

**สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง**

**เครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่**

**MULTIFUNCTION REMOTE CONTROL DUPLICATOR**



โดย  
นายเทิดศักดิ์ ตั้งบริบูรณ์รัตน์  
นายธนาวัฒน์ ตั้งบรรพิต  
นายวรัช สุขสมพรสกุล

๒๗.  
๗๗/๓๓  
๒๕๕๐  
เลขหมู่.....  
เลขทะเบียน 83296  
วัน,เดือน,ปี 11 ส.ค. 2551

11๑ ๖๖๖๕๖  
b.....  
i.....

**ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมเทคนิค  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่**  
**MULTIFUNCTION REMOTE CONTROL DUPLICATOR**

โดย

นายเทิดศักดิ์ ตั้งบริบูรณ์รัตน์ 47010292

นายธนาวัฒน์ ตั้งบรรพพิเชฐ 47010321

นายวรัช สุขสมพรสกุล 47010661

อาจารย์ที่ปรึกษา

ผศ.ดร.พิพัฒน์ พรหมมี

**ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต**  
**สาขาวิศวกรรมคอมพิวเตอร์**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ปีการศึกษา 2550**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2550

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่

**MULTIFUNCTION REMOTE CONTROL DUPLICATOR**

ผู้จัดทำ

- 1.นายเทิดศักดิ์ ตั้งบริบูรณ์รัตน์ 47010292
- 2.นายธนวัฒน์ ตั้งบรรพิตเวช 47010321
- 3.นายวัชร สุขสมพรตฤณ 47010661

.....

(ผศ.ดร.พิพัฒน์ พรหมมี)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## เครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่

### MULTIFUNCTION REMOTE CONTROL DUPLICATOR

โดย นายเทคศักดิ์ ตั้งบริบูรณ์รัตน์ 47010292  
นายธนาวัฒน์ ตั้งบวรพิเชฐ 47010321  
นายวรัช สุขสมพรสกุล 47010661

อาจารย์ที่ปรึกษา ผศ.ดร.พิพัฒน์ พรหมมี

#### บทคัดย่อ

โครงการนี้ได้นำเสนอเครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่ซึ่งประกอบด้วย 2 ส่วนสำคัญ คือ ส่วนตัวรับสัญญาณจากรีโมทคอนโทรลต้นแบบและส่วนของตัวส่งสัญญาณรีโมทควบคุม โดยใช้หลักการของระบบการรับส่งสัญญาณของรีโมทคอนโทรลแบบอินฟราเรดและไมโครคอนโทรลเลอร์ ซึ่งผู้ใช้งานสามารถนำเครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่ไปรับสัญญาณจากรีโมทคอนโทรลต้นแบบแล้วเครื่องจะทำการถอดรหัสสัญญาณและเก็บข้อมูลโดยไมโครคอนโทรลเลอร์ซึ่งจะทำให้สามารถทำงานได้เหมือนรีโมทคอนโทรลต้นแบบ โดยที่สามารถใช้งานกับรีโมทคอนโทรลเครื่องใช้ไฟฟ้าได้หลายชนิด

#### ABSTRACT

This project presents the Multifunction Remote Control Duplicator. It consists of 2 parts, part one is a signal receiver from prototype remote control. Part two is a signal transmitter. This project is based on infrared signal transceiver and microcontroller. The user can use Multifunction Remote Control Duplicator to receive signal from any prototype remote control then decode and retain their codes by microcontroller. The desired functions are actually same as a prototype remote control. Many types of remote control can be used for duplicating and applying

## สารบัญ

หน้า

บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ทฤษฎีอินฟราเรดรีโมทคอนโทรล	2
2.1.1 สัญญาณมาตรฐานรีโมทคอนโทรล	3
2.1.2 การใช้แสงอินฟราเรด	4
2.1.3 กระบวนการแผ่ (Radiative Process)	5
2.1.4 ตัวกำเนิดแสงอินฟราเรด	6
2.2 ระบบบัส I <sup>2</sup> C	7
2.2.1 ความรู้เบื้องต้นเกี่ยวกับ I <sup>2</sup> C	7
2.2.2 คุณสมบัติโดยทั่วไปของบัส I <sup>2</sup> C	7
2.2.3 หลักการของบัส I <sup>2</sup> C	8
2.2.4 สภาวะที่เกิดขึ้นบนบัส I <sup>2</sup> C	9
2.2.5 การทำงานบนบัส I <sup>2</sup> C	10
2.2.6 การอ้างถึงแบบ 7 บิต (7-bit addressing)	10
2.2.7 การเขียนโปรแกรมติดต่อบัส I <sup>2</sup> C	11
2.3 หน่วยความจำอีอีพรอมอนุกรม	12
2.3.1 รูปแบบการอ่านและเขียนข้อมูลของ 24LC512	13
2.3.2 การตรวจสอบสถานะการเขียนข้อมูล	15
2.3.3 การอ่านข้อมูล	17
2.4 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	19
2.4.1 การใช้งานพอร์ตต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51	20
2.4.2 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51	20
2.4.3 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช	21
2.4.4 ไทมเมอร์	21
2.5 แอลซีดี (LCD:Liquid Crystal Displays)	24
2.5.1 โครงสร้างภายในของตัวควบคุมโมดูลแอลซีดี	24
2.5.2 โมดูลแอลซีดีขนาด 16 ตัวอักษร 2 บรรทัด (แอลซีดี 16x2)	25
2.5.3 คำสั่งควบคุมโมดูลแอลซีดี	26
2.5.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูลแอลซีดี	29
2.5.5 จังหวะการทำงานของแอลซีดีโมดูล	29
2.6 สวิตช์เมตริกซ์ 4 x 3 จุด หรือคีย์แพด	30
2.7 โมดูลรับสัญญาณอินฟราเรดจากรีโมทคอนโทรล	31

## สารบัญ (ต่อ)

	หน้า
<b>บทที่ 3 การคำนวณและการสร้าง</b>	33
3.1 การออกแบบวงจรทางฮาร์ดแวร์	34
3.1.1 ส่วนประมวลผล	34
3.1.2 ภาคจ่ายไฟ	35
3.1.3 ส่วนของวงจรรับสัญญาณอินฟราเรดจากรีโมทคอนโทรล	36
3.1.4 ส่วนของวงจรส่งสัญญาณอินฟราเรด	36
3.1.5 ส่วนเก็บบันทึกสัญญาณ	38
3.1.6 ส่วนแสดงผล	39
3.1.7 ส่วนของคีย์เมตริกซ์	39
3.2 การออกแบบซอฟต์แวร์ควบคุมวงจร	40
3.2.1 การออกแบบซอฟต์แวร์ส่วนรับและบันทึกสัญญาณอินฟราเรด	40
3.2.2 การออกแบบซอฟต์แวร์ส่วนส่งสัญญาณอินฟราเรด	40
3.3 การจัดพื้นที่เก็บข้อมูลในหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC256	46
<b>บทที่ 4 การทดลองและการออกแบบ</b>	47
4.1 การทดลอง	47
4.1.1 ตอนที่ 1 การคัดลอกสัญญาณ	47
4.1.2 ตอนที่ 2 การส่งสัญญาณ	51
4.2 ผลการทดลอง	52
4.2.1 ผลการทดลองของรีโมทคอนโทรลโทรทัศน์สี Panasonic	52
4.2.2 ผลการทดลองของรีโมทคอนโทรลโทรทัศน์สี SONY	54
4.2.3 ผลการทดลองของรีโมทคอนโทรลเครื่องเล่นดีวีดี LG	56
<b>บทที่ 5 วิจารณ์และสรุปผล</b>	61
<b>ภาคผนวก</b>	
<b>หนังสืออ้างอิง</b>	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปภาพ

หน้า

รูปที่ 2.1 ลักษณะทางลอจิกในระบบสแกนพัลส์	2
รูปที่ 2.2 คาต้าสตรึม	3
รูปที่ 2.3 การไบอัสอินฟราเรด	5
รูปที่ 2.4 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆบนบัส I <sup>2</sup> C	7
รูปที่ 2.5 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบบัส I <sup>2</sup> C	8
รูปที่ 2.6 ไคอะแกรมเวลาแสดงสถานะต่างๆในบัส I <sup>2</sup> C	10
รูปที่ 2.7 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างอิงถึงแบบ 7 บิต	11
รูปที่ 2.8 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I <sup>2</sup> C เมื่อใช้การอ้างอิงถึงแบบ 7 บิต	11
รูปที่ 2.9 การจัดขาและข้อมูลกำหนดแอดเดรสของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512	13
รูปที่ 2.10 รูปแบบการเขียนข้อมูลแบบไบต์ของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512	14
รูปที่ 2.11 รูปแบบการเขียนข้อมูลแบบเพจของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512	14
รูปที่ 2.12 โฟลวชาร์ตแสดงวิธีการตรวจสอบบิต ACK	16
รูปที่ 2.13 รูปแบบการอ่านข้อมูลแบบแอดเดรสปัจจุบันของหน่วยความจำเบอร์ 24LC512	17
รูปที่ 2.14 รูปแบบการอ่านข้อมูลแบบสุ่มหรือแบบเจาะจงแอดเดรสของ 24LC512	18
รูปที่ 2.15 รูปแบบการอ่านข้อมูลแบบเรียงลำดับของ 24LC512	18
รูปที่ 2.16 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51	19
รูปที่ 2.17 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์เบอร์ AT89C52	21
รูปที่ 2.18 ไคอะแกรมการทำงานในโหมด 1 ของไทมเมอร์ 0	24
รูปที่ 2.19 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด	30
รูปที่ 2.20 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51	31
รูปที่ 2.21 โมดูลรับสัญญาณอินฟราเรดจากรีโมทคอนโทรล TSOP34838	32
รูปที่ 3.1 บล็อกไคอะแกรมของส่วนรับและบันทึกสัญญาณอินฟราเรด	33
รูปที่ 3.2 บล็อกไคอะแกรมของส่วนส่งสัญญาณอินฟราเรด	33
รูปที่ 3.3 การต่อใช้งานไมโครคอนโทรลเลอร์	34
รูปที่ 3.4 วงจรในส่วนของภาคจ่ายไฟ	35
รูปที่ 3.5 วงจรรับสัญญาณอินฟราเรดจากรีโมทคอนโทรล	36
รูปที่ 3.6 วงจรในส่วนของการส่งสัญญาณอินฟราเรด	38
รูปที่ 3.7 การต่อใช้งานหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512	38
รูปที่ 3.8 การต่อใช้งานโมดูลแอลซีดี	39
รูปที่ 3.9 วงจรภายในของคีย์แพดของโทรศัพท์ขนาด 4x3	39
รูปที่ 3.10 โฟลวชาร์ตในส่วนของโปรแกรมหลัก	41
รูปที่ 3.11 โฟลวชาร์ตในส่วนของโปรแกรมหลัก(ต่อ)	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูปรูปภาพ (ต่อ)

หน้า

รูปที่ 3.12 โฟลวชาร์ตโปรแกรมย่อยรับสัญญาณอินฟราเรดจากกรีโมทคอนโทรลมาเก็บในแรม	43
รูปที่ 3.13 โฟลวชาร์ตโปรแกรมย่อยสำหรับส่งสัญญาณอินฟราเรด	44
รูปที่ 3.14 โฟลวชาร์ตโปรแกรมย่อยสำหรับบันทึกข้อมูลลงในอีอีพรอม	45
รูปที่ 4.1 จอแอลซีดีแสดงข้อความให้เลือกว่าจะคัดลอกสัญญาณหรือใช้งานส่งสัญญาณ	47
รูปที่ 4.2 จอแอลซีดีแสดงข้อความรอรับสัญญาณจากกรีโมทคอนโทรล	47
รูปที่ 4.3 จอแอลซีดีแสดงข้อความให้เลือกว่าจะทดสอบสัญญาณหรือบันทึกสัญญาณ	48
รูปที่ 4.4 จอแอลซีดีแสดงข้อความให้เลือกว่าจะบันทึกในโหมดเครื่องปรับอากาศหรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ	48
รูปที่ 4.5 จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะเก็บบันทึกข้อมูล	48
รูปที่ 4.6 จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิสามารถบันทึกได้ตั้งแต่อุณหภูมิ 15-30 องศาเซลเซียส	49
รูปที่ 4.7 จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ	49
รูปที่ 4.8 จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสำรองสำหรับปุ่มอื่นๆ ที่เหลือ	49
รูปที่ 4.9 จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกตำแหน่งที่จะบันทึก	50
รูปที่ 4.10 จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกปุ่มที่จะบันทึก	50
รูปที่ 4.11 จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกปุ่มต่อไป หรือกลับไปเมนูหลัก	50
รูปที่ 4.12 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากกรีโมทคอนโทรลต้นแบบส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	52
รูปที่ 4.13 สัญญาณเมื่อกดปุ่ม Channel 7 โดยสัญญาณรูปบน เป็นสัญญาณจากกรีโมทคอนโทรลต้นแบบส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	52
รูปที่ 4.14 สัญญาณเมื่อกดปุ่มเพิ่มเสียง โดยสัญญาณรูปบน เป็นสัญญาณจากกรีโมทคอนโทรลต้นแบบส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	53
รูปที่ 4.15 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว	53
รูปที่ 4.16 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากกรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	54
รูปที่ 4.17 สัญญาณเมื่อกดปุ่ม Channel 7 โดยสัญญาณรูปบน เป็นสัญญาณจากกรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	54
รูปที่ 4.18 สัญญาณเมื่อกดปุ่มเพิ่มเสียง โดยสัญญาณรูปบน เป็นสัญญาณจากกรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	55
รูปที่ 4.19 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว	55

## สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 4.20 สัญญาณเมื่อคัปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	56
รูปที่ 4.21 สัญญาณเมื่อคัปุ่ม Play โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	56
รูปที่ 4.22 สัญญาณเมื่อคัปุ่ม Stop โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	57
รูปที่ 4.23 สัญญาณเมื่อคัปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว	57
รูปที่ 4.24 สัญญาณเมื่อคัปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	58
รูปที่ 4.25 สัญญาณเมื่อคัปุ่มเพิ่มอุณหภูมิเป็น 25 องศาเซลเซียส โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	58
รูปที่ 4.26 สัญญาณเมื่อคัปุ่มลดอุณหภูมิเป็น 24 องศาเซลเซียส โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก	59
รูปที่ 4.27 สัญญาณเมื่อคัปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว	59
รูปที่ 4.28 รูปสาคิการทำงานของเครื่องคัดลอกรีโมทคอนโทรล	60

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 Some of the material for LED device	5
ตารางที่ 2.2 บิตภายใน TMOD	22
ตารางที่ 2.3 บิตภายใน TCON	23
ตารางที่ 2.4 คำสั่งเลือกโหมดการป้อนข้อมูลของแอลซีดี	26
ตารางที่ 2.5 คำสั่งควบคุมการแสดงผลของแอลซีดี	26
ตารางที่ 2.6 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษรของแอลซีดี	27
ตารางที่ 2.7 คำสั่งกำหนดฟังก์ชันการทำงานของแอลซีดี	27
ตารางที่ 2.8 คำสั่งอ่านแฟลค BUSY และแอดเดรสของแอลซีดี	28



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

ในโลกยุคปัจจุบันเต็มไปด้วยเครื่องอำนวยความสะดวกต่างๆ มากมาย เพื่อช่วยให้การดำเนินชีวิตมีความสะดวกสบายมากยิ่งขึ้น โดยที่เครื่องอำนวยความสะดวกที่ใช้ไฟฟ้ามีใช้กันอย่างแพร่หลาย และเพื่อความสะดวกสบายมากยิ่งขึ้น ในการควบคุมเครื่องใช้ไฟฟ้าเหล่านี้มักนิยมใช้รีโมทคอนโทรลเป็นตัวควบคุม เนื่องจากใช้งานง่าย เครื่องส่งใช้กำลังไฟน้อยและราคาไม่แพง

แต่ยิ่งความนิยมในการใช้งานรีโมทคอนโทรลเพื่อควบคุมเครื่องใช้ไฟฟ้ามีมากขึ้น ทำให้ภายในบ้านต้องมีจำนวนรีโมทคอนโทรลที่เพิ่มมากขึ้นตามจำนวนเครื่องใช้ไฟฟ้าภายในบ้าน ทำให้ดูเกะเกะ กลายเป็นการลดความสะดวกสบายของการใช้รีโมทคอนโทรลซึ่งผิดจุดประสงค์ หรือเมื่อมีรีโมทคอนโทรลอยู่หลายอันการเก็บรักษาก็ยุ่งยาก จนอาจทำให้รีโมทคอนโทรลสูญหายได้

จากปัญหาที่ได้กล่าวมาข้างต้น จึงเป็นความคิดที่ดีที่จะมีรีโมทคอนโทรลเพียงตัวเดียวที่สามารถควบคุมเครื่องใช้ไฟฟ้าได้หลายๆ ชนิด ปัญหาในการใช้งานรีโมทคอนโทรลก็จะหมดไป แล้วยังเพิ่มความสะดวกสบายมากยิ่งขึ้น การเก็บรักษาก็ง่ายขึ้น นอกจากนั้นหากทำการเก็บข้อมูลของรีโมทคอนโทรลหลายๆ อันไว้ ถ้าอันใดอันหนึ่งเกิดชำรุดเสียหาย หรือไม่สามารถใช้งานได้ ก็ไม่ต้องไปหาซื้อใหม่ให้ยุ่งยาก

เครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่ ได้นำไมโครคอนโทรลเลอร์มาเป็นแกนหลักในการสร้าง เพราะมีความเหมาะสมในหลายๆ ด้าน ทั้งหาซื้อง่าย การประมวลผลรวดเร็ว มีขนาดเล็ก และมีฟังก์ชันการใช้งานที่ง่ายและเป็นประโยชน์กับโครงการนี้ โดยการใช้งานเครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่นี้ก็ไม่ยากนัก มีคำอธิบายการทำงานที่แสดงผลผ่านจอแอลซีดีทุกขั้นตอน ใช้แหล่งจ่ายไฟคือถ่านไฟฉายขนาด 9 โวลต์ มีหน่วยความจำอีพรอมเป็นตัวเก็บข้อมูล ทำให้สามารถเก็บรักษาข้อมูลอยู่ได้แม้ว่าจะหยุดจ่ายไฟให้กับอุปกรณ์ ใช้คีย์แพคของโทรศัพท์แทนปุ่มกดบนรีโมทคอนโทรล

## บทที่ 2 ทฤษฎีและหลักการ

### 2.1 ทฤษฎีอินฟราเรดรีโมทคอนโทรล

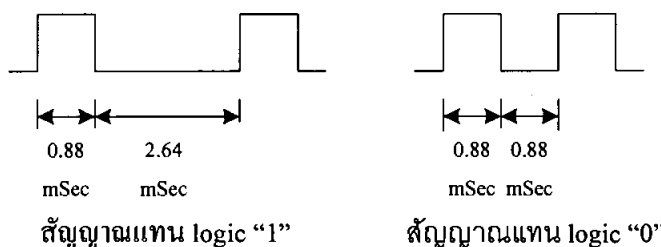
รีโมทคอนโทรลหมายถึง ระบบควบคุมระยะไกล ที่ทำหน้าที่เสมือนแขนขาของมนุษย์ จึงทำให้เป็นที่นิยมใช้กันอย่างแพร่หลาย และเริ่มเป็นสิ่งจำเป็นควบคู่กับเครื่องใช้ไฟฟ้าในยุคนี้จึงจะให้ความสนใจกันอย่างละเอียด การทำงานของระบบคาส์โค้ด

ระบบตัวเลขที่ใช้ในรีโมทคอนโทรลเป็นตัวเลขฐาน 2 ซึ่งมีสภาพทางลอจิก “0” และ “1” หรือหากพูดถึงระดับแรงไปสภาพลอจิก 1 ก็คือสภาพที่วงจรสามารถจ่ายแรงดันไฟออกไปยังอุปกรณ์ในวงจรได้ ส่วนสภาพลอจิก 0 ก็คือสภาพที่แรงดันไฟใกล้เคียงศูนย์โวลต์ นั่นคือระบบสัญญาณพัลส์นั่นเอง

กล่าวเฉพาะในส่วนของคุณสมบัติ การที่จะแยกออกมาได้อย่างชัดเจนว่าอันไหนเป็นลอจิก 0 อันไหนเป็นลอจิก 1 จำเป็นต้องใช้ฐานเวลาเข้ามาเป็นตัวแยกในทางปฏิบัติเป็นเรื่องค่อนข้างยากอยู่พอสมควร เนื่องจากบางครั้งผู้ใช้รีโมทอาจจะกดซ้ำกันแต่ละครั้งอาจไม่เท่ากัน บางคนอาจกดแค่ไว้ บางคนอาจกดก็ยแล้วปล่อยเลย

ดังนั้นเราจึงเลือกเอาระบบสแกนพัลส์ซึ่งเป็นรูปแบบการกวาดข้อมูลมาทดแทน เพื่อแก้ปัญหาเรื่อง การที่จะต้องกำหนดฐานเวลา โดยหากเป็นลอจิก 0 เราจะให้พัลส์ที่ออกมาเป็นพัลส์ชิดหรือแคบกว่าและหากถ้าเป็นลอจิก 1 เราจะให้ระยะของพัลส์ทอดยาวออกไป การกำหนดฟังก์ชันทำได้โดยการใช้หลักการของคีย์แบบเมตริกซ์ (Matrix) เพื่อลดข้อใช้งานของไอซีและสายเชื่อมโยงต่างๆ ให้น้อยลง

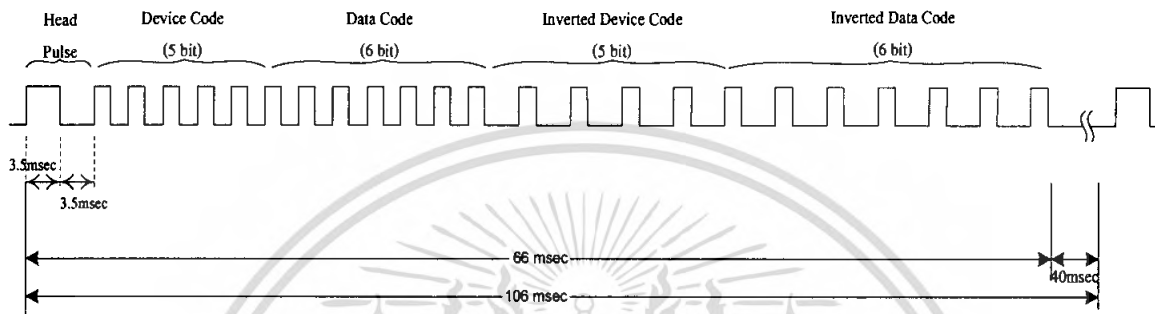
เมื่อเรากำหนดพัลส์ในลักษณะที่กล่าวมา มีผลทำให้ฐานเวลาของข้อมูล (Data Time) แต่ละตัวมีความแตกต่างกันออกไป ยกตัวอย่างเช่นในกรณีที่ข้อมูลนั้นมีสภาพลอจิก 0 เป็นจำนวนมาก บิตจะมีผลทำให้เวลาของข้อมูลสั้น หากมีสภาพทางลอจิก 1 มากกว่าจะมีผลเวลาของข้อมูลยาวนานขึ้น ดังนั้นในยุคปัจจุบันที่ใช้ข้อมูลหลายๆบิต จะเกิดปัญหาในเรื่องความเหลื่อมของเวลาขึ้น ปัญหานี้มีผลต่อการรับข้อมูลในเครื่องรับเป็นอันมาก รวมไปถึงหากเราจะใส่ข้อมูลอย่างอื่นฝากไปด้วยจะทำให้ลำบาก เราจะพบว่ารีโมทคอนโทรลในปัจจุบันนี้ สามารถผ่านข้อมูลพิเศษเข้าไปได้มากที่สุดทีเดียว เครื่องรับแยกได้อย่างไรหากเวลาไม่แน่นอนจึงต้องมีการอินเวิร์ต (Inverse) ข้อมูลทั้งหมดให้กลายเป็นตรงข้ามเพื่อรักษาเวลาให้คงที่ และเพื่อไม่ต้องส่งสัญญาณชิงโครโนซ์ไปทำการควบคุมเครื่องรับซึ่งนับเป็นความซับซ้อนยุ่งยาก



รูปที่ 2.1 ลักษณะทางลอจิกในระบบสแกนพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่างเช่นเราส่งข้อมูลหรือคาค่าไค้คเป็นข้อมูลขนาด 6 บิต เป็นดังนี้ 000001 เราจะพบว่าเมื่อถูกแปรสภาพเป็นพัลส์แล้ว ลอจิก 0 มีจำนวน 5 บิต มีลอจิก 1 จำนวน 1 บิต จะพบว่าเวลารวมของพัลส์สั้นมาก และหากเราส่งข้อมูลเป็น 111111 จะพบว่าเวลารวมของพัลส์จะยาวมาก แต่ถ้าหากเรานำมาอินเวิร์ตจะพบว่าเวลาทั้งหมดเท่ากัน นั่นคือข้อมูล 000001 เมื่อนำมาอินเวิร์ตแล้วจะเป็น 111110 ข้อมูลรวมคือ 000001111110 (แยกเป็นลอจิก 0 รวม 6 บิต ลอจิก 1 รวมเป็น 6 บิต) พบว่าระยะเวลาของข้อมูลจะเท่ากันโดยอัตโนมัติ ดังนั้นวิธีการอินเวิร์ตข้อมูลคือการแก้ปัญหาฐานเวลาที่เกิดขึ้นกับเครื่องรับหรือตัวรับนั่นเอง



รูปที่ 2.2 คาค่าสตรีม

### 2.1.1 สัญญาณมาตรฐานรีโมทคอนโทรล

การที่เราจะให้วงจรขารับของรีโมทคอนโทรลรับรู้และทำการแยกสัญญาณหรือข้อมูลต่างๆ ได้อย่างถูกต้องจะต้องมีข้อมูลตัวอื่นเข้ามาเพื่อแก้ไขสิ่งซึ่งอาจผิดพลาด ดังนั้นนอกจากข้อมูล (Data) ซึ่งเป็นข้อมูลหลักที่เราส่งไปเป็นตัวเลขฐาน 2 ซึ่งในตอนนี้อยู่ในรูปของสแกนพัลส์พร้อมด้วยการอินเวิร์ตข้อมูลเพื่อรักษาค่าเวลาแล้วยังต้องมีข้อมูลอื่นเป็นส่วนประกอบ ซึ่งในกรณีนี้เราขอใช้ตัวอย่างของเครื่องเนชั่นแนลเป็นตัวเดินเรื่อง (ซึ่งโดยหลักการจะเหมือนกันทุกยี่ห้อ เพียงแต่ว่าใครจะใช้ข้อมูลกี่บิตและใช้เวลานานเท่าไรนั่นเอง) นี่คือการยกตัวอย่างเพื่อนำไปสู่การทำความเข้าใจ และทำการเปรียบเทียบกับยี่ห้ออื่นๆต่อไป

1.คาค่าไค้ค (Data Code) ในกรณีของรีโมทคอนโทรลซึ่งเป็นตัวส่งของเนชั่นแนลใช้ข้อมูลขนาด 6 บิต เป็นข้อมูลหลักที่ส่งออกไปควบคุมวงจรในส่วนขารับโดยคีย์เมตริกซ์จะเป็นตัวส่งงานเข้าสู่ระบบการเข้ารหัสข้อมูล กำหนดความเป็นไปของแต่ละฟังก์ชัน

2.อินเวิร์ตคาค่าไค้ค (Inverse Data Code) เป็นการกลับลอจิกของข้อมูลหลักเพื่อรักษารฐานเวลาให้คงที่ทุกข้อมูล ซึ่งเป็นข้อมูลขนาด 6 บิตเหมือนกัน

3.ดีไวส์ไค้ค (Device Code) หรือบางครั้งใช้ศัพท์ “คัสตอม ไค้ค” (Custom Code) เนื่องจากในปัจจุบันเครื่องใช้ต่างๆ ล้วนเป็นระบบรีโมทคอนโทรลแบบอินฟราเรดกันทั้งนั้น การสั่งงานจากรีโมทคอนโทรลอาจจะมีกรรบกวนอุปกรณ์อื่นๆได้ อย่างเช่นเครื่องรับโทรทัศน์อาจจะพ่วงอยู่กับเครื่องเล่นวีดีโอเทป จูนเนอร์ พ่วงอยู่กับเครื่องขยายเสียง และคอมแพคดิสก์ เลเซอร์ดิสก์พ่วงอยู่กับเครื่องรับโทรทัศน์ หรืออื่นๆ กรณีเช่นนี้ หากเราส่งเครื่องหนึ่งเครื่องใด เครื่องที่ต่อรวมอยู่ด้วยก็สามารถมารับรับเอาข้อมูลฟังก์ชันการทำงานเข้าไปด้วย จึงมีการสร้างข้อมูลของ เครื่องเล่นแต่ละอย่างให้แตกต่างกันออกไป ตัวอย่างเช่น เครื่องรับโทรทัศน์เราใช้ดี

ไวส์โค้ด (Device Code) ซึ่งเป็นโค้ดที่ใช้แยกประเภทของเครื่องใช้ระบบ 00000 (5 บิต) ในขณะที่เครื่องเล่นวีดีโอเทปเราใช้ไวส์โค้ด 11111

4. อินเวอร์ตติไวส์โค้ด (Inverse Device Code) เป็นการกลับข้อมูลติไวส์โค้ดเพื่อรักษาเวลาเช่นเดียวกับระบบข้อมูลหลัก (Data Code) แน่นอนข้อมูลดังกล่าวต้องมี 5 ชนิด เหมือนติไวส์โค้ด

5. เฮดพัลส์ (Head Pulse) การกลับข้อมูลหรืออินเวอร์ตเป็นเพียงการรักษาเวลาของข้อมูล แต่การใส่เฮดพัลส์เป็นวิธีการตรวจเช็คข้อมูลเพิ่มความแน่นอนของข้อมูลขึ้นอีกเพราะในบางครั้งเราอาจจะกดคีย์แชไว้นานๆ ความต่อเนื่องของข้อมูลจะมีตลอด นั่นหมายความว่าเราจะแยกแยะได้อย่างไร ว่าอะไรคือข้อมูลหลัก อะไรคือข้อมูลรอง จึงมีเฮดพัลส์ขึ้นมา โดยเฮดพัลส์จะเป็นสัญญาณนำร่องก่อนจะมีข้อมูลต่างๆส่งเข้ามา และในกรณีที่เราส่งอย่างต่อเนื่องจะมีเฮดพัลส์ ส่งออกมาก็ค้นเป็นช่วงๆ ให้เครื่องรับสามารถแยกแยะกลุ่มข้อมูลออกมาได้

รีโมทคอนโทรลจะส่งสัญญาณออกมาแต่ละคีย์ที่กดจะใช้เวลาที่เท่ากัน เนื่องจากมีการอินเวอร์ตติไวส์โค้ด และ อินเวอร์ตคาค้าโค้ด ตามลำดับคาค้าสตรีม (Data Stream) หรือระบบในการเรียงข้อมูล จะเป็นไปตามรูปที่ 2.2 โดยเฮดพัลส์จะเข้ามาเป็นอันดับแรกเป็นการบอกว่าตอนนี้รอบชั้นภายในตัวส่งสัญญาณรีโมทคอนโทรลแบบอินฟราเรด พร้อมทั้งจะทำงานแล้วซึ่งการที่ตัวตรวจสอบหรือตัวกระตุ้นความพร้อมของตัวรับว่าต่อไปนี้จะมีการส่งข้อมูล

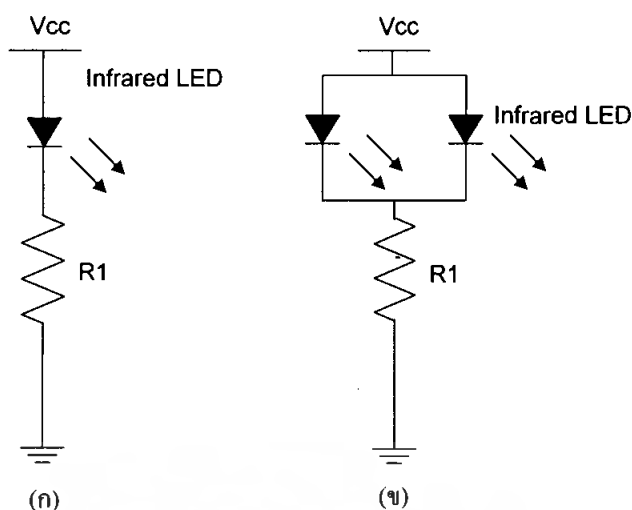
เมื่อเครื่องรับรับรู้การทำงาน (ซึ่งเรากดคีย์ออกไป สัญญาณเฮดพัลส์บางที่เราเรียกว่าตัว “คอลล์”) เครื่องรับแสดงการรับรู้การเรียกข้อมูล (Call) ด้วยการกระพริบของแอลอีดี และข้อมูลลำดับต่อมาคือติไวส์โค้ด 5 บิตตามด้วยคาค้าโค้ด 6 บิต อินเวอร์ตติไวส์ 5 บิตและปิดท้ายด้วยอินเวอร์ตคาค้าโค้ด 6 บิตก่อนทิ้งช่วงให้เกิดเฮดพัลส์ครั้งต่อไป ให้ตัวรับสามารถรับรู้ความต่อเนื่องของข้อมูลอีกครั้งหรือหลายๆครั้งต่อไป

การจะให้ไอซี ที่เป็นตัวส่งรีโมทคอนโทรลแบบรับรู้รหัสข้อมูลใดๆนั้น วงจรเอนโค้ดเดอร์จะต้องทำการเข้ารหัส โดยการนำเอาสแกนพัลส์ที่เกิดจากวงจรทมิ้งเจนเนอเรเตอร์ไปเป็นตัวอ้างอิงให้คีย์เมตริกแต่ละตัวส่งข้อมูลที่ไม่ซ้ำซ้อนกันเข้าสู่วงจรเข้ารหัสข้อมูล และจะมีการเรียงลำดับข้อมูลต่างๆ ให้เป็นขั้นตอนที่ถูกต้องเนื่องจากรีโมทคอนโทรลแบบนี้เป็นการส่งแบบไร้สายจึงอาจเกิดการรบกวนในระหว่างที่ความถี่เดินทางได้ (ความถี่ของแสงอินฟราเรด หรือ ความถี่ต่ำกว่าสีแดง) จึงต้องมีการผสมความถี่วิทยุพ่วงเข้าไปในขั้นตอนสุดท้าย

### 2.1.2 การใช้แสงอินฟราเรด

การใช้แสงอินฟราเรดเป็นสัญญาณควบคุมของรีโมทคอนโทรล สามารถตัดปัญหาการรบกวนแสงภายนอกอื่นๆ (ที่มองเห็น) ลงไปได้โดยเด็ดขาด ยิ่งไปกว่านั้นวงจรใช้งานของระบบอินฟราเรดยังเป็นวงจรที่ง่ายและไม่ซับซ้อน มีความเชื่อถือได้สูงในการใช้งาน และไม่เกิดอันตรายต่อผู้ใช้

การส่งสัญญาณแสงอินฟราเรดสามารถทำได้ด้วยวงจรง่ายๆ ดังรูปที่ 2.3(ก) ซึ่งประกอบด้วย แอลอีดีที่เปล่งแสงในย่านอินฟราเรดต่อเข้ากับแหล่งจ่ายไฟ โดยความต้านทาน R1 ทำหน้าที่จำกัดกระแส



รูปที่ 2.3 การไบอัสอินฟราเรด

สัญญาณแสงที่ส่งออกโดยแอลอีดีเพียงตัวเดียวจะเหมาะกับการใช้งานที่ไม่กี่เมตรเท่านั้น การเพิ่มกำลังส่งของอินฟราเรดให้ไปได้ไกลขึ้นนั้น ทำได้โดยใช้แอลอีดีหลายตัวต่อขนานกันดังรูปที่ 2.3(ข) โดยที่ R1 จะต้องมีค่าลดลงจากเดิมเพราะต้องขับกระแสมากขึ้น ปัจจุบันมีแอลอีดีอินฟราเรดรุ่นใหม่ ที่ให้กำลังส่งหรือความเข้มแสงสูงช่วยให้ส่งสัญญาณได้ไกลกว่าเดิม

### 2.1.3 กระบวนการแผ่ (Radiative Process)

1. เกิดจากอิเล็กตรอน และ โฮล และส่วนที่อยู่ใกล้กับจังก์ชัน (junction) ซึ่งจะให้พลังงานโฟตอนประมาณเท่ากับแบนด์แกปเอเนอร์จี (Band Gap Energy) ซึ่งมีค่าค่อนข้างสูง

2. การที่อิเล็กตรอน และ โฮลจะเคลื่อนที่มารวมกันบริเวณแทรปเลเวล (Trap Level) โฟตอนจะมีพลังงานออกมาเท่ากับเอเนอร์จีแกป (Energy Gap) ลบออกด้วยค่าไบนด์จิงเอเนอร์จี (Binding Energy) และจากกระบวนการทั้งสองจะได้โฟตอนออกมา ซึ่งจะมีค่าความยาวคลื่นตามสมการข้างล่างนี้

$$\lambda = \frac{1240}{\Delta E}$$

$\Delta E$  = ค่าความแตกต่างของพลังงาน หน่วยเป็นอิเล็กตรอนโวลต์

เพราะฉะนั้นเมื่อแอลอีดีมีอุณหภูมิเปลี่ยนไปจะทำให้  $\Delta E$  เปลี่ยนแปลงไปด้วยค่าความยาวคลื่นจึงเปลี่ยนไป จากที่กล่าวมาจะเห็นได้ว่าถ้าใช้สารต่างชนิดกัน ค่า  $\Delta E$  จะแตกต่างกันออกไป ซึ่งเราสามารถดูได้จากตารางที่ 2.1

ตารางที่ 2.1 Some of the material for LED device

Material	Band Gap Energy	Emission (nm)	Transition Type
Ge	0.66	1880	Indirect
Si	1.09	1140	Indirect
GaAs	1.43	910	Direct

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 Some of the material for LED device (ต่อ)

Material	Band Gap Energy	Emission (nm)	Transition Type
Gap	2.24	560	Indirect
GaAs60P40	1.91	650	Direct
AlSb	0.60	775	Indirect
InSb	0.18	6900	Direct
SiC	2.2-3.0	563.413	Indirect

จากตาราง เป็นการแสดงถึงแอลอีดีชนิดต่างๆซึ่งให้โฟตอน ที่มีความยาวคลื่นต่างๆกันและจะเห็นได้ว่าแอลอีดีที่สร้างจากแกเลียมอาร์เซไนด์ (GaAs) จะให้คลื่นอยู่ในช่วงอินฟราเรด ซึ่งมีค่าประมาณ 900 nm

### 2.1.4 ตัวกำเนิดแสงอินฟราเรด

จากหลักการกำเนิดแสงอินฟราเรด โดยนำเอาอุปกรณ์ชนิดหนึ่งที่สามารถเปลี่ยนพลังงานไฟฟ้าเป็นพลังงานแสง ซึ่งจะได้ความยาวคลื่นแสงที่ออกจากตัวอุปกรณ์ตัวนี้ที่มีความยาวอยู่ในช่วงอินฟราเรด ดังที่กล่าวมาอุปกรณ์ตัวนี้คือ Infrared Light Emitter Diode (Infrared LED) ซึ่งมีข้อดีหลายประการคือ

1. กระแสและแรงดันที่ใช้ในการผลิตคลื่นมีน้อยมาก
2. สามารถควบคุมการทำงานของแอลอีดีให้มีค่าแน่นอน
3. มีสปีดเรสปอนส์ (Speed Response) สูง

ลักษณะอินฟราเรดแอลอีดี ที่เหมือนกับแอลอีดีทั่วไป คือ จะเป็นสารกึ่งตัวนำที่ถูกโคป โดยสารที่มีอิเล็กตรอนชั้นนอกสุด (Valence electron) ต่างกัน โดยสารที่มีอิเล็กตรอนชั้นนอกสุดมากกว่าปกติจะเป็น N-type และสารที่มีอิเล็กตรอนน้อยกว่าปกติจะเป็นสารประเภท P-type เมื่อนำสารทั้งสองชนิดมาต่อกัน จะทำให้เกิดการแบ่งกันระหว่างอิเล็กตรอนขึ้นทำให้เกิดความแตกต่างศักดาไฟฟ้าที่เรียกว่าโพเทนเชียลแบเรียเออร์ (Potential Barrier) ซึ่งมีค่าน้อยกว่าขนาดของเอนเนอร์จี้แกปดังนั้นอิเล็กตรอนจากด้าน N-type จะเคลื่อนย้ายมายังด้าน P-type ต้องใช้พลังงานค่าหนึ่ง เมื่อเราป้อนแรงดันค่าหนึ่งให้กับแอลอีดีจะมีอิเล็กตรอนเคลื่อนที่จากด้าน N-type มารวมกับโฮลทางด้าน P-type ซึ่งการรวมตัวกันนี้มีอยู่ 2 ลักษณะ คือ กระบวนการแผ่ (Radiative Process) พลังงานจะแสดงออกมาในรูปโฟตอนหรือความร้อน

หลักการของตัวรับแสงอินฟราเรดเราใช้อุปกรณ์รับแสง (Detector) คือสามารถเปลี่ยนพลังงานแสงที่รับเข้ามาไปเป็นพลังงานไฟฟ้า อุปกรณ์ประเภทนี้เช่น โฟโตไดโอด (Photo Diode), โฟโตทรานซิสเตอร์ (Photo Transistor) เป็นต้น จัดเป็นสวิตซ์อิเล็กทรอนิกส์ทางแสง แต่อุปกรณ์ที่นำมาใช้ในวงจรระบบควบคุมโดยมากคือ โฟโตไดโอด และโฟโตทรานซิสเตอร์ เป็นอุปกรณ์รับแสงในระบบควบคุมระยะไกลแบบไร้สาย (Remote Control by Optoelectronic) ซึ่งจะรับแสงในย่านอินฟราเรดเป็นส่วนมาก เพราะว่าจะไม่ทำให้สภาพแสงตามธรรมชาติเข้าไปรบกวนในการควบคุมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

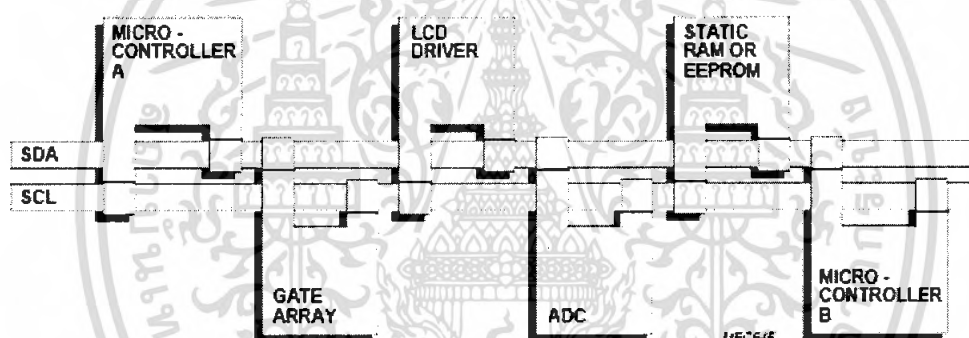
## 2.2 ระบบบัส I<sup>2</sup>C

### 2.2.1 ความรู้เบื้องต้นเกี่ยวกับ I<sup>2</sup>C

I<sup>2</sup>C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I<sup>2</sup>C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลัก คือ ต้องการให้ไอซีหรือ โมดูลสามารถติดต่อ ส่งงาน และควบคุมภายใต้สัญญาณเพียง 2 เส้น เส้นหนึ่ง คือ สายข้อมูล อีกเส้นหนึ่ง คือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์แบบบัส I<sup>2</sup>C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสถานะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I<sup>2</sup>C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock line)

ในรูปที่ 2.4 แสดงผังของการเชื่อมต่ออุปกรณ์ต่างๆบนบัส I<sup>2</sup>C จะเห็นได้ว่า อุปกรณ์ที่ทำการเชื่อมต่อบนบัส I<sup>2</sup>C มีหลากหลาย เช่น ไอซีแปลงสัญญาณแอนาล็อกเป็นดิจิทัล (ADC), ไอซีขับโมดูล แอลซีดี, หน่วยความจำอีพรอม และ ไมโครคอนโทรลเลอร์

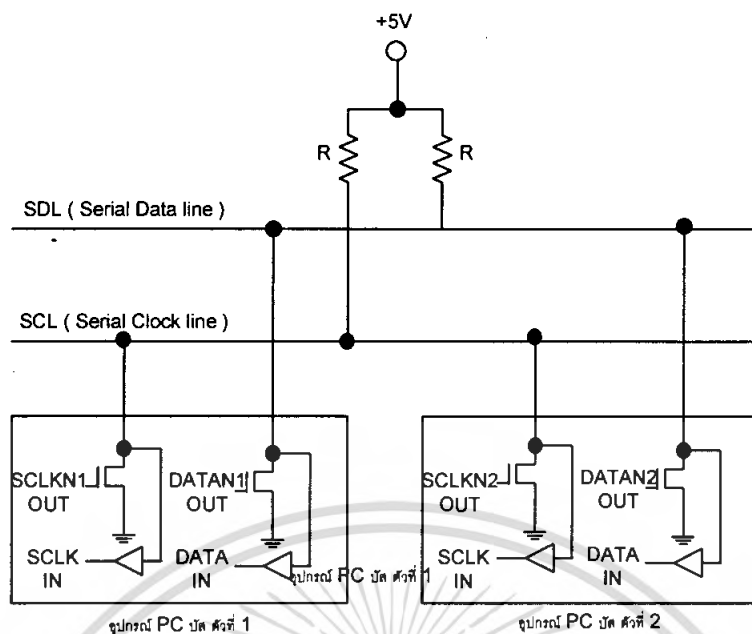


รูปที่ 2.4 ผังแสดงการเชื่อมต่อของอุปกรณ์ต่างๆบนบัส I<sup>2</sup>C

### 2.2.2 คุณสมบัติโดยทั่วไปของบัส I<sup>2</sup>C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5V ไว้ตลอดเวลาเพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งานทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสอง วงจรเอาต์พุตของอุปกรณ์ที่ต่ออยู่บนบัส I<sup>2</sup>C ต้องมีลักษณะเป็นวงจรทรานซิสเตอร์เปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collector) ดังแสดงรายละเอียดในรูปที่

2.5



รูปที่ 2.5 แสดงวงจรเอาต์พุตของอุปกรณ์ในระบบบัส I<sup>2</sup>C

อัตราการถ่ายเทข้อมูลบนบัส I<sup>2</sup>C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (fast mode) อุปกรณ์ที่ต่ออยู่บนบัส I<sup>2</sup>C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I<sup>2</sup>C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

ข้อเด่นอีกประการหนึ่งของบัส I<sup>2</sup>C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนบัส I<sup>2</sup>C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อร่วมกันบนบัส I<sup>2</sup>C สามารถกระทำได้ในลักษณะทำได้ในลักษณะเดียวกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องต่อตัวต้านทานพูลอัพเข้ากับแรงดัน +5V ไว้ด้วยเสมอ

### 2.2.3 หลักการของบัส I<sup>2</sup>C

บัส I<sup>2</sup>C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่กล่าวมาแล้วคือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โพรโตคอล (protocol) เพื่อให้ผู้ที่ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่งต่อไปนี้จะอธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I<sup>2</sup>C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I<sup>2</sup>C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (transmitter)

อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (receiver) ในอุปกรณ์บนบัส I<sup>2</sup>C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงตัวเดียว จะไม่มีอุปกรณ์ใดบนบัส I<sup>2</sup>C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I<sup>2</sup>C เรียกว่า มาสเตอร์ (master)

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าบนบัส I<sup>2</sup>C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I<sup>2</sup>C

(1) การถ่ายทอข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

(2) ในระหว่างการถ่ายทอข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

## 2.2.4 สถานะที่เกิดขึ้นบนบัส I<sup>2</sup>C

มีด้วยกัน 5 สถานะ ดังนี้

(1) บัสว่าง (Bus not busy) สถานะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายทอข้อมูลสามารถเริ่มต้นขึ้นได้

(2) เริ่มต้นการถ่ายทอข้อมูล (start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นนี้ว่า “สถานะเริ่มต้น (START)”

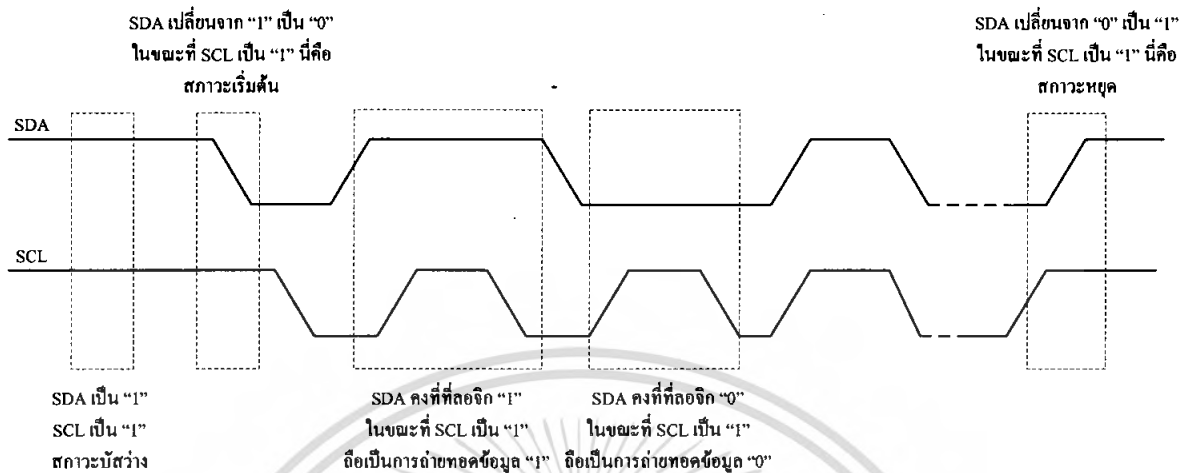
(3) หยุดการถ่ายทอข้อมูล (stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสถานะที่เกิดขึ้นว่า “หยุด (STOP)”

(4) ข้อมูลดำรงอยู่บนบัส (data valid) สถานะนี้เกิดขึ้นถัดจากสถานะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการถ่ายทอ เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่า เป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL มีลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่สาย SCL มีลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอข้อมูลจะแปลความหมายเป็นสถานะหยุดหรือสถานะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอนั้นเกิดความผิดพลาดขึ้น

(5) รับรู้ข้อมูล (acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า “บิตรับรู้ (acknowledge bit)” มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจะส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังติดต่ออยู่ในขณะนั้นก็จะกำเนิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.6 เป็นไคอะแกรมเวลาที่แสดงถึงการเกิดสถานะต่างๆ บนบัส I<sup>2</sup>C ไม่ว่าจะเป็นสภาวะบัสว่าง, เริ่มต้น, ถ่ายทอดข้อมูล, รั้บรู้ และหยุดการถ่ายทอดข้อมูล



รูปที่ 2.6 ไคอะแกรมเวลาแสดงสถานะต่างๆ ในบัส I<sup>2</sup>C

### 2.2.5 การทำงานบนบัส I<sup>2</sup>C

ก่อนที่จะเริ่มต้นการถ่ายทอดข้อมูลระหว่างอุปกรณ์ต่างๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงเสียก่อนโดยการอ้างถึงอุปกรณ์บนบัส I<sup>2</sup>C นั้นจะใช้การอ้างถึงแบบ 7 บิต หรือ 10 บิต ในกรณีที่มิ้ออุปกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตเพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่้อุปกรณ์แต่ละตัวได้เรียบร้อยแล้วก็จะเริ่มต้นการถ่ายทอดข้อมูลกันต่อไป

ดังนั้นหัวใจสำคัญในอันดับแรกของการทำงานบนบัส I<sup>2</sup>C คือ การอ้างถึงอุปกรณ์แต่ละตัวซึ่งในที่นี้จะอธิบายรายละเอียดของการอ้างถึงเฉพาะแบบ 7 บิตเท่านั้น

### 2.2.6 การอ้างถึงแบบ 7 บิต (7-bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้นคือข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อหรือ ข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงดังรูปที่ 2.7 ใน 7 บิตบนรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้้อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้นๆ หากบิต LSB เป็น "0" หมายถึงการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ

ข้อมูลไบต์ต่อมา คือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่างไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจร ADC หรือ DAC เป็นต้น

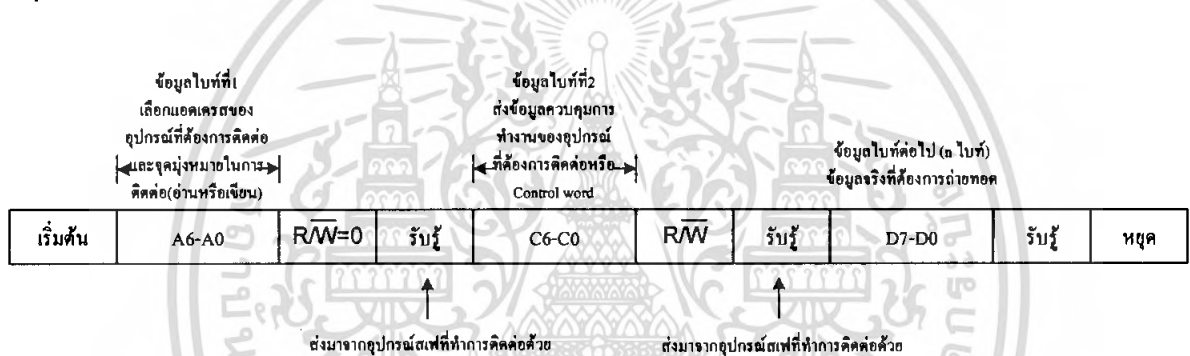
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
X	X	X	X	A2	A1	A0	R/W
← บิตกำหนดแอดเดรสทงที่ →				← บิตกำหนดแอดเดรสโปรแกรมได้ →			

รูปที่ 2.7 รูปแบบของข้อมูลกำหนดแอดเดรสที่ใช้ในการอ้างอิงแบบ 7 บิต

ข้อมูลในไบต์ต่อมา คือ ข้อมูลที่ทำการถ่ายทอดจริง(data)

หลังจากที่มีการถ่ายทอดข้อมูล ไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.21 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I<sup>2</sup>C ของการอ้างอิงแบบ 7 บิต



รูปที่ 2.8 รูปแบบของข้อมูลอนุกรมที่ใช้ติดต่อกับอุปกรณ์บัส I<sup>2</sup>C เมื่อใช้การอ้างอิงแบบ 7 บิต

การต่ออุปกรณ์ระบบบัส I<sup>2</sup>C กับไมโครคอนโทรลเลอร์ MCS-51 สามารถทำได้ง่ายมาก เพียงใช้ขาพอร์ต 2 ขา โดยกำหนดให้ขาหนึ่งเป็น SDA อีกขาหนึ่งเป็น SCL และต่อตัวต้านทานค่าประมาณ 4.7 กิโลโห์ม พูลอัพที่ขาพอร์ตทั้ง 2 ขา

### 2.2.7 การเขียนโปรแกรมติดต่อบัส I<sup>2</sup>C

เริ่มต้นด้วยการสร้างสถานะมาตรฐานของบัส I<sup>2</sup>C อันประกอบด้วย สถานะเริ่มต้น, สถานะสิ้นสุดการส่งข้อมูล, สถานะหยุด, สัญญาณนาฬิกาบนขา SCL, การเขียนและอ่านข้อมูลกับอุปกรณ์บนระบบบัส I<sup>2</sup>C การสร้างสถานะเริ่มต้น

1. เมื่อต้องการติดต่อบัส I<sup>2</sup>C สิ่งแรกที่ต้องทำสำหรับไมโครคอนโทรลเลอร์ซึ่งถือว่าเป็นอุปกรณ์มาสเตอร์คือ การทำให้บัสว่างด้วยการกำหนดให้ขา SCL และขา SDA มีลอจิกเป็น “1” ทั้งคู่
2. จากนั้นทำให้ขา SDA มีลอจิก “0” โดยที่ขา SCL ยังคงเป็นลอจิก “1” อยู่

3. กำหนดให้ขา SCL มีลอจิกเป็น “0” ถึงตอนนี้ทั้ง SCL และ SDA จะมีลอจิกเป็น “0” ทั้งคู่ พร้อมทั้งจะติดต่อกันได้แล้ว

#### การสร้างสภาวะหยุด

1. เมื่อต้องการหยุดส่งข้อมูล ต้องส่งสภาวะหยุดออกไป โดยในตอนแรกต้องกำหนดให้ขา SCL และ SDA เป็นลอจิก “0” ทั้งคู่ก่อน

2. กำหนดให้ขา SCL มีลอจิกเป็น “1” โดย SDA ยังคงมีลอจิกเป็น “0”

3. ทำให้ขา SDA มีลอจิกเป็น “1” ทำให้กลับเข้าสู่สภาวะว่างอีกครั้ง พร้อมทั้งจะรับหรือส่งข้อมูลต่อไป การส่งข้อมูลลอจิก “0” และลอจิก “1”

หลังจากส่งบิตเริ่มต้นแล้ว ลำดับต่อไปคือ ส่งข้อมูลควบคุมซึ่งจะเป็นขบวนของลอจิก “0” และ “1” ดังนี้

1. ทำให้ขา SDA เป็น “0” สำหรับการส่งข้อมูลลอจิก “0”

2. ทำให้ขา SCL เป็น “1” สำหรับการป้อนสัญญาณนาฬิกา ในขณะที่ขา SDA ยังคงเป็น “0” อยู่

3. จากนั้นทำให้ขา SCL กลับมามีสถานะเป็นลอจิก “0” เหมือนเดิม

ในขณะที่การส่งข้อมูลลอจิก “1” มีขั้นตอนดังนี้

1. ทำให้ขา SDA เป็น “1” สำหรับการส่งข้อมูลลอจิก “1”

2. ทำให้ขา SCL เป็น “1” สำหรับการป้อนสัญญาณนาฬิกา ในขณะที่ขา SDA ยังคงเป็น “1” อยู่

3. จากนั้นทำให้ขา SCL กลับมามีสถานะเป็นลอจิก “0” เหมือนเดิม

ข้อมูลที่ใช้ในการส่งไปยังขา SDA นั้นจะกำหนดที่แอกคิวมูลเตอร์แล้วทำการส่งออกไปยังแฟลชทวดด้วยการใช้คำสั่งหมุนข้อมูล (RLC A) เพื่อถ่ายทอดต่อไปยังขา SDA ต่อไป

### 2.3 หน่วยความจำอีอีพรอมอนุกรม

หน่วยความจำอีอีพรอมอนุกรม (serial EEPROM) เป็นหน่วยความจำแบบนอน-โวลไทล์ (non-volatile) หมายความว่า สามารถเก็บรักษาข้อมูลอยู่ได้โดยไม่ต้องจ่ายไฟเลี้ยง สามารถเขียน-อ่าน-ลบได้ด้วยสัญญาณไฟฟ้า ใช้การเชื่อมต่อในลักษณะอนุกรมแบบระบบบัส I<sup>2</sup>C สามารถต่อพ่วงกันได้โดยการกำหนดแอดเดรสทางฮาร์ดแวร์ในการเชื่อมต่อกับไมโครคอนโทรลเลอร์ MCS-51 จะใช้ขาพอร์ตเพียง 2 ขาเช่นเดียวกับอุปกรณ์ที่ใช้การเชื่อมต่อแบบบัส I<sup>2</sup>C ตัวอื่นๆ

24LC512 หน่วยความจำอีอีพรอมอนุกรม ความจุ 64 กิโลไบต์

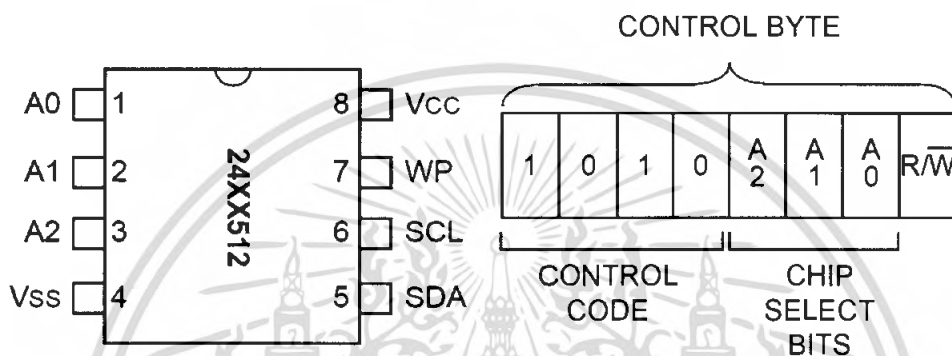
มีคุณสมบัติทางเทคนิคโดยสรุปดังนี้

- ขนาดของหน่วยความจำ 512 กิโลบิต หรือ 64 กิโลไบต์ สามารถเก็บข้อมูลได้นานมากกว่า 200 ปี
- แรงดันใช้งาน 2.5-5.5 โวลต์
- กระแสขณะสแตนด์บาย 100nA กระแสขณะอ่านข้อมูล 400μA และขณะเขียนข้อมูลสูง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความถี่ของสัญญาณนาฬิกา 400kHz
- ระยะเวลาในการเขียนสูงสุด 5 มิลลิวินาที สามารถเขียนทับใหม่ได้ 1 ล้านครั้ง
- มีฟังก์ชันป้องกันการเขียนข้อมูลทับ

ในรูปที่ 2.9 แสดงการจัดขา และข้อมูลกำหนดแอดเดรสของหน่วยความจำอีอีพรอมเบอร์ 24LC512 นี้ โดยใน 4 บิตบนของข้อมูลกำหนดแอดเดรส (Control Code) จะถูกกำหนดโดยผู้ผลิตไม่สามารถปรับเปลี่ยนได้ ส่วนบิตที่ 1-3 (Chip Select Bits) จะถูกกำหนดโดยผู้ใช้งานผ่านกระบวนการทางซอฟต์แวร์ หน้าที่การทำงานของขาต่างๆ ของ 24LC512 มีดังนี้



รูปที่ 2.9 การจัดขาและข้อมูลกำหนดแอดเดรสของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512

Vcc (ขาที่ 8) เป็นขาป้อนไฟเลี้ยงบวก

WP (Write protect : ขาที่ 7) ถ้าขานี้เป็นลอจิก “1” จะสามารถอ่านได้อย่างเดียว

SCL (Serial Clock : ขาที่ 6) เป็นขาอินพุตรับสัญญาณนาฬิกา

SDA (Serial data : ขาที่ 5) เป็นขาข้อมูลอนุกรม

Vss (ขาที่ 4) เป็นขาป้อนไฟเลี้ยงลบหรือกราวด์

A0-A2 (ขาที่ 1-3) เป็นขากำหนดแอดเดรสทางฮาร์ดแวร์ หากในระบบมี 24LC512 เพียงตัวเดียวให้ต่อลงกราวด์ทั้งหมด

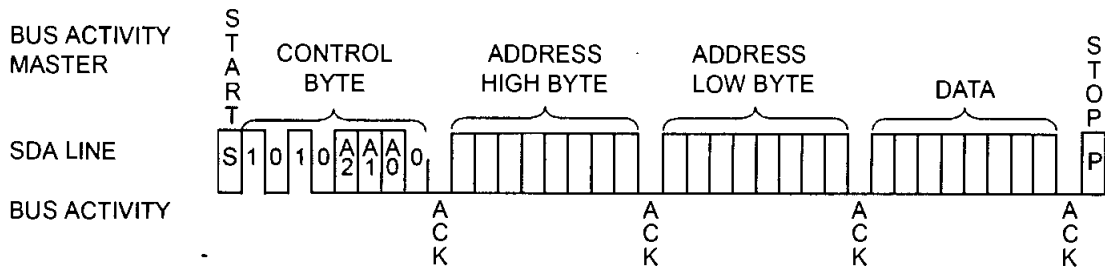
### 2.3.1 รูปแบบการอ่านและเขียนข้อมูลของ 24LC512

การเขียนข้อมูล

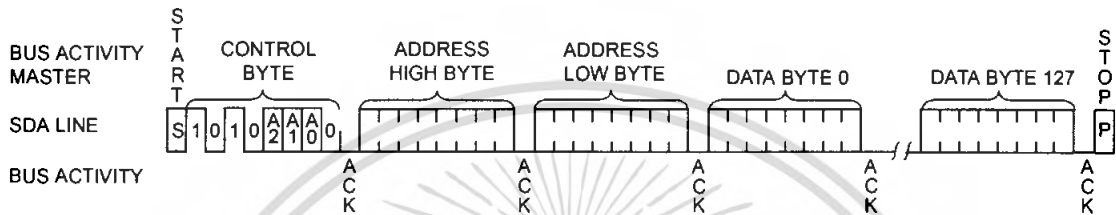
การเขียนข้อมูลของ 24LC512 มี 2 รูปแบบคือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 รูปแบบการเขียนข้อมูลแบบไบนารีของหน่วยความจำอีไอพรอมอนุกรมเบอร์ 24LC512



รูปที่ 2.11 รูปแบบการเขียนข้อมูลแบบเพชของหน่วยความจำอีไอพรอมอนุกรมเบอร์ 24LC512

#### การเขียนแบบไบนารี (Byte Write)

พิจารณารูปที่ 2.10 ไมโครคอนโทรลเลอร์ซึ่งเป็นอุปกรณ์มาสเตอร์จะทำหน้าที่ส่งข้อมูลและสัญญาณนาฬิกา ส่วน 24LC512 จะเป็นอุปกรณ์สเลฟทำหน้าที่รับข้อมูล การเขียนข้อมูลแบ่งออกเป็น 5 ช่วงด้วยกันดังนี้ ช่วงที่ 1 เป็นการเริ่มการติดต่อกับอุปกรณ์สเลฟ ในช่วงนี้ตัวมาสเตอร์จะสร้างสัญญาณเริ่มต้น (start) ขึ้น เพื่อให้อุปกรณ์สเลฟที่ต่ออยู่เตรียมพร้อมที่จะทำงาน

ช่วงที่ 2 อุปกรณ์มาสเตอร์ส่งข้อมูลควบคุมออกไปที่สาย SDA ซึ่งในข้อมูลควบคุมนี้ จะประกอบไปด้วยข้อมูลต่างๆ ดังในรูปที่ 2.10 จากนั้นรอรับสัญญาณรับรู้ (acknowledge : ACK) จากอุปกรณ์สเลฟ ซึ่งในที่นี้คือ 24LC512

ช่วงที่ 3 ไมโครคอนโทรลเลอร์ส่งแอดเดรสที่ต้องการเขียนไปให้ 24LC512 แล้วรอรับสัญญาณ ACK

ช่วงที่ 4 ไมโครคอนโทรลเลอร์ส่งข้อมูลที่ต้องการเขียนไปให้กับ 24LC512 แล้วรอรับสัญญาณ ACK ในช่วงเวลานี้ 24LC512 จะนำข้อมูลที่รับได้ไปเก็บไว้ในบัฟเฟอร์ภายในก่อน แต่ยังไม่ทำการเขียน

ช่วงที่ 5 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุด (stop) เพื่อหยุดการติดต่อกับ 24LC512

เมื่อผ่านช่วงที่ 5 ไปแล้ว 24LC512 จะเริ่มเขียนข้อมูลลงในหน่วยความจำจริงๆ ทั้งนี้ โดยนำข้อมูลที่เก็บไว้ในบัฟเฟอร์ภายในเขียนลงไปยังแอดเดรสที่กำหนดไว้ในช่วงที่ 3 จะใช้เวลาประมาณ 5 มิลลิวินาที ดังนั้นถ้าจะเขียนข้อมูลไบนารีถัดไป จะต้องหน่วงเวลาเพื่อให้การเขียนข้อมูลเสร็จสมบูรณ์ก่อน

การเขียนแบบเพจ (Page Write)

มี 5 ช่วงเช่นกัน ดังมีรายละเอียดต่อไปนี้

ช่วงที่ 1 ถึง 3 เหมือนกับการเขียนแบบไบนารีทุกประการ

ช่วงที่ 4 จะแตกต่างจากแบบแรกเล็กน้อยคือ สามารถส่งข้อมูลที่ต้องการเขียนในลักษณะต่อเนื่องกันได้ถึง 64 ข้อมูล ต่อการเขียน 1 เพจ โดยค่าของแอดเดรสจะเพิ่มขึ้นโดยอัตโนมัติ ในกรณีที่ส่งข้อมูลมากกว่า 64 ข้อมูล ข้อมูลตัวที่เกินจะถูกเขียนทับลงไปตำแหน่งแรกทันที กล่าวคือ ถ้าป้อนข้อมูล 66 ค่า ข้อมูลตัวที่ 65 จะเขียนทับข้อมูลตัวแรก และข้อมูลตัวที่ 66 จะเขียนทับข้อมูลตัวที่สอง ดังนั้นในการเขียนแบบเพจจะต้องระวังในเรื่องของความยาวข้อมูลด้วยเสมอ

ช่วงที่ 5 เช่นเดียวกับแบบแรกคือ ไมโครคอนโทรลเลอร์จะส่งสัญญาณหยุดเพื่อหยุดการติดต่อ หลังจากนั้นกระบวนการเขียนข้อมูลของ 24LC512 จึงเริ่มต้นขึ้น ซึ่งต้องหน่วงเวลาก่อนที่จะส่งข้อมูลชุดต่อไปเช่นกัน

### 2.3.2 การตรวจสอบสถานะการเขียนข้อมูล

นอกจากใช้การหน่วงเวลาเพื่อรอให้ 24LC512 เขียนข้อมูลเสร็จแล้ว ยังมีอีกวิธีการหนึ่งคือ ตรวจสอบบิต ACK เรียกว่า Acknowledge polling มีข้อดีคือ สามารถกำหนดช่วงเวลาที่น่านอนได้ โดยไม่เปลี่ยนแปลงตามความเร็วของตัวหน่วยความจำเอง มีโพล์ชาร์ตแสดงการทำงานดังรูปที่ 2.12

ขั้นตอนที่ 1 ในโพล์ชาร์ตคือช่วงที่ 1 ถึง 5 ของการเขียนข้อมูลตามปกติ

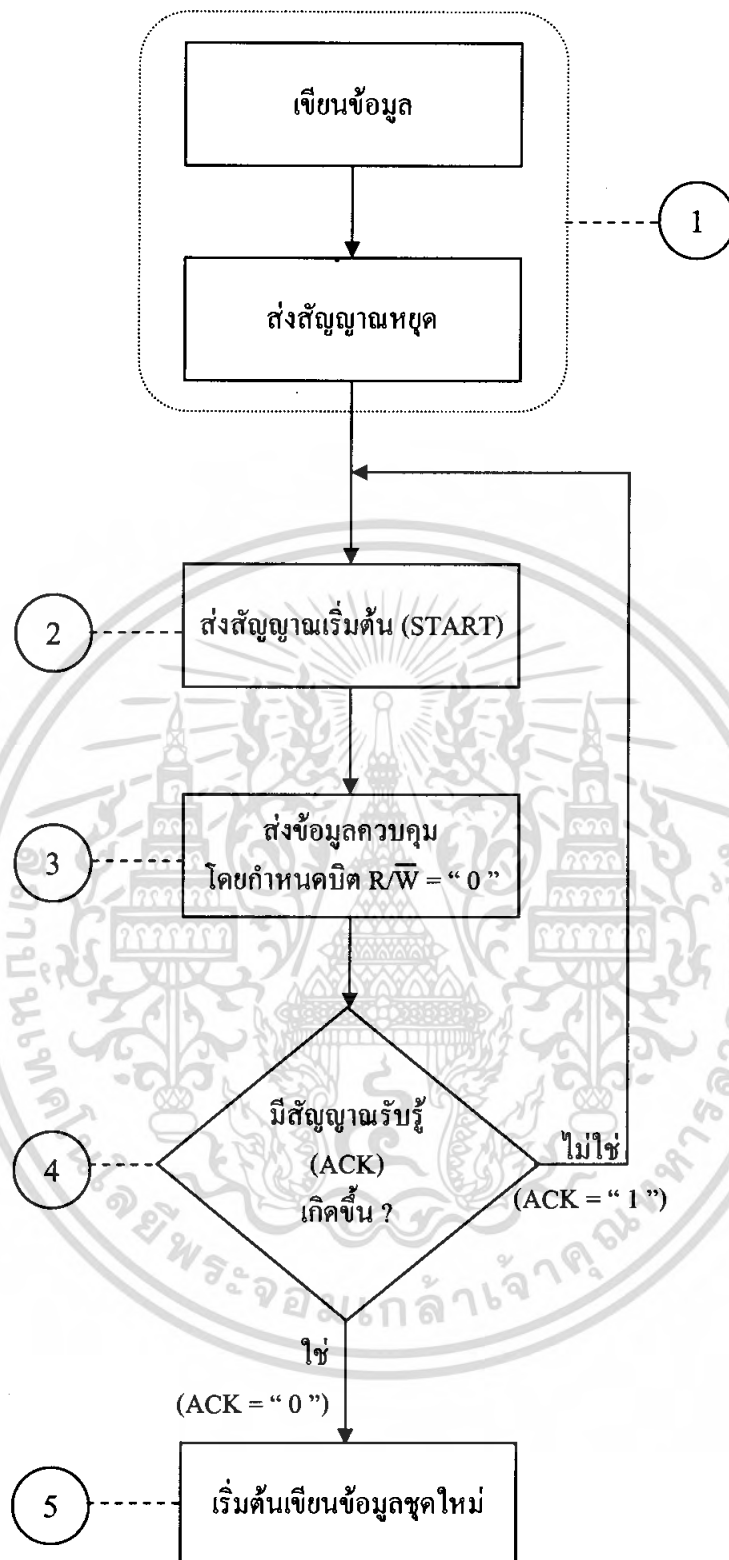
ขั้นตอนที่ 2 ไมโครคอนโทรลเลอร์จะส่งสัญญาณเริ่มต้น ไปให้กับ 24LC512

ขั้นตอนที่ 3 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุมไปให้ 24LC512 โดยกำหนดให้บิต

$R/W = 0$

ขั้นตอนที่ 4 เป็นขั้นตอนตรวจสอบว่าการเขียนข้อมูลของ 24LC512 เสร็จสิ้นหรือยัง โดยตรวจสอบสัญญาณรับหรือบิต ACK ที่ตัว 24LC512 จะส่งออกมา โดยถ้าบิต ACK เป็น “1” แสดงว่ายังเขียนข้อมูลไม่เสร็จ แต่ถ้าบิต ACK เป็น “0” แสดงว่าการเขียนข้อมูลเสร็จสิ้นแล้ว

ขั้นตอนที่ 5 เมื่อการเขียนข้อมูลภายในของ 24LC512 เสร็จสิ้นแล้วไมโครคอนโทรลเลอร์จะสามารถเขียนข้อมูลชุดถัดไปได้ทันที



รูปที่ 2.12 โฟลว์ชาร์ตแสดงวิธีการตรวจสอบบิต ACK

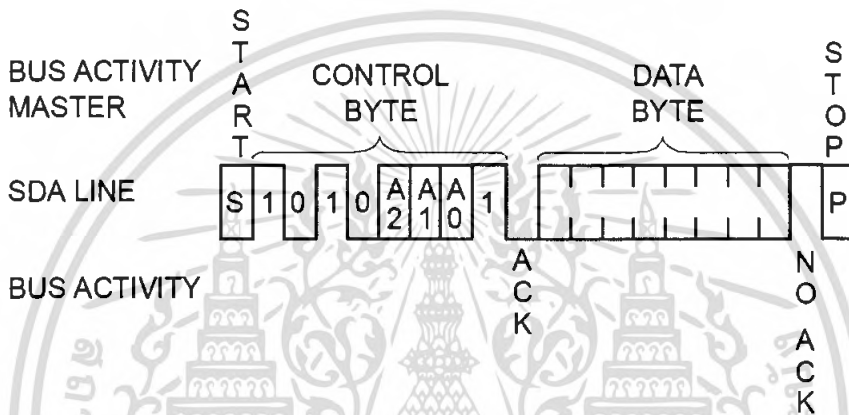
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การอ่านข้อมูล

ในการอ่านข้อมูลนี้ ไมโครคอนโทรลเลอร์หรืออุปกรณ์มาสเตอร์จะเป็นตัวรับข้อมูล ส่วน 24LC512 ซึ่งเป็นอุปกรณ์สเลฟจะเป็นตัวส่งข้อมูล ในการอ่านข้อมูลมีด้วยกัน 3 แบบคือ

แบบอ่านแอดเดรสปัจจุบัน (current address read)

การอ่านแบบนี้จะเป็นการอ่านข้อมูลจากแอดเดรสปัจจุบัน กล่าวคือ หลังจากเขียนหรืออ่านข้อมูลชุดล่าสุดเสร็จสิ้นแล้ว ตัวชี้แอดเดรสภายใน 24LC512 จะเพิ่มค่าขึ้น 1 ค่าโดยอัตโนมัติ ดังนั้นถ้าการอ่านหรือเขียนในครั้งที่แล้วกระทำที่แอดเดรส n การอ่านในลักษณะนี้จะอ่านข้อมูลที่แอดเดรส n+1 ในรูปที่ 2.13 แสดงรูปแบบของการอ่านข้อมูลแบบนี้



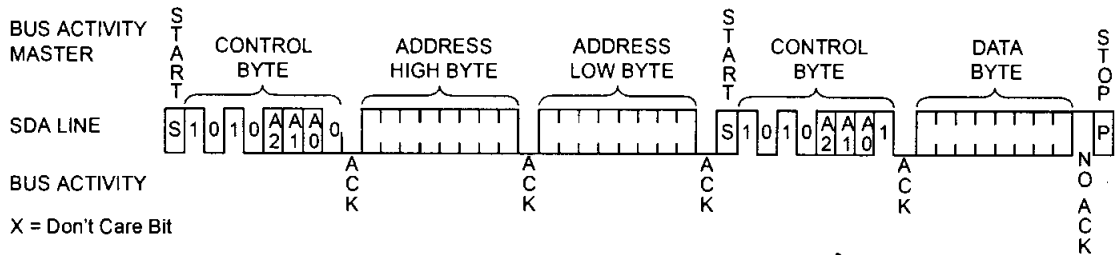
รูปที่ 2.13 รูปแบบการอ่านข้อมูลแบบแอดเดรสปัจจุบันของหน่วยความจำเบอร์ 24LC512

ช่วงการทำงานของการทำงานแบบอ่านแอดเดรสปัจจุบันมีดังนี้

- ช่วงที่ 1 ตัวมาสเตอร์ซึ่งในที่นี้คือ ไมโครคอนโทรลเลอร์จะส่งสัญญาณเริ่มต้นเพื่อติดต่อกับ 24LC512
- ช่วงที่ 2 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุม โดยที่บิต  $R/\overline{W} = "1"$  จากนั้นรอรับสัญญาณรับรู้หรือ ACK จาก 24LC512
- ช่วงที่ 3 24LC512 ส่งข้อมูลออกมา
- ช่วงที่ 4 ไมโครคอนโทรลเลอร์จะส่งสัญญาณ NACK ออกไป เพื่อให้ 24LC512 หยุดทำงานแล้วทำให้สาย SDA เป็น "1"
- ช่วงที่ 5 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุดเพื่อหยุดการติดต่อ

การอ่านแบบสุ่ม (random read)

เป็นการอ่านข้อมูลแบบเจาะจงแอดเดรส กล่าวคือ ถ้าต้องการอ่านข้อมูลที่ตำแหน่งใด ก็ให้ระบุแอดเดรสของ 24LC512 ที่ต้องการอ่าน มีรูปแบบแสดงในรูปที่ 2.14 ช่วงการทำงานมีดังนี้



รูปที่ 2.14 รูปแบบการอ่านข้อมูลแบบสุ่มหรือแบบเจาะจงแอดเดรสของ 24LC512

ช่วงที่ 1 ไมโครคอนโทรลเลอร์ส่งสัญญาณเริ่มต้นเพื่อติดต่อกับอุปกรณ์สเลฟ

ช่วงที่ 2 ไมโครคอนโทรลเลอร์ส่งข้อมูลควบคุมออกไป โดยต้องให้บิต  $R/\bar{W} = "0"$  จากนั้นรอรับสัญญาณ ACK จาก 24LC512

ช่วงที่ 3 ไมโครคอนโทรลเลอร์ส่งค่าแอดเดรส จากนั้นรอรับสัญญาณ ACK จาก 24LC512

ช่วงที่ 4 ไมโครคอนโทรลเลอร์ส่งสัญญาณเริ่มต้นอีกครั้งหนึ่ง เพื่อระงับการเขียนข้อมูล เนื่องจาก ในช่วงที่ 2 บิต  $R/\bar{W} = "0"$

ช่วงที่ 5 ไมโครคอนโทรลเลอร์ ส่งข้อมูลควบคุมออกไป โดยที่บิต  $R/\bar{W} = "1"$  จากนั้นรอรับสัญญาณ ACK จาก 24LC512

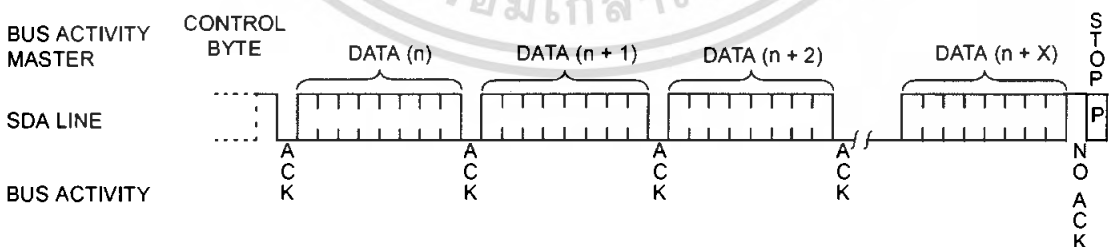
ช่วงที่ 6 24LC512 จะส่งข้อมูลในตำแหน่งที่ระบุไว้ มาให้กับไมโครคอนโทรลเลอร์

ช่วงที่ 7 ไมโครคอนโทรลเลอร์ส่งสัญญาณ NACK ออกไปเพื่อให้ 24LC512 หยุดทำงาน แล้วปล่อยให้สาย SDA เป็น "1"

ช่วงที่ 8 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุด

การอ่านแบบเรียงลำดับ (sequential read)

การอ่านแบบนี้จะเป็นการอ่านข้อมูลออกจาก 24LC512 แบบเรียงกันไปจากแอดเดรสเริ่มต้นจนถึงสุดท้ายแล้ววนใหม่ มีรูปแบบการอ่านแสดงในรูปที่ 2.15



รูปที่ 2.15 รูปแบบการอ่านข้อมูลแบบเรียงลำดับของ 24LC512

ช่วงการทำงานมีดังนี้

ช่วงที่ 1 มีการทำงานเหมือนกับการอ่านแบบสุ่มทุกประการ

ช่วงที่ 2 เป็นช่วงที่ 24LC512 ส่งข้อมูลมาให้กับไมโครคอนโทรลเลอร์เรียงลำดับกันไปโดยตัวชี้แอดเดรสภายในของ 24LC512 จะเพิ่มค่าขึ้นทีละหนึ่งค่า จากแอดเดรสที่กำหนดไว้ไปจนถึงแอดเดรสค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้าย จากนั้นก็จะกลับไปชี้ข้อมูลที่แอดเดรสเริ่มต้นใหม่ ดังนั้นการอ่านข้อมูลแบบนี้จะสามารถอ่านข้อมูลได้  
อย่างไม่มีวันจบ จนกว่าตัวมาสเตอร์จะสร้างสัญญาณ NACK

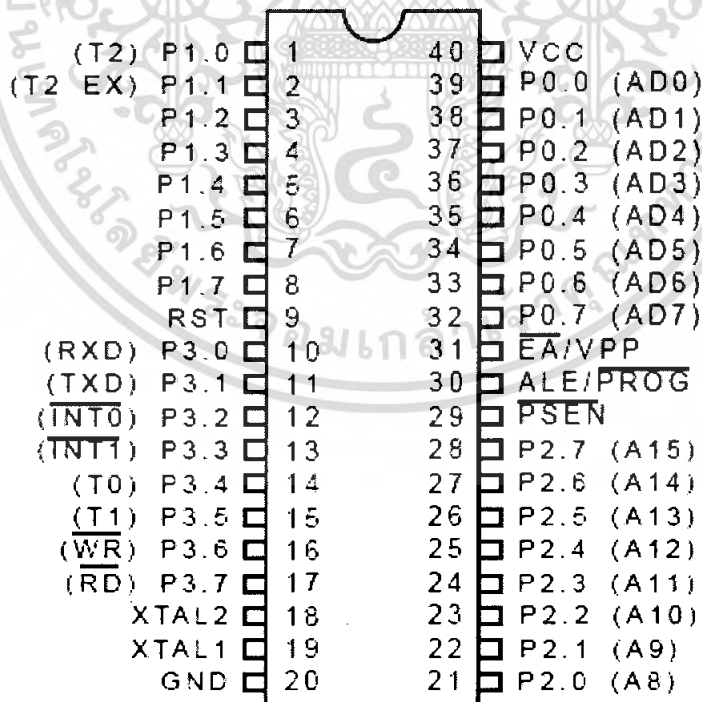
ช่วงที่ 3 ไมโครคอนโทรลเลอร์ส่งสัญญาณ NACK ออกไปเพื่อให้ 24LC512 หยุดทำงาน  
แล้วปล่อยให้สาย SDA เป็น “1”

ช่วงที่ 4 ไมโครคอนโทรลเลอร์สร้างสัญญาณหยุด

## 2.4 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์เป็นอุปกรณ์ที่นิยมนำมาใช้ในการควบคุมการทำงานของวงจรอิเล็กทรอนิกส์  
ต่างๆ เพราะภายในตัวไมโครคอนโทรลเลอร์มีทั้งหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก  
วงจรรับสัญญาณอินพุต วงจรจับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกา รวมไปถึง  
ทำให้สามารถนำไปใช้งานได้อย่างหลากหลาย และมีประสิทธิภาพ เพียงแค่เขียนโปรแกรมลงใน  
ไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะประมวลผลตามคำสั่งในโปรแกรมแล้วไปควบคุมการทำงาน  
ของอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ต่อไป

ไมโครคอนโทรลเลอร์ที่ใช้ในโครงงานนี้คือ ไมโครคอนโทรลเลอร์ของบริษัท ATMEL เบอร์  
AT89C52 ซึ่งใช้ซีพียูขนาด 8 บิต หน่วยความจำโปรแกรมเป็นแบบแฟลชขนาด 8 กิโลไบต์ สามารถลบและ  
เขียนใหม่ได้พันครั้ง คุณสมบัติทางเทคนิค โครงสร้างและสถาปัตยกรรมเหมือนกับไมโครคอนโทรลเลอร์  
ตระกูล MCS-51 โดยทั่วไป จึงขอกล่าวถึงเฉพาะส่วนที่สำคัญและเกี่ยวข้องกับโครงงานดังนี้



รูปที่ 2.16 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 2.4.1 การใช้งานพอร์ตต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุต ดังนั้นจึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

##### การใช้งานเป็นพอร์ตอินพุต

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล “1” มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อกับวงจรพูลอัพภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น “1” สามารถรับสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านก่อนเข้าไปเมื่อเป็นเช่นนี้ อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดการทำงานในสถานะลอจิก “0” จะดีและสะดวกที่สุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อกับไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก “0” แล้ว)

##### การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา กล่าวคือ เมื่อต้องการส่งข้อมูล “0” ไปยังวงจรแลตซ์ ซึ่งก็จะส่งผลต่อไปขับเฟต ทำให้เฟตทำงานที่ขาพอร์ตที่กำหนดให้ทำงานเกิดลอจิก “0” ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล “1” ออกไป ก็ให้เขียนข้อมูล “1” ที่ขาพอร์ตนั้น ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มี การอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่ในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

#### 2.4.2 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการเขียนโปรแกรมที่จะจัดการกับสัญญาณที่รับมาจากรีโมทคอนโทรลต้นแบบ จะต้องเข้าใจถึงจังหวะการทำงานของซีพียู เพื่อที่จะสามารถนำไปคำนวณเกี่ยวกับฐานเวลาของสัญญาณจากรีโมทได้ เพราะสัญญาณจากรีโมทคอนโทรลจะมีฐานเวลาของพัลส์ในระดับมิลลิวินาที

ในการทำงาน 1 รอบหรือ 1 แมกซ์ซินไซเคิลซีพียูในไมโครคอนโทรลเลอร์ MCS-51 จะใช้เวลา 12 คาบเวลาของสัญญาณนาฬิกา นั่นคือ เวลาในการทำงาน 1 ไซเคิลมีค่าเท่ากับ  $1 \mu\text{s}$  หรือมีความเร็วในการทำงานภายใน 1 MHz ในกรณีที่ใช้ความถี่สัญญาณนาฬิกา 12 MHz ดังนั้นถ้าต้องการทราบความเร็วของการทำงานภายในของไมโครคอนโทรลเลอร์ MCS-51 สามารถหาได้จาก ค่าความถี่สัญญาณนาฬิกาหารด้วย 12 และถ้าต้องการหาค่าเวลาของ 1 รอบการทำงานหรือ 1 แมกซ์ซินไซเคิล สามารถทำได้โดยการหาส่วนกลับของความเร็ว

ในการทำงานภายในของไมโครคอนโทรลเลอร์ MCS-51 สามารถสรุปเป็นสูตรทางคณิตศาสตร์ได้ดังนี้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์เท่ากับ  
 ความถี่ของสัญญาณนาฬิกา(ค่าของคริสตอลที่อยู่ที่ขา XTAL1 และ XTAL2) /12  
 เวลา 1 แมกซ์ซินไซเกิล = 1/ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์

### 2.4.3 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำภายในหลักๆอยู่ 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล สำหรับเบอร์ AT89C52 มีหน่วยความจำโปรแกรมขนาด 8 กิโลไบต์ ส่วนหน่วยความจำข้อมูลแบ่งเป็น 2 ส่วน คือหน่วยความจำสำหรับเก็บข้อมูลภายในชิพมี 256 ไบต์ และหน่วยความจำข้อมูลภายนอกชิพรองรับ 64 กิโลไบต์

FFH	หน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงแบบ โคยอ้อม	รีจิสเตอร์ฟังก์ชันพิเศษ ( SFR ) สามารถเข้าถึงแบบ โคยตรงได้
80H	เท่านั้น	
7FH	หน่วยความจำข้อมูลส่วนล่าง สามารถเข้าถึงได้ทั้งแบบโคยตรง	
00H	และโคยอ้อม	

รูปที่ 2.17 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์เบอร์ AT89C52

รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register : SFR) เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช สำหรับเบอร์ AT89C52 มีด้วยกัน 28 ตัว มีแอดเดรสอยู่ระหว่าง 80F-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบนสามารถเข้าถึงได้โดยตรง (direct addressing) รีจิสเตอร์ที่สำคัญในโครงงานนี้เช่น รีจิสเตอร์ไทมเมอร์ รีจิสเตอร์ TCON รีจิสเตอร์ TMOD แอกลิวมูลเตอร์ รีจิสเตอร์ชี้ข้อมูลหรือคาคาพอยน์เตอร์ รีจิสเตอร์พอร์ต เป็นต้น

### 2.4.4 ไทมเมอร์

ในการรับสัญญาณจากกรีโมทคอนโทรลดินแบบมาเก็บในหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ สำหรับโครงงานนี้ จะใช้การนับเวลาของไทมเมอร์เป็นหลัก(รายละเอียดอยู่ในบทที่ 3)

รีจิสเตอร์ไทมเมอร์ (Timer register)

เป็นรีจิสเตอร์ขนาด 16 บิต แต่จัดแบ่งเป็นไบต์สูงและไบต์ต่ำ เช่นเดียวกับรีจิสเตอร์ DPTR รีจิสเตอร์ไทมเมอร์ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (counter) ภายในไมโครคอนโทรลเลอร์เพื่อใช้ในการสร้างฐานเวลา, จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายในบางที่เรียกรีจิสเตอร์ตัวนี้ว่ารีจิสเตอร์ไทมเมอร์/เคาน์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช AT89C51 จะมีรีจิสเตอร์ไทมเมอร์/เคาน์เตอร์ 2 ตัว แบ่งเป็น T0 หรือ ไทมเมอร์ 0 และ T1 หรือ ไทมเมอร์ 1 ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทมเมอร์ไบต์ต่ำ (TL) และรีจิสเตอร์ไทมเมอร์ไบต์สูง (TH) เหมือนกัน โดยรีจิสเตอร์ TL0 มีแอดเดรสอยู่ที่ 8AH ในส่วนของ SFR รีจิสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH ตามลำดับ

รีจิสเตอร์แต่ละตัวมีขนาด 8 บิต แต่ในการใช้งานโดยทั่วไปมักใช้ร่วมกันโดยจัดเป็นคู่คือ TL0 กับ TH0 รวมกันเป็นรีจิสเตอร์ไทมเมอร์ 0 ขนาด 16 บิต และ TL1 กับ TH1 รวมกันเป็นรีจิสเตอร์ไทมเมอร์ 1 ขนาด 16 บิต โดยใน TL0 และ TL1 เก็บข้อมูล 8 บิตล่าง ส่วน TH0 และ TH1 เก็บข้อมูล 8 บิตบน รีจิสเตอร์ไทมเมอร์ทั้ง 2 คู่เมื่อนำมาใช้ร่วมกันจะสามารถเก็บค่าของการนับได้สูงสุด 65,536 หรือ FFFFH เมื่อนับถึงค่านี้แล้วจะวนไปเริ่มนับ 0000H ใหม่ และเมื่อเกิดการนับรอบใหม่ บิต TF0 หรือ TF1 ในรีจิสเตอร์ TCON ที่ใช้ควบคุมการทำงานของไทมเมอร์จะเกิดการเซต เพื่อแจ้งให้ทราบว่า นับเกินค่าสูงสุดแล้ว การเซตบิต TF0 หรือ TF1 ขึ้นอยู่กับว่าเลือกใช้งานรีจิสเตอร์ไทมเมอร์ตัวใด

สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89SXX จะมีรีจิสเตอร์ไทมเมอร์/เคาน์เตอร์ถึง 3 ตัว โดยมีรีจิสเตอร์ TL2 และ TH2 ซึ่งมีแอดเดรส อยู่ที่ CCH และ CDH ตามลำดับเพิ่มเติมเข้ามา

รีจิสเตอร์เลือกโหมดการทำงานหรือ TMOD

รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ตำแหน่งที่ 89H ไม่สามารถอ้างตำแหน่งแบบบิตได้ รายละเอียดของการกำหนดค่าบิตต่าง ๆ จะเป็นดังนี้

ตารางที่ 2.2 บิตภายใน TMOD

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0



Gate: ใช้เลือกลักษณะการควบคุมการทำงานของไทมเมอร์/เคาน์เตอร์ ถ้าเป็น “0” ไทมเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TR0 หรือ TR1 ในรีจิสเตอร์ TCON เป็น “1” เรียกว่า การควบคุมทางซอฟต์แวร์

ถ้าเป็น “1” ไทมเมอร์/เคาน์เตอร์จะทำงานเมื่อบิต TR0 หรือ TR1 ในรีจิสเตอร์ TCON เป็น “1” และสถานะลอจิกที่ขาอินพุตอินเตอร์รัปต์ INT0 และ INT1 เป็น “1” เรียกว่า การควบคุมทางฮาร์ดแวร์

C/T: ใช้เลือกลักษณะการทำงานของไทมเมอร์/เคาน์เตอร์ ถ้าเป็น “0” เลือกให้ทำงานเป็นไทมเมอร์ โดยใช้อินพุตจากสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์

ถ้าเป็น “1” เลือกให้ทำงานเป็นเคาน์เตอร์ โดยรับสัญญาณอินพุตทางขา T0 หรือ T1

M1,M0 : ใช้เลือกโหมดการทำงานของไทมเมอร์/เคาน์เตอร์

“00” เลือกให้ทำงานในโหมดไทมเมอร์/เคาน์เตอร์ 13 บิต

“01” เลือกให้ทำงานในโหมดไทมเมอร์/เคาน์เตอร์ 16 บิต

“10” เลือกให้ทำงานในโหมดไทมเมอร์/เคาน์เตอร์ขนาด 8 บิต แบบตั้งค่าอัตโนมัติ

“11” สำหรับไทมเมอร์ 0 เลือกให้ทำงานในโหมดไทมเมอร์/เคาน์เตอร์แยกส่วน โดยแยกออกเป็นไทมเมอร์/เคาน์เตอร์ 8 บิต 2 ตัว รีจิสเตอร์ TLO จะได้รับการควบคุมการเปิดปิดจากบิต TR0 ในรีจิสเตอร์ TCON และ รีจิสเตอร์ TH0 ซึ่งเป็นไทมเมอร์/เคาน์เตอร์ 8 บิต อีกตัวหนึ่ง จะได้รับการควบคุมจากบิต TR1 ในรีจิสเตอร์ TCON ในกรณีของไทมเมอร์ 1 เป็นการสั่งให้ไทมเมอร์/เคาน์เตอร์ 1 หยุดทำงาน (คิสเอเบิล)

รีจิสเตอร์ควบคุมการทำงานหรือ TCON

เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ตำแหน่งที่ 88H สามารถอ้างตำแหน่งแบบบิตได้ รายละเอียดของการกำหนดค่าบิตต่าง ๆ จะเป็นดังนี้

ตารางที่ 2.3 บิตภายใน TCON

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1,TF0 : เซตด้วยกระบวนการทางฮาร์ดแวร์ เมื่อค่าของรีจิสเตอร์ไทมเมอร์ 1 หรือไทมเมอร์ 0 เกิดการนับเกินหรือเกิดโอเวอร์โฟลว การเคลียร์บิตนี้ทำได้ด้วยกระบวนการทางฮาร์ดแวร์เช่นกัน โดยบิตนี้จะเคลียร์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

TR1,TR0 : ใช้ในการเปิดปิดการทำงานของไทมเมอร์ 1 หรือไทมเมอร์ 0 (เอ็นเอเบิลหรือคิสเอเบิล) ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้ไทมเมอร์ใดทำงานต้องเซตบิตนั้นให้เป็น “1”

IE1,IE0 : บิตนี้จะใช้ในกระบวนการอินเตอร์รัปต์ สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อสามารถตรวจจับขอบขาของสัญญาณอินเตอร์รัปต์จากภายนอกที่ขาอินพุตอินเตอร์รัปต์ 1(INT1) หรือ 0(INT0) ได้ และจะทำการเคลียร์เมื่อมีการบริการอินเตอร์รัปต์เกิดขึ้น

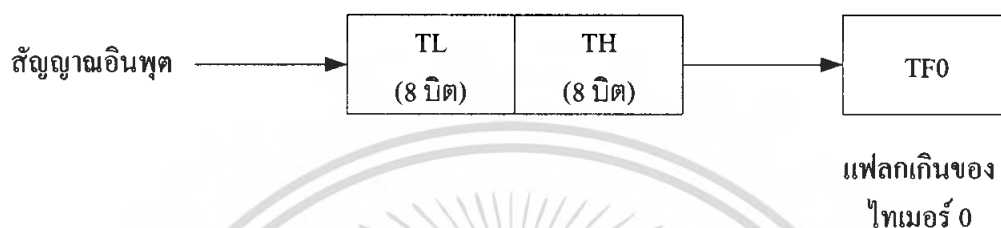
IT1,IT0 : บิตนี้จะใช้ในกระบวนการอินเตอร์รัปต์ โดยใช้ในการเลือกลักษณะของสัญญาณอินเตอร์รัปต์จากภายนอกที่ต้องการให้ทำการตอบสนองสำหรับขาอินพุตอินเตอร์รัปต์ 1(INT1) หรือ 0(INT0) การเซตและเคลียร์ทำได้ด้วยกระบวนการทางซอฟต์แวร์ โดยถ้าเป็น “0” เลือกขอบขาของสัญญาณ ถ้าเป็น “1” เลือกระดับลอจิกต่ำ

การใช้งานไทมเมอร์ในไมโครคอนโทรลเลอร์ MCS-51

เมื่อกำหนดให้ทำงานเป็นตัวตั้งเวลาหรือไทมเมอร์ ค่าของรีจิสเตอร์จะเพิ่มขึ้นในทุกๆ แมกซีนไซเคิล ดังนั้นเมื่อทำงานเป็นไทมเมอร์ รีจิสเตอร์จะทำการนับค่าของแมกซีนไซเคิลนั่นเอง และเนื่องจากแมกซีนไซเคิลประกอบด้วยคาบเวลาของวงจรกำเนิดสัญญาณนาฬิกา 12 คาบเวลา ดังนั้นอัตราในการนับของรีจิสเตอร์จึงเท่ากับ 1/12 ของความถี่สัญญาณนาฬิกา

## โหมดการทำงานของไทเมอร์

จะขอล่าวดึงเฉพาะที่ใช้ในโครงการเท่านั้น คือใช้ไทเมอร์ 0 ทำงานในโหมด 1 คือเป็นไทเมอร์ 16 บิต ควบคุมจากภายใน โดยกระบวนการทางซอฟต์แวร์ จะต้องกำหนดให้รีจิสเตอร์ TMOD เป็น 01H มีไคอะแกรมการทำงานแสดงในรูปที่ 2.18 โดยรีจิสเตอร์ TLO และ TH0 จะใช้งานร่วมกัน ค่าของการนับจะมีขนาด 16 บิต คือ 0000H-FFFFH เมื่อนับครบรอบ ค่าของการนับเปลี่ยนจาก FFFFH เป็น 0000H ก็จะเซตบิต TF0 ในรีจิสเตอร์ TCON



รูปที่ 2.18 ไคอะแกรมการทำงานในโหมด 1 ของไทเมอร์ 0

## 2.5 แอลซีดี (LCD:Liquid Crystal Displays)

รายละเอียดเกี่ยวกับ โมดูลแอลซีดี

ในโมดูลแอลซีดีจะมีส่วนประกอบหลักๆ 3 ส่วน ดังนี้

ตัวแสดงผล(display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอแอลซีดี

ตัวควบคุม(controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูลแอลซีดี เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุมแอลซีดีแบบอักษร ส่วน HD61830 ใช้ควบคุมแอลซีดีแบบกราฟฟิก

ตัวขับ(driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100 และ MSM5259 เป็นต้น

### 2.5.1 โครงสร้างภายในของตัวควบคุมโมดูลแอลซีดี

ในการใช้งานโมดูลแอลซีดีจำเป็นต้องทำความเข้าใจเกี่ยวกับ โครงสร้างและคำสั่งที่ใช้ในการควบคุม ให้ดีเสียก่อน ในที่นี้ขอยกตัวอย่าง โมดูลแอลซีดีแบบอักษร ซึ่งเป็นแอลซีดีที่ใช้ในโครงการเพราะสามารถเข้าใจง่าย ชิปควบคุมแอลซีดีเบอร์ HD44780 ซึ่งใช้ใน โมดูลแอลซีดีแบบอักษร ประกอบด้วย

บัฟเฟอร์อินพุตเอาต์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register: DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมตัวอักษร

- แรมเก็บข้อมูลแสดงผล (Display Data RAM: DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง ของตัวอักษรที่เก็บไว้ในหน่วยความจำรวมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รวมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรวมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงผลที่ตัวแสดงผลได้ มีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นมาใหม่ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิตการเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน CGRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

แฟลค (BUSY) เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลค BUSY นี้เสียก่อน

### 2.5.2 โมดูลแอลซีดีขนาด 16 ตัวอักษร 2 บรรทัด (แอลซีดี 16x2)

โมดูลแอลซีดีขนาด 16 x 2 มีขาต่อใช้งานทั้งสิ้น 14 ขา สำหรับรายละเอียดของการทำงานของแต่ละขามีดังนี้

V<sub>SS</sub> (ขา 1) : ต่อกาวด์

V<sub>DD</sub> (ขา 2) : ต่อไฟเลี้ยง +5 โวลต์

V<sub>0</sub> (ขา 3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มการแสดงผล

RS (ขา 4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น "0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขาเป็น "1" ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

$\overline{R/W}$  (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับแอลซีดี ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E (ขา 6) : เป็นขาอินพุตเปิดแอลซีดีให้ทำงาน

D0-D7 (ขา 7-14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่างแอลซีดีกับอุปกรณ์ภายนอกขนาด 8 บิต

### 2.5.3 คำสั่งควบคุมโมดูลแอลซีดี

ในการเขียนคำสั่งลงในตัวควบคุม แน่แน่นอนว่าต้องกำหนดให้ขา RS และ  $\overline{R/W}$  เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูลแอลซีดีของชิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (clear display) มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง (space) เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิกิวท์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น “1”

2. คำสั่ง return home ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลงนั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะ เป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

มีรายละเอียดของข้อมูลคำสั่งดังนี้

ตารางที่ 2.4 คำสั่งเลือกโหมดการป้อนข้อมูลของแอลซีดี

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

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

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น “0” แอดเดรสจะลดลง

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H – 07H (4ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดของข้อมูลรูปแบบคำสั่งดังนี้

ตารางที่ 2.5 คำสั่งควบคุมการแสดงผลของแอลซีดี

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต C ใช้ควบคุมการแสดงผลตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้บิตนี้เป็น “0” จะเป็นการปิดเคอร์เซอร์หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกะพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกะพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H -0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์และสั่งให้เคอร์เซอร์กะพริบ

5.คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.6 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษรของแอลซีดี

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับกำหนบบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

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

6.คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.7 คำสั่งกำหนดฟังก์ชันการทำงานของแอลซีดี

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

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น “0” จะแสดงผล 1 บรรทัด ถ้าเป็น “1” จะแสดงผล 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า

2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น “1” การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น “0” จะเป็นการแสดงผลแบบ 5x7 จุด และถ้าเป็น “1” จะเป็นการแสดงผลแบบ 5x10 จุด

ข้อมูลสำคัญที่ใช้บ่อยคือ 38H เป็นการกำหนดให้ไมโครแอลซีดีทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5x7 จุด

7.คำสั่งเลือกแอดเดรสของ CGRAM เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “0” บิต 6 เป็น “1” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

8.คำสั่งเลือกแอดเดรสของ DDRAM ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น “1” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสวิ่งขึ้นกับการกำหนดสถานะที่บิต N เป็น “1” แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

9.คำสั่งอ่านแฟล็ก BUSY และแอดเดรส  
มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

ตารางที่ 2.8 คำสั่งอ่านแฟล็ก BUSY และแอดเดรสของแอลซีดี

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านแฟล็ก BUSY (BF) โดยแฟล็กนี้จะเป็นตัวบอกสถานะของตัวควบคุมแอลซีดี ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุมแอลซีดีพร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่า ขณะนี้ตัวควบคุมแอลซีดียังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟล็กต้องกำหนดให้ขา  $R/\bar{W}$  เป็น “1” ด้วย แต่สัญญาณที่ RS ยังเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0-บิต 6 เป็นค่าข้อมูลของ แอดเดรสที่ต้องการอ่าน

### 2.5.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูลแอสซีดี

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูลแอสซีดีแสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมคการทำงานให้แก่โมดูลแอสซีดีก่อน จากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูลแอสซีดีมี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “0” หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูลแอสซีดีขณะนั้นเป็นคำสั่งในทางตรงข้าม หากขา RS ได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอสซีดีที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอสซีดี จากนั้นกำหนดให้ขา RS เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายในโมดูลแอสซีดีทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา  $R/\overline{W}$  เป็น “1” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอสซีดีของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอสซีดีและลอจิกเป็นลอจิก “1” ให้ขา RS แล้ว ต้องกำหนดให้ขา  $R/\overline{W}$  เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอกลงใน DDRAM ต่อไป

### 2.5.5 จังหวะการทำงานของแอสซีดีโมดูล

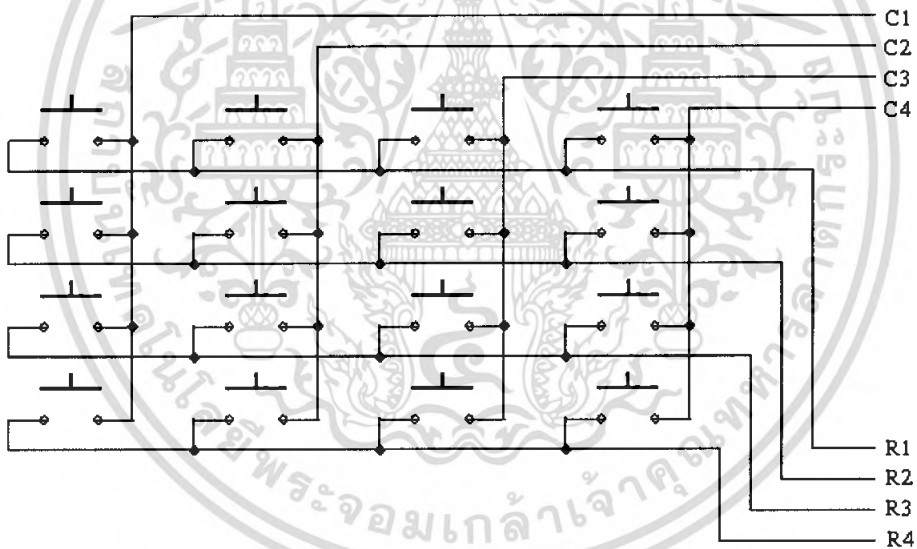
ในการติดต่อกับ โมดูลแอสซีดีจะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้คอนโทรลเลอร์ภายในแอสซีดีโมดูล แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้น ในการใช้งานโมดูลแอสซีดี ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูลแอสซีดีพร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูลแอสซีดี ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้โมดูลแอสซีดีทำการเตรียมความพร้อมหรืออินิเชียล (initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของโมดูลแอสซีดีแล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาทีเพื่อให้คอนโทรลเลอร์ในแอสซีดีโมดูลแปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมค 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิล โมดูลแอสซีดีให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าไปที่ขา E ของโมดูลแอสซีดีต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาทีทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของโมดูลแอสซีดี จะเห็นได้ว่ามีโปรแกรมย่อยคือ โปรแกรมอินิเชียลแอสซีดี, โปรแกรมหน่วงเวลา และ โปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิลโมดูลแอสซีดี

## 2.6 สวิตช์เมตริกซ์ 4 x 3 จุด หรือคีย์แพด

การอ่านค่าหรือรับค่าการกดสวิตช์ก็เป็นอีกลักษณะหนึ่งของการใช้งานของตัวไมโครคอนโทรลเลอร์ ซึ่งต้องสามารถรองรับและเชื่อมต่อใช้งานร่วมได้กับวงจรของสวิตช์ มี 2 ลักษณะใหญ่ๆ คือ ต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง เมื่อสวิตช์ตัวใดต่อวงจรจะสามารถอ่านค่าได้โดยตรง วงจรในลักษณะนี้ไม่มีความซับซ้อนมาก สามารถอ่านค่าของสวิตช์ได้ง่ายและรวดเร็ว แต่มีข้อเสียคือ ถ้าจำนวนของสวิตช์มีมากมาย จำนวนของสายข้อมูลก็จะมีมากตามไปด้วย ทำให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง

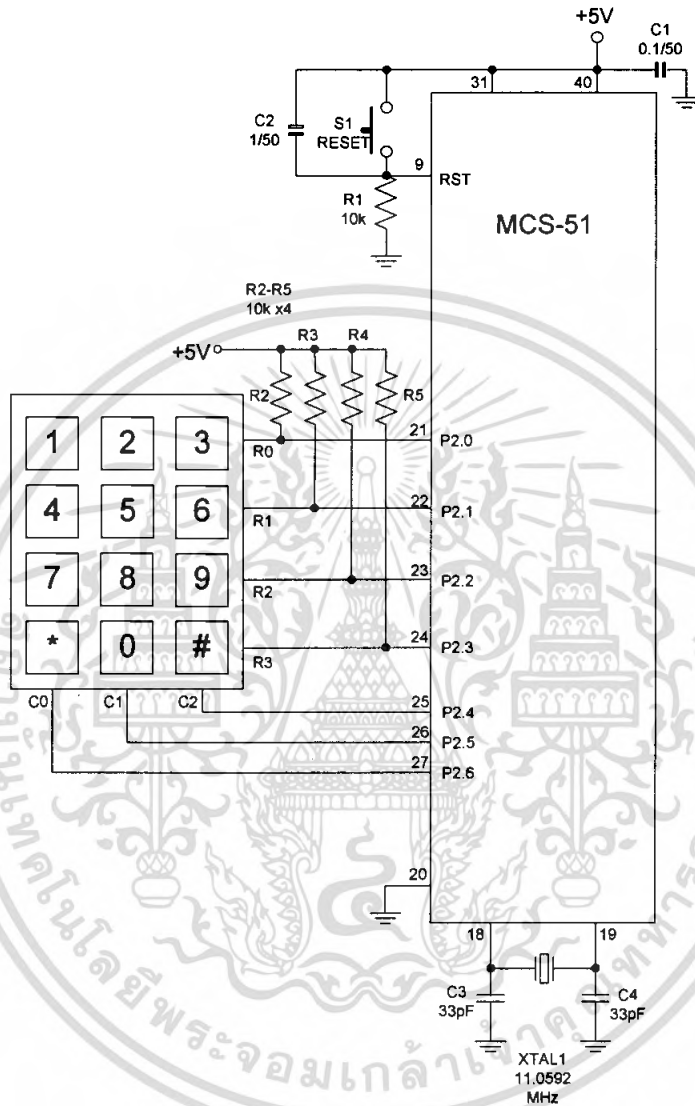
วงจรของสวิตช์อีกลักษณะหนึ่งคือ การต่อวงจรแบบเมตริกซ์ (matrix switch) ดังในรูปที่ 2.19 สวิตช์จะถูกต่อกันในแนวแกนตั้งและแนวนอน จะเรียกแนวตั้งว่า หลักหรือคอลัมน์ (column) ในขณะที่แนวนอนจะเรียกว่า แถวหรือโรว์ (row) ดังนั้นค่าของสวิตช์จะประกอบด้วยตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำได้มาซึ่งค่าของสวิตช์มีขั้นตอนซับซ้อนพอสมควร แต่วงจรของสวิตช์แบบนี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวก เพียงเพิ่มจำนวนสวิตช์ และแก้ไขซอฟต์แวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบเมตริกซ์เป็นที่นิยมใช้มากในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติที่จำนวนสวิตช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิตช์แบบเมตริกซ์นี้ว่า คีย์แพด (keypad)



รูปที่ 2.19 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด

การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51 จากตัวอย่างวงจรในรูปที่ 2.20 จะใช้พอร์ต 2 ไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้นเป็น สายคอลัมน์ 3 เส้นคือ C0-C2 และสายทางโรว์หรือแถวอีก 4 เส้นคือ R0-R3 โดยเฉพาะที่ขาพอร์ต P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัพไว้เพื่อกำหนดสถานะเริ่มต้นที่ไม่มีการกดคีย์ โดยที่ไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล “0” ไปยัง P2.6, P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพด ไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P2.0-P2.3 เข้ามาด้วย หากไม่มีการกดคีย์ของ P2.0-P2.3 ก็จะเป็น “1” ทั้งหมด ถ้าหากมีการกดคีย์ ค่าของ P2.0-P2.3 จะไม่เป็น 1111 อีกต่อไปเป็นการแจ้งให้ทราบว่ามีการกดคีย์แพดขึ้น จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งสิ่งที่จะได้มาอย่างแรกคือ ค่าตำแหน่งของคีย์นั้น จากนั้นก็ค่าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะนำค่าตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อจะได้หมายเลขของคีย์ที่กดอย่างแท้จริง ยกตัวอย่างจากวงจรในรูปที่ 2.20 หากคีย์ 0 ถูกกด ไมโครคอนโทรลเลอร์จะได้ค่าตำแหน่งของคีย์ 0 เป็น 0BH จากนั้นนำค่า 0BH นี้ไปเปิดตารางก็จะได้ข้อมูลเป็น 0 หมายถึงคีย์ 0 ซึ่งกระบวนการหลังจากนี้จะเป็นหน้าที่ของซอฟต์แวร์



รูปที่ 2.20 การเชื่อมต่อคีย์แปดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

## 2.7 โมดูลรับสัญญาณอินฟราเรดจากกริโมทคอนโทรล

TSOP34838 เป็นตัวรับสัญญาณขนาดเล็ก สำหรับรับสัญญาณจากกริโมทคอนโทรลอินฟราเรด สัญญาณเอาต์พุตที่ถูกคิมอดูเลตแล้วสามารถนำไปใช้งานกับไมโครคอนโทรลเลอร์ได้โดยตรง TSOP34838 สามารถรับสัญญาณจากกริโมทคอนโทรลอินฟราเรดมาตรฐานที่มีความถี่ของคลื่นพาห์ใกล้เคียง 38 kHz ได้ โดยหากความถี่ต่างจาก 38 kHz มากขึ้น ความไวในการรับสัญญาณก็จะยิ่งน้อยลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### คุณสมบัติที่สำคัญ

- มีโฟลต์ดีเทคเตอร์ และวงจรพรีแอมพลิฟายอยู่ในตัว
- มีฟิลเตอร์ภายใน
- เพิ่มการป้องกันการรบกวนจากสนามไฟฟ้า และสัญญาณคอนทินิวอัสที่มีความถี่ใกล้เคียง 38 kHz
- เพิ่มการป้องกันแสงอื่นๆ ที่อยู่โดยรอบเช่น แสงจากดวงอาทิตย์ แสงไฟจากหลอดฟลูออเรสเซนต์ เป็นต้น

รูปที่ 2.21 โมดูลรับสัญญาณอินฟราเรดจากกรีโมทคอนโทรล TSOP34838

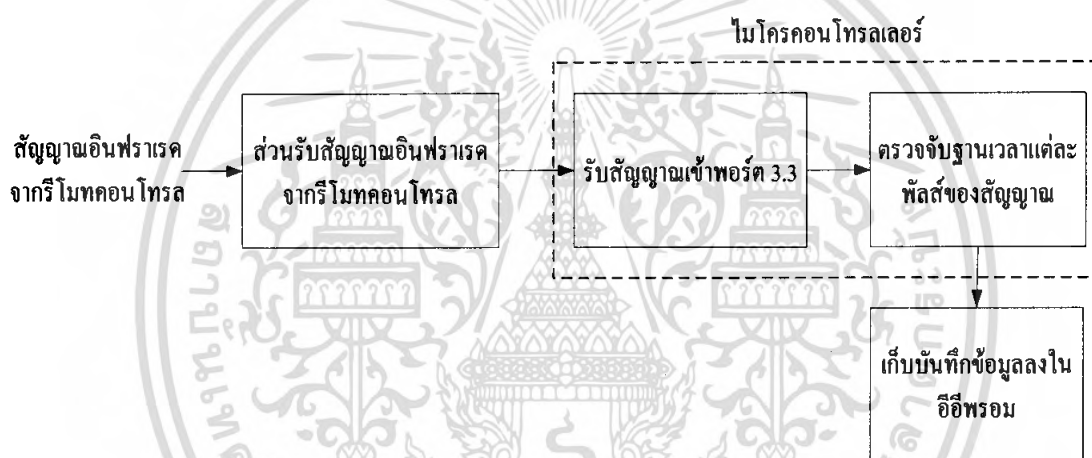


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

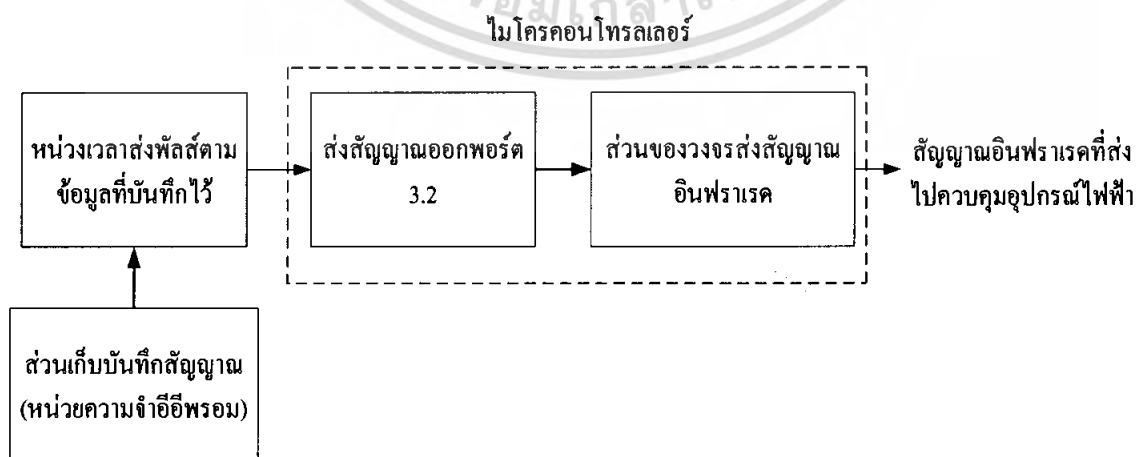
### บทที่ 3

#### การคำนวณและการสร้าง

ในโครงการนี้แบ่งการคำนวณและการสร้างเป็น 2 ส่วน คือ การออกแบบวงจรทางด้านฮาร์ดแวร์ และการเขียนซอฟต์แวร์ควบคุมวงจร โดยวงจรทางด้านฮาร์ดแวร์นี้จะใช้ไมโครคอนโทรลเลอร์ เป็นหัวใจหลักในการประมวลผลสัญญาณอินฟราเรดรีโมทคอนโทรล และควบคุมส่วนของการแสดงผล โดยเลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 มีหน่วยความจำ 8 kbyte และมีพอร์ตใช้งานเพียงพอกับโครงการนี้ ทางด้านซอฟต์แวร์ควบคุมวงจรใช้ภาษาแอสเซมบลีของ MCS-51 ในการเขียนโปรแกรมเพื่อกำหนดการรับสัญญาณอินฟราเรด บันทึกสัญญาณไว้ในหน่วยความจำอีอีพรอมเพื่อให้สามารถเก็บสัญญาณไว้ได้ แม้ว่าหยุดจ่ายไฟให้กับชิ้นงานแล้ว และนำสัญญาณที่เก็บไว้ส่งออกไปทางวงจรส่งสัญญาณเพื่อใช้ควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 3.1 บล็อกไดอะแกรมของส่วนรับและบันทึกสัญญาณอินฟราเรด



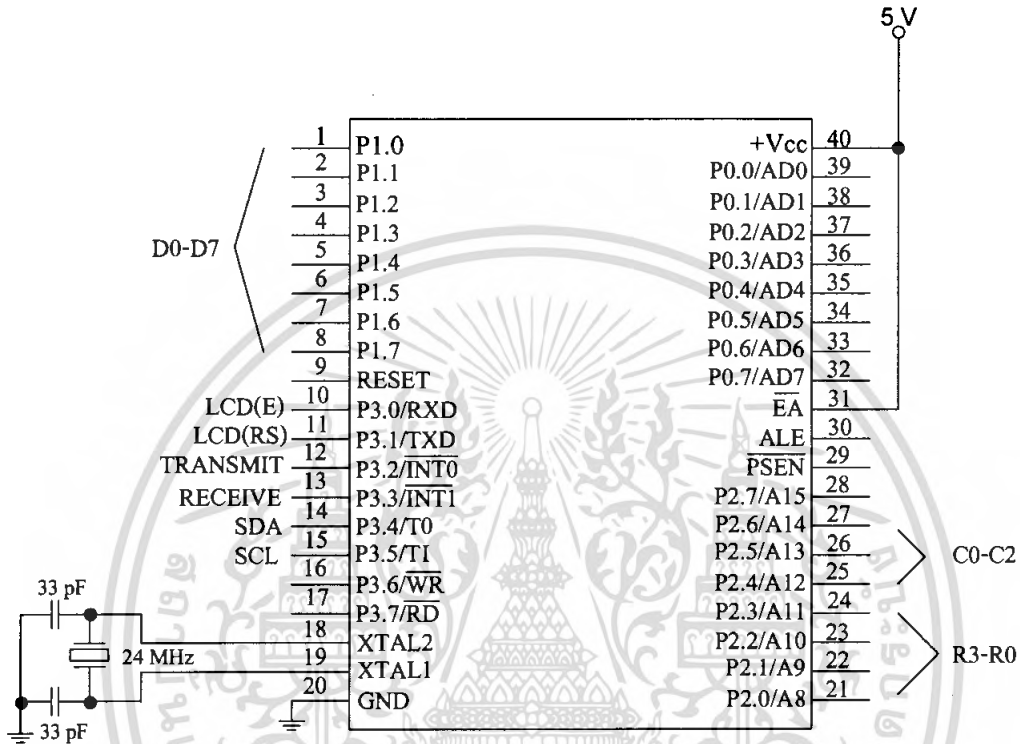
รูปที่ 3.2 บล็อกไดอะแกรมของส่วนส่งสัญญาณอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนสิทธิ์เพื่อการค้าเท่านั้น เมื่อผู้ดูแลเห็นนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 การออกแบบวงจรทางฮาร์ดแวร์

#### 3.1.1 ส่วนประมวลผล

การออกแบบส่วนประมวลผลในโครงการนี้ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 ของบริษัท ATMEL ซึ่งมีพอร์ตใช้งานและมีหน่วยความจำที่เพียงพอสำหรับโครงการนี้ สามารถโปรแกรมซ้ำได้ 1000 ครั้ง และมีการต่อขาใช้งานต่าง ๆ ดังต่อไปนี้-



รูปที่ 3.3 การต่อใช้งานไมโครคอนโทรลเลอร์

ขา 1 ถึง ขา 8 หรือพอร์ต 1.0 ถึง พอร์ต 1.7 ต่อกับขา D0 ถึง D7 เพื่อรับส่งข้อมูลและคำสั่งกับโมดูลแอลซีดีเพื่อใช้ในการส่งสัญญาณควบคุมการแสดงผล

ขา 9 เป็นขารีเซ็ต (Reset) ของไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์ เริ่มต้นการทำงานของโปรแกรมใหม่ทั้งหมด

ขา 10 หรือพอร์ต 3.0 ต่อกับขาอีนาเบิล (Enable) ของโมดูลแอลซีดีเพื่อใช้ในการส่งสัญญาณพัลส์ไปสั่งให้โมดูลแอลซีดีทำงาน

ขา 11 หรือพอร์ต 3.1 ต่อกับขา RS (Register Select) ของ โมดูลแอลซีดีเพื่อแยกชนิดของสัญญาณที่ขา D0-D7 ของโมดูลแอลซีดี ว่าเป็นคำสั่งหรือข้อมูลแสดงผล

ขา 12 หรือพอร์ต 3.2 ต่อกับวงจรส่งสัญญาณอินฟราเรดสำหรับส่งสัญญาณเพื่อใช้ควบคุมอุปกรณ์ไฟฟ้า

ขา 13 หรือพอร์ต 3.3 ทำหน้าที่รับสัญญาณอินฟราเรดที่ส่งมาจากโมดูลรับสัญญาณอินฟราเรด TSOP34838 เพื่อให้ไมโครคอนโทรลเลอร์ประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 14 หรือพอร์ต 3.4 ต่อกับขา SDA ซึ่งเป็นขาสำหรับรับส่งข้อมูลอนุกรมของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512

ขา 15 หรือพอร์ต 3.5 ต่อกับขา SCL ซึ่งเป็นขาอินพุตรับสัญญาณนาฬิกาของหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512

ขา 18 กับ 19 ต่อคริสตัลเพื่อเป็นฐานเวลาให้กับไมโครคอนโทรลเลอร์ โดยอาศัยความถี่ของคริสตัลเป็นตัวกำหนดฐานเวลา

ขา 20 เป็นขาราวด์ของไมโครคอนโทรลเลอร์

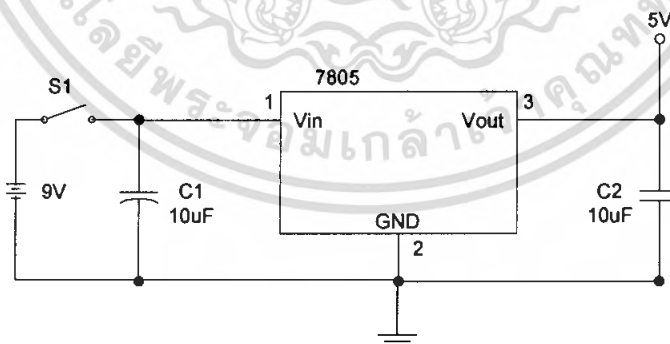
ขา 21 ถึง 27 หรือพอร์ต 2.0 ถึง พอร์ต 2.6 ต่อกับคีย์แพคเพื่อนำสัญญาณที่เกิดจากการกดคีย์ไปประมวลผลว่าได้ทำการกดคีย์แพคที่แถวใดและหลักใด เพื่อที่จะได้ทราบว่าเราได้กดปุ่มใดไปจากนั้นเมื่อรู้ว่าจะกดปุ่มใดแล้ว ไมโครคอนโทรลเลอร์ก็จะทำการประมวลผลว่าปุ่มนั้นมีคำสั่งให้ทำอะไรต่อไป

ขา 31 หรือ EA (External Access Enable) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เอง สำหรับโครงการนี้ไม่ใช้หน่วยความจำโปรแกรมภายนอก จึงต่อขานี้กับไฟ +5 โวลต์

ขา 40 ต่อกับแรงดันไฟ +5 โวลต์

### 3.1.2 ภาคจ่ายไฟ

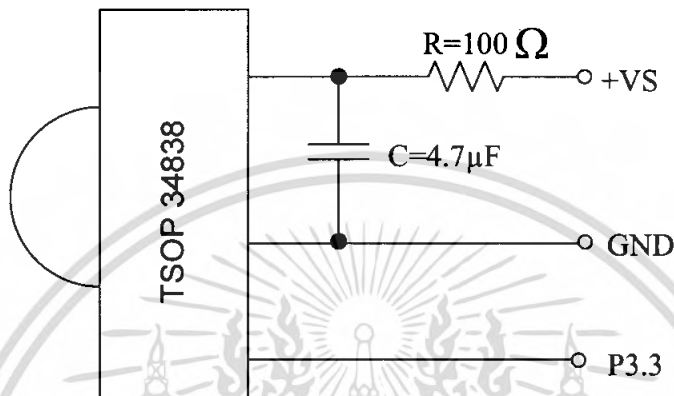
ในโครงการนี้ใช้ไอซีเรกูเลเตอร์ (REGULATOR) เบอร์ 7805 เพื่อแปลงแรงดันไฟจากถ่าน 9 โวลต์ ให้เหลือแรงดัน 5 โวลต์ และป้อนให้กับวงจรในส่วนต่างๆ เช่น ไมโครคอนโทรลเลอร์ จอแอลซีดี และตัวรับสัญญาณรีโมทคอนโทรล



รูปที่ 3.4 วงจรในส่วนของภาคจ่ายไฟ

### 3.1.3 ส่วนของวงจรรับสัญญาณอินฟราเรดจากรีโมทคอนโทรล

ในโครงงานนี้ใช้โมดูลรับสัญญาณอินฟราเรดจากรีโมทคอนโทรลเบอร์ TSOP34838 ของ VISHAY รับสัญญาณจากรีโมทคอนโทรลต้นแบบ โดยในตัวโมดูลจะมีโฟโตดีเทคเตอร์ที่รับสัญญาณอินฟราเรดที่ความถี่ 38 kHz หรือใกล้เคียง และวงจรดีมอดูเลตสัญญาณ สัญญาณที่ออกมาทางขาเอาต์พุตสามารถนำไปใช้งานกับไมโครคอนโทรลเลอร์ได้ทันที



รูปที่ 3.5 วงจรรับสัญญาณอินฟราเรดจากรีโมทคอนโทรล

### 3.1.4 ส่วนของวงจรส่งสัญญาณอินฟราเรด

วงจรส่งสัญญาณรีโมทคอนโทรลใช้ทรานซิสเตอร์ ทำหน้าที่เหมือนสวิตช์เพื่อเปิด-ปิดวงจรทำให้ LED อินฟราเรดส่งสัญญาณอินฟราเรดออกมาตามค่าที่อ่านได้ ค่าสัญญาณที่อ่านได้จะเก็บไว้ในอีมพรอม (ซึ่งสัญญาณถูกถอดคลื่นพาหะแล้ว) เมื่อนำมาสร้างใหม่ที่ส่วนวงจรส่งสัญญาณอินฟราเรดรีโมทคอนโทรล ซึ่งความละเอียดของสัญญาณใหม่นี้อาจจะไม่เท่ากับสัญญาณต้นแบบ

การใช้งาน IC 555 เป็นอะอสเตเบิลมัลติไวเบรเตอร์ (Astable Multivibrator)

การคำนวณ

จากสูตร

$$f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)C} \quad (1)$$

โดยโครงงานนี้ต้องการความถี่ 38 kHz จึงมาคำนวณเพื่อหาค่า  $R_A$ ,  $R_B$  และค่า  $C$  มีการคำนวณดังนี้ สมมติให้

ค่า  $R_B$  มีค่า 2.2 kΩ

ค่า  $C$  มีค่า 4.7 nF

เพื่อหาค่า  $R_A$  จากสูตร  $f = \frac{1}{T} = \frac{1.44}{(R_A + 2R_B)C}$  แทนค่า  $R_B, C$  ลงในสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$38k = \frac{1.44}{R_A + 2(2.2k)4.7n}$$

$$(R_A + 4.4k) = \frac{1.44}{38k \times 4.7n}$$

$$R_A + 4.4k = \frac{1.44}{0.1786m}$$

$$R_A = 8.06k - 4.4k$$

$$R_A \approx 3.6k$$

จะได้  $R_A \approx 3.6 \text{ k}\Omega$

เมื่อ  $R_A$  มีค่าประมาณ  $3.6 \text{ k}\Omega$  จึงเลือกใช้ค่า  $R_A$  ดังนี้

เลือกค่า  $R_A$  คงที่  $470 \Omega$  และ  $R_A$  ปรับค่า  $5 \text{ k}\Omega$  ดังนั้นความถี่ที่ได้มีค่าเป็นช่วงที่  $R_A$  มีค่าเท่ากับ  $470 \Omega$  จะได้ความถี่เท่ากับ

$$f = \frac{1}{T} = \frac{1.44}{(470 + 2(2.2k))4.7n} = 62.9 \text{ kHz}$$

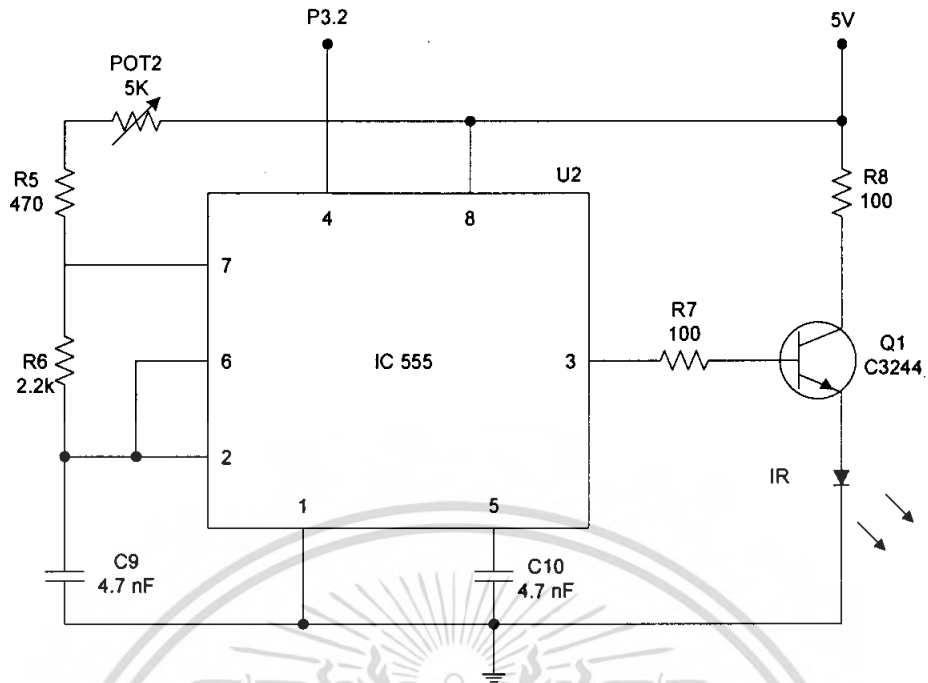
ที่  $R_A$  มีค่าเท่ากับ  $5470 \Omega$  จะได้ความถี่เท่ากับ  $= 31 \text{ kHz}$

$$f = \frac{1}{T} = \frac{1.44}{(5470 + 2(2.2k))4.7n} = 31 \text{ kHz}$$

ดังนั้นความถี่ที่ได้ก็จะอยู่ในช่วง  $31 - 62.9 \text{ kHz}$

นำค่า  $R_A$ ,  $R_B$  และ  $C$  มาเขียนวงจรได้ดังรูป

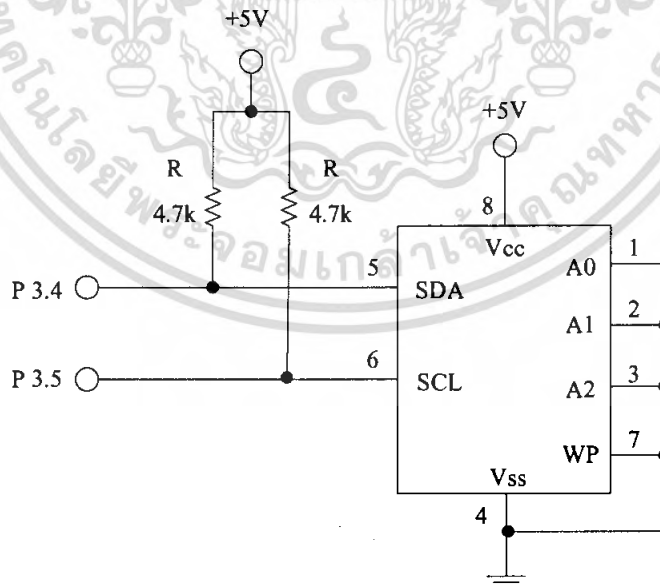
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรในส่วนของการส่งสัญญาณอินฟราเรด

### 3.1.5 ส่วนเก็บบันทึกสัญญาณ

สำหรับโครงการนี้จะใช้หน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512 ซึ่งสามารถบันทึกข้อมูลได้ 64 กิโลไบต์ สามารถเก็บรักษาข้อมูลอยู่ได้โดยไม่ต้องจ่ายไฟเลี้ยง เขียน-อ่าน-ลบได้ด้วยสัญญาณไฟฟ้า การเชื่อมต่อกับไมโครคอนโทรลเลอร์จะใช้ขาพอร์ตเพียง 2 ขา คือขา SDA กับ SCL ตามรูปที่ 3.7

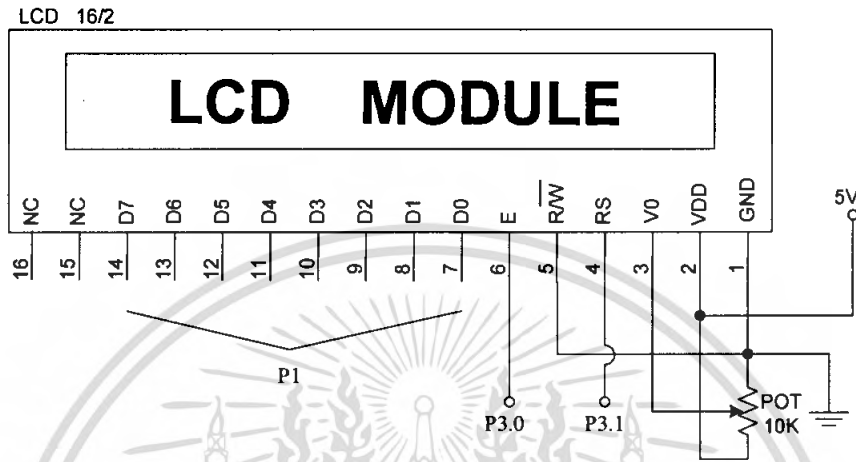


รูปที่ 3.7 การต่อใช้งานหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.6 ส่วนแสดงผล

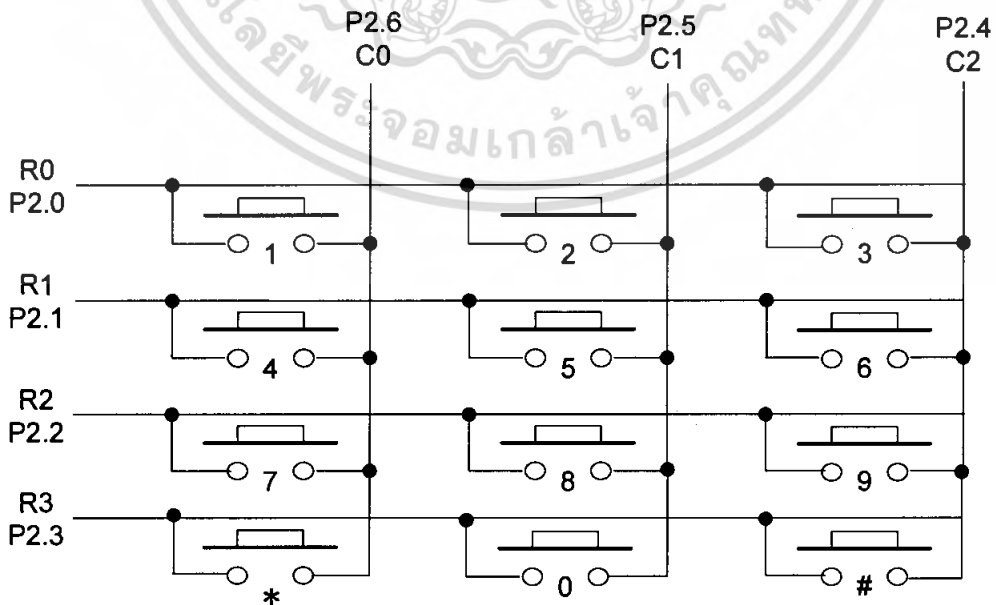
ส่วนแสดงผลใช้โมดูลแอลซีดีชนิดคอตเมตริกซ์ ขนาด 16 ตัวอักษร 2 บรรทัด เพื่อใช้แสดงผล โดยการต่อโมดูลแอลซีดีเข้ากับไมโครคอนโทรลเลอร์จะเป็นดังรูป 3.8 ขา D0-D7 จะต่อเข้ากับพอร์ต 1 ของไมโครคอนโทรลเลอร์ ขา E ต่อกับพอร์ต 3.0 ขา RS ต่อกับพอร์ต 3.1 ขา VO ต่อกับตัวต้านทานปรับค่าได้เพื่อปรับความเข้มของการแสดงผล



รูปที่ 3.8 การต่อใช้งาน โมดูลแอลซีดี

### 3.1.7 ส่วนของคีย์เมตริกซ์

ในส่วนของคีย์เมตริกซ์นี้จะใช้คีย์แพคของทรานซิสต์ขนาด 4x3 เป็นตัวรับคำสั่งจากผู้ใช้งาน ซึ่งจะนำมาต่อกับพอร์ต 2.0-2.6 ของไมโครคอนโทรลเลอร์โดยคีย์เมตริกซ์จะเป็นอินพุตให้ไมโครคอนโทรลเลอร์ทำงานตามฟังก์ชันต่างๆ ของคีย์แต่ละคีย์ตามที่กำหนดไว้ในโปรแกรมโดยสามารถอ่านค่าอธิบายได้ที่จอแสดงผลของแอลซีดี



รูปที่ 3.9 วงจรภายในของคีย์แพคของทรานซิสต์ขนาด 4x3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 การออกแบบซอฟต์แวร์ควบคุมวงจร

การออกแบบจะแบ่งออกเป็น 2 ส่วนหลักๆ คือ

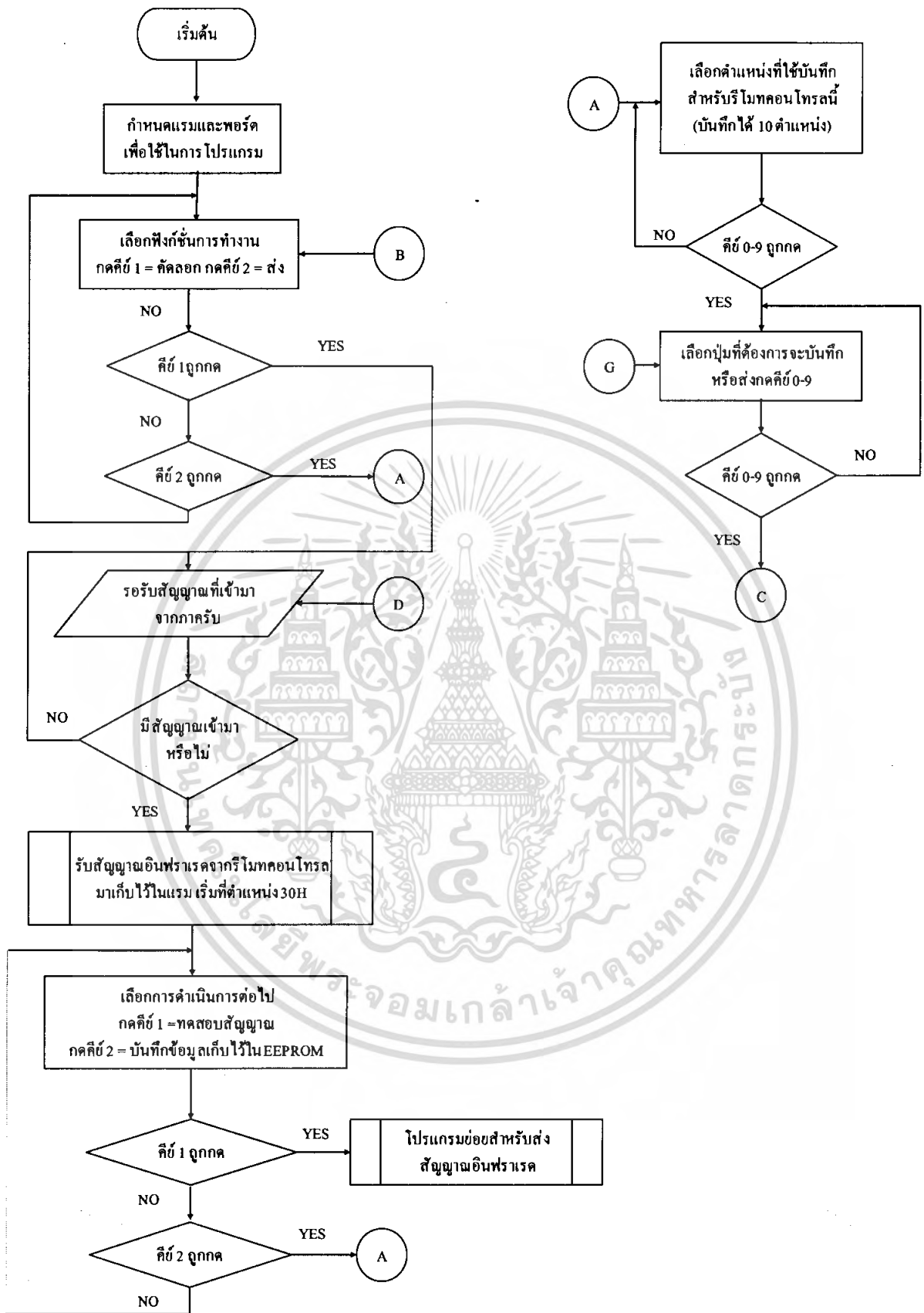
#### 3.2.1 การออกแบบซอฟต์แวร์ส่วนรับและบันทึกสัญญาณอินฟราเรด

การออกแบบซอฟต์แวร์ส่วนรับและบันทึกสัญญาณอินฟราเรดจากรีโมทคอนโทรลนั้น เมื่อเริ่มมีข้อมูลเข้ามาจะทำการเซตไทมเมอร์ให้เริ่มนับ แล้วเมื่อสถานะของสัญญาณอินฟราเรดเปลี่ยนไปให้นำค่าเวลาที่ไทมเมอร์นับได้ในขณะนั้นไปเก็บไว้ในแรมและเริ่มเซตไทมเมอร์ใหม่ เมื่อได้สัญญาณมาเก็บไว้ในแรมจนครบแล้ว จะทำการนำค่าที่เก็บไว้ในแรมไปบันทึกลงในหน่วยความจำอีพีรอมเพื่อให้สามารถเก็บสัญญาณไว้ได้แม้ว่าหยุดจ่ายไฟให้กับชิ้นงานแล้ว รายละเอียดการออกแบบซอฟต์แวร์ส่วนรับและบันทึกสัญญาณอินฟราเรดจากรีโมทคอนโทรลสามารถดูได้จากไฟล์ชาร์ตโปรแกรมย่อยส่วนรับสัญญาณอินฟราเรดและไฟล์ชาร์ตโปรแกรมย่อยสำหรับบันทึกข้อมูลลงในอีพีรอมในบทนี้

#### 3.2.2 การออกแบบซอฟต์แวร์ส่วนส่งสัญญาณอินฟราเรด

การออกแบบซอฟต์แวร์ส่วนส่งสัญญาณอินฟราเรด ต้องนำค่าสัญญาณของปุ่มที่ต้องการส่งที่บันทึกไว้ในอีพีรอมออกมาเก็บไว้ในแรมก่อนเสมอ เพื่อป้องกันการเกิดคีย์ของสัญญาณที่จะส่งอันเนื่องมาจากต้องคอยการอ่านข้อมูลมาจากอีพีรอม จากนั้นจะนำค่าสัญญาณในแรมสร้างเป็นรูปคลื่นพัลส์ส่งออกมา แล้วมอดูเลตรวมกับความถี่ที่ภาคส่งสร้างขึ้น ซึ่งใช้ความถี่ที่ใกล้เคียง 38 kHz เป็นความถี่พาหะ เนื่องจากเครื่องใช้ไฟฟ้าส่วนใหญ่ที่ใช้อินฟราเรดรีโมทคอนโทรลในการควบคุมนิยมใช้ความถี่ช่วงนี้ในการส่งสัญญาณควบคุม หากว่าความถี่ไม่ตรงกับที่เครื่องใช้ไฟฟ้าต้องการ ก็สามารถปรับได้ที่ภาคส่งของชิ้นงาน การส่งสัญญาณที่มีลอจิก "1" โปรแกรมจะทำการเซตพอร์ต 3.2 แล้วหน่วงเวลาที่ส่งนานตามค่าเวลาที่เก็บไว้ในแรม ส่วนสัญญาณที่เป็นลอจิก "0" โปรแกรมจะทำในลักษณะเดียวกัน แต่เปลี่ยนเป็นการเคลียร์พอร์ต 3.2 รายละเอียดการออกแบบซอฟต์แวร์ส่วนส่งสัญญาณอินฟราเรดสามารถดูได้จากไฟล์ชาร์ตโปรแกรมย่อยการส่งสัญญาณอินฟราเรด

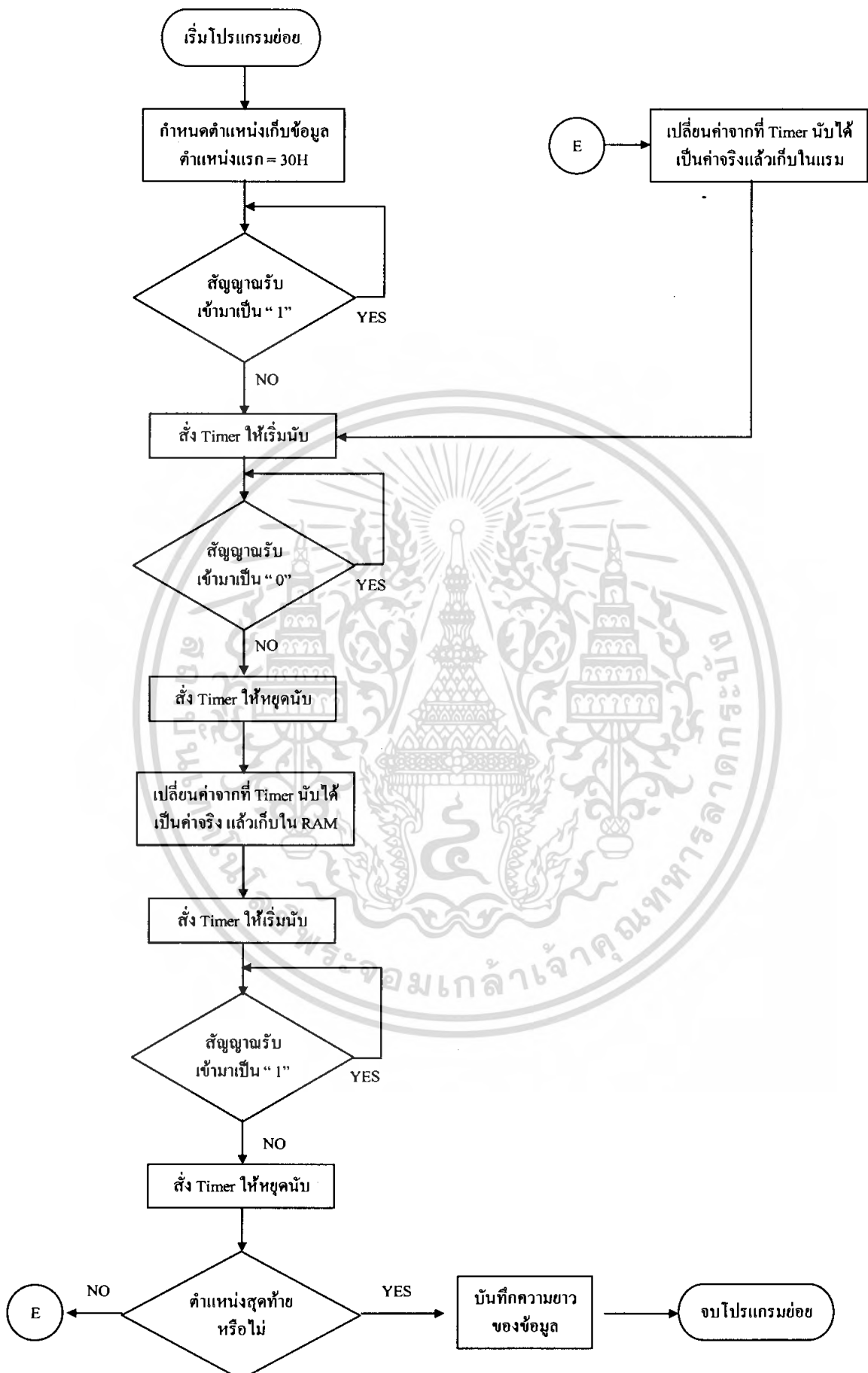
การออกแบบซอฟต์แวร์แต่ละส่วนนั้น สามารถเขียนเป็นไฟล์ชาร์ตการทำงานได้ดังนี้



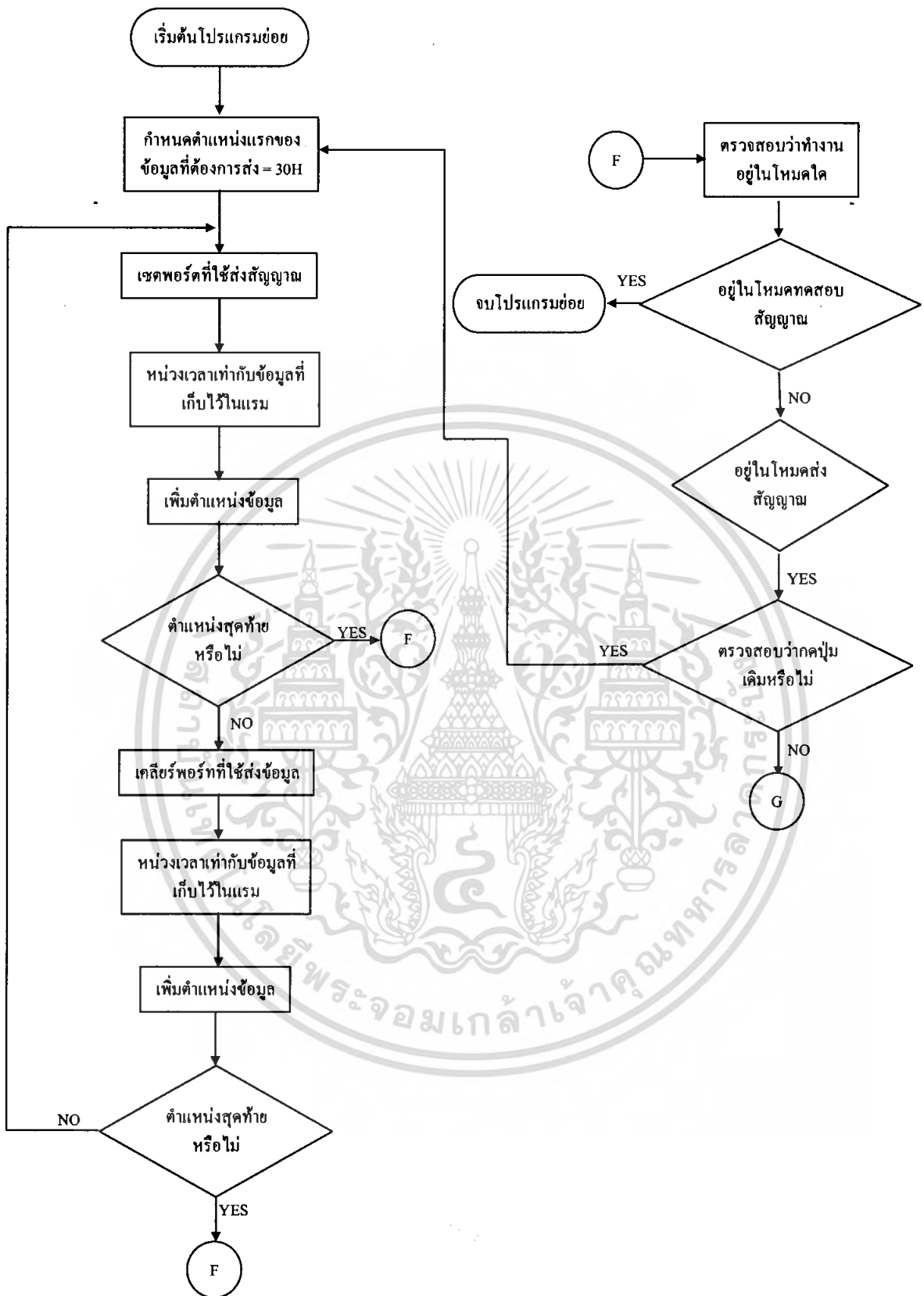
รูปที่ 3.10 โฟลว์ชาร์ตในส่วนของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



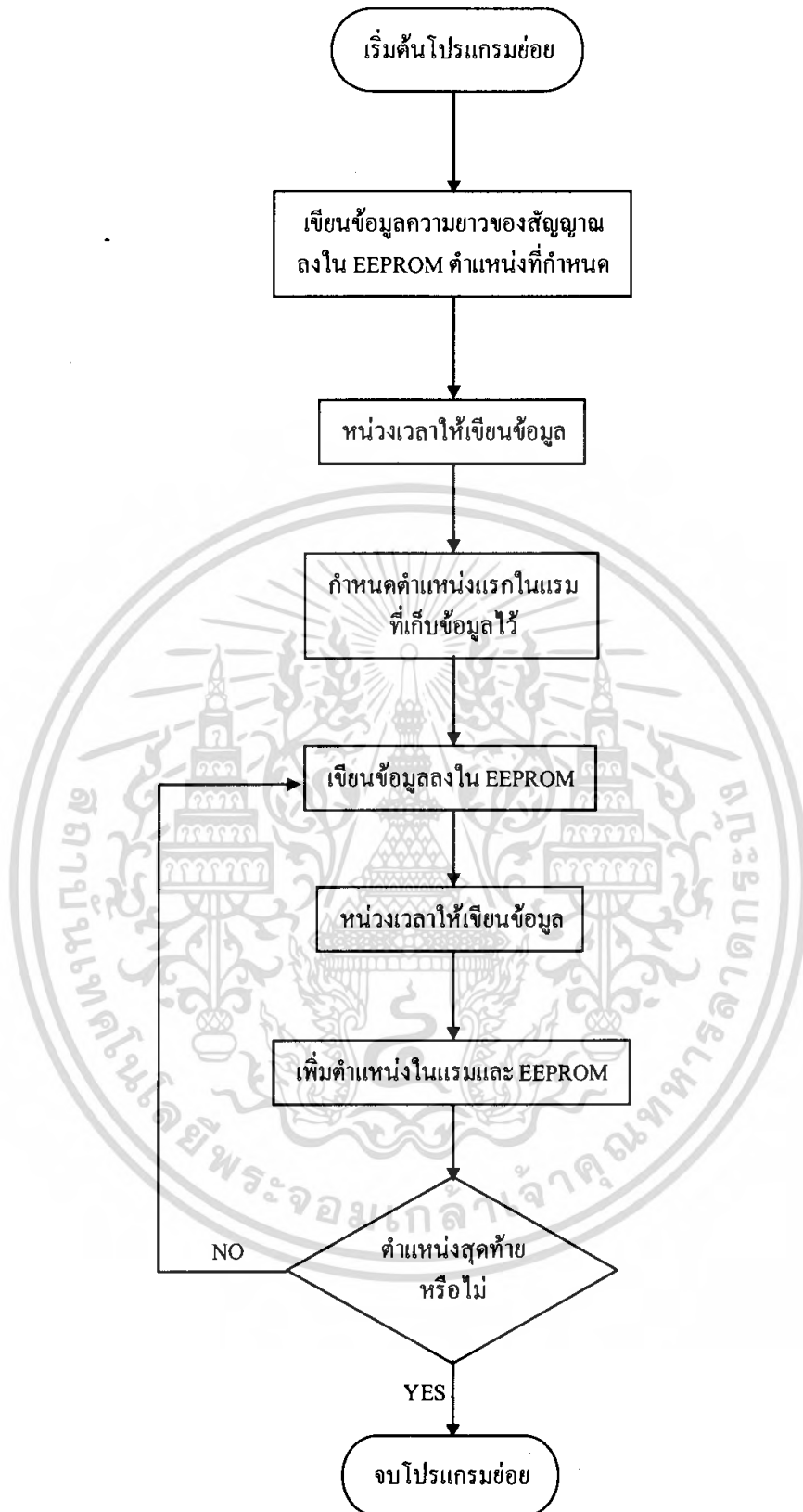


เอกสารนี้รูปที่ 3.12 โฟล์วชาร์ต โปรแกรมย่อยรับสัญญาณอินฟราเรดจากรีโมทคอนโทรลมาเก็บในแรมด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 ฟลิวชาร์ต โปรแกรมย่อยสำหรับส่งสัญญาณอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ฟลิวชาร์ต โปรแกรมย่อยสำหรับบันทึกข้อมูลลงในอีอีพรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 การจัดพื้นที่เก็บข้อมูลในหน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512

ในโครงการนี้ได้กำหนดให้อีอีพรอมสามารถเก็บข้อมูลสัญญาณอินฟราเรดจากรีโมทคอนโทรลทั่วไป เช่น รีโมทคอนโทรลสำหรับโทรทัศน์ หรือเครื่องเล่นดีวีดีได้ 10 ตำแหน่ง โดยในแต่ละตำแหน่งสามารถบันทึกข้อมูลได้ 10 บั้มกค เพราะฉะนั้นในการใช้งานจริง หากรีโมทคอนโทรลที่ต้องการบันทึกมีมากกว่า 10 บั้มกคก็สามารถจะบันทึกไว้ที่ตำแหน่งอื่นอีกหนึ่งตำแหน่ง หรือมากกว่าหนึ่งตำแหน่งก็ได้

แอดเดรสของอีอีพรอมเบอร์ 24LC512 เริ่มตั้งแต่ 0000H ถึง FFFFH (หน่วยความจำอีอีพรอมอนุกรมเบอร์ 24LC512 สามารถเก็บข้อมูลได้ 64 กิโลไบต์) ส่วนเก็บบันทึกข้อมูลของรีโมทคอนโทรลทั่วไป 10 ตำแหน่ง อยู่ที่แอดเดรสตั้งแต่ 0000H ถึง 9FFFH

ส่วนรีโมทคอนโทรลที่ใช้กับเครื่องปรับอากาศ โครงการนี้ได้แบ่งพื้นที่เก็บข้อมูลไว้ 3 ส่วนโดย ส่วนแรกสามารถเก็บข้อมูลได้ 16 ตำแหน่ง ซึ่งในส่วนนี้ได้ออกแบบให้แสดงผลบนจอแอลซีดีสำหรับบันทึกสัญญาณแต่ละอุณหภูมิ(15-30 องศาเซลเซียส)โดยเฉพาะเพื่อความสะดวกในการใช้งาน แต่หากบันทึกสัญญาณอื่นลงไปก็สามารถใช้งานได้เช่นกัน ส่วนแรกนี้อยู่ที่แอดเดรสตั้งแต่ A000H ถึง AFFFH

ส่วนที่สองออกแบบให้เป็นส่วนเก็บบันทึกสัญญาณเปิด-ปิด เครื่องปรับอากาศ เพื่อความสะดวกในการใช้งาน และเช่นเดียวกับส่วนแรก หากบันทึกสัญญาณอื่นลงไปก็สามารถใช้งานได้เช่นกัน ส่วนที่สองนี้อยู่ที่แอดเดรสตั้งแต่ F000H ถึง FFFFH

ส่วนที่สาม ได้ออกแบบไว้เป็นพื้นที่เก็บข้อมูลสำรอง เพื่อรองรับบั้มกคอื่นๆ บนรีโมทคอนโทรลของเครื่องปรับอากาศ สามารถบันทึกได้ 64 ตำแหน่ง อยู่ที่แอดเดรสตั้งแต่ B000H ถึง EFFFH

## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 การทดลอง

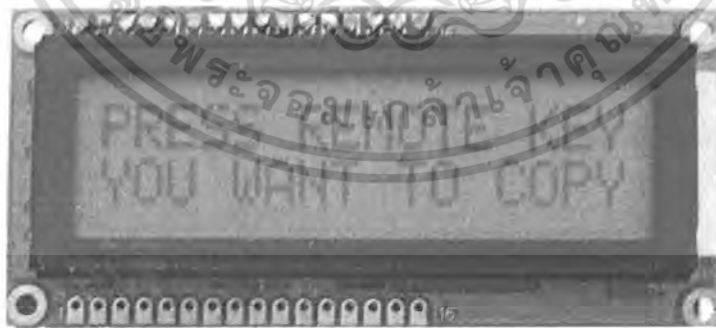
แบ่งเป็น 2 ตอน โดยตอนที่ 1 เป็นส่วนของการคัดลอกสัญญาณอินฟราเรดจากรีโมทคอนโทรล ตอนที่ 2 เป็นส่วนของการส่งสัญญาณเพื่อใช้งาน มีขั้นตอนการทดลองดังนี้

##### 4.1.1 ตอนที่ 1 การคัดลอกสัญญาณ

1.เมื่อจ่ายไฟให้กับชิ้นงาน จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะคัดลอกสัญญาณหรือใช้งานส่งสัญญาณ ให้เลือกคัดลอกสัญญาณแล้วรีโมทคอนโทรลต้นแบบมาส่งสัญญาณเข้าที่ส่วนรับสัญญาณของชิ้นงาน



รูปที่ 4.1 จอแอลซีดีแสดงข้อความให้เลือกว่าจะคัดลอกสัญญาณหรือใช้งานส่งสัญญาณ



รูปที่ 4.2 จอแอลซีดีแสดงข้อความรอรับสัญญาณจากรีโมทคอนโทรล

2.เมื่อรับสัญญาณมาแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะทดสอบสัญญาณหรือบันทึกสัญญาณ ให้ลองทดสอบดูก่อนว่ารับสัญญาณมาถูกต้องหรือไม่ ถ้ารับมาถูกต้องก็เลือกบันทึกสัญญาณได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 จอแอลซีดีแสดงข้อความให้เลือกว่าจะทดสอบสัญญาณหรือบันทึกสัญญาณ

3. เมื่อเลือกบันทึกสัญญาณแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกในโหมดเครื่องปรับอากาศ หรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ ถ้าเลือกโหมดเครื่องปรับอากาศจอแอลซีดีจะแสดงข้อความให้เลือกส่วนเก็บบันทึกข้อมูล 3 ส่วน คือ ส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิ ส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ และส่วนเก็บข้อมูลสำรองสำหรับบูมอื่นๆ ที่เหลือ



รูปที่ 4.4 จอแอลซีดีแสดงข้อความให้เลือกว่าจะบันทึกในโหมดเครื่องปรับอากาศหรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ



รูปที่ 4.5 จอแอลซีดีจะแสดงข้อความให้เลือกส่วนเก็บบันทึกข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิ สามารถบันทึกได้ตั้งแต่อุณหภูมิ 15-30 องศาเซลเซียส



รูปที่ 4.7 จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ



รูปที่ 4.8 จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสำรองสำหรับปุ่มอื่นๆ ที่เหลือ

4. ถ้าเลือกโหมดเครื่องใช้ไฟฟ้าชนิดอื่นๆ จอแอลซีดีจะแสดงข้อความให้เลือกตำแหน่งที่จะบันทึก และเลือกปุ่มที่จะบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 จอแอลซีดีจะแสดงข้อความให้เลือกตำแหน่งที่จะบันทึก



รูปที่ 4.10 จอแอลซีดีจะแสดงข้อความให้เลือกปุ่มที่จะบันทึก

5.เมื่อทำการบันทึกเสร็จแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกปุ่มต่อไป หรือกลับไป

เมนูหลัก



รูปที่ 4.11 จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกปุ่มต่อไป หรือกลับไปเมนูหลัก

ทำตามขั้นตอนนี้บันทึกจนครบทุกปุ่มแล้วเลือกกลับเมนูหลักเพื่อไปทำตอนที่ 2 ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.2 ตอนที่ 2 การส่งสัญญาณ

1. ในหน้าเมนูหลัก จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะคัดลอกสัญญาณหรือใช้งานส่งสัญญาณให้เลือกส่งสัญญาณ

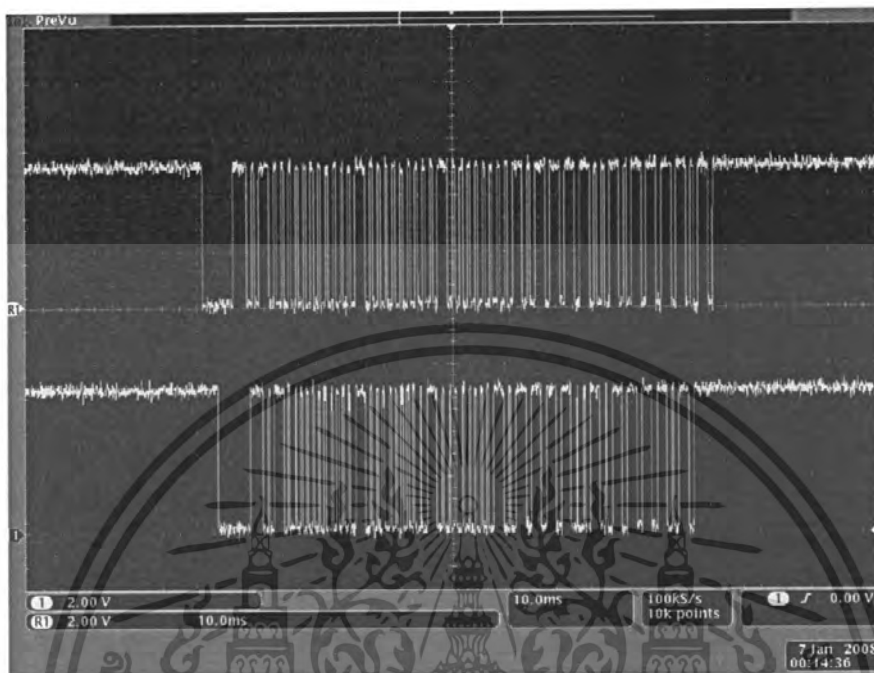
2. เมื่อเลือกส่งสัญญาณแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะส่งสัญญาณในโหมดเครื่องปรับอากาศ หรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ ถ้าเลือกโหมดเครื่องปรับอากาศจอแอลซีดีจะแสดงข้อความให้เลือกส่วนเก็บบันทึกข้อมูล 3 ส่วน คือ ส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิ ส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ และส่วนเก็บข้อมูลสำรองสำหรับปุ่มอื่นๆ ที่เหลือ ให้เลือกส่วนที่ต้องการแล้วจึงทำการส่งสัญญาณ

3. ถ้าต้องการส่งสัญญาณในโหมดเครื่องใช้ไฟฟ้าชนิดอื่น ให้เลือกตำแหน่งที่บันทึกรีโมทคอนโทรลที่ต้องการส่งไว้ และเลือกปุ่มที่จะส่งสัญญาณ เหมือนกับตอนที่บันทึก เมื่อกดปุ่มที่ต้องการแล้วไมโครคอนโทรลเลอร์จะทำหน้าที่ดึงข้อมูลที่เกี่ยวข้องอยู่ในอีพ롬จากตำแหน่งที่ระบุโดยตัวแปรข้างต้น มาสร้างเป็นพัลส์เลียนแบบ แล้วส่งออกไปทันที

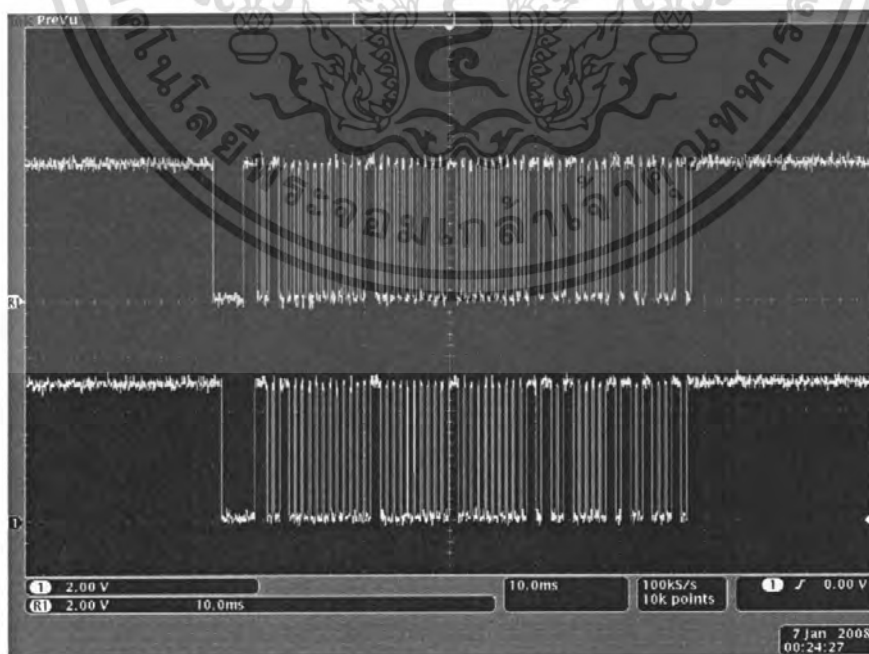
รีโมทคอนโทรลที่นำมาทดลองมีอยู่ 4 ชนิด คือรีโมทคอนโทรลของโทรทัศน์สี Panasonic, รีโมทคอนโทรลของโทรทัศน์สี Sony, รีโมทคอนโทรลของเครื่องเล่นดีวีดี LG และรีโมทคอนโทรลของเครื่องปรับอากาศ Saijo Denki ทำการส่งสัญญาณที่คัดลอกมาเปรียบเทียบกับสัญญาณจากรีโมทคอนโทรลต้นแบบ ได้ผลการทดลองดังนี้

## 4.2 ผลการทดลอง

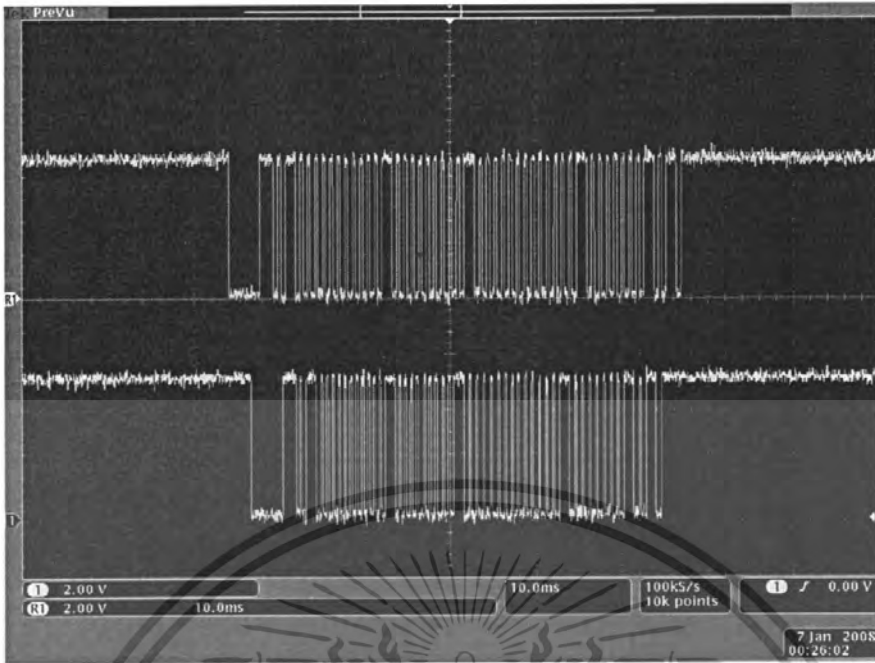
### 4.2.1 ผลการทดลองของรีโมทคอนโทรลโทรทัศน์ Panasonic



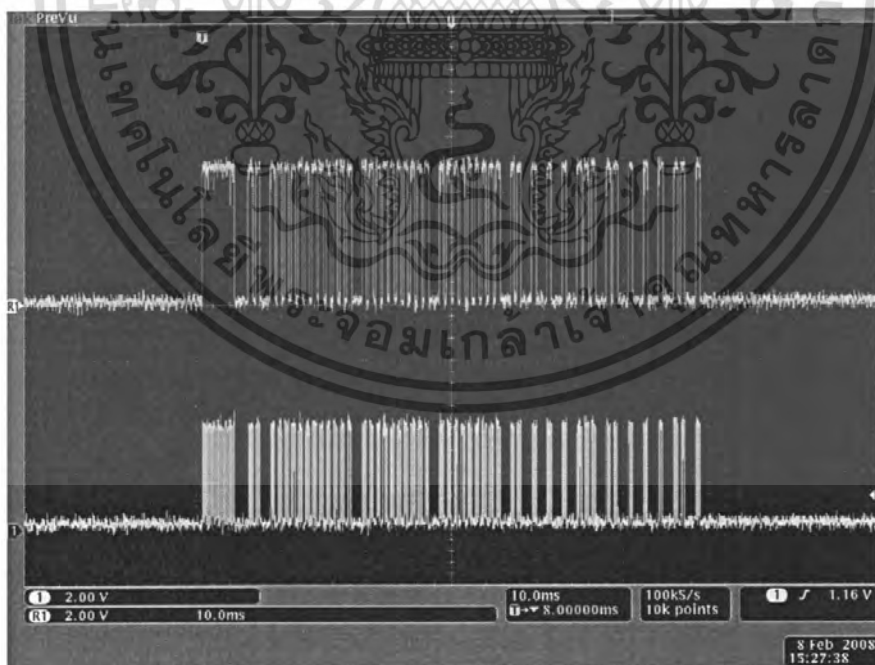
รูปที่ 4.12 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก



รูปที่ 4.13 สัญญาณเมื่อกดปุ่ม Channel 7 โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



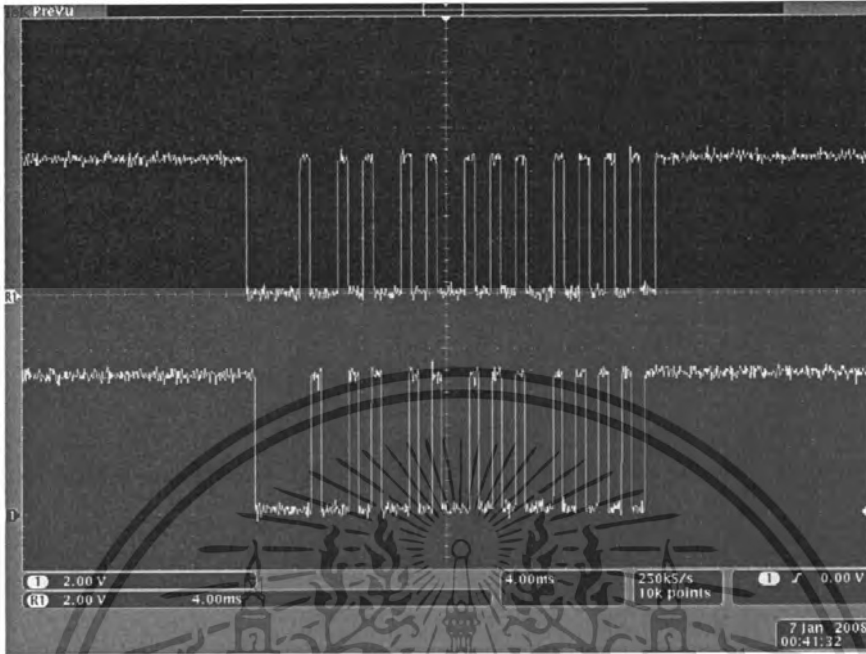
รูปที่ 4.14 สัญญาณเมื่อกดปุ่มเพิ่มเสียง โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการกดคอก



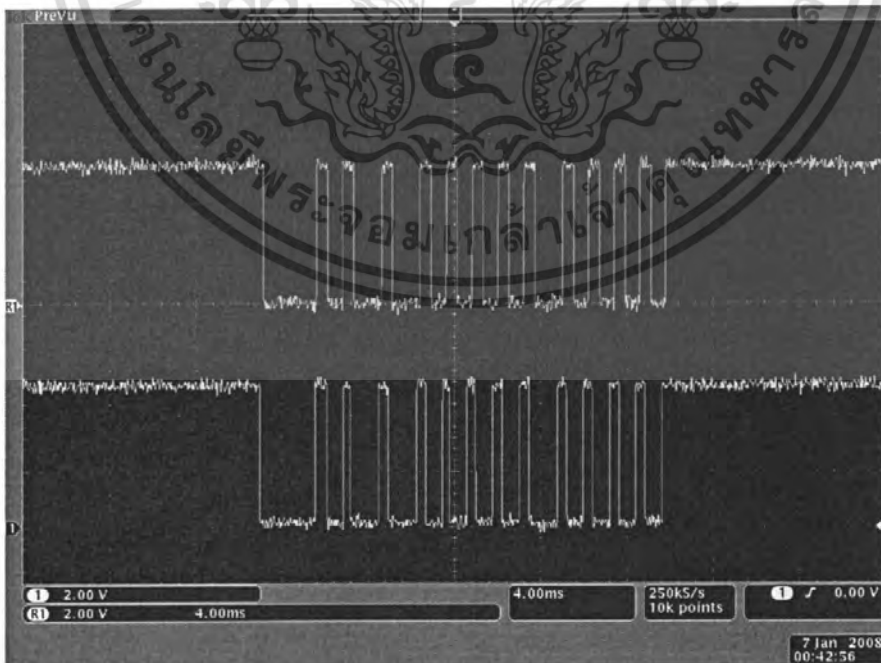
รูปที่ 4.15 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

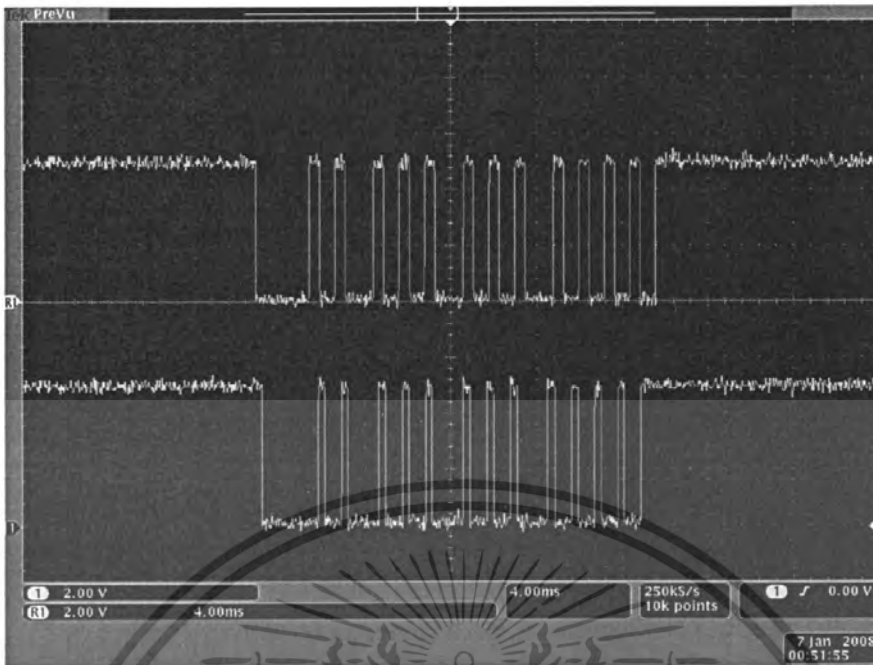
#### 4.2.2 ผลการทดลองของรีโมทคอนโทรลโทรทัศน์ SONY



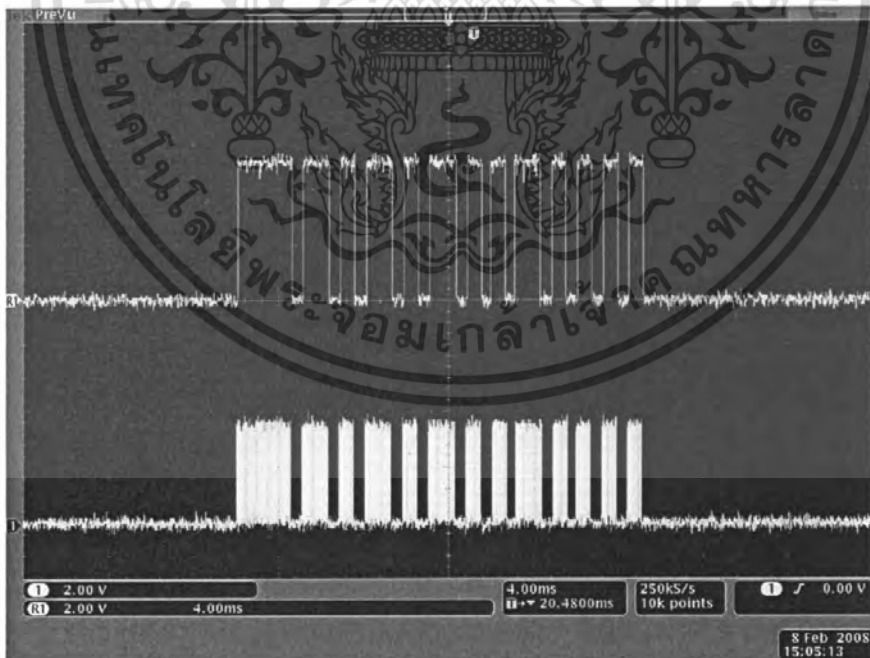
รูปที่ 4.16 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก



รูปที่ 4.17 สัญญาณเมื่อกดปุ่ม Channel 7 โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาระดับบัณฑิตศึกษาเท่านั้น ไม่ควรเอาไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



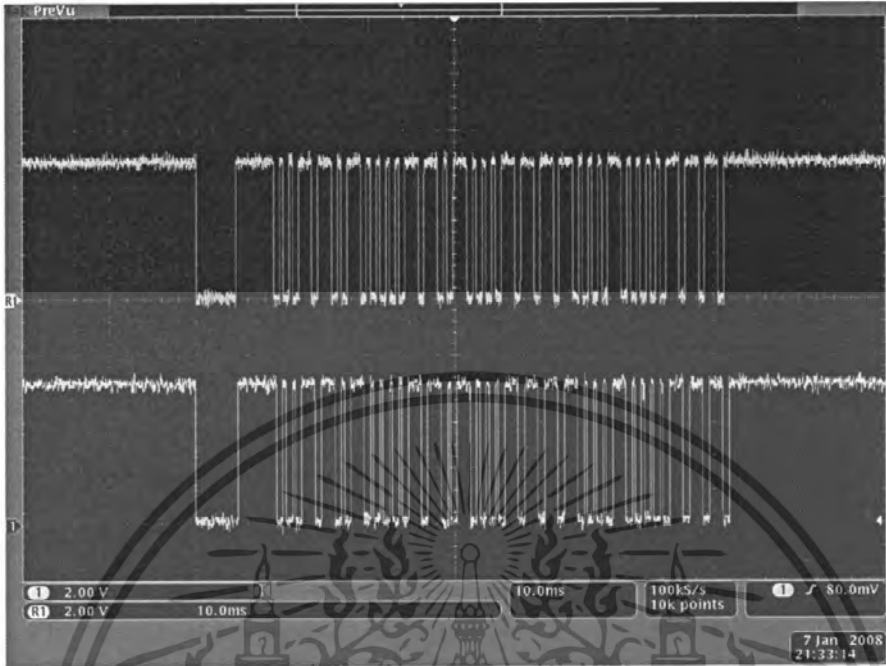
รูปที่ 4.18 สัญญาณเมื่อกดปุ่มเพิ่มเสียง โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก



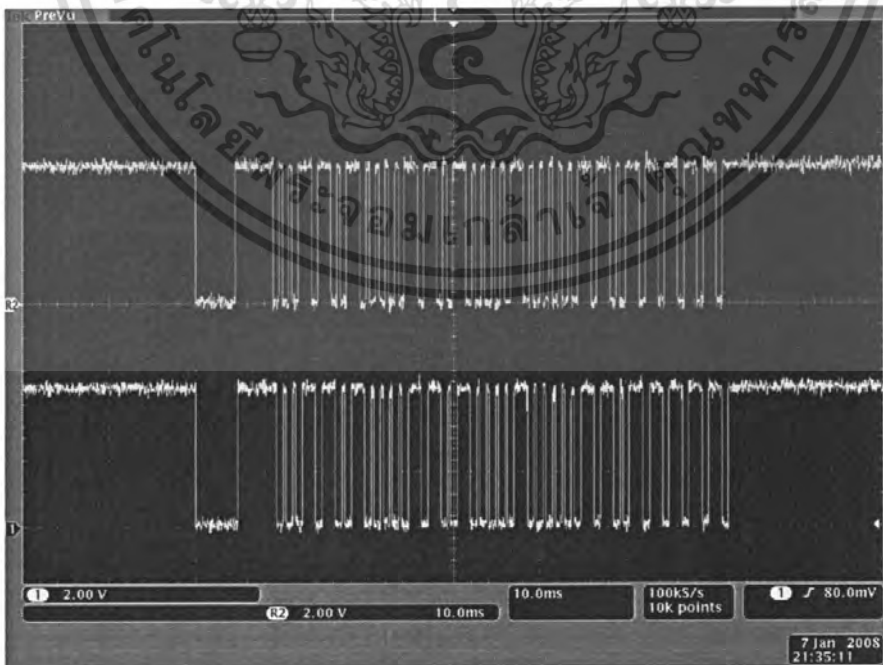
รูปที่ 4.19 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

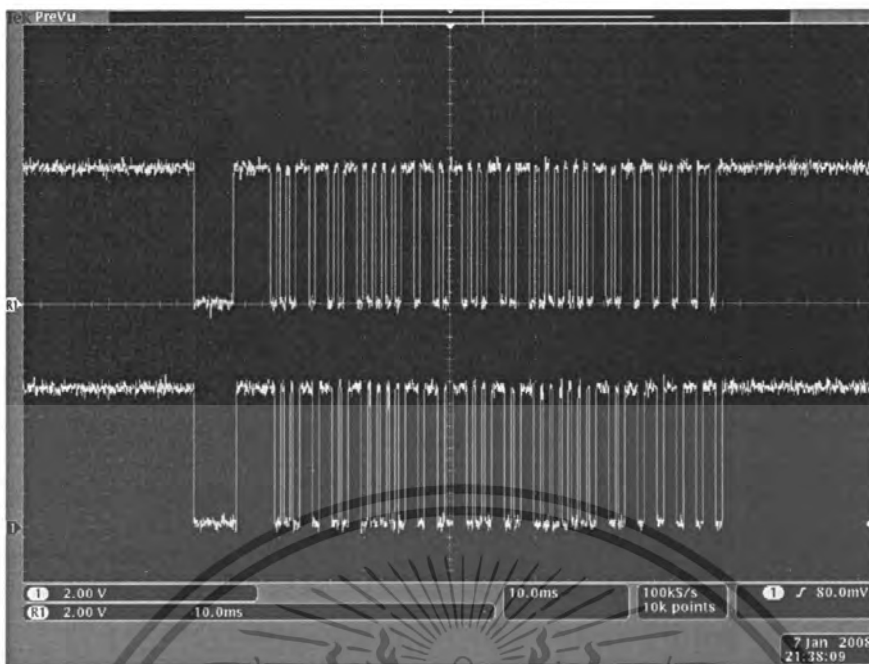
### 4.2.3 ผลการทดลองของรีโมทคอนโทรลเครื่องเล่นดีวีดี LG



รูปที่ 4.20 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก



รูปที่ 4.21 สัญญาณเมื่อกดปุ่ม Play โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ติดต่อแบบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



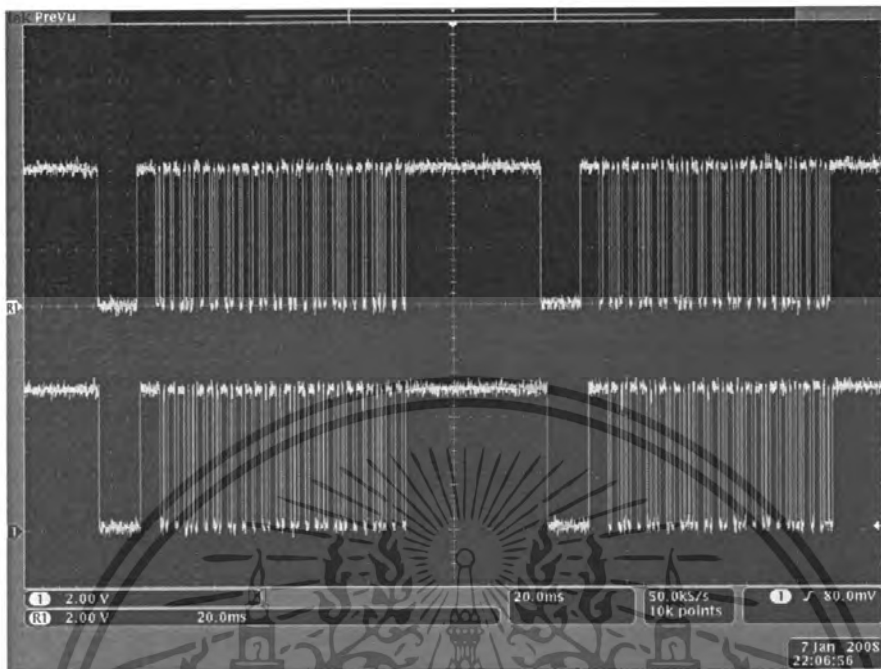
รูปที่ 4.22 สัญญาณเมื่อกดปุ่ม Stop โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการกดคีย์



รูปที่ 4.23 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.4 ผลการทดลองของรีโมทคอนโทรลเครื่องปรับอากาศ Saijo Denki

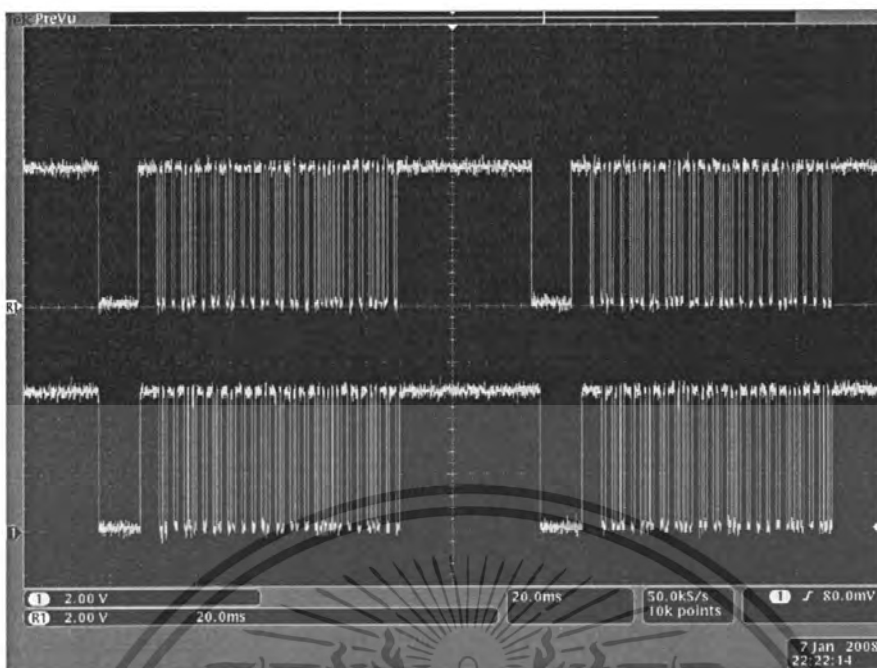


รูปที่ 4.24 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก



รูปที่ 4.25 สัญญาณเมื่อกดปุ่มเพิ่มอุณหภูมิเป็น 25 องศาเซลเซียส โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 สัญญาณเมื่อกดปุ่มลดอุณหภูมิเป็น 24 องศาเซลเซียส โดยสัญญาณรูปบน เป็นสัญญาณจากรีโมทคอนโทรลต้นแบบ ส่วนสัญญาณรูปล่าง เป็นสัญญาณที่ได้จากการคัดลอก



รูปที่ 4.27 สัญญาณเมื่อกดปุ่ม Power โดยสัญญาณรูปบน เป็นสัญญาณอินพุตของวงจรส่งสัญญาณอินฟราเรด ส่วนสัญญาณรูปล่างเป็นสัญญาณเอาต์พุตที่มีการผสมความถี่ 38 kHz เข้าไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 รูปสักริตการทำงานของเครื่องตัดสกรีนมือถือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5 บทวิจารณ์และสรุป

### ส่วนรับสัญญาณอินฟราเรดจากรีโมทคอนโทรลและเก็บบันทึกในหน่วยความจำอีอีพรอม

ในส่วนนี้พบว่าสามารถทำงานได้ตามที่ต้องการ คือ โมดูลรับสัญญาณอินฟราเรดจากรีโมทคอนโทรลสามารถรับสัญญาณจากรีโมทคอนโทรลได้ทุกอันเท่าที่สามารถหามาทดสอบได้ ระยะทางที่รับสัญญาณได้มีระยะไกล สามารถรับได้ไกลสุดอย่างน้อย 4 เมตร ในแนวเส้นตรง

ในส่วนของ การเก็บข้อมูลซึ่งใช้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผลและส่งข้อมูลไปเก็บไว้ที่หน่วยความจำอีอีพรอม (สามารถทดสอบสัญญาณก่อนที่จะบันทึกได้) สามารถเก็บข้อมูลฐานเวลาของพัลส์ได้สูงสุด 226 ข้อมูลหรือ 226 ไบต์สำหรับสัญญาณ 1 ปุ่มกด โดยที่ฐานเวลาของพัลส์แต่ละตัว ค่า “1” และ “0” ต้องยาวไม่เกิน 32 มิลลิวินาที (ถ้ายาวเกิน 32 มิลลิวินาที จะถือว่าข้อมูลสิ้นสุดตรงนั้น ทำให้สัญญาณที่รับมาได้ผิดพลาด) และสามารถเก็บข้อมูลสัญญาณของรีโมทคอนโทรลเครื่องใช้ไฟฟ้าทั่วไปได้ 10 ตำแหน่ง ตำแหน่งละ 10 ปุ่มกด (ในการใช้งานจริง หากรีโมทคอนโทรลที่ต้องการบันทึกมีมากกว่า 10 ปุ่มกดก็สามารถจะบันทึกไว้ที่ตำแหน่งอื่นอีกหนึ่งตำแหน่ง หรือมากกว่าหนึ่งตำแหน่งก็ได้)

สำหรับรีโมทคอนโทรลของเครื่องปรับอากาศ มีพื้นที่สำหรับเก็บข้อมูลทั้งหมด 96 ข้อมูล แบ่งเป็น 3 ส่วนเพื่อความสะดวกในการใช้งาน

ในการบันทึกข้อมูลเราสามารถบันทึกที่ตำแหน่งไหน หรือที่ปุ่มใดก็ได้ และไม่ต้องมีการเรียงลำดับการบันทึก การแบ่งเป็นหลายๆ ส่วนในโปรแกรมนั้นเพียงเพื่อเพิ่มความสะดวกในการใช้งานเท่านั้น เพราะฉะนั้นแม้ว่าจะบันทึกข้อมูลไม่ตรงกับพื้นที่สำหรับเก็บข้อมูลที่ตั้งไว้ ก็ยังสามารถใช้งานได้ตามปกติ

### ส่วนส่งสัญญาณอินฟราเรดที่รับมาไปควบคุมอุปกรณ์ไฟฟ้า

ในส่วนนี้ไมโครคอนโทรลเลอร์จะนำข้อมูลที่เก็บไว้ในหน่วยความจำอีอีพรอม มาสร้างเป็นพัลส์แล้วส่งออกไป แต่การส่งสัญญาณรีโมทคอนโทรลจะต้องผสมความถี่วิทยุเข้าไปด้วยเพื่อป้องกันสัญญาณรบกวนระหว่างส่งสัญญาณ จึงต้องนำสัญญาณที่ได้จากไมโครคอนโทรลเลอร์มาผ่านวงจรมอดูเลตที่สามารถปรับความถี่ได้ โดยใช้ตัวต้านทานปรับค่าได้ เพราะว่าเครื่องใช้ไฟฟ้าบางเครื่องอาจใช้ความถี่ต่างจากเครื่องอื่น แต่ก็อยู่ในช่วงใกล้ๆ กันมีทั้ง 36 kHz, 38 kHz และ 40 kHz เป็นต้น

ในการทดลองส่งสัญญาณเพื่อควบคุมอุปกรณ์ไฟฟ้า พบว่าสามารถใช้งานได้จริง สามารถควบคุมอุปกรณ์ไฟฟ้าได้เหมือนกับรีโมทคอนโทรลต้นแบบ แต่ก็มีปัญหาอยู่บ้างคือระยะทางที่ส่งได้ไกลสุด 4-5 เมตร ในแนวเส้นตรงน้อยกว่ารีโมทคอนโทรลทั่วไป และหากทิศทางผิดเพี้ยนไปเพียงเล็กน้อยก็สามารถทำให้ระยะทางที่ส่งได้สั้นลงมาก

## แนวทางพัฒนาต่อ

- 1.ปรับเปลี่ยนวงจรในส่วนของวงจรส่งสัญญาณอินฟราเรด เพื่อเพิ่มระยะทางที่สามารถส่งสัญญาณได้
- 2.ถ้าต้องการที่จะเก็บข้อมูลของรีโมทคอนโทรลให้มากกว่าเดิม อาจทำได้โดยการใช้หน่วยความจำอีอีพรอมที่มีความจุมากขึ้น หรืออาจจะพัฒนาในส่วนของโปรแกรมให้สามารถเก็บข้อมูลได้มากกว่าเดิมโดยที่ไม่ต้องเปลี่ยนหน่วยความจำ
- 3.พัฒนาให้สามารถใช้งานได้อย่างสะดวกมากยิ่งขึ้นเช่น เพิ่มจำนวนปุ่มกด ลดขนาดและน้ำหนักของรีโมทคอนโทรล เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

1. เจน สงสมพันธุ์, “โทรทัศน์สีภาคสนาม” โรงเรียนอิเล็กทรอนิกส์กรุงเทพ, 2534
2. ชัยวัฒน์ ลิ้มพรจิตรวิไล, วรพจน์ กรแก้ววัฒนกุล, “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลช” บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด -
3. รศ สมยศ จุณณะปิยะ , “ การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51” คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
4. [www.alldatasheet.com](http://www.alldatasheet.com)
5. [www.inex.in.th](http://www.inex.in.th)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
;PROJECT: __MULTIFUNCTION REMOTE CONTROL DUPLICATOR
;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

;-----
; Define Port&Pin Name
;-----

```

```

KPAD_ROW0      BIT      P2.0
KPAD_ROW1      BIT      P2.1
KPAD_ROW2      BIT      P2.2
KPAD_ROW3      BIT      P2.3
KPAD_COL2      BIT      P2.4
KPAD_COL1      BIT      P2.5
KPAD_COLO      BIT      P2.6

```

```

LCD_EN         BIT      P3.0
LCD_RS         BIT      P3.1
TX_IR_SIGNAL   BIT      P3.2
RX_IR_SIGNAL   BIT      P3.3
SDA            BIT      P3.4
SCL            BIT      P3.5

```

```

;-----
; Define User Register
;-----

```

```

KPAD_DATA      EQU      020H
LCD_ADDR       EQU      021H
LCD_DATA       EQU      022H
COUNT_DATA    EQU      023H
BUFF           EQU      024H
REPEAT         EQU      025H
DIGIT_1        EQU      026H
DIGIT_2        EQU      027H
SUB            EQU      028H
FLAG           EQU      02AH
I2C_ACK        BIT      FLAG.0
FUNC           BIT      FLAG.2
SEL_SEND       BIT      FLAG.3
DIGIT_CHK      BIT      FLAG.4
TEM_SPA        BIT      FLAG.5
EEP_DATA       EQU      02BH
EEP_AL         EQU      02CH
EEP_AH         EQU      02DH
I2C_ADDR       EQU      02EH
I2C_DATA       EQU      02FH

```

```

EEPROM_ID      EQU      10100000B ; Define I2C Slave Address

```

```

;-----
; Main Program.
;-----

```

```

ORG            0000H
MOV            P2,#0FFH
MOV            P1,#000H
CLR            LCD_EN
CLR            LCD_RS
CLR            TX_IR_SIGNAL

```

```

BUFF_INIT:    MOV            R0,#0FFH ;256_00

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOOP_BUFF_INIT:
MOV          @R0,#00          ; .
DJNZ        R0,LOOP_BUFF_INIT ;001_00
MOV          TMOD,#00000001B ;Timer0 16 bit

MAIN:
ACALL       INIT_LCD

MOV         LCD_ADDR,#000H
ACALL      SET_ADDR_LCD
MOV        DPTR,#DSPL_SELFUNC
ACALL      WRLINE_LCD
MOV        LCD_ADDR,#040H
ACALL      SET_ADDR_LCD
MOV        DPTR,#DSPL_SELFUNC2
ACALL      WRLINE_LCD

FIRST_FUNC:
MOV        R1,#01H
ACALL      CHK_KEY
CJNE      A,#00H,NEXT_FUNC
CLR        FUNC
SETB      SEL_SEND
SJMP      RX_MSG

NEXT_FUNC:
MOV        R1,#02H
ACALL      CHK_KEY
CJNE      A,#00H,FIRST_FUNC
SETB      FUNC

SELECT2:
MOV        LCD_ADDR,#000H
ACALL      SET_ADDR_LCD
MOV        DPTR,#DSPL_TEST
ACALL      WRLINE_LCD
MOV        LCD_ADDR,#040H
ACALL      SET_ADDR_LCD
MOV        DPTR,#DSPL_SELECT
ACALL      WRLINE_LCD

SEL_4:
MOV        R1,#01H
ACALL      CHK_KEY
CJNE      A,#00H,SEL_5
CLR        SEL_SEND
AJMP      AIR

SEL_5:
MOV        R1,#02H
ACALL      CHK_KEY
CJNE      A,#00H,SEL_11
SETB      SEL_SEND
AJMP      SEL_POS

SEL_11:
MOV        R1,#0AH
ACALL      CHK_KEY
CJNE      A,#00H,SEL_4
SJMP      BUFF_INIT

RX_MSG:
MOV        LCD_ADDR,#000H
ACALL      SET_ADDR_LCD
MOV        DPTR,#DSPL_PRESS
ACALL      WRLINE_LCD
MOV        LCD_ADDR,#040H
ACALL      SET_ADDR_LCD
MOV        DPTR,#DSPL_PRESS2
ACALL      WRLINE_LCD

```

```

                JB          RX_IR_SIGNAL,$
                ACALL       SAVE_IR_SIGNAL

TEST_MSG:      MOV          LCD_ADDR,#000H
                ACALL       SET_ADDR_LCD
                MOV         DPTR,#DSPL_TEST
                ACALL       WRLINE_LCD
                MOV         LCD_ADDR,#040H
                ACALL       SET_ADDR_LCD
                MOV         DPTR,#DSPL_TEST2
                ACALL       WRLINE_LCD

TEST:          MOV          R1,#01H
                ACALL       CHK_KEY
                CJNE        A,#00H,TEST2
                INC         COUNT_DATA

                AJMP        TEST_IR_SIGNAL
TEST2:         MOV          R1,#02H
                ACALL       CHK_KEY
                CJNE        A,#00H,SEL_12
                SJMP        SELECT
SEL_12:        MOV          R1,#0AH
                ACALL       CHK_KEY
                CJNE        A,#00H,TEST
                AJMP        BUFF_INIT

SELECT:        MOV          LCD_ADDR,#000H
                ACALL       SET_ADDR_LCD
                MOV         DPTR,#DSPL_TEST
                ACALL       WRLINE_LCD
                MOV         LCD_ADDR,#040H
                ACALL       SET_ADDR_LCD
                MOV         DPTR,#DSPL_SELECT
                ACALL       WRLINE_LCD

SEL_1:         MOV          R1,#01H
                ACALL       CHK_KEY
                CJNE        A,#00H,SEL_2
                AJMP        AIR
SEL_2:         MOV          R1,#02H
                ACALL       CHK_KEY
                CJNE        A,#00H,SEL_13
                AJMP        SEL_POS
SEL_13:        MOV          R1,#0AH
                ACALL       CHK_KEY
                CJNE        A,#00H,SEL_1
                SJMP        TEST_MSG

```

```

;-----
; Subroutine for Set EEPROM Address
;-----

```

```

AIR:          MOV          LCD_ADDR,#000H
                ACALL       SET_ADDR_LCD
                MOV         DPTR,#DSPL_AIR
                ACALL       WRLINE_LCD
                MOV         LCD_ADDR,#040H
                ACALL       SET_ADDR_LCD
                MOV         DPTR,#DSPL_AIR2
                ACALL       WRLINE_LCD

```

```

SEL_3:      MOV      R1,#01H
            ACALL   CHK_KEY
            CJNE   A,#00H,SEL_6
            CLR    TEM_SPA
            SJMP   TEMP

SEL_6:      MOV      R1,#02H
            ACALL   CHK_KEY
            CJNE   A,#00H,SEL_14
            AJMP   POWER

SEL_14:     MOV      R1,#03H
            ACALL   CHK_KEY
            CJNE   A,#00H,SEL_15
            SETB   TEM_SPA
            AJMP   SPARE

SEL_15:     MOV      R1,#0AH
            ACALL   CHK_KEY
            CJNE   A,#00H,SEL_3
            SJMP   SELECT

TEMP:       MOV      LCD_ADDR,#000H
            ACALL   SET_ADDR_LCD
            MOV     DPTR,#DSPL_TEMP
            ACALL   WRLINE_LCD
            MOV     LCD_ADDR,#040H
            ACALL   SET_ADDR_LCD
            MOV     DPTR,#DSPL_TEMP2
            ACALL   WRLINE_LCD

            MOV     LCD_ADDR,#0CH
            ACALL   SET_ADDR_LCD
            ACALL   LCD_BLINK
            CLR     DIGIT_CHK

TEMP_0:     MOV      R1,#0BH
            ACALL   CHK_KEY
            CJNE   A,#00H,TEMP_1
            MOV     R1,#00H
            SJMP   SHOW_NUM

TEMP_1:     MOV      R1,#01H
            ACALL   CHK_KEY
            CJNE   A,#00H,TEMP_2
            SJMP   SHOW_NUM

TEMP_2:     MOV      R1,#02H
            ACALL   CHK_KEY
            CJNE   A,#00H,TEMP_3
            SJMP   SHOW_NUM

TEMP_3:     MOV      R1,#03H
            ACALL   CHK_KEY
            CJNE   A,#00H,TEMP_4
            SJMP   SHOW_NUM

TEMP_4:     MOV      R1,#04H
            ACALL   CHK_KEY
            CJNE   A,#00H,TEMP_5
            SJMP   SHOW_NUM

TEMP_5:     MOV      R1,#05H
            ACALL   CHK_KEY
            CJNE   A,#00H,TEMP_6
            SJMP   SHOW_NUM

TEMP_6:     MOV      R1,#06H
            ACALL   CHK_KEY

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งาน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TEMP_7:      CJNE      A, #00H, TEMP_7
              SJMP      SHOW_NUM
              MOV       R1, #07H
              ACALL     CHK_KEY
              CJNE      A, #00H, TEMP_8
TEMP_8:      SJMP      SHOW_NUM
              MOV       R1, #08H
              ACALL     CHK_KEY
              CJNE      A, #00H, TEMP_9
TEMP_9:      SJMP      SHOW_NUM
              MOV       R1, #09H
              ACALL     CHK_KEY
              CJNE      A, #00H, SEL_16
SEL_16:      SJMP      SHOW_NUM
              MOV       R1, #0AH
              ACALL     CHK_KEY
              CJNE      A, #00H, TEMP_0
              AJMP      AIR

SHOW_NUM:    MOV       A, R1
              MOV       DPTR, #NUM
              MOVC      A, @A+DPTR
              MOV       LCD_DATA, A
              ACALL     WRCHAR_LCD
              JB        DIGIT_CHK, D_2
              MOV       DIGIT_1, R1
              SETB      DIGIT_CHK
              ACALL     DELAY_100ms
              ACALL     DELAY_100ms
D_2:         SJMP      TEMP_0
              MOV       DIGIT_2, R1

              MOV       A, DIGIT_1
              SWAP      A
              ADD       A, DIGIT_2
              MOV       DIGIT_2, A
              MOV       A, DIGIT_1
              MOV       B, #006H
              MUL       AB
              MOV       SUB, A
              MOV       A, DIGIT_2
              CLR       C
              SUBB      A, SUB
              JB        TEM_SPA, SEL_22

              CLR       C
              SUBB      A, #00FH
              ADD       A, #0A0H
              MOV       SUB, A
              ANL       A, #0F0H
              CJNE      A, #0A0H, INVALID
              MOV       EEP_AH, SUB
              MOV       EEP_AL, #000H
              AJMP      TEMP_SAVE
SEL_22:      AJMP      SPARE2

INVALID:     MOV       LCD_ADDR, #000H
              ACALL     SET_ADDR_LCD
              MOV       DPTR, #DSPL_INVA
              ACALL     WRLINE_LCD
              MOV       LCD_ADDR, #040H

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ACALL    SET_ADDR_LCD
MOV      DPTR,#DSPL_INVA2
ACALL    WRLINE_LCD

ACALL    DELAY_1s
ACALL    DELAY_1s
JB       TEM_SPA, SPARE
AJMP     TEMP

POWER:   MOV      LCD_ADDR,#000H
ACALL    SET_ADDR_LCD
MOV      DPTR,#DSPL_POW
ACALL    WRLINE_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#DSPL_POW2
ACALL    WRLINE_LCD

SEL_17:  MOV      R1,#01H
ACALL    CHK_KEY
CJNE    A,#00H,SEL_18
MOV      EEP_AH,#0F0H
MOV      EEP_AL,#000H
SJMP    TEMP_SAVE

SEL_18:  MOV      R1,#02H
ACALL    CHK_KEY
CJNE    A,#00H,SEL_19
MOV      EEP_AH,#0F1H
MOV      EEP_AL,#000H
SJMP    TEMP_SAVE

SEL_19:  MOV      R1,#0AH
ACALL    CHK_KEY
CJNE    A,#00H,SEL_17
AJMP     AIR

SPARE:   MOV      LCD_ADDR,#000H
ACALL    SET_ADDR_LCD
MOV      DPTR,#DSPL_SPA
ACALL    WRLINE_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#DSPL_SPA2
ACALL    WRLINE_LCD

MOV      LCD_ADDR,#09H
ACALL    SET_ADDR_LCD
ACALL    LCD_BLINK
CLR      DIGIT_CHK
AJMP     TEMP_0

SPARE2:  ADD      A,#0B0H
MOV      SUB,A
ANL     A,#0F0H
CJNE    A,#0B0H, SPARE3
SJMP    SPARE6

SPARE3:  CJNE    A,#0C0H, SPARE4
SJMP    SPARE6

SPARE4:  CJNE    A,#0D0H, SPARE5
SJMP    SPARE6

SPARE5:  CJNE    A,#0E0H, SEL_23
SPARE6:  MOV      EEP_AH, SUB
MOV      EEP_AL,#000H

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEL_23:          SJMP      TEMP_SAVE
                 AJMP      INVALID

TEMP_SAVE:      JB        FUNC,TEMP_SEND
                 MOV       LCD_ADDR,#040H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_SAVE
                 ACALL    WRLINE_LCD

                 ACALL    AIR_SAVE
                 AJMP      CONTINUE

TEMP_SEND:      MOV       LCD_ADDR,#040H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_LOAD
                 ACALL    WRLINE_LCD
                 MOV       R3,#007H
                 AJMP      FUNC_SEND

```

```

;-----
SEL_POS:        MOV       LCD_ADDR,#000H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_SELPOS
                 ACALL    WRLINE_LCD
                 MOV       LCD_ADDR,#040H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_SELPOS2
                 ACALL    WRLINE_LCD
                 ACALL    POSITION_0

SEL_BUTTON:    MOV       LCD_ADDR,#000H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_SELBUT
                 ACALL    WRLINE_LCD
                 MOV       LCD_ADDR,#040H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_SELBUT2
                 ACALL    WRLINE_LCD

SEL_BUTTON2:   ACALL    BUTTON_1
                 JB        FUNC,FUNC_SEND
                 ACALL    SAVE_TO_EEP
                 SJMP      CONTINUE

```

```

CONTINUE:      MOV       LCD_ADDR,#000H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_CON
                 ACALL    WRLINE_LCD
                 MOV       LCD_ADDR,#040H
                 ACALL    SET_ADDR_LCD
                 MOV       DPTR,#DSPL_CON2
                 ACALL    WRLINE_LCD

```

```

SAVE_NEXT:     MOV       R1,#01H
                 ACALL    CHK_KEY
                 CJNE    A,#00H,BACK_TO_MAIN
                 CLR      FUNC
                 AJMP    RX_MSG

```

```

BACK_TO_MAIN:    MOV        R1,#02H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,SAVE_NEXT
                 AJMP     MAIN

FUNC_SEND:      ACALL     EEPROM_RD
                 MOV      COUNT_DATA,EEP_DATA
                 INC      COUNT_DATA
                 MOV      R0,#30H

LOOP_SEND:      INC      EEP_AL
                 ACALL     EEPROM_RD
                 MOV      @R0,EEP_DATA
                 INC      R0
                 MOV      A,R0
                 CJNE     A,COUNT_DATA,LOOP_SEND

                 AJMP     TEST_IR_SIGNAL

```

```

;-----
; Subroutine for Set EEPROM Address
;-----

```

```

BUTTON_1:      MOV      R1,#01H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_2
                 SJMP     SET_ADDR

BUTTON_2:      MOV      R1,#02H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_3
                 SJMP     SET_ADDR

BUTTON_3:      MOV      R1,#03H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_4
                 SJMP     SET_ADDR

BUTTON_4:      MOV      R1,#04H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_5
                 SJMP     SET_ADDR

BUTTON_5:      MOV      R1,#05H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_6
                 SJMP     SET_ADDR

BUTTON_6:      MOV      R1,#06H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_7
                 SJMP     SET_ADDR

BUTTON_7:      MOV      R1,#07H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_8
                 SJMP     SET_ADDR

BUTTON_8:      MOV      R1,#08H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_9
                 SJMP     SET_ADDR

BUTTON_9:      MOV      R1,#09H
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_0
                 SJMP     SET_ADDR

BUTTON_0:      MOV      R1,#0BH
                 ACALL     CHK_KEY
                 CJNE     A,#00H,BUTTON_10
                 MOV      R1,#00H

```

```

BUTTON_10:    SJMP      SET_ADDR
              MOV       R1,#0AH
              ACALL     CHK_KEY
              CJNE      A,#00H,BUTTON_1
              AJMP      SEL_POS

```

```

SET_ADDR:     MOV       REPEAT,R1
              MOV       A,EED_AH
              ANL       A,#0F0H
              ADD       A,R1
              MOV       EED_AH,A
              MOV       EED_AL,#000H
              RET

```

```

;-----
;Subroutine for Select Position Number of Remote Control
;-----

```

```

POSITION_0:   MOV       R1,#0BH
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_1
              MOV       R1,#00H
              ACALL     SET_POS
              RET

```

```

POSITION_1:   MOV       R1,#01H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_2
              ACALL     SET_POS
              RET

```

```

POSITION_2:   MOV       R1,#02H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_3
              ACALL     SET_POS
              RET

```

```

POSITION_3:   MOV       R1,#03H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_4
              ACALL     SET_POS
              RET

```

```

POSITION_4:   MOV       R1,#04H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_5
              ACALL     SET_POS
              RET

```

```

POSITION_5:   MOV       R1,#05H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_6
              ACALL     SET_POS
              RET

```

```

POSITION_6:   MOV       R1,#06H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_7
              ACALL     SET_POS
              RET

```

```

POSITION_7:   MOV       R1,#07H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_8
              ACALL     SET_POS
              RET

```

```

POSITION_8:   MOV       R1,#08H
              ACALL     CHK_KEY
              CJNE      A,#00H,POSITION_9
              ACALL     SET_POS

```

```

POSITION_9:    RET
               MOV     R1,#09H
               ACALL   CHK_KEY
               CJNE    A,#00H,SEL_20
               ACALL   SET_POS
               RET
SEL_20:        MOV     R1,#0AH
               ACALL   CHK_KEY
               CJNE    A,#00H,POSITION_0
               JB     FUNC,SEL_21
               AJMP    SELECT
SEL_21:        AJMP    SELECT2
SET_POS:       MOV     A,R1
               SWAP   A
               MOV    EEP_AH,A
               RET

```

```

;-----
; Subroutine for Save Infrared Signal to EEPROM
;-----

```

```

SAVE_TO_EEP:   MOV     LCD_ADDR,#000H
               ACALL   SET_ADDR_LCD
               MOV     DPTR,#DSPL_SAVING
               ACALL   WRLINE_LCD
               MOV     LCD_ADDR,#040H
               ACALL   SET_ADDR_LCD
               MOV     DPTR,#DSPL_SAVING2
               ACALL   WRLINE_LCD
AIR_SAVE:      MOV     EEP_DATA,COUNT_DATA
               ACALL   EEPROM_WR
               MOV     R5,#05H
               ACALL   DELAY_VARIE
               MOV     R0,#30H
               INC     COUNT_DATA
LOOP_SAVE:     INC     EEP_AL
               MOV     EEP_DATA,@R0
               ACALL   EEPROM_WR
               MOV     R5,#05H
               ACALL   DELAY_VARIE
               INC     R0
               MOV     A,R0
               CJNE    A,COUNT_DATA,LOOP_SAVE
               DEC     COUNT_DATA
               RET

```

```

;-----
; I2C EEPROM Read
;-----

```

```

EEPROM_RD:     MOV     I2C_ADDR,#EEPROM_ID
               ACALL   I2C_SLAVE
               JB     I2C_ACK,EEPROM_RD_EXIT
               MOV     I2C_DATA,EEP_AH
               ACALL   I2C_DATA_WR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน I2C ที่ถูกบันทึกไว้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                ACALL      I2C_DATA_WR

                MOV        I2C_ADDR,#EEPROM_ID+1
                ACALL      I2C_SLAVE

                ACALL      I2C_DATA_RD
                MOV        EEP_DATA,I2C_DATA
                ACALL      I2C_NACK_BIT

                ACALL      I2C_STOP
EEPROM_RD_EXIT: RET

```

```

;-----
; I2C EEPROM Write
;-----

```

```

EEPROM_WR:      MOV        I2C_ADDR,#EEPROM_ID
                ACALL      I2C_SLAVE
                JB         I2C_ACK,EEPROM_WR_EXIT

                MOV        I2C_DATA,EEP_AH
                ACALL      I2C_DATA_WR

                MOV        I2C_DATA,EEP_AL
                ACALL      I2C_DATA_WR

                MOV        I2C_DATA,EEP_DATA
                ACALL      I2C_DATA_WR

                ACALL      I2C_STOP
EEPROM_WR_EXIT: RET

```

```

;-----
; Subroutine for Test Infrared Signal
;-----

```

```

TEST_IR_SIGNAL: MOV        R0,#30H
TEST_LOOP:      MOV        BUFF,@R0
                SETB      TX_IR_SIGNAL
                ACALL      IR_DELAY
                INC        R0
                MOV        A,R0
                CJNE      A,COUNT_DATA,TEST_LOOP2
                SJMP      END_TEST

TEST_LOOP2:     MOV        BUFF,@R0
                CLR        TX_IR_SIGNAL
                ACALL      IR_DELAY
                INC        R0
                MOV        A,R0
                CJNE      A,COUNT_DATA,TEST_LOOP
                SJMP      END_TEST

END_TEST:      DEC        COUNT_DATA
                JB         SEL_SEND,END_TEST4
                DJNZ      R3,TEST_IR_SIGNAL
                AJMP      AIR

END_TEST4:     JB         FUNC,END_TEST2
                AJMP      TEST

END_TEST2:     MOV        R1,REPEAT
                ACALL      CHK_KEY
                CJNE      A,#00H,END_TEST3
                INC        COUNT_DATA

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                SJMP          TEST_IR SIGNAL
END_TEST3:      AJMP          SEL_BUTTON2

;-----
; Subroutine for Save Infrared Signal to RAM
;-----
SAVE_IR_SIGNAL: MOV          R0,#30H
LOW_PULSE:     ACALL        INIT_TIMER
                JNB          RX_IR_SIGNAL,$
                CLR          TR0
                ACALL        SAVE_TO_RAM

                ACALL        INIT_TIMER
END_CHECK:     JB           RX_IR_SIGNAL,END_CHECK2
                CLR          TR0
                ACALL        SAVE_TO_RAM
                SJMP        LOW_PULSE

END_CHECK2:    JNB          TF0,END_CHECK
                SJMP        COMPLETE

COMPLETE:      MOV          COUNT_DATA,R0
                AJMP        TEST_MSG

SAVE_TO_RAM:   MOV          A,TH0          ;error no more than 0.25mSec
                MOV          @R0,A
                INC          R0
                RET

INIT_TIMER:    CLR          TR0
                CLR          TF0
                MOV          TH0,#000H
                MOV          TL0,#000H
                SETB        TR0
                RET

;-----
; I2C Data Write
;-----
I2C_DATA_WR:   PUSH        ACC
                SETB        I2C_ACK
                MOV          A,I2C_DATA
                MOV          R5,#008          ; Set loop 8 times
I2C_DATA_WR_1: RLC          A
                MOV          SDA,C
                ACALL        I2C_CLK
                DJNZ        R5,I2C_DATA_WR_1
                SETB        SDA
                ACALL        I2C_DELAY
                SETB        SCL
                ACALL        I2C_DELAY
                JB           SDA,I2C_DATA_WR_2
                CLR          I2C_ACK
I2C_DATA_WR_2: CLR          SCL
                POP          ACC
                RET

```

```

;-----
; I2C Data Read
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้วยการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```

I2C_DATA_RD:    PUSH    ACC
                CLR     A
                MOV     R5,#008           ; Set loop 8 times
I2C_DATA_RD_1:  ACALL   I2C_DELAY
                SETB    SCL
                ACALL   I2C_DELAY
                MOV     C,SDA
                RLC     A
                CLR     SCL
                DJNZ   R5,I2C_DATA_RD_1
                MOV     I2C_DATA,A
                POP     ACC
                RET

```

```

;-----
; I2C Slave Connect
;-----

```

```

I2C_SLAVE:     PUSH    ACC
                SETB    I2C_ACK
                MOV     A,I2C_ADDR
                ACALL   I2C_START
I2C_SLAVE_1:   MOV     R5,#008           ; Set loop 8 times
                RLC     A
                MOV     SDA,C
                ACALL   I2C_CLK
                DJNZ   R5,I2C_SLAVE_1

                SETB    SDA
                ACALL   I2C_DELAY
                SETB    SCL
                ACALL   I2C_DELAY
                JB      SDA,I2C_SLAVE_2
I2C_SLAVE_2:   CLR     I2C_ACK
                CLR     SCL
                POP     ACC
                RET

```

```

;-----
; I2C Condition
;-----

```

```

I2C_START:     JNB     SCL,I2C_START_1   ; Check current SCL set?
                CLR     SCL

I2C_START_1:   SETB    SDA
                SETB    SCL

                ACALL   I2C_DELAY
                CLR     SDA           ; Clear SDA during SCL set
                ACALL   I2C_DELAY
                CLR     SCL
                RET

I2C_STOP:      JNB     SCL,I2C_STOP_1   ; Check current SCL set?
                CLR     SCL

I2C_STOP_1:   CLR     SDA
                ACALL   I2C_DELAY
                SETB    SCL
                ACALL   I2C_DELAY
                SETB    SDA           ; Set SDA during SCL set
                RET

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

I2C_CLK:          ACALL      I2C_DELAY   ; Pulse SCL
                  SETB       SCL
                  ACALL      I2C_DELAY
                  CLR        SCL
                  RET

```

```

I2C_ACK_BIT:     CLR        SDA
                  ACALL      I2C_DELAY
                  ACALL      I2C_CLK
                  SETB       SDA
                  RET

```

```

I2C_NACK_BIT:    SETB       SDA
                  ACALL      I2C_DELAY
                  ACALL      I2C_CLK
                  SETB       SCL
                  RET

```

```

;-----
; Compare for Check Key
;-----

```

```

CHK_KEY:         ACALL      GET_KPAD
                  MOV        A, KPAD_DATA
                  XRL        A, R1
                  RET

```

```

;-----
; Keypad Scan key Subroutine
;-----

```

```

GET_KPAD:        MOV        P2, #0FFH
                  MOV        KPAD_DATA, #0

```

```

CHK_COLO:       CLR        KPAD_COLO
                  MOV        A, P2
                  ANL        A, #00FH
                  CJNE       A, #00FH, COLO_DETECT
                  AJMP       CHK_COL1

```

```

COLO_DETECT:    MOV        KPAD_DATA, #01
                  AJMP       GET_ROW

```

```

CHK_COL1:       SETB       KPAD_COLO
                  CLR        KPAD_COL1
                  MOV        A, P2
                  ANL        A, #00FH
                  CJNE       A, #00FH, COL1_DETECT
                  AJMP       CHK_COL2

```

```

COL1_DETECT:    MOV        KPAD_DATA, #02
                  AJMP       GET_ROW

```

```

CHK_COL2:       SETB       KPAD_COL1
                  CLR        KPAD_COL2
                  MOV        A, P2
                  ANL        A, #00FH
                  CJNE       A, #00FH, COL2_DETECT
                  RET

```

```

COL2_DETECT:    MOV        KPAD_DATA, #03

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GET_ROW:      CLR      KPAD_COL0
              CLR      KPAD_COL1
              CLR      KPAD_COL2

              JB      KPAD_ROW0,CHK_ROW1
              RET

CHK_ROW1:    JB      KPAD_ROW1,CHK_ROW2
              MOV      A,KPAD_DATA
              ADD      A,#3
              MOV      KPAD_DATA,A
              RET

CHK_ROW2:    JB      KPAD_ROW2,CHK_ROW3
              MOV      A,KPAD_DATA
              ADD      A,#6
              MOV      KPAD_DATA,A
              RET

CHK_ROW3:    MOV      A,KPAD_DATA
              ADD      A,#9
              MOV      KPAD_DATA,A
              RET

;-----
; Define LCD's command constant
;-----
LCD_CMD_CLR EQU 0000001B
LCD_CMD_HOME EQU 0000010B
LCD_CMD_AUTOINC EQU 0000110B
LCD_CMD_OFF EQU 00001000B
LCD_CMD_ON EQU 00001100B
LCD_CMD_BLINK EQU 00001111B

INIT_LCD:    ACALL   DELAY_100ms

              CLR      LCD_RS
              MOV      P1,#00111000B
              ACALL   LCD_CLK
              ACALL   DELAY_10ms

              MOV      P1,#00111000B      ; Force 8bit mode again
              ACALL   LCD_CLK

              ACALL   LCD_OFF      ; Display off
              ACALL   LCD_AUTOINC
              ACALL   LCD_HOME      ; Return cursor to home
              ACALL   LCD_CLR      ; Display cleared
              ACALL   LCD_ON      ; Display on
              RET

LCD_CLR:     MOV      LCD_DATA,#LCD_CMD_CLR
              AJMP    WRCMD_LCD

LCD_HOME:    MOV      LCD_DATA,#LCD_CMD_HOME
              AJMP    WRCMD_LCD

LCD_AUTOINC: MOV      LCD_DATA,#LCD_CMD_AUTOINC
              AJMP    WRCMD_LCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCD_OFF:      MOV      LCD_DATA,#LCD_CMD_OFF
              AJMP     WRCMD_LCD
```

```
LCD_ON:       MOV      LCD_DATA,#LCD_CMD_ON
              AJMP     WRCMD_LCD
```

```
LCD_BLINK:    MOV      LCD_DATA,#LCD_CMD_BLINK
              AJMP     WRCMD_LCD
```

```
-----
; Set LCD Address
-----
```

```
SET_ADDR_LCD: CLR      LCD_RS
              MOV      P1,LCD_ADDR
              ORL      P1,#10000000B ; Set MSB of P1
              ACALL   LCD_CLK
              RET
```

```
-----
; LCD Write Command
-----
```

```
WRCMD_LCD:   CLR      LCD_RS
              MOV      P1,LCD_DATA
              ACALL   LCD_CLK
              RET
```

```
-----
; Write Character to show LCD
-----
```

```
WRCHAR_LCD:  SETB     LCD_RS
              MOV      P1,LCD_DATA
              ACALL   LCD_CLK
              RET
```

```
-----
; Write Line of 20 Character from ROM
-----
```

```
WRLINE_LCD:  MOV      R0,#0 ; Clear loop counter
WRLINE_LCD_1: SETB     LCD_RS
              CLR      A
              MOVC    A,@A+DPTR ; Move data from @DPTR to ACC.
              MOV     P1,A
              ACALL   LCD_CLK
              INC     DPTR ; Increase Pointer
              INC     R0
              CJNE    R0,#16,WRLINE_LCD_1 ; Do until 16 times
              RET
```

```
-----
; LCD Clk
-----
```

```
LCD_CLK:     SETB     LCD_EN ; Pulse Clock to LCD_EN
              ACALL   LCD_DELAY
              CLR     LCD_EN
              ACALL   LCD_DELAY
              RET
```

```
-----
; Sub DELAY TIME AT 12MHz
-----
```

```
I2C_DELAY:  MOV      R6,#014H ; Each loop = 50 us
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

I2C_DELAY_1:      NOP
                  NOP
                  NOP
                  DJNZ      R6,I2C_DELAY_1
                  RET

IR_DELAY:         MOV        R6,#133          ; Each loop = 220 us
LOOP_IR_DELAY:    DJNZ      R6,LOOP_IR_DELAY
                  DJNZ      BUFF,IR_DELAY
                  RET

DELAY_VARIE:      ACALL     DELAY_1ms
                  DJNZ      R5,DELAY_VARIE    ;Varie By R5
                  RET

LCD_DELAY:        MOV        R7,#002
LCD_DELAY_1:      ACALL     DELAY_1ms
                  DJNZ      R7,LCD_DELAY_1
                  RET

DELAY_1ms:        MOV        R4,#002
DELAY_1ms_1:      MOV        R6,#0FAH
DELAY_1ms_2:      NOP
                  NOP
                  DJNZ      R6,DELAY_1ms_2
                  DJNZ      R4,DELAY_1ms_1
                  RET

DELAY_10ms:       MOV        R7,#010
DELAY_10ms_1:     ACALL     DELAY_1ms
                  DJNZ      R7,DELAY_10ms_1
                  RET

DELAY_100ms:      MOV        R7,#100
DELAY_100ms_1:    ACALL     DELAY_1ms
                  DJNZ      R7,DELAY_100ms_1
                  RET

DELAY_1s:         MOV        R7,#004
DELAY_1s_1:       MOV        R3,#250
DELAY_1s_2:       ACALL     DELAY_1ms
                  DJNZ      R3,DELAY_1s_2
                  DJNZ      R7,DELAY_1s_1
                  RET

```

```

;-----
;Define Constant < Store in Flash EEPROM Program Memory >
;-----

```

```

DSPL_SELFUNC:     DB          'SELECT FUNCTION '
DSPL_SELFUNC2:    DB          'COPY(1) SEND(2) '
DSPL_SELPOS:      DB          'SELECT POSITION '
DSPL_SELPOS2:     DB          ' (POSITION 0-9) '

DSPL_SELBUT:      DB          ' SELECT BUTTON '
DSPL_SELBUT2:     DB          ' (BUTTON 0-9) '

DSPL_PRESS:       DB          'PRESS REMOTE KEY'
DSPL_PRESS2:      DB          'YOU WANT TO COPY'
DSPL_BLANK:       DB          ' '
DSPL_TEST:        DB          'PLEASE SELECT '
DSPL_TEST2:       DB          'TEST(1) SAVE(2) '
DSPL_SAVING:      DB          'SAVING'

```

DSPL\_SAVING: การที่สงวนไว้สำหรับการใช้งานเพื่อ NOW SAVING นั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DSPL_SAVING2:    DB          '  PLEASE WAIT  '
DSPL_CON:        DB          '  SAVE NEXT(1)  '
DSPL_CON2:       DB          'BACK TO MAIN(2)  '
;-----
NUM:             DB          '0123456789'
DSPL_SELECT:     DB          'AIR(1)  OTHER(2) '
DSPL_AIR:        DB          'TEMP(1) POWER(2) '
DSPL_AIR2:       DB          'SPARE(3)      '
DSPL_TEMP:       DB          'TEMPERATURE:  '
DSPL_TEMP2:      DB          '(MIN=15,MAX=30) '
DSPL_INVA:       DB          'INVALID NUMBER '
DSPL_INVA2:      DB          'PLEASE TRY AGAIN'
DSPL_SAVE:       DB          '  SAVING      '
DSPL_LOAD:       DB          '  LOADING     '
DSPL_POW:        DB          '  POWER       '
DSPL_POW2:       DB          'ON(1)  OFF(2) '
DSPL_SPA:        DB          'POSITION:    '
DSPL_SPA2:       DB          '(SPARE 0-63)  '
END

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# คู่มือการใช้งานเครื่องคัดลอกรีโมทคอนโทรลหลายหน้าที่

## ตอนที่ 1 การคัดลอกสัญญาณ

1. เมื่อจ่ายไฟให้กับชิ้นงาน จอแอลซีดีจะแสดงข้อความให้เลือกที่จะคัดลอกสัญญาณหรือใช้งานส่งสัญญาณ ให้เลือกคัดลอกสัญญาณแล้วนำรีโมทคอนโทรลต้นแบบมาส่งสัญญาณเข้าที่ส่วนรับสัญญาณของชิ้นงาน



จอแอลซีดีแสดงข้อความให้เลือกที่จะคัดลอกสัญญาณหรือใช้งานส่งสัญญาณ



จอแอลซีดีแสดงข้อความรอรับสัญญาณจากรีโมทคอนโทรล

2. เมื่อรับสัญญาณมาแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกที่จะทดสอบสัญญาณหรือบันทึกสัญญาณ ให้ลองทดสอบดูก่อนว่ารับสัญญาณมาถูกต้องหรือไม่ ถ้ารับมาถูกต้องก็เลือกบันทึกสัญญาณได้เลย



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.เมื่อเลือกบันทึกสัญญาณแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกในโหมดเครื่องปรับอากาศ หรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ ถ้าเลือกโหมดเครื่องปรับอากาศจอแอลซีดีจะแสดงข้อความให้เลือกส่วนเก็บบันทึกข้อมูล 3 ส่วน คือ ส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิ ส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ และส่วนเก็บข้อมูลสำรองสำหรับปุ่มอื่นๆ ที่เหลือ



จอแอลซีดีแสดงข้อความให้เลือกว่าจะบันทึกใน โหมดเครื่องปรับอากาศหรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ

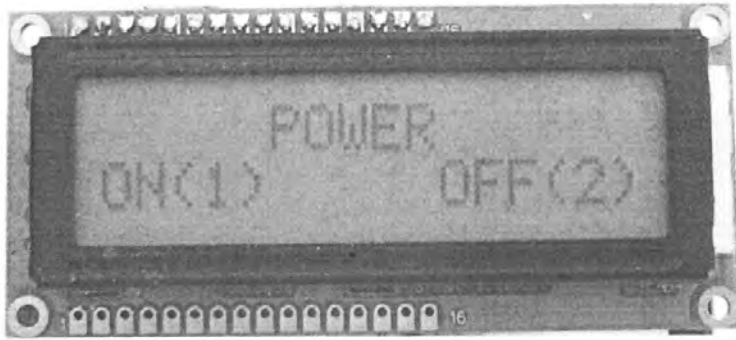


จอแอลซีดีจะแสดงข้อความให้เลือกส่วนเก็บบันทึกข้อมูล



จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิ สามารถบันทึกได้ตั้งแต่อุณหภูมิ 15-30 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

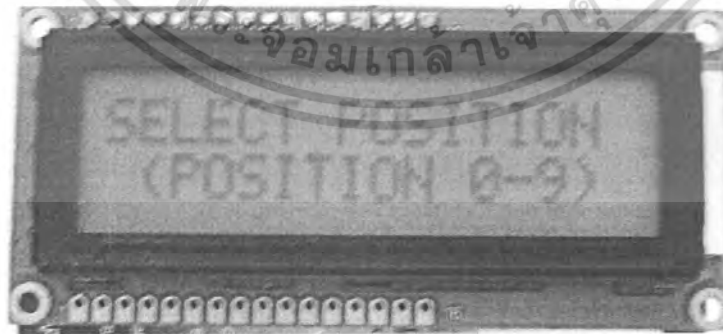


จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ



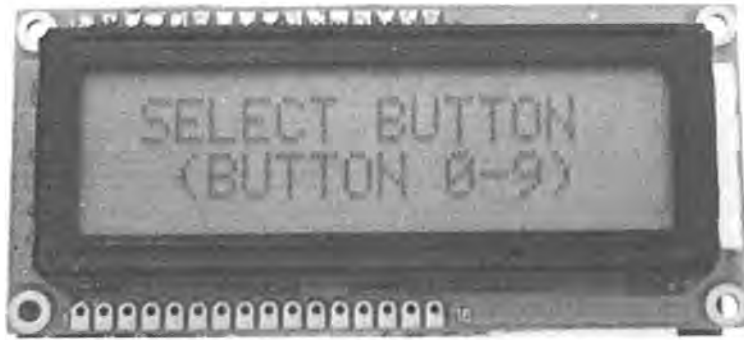
จอแอลซีดีแสดงข้อความส่วนเก็บข้อมูลสำรองสำหรับปุ่มอื่นๆ ที่เหลือ

4. ถ้าเลือกโหมดเครื่องใช้ไฟฟ้าชนิดอื่นๆ จอแอลซีดีจะแสดงข้อความให้เลือกตำแหน่งที่จะบันทึก และเลือกปุ่มที่จะบันทึก



จอแอลซีดีจะแสดงข้อความให้เลือกตำแหน่งที่จะบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จอแอลซีดีจะแสดงข้อความให้เลือกรุ่นที่จะบันทึก

5.เมื่อทำการบันทึกเสร็จแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกปุ่มต่อไป หรือกลับไปเมนูหลัก



จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะบันทึกปุ่มต่อไป หรือกลับไปเมนูหลัก

ทำตามขั้นตอนนี้บันทึกจนครบทุกปุ่มแล้วเลือกกลับเมนูหลักเพื่อไปทำตอนที่ 2 ต่อไป

## ตอนที่ 2 การส่งสัญญาณ

1. ในหน้าเมนูหลัก จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะกดลอคสัญญาณหรือใช้งานส่งสัญญาณให้เลือกลงสัญญาณ

2. เมื่อเลือกส่งสัญญาณแล้ว จอแอลซีดีจะแสดงข้อความให้เลือกว่าจะส่งสัญญาณในโหมดเครื่องปรับอากาศ หรือเครื่องใช้ไฟฟ้าชนิดอื่นๆ ถ้าเลือกโหมดเครื่องปรับอากาศจอแอลซีดีจะแสดงข้อความให้เลือกส่วนเก็บบันทึกข้อมูล 3 ส่วน คือ ส่วนเก็บข้อมูลสัญญาณเปลี่ยนอุณหภูมิ ส่วนเก็บข้อมูลสัญญาณเปิด-ปิดเครื่องปรับอากาศ และส่วนเก็บข้อมูลสำรองสำหรับปุ่มอื่นๆ ที่เหลือ ให้เลือกส่วนที่ต้องการแล้วจึงทำการส่งสัญญาณ

3. ถ้าต้องการส่งสัญญาณในโหมดเครื่องใช้ไฟฟ้าชนิดอื่น ให้เลือกตำแหน่งที่บันทึกรีโมทคอนโทรลที่ต้องการส่งไว้ และเลือกรุ่นที่จะส่งสัญญาณ เหมือนกับตอนที่บันทึก เมื่อกดปุ่มที่ต้องการแล้ว

ไมโครคอนโทรลเลอร์จะทำหน้าที่ดึงข้อมูลที่เก็บอยู่ในอีอีพรอมจากตำแหน่งที่ระบุโดยตัวแปรข้างต้น มาสร้างเอกสารเป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ทางการค้า เป็นลิขสิทธิ์แบบ แล้วส่งออกไปทันที

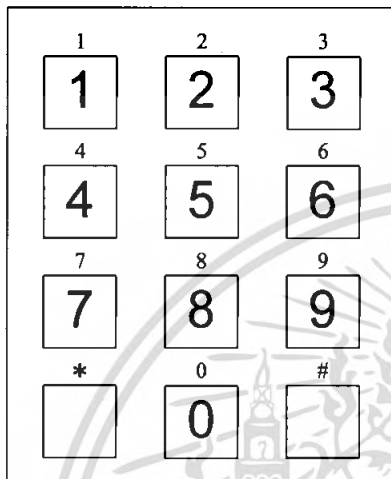
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ตอนที่ 3 แนะนำตำแหน่งสำหรับบันทึกสัญญาณ

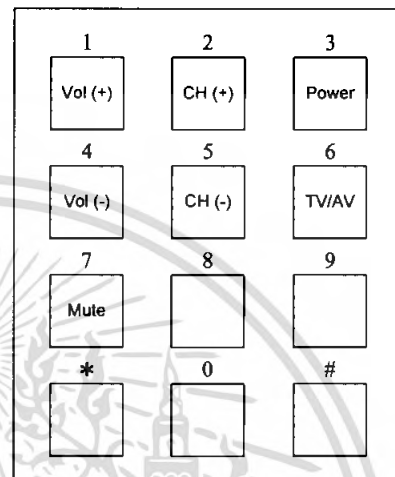
\*หมายเหตุ ตำแหน่งต่างๆ ที่จะนำเสนอต่อไปนี้เป็นเพียงตำแหน่งการบันทึกแนะนำสำหรับผู้เริ่มต้นใช้งานเท่านั้น ซึ่งงานนี้สามารถบันทึกสัญญาณใดก็ตามที่ตำแหน่งใดก็ได้แล้วแต่ความต้องการของผู้ใช้งาน

\* เมื่อกดปุ่ม “\*” จะเป็นการย้อนกลับไปหน้าที่แล้ว

#### POSITION 1 ~ 2 สำหรับบันทึกสัญญาณจากรีโมทคอนโทรลของโทรทัศน์ ดังนี้

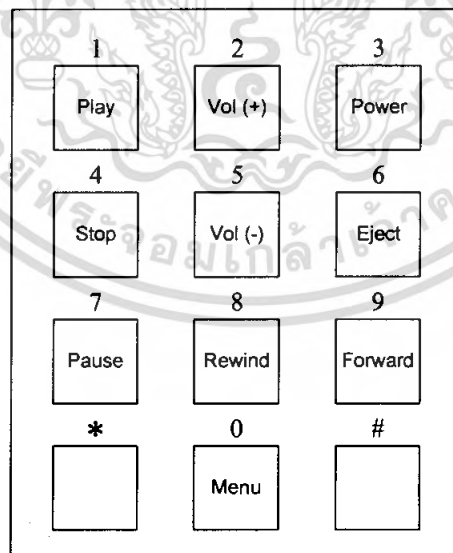


Position 1



Position 2

#### POSITION 3 สำหรับบันทึกสัญญาณจากรีโมทคอนโทรลของเครื่องเล่นวีซีดี ดังนี้



Position 3

หากต้องการบันทึกอุปกรณ์อื่นเพิ่ม ก็สามารถบันทึกลงในตำแหน่งอื่นๆ ที่เหลือได้ โดยใช้วิธีการตามที่กล่าวมาแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Phillips RC 5 infrared remote protocol

RC 5 เป็นโปรโตคอลโดยมาตรฐานโดยทั่วไปสำหรับการสื่อสารข้อมูลแบบอินฟราเรด ซึ่งถูกพัฒนาโดยฟิลิปส์ โคคชนิคนี้ประกอบด้วย instruction set 2048 แบบที่แตกต่างกัน และถูกแบ่งเป็น 32 address ในแต่ละ 64 instruction แต่ละอุปกรณ์จะใช้ address ของตัวมันเอง ซึ่งทำให้สามารถปรับ volume ของทีวีได้ โดยที่ volume ของ hifi ไม่เปลี่ยนแปลง

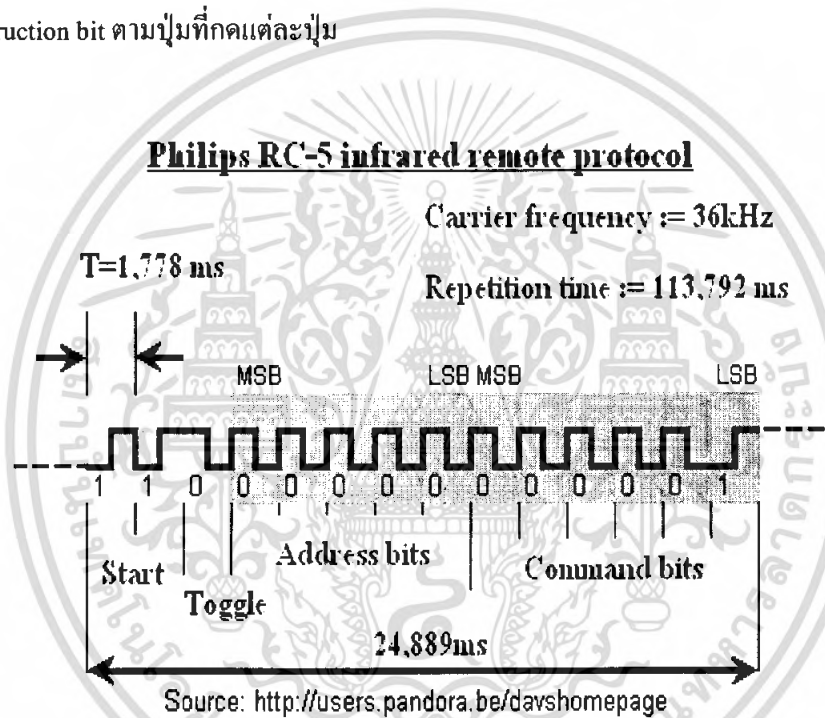
Transfer code เป็น dataword ที่มี 14 บิต ดังที่จะอธิบายต่อไปนี้

2 บิตแรก เป็น startbit สำหรับควบคุมเกนอัตโนมัติ ในเครื่องรับสัญญาณอินฟราเรด

1 บิตต่อไป เป็น toggle bit (จะเปลี่ยนทุกๆครั้งเมื่อมีการกดปุ่ม)

5 บิตต่อไป เป็น address ของระบบ

6 instruction bit ตามปุ่มที่กดแต่ละปุ่ม



RC 5 ใช้เทคนิคการ Bifase Modulation ซึ่งหมายความว่าทุกบิตที่จะประกอบไปด้วยสองส่วนจะไม่มีทางเหมือนกัน ดังนั้นในแต่ละบิตจะมีลักษณะ low/high หรือ high/low เสมอ ใน RC 5 code 1 จะเป็น low/high และ 0 จะเป็นแบบ high/low สำหรับบิตทั้งหมด บิตที่มีความสำคัญที่สุดจะถูกส่งออกไปก่อนเวลาในแต่ละบิตจะเท่ากับ 1778 ms และเวลาทั้งหมดของ RC 5 code ทั้งหมดคือ 24,778 ms. ช่องว่างระหว่างสอง transmitted bit code จะมีขนาด 50 เท่าของเวลาในแต่ละบิต หรือ 88,889 ms. และ RC 5 code มีความถี่พาห้ เป็น 36 kHz

## Daewoo infrared remote protocol

ในโปรโตคอลของ Daewoo บิตของลอจิก 0 จะมีช่วงที่โวลเตจสูง โดยมีช่วงเวลา 550 us ตามด้วยช่วงโวลเตจต่ำ 450 us ซึ่งหมายความว่าเวลาทั้งหมดของลอจิก 0 คือ 1 ms

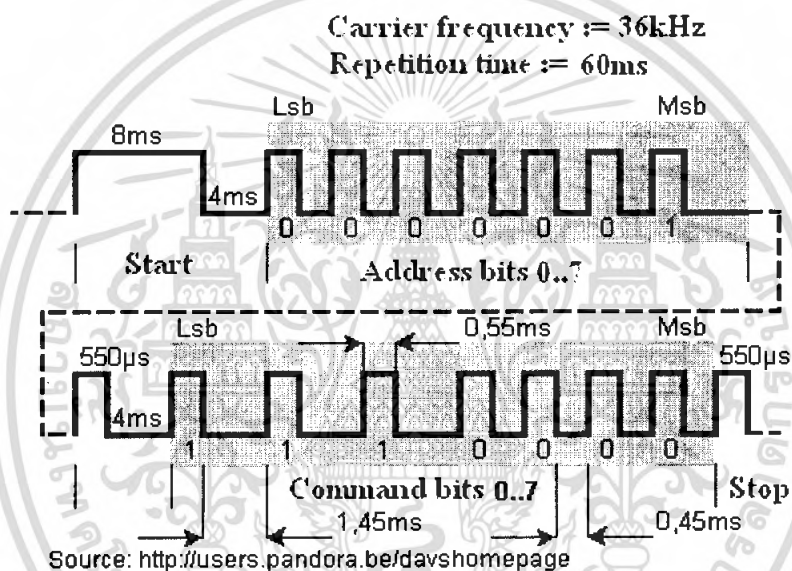
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบิตของลอจิก 1 จะขึ้นต้นด้วยช่วงของโวลเตจสูง ในเวลา 550 us ตามด้วยช่วงโวลเตจต่ำ 1450us ซึ่งบิตลอจิก 1 จะใช้เวลาทั้งหมด 2 ms

ความถี่พาห้ของสัญญาณอินฟราเรดที่ใช้คือ 38 kHz โดยในการส่งนั้นจะทำการส่ง start bit โดย start bit มีพัลส์ช่วงโวลเตจสูง 8 ms และตามด้วย พัลส์โวลเตจต่ำ 4 ms หลังจากนั้น address bit 7 บิต และ command bit 7 บิตถูกส่งออกไป แต่ระหว่าง address bit และ command bit ทั้งสองจะถูกแบ่งด้วย พัลส์ที่มีโวลเตจสูง 550 us ตามด้วยสัญญาณโวลเตจต่ำ 4 ms และจะยกเลิกการส่งเมื่อมีสัญญาณโวลเตจสูงปิดท้าย ซึ่งมีช่วงเวลา 550 us

ระยะเวลาระหว่าง start bit สองสตาร์ทบิต ของ bitstream เมื่อป้อนในรีโมทคอนโทรลถูกกด คือ 60 ms

### Daewoo infrared remote protocol



### **SONY infrared remote protocol**

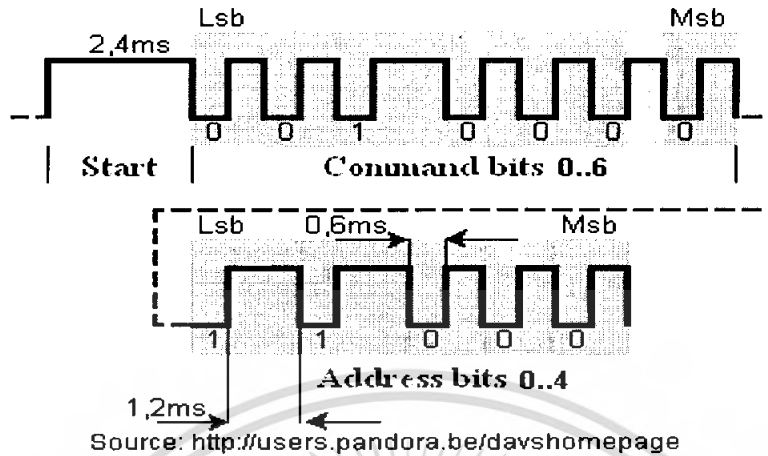
ในรีโมทคอนโทรลของ Sony จะใช้พื้นฐานการเข้ารหัสแบบ Pulse-Width signal coding ซึ่ง code จะประกอบไปด้วยบิต 12 บิต ส่งไปกับคลื่นพาห้ 40 MHz โคดจะเริ่มจาก header ขนาด 2 หรือ 4 ms หรือว่า 4T โดยที่ T คือ 600us จาก header จะตามไปด้วย command bit 7 บิต และ address bit 5 บิต ซึ่งทั้ง command bit และ address bit จะประกอบไปด้วยทั้งลอจิก 1 และ 0 ลอจิก 1 จะประกอบไปด้วย space เวลา 1T หรือ 600 us และพัลส์ขนาด 1200 us ลอจิก 0 จะประกอบไปด้วย space 600 us และ พัลส์ 600 us

ช่องว่างระหว่าง code ที่ส่ง ในระหว่างที่ปุ่มถูกกด มีเวลา 40 ms ความยาวทั้งหมดของทั้งบิตสตรีมจะมีขนาด 45 ms

## Sony infrared remote protocol

Carrier frequency := 40kHz

Repetition time := 40ms



## **JVC infrared remote protocol**

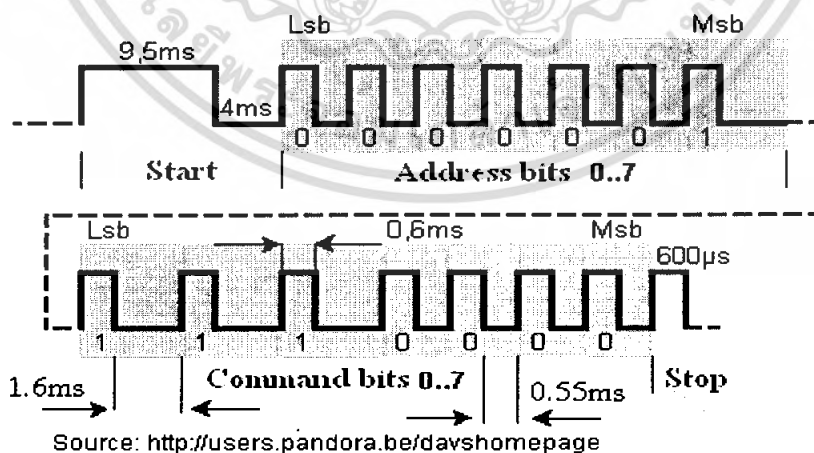
อินฟราเรด รีโมทโปรโตคอลจะมีลักษณะที่คล้ายกับของ Daewoo มาก ซึ่งจะมีแค่เวลาและช่องว่างระหว่าง address bit และ data bit ที่จะแตกต่างกันบ้างเล็กน้อย

ใน JVC โปรโตคอล โลอจิก 0 จะมีพัลส์โวลเตจสูงที่มีช่วงเวลา 600 us ตามด้วยพัลส์โวลเตจต่ำ 550us ส่วนโลอจิก 1 จะประกอบไปด้วย ช่วงโวลเตจสูง 600 us ตามด้วยพัลส์โวลเตจต่ำ 1600 us โปรโตคอลของ JVC จะใช้คลื่นความถี่พาห์ 38 kHz

## JVC infrared remote control

Carrier frequency := 38kHz

Repetition time := 60ms



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Panasonic's old infrared remote protocol

สำหรับการส่งข้อมูล โปรโตคอลของ Panasonic จะใช้การมอดูเลตแบบ pulse-place modulation

โดยมีการระบุโค้ดไว้ 2048 โค้ดในโปรโตคอลนี้ โดยแบ่งเป็น custom code 5 บิต และ data code 6 บิต custom code เป็นค่าที่ใช้ระบุโค้ดของผู้ผลิต และ data code เป็นค่าที่ใช้ระบุในการกดปุ่มต่างๆของรีโมทคอนโทรล

โค้ดที่ถูกส่งทั้งหมดจะมี 22 บิต ส่วนของ header จะเป็น custom code 5 บิต และตามด้วย คอ้ดโค้ด ต่อจากนั้นจะเป็นอินเวอร์สของ custom code และ data code และสุดท้ายเพื่อกเลิกการส่ง จะมี stop bit ปิดท้าย โดยที่บิตที่ส่งที่มีการอินเวอร์สจะมีประโยชน์ในการตรวจจับข้อผิดพลาดที่เกิดขึ้น แต่บิตที่ส่วนแรกของบิต จะต้องเป็นโวลเตจสูงเสมอ แล้วจึงตามด้วย โลจิกต่ำ สำหรับเวลานั้น ในโลจิก 1 และโลจิก 0 จะมีเวลาแตกต่างกันไป

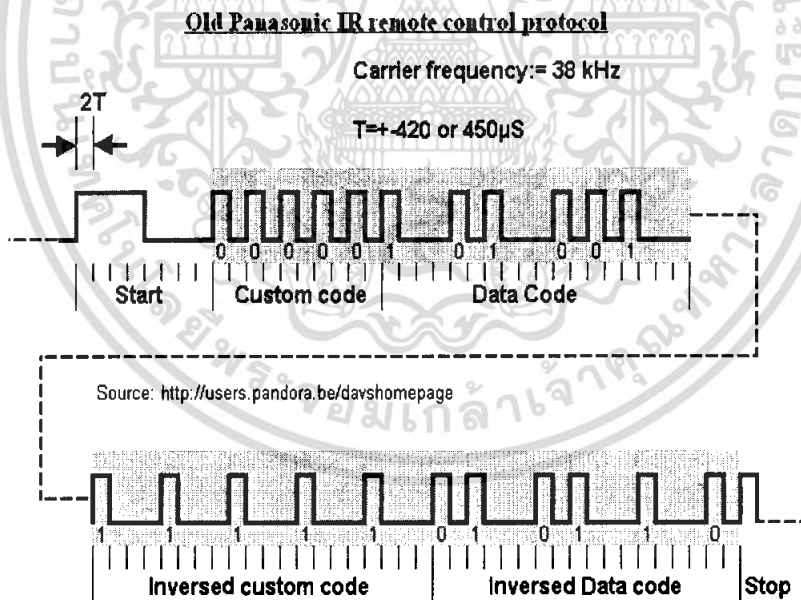
$T = 420 \mu s$  จะมีค่าประมาณ  $424 \mu s$  ในสหรัฐ และแคนาดา

$T = 454 \mu s$  จะมีค่าประมาณ  $460 \mu s$  ในยุโรป

เฮดเดอร์ จะมีเวลาเป็น  $8T$  high และ  $8T$  low

โลจิก 1 จะมี  $2T$  high และ  $6T$  low

โลจิก 0 จะมี  $2T$  high และ  $2T$  low



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# GaAs/GaAlAs IR Emitting Diode in $\varnothing$ 5 mm (T-1 $\frac{3}{4}$ ) Package

## Description

TSAL7400 is a high efficiency infrared emitting diode in GaAlAs on GaAs technology, molded in clear plastic packages.

In comparison with the standard GaAs on GaAs technology these emitters achieve more than 100 % radiant power improvement at a similar wavelength. The forward voltages at low current and at high pulse current roughly correspond to the low values of the standard technology. Therefore these emitters are ideally suitable as high performance replacements of standard emitters.



94 8389

## Features

- Extra high radiant power and radiant intensity
- High reliability
- Low forward voltage
- Suitable for high pulse current operation
- Standard T-1 $\frac{3}{4}$  ( $\varnothing$  5 mm) package
- Angle of half intensity  $\varphi = \pm 25^\circ$
- Peak wavelength  $\lambda_p = 940$  nm
- Good spectral matching to Si photodetectors

## Applications

Infrared remote control units with high power requirements  
Free air transmission systems  
Infrared source for optical counters and card readers  
IR source for smoke detectors

## Absolute Maximum Ratings

$T_{amb} = 25^\circ\text{C}$

Parameter	Test Conditions	Symbol	Value	Unit
Reverse Voltage		$V_R$	5	V
Forward Current		$I_F$	100	mA
Peak Forward Current	$t_p/T = 0.5, t_p = 100 \mu\text{s}$	$I_{FM}$	200	mA
Surge Forward Current	$t_p = 100 \mu\text{s}$	$I_{FSM}$	1.5	A
Power Dissipation		$P_V$	210	mW
Junction Temperature		$T_j$	100	$^\circ\text{C}$
Operating Temperature Range		$T_{amb}$	-55...+100	$^\circ\text{C}$
Storage Temperature Range		$T_{stg}$	-55...+100	$^\circ\text{C}$
Soldering Temperature	$t \leq 5\text{sec}, 2 \text{ mm from case}$	$T_{sd}$	260	$^\circ\text{C}$
Thermal Resistance Junction/Ambient		$R_{thJA}$	350	K/W

### Basic Characteristics

$T_{amb} = 25^{\circ}\text{C}$

Parameter	Test Conditions	Symbol	Min	Typ	Max	Unit
Forward Voltage	$I_F = 100\text{ mA}, t_p = 20\text{ ms}$	$V_F$		1.35	1.6	V
	$I_F = 1\text{ A}, t_p = 100\text{ }\mu\text{s}$	$V_F$		2.6	3	V
Temp. Coefficient of $V_F$	$I_F = 100\text{ mA}$	$TK_{V_F}$		-1.3	-	mV/K
Reverse Current	$V_R = 5\text{ V}$	$I_R$			10	$\mu\text{A}$
Junction Capacitance	$V_R = 0\text{ V}, f = 1\text{ MHz}, E = 0$	$C_j$		25		pF
Radiant Intensity	$I_F = 100\text{ mA}, t_p = 20\text{ ms}$	$I_e$	25	40		mW/sr
	$I_F = 1.0\text{ A}, t_p = 100\text{ }\mu\text{s}$	$I_e$	220	310		mW/sr
Radiant Power	$I_F = 100\text{ mA}, t_p = 20\text{ ms}$	$\phi_e$		35		mW
Temp. Coefficient of $\phi_e$	$I_F = 20\text{ mA}$	$TK_{\phi_e}$		-0.6		%/K
Angle of Half Intensity		$\varphi$		$\pm 25$		deg
Peak Wavelength	$I_F = 100\text{ mA}$	$\lambda_p$		940		nm
Spectral Bandwidth	$I_F = 100\text{ mA}$	$\Delta\lambda$		50		nm
Temp. Coefficient of $\lambda_p$	$I_F = 100\text{ mA}$	$TK_{\lambda_p}$		0.2		nm/K
Rise Time	$I_F = 100\text{ mA}$	$t_r$		800		ns
Fall Time	$I_F = 100\text{ mA}$	$t_f$		800		ns
Virtual Source Diameter	method: 63% encircled energy	$\varnothing$		2.8		mm

### Typical Characteristics ( $T_{amb} = 25^{\circ}\text{C}$ unless otherwise specified)

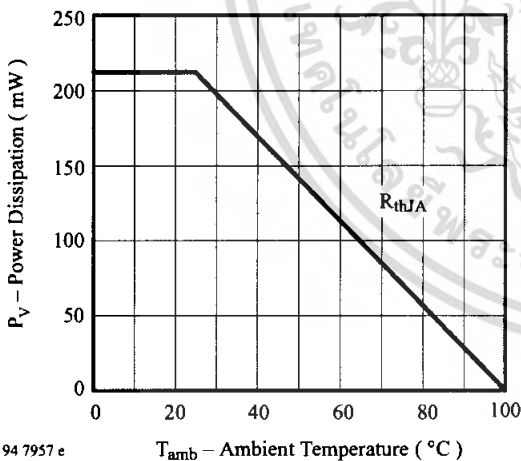


Figure 1. Power Dissipation vs. Ambient Temperature

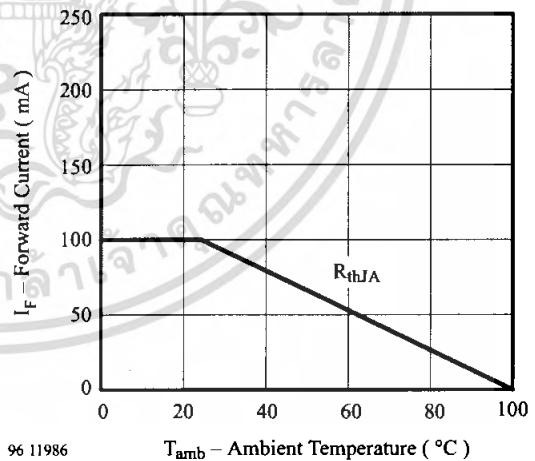
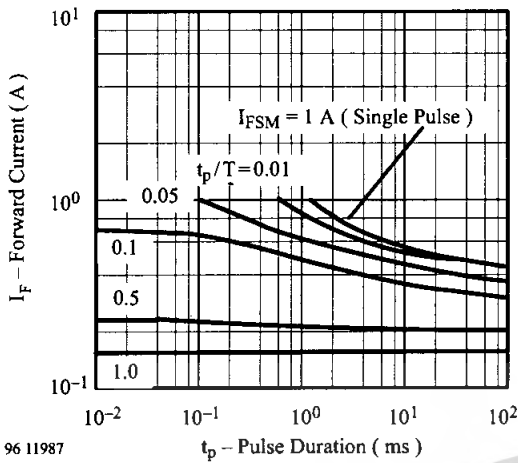
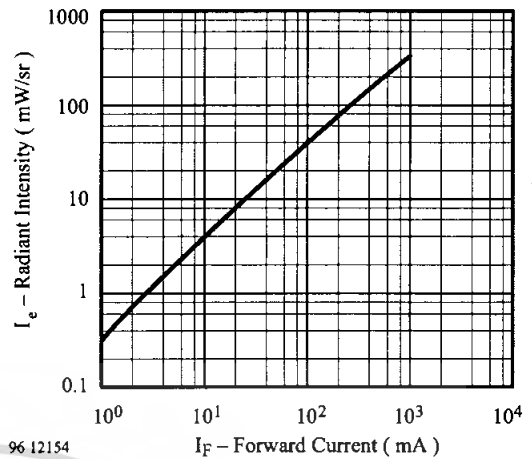


Figure 2. Forward Current vs. Ambient Temperature



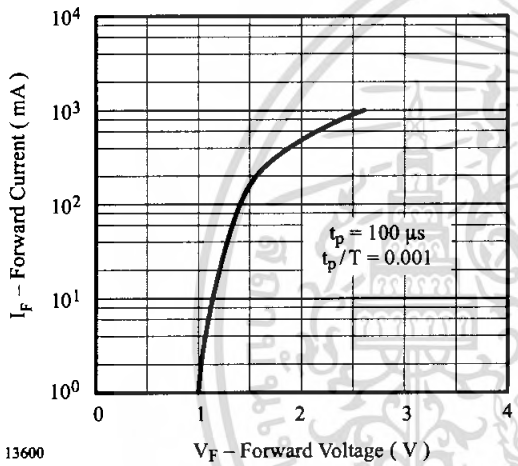
96 11987

Figure 3. Pulse Forward Current vs. Pulse Duration



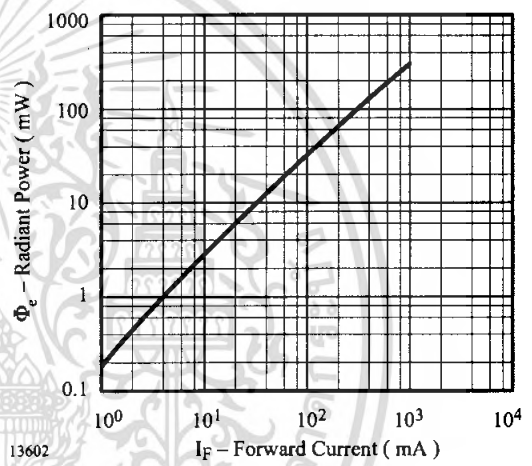
96 12154

Figure 6. Radiant Intensity vs. Forward Current



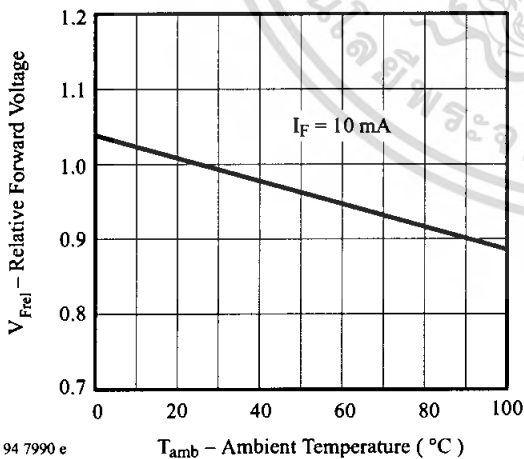
13600

Figure 4. Forward Current vs. Forward Voltage



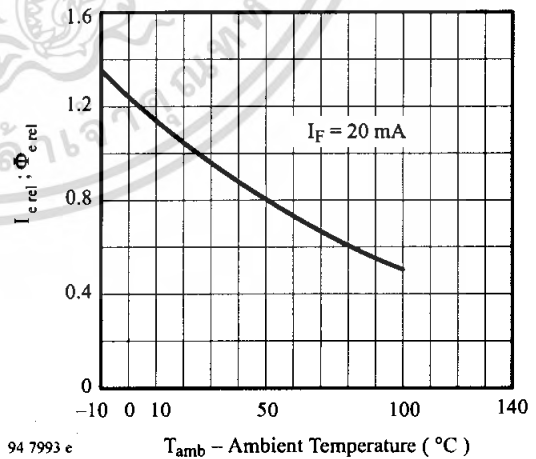
13602

Figure 7. Radiant Power vs. Forward Current



94 7990 e

Figure 5. Relative Forward Voltage vs. Ambient Temperature



94 7993 e

Figure 8. Rel. Radiant Intensity/Power vs. Ambient Temperature

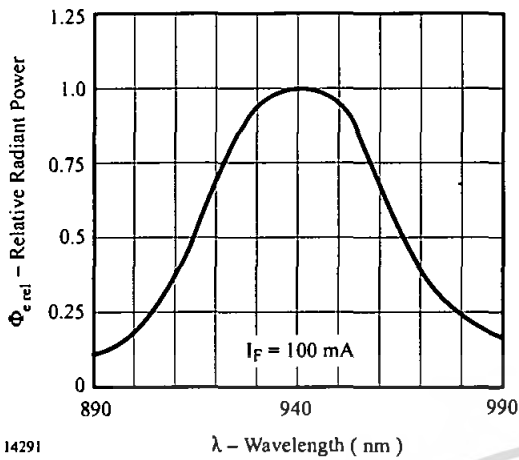


Figure 9. Relative Radiant Power vs. Wavelength

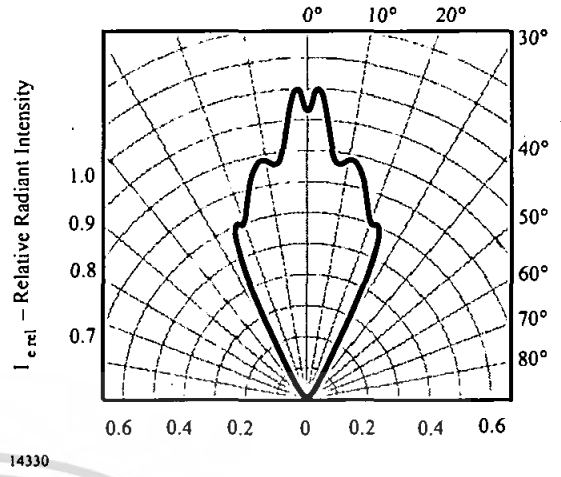
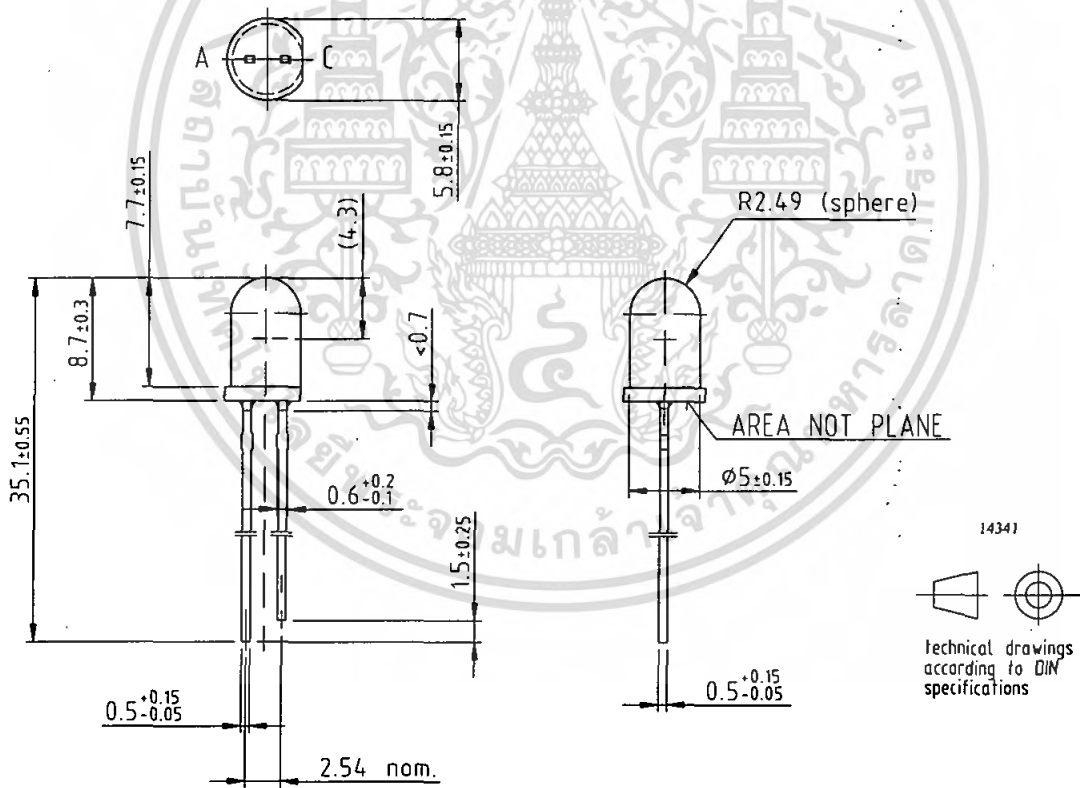


Figure 10. Relative Radiant Intensity vs. Angular Displacement

### Dimensions in mm



## Ozone Depleting Substances Policy Statement

It is the policy of **Vishay Semiconductor GmbH** to

1. Meet all present and future national and international statutory requirements.
2. Regularly and continuously improve the performance of our products, processes, distribution and operating systems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

**Vishay Semiconductor GmbH** has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively
2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

**Vishay Semiconductor GmbH** can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

**We reserve the right to make changes to improve technical design and may do so without further notice.**

Parameters can vary in different applications. All operating parameters must be validated for each customer application by the customer. Should the buyer use Vishay-Telefunken products for any unintended or unauthorized application, the buyer shall indemnify Vishay-Telefunken against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim of personal damage, injury or death associated with such unintended or unauthorized use.

Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany  
Telephone: 49 (0)7131 67 2831, Fax number: 49 (0)7131 67 2423