

ยานพาหนะหุ่นยนต์ควบคุมโดยคอมพิวเตอร์โมเด็มไร้สาย

VEHICLE ROBOT BY USING COMPUTER MODEM CONTROL VIA WIRELESS



เลขที่.....
เลขทะเบียน..... 42310
วัน, เดือน, ปี..... 16 พ.ค. 2545

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

611203710

**VEHICLE ROBOT BY USING COMPUTER MODEM
CONTROL VIA WIRELESS**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF THE TECHNOLOGY TELECOMMUNICATIONS
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์

ยานพาหนะหุ่นยนต์ควบคุมโดยคอมพิวเตอร์โมเด็มไร้สาย
VEHICLE ROBOT BY USING COMPUTER MODEM
CONTROL VIA WIRELESS

นักศึกษา

นายประสงค์ ตรีรักษา เลขประจำตัว 42015641
นายวัชรินทร์ นำภา เลขประจำตัว 42015654

อาจารย์ที่ปรึกษา

รองศาสตราจารย์ ดร. กนก เจนจิระพงศ์เวช
รองศาสตราจารย์ ขวลิต เบญจางคประเสริฐ

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2544

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้
นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

อาจารย์ที่ปรึกษา

(รศ.ดร.กนก เจนจิระพงศ์เวช)

อาจารย์ที่ปรึกษา

(รศ.ขวลิต เบญจางคประเสริฐ)

กรรมการ

()

กรรมการ

()

กรรมการ

()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ยานพาหนะหุ่นยนต์ ควบคุมโดยคอมพิวเตอร์โมเด็มไร้สาย		
นักศึกษา	นายประสงค์	ดรรักษา	รหัสประจำตัว 42015641
	นายวัชรินทร์	นำภา	รหัสประจำตัว 42015654
อาจารย์ที่ปรึกษา	รศ.ดร. กนก	เจนจิระพงศ์เวช	
	รศ. ชวลิต	เบญจางคประเสริฐ	
ภาควิชา	เทคนิคอุตสาหกรรม		
ปีการศึกษา	2544		

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอ วิธีการสร้างเครื่องรับส่งข้อมูลระหว่าง คอมพิวเตอร์กับ อุปกรณ์ประยุกต์ใช้งานภายนอก โดยใช้ข่าวสารจากคอมพิวเตอร์ไปควบคุมหุ่นยนต์ ด้วยอัตราเร็ว ในการส่งข้อมูล 1200บิตต่อวินาที ผ่านพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ ผ่านการเข้ารหัสแบบ เฟสเอสเค และทำการมอดูเลต แบบเอเอ็ม ก่อนนำสัญญาณข่าวสารส่งออกอากาศ ที่เครื่องรับจะทำการดีเทคสัญญาณ ได้เป็นสัญญาณ ข่าวสารข้อมูล ส่งผ่านวงจรกรองความถี่ และผ่านเข้าวงจรเฟส ล็อกกลุ๊ป เพื่อตรวจสอบสถานะของอีกด้านหนึ่ง โดยมีไมโครคอนโทรลเลอร์เป็นตัวตรวจเช็คสถานะ ของสัญญาณ ขณะเดียวกันก็นำเอาสัญญาณส่วนหนึ่งจากวงจรกรองความถี่ ไปทำการถอดรหัส เฟส เอสเคให้เป็นข้อมูลข่าวสารแบบดิจิตอลก่อนส่งให้ไมโครคอนโทรลเลอร์ประมวลผลและใช้ควบคุมอุปกรณ์อื่นๆ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title VEHICLE ROBOT BY USING COMPUTER MODEM
CONTROL VIA WIRELESS

Student Mr. Prasong Donruksa ID 42015641

Mr. Watcharin Numpa ID 42015654

Advisor Assoc.Prof. Dr.Knok Janchitrapongvej

Assoc.Prof. Chawalit Benjangkprasert

Academic Year 2001

ABSTRACT

This thesis presents a method of implementation data transmission between computer and robot via wireless. The signal from computer is sent to receiver with baudrate of 1200-bit per second to serial port and then coded by frequency shift keying. Before sending the information signal, the signal then modulated by AM type and send out by half duplex transmission to robot. The receiver detects information with filter and phase lock loop circuit. Where microcontroller is used to investigate the operation status of the other side. Microcontroller is checked the state of signal and convey the signal from filter output circuit to decoding the digital signal for FSK. Finally, Digital signal is sent to microcontroller for processing and used to control other equipments.

กิตติกรรมประกาศ

ปริญญาานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ ช่วยเหลือจากบุคคลหลายๆ ฝ่าย ดังนั้นจึงขอแสดงความขอบคุณทุกท่าน ดังมีรายนามต่อไปนี้

กราบขอบพระคุณ รศ.ดร. กนก เจนจิระพงษ์เวช และรศ.ชวลิต เบญจางคประเสริฐ อาจารย์ที่ปรึกษาปริญญาานิพนธ์ ผู้ที่ให้ทั้งความรู้ ความช่วยเหลือ คอยดูแล ชี้แนะส่งเสริมรวมทั้งให้หลักการและข้อคิดต่างๆ อันเป็นประโยชน์อย่างยิ่ง นับตั้งแต่เริ่มต้นปริญญาานิพนธ์ ตลอดมาจนกระทั่งจบปริญญาานิพนธ์ ทำให้ปริญญาานิพนธ์ครั้งนี้สำเร็จลุล่วงไปได้ด้วยดี

กราบขอบพระคุณ ผศ.อุทัย ศรีธีระวิโรจน์ หัวหน้าภาควิชาเทคนิคอุตสาหกรรม ที่ให้ความช่วยเหลือ อุปกรณ์และเครื่องมือ รวมทั้งให้สถานที่ในการทำปริญญาานิพนธ์ครั้งนี้ อันเป็นประโยชน์ต่อปริญญาานิพนธ์

กราบขอบพระคุณ อาจารย์ ภูงศ์ หงษ์สุวรรณ ที่ให้คำปรึกษาเกี่ยวกับ โปรแกรมภาษา Visual Basic ที่ใช้ในการควบคุมการติดต่อสื่อสาร

ขอขอบคุณเพื่อนๆ ที่คอยให้กำลังใจเป็นอย่างดีมาก

สุดท้ายนี้ขอกราบขอบพระคุณ คุณพ่อคุณแม่ และครูบาอาจารย์ทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้ และให้กำลังใจตลอดมา

นายประสงค์ ตรีรักษา

นายวัชรินทร์ น้าภา

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูปภาพ	ช
สารบัญตาราง	ฉ
บทที่ 1 บทนำ	
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่ได้รับ	2
บทที่ 2 ทฤษฎี	
2.1 บทนำ	3
2.2 การสื่อสารข้อมูล (Data Communication)	3
2.3 การส่งข้อมูลแบบซิงโครนัส (Synchronous Transmission)	5
2.4 การส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Transmission)	6
2.5 บิตเริ่มต้น (Start Bit)	7
2.6 บิตข้อมูล (Data bit)	7
2.7 บิตพาริตี (Parity bit)	8
2.8 บิตสิ้นสุดข้อมูล (Stop bit)	8
2.9 ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม	9
2.10 การมอดูเลตทางความถี่	9
2.11 ดัชนีการมอดูเลต	10
2.12 การมอดูเลตดิจิตอลทางความถี่ (Frequency Shift Keying; FSK)	10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
2.13 พอร์ท RS-232C	11
2.14 การเชื่อมต่อคอมพิวเตอร์กับคอมพิวเตอร์โดยตรง (Null Modem)	15
2.15 การสื่อสารข้อมูลแบบอะซิงโครนัส ของไมโครคอนโทรลเลอร์	17
2.16 การติดต่อกับวีซวลเบสิก	23
บทที่ 3 หลักการทำงานและการออกแบบวงจร	
3.1 บทนำ	29
3.2 การออกแบบส่วนส่งข้อมูล	30
3.3 การออกแบบส่วนรับข้อมูล	36
3.4 หลักการทำงานของวงจร LPT	40
3.5 หลักการทำงานของวงจรถ็กลงสัญญาณ SYNC	41
3.6 วงจรขับมอเตอร์	42
3.7 วงจรไมโครคอนโทรลเลอร์ทางด้านรับ	43
3.8 หลักการทำงานของวงจรถ็กลงส่ง	44
3.9 หลักการทำงานของวงจรถ็กลงรับ	45
3.10 ส่วนของเซนเซอร์ระยะทางการเคลื่อนที่	46
บทที่ 4 ผลการทดลอง	
4.1 บทนำ	47
4.2 สัญญาณดิจิทัลที่ต้องการส่ง กับสัญญาณเอาต์พุตของวงจร เอฟเอสเคมอดคูเลทเตอร์ที่ด้านส่ง	47
4.3 สัญญาณที่ออกจากวงจรเอฟเอสเคมอดคูเลทเตอร์กับสัญญาณที่ผ่านวงจรขยายแล้วทางด้านส่ง	48
4.4 สัญญาณที่ภาครับดีเทคได้ กับ สัญญาณที่ผ่านวงจรกรองความถี่ต่ำ	49
4.5 สัญญาณที่ผ่านวงจรกรองความถี่ต่ำ กับสัญญาณที่เอาต์พุตของ วงจรเอฟเอสเคมอดคูเลทเตอร์ที่ ด้านรับ	50
4.6 สัญญาณจากวงจร เอฟเอสเคมอดคูเลทเตอร์ กับ สัญญาณเอาต์พุตอินเวอร์เตอร์	51
4.7 สัญญาณอินพุตที่เครื่องส่ง กับสัญญาณเอาต์พุตที่เครื่องรับ	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
4.8 ตัวอย่างการออกแบบวงจรที่ใช้ในการทดลอง	53
บทที่ 5 สรุปผลการทดลอง	
5.1 สรุปผลการทดลอง	55
5.2 ปัญหาและแนวทางการแก้ไข	55
5.3 การพัฒนาโครงการ	56



สารบัญรูปภาพ

	หน้า
บทที่ 2 ทฤษฎี	
รูปที่ 2.1 แสดง การส่งข้อมูลแบบขนาน	4
รูปที่ 2.2 แสดง การส่งข้อมูลแบบอนุกรม	4
รูปที่ 2.3 แสดงการส่งข้อมูลแบบซิงโครนัส	5
รูปที่ 2.4 แสดงข้อมูลแบบซิงโครนัส	6
รูปที่ 2.5 แสดงรูปแบบข้อมูลแบบอะซิงโครนัส	7
รูปที่ 2.6 แสดงการมอดูเลตคิติดอลทางความถี่	11
รูปที่ 2.7 แสดงการใช้ RS-232C เชื่อมต่ออุปกรณ์นำ	11
รูปที่ 2.8 แสดง การต่อ RS – 232C ระหว่างคอมพิวเตอร์อย่างง่าย ๆ	16
รูปที่ 2.9 แสดง การต่อคอมพิวเตอร์ผ่าน RS –232C แบบมี Hand Shake	16
รูปที่ 2.10 แสดงรูปแบบข้อมูลอนุกรมแบบอะซิงโครนัส	17
รูปที่ 2.11 แสดงหน้าต่างการทำงานของโปรแกรมวิชวลเบสิก	24
บทที่ 3 หลักการทำงานและการออกแบบ	
รูปที่ 3.1 บล็อกไดอะแกรมของ โครงงานด้านส่ง	29
รูปที่ 3.2 บล็อกไดอะแกรมของ โครงงานด้านรับ	29
รูปที่ 3.3 วงจรแปลงระดับสัญญาณ RS-232C	31
รูปที่ 3.4 บล็อกไดอะแกรมของ XR2206	33
รูปที่ 3.5 วงจรกำเนิดสัญญาณฟรีควีนซีพียอ์	35
รูปที่ 3.6 วงจรมอดูเลตแบบฟรีควีนซีพียอ์	36
รูปที่ 3.7 บล็อกไดอะแกรมของ XR2211	37
รูปที่ 3.8 การต่อใช้งานวงจรมอดูเลเตอร์แบบเอฟเอสเค	38
รูปที่ 3.9 วงจรดีมอดูเลเตอร์แบบฟรีควีนซีพียอ์	40
รูปที่ 3.10 วงจรการทำงาน Filter	41
รูปที่ 3.11 วงจรเช็คสัญญาณ Sync	42
รูปที่ 3.12 วงจรขับมอเตอร์	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

บทที่ 2 ทฤษฎี

ตารางที่ 2.1	แสดงจำนวนบิตที่ขึ้นอยู่กับจำนวนคาร์แรกเตอร์	8
ตารางที่ 2.2	แสดงย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS-232	12
ตารางที่ 2.3	แสดงขาที่ใช้งานของ DB-9 และ DB-25 ในการรับส่งข้อมูลแบบอนุกรม	15
ตารางที่ 2.4	ใช้ในการเลือกโหมดในการเลือกพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51	19
ตารางที่ 2.5	ค่าคงที่สำหรับคุณสมบัติ Handshake	26
ตารางที่ 2.6	ค่าคงที่สำหรับคุณสมบัติ OnComm	26
ตารางที่ 2.7	ค่าคงที่สำหรับคุณสมบัติ InputMode	26
ตารางที่ 2.8	ค่าคงที่สำหรับคุณสมบัติ Error	27
ตารางที่ 2.9	ข้อความที่บ่งชี้ถึงค่าความผิดพลาดที่เกิดขึ้น MSComm Control	28

บทที่ 1

บทนำ

1.1 ความเป็นมาของโครงการ

ปัจจุบันวิวัฒนาการทางด้านเทคโนโลยีได้มีการพัฒนาไปมากโดยเฉพาะอย่างยิ่งในโรงงานอุตสาหกรรม เราจะเห็นได้ว่าการนำหุ่นยนต์มาทำงานแทนมนุษย์ ซึ่งจะทำให้ผลผลิตที่ออกมามีประสิทธิภาพที่แน่นอน รวมทั้งช่วยลดต้นทุนในด้านการผลิตทั้งในด้านแรงงาน และทางด้านทรัพยากร แต่ถึงกระนั้นก็ตาม การนำหุ่นยนต์หรือเครื่องมือเครื่องจักรบางอย่าง มาใช้ก็อาจจะประสบปัญหาเกี่ยวกับ การวางสายระบบควบคุมระหว่างตัวควบคุม(Computer) กับเครื่องจักรกล (Robot) จะต้องอาศัยตัวกลางเป็นตัวต่อเชื่อมในการควบคุม ซึ่งเป็นสิ่งที่ไม่สามารถหลีกเลี่ยงได้ถ้าพูดถึงงานควบคุมส่วนใหญ่แล้ว ยังคงอาศัยสายสัญญาณต่าง ๆ มาเป็นตัวกลางในการเชื่อมต่อระหว่างส่วนควบคุมกับเครื่องจักรกล และถ้าหากว่าพิจารณากันให้ดีแล้วจะพบว่า การใช้สายไฟหรือสายสัญญาณมาเป็นตัวกลางเชื่อมต่ออาจ เป็นปัญหาหรืออุปสรรคต่อการทำงานได้ สำหรับปัญหา และอุปสรรคของการใช้ สายสัญญาณมาใช้นั้นมี ข้อเสียอย่างไรบ้างนั้น เราสามารถจำแนกได้ดังต่อไปนี้

1. เกิดปัญหาที่เรียกว่า สายขาดในได้ง่าย
2. ในงานบางอย่างอาจเกิดปัญหาการเกาะกะของสายสัญญาณในขณะที่ปฏิบัติงาน
3. ต้องออกแบบการวางระบบการติดตั้ง สายสัญญาณ เมื่อเริ่มติดตั้งเครื่อง
4. ต้องคอยตรวจเช็คและเปลี่ยน สายสัญญาณ เมื่อสายนำสัญญาณหมดอายุ
5. ต้องมีการทวนสัญญาณ เมื่อมีการใช้สายนำสัญญาณที่ยาวมาก

อันที่จริงแล้วปัญหาและอุปสรรคอาจมีมากกว่านี้ สำหรับตัวอย่างที่ได้กล่าวมาแล้วนั้น

เป็นเพียงส่วนหนึ่งเท่านั้นนอกจากตามโรงงานอุตสาหกรรมต่าง ๆ ได้มีการนำหุ่นยนต์ไปใช้ในโรงงานอุตสาหกรรมแล้ว หน่วยงานบางหน่วยงานได้มีการนำหุ่นยนต์ มาช่วยในการปฏิบัติงาน เช่น งานที่เสี่ยงอันตรายต่อชีวิตมนุษย์ตัวอย่าง งานกู้วัตถุระเบิด หรืองานสำรวจที่มนุษย์ไม่สามารถไปสำรวจด้วยตัวเองได้ โดยอาจมีปัญหาทางด้านมลภาวะทางด้านสิ่งแวดล้อมการทำงานอื่น ๆ อันได้แก่ (อุณหภูมิ, รังสี , สารเคมี ฯลฯ)

1.2 วัตถุประสงค์ในการทำโครงการ

1. เพื่อแก้ปัญหาการแกะกะของสายสื่อนำสัญญาณ ในขณะที่ปฏิบัติงาน
2. เพื่อไม่ต้องออกแบบการวางระบบการติดตั้ง สายสื่อนำสัญญาณ
3. เพื่อไม่ต้องคอยตรวจเช็คและเปลี่ยนเมื่อสายนำสัญญาณหมดอายุ
4. เพื่อแก้ปัญหาสายขาดใน และ มีการทวนสัญญาณเมื่อมีการใช้สายนำสัญญาณที่ยาว ๆ
5. เพื่อประยุกต์ใช้งานการสื่อสารแบบไร้สาย

1.3 ขอบเขตของโครงการ

สามารถรับส่งข้อมูลโดยควบคุมการทำงาน โดยคอมพิวเตอร์

1. สามารถควบคุมการทำงานให้ทำงานตามต้องการ
2. ให้อ่านพารามิเตอร์ได้ตามต้องการ

1.4 วิธีการดำเนินงาน

1. ศึกษาการรับส่งข้อมูลการมอดูเลตสัญญาณดิจิทัล
2. ศึกษาการเชื่อมโยงข้อมูล (Interface) ระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอก
3. ทดลองวงจรต่าง ๆ ในโครงการ
4. เขียนโปรแกรมควบคุมการทำงานและบันทึกผลการทดลอง

1.5 ประโยชน์ที่ได้รับ

1. ประหยัดค่าใช้จ่ายจากปัญหาที่เกิดจากสายที่เชื่อมต่อ
2. มีความสะดวกในการติดต่อระหว่างคอมพิวเตอร์กับอุปกรณ์ที่เชื่อมต่อ
3. มีความปลอดภัยจากการตรวจเช็คในสถานที่ที่มีสิ่งแวดล้อมเป็นพิษ
4. ได้ข้อมูลที่ถูกต้องและแม่นยำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

2.1 บทนำ

ในการส่งข้อมูลขนาดของข้อมูล 1 ตัวอักษรจะขึ้นอยู่กับอุปกรณ์ การสื่อสารที่ใช้ซึ่งจะมีความยาวอยู่ระหว่าง 7-8 บิต ตัวอักษรแทนความหมายด้วย ตัวเลขฐานสอง ซึ่งจะมีการเปลือออกมาเป็นอักขระ, ตัวเลข หรือเครื่องหมาย

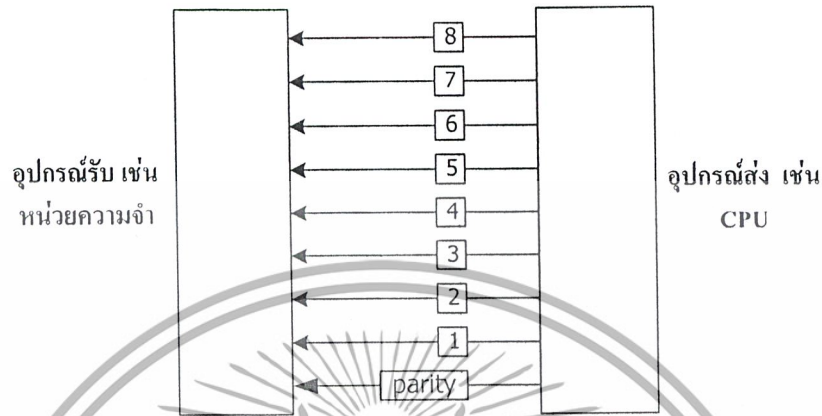
ชุดของกลุ่มเลขฐานสองที่มีการกำหนดความหมาย ต่าง ๆ มีรหัส จำนวนมากมายที่ใช้กันอยู่ในปัจจุบันในอุปกรณ์การสื่อสาร ซึ่งจะถูกออกแบบมาให้ใช้ได้เฉพาะกับรหัสชนิดต่าง ๆ ขึ้นอยู่กับวัตถุประสงค์ของผู้ออกแบบและการใช้งาน

2.2 การสื่อสารข้อมูล (Data Communication)

2.2.1 การรับส่งข้อมูลแบบขนาน(Parallel Transmission)

ลักษณะของการส่งข้อมูลแบบขนาน ทำโดยการส่งข้อมูลออกมาทีละ 1 ไบต์ คือ 8 บิต จากอุปกรณ์ส่งไปยัง อุปกรณ์รับ ตัวกลางระหว่างสองเครื่อง จะต้องมีช่องทางให้ข้อมูล อย่างน้อย 8 ช่องทาง โดยมากจะเป็นสายขนานให้กระแสไฟฟ้าวิ่งมากกว่าที่จะเป็นตัวกลางชนิดอื่น

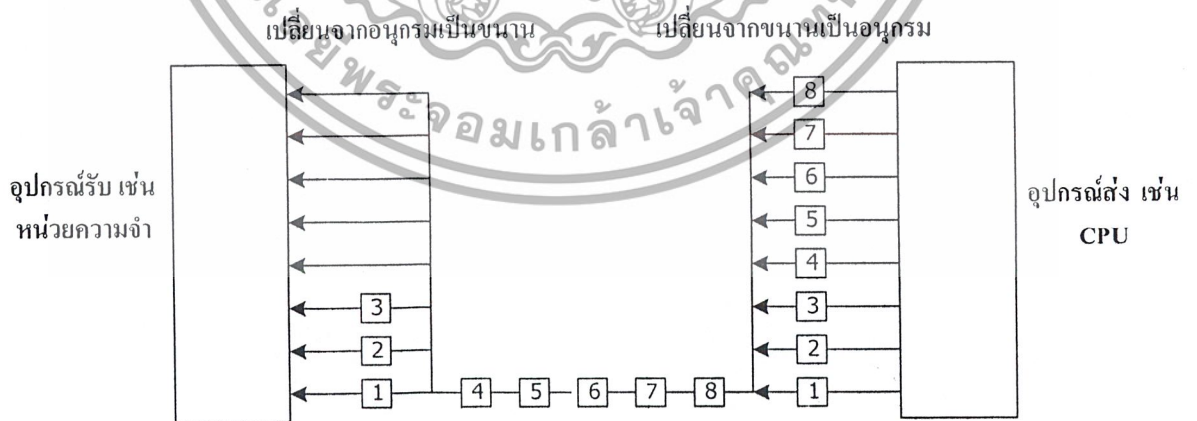
นอกจากสายที่เป็นทางเดินของข้อมูลแล้วอาจจะมีการเดินทางของสัญญาณควบคุมอื่น ๆ อีกเป็นต้นว่า บิตที่บอกพาริตี ของสัญญาณ เพื่อเป็นการตรวจสอบความผิดพลาดของการรับสัญญาณที่ปลายทาง หรือที่ควบคุมการโต้ตอบ จะเห็นได้ว่าการส่งแบบขนานส่วนมากจะทำใน ระยะใกล้ ๆ เนื่องจากจะต้องมีช่องทางเดินของสัญญาณมากกว่า 8 สาย และอุปกรณ์ที่ติด ต่อ แบบขนานกับคอมพิวเตอร์ได้แก่ เครื่องพิมพ์, ฟลอปปีดิสก์ เป็นต้น



รูปที่ 2.1 แสดง การส่งข้อมูลแบบขนาน

2.2.2 การรับส่งข้อมูลแบบอนุกรม (Serial Transmission)

การส่งข้อมูลแบบอนุกรม ข้อมูลถูกส่งออกมาทีละบิต ระหว่างจุดส่งและจุดรับ จะเห็นว่า การส่งข้อมูลแบบนี้จะช้ากว่าแบบขนาน เพราะตัวกลางการสื่อสารต้องการเพียงช่องเดี่ยว หรือสาย เพียงคู่เดียว ค่าใช้จ่ายในตัวกลางสื่อสารจะถูกกว่าแบบขนาน สำหรับการส่งระยะทางไกล ๆ



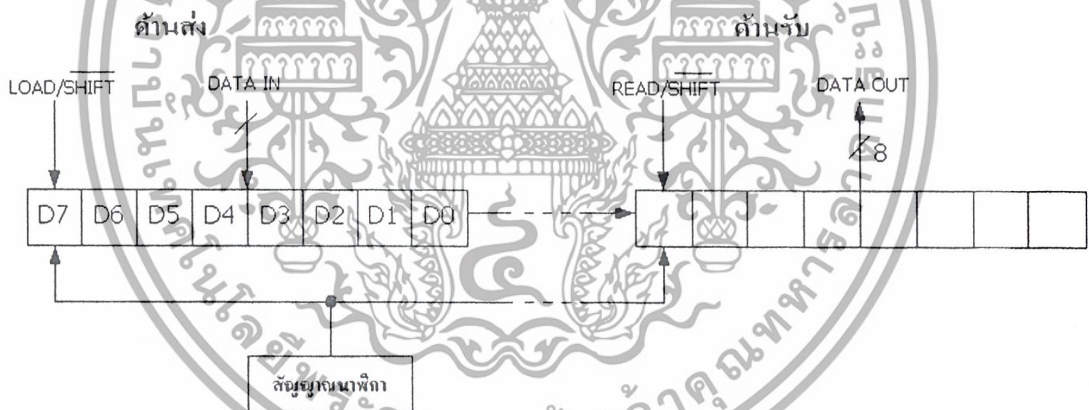
รูปที่ 2.2 แสดง การส่งข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปแสดงให้เห็นการส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้ เป็นอนุกรม เสียก่อนแล้วค่อยทยอยส่งออกที่ละบิตไปยังจุดรับ ณจุดที่รับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่ง มาที่ละบิตให้เป็นสัญญาณแบบขนานซึ่งลงตัวพอดี นั่นคือ บิต 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดี การที่ จะทำให้การแปลงสัญญาณจากอนุกรมที่ละบิตให้ลงพอดี นั้นจำเป็นจะต้องมีลักษณะการส่งที่ เหมาะสม เพื่อป้องกันการผิดพลาดในการรับ ลักษณะการส่งที่เหมาะสม แบ่งเป็น 2 แบบ คือ การ ส่งข้อมูลแบบซิงโครนัสและการส่งข้อมูลแบบอะซิงโครนัส

2.3 การส่งข้อมูลแบบซิงโครนัส (Synchronous Transmission)

การส่งข้อมูลแบบซิงโครนัสหรืออีกอย่างหนึ่งว่าการส่งข้อมูลแบบสัมพันธ์ หมายถึงการที่ ด้านรับอ่านข้อมูลเข้ามาในจังหวะเดียวกับด้านส่ง โดยใช้สัญญาณนาฬิกาเป็นตัวกำหนด จังหวะการ ทำงานของรีจิสเตอร์ทั้งสองให้ทำงานสัมพันธ์กัน วงจรกำเนิดสัญญาณนาฬิกา จะติดตั้งทางด้านส่ง



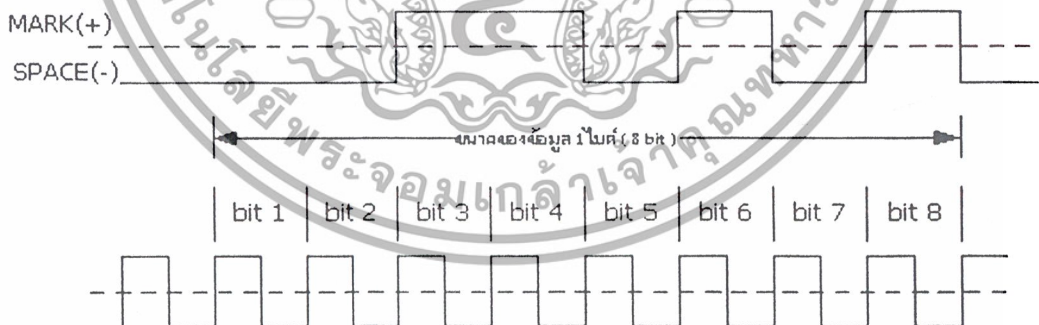
รูปที่ 2.3 แสดงการส่งข้อมูลแบบซิงโครนัส

เมื่อจังหวะเวลาถูกตั้งให้ซิงค์กับด้านรับได้แล้ว ข้อมูลจะถูกส่งไปบนทางติดต่อ ในแบบบิต ต่อบิตต่อเนื่องกันไป อาศัยช่วงเวลาระหว่างบิตต่อบิตมีค่าเท่ากัน โดยไม่ต้องมีบิตเริ่มหรือบิตจบ คอยกำกับ ทำให้ความเร็วในการส่งข้อมูลมีสูง นอกจากนี้ทางด้านรับต้องมีวงจรเฟสล็อกคูลูป ทำหน้าที่รับข้อมูลจังหวะเวลาจากด้านส่งและสร้างสัญญาณขึ้นใหม่ในด้านรับ

ข้อมูลแบบสัมพันธ์จะถูกจัดให้อยู่ในรูปข้อมูล ที่มีลักษณะพิเศษคือ ช่วงระยะเวลาระหว่างตัวอักษรด้วยกันจะไม่มี ทำให้การส่งข้อมูลเป็นไปอย่างต่อเนื่อง ข้อเสียของการส่งข้อมูลแบบสัมพันธ์คือการที่มีสัญญาณรบกวนนานไปกับข้อมูล ทำให้ต้องการทางติดต่อช่องที่สองเพิ่ม โดยเฉพาะกรณีระยะทางไกล ๆ เป็นการยากมากที่จะจัดหาทางติดต่อ แยกทางต่างหากสำหรับสัญญาณรบกวน

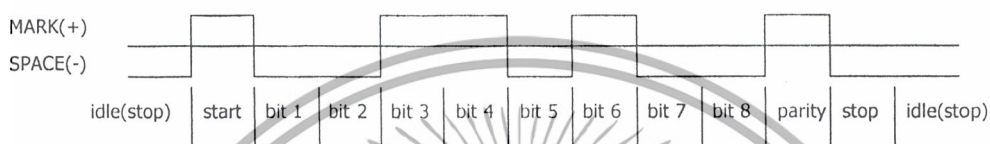
2.4 การส่งข้อมูลแบบอะซิงโครนัส(Asynchronous Transmission)

การส่งข้อมูล แบบอะซิงโครนัส หรือเรียกอีกอย่างหนึ่งว่าการส่งข้อมูล แบบไม่สัมพันธ์ จะแตกต่างกับแบบซิงโครนัส ตรงที่ไม่จำเป็นต้องมีการซิงโครไนส์กันตลอดเวลาบนทางติดต่อข้อมูล โดยจะซิงค์ก็ต่อเมื่อมีข้อมูลที่ได้รับหรือส่งเท่านั้น ในการส่งสัญญาณแบบอะซิงโครนัสการส่งตัวอักษรสามารถเริ่มจากเวลาใดก็ได้เมื่อสายว่าง มีข้อมูลตัวอักษรที่อยู่บนสายสภาวะบางอย่างที่ใช้บอกนั่นก็คือ บิตเริ่มต้น บิตเริ่มต้นไม่ใช่บิตข้อมูล แต่เป็นสัญญาณควบคุมที่จุดเริ่มต้น สถานะของสายส่งข้อมูลจะเปลี่ยนจาก “1” (Mark) มาเป็น “0” (Space) และ ถัดจากบิตเริ่มต้นก็จะเป็นบิตข้อมูลตัวอักษร ซึ่งจะป็นเนื้อหาของข้อมูลของตัวอักษร แต่ละตัวที่ถูกส่งบิตที่มีนัยสำคัญน้อยสุด(LSB) จะถูกส่งตามบิตเริ่มต้นออกมา



รูปที่ 2.4 แสดงข้อมูลแบบซิงโครนัส

และหลังจากบิตข้อมูลส่งออกมาครบแล้วจะตามด้วย บิตสิ้นสุด (Stop bit) โดยที่สถานะของสายส่งข้อมูลจะถูกบังคับกลับเป็นมาร์ค อีกครั้งหนึ่ง หน้าที่ของบิตสิ้นสุดก็เพื่อบอกการสิ้นสุดของข้อมูลที่รับส่ง เนื่องจากการส่งข้อมูลแบบอะซิงโครนัสเหมาะสำหรับงานประเภทการป้อนข้อมูลเข้าเพื่อส่ง มีลักษณะไม่ต่อเนื่องอย่างเช่น จากแป้นพิมพ์(Key board) ส่วนในการส่งไฟล์ข้อมูลที่มีขนาดใหญ่มาก ๆ ควรใช้วิธีแบบซิงโครนัส



รูปที่ 2.5 แสดงรูปแบบข้อมูลแบบอะซิงโครนัส

2.5 บิตเริ่มต้น(Start Bit)

การส่งข้อมูลอนุกรมแบบอะซิงโครนัส กำหนดให้สถานะมาร์ค(Mark State) เป็นสัญญาณลอจิก 1 เมื่อทางด้านส่งต้องการส่งข้อมูล ก็จะต้องส่งบิตเริ่มต้นโดยแทนด้วย สถานะสเปส(Space State) หรือสัญญาณลอจิก 0 จำนวน 1 บิตไปก่อน ซึ่งจะทำให้ด้านรับดีเทกซ์ (Detect) สถานะของสายส่งได้ว่าขณะนั้นสายส่ง กำลังส่งข้อมูลเข้ามา สำหรับปัญหาที่เกิดขึ้น และ มีผลต่อสัญญาณ ข้อมูลก็คือ สไปค์ (Spike) ทำให้สถานะลอจิกของสายส่งมี ช่วงเวลาสั้นเกินไปทำให้ทางรับไม่สามารถดีเทกซ์สถานะของสายส่งหรือสถานะเริ่มต้นได้ ดังนั้นส่วนใหญ่ทางด้านรับจะมีส่วนของวงจรสไปค์ ดีเทกชัน (Spike Detection) ที่ทำหน้าที่สุ่มจับสัญญาณสถานะ ของสายส่ง

ดังนั้นเราอาจสรุปได้ว่า หน้าที่ของบิตเริ่มต้นนั้นจะเป็นตัวบอกว่าข้อมูลเริ่มต้นตรงไหน และเมื่อใช้ร่วมกับบิตสิ้นสุด ก็จะทำให้ทราบได้ว่าข้อมูลสิ้นสุดตรงไหน โดยความกว้างของบิตเริ่มต้นนี้จะมีความกว้างเท่ากับ 1 บิตข้อมูล

2.6 บิตข้อมูล (Data bit)

หลังจากที่ด้านรับสามารถดีเทกซ์สัญญาณบิตเริ่มต้นได้แล้ว ก็ทำให้การตรวจสอบสถานะของชิป รีจิสเตอร์พร้อมที่จะรับบิตข้อมูลได้ โดยบิตข้อมูลจะมีจำนวนบิตเป็น 5, 7 หรือ 8 บิตขึ้นกับจำนวนคาร์แรกเตอร์ที่ใช้

จำนวนบิตข้อมูลใน 1 คาร์แรกเตอร์	จำนวนคาร์แรกเตอร์
5 บิต	32 (Baudot)
7 บิต	128 (ASCII)
8 บิต	256(EBCDIC)

รูปตารางที่ 2.1 แสดงจำนวนบิตที่ขึ้นอยู่กับจำนวนคาร์แรกเตอร์

2.7 บิตพาริตี (Parity bit)

บิตนี้จะทำหน้าที่ในการบอกให้ส่วนรับข้อมูลทราบว่า ข้อมูลรับเข้ามาผิดหรือไม่ (โดยบิตนี้จะทำหน้าที่ในการบอกให้ส่วนรับทราบว่าข้อมูลที่ส่งมาแต่ละไบต์ มีจำนวนบิตที่เป็น “1” อยู่เป็นจำนวนคี่หรือจำนวนคู่ เช่น ข้อมูล 54H = 0110111 จะมีบิตที่เป็น “1” อยู่เป็นจำนวนคี่ เป็นต้น ซึ่งบิตที่ใช้ในการตรวจสอบนี้ เราเรียกบิตพาริตี) บิตพาริตีจะถูกส่งออกมาพร้อมกับข้อมูล ซึ่งบิตนี้จะ เป็น “1” หรือ “0” นั้นขึ้นอยู่กับข้อมูลที่ส่งออกมามีจำนวนบิตที่เป็น “1” เป็นจำนวนคี่หรือคู่ และ ยังขึ้นอยู่กับอุปกรณ์รับส่งข้อมูลด้วยว่าถูกออกแบบไว้ให้รับส่งบิตพาริตีในลักษณะของพาริตีคู่ หรือคี่ อีกด้วย

สิ่งสำคัญอีกสิ่งหนึ่งก็คือ ถ้าอุปกรณ์ส่งข้อมูลทำการส่งในลักษณะพาริตีคู่ หรือคี่ก็ตามส่วนรับข้อมูลก็ต้องทำการรับในลักษณะพาริตีเดียวกับอุปกรณ์ส่งข้อมูลด้วย เช่น ในกรณีที่อุปกรณ์ส่งข้อมูลส่งในลักษณะพาริตีคู่ อุปกรณ์รับข้อมูลก็ต้องทำการรับข้อมูลในลักษณะของพาริตีคู่ด้วย เป็นต้น

2.8 บิตสิ้นสุดข้อมูล (Stop bit)

บิตสุดท้ายที่เพิ่มเข้าไปนี้ จะใช้ในการตรวจสอบจุดสิ้นสุดของข้อมูล บิตนี้จะถูกเพิ่มเข้าไปหลังพาริตีบิต ถ้าอุปกรณ์รับข้อมูลตรวจสอบแล้วไม่พบบิตนี้ ก็แสดงว่าข้อมูลที่รับเข้ามานั้นเกิดข้อผิดพลาดขึ้นมาในแต่ละไบต์นั้นไม่ใช่มีแต่ข้อมูล 8 บิต (ในกรณีที่ 1 ไบต์มี 8 บิต) เท่านั้น แต่อาจจะ มีถึง 12 บิต หรือมากกว่าก็เป็นได้

2.9 ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรม

ความเร็วในการถ่ายโอนข้อมูลแบบอนุกรมมีหน่วยเป็น บิตต่อวินาที (Bit per sec: bps) ส่วนการเปลี่ยนแปลงสัญญาณใน 1 วินาที เรียกว่า บอดเรต (baud rate) หรือ อัตราบอด การเปลี่ยนแปลงของสัญญาณ 1 ครั้ง อาจแสดงถึงการส่งข้อมูลแบบอนุกรมมากกว่า 1 บิต ก็ได้ ถ้าเขียนในรูปของสมการคณิตศาสตร์จะได้

$$\text{อัตราบิต (Bit rate)} = \text{อัตราบอด (baud rate)} \times (\text{บิตใน 1 บอด})$$

2.10 การมอดูเลตทางความถี่

เราใช้คลื่นรูปไซน์ที่มีความถี่สูงเป็นพาหะแล้วเปลี่ยนคุณสมบัติบางอย่างของพาหะด้วยสัญญาณข่าวสาร โดยทั่วไปสัญญาณข่าวสารได้แก่ สัญญาณออดิโอ (เสียงพูด), สัญญาณภาพ หรือข่าวสารอื่นๆ การเปลี่ยนแปลงคุณสมบัติของคลื่นพาหะนี้เราเรียกว่า การมอดูเลต คลื่นรูปไซน์ที่เราใช้เป็นพาหะนั้นเราสามารถเขียนสมการคณิตศาสตร์แทนได้ดังนี้

$$e = A \sin(\omega t + \phi)$$

เมื่อ e คือ ค่าแรงดัน (หรือกระแส) ของคลื่นพาหะใดๆ

A คือ แอมพลิจูด (หรือขนาด) สูงสุดของคลื่นพาหะ

ω คือ ความถี่เชิงมุม

t คือ เวลา

ϕ คือ เฟสหรือมุมทางไฟฟ้า

การมอดูเลตให้กับคลื่นพาหะแบ่งออกได้เป็น 3 แบบคือ

1. การมอดูเลตทางแอมพลิจูด (Amplitude Modulation ; AM)
2. การมอดูเลตทางความถี่ (Frequency Modulation ; FM)
3. การมอดูเลตทางเฟส (Phase Modulation ; PM)

2.11 ดัชนีการมอดดูเลท

ในระบบ FM เรวัตต์เปอร์เซ็นต์การมอดดูเลทโดยดูจากการเปลี่ยนแปลงความถี่ ซึ่งเรานิยมเรียกชื่อใหม่ว่า ดัชนีการมอดดูเลท ลองพิจารณาความหมายของดัชนีการมอดดูเลทต่อไปนี้

$$m = \frac{f_d}{f_m}$$

เมื่อ f_d คือช่วงความถี่เบี่ยงเบน

f_m คือความถี่ของสัญญาณที่เข้ามอดดูเลท

ค่าตัวเลขของดัชนีการมอดดูเลทจะมีค่าสูง (แตกต่างจากเปอร์เซ็นต์การมอดดูเลทซึ่งเมื่อคิดเป็นอัตราส่วนจะได้อยู่ระหว่าง 0 ถึง 1) ตัวอย่างเช่น ในระบบวิทยุกระจายเสียง FM เรากำหนดให้ความถี่เบี่ยงเบนของระบบสูงสุดไว้เท่ากับ 75 kHz สมมุติว่าเราใช้สัญญาณเสียง 1 kHz มอดดูเลทให้เกิดความถี่เบี่ยงเบนเต็มที ค่าดัชนีมอดดูเลทจะเป็น

$$m = \frac{75 \text{ kHz}}{1 \text{ kHz}} = 75$$

สังเกตว่า ค่าดัชนีการมอดดูเลทในระบบ FM ขึ้นอยู่กับความถี่ของสัญญาณเสียงที่เข้ามอดดูเลท ในทางปฏิบัติเรานิยมวัดเป็นอัตราส่วนการเบี่ยงเบน (deviation ratio) ซึ่งเป็นอัตราส่วนระหว่างความถี่เบี่ยงเบน ของระบบสูงสุด ($f_{d\max}$) ต่อความถี่สูงสุดของสัญญาณที่เข้ามอดดูเลท ($f_{m\max}$) ในระบบกระจายเสียง FM ค่าอัตราเบี่ยงเบน (Δ) จะเท่ากับ

$$\Delta = \frac{f_{d\max}}{f_{m\max}}$$

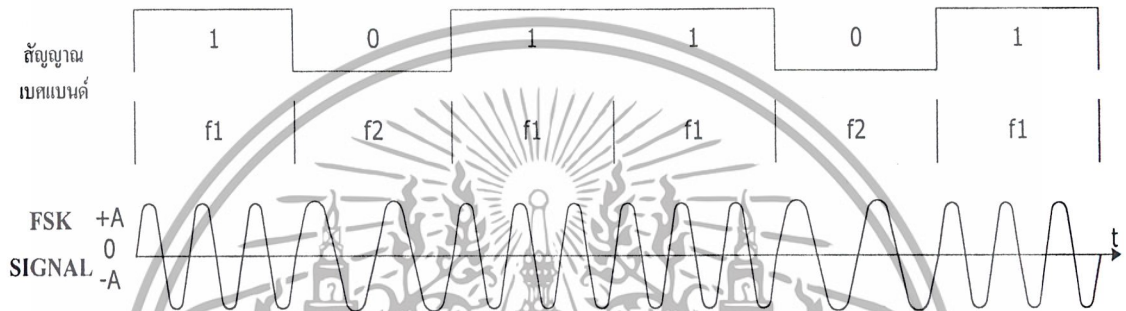
2.12 การมอดดูเลทดิจิตอลทางความถี่ (Frequency Shift Keying; FSK)

การมอดดูเลทดิจิตอลทางความถี่เมื่อข้อมูลเป็นลอจิก “1” ก็จะได้สัญญาณอนาลอกความถี่หนึ่งและเมื่อเป็นลอจิก “0” ก็จะได้สัญญาณอนาลอกอีกความถี่หนึ่ง ถ้าสัญญาณพาหะเป็นดังสมการข้างล่าง

ดังนั้นกรณีของการมอดูเลตแบบ FSK ความถี่ของคลื่นพาหะจะมี 2 ความถี่ เช่น ความถี่ f_1 สำหรับสัญญาณที่เป็นลอจิก “1” และความถี่ f_2 สำหรับสัญญาณที่เป็นลอจิก “0” แสดงดังสมการข้างล่าง

$$c = A \sin 2\pi f_1 t \quad ; \text{เมื่อสถานะบิตเป็น "1"}$$

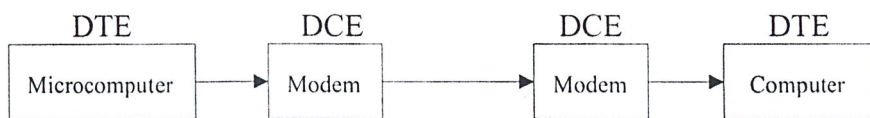
$$= A \sin 2\pi f_2 t \quad ; \text{เมื่อสถานะบิตเป็น "0"}$$



รูปที่ 2.6 แสดงการมอดูเลตคิจิตอลทางความถี่

2.13 พอร์ต RS-232C

พอร์ต RS-232C ทำหน้าที่รับและ ส่งข้อมูลแบบอนุกรมที่เรียกว่า Universal Asynchronous Adapter เนื่องจาก EIA ได้กำหนดมาตรฐาน ของอุปกรณ์เอาไว้ภายใต้ชื่อว่า RS-232C ความจริงมาตรฐานการส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมใช้กันมากที่สุดสำหรับไมโครคอมพิวเตอร์ก็คือ RS-232C คำว่า “RS” ย่อมาจาก “Recommend Standard” ส่วน “232” เป็นหมายเลขบ่งบอกมาตรฐาน และ “C” เป็นหมายเลขของฉบับท้ายสุดของมาตรฐานตัวนี้



RS 232C Interface

RS 232C Interface

รูปที่ 2.7 การใช้ RS-232C เชื่อมต่ออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.1 ลักษณะสมบัติทางไฟฟ้าของสัญญาณ RS-232C

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้องและอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นจะต้องมีข้อตกลงในเรื่องของสัญญาณที่ใช้ ซึ่งลักษณะสมบัติทางไฟฟ้าของสัญญาณ RS-232C สามารถแบ่งเป็นหัวข้อดังต่อไปนี้

สถานะ

แรงดันสูงสุดที่วงจรใน DTE และ DCE ไม่ควรเกิน 25 V และ Open Circuit Voltage ต้องไม่เกิน 2 V (วัดเทียบกับ Signal Ground)

มาตรฐานของการใช้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	ลอจิก	สัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	Space	ON
ลบ	1	Mark	OFF

รูปตารางที่ 2.2 แสดงย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS-232

วงจรขับสัญญาณ

หากเกิดสภาพการลัดวงจรของสายเชื่อมต่อกับใคร ๆ จะต้องมิกระแสดังวงจรไม่เกิน 0.5A และแรงดันที่ DCE ต้องไม่เกิน 25V เมื่อถอด Connector

ความต้านทาน

เมื่อมองจากจุด Interface Point ไปยัง DCE จะต้องมิตค่า 3-7k Ω

2.13.2 ความไวของการเปลี่ยนแปลงระดับของสัญญาณ

สัญญาณควบคุมจะต้องผ่านช่วงเปลี่ยนแปลงระดับ(-3V ถึง +3V) ภายในเวลาไม่เกิน 1 msec ส่วนสัญญาณข้อมูลจะต้องผ่านช่วงเปลี่ยนแปลงระดับภายในเวลาไม่เกิน 1 msec หรือ 4% ของความกว้างของรูปสัญญาณนั้น

2.13.3 ลักษณะสมบัติเชิงกลของข้อต่อ (Connector)

มาตรฐานของ RS232C ไม่ได้กำหนดต้องใช้ Connector รูปร่างอย่างไร แต่ที่นิยมใช้กันทั่วไปคือ DB-25 Connector ซึ่งแต่ละขาของข้อกำหนดได้แสดงดังตารางที่ 2.3 สัญญาณต่าง ๆ ทำหน้าที่ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Protective Ground (PG ที่ขา 1) หมายถึงตัวถังของเครื่องหรือสายดิน

Transmit Data (TD ที่ขา 2)

เป็นสัญญาณที่ส่งออกจาก DTE ไปยังโมเด็มหรือต่อเข้าโดยตรงกับโมโครคอมพิวเตอร์ตัวอื่นเมื่อไม่มีสัญญาณส่งออกสถานะของลอจิกจะมีค่าเท่ากับ “1” สภาวะ “OFF” หรือเทียบเท่ากับ Stop bit และไม่ว่าจะไม่ใช่ระบบอะไร DTE ต้องไม่ส่งข้อมูลออกไปจนกว่าสัญญาณ Request To Sent (RTS), Clear To Send (CTS), Data Set Ready (DSR) และ Data Terminal Ready (DTR) ทั้งหมดนี้อยู่ในสภาวะ “ON” อยู่

Receive Data (RD ที่ขา 3)

เป็นสัญญาณที่เข้าไปยัง DTE เมื่อไม่มีสัญญาณรับเข้ามาที่ขา นี้ จะมีสถานภาพทางลอจิกเป็น “1” หรือสภาวะ “OFF”

Request To Sent (RTS ที่ขา 4) จาก DTE ไปยัง DCE

สภาวะ “ON” คือบังคับให้ DCE อยู่ใน Transmitting Mode ต่อไป

สภาวะ “OFF” คือบังคับให้ DCE อยู่ใน Receiving Mode ต่อไป

Clear To Sent (CTS ที่ขา 5)

จาก DCE ไป DTE สภาวะ “ON” หมายความว่า ข้อมูลจาก DTE ขา 2 จะถูกส่งต่อไปในช่องสื่อสาร (โมเด็มส่งข้อมูลออกสายโทรศัพท์) ทันที CTS จะ “ON” หลังจาก DSR และ RTS อยู่ในสภาวะ “ON” และการต่อเชื่อมวงจรสื่อสาร (ขุมสายโทรศัพท์) เสร็จแล้ว

Data Set Ready (DSR ขาที่ 6)

จาก DCE ไป DTE คือความพร้อมของโมเด็มนั่นเองจะเป็น “ON” ก็ต่อเมื่อ DCE (โมเด็ม) เปิดเครื่องอยู่จะอยู่ในสภาวะ off-hook (เหมือนยกหูโทรศัพท์), DCE ไม่อยู่ใน test mode และ DCE ทำการส่งสัญญาณไปยังขุมสายเสร็จแล้ว

DSR อยู่ในสภาวะ “ON” เป็นการบอกว่า DTE ว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้ว และพร้อมที่จะส่งได้แล้ว

DSR อยู่ในสภาวะ “OFF” หมายถึงให้ DTE เลิกสนใจสัญญาณตัวอื่น ๆ ยกเว้น Ring Indicator

Signal Ground (SG ขาที่ 7)

SG ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกสายของสัญญาณจะมีแรงดันเป็น “0” เมื่อเปรียบเทียบกับ

Carrier Detect (CD ขาที่ 8) จาก DCE ไป DTE

สถานะ ON คือ DCE จับสัญญาณพาหะในช่องทางสื่อสารที่จะทำการ Demodulate

สถานะ OFF คือ DCE ไม่ได้จับสัญญาณอะไรเลยหรือได้รับสัญญาณแต่ไม่สามารถทำการ Demodulate เอาข้อมูลออกมาได้

Data Terminal Ready (DTR ขาที่ 20) จาก DTE ไป DCE

สถานะ “ON” หมายความว่า DCE ทำการเตรียมเพื่อเชื่อมต่อกับตัวอื่น และรักษาช่องทางติดต่อไว้ต่อไป (การเชื่อม Channel ทำได้หลายทางคือหมุนเรียกด้วยมือหรืออัตโนมัติ) ถ้า DCE สามารถตอบรับสัญญาณเรียก(Call)ได้ ก็ให้ตอบรับ (Answering)เมื่อมีสัญญาณเรียก Ring Indicator และ DTR อยู่ในสถานะ “ON”

สถานะ “OFF” คือวางหูและเมื่อ “OFF” แล้วต้องไม่ “ON” อีกจนกว่า DSR จะ “OFF”

Ring Indicate (RI ขาที่ 22)

จาก DCE ไป DTE เหมือนสัญญาณเรียกของโทรศัพท์แต่เป็นดิจิทัล ใช้ในระบบอัตโนมัติ (Auto-answer) สัญญาณนี้จะ “ON” เมื่อมีสัญญาณกระดิ่งเข้ามาและจะ “OFF” ระหว่างเสียงดังของกระดิ่ง

ส่วนขาอื่น ๆ ที่เหลือ นั้น ส่วนมากมีหน้าที่คล้ายกับ 8 ขาแรกทีกล่าวมาแล้วและบางเส้นใช้กับงานพิเศษเท่านั้น ขาที่เราใช้สำหรับรับส่งข้อมูลของข้อต่อแบบ DB-25 จึงเหลือเพียงขา 2, 3, 4, 5, 6, 7, 8, 20 และ 22 ยกเว้นการต่อใช้งานบางอย่างถึงจะต่อครบทุกเส้น

DB-25 และ DB-9

จากการที่ข้อต่อแบบ 25 ขาเราใช้งานจริงเพียง 9 ขาเท่านั้น เครื่องคอมพิวเตอร์รุ่นใหม่ๆ จึงได้ลดข้อต่อลงมาใช้แบบ 9 ขาแทน ซึ่งเราเรียกข้อต่อแบบนี้ว่า DB-9 การใช้ข้อต่อแบบ DB-9 นี้มีข้อดีหลายอย่างคือ ขนาดเล็กกะทัดรัด ราคาของข้อต่อถูกกว่า การต่อสายเคเบิลสะดวกขึ้น และการทำงานคล่องตัวขึ้น สัญญาณต่าง ๆ ของข้อต่อแบบ DB-9 บางเส้นจะตรงกับที่ใช้ใน DB-25 ดัง ที่แสดงในตารางเปรียบเทียบดังตารางที่ 2

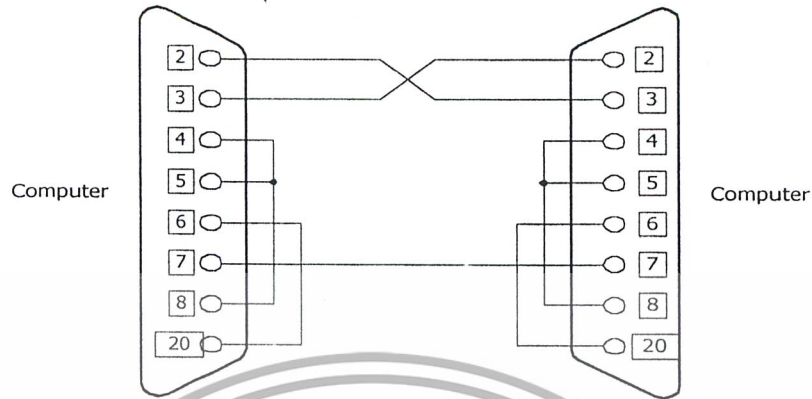
DB-25 Pin	DB-9 Pin	Assignment/Function
8	1	Carrier Detect (CD)
3	2	Receive Data (RD)
2	3	Transmit Data (TD)
20	4	Data Terminal Ready (DTR)
7	5	Signal Ground (SG)
6	6	Data Set Ready (DSR)
4	7	Request To Send (RTS)
5	8	Clear To Send (CTS)
22	9	Ring Indicator (RI)

รูปตารางที่ 2.3 แสดงขาที่ใช้งานของ DB-9 และ DB-25 ในการรับส่งข้อมูลแบบอนุกรม

2.14 การเชื่อมต่อคอมพิวเตอร์กับคอมพิวเตอร์โดยตรง (Null Modem)

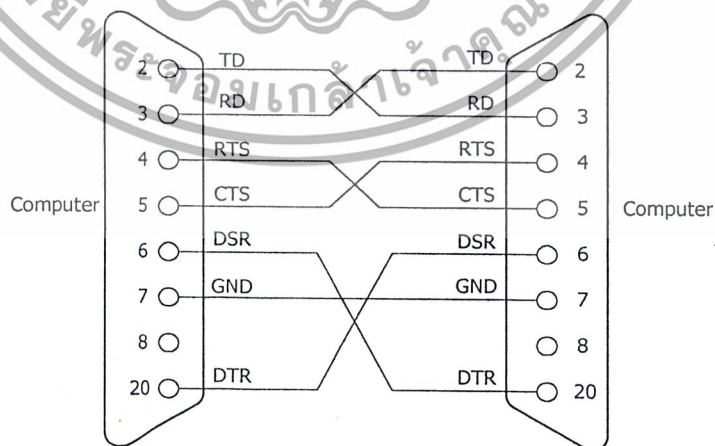
การต่อเครื่องคอมพิวเตอร์เข้าด้วยกันโดยใช้ RS-232C แล้วถ่ายโอนข้อมูลจากเครื่องหนึ่งไปอีกเครื่องหนึ่ง วิธีการต่อแบบนี้เรียกว่า Null Modem ก็คือการต่อโดยใช้โมเด็มนั่นเอง วิธีการต่อ RS-232C เข้าระหว่างเครื่องคอมพิวเตอร์โดยตรงมีอยู่หลายวิธีตามแต่ขบวนการที่จะใช้ ถ้าไม่ต้องการมีการตรวจสอบสัญญาณกันก็คือ RD เข้า TD ของอีกเครื่องหนึ่ง สายกราวด์ต่อถึงกันก็สามารถใช้งานถ่ายโอนข้อมูลได้แล้ว

ปกติ OS ที่ให้บริการเกี่ยวกับพอร์ต RS-232C จะส่งสัญญาณ Request To Sent (RTS) ออกมาที่ขา 4 ก่อน เมื่อ Clear To Send (CS) ที่ขา 5 เป็นลอจิก “ 1 ” จึงจะเริ่มทำงานส่งข้อมูลทีโอเปอเรเตอร์บอกให้ส่งออกไปที่ขา 2 ในกรณีที่เป็นการต่อแบบง่าย ๆ จึงถือว่าเป็นการหลอกคอมพิวเตอร์ โดยเอาขา 4 (RTS) ต่อเข้ากับขา 5 (CTS) เพื่อให้คอมพิวเตอร์ส่งข้อมูลได้ทันทีโดยไม่ต้องการความเรียบร้อยของฝ่ายรับ



รูปที่ 2.8 แสดง การต่อ RS-232C ระหว่างคอมพิวเตอร์อย่างง่าย ๆ

สำหรับขา 6 Data Set Ready (DSR) ต่อเข้ากับขาที่ 20 Data Terminal Ready (DTR) ก็ทำนองเดียวกัน โดยปกติคอมพิวเตอร์จะตามอุปกรณ์ที่มาต่อพ่วงกับ RS-232C ว่าพร้อมที่จะส่งหรือยัง โดยส่งสัญญาณขาที่ขา 20 และรอคำตอบที่ขา 6 จะได้คำตอบกลับที่ขา 6 ทันที ก่อนที่ฝ่ายส่งจะเป็นผู้ส่งไม่เช่นนั้นข้อมูลที่ออกมาจะหายไป เพราะฝ่ายส่งไม่ได้ตรวจสอบความเรียบร้อยของฝ่ายรับก่อน เราอาจจะต่อสายให้การตรวจสอบสัญญาณโต้ตอบ (Hand Shake) ซึ่งกรณีนี้จะมีการโต้ตอบที่ดีขึ้น เมื่อฝ่ายรับยังไม่พร้อมที่จะรับก็จะยังไม่มีสัญญาณ RTS ออกมา ฝ่ายส่งซึ่งถือเอา RTS ของฝ่ายรับเป็น CTS ก็จะยังไม่ส่ง



รูปที่ 2.9 แสดง การต่อคอมพิวเตอร์ผ่าน RS-232C แบบมี Hand Shake

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรมหรือ SCON (Serial port Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 98H ในพื้นที่ของรีจิสเตอร์ SFR สามารถเข้าถึงได้ในระดับบิตมีรายละเอียดการทำงานดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
SMO	SM1	SM2	REN	TB8	RB8	TI	RI

รูปตารางที่ 2.4 ใช้ในการเลือกโหมดในการเลือกพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51

SMO-SM1 (Serial port mode bit 0-1): ใช้ในการเลือกโหมดการทำงานของพอร์ตอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 ดังมีรายละเอียดต่อไปนี้

SM2: ใช้ในการเอ็นเอเปิดการสื่อสารในแบบมัลติโพรเซสเซอร์ (multiprocessor) ในการทำงานของโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 ถ้าบิตนี้เป็น “1” บิต RI จะไม่แอกติฟถ้าบิตที่ 9 ที่รับเข้ามาเป็น “0” (ข้อมูลบิตที่ 9 เก็บไว้ที่บิต RB8) ในการทำงานโหมด 1 ถ้าบิตนี้เซต บิต RI จะไม่แอกติฟถ้ายังไม่ได้รับบิตหยุด ส่วนในโหมด 0 บิตนี้ไม่มีการใช้งาน

REN (Enable serial reception): ใช้ในการเอ็นเอเปิดการรับข้อมูลของพอร์ตอนุกรม ทำการเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์ ถ้าต้องการให้มีการรับข้อมูลต้องเซตบิตนี้ให้เป็น “1”

TB8: ใช้สำหรับเก็บข้อมูลบิตที่ 9 ที่ต้องการส่งออกไปในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 เซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

RB8: ใช้สำหรับรับข้อมูลบิตที่ 9 ที่เข้ามาในการทำงานโหมด 2 และ 3 ของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 แต่ถ้าหากพอร์ตอนุกรมทำงานอยู่ในโหมด 1 และบิต SM2 เป็น “0” ข้อมูลที่บิต RB8 คือข้อมูลของบิตหยุด (STOP bit) สำหรับในการทำงานโหมด 0 บิตนี้จะไม่ใช้งาน บิต RB8 นี้สามารถเซตและเคลียร์ด้วยกระบวนการทางซอฟต์แวร์

TI (Transmit Interrupt flag): ใช้ในการแสดงการเกิดอินเทอร์รัปต์เมื่อมีการส่งข้อมูลออกจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการส่งข้อมูลบิตที่ 8 ไปเรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อมีการเริ่มต้นส่งบิตหยุดออกไป การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์

RI (Receive Interrupt flag): ใช้แสดงการเกิดอินเทอร์รัปต์เมื่อมีการรับข้อมูลเข้าสู่พอร์ตอนุกรม สามารถเซตได้ด้วยกระบวนการทางฮาร์ดแวร์ เมื่อทำการรับข้อมูลบิตที่ 8 เรียบร้อยแล้วในการทำงานโหมด 0 ส่วนในการทำงานโหมดอื่น บิตนี้จะเซตเมื่อสามารถรับบิตหยุดของข้อมูลอนุกรมไปได้ครึ่งทางแล้ว ยกเว้นในกรณีที่บิต SM2 มีการเซต บิตนี้จะเซตได้ก็ต่อเมื่อการรับบิตหยุดหรือบิตที่ 9 เกิดขึ้นอย่างสมบูรณ์แล้ว การเคลียร์บิตนี้ต้องใช้กระบวนการทางซอฟต์แวร์เท่านั้น

2.15.2 โหมดการทำงานของพอร์ตอนุกรมใน MCS-51

พอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 สามารถเลือกการทำงานได้ถึง 4 โหมดคือ

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะซีพรีจิสเตอร์
 2. โหมด 1 เป็นการกำหนดให้เป็น UART ขนาด 8 บิต สามารถเลือกอัตราบอดได้
 3. โหมด 2 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต โดยมีอัตราบอดคงที่
 4. โหมดที่ 3 เป็นการกำหนดให้เป็น UART ขนาด 9 บิต สามารถเลือกอัตราบอดได้
- การเลือกโหมดทำได้ด้วยการกำหนดข้อมูลให้แก่บิต SM0 และ SM1 ในรีจิสเตอร์ SCON

2.1 การทำงานในโหมด 1 ของวงจรพอร์ตอนุกรม

ในการรับส่งข้อมูลรวม 10 บิต โดยส่งข้อมูลออกทางขา P3.1 หรือ TxD และรับข้อมูลเข้าทางขา P3.0 หรือ RxD ข้อมูลทั้ง 10 บิตประกอบด้วย บิตเริ่มต้น (มีค่าเป็น “0”) ในการรับข้อมูลบิตจะเก็บไว้ในบิต RB8 ในรีจิสเตอร์ SCON อัตราบอดในโหมดนี้ได้รับการกำหนดโดยอัตราการเกิดโอเวอร์โพลวของไทมเมอร์ 1 ใน AT89C51 ส่วนในไมโครคอนโทรลเลอร์เบอร์ AT89Sxx สามารถเลือกใช้อัตราการเกิดโอเวอร์โพลวของไทมเมอร์ 1 หรือ ไทมเมอร์ 2 ในการกำหนดอัตราบอดได้

กระบวนการส่งข้อมูลเริ่มต้นด้วยการแอกตีฟสัญญาณเขียนข้อมูลมายังรีจิสเตอร์ SBUF ส่งมายังวงจรควบคุมการส่ง (TX control) จากนั้นวงจรควบคุมจะทำการแอกตีฟสัญญาณ SEND ที่สเตต 1 เฟส ของเมชีนไซ้เกิดต่อมา โดยสัญญาณ SEND จะเป็น “0” ตลอดการส่งข้อมูล เมื่อสัญญาณ SEND แอกตีฟ จะทำการส่งบิตเริ่มต้นก่อนเป็นบิตแรก โดยมีคาบเวลาของบิตเริ่มต้นเท่ากับ 1 เมชีนไซ้เกิด จากนั้นตามด้วยการส่งบิตข้อมูล 8 บิต เรียงลำดับจากบิต LSB โดยข้อมูลที่ทำการแอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งถูกเรียงออกมาจากรีจิสเตอร์บีฟเฟอร์สำหรับการส่งข้อมูล ในทุก ๆ บิตข้อมูลที่ทำการส่งออกไปจะเกิดสัญญาณพัลส์ SHIFT ขึ้นเพื่อให้เรียงข้อมูลในแต่ละบิตจากรีจิสเตอร์ในบีฟเฟอร์การกำหนดจังหวะการส่งข้อมูลใช้สัญญาณนาฬิกาในการส่ง (TX clock) เป็นตัวกำหนด โดยสัญญาณนาฬิกานี้ได้มาจากการหารสัญญาณ TCLK จากไทเทอร์ 1 ด้วย หลังจากการส่งบิตข้อมูลก็จะทำการส่งบิตหยุดหรือบิตปิดท้าย 1 บิต ดังนั้นการส่งข้อมูลจะใช้สัญญาณนาฬิกาทั้งหมด 10 ลูก เมื่อทำการส่งข้อมูลครบเรียบร้อยแล้ว จะทำการเซตบิต TI ในรีจิสเตอร์ SCON หากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการเอ็นเอเบิล ไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ หลังจากที่ทำบริการอินเตอร์รัปต์หรือส่งข้อมูลเรียบร้อยแล้ว ต้องทำการเคลียร์บิต TI เป็นอันดับแรก เพื่อให้การรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

ด้านการรับข้อมูลจะทำการตรวจจับการเปลี่ยนแปลงระดับลอจิกจาก “1” ที่ขา RxD โดยใช้ อัตราการสุ่มเท่ากับ 1/6 เท่าของอัตราบอด เมื่อตรวจจับพบ ไทมเมอร์/เคาน์เตอร์ที่ใช้ในการกำหนดอัตราบอดจะรีเซตและทำการเขียนข้อมูล 1F6H ไปยังรีพรีจิสเตอร์ ข้อมูลจะเริ่มต้นเดินทางเข้าสู่พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ผ่านทางขา RxD ในการตีความว่าบิตที่เข้ามาเป็น “0” หรือ “1” จะใช้ผลการสุ่มสเตตที่ 7,8 และ 9 หาก 2 ใน 3 ของการสุ่มพบว่าข้อมูลเป็นลอจิกข้อใด จะตีความข้อมูลในบิตนั้นเป็นตามเสียงข้างมาก ยกตัวอย่าง สุ่มพบลอจิก “1” 2 ในตัว 3 ครั้ง จะตีความว่าบิตของข้อมูลที่ได้รับนั้นเป็น “1”

ลำดับของการรับข้อมูลมีลักษณะเกี่ยวกับการส่งข้อมูลคือ เริ่มด้วยบิตเริ่มต้นก่อน ตามด้วยบิตข้อมูล และบิตปิดท้าย ในทุก ๆ การรับข้อมูลได้ 1 บิต จะมีพัลส์ SHIFT เกิดขึ้น เพื่อทำการเลื่อนข้อมูลเข้าสู่รีจิสเตอร์บีฟเฟอร์การรับข้อมูล การกำหนดจังหวะการรับข้อมูลใช้สัญญาณนาฬิกาการรับข้อมูล (RX clock) หลังจากสัญญาณนาฬิกาถูกสุ่มสุดท้าย อันหมายถึงสามารถรับข้อมูลได้ครบแล้ว วงจรควบคุมการรับข้อมูลจะทำการส่งข้อมูลจากรีจิสเตอร์บีฟเฟอร์ไปยังรีจิสเตอร์ SBUF และบิต RB8 ในรีจิสเตอร์ SCON โดยข้อมูลในบิต RB8 ก็คือข้อมูลของบิตหยุดนั่นเอง พร้อมกันนั้นยังทำการเซต RI ในรีจิสเตอร์ SCON ด้วยหากการอินเตอร์รัปต์จากพอร์ตอนุกรมได้รับการเอ็นเอเบิลไว้ ก็จะเกิดการอินเตอร์รัปต์ขึ้นในระบบ หลังจากบริการอินเตอร์รัปต์หรือรับข้อมูลเรียบร้อยแล้วต้องทำการเคลียร์บิต RI ก่อน เพื่อให้การรับส่งข้อมูลทางพอร์ตอนุกรมดำเนินต่อไปได้

2.2 อัตราบอดของพอร์ตอนุกรมในไมโครคอนโทรลเลอร์ MCS-51 โหมด 1 และ 3

เนื่องจากทั้งสองโหมดนี้สามารถเลือกแหล่งกำเนิดอัตราบอดได้ 2 ตำแหน่งคือ จากอัตราโอเวอร์โฟลวของไทมเมอร์ 1 และ 2 สำหรับอัตราบอดเมื่อใช้การโอเวอร์โฟลวของไทมเมอร์ 1 จะต้องใช้ค่าของบิต SMOD ในรีจิสเตอร์ PCON มาพิจารณาประกอบด้วย สามารถคำนวณค่าอัตราบอดได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราบอด = (2 ค่าของบิท SMOD/32) × อัตราโอเวอร์โพลวของไทมเมอร์ 1

ถ้าหากในไทมเมอร์ 1 ไม่ได้เอ็นเอเบิลการอินเตอร์รัปต์ไว้ สามารถคำนวณค่าอัตราบอดได้จาก

อัตราบอด = (2 ค่าในรีจิสเตอร์ SMOD /32) × ความถี่สัญญาณนาฬิกา / {12 × [265 - (TH1)]}

ในกรณีที่ใช้ไทมเมอร์ 2 ในการกำหนดโหมดกำเนิดอัตราบอด (Baud rate generator)

สามารถคำนวณหาอัตราบอดได้

อัตราบอด = อัตราโอเวอร์โพลวของไทมเมอร์ 2/6 หน่วยเป็น บิตต่อวินาที

ถ้าหากกำหนดให้ไทมเมอร์ 2 ทำงานในโหมดปกติ สามารถคำนวณบอดได้จาก

อัตราบอด = ความถี่ของสัญญาณนาฬิกา / (32 × (65536 - (RCAP2H, RCAP2L)))

โดยที่ (RCAP2H, RCAP2L) เป็นค่าของรีจิสเตอร์ RCAP2H และ RCAP2L มีขนาด 16บิต

ไม่คิดเครื่องหมาย

2.3 การเขียนหรือส่งข้อมูลออกจากพอร์ตอนุกรม

ข้อมูลที่ต้องการส่งออกทุกค่าต้องนำไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์ของพอร์ตอนุกรมซึ่งก็คือรีจิสเตอร์ SBUF ดังตัวอย่าง

```
MOV SBUF, #'A'
```

จากคำสั่งข้างต้นเป็นการส่งข้อมูลตัวอักษร A ออกไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์อย่างไรก็ตามก่อนทำการส่งข้อมูลทุกครั้งต้องแน่ใจว่าบิท TI เคลียร์หรือมีข้อมูลเป็น "0" และเมื่อทำการส่งข้อมูลเรียบร้อยแล้ว ก็จะทำให้เกิดการเซตบิท TI เพื่อแจ้งให้ทราบ ดังตัวอย่าง

```
CLR TI ; เคลียร์บิต TI เพื่อเตรียมการส่งข้อมูลออก
```

```
MOV SBUF, #'A' ; ส่งข้อมูลของตัวอักษร A ไปยังพอร์ตอนุกรม
```

```
JNB TI, $ ; รอการเซตของบิต TI เพื่อแจ้งการส่งข้อมูลที่เสร็จ
```

2.4 การอ่านหรือรับข้อมูลจากพอร์ตอนุกรม

การรับข้อมูลจากพอร์ตอนุกรมสามารถกระทำได้ง่ายมาก เพียงทำการตรวจสอบว่าบิต RI เกิดการเซตเกิดขึ้นแล้ว ให้ทำการอ่านค่าจากรีจิสเตอร์ SBUF โดยต้องการทำการโอนย้ายข้อมูลผ่านทางแอกคิวมูลเตอร์หรือรีจิสเตอร์ A ดังตัวอย่าง

CLR RI ; เคลียร์บิต เพื่อเตรียมการส่งข้อมูล
 JUB RI, \$; รอคอยการเซตบิต อันเป็นการแจ้งให้ทราบว่า การรับข้อมูล
 ; เสร็จสมบูรณ์และมีข้อมูลเกิดขึ้นที่รีจิสเตอร์ SBUF
 MOV A, SBUF ; อ่านค่าจากรีจิสเตอร์ โดยการโอนย้ายข้อมูลผ่านทางรีจิสเตอร์ A
 CLR RI ; หลังจากทำการอ่านรีจิสเตอร์เรียบร้อยแล้ว ต้องทำการเคลียร์บิต RI

2.5 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานวงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS 51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ ± 3 ถึง ± 12 V ในขณะที่ระดับสัญญาณไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับที่ทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์โดยตรง จึงอาศัยการเชื่อมต่อผ่านทางไอซี พิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ ไอซีที่ทำหน้าที่ในการแปลงระดับสัญญาณนี้ต้องการแปลงข้อมูลส่งของไมโครคอนโทรลเลอร์ MCS-51จากระดับที่ทีแอลไปเป็นระดับของ RS-232 และทำการแปลงข้อมูลรับจากคอมพิวเตอร์จากระดับของ RS-232 เป็นระดับที่ทีแอลเพื่อให้สามารถถ่ายทอดไปยังไมโครคอนโทรลเลอร์ MCS-51ได้อย่างสมบูรณ์ ไอซีดังกล่าวมีด้วยกันหลายเบอร์จากผู้ผลิต อาทิ MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น

2.16 การติดต่อกับวิซวลเบสิก

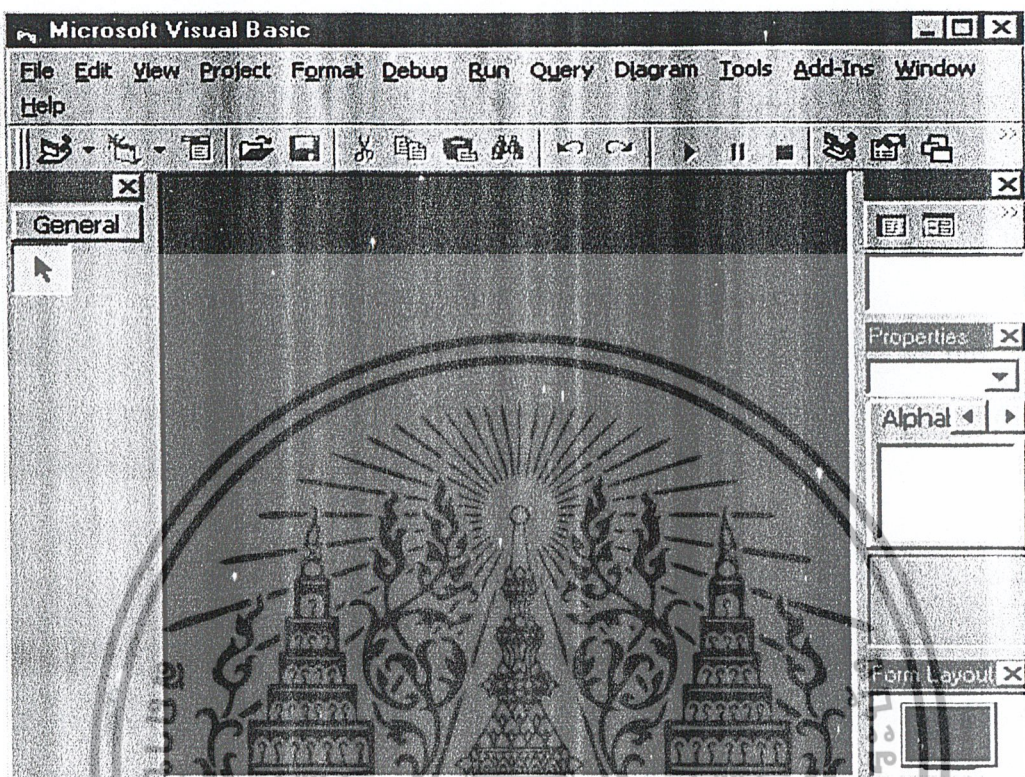
การเขียนโค้ด โปรแกรม และตัวแปลภาษา ตัวตรวจสอบความผิดพลาด ทั้งหมดที่กล่าวมานี้โปรแกรมวิซวลเบสิก แยกออกเป็น ส่วน ๆ จะแสดงดังรูปที่ 2.11

1. เมนูบาร์ เป็นส่วนแสดงคำสั่งของวิซวลเบสิก มีส่วนประกอบดังนี้

File ทำหน้าที่จัดการเพิ่มข้อมูลต่าง ๆ เช่น การสร้าง การจัดเก็บ

Edit ทำหน้าที่แก้ไขข้อมูลต่าง ๆ เช่น คัดลอก การยกเลิก

View ทำหน้าที่ซ่อนแสดง วินโดว์ (Windows) และ IDE ต่าง ๆ



รูปที่ 2.11 แสดงหน้าต่างการทำงานของโปรแกรมวิชวลเบสิก

Project ทำหน้าที่ลดเพิ่ม ฟอรัม (Form), แอกทีฟ-เอ็กซ์ (Active-x), คอนโทรล (Control) ต่าง ๆ

Debug ติดตามตรวจสอบความผิดพลาดของโปรแกรม

Run ทำหน้าที่สั่งให้โปรแกรมทำงาน, หยุดชั่วคราว, เลิกทำงาน

Tool ทำหน้าที่จัดการ โปรซีเจอร์ (Procedures) ออกแบบเปลี่ยนแปลงตัวเลือกต่าง ๆ

Add-Ins ทำหน้าที่เพิ่มเครื่องมือพิเศษมาจากผู้พัฒนารายอื่นหรือไม่โครซอฟต์แวร์

Windows ทำหน้าที่จัดการวินโดว์ ที่ใช้ในวิชวลเบสิก เช่นการวางตัววินโดว์

2. **Tool Bar** ทำหน้าที่แสดงคำสั่งในรูปแบบไอคอน (Icon) หรือสัญลักษณ์

3. **Tool Box** ทำหน้าที่เป็นส่วนที่เก็บเครื่องมือต่าง ๆ ที่ใช้ในการออกแบบฟอร์มของ วิชวล

เบสิก

4. **Project Explorer** ทำหน้าที่แสดงจำนวน ฟอรัม, โมดูล (Module) ในโปรเจกต์

5. **Form Designer** เป็นวินโดว์ที่ ทำหน้าที่กับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. **Properties** ทำหน้าที่แสดงคุณลักษณะของออบเจกต์ (Object) ต่างๆ บนฟอร์ม

7. **Code Edit windows** ทำหน้าที่ระบุโค้ด โปรแกรม กคF7

8. **Form Layout Windows** ทำหน้าที่แสดงตำแหน่งของฟอร์มเมื่อเริ่มทำงานครั้งแรก

2.16.1 การใช้งานโปรแกรมวิซวลเบสิก ติดต่อกับพอร์ตอนุกรม

ในการใช้งานโปรแกรมวิซวลเบสิก เพื่อติดต่อกับพอร์ตอนุกรมนั้นเป็นเรื่องที่ทำได้ง่ายมาก เพราะว่าทางผู้ผลิตโปรแกรมด้าน ไมโครซอฟต์ ได้จัดเตรียมเครื่องมือที่สามารถนำมาใช้เชื่อมต่อกับพอร์ตอนุกรมเอาไว้แล้ว สำหรับโปรแกรมวิซวลเบสิก เวอร์ชัน5 และเวอร์ชัน6 จะมีชื่อเรียกว่า MsComm32.ocx ซึ่งจัดทำขึ้นเพื่อใช้งานกับระบบปฏิบัติการ 32 บิตในวินโดวส์ 95 และ 98

2.16.2 วิธีการสื่อสารที่ใช้ใน MSComm Control

1. การสื่อสารด้วยการกระตุ้นจากเหตุการณ์ที่เกิดขึ้น (Even- Driven Communication) ซึ่งการสื่อสารลักษณะนี้จะมีประสิทธิภาพมากกว่าคือ MsComm32.ocx จะตอบสนองกับเหตุการณ์แบบทันทีทันใด คือจะเริ่มทำงานเมื่อพอร์ตอนุกรมถูกกระตุ้นทำให้เกิดการเปลี่ยนแปลงที่ขา Data Carrier Detect (DCD) หรือขา Request To Sent (RTS)

2. การสื่อสารด้วยการคอยตรวจสอบเหตุการณ์ (Polling) เป็นการคอยตรวจสอบค่าเหตุการณ์และค่าความผิดพลาดที่จะเกิดขึ้นในคุณสมบัติ Comm Event

2.16.3 การใช้งานของพอร์ตอนุกรม

1. การกำหนดหมายเลขของพอร์ต เช่น

MSComm.CommPort = 1 เป็นการ ใช้พอร์ตคอม1 (หมายเลขจะเป็นตัวเลขพอร์ตที่ใช้งาน)

2. กำหนดค่าเซตตั้ง (Settings) ซึ่งจะเป็นค่าที่จะใช้ในการติดต่อกับพอร์ตอนุกรม ประกอบด้วยความเร็วในการส่งข้อมูล, บิตพาริตี, ขนาดของข้อมูล, จำนวนบิตหยุด รูปแบบการกำหนดค่า ตัวอย่าง

MSComm1.Setting = 9600,N, 8,1 หมายถึงความเร็วในการส่งข้อมูลเท่ากับ 9600, ไม่มีบิตพาริตี, ข้อมูลเป็นแบบ8บิต, 1บิตหยุด

3. กำหนดค่า InputLen ซึ่งเป็นตัวอ่านค่าการอ่านข้อมูลของพอร์ตอนุกรมให้กำหนดค่าดังนี้ **MSComm1.InputLen = 0** เป็นการอ่านค่าทั้งหมดที่เข้ามาทางอินพุต

4. กำหนดค่าพอร์ต Port open ให้กำหนดดังนี้

MSComm1.Portopen = True เป็นการเปิดใช้งานพอร์ตอนุกรม

MSComm1.Portopen = False เป็นการปิดใช้งานพอร์ตอนุกรม

5. ในการเขียนโปรแกรมในการรับค่าพอร์ตอนุกรม ดังนี้

Data = MSComm1.Input เป็นการรับค่าจากพอร์ตอนุกรมมาเก็บไว้ในตัวแปรชื่อ Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ในการเขียนโปรแกรมรับส่งค่าไปยังพอร์ตอนุกรม ดังนี้

MSComm1.Out = Chr&(Data) รับส่งค่าข้อมูลที่อยู่ในตัวแปรชื่อ Data ไปยังพอร์ตอนุกรม

ค่าคงที่ของคุณสมบัติ MSComm32.ocx

ค่าคงที่	ค่าที่ได้	รายละเอียด
ComNone	0	ไม่ใช้ตรวจสอบ Handshake
ComXonZ.Xoff	1	ใช้การตรวจสอบ Handshake แบบ Xon/Xoff
ComRST	2	ใช้การตรวจสอบ Handshake ผ่านทางขา RTS และ CTS
ComRSTOnXoff	3	กำหนดการตรวจสอบ Handshake ทั้งแบบ RTS-CTS ,Xon/Xoff

รูปตารางที่ 2.5 ค่าคงที่สำหรับคุณสมบัติ Handshake

ค่าคงที่	ค่าที่ได้	รายละเอียด
ComEvSend	1	ส่งค่าเหตุการณ์ (Send Event)
ComEvReceive	2	รับค่าเหตุการณ์ (Receive Event)
ComEvCTS	3	มีการเปลี่ยนแปรที่ขา CTS
ComEvDSR	4	มีการเปลี่ยนแปรที่ขา DSR
ComEvCD	5	มีการเปลี่ยนแปรที่ขา DCD
ComEVRing	6	ตรวจรับสัญญาณกระดิ่งของโทรศัพท์
ComEvEOF	7	ตรวจพบตำแหน่งท้ายสุดของไฟล์ (End of file)

รูปตารางที่ 2.6 ค่าคงที่สำหรับคุณสมบัติ OnComm

ค่าคงที่	ค่าที่ได้	รายละเอียด
ComInputModeText	0	ข้อมูลที่รับมีคุณสมบัติเป็นข้อความ (ค่าปกติ)
ComInputModeBinary	1	ข้อมูลที่รับเข้ามาเป็นข้อมูลไบนารี

รูปตารางที่ 2.7 ค่าคงที่สำหรับคุณสมบัติ InputMode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าคงที่	ค่าที่ได้	รายละเอียด
ComEventBreak	1001	ได้รับสัญญาณ Break
ComEventCTSTO	1002	ขาเกิดไหม้เอาต์
ComEventDSRTO	1003	ขาเกิดไหม้เอาต์
ComEventFrame	1004	เกิดข้อผิดพลาดที่เฟรมข้อมูล (Framing error)
ComEventOverrun	1005	พอร์ตอนุกรมเกิดโอเวอร์รัน (Port Overtur)
ComEventCDTO	1006	ขาเกิดไหม้เอาต์
ComEventRxOver	1007	บัฟเฟอร์รับข้อมูลเกิด (Overflow)
ComEventRxParity	1008	เกิดข้อผิดพลาดที่พาริตี (Parity Error)
ComEventTxFull	1009	บัฟเฟอร์ส่งข้อมูลเต็ม
ComEventDCB	1010	เกิดข้อผิดพลาดขั้นที่กลไกการควบคุม DCB โดยที่ไม่ได้คาดคิด

รูปตารางที่ 2.8 ค่าคงที่สำหรับคุณสมบัติ Error

ข้อความที่บ่งชี้ถึงความผิดพลาด	ค่าที่ได้	คำอธิบายข้อความที่ปรากฏ
ComInvalid Property Value	380	กำหนดค่าไม่ถูกต้อง
ComSetNotSupported	383	ค่าที่ได้ตั้งไว้สามารถอ่านได้อย่างเดียว
ComGetNotSupported	394	ค่าที่ได้รับไว้สามารถอ่านได้อย่างเดียว
	8000	อ่านค่าไม่ได้ในขณะที่ Port ถูกเปิดใช้อยู่
ComPortInvalid	8001	ค่าของเวลาที่หาออกมาได้ต้องมีค่ามากกว่าศูนย์
	8002	หมายเลขของ Port ไม่ถูกต้อง
	8003	ผลลัพธ์จะเกิดขึ้นที่มีการทำงานเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อความที่บ่งชี้ถึงความผิดพลาด	ค่าที่ได้	คำอธิบายของข้อความที่ปรากฏ
ComPortAlreadyOpen	8004	Port จะสามารถอ่านค่าได้ในขณะมีการทำงานเท่านั้น
	8005	Port ได้ถูกเปิดเรียบร้อยแล้ว
	8006	อุปกรณ์เกิดการผิดพลาดหรือไม่สามารถรองรับค่าได้
	8007	อุปกรณ์ไม่ยอมรับค่าที่ Baud rate ตั้งเอาไว้
	8008	ขนาดของข้อมูลผิดพลาด
	8009	ค่าตัวแปรที่แสดงอยู่ผิดพลาด
	8010	Hardware ไม่พร้อมที่จะทำงาน (ถูกปกป้องโดยอุปกรณ์อื่น)
ComNoOpen	8011	ฟังก์ชัน ไม่สามารถกำหนดแถวของข้อมูลได้
	8012	Port ยังไม่พร้อมที่ถูกเปิดใช้งาน
	8013	Port พร้อมที่จะเปิดใช้งานได้
ComSetCommStateFailed	8014	Port ไม่สามารถทำงานได้
	8015	ไม่สามารถเซตสถานะของพอร์ตได้
ComPortNotOpen	8016	ไม่สามารถ SetPort ตามเหตุการณ์ที่กำหนดให้ได้
	8018	จะสามารถหาผลลัพธ์ได้ก็ต่อเมื่อ Port มีการทำงาน
ComReadError	8019	Port ไม่ว่างมีข้อมูลเต็มใน Port
ComDCBError	8020	เกิดความผิดพลาดขณะที่อ่านค่า

รูปตารางที่ 2.9 ข้อความที่บ่งชี้ถึงค่าความผิดพลาดที่เกิดขึ้น MSComm Control

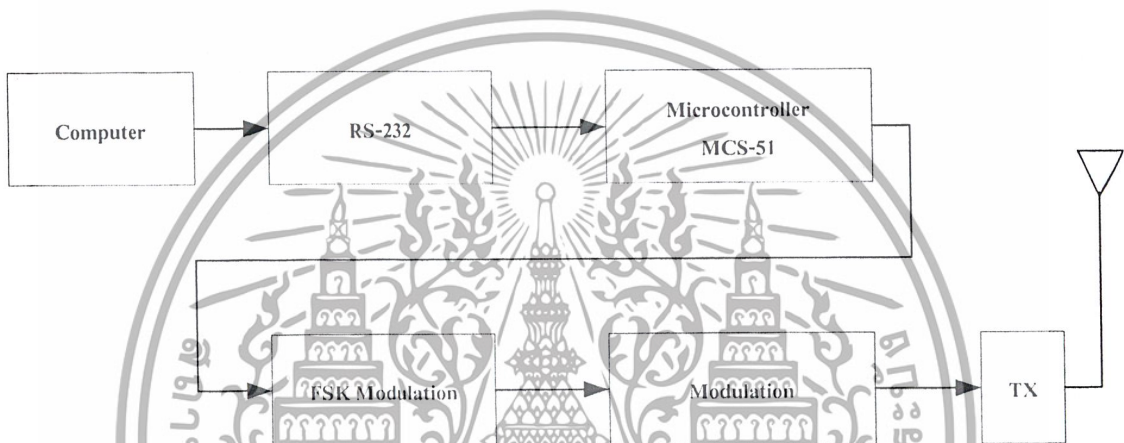
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

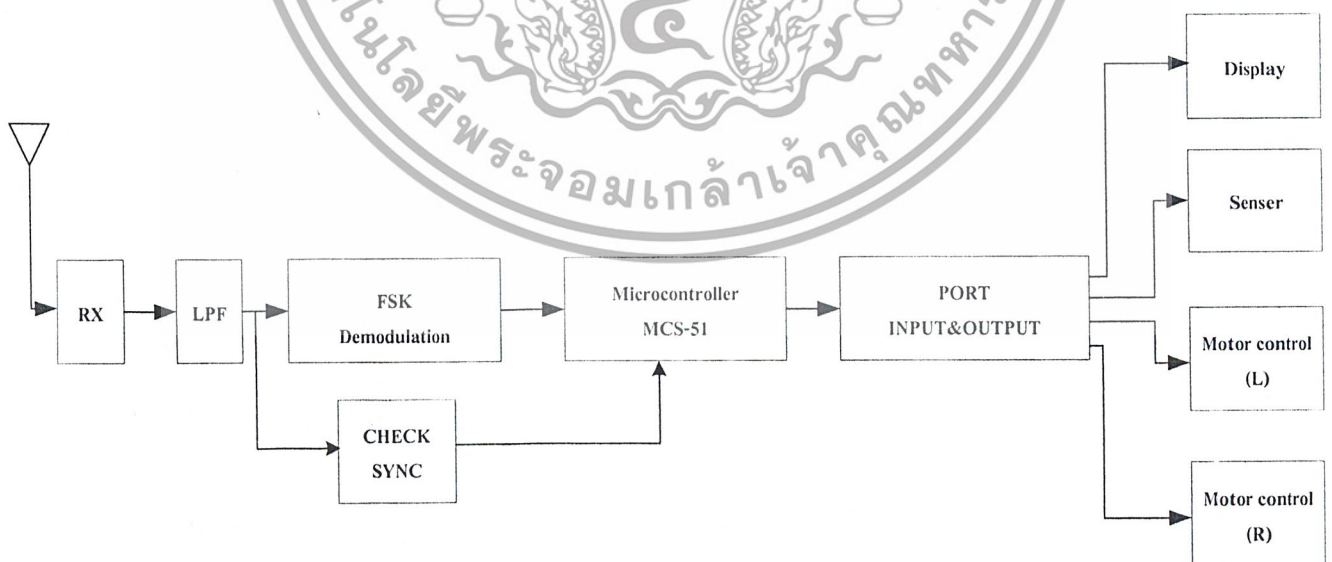
หลักการการทำงานและการออกแบบวงจร

3.1 บทนำ

ในบทนี้จะกล่าวถึงการทำงานของวงจร โดยวงจรทั้งหมดของโครงการจะแสดงเป็นบล็อกไดอะแกรมได้ดังรูป

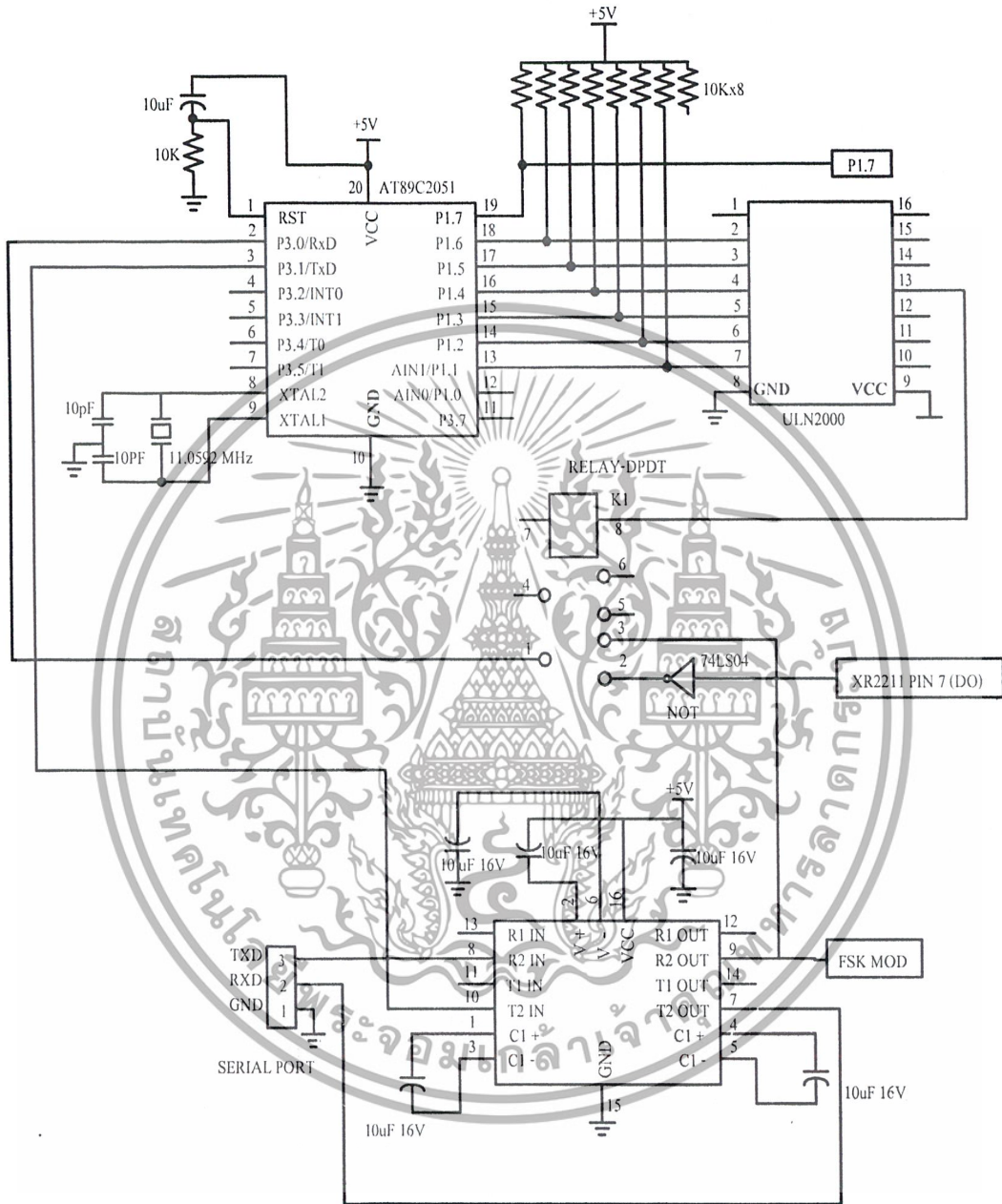


รูปที่ 3.1 บล็อกไดอะแกรมของโครงการด้านส่ง



รูปที่ 3.2 บล็อกไดอะแกรมของโครงการด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรแปลงระดับสัญญาณ RS-232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ไมโครคอนโทรลเลอร์ด้านส่งข้อมูล

จากรูปที่ 3.3 สัญญาณ TTL ที่ได้จากวงจรแปลงระดับสัญญาณ RS-232 จะเป็นสัญญาณอินพุตเข้าที่ขา RXD ของวงจรไมโครคอนโทรลเลอร์ซึ่งเป็นการรับข้อมูลแบบอนุกรมเข้ามาเพื่อตรวจเช็คสถานะการส่งข้อมูลของโมเด็ม

3.2.4 การทำงานของ FSK Modulator

XR2206

รายละเอียดทั่วไปของ XR2206

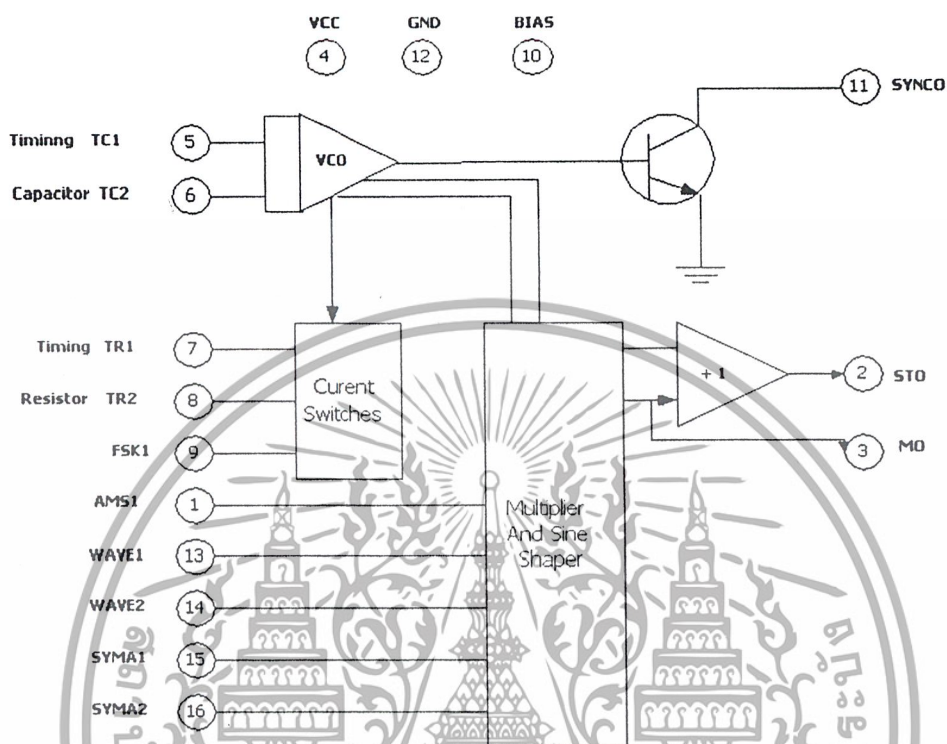
XR2206 เป็นอุปกรณ์โมโนลิทิกฟังก์ชันเจนเนอเรเตอร์ (Monolithic function generator) ใช้สำหรับสร้างสัญญาณไซน์ (Sine), สามเหลี่ยม (Triangle), สี่เหลี่ยม (Square), แรมป์ (Ramp), พัลส์ (Pulse) ที่มีคุณภาพ เสถียรภาพ และความเที่ยงตรงสูง รูปคลื่นเอาต์พุตสามารถมอดูเลตได้ทั้งทางขนาด และทางความถี่ ก้านแรงดันภายนอก โดยมีช่วงความถี่ใช้งานตั้งแต่ 0.01 เฮิร์ตซ์ ถึงมากกว่า 1 เมกะเฮิร์ตซ์ ไอซีเบอร์นี้เหมาะสำหรับการใช้งานในระบบสื่อสาร ในระบบเครื่องมือวัด และระบบที่ต้องการแหล่งกำเนิดสัญญาณไซน์ เอเอ็ม เอฟเอ็ม หรือสัญญาณที่ผ่านการเข้ารหัสแบบเลื่อนความถี่ มีค่าเฉลี่ยเบี่ยงเบนของอุณหภูมิเป็น 20 ppm/องศาเซลเซียส ความถี่ ออสซิลเลเตอร์สามารถกวาดเป็นเชิงเส้นได้สูงกว่า 2000:1 ช่วงความถี่โดยการควบคุมของแรงดันภายนอกในขณะที่มีความเพี้ยนต่ำ

ลักษณะสำคัญของ XR2206

ความเพี้ยนของคลื่นไซน์ต่ำ	0.5%, typical
เสถียรภาพทางอุณหภูมิสูง	20 ppm/°C Typical
ช่วงกวาดของความถี่กว้าง	2000:1 Typical
ความไวต่อไฟเลี้ยงต่ำ	0.01% Typical
ช่วงกว้างของไฟเลี้ยง	10 V ถึง 99%
คิวตี้ไซเคิล (Duty cycle)	1 % ถึง 99%
กำลังสูงสุดสัมบูรณ์	
ไฟเลี้ยงวงจร	26V
กำลังสูญเสีย	750mW
การสูญเสียเหนือ 25°C	5mW / °C
ช่วงอุณหภูมิใช้งาน	-65 °C ถึง +150°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างลักษณะหน้าที่การทำงานของ XR2206



รูปที่ 3.4 บล็อกโคเอเรทของ XR2206

รายละเอียดของระบบของ XR2206

XR2206 ประกอบด้วย 4 ภาคการทำงาน ได้แก่ ส่วนแรงดันควบคุมการทำงาน ของออสซิลเลเตอร์ (Voltage Controlled Oscillator: VCO) วงจรคูณอนาล็อก วงจรปรับสัญญาณชานน์ บัฟเฟอร์ (Buffer) และ สวิตช์กระแส ส่วนแรงดันควบคุมการทำงานของออสซิลเลเตอร์ จะผลิตเอาท์พุทของความถี่เป็นสัดส่วนกับกระแสอินพุท ซึ่งสามารถกำหนดได้ โดยตัวต้านที่ต่อเข้าที่ขาที่เกี่ยวข้องกับเวลาลงกราวด์ เนื่องจากมี 2 ขาที่เกี่ยวข้องกับเวลา ดังนั้นจึงสามารถสร้างสัญญาณได้ 2 ความถี่ สำหรับการกำเนิดสัญญาณเอฟเอสเค โดยใช้ขาควบคุมสัญญาณเอฟเอสเค (FSK Input control) ค่านี้จะไปควบคุมวงจรภายในส่วนสวิตช์ กระแสเพื่อให้เกิดการเลือกขาที่ต่อตัวต้านทานสำหรับวงจรในส่วนของแรงดันควบคุมการทำงานของออสซิลเลเตอร์

รายละเอียดการใช้งาน XR2206

การเข้ารหัสแบบเฟสเค XR2206 สามารถใช้งานโดยตัวต้านทาน (Timing resistor) 2 ตัวแยกกัน(R1, R2) ที่ขา 7 และที่ขา 8 ของ XR2206 โดยตัวต้านทานแต่ละตัว จะถูกใช้นั้นขึ้นอยู่กับสัญญาณ ลอจิกที่เข้ามาที่ขา 9 ของXR2206 หากขา 9 เปิดวงจรอยู่หรือต่อสัญญาณที่แรงดันสูงกว่า หรือเท่ากับ 2 โวลต์ และ R1 จะเป็นตัวต้านทานกำหนดเวลา ทำนองเดียวกันขา 9 ต่ออยู่กับแรงดันไฟที่ต่ำกว่าหรือเท่ากับ 1 โวลต์ แล้ว R2 จะเป็นตัวต้านทานกำหนดเวลา ดังนั้นความถี่ของสัญญาณเอาต์พุตสามารถเข้ารหัส 2 ระดับที่มีความถี่เป็น f1 และ f2 โดยที่

$$f_1 = \frac{1}{R_1 * C}$$

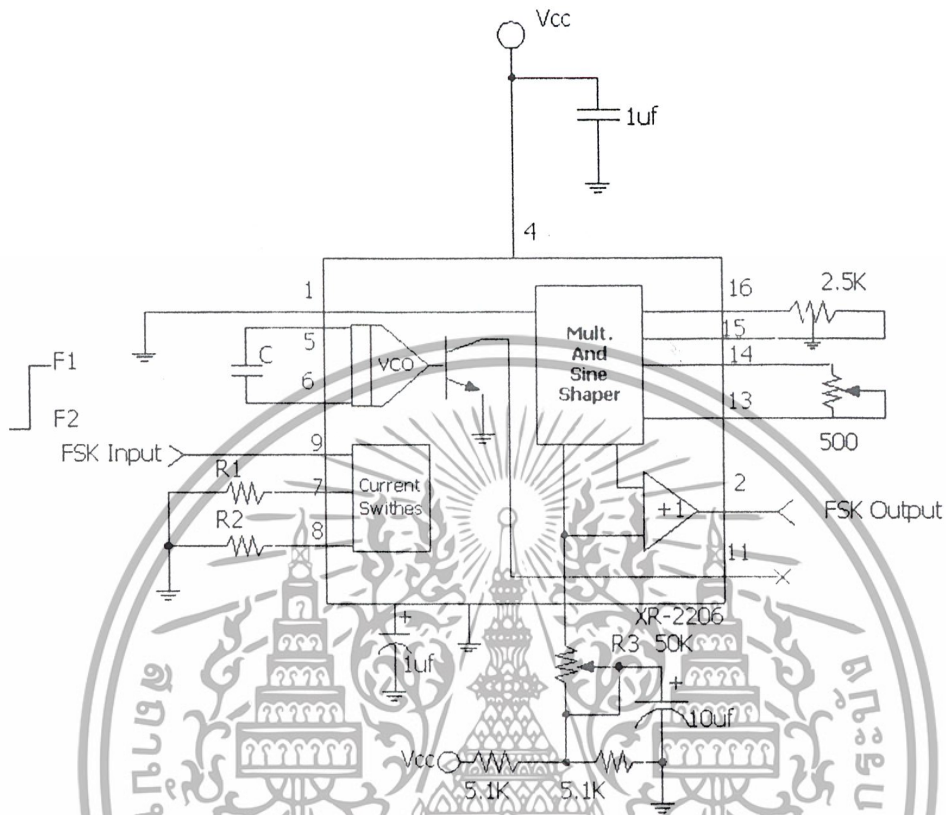
$$f_2 = \frac{1}{R_2 * C}$$

การควบคุมระดับแรงดันไฟตรงของเอาต์พุต

ระดับแรงดันไฟตรงของเอาต์พุตที่ขา 2 ประมาณเท่ากับแรงดันไฟตรงที่ไบอัสเข้าที่ขา 3 ของไอซี ซึ่งจากรูปบล็อกโคอะแกรมถูกไบอัสด้วยครึ่งหนึ่งของแรงดัน V+ เทียบกับกราวด์ดังนั้นจะได้ค่าแรงดันไฟตรงที่เอาต์พุตเท่ากับ V/2

การปรับละเอียดรูปคลื่นเอาต์พุต

ฮาร์โมนิกที่เกิดขึ้นของคลื่นไซน์สามารถลดลงเหลือ 0.5% ได้โดยเพิ่มตัวต้านทานปรับค่าได้ประมาณ 500 โอห์ม ที่ขา 13 และขา 14 ของไอซี และความต้านทานปรับค่าได้ประมาณ 2500 โอห์ม ที่ขา 15 และ 16 จากนั้นจึง ทำการปรับค่าความต้านทานที่ขา 15 , 16 โดยให้แท้ปมีค่าความต้านทานอยู่ตรงกลาง แล้วปรับตัวต้านทาน 500โอห์มจนได้ความเพี้ยนต่ำ สุดแล้วจึงทำการปรับตัวต้านทาน 2500โอห์มเพิ่มเติมเพื่อให้ได้ความเพี้ยนของสัญญาณต่ำที่สุด



รูปที่ 3.5 วงจรกำเนิดสัญญาณฟรีควอนซ์ซีฟดีอิ่ง

แนวทางการออกแบบ

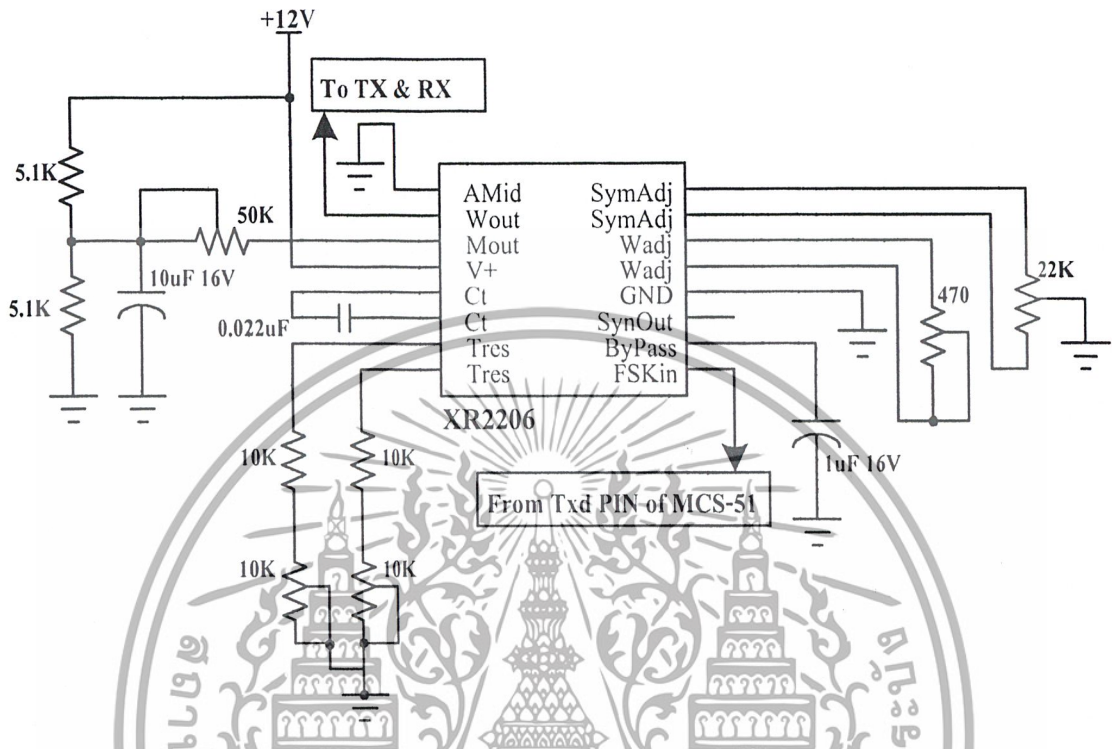
สำหรับการทดสอบช่วงแรกนั้นใช้ค่าความถี่สเปซ 3000 เฮิร์ตซ์และความถี่มาร์ค 2000 เฮิร์ตซ์ ดังนั้นจะได้ค่าอุปกรณ์ต่างๆ ดังนี้

$$R1 = \frac{1}{f1 * C} \qquad f1 = 3000 \text{ เฮิร์ตซ์}$$

$$R2 = \frac{1}{f2 * C} \qquad f2 = 2000 \text{ เฮิร์ตซ์}$$

เลือกค่า C = 22 นาโนฟารัด แล้ว R1 = 20 กิโลโอห์ม R2 = 45 กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 วงจรมอดูเลตแบบฟรีควีนซ์ซีพียู

3.3 การออกแบบส่วนรับข้อมูล

3.3.1 FSK ดิมอดูเลเตอร์

XR2211

ลักษณะทั่วไปของ XR2211 เป็นวงจรมอดูเลตแบบโมโนลิธิค (Monolithic) ซึ่งออกแบบสำหรับการใช้งานทางการสื่อสารข้อมูล โดยเฉพาะลักษณะพิเศษของไอซีเบอร์นี้เหมาะสำหรับเป็นตัวใช้งานเป็นตัวถอดรหัสแบบฟรีควีนซ์ซีพียู ไอซีทำงานในช่วงกว้างของไฟเลี้ยงคือ 4.5 โวลต์ ถึง 20 โวลต์ และมีช่วงความถี่ที่กว้างโดยอยู่ในช่วง 0.01 เฮิรตซ์ ถึง 300 กิโลเฮิรตซ์ สามารถใช้สัญญาณอนาลอกได้ในช่วง 2 มิลลิโวลต์ ถึง 3 โวลต์ อีกทั้งยังสามารถอินเทอร์เฟซได้กับทั้งวงจรถอดจิกตระกูลดีทีแอล (DTL), ทีทีแอล (TTL) และอีซีแอล (ECL) วงจรภายในประกอบด้วยเฟสล็อกสำหรับติดตามสัญญาณอินพุตในช่วงแถบผ่าน, วงจรครอดครเเจอร์เฟสดีเท็คเตอร์ (Quadrature Phase Detector) ใช้สำหรับตรวจสอบสัญญาณพาหะและตัวเปรียบเทียบแรงดันฟรีควีนซ์ซีพียู (FSK Voltage Comparator) ใช้สำหรับดิมอดูเลตสัญญาณฟรีควีนซ์ซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับอุปกรณ์ที่ต่อภายนอกกำหนดได้อย่างอิสระ แรงดันอ้างอิงภายในเป็นสัดส่วนกับไฟเลี้ยงวงจร ดังนั้นเหมาะสำหรับระบบที่มีการเปลี่ยนแปลงสมรรถภาพต่ำเมื่อ ไฟเลี้ยงวงจรเปลี่ยนแปลง

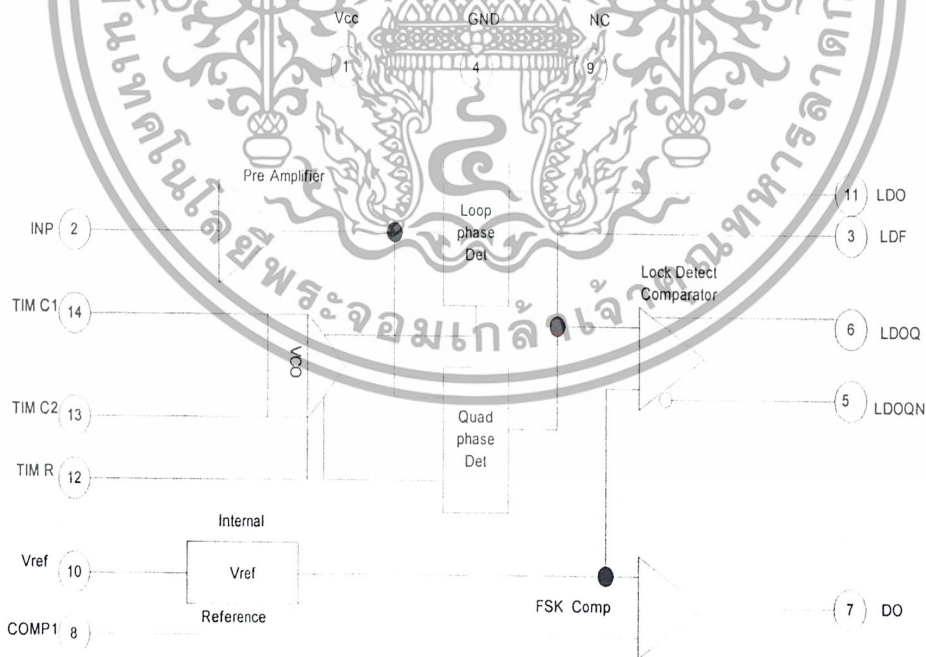
ลักษณะสำคัญของ XR2211

ช่วงกว้างของความถี่	0.01 ถึง 300 KHz
ช่วงกว้างของไฟเลี้ยงวงจร	4.5 ถึง 20 V
ช่วงกว้างทางพลศาสตร์	2 m V ถึง 2 Vrms
ช่วงของการตามสัญญาณ (Tracking) สามารถปรับได้	+/-1 ถึง +/-80%
เสถียรภาพอุณหภูมิสูง	20 ppm/°C

อัตราสูงสุดสัมบูรณ์

แหล่งจ่ายแรงดัน	20V
ระดับสัญญาณอินพุต	3 Vrms
กำลังงานสูญเสีย	900 mW

บล็อกไดอะแกรมของ XR2211



รูปที่ 3.7 บล็อกไดอะแกรมของ XR2211

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การถอดรหัสสัญญาณที่เข้ารหัสแบบเลื่อนความถี่เอฟเอสเคตามรูปแสดงการต่อพื้นฐานของ วงจรถอดรหัสแบบเอฟเอสเคโดยอ้างอิงจากรูป บล็อกไดอะแกรม และรูปข้างบน หน้าที่ของ อุปกรณ์ภายนอกแต่ละตัวมีดังนี้ R0 และ C0 กำหนดความถี่ศูนย์กลางของเฟสล็อกกลูป R1 กำหนด แถบความถี่ของระบบ C1 ใช้กำหนดค่าเวลาในวงจรรองความถี่แบบวนรอบ(Loop Damping) สำหรับข้อมูลเอาท์พุทของสัญญาณเอฟเอสเค ($R_b=510k\Omega$) จากขา 7 ไป ขา 8 มีไว้เพื่อเป็นการป้องกัน กลับแบบบวกสำหรับตัวเปรียบเทียบแรงดันของสัญญาณเอฟเอสเค เพื่อให้เกิดความเร็วในการ เปลี่ยนสถานะลอจิก

แนวทางการออกแบบ

วงจรตามรูปที่แสดงการต่อใช้งานวงจรดีมอดคูเลทเตอร์แบบเอฟเอสเค สามารถใช้กับวงจร ถอดรหัสเอฟเอสเค ทั่ว ๆ ไป โดยการเลือกอุปกรณ์ต่อภายนอกสำหรับการกำหนดความถี่มาร์ค และ ความถี่สเปซ (f_1, f_2) ค่าพารามิเตอร์สามารถคำนวณได้จากค่าความถี่ศูนย์กลางของเฟสล็อกกลูป : f_0

$$f_0 = (f_1 + f_2) / 2$$

เลือกค่าของตัวต้านทานกำหนดเวลา R0 ให้อยู่ในช่วง 10 กิโลโอห์ม ถึง 100 กิโลโอห์ม โดย สามารถเลือกได้ตามใจชอบสำหรับค่าอ้างอิงของ R0 เป็น 20 กิโลโอห์ม ซึ่งต่อความต้านทานปรับ ค่าได้ไว้สำหรับปรับค่าละเอียด(R_x)

คำนวณค่าของ C0 จากสมการ

$$C_0 = 1 / (f_0 * R_0)$$

คำนวณ ค่า R1 เพื่อกำหนด ความเบี่ยงเบนของความถี่มาร์ค และความถี่สเปซ

$$R_1 = (R_0 * f_0) / (f_1 - f_2)$$

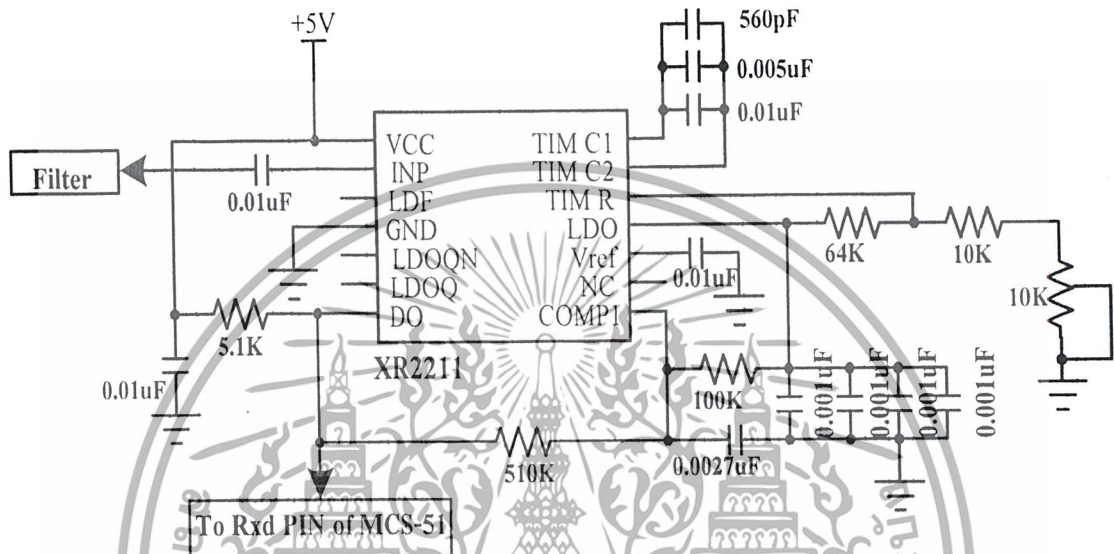
คำนวณค่า C1 เพื่อกำหนดการวนรอบ โดยการวนรอบมีค่าอ้างอิงเท่ากับ 1/2

$$C_1 = C_0 / 4$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณตัวเก็บประจุกรองข้อมูล : Cf สำหรับ Rf =100 กิโลโอห์ม, Rb =510 กิโลโอห์ม

$$C_f(\mu F) = 3 / (\text{Baud rate})$$



รูปที่ 3.9 วงจรดีมอดคูเลเตอร์แบบพีเรเนียนซีฟคีย์อิง

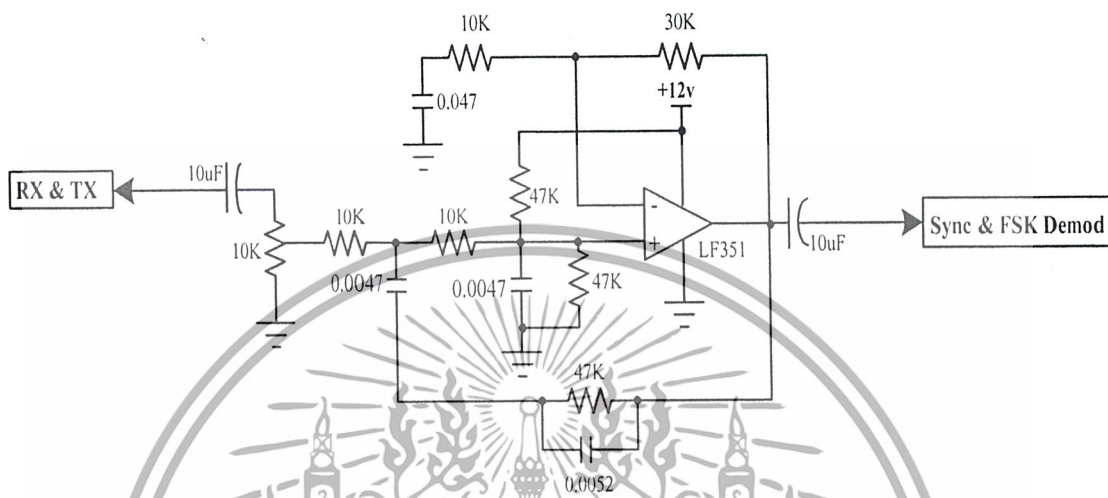
สัญญาณข้อมูลอนุกรมที่ดีมอดคูเลทได้จากวงจร XR2211 จะเข้าสู่ขา RXD ของไมโครคอนโทรลเลอร์ ซึ่งเป็นขารับข้อมูลแบบอนุกรมเพื่อที่จะนำข้อมูลที่ได้ไปประมวลผลและสั่งงานควบคุมอุปกรณ์อื่นๆ ต่อไป

3.4 หลักการทำงานของวงจร LPF

อินพุทของวงจร LPF จะผ่านเข้าที่ R3 ซึ่งทำหน้าที่ปรับขนาดของสัญญาณอินพุทให้เหมาะสม ก่อนที่ส่งผ่านวงจร LPF เนื่องจากไฟเลี้ยงวงจรเป็นไฟเลี้ยงเพียงด้านเดียว (โดยที่อีกด้านต่อกับกราวด์) ดังนั้นการต่อวงจรในลักษณะนี้ แรงดันที่เอาท์พุทของออปแอมป์ในขณะที่ไม่มีสัญญาณอินพุท ควรมีระดับอยู่ที่ $1/2(+V)$ จากรูปวงจรเราใช้หลักการแบ่งแรงดันโดยให้ R18 กับ R19เป็นตัวกำหนดจุดสวิงของสัญญาณ เมื่อมีสัญญาณไฟสลับเข้าที่อินพุทสัญญาณที่เอาท์พุทจะสามารถสวิงได้สูงสุดโดยจะสวิงจากจุด $1/2(V)$ ไปยัง $(+V)$ และ $1/2(V)$ ไปยังกราวด์แทนที่จะสวิงระหว่าง 0 โวลต์ ไป $\pm V$ โวลต์เช่นวงจรที่ใช้ไฟเลี้ยง 2 ชุด R7,R8,C24 และ C25 ทำหน้าที่กำหนดจุดคัทออฟของความถี่ โดยคำนวณจาก $f_c = 1 / (2\pi RC)$ เมื่อ R7 = R8 และ C24 = C25 ส่วน R12 กับ R6 จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นตัวกำหนดอัตราขยายสัญญาณของวงจร C18 ทำหน้าที่ป้องกันไฟกระชากตรงไม่ให้ผ่านไปยังวงจรถัดไป

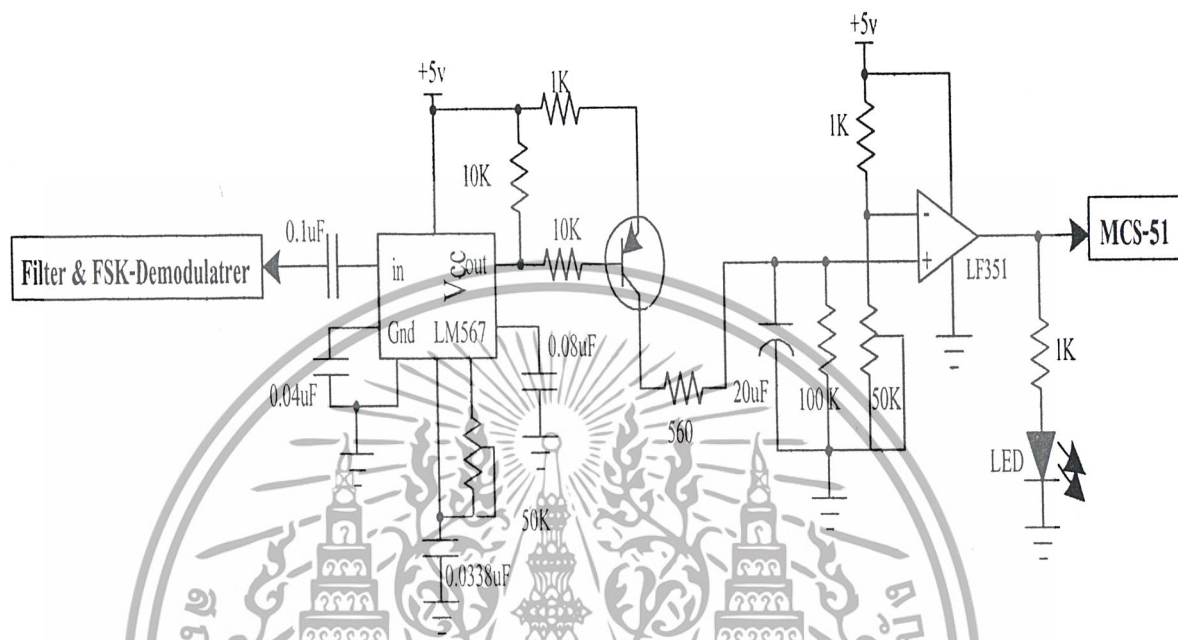


รูปที่ 3.10 วงจรการทำงาน Filter

3.5 หลักการทำงานของวงจรเช็กลสัญญาณ SYNC

ในการออกแบบได้ใช้ IC LM567 เป็นวงจรเฟสล็อก โดยที่สัญญาณอินพุทของวงจรเฟสล็อกจะได้อมาจากสัญญาณเอาต์พุทของวงจร LPF เมื่อมีสัญญาณอินพุทเข้ามาที่ขา 3 ของ IC LM567 วงจรจะทำการเปรียบเทียบเฟสและความถี่สัญญาณจากวงจร VCO ที่ประกอบอยู่ภายในตัว IC LM567 โดยที่ความถี่ที่ต้องการล็อก (f_o) จะได้จากความต้านทานที่ต่ออยู่ที่ขา 5 กับค่าของตัวเก็บประจุที่ต่ออยู่ที่ขา 6 เมื่อสัญญาณอินพุทที่เข้ามามีความถี่เท่ากับ f_o หรือใกล้เคียง ซึ่งวงจรสามารถล็อกความถี่ได้จะทำให้เอาต์พุทที่ขา 8 มีค่าแรงดัน ศูนย์ โวลต์ (สถานะที่ล็อกไม่ได้จะมีค่าแรงดันประมาณ 5 โวลต์) ซึ่งมีผลทำให้ Q5 ทำงาน เมื่อ Q5 นำกระแสจะทำให้แรงดันที่ขา Non Inverting ของออปแอมป์มีค่ามากกว่าแรงดันที่ขา Inverting ทำให้เอาต์พุทที่ขา 6 มีแรงดันเป็น + 5 โวลต์ที่สัญญาณดังกล่าวนี้จะนำไปต่อเข้าที่พอร์ตของไมโครคอนโทรลเลอร์เพื่อนำไปเช็กสถานะการรับข้อมูล

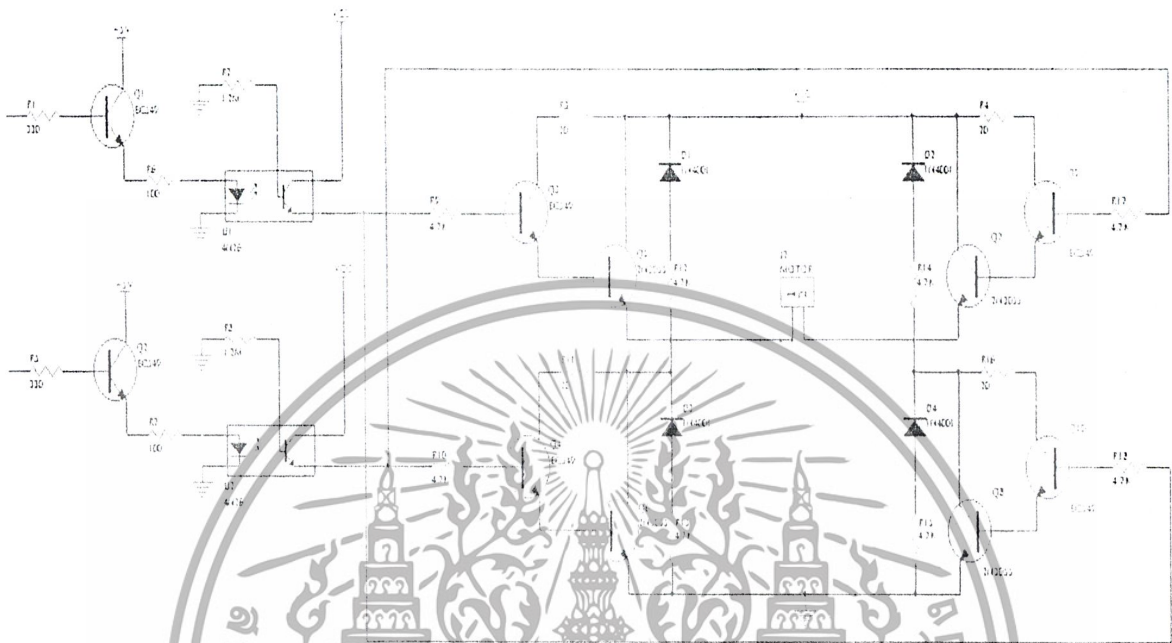
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 วงจรเช็คสัญญาณ Sync

3.6 วงจรขับมอเตอร์

วงจรขับมอเตอร์ถูกควบคุมจากระดับลอจิกของพอร์ตไมโครคอนโทรลเลอร์ในการควบคุมให้วงจรทำงานจะต้องส่งสถานะลอจิก ออกมาเป็นคู่เช่นอาจจะเป็นลอจิก 1 หรือลอจิก 0 เมื่อเราป้อนสถานะลอจิก 1 ให้กับ Q1 โดยที่ Q2 ได้รับ ลอจิก 0 จะทำให้ Q1 ทำงาน (Q2 ไม่ทำงาน) เป็นผลให้ไดโอดเปล่งแสงภายในตัว IC U1 ได้รับไบอัสตรง และไฟโด้ทราานซิสเตอร์ที่อยู่ภายใน IC U1 ก็จะทำงานเช่นกัน ทำให้ Q3 กับ Q5 นำกระแสผ่านขั้วไฟบวกของมอเตอร์ กระแสที่ขา 4 ของ IC U1 อีกส่วนจะนำมาเป็นกระแสไบอัสให้ Q10 กับ Q8 นำกระแสทำให้มีกระแสจากแหล่งจ่ายไฟ Vcc ไหลผ่านขั้วไฟบวกของมอเตอร์ ออกที่ขั้วไฟลบครบวงจรลงกราวด์ทำให้มอเตอร์หมุนได้



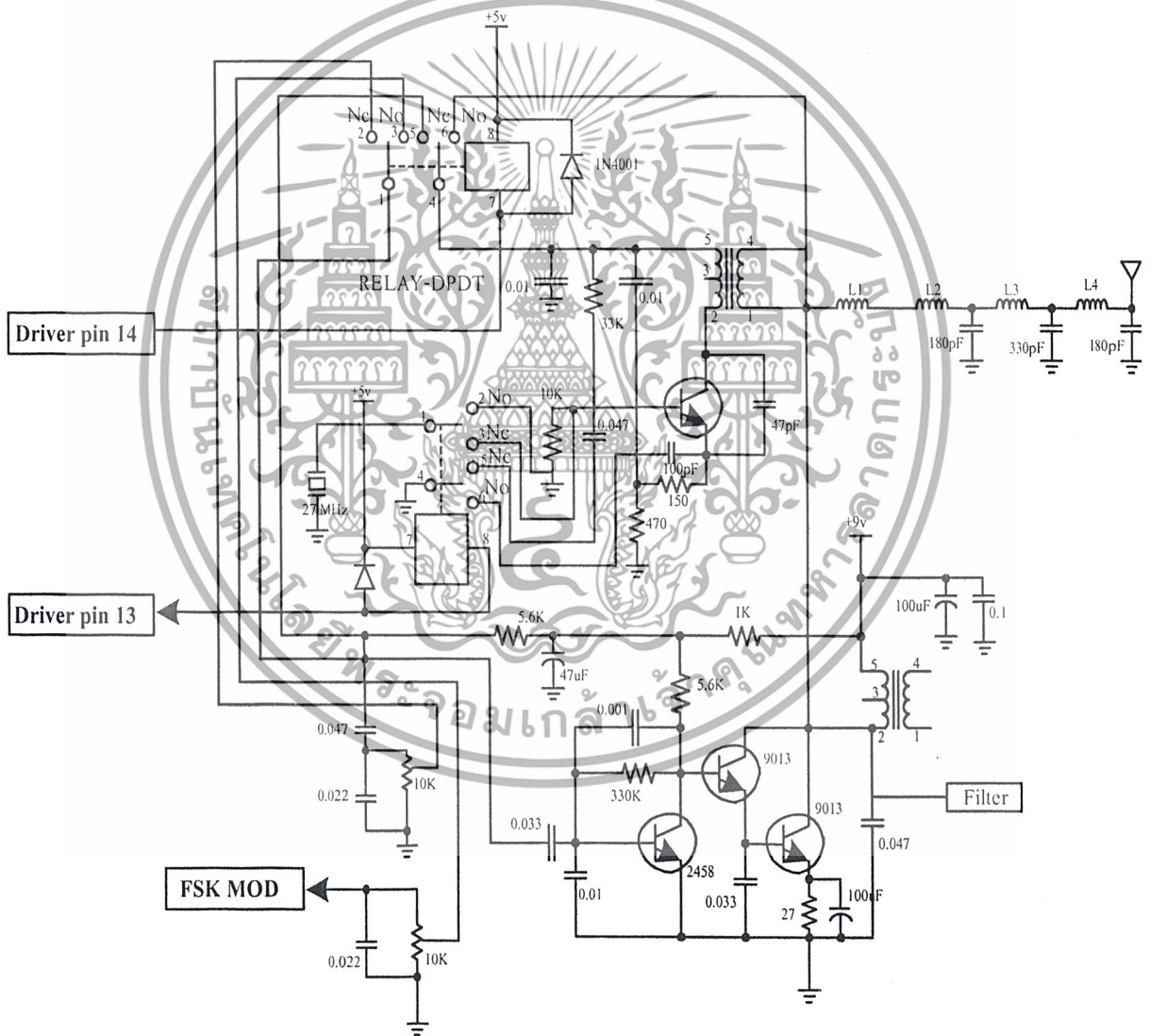
รูปที่ 3.12 วงจรขับมอเตอร์

3.7 วงจรไมโครคอนโทรลเลอร์ทางด้านรับ

เนื่องจากสัญญาณ โลกิกที่ออกจากวงจรถับข้อมูลสัญญาณ FSK มีสถานะกลับเฟส 180° กับข้อมูลจริงดังนั้น เราจะนำเอาสัญญาณดังกล่าวเข้าสู่วงจรถับกลับเฟสสัญญาณ 180° ก่อนที่จะส่งเข้าไปที่ขา RXD ของไมโครคอนโทรลเลอร์ โดยจะให้พอร์ต 0 เป็นพอร์ตที่ติดต่อกับ LCD เพื่อแสดงผลทางกราฟฟิก พอร์ต P2.0 จะต่ออยู่กับ IC DS1820 ซึ่งเป็น IC เซ็นเซอร์อุณหภูมิ พอร์ต 1 ใช้ควบคุมวงจรถับมอเตอร์พอร์ต P3.2 จะต่ออยู่กับ SENSOR เพื่อใช้เซ็นเซอร์ระยะทางของ Robot ซึ่งขบวนการเหล่านี้จะใช้การเขียน โปรแกรมควบคุม

3.9 หลักการทำงานของวงจรเครื่องรับ

สัญญาณที่สายอากาศรับเข้ามาได้ จะถูกจูน โดย T1 ให้ได้ความถี่ตรงกันกับทางด้านส่ง โดยมี Q1 ทำหน้าที่รับสัญญาณความถี่ที่ 27 MHz นี้สัญญาณที่รับได้จะส่งมาที่ R2 ซึ่งเป็นตัวต้านที่ปรับค่าได้ทำหน้าที่ปรับขนาดของสัญญาณที่รับได้ให้เหมาะสมก่อนส่งผ่านเข้าสู่วงจรเพื่อขยายสัญญาณให้แรงขึ้น สัญญาณที่ได้รับการขยายแล้วจะถูกป้อนไปยัง C7 ซึ่งทำหน้าที่ป้องกันไฟกระแสดตรงไม่ให้ผ่านเข้าไปสู่วงจรถัดไป



รูปที่ 3.14 วงจรเครื่องรับและเครื่องส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.10 ส่วนของเซนเซอร์ระยะทางการเคลื่อนที่

ส่วนการออกแบบการตรวจจับ (Sensor) ระยะทางการเคลื่อนที่นั้นจะอาศัยการออกแบบการนับจำนวนพัลส์ของการตรวจจับรูผ่านแสงเล็ก ๆ ที่ติดกับเพลตล้อ ซึ่ง 1 พัลส์ ข องสัญญาณที่ผ่านรูของแสงจะมีค่าเท่ากับ ประมาณ 2.5 เซนติเมตร การออกแบบการตรวจจับของรูผ่านแสงนั้นได้ อาศัย การทำงานของออป โต้คับเปิ้ล ซึ่งแสดงการต่อวงจรดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

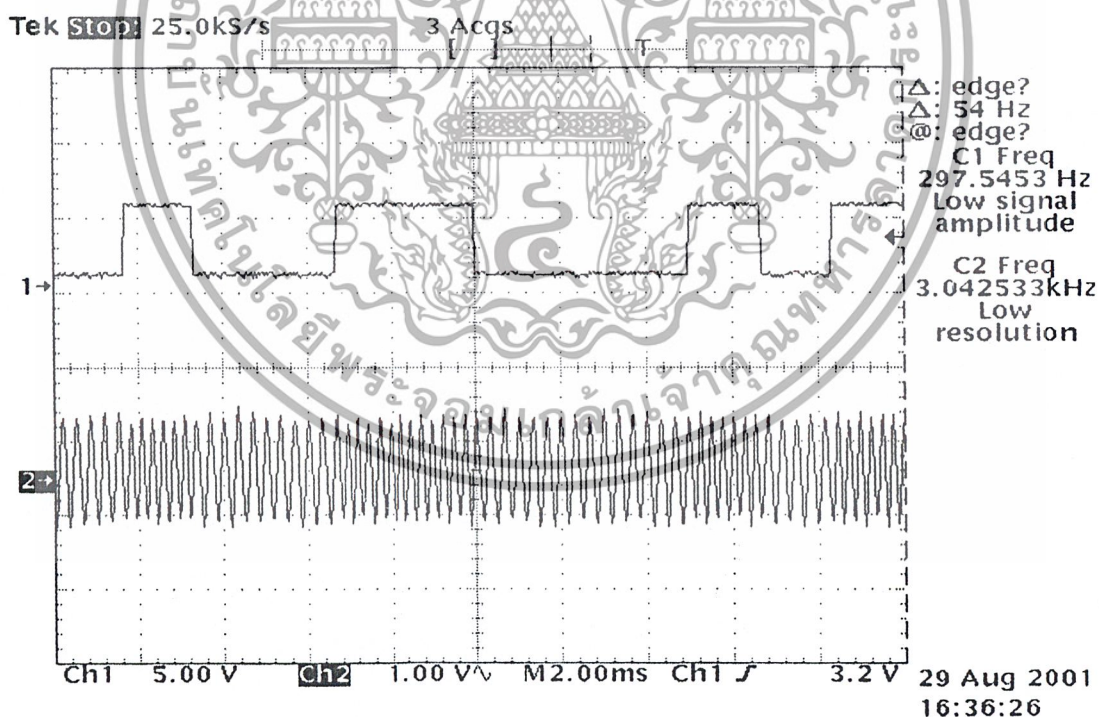
ผลการทดลอง

4.1 บทนำ

ในการทดลองวงจรจะทำการวัดสัญญาณของวงจรทางภาคส่งและภาครับที่ได้ออกแบบไว้ในโครงการซึ่งแบ่งออกเป็นส่วนๆ เมื่อแต่ละส่วนทำงานตามที่ได้ออกแบบไว้แล้วจึงนำมาต่อใช้งานร่วมกันเพื่อให้ได้ผลตามวัตถุประสงค์ที่ต้องการอีกที ซึ่งผลการวัดที่ได้มีดังนี้

4.2 สัญญาณดิจิทัลที่ต้องการส่ง กับสัญญาณเอาท์พุทของวงจร FSK Modulator ด้านส่ง

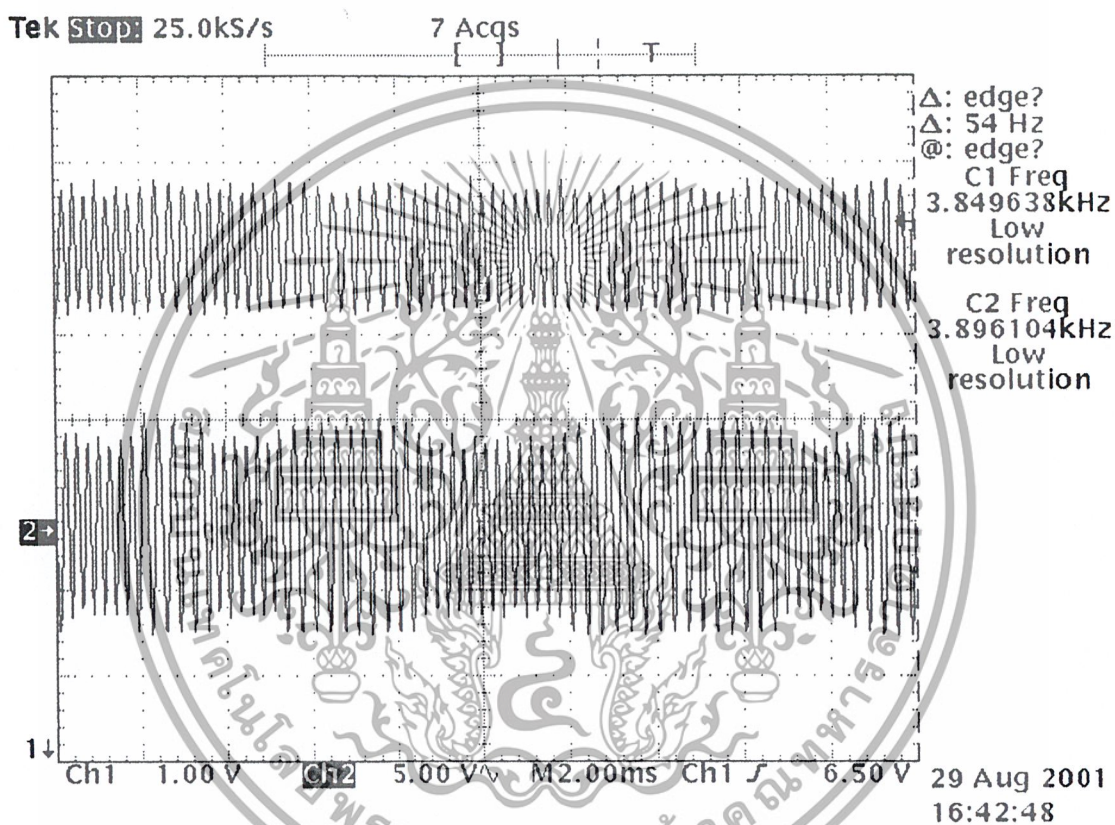
เป็นการส่งข้อมูลออกจากคอมพิวเตอร์ซึ่งเป็นสัญญาณดิจิทัลและผ่านวงจรปรับระดับแรงดันก่อนส่งเข้าวงจร เฟสแอสเคมีคอลลูเลทเตอร์ เพื่อใช้เป็นข้อมูลนาฬิกาในการส่งออกอากาศ



รูปที่ 4.1 เปรียบเทียบสัญญาณดิจิทัลที่ต้องการส่ง กับสัญญาณที่ผ่านวงจร FSK Modulator ด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สัญญาณที่ออกจากวงจร FSK Modulator กับสัญญาณที่ผ่านวงจรขยายแล้ว ด้านส่งเป็นการนำเอาสัญญาณของวงจรเฟสแอสเคมีออคคูเลทเตอร์ มาทำการขยายก่อนส่งไปมอดคูเลทกับความถี่พาหะ

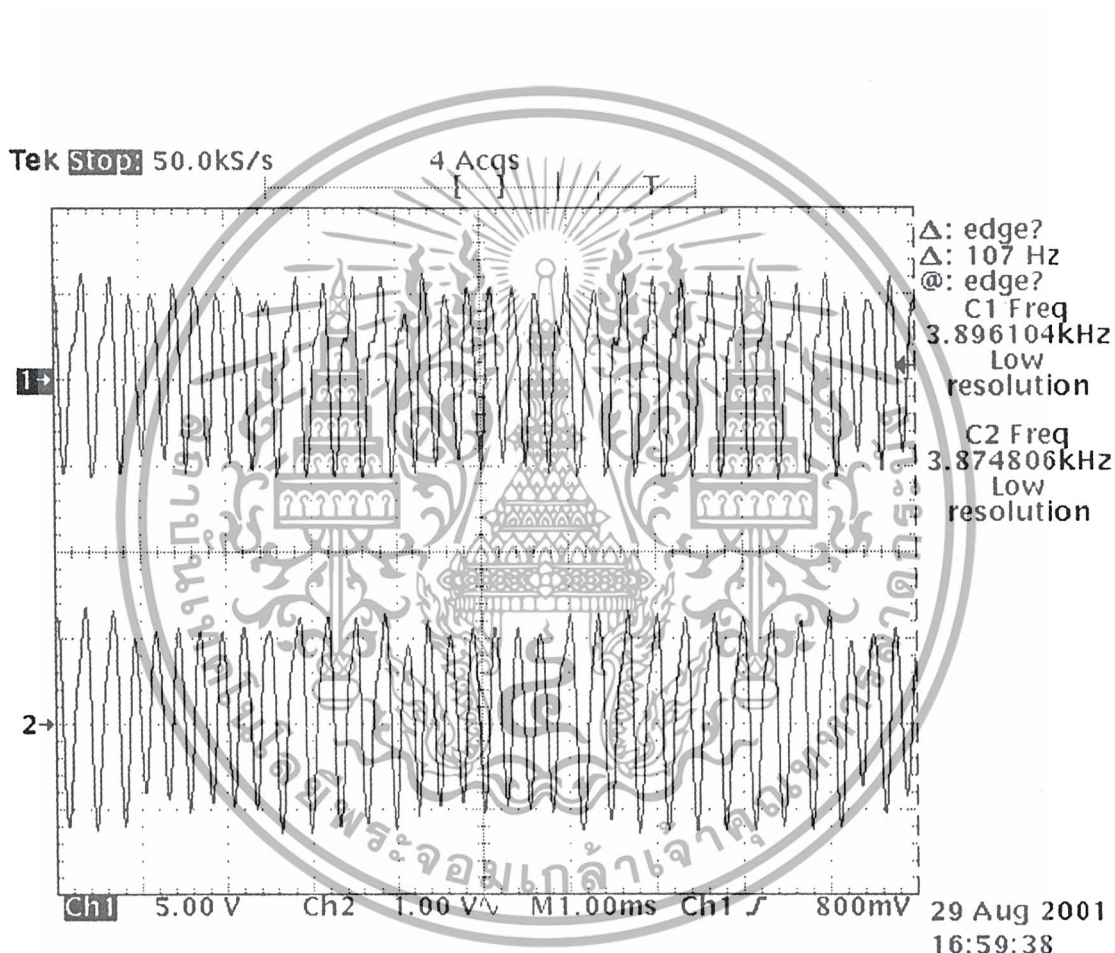


รูปที่ 4.2 เปรียบเทียบสัญญาณจากวงจร FSK Modulator กับสัญญาณที่ผ่านวงจรขยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 สัญญาณที่ภาครับตีเทคได้ กับ สัญญาณที่ผ่านวงจรกรองความถี่ต่ำผ่าน

สัญญาณที่ตีเทคได้ทางด้านรับยังมีรูปคลื่นสัญญาณไม่คืนันเนื่องจากมีสัญญาณรบกวน จึงใช้วงจรกรองความถี่ต่ำผ่านกรองเอาเฉพาะย่านความถี่ที่ต้องการ เพื่อตัดสัญญาณความถี่สูงที่ไม่ต้องการออกไป



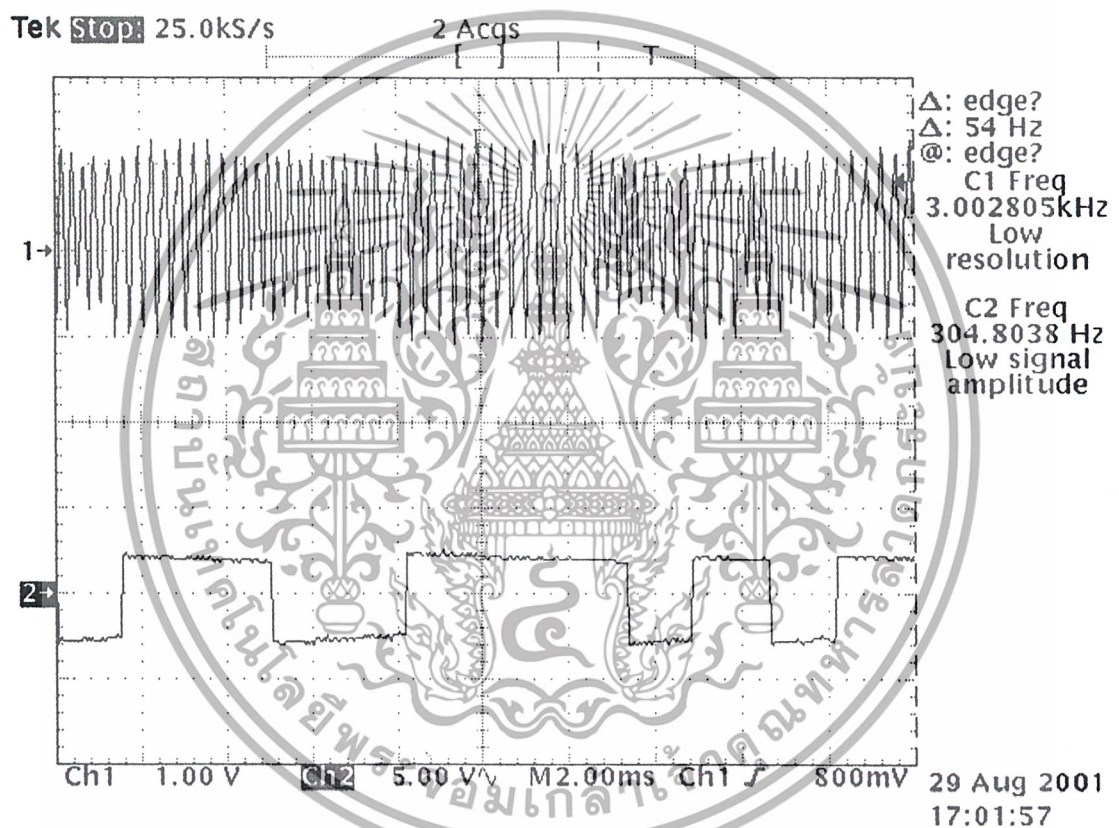
รูปที่ 4.3 เปรียบเทียบสัญญาณที่ตีเทคได้กับ สัญญาณที่ผ่านวงจรกรองความถี่ต่ำผ่านทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 สัญญาณที่ผ่านวงจรกรองความถี่ต่ำผ่าน กับสัญญาณที่เอาท์พุทของวงจร FSK

Demodulator ที่ด้านรับ

เพื่อให้ได้ข้อมูลที่เรากำลังต้องการจึงนำสัญญาณที่ผ่านวงจรกรองความถี่ต่ำผ่านแล้ว ป้อนเข้าวงจรเฟสแอสเคียมอสตูลเตเตอร์ เพื่อถอดรหัสให้ได้เป็นสัญญาณดิจิทัล

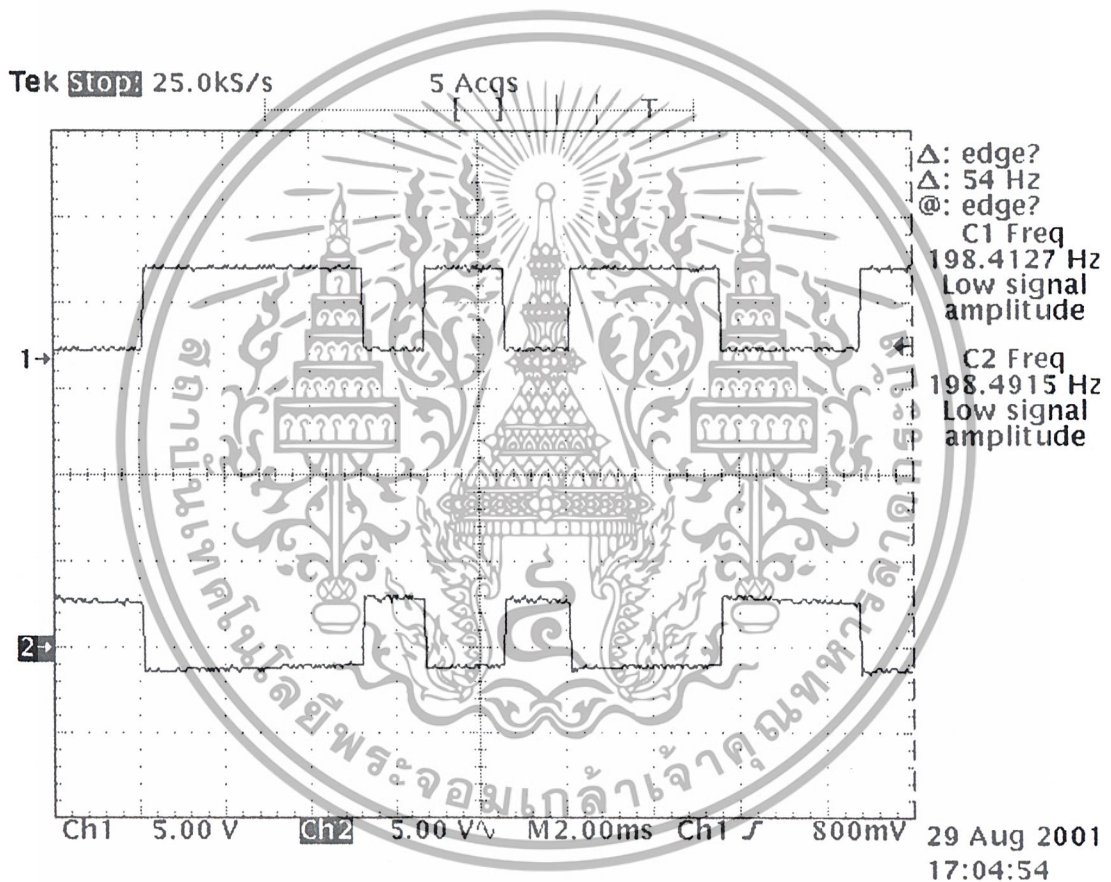


รูปที่ 4.4 เปรียบเทียบสัญญาณที่ผ่านวงจรกรองความถี่ต่ำผ่าน กับสัญญาณที่ผ่านวงจรเฟสแอสเคียมอสตูลเตเตอร์ ทางด้านรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 สัญญาณจากวงจรเอฟเอสเคเอ็มออคูลูเทเตอร์กับ สัญญาณเอาต์พุตอินเวอร์เตอร์

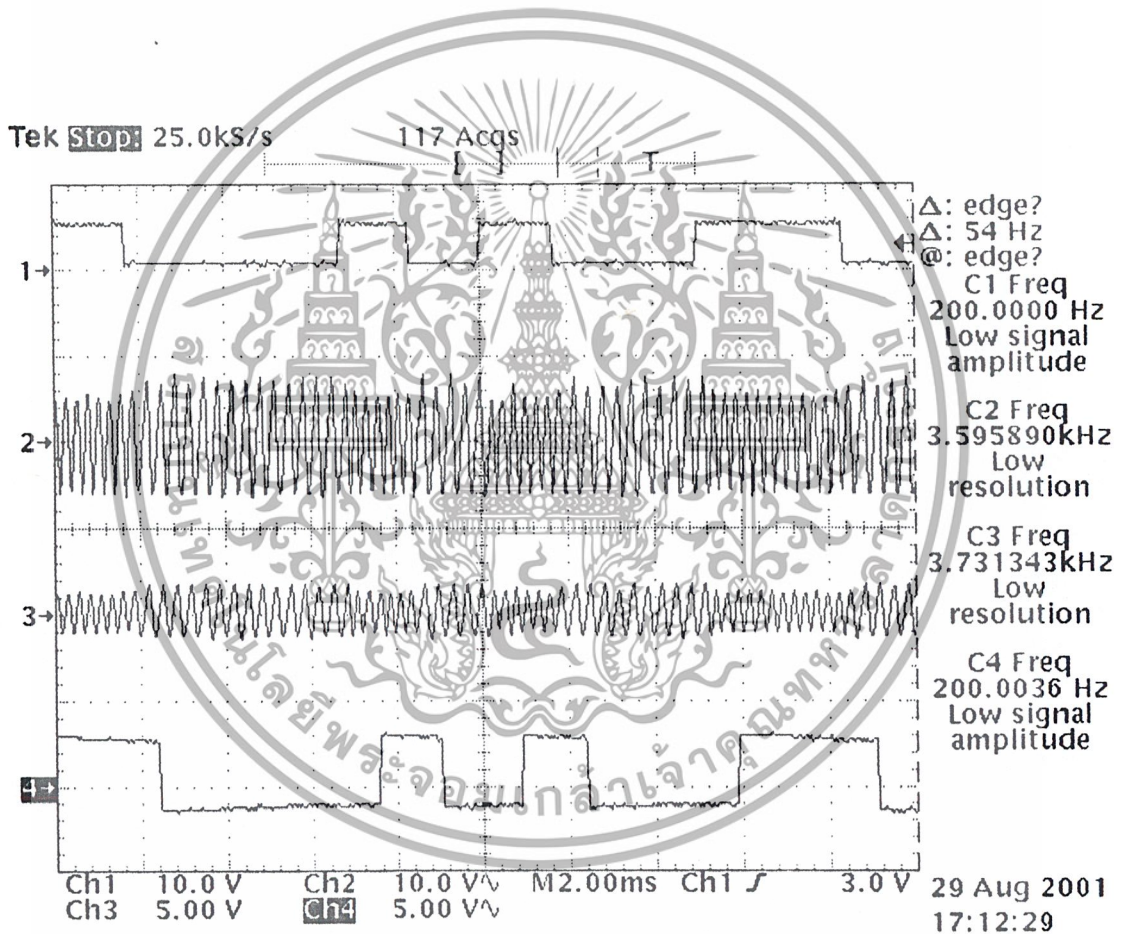
เนื่องจากในการทดลองผลที่ได้จากวงจร FSK Demodulator สัญญาณจะกลับเฟสตรงข้ามกันกับสัญญาณที่ขา TXD ของไมโครคอนโทรลเลอร์ทางด้านส่งข้อมูล ดังนั้นเพื่อให้ได้สัญญาณกลับคืนมาเหมือนเดิมจึงได้เพิ่มวงจรอินเวอร์เตอร์เข้ามาก่อนที่จะส่งสัญญาณเข้าสู่วงจรไมโครคอนโทรลเลอร์ทางด้านรับข้อมูลต่อไป โดยสัญญาณที่ได้แสดงดังรูปที่ 4.5



รูปที่ 4.5 เปรียบเทียบสัญญาณจากวงจรเอฟเอสเคเอ็มออคูลูเทเตอร์ กับสัญญาณที่ทำการกลับเฟส 180 องศา แล้ว

4.7 สัญญาณอินพุทที่เครื่องส่ง กับสัญญาณเอาต์พุทที่เครื่องรับ

จากรูปเมื่อนำสัญญาณทั้งหมดมาเปรียบเทียบกันตั้งแต่ป้อนสัญญาณเข้าอินพุทที่เป็นสัญญาณดิจิทัลผ่านวงจร เฟสเอสเคมอคดูเลทเตอร์แล้วส่งออกอากาศที่ทางด้านส่ง และทางด้านรับก็จะรับสัญญาณจากเสาอากาศที่เทคก่อนส่งเข้าวงจรเฟสเอสเคมอคดูเลทเตอร์และทำการกลับเฟสเพื่อให้ได้สัญญาณเหมือนที่ส่งมา



รูปที่ 4.6 สัญญาณเอาต์พุทที่เครื่องส่ง กับสัญญาณเอาต์พุทที่เครื่องรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.8 ตัวอย่างการออกแบบวงจรที่ใช้ในการทดลอง

วงจร FSK Modulator

ทำการกำหนดค่าคาปาซิเตอร์จับเวลา (Timing Capacitor) ซึ่งควรอยู่ในช่วง 1 nF ถึง 100 uF ดังนั้นในการทดลองได้กำหนดค่าคาปาซิเตอร์จับเวลาไว้เท่ากับ 0.022 uF กำหนดค่าความต้านทานจับเวลา (Timing Resistor) ได้ดังนี้

- ค่าความต้านทานจับเวลาที่ขา 7 ซึ่งใช้กับสัญญาณดิจิทัลที่มีลิจิก “1” โดยความถี่ที่กำหนดมีค่าเท่ากับ 3700 เฮิรท์ ดังนั้นจะได้ค่าความต้านทานที่ขา 7

$$= 1 / (0.022 * 10^{-6} * 3700)$$

$$= 12,285 \text{ โอห์ม}$$

- ค่าความต้านทานจับเวลาที่ขา 8 ซึ่งใช้กับสัญญาณดิจิทัลที่มีลิจิก “0” โดยความถี่ที่กำหนดมีค่าเท่ากับ 2700 เฮิรท์ ดังนั้นจะได้ค่าความต้านทานที่ขา 8

$$= 1 / (0.022 * 10^{-6} * 2700)$$

$$= 16,835 \text{ โอห์ม}$$

วงจร FSK Demodulator

ขั้นตอนการกำหนดค่าความต้านทานและคาปาซิเตอร์

1. กำหนดค่าความถี่กลางของเพลตีสีกฎ: f_0 เมื่อด้านส่งได้กำหนดอัตราการส่งข้อมูลเท่ากับ 1200 บิต ซึ่งความถี่ f_1 และ f_2 มีค่าเท่ากับ 3,700 เฮิรท์ และ 2,700 เฮิรท์ตามลำดับ

$$f_0 = (f_1 + f_2) / 2$$

$$= (3,700 + 2,700) / 2$$

$$= 3,200 \text{ เฮิรท์}$$

2. เลือกค่าความต้านทาน R_0 ให้อยู่ในช่วง 10 กิโลโอห์ม ถึง 100 กิโลโอห์ม สำหรับค่าอ้างอิงของ R_0 ที่ใช้ในการทดลองเลือกค่าเท่ากับ 20 กิโลโอห์ม โดยอาจจะต่อเป็นค่าความต้านทานปรับค่าได้เพื่อปรับความละเอียด

3. กำหนดค่าของ C_0

$$C_0 = 1 / (f_0 * R_0)$$

$$= 1 / (3,200 * 20,000)$$

$$= 0.0156 \text{ uF}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คำนวณค่า R1

$$R1 = (R0 * f0) / (f1 - f2)$$

$$R1 = (20,000 * 3,200) / (3,700 - 2,700)$$

$$= 64 \text{ กิโลโอม์}$$

5. คำนวณค่า C1

$$C1 = C0 / 4$$

$$= 0.0156 * 10^{-6} / 4$$

$$= 0.0039 \text{ uF}$$

6. คำนวณค่าคาปาซิเตอร์ Cf สำหรับ Rf = 100 กิโลโอม์ ; Rb = 510 กิโลโอม์

$$Cf (\text{uF}) = 3 / (\text{Baud rate})$$

$$= 3 / 1200$$

$$= 0.0025 \text{ uF}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

ปริญญานิพนธ์ฉบับนี้เสนอผลงานเกี่ยวกับระบบการควบคุมยานพาหนะหุ่นยนต์โดยคอมพิวเตอร์โมเด็มไร้สาย

ผลจากการวิจัยโดยทำการออกแบบในส่วน Modem และในส่วนยานพาหนะหุ่นยนต์ เมื่อทดลองส่งข้อมูลเชื่อมต่อกันพบว่าสามารถทำงานได้อย่างมีประสิทธิภาพในระดับที่เป็นที่น่าพอใจ คือ หากข้อมูลถูกส่งมาอย่างไรยานพาหนะหุ่นยนต์ก็จะปฏิบัติงานตามโปรแกรมคำสั่งนั้น ถ้าหากคำสั่งนั้นถูกแต่ถ้า หากข้อมูลของคำสั่งเปลี่ยนไปยานพาหนะหุ่นยนต์จะไม่ทำงาน ซึ่งจากการทดลองในพื้นที่รัศมีที่ 15 เมตร พบว่าการทำงานของยานพาหนะหุ่นยนต์สามารถทำงานได้อย่างมีประสิทธิภาพ 90% แต่ถ้าหากกระยะทางยิ่งห่างไกลออกไปจะทำให้ประสิทธิภาพการทำงานลดลง สำหรับพื้นที่รัศมีการรับส่งของข้อมูลนั้นยังพบว่าขึ้นอยู่กับอุณหภูมิ ความชื้นของอากาศ และบริเวณพื้นที่ ที่ทำการติดต่อสื่อสารด้วยว่าเป็นที่โล่งหรือเปล่า

5.2 ปัญหาและแนวทางการแก้ไข

จากการทดลองการทำงานของระบบ ทำให้ทราบถึงปัญหาที่เกิดขึ้นและสามารถแก้ไขได้ดังนี้

1. ปัญหาจากสัญญาณรบกวนจากความถี่สูง ๆ สอดแทรกเข้ามากับสัญญาณข่าวสารด้วยแนวทางแก้ไขได้ใช้วงจรกรองย่านความถี่ของสัญญาณข่าวสารผลที่ได้รับคือ สามารถลดทอนการรบกวนจากสัญญาณรบกวนได้พอสมควร

2. ปัญหาจากระยะทางการรับส่งของข้อมูล จากวงจรที่ออกแบบไว้ยังมีกำลังส่งต่ำ ดังนั้นหากต้องการควบคุมการทำงานของยานพาหนะหุ่นยนต์ให้มีประสิทธิภาพควรเพิ่มกำลังส่งให้สูงขึ้น สำหรับแนวทางที่ได้ดำเนินไปคือ เพิ่มไฟเลี้ยงให้กับวงจร แต่สำหรับแนวทางแก้ไขให้ดีขึ้นนั้นจะต้องออกแบบย่านความถี่การรับส่งให้สูงขึ้น ทั้งนี้วงจรที่ได้ทำการวิจัยนั้นได้ใช้ความถี่ต่ำซึ่งไม่สามารถออกแบบให้มีกำลังส่งสูงได้เพราะจะต้องใช้สายอากาศที่ยาวมากๆ ซึ่งเป็นปัญหาเกี่ยวกับ SWR เมื่อออกแบบเพิ่มกำลังส่งให้สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ปัญหาทางด้านวงจรจ่ายกำลังในตอนแรกได้เลือกใช้แหล่งจ่ายพลังงาน คือ แบตเตอรี่ลูกเดียวซึ่งจะแบ่งจ่ายแรงดันไปให้กับส่วนควบคุม และส่วนขับเคลื่อน ผลการทำงานพบว่าเมื่อยานพาหนะหุ่นยนต์มีการเคลื่อนที่จะมีการดึงกระแสในตอน Start สูงซึ่งทำให้ยานพาหนะหุ่นยนต์เกิดการทำงานผิดพลาดขึ้น สำหรับแนวทางแก้ไขนั้นสามารถแก้ไขได้ 2 วิธี คือวิธีแรกเพิ่มแบตเตอรี่ให้จ่ายกระแสให้สูงขึ้น

วิธีที่ สอง คือเพิ่มแบตเตอรี่อีกลูก เพิ่มต้องการแยกแหล่งจ่ายไฟทั้งสองส่วนออกจากกันโดยเด็ดขาด สำหรับในการออกแบบการทำงานนั้น ได้ตัดสินใจเลือกใช้แบตเตอรี่สองลูกแทนการใช้แบตเตอรี่ลูกเดียวที่จ่ายกระแสได้สูง

5.3 การพัฒนาโครงการ

โครงการที่สร้างขึ้นนี้ สามารถทำงานได้ตามขีดความสามารถในวัตถุประสงค์ที่ได้กำหนดไว้ แต่สามารถเพิ่มขีดความสามารถในการทำงาน ได้อีกถ้ามีการพัฒนาเพิ่มเติมอย่างเช่น

1. สามารถที่จะพูดตอบโต้ได้
2. ทำให้เห็นตำแหน่งที่ยานพาหนะทำงานอยู่ได้
3. สามารถสั่งงานให้ทำงานในระยะไกลๆ ได้ตามที่ต้องการ
4. สามารถสั่งงานให้ส่งข้อมูลที่ต้องการในพื้นที่ที่ห่างไกลกลับมาได้

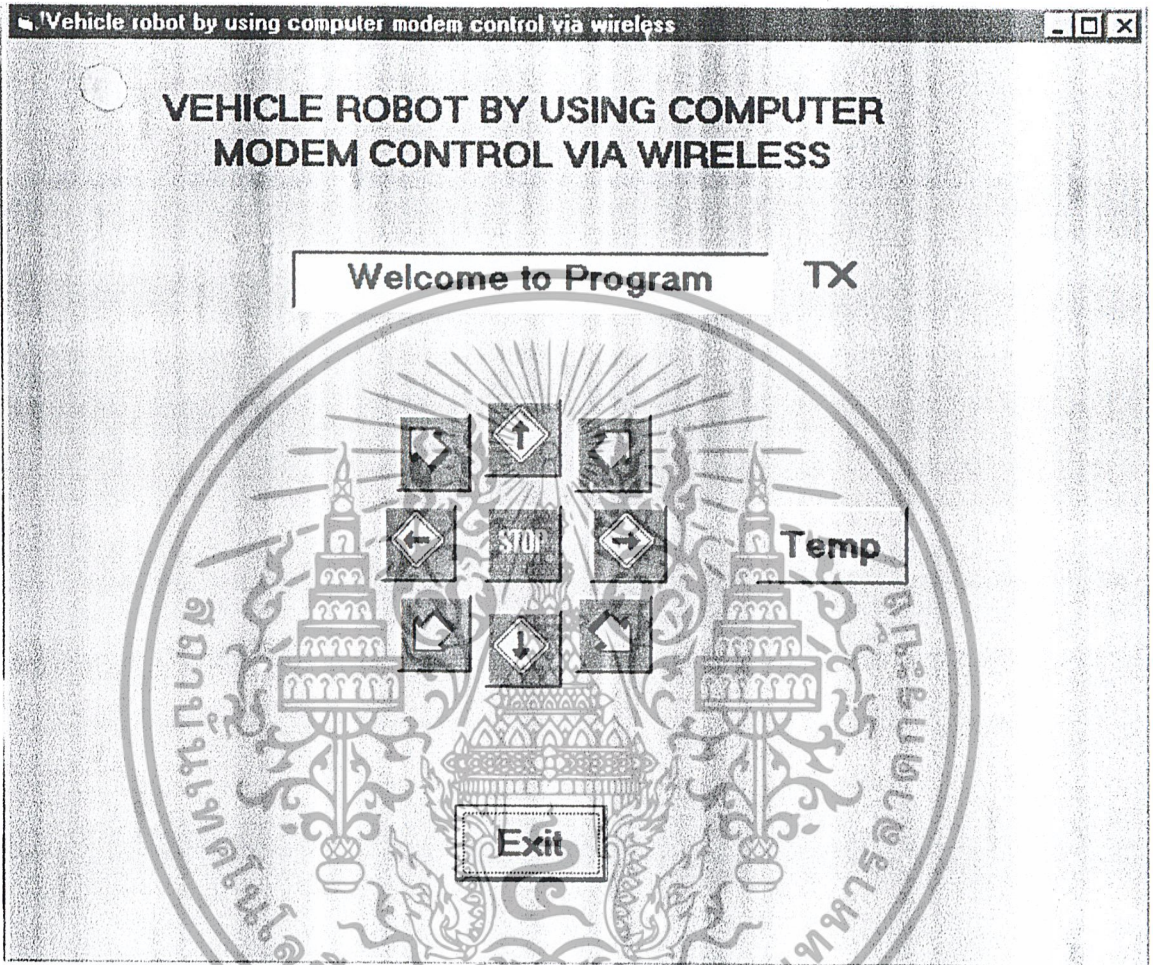
บรรณานุกรม

1. ชีรวัดน์ ประกอบผล. การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ฉบับปรับปรุง. พิมพ์ครั้งที่ 2 กรุงเทพฯ : แชนโพรพรินติ้ง , 2541
2. วิสาร กำจรเวทย์. Visual Basic ฉบับ Database. พิมพ์ครั้งที่ 2 กรุงเทพฯ : โปรวิชั่น, 2539
3. ไกรวุฒิ ไรจน์ประเสริฐสุด. เข้าใจ/สร้าง/เล่น ไมโครโปรเซสเซอร์ 2 .กรุงเทพฯ ฯ : ซีเอ็ดยูเคชั่น, 2539
4. ธนัญชัย จำนงค์ภักดิ์ , ธาริน สิริธรรมชารี. Microsoft Visual Basic Version 5 . กรุงเทพฯ : ชัค เวศ มีเดีย
5. กิตติ ภัคดีวัฒนะกุล, จำลอง กระจุดสาหะ . Visual Basic ฉบับ โปรแกรมเมอร์. กรุงเทพฯ : ไทย เจริญการพิมพ์, 2541
6. ชัยพัฒน์ สัมพรจิตรวิไล, วรพจน์ กรแก้ววัฒนกุล. เรียนรู้และปฏิบัติไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ฉบับ AT89C5x ของ Atmel .กรุงเทพฯ : อิน โนเวตีฟ เอ็กเพอริเมนต์ , 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

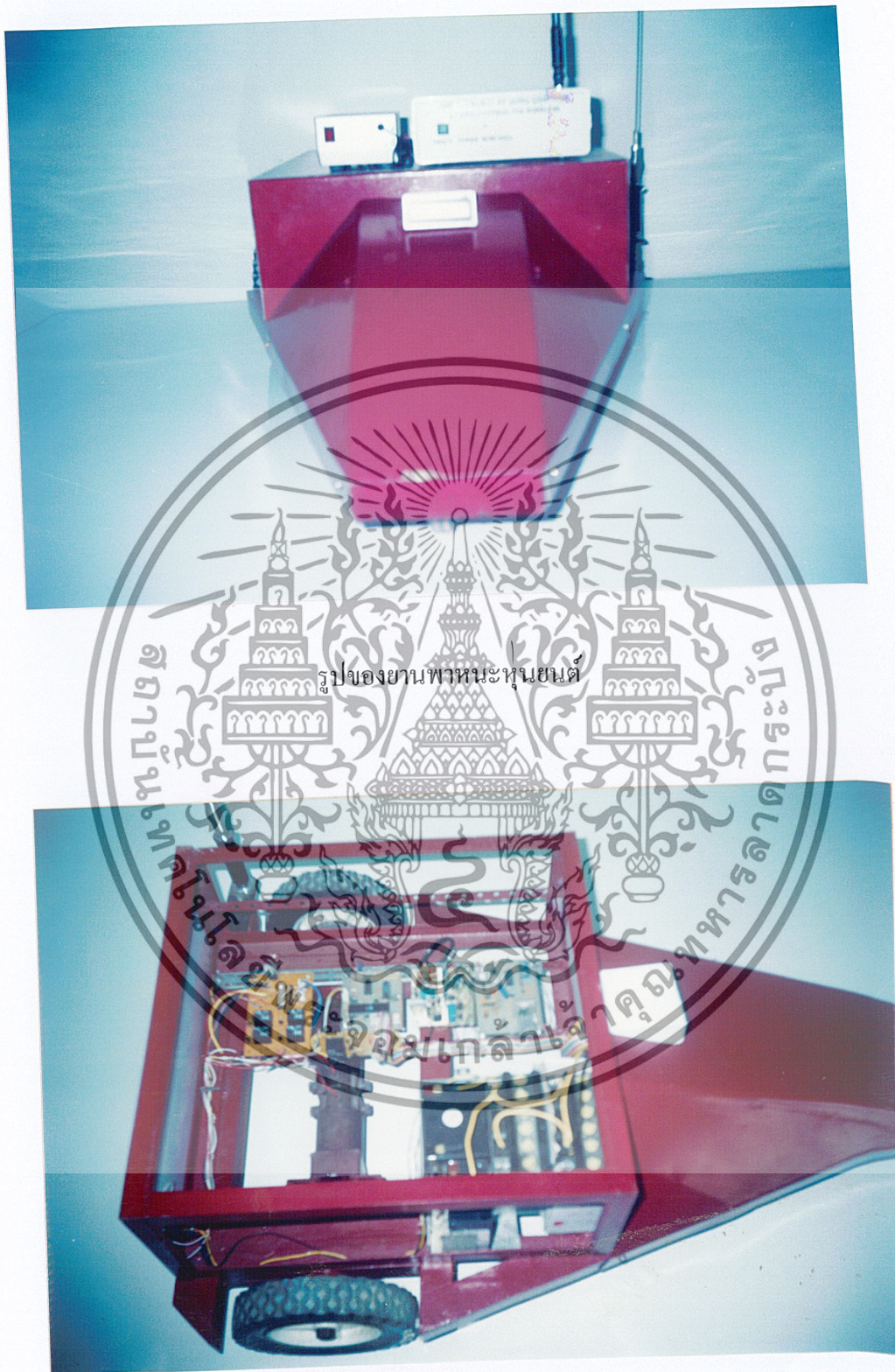


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปของหน้าจอแสดงผลการทำงานของโปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

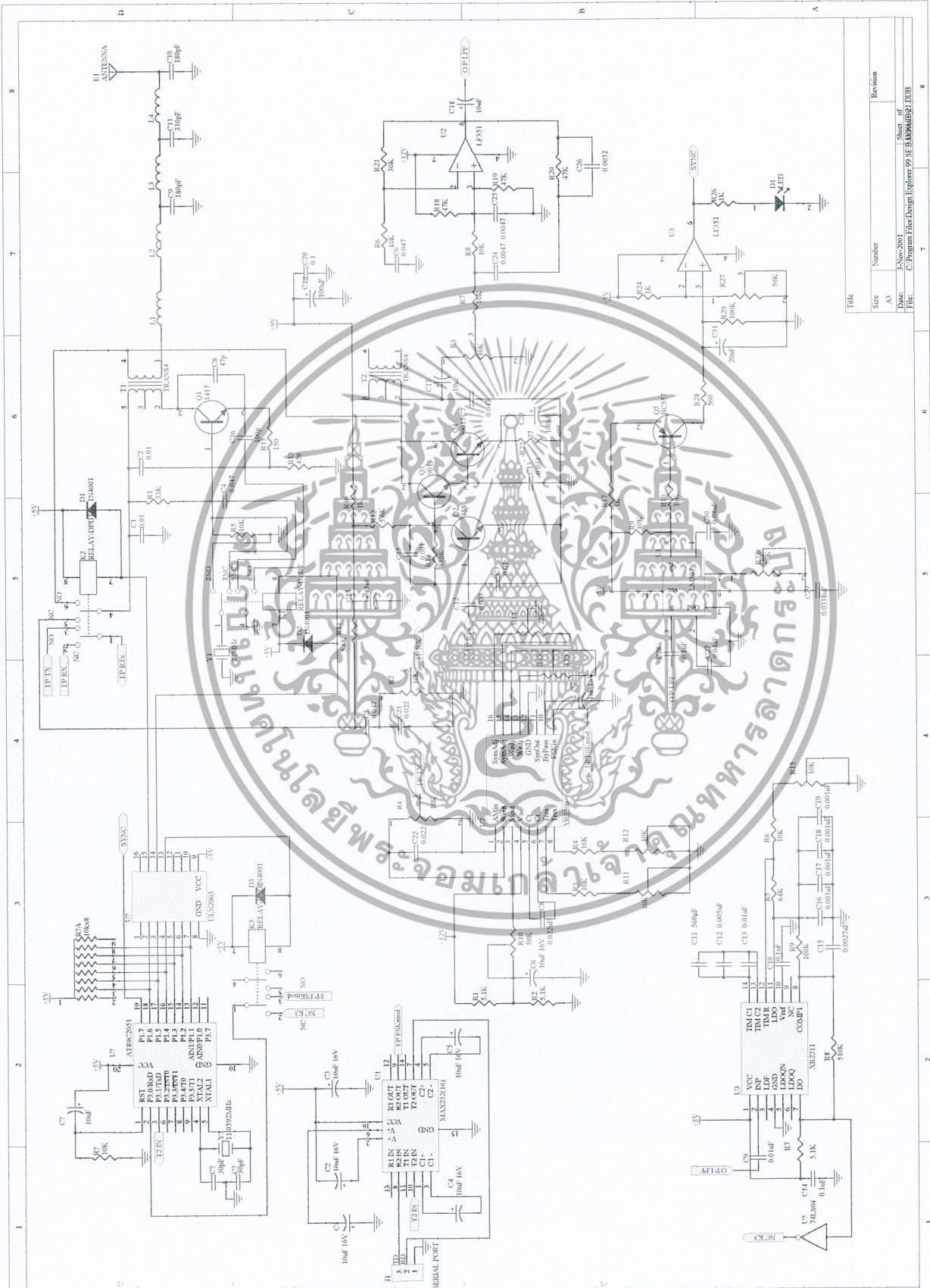


รูปตำแหน่งของการวางอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

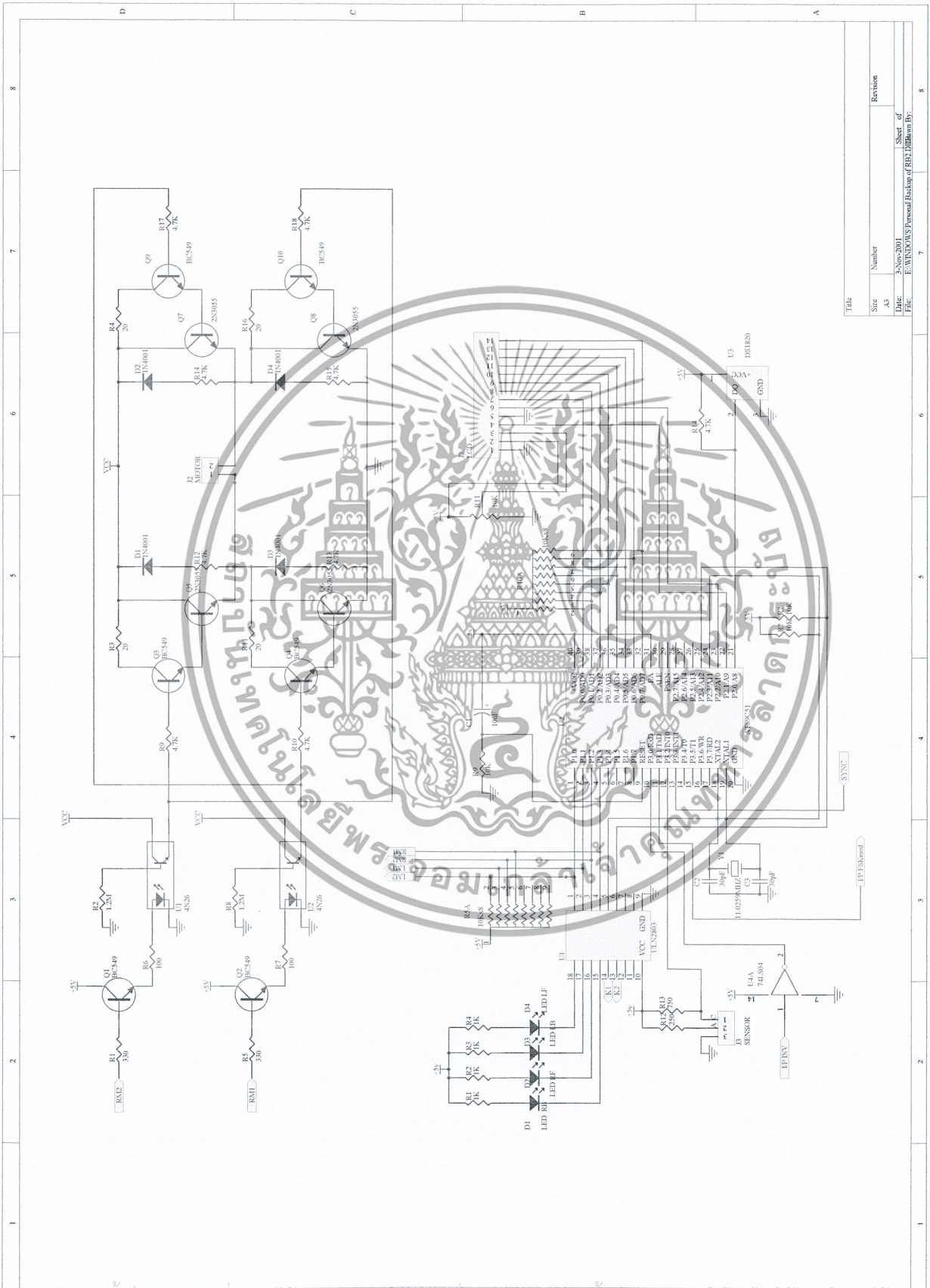


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



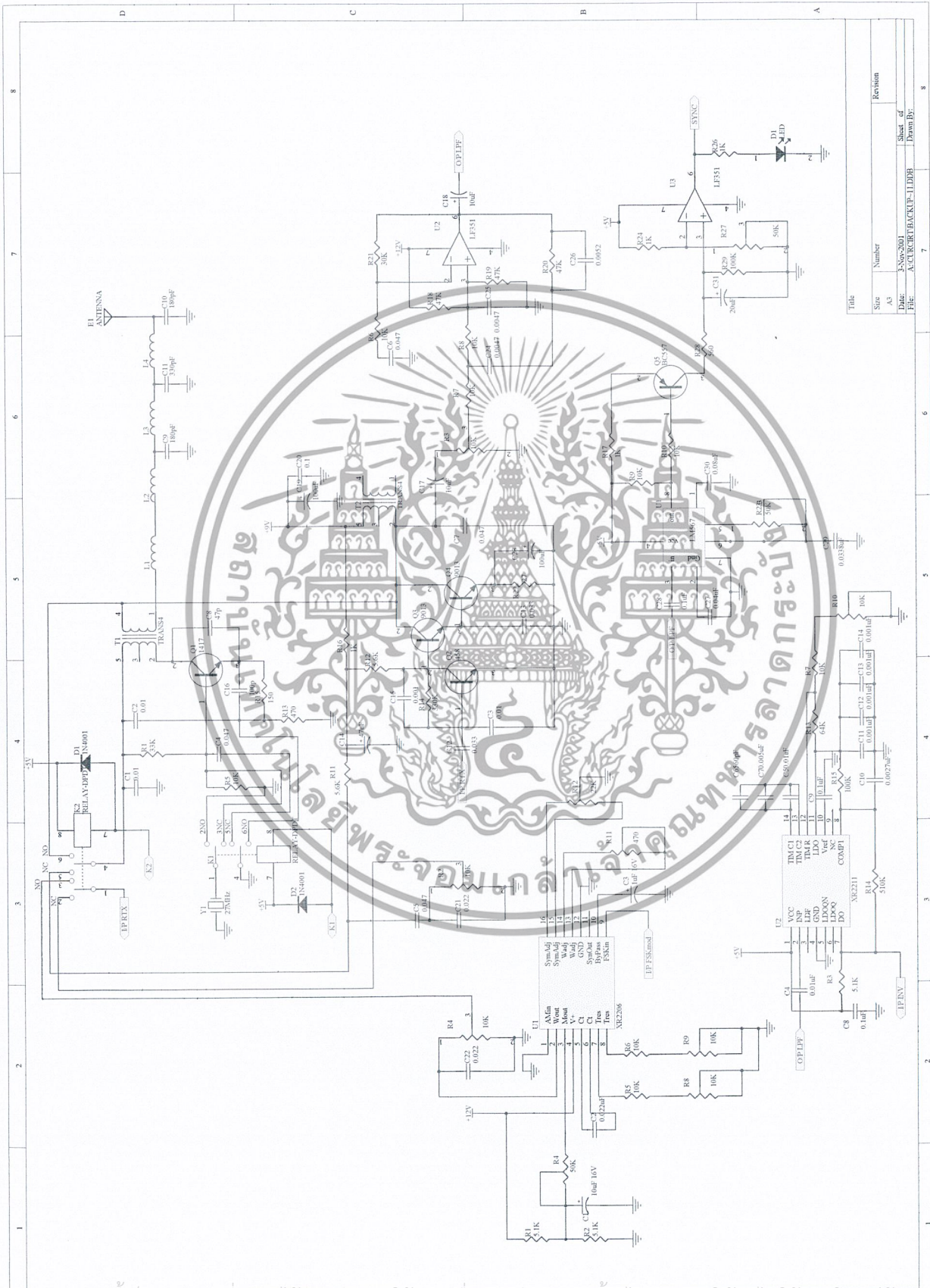
Title		Revision	
Size	A3	Number	
Date	3-Nov-2001	Sheet of	
File	C:\Program Files\Design Explorer\99 SE\B\MEMPH.DDB	File	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



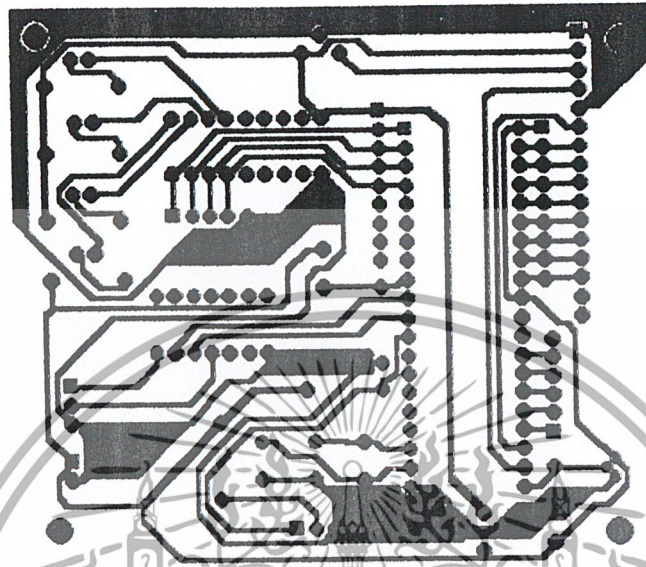
Title		
Size	Number	Revision
A3		
Date:	2-Nov-2001	Sheet of
File:	E:\WINDOWS\Faculty Backlog of (B3) Dikham BY	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

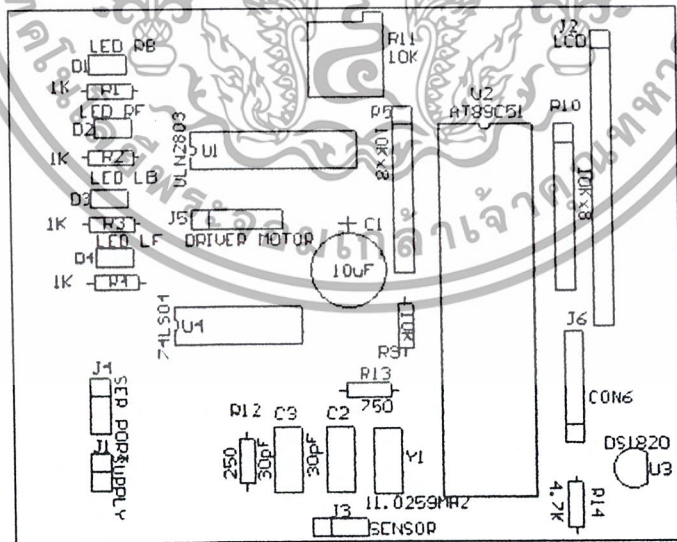


Title		Revision	
Size	Number	33	
Date	3 Nov 2003	Sheet of	8
File	ACCURBACKUP-11.DDB	Drawn By	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

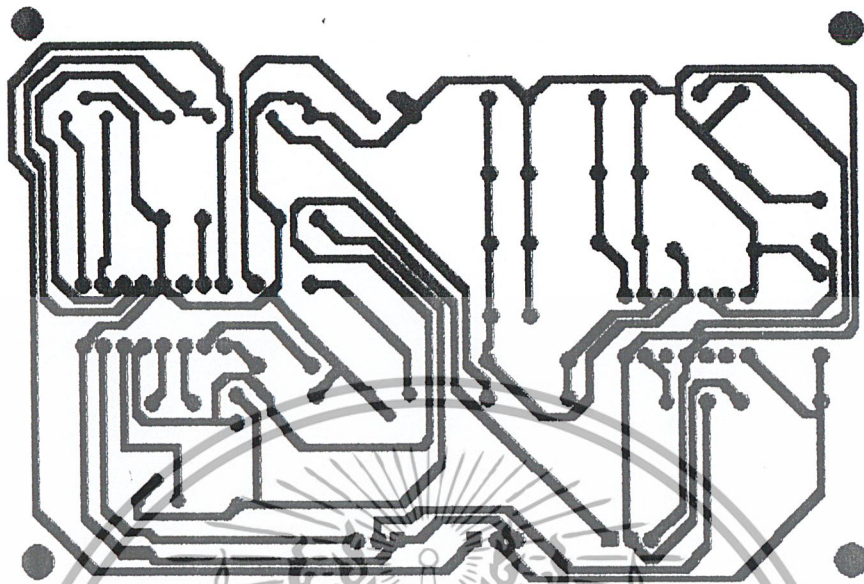


รูป ก.1 แสดงลายวงจรพิมพ์ของส่วนควบคุม

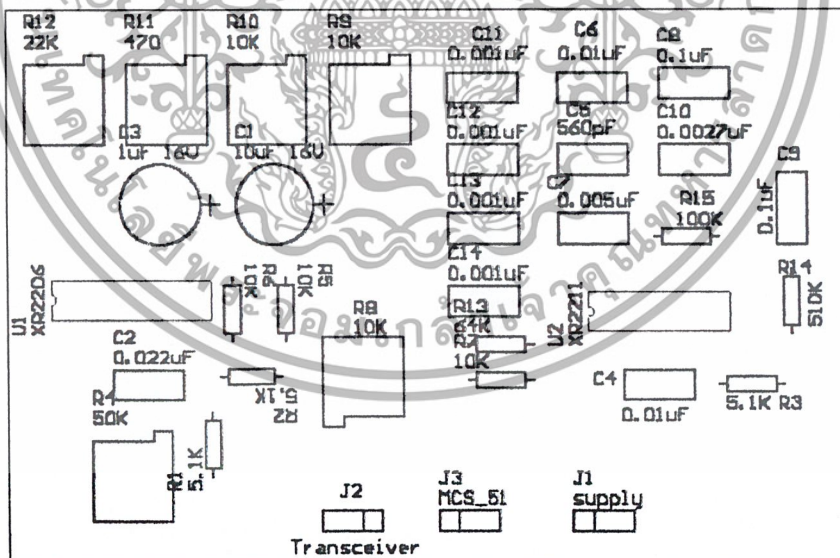


๓-๑๖ การวางอุปกรณ์บนลายวงจรส่วนควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

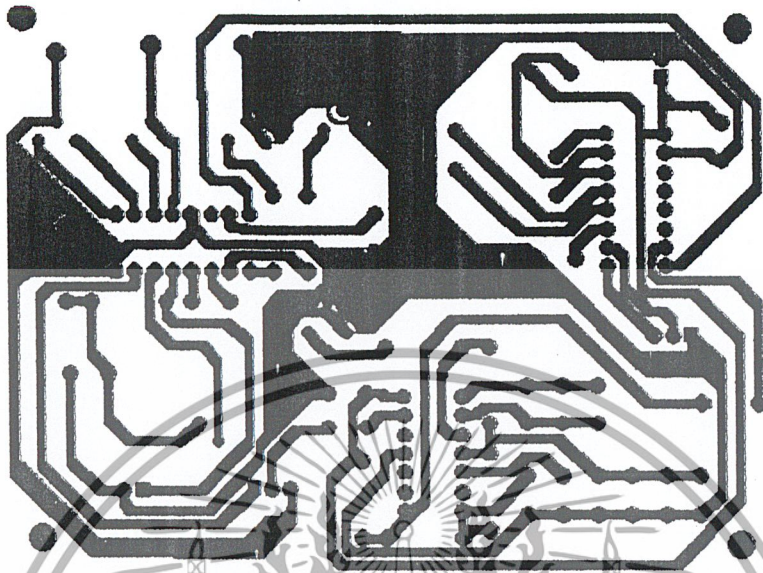


รูป ก.3 แสดงลายวงจรพิมพ์ของส่วนเข้ารหัสสัญญาณ

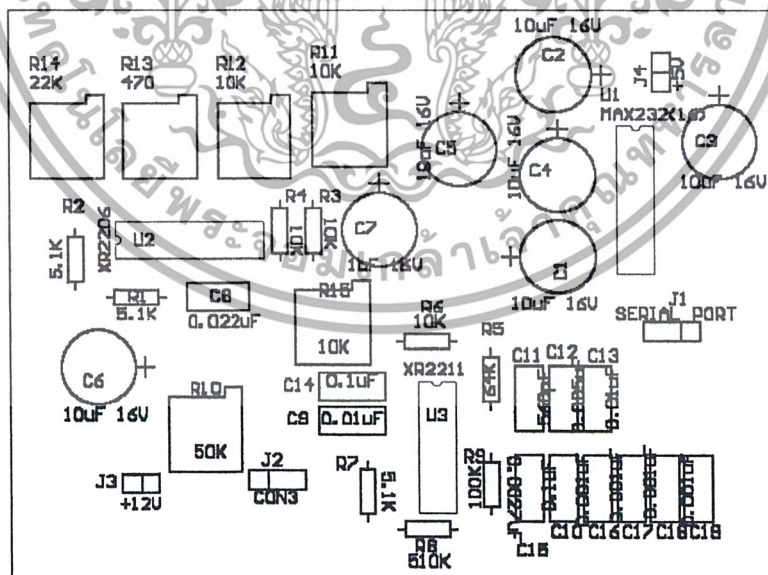


รูป ก.4 การวางอุปกรณ์บนลายวงจรส่วนเข้ารหัสสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

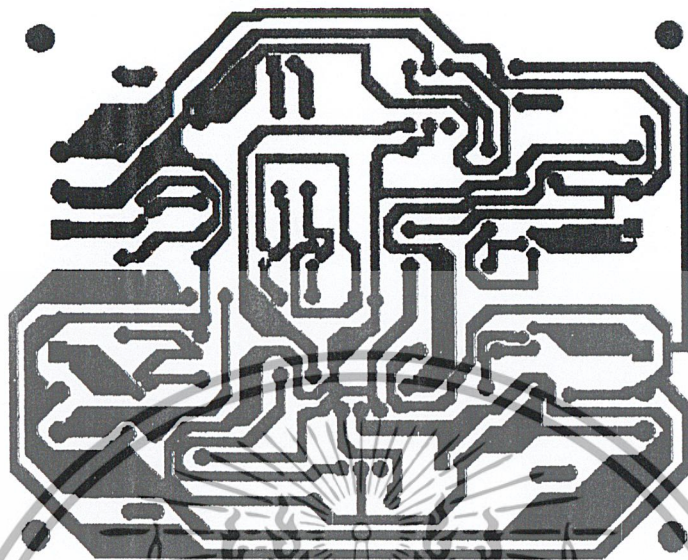


รูป ก.5 แสดงลายวงจรพิมพ์ของส่วนถอดรหัสสัญญาณ

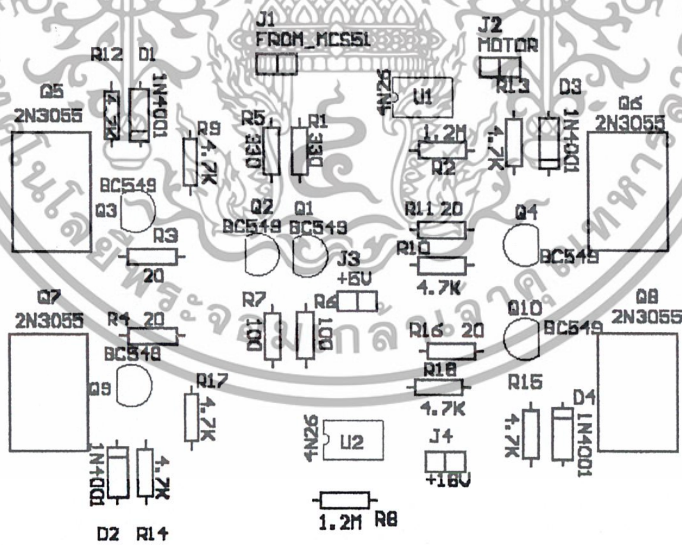


รูป ก.6 การวางผังกรณึบนลายวงจรของส่วนถอดรหัสสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

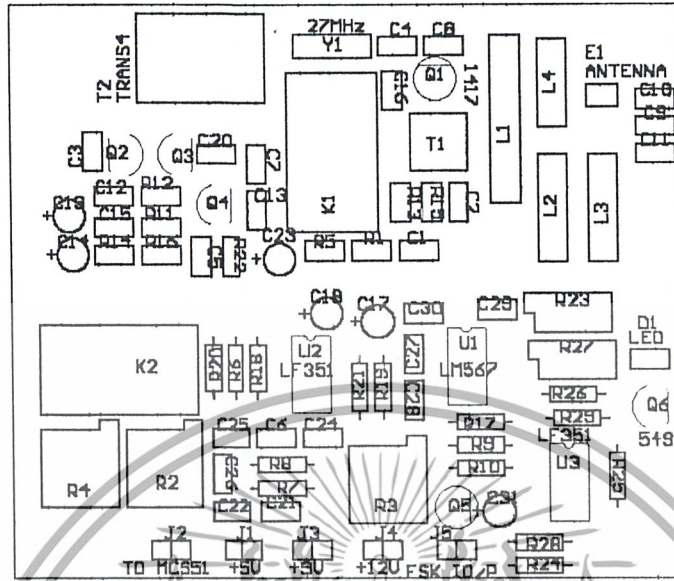


รูป ก.7 แสดงลายวงจรพิมพ์ของส่วนขับเคลื่อนมอเตอร์



รูป ก.8 การวางอุปกรณ์บนลายวงจรของส่วนขับเคลื่อนมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป ก.11 การวางอุปกรณ์บนสายวงจรของส่วนเครื่องรับและส่ง

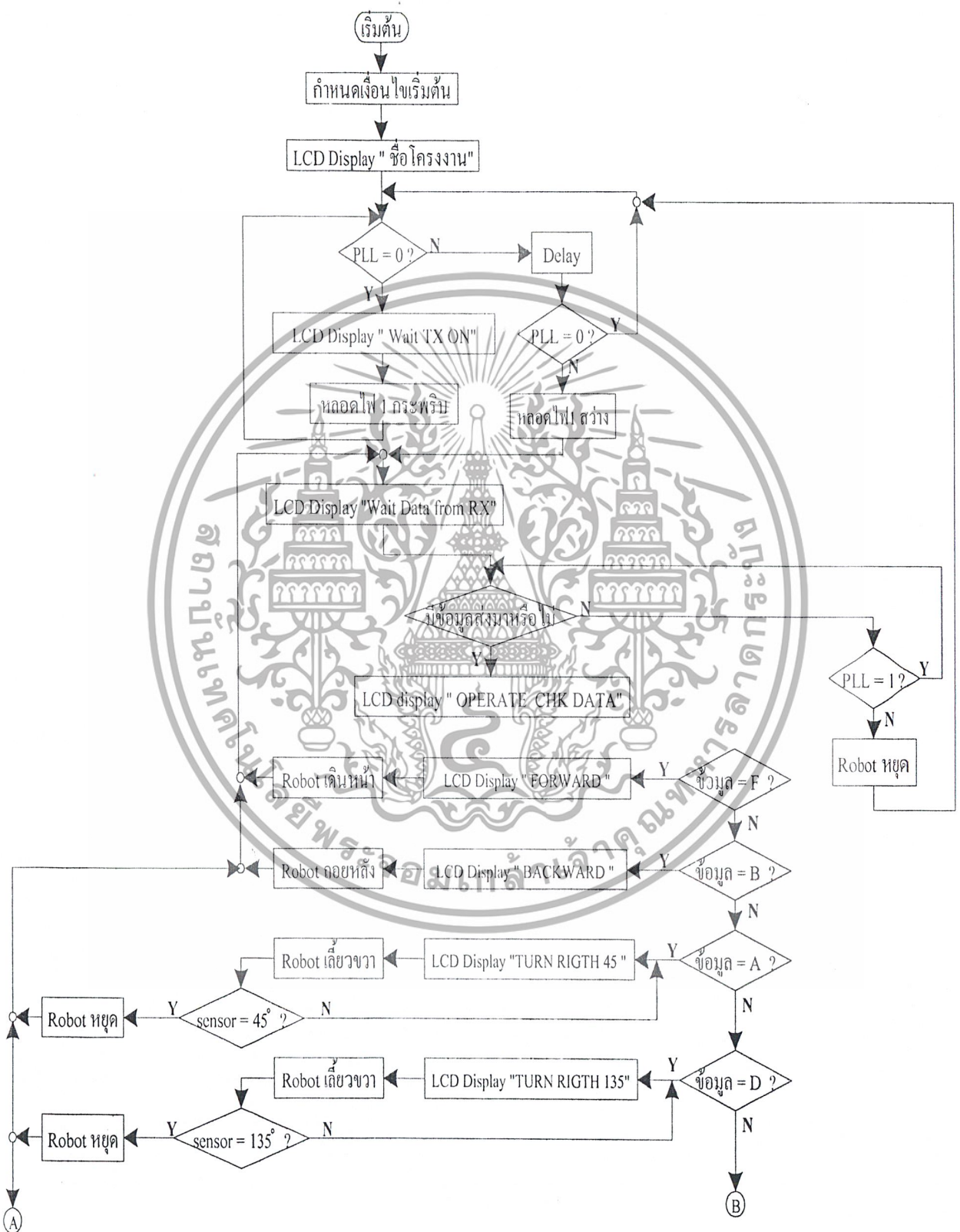


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

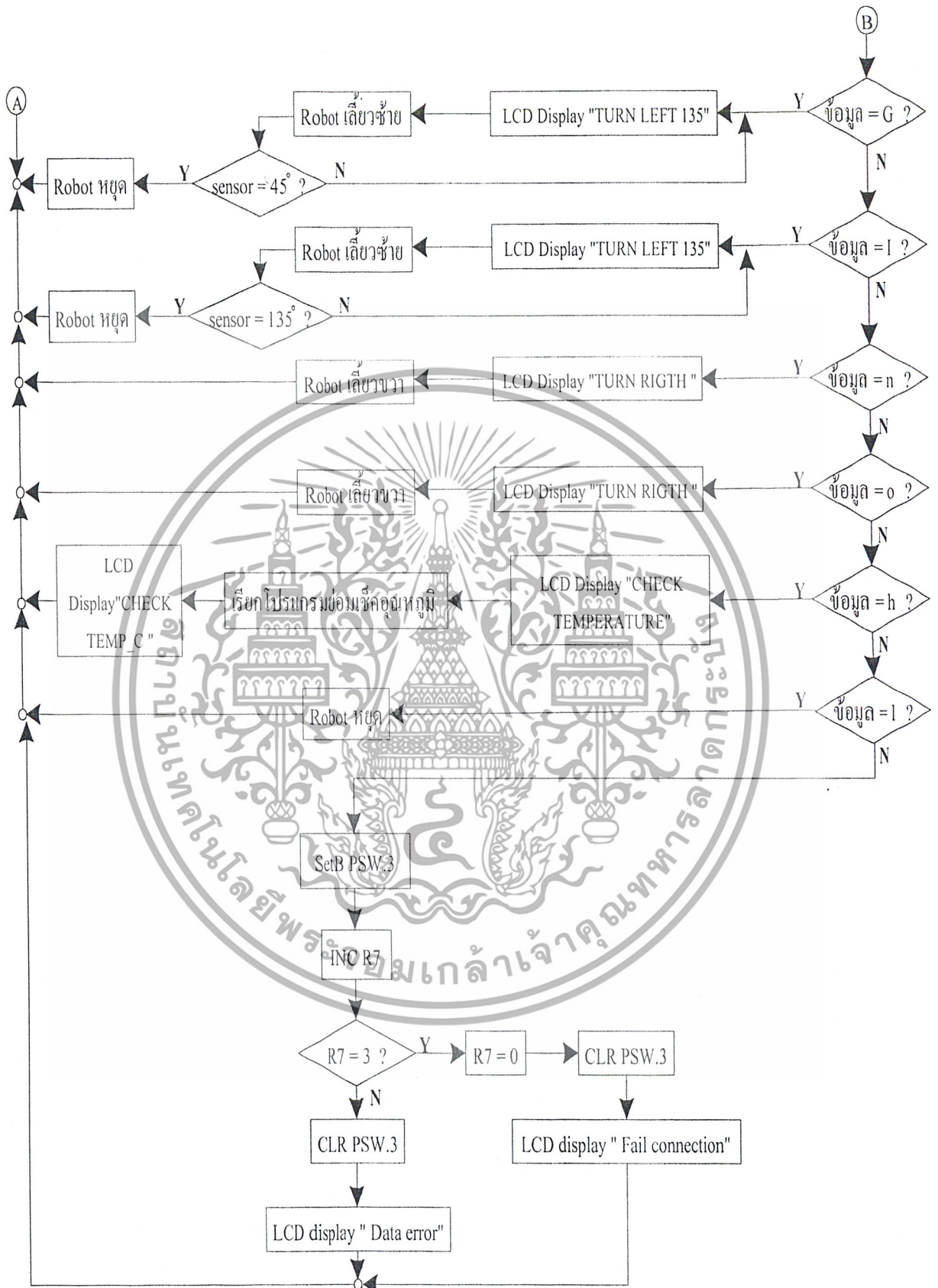


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลวชาร์ตโปรแกรมการทำงานของ Robot



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โฟลชาร์ตโปรแกรมย่อยเช็คอุณหภูมิ

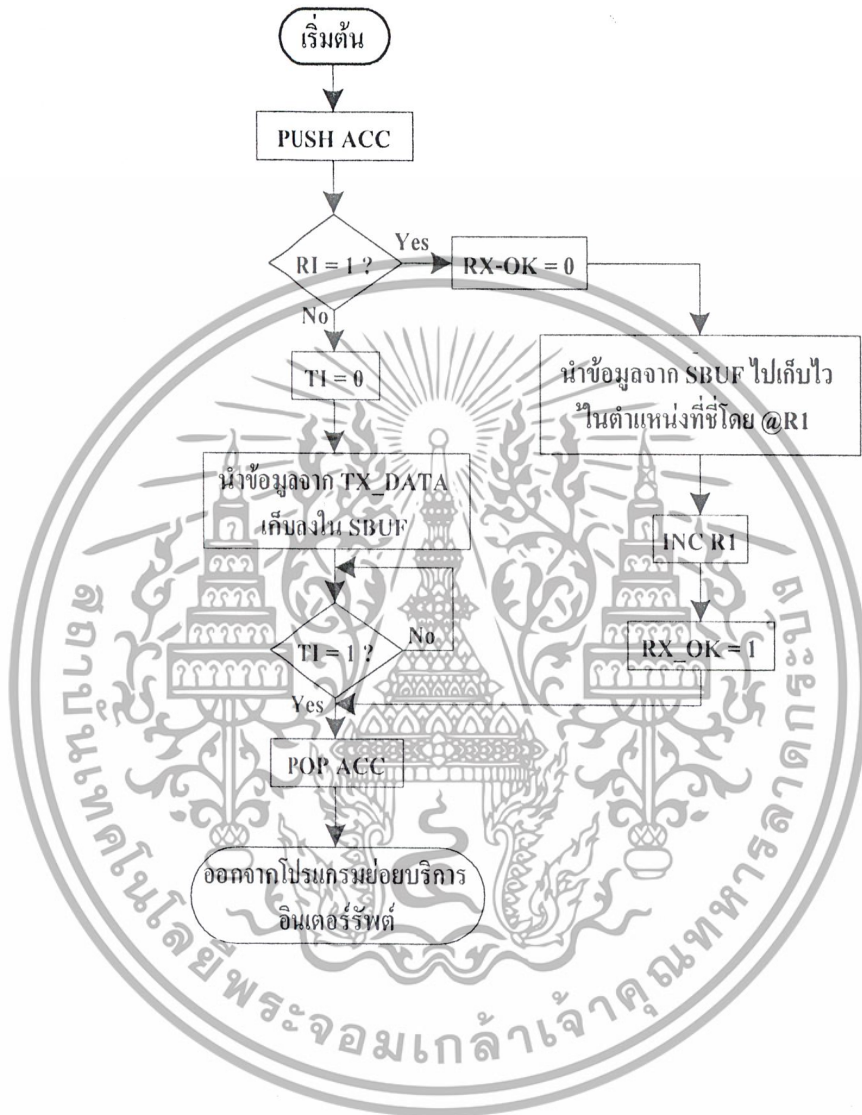


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





โปรแกรมย่อยการอินเตอร์รัพต์ การรับส่งข้อมูลผ่านพอร์ตอนุกรม



โปรแกรมควบคุมการทำงานของยานพาหนะหุ่นยนต์

```

;-----
; Program          : ROBOT WIRELESS
;-----

ONEWIRE           BIT           P2.0   ; Wire Interface DS1820
LCD_RS            BIT           P2.1   ; LCD Module Register Select
LCD_EN           BIT           P2.2   ; LCD Module Enable
RELAY1           BIT           P2.3
RELAY2           BIT           P2.4
WAIT_LAMP        BIT           P2.5
MAIN_SW_M        BIT           P2.6
WAIT_LAMP2       BIT           P2.7
INT_SENSOR       BIT           P3.2
PLL              BIT           P3.3
ENABLE_INT       BIT           IE.7
EN_SER_INT       BIT           IE.4
EN_INT0          BIT           IE.0
;-----

F_WARD           EQU           'F'
B_WARD           EQU           'B'
TR_45            EQU           'A'
TR_135           EQU           'D'
TL_45            EQU           'G'
TL_135           EQU           'I'
TR_SELF          EQU           'n'
TL_SELF          EQU           'o'
TEMP             EQU           'h'
STOP             EQU           'I'
STOP_TR45        EQU           040
    
```

STOP_TR135	EQU	080
STOP_TL45	EQU	040
STOP_TL135	EQU	080
FLAG	EQU	020H ; User FLAG
RX_OK	BIT	FLAG.0; Define BUSY as bit
BUSY	BIT	FLAG.1; Define BUSY as bit
SET_TEMP	BIT	FLAG.2 ; Initial condition set temp
WAIT_TX_OK	BIT	FLAG.3
FW_OK	BIT	FLAG.4
BW_OK	BIT	FLAG.5
TR_OK	BIT	FLAG.6
TL_OK	BIT	FLAG.7
FLAG1	EQU	021H
TEMP_OK	BIT	FLAG1.0
FAIL_OK	BIT	FLAG1.1
LCD_ADDR	EQU	022H ; For keep LCD Address
LCD_DATA	EQU	023H ; For keep LCD Data
ONEWIRE_DATA	EQU	024H ; For keep ONEWIRE Data
TEMP_DT	EQU	025H ; For keep Temp. Data (Temp L only)
TX_DATA	EQU	026H
BUFFER	EQU	029H ;Buffer for keep data from pc
STACK	EQU	0A0H

```

ORG      0000H
AJMP    Main_Program

ORG      0003H
AJMP    SENSOR_ON

ORG      0023H
AJMP    SERIAL_PORT_INT

```

----- MAIN PROGRAM -----

```

Main_Program:  ORG      0030H
                CLR      MAIN_SW_M
                MOV      P0,#0          ; Clear Databus
                CLR      LCD_EN        ; Clear LCD Enable
                CLR      LCD_RS        ; Clear LCD RS
                MOV      P1,#000H
                CLR      RX_OK
                CLR      WAIT_TX_OK
                CLR      WAIT_LAMP
                CLR      WAIT_LAMP2
                CLR      FW_OK
                CLR      BW_OK
                CLR      TR_OK
                CLR      TL_OK
                CLR      TEMP_OK
                CLR      FAIL_OK
                MOV      SP,#STACK
                SETB     SET_TEMP
                SETB     ONEWIRE
                SETB     ENABLE_INT    ;Enable all interrupt
                SETB     EN_SER_INT   ;Enable serial port interrupt
                MOV      TMOD,#020H   ;T1 8 BIT AUTO
                MOV      TH1,#0E8H    ; 1200 bps
                MOV      SCON,#040H   ; MODE 1 RX disable
                SETB     TR1           ; Start TIMER 1
                ACALL    DS1820       ; initial temp.

SET_TEMP_OK:   CLR      SET_TEMP
                SETB     PSW.3        ; Initial register bank 1
                MOV      R7,#0
                MOV      R6,#0

```



```

MOV        R5,#0
MOV        R4,#0
CLR        PSW.3          ; Return to register bank 0
ACALL     DELAY_100ms
ACALL     INIT_LCD       ; Call LCD Initial subroutine
ACALL     TITLE_SYSTEM
ACALL     DELAY_10ms     ; Wait system OK

;----- check tx ok ? -----
WAIT_PLL:  JNB        PLL,WAIT_PLL2      ; LAMP DISPLAY
           ACALL     DELAY_1s
           JNB        PLL,WAIT_PLL
           SETB     WAIT_LAMP
           SETB     WAIT_LAMP2
           CLR      WAIT_TX_OK
           AJMP    WAIT
WAIT_PLL2: JB        WAIT_TX_OK,PASS2
           ACALL     LCD_CLR
           MOV      DPTR,#WAIT_TX_ON1; LCD display "Wait Tx ON"
           ACALL     WR_LCD
           MOV      LCD_ADDR,#040H    ; Set address 40h
           ACALL     SET_ADDR_LCD
           MOV      DPTR,#WAIT_TX_ON2
           ACALL     WR_LCD
           SETB     WAIT_TX_OK
PASS2:     SETB     WAIT_LAMP
           SETB     WAIT_LAMP2
           ACALL     DELAY_1s
           CLR      WAIT_LAMP
           CLR      WAIT_LAMP2
           ACALL     DELAY_1s

```

```

                                AJMP      WAIT_PLL
;-----
WAIT:      JB          FW_OK,PASS3
           JB          BW_OK,PASS3
           JB          TR_OK,PASS3
           JB          TL_OK,PASS3
           JB          TEMP_OK,PASS3
           JB          FAIL_OK,PASS3
           ACALL      LCD_CLR
           ACALL      WAIT_DATA_RX; LCD display "WaitData from Rx"
PASS3:    MOV          R1,#BUFFER
           SETB       SCON,4
WAIT3:    JB          RX_OK,PASS1 ; Wait message received
           JB          PLL,WAIT3
           CLR        SCON,4
           CLR        MAIN_SW_M
           MOV        P1,#00H ; ROBOT STOP
           ACALL      LCD_CLR
           CLR        FW_OK
           CLR        BW_OK
           CLR        TR_OK
           CLR        TL_OK
           CLR        TEMP_OK
           CLR        FAIL_OK
           AJMP      WAIT_PLL

PASS1:    AJMP      ACKNOWLEDGE
;-----
SERIAL_PORT_INT:  PUSH      ACC
                JBC      RI,SER_RX ; RX or TX?
SER_TX:         CLR        TI ; SER_TX

```

```

MOV          SBUF, TX_DATA      ;Send data to PC
SER_TX_WAIT: JBC          TI,SER_EXIT
AJMP        SER_TX_WAIT; Wait until send data finished
SER_RX:     CLR          RX_OK    ; Clear RI
MOV         A,SBUF          ;Get data to ACC
MOV         @R1,A
INC         R1
SETB       RX_OK          ;Set RX_OK received data OK
SER_EXIT:   POP          ACC
            RETI
;-----
ACKNOWLEDGE: ACALL       LCD_CLR
            ACALL       OP_CHK_DATA
            AJMP        COMPARE_MESSAGE
;-----
COMPARE_MESSAGE:SETB     EN_INT0
            SETB       ITO
            MOV         R1,#BUFFER
COMPARE:    CJNE        @R1,#F_WARD,CHECK_BW
            ACALL       LCD_CLR
            MOV         DPTR,#OP_FW1; LCD Show "ROBOT FORWARD"
            ACALL       WR_LCD
            MOV         LCD_ADDR,#040H
            ACALL       SET_ADDR_LCD
            MOV         DPTR,#OP_FW2
            ACALL       WR_LCD
            SETB       PSW.3
            MOV         R7,#0
            CLR         PSW.3
            ACALL       INT0_CONDITION

```

```

ACALL    FORWARD
SETB     FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_BW: CJNE     @R1,#B_WARD,CHECK_TR45
ACALL    LCD_CLR
MOV      DPTR,#OP_BW1;LCDshow "ROBOT BACKWARD"
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#OP_BW2
ACALL    WR_LCD
SETB     PSW.3
MOV      R7,#0
CLR      PSW.3
ACALL    INTO_CONDITION
ACALL    BACKWARD
SETB     BW_OK
CLR      FW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_TR45: CJNE     @R1,#TR_45,CHECK_TR135 ; Turn right 45 ?
ACALL    LCD_CLR

```

```

ACALL    INT0_CONDITION
CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_TL45:
CJNE     @R1,#TL_45,CHECK_TL135 ; Turn left 45 ?
ACALL    LCD_CLR
MOV      DPTR,#OP_TL451; LCD Show "ROBOT TL 45"
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#OP_TL452
ACALL    WR_LCD
SETB     PSW.3
MOV      R7,#0
CLR      PSW.3
ACALL    TURN_LEFT45
ACALL    INT0_CONDITION
CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_TL135:
CJNE     @R1,#TL_135,CHECK_TR_SELF; Turn left 135 ?
ACALL    LCD_CLR

```

```

MOV      DPTR,#OP_TR451; LCD Show "ROBOT TR 45"
ACALL   WR_LCD
MOV      LCD_ADDR,#040H
ACALL   SET_ADDR_LCD
MOV      DPTR,#OP_TR452
ACALL   WR_LCD
SETB    PSW.3
MOV      R7,#0
CLR     PSW.3
ACALL   TURN_RIGHT45
ACALL   INTO_CONDITION
CLR     FW_OK
CLR     BW_OK
CLR     TR_OK
CLR     TL_OK
CLR     TEMP_OK
CLR     FAIL_OK
AJMP    WAIT_RX
CHECK_TR135: CJNE @R1,#TR_135,CHECK_TL45 ; Turn right 135 ?
ACALL   LCD_CLR
MOV      DPTR,#OP_TR1351; LCD Show "ROBOT TR 135"
ACALL   WR_LCD
MOV      LCD_ADDR,#040H
ACALL   SET_ADDR_LCD
MOV      DPTR,#OP_TR1352
ACALL   WR_LCD
SETB    PSW.3
MOV      R7,#0
CLR     PSW.3
ACALL   TURN_RIGHT135

```

```

MOV          DPTR,#OP_TL1351; LCD Show "ROBOT TL 135"
ACALL       WR_LCD
MOV          LCD_ADDR,#040H
ACALL       SET_ADDR_LCD
MOV          DPTR,#OP_TL1352
ACALL       WR_LCD
SETB        PSW.3
MOV          R7,#0
CLR         PSW.3
ACALL       TURN_LEFT135
ACALL       INTO_CONDITION
CLR         FW_OK
CLR         BW_OK
CLR         TR_OK
CLR         TL_OK
CLR         TEMP_OK
CLR         FAIL_OK
AJMP        WAIT_RX
CHECK_TR_SELF: CJNE @R1,#TR_SELF,CHECK_TL_SELF ; = n ?
ACALL       LCD_CLR
MOV          DPTR,#OP_TR1; LCD Show "ROBOT TURNRIGHT"
ACALL       WR_LCD
MOV          LCD_ADDR,#040H
ACALL       SET_ADDR_LCD
MOV          DPTR,#OP_TR2
ACALL       WR_LCD
SETB        PSW.3
MOV          R7,#0
CLR         PSW.3
ACALL       TURN_RIGHT_SELF

```

```

ACALL    INTO_CONDITION
SETB     TR_OK
CLR      FW_OK
CLR      BW_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_TL_SELF: CJNE    @R1,#TL_SELF,CHECK_TEMP ; = 0 ?
ACALL    LCD_CLR
MOV      DPTR,#OP_TL1; LCD Show "ROBOT TURNLEFT"
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#OP_TL2
ACALL    WR_LCD
SETB     PSW.3
MOV      R7,#0
CLR      PSW.3
ACALL    TURN_LEFT_SELF
ACALL    INTO_CONDITION
SETB     TL_OK
CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_TEMP: CJNE    @R1,#TEMP,CHECK_STOP
ACALL    LCD_CLR

```

```

MOV      DPTR,#OP_TEMP1;LCDShow"CHK TEMPERATURE"
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#OP_TEMP2
ACALL    WR_LCD
SETB     PSW.3
MOV      R7,#0
CLR      PSW.3
ACALL    DS1820
SETB     TEMP_OK
CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      FAIL_OK
AJMP     WAIT_RX
CHECK_STOP: CJNE    @R1,#STOP,OTHER
ACALL    INTO_CONDITION
ACALL    LCD_CLR
MOV      DPTR,#OP_STOP1;LCD Show "ROBOT STOP"
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#OP_STOP2
ACALL    WR_LCD
SETB     PSW.3
MOV      R7,#0
CLR      PSW.3
ACALL    STOP_ROBOT

```

```

CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX

```

OTHER:

```

ACALL    INTO_CONDITION
SETB     PSW.3 ; Define used register bank 1
INC      R7
CINE     R7,#2,ERROR
MOV      R7,#0
CLR      PSW.3
ACALL    LCD_CLR
MOV      DPTR,#FAIL_CON1; LCD Display "FAIL CONNECTION"
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#FAIL_CON2
ACALL    WR_LCD
SETB     FAIL_OK
CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      MAIN_SW_M
AJMP     WAIT_RX
ERROR:   CLR      PSW.3
         ACALL    LCD_CLR

```

```

MOV      DPTR,#DATA_ERROR1      ; Data error
ACALL    WR_LCD
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#DATA_ERROR2
ACALL    WR_LCD
CLR      FW_OK
CLR      BW_OK
CLR      TR_OK
CLR      TL_OK
CLR      TEMP_OK
CLR      FAIL_OK
AJMP     WAIT_RX
;-----
WAIT_RX: CLR      RX_OK
        JB      PLL_OK1
        CLR     SCON.4
        CLR     FW_OK
        CLR     BW_OK
        CLR     TR_OK
        CLR     TL_OK
        CLR     TEMP_OK
        CLR     FAIL_OK
        MOV     P1,#00001010B
        ACALL   DELAY_1mS
        ACALL   DELAY_1ms
        CLR     MAIN_SW_M
        MOV     P1,#0
        AJMP    WAIT_PLL
OK1:    AJMP    WAIT

```

;

```
INT0_CONDITION: CLR EN_INT0
                 CLR ITO
                 RET
```

;

; OPERATE

;

```
FORWARD:        MOV P1,#00001010B
                 ACALL DELAY_1ms
                 ACALL DELAY_1ms
                 CLR MAIN_SW_M
                 ACALL DELAY_1ms
                 ACALL DELAY_1ms
                 MOV P1,#00000000B ; Drive motor left - right
                 ACALL DELAY_1ms
                 ACALL DELAY_1ms
                 SETB MAIN_SW_M
                 RET

BACKWARD:       MOV P1,#00001010B
                 ACALL DELAY_1ms
                 ACALL DELAY_1ms
                 CLR MAIN_SW_M
                 ACALL DELAY_1ms
                 ACALL DELAY_1ms
                 MOV P1,#00001111B ; Drive motor left - right
                 ACALL DELAY_1ms
                 ACALL DELAY_1ms
                 SETB MAIN_SW_M
                 RET
```

```
TURN_RIGHT45:  MOV P1,#00001010B
```

```

ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
ACALL    DELAY_1ms
ACALL    DELAY_1ms
MOV      R2,#000H          ; Clear counter
MOV      P1,#00001100B ; Drive motor left - right
ACALL    DELAY_1ms
ACALL    DELAY_1ms
SETB     MAIN_SW_M
CJNE     R2,#STOP_TR45,$   ; Check sensor
MOV      P1,#00001010B ; Stop motor
ACALL    DELAY_1ms
CLR      MAIN_SW_M
MOV      P1,#0
RET
TURN_RIGHT135: MOV      P1,#00001010B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
ACALL    DELAY_1ms
ACALL    DELAY_1ms
MOV      R2,#000H          ; Clear counter
MOV      P1,#00001100B ; Drive motor left - right
ACALL    DELAY_1ms
ACALL    DELAY_1ms
SETB     MAIN_SW_M
CJNE     R2,#STOP_TR135,$   ; Check sensor
MOV      P1,#00001010B ; Stop motor
ACALL    DELAY_1ms

```

```

CLR      MAIN_SW_M
MOV      P1,#0
RET

TURN_LEFT45:
MOV      P1,#00001010B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
ACALL    DELAY_1ms
ACALL    DELAY_1ms
MOV      R2,#000H      ; Clear counter
MOV      P1,#00000011B ; Drive motor left - right
ACALL    DELAY_1ms
ACALL    DELAY_1ms
SETB     MAIN_SW_M
CJNE     R2,#STOP_T145,$ ; Check sensor
MOV      P1,#00001010B ; Stop motor
ACALL    DELAY_1ms
CLR      MAIN_SW_M
MOV      P1,#0
RET

TURN_LEFT135:
MOV      P1,#00001010B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
ACALL    DELAY_1ms
ACALL    DELAY_1ms
MOV      R2,#000H      ; Clear counter
MOV      P1,#00000011B ; Drive motor left - right
ACALL    DELAY_1ms
ACALL    DELAY_1ms

```

```

SETB      MAIN_SW_M
CJNE     R2,#STOP_TL135,$    ; Check sensor
MOV      P1,#00001010B ; Stop motor
ACALL    DELAY_1ms
CLR      MAIN_SW_M
MOV      P1,#0
RET

TURN_RIGHT_SELF: MOV      P1,#00001010B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
ACALL    DELAY_1ms
ACALL    DELAY_1ms
MOV      P1,#00001100B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
SETB     MAIN_SW_M
RET

TURN_LEFT_SELF:  MOV      P1,#00001010B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
ACALL    DELAY_1ms
ACALL    DELAY_1ms
MOV      P1,#00000011B
ACALL    DELAY_1ms
ACALL    DELAY_1ms
SETB     MAIN_SW_M
RET

STOP_ROBOT:  MOV      P1,#00001010B

```

```

ACALL    DELAY_1ms
ACALL    DELAY_1ms
CLR      MAIN_SW_M
MOV      P1,#0
RET

;-----
; INTO When sensor on
;-----

SENSOR_ON:  PUSH    ACC
            INC     R2
            POP     ACC
            RETI

;----- Check Temperature by IC DS1820 -----
DS1820:    ACALL    DS1820_RST    ; DS1820 Reset
            ACALL    DS1820_PRESENCE ; DS1820 Presence
            MOV     ONEWIRE_DATA,#0CCH ; Write Skip ROM
            ACALL    DS1820_WR
            MOV     ONEWIRE_DATA,#044H ; Write Convert Command
            ACALL    DS1820_WR
            SETB    BUSY          ; Set bit BUSY

PRES_CHK_LOOP: ACALL    DS1820_RST    ; DS1820 Reset
              ACALL    DS1820_PRESENCE ; DS1820 Presence
              JB     BUSY,PRES_CHK_LOOP ; Wait for Busy
              NOP
              NOP
              NOP
              NOP
              ACALL    DS1820_RST    ; DS1820 Reset
              ACALL    DS1820_PRESENCE ; DS1820 Presence
              MOV     ONEWIRE_DATA,#0CCH ; Write Skip ROM

```

```

ACALL    DS1820_WR
MOV      ONEWIRE_DATA,#0BEH
ACALL    DS1820_WR
ACALL    DS1820_RD          ; Read DS1820
MOV      TEMP_DT,ONEWIRE_DATA
ACALL    DS1820_RST        ; DS1820 Reset
ACALL    DS1820_PRES      ; DS1820 Presence
MOV      LCD_ADDR,#040H   ; Set Address 40H
ACALL    SET_ADDR_LCD
MOV      A,TEMP_DT        ; Get Temp. Data
CLR      C                ; Clear Carry Flag
RRC      A
MOV      LCD_DATA,A
ACALL    HEX_LCD
JB       SET_TEMP,SET_TEMP_OK_1
MOV      LCD_ADDR,#043H   ; Set Address 44H
ACALL    SET_ADDR_LCD
MOV      LCD_DATA,#'.'    ; Write point on lcd
ACALL    WRCHAR_LCD
MOV      A,TEMP_DT        ; Get Temp. Data
JNB     ACC.0,WRITE_0C    ; Check LSB was set?
MOV      LCD_DATA,#'5'
AJMP    WRITE_NEXT       ; Jump to write LCD
WRITE_0C:
MOV      LCD_DATA,#'0'
WRITE_NEXT:
ACALL    WRCHAR_LCD
MOV      LCD_DATA,#' '
ACALL    WRCHAR_LCD
MOV      LCD_DATA,#'C'
ACALL    WRCHAR_LCD
MOV      LCD_DATA,#' '

```

```

                                ACALL    WRCHAR_LCD

SET_TEMP_OK_1:  RET

;-----
; HEX Code send to PC
;-----

HEX_LCD:        PUSH    ACC

                MOV     A,LCD_DATA    ; Get Data

                MOV     B,#100

                DIV    AB              ; Divide by 100

                ADD    A,#030H        ; Convert to ASCII

                CJNE   A,#030H,HEX2_LCD_NX ; Check x100 = 0 ?

                MOV    A,#' '         ; Write Space

HEX2_LCD_NX:    MOV     LCD_DATA,A

                JB     SET_TEMP,PASS_1

                ACALL  WRCHAR_LCD

PASS_1:         MOV     A,B

                MOV     B,#10

                DIV    AB              ; Divide by 10

                ADD    A,#030H        ; Convert to ASCII

                MOV    LCD_DATA,A

                JB     SET_TEMP,PASS_2

                ACALL  WRCHAR_LCD

PASS_2:         MOV     A,B

                ADD    A,#030H        ; Convert to ASCII

                MOV    LCD_DATA,A ;

                JB     SET_TEMP,PASS_3

                ACALL  WRCHAR_LCD

PASS_3:         POP     ACC

                RET

;-----

```

; DS1820 Data Read

```
-----  
DS1820_RD:      MOV      R4,#8          ; Set loop 8 times  
                CLR      A  
DS1820_RD_LOOP: CLR      ONEWIRE  
                NOP  
                NOP  
                SETB    ONEWIRE  
                NOP  
                NOP  
                NOP  
                NOP  
                MOV     C,ONEWIRE ; Get ONEWIRE to Carry Flag  
                ACALL  ONEWIRE_DELAY ; Delay 75 us  
                RRC     A  
                DJNZ   R4,DS1820_RD_LOOP ; Do until 8 times  
                MOV     ONEWIRE_DATA,A  
                RET
```

;

; DS1820 Data Write

;

```
DS1820_WR:      MOV      R4,#8          ; Set loop 8 times  
                MOV     A,ONEWIRE_DATA ; Get ONEWIRE_DATA  
DS1820_WR_LOOP: RRC      A            ; Rotate ACC. with Carry Flag  
                JNC     DS1820_WR_L ; Carry Flag was set ?  
                CLR     ONEWIRE  
                NOP  
                NOP  
                NOP  
                NOP
```

```

SETB      ONEWIRE      ; Set ONEWIRE
ACALL     ONEWIRE_DELAY ; Delay 75 us
AJMP      DS1820_WR_NX ; Jump to next write
DS1820_WR_L:
CLR       ONEWIRE
ACALL     ONEWIRE_DELAY ; Delay 75 us
SETB      ONEWIRE
NOP
NOP
NOP
NOP
DS1820_WR_NX:
DJNZ     R4,DS1820_WR_LOOP ; Do until 8 times
RET
;-----
; DS1820 Reset
;-----
DS1820_RST:
CLR      ONEWIRE      ; Clear ONEWIRE
ACALL    DELAY_1ms    ; Delay
SETB     ONEWIRE      ; Set ONEWIRE
MOV      R4,#8
DJNZ     R4,$
RET
;-----
; DS1820 Receive Presence Pulse
;-----
DS1820_PRES:
MOV      R4,#8      ; Set Loop wait 1
DS1820_PRES_1:
MOV      R3,#0      ; Set Loop wait 2
DS1820_PRES_2:
JNB      ONEWIRE,DS1820_PRES_3
DJNZ     R3,DS1820_PRES_2 ; Wait loop check 2
DJNZ     R4,DS1820_PRES_1 ; Wait loop check 1
RET

```

```

DS1820_PRES_3:  JNB      ONEWIRE,$    ; Wait until ONEWIRE set
                 MOV      R4,#8
                 DJNZ     R4,$
                 CLR     BUSY      ; Clear BUSY Flag
                 RET

```

```

;-----

```

```

; INITIAL LCD DISPLAY

```

```

;-----

```

```

TITLE_SYSTEM:  MOV      LCD_ADDR,#000H    ; Set Address 00H
                ACALL   SET_ADDR_LCD
                MOV     DPTR,#TITLE_1     ; LCD Show "VECHICLE"
                ACALL   WR_LCD
                MOV     DPTR,#TITLE_3     ; LCD Show "USING..."
                ACALL   WR_LCD
                MOV     LCD_ADDR,#040H    ; Set Address 40H
                ACALL   SET_ADDR_LCD
                MOV     DPTR,#TITLE_2     ; LCD Show "ROBOT BY"
                ACALL   WR_LCD
                MOV     DPTR,#TITLE_4     ; LCD Show "COMPUTER"
                ACALL   WR_LCD
                ACALL   DELAY_1s
                ACALL   DELAY_1s
                MOV     R4,#8              ; Set Loop 8 times
LOOP_LCD_L_SHF: ACALL   LCD_LSHF          ; Left Shift LCD Display
                ACALL   DELAY_100ms
                ACALL   DELAY_100ms
                DJNZ   R4,LOOP_LCD_L_SHF ; Do until 8 times
                MOV     LCD_ADDR,#000H    ; Set Address 00H
                ACALL   SET_ADDR_LCD
                MOV     DPTR,#TITLE_5     ; LCD Show "SYSTEM.."

```

```

ACALL    WR_LCD
MOV      LCD_ADDR,#040H    ; Set Address 40H
ACALL    SET_ADDR_LCD
MOV      DPTR,#TITLE_6    ; LCD Show "...OK..."
ACALL    WR_LCD            ; 40H-47H (Increase automatic)
ACALL    DELAY_1s
ACALL    DELAY_1s
MOV      R4,#8            ; Set Loop 8 times
LOOP_LCD_R_SHF: ACALL    LCD_RSHF    ; Right Shift LCD Display
ACALL    DELAY_100ms
ACALL    DELAY_100ms
DJNZ    R4,LOOP_LCD_R_SHF ; Do until 8 times
ACALL    DELAY_1s
ACALL    DELAY_1s
RET
;-----
; WAIT DATA FROM RX
;-----
WAIT_DATA_RX: MOV      DPTR,#WAIT1    ; Index Pointer ROM to Show LCD
ACALL    WR_LCD            ; 00H-07H (Increase automatic)
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD
MOV      DPTR,#WAIT2
ACALL    WR_LCD
RET
;-----
OP_CHK_DATA: MOV      DPTR,#OPERATING1
ACALL    WR_LCD            ; 00H-07H (Increase automatic)
MOV      LCD_ADDR,#040H
ACALL    SET_ADDR_LCD

```

```

MOV      DPTR,#OPERATING2
ACALL   WR_LCD
RET

```

```

;-----
; LCD Initialize
;-----

```

```

INIT_LCD:    ACALL    DELAY_10ms
             CLR      LCD_RS
             MOV      P0,#00111000B ; ( Order define function of operate )
             ; 8 bit, show 2 line,type of display 5*7 points
             ACALL   LCD_CLK      ; Pulse LCD Clock ( falling end )
             ACALL   DELAY_10ms
             MOV      P0,#00000110B ; Entry Mode
             ACALL   LCD_CLK      ; Pulse LCD Clock
             ACALL   LCD_CLR      ; Clear Display
             ACALL   LCD_ON       ; Display on
             RET

```

```

;-----
; LCD Clear Display
;-----

```

```

LCD_CLR:    CLR      LCD_RS      ; Clear LCD_RS Pin for order
             MOV      P0,#00000001B ; Display Clear for emtry lcd
             ACALL   LCD_CLK      ; Pulse LCD Clock
             RET

```

```

;-----
; LCD Return Home
;-----

```

```

LCD_HOME:   CLR      LCD_RS
             MOV      P0,#00000010B ; Return Home
             ACALL   LCD_CLK      ; Pulse LCD Clock

```

RET

; LCD Display Off

LCD_OFF: CLR LCD_RS ; Clear LCD_RS Pin for order

MOV P0,#00001000B ; Display Off

ACALL LCD_CLK ; Pulse LCD Clock

RET

; LCD Clk

LCD_CLK: SETB LCD_EN ; Pulse Clock to LCD_EN

ACALL LCD_DELAY

CLR LCD_EN

ACALL LCD_DELAY

RET

; LCD Display On

LCD_ON: CLR LCD_RS

MOV P0,#00001100B ; Display On

ACALL LCD_CLK ; Pulse LCD Clock

RET

; LCD Left Shift Display

LCD_LSHF: CLR LCD_RS

MOV P0,#00011000B ; Left Shift Display

ACALL LCD_CLK ; Pulse LCD Clock

RET

;

; LCD Right Shift Display

;

```
LCD_RSHF:      CLR      LCD_RS
                MOV      P0,#00011100B ; Right Shift Display
                ACALL    LCD_CLK      ; Pulse LCD Clock
                RET
```

;

; Set LCD Address

;

```
SET_ADDR_LCD:  CLR      LCD_RS
                MOV      A,LCD_ADDR
                SETB     ACC.7
                MOV      P0,A
                ACALL    LCD_CLK      ; Pulse LCD Clock
                RET
```

;

```
WR_LCD:        MOV      R0,#0      ; Clear loop counter
```

```
WR_LCD_1:      SETB     LCD_RS
                CLR      A
                MOVC    A,@A+DPTR
                MOV      P0,A
                ACALL    LCD_CLK      ; Pulse LCD Clock
                INC      DPTR        ; Increase Pointer
                INC      R0          ; Increase loop counter
                CJNE    R0,#8,WR_LCD_1 ; Do until 8 times
                RET
```

```
WRCHAR_LCD:    SETB     LCD_RS
                MOV      P0,LCD_DATA
                ACALL    LCD_CLK
```

```

ACALL      LCD_ON
RET

;-----
; Delay time  ONEWIRE_DELAY, LCD_DELAY, 50u, 100u, 1m, 10m, 100m, 1s
;-----

ONEWIRE_DELAY:  MOV      R6,#012H          ; Each loop = 75 us
ONEWIRE_DELAY_1: NOP
                NOP
                DJNZ     R6,ONEWIRE_DELAY_1
                RET
DELAY_50us:     MOV      R6,#00CH          ; Each loop = 50 us
DELAY_50us_1:  NOP
                NOP
                DJNZ     R6,DELAY_50us_1
                RET
DELAY_100us:   MOV      R6,#017H          ; Each loop = 100 us
DELAY_100us_1: NOP
                NOP
                DJNZ     R6,DELAY_100us_1
                RET
DELAY_1ms:     MOV      R6,#0E6H          ; Each loop = 1 ms
DELAY_1ms_1:   NOP
                NOP
                DJNZ     R6,DELAY_1ms_1
                RET
LCD_DELAY:     MOV      R7,#002          ; Do 2 times
LCD_DELAY_1:   MOV      R6,#0E6H          ; Each loop = 1 ms
LCD_DELAY_2:   NOP
                NOP
                DJNZ     R6,LCD_DELAY_2

```

```

                                DJNZ      R7,LCD_DELAY_1
                                RET
DELAY_10ms:      MOV      R7,#010          ; Do 10 times
DELAY_10ms_1:   MOV      R6,#0E6H        ; Each loop = 1 ms
DELAY_10ms_2:   NOP
                                NOP
                                DJNZ      R6,DELAY_10ms_2
                                DJNZ      R7,DELAY_10ms_1
                                RET
DELAY_100ms:    MOV      R7,#100         ; Do 100 times
DELAY_100ms_1:  MOV      R6,#0E6H        ; Each loop = 1 ms
DELAY_100ms_2:  NOP
                                NOP
                                DJNZ      R6,DELAY_100ms_2
                                DJNZ      R7,DELAY_100ms_1
                                RET
DELAY_1s:       MOV      R5,#100         ; Do 100 times
DELAY_1s_1:     ACALL    DELAY_10ms
                                DJNZ      R5,DELAY_1s_1
                                RET

```

;Define Constant < Store in Flash Program Memory >

```

TITLE_1:      DB      'VEHICLE '
TITLE_2:      DB      'ROBOT BY'
TITLE_3:      DB      'USING...'
TITLE_4:      DB      'COMPUTER'
TITLE_5:      DB      'SYSTEM..'
TITLE_6:      DB      '...OK...'
OPERATING1:   DB      'OPERATE '

```

OPERATING2:	DB	'CHK DATA'
WAIT1:	DB	'WaitData'
WAIT2:	DB	' from Rx'
WAIT_TX_ON1:	DB	'Wait Tx '
WAIT_TX_ON2:	DB	'ON '
DATA_ERROR1:	DB	'Data err'
DATA_ERROR2:	DB	'or.....'
FAIL_CON1:	DB	'FAIL.CON'
FAIL_CON2:	DB	'NECTION!'
OP_FW1:	DB	'ROBOT...'
OP_FW2:	DB	'FORWARD.'
OP_BW1:	DB	'ROBOT..'
OP_BW2:	DB	'BACKWARD'
OP_TR451:	DB	'ROBOT...'
OP_TR452:	DB	'TR 45...'
OP_TR1351:	DB	'ROBOT...'
OP_TR1352:	DB	'TR 135..'
OP_TL451:	DB	'ROBOT...'
OP_TL452:	DB	'TL 45...'
OP_TL1351:	DB	'ROBOT...'
OP_TL1352:	DB	'TL 135..'
OP_TR1:	DB	'ROBOT TU'
OP_TR2:	DB	'RNRIGHT.'
OP_TL1:	DB	'ROBOT TU'
OP_TL2:	DB	'RNLEFT..'
OP_TEMP1:	DB	'CHK.TEMP'
OP_TEMP2:	DB	'ERATURE.'
OP_STOP1:	DB	'ROBOT...'
OP_STOP2:	DB	'STOP....'
		END



โปรแกรมควบคุมการทำงานของยานพาหนะหุ่นยนต์

```
Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

Private Sub Command11_Click ()

Label2.Caption = "STOP"

MSComm1.Output = "I"

End Sub

Private Sub Command1_Click()

End

End Sub

Private Sub Command10_Click()

MSComm1.Output = "I"

Sleep (400)

Label2.Caption = "TURN LEFT 45"

MSComm1.Output = "G"

End Sub

Private Sub Command16_Click()

MSComm1.Output = "J"

Sleep (400)

Label2.Caption = "CHECK TEMPERATURE"

MSComm1.Output = "h"

End Sub

Private Sub Command3_Click()

MSComm1.Output = "I"

Sleep (400)

Label2.Caption = "FORWARD"

MSComm1.Output = "F"

End Sub

Private Sub Command4_Click()
```

```

MSComm1.Output = "I"
Sleep (400)
Label2.Caption = "TURN RIGHT 45"
MSComm1.Output = "A"
End Sub

Private Sub Command5_Click()
MSComm1.Output = "I"
Sleep (400)
Label2.Caption = "TURN RIGHT"
MSComm1.Output = "n"
End Sub

Private Sub Command6_Click()
MSComm1.Output = "I"
Sleep (400)
Label2.Caption = "TURN RIGHT 135"
MSComm1.Output = "D"
End Sub

Private Sub Command7_Click()
MSComm1.Output = "I"
Sleep (400)
Label2.Caption = "BACKWARD"
MSComm1.Output = "B"
End Sub

Private Sub Command8_Click()
MSComm1.Output = "I"
Sleep (400)
Label2.Caption = "TURN LEFT 135"
MSComm1.Output = "I"
End Sub

Private Sub Command9_Click()

```



```
MSComm1.Output = "l"
Sleep (400)
Label2.Caption = "TURN LEFT"
MSComm1.Output = "o"
End Sub

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
MSComm1.Output = "l"
Sleep (400)
Select Case KeyCode
Case 96: Call Command1_Click
Case 97: Call Command8_Click
Case 98: Call Command7_Click
Case 99: Call Command6_Click
Case 100: Call Command9_Click
Case 101: Call Command11_Click
Case 102: Call Command5_Click
Case 103: Call Command10_Click
Case 104: Call Command3_Click
Case 105: Call Command4_Click
Case 107: Call Command16_Click
End Select
End Sub

Private Sub Form_Load()
MSComm1.CommPort = 2
MSComm1.Settings = "1200,N,8,1"
MSComm1.InputLen = 0
MSComm1.PortOpen = True
End Sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTICE

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Copyright 1995 EXAR Corporation

Datasheet June 1997

Reproduction, in part or whole, without the prior written consent of EXAR Corporation is prohibited.

Notes

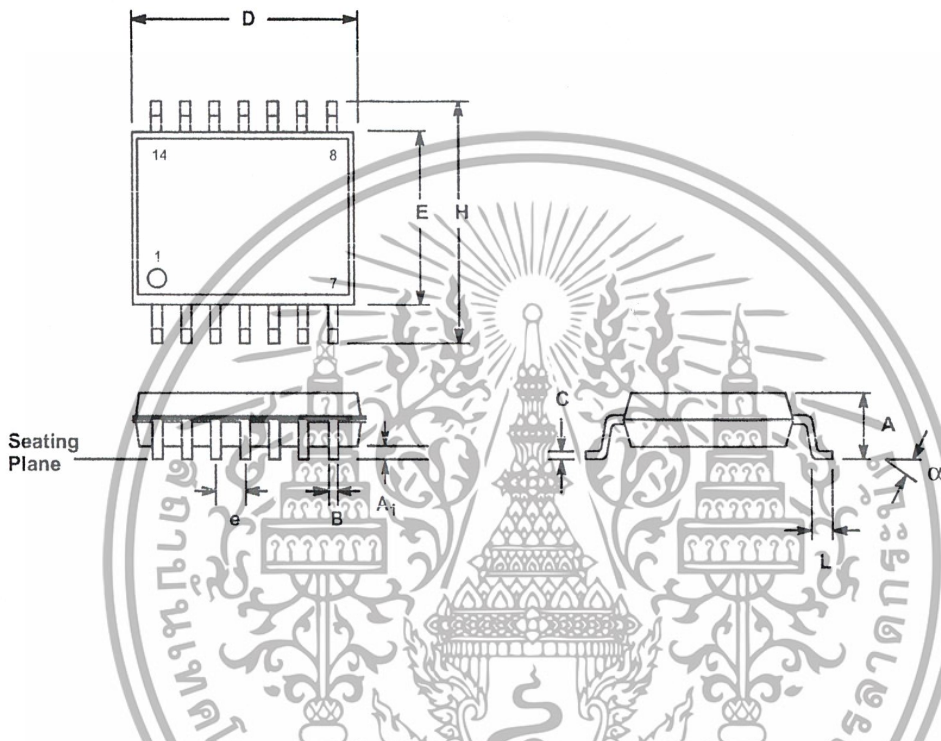


Notes



14 LEAD SMALL OUTLINE
(150 MIL JEDEC SOIC)

Rev. 1.00

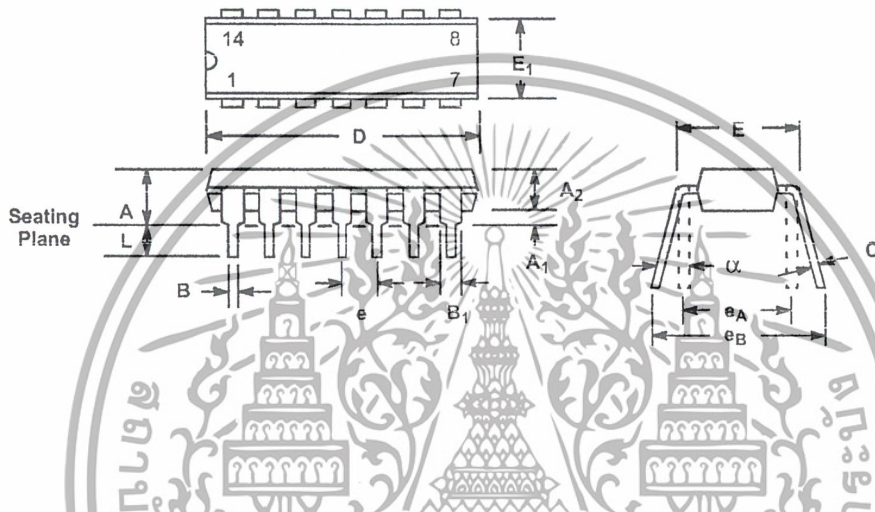


SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.053	0.069	1.35	1.75
A ₁	0.004	0.010	0.10	0.25
B	0.013	0.020	0.33	0.51
C	0.007	0.010	0.19	0.25
D	0.337	0.344	8.55	8.75
E	0.150	0.157	3.80	4.00
e	0.050 BSC		1.27 BSC	
H	0.228	0.244	5.80	6.20
L	0.016	0.050	0.40	1.27
α	0°	8°	0°	8°

Note: The control dimension is the millimeter column

14 LEAD PLASTIC DUAL-IN-LINE
(300 MIL PDIP)

Rev. 1.00

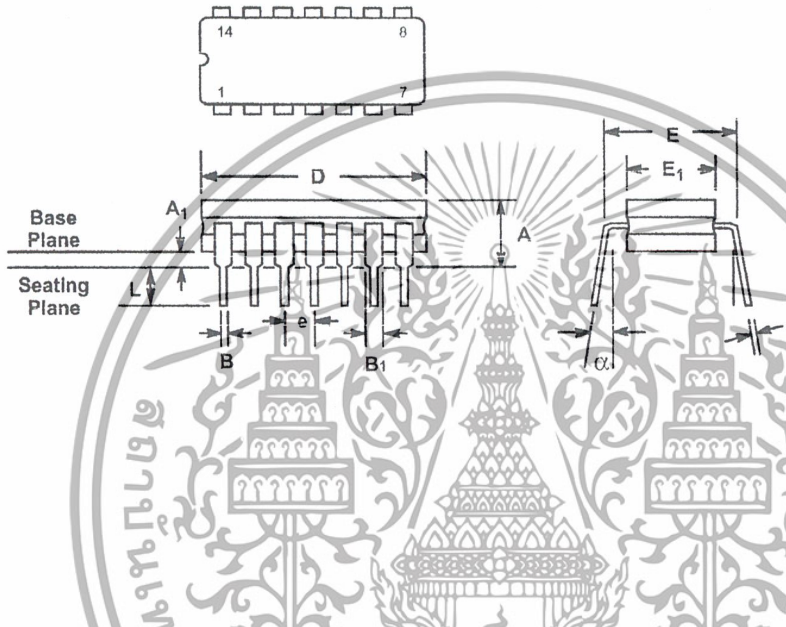


SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.145	0.210	3.68	5.33
A ₁	0.015	0.070	0.38	1.78
A ₂	0.115	0.195	2.92	4.95
B	0.014	0.024	0.36	0.56
B ₁	0.030	0.070	0.76	1.78
C	0.008	0.014	0.20	0.38
D	0.725	0.795	18.42	20.19
E	0.300	0.325	7.62	8.26
E ₁	0.240	0.280	6.10	7.11
e	0.100 BSC		2.54 BSC	
e _A	0.300 BSC		7.62 BSC	
e _B	0.310	0.430	7.87	10.92
L	0.115	0.160	2.92	4.06
α	0°	15°	0°	15°

Note: The control dimension is the inch column

14 LEAD CERAMIC DUAL-IN-LINE
(300 MIL CDIP)

Rev. 1.00



SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.100	0.200	2.54	5.08
A ₁	0.015	0.060	0.38	1.52
B	0.014	0.026	0.36	0.66
B ₁	0.045	0.065	1.14	1.65
c	0.008	0.018	0.20	0.46
D	0.685	0.785	17.40	19.94
E ₁	0.250	0.310	6.35	7.87
E	0.300 BSC		7.62 BSC	
e	0.100 BSC		2.54 BSC	
L	0.125	0.200	3.18	5.08
α	0°	15°	0°	15°

Note: The control dimension is the inch column

Linear FM Detection

XR-2211 can be used as a linear FM detector for a wide range of analog communications and telemetry applications. The recommended circuit connection for this application is shown in *Figure 13*. The demodulated output is taken from the loop phase detector output (pin 11), through a post-detection filter made up of R_F and C_F and an external buffer amplifier. This buffer amplifier is necessary because of the high impedance output at pin 11. Normally, a non-inverting unity gain op amp can be used as a buffer amplifier, as shown in *Figure 13*.

The FM detector gain, i.e., the output voltage change per unit of FM deviation can be given as:

$$V_{OUT} = \frac{R_1 \cdot V_{REF}}{100 \cdot R_0}$$

where V_R is the internal reference voltage ($V_{REF} = V_{CC} / 2 - 650mV$). For the choice of external components R_1 , R_0 , C_D , C_1 and C_F see the section on design equations.

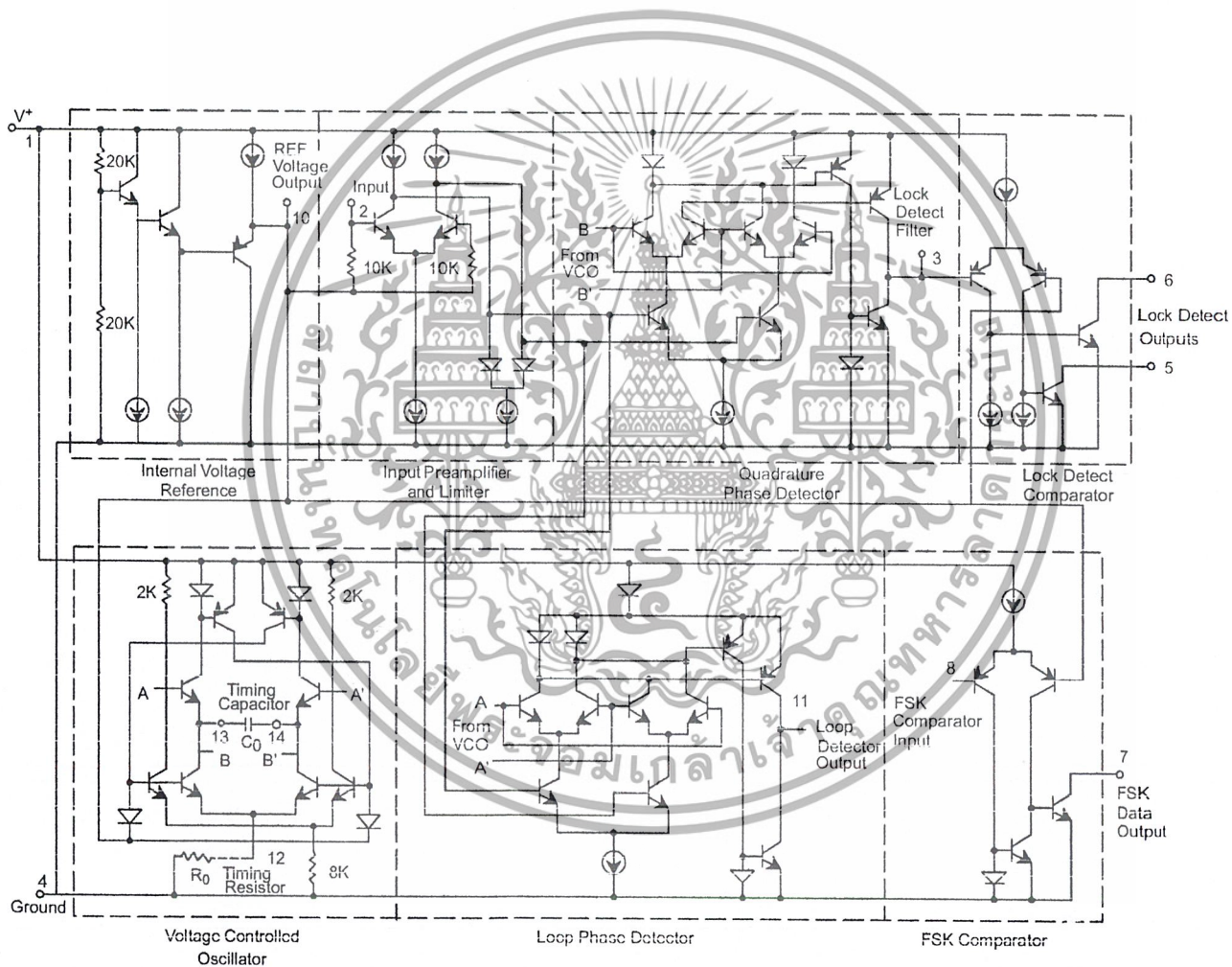


Figure 14. Equivalent Schematic Diagram

c) Calculate R_1 to set the bandwidth $\pm\Delta f$ (See design equation 5):

$$R_1 = \frac{R_0 \cdot f_0 \cdot 2}{\Delta f} = \frac{20,000 \cdot 1,000 \cdot 2}{100} = 400K$$

Note: The total detection bandwidth covers the frequency range of $f_0 \pm \Delta f$

d) Calculate value of C_0 for a given loop damping factor:

Normally, $\zeta = 0.5$ is recommended.

$$C_1 = \frac{1250 \cdot C_0}{R_1 \cdot \zeta^2} = \frac{1250 \cdot 50 \cdot 10^{-9}}{400,000 \cdot 0.5^2} = 6.25 pF$$

Increasing C_1 improves the out-of-band signal rejection, but increases the PLL capture time.

e) Calculate value of the filter capacitor C_D . To avoid chatter at the logic output, with $R_D = 470K\Omega$, C_D must be:

$$C_D = \frac{16}{\Delta f} \geq \frac{16}{200} \geq 80 nF$$

Increasing C_D slows down the logic output response time.

f) Fine tune center frequency with $5K\Omega$ potentiometer, R_X .

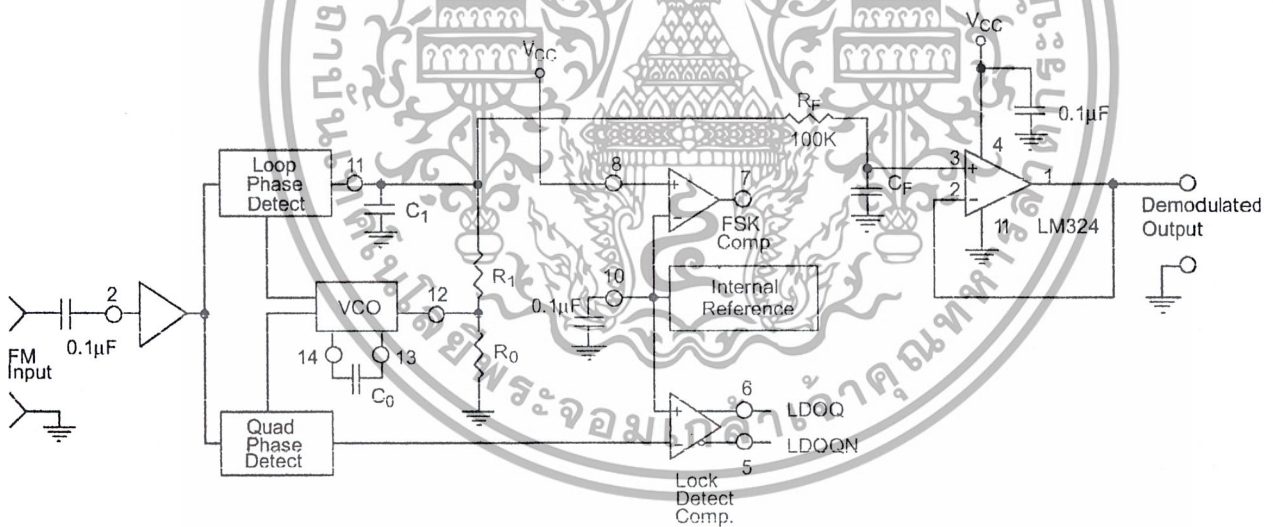


Figure 13. Linear FM Detector Using XR-2211 and an External Op Amp.
(See Section on Design Equation for Component Values.)

Design Instructions:

The circuit of *Figure 12* can be optimized for any tone detection application by the choice of the 5 key circuit components: R_0 , R_1 , C_0 , C_1 and C_D . For a given input, the tone frequency, f_S , these parameters are calculated as follows:

(All resistance in Ω 's, all frequency in Hz and all capacitance in farads, unless otherwise specified)

- Choose value of timing resistor R_0 to be in the range of $10K\Omega$ to $50K\Omega$. This choice is dictated by the max./min. current that the internal voltage reference can deliver. The recommended value is $R_0 = 20K\Omega$. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X .
- Calculate value of C_0 from design equation (1) or from *Figure 7* $f_S = f_O$:

$$C_0 = \frac{1}{R_0 \cdot f_S}$$

- Calculate R_1 to set the bandwidth $\pm \Delta f$ (See design equation 5):

$$R_1 = \frac{R_0 \cdot f_0 \cdot 2}{\Delta f}$$

Note: The total detection bandwidth covers the frequency range of $f_0 \pm \Delta f$

- Calculate value of C_1 for a given loop damping factor:

Normally, $\zeta = 0.5$ is recommended.

$$C_1 = \frac{1250 \cdot C_0}{R_1 \cdot \zeta^2}$$

Increasing C_1 improves the out-of-band signal rejection, but increases the PLL capture time.

- Calculate value of the filter capacitor C_D . To avoid chatter at the logic output, with $R_D = 470K\Omega$, C_D must be:

$$C_D > \frac{16}{\Delta f} \quad C \text{ in } \mu F$$

Increasing C_D slows down the logic output response time.

Design Examples:

Tone detector with a detection band of $\pm 100\text{Hz}$:

- Choose value of timing resistor R_0 to be in the range of $10K\Omega$ to $50K\Omega$. This choice is dictated by the max./min. current that the internal voltage reference can deliver. The recommended value is $R_0 = 20K\Omega$. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X .
- Calculate value of C_0 from design equation (1) or from *Figure 6* $f_S = f_O$:

$$C_0 = \frac{1}{R_0 \cdot f_S} = \frac{1}{20,000 \cdot 1,000} = 50nF$$

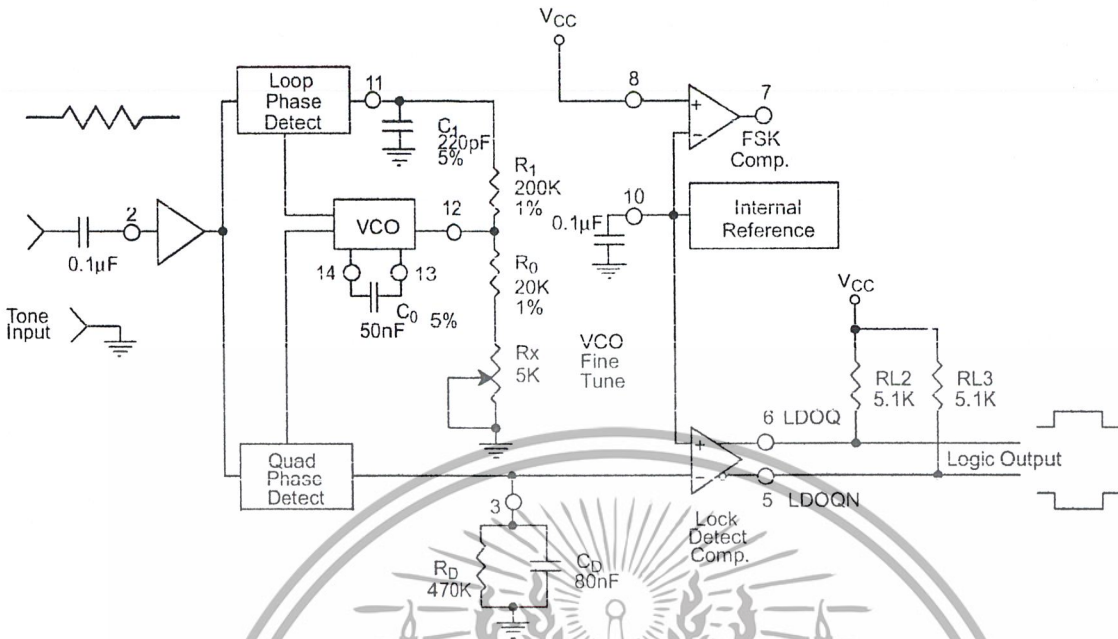


Figure 12. Circuit Connection for Tone Detection

FSK Decoding with Carrier Detect

The lock detect section of XR-2211 can be used as a carrier detect option for FSK decoding. The recommended circuit connection for this application is shown in Figure 11. The open collector lock detect output, pin 6, is shorted to data output (pin 7). Thus, data output will be disabled at "low" state, until there is a carrier within the detection band of the PLL and the pin 6 output goes "high" to enable the data output.

Note: Data Output is "Low" When No Carrier is Present.

The minimum value of the lock detect filter capacitance C_D is inversely proportional to the capture range, $\pm\Delta f_c$. This is the range of incoming frequencies over which the loop can acquire lock and is always less than the tracking range. It is further limited by C_1 . For most applications, $\Delta f_c > \Delta f/2$. For $R_D = 470K\Omega$, the approximate minimum value of C_D can be determined by:

$$C_D > \frac{16}{\Delta f} \quad C \text{ in } \mu F \text{ and } f \text{ in Hz.}$$

C in μF and f in Hz.

With values of C_D that are too small, chatter can be observed on the lock detect output as an incoming signal

frequency approaches the capture bandwidth. Excessively large values of C_D will slow the response time of the lock detect output. For Caller I.D. applications choose $C_D = 0.1\mu F$.

Tone Detection

Figure 12 shows the generalized circuit connection for tone detection. The logic outputs, LDOQN and LDOQ at pins 5 and 6 are normally at "high" and "low" logic states, respectively. When a tone is present within the detection band of the PLL, the logic state at these outputs become reversed for the duration of the input tone. Each logic output can sink 5mA of load current.

Both outputs at pins 5 and 6 are open collector type stages, and require external pull-up resistors R_{L2} and R_{L3} , as shown in Figure 12.

With reference to Figure 3 and Figure 12, the functions of the external circuit components can be explained as follows: R_0 and C_0 set VCO center frequency; R_1 sets the detection bandwidth; C_1 sets the low pass-loop filter time constant and the loop damping factor.

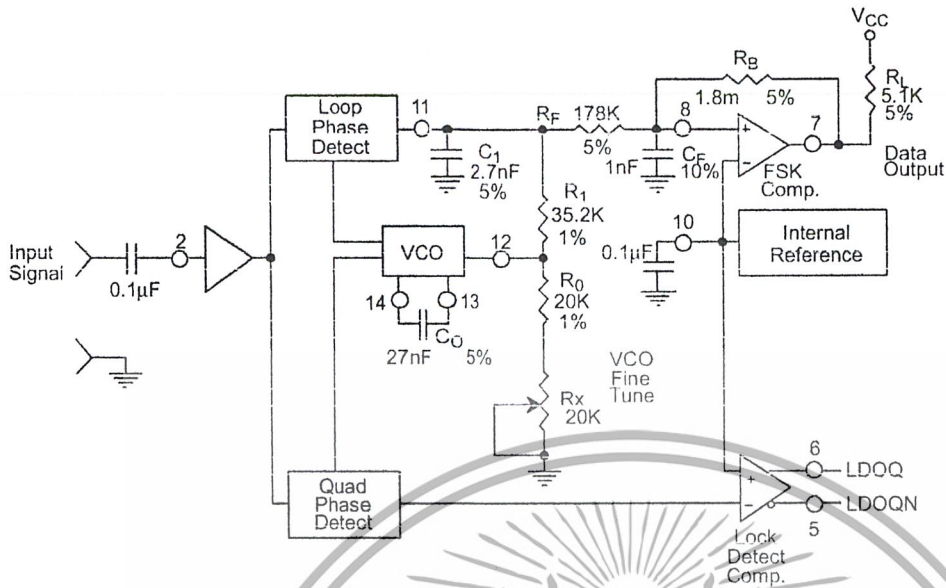


Figure 10. Circuit Connection for FSK Decoding of Caller Identification Signals (Bell 202 Format)

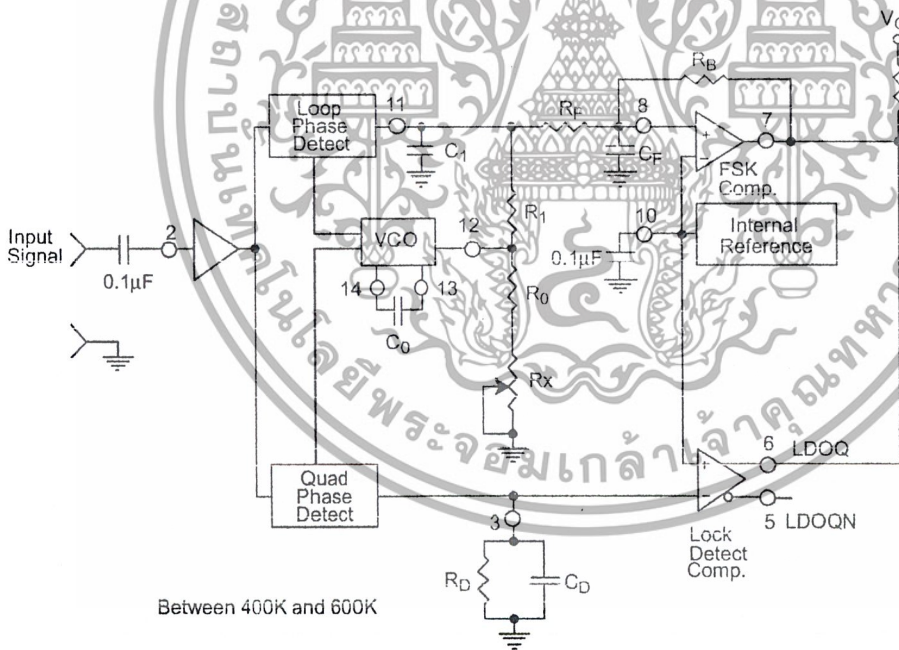


Figure 11. External Connectors for FSK Demodulation with Carrier Detect Capability

Design Example:**1200 Baud FSK demodulator with mark and space frequencies of 1200/2200.**

Step 1: Calculate f_0 : from design instructions

$$(a) f_0 = \sqrt{1200 \cdot 2200} = 1624$$

Step 2: Calculate R_0 : $R_0 = 10K$ with a potentiometer of 10K. (See design instructions (b))

$$(b) R_T = 10 + \left(\frac{10}{2}\right) = 15K$$

Step 3: Calculate C_0 from design instructions

$$(c) C_0 = \frac{1}{15000 \cdot 1624} = 39nF$$

Step 4: Calculate R_1 : from design instructions

$$(d) R_1 = \frac{20000 \cdot 1624 \cdot 2}{(2200 - 1200)} = 51,000$$

Step 5: Calculate C_1 : from design instructions

$$(e) C_1 = \frac{1250 \cdot 39nF}{51000 \cdot 0.5^2} = 3.9nF$$

Step 6: Calculate R_F : R_F should be at least five times R_1 , $R_F = 51,000 \cdot 5 = 255 K\Omega$

Step 7: Calculate R_B : R_B should be at least five times R_F , $R_B = 255,000 \cdot 5 = 1.2 M\Omega$

Step 8: Calculate R_{SUM} :

$$R_{SUM} = \frac{(R_F + R_1) \cdot R_B}{(R_F + R_1 + R_B)} = 240K\Omega$$

Step 9: Calculate C_F :

$$C_F = \frac{0.25}{(R_{SUM} \cdot \text{Baud Rate})} = 1nF$$

Note: All values except R_0 can be rounded to nearest standard value.

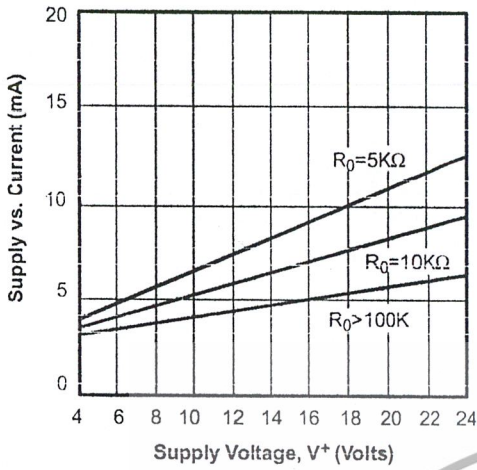


Figure 5. Typical Supply Current vs. V+ (Logic Outputs Open Circuited)

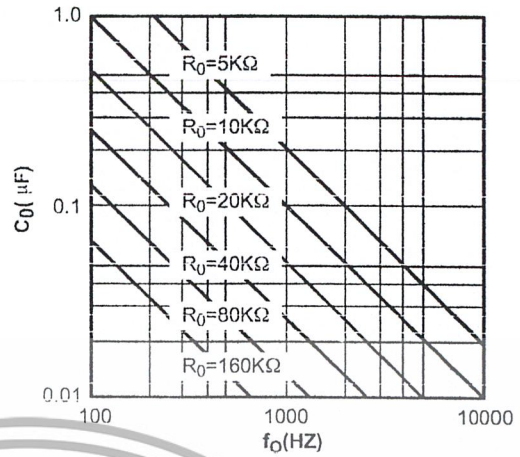


Figure 6. VCO Frequency vs. Timing Resistor

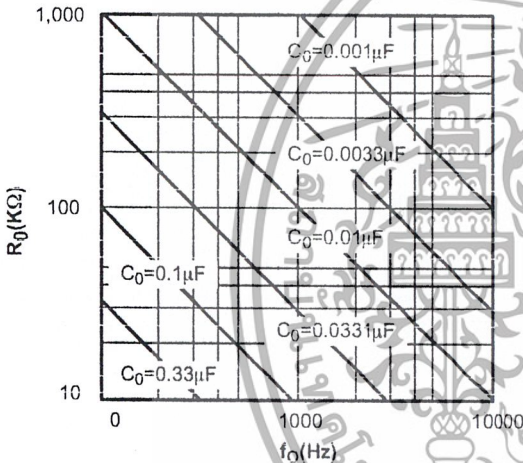


Figure 7. VCO Frequency vs. Timing Capacitor

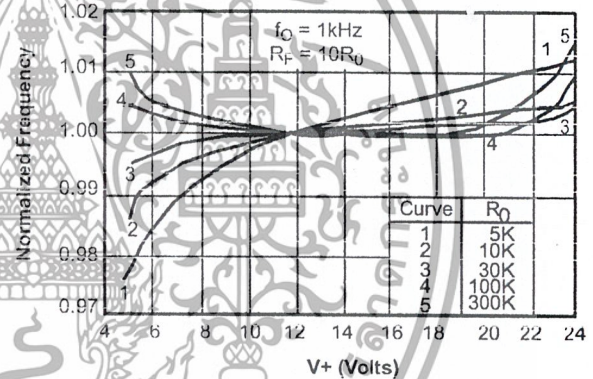


Figure 8. Typical f_0 vs. Power Supply Characteristics

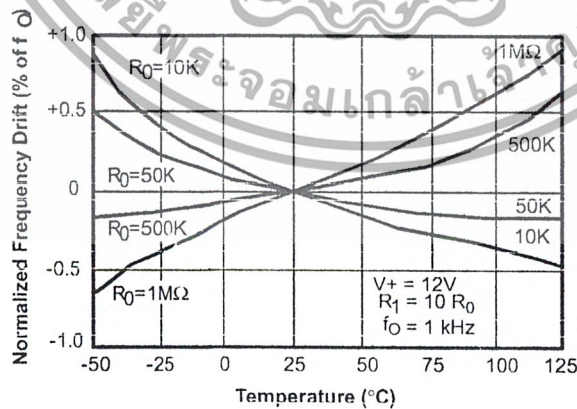


Figure 9. Typical Center Frequency Drift vs. Temperature

- f) The input to the XR-2211 may sometimes be too sensitive to noise conditions on the input line. *Figure 4* illustrates a method of de-sensitizing the XR-2211 from such noisy line conditions by the use of a resistor, R_x , connected from pin 2 to ground. The value of R_x is chosen by the equation and the desired minimum signal threshold level.

$$V_{IN \text{ minimum (peak)}} = V_a - V_b = \Delta V \pm 2.8mV \text{ offset} = V_{REF} \frac{20,000}{(20,000 + R_x)} \text{ or } R_x = 20,000 \left(\frac{V_{REF}}{\Delta V} - 1 \right)$$

V_{IN} minimum (peak) input voltage must exceed this value to be detected (equivalent to adjusting V threshold)

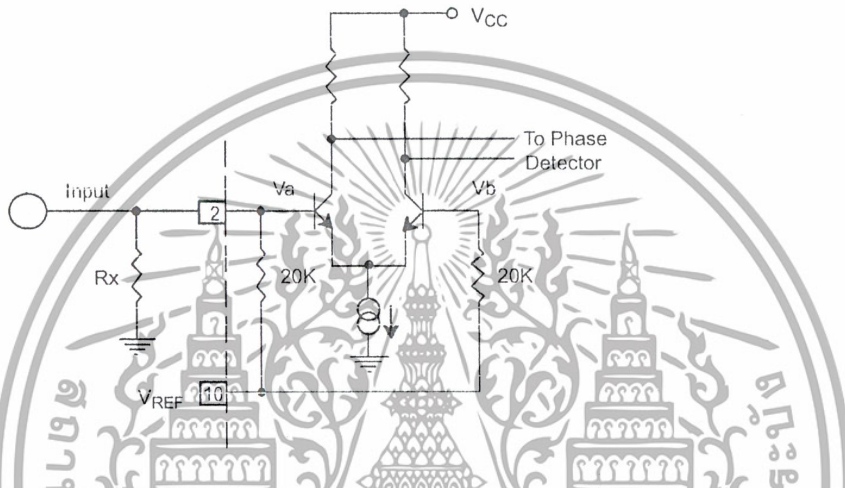


Figure 4. Desensitizing Input Stage

- g) Calculate Data Filter Capacitance, C_F :

$$R_{sum} = \frac{(R_F + R_1) \cdot R_B}{(R_1 + R_F + R_B)}$$

$$C_F = \frac{0.25}{(R_{sum} \cdot \text{Baud Rate})} \quad \text{Baud rate in } \frac{1}{\text{seconds}}$$

Note: All values except R_0 can be rounded to nearest standard value.

APPLICATIONS INFORMATION

FSK Decoding

Figure 10 shows the basic circuit connection for FSK decoding. With reference to Figure 3 and Figure 10, the functions of external components are defined as follows: R_0 and C_0 set the PLL center frequency, R_1 sets the system bandwidth, and C_1 sets the loop filter time constant and the loop damping factor. C_F and R_F form a one-pole post-detection filter for the FSK data output. The resistor R_B from pin 7 to pin 8 introduces positive feedback across the FSK comparator to facilitate rapid transition between output logic states.

Design Instructions:

The circuit of Figure 10 can be tailored for any FSK decoding application by the choice of five key circuit components: R_0 , R_1 , C_0 , C_1 and C_F . For a given set of FSK mark and space frequencies, f_0 and f_1 , these parameters can be calculated as follows:

(All resistance in Ω 's, all frequency in Hz and all capacitance in farads, unless otherwise specified)

- a) Calculate PLL center frequency, f_0 :

$$f_0 = \sqrt{F_1 \cdot F_2}$$

- b) Choose value of timing resistor R_0 , to be in the range of 10K Ω to 100K Ω . This choice is arbitrary. The recommended value is $R_0 = 20K\Omega$. The final value of R_0 is normally fine-tuned with the series potentiometer, R_X .

$$R_o = R_0 + \frac{R_X}{2}$$

- c) Calculate value of C_0 from design equation (1) or from Figure 7:

$$C_o = \frac{1}{R_o \cdot f_0}$$

- d) Calculate R_1 to give the desired tracking bandwidth (See design equation 5).

$$R_1 = \frac{R_o \cdot f_0}{(f_1 - f_2)} \cdot 2$$

- e) Calculate C_1 to set loop damping. (See design equation 4):

Normally, $\zeta = 0.5$ is recommended.

$$C_1 = \frac{1250 \cdot C_o}{R_1 \cdot \zeta^2}$$

6. FSK Data filter time constant, t_F :

$$\tau_F = \frac{R_B \cdot R_F}{(R_B + R_F)} \cdot C_F \text{ (seconds)}$$

7. Loop phase detector conversion gain, K_d : (K_d is the differential DC voltage across pin 10 and pin11, per unit of phase error at phase detector input):

$$K_d = \frac{V_{REF} \cdot R_1}{10,000 \cdot \pi} \left[\frac{\text{volt}}{\text{radian}} \right]$$

Note: For derivation/explanation of this equation, please see TAN-011.

8. VCO conversion gain, K_o : (K_o is the amount of change in VCO frequency, per unit of DC voltage change at pin 11):

$$K_o = \frac{-2\pi}{V_{REF} \cdot C_o \cdot R_1} = \left(\frac{\text{radian/second}}{\text{volt}} \right)$$

9. The filter transfer function:

$$F(s) = \frac{1}{1 + sR_1 \cdot C_1} \text{ at } 0 \text{ Hz. } \quad S = j\omega \text{ and } \omega = 0$$

10. Total loop gain, K_T :

$$K_T = K_o \cdot K_d \cdot F(s) = \left(\frac{R_F}{5,000 \cdot C_o \cdot (R_1 + R_F)} \right) \left[\frac{1}{\text{seconds}} \right]$$

11. Peak detector current I_A :

$$I_A = \frac{V_{REF}}{20,000} \text{ (} V_{REF} \text{ in volts and } I_A \text{ in amps)}$$

Note: For derivation/explanation of this equation, please see TAN-011.



DESIGN EQUATIONS

(All resistance in Ω , all frequency in Hz and all capacitance in farads, unless otherwise specified)

(See Figure 3 for definition of components)

1. VCO Center Frequency, f_0 :

$$f_0 = \frac{1}{R_0 \cdot C_0}$$

2. Internal Reference Voltage, V_{REF} (measured at pin 10):

$$V_{REF} = \left(\frac{V_{CC}}{2} \right) - 650mV \text{ in volts}$$

3. Loop Low-Pass Filter Time Constant, τ :

$$\tau = C_1 \cdot R_{PP} \text{ (seconds)}$$

where:

$$R_{PP} = \left(\frac{R_1 \cdot R_F}{R_1 + R_F} \right)$$

if R_F is ∞ or C_F reactance is ∞ , then $R_{PP} = R_1$

4. Loop Damping, ζ :

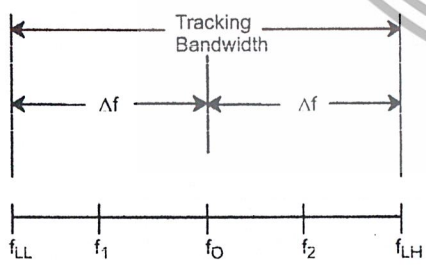
$$\zeta = \sqrt{\left(\frac{1250 \cdot C_0}{R_1 \cdot C_1} \right)}$$

Note: For derivation/explanation of this equation, please see TAN-011.

5. Loop-tracking

bandwidth, $\pm = \frac{\Delta f}{f_0}$

$$\frac{\Delta f}{f_0} = \frac{R_0}{R_1}$$



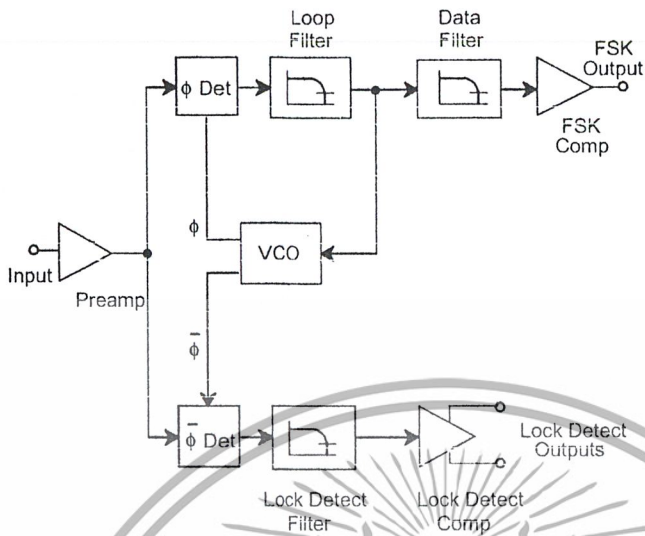


Figure 2. Functional Block Diagram of a Tone and FSK Decoding System Using XR-2211

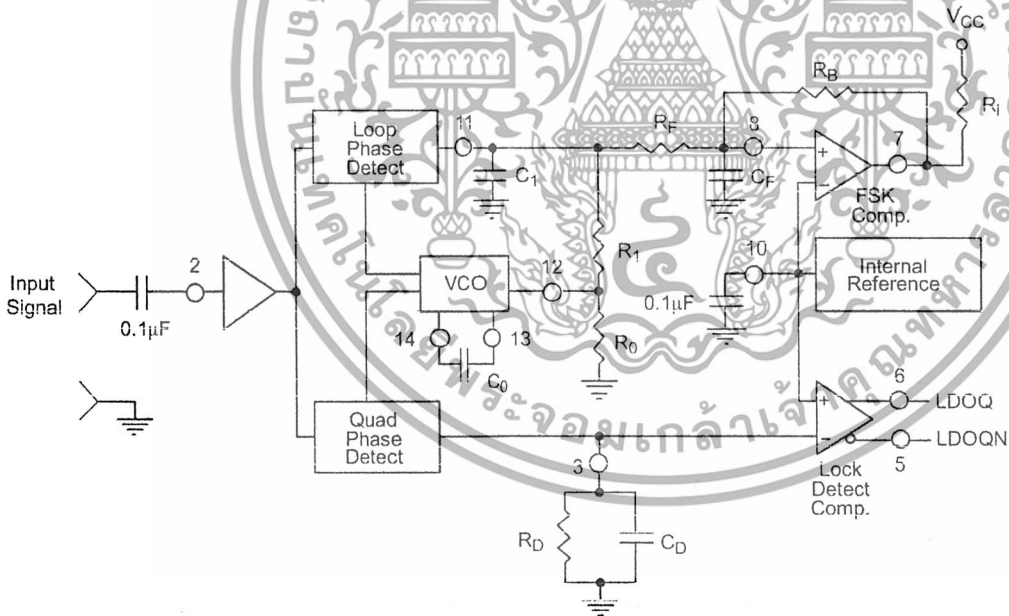


Figure 3. Generalized Circuit Connection for FSK and Tone Detection

PRINCIPLES OF OPERATION

Signal Input (Pin 2): Signal is AC coupled to this terminal. The internal impedance at pin 2 is 20KΩ. Recommended input signal level is in the range of 10mV rms to 3V rms.

Quadrature Phase Detector Output (Pin 3): This is the high impedance output of quadrature phase detector and is internally connected to the input of lock detect voltage comparator. In tone detection applications, pin 3 is connected to ground through a parallel combination of R_D and C_D (see *Figure 3*) to eliminate the chatter at lock detect outputs. If the tone detect section is not used, pin 3 can be left open.

Lock Detect Output, Q (Pin 6): The output at pin 6 is at "low" state when the PLL is out of lock and goes to "high" state when the PLL is locked. It is an open collector type output and requires a pull-up resistor, R_L , to V_{CC} for proper operation. At "low" state, it can sink up to 5mA of load current.

Lock Detect Complement, (Pin 5): The output at pin 5 is the logic complement of the lock detect output at pin 6. This output is also an open collector type stage which can sink 5mA of load current at low or "on" state.

FSK Data Output (Pin 7): This output is an open collector logic stage which requires a pull-up resistor, R_L , to V_{CC} for proper operation. It can sink 5mA of load current. When decoding FSK signals, FSK data output is at "high" or "off" state for low input frequency, and at "low" or "on" state for high input frequency. If no input signal is present, the logic state at pin 7 is indeterminate.

FSK Comparator Input (Pin 8): This is the high impedance input to the FSK voltage comparator. Normally, an FSK post-detection or data filter is connected between this terminal and the PLL phase detector output (pin 11). This data filter is formed by R_F and C_F (see *Figure 3*.) The threshold voltage of the comparator is set by the internal reference voltage, V_{REF} , available at pin 10.

Reference Voltage, V_{REF} (Pin 10): This pin is internally biased at the reference voltage level, V_{REF} : $V_{REF} = V_{CC}/2 - 650mV$. The DC voltage level at this pin forms an internal reference for the voltage levels at pins 5, 8, 11 and 12. Pin

10 must be bypassed to ground with a 0.1μF capacitor for proper operation of the circuit.

Loop Phase Detector Output (Pin 11): This terminal provides a high impedance output for the loop phase detector. The PLL loop filter is formed by R_1 and C_1 connected to pin 11 (see *Figure 3*.) With no input signal, or with no phase error within the PLL, the DC level at pin 11 is very nearly equal to V_{REF} . The peak to peak voltage swing available at the phase detector output is equal to $2 \times V_{REF}$.

VCO Control Input (Pin 12): VCO free-running frequency is determined by external timing resistor, R_0 , connected from this terminal to ground. The VCO free-running frequency, f_0 , is:

$$f_0 = \frac{1}{R_0 \cdot C_0} \text{ Hz}$$

where C_0 is the timing capacitor across pins 13 and 14. For optimum temperature stability, R_0 must be in the range of 10KΩ to 100KΩ (see *Figure 9*.)

This terminal is a low impedance point, and is internally biased at a DC level equal to V_{REF} . The maximum timing current drawn from pin 12 must be limited to $\leq 3mA$ for proper operation of the circuit.

VCO Timing Capacitor (Pins 13 and 14): VCO frequency is inversely proportional to the external timing capacitor, C_0 , connected across these terminals (see *Figure 6*.) C_0 must be non-polar, and in the range of 200pF to 10μF.

VCO Frequency Adjustment: VCO can be fine-tuned by connecting a potentiometer, R_X , in series with R_0 at pin 12 (see *Figure 10*.)

VCO Free-Running Frequency, f_0 : XR-2211 does not have a separate VCO output terminal. Instead, the VCO outputs are internally connected to the phase detector sections of the circuit. For set-up or adjustment purposes, the VCO free-running frequency can be tuned by using the generalized circuit in *Figure 3*, and applying an alternating bit pattern of 0's and 1's at the known mark and space frequencies. By adjusting R_0 , the VCO can then be tuned to obtain a 50% duty cycle on the FSK output (pin 7). This will ensure that the VCO f_0 value is accurately referenced to the mark and space frequencies.

DC ELECTRICAL CHARACTERISTICS (CONT'D)

Test Conditions: $V_{CC} = 12V$, $T_A = +25^\circ C$, $R_O = 30K\Omega$, $C_O = 0.033\mu F$, unless otherwise specified.

Parameter	Min.	Typ.	Max.	Unit	Conditions
Voltage Comparator Section					
Input Impedance		2		M Ω	Measured at Pins 3 and 8
Input Bias Current		100		nA	
Voltage Gain	55	70		dB	$R_I = 5.1K\Omega$
Output Voltage Low		300	500	mV	$I_C = 3mA$
Output Leakage Current		0.01	10	μA	$V_O = 20V$
Internal Reference					
Voltage Level	4.9	5.3	5.7	V	Measured at Pin 10
Output Impedance		100		Ω	AC Small Signal
Maximum Source Current		80		μA	

Notes
 Parameters are guaranteed over the recommended operating conditions, but are not 100% tested in production.
Bold face parameters are covered by production test and guaranteed over operating temperature range.

Specifications are subject to change without notice

ABSOLUTE MAXIMUM RATINGS

Power Supply	20V	Package Power Dissipation Ratings	
Input Signal Level	3V rms	CDIP	750mW
Power Dissipation	900mW	Derate Above $T_A = 25^\circ C$	8mW/ $^\circ C$
		PDIP	800mW
		Derate Above $T_A = 25^\circ C$	60mW/ $^\circ C$
		SOIC	390mW
		Derate Above $T_A = 25^\circ C$	5mW/ $^\circ C$

SYSTEM DESCRIPTION

The main PLL within the XR-2211 is constructed from an input preamplifier, analog multiplier used as a phase detector and a precision voltage controlled oscillator (VCO). The preamplifier is used as a limiter such that input signals above typically 10mV rms are amplified to a constant high level signal. The multiplying-type phase detector acts as a digital exclusive or gate. Its output (unfiltered) produces sum and difference frequencies of the input and the VCO output. The VCO is actually a current controlled oscillator with its normal input current (f_O) set by a resistor (R_O) to ground and its driving current with a resistor (R_1) from the phase detector.

The output of the phase detector produces sum and difference of the input and the VCO frequencies

(internally connected). When in lock, these frequencies are $f_{IN} + f_{VCO}$ (2 times f_{IN} when in lock) and $f_{IN} - f_{VCO}$ (0Hz when lock). By adding a capacitor to the phase detector output, the 2 times f_{IN} component is reduced, leaving a DC voltage that represents the phase difference between the two frequencies. This closes the loop and allows the VCO to track the input frequency.

The FSK comparator is used to determine if the VCO is driven above or below the center frequency (FSK comparator). This will produce both active high and active low outputs to indicate when the main PLL is in lock (quadrature phase detector and lock detector comparator).

ELECTRICAL CHARACTERISTICS

Test Conditions: $V_{CC} = 12V$, $T_A = +25^{\circ}C$, $R_0 = 30K\Omega$, $C_0 = 0.033\mu F$, unless otherwise specified.

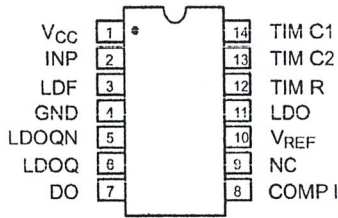
Parameter	Min.	Typ.	Max.	Unit	Conditions
General					
Supply Voltage	4.5		20	V	
Supply Current		4	7	mA	$R_0 \geq 10K\Omega$. See Figure 4.
Oscillator Section					
Frequency Accuracy		± 1	± 3	%	Deviation from $f_0 = 1/R_0 C_0$
Frequency Stability					
Temperature		± 20	± 50	ppm/ $^{\circ}C$	See Figure 8.
Power Supply		0.05	0.5	%/V	$V_{CC} = 12 \pm 1V$. See Figure 7.
		0.2		%/V	$V_{CC} = \pm 5V$. See Figure 7.
Upper Frequency Limit	100	300		KHz	$R_0 = 8.2K\Omega$, $C_0 = 400pF$
Lowest Practical Operating Frequency			0.01	Hz	$R_0 = 2M\Omega$, $C_0 = 50\mu F$
Timing Resistor, R_0 - See Figure 5					
Operating Range	5		2000	K Ω	
Recommended Range	5			K Ω	See Figure 7 and Figure 8.
Loop Phase Detector Section					
Peak Output Current	± 150	± 200	± 300	μA	Measured at Pin 11
Output Offset Current		1		μA	
Output Impedance		1		M Ω	
Maximum Swing	± 4	± 5		V	Referenced to Pin 10
Quadrature Phase Detector					
					Measured at Pin 3
Peak Output Current	100	300		μA	
Output Impedance		1		M Ω	
Maximum Swing		11		V _{PP}	
Input Preempt Section					
					Measured at Pin 2
Input Impedance		20		K Ω	
Input Signal					
Voltage Required to Cause Limiting		2	10	mV rms	

Notes

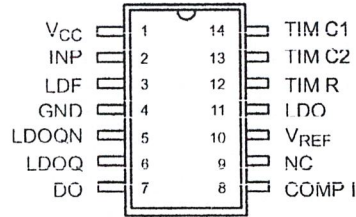
Parameters are guaranteed over the recommended operating conditions, but are not 100% tested in production.

Bold face parameters are covered by production test and guaranteed over operating temperature range.

PIN CONFIGURATION



14 Lead CDIP, PDIP (0.300")



14 Lead SOIC (Jedec, 0.150")

PIN DESCRIPTION

Pin #	Symbol	Type	Description
1	V _{CC}		Positive Power Supply.
2	INP	I	Receive Analog Input.
3	LDF	O	Lock Detect Filter.
4	GND		Ground Pin.
5	LDOQN	O	Lock Detect Output Not. This output will be low if the VCO is in the capture range.
6	LDOQ	O	Lock Detect Output. This output will be high if the VCO is in the capture range.
7	DO	O	Data Output. Decoded FSK output.
8	COMP I	I	FSK Comparator Input.
9	NC		Not Connected.
10	V _{REF}	O	Internal Voltage Reference. The value of V _{REF} is V _{CC} /2 - 650mV.
11	LDO	O	Loop Detect Output. This output provides the result of the quadrature phase detection.
12	TIM R	I	Timing Resistor Input. This pin connects to the timing resistor of the VCO.
13	TIM C2	I	Timing Capacitor Input. The timing capacitor connects between this pin and pin 14.
14	TIM C1	I	Timing Capacitor Input. The timing capacitor connects between this pin and pin 13.

BLOCK DIAGRAM

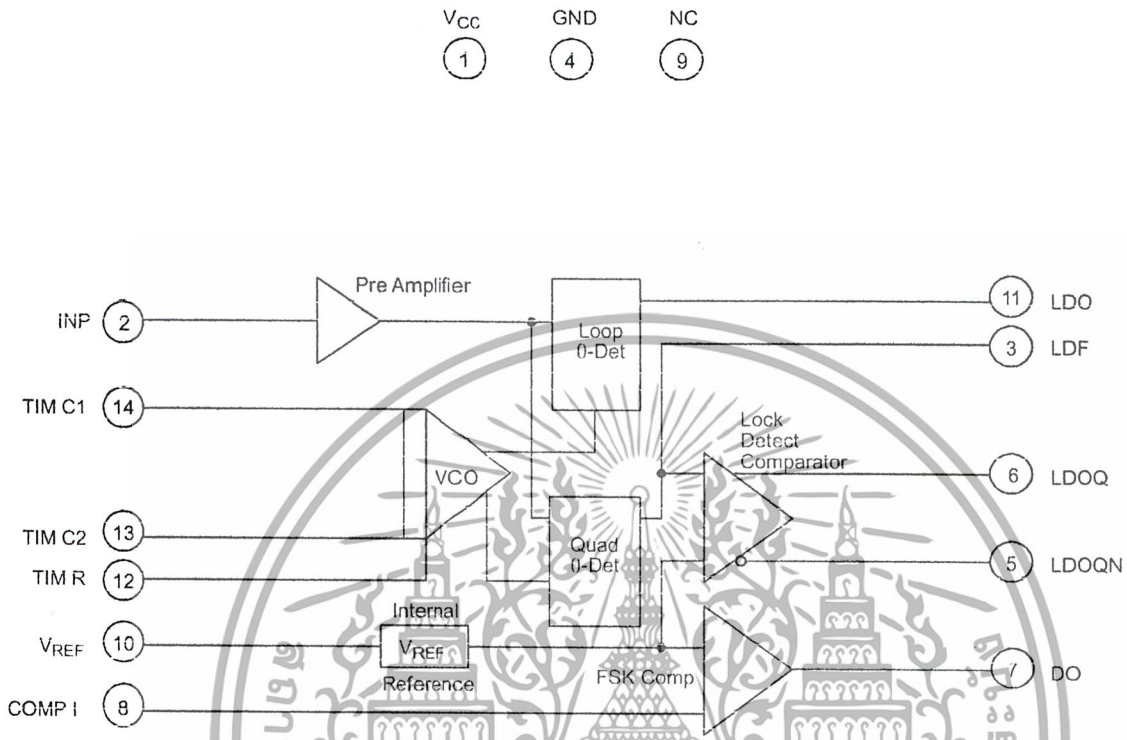


Figure 1. XR-2211 Block Diagram

FEATURES

- Wide Frequency Range, 0.01Hz to 300kHz
- Wide Supply Voltage Range, 4.5V to 20V
- HCMOS/TTL/Logic Compatibility
- FSK Demodulation, with Carrier Detection
- Wide Dynamic Range, 10mV to 3V rms
- Adjustable Tracking Range, $\pm 1\%$ to 80%
- Excellent Temp. Stability, $\pm 50\text{ppm}/^\circ\text{C}$, max.

APPLICATIONS

- Caller Identification Delivery
- FSK Demodulation
- Data Synchronization
- Tone Decoding
- FM Detection
- Carrier Detection

GENERAL DESCRIPTION

The XR-2211 is a monolithic phase-locked loop (PLL) system especially designed for data communications applications. It is particularly suited for FSK modem applications. It operates over a wide supply voltage range of 4.5 to 20V and a wide frequency range of 0.01Hz to 300kHz. It can accommodate analog signals between 10mV and 3V, and can interface with conventional DTL, TTL, and ECL logic families. The circuit consists of a basic PLL for tracking an input signal within the pass band, a

quadrature phase detector which provides carrier detection, and an FSK voltage comparator which provides FSK demodulation. External components are used to independently set center frequency, bandwidth, and output delay. An internal voltage reference proportional to the power supply is provided at an output pin.

The XR-2211 is available in 14 pin packages specified for military and industrial temperature ranges.

ORDERING INFORMATION

Part No.	Package	Operating Temperature Range
XR-2211M	14 Pin CDIP (0.300")	-55°C to +125°C
XR-2211N	14 Pin CDIP (0.300")	-40°C to +85°C
XR-2211P	14 Pin PDIP (0.300")	-40°C to +85°C
XR-2211ID	14 Lead SOIC (Jedec, 0.150")	-40°C to +85°C



NOTICE

EXAR Corporation reserves the right to make changes to the products contained in this publication in order to improve design, performance or reliability. EXAR Corporation assumes no responsibility for the use of any circuits described herein, conveys no license under any patent or other right, and makes no representation that the circuits are free of patent infringement. Charts and schedules contained here in are only for illustration purposes and may vary depending upon a user's specific application. While the information in this publication has been carefully checked; no responsibility, however, is assumed for inaccuracies.

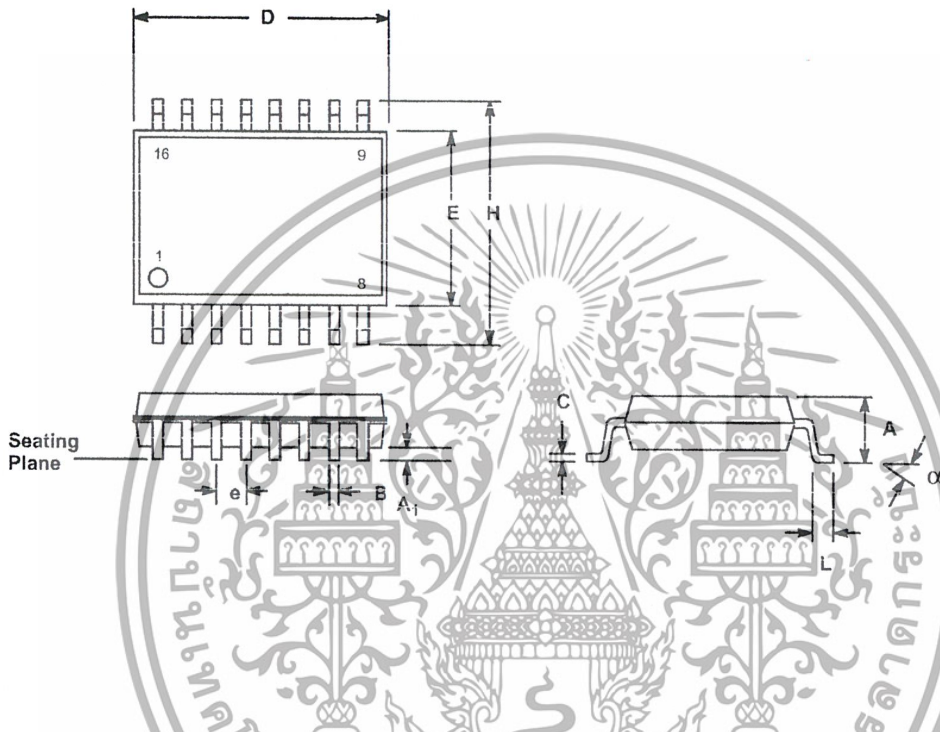
EXAR Corporation does not recommend the use of any of its products in life support applications where the failure or malfunction of the product can reasonably be expected to cause failure of the life support system or to significantly affect its safety or effectiveness. Products are not authorized for use in such applications unless EXAR Corporation receives, in writing, assurances to its satisfaction that: (a) the risk of injury or damage has been minimized; (b) the user assumes all such risks; (c) potential liability of EXAR Corporation is adequately protected under the circumstances.

Copyright 1972 EXAR Corporation
Datasheet June 1997

Reproduction, in part or whole, without the prior written consent of EXAR Corporation is prohibited.

16 LEAD SMALL OUTLINE
(300 MIL JEDEC SOIC)

Rev. 1.00

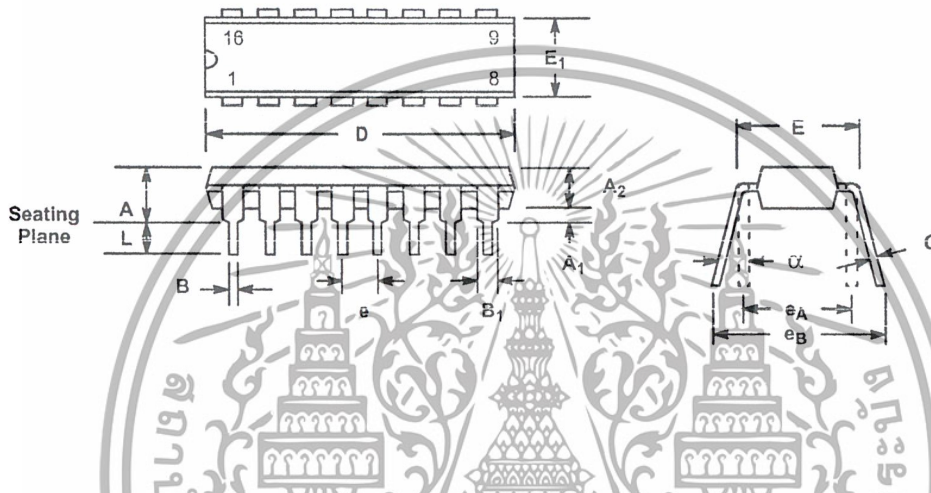


SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.093	0.104	2.35	2.65
A ₁	0.004	0.012	0.10	0.30
B	0.013	0.020	0.33	0.51
C	0.009	0.013	0.23	0.32
D	0.398	0.413	10.10	10.50
E	0.291	0.299	7.40	7.60
e	0.050 BSC		1.27 BSC	
H	0.394	0.419	10.00	10.65
L	0.016	0.050	0.40	1.27
α	0°	8°	0°	8°

Note: The control dimension is the millimeter column

16 LEAD PLASTIC DUAL-IN-LINE
(300 MIL PDIP)

Rev. 1.00

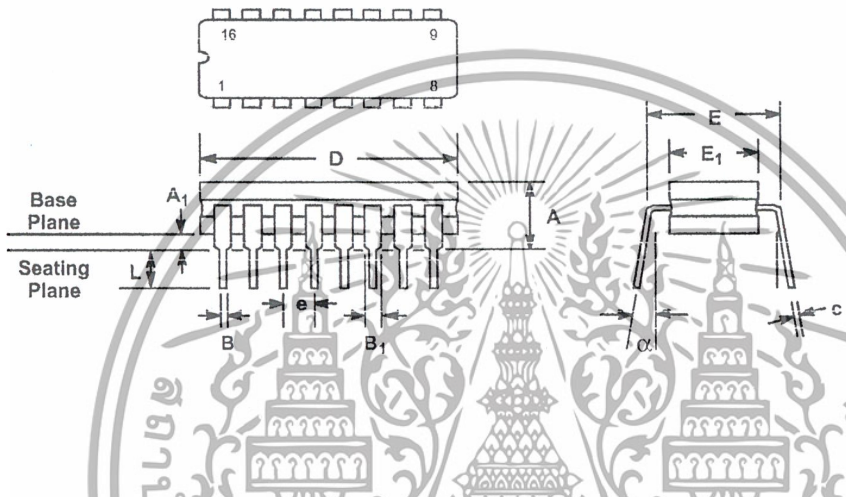


SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.145	0.210	3.68	5.33
A ₁	0.015	0.070	0.38	1.78
A ₂	0.115	0.195	2.92	4.95
B	0.014	0.024	0.36	0.56
B ₁	0.030	0.070	0.76	1.78
C	0.008	0.014	0.20	0.38
D	0.745	0.840	18.92	21.34
E	0.300	0.325	7.62	8.26
E ₁	0.240	0.280	6.10	7.11
e	0.100 BSC		2.54 BSC	
e _A	0.300 BSC		7.62 BSC	
e _B	0.310	0.430	7.87	10.92
L	0.115	0.160	2.92	4.06
α	0°	15°	0°	15°

Note: The control dimension is the inch column

16 LEAD CERAMIC DUAL-IN-LINE
(300 MIL CDIP)

Rev. 1.00



SYMBOL	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.100	0.200	2.54	5.08
A ₁	0.015	0.060	0.38	1.52
B	0.014	0.026	0.36	0.66
B ₁	0.045	0.065	1.14	1.65
c	0.008	0.018	0.20	0.46
D	0.740	0.840	18.80	21.34
E ₁	0.250	0.310	6.35	7.87
E	0.300 BSC		7.62 BSC	
e	0.100 BSC		2.54 BSC	
L	0.125	0.200	3.18	5.08
α	0°	15°	0°	15°

Note: The control dimension is the inch column

Output Amplitude:

Maximum output amplitude is inversely proportional to the external resistor, R_3 , connected to Pin 3 (see Figure 3). For sine wave output, amplitude is approximately 60mV peak per k Ω of R_3 ; for triangle, the peak amplitude is approximately 160mV peak per k Ω of R_3 . Thus, for example, $R_3 = 50\text{k}\Omega$ would produce approximately 13V sinusoidal output amplitude.

Amplitude Modulation:

Output amplitude can be modulated by applying a dc bias and a modulating signal to Pin 1. The internal impedance

at Pin 1 is approximately 100k Ω . Output amplitude varies linearly with the applied voltage at Pin 1, for values of dc bias at this pin, within 14 volts of $V_{CC}/2$ as shown in Figure 6. As this bias level approaches $V_{CC}/2$, the phase of the output signal is reversed, and the amplitude goes through zero. This property is suitable for phase-shift keying and suppressed-carrier AM generation. Total dynamic range of amplitude modulation is approximately 55dB.

CAUTION: AM control must be used in conjunction with a well-regulated supply, since the output amplitude now becomes a function of V_{CC} .

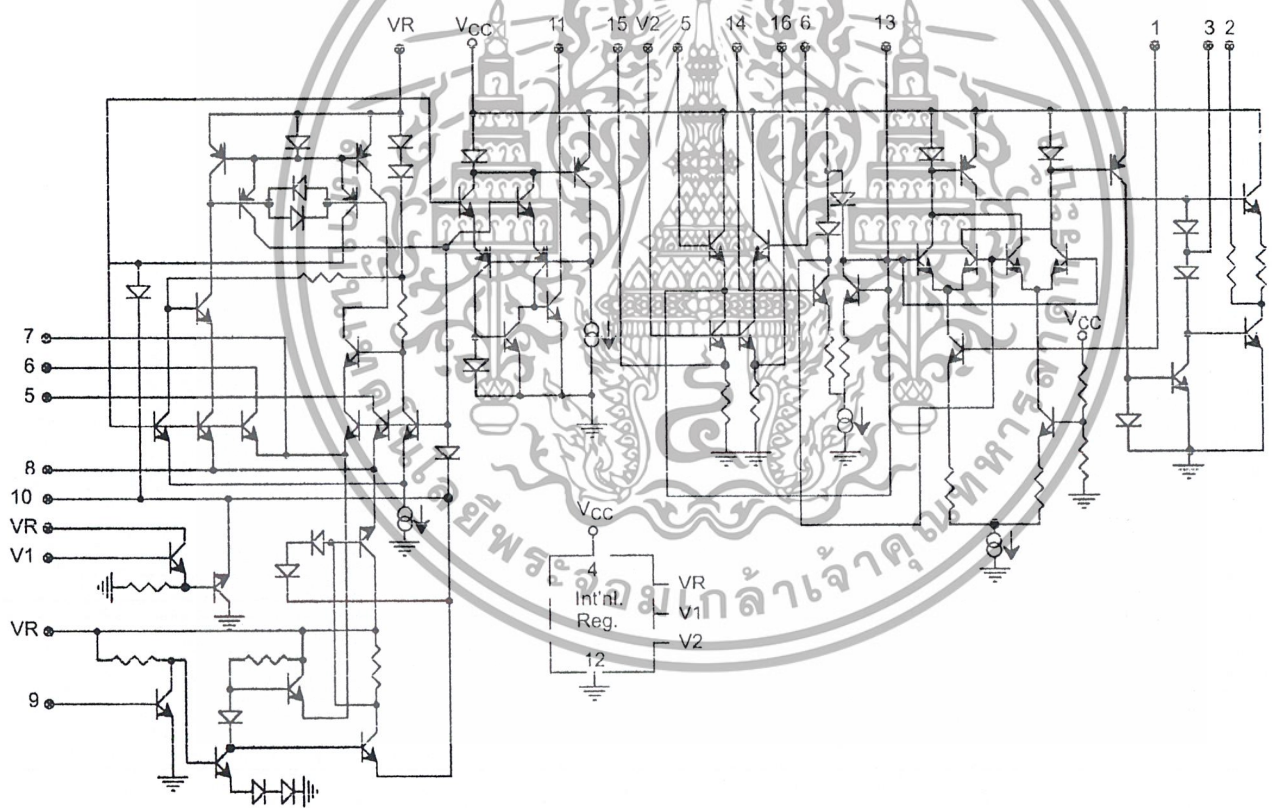


Figure 15. Equivalent Schematic Diagram

With External Adjustment:

The harmonic content of sinusoidal output can be reduced to -0.5% by additional adjustments as shown in *Figure 12*. The potentiometer, R_A , adjusts the sine-shaping resistor, and R_B provides the fine adjustment for the waveform symmetry. The adjustment procedure is as follows:

1. Set R_B at midpoint and adjust R_A for minimum distortion.
2. With R_A set as above, adjust R_B to further reduce distortion.

Triangle Wave Generation

The circuits of *Figure 11* and *Figure 12* can be converted to triangle wave generation, by simply open-circuiting Pin 13 and 14 (i.e., S_1 open). Amplitude of the triangle is approximately twice the sine wave output.

FSK Generation

Figure 13 shows the circuit connection for sinusoidal FSK signal operation. Mark and space frequencies can be independently adjusted by the choice of timing resistors, R_1 and R_2 : the output is phase-continuous during transitions. The keying signal is applied to Pin 9. The circuit can be converted to split-supply operation by simply replacing ground with V^- .

Pulse and Ramp Generation

Figure 14 shows the circuit for pulse and ramp waveform generation. In this mode of operation, the FSK keying terminal (Pin 9) is shorted to the square-wave output (Pin 11), and the circuit automatically frequency-shift keys itself between two separate frequencies during the positive-going and negative-going output waveforms. The pulse width and duty cycle can be adjusted from 1% to 99% by the choice of R_1 and R_2 . The values of R_1 and R_2 should be in the range of $1k\Omega$ to $2M\Omega$.

PRINCIPLES OF OPERATION**Description of Controls****Frequency of Operation:**

The frequency of oscillation, f_o , is determined by the external timing capacitor, C , across Pin 5 and 6, and by the timing resistor, R , connected to either Pin 7 or 8. The frequency is given as:

$$f_o = \frac{1}{RC} \text{ Hz}$$

and can be adjusted by varying either R or C . The recommended values of R , for a given frequency range, as shown in *Figure 5*. Temperature stability is optimum for $4k\Omega \leq R < 200k\Omega$. Recommended values of C are from $1000pF$ to $100\mu F$.

Frequency Sweep and Modulation:

Frequency of oscillation is proportional to the total timing current, I_T , drawn from Pin 7 or 8:

$$f = \frac{320 I_T (\text{mA})}{C (\mu F)} \text{ Hz}$$

Timing terminals (Pin 7 or 8) are low-impedance points, and are internally biased at $+3V$, with respect to Pin 12. Frequency varies linearly with I_T , over a wide range of current values, from $1\mu A$ to $3mA$. The frequency can be controlled by applying a control voltage, V_C , to the activated timing pin as shown in *Figure 10*. The frequency of oscillation is related to V_C as:

$$f = \frac{1}{RC} \left(1 + \frac{R}{R_c} \left(1 - \frac{V_C}{3} \right) \right) \text{ Hz}$$

where V_C is in volts. The voltage-to-frequency conversion gain, K , is given as:

$$K = \partial f / \partial V_C = - \frac{0.32}{R_c C} \text{ Hz/V}$$

CAUTION: For safety operation of the circuit, I_T should be limited to $\leq 3mA$.

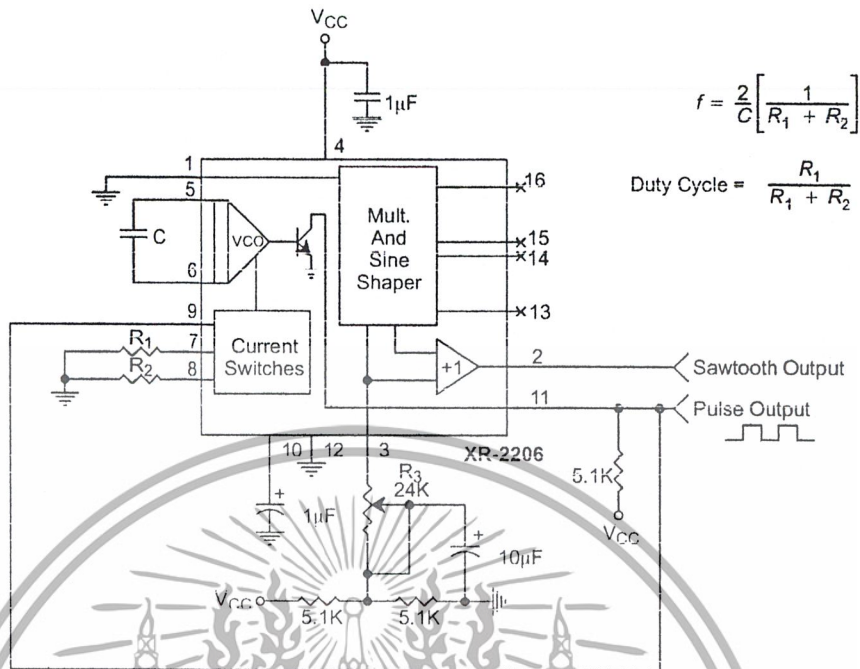


Figure 14. Circuit for Pulse and Ramp Generation.

APPLICATIONS INFORMATION

Sine Wave Generation

Without External Adjustment

Figure 11 shows the circuit connection for generating a sinusoidal output from the XR-2206. The potentiometer, R_1 at Pin 7, provides the desired frequency tuning. The maximum output swing is greater than $V^+/2$, and the typical distortion (THD) is $< 2.5\%$. If lower sine wave distortion is desired, additional adjustments can be provided as described in the following section.

The circuit of Figure 11 can be converted to split-supply operation, simply by replacing all ground connections with V^- . For split-supply operation, R_3 can be directly connected to ground.

Frequency-Shift Keying

The XR-2206 can be operated with two separate timing resistors, R_1 and R_2 , connected to the timing Pin 7 and 8, respectively, as shown in Figure 13. Depending on the polarity of the logic signal at Pin 9, either one or the other of these timing resistors is activated. If Pin 9 is open-circuited or connected to a bias voltage $\geq 2V$, only R_1 is activated. Similarly, if the voltage level at Pin 9 is $\leq 1V$, only R_2 is activated. Thus, the output frequency can be keyed between two levels, f_1 and f_2 , as:

$$f_1 = 1/R_1C \text{ and } f_2 = 1/R_2C$$

For split-supply operation, the keying voltage at Pin 9 is referenced to V^- .

Output DC Level Control

The dc level at the output (Pin 2) is approximately the same as the dc bias at Pin 3. In Figure 11, Figure 12 and Figure 13, Pin 3 is biased midway between V^+ and ground, to give an output dc level of $\approx V^+/2$.

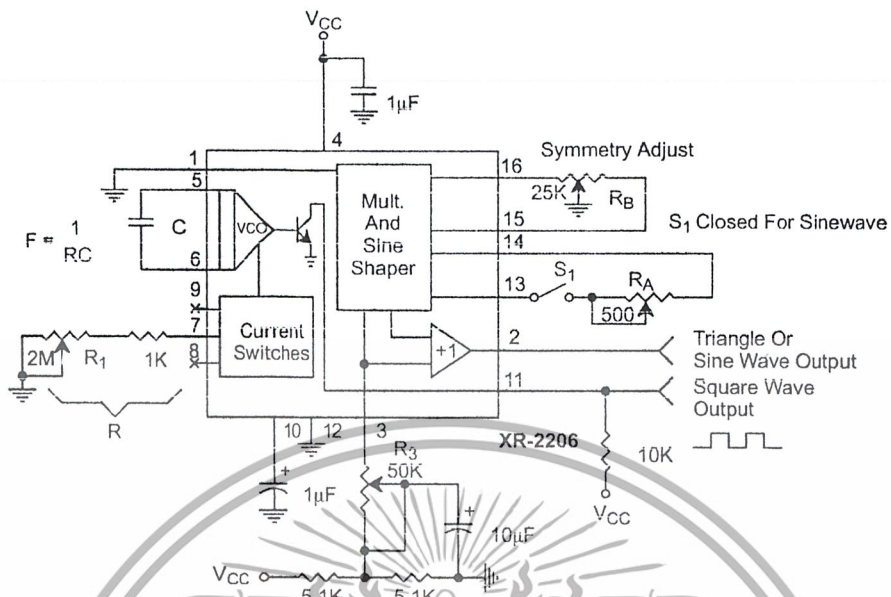


Figure 12. Circuit for Sine Wave Generation with Minimum Harmonic Distortion. (R_3 Determines Output Swing - See Figure 3)

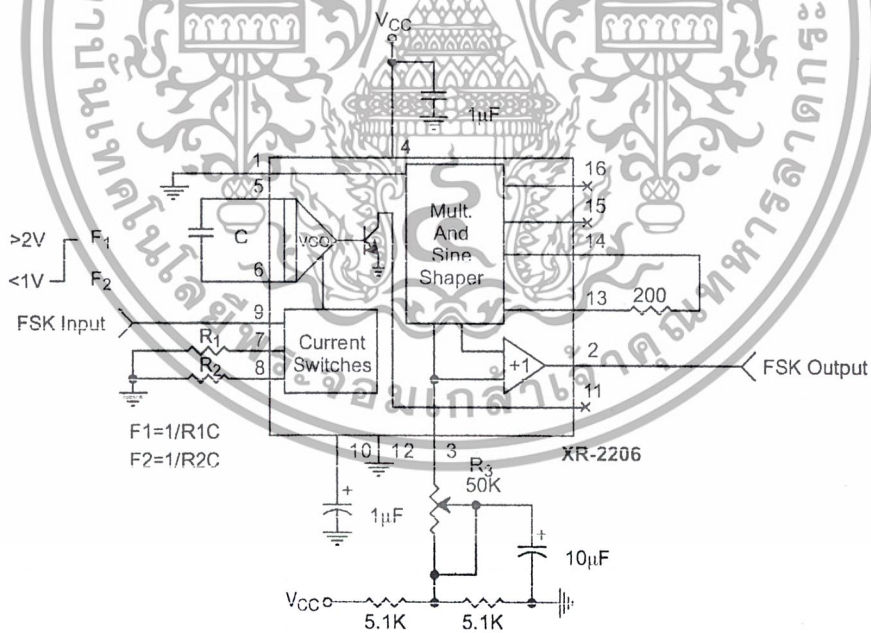


Figure 13. Sinusoidal FSK Generator

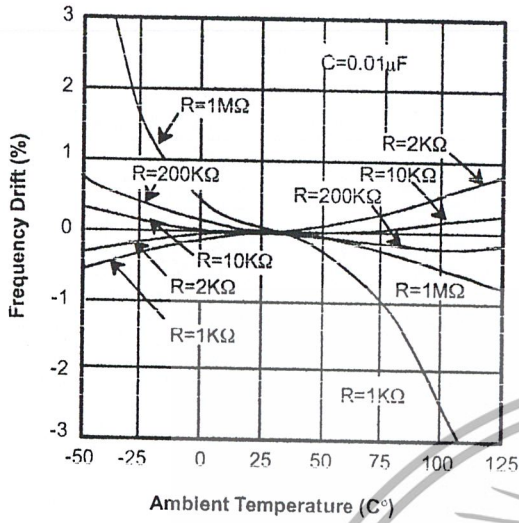


Figure 9. Frequency Drift versus Temperature.

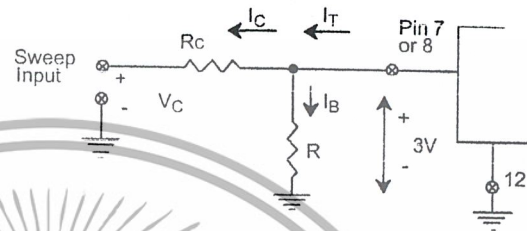


Figure 10. Circuit Connection for Frequency Sweep.

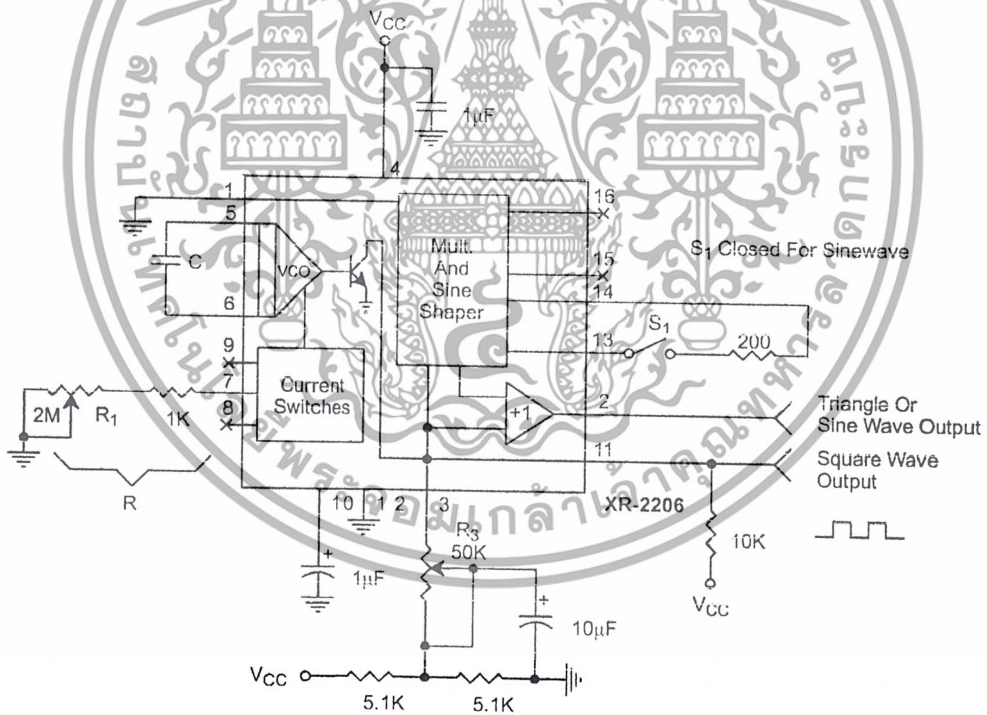


Figure 11. Circuit for Sine Wave Generation without External Adjustment.
(See Figure 3 for Choice of R₃)

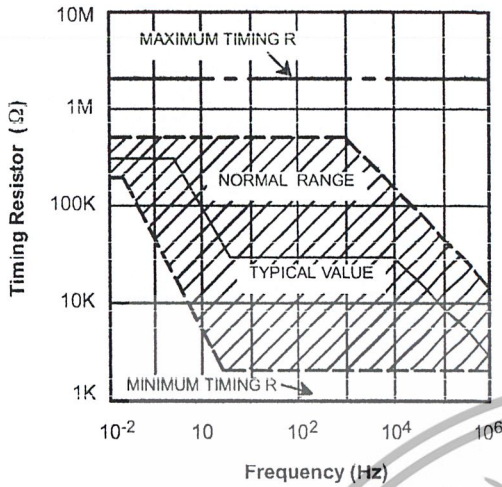


Figure 5. R versus Oscillation Frequency.

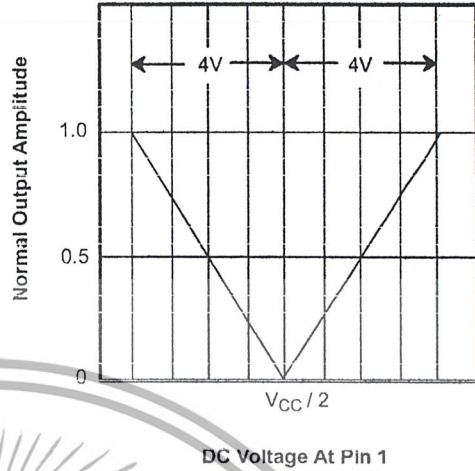


Figure 6. Normalized Output Amplitude versus DC Bias at AM Input (Pin 1)

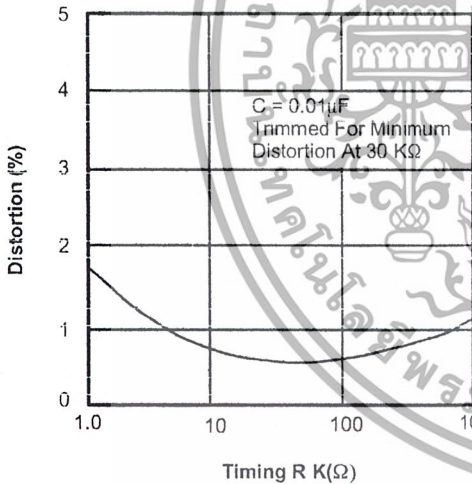


Figure 7. Trimmed Distortion versus Timing Resistor.

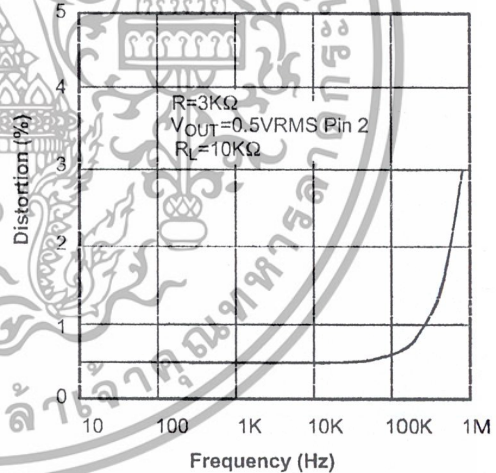


Figure 8. Sine Wave Distortion versus Operating Frequency with Timing Capacitors Varied.

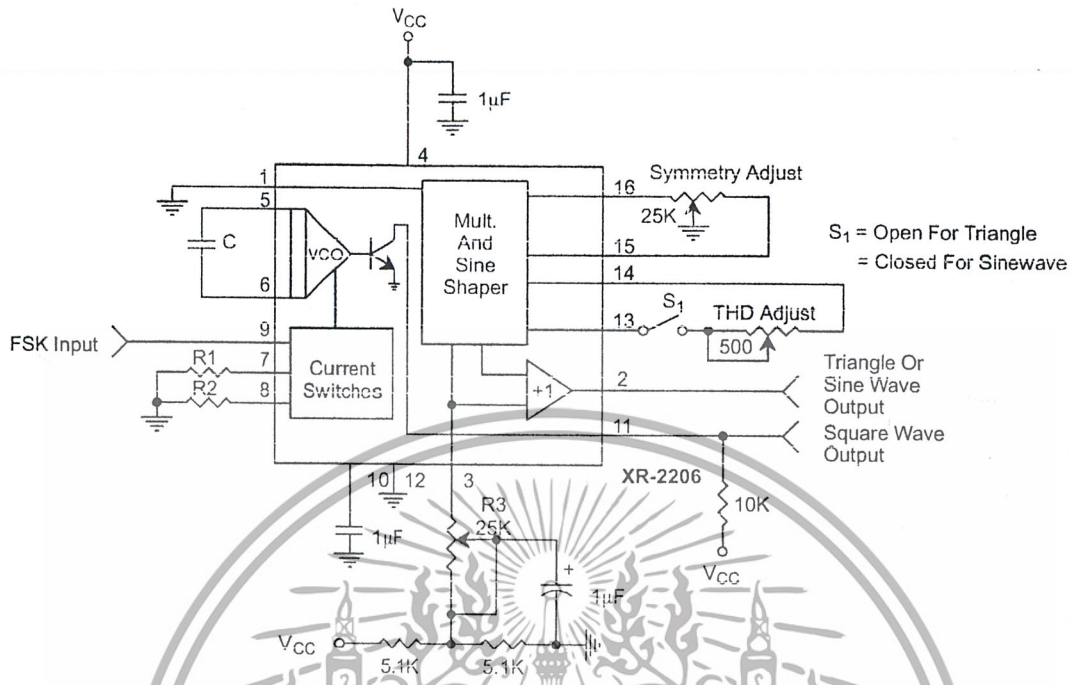


Figure 2. Basic Test Circuit

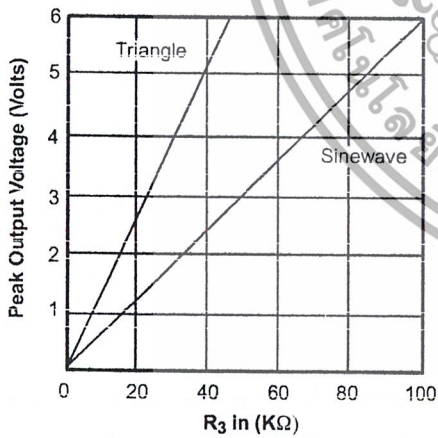


Figure 3. Output Amplitude as a Function of the Resistor, R3, at Pin 3

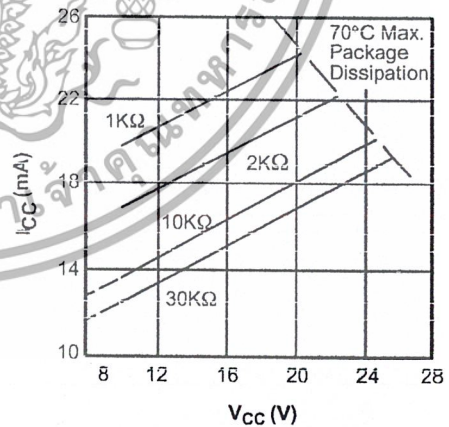


Figure 4. Supply Current vs Supply Voltage, Timing, R

DC ELECTRICAL CHARACTERISTICS (CONT'D)

Parameters	XR-2206M/F			XR-2206CP/D			Units	Conditions
	Min.	Typ.	Max.	Min.	Typ.	Max.		
Amplitude Modulation								
Input Impedance	50	100		50	100		k Ω	
Modulation Range		100			100		%	
Carrier Suppression		55			55		dB	
Linearity		2			2		%	For 95% modulation
Square-Wave Output								
Amplitude		12			12		Vp-p	Measured at Pin 11.
Rise Time		250			250		ns	$C_L = 10\text{pF}$
Fall Time		50			50		ns	$C_L = 10\text{pF}$
Saturation Voltage		0.2	0.4		0.2	0.6	V	$I_L = 2\text{mA}$
Leakage Current		0.1	20		0.1	100	μA	$V_{CC} = 26\text{V}$
FSK Keying Level (Pin 9)	0.8	1.4	2.4	0.8	1.4	2.4	V	See section on circuit controls
Reference Bypass Voltage	2.9	3.1	3.3	2.5	3	3.5	V	Measured at Pin 10.

Notes

¹ Output amplitude is directly proportional to the resistance, R_3 , on Pin 3. See Figure 3.

² For maximum amplitude stability, R_3 should be a positive temperature coefficient resistor.

Bold face parameters are covered by production test and guaranteed over operating temperature range.

Specifications are subject to change without notice

ABSOLUTE MAXIMUM RATINGS

Power Supply	26V	Total Timing Current	6mA
Power Dissipation	750mW	Storage Temperature	-65°C to +150°C
Derate Above 25°C	5mW/°C		

SYSTEM DESCRIPTION

The XR-2206 is comprised of four functional blocks; a voltage-controlled oscillator (VCO), an analog multiplier and sine-shaper; a unity gain buffer amplifier; and a set of current switches.

The VCO produces an output frequency proportional to an input current, which is set by a resistor from the timing

terminals to ground. With two timing pins, two discrete output frequencies can be independently produced for FSK generation applications by using the FSK input control pin. This input controls the current switches which select one of the timing resistor currents, and routes it to the VCO.

DC ELECTRICAL CHARACTERISTICS

Test Conditions: Test Circuit of *Figure 2* $V_{CC} = 12V$, $T_A = 25^\circ C$, $C = 0.01\mu F$, $R_1 = 100k\Omega$, $R_2 = 10k\Omega$, $R_3 = 25k\Omega$
 Unless Otherwise Specified. S_1 open for triangle, closed for sine wave.

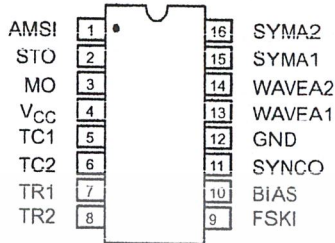
Parameters	XR-2206M/P			XR-2206CP/D			Units	Conditions
	Min.	Typ.	Max.	Min.	Typ.	Max.		
General Characteristics								
Single Supply Voltage	10		26	10		26	V	
Split-Supply Voltage	± 5		± 13	± 5		± 13	V	
Supply Current		12	17		14	20	mA	$R_1 \geq 10k\Omega$
Oscillator Section								
Max. Operating Frequency	0.5	1		0.5	1		MHz	$C = 1000pF$, $R_1 = 1k\Omega$
Lowest Practical Frequency		0.01			0.01		Hz	$C = 50\mu F$, $R_1 = 2M\Omega$
Frequency Accuracy		± 1	± 4		± 2		% of f_0	$f_0 = 1/R_1C$
Temperature Stability Frequency		± 10	± 50		± 20		ppm/ $^\circ C$	$0^\circ C \leq T_A \leq 70^\circ C$ $R_1 = R_2 = 20k\Omega$
Sine Wave Amplitude Stability ²		4800			4800		ppm/ $^\circ C$	
Supply Sensitivity		0.01	0.1		0.01		%/V	$V_{LOW} = 10V$, $V_{HIGH} = 20V$, $R_1 = R_2 = 20k\Omega$
Sweep Range	1000:1	2000:1			2000:1		$f_H = f_L$	$f_H @ R_1 = 1k\Omega$ $f_L @ R_1 = 2M\Omega$
Sweep Linearity								
10:1 Sweep		2			2		%	$f_L = 1kHz$, $f_H = 10kHz$
1000:1 Sweep		8			8		%	$f_L = 100Hz$, $f_H = 100kHz$
FM Distortion		0.1			0.1		%	$\pm 10\%$ Deviation
Recommended Timing Components								
Timing Capacitor: C	0.001		100	0.001		100	μF	<i>Figure 5</i>
Timing Resistors: R_1 & R_2	1		2000	1		2000	k Ω	
Triangle Sine Wave Output¹								
Triangle Amplitude		160			160		mV/k Ω	<i>Figure 2</i> , S_1 Open
Sine Wave Amplitude	40	60	80		60		mV/k Ω	<i>Figure 2</i> , S_1 Closed
Max. Output Swing		6			6		V _{p-p}	
Output Impedance		600			600		Ω	
Triangle Linearity		1			1		%	
Amplitude Stability		0.5			0.5		dB	For 1000:1 Sweep
Sine Wave Distortion								
Without Adjustment		2.5			2.5		%	$R_1 = 30k\Omega$
With Adjustment		0.4	1.0		0.5	1.5	%	See <i>Figure 7</i> and <i>Figure 8</i>

Notes

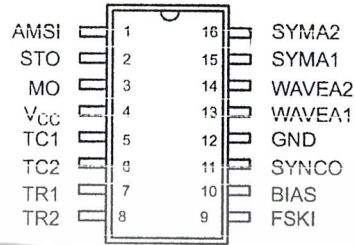
¹ Output amplitude is directly proportional to the resistance, R_3 , on Pin 3. See *Figure 3*.

² For maximum amplitude stability, R_3 should be a positive temperature coefficient resistor.

Bold face parameters are covered by production test and guaranteed over operating temperature range.



16 Lead PDIP, CDIP (0.300")



16 Lead SOIC (Jedec, 0.300")

PIN DESCRIPTION

Pin #	Symbol	Type	Description
1	AMSI	I	Amplitude Modulating Signal Input.
2	STO	O	Sine or Triangle Wave Output.
3	MO	O	Multiplier Output.
4	V _{CC}		Positive Power Supply.
5	TC1	I	Timing Capacitor Input.
6	TC2	I	Timing Capacitor Input.
7	TR1	O	Timing Resistor 1 Output.
8	TR2	O	Timing Resistor 2 Output.
9	FSKI	I	Frequency Shift Keying Input.
10	BIAS	O	Internal Voltage Reference.
11	SYNCO	O	Sync Output. This output is a open collector and needs a pull up resistor to V _{CC} .
12	GND		Ground pin.
13	WAVEA1	I	Wave Form Adjust Input 1.
14	WAVEA2	I	Wave Form Adjust Input 2.
15	SYMA1	I	Wave Symetry Adjust 1.
16	SYMA2	I	Wave Symetry Adjust 2.

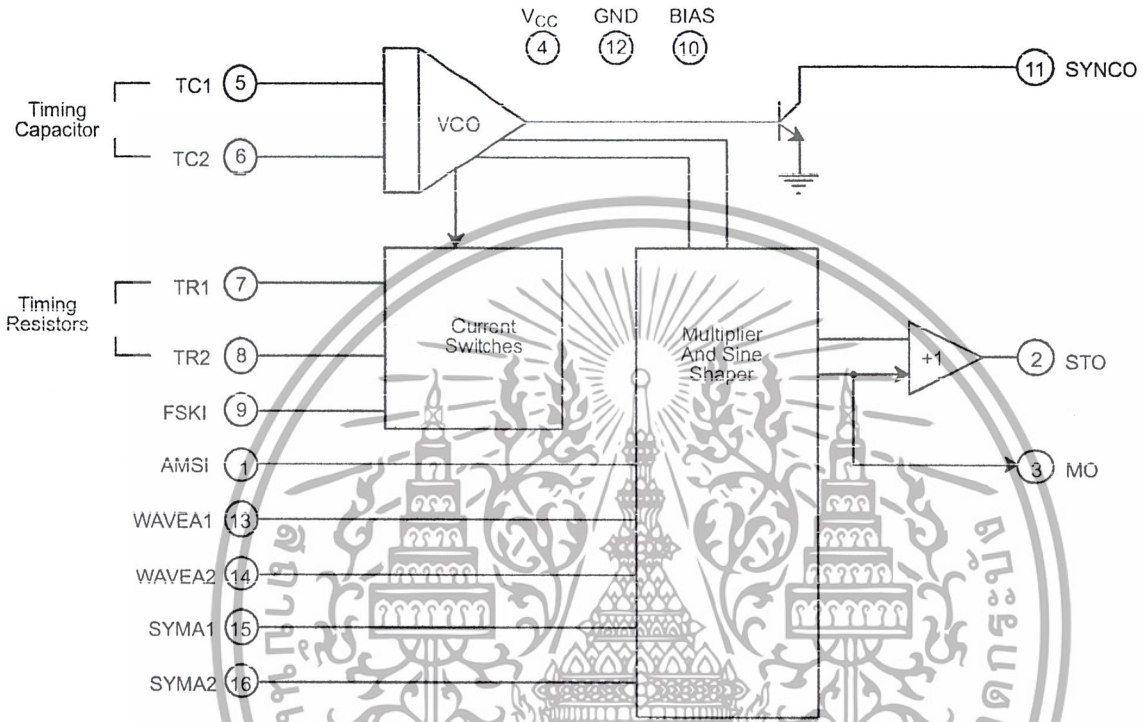


Figure 1. XR-2206 Block Diagram

June 1997-3

FEATURES

- Low-Sine Wave Distortion, 0.5%, Typical
- Excellent Temperature Stability, 20ppm/°C, Typ.
- Wide Sweep Range, 2000:1, Typical
- Low-Supply Sensitivity, 0.01%V, Typ.
- Linear Amplitude Modulation
- TTL Compatible FSK Controls
- Wide Supply Range, 10V to 26V
- Adjustable Duty Cycle, 1% TO 99%

APPLICATIONS

- Waveform Generation
- Sweep Generation
- AM/FM Generation
- V/F Conversion
- FSK Generation
- Phase-Locked Loops (VCO)

GENERAL DESCRIPTION

The XR-2206 is a monolithic function generator integrated circuit capable of producing high quality sine, square, triangle, ramp, and pulse waveforms of high-stability and accuracy. The output waveforms can be both amplitude and frequency modulated by an external voltage. Frequency of operation can be selected externally over a range of 0.01Hz to more than 1MHz.

The circuit is ideally suited for communications, instrumentation, and function generator applications requiring sinusoidal tone, AM, FM, or FSK generation. It has a typical drift specification of 20ppm/°C. The oscillator frequency can be linearly swept over a 2000:1 frequency range with an external control voltage, while maintaining low distortion.

ORDERING INFORMATION

Part No.	Package	Operating Temperature Range
XR-2206M	16 Lead 300 Mil CDIP	-55°C to +125°C
XR-2206P	16 Lead 300 Mil PDIP	-40°C to +85°C
XR-2206CP	16 Lead 300 Mil PDIP	0°C to +70°C
XR-2206D	16 Lead 300 Mil JEDEC SOJC	0°C to +70°C