สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รีโมทควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแสดงผลผ่านจอแอลซีดี

REMOTE CONTROL ELECTRICAL EQUIPMENT IN HOME DISPLAY BY LCD

โดย

นายศุภชัย อักษรสิทธิ์

นายอนันตพร บัวเงิน

เลขหมู่.....
เลขทะเบียน 62677
วัน,เดือน,ปี 21 ส.ค. 2549
.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีโมทควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแสดงผลผ่านจอแอลซีดี

REMOTE CONTROL ELECTRICAL EQUIPMENT IN HOME DISPLAY BY LCD

โดย

นายศุภชัย อักษรสิทธิ์ 46015073

นายอนันตพร บัวเงิน 46015078

อาจารย์ที่ปรึกษา

รศ.สมยศ จุณณะปิยะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **รีโมทควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแสดงผลผ่านจอแอลซีดี**

REMOTE CONTROL ELECTRICAL EQUIPMENT IN HOME DISPLAY BY LCD

ผู้จัดทำ

1. นายศุภชัย อักษรสิทธิ์ 46015073
2. นายอนันตพร บัวเงิน 46015078

..... อาจารย์ที่ปรึกษา

(รศ.สมยศ จุณณะปิยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีโมทควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแสดงผลผ่านจอแอลซีดี

REMOTE CONTROL ELECTRICAL EQUIPMENT IN HOME DISPLAY BY LCD

โดย นายสุภชัย อักษรสิทธิ์ 46015073

นายอนันตพร บัวเงิน 46015078

อาจารย์ที่ปรึกษา รศ.สมยศ จุณณะปิยะ

บทคัดย่อ

โครงงานนี้ทำขึ้นเพื่อจุดประสงค์คือเพื่ออำนวยความสะดวกในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน โดยใช้รีโมทควบคุมเพียง 1 ตัว ก็สามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้ทั้งหมด 8 ช่อง โดยการส่งสัญญาณผ่านตัวกลางซึ่งเป็นตัวรับสัญญาณ และส่งสัญญาณควบคุมเครื่องใช้ไฟฟ้าแต่ละเครื่องและสามารถตรวจสอบสถานะอุปกรณ์เครื่องใช้ไฟฟ้าผ่านหน้าจอแอลซีดี

ABSTRACT

This project male for comfortable controller electrical equipment in home. By only a remote , can control all electrical equipment in home by sending signal to receiver for control electrical equipment that want to use and can check status of electrical equipment by LCD.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำโครงงานรู้สึกซาบซึ้งต่อความกรุณาที่ รศ. สมยศ จุณณะปิยะ อาจารย์ที่ปรึกษา ที่ได้ให้คำแนะนำ ตลอดจนอาจารย์ในภาควิชาที่ประสิทธิประสาทความรู้ให้กับผู้จัดทำในครั้งนี้ ถ้าปราศจากท่านโครงงานนี้ก็คงไม่สำเร็จลุล่วงไปได้

นอกจากนั้นผู้จัดทำต้องขอขอบคุณ เพื่อนๆ ที่เอื้อเฟื้อเครื่องมืออุปกรณ์ และให้การช่วยเหลือทั้งแรงกาย แรงใจ ขอกราบขอบพระคุณบิดา มารดา ที่ให้กำเนิด และส่งเสียเลี้ยงดูให้เรียนหนังสือจนจบ ซึ่งความดีและความรู้ในปริญญานิพนธ์ฉบับนี้ขอมอบแด่ท่านผู้มีพระคุณทั้งหลาย

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ

1.1 แนวความคิดที่มาของการทำปริญญานิพนธ์ 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของการทำปริญญานิพนธ์ 1

1.3 ขอบเขตของการทำปริญญานิพนธ์ 1

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ MCS-51 3

2.1.1 โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 3

2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51 4

2.1.3 โครงสร้างและการทำงานของพอร์ต 8

2.1.4 การทำงานของไมโครคอนโทรลเลอร์ MCS-51 10

2.1.5 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 13

2.2 ทฤษฎีเบื้องต้นสแกนดีย์ 22

2.3 การใช้งานด็อตเมตริกซ์แอลซีดีโมดูล 24

2.3.1 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์ 24

2.3.2 รายละเอียดของแต่ละคำสั่ง 28

2.3.3 การอ่านและการเขียนข้อมูลกับดีดีแรม/ซีจีแรม 30

2.3.4 การเขียนโปรแกรมควบคุม 30

2.4 ทฤษฎีรีโมทคอนโทรล 31

2.4.1 ความแตกต่างของการเข้ารหัส 31

2.5 ทฤษฎีทรานซิสเตอร์ 32

2.5.1 ช่วงเวลาการสวิตช์ในทางปฏิบัติ 32

2.5.2 การใช้สวิตช์ทรานซิสเตอร์ควบคุมโหลด 33

บทที่ 3 การออกแบบและการสร้าง

3.1 บล็อกไดอะแกรมของโครงงาน 34

3.1.1 บล็อกไดอะแกรมของตัวแม่ 34

3.1.2 บล็อกไดอะแกรมของรีโมทคอนโทรล 35

3.2 การออกแบบฮาร์ดแวร์ 35

3.2.1 ส่วนประมวลผล 35

3.2.2 ภาคจ่ายไฟ 36

3.2.3 ส่วนการแสดงผล 37

3.2.4 ภาคไดรฟ์เวอร์ 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
3.2.5 โมดูลการรับและส่งสัญญาณความถี่วิทยุ	36
3.3 การออกแบบซอฟต์แวร์ควบคุมวงจร	39
3.3.1 หลักการออกแบบซอฟต์แวร์ด้านส่งหรือรีโมทคอนโทรล	39
3.3.2 หลักการออกแบบซอฟต์แวร์ด้านรับหรือตัวแม่	39
บทที่ 4 การทดลอง	
4.1 การตรวจวัดสัญญาณที่ออกมาจากรีโมทคอนโทรล	42
4.2 การแสดงผลหน้าจอแอลซีดี	48
4.3 รูปแสดงชิ้นงานจริง	50
บทที่ 5 สรุปและวิจารณ์	54
ภาคผนวก ก	
ภาคผนวก ข	
กิตติกรรมประกาศ	
เอกสารอ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx 5

รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Sxx 6

รูปที่ 2.3 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช Atmel 7

รูปที่ 2.4 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x 7

รูปที่ 2.5 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 9

รูปที่ 2.6 วงจรพูลอัปภายในพอร์ตไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 11

รูปที่ 2.7 ไซเกิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 11

รูปที่ 2.8 ไดอะแกรมเวลาแสดงการติดต่อและเข้าถึงหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 12

รูปที่ 2.9 การจัดสรรหน่วยความจำโปรแกรมไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 13

รูปที่ 2.10 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51 15

รูปที่ 2.11 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 16

รูปที่ 2.12 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 17

รูปที่ 2.13 แสดงโครงสร้างภายในและการจัดสรรพื้นที่ใช้งานภายในของระบบไมโครคอนโทรลเลอร์ MCS-51 18

รูปที่ 2.14 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด 23

รูปที่ 2.15 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51 23

รูปที่ 2.16 การต่อแอลซีดีโมดูลเข้ากับไมโครคอนโทรลเลอร์ 25

รูปที่ 2.17 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ความกว้างของพัลส์ 31

รูปที่ 2.18 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ช่องว่างของพัลส์ 31

รูปที่ 2.19 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ช่องการเลื่อนของพัลส์ 32

รูปที่ 2.20 แสดงความสัมพันธ์ระหว่างสัญญาณทางอินพุตและเอาต์พุตของวงจรสวิตช์ทรานซิสเตอร์ 32

รูปที่ 2.21 แสดงวงจรสวิตช์ทรานซิสเตอร์ที่ใช้งาน 33

รูปที่ 3.1 แสดงบล็อกไดอะแกรมของตัวแม่ 34

รูปที่ 3.2 แสดงบล็อกไดอะแกรมของรีโมทคอนโทรล 35

รูปที่ 3.3 วงจรภาคจ่ายไฟของชุดรีโมทคอนโทรล 36

รูปที่ 3.4 วงจรภาคจ่ายไฟของตัวแม่ 36

รูปที่ 3.5 การต่อวงจรเพื่อควบคุมจอแอลซีดี 37

รูปที่ 3.6 วงจรไดรฟ์เวอร์ 37

รูปที่ 3.7 โมดูลสำเร็จรูปรับและส่งสัญญาณความถี่วิทยุ 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่ 3.8 โฟลว์ชาร์ตแสดงหลักการทำงานของตัวรีโมทคอนโทรล 40
รูปที่ 3.9 โฟลว์ชาร์ตหลักการทำงานของตัวแม่ 41
รูปที่ 4.1 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 1 42
รูปที่ 4.2 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 1 42
รูปที่ 4.3 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 2 43
รูปที่ 4.4 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 2 43
รูปที่ 4.5 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 3 43
รูปที่ 4.6 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 3 44
รูปที่ 4.7 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 4 44
รูปที่ 4.8 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 4 44
รูปที่ 4.9 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 5 45
รูปที่ 4.10 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 5 45
รูปที่ 4.11 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 6 45
รูปที่ 4.12 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 6 46
รูปที่ 4.13 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 7 46
รูปที่ 4.14 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 7 46
รูปที่ 4.15 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 8 47
รูปที่ 4.16 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 8 47
รูปที่ 4.17 แสดงสัญญาณของการเปิด (ON) ทั้ง 8 ช่อง 47
รูปที่ 4.18 แสดงสัญญาณของการปิด (OFF) ทั้ง 8 ช่อง 48
รูปที่ 4.19 แสดงปุ่มทั้งหมดของรีโมทคอนโทรล 48
รูปที่ 4.20 แสดงหน้าจอแอลซีดี "# Module Ready #" รีโมทคอนโทรลพร้อมรับคำสั่งการใช้งาน 49
รูปที่ 4.21 แสดงตัวอย่างการแสดงผลการเปิดช่องสัญญาณโดยมีช่อง 1- 8 49
รูปที่ 4.22 แสดงตัวอย่างการแสดงผลการปิดช่องสัญญาณโดยมีช่อง 1- 8 49
รูปที่ 4.23 แสดง "Control # ALL #" ซึ่งเป็นการเปิด/ปิดช่องสัญญาณทั้ง 8 ช่อง 50
รูปที่ 4.24 แสดง "Link Wait" รีโมทคอนโทรลไม่พร้อมรับคำสั่งการใช้งาน 50
รูปที่ 4.25 แสดงส่วนที่เป็นตัวแม่ 50
รูปที่ 4.26 แสดงโครงสร้างภายในส่วนของตัวแม่ 51
รูปที่ 4.27 แสดงโครงสร้างด้านหลังส่วนของตัวแม่ 51
รูปที่ 4.28 แสดงโครงสร้างภายในของรีโมทคอนโทรล 52
รูปที่ 4.29 แสดงรีโมทคอนโทรล 52
รูปที่ 4.30 แสดงโมดูลอาร์เอฟที่ต่ออยู่กับตัวรีโมทและตัวแม่ 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช Atmel 4

ตารางที่ 2.2 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ Atmel การใช้งานเป็นพอร์ตเอาท์พุต 10

ตารางที่ 2.3 การเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์ R0-R7 [illegible]

ตารางที่ 2.4 ขาสัญญาณของแอลซีดีโมดูล 26

ตารางที่ 2.5 ชุดคำสั่งควบคุมและการแสดงข้อความ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

การดำรงชีวิตของมนุษย์ในยุคที่โลกมีความเจริญก้าวหน้าทางด้านเทคโนโลยี ไม่ว่าจะเป็นเรื่องของการติดต่อสื่อสาร มนุษย์ก็สามารถที่จะติดต่อสื่อสารถึงกันได้แม้ว่าจะอยู่กันคนละซีกของโลกโดยผ่านทางอินเทอร์เน็ต สิ่งต่าง ๆ ที่อยู่รอบตัวของมนุษย์ มีการพัฒนาอย่างต่อเนื่องในช่วงระยะเวลาไม่กี่ปีจนถึงปัจจุบัน มนุษย์พยายามที่จะคิดค้นสิ่งประดิษฐ์ใหม่ ๆ ขึ้น เพื่อช่วยอำนวยความสะดวกให้การดำรงชีวิตง่ายขึ้น อย่างเช่นอุปกรณ์เครื่องใช้ไฟฟ้านับได้ว่าเป็นสิ่งที่จำเป็นสำหรับการดำรงชีวิตของมนุษย์ทุกครอบครัวล้วนต้องพึ่งพาสิ่งอำนวยความสะดวกเหล่านี้

1.1 แนวความคิดและที่มาของของการทำปริญญานิพนธ์

อุปกรณ์เครื่องใช้ไฟฟ้านับได้ว่าเป็นส่วนสำคัญต่อการดำรงชีวิตของมนุษย์เป็นอย่างมาก ไม่ว่าจะเป็น โทรทัศน์ , วิทยุ , พัดลม , วีดีโอ แม้กระทั่งหลอดไฟ ปัจจุบันเครื่องใช้ไฟฟ้าต่าง ๆ เหล่านี้พัฒนาก้าวหน้าไปอย่างรวดเร็วมาก นอกจากการเปิดปิดโดยปกติที่ใช้สวิตซ์แล้ว ยังเพิ่มการควบคุมด้วยรีโมทคอนโทรล ซึ่งถือได้ว่าเป็นเสมือนแขน ขา ของมนุษย์ ช่วยให้มนุษย์สะดวก สบายมากยิ่งขึ้น โดยเครื่องใช้ไฟฟ้าเหล่านี้มีข้อจำกัดในการส่งงานเฉพาะของแต่ละชนิด

จากเหตุผลดังกล่าว การที่จะสร้างเครื่องควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าเหล่านี้ให้สามารถเปิดปิดโดยใช้รีโมทเพียงตัวเดียวเป็นเรื่องน่าสนใจ และน่าจะเป็นไปได้โดยเครื่องควบคุมอุปกรณ์ไฟฟ้านี้จะใช้ไมโครคอนโทรลเลอร์เป็นส่วนประกอบหลักที่สำคัญโดยที่ไมโครคอนโทรลเลอร์จะทำงานตามโปรแกรมที่ออกแบบขึ้น

1.2 ความมุ่งหมายและวัตถุประสงค์ของการทำปริญญานิพนธ์

1.2.1 เพื่อศึกษาโครงสร้างทางสถาปัตยกรรม ตลอดจนลักษณะการเขียนโปรแกรมคำสั่งของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

1.2.2 เพื่อศึกษาหลักการทำงานของระบบรีโมทคอนโทรล

1.2.3 เพื่อฝึกทักษะในการเขียนโปรแกรมคำสั่งให้ไมโครคอนโทรลเลอร์ทำงานตามที่ได้ออกแบบไว้

1.2.4 เพื่อสร้างเครื่องควบคุมอุปกรณ์ไฟฟ้าชนิดต่าง ๆ ให้สามารถควบคุมการเปิด-ปิดโดยการสั่งงานจากรีโมทคอนโทรลและรับค่าสถานะกลับมาเพื่อแสดงผลได้

1.3 ขอบเขตของการทำปริญญานิพนธ์

สร้างรีโมทคอนโทรลเพื่อควบคุมอุปกรณ์ไฟฟ้า 8 ช่องสัญญาณด้วยย่านความถี่วิทยุโดยใช้โมดูลสำเร็จรูปโดยใช้รีโมทคอนโทรล ที่สร้างรหัสคำสั่งขึ้นมาส่วนตัวควบคุมใช้คอนโทรลเลอร์ เป็นตัวประมวลผลสัญญาณที่รับจากรีโมทคอนโทรล โดยผ่านทางตัวรับสัญญาณโมดูลอาร์เอฟ (Radio Frequency: RF) และทำการส่งสัญญาณไปทำการประมวลเพื่อควบคุมการเปิด-ปิดอุปกรณ์เครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งต่ออยู่กับตัวแม่ (Master) โดยทำการประมวลผลตามรหัสคำสั่งที่ได้รับ และจะส่งค่าสถานะของการประมวลกลับไปยังรีโมทคอนโทรลซึ่งรับสัญญาณโดยโมดูลอาร์เอฟ เพื่อส่งสัญญาณไปประมวล เพื่อแสดงสถานะผ่านจอแอลซีดี (LCD) ของรีโมทคอนโทรลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ไมโครคอนโทรลเลอร์ MCS-51

2.1.1 โครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่ใช้จะพูดถึงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ซึ่งมีหน่วยความจำแบบแฟลช (flash memory) ของเอทเมล (Atmel Corporation) มีเบอร์ขึ้นต้นด้วย AT89 เหตุผลที่ใช้ไมโครคอนโทรลเลอร์แบบนี้ในการเรียนรู้เพื่อใช้งาน ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มีด้วยกันหลายประการดังนี้

หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์เป็นแบบแฟลช ทำให้สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิปเดี่ยวจึงไม่มีหน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ

ต้นทุนและเวลาในการพัฒนาระบบไมโครคอนโทรลเลอร์ลดลงอย่างมาก เนื่องมาจากไม่ต้องใช้เครื่องมือพัฒนาจำพวกอีมูเลเตอร์และเครื่องโปรแกรมอีพรอม

บริษัทผู้ผลิตได้ทำการผลิตไมโครคอนโทรลเลอร์ตระกูลนี้ออกมาหลายเบอร์และมีความสามารถแตกต่างกันไปทำให้มีทางเลือกในการใช้งานสูง

ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ซึ่งจะทำให้สามารถทำการป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้อย่างดี

ในบางเบอร์ของไมโครคอนโทรลเลอร์ที่ผลิตโดย เอทเมลสามารถที่จะทำการโปรแกรมข้อมูลในหน่วยความจำโปรแกรมได้ โดยที่ไม่ต้องถอดตัวไมโครคอนโทรลเลอร์ออกมาทำการโปรแกรมใหม่หรือเรียกว่า การโปรแกรมในวงจรหรือ ในระบบ (In-system Programming) ทำให้การพัฒนาหรือการซ่อมบำรุงจนถึงการปรับปรุงหรืออัพเกรดข้อมูลในหน่วยความจำโปรแกรมได้อย่างสะดวก ภายใต้งบประมาณที่ไม่สูงมากนัก

ชุดคำสั่งและสถาปัตยกรรมพื้นฐานเหมือนระบบไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตอื่นไม่ว่าจะเป็นอินเทล ซีเมนต์ หรือ ดัลลัส

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
- ภายในหน่วยความจำโปรแกรมเป็นแบบแฟลชสามารถลบและเขียนใหม่ได้พันครั้ง
- หน่วยความจำพื้นฐานเป็นหน่วยความจำแบบแรม บางเบอร์มีหน่วยความจำแบบอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานได้ทั้งอินพุตและเอาต์พุต
- ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

ในรูปที่ 2.1 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx จะเห็นได้ว่าโครงสร้างของ AT89Cxx จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐานหากแตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา

สำหรับรูปที่ 2.2 เป็นโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Sxx จะเห็นได้ว่ามีส่วนประกอบเพิ่มเติมแตกต่างจาก AT89Cxx อยู่หลายส่วน อาทิวงจรเชื่อมต่ออนุกรมแบบเอสพีไอ(SPI)ซึ่งในไมโครคอนโทรลเลอร์อนุกรมนี้ใช้ในการเขียนข้อมูลลงในหน่วยความจำโปรแกรมโดยไม่ต้องถอดตัวชิปออกไปจากระบบหรือเรียกว่าการโปรแกรมภายในวงจร ไทเมอร์/เคาน์เตอร์ขนาด 16 บิตที่เพิ่มเติมเข้ามาอีกหนึ่งตัวเป็นไทเมอร์ 2 และวงจรวอตช์ด็อกใช้ในการตรวจสอบการทำงานผิดพลาดของซีพียู

ในตารางที่ 2.1 แสดงรายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แต่ละเบอร์ที่เอทเมล ผลิตขึ้น และมีใช้งานอยู่ในปัจจุบัน

ตารางที่ 2.1 รายละเอียดโดยสรุปบางส่วนของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช Atmel

เบอร์ของไมโครคอนโทรลเลอร์	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	จำนวนไทเมอร์/เคาน์เตอร์ 16 บิต
AT89C1051	แบบแฟลช ขนาด 1 กิโลไบต์	แรม 64 ไบต์	1
AT89C2051	แบบแฟลช ขนาด 2 กิโลไบต์	แรม 128 ไบต์	2
AT89C51	แบบแฟลช ขนาด 4 กิโลไบต์	แรม 128 ไบต์	2
AT89C52	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์	3
AT89C55	แบบแฟลช ขนาด 20 กิโลไบต์	แรม 256 ไบต์	3
AT89S8252	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์ อีอีพรอม 2 กิโลไบต์	3
AT89S53	แบบแฟลช ขนาด 12 กิโลไบต์	แรม 64 ไบต์	3

2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.3 และ 2.4 โดยมีรายละเอียดขั้นต้น ดังนี้

ขาแหล่งจ่าย (Vcc) ใช้ต่อไฟเลี้ยง 5โวลต์ ขากราวด์ (GND) สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำการได้โดยการเขียน

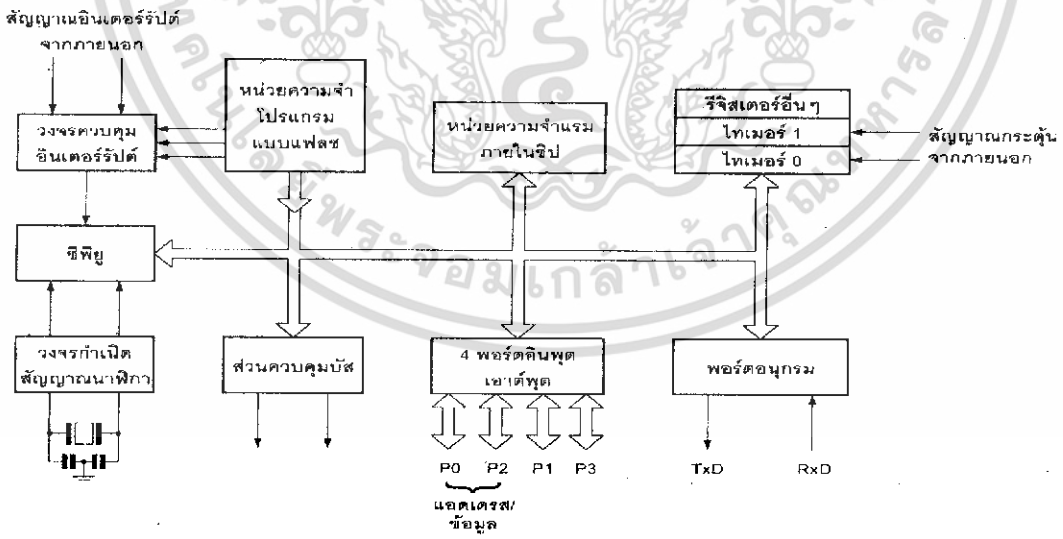
เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลทำให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำติดต่อภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กเข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทเมอร์ 2 และขา P1.1 เป็นขาอินพุตทริกเกอร์ของไทเมอร์ 2 ในขณะที่ขา P1.4 - P1.7 เป็นขาเชื่อมต่อแบบเอสพีไอเพื่อทำการโปรแกรม

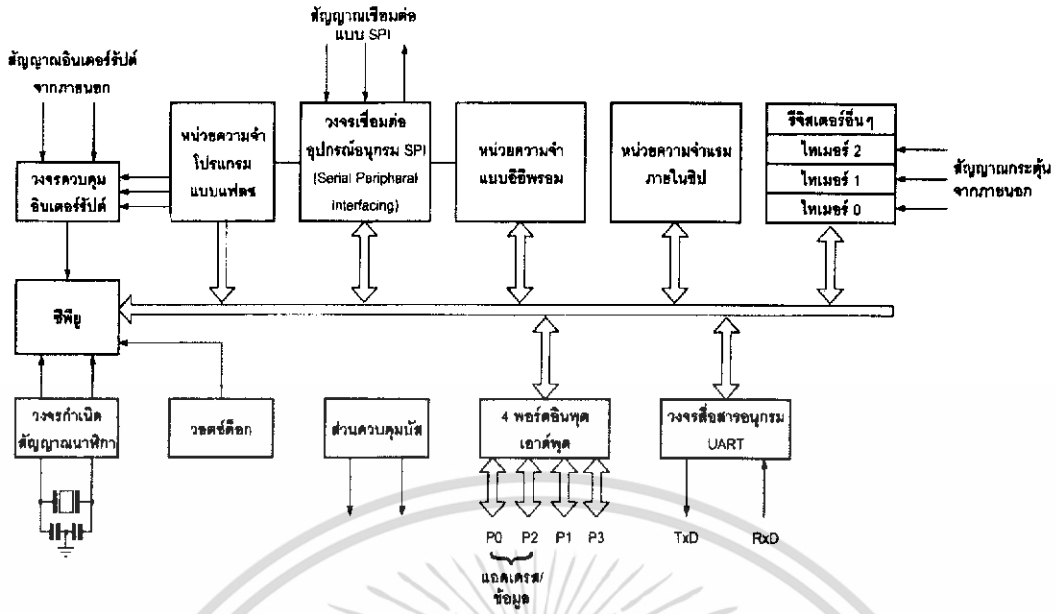
ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไปถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลทำให้ขาพอร์ตนั้นมีสถานะปล่อยลอย จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำติดต่อภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไปถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำการได้โดยการเขียนข้อมูล "1"ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลทำให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่ใช้งานพิเศษ ดังมีรายละเอียดดังต่อไปนี้



รูปที่ 2.1 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Cxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89Sxx

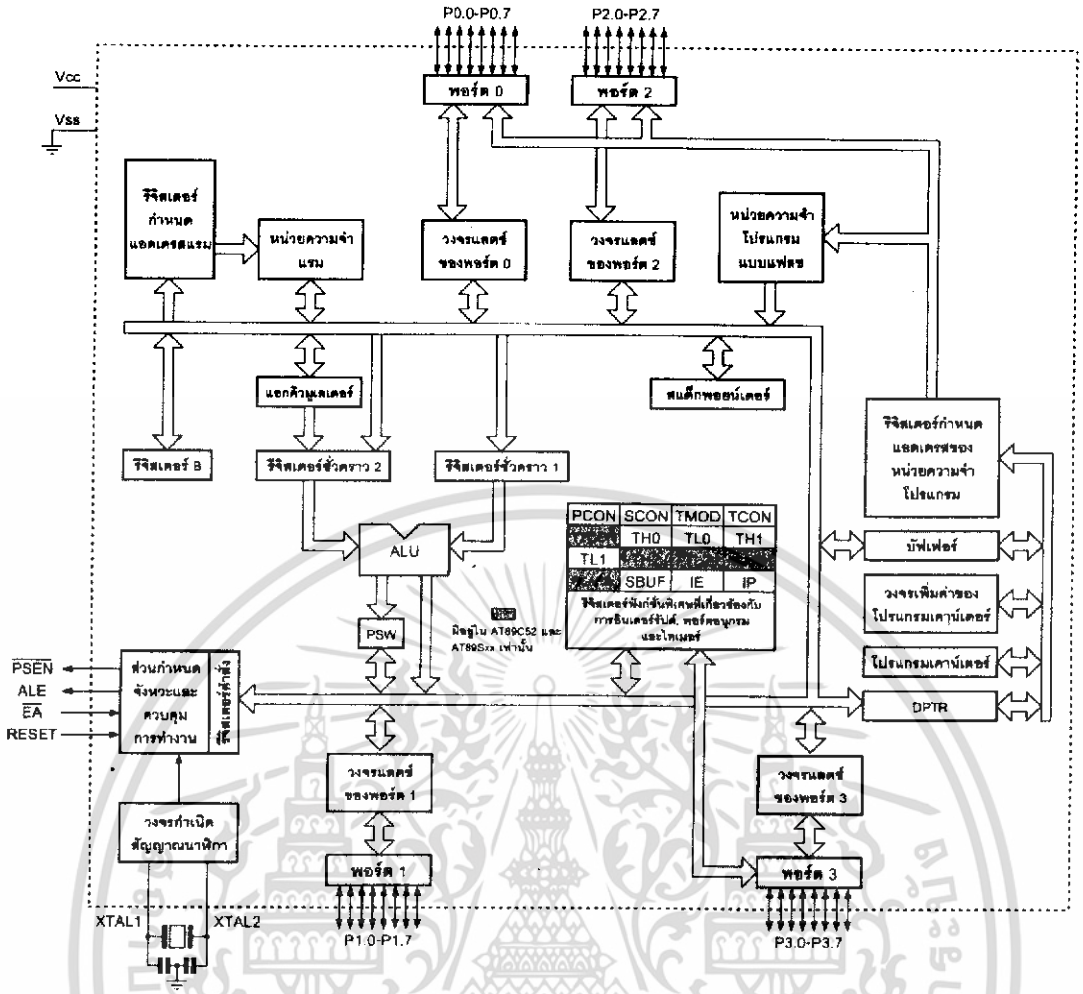
- P 3.0 ใช้เป็นขาอินพุตสำหรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD
- P 3.1 ใช้เป็นขาอินพุตสำหรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD
- P 3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปจากภายนอกช่อง 0 หรือขา INT0
- P 3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปจากภายนอกช่อง1 หรือขา INT1
- P 3.4 ใช้เป็นขารับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0
- P 3.5 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปจากภายนอกช่อง 1 หรือขา T1
- P 3.6 ใช้เป็นขาสัญญาณ WR ในกรณีใช้กับหน่วยความจำภายนอก
- P 3.7 ใช้เป็นขาสัญญาณ RD ในกรณีใช้กับหน่วยความจำภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณรีเซตสถานะ ที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเกิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

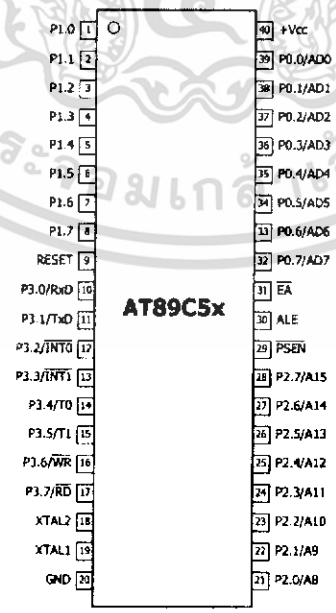
ขา ALE/PROG (Address Larch Enable/Program Pulse Input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนี้ขานี้ยังใช้เป็นขาสำหรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณ เพื่อร้องกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมที่อยู่ภายนอก ตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก เนื่องจากตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชีนไซเกิลแต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช



รูปที่ 2.4 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอกแต่ถ้าขานี้เป็น "1" เป็นการเลือกให้ตัวไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในของตัวคอนโทรลเลอร์

2.1.3 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ตคือ พอร์ต 0 ถึง พอร์ต 3 แต่ละพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับป้อนข้อมูลเข้าและเอาต์พุตสำหรับส่งข้อมูลออกทุกพอร์ตไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชเบอร์ใด ๆ ดังสรุปได้ในตารางที่ 2.2

ในรูปที่ 2.5 แสดงวงจรภายในของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยในรูปที่ 2.5 (ก) เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิปฟลอปนั่นเอง การอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณ มายังขา CLK ของดีฟลิปฟลอปในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิปฟลอป ที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับการกำหนดลักษณะการทำงานของพอร์ตว่า ต้องการใช้งานเป็นขาพอร์ตอินพุตเอาท์พุทปกติ หรือใช้การติดต่อกับหน่วยความจำภายนอกไมโครคอนโทรลเลอร์

เนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัปภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นอินพุตจะต้องต่อตัวต้านทานพูลอัปภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

ในรูปที่ 2.5 (ข) เป็นวงจรของพอร์ต 1 ซึ่งมีลักษณะโดยทั่วไปคล้ายกับพอร์ต 0 หากแต่ไม่มีวงจรมัลติเพล็กซ์เนื่องจากพอร์ตนี้จะไม่ใช้ในการติดต่อกับหน่วยความจำภายนอก แต่จะมีวงจรพูลอัปภายในที่ละบิตของพอร์ตนี้แทน สำหรับรายละเอียดของวงจรพูลอัป แสดงในรูปที่ 2.6

ในรูปที่ 2.5 (ค) เป็นวงจรภายในของพอร์ต 2 จะคล้ายกับพอร์ต 0 มาก ต่างเพียงมีวงจรพูลอัปเพิ่มเข้ามา ส่วนในรูปที่ 2.5 (ง) เป็นวงจรภายในของพอร์ต 3 จะเห็นว่าคล้ายกับพอร์ต 1 มีการเพิ่มเติมวงจรบัฟเฟอร์ และวงจรอินพุตเอาต์พุตเมื่อทำงานในฟังก์ชั่นพิเศษเข้ามา เนื่องจากพอร์ต 3 สามารถนำไปใช้งานในหน้าที่พิเศษได้ทุกขา

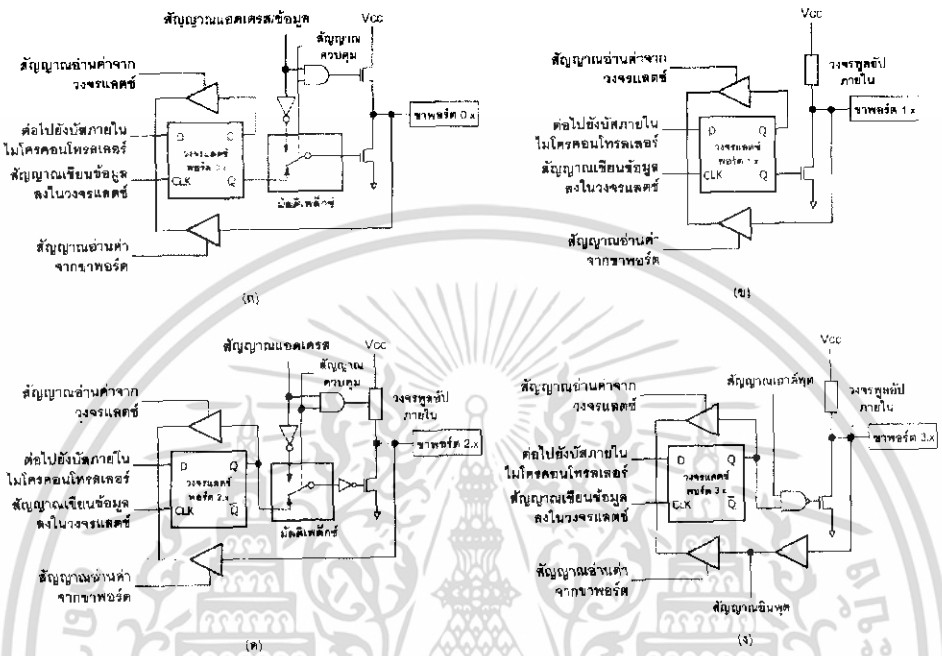
2.1.3.1 การใช้งานเป็นพอร์ตอินพุต

เนื่องจากพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุตดังนั้น จึงมีความจำเป็นอย่างยิ่งต้องทำความเข้าใจถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟต (FET) ที่ใช้ในการขับสัญญาณเอาต์พุตของบิตนั้น ๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรงส่งผลให้ขาพอร์ตนั้นมีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น"1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้ซีพียูมาอ่านเข้าไปเมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อกับอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0" จะดีและสะดวกที่สุด



รูปที่ 2.5 วงจรภายในของพอร์ตทุกพอร์ตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะที่เป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูล ออกไปได้อย่างง่ายดาย และตรงไปตรงมา กล่าวคือเมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาต์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับเฟต ทำให้เฟตทำงาน ที่ขาพอร์ตที่กำหนดให้ทำ... ก็จะเกิดลอจิก "0" ขึ้น ในทางตรงข้ามหากต้องการส่งข้อมูล "1" ออกไปก็ให้เขียนข้อมูล "1" ไปยังวงจรแลตช์วงจรก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัปภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมากเพียงแต่แตกต่างกัน ที่กระบวนการของการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่กรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานเป็นพอร์ตเอาต์พุต แต่ละขา (หรือละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (Source current) ได้สูงสุด 10 มิลลิแอมป์ (mA) และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 มิลลิแอมป์ สำหรบพอร์ต 0 และ 15 มิลลิแอมป์ สำหรับพอร์ต 1-3 ในก...ที่ใช้งานทุกพอร์ตเอาต์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 มิลลิแอมป์ ดังนั้นในการใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแส ควรต่อวงจรบัฟเฟอร์นี้ทางเอาต์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 การอ่านค่าลอจิกจากพอร์ต

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะ คือ อ่านจากขาพอร์ตโดยตรง และอ่านจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อกับขาเบสของทรานซิสเตอร์ ชนิดที่เป็น เอ็นพีเอ็น (NPN) และขาอิมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่ขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงาน จะเสมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงข้ามกับที่ส่งออกมา แต่หากทำการอ่านค่าลอจิกที่วงจรแลตช์ จะได้ค่าที่ต้องการส่งจริง ดังนั้นในการอ่านค่าลอจิกจากพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

ตารางที่ 2.2 หน้าที่พิเศษของพอร์ต 1 ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชของ Atmel การใช้งานเป็นพอร์ตเอาท์พุต

ขา	เบอร์ของไมโครคอนโทรลเลอร์	หน้าที่พิเศษ
P1.0	AT89C52/AT89Sxx	ขา T2 เป็นขาอินพุตนับค่าของไทเมอร์/เคาน์เตอร์ 2
P1.1	AT89C52/AT89Sxx	ควบคุมทิศทางของสัญญาณ
P1.4	AT89Sxx	ขา SS (Slave Select) เป็นขาเลือกการติดต่อในกรณีที่ไมโครคอนโทรลเลอร์เป็นอุปกรณ์สเลฟ ในระบบการติดต่อแบบ SPI
P1.5	AT89Sxx	ขา MOSI (Master data output, Slave data input) ใช้ในการติดต่อกับพอร์ต SPI
P1.6	AT89Sxx	ขา MOSI (Master data input, Slave data output) ใช้ในการติดต่อกับพอร์ต SPI
P1.7	AT89Sxx	ขา SCK (Master clock output) เป็นขาสัญญาณนาฬิกาของการติดต่อกับพอร์ต SPI

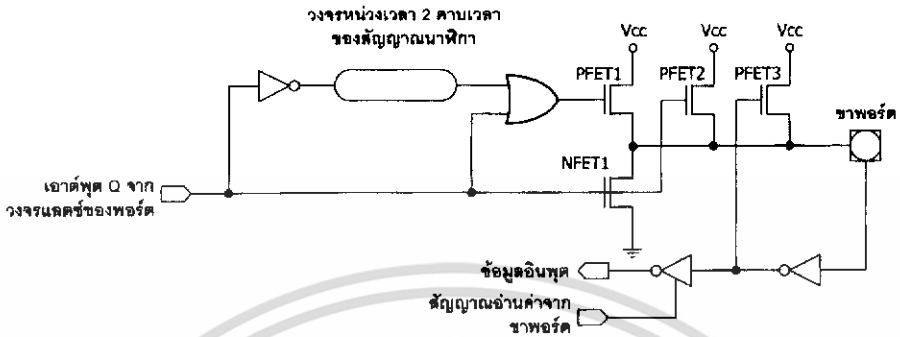
2.1.4 การทำงานของไมโครคอนโทรลเลอร์ MCS-51

ในการใช้งานไมโครคอนโทรลเลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียู และลำดับขั้นตอนการประมวลผลคำสั่งและในการประมวลผลคำสั่งของซีพียูจะมีขั้นตอนหลัก ๆ 2 ขั้นตอนคือ กระบวนการเฟตช์ (Fetch) เป็นการเรียกคำออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผลขั้นตอนต่อมาคือกระบวน การเอ็กซ์คิวต์ (Execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมา โดยกระบวนการก่อนหน้านี้เมื่อทำการเอ็กซ์คิวต์คำสั่งเรียบร้อยแล้วก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์จะเกิดการรีเซตลักษณะที่เรียกว่า เพาเวอร์ออนรีเซต

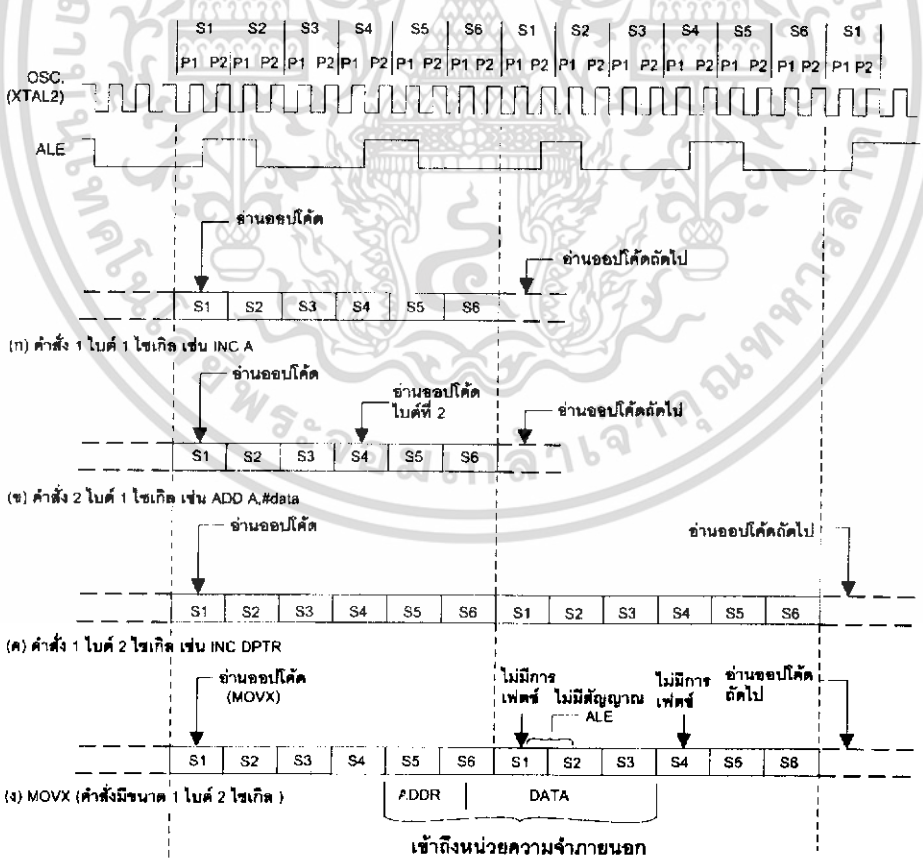
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ หาประโยชน์ในเชิงการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Power on reset) ซีพียูเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรมจังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบโดยได้รับการกำหนดมาจากรอบการทำงานแมชีนไซเกิลในรูปที่ 2.7 เป็นไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยในหนึ่งรอบการทำงาน



วงจรพูลอัปประกอบด้วยเฟตชนิดพีแชนเนล 3 ตัวคือ PFET1-PFET3 โดย NFET1 จะทำงานเมื่อได้รับลอจิก "1" จากขา Q และหยุดทำงานเมื่อได้รับลอจิก "0" วงจรพูลอัปจะเริ่มต้นทำงานเมื่อ NFET1 ได้รับลอจิก "1" จาก PFET1 จะทำงานประมาณ 2 คาบเวลาของสัญญาณนาฬิกาภายใน หลัง จากที่เกิดการเปลี่ยนแปลงจากลอจิก "0" เป็นลอจิก "1" ในขณะที่ PFET1 ทำงานจะทำให้ PFET3 ทำงานตามไปด้วย ทำให้เกิดการพูลอัปขาพอร์ต

รูปที่ 2.6 วงจรพูลอัปภายในพอร์ตไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช



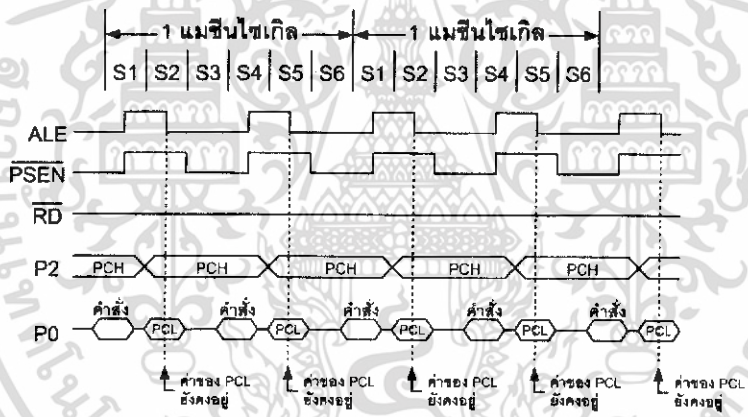
รูปที่ 2.7 ไซเกิลการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

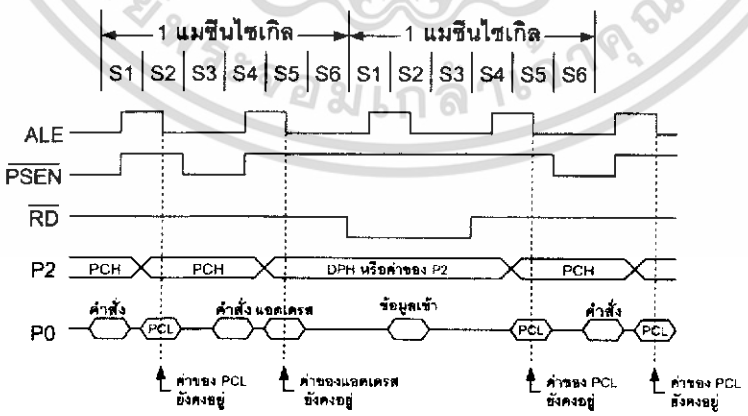
หรือแมชีนไซเกิลจะแบ่งย่อยออกเป็น 6 สเตต (State) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตต มีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกาถ้าสัญญาณนาฬิกามีความถี่ 12 เมกกะเฮิร์ต (MHz) จะมีคาบเวลาเท่ากับ 1 มิลลิวินาที (mS) คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่าเฟส 1 (Phase 1) และเฟส 2 (Phase 2)

ในรูปที่ 2.7 (ก) และ (ข) จะเป็นการเอ็กซ์คิวต์คำสั่งที่ใช้เวลา 1 ไซเกิล เริ่มต้นที่สเตต 1 จะเป็นการอ่านค่าออฟโค้ด อันเป็นกระบวนการแลตช์ค่าของออฟโค้ด ส่งไปให้รีจิสเตอร์คำสั่ง (Instruction Register : IR) การเฟตช์ครั้งที่ 2 จะเกิดขึ้นที่สเตต 4 ภายในแมชีนไซเกิลเดียวกันในกรณีที่เป็นคำสั่งไบต์เดียว การเฟตช์ครั้งที่ 2 ภายในแมชีนไซเกิลเดียวกันจะถูกตัดทิ้งไป ในคำสั่งที่มีใช้เวลา 1 ไซเกิล จะสิ้นสุดการทำงานลงในสเตต 6 ของแมชีนไซเกิลเดียวกัน

ในกรณีที่คำสั่งใช้เวลา 2 ไซเกิล การทำงานของคำสั่งนั้นจะสิ้นสุดลงในสเตต 6 ของแมชีนไซเกิลที่ 2 ดังในไดอะแกรมรูปที่ 2.7 (ค) สำหรับในการกระทำคำสั่ง MOVX ซึ่งเป็นคำสั่งขนาด 1 ไบต์ 2 ไซเกิล จะไม่มีการเฟตช์เกิดขึ้นในไซเกิลที่สองของคำสั่ง MOVX นี้เนื่องจากซีพียูจะไปทำการติดต่อกับหน่วยความจำภายนอกดังแสดงในไดอะแกรมรูปที่ 2.7 (ง) และจะเห็นได้ว่าเวลาในการเอ็กซ์คิวต์จะไม่ได้ขึ้นอยู่กับว่าทำการติดต่อหน่วยความจำโปรแกรมภายในหรือภายนอก



(ก) เมื่อไม่กระทำคำสั่ง MOVX



(ข) เมื่อกระทำคำสั่ง MOVX

รูปที่ 2.8 ไดอะแกรมเวลาแสดงการติดต่อและเข้าถึงหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.8 สัญญาณและไดอะแกรมเวลาของการเข้าถึงหน่วยความจำโปรแกรมภายนอกมีอยู่ในรูปที่ 2.8 (ก) เป็นไดอะแกรมเวลาในขณะหนึ่งที่ยังไม่มีการกระทำคำสั่ง MOVX สัญญาณที่ขา ALE และ PSEN เกิดการแอกตีฟ 2 ครั้งภายในหนึ่งแมชีนไซเกิลในทุกครั้งที่ ALE เกิดการแอกตีฟที่พอร์ต 0 (P0) จะมีค่าของรีจิสเตอร์พีซีในไบต์ต่ำออกมา ในขณะที่พอร์ต 2 (P2) ก็มีค่าของพีซีในไบต์สูงเพื่อชี้ไปยังแอดเดรสต่อไปที่ต้องไปดำเนินการสำหรับขา PSEN ก็จะเกิดการแอกตีฟขึ้น เมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอก ในกรณีที่กระทำคำสั่ง MOVX เพื่อเข้าถึงหน่วยความจำข้อมูลภายนอก ที่ขา PSEN จะไม่เกิดการแอกตีฟ 2 ครั้งภายใน 1 แมชีนไซเกิล เนื่องจากบัสแอดเดรสและบัสข้อมูลจะใช้ในการติดต่อหน่วยความจำภายนอกแทนแต่สำหรับสัญญาณ ALE ยังคงแอกตีฟตามจังหวะการทำงานเหมือนเดิม จากไดอะแกรมเวลาสามารถสรุปได้ว่า ในการทำงาน 1 รอบหรือ 1 แมชีนไซเกิล ซีพียูในไมโครคอนโทรลเลอร์ MCS-51 จะใช้เวลา 12 คาบเวลาของสัญญาณนาฬิกา นั่นคือ เวลาในการทำงาน 1 ไซเกิลมีค่าเท่ากับ 1 มิลลิวินาทีหรือมีความเร็วในการทำงานภายใน 1 เมกกะเฮิร์ต ในกรณีที่ใช้ความถี่สัญญาณนาฬิกา 12 เมกกะเฮิร์ต ดังนั้นถ้าต้องการทราบความเร็วของการทำงานภายในของตัวไมโครคอนโทรลเลอร์ MCS-51 สามารถหาได้จากค่าความถี่สัญญาณนาฬิกาหารด้วย 12 และถ้าต้องการหาค่าเวลาของ 1 รอบของการทำงานหรือ 1 แมชีนไซเกิล ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถทำได้โดยการหาส่วนกลับของความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์ MCS-51 สามารถสรุปเป็นสูตรทางคณิตศาสตร์ได้ดังนี้

ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์ เท่ากับความถี่ของสัญญาณนาฬิกา (ค่าของคริสตอลที่ต่ออยู่ที่ขา XTAL1 และ XTAL2)/12

เวลา 2 แมชีนไซเกิล = 1/ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์

2.1.5 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำภายในหลัก ๆ อยู่ 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูลซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปแต่ละเบอร์ ในบทนี้จะกล่าวถึงรายละเอียดของการจัดสรรหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช การเชื่อมต่อกับหน่วยความจำภายนอกและข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชันพิเศษที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

2.1.5.1 หน่วยความจำโปรแกรม (Program Memory)

รูปที่ 2.9 แสดงการจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในเบอร์ต่าง ๆ ที่นิยมใช้ อันประกอบด้วยเบอร์ AT89C51 และ AT89C52 จะเห็นได้ว่าทั้งสองเบอร์สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ โดยสามารถเลือกใช้หน่วยความจำโปรแกรมภายในอย่างเดียวหรือร่วมกับภายนอกหรือเลือกใช้หน่วยความจำภายนอกอย่างเดียวก็ได้ ดังในรูปที่ 2.9 (ก) โดยภายใน AT89C51 จะมีหน่วย ความจำโปรแกรมภายใน 4 กิโลไบต์ ในขณะที่ AT89C52 มีขนาด 8 กิโลไบต์

ในกรณีใช้หน่วยความจำภายในภายนอกร่วมกัน หากใช้ AT89C51 ก็จะสามารถติดต่อกับหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำภายนอกได้ 60 กิโลไบต์ และถ้าใช้เบอร์ AT89C52 ก็จะสามารถติดต่อกับหน่วยความจำโปรแกรมภายนอกได้ 56 กิโลไบต์

หน่วยความจำที่ใช้เก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ หรือที่เรียกว่าโปรแกรมมอนิเตอร์ (Monitor Program) หากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม (EPROM: Erasable Programmable Read-only Memory) ซึ่งสามารถทำการอ่านได้เพียงอย่างเดียว

หน่วยความจำโปรแกรมมีแอดเดรสเริ่มต้นที่ 0000H เมื่อซีพียูได้รับการรีเซตให้เริ่มต้นการทำงานจะต้องมาเริ่มต้นที่แอดเดรส 0000H นี้เสมอ อย่างไรก็ตามในพื้นที่หน่วยความจำโปรแกรมไม่ว่าจะใช้งานจากภายในหรือภายนอกก็ตาม ต้องมีการสงวนพื้นที่บางตำแหน่งเอาไว้สำหรับการบริการอินเตอร์รัปต์ 6 ประเภท ประเภทละ 8 ไบต์ ประกอบด้วย

- พื้นที่สำหรับการบริการอินเตอร์รัปต์ 0 จากภายนอก กำหนดไว้ที่แอดเดรส 0003H
- พื้นที่สำหรับการบริการอินเตอร์รัปต์จากไทเมอร์ 0 กำหนดไว้ที่แอดเดรส 000BH
- พื้นที่สำหรับการบริการอินเตอร์รัปต์ 1 จากภายนอก กำหนดไว้ที่แอดเดรส 0013H
- พื้นที่สำหรับการอินเตอร์รัปต์จากไทเมอร์ 1 กำหนดไว้ที่แอดเดรส 001BH
- พื้นที่สำหรับการบริการอินเตอร์รัปต์ของแบบอนุกรม กำหนดไว้ที่แอดเดรส 0023H
- พื้นที่สำหรับการบริการอินเตอร์รัปต์จากไทเมอร์ 2 กำหนดไว้ที่แอดเดรส 002BH

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ที่มีหน่วยความจำโปรแกรมภายใน แต่ต้องการติดต่อกับหน่วยความนำโปรแกรมภายนอกด้วยสามารถทำได้โดยต้องกำหนดแอดเดรสของหน่วยความจำโปรแกรมให้ต่อจากแอดเดรสสุดท้ายของหน่วยความจำโปรแกรมภายในของตัวไมโครคอนโทรลเลอร์ ยกตัวอย่างไมโครคอนโทรลเลอร์ AT89C51 มีหน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H-0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอกต้องกำหนดให้แอดเดรสอยู่ในช่วง 1000H-FFFFH

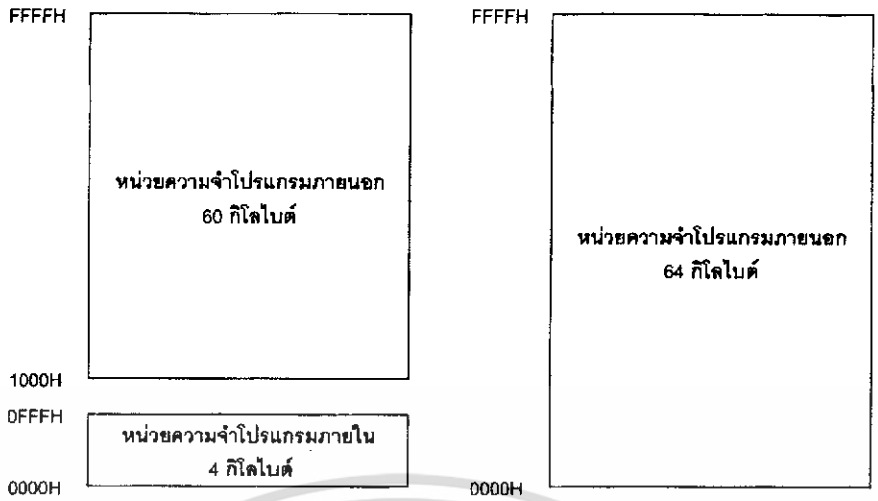
การต่อหน่วยความจำโปรแกรมภายนอกแสดงดังในรูปที่ 2.10 ขาพอร์ต P0.0-P0.7 ใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำโดยผ่านวงจรแลตช์ซึ่งปกติใช้ไอซีเบอร์ 74HC573 และใช้สัญญาณ ALE และ PSEN ในการใช้งานขา P0.0-P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรสในขณะที่ขา P2.0-P2.7 ใช้ในการเชื่อมต่อกับขาแอดเดรสที่ไบต์สูง A8 -A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอกไมโครคอนโทรลเลอร์จะเหลือขาพอร์ตเพียง 16 บิต คือที่ขาพอร์ต P1.0-P1.7 และ P3.0-P3.7

2.1.5.2 หน่วยความจำข้อมูล (Data memory)

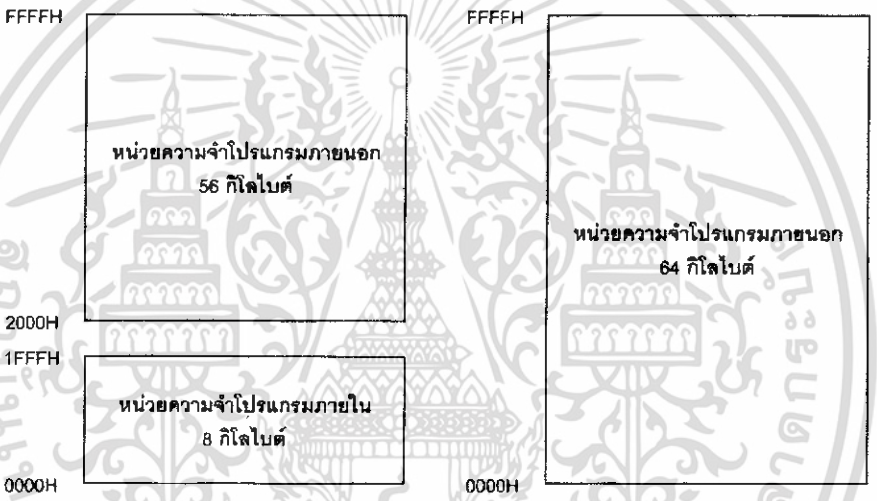
มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและภายใน โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89xx สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอกการติดต่อกับหน่วยความ

จำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงดังรูปที่ 2.11 เห็นได้ว่า มีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมภายนอกแตกต่างกันที่มีสัญญาณ ที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา RD และ WR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

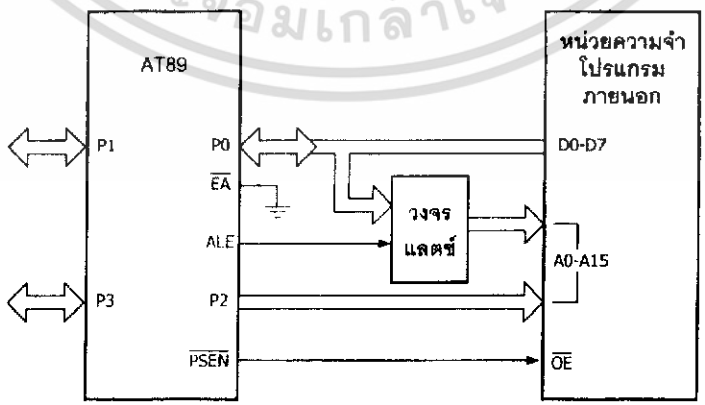


(ก) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C51



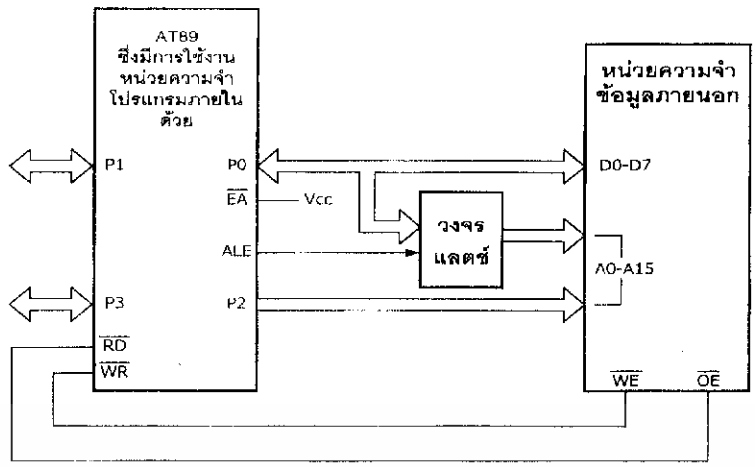
(ข) การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์เบอร์ AT89C52

รูปที่ 2.9 การจัดสรรหน่วยความจำโปรแกรมไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช



รูปที่ 2.10 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกของไมโครคอนโทรลเลอร์ MCS-51

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89xx ทุกเบอร์มีหน่วยความจำข้อมูลภายในเป็นแบบแรม (RAM : Random Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วน คือ หน่วยความจำข้อมูลส่วนล่าง (Lower) ส่วนบน (Upper) และรีจิสเตอร์ฟังก์ชันพิเศษ (SFR : Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์

หน่วยความจำส่วนบนและรีจิสเตอร์ฟังก์ชันพิเศษมีตำแหน่งทับซ้อนกันแต่จะใช้การติดต่อที่แตกต่างกัน และไมโครคอนโทรลเลอร์ MCS-51 บางเบอร์จะไม่มีหน่วยความจำข้อมูลส่วนบน

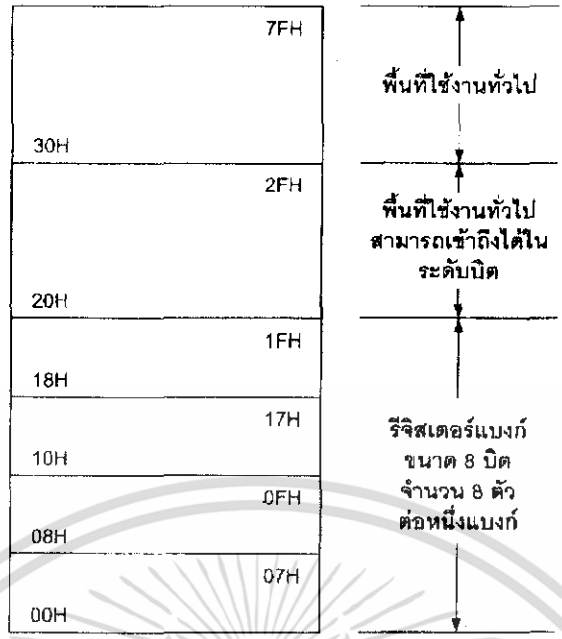
ขนาดของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยแท้จริงแล้วมีเพียง 256 ไบต์ แต่ด้วยการจัดการที่เข้าถึงที่แตกต่างกัน จึงดูเสมือนว่าไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อม สำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์ เช่นกัน มีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงแบบโดยอ้อมได้เท่านั้น ในขณะที่รีจิสเตอร์ฟังก์ชันพิเศษ มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ฟังก์ชันพิเศษ ใช้ในการเข้าถึงแบบโดยตรงดังนั้นเพื่อความสะดวกและง่ายดายตลอดจนป้องกันความสับสนในการเขียนโปรแกรม สำหรับผู้เริ่มต้นจึงควรใช้หน่วยความจำข้อมูลภายในเพียง 128 ไบต์ จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ฟังก์ชันพิเศษ

ในรูปที่ 2.12 แสดงการจัดสรรข้อมูลส่วนล่าง หน่วยความจำ 32 ไบต์ต่ำสุดที่แอดเดรส 00H-1FH แบ่งเป็น 4 กลุ่ม เรียกว่า 4 แบงก์ (Bank) แต่ละแบงก์มีรีจิสเตอร์ 8 ตัวคือ R0-R7 การติดต่อกับหน่วยความจำมีแบงก์ใดให้กำหนดที่รีจิสเตอร์แสดงสถานะของโปรแกรม

หน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไปสามารถเข้าถึงได้ในระดับบิต (Bit Addressable) และหน่วยความจำข้อมูลที่เหลือ 80 ไบต์จะต้องแบ่งส่วนหนึ่งไว้สำรองพื้นที่ของสแต็ก (Stack : ที่พักชั่วคราวในกรณีที่ซีพียูมีการกระโดดไปทำงานในโปรแกรมย่อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.12 การจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในส่วนล่างของไมโครคอนโทรลเลอร์ MCS- 51 แบบแฟลช

ในรูปที่ 2.13 (ก) แสดงโครงสร้างของหน่วยความจำข้อมูลส่วนบน ซึ่งจะมีลักษณะที่คล้ายกับหน่วยความจำข้อมูลส่วนล่างหากแต่ใน 80 ไบต์บนไม่จำเป็นต้องสำรองไว้สำหรับสแต็ก และต้องใช้การเข้าถึงในลักษณะโดยอ้อมเท่านั้น

2.1.5.3 รีจิสเตอร์ฟังก์ชันพิเศษ (SFR: Special Function Register)

เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 มีด้วยกัน 22 ตัว สำหรับเบอร์ AT89C51 และ 28 ตัวในเบอร์ AT89C52 และอนุกรม AT89S5xx ทั้งนี้เนื่องจากใน AT89C52 และ AT89S5xx มีจำนวนไทเมอร์เคาน์เตอร์มากกว่า AT89C51

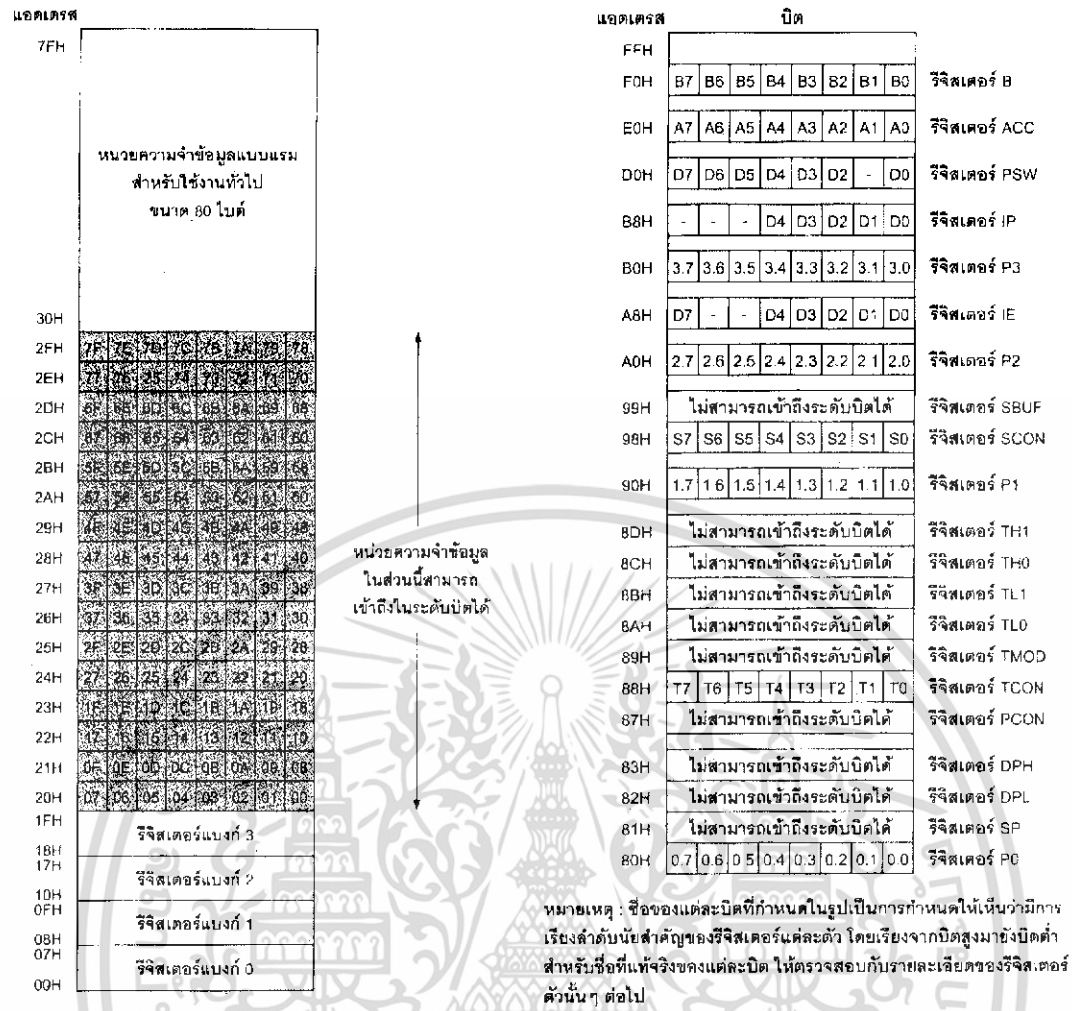
รีจิสเตอร์ฟังก์ชันพิเศษ มีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบนสามารถเข้าถึงได้โดยตรง (Direct addressing) ในรูปที่ 2.13 (ข) แสดงการจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษ แต่ละตัวในหน่วยความจำข้อมูลส่วนบน

2.1.5.4 รีจิสเตอร์แสดงสถานะของโปรแกรม (Program Status Word: PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต จึงสามารถกำหนดค่าในแต่ละบิตของรีจิสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสที่ D0H ทำหน้าที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้นจะเรียกสถานะต่าง ๆ ของโปรแกรมว่า แฟลก (Flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนสถานะขึ้น ผลของการเปลี่ยนแปลงนั้นจะมาปรากฏที่บิตต่าง ๆ ของรีจิสเตอร์แสดงสถานะของโปรแกรม รายละเอียดของแต่ละบิตในรีจิสเตอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

62677



(ก) โครงสร้างของหน่วยความจำข้อมูลส่วนบนของภายใน MCS-51 (ข) การจัดสรรพื้นที่ของรีจิสเตอร์ฟังก์ชันพิเศษ

รูปที่ 2.13 แสดงโครงสร้างภายในและการจัดสรรพื้นที่ใช้งานภายในของระบบไมโครคอนโทรลเลอร์ MCS-51

จะเห็นได้ว่านอกจากรีจิสเตอร์แสดงสถานะของโปรแกรมถูกใช้ในการเก็บสถานะของโปรแกรมแล้วที่บิต RS0 และ RS1 ยังใช้ในการเลือกแบงก์ของหน่วยความจำส่วนล่าง ซึ่งเป็นพื้นที่ของรีจิสเตอร์ R0-R7 ด้วย ดังมีรายละเอียดแสดงในตารางที่ 2.3 โดยปกติแล้วในการใช้งานรีจิสเตอร์ R0-R7 ด้วย มักนิยมใช้แบงก์ 0 เป็นลำดับแรก หากไม่เพียงพอจึงเลือกแบงก์อื่น ๆ มาใช้ แต่ต้องระมัดระวังในการกำหนดค่าและลำดับการติดต่อให้ดี มิเช่นนั้นอาจทำให้การเขียนโปรแกรมเกิดการสับสนดังนั้นสำหรับผู้เริ่มต้นใช้งานไมโครคอนโทรลเลอร์ MCS-51 จึงควรเลือกใช้รีจิสเตอร์ R0-R7 ในแบงค์ 0 เพียงเบอร์เดียวให้ชำนาญก่อน

การกำหนดค่าของรีจิสเตอร์แสดงสถานะของโปรแกรมเพื่อเลือกใช้งานรีจิสเตอร์ R0-R7 ควรกำหนดไว้ตอนต้นของโปรแกรมเสมอเพื่อจะนำไปเขียนโปรแกรมเพื่อติดต่อกับรีจิสเตอร์ R0-R7 ได้อย่างสะดวกและไม่เกิดความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.5 แอกคิวมูเลเตอร์ (Accumulator: ACC)

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์ โดยเฉพาะอย่างยิ่งในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ที่ได้แก่ซีพียู เพื่อทำการประมวลผลต่อไป อาจเรียกสั้น ๆ ว่า รีจิสเตอร์ A หรือแอกคิวมูเลเตอร์รีจิสเตอร์นี้สามารถเข้าถึงในระดับบิตได้

รีจิสเตอร์ B มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง F0H มีหน้าที่พิเศษคือ หากต้องการคูณหรือการหารทางคณิตศาสตร์ ต้องนำข้อมูลที่ต้องการหารหรือคูณมาเก็บไว้ในรีจิสเตอร์ B แล้วจึงกระทำคำสั่งการคูณหรือหารกับค่าในรีจิสเตอร์ A ต่อไป

ในกรณีที่ไม่ได้มีการคูณหรือการหารข้อมูลสามารถใช้รีจิสเตอร์ B นี้ในการเก็บข้อมูลทั่วไปได้เหมือนกับรีจิสเตอร์ปกติและสามารถเข้าถึงในระดับบิตได้เช่นเดียวกับรีจิสเตอร์ A

2.1.5.6 โปรแกรมเคาน์เตอร์ (Program Counter: PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียู ต้องไปทำงานรีจิสเตอร์เป็นโปรแกรมเคาน์เตอร์ รีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้รวมกับรีจิสเตอร์ฟังก์ชั่นพิเศษตัวอื่น ๆ การเปลี่ยนแปลงค่าของรีจีสเตอร์โปรแกรมเคาน์เตอร์ ขึ้นอยู่กับผลของการกระทำคำสั่งภายในหน่วยความจำโปรแกรมที่ผู้เขียนโปรแกรมกำหนด

รีจิสเตอร์โปรแกรมเคาน์เตอร์มีความสำคัญอย่างยิ่ง ในการตรวจสอบการทำงานของโปรแกรมว่าดำเนินไปตามลำดับขั้นตอนตามที่กำหนดไว้หรือไม่

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
CY	AC	F0	RS1	RS2	OV	-	P

CY : แฟลกทด (Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิก แล้วค่าของแอกคิวมูเลเตอร์เกิน 255 (ฐานสิบ) หรือ FFH

AC : แฟลกทดเสริม (Auxiliary Carry flag) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์แล้วทำให้เกิดการหารทดข้ามจากบิต 3 มายังบิต 4

F0: แฟลกใช้งานทั่วไป เมื่อผู้เขียนโปรแกรมกำหนดค่าที่บิตนี้แล้ว ไม่ว่าจะทำคำสั่งใด ๆ ที่บิตนี้จะไม่มีการเปลี่ยนแปลง

RS1: บิตเลือกรีจิสเตอร์แบงก์ (Register Select1) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

RS2: บิตเลือกรีจิสเตอร์แบงก์ (Register Select0) ใช้งานร่วมกับบิต RS0 เพื่อเลือกแบงก์ของรีจิสเตอร์ R0-R7

OV : บิตเกิน (Overflow) เป็น "1" เมื่อมีการกระทำคำสั่งทางคณิตศาสตร์และลอจิก แล้วทำให้เกิดการทดข้ามบิต 6 มายังบิต 7 ของแอกคิวมูเลเตอร์หรือแอกคิวมูเลเตอร์ มีค่าเกิน 127 (ฐานสิบ) นอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นยังมีการใช้แสดงเป็นค่าลบด้วย

- : บิตนี้ผู้ใช้งานสามารถกำหนดได้อย่างอิสระ

P: บิตพาราตี้ (Parity) ใช้ในการตรวจสอบจำนวนค่า "1" ภายในแอกคิวมูเลเตอร์

ภายในแอกคิวมูเลเตอร์มีจำนวนบิตที่เป็น "1" รวมกันเป็นเลขคู่ บิตนี้จะเป็น "0" ถ้ารวมกันเป็นเลขคี่ บิตนี้จะเป็น "1"

ตารางที่ 2.3 การเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์ R0-R7

RS1	RS2	แบงก์ของรีจิสเตอร์	ช่วงแอดเดรส
0	0	แบงก์0	00H-07H
0	1	แบงก์1	08H-0FH
1	0	แบงก์2	10H-17H
1	1	แบงก์3	18H-0FH

2.1.5.7 สแต็กพอยน์เตอร์ (Stack Pointer: SP)

รีจิสเตอร์ตัวชี้สแต็ก มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง 81H ใช้ในการเก็บค่าตำแหน่งของตัวชี้สแต็กซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียู มีการกระโดดไปทำงานที่โปรแกรมย่อยหรือกระโดดโปรแกรมย่อยกลับมายังโปรแกรมหลัก เมื่อมีการรีเซตเกิดขึ้น (รีเซต: การกระทำที่ส่งผลให้ซีพียูต้องเริ่มการทำงานใหม่ตั้งแต่ต้น) ค่าของรีจิสเตอร์สแต็กพอยน์เตอร์ จะเท่ากับ 07H ดังนั้นที่แอดเดรสของพื้นที่ที่สำรองไว้ทำหน้าที่เป็นสแต็กจะเท่ากับ 08H

2.1.5.8 รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer: DPTR)

มีขนาด 16 บิต โดยแบ่งเป็นรีจีสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจีสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H สำหรับรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ และ 83H สำหรับรีจีสเตอร์ชี้ข้อมูลไบต์สูงรีจีสเตอร์ชี้ข้อมูลนี้ใช้เก็บค่าแอดเดรสของหน่วยความจำภายนอก ที่ไมโครคอนโทรลเลอร์ต้องการติดต่อ

2.1.5.9 รีจิสเตอร์พอร์ต (Port Register)

เป็นรีจิสเตอร์ขนาด 8 บิตที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มี 4 ตัวคือรีจิสเตอร์พอร์ต 0 มีแอดเดรสอยู่ที่ 80H รีจิสเตอร์พอร์ต 1 มีแอดเดรสอยู่ที่ 90 H รีจิสเตอร์พอร์ต 2 มีแอดเดรสอยู่ที่ A0H รีจิสเตอร์พอร์ต 3 มีแอดเดรสอยู่ที่ B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงในระดับบิตได้ เมื่อต้องการอ่านหรือเขียนข้อมูลออกไปยังพอร์ตของไมโครคอนโทรลเลอร์ จะต้องทำการผ่านรีจิสเตอร์นี้ทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.10 รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer: SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิตมีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ส่งออกหรือรับเข้าของวงจรสื่อสารอนุกรมที่มีอยู่ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยภายในรีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรมนี้จะแบ่งออกเป็น 2 ส่วน คือรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล (Transmit Buffer Register) และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (Receive Buffer Register) เมื่อมีการเขียนข้อมูลมายังรีจิสเตอร์ DPL ข้อมูลนั้นจะถูกส่งออกไปยังบัฟเฟอร์สำหรับส่งข้อมูลเพื่อส่งออกจากทางขา TxD หรือพอร์ต 3.1 ในกรณีที่มีการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม ข้อมูลที่จะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งไปยังไมโครคอนโทรลเลอร์ต่อไป สำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นผ่านทางขา RxD หรือพอร์ต 3.0 ของไมโครคอนโทรลเลอร์ MCS-51

2.1.5.11 รีจิสเตอร์ไทเมอร์ (Timer Register)

เป็นรีจีสเตอร์ขนาด 16 บิต แบ่งเป็นไบต์สูงและไบต์ต่ำเช่นเดียวกับรีจิสเตอร์ชี้ข้อมูล ซึ่งเป็นรีจิสเตอร์ไทเมอร์ใช้ในการเก็บค่าตัวนับหรือเคาน์เตอร์ (Counter) ภายในไมโครคอนโทรลเลอร์เพื่อที่จะใช้ในการสร้างเวลาจับเวลา หรือนับจำนวนสัญญาณนาฬิกาภายในบางทีเรียกรีจิสเตอร์ตัวนี้ว่า รีจิสเตอร์ไทเมอร์/เคาน์เตอร์

ในไมโครคอนโทรลเลอร์เบอร์ AT89C51 มีรีจิสเตอร์ไทเมอร์/เคาน์เตอร์ 2 ตัว แบ่งเป็นไทเมอร์ 0 และไทเมอร์ 1 ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทเมอร์ไบต์ต่ำ (TL) และรีจิสเตอร์ไทเมอร์ไบต์สูง (TH) เหมือนกัน โดยในรีจิสเตอร์ไทเมอร์ไบต์ต่ำ 0 มีแอดเดรสอยู่ที่ 8AH รีจิสเตอร์ไทเมอร์ไบต์สูง 0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ไทเมอร์ไบต์ต่ำ 1 และไทเมอร์ไบต์สูง 1 มีแอดเดรสอยู่ที่ 8CH และ 8DH สำหรับในเบอร์ AT89C52 และในอนุกรม AT89Sxx จะมีรีจิสเตอร์ไทเมอร์/เคาน์เตอร์ถึง 3 ตัว โดยมีรีจิสเตอร์ไทเมอร์ไบต์ต่ำ 2 และไทเมอร์ไบต์สูง 2 มีแอดเดรสอยู่ที่ 0CCH และ 0CDH เพิ่มเติมเข้ามา

2.1.5.12 รีจิสเตอร์แคปเจอร์ (Capture Register)

รีจิสเตอร์ขนาด 16 บิต มีเฉพาะในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89Sxx เท่านั้นเนื่องจากใช้ร่วมกับไทเมอร์/เคาน์เตอร์ 2 โดยรีจิสเตอร์แคปเจอร์นี้มีชื่อเรียกย่อว่า รีจิสเตอร์ RCAP2 ซึ่งแบ่งออกเป็นไบต์ต่ำคือ RCAP2 มีแอดเดรสอยู่ที่ 0CAH และไบต์สูงคือ RCAP2H มีแอดเดรสอยู่ที่ 0CBH

รีจิสเตอร์แคปเจอร์นี้จะถูกใช้งานเมื่อกำหนดให้ไทเมอร์2 ทำงานในโหมดแคปเจอร์ซึ่งเป็นโหมดที่กำหนดให้ไมโครคอนโทรลเลอร์ ทำการตรวจจับการเปลี่ยนแปลงสถานะลอจิกที่ขา T2EX ทั้งนี้เพื่อประโยชน์ในการวัดคาบเวลา ความถี่ และการเปลี่ยนแปลงสัญญาณที่ขา T2EX

2.1.5.13 รีจิสเตอร์ควบคุม (Control Register)

รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อ สารอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

เอกสารนี้มีลิขสิทธิ์ ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของวงจรสื่อสารอนุกรม ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ TCON กับ T2CON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2CON ใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ AT89C52 และในอนุกรม AT89Sxx

รีจิสเตอร์ TMOD และ T2MOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทเมอร์/เคาน์เตอร์ ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2MOD ใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

รีจิสเตอร์ไออี (IE) และไอพี (IP) เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ (Interrupt: การขัดจังหวะการทำงานปกติของซีพียู) โดยไออี เป็นรีจิสเตอร์สำหรับเอ็นเอเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ไอพีจะเป็นรีจิสเตอร์สำหรับกำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่าจะให้ซีพียูตอบสนอง การเกิดอินเตอร์รัปต์ในลักษณะก่อนหรือหลัง

2.2 ทฤษฎีเบื้องต้นสแกนคีย์

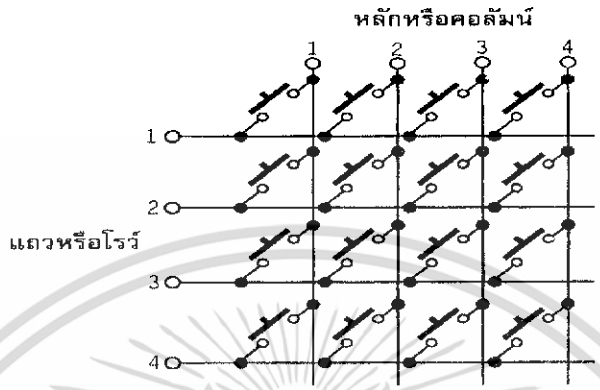
การอ่านค่าหรือรับค่าการกดสวิตช์ซึ่งก็เป็นอีกลักษณะหนึ่ง ของการใช้งานของระบบไมโครคอนโทรลเลอร์ต้องสามารถรองรับและเชื่อมต่อใช้งานร่วมกับวงจรของสวิตช์มี 2 ลักษณะใหญ่ ๆ คือ ต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง เมื่อสวิตช์ตัวใดต่อวงจรสามารถอ่านค่าโดยตรง วงจรในลักษณะนี้ไม่มีความซับซ้อน สามารถอ่านค่าของสวิตช์ได้ง่ายและรวดเร็ว มีข้อเสียคือ ถ้าจำนวนของสวิตช์มีมาก ๆ จำนวนของสายข้อมูลก็จะมีมากตาม ทำให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง วงจรของสวิตช์อีกลักษณะหนึ่งคือ

การต่อวงจรแบบเมตริกซ์ (Matrix Switch) ดังในรูปที่ 2.14 สวิตช์จะถูกต่อกันในแนวแกนตั้งและแกนนอนจะเรียกแนวตั้งว่า หลักหรือคอลัมน์ (Column) ในขณะที่แนวนอนจะเรียกว่า แถวหรือโรว์ (Row) ดังนั้นค่าของสวิตช์จะประกอบด้วยตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำให้ได้มาซึ่งค่าของสวิตช์มีขั้นตอนซับซ้อนพอสมควร แต่วงจรของสวิตช์แบบนี้มีข้อดีคือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวกเพียงเพิ่มจำนวนสวิตช์โดยแก้ไขซอฟต์แวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบเมตริกซ์เป็นที่นิยมใช้มากในระบบควบคุมอัตโนมัติ หรือกึ่งอัตโนมัติที่จำนวนสวิตช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิตช์แบบเมตริกซ์นี้ว่า คีย์แพด (Keypad)

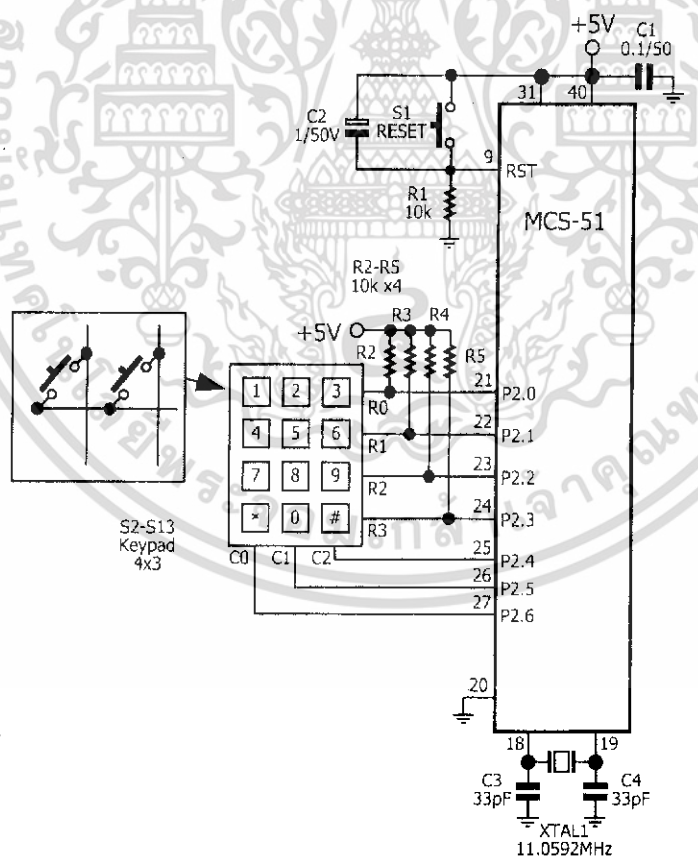
ตัวอย่างวงจรในรูปที่ 2.15 จะใช้พอร์ต 2 ไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้นคือ สายคอลัมน์ 3 เส้นคือ C0-C2 และสายทางโรว์หรือแถวอีก 4 เส้นคือ R0-R3 โดยเฉพาะที่ขาพอร์ต P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัปไว้ เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ โดยที่ไมโครคอนโทรลเลอร์จะถูกทำการส่งข้อมูล "0" ไปยัง P2.6 , P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพดไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P2.0-P2.3 เข้ามาด้วย หากไม่มีการกดค่าของ P2.0-P2.3 ก็จะเป็น "1" ทั้งหมด ถ้าหากมีการกดคีย์ค่าของ P2.0-P2.3 จะไม่เป็น 1111 อีกต่อไปเป็นการแจ้งให้ทราบว่ามีการกดคีย์แพดขึ้น จากนั้นไมโครคอนโทรลเลอร์ก็จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งสิ่งที่จะได้มาอย่างแรกคือค่าตำแหน่งของคีย์นั้น จากนั้นก็จะส่งค่าตำแหน่งนั้นไปเปิดตารางข้อมูลเพื่อจะได้หมายเลขของคีย์ที่กดอย่างแท้จริง ยกตัวอย่างจากวงจรในรูปที่ 2.15 หากคีย์ 0 ถูกกด ไมโครคอนโทรลเลอร์จะได้ค่าตำแหน่งของคีย์ 0 เป็น 0BH จากนี้ไปเปิดตารางก็จะได้ข้อมูลเป็น 0 หมายถึงคีย์ 0 ซึ่งกระบวนการหลังจากนี้เป็นหน้าที่ของซอฟต์แวร์



รูปที่ 2.14 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด



รูปที่ 2.15 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51 การเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การใช้งานด็อตแมตริกซ์แอลซีดีโมดูล (Dot Matrix LCD Module)

ปัจจุบันแอลซีดีโมดูลเป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องใช้ต่าง ๆ ทั้งนี้เนื่องจากมีความเหมาะสมในหลาย ๆ ด้าน เช่น การใช้กระแสต่ำ สามารถแสดงผลเป็นตัวอักษรและตัวเลข หรือแสดงเป็นกราฟิก (เฉพาะรุ่น) และในขณะนี้ผู้ผลิตแอลซีดีโมดูลจะผลิตแอลซีดีรุ่นที่เป็นโมดูลที่มีตัวแอลซีดีและวงจรควบคุมมาให้พร้อมออกมา ทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครคอน - โทรลเลอร์ได้ง่ายและสะดวกสำหรับการเขียนโปรแกรม

แอลซีดีโมดูลมีอยู่หลายรุ่นและคุณสมบัติแตกต่างกันออกไป แบ่งได้เป็น 2 กลุ่มคือแบบ ด็อตแมตริกซ์ และ กราฟิก (Graphic) โดยแบบ ด็อตแมตริกซ์ จะแสดงผลเป็นตัวอักษรขนาด 5×8 ด็อต (Dot) และมีจำนวนตัวอักษรและบรรทัดแตกต่างกันออกไปในแต่ละรุ่น ส่วนแบบกราฟิกจะแสดงผลในแบบบิตแมพ (Bit Map) คือสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ แนวทางในการใช้งานของทั้ง 2 แบบต่างก็มีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะเป็นแบบด็อตแมตริกซ์มากกว่า เนื่องจากราคาถูกกว่าและเพียงพอต่องานส่วนใหญ่

คุณสมบัติของ ด็อตแมตริกซ์แอลซีดีโมดูล และสามารถสรุปได้เป็นข้อ ๆ ดังนี้

- ตัวอักษรแสดงด้วยด็อตแมตริกซ์ ขนาด 5×8 ด็อต
- สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะคือแบบ เมโมรีแมพ และแบบผ่านพอร์ตซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา (ขั้วต่อ 16 ขา)
- การใช้งานสะดวกกับระบบไมโครคอนโทรลเลอร์ เพียงแค่ส่งข้อมูลให้กับแอลซีดีโมดูลเท่านั้นข้อความก็จะปรากฏบนแผงแสดงและจะค้างไว้ตลอดทำให้ไม่ต้องเสียเวลาของระบบ
- มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น Clear Display, Home Cursor on, Off Cursor, Blank Characteristic ฯลฯ
- สามารถแสดงผลตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัวและสัญลักษณ์พิเศษอีก 32 ตัว
- กินกระแสน้อยและมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์

2.3.1 การต่อเข้ากับระบบไมโครคอนโทรลเลอร์

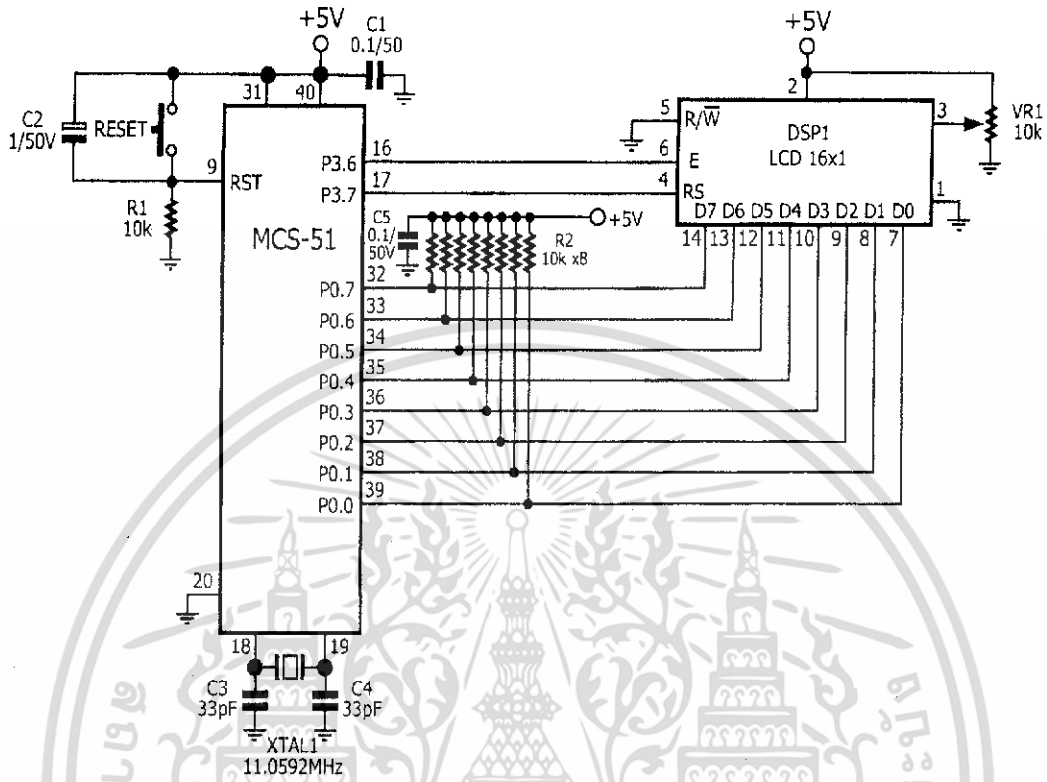
แอลซีดีโมดูลจะต่อเข้ากับระบบไมโครคอนโทรลเลอร์ดังแสดงในรูปที่ 2.16 ได้ 2 ลักษณะดังได้กล่าวไปแล้วซึ่งทั้ง 2 แบบนี้จะมีข้อดีและข้อเสียแตกต่างกันออกไป โดยแต่ละแบบมีหลักการดังนี้

2.3.1.1 การต่อแบบเมโมรีแมพ (Memory Map)

- สามารถต่อเข้ากับชิปเบอร์ทั่ว ๆ ไปได้เช่น 8051 หรือ Z80 โดยจะทำให้ระบบที่จะใช้ตัวไมโครคอนโทรลเลอร์สามารถมองเห็นแอลซีดีโมดูลในลักษณะของเมโมรีได้ทันที
- ผู้ใช้สามารถเขียนและอ่านข้อมูลจากแอลซีดีได้ ซึ่งจะทำให้มองเสมือนว่าเป็นเมโมรีบัฟเฟอร์ (Memory Buffer) ไปในตัว
- เนื่องจากสามารถอ่านข้อมูลกลับได้จึงทำให้สามารถตรวจสอบแฟลก ความพร้อมในขณะที่แอลซีดีโมดูลกำลังทำงานได้
- ใช้ได้กับบอร์ดที่มีแอลซีดีบัสมาให้พร้อมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่งและต้องมีการดีโคดละเอียดพอสมควร
- การจัดขาสัญญาณจะต้องเป็นไปตามแบบของชิป



รูปที่ 2.16 การต่อแอลซีดีโมดูลเข้ากับไมโครคอนโทรลเลอร์

2.3.1.2 การต่อแบบไอโอพอร์ต (I/O Port)

- สามารถต่อเข้ากับไอโอพอร์ตใด ๆ ก็ได้โดยใช้สายสัญญาณจำนวน 11 เส้นและใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมา ให้ตรงกับข้อกำหนดของแอลซีดีโมดูล
- ผู้ใช้เขียนข้อมูลให้แอลซีดีโมดูลได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนดเมโมรีส่วนหนึ่งให้เสมือนบัฟเฟอร์ให้กับแอลซีดีโมดูล
- เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบเอง เพื่อรอให้แอลซีดีโมดูลกระทำขบวนการต่าง ๆ
- ใช้ได้กับบอร์ดทั่ว ๆ ไปที่มีพอร์ต
- ไม่เปลืองส่วนของเมโมรีในการใช้งาน

2.3.1.3 ชุดคำสั่งควบคุมและการแสดงข้อความ

ขาสัญญาณที่มีไว้สำหรับกำหนดความเข้มของตัวอักษร (Vee) โดยถ้าต่อกับกราวด์จะมีความเข้มสูงสุดแต่ถ้ากับแรงดันจะมีความเข้มต่ำสุด โดยปกติแอลซีดีโมดูลรุ่นธรรมดาอาจจะต่อกันไว้ให้เลยไม่ต้องใส่ตัวปรับโวลต์ให้สิ้นเปลืองแต่ถ้ารุ่น SNT (มุมมองกว้าง) ให้ใช้ R2K ต่อลงกราวด์เพื่อให้ความเข้มที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหมาะสมการเขียนหรืออ่านข้อมูลกับแอลซีดีโมดูล ก็คือ การกำหนดคุณสมบัติต่าง ๆ ในการใช้งานของแอลซีดีตามชุดคำสั่งควบคุม และรวมไปถึงการเขียนข้อมูลที่เป็นข้อความเพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดดังตารางที่2.5 และสามารถดูขาสัญญาณของแอลซีดีโมดูลดังตารางที่2.4

ตารางที่ 2.4 ขาสัญญาณของแอลซีดีโมดูล

PIN	SYMBOL	LEVEL	FUNCTION
1	Vss	-	0 V
2	Vcc	-	+5 V Power Supply
3	Vee	-	+V For Liquid Crystal Drive
4	Rs	H/L	Register Select H : data Input L : Instruction
5	R/W	H/L	Input
6	E	H/L	H : Data Input L : Data Write Enable Signal (L->H)
7	DB 0	H/L	Data Bus Bit 0
8	DB 1	H/L	Data Bus Bit 1
9	DB 2	H/L	Data Bus Bit 2
10	DB 3	H/L	Data Bus Bit 3
11	DB 4	H/L	Data Bus Bit 4
12	DB 5	H/L	Data Bus Bit 5
13	DB 6	H/L	Data Bus Bit 6
14	DB 7	H/L	Data Bus Bit 7

2.3.1.4 ความเข้าใจพื้นฐาน

- การเขียนข้อมูลให้กับแอลซีดีโมดูลจะแบ่งเป็น 2 ลักษณะ คือ Instruction และ Data โดยที่จะกำหนดด้วยขาของสัญญาณ RS คือถ้า RS=0 จะหมายถึงส่งสัญญาณควบคุม (Instruction) หรืออ่านค่าแฟลกสภาพการทำงานของแอลซีดีโมดูลและถ้า RS=1 จะหมายถึงการเขียนหรืออ่านข้อมูล (Data) กับแอลซีโมดูล

- หลักการในการเขียนข้อมูลให้แอลซีดีโมดูล นี้คือเมื่อมีการเขียนข้อมูลไปแล้วตัวแอลซีดีโมดูลจะใช้เวลาในการทำงานชั่วขณะหนึ่ง ซึ่งระบบไมโครคอนโทรลเลอร์สามารถตรวจสอบได้จากบิวซี่แฟลก (Busy Flag: BF) และถ้าเรียบร้อยแล้วจึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่มีการต่อวงจรเป็นแบบไอโอพอร์ต คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครคอนโทรลเลอร์ก็จะต้องใช้วิธีหน่วงเวลาแทน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเขียนข้อมูลให้กับแอลซีดีโมดูลนี้สามารถทำได้ทั้งแบบ 8 บิต และ 4 บิต โดยในกรณี 4 บิต จะใช้สายสัญญาณข้อมูลเพียง 4 เส้นคือ DB4-DB7 (ใช้สำหรับไมโครคอนโทรลเลอร์แบบ 4 บิต หรือเพื่อเป็นการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 บิต เพียงแต่ให้เขียน 2 ครั้งคือ DB4-DB7 ก่อนแล้วค่อยตามด้วย DB0-DB3 และจะต้องกำหนดคุณสมบัติตามค่าดีแอล (DL) ในคำสั่งเซตฟังก์ชั่น (Function Set) ด้วย

- ดีดีแรม (DDRAM: Display Data Ram) คือหน่วยความจำภายในตัวแอลซีดีโมดูลเป็นบัฟเฟอร์ของข้อมูลโดยถ้าเขียนรหัสแอสกี (ASCII) ใดๆ ลงไปในหน่วยความจำนี้ก็จะปรากฏเป็นตัวอักษรที่แผงแสดงทันที

- ซีจีแรม (CGRAM: Character Generator Ram) คือหน่วยความจำภายในตัว แอลซีดีโมดูล สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สร้างขึ้นได้เอง (8 ตัว) โดยจะอ้างแอดเดรสได้ทั้งหมด 64 ไบต์ คือ 8 ตัวอักษรคูณกับ 8 แถว

ตารางที่ 2.5 ชุดคำสั่งควบคุมและการแสดงข้อความ

INSTRUCTION	RS	R/W	DATA BIT								EXETIME (μS)
			7	6	5	4	3	2	1	0	
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	1	SC	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS							40
BUSY ADD. READ	0	1	BF	ADDRESS							0
CGRAM , DDRAM WR	1	0	WRITE DATA								40
CGRAM , DDRAM RD	1	1	READ DATA								0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 รายละเอียดของแต่ละคำสั่ง

CLEAR DISPLAY

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

สำหรับการเคลียร์หน้าจอ (Clear Display) โดยจะทำการเขียนตัวอักษรสเปซ (Space) ลงไปในดีดีแรม (DDRAM) ทั้งหมดและกำหนดค่าดีดีแรมแอดเดรส (DDRAM Address) ให้เป็น 0 พร้อมทั้งเคอเซอร์ (Cursor) จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

CURSOR AT HOME

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

สำหรับการกำหนดค่าดีดีแรมแอดเดรส ให้เป็น 0 พร้อมทั้งเคอเซอร์ จะไปที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลในดีดีแรม ไม่มีการเปลี่ยนแปลง

ENTRY MODE SET

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

I/D – 0 กำหนดทิศทางของเคอเซอร์ และดีดีแรมให้เป็นแบบ Decrement

I/D – 1 กำหนดทิศทางของเคอเซอร์ และดีดีแรม ให้เป็นแบบ Increment

S – 0 เมื่อเขียนข้อมูลแล้ว ตัวเคอเซอร์ จะถูกเลื่อนไปตามทิศทางของค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัวเคอเซอร์ จะอยู่กับที่ และตัวอักษรจะถูกดันไปตามทิศทางของค่า I/D

กำหนดค่า I/D และ S นี้ให้กำหนดก่อนการเขียนข้อมูลในดีดีแรมและเมื่อกำหนดแล้วจะไม่ต้องทำคำสั่งนี้อีก

DISPLAY ON/OFF

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

D = 0 กำหนดให้ Off Display

D = 1 กำหนดให้ ON Display

C – 0 กำหนดให้ Off Cursor

C = 1 กำหนดให้ ON Cursor โดยเคอเซอร์จะเป็นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่งเคอเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่งเคอเซอร์ (กระพริบเป็นรูป [])

DISPLAY SHIFT

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

S/C = 0 กำหนดให้เลื่อนเคอเซอร์ ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความบนแผงแสดงตามทิศทาง R/L ไป 1 คอลัมน์ (เลื่อนทุกบรรทัด)

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย R/L = 1 กำหนดให้มีทิศทางไปทางขวา

FUNCTION SET

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

D/L = 0 กำหนดให้การติดต่อกับแอลซีดีโมดูล เป็นแบบ 4 บิต

D/L = 1 กำหนดให้การติดต่อกับแอลซีดีโมดูล เป็นแบบ 8 บิต จะสังเกตว่าการกำหนดค่า DL นี้สามารถกระทำได้ที่ DB4-DB7 ซึ่งถ้ากำหนดให้เป็นแบบ 4 บิต ตั้งแต่แรก หลังจากจ่ายไฟเลี้ยงก็จะทำให้แอลซีดีโมดูล มีการรับข้อมูลแบบ 4 บิต ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 Duty และ 1/11 Duty

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 Duty

F = 0 กำหนดให้มีตัวอักษรเป็นแบบ 5×7 Dots

F = 1 กำหนดให้มีตัวอักษรเป็นแบบ 5×10 Dots (กรณีที่แอลอีดีโมดูล เป็นแบบ 5×7 ก็ไม่มีผลอะไร)

SET CGRAM ADDRESS

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

สำหรับการกำหนดแอดเดรสของซีจีแรม (CGRAM) เมื่อได้ทำการกำหนดไว้แล้วการอ่านและการเขียนข้อมูลที่ต่อจากนี้จะเป็นไปตามแอดเดรสที่กำหนดทันที

SET DDRAM ADDRESS

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

สำหรับการกำหนดแอดเดรส ของดีดีแรมเมื่อได้ทำการกำหนดไว้แล้วการอ่าน และเขียนข้อมูลที่ต่อจากนี้จะเป็นไปตามแอดเดรสที่กำหนดทันที ตำแหน่งของแอดเดรสในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BUSY FLAG AND ADDRESS READ

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

สำหรับการอ่านค่า BF (Busy Flag) ซึ่งบอกถึงความพร้อมของแอลซีดีโมดูล ในการรับข้อมูล ถ้า BF = 0 หมายความว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายความว่ายังไม่พร้อม นอกจากนี้ยังเป็นการอ่านค่าแอดเดรสของ ซีจีแรม หรือ ดีดีแรม ด้วย

2.3.3 การอ่านและการเขียนข้อมูลกับ ดีดีแรม/ซีจีแรม

WRITE DATA TO DDGRAM OR CGRAM

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

สำหรับการเขียนข้อมูลลงหน่วยความจำ ดีดีแรมหรือซีจีแรม โดยเมื่อทำการเขียนแล้วแอสเดรสจะถูกเพิ่มหรือลดลงโดยอัตโนมัติตามที่กำหนดจากค่า I/D ในคำสั่ง Entry Mode Set และการเขียนจะเป็นดีดีแรมหรือซีจีแรม ทั้งนี้ก็ขึ้นกับว่าก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ใด

READ DATA FROM DDGRAM OR CGRAM

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

สำหรับการอ่านข้อมูลจากหน่วยความจำ ดีดีแรม หรือ จีแรม โดยเมื่อทำการอ่านแล้วแอดเดรส จะถูกเพิ่มหรือลดลงโดยอัตโนมัติตามที่กำหนดจากค่าไอดีในคำสั่ง Entry Mode Set และการอ่านจะเป็น ซีจีแรม หรือ ดีดีแรม ทั้งนี้ก็ขึ้นกับว่าก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ใด

2.3.4 การเขียนโปรแกรมควบคุม

เมื่อจ่ายไฟเลี้ยงให้กับแอลซีดีโมดูลครั้งแรก ภายในจะมีการรีเซตระบบโดยอัตโนมัติซึ่งจะใช้เวลา 10 มิลลิวินาที หลังจากแรงไฟขึ้นถึง 4.5 โวลต์ แล้วทั้งนี้ระบบ รีเซต ดังกล่าวจะกระทำสิ่งต่าง ๆ ต่อไปดังนี้

- ทำการเคลียร์จอภาพทั้งหมด (Clear Display)
- กำหนดคุณสมบัติด้วยคำสั่งฟังก์ชันเซต (Function Set) คือ N = 1(แสดงข้อมูล 1 บรรทัด) F = 0 (กำหนดตัวอักษรแบบ 5×7 ด็อต) , DL = 1 (ติดต่อกับระบบไมโครคอนโทรลเลอร์ในแบบ 8 บิต)
- กำหนดคุณสมบัติด้วยคำสั่ง Entry Mode Set คือ I/D = 1 (Increment) , S = 0 (NO Shift)

การใช้งาน แอลซีดีโมดูล ต้องรอให้ขบวนการรีเซต ภายในทำงานเรียบร้อยก่อน ซึ่งจะตรวจสอบได้ด้วยบิวซี่แฟลกหรืออาจจะใช้การหน่วงเวลาก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน แอลซีดีโมดูล จะต้องเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่างๆ รวมทั้งการอ่านหรือเขียนข้อมูลนั้น จะถูกกำหนดด้วยขาสัญญาณทั้งหมดที่มีอยู่ปกติโปรแกรมจะต้องกำหนดคุณสมบัติต่าง ๆ ที่ต้องไว้ที่ส่วนต้น จากนั้นจะเป็นการอ่านและเขียนข้อมูลลงใน ดีดีแรม (DDRAM) ซึ่งก็คือข้อความที่จะให้แสดงผลนั่นเอง

2.4 ทฤษฎีรีโมทคอนโทรล

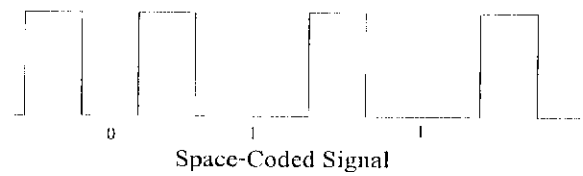
รีโมทคอนโทรลประเภทที่ใช้แสงอินฟราเรดทุกชนิด จะมีการส่งข้อมูลหรือสัญญาณพัลส์ผ่านตัวส่งอินฟราเรด (LED IR) โดยจะใช้ค่าความถี่ในช่วง 30-40 กิโลเฮิรตซ์ สาเหตุที่ต้องใช้ค่าความถี่สูงเพื่อเป็นการป้องกันแสงจากแหล่งกำเนิดแสงอื่น ๆ มารบกวน โดยสัญญาณที่ส่งมาจากรีโมทอินฟราเรดจะมีรูปแบบเหมือนเลขฐานสองเนื่องจากผู้ผลิตรีโมทคอนโทรลมีอยู่หลายบริษัท ซึ่งจะทำให้การเข้ารหัสของรีโมทแต่ละยี่ห้อไม่เหมือนกัน แต่จะอยู่ในพื้นฐานเดียวกันคือ จะใช้รหัสเลขฐานสองและความกว้างความแคบของพัลส์ในช่วง “High” และในช่วง “Low” เป็นตัวกำหนดการทำงาน

2.4.1 ความแตกต่างของการเข้ารหัส

2.4.1.1 Pulse-Width-Code Signals (ใช้ความกว้างของพัลส์กำหนดรหัส) จะใช้หลักการกำหนดความกว้างของพัลส์แทนลอจิก “0” หรือ “1”โดยทั่วไปจะให้ลอจิก “0” มีความกว้างของพัลส์ประมาณ 550 ไมโครวินาที และให้ลอจิก “1” มีความกว้างของพัลส์ประมาณ 2,200 ไมโครวินาที ดังแสดงในรูปที่ 2.17

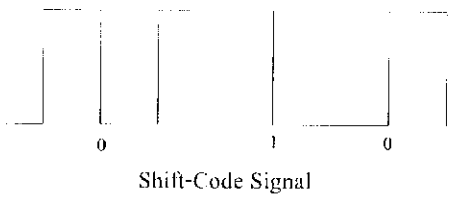
2.4.1.2 Space-Code Signal (ใช้ช่องว่างของพัลส์กำหนดรหัส) จะใช้หลักการกำหนดช่องว่างของสัญญาณแทนลอจิก “0” หรือ “1” โดยทั่วไปแล้วจะให้ลอจิก “0” มีความกว้างของช่องว่างประมาณ 550 ไมโครวินาทีและให้ลอจิก“1”มีความกว้างของช่องว่างประมาณ 1,650 ไมโครวินาที ดังแสดงในรูปที่ 2.18

รูปที่ 2.17 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ความกว้างของพัลส์



รูปที่ 2.18 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ช่องว่างของพัลส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 รูปแบบของสัญญาณพัลส์ที่มีการเข้ารหัสโดยใช้ช่องการเลื่อนของพัลส์

2.4.1.3 Shift-Code Signal (ใช้ช่องของการเลื่อนของพัลส์กำหนดรหัส) จาก 2 แบบข้างต้นนำไปสู่แบบที่ 3 จะใช้การกำหนดทั้งความกว้างของพัลส์และความกว้างของช่องว่างของพัลส์เป็นตัวกำหนดลอจิก "0" หรือ "1" โดยถ้าความกว้างของพัลส์แคบ (ประมาณ 550 ไมโครวินาที) และความกว้างของพัลส์ยาว (ประมาณ 1,100 ไมโครวินาที) จะสื่อความหมายว่าเป็นลอจิก "1" แต่ถ้าความกว้างของช่องว่างยาว และความกว้างของพัลส์แคบจะแทนความหมายว่าเป็นลอจิก "0" ดังแสดงในรูปที่ 2.19

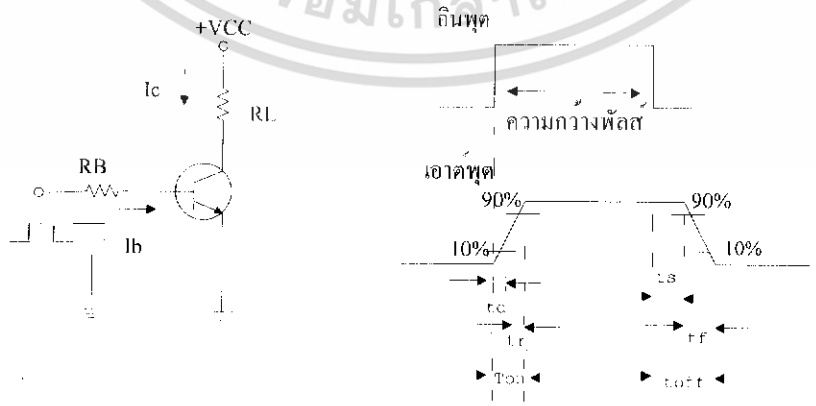
2.5 ทฤษฎีทรานซิสเตอร์

ทรานซิสเตอร์เป็นอุปกรณ์สารกึ่งตัวนำที่มีการใช้งานอย่างกว้างขวางในวงจรอิเล็กทรอนิกส์ เช่น การขยายสัญญาณ การควบคุมแรงดันให้คงที่ การขยายความถี่สูง เป็นต้น แต่ถ้าต้องการความถี่สูง , ทรานซิสเตอร์กำลัง และสวิตช์ทรานซิสเตอร์

ในการใช้ทรานซิสเตอร์ทำหน้าที่เป็นสวิตช์ทำได้โดยการจ่ายไบแอสให้แก่ทรานซิสเตอร์ ซึ่งมีการทำงานอยู่ในช่วงสถานะอิ่มตัว (Saturation State) และหยุดทำงานอยู่ในช่วงสถานะตัด (Cut off State)

2.5.1 ช่วงเวลาการสวิตช์ในทางปฏิบัติ

ช่วงเวลาการสวิตช์เป็นสิ่งสำคัญมากโดยเฉพาะอย่างยิ่งเพื่อใช้กับความถี่สูงซึ่งช่วงเวลาการสวิตช์นี้จะเป็นตัวชี้บอกถึงคุณสมบัติของอุปกรณ์ที่ทำหน้าที่เป็นตัวสวิตช์ว่าเร็วหรือช้า เมื่อพิจารณารูปสัญญาณของการสวิตช์จะประกอบด้วย 2 ส่วน คือ ช่วงเวลานำกระแส และช่วงไม่นำกระแสดังแสดงในรูปที่ 2.20



รูปที่ 2.20 แสดงความสัมพันธ์ระหว่างสัญญาณทางอินพุตและเอาต์พุตของวงจรสวิตช์ทรานซิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงเวลานำกระแส (On Time) หมายถึงช่วงเวลาที่ใช้เพื่อให้แรงดันที่ขาคอลเลกเตอร์ของทรานซิสเตอร์เปลี่ยนแปลงค่าขึ้นไปที่ 90% ของแหล่งจ่ายแรงดัน ซึ่งช่วงเวลานำกระแส (t_{on}) จะประกอบด้วยช่วงเวลาย่อย 2 ส่วน คือ

ช่วงเวลาประวิง (Delay Time) คือ ช่วงเวลาที่เป็นระยะเวลาห่างจากจุดเริ่มต้นของการสวิตช์ทางอินพุตจนถึงแรงดันเอาต์พุตเปลี่ยนแปลงเป็น 10 % ของแหล่งจ่าย ใช้ตัวอักษรตัวย่อคือ "t_d"

ช่วงเวลาขึ้น (Rise Time) คือช่วงเวลาที่แรงดันเอาต์พุตเปลี่ยนแปลงจาก 10% ถึง 90% ของแหล่งจ่ายแรงดัน ใช้อักษรตัวย่อคือ "t_r" เพราะฉะนั้น ช่วงเวลานำกระแส $(t_{on}) = t_d + t_r$

ช่วงเวลาไม่นำกระแส (Off Time) คือช่วงเวลาที่ใช้เพื่อให้ค่าของแรงดันเอาต์พุตลดลงจาก 90% ของแหล่งจ่ายแรงดัน มาที่ 0 โวลต์ ซึ่งเวลาไม่นำกระแส (t_{off}) จะประกอบด้วยช่วงเวลาย่อย 3 ส่วน คือ

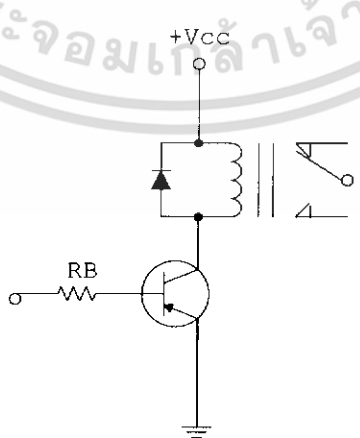
ช่วงเวลาสะสม (Storage Time) คือขณะเมื่อสัญญาณทางอินพุตอยู่ในช่วงไม่นำกระแสแล้วแต่สัญญาณทางเอาต์พุตยังคงอยู่ในช่วงนำกระแส ซึ่งช่วงเวลาสะสมนี้จะมีระดับแรงดันเอาต์พุตลดลงมาที่ 90% ของแหล่งจ่ายแรงดัน ใช้ตัวอักษรตัวย่อ คือ "t_s"

ช่วงเวลาลง (Fall Time) คือ ช่วงเวลาที่แรงดันเอาต์พุตมีระดับลดลงจาก 90% มาที่ 10% ของแหล่งจ่ายแรงดัน ซึ่งต่อจากช่วงเวลาสะสม ใช้อักษรย่อของช่วงเวลาลง คือ "t_f"

ช่วงเวลาประวิง (Delay Time) คือ ช่วงเวลาที่ต่อจากช่วงเวลาลงจากระดับแรงดันเอาต์พุตที่ 10% ของแหล่งจ่ายแรงดันมาที่ระดับศูนย์ เพื่อให้ทรานซิสเตอร์อยู่ในสภาวะไม่ทำงานอย่างสมบูรณ์เพราะฉะนั้นช่วงเวลาไม่นำกระแส $(t_{off}) = t_s + t_f + t_d$

2.5.2 การใช้สวิตช์ทรานซิสเตอร์ควบคุมโหลด

ในการนำเอาทรานซิสเตอร์ไปใช้ในการควบคุมโหลดประเภทต่าง ๆ นั้นสามารถกระทำได้ดังตัวอย่าง เช่น การควบคุมติด-ดับของแอลอีดี , การควบคุมการปิด-เปิดหลอดไฟ , การควบคุมรีเลย์ , การควบคุมมอเตอร์ไฟฟ้ากระแสตรงให้หมุนทิศทางเดียว หรือหมุนสองทิศทาง เป็นต้น ดังแสดงในรูปที่ 2.21 แต่สิ่งที่จะสังเกตได้จากวงจรคือ สวิตช์ทรานซิสเตอร์จะต่ออันดับกับโหลดเสมอ



รูปที่ 2.21 แสดงวงจรสวิตช์ทรานซิสเตอร์ที่ใช้งาน

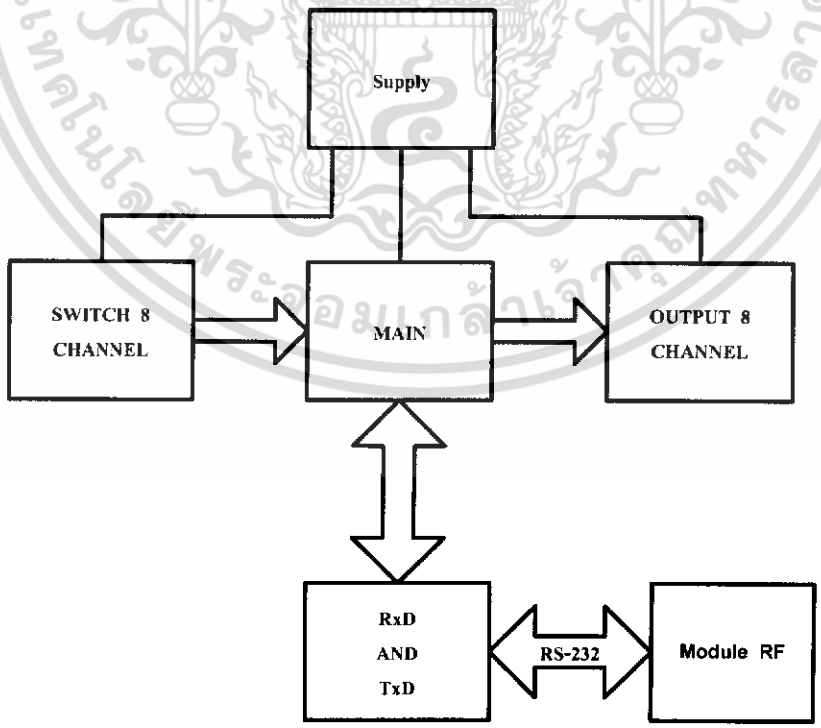
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

การสร้างและการออกแบบเครื่องรับส่งสัญญาณด้วยรีโมทคอนโทรล ในโครงงานนี้แบ่งการออกแบบเป็น 2 ส่วน คือวงจรทางด้านฮาร์ดแวร์ และซอฟแวร์ควบคุมวงจร โดยทางด้านฮาร์ดแวร์นี้จะใช้ไมโครคอนโทรลเลอร์เป็นหัวใจหลักในการประมวลผลสัญญาณจากรีโมทคอนโทรล และควบคุมส่วนของการแสดงผล ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีคุณสมบัติเหมาะสมเนื่องจากมีพอร์ตในตัวใช้งานง่าย และในโครงงานนี้เลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 มีหน่วยความจำ 8 กิโลไบต์ ซึ่งเพียงพอกับโครงงานนี้ ทางด้านซอฟแวร์ควบคุมวงจรใช้ภาษาแอสแซมบลีของ MCS-51 ในการเขียนโปรแกรมเพื่อกำหนดการส่งและรับสัญญาณที่รีโมทคอนโทรล และอ่านสัญญาณที่รับเข้ามาทางด้านโมดูลอาร์เอฟ และส่งต่อสัญญาณเพื่อทำการประมวลผลที่ตัวแม่ จากนั้นตัวแม่จะทำการประมวลผล เมื่อมีการประมวลผลเสร็จตามคำสั่งของสัญญาณที่ได้รับมา ก็จะมีการส่งสถานะที่ทำการประมวลผลกลับไปยังเครื่องรับสัญญาณที่รีโมทคอนโทรลโดยผ่านทางโมดูลอาร์เอฟ ซึ่งติดตั้งอยู่ที่ตัวแม่ส่งไปยังโมดูลอาร์เอฟ ที่ตัวรีโมทคอนโทรลและประมวลผล เพื่อแสดงสถานะ การทำงานผ่านหน้าจอแอลซีดี

3.1 บล็อกไดอะแกรมของโครงงาน

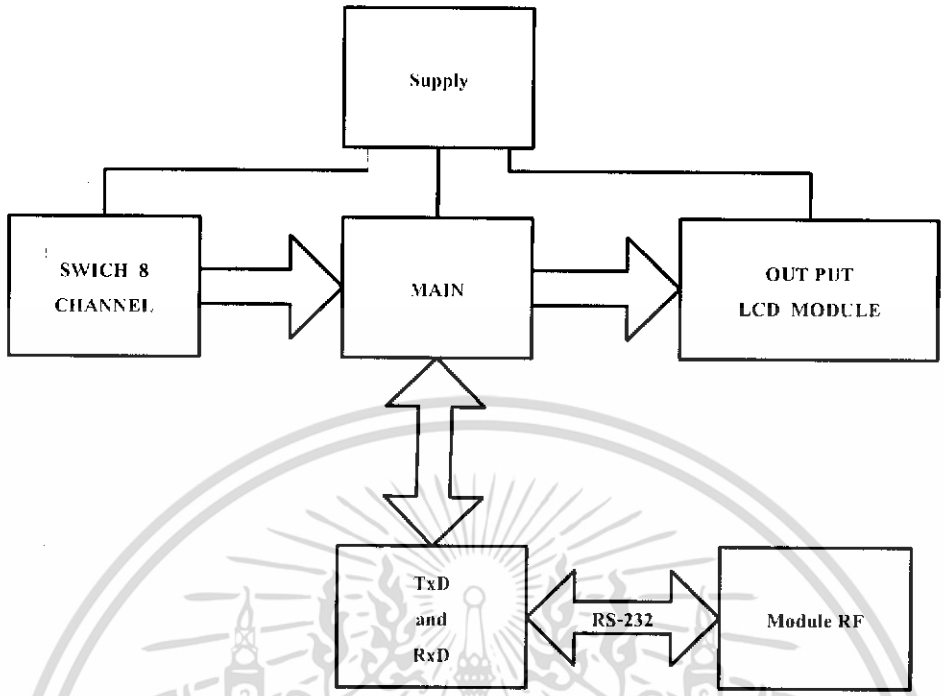
3.1.1 บล็อกไดอะแกรมของตัวแม่



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 บล็อกไดอะแกรมของรีโมทคอนโทรล



รูปที่ 3.2 แสดงบล็อกไดอะแกรมของรีโมทคอนโทรล

3.2 การออกแบบฮาร์ดแวร์

3.2.1 ส่วนประมวลผล

การออกแบบในโครงงานนี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C52 ของบริษัท Atmel ซึ่งมีพอร์ตการใช้งานที่เพียงพอสำหรับโครงงานนี้ สามารถโปรแกรมซ้ำได้ 1000 ครั้ง และมีการต่อขาการใช้งานทั้งหมดดังนี้

ขา 1 ถึง 8 คือ พอร์ต 1.0 ถึง 1.7 ต่อกับสวิตช์

ขา 9 เป็นขารีเซตของไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์เริ่มต้นการทำงานของโปรแกรมใหม่ทั้งหมด

ขา 10 คือ พอร์ต 3.0 ทำหน้าที่รับสัญญาณจากโมดูลอาร์เอฟ คือ ซีพียู (CPU) ตัวที่อยู่ในส่วนของรีโมทคอนโทรล และทำหน้าที่รับสัญญาณจากรีโมทคอนโทรลคือ ซีพียู ตัวที่อยู่กับตัวแม่

ขา 11 คือ พอร์ต 3.1 ทำหน้าที่ส่งสัญญาณจากรีโมทคอนโทรลคือ ซีพียู ตัวที่อยู่ในส่วนของรีโมทคอนโทรลและทำหน้าที่ส่งสัญญาณจากโมดูลอาร์เอฟกลับมาที่รีโมทคอนโทรล คือ ซีพียู ตัวที่อยู่กับตัวแม่

ขา 16 คือ พอร์ต 3.6 ต่อ RS (Register Selection) ของแอลซีดีโมดูล เพื่อใช้ในการเซตการทำงานของแอลซีดีโมดูล

ขา 17 คือ พอร์ต 3.7 ต่อกับขาเอ็นเอเบิลของแอลซีดีโมดูล เพื่อใช้ในการกำหนดสภาพการรับเขียนอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา 18 กับ 19 ต่อคริสตอลเพื่อเป็นฐานเวลาให้กับไมโครคอนโทรลเลอร์

ขา 20 เป็นขากราวด์ของไมโครคอนโทรลเลอร์

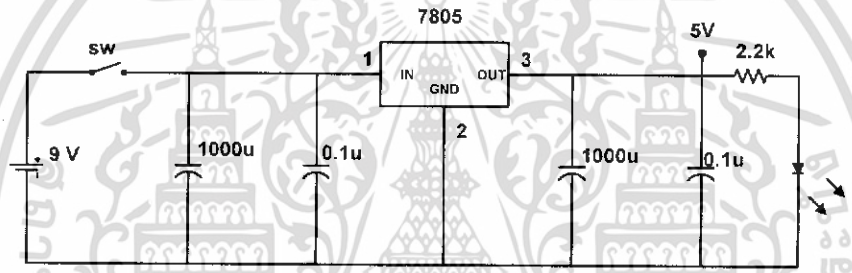
ขา 31 เป็นขาเอ็นเอเบิลของไมโครคอนโทรลเลอร์ เนื่องจากขานี้ไม่ใช้จึงต่อไว้กับไฟ 5 โวลต์ ขาเอ็นเอเบิลปีนแอกตีฟโลว์

ขา 32 ถึง 39 พอร์ต 0.0 ถึงพอร์ต 0.7 ต่อกับขา D1 ถึง D7 เพื่อส่งข้อมูลให้กับ ดีดีแรมของแอลซีดีโมดูล เพื่อใช้ในการส่งสัญญาณควบคุมการแสดงผล

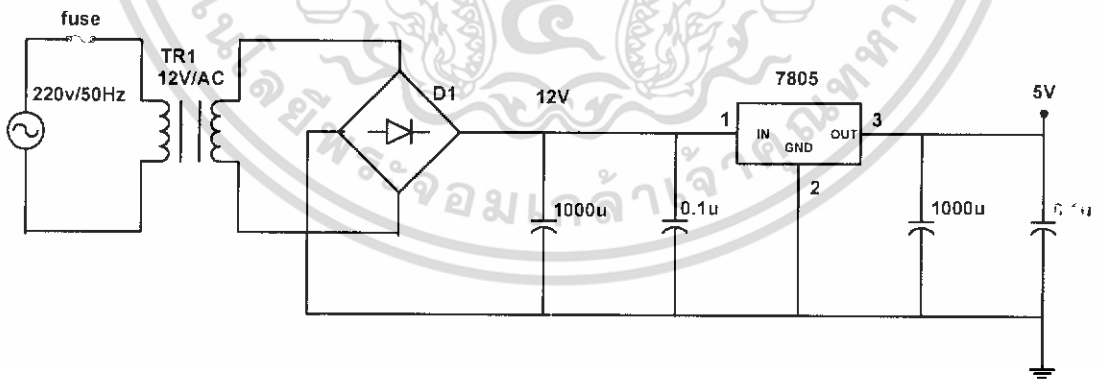
ขา 40 ต่อกับแรงดันไฟ 5 โวลต์

3.2.2 ภาคจ่ายไฟ

ในโครงงานนี้ใช้ไอซีเร็กกูเลต (Recgurate) เบอร์ 7805 เพื่อแปลงแรงดันไฟ 9 โวลต์ให้เหลือแรงดัน 5 โวลต์ และป้อนให้กับวงจรในส่วนของรีโมทคอนโทรล และใช้ไอซีเร็กกูเลต เบอร์ 7805 แปลงแรงดันไฟตรงจาก 12 โวลต์ ให้เป็นแรงดันไฟ 5 โวลต์และป้อนให้กับวงจรในส่วนของตัวแม่



รูปที่ 3.3 วงจรภาคจ่ายไฟของชุดรีโมทคอนโทรล

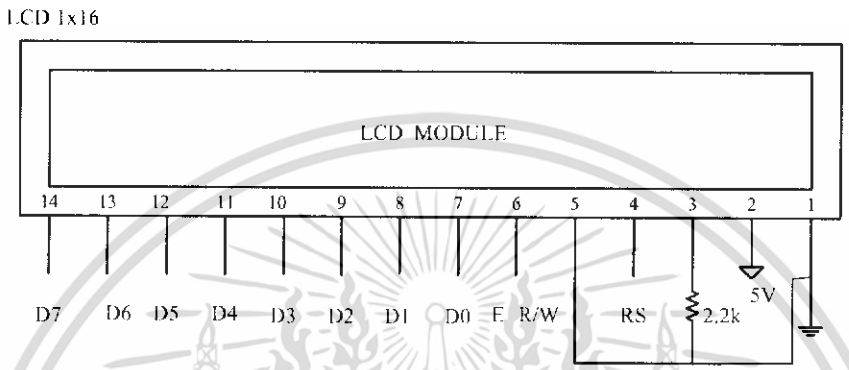


รูปที่ 3.4 วงจรภาคจ่ายไฟของตัวแม่

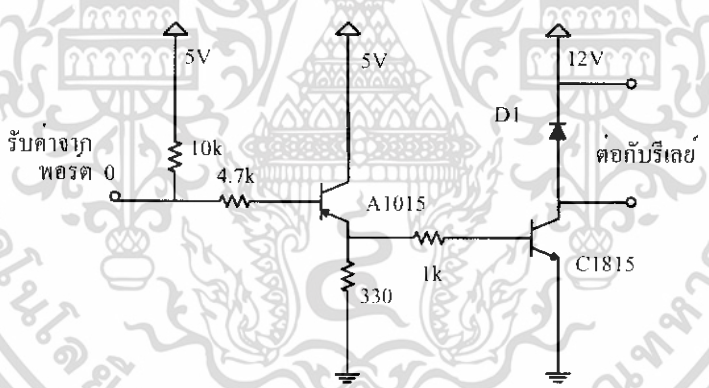
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ส่วนการแสดงผล

ส่วนการแสดงผลใช้ ด็อตเมตริกซ์แอลซีดีโมดูล ขนาด 16 ตัวอักษร 1 บรรทัด เพื่อใช้แสดงผล ซึ่งภายในตัวแอลซีดีโมดูลประกอบด้วย HD44780 ซึ่งเป็นตัวคอนโทรลเลอร์ทำหน้าที่ควบคุมการแสดงผล รูปแบบตัวอักษรหรือสัญลักษณ์ต่างๆ โดยที่การต่อแอลซีดีโมดูล เราสามารถติดต่อกับแอลซีดีโมดูลแบบ 4 บิต หรือ แบบ 8 บิต ก็ได้ ซึ่งในโครงงานนี้จะใช้การติดต่อแบบ 8 บิต คือ D1-D7 การต่อวงจรแสดงดังรูปที่ 3.5



รูปที่ 3.5 การต่อวงจรเพื่อควบคุมจอแอลซีดี



รูปที่ 3.6 วงจรไดรฟ์เวอร์

3.2.4 ภาคไดรฟ์เวอร์ (Driver)

ต่อวงจรทรานซิสเตอร์แบบดาร์ลิงตัน ซึ่งมีทรานซิสเตอร์เบอร์ A1015 และ C1815 ประกอบอยู่ในส่วนของไดรฟ์เวอร์ เพื่อเป็นตัวขับให้รีเลย์ทำงาน แสดงดังรูปที่ 3.6

3.2.5 โมดูลการรับและส่งสัญญาณความถี่วิทยุ (Radio Frequency : RF)

ET-RF24G V1.0 เป็นชุดสำเร็จ (Signal Converter) ที่จะเปลี่ยนแปลงสัญญาณในระบบอา. ส สองสามสองที่เป็นสายให้สามารถส่งข้อมูลออกไปได้ในแบบไร้สายเป็นสัญญาณอาร์เอฟความถี่ 2.4 กิกะเฮิร์ต โดยทางอีทีที (ETT) ได้นำชุดโมดูลสำเร็จที่ใช้ส่งสัญญาณอาร์เอฟ รุ่น TRW-2.4G มาใช้เป็นตัวส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณอาร์เอฟ ร่วมกับบอร์ดคอนโทรลเลอร์ ขนาดเล็กที่ทำขึ้น สามารถรับส่งได้ในระยะไกล 50 เมตร ถึง 280 เมตร ลดปัญหาการต่อการทดลองกับวงจรอาร์เอฟต่างๆ ที่ประกอบขึ้นเองแล้วไม่ได้คุณภาพรับและส่งไม่ได้ ซึ่งมีคุณสมบัติดังนี้

- มีรูปแบบและความเร็วสูงในการ รับ-ส่ง ข้อมูลสูง โดยส่งแบบ GFSK (Gaussian Frequency Shift Keying) ความเร็ว รับ-ส่ง ข้อมูลสูงถึง 1 เมกกะบิตต่อวินาที, 250 เมกกะบิตต่อวินาที มีทั้งตัวรับและตัวส่งในชุดเดียวกัน โดยเป็นแบบสลับกันทำงานรับส่ง

- ความถี่ในการใช้งานที่ 2.4 กิกะเฮิร์ต - 2.524 กิกะเฮิร์ต

- กำลังส่ง 10 มิลลิวัตต์ พร้อมเสาอากาศในตัว

- สามารถตั้งช่องความถี่ใช้งานด้วยโปรแกรมได้ถึง 125 ช่องสัญญาณ ทำให้สามารถใช้ชุด ET-RF24G นี้ได้หลายๆ ตัว ในพื้นที่เดียวกัน และนอกจากนี้ยังสามารถเซตตั้งไอดีโค้ด (ID Code) ในการรับหรือ ไอดีโค้ดในการส่ง ก็ได้ถึง 256 ตำแหน่งต่อช่องทำให้สามารถประยุกต์ใช้เป็นแบบโครงข่าย (RF Network) ได้อีกด้วย

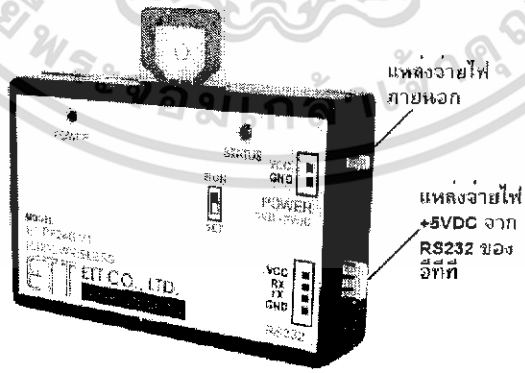
- ระยะทางในการรับ-ส่ง สัญญาณ 280 เมตร (250 เมกกะบิตต่อวินาที) -150 เมตร (1 เมกกะบิตต่อวินาที) กลางแจ้งและในตัวอาคารระยะทาง 50 เมตร

- ขั้วต่ออาร์เอสสองสามสอง เป็นแบบขั้ว 4 ขา มาตรฐาน อีทีที สามารถต่อเข้ากับบอร์ดต่างๆ ของทาง อีทีที ได้โดยตรง พร้อมใช้แหล่งจ่ายร่วมกับบอร์ดของ อีทีที จ่ายให้กับชุด ET-RF24G ได้เลย หรือจะต่อกับขั้วแบบ DB 9 ในชุดก็จะมีสายต่อ DB 9 เข้าคอมพิวเตอร์ให้อีกด้วย

- ใช้แหล่งจ่ายกำลังกระแสตรง 5 โวลต์ และ 9โวลต์ กระแสสูงสุด 30 มิลลิแอมป์ เป็นขั้วแบบ 4 ขา และแบบ 2 ขา

- ขนาดกล่อง 2.2 x 5 x 7.5 เซนติเมตร น้ำหนัก 0.05 กิโลกรัม

- พร้อมโปรแกรมการใช้งานตั้งค่าจาก อีทีที ที่คุณสามารถเปลี่ยนแปลงตั้งการทำงานได้โดยง่าย เช่น ตั้งค่าบิตเรท 1200-19200 บิตต่อวินาที



รูปที่ 3.7 โมดูลสำเร็จรูปรับและส่งสัญญาณความถี่วิทยุ (Radio Frequency : RF)

- ชุด ET-RF24G สามารถประยุกต์ใช้งานต่าง ๆ ได้มากมาย เช่น รีโมทคอนโทรลงานควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ แสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หุ่นยนต์ระยะไกล ป้ายแสดงต่างๆ ไม่ต้องเดินสาย, ล็อกข้อมูล, อาร์เอสสองสามสองแบบออปโต้ไอโซเลชั่น (Opto Isolation) สามารถประยุกต์ใช้งานแบบฟูลดูเพ็ลกซ์ (Full Duplex) ได้โดยใช้ ET-RF24G ด้านละ 2 ชุด

- ชุด ET-RF24G ประกอบด้วย ET-RF24G พร้อมกล่อง สายต่อแบบ DB9 สายต่อแบบ 4 ขา เข้ากับบอร์ดของ อีทีที และซีดีรอมโปรแกรม แสดงอุปกรณ์ดังรูปที่ 3.7

3.3 การออกแบบซอฟต์แวร์ควบคุมวงจร

3.3.1 หลักการออกแบบซอฟต์แวร์ด้านส่งหรือรีโมทคอนโทรล

ส่วนควบคุม ใช้สวิตซ์ในการควบคุมโดยเรากำหนดให้ใช้เพียงแค่ 4 ปุ่ม เมื่อเรากดปุ่มที่ 1 คือ ทำการเลือก ช่องสัญญาณซึ่งมีทั้งหมด 8 ช่อง และอีก 1 ช่องคือ "Control All" จะควบคุมการเปิดปิดทั้งหมดทั้ง 8 ช่อง รวมทั้งหมด 9 ช่อง ส่วนปุ่มที่ 2 เป็นการสั่งให้ช่องสัญญาณนั้น " ทำงาน" ส่วนปุ่มที่ 3 เป็นการสั่งให้ ช่องสัญญาณนั้น "หยุดทำงาน" ส่วนปุ่มที่ 4 เป็นการสั่งให้ออกจากโปรแกรม แล้วกลับมาอยู่ที่สภาวะเริ่มแรก "Module Ready"

ส่วนแสดงผล มีหน้าจอแอลซีดีในการแสดงผลว่าขณะนั้นช่องสัญญาณไหน "ทำงาน" หรือ "หยุดทำงาน" หรือว่าขณะนี้ขาดการสื่อสารระหว่างเครื่องรับและส่ง

ส่วนของการทำงานของด้านส่งนั้นเมื่อต้องการกดเปิดจะทำการส่งข้อมูลเป็นตัวอักษร A, B, C, D, E, F, G, H, Y จากช่องที่ 1 ถึง ช่องที่ 8 ตามลำดับ และควบคุมเปิดทั้ง 8 ช่องสัญญาณ และเมื่อต้องการกดปิดจะทำการส่งข้อมูลเป็นตัวอักษร a, b, c, d, e, f, g, h, X จากช่องที่ 1 ถึง ช่องที่ 8 และควบคุมปิดทง 8 ช่องสัญญาณตามลำดับ เพื่อนำไปประมวลผลทางด้านตัวแม่ต่อไป และเมื่อไม่มีการกดเปิดหรือปิดด้านส่งจะมีการส่งตัวอักษร S ร้องขอสถานะ และทางตัวแม่จะส่งสถานะกลับมาเพื่อแสดงผลบนหน้าจอแอลซีดีว่าอุปกรณ์ทำงานหรือไม่ทำงาน

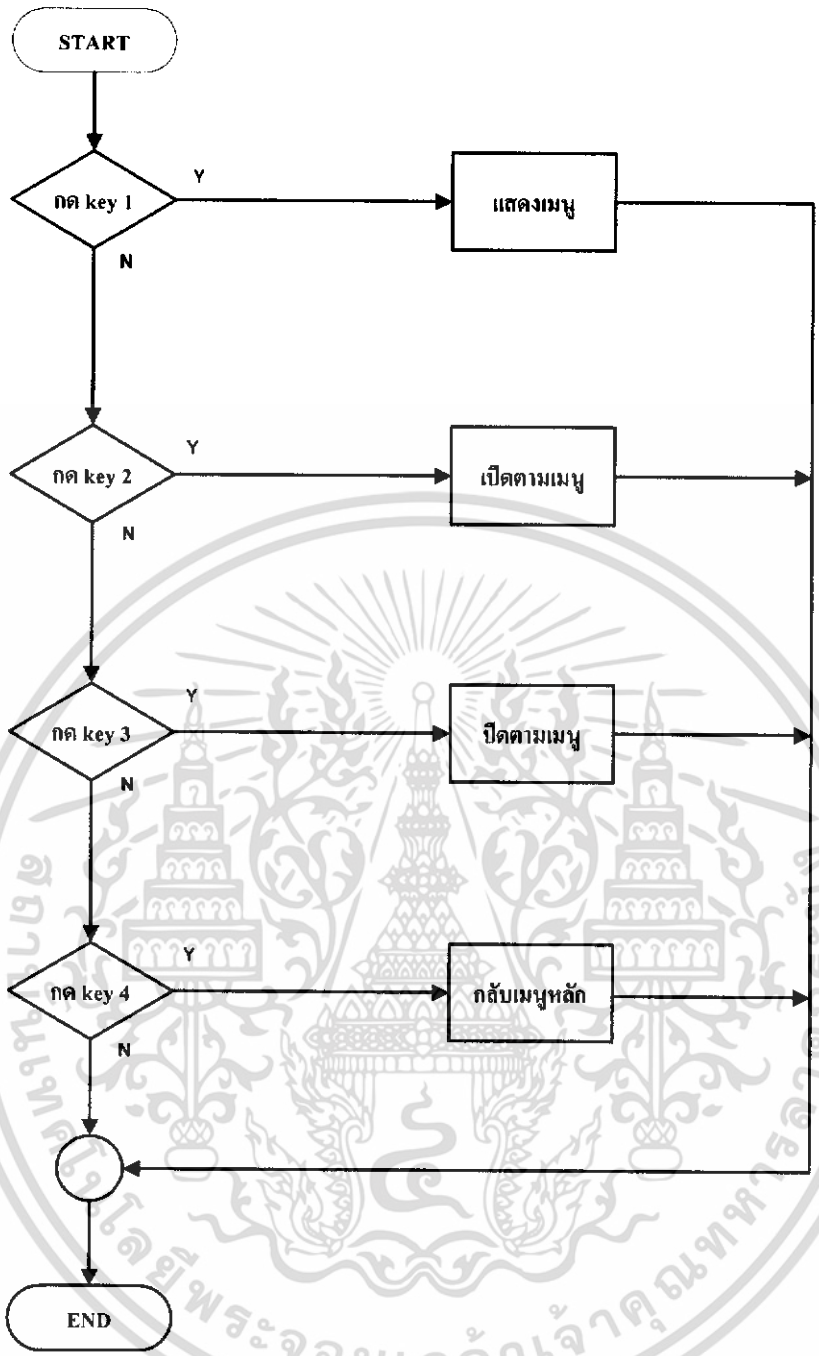
ส่วนของการทำงานการประมวลผลข้อมูลที่ทางตัวแม่ ส่งข้อมูลกลับมาเพื่อแสดงผลนั้นข้อมูลจะเป็นข้อมูลตัวอักษรติดกันมาทั้ง 8 ตัว โดยซีพียูของรีโมทคอนโทรลจะประมวลผลตามค่าที่ได้รับกลับมาดังแสดงโฟลว์ชาร์ตหลักการทำงานดังรูปที่ 3.8

3.3.2 หลักการออกแบบซอฟต์แวร์ด้านรับหรือตัวแม่

ส่วนควบคุมจากสวิตซ์โดยตรง โดยทำการกดสวิตซ์ทั้ง 8 ตัวที่ติดตั้งไว้ที่ตัวแม่ควบคุมอุปกรณ์ เมื่อเราทำการกด 1 ครั้งอุปกรณ์จะเปิด และกดอีก 1 ครั้งเพื่อปิด เมื่อกดเปิดทำการส่งข้อมูลเป็นตัวอักษร A, B, C, D, E, F, G, H และเมื่อกดปิดจะทำการส่งข้อมูลเป็นตัวอักษร a, b, c, d, e, f, g, h ออกไปครั้งละ 1 ตัว จากทั้งหมด 16 ตัว เพื่อทำการเปิดปิดอุปกรณ์ และแสดงสถานะเปิดปิดเป็นลำดับต่อไป

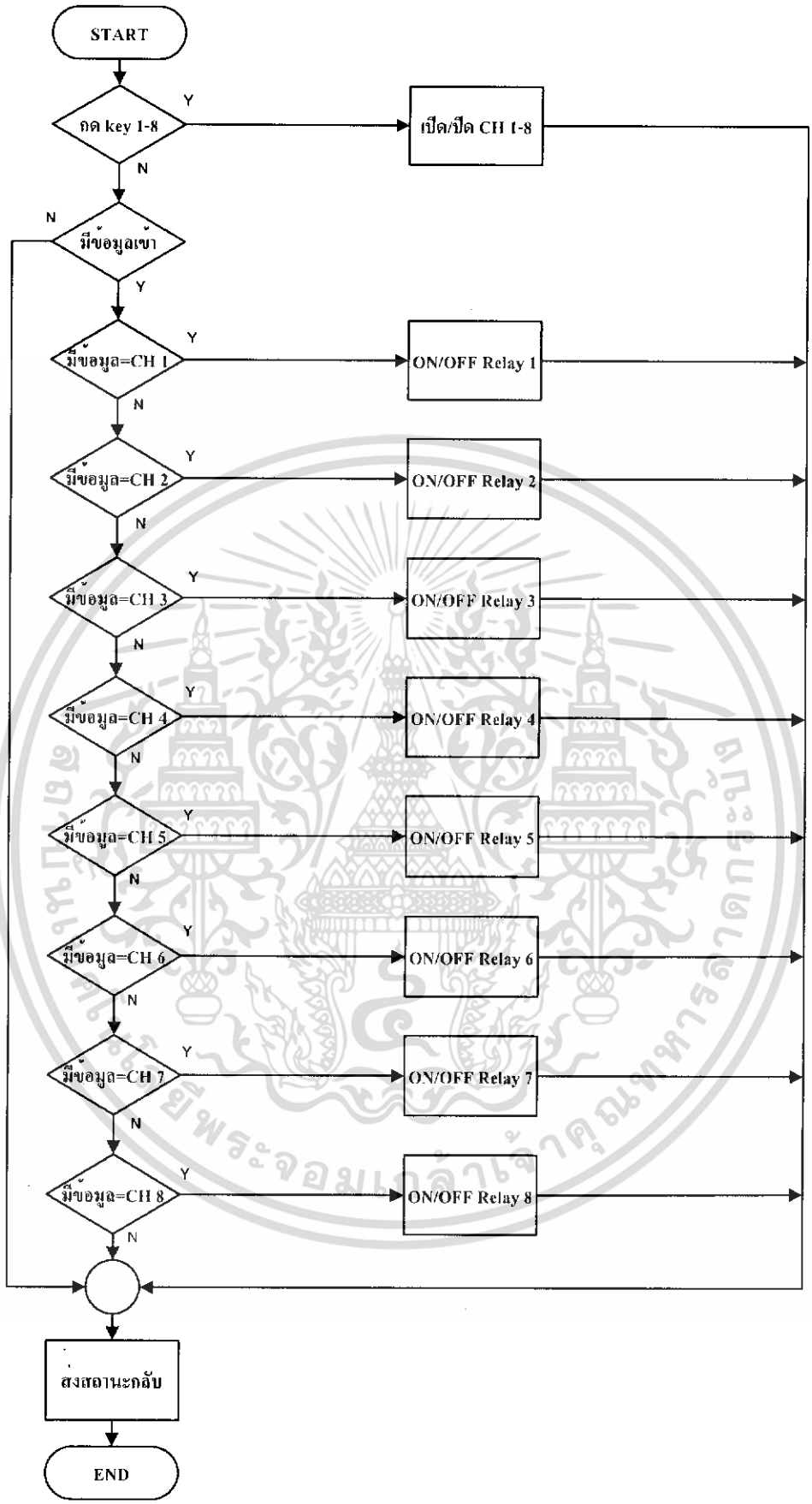
ส่วนที่รับข้อมูลจากด้านส่งหรือรีโมทโดยทางด้านรับ จะทำการรับค่าตัวอักษรจากรีโมทว่าเป็นข้อมูลตัวอักษรตัวอักษรตัวใด และซีพียูตัวแม่จะทำการประมวลผลเพื่อเปิดหรือปิดในแต่ละช่องสัญญาณแล้วทำการส่งสถานะกลับไปที่ด้านส่งว่าขณะนั้นอุปกรณ์นั้นเปิดหรือปิด โดยทำการส่งตัวอักษรติดกันไปทั้ง 8 ตัว เช่น เมื่อช่องสัญญาณ 1-4 ปิดอยู่และช่องสัญญาณ 5-8 เปิดอยู่ ข้อมูลที่ส่งกลับมาเพื่อแสดงผลของรีโมทคอนโทรลจะเป็น a, b, c, d, E, F, G, H ตามลำดับ ดังแสดงโฟลว์ชาร์ตหลักการทำงานดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 โฟลว์ชาร์ตแสดงหลักการทำงานของตัวรีโมทคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



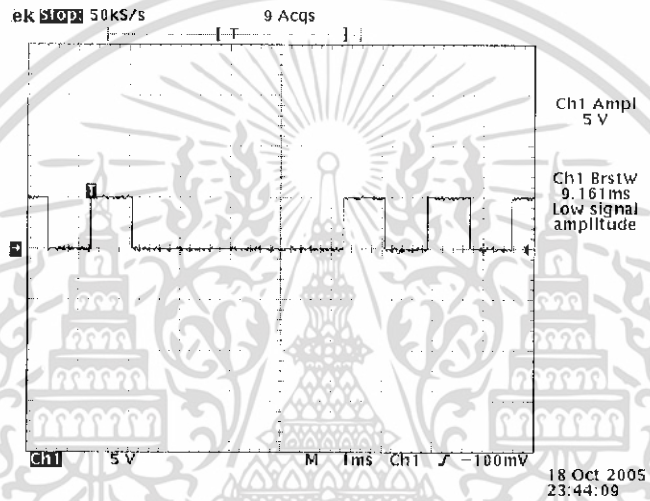
รูปที่ 3.9 โฟลว์ชาร์ตหลักการทำงานของตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

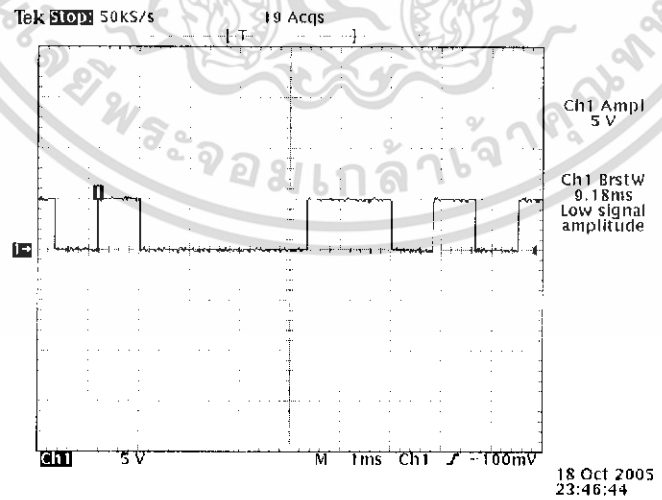
บทที่ 4
การทดลอง

4.1 การตรวจวัดสัญญาณที่ออกมาจากรีโมทคอนโทรล

ซึ่งวัดจากขา 10 คือ RxD ของซีพียูที่อยู่กับตัวแม่ และวัดสัญญาณที่ออกมาจากขา 11 คือขา TxD จากรีโมทคอนโทรลเพื่อที่จะนำไปส่งเป็นสัญญาณวิทยุโดยผ่านโมดูลต่อไป ซึ่งสัญญาณที่วัดได้เป็นรูปคลื่นพัลส์ A, B, C, D, E, F, G, H, Y เมื่อกดเปิดช่องสัญญาณ 1-8 และควบคุมเปิดทั้ง 8 ช่องสัญญาณตามลำดับ และวัดได้เป็นรูปคลื่นพัลส์ a, b, c, d, e, f, g, h, X เมื่อกดปิดช่องสัญญาณ 1-8 และควบคุมปิดทั้ง 8 ช่องสัญญาณตามลำดับ

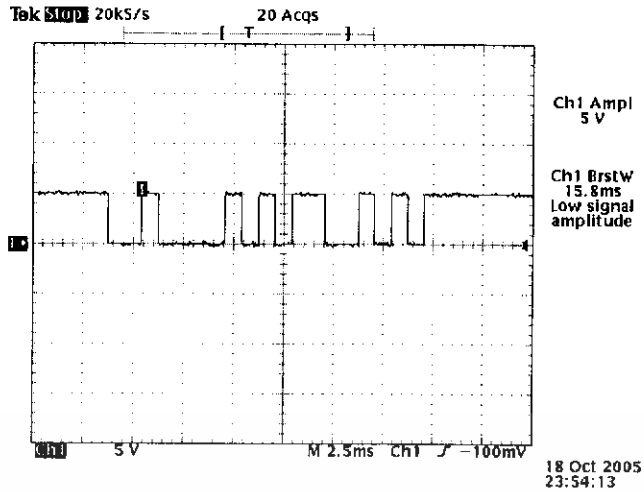


รูปที่ 4.1 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 1

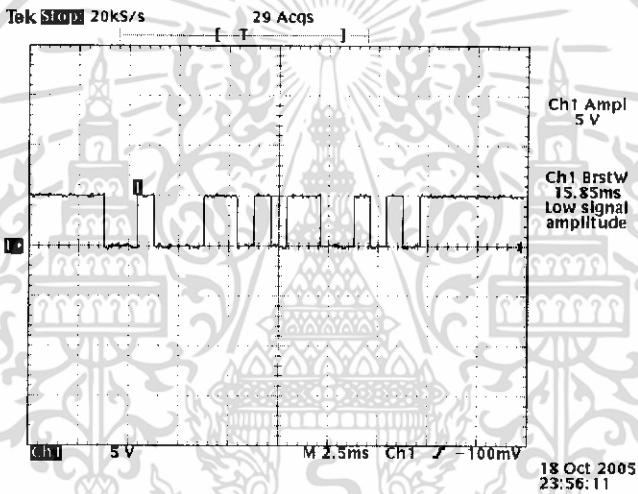


รูปที่ 4.2 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 1

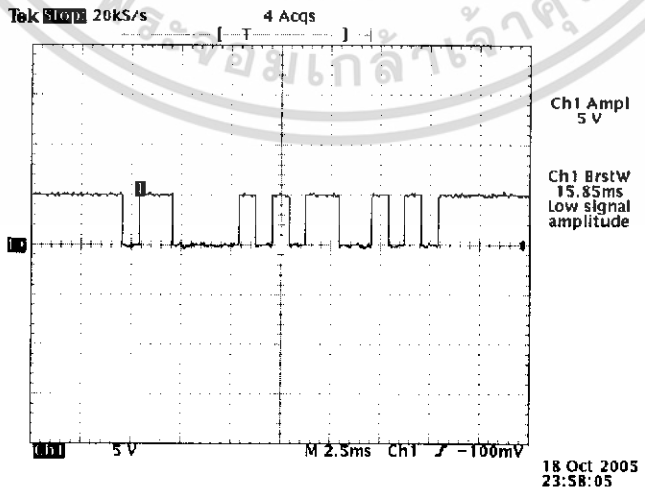
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 2

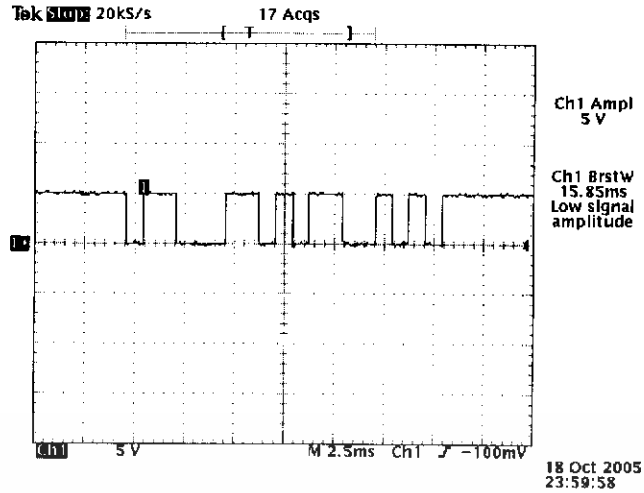


รูปที่ 4.4 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 2

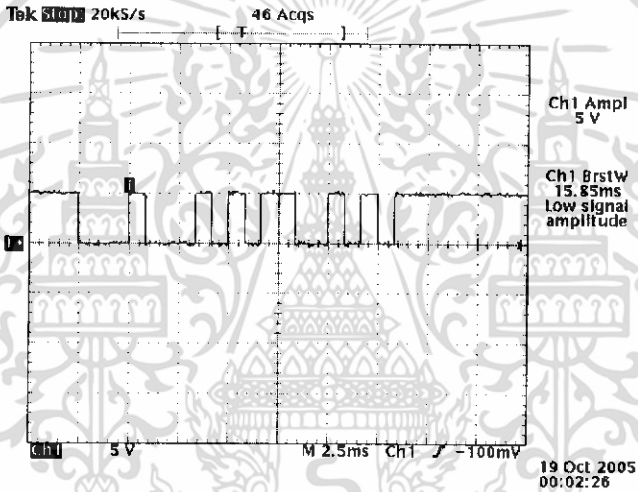


รูปที่ 4.5 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 3

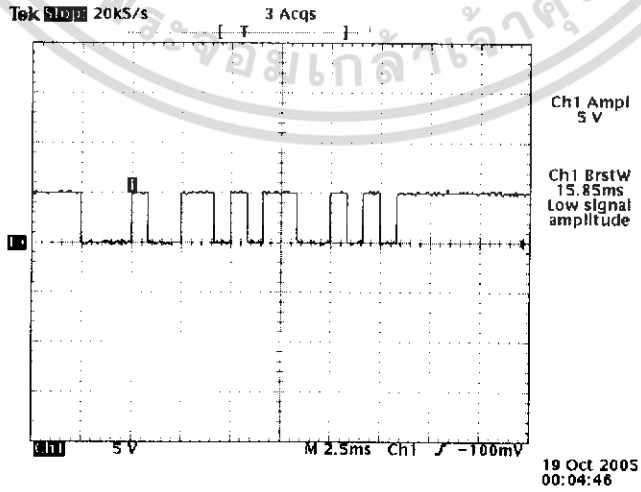
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 3

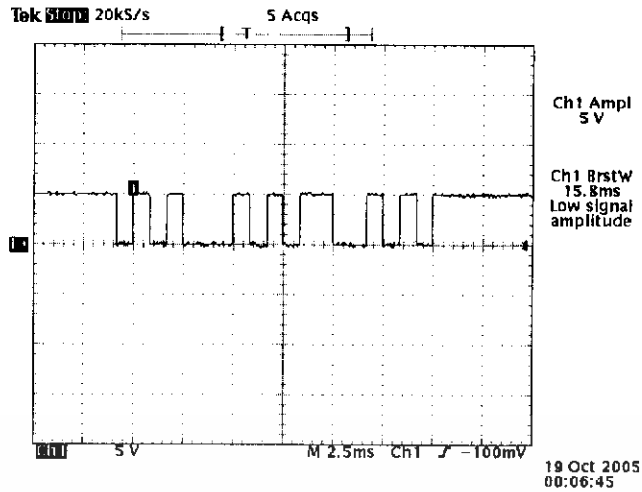


รูปที่ 4.7 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 4

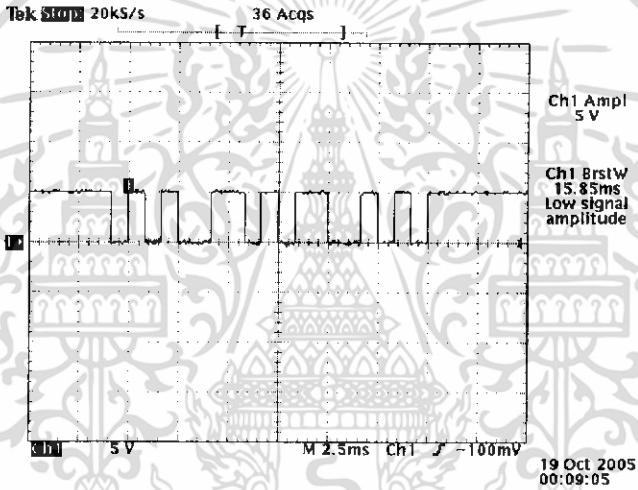


รูปที่ 4.8 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 4

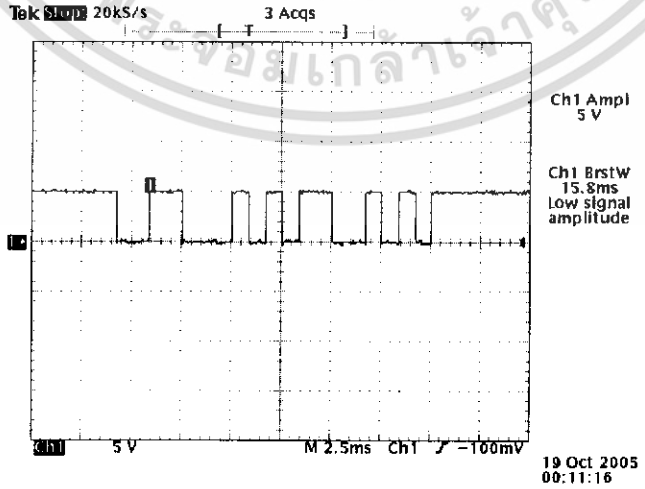
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 5

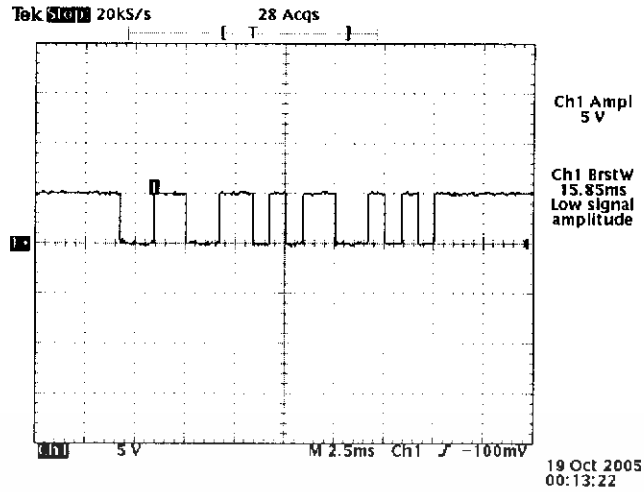


รูปที่ 4.10 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 5

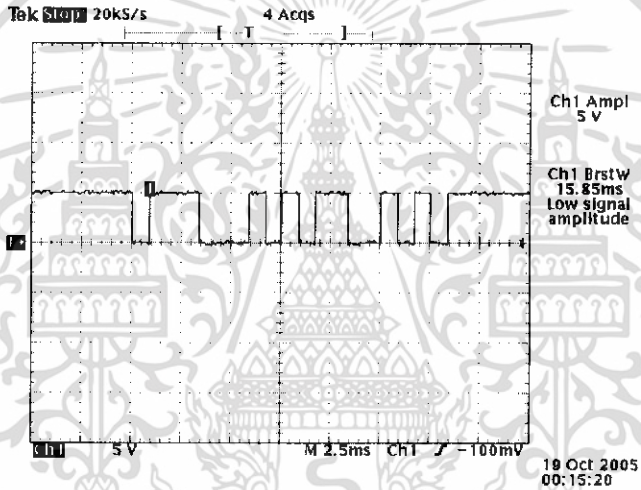


รูปที่ 4.11 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 6

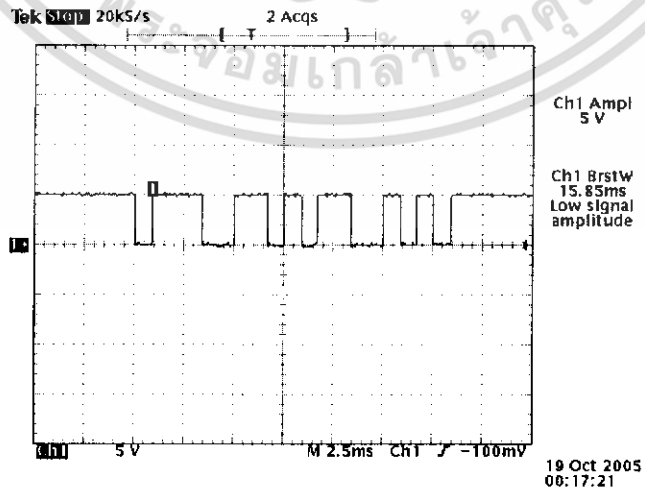
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 6

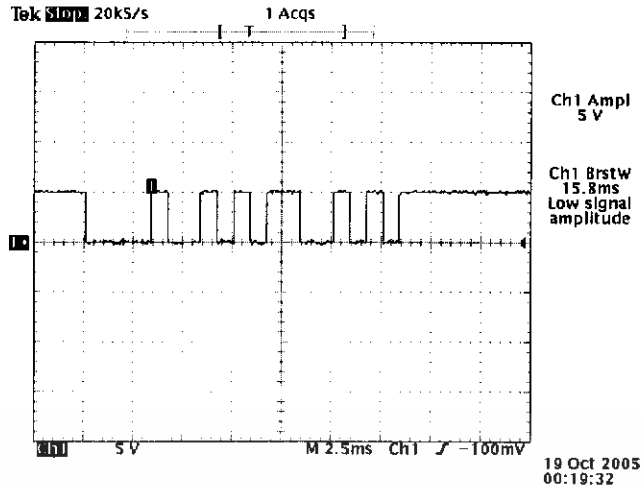


รูปที่ 4.13 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 7

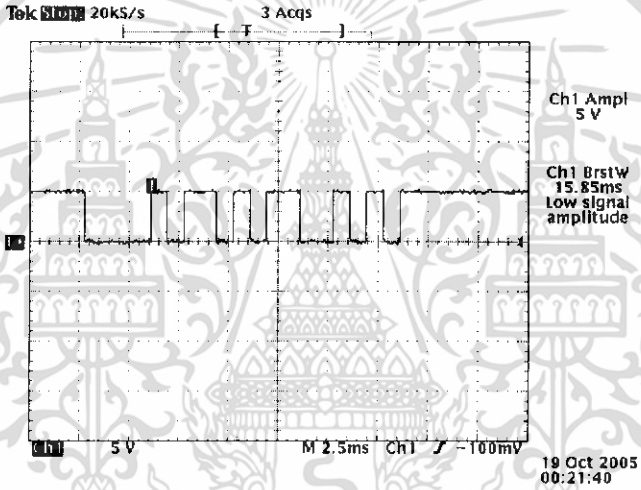


รูปที่ 4.14 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 7

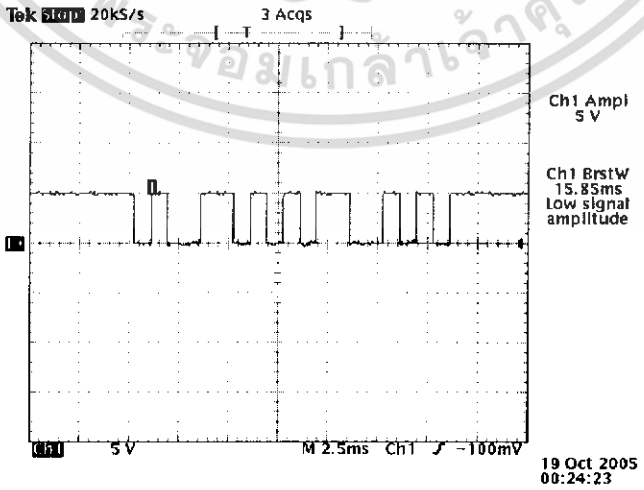
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 แสดงสัญญาณของการเปิด (ON) ช่องสัญญาณที่ 8

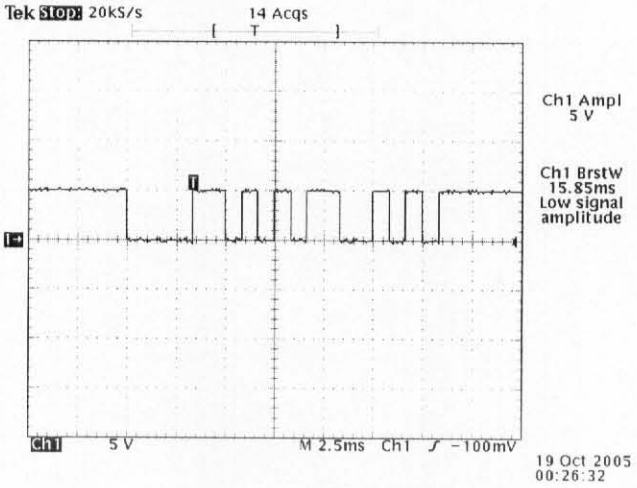


รูปที่ 4.16 แสดงสัญญาณของการปิด (OFF) ช่องสัญญาณที่ 8



รูปที่ 4.17 แสดงสัญญาณของการเปิด (ON) ทั้ง 8 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.18 แสดงสัญญาณของการปิด (OFF) ทั้ง 8 ช่อง

4. 2 การแสดงผลหน้าจอแอลซีดี

รีโมทคอนโทรลมีทั้งหมด 4 ปุ่ม

ปุ่มที่ 1 MENU ใช้เพื่อกดเลือกช่องว่าจะเลือกเปิดหรือปิดช่องสัญญาณช่องใด

ปุ่มที่ 2 ON ใช้เพื่อกดเปิดช่องสัญญาณที่ต้องการ

ปุ่มที่ 3 OFF ใช้เพื่อกดปิดช่องสัญญาณที่ต้องการ

ปุ่มที่ 4 Module Ready ใช้เพื่อกดให้รู้ว่ารีโมทพร้อมจะใช้งาน

รูปที่ 4.19 แสดงปุ่มทั้งหมดของรีโมทคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการเปิดรีโมทคอนโทรลหน้าจอจะแสดงผลหน้าจอแอลซีดี "# Module Ready #" คือรีโมทคอนโทรลพร้อมที่จะรับคำสั่งการใช้งานจากผู้ใช้งานดังแสดงในรูปที่ 4.20

เมื่อกดเปิดช่องสัญญาณหน้าจอแอลซีดี จะแสดง "CH: … => ...ON" ในที่นี้แสดงให้เห็นการเปิดช่องสัญญาณที่ 1 ดังแสดงในรูปที่ 4.21

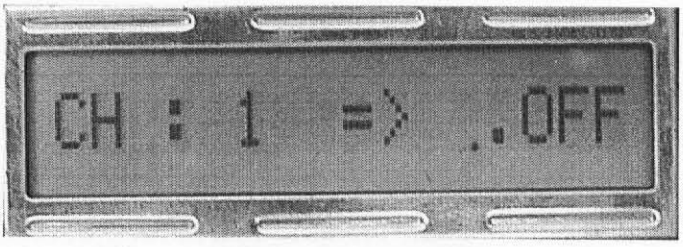
เมื่อกดปิดช่องสัญญาณหน้าจอแอลซีดี จะแสดง "CH: … => .OFF" ในที่นี้แสดงให้เห็นการปิดช่องสัญญาณที่ 1 ดังแสดงในรูปที่ 4.22

เมื่อกดเลื่อนปุ่ม Menu ตั้งแต่ช่อง 1-8 แล้วช่องสุดท้ายจะเป็นการเปิดหรือปิดช่องสัญญาณทั้งหมดโดยหน้าจอแอลซีดีจะแสดง "Control # ALL #" โดยจะทำการกดเปิดหรือปิดได้ตามความต้องการ ดังแสดงในรูปที่ 4.23

เมื่อทำการเปิดรีโมทคอนโทรลหรือขณะที่กำลังใช้งานอยู่หน้าจอแอลซีดีเกิดการแสดงผลหน้าจอ ">- Link Wait -<" คือรีโมทคอนโทรลไม่พร้อมที่จะรับคำสั่งการใช้งานจากผู้ใช้งาน หน้าจอแอลซีดีต้องแสดง "# Module Ready #" เท่านั้นรีโมทคอนโทรลถึงจะสามารถใช้งานได้ ดังแสดงในรูปที่ 4.24

รูปที่ 4.20 แสดงหน้าจอแอลซีดี "# Module Ready #" รีโมทคอนโทรลพร้อมรับคำสั่งการใช้งาน

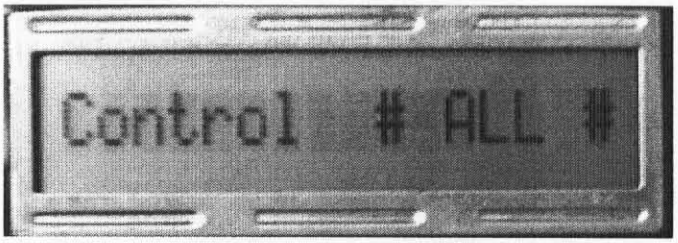
รูปที่ 4.21 แสดงตัวอย่างการแสดงผลการเปิดช่องสัญญาณโดยมีช่อง 1- 8



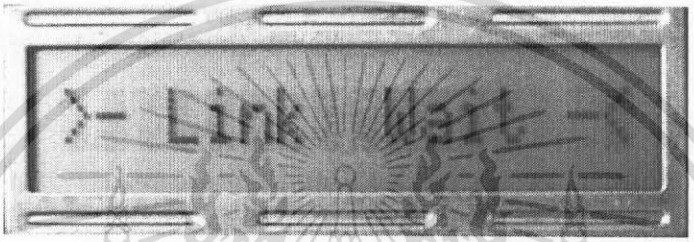
รูปที่ 4.22 แสดงตัวอย่างการแสดงผลการปิดช่องสัญญาณโดยมีช่อง 1- 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.22 แสดงตัวอย่างการแสดงผลการปิดช่องสัญญาณโดยมีช่อง 1- 8



รูปที่ 4.23 แสดง "Control # ALL #" ซึ่งเป็นการเปิด/ปิดช่องสัญญาณทั้ง 8 ช่อง

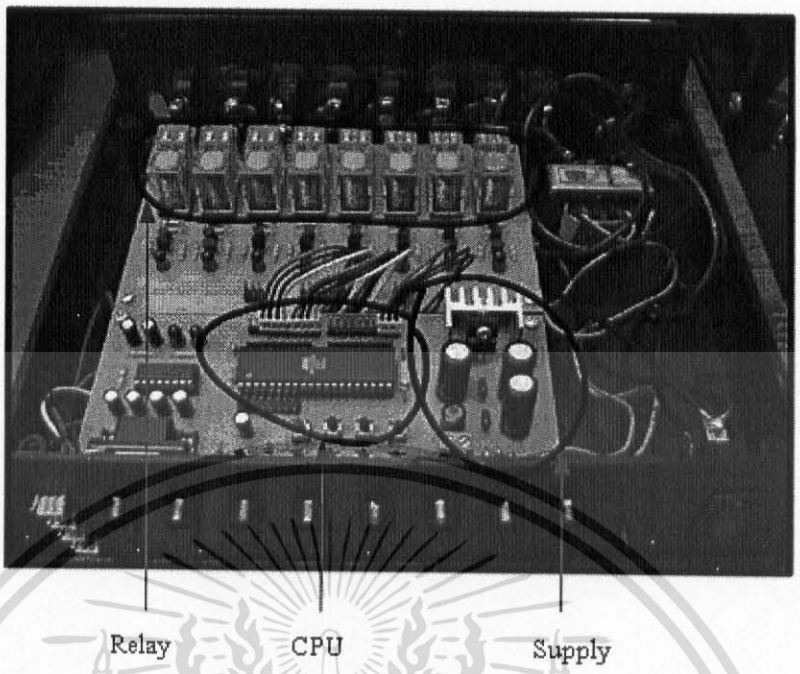


รูปที่ 4.24 แสดง "Link Wait" รีโมทคอนโทรลไม่พร้อมรับคำสั่งการใช้งาน

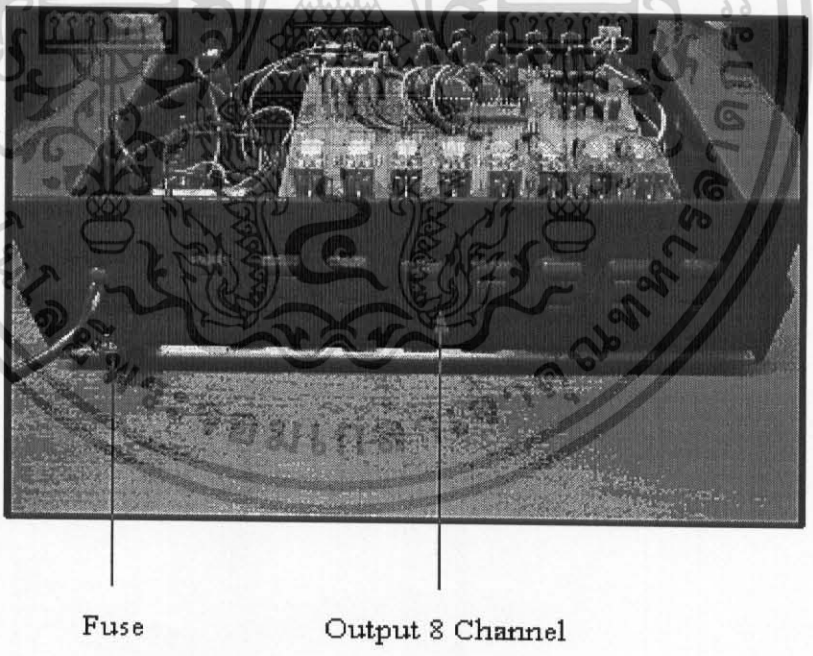
4.3 รูปแสดงชิ้นงานจริง

รูปที่ 4.25 แสดงส่วนที่เป็นตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

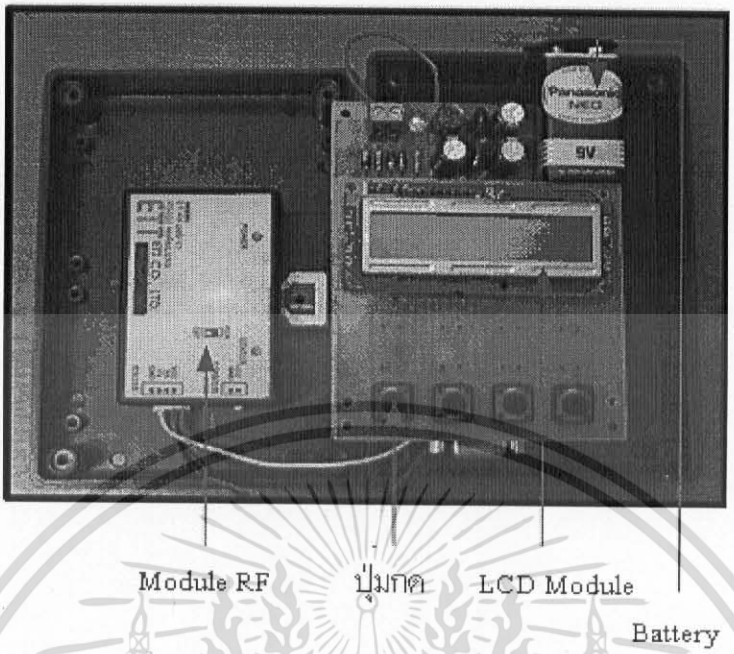


รูปที่ 4.26 แสดงโครงสร้างภายในส่วนของตัวแม่



รูปที่ 4.27 แสดงโครงสร้างด้านหลังส่วนของตัวแม่

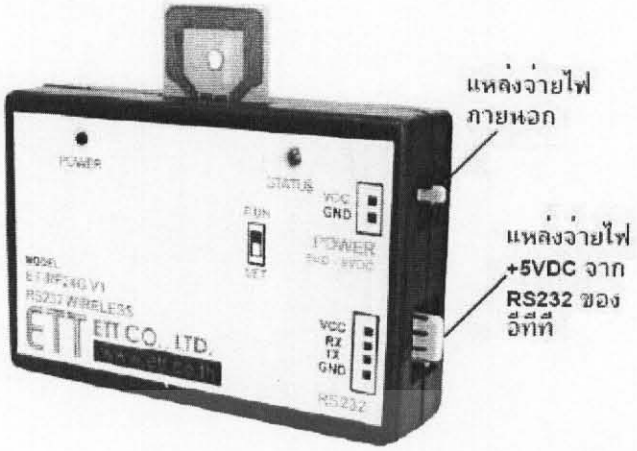
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 แสดงโครงสร้างภายในของรีโมทคอนโทรล

รูปที่ 4.29 แสดงรีโมทคอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 แสดงโมดูลอาร์เอฟที่ต่ออยู่กับตัวรีโมทและตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและวิจารณ์

รีโมทคอนโทรลควบคุมอุปกรณ์ไฟฟ้าที่ได้จัดทำขึ้นมานี้ สามารถที่จะควบคุมการเปิดและปิดอุปกรณ์ไฟฟ้าได้ นอกจากนี้ยังสามารถตรวจสอบสถานะโดยสามารถตรวจสอบได้จากหน้าจอแอลซีดีของรีโมทคอนโทรล โดยที่รีโมทคอนโทรลจะติดต่อกับตัวแม่ซึ่งมีตัวรับสัญญาณเป็นโมดูลอาร์เอฟ โดยที่ตัวแม่ตัวจะเชื่อมต่อกับอุปกรณ์ไฟฟ้า รีโมทคอนโทรลกับตัวแม่จะติดต่อสื่อสารกันด้วยคลื่นวิทยุ 2.4 กิกะเฮิร์ต ซึ่งสามารถรับส่งสัญญาณกันภายในอาคารได้ระยะทางประมาณ 50 เมตร รีโมทคอนโทรลสามารถควบคุมช่องสัญญาณทั้งหมด 8 ช่องสัญญาณ และสามารถเพิ่มช่องสัญญาณได้ตามความต้องการของผู้ใช้

แนวทางการแก้ไขและพัฒนาโครงงานนี้ สามารถที่จะพัฒนาให้มีประสิทธิภาพในการใช้งานและสามารถนำไปใช้งานจริงได้อย่างสะดวกขึ้นซึ่งมีแนวทางการแก้ไขดังนี้

1. รีโมทคอนโทรลมีขนาดใหญ่เกินไปควรมีการพัฒนาให้มีขนาดที่เล็กลงจากเดิม
2. รีโมทคอนโทรลควรที่จะมีการเข้ารหัสข้อมูลเพื่อใช้รักษาความปลอดภัยเกี่ยวกับผู้ใช้
3. ควรคำนึงถึงหลักการประหยัดพลังงาน เพราะมีความสำคัญมากต่อการแข่งขันในตลาดของเทคโนโลยีใหม่ ๆ ในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

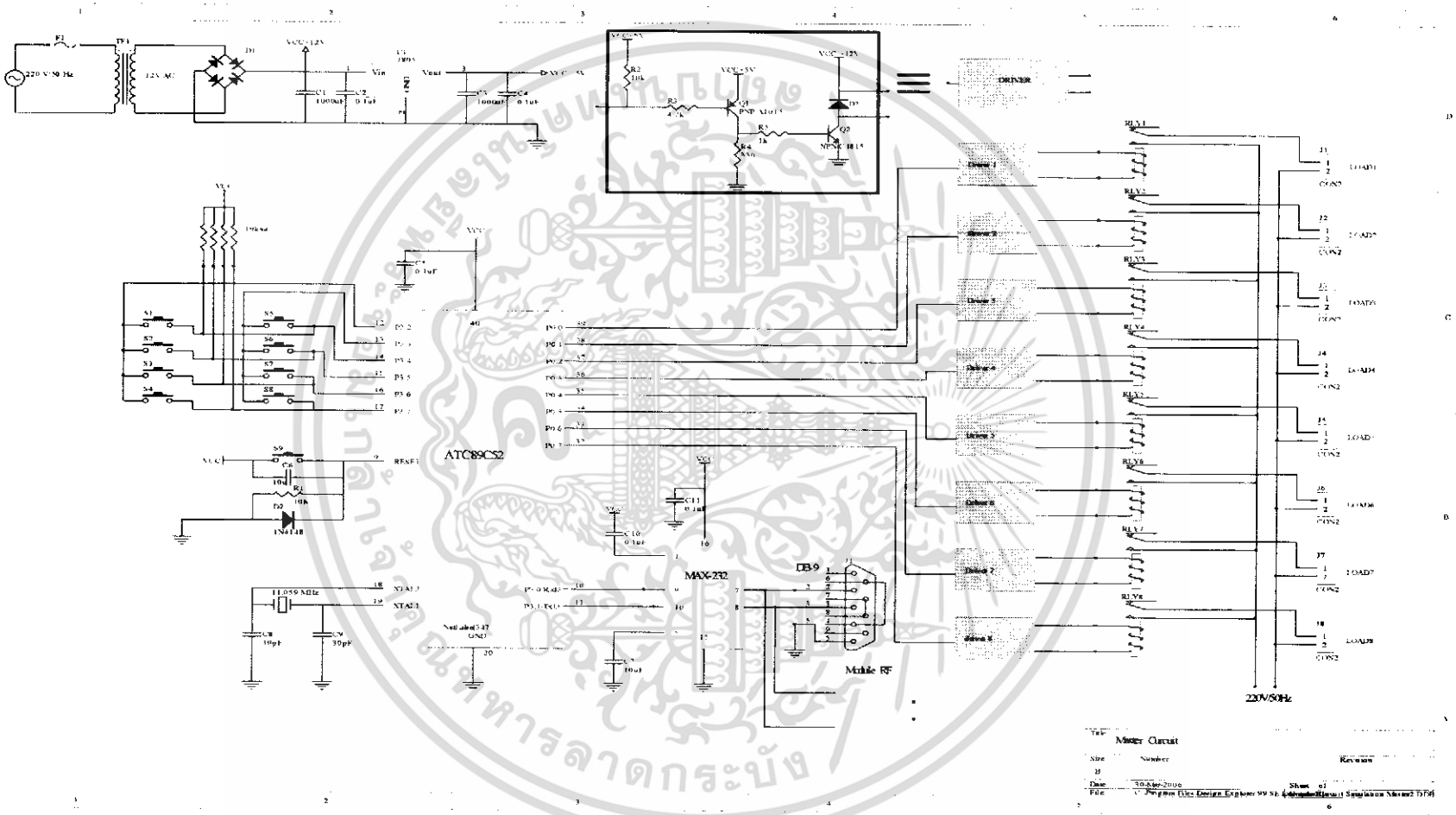
หนังสืออ้างอิง

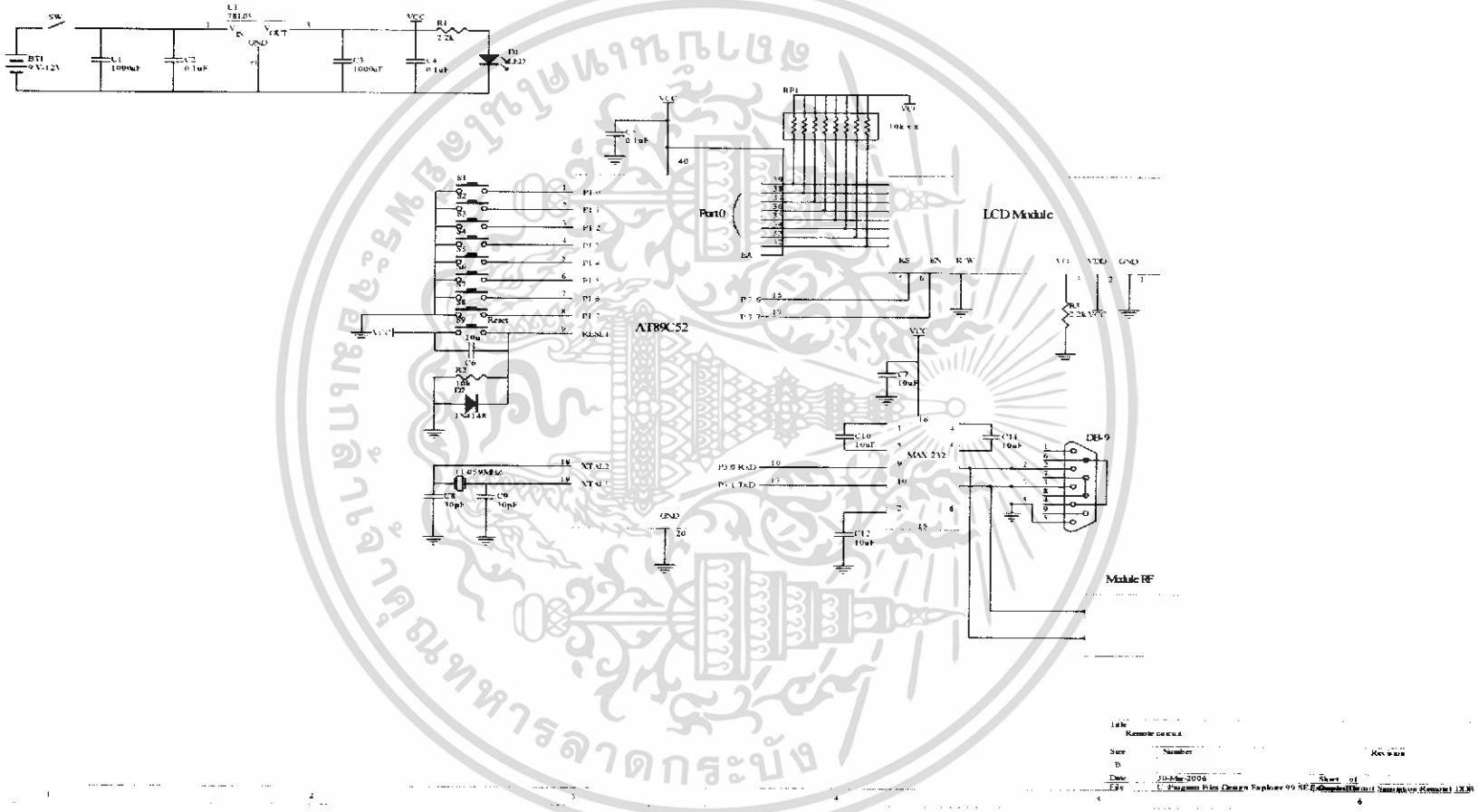
1. คู่มือ DOT MATRIX LCD MODULE, SILA CO., LTD.
2. จรัญ จันทรักษา, "รีโมทคืออะไร". ELECTRONICS HANDBOOK.2542.
3. ชัยวัฒน์ ลิ้มพรจิตรวิไล,วระจน์ กรแก้ววัฒนกุล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช" บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด
4. อนัน คัมภิรานนท์, ทฤษฎีอิเล็กทรอนิกส์อุตสาหกรรม 1,กรุงเทพฯ, 2544.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


220 V/50 Hz
12V AC
VCC=12V
Vin
Vout
VCC=5V
R2
10k
R3
4.7k
Q1
PNP 9012
R5
1k
R4
330
Q2
NPN C 1815
D2
DRIVER
Driver 1
Driver 2
Driver 3
Driver 4
Driver 5
Driver 6
Driver 7
Driver 8
RLY1
RLY2
RLY3
RLY4
RLY5
RLY6
RLY7
RLY8
J1
J2
J3
J4
J5
J6
J7
J8
LOAD1
LOAD2
LOAD3
LOAD4
LOAD5
LOAD6
LOAD7
LOAD8
COM2
220V/50Hz
ATC89C52
MAX-232
DB-9
Mobile RF
11.059 MHz
XTAL1
XTAL2
RESET
1N4148
Master Circuit
Size
Number
Revision
Date
30-Sep-2006
Sheet of


SW
BT1
9 V-12V
C1
1000uF
C2
0.1uF
U1
7805
VCC
VOUT
GND
C3
1000uF
C4
0.1uF
R1
2.2k
D1
LED
AT89C52
Reset
XTAL2
XTAL1
11.0592MHz
C8
30pF
C9
30pF
C6
10uF
R2
10k
D2
1N4148
RP1
10k x 8
Port0
LCD Module
RS
EN
R/W
VDD
GND
R3
2.2k
C7
10uF
C10
10uF
C11
10uF
C12
10uF
C14
10uF
MAX 232
DB-9
Module RF
Title
Remote control
Size
B
Number
Revision
Date:
10-Mar-2006
Sheet of
File:
C:\Program Files\Design Explorer 99 SE\Examples\Circuit Simulation\Remote1.DDB

Master Program

```
;       Computer Interface
;       8 Relay Output
;       8 Switch Input
;       Master.PCB
;

SPAC:   DS      36
MENU:   DS      1

KLOC    BIT     00H

        ORG     0000H
        LJMP    INIT

        ORG     0050H
INIT:   MOV     P0,#0FFH                ; Initial Port
        MOV     P1,#0FFH
        MOV     P2,#0FFH
        MOV     P3,#0FFH

        MOV     R0,#00
        MOV     R1,#00
        MOV     R2,#00
        MOV     R3,#00
        MOV     R4,#00
        MOV     R5,#00
        MOV     R6,#00
        MOV     R7,#00

        CLR     KLOC

        MOV     SCON,#50H               ; Serial Mode 1
        MOV     TMOD,#20H               ; Timer 1 8 Bit Auto Reload
        MOV     TH1,#0FDH               ; 9600 Bps
        MOV     TL1,#0FDH
        SETB    TR1                     ; Start Timer
        CLR     RI                      ; Clear Receive Bit
        CLR     TI                      ; Clear Transmitt Bit

MAIN:   LCALL   KEYB                    ; Check Switch

        LCALL   COMM                    ; Computer Control

        LJMP    MAIN


;--->   Switch  ------------------------------------------------------

KEYB:   MOV     P3,#0FFH                ; Pull Up
        CLR     P3.2                    ; Key 1-4 Enable
        CLR     P3.3                    ; Key 5-8 Enable
        MOV     A,P3                    ; Check Key On
        ORL     A,#00001111B            ;
        CJNE    A,#0FFH,KEON            ;
        SETB    P3.2
        SETB    P3.3
        CLR     KLOC
JEKEY:  LJMP    EKEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KEON:   JB      KLOC,JEKEY              ; Key Hold
        SETB    KLOC

        MOV     R6,#00
KEDL:   LCALL   DLAY                    ; Debounce
        DJNZ    R6,KEDL

        CLR     P3.2
        SETB    P3.3
        JNB     P3.4,JKE10
        JNB     P3.5,JKE20
        JNB     P3.6,JKE30
        JNB     P3.7,JKE40

        SETB    P3.2
        CLR     P3.3
        JNB     P3.4,JKE50
        JNB     P3.5,JKE60
        JNB     P3.6,JKE70
        JNB     P3.7,JKE80
        LJMP    EKEY

JKE10:  LJMP    KE10
JKE20:  LJMP    KE20
JKE30:  LJMP    KE30
JKE40:  LJMP    KE40
JKE50:  LJMP    KE50
JKE60:  LJMP    KE60
JKE70:  LJMP    KE70
JKE80:  LJMP    KE80

KE10:   CPL     P0.0                    ; Relay 1
        JNB     P0.0,KE11
        MOV     SBUF,#'a'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE11:   MOV     SBUF,#'A'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE20:   CPL     P0.1                    ; Relay 2
        JNB     P0.1,KE21
        MOV     SBUF,#'b'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE21:   MOV     SBUF,#'B'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE30:   CPL     P0.2                    ; Relay 3
        JNB     P0.2,KE31
        MOV     SBUF,#'c'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE31:   MOV     SBUF,#'C'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE40:   CPL     P0.3                    ; Relay 4
        JNB     P0.3,KE41
        MOV     SBUF,#'d'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE41:   MOV     SBUF,#'D'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE50:   CPL     P0.4                    ; Relay 5
        JNB     P0.4,KE51
        MOV     SBUF,#'e'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE51:   MOV     SBUF,#'E'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE60:   CPL     P0.5                    ; Relay 6
        JNB     P0.5,KE61
        MOV     SBUF,#'f'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE61:   MOV     SBUF,#'F'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE70:   CPL     P0.6                    ; Relay 7
        JNB     P0.6,KE71
        MOV     SBUF,#'g'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE71:   MOV     SBUF,#'G'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

KE80:   CPL     P0.7                    ; Relay 8
        JNB     P0.7,KE81
        MOV     SBUF,#'h'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY
KE81:   MOV     SBUF,#'H'
        JNB     TI,$
        CLR     TI
        LJMP    EKEY

EKEY:   NOP
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;--->   Computer Control  ------------------------------------------
JECOM:  LJMP    ECOM

COMM:   JNB     RI,JECOM                ; Check Data from Computer
        CLR     RI
        MOV     A,SBUF                  ; Read Data

CM10:   CJNE    A,#'a',CM11
        SETB    P0.0                    ; Off Relay 1
        RET
CM11:   CJNE    A,#'A',CM20
        CLR     P0.0                    ; On Relay 1
        RET

CM20:   CJNE    A,#'b',CM21
        SETB    P0.1                    ; Off Relay 2
        RET
CM21:   CJNE    A,#'B',CM30
        CLR     P0.1                    ; On Relay 2
        RET

CM30:   CJNE    A,#'c',CM31
        SETB    P0.2                    ; Off Relay 3
        RET
CM31:   CJNE    A,#'C',CM40
        CLR     P0.2                    ; On Relay 3
        RET

CM40:   CJNE    A,#'d',CM41
        SETB    P0.3                    ; Off Relay 4
        RET
CM41:   CJNE    A,#'D',CM50
        CLR     P0.3                    ; On Relay 4
        RET

CM50:   CJNE    A,#'e',CM51
        SETB    P0.4                    ; Off Relay 5
        RET
CM51:   CJNE    A,#'E',CM60
        CLR     P0.4                    ; On Relay 5
        RET

CM60:   CJNE    A,#'f',CM61
        SETB    P0.5                    ; Off Relay 6
        RET
CM61:   CJNE    A,#'F',CM70
        CLR     P0.5                    ; On Relay 6
        RET

CM70:   CJNE    A,#'g',CM71
        SETB    P0.6                    ; Off Relay 7
        RET
CM71:   CJNE    A,#'G',CM80
        CLR     P0.6                    ; On Relay 7
        RET

CM80:   CJNE    A,#'h',CM81
        SETB    P0.7                    ; Off Relay 8
        RET
CM81:   CJNE    A,#'H',CM90
        CLR     P0.7                    ; On Relay 8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
CM90:   CJNE    A,#'X',CM91
        ;MOV     P0,#0FFH                ; All Relay Off
        SETB    P0.1
        SETB    P0.2
        SETB    P0.3
        SETB    P0.4
        SETB    P0.5
        SETB    P0.6
        SETB    P0.7
        RET
CM91:   CJNE    A,#'Y',CMA0
        ;MOV     P0,#00                  ; All Relay On
        CLR     P0.1
        CLR     P0.2
        CLR     P0.3
        CLR     P0.4
        CLR     P0.5
        CLR     P0.6
        CLR     P0.7
        RET

JENS:   LJMP    ECOM

CMA0:   CJNE    A,#'S',JENS             ; Read Status

        JNB     P0.0,CMA1               ; Stat 1
        MOV     SBUF,#'a'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    CMB0
CMA1:   MOV     SBUF,#'A'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

CMB0:   JNB     P0.1,CMB1               ; Stat 2
        MOV     SBUF,#'b'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    CMC0
CMB1:   MOV     SBUF,#'B'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

CMC0:   JNB     P0.2,CMC1               ; Stat 3
        MOV     SBUF,#'c'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    CMD0
CMC1:   MOV     SBUF,#'C'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
CMD0:   JNB     P0.3,CMD1               ; Stat 4
        MOV     SBUF,#'d'
        JNB     TI,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR     TI
        LCALL   DLAY
        LJMP    CME0
CMD1:   MOV     SBUF,#'D'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

CME0:   JNB     P0.4,CME1               ; Stat 5
        MOV     SBUF,#'e'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    CMF0
CME1:   MOV     SBUF,#'E'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

CMF0:   JNB     P0.5,CMF1               ; Stat 6
        MOV     SBUF,#'f'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    CMG0
CMF1:   MOV     SBUF,#'F'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

CMG0:   JNB     P0.6,CMG1               ; Stat 7
        MOV     SBUF,#'g'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    CMH0
CMG1:   MOV     SBUF,#'G'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

CMH0:   JNB     P0.7,CMH1               ; Stat 8
        MOV     SBUF,#'h'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        LJMP    ECOM
CMH1:   MOV     SBUF,#'H'
        JNB     TI,$
        CLR     TI
        LCALL   DLAY

ECOM:   NOP
        RET

DLAY:   MOV     R7,#250                 ; Delay
        DJNZ    R7,$
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Remote Control Program

```
;
;       Terminal Set Up
;       Remote .PCB
;

SPAC:   DS      30H

DP01:   DS      1
DP02:   DS      1
DP03:   DS      1
DP04:   DS      1
DP05:   DS      1
DP06:   DS      1
DP07:   DS      1
DP08:   DS      1

CT00:   DS      1
CT01:   DS      1
MENU:   DS      1

KLOC    BIT     00H
CLON    BIT     01H

        ORG     0000H
        LJMP    INIT

        ORG     000BH
        LJMP    TIME

        ORG     0030H
INIT:   MOV     P0,#0FFH
        MOV     P2,#0FFH
        MOV     P1,#0FFH
        MOV     P3,#0FFH
        MOV     R0,#00                  ; GP
        MOV     R1,#00                  ; GP
        MOV     R2,#00                  ;
        MOV     R3,#00                  ;
        MOV     R4,#00                  ; Hold
        MOV     R5,#00                  ; Delay
        MOV     R6,#00                  ; Delay
        MOV     R7,#00                  ; Delay

        MOV     DP01,#00
        MOV     DP02,#00
        MOV     DP03,#00
        MOV     DP04,#00
        MOV     DP05,#00

        MOV     CT00,#40
        MOV     CT01,#100
        MOV     MENU,#00
        CLR     KLOC
        CLR     CLON

        MOV     TMOD,#22H               ; Timer 0-1 8 Bit Auto Reload
        MOV     TH0,#256-250            ; Timer 0
        MOV     TL0,#256-250            ;
        CLR     TCON.4                  ; Stop Timer
        SETB    IE.1                    ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB    IE.7                    ;

        MOV     SCON,#50H               ; Serial Mode 1
        MOV     TMOD,#22H               ; Timer 0-1 8 Bit Auto Reload
        MOV     TH1,#0E8H               ; 1200 Bps
        MOV     TL1,#0E8H
        SETB    TR1                     ; Start Timer
        CLR     RI                      ; Clear Receive Bit
        CLR     TI                      ; Clear Transmit Bit

        LCALL   INLD

;--->   Main Loop  ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
---

MAIN:   NOP                             ; Main Loop

MN00:   MOV     SBUF,#'S'               ; Request Status
        JNB     TI,$
        CLR     TI
        CLR     RI

        MOV     R7,#00                  ; Hold
MN11:   JNB     RI,MN12                 ; Data 1
        LJMP    MN13
MN12:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN11                 ;
        LJMP    MNA1
MN13:   CLR     RI
        MOV     DP01,SBUF

        MOV     R7,#00                  ; Hold
MN21:   JNB     RI,MN22                 ; Data 2
        LJMP    MN23
MN22:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN21                 ;
        LJMP    MNA1
MN23:   CLR     RI
        MOV     DP02,SBUF

        MOV     R7,#00                  ; Hold
MN31:   JNB     RI,MN32                 ; Data 3
        LJMP    MN33
MN32:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN31                 ;
        LJMP    MNA1
MN33:   CLR     RI
        MOV     DP03,SBUF

        MOV     R7,#00                  ; Hold
MN41:   JNB     RI,MN42                 ; Data 4
        LJMP    MN43
MN42:   NOP
        MOV     R6,#50
        DJNZ    R6,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ    R7,MN41                 ;
        LJMP    MNA1
MN43:   CLR     RI
        MOV     DP04,SBUF

        MOV     R7,#00                  ; Hold
MN51:   JNB     RI,MN52                 ; Data 5
        LJMP    MN53
MN52:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN51                 ;
        LJMP    MNA1
MN53:   CLR     RI
        MOV     DP05,SBUF

        MOV     R7,#00                  ; Hold
MN61:   JNB     RI,MN62                 ; Data 6
        LJMP    MN63
MN62:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN61                 ;
        LJMP    MNA1
MN63:   CLR     RI
        MOV     DP06,SBUF

        MOV     R7,#00                  ; Hold
MN71:   JNB     RI,MN72                 ; Data 7
        LJMP    MN73
MN72:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN71                 ;
        LJMP    MNA1
MN73:   CLR     RI
        MOV     DP07,SBUF

        MOV     R7,#00                  ; Hold
MN81:   JNB     RI,MN82                 ; Data 8
        LJMP    MN83
MN82:   NOP
        MOV     R6,#50
        DJNZ    R6,$
        DJNZ    R7,MN81                 ;
        LJMP    MNA1
MN83:   CLR     RI
        MOV     DP08,SBUF

        LCALL   DISP
        MOV     R4,#24
        LJMP    KEYS

MNA1:   DJNZ    R4,EMNA
        MOV     R4,#24
        MOV     DPTR,#ER01              ; Display Error
        LCALL   LCD0
        LCALL   HLAY
EMNA:   LJMP    KEYS

ER01:   DB      '>- Link  Wait -<'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
KEYS:   MOV     P1,#0FFH                ; Key

        JNB     P1.0,KEON
        JNB     P1.1,KEON
        JNB     P1.2,KEON
        JNB     P1.3,KEON
        JNB     P1.4,KEON
        JNB     P1.5,KEON
        JNB     P1.6,KEON
        JNB     P1.7,KEON

        CLR     KLOC
JEKEY:  LJMP    EKEY

KEON:   JB      KLOC,JEKEY
        SETB    KLOC

K000:   JB      P1.0,K001
        LJMP    KEY1
K001:   JB      P1.1,K002
        LJMP    KEY2
K002:   JB      P1.3,K003
        LJMP    KEY3
K003:   JB      P1.2,K004
        LJMP    KEY4
K004:   JB      P1.6,K005
        LJMP    KEY5
K005:   JB      P1.7,K006
        LJMP    KEY6
K006:   JB      P1.4,K007
        LJMP    KEY7
K007:   JB      P1.5,K008
        LJMP    KEY8
K008:   NOP
EKEY:   LJMP    MAIN

KEY1:   NOP
        LJMP    MAIN

KEY2:   NOP
        INC     MENU
        MOV     A,MENU
        CJNE    A,#10,EKE2
        MOV     MENU,#00
EKE2:   LJMP    MAIN

KEY3:   NOP
        LJMP    MAIN

KEY4:   NOP
        MOV     A,MENU

KE41:   CJNE    A,#01,KE42
        MOV     SBUF,#'A'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE42:   CJNE    A,#02,KE43
        MOV     SBUF,#'B'
        JNB     TI,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CLR     TI
        LJMP    EKE4

KE43:   CJNE    A,#03,KE44
        MOV     SBUF,#'C'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE44:   CJNE    A,#04,KE45
        MOV     SBUF,#'D'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE45:   CJNE    A,#05,KE46
        MOV     SBUF,#'E'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE46:   CJNE    A,#06,KE47
        MOV     SBUF,#'F'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE47:   CJNE    A,#07,KE48
        MOV     SBUF,#'G'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE48:   CJNE    A,#08,KE49
        MOV     SBUF,#'H'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

KE49:   CJNE    A,#09,EKE4
        MOV     SBUF,#'Y'
        JNB     TI,$
        CLR     TI
        LJMP    EKE4

EKE4:   LJMP    MAIN

KEY5:   NOP
        LJMP    MAIN

KEY6:   NOP
        MOV     A,MENU

KE61:   CJNE    A,#01,KE62
        MOV     SBUF,#'a'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE62:   CJNE    A,#02,KE63
        MOV     SBUF,#'b'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE63:   CJNE    A,#03,KE64
        MOV     SBUF,#'c'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE64:   CJNE    A,#04,KE65
        MOV     SBUF,#'d'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE65:   CJNE    A,#05,KE66
        MOV     SBUF,#'e'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE66:   CJNE    A,#06,KE67
        MOV     SBUF,#'f'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE67:   CJNE    A,#07,KE68
        MOV     SBUF,#'g'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE68:   CJNE    A,#08,KE69
        MOV     SBUF,#'h'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

KE69:   CJNE    A,#09,EKE6
        MOV     SBUF,#'X'
        JNB     TI,$
        CLR     TI
        LJMP    EKE6

EKE6:   LJMP    MAIN

KEY7:   NOP
        LJMP    MAIN

KEY8:   NOP
        MOV     MENU,#00
EKE8:   LJMP    MAIN


;--->   Display Scan  ----------------------------------------------
---

DISP:   NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,MENU
        JNZ     D000

        MOV     DPTR,#ACTV              ; Display Error
        LCALL   LCD0
        LCALL   HLAY
        LJMP    EDIP

ACTV:   DB      '# Module Ready #'

D000:   CJNE    A,#09,D001

        MOV     DPTR,#ALLC              ; Display Error
        LCALL   LCD0
        LCALL   HLAY
        LJMP    EDIP

ALLC:   DB      'Control  # ALL #'

D001:   MOV     A,#80H                  ; LCD Address
        LCALL   LCDX

        MOV     A,#'C'                  ; CH : X  =>  XXXX
        LCALL   WRCH                    ;
        MOV     A,#'H'                  ;
        LCALL   WRCH
        MOV     A,#' '                  ;
        LCALL   WRCH
        MOV     A,#':'                  ;
        LCALL   WRCH
        MOV     A,#' '                  ;
        LCALL   WRCH

        MOV     A,MENU                  ;
        ORL     A,#00110000B
        LCALL   WRCH

        MOV     A,#' '                  ;
        LCALL   WRCH
        MOV     A,#' '                  ;
        LCALL   WRCH

        MOV     A,#0C0H                 ; LCD Address
        LCALL   LCDX

        MOV     A,#'='                  ;
        LCALL   WRCH
        MOV     A,#'>'                  ;
        LCALL   WRCH
        MOV     A,#' '                  ;
        LCALL   WRCH
        MOV     A,#' '                  ;
        LCALL   WRCH

        MOV     A,MENU

D010:   CJNE    A,#01,D020
        MOV     A,DP01
        CJNE    A,#'A',D011
        LJMP    D100
D011:   LJMP    D200
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
D020:   CJNE    A,#02,D030
        MOV     A,DP02
        CJNE    A,#'B',D021
        LJMP    D100
D021:   LJMP    D200

D030:   CJNE    A,#03,D040
        MOV     A,DP03
        CJNE    A,#'C',D031
        LJMP    D100
D031:   LJMP    D200

D040:   CJNE    A,#04,D050
        MOV     A,DP04
        CJNE    A,#'D',D041
        LJMP    D100
D041:   LJMP    D200

D050:   CJNE    A,#05,D060
        MOV     A,DP05
        CJNE    A,#'E',D051
        LJMP    D100
D051:   LJMP    D200

D060:   CJNE    A,#06,D070
        MOV     A,DP06
        CJNE    A,#'F',D061
        LJMP    D100
D061:   LJMP    D200

D070:   CJNE    A,#07,D080
        MOV     A,DP07
        CJNE    A,#'G',D071
        LJMP    D100
D071:   LJMP    D200

D080:   CJNE    A,#08,EDIP
        MOV     A,DP08
        CJNE    A,#'H',D081
        LJMP    D100
D081:   LJMP    D200

D100:   MOV     A,#'.'                  ;
        LCALL   WRCH
        MOV     A,#'.'                  ;
        LCALL   WRCH
        MOV     A,#'O'                  ;
        LCALL   WRCH
        MOV     A,#'N'                  ;
        LCALL   WRCH
        LCALL   HLAY
        LJMP    EDIP

D200:   MOV     A,#'.'                  ;
        LCALL   WRCH
        MOV     A,#'O'                  ;
        LCALL   WRCH
        MOV     A,#'F'                  ;
        LCALL   WRCH
        MOV     A,#'F'                  ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   WRCH
        LCALL   HLAY

EDIP:   RET


;--->   Timer 0 Interrupt  ----------------------------------------
---

JETIM:  LJMP    ETIM

TIME:   DJNZ    CT00,JETIM              ; 0.01 Sec
        MOV     CT00,#40
        DJNZ    CT01,JETIM              ; 1 Sec
        MOV     CT01,#100

        CPL     CLON

ETIM:   RETI


;--->   LCD Activity  ---------------------------------------------
---

LCDP    EQU     P0                      ; LCD Data Port
LCRS    EQU     P3.7                    ; LCD RS
LCEN    EQU     P3.6                    ; LCD Chip Select

INLD:   CLR     LCRS                    ; Initial LCD
        MOV     LCDP,#00111000B         ; 8 Bit 1/16 Duty
        LCALL   WRT
        MOV     LCDP,#00001100B         ; Display On
        LCALL   WRT
        MOV     LCDP,#00000001B         ; Clear Display
        LCALL   WRT
        LCALL   HHLY                    ; Delay

        MOV     DPTR,#LGO1              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        LCALL   HHLY
        MOV     DPTR,#LGO2              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        LCALL   HHLY
        MOV     DPTR,#LGO3              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        LCALL   HHLY
        MOV     DPTR,#LGO4              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        LCALL   HHLY
        MOV     DPTR,#LGO5              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        RET

LCD0:   MOV     LCDP,#80H               ; 16 x 1
        CLR     LCRS
        LCALL   WRT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL   WDAT
        MOV     LCDP,#0C0H
        CLR     LCRS
        LCALL   WRT
        LCALL   WDAT
        RET

LCD1:   MOV     LCDP,#80H
        CLR     LCRS
        LCALL   WRT
        LCALL   WDAT
        RET

LCD2:   MOV     LCDP,#0C0H
        CLR     LCRS
        LCALL   WRT
        LCALL   WDAT
        RET

LCDX:   MOV     LCDP,A
        CLR     LCRS
        LCALL   WRT
        RET

WDAT:   MOV     R0,#08
WDT1:   MOV     A,#00
        MOVC    A,@A+DPTR
        MOV     LCDP,A
        SETB    LCRS
        LCALL   WRT
        INC     DPTR
        DJNZ    R0,WDT1
        MOV     LCDP,#0FFH
        RET

WR3B:   MOV     R1,#03
WR31:   MOV     A,R0
        MOVC    A,@A+DPTR
        MOV     LCDP,A
        SETB    LCRS
        LCALL   WRT
        INC     DPTR
        DJNZ    R1,WR31
        MOV     LCDP,#0FFH
        RET

WRBT:   MOVC    A,@A+DPTR
WRCH:   MOV     LCDP,A
        SETB    LCRS
        LCALL   WRT
        MOV     LCDP,#0FFH
        RET

WRT:    CLR     LCEN
        MOV     R7,#00
        DJNZ    R7,$
        SETB    LCEN
        MOV     R7,#00
        DJNZ    R7,$
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

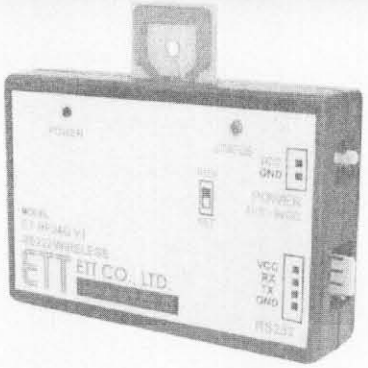
```
LGO1:   DB      ' Initial System '
LGO2:   DB      '................'
LGO3:   DB      '                '
LGO4:   DB      '>---  PASS  ---<'
LGO5:   DB      '                '


DLAY:   MOV     R7,#00
        DJNZ    R7,$
        RET

HLAY:   MOV     R6,#00
HL00:   MOV     R7,#00
        DJNZ    R7,$
        DJNZ    R6,HL00
        RET

HHLY:   MOV     R5,#03
HH00:   MOV     R6,#00
HH01:   MOV     R7,#00
        DJNZ    R7,$
        DJNZ    R6,HH01
        DJNZ    R5,HH00
        RET

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ET-RF24G V1.0



ลักษณะโดยทั่วไป

ET-RF24G V1.0 เป็นชุด Signal Converter สำหรับใช้แปลงสัญญาณระหว่าง RS232 และ RF-Wireless โดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำหน้าที่รอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232 จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ชุด ET-RF24G V1.0 ก็จะทำหน้าที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่ (GFSK) จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX ได้ด้วย

ซึ่งจะเห็นได้ว่าชุดแปลงสัญญาณ ET-RF24G V1.0 นั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สาย (Wireless Transceiver) ได้โดยตรง โดยจะมีข้อดีกว่า คือ สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลกว่า RS232 หลายเท่าตัว และประการสำคัญ คือ ไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวนำสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกัน ทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับส่งข้อมูลได้ตลอดเวลา ซึ่งถ้าเป็นการรับส่งข้อมูลด้วยระบบ RS232 แบบที่ใช้สายสัญญาณนั้นจะเกิดความยุ่งยากในการติดตั้งสายสัญญาณเป็นอย่างมาก

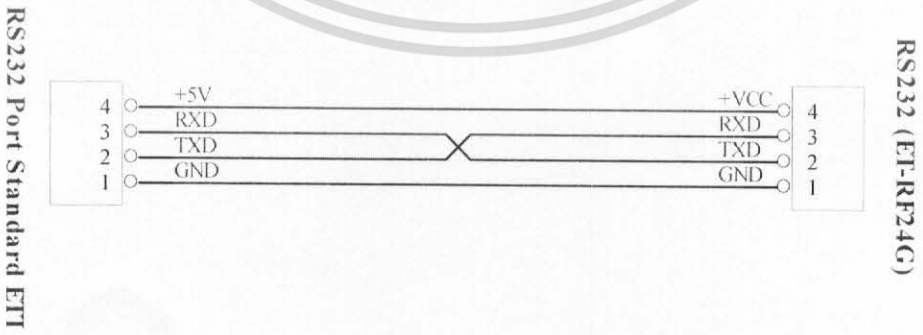
แต่อย่างไรก็ตามการรับส่งข้อมูลโดยใช้อากาศเป็นตัวกลางในการสื่อสารนั้น ก็มีข้อจำกัดบางประการเหมือนกัน โดยเฉพาะอย่างยิ่ง เรื่องความน่าเชื่อถือของข้อมูลที่รับส่งกัน ซึ่งมีโอกาสผิดพลาดหรือสูญหายได้เหมือนกัน เนื่องจากในการลำเลียงข้อมูลนั้นไม่ได้ใช้สายสัญญาณเป็นตัวกลางในการรับส่งข้อมูล แต่ใช้อากาศเป็นตัวกลางในการรับส่งข้อมูลแทน ซึ่งมีโอกาศที่ข้อมูลจะเกิดการรบกวนจากสัญญาณอื่นๆที่มีย่านความถี่ใกล้เคียงกันแล้วทำให้ข้อมูลผิดเพี้ยนไปได้บ้างเหมือนกัน ซึ่งระบบการจัดการข้อมูลของเครื่อง ET-RF24G V1.0 นั้น มีระบบการเข้ารหัสและถอดรหัสข้อมูลที่มีความน่าเชื่อถืออยู่ในเกณฑ์ที่จัดว่าดี โดยข้อมูลแต่ละ Byte ที่มีการรับส่งกันนั้น จะมีการตรวจสอบความถูกต้องของข้อมูลให้ด้วยแล้ว โดยข้อมูลที่รับได้จากด้าน RF นั้นรับประกันได้ว่าเป็นข้อมูลที่มีความถูกต้องแน่นอน แต่อย่างไรก็ตามการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกัน เนื่องจากกลไกในการรับส่งข้อมูลของเครื่อง ET-RF24G V1.0 นั้น จะมีการตรวจสอบข้อมูลทุก Byte ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีความผิดพลาดเกิดขึ้นจะทิ้งข้อมูล Byte นั้นไป ซึ่งผู้ใช้ควรมีกลไกในการตรวจสอบข้อมูลที่รับส่งกันว่าครบถ้วนหรือไม่ด้วย ซึ่งหากพบว่ามีการสูญหายของข้อมูลเกิดขึ้นก็ให้ร้องขอให้มีการส่งข้อมูลนั้นซ้ำนั้นๆใหม่อีกครั้งหนึ่ง ก็จะสามารถแก้ไขปัญหาดังกล่าวได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power Supply

สำหรับการต่อแหล่งจ่ายไฟให้กับเครื่อง ET-RF24G V1.0 นั้น จะสามารถเลือกต่อแหล่งจ่ายไฟให้กับตัวเครื่องได้ 2 ทางด้วยกัน โดยเครื่อง ET-RF24G V1.0 นั้น ต้องการไฟเลี้ยงวงจร ซึ่งเป็นแหล่งจ่ายกระแสตรง ขนาดประมาณ +5VDC ถึง +9VDC โดยจุดเชื่อมต่อแหล่งจ่ายไฟของเครื่อง ET-RF24G V1.0 นี้ สามารถเชื่อมต่อได้ 2 จุดด้วยกัน โดยผู้ใช้สามารถเลือกต่อแหล่งจ่ายไฟให้กับเครื่อง ET-RF24G V1.0 จุดใดจุดหนึ่งก็ได้

ในกรณีที่นำเครื่อง ET-RF24G V1.0 ไปเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์รุ่นต่างๆของ อีทีที นั้น สามารถใช้แหล่งจ่ายไฟจากบอร์ดไมโครคอนโทรลเลอร์ เพื่อจ่ายให้กับตัวเครื่อง ET-RF24G V1.0 ได้ทันที โดยไม่ต้องใช้แหล่งจ่ายไฟจากภายนอก เนื่องจากขั้วต่อสัญญาณ RS232 ของบอร์ดไมโครคอนโทรลเลอร์รุ่นต่างๆ ของบริษัท อีทีที นั้น ได้จัดเตรียมแหล่งจ่ายไฟตรง ขนาด +5V เตรียมไว้ให้ด้วยแล้ว โดยผู้ใช้เพียงแต่นำสายสัญญาณ RS232 ซึ่งทำการต่อสายสัญญาณครบทั้ง 4 เส้น ดังรูปมาเชื่อมต่อก็สามารถใช้งานได้แล้ว



รูปแสดง การต่อสายสัญญาณ RS232 เพื่อใช้แหล่งจ่ายไฟจากบอร์ดไมโครฯ ของ อีทีที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่สำหรับกรณีที่นำเครื่อง ET-RF24G V1.0 ไปต่อใช้งานกับอุปกรณ์อื่นๆที่ไม่ได้มีการจัดเตรียมจุดต่อไฟเลี้ยงไว้ให้ด้วย ผู้ใช้จำเป็นต้องจัดหา Adapter จ่ายไฟจากภายนอกมาต่อให้กับเครื่อง ET-RF24G V1.0 ด้วย โดยให้เลือกแหล่งจ่ายไฟที่มีขนาดแรงดันไฟตรงประมาณ +5VDC และสามารถจ่ายกระแสได้ประมาณ 300mA เป็นอย่างน้อย ซึ่งในกรณีนี้ขอแนะนำให้เลือกใช้ Power Supply รุ่น "ACH-4E" ซึ่งเป็นแหล่งจ่ายไฟแบบ Switching Power ใช้กับไฟบ้าน 220VAC และให้เอ้าพุตเป็นไฟกระแสตรง ขนาดประมาณ 5VDC / 750mA เพราะ Power Supply รุ่นนี้ สามารถใช้งานร่วมกับเครื่อง ET-RF24G V1.0 ได้อย่างต่อเนื่องเป็นเวลานานๆ โดยไม่เกิดความร้อนสะสมที่วงจร Regulate ของบอร์ด ET-RF24G V1.0 มากนัก ซึ่งถ้าผู้ใช้เลือกใช้แหล่งจ่ายไฟรุ่นอื่นๆ ที่มีขนาดแรงดันสูงกว่า +5V มากๆ ซึ่งถึงแม้ว่าจะสามารถใช้งานร่วมกันกับเครื่อง ET-RF24G V1.0 ได้ แต่ถ้ามีการใช้งานอย่างต่อเนื่องเป็นเวลานานๆแล้ว อาจทำให้เกิดความร้อนสะสมที่ตัวไอซี Regulate มากเกินไป จนอาจทำให้ภาค Power ของเครื่อง ET-RF24G V1.0 หยุดจ่ายไฟทำให้เครื่องหยุดทำงานได้

แสดง การต่อ แหล่งจ่ายไฟรุ่น "ACH-4E" จากภายนอกให้กับเครื่อง ET-RF24G V1.0

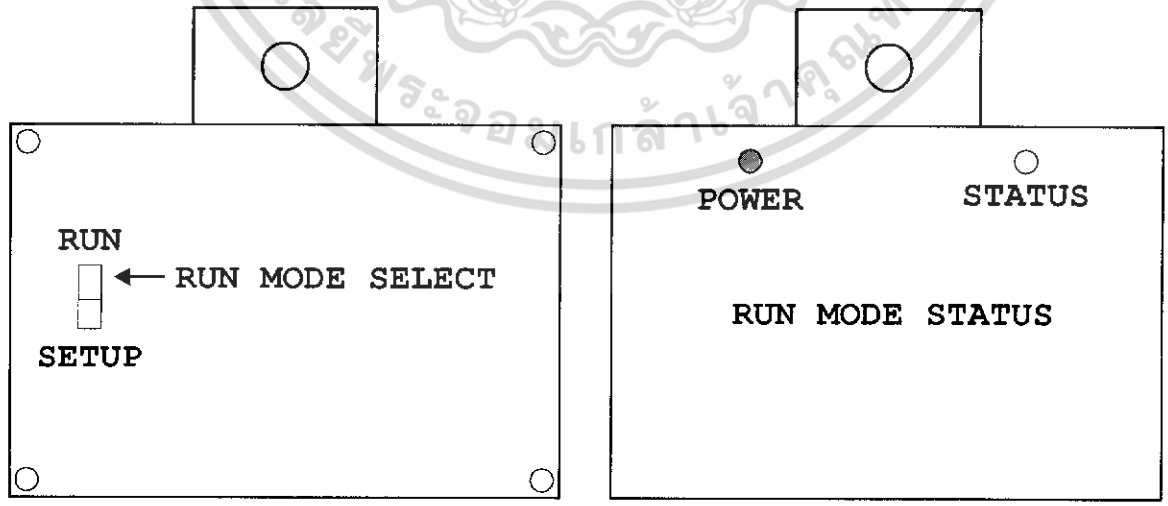
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงาน

สำหรับโหมดการทำงานของ ET-RF24G V1.0 นั้นจะแบ่งออกเป็น 2 โหมด ด้วยกัน โดยการกำหนดโหมดการทำงานของ ET-RF24G V1.0 นั้นจะกระทำผ่าน Switch เลือกโหมด ซึ่งอยู่ด้านใต้กล่อง โดยการเลือกโหมดการทำงานนั้นจะต้องกระทำให้เสร็จเรียบร้อยก่อนการจ่ายไฟให้กับ ET-RF24G V1.0 ด้วยเสมอ เนื่องจากการทำงานของเครื่อง ET-RF24G V1.0 นั้นจะทำการตรวจสอบโหมดการทำงานของเครื่องจาก Switch เลือกโหมด เฉพาะในช่วงของการจ่ายไฟเลี้ยงให้เครื่องเริ่มต้นทำงานครั้งแรก (Power-ON) เท่านั้น ซึ่งการเปลี่ยนแปลงตำแหน่งการทำงานของ Switch เลือกโหมด หลังจากทำการจ่ายไฟให้กับ ET-RF24G V1.0 ไปแล้ว จะไม่มีผลต่อการทำงานของเครื่องแต่อย่างใด โดยการทำงานของเครื่อง ET-RF24G V1.0 นั้นจะมี LED แสดงสถานะการทำงานของเครื่องจำนวน 2 หลอด คือ LED POWER ซึ่งเป็น LED สีแดง โดยที่ LED POWER นี้จะติดสว่างให้เห็นตลอดเวลาที่มีการจ่ายไฟเลี้ยงให้เครื่องทำงานอยู่ ส่วน LED อีกดวงหนึ่งนั้นจะเป็น LED สีเขียว ใช้แสดงสถานะการทำงานของเครื่องซึ่งเรียกว่า LED STATUS โดย LED STATUS นี้จะเกิดการกระพริบตามจังหวะของการรับส่งข้อมูลกันในแต่ละครั้ง โดยในสภาวะปรกตินั้น ถ้าเครื่องทำงานอยู่ใน RUN MODE หลอด LED STATUS จะดับอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล แต่ถ้าตัวเครื่องทำงานอยู่ใน SETUP MODE หลอด LED STATUS จะติดอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล โดยโหมดการทำงานของ ET-RF24G V1.0 จะมีอยู่ด้วยกัน 2 โหมด คือ

1. การใช้งานเครื่อง ET-RF24G V1.0 ใน Run Mode

การใช้งานใน Run Mode ซึ่งเป็นโหมดของการใช้งานตามปรกติของเครื่อง โดยเมื่อเครื่อง ET-RF24G V1.0 เข้าทำงานในโหมดนี้แล้ว จะสังเกตุเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LED STATUS ดับอยู่ แต่เมื่อมีการรับ หรือ ส่ง ข้อมูล เกิดขึ้น สถานะการทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูลนั้นๆ แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะดับอยู่ตลอดเวลา



รูปแสดง การเลือกโหมดการทำงาน สำหรับใช้งานปรกติ (Run Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการทำงานใน Run Mode นั้น จะแบ่งลักษณะการทำงานออกเป็น 3 แบบด้วยกัน โดยลักษณะการทำงานนี้ จะถูกกำหนดไว้แล้วใน Configuration ของเครื่องใน Setup Mode ดังนั้นก่อนการใช้งานเครื่อง ในครั้งแรกจะต้องทำการกำหนดค่า Configuration ต่างๆให้เรียบร้อยเสียก่อน โดยเมื่อเครื่อง ET-RF24G V1.0 เริ่มต้นเข้าทำงานใน Run Mode แล้วมันจะทำการอ่านค่า Configuration ที่เก็บไว้ออกมา เพื่อใช้เป็นเงื่อนไขในการทำงานตามค่าที่ได้กำหนดไว้ โดยลักษณะการทำงานใน Run Mode แบ่งออกเป็นดังนี้

1.1 การทำงานแบบ RF Receive Only

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้ จะเป็นการรอรับข้อมูลความถี่แบบ GFSK จากด้าน RF แล้วเปลี่ยนเป็นข้อมูลอนุกรมส่งออกไปทางขา TX (Transmit) ของ RS232 โดยการทำงานจะวนรอบอยู่เช่นนี้ไปตลอด ซึ่งในการใช้งานเครื่อง ET-RF24G V1.0 ในโหมดนี้จะต้องนำสัญญาณ TX(Transmit) ไปต่อกับขาสัญญาณ RX (Receive) ของอุปกรณ์ด้านตรงข้าม (RS232 ของคอมพิวเตอร์ PC) โดยในโหมดนี้ การทำงานของขาสัญญาณ RX ด้าน RS232 ของเครื่อง ET-RF24G V1.0 จะถูกเปลี่ยนหน้าที่เป็นสัญญาณ CTS (Clear To Send) สำหรับใช้ตรวจสอบความพร้อมในการส่งข้อมูลไปให้อุปกรณ์ด้านตรงข้ามแทน ซึ่งในการใช้งานจะต้องนำสัญญาณนี้ไปต่อเข้ากับสัญญาณ RTS (Ready To Send) ของอุปกรณ์ด้านตรงข้าม โดยเครื่อง ET-RF24G V1.0 จะทำการตรวจสอบสถานะของสัญญาณ RX ซึ่งในโหมดนี้เปรียบเสมือน CTS ว่ามีค่าเป็น "0" หรือไม่ โดยถ้าพบว่าเป็น "0" จึงจะส่งข้อมูลออกไปให้ทางขา TX แต่ถ้าพบว่าสถานะของขาสัญญาณนี้มีค่าเป็น "1" แสดงว่าอุปกรณ์ด้านตรงข้ามยังไม่พร้อมรับข้อมูลก็จะรอจนกว่าจะพบว่าสถานะของสัญญาณดังกล่าวมีค่าเป็น "0" จึงจะส่งข้อมูลออกไปให้ โดยเครื่อง ET-RF24G V1.0 จะสามารถจัดเก็บข้อมูลไว้ใน Buffer เพื่อรอการส่งได้สูงสุด 64 Byte เท่านั้น ซึ่งถ้าในระหว่างที่รอความพร้อมอยู่นั้น มีข้อมูลด้าน RF ส่งเข้ามาเกินกว่า 64 Byte จะทำให้ข้อมูลที่เกินมานั้นสูญหายไป

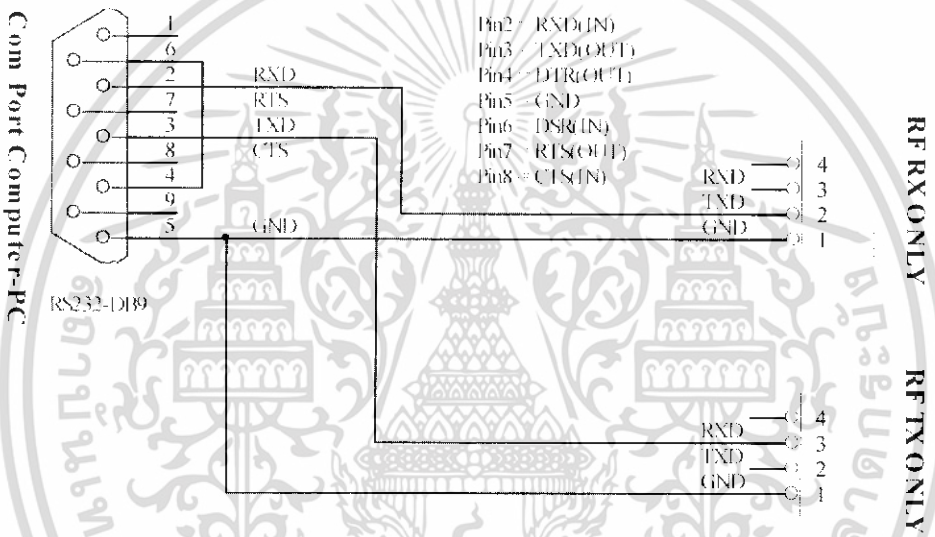
1.2 การทำงานแบบ RF Transmit Only

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้จะมีลักษณะตรงกันข้ามกับ RF Receive Only กล่าวคือ เครื่อง ET-RF24G V1.0 จะทำหน้าที่รอรับข้อมูลจากขา RX (Receive) ด้าน RS232 แล้วเปลี่ยนเป็นข้อมูลแบบ GFSK ส่งออกไปทางด้าน RF โดยการใช้งานเครื่องในโหมดนี้ จะต้องนำสัญญาณ TX (Transmit) ซึ่งเป็นขาส่งข้อมูลจาก RS232 ของอุปกรณ์ด้านตรงข้ามมาต่อเข้ากับขา RX(Receive) ของเครื่อง ET-RF24G V1.0 ส่วนขาสัญญาณ TX จะถูกเปลี่ยนหน้าที่เป็น RTS (Ready To Send) เพื่อใช้แสดงสถานะความพร้อมในการรับข้อมูลจากด้าน RS232 ซึ่งในการใช้งานจะต้องนำสัญญาณ TX ซึ่งในขณะนี้เปรียบเสมือนกับ RTS นำไปต่อเข้ากับสัญญาณ CTS (Clear To Send) ของอุปกรณ์ด้านตรงข้าม เพื่อใช้ในการตรวจสอบความพร้อมในการรับข้อมูล โดยอุปกรณ์ด้านตรงข้ามจะต้องทำการตรวจสอบสถานะของสัญญาณ RTS นี้ เพื่อตรวจสอบความพร้อมในการรับข้อมูลของเครื่อง ET-RF24G V1.0 ด้วย โดยถ้าเครื่อง ET-RF24G V1.0 พร้อมรับข้อมูลจาก RS232 มันจะส่งสัญญาณ RTS ให้มีค่าเป็น "0" รอไว้ และเมื่อใดก็ตามที่การรับข้อมูลทางด้านของ RS232 มีจำนวนข้อมูลที่ยังไม่สามารถเปลี่ยน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น GFSK เพื่อส่งออกไปทางด้าน RF ได้ทันจนเกือบจะเต็ม Buffer แล้ว เครื่อง ET-RF24G V1.0 จะทำการส่งสัญญาณ RTS ให้มีค่าเป็น "1" ออกไปบอกให้อุปกรณ์ด้านตรงข้ามทราบเพื่อจะได้หยุดการส่งข้อมูลออกมา โดยอุปกรณ์ด้านตรงข้ามจะต้องหยุดการส่งข้อมูลและรอจนกว่าสถานะของสัญญาณ RTS จะกลับเป็น "0" จึงจะเริ่มต้นส่งข้อมูลออกมาใหม่ ซึ่งหลังจากที่เครื่อง ET-RF24G V1.0 ส่งสัญญาณ RTS ด้วยค่า "1" ออกไปแล้ว จะยังคงสามารถรับข้อมูลได้เพิ่มเติมอีกไม่เกิน 16 Byte เท่านั้น ซึ่งถ้าอุปกรณ์ด้านตรงข้ามยังส่งข้อมูลต่อเนื่องมาอีกจนเกินขนาดของ Buffer ที่เครื่อง ET-RF24G V1.0 จะรับไว้ได้จะทำให้ข้อมูลที่เกินมานั้นเกิดการสูญหายได้

โดยเราสามารถนำเครื่อง ET-RF24G V1.0 จำนวน 4 ชุด มาต่อใช้งานร่วมกัน เพื่อใช้งานในการรับส่งข้อมูลกันแบบ Full Duplex โดยแบ่งการใช้งานออกเป็น 2 ด้าน คือ ต้นทาง และ ปลายทาง ด้านละ 2 ชุด โดยแต่ละด้านให้กำหนดหน้าที่การทำงานเป็น RF Receive Only 1 ชุด และ RF Transmit Only อีก 1 ชุด



รูปแสดงสายสัญญาณ RS232 เพื่อใช้กับ ET-RF24G ในโหมด RF Receive Only และ RF Transmit Only

1.3 การทำงานแบบ RF Auto Direction

เป็นการทำงานชนิด 2 ทิศทาง แบบ Half Duplex หรือ ผลัดกันรับผลัดกันส่ง ซึ่งสามารถใช้รับส่งข้อมูลระหว่างต้นทาง และ ปลายทาง ได้ โดยใช้เครื่อง ET-RF24G V1.0 ด้านละ 1 ชุด เท่านั้น เพียงแต่การรับส่งข้อมูลแบบนี้จะไม่สามารถส่งข้อมูลสวนทางกันได้เหมือนกับแบบ Full Duplex แต่จะต้องใช้วิธีการผลัดกันรับข้อมูลและส่งข้อมูลแทน โดยเมื่อฝ่ายรับทำการรับข้อมูลได้จนครบแล้วจึงจะสลับหน้าที่เป็นฝ่ายส่งเพื่อส่งข้อมูลย้อนกลับไป

โดยในโหมดนี้ เครื่อง ET-RF24G V1.0 จะทำหน้าที่เป็นทั้ง ฝ่ายรับ และ ฝ่ายส่ง ข้อมูล แบบอัตโนมัติ โดยในสภาวะปรกติจะอยู่ในสภาวะของการรอรับข้อมูล ทั้งด้าน RF และ RS232 ซึ่งถ้าพบว่ามีข้อมูลส่งเข้ามาทางด้านของ RF ก็จะนำข้อมูลนั้นส่งออกไปทางด้านขา TX ของ RS232 ทันที และในทำนองเดียวกัน ถ้าพบว่ามีข้อมูลส่งเข้ามาทางด้าน RX ของ RS232 มันก็จะทำการรับข้อมูลนั้นจาก RS232 พร้อมกับเปลี่ยนทิศทางของอุปกรณ์ RF จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรอรับข้อมูลให้ทำหน้าที่เป็นตัวส่งข้อมูลแทน เพื่อทำการส่งข้อมูลที่รับได้จาก RS232 ออกไปทาง RF ในทันที ซึ่งหลังจากที่เครื่อง ET-RF24G V1.0 ทำการสลับโหมดการทำงานของอุปกรณ์ด้าน RF จากการรอรับเป็นการส่งและทำการเริ่มต้นส่งข้อมูลออกไปทางด้าน RF เรียบร้อยแล้ว มันจะวนกลับไปตรวจสอบการรับข้อมูลจากด้าน RS232 อีกว่ายังมีข้อมูลส่งเข้ามาอีกหรือไม่ ถ้าพบว่ายังมีข้อมูลส่งเข้ามีอีกก็จะทำการแปลงข้อมูลนั้นเพื่อส่งออกไปยังด้าน RF ต่อไปอีกจนกว่าการส่งข้อมูลด้าน RS232 จะสิ้นสุดลง ซึ่งข้อมูลด้าน RS232 ที่ส่งเข้ามานั้น ควรส่งอย่างต่อเนื่อง โดยเมื่อเครื่อง ET-RF24G V1.0 ทำการส่งข้อมูลแต่ละ Byte ออกไปทางด้าน RF เรียบร้อยแล้วมันจะวนรอบรอรับข้อมูล Byte ถัดไปจาก RS232 ภายในเวลา 2.5 mS ถ้าไม่พบข้อมูลส่งเข้ามาอีกภายในระยะเวลาดังกล่าวมันจึงจะทำการเปลี่ยนหน้าที่ของอุปกรณ์ด้าน RF ให้กลับมาทำหน้าที่เป็นการรอรับข้อมูลตามเดิม โดยในขณะที่อุปกรณ์ด้าน RF ถูกกำหนดให้เป็นฝ่ายส่งข้อมูลอยู่นั้น จะไม่สามารถทำการรับข้อมูลจาก RF ได้ ซึ่งถ้ามีการส่งข้อมูลเข้ามาในขณะนั้นก็จะไม่สามารถรับได้ โดยค่าเวลาที่จะใช้ในการสลับโหมดการทำงานของ RF จากฝ่ายส่งข้อมูลให้เป็นฝ่ายรับข้อมูลนั้น จะมีค่าเป็น 2.5mS ดังนั้นเมื่อฝ่ายรับสามารถรับข้อมูลได้ครบหมดแล้วก่อนที่จะทำการส่งข้อมูลเพื่อตอบกลับไปยังฝ่ายตรงข้ามนั้น ควรทำการหน่วงเวลาไว้ไม่น้อยกว่า 3mS นับจากรับข้อมูล Byte สุดท้ายได้เรียบร้อยแล้วจึงเริ่มต้นส่งข้อมูล Byte แรกย้อนกลับไป ซึ่งถ้าฝ่ายรับทำการส่งข้อมูลตอบกลับไปยังฝ่ายตรงข้ามเร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทัน

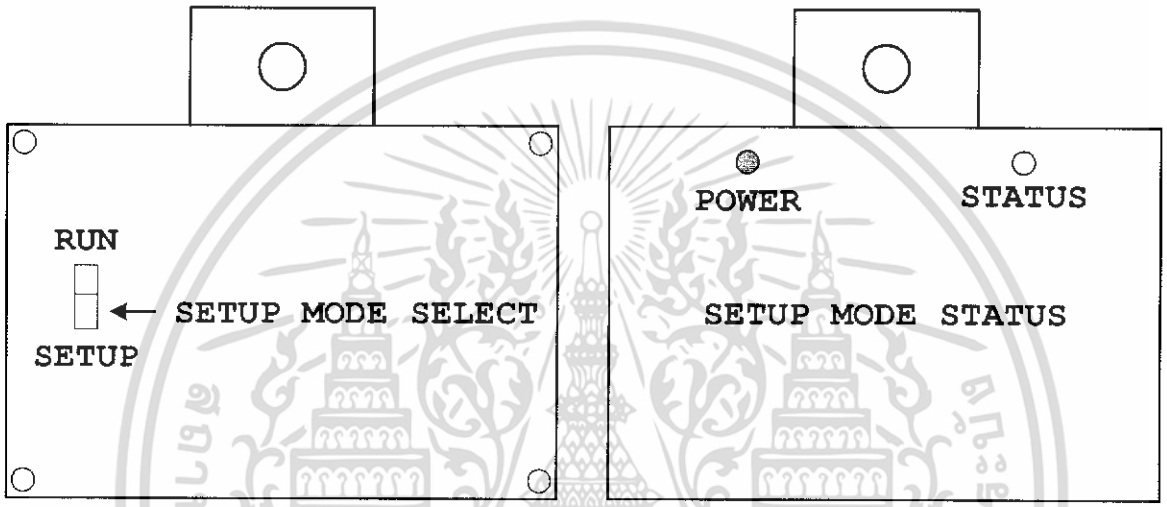
สำหรับการใช้งานเครื่อง ET-RF24G V1.0 ในโหมด RF Auto Direction นี้ การ รับ และ ส่ง ข้อมูล ด้าน RS232 จะไม่มีการตรวจสอบความพร้อมของฝ่ายรับและส่ง ด้วยสัญญาณทางไฟฟ้า (CTS/RTS) เหมือนกับการใช้งานใน 2 โหมดที่ผ่านมาแล้ว โดยเมื่อมันสามารถรับข้อมูลจาก RF ได้ ก็จะทำการส่งข้อมูลนั้นออกไปทางขา TX (Transmit) ของ RS232 ในทันที โดยไม่สนใจว่า อุปกรณ์ที่ต่อไว้ด้าน RS232 จะพร้อมรับข้อมูลหรือไม่ ซึ่งถ้าด้าน RS232 ไม่พร้อมรับข้อมูลก็จะทำให้ข้อมูล Byte นั้นสูญหายไปทันที ซึ่งในการใช้งานนั้น ผู้ใช้ควรกำหนดค่าความเร็วในการรับส่งข้อมูลด้าน RS232 ที่จะใช้กับเครื่อง ET-RF24G V1.0 ทุกๆตัวด้วยค่าความเร็วที่เท่ากันด้วย เพื่อให้การรับและส่งข้อมูลเกิดความสัมพันธ์กันอย่างเหมาะสม

สำหรับความสามารถในการรอรับข้อมูลจาก RS232 ของเครื่อง ET-RF24G V1.0 ในโหมดนี้ จะสามารถรับข้อมูลได้อย่างต่อเนื่องสูงสุด ไม่เกิน 64 Byte ดังนั้นในกรณีที่มีการส่งข้อมูลจากด้าน RS232 ด้วยข้อมูลจำนวนมากกว่า 64 Byte ต่อเนื่องกันนั้น ควรทำการแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดชุดละไม่เกิน 64 Byte ซึ่งหลังจากทำการส่งข้อมูลอย่างต่อเนื่องไปได้ 1 ชุด (64 Byte) แล้วควรทำการหน่วงเวลาไว้ชั่วขณะหนึ่งอย่างน้อย 1mS แล้วจึงเริ่มส่งข้อมูลชุดถัดไป สลับกับการหน่วงเวลา อย่างนี้เรื่อยๆ เพื่อให้เครื่อง ET-RF24G V1.0 สามารถนำข้อมูลที่รับได้จากด้าน RS232 ส่งออกไปทางด้าน RF ได้ทัน ซึ่งถ้าทำการส่งข้อมูลอย่างต่อเนื่องโดยไม่มีการหน่วงเวลาเลยอาจทำให้ข้อมูลบาง Byte เกิดการสูญหายไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode

การใช้งานเครื่อง ET-RF24G V1.0 ใน Setup Mode ซึ่งเป็นโหมดสำหรับใช้กำหนดค่า Configuration ต่างๆ สำหรับควบคุมการทำงานของเครื่อง ET-RF24G V1.0 ที่จะใช้ในขณะที่เครื่องทำงานอยู่ใน Run Mode โดยในการ Setup ค่า Configuration ต่างๆนั้นจะกระทำร่วมกับโปรแกรม "ET_RF24G_V1.EXE" ของ อีทีที ซึ่งเมื่อเครื่อง ET-RF24G V1.0 เข้าทำงานในโหมด Setup แล้ว จะสังเกตุเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LED STATUS ติดสว่างค้างอยู่ตลอดเวลา แต่เมื่อมีการสั่งอ่านหรือเขียนข้อมูลกับบอร์ด สถานะการทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูล แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะติดค้างอยู่ตลอดเวลา

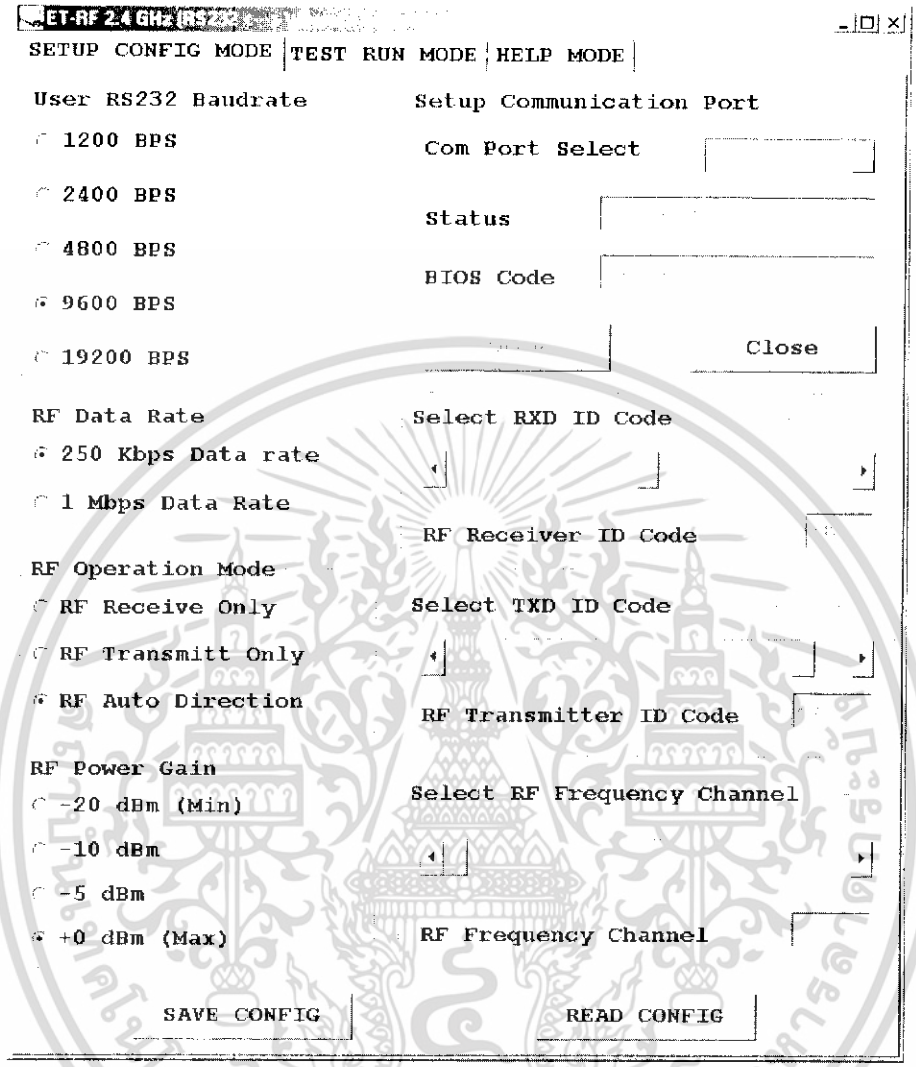


รูปแสดง การเลือกโหมดการทำงาน สำหรับกำหนดค่า Configuration (Setup Mode)

ซึ่งการกำหนดค่า Configuration ให้กับ ET-RF24G V1.0 นั้น จะต้องกระทำในขณะที่ตัวเครื่องทำงานอยู่ใน Setup Mode เท่านั้น (เลือก Switch กำหนดโหมดไว้ทางด้าน Setup แล้วจ่ายไฟให้เครื่องเริ่มต้นทำงาน) โดยค่าของ Configuration ต่างๆนั้นจะถูกใช้สำหรับเป็นเงื่อนไขในการทำงานของ ET-RF24G V1.0 ในขณะที่อยู่ใน Run Mode ดังนั้น ก่อนการเริ่มต้นใช้งานเครื่องในครั้งแรกนั้น จึงจำเป็นอย่างยิ่งที่จะต้องทำการกำหนดค่าของ Configuration ต่างๆให้ถูกต้องและตรงกับความต้องการที่จะใช้งานเสียก่อน โดยเมื่อทำการกำหนดค่าตัวเลือกต่างๆของ Configuration เรียบร้อยแล้ว ก็สามารถเปลี่ยนโหมดการทำงานของตัวเครื่องกลับเป็น Run Mode พร้อมกับการปิดไฟที่จ่ายให้กับตัวเครื่อง (Power-OFF) ชั่วขณะหนึ่ง จากนั้นจึงเริ่มต้นจ่ายไฟให้กับตัวเครื่องใหม่ (Power-ON) ก็สามารถใช้งาน ET-RF24G V1.0 ตามค่าของ Configuration ที่กำหนดไว้แล้วได้ทันที โดยค่าตัวเลือกต่างๆของ Configuration ที่ได้กำหนดไว้แล้วจะถูกเก็บไว้ภายในตัวเครื่องอย่างถาวร ถึงแม้ว่าจะไม่ได้ทำการจ่ายไฟให้กับตัวเครื่องแล้วก็ตาม ดังนั้นเมื่อทำการกำหนดค่า Configuration ต่างๆเรียบร้อยแล้ว ถ้าไม่มีการเปลี่ยนแปลงเงื่อนไขการทำงานของตัวเครื่องต่างไปจากเงื่อนไขเดิมที่ได้กำหนดไว้แล้ว ก็ไม่จำเป็นต้องทำการกำหนดค่า Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ในเชิงการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใหม่อีกแต่อย่างใด โดยทุกๆครั้งที่เริ่มต้นจ่ายไฟเข้าเครื่องในครั้งแรกนั้น การทำงานของ ET-RF24G V1.0 จะเป็นไปตามเงื่อนไขที่กำหนดไว้ใน Configuration เสมอทุกๆครั้ง โดยคุณสมบัติของ Configuration ต่างๆนั้นมีดังนี้



แสดง รูปโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V1.0

- User RS232 Baudrate ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้าน RS232 ของตัวเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ
  - 1200 BPS
  - 2400 BPS
  - 4800 BPS
  - 9600 BPS
  - 19200 BPS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RF Data Rate ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V1.0 ซึ่งจะต้องกำหนดให้เครื่อง ET-RF24G V1.0 ทุกๆตัว ที่จะนำมาใช้ติดต่อสื่อสารกัน มีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมด ซึ่งถ้ากำหนดค่าความเร็วต่างกันจะไม่สามารถรับส่งข้อมูลกันได้ ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูลด้วย ซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF Data Rate สามารถกำหนดได้ 2 ค่า คือ
  - 250 Kbps
  - 1 Mbps

- RF Operation Mode ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V1.0 ซึ่งสามารถกำหนดหน้าที่การทำงานได้ 3 แบบ ด้วยกันคือ
  - RF Receive Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา
  - RF Transmit Only เป็นการกำหนดให้ ET-RF24G V1.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา
  - RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้เครื่อง ET-RF24G V1.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่ตลอดเวลา โดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RF จากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS232 โดยอัตโนมัติ

- RF Power Gain เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBm เป็นค่ากำลังส่งสูงสุด ส่วน –20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับคือ
  - -20dBm (กำลังส่งต่ำสุด)
  - -10dBm
  - -5dBm
  - -0dBm (กำลังส่งสูงสุด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RXD ID Code เป็นรหัส ID Code ของเครื่อง ET-RF24G V1.0 ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่อง ET-RF24G V1.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่อง ET-RF24G V1.0 ที่อยู่ทางด้านรับทำการรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอาเฉพาะส่วนของข้อมูลที่รับเข้ามาได้เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของ RS232 แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่อง ET-RF24G V1.0 จะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

- TXD ID Code เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่อง ET-RF24G V1.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหานั่นเอง โดยค่า TXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

- RF Frequency Channel เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่อง ET-RF24G V1.0 จะทำการรับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกัน และ ใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency Channel ได้นั้น จะมีประโยชน์เป็นอย่างมากในกรณีที่มีการใช้งานเครื่อง ET-RF24G V1.0 จำนวนหลายๆกลุ่ม ในบริเวณพื้นที่ใกล้เคียงกัน โดยให้กำหนดช่องความถี่ของ ET-RF24G V1.0 กลุ่มที่จะสื่อสารข้อมูลร่วมกันไว้ที่ช่องความถี่เดียวกัน ส่วนกลุ่มอื่นๆก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไป เพื่อลดปัญหาการรบกวนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

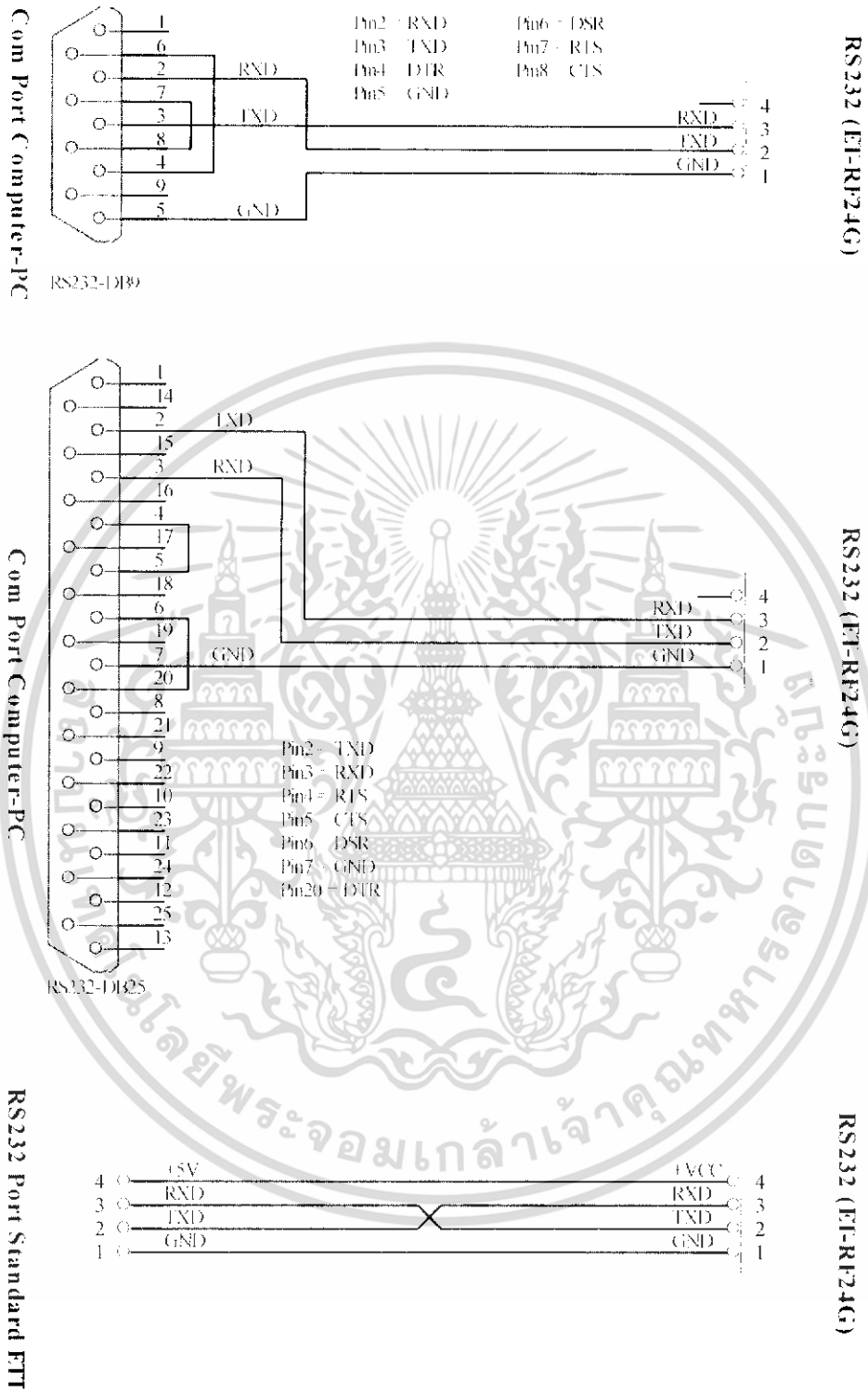
ข้อแนะนำในการกำหนดค่า Configuration

การกำหนดค่า Configuration ให้กับเครื่อง ET-RF24G V1.0 นั้น สามารถเลือกกำหนดได้ตามความต้องการและจุดประสงค์ของการใช้งาน โดยแต่ละโหมดของการใช้งานนั้นจะมีค่า Configuration ที่เหมาะสมต่างกัน ซึ่งขอแนะนำวิธีการกำหนดค่า Configuration ดังแนวทางต่อไปนี้

- ความเร็วในการรับส่งข้อมูลด้าน RS232 หรือ User RS232 Baudrate ที่ความเร็ว 19200 Bps นั้น เหมาะกับการใช้งาน ET-RF24G V1.0 แบบ Receive Only หรือ Transmit Only ซึ่งมีการตรวจสอบความพร้อมของสัญญาณในการรับส่งข้อมูลกันด้วย แต่ถ้าต้องการใช้งานเครื่อง ET-RF24G V1.0 ในโหมด Auto Direction นั้น ควรกำหนดค่า User RS232 Baudrate ไว้ที่ความเร็วไม่เกิน 9600 Bps จะดีที่สุด และควรกำหนดค่า Baudrate ของทั้งสองฝ่ายให้มีค่าเท่ากันด้วย
- ค่าความเร็วของการรับส่งข้อมูลด้าน RF หรือ RF Data Rate ที่สามารถรับส่งข้อมูลกันได้ระยะทางไกลมากที่สุด และ มีโอกาสผิดพลาดน้อยที่สุด คือ 250Kbps
- ค่า RF Power Gain ที่ดีที่สุดคือ 0dBm ซึ่งเป็นค่ากำลังส่งสูงสุด ซึ่งจะทำให้สามารถส่งข้อมูลได้ระยะทางไกลที่สุด แต่ถ้าระยะการรับส่งข้อมูลไม่ไกลกันมาก และมีการใช้งานเครื่อง ET-RF24G V1.0 จำนวนหลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ก็อาจทำการลดกำลังส่งให้ต่ำลงเพื่อลดปัญหาการรบกวนกัน หรือกำหนดช่องความถี่ RF Frequency Channel ให้ห่างกันมากๆ
- ในกรณีที่มีการใช้เครื่อง ET-RF24G V1.0 หลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ควรกำหนดช่องความถี่ในการใช้งาน หรือ RF Frequency Channel ให้ห่างกันด้วยเพื่อป้องกันการรบกวนกัน
- การใช้งานเครื่อง ET-RF24G V1.0 แบบ Auto Direction นั้น ถ้ามีการส่งข้อมูลจำนวนมากๆ ควรจัดแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดข้อมูลชุดละไม่เกิน 64 Byte โดยในการส่งข้อมูลแต่ละชุดนั้นให้ทำการส่งข้อมูลอย่างต่อเนื่องโดยให้ข้อมูลแต่ละ Byte มีระยะเวลาห่างกันไม่เกิน 2.5mS เนื่องจากถ้าข้อมูลขาดหายไปนานกว่านี้ เครื่อง ET-RF24G V1.0 จะทำการเปลี่ยนโหมดของการส่งข้อมูลกลับเป็นโหมดของการรับข้อมูลแทน ซึ่งเมื่อมีการส่งข้อมูล Byte ถัดไปมาอีกก็จะต้องเสียเวลาในการสลับโหมดจากฝ่ายรอรับข้อมูลให้เป็นฝ่ายส่งข้อมูลอีก ซึ่งจะทำให้ประสิทธิภาพในการจัดส่งข้อมูลลดลงเนื่องจากต้องเสียเวลาในการสลับโหมดการทำงานของวงจรภาค RF อยู่ตลอดเวลา โดยที่เมื่อทำการจัดส่งข้อมูลครบ 64 Byte แล้ว ให้ทำการหน่วงเวลาไว้ชั่วขณะหนึ่ง ประมาณ 1mS-2mS แล้วจึงส่งข้อมูลชุดถัดไปอีกอย่างนี้เรื่อยๆ จะทำให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด
- การใช้งานเครื่อง ET-RF24G V1.0 แบบ Auto Direction นั้น ควรหน่วงเวลาในการสลับโหมดจากฝ่ายของการรอรับข้อมูลเป็นฝ่ายส่งข้อมูล อย่างน้อยที่สุด 3mS - 5mS ซึ่งถ้าส่งข้อมูลย้อนกลับด้วยเวลาที่เร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมต่อสัญญาณ RS232



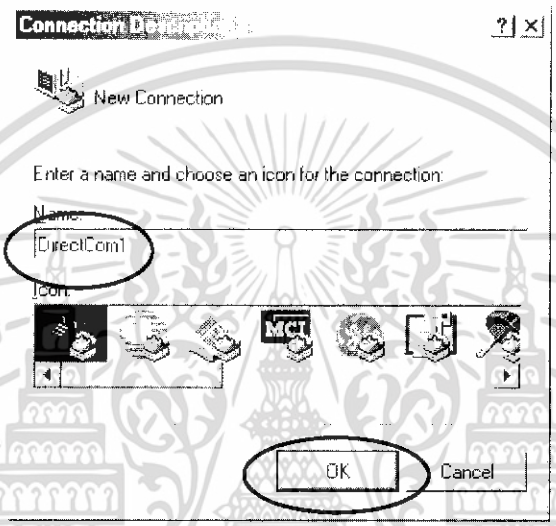
รูปแสดง แผนผังการต่อสาย RS232 เพื่อใช้งานกับ ET-RF24G V1.0 ในโหมด Auto Direction

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

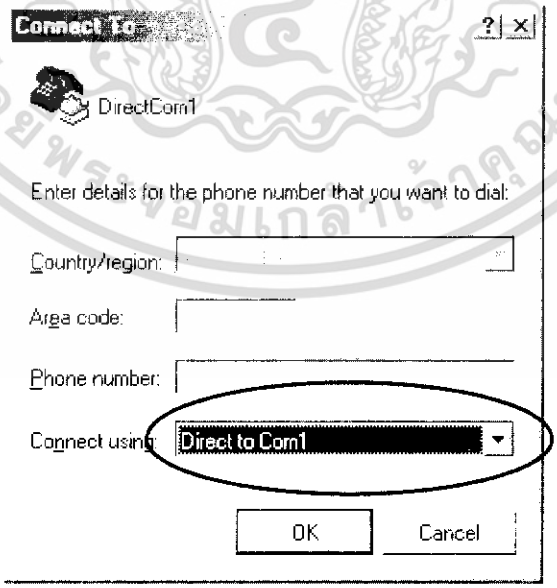
ตัวอย่างการใช้งาน

สำหรับตัวอย่างการใช้งานนั้น จะขอแสดงให้เห็นโดยใช้คอมพิวเตอร์ PC เป็นอุปกรณ์การทดลอง โดยในที่นี้จะขอเลือกใช้โปรแกรมสำเร็จรูปสำหรับใช้ในการสื่อสารของ Windows ซึ่งก็คือ Hyper Terminal โดยใน 2 ตัวอย่างแรกนั้นจะใช้งานกับเครื่อง ET-RF24G V1.0 ในโหมด Auto Direction ซึ่งมีวิธีการใช้งานดังต่อไปนี้

1. เรียกใช้โปรแกรม Hyper Terminal ของ Windows โดยเรียกจาก Start → Programs →Accessories → Communications → Hyper Terminal ซึ่งจะได้ผลดังรูป

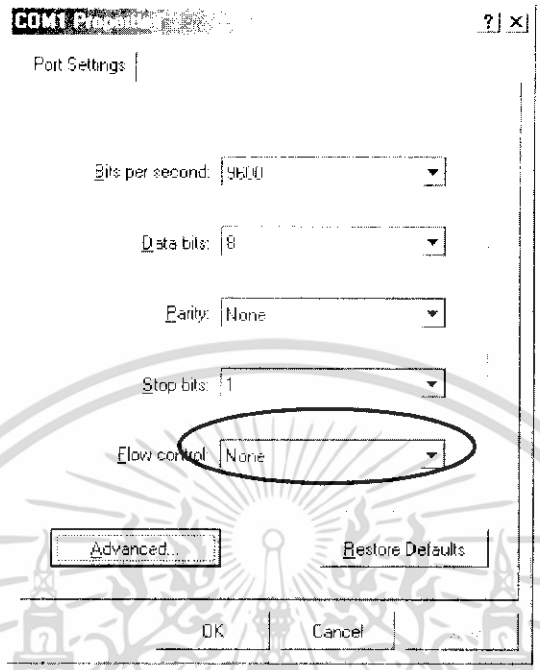


2. ให้เลือกกำหนดชื่อสำหรับใช้ในการเชื่อมต่อ ซึ่งสามารถกำหนดได้เองตามต้องการ โดยในตัวอย่างจะกำหนดเป็น DirectCom1จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป

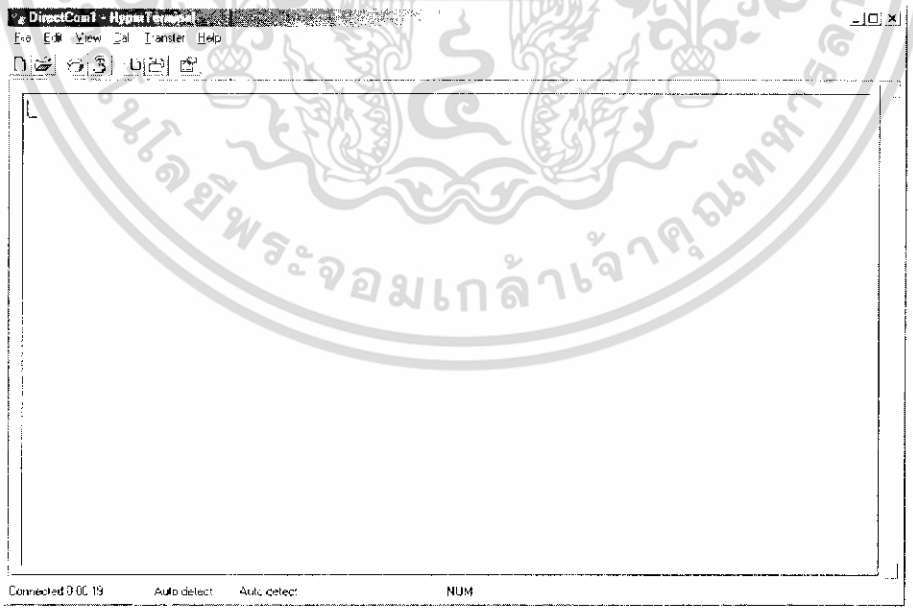


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ให้เลือกกำหนดการเชื่อมต่อเป็น Direct to Com1 ซึ่งถ้าเครื่องคอมพิวเตอร์ที่ใช้เป็น Comport อื่นที่ไม่ใช่ Com1 ก็ให้เลือกให้ตรงกับความเป็นจริง จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป



4. ในขั้นตอนนี้ จะใช้สำหรับกำหนดคุณสมบัติของพอร์ตอนุกรม RS232 โดยให้เลือก Bit per second - 9600 ,Data Bit = 8 ,Parity = None ,Stop Bit = 1 ส่วน Flow Control ให้เลือกเป็น None จากนั้นเลือก OK ซึ่งจะเข้าสู่หน้าต่างโปรแกรมหลักของ Hyper Terminal ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่1 การรับส่งข้อมูล 2 ทิศทาง (Half Duplex) แบบ จุดต่อจุด (Point-to-Point)

สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบ RS232 จำนวน 2 ชุด โดยต้องใช้รูปแบบการสื่อสารแบบ Half Duplex หรือ ผลัดกันรับ ผลัดกันส่ง กล่าวคือ ด้านรับจะต้องทำการรอรับข้อมูลจากด้านส่งจนครบทั้งหมด แล้วจึงจะส่งข้อมูลตอบกลับไปได้ ซึ่งจะไม่สามารถส่งข้อมูลสวนทางกลับไปในขณะที่กำลังรับข้อมูลอยู่ได้ โดยการสื่อสารแบบนี้ฝ่ายรับข้อมูลจะต้องรอให้รับข้อมูลได้ครบทั้งหมดเสียก่อน จากนั้นจึงจะส่งข้อมูลตอบกลับไปได้ โดยให้กำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V1.0 เป็นดังนี้

ค่า Configuration	ET-RF24G V1.0 ตัวที่1	ET-RF24G V1.0 ตัวที่2
User RS232 Baudrate	9600 Bps	9600 Bps
RF Data Rate	250 Kbps	250 Kbps
RF Operation Mode	Auto Direction	Auto Direction
RF Power Gain	+0dBm	+0dBm
RXD ID Code	01	02
TXD ID Code	02	01
RF Frequency Channel	0	0

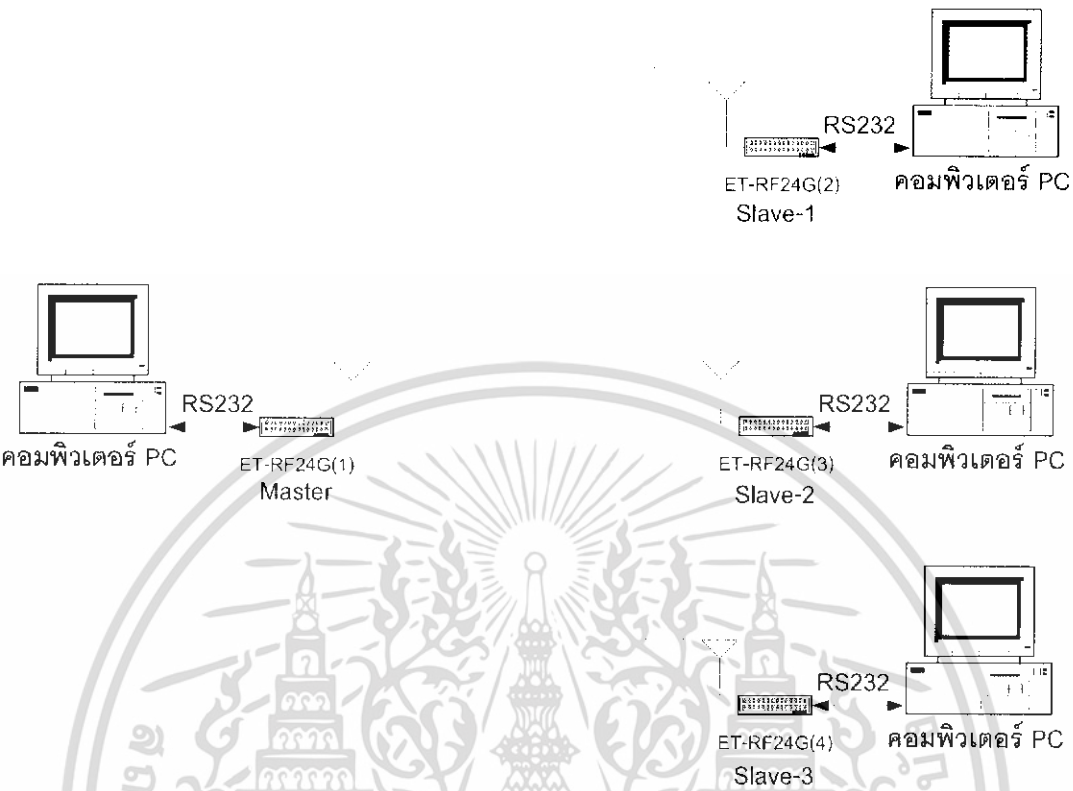
ข้อสังเกตุในการกำหนด Configuration

- ค่า RF Frequency Channel ต้องกำหนดให้ตรงกันทั้ง 2 ตัว
- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้ง 2 ตัว
- ค่า RXD ID Code ของตัวที่1 ต้องตรงกับ TXD ID Code ของตัวที่2
- ค่า TXD ID Code ของตัวที่1 ต้องตรงกับ RXD ID Code ของตัวที่2

สำหรับการทดสอบการทำงานด้วย Hyper Terminal นั้นให้ทดลองกดคีย์ใดๆ ในขณะที่ Run โปรแกรม Hyper Terminal อยู่ โดยจะสังเกตุเห็นตัวอักษรจากแป้นพิมพ์ของฝ่ายที่เป็นฝ่ายส่งข้อมูล จะถูกส่งออกไปแสดงผลที่หน้าจอโปรแกรม Hyper Terminal ของอีกฝ่ายหนึ่งในทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่2 การรับส่งข้อมูล 2 ทิศทาง (Half Duplex) แบบ หลายๆจุด (RF Network)



สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบ RS232 จำนวน หลายๆ ตัวร่วมกัน โดยหลักการสื่อสารแบบนี้จะให้ตัว Master เป็นตัวควบคุมการสื่อสารกับ Slave แต่ละตัวในระบบ โดยเมื่อ Master จะทำการส่งข้อมูลออกไปจะมีการใส่รหัส ID Code ของ Slave ที่ต้องการสื่อสารด้วย รวมไปในชุดข้อมูลนั้นๆด้วย ซึ่ง Slave ทุกๆตัวจะรับข้อมูลจาก Master ได้เหมือนๆกัน แต่จะมี Slave เพียงตัวเดียวที่ตอบสนองต่อข้อมูลนั้นๆ โดยให้กำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V1.0 เป็นดังนี้

ค่า Configuration	ET-RF24G V1.0 ตัวที่1(Master)	ET-RF24G V1.0 ตัวที่2 (Slave1)	ET-RF24G V1.0 ตัวที่3 (Slave2)	ET-RF24G V1.0 ตัวที่4 (Slave3)
User RS232 Baudrate	9600 Bps	9600 Bps	9600 Bps	9600 Bps
RF Data Rate	250 Kbps	250 Kbps	250 Kbps	250 Kbps
RF Operation Mode	Auto Direction	Auto Direction	Auto Direction	Auto Direction
RF Power Gain	+0dBm	+0dBm	+0dBm	+0dBm
RXD ID Code	01	02	02	02
TXD ID Code	02	01	01	01
RF Frequency Channel	0	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อสังเกตุในการกำหนด Configuration

- ค่า RF Frequency Channel ต้องกำหนดให้ตรงกันทั้งหมดทุกตัว
- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้งหมดทุกตัว
- ค่า RXD ID Code ของตัวที่1(Master) ต้องตรงกับ TXD ID Code ของตัวที่2-4(Slave1-3)
- ค่า TXD ID Code ของตัวที่1(Master) ต้องตรงกับ RXD ID Code ของตัวที่2-4(Slave1-3)

สำหรับการสื่อสารแบบนี้ จะต้องมีการกำหนด Protocol ขึ้นมาใช้ในการรับส่งข้อมูลกันด้วย ตัวอย่างเช่น กำหนดให้ใช้รหัส เครื่องหมาย '*' เป็นรหัสเริ่มต้นของชุดข้อมูล ตามด้วยรหัสหมายเลข ID Code ของ Slave ปลายทางเป็นตัวเลข 2 หลัก และจบด้วยรหัส Enter ดังนั้น ในการส่งข้อมูลแต่ละครั้ง Master จะต้องทำการส่งข้อมูลขนาด 4 Byte เสมอ และทางด้าน Slave ก็จะต้องรอรับข้อมูล โดยจะรอรับรหัสเครื่องหมาย '*' เป็นอันดับแรก ซึ่งเมื่อรับรหัสเครื่องหมาย '*' ได้แล้ว จึงรอรับข้อมูลถัดไปอีก 2 Byte จากนั้นจึงรอรับข้อมูล Byte ที่ 4 ซึ่งจะต้องตรวจสอบว่าเท่ากับรหัส Enter หรือไม่ ซึ่งถ้าใช่ก็แสดงว่ารับข้อมูลได้ถูกต้อง จากนั้นจึงทำการตรวจสอบข้อมูลใน Byte ที่ 2 และ 3 ว่าตรงกับรหัส ID Code ของตัวเองหรือไม่ โดย Slave-1 จะมีรหัสเป็น '0','1' ส่วน Slave-2 และ Slave-3 ก็จะมีรหัส ID Code เป็น '0','2' และ '0','3' ตามลำดับ ซึ่งถ้าตรวจสอบแล้วพบว่าข้อมูลใน Byte ที่ 2 และ 3 ตรงกับค่ารหัส ID Code ของตนเอง ก็ให้ตอบกลับด้วย รหัส ID Code ตามด้วยข้อความ 'OK'

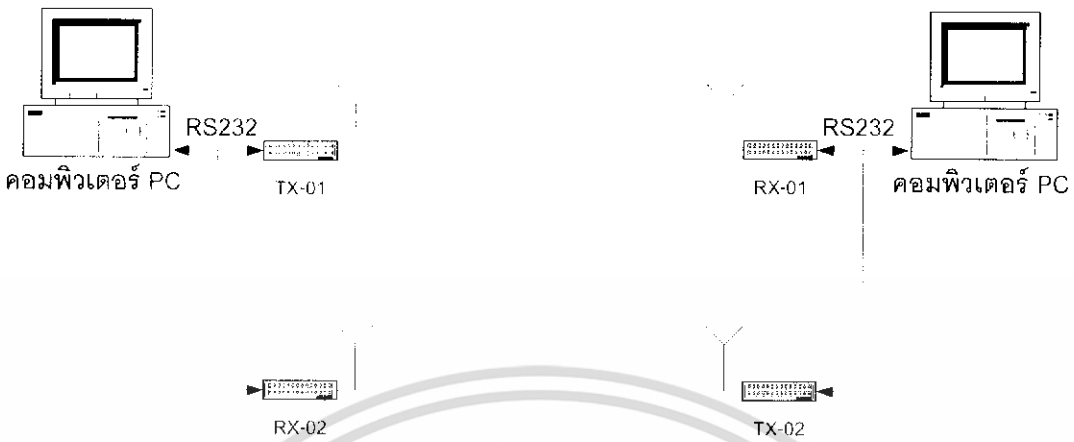
ซึ่งจากตัวอย่าง Protocol ข้างต้นจะได้ว่า เมื่อ Master ต้องการส่งข้อมูลไปยัง Slave-1 จะต้องมีการระบุหมายเลข ID Code ของ Slave-1 รวมไปในชุดข้อมูลด้วย โดยจะส่งข้อมูลเป็น '*','0','1',Enter ออกไป ซึ่งข้อความดังกล่าวที่ส่งออกไปจาก Master นั้น ตัว Slave ทุกตัวจะสามารถรับข้อมูลได้เหมือนกันทั้งหมด ซึ่ง Slave ทุกตัวจะต้องทำการวิเคราะห์ข้อมูลที่รับได้ ซึ่งในที่นี้ Slave-1 จะต้องตอบกลับด้วยข้อมูล '*','0','1','O','K',Enter เป็นต้น

ซึ่งจากตัวอย่างที่ได้กล่าวอธิบายมานี้ เป็นเพียงตัวอย่างแนวทางขั้นต้น เท่านั้น ซึ่งในการนำไปใช้งานจริงนั้นจะต้องมีการดัดแปลงและเพิ่มเติมข้อกำหนดต่างๆเข้าไปในชุดข้อมูลอีก เช่น รหัสคำสั่ง รหัสตรวจสอบความถูกต้องของข้อมูล (Checksum) เป็นต้น ซึ่งข้อกำหนดต่างๆเหล่านี้ ผู้ใช้สามารถ คิดค้น ออกแบบ รูปแบบของข้อมูลและคำสั่งต่างๆขึ้นมาใช้งานได้เองตามต้องการ

สำหรับการทดสอบการทำงานด้วยโปรแกรม Hyper Terminal นั้น อันดับแรกให้ทดสอบกดแป้นพิมพ์จากเครื่องที่เป็น Master ด้วยข้อความ *01 และ Enter ดู ซึ่งจะเห็นข้อความดังกล่าวไปแสดงอยู่ที่หน้าจอโปรแกรมที่เป็นของตัว Slave ทุกๆตัวเหมือนกันหมด จากนั้นให้ทดลองคีย์ข้อความ *01OK และ Enter จาก Slave-1 ซึ่งก็จะเห็นข้อความนั้นไปปรากฏที่หน้าจอโปรแกรมของตัว Master ทันที ซึ่งในการทดสอบการทำงานนั้นจะเห็นได้ว่า เมื่อมีการส่งข้อมูลใดๆจาก Master ข้อมูลนั้นจะถูกส่งไปแสดงผลยังหน้าจอของ Slave ทุกตัวเหมือนกันหมด และเมื่อมีการส่งข้อมูลใดๆจาก Slave ไม่ว่าตัวใด ข้อมูลนั้นๆก็จะถูกส่งไปแสดงผลยังหน้าจอของ Master เช่นเดียวกัน แต่ข้อมูลที่ถูกส่งจาก Slave จะไม่ถูกส่งไปแสดงผลที่หน้าจอของ Slave ตัวอื่นๆเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างที่ 3 การรับส่งข้อมูลแบบ Full Duplex ด้วยโปรแกรมสำเร็จรูป



ในตัวอย่างนี้จะเป็นการประยุกต์ใช้งานเครื่อง ET-RF24G V1.0 สำหรับทำการรับส่งข้อมูลกันแบบ Full Duplex โดยกำหนดโหมดการใช้งานเป็น RF Receive Only และ RF Transmit Only ฝ่ายละ 1 ชุด

ค่า Configuration	ET-RF24G V1.0 ฝ่ายต้นทาง		ET-RF24G V1.0 ฝ่ายปลายทาง	
	ตัวที่1 (RF RX1)	ตัวที่2 (RF TX1)	ตัวที่3 (RF RX2)	ตัวที่4 (RF TX2)
User RS232 Baudrate	19200 Bps	19200 Bps	19200 Bps	19200 Bps
RF Data Rate	250 Kbps	250 Kbps	250 Kbps	250 Kbps
RF Operation Mode	RF RX Only	RF TX Only	RF RX Only	RF TX Only
RF Power Gain	+0dBm	+0dBm	+0dBm	+0dBm
RXD ID Code	01	-	02	-
TXD ID Code	-	02	-	01
RF Frequency Channel	0	124	124	0

ข้อสังเกตุในการกำหนด Configuration

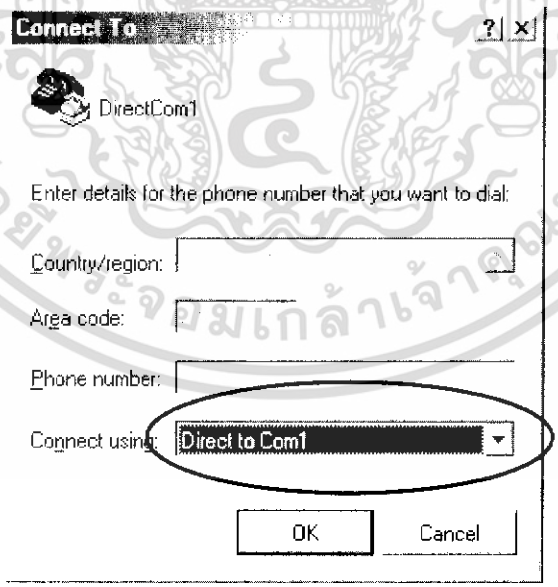
- ค่า RF Data Rate ต้องกำหนดให้ตรงกันทั้งหมดทุกตัว
- ค่า RF Frequency Channel ของ ตัวรับ ด้านต้นทาง ต้องกำหนดให้ตรงกับ ตัวส่ง ด้านปลายทาง
- ค่า RF Frequency Channel ของ ตัวส่ง ด้านต้นทาง ต้องกำหนดให้ตรงกับ ตัวรับ ด้านปลายทาง
- ค่า RXD ID Code ของ ตัวรับ ด้านต้นทาง ต้องตรงกับ TXD ID Code ของ ตัวส่ง ด้านปลายทาง
- ค่า TXD ID Code ของ ตัวส่ง ด้านต้นทาง ต้องตรงกับ RXD ID Code ของ ตัวรับ ด้านปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการทดสอบการใช้งาน ตามตัวอย่างนี้ สามารถเรียาใช้โปรแกรมสำเร็จรูปที่จัดการเรื่องการสื่อสารอนุกรมของ Windows ซึ่งก็คือ Hyper Terminal ได้ทันที โดยในการใช้งานนั้นสามารถกระทำได้ดังขั้นตอนต่อไปนี้

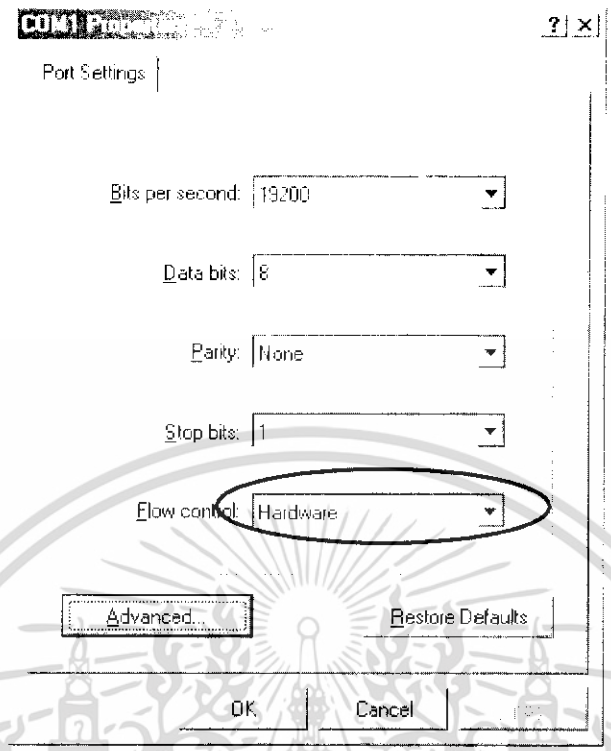
1. เรียกใช้โปรแกรม Hyper Terminal ของ Windows โดยเรียกจาก Start → Programs →Accessories → Communications → Hyper Terminal ซึ่งจะได้ผลดังรูป

2. ให้เลือกกำหนดชื่อสำหรับใช้ในการเชื่อมต่อ ซึ่งสามารถกำหนดได้เองตามต้องการ โดยในตัวอย่างจะกำหนดเป็น DirectCom1จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป

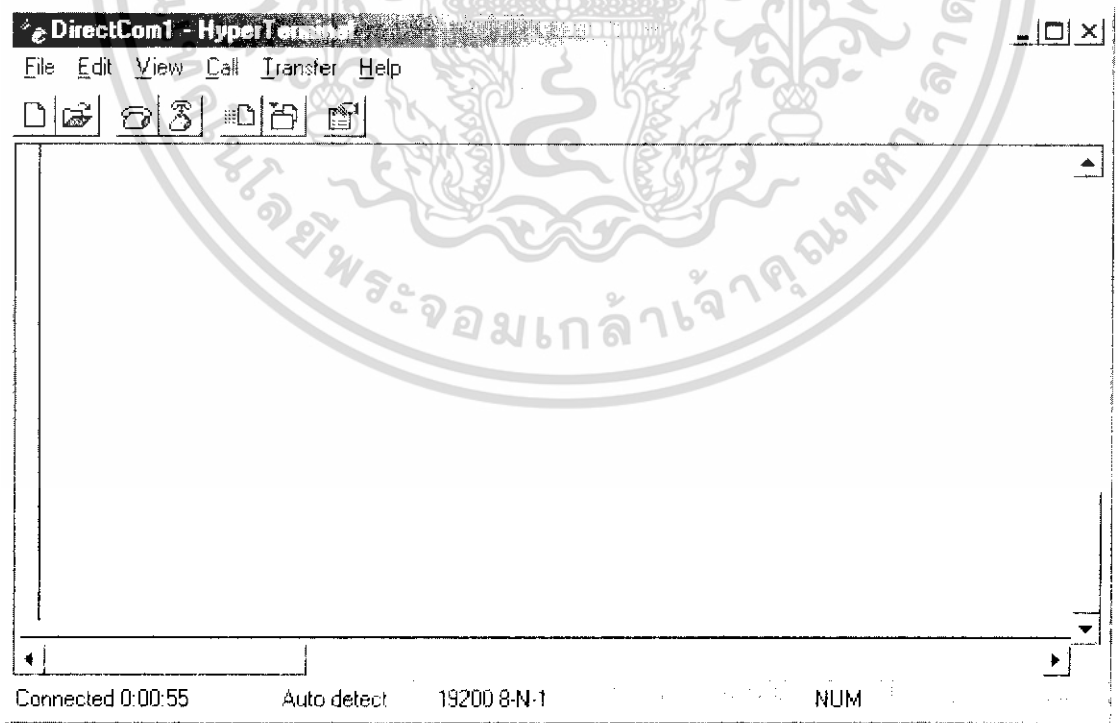


3. ให้เลือกกำหนดการเชื่อมต่อเป็น Direct to Com1 ซึ่งถ้าเครื่องคอมพิวเตอร์ที่ใช้เป็น Comport อื่นที่ไม่ใช่ Com1 ก็ให้เลือกให้ตรงกับความเป็นจริง จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



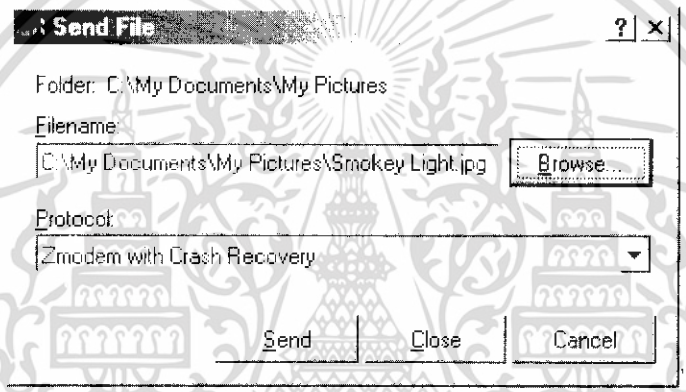
4. ในขั้นตอนนี้ จะใช้สำหรับกำหนดคุณสมบัติของพอร์ตอนุกรม RS232 โดยให้เลือก Bit per second = 19200 ,Data Bit = 8 ,Parity = None ,Stop Bit=1 ส่วน Flow Control ให้เลือกเป็น Hardware จากนั้นเลือก OK ซึ่งจะเข้าสู่หน้าต่างโปรแกรมหลักของ Hyper Terminal ดังรูป



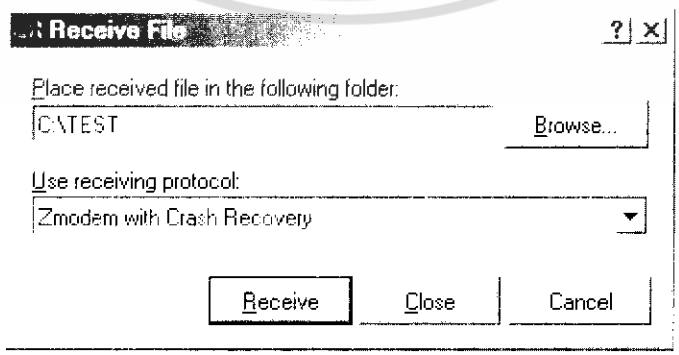
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ในขั้นตอนนี้สามารถทำการรับส่งข้อมูลระหว่างทั้ง 2 ฝ่ายได้แล้ว ซึ่งสามารถทดสอบได้โดยการกดคีย์ใดๆจากฝ่ายหนึ่ง ซึ่งตัวอักขระบนคีย์นั้นๆจะถูกส่งไปแสดงผลยังฝั่งตรงข้ามทันที แต่ในที่นี้เราจะทำการทดสอบการรับและส่งไฟล์ โดยใช้ Protocol สำเร็จรูปของ Hyper Terminal ซึ่งมีให้เลือกใช้มากมากหลาย Protocol โดยต้องกำหนด Protocol ให้ตรงกันทั้งฝ่ายส่งและฝ่ายรับ ซึ่งในขั้นตอนของการทดลอบนั้นต้องกำหนดให้ฝ่ายหนึ่งเป็นฝ่ายรับและให้อีกฝ่ายหนึ่งเป็นฝ่ายส่ง ซึ่งในที่นี้จะขอแนะนำให้ทดสอบโดยเลือกใช้ Protocol ของ Zmodem with Crash Recovery ซึ่งมีวิธีการทดสอบการรับส่งข้อมูลดังนี้

ทางด้านฝ่ายส่งให้ทำการเลือกกำหนดไฟล์ที่จะส่งจากเมนูคำสั่ง Transfer → Send File... จากนั้นให้เลือกกำหนดชื่อและที่อยู่ของไฟล์ที่ต้องการจะส่ง โดยคลิกเมาส์ที่ปุ่ม Browse พร้อมกับกำหนดชื่อและที่อยู่ของไฟล์ตามต้องการ จากนั้นให้เลือกกำหนด Protocol ของการรับส่งข้อมูลเป็น Zmodem with Crash Recovery แล้วเลือกคลิกเมาส์ที่ปุ่ม Send เพื่อทำการเริ่มต้นส่งข้อมูลดังรูป

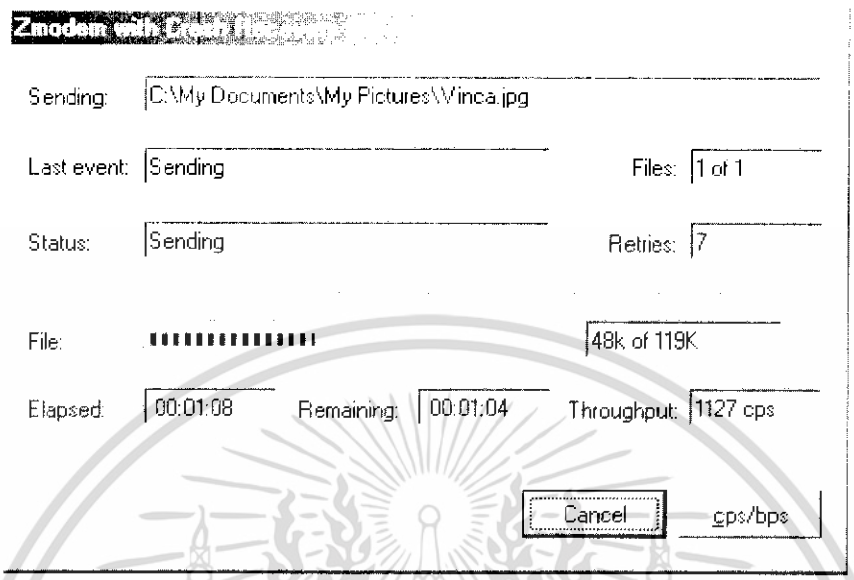


สำหรับในด้านที่เป็นฝ่ายรับข้อมูลนั้นก็ให้เลือกกำหนดการทำงานให้เป็นฝ่ายรับ โดยกำหนดจากเมนูคำสั่งของ Transfer → Receive File... จากนั้นให้เลือกกำหนดตำแหน่งของ Folder สำหรับใช้บันทึกไฟล์ที่รับได้จากฝ่ายส่ง โดยการเลือกจากปุ่ม Browse แล้วเลือกกำหนด Folder ที่ต้องการ ส่วนชื่อนั้นไม่ต้องกำหนด โดยโปรแกรม Hyper Terminal จะตั้งให้เองตามชื่อไฟล์จริงที่ส่งมา และในส่วนของ Protocol ที่ใช้นั้นก็ต้องกำหนดให้ตรงกับทางด้านส่ง คือ Zmodem with Crash Recovery จากนั้นให้เลือก Receive เพื่อให้โปรแกรมรอรับไฟล์จากด้านส่ง



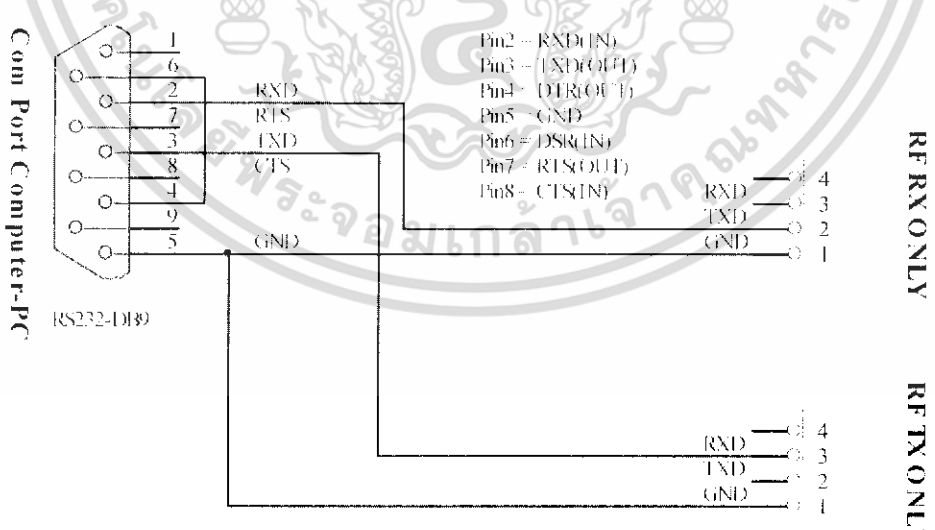
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในขณะที่มีการรับส่งข้อมูลกันอยู่นั้น โปรแกรม Hyper Terminal ทั้ง 2 ด้านจะแสดงสถานะการทำงานให้ทราบอยู่ตลอดเวลา ดังรูป



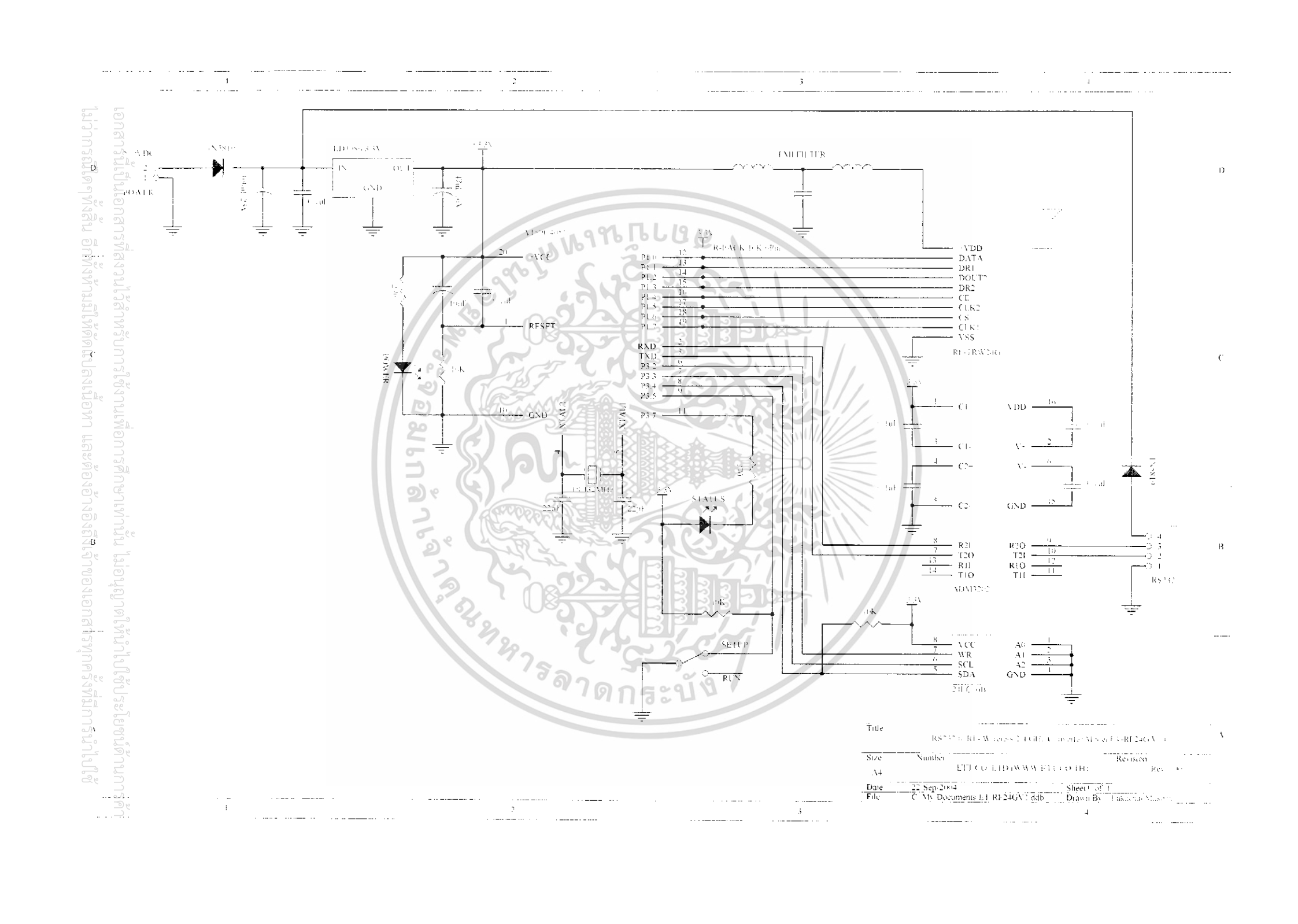
โดยให้รอจนกว่าการทำงานจะเสร็จสมบูรณ์ ซึ่งหน้าต่างที่แสดงสถานะการทำงานของโปรแกรมจะถูกปิดไปเองโดยอัตโนมัติหลังจากทำการรับส่งข้อมูลกันเสร็จเรียบร้อยแล้ว

โดยในการทดสอบการทำงานของโปรแกรมตามตัวอย่างนี้ จะต้องกำหนดรูปแบบการสื่อสารของ RS232 ให้มีการตรวจสอบความพร้อมในการรับส่งข้อมูลกันด้วยสัญญาณทาง Hardware ด้วย โดยเลือกกำหนดรูปแบบการสื่อสารของ RS232 ในหัวข้อ Flow Control เป็น Hardware พร้อมกับต่อสายสัญญาณดังวงจรต่อไปนี้



รูปแสดง วงจรของสายที่ใช้สำหรับทดสอบการรับส่งข้อมูลแบบ Full Duplex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


EXTERNAL
FILTER
VDD
DATA
DR1
DOUT
DR2
CE
CLK2
CS
CLK1
VSS
RXD
TXD
RESET
GND
STATUS
SETUP
RUN
VCC
WR
SCL
SDA
VDD
VSS
C1+
C1-
C2+
C2-
R2I
T2O
R1I
T1O
R2O
T2I
R1O
T1I
A0
A1
A2
GND
Title
Size
A4
Number
Revision
Date
22-Sep-2004
Sheet of
File
Drawn By