สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมคอมพิวเตอร์ระยะไกล

Computer Remote Control

โดย

นายชุมพล ผู้เจริญวิบูลย์

นายฉัชพงษ์ เจียมหาทรัพย์

นายศุภกิจ ภูสีมา

เลขหมู่.....
เลขทะเบียน 61946
วัน,เดือน,ปี 25 ก.ค. 2549

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

ภาควิชา
วิศวกรรมโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เครื่องควบคุมคอมพิวเตอร์ระยะไกล

Computer Remote Control

โดย

นายชุมพล ผู้เจริญวิบูลย์ 44010120

นายณัชพงษ์ เจียมหาทรัพย์ 44010121

นายศุภกิจ ภูสีมา 44010489

อาจารย์ที่ปรึกษา

ดร. พิพัฒน์ พรหมมี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2547

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **เครื่องควบคุมคอมพิวเตอร์ระยะไกล**

Computer Remote Control

ผู้จัดทำ

1. นายชุมพล ผู้เจริญวิบูลย์ 44010120

2. นายณัชพงษ์ เจียมหาทรัพย์ 44010121

3. นายศุภกิจ ภูสีมา 44010489

..... อาจารย์ที่ปรึกษา

(ดร. พิพัฒน์ พรหมมี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมคอมพิวเตอร์ระยะไกล

Computer Remote Control

โดย	นายชุมพล	ผู้เจริญวิบูลย์	44010120
	นายณัชพงษ์	เจียมหาทรัพย์	44010121
	นายศุภกิจ	ภูสีมา	44010489

อาจารย์ที่ปรึกษา ดร. พิพัฒน์ พรหมมี

บทคัดย่อ

โครงงานนี้เสนอการสร้างเครื่องควบคุมคอมพิวเตอร์ระยะไกล โดยใช้อุปกรณ์รีโมทคอนโทรลแบบอินฟราเรด โดยสร้างอุปกรณ์รับสัญญาณต่อเชื่อมกับคอมพิวเตอร์ผ่านพอร์ทสื่อสาร เมื่อคอมพิวเตอร์ได้รับสัญญาณจากรีโมทก็จะนำสัญญาณมาทำการกำหนดเป็นสัญญาณควบคุมด้วยโปรแกรมที่เขียนขึ้น โดยสามารถกำหนดการใช้งานได้อย่างเป็นอิสระ สามารถนำไปประยุกต์ใช้งานควบคุมโปรแกรมต่าง ๆ ที่ติดตั้งอยู่ในคอมพิวเตอร์ได้

Abstract

The Computer Remote Control based on common infrared remote control equipment is presented. The receiver is implemented and interfaced with computer communication port. The received signals have been converted to controlling signal according to developed program. The controlling functions can be flexibly define for control any function of application programs in the computer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ทฤษฎีรีโมทแบบอินฟราเรด	2
2.1.1 แสงอินฟราเรด	2
2.1.2 การส่งรหัสควบคุมแบบพีพีเอ็ม	2
2.1.3 ตัวส่งสัญญาณ	3
2.1.4 ตัวรับสัญญาณ	4
2.1.5 ความแตกต่างของการเข้ารหัส	5
2.1.6 รูปแบบการส่งสัญญาณ (Protocol)	6
2.1.6.1 Sharp Protocol	6
2.1.6.2 Sony SIRC Protocol	8
2.1.6.3 Philips RC-5 Protocol	9
2.1.6.4 Nokia NRC17 Protocol	12
2.2 ทฤษฎีไมโครคอนโทรลเลอร์	14
2.2.1 โครงสร้างทางฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ ตระกูล PIC 16F877	14
2.2.2 การจัดสรรหน่วยความจำโปรแกรม	18
2.2.3 การจัดสรรหน่วยความจำข้อมูลแรมและรีจิสเตอร์ไฟล์	19
2.2.4 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F87x	20
2.3 การเชื่อมต่อกับเครื่องคอมพิวเตอร์	23
2.3.1 การสื่อสารแบบอนุกรม	24
2.3.1.1 การสื่อสารอนุกรมแบบซิงโครนัส	24
2.3.1.2 การสื่อสารข้อมูลแบบอะซิงโครนัส	24
2.3.2 มาตรฐานพอร์ทอนุกรมแบบอาร์เอสสองสามสอง (RS – 232)	26
2.3.3 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรมอาร์เอสสองสามสอง	39
2.3.4 ข้อจำกัดในการใช้งานของมาตรฐาน อาร์เอสสองสามสอง	30
2.3.5 การสื่อสารอนุกรมยูเออาร์ที (UART : Universal Asynchronous Receiver -Transmitter)	33
2.3.5.1 ชนิดของยูเออาร์ที	34
บทที่ 3 หลักการออกแบบและการสร้าง	36
3.1 การออกแบบฮาร์ดแวร์	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 รีโมตคอนโทรล 36
3.1.2 ภาครับสัญญาณอินฟราเรด 37
3.2 การออกแบบซอฟท์แวร์ 39
3.2.1 การออกแบบซอฟท์แวร์ของไมโครคอนโทรลเลอร์ 39
3.2.2 การออกแบบซอฟท์แวร์ในคอมพิวเตอร์ 39
บทที่ 4 การทดลองและผลการทดลอง 50
4.1 ผลการทดลองส่วนรับสัญญาณ 50
4.2 ผลการทดลองส่วนการเขียนโปรแกรมในคอมพิวเตอร์ 51
4.3 วิธีการใช้โปรแกรม 52
4.4 วงจรและอุปกรณ์ที่ใช้ 57
บทที่ 5 บทวิจารณ์และบทสรุป 59
กิตติกรรมประกาศ
หนังสืออ้างอิง
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 บล็อกไดอะแกรมของภาคส่งรีโมตอินฟราเรด	2
รูปที่ 2.2 รูปแบบของสัญญาณพีพีเอ็ม	2
รูปที่ 2.3 วงจรส่ง	3
รูปที่ 2.4 วงจรส่งแบบ emitter follower	4
รูปที่ 2.5 ตัวอย่างบล็อกไดอะแกรมของตัวรับสัญญาณอินฟาเรด	4
รูปที่ 2.6 ตัวรับสัญญาณ	5
รูปที่ 2.7 รูปแบบของสัญญาณพัลซ์ที่มีการเข้ารหัสโดยใช้ความกว้างของพัลซ์	5
รูปที่ 2.8 รูปแบบของสัญญาณพัลซ์ที่มีการเข้ารหัสโดยใช้ช่องว่างของพัลซ์	5
รูปที่ 2.9 รูปแบบของสัญญาณพัลซ์ที่มีการเข้ารหัสโดยใช้ช่องการเลื่อนของพัลซ์	6
รูปที่ 2.10 รูปค่าลอจิกของ sharp protocol	7
รูปที่ 2.11 รูปแสดงตัวอย่างขบวนสัญญาณพัลซ์ของ Sharp Protoccol	7
รูปที่ 2.12 รูปแสดงการส่งสัญญาณซ้ำเมื่อมีการกดปุ่มค้างของ Sharp Protocol	7
รูปที่ 2.13 รูปค่าลอจิกของ Sony SIRC Protocol	8
รูปที่ 2.14 รูปแสดงตัวอย่างขบวนสัญญาณของพัลซ์ Sony SIRC Protocol	8
รูปที่ 2.15 รูปค่าลอจิกของ Philips RC-5 Protocol	10
รูปที่ 2.16 รูปแสดงตัวอย่างขบวนสัญญาณพัลซ์ของ Philips RC-5 Protocol	10
รูปที่ 2.17 รูปค่าลอจิกของ Nokia NRC-17 Protocol	13
รูปที่ 2.18 รูปแสดงตัวอย่างขบวนสัญญาณพัลซ์พัลซ์ของ Nokia NRC-17 Protocol	13
รูปที่ 2.19 รูปแสดงการส่งสัญญาณซ้ำเมื่อกดปุ่มค้างของ Nokia NRC-17 Protocol	14
รูปที่ 2.20 ไดอะแกรมแสดงรูปแบบสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ต	15
รูปที่ 2.21 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบซิงโครนัส	24
รูปที่ 2.22 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส	24
รูปที่ 2.23 การจัดขาของConnectorพอร์ตอนุกรมมาตรฐานอาร์เอสสองสามสอง ทั้งแบบ DB – 9 และ DB – 25	27
รูปที่ 2.24 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ	28
รูปที่ 2.25 แสดงวงจรขับแบบอาร์เอสสองสามสอง โดยใช้ MAX 232	30
รูปที่ 2.26 การกำหนดระดับสัญญาณตามมาตรฐาน RS – 232 ทางด้านอุปกรณ์ภาครับ	31
รูปที่ 2.27 ปัญหาจากความแตกต่างของกราวนด์ในการเชื่อมต่อผ่านมาตรฐาน RS – 232	32
รูปที่ 2.28 บล็อกไดอะแกรมของยูเออาร์ที (UART)	33
รูปที่ 3.1 บล็อกไดอะแกรมของฮาร์ดแวร์	36
รูปที่ 3.2 มาตรฐานการส่งข้อมูลแบบ RC5 ของฟิลิปส์	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.3 ตัวอย่างบล็อกไดอาแกรมของตัวรับสัญญาณอินฟราเรด 37
รูปที่ 3.4 วงจรรับสัญญาณอินฟราเรด 38
รูปที่ 3.5 รูปวงจรทั้งหมด 38
รูปที่ 3.6 แสดงถึงคอมมานด์บิตซึ่งเป็นส่วนที่แต่ละปุ่มจะมีค่าไม่เท่ากัน 39
รูปที่ 3.7 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมการนำข้อมูลจากรีโมตไปถอดรหัส 40
รูปที่ 3.8 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมย่อยในไมรโครคอนโทรลเลอร์ 41
รูปที่ 3.9 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมที่ใช้ควบคุมโปรแกรมใน
คอมพิวเตอร์ด้วยรีโมท 42
รูปที่ 3.10 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมย่อย 43
รูปที่ 3.11 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Winamp 44
รูปที่ 3.12 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Power DVD 45
รูปที่ 3.13 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Power Point 46
รูปที่ 3.14 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Pin ball 47
รูปที่ 3.15 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Windows Media Player 48
รูปที่ 3.16 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Magic ball 49
รูปที่ 4.1 รูปตัวอย่างสัญญาณที่วัดจากเอาท์พุตของโฟโต้โมดูล และค่าที่ถูกถอดรหัสออกมา 50
รูปที่ 4.2 แสดงหน้าโปรแกรมการใช้งาน 51
รูปที่ 4.3 แสดงหน้าต่างเมื่อมีการกดปุ่มผิด หรือปุ่มที่ไม่สามารถใช้งานได้ 52
รูปที่ 4.4 แสดงหน้าตาโปรแกรม Winamp ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “1” 54
รูปที่ 4.5 แสดงหน้าตาโปรแกรม Power DVD ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “2” 55
รูปที่ 4.6 แสดงหน้าตาโปรแกรม Power Point ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “3” 55
รูปที่ 4.7 แสดงหน้าตาโปรแกรม Pin ball ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “4” 56
รูปที่ 4.8 แสดงหน้าตาโปรแกรม Windows Media Player ที่ใช้ในการทดลอง
เมื่อมีการกดรีโมทที่ปุ่ม “5” 56
รูปที่ 4.9 แสดงหน้าตาโปรแกรม Magic ball ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “6” 57
รูปที่ 4.10 แสดงรูปวงจรและอุปกรณ์ทั้งหมดที่ใช้ในโครงงาน 57
รูปที่ 4.11 แสดงวงจรที่ใช้ในการรับสัญญาณ, ถอดรหัส และส่งข้อมูลไปยังคอมพิวเตอร์ 58
รูปที่ 4.12 แสดงรูปรีโมทที่ใช้ในการทดลอง 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ค่าสัญญาณ Address และ Command ของ Sony SIRC Protocol	9
ตารางที่ 2.2 ค่าสัญญาณ Address และ Command ของ Philips RC-5 Protocol	11-12
ตารางที่ 2.3 ตารางสรุปการทำงานของขาพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ PIC16F877	15-18
ตารางที่ 2.4 แสดงบิตพาริตี้ของข้อมูล	26
ตารางที่ 2.5 แสดงตำแหน่งและชื่อขาของ DB – 9 และ DB – 25	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การดำรงชีวิตของมนุษย์ในยุคที่โลกมีความเจริญก้าวหน้าทางด้านเทคโนโลยี ไม่ว่าจะเป็นเรื่องของการติดต่อสื่อสาร มนุษย์ก็สามารถที่จะติดต่อสื่อสารถึงกันได้แม้ว่าจะอยู่กันคนละซีกของโลกโดยผ่านทางอินเตอร์เน็ต สิ่งต่าง ๆ ที่อยู่รอบตัวมนุษย์ มีการพัฒนาอย่างต่อเนื่องในช่วงระยะเวลาไม่กี่ปีจนถึงปัจจุบัน มนุษย์พยายามที่จะคิดค้นสิ่งประดิษฐ์ใหม่ ๆ ขึ้น เพื่อช่วยอำนวยความสะดวกให้การดำรงชีวิตง่ายขึ้น อย่างเช่น อุปกรณ์เครื่องใช้ไฟฟ้านับได้ว่าเป็นสิ่งที่จำเป็นสำหรับการดำรงชีวิตของมนุษย์ ทุกครอบครัวล้วนต้องพึ่งพาสิ่งอำนวยความสะดวกเหล่านี้

1.1 แนวความคิดและที่มาของการทำปริญญานิพนธ์

อุปกรณ์เครื่องใช้ไฟฟ้านับได้ว่าเป็นส่วนสำคัญต่อการดำรงชีวิตของมนุษย์เป็นอย่างมากไม่ว่าจะเป็น โทรทัศน์, วิทยุ, พัดลม, วีดีโอ ปัจจุบันเครื่องใช้ไฟฟ้าต่าง ๆ เหล่านี้พัฒนาก้าวหน้ามาก นอกจากการเปิดปิดโดยปกติที่ใช้สวิตซ์แล้ว ยังเพิ่มการควบคุมด้วยรีโมทคอนโทรล ซึ่งถือได้ว่าเป็นเสมือนแขน ขา ของมนุษย์ช่วยให้มนุษย์สะดวกสบายมากยิ่งขึ้น โดยเครื่องใช้ไฟฟ้าเหล่านี้มีข้อจำกัดในการส่งงานเฉพาะของแต่ละชนิด

คอมพิวเตอร์ก็เป็นเครื่องใช้ไฟฟ้าที่เริ่มมีบทบาทในชีวิตประจำวันของเรามากขึ้น ทั้งเพื่อความบันเทิง เพื่อทำงานต่าง ๆ ที่ผู้ใช้ต้องการคงเป็นเรื่องดีไม่น้อยหากเราสามารถควบคุมการทำงานต่าง ๆ ในคอมพิวเตอร์โดยไม่ต้องนั่งหน้าเครื่องคอมพิวเตอร์เท่านั้น แต่ใช้รีโมทเพียงตัวเดียว

1.2 ความมุ่งหมายและวัตถุประสงค์ของการทำปริญญานิพนธ์

1.2.1 เพื่อศึกษาหลักการทำงานของระบบรีโมทคอนโทรล

1.2.2 เพื่อศึกษาลักษณะการเขียนโปรแกรมคำสั่งของไมโครคอนโทรลเลอร์ด้วยภาษาซี

1.2.3 เพื่อศึกษาลักษณะการเขียนโปรแกรมคำสั่งของคอมพิวเตอร์ด้วยภาษาซี

1.2.4 เพื่อฝึกทักษะในการเขียนโปรแกรมคำสั่งให้คอมพิวเตอร์ให้ทำงานตามที่ได้ออกแบบไว้

1.3 ขอบเขตของการทำปริญญานิพนธ์

สร้างเครื่องควบคุมคอมพิวเตอร์ระยะไกล ด้วยระบบอินฟราเรด โดยรีโมทใช้รีโมทสำเร็จรูป (รีโมทโทรทัศน์) ส่วนตัวรับสัญญาณจากรีโมท ใช้โมดูลรับสัญญาณอินฟราเรด นำสัญญาณที่ได้มาถอดรหัสและประมวลผลโดยไมโครคอนโทรลเลอร์ (Microchip PIC16F877) เพื่อนำไปควบคุมโปรแกรมในคอมพิวเตอร์ที่ได้เขียนโปรแกรมรองรับไว้แล้ว โดยผ่านพอร์ตอนุกรม คอมพิวเตอร์จะนำข้อมูลที่ได้ไปประมวลผลว่า ข้อมูลที่ส่งมานั้นเป็นคำสั่งให้ทำอะไร แล้วนำไปทำงานทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

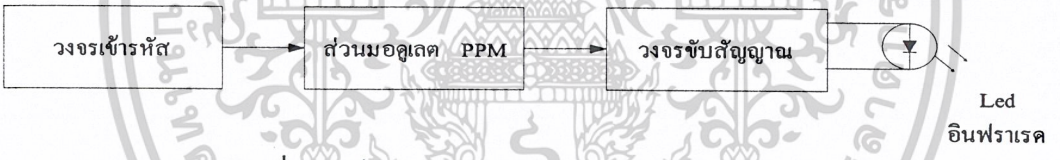
2.1 ทฤษฎีรีโมทแบบอินฟราเรด

2.1.1 แสงอินฟราเรด

อินฟราเรด คือ แสงที่มนุษย์ไม่สามารถมองเห็นได้ เนื่องจากความยาวคลื่นของแสงอยู่ที่ 950 nm ซึ่งน้อยกว่าแสงที่สามารถมองเห็นได้ เป็นเหตุผลที่เลือกใช้ในรีโมทเพราะเราต้องการใช้แต่ไม่ต้องการที่จะเห็นแสงอินฟราเรด โดยจะถูกส่งด้วยค่าความถี่ในช่วง 30 – 40 kHz สาเหตุที่ต้องใช้ค่าความถี่สูงเพื่อเป็นการป้องกันแสงจากแหล่งกำเนิดแสงแหล่งอื่น ๆ มารบกวน

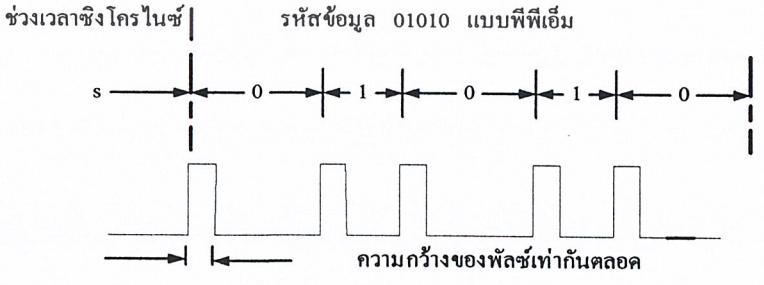
2.1.2 การส่งรหัสควบคุมแบบพีพีเอ็ม

รีโมทโทรทัศน์ที่ใช้ในปัจจุบันนี้ ใช้หลักการส่งสัญญาณควบคุมแบบพีพีเอ็ม โดยในภาคส่งสัญญาณควบคุมทำหน้าที่จัดแบบของรหัสควบคุมตามที่กำหนด จากนั้นทำการมอดูเลตให้เป็นสัญญาณพีพีเอ็ม ก่อนส่งไปยังวงจรขับสัญญาณเพื่อแปลงให้เป็นแสงสำหรับส่งออกไปดังรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรมของภาคส่งรีโมทอินฟราเรด

สัญญาณพีพีเอ็มเกิดจากการมอดูเลตสัญญาณในลักษณะของตำแหน่งพัลส์ ขนาดความกว้างของสัญญาณพัลส์จะมีค่าเท่ากันตลอดจึงไม่ใช้บอกชนิดของข้อมูลแต่จะใช้คาบเวลาของพัลส์แต่ละลูกเป็นตัวกำหนดชนิดของข้อมูลดังรูปที่ 2.2



รูปที่ 2.2 รูปแบบของสัญญาณพีพีเอ็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

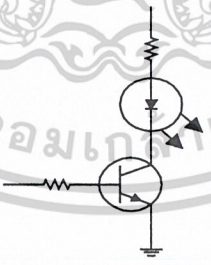
โดยหลักการแล้ว การมอดูเลตแบบพีพีเอ็มจะใช้แบ่งช่วงสัญญาณด้วยคาบเวลาที่เท่ากัน แต่จุดเวลาที่แสดงสัญญาณพัลส์แตกต่างกัน เช่น หากสัญญาณเป็นศูนย์ สัญญาณพัลส์จะปรากฏ ณ ตำแหน่งที่กึ่งกลางของคาบเวลาที่กำหนดหากสัญญาณมีแอมปลิจูดเป็นบวก สัญญาณพัลส์จะปรากฏในตำแหน่งที่ลึกไปทางขวาของจุดกึ่งกลางคาบเวลาโดยมีระยะห่างขึ้นกับค่าของแอมปลิจูดในลักษณะเป็นสัดส่วนกัน

ในทำนองกลับกันหากสัญญาณมีแอมปลิจูดลบ สัญญาณพัลส์จะปรากฏอยู่ครึ่งช่วงแรกของคาบเวลา ดังนั้นการมอดูเลตแบบพีพีเอ็มจึงสามารถใช้ได้ทั้งสัญญาณที่เป็นอนาลอกและดิจิตอล เพียงแต่ในสัญญาณแบบดิจิตอลเราจะเห็นระยะห่างของพัลส์ไม่แน่นอนกว่า เพราะมีขนาดสัญญาณเพียง 2 ระดับ

2.1.3 ตัวส่งสัญญาณ

ตัวส่งสัญญาณมักเป็นแบบพกพาที่ใช้พลังงานแบตเตอร์รี่ ตัวส่งควรใช้พลังงานให้น้อยที่สุดเท่าที่ทำได้และสัญญาณอินฟราเรดควรเข้มมากพอที่จะส่งสัญญาณควบคุมได้ในระยะทางที่ยอมรับได้ ชิพจำนวนมากถูกออกแบบมาเพื่อใช้กับตัวส่งสัญญาณอินฟราเรด แต่ในทุกวันนี้ใช้ไมโครคอนโทรลเลอร์กำลังต่ำแทนเพื่อเหตุผลง่ายๆที่ว่าให้ใช้ได้ยืดหยุ่นมากขึ้น เมื่อไม่ได้กดปุ่มตัวส่งจะอยู่ในภาวะกำลังต่ำ (Sleep mode) ซึ่งไม่กินกระแสมากนัก ตัวประมวลผลสั่งงานให้ส่งคำสั่งอินฟราเรดเมื่อกดปุ่มเท่านั้น

กระแสที่ไหลผ่านตัว LED มีค่าได้ตั้งแต่ 100 mA ถึงมากกว่า 1A เพื่อที่จะหาระยะควบคุมที่ยอมรับได้ กระแสที่ LED จำเป็นต้องสูงเท่าที่จะทำได้ ตัวส่งสัญญาณควรสร้างตามค่าพารามิเตอร์ของ LED เช่น อายุการใช้งานของแบตเตอรี่และระยะการควบคุมมากสุด กระแส LED เป็นค่าสูงได้เพราะสัญญาณพัลส์ที่ขับ LED สั้นมาก กำลังสูญเสียโดยเฉลี่ยของ LED ไม่ควรเกินค่าที่มากที่สุดที่ยอมให้ผ่านได้

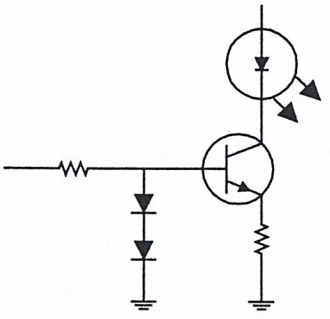


รูปที่ 2.3 วงจรส่ง

วงจรทรานซิสเตอร์แบบง่าย ๆ ที่ใช้ขับ LED เลือกทรานซิสเตอร์ที่เหมาะสม ค่าความต้านทานที่สามารถคำนวณได้จากกฎของโอห์ม โดยค่าศักดาตกคร่อม LED มีค่าประมาณ 1.1 โวลท์

ตัวขับสัญญาณปกติที่ได้อธิบายไว้ข้างต้นมีข้อเสีย คือ เมื่อค่าศักดาของแบตเตอรี่ตกลงกระแสไหลผ่าน LED จะลดลงด้วย ส่งผลให้ระยะควบคุมสั้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

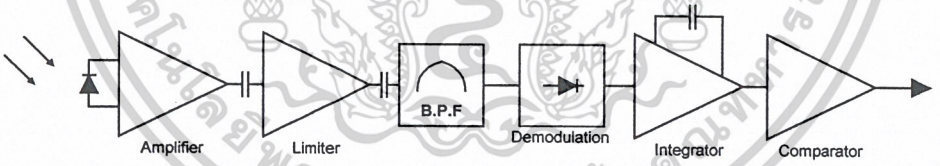


รูปที่ 2.4 วงจรส่งแบบ Emitter follower

วงจร Emitter follower สามารถหลีกเลี่ยงเหตุการณ์นี้โดยนำไดโอด 2 ตัวมาต่ออนุกรมกันจะจำกัดสัญญาณ พัลส์ที่ขาเบสของทรานซิสเตอร์ไว้ที่ 1.2 โวลท์ ค่าศักดาตกคร่อมขาเบสและอิมิตเตอร์ทรานซิสเตอร์ลบออก 0.6 โวลท์ จากค่านั้น ผลที่ได้จะมีค่าแอมปลิจูดคงที่ขนาด 0.6V ที่ขาอิมิตเตอร์ ค่าแอมปลิจูดที่ตกคร่อมตัวต้านทานคงที่ให้ผลเป็นสัญญาณกระแสที่เป็นพัลส์มีขนาดคงที่ การคำนวณค่ากระแสที่ผ่านก็ใช้กฎของโอห์มอีกครั้ง

2.1.4 ตัวรับสัญญาณ

วงจรตัวรับสัญญาณมากมายที่มีอยู่ในท้องตลาด หลักการเลือกที่สำคัญที่สุดคือ ความถี่ที่ใช้ในการ modulation



รูปที่ 2.5 ตัวอย่างบล็อกไดอะแกรมของตัวรับสัญญาณอินฟราเรด

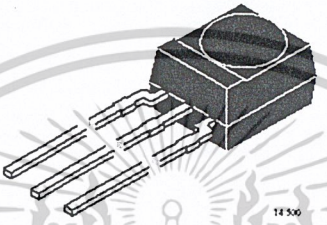
ตัวรับสัญญาณอินฟราเรดจับสัญญาณโดยใช้ไดโอดอินฟราเรดจับสัญญาณที่อยู่ทางด้านซ้ายมือของบล็อก ไดอะแกรม สัญญาณนี้ถูกขยายและจำกัดโดย 2 ตอนแรกตัว Limitter แสดงตัวเป็นวงจร AGC เพื่อหาระดับสัญญาณพัลส์คงที่โดยไม่คำนึงถึงระยะห่างอุปกรณ์

ดังรูปที่ 2.5 สัญญาณไฟฟ้ากระแสสลับเท่านั้นที่ส่งไปที่ Band Pass Filter ตัว Band Pass Filter ถูกปรับไปที่ความถี่ modulation ของอุปกรณ์ส่ง ผู้ใช้อุปกรณ์ทางอิเล็กทรอนิกส์ใช้ช่วงความถี่ปกติมีค่าตั้งแต่ 30 kHz ถึง 60 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนต่อไปเป็นตัวจับสัญญาณ (Integrator และ Comparator) จุดมุ่งหมายหลักของ 3 บล็อกนี้คือ เพื่อจับความถี่การ modulation แล้วทำการ demodulation เอาส่วนที่เป็นความถี่พาหะออกและให้เอาท์พุต เฉพาะส่วนที่เป็นข้อมูลเท่านั้น

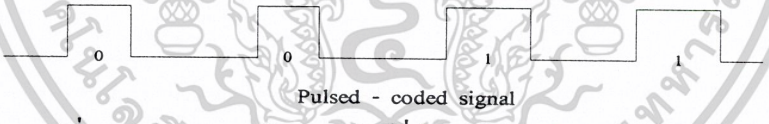
ดังที่กล่าวไปแล้ว บล็อกเหล่านี้ถูกรวมอยู่ในอุปกรณ์ทางอิเล็กทรอนิกส์ตัวเดียว มีผู้ผลิตหลายบริษัท ที่ผลิตอุปกรณ์เหล่านี้ออกสู่ท้องตลาด และอุปกรณ์ส่วนใหญ่ที่หาได้มีหลายแบบและได้ถูกปรับให้มีความถี่ ของการ modulation เฉพาะตัว



รูปที่ 2.6 ตัวรับสัญญาณ

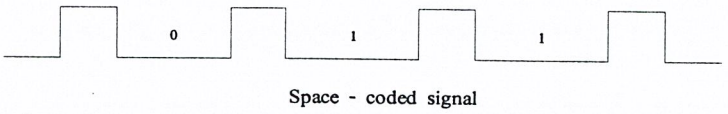
2.1.5 ความแตกต่างของการเข้ารหัส

1. Pulse - Width - Code Signals (ใช้ความกว้างของพัลส์กำหนดรหัส) ใช้หลักการกำหนดความกว้างของพัลส์แทนลอจิก “0” หรือ “1” โดยทั่วไปแล้วจะให้ลอจิก “0“ มีความกว้างของพัลส์ประมาณ 550 ไมโครวินาที และให้ลอจิก “1“ มีความกว้างของพัลส์ประมาณ 2,200 ไมโครวินาที



รูปที่ 2.7 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ความกว้างของพัลส์

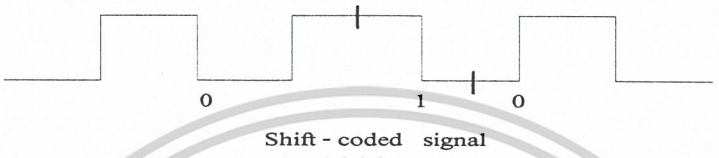
2. Space - Code Signal (ใช้ช่องว่างของพัลส์กำหนดรหัส) ใช้หลักการกำหนดช่องว่างของสัญญาณแทนลอจิก “0“ หรือ “1“ โดยทั่วไปจะให้ลอจิก “0“ มีความกว้างของช่องว่างประมาณ 550 ไมโครวินาที และให้ลอจิก “1“ มีความกว้างของช่องว่างประมาณ 1,650 ไมโครวินาที



รูปที่ 2.8 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ช่องว่างของพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Shift - Code Signal (ใช้ช่องการเลื่อนของพัลส์กำหนดรหัส) จาก 2 แบบข้างต้นนำไปสู่แบบที่ 3 ใช้การกำหนดทั้งความกว้างของพัลส์และความกว้างของช่องว่างของพัลส์เป็นตัวกำหนดลอจิก “0“ หรือ “1“ โดยถ้าความกว้างของช่องว่างของพัลส์แคบ (ประมาณ 550 ไมโครวินาที) และความกว้างของพัลส์ยาว (ประมาณ 1,100 ไมโครวินาที) จะสื่อความหมายว่าเป็นลอจิก “1“ แต่ถ้าความกว้างของช่องว่างยาว และความกว้างของพัลส์แคบจะแทนความหมายว่าเป็นลอจิก “0“



รูปที่ 2.9 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ช่องการเลื่อนของพัลส์

2.1.6 รูปแบบการส่งสัญญาณ (Protocol)

ที่ผ่านมาเป็นการอธิบายถึงทฤษฎีพื้นฐานรีโมทคอนโทรลอินฟราเรดแต่ไม่ได้กล่าวถึง โปรโตคอลที่ใช้ในการสื่อสารระหว่างตัวส่งสัญญาณและตัวรับสัญญาณ โปรโตคอลโดยมากถูกออกแบบโดยหลายบริษัทผู้ผลิตสินค้า เนื่องจากผู้ผลิตรีโมทคอนโทรลมีอยู่หลายบริษัท ซึ่งจะทำให้การเข้ารหัสของรีโมทแต่ละยี่ห้อไม่เหมือนกัน แต่จะอยู่ในพื้นฐานเดียวกันคือ จะใช้รหัสเลขฐานสองและความกว้างความแคบของพัลส์ในช่วง “HIGH“ และในช่วง “LOW“ เป็นตัวกำหนดการทำงาน

ในที่นี้ยกตัวอย่างโปรโตคอลบางตัว คือ

- Sharp Protocol
- Sony SIRC Protocol
- Philips RC5 Protocol
- Nokia KRC17 Protocol

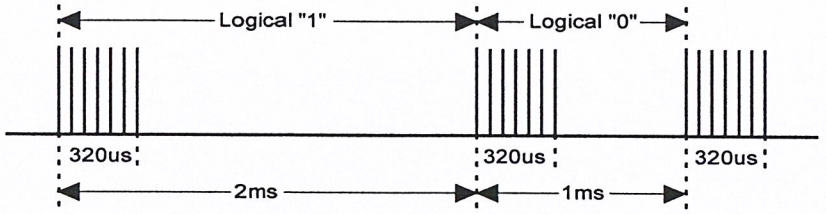
2.1.6.1 Sharp Protocol

ใช้ในเครื่องใช้จำพวก VCR ที่ผลิตโดยบริษัท Sharp

ลักษณะเด่น (Fearures)

- ความยาว 8 bit command, 5 bit address
- Pulse distance modulation
- ค่าความถี่พาหะ 38kHz
- ข้อมูล 1 bitใช้เวลา 1 ms หรือ 2 ms

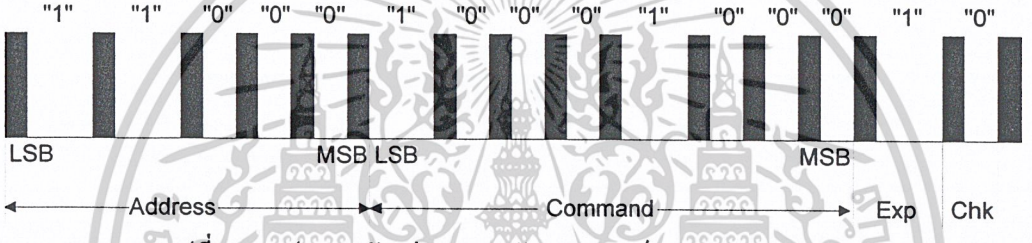
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 รูปค่าลอจิกของ Sharp protocol

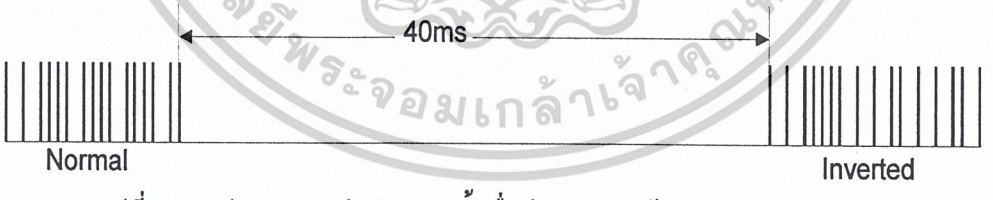
Protocol

โปรโตคอล Sharp ใช้การเข้ารหัส pulse distance ของข้อมูล ในแต่ละพัลส์ กว้าง 320 μs มีค่าความถี่ของพาหะนำสัญญาณอินฟราเรด 38kHz (ประมาณ 12 cycle) ค่าลอจิก "1" ใช้เวลา 2 ms ในขณะที่ลอจิก "0" ใช้เวลาเพียง 1 ms เท่านั้น ค่า duty-cycle ของพาหะที่แนะนำอยู่ที่ประมาณ 1/4 หรือ 1/3



รูปที่ 2.11 รูปแสดงตัวอย่างขบวนสัญญาณพัลส์ของ Sharp Protocol

ภาพนี้แสดงตัวอย่างขบวนสัญญาณพัลส์ที่ส่งคำสั่ง 00010001B(ขนาด 8 bit) และaddress 00011B (ขนาด 5 bit) โดย address ถูกส่งออกไปเป็นอันดับแรกตามด้วยคำสั่งและทั้ง 2 อย่างจะส่งบิต LSB ออกไปก่อน (ส่งแบบโดยตรง)



รูปที่ 2.12 รูปแสดงการส่งสัญญาณซ้ำเมื่อมีการกดปุ่มค้างของ Sharp Protocol

คำสั่งที่สมบูรณ์ตอนหนึ่งๆประกอบด้วย 2 ข้อความ การส่งแบบโดยตรงได้อธิบายไว้แล้วข้างต้น การส่งครั้งที่ 2 (ตามหลังครั้งแรกห่างกัน 40 ms) ข้อมูลเหมือนเดิม แต่แตกต่างกันที่ทุกบิตที่รับมาใน address ถูกกลับค่า $(0 \rightarrow 1, 1 \rightarrow 0)$ ดังนั้นตัวรับจึงสามารถตรวจสอบค่าได้ว่าข้อความที่รับมาเชื่อถือได้หรือไม่

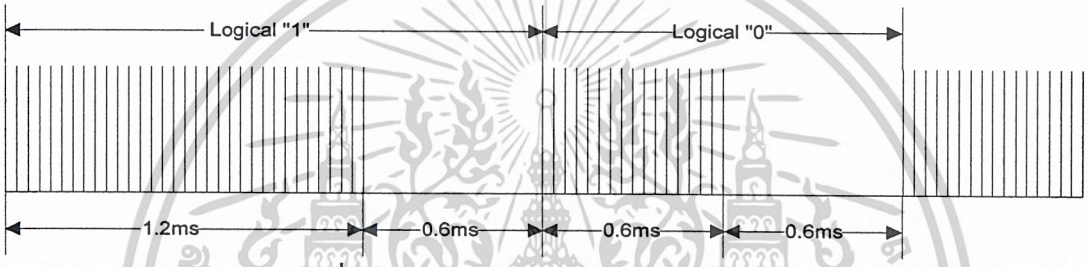
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6.2 Sony SIRC Protocol

ใช้ในเครื่องใช้ที่ผลิตโดยบริษัท Sony

ลักษณะเด่น (Features)

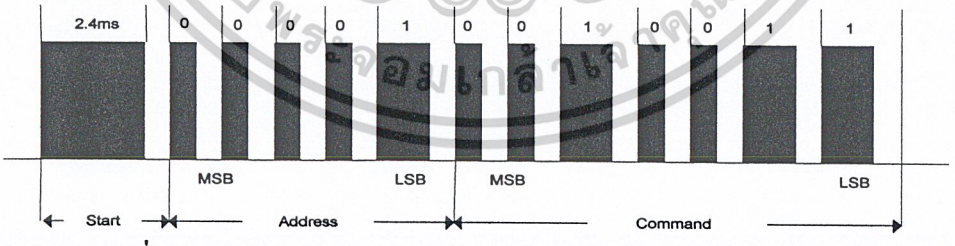
- Address ยาว 5 บิตและคำสั่งยาว 7 บิต (โปรโตคอล 12 บิต)
- Pulse width modulation
- ความถี่พาหะ 40 kHz
- ข้อมูล 1 บิตใช้เวลา 1.8 ms หรือ 1.2 ms



รูปที่ 2.13 รูปค่าลอจิกของ Sony SIRC Protocol

Protocol

โปรโตคอล SIRC ใช้การเข้ารหัส Pulse width กับบิตใด ๆ ในข้อมูล สัญญาณพัลส์ที่แสดง ค่าลอจิก "1" พาหะนำสัญญาณอินฟราเรดกว้าง 1.2 ms 40 kHz ในขณะที่ค่าลอจิก "0" กว้าง 0.6 ms แต่ละช่วงของการส่งสัญญาณเว้น 0.6 ms ค่า Duty - cycle ของพาหะนำสัญญาณอินฟราเรดที่แนะนำอยู่ที่ 1/4 หรือ 1/3



รูปที่ 2.14 รูปแสดงตัวอย่างขบวนสัญญาณของพัลส์ Sony SIRC Protocol

ภาพด้านบนแสดงตัวอย่างขบวนสัญญาณพัลส์ของโปรโตคอล SIRC โปรโตคอลนี้จะส่ง MSB ออกไปอันดับแรก บิตเริ่มต้นมักจะกว้าง 2.4 ms ตามด้วยช่องว่างขนาดมาตรฐานกว้าง 0.6 ms เป็นการแยกส่วนเริ่มออกจากสัญญาณข้อมูล หลังจากนั้นจึงส่ง Address 5 บิต ตามด้วยคำสั่ง 7 บิต ในกรณีนี้ Address 00001B และคำสั่ง 0010011B โดยคำสั่งทุกๆ 45 ms (วัดจากจุดเริ่มถึงจุดเริ่ม) เมื่อกดคีย์ของรีโมทค้างไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างค่าคำสั่ง

ตารางที่ 2.1 แสดงข้อความที่รีโมทคอนโทรลของ Sony ส่งออกมาในโปรโตคอลแบบ 12 บิต

Address	Device
1	TV
2	VCR 1
3	VCR 2
6	Laser Disc Unit
12	Surround Sound
16	Cassete deck / Tuner
17	CD Player
18	Equaliser

Command	Command
0 – 9	Digit keys 0 – 9
16	Channel +
17	Channel -
18	Volume +
19	Volume -
20	Mute
21	Power
22	Reset
23	Audio Mode
24	Contrast +
25	Contrast -
26	Colour +
27	Colour -
30	Brightness +
31	Brightness -
38	Balance Left
39	Balance Right
47	Standly

ตารางที่ 2.1 ค่าสัญญาณ Address และ Command ของ Sony SIRC Protocol

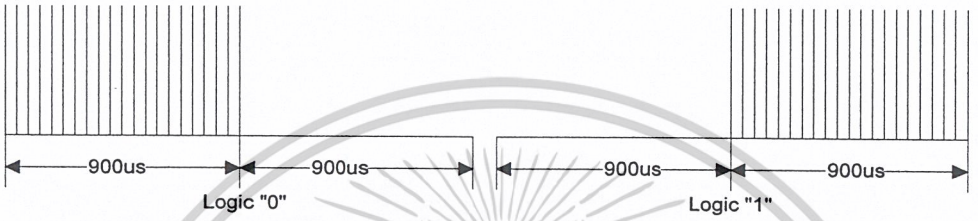
2.1.6.3 Philips RC-5 Protocol

รหัส RC5 จากบริษัท Philips ค่อนข้างเป็นโปรโตคอลที่ใช้อย่างแพร่หลายในหมู่นักประดิษฐ์ อาจเป็นเพราะเป็นรีโมทคอนโทรลที่หาได้ง่ายและราคาถูก โปรโตคอลนี้กำหนดให้ทำงานสำหรับอุปกรณ์ต่างชนิดกัน ทำให้แน่ใจว่าจะเข้าได้กับระบบเพื่อความบันเทิงของคุณได้ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะเด่น (Features)

- Address ขนาด 5 บิตและคำสั่งขนาด 6 บิต
- Bi - phase coding (Aka Manchester coding)
- ความถี่พาหะ 38 kHz
- ข้อมูล 1 บิตใช้เวลา 1.8 ms

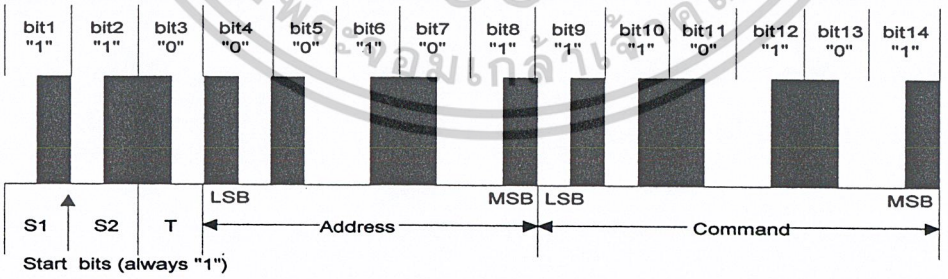


รูปที่ 2.15 รูปค่าลอจิกของ Philips RC-5 Protocol

โปรโตคอลนี้ใช้ Bi - phase modulation หรือที่เรียกว่า Manchester coding ความถี่พาหะของอินฟราเรด คือ 38 kHz ทุกบิตกว้างเท่ากันหมด คือ 1.8 ms ในโปรโตคอลนี้ ครึ่งหนึ่งของเวลาใน 1 บิต ประกอบด้วยการส่งสัญญาณโดยความถี่พาหะ 38 kHz และ อีกครึ่งหนึ่งปล่อยไว้เฉยๆ ค่าลอจิก “0” แสดงการส่งข้อมูลในช่วงเวลาของครึ่งบิตแรก ค่าลอจิก “1” แสดงการส่งสัญญาณในช่วงเวลาครึ่งบิตที่สอง โดยอัตราของ pulse/pause ratio เท่ากับ 38 kHz ความถี่พาหะ 1/3 หรือ ¼ (เพื่อลดกำลัง)

Protocol

ดังรูปสัญญาณที่แสดงตัวอย่างขบวนพัลส์ ของสัญญาณ RC5 โดยตัวอย่างนี้ส่งคำสั่ง 101011B ไปที่ address 10100B



รูปที่ 2.16 รูปแสดงตัวอย่างขบวนสัญญาณพัลส์ของ Philips RC-5 Protocol

พัลส์ 2 ลูกแรกเป็นพัลส์เริ่มต้น และ มีค่าลอจิก “1” ทั้งคู่ ครึ่งช่วงเวลาบิตที่สองปล่อยว่างไว้ก่อนที่ตัวรับจะรับรู้ว่าเริ่มมีข้อความส่งมา การขยายการใช้งานของ RC5 ใช้เฉพาะ บิตเริ่ม 1 บิต, บิต S2 จะถูกแปลงคำสั่งให้อยู่ในรูป คำสั่ง 6 บิต ทำให้บิตคำสั่งมีทั้งหมด 7 บิต บิตที่ 3 คือ บิตทอกเกิล บิตนี้จะกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าทุกครั้งที่คีย์ถูกกด วิธีนี้จะทำให้ตัวรับรับรู้ว่าคีย์ยังกดอยู่หรือกดซ้ำ บิตถัดไป 5 บิต แสดง Address ของอุปกรณ์อินฟราเรดซึ่งส่ง LSB ก่อน จึงตามด้วยคำสั่ง 6 บิต ข้อความประกอบด้วย 14 บิต ซึ่งอาจกินเวลาถึง 25.2 ms บางครั้งข้อความอาจสั้นลงเพราะครึ่งช่วงแรกของเวลา 1 บิตในบิตเริ่ม S1 ยังปล่อยว่างอยู่และถ้าบิตสุดท้ายของข้อความเป็นลอจิก “0” ครึ่งช่วงท้ายของเวลา 1 บิตในบิตสุดท้ายจะถูกปล่อยด้วย ตราบใดที่คีย์ถูกกดค้าง ข้อความจะซ้ำในเวลา 114 ms บิตทอกเกิลจะยังเหมือนเดิมในช่วงนี้ ขึ้นอยู่กับ Software ของตัวรับว่าจะรับรู้ว่ามีการส่งลักษณะซ้ำๆแบบนี้ได้หรือไม่

ตัวอย่างค่าคำสั่ง

RC5 Address	Device
$00 – 0	TV1
$01 – 1	TV2
$02 – 2	Teletext
$03 – 3	Video
$04 – 4	LV1
$05 – 5	VCR1
$06 – 6	VCR2
$07 – 7	Experimental
$08 – 8	Sat1
$09 – 9	Camera
$0A – 10	Sat2
$0B – 11	
$0C – 12	CDV
$0D – 13	Camcorder
$0E – 14	
$0F – 15	
$10 – 16	Pre-amp
$11 – 17	Tuner
$12 – 18	Recorder1
$13 – 19	Pre-amp
$14 – 20	CD Player
$15 – 21	Phono
$16 – 22	SatA
$17 – 23	Recorder2
$18 – 24	
$19 – 25	
$1A – 26	CDR
$1B – 27	
$1C – 28	
$1D – 29	Lighting
$1E – 30	Lighting
$1F – 31	Phone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RC5 Command	TV Command	VCR Command
$00 – 0	1	1
$01 – 1	2	2
$02 – 2	2	2
$03 – 3	3	3
$04 – 4	4	4
$05 – 5	5	5
$06 – 6	6	6
$07 – 7	7	7
$08 – 8	8	8
$09 – 9	9	9
$00 – 12	Standby	Standby

RC5 Command	TV Command	VCR Command
$10 – 16	Volume +	
$11 – 17	Volume -	
$12 – 18	Brightness +	
$13 – 19	Brightness -	
$32 – 50		Fast Rewind
$34 – 52		Fast Forward
$35 – 53		Play
$36 – 54		Stop
$37 - 55		Recording

ตารางที่ 2.2 คำสัญญาณ Address และ Command ของ Philips RC-5 Protocol

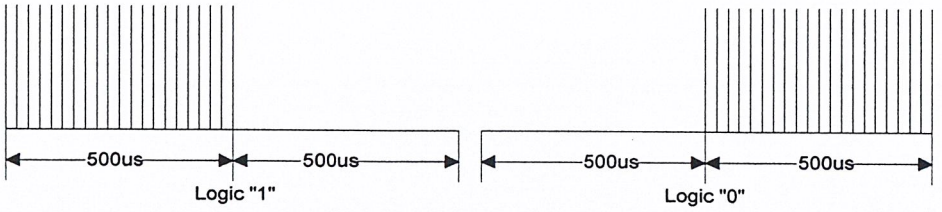
2.1.6.4 Nokia NRC17 Protocol

รีโมทคอนโทรลที่ใช้โปรโตคอล Nokia ใช้การส่งคำสั่งทางสัญญาณอินฟราเรด 17 บิต โปรโตคอลนี้ถูกออกแบบมาเพื่อผู้ใช้สินค้าอิเล็กทรอนิกส์ของ Nokia ถูกใช้เมื่อไม่กี่ปีที่ผ่านซึ่งทางบริษัท Nokia ได้ผลิตชุด TV และ VCR เช่นเดียวกันกับยี่ห้อ Finlux และ Salora ก็ใช้โปรโตคอลนี้เช่นกัน ในทุกวันนี้โปรโตคอลนี้ใช้มากในตัวรับดาวเทียมของ Nokia

ลักษณะเด่น (Features)

- คำสั่งขนาด 8 บิต, Address ขนาด 4 บิตและ Subcode ขนาด 4 บิต
- Bi - phase coding
- ความถี่พาหะ 38 kHz
- ข้อมูล 1 บิตใช้เวลา 1 ms
- บริษัทผู้ผลิต : Nokia CE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

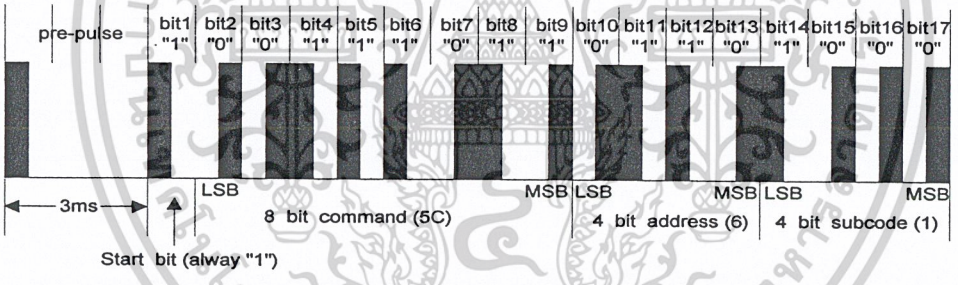


รูปที่ 2.17 รูปค่าลอจิกของ Nokia NRC-17 Protocol

โปรโตคอลนี้ใช้ Bi - phase (หรือที่เรียกว่า "NRZ" Non Return to Zero) modulation ของพาหะของสัญญาณอินฟราเรดมีความถี่ 38 kHz โปรโตคอลนี้ทุกบิตยาว 1 ms โดยช่วงครึ่งหนึ่งของเวลาที่ใช้ใน 1 บิตประกอบด้วยพาหะความถี่ 38 kHz และอีกครึ่งหนึ่งปล่อยว่างไว้ ค่าลอจิก "1" แสดงถึงการส่งสัญญาณในช่วงครึ่งเวลาแรกของเวลาใน 1 บิต ค่าลอจิก "0" แสดงถึงการส่งสัญญาณในช่วงครึ่งที่ 2 ของเวลาใน 1 บิต ค่าอัตรา pulse/pause ของความถี่พาหะ 38 kHz คือ 1/4 (เพื่อเป็นการลดการใช้กำลังงาน)

Protocol

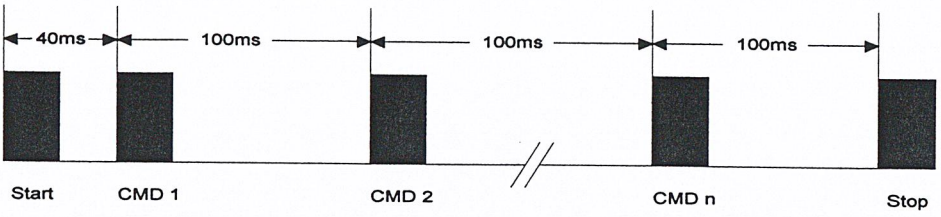
ภาพข้างใต้แสดงตัวอย่างของขบวนสัญญาณพัลส์ของข้อความในโปรโตคอล NRC17 ตัวอย่างนี้ส่งข้อมูล 01011100B ไป Address 0110B Subcode 0001B



รูปที่ 2.18 รูปแสดงตัวอย่างขบวนสัญญาณพัลส์ของ Nokia NRC-17 Protocol

พัลส์ลูกแรกเรียกว่า Pre-pulse สร้างจากการส่งสัญญาณกว้าง 500 μs ตามด้วยการหยุดส่งอีก 2.5 ms ใช้เวลาทั้งหมดเท่ากับเวลาในข้อมูล 3 บิต หลังจากนั้นบิตเริ่มต้นจะถูกส่งออกไปซึ่งมักจะเป็นลอจิก "1" ซึ่งสัญญาณพัลส์นี้สามารถใช้เป็นตัวเทียบวัดเวลาใน 1 บิตทางด้านตัวรับได้ เพราะเวลาส่งสัญญาณจริงๆใช้แค่ครึ่งหนึ่งของเวลาใน 1 บิต 8 บิตถัดไปแสดงคำสั่งของสัญญาณอินฟราเรดส่ง LSB ก่อน ตามด้วย Address ของอุปกรณ์อีก 4 บิตสุดท้ายตามด้วย Subcode 4 บิต ซึ่งจะเห็นได้จากส่วนที่ต่อจากบิต Address ข้อความประกอบด้วย Pre-pulse ขนาด 3 ms และ 17 บิตในแต่ละ 1 ms (จึงใช้เวลาทั้งหมด 20 ms ต่อการส่ง 1 ข้อความ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 รูปแสดงการส่งสัญญาณซ้ำเมื่อกดปุ่มค้างของ Nokia NRC-17 Protocol

ทุกครั้งที่คีย์ของรีโมทคอนโทรลถูกกด บิตเริ่มที่ถูกส่งออกไปประกอบด้วย คำสั่ง 11111110B และ Address/Subcode 11111111B ข้อความจริงๆถูกส่งใน 40 ms ต่อมาและส่งซ้ำอีกในทุกๆ 100 ms ถ้าคีย์ของรีโมทคอนโทรลยังคงถูกกดอยู่ เมื่อปล่อยคีย์ ข้อความหยุดถูกปล่อยออกมาว่าเป็นการจบชุดข้อความ ข้อความหยุดยังคงใช้คำสั่ง 11111110B และ Address/Subcode 11111111B ชุดคำสั่งทุกชุดสามารถทำให้เป็นชุดคำสั่งเดียวได้ที่ท้ายตัวรับ เนื่องจากข้อความเริ่มและข้อความหยุด ทำให้คีย์ที่กดโดยบังเอิญจะไม่มีผลอะไร ตัวรับอาจจะตัดสินใจที่จะเชื่อฟังข้อความที่ซ้ำมาหรือไม่ก็ได้ ยกตัวอย่างเช่น การเลื่อนเคอร์เซอร์อาจทำซ้ำๆเท่าที่คีย์ถูกกด ค่าอินพุตที่เป็นตัวเลขจะดีกว่าตรงที่ไม่รับการทำซ้ำเรื่อยๆ

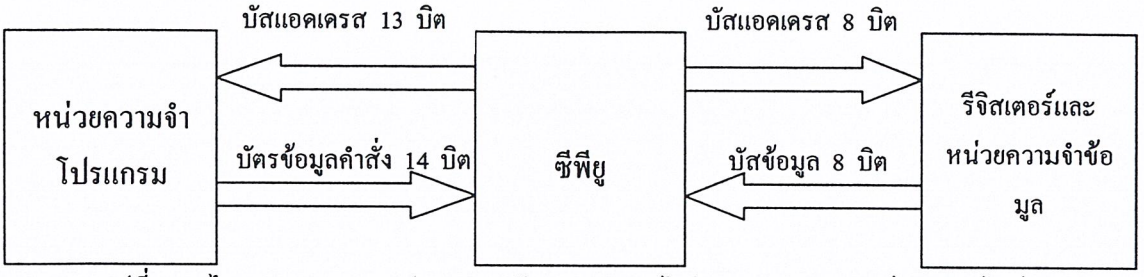
2.2 ทฤษฎีไมโครคอนโทรลเลอร์

2.2.1 โครงสร้างทางฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ตระกูล PIC 16F877

ไมโครคอนโทรลเลอร์ตระกูล PIC มีสถาปัตยกรรมแบบฮาร์วาร์ด (Harvard architecture) กล่าวคือมีการแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกัน โดยมีบัสสำหรับติดต่อแยกกันด้วย ดังแสดงในรูป 2.20 จะเห็นว่าซีพียู่ภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมด้วยบัสแอดเดรส 13 บิต และบัสข้อมูลหน่วยความจำโปรแกรม 14 บิต ในขณะที่บัสสำหรับติดต่อกับหน่วยความจำข้อมูลและรีจิสเตอร์ภายในเป็นแบบ 8 บิตทั้งบัสแอดเดรสและบัสข้อมูล

นอกจากการจัดสถาปัตยกรรมแบบนี้แล้ว การกระทำคำสั่งของไมโครคอนโทรลเลอร์ PIC ยังใช้กระบวนการที่เรียกว่า ไปป์ไลน์ (Pipeline) ทำให้สามารถเฟตช์คำสั่งถัดไป ในขณะที่กำลังเอ็คซีคิวต์ทำสั่งในปัจจุบัน เป็นผลให้ความเร็วในการทำงานของไมโครคอนโทรลเลอร์นั้นเพิ่มมากขึ้น นั่นเป็นที่มาของความสามารถในการทำคำสั่ง 1 คำสั่งในสัญญาณนาฬิกา 1 ลูก (เฟตช์ : fetch เป็นกระบวนการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วแปลเป็นเลขฐานสิบหกเพื่อให้ซีพียูเข้าใจ ส่วนกระบวนการเอ็คซิคิวต์ (Execute) เป็นการกระทำคำสั่งให้เกิดผลลัพธ์ตามที่คำสั่งนั้นๆกำหนด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 ไดอะแกรมแสดงรูปแบบสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด

ไมโครคอนโทรลเลอร์ PIC16F87x มีรีจิสเตอร์พิเศษตัวหนึ่งที่บรรจุข้อมูลสำหรับกำหนดการทำงานทั้งหมดเอาไว้ คือ Configuration word โดยภายใน Configuration word จะบรรจุข้อมูลของการเลือกป้องกันการอ่านข้อมูล , เลือกความสามารถการรีเซตอัตโนมัติเมื่อไฟเลี้ยงลดต่ำถึงค่าที่กำหนด , ควบคุมการทำงานของวอตช์ด็อกไทเมอร์ หรือกระทั่งการเลือกชนิดของวงจรกำเนิดสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ การกำหนดข้อมูลสำหรับรีจิสเตอร์ตัวนี้สามารถกระทำได้ 2 ทางคือ ด้วยคำสั่ง _CONIG ในส่วนต้นของโปรแกรมภาษาแอสเซมบลี แล้วแอสเซมเบลอร์ด้วย MPASM ซึ่งบรรจุอยู่ในชุดของโปรแกรม MPLAB อันเป็น Software ในการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ PIC ทางที่ 2 คือกำหนด Software ที่ใช้ในการโปรแกรมหน่วยความจำของไมโครคอนโทรลเลอร์ การกำหนดสามารถกระทำในทางใดทางหนึ่งหรือทั้งสองทางก็ได้ แต่ถ้าการกำหนดทั้ง 2 ทางแตกต่างกัน การกำหนดที่ Software ของเครื่องโปรแกรมจะมีนัยสำคัญที่สูงกว่า

ชื่อขา	ตำแหน่งขา	ชนิดของขา	ชนิดของวงจรบัฟเฟอร์	รายละเอียดการทำงาน
OSC1/CLKIN	9 (13)	อินพุต	ชมิตต์ทริกเกอร์/ซีมอส	-ขาต่อคริสตอล/รับสัญญาณนาฬิกาจากภายนอก
OSC2/CLKOUT	10 (14)	เอาต์พุต	-	
$\overline{\text{MCLR}}$ /VPP	1	อินพุต	ชมิตต์ทริกเกอร์	-ขาต่อคริสตอล/ในโหมด RC เป็นขาเอาต์พุต -สัญญาณนาฬิกาความถี่¼ของสัญญาณที่ขาosc1
ขาพอร์ต A เป็นขาพอร์ต 2 ทิศทาง				
RA0/AN0	2	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RA0 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 0
RA1/AN1	3	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RA1 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 1
RA2/AN2/V_{REF}/ CV_{REF}*	4	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RA2 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 2 -อินพุตแรงดันอ้างอิงลบของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล -เอาต์พุตของแรงดันอ้างอิงของโมดูลแรงดันอ้างอิง (PIC16F877)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อขา	ตำแหน่งขา	ชนิดของขา	ชนิดของวงจรบัฟเฟอร์	รายละเอียดการทำงาน
RA3/AN3/V_{REF+}	5	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RA3 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 3 -อินพุตแรงดันอ้างอิงบวกของวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล
RA4/TOCK1/ C1OUT*	6	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RA4 กรณีใช้พอร์ตเอาต์พุตมีโครงสร้างแบบเดรนเปิด -อินพุตสัญญาณนาฬิกาของไทเมอร์ 0 -เอาต์พุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 1 (PIC16F87xA)
RA5/AN4/ $\overline{SS}$ / C2OUT*	7	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RA5 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 5 -ขาสัญญาณ Slave Select ใช้ในการสื่อสารข้อมูลอนุกรมแบบซิงโครนัส -เอาต์พุตวงจรเปรียบเทียบแรงดันอนาลอกช่อง 2 (PIC16F87xA)
ขาพอร์ต B เป็นขาพอร์ต 2 ทิศทาง สามารถกำหนดให้ต่อตัวต้านทานพูลอัปภายในเมื่อทำงานเป็นอินพุตได้ทางซอฟแวร์				
RB0/INT	21 (33)	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขอพอร์ต RA5 -อินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอก
RB1	22 (33)	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RB1
RB2	23 (35)	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RB2
RB3/LVP	24 (36)	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RB3 -อินพุตรับแรงดันโปรแกรมต่ำ (+5v) ถ้าเอ็นเอเบิลไว้
RB4	25 (37)	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RB4 และสามารถเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงลอจิกขึ้นที่ขานี้ หากเอ็นเอเบิลไว้
RB5	26 (38)	อินพุต/เอาต์พุต	ทีทีแอล/อนาลอก	-ขาพอร์ต RB5 และสามารถเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงลอจิกขึ้นที่ขานี้ หากเอ็นเอเบิลไว้
RB6/PGC	27 (39)	อินพุต/เอาต์พุต	ทีทีแอล/ชมิตต์ทริกเกอร์[2]	-ขาพอร์ต RB6 -เป็นขาสัญญาณนาฬิกาของการดีบักในวงจร(ICD) -สามารถเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงลอจิกขึ้นที่ขานี้ หากเอ็นเอเบิลไว้
RB7/PGD	28 (40)	อินพุต/เอาต์พุต	ทีทีแอล/ชมิตต์ทริกเกอร์[2]	-ขาพอร์ต RB6 -เป็นขาสัญญาณนาฬิกาของการดีบักในวงจร(ICD) -สามารถเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงลอจิกขึ้นที่ขานี้ หากเอ็นเอเบิลไว้
ขาพอร์ต C เป็นขาพอร์ต 2 ทิศทาง				
RC0/T1OSO/ T1CK1	11 (16)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC0 -เอาต์พุตวงจรออสซิลเลเตอร์ของไทเอมร์ -อินพุตของสัญญาณนาฬิกาของไทเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ชื่อขา	ตำแหน่งขา	ชนิดของขา	ชนิดของวงจรบัฟเฟอร์	รายละเอียดการทำงาน
RC1/TLOS1/ CCP2	12 (16)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC1 -อินพุตวงจรออสซิเลเตอร์ของไทเมอร์ -อินพุตวงจรแคปเจอร์เอาต์พุตวงจรเปรียบเทียบเอาต์พุต PWM สำหรับโมดูล CCP
RC2/CCP1	13 (17)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC2 -อินพุตวงจรแคปเจอร์/เอาต์พุตวงจรเปรียบเทียบเอาต์พุต PWM สำหรับโมดูล CCP
RC3/SCK/SCL	14 (18)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC3 -ขาสัญญาณนาฬิกาของวงจร SPI และระบบบัส I^2C
RC4/SDI/SDA	15 (23)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC4 -ขาข้อมูลอินพุตวงจร SPI -ขาข้อมูลอนุกรมของระบบบัส I^2C
RC5/SDO	16 (24)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC5 -ขาข้อมูลเอาต์พุตของวงจร SPI
RC6/TxD	17 (25)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC6 -ขาเอาต์พุตวงจร USART สำหรับเชื่อมพอร์ตอนุกรม
RC7/RxD	18 (26)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์	-ขาพอร์ต RC7 -ขาอินพุตวงจร USART สำหรับเชื่อมต่อพอร์ตอนุกรม
ขาพอร์ต D เป็นขาพอร์ต 2 ทิศทาง สามารถใช้เป็นส่วนขยายพอร์ตแบบขนานเพื่อติดต่อกับระบบบัสอื่น ไม่มีใน PIC16F873(A)/876(A)				
RD0/PSP0	(19)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD0 -ขาขยายพอร์ตแบบขนานบิต 0
RD1/PSP1	(20)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD1 -ขาขยายพอร์ตแบบขนานบิต 1
RD2/PSP2	(21)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD2 -ขาขยายพอร์ตแบบขนานบิต 2
RD3/PSP3	(22)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD3 -ขาขยายพอร์ตแบบขนานบิต 3
RD4/PSP4	(27)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD4 -ขาขยายพอร์ตแบบขนานบิต 4
RD5/PSP5	(28)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD5 -ขาขยายพอร์ตแบบขนานบิต 5
RD6/PSP6	(29)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD6 -ขาขยายพอร์ตแบบขนานบิต 6
RD7/PSP7	(30)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล	-ขาพอร์ต RD7 -ขาขยายพอร์ตแบบขนานบิต 7
ขาพอร์ต E เป็นขาพอร์ต 2 ทิศทางไม่มีใน PIC16F873(A)/876(A)				
RE0/AN5/ $\overline{RD}$	(8)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล(3)	-ขาพอร์ต RE0 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

61946

ชื่อขา	ตำแหน่งขา	ชนิดของขา	ชนิดของวงจรบัฟเฟอร์	รายละเอียดการทำงาน
				-ขาสัญญาณ $\overline{RD}$ สำหรับส่วนขยายพอร์ตแบบขนาน
RE1/AN6/ $\overline{WR}$	(9)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล(3)	-ขาพอร์ต RE1 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 6 -ขาสัญญาณ $\overline{WR}$ สำหรับส่วนขยายพอร์ตแบบ-ขนาน
RE2/AN7/ $\overline{CS}$	(10)	อินพุต/เอาต์พุต	ชมิตต์ทริกเกอร์/ทีทีแอล(3)	-ขาพอร์ต RE2 -อินพุตวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลช่อง 7 -ขาสัญญาณ $\overline{CS}$ สำหรับส่วนขยายพอร์ตแบบขนาน
ขาต่อไฟเลี้ยง				
V_{dd}	20 (11,32)	อินพุต	-	-ขาต่อไฟเลี้ยงใช้ได้ตั้งแต่ +2 ถึง +5.5v
V_{ss}	8,19 (12,31)	อินพุต	-	-ขาต่อกราวด์

ตารางที่ 2.3 ตารางสรุปการทำงานของขาพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ PIC16F877

2.2.2 การจัดสรรหน่วยความจำโปรแกรม

หน่วยความจำโปรแกรม (Program memory) เป็นส่วนสำคัญมากเพราะเป็นที่เก็บข้อมูลคำสั่งทั้งหมด ซึ่งใช้ในการกำหนดให้ไมโครคอนโทรลเลอร์ทำงาน หน่วยความจำโปรแกรมของ PIC16F877 เป็นแบบแฟลช (flash memory) ทำให้สามารถลบ และเขียนใหม่ได้แสนครั้ง แต่อย่างไรก็ตามโดยปกติหน่วยความจำโปรแกรม หลังจากทำการเขียนในขั้นตอนของการโปรแกรมแล้วก็จะมีไว้สำหรับอ่านออกมาได้เพียงทางเดียว

PIC16F877 มีโปรแกรมเคาน์เตอร์ (PC) ขนาด 13 บิต เพื่อกำหนดการเข้าถึงหน่วยความจำโปรแกรม มีขนาด 8 k 14 บิต (หรือ 8 กิโลเวิร์ด) เนื่องจากไมโครคอนโทรลเลอร์ ในอนุกรมนี้มีขนาดหน่วยความจำที่ค่อนข้างใหญ่มากจึงต้องจัดสรรเป็นเพจ (page) หรือเป็นหน้า โดยแต่ละเพจจะมีขนาด 2 กิโลเวิร์ด ทั้งนี้เนื่องจากชุดคำสั่งเกี่ยวกับการกระโดดของไมโครคอนโทรลเลอร์ ตระกูล PIC สามารถอ้างถึงหน่วยความจำได้สูงสุด 2,048 ตำแหน่ง

สำหรับ PIC16F877 ซึ่งมีขนาดหน่วยความจำโปรแกรม 8 กิโลเวิร์ด มีการสงวนแอดเดรส 0x0000 และ 0x0004 ไว้เช่นกัน หรืออาจกล่าวได้ว่าเป็นรูปแบบมาตรฐานของการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ สำหรับ PIC16F877 มีการแบ่งหน่วยความจำออกเป็น 4 เพจ ดังนี้

เพจ 0 มีแอดเดรสในช่วง 0x0000-0x07FF (โดยควรสงวนแอดเดรส 0x0000 และ 0x0004 ไว้)

เพจ 1 มีแอดเดรสในช่วง 0x0800-0x0FFF

เพจ 2 มีแอดเดรสในช่วง 0x1000-0x17FF

เพจ 3 มีแอดเดรสในช่วง 0x1800-0x1FFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนั้นใน PIC16F877 ยังมีพื้นที่หน่วยความจำพิเศษสำหรับเก็บค่าของโปรแกรมชั่วคราว ขนาด 13 บิต เรียกว่า สแต็ก(Stack) ซึ่งมีบทบาทมากสำหรับการกระโดดไปทำงานยังโปรแกรมย่อยของ PIC16F877 โดยเมื่อกระทำคำสั่งให้กระโดดไปทำงานยังโปรแกรมย่อย ซีพียูจะทำการเก็บค่าโปรแกรมเคาร์เตอร์หรือ PC ในขณะนั้นไว้ในสแต็ก จากนั้นกระโดดไปทำงานยังโปรแกรมย่อยเมื่อทำงานเรียบร้อยแล้ว ซีพียูจะไปอ่านค่า PC จากสแต็กกลับมา แล้วทำงานตามกระบวนการในโปรแกรมหลักต่อไปสำหรับสแต็กใน PIC16F877 มีขนาด 13 บิต สามารถเก็บค่าของ PC ได้ 8 ระดับ

2.2.3 การจัดสรรหน่วยความจำข้อมูลแรมและรีจิสเตอร์ไฟล์

ใน PIC16F877 มีหน่วยความจำข้อมูลแรมสำหรับใช้งานทั่วไป 368 ไบต์ และรีจิสเตอร์ไฟล์ 8 บิต 57 ตัว แต่ละแบงค์มีขนาดสูงสุด 128 ไบต์ แต่มีการใช้งานได้จริงในแต่ละแบงค์ต่างกัน โดยในแต่ละแบงค์มีการจัดสรรพื้นที่ดังนี้

แบงค์ 0 มีช่วงแอดเดรส 0x00-0x7F

แอดเดรส 0x00-0x1F เป็นพื้นที่ของรีจิสเตอร์ไฟล์

แอดเดรส 0x1F-0x7F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 96 ไบต์

แบงค์ 1 มีช่วงแอดเดรส 0x80-0xFF

แอดเดรส 0x80-0x9F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0xA0-0xEF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แบงค์ 2 มีช่วงแอดเดรส 0x100-0x17F

แอดเดรส 0x100-0x10F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0x110-0x11F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 ไบต์

แอดเดรส 0x120-0x16F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แอดเดรส 0x170-0x17F บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบงค์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x70-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบงค์

แบงค์ 3 มีช่วงแอดเดรส 0x190-0x1FF

แอดเดรส 0x180-0x1FF เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0x190-0x19F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 ไบต์

แอดเดรส 0x1A0-0x1AF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แอดเดรส 0x1F0-0x1FF บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบงค์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x70-0x1FF ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบงค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC16F87x

PIC16F87 มีพอร์ตให้ใช้งานตั้งแต่ 3-5 พอร์ต จำนวน 20-30 บิต ขึ้นอยู่กับเบอร์ของ ไมโครคอนโทรลเลอร์ และด้วยความสารถของพอร์ตใน PIC16F877 ที่สามารถทำงานได้หลายอย่าง

ความสามารถในการกระจายกระแสเอาต์พุตของพอร์ตที่ไฟเลี้ยง +5 โวลท์ คือ 25 mA ต่อขาทั้งกระแสซิงก์และกระแสซอร์ซ ในขณะที่กระแสเอาต์พุต รวมของพอร์ต A, B และ E มีค่าสูงสุด 200 mA ส่วนกระแสเอาท์พุตรวมของพอร์ต C และ D มีค่าสูงสุด 200 mA ดังนั้นในการออกแบบเพื่อขับโหลดทางเอาต์พุตรวมของพอร์ตต้องระวังเรื่องเอาต์พุตรวมที่ไมโครคอนโทรลเลอร์สามารถขับได้ด้วย

พอร์ต A

มีทั้งสิ้น 6 ช่องหรือ 6 บิต กำหนดชื่อขาเป็น RAO-RA5 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ Port A มีแอดเดรสอยู่ที่ 0x05 (แบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานจริงเพียง 6 บิต ที่เหลือ 2 บิตต้องกำหนดให้เป็น “0” ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำทำผ่านรีจิสเตอร์ TRISA ซึ่งมีแอดเดรสอยู่ที่ 0x85 (แบงค์ 1) มีขนาด 8 บิต และใช้เพียง 6 บิตเช่นกัน 2 บิตบนคือบิต 6 และบิต 7 ต้องกำหนดให้เป็น “0” บิตของ TRISA ใช้กำหนดทิศทางของ RAO ไล่เรียงลำดับจนถึงบิต 5 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA5 หากต้องการกำหนดให้ขาพอร์ทในบิตเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตนั้น และในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล “0” ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์ของพอร์ต A

พอร์ต A สามารถทำงานเป็นขาพอร์ตอินพุตปกติและเป็นขาอินพุตสัญญาณอนาลอกสำหรับวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลขนาด 10 บิตภายในไมโครคอนโทรลเลอร์ โดยขา RA0-RA3 และ RA5 จะมีการทำงานที่เหมือนกัน ส่วน RA4 จะแตกต่างตรงที่ขานี้นอกจากเป็นขาพอร์ตอินพุตเอาต์พุตปกติแล้วยังใช้เป็นขาอินพุตสำหรับไทเมอร์ 0 ภายในไมโครคอนโทรลเลอร์ด้วย และขา RA4 นี้ไม่สามารถใช้งานเป็นขาอินพุตรับสัญญาณอนาลอกได้

เมื่อขาพอร์ต RA0-RA3 ทำงานเป็นขาพอร์ตอินพุตดิจิตอล จะสามารถรับสัญญาณดิจิตอลระดังทีทีแอล (0-5 โวลท์) ได้โดยตรง หากการทำงานเป็นเอาต์พุตจะสามารถขับโหลดที่ต้องการกระแส 20 mAได้ หากนำมาขับ LED ต้องต่อตัวต้านทานจำกัดกระแส หรือถ้าใช้ไฟเลี้ยง 3 โวลท์ ก็จะขับ LED ได้โดยตรง

พอร์ต B

มี 8 บิต กำหนดชื่อขาเป็น RB0-RB7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTA มีแอดเดรสอยู่ที่ 0x06 (แบงค์ 0) และ 0x106 (แบงค์ 2) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISB ซึ่งมีแอดเดรสอยู่ที่ 0x86 (แบงค์ 1) และ 0x186 (แบงค์ 3) มีขนาดบิต เช่นเดียวกับพอร์ต A บิต 0 ของ TRISB ใช้กำหนดทิศทางของขาพอร์ต RB0 ไล่เรียงลำดับจนถึงบิต 7 ของ TRISB ใช้กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทิศทางของขาพอร์ต RB7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตนั้น ในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล “0” ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์ของพอร์ต B

พอร์ต B สามารถใช้งานในลักษณะต่างๆได้ 5 แบบคือ

1. เป็นขาพอร์ตอินพุตเอาต์พุตปกติ
2. เป็นขาพอร์ตอินพุตสัญญาณอินเตอร์รัปต์จากภายนอก โดยใช้ขา RB0/INT
3. เป็นขาพอร์ตอินพุตสำหรับรับแรงดันโปรแกรมระดับต่ำ (Low voltage programming) โดยใช้ขา RB3/PGM
4. เป็นขาข้อมูลอนุกรมและสัญญาณนาฬิกาอนุกรมสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ ซึ่งใช้ 2 ขา คือ RB7/PGD และ RB6/PGC
5. ใช้เป็นแหล่งกำเนิดสัญญาณอินเตอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงข้อมูลหรือระดับสัญญาณที่ขา RB4-RB7

วงจรอินพุตบัฟเฟอร์ที่ขาพอร์ตนี้มีทั้งแบบทีทีแอลและชมิตต์ทริกเกอร์ ทั้งนี้เพื่อจัดการให้สัญญาณอินพุตที่เข้ามามีความเหมาะสมและสมบูรณ์มากที่สุด และยังคงสามารถรองรับการพูลอัปภายในแบบอัตโนมัติได้

ในกรณีมีการเอ็นเอเบิลตอบสนองอินเตอร์แบบตรวจสอบ การเปลี่ยนแปลงลอจิกที่พอร์ต RB4-RB7 หากเกิดการอินเตอร์รัปต์ขึ้น บิต RBIF (บิต 0 ในรีจิสเตอร์ INTCON) จะเซตและหลังจากการตอบสนองการอินเตอร์รัปต์แล้ว ต้องเคลียร์บิต RBIF ด้วยกระบวนการทางซอฟต์แวร์เสมอ

พอร์ต C

มีทั้งสิ้น 8 บิต กำหนดชื่อขาเป็น RC0-RC7 รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ PORTC มีแอดเดรสอยู่ที่ 0x07(แบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISC มีแอดเดรสอยู่ที่ 0x87 (แบงค์ 1 มีขนาด 8 บิต เช่นเดียวกับพอร์ต A และ B ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC7 ไล่เรียงลำดับจนถึงบิต 7 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุต ให้เขียนข้อมูล “0” ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์ของพอร์ต C

พอร์ต C สามารถใช้งานในลักษณะต่าง ๆ ได้หลายรูปแบบ เป็นขาพอร์ตที่มีความสามารถสูงมาก ไม่ว่าจะเป็นขาพอร์ตอินพุตเอาต์พุตปกติ ขาอินพุตเอาต์พุตออสซิลเลเตอร์ ของโมดูลไทเมอร์ 1, ขาอินพุตสำหรับรับสัญญาณนาฬิกาของโมดูลไทเมอร์ 1, ขาเชื่อมต่อระบบบัส I2C, ขาเชื่อมต่อแบบ SPI(Serial Peripheral Interface), ขาเชื่อมต่ออนุกรมแบบ USART, ขาอินพุตวงจรแคปเจอร์ (capture) หรือวงจรตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จับสัญญาณ ขาเอาต์พุตของวงจรเปรียบเทียบสัญญาณ (compare) และขาเอาต์พุตวงจร PWM (Pulse Width Modulation) หรืออาจกล่าวได้ว่าพอร์ต C เป็นพอร์ตสำหรับเชื่อมต่ออุปกรณ์ภายนอก (Peripheral function port) ที่มีความสมบูรณ์แบบมากที่สุดสามารถสรุปหน้าที่และการทำงานที่หลากหลายของพอร์ต C

พอร์ต D

มี 8 บิต กำหนดชื่อขาเป็น RD0-RD7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORT D มีแอดเดรสอยู่ที่ 0x08 (แบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISD มีแอดเดรสอยู่ที่ 0x088 (แบงค์ 1) มีขนาด 8 บิต หากต้องการกำหนดให้ที่พอร์ตในบิตใดเป็นอินพุต ต้องเขียนข้อมูล “1” ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้ขาเอาต์พุต ให้เขียนข้อมูล “0” ไปยังบิตนั้น สำหรับพอร์ต D นี้จะมีเฉพาะไมโครคอนโทรลเลอร์ PIC รุ่น 40 ขาเท่านั้น สำหรับในอนุกรม PIC19F877

พอร์ต D สามารถใช้งานได้ 2 โหมดคือ เป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นส่วนขยายพอร์ตแบบขนาน (Parallel Slave Port : PSP) สำหรับเชื่อมต่ออุปกรณ์ภายนอกที่มีการจัดระบบบัสแบบไมโครโปร-เซสเซอร์ คือ สายสัญญาณควบคุมการอ่าน (RD : Read) เขียน(WR : Write) และเลือกอุปกรณ์(CS : Chip Select)

พอร์ต D มีโครงสร้างเหมือนกันทุกบิต เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ วงจรอินพุตจะเป็นแบบชมิตต์ทริกเกอร์ แต่เมื่อทำงานในโหมด ขยายพอร์ตแบบขนานหรือ PSP วงจรอินพุตเป็นแบบทีทีแอล

การเลือกโหมดการทำงานของพอร์ต D นี้ขึ้นกับบิต PSPMODE (บิต 4 ในรีจิสเตอร์ TRISE) ถ้าเป็น “0” เป็นการกำหนดให้พอร์ต D เป็นพอร์ตปกติ และถ้าเป็น “1” พอร์ต D จะทำงานในโหมด PSP

พอร์ต E

มี 3 บิตกำหนดชื่อขาเป็น RE0-RE2 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTE มีแอดเดรสอยู่ที่ 0x09 (แบงค์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานเพียง 3 บิตล่างคือ บิต 0 บิต 2 เท่านั้น ที่เหลือกำหนดเป็น “0” ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISE ซึ่งมีแอดเดรสอยู่ที่ 0x89 (แบงค์ 1) มีขนาด 8 บิต โดยใช้ 3 บิตบ่างในการกำหนดทิศทางของพอร์ต E ส่วนที่เหลือใช้ควบคุมการทำงานในโหมด PSP ของพอร์ต D

พอร์ต E สามารถใช้เป็นพอร์ตอินพุตเอาต์พุตปกติ ขาอินพุตอนาลอกจะแปลงสัญญาณอนาลอกเป็นดิจิตอล และขาควบคุมการติดต่อกับอุปกรณ์ภายนอกแบบ PSP ทั้งนี้ขึ้นอยู่กับการกำหนดข้อมูลของรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ตนี้ เช่นเดียวกับพอร์ต D พอร์ต E จะมีเฉพาะในไมโครคอนโทรลเลอร์ PIC รุ่น 40 ขาเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างทางฮาร์ดแวร์ของพอร์ต E

พอร์ต E บิต RE0-RE2 เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ จะเห็นได้ว่าความคล้ายคลึงกับพอร์ตนี้จะเป็นแบบชมิตต์ทริกเกอร์ ในขณะที่หากทำงานในโหมดการแปลงสัญญาณอนาลอกเป็นดิจิตอลวงจรอินพุตจะเปลี่ยนเป็นแบบทีทีแอล

ถึงแม้ว่าพอร์ต E ใน PIC16F877 มีจำนวนน้อยเพียง 3 บิต แต่สามารถเลือกรูปแบบการทำงานได้มากถึง 3 แบบ คือเป็นพอร์ตอินพุตเอาต์พุตปกติ, อินพุตสำหรับวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลขนาด 10 บิตในไมโครคอนโทรลเลอร์ และพอร์ตสัญญาณควบคุมสำหรับการติดต่อกับอุปกรณ์ในโหมด PSP ดังนั้นในการเลือกรูปแบบการทำงานต้องระมัดระวังเช่นเดียวกับพอร์ต C ที่ได้กล่าวมาแล้วก่อนหน้านี้

2.3 การเชื่อมต่อกับเครื่องคอมพิวเตอร์

หากเราต้องการที่จะเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์อื่นๆ หรือคอมพิวเตอร์ด้วยกันมี 2 ทางเลือกนั่นคือ การรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบขนานนั้นเป็นการส่งข้อมูลคราวละ 4 หรือ 8 บิต ในเวลาเดียวกัน ซึ่งจะทำให้การรับส่งข้อมูลทำได้ที่ความเร็วสูง ซึ่งหมายความว่าจำนวนของสายที่ใช้ในการส่งจะต้องมีมากเท่ากับจำนวนบิตที่จะส่งด้วย นอกจากนี้ยังต้องรวมถึงสายที่ใช้สำหรับการควบคุมและการตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายมากขึ้นเป็น 2 เท่าของจำนวนบิตข้อมูลที่ส่งได้

ในขณะที่การรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลคราวละหลาย ๆ บิตได้ แต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่าจะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอรับข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ทำให้การสื่อสารข้อมูลแบบอนุกรมจะใช้จำนวนสายที่น้อยกว่ามากและระยะทางในการส่งข้อมูลได้ไกลกว่า

รูปแบบการสื่อสารข้อมูลแบบอนุกรม

1. แบบซิมเพล็กซ์ (Simplex) ข้อมูลส่งได้ในทางเดียวเท่านั้น การสื่อสารแบบนี้อุปกรณ์สื่อสารด้านหนึ่งจะส่งข้อมูลไปในช่องสัญญาณได้เท่านั้น แต่จะรับข้อมูลจากช่องสัญญาณไม่ได้ และอุปกรณ์สื่อสารอีกด้านหนึ่งจะรับข้อมูลจากช่องสัญญาณได้เท่านั้น แต่ส่งข้อมูลไปในช่องสัญญาณไม่ได้

2. แบบฮาล์ฟดูเพล็กซ์ (Half Duplex) หมายถึง การสื่อสารข้อมูลสองทิศทางโดยอุปกรณ์สื่อสาร ทั้ง 2 ด้านจะผลัดกันรับส่ง การสื่อสารแบบนี้ส่วนใหญ่แล้วจะใช้ระบบสาย 2 เส้น

3. แบบฟลูดูเพล็กซ์ (Full Duplex) หมายถึง การสื่อสารข้อมูล 2 ทิศทางพร้อมกัน การสื่อสารแบบนี้ใช้ทั้งระบบสาย 2 เส้นและสาย 4 เส้น

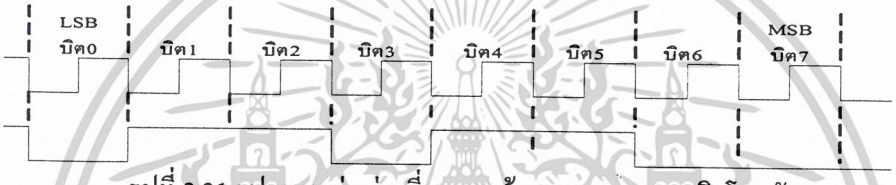
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นเราสามารถแบ่งออกได้เป็น 2 แบบ คือ การสื่อสาร การอนุกรมแบบซิงโครนัส และ การสื่อสารอนุกรมแบบอะซิงโครนัส

2.3.1.1 การสื่อสารอนุกรมแบบซิงโครนัส

การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือ คีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา, ข้อมูล, และกราวนด์ รูปที่ 2.21 แสดงให้เห็นถึงไทม์มิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส



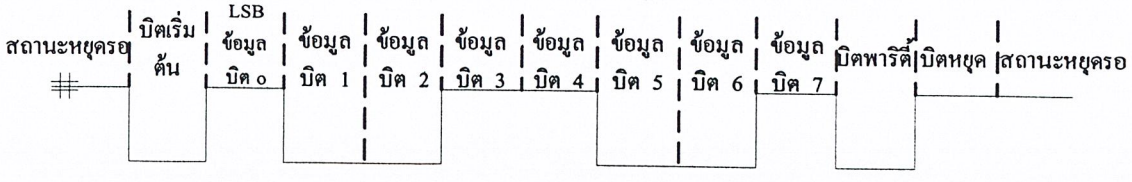
รูปที่ 2.21 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบซิงโครนัส

2.3.1.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือ การรับและการส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกามาร่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baudrate) มีหน่วยเป็น บิตต่อวินาที (Bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

1. บิตเริ่มต้น (Start bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity bit) จะมีขนาด 1 บิตหรือไม่มี
4. บิตปิดท้าย (Stop bit) จะมีขนาด 1, 1.5 หรือ 2 บิต



รูปที่ 2.22 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.22 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่ง ขา DATA จะมีสถานะลอจิกเป็น "1" ซึ่งจะเรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น (Start bit) จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวน 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นจะตามด้วยบิตพาริตี้ ซึ่งไว้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งก็คือบิตปิดท้าย (Stop bit) ซึ่งจะทำให้ขา DATA มีสถานะเป็นแสดงว่า สิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและการส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver / Transmitter หรือ UART อัตราความเร็วในการรับส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต (Baudrat) ซึ่งก็คือ ค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูลบอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม อาร์เอสสองสามสอง ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ ซึ่งการรับส่งแบบ อนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115,200 บิตต่อวินาที เนื่องจาก บอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่ (odd) หรือ แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ค่าในพาริตี้จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก "1" มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 2.4 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คู่หรือพาริตี้คี่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาเป็นตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้เป็นนัน (NONE) นั้น ทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

ตารางที่ 2.4 แสดงบิตพาริตี้ของข้อมูล

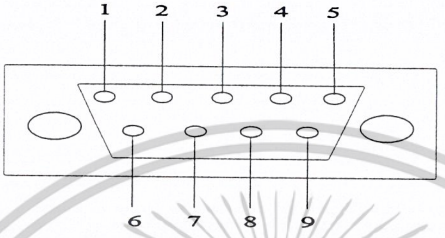
2.3.2 มาตรฐานพอร์ทอนุกรมแบบอาร์เอสสองสามสอง (RS – 232)

มาตรฐานการเชื่อมต่อแบบอนุกรม อาร์เอสสองสามสอง เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบซิงโครนัส 2 ทิศทางโดยมาตรฐาน อาร์เอสสองสามสอง ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดหนึ่งซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS – 232 มาตรฐานนี้ช่วงแรกจะใช้ Connector เป็นแบบ DB – 25 โดยกำหนดความยาวของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง - 5โวลท์ แสดงว่ามีข้อมูล (Mark) และ + 3 ถึง + 15 แสดงว่าเป็นช่องว่าง (Space)

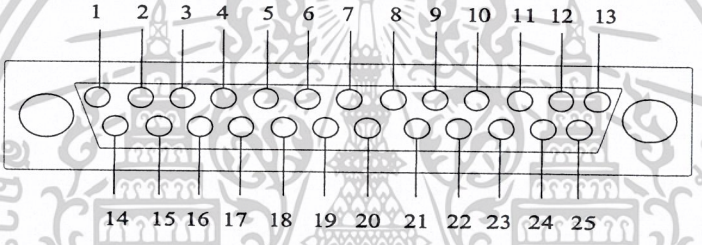
มาตรฐานอาร์เอสสองสามสอง ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูลไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์ หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS – 232 ข้อแตกต่างระหว่างอุปกรณ์ของ DTE และ อุปกรณ์ DCE คือ Connector ของ DTE จะเป็นตัวผู้ ส่วน Connector ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้อยู่กันอย่างทั่วไปจะเป็นแบบ DTE ส่วน Connector ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม อาร์เอสสองสามสอง มักถูกใช้เชื่อมต่อกับโมเด็ม หรือ เมาส์โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร และ Connector สำหรับพอร์ตอาร์เอสสองสามสอง จะใช้ Connector แบบ DB – 25 ตัวผู้ หรือ DB – 9 ตัวผู้ซึ่ง Connector แบบ DB – 25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับ Connector แบบ DB – 9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาดังรูปที่ 2.23



ก) Connector อนุกรม 9 ขา หรือ แบบ DB – 9 (มองจากด้านหลังคอมพิวเตอร์)



ข) Connector อนุกรม 25 ขา หรือ แบบ DB – 25 (มองจากด้านหลังคอมพิวเตอร์)

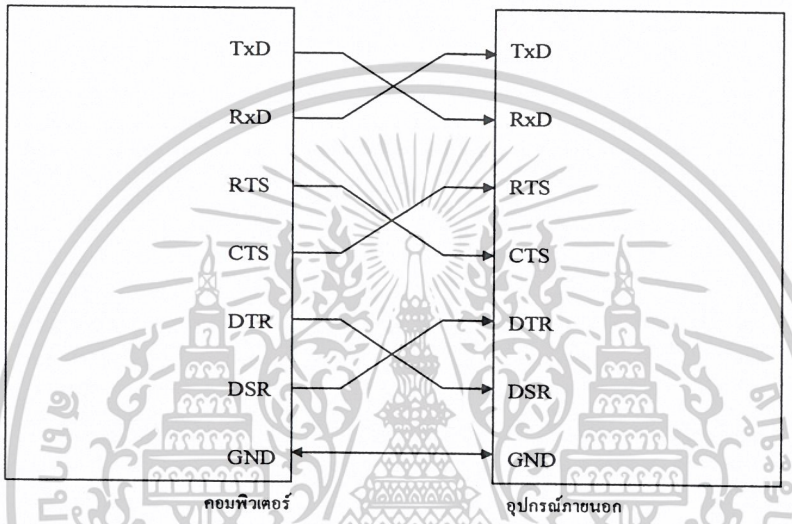
รูปที่ 2.23 การจัดขาของ Connector พอร์ตอนุกรมมาตรฐานอาร์เอสสองสามสอง ทั้งแบบ DB-9 และ DB–25

Connector DB - 9	Connector DB - 25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	Input
2	3	Receivered Data : RxD	Input
3	2	Trasmitted Data : TxD	Output
4	20	Data Terminal Ready : DTR	Output
5	7	Signal Ground : GND	
6	6	Data Set Ready : DSR	Input
7	4	Request To Send : RTS	Output
8	5	Clear To Send : CTS	Input
9	22	Ring Indicator : RI	Input

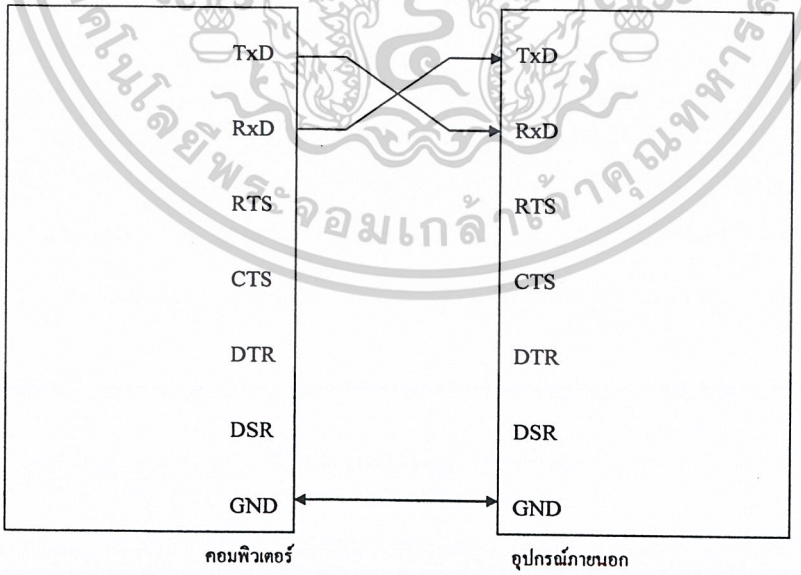
ตารางที่ 2.5 แสดงตำแหน่งและชื่อขาของ DB – 9 และ DB – 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกดังในรูปที่ 2.24 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 2.24 (ก) เป็นการเชื่อมต่อแบบนัลโมเด็ม (Null modem) หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เว็กเต็มรูปแบบ ส่วนในรูปที่ 2.24 (ข) เป็นการเชื่อมต่อแบบนัลโมเด็ม ในลักษณะที่ใช้สายสัญญาณเพียง 3 สาย โดยสายเส้นหนึ่งใช้สำหรับในการส่งข้อมูล สายอีกเส้นหนึ่งใช้สำหรับในการรับข้อมูล และสายเส้นสุดท้ายจะใช้สำหรับไว้เป็นกราวนด์



(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบนัลโมเด็ม (Null moderm)



(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบอาร์เอสสองสามสองโดยใช้สายสัญญาณ 3 เส้น

รูปที่ 2.24 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรายละเอียดการทำงานในแต่ละขาของพอร์ตอนุกรม อาร์เอสสองสามสอง มีดังนี้

- **Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD** ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณคลื่นพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

- **Received Data : RD** ขานี้ใช้สำหรับเพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บในรีจิสเตอร์ บัฟเฟอร์

- **Transmitted Data : TD หรือ TxD** ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์โดยที่จะนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

- **Data Terminal Ready : DTR** เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบนัลโมเด็ม ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

- **Signal Ground** ขากราวนด์ของระบบ

- **Data Set Ready : DSR** ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

- **Request To Send : RTS** เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางที่ส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือ ขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบนัลโมเด็ม 3 สายจะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

- **Clear To Send : CTS** ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

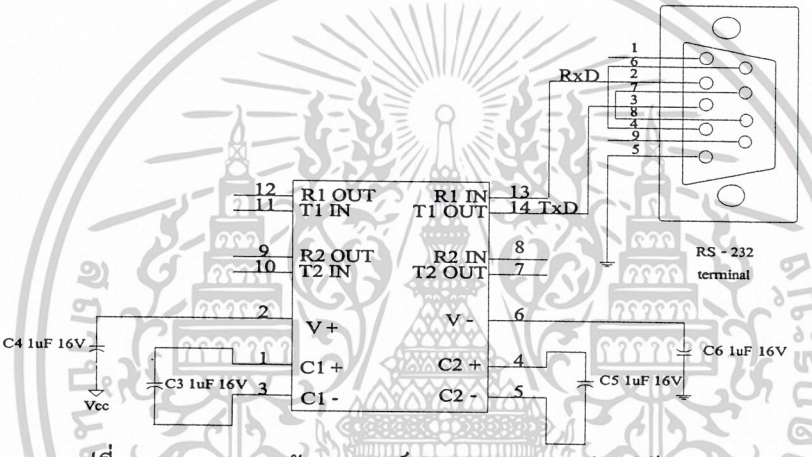
- **Ring Indicator : RI** ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.3.3 ระดับแรงดันที่ใช้งานสำหรับพอร์ตอนุกรมอาร์เอสสองสามสอง

มาตรฐานการสื่อสารข้อมูลของพอร์ตอนุกรม ได้ระบุช่วงระดับแรงดันสำหรับการทำงานของพอร์ตอนุกรมไว้ว่า ที่ลอจิก “0“ จะมีระดับสัญญาณ +3 ถึง + 15 โวลท์ ส่วนลอจิก “1“ จะมีระดับสัญญาณ - 3 ถึง –15 โวลท์ ระดับสัญญาณนี้ทำให้ไม่สามารถนำขาเอาต์พุตใด ๆ ต่อเข้ากับลอจิกเกตเพื่อใช้งานได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะต้องผ่านวงจรเพื่อเปลี่ยนระดับแรงดันเสียก่อน นั่นก็คือ วงจรขับแบบมาตรฐานอาร์เอสสองสามสอง นั้น ด้านภาคส่งต้องสามารถเปลี่ยนระดับสัญญาณลอจิกให้เป็นระดับแรงดันตามที่ได้กำหนดไว้ และสำหรับในส่วนของภาครับเองนั้น ก็ต้องสามารถตรวจจับระดับแรงดันที่รับเข้ามาแล้วเปลี่ยนกลับให้เป็นสัญญาณลอจิกได้อย่างถูกต้องด้วยเช่นกัน โดยปกติจะใช้ไอซีจำพวก RS – 232 Transceiver ที่นิยมมาก คือ MAX 232 หรือ ICL 232 ไอซีกลุ่มนี้จะทำหน้าที่แปลงระดับทีทีแอล ให้เป็นสัญญาณในระดับแรงดันของอาร์เอสสองสามสอง และยังทำหน้าที่แปลงสัญญาณในระดับแรงดันของ อาร์เอสสองสามสองให้กลับลับมาอยู่ในระดับทีทีแอล โดยลอจิก "0" ซึ่งเดิมมีระดับสัญญาณ +3 ถึง +15 โวลท์จะถูกแปลงเป็น 0 โวลท์ ส่วนลอจิก " 1 " ซึ่งมีระดับสัญญาณ -3 ถึง -15 โวลท์ จะแปลงเป็น +5 โวลท์ ทั้งนี้เพื่อให้สามารถเชื่อมต่อกับอุปกรณ์ดิจิตอลอื่นที่ใช้ระดับแรงดันทีทีแอลได้



รูปที่ 2.25 แสดงวงจรขับแบบอาร์เอสสองสามสอง โดยใช้ MAX 232

2.3.4 ข้อจำกัดในการใช้งานของมาตรฐานอาร์เอสสองสามสอง

แม้ว่ามาตรฐานอาร์เอสสองสามสอง มีจุดด้อยเรื่องขีดความสามารถในการควบคุมการรับส่งข้อมูลก็ตามแต่ในทางปฏิบัติแล้วการควบคุมการรับส่งข้อมูลจะเป็นหน้าที่ของซอฟต์แวร์ใช้งาน ซึ่งให้ประสิทธิภาพในการทำงานดีกว่า โดยฮาร์ดแวร์ (Hardware) ซึ่งแนวคิดดังกล่าวก็สอดคล้องกับข้อกำหนดแบบจำลอง OSI (Open System Interconnect) อยู่แล้ว อย่างไรก็ตามมาตรฐานการเชื่อมต่อแบบอาร์เอสสองสามสองยังมีข้อจำกัดอื่น ๆ ในการใช้งานอันเกิดจากโครงสร้างทางกายภาพในส่วนสุดท้ายนี้ผู้เขียนจะได้พูดถึงข้อกำหนดทางกายภาพซึ่งแบ่งออกเป็นข้อจำกัดทางด้านระยะทาง ข้อจำกัดในด้านอัตราเร็ว และข้อจำกัดอันเกิดจากระบบกราวนด์

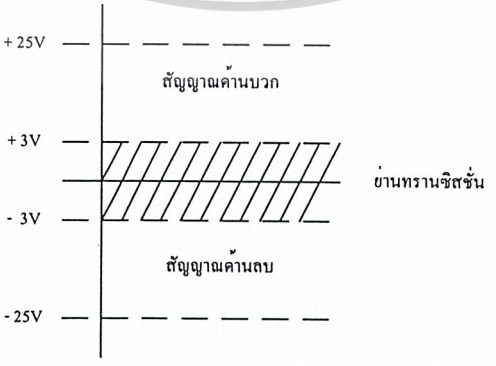
ข้อจำกัดประการแรกของอาร์เอสสองสามสอง คือ ระยะทางในการใช้งานซึ่งไม่สามารถใช้งานได้เกิน 50 ฟุต แม้พบว่าสามารถเพิ่มระยะทางออกไป และยังสามารถส่งข้อมูลได้ในอัตราเร็วที่สูง แต่ความยาวที่เพิ่มขึ้นได้นี้ก็ไม่มากนัก และยังเป็นการเพิ่มความเสี่ยงต่อเนื้อหาข้อมูลที่ส่งผ่านไปด้วย อย่างไรก็ตามข้อจำกัดดังกล่าวก็ไม่ถึงกับเป็นจุดด้อยของอาร์เอสสองสามสอง ทั้งนี้เพราะในการใช้งานโดยทั่วไปโมเด็มจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อยู่ใกล้กับเครื่องคอมพิวเตอร์หรืออุปกรณ์เทอร์มินอล ส่วนในระยะไกลจะกระทำผ่านคู่สายโทรศัพท์โดยผ่านการมอดูเลตสัญญาณแต่สำหรับการเชื่อมต่อเครื่องคอมพิวเตอร์ที่อยู่ใกล้กัน โดยตรงด้วยการใช้สายนำสัญญาณ RS – 232 ซึ่งในกรณี นี้อาจไม่จำเป็นต้องใช้โมเด็มเป็นสื่อกลางข้อจำกัดของระยะทางจะกลายเป็นปัจจัยที่สำคัญที่ใช้ในการพิจารณาทันที

มาตรฐาน RS – 232 กำหนดให้อุปกรณ์ด้านส่งสร้างแรงดันในช่วง +5 โวลท์ ถึง +25 โวลท์ เพื่อใช้แทนลอจิก “0” และสร้างแรงดันในช่วง -5 โวลท์ ถึง -25 โวลท์ สำหรับใช้แทนลอจิก “1” ปัญหาที่เกิดขึ้นก็คือช่วงแรงดันดังกล่าวแตกต่างจากช่วงแรงดันที่ใช้ในการอ้างอิงระดับลอจิกที่ใช้งานในเครื่องคอมพิวเตอร์ ซึ่งจะเป็นแรงดันในช่วงการทำงานของวงจรรวมแบบทีทีแอล (TTL) หรือ มอส (MOS) นั่นเป็นสาเหตุที่ทำให้ต้องมีการป้อน แรงดัน + / - 12 โวลท์ ให้กับเครื่องคอมพิวเตอร์เพื่อใช้สำหรับการสื่อสารผ่านจุดเชื่อมต่อแบบ RS – 232 จากรูปที่ 2.17 จะเห็นว่าอุปกรณ์ภาครับจะถือว่าแรงดันที่ได้รับสูงเกินกว่า + 3 โวลท์ เป็นลอจิก “0” และแรงดันที่ต่ำกว่า -3 โวลท์เป็นลอจิก “1” การเปลี่ยนแปลงขั้วแรงดันจะต้องทำให้เสร็จภายใน 1 / 20 ของความกว้างบิตข้อมูล นอกจากนี้มาตรฐานอาร์เอสสองสามสองยังมีการจำกัดค่าความจุสเตรย์ (Stray Capacitance) ในสายนำสัญญาณไม่ให้สูงเกินกว่า 2500 พิโกฟารัด มิฉะนั้นจะส่งผลกระทบต่อช่วงเวลาขาขึ้นของสัญญาณ ทำให้ใช้เวลามากขึ้นเกินกว่า 2500 พิโกฟารัด มิฉะนั้นจะส่งผลกระทบต่อช่วงเวลาขาขึ้นของสัญญาณทำให้ใช้เวลามากขึ้นเกินกว่า 1 / 20 ของความกว้างบิตข้อมูล ด้วยเหตุดังกล่าว เมื่อพิจารณาถึงสายนำสัญญาณทั่วไปซึ่งมีค่าความจุในช่วง 40 - 50 พิโกฟารัด ต่อระยะทางหนึ่งฟุต จึงทำให้ระยะทางสูงสุดมีค่าได้ไม่เกิน 50 ฟุต (2500 / 50 = 50)

ประการที่สองคือ อัตราเร็ว อัตราเร็วในการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับโมเด็มผ่านพอร์ตอนุกรมแบบอาร์เอสสองสามสองอาจกลายเป็นข้อจำกัดในการใช้งานอีกประการหนึ่งอัตราเร็วสูงสุดที่สามารถรับส่งผ่านพอร์ตอนุกรมสื่อสารอาร์เอสสองสามสอง โดยใช้สายนำสัญญาณยาว 50 ฟุต จะถูกจำกัดไว้ที่ 20000 บิตต่อวินาที ดังนั้นอัตราเร็วในการสื่อสารระหว่างเครื่องคอมพิวเตอร์กับโมเด็มจะทำได้สูงสุดที่ 19200 บิตต่อวินาที ซึ่งก็เรียกได้ว่าไม่เลวร้ายนักสำหรับการใช้ในปัจจุบัน อย่างไรก็ตามสายนำสัญญาณ

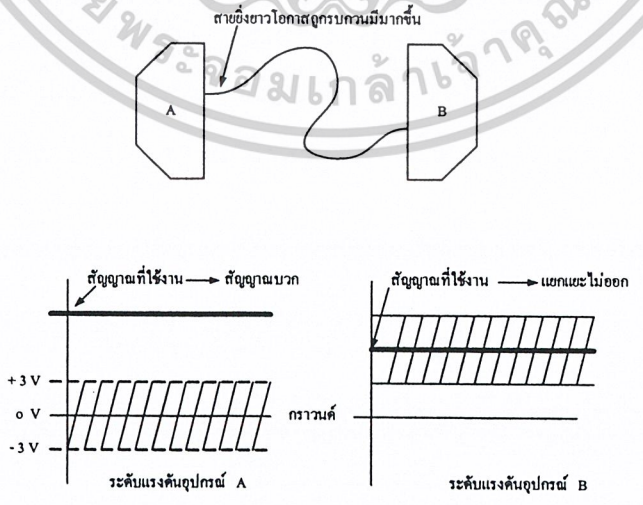


รูปที่ 2.26 การกำหนดระดับสัญญาณตามมาตรฐาน RS – 232 ทางด้านอุปกรณ์ภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาร์เอสสองสามสองที่มีใช้งานอยู่ทั่วไปมักจะมีความยาวไม่มากนักอย่างมากก็ไม่เกิน 12 ฟุต จึงทำให้สามารถทำการรับส่งข้อมูลได้ที่อัตราเร็วสูงกว่า 20000 บิตต่อวินาที ซึ่งไม่ถือเป็นการฝืนมาตรฐานของอาร์เอสสองสามสองแต่ประการใด เนื่องจากสายนำสัญญาณนั้นจะสั้นกว่าขีดจำกัดสูงสุด ในกรณีของสายนำสื่อสารผ่านโมเด็มโดยไม่มีการบีบอัดข้อมูล ตัวอย่างเช่น โมเด็มอัตราเร็ว 33.6 กิโลบิตต่อวินาที อัตราเร็วของข้อมูลที่ส่งจากเครื่องคอมพิวเตอร์ไปสู่โมเด็ม และอัตราเร็วของข้อมูลที่โมเด็มส่งผ่านคู่สายโทรศัพท์จะมีค่าเท่ากัน ผู้ใช้งานจึงสามารถส่งข้อมูลจากเครื่องคอมพิวเตอร์ของตนด้วยอัตราเร็วค่าดังกล่าวได้โดยไม่มีปัญหาแต่ประการใด

อย่างไรก็ตาม หากโมเด็มที่ใช้มีฟังก์ชั่นบีบอัดข้อมูลร่วมอยู่ด้วยก็จะทำให้อัตราเร็วของข้อมูลที่ส่งผ่านคู่สายโทรศัพท์ต่ำกว่าอัตราเร็วของข้อมูลที่ได้รับจากเครื่องคอมพิวเตอร์ส่วนจะต่ำกว่าเท่าใดนั้นก็จะขึ้นอยู่กับประสิทธิภาพของการบีบอัดข้อมูล และเนื้อหาของข้อมูลที่กำลังส่งในขณะนั้นสมมติง่าย ๆ ว่าประสิทธิภาพในการบีบอัดข้อมูลมีค่าเป็น 4 เท่าตายตัว การส่งข้อมูล 10000 บิต ภายในหนึ่งวินาทีจากเครื่องคอมพิวเตอร์ไปยังโมเด็มจะทำให้มีข้อมูลส่งออกจากโมเด็มผ่านคู่สายโทรศัพท์เพียง 2500 บิตภายในหนึ่งวินาทีเช่นกัน ดังนั้นเมื่อคิดถึงโมเด็มที่อัตราเร็ว 33.6 กิโลบิตต่อวินาที ที่มีฟังก์ชั่นบีบอัดข้อมูลภายในช่วง 2 – 4 เท่า หากผู้ใช้งานต้องการให้โมเด็มส่งข้อมูลออกไปด้วยอัตราเร็วสูงสุดตลอดเวลาก็จะต้องทำให้พอร์ตแบบอนุกรมอาร์เอสสองสามสองของตนสามารถรองรับการส่งข้อมูลได้ในช่วงอัตราเร็วระหว่าง 67.2 กิโลบิตต่อวินาที ถึง 134.4 กิโลบิตต่อวินาที อย่างไรก็ตามอุปกรณ์ควบคุมการอินเตอร์เฟสผ่านพอร์ตอนุกรมกับพอร์ตอนุกรม ที่ฝังอยู่ในวงจรควบคุมพอร์ตอนุกรมของเครื่องคอมพิวเตอร์จะไม่สามารถควบคุมคุณภาพการรับส่งข้อมูลที่อัตราเร็ว สูงกว่า 115.2 กิโลบิตต่อวินาที ผู้ผลิตโมเด็มบางรายจึงกำหนดให้ใช้งานโมเด็มของตนกับพอร์ตอนุกรมด้วยอัตราเร็วสูง หรือมิฉะนั้นก็ออกแบบโมเด็มให้สามารถเชื่อมต่อกับพอร์ตสื่อสารแบบขนานบนเครื่องคอมพิวเตอร์โดยใช้การควบคุมจากซอฟต์แวร์โดยตรง



รูปที่ 2.27 ปัญหาจากความแตกต่างของกราวนด์ในการเชื่อมต่อผ่านมาตรฐานอาร์เอสสองสามสอง

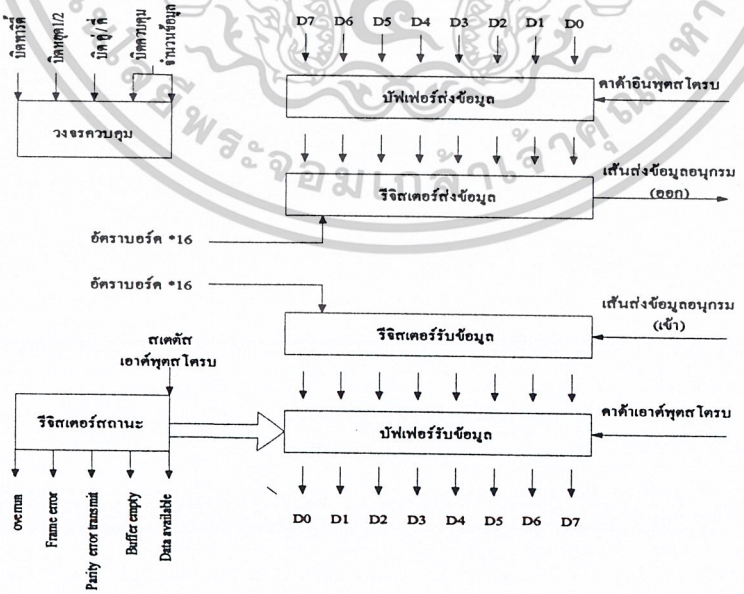
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อจำกัดประการที่สาม เป็นเรื่องระบบกราวนด์ จากรูปที่ 2.27 เมื่อสายนำสัญญาณมีความยาวมากขึ้นแม้จะมีการกำหนดให้สัญญาณควบคุม และสัญญาณแต่ละขาอ้างอิงกับสัญญาณกราวนด์ขาที่ 7 ก็ตาม แต่เมื่อใดก็ตามที่ความต่างศักย์ของกราวนด์ระหว่างอุปกรณ์สื่อสารต้นทางกับอุปกรณ์สื่อสารปลายทางมีความแตกต่างกัน การตัดสินข้อมูลจากระดับแรงดันสัญญาณที่ได้รับที่อุปกรณ์สื่อสารปลายทางจะเกิดความผิดพลาดทั้งนี้เพราะความแตกต่างของระดับกราวนด์อาจทำให้ย่านการตัดสินใจถูกยกสูงขึ้น ดังรูปที่ 2.27 ซึ่งอุปกรณ์สื่อสารปลายทางไม่สามารถบอกลอจิกของสัญญาณที่ได้รับอย่างถูกต้อง นอกจากนั้นในกรณีของการใช้สายนำสัญญาณที่มีความยาวมาก โดยอุปกรณ์ทั้งสองด้านมีความต่างศักย์ของระดับกราวนด์หากอัตราเร็วในการรับส่งข้อมูลสูงมากเท่าใด โอกาสในการเกิดปัญหาลักษณะดังกล่าวก็จะยิ่งเพิ่มมากขึ้นเท่านั้น

2.3.5 การสื่อสารอนุกรมยูเออาร์ที (UART : Universal Asynchronous Receiver -Transmitter)

ยูเออาร์ที เป็นอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัส สำหรับการสื่อสารอนุกรมบนเครื่องคอมพิวเตอร์แล้ว ยูเออาร์ที ถือว่าเป็นหัวใจสำคัญของการสื่อสารแบบอนุกรม

หน้าที่หลักของยูเออาร์ที คือ ทำหน้าที่ในการแปลงข้อมูลรูปแบบขนานจากคอมพิวเตอร์เป็นรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่ในการแปลงสัญญาณอนุกรมแบบ อะซิงโครนัสที่ป้อนเข้ามายังยูเออาร์ทีให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ อีกทั้งยังแจ้งข้อมูลอื่น ๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (Baudrate), รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน)



รูปที่ 2.28 บล็อกไดอะแกรมของยูเออาร์ที (UART)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการส่งข้อมูล 8 บิตแบบอนุกรมนี้จะต้องเริ่มต้นมีบิตเริ่มต้นและบิตหยุดเพิ่มขึ้นมา ซึ่งจะทำให้ข้อมูลที่ถูกส่งไปจริง ๆ มีขนาด 10 บิต โดยบิตเริ่มต้นจะมีค่าเป็น “0” เสมอ เพื่อจะบอกยูเออาร์ทีที่รับข้อมูลให้รู้ว่ามีข้อมูลกำลังเข้ามา และบิตหยุดมีค่าเป็น “1” เสมอ เพื่อจะบอกยูเออาร์ทีที่รับข้อมูลให้รู้ว่าการส่งข้อมูลได้สิ้นสุดลงแล้ว

ส่วนที่มีหน้าที่ส่งข้อมูลของ ยูเออาร์ที แยกออกได้เป็น 2 ส่วน คือ บัฟเฟอร์ส่งข้อมูล (Transmitted data output buffer) กับ รีจิสเตอร์ส่งข้อมูล (Transmitted register) โดยรีจิสเตอร์ส่งข้อมูลจะมีหน้าที่เลื่อนข้อมูลส่งออกไปยังเส้นส่งข้อมูลอนุกรม โดยเริ่มต้นจากบิตเริ่มต้น ถัดมาเป็นบิตข้อมูล และตามด้วยบิตหยุดข้อมูลขนาด 8 บิต ที่จะส่งจะถูกนำไปเก็บไว้ในบัฟเฟอร์ส่งข้อมูลเมื่อสัญญาณควบคุม DATA อินพุตสโตรบ (Data Input Stobe) เปลี่ยนค่าจาก 1 เป็น 0 และการส่งข้อมูลแบบอนุกรมจะเริ่มต้นขึ้นเมื่อสัญญาณควบคุมนี้เปลี่ยนค่าจาก 0 เป็น 1

ส่วนที่ทำหน้าที่รับข้อมูลของยูเออาร์ที เหมือนเป็นภาพสะท้อนของส่วนที่ทำหน้าที่ในการส่งข้อมูล ข้อมูลที่เข้ามาทางเส้นรับข้อมูลอนุกรมจะถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูลโดยการเลื่อนค่า 10 หรือ 11 ครั้ง ขั้นตอนดังกล่าวจะเริ่มมีบิตเริ่มต้นเข้ามา และทันทีที่ข้อมูลถูกเลื่อนเข้าไปเก็บในรีจิสเตอร์รับข้อมูลแล้วข้อมูลในรีจิสเตอร์นี้จะถูกนำไปเก็บไว้ในบัฟเฟอร์รับข้อมูลเมื่อมีสัญญาณควบคุมดาต้าเอาท์พุท สโตรบ

ภายในยูเออาร์ที จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (Programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของยูเออาร์ที โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1 – 65,535 ยูเออาร์ทีสามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (Half duplex) และ ฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

2.3.5.1 ชนิดของยูเออาร์ที

ในเครื่องคอมพิวเตอร์ทั่วไปมียูเออาร์ทีที่ใช้งานกันอยู่ทั่วไปมีอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็นยูเออาร์ทีมาตรฐานที่ใช้กันมาอย่างยาวนาน ยูเออาร์ทีเบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลได้ในคราวเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ยูเออาร์ทีเบอร์นี้ถือว่าเป็นต้นแบบของยูเออาร์ทีที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุก ๆ รุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของยูเออาร์ทีเบอร์นี้

ยูเออาร์ทีอีกเบอร์หนึ่งคือ 16450 มีความสามารถในการรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับยูเออาร์ที นอกจากนี้ยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบเอฟไอเอฟโอ (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ยูเออาร์ที เบอร์นี้หรือใหม่กว่านี้ เช่นเบอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TL16C750 ซึ่งมีรีจิสเตอร์แบบเอฟไอเอฟโอ ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5 โวลท์ และ +3 โวลท์ มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกกะบิตต่อวินาที เมื่อใช้สัญญาณนาฬิกา 16 MHz

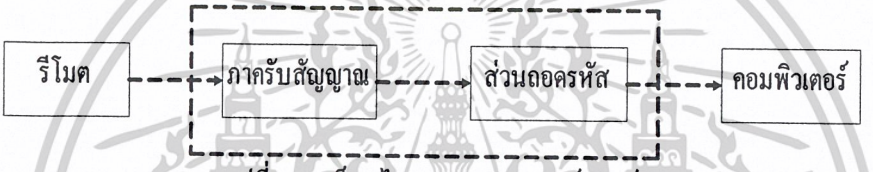
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการออกแบบและการสร้าง

ในการออกแบบโครงงาน เครื่องควบคุมคอมพิวเตอร์ระยะไกล (Computer Remote Control) ได้แบ่งการออกแบบเป็น 2 ส่วนหลัก คือ

1. ฮาร์ดแวร์ (Hardware)
2. ซอฟท์แวร์ (Software)

3.1 การออกแบบฮาร์ดแวร์



รูปที่ 3.1 บล็อกไดอะแกรมของฮาร์ดแวร์

3.1.1 รีโมทคอนโทรล

ทำหน้าที่ควบคุมการทำงาน โดยใช้รีโมทสำเร็จรูปที่หาง่าย (เลือกใช้รีโมททีวี เพราะมีฟังก์ชั่นในการกำหนดการควบคุมเอาท์พุท) เพื่อสะดวกในการกำหนดช่องสัญญาณที่ใช้ควบคุมคอมพิวเตอร์

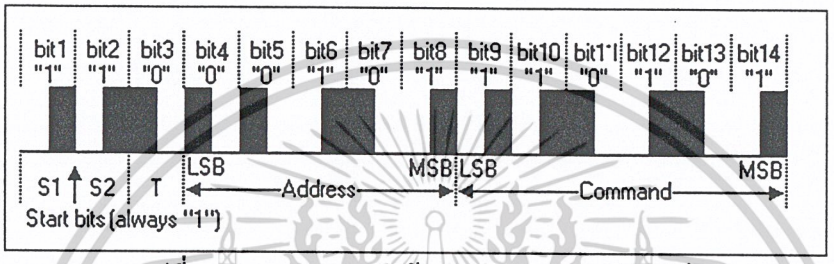
รูปแบบข้อมูลของรีโมทที่นำมาใช้ในโครงงานนี้เป็นแบบ Shift Code Signal และมาตรฐานการส่งข้อมูลเป็นแบบRC5 เป็นมาตรฐานการส่งข้อมูลแบบอนุกรมที่ใช้ในเครื่องใช้ไฟฟ้าที่บริษัท ฟิลิปส์ และบริษัท ผู้ผลิตรายอื่น ๆ อีกหลายรายใช้กัน เนื่องจากเป็นมาตรฐานที่ฟิลิปส์เป็นผู้เริ่มต้นขึ้นเป็นครั้งแรก บางครั้งมาตรฐานนี้จึงถูกเรียกว่า “ฟิลิปส์ RC5” ด้วย โดยรีโมทคอนโทรลที่ใช้มาตรฐานนี้จะใช้การส่งข้อมูลขนาด 14 บิต ด้วยการส่งแบบอนุกรมไปยังเครื่องใช้ไฟฟ้า ซึ่งเครื่องใช้ไฟฟ้าที่รองรับมาตรฐานเดียวกันก็จะสามารถเข้าใจกับข้อมูลที่รีโมทคอนโทรลได้ส่งมา และทำงานตามคำสั่งที่ระบุมาในข้อมูลนั้นเอง โดยโครงสร้างของข้อมูลขนาด 14 บิตที่กล่าวมานั้นประกอบด้วย ข้อมูล 2 บิตแรกเป็น บิตเริ่มต้น (Start Bits), 1 บิต ถัดมาเป็น บิตทอกเกิล (Toggle Bits), ตามด้วย รหัสแอดเดรส (Address Bits) ขนาด 5 บิต และรหัสคำสั่ง (Command Bits) ขนาด 6 บิต

กลไกการทำงานของรีโมทคอนโทรลจะเริ่มต้นขึ้นหลังจากกดที่ปุ่มใดปุ่มหนึ่งบนรีโมท บิตเริ่มต้น (Start Bits) 2 บิตแรกจะถูกส่งออกไปก่อน ตามติดมาด้วยบิตทอกเกิลซึ่งจะมีสถานะเปลี่ยนไปทุกครั้งที่มีการกดปุ่มซ้ำเดิม แต่ในกรณีที่ปุ่มถูกกดค้างเอาไว้รีโมทคอนโทรลจะทำการส่งข้อมูลอย่างซ้ำ ๆ ภายในเวลา 113.778 มิลลิวินาที โดยสถานะของบิตทอกเกิลก็จะค้างอยู่ที่ลอจิกหนึ่ง (ลอจิก “1” หรือ “0”) โดยไม่มีการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงใด ๆ เกิดขึ้นกับบิตนี้ อุปกรณ์ไฟฟ้าจะสามารถรับรู้ได้ว่ามีการกดปุ่มเดียวกันซ้ำ ๆ หรือกดปุ่มค้างเอาไว้ก็จากการตรวจสอบสถานะของบิตทอกเกิลนี่เอง

รหัสแอดเดรสจะเป็นชุดข้อมูลที่ถูกส่งตามท้ายบิตทอกเกิลมาติด ๆ โดยรหัสแอดเดรสที่หน้าที่เป็นรหัสระบุชนิดของอุปกรณ์ไฟฟ้าที่รีโมทจะเข้าไปควบคุม โดยในมาตรฐาน RC5 มีการกำหนดรหัสแดเดรสเพื่อใช้แทนชนิดของอุปกรณ์ไฟฟ้าเป็นค่าที่ตายตัวเอาไว้แล้ว เช่น เครื่องรับโทรทัศน์มีรหัสแอดเดรสเป็น 00000, เครื่องวีดีโอเทปมีรหัสแอดเดรสเป็น 00101 เป็นต้น



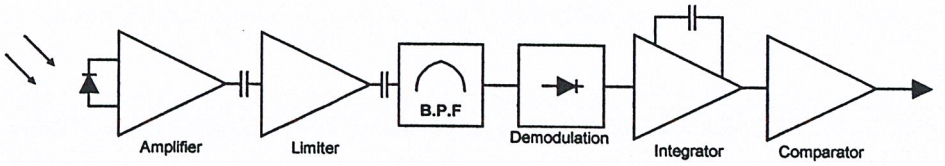
รูปที่ 3.2 มาตรฐานการส่งข้อมูลแบบ RC5 ของฟิลิปส์

หลังจากนั้นรหัสคำสั่งซึ่งเป็นรหัสแทนปุ่มกดแต่ละปุ่มบนรีโมทคอนโทรลจะถูกส่งตามมาโดยปุ่มแต่ละปุ่มบนรีโมทคอนโทรล RC5 จะมีรหัสคำสั่งที่เป็นค่าตายตัวเช่นเดียวกัน เช่น ปุ่ม 0 มีรหัสคำสั่งเป็น 000000, ปุ่ม 1 มีรหัสคำสั่งเป็น 000001, ปุ่ม 2 มีรหัสคำสั่งเป็น 000010 เป็นต้น เมื่ออุปกรณ์ไฟฟ้าได้รับข้อมูล 14 บิต ที่รีโมทคอนโทรลส่งไป ก็จะนำข้อมูลที่ได้รับไปตีความเพื่อทำงานตามคำสั่งที่ได้รับต่อไป

ซึ่งโดยการวิเคราะห์รูปแบบข้อมูลของรีโมทคอนโทรลที่ใช้แล้วนั้น จะเห็นได้ว่ารูปแบบนี้คล้ายกับการเข้ารหัสข้อมูลแบบ Manchester กล่าวคือ จะใช้การเปลี่ยนแปลงของระดับสัญญาณที่จุดกึ่งกลางเพื่อแสดงข้อมูลที่ส่งไป โดยการเปลี่ยนระดับสัญญาณจาก ลบเป็นบวก เป็นรหัส “1” ส่วน การเปลี่ยนระดับสัญญาณจาก บวกเป็นลบ เป็นรหัส “0”

3.1.2 ภาครับสัญญาณอินฟราเรด

ในภาครับสัญญาณได้ทดลองทำการเก็บค่าสัญญาณรีโมทที่จะนำมาเป็นสัญญาณอินพุตของไมโครคอนโทรลเลอร์ โดยเลือกการส่งสัญญาณแบบ RC5 ในการทดลองและได้ทำการเก็บค่าสัญญาณจากการกดปุ่มต่าง ๆ ของรีโมท โดยเลือกใช้ IC เบอร์ IRM-8601S เป็นตัวรับสัญญาณอินฟราเรด

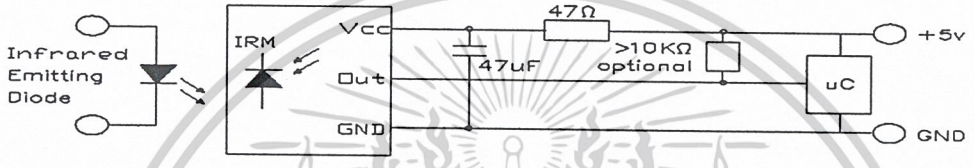


รูปที่ 3.3 ตัวอย่างบล็อกไดอาแกรมของตัวรับสัญญาณอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

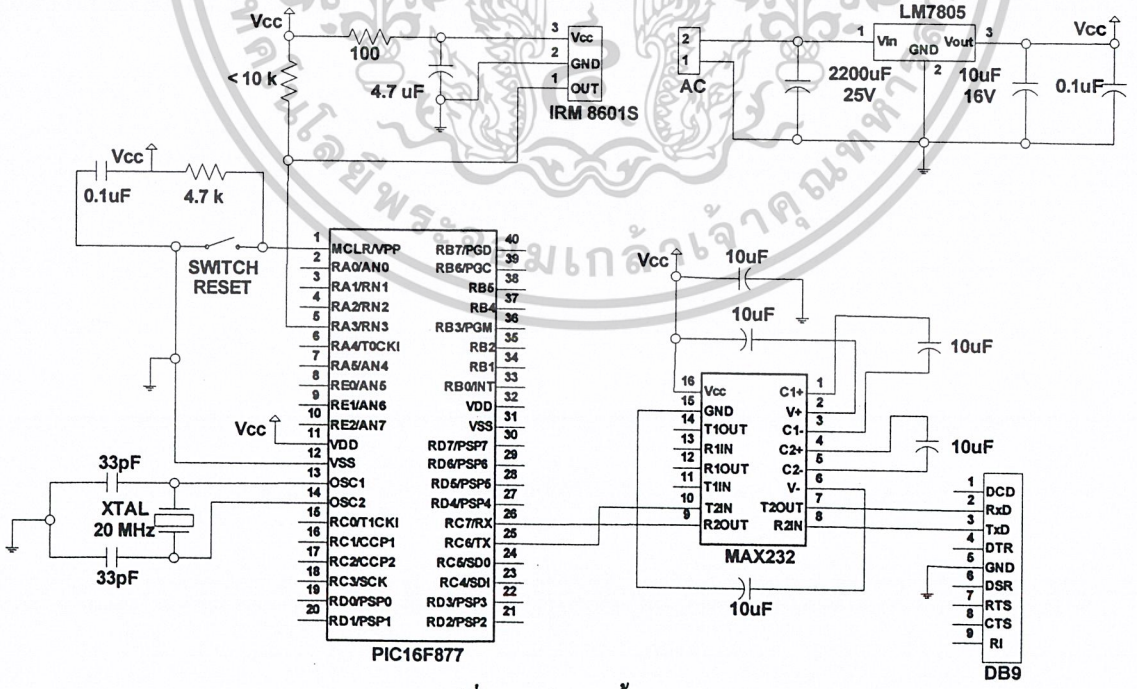
คุณสมบัติ IC IRM-8601S มีดังนี้

- ใช้ low voltage และ กำลังไฟฟ้าต่ำ
- มีวงจรขยายสัญญาณ กรองความถี่แบนด์พาสและดีมอดูเลตอยู่ภายใน
- รวมการใช้ TTL และ CMOS ภายใน IC
- ป้องกันการรบกวนของแสงได้ดี
- สามารถป้องกันการรบกวนจากสนามแม่เหล็กได้
- รับสัญญาณได้ในระยะไกลและมีความไวสูง



รูปที่ 3.4 วงจรรับสัญญาณอินฟราเรด

สามารถรับสัญญาณของรีโมทคอนโทรลของอุปกรณ์ต่าง ๆ ได้ เช่น TVs, VCRs, เครื่องเสียง, เครื่องปรับอากาศ, พัดลมไฟฟ้า, สวิตช์ที่ใช้แสงควบคุม เป็นต้น ทำให้ส่วนของภาครับสัญญาณสามารถรับสัญญาณได้หลายรูปแบบ โดยจะส่งสัญญาณต่อไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผลต่อไป



รูปที่ 3.5 รูปวงจรทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

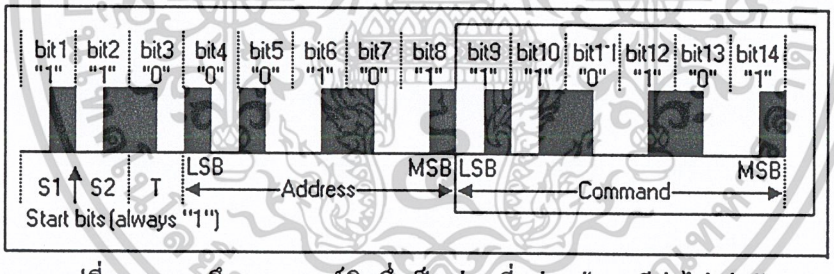
3.2 การออกแบบซอฟท์แวร์

ในส่วนของการออกแบบซอฟท์แวร์นี้ มีสองส่วน ส่วนแรก คือ การออกแบบโปรแกรมของไมโคร - คอนโทรลเลอร์ PIC16F877 โดยรับค่าสัญญาณอินฟราเรดจากรีโมทอินฟราเรด แบบ RC5 จากนั้นนำข้อมูลที่ได้จากการรับสัญญาณมาประมวลผล เพื่อส่งสัญญาณเอาต์พุตไปควบคุมโปรแกรมต่าง ๆ ในคอมพิวเตอร์ และส่วนที่สองก็คือ การออกแบบโปรแกรมในคอมพิวเตอร์โดยใช้ ภาษา C โดยโปรแกรมประยุกต์ Visual C++ โดยการรับค่าจากไมโครคอนโทรลเลอร์ ผ่านทาง พอร์ตอนุกรม แล้วนำค่าอินพุตที่ได้รับไปประมวลผลควบคุมโปรแกรมต่าง ๆ ในคอมพิวเตอร์

3.2.1 การออกแบบซอฟท์แวร์ของไมโครคอนโทรลเลอร์

สัญญาณอินฟราเรดที่รับเข้ามานั้น จะใช้ IRM 8601S รับสัญญาณอินฟราเรด แล้วต่อสัญญาณเอาท์พุตเข้ากับขา RA3 ของ PIC16F877 ซึ่งมีโปรแกรมการรับข้อมูล RC5 โดยการทำงานของโปรแกรมรับสัญญาณอินฟราเรดเริ่มจาก การตรวจสอบว่ามีสัญญาณจากตัวรับอินฟราเรดส่งมาให้หรือยัง เมื่อมีสัญญาณเข้ามาก็โปรแกรมการรับสัญญาณและเริ่มการถอดรหัสสัญญาณ

สำหรับโปรแกรมประมวลผลค่าสัญญาณในแบบ RC5 โปรแกรมจะทำการสุ่มค่าสัญญาณทุก ๆ 1700 μsec ตั้งแต่บิตแรกของ command หรือ บิตที่ 9 ของสัญญาณจนถึง บิตสุดท้ายของ command หรือบิตที่ 14 ของสัญญาณ แล้วนำค่าที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์ Data แล้วส่งค่าที่เก็บได้ของแต่ละปุ่มส่งผ่าน พอร์ตอนุกรมไปยังคอมพิวเตอร์ต่อไป ดังรูปโฟลวชาร์ทที่ 3.7 และ 3.8



รูปที่ 3.6 แสดงถึงคอมมานด์บิตซึ่งเป็นส่วนที่แต่ละปุ่มจะมีค่าไม่เท่ากัน

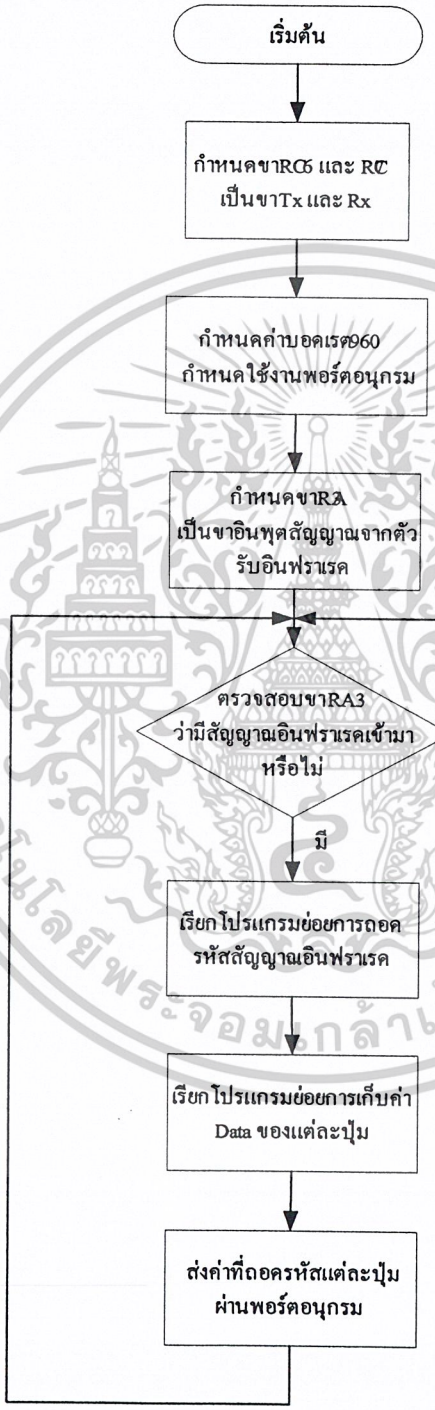
3.2.2 การออกแบบซอฟท์แวร์ในคอมพิวเตอร์

ส่วนการออกแบบซอฟท์แวร์ในคอมพิวเตอร์ หลังจากที่รีจิสเตอร์ Data เก็บค่าสัญญาณอินฟราเรดจากส่วนภาครับ แล้วนำค่าที่เก็บไว้ส่งต่อไปยังคอมพิวเตอร์ผ่านพอร์ตอนุกรม โปรแกรมประมวลผลในคอมพิวเตอร์ก็จะรับเอาค่าที่ส่งมาไปวิเคราะห์ว่า ให้ทำอะไรบ้าง

ส่วนโปรแกรมประมวลผลในคอมพิวเตอร์ แสดงที่รูปโฟลวชาร์ตที่ 3.9 ถึง 3.16 ซึ่งเมื่อคอมพิวเตอร์รับข้อมูลเข้ามาก็จะตรวจสอบดูว่าเป็นคำสั่งให้ทำอะไร เช่น ให้เพิ่มหรือลดเสียง หรือให้ทำการเปิดโปรแกรมใดขึ้นมา เป็นต้น ซึ่งโปรแกรมที่เกี่ยวกับเสียงจะพิเศษกว่าโปรแกรมอื่นก็คือ จะสามารถทำงานได้ตลอดไม่ว่าจะเปิดโปรแกรมใดไว้ก็ตาม แต่หากต้องการเปิดฟังก์ชั่นของโปรแกรมต่าง จะต้องเปิดโปรแกรมนั้นเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น หากจะเล่นเพลง Winamp ก็จะต้องเปิด Winamp ถึงจะทำงาน แต่ถ้าหากเปิดโปรแกรมอื่นซ้อนขึ้นมาทับโปรแกรม Winamp ก็จะไม่สามารถสั่งงาน Winamp ได้ เป็นต้น



รูปที่ 3.7 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมการนำข้อมูลจากรีโมทไปถอดรหัส

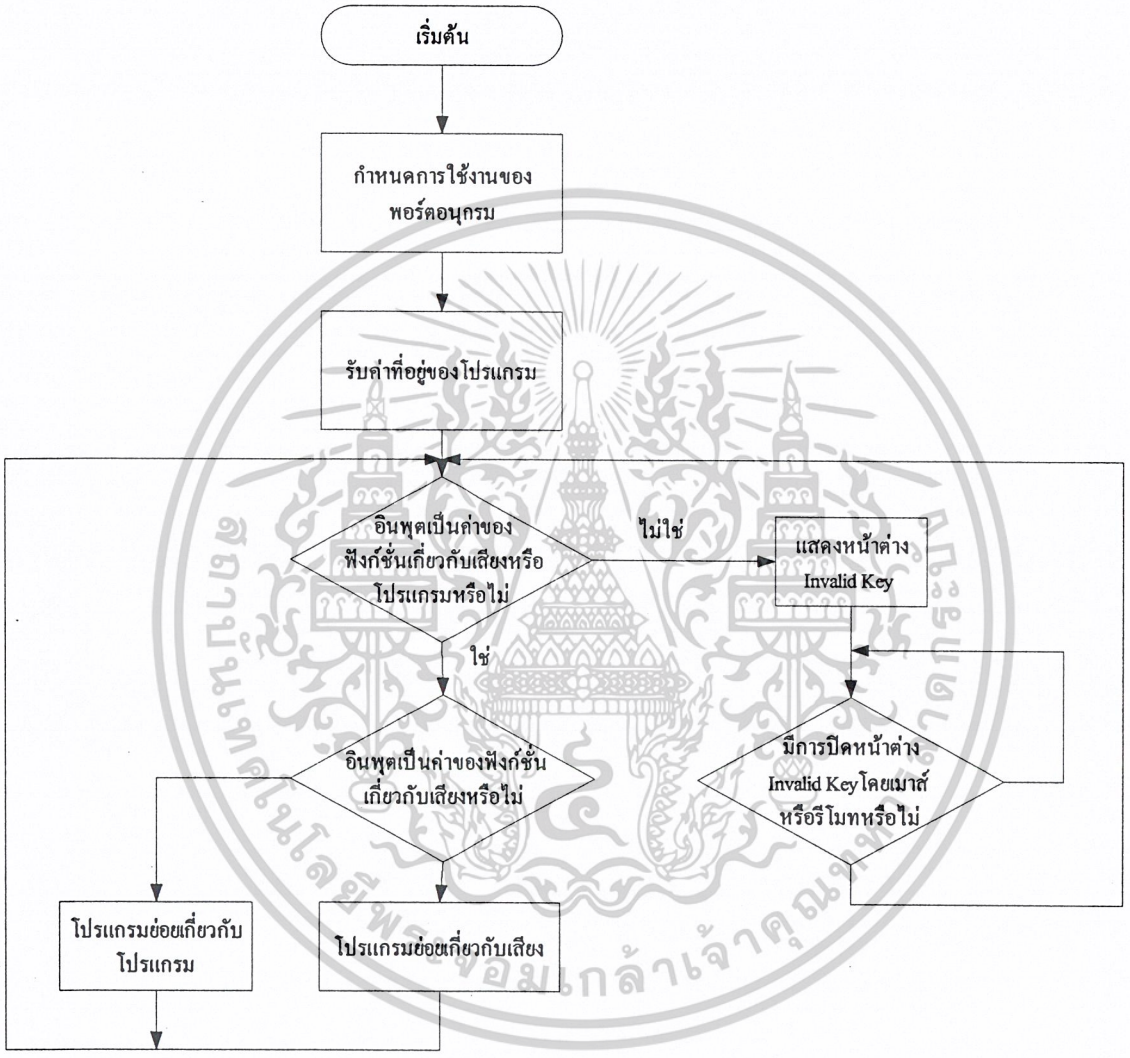
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(ก) แสดงโฟล์วชาร์ทการทำงานของโปรแกรมย่อยการถอดรหัสสัญญาณอินฟราเรด

(ข) แสดงโฟล์วชาร์ทการทำงานของโปรแกรมย่อยการเก็บค่าดาต้าของแต่ละปุ่ม

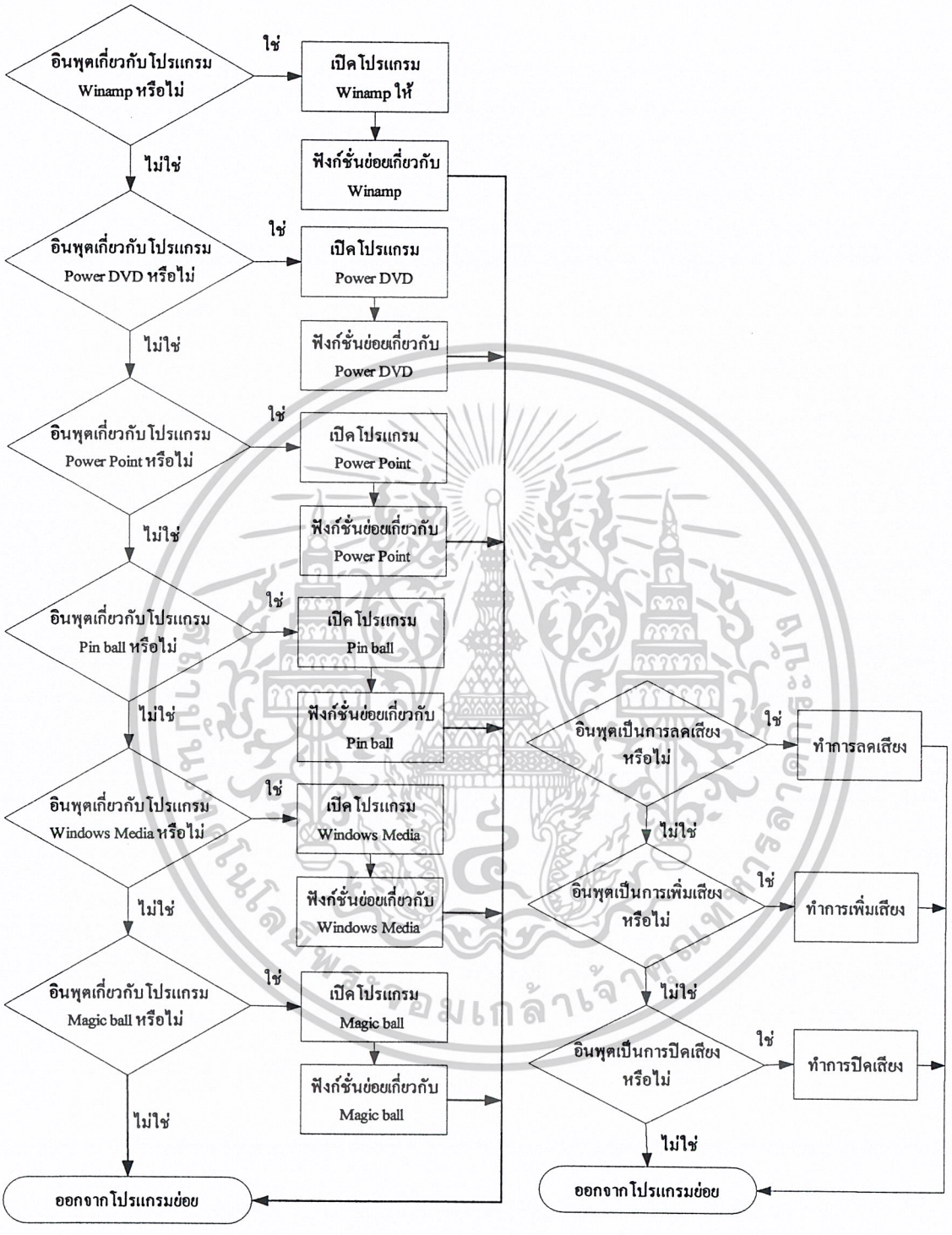
รูปที่ 3.8 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมย่อยในไมรโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมที่ใช้ควบคุมโปรแกรมในคอมพิวเตอร์ด้วยรีโมท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

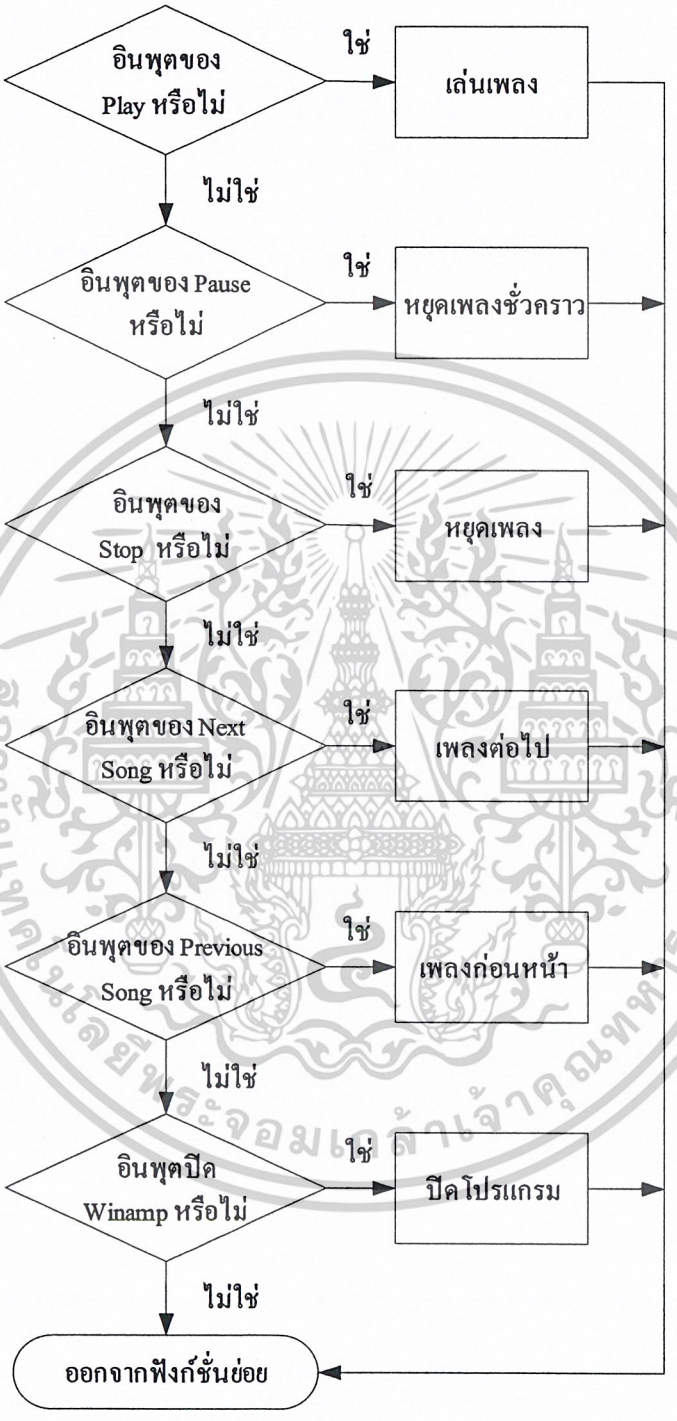


(ก) โปรแกรมย่อยการทำงานเกี่ยวกับโปรแกรมต่าง ๆ ที่ใช้รีโมทควบคุม

(ข) โปรแกรมย่อยการทำงานเกี่ยวกับเสียง

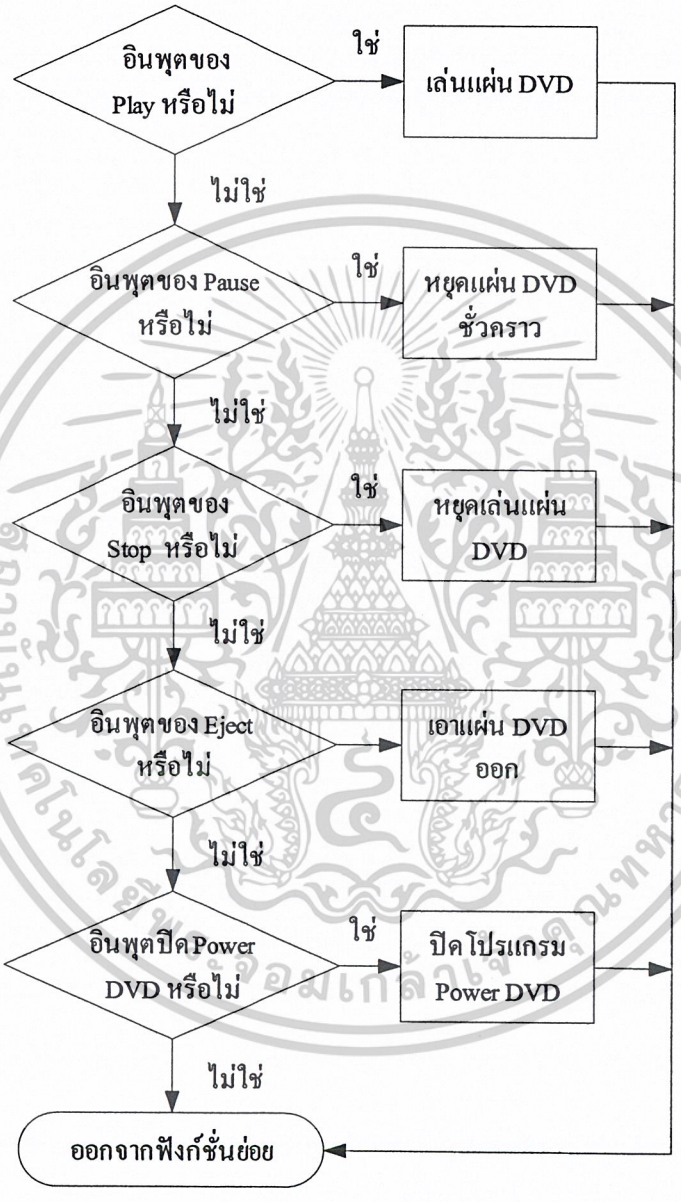
รูปที่ 3.10 แสดงโฟล์วชาร์ทการทำงานของโปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Winamp

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

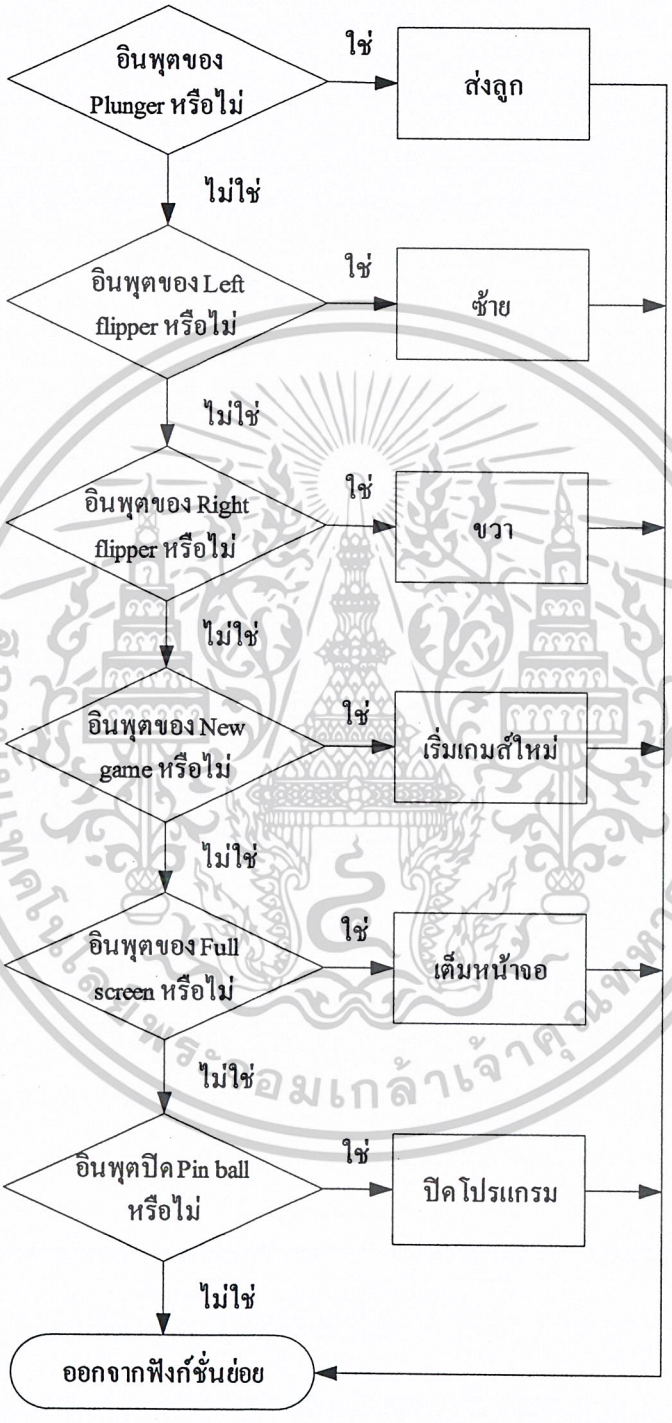


รูปที่ 3.12 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Power DVD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

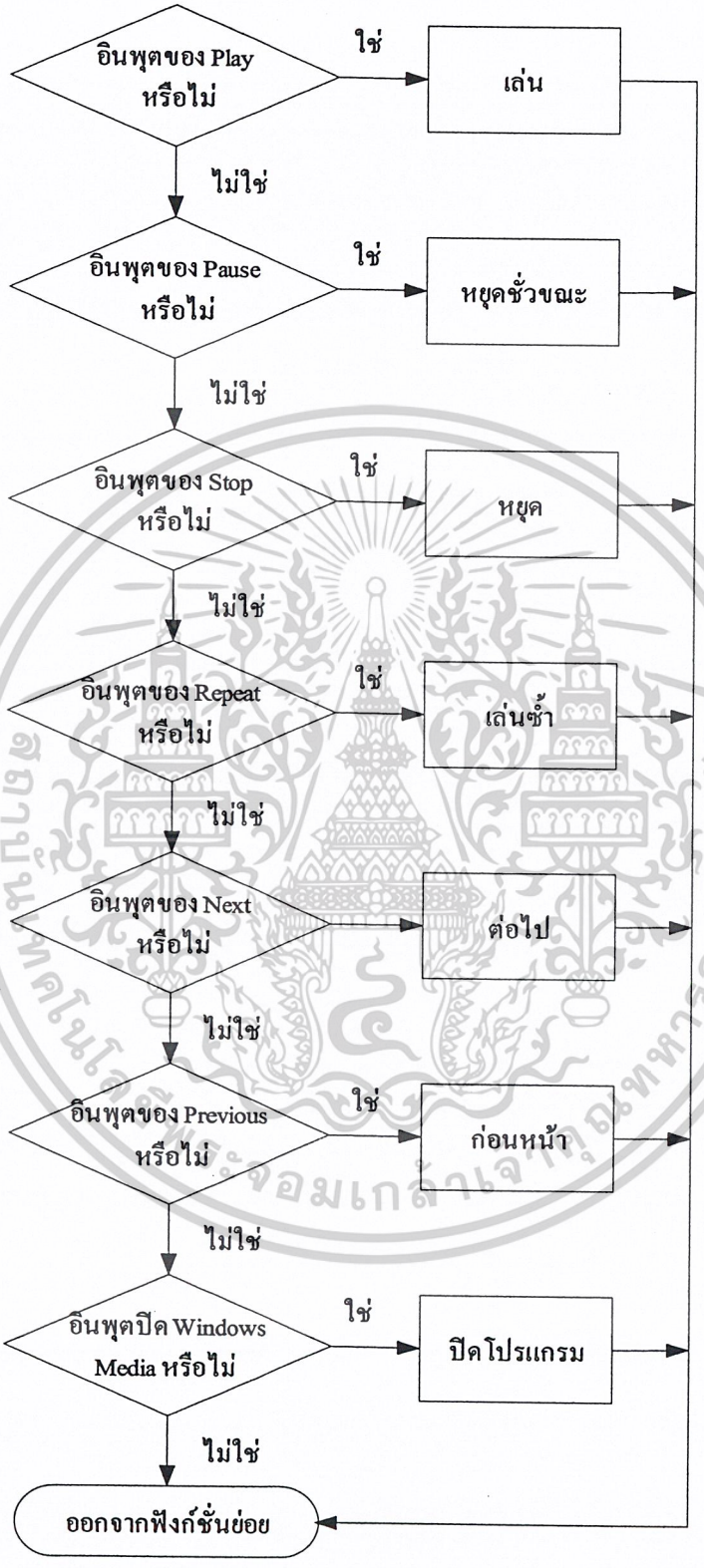
รูปที่ 3.13 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Power Point

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



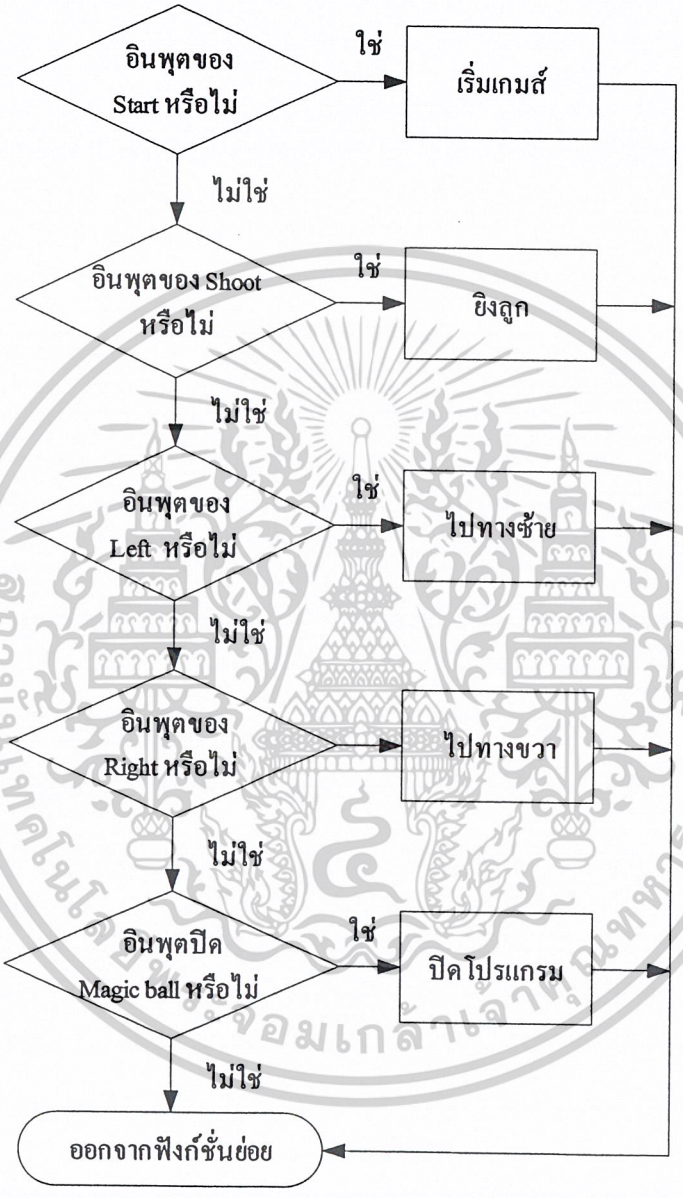
รูปที่ 3.14 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Pin ball

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Windows Media Player

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



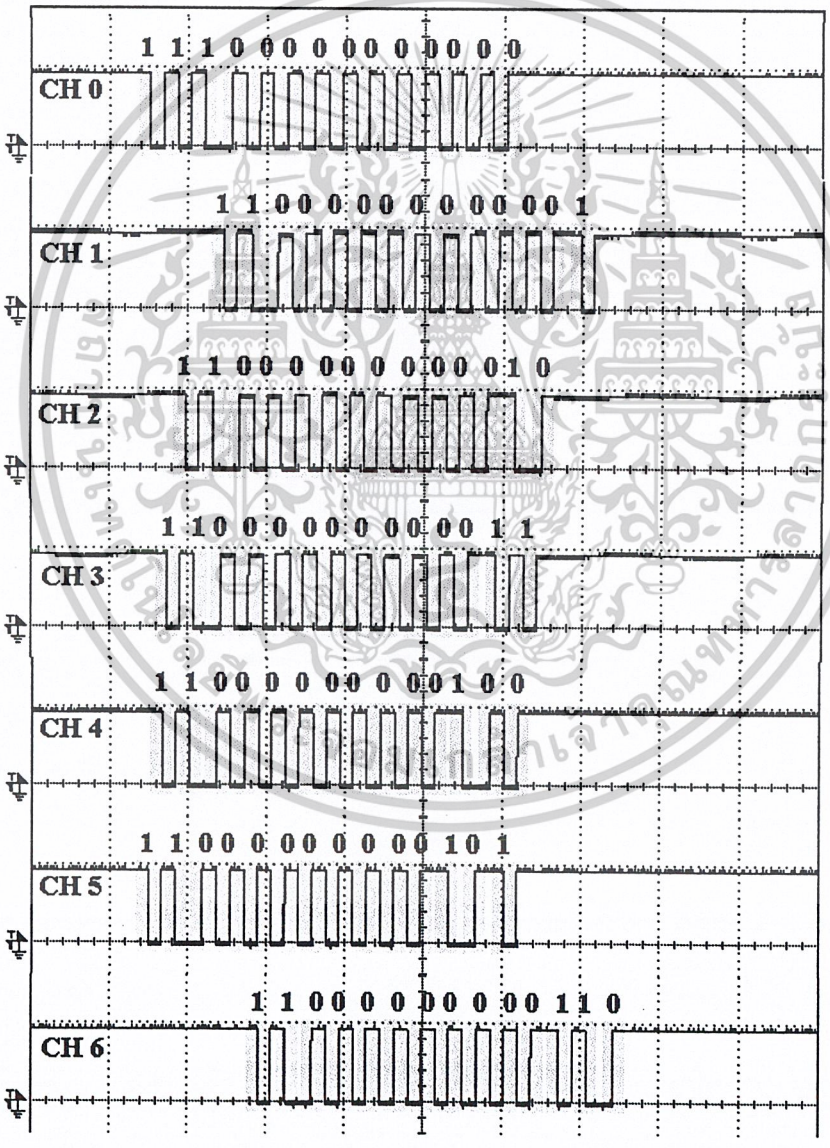
รูปที่ 3.16 แสดงโฟล์วชาร์ทการทำงานของฟังก์ชั่นย่อยของโปรแกรม Magic ball

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 ผลการทดลองส่วนรับสัญญาณ

ในภาครับสัญญาณได้ทำการทดลองและได้ทำการเก็บค่าสัญญาณจากการกดปุ่มต่าง ๆ ของรีโมท โดยเลือกใช้ IC เบอร์ IRM-8601S เป็นตัวรับสัญญาณอินฟราเรด จากรูปที่ 4.1 จะเห็นได้ว่าสัญญาณที่รับได้มานั้นเมื่อผ่าน IC เบอร์ IRM-8601S แล้วนั้นเอาท์พุตที่ได้จะกลับเฟส 180 องศา

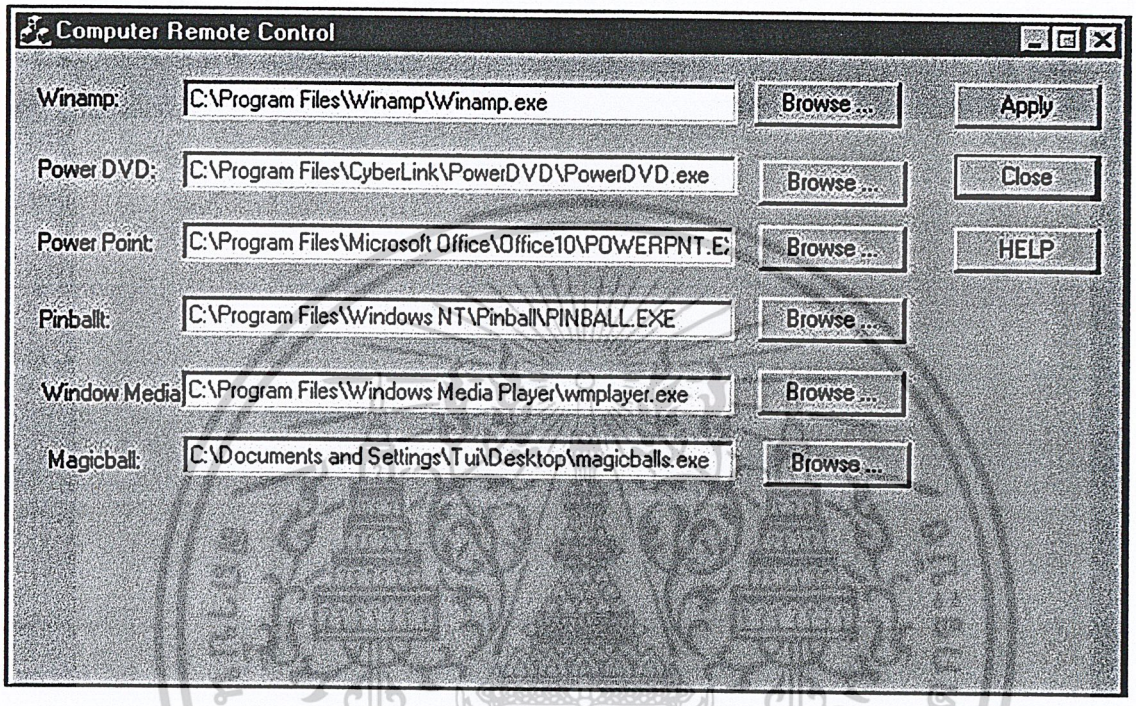


รูปที่ 4.1 รูปตัวอย่างสัญญาณที่วัดจากเอาท์พุตของโฟโต้โมดูล และค่าที่ถูกถอดรหัสออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองส่วนการเขียนโปรแกรมในคอมพิวเตอร์

เมื่อเปิดโปรแกรมขึ้นมา จะมีหน้าต่างของโปรแกรมดังรูปที่ 4.2 ซึ่งในความจริงแล้วหน้าต่างดังกล่าวจะถูกย่อไว้ไม่ให้แสดงขึ้นมา จะต้องคลิกที่แถบเครื่องมือล่างสุดของวินโดวส์ถึงจะเห็นหน้าตาของโปรแกรม



รูปที่ 4.2 แสดงหน้าต่างโปรแกรมการใช้งาน

จากรูปที่ 4.2 จะเห็นว่ามีปุ่มและมีช่องต่าง ๆ ในหน้าต่างของโปรแกรม ซึ่งมีหน้าที่การใช้งานดังนี้

- ช่อง Winamp ใช้ระบุตำแหน่งของโปรแกรม Winamp ในคอมพิวเตอร์
- ช่อง Power DVD ใช้ระบุตำแหน่งของโปรแกรม Power DVD ในคอมพิวเตอร์
- ช่อง Power Point ใช้ระบุตำแหน่งของโปรแกรม Power Point ในคอมพิวเตอร์
- ช่อง Pinball ใช้ระบุตำแหน่งของโปรแกรม Pinball ในคอมพิวเตอร์
- ช่อง Window Media ใช้ระบุตำแหน่งของโปรแกรม Windows Media Player ในคอมพิวเตอร์
- ช่อง Magicball ใช้ระบุตำแหน่งของโปรแกรม Magic ball ในคอมพิวเตอร์
- ปุ่ม Browse... ใช้ในการค้นหาตำแหน่งของโปรแกรมต่าง ๆ ในคอมพิวเตอร์
- ปุ่ม Apply ใช้สำหรับอัพเดทข้อมูลเมื่อมีการเปลี่ยนแปลงข้อมูลในช่อง
- ปุ่ม Close ใช้ปิดโปรแกรมเมื่อเลิกการใช้งาน
- ปุ่ม HELP ใช้สำหรับการเปิดดูวิธีการใช้รีโมทสำหรับโปรแกรมทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 แสดงหน้าต่างเมื่อมีการกดปุ่มผิด หรือปุ่มที่ไม่สามารถใช้งานได้

จากรูปที่ 4.3 เป็นหน้าต่างที่จะเกิดขึ้นมาเมื่อเวลาเริ่มโปรแกรมแล้วผู้ใช้ไม่ได้กดปุ่มที่เป็นโปรแกรม คือ ปุ่ม "1" - "6" นั้นเอง ซึ่งสามารถปิดหน้าต่างนี้ได้ 2 วิธี ก็คือ ใช้เมาส์คลิ๊กที่ปุ่ม OK หรือใช้รีโมทกดที่ปุ่ม "PSM"

4.3 วิธีการใช้โปรแกรม

สำหรับวิธีการใช้รีโมทเพื่อควบคุมโปรแกรมในคอมพิวเตอร์นั้น แต่ละปุ่มก็จะมีหน้าที่การใช้งานแตกต่างกันออกไปตามชนิดของโปรแกรม ดังนี้

4.2.1.1 โปรแกรมเกี่ยวกับเสียง

เป็นปุ่มที่สามารถกดได้ไม่ว่าจะอยู่โปรแกรมไหนก็ตาม

ปุ่ม "POWER"	ปิดคอมพิวเตอร์ (Shut down)
ปุ่ม "MUTE"	ปิดเสียง (Mute)
ปุ่ม "PR→"	เพิ่มเสียง (Volume up)
ปุ่ม "PR←"	ลดเสียง (Volume down)

4.2.1.2 โปรแกรม Winamp

ปุ่ม "1"	เรียกโปรแกรม
ปุ่ม "OK"	เล่นเพลง (Play)
ปุ่ม "7"	หยุดชั่วขณะ (Pause)
ปุ่ม "8"	หยุด (Stop)
ปุ่ม "9"	เปิดแฟ้มเพลง (Open folder)
ปุ่ม "PR↑"	เพลงต่อไป (Previous)
ปุ่ม "PR↓"	เพลงก่อนหน้า (Next)
ปุ่ม "SLEEP"	ปิดโปรแกรม (Close)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.3 โปรแกรม Power DVD

ปุ่ม “2”	เรียกโปรแกรม
ปุ่ม “OK”	เล่นวีซีดี (Play)
ปุ่ม “7”	หยุดชั่วขณะ (Pause)
ปุ่ม “8”	หยุด (Stop)
ปุ่ม “9”	เปิดถาดใส่แผ่น (Eject)
ปุ่ม “SYSTEM”	ปิดโปรแกรม (Close)

4.2.1.4 โปรแกรม Power Point

ปุ่ม “3”	เรียกโปรแกรม
ปุ่ม “OK”	นำเสนอภาพนิ่ง (Present)
ปุ่ม “7”	เปิดไฟล์ภาพนิ่ง (Open file)
ปุ่ม “8”	บันทึกภาพนิ่ง (Save)
ปุ่ม “9”	สั่งพิมพ์ภาพนิ่ง (Print)
ปุ่ม “PR↑”	หน้าภาพนิ่งก่อนหน้า (Previous)
ปุ่ม “PR↓”	หน้าภาพนิ่งต่อไป (Next)
ปุ่ม “SLEEP”	ปิดโปรแกรม (Close)

4.2.1.5 โปรแกรม Pin ball

ปุ่ม “4”	เรียกโปรแกรม
ปุ่ม “OK”	ส่งลูก (Plunger)
ปุ่ม “7”	ซ้าย (Left flipper)
ปุ่ม “8”	ขวา (Right flipper)
ปุ่ม “9”	เริ่มเกมส์ใหม่ (New game)
ปุ่ม “-,- -”	เต็มหน้าจอ (Full screen)
ปุ่ม “SLEEP”	ปิดโปรแกรม (Close)

4.2.1.6 โปรแกรม Windows Media Player

ปุ่ม “5”	เรียกโปรแกรม
ปุ่ม “OK”	เล่น (Play)
ปุ่ม “7”	หยุดชั่วขณะ (Pause)

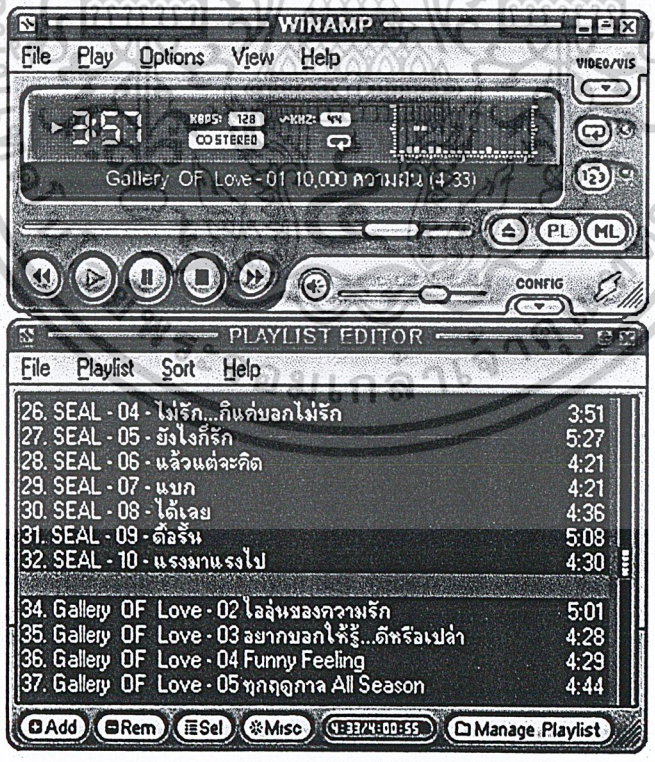
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปุ่ม “8”	หยุด (Stop)
ปุ่ม “9”	เล่นซ้ำ (Repeat)
ปุ่ม “PR↑”	ต่อไป (Previous)
ปุ่ม “PR↓”	ก่อนหน้า (Next)
ปุ่ม “SLEEP”	ปิดโปรแกรม (Close)

4.2.1.7 โปรแกรม Magic ball

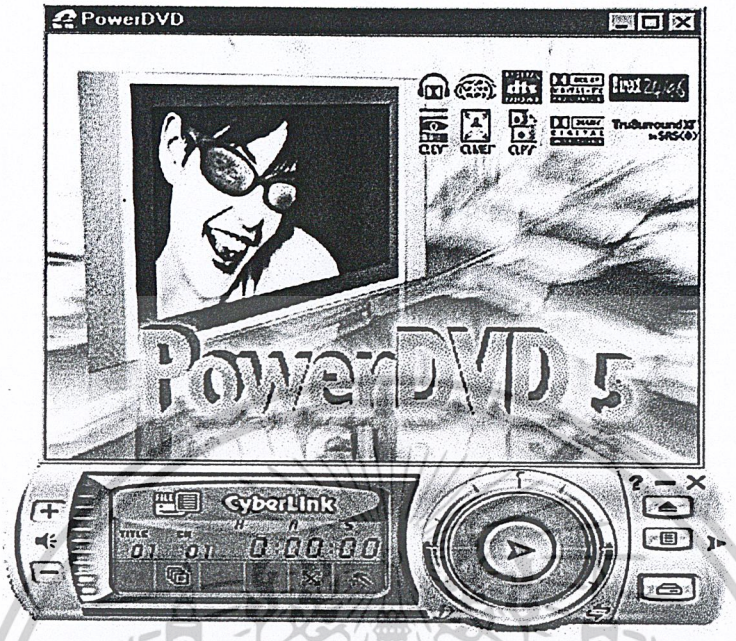
ปุ่ม “6”	เรียกโปรแกรม
ปุ่ม “OK”	เล่นเกมส์, ยิงบอล (Play, shoot)
ปุ่ม “7”	ซ้าย (Left)
ปุ่ม “8”	ขวา (Right)
ปุ่ม “SLEEP”	ปิดโปรแกรม (Close)

ซึ่งวิธีการใช้ต่าง ๆ นี้ สามารถดูได้จาก HELP ของโปรแกรม โดยถ้าเป็นวิธีการใช้รวมทั้งหมดกดที่ปุ่ม “TV/AV” ถ้าเป็นของแต่ละโปรแกรมให้กดปุ่ม “MENU” เมื่ออยู่ในโปรแกรมนั้น ๆ

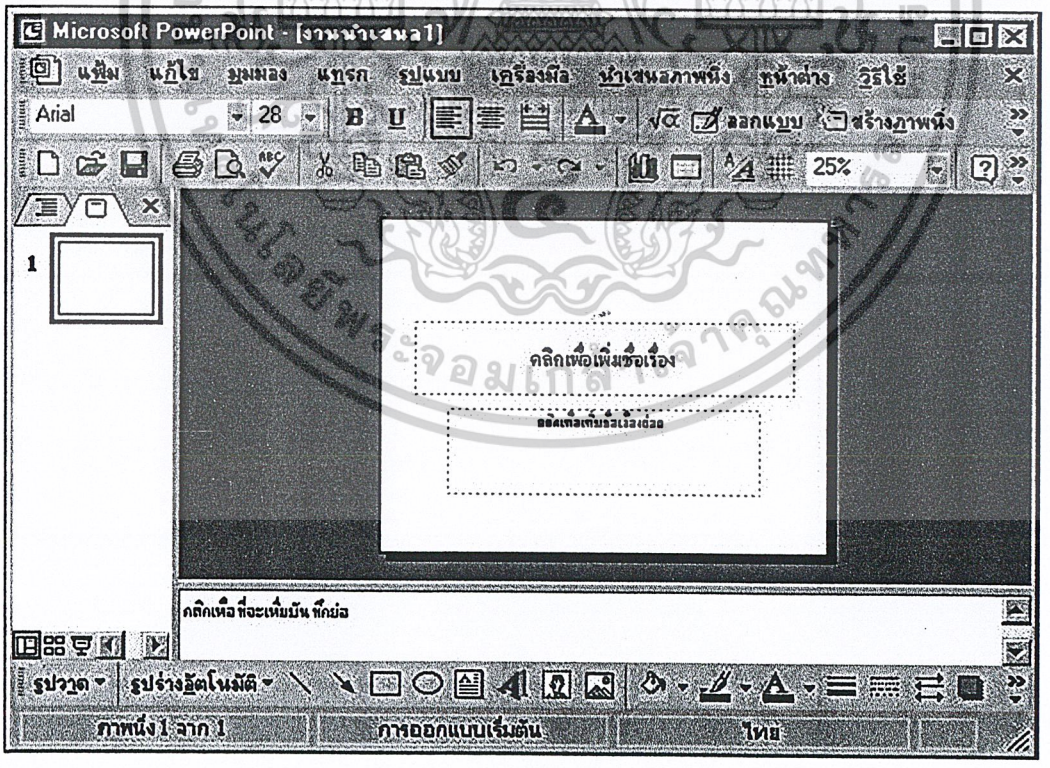


รูปที่ 4.4 แสดงหน้าตาโปรแกรม Winamp ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



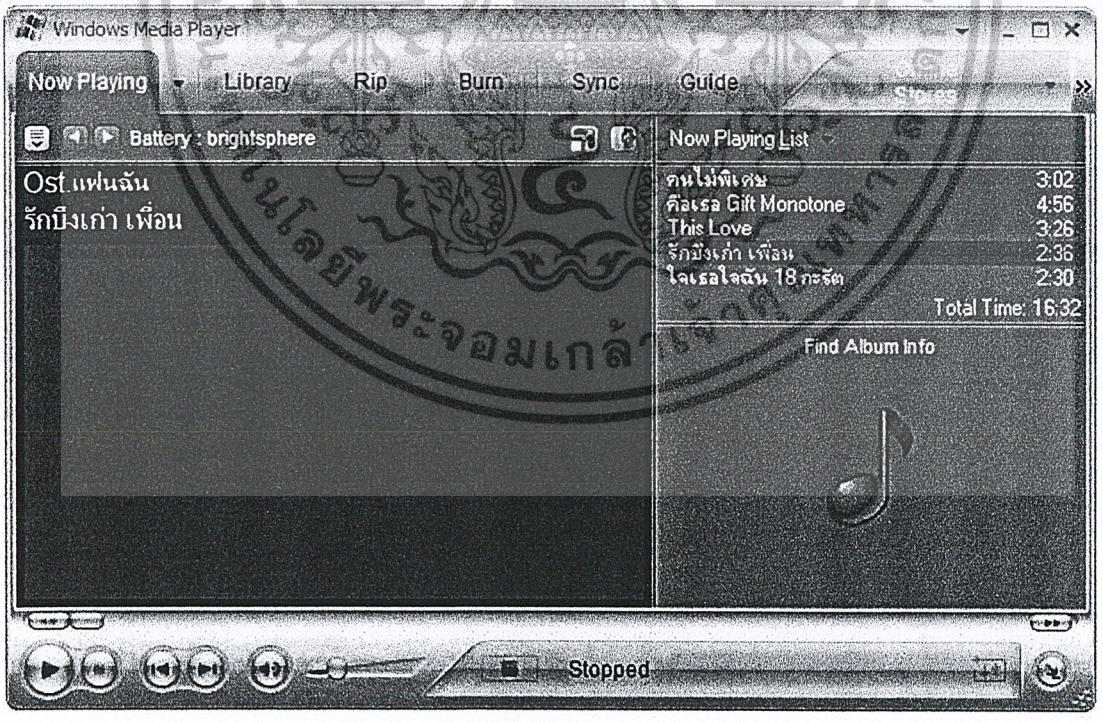
รูปที่ 4.5 แสดงหน้าตาโปรแกรม Power DVD ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “2”



รูปที่ 4.6 แสดงหน้าตาโปรแกรม Power Point ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “3”

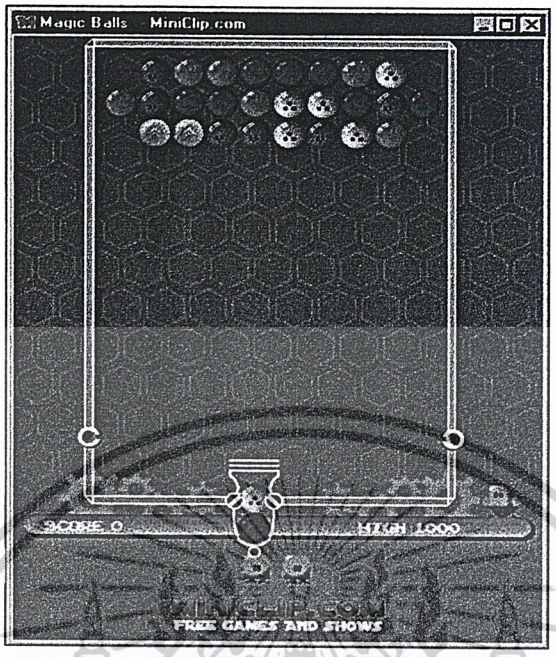
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 แสดงหน้าตาโปรแกรม Pinball ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “4”



รูปที่ 4.8 แสดงหน้าตาโปรแกรม Windows Media Player ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “5”

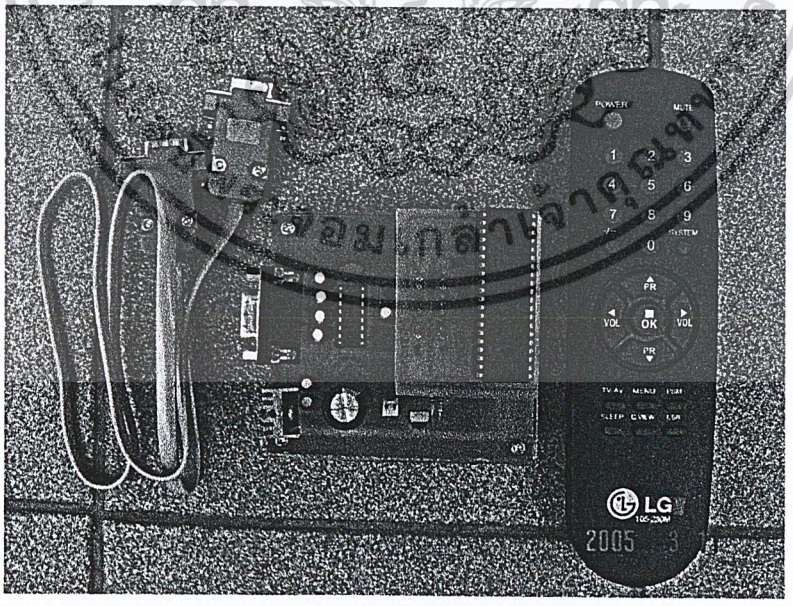
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงหน้าตาโปรแกรม Magic ball ที่ใช้ในการทดลองเมื่อมีการกดรีโมทที่ปุ่ม “6”

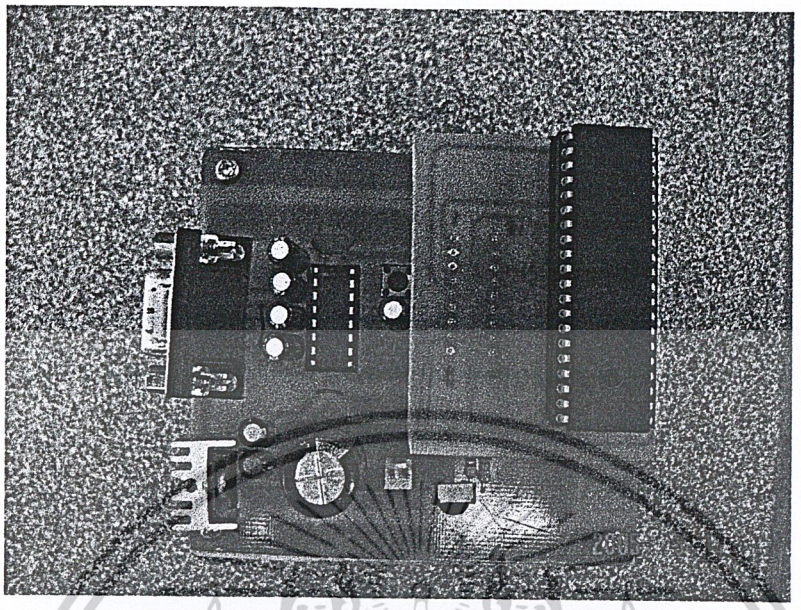
4.4 วงจรและอุปกรณ์ที่ใช้

วงจรและอุปกรณ์ที่ใช้ มี สายต่อพอร์ตอนุกรม, วงจรรับและถอดรหัส, รีโมททีวีทั่วไป

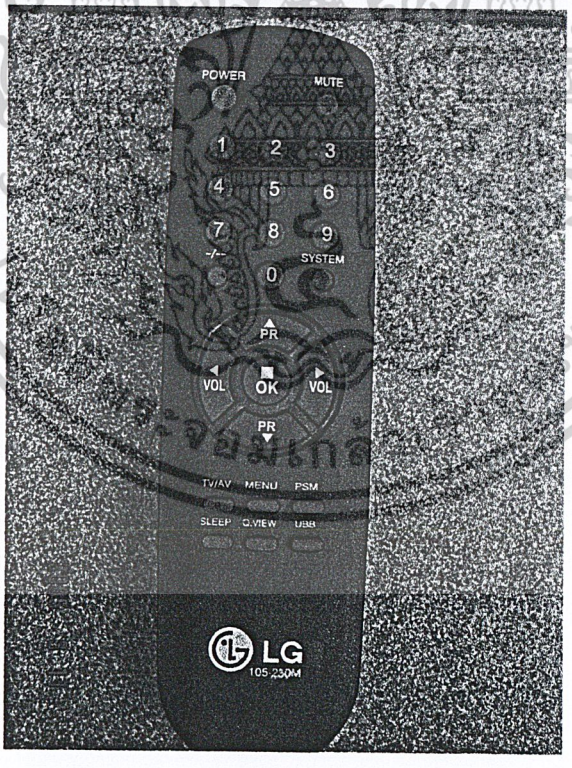


รูปที่ 4.10 แสดงรูปวงจรและอุปกรณ์ทั้งหมดที่ใช้ในโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงวงจรที่ใช้ในการรับสัญญาณ, ถอดรหัส และส่งข้อมูลไปยังคอมพิวเตอร์



รูปที่ 4.12 แสดงรูปรีโมทที่ใช้ในการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและบทวิจารณ์

5.1 สรุปผลการทดลอง

จากการทดลองพบว่าสามารถใช้รีโมทควบคุมโปรแกรมในคอมพิวเตอร์ได้ โดยทำการป้อนที่อยู่ของโปรแกรมที่ได้กำหนดไว้ และยังสามารถดูวิธีการใช้รีโมทจากโปรแกรมได้ ส่วนการถอดรหัสก็สามารถถอดรหัสรีโมทได้ และสามารถส่งค่าไปยังคอมพิวเตอร์ได้

5.2 ปัญหาที่พบระหว่างการดำเนินงาน

1. มีข้อจำกัดทางด้านความเร็วในเรื่องการประมวลผล
2. วงจรที่นำมาใช้ถอดรหัสในตอนแรกมีปัญหามากไม่สามารถนำมาใช้ได้ เนื่องจากเอาต์พุต และอินพุตไม่สอดคล้องกัน ทำให้ไม่สามารถถอดรหัสออกมาได้

5.3 แนวทางในการพัฒนาต่อ

1. ปรับปรุงเทคนิคในการถอดรหัส เพื่อให้มีประสิทธิภาพมากขึ้น
2. ออกแบบส่วนถอดรหัสให้สามารถถอดรหัสให้ได้หลายยี่ห้อมากกว่านี้
3. เพิ่มความเร็วในการประมวลผล และในการส่งข้อมูลให้มากขึ้น จะสามารถทำให้ทำงานได้เร็วขึ้น
4. ออกแบบโปรแกรมในคอมพิวเตอร์ให้ยืดหยุ่นกว่านี้ โดยให้สามารถกำหนดปุ่มได้เองว่าจะให้ปุ่มนี้ทำงานอะไรได้บ้าง
5. ออกแบบการเชื่อมต่อกับคอมพิวเตอร์ให้สะดวกกว่านี้ โดยเชื่อมต่อกับพอร์ต USB ได้ เนื่องจากในปัจจุบันพอร์ตอนุกรมไม่เป็นที่นิยมใช้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก. SOURCE CODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code in Microcontroller (PIC16F877)

```
/*****************************************************
*          Program Encoder RC5 (Phillip)
*****************************************************/
#define _PIC16F877A_
#include <16F877A.h>
#define TxD PIN_C6
#define RxD PIN_C7
#define CLOCK_SP 20000000
#define IRM PIN_A3

/*****************************************************
*          Device Specification
*****************************************************/
#fuses HS
#fuses NOLVP,NOWDT
#fuses NOPROTECT
#use delay (clock=CLOCK_SP)
#use rs232(baud=9600, xmit=TxD,rcv=RxD)
#use fast_io(A)

/*****************************************************
*          Develop File
*****************************************************/
const int channel=16;                    //  MEMORY Channel  //
int1 p0,p1,p2,p3,p4,p5,p6,p7,p8,p9,p10,p11,p12;
int8 a,i,time,t0,t1,t2,t3,t4,t5,t6,t7,t8,t9,t10,t11,t12,t13,t14,t15;
int data;

void Print_data(void){
   printf("%d", data);}
void LCD_notuse(void){
   printf (" This bottom is Not Used\n\r");}
void move_add(void){data = data<<1;data = data+1;}
void move(void){data = data<<1;}
void rc5(void){
   while(input(IRM));
   delay_us(3400);
   if(input(IRM)){p1 = 1;}
   delay_us(1700);
   if(input(IRM)){p2 = 1;}
   delay_us(1700);
   if(input(IRM)){p3 = 1;}
   delay_us(1700);
   if(input(IRM)){p4 = 1;}
   delay_us(1700);
   if(input(IRM)){p5 = 1;}
   delay_us(1700);
   if(input(IRM)){p6 = 1;}
   delay_us(1700);
   if(input(IRM)){p7 = 1;}          //check_1
   delay_us(1700);
   if(input(IRM)){p8 = 1;}          //check_2
   delay_us(1700);
   if(input(IRM)){p9 = 1;}          //check_3
   delay_us(1700);
   if(input(IRM)){p10 = 1;}         //check_4
   delay_us(1700);
   if(input(IRM)){p11 = 1;}         //check_5
   delay_us(1700);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   if(input(IRM)){p12 = 1;}          //check_6
}
void rc5_encode(void){
   data=0;
   if(!p7){move_add();} else {move();}
   if(!p8){move_add();} else {move();}
   if(!p9){move_add();} else {move();}
   if(!p10){move_add();} else {move();}
   if(!p11){move_add();} else {move();}
   if(!p12){move_add();} else {move();}
}
/***************************************
*          Clear Variable
***************************************/
void
clear_tp(void){p1=0;p2=0;p3=0;p4=0;p5=0;p6=0;p7=0;p8=0;p9=0;p10=0;p11
=0;p12=0;t0=0;t1=0;t2=0;t3=0;
t4=0;t5=0;t6=0;t7=0;t8=0;t9=0;t10=0;t11=0;t12=0;t13=0;t14=0;t15=0;}
/***************************************
           START MAIN PROGRAM
***************************************/
void main(void){
SET_TRIS_A( 0xFF );
setup_timer_0(RTCC_INTERNAL | RTCC_DIV_64);
start://LCD_Show1();
clear_tp();
time=0x00;
while(input(IRM));
set_timer0(0);
while(!input(IRM));
time = get_timer0();

/***************************************
           Encode
***************************************/
  delay_ms(1000);
  rc5();
  rc5_encode();
        switch (data) {
           case 1:Print_data();goto start;  //use
           case 2:Print_data();goto start;  //use
           case 3:Print_data();goto start;  //use
           case 4:Print_data();goto start;  //use
           case 5:Print_data();goto start;  //use
           case 6:Print_data();goto start;  //use
           case 7:Print_data();goto start;  //use
           case 8:Print_data();goto start;  //use
           case 9:Print_data();goto start;  //use
           case 0:Print_data();goto start;  //use

           case 10:printf("F");goto start;
           case 12:printf("S");;goto start;  //use
           case 13:printf("M");goto start;   //use
           case 14:printf("P");goto start;
           case 15:printf("Y");goto start;
           case 16:printf("U");goto start;   //use
           case 17:printf("D");goto start;   //use
           case 32:printf("A");goto start;   //use
           case 33:printf("B");goto start;   //use
           case 34:printf("Q");goto start;
           case 36:printf("C");goto start;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case 37:printf("K");goto start;
            case 38:printf("L");goto start;
            case 56:printf("V");goto start;
            case 59:printf("E");goto start;
          }
          LCD_notuse();
  goto start;

}
```

Source Code Visual C++

```
// AcrylicDlg.cpp : implementation file
#include "stdafx.h"
#include "Acrylic.h"
#include "AcrylicDlg.h"
#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif
/////////////////////////////////////////////////////////////////////////////
// CAcrylicDlg dialog
CAcrylicDlg::CAcrylicDlg(CWnd* pParent /*=NULL*/)
      : CDialog(CAcrylicDlg::IDD, pParent)
{
      //{{AFX_DATA_INIT(CAcrylicDlg)
      m_Winamp = _T("");
      m_DVD = _T("");
      m_PowerPt = _T("");
      m_Pinball = _T("");
      m_Winmedia = _T("");
      m_Solitriae = _T("");
      //}}AFX_DATA_INIT
      // Note that LoadIcon does not require a subsequent DestroyIcon
in Win32
      m_hIcon = AfxGetApp()->LoadIcon(IDR_MAINFRAME);
}
void CAcrylicDlg::DoDataExchange(CDataExchange* pDX)
{
      CDialog::DoDataExchange(pDX);
      //{{AFX_DATA_MAP(CAcrylicDlg)
      DDX_Control(pDX, IDC_MSCOMM, m_Comm);
      DDX_Text(pDX, IDC_WINAMP, m_Winamp);
      DDX_Text(pDX, IDC_DVD, m_DVD);
      DDX_Text(pDX, IDC_POWERPT, m_PowerPt);
      DDX_Text(pDX, IDC_PINBALL, m_Pinball);
      DDX_Text(pDX, IDC_WINMEDIA, m_Winmedia);
      DDX_Text(pDX, IDC_SOLITRIAE, m_Solitriae);
      //}}AFX_DATA_MAP
}
BEGIN_MESSAGE_MAP(CAcrylicDlg, CDialog)
      //{{AFX_MSG_MAP(CAcrylicDlg)
      ON_WM_PAINT()
      ON_WM_QUERYDRAGICON()
      ON_WM_CLOSE()
      ON_BN_CLICKED(IDC_APPLY, OnApply)
      ON_BN_CLICKED(IDC_B_WINAMP, OnBWinamp)
      ON_BN_CLICKED(IDC_B_DVD, OnBDvd)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ON_BN_CLICKED(IDC_B_POWERPT, OnBPowerpt)
	ON_BN_CLICKED(IDC_B_HELP, OnHelp)
	ON_BN_CLICKED(IDC_B_WINMEDIA, OnBWinmedia)
	ON_BN_CLICKED(IDC_B_PINBALL, OnBPinball)
	ON_BN_CLICKED(IDC_B_SOLITRIAE, OnBSolitriae)
	//}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////////////////////////////////////////
////////
// CAcrylicDlg message handlers
BOOL CAcrylicDlg::OnInitDialog()
{
	CDialog::OnInitDialog();
	// Set the icon for this dialog.  The framework does this
automatically
	//  when the application's main window is not a dialog
	SetIcon(m_hIcon, TRUE);			// Set big icon
	SetIcon(m_hIcon, FALSE);		// Set small icon
	ShowWindow(SW_SHOWMINIMIZED);
	// Default value
	m_Alerted = FALSE;
	m_Winamp = "C:\\Program Files\\Winamp\\Winamp.exe";
	m_DVD = "C:\\Program Files\\CyberLink\\PowerDVD\\PowerDVD.exe";
	m_PowerPt = "C:\\Program Files\\Microsoft
Office\\Office10\\POWERPNT.EXE";
	m_Pinball="C:\\Program Files\\Windows
NT\\Pinball\\PINBALL.EXE";
	m_Winmedia = "C:\\Program Files\\Windows Media
Player\\wmplayer.exe" ;
	m_Solitriae = "C:\\Documents and
Settings\\Tui\\Desktop\\magicballs.exe";
	// Open from config
	UpdateData(FALSE);
	// TODO: Add extra initialization here
	m_Comm.SetPortOpen(TRUE);
	return TRUE;  // return TRUE  unless you set the focus to a
control
}
// If you add a minimize button to your dialog, you will need the
code below
//  to draw the icon.  For MFC applications using the document/view
model,
//  this is automatically done for you by the framework.
void CAcrylicDlg::OnPaint()
{
	if (IsIconic())
	{
		CPaintDC dc(this); // device context for painting
		SendMessage(WM_ICONERASEBKGND, (WPARAM) dc.GetSafeHdc(),
0);
		// Center icon in client rectangle
		int cxIcon = GetSystemMetrics(SM_CXICON);
		int cyIcon = GetSystemMetrics(SM_CYICON);
		CRect rect;
		GetClientRect(&rect);
		int x = (rect.Width() - cxIcon + 1) / 2;
		int y = (rect.Height() - cyIcon + 1) / 2;
		// Draw the icon
		dc.DrawIcon(x, y, m_hIcon);
	}
	else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	{
		CDialog::OnPaint();
	}
}
// The system calls this to obtain the cursor to display while the
user drags
//  the minimized window.
HCURSOR CAcrylicDlg::OnQueryDragIcon()
{
	return (HCURSOR) m_hIcon;
}
void CAcrylicDlg::OnClose()
{
	// TODO: Add your message handler code here and/or call default
	m_Comm.SetPortOpen(FALSE);
	// Save to config
	CDialog::OnClose();
}
BEGIN_EVENTSINK_MAP(CAcrylicDlg, CDialog)
    //{{AFX_EVENTSINK_MAP(CAcrylicDlg)
	ON_EVENT(CAcrylicDlg, IDC_MSCOMM, 1 /* OnComm */, OnComm,
VTS_NONE)
	//}}AFX_EVENTSINK_MAP
END_EVENTSINK_MAP()
void CAcrylicDlg::OnComm()    // Receive from serial port
{
	//if (m_Alerted) return;//wait for  later value buttom
	// TODO: Add your control notification handler code here
	static char Prg = '0';  // Current program
	VARIANT InChar = m_Comm.GetInput();
	char C = (char)InChar.bstrVal[0];
	static BOOL plunger_pressed = FALSE;//first status for plunger
bottom
	if (m_Alerted) {
		if (C == 'P') {
			keybd_event(0x0D,0,0,0);
			keybd_event(0x0D,0,KEYEVENTF_KEYUP,0);
	} else return;}
	// Special button
	if (C == 'M') {     // Mute
		keybd_event(0xAD,0,0,0);
		keybd_event(0xAD,0,KEYEVENTF_KEYUP,0);
	} else if (C == 'U') {    // Volume up
		keybd_event(0xAF,0,0,0);
		keybd_event(0xAF,0,KEYEVENTF_KEYUP,0);
	} else if (C == 'D') {    // Volume down
		keybd_event(0xAE,0,0,0);
		keybd_event(0xAE,0,KEYEVENTF_KEYUP,0);
	} else if (C == 'S') {    // Shut down
		::WinExec(  "shutdown -s -f -t 5",SW_SHOW);
	} else if (((C >= '1') && (C <= '6')) || (C == 'V') || (C ==
'P') ){
		switch (C) {
		case '1':     // Winamp
			ShowWindow(SW_SHOWMINIMIZED);
			::WinExec(m_Winamp, SW_SHOW);
			Prg = '1';
			break;
		case '2':     // PowerDVD
			ShowWindow(SW_SHOWMINIMIZED);
			::WinExec(m_DVD,SW_SHOW);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Prg = '2';
                break;
            case '3':   // Powerpoint
                ShowWindow(SW_SHOWMINIMIZED);
                ::WinExec(m_PowerPt,SW_SHOW);
                Prg = '3';
                break;
            case '4':  //Pinball
                ShowWindow(SW_SHOWMINIMIZED);
                ::WinExec(m_Pinball,SW_SHOW);
                Prg = '4';
                break;
            case '5':  //Winmedia
                ShowWindow(SW_SHOWMINIMIZED);
                ::WinExec(m_Winmedia,SW_SHOW);
                Prg = '5';
                break;
            case '6':  //magicball
                ShowWindow(SW_SHOWMINIMIZED);
                ::WinExec(m_Solitriae,SW_SHOW);
                Prg = '6';
                break;
            case 'V':  //helpall
                ShowWindow(SW_SHOWMINIMIZED);
                ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe helpall.rtf",SW_SHOW);
                Prg = 'V';
                break;
            case 'P':  // Enter
                keybd_event(0x0D,0,0,0);
                keybd_event(0x0D,0,KEYEVENTF_KEYUP,0);
                break;
            }
        } else if (((C >= '7') && (C <= '9')) || (C == 'L') || (C ==
'A') || (C == 'B') || (C == 'K') || (C == 'E') || (C == 'Y') || (C ==
'F') || (C == 'Q') || (C == 'C')) {
            switch (Prg) {
            case '1': // Winamp
                switch (C) {
                case 'K':           //Play
                    keybd_event(0x58,0,0,0);
                  keybd_event(0x58,0,KEYEVENTF_KEYUP,0);
                    break;
                case '7':           //Pause
                    keybd_event(0x43,0,0,0);
                  keybd_event(0x43,0,KEYEVENTF_KEYUP,0);
                    break;
                case '8':           //Stop
                    keybd_event(0x56,0,0,0);
                  keybd_event(0x56,0,KEYEVENTF_KEYUP,0);
                    break;
                case '9':           //Open folder
                    keybd_event(0x10,0,0,0);
                    keybd_event(0x4C,0,0,0);
                keybd_event(0x4C,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x10,0,KEYEVENTF_KEYUP,0);
                    break;
                case 'B':           //Next
                    keybd_event(0x42,0,0,0);
                  keybd_event(0x42,0,KEYEVENTF_KEYUP,0);
                    break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case 'A':           //Previous
                      keybd_event(0x5A,0,0,0);
                     keybd_event(0x5A,0,KEYEVENTF_KEYUP,0);
                      break;
                case 'L':           // Close program
                      keybd_event(0x12,0,0,0);
                keybd_event(0x73,0,0,0);
                keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                      break;
                case 'E':                 // Help
                      ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe winamp.rtf", SW_SHOW);
                      break;
                default:

                      if (m_Alerted) return;
                      m_Alerted = TRUE;
                      MessageBox("Invalid Key.\n Press '1' - '6'
button only.\n Press 'PSM' to exit.", "Warning", MB_OK |
MB_SYSTEMMODAL | MB_ICONEXCLAMATION);
                      m_Alerted = FALSE;
                }
                break;
          case '2':   // PowerDVD
                switch (C) {
                      case 'K':         // Play
                      keybd_event(0x0D,0,0,0);
                     keybd_event(0x0D,0,KEYEVENTF_KEYUP,0);
                      break;
                case '7':               // Pause
                      keybd_event(0x20,0,0,0);
                    keybd_event(0x20,0,KEYEVENTF_KEYUP,0);
                      break;
                 case '8':              // Stop
                      keybd_event(0x53,0,0,0);
                keybd_event(0x53,0,KEYEVENTF_KEYUP,0);
                      break;
          case '9':             // Eject
                keybd_event(0x11,0,0,0);
                keybd_event(0x45,0,0,0);
                keybd_event(0x45,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                break;

          case 'L':           // Close program
                keybd_event(0x12,0,0,0);
                keybd_event(0x73,0,0,0);
                keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                    break;
                case 'E':                 // Help
                      ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe powerdvd.rtf", SW_SHOW);
                      break;
                case 'Y':           // Close Help
                      keybd_event(0x11,0,0,0);
                keybd_event(0x58,0,0,0);
                keybd_event(0x58,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                      break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                default:
                        //m_Alerted = TRUE;
                        //if (m_Alerted) return;
                        MessageBox("Invalid Key.\n Press '1' - '6'
button only.\n Press 'PSM' to exit.", "Warning", MB_OK |
MB_SYSTEMMODAL | MB_ICONEXCLAMATION);
                        m_Alerted = FALSE;
                }
                break;
            case '3': // Powerpoint
                switch (C) {
                case '7':   // Open
                keybd_event(0x11,0,0,0);
                keybd_event(0x4F,0,0,0);
                keybd_event(0x4F,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                    break;
            case '8':   // Save
                keybd_event(0x11,0,0,0);
                keybd_event(0x53,0,0,0);
                keybd_event(0x53,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                break;
            case '9':   // Print
                keybd_event(0x11,0,0,0);
                keybd_event(0x50,0,0,0);
                keybd_event(0x50,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                break;
             case 'K':   // Present
                keybd_event(0x74,0,0,0);
                keybd_event(0x74,0,KEYEVENTF_KEYUP,0);
                break;
            case 'B':   // Next
               keybd_event(0x20,0,0,0);
                keybd_event(0x20,0,KEYEVENTF_KEYUP,0);
                    break;
              case 'A':   // Previous
                    keybd_event(0x08,0,0,0);
                keybd_event(0x08,0,KEYEVENTF_KEYUP,0);
                break;
            case 'L':   // Close
                keybd_event(0x12,0,0,0);
                keybd_event(0x73,0,0,0);
                keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                break;
                case 'E':   //Help
                    ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe powerpoint.rtf", SW_SHOW);
                    break;
                default:

                    if (m_Alerted) return;
                    m_Alerted = TRUE;
                    MessageBox("Invalid Key.\n Press '1' - '6'
button only.\n Press 'PSM' to exit.", "Warning", MB_OK |
MB_SYSTEMMODAL | MB_ICONEXCLAMATION);
                    m_Alerted = FALSE;
                }
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            case '4': // Pinball
                  switch (C) {
                  case 'K':           //Plunger
                        if (plunger_pressed)
                              keybd_event(0x20,0,KEYEVENTF_KEYUP,0);
                        else
                              keybd_event(0x20,0,0,0);
                        plunger_pressed = !plunger_pressed;
                        break;
                  case '7':           //Left
                        keybd_event(0x5A,0,0,0);
                        ::Sleep(150);
                      keybd_event(0x5A,0,KEYEVENTF_KEYUP,0);
                        break;
                  case '8':           //Right
                        keybd_event(0xBF,0,0,0);
                        ::Sleep(150);
                      keybd_event(0xBF,0,KEYEVENTF_KEYUP,0);
                        break;
                  case '9':           //New game
                        keybd_event(0x71,0,0,0);
                        keybd_event(0x71,0,KEYEVENTF_KEYUP,0);
                        break;
                  case 'F':           //full screen
                        keybd_event(0x73,0,0,0);
                        keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                        break;
                  case 'L':           // Close program
                        keybd_event(0x12,0,0,0);
                  keybd_event(0x73,0,0,0);
                  keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                  keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                        break;
                  case 'E':           // Help
                        ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe pinball.rtf", SW_SHOW);
                        break;
                  default:

                        if (m_Alerted) return;
                        m_Alerted = TRUE;
                        MessageBox("Invalid Key.\n Press '1' - '6'
button only.\n Press 'PSM' to exit.", "Warning", MB_OK |
MB_SYSTEMMODAL | MB_ICONEXCLAMATION);
                        m_Alerted = FALSE;
                  }
                  break;


            case '5': // Window media player
                  switch (C) {

                  case 'K':           //Play
                        keybd_event(0x11,0,0,0);
                  keybd_event(0x50,0,0,0);
                  keybd_event(0x50,0,KEYEVENTF_KEYUP,0);
                  keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                        break;
                  case '7':           //Pause
                        keybd_event(0x11,0,0,0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        keybd_event(0x50,0,0,0);
                        keybd_event(0x50,0,KEYEVENTF_KEYUP,0);
                        keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                              break;
                        case '8':           //Stop
                              keybd_event(0x11,0,0,0);
                        keybd_event(0x53,0,0,0);
                        keybd_event(0x53,0,KEYEVENTF_KEYUP,0);
                        keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                              break;
                        case '9':           //repeat
                              keybd_event(0x11,0,0,0);
                              keybd_event(0x54,0,0,0);
                        keybd_event(0x54,0,KEYEVENTF_KEYUP,0);
                        keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                              break;
                        case 'B':           //Next
                              keybd_event(0x11,0,0,0);
                              keybd_event(0x46,0,0,0);
                        keybd_event(0x46,0,KEYEVENTF_KEYUP,0);
                        keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                              break;
                        case 'A':           //Previous
                              keybd_event(0x11,0,0,0);
                              keybd_event(0x42,0,0,0);
                        keybd_event(0x42,0,KEYEVENTF_KEYUP,0);
                        keybd_event(0x11,0,KEYEVENTF_KEYUP,0);
                              break;

                        case 'L':           // Close program
                              keybd_event(0x12,0,0,0);
                        keybd_event(0x73,0,0,0);
                        keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                        keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                              break;
                        case 'E':           // Help
                              ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe winmedia.rtf", SW_SHOW);
                              break;

                        default:

                              if (m_Alerted) return;
                              m_Alerted = TRUE;
                              MessageBox("Invalid Key.\n Press '1' - '6'
button only.\n Press 'PSM' to exit.", "Warning", MB_OK |
MB_SYSTEMMODAL | MB_ICONEXCLAMATION);
                              m_Alerted = FALSE;
                        }
                        break;
                  case '6': // magicball
                        switch (C) {
                        case 'K':           //start
                              keybd_event(0x20,0,0,0);
                              keybd_event(0x20,0,KEYEVENTF_KEYUP,0);
                              break;
                        case '7':           //left
                              keybd_event(0x25,0,0,0);
                              keybd_event(0x25,0,KEYEVENTF_KEYUP,0);
                              break;
                        case '8':           //Right
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        keybd_event(0x27,0,0,0);
                    keybd_event(0x27,0,KEYEVENTF_KEYUP,0);
                        break;
                    case '9':           //space
                        keybd_event(0x20,0,0,0);
                    keybd_event(0x20,0,KEYEVENTF_KEYUP,0);
                        break;
                    case 'L':           // Close program
                        keybd_event(0x12,0,0,0);
                    keybd_event(0x73,0,0,0);
                    keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                    keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                        break;
                    case 'E':               // Help
                        ::WinExec("C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe magicball.rtf", SW_SHOW);
                        break;
                    default:

                        if (m_Alerted) return;
                        m_Alerted = TRUE;
                        MessageBox("Invalid Key.\n Press '1' - '6'
button only.\n Press 'PSM' to exit.", "Warning", MB_OK |
MB_SYSTEMMODAL | MB_ICONEXCLAMATION);

                        m_Alerted = FALSE;
                    }

                    break;
                case 'V': // helpall
                    switch (C) {

                    case 'A':           //left arrow
                        keybd_event(0x26,0,0,0);
                    keybd_event(0x26,0,KEYEVENTF_KEYUP,0);
                        break;
                    case 'B':           //right arrow
                        keybd_event(0x28,0,0,0);
                    keybd_event(0x28,0,KEYEVENTF_KEYUP,0);
                        break;
                    case 'C':           //page up
                        keybd_event(0x21,0,0,0);
                    keybd_event(0x21,0,KEYEVENTF_KEYUP,0);
                        break;
                    case 'Q':           //page down
                        keybd_event(0x22,0,0,0);
                    keybd_event(0x22,0,KEYEVENTF_KEYUP,0);
                        break;
                    case 'L':           // Close programx
                        keybd_event(0x12,0,0,0);
                    keybd_event(0x73,0,0,0);
                    keybd_event(0x73,0,KEYEVENTF_KEYUP,0);
                    keybd_event(0x12,0,KEYEVENTF_KEYUP,0);
                        break;

                    }
                    break;

                default:

                    if (m_Alerted) return;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    m_Alerted = TRUE;
                    MessageBox("Invalid Key.\n Press '1' - '6' button
only.\n Press 'PSM' to exit.", "Warning", MB_OK | MB_SYSTEMMODAL |
MB_ICONEXCLAMATION);
                    m_Alerted = FALSE;
              }
       }
       else {

              if (m_Alerted) return;
              m_Alerted = TRUE;
              MessageBox("Invalid Key.\n Press '1' - '6' button only.\n
Press 'PSM' to exit.", "Warning", MB_OK | MB_SYSTEMMODAL |
MB_ICONEXCLAMATION);
              m_Alerted = FALSE;
       }

}

void CAcrylicDlg::OnApply()
{
       // TODO: Add your control notification handler code here
       UpdateData(TRUE);

       // Add backslash
       m_Winamp.Replace("\\", "\\\\");
       m_DVD.Replace("\\", "\\\\");
       m_PowerPt.Replace("\\", "\\\\");
       m_Pinball.Replace("\\","\\\\");
       m_Winmedia.Replace("\\","\\\\");
       m_Solitriae.Replace("\\", "\\\\");
}
void CAcrylicDlg::OnBWinamp()
{
       // TODO: Add your control notification handler code here
       char szExtBuffer[256];
       CString filter = "Executable (*.exe)|*.exe|All files|*.*|";
       filter.Replace('|', 0);
       OPENFILENAME ofn;

       strcpy(szExtBuffer, "");

       ofn.lStructSize = sizeof(ofn);
       ofn.hwndOwner = NULL;
       ofn.hInstance = NULL;
       ofn.lpstrFilter = filter;
       ofn.lpstrCustomFilter = NULL;
       ofn.nMaxCustFilter = NULL;
       ofn.nFilterIndex = 0;
       ofn.lpstrFile = szExtBuffer;
       ofn.nMaxFile = MAX_PATH;
       ofn.lpstrFileTitle = NULL;
       ofn.nMaxFileTitle = NULL;
       ofn.lpstrInitialDir = NULL;
       ofn.lpstrTitle = "Open file";
       ofn.Flags = NULL;
       ofn.nFileOffset = NULL;
       ofn.nFileExtension = NULL;
       ofn.lpstrDefExt = NULL;
       ofn.lCustData = NULL;
       ofn.lpfnHook = NULL;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ofn.lpTemplateName = NULL;

	// Open dialog & set if user press ok
	if (::GetOpenFileName(&ofn) != 0)
	{
		SetDlgItemText(IDC_WINAMP, ofn.lpstrFile);
	}
}

void CAcrylicDlg::OnBDvd()
{
	// TODO: Add your control notification handler code here
	char szExtBuffer[256];
	CString filter = "Executable (*.exe)|*.exe|All files|*.*|";
	filter.Replace('|', 0);
	OPENFILENAME ofn;

	strcpy(szExtBuffer, "");

	ofn.lStructSize = sizeof(ofn);
	ofn.hwndOwner = NULL;
	ofn.hInstance = NULL;
	ofn.lpstrFilter = filter;
	ofn.lpstrCustomFilter = NULL;
	ofn.nMaxCustFilter = NULL;
	ofn.nFilterIndex = 0;
	ofn.lpstrFile = szExtBuffer;
	ofn.nMaxFile = MAX_PATH;
	ofn.lpstrFileTitle = NULL;
	ofn.nMaxFileTitle = NULL;
	ofn.lpstrInitialDir = NULL;
	ofn.lpstrTitle = "Open file";
	ofn.Flags = NULL;
	ofn.nFileOffset = NULL;
	ofn.nFileExtension = NULL;
	ofn.lpstrDefExt = NULL;
	ofn.lCustData = NULL;
	ofn.lpfnHook = NULL;
	ofn.lpTemplateName = NULL;

	// Open dialog & set if user press ok
	if (::GetOpenFileName(&ofn) != 0)
	{
		SetDlgItemText(IDC_DVD, ofn.lpstrFile);
	}
}

void CAcrylicDlg::OnBPowerpt()
{
	// TODO: Add your control notification handler code here
	char szExtBuffer[256];
	CString filter = "Executable (*.exe)|*.exe|All files|*.*|";
	filter.Replace('|', 0);
	OPENFILENAME ofn;

	strcpy(szExtBuffer, "");

	ofn.lStructSize = sizeof(ofn);
	ofn.hwndOwner = NULL;
	ofn.hInstance = NULL;
	ofn.lpstrFilter = filter;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ofn.lpstrCustomFilter = NULL;
	ofn.nMaxCustFilter = NULL;
	ofn.nFilterIndex = 0;
	ofn.lpstrFile = szExtBuffer;
	ofn.nMaxFile = MAX_PATH;
	ofn.lpstrFileTitle = NULL;
	ofn.nMaxFileTitle = NULL;
	ofn.lpstrInitialDir = NULL;
	ofn.lpstrTitle = "Open file";
	ofn.Flags = NULL;
	ofn.nFileOffset = NULL;
	ofn.nFileExtension = NULL;
	ofn.lpstrDefExt = NULL;
	ofn.lCustData = NULL;
	ofn.lpfnHook = NULL;
	ofn.lpTemplateName = NULL;
}

void CAcrylicDlg::OnBPinball()
{
	// TODO: Add your control notification handler code here
	char szExtBuffer[256];
	CString filter = "Executable (*.exe)|*.exe|All files|*.*|";
	filter.Replace('|', 0);
	OPENFILENAME ofn;

	strcpy(szExtBuffer, "");

	ofn.lStructSize = sizeof(ofn);
	ofn.hwndOwner = NULL;
	ofn.hInstance = NULL;
	ofn.lpstrFilter = filter;
	ofn.lpstrCustomFilter = NULL;
	ofn.nMaxCustFilter = NULL;
	ofn.nFilterIndex = 0;
	ofn.lpstrFile = szExtBuffer;
	ofn.nMaxFile = MAX_PATH;
	ofn.lpstrFileTitle = NULL;
	ofn.nMaxFileTitle = NULL;
	ofn.lpstrInitialDir = NULL;
	ofn.lpstrTitle = "Open file";
	ofn.Flags = NULL;
	ofn.nFileOffset = NULL;
	ofn.nFileExtension = NULL;
	ofn.lpstrDefExt = NULL;
	ofn.lCustData = NULL;
	ofn.lpfnHook = NULL;
	ofn.lpTemplateName = NULL;

	// Open dialog & set if user press ok
	if (::GetOpenFileName(&ofn) != 0)
	{
		SetDlgItemText(IDC_PINBALL, ofn.lpstrFile);
	}
}

void CAcrylicDlg::OnBWinmedia()
{

	// TODO: Add your control notification handler code here
	char szExtBuffer[256];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CString filter = "Executable (*.exe)|*.exe|All files|*.*|";
        filter.Replace('|', 0);
        OPENFILENAME ofn;

        strcpy(szExtBuffer, "");

        ofn.lStructSize = sizeof(ofn);
        ofn.hwndOwner = NULL;
        ofn.hInstance = NULL;
        ofn.lpstrFilter = filter;
        ofn.lpstrCustomFilter = NULL;
        ofn.nMaxCustFilter = NULL;
        ofn.nFilterIndex = 0;
        ofn.lpstrFile = szExtBuffer;
        ofn.nMaxFile = MAX_PATH;
        ofn.lpstrFileTitle = NULL;
        ofn.nMaxFileTitle = NULL;
        ofn.lpstrInitialDir = NULL;
        ofn.lpstrTitle = "Open file";
        ofn.Flags = NULL;
        ofn.nFileOffset = NULL;
        ofn.nFileExtension = NULL;
        ofn.lpstrDefExt = NULL;
        ofn.lCustData = NULL;
        ofn.lpfnHook = NULL;
        ofn.lpTemplateName = NULL;

        // Open dialog & set if user press ok
        if (::GetOpenFileName(&ofn) != 0)
        {
            SetDlgItemText(IDC_WINMEDIA, ofn.lpstrFile);
        }
}

void CAcrylicDlg::OnBSolitriae()
{
        // TODO: Add your control notification handler code here
        char szExtBuffer[256];
        CString filter = "Executable (*.exe)|*.exe|All files|*.*|";
        filter.Replace('|', 0);
        OPENFILENAME ofn;

        strcpy(szExtBuffer, "");

        ofn.lStructSize = sizeof(ofn);
        ofn.hwndOwner = NULL;
        ofn.hInstance = NULL;
        ofn.lpstrFilter = filter;
        ofn.lpstrCustomFilter = NULL;
        ofn.nMaxCustFilter = NULL;
        ofn.nFilterIndex = 0;
        ofn.lpstrFile = szExtBuffer;
        ofn.nMaxFile = MAX_PATH;
        ofn.lpstrFileTitle = NULL;
        ofn.nMaxFileTitle = NULL;
        ofn.lpstrInitialDir = NULL;
        ofn.lpstrTitle = "Open file";
        ofn.Flags = NULL;
        ofn.nFileOffset = NULL;
        ofn.nFileExtension = NULL;
        ofn.lpstrDefExt = NULL;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	ofn.lCustData = NULL;
	ofn.lpfnHook = NULL;
	ofn.lpTemplateName = NULL;
	// Open dialog & set if user press ok
	if (::GetOpenFileName(&ofn) != 0)
	{
		SetDlgItemText(IDC_SOLITRIAE, ofn.lpstrFile);
	}
}
void CAcrylicDlg::OnHelp()
{
	// TODO: Add your control notification handler code here
	::WinExec(  "C:\\Program Files\\Windows
NT\\Accessories\\wordpad.exe helpall.rtf",SW_SHOW);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข. DATA SHEET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EVERLIGHT ELECTRONICS CO.,LTD.

Technical Data Sheet

Infrared Remote-control Receiver Module

IRM-8601S

Features

- High protection ability against EMI .
- Oval lens to improve the receive angles.
- Line-up for various center carrier frequencies.
- Low voltage and low power consumption.
- High immunity against ambient light.
- Photodiode with integrated circuit.
- TTL and CMOS compatibility.
- Long reception distance.
- High sensitivity.

Descriptions

The device is a miniature type infrared remote control system receiver which has been developed and designed by utilizing the most updated IC technology. The PIN diode and preamplifier are assembled on lead frame, the epoxy package is designed as an IR filter. The demodulated output signal can directly be decoded by a microprocessor.

Applications

1. Optical switch
2. Light detecting portion of remote control
   - AV instruments such as Audio, TV, VCR, CD, MD, etc.
   - Home appliances such as Air-conditioner, Fan , etc.
   - The other equipments with wireless remote control.
   - CATV set top boxes
   - Multi-media Equipment

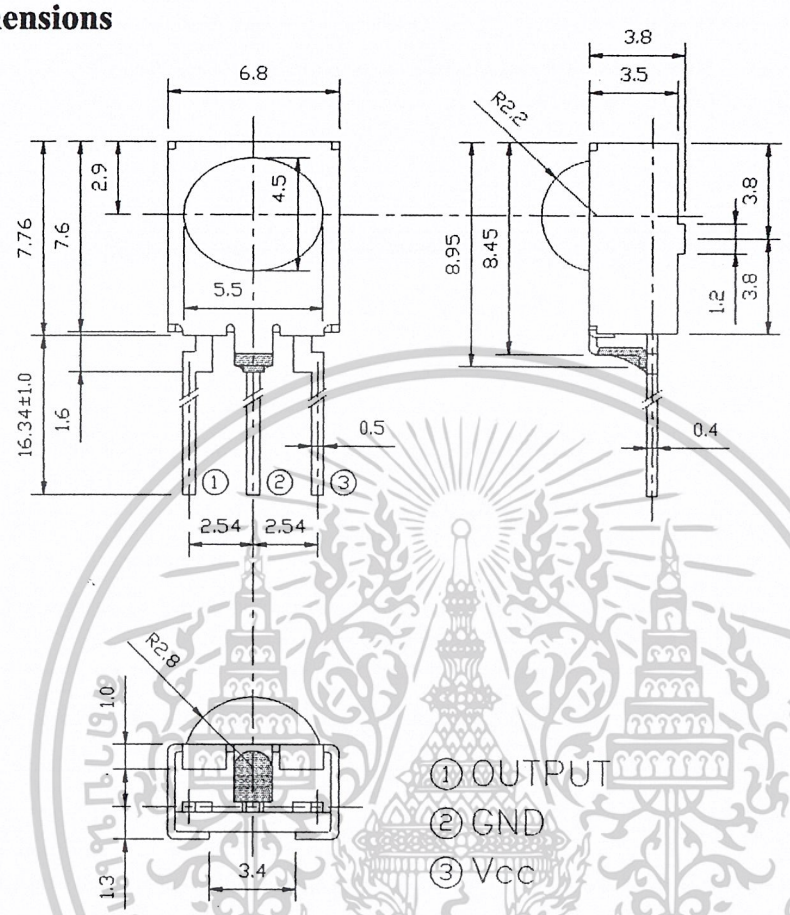
Device Selection Guide

PART	MATERIAL	COLOR
Chip	Silicon	-----
Metal can	Tinplate	Silver-white
Package	Epoxy	Black

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปแสวงหาผลประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IRM-8601S

Package Dimensions



Notes: 1.All dimensions are in millimeters.
2.Tolerances unless dimensions ±0.3mm.

Absolute Maximum Ratings (Ta=25℃)

Parameter	Symbol	Rating	Unit	Notice
Supply Voltage	Vcc	0~6	V	
Operating Temperature	Topr	-25 ~ +85	℃	
Storage Temperature	Tstg	-40 ~ +85	℃	
Soldering Temperature	Tsol	260	℃	4mm from mold body less than 10 seconds

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recommended Operating Condition

Supply Voltage Rating: Vcc 4.5V to 5.5V

Electro-Optical Characteristics (Ta=25℃, and Vcc=5 V)

Parameter	Symbol	MIN.	TYP.	MAX.	Unit	Condition
Consumption Current	Icc	---	---	3	mA	No signal input
B.P.F Center Frequency	Fo	---	38	---	KHz	
Peak Wavelength	λ_p	---	940	---	nm	
Reception Distance	L_0	8	---	---	m	At the ray axis *1
	L_{45}	4	---	---		
Half Angle(Horizontal)	Θ_h	---	45	---	deg	
Half Angle(Vertical)	Θ_v	---	45	---	deg	
High Level Pulse Width	T_H	400	---	800	μs	At the ray axis *2
Low Level Pulse Width	T_L	400	---	800	μs	
High Level Output Voltage	V_H	4.5	---	---	V	
Low Level Output Voltage	V_L	---	0.2	0.5	V	

Notes:

*1:The ray receiving surface at a vertex and relation to the ray axis in the range of θ= 0° and θ=45°.

*2:A range from 30cm to the arrival distance. Average value of 50 pulses.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Test Method :

The specified electro-optical characteristics is satisfied under the following Conditions at the controllable distance.

①Measurement place

A place that is nothing of extreme light reflected in the room.

②External light

Project the light of ordinary white fluorescent lamps which are not high Frequency lamps and must be less then 10 Lux at the module surface. (Ee≦10Lux)

③Standard transmitter

A transmitter whose output is so adjusted as to **Vo=400mVp-p** and the output Wave form shown in Fig.-1.According to the measurement method shown in Fig.-2 the standard transmitter is specified.

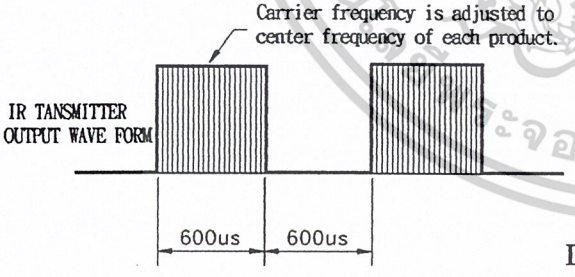
However , the infrared photodiode to be used for the transmitter should be λp=940nm,Δλ=50nm. Also, photodiode is used of PD438B(Vr=5V).

(Standard light / Light source temperature 2856°K).

④Measuring system

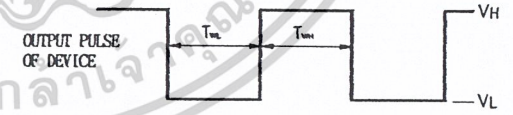
According to the measuring system shown in Fig.-3

Fig.-1 Transmitter Wave Form



Duty=0.5

D.U.T output Pulse



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fig.-2 Measuring Method

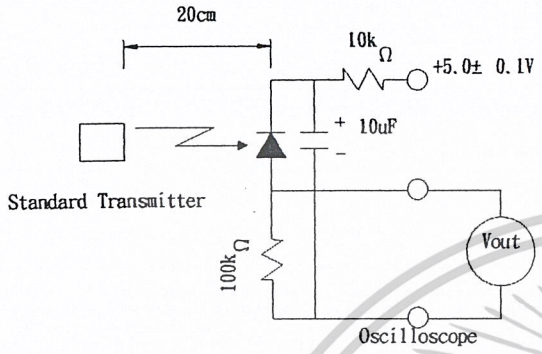
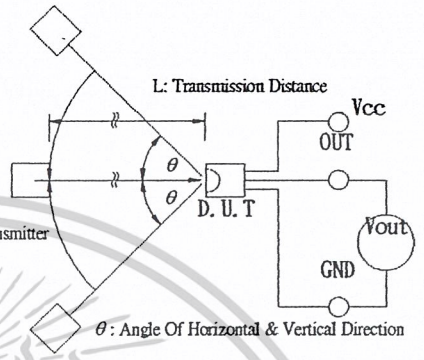
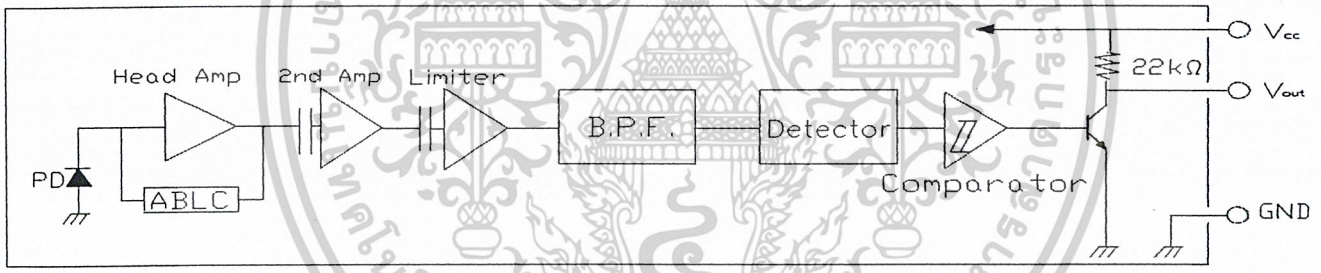


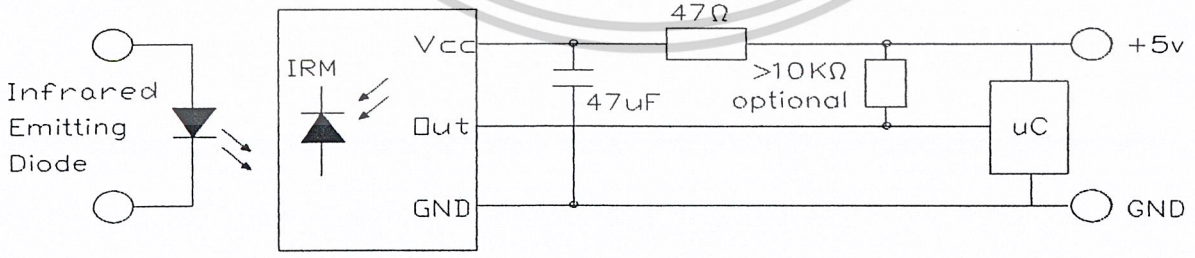
Fig.-3 Measuring System



Block Diagram :



Application Circuit :



RC Filter should be connected closely between Vcc pin and GND pin.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Electro-Optical Characteristics Curves

Fig.-4 Relative Spectral Sensitivity vs. Wavelength

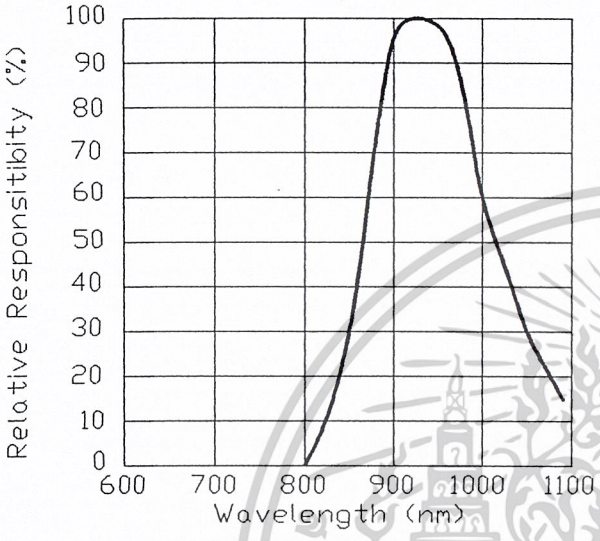


Fig.-5 Relative Transmission Distance vs. Direction

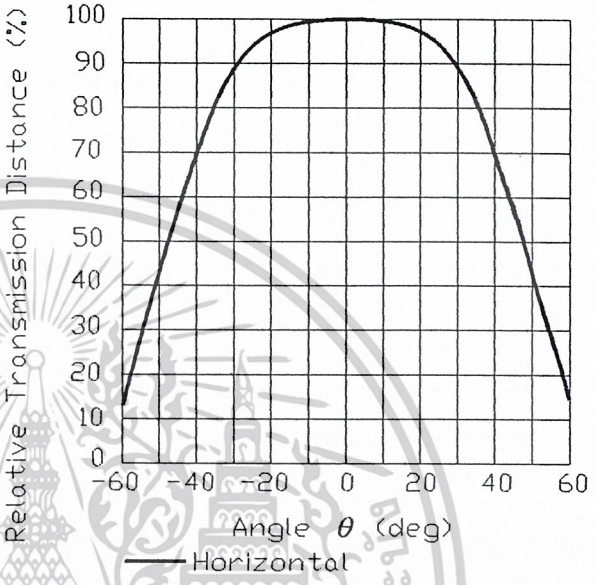


Fig.-6 Output Pulse Length vs. Arrival Distance

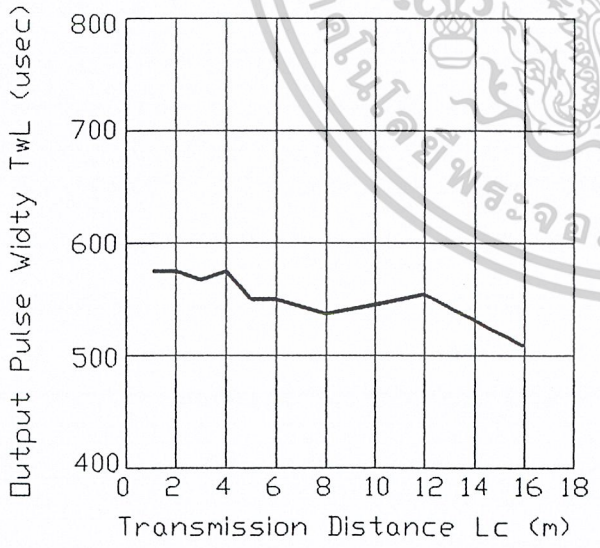
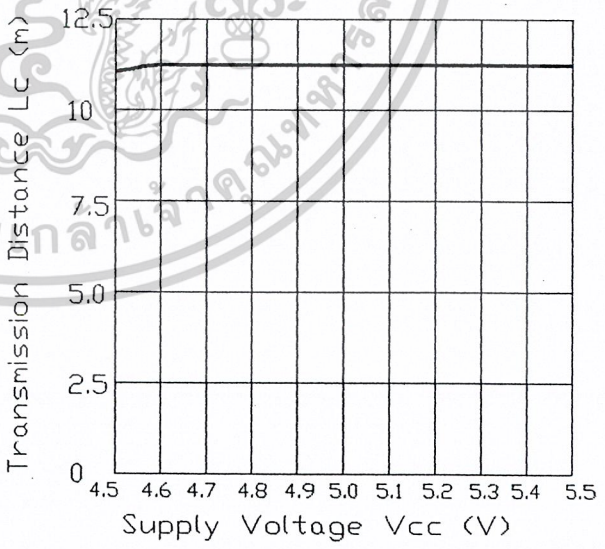


Fig.-7 Arrival Distance vs. Supply Voltage



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Electro-Optical Characteristics Curves

Fig.-8 Relative Transmission Distance vs. Center Carrier Frequency

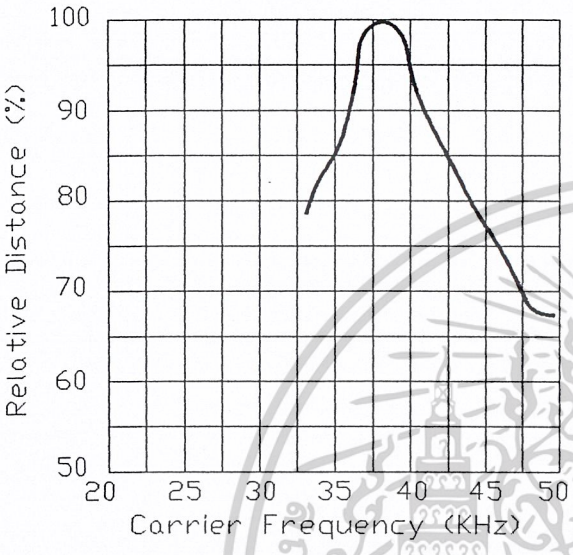
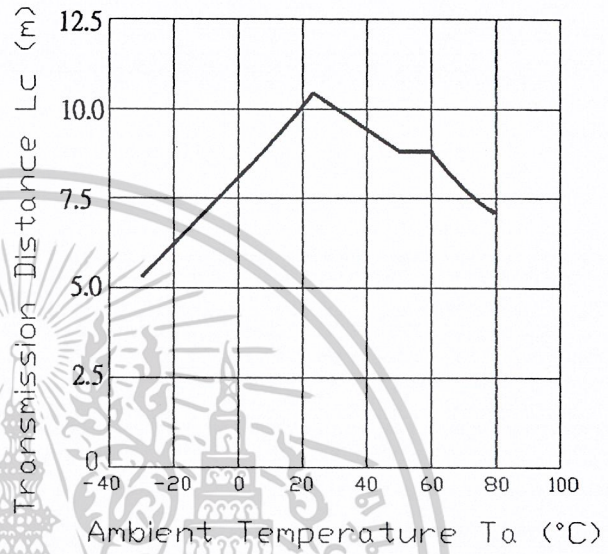


Fig.-9 Arrival Distance vs. Ambient Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reliability Test Item And Condition

The reliability of products shall be satisfied with items listed below.

Confidence level：90%

LTPD：10%

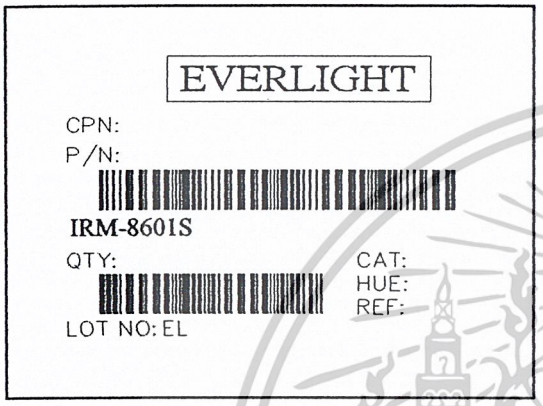
Test Items	Test Conditions	Failure Judgement Criteria	Samples(n) Defective(c)
Temperature cycle	1 cycle -25℃ ⟷ +85℃ (30min)(5min)(30min) 300 cycle test	$L_0 \leqq L \times 0.8$ $L_{45} \leqq L \times 0.8$ L: Lower specification limit	n=22,c=0
High temperature test	Temp: +85℃ Vcc:5V 1000hrs		n=22,c=0
Low temperature storage	Temp: -40℃ 1000hrs		n=22,c=0
High temperature High humidity	Ta: 85℃,RH:85% 1000hrs		n=22,c=0
Solder heat	Temp: 260±5℃ 10sec 4mm From the bottom of the package.		n=22,c=0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packing Quantity Specification

1. 1000 PCS/1Box
2. 10 Boxes/1Carton

Label Form Specification



CPN: Customer's Production Number
P/N : Production Number
QTY: Packing Quantity
CAT: Ranks
HUE: Peak Wavelength
REF: Reference
LOT No: Lot Number

Notes

1. Above specification may be changed without notice. EVERLIGHT will reserve authority on material change for above specification.
2. When using this product, please observe the absolute maximum ratings and the instructions for using outlined in these specification sheets. EVERLIGHT assumes no responsibility for any damage resulting from use of the product which does not comply with the absolute maximum ratings and the instructions included in these specification sheets.
3. These specification sheets include materials protected under copyright of EVERLIGHT corporation. Please don't reproduce or cause anyone to reproduce them without EVERLIGHT's consent.

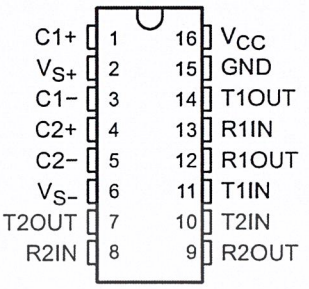
EVERLIGHT ELECTRONICS CO., LTD.
Office: No 25, Lane 76, Sec 3, Chung Yang Rd, Tucheng, Taipei 236, Taiwan, R.O.C
Tel: 886-2-2267-2000, 2267-9936
Fax: 886-2267-6244, 2267-6189, 2267-6306
http:\\www.everlight.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLLS047L – FEBRUARY 1989 – REVISED MARCH 2004

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0-μF Charge-Pump Capacitors
- Operates Up To 120 kbit/s
- Two Drivers and Two Receivers
- ±30-V Input Levels
- Low Supply Current . . . 8 mA Typical
- ESD Protection Exceeds JESD 22
  - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1-μF Charge-Pump Capacitors is Available With the MAX202
- Applications
  - TIA/EIA-232-F, Battery-Powered Systems, Terminals, Modems, and Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
MAX232I . . . D, DW, OR N PACKAGE
(TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ±30-V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube of 25	MAX232N	MAX232N
	SOIC (D)	Tube of 40	MAX232D	MAX232
		Reel of 2500	MAX232DR	
	SOIC (DW)	Tube of 40	MAX232DW	MAX232
		Reel of 2000	MAX232DWR	
	SOP (NS)	Reel of 2000	MAX232NSR	MAX232
−40°C to 85°C	PDIP (N)	Tube of 25	MAX232IN	MAX232IN
	SOIC (D)	Tube of 40	MAX232ID	MAX232I
		Reel of 2500	MAX232IDR	
	SOIC (DW)	Tube of 40	MAX232IDW	MAX232I
		Reel of 2000	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2004, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

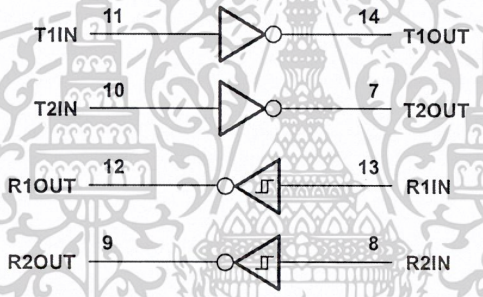
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1) ..... −0.3 V to 6 V
Positive output supply voltage range, V_{S+} ..... V_{CC} − 0.3 V to 15 V
Negative output supply voltage range, V_{S-} ..... −0.3 V to −15 V
Input voltage range, V_I: Driver ..... −0.3 V to V_{CC} + 0.3 V
Receiver ..... ±30 V
Output voltage range, V_O: T1OUT, T2OUT ..... V_{S-} − 0.3 V to V_{S+} + 0.3 V
R1OUT, R2OUT ..... −0.3 V to V_{CC} + 0.3 V
Short-circuit duration: T1OUT, T2OUT ..... Unlimited
Package thermal impedance, θ_{JA} (see Notes 2 and 3): D package ..... 73°C/W
DW package ..... 57°C/W
N package ..... 67°C/W
NS package ..... 64°C/W
Operating virtual junction temperature, T_J ..... 150°C
Storage temperature range, T_{stg} ..... −65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTES: 1. All voltages are with respect to network GND.
2. Maximum power dissipation is a function of $T_J(max)$, θ_{JA}, and T_A. The maximum allowable power dissipation at any allowable ambient temperature is $P_D = (T_J(max) - T_A)/\theta_{JA}$. Operating at the absolute maximum T_J of 150°C can affect reliability.
3. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

			MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage		4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN,T2IN)		2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)				0.8	V
R1IN, R2IN	Receiver input voltage				±30	V
T_A	Operating free-air temperature	MAX232	0		70	°C
		MAX232I	−40		85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 4 and Figure 4)

PARAMETER		TEST CONDITIONS	MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	V_{CC} = 5.5 V, All outputs open, T_A = 25°C		8	10	mA

‡ All typical values are at V_{CC} = 5 V and T_A = 25°C.
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	T1OUT, T2OUT	R_L = 3 kΩ to GND		5	7		V
V_{OL}	Low-level output voltage‡	T1OUT, T2OUT	R_L = 3 kΩ to GND			−7	−5	V
r_o	Output resistance	T1OUT, T2OUT	V_{S+} = V_{S-} = 0,	V_O = ±2 V	300			Ω
I_{OS}§	Short-circuit output current	T1OUT, T2OUT	V_{CC} = 5.5 V,	V_O = 0		±10		mA
I_{IS}	Short-circuit input current	T1IN, T2IN	V_I = 0				200	µA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
§ Not more than one output should be shorted at a time.
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R_L = 3 kΩ to 7 kΩ, See Figure 2			30	V/µs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/µs
	Data rate	One TOUT switching		120		kbit/s

NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 4)

PARAMETER			TEST CONDITIONS		MIN	TYP†	MAX	UNIT
V_{OH}	High-level output voltage	R1OUT, R2OUT	I_{OH} = −1 mA		3.5			V
V_{OL}	Low-level output voltage‡	R1OUT, R2OUT	I_{OL} = 3.2 mA				0.4	V
V_{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V,	T_A = 25°C		1.7	2.4	V
V_{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN	V_{CC} = 5 V,	T_A = 25°C	0.8	1.2		V
V_{hys}	Input hysteresis voltage	R1IN, R2IN	V_{CC} = 5 V		0.2	0.5	1	V
r_i	Receiver input resistance	R1IN, R2IN	V_{CC} = 5,	T_A = 25°C	3	5	7	kΩ

† All typical values are at V_{CC} = 5 V, T_A = 25°C.
‡ The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

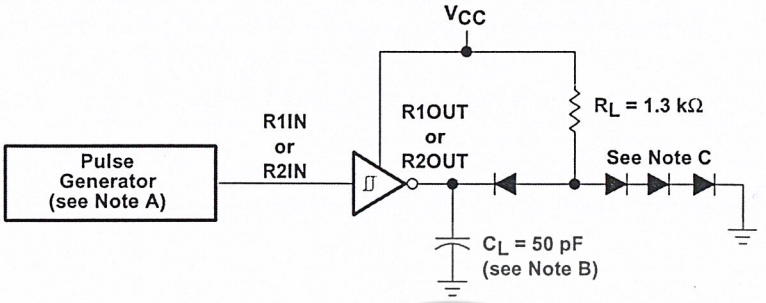
switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 4 and Figure 1)

PARAMETER		TYP	UNIT
$t_{PLH(R)}$	Receiver propagation delay time, low- to high-level output	500	ns
$t_{PHL(R)}$	Receiver propagation delay time, high- to low-level output	500	ns

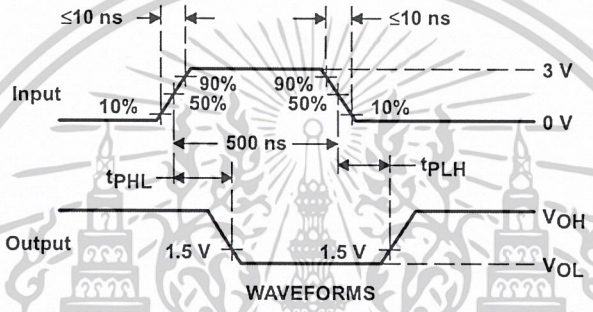
NOTE 4: Test conditions are C1–C4 = 1 µF at V_{CC} = 5 V ± 0.5 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



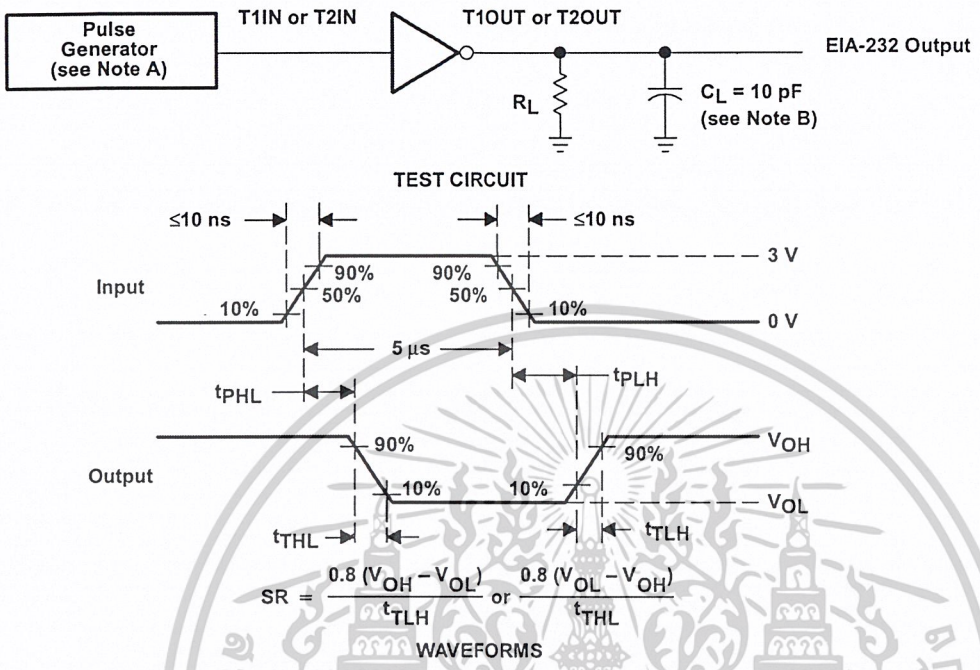
WAVEFORMS

NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.
C. All diodes are 1N3064 or equivalent.

Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

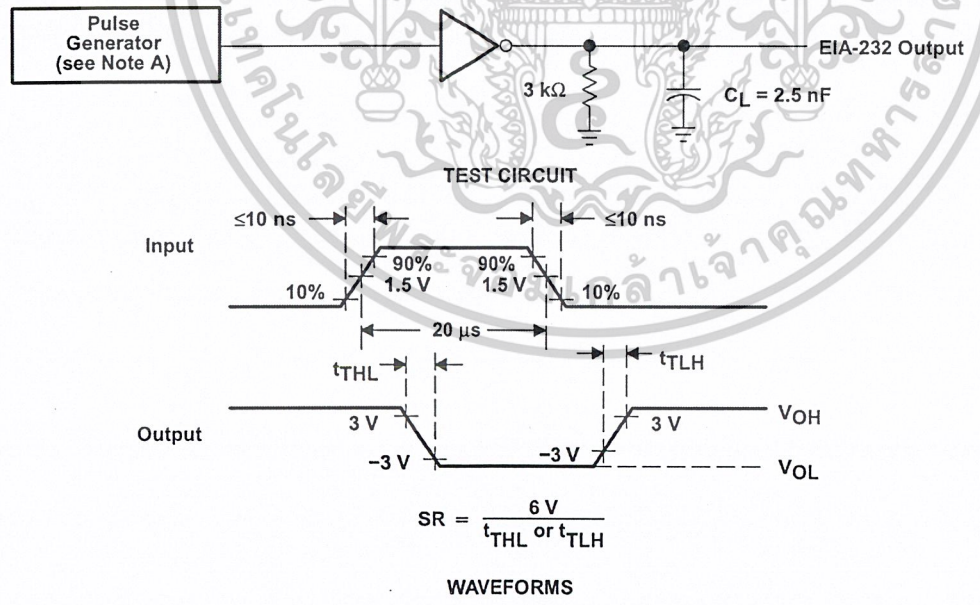
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARAMETER MEASUREMENT INFORMATION



NOTES: A. The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.
B. C_L includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5-µs Input)

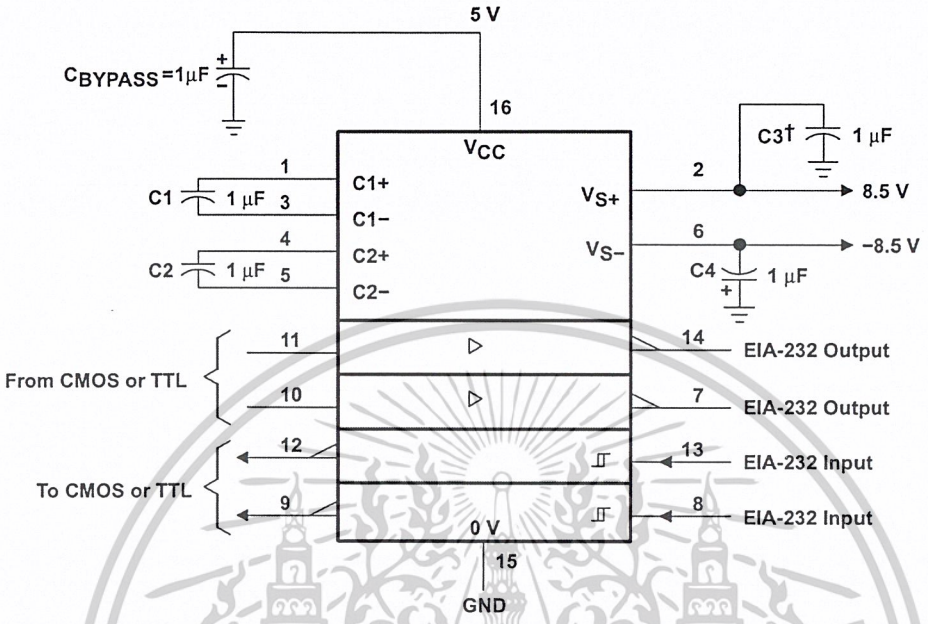


NOTE A: The pulse generator has the following characteristics: Z_O = 50 Ω, duty cycle ≤ 50%.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20-µs Input)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

NOTES: A. Resistor values shown are nominal.

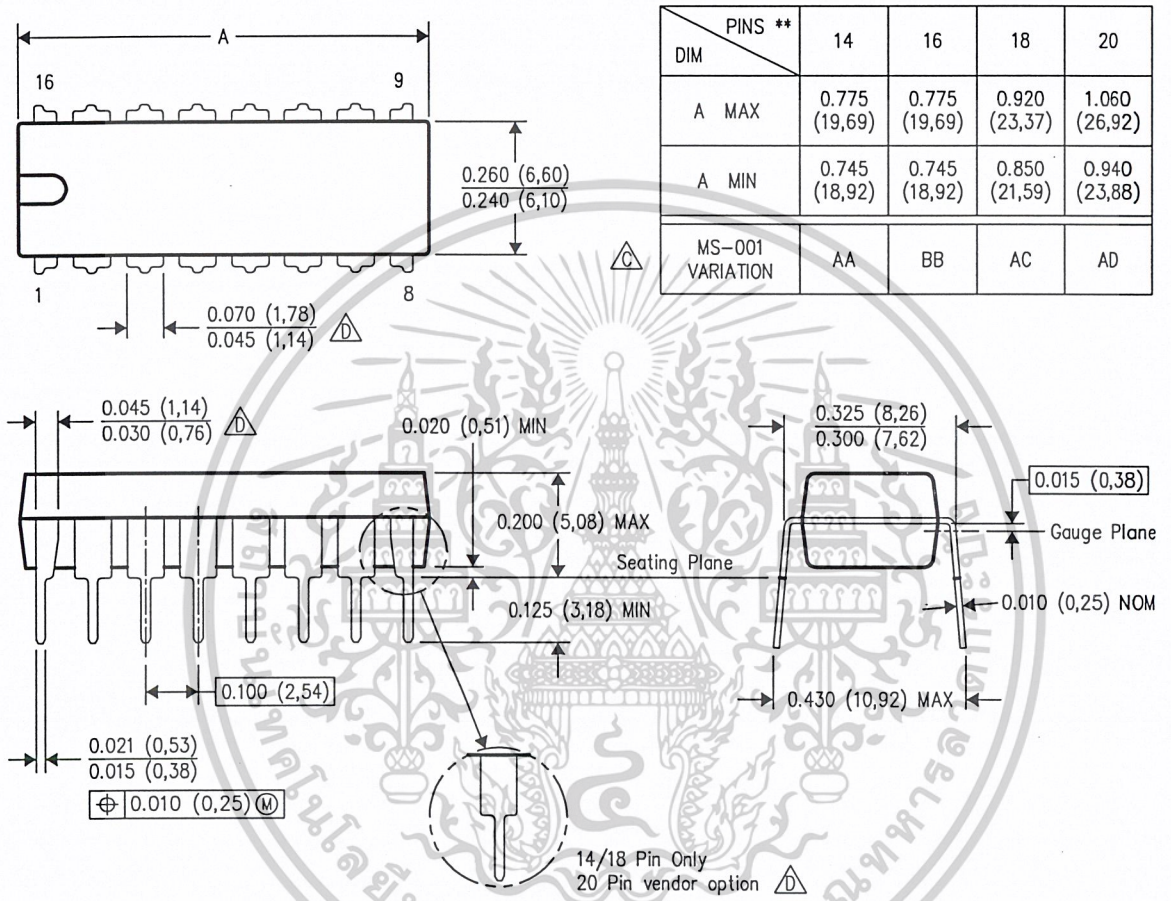
B. Nonpolarized ceramic capacitors are acceptable. If polarized tantalum or electrolytic capacitors are used, they should be connected as shown. In addition to the 1-μF capacitors shown, the MAX202 can operate with 0.1-μF capacitors.

Figure 4. Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N (R–PDIP–T**) PLASTIC DUAL–IN–LINE PACKAGE

16 PINS SHOWN



DIM \ PINS **	14	16	18	20
A MAX	0.775 (19,69)	0.775 (19,69)	0.920 (23,37)	1.060 (26,92)
A MIN	0.745 (18,92)	0.745 (18,92)	0.850 (21,59)	0.940 (23,88)
MS–001 VARIATION	AA	BB	AC	AD

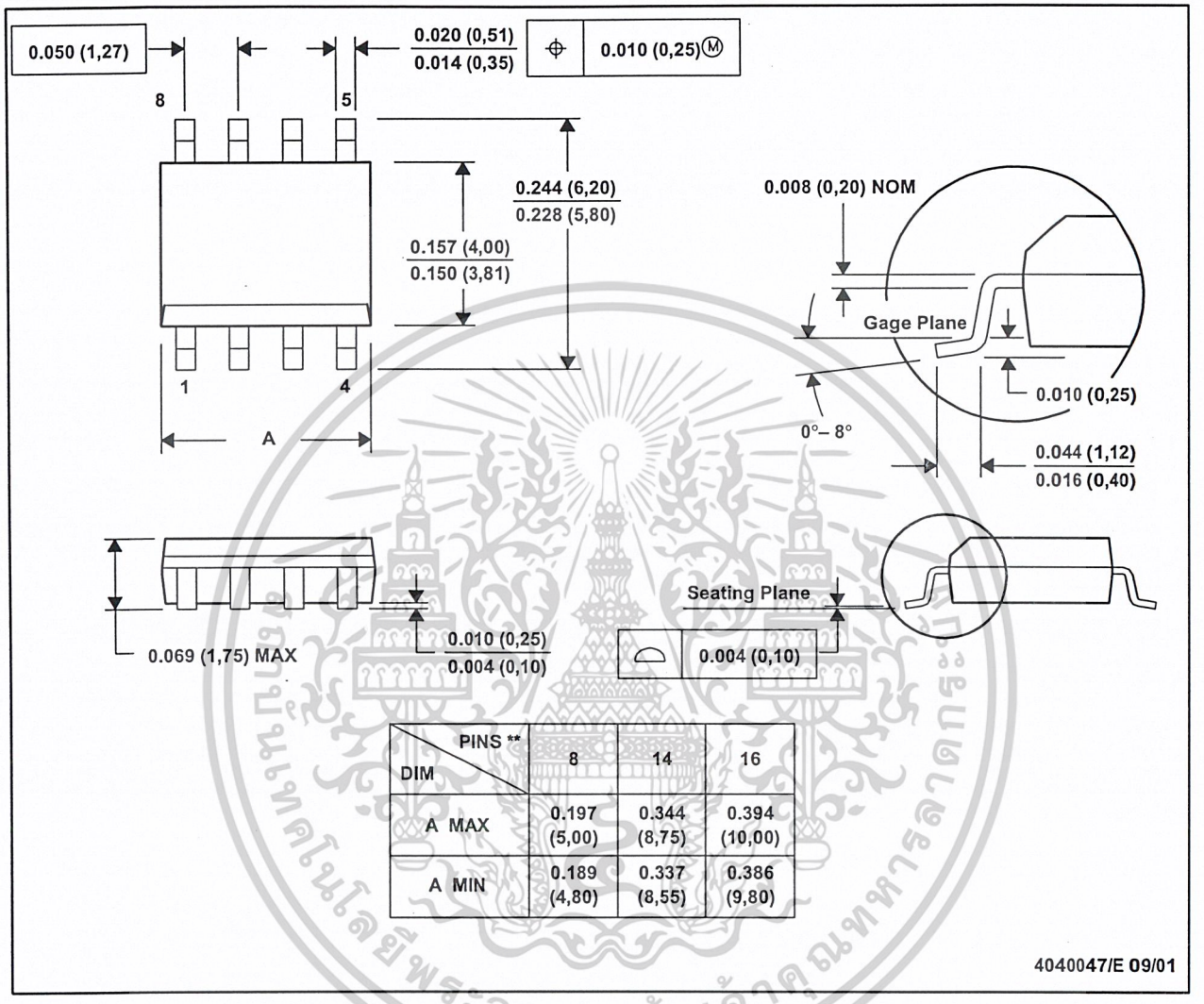
NOTES:
A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Falls within JEDEC MS–001, except 18 and 20 pin minimum body length (Dim A).
D. The 20 pin end lead shoulder width is a vendor option, either half or full width.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D (R-PDSO-G**)

PLASTIC SMALL-OUTLINE PACKAGE

8 PINS SHOWN



DIM \ PINS **	8	14	16
A MAX	0.197 (5,00)	0.344 (8,75)	0.394 (10,00)
A MIN	0.189 (4,80)	0.337 (8,55)	0.386 (9,80)

4040047/E 09/01

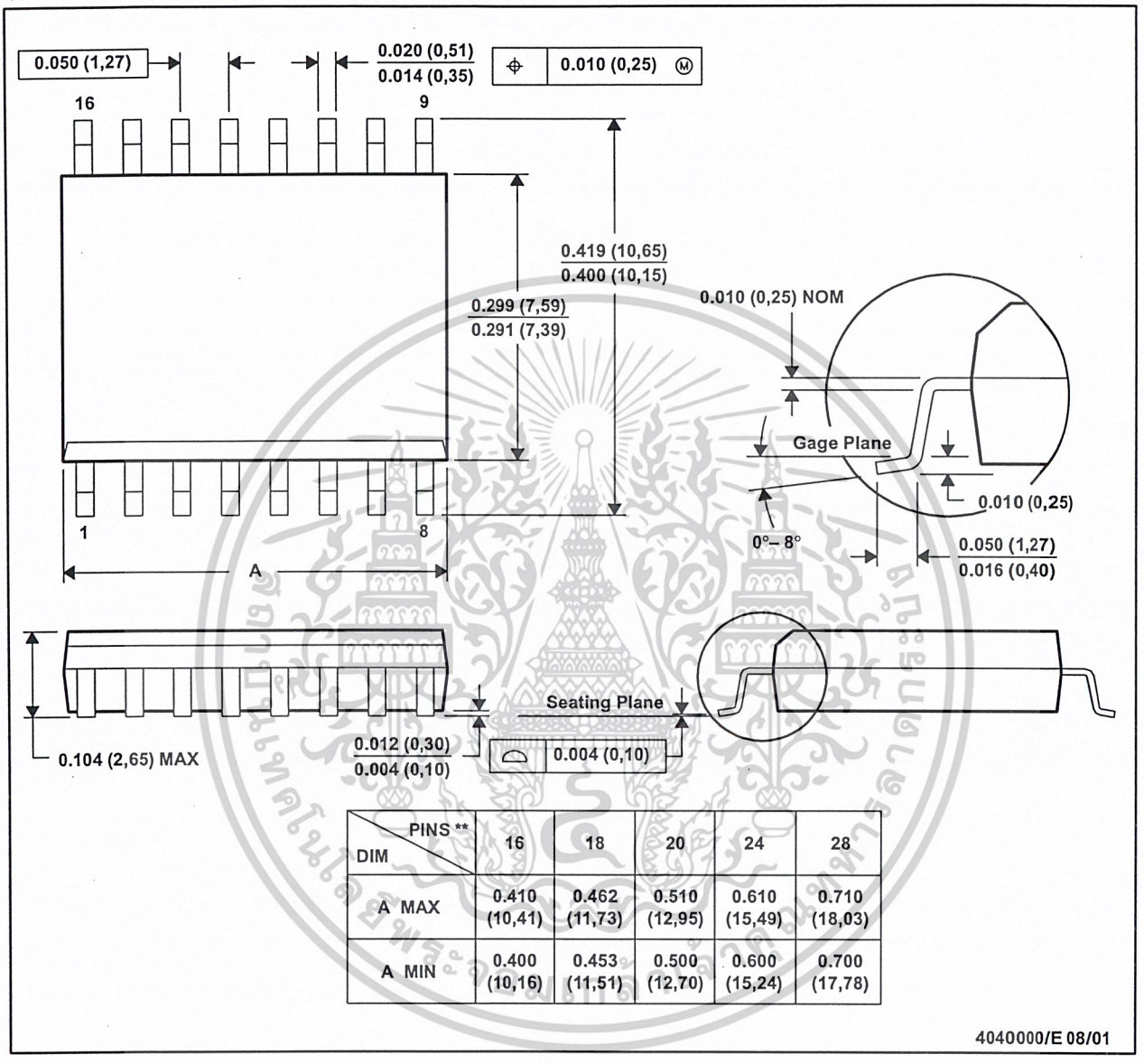
NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-012

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MSOI003E – JANUARY 1995 – REVISED SEPTEMBER 2001

DW (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

16 PINS SHOWN



DIM \ PINS **	16	18	20	24	28
A MAX	0.410 (10,41)	0.462 (11,73)	0.510 (12,95)	0.610 (15,49)	0.710 (18,03)
A MIN	0.400 (10,16)	0.453 (11,51)	0.500 (12,70)	0.600 (15,24)	0.700 (17,78)

4040000/E 08/01

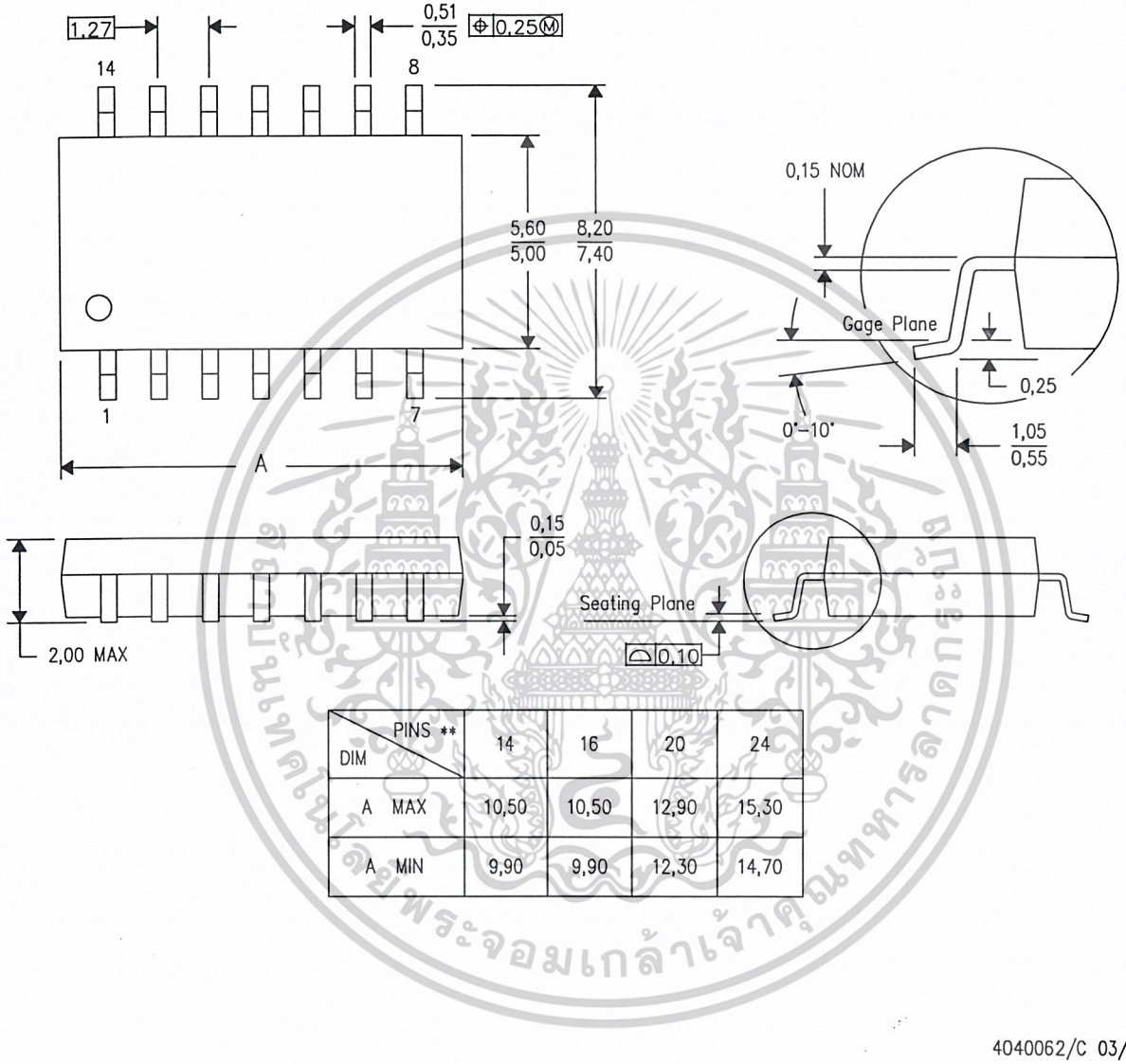
NOTES: A. All linear dimensions are in inches (millimeters).
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion not to exceed 0.006 (0,15).
D. Falls within JEDEC MS-013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MECHANICAL DATA

NS (R-PDSO-G)** **PLASTIC SMALL-OUTLINE PACKAGE**

14-PINS SHOWN



DIM \ PINS **	14	16	20	24
A MAX	10,50	10,50	12,90	15,30
A MIN	9,90	9,90	12,30	14,70

4040062/C 03/03

NOTES:
A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Body dimensions do not include mold flash or protrusion, not to exceed 0,15.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

TI assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using TI components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address: Texas Instruments
Post Office Box 655303 Dallas, Texas 75265

Copyright © 2004, Texas Instruments Incorporated

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้จะประสบสำเร็จลุล่วงไปได้ด้วยดีมิได้เลย ถ้าไม่ได้รับการสนับสนุนและความช่วยเหลือจาก ดร.พิพัฒน์ พรหมมี โดยอาจารย์ได้ให้คำแนะนำช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นระหว่างที่ทำโครงงานชิ้นนี้ อีกทั้งเพื่อน ๆ ทุกคนที่คอยช่วยเหลือในการแนะนำหรือให้ยืมอุปกรณ์ในการทดลองและเป็นกำลังใจให้ตลอดมา ตลอดจนอาจารย์ในภาควิชาโทรคมนาคมทุกท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้กับกลุ่มข้าพเจ้ามาโดยตลอด และที่ลืมไม่ได้ก็คือ คุณพ่อและคุณแม่ที่ให้กำเนิดพวกเรามา

จึงขอขอบคุณอย่างสูง ณ โอกาสนี้

นายชุมพล ผู้เจริญวิบูลย์
นายณัชพงษ์ เจียมหาทรัพย์
นายศุภกิจ ภูสีมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] Digital – Integrated – Circuit , Operational – Amplifier , and Optoelectronic Circuit Design Edited by Bryan Norris , Manager , Applications Laboratory , Texas Instruments Limited

[2] เจน สงสมพันธุ์ , " โทรทัศน์สีภาคสนาม",โรงเรียนอิเล็กทรอนิกส์กรุงเทพมหานคร , 2534

[3] กฤษดา ใจเย็น , อรรถพล บุญยะโภคา , ชัยวัฒน์ ลิ้มพรจิตรวิไล , " เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม ", บริษัท อินโนเวทีฟเอ็กเพอริเมนต์ จำกัด

[4] "รีโมตคอนโทรล เครื่องควบคุมไร้สาย", บริษัทซีเอ็ดยูเคชั่น จำกัด พ.ศ. 2538

[5] ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรมและใช้งาน Visual c++ 6.0 ฉบับโปรแกรมเมอร์", สำนักพิมพ์ อินโฟเพรส

[6] ประจิน พลังสันติกุล, " เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC ", บริษัท อินโนเวทีฟเอ็กเพอริเมนต์ จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้