เครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless

โดย

นางสาวชมพูนุท ยอดนวล รหัสนักศึกษา B4802051

นางสาวเปรมวดี ถาวรธนทรัพย์ รหัสนักศึกษา B4602767

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2545

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ประจำภาคการศึกษาที่ 1 ปีการศึกษา 2551

โครงงาน	เครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless			
โดย	นางสาวชมพูนุท	ยอดนวล	รหัสนักศึกษา	B4802051
	นางสาวเปรมวดี	ถาวรธนทรัพย์	รหัสนักศึกษา	B4806530
อาจารย์ที่ปรึกษา	ผศ.ร.อ.ดร.ประโยชน์	คำสวัสดิ์		
สาขาวิชา	วิศวกรรมโทรคมนาคม			
ภาคการศึกษาที่	1/2551			

บทคัดย่อ

(Abstract)

ปัจจุบันประชากรส่วนใหญ่ของประเทศมีความเร่งรีบในการดำเนินชีวิตประจำวันจนบางครั้งอาจเผลอลืมปิดเครื่องใช้อุปกรณ์ไฟฟ้า ประกอบกับทรัพยากรธรรมชาติที่นำมาใช้ในการผลิตกระแสไฟฟ้ามีปริมาณลดน้อยลง เราจึงมีความจำเป็นที่จะต้องช่วยกันประหยัดพลังงาน โครงงานนี้นำเสนอการสร้างสร้างเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless อุปกรณ์ดังกล่าวประกอบด้วยชุดควบคุการเปิด-ปิดอุปกรณ์ไฟฟ้าภาครับสัญญาณและภาคส่งสัญญาณแบบไร้สาย โดยภาคส่งจะสั่งงานด้วยโปรแกรมคอมพิวเตอร์ ส่งผ่านทางพอร์ต RS232 ไปยัง RF-Wireless เพื่อส่งสัญญาณออกไป เมื่อภาครับได้รับคำสั่งก็จะนำคำสั่งดังกล่าวไปประมวลผลที่ไมโครคอนโทรลเลอร์ เพื่อทำการควบคุมอุปกรณ์ไฟฟ้า และจะมีการแสดงสถานะการทำงานในขณะนั้นบนหน้าจอคอมพิวเตอร์ตามโปรแกรมที่ได้ออกแบบไว้

เครื่องต้นแบบสำหรับควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าที่ได้ จะถูกนำไปใช้ในการวิจัยและพัฒนาการควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายสื่อสารต่อไป

กิตติกรรมประกาศ

(Acknowledgement)

จากการที่คณะผู้จักทำรายงานได้รับมอบหมายให้ทำโครงงานเรื่อง “เครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless” ส่งผลให้ผู้จัดทำรายงานได้รับความรู้และประสบการณ์ต่างๆเกี่ยวกับการออกแบบเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wirelessอย่างมาก บัดนี้โครงงานดังกล่าวพร้อมทั้งรายงานได้สำเร็จลงแล้ว ทั้งนี้ด้วยความร่วมมือและสนับสนุนจากบุคคลต่างๆดังต่อไปนี้

1. ผศ.ร.อ.ดร.ประโยชน์ คำสวัสดิ์ (อาจารย์ที่ปรึกษาโครงงาน)
2. อาจารย์วิชัย ศรีสุรักษ์ (ที่ปรึกษาด้าน Software)
3. นายนพวิทย์ พันธุ์บัว (วิศวกรส่วนวางแผน บริษัท TT&Tจำกัด เขตภาคเหนือตอนล่าง)
4. นายสมิง เติมพรมราช (ที่ปรึกษา ด้าน Software)
5. นายไพโรจน์ บุญไทย (พนักงาน บริษัท เชิดชัย จำกัด)
6. นายเอกสิทธิ์ เตียนพลกรัง (ที่ปรึกษา ด้าน Hardware)

ผู้จัดทำใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูลและที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจในการใช้อุปกรณ์และเครื่องมือต่างๆ ซึ่งผู้จัดทำขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

นางสาวชมพูนุท ยอดนวล
นางสาวเปรมวดี ถาวรธนทรัพย์

สารบัญ

เรื่อง **หน้า**

บทคัดย่อ.....ก
กิตติกรรมประกาศ..... ข
สารบัญ.....ค
บทที่ 1 บทนำ.....1
1.1 ความเป็นมา.....1
1.2 วัตถุประสงค์ของโครงงาน.....2
1.3 ขอบเขตการทำงาน.....2
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....2
1.5 ขั้นตอนการดำเนินงาน.....3
บทที่ 2 ทฤษฎีและหลักการ.....4
2.1 ไมโครคอนโทรลเลอร์.....4
2.2 การรับส่งข้อมูลผ่านระบบการสื่อสารไร้สาย.....32
2.3 รีเลย์.....40
บทที่ 3 การออกแบบระบบและสร้างอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้า.....44
3.1 แผนภาพของระบบ.....44
3.1.1 ภาคส่งสัญญาณ.....44
3.1.2 ภาครับสัญญาณ.....45
3.1.3 ชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้า.....46
3.2 การออกแบบทางHardware.....49
3.2.1 แผงเชื่อมต่ออุปกรณ์ไฟฟ้า.....49
3.3 การออกแบบทาง Software.....52
3.3.1 โปรแกรมการส่งข้อมูล.....52
3.3.2 โปรแกรมการรับข้อมูล.....58
3.4 รูปแบบของระบบ.....67
3.4.1 ชุดรับส่งข้อมูลผ่านระบบการสื่อสารไร้สาย.....67
3.4.2 แผงเชื่อมต่ออุปกรณ์ไฟฟ้า.....68

สารบัญ (ต่อ)

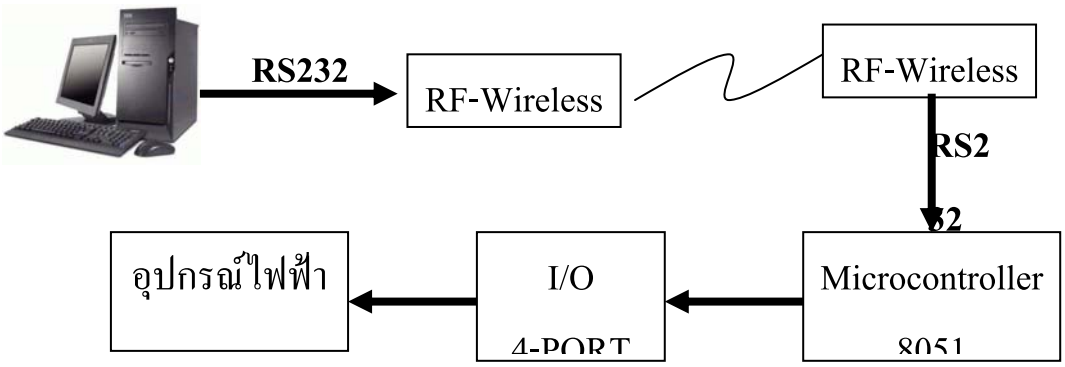
เรื่อง **หน้า**

บทที่ 4 ผลการทดลอง.....71
การทดลองที่ 1 การทดสอบคำสั่งของโปรแกรม.....71
การทดลองที่ 2 การทดสอบระยะทางการรับ-ส่งข้อมูล.....72
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ.....78
5.1 สรุปผลการทดลอง.....78
5.2 ข้อเสนอแนะและแนวทางการวิจัยในอนาคต.....78
ภาคผนวก ก โปรแกรมคอมพิวเตอร์.....79
ภาคผนวก ข โปรแกรมควบคุมการทำงาน.....94
ภาคผนวก ค อุปกรณ์รับ-ส่งการสื่อสารไร้สาย.....104
ประวัติผู้เขียน.....127
บรรณานุกรม.....128

บทที่ 1
บทนำ

1.1 ความเป็นมา

ปัจจุบันประชากรส่วนใหญ่ของประเทศมีความเร่งรีบในการดำเนินชีวิตประจำวันจนบางครั้งอาจเผลอลืมปิดเครื่องใช้อุปกรณ์ไฟฟ้า ประกอบกับทรัพยากรธรรมชาติที่นำมาใช้ในการผลิตกระแสไฟฟ้ามีปริมาณลดน้อยลง เราจึงมีความจำเป็นที่จะต้องช่วยกันประหยัดพลังงาน และนอกจากนี้ยังช่วยอำนวยความสะดวกให้แก่ผู้ใช้งาน ดังนั้นคณะผู้จัดทำจึงได้จัดทำโครงงานนี้นำเสนอการออกแบบ และสร้างเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless อุปกรณ์ดังกล่าวประกอบด้วยชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าภาครับสัญญาณและภาคส่งสัญญาณแบบไร้สาย โดยภาคส่งจะสั่งงานด้วยโปรแกรมคอมพิวเตอร์ ส่งผ่านทางพอร์ต RS232 ไปยัง RF-Wireless เพื่อส่งสัญญาณออกไป เมื่อภาครับได้รับคำสั่งจะนำคำสั่งไปประมวลผลที่ไมโครคอนโทรลเลอร์เพื่อทำการควบคุมอุปกรณ์ไฟฟ้า หลังจากนั้นจะทำการส่งค่าการทำงานกลับไปยังภาคส่งสัญญาณเพื่อแสดงผลที่คอมพิวเตอร์ ซึ่งโครงงานนี้จะสามารถสร้างความสะดวกสบายและประโยชน์ให้กับผู้ใช้งานเป็นอย่างมาก



รูปที่ 1.1 ผังไดอะแกรมของโครงงาน

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาและออกแบบอุปกรณ์เพื่อทำงานร่วมกับระบบคอมพิวเตอร์ผ่าน RF-Wireless
2. เพื่อศึกษาการเขียนโปรแกรมเชื่อมต่ออุปกรณ์ฮาร์ตแวร์กับคอมพิวเตอร์
3. เพื่อศึกษาการเขียนโปรแกรมการใช้งานการทำงานควบคุมไมโครคอนโทรลเลอร์ และ Visual Basic 6.0
4. เพื่อศึกษาและออกแบบวงจรเชื่อมต่ออุปกรณ์ฮาร์ตแวร์เข้าด้วยกัน

1.3 ขอบเขตการทำงาน

1. ศึกษาการเขียนโปรแกรมควบคุมการทำงานสั่ง ปิด – เปิดอุปกรณ์ไฟฟ้า ผ่าน RF-Wireless
2. เขียนโปรแกรมควบคุมการเชื่อมต่ออุปกรณ์ฮาร์แวร์กับคอมพิวเตอร์
3. ออกแบบการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์ กับ ปลั๊กเสียบอุปกรณ์ไฟฟ้า
4. สร้างอุปกรณ์ต้นแบบทั้งหมดและทดสอบเพื่อให้ได้ตามวัตถุประสงค์
5. ออกแบบ และเขียนโปรแกรมควบคุม บอกสถานะการทำงานของอุปกรณ์ไฟฟ้าให้แสดงผลบนหน้าจอคอมพิวเตอร์

1.4 ประโยชน์จะได้รับ

1. สามารถนำความรู้ที่ได้มาใช้การประกอบวิชาชีพ
2. สามารถทำงานเป็นทีมได้
3. สามารถนำความรู้ทางทฤษฎีมาประยุกต์ใช้ในทางปฏิบัติ

1.5 ขั้นตอนการดำเนินงาน

1. ศึกษา ค้นคว้าหาข้อมูล
2. เขียนโครงการและเสนอโครงการกับอาจารย์ที่ปรึกษา
3. ออกแบบระบบการทำงาน
4. จัดซื้ออุปกรณ์ที่ใช้ในโครงการ
5. ออกแบบกลไกการทำงานของภาครับและภาคส่ง
6. ออกแบบวงจรเชื่อมต่ออุปกรณ์ฮาร์ดแวร์
7. เขียนโปรแกรมควบคุมการทำงานของอุปกรณ์ภาครับและภาคส่ง
8. เขียนโปรแกรม Visual Basic 6 เพื่อแสดงสถานะการทำงานบนหน้าจอคอมพิวเตอร์
9. ทำการเชื่อมต่อสัญญาณที่โปรแกรม Visual Basic 6 กับอุปกรณ์ฮาร์ดแวร์
10. ประกอบอุปกรณ์ทั้งหมดเข้าด้วยกัน
11. ทดสอบการใช้งานของอุปกรณ์เพื่อให้เป็นไปตาม วัตถุประสงค์

บทที่ 2
ทฤษฎีและหลักการ

ในบทนี้จะเป็นการอธิบายเกี่ยวกับทฤษฎีและหลักการของอุปกรณ์แต่ละส่วนที่ประกอบด้วยไมโครคอนโทรลเลอร์ ชุดอุปกรณ์รับส่งข้อมูลแบบไร้สาย ทั้งภาครับและภาคส่ง และอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้า

2.1 ไมโครคอนโทรลเลอร์

ปัจจุบันไมโครคอนโทรลเลอร์ได้เข้ามามีบทบาทมากกับการควบคุมในงานอุตสาหกรรมไมโครคอนโทรลเลอร์ 8051 เป็นชิป ที่นิยมมากตัวหนึ่งที่ใช้ในงานควบคุมเนื่องจากมีความสามารถที่สูงและง่ายต่อการใช้งาน ทั้งนี้ต้องขึ้นอยู่กับผู้ใช้งานที่จะมีวิธีการอย่างไรในการดึงความสามารถของ ชิป มาให้ใช้เต็มที่เพื่อให้เกิดประสิทธิภาพสูงสุด ในการใช้ไมโครคอนโทรลเลอร์ เพื่อใช้ในงานควบคุมต่างๆนั้นจำเป็นที่จะต้องใช้ Memory ในการเก็บข้อมูลและใช้การส่งข้อมูลผ่านทาง I/O เพื่อรับข้อมูลเข้ามาประมวณผลและส่งสัญญาณออกไปควบคุมอุปกรณ์ที่ต้องการ และเพื่อความสะดวกในการใช้งานตัวไมโครคอนโทรลเลอร์จึงได้รวม Memory ไว้ภายในตัวชิป ซึ่งจะได้อธิบายต่อไปนี้

ในการอธิบายหลักการทำงานของไมโครคอนโทรลเลอร์นั้น ได้แบ่งการทำงานออกเป็น 4 แบบหลัก ๆ คือ

2.1.1 หน่วยความจำ

2.1.2 รีจิสเตอร์พิเศษ

2.1.3 พอร์ตอินพุตและพอร์ตเอาท์พุต

2.1.4 อินเตอร์รัป

2.1.1 หน่วยความจำ

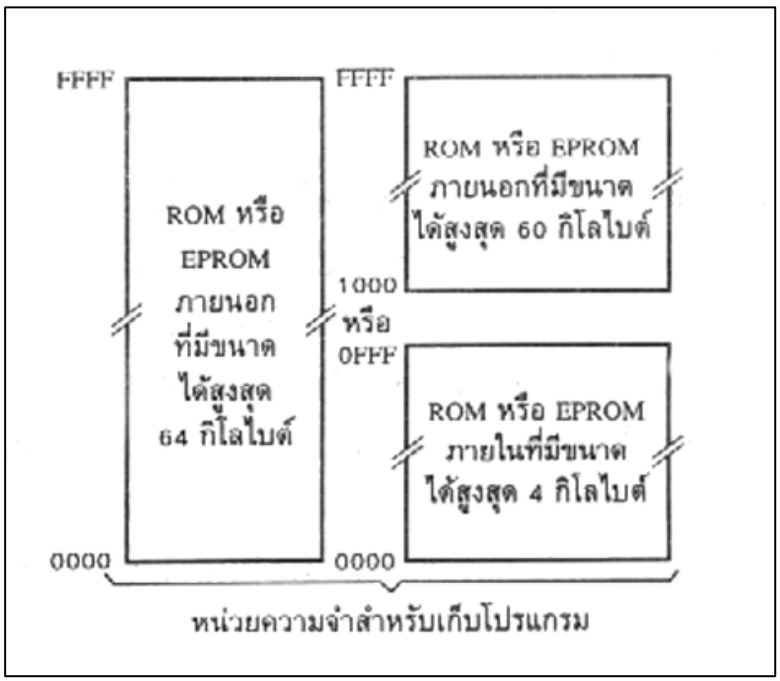
การจัดการหน่วยความจำและการเชื่อมต่อ

หน่วยความจำของไมโครคอนโทรลเลอร์ 8051 แบ่งเป็น 2 ประเภท คือ

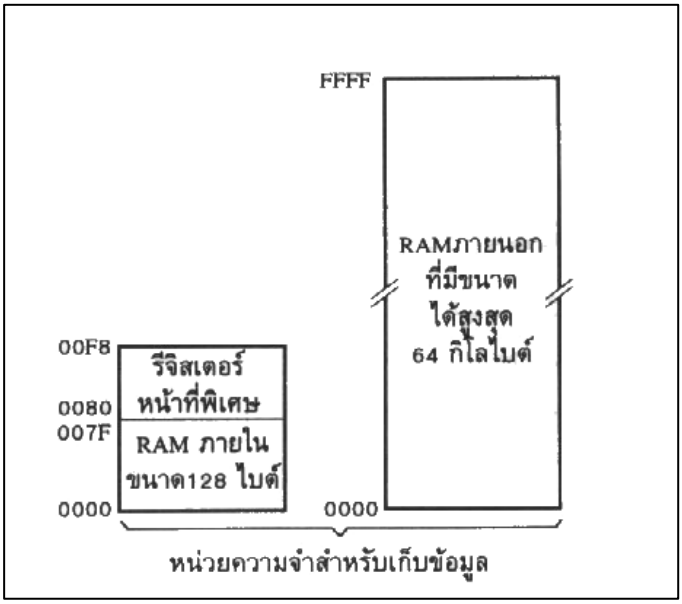
1. หน่วยความจำโปรแกรม(Program memory) หน่วยความจำประเภทนี้ คือ ROM ใช้เก็บโปรแกรมที่ใช้ในการควบคุมระบบ ซึ่งเป็นหน่วยความจำประเภท non-volatile
2. หน่วยความจำข้อมูล (Data memory) หน่วยความจำประเภทนี้ได้แก่ RAM

ระบบ Memory ของ 8051

สำหรับ ไมโครคอนโทรลเลอร์ 8051 จะมี ROM (หรือ EPROM) ขนาด 4 กิโลไบต์, RAM ขนาด 128 ไบต์ ตัว 8051 สามารถทำการอ้างอิงหน่วยความจำภายนอกได้ ถ้า RAM หรือ ROM ภายในนั้นมีขนาดไม่เพียงพอ โดยจะใช้ พอร์ต 2 ตัวในการทำหน้าที่อ้างอิงข้อมูลกับหน่วยความจำภายนอกซึ่งการทำเช่นนี้จะส่งผลให้พอร์ตที่ทำหน้าที่เป็น I/O มีจำนวนลดลง แต่ 8051 จะมีความยืดหยุ่นในการทำงานมากขึ้น



รูปที่ 2.1 ระบบหน่วยความจำสำหรับการของ 8051 โปรแกรม



รูปที่ 2.2 หน่วยความจำที่อ่านและเขียนข้อมูลลงไปได้

จากรูปแสดงแผนภาพการจัดสรรหน่วยความจำ-รีจิสเตอร์ (Memory-register map) ของ 8051 จากรูปเราจะเห็นได้ว่า 8051 มีรีจิสเตอร์หน้าที่พิเศษรวมอยู่ภายในด้วย สำหรับส่วนของ RAM และ ROM ภายนอกนั้นเป็นส่วนที่เราต้องทำการเพิ่มเข้าไปเอง แผนภาพการจัดสรรหน่วยความจำรีจิสเตอร์ยังแสดงถึงคุณสมบัติอีกอย่างหนึ่งของ 8051 ที่ทำหน้าที่แตกต่างจากไมโครคอนโทรลเลอร์ ทั่ว ๆไป โดย 8051 นี้จะมีหน่วยความจำอยู่ 2 ประเภทประเภทแรกเป็นหน่วยความจำสำหรับเก็บโปรแกรม และอีกประเภทเป็นหน่วยความจำสำหรับเก็บข้อมูลที่เปลี่ยนได้ หน่วยความจำสำหรับเก็บโปรแกรมนั้นจะเป็นหน่วยความจำที่เราสามารถทำการอ่านข้อมูลได้อย่างเดียว เราอ่านคำสั่งของโปรแกรมได้แต่ไม่สามารถสั่งให้หน่วยประมวณผลทำการเขียนข้อมูลลงในหน่วยความจำประเภทนี้ได้ ซึ่งเราเรียกหน่วยความจำประเภทนี้ว่า ROM โดย ROM ภายในของ 8051 ถูกกำหนดให้เป็นหน่วยความจำสำหรับเก็บโปรแกรม และไมโครคอนโทรลเลอร์จะทำการ Fetch คำสั่งทุกคำสั่งจากหน่วยความจำประเภทนี้เท่านั้น หน่วยความจำสำหรับเก็บข้อมูลเป็นหน่วยความจำประเภทที่สามารถทำการอ่านและเขียนข้อมูลได้ หน่วยประมวลผลสามารถทำการอ่านข้อมูลและเขียนข้อมูลลงในหน่วยความจำประเภทนี้ แต่ไม่สามารถทำงานตามคำสั่งของโปรแกรมที่อยู่ในหน่วยคความจำประเภทนี้ได้ เราจะพบว่า RAM ภายในของ 8051 จะถูกกำหนดให้เป็นหน่วยความจำสำหรับเก็บข้อมูล

8051 มี RAM ภายในขนาด 128 ไบต์ (อยู่ในตำแหน่งตั้งแต่ 00H จนถึง 7FH ของหน่วยความจำ) สำหรับใช้ในการเขียนและอ่านข้อมูลทั่วๆ ไป ถึงแม้เราจะกล่าวว่า 8051 มี RAM ภายในขนาด 128 ไบต์ แต่ในบางครั้งบางส่วนของหน่วยความจำนี้อาจจะถูกนำมาอ้างอิงเป็นรีจิสเตอร์ที่ทำหน้าที่ทั่วไป ในกรณีที่เราต้องการเพิ่มหน่วยความจำสำหรับเก็บโปรแกรม(ROM) เราสามารถทำการขยายหน่วยความจำภายในสำหนับเก็บโปรแกรมที่มีขนาด 4 กิโลไบต์ ให้เป็นหน่วยความจำที่มีขนาดได้ถึง 64 กิโลไบต์ โดยการเพิ่มหน่วยความจำอีก 60 กิโลไบต์ หรือเราอาจทำการขยายหน่วยความจำสำหรับเก็บโปรแกรมโดยการต่อขา EA' ลงกราวด์ซึ่งจะทำให้ 8051 ไม่ใช้ ROM ภายในขนาด 4 กิโลไบต์ แต่จะใช้หน่วยความจำภายนอกซึ่งตำแหน่งของหน่วยความจำภายนอกจะเริ่มจากตำแหน่งที่ 0000H และมีขนาดสูงสุดได้ถึง 64 กิโลไบต์ เราสามารถเพิ่ม RAM ได้ ถ้าเราต้องการ RAM จำนวนมากขึ้น จากรูปจะเห็นได้ว่าเราสามารถทำการอ้างอิงพื้นที่ RAM

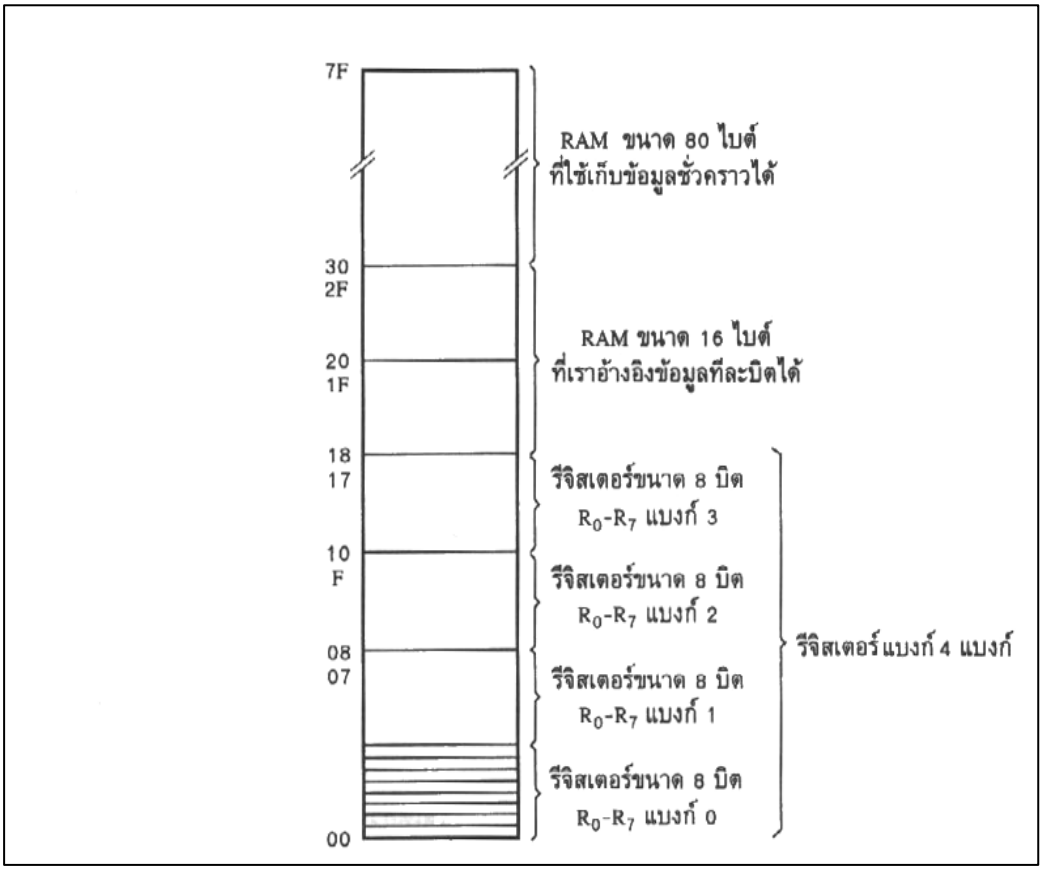
ภายนอกได้ถึง 64 กิโลไบต์ การอ้างอิงแอดเดรสในหน่วยความจำส่วนนี้จะไม่เกี่ยวข้องกับการอ้างอิงแอดเดรส RAM ภายในขนาด 128 ไบต์ ดังนี้นเราจะมีหน่วยความจำสำหรับเก็บข้อมูลทั้งสิ้น 64 กิโลไบต์ รวมกับ RAM ภายในที่มีอยู่เดิมอีก 128 ไบต์

โดยทั่วไปแล้ว 8051 จะทำงานโดยมี RAM และ ROM แยกออกจากกัน แต่ในงานบางชนิดอาจมีความต้องการให้หน่วยความจำทั้งสองประเภทนี้ทำงานรวมกันเหมือนเป็นหน่วยความจำโดยทั่วไป ซึ่งการทำเช่นนี้จะทำให้ 8051 มีหน่วยความจำภายนอกเพียง 64 กิโลไบต์ เท่านั้น แต่อย่างไรก็ตามเมื่อใช้วิธีการนี้แล้วจะทำให้ 8051 สามารถทำการรับข้อมูลเป็น Block ผ่านพอร์ตอนุกรม นำข้อมูลนี้ไปเก็บในหน่วยความจำและทำการ Execute ข้อมูลนี้เป็นโปรแกรมได้ ขั้นตอนนี้เรียกว่าการ Download program เทคนิคนี้นิยมใช้ในการเปลี่ยนโปรแกรมของระบบควบคุมในระยะทางไกลที่มีไมโครคอนโทรลเลอร์ เป็นพื้นฐาน โดยปกติแล้วโปรแกรมที่กำกับขั้นตอนการ Download program จะถูกเก็บอยู่ใน ROM

RAM ภายในของ 8051

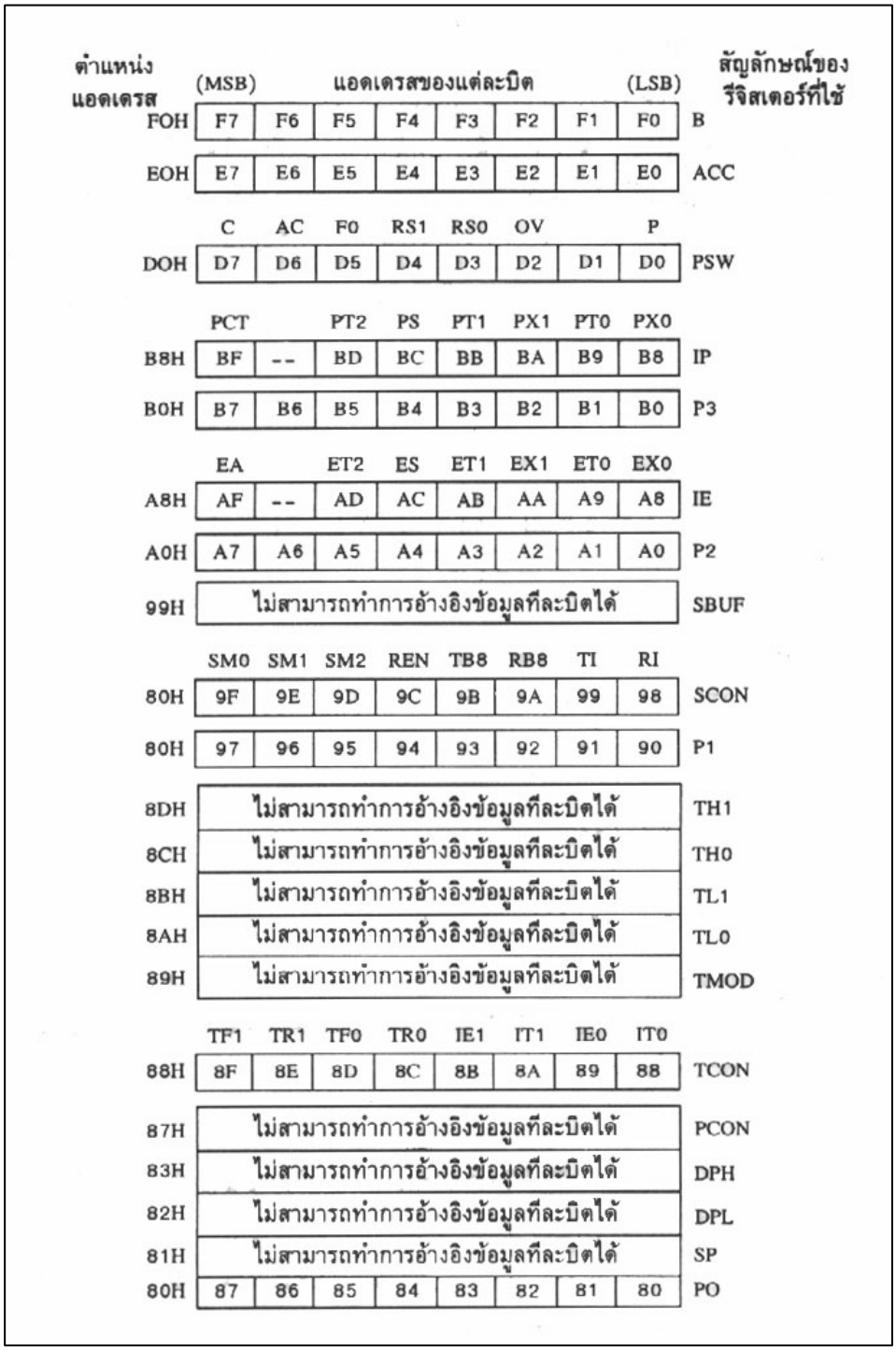
เพื่อให้การเขียนโปรแกรมนั้นเป็นไปได้อย่างคล่องตัว 8051 สามารถทำการอ้างอิงแอดเดรสของ RAM ภายในได้หลายวิธี โดยเราสามารถอ้างอิง RAM 32 ตำแหน่งแรกเป็นรีจิสเตอร์ หรืออ้างอิงเป็นตำแหน่งหนึ่งในหน่วยความจำก็ได้ การเข้าถึงข้อมูลในตำแหน่งที่ถูกอ้างอิงเป็นรีจิสเตอร์ นั้นเราสามารถทำได้โดยใช้คำสั่งที่มีขนาดเพียงแค่ 1 ไบต์ ซึ่งการใช้คำสั่งชนิดนี้จะช่วยสามารถเขียนโปรแกรมได้อย่างมีประสิทธิภาพ

RAM ของ 8051 นี้จะมีพื้นที่ส่วนหนึ่งขนาด 16 ไบต์ ซึ่งเราสามารถทำการอ้างอิงข้อมูล RAM ส่วนนี้ทีละบิตหรือทีละไบต์ก็ได้ โดยคำสั่งทั่วไปจะสามารถทำการอ้างอิงข้อมูลทีละไบต์จากหน่วยความจำส่วนนี้ แต่ถ้าเราต้องการอ้างอิงข้อมูลทีละบิตในหน่วยความจำส่วนนี้เราจะต้องใช้คำสั่งพิเศษ คำสั่งพิเศษนี้มีประโยชน์มากเมื่อเราต้องการทำการประมวลข้อมูลที่รับมาจากอุปกรณ์ภายนอก เช่น ในงานที่มีการใช้ระบบควบคุม ในหัวข้อต่อไปเราจะพบว่าคำสั่งที่ติดต่อกับ I/O เพื่อทำการเคลื่อนย้ายข้อมูลของ 8051 สามารถเคลื่อนย้ายข้อมูลที่มีขนาดเป็นไบต์หรือบิตก็ได้



รูปที่ 2.3 การจัดสรรหน่วยความจำ RAM ภายในของ 8051

รูปที่ 2.3 แสดงแผนภาพการจัดสรรหน่วยความจำภายในช่วง 128 ตำแหน่งแรกของ 8051 จากรูปเราจะพบว่า 32 ตำแหน่งแรกในหน่วยความจำจะถูกกำหนดให้เป็นรีจิสเตอร์แบก์ จำนวน 4 แบงก์ ในแต่ละ แบงก์ จะมีรีจิสเตอร์ 8 ตัวรีจิสเตอร์ ในที่นี้เป็นตำแหน่งในหน่วยความจำสำหรับเขียนและอ่านข้อมูลขนาด 8 บิต ซึ่งเราสามารถอ้างอิงได้โดยใช้คำสั่งที่มีขนาด 1 ไบต์ และเนื่องจากว่าคำสั่งขนาด 1 ไบต์ นี้จะใช้บิตเพียงแค่ 3 บิต ในการระบุตำแหน่งของรีจิสเตอร์ ที่เราต้องการดังนั้นเราจึงใช้คำสั่งนี้อ้างรีจิสเตอร์ ที่แตกต่างกันได้ เพียง 8 ตัว เราจะทราบได้อย่างไรว่าแบงก์รีจิสเตอร์แบงก์ใดจะถูกนำมาใช้งาน แบงก์รีจิสเตอร์ที่กำลังใช้งานอยู่ในขณะนั้นจะถูกเลือกโดยการตั้งค่าที่แฟลกสำหรับเลือกแบงก์ เมื่อแบงก์รีจิสเตอร์หนึ่งๆ ถูกเลือกแล้ว การเคลื่อนย้ายข้อมูลจะเกิดขึ้นกับรีจิสเตอร์ R0 และ R7 ของแบงก์นั้น ถ้าเราต้องการเปลี่ยนไป ใช้รีจิสเตอร์ในแบงก์อื่น เราต้องเปลี่ยนค่าในแฟลกสำหรับเลือกแบงก์ก่อนเพื่อเลือกแบงก์ที่จะทำงาน รูปที่ 2.3 แสดงพื้นที่ขนาด 16 ไบต์ ในหน่วยความจำตั้งแต่ตำแหน่งที่ 20H ถึง 30H ซึ่งเป็นส่วนของ RAM ที่เราสามารถอ้างอิงข้อมูลเป็นบิตได้ โดยเราสามารถใช้คำสั่งพิเศษเพื่ออ้างอิงข้อมูลในหน่วยความจำส่วนนี้ทีละบิต หรือใช้คำสั่งทั่วไปเพื่ออ้างอิงข้อมูลทีละไบต์ก็ได้ การใช้คำสั่งพิเศษใน 8051 ช่วยให้สามารถทำการ Complement บิตใดก็ได้หรือทำการ AND บิตหนึ่งกับอีกบิตหนึ่งที่อยู่ใน แอกคิวมูเลเตอร์ หรืออยู่ในรีจิสเตอร์สถานะเป็นต้น พื้นที่ส่วนสุดท้ายใน RAM ที่มีขนาด 80 ไบต์ นั้นมีชื่อว่า Scratchpad ซึ่งสามารถนำข้อมูลที่เปลี่ยนแปลงได้ไปเก็บ



รูปที่ 2.4 รีจิสเตอร์หน้าที่พิเศษใน 8051

2.1.2 รีจิสเตอร์พิเศษ

รีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่และการทำงานของอุปกรณ์หรือพอร์ตของ 8051 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH
การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้ สามารถทำได้ทั้งการระบุถึงชื่อของรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

ตารางต่อไปนี้แสดงเห็นลักษณะการจัดพื้นที่หน่วยความจำ สำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้โดยมีข้อสังเกตว่ารีจิสเตอร์ที่อยู่ตำแหน่งแอดเดรสที่เป็นจำนวนทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือ แอดเดรส 80H,88H,90H,98H,A0H,A8H,B0H,B8H,D0H,E0H และ F0H)

- แอกคิวมูเลเตอร์ (Accumulator) หรือ ACC

เป็นรีจิสเตอร์ขนาด 8 บิตทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียูและเก็บผลลัพธ์ ที่ได้จากการทำงานนั้น การทำงานของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไปการใช้งานภาย
ในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

- รีจิสเตอร์ B เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณแลหารตัวเลข ในกรณีที่ไม่ใช้ในการคำนวณทางด้าน คณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

- โปรแกรมเคาน์เตอร์ (Program counter)

เป็นรีจิสเตอร์ที่ใช้ในการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัด ไป การใช้งานภายในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

- สแต็กพอยน์เตอร์ (Stack pointer)

เป็นรีจิสเตอร์ขนาด 8 บิตทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ (Pointer) ของบริเวณสแต็ก (Stack) สำหรับเก็บขอ้มูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรมโดยปกติแล้วเมื่อทำการเริ่มต้นระบบใหม่ภายหลังจากการเริ่มจ่ายไฟฟ้า หรือมีการรีเซต (Reset) เกิดขึ้นค่าภายในสแต็กพอยน์เตอร์จะมีค่า 07H ซึ่ง เป็นตำแหน่งแอดเดรสภายในบริเวณเนื้อที่ 128 ไบต์แรกของหน่วยความจำข้อมูลภายใน การใช้งานภายในโปรแกรม จะเรียกว่า รีจิสเตอร์ SP

- ตัวชี้ข้อมูล หรือ ดาต้าพอยน์เตอร์ (Data pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิตสองตัวคือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าของแอดเดรสของหน่วยความจำที่ต้องใช้งานภายในโปรแกรมหรืออาจจะเป็นแอดเดรสของอุปกรณ์ภายนอก ซึ่งกำหนดให้ติดต่อกันโดยใช้ตำแหน่งของหน่วยความจำภายในโปรแกรม

- โปรแกรมสเตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟลกสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์ (Bank) ของรีจิสเตอร์ที่ใช้งานด้วยดังแสดง

*ชื่อบิต: PSW ตำแหน่ง: D0h ค่าบิตเริ่มต้น: 0000 0111

ตารางที่ 2.1 โปรแกรมสเตตัสเวิร์ด (PSW)

ชื่อบิต	ตำแหน่ง	ความหมาย
CY	PSW.7	Carry Flag
AC	PSW.6	Auxiliary Carry Flag
F0	PWS.5	Flag 0
RS1	PSW.4	Select Blank bit 1
RS0	PSW.3	Select Blank bit 0
OV	PSW.2	Overflow Flag
_	PSW.1	
P	PSW.0	Parity Flag

- รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาท์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุต หรือการเอาต์พุตข้อมูลได้ การดำเนินการใดๆที่เกี่ยวกับพอร์ตทั้งสี่นี้จะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรส 8 บิตบนของหน่วยความจำ ส่วนพอร์ต P0 นั้นในช่วงเริ่มแรกจะเป็นค่าของแอดเดรส 8 บิต ล่างของหน่วยความจำช่วงเวลาต่อมาจึงจะนำพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือส่งข้อมูลกับหน่วยอุปกรณ์ ภายนอก สำหรับพอร์ต P3 นั้นนอกเหนือจากจะใช้ในฐานะของพอร์ตอินพุต/เอาต์พุตเช่นปกติแล้ว ยังนำมาใช้ในฐานะ บัสควบคุมเกี่ยวกับสัญญาณอินเตอร์รัปได้อีกด้วย

- รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุดและแยกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดยซีพียูจะทำการจัด การเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

- รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรโซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราการทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟลกสภาวะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

ตารางที่ 2.2 รีสเตอร์ที่เกี่ยวข้องกับพอร์ต

ชื่อ	ตำแหน่ง	ความหมาย
SMOD	PCON.7	บิตทวีคูณของอัตราบอดปกติ
-	PCON.6	
-	PCON.5	
-	PCON.4	
GF1	PCON.3	แฟลกสำหรับให้ผู้ใช้ ใช้งานทั่วไป Flag 0
GF0	PCON.2	แฟลกสำหรับให้ผู้ใช้ ใช้งานทั่วไป Flag 1
PD	PCON.1 บิต	สำหรับการกำหนด Power down
IDL	PCON.0	บิตสำหรับการกำหนด Idel mode

บิต PD (Power down) เป็นการกำหนดให้ลดกำลังไฟฟ้าที่จ่ายให้กับส่วนของโปรเซสเซอร์ภายในลงโดยยังคงมีกำลังไฟฟ้าจ่ายให้กับส่วนหน่วยความจำข้อมูลภายในผ่านทางขาสัญญาณ RST วิธีการนี้มักนำมาใช้ในกรณีที่มีการตรวจสอบการไม่มีกำลังไฟฟ้า (Power failure) โดยวงจรตรวจสอบภายนอกจะต้องมีการอินเตอร์รัปเข้ามา เพื่อทำการเก็บข้อมูลที่กำลังประมวลผลอยู่ก่อนและเมื่อมีกระแสไฟฟ้าจ่ายให้เป็นปกติแล้ว จึงค่อยนำข้อมูลนั้นมาประมวลผลต่อไป

บิต IDL (Idle mode) เป็นการกำหนดให้โปรเซสเซอร์หยุดการทำงานชั่วขณะ (Sleep) และจะกลับมาอยู่ในสภาพปกติอีกครั้งเมื่อ ทำการรีเซตทางฮาร์ดแวร์ หรือมีการอินเตอร์รัปอย่างใดอย่างหนึ่งเกิดขึ้นการทำงานในลักษณะนี้ สามารถเกิดขึ้นได้ ก็เนื่องจากว่าสภาวะการหยุดการทำงานชั่วขณะนั้น เป็นเพียงการห้ามไม่ให้มีสัญญาณนาฬิกาจ่ายให้ส่วนของ โปรเซสเซอร์เท่านั้น ส่วนของวงจรการอินเตอร์รัปพอร์ตอนุกรมและวงจรนับ/จับเวลา ยังคงมีสัญญาณนาฬิกาอยู่เป็นปกติ

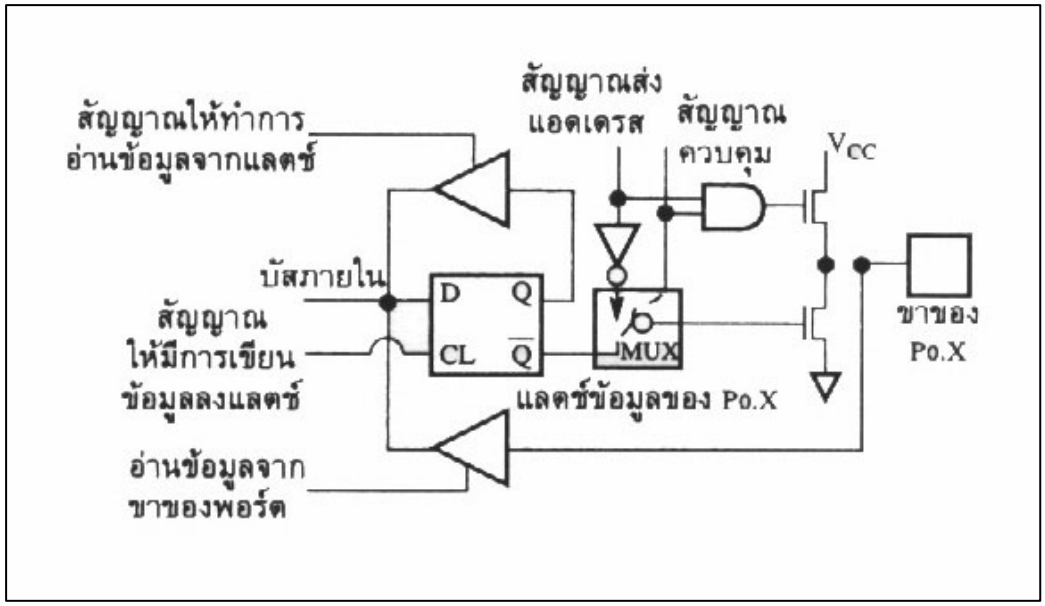
-รีจิสเตอร์ IP,IE,TMOD,TMOD,SCON

เป็นกลุ่มของรีจิสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของการอินเตอร์รัปต่างๆ ของ 8051

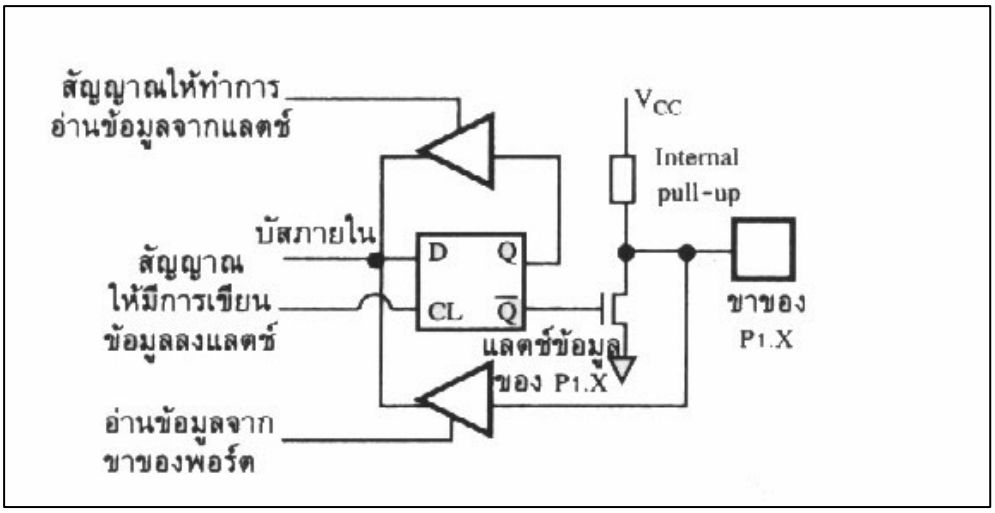
2.1.3 พอร์ตอินพุตและพอร์ตเอาท์พุต

ระบบ I/O ของ 8051

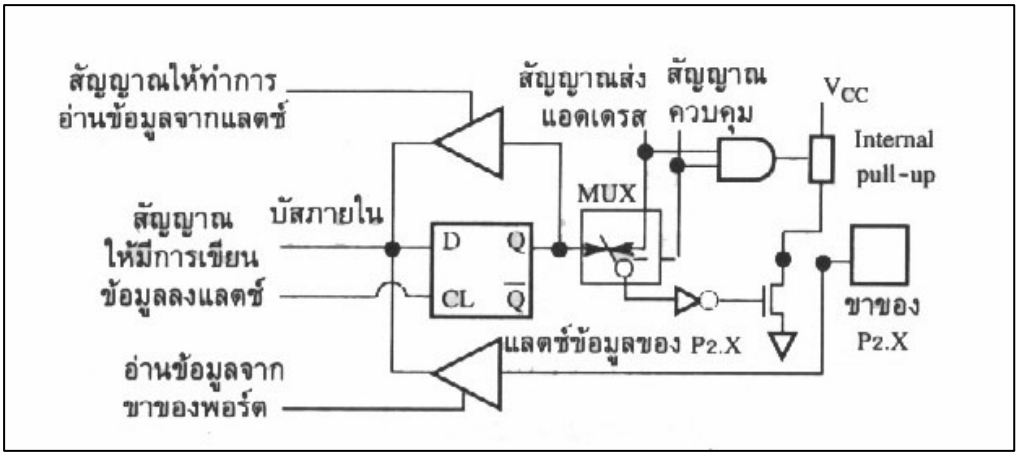
จากองค์ประกอบของชิป 8051 เราจะพบว่ามีพอร์ต I/O ที่รับส่งข้อมูลได้ 2 ทิศทางขนาด 8บิตอยู่ 4 พอร์ต ซึ่งพอร์ตแต่ละตัวนี้จะมีคุณสมบัติพิเศษเฉพาะตัว ดังรูปที่ 5 ถึง รูปที่ 8 ซึ่งแสดงแผนภาพทางตรรกะสำหรับบิตหนึ่งๆ ในพอร์ตแต่ละตัว เราจะเห็นได้ว่าพอร์ตเหล่านี้ (P0-P3) มีคุณสมบัติทาง logic และทางไฟฟ้าที่แตกต่างกันเล็กน้อย พอร์ตแต่ละตัวจะมีแลตช์ข้อมูล ซึ่งจะทำหน้าที่เก็บข้อมูลที่เข้าหรือออกจากพอร์ต โดยแลตช์ข้อมูลนี้สามารถนำข้อมูลจากขาของพอร์ตหรือจากบัสข้อมูลของไมโครโปรเซสเซอร์เข้ามาเก็บได้ และแลตช์ข้อมูลนี้ยังสามารถทำการส่งข้อมูลไปยังบัสข้อมูลของไมโครโปรเซสเซอร์ หรือไปยังขาของพอร์ตได้ เราจะเห็นได้จากรูปที่ 2.5 และ 2.7 ว่าบิตในพอร์ตเบอร์ 0 และพอร์ตเบอร์ 2 มี Controlled pull-up ซึ่งรูปแบบการทำงานของมันจะขึ้นอยู่กับโหมดการทำงานของพอร์ต



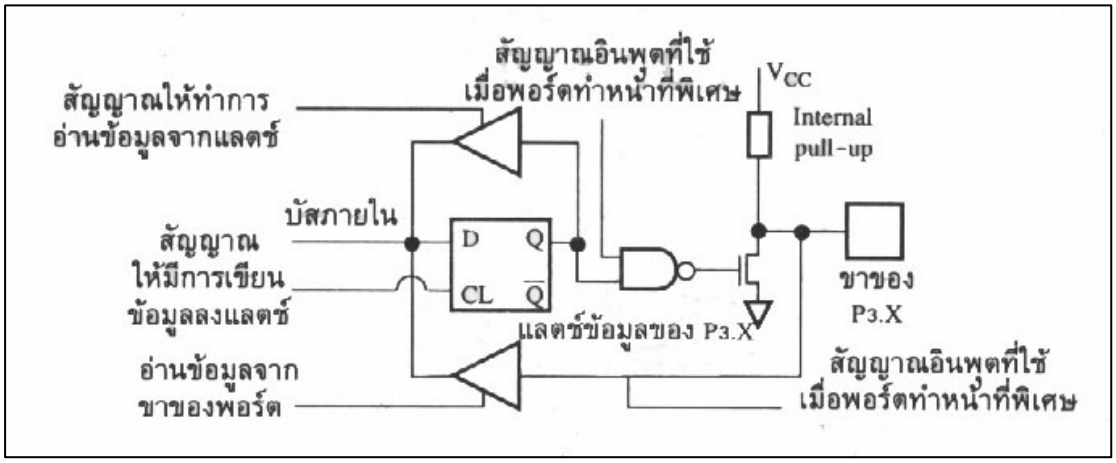
รูปที่ 2.5 แผนผังทาง logic สำหรับบิตหนึ่งๆ ในพอร์ต P0



รูปที่ 2.6 แผนผังทาง logic สำหรับบิตหนึ่งๆ ในพอร์ต P1



รูปที่ 2.7 แผนผังทาง logic สำหรับบิตหนึ่งๆ ในพอร์ต P2



รูปที่ 2.8 แผนผังทาง logic สำหรับบิตหนึ่งๆ ในพอร์ต P3

ถ้ามีการเชื่อมต่อ 8051 กับหน่วยความจำภายนอก (ซึ่งอาจเป็น ROM หรือ RAM) เราจะพอร์ตเบอร์ 0 และพอร์ตเบอร์ 2 มาใช้ในการเก็บค่าแอดเดรสที่ใช้อ้างอิงหน่วยความจำนี้ นอกจากนี้เรายังจะนำพอร์ตเบอร์ 0 มาใช้ในการแลกเปลี่ยนข้อมูลกับหน่วยความจำนี้ด้วย ซึ่งหมายความว่าพอร์ตเบอร์ 0 จะมีถึง 2 หน้าที่ หน้าที่แรกของพอร์ตเบอร์ 0 คือ ทำการส่งค่าไบต์ล่างของแอดเดรสขนาด 16 บิต (ที่เป็นตำแหน่งของหน่วยความจำภายนอกที่ต้องการอ้างอิง) หน้าที่ที่สองคือ ทำการรับข้อมูลจากหน่วยความจำหรือส่งข้อมูลไปยังหน่วยความจำที่อ้างอิง พอร์ตเบอร์ 0 เพียงตัวเดียวสามารถทำงานทั้งสองอย่างได้อย่างไร เราทราบดีว่า ไมโครโปรเซสเซอร์ จะทำการอ้างอิงหน่วยความจำใน cycle แรก และแลกเปลี่ยนข้อมูลกับหน่วยความจำใน cycle ที่ 2 ของการทำงาน เพราะฉนั้น การอ้างอิงหน่วยความจำและการส่งข้อมูลจะเกิดในเวลาที่ต่างกัน เราจึงสามาถกำหนดให้บัสที่เชื่อมต่อกับพอร์ตเบอร์ 0 ทำหน้าที่เป็นบัสข้อมูลในเวลาหนึ่งและเป็นบัสแอดเดรสในอีกเวลาหนึ่งได้ การทำงานแบบสลับหน้าที่นี้เรียกว่า multiplex และเราเรียกบัสที่ทำงานเช่นนี้ว่า multiplexedaddress data bus โดยในตอนแรกเราจะนำค่าแอดเดรสขนาด 16 บิต ที่ใช้ในการอ้างอิงหน่วยความจำภายนอกไปให้กับไบต์บนและไบต์ล่างของบัสแอดเดรส ต่อมาเราจะนำค่าแอดเดรสไบต์ล่างขนาด 8 บิตไปเก็บลงในแลตช์ข้อมูล (ที่เป็น RAM ภายนอก) และในช่วง cycle ที่ 2 ของการทำงานเราก็จะนำ 8 บิตล่างของบัสแอดเดรสนี้มาใช้เป็นบัสข้อมูลเพื่อใช้ในการโอนย้ายข้อมูล ไมโครโปรเซสเซอร์ ที่มีบัสแบบบัสแอดเดรสข้อมูลร่วมจะใช้สัญญาณ ALE (address latch enable) ในการบอกกับแลตช์แอดเดรส (address latch) ว่าให้มันทำการเก็บค่าแอดเดรส 8 บิตล่างที่อยู่ในพอร์ตเบอร์ 0 (ที่เชื่อมต่อกับบัสแบบบัสแอดเดรสข้อมูลร่วมซึ่งทำหน้าที่เป็นบัสแอดเดรสใน

เวลานั้น) และหลังจากที่แลตช์แอดเดรสได้ทำการเก็บค่าแอดเดรส 8 บิตล่างลงในหน่วยความจำแล้ว 8 บิตล่างของบัสก็จะเปลี่ยนหน้าที่เป็นบัสข้อมูล แต่บัสแอดเดรส 8 บิตบนที่เชื่อมต่อกับพอร์ตเบอร์ 2 จะไม่ถูกเปลี่ยนแปลง เนื่องจากเราไม่มีการนำบัสแอดเดรส 8 บิตบนนี้ไปทำหน้าที่เป็นบัสข้อมูล ดังนี้นจึงไม่ต้องทำการเก็บค่าแอดเดรส 8 บิตบนนี้ เราจะพบว่า ไมโครโปรเซสเซอร์ เบอร์ 8085 ของบริษัท intel ได้นำบัสแอดเดรสข้อมูลร่วมนี้มาใช้เช่นเดียวกัน เทคนิคนี้จะช่วยลดจำนวนขาที่ต้องใช้สำหรับบัสแอดเดรสและบัสข้อมูลได้ เราสามารถนำพอร์ตเบอร์ 3 มาทำงานเป็นพอร์ต I/O สองทิศทางแบบขนานได้ดังพอร์ต เบอร์ 0 และพอร์ตเบอร์ 2 และเรายังสามารถนำพอร์ตเบอร์ 3 นี้มาใช้งานในหน้าที่พิเศษอื่นๆ ได้ด้วย ดังตารางด้านล่างแสดงหน้าที่ต่างๆของแต่ละบิตในพอร์ตเบอร์ 3 ในโหมดการทำงานปกติ และโหมดการทำงานพิเศษ

ตารางที่ 2.3 แสดงหน้าที่พิเศษของขาต่างๆในพอร์ตเบอร์ 3

เบอร์ขาของพอร์ต	หน้าที่ปกติ	หน้าที่พิเศษ
P3.0	บิต 0	RXD (พอร์ตอนุกรมที่ทำหน้าที่รับข้อมูล)
P3.1	บิต 1	TXD (พอร์ตอนุกรมที่ทำหน้าที่ส่งข้อมูล)
P3.2	บิต 2	(INT0)' (สัญญาณอินเตอร์รัปภายนอกเบอร์ 0)
P3.3	บิต 3	(INT1)' (สัญญาณอินเตอร์รัปภายนอกเบอร์ 1)
P3.4	บิต 4	T0 (สัญญาณอินพุตเบอร์ 0 ที่ป้อนให้กับเคาน์เตอร์ไทเมอร์)
P3.5	บิต 5	T1 (สัญญาณอินพุตเบอร์ 1 ที่ป้อนให้กับเคาน์เตอร์ไทเมอร์
P3.6	บิต 6	(WR)' (สัญญาณให้ทำการเขียนข้อมูลลงหน่วยความจำภายนอกที่ใช้เก็บข้อมูล)
P3.7	บิต 7	(RD)' (สัญญาณให้ทำการอ่านข้อมูลจากหน่วยความจำภายนอกที่ใช้เก็บข้อมูล)

ขา 2 ขาของพอร์ตเบอร์ 3 จะถูกนำมาใช้งานร่วมกับพอร์ตเบอร์ 0 และ พอร์ตเบอร์ 2 เมื่อ 8051 มีการใช้หน่วยความจำภายนอกซึ่งได้แก่ ขา P3.6 โดยขานี้จะเป็นขาส่งสัญญาณให้ทำการเขียนข้อมูลลงในหน่วยความจำภายนอก (ขา (WR)') และขา P3.7 โดยขานี้เป็นขาส่งสัญญาณให้ทำการอ่านข้อมูลจากหน่วยความจำภายนอก (ขา (RD)') เราจะไม่นำขาทั้งสองนี้มาใช้ถ้าหน่วยความจำ

ภายนอกเป็นหน่วยความจำสำหรับเก็บโปรแกรม เนื่องจากว่าเราไม่สามารถเขียนข้อมูลลงในหน่วยความจำประเภทนี้ แต่เราจะอ่านข้อมูลจากหน่วยความจำสำหรับเก็บโปรแกรมนี้ได้โดยใช้สัญญาณควบคุม (PSEN)'

ขาอื่นๆ ในพอร์ตเบอร์ 3 จะถูกนำมาใช้ในการรับและส่งข้อมูลแบบอนุกรม (ขา P3.0 และขา P3.1) รับสัญญาณอินเตอร์รัปจากภายนอก (ขา INT0 และ INT1) และป้อนสัญญาณอินพุตจากภายนอกไปให้กับเคาน์เตอร์ไทเมอร์ ขนาด 16 บิต (ขา T0 และ T1) จากรูปที่ 6 เราจะพบว่าพอร์ตเบอร์ 1 มี controlled pull-up ซึ่งโดยทั่วไปจะถูกใช้ในการโอนย้ายข้อมูลแบบขนานทีละ 8 บิต จากบล็อกไดอะแกรมที่แสดงสถาปัตยกรรมของ 8051 เราจะเห็นได้ว่าไม่มีรีจิสเตอร์ควบคุมการทำงานพอร์ต I/O แต่ละตัวเลย พอร์ต I/O เหล่านี้จะรู้ได้อย่างไรว่ามันต้องทำงานในโหมดอ่านหรือเขียนข้อมูล คำตอบคือ คำสั่งที่ใช้งานพอร์ต I/O จะเป็นตัวกำหนดการทำงานของพอร์ต เช่น คำสั่งอ่านข้อมูลจากพอร์ต I/O (ที่อ่านข้อมูลทีละบิตหรือทีละไบต์) จะทำให้ 8051 ทำการอ่านข้อมูลจากพอร์ต I/O และคำสั่งเคลื่อนย้ายข้อมูลไปยังรีจิสเตอร์ในพอร์ตส่งข้อมูลออกก็จะเป็นการเขียนข้อมูลลงในพอร์ตส่งข้อมูลออกของ I/O

เนื่องจากพอร์ตเบอร์ 0 และพอร์ตเบอร์ 2 มี controlled pull-up ดังนั้นก่อนที่จะให้พอร์ตทำการรับข้อมูล เราจะต้องทำการเปลี่ยนค่าในบิตต่างๆ ของพอร์ตนั้นให้มีค่าเป็น 1 จากนั้นอุปกรณ์ภายนอกที่จะส่งข้อมูลไปยังพอร์ตก็จะทำการเปลี่ยนแปลงค่าในบิตเหล่านี้ให้เป็น 0 หรือคงค่าเดิมไว้ และเมื่อเราทำการอ่านข้อมูลจากพอร์ตเราก็จะได้ข้อมูลที่ถูกส่งออกจากอุปกรณ์ภายนอก มีคำสั่ง 2 ชนิดที่เราสามารถใช้ในการอ่านข้อมูลจากพอร์ต I/O คำสั่งชนิดแรกจะทำการอ่านข้อมูลจากแลตช์ข้อมูล ซึ่งเราสามารถใช้คำสั่งนี้ในการอ่านข้อมูลจากแลตช์ข้อมูลเปลี่ยนแปลงข้อมูลที่อ่านได้ (ถ้าจำเป็น) และเก็บข้อมูลลงในแลตช์ข้อมูลตามเดิมได้ คำสั่งชนิดนี้มีชื่อว่า read-modify-write

คำสั่งอีกชนิดที่ใช้ในการอ่านข้อมูลจากพอร์ต I/O จะทำการอ่านข้อมูลจากขาของพอร์ตโดยตรง เหตุผลที่เรามีคำสั่งในการอ้างอิง I/O ถึง 2 ชนิด เนื่องจากคำสั่งแบบ read-modify-write ที่ทำการอ้างอิงข้อมูลจากแลตช์ข้อมูลนั้นสามารถหลีกเลี่ยงความผิดพลาดในการอ่านข้อมูลที่อาจเกิดขึ้น

เนื่องจากระดับศักดาไฟฟ้าที่ขาของพอร์ตได้ เช่น ถ้าเรานำบิตหนึ่งของพอร์ตมาเชื่อมต่อกับขาเบสของทรานซิสเตอร์ตัวหนึ่งโดยทรานซิสเตอร์จะทำงานเมื่อเรากำหนดให้บิตนั้นมีค่าทางตรรกะเป็น 1 เมื่อ 8051 ใช้คำสั่งที่ทำการอ่านค่าจากขาของพอร์ตโดยตรง ค่าที่อ่านได้จะมีค่าทางตรรกะเป็น 0 เนื่องจากข้อต่อระหว่างขาเบสกับขาอิมิตเตอร์ (base-emitter junction) ของทรานซิสเตอร์ที่เป็นการไบอัสแบบฟอร์เวิร์ด (forward bias) จะไม่ให้ระดับศักดาไฟฟ้าที่ขาของ

พอร์ตมีขนาดเกิน 0.7 โวลต์ แต่ถ้าเราใช้คำสั่งที่อ่านค่าของบิตนี้จากแลตช์ข้อมูลในพอร์ตเราจะได้ค่าทางตรรกะเป็น 1

พอร์ตอนุกรมของ 8051

คุณสมบัติที่สำคัญข้อหนึ่งของ 8051 คือ พอร์ตแบบอนุกรมที่มีอยู่ในตัวชิป 8051 ในไมโครโปรเซสเซอร์ทั่วไป การเพิ่มพอร์ตอนุกรมจะต้องเพิ่ม UART และวงจรควบคุมไอซีที่สนับสนุนการเชื่อมต่อแบบอนุกรม ซึ่งอุปกรณ์เหล่านี้มีความซับซ้อนและมีราคาค่อนข้างสูง

การเชื่อมต่อแบบอนุกรมของ 8051 นั้นเป็นแบบ full duplex ซึ่งหมายความว่ามันสามารถทำการรับและส่งข้อมูลได้ในเวลาเดียวกัน การเชื่อมต่อแบบอนุกรมนี้จะมีการพักข้อมูลที่รับเข้ามา (receive-buffered) ซึ่งหมายความว่าก่อนที่ข้อมูลชิ้นแรกจะถูกส่งไปยังไมโครโปรเซสเซอร์ พอร์ตอนุกรมจะสามารถรอรับข้อมูลอนุกรมชิ้นที่สองได้ อย่างไรก็ตามข้อมูลชิ้นแรกจะต้องถูกส่งไปยังไมโครโปรเซสเซอร์ก่อนที่ข้อมูลชิ้นที่สองที่รับเข้ามาจะถูกนำไปเก็บในแลตช์ข้อมูล มิฉะนั้นข้อมูลชิ้นแรกที่อยู่ในนั้นจะถูกเขียนทับ

การเชื่อมต่อแบบอนุกรมนี้มีการใช้รีจิสเตอร์ 2 ตัว ที่ทำหน้าที่รับและส่งข้อมูล เราติดต่อกับรีจิสเตอร์ทั้งสองตัวนี้ได้โดยการอ้างอิงรีจิสเตอร์พิเศษตัวหนึ่งที่มีชื่อว่า SBUF ถ้าเราทำการเขียนข้อมูลลง SBUF แสดงว่าเราได้เขียนข้อมูลลงรีจิสเตอร์ส่งข้อมูล แต่ถ้าเราทำการอ่านข้อมูลจาก SBUF แสดงว่าเราอ่านข้อมูลที่อยู่ในรีจิสเตอร์รับข้อมูล จะเห็นได้ว่ารีจิสเตอร์ที่ทำหน้าที่รับและส่งข้อมูลทั้งสองนี้มีค่าแอดเดรสค่าเดียวกัน รีจิสเตอร์ควบคุมพอร์ตอนุกรม (serial port control register :SCON) จะเก็บข้อมูลที่ควบคุมการทำงานของพอร์ตอนุกรม

ตารางที่ 2.4 โหมดการทำงาน

SM1	SM2	SM3	REN	TB8	RB8	TI	RI

SM0 SCON.7	กำหนดโหมดการทำงานของพอร์ตอนุกรม
SM1 SCON.6	กำหนดโหมดการทำงานของพอร์ตอนุกรม
SM2 SCON.5	บิตนี้จะถูกนำมาใช้ในการสื่อสารระหว่างไมโครโปรเซสเซอร์หลายตัวในโหมดการทำงานที่ 2 และ 3 สำหรับในโหมด 2 หรือโหมด 3 นี้ถ้าบิต SM2 มีค่าเป็น 1 บิต RI จะไม่เกิดการเปลี่ยนแปลง หากค่าในบิตที่ 9 ที่รับเข้ามามีค่าเป็น 0 สำหรับในโหมด 1 ถ้าบิต SM2 มีค่าเป็น 1 บิต RI จะไม่เปลี่ยนแปลงถ้าเกิดการรับบิตสตอปผิด สำหรับในโหมด 0 บิต SM2 จะมีค่าเป็น 0
REN SCON.4	บิตนี้จะถูกเซตหรือเคลียร์โดยโปรแกรมเพื่อทำการรับ/ไม่รับข้อมูล
TB8 SCON.3	บิตที่ 9 ในข้อมูลอนุกรมที่จะถูกส่งไป
RB8 SCON.2	บิตนี้จะมีค่าเท่ากับบิตที่ 9 ในข้อมูลอนุกรมที่รับเข้ามาในโหมดที่ 2 และโหมดที่ 3 สำหรับในโหมดที่ 1 ถ้าบิต SM2 มีค่าเป็น 0 บิต RB8 จะมีค่าเท่ากับค่าในบิตสตอปของข้อมูลที่รับเข้ามา สำหรับในโหมด 0 บิต RB8 จะไม่ถูกนำมาใช้งาน
TI SCON.1	แฟลก transmit interrupt จะถูกเซตโดยฮาร์ดแวร์ หลังจากได้มีการเลื่อนข้อมูลขนาด 8 บิตออกในโหมด 0 หรือเมื่อได้ทำการส่งข้อมูลจนพบบิตสตอปในโหมดอื่น หลังการส่งข้อมูลแบบอนุกรมค่าในบิตนี้จะถูกเคลียร์โดยซอฟต์แวร์
RI SCON.0	แฟลก receive interrupt จะถูกเซตโดยฮาร์ดแวร์ หลังจากได้มีการเลื่อนข้อมูลขนาด 8 บิตเข้ามาในโหมด 0 หรือที่จุดครึ่งทางของช่องบิตสตอปในโหมดอื่น (ยกเว้นกรณี SM2 มีค่าเป็น 1) หลังการรับข้อมูลแบบอนุกรมค่าในบิตนี้จะถูกเคลียร์โดยซอฟต์แวร์

ตารางที่ 2.5 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม (serial port control register: SCON) เราสามารถทำการอ้างอิงข้อมูลในรีจิสเตอร์นี้ทีละบิตได้

SM0	SM1	โหมดการทำงานของพอร์ตอนุกรม	รายละเอียด	baud rate
0	0	0	รีจิสเตอร์ที่เลื่อนค่าได้	ความถี่ออสซิลเลเตอร์ /12 เปลี่ยนแปลงได้
0	1	1	UART ขนาด 8 บิต	ความถี่ออสซิลเลเตอร์ /64
1	0	2	UART ขนาด 9 บิต	หรือความถี่ออสซิลเลเตอร์ /32
1	1	3	UART ขนาด 9 บิต	เปลี่ยนแปลงได้

- บิต SM0 และ SM1 จะใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม (ซึ่งมีทั้งหมด 4 โหมดคือ โหมด 0,1,2 และ 3) ต่อไปเราจะกล่าวถึงการทำงานของพอร์ตอนุกรมในแต่ละโหมด
- ในโหมด 0 พอร์ตอนุกรมจะเป็นเหมือนรีจิสเตอร์ที่เลื่อนค่าได้ (shift register) ซึ่งจะเลื่อนค่าตามสัญญาณนาฬิกาที่มีอัตราความถี่เท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์ใน 8051 สัญญาณนาฬิกานี้เป็นตัวกำหนดอัตราความเร็วในการเคลื่อนย้ายข้อมูลแบบอนุกรมที่เราเรียกว่า อัตราบอด (baud rate)
- ในโหมดที่ 1 พอร์ตอนุกรมจะรับและส่งข้อมูลขนาด 10 บิต ซึ่งข้อมูลขนาด 10 บิตนี้ประกอบด้วยบิตสตาร์ต บิตข้อมูลขนาด 8 บิต และบิตสตอป 1 บิต ในโหมดการทำงานนี้ timer ตัวหนึ่งใน 8051 จะถูกนำมาใช้กำหนดอัตราเร็วในการเคลื่อนย้ายข้อมูลของพอร์ตอนุกรม
- ในโหมดที่ 2 พอร์ตอนุกรมจะทำการรับและส่งข้อมูลทีละ 11 บิต ซึ่งข้อมูลขนาด 11 บิตนี้ประกอบด้วยข้อมูลขนาด 8 บิต บิตที่ 9 (เป็นบิตที่เราสามารถกำหนดค่า

ได้) บิตสตาร์ต และ บิตสตอป โดยบิตที่ 9 นี้จะมีค่าตรงกับบิต TB8 ของรีจิสเตอร์ควบคุมพอร์ตแบบอนุกรม (SCON) ในโหมดการทำงานนี้เราสามารถเลือก baud rate ให้มีค่าเท่าเป็น 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ใน 8051 ได้

- การทำงานของพอร์ตอนุกรมในโหมดที่ 3 จะทำการรับและส่งข้อมูลทีละ 11 บิต และมีบิตที่ 9 ที่กำหนดค่าได้ โดยบิตนี้จะมีค่าตรงกับบิต RB8 ในรีจิสเตอร์ควบคุมพอร์ตแบบอนุกรม (SCON) โหมดที่ 3 นี้จะทำหน้าที่เช่นเดียวกับโหมดที่ 2 ยกเว้นแต่ว่าอัตราเร็วในการเคลื่อนย้ายข้อมูลสามารถเปลี่ยนแปลงได้

ไมโครโปรเซสเซอร์ 8051 หลายตัวสามารถติดต่อกันได้โดยการเชื่อมต่อกันแบบอนุกรม เมื่อมีข้อมูลส่งมายังไมโครคอนโทรลเลอร์ บิตที่ 9 ในข้อมูลที่รับเข้ามาจะทำให้เกิดการอินเตอร์รัป ซึ่งจะบอกให้ไมโครคอนโทรลเลอร์รู้ว่าพอร์ตอนุกรมของมันได้ทำการรับข้อมูลเข้ามาเก็บในที่พักข้อมูล และมันควรทำการโอนย้ายข้อมูลนี้เพื่อนำไปใช้ต่อไป ใน 8051 baud rate ในโหมด 1 และโหมด 3 จะถูกกำหนดโดยอัตราการเกิดโอเวอร์โฟลว์ของไทเมอร์เบอร์ 1 และเราสามรถเพิ่ม baud rate ให้เป็นสองเท่าได้โดยการกำหนดค่าของบิต SMOD ในรีจิสเตอร์ควบคุมกำบังให้มีค่าเป็น 1

ไมโครคอนโทรลเลอร์ มีการรวมเอา ROM และ RAM เข้าไปไว้ในตัวเองเพื่อความสะดวกในการใช้งานทางด้านการควบคุม โดยที่ตัว ไมโครคอนโทรลเลอร์ยังสามารถที่จะต่อ ROM และ RAM ภายนอกได้ เพื่อให้มีความยืดหยุ่นในการทำงานมากขึ้นในกรณีที่งานที่ใช้ต้องใช้หน่วยความจำเป็นจำนวนมาก ในส่วนของ I/O นั้นในตัว ไมโครคอนโทรลเลอร์ ได้มีจำนวนพอร์ตที่จะให้ใช้ในการควบคุมเป็นจำนวนมากพอ ดังเช่นของ 8051 จะมีให้ใช้ถึง 4 พอร์ต

2.1.4 อินเตอร์รัป (Interrupt)

ประเภทของการ อินเตอร์รัป

- **External interrupt** การตรวจสอบสัญญาณที่มาอินเตอร์รัป นี้ จะสามารถกำหนดให้มีการตรวจสอบในลักษณะเมื่อได้มีการเปลี่ยนแปลงระดับสัญญาณ (Level-sensitive) ไปแล้ว หรือในช่วงเวลาขณะเริ่มมี การเปลี่ยนแปลงสัญญาณจาก logic สูงไปต่ำ (Edge-sensitive)

- **Internal interrupt** แหล่งกำเนิดสัญญาณนี้จะเป็นวงจรภายใน ไมโครคอนโทรลเลอร์เอง เช่น วงจรนับ/จับเวลา วงจรเชื่อมต่อสัญญาณอนุกรม เป็นต้น

โครงสร้างการ อินเตอร์รัป เกิดได้ 5 ลักษณะ คือ

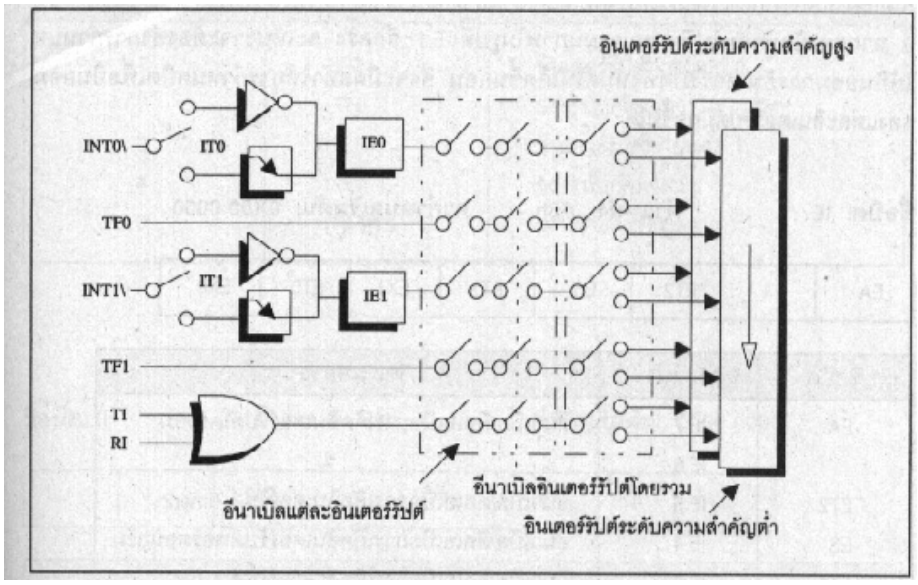
- **INT0 สัญญาณ อินเตอร์รัป จากภายนอก** ทางขาสัญญาณ P3.2 โดย 8051 จะทำการสุ่มตัวอย่าง สัญญาณเมื่อสิ้นสุดทุก Machine Cycle

- **INT1 สัญญาณ อินเตอร์รัป** จากภายนอกทางขาสัญญาณ P3.3 โดย 8051 จะทำการสุ่มตัวอย่าง สัญญาณเมื่อสิ้นสุดทุก Machine Cycle

- **Timer0** สัญญาณการเกิด Overflow ของ Timer 0

- **Timer1** สัญญาณการเกิด Overflow ของ Timer 1

- **Serial Port** การเกิดอินเตอร์รัป ที่เกิดขึ้นจากการรับ/ส่งข้อมูลอนุกรม ทำให้มีผผลต่อ flag interrupt RI และ TI ตามลำดับ



รูปที่ 2.9 โครงสร้างระบบการอินเตอร์รัปของ 8051

จากแผนภาพโครงสร้างระบบอินเตอร์รัป ของ 8051 จะเห็นว่าเมื่อเกิดการอินเตอร์รัป สัญญาณ ต่างๆขึ้นจะส่งผลให้มีการควบคุมเพื่อสั่งให้โปรเซสเซอร์ กระโดดไปทำงานที่ตำแหน่งแอดเดสต่างๆ ตามประเภทของแหล่งกำเนิดสัญญาณอินเตอร์รัป ที่เกิดขึ้น ซึ่งปรกติควรมีการสร้างโปรแกรมเหล่านี้ไว้ เพื่อทำหน้าที่ย่อยบริการ อินเตอร์รัป

การกำหนดให้ 8051 สามารถตอบรับการอินเตอร์รัป แต่ละประเภท ทำได้โดยกำหนดบิตของข้อมูลที่เกี่ยวข้องซึ่งมักอยู่ภายใน รีจิสเตอร์ TCON และ SCON หากได้ว่ามีการกำหนด ค่าของบิต ซึ่งอยู่ภายในรีจิสเตอร์ IE (Intereupt Enable Register) ด้วยแล้ว ก็สามารถตอบรับการอินเตอร์รัป ของสัญญาณนั้นๆ ได้ นอกจากหนี้ตามแผนภาพในรูป ยังแสดงให้เห็นว่าสัญญาณอินเตอร์รัป แต่ละ ประเภท ยังสามารถกำหนด piority ของการอินเตอร์รัป ได้ 2 ลักษณะ คือ High Low piority กล่าว คือขณะที่ประมวลผลอยู่ภายในส่วนของโปรแกรมย่อย บริเวณอินเตอร์รัป ของสัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็สามารถถูกอินเตอร์รัป ที่มี piorrity สูงกว่าได้ แต่หากว่าเป็นสัญญาณอินเตอร์รัป ที่มี piority เดียวกันหรือต่ำกว่าแล้วก็จะต้องรอให้เสร็จสิ้นการประมวลผลที่ดำเนินอยู่ก่อน

การควบคุมอินเตอร์รัป

ตามโครงสร้างที่ด้านการจัดการอินเตอร์รัป ของ 8051 สามารถกำหนดเรียกเพื่อยินยอมหรือไม่ยินยอม (Enable/Disabble) ให้มีการอินเตอร์รัป แต่ละสัญญาณได้ โดยใช้วิธีการกำหนด ค่าของบิตภายใน รีจิสเตอร์ IE

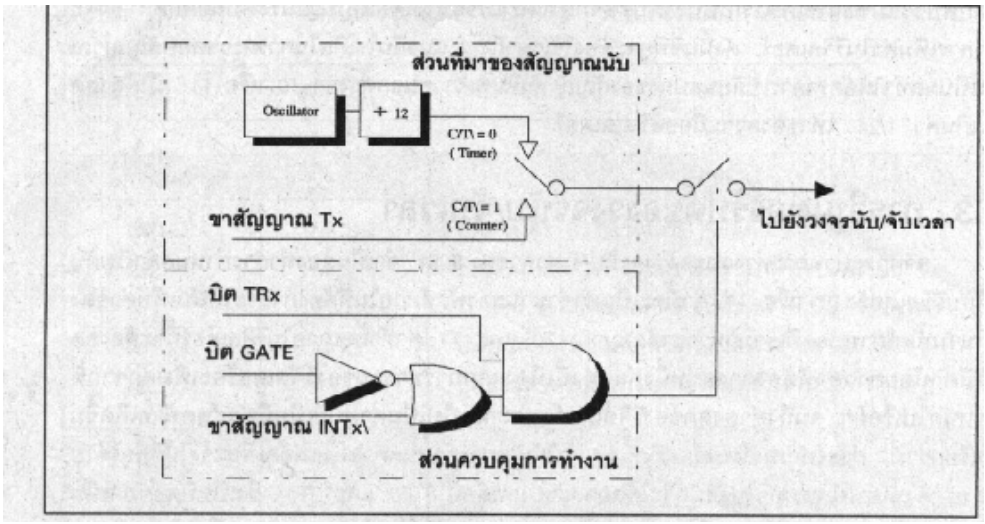
วงจรนับ/จับเวลา

8051 ประกอบด้วย รีจิสเตอร์ ขนาด 16 บิต จำนวน 2 ตัว คือ T0 (Timer0) และ T1 (Timer1) ซึ่งสามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถควบคุมให้ทำหน้าที่เป็นตัวจับเวลา (Timer) เพื่อนับจำนวน plus สัญญาณนาฬิกาภายใน หรือควบคุมให้ทำหน้าที่เป็นตัวนับ (Counter) เพื่อนับจำนวน plus ของระบบ ได้ ภายใน รีจิสเตอร์ แต่ละตัวยังสามารถแยกออกได้เป็น รีจิสเตอร์ขนาด 8 บิต คือ TH0, TL0, TH1 และ TL1 โดยการทำงานของ รีจิสเตอร์ ทั้ง 2 ตัวนี้มีผลมาจากการกำหนดค่าของ บิต ที่อยู่ภายใน TMOD (Timer mode control register) และ TCON (Timer/Counter control register) บิต ต่างๆ ภายใน รีจิสเตอร์ TMOD บิต ต่างๆ ภายใน รีจิสเตอร์ TCON

การอินเตอร์รัป วงจรนับตรวจ/จับเวลา

จากกระบวนการทำงานของวงจรนับ/จับเวลาของ 8051 จำเป็นต้องกำหนดค่าเริ่มต้นให้กับ รีจิสเตอร์ T0 หรือ T1 ค่านี้เป็นค่าจำนวน plus ภายในที่จะต้องนับหรือค่าของจำนวน plus ภายนอกที่เข้ามาทางขาสัญญาณสัญญาณ T0 หรือ T1 ค่าตัวเลขภายใน รีจิสเตอร์ นี้จะต้องลดให้มีค่าที่น้อยกว่า ค่าที่ต้องการอยู่หนึ่งค่า ทั้งนี้เนื่องจากการทำงานของ รีจิสเตอร์ จะเพิ่มค่าจากที่กำหนดไปเรื่อยๆ จนถึง ค่าสูงสุดของ รีจิสเตอร์ และกลับไปเป็นค่า 0 เมื่อมีการเกิดโอเวอร์โฟลว์เกิดขึ้น ทำให้

เกิดการกำหนดค่าแฟลก เพื่อแจ้งให้ซีพียู ได้รับทราบ ดังนั้นโปรแกรมทั่วไปจึงมักใช้สภาวะของแฟลกนี้ (TF0 และ TF1) ซึ่งเป็นบิต อยู่ภายใน รีจิสเตอร์ TCON เพื่อตรวจสอบว่ากระบวนการนั้นได้เสร็จสิ้นลงแล้ว หรือใช้เพื่อ ทำการอินเตอร์รัปโปรแกรมต่อไป ส่วนควบคุมการทำงานของวงจรนับ/จับเวลา ซึ่งประกอบด้วยส่วนของการกำหนดที่มาของสัญญาณ (Timer) หรือ (Counter) และ บิต หรือขาสัญญาณสำหรับการหยุดหรือทำงานของวงจรนับ

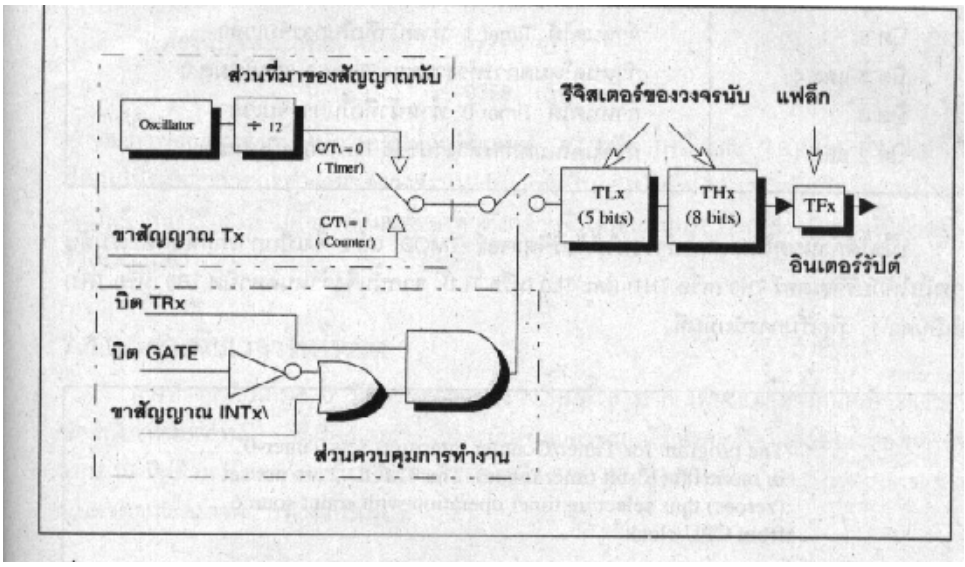


รูปที่ 2.10 ส่วนควบคุมการทำงานของวงจรนับจับเวลา

การทำงานเป็นตัวจับเวลา

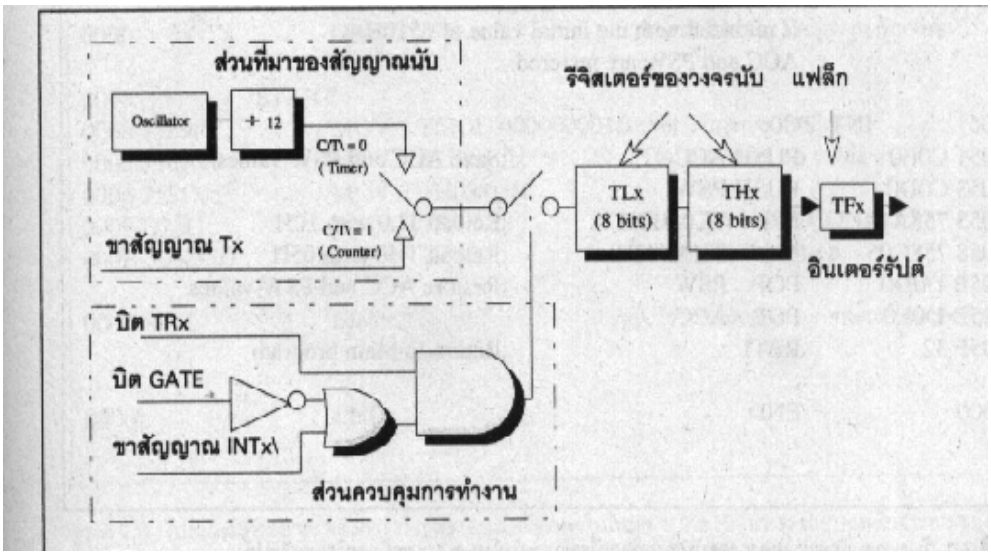
ก่อนที่ทำงานเป็นวงจรจับเวลา ต้องมีการกำหนดค่าให้อยู่ในสภาวะดังตารางก่อน

-การจับเวลาใน โหมด 0: การทำงานใน โหมด 0 วงจรนับจับเวลาจะทำหน้าที่เป็นตัวนับขนาด 13 บิต (โดยใช้ รีจิสเตอร์ TH0 หรือ TH1 เป็นตัวนับขนาด 8 บิต และ รีจิสเตอร์ TL0 หรือ TL1 มีขนาด 5 บิต) ตามลักษณะของแผนภาพในรูปที่ 2.11



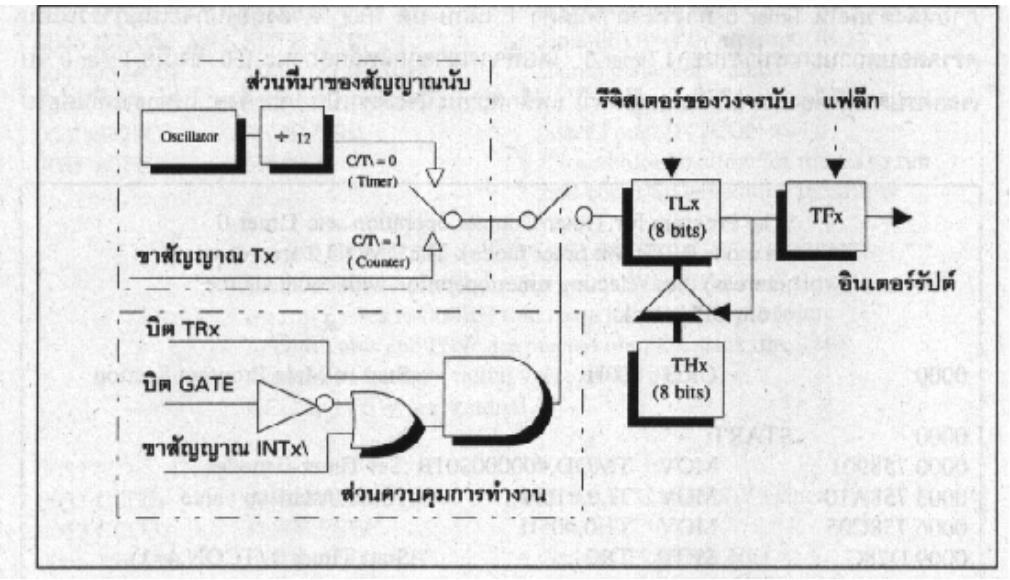
รูปที่ 2.11 การทำงานของ Timer/Counter 0 หรือ 1 ในโหมด 0

-การจับเวลาใน โหมด 1 : การทำงานใน โหมด 1 มีความคล้ายคลึงใน โหมด 0 มาก แตกต่างกันที่ โหมด 1 เป็นตัวนับขนาด 16 บิต เต็ม ดังรูปที่ 2.12



รูปที่ 2.12 การทำงานของ Timer/Counter 0 หรือ 1 ในโหมด 1

-การจับเวลาในโหมด 2 : การทำงานใน โหมด 2 ของวงจรนับ/จับเวลาแตกต่างกันออกไปเพียงใช้ รีจิสเตอร์ TL0(TL1) เป็น ตัวนับขนาด 8 บิต ส่วน รีจิสเตอร์ TH0(TH1) เก็บค่าเริ่มต้นของการนับไว้ ดังรูป 2.13



รูปที่ 2.13 การทำงานของ Timer/Counter 0 หรือ 1 ในโหมด 2

-การจับเวลาใน โหมด 3 : การทำงานใน โหมด 3 จะสามารถใช้ได้เฉพาะกับ Timer 0 เท่านั้น หากว่านำไปกำหนดให้กับ Timer 1 จะทำให้หยุดการทำงานไป เมื่อ Timer 0 ได้รับการกำหนดทำงานในโหมด 3 จะมีผลทำให้ รีจิสเตอร์ ของมันแยกการทำงานเป็นอิสระ โดยรีจิสเตอร์ TL0 จะถูกควบคุมจากบิต ภายใน รีจิสเตอร์ TCON และ ขาสัญญาณ INT0 ดังแสดงในรูป และเมื่อมีการโอเวอร์โฟลว์เกิดขึ้น จากค่า 0FFH เป็น 00H ก็ จะมีผลให้ flag TF0 มีการเปลี่ยนแปลงเกิดขึ้น สำหรับ รีจิสเตอร์ TH0 จะถูกกำหนดให้ทำงานในแบบ ของตัวจับเวลาภายใต้การควบคุมของบิต TR1 ในรีจิสเตอร์ TCON เท่านั้น และหากเกิดโอเวอร์โฟลว์ จะมีผลเฉพาะต่อ flag TF1 ในส่วน Timer 1 ขณะเมื่อ Timer 0 ถูกกำหนดให้ทำงานโหมด 3 ก็ยังจะสามารถทำงานในโหมดอื่นๆ ที่ไม่ใช่ โหมด 3 ได้ เช่นเดิม ยกเว้นจะไม่มีการอินเตอร์รัปเกิดขึ้นเท่านั้น (เนื่องจากflag TF1 ถูกใช้โดย Timer 0 ไปแล้ว)

การทำงานเป็นตัวนับสัญญา

การใช้งานในลักษณะตัวนับ (counter) โดยหลักแล้วจะเหมือนกับลักษณะการทำงานเป็น ตัวจับเวลา (Timer) ดังได้กล่าวในหัวข้อที่ผ่านมา ข้อแตกต่างประการเดียวคือ แทนที่จะนับ plus สัญญาณภายในและผ่านวงจรหาร 12 มาเป็นการนับ plus สัญญาณทางขาสัญญาณ T0 (P3.4) ให้ กับ Timer 0 หรือขาสัญญาณ T1 (P3.5) ให้กับ Timer 1 เท่านั้น นอกจากนี้ก่อนการเริ่มต้นใช้งานจะ ต้องกำหนดค่าของ บิต C/T ภายใน รีจิสเตอร์ TCON ให้มีค่าเป็น 1 เสียก่อน

วงจรนับ/จับเวลา 2 (Timer2)

โหมด การทำงานของ Timer 2 ประกอบด้วย

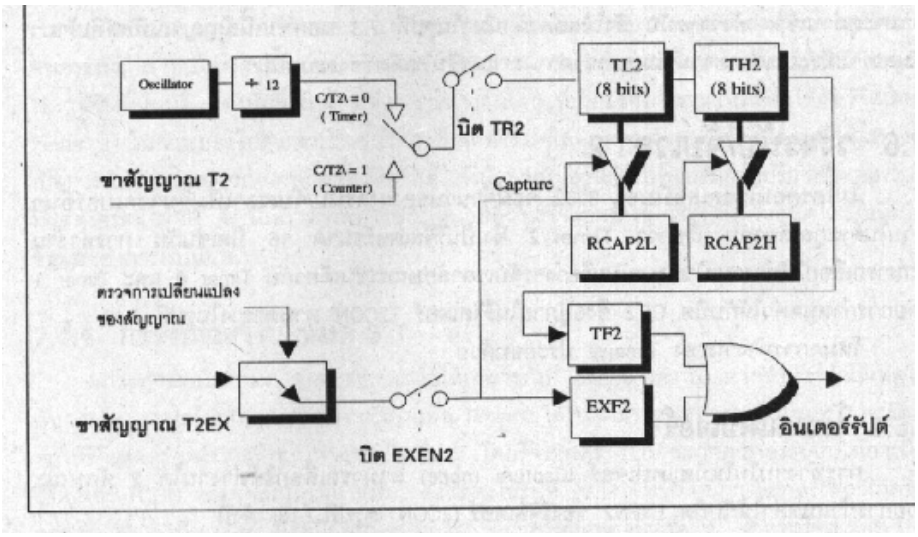
-**Capture mode**: สามารถเลือกใช้งานได้ 2 ลักษณะ ด้วยการกำหนดให้กับ บิต EXEN2 ของ รีจิสเตอร์ T2CON ดังนี้

1. เมื่อกำหนด บิต EXEN2 เป็น 0 Timer 2 ยังทำงานเป็นวงจรนับ/ตรวจจับเวลา เมื่อมีการ โอเวอร์โฟลว์ขึ้น บิตใน รีจิสเตอร์ TF2 จะถูกเซต และสามารถนำไปสร้างการอินเตอร์รัปขึ้นได้

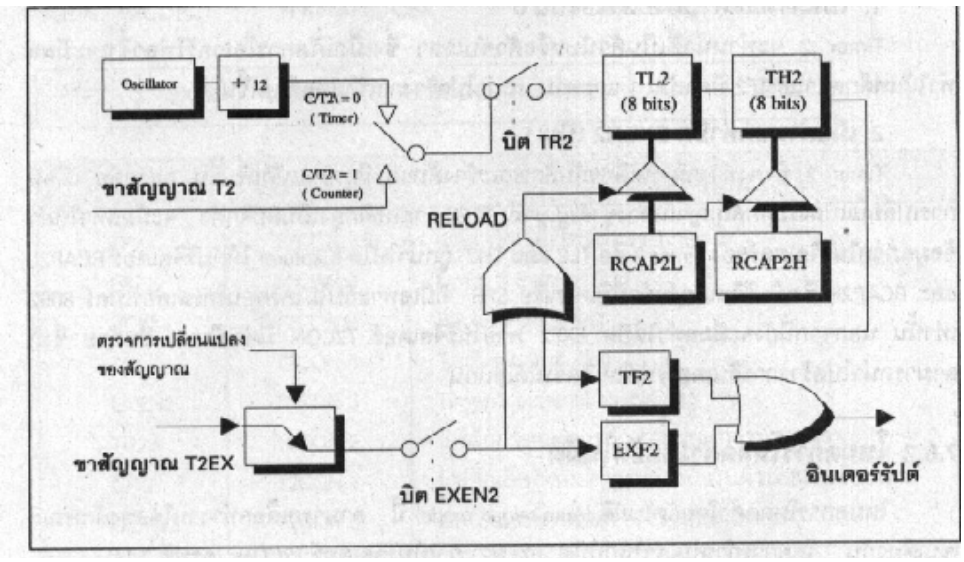
2. เมื่อกำหนดค่า บิต EXEN2 เป็น 1 การทำงานจะครอบคลุมการทำงานลักษณะข้างต้น แต่จะเพิ่มเมื่อมีการเปลี่ยนแปลงระดับ สัญญาณทางขาสัญญาณ T2EX จาก logic สูง ไปเป็น logic ต่ำ จะมีผลทำให้ค่าข้อมูลภายใน รีจิสเตอร์ ของ Timer2 คือ TL2 และ TH2 ถูกนำไปใส่ (Capture) ให้กับ รีจิสเตอร์ RCAP2L และ RCAP2H ซึ่งเป็น รีจิสเตอร์หน้าที่พิเศษ หรือ SFR ที่มีในไมโครคอนโทรลเลอร์ เบอร์ 8052 เท่านั้น นอกจากนี้จะมีผลทำให้ บิต EXF2 ภายใน รีจิสเตอร์ T2CON มีค่าเป็น 1 สามารถนำไปใช้งานในการอินเตอร์รัป ได้เช่นกัน

-**Auto-reload mode**: สามารถทำงานได้ 2 ลักษณะเช่นเดียวกัน

-**Baud rate Generator**: ของ Timer 2 จะมีความแตกต่างจาก Timer 0 และ Timer 1 โดยวงจรรับ และการส่ง สามารถเป็นค่าที่ต่างกันได้ ขึ้นอยู่กับการกำหนดค่าให้กับ บิต TCLK และ RCLK ของ Timer 2 ดังแสดงให้เห็นจากภาพการทำงาน

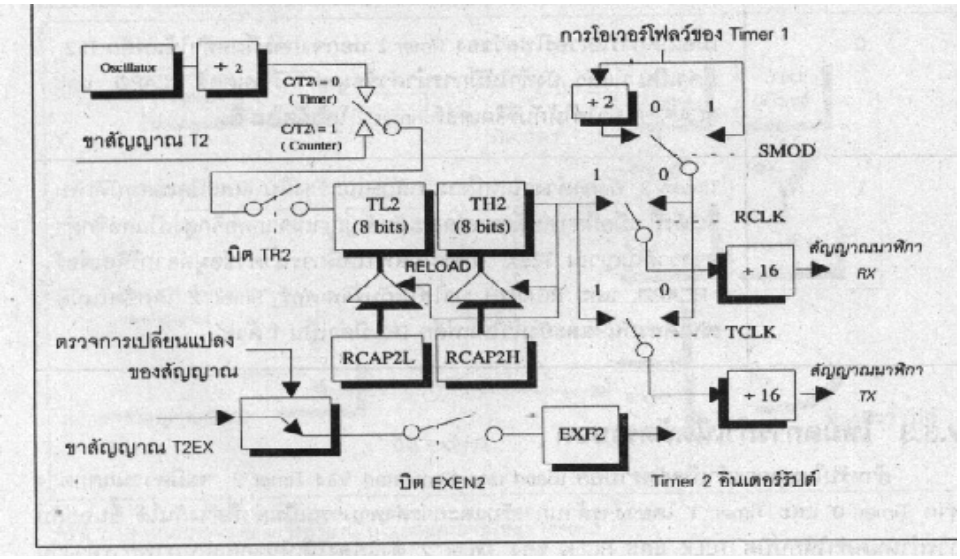


รูปที่ 2.14 การทำงานในโหมดแคปเจอร์ของ Timer 2



รูปที่ 2.15 การทำงานในโหมดโหลดค่าโดยอัตโนมัติของ Timer 2

การทำงานของโหมดนี้คล้ายคลึงกับการทำงานใน Auto-reload mode กล่าวคือค่าในรีจิสเตอร์ TH2 เปลี่ยนแปลงจากค่า 0FFH ไปเป็นค่า 0 หรือที่เรียกว่าโอเวอร์โฟลว์ จะมีผลให้มีการโหลดข้อมูลขนาด 16 บิต จาก รีจิสเตอร์ RCAP2H และ RCAP2L ซึ่งมีการเตรียมค่าล่วงหน้าแล้วโดยอัตโนมัติการจับเวลาใน



รูปที่ 2.16 การทำงานในโหมดการกำเนิดอัตราบอดของ Timer 2

2.2 การรับส่งข้อมูลผ่านระบบการสื่อสารไร้สาย

ระบบเครือข่ายไร้สาย คือ ระบบการสื่อสารข้อมูลที่นำมาใช้ทดแทนหรือเพิ่มต่อกับระบบเครือข่ายแลนใช้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และ คลื่นอินฟราเรด ในการรับและส่งข้อมูลระหว่างคอมพิวเตอร์แต่ละเครื่อง ผ่านอากาศ, ทะลุกำแพง, เพดานหรือสิ่งก่อสร้างอื่นๆ โดยปราศจากความต้องการของการเดินสาย นอกจากนั้นระบบเครือข่ายไร้สายก็ยังมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบ LAN (Local area network) แบบใช้สาย ที่สำคัญก็คือ การที่มันไม่ต้องใช้สายทำให้การเคลื่อนย้ายการใช้งานทำได้โดยสะดวก ไม่เหมือนระบบ LAN แบบใช้สาย ที่ต้องใช้เวลาและการลงทุนในการปรับเปลี่ยนตำแหน่งการใช้งานเครื่องคอมพิวเตอร์

2.2.1พื้นฐานเทคโนโลยีไร้สาย

ข้อมูล (Data) ข่าวสารต่าง ๆ สามารถถูกส่งแบบไร้สาย โดยเปลี่ยนจากสัญญาณวิทยุและสัญญาณโทรทัศน์เป็นเสียงและภาพ หรือเป็นข้อมูลด้านคอมพิวเตอร์ข่าวสารต่าง ๆ จะถูกส่งโดยผสมไปกับคลื่นวิทยุ ซึ่งคลื่นวิทยุเป็นแค่ส่วนหนึ่งในสเปกตรัมคลื่นแม่เหล็กไฟฟ้าในช่วงที่เรียกว่าความถี่วิทยุ (Radio frequency) ข้อมูลทุกชนิดสามารถถูกส่งโดยการใช้ความถี่วิทยุ ความถี่วิทยุมีมากมายไม่ใช่มีแค่คลื่นวิทยุเอเอ็มหรือเอฟเอ็มที่เรารู้จักเท่านั้น

โมดูเลชัน (Modulation) ข้อมูลจะถูกผสมไปกับคลื่นความถี่วิทยุโดยกระบวนที่เรียกว่าโมดูเลชัน เมื่อได้รับคลื่นสัญญาณจะผ่านกระบวนการ ดิโมดูเลชัน (Demodulation) เพื่อแยกเอาข้อมูลออกมา

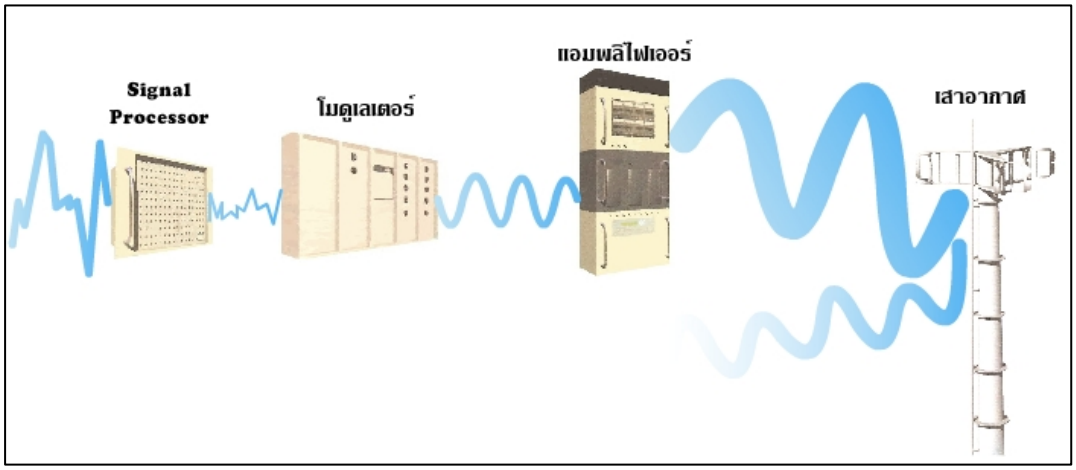
สถานีฐาน (Base Station) ภายในแต่ละเซลล์จะมีสถานีฐานซึ่งจะส่งและรับสัญญาณสื่อสารทั้งรับและส่งไปยังโทรศัพท์มือถือ (หรือโทรศัพท์เซลลูลาร์) ภายในเซลล์นั้น ๆ

อุปกรณ์ส่งสัญญาณและอุปกรณ์รับสัญญาณ คลื่นความถี่วิทยุพร้อมด้วยข้อมูลข่าวสารจะถูกส่งโดยอุปกรณ์ส่งสัญญาณ (Transmitters) และรับโดยปุปกรณ์ที่รียกว่าอุปกรณ์รับสัญญาณ (Receivers)

ข้อมูลหลาย ๆ ชนิดสามารถส่งแบบไร้สายได้ เช่น การส่งข้อมูลคอมพิวเตอร์ เสียงโทรศัพท์ การส่งสัญญาณโทรทัศน์และวิทยุ ในขั้นตอนแรกข้อมูลที่ถูกส่งจะถูกสร้างจากอุปกรณ์ที่สร้างข้อมูล เช่น คอมพิวเตอร์มือถือ สถานีวิทยุ หรือโทรศัพท์มือถือ เป็นต้น สำหรับข้อมูลที่จะถูกส่งมันจะต้องผสมไปกับคลื่นความถี่วิทยุ (RF) (ซึ่งเรียกว่า “สัญญาณ” หรือ Signal) โดยกระบวนการโมดูเลชัน (Modulation) สัญญาณที่จะเป็นตัวส่งข้อมูลเรียกว่าคลื่นตัวนำ (Carrier Wave) ข้อมูลจะถูกผสมไปกับคลื่นตัวนำ โดยอุปกรณ์ที่เรียกว่าโมดูเลเตอร์ (Modulator) ซึ่งเป็นอุปกรณ์ที่มีวิธีการหลายวิธีในการโมดูเลตข้อมูลไปกับคลื่นตัวนำโมดูเลเตอร์อาจจะรวมอยู่กับ

อุปกรณ์ที่สร้างข้อมูลอย่างเช่น โทรศัพท์มือถือ คอมพิวเตอร์มือถือ สัญญาณจะถูกส่งโดยอุปกรณ์ส่งสัญญาณที่จะนำสัญญาณและส่งออกไปโดยผ่านทางอากาศ อุปกรณ์ส่งสัญญาณนั้นมีหลายแบบโดยขึ้นอยู่กับชนิดข้อมูลที่จะส่งระยะทาง และความแรงของสัญญาณ และขนาดนั้นอาจเล็กมากเหมือนที่อยู่ในโทรศัพท์มือถือหรืออาจจะใหญ่มากเหมือนเสาอากาศส่งสัญญาณของโทรทัศน์ อุปกรณ์รับสัญญาณสามารถรับสัญญาณได้โดยตรง หรืออาจผ่านทางระบบเครือข่ายโดยขึ้นกับชนิดของข้อมูลที่ส่ง ในกรณีของโทรศัพท์มือถือ หรือคอมพิวเตอร์มือถือเมื่อจะติดต่อกับอินเตอร์เน็ตมันจะส่งสัญญาณไปที่เครือข่ายและส่งต่อไปนังผู้รับโดยใช้อุปกรณ์ส่งสัญญาณ (Transmitters) ที่จุดรับสัญญาณ เสาอากาศหรือสายอากาศรับสัญญาณที่ส่งมาให้อุปกรณ์รับสัญญาณ เสาอากาศหรือสายอากาศจะรับคลื่นวิทยุที่ต้องการและไม่รับคลื่นที่เหลือ อุปกรณ์รับสัญญาณจะใช้แอมพลิไฟเออร์ (Amplifier) เพื่อเพิ่มความเข้มของสัญญาณเนื่องจากสัญญาณที่รับมานั้นจะอ่อนมาก โมดูเลเตอร์ (หรืออาจเรียกว่าดีโมดูเลเตอร์) จะทำการแปลสัญญาณและแยกคลื่นตัวนำออกจากข้อมูลที่ถูกส่งมาพร้อมกัน เพื่อที่จะเปลี่ยนกลับไปเป็นข้อมูลดังเดิมที่ส่งมา ข้อมูลที่ส่งมายังอุปกรณ์รับสัญญาณ เช่น โทรศัพท์มือถือ ชุดรับโทรทัศน์ หรือคอมพิวเตอร์มือถือ ซึ่งตอนนี้สามารถแสดงข้อมูลที่ส่งมาได้แล้ว

2.2.2 ข้อมูลถูกส่งไปกับคลื่น



รูปที่ 2.17 การส่งข้อมูลผ่านคลื่น

1. ข้อมูลที่ต้องการจะส่งนั้นจะถูกผสมไปกับคลื่นตัวนำ ในกระบวนการ โมดูเลชัน (Modulation)
2. สัญญาณของข้อมูลอาจจะต้องมีการประมวลผลของสัญญาณโดยกระบวนการประมวลผล หรือ Signal Processing เพื่อให้การส่งข้อมูลมีประสิทธิภาพดี โดยขึ้นกับชนิดของข้อมูลที่จะส่ง ตัวอย่างเช่น ในกรณีของการส่งสัญญาณเสียง หลายความถี่ในสัญญาณสามารถที่จะถูกกำจัดออกไป เนื่องจากหูของมนุษย์ไม่สามารถได้ยินเสียงที่มีความถี่สูงหรือต่ำขนาดนั้นได้ ดังนั้นความถี่เหล่านั้นจะถูกกำจัดออกไป ตัวประมวลผลสัญญาณเสียงจะจัดการกับการส่งข้อมูลเสียง และตัวประมวลผลดิจิตอลจะจัดการกับการส่งแบบดิจิตอล ตัวประมวลผลสัญญาณนั้นมีหลายชนิด และมีเทคโนโลยีหลายแบบสำหรับตัวประมวลผล เช่น ชิปในเครื่องคอมพิวเตอร์ เป็นต้น

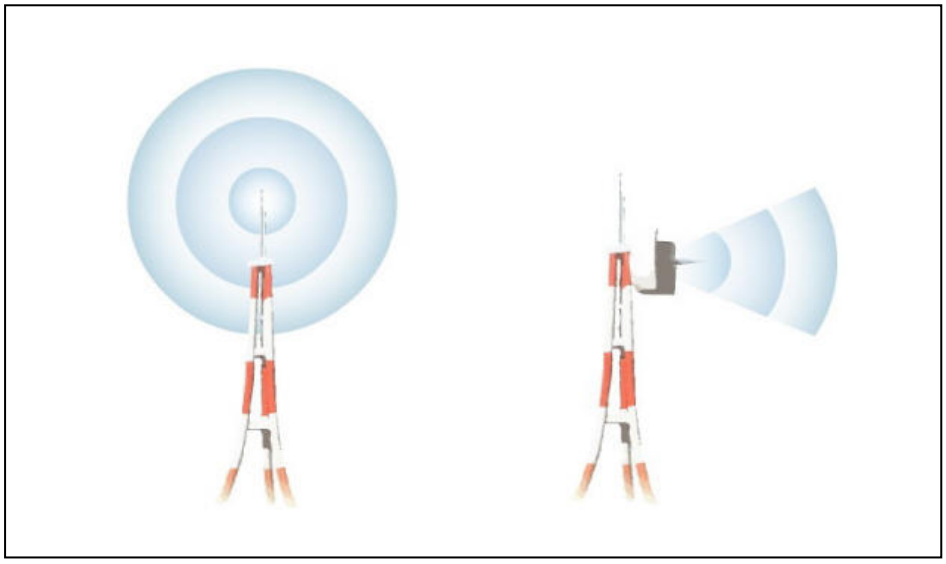
3. ก่อนการส่งสัญญาณ ตัวสัญญาณเองอาจต้องถูกขยายสัญญาณเพื่อทำให้ที่ปลายทางสามารถรับสัญญาณได้ดีขึ้น

4. เพื่อให้เราแน่ใจว่าสัญญาณมีความแรงเพียงพอเสาอากาศที่ถูกติดตั้งจะทำหน้าที่เพิ่มเกน (Gain) ให้กับสัญญาณ เสาอากาศไม่ได้ทำการเพิ่มความแรงของสัญญาณ แต่ถ้าสัญญาณมีการเดินทางไปในทิศทางเดียวกันก็จะทำให้สัญญาณมีความแรงมากขึ้นโดยไม่มีการกระจายของสัญญาณ
5. สาเหตุที่ต้องเพิ่มความเข้มของสัญญาณก่อนการส่งนั้น ก็เนื่องจากจะมีสัญญาณรบกวนที่เรียกว่านอยซ์ (Noise) ซึ่งเป็นคลื่นแม่เหล็กไฟฟ้าชนิดหนึ่งที่เรียกว่า เทอร์มอลนอยซ์ (Thermal Noise) หรือไวท์นอยซ์ (White Noise) ซึ่งปกติจะมาจากการแผ่รังสีของดวงอาทิตย์ ส่วนอีกชนิดหนึ่งเรียกว่าอิมพัลซ์นอยซ์ (Impuise Noise) ซึ่งเกิดจากเครื่องกำเนินแสง เครื่องจักร การส่งสัญญาณจากอุปกรณ์ส่งสัญญาณต่าง ๆ เป็นต้น สำหรับสัญญาณที่เราส่งนั้นจะต้องเข้มกว่าสัญญาณรบกวน อัตราส่วนระหว่างสัญญาณที่ส่งต่อสัญญาณรบกวนเรียกว่า signal-to noise ratio
6. ขณะที่คลื่นเดินทาง คลื่นจะมีสัญญาณอ่อนลงเนื่องจากกระบวนการที่เรียกว่า การสูญเสียลดทอน หรือ Propagation Loss ทุกอย่างที่คลื่นไปสัมผัส อย่างเช่น โมเลกุลของอากาศ ไอน้ำ และสายฝน จะทำให้คลื่นนั้นอ่อนลงจากกระบวนการทีเรียกว่า กระบวนการดูดซับ หรือ Absorption ยิ่งคลื่นเดินทางไปไกลเท่าใด การสูญเสียก็จะยิ่งมากขึ้นเท่านั้น โดยทั่วไปถ้ามีความถี่สูง การสูญเสียก็จะมาก ถ้าความถี่ต่ำ การสูญเสียก็จะน้อย ดังนั้นคลื่นวิทยุเอเอ็มจึงสามารถเดินทางได้ไกลกว่าคลื่นวิทยุเอฟเอ็มเนื่องจากถูกส่งสัญญาณที่มีความถี่ต่ำกว่า

2.2.3 สายอากาศ

การส่งสัญญาณและการรับสัญญาณ สายอากาศใช้สำหรับส่งและรับสัญญาณคลื่นวิทยุหรือคลื่น RF (Radio frequency) เมื่อสายอากาศถูกใช้สำหรับส่งสัญญาณมันจะเปลี่ยนสัญญาณไฟฟ้าให้เป็นคลื่น RF ซึ่งสัญญาณไฟฟ้านั้นถูกสร้างมาจากอุปกรณ์ส่งสัญญาณ (Transmitter) และสัญญาณไฟฟ้าจะเคลื่อนที่ไปยังสายอากาศและเมื่อพบกับความต้านทานของสายอากาศมันจะสร้างคลื่น RF ออกมาและถูกส่งออกไป

การรับสัญญาณ เมื่อสายอากาศถูกใช้ในการรับสัญญาณ การทำงานจะตรงกันข้ามกับการส่ง มันจะรับคลื่น RF และเปลี่ยนเป็นสัญญาณไฟฟ้า เนื่องจากสัญญาณที่ส่งมาอาจจะไม่แรงเพียงพอ สายอากาศบางประเภทจะมีตัวขยายสัญญาณเบื้องต้น หรือ Preamplifier ที่ทำหน้าที่เพิ่มความแรงของสัญญาณก่อนที่จะส่งต่อไปยังอุปกรณ์รับสัญญาณต่อไป

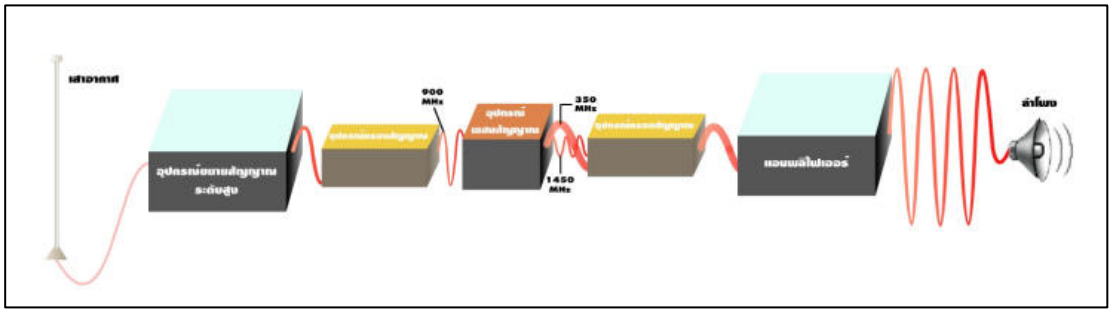


รูปที่ 2.18 สายอากาศแบบรอบทิศทางกับแบบมีทิศทาง

สายอากาศแบบรอบทิศทางกับแบบมีทิศทาง (Omnidirectional และ Directional) สายอากาศมีสองชนิดคือ สายอากาศแบบรอบทิศทางกับสายอากาศแบบมีทิศทาง สายอากาศแบบรอบทิศทางจะส่งสัญญาณในทุก ๆ ด้านไม่เจาะจง ขณะที่แบบมีทิศทางจะส่งแบบมีทิศทางแน่นอน สายอากาศแบบมีทิศทางนั้นมีวัตถุประสงค์ในการใช้ เช่น กรณีที่มีภูเขาหรือเนินเขาอยู่ด้านหลังสายอากาศ การส่งแบบมีทิศทางจะเหมาะสมกว่า เนื่องจากไม่ต้องสูญเสียพลังงานในการส่งสัญญาณไปด้านหลังเนื่องจากมีภูเขาบังสัญญาณอยู่ โดยจะทำการส่งสัญญาณในทิศทางเดียวและใช้พลังงานอย่างคุ้มค่า

ชนิดของสายอากาศ สายอากาศมีการออกแบบที่ซับซ้อน ส่วนการออกแบบก็มีหลายแบบ โดยขึ้นอยู่กับความยาวของคลื่น RF ความแรงของสัญญาณ วัตถุประสงค์ของตัวส่งสัญญาณหรือตัวรับสัญญาณ ราคา หรืออาจจะเป็นเฉพาะตัวรับหรือตัวส่งเท่านั้น และยังมีอีกหลายปัจจัย รูปที่เห็นอยู่แสดงแบบของสายอากาศ 2-3 แบบ สายอากาศแบบ “ยากิ” (Yagi) จะใช้สำหรับการรับสัญญาณโทรทัศน์และวิทยุสมัครเล่น สายอากาศแบบ “วิพส์” (Whips) ใช้สำหรับโทรศัพท์และรับคลื่นวิทยุ และอื่นๆ ซึ่งมีทั้งความถี่ระดับกลางหรือ MF (Medium Frequency) และช่วงกว้างอื่น ๆ เป็นต้น

ขนาดของสายอากาศ ขนาดของสายอากาศนั้นจะขึ้นกับความถี่ของสัญญาณที่จะรับ ถ้าความถี่มีความมากเท่าไรความยาวคลื่นก็จะน้อย ดังนั้นคลื่นที่มีความถี่สูงจะมีความยาวคลื่นต่ำ และถ้ามีความถี่ต่ำจะมีความยาวคลื่นสูง ขนาดของสายอากาศจะขึ้นกับความยาวของความยาวคลื่นเป็นหลัก ดังนั้นหากความยางคลื่นมากหรือมีความถี่ต่ำ สายอากาศที่ใช้ก็จะต้องมีขนาดใหญ่ด้วย นั่นคือสาเหตุที่ว่าทำไมโทรศัพท์มือถือจึงมีสายอากาศขนาดเล็ก ก็เนื่องจากคลื่นที่ใช้นั้นมีความถี่สูงหรือความยาวคลื่นต่ำ สายอากาศจึงมีขนาดเล็กมาก แต่ในการใช้งานจริงนั้นขนาดของสายอากาศจะมีขนาดไม่เท่ากับความยาวคลื่นอย่างพอดี แต่จะเป็นอัตราส่วนแทน เช่น 1/2 หรือ 1/4 เป็นต้น



รูปที่ 2.19 อุปกรณ์รับสัณญาณ

2.2.4 อุปกรณ์รับส่งสัญญาณ

1. อุปกรณ์รับสัญญาณมีการทำงานเหมือนกับอุปกรณ์ส่งสัญญาณมาก แต่เป็นกระบวนการที่ตรงข้ามกัน เริ่มจากเสาอากาศรับสัญญาณ และจะเปลี่ยนคลื่น RF ไปเป็นสัญญาณไฟฟ้า

2 สัญญาณไฟฟ้าอาจจะอ่อน และจำเป็นต้องทำให้เข้มขึ้นโดยการขยายสัญญาณ ดังนั้นสัญญาณไฟฟ้าจะเดินทางไปที่อุปกรณ์ขยายสัญญาณ (Amplifier) ตัวขยายสัญญาณในอุปกรณ์รับ

สัญญาณเรียกว่า อุปกรณ์ขยายสัญญาณรบกวนต่ำ หรือ Low-noise amplifier เนื่องจากอุปกรณ์นี้จะรับสัญญาณที่มีสัญญาณรบกวนต่ำ (Low noise) และทำการขยายสัญญา

3. สัญญาณไฟฟ้าที่รับการขยายแล้วจะเดินทางไปที่อุปกรณ์กรองสัญญาณ ซึ่งจะทำการกรอบสัญญาณที่เกินมาและสัญญาณ RF ด้วย เสาอากาศจะรับสัญญาณ RF เข้ามาซึ่งสัญญาณเหล่านี้อาจส่งมาจากโทรศัพท์มือถือดาวเทียมสื่อสาร หรือดวงอาทิตย์ ซึ่งสัญญาณเหล่านี้มีความถี่ที่แตกต่างกัน อุปกรณ์รับสัญญาณนั้นจะแปลงสัญญาณที่มีความถี่ที่ได้มานั่นคือตัวกรองจะทำการกรองสัญญาณที่ไม่จำเป็นออกไป ในตัวอย่างของเรา ตัวกรองจะกำจัดสัญญาณทุกตัวออกไปยกเว้นที่ความถี่ 900 MHz เท่านั้น ซึ่งเป็นความถี่ที่คลื่นนั้นถูกส่งมา

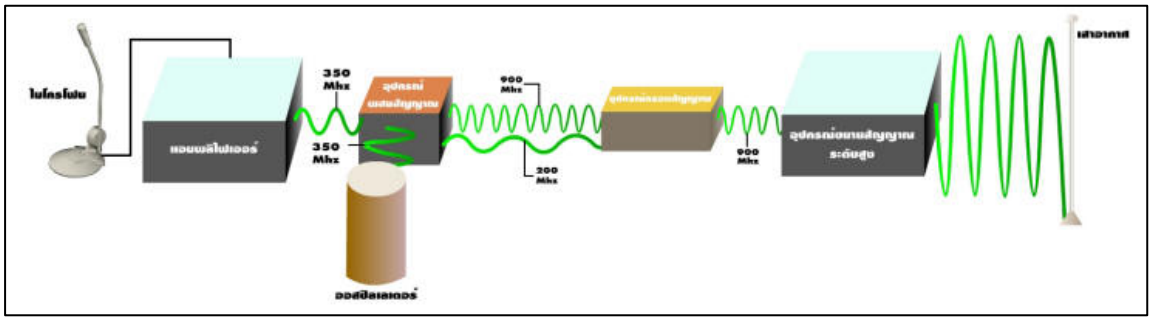
4. อุปกรณ์รับสัญญาณในขณะนี้จำเป็นต้องแยกข้อมูลในสัญญาณออกจากคลื่นตัวนำ ดังนั้นสัญญาณจะถูกส่งไปยังอุปกรณ์ผสมสัญญาณหรือ Mixer ซึ่งทำหน้าที่ในการแยกคลื่นออกมา

5. เพื่อที่จะแยกข้อมูลออกมา คลื่นที่มีความถี่เฉพาะนั้นจำเป็นต้องถูกสร้างมาจากออสซิสเลเตอร์ ในตัวอย่าง ข้อมูลที่มีความถี่ 350 MHz และสัญญาณทั้งหมดที่ส่งมานั้นมีความถี่อยู่ที่ 900 MHz ดังนั้น ออสซิลเลเตอร์จำเป็นต้องสร้างสัญญาณความถี่ที่ 550 MHz

6. สัญญาณที่ 2 ความถี่เดินทางออกจากอุปกรณ์ผสมสัญญาณที่ความถี่ 1,450 MHz (มาจาก 900+550) และความถี่ที่ 350 MHz (มาจาก 900-550) อุปกรณ์รับสัญญาณไม่ต้องการที่ความถี่ 1,450 MHz ดังนั้นสัญญาณจะถูกส่งไปยังตัวกรองตัวที่ 2 ที่จะกรองสัญญาณความถี่ 1,450 MHz ออกไป

7. ขณะนี้กระบวนการดิโมดูเลชัน (Demodulation) จะถูกนำมาใช้ โมดูเลเตอร์เปลี่ยนข้อมูลข่าวสารในคลื่นไปเป็นข้อมูลเริ่มต้นที่ส่งมา อย่างเช่น การออกอากาศเสียง โมดูเลเตอร์ทำงานหลาย ๆ อย่างขึ้นกับข้อมูลที่กำลังส่งมาส่วนมากจะใช้ตัวประมวลสัญญาณแบบดิจิตอลในการเปลี่ยนกลับไป

8. หลังจากผ่านกระบวนการของอุปกรณ์กรองสัญญาณและอุปกรณ์ผสมสัญญาณแล้ว จะทำให้สัญญาณที่ได้นั้นอ่อนลง จึงจำเป็นต้องทำให้สัญญาณแรงขึ้นโดยการส่งไปยังแอมพลิไฟเออร์ตัวที่สอง ตอนนี้สัญญาณก็สามารถนำมาใช้ได้แล้ว เช่น รับฟังเสียงจากลำโพง หรือโทรศัพท์มือถือเป็นต้น



รูปที่ 2.20 อุปกรณ์ส่งสัญญาณ

1. อุปกรณ์ส่งสัญญาณถูกออกแบบมาเพื่อส่งสัญญาณที่มีความถี่แน่นอน ยกตัวอย่างเช่น เราพูดด้วยสัญญาณเสียงที่มีความถี่ 900 MHz เริ่มต้นจากข้อมูลที่ต้องการส่งจะถูกสร้างขึ้นก่อน เช่น คนพูดผ่านเข้าไปในไมโครโฟนหรือโทรศัพท์มือถือ อุปกรณ์เหล่านี้จะทำการสร้างความถี่เฉพาะ ในตัวอย่างนี้จะสร้างความถี่ที่ 350 MHz สัญญาณที่สร้างขึ้นนี้ถูกสร้างโดยการกระตุ้นของกระแสไฟฟ้าไม่ใช่คลื่นวิทยุ

2. สัญญาณจะต้องถูกขยายโดยแอมพลิไฟเออร์ก่อนที่จะส่งต่อไปยังส่วนที่เหลือของตัวขยายสัญญาณ มิฉะนั้นจะไม่สามารถส่งสัญญาณนั้นออกไปได้

3. ในตัวอย่างนี้ สัญญาณคือเสียงร้องเพลงของคนจะต้องถูกใส่ไปในคลื่นตัวนำ (พาหะ) เพื่อที่จะถูกส่งต่อไป ความถี่ของคลื่นที่ได้จากการรวมกันคือ 900 MHz ดังนั้น คลื่นตัวนำจำเป็นต้องถูกสร้างขึ้น โดยออสซิลเลเตอร์ (Oscillator) จะสร้างคลื่นตัวนำ ซึ่งจะสร้างคลื่นที่สมบูรณ์ที่สุดเท่าที่จะทำได้ ในตัวอย่างสัญญาณจากตัวขยายสัญญาณคือ 350 MHz แต่เนื่องจากความถี่คลื่นที่จะส่งโดยอุปกรณ์ส่งสัญญาณนั้นต้องการอยู่ที่ 900 MHz ดังนั้นออสซิลเลเตอร์จะสร้างคลื่นที่สมบูรณ์และมีความถี่อยู่ที่ 550 MHz

4. ทั้งคลื่นที่ได้จากออสซอลเลเตอร์ที่มีความถี่ 550 MHz และคลื่นที่ได้จากตัวขยายสัญญาณที่มีความถี่ 350 MHz จะถูกส่งไปที่ตัวผสมสัญญาณหรือ Mixer เพื่อรวมคลื่นทั้งสองเข้าด้วยกัน คลื่นที่ผสมแล้วจะถูกส่งออกมา โดยจะมีอยู่สองส่วนคือส่วนที่ได้จากการรวมกันได้ 900 MHz (350+550) และส่วนที่ได้จากการลบกันคือ 200 MHz (550-350)

5. ก่อนที่คลื่นจะถูกส่งไปได้ คลื่นนั้นจะต้องไม่มีการรบกวนของคลื่นความถี่ที่ไม่ต้องการอยู่ด้วย อุปกรณ์ส่งสัญญาณถูกออกแบบมาให้ส่งคลื่นความถี่ 900 MHz ดังนั้นจะต้องมีวิธีการที่จะกำจัดคลื่นความถี่ 200 MHz ออกไป อุปกรณ์กรองสัญญาณ (Filler) ซึ่งมีอยู่ 4 ชนิด คือ Low pass ซึ่งจะให้คลื่นที่มีความถี่ต่ำกว่ากำหนดเท่านั้นผ่านได้ และกำจัดความถี่อื่น ๆ ออกไป ตัวกรองชนิด

High pass จะให้คลื่นที่มีความถี่สูงกว่าผ่านไปได้ นอกจากนั้นจำกำจัดทิ้งไป ส่วนตัวกรองชนิด Band pass จะให้ความถี่ที่อยู่ในช่วงที่กำหนดผ่านได้เท่านั้น ตัวกรองชนิด Band reject จะให้ความถี่ที่อยู่ระหว่างนอกเหนือที่กำหนดผ่านไปได้นอกจากนั้นจะกำจัดทิ้งไป

6. สัญญาณนั้นยังอ่อนอยู่มากที่จะถูกส่งไปยังที่ไกล ๆ ได้ ดังนั้นสัญญาณจะเดินทางผ่านเข้าไปยังตัวขยายสัญญาณอีกตัวหนึ่งซึงมีกำลังขยายที่ดีกว่าตัวแรก ตัวขยายสัญญาณในอุปกรณ์ส่งสัญญาณเรียกว่า ตัวขยายสัญญาณกำลังสูง หรือ High power amplifiers (HAP) ซึ่งถูกออกแบบมาเพื่อเพิ่ม (Boost) ความเข้มของสัญญาณเท่าที่จะทำได้ การเพิ่มความเข้มนั้นจะขึ้นกับอุปกรณ์และระยะทางที่ต้องการส่งสัญญาณ ตัวอย่างเช่น สถานีฐานของโทรศัพท์มือถือจะมีการเพิ่มความเข้มของสัญญาณถึง 50 เท่า เพื่อส่งไปยังโทรศัพท์มือถือที่ใช้อยู่ เป็นต้น

2.2.5 ข้อดี – ข้อเสียของอุปกรณ์รับส่งสัญญาณ

ข้อดี

1. ช่วยลดปัญหาในการติดตั้งระบบเครือข่าย
2. ช่วยลดปัญหาในการวางสายระบบเครือข่าย
3. ไม่ต้องใช้สาย cable
4. ช่วยให้เกิดความเป็นระเบียบ เรียบร้อย

ข้อเสีย

1. มีอัตราการลดทอนสัญญาณสูง นั่นหมายความว่า “ ส่งสัญญาณได้ระยะสั้น ”
2. มีสัญญาณรบกวนสูง
3. ต้องแชร์กันใช้ช่องสัญญาณคลื่นความถี่เดียวกัน
4. ยังมี หลายมาตรฐาน ตามผู้ผลิต แต่ละราย ทำให้ มีปัญหา ในการ ใช้งาน ร่วมกัน
5. ราคาแพงกว่าระบบเครือข่ายแบบมีสาย
6. มีความเร็วไม่สูงมากนัก

2.3 รีเลย์ (Relay)

รีเลย์ คือ อุปกรณ์อิเล็กทรอนิกส์ ที่ทำหน้าที่ ตัด-ต่อวงจร คล้ายกับสวิตช์ โดยใช้หลักการหน้าสัมผัส และการที่จะให้มันทำงานก็ต้องจ่ายไฟให้มันตามที่กำหนด เพราะเมื่อจ่ายไฟให้กับตัวรีเลย์ มันจะทำให้หน้าสัมผัสติดกัน กลายเป็นวงจรปิด และตรงข้ามทันทีที่ไม่ได้จ่ายไฟให้มัน มันก็จะกลายเป็นวงจรเปิด ไฟที่เราใช้ป้อนให้กับตัวรีเลย์ก็จะเป็นไฟที่มาจาก เพาเวอร์ฯ ของเครื่องเราดังนั้นทันทีที่เปิดเครื่อง ก็จะทำให้รีเลย์ทำงาน

รีเลย์ (Relay) เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา
2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุมบางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

ชนิดของรีเลย์

การแบ่งชนิดของรีเลย์สามารถแบ่งได้ 3 แบบ คือ

2.3.1 ชนิดของรีเลย์แบ่งตามลักษณะของคอยล์ หรือ แบ่งตามลักษณะการใช้งาน (Application) ได้แก่รีเลย์ดังต่อไปนี้

1. รีเลย์กระแส (Current relay) คือ รีเลย์ที่ทำงานโดยใช้กระแสมีทั้งชนิดกระแสขาด (Under- voltage) และกระแสเกิน (Over current)
2. รีเลย์แรงดัน (Voltage relay) คือ รีเลย์ ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และ แรงดันเกิน (Over voltage)
3. รีเลย์ช่วย (Auxiliary relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่น จึงจะทำงานได้
4. รีเลย์กำลัง (Power relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน
5. รีเลย์เวลา (Time relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งมีอยู่ด้วยกัน 4 แบบ คือ

- รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse time over current relay) คือ รีเลย์ ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส
- รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous over current relay) คือรีเลย์ที่ทำงานทันทีทันใดเมื่อมีกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้
- รีเลย์แบบดิฟฟินิตไทม์เล็ก (Definite time lag relay) คือ รีเลย์ ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากน้อยของกระแสหรือค่าไฟฟ้าอื่นๆ ที่ทำให้เกิดงานขึ้น
- รีเลย์แบบอินเวอสดิฟฟินิตมินิมั่มไทม์เล็ก (Inverse definite time lag relay) คือ รีเลย์ ที่ทำงานโดยรวมเอาคุณสมบัติของเวลาผกผันกับกระแส (Inverse time) และแบบดิฟฟินิตไทม์แล็ก (Definite time lag relay) เข้าด้วยกัน

6. รีเลย์กระแสต่าง (Differential relay) คือ รีเลย์ที่ทำงานโดยอาศัยผลต่างของกระแส
7. รีเลย์มีทิศ (Directional relay) คือรีเลย์ที่ทำงานเมื่อมีกระแสไหลผิดทิศทาง มีแบบรีเลย์กำลังมีทิศ (Directional power relay) และรีเลย์กระแสมีทิศ (Directional current relay)
8. รีเลย์ระยะทาง (Distance relay) คือ รีเลย์ระยะทางมีแบบต่างๆ ดังนี้
   - รีแอกแตนซ์รีเลย์ (Reactance relay)
   - อิมพีแดนซ์รีเลย์ (Impedance relay)
   - โมห์รีเลย์ (Mho relay)
   - โอห์มรีเลย์ (Ohm relay)
   - โพลาไรซ์โมห์รีเลย์ (Polaized mho relay)
   - ออฟเซทโมห์รีเลย์ (Off set mho relay)
9. รีเลย์อุณหภูมิ (Temperature relay) คือ รีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว้
10. รีเลย์ความถี่ (Frequency relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้

2.3.2 ชนิดของรีเลย์แบ่งตามลักษณะทิศในการทำงาน

แบ่งได้สองแบบคือ แบบไม่มีทิศทางในการทำงานและแบบมีทิศทางในการทำงาน

1. รีเลย์แบบไม่มีทิศทางในการทำงาน รีเลย์แบบนี้สามารถทำงานได้เลย โดยไม่คำนึงถึงทิศทางของกระแสหรือแรงดัน จะเห็นว่ารีเลย์ไม่มีทิศมีทั้งแบบที่ใช้ค่ากระตุ้นเพียงค่าเดียวอาจจะเป็นกระแสหรือแรงดันก็ได้ เช่น แบบขั้ววงแหวน (Shaded pole) ส่วนแบบวัตต์ชั่วโมง ลูกถ้วยคู่เหนี่ยวนำและแบบลูกถ้วยเดี่ยวเหนี่ยวนำ จะใช้ค่ากระตุ้นทั้งสองชุด แต่ถ้ามีกระแสเพียงค่าเดียว ก็จะสามารถดัดแปลงให้รีเลย์ทำงานได้ โดยต่อขดลวดทั้งสองขนานหรืออนุกรมกันก็ได้และในการออกแบบรีเลย์เราให้ขดลวดทั้งสองมีค่า R / X ต่างกันโดยการต่อ C หรือ R เข้าไปอนุกรมกับขดลวดขดใดขดหนึ่งในสองขดนั้น ทั้งนี้เพื่อให้เส้นแรงแม่เหล็กที่เกิดจากขดลวดทั้งสองนั้นมีเฟสต่างกันจึงทำให้เกิดแรงหมุนได้

รีเลย์แบบไม่มีทิศถ้ารีเลย์ทำงานด้วยกระแสจะได้แรงบิด

$$T = K_1 I^2 - K_2 \qquad (1)$$

เมื่อ I = ค่ากระแสในขดลวดทั้งสอง

สำหรับมุมของกระแสทั้งสองมีค่าคงที่จากการออกแบบแล้วจึงรวมเข้าไปกับ K_1 ได้ถ้ารีเลย์นี้ทำงานด้วยแรงดันจะได้ว่า

$$\text{แรงบิด } T = K_1 V^2 - K_2 \qquad (2)$$

เมื่อ V = แรงดันที่ใส่เข้าไปในขดลวดทั้งสอง

2. รีเลย์แบบมีทิศในการทำงาน รีเลย์เหนี่ยวนำชนิดนี้จะทำงานเมื่อฟอลต์ เกิดขึ้นในทิศทางที่กำหนดไว้เท่านั้น รีเลย์แบบนี้มีค่ากระตุ้นอยู่สองค่าซึ่งมาจากคนละแหล่งกันและมุมระหว่างค่ากระตุ้นทั้งสองนั้นเปลี่ยนแปลงไปตามแต่ละชนิด รีเลย์แบบวัตต์ชั่วโมงแบบลูกถ้วยเหนี่ยวนำก็สามารถทำเป็นรีเลย์แบบมีทิศได้ รีเลย์แบบมีทิศนี้แบ่งตามค่ากระตุ้นได้ 4 แบบคือ

1) ค่ากระตุ้นแบบ กระแส-กระแส กระแสที่ไหลเข้าไปในขดลวดทั้งสองจะได้จากแหล่งละตัว

2) ค่ากระตุ้นเป็นแรงดัน–แรงดัน

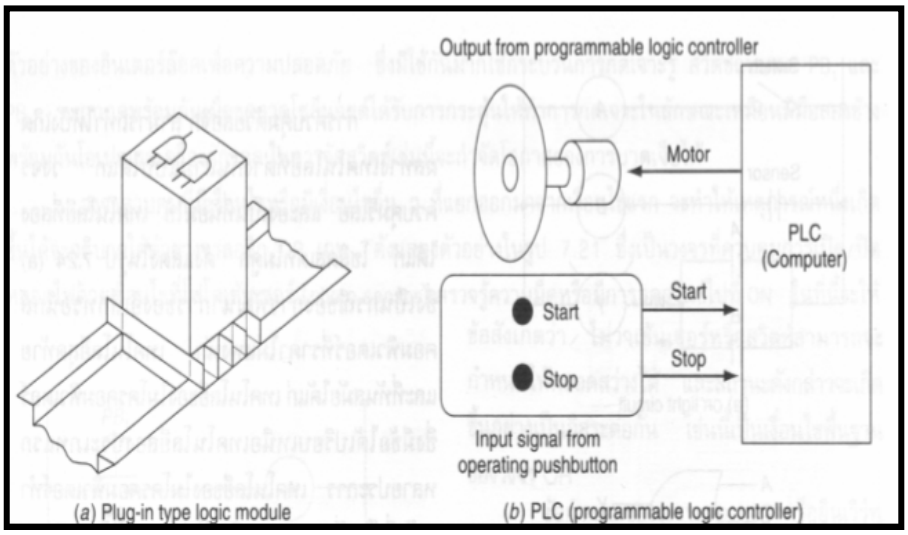
$$T = K_1 V_1 V_2 \cos(\theta - \tau) - K_2 \qquad (3)$$

3) ค่ากระตุ้นที่เป็นแรงดันและกระแส-กระแสที่มาจากแหล่งเดียวกัน รีเลย์แบบนี้มักจะใช้ทำเป็นรีเลย์แบบมีทิศในการทำงานแบบกระแสเกิน (Directional over current) การทำงานของรีเลย์ชนิดนี้จะเป็นดังนี้ คือเมื่อมีกระแสเกินหรือฟอลต์เข้ามา รีเลย์ตัวล่างจะทำการต่อวงจรของคอยล์ให้รีเลย์ตัวบนเมื่อรีเลย์ตัวบนเริ่มทำงานก็จะสั่งตัดวงจรของเซอร์กิตเบรกเกอร์ออก ซึ่งวงจรของเซอร์กิตเบรกเกอร์ จะต่อจากวงจรทริฟของรีเลย์ตัวบนดังนั้น รีเลย์ชนิดนี้จึงเหมาะสำหรับใช้ป้องกันฟอลต์ชนิดเฟสต่อเฟส และเฟสลงดิน

4) ค่ากระตุ้นที่เป็นแรงดันและกระแส

$$T = K_1 V I \cos(\theta - \tau) - K_2 \qquad (4)$$

จากการกระตุ้นของกระแสและแรงดันนี้ จึงนิยมนำมาทำเป็นรีเลย์กำลังแบบมีทิศทาง (Power directional relay) เราต่อวงจรแรงดันให้เป็นตัวสร้างขั้วที่แน่นอน (Polarized) และใช้กระแสเป็นตัวกระตุ้น ซึ่งเราจะออกแบบให้แรงบิด (Torque) ของรีเลย์มีค่ามากที่สุดเมื่อตัวประกอบกำลัง (Power factor) ของโหลดมีค่าเป็นหนึ่ง ดังนั้นรีเลย์จะทำงานเมื่อมีกระแสไฟฟ้าไหลผิดทิศทางและรีเลย์ต้านการทำงาน เมื่อมีไฟฟ้าไหลถูกทิศทาง รีเลย์เหล่านี้ส่วนมากจะเป็นแบบหนึ่งเฟสใช้ป้องกันกำลังไฟฟ้ากลับทิศทางมากกว่าที่จะนำมาป้องกันการลัดวงจร



รูปที่ 2.21 การควบคุมด้วยโซลิดสเตทรีเลย์

บทที่ 3

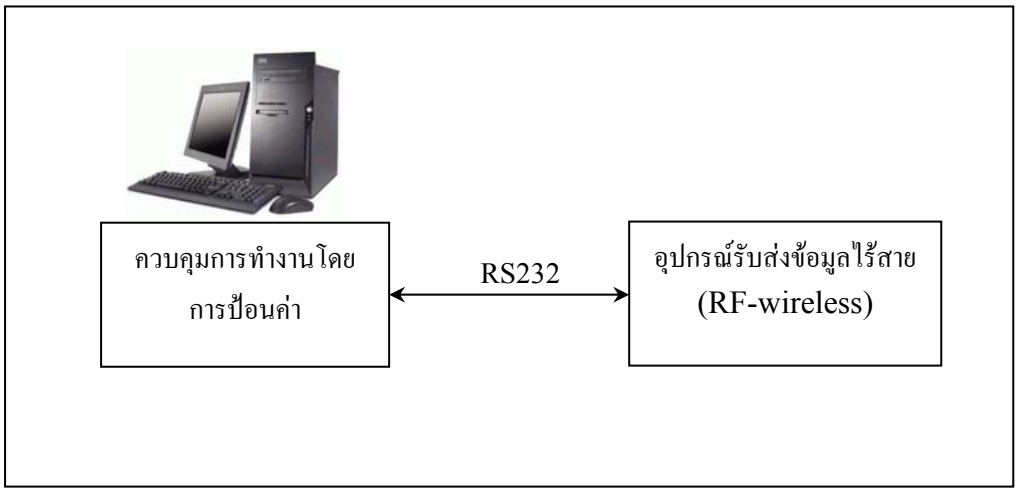
การออกแบบระบบเปิด-ปิดอุปกรณ์ไฟฟ้า

ในการออกแบบระบบการทำงานของตัวอุปกรณ์รับส่งสัญญาณควบคุมการทำงานปิด-เปิดอุปกรณ์เครื่องใช้ไฟฟ้าผ่านเครือข่ายไร้สายนั้น จะมีส่วนประกอบต่างๆ ซึ่งในแต่ละส่วนนั้นจะทำหน้าที่ต่างๆ โดยจะอธิบายแต่ละส่วนได้ดังนี้

3.1 แผนภาพของระบบ

3.1.1 ภาคส่งสัญญาณ

ในทางภาคส่งสัญญาณจะมีการออกแบบขั้นต้น โดยทำการรับค่าข้อมูลจากการป้อนคำสั่งเข้าไปในตัวโปรแกรม และส่งสัญญาณข้อมูลที่ได้นั้นผ่านระบบเครือข่ายไร้สาย โดยสามารถอธิบายได้ดังรูปที่ 3.1



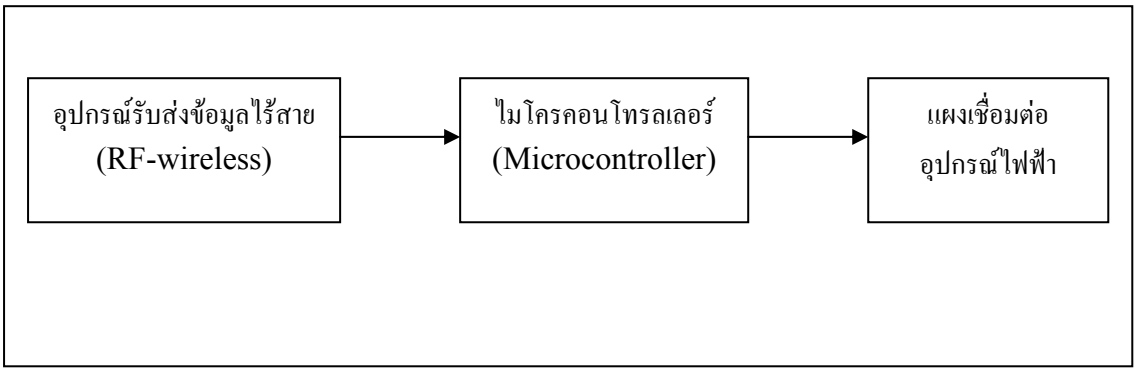
รูปที่ 3.1 แผนภาพภาคส่งสัญญาณ

จากรูปที่ 3.1 เป็นแผนภาพบล๊อคไดอะแกรมภาคส่งสัญญาณ มีขั้นตอนการทำงานดังนี้

1. ในส่วนของตัวควบคุมการทำงานโดยการป้อนค่า เมื่อ Login เข้าสู่โปรแกรม โปรแกรมจะเปิดขึ้นมา แล้วเลือกคำสั่ง หลังจากนั้นคำสั่งจะถูกส่งออกทาง RS232
2. เมื่อ RF-wireless รับคำสั่งจาก RS232 RF-wireless จะส่งคำสั่งไปให้ RF-wireless ของภาครับ

3.1.2 ภาครับสัญญาณ

ในทางภาครับสัญญาณมีการออกแบบขั้นต้นโดยการแบ่งเป็นส่วนๆเพื่อทำการรับสัญญาณข้อมูลไร้สายแล้วข้อมูลที่รับได้จะทำการประมวลผลสัญญาณข้อมูลที่ไมโครคอนโทรลเลอร์ 8051 เพื่อเปิด-ปิดอุปกรณ์ไฟฟ้า และส่งข้อมูลกลับไปยังภาคส่งสัญญาณด้วย โดยสามารถอธิบายได้จากรูปที่ 3.2



รูปที่ 3.2 แผนภาพภาครับสัญญาณ

จากรูปที่ 3.2 เป็นแผนภาพภาครับสัญญาณ โดยมีขั้นตอนการทำงานดังนี้

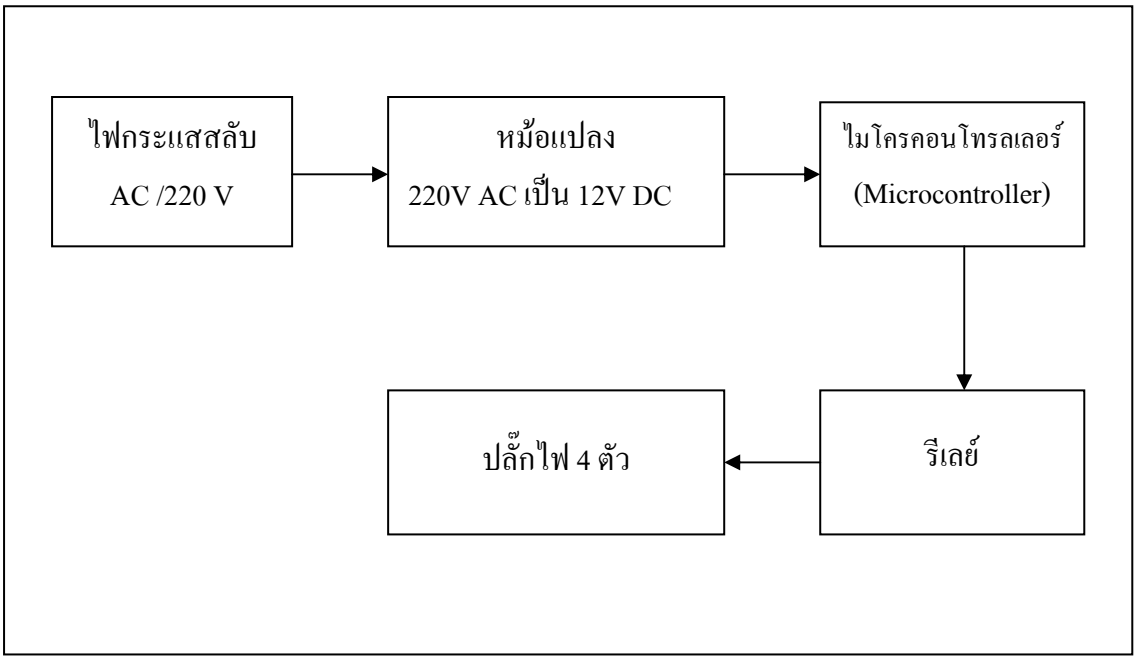
1. ในส่วนของชุดรับส่งข้อมูลไร้สาย จะรับสัญญาณจากภาคส่งสัญญาณของตัวอุปกรณ์เครือข่ายไร้สายแล้วส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ผ่าน RS232
2. ส่วนของไมโครคอนโทรลเลอร์ จะรอรับสัญญาณจากอุปกรณ์เครือข่ายไร้สายเพื่อที่จะส่งสัญญาณไปควบคุมรีเลย์
3. ในส่วนของแผงเชื่อมต่ออุปกรณ์ไฟฟ้าจะรอรับสัญญาณจาก ไมโครคอนโทรลเลอร์ โดยเมื่อได้รับสัญญาณมา ไมโครคอนโทรลเลอร์จะจ่ายไฟให้กับแผงเชื่อมต่ออุปกรณ์ไฟฟ้า

3.1.3 ชุดควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้า

ในส่วนของชุดควบคุมนี้จะทำงานในส่วนของภาครับสัญญาณ ซึ่งอธิบายแผนภาพการทำงานได้สองส่วนคือ ชุดอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้า และชุดอุปกรณ์รับ-ส่งข้อมูลผ่านระบบการสื่อสารไร้สาย

ชุดอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้า

ในส่วนควบคุมชุดอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้าจะมีการออกแบบขั้นต้น เพื่อทำการเชื่อมต่อระหว่างแผงอุปกรณ์ไฟฟ้าและไมโครคอนโทรลเลอร์ ซึ่งอธิบายได้ดังรูปที่ 3.3



รูปที่ 3.3 แผนภาพชุดอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้า

จากรูปที่ 3.3 เป็นแผนภาพชุดอุปกรณ์ปลั๊กเสียบเครื่องใช้ไฟฟ้า โดยมีขั้นตอนการทำงานดังนี้

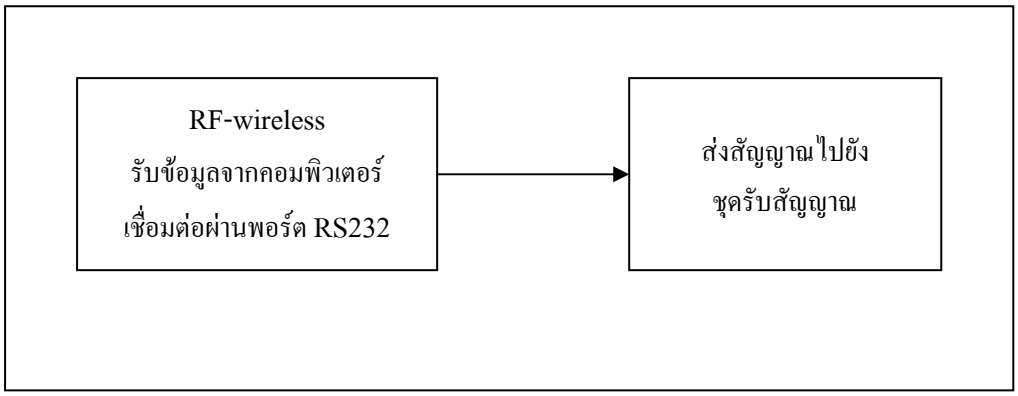
1. รับไฟกระแสสลับ 220V AC เพื่อจ่ายให้หม้อแปลง
2. หม้อแปลงทำหน้าที่ แปลงไฟ 220VAC เป็น 12V DC เพื่อจ่ายให้ไมโครคอนโทรลเลอร์

3. รีเลย์รอรับสัญญาณจากไมโครคอนโทรลเลอร์เมื่อมีสัญญาณจาก
   ไมโครคอนโทรลเลอร์ รีเลย์จะเชื่อมต่อหน้าสัมผัส จ่ายไฟ 220V ให้กับปลั๊กไฟ 4 ตัว
4. ปลั๊กไฟก็จะทำหน้าที่ ปิด-เปิดอุปกรณ์ไฟฟ้า

ชุดอุปกรณ์รับส่งข้อมูลผ่านระบบการสื่อสารไร้สาย

ในส่วนของชุดอุปกรณ์รับส่งข้อมูลแบบไร้สายนั้นจะเป็นอุปกรณ์ที่ทำหน้าที่ติดต่อรับส่งข้อมูลกันระหว่างด้านส่งสัญญาณและด้านรับสัญญาณ โดยจะทำการส่งข้อมูลกันเป็นระยะทางไกล โดยมีการออกแบบระบบ และขั้นตอนการทำงานดังรูปที่ 3.4

ภาคส่งสัญญาณ

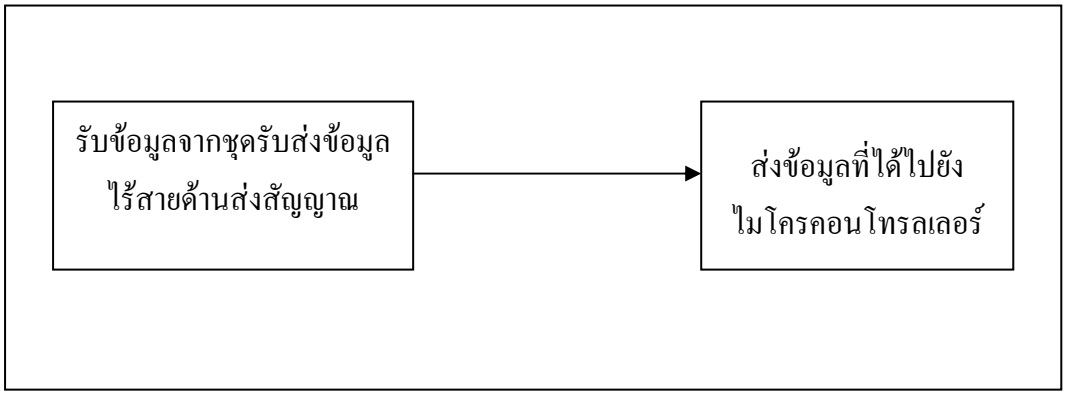


รูปที่ 3.4 แผนภาพชุดส่งข้อมูลผ่านระบบการสื่อสารไร้สายด้านส่งสัญญาณ

จากรูปที่ 3.4 เป็นแผนภาพชุดส่งสัญญาณผ่านรับการสื่อสารไร้สายด้านส่งสัญญาณ โดยมีขั้นตอนดังนี้

1. ชุดอุปกรณ์รับส่งข้อมูล (RF-wireless) ด้านส่งสัญญาณจะรับข้อมูลจากคอมพิวเตอร์ โดยการป้อนค่ารับคำสั่งจากโปรแกรม Visual Basic 6.0 เชื่อมต่อผ่านพอร์ต RS232
2. ส่งสัญญาณที่ได้ต่อไปยังภาครับสัญญาณที่ตัวชุดอุปกรณ์รับส่งข้อมูลด้านรับสัญญาณ

ภาครับสัญญาณ



รูปที่ 3.5 แผนภาพชุดส่งข้อมูลผ่านระบบการสื่อสารไร้สายด้านรับสัญญาณ

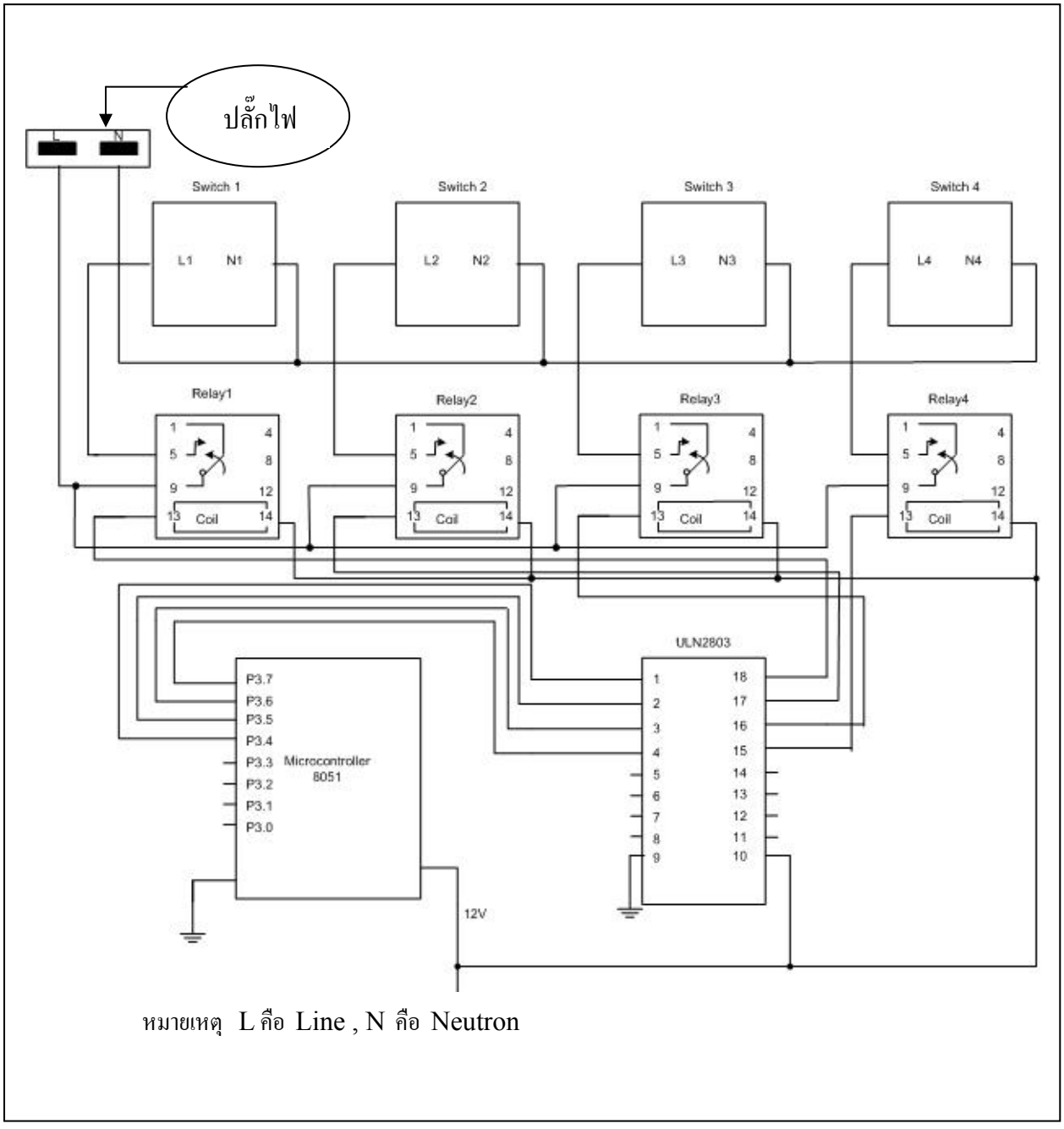
จากรูปที่ 3.5 เป็นแผนภาพชุดส่งสัญญาณผ่านรับการสื่อสารไร้สายด้านส่งสัญญาณ โดยมีขั้นตอนดังนี้

1. ชุดอุปกรณ์การสื่อสารไร้สายจะรอรับสัญญาณจากชุดรับ-ส่งข้อมูลไร้สายด้านส่งสัญญาณ
2. ส่งข้อมูลที่ได้รับมาไปยังไมโครคอนโทรลเลอร์โดยผ่านพอร์ต RS232 โดยจะส่งข้อมูลมายังไมโครคอนโทรลเลอร์ได้จะต้องทำการตรวจสอบช่องสัญญาณว่าตรงกับด้านส่งสัญญาณหรือไม่ ถ้าไม่ตรงกันก็จะทำการลบข้อมูลนั้นทิ้งไป หากช่องสัญญาณตรงกันก็ทำการส่งข้อมูล

3.2 การออกแบบทาง Hardware

3.2.1 แผงเชื่อมต่ออุปกรณ์ไฟฟ้า

ในการออกแบบอุปกรณ์นั้น จะออกแบบชุดควบคุมด้านรับสัญญาณ โดยนำอุปกรณ์ต่างๆ มาประกอบกันขึ้น และแสดงผลที่แผงอุปกรณ์ไฟฟ้าปลั๊กไฟ 4 ตัว โดยอธิบายเป็นส่วน ๆ มีขั้นตอนการทำงาน แผงผังวงจรดังรูปที่ 3.6

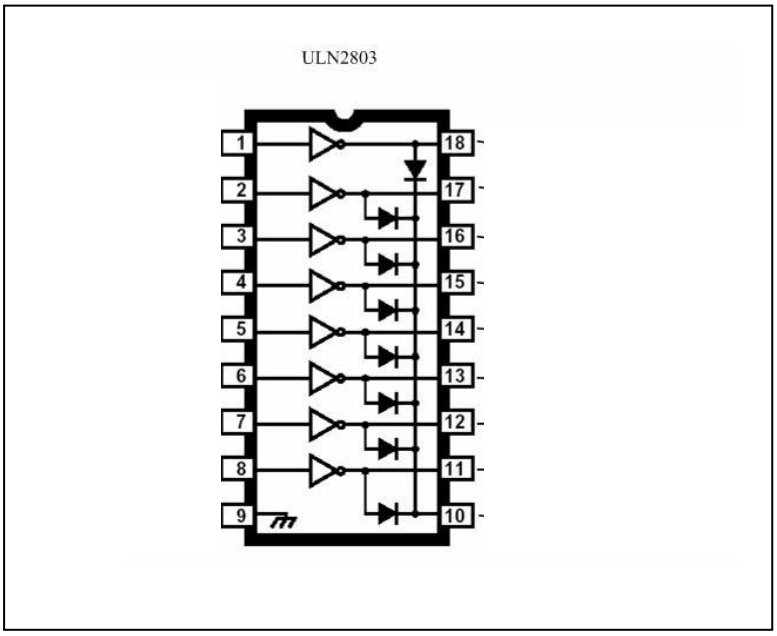


รูปที่ 3.6 แผงผังการทำงานของชุดแผงเชื่อมต่ออุปกรณ์ไฟฟ้า

จากรูปที่ 3.6 แสดงแผงผังการทำงานของชุดแผงเชื่อมต่ออุปกรณ์ไฟฟ้า ซึ่งสามารถอธิบายขั้นตอนการทำงานในแต่ละส่วนได้ดังนี้

<u>ส่วนที่ 1</u> ไมโครคอนโทรลเลอร์ ซึ่งออกแบบให้ P3.7, P3.6, P3.5 และ P3.4 เป็นพอร์ตเอาท์พุตของไมโครคอนโทรลเลอร์ เพื่อควบคุมการเปิด-ปิดสวิตช์ไฟ 4 ตัว

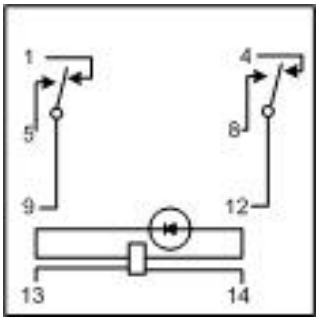
<u>ส่วนที่ 2</u> ไอซี ULN 2803 ดังรูปที่ 3.7



รูปที่ 3.7 ไอซี ULN2803

จากรูปที่ 3.7 อธิบายหลักการทำงานได้ดังนี้

เมื่ออินพุตเป็น 1 ลอจิกอินเวิร์ตภายในไอซีจะทำหน้าที่กลับบิตจาก 1 เป็น 0 และแปลงไฟ 5 โวลท์ จากลอจิก 1 เป็นกราวด์โดยสมบูรณ์คือ ลอจิกศูนย์ เพื่อเป็นขั้วลบ จ่ายให้ขดลวดเพื่อเหนี่ยวนำให้รีเลย์ทำงานครบวงจร หน้าสัมผัสของรีเลย์ก็จะทำงาน



รูปที่ 3.8 วงจรของรีเลย์

<u>ส่วนที่ 3</u> จากรูปที่ 3.8 วงจรของรีเลย์ ในส่วนของรีเลย์นี้จะแบ่งออกเป็น

1. ส่วนประกอบของรีเลย์ได้แก่
   -ขดลวดเหนี่ยวนำคือ ขาที่ 13 และ 14
   -หน้าสัมผัสจะมี 2 สถานะคือ สถานะปิด Normal close ขาที่ 9 และ 1 เชื่อมต่อกัน
   สถานะเปิด Normal open ขาที่ 9 และ 5 ไม่เชื่อมต่อกัน
2. หลักการทำงานของรีเลย์ เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดเหนี่ยวนำ คือขาที่ 13 และ 14 ของรีเลย์ จะทำให้ขดลวดเหนี่ยวนำ ดูดหน้าสัมผัสของรีเลย์ ไปด้านตรงข้ามจากสถานะปกติแต่เมื่อไม่มีกระแสไฟฟ้าไหลผ่านขดลวดเหนี่ยวนำ หน้าสัมผัสของรีเลย์จะกลับมาเข้าสู่สถานะปกติ
3. การออกแบบเลือกใช้รีเลย์
   - ขดลวดเหนี่ยวนำเลือกใช้รีเลย์ที่มีขดลวดเหนี่ยวนำทนแรงดันได้สูงสุด 12 V DC เนื่องจากไมโครคอนโทรลเลอร์ใช้แหล่งจ่ายไฟ 12 V DC
   - หน้าสัมผัส (contact) ต้องเลือกใช้รีเลย์ที่มีหน้าสัมผัสทนแรงดันได้สูงสุด 220 V AC ใช้หน้าสัมผัสที่มีสถานะปกติเปิดคือ ขา 9 ไม่เชื่อมต่อกับขา 5 เพื่อใช้สถานะปกติของอุปกรณ์ไฟฟ้าปิดการทำงานอยู่

<u>ส่วนที่ 4</u> ปลั๊กไฟ หรือเต้ารับ จะมีส่วนที่เป็นโลหะ ทำหน้าที่นำไฟฟ้า และเชื่อมต่อเต้าเสียบของอุปกรณ์ไฟฟ้า เมื่อมีไฟมาจ่ายให้กลับปลั๊กไฟ (เต้ารับ) โลหะจะสัมผัสเชื่อมต่อกันทำให้กระแสไฟฟ้าไหลผ่านไปสู่อุปกรณ์ไฟฟ้า

3.3 การออกแบบทาง Software

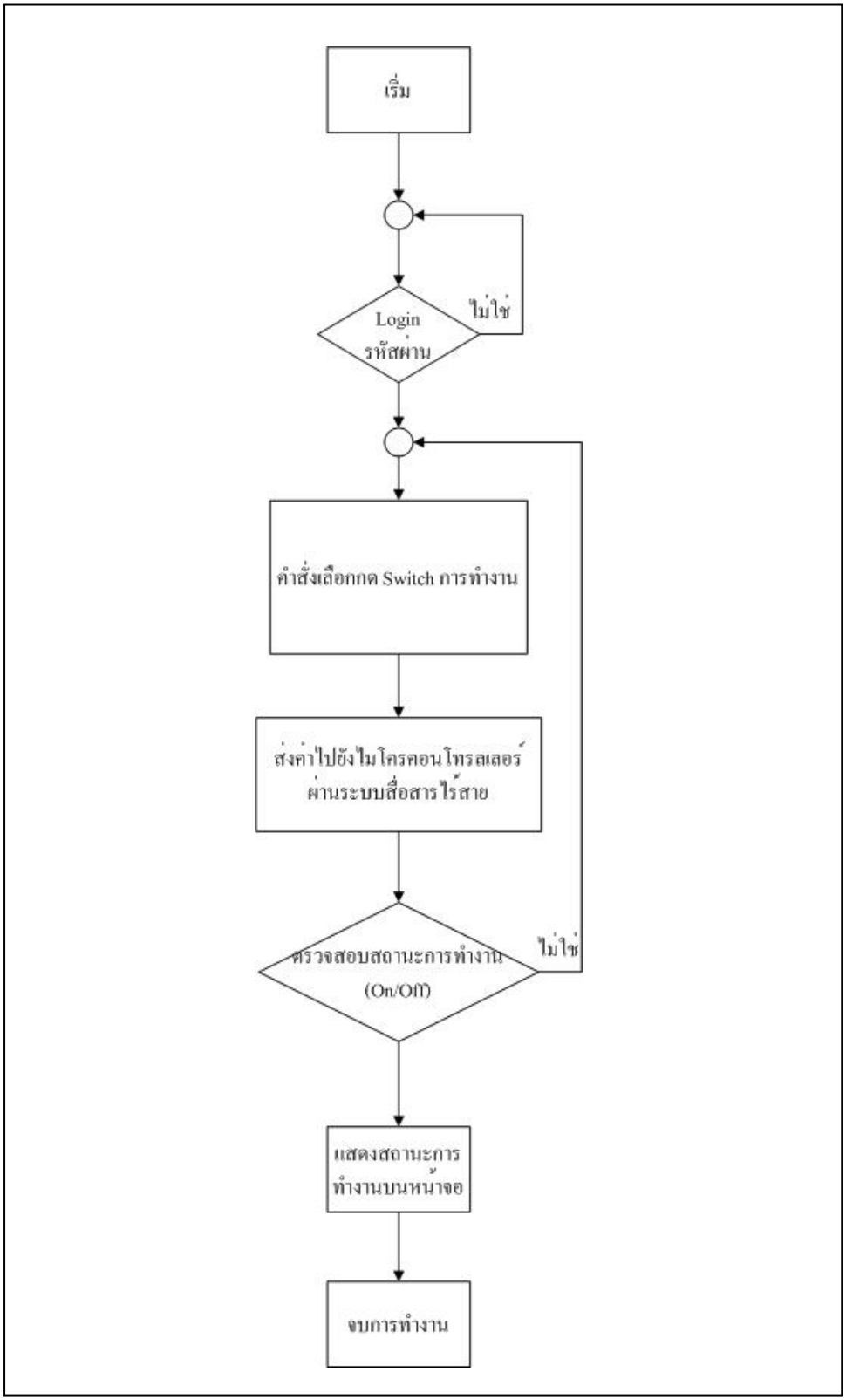
โปรแกรมพื้นฐานที่ได้ทำการเรียนรู้ คือ โปรแกรม Visual Basic 6.0 โปรแกรม Ultra Edit 32 โดยโปรแกรมที่ใช้ที่ใช้เขียน คือ ภาษาแอสเซมบลี (Assembly) และแปลงภาษาแอสเซมบลีให้เป็นภาษาเครื่อง นอกจากนี้ยังมีโปรแกรม FLIP ซึ่งใช้ในการโหลดไฟล์ให้กับหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ ซึ่งแสดงไว้ในภาคผนวกโปรแกรมควบคุมการทำงาน

ในส่วนนี้จะแบ่งการควบคุมการทำงานออกเป็นสองส่วนหลักๆ คือ ส่วนของโปรแกรมด้านส่งข้อมูล และส่วนของโปรมแกรมด้านรับข้อมูล และในส่วนของอุปกรณ์ชุดรับ-ส่งไร้สายจะต้องตั้งค่าโปรแกรมที่ทางบริษัทให้มา ทั้งตัวส่งสัญญาณและตัวรับสัญญาณ ซึ่งอธิบายในส่วนท้ายของการออกแบบทาง Software นี้

3.3.1 โปรแกรมการส่งข้อมูล

ในส่วนของโปรแกรมด้านส่งข้อมูลนั้น จะเริ่มจากการส่งข้อมูลไปยังฝั่งรับข้อมูลตลอดเวลา ซึ่งเป็นข้อมูล 8 บิต โดยข้อมูลนั้นจะเป็นข้อมูลที่บอกว่าให้สวิตช์ตัวไหนทำงาน โดยการทำงานของโปรแกรมจะส่งสัญญาณไปให้ชุดส่งสัญญาณโดยอธิบายดังแผนภาพรูปที่ 3.9

โฟลวชาร์ตโปรแกรม Visual Basic 6.0 ด้านส่งสัญญาณ



รูปที่ 3.9 แผนภาพโฟลวชาร์ตด้านส่งข้อมูล

โปรแกรมส่วนของภาคส่งสัญญาณ โปรแกรม Visual Basic 6.0

```
Private Sub Command10_Click()                    ; ในส่วนนี้จะแสดงข้อความ
MsgBox "เกี่ยวกับโปรแกรม" & vbCrLf & "1.โปรแกรมนี้จัดทำขึ้นเพื่อใช้ควบคุมการเปิด-ปิด อุปกรณ์ไฟฟ้า " & vbCrLf & "2.โปรแกรมนี้เขียนขึ้นเมื่อ มิถุนายน 2551" & vbCrLf & "3.โปรแกรมนี้เขียนโดย นางสาวชมพูนุท ยอดนวล และ นางสาวเปรมวดี ถาวรธนทรัพย์ ", vbOKOnly, "About"
End Sub

Private Sub Command1_Click()
clearbox
MSComm1.Output = "A"                             ; ให้ Output ของ Port Com 1 เป็น A
End Sub


Private Sub Command2_Click(Index As Integer)
clearbox
MSComm1.Output = "B"                             ; ให้ Output ของ Port Com 1 เป็น B
End Sub


Private Sub Command3_Click(Index As Integer)
clearbox
MSComm1.Output = "C"                             ; ให้ Output ของ Port Com 1 เป็น C
End Sub

Private Sub Command4_Click(Index As Integer)
clearbox
MSComm1.Output = "D"                             ; ให้ Output ของ Port Com 1 เป็น D
End Sub
```

```
Private Sub Command5_Click(Index As Integer)
clearbox
MSComm1.Output = "E"                    ; ให้ Output ของ Port Com 1 เป็น E
End Sub

Private Sub Command6_Click(Index As Integer)
clearbox
MSComm1.Output = "F"                    ; ให้ Output ของ Port Com 1 เป็น F
End Sub

Private Sub Command7_Click(Index As Integer)
clearbox
MSComm1.Output = "G"                    ; ให้ Output ของ Port Com 1 เป็น G
End Sub

Private Sub Command8_Click(Index As Integer)
clearbox
MSComm1.Output = "H"                    ; ให้ Output ของ Port Com 1 เป็น H
End Sub

Private Sub Command9_Click()
MsgBox "ขั้นตอนการใช้งาน" & vbCrLf & "1.เมื่อเรา Click ที่ Switch จะแสดงสถานะการทำงาน
ขึ้นมา เพื่อที่จะสามารถใช้ในการตรวจสอบสถานะการทำงานของ Switch ที่เราต้องการได้ " &
vbCrLf & "2.เมื่อเรา Click ที่ About จะแสดงรายละเอียดของโปรแกรม" & vbCrLf & "3.หากท่าน
ต้องการทราบรายละเอียดของโปรแกรม กรุณา Click ที่ About", vbOKOnly, "Help"    ; แสดง
ข้อความ

'frmLogin.Show
End Sub
```

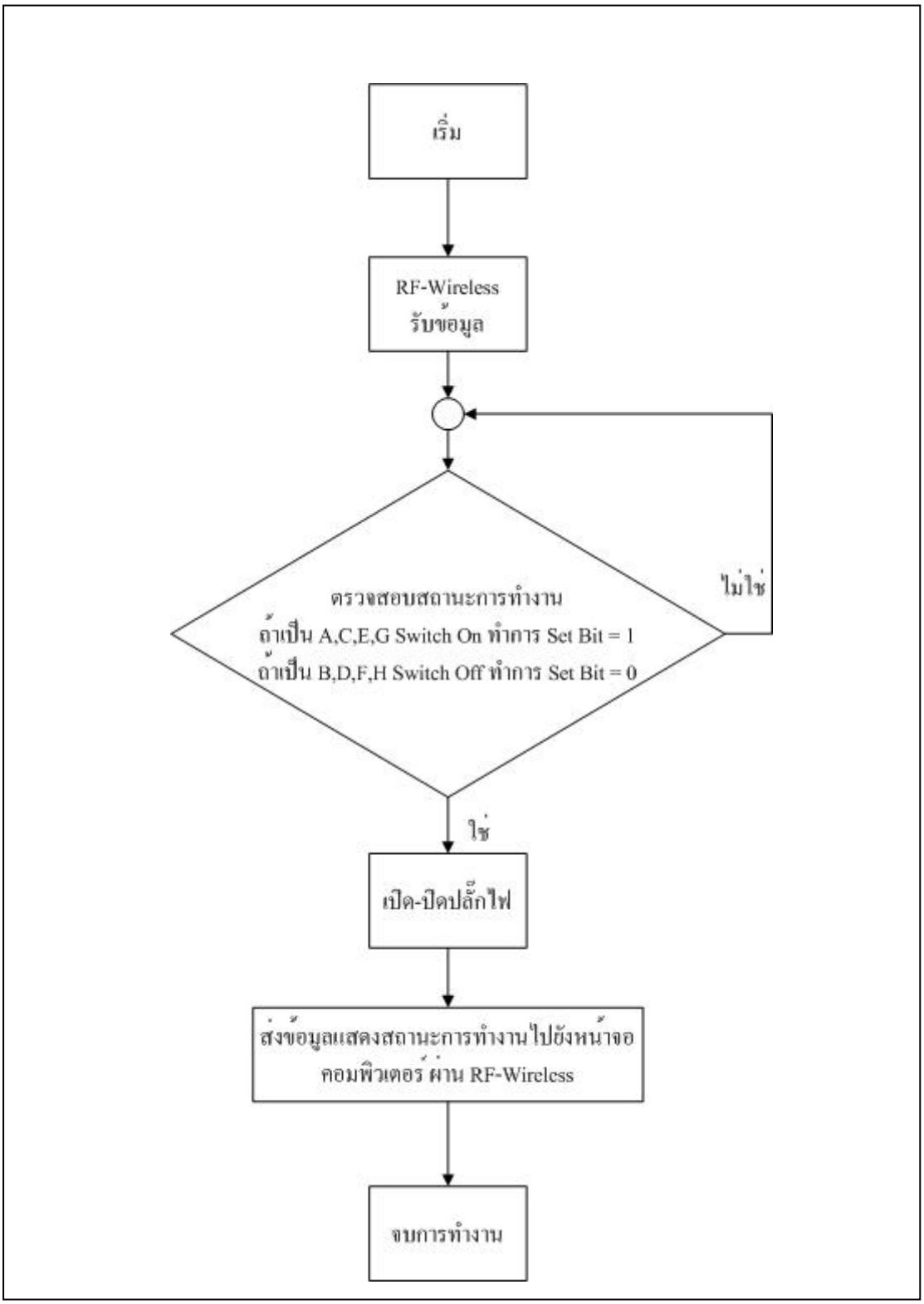
```
Private Sub clearbox()
RichTextBox1.Text = ""
End Sub
Private Sub Form_Load()

If LoginSuccess = False Then                    ; คำสั่งใช้ Login
frmLogin.Show                                   ; แสดง Login
Form1.Hide
End If


   MSComm1.CommPort = "1"                       ; Set ค่า ให้ใช้ Port com 1
   MSComm1.Settings = "9600, n , 8, 1"          ; ให้มีการหน่วงเวลา
   MSComm1.PortOpen = True

   Timer1.Interval = 100                        ; ประกาศ ใช้ Timer
   Timer1.Enabled = True
   RichTextBox1 = ""

End Sub

Private Sub Timer1_Timer()
On Error Resume Next
Const SpeedBaud = 16000
Dim LngSize As Long, X
   MSComm1.DTREnable = False
   MSComm1.DTREnable = True
   MSComm1.InputLen = 0
   Data = MSComm1.Input                                 ; ให้ข้อมูล ส่งผ่าน Port Com1
```

```
    If Data = "A" Then                                  ; ถ้าข้อมูลเป็น A
    Data = "Switch 1 On"                                ; ให้ Switch 1 ทำงาน (เปิด)
    End If
    If Data = "B" Then                                  ; ถ้าข้อมูลเป็น B
    Data = "Switch 1 Off"                               ; ให้ Switch 1 ทำงาน (ปิด)
    End If
    If Data = "C" Then                                  ; ถ้าข้อมูลเป็น  C
    Data = "Switch 2 On"                                ; ให้ Switch 1 ทำงาน (เปิด)
    End If
    If Data = "D" Then                                  ; ถ้าข้อมูลเป็น D
    Data = "Switch 2 Off"                               ;ให้ Switch 1 ทำงาน (ปิด)
    End If
    If Data = "E" Then                                  ; ถ้าข้อมูลเป็น E
    Data = "Switch 3 On"                                ; ให้ Switch 1 ทำงาน (เปิด)
    End If
    If Data = "F" Then                                  ; ถ้าข้อมูลเป็น F
    Data = "Switch 3 Off"                               ; ให้ Switch 1 ทำงาน (ปิด)
    End If
    If Data = "G" Then                                  ; ถ้าข้อมูลเป็น G
    Data = "Switch 4 On"                                ; ให้ Switch 1 ทำงาน (เปิด)
    End If
    If Data = "H" Then                                  ; ถ้าข้อมูลเป็น H
    Data = "Switch 4 Off"                               ; ให้ Switch 1 ทำงาน (ปิด)
    End If
    RichTextBox1.SelText = Data                         ; ให้ใส่ข้อมูล
    RichTextBox1.SelStart = Len(RichTextBox1.Text)

End Sub
```

3.3.2 โปรแกรมการรับข้อมูล

ในส่วนนี้จะใช้โปรแกรมไมโครคอนโทรลเลอร์ในการควบคุมสัญญาณด้านรับข้อมูล จะรอรับข้อมูลจากชุดอุปกรณ์เครือข่ายไร้สายผ่านพอร์ต RS232 และนำข้อมูลที่ได้ไปยังไมโครคอนโทรลเลอร์ เพื่อนำไปประมวลผลเปรียบเทียบกับค่าที่ได้ในโปรแกรมว่าตรงกันหรือไม่ ถ้าตรงกันก็จะทำการ SETB หรือ CLEAR P3 ที่บิตต่างๆ ตามคำสั่งที่ได้รับมาแล้วส่งสัญญาณไปยังชุดแผงเชื่อมต่ออุปกรณ์ไฟฟ้าให้ทำการเปิด-ปิดเครื่องใช้ไฟฟ้า ซึ่งอธิบายได้ดังแผนภาพรูปที่ 3.10

โฟลวชาร์ตโปรแกรมไมโครคอนโทรลเลอร์ ด้านรับสัญญาณ



รูปที่ 3.10 แผนภาพโฟลวชาร์ตด้านรับข้อมูล

โปรแกรมส่วนของการรับข้อมูล

```
CONTROL1    EQU    P3.4
CONTROL2    EQU    P3.5
CONTROL3    EQU    P3.6
CONTROL4    EQU    P3.7
```

ในส่วนนี้คือการตั้งค่าการใช้งานพอร์ต P3.4-P3.7 เมื่อกำหนดให้ตัวควบคุมที่ 1 ใช้งานพอร์ต P3.4 , ตัวควบคุมที่ 2 ใช้งานพอร์ต P3.5, ตัวควบคุมที่ 3 ใช้งานพอร์ต P3.6, ตัวควบคุมที่ 4 ใช้งานพอร์ต P3.7

```
        ORG    0000H
        JMP    0100H

        ORG    0100H
        MOV    SP,#2FH
        CLR    EA      ; Disable Intr All
        CALL   Initial_Serial
        CALL   Send_ClrScr
        CALL   DELAY_500M                    ;หน่วงเวลา

        CLR    CONTROL1
        CLR    CONTROL2
        CLR    CONTROL3
        CLR    CONTROL4
        CALL   Send_ClrScr
```

ในส่วนนี้เป็นเขียนโปรแกรมเริ่มต้นสถานะการทำงานที่ ตัวควบคุมทุกตัวเป็น 0 คือ CLEAR P3.4, CLEAR P3.5, CLEAR P3.6, CLEAR P3.7 ตั้งค่าเริ่มต้นของโปรแกรมการรับข้อมูล และมีคำสั่งการหน่วงเวลา

```
Main_LOOP:

LOOP_S1:
        CALL   DELAY_500M
        JNB    RI,$                    ;Wait data
        CLR    RI                      ;เคลียร์ค่า RI
        MOV    R0,SBUF
        MOV    A,R0
        clr    c                       ;เคลียร์ค่า c
        CJNE   A,#'B',S1_ON            ;ถ้า A ไม่เท่ากับ B กระโดดไปทำงาน
                                       ที่ฟังก์ชัน S1_ON
        CLR    CONTROL1                ;ถ้า A เท่ากับ B สวิตซ์ที่ 1 ปิด และ
                                       แสดง   SWITCH 1 ---> OFF
        CALL   DELAY_500M              ;หน่วงเวลา
        CALL   Send_ClrScr
        MOV    DPTR,#_OFF
        CALL   Send_Table
        JMP    Main_LOOP

S1_ON:  MOV    A,R0
        clr    c
        CJNE   A,#'A',LOOP_S2          ;ถ้า A ไม่เท่ากับ A กระโดดไปทำงาน
                                       ที่ฟังก์ชัน LOOP_S2
        SETB   CONTROL1                ;ถ้า A เท่ากับ A สวิตซ์ที่ 1 เปิด และ
                                       แสดง   SWITCH 1 ---> ON
        CALL   DELAY_500M
        CALL   Send_ClrScr
        MOV    DPTR,#_ON
        CALL   Send_Table
        JMP    Main_LOOP
```

```
LOOP_S2:
        MOV   A,R0
        clr   c
        CJNE  A,#'D',S2_ON          ;ถ้า A ไม่เท่ากับ  D กระโดดไปทำงาน
                                    ที่ฟังก์ชัน S2_ON
        CLR   CONTROL2              ;ถ้า A เท่ากับ D สวิตซ์ที่ 2  ปิด และ
                                    แสดง   SWITCH 2 ---> OFF
        CALL  DELAY_500M
        CALL  Send_ClrScr
        MOV   DPTR,#_OFF1
        CALL  Send_Table
        JMP   Main_LOOP

S2_ON:  MOV   A,R0
        clr   c
        CJNE  A,#'C',LOOP_S3        ;ถ้า A ไม่เท่ากับ  C กระโดดไปทำงาน
                                    ที่ฟังก์ชัน LOOP_S3
        SETB  CONTROL2              ;ถ้า A เท่ากับ C สวิตซ์ที่ 2  เปิด และ
                                    แสดง   SWITCH 2 ---> ON
        CALL  DELAY_500M
        CALL  Send_ClrScr
        MOV   DPTR,#_ON1
        CALL  Send_Table
        JMP   Main_LOOP

LOOP_S3:
        MOV   A,R0
        clr   c
        CJNE  A,#'F',S3_ON          ;ถ้า A ไม่เท่ากับ  F กระโดดไปทำงาน
                                    ที่ฟังก์ชัน S3_ON
```

```
        CLR     CONTROL3                ;ถ้า A เท่ากับ F สวิตซ์ที่ 3  ปิด และ
                                        แสดง   SWITCH 3 ---> OFF
        CALL    DELAY_500M
        CALL    Send_ClrScr
        MOV     DPTR,#_OFF2
        CALL    Send_Table
        JMP     Main_LOOP
S3_ON:  MOV     A,R0
        clr     c
        CJNE    A,#'E',LOOP_S4          ;ถ้า A ไม่เท่ากับ  E กระโดดไปทำงาน
                                        ที่ฟังก์ชัน LOOP_S4
        SETB    CONTROL3                ;ถ้า A เท่ากับ E สวิตซ์ที่ 3  เปิด และ
                                        แสดง   SWITCH 3 ---> ON
        CALL    DELAY_500M
        CALL    Send_ClrScr
        MOV     DPTR,#_ON2
        CALL    Send_Table
        JMP     Main_LOOP
LOOP_S4:
        MOV     A,R0
        CPL     A
        ANL     A,#'H'
        JNZ     S4_ON
        ;clr    c
        ;CJNE   A,#'H',S4_ON            ;ถ้า A ไม่เท่ากับ  H กระโดดไปทำงาน
                                        ที่ฟังก์ชัน S4_ON
        CLR     CONTROL4                ;ถ้า A เท่ากับ H สวิตซ์ที่ 4  ปิด และ
                                        แสดง   SWITCH 4 ---> OFF
        CALL    DELAY_500M
        CALL    Send_ClrScr
```

```
        MOV    DPTR,#_OFF3
        CALL   Send_Table
        JMP    Main_LOOP
S4_ON:  MOV    A,R0
        CPL    A
        ANL    A,#'G'
        JNZ    N1
        ;clr   c
        ;CJNE  A,#'G',Main_LOOP          ;ถ้า A ไม่เท่ากับ G กระโดดไปทำงาน
                                         ที่ฟังก์ชัน Main_LOOP
        SETB   CONTROL4                  ;ถ้า A เท่ากับ G สวิตซ์ที่ 4 เปิด และ
                                         แสดง  SWITCH 4 ---> ON
        CALL   DELAY_500M
        CALL   Send_ClrScr
        MOV    DPTR,#_ON3
        CALL   Send_Table
        JMP    Main_LOOP

N1:     JMP    Main_LOOP
        $INCLUDE    "Serial.Sub"
```

คำอธิบายโปรแกรม

ในส่วนนี้จะเป็นการประมวลผลข้อมูลเปรียบเทียบโดยใช้คำสั่ง CJNE ซึ่งเป็นตัวรีจิสเตอร์ใช้เทียบค่า ASCII โดยการเปรียบเทียบว่าข้อมูลที่ได้รับมานั้นเท่ากับเงื่อนไขหรือไม่

ที่ Switch 1 ถ้าหากเท่ากับ ‘A’ ให้ SETB P3.4 คือ Switch 1 on

ที่ Switch 2 ถ้าหากเท่ากับ ‘C’ ให้ SETB P3.5 คือ Switch 2 on

ที่ Switch 3 ถ้าหากเท่ากับ ‘E’ ให้ SETB P3.6 คือ Switch 3 on

ที่ Switch 4 ถ้าหากเท่ากับ ‘G’ ให้ SETB P3.7 คือ Switch 4 on

แต่ถ้าไม่เท่ากับค่า ASCII ที่กำหนดให้กระโดดไปทำฟังก์ชันที่กำหนดไว้โดย

ที่ Switch 1 ถ้าหากเท่ากับ ‘B’ ให้ CLR P3.4 คือ Switch 1 off

ที่ Switch 2 ถ้าหากเท่ากับ ‘D’ ให้ CLR P3.5 คือ Switch 2 off

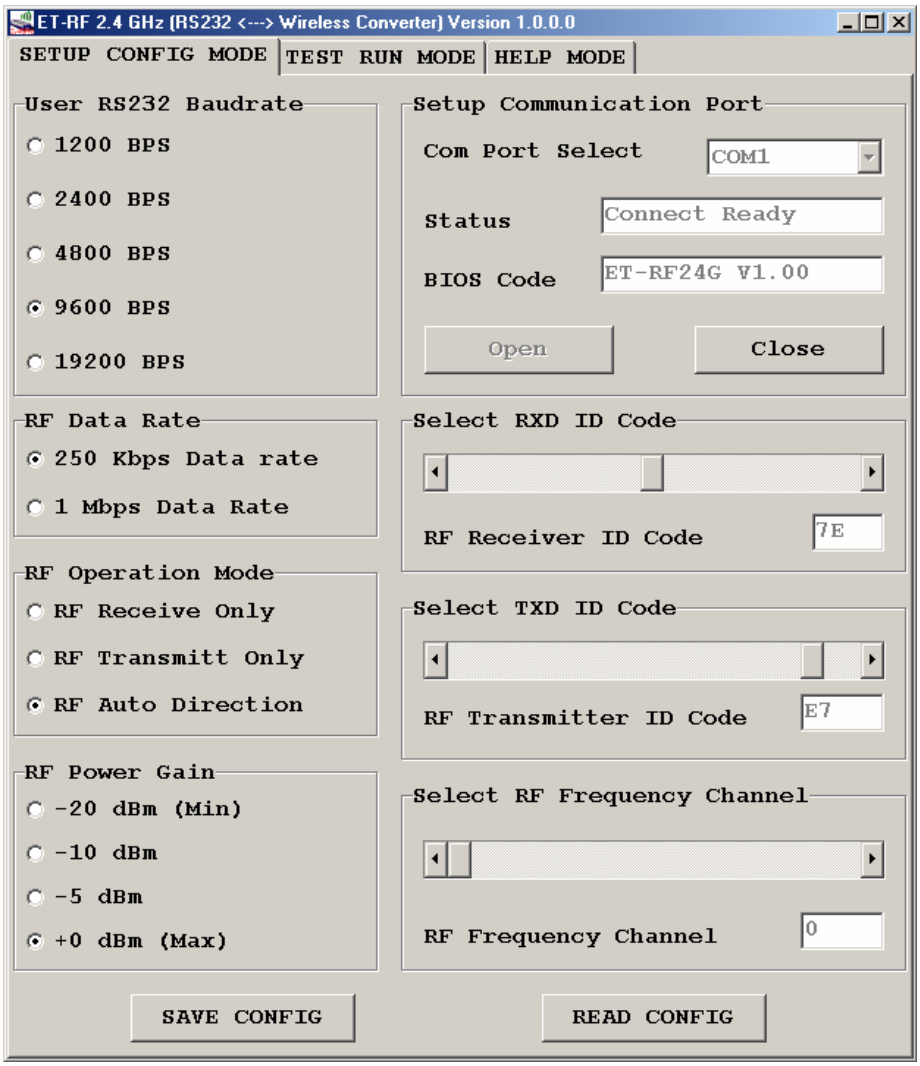
ที่ Switch 3 ถ้าหากเท่ากับ ‘F’ ให้ CLR P3.6 คือ Switch 3 off

ที่ Switch 4 ถ้าหากเท่ากับ ‘H’ ให้ CLR P3.7 คือ Switch 4 off

```
_ON:    DB      'SWITCH 1 ---> ON',00           ;แสดงค่าว่า SWITCH 1 ---> ON
_OFF:   DB      'SWITCH 1 ---> OFF',00          ;แสดงค่าว่า SWITCH 1 ---> OFF
_ON1:   DB      'SWITCH 2 ---> ON',00           ;แสดงค่าว่า SWITCH 2 ---> ON
_OFF1:  DB      'SWITCH 2 ---> OFF',00          ;แสดงค่าว่า SWITCH 2 ---> OFF
_ON2:   DB      'SWITCH 3 ---> ON',00           ;แสดงค่าว่า SWITCH 3 ---> ON
_OFF2:  DB      'SWITCH 3 ---> OFF',00          ;แสดงค่าว่า SWITCH 3 ---> OFF
_ON3:   DB      'SWITCH 4 ---> ON',00           ;แสดงค่าว่า SWITCH 4 ---> ON
_OFF3:  DB      'SWITCH 4 ---> OFF',00          ;แสดงค่าว่า SWITCH 4 ---> OFF

        END                                     ;จบการทำงาน
```

การตั้งค่าของชุดรับส่งไร้สาย

ในส่วนนี้จะอธิบายการตั้งค่าของชุดรับส่งข้อมูลไร้สาย ซึ่งจะแตกต่างกันระหว่าง ตัวส่งสัญญาณ และตัวรับสัญญาณ โดยจะอธิบายจากโปรแกรมที่ทางบริษัทได้ผลิตตั้งค่าไว้ ซึ่งจะแสดงรายละเอียดนั้นในภาคผนวก



รูปที่ 3.11 โปรแกรมการตั้งค่าของชุดรับส่งข้อมูลไร้สาย

3.4 รูปภาพของระบบ

ในส่วนนี้จะแสดงภาพลักษณะการเชื่อมต่อของแต่ละอุปกรณ์ การใช้งานของแต่ละอุปกรณ์ และการติดตั้งอุปกรณ์เข้าด้วยกัน ซึ่งอธิบายได้ดังนี้

3.4.1 ชุดอุปกรณ์รับส่งข้อมูลผ่านระบบการสื่อสารไร้สาย

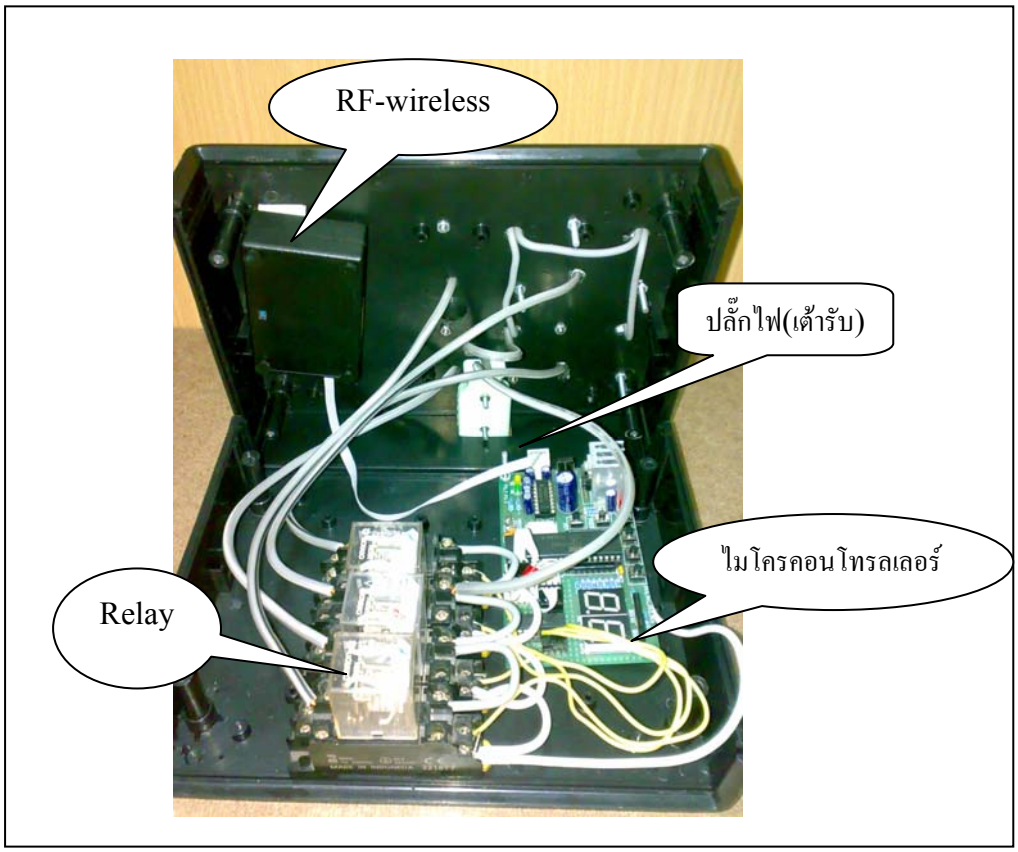
ชุดรับส่งข้อมูลแบบไร้สายในส่วนของภาคส่งสัญญาณนั้นมีการเชื่อมต่อกับคอมพิวเตอร์ผ่าน RS232 แต่ในส่วนของภาครับสัญญาณจะมีการเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านRS232 โดยใช้สายไฟ 4 เส้นต่อกับคอนเน็กเตอร์เพื่อเชื่อมต่อกับไมโครคอนโทรลเลอร์

รูปที่ 3.12 ชุดอุปกรณ์รับส่งข้อมูลแบบไร้สาย

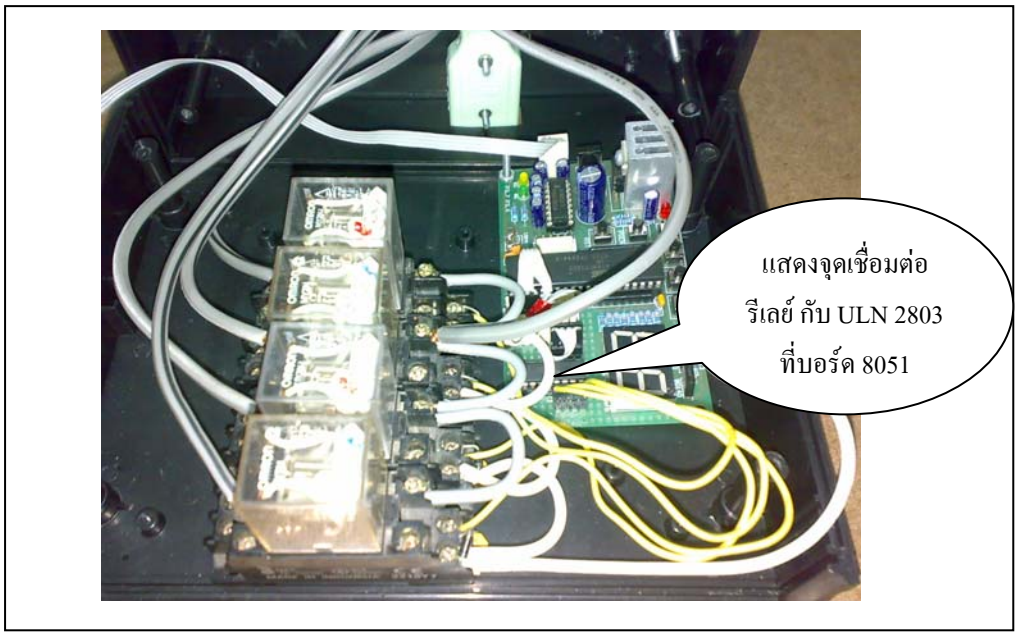
รูปที่ 3.13 พอร์ต RS232

3.4.2 แผงเชื่อมต่ออุปกรณ์ไฟฟ้า

ในส่วนของอุปกรณ์ของด้านรับสัญญาณจะมีการรวมอุปกรณ์ทั้งหมด คือปลั๊กไฟ รีเลย์ และไมโครคอนโทรลเลอร์กับULN2803 ได้ออกแบบสร้างขึ้นเพื่อนำมาประกอบรวมไว้ภายในกล่องอุปกรณ์ โดยจะมี RF-wireless ชุดรับส่งข้อมูลไร้สายรวมอยู่ภายในกล่องด้วย



รูปที่ 3.14 กล่องแผงเชื่อมต่ออุปกรณ์ไฟฟ้าที่ด้านรับสัญญาณ



รูปที่ 3.15 การเชื่อมต่ออุปกรณ์ภายในกล่อง

รูปที่ 3.16 การเชื่อมต่อระหว่างชุดรับส่งข้อมูลไร้สายกับแผงเชื่อมต่ออุปกรณ์ไฟฟ้าภายนอก

รูปที่ 3.17 กล่องอุปกรณ์แผงเชื่อมต่ออุปกรณ์ไฟฟ้าภายนอก

บทที่ 4
ผลการทดลอง

การทดลองที่ 1 การทดสอบคำสั่งของโปรแกรม

ในการทดสอบคำสั่งของโปรแกรม เราได้ทำการทดสอบสวิตช์ทั้งหมด 4 ตัว โดยแต่ละตัวเราจะทำการทดสอบทั้งคำสั่งเปิด และคำสั่งปิด

วัตถุประสงค์

เพื่อทดสอบคำสั่งของโปรแกรม

ขั้นตอนการทดลอง

1. ติดตั้งอุปกรณ์ภาครับและภาคส่ง
2. เลือกสวิตช์ตัวที่ 1 ทดสอบคำสั่งเปิดและคำสั่งปิด บันทึกผล
3. เลือกสวิตช์ตัวที่ 2 ทดสอบคำสั่งเปิดและคำสั่งปิด บันทึกผล
4. เลือกสวิตช์ตัวที่ 3 ทดสอบคำสั่งเปิดและคำสั่งปิด บันทึกผล
5. เลือกสวิตช์ตัวที่ 4 ทดสอบคำสั่งเปิดและคำสั่งปิด บันทึกผล

ผลการทดลองที่ 1

ผลการทดลองที่ได้แสดงดังตารางที่ 1

ตารางแสดงผลการทดลองที่ 1 การทดสอบคำสั่งของโปรแกรม

อุปกรณ์ที่	คำสั่ง	สถานะโปรแกรม	สถานะของอุปกรณ์ที่ต้องการควบคุม
1	เปิด	เปิด	ทำงาน
	ปิด	ปิด	ไม่ทำงาน
2	เปิด	เปิด	ทำงาน
	ปิด	ปิด	ไม่ทำงาน
3	เปิด	เปิด	ทำงาน
	ปิด	ปิด	ไม่ทำงาน
4	เปิด	เปิด	ทำงาน
	ปิด	ปิด	ไม่ทำงาน

วิเคราะห์ผลการทดลองที่ 1

จากการทดลองจะพบว่า เมื่อเราใช้คำสั่งเปิด สวิตช์ทุกตัว จะทำงานตามคำสั่งเปิด และเมื่อเราใช้คำสั่งปิด สวิตช์ทุกตัว จะทำงานตามคำสั่งปิด

สรุปผลการทดลองที่ 1

การทดลองนี้สามารถสรุปได้ว่า สวิตช์ทุกตัวสามารถทำตามคำสั่งได้ 100%

การทดลองที่ 2 การทดสอบระยะทางการรับ-ส่งข้อมูล

การทดลองในส่วนนี้จะทดสอบระยะทางในการรับ-ส่งข้อมูล โดยการทดสอบระยะทางในการรับ-ส่งข้อมูลกลางแจ้งนั้นจะแบ่งระยะการรับ-ส่งข้อมูลออกเป็น 50 เมตร, 100 เมตร, 150 เมตร, 200 เมตร, 250 เมตร, 300 เมตร, 350 เมตร สำหรับการทดสอบตำแหน่งการทำงานภายในห้อง-นอกห้องเราจะแบ่งตำแหน่งออกเป็น 5 ตำแหน่ง (ตำแหน่งอ้างอิงตามหมายเหตุ)

วัตถุประสงค์

เพื่อทดสอบระยะทางการรับ-ส่งข้อมูลเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless

ขั้นตอนการทดลอง

1. ติดตั้งอุปกรณ์ภาครับและภาคส่ง
2. ทดสอบระยะกลางแจ้ง โดยแบ่งระยะทางออกเป็นดังนี้ 50 เมตร, 100 เมตร, 150 เมตร, 200 เมตร, 250 เมตร, 300 เมตร, 350 เมตร กำหนดให้แต่ละระยะทางทดสอบสวิซท์ทั้งหมด 4 ตัว แล้วบันทึกผลการทดลองในตารางที่ 1
3. ทดสอบตำแหน่งการทำงานภายในห้อง-นอกห้อง โดยกำหนดจุดทดสอบทั้งหมด 5 จุด (แสดงดังหมายเหตุ) กำหนดให้แต่ละตำแหน่งการทดสอบ สวิซท์ทั้งหมด 4 ตัว แล้วบันทึกผลการทดลองในตารางที่ 2

ผลการทดลอง

ผลการทดลองที่ได้แสดงดังตารางที่ 1 และตารางที่ 2

<u>ตารางแสดงผลการทดลองที่ 2 แสดงระยะทางการรับ-ส่งข้อมูล</u>

ตารางที่ 2.1 แสดงระยะทางกลางแจ้ง

ระยะทาง (เมตร)	อุปกรณ์ที่ (สวิตช์ที่)	คำสั่ง	สถานะอุปกรณ์ที่ต้องการควบคุม
50	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด
100	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด
150	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด

ระยะทาง (เมตร)	อุปกรณ์ที่ (สวิตช์ที่)	คำสั่ง	สถานะอุปกรณ์ที่ต้องการควบคุม
200	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด
250	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด
300	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด

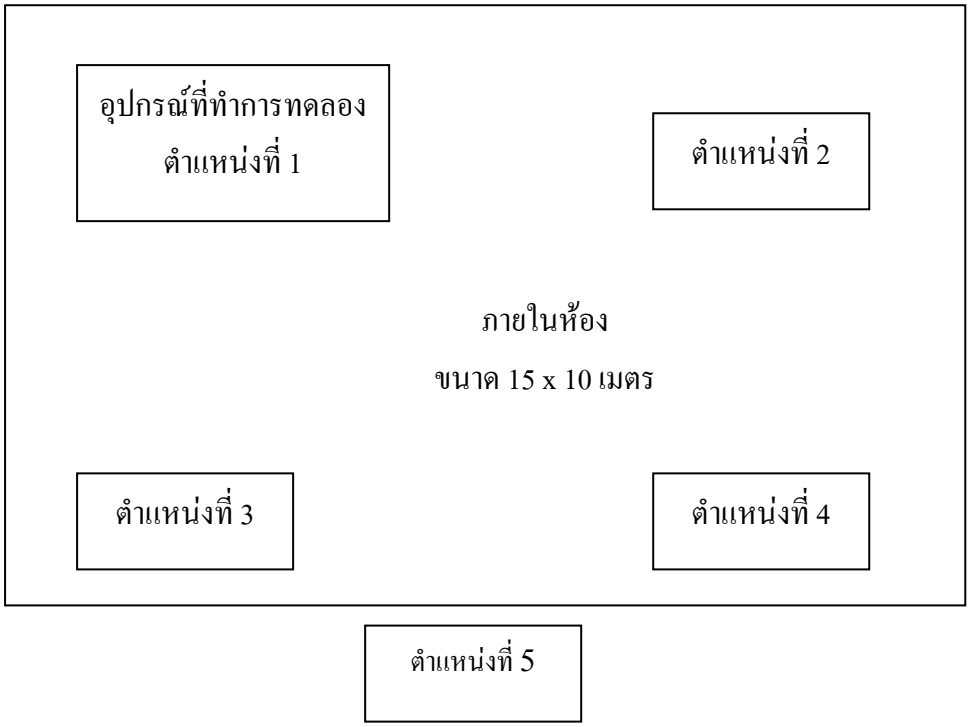
ระยะทาง (เมตร)	อุปกรณ์ที่ (สวิตช์ที่)	คำสั่ง	สถานะอุปกรณ์ที่ต้องการควบคุม
350	1	เปิด	ไม่ตอบสนอง
		ปิด	ไม่ตอบสนอง
	2	เปิด	ไม่ตอบสนอง
		ปิด	ไม่ตอบสนอง
	3	เปิด	ไม่ตอบสนอง
		ปิด	ไม่ตอบสนอง
	4	เปิด	ไม่ตอบสนอง
		ปิด	ไม่ตอบสนอง

ตารางที่ 2.2 แสดงตำแหน่งการทำงานภายใน – ภายนอกห้อง

ตำแหน่งที่	อุปกรณ์ที่ (สวิตช์ที่)	คำสั่ง	สถานะอุปกรณ์ที่ต้องการควบคุม
1	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด

ตำแหน่งที่	อุปกรณ์ที่ (สวิตช์ที่)	คำสั่ง	สถานะอุปกรณ์ที่ต้องการควบคุม
2	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด
3	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด
4	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด

ตำแหน่งที่	อุปกรณ์ที่ (สวิตช์ที่)	คำสั่ง	สถานะอุปกรณ์ที่ต้องการควบคุม
5	1	เปิด	เปิด
		ปิด	ปิด
	2	เปิด	เปิด
		ปิด	ปิด
	3	เปิด	เปิด
		ปิด	ปิด
	4	เปิด	เปิด
		ปิด	ปิด



รูปที่ 4.1 แผนผังแสดงตำแหน่งของภาคส่งสัญญาณ

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

5.1.1 สรุปผลการทดลองการทดสอบคำสั่งของโปรแกรม

จากการทดลองจะพบว่าเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless สามารถทำงานได้ตามคำสั่ง ทั้งคำสั่งเปิดและคำสั่งปิด

5.1.2 สรุปผลการทดลองการทดลองระยะรับ-ส่งข้อมูล

จากการทดลองพบว่าเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless สามารถรับ-ส่งข้อมูลได้ไกลสุดที่ระยะ 300 เมตรในบริเวณกลางแจ้ง และเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน RF-Wireless สามารถรับ-ส่งข้อมูลได้ทั้งภายในห้องและภายนอกห้อง แม้จะมีผนังห้องหรือสิ่งของกั้นก็ตาม

5.2 ข้อเสนอแนะและแนวทางการวิจัยในอนาคต

1. ในการออกแบบ ควรจะทำให้ส่วนแสดงผลแสดงสถานะการทำงานเป็นเสียง และสามารถแสดงภาพออกมาได้ ปัญหานี้อาจจะทำได้โดยการออกแบบโปรแกรมใหม่โดยเพิ่มคำสั่งการทำงานเพิ่มขึ้นทั้งภาคส่งและภาครับ

2. ในการทดลองจะพบว่าระยะทางในการทดลองน้อยเกินไป น่าจะรับ-ส่งสัญญาณได้ไกลกว่านี้ ปัญหารนี้อาจจะทำได้โดยออกแบบให้ RF-Wireless ส่งสัญญาณด้วยกำลังส่งที่มากกว่านี้

3. เนื่องจากอุปกรณ์ภาครับต้องใช้ไฟ 220 โวลท์ เป็นการลำบากในการใช้งาน น่าจะหาแหล่งจ่ายไฟอื่นมาแทนไฟ 220 โวลท์

4. ออกแบบระบบให้สามารถทำการควบคุมจากระยะไกลโดยผ่านทางระบบโทรศัพท์พื้นฐาน โทรศัพท์เคลื่อนที่ หรือ อินเตอร์เน็ต

ภาคผนวก ก
โปรแกรมคอมพิวเตอร์

1. ไมโครคอนโทรลเลอร์

บอร์ดไมโครคอนโทรลเลอร์ในตระกูล MCS51 ขนาด 40 Pin ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89C51ED2 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ภายในตัวถังแบบ DIP40 โดย MCU ตัวนี้จะมีจุดเด่น คือ เรื่องของความเร็วในการประมวลผล ซึ่งสามารถทำงานได้ด้วยความถี่สูงสุด 60MHz ที่ 12 Clock / 1 Machine Cycle นอกจากนี้แล้วยังมีความเพียบพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ EEPROM ขนาด 2 KByte หรือหน่วยความจำใช้งานแบบ RAM ซึ่งมีมากถึง 1792 Byte ส่วนในด้านของ
อุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดี โดยจะมีทั้ง SPI, UART, Watchdog, Timer/Counter, PWM ฯลฯ โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งาน และสะดวกต่อการพัฒนาโปรแกรม

คุณสมบัติของบอร์ด

เลือกใช้ MCU ตระกูล MCS51 เบอร์ AT89C51ED2 ของ ATMEL เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งสามารถกำหนดการทำงานของ MCU ให้ทำงานในโหมดความเร็ว 2 เท่า (X2 Mode) ได้ ทำให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 58.9824 MHz โดยคุณสมบัติเด่นๆของ MCU ได้แก่

o มีหน่วยความจำ Flash สำหรับเขียนโปรแกรมขนาด 64KByte

o มี EEPROM ขนาด 2 KByte สำหรับเก็บข้อมูล และ สามารถเขียนซ้ำได้กว่า 1 ล้านครั้ง

o มีพอร์ต I/O ขนาด 8 บิต จำนวน 4 พอร์ต (P0,P1,P2 และ P3)

o มี RAM ใช้งาน 1792 Byte เพียงพอต่อการใช้งาน

o มีวงจรสื่อสารอนุกรม UART จำนวน 1 พอร์ต และมีวงจรสื่อสาร SPI จำนวน 1 พอร์ต

o มีวงจร Timer/Counter ขนาด 16 บิต จำนวน 3 ชุด

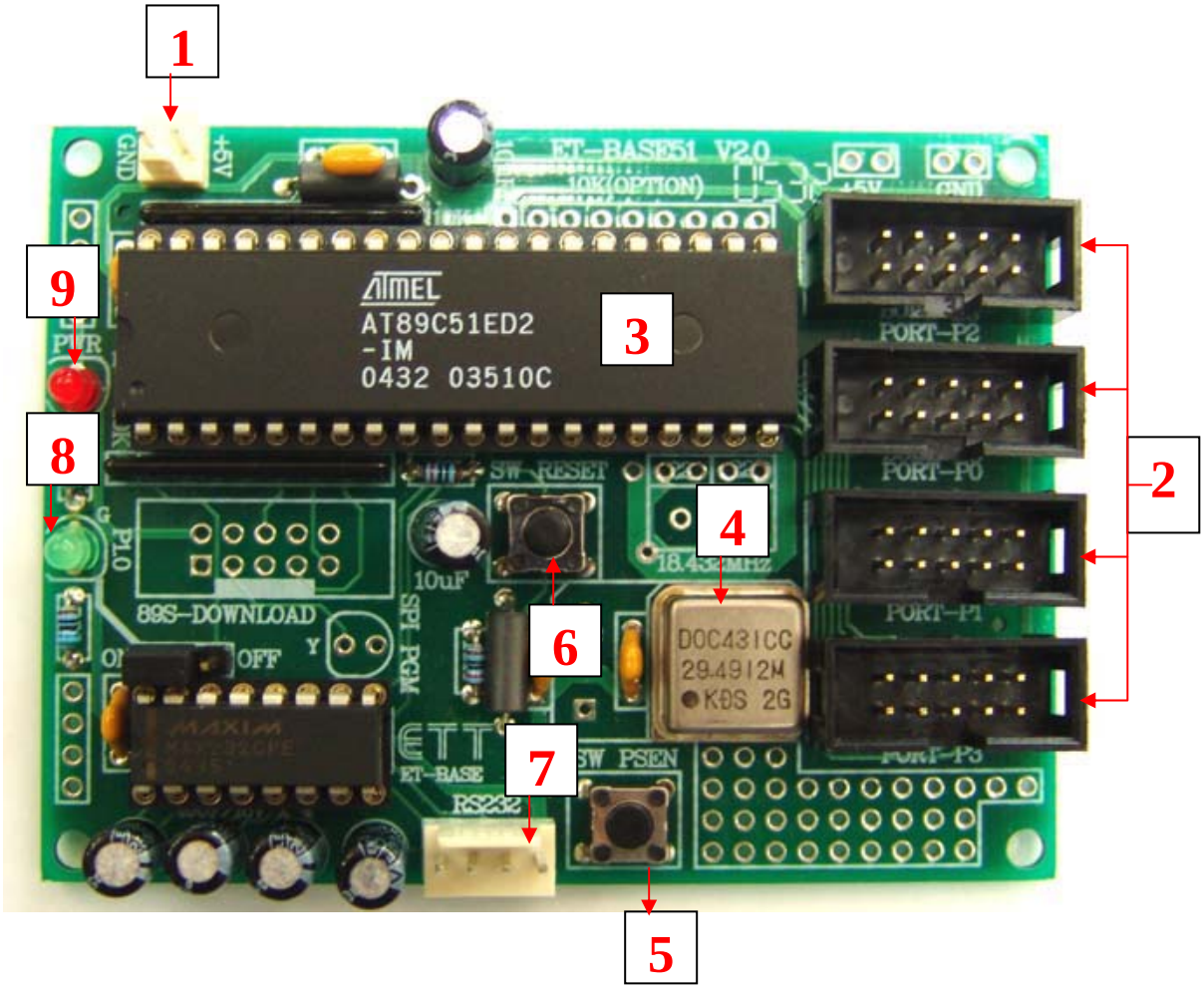
o มีวงจร Watchdog, Power-ON Reset, Capture/Compare ,PWM

มีพอร์ตสื่อสารอนุกรมแบบ RS232 จำนวน 1 ช่อง สำหรับใช้ในการ Download โปแกรมให้กับบอร์ด และประยุกต์ใช้งานทั่วไป

มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header ขนาด 2x5 จำนวน 4 ชุด (P0,P1,P2 และ P3)

มี LED แสดงสถานะแหล่งจ่าย Power และ Self-Test สำหรับใช้ทดสอบการทำงานของบอร์ดใช้แหล่งจ่ายไฟขนาด +5VDC ขนาด PCB Size เล็กเพียง 8 x 6 cm.

โครงสร้างบอร์ด



รูปที่ ก.1 ไมโครคอนโทรลเลอร์ตระกูล MCS 51 เบอร์ AT89C51ED2

• **หมายเลข 1** คือ ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่ายไฟตรง +5VDC

• **หมายเลข 2** คือ PORT-P2, PORT-P0, PORT-P1 และ PORT-P3 ตามลำดับ

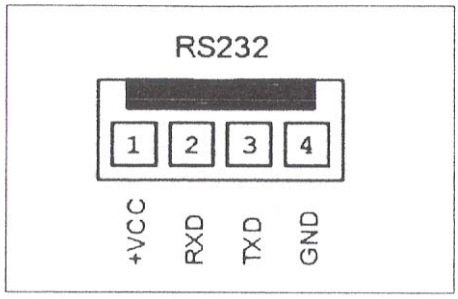
• **หมายเลข 3** คือ MCU เบอร์ AT89C51ED2 ซึ่งเป็น MCU ตระกูล MCS51 จาก ATMEL

• **หมายเลข 4** คือ Oscillator Module ค่า 29.4912 MHz

• **หมายเลข 5** คือ SW PSEN ใช้สำหรับกำหนดสถานะโลจิก “0” ให้ขา PSEN สำหรับใช้ในขั้นตอนของการ Download HEX File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ด

• **หมายเลข 6** คือ SW RESET ใช้สำหรับรีเซ็ตการทำงานของ MCU ภายในบอร์ด

• **หมายเลข 7** คือ พอร์ต RS232 หรือ Serial Port สำหรับติดต่อกับอุปกรณ์มาตรฐาน RS232 และใช้เป็น ISP Download Connector สำหรับ Download HEX File ให้กับ MCU ของบอร์ด



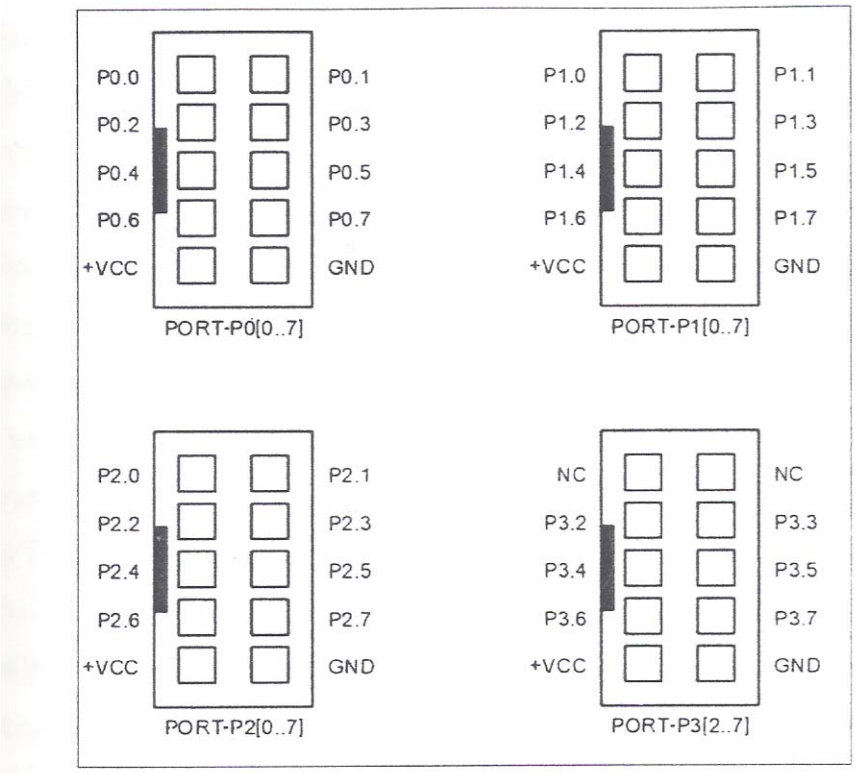
รูปที่ ก.2

• **หมายเลข 8** คือ LED แสดงสถานะของ P1.0 ในกรณีที่เลือก Jumper ไว้ทางด้าน ON เพื่อ Test

• **หมายเลข 9 เป็น** LED Power สีแดง ใช้แสดงสถานะของแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด

1.1 ขั้วต่อสัญญาณต่างๆ

สำหรับขั้วต่อสัญญาณของพอร์ต I/O จาก MCU นั้น จะถูกออกแบบและจัดเตรียมไว้ผ่านทางขั้วต่อแบบ IDE Header ขนาด 2x5 จำนวน 4 ชุด คือ PORT-P0, PORT-P1, PORT-P2 และ PORT-P3ตามลำดับ โดยที่ขั้วต่อสัญญาณแต่ละชุด จะประกอบไปด้วยสัญญาณของ I/O ที่เชื่อมต่อมาจากขาสัญญาณของ MCU โดยตรงทั้งหมด โดยแต่ละพอร์ตจะมีสัญญาณพอร์ตละ 8 บิต ยกเว้น PORT-P3 ซึ่งจะมีเพียง 6 บิต คือ P3.2-P3.7 เท่านั้น ส่วน P3.0 และ P3.1 จะถูกสงวนไว้ใช้งานเป็นขาสัญญาณ RXDและ TXD สำหรับรับส่งข้อมูลของ RS232 ซึ่งสัญญาณทั้ง 2 เส้น (P3.0 และ P3.1) จะถูกเชื่อมต่อผ่านวงจรLine Driver (MAX232) สำหรับแปลงระดับสัญญาณจากระดับโลจิก TTL ของ MCU ให้เป็นสัญญาณแรงดันตามมาตรฐานของ RS232 โดยสัญญาณที่ได้รับการแปลงเป็นแบบ RS232 จะถูกเชื่อมต่อไปรอไว้ที่ขั้วต่อแบบ CPA ขนาด 4 PIN (RS232) โดยการจัดเรียงสัญญาณของแต่ละชุด จะเป็นดังรูปที่ ก.3



รูปที่ ก.3 การจัดเรียงสัญญาณของพอร์ต I/O ต่างๆของบอร์ด

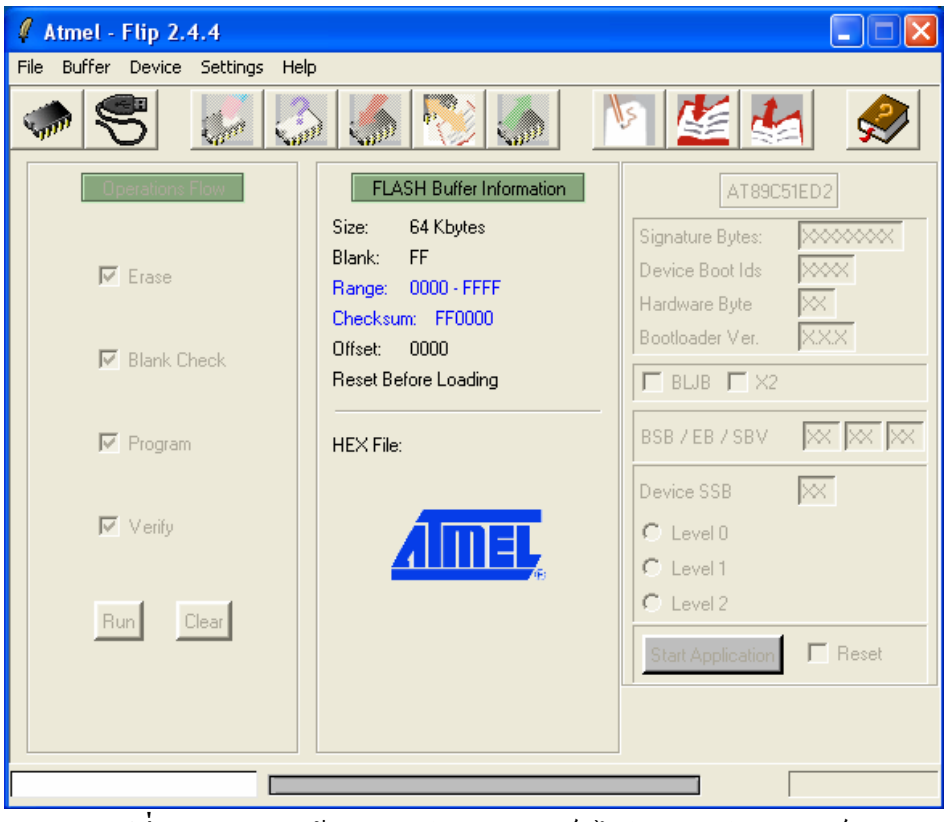
1.2 การ Download Hex file ให้กับ MCU ของบอร์ด

การ Download Hex File ให้กับหน่วยความจำ Flash ของ MCU ในบอร์ดนั้น จะใช้โปรแกรมชื่อ“FLIP” ของ ATMEL ซึ่งจะติดต่อกับ MCU ผ่าน Serial Port ของคอมพิวเตอร์ PC โดยโปรแกรมดังกล่าวสามารถดาวน์โหลดได้จาก WWW.ATMEL.COM โดยไม่เสียค่าใช้จ่ายใดๆ แต่สำหรับในกรณีที่ซื้อบอร์ดจาก อีทีที นั้น โปรแกรมดังกล่าวจะจัดเตรียมไว้ให้ในแผ่น CD ROM อยู่แล้วโปรแกรม FLIP (Flexible In-system Programmer) เป็นโปรแกรมสำหรับพัฒนาระบบของไมโครคอนโทรลเลอร์ของ ATMEL โดยสามารถใช้สนับสนุนการพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล MCS51 ในกลุ่มที่ใช้การพัฒนาแบบ ISP ซึ่งรวมถึงเบอร์ **AT89C51ED2** ด้วย โดยโปรแกรมจะทำงานภายใต้ระบบปฏิบัติการของWindows9X/Me/NT/2000 และ Windows XP โดยสนับสนุนการเชื่อมต่อกับระบบฮาร์ดแวร์ที่ใช้การเชื่อมต่อแบบ RS232 หรือ CAN หรือ USB ซึ่งวิธีการเชื่อมต่อของโปรแกรม FLIP กับระบบฮาร์ดแวร์ของไมโครคอนโทรลเลอร์นั้น จะขึ้นอยู่กับความสามารถของตัวไมโครคอนโทรลเลอร์ที่จะนำมาทำการพัฒนาว่าสามารถใช้การติดต่อสื่อสารด้วยวิธีใดได้บ้าง แต่สำหรับไมโครคอนโทรลเลอร์เบอร์

AT89C51ED2 นั้นจะสามารถใช้การเชื่อมต่อผ่านทางพอร์ตอนุกรม RS232เท่านั้น ไม่สามารถเชื่อมต่อผ่านระบบการสื่อสารของ CAN หรือ USB ได้ โดยโปรแกรม FLIP จะใช้สำหรับ Download ข้อมูลให้กับหน่วยความจำของไมโครคอนโทรลเลอร์ที่ทำงานใน Monitor Mode เพื่อให้ผู้ใช้สั่งจัดการกับหน่วยความจำภายในตัว CPU ไม่ว่าจะเป็นการ สั่งลบข้อมูล(Erase) สั่งตรวจสอบข้อมูลในหน่วยความจำ(Blank Check) สั่งโปรแกรมข้อมูลให้กับหน่วยความจำโปรแกรมของ CPU (Program) สั่งเปรียบเทียบข้อมูลจาก Buffer กับหน่วยความจำในตัว CPU (Verify) หรือสั่งอ่านข้อมูลจากหน่วยความจำของ CPU (Read) เป็นต้น โดยในการ Download HEX File จาก PC ให้กับบอร์ดจะใช้กับสาย RS232 แบบ4 Pin ในการสั่ง Download โปรแกรม ซึ่งสามารถใช้งานได้กับโปรแกรม FLIP ทุกรุ่น แต่ในปัจจุบัน (ตุลาคม2548) โปรแกรม FLIP ได้รับการปรับปรุงเป็น “FLIP V2.4.4” แล้ว ซึ่งเมื่อต้องการให้โปรแกรม FLIP ติดต่อกับ CPU ใน Monitor Mode ด้วยวิธีการ Download แบบ Manual นั้น จะต้องสั่ง Reset ให้ CPU เข้าทำงานใน Monitor Mode ก่อน จึงจะสามารถสั่งงาน CPU ผ่านทางโปรแกรม FLIP ได้ ซึ่งหลักการสำหรับReset ให้ CPU เข้าทำงานใน Monitor Mode จะต้องกำหนดให้ขาสัญญาณ PSEN มีสภาวะเป็น “0”ในขณะที่ CPU หลุดพ้นจากสภาวะของการ Reset ซึ่งตามปรกติแล้วหลังการ Reset ทุกครั้ง CPU จะตรวจสอบสภาวะของขาสัญญาณ PSEN ว่าเป็น “0” หรือไม่ถ้าไม่ใช่ก็จะทำงานในโหมดการทำงานปรกติแต่ถ้าใช่ก็จะตรวจสอบสถานะของสัญญาณอื่นๆที่เกี่ยวข้องกับการทำงานใน Monitor Mode ถ้าเงื่อนไขอื่นๆถูกต้องก็จะเข้าทำงานใน Monitor Mode ทันที สำหรับบอร์ด รุ่น ET-BASE51 V2.0 (AT89C51ED2)นั้น การที่จะสั่ง Reset ให้ CPU ของ ATMEL เข้าทำงานใน Monitor Mode ด้วยวิธีการแบบ Manual นั้นจะต้องทำตามลำดับขั้นตอนดังต่อไปนี้ (อ้างอิงการใช้งานกับโปรแกรม FLIP Version 2.4.4 เป็นหลัก ถ้าใช้รุ่นที่แตกต่างไปจากนี้ลักษณะโปรแกรมอาจแตกต่างไปจากนี้ได้)

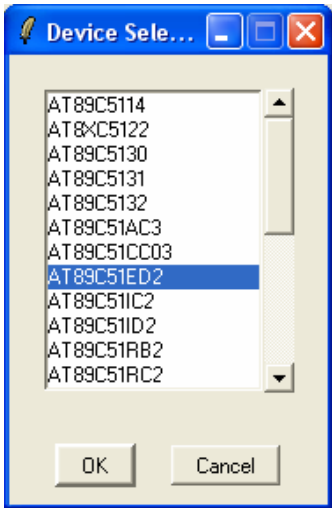
1.3 ลำดับขั้นตอนการ Download HEX File ด้วยโปรแกรม FLIP 2.4.4

1. ต่อสายสัญญาณ RS232 จาก Com Port ของเครื่องคอมพิวเตอร์ PC เข้ากับขั้ว RS232 แบบ 4 Pin ของบอร์ด
2. จ่ายไฟเลี้ยงวงจรให้บอร์ด ซึ่งจะสังเกตุเห็น LED แสดงสถานะของ PWR สีแดงติดสว่างอยู่
3. สั่ง Run โปรแกรม FLIP V2.4.4 ซึ่งจะได้ผลดังรูปที่ ก.4



รูปที่ ก.4 แสดงหน้าของการโหลดลงบอร์ดไมโครคอนโทรลเลอร์

4. สั่งเลือกกำหนดเบอร์ของ MCU ที่ติดตั้งไว้ในบอร์ด โดยเลือก Device ⟶ Select ซึ่งต้องเลือกกำหนดให้ตรงกับที่ทำการติดตั้งไว้จริงๆในบอร์ดด้วย ดังตัวอย่าง (AT89C51ED2)



รูปที่ ก.5 แสดงการเลือกกำหนดเบอร์ CPU ของ ET-BASE51 V2.0 (AT89C51ED2)

5. คลิกเมาส์ที่คำสั่ง Setting ⟶ Communication ⟶ RS232 จากนั้นเลือกกำหนด Comport ให้ตรงกับที่ต่อสายไว้จริง ดังรูปที่ ก.6 (ในตัวอย่างใช้ Com1)

รูปที่ ก.6 แสดงการเลือกพอร์ตเชื่อมต่อกับไมโครคอนโทรลเลอร์

6. ทำการรีเซ็ต MCU ให้เข้าทำงานใน Monitor โดยมีลำดับขั้นตอนดังนี้

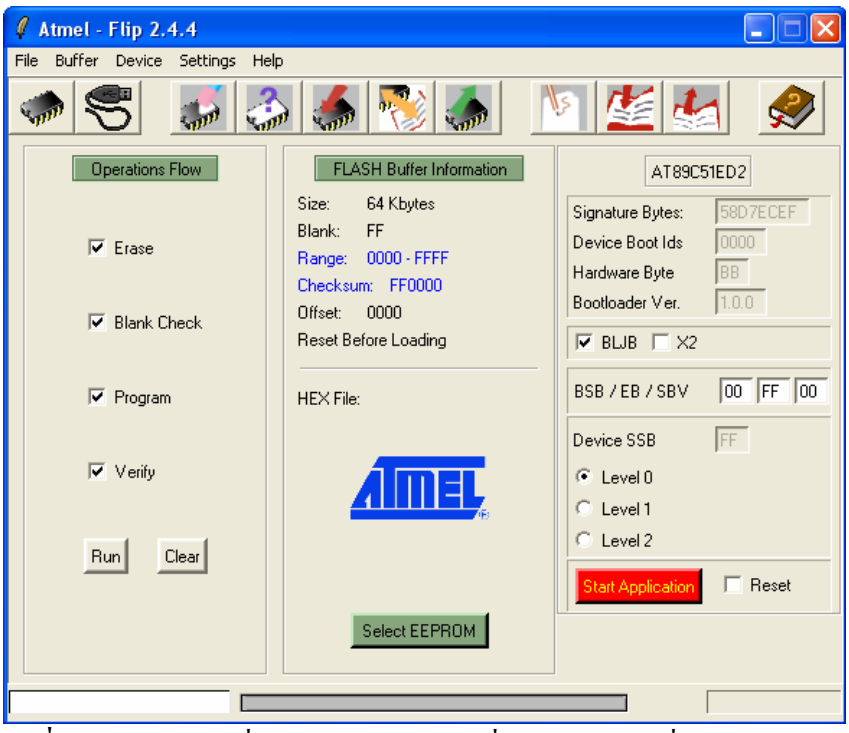
a) กดสวิตช์ PSEN ค้างไว้เพื่อกำหนดสถานะขาสัญญาณ PSEN ให้เป็น “0”

b) กดสวิตช์ RESET เพื่อส่งสัญญาณ RESET ให้กับ CPU โดยสวิตช์ PSEN ต้องกดค้างอยู่เช่นเดิม

c) ปล่อยสวิตช์ RESET เพื่อปล่อยให้ CPU พ้นจากสภาวะการ Reset (สวิตช์ PSEN ยังกดค้างอยู่)

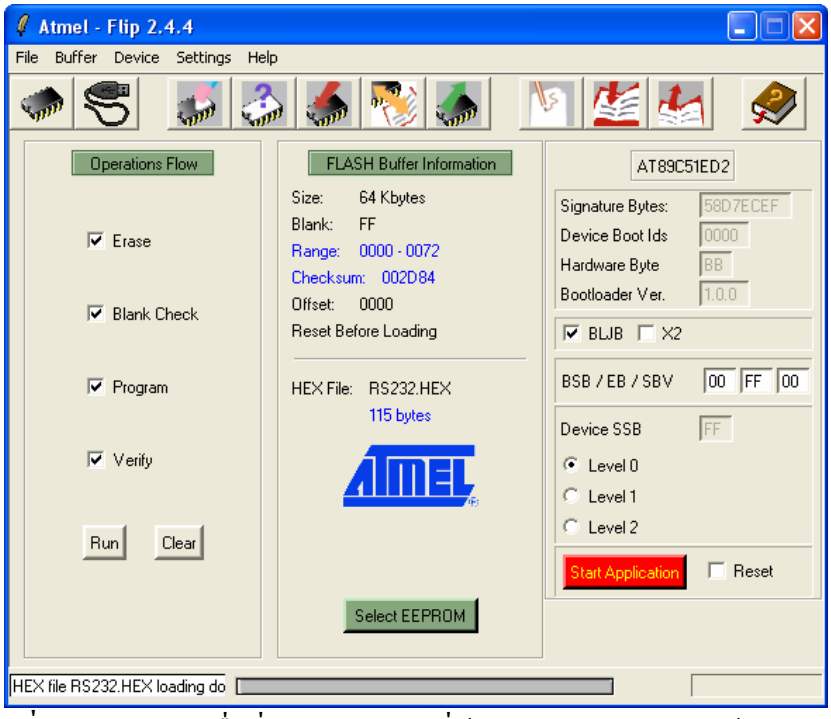
d) ปล่อยสวิตช์ PSEN เป็นลำดับสุดท้าย

7. คลิกเมาส์ที่ปุ่ม Connect เพื่อทำการติดต่อสื่อสารกับ MCU ใน Monitor Mode ซึ่งจะได้ผลดังรูปที่ ก.7



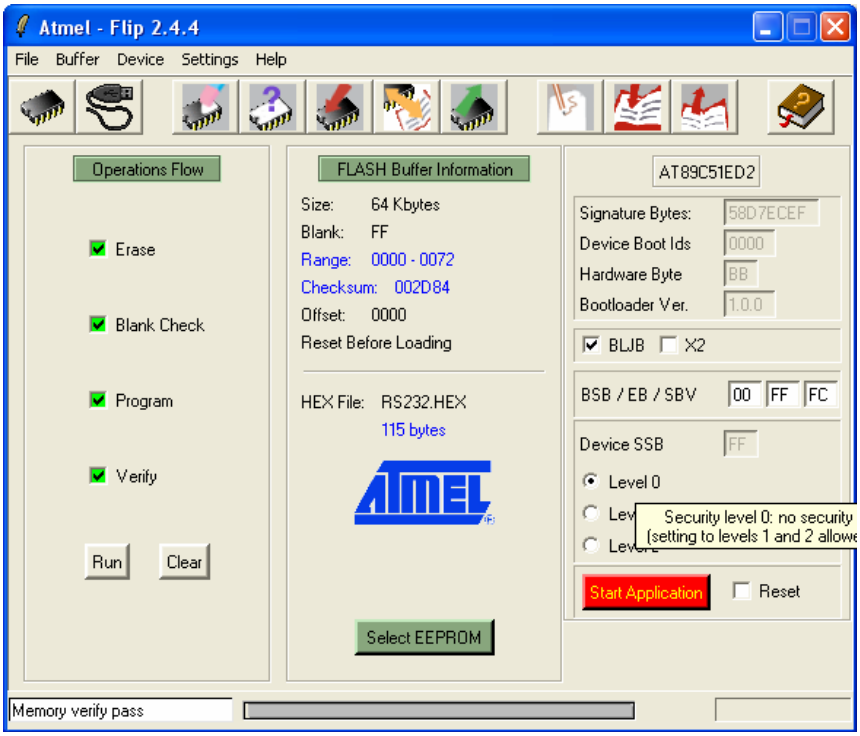
รูปที่ ก.7 แสดงภาพเมื่อคลิกปุ่ม Connect เพื่อทำการติดต่อสื่อสารกับ MCU

8. สั่งเปิด Hex File ที่ต้องการจะ Download ให้กับ MCU มารอไว้ใน Buffer ของโปรแกรม FLIP โดยใช้คำสั่ง “File → Load Hex File…”



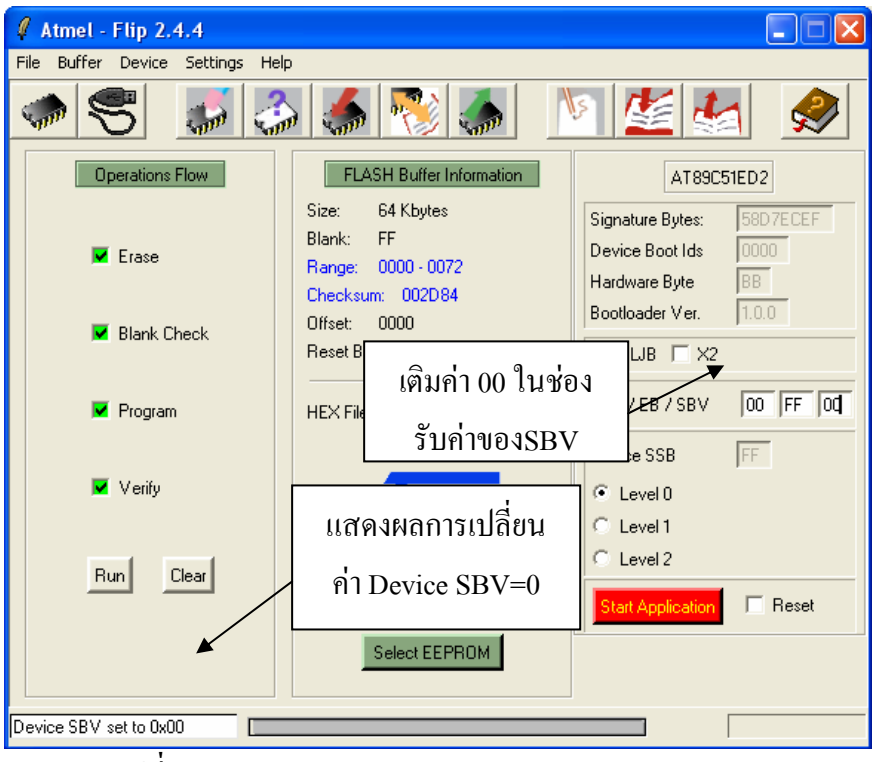
รูปที่ ก.8 แสดงภาพเมื่อสั่งเปิด Hex File ที่ต้องการจะ Download ให้กับ MCU

9. คลิกเมาส์ที่หน้าตัวเลือกคำสั่งใน Tab ของ Operation Flow ให้ครบทุกคำสั่ง ซึ่งได้แก่ Erase,Bank Check, Program, Verify จากนั้นคลิกเมาส์ที่ปุ่มคำสั่ง Run และรอจนการทำงานของโปรแกรมเสร็จเรียบร้อยดังรูปที่ ก.9



รูปที่ ก.9 แสดงภาพเมื่อคลิกเมาส์ที่ปุ่มคำสั่ง Run

10. ตรวจสอบค่า Device BSB และ SBV ว่ามีค่าเป็น 00 ทั้งหมดแล้วหรือยัง ซึ่งถ้ายังไม่เป็น 00 ให้ทำการแก้ไขค่าให้เป็น 00 โดยคลิกเมาส์ในช่องตัวเลขแล้วพิมพ์ค่า 00 แทนที่ลงไปทั้ง 2 ช่องดังรูปที่ ก.10



รูปที่ ก.10 แสดงภาพตรวจสอบค่า Device BSB และ SBV

11. ทำการกดสวิตช์ Reset ให้กับบอร์ดเพื่อให้บอร์ดเริ่มต้นทำงานตามโปรแกรมที่ได้ทำการ Downloadไปให้ ซึ่งถ้าไม่เกิดความผิดพลาดใดๆจะเห็น MCU เริ่มต้นทำงานทันที

1.4 ปัญหาต่างๆในขณะใช้งานโปรแกรม FLIP และแนวทางการแก้ไข

ในบางครั้งเมื่อเรียกใช้คำสั่งต่างๆของโปรแกรม FLIP แล้ว อาจเกิดความผิดพลาดบางประการขึ้นซึ่งอาจไม่ใช่ปัญหาที่เกิดจากความบกพร่องของระบบฮาร์ดแวร์ แต่อาจเกิดการการกำหนดพารามิเตอร์บางอย่างในโปรแกรมไม่ถูกต้องหรือข้ามขั้นตอนบางประการไป ซึ่งเมื่อโปรแกรม FLIP ไม่สามารถปฏิบัติตามคำสั่งที่ผู้ใช้งานสั่งไปได้สำเร็จจะแสดงอาการ Error ต่างๆให้ทราบ ซึ่งพอสรุปได้ดังนี้

1. Time Out Error เป็นความผิดพลาดที่เกิดจากการที่โปรแกรม FLIP ไม่สามารถทำการสื่อสารกับ CPUใน Monitor Mode ได้ ซึ่งอาจเกิด หลายสาเหตุ เช่น

- การต่อสายสัญญาณระหว่างขั้วต่อ RS232 ของบอร์ด ET-BASE51 V2.0 กับขั้วต่อพอร์ตสื่อสารอนุกรม RS232 ของคอมพิวเตอร์ยังไม่เรียบร้อยหรือต่อไม่ตรงกับที่กำหนดตัวเลือกไว้ในโปรแกรม หรือการกำหนดรูปแบบและตัวเลือกต่างๆในการสื่อสารไม่ถูกต้อง เมื่อพบปัญหานี้ให้ลองทำการ

ตรวจสอบค่าต่างๆในการสื่อสารใน “Setting ⟶ Preferences.. และ Setting ⟶Communication ⟶ RS232”

- ยังไม่ได้รีเซ็ตให้ CPU เข้าทำงานใน Monitor Mode รอไว้ก่อนที่จะสั่งงานโปรแกรมในกรณี Download แบบ Manual หรือบอร์ดยังไม่พร้อมทำงาน เช่น ยังไม่ได้จ่ายไฟเลี้ยงให้บอร์ด
- กำหนดค่า Baudrate เร็วเกินไป ซึ่งในกรณีที่ใช้งานกับเครื่องคอมพิวเตอร์ที่มีความเร็วมากๆนั้น ควรกำหนดค่า Baudrate ในการสื่อสารให้ช้าลง ซึ่งอาจใช้ค่า 19200 หรือ 9600 ก็พอเพราะถ้ากำหนดให้ความเร็วมากเกินไป เมื่อโปรแกรม FLIP ส่งข้อมูลให้กับ CPUแบบต่อเนื่องนั้น อาจทำให้ CPU ไม่สามารถประมวลผลคำสั่งหรือข้อมูลต่างๆที่ส่งไปให้ทันก็จะทำให้เกิดความผิดพลาดบ่อยครั้งขึ้น

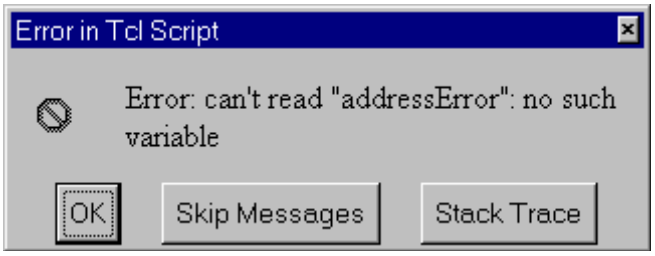
2. Software Security Bit Set. Cannot access device Data เป็นความผิดพลาดที่เกิดจากการนำ CPUที่มีการสั่ง Lock Bit ของ Security Bit ไว้ก่อนแล้ว จึงมาสั่ง Program หรือ Verify หรือ Read ในภายหลังโดยยังไม่ได้สั่งลบข้อมูลเก่าออกเสียก่อน ซึ่งให้แก้ปัญหาด้วยการสั่งลบข้อมูล (Erase) เสียก่อนแล้วจึงสั่งเขียนข้อมูลใหม่อีกครั้งหนึ่ง

3. The board reply is not correct เป็นความผิดพลาดที่เกิดจากการสื่อสารข้อมูลระหว่างโปรแกรม FLIP กับ ไมโครคอนโทรลเลอร์ เกิดความผิดพลาดในลักษณะของ Framing Error ขึ้น ซึ่งปัญหาอาจเกิดจากการกำหนดค่า Baudrate ไม่ถูกต้องกับค่าความถี่ของ Crystal ที่ใช้กับบอร์ด

4. The RS232 port could not be opened เป็นความผิดพลาดที่เกิดจากโปรแกรม FLIP ไม่สามารถสั่งเปิดการทำงานของพอร์ตสื่อสารอนุกรม RS232 ของเครื่องคอมพิวเตอร์ PC ได้ ซึ่งอาจเกิดจากการกำหนดหมายเลข Comport ในโปรแกรมที่เลือกไว้ไม่มีอยู่จริง หรือมีโปรแกรมอื่นเรียกใช้งาน Comportนั้นค้างอยู่ หรือเรียกใช้งานโปรแกรม FLIP ในขณะที่กำลังสั่งเปิดโปรแกรมอื่นๆที่มีการใช้งาน Comport__อยู่ด้วย ซึ่งให้ลองปิดโปรแกรม FLIP แล้วสั่งเปิดโปรแกรมใหม่ดู ถ้ายังเกิดปัญหาเดิมอยู่อีกอาจลองตรวจสอบสาเหตุอื่นๆที่เกี่ยวข้องและทำการแก้ไข

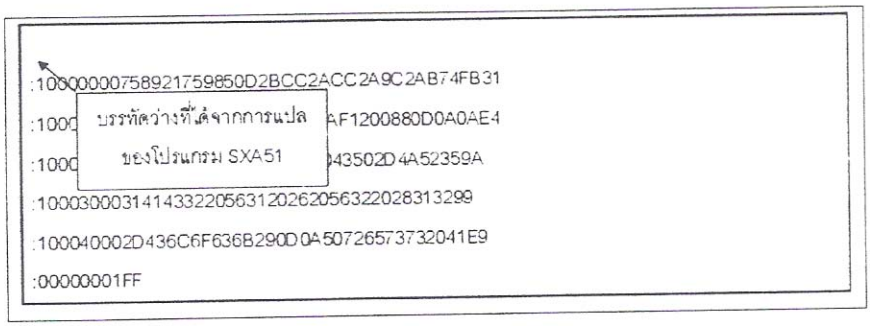
5. Check sum error เป็นความผิดพลาดที่เกิดจากการที่ CPU รับข้อมูลที่ส่งไปจากคอมพิวเตอร์ PC ไม่ครบถูกต้องทั้งหมด ซึ่งปัญหาอาจเกิดจากการกำหนดความเร็วในการสื่อสาร Baudrate เร็วเกินไป หรือกำหนดไว้ไม่เหมาะสมกับค่าความถี่ Crystal ให้ลองเปลี่ยนค่า Baudrate ให้ช้าลงกว่าเดิม ซึ่งค่าที่เหมาะสมได้แก่ 9600,19200 และ 38400 แต่ถ้าคอมพิวเตอร์ไม่เร็วมากนักก็อาจกำหนดเป็น 57600หรือ 115200 ก็ได้ แต่ถ้ากำหนดค่าสูงๆแล้วเกิด Error ควรลดค่า Baudrate ให้ช้าลงกว่าเดิม

6. การสั่ง Load HEX ไม่ได้ เป็นความผิดพลาดที่เกิดจากการที่โปรแกรม FLIP ไม่สามารถอ่านข้อมูลในHEX File ออกมาได้ ซึ่งอาจเกิดจากไฟล์ที่สั่งโหลดนั้น ไม่ใช่ไฟล์แบบ Intel HEX เนื่องจากโปรแกรมFLIP สามารถใช้งานกับไฟล์แบบ Intel HEX เท่านั้น ส่วนไฟล์ในรูปแบบอื่นๆ จะไม่สามารถนำมาใช้งานกับโปรแกรมนี้ได้ ส่วนปัญหาอีกประการหนึ่งที่มักพบอยู่บ่อยๆ คือ โปรแกรม FLIP ไม่สามารถอ่าน HEXFile ได้ทั้งๆที่ไฟล์ที่สั่งให้อ่านนั้นเป็นไฟล์แบบ Intel HEX อยู่แล้ว ซึ่งที่พบอยู่บ่อยๆก็ได้แก่ HEX File ที่สั่งแปลโดยใช้โปรแกรม Assembler ของ SXA51.EXE เนื่องจาก HEX File ที่ได้จากการแปลของโปรแกรมตัวนี้จะเกิดบรรทัดว่างอยู่ในไฟล์ในส่วนเริ่มต้นบรรทัดแรกด้วย 1 บรรทัด ซึ่งตามรูปแบบของHEX File แล้ว ในแต่ละบรรทัดของไฟล์จะต้องเริ่มต้นด้วยเครื่องหมายโคลอน (:) แล้วตามด้วยข้อมูลต่างๆในบรรทัดนั้น แต่เมื่อบรรทัดแรกเป็นบรรทัดว่างโปรแกรมจึงแสดง Error ว่าไม่ใช่ HEX File โดยโปรแกรม FLIP จะแสดง Error .ให้ทราบดังรูปที่ ก.11

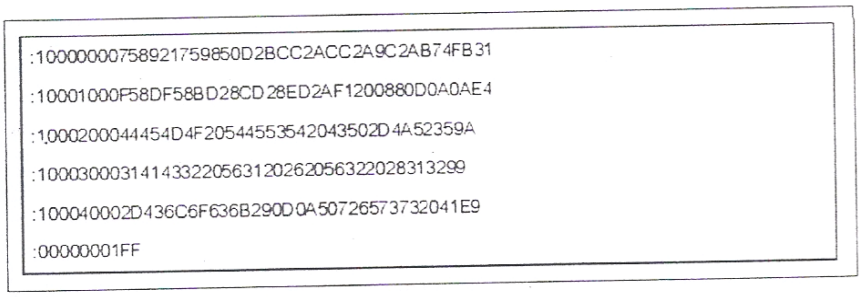


รูปที่ ก.11 แสดง Error ว่าไม่ใช่ HEX File

สำหรับวิธีการแก้ไขปัญหานี้ให้ใช้โปรแกรม Text Editor เปิด HEX File ที่ได้จากการแปลของSXA51.EXE แล้วตัดบรรทัดว่างในไฟล์นั้นทิ้งไปแล้วสั่งบันทึกใหม่ก็จะสามารถนำไปใช้ได้แล้ว



รูปที่ ก.12 แสดง ลักษณะของ HEX File ที่ได้จาก SXA51 ซึ่งจะเกิดบรรทัดว่างอยู่ 1 บรรทัด



รูปที่ ก.13 แสดงลักษณะของ HEX File ที่สามารถใช้กับโปรแกรม FLIP ได้ หลังตัดบรรทัดว่างทิ้งไป

7. เมื่อสั่งโปรแกรมข้อมูลให้กับ CPU เรียบร้อยแล้วหลังจากรีเซ็ตบอร์ดแล้วไม่ทำงาน ซึ่งปัญหานี้อาจเกิดจากสาเหตุความผิดพลาดหลายประการ ซึ่งพอสรุปได้ดังนี้ คือ

- โปรแกรมที่เขียนขึ้นไม่ถูกต้องยังไม่สามารถทำงานได้เอง ซึ่งปัญหานี้ผู้ใช้ต้องหาทางตรวจสอบและแก้ไขความผิดพลาดที่เกิดขึ้นเอง

- ยังไม่ได้มีการสั่ง Load HEX เข้ามารอไว้ยัง Buffer แล้วสั่งโปรแกรม (Program Device) ซึ่งโปรแกรม FLIP จะนำข้อมูลที่อยู่ใน Buffer เขียนไปยังหน่วยความจำของโปรแกรม

- สวิตช์ PSEN อาจถูกกดค้างอยู่ จึงทำให้การรีเซ็ตบอร์ดทุกๆครั้งนั้น CPU จะเข้าไปทำงานใน Monitor Mode เสมอ ซึ่งปัญหานี้สามารถตรวจสอบได้โดยการวัดระดับลอจิกทีขาสัญญาณ PSEN ของ CPU ซึ่งอยู่ที่ขา 29 (DIP 40) ซึ่งควรมีสภาวะเป็น “1” ถ้าไม่มีการกดสวิตช์ PSEN ไว้ และควรมีสภาวะเป็น “0” ถ้ามีการกดสวิตช์ PSEN ไว้

- สวิตช์ RESET อาจถูกกดค้างอยู่ จึงทำให้ CPU ไม่สามารถหลุดพ้นจากสภาวะการรีเซ็ตได้ ซึ่งปัญหานี้สามารถตรวจสอบได้โดยการวัดระดับลอจิกที่ขาสัญญาณ RESET ของ CPU ซึ่งอยู่ที่ขา 9(DIP 40) ซึ่งควรมีสภาวะเป็น “0” ถ้าไม่มีการกดสวิตช์ RESET ไว้ และควรมีสภาวะเป็น “1” ถ้ามีการกดสวิตช์ RESET ไว้

- ค่าของ Device BSB และ SBV ยังไม่ได้ถูกกำหนดให้มีค่าเป็น 00H ไว้ ซึ่งจะทำให้โปรแกรมกระโดดไปทำงานยังตำแหน่งที่ชี้โดย Device SBV แทน ซึ่งถ้าค่าของ Device SBV ไม่ใช้ศูนย์ก็จะเหมือนกับว่าโปรแกรมไม่ทำงาน ซึ่งการแก้ไข ปัญหานี้ หลังจากสั่งโปรแกรมข้อมูลให้กับ CPU เรียบร้อยแล้ว ควรกำหนดให้ค่าของ Device BSB และ Device SBV มีค่าเป็น 00H ไว้ทั้งคู่

ภาคผนวก ข
โปรแกรมควบคุมการทำงาน

โปรแกรม Visual Basic 6.0 ในส่วนของภาคส่งสัญญาณ

```
Private Sub Command10_Click()
MsgBox "เกี่ยวกับโปรแกรม" & vbCrLf & "1. โปรแกรมนี้จัดทำขึ้นเพื่อใช้ควบคุมการเปิด-ปิด
อุปกรณ์ไฟฟ้า " & vbCrLf & "2. โปรแกรมนี้เขียนขึ้นเมื่อ มิถุนายน 2551" & vbCrLf &
"3. โปรแกรมนี้เขียนโดย นางสาวชมพูนุท ยอดนวล และ นางสาวเปรมวดี ถาวรธนทรัพย์ ",
vbOKOnly, "About"
End Sub

Private Sub Command1_Click()
clearbox
MSComm1.Output = "A"
End Sub


Private Sub Command2_Click(Index As Integer)
clearbox
MSComm1.Output = "B"
End Sub


Private Sub Command3_Click(Index As Integer)
clearbox
MSComm1.Output = "C"
End Sub

Private Sub Command4_Click(Index As Integer)
clearbox
MSComm1.Output = "D"
End Sub
```

```
Private Sub Command5_Click(Index As Integer)
clearbox
MSComm1.Output = "E"
End Sub

Private Sub Command6_Click(Index As Integer)
clearbox
MSComm1.Output = "F"
End Sub

Private Sub Command7_Click(Index As Integer)
clearbox
MSComm1.Output = "G"
End Sub

Private Sub Command8_Click(Index As Integer)
clearbox
MSComm1.Output = "H"
End Sub

Private Sub Command9_Click()
MsgBox "ขั้นตอนการใช้งาน" & vbCrLf & "1.เมื่อเรา Click ที่ Switch จะแสดงสถานะการทำงาน
ขึ้นมา เพื่อที่จะสามารถใช้ในการตรวจสอบสถานะการทำงานของ Switch ที่เราต้องการได้ " &
vbCrLf & "2.เมื่อเรา Click ที่ About จะแสดงรายละเอียดของโปรแกรม" & vbCrLf & "3.หากท่าน
ต้องการทราบรายละเอียดของโปรแกรม กรุณา Click ที่ About", vbOKOnly, "Help"    ; แสดง
ข้อความ

'frmLogin.Show
End Sub
```

```
Private Sub clearbox()
RichTextBox1.Text = ""
End Sub
Private Sub Form_Load()

If LoginSuccess = False Then
frmLogin.Show
Form1.Hide
End If



    MSComm1.CommPort = "1"
    MSComm1.Settings = "9600, n , 8, 1"
    MSComm1.PortOpen = True

    Timer1.Interval = 100
    Timer1.Enabled = True
    RichTextBox1 = ""

End Sub

Private Sub Timer1_Timer()
On Error Resume Next
Const SpeedBaud = 16000
Dim LngSize As Long, X
    MSComm1.DTREnable = False
    MSComm1.DTREnable = True
    MSComm1.InputLen = 0
    Data = MSComm1.Input
```

```
    If Data = "A" Then
    Data = "Switch 1 On"
    End If
    If Data = "B" Then
    Data = "Switch 1 Off"
    End If
    If Data = "C" Then
    Data = "Switch 2 On"
    End If
    If Data = "D" Then
    Data = "Switch 2 Off"
    End If
    If Data = "E" Then
    Data = "Switch 3 On"
    End If
    If Data = "F" Then
    Data = "Switch 3 Off"
    End If
    If Data = "G" Then
    Data = "Switch 4 On"
    End If
    If Data = "H" Then
    Data = "Switch 4 Off"
    End If
    RichTextBox1.SelText = Data
    RichTextBox1.SelStart = Len(RichTextBox1.Text)

End Sub
```

โปรแกรมไมโครคอนโทรลเลอร์ ในส่วนของภาครับสัญญาณ

```
CONTROL1    EQU    P3.4
CONTROL2    EQU    P3.5
CONTROL3    EQU    P3.6
CONTROL4    EQU    P3.7

        ORG    0000H
        JMP    0100H

        ORG    0100H
        MOV    SP,#2FH
        CLR    EA      ; Disable Intr All
        CALL   Initial_Serial
        CALL   Send_ClrScr
        CALL   DELAY_500M

        CLR    CONTROL1
        CLR    CONTROL2
        CLR    CONTROL3
        CLR    CONTROL4
        CALL   Send_ClrScr

Main_LOOP:

LOOP_S1:
        CALL   DELAY_500M
        JNB    RI,$      ;Wait data
        CLR    RI
        MOV    R0,SBUF
```

```
        MOV   A,R0
        clr   c
        CJNE  A,#'B',S1_ON
        CLR   CONTROL1     ;B--->S1_OFF
        CALL  DELAY_500M
        CALL  Send_ClrScr
        MOV   DPTR,#_OFF
        CALL  Send_Table
        JMP   Main_LOOP

S1_ON:  MOV   A,R0
        clr   c
        CJNE  A,#'A',LOOP_S2       ;A---->S1_ON
        SETB  CONTROL1
        CALL  DELAY_500M
        CALL  Send_ClrScr
        MOV   DPTR,#_ON
        CALL  Send_Table
        JMP   Main_LOOP

LOOP_S2:
        MOV   A,R0
        clr   c
        CJNE  A,#'D',S2_ON
        CLR   CONTROL2     ;D--->S2_OFF
        CALL  DELAY_500M
        CALL  Send_ClrScr
        MOV   DPTR,#_OFF1
        CALL  Send_Table
        JMP   Main_LOOP
```

```
S2_ON: MOV   A,R0
       clr   c
       CJNE  A,#'C',LOOP_S3        ;C---->S2_ON
       SETB  CONTROL2
       CALL  DELAY_500M
       CALL  Send_ClrScr
       MOV   DPTR,#_ON1
       CALL  Send_Table
       JMP   Main_LOOP

LOOP_S3:
       MOV   A,R0
       clr   c
       CJNE  A,#'F',S3_ON
       CLR   CONTROL3     ;F--->S3_OFF
       CALL  DELAY_500M
       CALL  Send_ClrScr
       MOV   DPTR,#_OFF2
       CALL  Send_Table
       JMP   Main_LOOP
S3_ON: MOV   A,R0
       clr   c
       CJNE  A,#'E',LOOP_S4
       SETB  CONTROL3
       CALL  DELAY_500M
       CALL  Send_ClrScr
       MOV   DPTR,#_ON2
       CALL  Send_Table
       JMP   Main_LOOP
```

```
LOOP_S4:
        MOV     A,R0
        CPL     A
        ANL     A,#'H'
        JNZ     S4_ON
        ;clr    c
        ;CJNE   A,#'H',S4_ON
        CLR     CONTROL4      ;H--->S4_OFF
        CALL    DELAY_500M
        CALL    Send_ClrScr
        MOV     DPTR,#_OFF3
        CALL    Send_Table
        JMP     Main_LOOP
S4_ON:  MOV     A,R0
        CPL     A
        ANL     A,#'G'
        JNZ     N1
        ;clr    c
        ;CJNE   A,#'G',Main_LOOP
        SETB    CONTROL4
        CALL    DELAY_500M
        CALL    Send_ClrScr
        MOV     DPTR,#_ON3
        CALL    Send_Table
        JMP     Main_LOOP

N1:     JMP     Main_LOOP
        $INCLUDE      "Serial.Sub"

_ON:    DB      'SWITCH 1 ---> ON',00
```

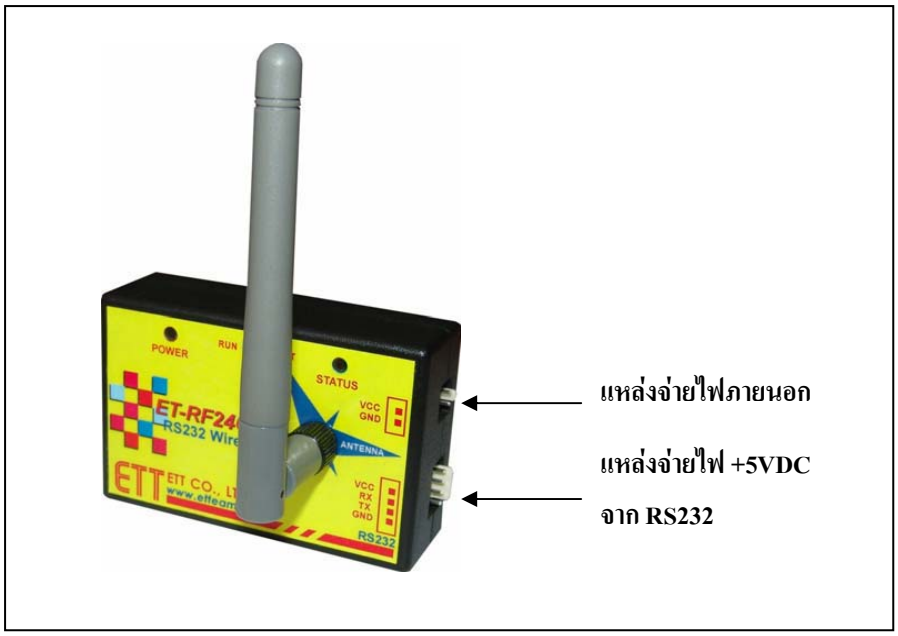
```
_OFF:   DB      'SWITCH 1 ---> OFF',00
_ON1:   DB      'SWITCH 2 ---> ON',00
_OFF1: DB       'SWITCH 2 ---> OFF',00
_ON2:   DB      'SWITCH 3 ---> ON',00
_OFF2: DB       'SWITCH 3 ---> OFF',00
_ON3:   DB      'SWITCH 4 ---> ON',00
_OFF3: DB       'SWITCH 4 ---> OFF',00

        END
```

ภาคผนวก ค
อุปกรณ์รับส่ง Wireless

1. ชุดอุปกรณ์รับส่ง Wireless

1.1 ลักษณะทั่วไป

ET-RF24G V2.0 เป็นชุด Signal Converter สำหรับใช้แปลงสัญญาณระหว่าง RS232 และ RF-Wirelessโดยในโหมดการทำงานของการส่งข้อมูล (Transmitter) จะทำหน้าที่รอรับข้อมูลจากพอร์ตสื่อสารอนุกรม RS232 จากขา RX แล้วแปลงเป็นสัญญาณความถี่ (GFSK) ส่งออกไปในอากาศ และในทางกลับกันในโหมดการทำงานแบบรับ (Receiver) ชุด ET-RF24G V2.0 ก็จะทำหน้าที่คอยตรวจจับข้อมูลที่อยู่ในรูปของสัญญาณความถี่ (GFSK) จากด้าน RF เพื่อแปลงกลับเป็นข้อมูลแบบ RS232 ส่งออกไปทางขา TX ได้ด้วยซึ่งจะเห็นได้ว่าชุดแปลงสัญญาณ ET-RF24G V2.0 นั้น สามารถนำไปต่อใช้งานร่วมกับพอร์ตสื่อสารอนุกรม แบบ RS232 เพื่อใช้งานในลักษณะของการสื่อสารอนุกรมแบบไร้สาย (Wireless Transceiver) ได้โดยตรง โดยจะมีข้อดีกว่า คือ สามารถรับส่งข้อมูลกันได้ในระยะทางที่ไกลกว่า RS232 หลายเท่าตัว และประการสำคัญ คือไม่จำเป็นต้องใช้สายสัญญาณที่เป็นตัวนำสัญญาณทางไฟฟ้าในการสื่อสารข้อมูลกัน ทำให้สามารถเปลี่ยนแปลงหรือเคลื่อนย้ายจุดรับส่งข้อมูลได้ตลอดเวลา ซึ่งถ้าเป็นการรับส่งข้อมูลด้วยระบบ RS232 แบบที่ใช้สายสัญญาณนั้น จะเกิดความยุ่งยากในการติดตั้งสายสัญญาณเป็นอย่างมากแต่อย่างไรก็ตามการรับส่งข้อมูลโดยใช้อากาศเป็นตัวกลางในการสื่อสารนั้น ก็มีข้อจำกัดบางประการเหมือนกัน โดยเฉพาะอย่างยิ่ง เรื่องความน่าเชื่อถือของข้อมูลที่รับส่งกัน ซึ่งมีโอกาสผิดพลาดหรือสูญหายได้เหมือนกัน เนื่องจากในการลำเลียงข้อมูลนั้นไม่ได้ใช้สายสัญญาณเป็นตัวกลางในการรับส่งข้อมูล แต่ใช้อากาศเป็นตัวกลางในการรับส่งข้อมูลแทน ซึ่งมีโอกาศที่ข้อมูลจะเกิดการรบกวนจากสัญญาณอื่นๆที่มีย่านความถี่ใกล้เคียงกันแล้วทำให้ข้อมูลผิดเพี้ยนไปได้บ้างเหมือนกัน ซึ่งระบบการจัดการข้อมูลของเครื่อง ET-RF24G V2.0 นั้น มีระบบการเข้ารหัสและถอดรหัสข้อมูลที่มีความน่าเชื่อถืออยู่ในเกณฑ์ที่จัดว่าดี โดยข้อมูลแต่ละ Byte ที่มีการรับส่งกันนั้น จะมีการตรวจสอบความถูกต้องของข้อมูลให้ด้วยแล้ว โดยข้อมูลที่รับได้จากด้าน RF นั้นรับประกันได้ว่าเป็นข้อมูลที่มีความถูกต้องแน่นอน แต่อย่างไรก็ตามการรับส่งข้อมูลนั้นมีโอกาสผิดพลาดในเรื่องของการสูญหายของข้อมูลบ้างเหมือนกัน เนื่องจากกลไกในการรับส่งข้อมูลของเครื่อง ET-RF24G V2.0 นั้น จะมีการตรวจสอบข้อมูลทุก Byte ที่รับได้จาก RF เสมอ ซึ่งถ้าพบว่ามีความผิดพลาดเกิดขึ้นจะทิ้งข้อมูล Byte นั้นไป ซึ่งผู้ใช้ควรมีกลไกในการตรวจสอบข้อมูลที่รับส่งกันว่าครบถ้วนหรือไม่ด้วย ซึ่งหากพบว่ามีการสูญหายของข้อมูลเกิดขึ้นก็ให้ร้องขอให้มีการส่งข้อมูลนั้นซ้ำนั้นๆใหม่อีกครั้งหนึ่ง ก็จะสามารถแก้ไขปัญหาดังกล่าวได้



รูปที่ ค.1 อุปกรณ์รับ-ส่ง RS232 to RF-Wireless(RF2.4GHz) CONVERTER รุ่น ET-RF24G V2.0

1.2 Power Supply

สำหรับการต่อแหล่งจ่ายไฟให้กับเครื่อง ET-RF24G V2.0 นั้น จะสามารถเลือกต่อแหล่งจ่ายไฟให้กับตัวเครื่องได้ 2 ทางด้วยกัน โดยเครื่อง ET-RF24G V2.0 นั้น ต้องการไฟเลี้ยงวงจร ซึ่งเป็นแหล่งจ่ายกระแสตรง ขนาดประมาณ +5VDC ถึง +9VDC โดยจุดเชื่อมต่อแหล่งจ่ายไฟของเครื่อง ET-RF24G V2.0 นี้ สามารถเชื่อมต่อได้ 2 จุดด้วยกัน โดยผู้ใช้สามารถเลือกต่อแหล่งจ่ายไฟให้กับเครื่อง ET-RF24G V2.0 จุดใดจุดหนึ่งก็ได้ในกรณีที่นำเครื่อง ET-RF24G V2.0 ไปเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์รุ่นต่างๆของ อีทีที นั้นสามารถใช้แหล่งจ่ายไฟจากบอร์ดไมโครคอนโทรลเลอร์ เพื่อจ่ายให้กับตัวเครื่อง ET-RF24G V2.0 ได้ทันที โดยไม่ต้องใช้แหล่งจ่ายไฟจากภายนอก เนื่องจากขั้วต่อสัญญาณ RS232 ของบอร์ดไมโครคอนโทรลเลอร์รุ่นต่างๆ ของบริษัท อีทีที นั้น ได้จัดเตรียมแหล่งจ่ายไฟตรง ขนาด +5V เตรียมไว้ให้ด้วยแล้ว โดยผู้ใช้เพียงแต่นำสายสัญญาณRS232 ซึ่งทำการต่อสายสัญญาณครบทั้ง 4 เส้น แต่สำหรับกรณีที่นำเครื่อง ET-RF24G V2.0 ไปต่อใช้งานกับอุปกรณ์อื่นๆที่ไม่ได้มีการจัดเตรียมจุดต่อไฟเลี้ยงไว้ให้ด้วย ผู้ใช้จำเป็นต้องจัดหา Adapter จ่ายไฟจากภายนอกมาต่อให้กับเครื่อง ET-RF24G V2.0 ต่างหากด้วย โดยให้เลือกแหล่งจ่ายไฟที่มีขนาดแรงดันไฟตรงประมาณ +5VDC และสามารถจ่ายกระแสได้ประมาณ 300mAเป็นอย่างน้อย ซึ่งในกรณีนี้ขอแนะนำให้เลือกใช้ Power Supply รุ่น “ACH-4E” ซึ่งเป็นแหล่งจ่ายไฟแบบSwitching Power ใช้กับไฟบ้าน 220VAC และให้เอ้าพุตเป็นไฟกระแสตรง ขนาดประมาณ 5VDC / 750mA เพราะ Power Supply รุ่นนี้ สามารถใช้งานร่วมกับเครื่อง ET-RF24G

V2.0 ได้อย่างต่อเนื่องเป็นเวลานานๆ โดยไม่เกิดความร้อนสะสมที่วงจร Regulate ของบอร์ด ET-RF24G V2.0 มากนัก ซึ่งถ้าผู้ใช้เลือกใช้แหล่งจ่ายไฟรุ่นอื่นๆ ทีมีขนาดแรงดันสูงกว่า +5V มากๆ ซึ่งถึงแม้ว่าจะสามารถใช้งานร่วมกันกับเครื่อง ET-RF24G V2.0 ได้ แต่ถ้ามีการใช้งานอย่างต่อเนื่องเป็นเวลานานๆแล้ว อาจทำให้เกิดความร้อนสะสมที่ตัวไอซี Regulate มากเกินไป จนอาจทำให้ภาค Power ของเครื่อง ET-RF24G V2.0 หยุดจ่ายไฟทำให้เครื่องหยุดทำงานได้

รูปที่ ค.2 แสดง การต่อ แหล่งจ่ายไฟรุ่น “ACH-4E” จากภายนอกให้กับเครื่อง ET-RF24G V2.0

1.3 โหมดการทำงาน

สำหรับโหมดการทำงานของ ET-RF24G V2.0 นั้นจะแบ่งออกเป็น 2 โหมด ด้วยกัน โดยการกำหนดโหมดการทำงานของ ET-RF24G V2.0 นั้นจะกระทำผ่าน Switch เลือกโหมด ซึ่งอยู่ด้านใต้กล่อง โดยการเลือกโหมดการทำงานนั้นจะต้องกระทำให้เสร็จเรียบร้อยก่อนการจ่ายไฟให้กับ ET-RF24G V2.0 ด้วยเสมอ เนื่องจากการทำงานของเครื่อง ET-RF24G V2.0 นั้นจะทำการตรวจสอบโหมดการทำงานของเครื่องจาก Switch เลือกโหมด เฉพาะในช่วงของการจ่ายไฟเลี้ยงให้เครื่องเริ่มต้นทำงานครั้งแรก (Power-ON) เท่านั้น ซึ่งการเปลี่ยนแปลงตำแหน่งการทำงานของ Switch เลือกโหมด หลังจากทำการจ่ายไฟให้กับ ET-RF24G V2.0 ไปแล้ว จะไม่มีผลต่อการทำงานของเครื่องแต่อย่างใด โดยการทำงานของเครื่อง ET-RF24G V2.0 นั้นจะมี LED แสดงสถานะการทำงานของเครื่องจำนวน 2 หลอด คือ LED POWER ซึ่งเป็น LED สีแดง โดยที่ LED POWER นี้จะติดสว่างให้เห็นตลอดเวลาที่มีการจ่ายไฟเลี้ยงให้เครื่องทำงานอยู่ ส่วน LED อีกดวงหนึ่งนั้นจะเป็น

LED สีเขียว ใช้แสดงสถานะการทำงานของเครื่อง ซึ่งเรียกว่าLED STATUS โดย LED STATUS นี้จะเกิดการกระพริบตามจังหวะของการรับส่งข้อมูลกันในแต่ละครั้งโดยในสภาวะปรกตินั้น ถ้าเครื่องทำงานอยู่ใน RUN MODE หลอด LED STATUS จะดับอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูล แต่ถ้าตัวเครื่องทำงานอยู่ใน SETUP MODE หลอด LED STATUS จะติดอยู่ตลอดเวลาถ้าไม่มีการรับส่งข้อมูลโดยโหมดการทำงานของ ET-RF24G V2.0 จะมีอยู่ด้วยกัน 2 โหมด คือ

1. การใช้งานเครื่อง ET-RF24G V2.0 ใน Run Mode

การใช้งานใน Run Mode ซึ่งเป็นโหมดของการใช้งานตามปรกติของเครื่อง โดยเมื่อเครื่อง ET-RF24G V2.0 เข้าทำงานในโหมดนี้แล้ว จะสังเกตุเห็นหลอดไฟแสดงสถานะของการทำงาน หรือ LED STATUS ดับอยู่ แต่เมื่อมีการ รับ หรือ ส่ง ข้อมูล เกิดขึ้น สถานะการทำงานของ LED STATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูลนั้นๆ แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะดับอยู่ตลอดเวลา

สำหรับการทำงานใน Run Mode นั้น จะแบ่งลักษณะการทำงานออกเป็น 3 แบบด้วยกัน โดยลักษณะการทำงานนี้ จะถูกกำหนดไว้แล้วใน Configuration ของเครื่องใน Setup Mode ดังนั้นก่อนการใช้งานเครื่อง ในครั้งแรกจะต้องทำการกำหนดค่า Configuration ต่างๆให้เรียบร้อยเสียก่อน โดยเมื่อเครื่อง ET-RF24G V2.0 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 V2.0 ในโหมดนี้จะต้องนำสัญญาณ TX(Transmit) ไปต่อกับขาสัญญาณ RX (Receive) ของอุปกรณ์ด้านตรงข้าม (RS232 ของคอมพิวเตอร์ PC) โดยในโหมดนี้ การทำงานของขาสัญญาณ RX ด้าน RS232 ของเครื่อง ET-RF24G V2.0 จะถูกเปลี่ยนหน้าที่เป็นสัญญาณ CTS (Clear To Send) สำหรับใช้ตรวจสอบความพร้อมในการส่งข้อมูลไปให้อุปกรณ์ด้านตรงข้ามแทน ซึ่งในการใช้งานจะต้องนำสัญญาณนี้ไปต่อเข้ากับสัญญาณ RTS (Ready To Send) ของอุปกรณ์ด้านตรงข้าม โดยเครื่อง ET-RF24G V2.0 จะทำการตรวจสอบสถานะของสัญญาณ RX ซึ่งในโหมดนี้เปรียบเสมือน CTS ว่ามีค่าเป็น “0” หรือไม่ โดยถ้าพบว่าเป็น “0” จึงจะส่งข้อมูลออกไปให้ทางขา TX แต่ถ้าพบว่าสถานะของขาสัญญาณนี้มีค่าเป็น “1” แสดงว่าอุปกรณ์ด้านตรงข้าม ยังไม่พร้อมรับข้อมูลก็จะรอจนกว่าจะพบว่าสถานะของสัญญาณ

ดังกล่าวมีค่าเป็น “0” จึงจะส่งข้อมูลออกไปให้ โดยเครื่อง ET-RF24G V2.0 จะสามารถจัดเก็บข้อมูลไว้ใน Buffer เพื่อรอการส่งได้สูงสุด 64 Byte เท่านั้น ซึ่งถ้าในระหว่างที่รอความพร้อมอยู่นั้นมีข้อมูลด้าน RF ส่งเข้ามาเกินกว่า 64 Byte จะทำให้ข้อมูลที่เกินมานั้นสูญหายไป

1.2 การทำงานแบบ RF Transmit Only

เป็นการทำงานแบบทิศทางเดียว โดยการทำงานในโหมดนี้จะมีลักษณะตรงกันข้ามกับ RF Receive Onlyกล่าวคือ เครื่อง ET-RF24G V2.0 จะทำหน้าที่รอรับข้อมูลจากขา RX (Receive) ด้าน RS232 แล้วเปลี่ยนเป็นข้อมูลแบบ GFSK ส่งออกไปทางด้าน RF โดยการใช้งานเครื่องในโหมดนี้จะต้องนำสัญญาณ TX (Transmit) ซึ่งเป็นขาส่งข้อมูลจาก RS232 ของอุปกรณ์ด้านตรงข้ามมาต่อเข้ากับขา RX(Receive) ของเครื่อง ET-RF24G V2.0 ส่วนขาสัญญาณ TX จะถูกเปลี่ยนหน้าที่เป็น RTS (Ready To Send) เพื่อใช้แสดงสถานะความพร้อมในการรับข้อมูลจากด้าน RS232 ซึ่งในการใช้งานจะต้องนำสัญญาณ TX ซึ่งในขณะนี้เปรียบเสมือนกับ RTS นำไปต่อเข้ากับสัญญาณCTS (Clear To Send) ของอุปกรณ์ด้านตรงข้าม เพื่อใช้ในการตรวจสอบความพร้อมในการรับข้อมูล โดยอุปกรณ์ด้านตรงข้ามจะต้องทำการตรวจสอบสถานะของสัญญาณ RTS นี้ เพื่อตรวจสอบความพร้อมในการรับข้อมูลของเครื่อง ET-RF24G V2.0 ด้วย โดยถ้าเครื่อง ET-RF24G V2.0 พร้อมรับข้อมูลจาก RS232 มันจะส่งสัญญาณ RTS ให้มีค่าเป็น “0” รอไว้ และเมื่อใดก็ตามที่การรับข้อมูลทางด้านของ RS232 มีจำนวนข้อมูลที่ยังไม่สามารถเปลี่ยนเป็นGFSK เพื่อส่งออกไปทางด้าน RF ได้ทันจนเกือบจะเต็ม Buffer แล้ว เครื่อง ET-RF24G V2.0 จะทำการส่งสัญญาณRTS ให้มีค่าเป็น “1” ออกไปบอกให้อุปกรณ์ด้านตรงข้ามทราบเพื่อจะได้หยุดการส่งข้อมูลออกมา โดยอุปกรณ์ด้านตรงข้ามจะต้องหยุดการส่งข้อมูลและรอจนกว่าสถานะของสัญญาณ RTS จะกลับเป็น “0” จึงจะเริ่มต้นส่งข้อมูลออกมาใหม่ ซึ่งหลังจากที่เครื่อง ET-RF24G V2.0 ส่งสัญญาณ RTS ด้วยค่า “1” ออกไปแล้ว จะยังคงสามารถรับข้อมูลได้เพิ่มเติมอีกไม่เกิน 16 Byte เท่านั้น ซึ่งถ้าอุปกรณ์ด้านตรงข้ามยังส่งข้อมูลต่อเนื่องมาอีกจนเกินขนาดของBuffer ที่เครื่อง ET-RF24G V2.0 จะรับไว้ได้จะทำให้ข้อมูลที่เกินมานั้นเกิดการสูญหายได้โดยเราสามารถนำเครื่อง ET-RF24G V2.0 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

1.3 การทำงานแบบ RF Auto Direction

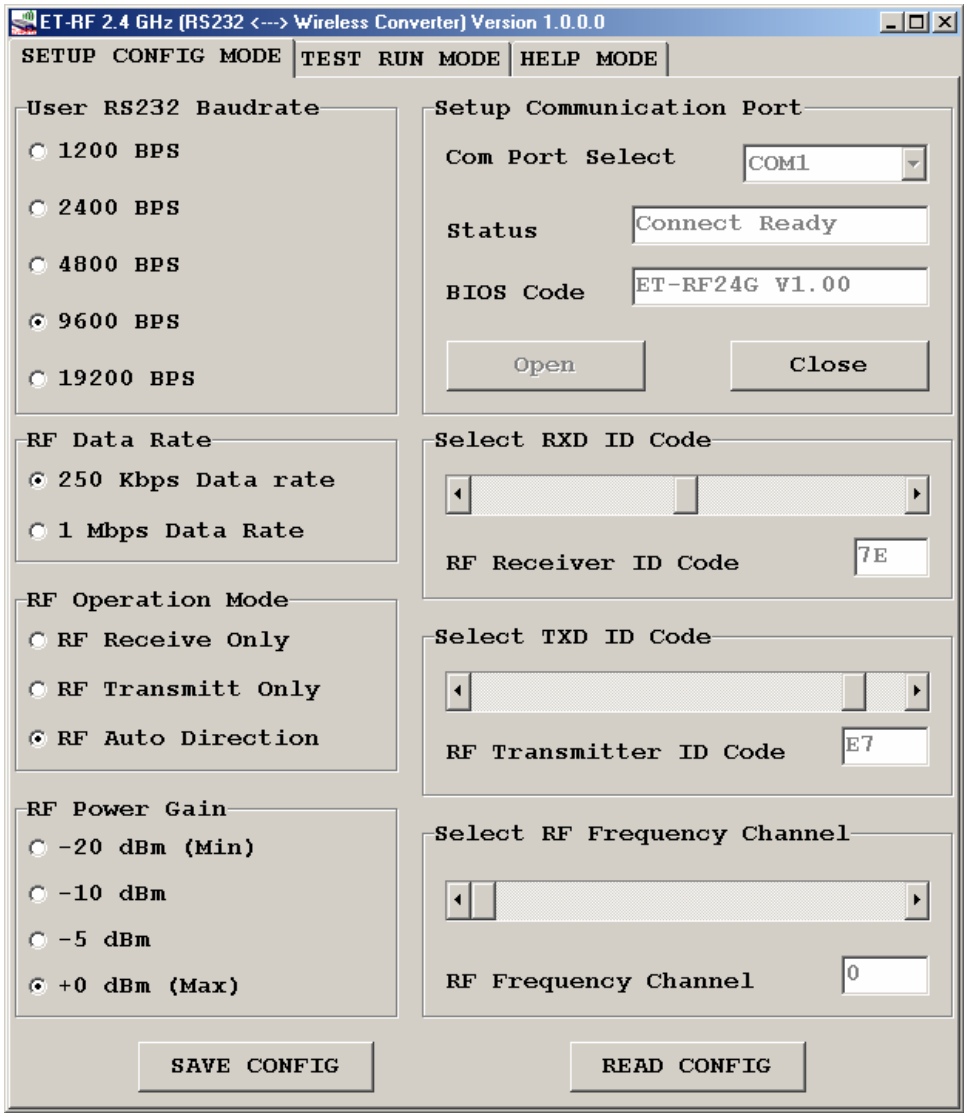
เป็นการทำงานชนิด 2 ทิศทาง แบบ Half Duplex หรือ ผลัดกันรับผลัดกันส่ง ซึ่งสามารถใช้รับส่งข้อมูลระหว่างต้นทาง และ ปลายทาง ได้ โดยใช้เครื่อง ET-RF24G V2.0 ด้านละ 1 ชุด เท่านั้น เพียงแต่การรับส่งข้อมูลแบบนี้จะไม่สามารถส่งข้อมูลสวนทางกันได้เหมือนกับแบบ Full Duplex แต่จะต้องใช้วิธีการผลัดกันรับข้อมูลและส่งข้อมูลแทน โดยเมื่อฝ่ายรับทำการรับข้อมูลได้จนครบแล้วจึงจะสลับหน้าที่เป็นฝ่ายส่งเพื่อส่งข้อมูลย้อนกลับไปโดยในโหมดนี้ เครื่อง ET-RF24G V2.0 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 จากการรอรับข้อมูลให้ทำหน้าที่เป็นตัวส่งข้อมูลแทน เพื่อทำการส่งข้อมูลที่รับได้จาก RS232 ออกไปทาง RF ในทันที ซึ่งหลังจากที่เครื่อง ET-RF24G V2.0 ทำการสลับโหมดการทำงานของอุปกรณ์ด้าน RF จากการรอรับเป็นการส่งและทำการเริ่มต้นส่งข้อมูลออกไปทางด้าน RF เรียบร้อยแล้ว มันจะวนกลับไปตรวจสอบการรับข้อมูลจากด้าน RS232 อีกว่ายังมีข้อมูลส่งเข้ามาอีกหรือไม่ ถ้าพบว่ายังมีข้อมูลส่งเข้ามีอีกก็จะทำการแปลงข้อมูลนั้นเพื่อส่งออกไปยังด้าน RFต่อไปอีกจนกว่าการส่งข้อมูลด้าน RS232 จะสิ้นสุดลง ซึ่งข้อมูลด้าน RS232 ที่ส่งเข้ามานั้น ควรส่งอย่างต่อเนื่อง โดยเมื่อเครื่อง ET-RF24G V2.0 ทำการส่งข้อมูลแต่ละ Byte ออกไปทางด้าน RF เรียบร้อยแล้วมันจะวนรอบรอรับข้อมูล Byte ถัดไปจาก RS232 ภายในเวลา 2.5 mS ถ้าไม่พบข้อมูลส่งเข้ามาอีกภายในระยะเวลาดังกล่าวมันจึงจะทำการเปลี่ยนหน้าที่ของอุปกรณ์ด้าน RF ให้กลับมาทำหน้าที่เป็นการรอรับข้อมูลตามเดิม โดยในขณะที่อุปกรณ์ด้าน RFถูกกำหนดให้เป็นฝ่ายส่งข้อมูลอยู่นั้น จะไม่สามารถทำการรับข้อมูลจาก RF ได้ ซึ่งถ้ามีการส่งข้อมูลเข้ามาในขณะนั้นก็จะไม่สามารถรับได้ โดยค่าเวลาที่จะใช้ในการสลับโหมดการทำงานของ RF จากฝ่ายส่งข้อมูลให้เป็นฝ่ายรับข้อมูลนั้นจะมีค่าเป็น 2.5mS ดังนั้นเมื่อฝ่ายรับสามารถรับข้อมูลได้ครบหมดแล้วก่อนที่จะทำการส่งข้อมูลเพื่อตอบกลับไปยังฝ่ายตรงข้ามนั้น ควรทำการหน่วงเวลาไว้ไม่น้อยกว่า 3mS นับจากรับข้อมูล Byte สุดท้ายได้เรียบร้อยแล้วจึงเริ่มต้นส่งข้อมูล Byte แรกย้อนกลับไป ซึ่งถ้าฝ่ายรับทำการส่งข้อมูลตอบกลับไปยังฝ่ายตรงข้ามเร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทันสำหรับการใช้งานเครื่อง ET-RF24G V2.0 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 อุปกรณ์ที่ต่อไว้

ด้าน RS232 จะพร้อมรับข้อมูลหรือไม่ ซึ่งถ้าด้านRS232 ไม่พร้อมรับข้อมูลก็จะทำให้ข้อมูล Byte นั้นสูญหายไปทันที ซึ่งในการใช้งานนั้น ผู้ใช้ควรกำหนดค่าความเร็วในการรับส่งข้อมูลด้าน RS232 ที่จะใช้กับเครื่อง ET-RF24G V2.0 ทุกๆตัวด้วยค่าความเร็วที่เท่ากันด้วยเพื่อให้การรับและส่งข้อมูลเกิดความสัมพันธ์กันอย่างเหมาะสมสำหรับความสามารถในการรอรับข้อมูลจาก RS232 ของเครื่อง ET-RF24G V2.0 ในโหมดนี้ จะสามารถรับข้อมูลได้อย่างต่อเนื่องสูงสุด ไม่เกิน 64 Byte ดังนั้นในกรณีที่มีการส่งข้อมูลจากด้าน RS232 ด้วยข้อมูลจำนวนมากกว่า 64 Byte ต่อเนื่องกันนั้น ควรทำการแบ่งข้อมูลออกเป็นดๆ โดยให้มีขนาดชุดละไม่เกิน 64 Byte ซึ่งหลังจากทำการส่งข้อมูลอย่างต่อเนื่องไปได้ 1 ชุด (64 Byte) แล้วควรทำการหน่วงเวลาไว้ชั่วขณะหนึ่งอย่างน้อย1mS แล้วจึงเริ่มส่งข้อมูลชุดถัดไป สลับกับการหน่วงเวลา อย่างนี้เรื่อยๆ เพื่อให้เครื่อง ET-RF24G V2.0 สามารถนำข้อมูลที่รับได้จากด้าน RS232 ส่งออกไปทางด้าน RF ได้ทัน ซึ่งถ้าทำการส่งข้อมูลอย่างต่อเนื่องโดยไม่มีการหน่วงเวลาเลยอาจทำให้ข้อมูลบาง Byte เกิดการสูญหายไปได้

2. การใช้งานเครื่อง ET-RF24G V2.0 ใน Setup Mode

การใช้งานเครื่อง ET-RF24G V2.0 ใน Setup Mode ซึ่งเป็นโหมดสำหรับใช้กำหนดค่า Configuration ต่างๆ สำหรับควบคุมการทำงานของเครื่อง ET-RF24G V2.0 ที่จะใช้ในขณะที่เครื่องทำงานอยู่ใน Run Mode โดยในการ Setup ค่า Configuration ต่างๆนั้นจะกระทำร่วมกับโปรแกรม **“ET_RF24G_V1.EXE”** ของ อีทีที ซึ่งเมื่อเครื่อง ET-RF24G V2.0 เข้าทำงานในโหมด Setup แล้วจะสังเกตุเห็นหลอดไฟแสดงสถานะการทำงาน หรือ LEDSTATUS ติดสว่างค้างอยู่ตลอดเวลา แต่เมื่อมีการสั่งอ่านหรือเขียนข้อมูลกับบอร์ด สถานะการทำงานของ LEDSTATUS จึงจะกระพริบตามจังหวะของการรับส่งข้อมูล แต่ถ้ายังไม่มีการรับส่งข้อมูลกัน LED STATUS จะติดค้างอยู่ตลอดเวลา ซึ่งการกำหนดค่า Configuration ให้กับ ET-RF24G V2.0 นั้น จะต้องกระทำในขณะที่ตัวเครื่องทำงานอยู่ใน Setup Mode เท่านั้น (เลือก Switch กำหนดโหมดไว้ทางด้าน Setup แล้วจ่ายไฟให้เครื่องเริ่มต้นทำงาน) โดยค่าของ Configuration ต่างๆนั้นจะถูกใช้สำหรับเป็นเงื่อนไขในการทำงานของ ET-RF24G V2.0 ในขณะที่อยู่ใน Run Mode ดังนั้น ก่อนการเริ่มต้นใช้งานเครื่องในครั้งแรกนั้น จึงจำเป็นอย่างยิ่งที่จะต้องทำการกำหนดค่าของ Configuration ต่างๆให้ถูกต้องและตรงกับความต้องการที่จะใช้งานเสียก่อน โดยเมื่อทำการกำหนดค่าตัวเลือกต่างๆของ Configuration เรียบร้อยแล้ว ก็สามารถเปลี่ยนโหมดการทำงานของตัวเครื่องกลับเป็น Run Mode พร้อมกับการปิดไฟที่จ่ายให้กับตัวเครื่อง (Power-OFF) ชั่วขณะหนึ่ง จากนั้นจึงเริ่มต้นจ่ายไฟให้กับตัวเครื่องใหม่ (Power-ON) ก็สามารถใช้งาน ET-RF24G V2.0 ตามค่าของ Configuration ที่กำหนดไว้แล้วได้ทันที โดยค่าตัวเลือกต่างๆของ Configuration ที่ได้กำหนดไว้แล้วจะถูกเก็บไว้ภายในตัวเครื่องอย่างถาวร ถึงแม้ว่าจะไม่ได้ทำการจ่ายไฟให้กับตัวเครื่องแล้วก็ตาม ดังนั้นเมื่อทำการกำหนดค่า Configuration

ต่างๆเรียบร้อยแล้ว ถ้าไม่มีการเปลียนแปลงเงื่อนไขการทำงานของตัวเครื่องต่างไปจากเงื่อนไขเดิมที่ได้กำหนดไว้แล้ว ก็ไม่จำเป็นต้องทำการกำหนดค่า Configuration ใหม่อีกแต่อย่างใด โดยทุกๆครั้งที่เริ่มต้นจ่ายไฟเข้าเครื่องในครั้งแรกนั้น การทำงานของET-RF24G V2.0 จะเป็นไปตามเงื่อนไขที่กำหนดไว้ใน Configuration เสมอทุก ๆครั้ง โดยคุณสมบัติของConfiguration ต่างๆนั้นมีดังนี้



รูปที่ ค.3 แสดง รูปโปรแกรมที่ใช้สำหรับกำหนดค่า Configuration ของ ET-RF24G V2.0

• **User RS232 Baudrate** ใช้สำหรับกำหนดค่าความเร็วในการรับส่งข้อมูลทางด้าน RS232 ของตัวเครื่อง ในขณะที่ทำงานอยู่ใน Run Mode ซึ่งสามารถกำหนดได้ 5 ค่าคือ

- 1200 BPS
- 2400 BPS
- 4800 BPS
- 9600 BPS
- 19200 BPS

• **RF Data Rate** ใช้สำหรับกำหนดความเร็วในการรับส่งข้อมูลทางด้าน RF ของ ET-RF24G V2.0 ซึ่งจะต้องกำหนดให้เครื่อง ET-RF24G V2.0 ทุกๆตัว ที่จะนำมาใช้ติดต่อสื่อสารกัน มีค่าอัตราความเร็วในการรับส่งข้อมูลด้าน RF หรือ RF Data Rate นี้มีค่าเท่ากันทั้งหมด ซึ่งถ้ากำหนดค่าความเร็วต่างกันจะไม่สามารถรับส่งข้อมูลกันได้ ซึ่งค่าอัตราความเร็วในการส่งข้อมูลนี้จะมีผลต่อระยะทางการรับส่งข้อมูลด้วย ซึ่งถ้าใช้ความเร็วในการส่งสูง (1Mbps) จะทำให้รัศมีการรับส่งข้อมูลได้ระยะทางสั้นลง แต่ถ้าใช้ความเร็วในการรับส่งข้อมูลที่ช้าลง (250Kbps) จะทำให้ได้รัศมีการรับส่งไกลขึ้น โดยค่า RF DataRate สามารถกำหนดได้ 2 ค่า คือ

- 250 Kbps
- 1 Mbps

• **RF Operation Mode** ใช้สำหรับกำหนดโหมดการทำงานของ ET-RF24G V2.0 ซึ่งสามารถกำหนดหน้าที่การทำงานได้ 3 แบบ ด้วยกันคือ- RF Receive Only เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้านRF เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 และส่งออกไปทางด้านขา TX ของ RS232 ตลอดเวลา

- RF Transmit Only เป็นการกำหนดให้ ET-RF24G V2.0 ทำหน้าที่เป็นฝ่ายรอรับข้อมูลทางด้าน RS232 จากขา RX เพื่อเปลี่ยนเป็นข้อมูลแบบ GFSK และส่งออกไปทางด้าน RF ตลอดเวลา

- RF Auto Direction เป็นการกำหนดโหมดการทำงานแบบ Half Duplex 2 ทิศทาง ซึ่งสามารถสลับโหมดการทำงานระหว่างการรับและส่งข้อมูลได้เองโดยอัตโนมัติ โดยในโหมดการทำงานนี้ เครื่อง ET-RF24G V2.0 จะรอตรวจสอบข้อมูลทั้งจากด้าน RS232 และด้าน RF อยู่ตลอดเวลา โดยถ้าได้รับข้อมูลจากด้าน RS232 ก็จะทำการแปลงแล้วส่งออกทางด้าน RFจากนั้นก็จะกำหนดให้ด้าน RF กลับมาเป็นฝ่ายรอรับข้อมูลตามเดิม และเมื่อได้รับข้อมูลจากด้าน RF ก็จะแปลงเป็นข้อมูลแล้วส่งออกไปทางด้าน RS232 โดยอัตโนมัติ

• **RF Power Gain** เป็นการกำหนดกำลังส่งของวงจร RF Power ที่ใช้ในการส่งข้อมูล โดยค่า +0dBm เป็นค่ากำลังส่งสูงสุด ส่วน –20dBm เป็นค่ากำลังส่งต่ำสุด โดยสามารถกำหนดได้ 4 ระดับ คือ

- -20dBm (กำลังส่งต่ำสุด)
- -10dBm
- -5dBm
- +0dBm (กำลังส่งสูงสุด)

• **RXD ID Code** เป็นรหัส ID Code ของเครื่อง ET-RF24G V2.0 ในโหมดของการรับข้อมูลจาก RF โดยเมื่อเครื่อง ET-RF24G V2.0 ด้านส่งจะทำการส่งข้อมูลออกไปทาง RF นั้นจะมีการระบุหมายเลข ID Code ของด้านรับรวมไปกับชุดข้อมูลด้วยเสมอ โดยเมื่อเครื่อง ET-RF24G V2.0 ที่อยู่ทางด้านรับทำการรับข้อมูลจากด้าน RF ได้ อันดับแรกมันจะทำการเปรียบเทียบรหัส ID Code ที่รวมมากับข้อมูลที่รับมาได้ว่าตรงกับรหัสของ RXD ID Code ที่กำหนดไว้ในตัวมันหรือไม่ ซึ่งถ้าถูกต้องก็จะแยกเอาเฉพาะส่วนของข้อมูลที่รับเข้ามาได้เพื่อเปลี่ยนเป็นข้อมูลแบบ RS232 แล้วส่งออกไปทางด้าน TX ของRS232 แต่ถ้ารหัส ID Code ที่รับมาได้ไม่ตรงกับรหัส RXD ID Code ที่กำหนดไว้ เครื่อง ET-RF24GV2.0 จะทิ้งข้อมูลชุดนั้นไปทันที โดยค่า RXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **TXD ID Code** เป็นรหัส ID Code ปลายทางที่จะส่งข้อมูลไปหา โดยที่เครื่อง ET-RF24G V2.0 ที่ถูกกำหนดให้ทำหน้าที่เป็นฝ่ายส่งข้อมูลนั้น เมื่อมันสามารถรับข้อมูลจาก RS232 ได้แล้ว มันจะทำการนำเอาข้อมูลนั้นไปเข้ารหัสรวมกับ TXD ID Code ที่กำหนดไว้ แล้วส่งออกไปทางด้าน RF โดยรหัสของ TXD ID Code นี้หมายถึง รหัส RXD ID Code ของฝ่ายรับที่ต้องการส่งข้อมูลไปหานั่นเอง โดยค่าTXD ID Code นั้นสามารถกำหนดได้ 256 ค่าในรูปแบบของเลขฐานสิบหก (00H-FFH)

• **RF Frequency Channel** เป็นการกำหนดค่าของช่องความถี่ที่จะใช้ในการรับส่งข้อมูลกัน โดยสามารถเลือกกำหนดช่องความถี่ได้สูงสุดมากถึง 125 ช่อง (0-124) โดยการที่เครื่อง ET-RF24G V2.0 จะทำการรับส่งข้อมูลกันได้นั้นจะต้องกำหนดช่องความถี่ที่ตรงกัน และ ใช้อัตราความเร็ว RF Data Rate ที่เท่ากันด้วย ซึ่งที่สามารถเลือกกำหนดช่องความถี่ RF Frequency Channel ได้นั้น จะมีประโยชน์เป็นอย่างมากในกรณีที่มีการใช้งานเครื่อง ET-RF24G V2.0 จำนวนหลายๆกลุ่ม ในบริเวณพื้นที่ใกล้เคียงกันโดยให้กำหนดช่องความถี่ของ ET-RF24G V2.0 กลุ่มที่จะสื่อสารข้อมูลร่วมกันไว้ที่ช่องความถี่เดียวกันส่วนกลุ่มอื่นๆก็ให้เลือกกำหนดช่องความถี่ที่แตกต่างกันออกไป เพื่อลดปัญหาการรบกวนกันการกำหนดค่า Configuration ให้กับเครื่อง ET-RF24G V2.0 นั้นสามารถเลือกกำหนดได้ตามความต้องการและจุดประสงค์ของการใช้งาน โดยแต่ละโหมดของการ

ใช้งานนั้นจะมีค่า Configuration ที่เหมาะสมต่างกัน ซึ่งขอแนะนำวิธีการกำหนดค่า Configuration ดังแนวทางต่อไปนี้

- ความเร็วในการรับส่งข้อมูลด้าน RS232 หรือ User RS232 Baudrate ที่ความเร็ว 19200 Bps นั้น เหมาะกับการใช้งาน ET-RF24G V2.0 แบบ Receive Only หรือ Transmit Only ซึ่งมีการตรวจสอบความพร้อมของสัญญาณในการรับส่งข้อมูลกันด้วย แต่ถ้าต้องการใช้งานเครื่อง ET-RF24G V2.0 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 V2.0 จำนวนหลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ก็อาจทำการลดกำลังส่งให้ต่ำลงเพื่อลดปัญหาการรบกวนกันหรือกำหนดช่องความถี่ RF Frequency Channel ให้ห่างกันมากๆ
- ในกรณีที่มีการใช้เครื่อง ET-RF24G V2.0 หลายๆกลุ่มในพื้นที่ใกล้เคียงกัน ควรกำหนดช่องความถี่ในการใช้งาน หรือ RF Frequency Channel ให้ห่างกันด้วยเพื่อป้องกันการรบกวนกัน
- การใช้งานเครื่อง ET-RF24G V2.0 แบบ Auto Direction นั้น ถ้ามีการส่งข้อมูลจำนวนมากๆ ควรจัดแบ่งข้อมูลออกเป็นชุดๆ โดยให้มีขนาดข้อมูลชุดละไม่เกิน 64 Byte โดยในการส่งข้อมูลแต่ละชุดนั้นให้ทำการส่งข้อมูลอย่างต่อเนื่องโดยให้ข้อมูลแต่ละ Byte มีระยะเวลาห่างกันไม่เกิน 2.5mS เนื่องจากถ้าข้อมูลขาดหายไปนานกว่านี้ เครื่อง ET-RF24G V2.0 จะทำการเปลี่ยนโหมดของการส่งข้อมูลกลับเป็นโหมดของการรับข้อมูลแทน ซึ่งเมื่อมีการส่งข้อมูล Byte ถัดไปมาอีกก็จะต้องเสียเวลาในการสลับโหมดจากฝ่ายรอรับข้อมูลให้เป็นฝ่ายส่งข้อมูลอีก ซึ่งจะทำให้ประสิทธิภาพในการจัดส่งข้อมูลลดลงเนื่องจากต้องเสียเวลาในการสลับโหมดการทำงานของวงจรภาค RF อยู่ตลอดเวลา โดยที่เมื่อทำการจัดส่งข้อมูลครบ 64 Byte แล้ว ให้ทำการหน่วงเวลาไว้ชั่วขณะหนึ่งประมาณ 1mS-2mS แล้วจึงส่งข้อมูลชุดถัดไปอีกอย่างนี้เรื่อยๆ จะทำให้การรับส่งข้อมูลมีประสิทธิภาพสูงสุด
- การใช้งานเครื่อง ET-RF24G V2.0 แบบ Auto Direction นั้น ควรหน่วงเวลาในการสลับโหมดจากฝ่ายของการรอรับข้อมูลเป็นฝ่ายส่งข้อมูล อย่างน้อยที่สุด 3mS – 5mS ซึ่งถ้าส่งข้อมูลย้อนกลับด้วยเวลาที่เร็วกว่านี้อาจทำให้ฝ่ายตรงข้ามไม่สามารถรับข้อมูล Byte แรกได้ทัน

ตัวอย่างการใช้งาน

สำหรับตัวอย่างการใช้งานนั้น จะขอแสดงให้เห็นโดยใช้คอมพิวเตอร์ PC เป็นอุปกรณ์การทดลอง โดยในที่นี้จะขอเลือกใช้โปรแกรมสำเร็จรูปสำหรับใช้ในการสื่อสารของ Windows ซึ่งก็คือ Hyper Terminal โดยใน 2 ตัวอย่างแรกนั้นจะใช้งานกับเครื่อง ET-RF24G V2.0 ในโหมด Auto Direction ซึ่งมีวิธีการใช้งานดังต่อไปนี้

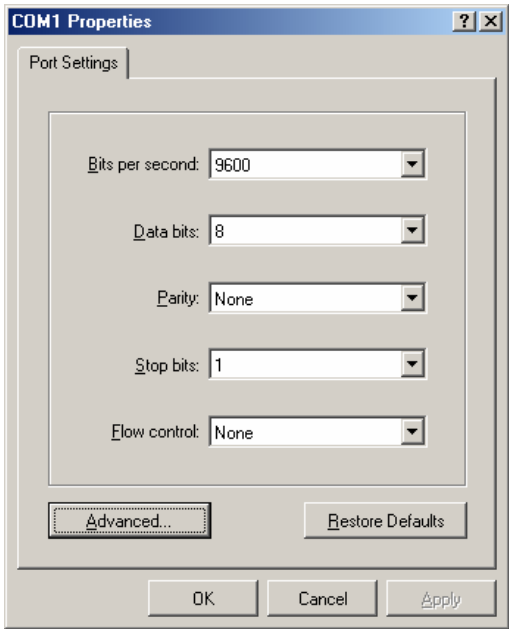
1. เรียกใช้โปรแกรม Hyper Terminal ของ Windows โดยเรียกจาก Start ⟶ Programs ⟶Accessories⟶Communications ⟶ Hyper Terminal ซึ่งจะได้ผลดังรูปที่ ค.4

รูปที่ ค.4

2. ให้เลือกกำหนดชื่อสำหรับใช้ในการเชื่อมต่อ ซึ่งสามารถกำหนดได้เองตามต้องการ โดยในตัวอย่างจะกำหนดเป็น DirectCom1จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป

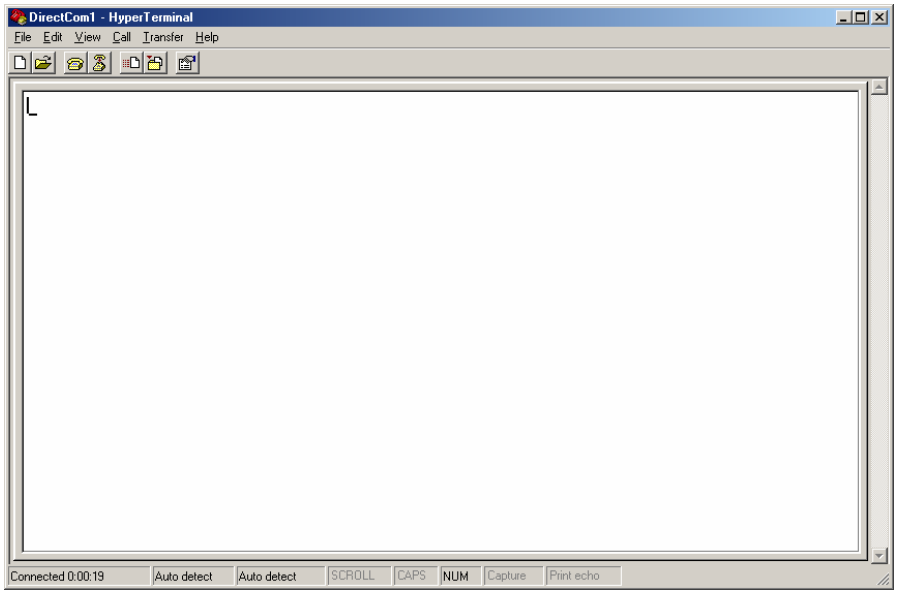
รูปที่ ค.5

3. ให้เลือกกำหนดการเชื่อมต่อเป็น Direct to Com1 ซึ่งถ้าเครื่องคอมพิวเตอร์ที่ใช้เป็น Comport อื่นที่ไม่ใช่Com1 ก็ให้เลือกให้ตรงกับความเป็นจริง จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป



รูปที่ ค.6

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 ดังรูปที่ ค.7



รูปที่ ค.7

ตัวอย่างที่ 1 การรับส่งข้อมูล 2 ทิศทาง (Half Duplex) แบบ จุดต่อจุด (Point-to-Point)

สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบ RS232 จำนวน 2 ชุดโดยต้องใช้รูปแบบการสื่อสารแบบ Half Duplex หรือ ผลัดกันรับ ผลัดกันส่ง กล่าวคือ ด้านรับจะต้องทำการรอรับข้อมูลจากด้านส่งจนครบทั้งหมด แล้วจึงจะส่งข้อมูลตอบกลับไปได้ ซึ่งจะไม่สามารถส่งข้อมูลสวนทางกลับไปในขณะที่กำลังรับข้อมูลอยู่ได้ โดยการสื่อสารแบบนี้ฝ่ายรับข้อมูลจะต้องรอให้รับข้อมูลได้ครบทั้งหมดเสียก่อน จากนั้นจึงจะส่งข้อมูลตอบกลับไปได้ โดยให้กำหนดค่า Configuration ของตัวเครื่อง ET-RF24G V2.0 เป็นดังนี้

ค่า Configuration	ET-RF24G V2.0 ตัวที่1	ET-RF24G V2.0 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	0 1	0 2
TXD ID Code	0 2	0 1
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 โปรแกรมHyper Terminal อยู่ โดยจะสังเกตุเห็นตัวอักษรจากแป้นพิมพ์ของฝ่ายที่เป็นฝ่ายส่งข้อมูล จะถูกส่งออกไปแสดงผลที่หน้าจอโปรแกรม Hyper Terminal ของอีกฝ่ายหนึ่งในทันที

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

สำหรับตัวอย่างนี้จะเป็นการรับส่งข้อมูลระหว่างอุปกรณ์ที่มีการสื่อสารอนุกรมแบบ RS232 จำนวน หลายๆตัวร่วมกัน 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 V2.0 เป็นดังนี้

ค่า Configuration	ET-RF24G V2.0 ตัวที่1(Master)	ET-RF24G V2.0 ตัวที่2 (Slave1)	ET-RF24G V2.0 ตัวที่3 (Slave2)	ET-RF24G V2.0 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	0 1	0 2	0 2	0 2
TXD ID Code	0 2	0 1	0 1	0 1
RF Frequency Channel	0	0	0	0

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 ตัวอย่างเช่นกำหนดให้ใช้รหัส เครื่องหมาย ‘*’ เป็นรหัสเริ่มต้นของชุดข้อมูล ตามด้วยรหัสหมายเลข ID Code ของ Slaveปลายทางเป็นตัวเลข 2 หลัก และจบด้วยรหัส Enter ดังนั้น ในการส่งข้อมูลแต่ละครั้ง Master จะต้องทำการส่งข้อมูลขนาด 4 Byte เสมอ และทางด้าน Slave ก็จะต้องรอรับข้อมูล โดยจะรอรับรหัสเครื่องหมาย ‘*’ เป็นอันดับแรก ซึ่งเมื่อรับรหัสเครื่องหมาย ‘*’ ได้แล้วจึงรอรับข้อมูลถัดไปอีก 2 Byte จากนั้นจึงรอรับข้อมูล Byte ที่ 4 ซึ่งจะต้องตรวจสอบว่าเท่ากับรหัส Enter หรือไม่ ซึ่งถ้าใช่ก็แสดงว่ารับข้อมูลได้ถูกต้อง จากนั้นจึงทำการตรวจสอบข้อมูลในByte ที่ 2 และ 3 ว่าตรงกับรหัส ID Code ของตัวเองหรือไม่ โดย Slave-1 จะมีรหัสเป็น ‘0’,’1’ ส่วน Slave-2 และSlave-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 ทุกตัวจะ

สามารถรับข้อมูลได้เหมือนกันทั้งหมด ซึ่ง Slave ทุกตัวจะต้องทำการวิเคราะห์ข้อมูลที่รับได้ ซึ่งในที่นี้ Slave-1 จะต้องตอบกลับด้วยข้อมูล '*','0','1','O','K',Enter เป็นต้นซึ่งจากตัวอย่างที่ได้กล่าวอธิบายมานี้ เป็นเพียงตัวอย่างแนวทางขั้นต้น เท่านั้น ซึ่งในการนำไปใช้งานจริงนั้นจะต้องมีการดัดแปลงและเพิ่มเติมข้อกำหนดต่างๆเข้าไปในชุดข้อมูลอีก เช่น รหัสคำสั่ง รหัสตรวจสอบความถูกต้องของข้อมูล (Checksum) เป็นต้น ซึ่งข้อกำหนดต่างๆเหล่านี้ ผู้ใช้สามารถ คิดค้น ออกแบบรูปแบบของข้อมูลและคำสั่งต่างๆขึ้นมาใช้งานได้เองตามต้องการ

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

ตัวอย่างที่ 3 การรับส่งข้อมูลแบบ Full Duplex ด้วยโปรแกรมสำเร็จรูป

ในตัวอย่างนี้จะเป็นการประยุกต์ใช้งานเครื่อง ET-RF24G V2.0 สำหรับทำการรับส่งข้อมูลกันแบบ Full Duplex โดยกำหนดโหมดการใช้งานเป็น RF Receive Only และ RF Transmit Only ฝ่ายละ 1 ชุด

ค่า Configuration	**ET-RF24G V2.0 ฝ่ายต้นทาง**		**ET-RF24G V2.0 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	0 1	-	0 2	-
TXD ID Code	-	0 2	-	0 1
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

สำหรับการทดสอบการใช้งาน ตามตัวอย่างนี้ สามารถเรียกใช้โปรแกรมสำเร็จรูปที่จัดการเรื่องการสื่อสารอนุกรมของ Windows ซึ่งก็คือ Hyper Terminal ได้ทันที โดยในการใช้งานนั้นสามารถกระทำได้ดังขั้นตอนต่อไปนี้

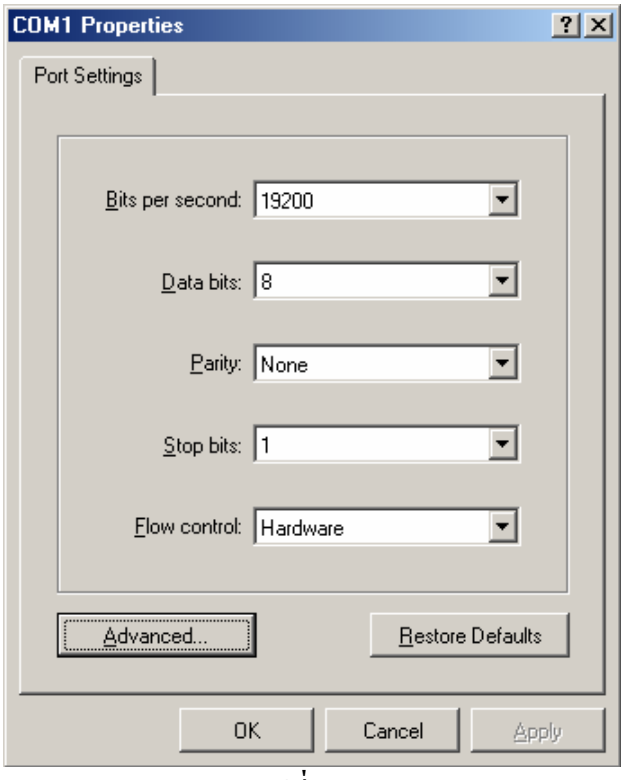
1. เรียกใช้โปรแกรม Hyper Terminal ของ Windows โดยเรียกจาก Start ⟶ Programs ⟶Accessories⟶Communications ⟶ Hyper Terminal ซึ่งจะได้ผลดังรูปที่ ค.8

รูปที่ ค.8

2. ให้เลือกกำหนดชื่อสำหรับใช้ในการเชื่อมต่อ ซึ่งสามารถกำหนดได้เองตามต้องการ โดยในตัวอย่างจะกำหนดเป็นDirectCom1จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป

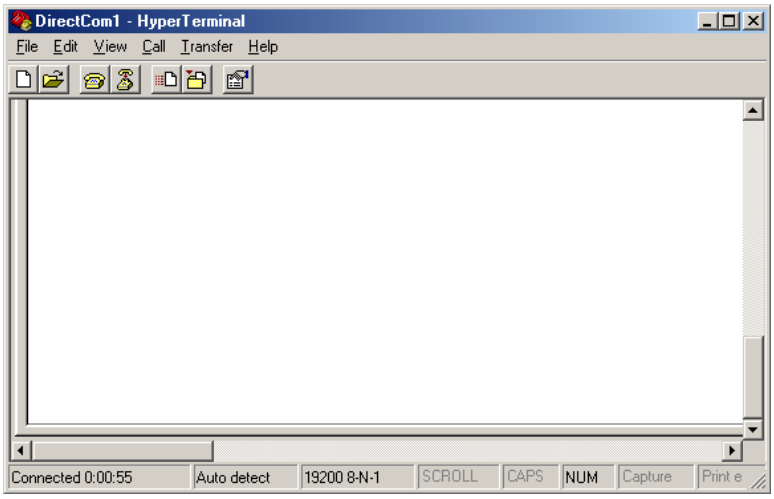
รูปที่ ค.9

3. ให้เลือกกำหนดการเชื่อมต่อเป็น Direct to Com1 ซึ่งถ้าเครื่องคอมพิวเตอร์ที่ใช้เป็น Comport อื่นที่ไม่ใช่ Com1ก็ให้เลือกให้ตรงกับความเป็นจริง จากนั้นให้เลือก OK เพื่อข้ามไปยังขั้นตอนถัดไป



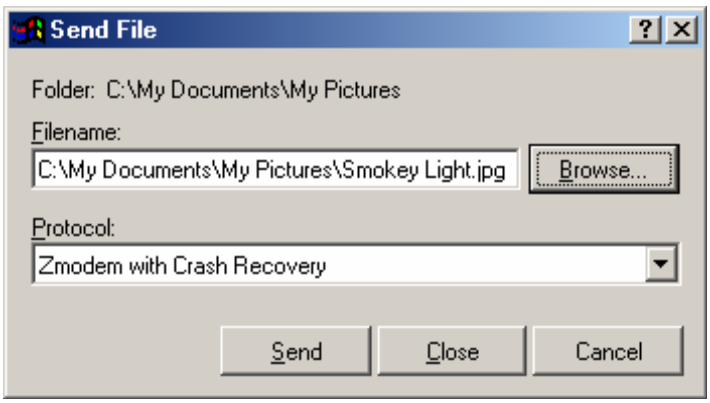
รูปที่ ค.10

4. ในขั้นตอนนี้ จะใช้สำหรับกำหนดคุณสมบัติของพอร์ตอนุกรม RS232 โดยให้เลือก Bit per second =19200 ,Data Bit = 8 ,Parity = None ,Stop Bit=1 ส่วน Flow Control ให้เลือกเป็น Hardware จากนั้นเลือก OKซึ่งจะเข้าสู่หน้าต่างโปรแกรมหลักของ Hyper Terminal ดังรูปที่ ค.11



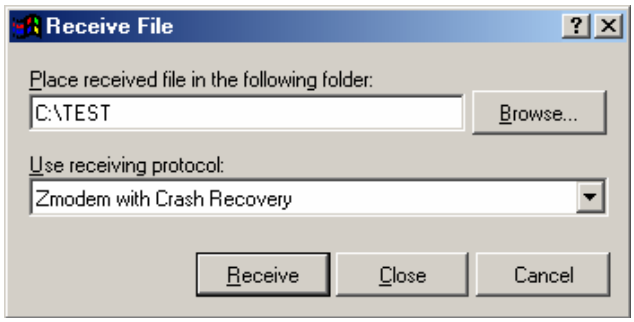
รูปที่ ค.11

5. ในขั้นตอนนี้สามารถทำการรับส่งข้อมูลระหว่างทั้ง 2 ฝ่ายได้แล้ว ซึ่งสามารถทดสอบได้โดยการกดคีย์ใดๆจากฝ่ายหนึ่ง ซึ่งตัวอักขระบนคีย์นั้นๆจะถูกส่งไปแสดงผลยังฝั่งตรงข้ามทันที แต่ในที่นี้เราจะทำการทดสอบการรับและส่งไฟล์ โดยใช้ Protocol สำเร็จรูปของ Hyper Terminal ซึ่งมีให้เลือกใช้มากมากหลาย Protocol โดยต้องกำหนดProtocol ให้ตรงกันทั้งฝ่ายส่งและฝ่ายรับ ซึ่งในขั้นตอนของการทดสอบนั้นต้องกำหนดให้ฝ่ายหนึ่งเป็นฝ่ายรับและให้อีกฝ่ายหนึ่งเป็นฝ่ายส่ง 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 เพื่อทำการเริ่มต้นส่งข้อมูลดังรูปที่ ค.12



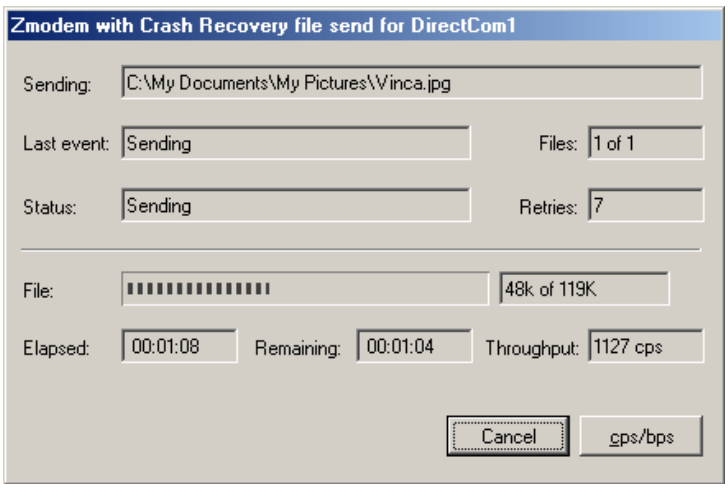
รูปที่ ค.12

สำหรับในด้านที่เป็นฝ่ายรับข้อมูลนั้นก็ให้เลือกกำหนดการทำงานให้เป็นฝ่ายรับ โดยกำหนดจากเมนูคำสั่งของ Transfer ⟶ Receive File… จากนั้นให้เลือกกำหนดตำแหน่งของ Folder สำหรับใช้บันทึกไฟล์ที่รับได้จากฝ่ายส่ง โดยการเลือกจากปุ่ม Browse แล้วเลือกกำหนด Folder ที่ต้องการ ส่วนชื่อนั้นไม่ต้องกำหนด โดยโปรแกรมHyper Terminal จะตั้งให้เองตามชื่อไฟล์จริงที่ส่งมา และในส่วนของ Protocol ที่ใช้นั้นก็ต้องกำหนดให้ตรงกับทางด้านส่ง คือ Zmodem with Crash Recovery จากนั้นให้เลือก Receive เพื่อให้โปรแกรมรอรับไฟล์จากด้านส่ง



รูปที่ ค.13

โดยในขณะที่มีการรับส่งข้อมูลกันอยู่นั้น โปรแกรม Hyper Terminal ทั้ง 2 ด้านจะแสดงสถานะการทำงานให้ทราบอยู่ตลอดเวลา ดังรูปที่ ค.14



รูปที่ ค.14

โดยให้รอจนกว่าการทำงานจะเสร็จสมบูรณ์ ซึ่งหน้าต่างที่แสดงสถานะการทำงานของโปรแกรมจะถูกปิดไปเองโดยอัตโนมัติหลังจากทำการรับส่งข้อมูลกันเสร็จเรียบร้อยแล้วโดยในการทดสอบการทำงานของโปรแกรมตามตัวอย่างนี้ จะต้องกำหนดรูปแบบการสื่อสารของ RS232 ให้มีการตรวจสอบความพร้อมในการรับส่งข้อมูลกันด้วยสัญญาณทาง Hardware ด้วย โดยเลือกกำหนดรูปแบบการสื่อสารของ RS232 ในหัวข้อ Flow Control เป็น Hardware พร้อมกับต่อสายสัญญาณ

ประวัติผู้จัดทำ

นางสาวชมพูนุท ยอดนวล เกิดเมื่อวันที่ 14 กุมภาพันธ์ พ.ศ.2529 ภูมิลำเนาอยู่ที่บ้านเลขที่ 3/1 หมู่ 1ตำบลเมืองบางยม อำเภอสวรรคโลก จังหวัดสุโขทัย สำเร็จการศึกษาระดับประถมศึกษาที่โรงเรียนสวรรค์อนันต์วิทยา อำเภอศรีสำโรง จังหวัดสุโขทัย สำเร็จการศึกษาระดับมัธยมศึกษาที่โรงเรียนสวรรค์อนันต์วิทยา อำเภอสวรรคโลก จังหวัดสุโขทัย ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ตำบลสุรนารี อำเภอเมือง จังหวัดนครราชสีมา

นางสาวเปรมวดี ถาวรธนทรัพย์ เกิดเมื่อวันที่ 9 กรกฎาคม พ.ศ.2529 ภูมิลำเนาอยู่ที่บ้านเลขที่ 61/1 หมู่ 4 ตำบลท่าคอย อำเภอท่ายาง จังหวัดเพชรบุรี สำเร็จการศึกษาระดับประถมศึกษาที่โรงเรียนอรุณประดิษฐ์ อำเภอเมือง จังหวัดเพชรบุรี สำเร็จการศึกษาระดับมัธยมศึกษาตอนต้นที่โรงเรียนอรุณประดิษฐ์ อำเภอเมือง จังหวัดเพชรบุรี สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลายที่โรงเรียนพรหมานุสรณ์จังหวัดเพชรบุรี อำเภอเมือง จังหวัดเพชรบุรี ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี ตำบลสุรนารี อำเภอเมือง จังหวัดนครราชสีมา

บรรณานุกรม

1. Visual Basic Version 6.0 ฉบับสมบูรณ์, ธานิน สิทธิธรรมชารี, 13 มกราคม 2551, บริษัท ซัคเซส มีเดีย จำกัด

2. Advanced Visual Basic ควบคุมอุปกรณ์ฮาร์ดแวร์, ชัยวัฒน์ ทวีจันทร์, 1 มีนาคม 2551, บริษัท ซัคเซส มีเดีย จำกัด

3. http://www.cpe.ku.ac.th/

4. http://www.thaiio.com

5. http://www.ett.co.th

6. http://www.smartdata.com.au/8051/serial.txt

7. http://www.lkh.ac.th/kuanjit/manu_vb.htm

8. http://www.xploiter.com/mirrors/vbtu/

9. http://www.vbhelp.net/

10. http://searchvb.techtarget.com/

11. http://planet-source-code.com/

12. http://www.vb-helper.com/

13. http://www.vbwrie.com/