การสั่งงานเปิด-ปิดอุปกรณ์ไฟฟ้าแบบไร้สายผ่านทางระบบวิทยุFM

นายยุทธชัย ทูลพันธ์

มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตร์บัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี

CONTRIBUTION

A Wireless Web Remote Control For On-Off Electric Devices Using FM Radio System

Mr. YUTTHACHAI TOONPUN

มหาวิทยาลัยเทคโนโลยีสุรนารี

A Project Submitted in Partial Fulfillment of the Requirements for the Degree of Engineer in Telecommunication Engineering

Suranaree University of Technology

Academic year 2001

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

บทที่ 1 บทนำ	หน้า
1.1 กล่าวนำ	1
1.2 วัตถุประสงค์และเป้าหมายของโครงงาน	1
1.3 เค้าโครงของงาน	1
1.4 ขอบเขตของงาน	2
1.5 ความสำคัญหรือประโยชน์ที่คาดว่าจะได้รับ	2
1.6 การจัดรูปเล่มรายงาน	2
บทที่ 2 อธิบายส่วนต่างๆ	
2.1 อุปกรณ์อิเล็คทรอนิคส์ที่สำคัญ	3
2.2 โปรแกรมที่ใช้	3
2.3 ทำไมต้องใช้ไมโครคอนโทรลเลอร์	3
2.4 ทำไมต้องใช้ Port อนุกรม	3
บทที่ 3 ทฤษฎีและหลักการทำงานของโปรแกรม	
3.1 กล่าวนำและทฤษฎี	4
3.2 การรับส่งข้อมูลแบบ Serial ด้วยไอซี MCS-51	5
บทที่ 4 ทฤษฎีและหลักการทำงานของ Hardware	
4.1 ทฤษฎีเกี่ยวกับการรับ-ส่งแบบ FM	7
4.2 RS232	17
4.3 รูปแบบการส่ง-รับข้อมูลจากคอมพิวเตอร์กับไอซี AT89S8252 แบบ RS232	18
4.4 Data sheet ของไอซี AT89S8252 และลักษณะการทำงาน	21
4.5 Data sheet ของไอซี MT8888C และ MT8870 และลักษณะการทำงาน	22
4.6 การออกอากาศ	24
บทที่ 5 การใช้งาน	
5.1 การต่ออุปกรณ์	26
5.2 การใช้โปรแกรมเพื่อเริ่มทำงาน	26
5.3 การใช้โปรแกรมเพื่อสั่งงานผ่านระบบอินเทอร์เน็ต	27
บทที่ 6 บทสรุป	
6.1 สรุป	30
6.2 ปัญหาและอุปสรรค	30
6.3 ข้อเสนอแนะ	31

มหาวิทยาลัยเทคโนโลยีสุรนารี

6.4 แนวทางการพัฒนาในอนาคต 31

หนังสืออ้างอิง 32

ภาคผนวก

1.โปรแกรม VISUAL C++ ใช้สั่งงานผ่าน Computer 34

2. วิธีCompile โปรแกรม Microcontroller 37

3. ข้อมูลใน Fileที่ใช้ Compile 38

4. โปรแกรม microcontroller 39

5. โปรแกรมไมโครคอนโทรลเลอร์ที่ใช้ทดสอบต่างๆในการทำงาน 43

6. วงจรตัวรับและตัวส่ง 48

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1
บทนำ

1.1 กล่าวนำ

เนื่องจากเทคโนโลยีทางด้านการสื่อสารโทรคมนาคมเข้ามามีความสำคัญในการดำรงชีวิตอยู่ของมนุษย์ไม่ว่าจะเป็นทางด้านธุรกิจ การศึกษา ส่วนราชการ ฯลฯ

เนื่องจากการทำงานของเครื่องของเครื่องเปิด-ปิดอุปกรณ์ไฟฟ้าด้วยอินเตอร์เนตเดิมเป็นแบบต่อกับอุปกรณ์ แต่ที่พัฒนาใหม่นี้เป็นแบบไร้สายซึ่งสามารถสั่งเปิด-ปิดเครื่องใช้ไฟฟ้าได้ไกลขึ้นและได้ใช้การส่งข้อมูลแบบอนุกรมจากเดิมซึ่งเป็นแบบขนานซึ่งสามารถขยายการเปิด-ปิดอุปกรณ์ไฟฟ้าได้เพิ่มมากขึ้น และได้ใช้ไมโครคอนโทรลเลอร์ AT89S8252 ควบคุมการทำงานทั้งหมด และการสั่งงานทั้งหมดนี้จะใช้การสั่งงานผ่านเครือข่ายอินเทอร์เนตซึ่งสามารถสั่งงานเครื่องใช้ไฟฟ้าได้จากที่ใดก็ได้ทั่วโลกที่มีอินเทอร์เนต

1.2 วัตถุประสงค์และเป้าหมายของโครงงาน

- เพื่อสามารถพัฒนาความสามารถของการสั่งงานเครื่องใช้ไฟฟ้าเดิมจากต่อตรงเป็นแบบไร้สาย
- เพื่อสามารถควบคุมการทำงานโดยสั่งงานผ่านพอร์ตอนุกรม
- เพื่อศึกษาการทำงาน การใช้งาน และการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์
- เพื่อนำความรู้ในการศึกษาไมโครคอนโทรลเลอร์สามารถควบคุมอุปกรณ์ไฟฟ้าได้
- สามารถใช้โปรแกรมที่สามารถใช้งานผ่านเครือข่ายอินเทอร์เนต

1.3 เค้าโครงของงาน

- ชื่อโครงงาน

  การสั่งงานเปิด-ปิดเครื่องใช้ไฟฟ้าแบบไร้สายผ่านทางวิทยุ FM

- วิธีดำเนินงาน

  - ศึกษาและรวบรวมข้อมูลของโครงงานนี้ให้ได้มากที่สุดและนำไปปรึกษาอาจารย์ที่ปรึกษาโครงงาน
  - ศึกษาโครงสร้างและการทำงานของไมโครคอนโทรลเลอร์
  - ออกแบบวงจรและเชื่อมต่ออุปกรณ์
  - ทดลองออกอากาศ
  - ทดสอบเปิด-ปิดอุปกรณ์ไฟฟ้าด้วยโปรแกรมที่เขียนไว้

มหาวิทยาลัยเทคโนโลยีสุรนารี

1.4 ขอบเขตของงาน

ทำการสร้างเครื่องเปิด-ปิดเครื่องใช้ไฟฟ้าแบบไร้สายผ่านทางวิทยุ FM มีหลักการทำงานดังนี้

- สั่งงานผ่านเวปไซค์ที่ฝั่ง Client ให้ Server
- Sever ส่งข้อมูลจากคอมพิวเตอร์แบบอนุกรมผ่านพอร์ตอนุกรม
- กล่องรับข้อมูลรับและส่งข้อมูลแบบ DTMF ออกอากาศให้กล่องที่ต่อกับเครื่องใช้ไฟฟ้า
- กล่องที่ต่อกับเครื่องใช้ไฟฟ้าส่งสถานะกลับไปกล่องที่ต่อเครื่องคอมพิวเตอร์และแสดงผลทางจอคอมพิวเตอร์

1.5 ความสำคัญหรือประโยชน์ที่คาดว่าจะได้รับ

- ได้รับความรู้เกี่ยวกับหลักการส่ง-รับข้อมูลแบบต่างๆ
- ได้ความรู้ทางด้านอุปกรณ์อิเล็กทรอนิคส์
- ได้ความรู้จากการเขียนโปรแกรมหลายๆ ภาษา อาทิเช่น C++ , ไมโครคอนโทรลเลอร์
- ได้รับความรู้เกี่ยวกับวิธีวัดผลการทำงานของอุปกรณ์และนำไปประยุกต์ใช้งานได้

1.6 การจัดรูปเล่มรายงาน

รายงานผลการศึกษาเครื่องเปิด-ปิดเครื่องใช้ไฟฟ้าแบบไร้สายแบ่งเป็น

บทที่ 1 กล่าวนำ ขอบเขตและการดำเนินงาน

บทที่ 2 อธิบายส่วนต่างๆ

บทที่ 3 ทฤษฎีและหลักการทำงานของโปรแกรม

บทที่ 4 ทฤษฎีและหลักการทำงานของ Hardware

บทที่ 5 การใช้งาน

บทที่ 6 บทสรุป

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 2
อธิบายส่วนต่าง ๆ

2.1 อุปกรณ์อิเล็คทรอนิคส์ที่สำคัญ

IC AT89S8252 เป็นไอซีไมโครคอนโทรลเลอร์ที่ใช้สำหรับควบคุมการทำงานโดยรับข้อมูลจากคอมพิวเตอร์ทางพอร์ต DB9 แบบ serial ซึ่งโดยปรกติจะเป็นแบบ parallel ทางพอร์ต printer

IC MT8888C เป็นไอซีที่สามารถรับข้อมูลที่เป็น binary มาแปลง (Encoder) เป็น Dual Tone Multi Frequency (DTMF) ได้เพื่อใช้ในการออกอากาศ

IC MT8870 เป็นไอซีที่สามารถรับข้อมูลที่เป็น DTMF มาแปลง (Decoder) เป็น binary ได้

IC MAX232 เป็นไอซีที่ใช้สำหรับรับ-ส่งข้อมูลแบบอนุกรม

2.2 โปรแกรมที่ใช้

C++ ใช้เขียนรับ-ส่งข้อมูลออกจาก computer

Microcontroller ใช้ในการเขียนโปรแกรมคำสั่งใส่ไอซี AT89S8252

Java เชื่อมต่อเครือข่ายอินเทอร์เน็ต

2.3 ทำไมต้องใช้ไมโครคอนโทรลเลอร์

เนื่องมาจากไมโครคอนโทรลเลอร์ได้มีเรียนในหลักสูตรวิชาสาขาโทรคมนาคมจึงทำให้ถนัดในการใช้งานและทำความเข้าใจได้ไม่ยากมาก และตัวไอซีก็มีความสามารถในการทำงานมากที่จะแทน Hardware เพียงอัดโปรแกรมใส่ด้วยภาษา exembly เท่านั้นไอซีก็จะสามารถทำงานได้ตามคำสั่ง ซึ่งใน Project นี้เราใช้ ไอซี AT89S8252 ซึ่งเป็นตระกูล MSC51

2.4 ทำไมต้องใช้ภาษาC++

เนื่องจากในภาษาC++ นั้นสามารถที่จะควบคุมพอร์ตต่าง ๆ ได้เพราะมีโปรแกรมที่ใช้การเขียนและเป็นแบบฟอร์มที่ศึกษาได้ไม่ยากและสามารถ Compile ภาษาC++ เป็น File *.dll เพื่อใช้ในการคุยกับภาษา JAVA APPLETได้

2.5 ทำไมต้องใช้ JAVA

เพราะ JAVA เป็นโปรแกรมที่ สามารถเขียน Home page และสามารถคุยกับ File *. dll ได้

2.6 ทำไมต้องใช้ FM.ในการมอดดูเลตและดีมอดดูเลตสัญญาณ

เนื่องจากมอดดูเลตและดีมอดดูเลตสัญญาณนั้นมีด้วยกันหลายวิธีที่น่าสนใจแต่ที่เลือกใช้การมอดดูเลตและการดีมอดดูเลตแบบ FM. นั้นเพราะเป็นวงจรที่มีความถี่สูงและสามารถหาวงจรและทดลองในการทำงานได้ง่ายและมีราคาไม่แพง

2.7 ทำไมต้องใช้ Port อนุกรม

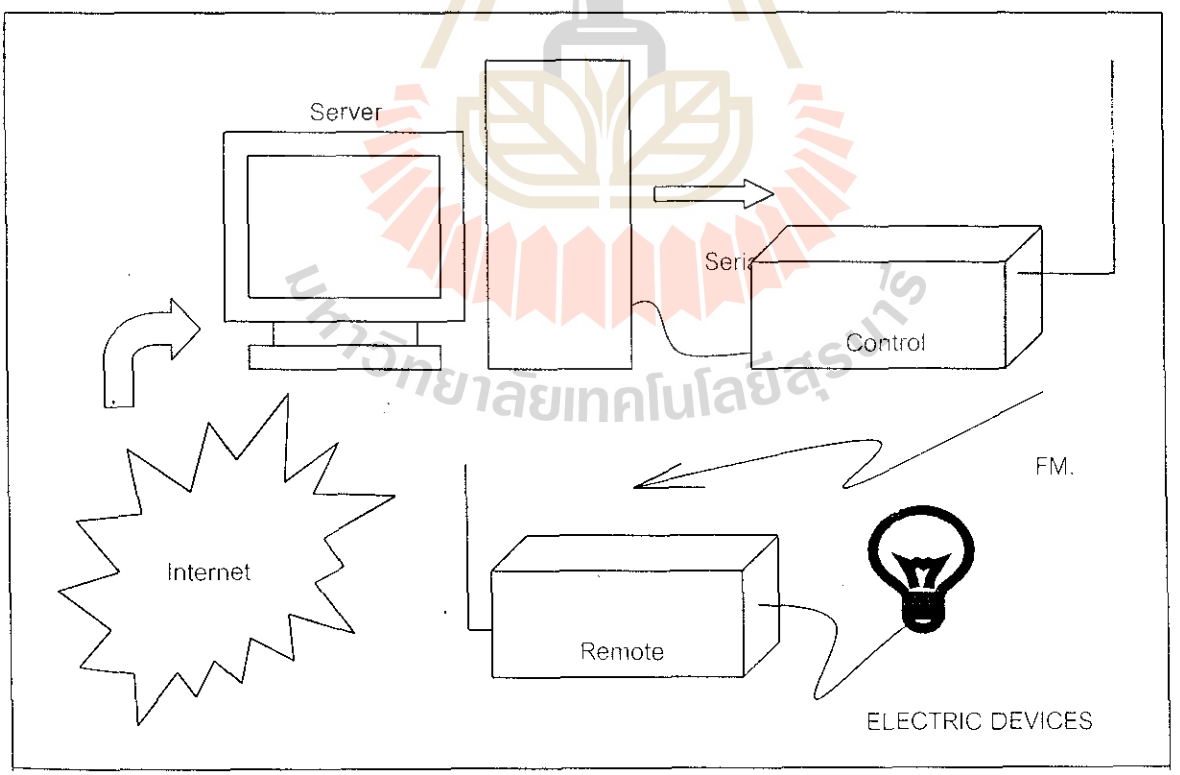
ที่ใช้ Port อนุกรมนั้นเนื่องจากใน Project นี้เราต้องส่งข้อมูลทีละหลาย bit และ สามารถขยายอุปกรณ์ได้อีกมากขึ้นอยู่กับการตั้งรูปแบบของข้อมูล เพราะสามารถส่งข้อมูลทีละหลายๆบิตได้เพื่อแปลเป็นหลาย ๆ ความหมาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 3
ทฤษฎีและหลักการทำงานของโปรแกรม

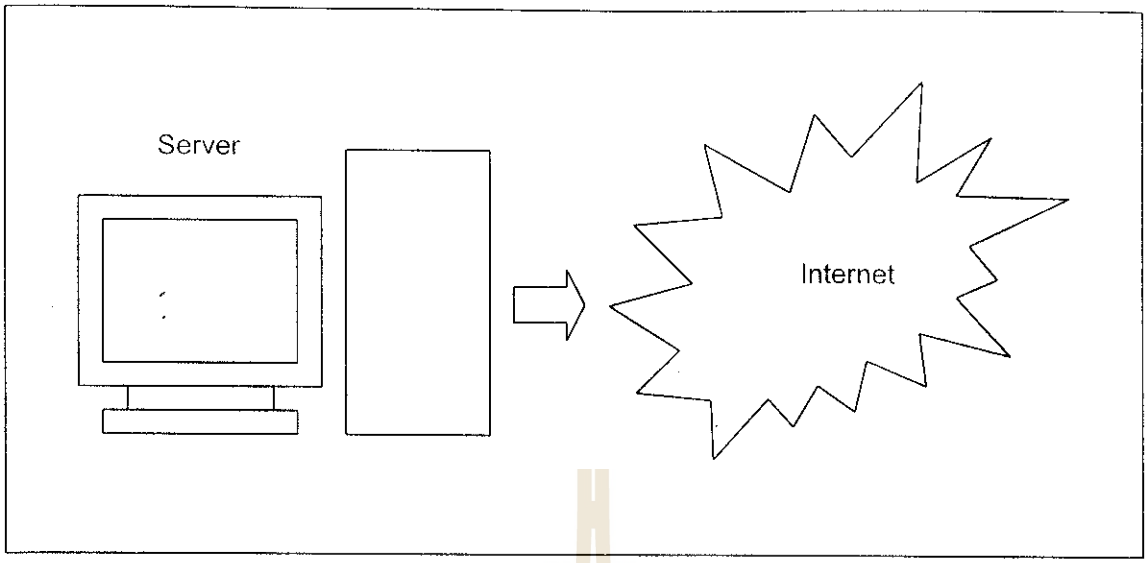
3.1 กล่าวนำและทฤษฎี

Client สั่งงานผ่านเครือข่ายอินเทอร์เนตเข้าสู่ Sever และให้ภาษา Java Applet คุยกับภาษา C++ เพื่อสั่งงานผ่าน Serial Port แบบ อนุกรมซึ่งโดยปรกติจะเป็นแบบขนาน โดยส่งข้อมูลโดยให้ข้อมูลออก 1 เส้นทาง และรับ ข้อมูล 1 เส้นทาง เช่นกัน เพื่อส่งข้อมูลที่ได้ไปยัง Hardware คือ ไอซี AT89S8252 ซึ่งไอซีตัวนี้จะส่งข้อมูลแบบบิตไปยังไอซี MT8888 เพื่อแปลงข้อมูลเป็น DTMF และส่ง DTMF ออกอากาศด้วยวิธีต่อไมล์ลอยย่าน FM (88-108MHz) และใช้วิทยุ FM ในย่านเดียวกัน จูนความถี่ให้ตรงกันเพื่อรับข้อมูลและต่อเข้ากับไอซี MT8870 เพื่อแปลงสัญญาณ DTMF เป็นข้อมูลแบบบิตเพื่อไปสั่งงานเครื่องใช้ไฟฟ้าต่อไป และฝั่งเครื่องใช้ไฟฟ้าเมื่อได้รับคำสั่งก็จะส่งข้อมูลกลับไปว่ามีเครื่องใช้ไฟฟ้าตัวใดติดบ้างเพื่อที่ให้ผู้สั่งงานที่ฝั่ง Client ได้ทราบถึงสถานะของเครื่องใช้ไฟฟ้า



รูปที่1 ฝั่ง Server

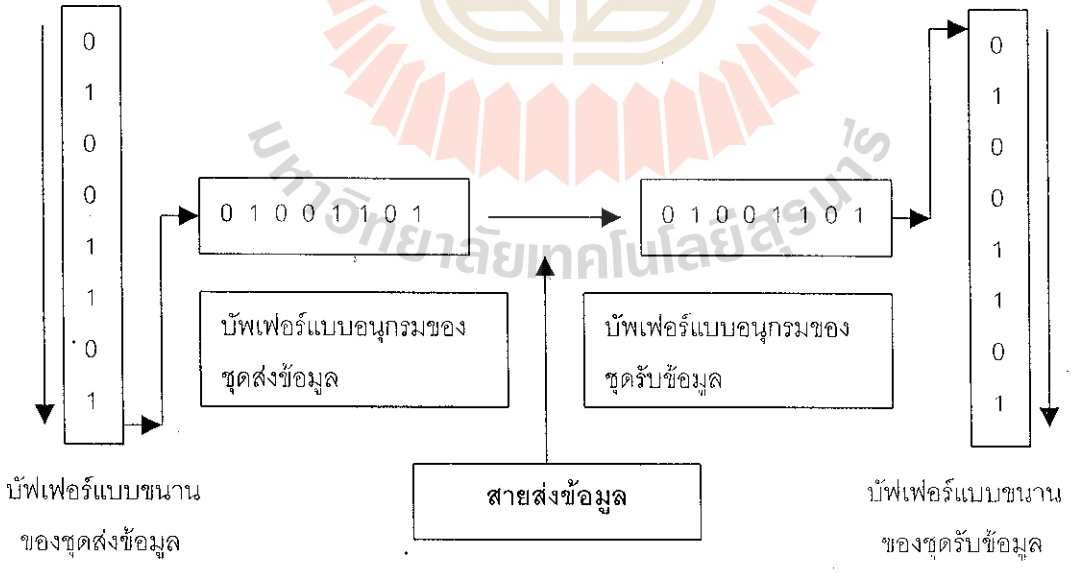
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 2

3.2 การรับส่งข้อมูลแบบ Serial ด้วยไอซี MCS-51

โดยปรกติทั่วไปข้อมูลที่เก็บอยู่ในหน่วยความจำของระบบไมโครโปรเซสเซอร์จะต้องเป็นแบบขนานเสมอ แต่เมื่อต้องการส่งข้อมูลแบบอนุกรมก็จำเป็นต้องแปลงข้อมูลในหน่วยความจำจากแบบขนานให้เป็นแบบอนุกรม และตรงกันข้ามหากต้องการรับข้อมูลแบบอนุกรมก็ต้องนำมาแปลงเป็นแบบขนานก่อนที่จะนำไปใช้งานต่อไป



รูปที่ 3 รูปแบบการส่งข้อมูลแบบอนุกรม

ข้อกำหนดและขั้นตอนการรับหรือส่งข้อมูลแบบอนุกรม

1. การส่งหรือการรับต้องเริ่มที่บิตต่ำก่อนเสมอ
2. ช่วงเวลาในการรับหรือการส่งข้อมูลแต่ละบิตต้องเท่ากัน
3. ข้อมูลจากซีพียูต้องแปลงเป็นอนุกรมก่อนที่จะรับหรือส่ง
4. Marking ต้องเป็นสถานะลอจิกเดียวกันตลอดเวลาไม่มีการส่งหรือการรับข้อมูล
5. ชุดข้อมูลต้องส่งรหัส 1 บิตต่อจาก Marking ซึ่งเรียกว่าบิตเริ่มต้น (Start bit) และต้องมีสถานะลอจิกตรงกันข้ามกับ Marking
6. ตัวส่งต้องส่งรหัสพาริตี้ (Parity bit) อีก 1 บิตต่อจากข้อมูลในแต่ละไบต์เพื่อบอกสถานะข้อมูลว่าเป็นคู่หรือคี่
7. ตัวส่งต้องส่งบิตหยุดส่งข้อมูล (Stop Bit) ต่อจากบิตพาริตี้อีก 1 หรือ 1 ½ หรือ 2 บิตก็ได้ปิดท้ายข้อมูลโดยกำหนดให้มีสถานะเหมือนกับ Marking

คำศัพย์เกี่ยวกับการรับหรือส่งข้อมูลแบบอนุกรม

Marking คือ สถานะของสายข้อมูล ซึ่งอาจเป็น 0 หรือ 1 ก็ได้ แต่ต้องมีสถานะเป็นตรงกันข้ามกับ Start bit

Start bit คือ บิตเริ่มต้นในการส่งข้อมูลเป็นบิตแรกเพื่อไปบอกให้ตัวรับทราบว่าจะมีการส่งข้อมูลตามมา

Parity bit คือ สถานะของบิตที่ส่งตามหลังจากข้อมูลเพื่อบอกข้อมูลนั้นเป็นคู่หรือคี่(Even or Odd

Stop bit คือ บิตสถานะบอกการสิ้นสุดของการส่งข้อมูลอาจเป็นขนาด 1 บิต 1 ½ บิต หรือ 2 บิต ก็ได้

Baud rate คือ อัตราการส่งข้อมูลคิดเป็นจำนวนบิตต่อวินาที มีค่าเป็น 110 150 300 600 900 1200 2400 ... 9600 หรือ มากกว่านี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 4
ทฤษฎีและหลักการทำงานของ Hardware

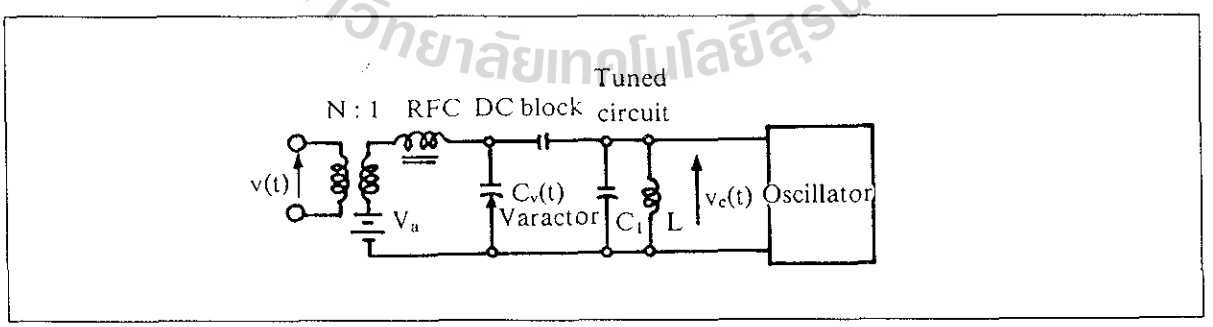
4.1 ทฤษฎีเกี่ยวกับการรับ-ส่งแบบ FM

การสร้างสัญญาณ FM

การสร้างสัญญาณ FM นั้นสามารถแบ่งได้กว้าง ๆ เป็น 2 แบบด้วยกันคือ แบบทางตรงกับแบบทางอ้อม แบบทางตรงนั้นหมายถึงการนำสัญญาณที่เข้ามอดดูเลตไปเปลี่ยนความถี่ของคลื่นพาห์โดยตรง สำหรับแบบทางอ้อมนั้นก็จะเป็นวิธีการที่อาศัยวงจรหลาย ๆ อย่างประกอบกัน ในหัวข้อนี้จะกล่าวถึงการสร้างสัญญาณ FM โดยวิธีการทั้งสองนี้

1. การสร้างสัญญาณ FM แบบทางตรง

การสร้างสัญญาณ FM แบบทางตรงนั้นเป็นการนำเอาสัญญาณที่จะมอดูเลตไปเปลี่ยนความถี่ของคลื่นพาห์โดยตรง ซึ่งออสซิลเลเตอร์ที่ทำงานในลักษณะนี้ได้ก็มีรีเฟลกซ์ไคลสตรอน (reflex klystron) และกันน์ไดโอด (Gunn diode) ที่ใช้ในย่านความถี่ไมโครเวฟเป็นต้น ออสซิลเลเตอร์ทั้งสองแบบนี้ ความถี่ที่ออสซิเลลตออกมาจะเปลี่ยนไปตามแรงดันไบอัสในรูปเชิงเส้น ดังนั้นเมื่อนำสัญญาณที่เข้ามามอดูเลตไปเปลี่ยนแรงดันไบอัสก็จะทำให้ได้สัญญาณ FM ออกมาโดยตรง สำหรับความถี่ย่านต่ำกว่าไมโครเวฟลงมานั้น เรามักจะใช้ชิ้นส่วนอิเล็กทรอนิกส์ที่เรียนว่าวาแรกเตอร์ (varactor หรือ variable reactor) ชิ้นส่วนนี้จะมีคุณสมบัติเฉพาะคือค่าคาแปซิแตนซ์จะเปลี่ยนแปลงไปตามระดับของแรงดันที่คร่อมอยู่ ดังนั้นถ้าเราใช้วาแรคเตอร์นี้เป็นส่วนหนึ่งของวงจรเรโซแนนท์ที่ใช้ในวงจรออสซิเลเตอร์ เราก็จะสามารถสร้างสัญญาณ FM ได้โยป้อนสัญญาณที่เข้ามอดูเลตไปเปลี่ยนค่าคาแปซิแตนซ์ของวาแรคเตอร์



รูปที่ 4 ตัวอย่างวงจรที่ใช้วาแรคเตอร์ในการสร้างสัญญาณ FM

ในการวิเคราะห์ค่าความเบี่ยงเบนที่จะสร้างได้นั้นอาจจะทำได้โดยให้ค่าคาแปซิแตนซ์ของส่วนที่เป็นคาแปซิเตอร์เขียนได้เป็น

$$C(t) = C_o - C_v v(t) \quad (1)$$

โดยที่ค่า $C_v v(t)$ มีขนาดเล็กเมื่อเทียบกับ C_o เนื่องจากความถี่เรโซแนนท์ของวงจรเรโซแนนท์เขียนได้เป็น $f_c = 1/(2\pi\sqrt{LC})$ เมื่อใช้ค่า C (t) ตามสมการ (1) จะเขียนความถี่เรโซแนนท์นี้ได้ในรูปต่อไปนี้

$$\therefore \quad f_c(t) = 1/(2\pi\sqrt{LC(t)}) = 1/(2\pi\sqrt{LC_0})\{1 - \frac{Cv}{Co}v(t)\}^{-1/2}$$

$$\cong f_c\{1+\frac{Cv}{2Co}v(t)$$

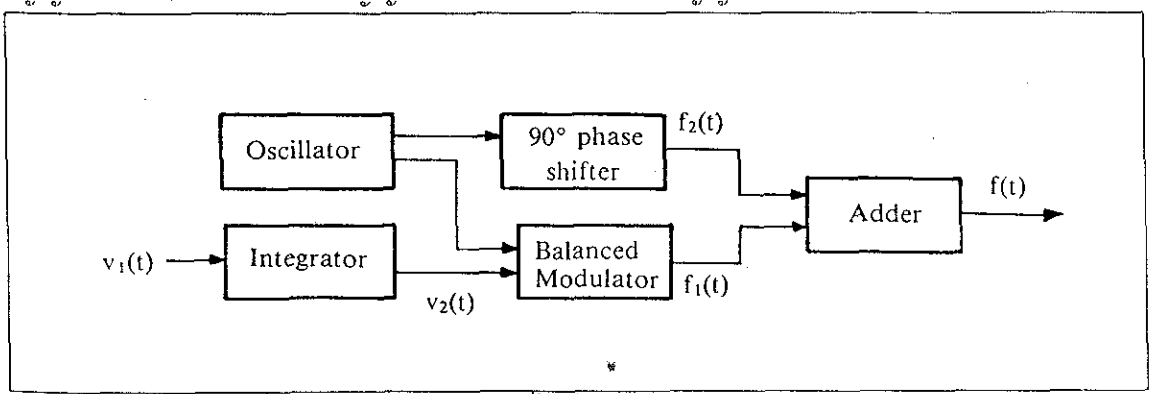
$$\cong f_c + \frac{Cv}{2Co} f_c\, v(t) \quad (2)$$

ตามผลที่ได้ในสมการ (2) นี้เราจะเห็นได้ว่าค่าความเบี่ยงเบนชั่วขณะจะแปรผันตรงกับสัญญาณที่เข้ามามอดูเลต แสดงว่าเราสามารถสร้างสัญญาณ FM ได้โดยตรงจากการเปลี่ยนค่าคาปาซิแตนซ์ของวาแรคเตอร์ดังกล่าวข้างต้น

วิธีสร้างสัญญาณ FM แบบทางตรงที่กล่าวมานี้มีข้อดีที่ทำได้ง่าย และค่าเบี่ยงเบนความถี่ที่ได้จะสูงพอสมควร แต่ก็มีข้อเสียที่สำคัญคือ ค่า f_c จะดริฟท์ (drift) ได้ง่าย จึงจำเป็นต้องมีวงจรชดเชยอุณหภูมิและวงจรควบคุมความถี่ f_c ให้คงที่ซึ่งก็เพิ่มความยุ่งยากของวงจรให้สูงขึ้น

2. การสร้างสัญญาณ FM แบบทางอ้อม

การสร้างสัญญาณ FM แบบทางอ้อมโดยทั่วไปนั้นจะใช้ส่วนประกอบของวงจรตามที่แสดงไว้ในรูป กล่าวคือจะใช้วงจรอินทิเกรเตอร์ในการอินทิเกรตสัญญาณที่เข้ามอดูเลต และส่วนที่เป็นบาลานซ์มอดูเลเตอร์ เฟสชิฟเตอร์และวงจรบวกจะทำหน้าที่เป็นเฟสมอดูเลเตอร์ ซึ่งจะทำให้สัญญาณที่ผ่านวงจรนี้จะเป็นสัญญาณ FM ในการแสดงว่าสัญญาณที่



รูปที่ 5 ส่วนประกอบของวงจรที่ใช้สร้างสัญญาณ FM แบบทางอ้อม

แต่ละจุดเปลี่ยนแปลงไปอย่างไรนั้นจะพิจารณาได้โดยใช้ รูปประกอบดังนี้ ก่อนอื่นสัญญาณคลื่นพาห์และสัญญาณ $v_2(t)$ ที่เข้าสู่วงจรบารานซ์มอดูเลเตอร์ จะทำให้เกิดสัญญาณขาออกเป็นสัญญาณ DSB ที่เขียนได้ในรูปต่อไปนี้

$$f_1(t) = A\, v_2(t) \cos \omega_c t \qquad (3)$$

ส่วนสัญญาณคลื่นพาห์ที่ผ่านเฟสชิฟเตอร์ 90° นั้น จะเขียนได้ดังนี้

$$f_2(t) = B\, v_2(t) \cos\left(\omega_c t - \frac{\pi}{2}\right) = B \sin \omega_c t \qquad (4)$$

เมื่อสัญญาณทั้งสองนี้รวมกันที่วงจรบวก สัญญาณขาออกจะเป็นดังนี้

$$f(t) = f_1(t) + f_2(t) = A\, v_2(t) \cos \omega_c t + B \sin \omega_c t$$

$$= B\sqrt{\frac{A^2}{B^2} v^2{}_2(t)} \sin[\omega_c t + \phi(t)] \qquad (5)$$

โดยที่

$$\phi(t) = \tan^{-1}\left[\frac{A}{B} v_2(t)\right] \qquad (6)$$

ถ้าระดับของสัญญาณคลื่นพาห์สูงมากจนทำให้ $A/B \ll 1$ ตามสมการ (6) ก็จะประมาณค่าได้เป็น

$$\phi(t) \cong \frac{A}{B} v_2(t) \qquad (7)$$

และเนื่องจาก $v_2(t)$ ก็มีขนาดเล็กเช่นเดียวกันเมื่อเทียบกับขนาดคลื่นพาห์ดังนั้นสมการ (5) จะประมาณค่าได้เป็น

$$f(t) \cong B \sin[\omega_c t + \frac{A}{B} v_2(t)] \qquad (8)$$

สมการ (8) นี้ก็จะแสดงสัญญาณ PM ที่มี $v_2(t)$ เป็นสัญญาณที่เข้ามามอดูเลต และมี A/B เป็นมอดูเลชันอินเด็กซ์ซึ่งในกรณีนี้ก็จะมีขนาดเล็กมาก เนื่องจากสัญญาณ $v_2(t)$ ได้จากการอินทิเกรตสัญญาณที่เข้ามามอดูเลตจริง ๆ คือ $v_1(t)$ ดังนั้นสมการ (8) จะเขียนได้เป็น

$$f(t) \cong B \sin[\omega_c t + \int_{-\infty}^{t} v_2(t)d(t)\ ] \qquad (9)$$

ซึ่งในสมการ (9) นี้ก็จะเป็นสัญญาณ FM ที่มีความถี่ต่ำ วิธีสร้างสัญญาณแบบทางอ้อมที่กล่าวมานี้มีชื่อว่า วิธีมอดูเลตแบบอาร์มสตรอง (Armstrong's modulation method) ในการมอดูเลตแบบนี้จะมีข้อจำกัดที่ค่าเบี่ยงเบนความถี่ต่ำ ซึ่งหมายถึงสัญญาณที่ได้ในขั้นนี้ก็จะเป็นสัญญาณ FM

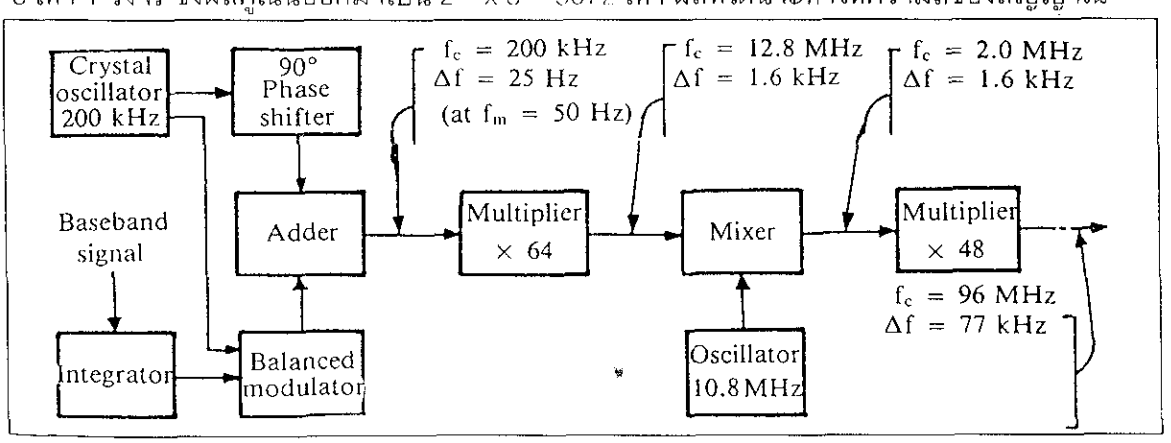
มหาวิทยาลัยเทคโนโลยีสุรนารี

แบบแบนด์วิดท์แคบ กรณีที่ต้องการสัญญาณ FM แบบแบนด์วิธกว้างก็จะต้องมีการออกแบบตั้งแต่ต้นคือ ใช้วงจรคูณความถี่เข้าช่วย จะทำให้ความถี่ของคลื่นพาห์สูงขึ้นพร้อม ๆ กับทำให้ค่าเบี่ยงเบนความถี่สูงขึ้น สัญญาณที่ผ่านออกจากวงจรคูณความถี่ฟิลเตอร์ผ่านแถบความถี่อันดับที่ n แล้วสามารถเขียนได้ในรูปต่อไปนี้

$$v_c(t) = C \sin[n\omega_c t + nk \int_{-\infty}^{t} v_1(t)dt\ ] \qquad (10)$$

ในทางปฏิบัติค่า n ที่ใช้มักเป็น 2 หรือ 3 เพราะเป็นช่วงที่ใช้งานได้ผลที่สุด กรณีที่ต้องการวงจรคูณความถี่ที่มีตัวคูณสูงกว่านี้ ก็จะทำได้โดยใช้ตัวคูณหลาย ตัวประกอบกัน เช่น ต้องการตัวคูณความถี่ 64 เท่า ก็จะทำได้โดยใช้ตัวคูณความถี่ 2 เท่าเป็นจำนวน 6 ตัว เป็นต้น ข้อสังเกตที่สำคัญของการคูณความถี่ที่กล่าวมานี้คือ ค่ามอดูเลชันอินเด็กซ์หรือ β สูงขึ้น ซึ่งผลที่ตามมาก็คือค่าเบี่ยงเบนความถี่ หรือ Δf จะสูงขึ้น นั่นหมายความว่าสัญญาณเบสแบนด์ที่เข้ามามอดูเลตมีความถี่เป็น f_m ขนาดของไซด์แบนด์ก็จะกว้างออกไป โดยที่ช่วงห่างระหว่างสเป็กตรัมที่อยู่ติดกันในไซด์แบนด์ก็ยังเป็น f_m เหมือนเดิม แต่สิ่งที่แตกต่างไปจากเดิมก็คือขนาดของสเป็กตรัมจะเปลี่ยนไปจากเดิม

ในทางปฏิบัตินั้นการใช้วงจรคูณหลาย ๆ ชั้นเพื่อให้ได้ค่าเบี่ยงเบนความถี่ตามที่ต้องการนั้นถ้าค่าเบี่ยงเบนความถี่ข้างต้นมีค่าต่ำมาก ก็จะต้องมีการคูณความถี่หลายชั้นมาก ซึ่งอาจจะทำให้ความถี่ของสัญญาณคลื่นพาห์สูงกว่าความถี่ของคลื่นพาห์ที่ต้องการได้ ดังแสดงในตัวอย่างของระบบที่ต้องการสร้างสัญญาณ FM ความถี่ 96 MHz จากความถี่คลื่นพาห์เริ่มต้นเป็น 200 kHz โดยที่วงจรมอดูเลเตอร์ในขั้นแรกสามารถสร้างค่าเบี่ยงเบนความถี่เพียง 25 Hz แต่ตามมาตรฐานของวิทยุกระจายเสียง FM จะยอมให้ใช้ค่าเบี่ยงเบนความถี่ถึง 75 kHz การเพิ่มค่าเบี่ยงเบนความถี่ให้ได้ตามต้องการนี้ จะต้องมีวงจรคูณความถี่หลาย ๆ ชั้นมาทำการคูณความถี่ขึ้นไปอีกประมาณ 75 kHz/25 Hz = 3,000 เท่า ซึ่งอาจจะใช้วงจรคูณความถี่ 2 เท่า 10 วงจร ร่วมกับวงจรคูณความถี่ 3 เท่า 1 วงจร ซึ่งผลคูณนี้ออกมาเป็น $2^{10} \times 3 = 3072$ เท่า ผลที่ได้นี้จะทำให้ความถี่ของสัญญาณ



รูปที่ 6 ตัวอย่างส่วนประกอบของระบบสร้างสัญญาณ FM วิธีอาร์มสตรองร่วมกับวงจรคูณความถี่และมิกเซอร์

คลื่นพาห์สูงขึ้นเป็น 3072 x 200 kHz = 614.4 MHz และเมื่อต้องการใช้มิกเซอร์ในการแปลงความถี่นี้มาเป็น 96 MHz สัญญาณจากโลคอลออสซิลเลเตอร์ที่มีความถี่เป็น 614.4 $\pm$ 96 MHz = 710.4 MHz หรือ 518.4 MHz ซึ่งก็หมายถึงสูงมาก ถ้าต้องการหลีกเลี่ยงสภาพดังกล่าวนี้ก็จะทำได้โดยวิธีที่แสดงไว้ในรูป คือการแทรกโดลคอลออสซิลเลเตอร์ไว้ระหว่างกลางของกลุ่มวงจรคูณความถี่ในกรณีที่แสดงไว้ในรูปนั้นความถี่ของโลคอลออสซิลเลเตอร์ที่ต้องการก็จะลดต่ำลงเหลือ 10.8 MHz ซึ่งความถี่ขนาดนี้จะสามารถสร้างให้มีเสถียรภาพของความถี่สูง ๆ ได้โดยใช้ผลึกแร่ และอันที่จริงความถี่ 10.8 MHz นี้อาจจะสร้างได้โดยใช้ผลึกแร่ตัวเดียวกับที่ใช้สร้างความถี่ 200 kHz ได้โดยทำการคูณความถี่ 54 เท่า เมื่อใช้ผลึกแร่ร่วมกันก็จะมีข้อดีที่สำคัญคือดริฟท์ของความถี่ที่เกิดขึ้นในออสซิลเลเตอร์ต้นทางก็จะเกิดขึ้นในโลคอลออสซิลเลเตอร์พร้อมกันไป และผลเสียจากการดริฟท์ก็จะน้อยลงทำให้สัญญาณขาออกมีความถี่คลื่นพาห์ที่คงที่ตลอดเวลา

การดีมอดูเลตสัญญาณ FM

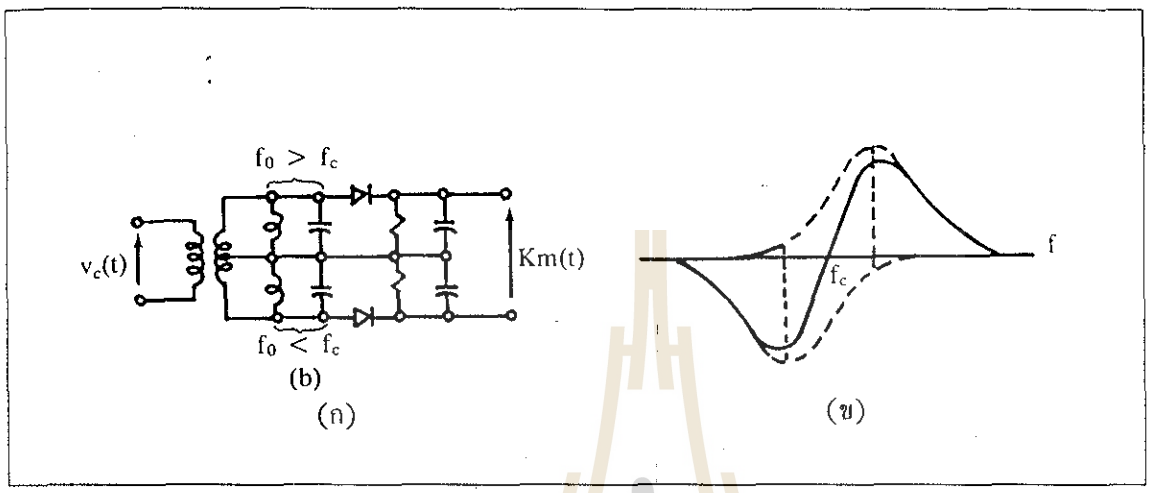
การดีมอดูเลตสัญญาณ FM นั้นจะทำได้หลายแบบด้วยกัน เช่น คือ แบบที่ใช้การแปลงสัญญาณ FM เป็นสัญญาณ AM แบบควอดเดรเจอร์ดีเทคชัน และแบบที่ใช้เฟสล็อคลูป (phase locked loop ย่อว่า PLL) ในที่นี้จะอธิบายแต่วิธีที่ใช้การแปลงสัญญาณ FM เป็นสัญญาณ AM

1. **การดีมอดูเลตสัญญาณ FM โดยการแปลงเป็นสัญญาณ AM**

เป้าหมายของการแปลงสัญญาณ FM เป็นสัญญาณ AM นั้นก็เพื่อให้สามารถดีเทคสัญญาณเบสแบนด์กลับออกมาได้โดยวิธีเอ็นเวลโลปดีเทคชั่น การแปลงสัญญาณ FM เป็นสัญญาณ AM จะทำได้โดยใช้วงจรที่สามารถแปลงการเบี่ยงเบนความถี่ชั่วขณะให้เป็นการเปลี่ยนแปลงของขนาดได้ วงจรที่ทำหน้าที่ได้ดีก็คือวงจรดิฟเฟอเรนชิเอเตอร์ ทั้งนี้เนื่องจากเมื่อเราดิฟเฟอเรนชิเอทสัญญาณ FM สัญญาณที่ได้จะเป็นสัญญาณที่มีขนาดเปลี่ยนแปลงเหมือนสัญญาณ AM ดังนี้ ถ้าให้สัญญาณ FM เป็น $A_c\cos[\omega_c t + k\int m(t)\,d(t)]$ สัญญาณที่ผ่านวงจรดิฟเฟอเรนชิเอเตอร์จะเขียนได้เป็น

$$\frac{d}{dt}[A_c\cos\{\omega_c t + k\int m(t)\,d(t)\}] = A_c[\omega_c - km(t)]\sin\{\omega_c t + k\int m(t)\,d(t)\} \qquad (1)$$

$$y(t) = Km(t) \quad (2)$$



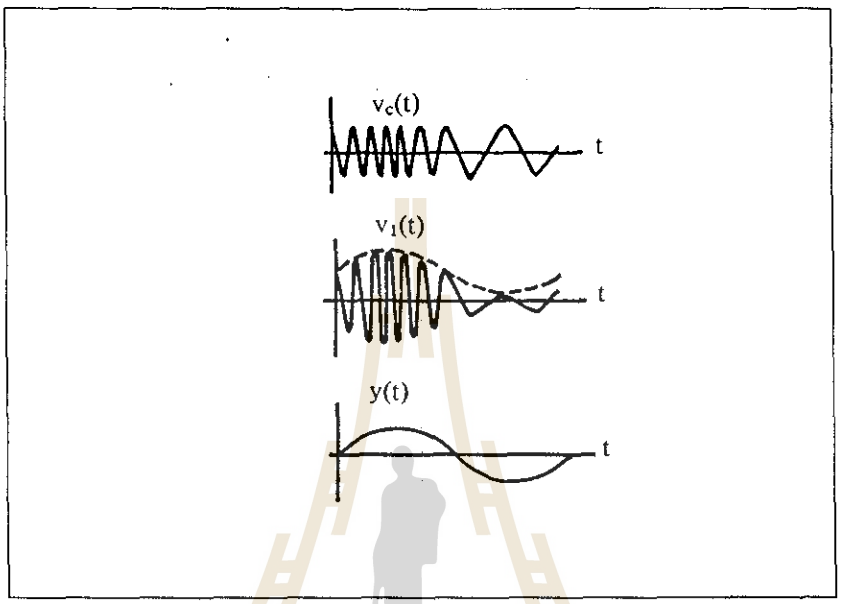
รูปที่ 8 วงจรแยกแยะความถี่ และคุณสมบัติในการเปลี่ยนความถี่เบี่ยงเบนให้เป็นการเปลี่ยนแปลงของขนาดสัญญาณ

โดยที่ K เป็นค่าคงที่ และวงจรแยกแยะความถี่นี้จะต้องมีคุณสมบัติเป็นเชิงเส้นตลอดช่วงความถี่ $f_c \pm \Delta f$

เมื่อพิจารณารายละเอียดเกี่ยวกับการเปลี่ยนแปลงของสัญญาณที่เข้ามาในวงจรแยกแยะความถี่โดยใช้เส้นประ เราจะพบว่า ตรงตำแหน่งที่สัญญาณออกจากวงจรเรโซแนนท์ที่มีความถี่เรโซแนนท์สูงกว่า f_c ขนาดของสัญญาณจะเปลี่ยนแปลงไปตามที่ความถี่กล่าวคือ ถ้าความถี่เป็น f_c ขนาดของสัญญาณ V_c เมื่อความถี่สูงขึ้นกว่า f_c สัญญาณขาออกก็จะมีขนาดสูงขึ้นกว่า V_c ในทางกลับกันถ้าความถี่ต่ำลงกว่า f_c สัญญาณขาออกก็จะต่ำกว่า V_c และถ้าความชันตรงบริเวณสูงและต่ำกว่า f_c สามารถอนุโลมได้ว่าเป็นเส้นตรง สัญญาณที่ออกจากวงจรเรโซแนนท์ก็จะมีขนาดเปลี่ยนแปลงไปกกับค่าเบี่ยงเบนความถี่ชั่วขณะในลักษาณะของสัญญาณ AM หรืออาจจะเขียนสัญญาณดังกล่าวได้ในรูปต่อไปนี้

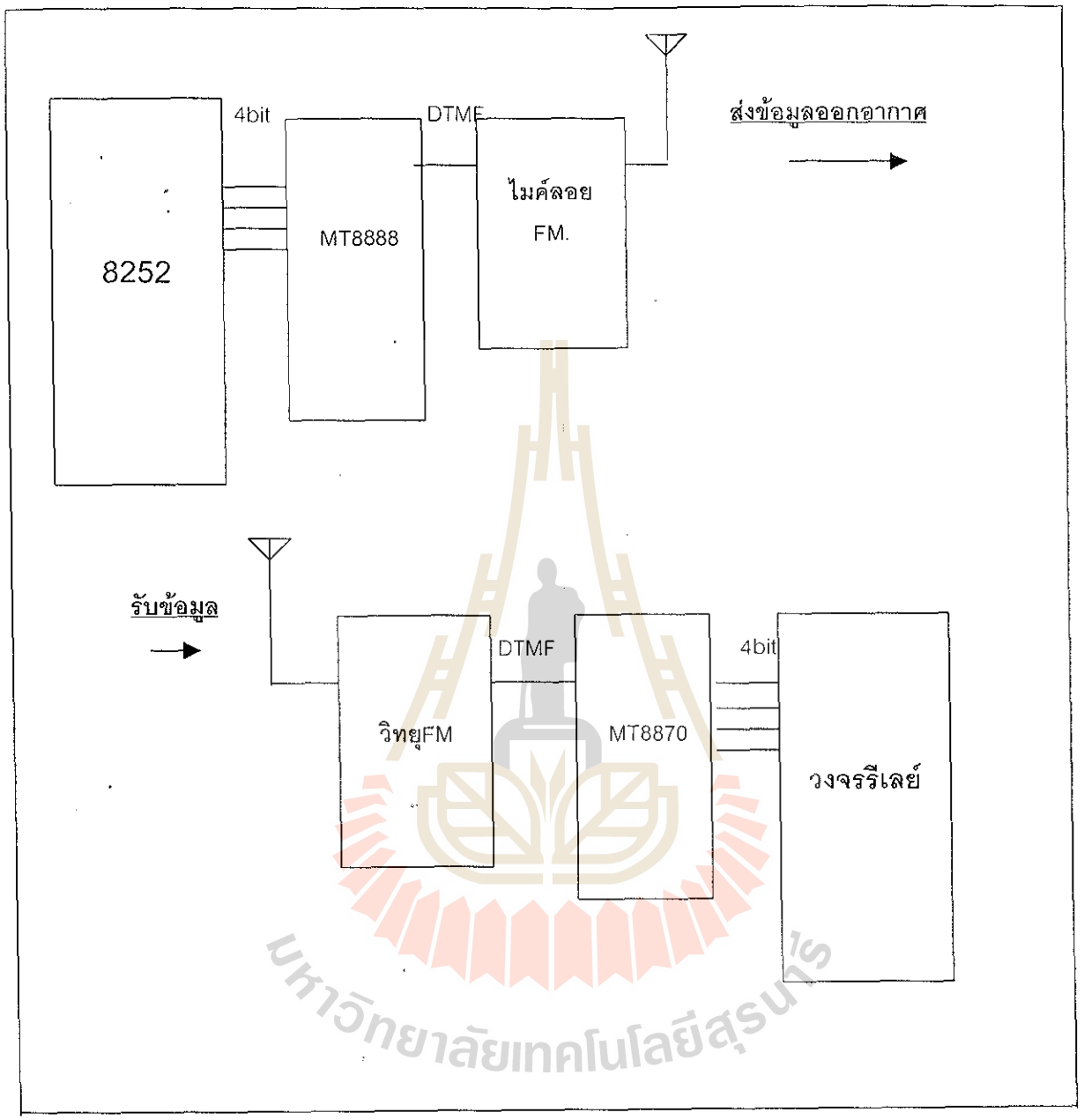
$$V_f(t) = V_c\{1 + km(t)\}\cos[\omega_c t + k\int m(t)\,d(t)] \quad (3)$$

จึงอาจจะกล่าวได้ว่าวงจรเรโซแนนท์จะทำการแปลงสัญญาณ FM ที่เข้ามาให้เป็นสัญญาณ AM ซึ่งสัญญาณ AM นี้จะถูกวงจรเอ็นเวลโลปดีเทคเตอร์ดีเทคออกมาเป็นสัญญาณเบสแบนด์ทันที



รูปที่ 9 รูปร่างของสัญญาณ FM สัญญาณ AM ที่ขาออกของวงจรเรโซแนนท์และสัญญาณที่ดีเทคได้

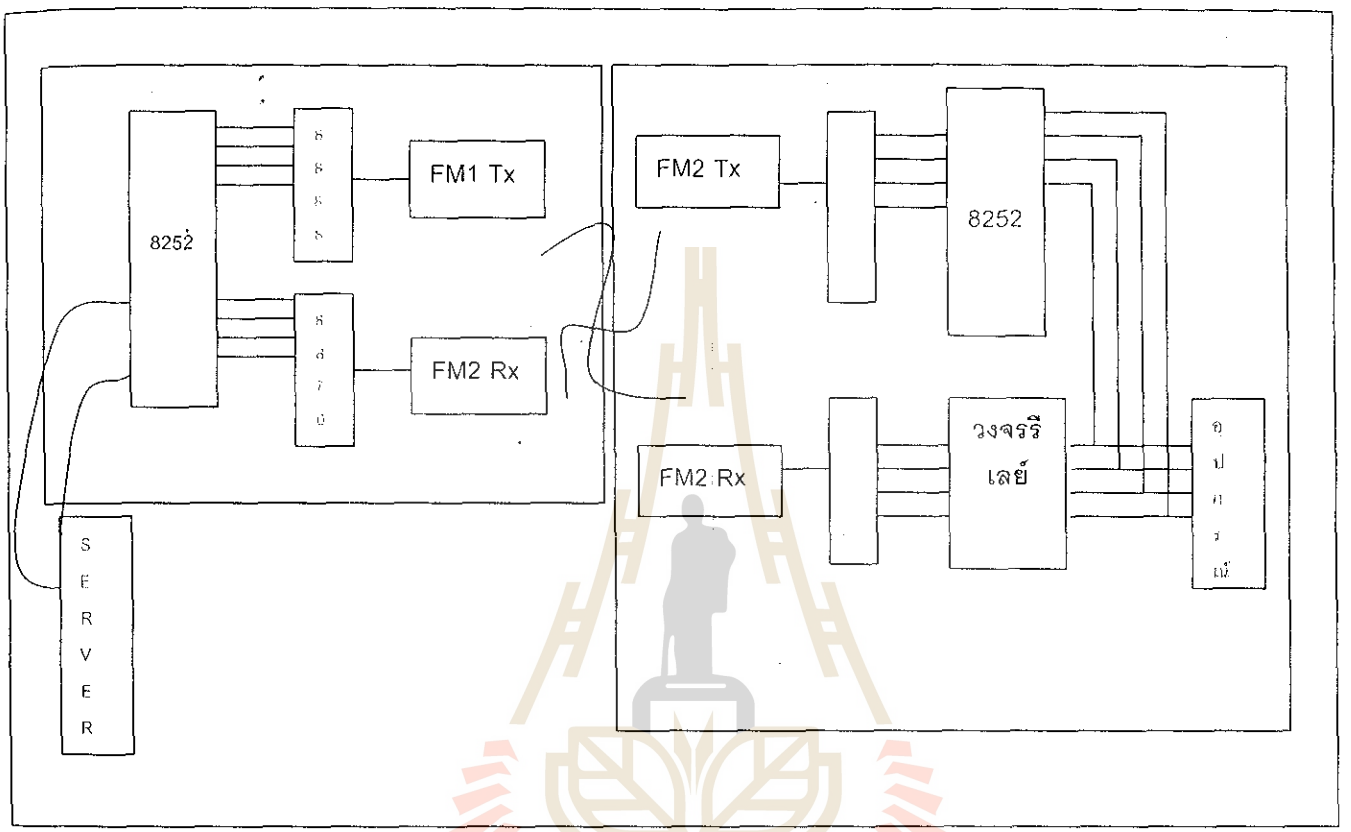
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 10 แสดงการต่อวงจร

มหาวิทยาลัยเทคโนโลยีสุรนารี

วงจรรวมตัวสั่งงานและตัวรับคำสั่ง



รูปที่ 11 รูปแบบการต่ออุปกรณ์และชิ้นงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.2 RS232

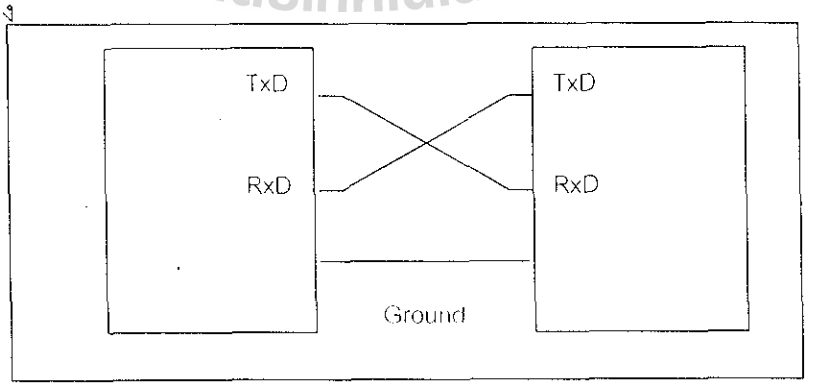
- RS232 เป็นมาตราฐานของการสื่อสารข้อมูลแบบอนุกรมกำหนดขึ้นโดย EIA (Electronics Industries Association)
- สามารถใช้ได้กับการส่งข้อมูลแบบ Synchronous และ Asynchronous Transmission
- สามารถส่งข้อมูลด้วยอัตราบอดเรดสูงสุด 20,000 bps ในระยะทาง 15 เมตร

ระดับแรงดันของ RS232

- Logic1 แทนด้วยระดับแรงดัน –3 ถึง –25
- Logic0 แทนด้วยระดับแรงดัน +3 ถึง +25
- ส่วนระดับแรงดัน +3 ถึง –3 ไม่ถูกกำหนด
- สัญญาณที่ใช้ในไมโครคอนโทรลเลอร์เป็นสัญญาณ TTL (Transistor Transistor Logic) มีระดับแรงดัน 5 V สำหรับ Logic 1 และ 0 V สำหรับ Logic 0
- การอินเตอร์เฟส MSC-51 กับอุปกรณ์ที่ปลายทางข้อมูล (DTE) ผ่านพอร์ตอนุกรมโดยใช้มาตราฐาน RS232 จะต้องมีตัวขับสาย (Line driver) หรือตัวแปลงแรงดัน (Voltage converter) เพื่อเปลี่ยนระดับแรงดันของ RS232 หรือจากสัญญาณ RS232 เป็น TTL โดยมีไอซีที่นิยมใช้คือ MAX232

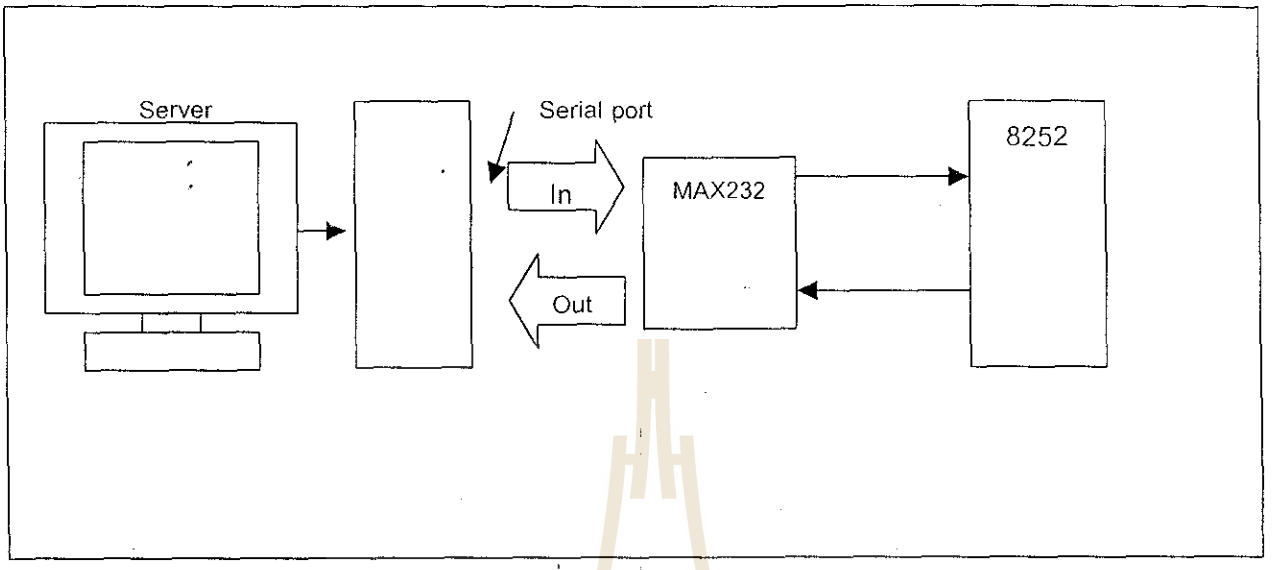
Null Modem Connection

- เราสามารถต่ออุปกรณ์ DTE เข้าด้วยกันโดยใช้นัลล์โมเดม (Null Modem) สำหรับส่งข้อมูลระยะสั้น ๆ โดยไม่ต้องใช้อุปกรณ์ปลายทางสายวงจรข้อมูล (Data Circuit-Terminating Equipment, DCE) แต่ต้องมีการไขว้สายสัญญาณนะหว่างอุปกรณ์ DTE ทั้งสอง



รูปที่ 12 แสดงการต่อ NULL MODEM

4.3 รูปแบบการส่ง-รับข้อมูลจากคอมพิวเตอร์กับไอซี AT89S8252 แบบ RS232



รูปที่ 13

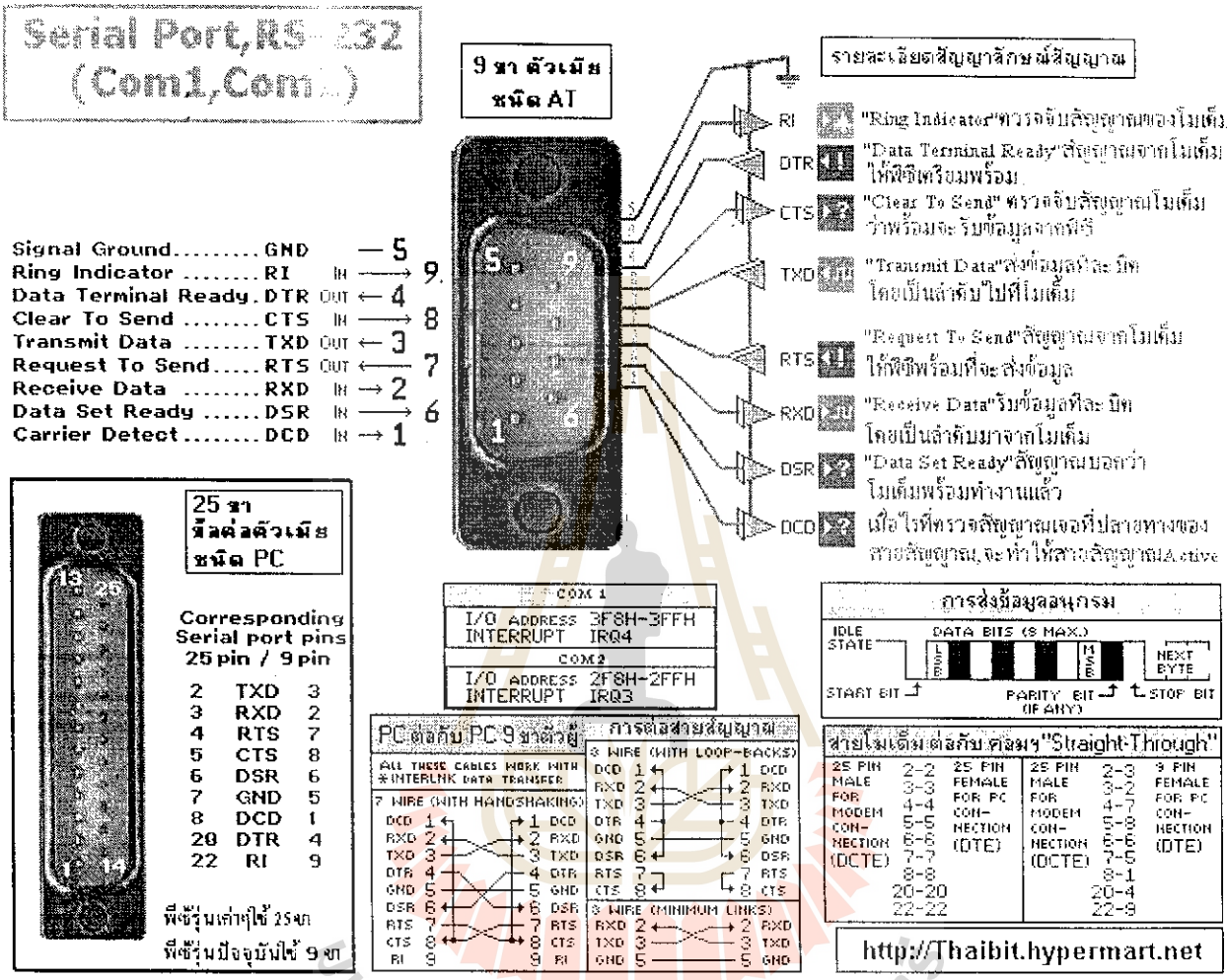
Input data ใช้ port serial ขา 3 ของ DB9

Output data ใช้ port serial ขา 2 ของ DB9

มหาวิทยาลัยเทคโนโลยีสุรนารี

ลักษณะของตัวต่อเชื่อมสัญญาณแบบ DB9

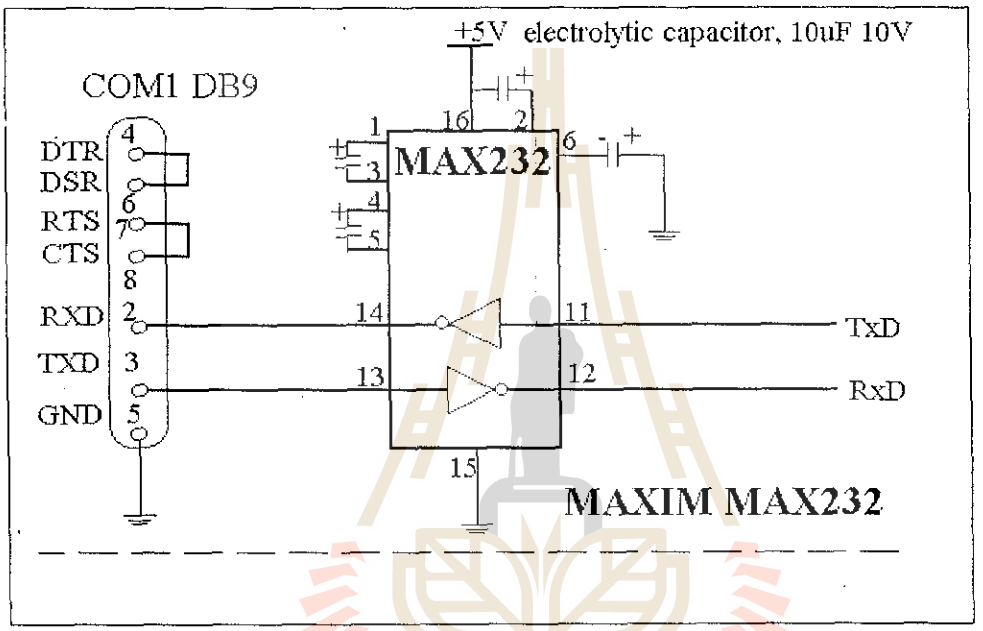
รูปที่ 14



มหาวิทยาลัยเทคโนโลยีสุรนารี

4.4 ไอซี MAX232

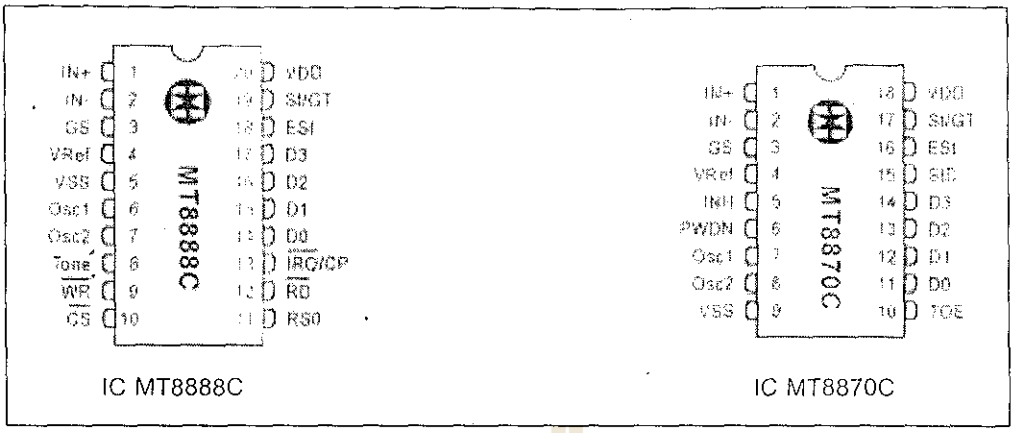
การสื่อสารแบบ RS232 โดยการใช้พอร์ต Serial ของ microcontroller และเปลี่ยนระดับสัญญาณแรงดันเป็น STANDARD RS232 โดยผ่านไอซี MAX232 เพื่อรับข้อมูลแบบ Serial จาก Computer ส่งให้ ไอซี MT89S8252 และรับข้อมูลจากไอซี MT89S8252 เพื่อส่งให้ Computer



รูปที่ 15

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.6 Data sheet ของไอซี MT8888C และ MT8870 และลักษณะการทำงาน

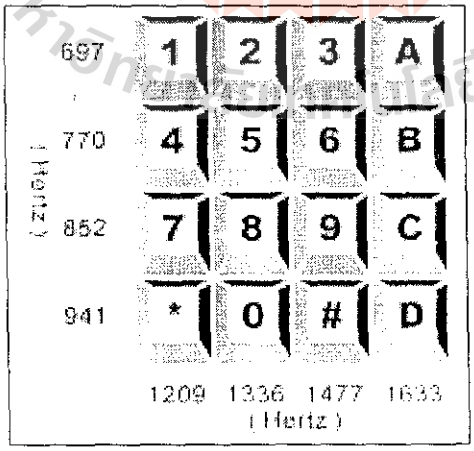


รูปที่ 17 ไอซี MT8888C และ MT8870

ไอซี MT8888C รับข้อมูลจากไอซี AT89S8252 เข้าขา D0 D1 D2 D3 ที่ขา 14 15 16 และ 17 และต่อ ไมล์ลอย FM.ที่ขา Tone ที่ขา 8

ไอซี MT8870 รับ DTMF จากการออกอากาศที่ขา IN ที่ขา 1 และส่งข้อมูลแบบบิตออกที่ขา D0 D1 D2 D3 เข้าไอซี AT89S8252 ที่กล่องฝั่งComputer และต่อเข้ารีเลย์ที่ฝั่งเครื่องใช้ไฟฟ้า

ผลจากการเข้ารหัส



รูปที่18 ผลการเข้ารหัส

ผลจากการถอดรหัส

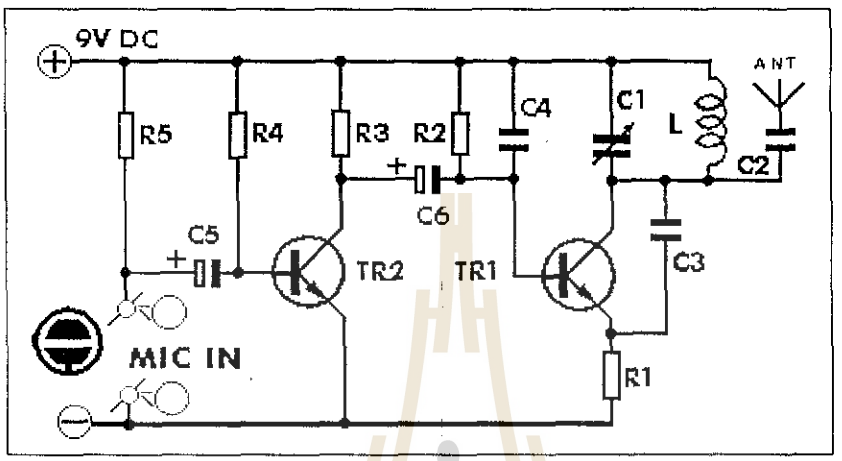
ปุ่มกด	รหัสที่ถอดได้ MT8870,MT8888C			
	D3	D2	D1	D0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
0	1	0	1	0
#	1	0	1	1
*	1	1	0	0

ตารางที่ 1

จากผลที่ได้ในตารางเมื่อไปเปรียบเทียบกับคู่มือไอซีทั้งสองตัว (MT8870 และ MT8888C) ผลที่ได้ถูกต้องไม่ผิดเพี้ยน ดังนั้นในขั้นต่อไป เราจะทำการทดสอบวงจรสร้างสัญญาณ DTMF โดยใช้ไอซี MT8888C ซึ่งในชุดวงจรนี้ต้องใช้ไมโครคอนโทรลเลอร์ร่วมด้วยโดยต้องเขียนโปรแกรมทดสอบ ซึ่งผลจากการทดลองเมื่อให้ไอซี MT8888C สร้างโทน DTMF แล้วใช้ไอซี MT8870 มาทำการถอดรหัส ผลที่ได้คือ ไอซี MT8888C สร้างโทน DTMF ได้เหมือนโทรศัพท์ทุกประการ

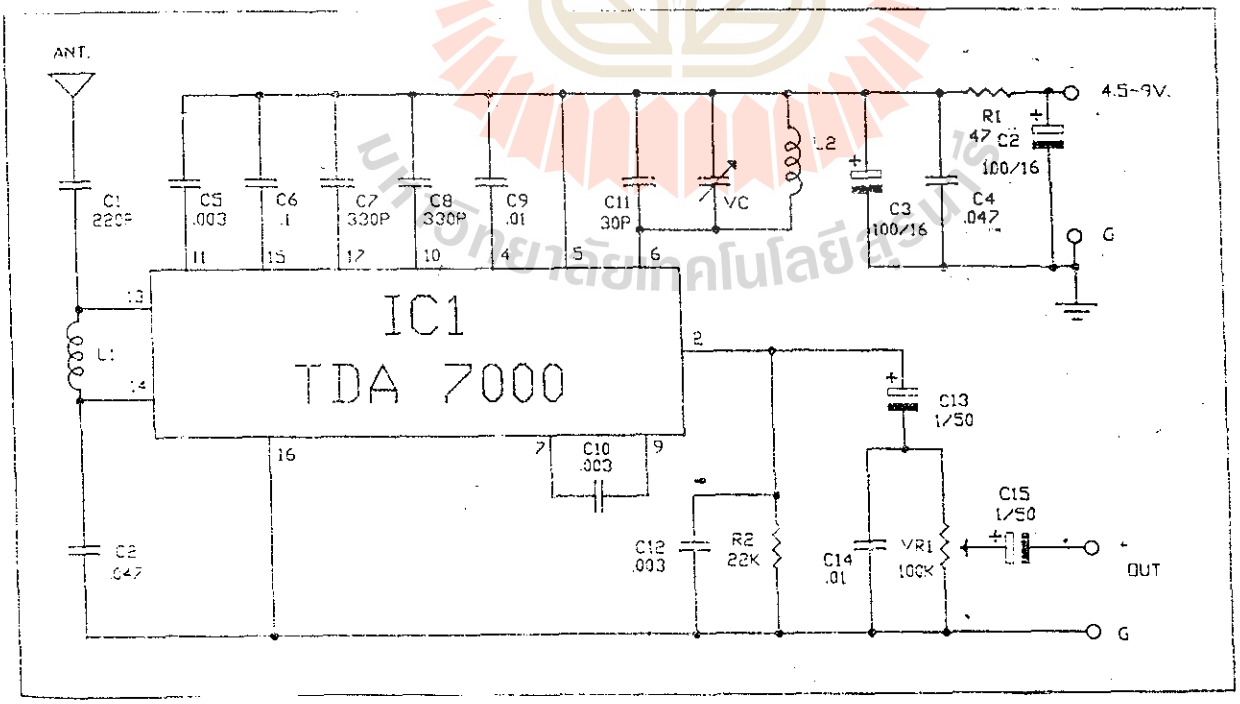
4.6 การออกอากาศ

วงจรไมค์ลอย FM. (ภาคส่ง) รับข้อมูลจากไอซี MT8888C เพื่อ modulate สัญญาณ DTMF ออกอากาศในย่าน FM



รูปที่ 19 วงจร FM 2 ทรานซิสเตอร์

ภาครับใช้วิทยุ FM เพื่อรับสัญญาณ DTMF ที่ modulate มาในย่าน FM มาเข้าไอซี MT8870 แล้วเปลี่ยนเป็น binary 4 บิต

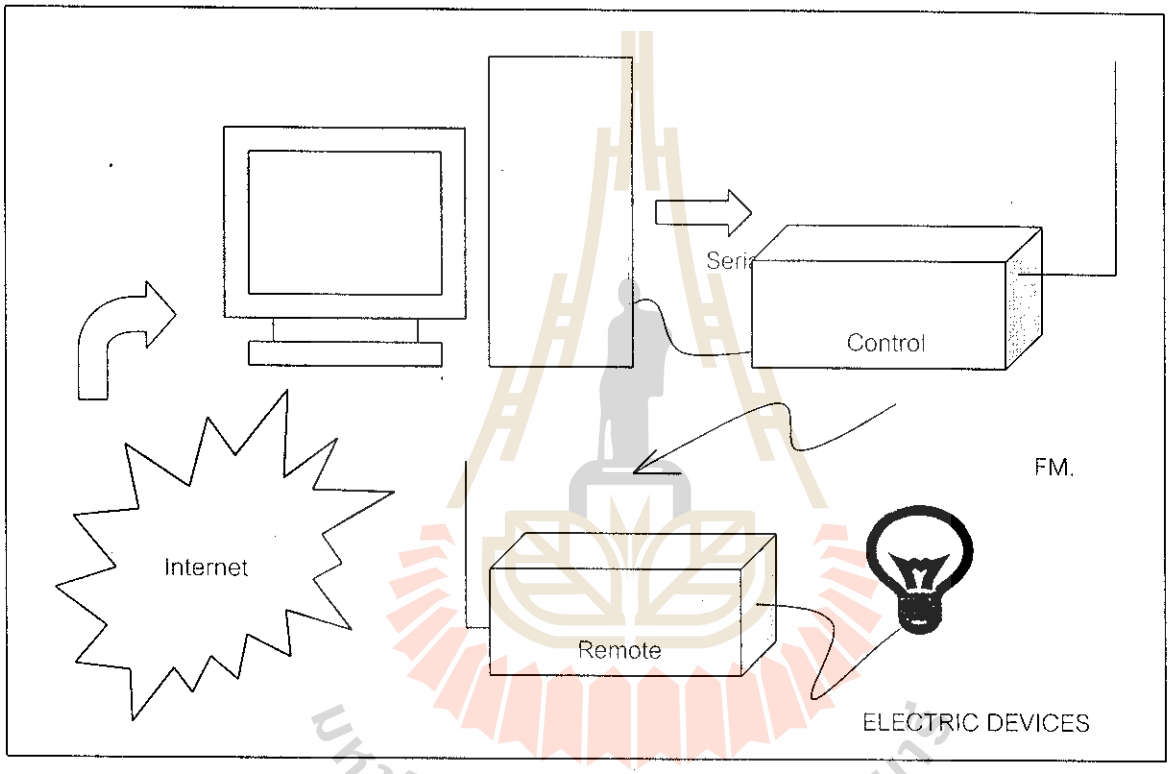


รูปที่ 20 วงจร FM ภาครับ

บทที่ 5
การใช้งาน

5.1 การต่ออุปกรณ์

นำกล่องอุปกรณ์ต่อเข้ากับคอมพิวเตอร์ โดยนำพอณีตอนุกรมต่อเข้ากับช่องพอร์ตอนุกรมของคอมพิวเตอร์แล้วนำปลายสายอีกด้านมาต่อเข้าที่อุปกรณ์ เมื่อเปิดสวิตช์จะมีแสงสว่างที่สวิตชี เพื่อเป็นการบอกว่ามีกระแสไฟฟ้าเข้ามาในกล่องอุปกรณ์ เปิดกล่องอุปกรณ์ตัวรับเช่นกันนำเครื่องใช้ไฟฟ้าเสียบเข้ากับกล่องอุปกรณ์ตัวรับ 4 ชนิด



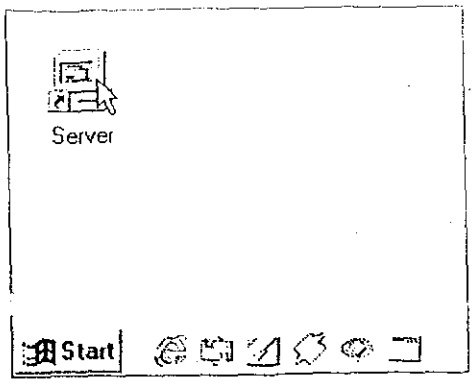
รูปที่ 21 การต่ออุปกรณ์

มหาวิทยาลัยเทคโนโลยีสุรนารี

5.2 การใช้โปรแกรมเพื่อเริ่มทำงาน

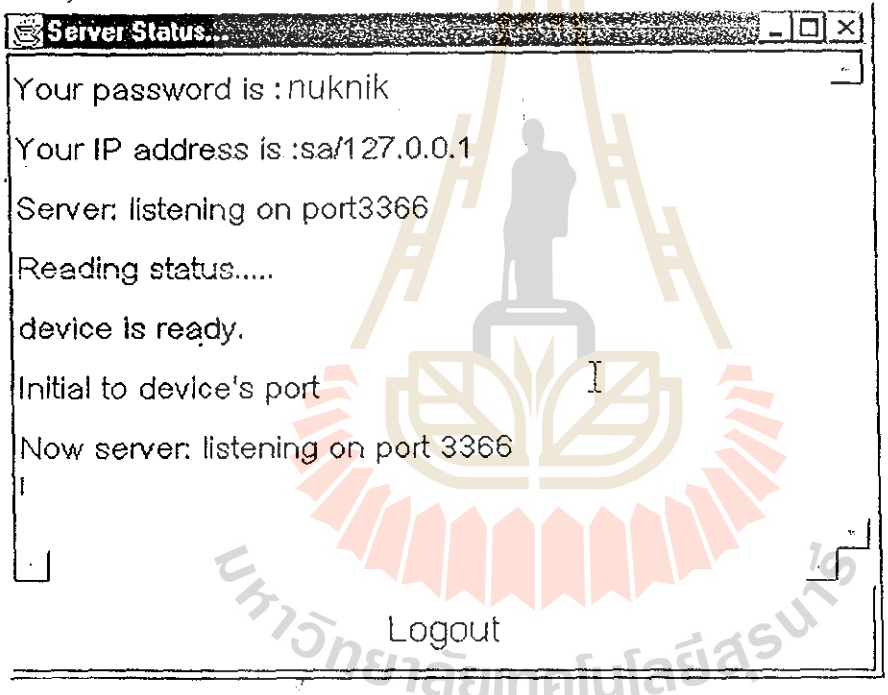
ทำการติดตั้งโปรแกรมที่ทำให้เครื่องคอมพิวเตอร์ทำงานเป็น Server ได้ แล้วนำโปรแกรมทั้งหมดใส่ในไดเรคทอรี่เดียวกัน

1. ทำการประมวลผล โดยดับเบิ้ลคลิ๊กที่ไอคอนของโปรแกรม Server ซึ่งเปิดเป็นหน้าต่างขึ้นมาดังรูปที่ 22



รูปที่ 22

2.หน้าจอของMyserver



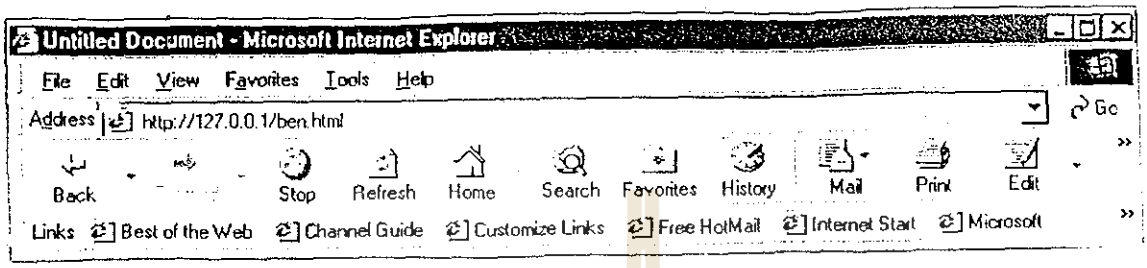
รูปที่ 23

3.เมื่อต้องการปิดโปรแกรม สามารถคลิ๊กที่ปุ่ม Logout

5.3 การใช้โปรแกรมเพื่อสั่งงานผ่านระบบอินเทอร์เนต

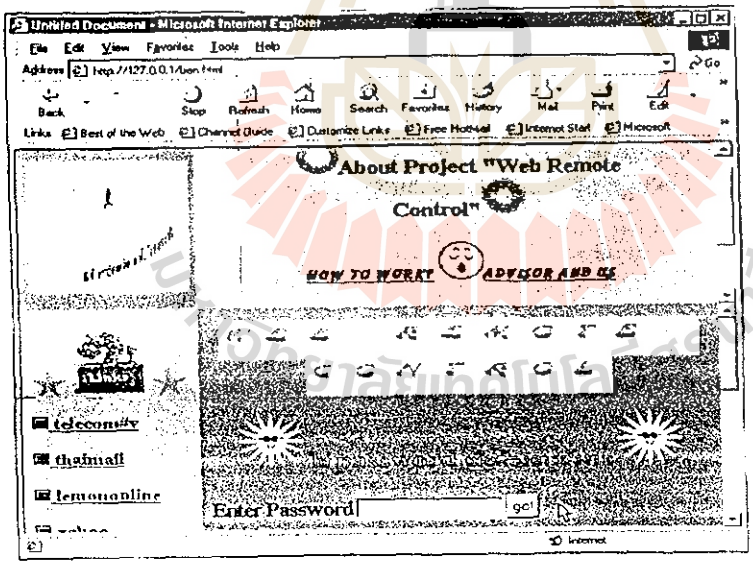
การสั่งงานผ่านระบบอินเทอร์เนตมีขั้นตอนดังนี้

1. ทำการเรียก URL ที่ได้ IP address จากโปรแกรม Server ดังรูปที่ 23



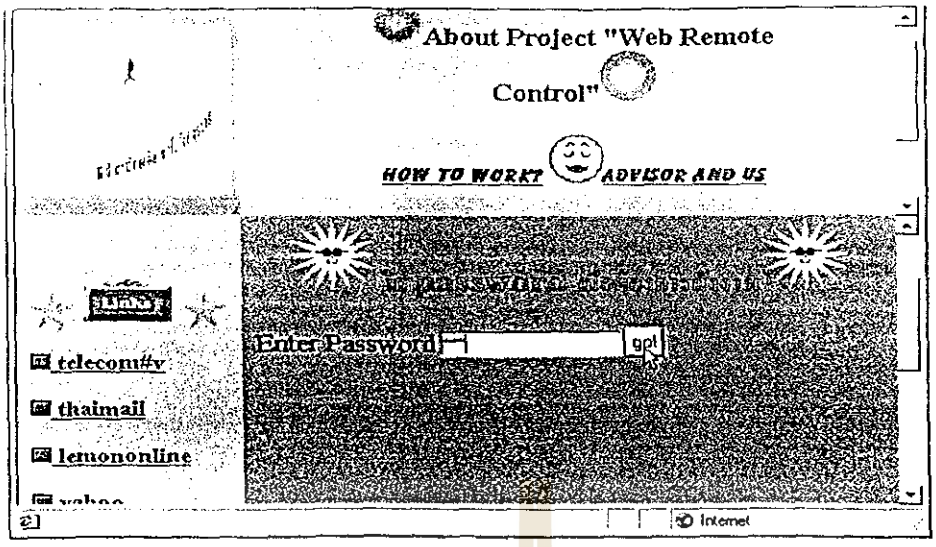
รูปที่ 24

2. เมื่อต่อกับ URL สำเร็จ Web page จะแสดงดังรูปที่ 25



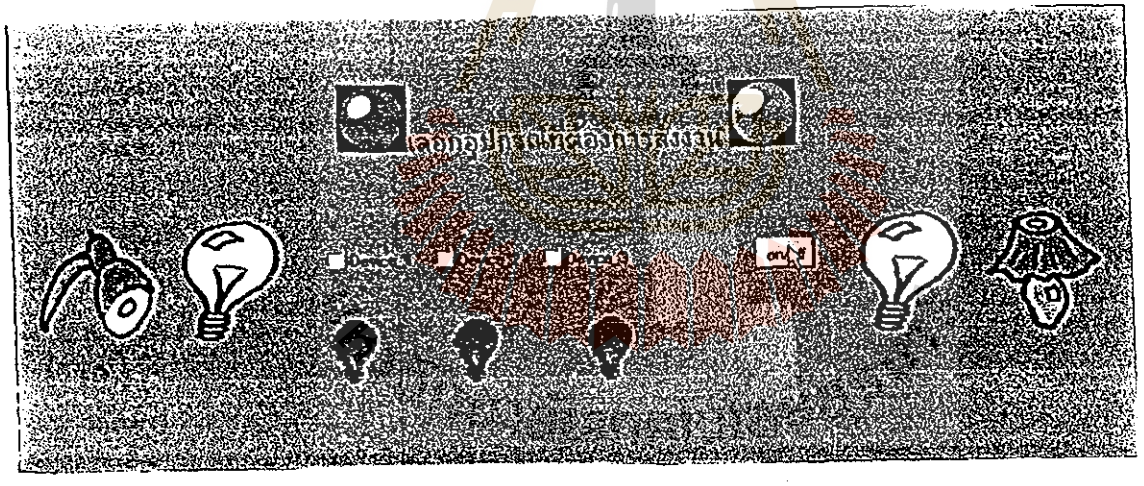
รูปที่ 25

3.ป้อน password เพื่อเข้าสู่หน้าสั่งงานดังรูปที่ 26



รูปที่ 26

4. เมื่อเข้าสู่หน้าสั่งงานจะเห็นว่าที่หน้าสั่งงานจะมีช่องที่ให้เลือกว่าอุปกรณ์ตัวไหนที่ต้องการให้ติดหรือดับ โดยเลือกที่ช่องของอุปกรณ์นั้นแล้วคลิ๊กที่ปุ่ม On/off ดังรูปที่ 26



รูปที่ 27

บทที่ 6
บทสรุป

6.1 สรุป

โครงงานการสั่งงานอุปกรณ์ไฟฟ้าแบบไร้สาย

1. เพื่อทำโครงงานต่อจากรุ่นพี่ซึ่งทำแบบอุปกรณ์อยู่ใกล้กับเครื่องคอมพิวเตอร์
2. เพื่อสามารถเปิด-ปิดอุปกรณ์ไฟฟ้าที่อยู่ไกลๆได้ โดยไม่ต้องเดินสายไฟ
3. เพื่อสามารถเขียนโปรแกรมรับ-ส่งข้อมูลแบบอนุกรม
4. สามารถส่งข้อมูลออกอากาศได้
5. สามารถรับข้อมูลจากการออกอากาศได้

6.2 ปัญหาและอุปสรรค

1. เมื่อเชื่อมต่ออุปกรณ์แล้วมักจะลืมต่ออะไรบางอย่างทำให้ระบบไม่สามารถทำงานได้ ต้องใช้เวลานานในการตรวจสอบและไล่วงจร
2. การเขียนโปรแกรมเป็นปัญหาตอนตั้งค่าหน่วงเวลาไม่ตรงกันระหว่างภาษาซีกับไมโครคอนโทรลเลอร์จึงทำให้รับค่าไม่ตรงกันและเมื่อตั้งค่าหน่วงเวลาได้แล้วเวลาไปใช้เครื่องคอมพิวเตอร์เครื่องอื่นก็เกิดปัญหาเช่นเดียวกันเพราะความเร็วของ CPU แต่ละเครื่องไม่เท่ากัน
3. การส่งข้อมูลออกอากาศซึ่งไม่เคยได้ทำมาก่อนเป็นเรื่องยากมากที่จะหาไอซีที่จะมาใช้งานและรูปแบบการออกอากาศ
4. เวลาจูนปรับความถี่ FM ยากมากที่จะปรับได้ตรงกับความถี่ที่ส่งมาจึงใช้เวลานานมากในการปรับ

มหาวิทยาลัยเทคโนโลยีสุรนารี

6.3 ข้อเสนอแนะ

การใช้งานเปิด-ปิด อุปกรณ์ไฟฟ้าไม่ควรใช้อุปกรณ์ไฟฟ้าที่กินกระแสไฟฟ้าสูงมากเกิน 5 แอมป์ เพราะใช้ฟิวส์ ขนาด 5 แอมป์ และสวิตซ์ เปิด-ปิดอุปกรณ์ไฟฟ้าที่นี้ คือ รีเลย์ที่ใช้เป็นสวิตซ์ทนกระแสได้สูงสุด 6 แอมป์ 250 V_{AC}

6.4 **แนวทางการพัฒนาในอนาคต**

1. ติดตั้งกล้องและบันทึกภาพ
2. ระบบป้องกันความปลอดภัย
3. ฯลฯ

มหาวิทยาลัยเทคโนโลยีสุรนารี

หนังสืออ้างอิง

เจนวิทย์ เหลืองอร่าม และ ปิยวิทย์ เหลืองอร่าม. (2543). การเขียนโปรแกรมสำหรับ Applications ด้วย C/C++. กรุงเทพฯ: ธรรมสาร

ดอนสัน ปงผาบ. (2543). การเขียนภาษาซีในงานควบคุม พิมพ์ครั้งที่2 กรุงเทพฯ: สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

ธนัท ชัยยุทธ. การใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก

มหาวิทยาลัยเทคโนโลยีสุรนารี

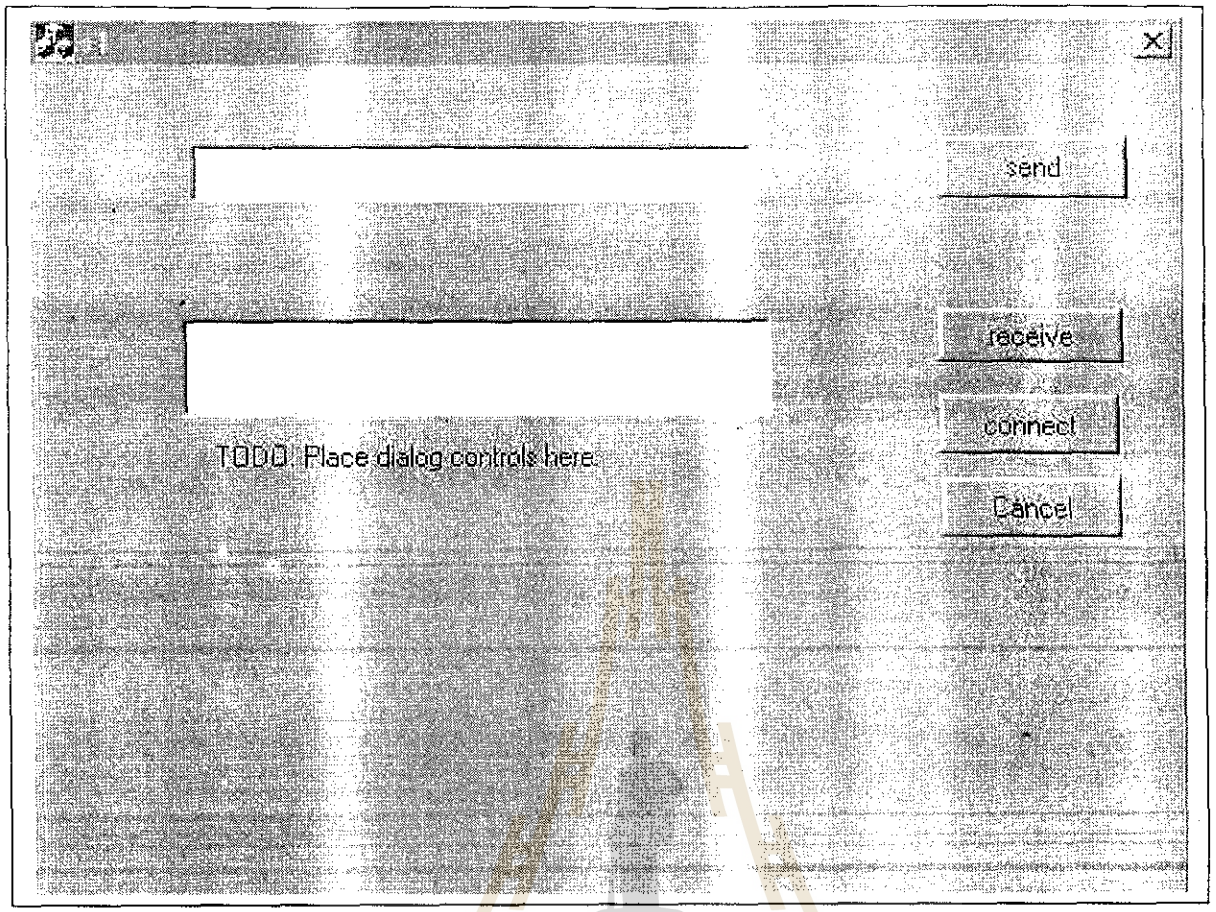
1. โปรแกรม VISUAL C++ ใช้สั่งงานผ่าน Computer

```
// s1.cpp : Defines the class behaviors for the application.
//

#include "stdafx.h"
#include "s1.h"
#include "s1Dlg.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CS1App

BEGIN_MESSAGE_MAP(CS1App, CWinApp)
        //{{AFX_MSG_MAP(CS1App)
                // NOTE - the ClassWizard will add and remove mapping macros here.
                //    DO NOT EDIT what you see in these blocks of generated code!
        //}}AFX_MSG
        ON_COMMAND(ID_HELP, CWinApp::OnHelp)
END_MESSAGE_MAP()

/////////////////////////////////////////////////////////////////////////////
// CS1App construction

CS1App::CS1App()
{
        // TODO: add construction code here,
        // Place all significant initialization in InitInstance
}

/////////////////////////////////////////////////////////////////////////////
// The one and only CS1App object

CS1App theApp;
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
//////////////////////////////////////////////////////////////////////
// CS1App initialization

BOOL CS1App::InitInstance()
{
	AfxEnableControlContainer();

	// Standard initialization
	// If you are not using these features and wish to reduce the size
	//  of your final executable, you should remove from the following
	//  the specific initialization routines you do not need.

#ifdef _AFXDLL
	Enable3dControls();			// Call this when using MFC in a shared DLL
#else
	Enable3dControlsStatic(); // Call this when linking to MFC statically
#endif

	CS1Dlg dlg;
	m_pMainWnd = &dlg;
	int nResponse = dlg.DoModal();
	if (nResponse == IDOK)
	{
		// TODO: Place code here to handle when the dialog is
		//  dismissed with OK
	}
	else if (nResponse == IDCANCEL)
	{
		// TODO: Place code here to handle when the dialog is
		//  dismissed with Cancel
	}

	// Since the dialog has been closed, return FALSE so that we exit the
	//  application, rather than start the application's message pump.
	return FALSE;
}
```

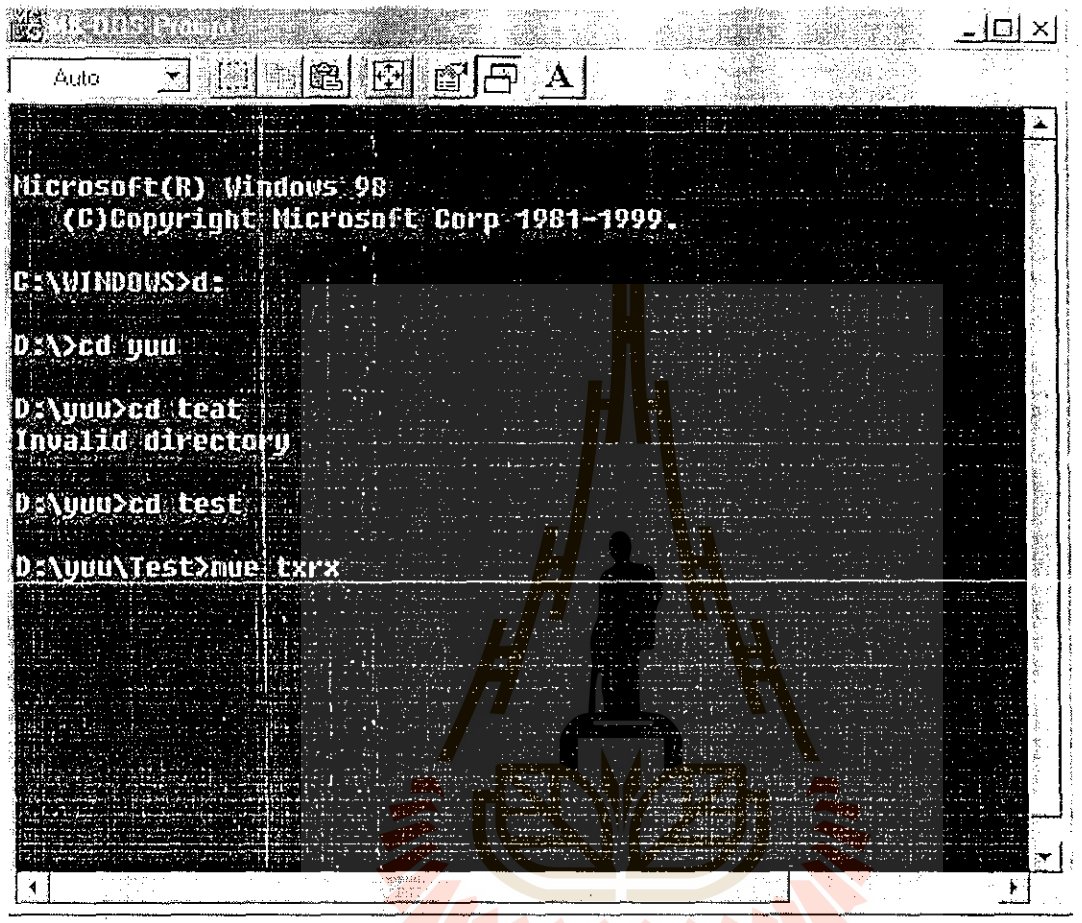
มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปแสดงหลังจาก Compile

มหาวิทยาลัยเทคโนโลยีสุรนารี

2. วิธีCompile โปรแกรม Microcontroller



มหาวิทยาลัยเทคโนโลยีสุรนารี

เข้า ใน DOS แล้วเข้าไปใน DIRECTORY ของโปรแกรมที่เรา SAVE อยู่ แล้วพิมพ์ ว่า MUE แล้วตามด้วยชื่อโปรแกรม *. ASM

3. ข้อมูลใน Fileที่ใช้ Compile

```
sxa %1.asm
pause
s8252v2.exe %1.hex -a -9
```

เป็นFILE *.bat เพื่อใช้ Compile โปรแกรม microcontroller ใน DOS

มหาวิทยาลัยเทคโนโลยีสุรนารี

4. โปรแกรม microcontroller

ฝั่งตัวส่ง

```
        ORG   0000H                ; กำนหนดค่าเริ่มต้น
        LJMP   MAIN                ; reset vecter

MAIN:
        MOV   SCON,#52H    ; กำหนดโหมดในการส่งข้อมูลแบบ serial port
        MOV   PCON,#00H    ; กำหนดโหมดในการส่งข้อมูลไม่เป็นแบบ parallell port
        MOV   TMOD,#20H    ; กำหนด Timer 1 mode 2
        MOV   A,#0FDH      ; กำหนด baud rate = 9600 bps
        MOV   TH1,A                ; ต้งค่าการรับข้อมูล
;       MOV   TL1,A        ; ตั้งค่าการรับข้อมูล
        CLR    ET1                 ; clear timer1 interrupt
      SETB    TR1                  ; กำหนดให้ Timer1 ทำงาน
      CLR     ES
      CLR     EA

        MOV   P2,#00H      ; กำหนดค่าเริ่มต้น port 2 เป็น 0 หมด
        MOV   P1,#0FFH     ; กำหนดค่าเริ่มต้น port 1 เป็น 1 หมด
        MOV   R0,#45H
        MOV   @R0,P1               ; เก็บค่า port 1 ไว้ใน R0
LOOP:
        ACALL CHECK                ; ตรวจสอบการเปลี่ยนสถานะจาก port
        MOV   R3,#0AH
REP:
        ACALL DELAY
        DJNZ  R3,REP
        SJMP  LOOP                 ; วนซำไปเรื่อยๆ
CHECK:
        ACALL DELAY                ; หน่วงเวลา 3 วินาที
        MOV   A,P1         ; ย้ายค่า port 1 ไป A
        ACALL BLE
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
        MOV    A,#00H
        ACALL TRANS
        RET                    ; กลับไปใน LOOP
BLE:
        ACALL TRANS
        ACALL DELAY
        ACALL TRANS
        ACALL DELAY
        ACALL TRANS
        ACALL DELAY
        ACALL TRANS
        ACALL DELAY
        ACALL TRANS
        ACALL DELAY
      RET
TRANS:
        PUSH        IE
      CLR     TI
      MOV     SBUF,A       ; ย้าย A ไป Sbuf เพื่อเตรียมส่ง
        JNB     TI,$         ; เปรียบเทียบจนกว่าข้อมูลส่งครบไบต์
        CLR     TI           ; เคลียร์ TI ใน SCON register
        MOV     @R0,P1       ; เก็บค่าสถานะ port 1 ปัจจุบันไว้ใน R0
      POP     IE
      RET                    ; กลับไปใน CHECK
DELAY:
        MOV     R6,#00H
        MOV     R7,#00H
DLY00: DJNZ   R6,DLY00
        DJNZ   R7,DLY00
        RET
        END
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

ฝั่งตัวรับ

```
ORG   0000H                ;กำหนดค่าเริ่มต้น
      LJMP   MAIN              ;reset vecter
       ORG  0100H
MAIN:
      CLR    EA
      MOV    P2,#00H       ; กำหนดค่าเริ่มต้น port 2 เป็น 0 หมด เพื่อส่งค่าให้ 8888
      MOV    P1,#0FFH      ; กำหนดค่าเริ่มต้น port 1 เป็น 1 หมด เพื่อรับค่าจาก
อุปกรณ์
      MOV    R0,P1                ; เก็บค่า port 1 ไว้ใน R0
LOOP:
      MOV    A,P1          ; รับค่าสถานะอุปกรณ์จาก port 1ไว้ใน A
      MOV    R0,A                 ; นำค่าในสถานะ A ไปไว้ใน R0 เพื่อส่งให้ 8888
      ACALL S8888          ; ส่งค่าสถานะอุปกรณ์ไปให้ 8888
      SJMP   LOOP                 ; วนซ้ำไปเรื่อย ๆ
S8888:
      MOV    SP,#30H
      MOV    P2,#0EFH
      ACALL DELAY
      MOV    R3,#00H
      MOV    A,#10101101B
      ACALL DATA_OUT
      MOV    A,#10100000B
      ACALL DATA_OUT
AGAIN:
      DEC    R3
      MOV    A,#2FH
      ANL    A,R0
      MOV    P2,#0E0H
      CALL   DATA_OUT
      CJNE   R3,#00H,AGAIN
      RET
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
DATA_OUT:
        MOV   P2,A
        CALL  DLY00
        MOV   P2,#0EFH
        CALL  DLY00
        RET
DELAY:
        MOV   R6,#00H
        MOV   R7,#00H
DLY00:  DJNZ  R6,DLY00
        DJNZ  R7,DLY00
        RET
        END
```

5. โปรแกรมไมโครคอนโทรลเลอร์ที่ใช้ทดสอบต่าง ๆในการทำงาน

โปรแกรมที่ใช้ส่งค่า ๆเดียวเพื่อทดสอบวงจร Tuner FM

```
ORG   0000H              ;กำนหดค่าเริ่มต้น
      LJMP   MAIN              ;reset vecter

MAIN:
      MOV   P2,#00H      ; กำหนดค่าเริ่มต้น port 2 เป็น 0 หมด
      MOV   P1,#0FFH     ; กำหนดค่าเริ่มต้น port 1 เป็น 1 หมด
      MOV   R0,#45H
      MOV   @R0,P1             : เก็บค่า port 1 ไว้ใน R0

;START MT8888
      MOV   R2,#65H
      CLR   EA
      MOV   SP,#30H
      MOV   P2,#0EFH
      CALL  DELAY

      MOV   A,#10101101B
      CALL  DATA_OUT

      MOV   A,#10100000B
      CALL  DATA_OUT

REPEAT:
      MOV   A,#2FH
      ANL   A,R2
      CALL  DATA_OUT
      SJMP  REPEAT

;END SEND MT8888

DATA_OUT:
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
        MOV   P2,A
        CALL  DELAY
        MOV   P2,#0EFH
        CALL  DELAY
        RET

DELAY:
        MOV    R6,#00H
        MOV   R7,#00H
DLY00:  DJNZ  R6,DLY00
        DJNZ  R7,DLY00
        RET
        END
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

โปรแกรมทดสอบการส่งข้อมูลแบบ Serial

```
;*****************************************************
;*          Program Test Serial Port                 *
;*          Hardware : CP-S8252  V2.0                *
;*          Complier : SXA51                         *
;*****************************************************
        ORG     0000H
        LJMP    MAIN                    ;reset vecter
        ;
MAIN:     MOV     SP,#256-32                ;define stack = 32 byte
        ;
INIT_SER:  MOV     TMOD,#00100000B                ;time1 mode2
        MOV     SCON,#01010000B              ;mode1 serial port
        MOV     TH1,#0FBH
        MOV     A,#00H
        MOV     PCON,A                    ;SMOD = 0
        CLR     ET1                     ;clear timer1 interrupt
        SETB    TR1                     ;start timer1
        CLR     ES
        CLR     EA
        ;
SER:      MOV     DPTR,#TEXT_TAB
        MOV     A,#0
SER1:      PUSH    ACC
        MOVC    A,@A+DPTR
        CJNE    A,#00H,TX1
        POP     ACC
        SJMP    TX2
TX1:      LCALL   TX_BYTE
        POP     ACC
        INC     A
        SJMP    SER1
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
TX2:    LCALL  RX_BYTE
        MOV    DPTR,#TX_TAB
        MOV    A,#00H
TX4:    PUSH   ACC
        MOVC   A,@A+DPTR
        CJNE   A,#00H,TX3
        LJMP   TX2
        POP    ACC
TX3:    LCALL  TX_BYTE
        POP    ACC
        INC    A
        SJMP   TX4

;*********************************************************
;*          Send 1-Byte to SERIAL                        *
;*          Input  : ACC                                 *
;*          Output : Serial port                         *
;*********************************************************
TX_BYTE:  PUSH  IE
        CLR   TI
        MOV   SBUF,A
        JNB   TI,$
        CLR   TI
        POP   IE
        RET

;*********************************************************
;*          Receive Data From SERIAL                     *
;*          Input  : Serial Port                         *
;*          Output : ACC                                 *
;*********************************************************
RX_BYTE:  PUSH  IE
        JNB   RI,$          ; Wait data
        CLR   RI
```

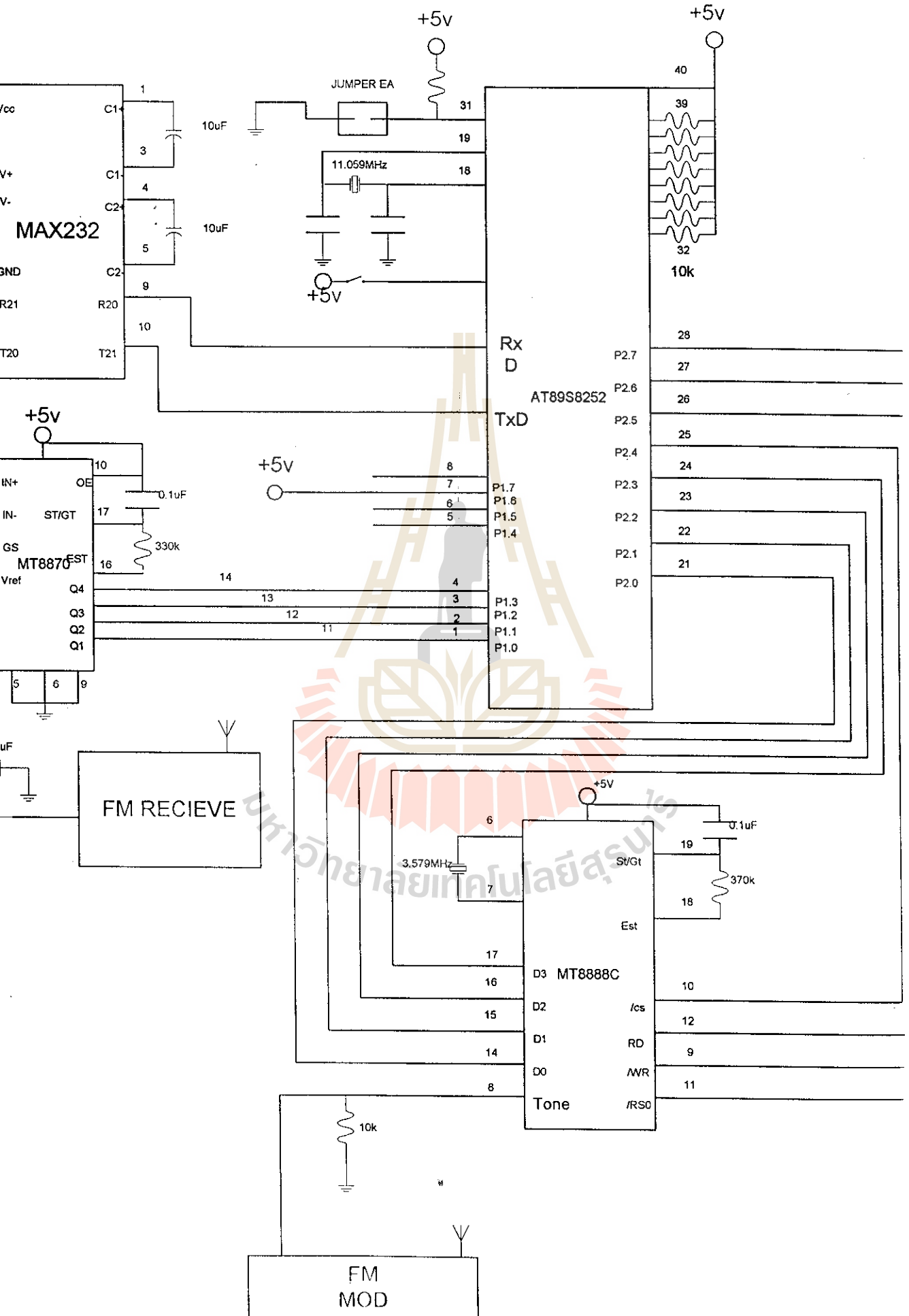
มหาวิทยาลัยเทคโนโลยีสุรนารี

```
        MOV     A,SBUF
        POP     IE
        RET


TEXT_TAB:   DB     0CH,'TEST SERIAL PORT (RS232)',0AH,0DH
        DB     ''CP-S8252 (AT89S8252)',0AH,0DH
        DB     'PRESS ANY KEY',0AH,0DH,0AH,0DH,00H

TX_TAB:     DB     'CP-S8252 TX WORD!',0AH,0DH
        DB     '"SYSTEMS COMPLETE"',0AH,0DH
        DB     'THANK YOU VERY MUCH',0AH,0DH,0AH,0DH,00H

        END
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

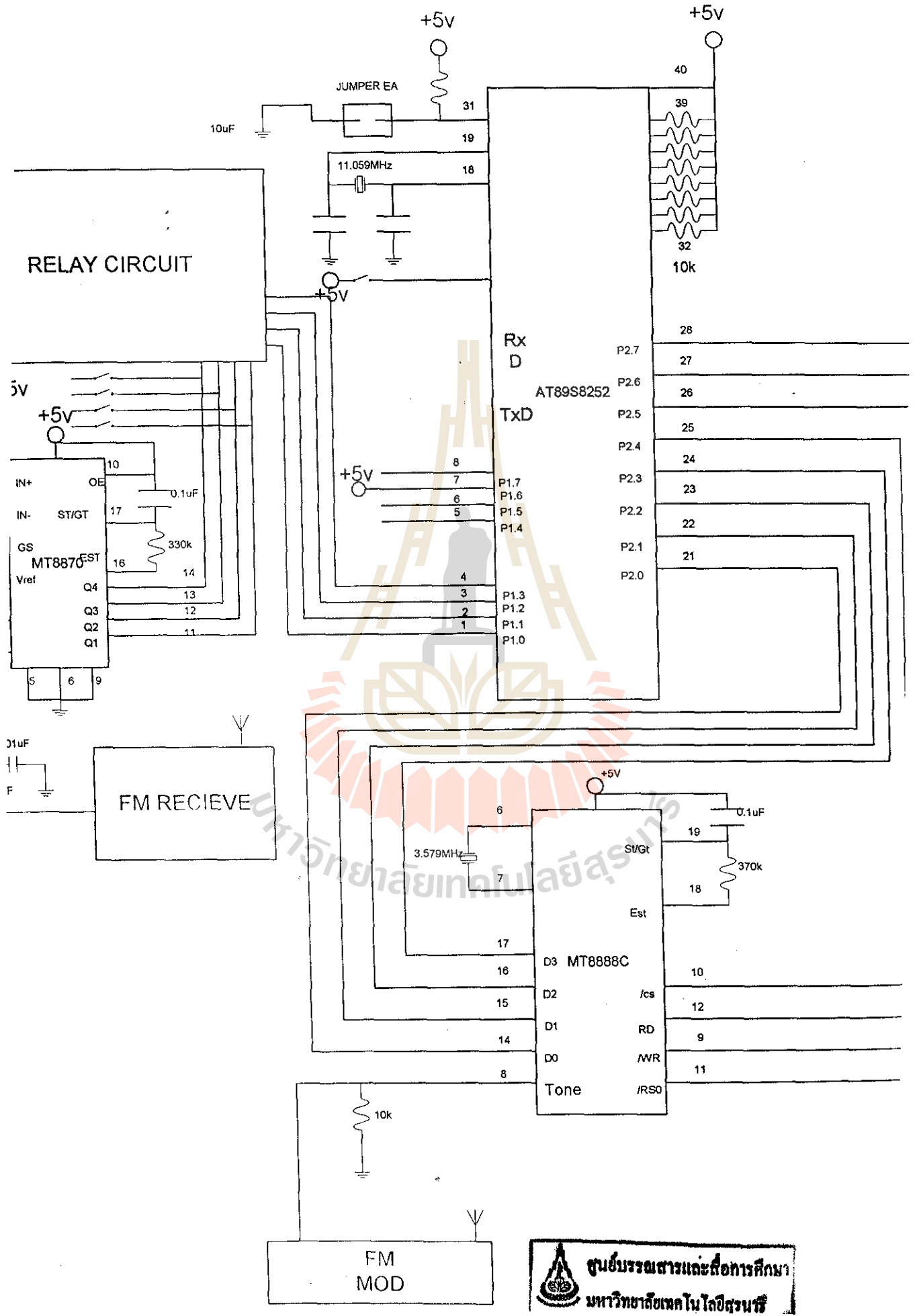
6. วงจรตัวส่งและรับข้อมูล

มหาวิทยาลัยเทคโนโลยีสุรนารี


+5v
+5v
JUMPER EA
10uF
11.059MHz
10uF
+5v
MAX232
Vcc
V+
V-
GND
R21
T20
C1+
C1-
C2+
C2-
R20
T21
10k
Rx
D
TxD
AT89S8252
P2.7
P2.6
P2.5
P2.4
P2.3
P2.2
P2.1
P2.0
P1.7
P1.6
P1.5
P1.4
P1.3
P1.2
P1.1
P1.0
+5v
0.1uF
330k
MT8870
IN+
IN-
GS
Vref
OE
ST/GT
EST
Q4
Q3
Q2
Q1
FM RECIEVE
+5V
3.579MHz
0.1uF
370k
St/Gt
Est
MT8888C
D3
D2
D1
D0
Tone
/cs
RD
/WR
/RS0
10k
FM
MOD

ตัวรับ

มหาวิทยาลัยเทคโนโลยีสุรนารี



มหาวิทยาลัยเทคโนโลยีสุรนารี
ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี