



การส่งข้อมูลในระบบกระจายเสียงแบบ AM



โดย

นายศุภชัย ประสพโชค
นายชาญอนันต์ แก้วใส
นายบัณฑิต เขตพิทักษ์
นายสมบัติ พุทธา

วัน เดือน ปี..... ๑1 ก.ค. ๒5๕๒
เลขทะเบียน..... ๐๖๒๙๖
เลขเรียกหนังสือ..... T 38๐๘๙1 ๑ ๖๖๖ ก

ปริญญาโทฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ผู้จัดทำ

นายศุภชัย	ประสพโชค	36012072
นายชาญอนันต์	แก้วใส	36013239
นายบัณฑิต	เขตพิทักษ์	36013252
นายสมบัติ	พุทธา	36013265

อาจารย์ที่ปรึกษา อาจารย์ กฤดากร กล่อมการ

ภาควิชาเทคนิคอุตสาหกรรม
ปีการศึกษา 2538

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้
ปฏิญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปฏิญานิพนธ์

- ประธานกรรมการ
- ()
- กรรมการ
- ()
- กรรมการ
- ()
- กรรมการ
- ()
- กรรมการ
- ()

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2538

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง การส่งข้อมูลในระบบกระจายเสียงแบบ AM

ผู้จัดทำ

นายศุภชัย	ประสพโชค	36012072
นายชาญอนันต์	แก้วใส	36013239
นายบัณฑิต	เขตพิทักษ์	36013252
นายสมบัติ	พุกจา	36013265



..... อาจารย์ที่ปรึกษา
(กฤตากร กล่อมการ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	แนวคิดและทฤษฎี	2
	- การมอดูเลทแบบ AM	
	- ระบบการสื่อสารข้อมูล	
	- ไมโครคอนโทรลเลอร์	
	- WALSH FUNCTION	
	- BLOCK DIAGRAM	
บทที่ 3	การออกแบบวงจร	33
	- BAND PASS FILTER	
	- FSK DEMODULATION	
บทที่ 4	วงจรการใช้งานและการทำงานของระบบ	36
	- วงจร ENCODER	
	- วงจร WALSH	
	- วงจร DECODER (FSK DEMODULATION)	
	- วงจร BAND PASS FILTER	
	- วงจรภาครับแบบ AM	
บทที่ 5	ผลการทดลองและสรุปผลการทดลอง	46
	บทสรุป	
	เอกสารอ้างอิง	
	กิตติกรรมประกาศ	
	ภาคผนวก	49
	FLOW CHART	
	-FLOW CHART ภาคส่ง	
	-โปรแกรมภาษา C ภาครับและภาคส่ง	
	-ภาษา ASSEMBLY	
	-FLOW CHART ภาครับ	
	-ภาษา ASSEMBLY	
	-อุปกรณ์ Z-80	
	-IC XR - 2211	

บทคัดย่อ

โครงการนี้เป็นกรปรับปรุงและพัฒนาการออกแบบระบบการส่งข้อมูล ผ่านช่องสัญญาณระบบการกระจายคลื่นวิทยุแบบ AM โดยทำการส่งข้อมูลผ่านคอมพิวเตอร์ไปยังเครื่องส่งข้อมูล หลังจากนั้น ข้อมูลจะถูกส่งออกไปเพื่อทำการมอดูเลท ด้วยอัตราการส่งความเร็วต่ำ กับเครื่องส่งวิทยุกระจายเสียงแบบ AM เพื่อให้สามารถรับข้อมูลได้อย่าง ถูกต้องและแม่นยำ มีความผิดพลาดต่ำ เมื่อเครื่องรับวิทยุรับสัญญาณออกมาแล้ว จะผ่านการแปลงสัญญาณนั้นให้เป็นข้อมูล เพื่อทำการแสดง ผลซึ่งจะแสดงผลออกบนจอแสดงผล หรือแสดงผล ออกทางจอคอมพิวเตอร์

ABSTRACT.

In this project, The design and developed of data transmission over AM broadcasting system. The data message have been transmitted from the computer to AM transmitter with the low bit rate . For receiving with accuracy and low bit error rate . At the receive the data display on LCD or monitor and received by ordinary AM receiver .

1. บทนำ

ในระบบการสื่อสารปัจจุบัน การส่งข่าวสารจะอาศัยหลักการส่งของการแพร่กระจาย คลื่นวิทยุเป็น ส่วนมากและถูกนำไปใช้อย่างกว้างขวาง ในการดำเนินธุรกิจทั่วไปทั้งขนาดใหญ่ และเล็ก รวมถึงชีวิตประจำวัน

ดังนั้นทำให้ความเจริญทางด้านโทรคมนาคม ถูกพัฒนาไปอย่างรวดเร็ว การรับส่งข่าวสารด้วย คลื่นวิทยุ AM ที่มีใช้กันอยู่ทั่วไป ก็เป็นอีกรูปแบบหนึ่ง ที่เราสามารถจะนำมาประยุกต์ใช้งาน เพื่อทำให้เกิด ประโยชน์สูงสุด ดังนั้นเพื่อให้เป็นไปตามวัตถุประสงค์ดังกล่าว โครงการนี้จึงได้ทำการศึกษา รายละเอียด และประกอบกับความรู้ที่ได้ศึกษามาทำการพัฒนาระบบ และอุปกรณ์ ในการส่งข้อมูล ส่วนทางด้านภาครับ จะเป็นการนำเอาข่าวสารที่ถูกส่งมา นำไปแสดงผลออกทางจอ LCD โดยในส่วน นี้ จะถูกควบคุมจากการทำงานของระบบ โปรแกรมของไมโครคอนโทรลเลอร์ กับไมโครคอมพิวเตอร์ ทำให้ข้อมูลที่ถูส่งออกมาสามารถที่จะเข้าใจง่ายและสะดวกต่อการใช้งาน



บทที่ 2 แนวคิดและทฤษฎี

-การมอดูเลทแบบ AM

-ระบบการสื่อสารข้อมูล

-ไมโครคอนโทรลเลอร์

-WALSH FUNCTION

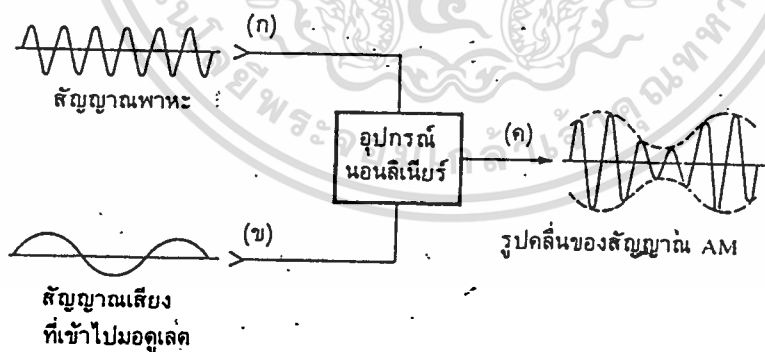
-BLOCK DIAGRAM



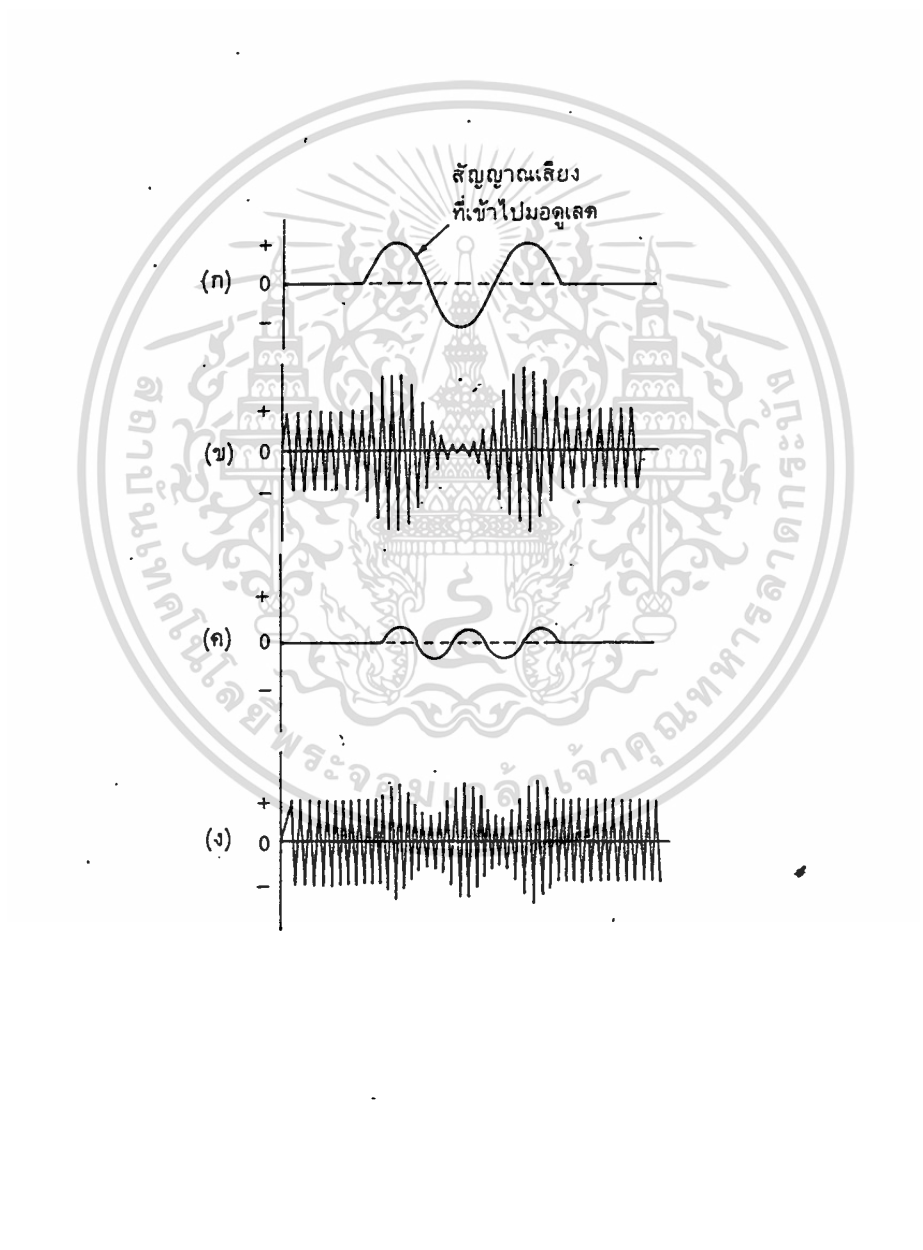
บทที่ 2 แนวคิดและทฤษฎี

การมอดูเลตทางแอมพลิจูด

การมอดูเลตแบบ AM นั้น เราใช้สัญญาณข่าวสาร สมมติว่าให้สัญญาณเสียงมอดูเลตลงบนสัญญาณพาหะเพื่อเปลี่ยนคุณสมบัติทางแอมพลิจูดของพาหะ ในรูปที่ 2.1 เราใช้สัญญาณพาหะ(ก) ผสมกับสัญญาณเสียง(ข)ลงในวงจรอนลิเนียร์ เช่น ใช้ไดโอดหรือ ทรานซิสเตอร์ โดยให้มีจุดทำงานอยู่ในบริเวณที่ไม่เป็นลิเนียร์ ในอุปกรณ์แบบอนลิเนียร์ จะทำให้เกิดสัญญาณ AM ดังรูปที่ 2.1(ค) ขึ้นจะสังเกตว่าสัญญาณพาหะซึ่งถูกมอดูเลตแล้วจะมีแอมพลิจูด เปลี่ยนแปลงตามสัญญาณเสียง สัญญาณเสียงที่ปนอยู่ในสัญญาณ AM จะปรากฏเป็นกรอบคลื่น (ENVELOPE) บนและล่างดังเช่นรูปที่ 2.2 (ก) เป็นสัญญาณเสียงที่มีแอมพลิจูดขนาดหนึ่ง โดยรูปที่ 2.2 (ข) คือสัญญาณ AM ที่มีสัญญาณเสียงในรูปที่ 2.2 (ก) มอดูเลต ในทางตรงข้ามถ้าสัญญาณเสียงมีแอมพลิจูดเล็กดังรูปที่ 2.2 (ค) สัญญาณ AM ที่เกิดขึ้น ก็จะมีกรอบ (การเปลี่ยนแปลงทางแอมพลิจูด) เล็กดังด้วย ดังรูปที่ 2.2 (ง)



รูปที่ 2.1 การมอดูเลตทางแอมพลิจูดโดยอุปกรณ์อนลิเนียร์



รูปที่ 2.2 การใช้สัญญาณเสียงที่มีขนาดมากและน้อยเพื่อมอดูเลตบนคลื่นพาหะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไซด์แบนด์ AM

เราลองวิเคราะห์สัญญาณ AM (รูปที่ 2.3) ในเชิงความถี่ดูว่าเป็นอย่างไรและมีองค์ประกอบอะไรบ้าง ในที่นี้สัญญาณ AM เกิดจากสัญญาณพาหะรูปไซน์ความถี่ 1 เมกะเฮิร์ตซ์ มอดูเลตด้วยสัญญาณเสียง รูปไซน์ความถี่ 10 กิโลเฮิร์ตซ์ ดูเผินๆ อาจจะพบว่าผลของการมอดูเลตแบบ AM ของสัญญาณ 1 เมกะเฮิร์ตซ์กับ 10 กิโลเฮิร์ตซ์ น่าจะได้เป็นพาหะ 1 เมกะเฮิร์ตซ์กับสัญญาณเสียง 10 กิโลเฮิร์ตซ์เท่านั้น อย่างไรก็ตามถ้าเราป้อนสัญญาณทั้งหมด ให้แก่วงจรฟิลเตอร์ชนิดแบนด์พาสความถี่ 10 กิโลเฮิร์ตซ์กับ ความถี่ 1 เมกะเฮิร์ตซ์ เราจะพบว่าเอาต์พุตจากวงจรฟิลเตอร์ชนิดแบนด์พาสความถี่ 10 กิโลเฮิร์ตซ์ เป็นศูนย์ ดังรูปที่ 2.3 (ข) แต่เอาต์พุตจากวงจรแบนด์พาสความถี่ 1 เมกะเฮิร์ตซ์จะไม่เป็นศูนย์ สรุปได้ว่า สัญญาณพาหะ 1 เมกะเฮิร์ตซ์ เมื่อถูกมอดูเลตด้วยสัญญาณเสียง 10 กิโลเฮิร์ตซ์แล้ว ผลปรากฏว่าสัญญาณ 10 กิโลเฮิร์ตซ์ จะไม่มีอยู่ในคลื่นพาหะที่มอดูเลตแล้วเลยแต่ไปปรากฏเป็นกรอบ คลื่นแทน อย่างไรก็ตามเราทราบดีว่าคลื่น AM จะต้องมีส่วนประกอบของสัญญาณเสียงปนอยู่แน่นอน เพราะกรอบของรูปคลื่นเป็นสิ่งที่ยืนยันอย่างชัดเจน

ถ้าเราใช้วงจรฟิลเตอร์ชนิดแบนด์พาสแบบพิเศษที่สามารถจูนความถี่ใกล้เคียงกับความถี่ 1 เมกะเฮิร์ตซ์ หรือใช้สเปกตรัมอนาไลเซอร์ เพื่อค้นหาสัญญาณที่เข้าไปมอดูเลตบนพาหะว่าไปหลบซ่อนอยู่ในส่วนใด ของพาหะ ด้วยวิธีนี้เราจะพบว่า สัญญาณ AM นั้นนอกจากจะมีพาหะตัวเดิม 1 เมกะเฮิร์ตซ์แล้ว ยังมีคลื่นข้างเคียงเกิดขึ้นอีก 2 ข้างคือที่ความถี่ 1.01 เมกะเฮิร์ตซ์ กับ 0.99 เมกะเฮิร์ตซ์ คลื่นข้างเคียงทั้ง 2 ข้างนี้เรียกว่า ไซด์แบนด์ (SIDE BAND) ซึ่งเราจะตรวจ พบได้โดยใช้ฟิลเตอร์ที่มีความคมหรือความละเอียดในการจูนตรวจค้นหาสัญญาณ ดังรูปที่ 1.13 (ค) ไซด์แบนด์ที่มีความถี่สูงกว่า เรียกว่า ไซด์แบนด์ด้านบน (UPPER SIDE BAND หรือ USB) ความถี่ของ USB เท่ากับผลรวม ความถี่พาหะกับความถี่สัญญาณมอดูเลต

$$\text{ความถี่ USB} = f_c + f_m$$

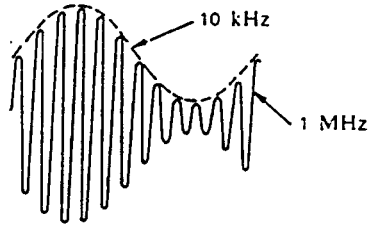
ในที่นี้ f_c คือ ความถี่พาหะ
 f_m คือ ความถี่ของสัญญาณที่เข้าไปมอดูเลต

ตามตัวอย่าง $f_c = 1 \text{ MHz}$ $f_m = 10 \text{ KHz}$ ฉะนั้นความถี่ของ USB จะเท่ากับ

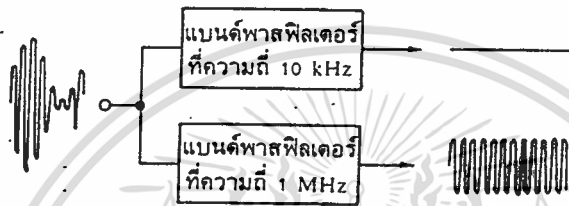
$$\begin{aligned} \text{ความถี่ USB} &= 1 \text{ MHz} + 10 \text{ kHz} \\ &= 1.01 \text{ MHz} \end{aligned}$$

สำหรับไซด์แบนด์ที่มีความถี่ต่ำกว่าเรียกว่า ไซด์แบนด์ด้านล่าง (LOWER SIDE BAND หรือ LSB) ความถี่ของ LSB เท่ากับความถี่พาหะลบด้วยความถี่ของสัญญาณที่เข้าไปมอดูเลต ในกรณีนี้จะได้

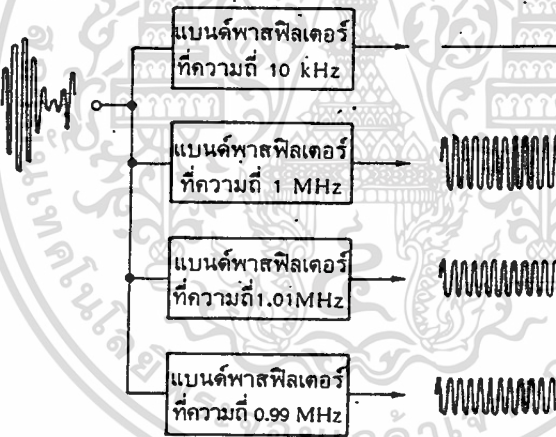
$$\begin{aligned} \text{ความถี่ LSB} &= f_c - f_m \\ &= 1 \text{ MHz} - 10 \text{ kHz} \end{aligned}$$



(ก)



(ข)



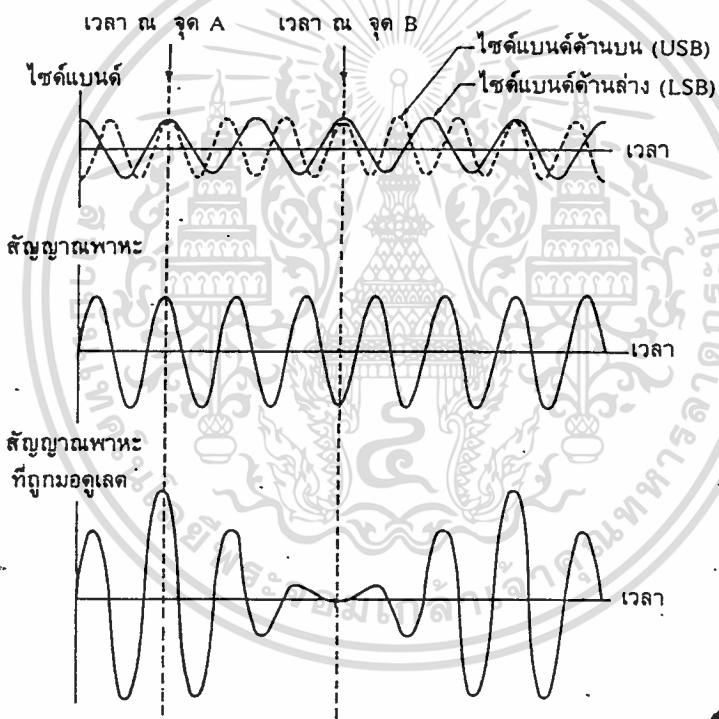
(ค)

รูปที่ 2.3 วิเคราะห์คลื่น AM ในเชิงความถี่

กล่าวโดยสรุปได้ว่า คลื่น AM ประกอบด้วยคลื่นพาหะ และคลื่นไซด์แบนด์ 2 ข้างคือ USB กับ LSB ถ้าเราลองพิจารณาแอมพลิจูดจากฟิลเตอร์ต่างๆ ในรูปที่ 2.3 (ค) จะพบว่า แอมพลิจูดของสัญญาณพาหะ และไซด์แบนด์มีค่าคงที่ ซึ่งเราทราบมาในคอนต้นแล้วว่าพาหะ จะมีแอมพลิจูด คงเดิมเสมอจนกว่า จะมีสัญญาณเข้าไปมอดูเลต แต่แอมพลิจูดเฉลี่ยของพาหะ ยังมีค่าคงเดิมตลอดไป ไม่ว่าจะมอดูเลตหรือไม่ก็ตาม

อาจจะมีส่วนสูงสั้นกว่าแอมพลิจูดของสัญญาณ แต่ละตัว ที่ประกอบขึ้นเป็น AM คือพาหะกับคลื่นไซด์แบนด์นั้นมีแอมพลิจูดคงที่ได้อย่างไร ทั้ง ๆ ที่แอมพลิจูดหรือขนาดของสัญญาณ AM ก็เปลี่ยนแปลงตามสัญญาณที่เข้าไปมอดูเลต ให้พิจารณารูปที่ 2.4 ซึ่งแสดงให้เห็นคลื่นไซด์แบนด์ทั้งคู่และคลื่นพาหะ สังเกตว่าพาหะมีแอมพลิจูดคงที่ และไซด์แบนด์ก็มีแอมพลิจูดเท่ากันทั้งคู่ และคงที่ด้วยอย่าลืมว่า ความถี่ของคลื่นพาหะต้องอยู่ระหว่างกลางของความถี่ไซด์แบนด์ทั้งสอง สมมติพิจารณาที่จุด A ซึ่งสัญญาณทั้งสามมีเฟสตรงกันแต่ละสัญญาณอยู่ตรงขอบบวกพอดี ทำให้ผลรวมได้แอมพลิจูด ของ สัญญาณ AM ค่าสูง พิจารณาที่จุด B ซึ่งไซด์แบนด์มีเฟสตรงกัน แต่พาหะมีเฟสต่างจากไซด์แบนด์ 180 องศา ผลรวมจะได้แอมพลิจูดค่าต่ำของสัญญาณ AM

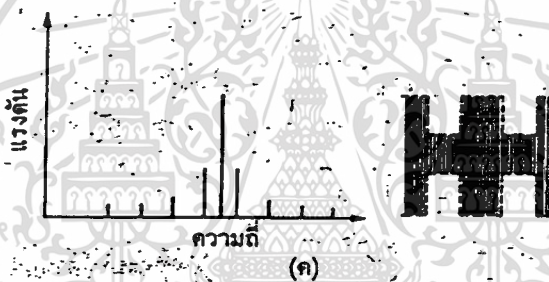
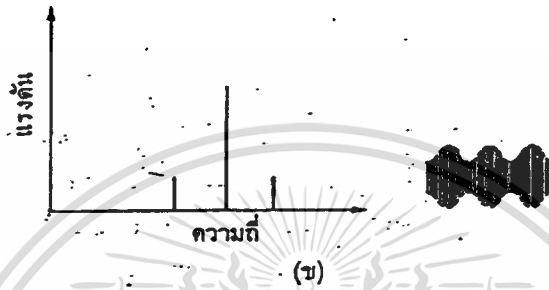
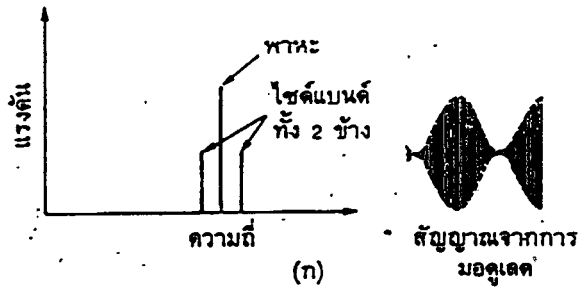
จากการวิเคราะห์ในรูปที่ 2.4 สรุปได้ว่ารูปร่างของกรอบคลื่นไม่ได้ขึ้นอยู่กับแอมพลิจูดของ สัญญาณไซด์แบนด์ แต่ความถี่ของสัญญาณไซด์แบนด์ทั้งคู่จะเป็นตัวกำหนด เฟสของสัญญาณไซด์แบนด์ว่าจะตรงเฟส หรือต่างเฟสกับพาหะ ซึ่งทำให้เกิดการเปลี่ยนแปลงของ แอมพลิจูดของสัญญาณ AM หรือ กรอบคลื่น แอมพลิจูดของไซด์แบนด์จะขึ้นอยู่กับแอมพลิจูดของกรอบคลื่น นั่นคือเป็นตัวกำหนด เปอร์เซนต์การมอดูเลต ทั้งนี้เนื่องจากไซด์แบนด์อาจมาเสริมหรือหักล้าง กับแอมพลิจูด ของพาหะก็ได้ กล่าวโดยสรุปอีกครั้งเกี่ยวกับสัญญาณ AM นั่นคือ สัญญาณ AM เกิดจากสัญญาณข่าวสารเข้ามอดูเลต บนพาหะแล้วปรากฏเป็นสัญญาณไซด์แบนด์ทั้งสองข้าง คือ USB กับ LSB



รูปที่ 2.4 ความสัมพันธ์ทางเฟสระหว่างไซด์แบนด์กับพาหะ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 เปรียบเทียบคลื่น AM ในเชิงความถี่และเชิงเวลา

ในรูปที่ 2.5 แสดงการวิเคราะห์สัญญาณ AM ซึ่งมอดูเลตด้วยสัญญาณเสียงที่มีรูปคลื่นชนิดต่างๆ ถ้าตรวจสอบด้วยออสซิลโลสโคป จะเห็นว่าไซด์แบนด์จะผสมเสริม หรือหักล้างกับพาหะทำให้ออมพลิจูดของสัญญาณเปลี่ยนแปลง สังเกตว่าในรูปที่ 2.5 (ก) ออมพลิจูดของไซด์แบนด์จะเท่ากับครึ่งหนึ่งของพาหะสถานะเช่นนี้เป็นสถานะของ การมอดูเลตเต็มที หรือเปอร์เซ็นต์การมอดูเลตเท่ากับ 100 เปอร์เซ็นต์ ฉะนั้นเมื่อสัญญาณทั้งพาหะและไซด์แบนด์มีเฟสตรงกัน ออมพลิจูดของไซด์แบนด์รวมกับพาหะจะได้ 2 เท่าของพาหะและเมื่อสัญญาณไซด์แบนด์ทั้ง (คู่) และพาหะมีเฟสตรงข้ามกัน ออมพลิจูดรวมจะได้ศูนย์

ระบบการส่งข้อมูล

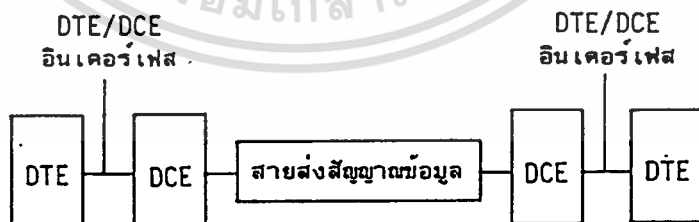
นิยามของการส่งสัญญาณข้อมูล (DATA TRANSMISSION) คือการส่งข่าวสาร (INFORMATION) เพื่อป้อนให้แก่อุปกรณ์ประมวลข่าวสาร (INFORMATION PROCESSING EQUIPMENT) หรือส่งข่าวสารที่ได้รับการประมวลแล้วอุปกรณ์ประมวลข่าวสารที่เป็นที่รู้จักกันดีก็คือคอมพิวเตอร์ ดังนั้นระบบส่งสัญญาณข้อมูลจึงเป็นระบบที่เชื่อมโยงคอมพิวเตอร์กับอุปกรณ์รับส่งข้อมูลปลายทางที่อยู่ห่างไกลกัน ให้สามารถส่งผ่านข่าวสารระหว่างกันได้ซึ่งหมายถึงจะต้องประกอบด้วยอุปกรณ์ที่ใช้รับส่งข้อมูล และตัวกลางนำสัญญาณแบบต่าง ๆ

รูปที่ 2.6 แสดงโครงสร้างของระบบสื่อสารข้อมูล (DATA COMMUNICATION SYSTEM) ซึ่งรวมเอาระบบส่งสัญญาณข้อมูลคอมพิวเตอร์และอุปกรณ์รับส่งข้อมูลแบบต่าง ๆ เข้าเป็นระบบประมวลข่าวสารอันหนึ่ง ดังนั้นจึงอาจกล่าวได้ว่าระบบส่งสัญญาณข้อมูลเป็นปัจจัยสำคัญในการกำจัดอุปสรรคที่เกิดจาก ระยะทาง ของระบบสื่อสารข้อมูล



รูปที่ 2.6 ตัวอย่างของระบบสื่อสารข้อมูล

อุปกรณ์ที่กำเนิดและรับส่งข้อมูลในข่ายคอมพิวเตอร์เรียกว่า อุปกรณ์รับส่งข้อมูลปลายทาง (DATA TERMINAL EQUIPMENT : DTE) ส่วนอุปกรณ์ที่ทำหน้าที่สื่อสารข้อมูลที่คู่กับอุปกรณ์รับ-ส่งข้อมูลปลายทาง เรียกว่า อุปกรณ์สื่อสารข้อมูล (ATA TERMINATING EQUIPMENT : DCE) ดังในรูปที่ 2.7



รูปที่ 2.7 DTE และ DCE

(1) รหัสที่ใช้สำหรับการแลกเปลี่ยนข่าวสาร

การรับส่งข่าวสารระหว่าง DTE มีขั้นตอนดังนี้ DCE จะโมดูเลขข่าวสารจาก DTE ด้านส่งแล้วส่งไปยัง DTE ทางด้านรับโดยผ่าน DCE ทางด้านรับ สัญญาณที่ทำการส่งจะมีค่าเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

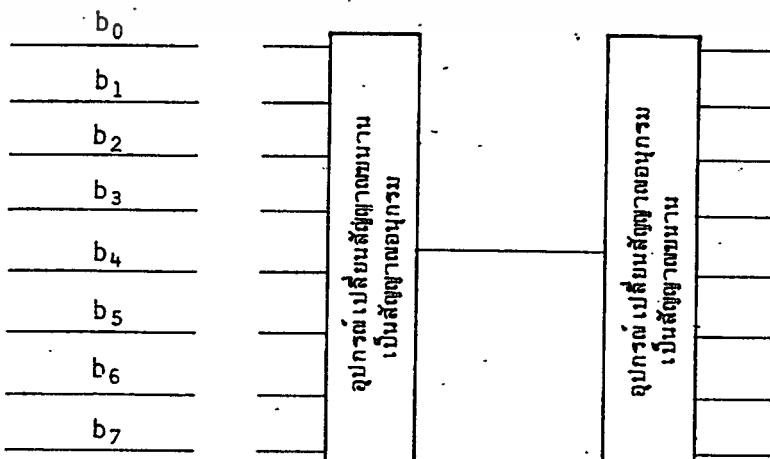
หน่วยพื้นฐานของข่าวสารซึ่งจะเป็น 0 หรือ 1 แต่ละตัวนั้น เรียกว่า บิต (BIT) รหัสที่ใช้สัญญาณ 2 ค่าประกอบกันขึ้นเรียกว่า ไบนารีโค้ด (BINARY CODE) ไบนารีโค้ดที่ใช้ในการสื่อสารข้อมูลถูกกำหนดเป็นมาตรฐานโดย ISO ประเทศไทยใช้ทั้งหมดรหัส 7 หน่วย และรหัส 8 หน่วย

คำว่ารหัส 7 หน่วย และรหัส 8 หน่วย ในที่นี้หมายถึง รหัสที่ใช้สัญญาณไบนารี 7 บิต และ 8 บิต ตามลำดับ นอกจากนี้รหัส 7 หน่วย และรหัส 8 หน่วยแล้ว ยังมีรหัสที่ใช้กับการส่งโทรพิมพ์ ซึ่งมีแบบรหัส 5 หน่วย (ใช้ส่งตัวอักษรอังกฤษ) และรหัส 6 หน่วย (ใช้ส่งตัวอักษรไทย)

(2) การส่งแบบขนานและการส่งแบบอนุกรม

เนื่องจาก DTE เป็นอุปกรณ์ที่จัดการเก็บไบนารีโค้ดอยู่แล้ว ดังนั้นการแลกเปลี่ยนข่าวสารระหว่าง DTE จึงอยู่ในหน่วยของไบนารีโค้ด ซึ่งอาจจะเป็น 8 บิต เป็นต้น การส่งไบนารีโค้ดนี้จะทำให้ง่ายที่สุด ดังรูปที่ 2.8 (ก) คือเตรียมจำนวนสายให้เท่ากับจำนวนบิตในไบนารีโค้ดแล้วส่งออกไปแบบขนาน แต่วิธีนี้จะต้องใช้จำนวนสายมาก จึงใช้ได้เฉพาะกรณีที่ต่อกันเป็นระยะทางสั้น ๆ เท่านั้น โดยทั่วไปมักจะใช้วิธีที่แสดงไว้ในรูปที่ 2.8 (ข) กล่าวคือจัดการกับสัญญาณไบนารีโค้ดที่ประกอบด้วยหลาย ๆ บิตนั้นให้มาเรียงกันแบบอนุกรมเสียก่อน แล้วจึงส่งไปยังจุดหมายปลายทางด้วยสายส่งสัญญาณเพียงเส้นเดียว

ในกรณีเช่นนี้ โดยทั่วไปอุปกรณ์ที่คอยทำหน้าที่ดังกล่าวนี้จะติดตั้งอยู่ใน DTE ดังนั้น ระบบส่งสัญญาณระหว่าง DCE ของเครื่องปลายทาง จึงไม่จำเป็นต้องคำนึงถึงโครงสร้างของข้อมูลที่ส่งอยู่ เพียงแต่ทำหน้าที่ส่งผ่านข้อมูลแบบโปร่งใส (TRANSPARENT) ออกไปเป็นใช้ได้ ลักษณะเช่นนี้ จะทำให้ระบบส่งข้อมูลสามารถส่งข้อมูลได้ทุกโครงสร้าง และยังสามารถส่งสัญญาณชนิดอื่น ๆ ที่สามารถแสดงออกมาในรูปของสัญญาณไบนารีได้ ตัวอย่างเช่น สัญญาณ FACSIMILE เป็นต้น



เอกสารนี้เป็นเอกสาร (ก) การส่งแบบขนาน (ข) การส่งแบบอนุกรม ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) ส่วนประกอบของระบบส่งสัญญาณข้อมูล

ระบบส่งสัญญาณข้อมูลนั้นพิจารณาจากภายนอกจะมีส่วนประกอบดังรูปที่ 2.7 คือมี DCE อยู่ปลายทางสองด้านและเชื่อมโยงกันด้วยสายส่ง

ส่วน DCE นั้นก็คือ อุปกรณ์ที่ทำหน้าที่ในการเปลี่ยนสัญญาณนั่นเอง กล่าวคือ จะทำหน้าที่ในการเปลี่ยนสัญญาณข้อมูลที่ได้จาก DTE/DCE อินเตอร์เฟส ซึ่งมีเงื่อนไขแบบหนึ่งไปเป็นสัญญาณข้อมูลที่สามารถส่งไปตามช่องสัญญาณซึ่งจะมีเงื่อนไขทางสัญญาณ อีกแบบหนึ่งได้ การเปลี่ยนสัญญาณจะเปลี่ยนทั้งในเชิงไฟฟ้าและเชิงกายภาพ

สำหรับตัวกลางนำสัญญาณนั้น อาจจะเป็นสายไขว้คู่ สายโคแอกเชียล เส้นใยแสง หรือ อาจจะเป็นคลื่นวิทยุ หรือใช้แสงส่งไปในบรรยากาศ นอกจากนั้น ถ้าระยะทางไกลมากก็อาจจะต้องมีสถานี ทวนสัญญาณ (REPEATER ระหว่างทาง หรือมีอุปกรณ์อื่น ๆ ที่จะทำให้ การใช้งานมีประสิทธิภาพ สูงขึ้นประกอบอยู่ด้วย เช่น อุปกรณ์มัลติเพล็กซ์สัญญาณ เป็นต้น

รูปแบบของการสื่อสารและช่องสัญญาณ

รูปแบบของการสื่อสารคือ ลักษณะของการติดต่อสื่อสารระหว่าง DTE คู่หนึ่ง ซึ่งแบ่ง ออกได้เป็น 3 แบบคือ

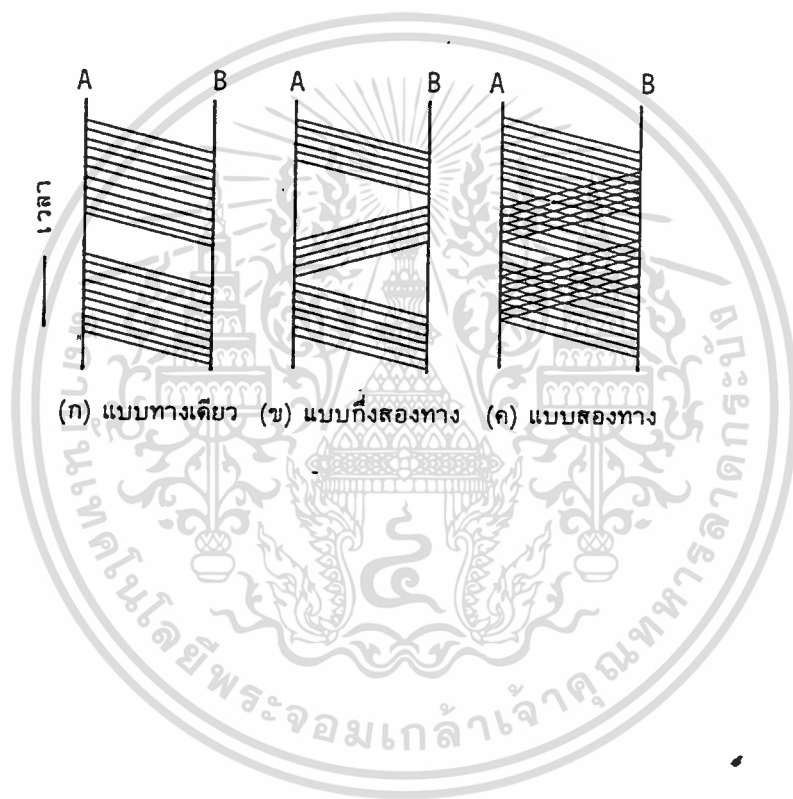
1. การติดต่อทางเดียว (SIMPLEX)
2. การติดต่อทั้งสองทาง (HALF-DUPLEX)
3. การติดต่อสองทาง (DUPLEX)

รูปแบบของการสื่อสารนี้จะเกี่ยวข้องโดยตรงกับส่วนประกอบต่าง ๆ ของระบบส่งสัญญาณ ข้อมูลตั้งแต่ DCE จนถึงอุปกรณ์ส่งสัญญาณข้อมูลอื่น ๆ

รูปที่ 2.9 แสดงรูปแบบของการสื่อสารทั้งสามแบบนี้คือ การสื่อสารแบบทางเดียว ก็คือ มีการส่งสัญญาณจากอุปกรณ์เพียงด้านเดียว และสัญญาณเดินทางไป ทิศทางเดียว ส่วน แบบกึ่ง -สองทางนั้น การส่งสัญญาณจะเกิดขึ้นได้ทั้งสองด้าน แต่เวลาจะไม่ตรงกัน สำหรับแบบ สองทางนั้น การส่งสัญญาณจากทั้งสองด้านจะเกิดขึ้นได้พร้อมกัน

รูปแบบของการสื่อสารจะมีความสัมพันธ์กับความเชื่อถือได้ของการส่งผ่านข่าวสาร และประสิทธิภาพในเชิงการใช้เวลา รหัสข้อมูลที่ใช้ในการสื่อสารข้อมูลนั้น โดยปกติจะพยายามใช้ จำนวนบิตที่น้อยที่สุด จึงทำให้เกิดปัญหาด้าน REDUNDANCY คือถ้าในระหว่างทางที่ส่งอยู่เกิดการผิดพลาด ขึ้นเพียง 1 บิต อุปกรณ์ปลายทางก็จะตีความรหัสนั้นผิดพลาดไปได้ และในระบบส่งสัญญาณข้อมูล โดยทั่วไปมักจะมีการผิดพลาดเกิดขึ้นในระหว่างการส่งได้เสมอ ดังนั้นการหา

มาตรการจัดการกับความผิดพลาดที่อาจเกิดขึ้นได้นั้น จึงเป็นหัวข้อสำคัญในงานด้านสื่อสารข้อมูล
วิธีการโดยทั่วไปอาจจะใช้วิธีส่งรหัสตรวจจับการผิดพลาดหรือรหัสแก้ไขการผิดพลาดพ่วงเข้าไปด้วย
นอกจากนั้นอาจจะใช้วิธีควบคุมการส่งข้อมูลระหว่าง DTE ในการแก้ไขความผิดพลาดที่เกิดขึ้นได้



รูปที่ 2.9 รูปแบบของการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์

CP-Z84C11

บอร์ด CP-Z84C11 เป็นบอร์ดที่ใช้ CPU Z84C11 ของบริษัท ZILOG มาเป็น CPU ประจำบอร์ด RUN ที่ SPEED 10 MHz CPU Z84C11 นี้ก็คือ CPU ที่รวบรวมเอาชิพวงจรต่างๆ ของ ZILOG เข้าด้วยกัน คือ

- Z84C00 เป็น CPU Z80 แบบ CMOS RUN ที่ 10 MHz
- Z84C30 เป็น Z80 CTC แบบ CMOS RUN ที่ 10 MHz
- CGC เป็น CLOCK GENERATOR CONTROL CIRCUIT
- WDT เป็นวงจร WATCH DOG TIMER
- POWER ON RESET เป็น วงจร RESET CPU เมื่อ VOLT VCC ต่ำกว่า 2.2 V
- 40 BIT PARALLEL PORT เป็น PORT 8 BIT จำนวน 5 PORT ใช้งาน

จะเห็นว่าบอร์ด CP-Z84C11 ก็คือ บอร์ดที่มี CPU เป็น Z80 พร้อมด้วยวงจรต่างๆ เพิ่มเข้ามาเหมาะสำหรับผู้ที่ใช้ทำงาน Z80 ที่ต้องการขีดความสามารถสูงๆ ขึ้นอีกประมวลผลได้รวดเร็วขึ้น พร้อมทั้งบอร์ด CP-84C11 นี้ยังสามารถต่อกับอุปกรณ์ต่างๆ ที่ไม่ได้มีอีก เช่น LCD , PRINTER, KEYBOARD EEPROM

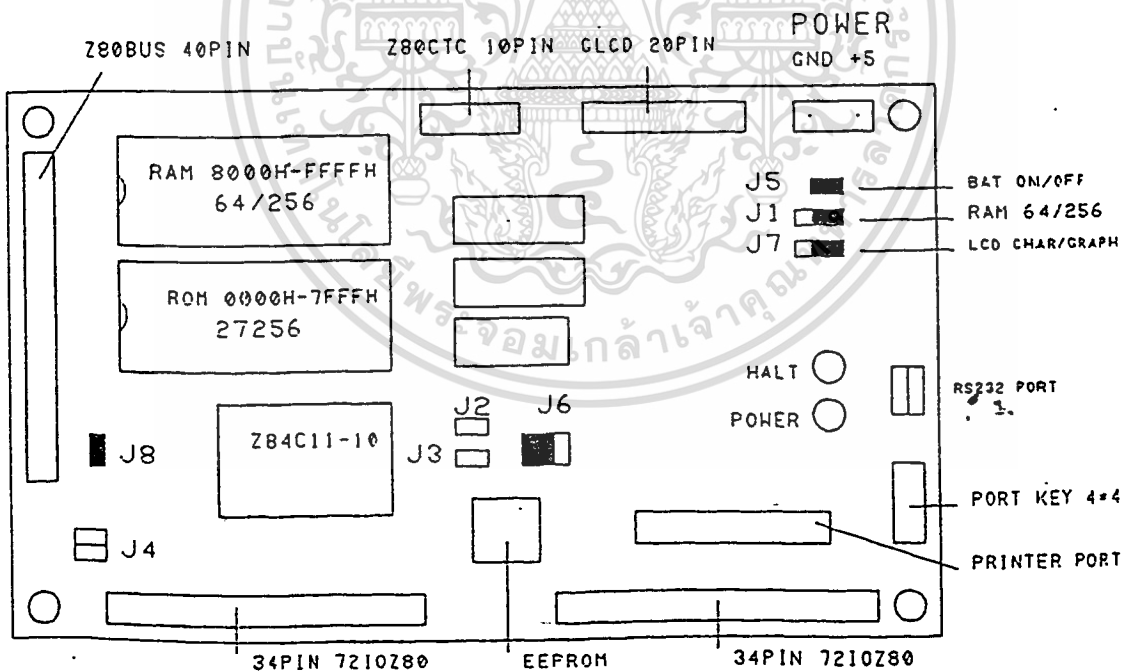
นอกจากนี้บอร์ด CP-Z84C11 ยังสามารถต่อร่วมกับ ET-DEBUGGER Z84C11 ทำให้สามารถพัฒนา ระบบ Z84C11 นี้ ร่วมกับเครื่อง PC ผ่านทาง RS232 PORT ได้ โดยสามารถสั่ง RUN SINGLE STEP , ดูค่า REGISTER , LOAD FILE จากเครื่อง PC , หรือทำ ONE LINE ASM, DASM ได้ด้วย

ข้อมูลของ BOARD CP-Z84C11

CPU

Z84C11 ของบริษัท ZILOG โดยเป็น HIGH SPEED OPERATION โดยมีให้เลือก 2 รุ่น คือ RUN ที่ความถี่ 6 MHz และที่ 10 MHz ในบอร์ดนี้เราใช้ CPU รุ่น RUN 10 MHz CPU เบอร์นี้ ของ ZILOG จะเป็นแบบ CMOS ทำให้กินกระแสต่ำมากในขณะที่เรา SET ให้อยู่ใน STOP MODE จะกินกระแสเพียง 50 UA ข้อดีอีกอย่างหนึ่งของ CPU ตัวนี้ก็คือ ถึงแม้จะ RUN 10 MHz ก็ตาม เราก็สามารถ SET ให้ RUN เพียงครั้งเดียวก็ได้คือ 5 MHz ในกรณีที่ ใช้ RAM หรือ ROM ที่มี ACCESS TIME ต่ำๆ โดยเมื่อ POWER ON แล้ว CPU จะ RUN เพียง 5 MHz เท่านั้น ตอนเริ่มต้นและเมื่อเราต้องการ RUN 10 MHz ก็สามารถใช้ SET ได้ด้วย โปรแกรม รูปแบบลักษณะของ CPU จะเป็น PACKAGE แบบ 100 PIN QFP

CP-Z84C11 PARALLEL I/O CONTROLLER



รูปที่ 2.10 CP-Z84C11 PARALLEL I/O CONTROLLER

MEMORY

CP-Z84C11 สามารถใส่หน่วยความจำได้สูงสุด 64 KBYTE โดยแบ่งเป็น

SOCKET 1 EPROM สามารถใส่ EPROM ขนาด 32 KBYTE ได้โดยเป็นเบอร์ 27256 มีหน่วยความจำเริ่มจาก ADDRESS 0000H - ADDRESS 7FFFH

SOCKET 2 RAM สามารถใส่ RAM ขนาด 8-32 KBYTE ได้โดยเป็นเบอร์ 6264 หรือ 62256 มีหน่วยความจำเริ่มจาก ADDRESS 8000H-FFFFH โดย SET ตำแหน่ง JUMPER ในการกำหนดเบอร์ไอซีดังรูป:-

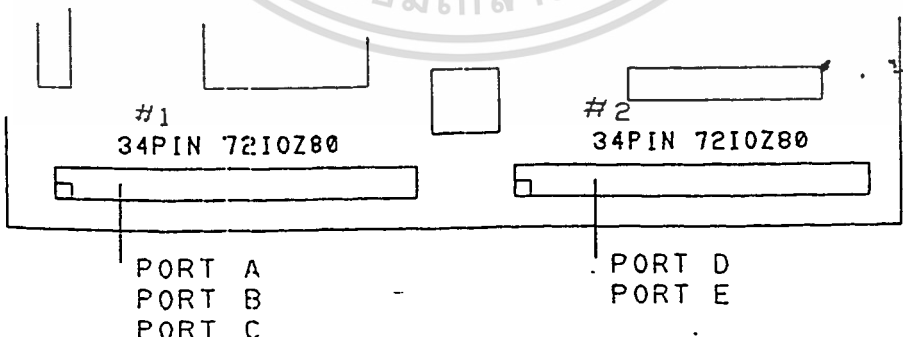


รูปที่ 2.11 SOCKET RAM

หน่วยความจำ RAM ส่วนนี้เราสามารถต่อ BATTERY 3.6 V ใช้ BACKUP ข้อมูลของหน่วยความจำนี้ได้ด้วย

PORT

CP-Z84C11 จะมี PORT ใช้งาน 40 BIT I/O หรือ 5 PORT ด้วยกัน โดย 5 PORT นี้จะเป็น PORT ในตัว CPU โดยจะมีตำแหน่งต่อออกมาดังรูป



รูปที่ 2.12 PORT

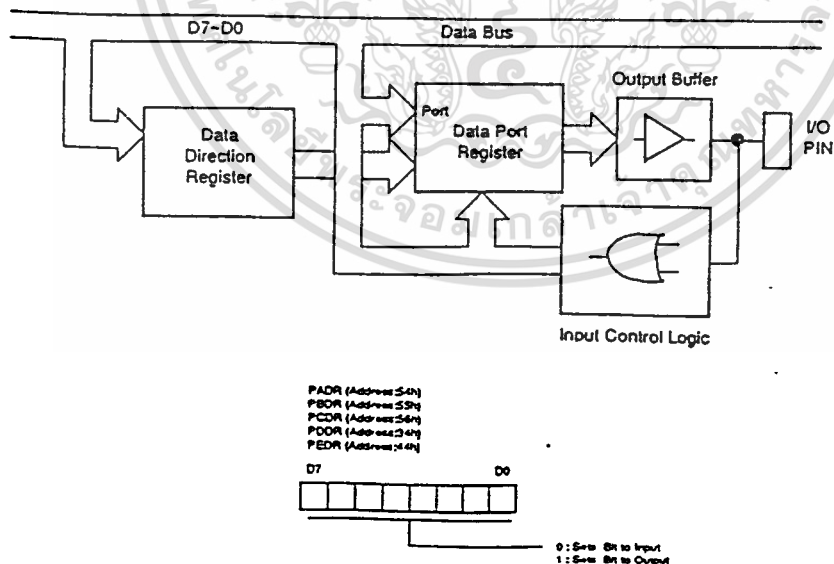
PORT ที่ต่อออกมานี้จะอยู่เป็น CONNECTOR 34 PIN (72IOZ80) สามารถต่อร่วมใช้กับอุปกรณ์สนับสนุนต่างๆ ของทาง อีทีที ได้มากมาย เช่น ชุด ET-SSRAC, ชุด ET-AD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHANNEL PORT	ADDRESS PORT
PORT A DATA PORT	50 H
PORT B DATA PORT	51 H
PORT C DATA PORT	52 H
PORT D DATA PORT	30 H
PORT E DATA PORT	40 H

CHANNEL CONTROL PORT	ADDRESS PORT
PORT A DATA DIRECTION REGISTER	54 H
PORT B DATA DIRECTION REGISTER	55 H
PORT C DATA DIRECTION REGISTER	56 H
PORT D DATA DIRECTION REGISTER	34 H
PORT E DATA DIRECTION REGISTER	44 H

เราสามารถสั่งให้ PORT ของ Z84C11 เป็น INPUT ได้โดยอิสระ BIT ต่อ BIT โดยถ้าเรา SET ค่าออกที่ PORT DATA DIRECTION REGISTER ทำให้ BIT ใดเป็น "1" ก็คือให้ PORT DATA ของ BIT นั้นๆ เป็น OUTPUT PORT (ถ้า SET PORT นั้นเป็น OUTPUT PORT : ค่าเริ่มต้นจะเป็นศูนย์) และถ้าเราต้องการให้เป็น INPUT PORT ก็ SET ค่าออกที่ PORT DATA DIRECTION REGISTER เป็นค่า "0"



รูปที่ 2.13 PORT DIRECTION REGISTER

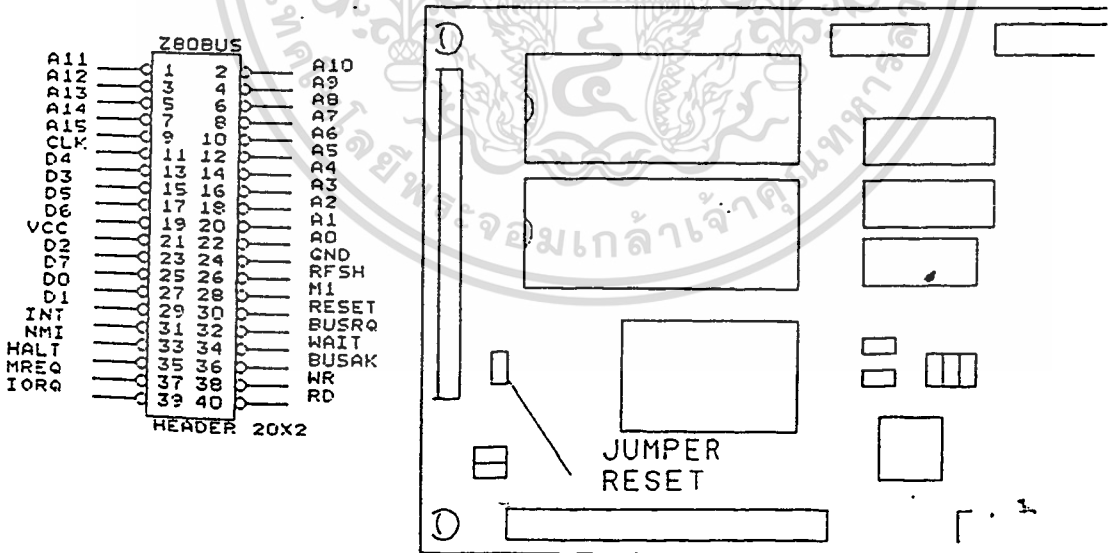
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80 BUS

CP-Z84C11 จะมีส่วนขยายระบบได้ทั้ง Z80 BUS โดยมีลักษณะเหมือนกับ CPU Z80 40 PIN ทำให้เราสามารถต่อขยายบอร์ดได้เช่น ต่อขยาย PORT 8255 ได้อีกโดยใช้บอร์ด 72IOZ80 เป็นต้น มีข้อพิเศษคือเนื่องจาก CPU Z84C411 นี้มีส่วนวงจร POWER ON RESET ในตัวอยู่แล้ว ต้องใช้วงจร POWER ON RESET ต่อจากภายนอก เช่น CPU เบอร์อื่นๆ ฉะนั้นถ้าเราไปต่อร่วมกับ Z80 BUS อื่นๆที่มี R, C ต่ออยู่แล้วอาจจะทำให้ CPU Z84C11 นี้ อยู่ในสภาพ RESET ตลอดเวลาได้บอร์ด CP--Z84C11 จึงมี JUMPER ในการเลือกไม่ให้ต่อ RESET จากภายนอกได้ ดังรูป โดยถ้ามีการต่อ R, C ที่ RESET ภายนอกก็ให้ถอด JUMPER RESET ออก

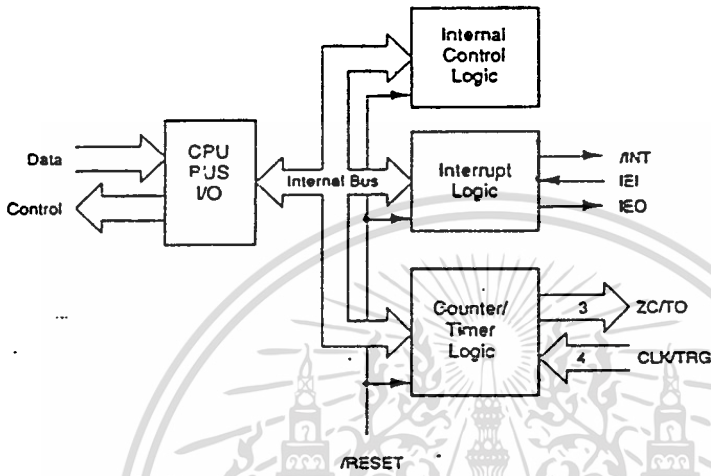
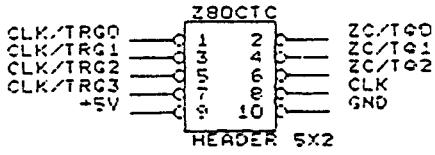
CTC

CP--Z84C11 จะมีวงจร Z84C30 (Z80 CTC) ต่อร่วมกันอยู่ในตัวเรียบร้อยแล้ว โดยบอร์ด CP--Z84C11 จะต่อขงใช้งานของ CTC ออกมาที่ CONNECTOR 10 PIN แล้วดังรูป ส่วนขา INT ของ CTC นั้นจะต่อกับขา INT ของ Z80 ในตัวโดยเป็นแบบ WIRED-OR เรียบร้อยในตัวแล้ว รายละเอียดของ CTC ในบอร์ดนี้จะเหมือนกับ Z80CTC ทั่วไป



รูปที่ 2.14 Z80 BUS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 Z80 CTC

CTC ADDRESS PORT
ADDRESS PORT

CHANNEL

CH 0	10 H
CH 1	11 H
CH 2	12 H
CH 3	13 H



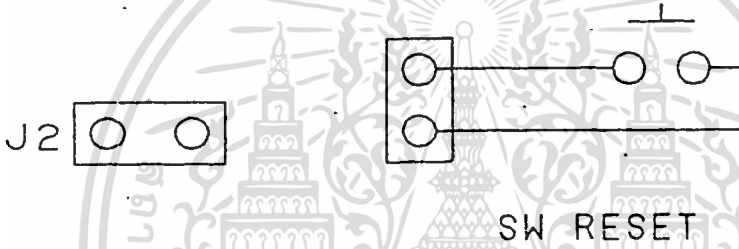
MS1, MS2 (I4)

MS1, MS2 จะเป็นขา INPUT ของ CPU Z84C11 โดยเป็นการ SET ให้ CPU ถ้อยู่ในคำสั่ง HALT แล้วจะให้อยู่ในสภาพใด (RUN, IDLE1, IDLE2, STOP)

MS1	MS2	HALT STATE
1	1	RUN MODE
0	0	IDLE1 MODE
0	1	IDLE2 MODE
1	0	STOP MODE

RESET (I2)

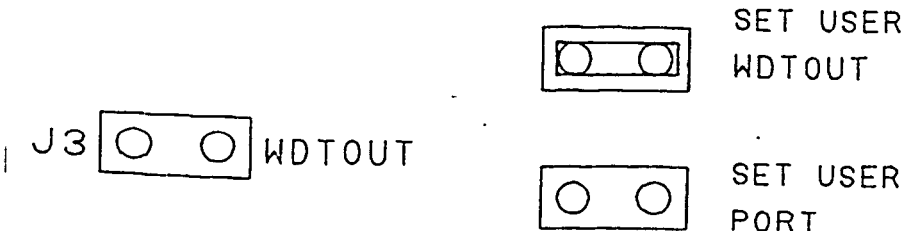
เป็น INPUT ต่อเข้ากับขา RESET CPU Z84C11 ห้ามต่อกับวงจรประเภท R, C RESET



รูปที่ 2.16 SW. RESET

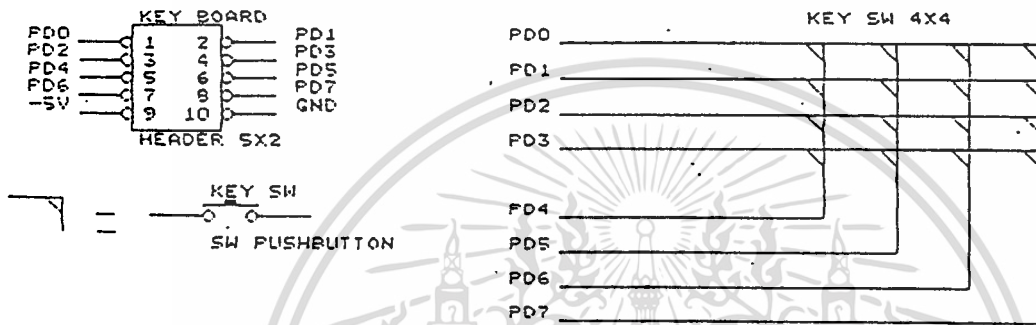
WDTOUT (I3)

เป็น PIN จาก PORT PE BIT 7 ซึ่งทำหน้าที่ 2 อย่างคือเป็น DATA PORT ทั่วไปแล้วก็ยังเป็น WATCH DOG TIMER OUTPUT ด้วยโดยจะเป็น OPEN-DRAIN I/O เมื่อถูก SET ให้เป็น WDTOUT โดย PIN นี้จะถูกต่อโดยตรงเข้ากับ PIN RESET ของ CPU โดยปกติถ้าเราไม่มีการ SET WDTOUT แล้วให้ถอด JUMPER ออก ห้ามต่อถ้าไม่มีการใช้ WDTOUT เพราะเมื่อเราใช้ PORT PE 7 จะทำให้ CPU ถูก RESET ตามคำสั่ง OUT PORT PE 7 นั้นด้วย



KEYBOARD

เป็น CONNECTOR ขนาด 10 PIN ต่อจาก PORT PD โดยมี R-PULL-UP 10K ต่ออยู่ด้วยโดย CONNECTOR 10 PIN นี้จะสามารถต่อเป็นรูปแบบ KEY ขนาด 4X4 คือ 16 KEY โดยเขียนโปรแกรม SCAN KEY ขึ้นหรือจะนำไปใช้เป็น PORT อีกระยะในงานอื่นๆ ก็ได้



รูปที่ 2.20 KEYBOARD

BAT ON/ OFF (J15)

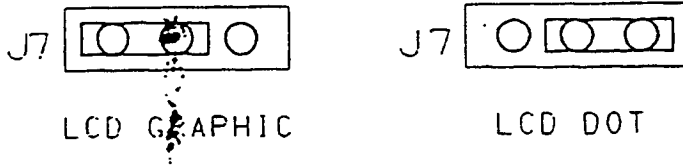
เป็น CONNECTOR 2 PIN โดยเราใช้ JUMPER ในการปิด , เปิดไฟจาก BATTERY 3.6 V ถ้ามีการต่อใช้งาน BATTERY

64 , 256 (J6)

เป็น CONNECTOR 3 PIN ใช้ JUM เลือกว่าหน่วยความจำ RAM ของเราจะใช้หน่วยความจำ ไอสี่เบอร์อะไร 6264 (8 KBYTE) หรือ 62256 (32 KBYTE)

LCD J7

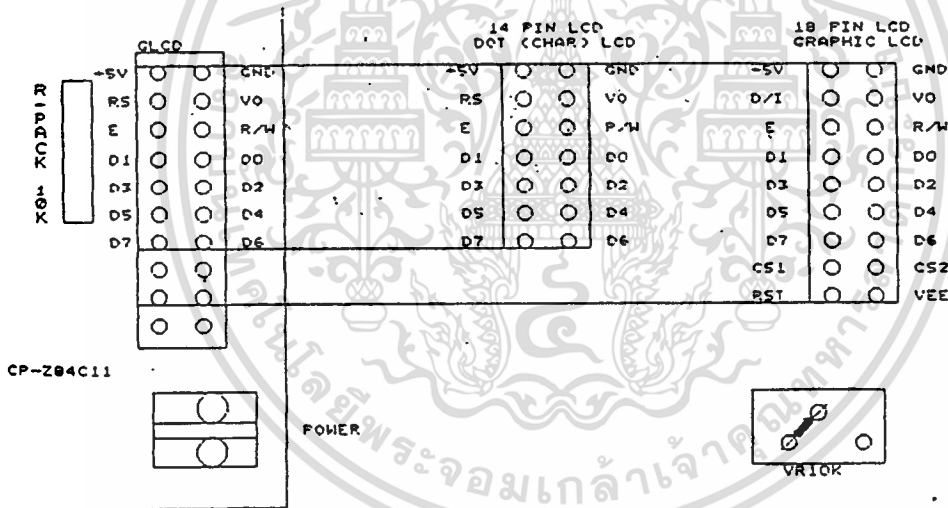
เป็น CONNECTOR 3 PIN ใช้เลือกว่าจะต่อ LCD MODULE ประเภทใดคือ LCD MODULE แบบตัวอักษรหรือ LCD MODULE แบบ GRAPHIC



รูปที่ 2.21 LCD

GLCD

เป็น CONNECTOR ขนาด 20 PIN โดยสามารถต่อ LCD ได้ 2 แบบคือ DOT TYPE หรือ GRAPHIC TYPE โดยเราเลือกต่อได้ถ้าเป็น DOT TYPE คือ ตัวอักษรที่จะใช้ 14 PIN ต่อได้ดังรูป หรือถ้าจะเลือกต่อแบบ GRAPHIC TYPE ก็จะต่อใช้งาน 18 PIN



รูปที่ 2.22 GLCD

เราสามารถปรับความคมชัดของ LCD ได้ด้วยการต่อ VR 10 K เข้าในบอร์ดหรือ ถ้าต้องการให้ชัดสุดก็จะต่อ JUM VR ตามรูปเลขก็ได้ (BOARD CP-Z84C11 จะไม่ต่อ VR 10 K ให้)

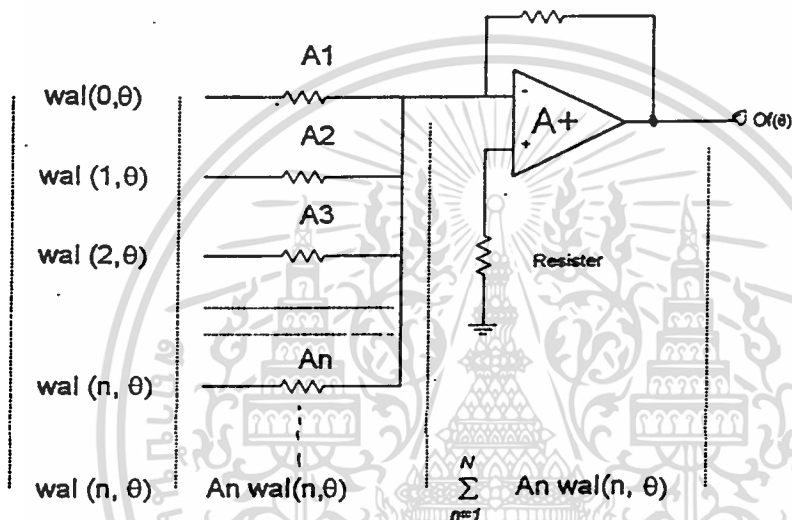
LCD ADDRESS PORT

CHANNEL PORT	ADDRESS PORT
WRITE DATA INSTRUCTION	80 H
WRITE DATA TO CG OR DD RAM	82 H
READ BUSY FLAG AND ADDRESS	84 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วอสฟังก์ชัน (WALSH FUNCTION)

WALSH FUNCTION เรียกว่าเป็น DIGITAL FOURIER SERIES คือใช้เทคนิคของ FOURIER SERIES ในการกระจายฟังก์ชันรายคาบใดๆ ให้อยู่ในรูปของอนุกรม SQUARE WAVE ซึ่งเราจะเรียก SQUARE WAVE แต่ละตัวที่ได้ว่าเป็น WALSH FUNCTION ดังนั้นในการปฏิบัติ เราสามารถสร้าง สัญญาณที่เป็นฟังก์ชันรายคาบใด ๆ ได้จาก WALSH FUNCTION ที่เป็นส่วนประกอบของฟังก์ชันนั้น โดยใช้ค่าความต้านทานในการ WEIGHTING แทนค่าสัมประสิทธิ์ของ WALSH FUNCTION แต่ละตัวแล้วนำไปรวมกัน ดังรูป



รูปที่ 2.23 แสดงการสร้างสัญญาณที่เป็นฟังก์ชันรายคาบจาก WALSH FUNCTION

โดยที่ $f(\theta) = A_0 + \sum_{n=1}^{\infty} A_n \text{Wal}(n, \theta)$, $0 \leq \theta \leq 1$ (1)

เมื่อ $\theta = 1/T$, T เป็นเวลา 1 Period ของ $f(\theta)$

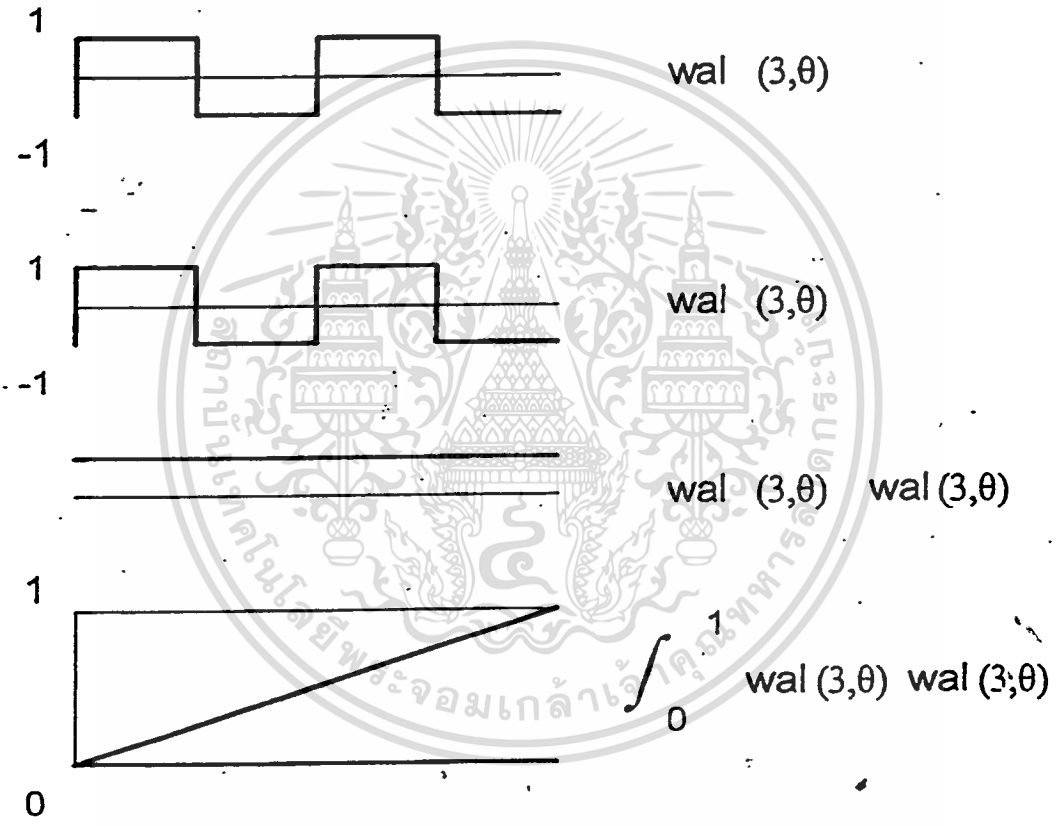
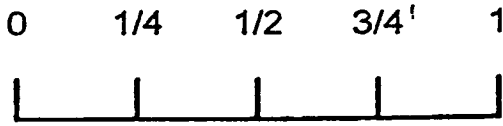
A_0 เป็นสัมประสิทธิ์ทาง DC

A_n เป็นสัมประสิทธิ์ของ WALSH FUNCTION ORDER ที่ n

สัมประสิทธิ์ A_0 และ A_n สามารถหาค่าได้โดย ใช้คุณสมบัติของ ORTHOGONAL เนื่องจาก WALSH FUNCTION เป็นเซตของ ORTHOGONAL FUNCTION ซึ่งมีคุณสมบัติดังนี้

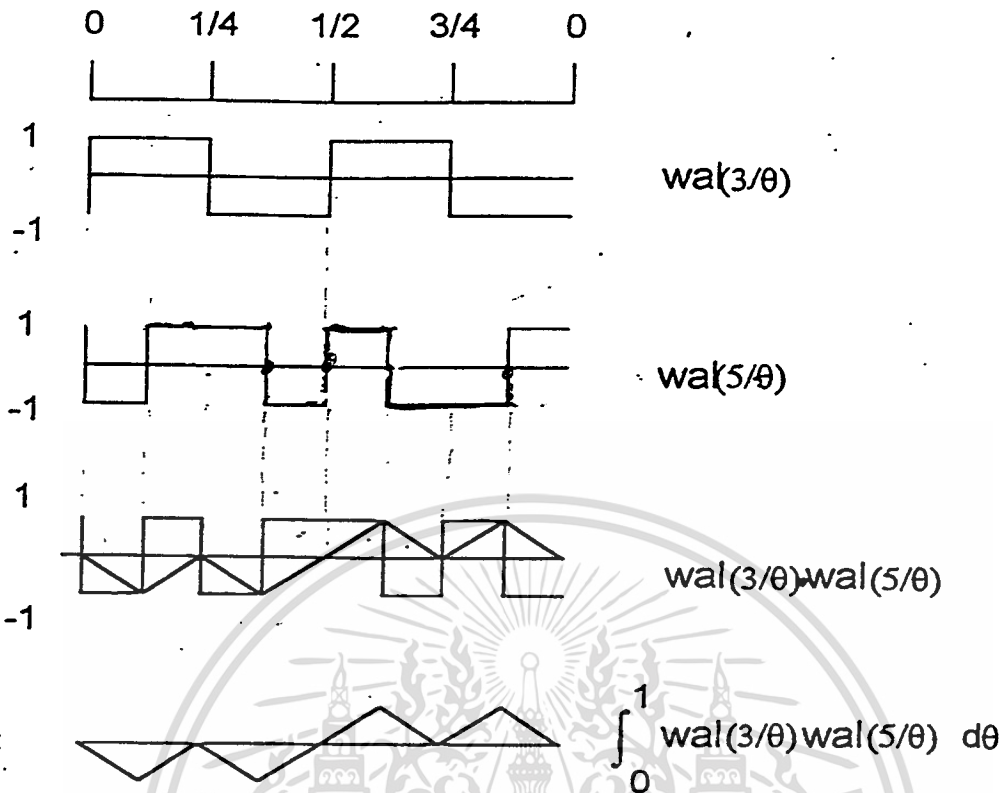
$$\int_0^1 \text{Wal}(n, \theta) \text{Wal}(m, \theta) d\theta = \begin{cases} 1 ; n = m \\ 0 ; n \neq m \end{cases} \quad (2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 แสดงคุณสมบัติการ . ORTHOGONAL เมื่อ $n=m$

$$\int_0^1 \text{Wal}(3, \theta) \text{Wal}(3, \theta) = 1 ; 0 \leq \theta \leq 1$$



รูปที่ 2.25 แสดงคุณสมบัติการ ORTHONORMAL เมื่อ $n \neq m$

$$\int_0^1 \text{Wal}(3, \theta) \text{Wal}(5, \theta) d\theta = 0 ; 0 \leq \theta \leq 1$$

เราจะใช้คุณสมบัติ ORTHONORMAL ในการหาค่าสัมประสิทธิ์ A_0 และ A_n ดังนี้จากสมการ

(1)

$$f(\theta) = A_0 + \sum_{n=1}^{\infty} A_n \text{Wal}(n, \theta) ; 0 \leq \theta \leq 1$$

$$f(\theta) \text{Wal}(0, \theta) = A_0 \text{Wal}(0, \theta) + \sum_{n=1}^{\infty} A_n \text{Wal}(n, \theta) \text{Wal}(0, \theta)$$

$$\int_0^1 f(\theta) \text{Wal}(0, \theta) d\theta = \int_0^1 A_0 \text{Wal}(0, \theta) d\theta + \sum_{n=1}^{\infty} \int_0^1 A_n \text{Wal}(n, \theta) \text{Wal}(0, \theta) d\theta$$

เนื่องจากคุณสมบัติของ ORTHOGONAL เทอมอื่น ๆ จะมีค่าเป็น ศูนย์ยกเว้น

$$A_0 = \int_0^1 f(\theta) d\theta ; \text{Wal}(0, \theta) = 1 \tag{3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกันสัมประสิทธิ์ A_n หาได้จากสมการ (1) ดังนี้

$$\begin{aligned} f(\theta) \cdot \text{Wal}(n, \theta) &= A_0 \text{Wal}(n, \theta) + \sum_{n=1}^{\infty} A_n \text{Wal}(n, \theta) \cdot \text{Wal}(n, \theta) \\ \int_0^1 f(\theta) \text{Wal}(n, \theta) d\theta &= \int_0^1 A_0 \text{Wal}(n, \theta) d\theta + \sum_{n=1}^{\infty} \int_0^1 A_n \text{Wal}(n, \theta) \text{Wal}(n, \theta) d\theta \\ A_n &= \int_0^1 f(\theta) \text{Wal}(n, \theta) d\theta \end{aligned} \quad (4)$$

WALSH FUNCTION ประกอบด้วย RADEMACHER FUNCTION $\{R(\theta)\}$

หลายตัว EXCLUSIVE OR กันโดย

$$\text{Wal}(n, \theta) = \pi_{i=1}^n g_i R_i(\theta) \quad (5)$$

เมื่อ n เป็น ORDER ของ WALSH FUNCTION

g เป็น GRAY CODE ของเลข BINARY n

$R(\theta)$ เป็น RADEMACHER FUNCTION

ตัวอย่างเช่น $\text{Wal}(13, \theta)$ สามารถหาได้ดังนี้

1. แปลงเลข 13 เป็น BINARY

2. แปลง BINARY ของเลข 13 เป็น GRAY CODE

$$13 = 1101_2 = 1011_g$$

ดังนั้น $\text{Wal}(13, \theta) = R_4(\theta) R_2(\theta) R_1(\theta)$

RADEMACHER FUNCTION : $R_1(\theta)$ นิยามโดย

$$R_1(\theta) = \text{sgn} [\text{Sin}(2\pi\theta)] \quad (6)$$

$$\text{sgn } x = \begin{cases} +1 & ; x \geq 0 \\ -1 & ; x < 0 \end{cases} \quad (7)$$

sgn เป็น SIGNUM FUNCTION ใช้ NORMALIZED ให้ $R_i(\theta)$ เป็น

ORTHONORMAL FUNCTION จากนิยามของ RADEMACHER FUNCTION

จะเห็นได้ว่าการเปลี่ยนแปลงเครื่องหมาย (จุดตัดศูนย์) เท่ากับ 2 ครั้งในช่วงเวลา 1 PERIOD

สรุปแล้ว RADEMACHER FUNCTION คือ สัญญาณนาฬิกา ซึ่งมีขนาด +1 และ -1 มีความถี่เป็น 2^{i-1} ของสัญญาณ $f(\theta)$ คือ

$R_1(\theta)$ มีความถี่เท่ากับ $f(\theta)$

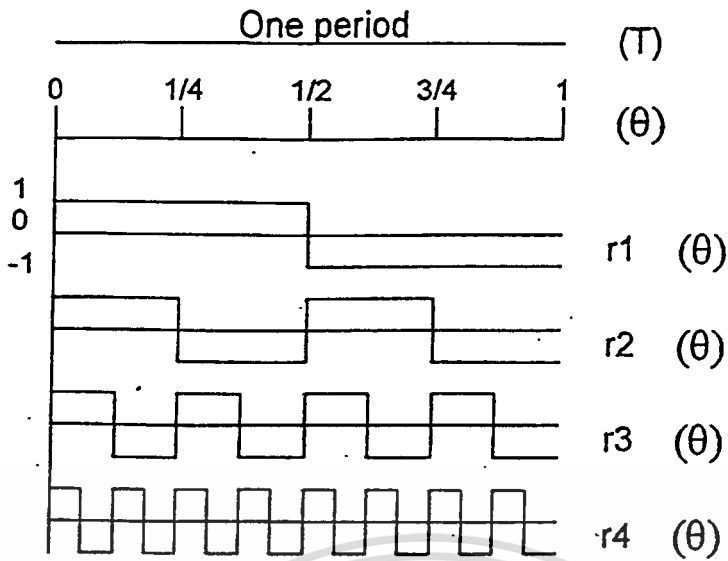
$R_2(\theta)$ มีความถี่เป็น 2 เท่าของ $f(\theta)$

$R_3(\theta)$ มีความถี่เป็น 4 เท่าของ $f(\theta)$

$R_4(\theta)$ มีความถี่เป็น 8 เท่าของ $f(\theta)$

แสดงในรูปที่ 2.26

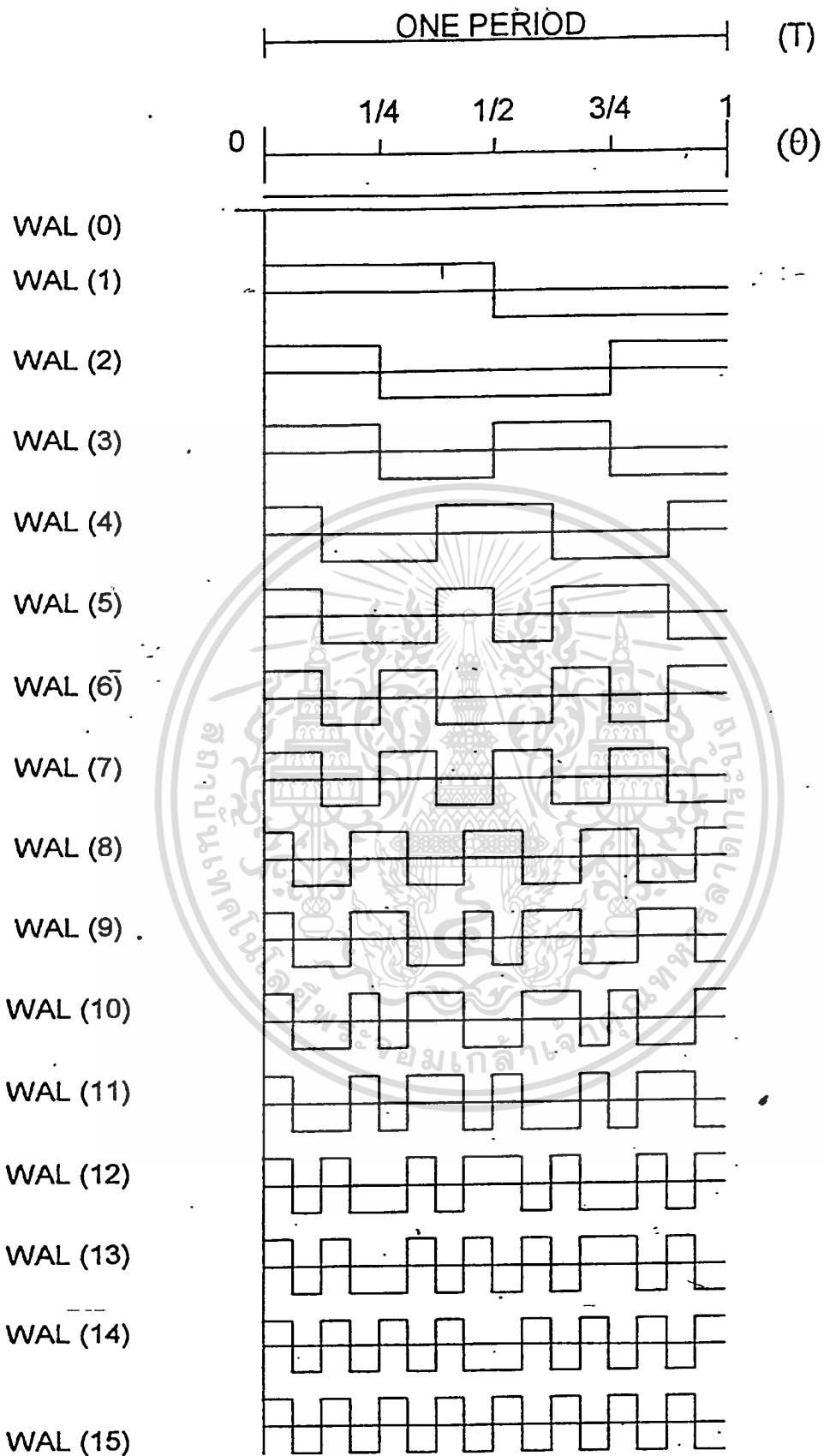
ดังนี้



รูปที่ 2.26 RADEMACHER FUNCTION อันดับ 1 ถึง 4

WALSH FUNCTION ที่ GENERATE ได้จาก $R_1(\theta)$ ถึง $R_4(\theta)$ ในรูปที่ 6 ก็คือ $Wal(0)$ ถึง $Wal(15)$

- Walsh (0) - 1
 Walsh (1) - R_1
 Walsh (2) - $R_2 \oplus R_1$
 Walsh (3) - R_2
 Walsh (4) - $R_3 \oplus R_2$
 Walsh (5) - $R_3 \oplus R_2 \oplus R_1$
 Walsh (6) - $R_3 \oplus R_1$
 Walsh (7) - R_3
 Walsh (8) - $R_4 \oplus R_3$
 Walsh (9) - $R_4 \oplus R_3 \oplus R_1$
 Walsh (10) - $R_4 \oplus R_3 \oplus R_2 \oplus R_1$
 Walsh (11) - $R_4 \oplus R_3 \oplus R_2$
 Walsh (12) - $R_4 \oplus R_2$
 Walsh (13) - $R_4 \oplus R_2 \oplus R_1$
 Walsh (14) - $R_4 \oplus R_1$
 Walsh (15) - R_4

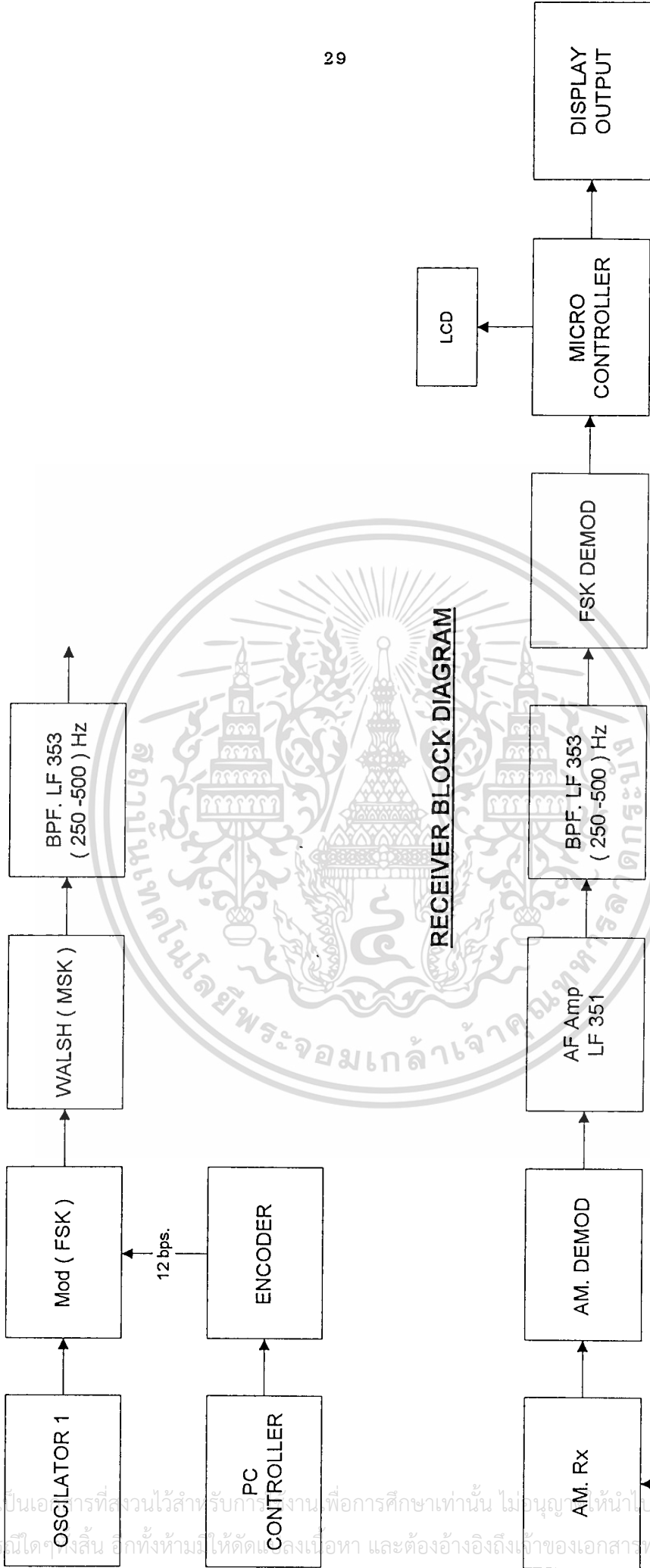


รูปที่ 2.27 แสดง WALSH FUNCTION จาก Wal (0) ถึง Wal (15)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการแจ้งให้ทราบก่อนการตีพิมพ์ โดยผู้จัดทำเอกสารนี้เพื่อใช้ในการศึกษาวิจัยเท่านั้น ไม่ควรนำเอกสารนี้ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRANSMITTER BLOCK DIAGRAM



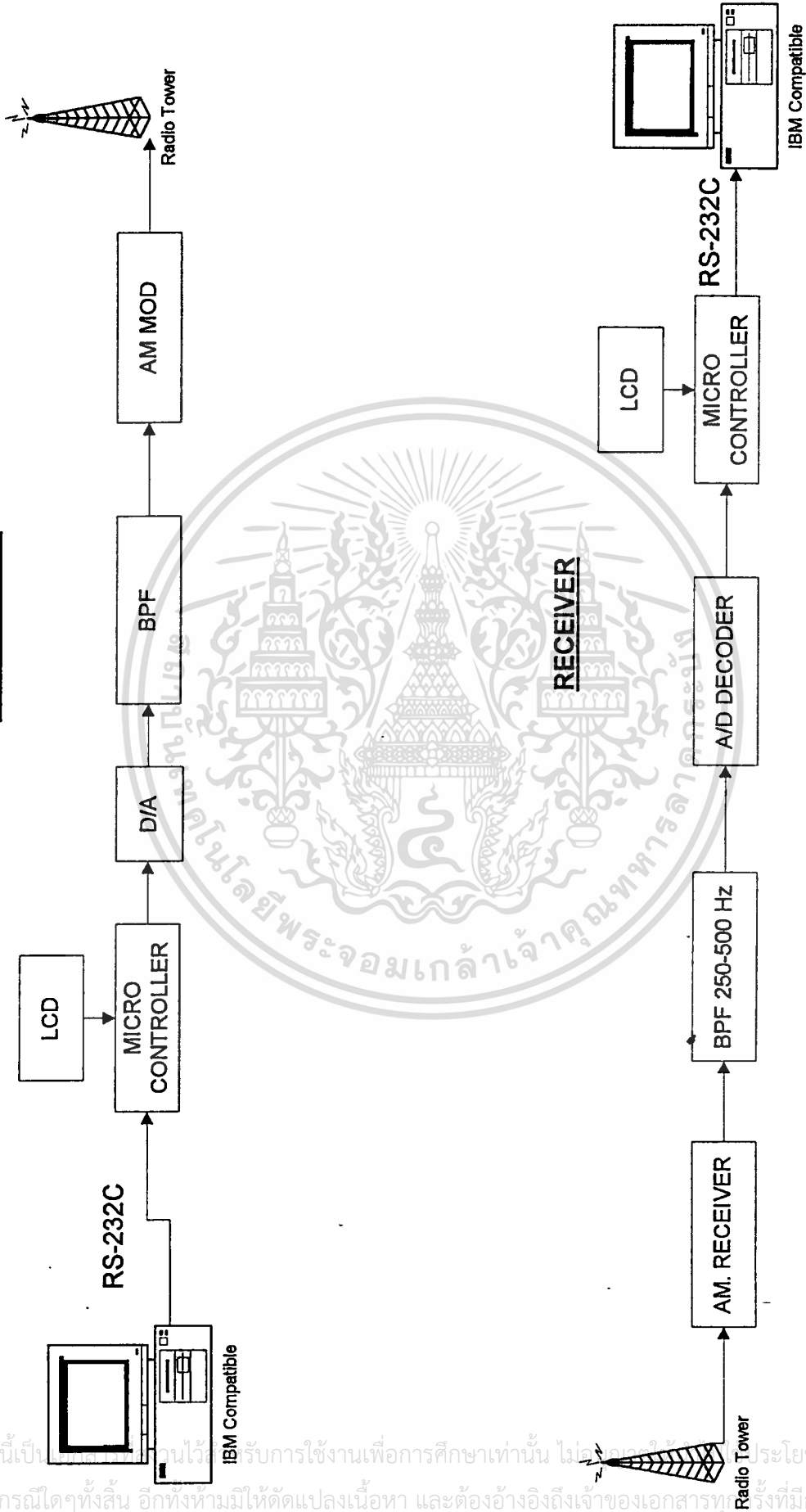
RECEIVER BLOCK DIAGRAM

รูปที่ 2.28 BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านอื่นๆ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกานำไปใช้

TRANSMITTER

RECEIVER



รูปที่ 2.29 BLOCK DIAGRAM

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารที่
มีลิงก์ที่มีการนำไปใช้

ภาครับประกอบด้วย

1. ภาครับระบบ AM ซึ่งจะรับสัญญาณที่ถูกส่งออกมาผ่านวงจร DEMODULATION เพื่อกำจัดสัญญาณ CARRIER ออกไป และผ่านวงจร BAND PASS FILTER ให้ความถี่ 250 Hz - 500 Hz ผ่านเข้ามา
2. วงจรแปลงสัญญาณข้อมูลใช้วงจร FREQUENCY SHIFT KEYING เพื่อแปลงสัญญาณ 250 Hz และ 500 Hz ออกมาเป็นข้อมูลดิจิทัลดังนี้ โดยจะแปลงสัญญาณความถี่ 250 Hz ให้ได้แฮ็ก์ทุกเป็น ' 1 ' และจะแปลงสัญญาณความถี่ 500 Hz ให้ได้แฮ็ก์ทุกเป็น ' 0 '
3. ไมโครคอนโทรลเลอร์จะรับข้อมูลจากวงจร FSK เข้ามาทาง PORT C0 เพื่อนำเข้ามาเก็บไว้ในหน่วยความจำที่ แอดเดรส AE00H การทำงานของไมโครคอนโทรลเลอร์เริ่มจากโปรแกรมหลักเซตค่าต่าง ๆ ของอุปกรณ์ แล้วจะเริ่มเข้าสู่โปรแกรมทำงานโดยแบ่งเป็น 3 โหมด คือ

MODE 1 รับข้อมูลจาก PORT C0 โดยจะรับข้อมูลจากวงจร FSK แล้วเก็บไว้ที่ ADDRESS AE 00 H จนกว่าจะหมดข้อมูล แล้วจะกลับเข้าสู่ MODE INITIAL

MODE 2 แสดงผลข้อมูลที่รับเข้ามาบนจอ LCD

MODE 3 ส่งข้อมูลออกไปที่ PC โดยผ่าน PORT RS - 232 C ของคอมพิวเตอร์จนกว่าจะหมดข้อมูล

4. โปรแกรมรับข้อมูลเข้าไมโครคอมพิวเตอร์ โดยจะรับข้อมูลจาก PORT RS - 232 C COM 1 โดยเซตค่าการส่งข้อมูลดังนี้คือ BITRATE 4800 bps. สตาร์ทบิต 1 บิต คาตัมบิต 8 บิต และสตีอปบิต 2 บิต โดยจะรับข้อมูลขึ้นมาแสดงผลบนจอภาพ และสามารถบันทึกข้อมูลเก็บไว้ใน DISK ได้

บทที่ 3 การออกแบบวงจร

-BAND PASS FILTER

-FSK DEMODULATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การออกแบบวงจร

การออกแบบ BAND PASS FILTER

การออกแบบ ACTIVE BAND PASS FILTER (250 Hz - 500 Hz)

ในการออกแบบใช้ IC LF 353 โดยมี OP-AMP 2 ตัว ถูกสร้างอยู่ในตัว IC LF 353

ขั้นตอนในการออกแบบ

1. เลือกให้ LF 353/1 ทำหน้าที่ LPF และให้ LF 353/2 ทำหน้าที่เป็น HPF ดังนั้นหา

$$f_L = 1 / 2 R_1 C_1$$

กำหนดให้ $C_1 = 0.01 \text{ mF}$

$$\text{แทนค่า } 250 \text{ Hz} = 1 / 2 \cdot 0.01 \cdot 10^{-6} \cdot R_1$$

$$R_1 = 63.66 \text{ K}\Omega$$

เลือกใช้ VR $100 \text{ K}\Omega$ เพื่อปรับค่าให้ได้ตามต้องการ

กำหนดให้ $C_2 = 0.01 \text{ mF}$

$$f_H = 1 / 2 R_4 C_2$$

แทนค่า $500 \text{ Hz} = 1 / 2 \cdot 0.01 \cdot 10^{-6} \cdot R_4$

$$R_4 = 31.85 \text{ K}\Omega$$

เลือกใช้ VR $50 \text{ K}\Omega$ เพื่อปรับค่าให้ได้ตามต้องการ

2. PASS BAND GAIN (AFT)

$$\text{หาได้จาก } (1 + R_3 / R_2) * (1 + R_5 / R_6) = \text{AFT}$$

แทนค่าในสูตร

$$\begin{aligned} & (1 + 10 \text{ K}\Omega / 10 \text{ K}\Omega) * (1 + 10 \text{ K}\Omega / 10 \text{ K}\Omega) \\ & = 4 = \text{AFT} \end{aligned}$$

การออกแบบ FREQUENCY SHIFT KEYING DEMODULATION (FSK DEMOD.)

1. ต้องการให้สถานะ ' 1 ' เป็น 250 Hz และ ' 0 ' เป็น 500 Hz และกำหนดให้ความเร็วในการส่งข้อมูล 12 bps. ซึ่งใช้ IC XR - 2211 CP มาใช้งาน

2. หา CENTER FREQUENCY (f_0)

$$\text{จากสูตร } f_0 = f_m + f_s / 2$$

$$\begin{aligned} \text{แทนค่า } f_0 &= (500 + 250) / 2 \\ &= 375 \text{ Hz} \end{aligned}$$

3. ในข้อกำหนดของ XR-2211CP นี้ R_0 จะอยู่ในช่วง $10 \text{ K}\Omega - 100 \text{ K}\Omega$

$$\text{ให้ } R_0 = R_x + R_x'$$

$$\text{เลือก } R_x = 20 \text{ K}\Omega \quad R_x' = 80 \text{ K}\Omega \quad \text{โดยใช้ VR } 100 \text{ K}\Omega$$

เพื่อปรับค่าให้ได้ตามต้องการ

4. หาค่า C_0

$$\text{จาก } f_0 = 1 / C_0 R_0$$

$$\begin{aligned} \text{ได้ } C_0 &= 1 / 375 \text{ Hz} * 100 \text{ K}\Omega \\ &= 0.026 \text{ mF} \end{aligned}$$

5. กำหนดค่า BAND WIDTH จากค่า R_1

$$\text{สูตร } R_1 = R_0 (f_0 / \Delta f)$$

$$\Delta f = f_m - f_s$$

$$= 250 - 500$$

$$= 250 \text{ Hz}$$

$$R_1 = 100 \text{ K}\Omega * 375 \text{ Hz} / 250$$

Hz

$$= 150 \text{ K}\Omega$$

6. หาค่า C_1

$$C_1 = C_0 / 4 \quad (\text{สูตรที่กำหนดมาจากคู่มือผู้ผลิต})$$

$$C_1 = 0.026 \text{ mF} / 4$$

$$= 0.0065 \text{ mF}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับขอใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. ทา Cf

จากสูตร $C_f = 3 / \text{BASE BAND SPEED (bps)}$

$$C_f = 3 / 12 \text{ bps.}$$

$$= 0.25 \text{ mF}$$

โดยที่ RF = 100 K Ω R8 = 510 K Ω สรุปได้ว่า

ถ้าต้องการส่งข้อมูลด้วยความเร็ว 12 bps. และต้องการ MARK FREQUENCY

(fm = 250 Hz) SPACE FREQUENCY (fs = 500 Hz)

เมื่อใช้ IC XR - 2211 CP สามารถหาค่าต่างๆ จากคู่มือผู้ผลิต จะได้ค่าต่างๆ ดังนี้

Ro = 100 K Ω

Co = 0.026 mF เลือกใช้ 0.027 mF

R1 = 150 K Ω

C1 = 0.0065 mF เลือกใช้ 0.0067 mF

R8 = 510 K Ω

Cf = 0.25 mF

RF = 100 K Ω RL = 5.1 K Ω

บทที่ 4 วงจรการใช้งานและการทำงาน

-วงจร ENCODER

-วงจร WALSH FUNCTION

-วงจร DECODER (FSK DEMOD.)

-วงจร BAND PASS FILTER

-วงจรภาครับ AM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

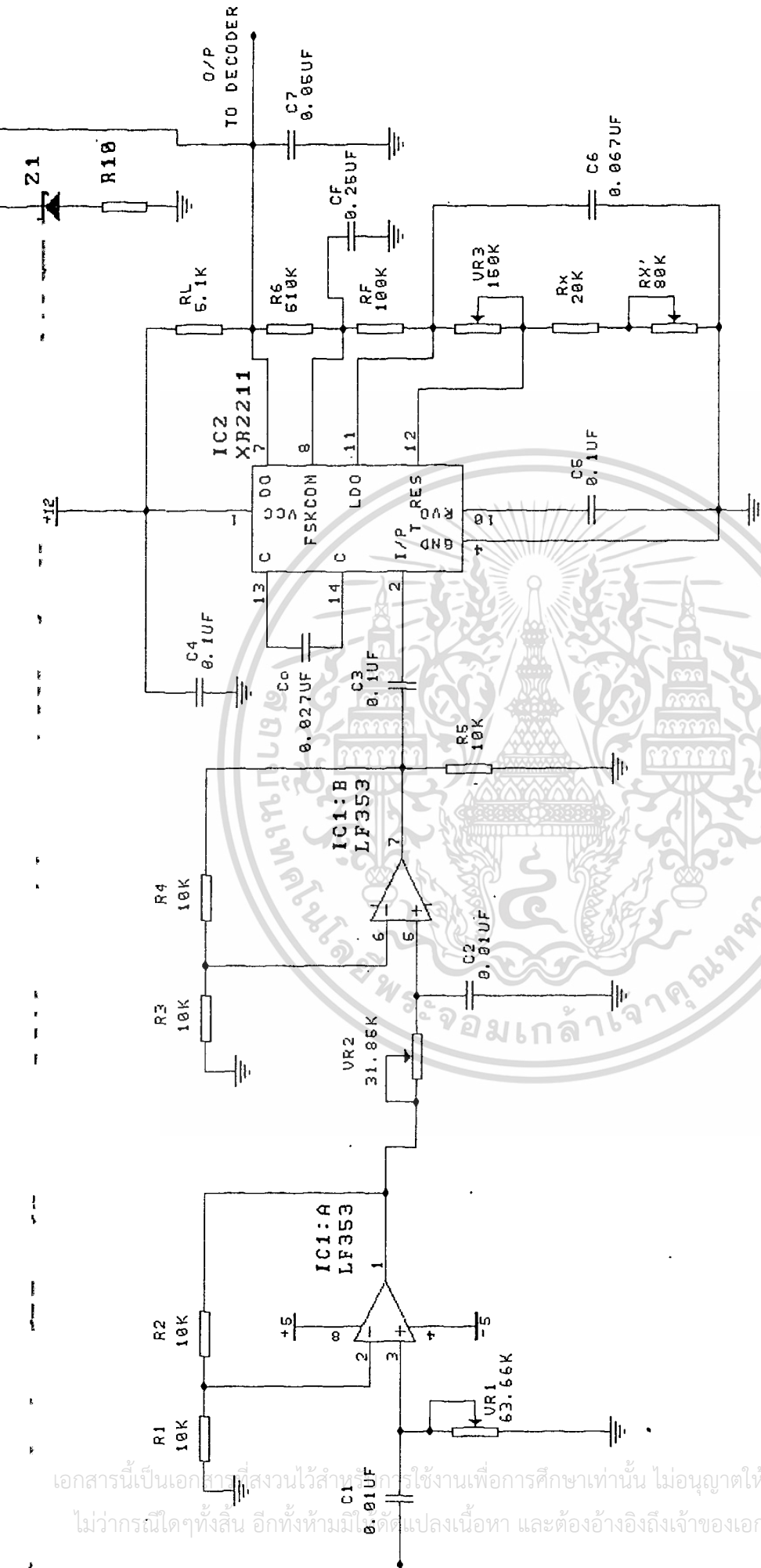
การทำงานของภาค ENCODER

การทำงานของภาคนี้ OSCILLATOR ใช้ CRYSTAL เป็นตัวกำเนิดความถี่ขนาด 10.24 Mhz ต่อใช้งานร่วมกับวงจร R, C สัญญาณความถี่ที่ได้นำไปเป็นสัญญาณ CLOCK ให้กับ วงจรทรา 64 โดยวงจรนี้ใช้ IC 74HC4060 โดยมี OUTPUT คือ ที่ Q6 ได้ความถี่ ขนาด 160 KHz หรือ (32×5 KHz) หลังจากนั้นจำความถี่ที่ได้ไปเข้าวงจรทรา 5 และ 10 จะได้ความถี่ OUTPUT ขนาด 32 KHz หรือ (32×500 Hz) ตามลำดับ สำหรับวงจรทรา ความถี่นั้นใช้ IC 74LS90 ซึ่งทำ การ ต่อเป็น BI-QUINRY โดยใช้สัญญาณ CLOCK เข้าที่ขา INPUT B และ OUTPUT QD ต่อกลับ มาเป็น INPUT A จะได้สัญญาณ OUTPUT ดังนี้ OUTPUT QA จะเป็นการทรา 10 และ OUTPUT QD จะเป็นการทรา 5 จากนั้น นำไป AND กับสัญญาณ INPUT คือตรวจสอบกับ สัญญาณ INPUT ค่าเป็น ' 0 ' หรือ ' 1 '



การทำงานของภาค WALSHP FUNCTION

WALSHP FUNCTION เป็นการใช้เทคนิคของการกระจายฟังก์ชันรายคาบใดๆ ให้อยู่ในรูปของอนุกรม SQUARE WAVE โดยใช้ค่าความต้านทานในการ WEIGHTING แทนค่าสัมประสิทธิ์ของ WALSHP FUNCTION แต่ละตัวแล้วนำไปรวมกัน ถ้าหากเราต้องการค่าที่ละเอียดใกล้เคียงค่าจริงมากเท่าใด ทำได้โดยการเพิ่ม ORDER ของ WALSHP FUNCTION ที่ใช้ วงจร WALSHP FUNCTION จะทำหน้าที่ในการสังเคราะห์สัญญาณ SINE WAVE จากสัญญาณ DIGITAL ที่ได้มาโดยใช้วงจร SYNCHRONOUS BINARY COUNTER 74C163 เป็นตัวแบ่งสัญญาณ โดยใช้หลักการตาม WALSHP FUNCTION เมื่อได้ OUTPUT 74C163 นำเข้าไป EXCLUSIVE OR ตาม RADEMACHER COMPOSITION ของ WALSHP FUNCTION แล้วป้อนเข้า D-FLIP FLOP เพื่อได้ OUTPUT ตามลำดับ แล้วนำไปเข้าวงจร SUMMING AMP. จะได้สัญญาณ OUTPUT ในลักษณะเป็น SINE WAVE ที่มีสัญญาณ DIGITAL ร่วมกัน หลังจากนั้นเข้าสู่วงจร BPF แล้วนำ สัญญาณเข้าเครื่องส่งออกอากาศต่อไป



Title		FSK DECODER	
Size	Number	Revision	Revision
A4	KMITL01	A	A
Date:	26-OCT-1995	Sheet	2 of PROJFC
File:	PROJEC02/1	Drawn	By: P. SOMBUT

รูปที่ 4.3 วงจร DECODER (FSK DEMODULATOR)

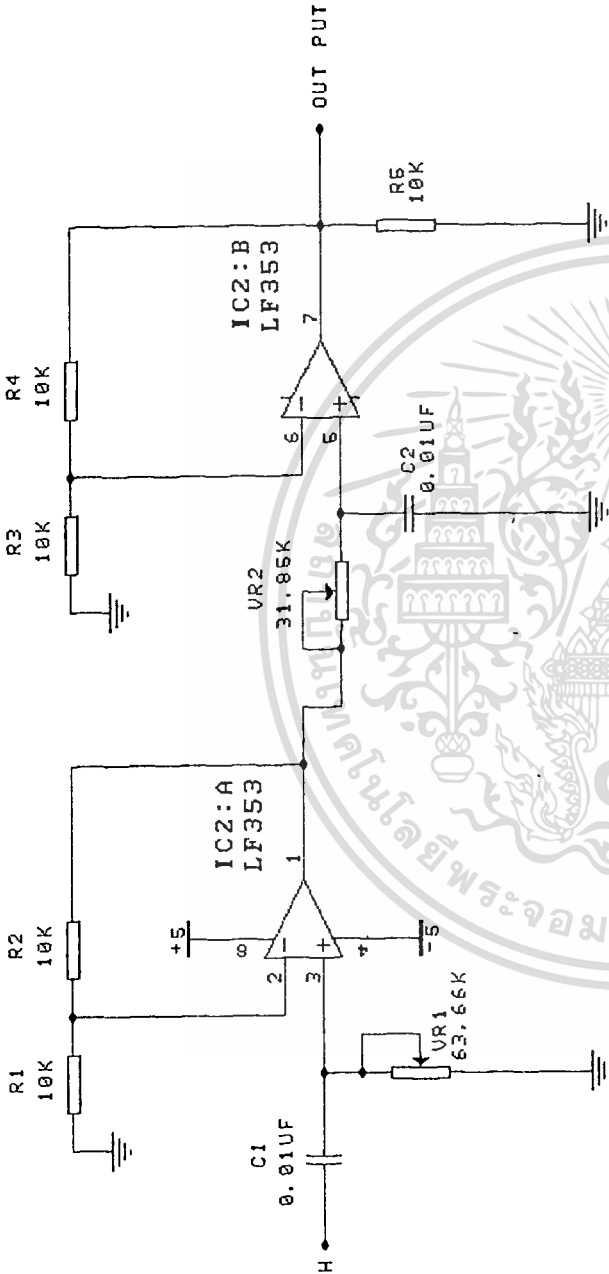
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิได้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของภาค DECODER

การทำงานของภาคนี้จะอาศัย IC XR-2211 โดยมีย่านความถี่ในการใช้งาน ตั้งแต่ 0.01 Hz - 300 KHz ค่าความต้านทาน R6 (510 Kohm) ระหว่างขา 7 กับ ขา 8 เป็นตัวป้องกันกลับทางบวก เพื่อให้ TRANSITION TIME ของ OUTPUT FSK เร็วขึ้น หลังจากวงจร BPF จะทำให้เราได้ สัญญาณ ความถี่ 2 ตัว คือ 250 Hz และ 500 Hz ซึ่งวงจรมีจะทำแปลงสัญญาณความถี่นี้เป็นสัญญาณ DIGITAL คือ '1' และ '0' ตามลำดับดังรูปที่ 4.3



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		
Size	Number	Revision
A4	KMITL	A
Date:	26-OCT 1995	Sheet 6 of KMITL
File:	PROJEC0671	Drawn By: P. SOMBAT

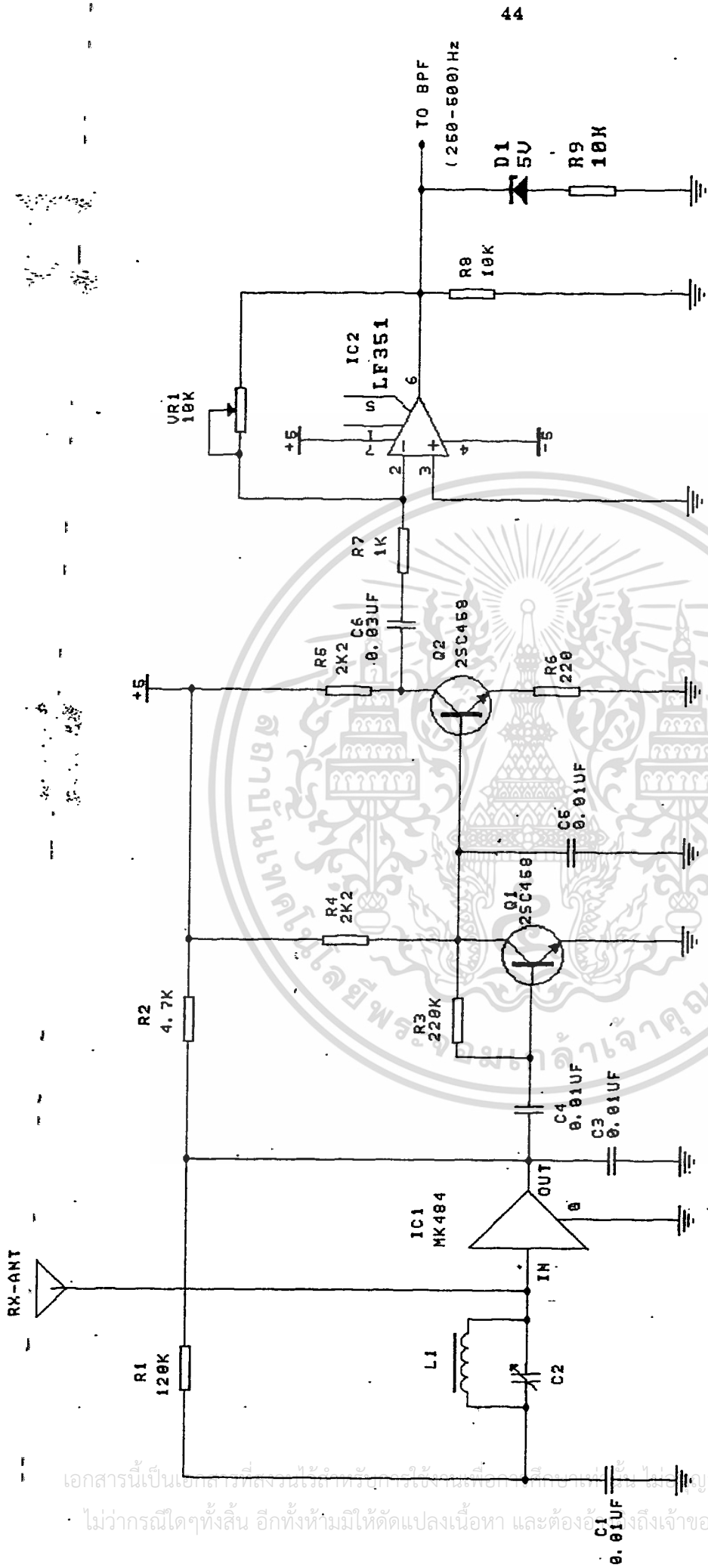
รูปที่ 4.4 วงจร BAND PASS FILTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของภาค BAND PASS FILTER

การทำงานของภาคนี้ จะใช้ IC LF 353 ทั้งที่ภาครับและภาคส่ง เพื่อใช้ในการปรับย่านความถี่ที่ต้องการ โดย IC LF 353 จะมีความต้านทานแบบปรับค่าได้ VR1 63.66 Kohm. ต่ออยู่ที่ขา 3 และ อพ2 31.85 Kohm. จะต่ออยู่ที่ขา 5 ร่วมกับคาปาซิเตอร์ C2 ที่ทำหน้าที่ BYPASS ให้กับวงจร ส่วน R1 และ R2 จะทำหน้าที่เป็นตัวจัดเกนการขยายให้กับ OP-AMP. ตัวแรก R4 และ R3 จะจัดเกนให้กับ OP-AMP. ตัวถัดมา โดยต่ออยู่ในลักษณะ NON-INVERTING AMP. ดังรูปที่ 4.4





Title		RECIVER	
Size	Number	Revision	
A4	KMITL01	A	
Date:	26-OCT-1995	Sheet 1 of PROJEC	
File:	PROJECT01/1	Drawn By: P. SOMBUT	

รูปที่ 4.5 วงจร AM RECEIVER

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานหรือการเผยแพร่โดยไม่ได้รับอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของภาครับ AM

การทำงานเริ่มจาก IC MK 484 จะทำหน้าที่เป็นภาครับ โดยมีวาริเอเบิลและคอสส์ OSC. เป็นตัวปรับความถี่ของสถานี OUTPUT ของไอซี จะต่อผ่าน C3 เพื่อส่งเข้าขยายโดย TR1 และ TR2 ส่งออกลำโพงต่อไป คอสส์ OSC ใช้แบบสีแดง วาริเอเบิลใช้ของธรรมดาทั่วไป เอ้าท์พุทต่อเข้าหูฟังไดนามิคหรือต่อกับลำโพงเล็กก็ได้ แต่ต้องเป็น 30 โอห์มขึ้นไป จุด ANT มีไว้สำหรับต่อสายอากาศ โดยใช้สายไฟขนาดเล็กยาวประมาณ 1 เมตร ต่อเป็นสายอากาศ วงจรนี้ใช้ไฟ 3 โวลท์ IC LPF 351 เป็นวงจรเปรียบเทียบ (COMPARATOR) ที่ขา 3 จะมีค่าของแรงไฟตรงประมาณ 2.7 V เพื่อเป็นค่าอ้างอิงของสัญญาณที่เข้ามา ผลลัพธ์จะปรากฏที่ขา 6 ซึ่งจะได้รูปแบบสัญญาณเหมือนกับด้านส่งทุกประการ

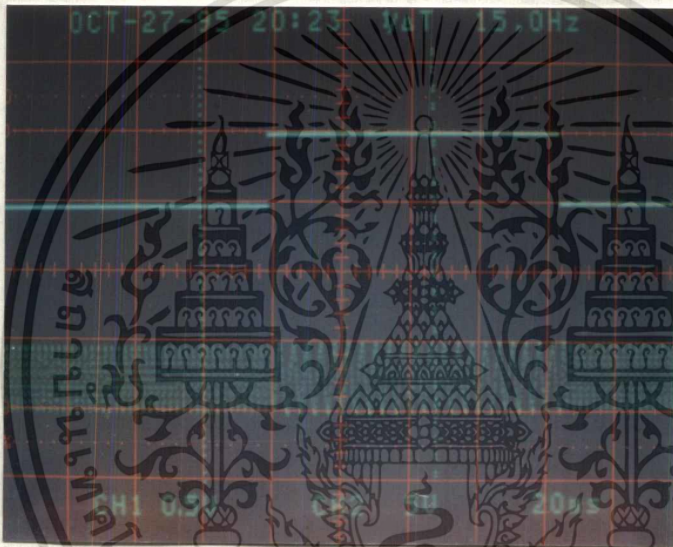


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

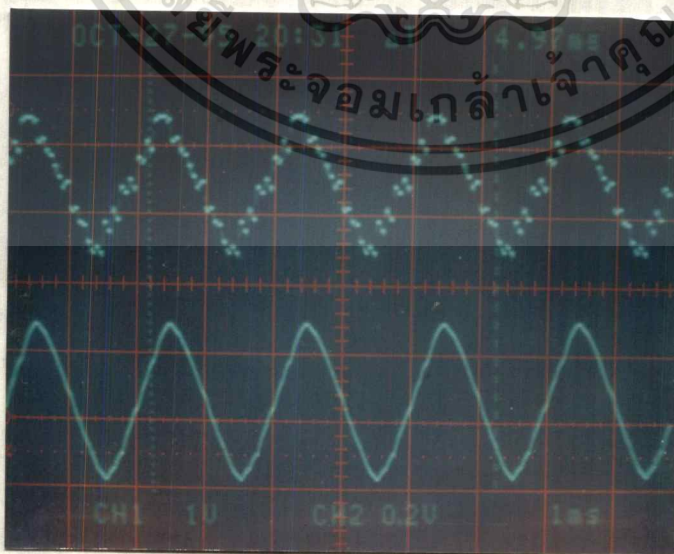
บทที่ 5 ผลการทดลองและสรุปผล

ผลการทดลองของภาคส่ง

เมื่อป้อนข้อมูลเข้าที่ ENCODER โดยผ่าน PC. ซึ่งข้อมูลที่ออกจาก ENCODER จะกำหนดให้มีความเร็ว 12 bit/sec. สัญญาณจะถูกนำมารวมกับสัญญาณจากภาค OSC. (1) ทำการ MODULATION แบบ FSK ภาคส่งเคราะห์ความถี่จากสัญญาณที่เป็น DIGITAL ให้เป็นสัญญาณ SINE WAVE ส่งผลให้เกิดความถี่ SINE WAVE เป็น 2 ค่าคือ 250 Hz สำหรับข้อมูลที่มีสถานะเป็น '1' และ 500 Hz สำหรับข้อมูลที่เป็น '0' ดังรูปแสดง และเมื่อสัญญาณ SINE WAVE 2 ค่านี้ จะถูกกรองความถี่ที่วงจรถ่าย BPF เพื่อให้ได้รูปร่างของ SINE WAVE ที่สมบูรณ์ เพื่อส่งเข้าไป MODULATE กับ CARRIER ของภาคส่งแบบ AMPLITUDE MODULATION เพื่อกระจายเสียงออกอากาศต่อไป



รูปที่ 5.1 เปรียบเทียบสัญญาณข้อมูลกับความถี่ของสัญญาณที่ส่ง



รูปที่ 5.2 สัญญาณ SINE WAVE จาก WALSH FUNCTION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลองของภาครับ

สัญญาณที่รับได้จากสายอากาศของเครื่องรับวิทยุกระจายเสียงแบบ AM จะผ่านภาค AM MODULATION และขยายให้มี AMPLITUDE สูงขึ้นที่ภาค AF AMP. (LF 351) จากนั้นจะผ่านเข้าสู่วงจรกรองความถี่ SINE WAVE อยู่ในย่าน 250 - 500 Hz ดังรูปที่ภาค FSK DEMODULATOR จะทำการแปลงความถี่ 250 Hz และ 500 Hz ออกมาให้เป็นสัญญาณ DIGITAL ที่มีสถานะเป็น '1' และ '0' ตามลำดับ ตามค่าการส่งสัญญาณในอัตรา 12 บิตต่อวินาที เพื่อส่งสู่ภาค DECODE DATA ต่อไปดังรูป



รูปที่ 5.3 แสดงสัญญาณของภาครับ

สรุปผลการทดลอง

จากผลการทดลองปรากฏว่า การส่งสัญญาณผ่านภาคส่งนั้นสามารถทำงาน ได้ตามจุดประสงค์ ทุกประการ ส่วนภาครับสัญญาณจากเครื่องรับนั้น ถ้าเครื่องรับไม่มีประสิทธิภาพดีพอ ก็จะไม่ สามารถรับสัญญาณที่ส่งมาได้ดี เพราะการรบกวนกันและกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป

ในระบบการสื่อสารปัจจุบัน ได้มีการนำเอาเทคโนโลยีต่าง ๆ มาใช้งาน เพื่อให้บรรลุถึงความต้องการของผู้ใช้ ดังนั้นโครงการนี้ก็จะเป็นอีกวิธีการหนึ่งของการสื่อสารข้อมูล ที่จะแสดงให้เห็นถึงการส่งข้อมูล ในรูปแบบการกระจายของคลื่นวิทยุ ที่มีความแม่นยำสูง ทำให้ผู้รับข่าวสารมีความเข้าใจถึงข้อมูลที่ส่งออกมา ได้อย่างชัดเจน ซึ่งถูกแสดงออกมาในรูปแบบ ของจอแสดงผล หรือจอคอมพิวเตอร์ ประกอบกับการทำงานของระบบไมโครคอนโทรลเลอร์และไมโครคอมพิวเตอร์ ที่ถูกออกแบบระบบการทำงานของโปรแกรมไว้เป็นอย่างดี ทำให้เราสามารถที่จะนำเอาข้อมูล ต่าง ๆ ที่ถูกส่งออกไปเรียกกลับมาดูได้ตามต้องการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

ผู้จัดทำขอขอบพระคุณท่านอาจารย์ที่ปรึกษา อาจารย์กฤตดากร กล่อมการ
ซึ่งให้คำปรึกษาและสนับสนุนในการทำโครงการนี้เป็นอย่างดี รวมทั้งขอขอบคุณผู้รวบรวม
และจัดพิมพ์รูปเล่ม มา ณ. ที่นี้ด้วย



เอกสารอ้างอิง

ดร. ประสิทธิ์ ประพัฒน์มงคลการ : ‘ หลักการระบบสื่อสาร ‘ SCIENCE, ENGINEERING & EDUCATION CO.,LTD.

สุชาติ กังวารจิตต์ : ‘ เครื่องรับส่งวิทยุและระบบสื่อสาร ‘ SCIENCE, ENGINEERING & EDUCATION CO.,LTD.

VIJAY AHUJA : ‘ DESIGN AND ANALYSIS OF COMPUTER COMMUNICATION NETWORK , ‘

MC GRAW - HILL INTERNATIONAL BOOK COMPANY , SINGAPORE , 1985



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

-FLOW CHART ภาคส่ง

-โปรแกรมภาษา C ภาครับและภาคส่ง

-ภาษา ASSEMBLY

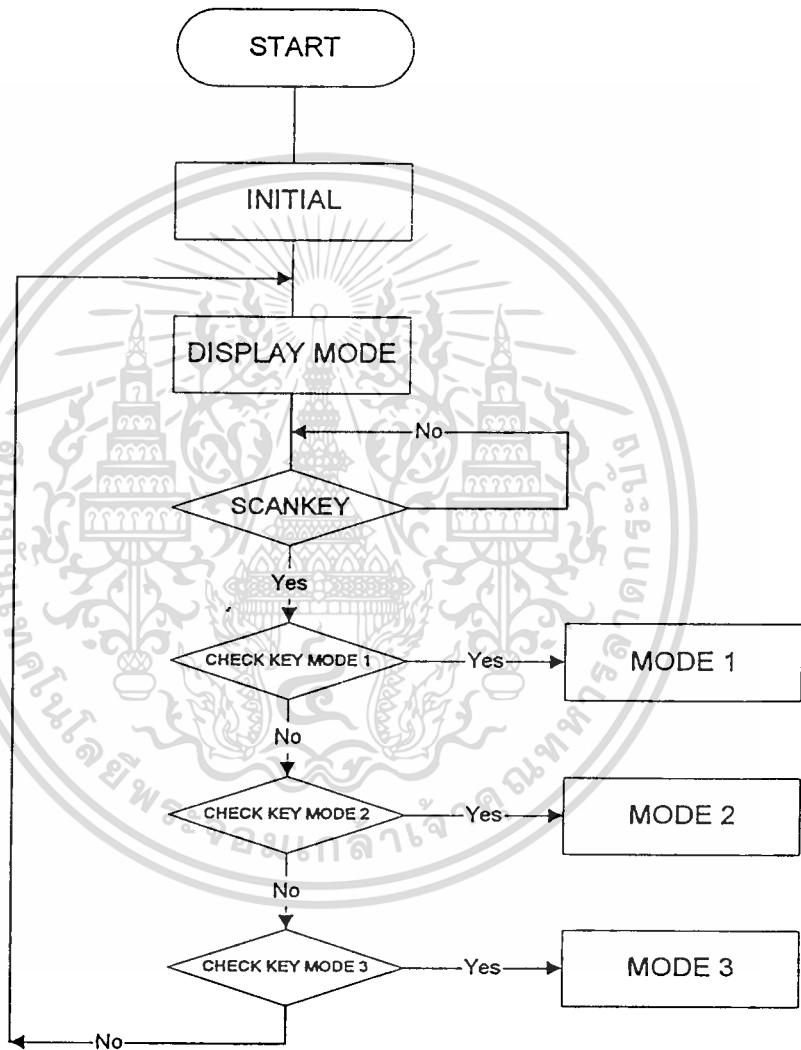
-FLOW CHART ภาคส่ง

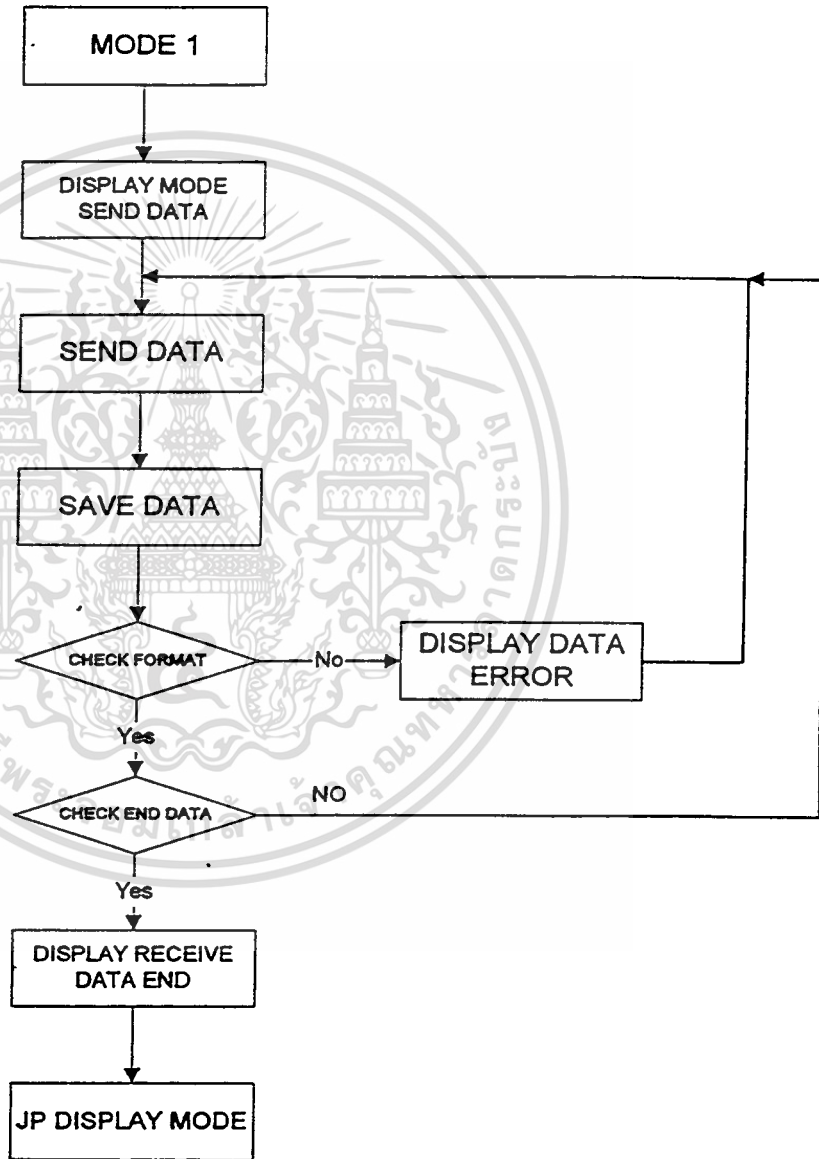
-ภาษา ASSEMBLY

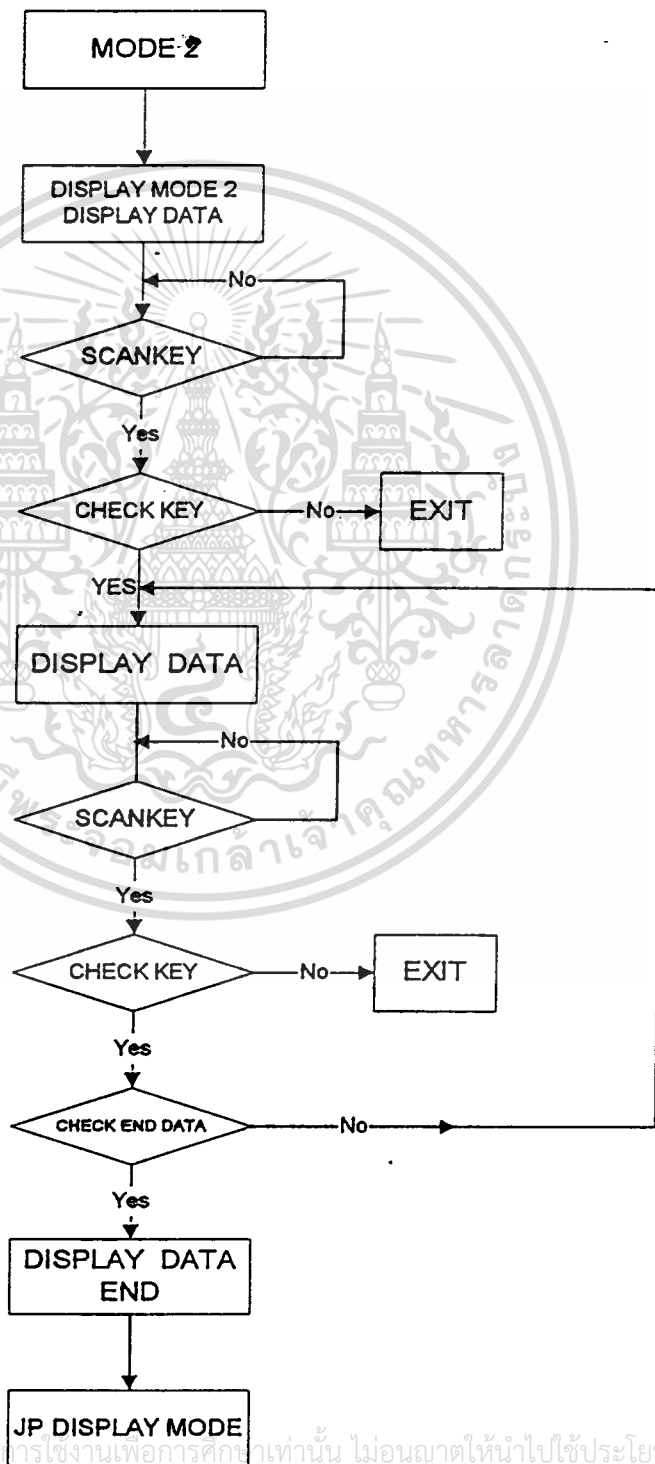


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

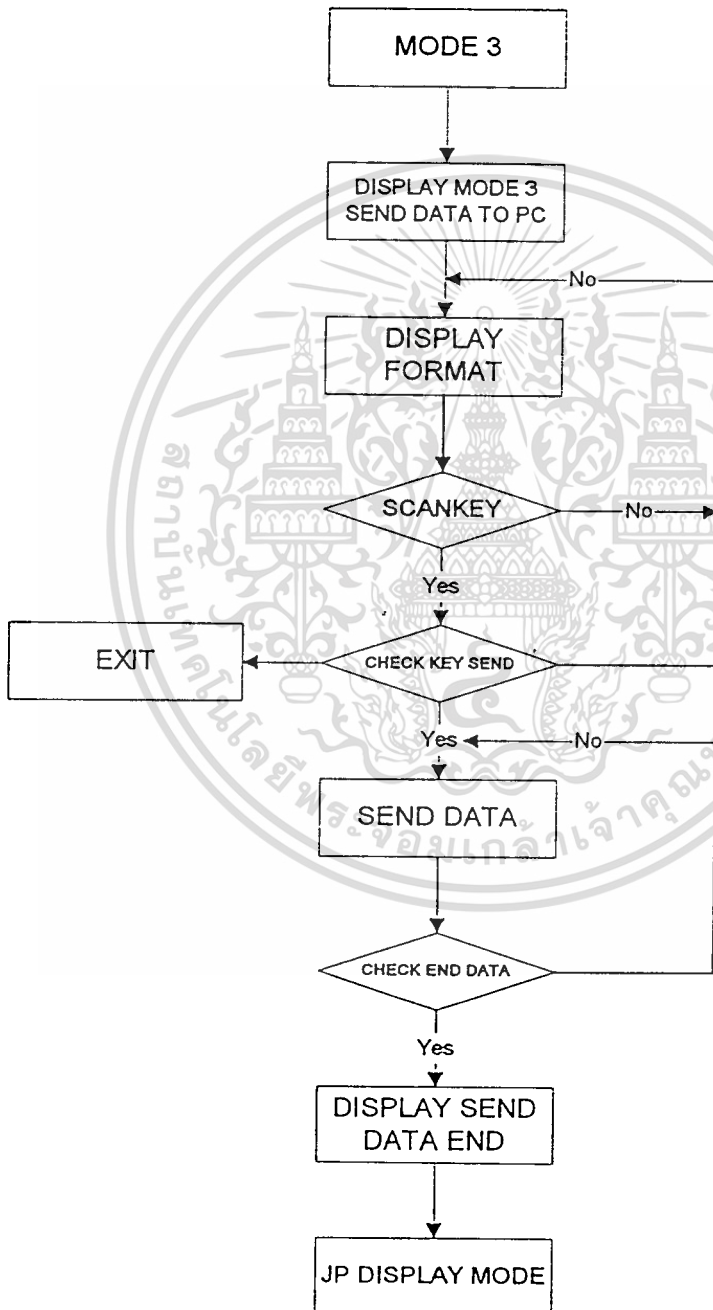
FLOW CHART (TRANSMIT DATA)







เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```
/* program send and receive data */
```

```
#include "conio.h"
```

```
#include "dos.h"
```

```
#include "stdio.h"
```

```
#include "stdlib.h"
```

```
main()
```

```
{
```

```
    buad_rate();
```

```
    line_control();
```

```
    clrscr();
```

```
    menu();
```

```
}
```

```
key_data()
```

```
{
```

```
char m[2000],name[20],s;
```

```
int n=0,x,y,o=0;
```

```
FILE *fp;
```

```
    clrscr();
```

```
    printf("\n\n*** TRANSMITRECEIVE DATA PROGRAM ***\n");
```

```
    printf("IF END DATA KEY (shift + 3) ----> [#] \n");
```

```
    printf("IF EXIT PROGRAM KEY ----> [ESC] \n");
```

```
    box(4,4,78,23);
```

```
    window(5,5,77,22);
```

```
while(n<=2000){
```

```
    cprintf(" \a");
```

```
geta:    m[n]=getche();
```

```
    if(m[n]==0x0d)cprintf("\n");
```

```
    if(m[n]==0x23)goto tx;
```

```
    if(m[n]==0x1b)goto dx;
```

```
    if((m[n]==0x08)||(m[n]==0x00)){
```

```
        n--;
```

```
        goto geta;
```

```
    }
```

```
    n++;
```

```
    o++;
```

```
    goto geta;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tx:  o=0;

      n=0;
      clrscr();
      delay(100);
      cprintf("\a");

tx1:  outputport(0x3f8,m[n]);      /* out data com2(2f8) */
      if(m[n]==0x0d)cprintf("\n");
      cprintf("%c",m[n]);
      if(m[n]==0x23)break;
      if(kbhit())goto dx;
      delay(30);
      n++;
      o++;
      goto tx1;
}

send: cprintf("\n");
      cprintf("DATA SEND END\n");
      delay(1000);
      cprintf("\n");
      cprintf("### EXIT KEY ENTER ###\n");
      cprintf("SAVE DATA WRITE TO FILENAME : ");
      gets(mame);
      if((fp = fopen(mame,"w")) == NULL){
          goto dx;
      }

      n=0;
      o=0;
      while(m[n]!=(0x23)) {
          putc(m[n],fp);
          n++;
      }

      fclose(fp);
      cprintf("\n");
      cprintf("SAVE DATA COMPLETE\n");
dx:  cprintf("\n");
      cprintf("EXIT SEND DATA PROGRAM\n");
      delay(1000);

```

```

receive_data()
{
char ch[2000],fname[20],s;
int i=0,x,y,j=0,d;
FILE *fp;

clrscr();
printf("\n\n*** RECEIVE DATA PROGRAM ***\n");
printf("IF EXIT PROGRAM PRESS ANY KEY  \n\n");
    box(4,4,78,23);
    window(5,5,77,22);

while (j<=2000){

if(kbhit()){
    goto ex;
}
rec: if((inport(0x3fd)&0x01)==1) {
    ch[i]=inport(0x3f8);    /* inport data com1(3f8)*/

    if(ch[i]==(0x23)) goto endd;
    if(ch[i]==(0x0d))cprintf("\n");
    if(ch[i]==(0x00)) {
        i--;
        goto rec;
    }

    cprintf("%c",ch[i]);
        i++;
        j++;
        goto rec;
    }
    if(kbhit())goto ex;

}

endd:
cprintf("\n");

```

```

cprintf("RECEIVE DATA END\n");
cprintf("### EXIT KEY ENTER ### \n");
cprintf("SAVE DATA WRITE TO FILENAME : ");
gets(fname);
if((fp = fopen(fname,"w")) == NULL){
    goto ex;
}

```

```

i=-3;
j=0;
while(ch[i!=(0x23)) {
    puts(ch[i],fp);
    i++;
}

```

```

fclose(fp);
cprintf(" \n");
cprintf("SAVE DATA COMPLETE \n");

```

```

ex: cprintf(" \n");
cprintf("EXIT RECEIVE DATA PROGRAM \n");
delay(1000);

```

```

buad_rate()
/* determind buad rate */
{
    outport(0x3fb,0x80); /* set DLAB=1 */
    outport(0x3f8,0x18);
    outport(0x3f9,0x00); /* set baud rate 4800 */
}

```

```

line_control() /* determind parameter line control */
{
    outport(0x3fb,0x3f); /* set line control REG. */
} /* even parity,2 stop bit,word length 8 bit*/

```

```

box(a,b,c,d)
int a,b,c,d;
{
    int i=0;

```

```

gotoxy(a,b);

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=a;i<=c;i++)
putch(196);
gotoxy(a,d);
for(i=a;i<=c;i++)
    putch(196);
for(i=b;i<=d;i++){
    gotoxy(a,i);
    putch(179);
    gotoxy(c,i);
    putch(179);
}
gotoxy(a,b);putch(218);
gotoxy(c,b);putch(191);
gotoxy(c,d);putch(217);
gotoxy(a,d);putch(192);
}
menu()
{
char c;
clrscr();
for(;;){
window(1,1,80,25);
clrscr();
box(28,9,55,16);
gotoxy(30,11);
printf(" T : TRANSMIT DATA\n");
gotoxy(30,12);
printf(" R : RECEIVE DATA\n");
gotoxy(30,13);
printf(" Q : EXIT PROGRAM\n");
gotoxy(30,14);
printf(" PLEASE SELECT :_");
c=getche();
printf("\n");
if(c=='t' || c=='T')key_data();
else if(c=='r' || c=='R')receive_data();
else if(c=='q' || c=='Q'){
window(1,1,80,25);
clrscr();
break;
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

*****
**      PROGRAM TRANSMIT      **
**      DATA FROM BOARD     **
**      CP Z84C11            **
*****

```

```

BEGIN      EQU      6000H

```

```

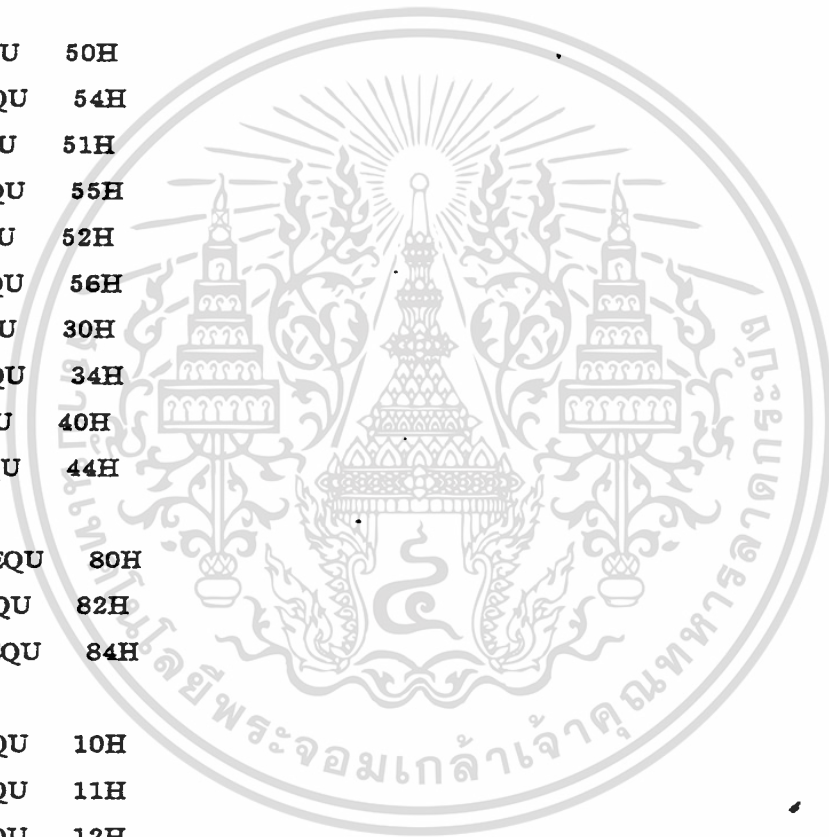
ORG      6000H

```

```

;
PA:      EQU      50H
PAC:     EQU      54H
PB:      EQU      51H
PBC:     EQU      55H
PC:      EQU      52H
PCC:     EQU      56H
PD:      EQU      30H
PDC:     EQU      34H
PE:      EQU      40H
PEC:     EQU      44H
;
PDATA:   EQU      80H
PSING:   EQU      82H
PREAD:   EQU      84H
;
CTC0:    EQU      10H
CTC1:    EQU      11H
CTC2:    EQU      12H
CTC3:    EQU      13H
;
RTC:     EQU      0A0H
;
RST38:   EQU      9F93H
RXBYTE:  EQU      81H
TXBYTE:  EQU      82H
TXSTG:   EQU      85H
DELAY:   EQU      93H
SYSCALL: EQU      10H
BS:      EQU      8H

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AUTOROM: EQU 7FFFH
KBUFF: EQU 0E000H
STAKEY: EQU 0E001H
SRUN: EQU 0E002H
ON: EQU 22H
OFF: EQU 21H
F1: EQU 0EH
F2: EQU 05H
F3: EQU 00H
F4: EQU 03H
F5: EQU 0AH
F6: EQU 04H
F7: EQU 07H
F8: EQU 02H
F9: EQU 0BH
F10: EQU 06H
F11: EQU 0FH
F12: EQU 01H

```

```

*****
INITIAL LCD
*****

```

```

INITIAL: LD A,10000000B ;0 INPUT , 1 OUTPUT
OUT (PCC),A ;SET PC7 OUTPUT
LD A,00H
OUT (PAC),A ;SET PA INPUT
OUT (PBC),A ;SET PB INPUT
OUT (PDC),A ;SET PD INPUT
ST1: LD A,38H ;FUNC SET 38H
OUT (PDATA),A ;DL - 1 :8 BIT,N - 1:
CALL DELAY2 ;1/16 DUTY, F - 0:7*7
;
LD A,0FH ;DISPLAY ON-OFF
OUT (PDATA),A ;D - 1 OFF,C - 1 CURSOR ON ,B - 1 BLINK
CALL DELAY2
;
LD A,06H ;ENTRY MODE SET
OUT (PDATA),A ;L/D - 1 INCREMENT ,S - 0 RIGHT
CALL DELAY2

```

```

LD A,01H ;CLEAR DISPLAY
OUT (PDATA),A
LD A,00H ;SET PC '0'
OUT (PC),A
CALL DELAY2

```

```
*****
```

```
* SELECT FUNCTION *
```

```
*****
```

```

MODE: CALL CLRLCD
CALL ALARM
LD HL,TAB1 ;PROJECT KMITL
CALL LINE1
LD HL,TAB2 ;TELECOM 3U
CALL LINE2
CALL BEEP
CALL CLRLCD
LD HL,TAB3 ;DATA TRANSMITION
CALL LINE1
LD HL,TAB4 ;EQUIPMENT
CALL LINE2
CALL BEEP

MODEO: CALL CLRLCD
LD HL,TAB5 ;SELECT MODE
CALL LINE1
CALL BEEP
LD HL,TAB6 ;1.RECEIVE DATA FORM PC.
CALL LINE2
CALL BEEP
LD HL,TAB7 ;2.DISPLAY DATA IN MEMORY
CALL LINE2
CALL BEEP
LD HL,TAB8 ;3.TRANSMIT DATA
CALL LINE2
CALL BEEP
LD HL,TAB9 ;12.EXIT PROG
CALL LINE2
CALL BEEP

```

```

LD HL,TAB23
CALL LINE2
CALL SCANK
CALL BEEP
LD A,(KBUFF)
CP F1
JP Z,MODE1 ;RECEIVE DATA FORM PC
CP F2
JP Z,MODE2 ;DISPLAY DATA
CP F3
JP Z,MODE3 ;SEND DATA
CP F12
JP Z,EXIT
JP MODE0

```

```
*****
```

```
* RECEIVE DATA *
* FORM PC. *
```

```
*****
```

```

MODE1: CALL CLRLCD
CALL BEEP
LD HL,TAB10 ;MODE 1
CALL LINE1
LD HL,TAB11 ;RECEIVE DATA FORM PC.
CALL LINE2
CALL BEEP
CALL CLRLCD2
LD HL,TAB22 ;FORMAT DATA 48,8,E,2
CALL LINE2
CALL PCIN ;RECEIVE DATA

MODE10: LD HL,TAB12 ;RECEIVE DATA COMPLETE
CALL LINE2
CALL ALARM
JP MODE0

```

```
DAT DEFW (EA00H)
```

```
*****
```

```
* GET CHARECTER *
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* FORM PC *

```

TBUT:   DS    1
PCIN:   LD    A,(B48)
        LD    (TBUT),A

        LD    A,00100000B    ;0 INPUT,1 OUTPUT
        OUT   (PEC),A

        PUSH  HL
        PUSH  IX
        LD    HL,(1000H)
        LD    IX,(DAT)
PCIN1:  CALL  PCIN2    ;RXREADY
        LD    A,D
        LD    (IX),A
        CP    23H
        JP    Z,MODE10
        INC  IX
        DEC  HL
        LD    A,H
        OR   L
        JP   NZ,PCIN1
        POP  IX
        POP  HL
        RET

```

* RX BYTE *

```

PCIN2:  IN    A,(PE)
        BIT   6,A
        JR   NZ,PCIN2
        NOP
        CALL PCDL    ;START

        PUSH BC
        LD   D,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD B,8
PCIN21: IN A,(PE)
RLA
RLA
RR D
CALL PCDL
DJNZ PCIN21
CALL PCDL ;PARITY
CALL PCDL ;STOP
CALL PCDL ;STOP
POP BC
RET
;
PCDL: LD A,(TBUT)
PCDL1: DEC A
JR NZ,PCDL1
RET
;
*****
* DISPLAY DATA *
* FORM MEMORY *
*****
MODE2: CALL CLRLCD
CALL BEEP
LD HL,TAB13 ;MODE 2 DISPLAY DATA
CALL LINE1
LD HL,TAB14 ;PRESS ANY KEY
CALL LINE2
CALL SCANK
LD A,(KBUFF)
CP F12
JP Z,EXIT
LD HL,(DAT)
MODE20: CALL CLRLCD
CALL LINE21
CALL LINE22
PUSH HL
CALL SCANK
LD A,(KBUFF)
CP F12
JP Z,EXIT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP HL
JP MODE20 ;DISPLAY UNTILL END

```

```

DISE: CALL SCANK
CALL CLRLCD
LD HL,TAB15 ;END DATA
CALL LINE2
CALL ALARM
JP MODE0

```

```

; *****

```

```

; * TRANSMIT DATA *

```

```

; * PORTC #7 *

```

```

; *****

```

```

MODE3: CALL CLRLCD

```

```

CALL BEEP

```

```

LD HL,TAB16 ;MODE 3

```

```

CALL LINE1

```

```

LD HL,TAB17 ;SEND DATA

```

```

CALL LINE2

```

```

CALL CLRLCD

```

```

LD HL,TAB18 ;PRESS KEY (9)

```

```

CALL LINE1

```

```

LD HL,TAB19 ;FOR SEND

```

```

CALL LINE2

```

```

CALL SCANK

```

```

LD A,(KBUFF)

```

```

CP F11

```

```

JP Z,SENDD ;SEND DATA

```

```

LD A,(KBUFF)

```

```

CP F12

```

```

JP Z,EXIT ;EXIT PROG

```

```

JP MODE3

```

```

; *****

```

```

; * MODE 3 *

```

```

; * SUB FOR SEND DATA *

```

```

; *****

```

```

;FORMAT DATA : START * D0 D1 D2 D3 D4 D5 D6 D7 * PARITY STOP STOP

```

```

; (1) ***** (E) (1) (1)

```

```

;OUT DATA --> PC7

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SENDD:    CALL  CLRLCD
          LD   A,80H
          OUT  (PCC),A
          LD   HL,TAB20      ;SEND DATA NOW
          CALL LINE2
          LD   IX,(DAT)
SENDD1:   LD   B,08H      ;DATA 8 BIT
          LD   D,A
; $START BIT
          LD   A,0FFH      ;START BIT
          OUT  (PC),A
          CALL BIT1        ;DELAY 1 BIT
          LD   D,(IX)
          LD   E,D
;
;
SDATA:    LD   A,D      ;SEND DATA
          OUT  (PC),A
          LD   D,A
          RLC  D
          CALL BIT1
          DJNZ SDATA      ;SEND DATA 8 BIT COUNTER 'B'
;
PARITY:   LD   A,E
          AND  A
          JP   PE,EVEN    ;P - 1
ODD:     LD   A,0FFH      ;P - 0
          OUT  (PC),A
          CALL BIT1
          JR   STOPB
EVEN:    LD   A,00H      ;P - 1
          OUT  (PC),A
          CALL BIT1
;
STOPB:   LD   A,0FFH
          OUT  (PC),A
          CALL BIT1      ;STOP BIT
          CALL BIT1

```

เอกสารนี้ LD เอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
SENDD:    CALL  CLRLCD
          LD   A,80H
          OUT  (PCC),A
          LD   HL,TAB20      ;SEND DATA NOW
          CALL LINE2
          LD   IX,(DAT)
SENDD1:   LD   B,08H        ;DATA 8 BIT
          LD   D,A
;START BIT
          LD   A,0FFH      ;START BIT
          OUT  (PC),A
          CALL BIT1        ;DELAY 1 BIT
          LD   D,(IX)
          LD   E,D
;
;
SDATA:    LD   A,D          ;SEND DATA
          OUT  (PC),A
          LD   D,A
          RLC  D
          CALL BIT1
          DJNZ SDATA       ;SEND DATA 8 BIT COUNTER 'B'
;
;
PARITY:   LD   A,E
          AND  A
          JP  PE,EVEN      ;P = 1
;
;
ODD:      LD   A,0FFH      ;P = 0
          OUT  (PC),A
          CALL BIT1
          JR  STOPB
;
;
EVEN:     LD   A,00H       ;P = 1
          OUT  (PC),A
          CALL BIT1
;
;
STOPB:    LD   A,0FFH
          OUT  (PC),A
          CALL BIT1        ;STOP BIT
          CALL BIT1
          LD   A,00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUT (PC),A
```

```
*****
```

```
CHECK END SEND DATA
```

```
*****
```

```
CALL DELAY3
```

```
INC IX
```

```
LD C,(IX-1)
```

```
LD A,C
```

```
CP 23H
```

```
JP NZ,SENDD1 ;GO TO SENDD UNTIL DATA END
```

```
CALL CLRLCD
```

```
LD HL,TAB21 ;SEND DATA END
```

```
CALL LINE2
```

```
CALL DELAY3
```

```
CALL ALARM
```

```
JP MODE0
```

```
*****
```

```
SUB RPOGRAM
```

```
*****
```

```
*****
```

```
* DATA TABLE *
```

```
*****
```

```
TAB1 DB " **PROJECT KMITL.** "
TAB2 DB " ##TELECOM 3U.##-> "
TAB3 DB " DATA TRANSMITTER "
TAB4 DB " EQUIPMENT "
TAB5 DB " PRESS SELECT MODE "
TAB6 DB "(1).RX. DATA FORM PC"
TAB7 DB "(2). DISPLAY DATA "
TAB8 DB "(3). SEND DATA --> "
TAB9 DB "(12). EXIT PROGRAM "
TAB10 DB " ** MODE 1 OPERATE **"
TAB11 DB "RECEIVE DATA FORM PC"
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TAB12  DB  'RECEIVE DATA END '
TAB13  DB  'MODE 2 DISPLAY DATA '
TAB14  DB  'PRESS ANY KEY ! '
TAB15  DB  'DISPLAY DATA END '
TAB16  DB  '*** MODE 3 OPERATE ***'
TAB17  DB  'SEND DATA 12 BPS '
TAB18  DB  'PRESS KEY (9) SEND '
TAB19  DB  'FOR SEND DATA '
TAB20  DB  'SEND DATA ## NOW !'
TAB21  DB  'SEND DATA ## END !'
TAB22  DB  'FORMAT DATA 48,8,E,2'
TAB23  DB  '(1)-(2)-(3)-(12) '

```

;

;

```

BEEP:  LD  A,00001111B      ;SP - PE1

```

```

OUT  (PEC),A

```

```

LD  B,30

```

```

LD  A,B

```

```

OUT  (PE),A

```

```

B1:   LD  A,02

```

```

CALL BEEP1

```

```

XOR  A

```

```

OUT  (PE),A

```

```

CALL BEEP2

```

```

DJNZ B1

```

```

LD  A,00100000B

```

```

OUT  (PEC),A

```

```

RET

```

;

```

BEEP1: LD  A,30

```

```

B1:   NOP

```

```

NOP

```

```

DEC  A

```

```

JR  NZ,B1

```

```

RET

```

```

BEEP2: LD  A,60

```

```

B2:   NOP

```

```

NOP

```

```

DEC  A

```

```

JR  NZ,B2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

ALARM: PUSH BC

LD C,50

ALR1: NOP

CALL BEEP

DEC C

JR NZ,ALR1

POP BC

RET

CLRLCD: LD A,00000001B ;CLEAR DISPLAY

OUT (PDATA),A

CALL DELAY2

RET

CLRLCD1: LD C,00

LD A,8DH

RST 10H

LD A,91H

RST 10H

DB "

DB 00H

LD A,8DH

RST 10H

RET

CLRLCD2: LD C,40H

LD A,8DH

RST 10H

LD A,91H

RST 10H

DB "

DB 00H

LD A,8DH

RST 10H

RET

DELAY 1 BIT

BIT: DEFW 4000H ;27B2H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BIT1:    PUSH HL
          LD HL,(BIT)      ;12 BIT /SEC
DELBIT:  DEC HL
          LD A,H
          OR L
          JR NZ,DELBIT
          POP HL
          RET
;
;
;

```

```

LINE1:   LD A,00H
          CALL GOTO
          CALL WRLINE
          CALL DELAY3
          RET

```

```

LINE2:   LD A,40H
          CALL GOTO
          CALL WRLINE
          CALL DELAY3
          RET

```

```

GOTO:    SET 7,A
          OUT (PDATA),A
          CALL READ
          RET

```

```

READ:    IN A,(PREAD)
          BIT 7,A
          JR NZ,READ
          RET

```

```

WRLINE:  LD B,14H

```

```

LOOP:    LD A,(HL)
          LD D,A
          CALL WRBYTE
          INC HL
          DJNZ LOOP
          RET

```

```

WRBYTE:  LD D,A
          OUT (PSING),A
          CALL DELAY2
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
LINE21:    LD    A,00H
           CALL  GOTO
           CALL  WRLINE1
           RET

LINE22:    LD    A,40H
           CALL  GOTO
           CALL  WRLINE1
           RET

WRLINE1:   LD    B,14H

LOOP1:     LD    A,(HL)
           CP    23H
           JP    Z,DISE
           LD    D,A
           CALL  WRBYTE
           INC  HL
           DJNZ LOOP1
           RET

;
DELAY2:    PUSH  BC
           PUSH  AF
           LD    B,10H

DELAY21:   LD    A,50H
           LD    A,C

DELAY22:   DEC  C
           JR   NZ,DELAY22
           DJNZ DELAY21
           POP  AF
           POP  BC
           RET

;
DELAY3:     PUSH  HL
           PUSH  AF
           LD    B,04H

DELAY32:   LD    HL,OFF1FH

DELAY31:   DEC  HL
           LD    A,H
           OR   L
           JR   NZ,DELAY31
           DJNZ DELAY32

```

POP AF

POP HL

RET

* SCANKEY *

SCANKEY: LD A,94H ;SCAN KEY

RST 10H

LD (KBUF),A ;SAVE KEY IN BUFFER

CALL BEEP

CALL CLRLCD

RET

OUT PUT TO CON

B12 DB 00H

B24 DB 81H

B48 DB 3DH

B96 DB 1EH

B192 DB 0DH

AUTO: LD A,0A3H

LD (AUTOROM),A

RST 10H

EXIT: CALL CLRLCD

LD HL,TAB9 ;EXIT PROGRAM

CALL LINE2

CALL ALARM

CALL ALARM

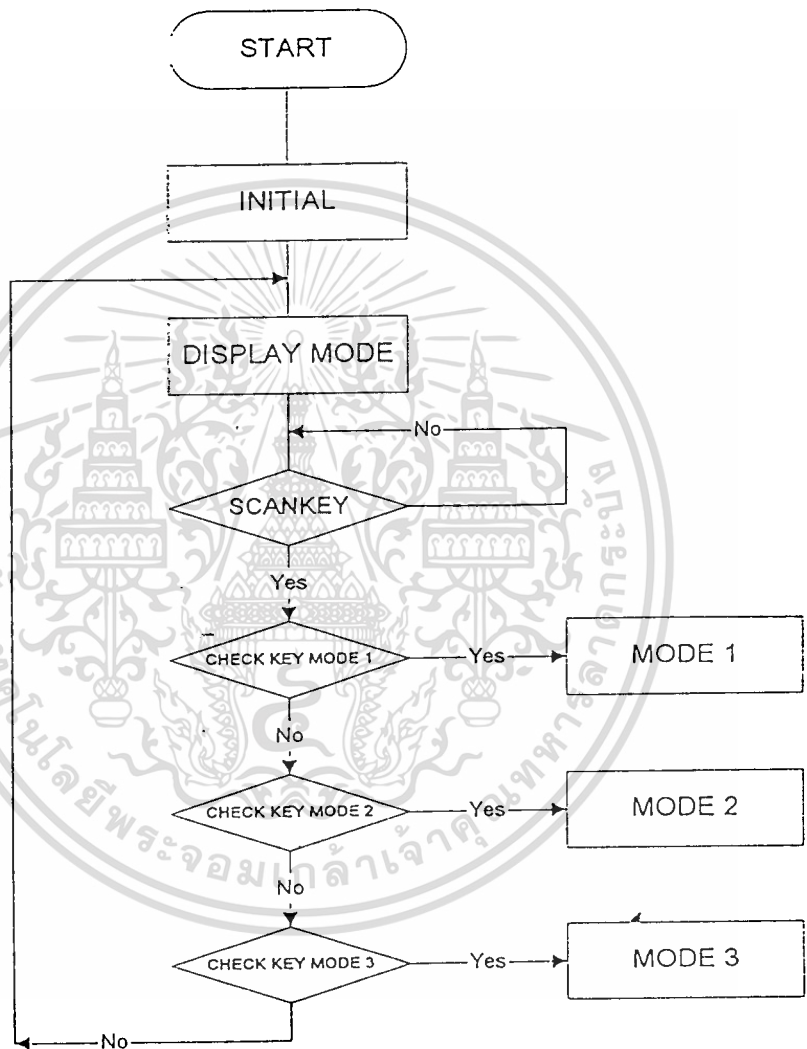
JP MODE0

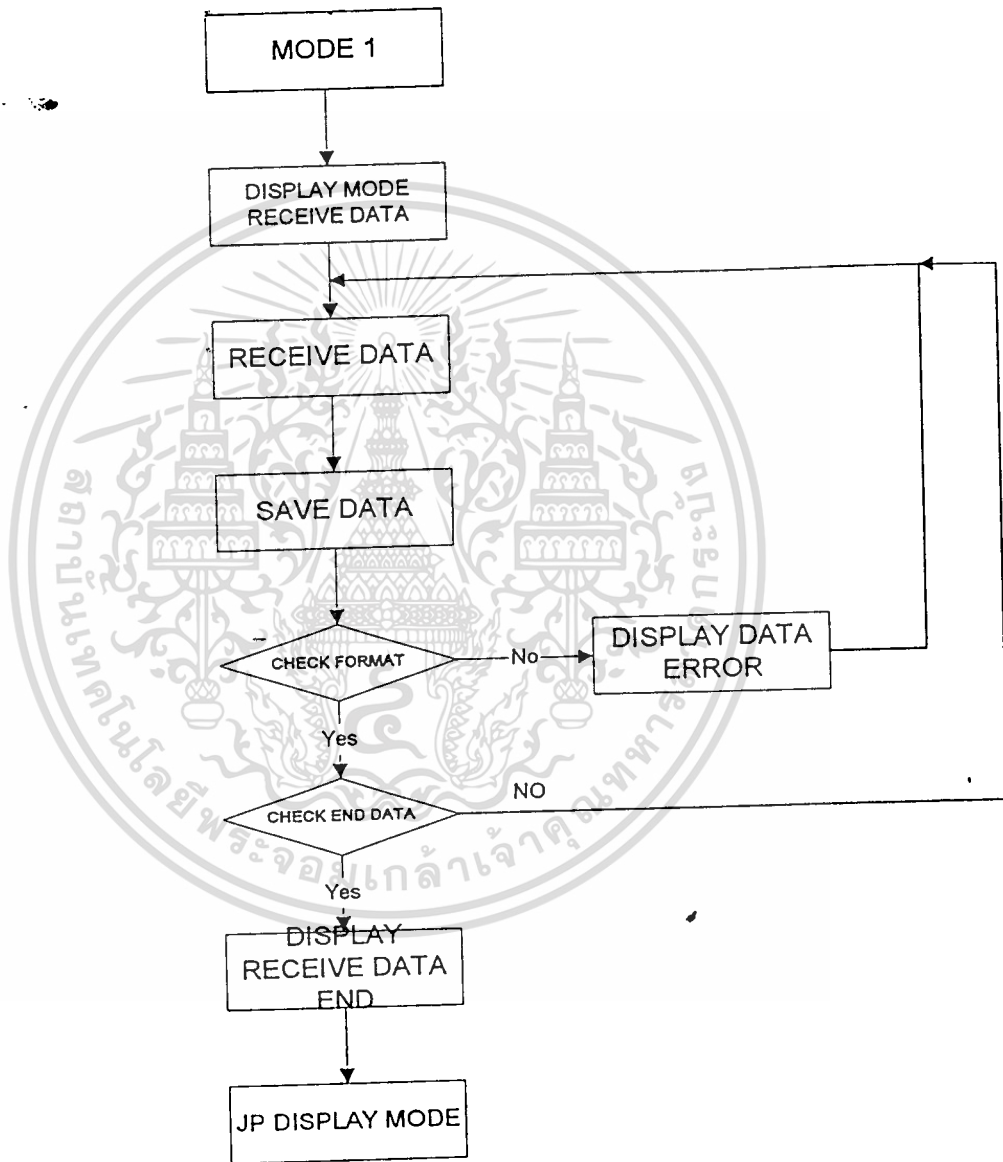
END

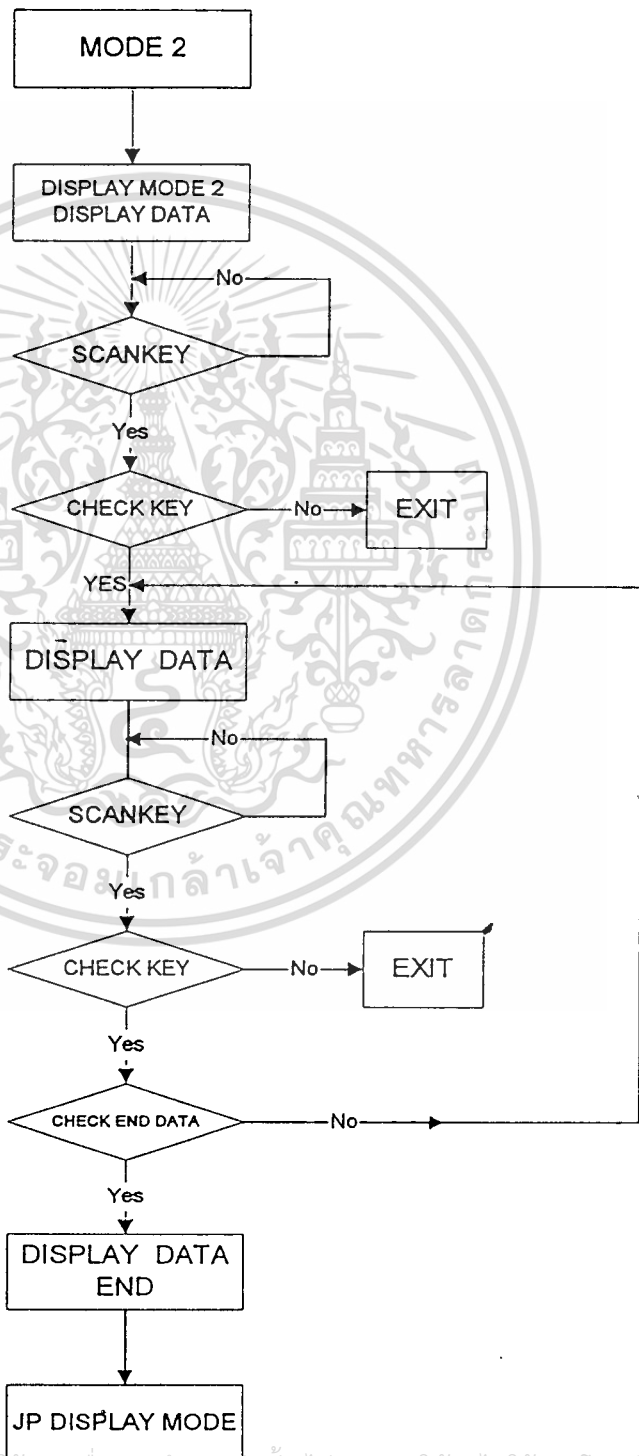
END PROGRAM_TX*****
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

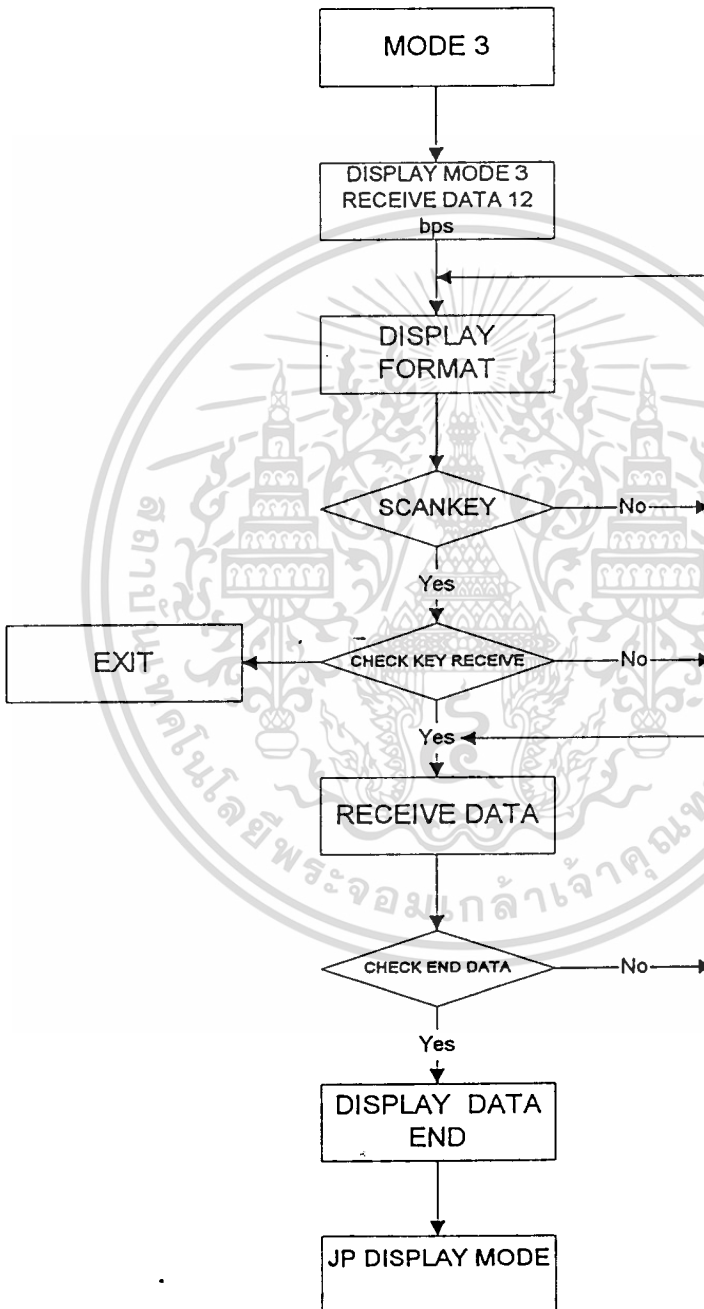
FLOW CHART (RECEIVE DATA)







เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*****
```

```
** PROGRAM RECEIVE **
```

```
** DATA FOR CP-Z84C11 **
```

```
*****
```

```
BEGIN EQU 6000H
```

```
ORG 6000H
```

```
*****
```

```
PA: EQU 50H
```

```
PAC: EQU 54H
```

```
PB: EQU 51H
```

```
PBC: EQU 55H
```

```
PC: EQU 52H
```

```
PCC: EQU 56H
```

```
PD: EQU 30H
```

```
PDC: EQU 34H
```

```
PE: EQU 40H
```

```
PEC: EQU 44H
```

```
PDATA: EQU 80H
```

```
PSING: EQU 82H
```

```
PREAD: EQU 84H
```

```
CTC0: EQU 10H
```

```
CTC1: EQU 11H
```

```
CTC2: EQU 12H
```

```
CTC3: EQU 13H
```

```
RTC: EQU 0A0H
```

```
RST38: EQU 9F93H
```

```
RXBYTE: EQU 81H
```

```
TXBYTE: EQU 82H
```

```
TXSTG: EQU 85H
```

```
DELAY: EQU 93H
```

```
SYSCALL: EQU 10H
```

```
RXDATA: EQU 0A000H
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BS:      EQU   8H
KBUFF:   EQU   0E000H
AUTOROM: EQU   7FFFH
;
;
; *****
;      INITIAL LCD
; *****
;
INITIAL: LD   A,00H      ;0 INPUT , 1 OUTPUT
         OUT  (PAC),A    ;SET PA INPUT
         OUT  (PBC),A    ;SET PB INPUT
         OUT  (PDC),A    ;SET PD INPUT
         LD   A,0FEH
         OUT  (PCC),A    ;SET PC0 INPUT
;
ST1:    LD   A,38H      ;FUNC SET 38H
         OUT  (PDATA),A  ;DL - 1 :8 BIT,N - 1:
         CALL DELAY2     ;1/16 DUTY, F = 0:7*7
;
         LD   A,0FH     ;DISPLAY ON-OFF
         OUT  (PDATA),A  ;D - 1 OFF,C - 1 CURSOR ON ,B - 1 BLINK
         CALL DELAY2
;
         LD   A,06H     ;ENTRY MODE SET
         OUT  (PDATA),A  ;I/D - 1 INCREMENT ,S - 0 RIGHT
         CALL DELAY2
;
         LD   A,01H     ;CLEAR DISPLAY
         OUT  (PDATA),A
         CALL DELAY2
;
;
; *****
; * SELECT FUNCTION *
; *****
;
DAT      DEFW  (EA00H)
MODE:    CALL  CLRLCD
         CALL  ALARM
         LD   HL,TAB1   ;PROJECT KMITL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL LINE1
LD HL,TAB2 ;TELECOM 3U
CALL LINE2
CALL BEEP
CALL DELAY3
CALL CLRLCD
LD HL,TAB3 ;DATA RECEIVER
CALL LINE1
LD HL,TAB4 ;EQUIPMENT
CALL LINE2
CALL BEEP

```

```

MODE0: CALL CLRLCD
LD HL,TAB5 ;PRESS SERECT MODE
CALL LINE1
LD HL,TAB6
CALL LINE2 ;(1). RECEIVE DATA
CALL DELAY3
CALL BEEP
LD HL,TAB7
CALL LINE2 ;(2). DISPLAY DATA
CALL DELAY3
CALL BEEP
LD HL,TAB14 ;(3). TX.DATA TO PC
CALL LINE2
CALL DELAY3
CALL BEEP
LD HL,TAB22 ;(12).EXIT
CALL LINE2
CALL BEEP
LD HL,TAB24
CALL LINE2
CALL ALARM

```

```

CALL SCANK
LD A,(KBUFF)
CP F1
JP Z,MODE1
LD A,(KBUFF)
CP F2
JP Z,MODE2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP   F3
JP   Z,MODE3
CP   F12
JP   Z,EXIT
JP   MODE0

```

```
MODE1:   CALL  CLRLCD
```

```

CALL  BEEP
LD   HL,TAB8      ;MODE 1
CALL  LINE1
LD   HL,TAB9      ;RECEIVE DATA OPERATE
CALL  LINE1
JP   RECB

```

```

;
MODE2:   CALL  CLRLCD

```

```

CALL  BEEP
LD   HL,TAB12     ;MODE 2 DISPLAY DATA
CALL  LINE1
LD   HL,TAB13     ;PRESS ANY KEY
CALL  LINE2
JP   DISPD

```

```

;
MODE3:   CALL  CLRLCD

```

```

LD   HL,TAB15     ;MODE 3
CALL  LINE1
LD   HL,TAB16     ;TX. DATA TO PC
CALL  LINE2
CALL  DELAY3
LD   HL,TAB17     ;FORMAT DATA 48,8,E,2
CALL  LINE2
CALL  DELAY3
LD   HL,TAB25
CALL  LINE2       ;PRESS KEY SEND
CALL  SCANK
LD   A,(KBUF)
CP   F11
JP   Z,MODE31
CP   F12
JP   Z,EXIT
JP   MODE3

```

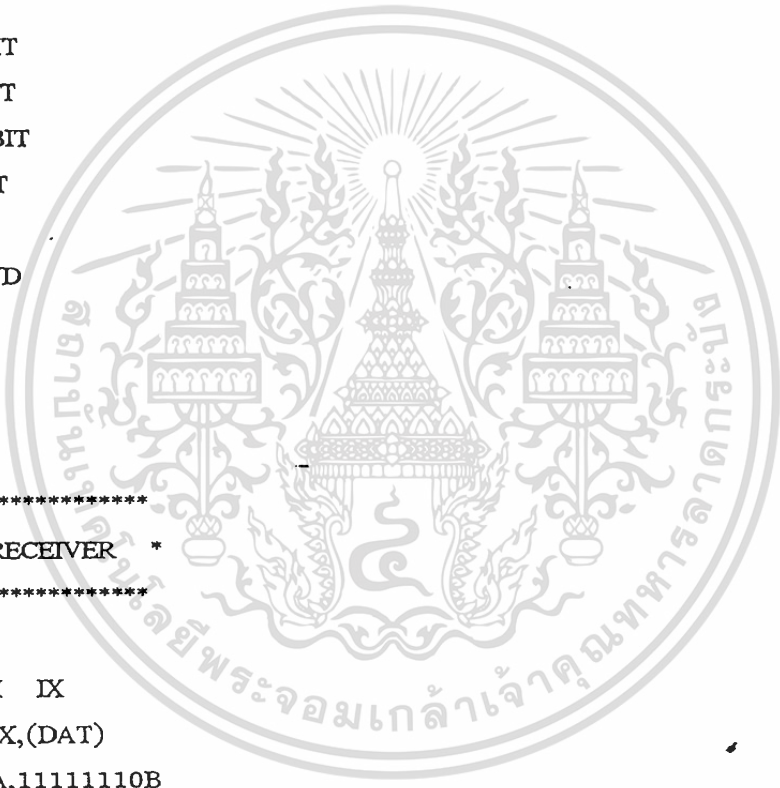
```
MODE31:  LD   HL,TAB21     ;SEND DATA NOW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL LINE2
CALL BOUT
BEND: LD HL,TAB18 ;TRANSMIT DATA END
CALL LINE2
CALL ALARM
JP MODE0
;
;
;*****
; * MODE 1 *
; * RECEIVE DATA *
;*****
;
;
;1.CHECK START BIT
;2.READ DATA 8 BIT
;3.CHECK PARITY BIT
;4.CHECK STOP BIT
;5.SAVE DATA
;6.CHECK DATA END
;7.BEEP
;
;PC0 = INPUT
;
;*****
; * HOLD RECEIVER *
;*****
;
;
RECB: PUSH IX
LD IX,(DAT)
LD A,11111110B
OUT (PCC),A
NOP
;
;*****
; CHECK START BIT
;*****
;
;
CHKS: IN A,(PC)
NOP
NOP
NOP

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NOP
BIT 0,A
JP Z,CHKS
CALL BIT1
CALL BIT.5

```

;

```
READD: LD D,00H
```

```
LD E,08H
```

```
DATAIN: XOR A
```

```
IN A,(PC)
```

```
NOP
```

```
NOP
```

```
NOP
```

```
NOP
```

```
BIT 0,A
```

```
JR Z,SPACE ;JUMP IF INPUT '0'
```

```
MARK: SET .0,D
```

```
JR CHKF
```

```
SPACE: RES 0,D
```

```
CHKF: CALL BIT1
```

```
RL D
```

```
DEC E
```

```
JR NZ,DATAIN
```

```
;OK DATA 8 BIT
```

```
RR D
```

```
LD (IX),D ;SAVE DATA
```

```
LD A,D
```

```
CP 23H
```

```
JP Z,ENDD ;END DATA
```

```
INC IX
```

;

;

```
*****
```

```
* CHECK PARITY BIT *
```

```
*****
```

;

```
PAR: LD A,D
```

```
AND A
```

```
JP PE,EVEN
```

```
ODD: LD C,01H ;P = 0
```

```
JR PAR1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EVEN:      LD    C,00H          ;P = 1
PAR1:      XOR   A
           IN    A,(PC)
           NOP
           NOP
           NOP
           NOP
           XOR   C
           JR    NZ,DATAER
           CALL  BIT1
;
;
; *****
; CHECK STOP BIT
; *****
;
CHKST:     IN    A,(PC)
           NOP
           NOP
           NOP
           NOP
           BIT   0,A
           JR    Z,STERR
           CALL  BIT1
           IN    A,(PC)
           NOP
           BIT   0,A
           JR    Z,STERR1
           CALL  BIT1
           JP    CHKS
;
;
; *****
; * DISPLAY DATA ERROR
; *****
;
DATAER:    LD    HL,TAB1        ;DATA ERROR
           CALL  LINE2
           CALL  BEEP
           CALL  CLRLCD2
           JP    CHKST

```



```

STERR:    LD    HL,TAB23          ;STOPBIT 1 ERROR
          CALL  LINE2
          CALL  BEEP
          CALL  CLRLCD2
          CALL  BIT1
          CALL  BIT1
          JP    CHKS

```

```

;
STERR1:   LD    HL,TAB23          ;STOPBIT 2 ERROR
          CALL  LINE2
          CALL  BEEP
          CALL  CLRLCD2
          CALL  BIT1
          JP    CHKS

```

```

;
ENDD:     CALL  BEEP
          CALL  CLRLCD
          LD    HL,TAB10          ;END RECEIVE DATA
          CALL  LINE2
          CALL  ALARM
          POP   IX
          JP    MODE0

```

```

;
;
; *****
; * MODE 2 DISPLAY DATA *
; *****

```

```

DISPD:    CALL  SCANK
          LD    A,(KBUFF)
          CP    F12
          JP    Z,EXIT
          LD    HL,(DAT)

```

```

DISPD1:   CALL  CLRLCD
          CALL  LINE21
          CALL  LINE22
          PUSH  HL
          CALL  SCANK
          LD    A,(KBUFF)
          CP    F12
          JP    Z,EXIT
          POP   HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JP DISPD1 ;DISPLAY UNTILL END
```

```
DISE: CALL SCANK
```

```
CALL CLRLCD
```

```
LD HL,TAB19 ;END DATA
```

```
CALL LINE2
```

```
CALL ALARM
```

```
JP MODE0
```

```
*****
```

```
* MODE 3 *
```

```
* TX. CHARECTER TO PC. *
```

```
*****
```

```
BOUT: LD A,(B48)
```

```
LD (RBUT),A
```

```
LD A,00100000B ;0 INPUT,1 OUTPUT
```

```
OUT (PEC),A
```

```
LD IX,(DAT)
```

```
PCOUT: PUSH BC
```

```
LD BC,(1000H)
```

```
PCOUT1: CALL PCFM ;RXREADY
```

```
INC IX
```

```
LD A,(IX-2)
```

```
CP 23H
```

```
JP Z,BEND ;END DATA
```

```
CP 00H
```

```
JP Z,BEND
```

```
DEC BC
```

```
LD A,B
```

```
OR C
```

```
JR NZ,PCOUT1
```

```
POP BC
```

```
RET
```

```
*****
```

```
* TX BYTE *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PCFM: LD A,10000000B ;START BIT '0'
      OUT (PE),A
      CALL PCDL ;START
```

```
PUSH BC
LD B,8
LD D,(IX-1)
```

```
PCFM1: XOR A
```

```
RRC D
```

```
RRA
```

```
RRA
```

```
RRA
```

```
SET 7,A
```

```
OUT (PE),A
```

```
CALL PCDL
```

```
DJNZ PCFM1
```

```
JP EDD
```

```
LD D,(IX)
```

```
AND A
```

```
JP PE,EV2
```

```
LD A,00100000B ;PARITY ODD
```

```
OUT (PE),A
```

```
CALL PCDL
```

```
EV2: LD A,00H ;PARITY EVEN
```

```
OUT (PE),A
```

```
CALL PCDL ;PARITY
```

```
EDD: LD A,10100000B
```

```
OUT (PE),A
```

```
CALL PCDL ;STOP '0'
```

```
CALL PCDL ;STOP '0'
```

```
POP BC
```

```
RET
```

```
RBUT: DS 1
```

```
PCDL: LD A,(RBUT)
```

```
SUB 2
```

```
PCDL1: DEC A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STAKEY: EQU 0E001H

SRUN: EQU 0E002H

ON: EQU 22H

OFF: EQU 21H

F1: EQU 0EH

F2: EQU 05H

F3: EQU 00H

F4: EQU 03H

F5: EQU 0AH

F6: EQU 04H

F7: EQU 07H

F8: EQU 02H

F9: EQU 0BH

F10: EQU 06H

F11: EQU 0FH

F12: EQU 01H

BEEP: LD A,00001111B ;SP - PE1

OUT (PEC),A

LD B,50

LD A,B

OUT (PE),A

B1: LD A,02

CALL BEEP1

XOR A

OUT (PE),A

CALL BEEP2

DJNZ B1

LD A,00100000B

OUT (PEC),A

RET

BEEP1: LD A,30

BE1: NOP

NOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC  A
JR   NZ,BE1
RET

```

```

BEEP2:  LD  A,60

```

```

B2:     NOP

```

```

NOP

```

```

DEC  A

```

```

JR   NZ,B2

```

```

RET

```

```

;

```

```

ALARM:  PUSH BC

```

```

LD  C,50

```

```

ALR1:   NOP

```

```

CALL  BEEP

```

```

DEC  C

```

```

JR   NZ,ALR1

```

```

POP  BC

```

```

RET

```

```

;

```

```

;

```

```

CLRLCD: LD  A,00000001B ;CLEAR DISPLAY

```

```

OUT  (PDATA),A

```

```

CALL  DELAY2

```

```

RET

```

```

CLRLCD1: LD  C,00

```

```

LD  A,8DH

```

```

RST 10H

```

```

LD  A,91H

```

```

RST 10H

```

```

DB  "

```

```

DB  00H

```

```

LD  A,8DH

```

```

RST 10H

```

```

RET

```

```

CLRLCD2: LD  C,40H

```

```

LD  A,8DH

```

```

RST 10H

```

```

LD  A,91H

```

```

RST 10H

```

```

DB  "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 00H
LD A,8DH
RST 10H
RET

```

```

*****

```

```

DELAY 1 BIT

```

```

*****

```

```

BIT: DEFW 4000H

```

```

BIT1: PUSH HL

```

```

PUSH DE

```

```

LD HL,(BIT) ;12 BIT /SEC

```

```

DELBIT: DEC HL

```

```

LD A,H

```

```

OR L

```

```

JR NZ,DELBIT

```

```

POP DE

```

```

POP HL

```

```

RET

```

```

*****

```

```

DELAY 1/2 BIT

```

```

*****

```

```

BIT5: DEFW 4500H

```

```

BIT.5: PUSH HL

```

```

PUSH DE

```

```

LD HL,(BIT5) ;DELAY 1/2 BIT,12 BIT /SEC

```

```

DELBIT.5: DEC HL

```

```

LD A,H

```

```

OR L

```

```

JR NZ,DELBIT.5

```

```

POP DE

```

```

POP HL

```

```

RET

```

```

LINE1: LD A,00H

```

```

CALL GOTO

```

```

CALL WRLINE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

```
LINE2:   LD   A,40H
        CALL GOTO
        CALL WRLINE
        RET
```

```
GOTO:   SET  7,A
        OUT  (PDATA),A
        CALL READ
        RET
```

```
READ:   IN   A,(PREAD)
        BIT  7,A
        JR   NZ,READ
        RET
```

```
WRLINE: LD   B,14H
LOOP    LD   A,(HL)
        LD   D,A
        CALL WRBYTE
        INC  HL
        DJNZ LOOP
        RET
```

```
WRBYTE: LD   D,A
        OUT  (PSING),A
        CALL DELAY2
        RET
```

```
;
LINE21: LD   A,00H
        CALL GOTO
        CALL WRLINE1
        RET
```

```
LINE22: LD   A,40H
        CALL GOTO
        CALL WRLINE1
        RET
```

```
WRLINE1: LD   B,14H
LOOP1:   LD   A,(HL)
        CP   23H
        JP   Z,DISE
        LD   D,A
        CALL WRBYTE
        INC  HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DJNZ LOOP1
```

```
RET
```

```
;
```

```
DELAY2:   PUSH  BC
```

```
        PUSH  AF
```

```
        LD    B,20H
```

```
DELAY21:  LD    A,20H
```

```
        LD    A,C
```

```
DELAY22:  DEC   C
```

```
        JR   NZ,DELAY22
```

```
        DJNZ DELAY21
```

```
        POP  AF
```

```
        POP  BC
```

```
        RET
```

```
;
```

```
DELAY3:   PUSH  HL
```

```
        PUSH  AF
```

```
        LD    B,04H
```

```
DELAY32:  LD    HL,0F01FH
```

```
DELAY31:  DEC   HL
```

```
        LD    A,H
```

```
        OR   L
```

```
        JR   NZ,DELAY31
```

```
        DJNZ DELAY32
```

```
        POP  AF
```

```
        POP  HL
```

```
        RET
```

```
*****
```

```
*   SCANKEY   *
```

```
*****
```

```
SCANK:   LD    A,94H   ;SCAN KEY
```

```
        RST  10H
```

```
        LD    (KBUFF),A ;SAVE KEY IN BUFFER
```

```
        CALL BEEP
```

```
        CALL CLRLCD
```

```
        RET
```

```
*****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUT PUT TO CON

```

B12      DB    00H
B24      DB    81H
B48      DB    3DH
B96      DB    1EH
B192     DB    0DH

```

```

EXIT:    CALL  CLRLCD
LD       HL,TAB20      ;***EXIT PROGRAM
CALL    LINE1
CALL    ALARM
JP      MODE0

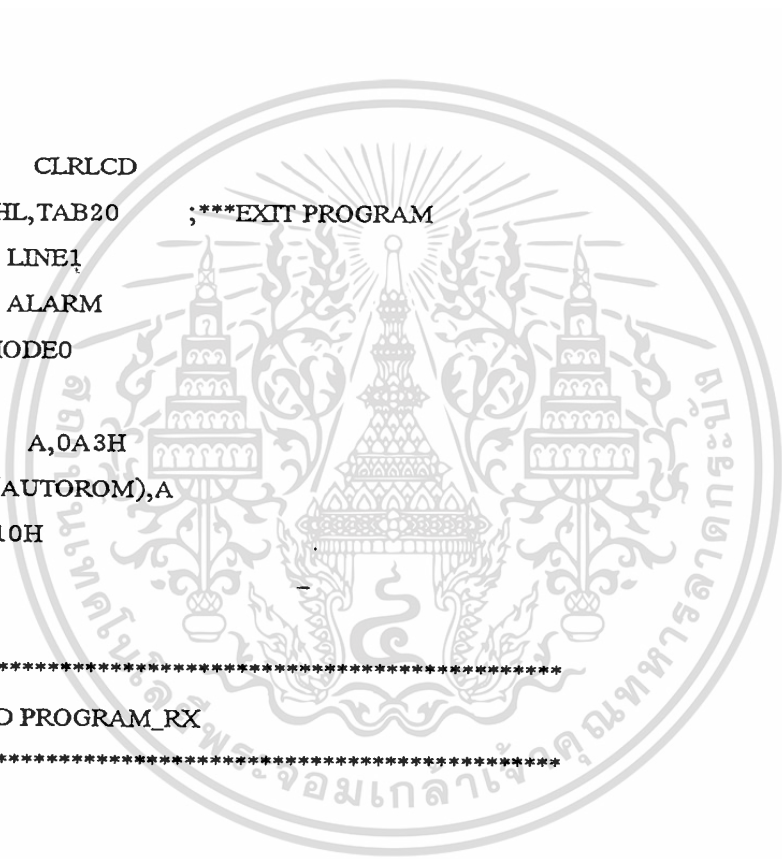
```

```

AUTO:    LD    A,0A3H
LD       (AUTOROM),A
RST     10H
END

```

END PROGRAM_RX



XR-2211

The tone frequency, f_0 , these parameters are calculated as follows:

- Choose R_0 to be in the range of 15 K Ω to 100 K Ω . This choice is arbitrary.
- Calculate C_0 to set center frequency, f_0 equal to f_c (see Figure 6): $C_0 = 1/R_0 f_0^2$
- Calculate R_1 to set bandwidth $\pm \Delta f$ (see design equation No. 5):
$$R_1 = R_0(f_0/\Delta f)$$

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:

$$C_1 = C_0 / 18 \zeta^2$$

Normally $\zeta = 1/2$ is optimum for most tone detector applications, giving $C_1 = 0.25 C_0$.

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

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

$$C_D(\mu F) \geq (18/\text{capture range in Hz})$$

Increasing C_D slows down the logic output response time.

Design Examples:

Tone detector with a detection band of 1 kHz \pm 20 Hz:

- Choose $R_0 = 20$ K Ω (18 K Ω in series with 5 K Ω potentiometer).
- Choose C_0 for $f_0 = 1$ kHz (from Figure 6): $C_0 = 0.05 \mu F$.
- Calculate R_1 : $R_1 = (R_0)(1000/20) = 1$ M Ω .
- Calculate C_1 : for $\zeta = 1/2$, $C_1 = 0.25 C_0 = 0.013 \mu F$.
- Calculate C_D : $C_D = 18/33 = 0.42 \mu F$.
- Fine-tune center frequency with 5 K Ω potentiometer, R_X .

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 12. The demodulated output is taken from the lock 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 12.

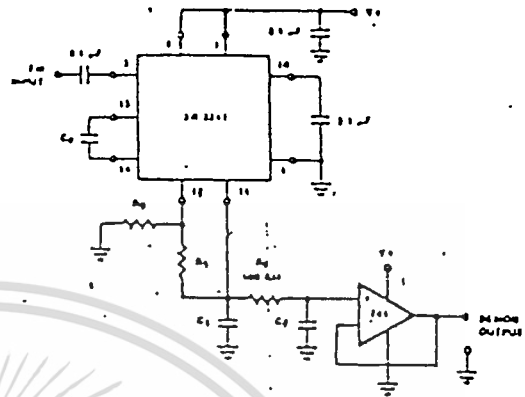


Figure 12. Linear FM Detector Using XR-2211 and an External Op Amp. (See Section on Design Equation for Component Values.)

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

$$V_{out} = R_1 V_{PP}/100 R_0 \text{ Volts}/\% \text{ deviation.}$$

where V_{PP} is the internal reference voltage ($V_{PP} = V + 12 = 650$ mV). For the choice of external components R_1 , R_0 , C_0 , C_1 and C_F see section on design equations.

PRINCIPLES OF OPERATION

Signal Input (Pin 2): Signal is ac coupled to this terminal. The internal impedance at Pin 2 is 20 K Ω . Recommended input signal level is in the range of 10 mV rms to 2V 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 2) to eliminate the chatter at lock detect outputs. If the tone detect section is not used, Pin 3 can be left open circuited.

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

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

XR-2211

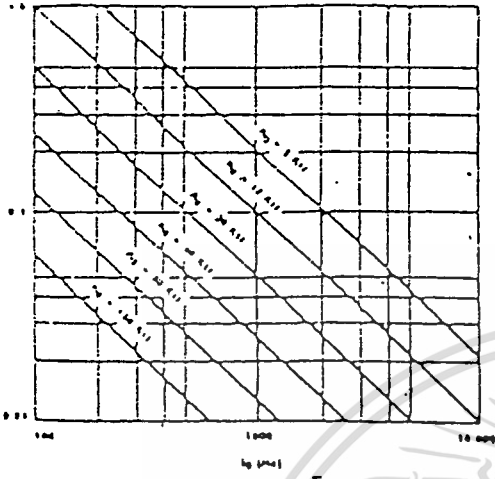


Figure 5. VCO Frequency vs Timing Resistor

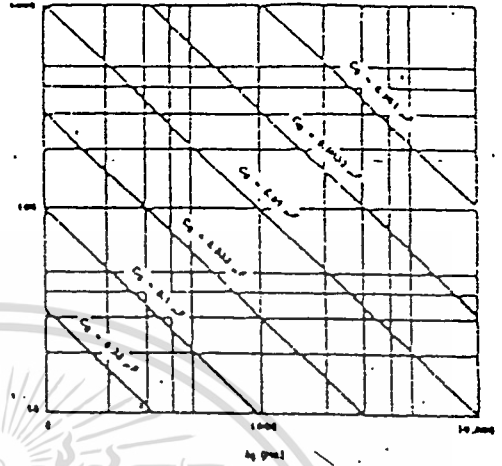


Figure 6. VCO Frequency vs Timing Capacitor

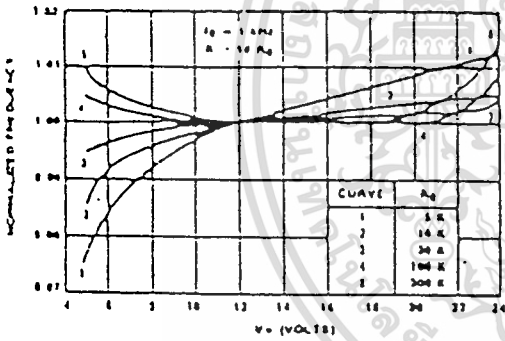


Figure 7. Typical I_Q vs Power Supply Characteristics

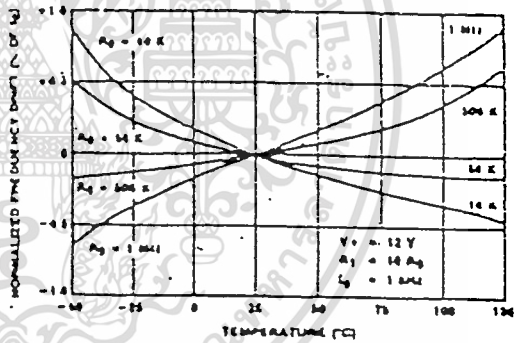


Figure 8. Typical Center Frequency Drift vs Temperature

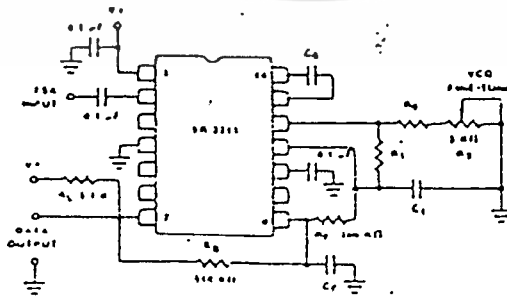


Figure 9. Circuit Connection for FSK Decoding

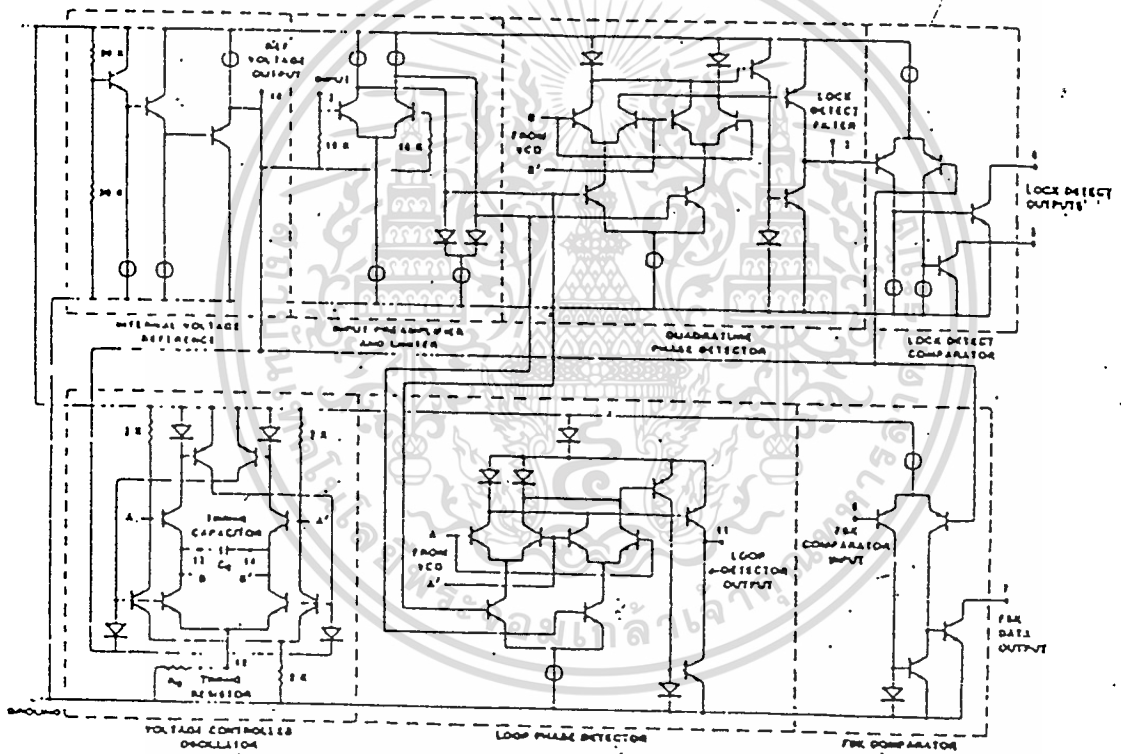
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XR-2211

FSK Data Output (Pin 7) This output is an open collector logic stage which requires a pull up resistor, R_L , to V^+ for proper operation. It can sink 5 mA 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 of Figure 2. The threshold voltage of the comparator is set by the internal reference voltage, V_R , available at Pin 10.

EQUIVALENT SCHEMATIC DIAGRAM



XR-2211

ELECTRICAL CHARACTERISTICS

Test Conditions: Test Circuit of Figure 1, $V^+ = V^- = 6V$, $T_A = +25^\circ C$, $C = 5000 \text{ pF}$, $R_1 = R_2 = R_3 = R_4 = 20 \text{ k}\Omega$, $R_L = 4.7 \text{ k}\Omega$. Binary Inputs grounded, S_1 and S_2 closed, unless otherwise specified.

PARAMETER	XR-2211/2211M			XR-2211C			UNITS	CONDITIONS
	MIN	TYP	MAX	MIN	TYP	MAX		
GENERAL								
Supply Voltage	4.5		20	4.5		20	V	$R_0 \geq 10 \text{ k}\Omega$. See Fig. 4.
Supply Current		4	7		5	9	mA	
OSCILLATOR SECTION								
Frequency Accuracy		± 1	± 3		± 1		%	Deviation from $f_0 = 1/R_0 C_0$ $R_1 = 1/2$ See Fig. 8.
Frequency Stability								
Temperature		± 20	± 50		± 20		ppm/°C	$V^+ = 12 \pm 1V$. See Fig. 7. $V^+ 5 \pm 0.5V$. See Fig. 7.
Power Supply		0.05	0.5		0.05		%/V	
Upper Frequency Limit	100	300			300		kHz	$R_0 = 8.2 \text{ k}\Omega$, $C_0 = 400 \text{ pF}$.
Lowest Practical								
Operating Frequency			0.01		0.01		Hz	$R_0 = 2 \text{ M}\Omega$, $C_0 = 50 \text{ }\mu\text{F}$ See Fig. 3.
Timing Resistor, R_0							k Ω	
Operating Range	5		2000	5		2000	k Ω	See Figs. 7 and 8.
Recommended Range	15		100	15		100	k Ω	
LOOP PHASE DETECTOR SECTION								
Peak Output Current	± 150	± 200	± 300	± 100	± 200	± 300	μA	Measured at Pin 11.
Output Offset Current		± 1			± 2		μA	
Output Impedance		1			1		M Ω	Referenced to Pin 10.
Maximum Swing	± 4	± 5		± 4	± 5		V	
QUADRATURE PHASE DETECTOR								
Measured at Pin 3.								
Peak Output Current	100	150		150			μA	
Output Impedance		1		1			M Ω	
Maximum Swing		11		11			V pp	
INPUT PREAMP SECTION								
Measured at Pin 2.								
Input Impedance		20		20			k Ω	
Input Signal								
Voltage Required to Cause Limiting		2	10		2		mV rms	
VOLTAGE COMPARATOR SECTIONS								
Input Impedance		2		2			M Ω	Measured at Pins 3 and 8.
Input Bias Current		100		100			nA	
Voltage Gain	55	70		55	70		dB	$R_L = 5.1 \text{ k}\Omega$ $I_C = 3 \text{ mA}$ $V_O = 12V$
Output Voltage Low		300		300			mV	
Output Leakage Current		0.01		0.01			μA	
INTERNAL REFERENCE								
Voltage Level	4.9	5.3	5.7	4.75	5.3	5.85	V	Measured at Pin 10.
Output Impedance		100		100			Ω	