เครื่องสั่งอาหารระยะไกล

(Ordering Food Remote Control)

โดย

นายเฉลิมภัณฑ์ ฟองสมุทร

นายอนุโรจน์ แซ่โค้ว

นายอมรพงษ์ ชุมสาย ณ อยุธยา

วัน เดือน ปี 1 ต.ค. 2540

เลขทะเบียน 037134

เลขเรียกหนังสือ T 58.2 ฉ7 ร434 ค.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาตรีวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิคส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037134

เครื่องสั่งอาหารระยะไกล

(Ordering Food Remote Control)

โดย

นายเฉลิมภัณฑ์	ฟองสมุทร	36013102
นายอนุโรจน์	แซ่โค้ว	36013137
นายอมรพงษ์	ชุมสาย ณ อยุธยา	36013138

อาจารย์ที่ปรึกษา

รศ.ดร.รัตติกร วรากูลศิริพันธุ์

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิคส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์หรือรายงานการศึกษา 2538

ภาควิชา อิเล็กทรอนิคส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องสั่งอาหารระยะไกล

ผู้จัดทำ

1. นายเฉลิมภัณฑ์	ฟองสมุทร	36013102
2. นายอนุโรจน์	แซ่โค้ว	36013137
3. นายอมรพงษ์	ชุมสาย ณ อยุธยา	36013138

.....อาจารย์ที่ปรึกษา

(รศ.ดร.รัตติกร วรากูลศิริพันธุ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องสั่งอาหารระยะไกล

นายเฉลิมภัณฑ์ ฟองสมุทร
นายอนุโรจน์ แซ่โค้ว
นายอมรพงษ์ ชุมสาย ณ อยุธยา
รศ.ดร.รัตติกร วรากูลศิริพันธุ์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2538

บทคัดย่อ

โครงงานนี้เป็นการเป็นการออกแบบการส่งข้อมูลผ่านคลื่นวิทยุซึ่งเป็นการจำลองระบบสั่งอาหารผ่านเครื่องส่งระยะไกล โดยโครงงานนี้ได้พัฒนาทั้ง Hardware และ Software ของเครื่องส่งวิทยุโดยลดความผิดพลาดของข้อมูลลงอย่างมากเพื่อให้สามารถนำมาใช้งานได้และยังปรับปรุงการทำงานของภาครับ Software โดยได้เปลี่ยนรูปแบบการประมวลผลข้อมูลให้มีความยืดหยุ่นและง่ายต่อการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Food Remote Control

Chalermpan Fongsamut
Anuroj Saekhow
Amonpong Chumsai Na Ayudtaya
Ruttikorn Varagunsiripon Advisor
1995

ABSTACT

This project presents a new partial design and constuction of remote food ordering for using in rastaurant. Based on the project of data transmitting by radio we proposed a new designed headware providing lower error menu model. We also develope a received control software on PC providing more capacity, flexibility and easy to use.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บล็อกไดอะแกรมของระบบ	2
บทที่ 2 ทฤษฎีและหลักการเบื้องต้น	3
2.1 ประเภทของการส่งข้อมูล	3
2.2 ทิศทางการส่งผ่านข้อมูล	5
2.3 อัตราบิทและอัตราบ็อด	6
2.4 โมเด็ม	6
2.5 เทคนิคในการมอดดูเลท	6
2.6 การอินเตอร์เฟส EIA RS-232C	8
2.7 การเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณ FSK	12
2.8 การเปลี่ยนสัญญาณอะนาลอก FSK เป็นสัญญาณดิจิตอล	16
บทที่ 3 ข้อมูลของบอร์ด CP-Z84C11	20
3.1 พอร์ทสื่อสาร	21
3.2 การควบคุม LCD Graphic	28
บทที่ 4 การแสดงตัวอักษรภาษาไทย	32
4.1 การแสดงภาษาไทย 4 ระดับ	32
4.2 บิตแม็พกับเว็กเตอร์	33
บทที่ 5 หลักการออกแบบโปรแกรมบนเครื่องไมโครคอมพิวเตอร์	35
บทที่ 6 หลักการออกแบบโปรแกรมบนบอร์ด CP-Z84C11	41
6.1 การแบ่งหน้าจอและการออกแบบตัวอักษร	42
6.2 การใช้งานและการออกแบบโปรแกรมจัดระบบ	49
6.3 รูปเฟรมข้อมูลของระบบ	52
บทที่ 7 ผลการทดลอง	53
บทที่ 8 สรุปและวิจารณ์	56
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	เปรียบเทียบการส่งข้อมูลแบบขนานและแบบอนุกรม	4
3.1	แสดงขาที่ใช้เชื่อมต่อ	28
7.1	ตารางแสดงผลการทดลอง	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
	1.1 บล็อกไดอะแกรมของชุดเครื่องส่ง	2
	1.2 บล็อกไดอะแกรมของชุดเครื่องรับ	2
	2.1 การส่งข้อมูลในลักษณะต่างๆ	5
	2.2 การมอดดูเลทสัญญาณแบบขนาด	7
	2.3 การมอดดูเลทสัญญาณแบบความถี่	7
	2.4 การใช้ PSK ผสมสัญญาณแบบหนึ่งบิต	8
	2.5 ขาสัญญาณบนจุดต่อ (Connector) 25 ขา	9
	2.6 แสดงการอินเตอร์เฟสกับเครื่องคอมพิวเตอร์	12
	2.7 แสดงบล็อกโครงสร้างแต่ละส่วนของ XR-2206	13
	2.8 แสดงการนำไอซี XR-2206 ไปใช้เป็นวงจรฟรีเควนซี่ชิฟท์คีย์อิงโมดูเลเตอร์	13
	2.9 แสดงโครงสร้างภายในของไอซี XR-2211	16
	2.10 แสดถึงการนำไอซี XR-2211 ไปใช้เพื่อทำการดีเทครหัสสัญญาณ	17
	3.1 ลักษณะทั่วไปของบอร์ด	20
	3.2 แสดงการเซตตำแหน่งจัมเปอร์	21
	3.3 แสดงพอร์ทใช้งาน	21
	3.4 ตำแหน่ง Z80 บัส	23
	3.5 การเซตพอร์ท RS-232	23
	3.6 การต่อกับคีย์บอร์ด	24
	3.7 การเลือก LCD	24
	3.8 การต่อกับ LCD	25
	3.9 แสดงแผนผังของหน่วยความจำ	25
	3.10 แสดงรายละเอียดของขา Z84C11	26
	3.11 แสดงวงจรโดยละเอียดของบอร์ด CP-Z84C11	27
	3.12 บล็อกไดอะแกรมและแหล่งจ่ายไฟ	29
	4.1 แสดงภาษาไทยเมื่อพิมพ์ช่องละ 1 อักขระ	32
	4.2 แสดงภาษาไทยแบ่งช่องพิมพ์ออกเป็น 2 ส่วนเมื่อลดความสูงลงแล้ว	33

ต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ หน้า

4.3 การสร้างฟอนต์แบบบิทแม็พขนาด 8 คูณ 16 34

4.4 การสร้างฟอนต์แบบเวคเตอร์ขนาด 8 คูณ 16 34

5.1 โฟร์ชาร์ทการทำงาน 35

5.2 โฟร์ชาร์ทแสดงเมนูหลัก 36

5.3 โฟร์ชาร์ทแสดงเมนูพ่อครัว 37

5.4 โฟร์ชาร์ทแสดงเมนูแคชเชียร์ 38

5.5 โฟร์ชาร์ทแสดงเมนูแก้ไข 39

5.6 โฟร์ชาร์ทแสดงการส่งข้อมูลออก 40

5.7 โฟร์ชาร์ทการตั้งเวลา 40

6.1 แสดงตำแหน่งจอ LCD 41

6.2 การแสดงตัวอักษรภาษาไทย 41

6.3 การทำงานของแต่ละกระบวนการของจอ LCD 42

6.4 แสดงประเภทอักขระต่างๆ 43

6.5 โฟร์ชาร์ทการแยกประเภทอักขระต่างๆ 43

6.6 รูปแบบตัวอักษรภาษาไทย 44

6.7 รูปแบบตัวอักษรภาษาอังกฤษและตัวแปรต่างๆ 45

6.8 การใช้งานไบท์กำหนดแอททริบิว 46

6.9 โฟร์ชาร์ทกระบวนการเขียนสตริงในแมมโมรี่ 47

6.10 โฟร์ชาร์ทนำค่าหน่วยความจำออกแสดงผล 48

6.11 โฟร์ชาร์ทการทำงานของโปรแกรมหลัก 49

6.12 ตัวเลือกเมนูหลัก 50

6.13 เมนูย่อยที่ 1 50

6.14 เมนูย่อยที่ 2 51

6.15 เมนูย่อยที่ 3 51

6.16 เมนูย่อยที่ 4 51

6.17 ส่วนประกอบของเฟรมข้อมูล 52

7.1 แสดงรูปข้อมูลที่ถูกหน่วงเวลาจากการป้อนสัญญาณที่อินพุท 54

7.2 แสดงการทดสอบขั้นที่ 3 54

7.3 แสดงการทดสอบขั้นที่ 4 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

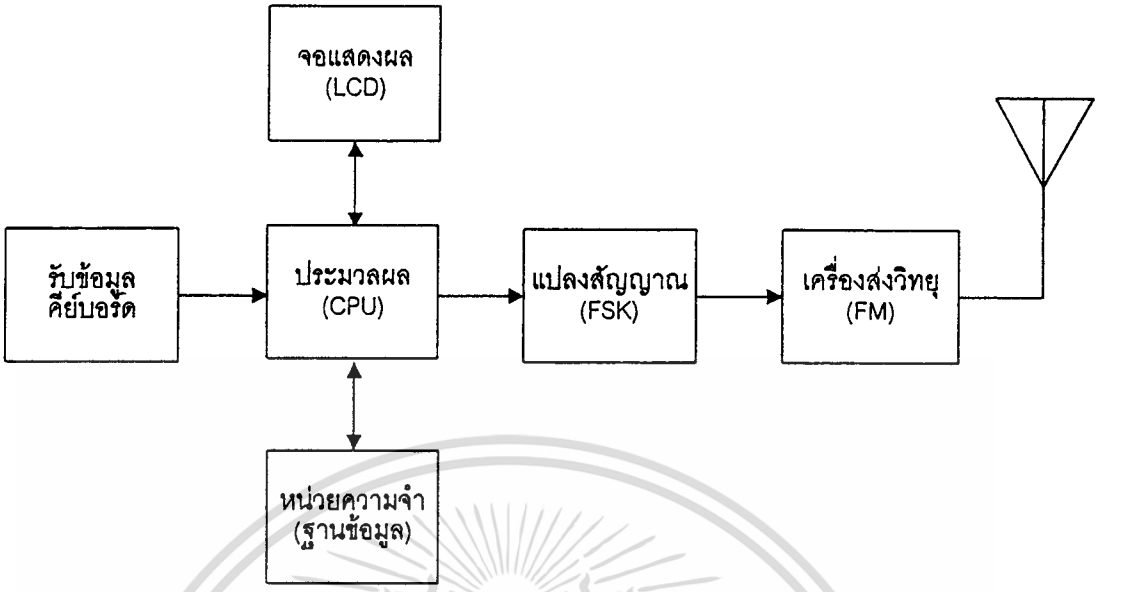
บทที่ 1
บทนำ

ปัจจุบันความก้าวหน้าทางเทคโนโลยีสื่อสารและประมวลผลคอมพิวเตอร์ได้ก้าวหน้าไปอย่างมาก และก้าวหน้าไปพร้อม ๆ กันกับแนวความคิดในการเชื่อมโยงคอมพิวเตอร์แต่ละหน่วยให้เป็นระบบโครงข่ายขนาดใหญ่ที่มีประสิทธิภาพสูงในการทำงานเป็นแนวความคิดใหม่ที่ท้าทายผลประโยชน์ที่ได้รับจากการสร้างระบบส่งข้อมูลสื่อสารดิจิตอลระยะไกลที่กว้างขวางและทันสมัย โดยระบบส่งข้อมูลแบบนี้ช่วยในการประหยัดค่าใช้จ่ายในการเดินสายให้ถูกลง รวมทั้งก่อให้เกิดการประยุกต์ใช้กับงานต่างๆมากมาย

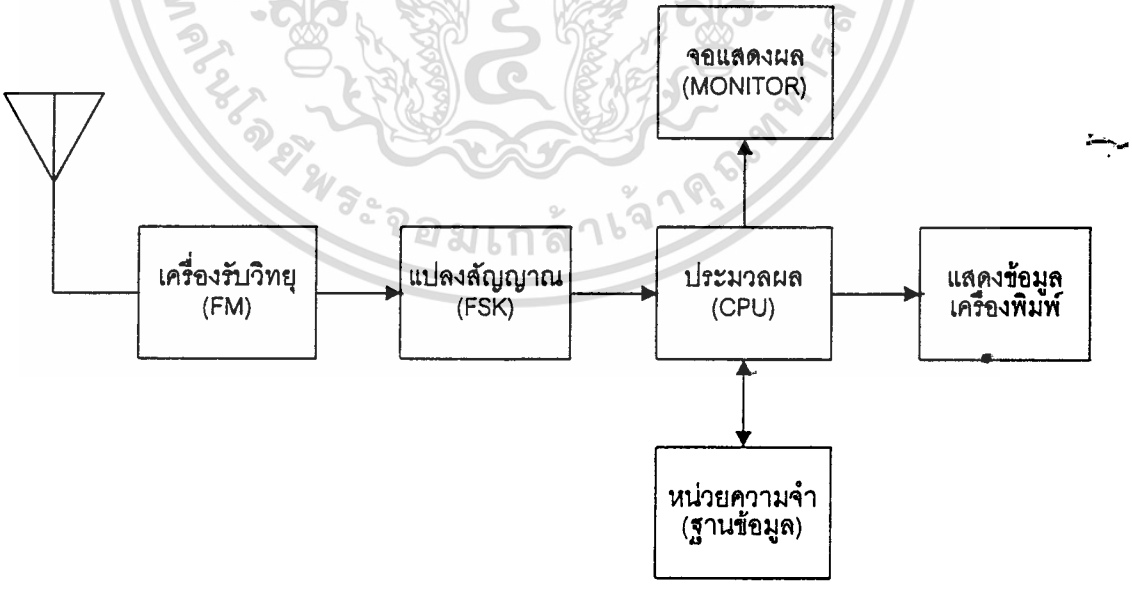
ส่วนสำคัญของระบบส่งข้อมูลโดยใช้คลื่นวิทยุที่จะต้องก้าวไปให้ทันกับความสามารถในการรับส่งและประมวลผลข้อมูลของคอมพิวเตอร์ การสื่อสารข้อมูลในปัจจุบันมีทั้งแบบที่ใช้ความเร็วสูงมาก ๆ และความเร็วต่ำซึ่งก็ขึ้นอยู่กับความสามารถของอุปกรณ์และระบบสื่อสารที่ใช้ โครงงานนี้เป็นการประยุกต์ส่งข้อมูลมาใช้งานเป็นรีโมทสั่งอาหารที่ใช้ในร้านอาหารใหญ่เพื่อประหยัดเวลาในการเดินสั่งอาหารบ่อย ๆ และยังลดการจ้างเด็กเสิร์ฟลงทำให้ประหยัดค่าใช้จ่ายได้ และยังสะดวกรวดเร็วในการสั่งอาหารอีกด้วย

เหตุผลอันหนึ่งที่ไม่เดินสายคอมพิวเตอร์ไปห้องครัว แต่ใช้คลื่นวิทยุแทนก็คือ แบนด์วิดส์ของอากาศที่สามารถตอบสนองความถี่ของสัญญาณได้อย่างไม่จำกัด ทำให้สามารถพัฒนาเพื่อเพิ่มความเร็วในการส่งสัญญาณได้ รวมทั้งความสะดวกในการเคลื่อนย้ายคอมพิวเตอร์ แต่เนื่องจากการแพร่กระจายคลื่นวิทยุในอากาศเป็นสิ่งที่มีการควบคุมอย่างเข้มงวดดังนั้นการใช้งานจึงถูกจำกัด และทำให้ขนาดของโครงข่ายไม่สามารถทำให้กว้างขวาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 บล็อกไดอะแกรมของชุดเครื่องส่ง



รูปที่ 1.2 บล็อกไดอะแกรมของชุดเครื่องรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการเบื้องต้น

การสื่อสารข้อมูลดิจิตอลเป็น ขบวนการในการแลกเปลี่ยนข้อมูลหรือข่าวสาร ซึ่งในการสื่อสารกันนั้นต้องประกอบไปด้วย ผู้ส่ง (Sender) ผู้รับ (Receiver) และตัวกลางในการส่งข้อมูล (Medium) โดยที่ข้อมูลที่จะส่งนั้นเป็นสัญญาณดิจิตอล คืออยู่ในรูปเลขฐานสองข้อมูลข่าวสารที่จะส่งจะเป็นรหัสตัวอักษร ตัวเลข หรือเครื่องหมายที่อยู่ในรูปเลขฐานสอง เช่น รหัสแอสกี (ASCII) หรือ รหัส (EBCDIC)

2.1 ประเภทของการส่งข้อมูล

โดยทั่ว ๆ ไปหลักใหญ่ของการส่งข้อมูลในคอมพิวเตอร์ หรือ ระหว่างคอมพิวเตอร์ด้วยกันจะมีลักษณะของการส่งข้อมูลอยู่ 2 แบบ คือ การส่งข้อมูลแบบขนาน และการส่งข้อมูลแบบอนุกรม

1. การส่งข้อมูลแบบขนาน

ลักษณะของการส่งข้อมูลแบบขนาน (Parallel Interface) ทำได้โดยการส่งข้อมูลออกมาที่ละไบท์ ซึ่งถ้าใน 1 ไบท์มี 8 บิท ทั้ง 8 บิท จะถูกส่งจากอุปกรณ์ส่งไปอุปกรณ์รับพร้อมกัน และช่องสัญญาณในการส่งข้อมูลจะต้องมีอย่างน้อย 8 ช่องสัญญาณสำหรับสัญญาณแต่ละบิท

2. การส่งข้อมูลแบบอนุกรมมี 2 แบบคือ

ก. การส่งข้อมูลแบบอะซิงโครนัส (Asynchronous Transmission)

ในการส่งแบบอะซิงโครนัส การส่งข้อมูลแต่ละตัวอักษรไม่มีกำหนดเวลาที่แน่นอน คือแต่ละตัวอักษรอยู่ห่างกันเท่าไรก็ได้ หรือจะส่งติดกันไปตลอดก็ได้เช่นกัน ในการส่งข้อมูลแต่ละตัวหรือแต่ละไบท์นั้นจะมีสัญญาณสำหรับตรวจสอบบิทแรกภายในตัวของมันเอง โดยแต่ละไบท์จะถูกเพิ่มด้วยบิทเริ่มต้น (Start Bit) นำหน้าไบท์นั้น และบิทสิ้นสุด (Stop Bit) ตามหลังไบท์นั้น ซึ่งอาจจะมีการเพิ่มบิทพาริตี้ (Parity Bit) ก่อนบิทสิ้นสุดก็ได้

ข. การส่งข้อมูลแบบซิงโครนัส (Synchronous Transmission)

การส่งข้อมูลแบบซิงโครนัสหมายถึง การส่งข้อมูลแบบอนุกรมที่มีการกำหนดจำนวนอักขระที่จะส่งในแต่ละครั้งเป็นจำนวนที่แน่นอนเรียกว่า เฟรมข้อมูล(data frame) การส่งข้อมูลแบบนี้จะต้องมีการส่งสัญญาณนาฬิกาไปพร้อม ๆ กับสัญญาณข้อมูล ในการส่งข้อมูลระยะสั้น ๆ สัญญาณนาฬิกาซึ่งใช้เป็นสัญญาณซิงค์อาจจะส่งแยกไปในสายส่งสัญญาณอีกเส้นหนึ่งไม่ส่งรวมไปในสายส่งข้อมูลก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเปรียบเทียบระหว่างการส่งข้อมูลแบบขนานและแบบอนุกรม

	แบบขนาน	แบบอนุกรม
1. ระยะทาง	ปกติจะน้อยกว่า 100 ฟุต	ส่งได้ตั้งแต่ระยะทางสั้น ๆ จนถึงระยะทางเป็นไมล์
2. ความเร็ว	อัตราความเร็วสูงมากในระยะที่ไม่ไกล นักกำหนดได้เป็นจำนวนบิทต่อวินาที	อัตราความเร็วของข้อมูลที่ใช้กันอยู่ทั่วไปจะอยู่ในช่วง 0 ถึง 2 ล้านบิทต่อวินาที
3. ระดับของสัญญาณ	ในการอินเตอร์เฟสจะใช้ระดับของสัญญาณที่ใช้กับอุปกรณ์ TTL คือ สัญญาณลอจิก 1 และ 0 จะแทนด้วยระดับแรงดัน +5V และ 0V ตามลำดับ	ใช้มาตรฐานของ EIA-RS 232C คือ มีระดับสัญญาไฟฟ้าขนาด 12V หรืออาจจะใช้มาตรฐาน 20mA current loop หรืออาจจะใช้ระดับสัญญาณของ TTL ก็ได้ (ใช้กันน้อยมาก)
4. ความผิดพลาดของสัญญาณ	ถ้าส่งในระยะทางไกล ๆ ความผิดพลาดของข้อมูลจะเกิดขึ้นได้ง่าย	การผิดพลาดของสัญญาณจะมีน้อยลง
5. ค่าใช้จ่าย	ถ้าส่งในระยะทางไกล ๆ จะสิ้นเปลืองค่าใช้จ่ายมากเพราะต้องใช้สายสัญญาณหลายเส้น	สิ้นเปลืองน้อยกว่าหลายเท่า ถึงแม้ว่าจะใช้อุปกรณ์เปลี่ยนสัญญาณของข้อมูลจากแบบขนานไปเป็นแบบอนุกรมแล้วส่งผ่านสายส่งใช้อุปกรณ์ในการแปลงสัญญาณกลับมาเป็นแบบขนานอีกก็ยังลงทุนน้อยกว่า

ตารางที่ 2.1 เปรียบเทียบการส่งข้อมูลแบบขนานและอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทิศทางของการส่งผ่านข้อมูล

ในการรับส่งข้อมูลระหว่างคอมพิวเตอร์นั้น อาจจะแบ่งตามลักษณะของการรับส่งได้เป็น 3 วิธีใหญ่ ๆ คือ

1) การรับหรือส่งทางเดียว (Simplex)

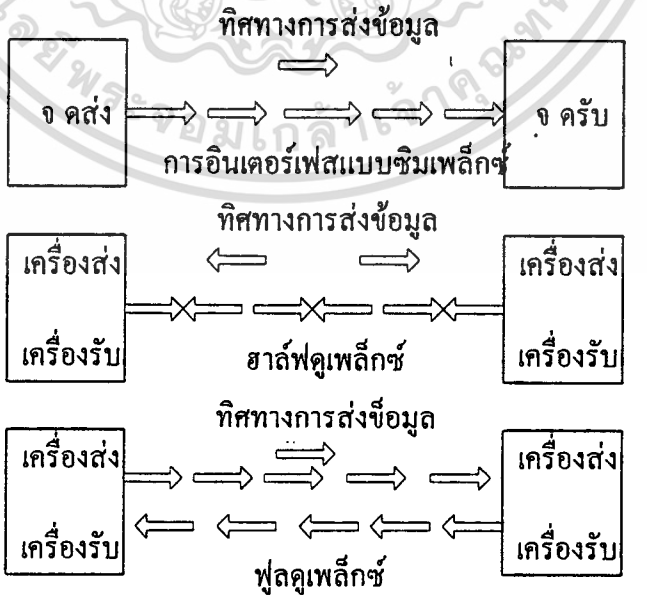
เป็นการส่งข้อมูลแบบไปในทางเดียว เช่นการส่งข้อมูลจากคอมพิวเตอร์ A ไปยังเทอร์มินอล B ในกรณีนี้คอมพิวเตอร์จะต้องเป็นเครื่องส่ง และ เทอร์มินอล B จะเป็นเครื่องรับเท่านั้น

2) การรับส่งแบบผลัดกันส่ง (Half Duplex)

เป็นการรับส่งข้อมูลใน 2 ทิศทางสามารถทำหน้าที่เป็นเครื่องรับ และเครื่องส่งได้แต่ว่าต้องผลัดกันรับผลัดกันส่ง เมื่อฝ่ายใดฝ่ายหนึ่งส่งอีกฝ่ายหนึ่งจะทำหน้าที่รับ จนกระทั่งฝ่ายแรกส่งจบ ฝ่ายหลังจึงจะกลับมาเป็นผู้ส่งได้

3) การรับส่งสวนทางได้พร้อมกัน (Full Duplex)

เป็นการรับส่งข้อมูลใน 2 ทิศทางพร้อมกัน เช่น คอมพิวเตอร์ A ส่งข้อมูลไปให้คอมพิวเตอร์ B ซึ่งเป็นเวลาเดียวกันกับที่คอมพิวเตอร์ B ก็ส่งข้อมูลไปให้คอมพิวเตอร์ A คอมพิวเตอร์ A และ B จะมีการทำงานที่เป็นอิสระต่อกัน ส่วนอีกระบบหนึ่งจะใช้สายสัญญาณส่งและรับร่วมกัน และในระบบจะมีสายสัญญาณเพียง 2 เส้น ในลักษณะนี้เรียกว่าเป็นระบบฟูลดูเพล็กซ์ 2 เส้น (Full duplex 2 wires) แต่ในระบบฟลูดูเพล็กซ์ 2 เส้นนี้ จะต้องอาศัยเทคนิคการแบ่งความถี่เข้ามาช่วย คือ จะส่งในความถี่ช่วงหนึ่ง และจะรับความถี่อีกช่วงหนึ่ง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.1 การส่งข้อมูลในลักษณะต่าง ๆ

2.3 อัตราบิทและอัตราบ็อด (Bit Rate Versus Baud Rate)

อัตราบิทเป็นการส่งข้อมูลแบบอนุกรมจากเครื่องคอมพิวเตอร์เครื่องหนึ่งไปยังอุปกรณ์ต่อพ่วง หรือไปยังคอมพิวเตอร์อีกเครื่องหนึ่ง วัดเป็นจำนวนบิทต่อวินาที อาจมีค่าไม่เท่ากับอัตราการเปลี่ยนแปลงในสายส่ง (Baud Rate) ก็ได้

อัตราบ็อด เป็นอัตราการเปลี่ยนแปลงของสัญญาณอนาล็อกในสายส่ง ซึ่งอาจจะเป็นอัตราการเปลี่ยนความถี่ อัตราการเปลี่ยนขนาดของสัญญาณหรืออัตราการเปลี่ยนช่วงต่อเนื่องของมุม (Phase) ในหนึ่งวินาทีก็ได้ และ อัตราบ็อดไม่จำเป็นต้องมีค่าเท่ากับอัตราการส่งข้อมูล ซึ่งวัดเป็นบิทต่อวินาที โดยอัตราบ็อดอาจจะมีค่าน้อยกว่าหรือมากกว่าอัตราการส่งข้อมูลก็ได้ ขึ้นอยู่กับเทคนิคการผสมสัญญาณที่ใช้

2.4 โมเด็ม (Modem)

เราสามารถแบ่งชนิดของโมเด็มตามอัตราการส่งข้อมูลได้เป็น

1. อัตราการส่งข้อมูลต่ำ (Low-speed)
   มีอัตราการส่งข้อมูลไม่เกิน 600 บิทต่อวินาที
2. อัตราการส่งข้อมูลปานกลาง (Medium-speed)
   มีอัตราการส่งข้อมูลไม่เกิน 1200 บิทต่อวินาที
3. อัตราการส่งข้อมูลสูง (High-speed)
   มีอัตราการส่งข้อมูล 9600 บิทต่อวินาที

นอกจากนี้เรายังแบ่งชนิดของโมเด็มได้ตามเทคนิคการมอดดูเลทมีดังนี้

1. Frequency Shift Keying (FSK)
2. Phase Shift Keying (PSK)
3. Phase Amplitude Modulation (PAM)

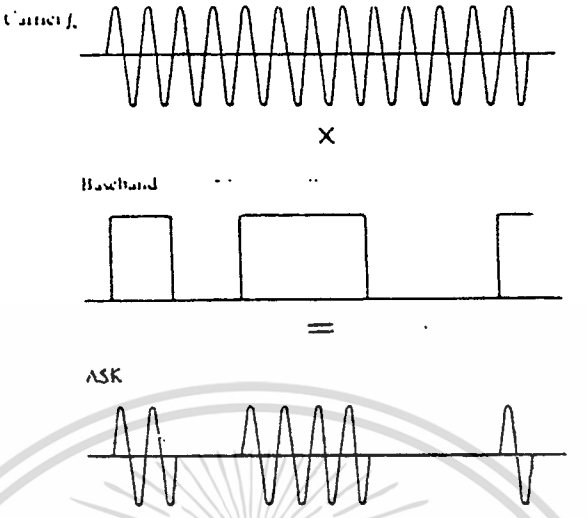
2.5 เทคนิคในการมอดดูเลทสัญญาณ (Digital Modulation Techniques)

การส่งผ่านข้อมูลจะต้องมีการมอดดูเลทสัญญาณข้อมูลแบ่งเป็นวิธีการต่าง ๆ ดังนี้

1. Amplitude Shift Keying (ASK)

เป็นการเปลี่ยนแปลงสัญญาณดิจิตอลให้อยู่ในรูปความถี่เดียว ที่มีความแตกต่างกันทางด้านระดับสัญญาณ โดยให้ส่วนที่มีแอมปลิจุดสูงกว่าเป็นระดับลอจิก “1” และระดับแอมปลิจุดต่ำเป็นระดับลอจิก “0”

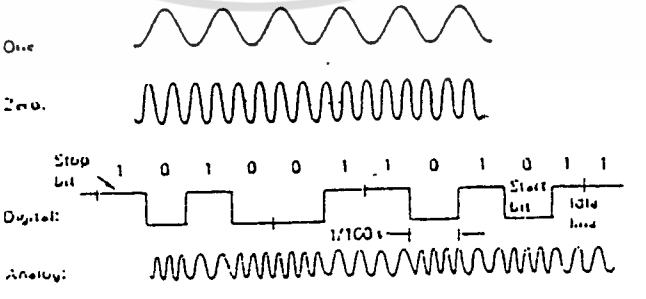
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การมอดดูเลทสัญญาณแบบขนาด

2) Frequency Shift Keying (FSK)

การมอดดูเลทสัญญาณด้วยวิธีนี้ เราทราบกันดีแล้วว่าใช้ในโมเด็มความเร็วต่ำโดยแทน “0” และ “1” ด้วยความถี่ต่างกัน ผู้ส่งจะใช้ความถี่สองความถี่ แทน “0” และ “1” ของด้านส่ง ส่วนผู้รับก็ใช้ความถี่อีกสองความถี่แทน “0” และ “1” ของด้านรับเช่นกัน



รูปที่ 2.3 การมอดดูเลทสัญญาณแบบความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเด็มแบบ FSK แบ่งออกได้ 2 แบบ คือ

ก) ชนิดที่ใช้ในการส่งข้อมูลในระบบฟูลดูเพล็กซ์

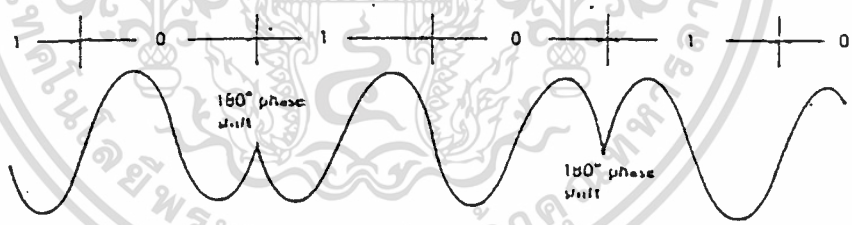
โมเด็มชนิดนี้จะแบ่งแบนด์วิดธ์ของช่องสัญญาณเสียงออกเป็น 2 แบนด์เท่า ๆ กัน โดยแบนด์หนึ่งใช้ในการส่งข้อมูล ส่วนอีกแบนด์หนึ่งจะใช้ในการรับข้อมูล ทำให้สามารถส่งข้อมูลเข้าไปในสายได้พร้อม ๆ กัน ผลตอบสนองความถี่ (Frequency Response)

ข) ชนิดที่ใช้ในการส่งข้อมูลในระบบ าล์ฟดูเพล็กซ์

โมเด็มในระบบนี้จะมีแบนด์ 2 แบนด์เหมือนกัน ส่งแบนด์วิดธ์ที่แคบจะใช้ในการส่งสัญญาณแนะนำควบค ม (Supervisory Signal) ของตัวรับไปยังตัวส่งเพื่อใช้ในการตรวจสอบสภาพการส่งข้อมูลว่าถูกต้องหรือไม่เรียกแบด์วิดธ์นี้ว่า ช่องสัญญาณย้อนกลับ (Reverse Channel)

3) Phase Shift Keying (PSK)

สำหรับการรับส่งสัญญาณแบบ Phase Shift Keying (PSK) นั้นใช้ หลักการที่ว่าแทนข้อมูล "0" และ "1" ด้วยการเปลี่ยนแปลงม มของช่วงสัญญาณในสายส่ง (Phase) เช่นเราอาจกำหนดว่า "0" แทนด้วยม มต่อเนื่องกันไป และ "1" แทนด้วยม มเปลี่ยนไปจากเดิม 180 องศา ถ้าเราส่งข้อมูล 0101 รูปร่างคลื่นในสายส่งจะเป็นดังในรูป



รูปที่ 2.4 การใช้ PSK ผสมสัญญาณแบบหนึ่งบิต

2.6 อินเตอร์เฟส EIA RS-232C

โดยปกติไมโครคอมพิวเตอร์จะมีพอร์ทที่เป็นแบบอน กรมเรียกกันว่า RS-232C อยู่ในตัวเองอยู่แล้วหลายเครื่องไม่มีมากับเครื่อง เช่น IBM PC จึงจำเป็นจะต้องมีการ์ดที่เรียกว่า Asynchronous Communication Adapter มาเสียบใส่

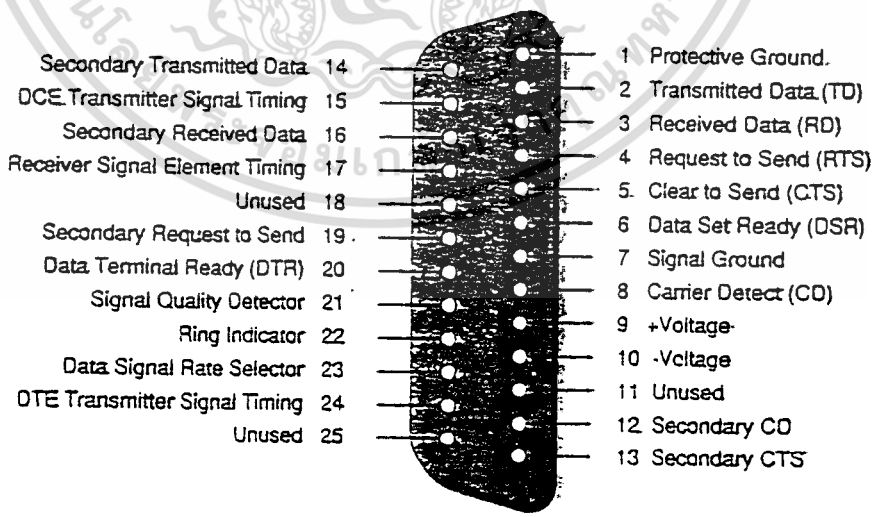
พอร์ท RS-232C นี้ทำหน้าที่รับและส่งข้อมูลในแบบอน กรมเรียกว่า Univercal Asynchronous Aadapter เหต ที่มีชื่อเรียกว่า RS-232C ก็เนื่องจากสมาคมผ ผลิตอ ปกรณ์อิเล็กทรอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิคส์ของอเมริกา หรือ EIA (Electronics Industries Association) ได้กำหนดมาตรฐานของอุปกรณ์การสื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS-232C ความจริงมาตรฐานของการส่งข้อมูลแบบอนุกรมมีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับไมโครคอมพิวเตอร์ก็คือ RS-232C

มาตรฐาน RS-232C ได้จัดพิมพ์ขึ้นเมื่อปี ค.ศ. 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิคส์แห่งสหรัฐอเมริกา RS ย่อมาจาก Recommended Standard ส่วน 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ C เป็นหมายเลขของฉบับท้ายสุดของมาตรฐานตัวนี้ จุดประสงค์ของมาตรฐานตัวนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment) สำหรับผู้ใช้ไมโครคอมพิวเตอร์ DTE ก็หมายถึงตัวไมโครคอมพิวเตอร์ และ DCE (Data Communication Equipment) ก็หมายถึงโมเด็มหรืออุปกรณ์อื่น ๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรม อาจจะเป็นได้ทั้ง DTE และ DCE ขึ้นอยู่กับผู้ผลิต ความจริงอีกประการหนึ่งของ RS-232C ก็คือ ความเร็วและระยะทางของการเชื่อมต่อ RS-232C สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้จาก 0 ถึง 10,000 บิทต่อวินาที ซึ่งเพียงพอสำหรับไมโครคอมพิวเตอร์ที่มีขนาดอัตราบ็อด 110 ถึง 9,600 บ็อด ความยาวของสายเชื่อมต่อสัญญาณตามมาตรฐานของ RS-232C จำกัดอยู่แค่ 50 ฟุตซึ่งเพียงพอสำหรับการสื่อสารระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์รอบนอก

อินเตอร์เฟส RS-232C จะใช้ต่อแบบ DB-25 (D-type 25 pin connector) เป็นมาตรฐาน โดยข้อต่อบน DTE จะเป็นตัวผู้ ส่วนข้อต่อที่ DCE จะเป็นตัวเมียโดยลักษณะของข้อต่อพร้อมทั้งชื่อขา แสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 ขาสัญญาณบนจุดต่อ (Connector) 25 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระดับสัญญาณที่ใช้ใน RS-232C จะเป็นสัญญาณโวลท์เตจสองระดับ (Bipolar Voltage Signal) ระดับสัญญาณต่าง ๆ เมื่อเทียบกับกราวด์จะต้องไม่เกิน ± 25 โวลท์ และเมื่อเกิดการลัดวงจรระหว่างขาต่าง ๆ จะต้องไม่เกิดการเสียหายขึ้นกับอุปกรณ์ใด ๆ ที่ต่ออยู่กับ RS-232C รวมทั้งตัวมันเองด้วย สัญญาณลอจิกเอาท์พุท "0" เมื่อผ่าน RS-232C จะมีระดับโวลท์เตจอยู่ในช่วง +5 ถึง +15 โวลท์ และลอจิก "1" จะมีระดับโวลเตจอยู่ในช่วง -5 ถึง -15 โวลท์ ส่วนที่ตัวรับสัญญาณอินพุทลอจิก "0" จะมีค่าในช่วง +3 ถึง +15 โวลท์ และสัญญาณอินพุทลอจิก "1" จะมีค่าในช่วง -3 ถึง -15 โวลท์ สำหรับระดับโวลเตจในช่วง ± 3 โวลท์ จะไม่มีความหมายใด ๆ

สัญญาณต่าง ๆ ที่ใช้ในบน RS-232C ที่สำคัญ แสดงได้ดังนี้

Transmit Data (TD ขาที่ 2)

เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อเข้าโดยตรงกับคอมพิวเตอร์ตัวอื่นหรือตัวเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออกสถานภาพของลอจิกที่ขานี้จะมีค่าเท่ากับ "1" หรือเท่ากับบิทสิ้นสุด

Receive Data (RD ขาที่ 3)

เป็นทางของสัญญาณเข้าไปยัง DTE หรือไมโครคอมพิวเตอร์ เมื่อไม่มีสัญญาณรับเข้ามาขานี้จะมีสถานภาพทางลอจิกเป็น "1"

Request To Send (RTS ขาที่ 4)

ใช้สำหรับส่งสัญญาณไปยังโมเด็ม หรือเครื่องพิมพ์เป็นการเรียกร้องที่จะส่งสัญญาณมาทางขา 2 สัญญาณนี้จะใช้คู่กับ CTS อุปกรณ์รับหากได้รับสัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณหรือยัง หากพร้อมที่จะรับส่งสัญญาณออกไปที่สาย CTS Clear To Send (CTS ขาที่ 5)

ดังอธิบายไว้ใน RTS เมื่อสัญญาณนี้อยู่ในสถานภาพออฟ (negative voltage หรือ ลอจิก "1") หมายความว่า อุปกรณ์รับกำลังบอกว่าพร้อมที่จะรับข้อมูลแล้ว

Data Set Ready (DSR ขาที่ 6)

เมื่อสัญญาณสายนี้อยู่ในสถานะออน (หรือลอจิก "0") เป็นการบอกไมโครคอมพิวเตอร์หรือฝ่ายส่งว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้ว และพร้อมที่จะส่งได้แล้ว โมเด็มที่มีการหมุนเวียนเลยอัตโนมัติ จะส่งสัญญาณสายนี้ไปบอกให้คอมพิวเตอร์รู้ว่า ต่อโทรศัพท์ได้สำเร็จแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal Ground (SG ขาที่ 7)

SG ทำหน้าที่เป็นระดับแรงดันหรือแรงดันอ้างอิงสำหรับทุก ๆ สายของสัญญาณจะมีแรงดันเป็น "0" เมื่อเทียบกับสัญญาณตัวอื่น

Carrier Detect (CD ขาที่ 8)

โมเด็มจะส่งสัญญาณที่อยู่ในสถานะออน (ลอจิก "0") ไปบอกไมโครคอมพิวเตอร์ เมื่อได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่ง สัญญาณนี้จะนำไปจุด LED บอกว่า ได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่งแล้ว ไฟ LED จะอยู่หน้าปิดของโมเด็มเอง

Data Terminal Ready (DTR ขาที่ 20)

คอมพิวเตอร์เปิดสัญญาณสายนี้ให้ออน (ลอจิก "0") เมื่อพร้อมที่จะติดต่อกับโมเด็ม โมเด็มส่วนมากจะไม่รายงานสถานภาพของตัวเอง (CD, DSR, CTS) ให้คอมพิวเตอร์รู้ หากคอมพิวเตอร์ไม่เปิดสัญญาณ DTR

Ring Indicator (RI ขาที่ 22)

สัญญาณที่ใช้ในโมเด็มที่เป็นระบบตอบโต้อัตโนมัติ (Auto-answer) สัญญาณนี้จะออนเมื่อมีสัญญาณกระดิ่งมา และออฟระหว่างเสียงดังของกระดิ่ง

การทำงานของวงจรอินเตอร์เฟส

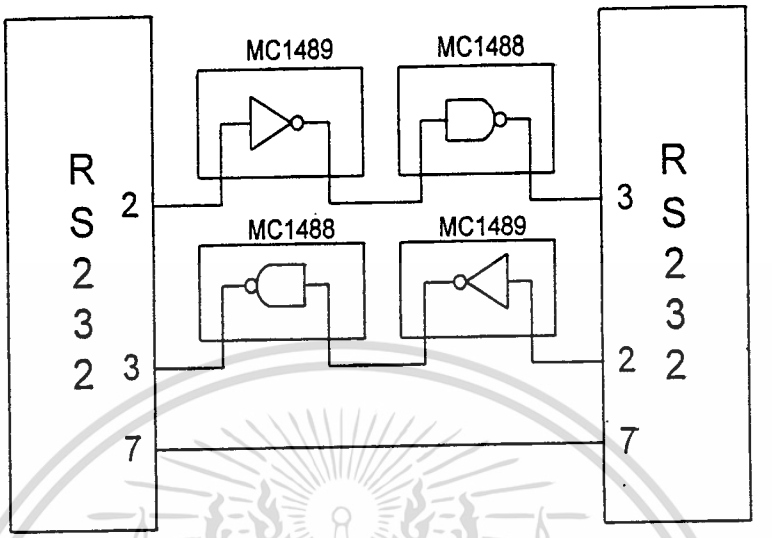
การส่งข้อมูลจากคอมพิวเตอร์มายังโมเด็มที่นิยมใช้กันนั้นเป็นการส่งข้อมูลออกมาที่พอร์ท RS-232C ซึ่งเป็นการส่งข้อมูลแบบอนุกรม โดยระดับแรงดันที่ส่งผ่านพอร์ท RS-232 นี้จะอยู่ในระดับ ±12 โวลท์ ดังนั้นตัวโมเด็มจึงจำเป็นต้องมีตัวแปลงระดับแรงดัน ±12 โวลท์ ให้อยู่ในระดับ TTL (0 - 5 โวลท์) ในที่นี้เราใช้ ไอซี MC1489 โดยระดับแรงดันที่เข้าตัวไอซีนี้

- หากเป็น 12 โวลท์ เมื่อผ่าน MC1489 จะได้รับระดับแรงดัน 0 โวลท์
- หากเป็น -12 โวลท์ เมื่อผ่าน MC1489 จะได้รับระดับแรงดัน 5 โวลท์

ในการส่งข้อมูลเข้าพอร์ท RS-232 ก็เช่นกันจำเป็นที่จะต้องเปลี่ยนแรงดัน ในระดับ TTL (0 - 5 โวลท์) ให้อยู่ในระดับแรงดัน ±12 โวลท์ ในที่นี้เราใช้ ไอซี MC1488 โดยระดับแรงดันที่เข้าตัวไอซีนี้

- หากเป็น 0 โวลท์ เมื่อผ่าน MC1489 จะได้รับแรงดัน 12 โวลท์
- หากเป็น 5 โวลท์ เมื่อผ่าน MC1489 จะได้รับแรงดัน -12 โวลท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงการอินเตอร์เฟสกับเครื่องคอมพิวเตอร์

2.7 การเปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอะนาลอก FSK

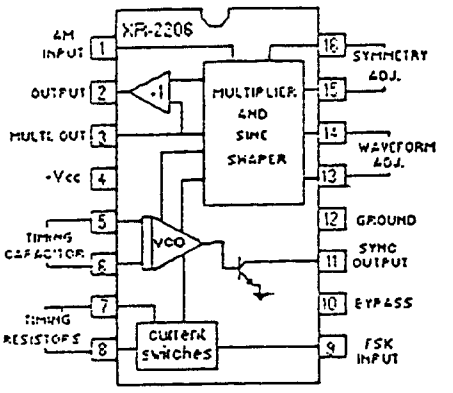
โมเด็มไร้สายที่ได้ทำการออกแบบนั้นใช้ไอซีเบอร์ XR-2206 มาทำหน้าที่เปลี่ยนสัญญาณดิจิตอลเป็นสัญญาณอนาลอก FSK

โครงสร้างภายในของ XR-2206

ไอซีเบอร์ XR-2206 นี้เป็นโมโนลิทิคฟังก์ชั่นเจนเนอเรเตอร์ มีความสามารถในการผลิตคลื่นรูปไซน์ (sine) รูปคลื่นสามเหลี่ยม (triangle) สี่เหลี่ยม (square) แรมพ์ (ramp) ได้โดยที่มีความถี่ตั้งแต่ไม่กี่เฮิร์ทจนถึงหลายกิโลเฮิร์ทโดยต่อกับวงจรภายนอกอีกเล็กน้อย นอกจากนี้ยังสามารถนำไอซี XR-2206 มาควบคุมขนาดและความถี่ (AM หรือ FM) และ Phase Shift of Frequency Shift Keying ได้ด้วย

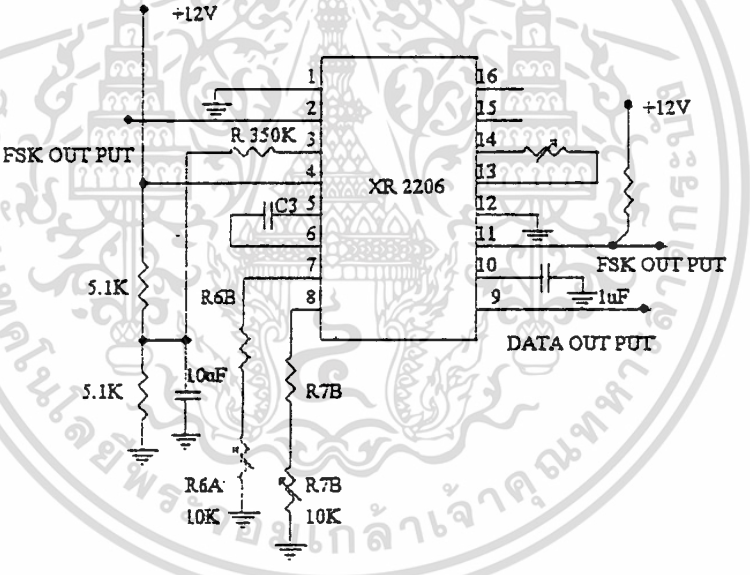
สำหรับ XR-2206 นี้อยู่ในแพคเกจไอซี 16 ขา สามารถที่จะใช้กับไฟเลี้ยง (power supply) ตัวเดียว คือในช่วง 10 ถึง 26 โวลท์ได้ หรืออาจจะใช้ไฟเลี้ยงคู่ในช่วง 5 ถึง 13 โวลท์ ขณะที่ต้องผลิตสัญญาณคลื่นซายน์ (sine) นั้นค่า t.h.d. ของสัญญาณนั้นจะมีค่า 2.5% เมื่อยังไม่มีการปรับแต่ง แต่ก็สามารถปรับให้เหลือเพียง 0.5% ได้โดยการควบคุมของวงจรที่นำมาต่อร่วมโดยที่สัญญาณเอาท์พุทรูปไซน์นี้จะมีขนาดสูงสุด 2 โวลท์ (r.m.s.) และมีเอาท์พุทอิมพิแดนซ์เท่ากับ 600 โอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงบล็อกโครงสร้างแต่ละส่วนของ XR-2206

รูปที่ 2.7 แสดงบล็อกไดอะแกรมแต่ละส่วนของ XR-2206 หัวใจสำคัญของส่วนนี้คือ V.C.O. (Voltage Control Oscillator) ซึ่งเมื่อนำมาใช้เป็นฟรีเควนซี่ ชิฟท์คีย์อิงโมดูเลเตอร์ที่ทำหน้าที่ในการเปลี่ยนสัญญาณดิจิตอลขนาด 0-5 โวลท์เป็นสัญญาณอนาลอกไซน์เวฟ 2 ความถี่ โดยแต่ละความถี่แทนข้อมูล 1 และ 0 จะได้วงจรดังรูปที่ 2.11



รูปที่ 2.8 แสดงการนำไอซี XR-2206 ไปใช้เป็นวงจรฟรีเควนซี่คีย์ชิฟท์คีย์อิงโมดูเลเตอร์

การออกแบบดังกล่าวสามารถอธิบายได้โดยดูรูปและทั้งสองประกอบดังนี้

คาปาซิเตอร์จัดเวลา (Timing Capacitor) ต่อระหว่างขา 5 และ 6 จะเป็นอินพุทของ VCO. มีค่าที่เหมาะสมในการใช้งานอยู่ระหว่าง 1,000 pF - 100uF จะทำงานร่วมกับตัวต้านทานจัดเวลา (Timing Resister) เพื่อใช้กำหนดความถี่ของสัญญาณไซน์เวฟทั้งสอง โดยความถี่ที่ต้องการสามารถปรับแต่งได้ โดยใช้ความสัมพันธ์ตามสมการ

$$\text{ความถี่ (Hz)} = 1 / RC \quad \text{.....(1)}$$

เมื่อ C คือคาปาซิเตอร์จัดเวลา(Timing Capacitor) และ R คือตัวต้านทานจัดเวลา (Timing Resister)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากเราต้องการความถี่สองความถี่ สำหรับแทนสัญญาณดิจิตอลสูงและต่ำ และกำหนด ให้ใช้ค่าคาปาซิเตอร์เพียงค่าเดียว คือคาปาซิเตอร์จัดเวลา (Timing Capacitor) ที่ต่อระหว่างขา 5 และ 6 ดังนั้นค่าตัวต้านทานจัดเวลา (Timing Resister) จะต้องมี 2 ค่า ซึ่งใช้ตัวต้านทานจัดเวลา (Timing Resister) 2 กลุ่มต่อเข้าที่ขา 7 และ 8 โดยกลุ่มที่มีค่าตัวต้านทานมากกว่าจะเป็นตัวสร้างความถี่ต่ำ และกลุ่มที่มีค่าตัวต้านทานน้อยกว่าจะเป็นตัวสร้างความถี่สูง แต่ละกลุ่มจะประกอบด้วยตัวต้านทานที่มีค่าคงที่ และตัวต้านทานที่ปรับค่าได้ที่ต่ออนุกรมกันเพื่อให้ปรับความถี่ได้ถูกต้อง และละเอียดยิ่งขึ้น

สำหรับช่วงของตัวต้านทานจัดเวลา (Timing Resister) ที่เหมาะสมในการใช้งานจะอยู่ในช่วงระหว่าง 4-200 กิโลโอห์ม เพื่อรักษาความคงค่าของอุณหภูมิ (Terperature stability) และความเพี้ยนของสัญญาณซายน์ให้อยู่ในช่วงที่ใช้งานได้อย่างมีประสิทธิภาพ

เมื่อให้สัญญาณดิจิตอลอินพุทขนาด 0-5 โวลท์เข้ามาที่ขาที่ 9 ของไอซี

- ไอซีจะใช้ตัวต้านทานจัดเวลา (Timing Resister) ที่ขา 7 ในการให้กำเนิดสัญญาณรูปซายน์ ถ้าหากระดับสัญญาณที่เข้ามามีขนาดมากกว่า 2 โวลท์ หรือขา 9 มีการเปิดวงจร (open circuit)
- ไอซีจะใช้ตัวต้านทานจัดเวลา (Timing Resister) ที่ขา 8 ในการให้กำเนิดสัญญาณรูปซายน์ ถ้าหากระดับสัญญาณที่เข้ามามีขนาดน้อยกว่า 2 โวลท์

การเกิดคลื่นรูปซายน์นี้เริ่มมาจากส่วน VCO ของไอซีซึ่งส่วนนี้จะผลิตรูปคลื่นได้ 2 ชนิด คือรูปคลื่นแรมพ์ซึ่งจะป้อนไปยังส่วนของวงจรคูณสัญญาณรูปซายด์ (Multiplier and Sine Shaper)อีกทีหนึ่ง และคลื่นรูปสี่เหลี่ยม (rectangular) ซึ่งจะป้อนออกที่ขาเอาท์พุทที่ขา 11 โดยผ่านทรานซิสเตอร์ ซึ่งการผลิตคลื่นนี้ก็ขึ้นอยู่กับคาปาซิเตอร์จัดเวลา (Timing Capacitor) โดยคาปาซิเตอร์จัดเวลานี้จะเริ่มต้นเก็บประจุ เป็นผลทำให้เกิดช่วงพุ่งขึ้นของคลื่นรูปแรมพ์ และที่อีกเอาท์พุทจะได้สัญญาณ High ที่คลื่นรูปสี่เหลี่ยม (rectangular) จนกระทั่งแรงดันไฟฟ้านั้นจะถึงจุด ๆ หนึ่ง ที่จุดนี้จะทำให้ได้สัญญาณคลื่นรูปสี่เหลี่ยมกลับกลายเป็นสัญญาณ “LOW” และคาปาซิเตอร์จัดเวลา (Timing Capacitor) จะเก็บประจุในทิศทางตรงกันข้ามกับตอนต้นเป็นผลทำให้เกิดช่วงตกลงของคลื่นรูปแรมพ์ซึ่งก็จะตกลงจนถึงจุด “firing voltage” เช่นกันที่จุดนี้ จะทำให้คลื่นรูปสี่เหลี่ยม (rectangular) กลับกลายเป็นระดับสัญญาณ “High” และกระบวนการต่าง ๆ ก็จะกลับไปกลับมาเช่นนี้เหมือนเดิม

รูปคลื่นแรมพ์ที่ได้จากส่วนของ VCO ของไอซี XR-2206 นี้ จะถูกนำไปเข้ายังส่วนของ multiplier and sine shaper block อีกทีซึ่งส่วนนี้นั้นทำหน้าที่คล้ายกับวงจรขยายความต่างซึ่งจะทำให้เอาท์พุทอิมพิแดนซ์ ที่ขา 3 มีค่าสูง และที่ขา 2 จะเป็นเอาท์พุทบัฟเฟอร์ ที่มีค่าอิมพิแดนซ์ เท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ 600 โอห์ม ในกรณีที่เปิดวงจรที่ขา 13 และ 14 นั้นจะมีผลทำให้เอาท์พุท ที่ขา 2 และ 3 นั้นจะให้คลื่นรูปแรมพ์ออกมา แต่ถ้าต่อตัวต้านทานที่มีค่าประมาณ 200 - 300 โอห์มที่ขา 13 และ 14 จะทำให้ยอดของคลื่นรูปแรมพ์ถูกตัดออก ทำให้เอาท์พุทที่ขา 2 และ 3 นั้น ผลิตคลื่นชายน์ออกมา ถ้ามีการปรับแต่งที่ถูกต้องเหมาะสม จะทำให้คลื่นรูปชายน์ที่ได้มีความเพี้ยนเพียง 0.5% เท่านั้น การควบคุมอัตราการขยายสัญญาณเอาท์พุท FSK ทำได้โดยการควบคุมที่ขา 1 หรือขา 3 โดยในโมเด็มไร้สายนี้ใช้การควบคุมอัตราการขยายที่ขา 3 โดยจากรูปที่ จะเห็นว่าที่ขา 3 จะต่อกับอินพุทของบัฟเฟอร์แอมปลิไฟเออร์ที่มีอัตราการขยายเท่ากับ 1 ซึ่งทำให้ขาที่ 2 มีเอาท์พุทอิมพิแดนซ์เท่ากับ 600 โอห์ม ดังนั้นเราสามารถใช้ขา 3 ทำหน้าที่เป็นตัวควบคุมแรงดัน สำหรับการกำหนดอัตราการขยาย โดยใช้วงจรแบ่งแรงดัน (potential divider) ต่อเข้าที่ขา 3 ทำให้สัญญาณอินพุทที่เข้าไปยังบัฟเฟอร์แอมปลิไฟเออร์ ถูกควบคุมศักดาไฟฟ้าตามวงจรแบ่งแรงดันนั้น โดยปกติแรงดันดังกล่าวจะเป็นครึ่งหนึ่งของไฟเลี้ยงวงจร

ขั้นตอนการคำนวณค่าของตัวต้านทานและค่าของคาปาซิเตอร์

สำหรับโมเด็มไร้สายนี้ได้ใช้เทคนิคการมอดูเลทสัญญาณแบบ FSK ซึ่งได้กำหนดความถี่ต่าง ๆ ตัวอย่างเช่น ถ้าใช้ความถี่ตามมาตรฐานของ CCITT V.23 คือที่บอดเรท 1200 บิทต่อวินาที ความถี่ f_1 และ f_2 ค่า 1300 และ 2100 Hz ตามลำดับดังนั้นสามารถคำนวณค่าของความต้านทานและค่าคาปาซิแตอร์ได้ดังนี้

1.ทำการกำหนดค่าของคาปาซิเตอร์จัดเวลา (Timing Capacitor) ซึ่งควรจะอยู่ในช่วง 1 nF ถึง 100 uF ดังนั้นในโมเด็มไร้สายนี้ได้กำหนดค่าของคาปาซิเตอร์จัดเวลาไว้เท่ากับ 22 nF

2.ทำการคำนวณค่าของตัวต้านทานจัดเวลา (Timing Resistor) จากสมการที่ 1 และความถี่กำหนด จะได้ดังนี้

- ค่าของตัวต้านทานจัดเวลาที่ขา 7 ซึ่งขานี้ใช้กับสัญญาณดิจิตอลที่มีลอจิกเป็น "1" และเนื่องจากความถี่ที่กำหนดลอจิก "1" นั้นมีค่าเท่ากับ 1300 Hz ดังนั้นจะได้

ค่าความต้านทานที่ขา 7 = $1 / (1300 \times 22 \times 10^{-9})$

= 34965.034965 โอห์ม

- ค่าของตัวต้านทานจัดเวลาที่ขา 8 ซึ่งขานี้ใช้กับสัญญาณดิจิตอลที่มีลอจิเป็น "0" และเนื่องจากความถี่ที่กำหนดลอจิก "0" นั้นมีค่าเท่ากับ 2100 Hz ดังนั้นจะได้

ค่าความต้านทานที่ขา 8 = $1 / (2100 \times 22 \times 10^{-9})$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

= 21645.021645 โอห์ม

2.8 การเปลี่ยนสัญญาณอะนาลอก FSK ให้เป็นสัญญาณดิจิตอล

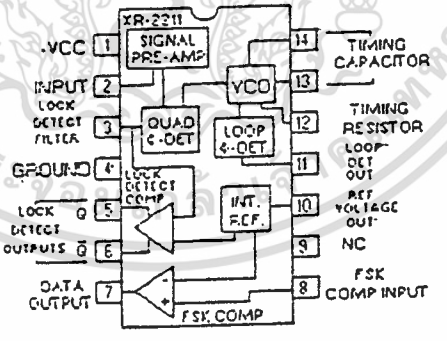
ภาครับสัญญาณ FSK นี้จะรับสัญญาณจากภาครับสัญญาณ RF โดยที่ภาคนี้จะทำการแปลงสัญญาณ FSK ที่มี 2 ความถี่ให้กลายเป็นสัญญาณดิจิตอล 0 (แทนด้วยแรงดันประมาณ 0 โวลท์) และ 1 (แทนด้วยแรงดันประมาณ 5 โวลท์)

ภาครับนี้ใช้ไอซีเบอร์ XR-2211 ซึ่งได้ถูกออกแบบมาโดยเฉพาะเพื่อทำการแปลงสัญญาณ FSK ให้เป็นสัญญาณดิจิตอล (FSK Demodulation) การซิงโครไนเซชั่นข้อมูล (Data Synchronization) การถอดรหัสสัญญาณเสียง (Tone Decoding) การดีเทคสัญญาณ FM (FM Detection) และการดีเทคสัญญาณคลื่นพาห์ (Carrier Detection)

ลักษณะโดยทั่วไปของไอซี XR-2211

ไอซีเบอร์ XR-2211 นี้เป็นไอซีที่ทำงานแบบเฟสล็อกลูป (phase Locked Loop : PLL) ไฟเลี้ยงที่ใช้กับไอซีนี้อยู่ในช่วงตั้งแต่ 4.5 ถึง 20 โวลท์ และสามารถทำงานในย่านความถี่ตั้งแต่ 0.01 Hz จนถึง 300 kHz นอกจากนี้ยังสามารถรับสัญญาณอินพุทในช่วงที่กว้างได้ตั้งแต่ 2 มิลลิโวลท์จนถึงสัญญาณที่เข้ามามีขนาดน้อยกว่า 2 โวลท์ อีกอย่างหนึ่งคือ สามารถใช้งานร่วมกับอุปกรณ์ทางลอจิกที่เป็นมาตรฐานได้แก่ ตระกูล DTL, TTL และ ECL ได้อีกด้วย

โครงสร้างภายในของไอซี XR-2211



รูปที่ 2.9 แสดงโครงสร้างภายในของไอซี XR-2211

โครงสร้างภายในแสดงดังรูปที่ 3:5 โดยมีโครงสร้างหลักเป็นวงจรเฟสล็อกลูปซึ่งประกอบด้วยวงจรปรีแอมปลิฟายเออร์ (Preamplifier) , วงจรคูณสัญญาณอนาลอก (Analog Multiplier) ซึ่งใช้เป็นวงจรเฟสดีเทกเตอร์ และวงจร VCO (Voltage Control Oscillator) โดยวงจรปรีแอมปลิฟายเออร์นี้ใช้สำหรับขยายสัญญาณอินพุทที่มีขนาดต่ำ ๆ (สูงกว่า 2 มิลลิโวลท์) ให้มีขนาดสูงขึ้น ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรเฟสดีเทกเตอร์แบบคูณนี้ทำงานคล้ายกับเอกคลูซีฟออร์เกต ส่วนวงจร VCO นั้นจะถูกควบคุมความถี่โดยตัวต้านทาน R_O และกระแสจากวงจรเฟสดีเทคเตอร์

แรงดันอ้างอิง (Reference Voltage : V_R ที่ขา 10) แรงดันที่ขา 10 นี้ใช้เป็นแรงดันอ้างอิงสำหรับแรงดันที่ขา 5, 8, 10 และ 11 โดยที่ขา 10 นี้จะต้องต่อคาปาซิเตอร์ขนาด 0.1 uF กับกราวนด์เพื่อบายพาสสัญญาณความถี่สูงลงกราวนด์ และให้วงจรทำงานสม่ำเสมอ

สัญญาณเอาท์พุทจากวงจรลูปเฟสดีเทคเตอร์ (Loop Phase Detector Output) ที่ขา 11 นี้เป็นเอาท์พุทที่มีความต้านทานสูงใช้สำหรับลูปเฟสดีเทกเตอร์โดยมีตัวต้านทาน R_1 และคาปาซิเตอร์ C_1 ทำหน้าที่เป็นวงจรลูปฟิลเตอร์ของเฟสล็อกลูป ในกรณีที่ยังไม่มีสัญญาณอินพุทหรือไม่มีความแตกต่างทางเฟสของวงจรเฟสล็อกลูป ระดับแรงดันที่ขา 11 นี้จะมีค่าใกล้เคียงกับแรงดันอ้างอิง VR

การควบคุมความถี่ของวงจร VCO ความถี่ของวงจร VCO ถูกควบคุมจากตัวต้านทาน R_O ซึ่งสามารถหาได้จากสมการที่ 2

$$f_O = 1 / (R_O C_O) \quad HZ \qquad .....(2)$$

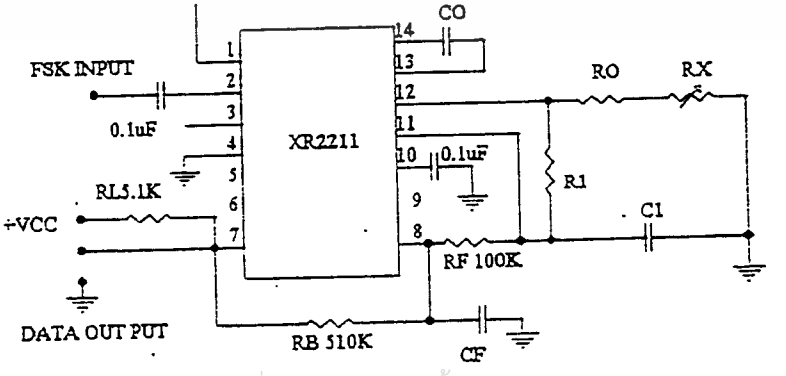
โดยที่ C_O คือคาปาซิเตอร์ที่ต่อระหว่าง 13 กับ 14 และเพื่อความเสถียรภาพของวงจรควรใช้ตัวต้านทาน R_O มีค่าอยู่ในช่วงระหว่า 10 กิโลโอห์ม

คาปาซิเตอร์ควบคุมวงจร VCO (ที่ขา 13 และ 14) ความถี่ที่ได้จากวงจร VCO แปรผกผันกับค่าของปาซิเตอร์ C_O ที่ต่อระหว่างขา 13 และ 14 คาปาซิเตอร์ C_O ต้องใช้แบบไม่มีขั้วอยู่ในย่าน 200 พีโคฟารัดจดถึง 10 ไมโคฟารัด

การปรับความถี่ของวงจร VCO ควรใช้ตัวต้านทานที่สามารถปรับค่าได้ต่ออนุกรมกับตัวต้านทานค่าหนึ่ง โดยผลรวมของตัวต้านทานทั้งสองใช้แทนตัวต้านทาน R_O

การนำไอซี XR-2211 ไปใช้ในการดีเทกสัญญาณ FSK

การนำไอซี XR-2211 ไปใช้เพื่อทำการดีเทกรหัสสัญญาณ FSK แสดงรูป 2.10



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.10 แสดงถึงการนำไอซี XR-2211 ไปใช้เพื่อทำการดีเทกรหัสสัญญาณ FSK

จากรูปที่ 2.10 ตัวต้านทาน R_o และคาปาซิเตอร์ C_o ใช้เพื่อกำหนดความถี่กลางของเฟสล็อกลูป (f_o) ตัวต้านทาน R_1 นั้นใช้เพื่อกำหนดแบนด์วิตธ์, คาปาซิเตอร์ C_1 ใช้กำหนดค่าคงที่ทางเวลาของลูปฟิลเตอร์และค่าลูปแดมปิง, คาปาซิเตอร์ C_F และตัวต้านทาน R_F ทำหน้าที่เป็น One Role Post-detection สำหรับสัญญาณเอาท์พุท, ตัวต้านทาน R_8 (มีค่าประมาณ 510 กิโลโอห์ม) ซึ่งต่อระหว่าง 7 และ 8 ทำหน้าที่เป็นตัวป้อนกลับทางบวก (Positive Feedback)

ขั้นตอนการกำหนดค่าตัวต้านทานและคาปาซิเตอร์

1.คำนวณความถี่กลางของเฟสล็อกลูป f_o ดังสมการที่ 3 โดยค่า f_1 และ f_2 คือความถี่ทั้งสองของสัญญาณอินพุท FSK

$$f_o = (f_1 + f_2)/2 \qquad .....(3)$$

ถ้าทางด้านส่งนั้นได้กำหนดความถี่ไว้ตามมาตรฐานของ CCITT V.23 คือที่บอดเรท 1200 บิท ต่อวินาที ซึ่งมีความถี่ f_1 และ f_2 มีค่า 1300 และ 2100 Hz ตามลำดับดังนั้นจะได้

$$f_o = (1300 + 2100)/2$$
$$= 1700 \text{ Hz}$$

2.เลือกค่าความต้านทาน R_o ซึ่งค่า R_o นี้ควรจะอยู่ในช่วง 10 กิโลโอห์มจนถึง 100 กิโลโอห์ม สำหรับค่าความต้านทานของ R_o ที่ใช้ในโมเด็มไร้สายนั้นมีค่า 10 กิโลโอห์ม

3.คำนวณค่าคาปาซิเตอร์ C_o ดังสมการที่ 4

$$C_o = 1/R_o f_o \qquad .....(4)$$
$$C_o = 1/(10 \times 10^3 \times 1700)$$
$$= 58.8 \text{ nF}$$

4.คำนวณค่าความต้านทาน R_1 จากสมการที่ 5

$$R_1 = R_o \times [f_o/(f_1 - f_2)] \qquad .....(5)$$

จากความถี่จากขั้นตอนที่ 1 และค่าความต้านทานในขั้นตอนที่ 2 จะได้ค่าของ R_1 ดังนี้

$$R_1 = 10 \times 10^3 \times [1700/(2100 - 1300)]$$
$$= 21{,}250 \text{ โอห์ม}$$

5.คำนวณค่าคาปาซิเตอร์ C_1 เพื่อกำหนดลูปแดมปิง (Loop Damping) ซึ่งควรจะมีค่าประมาณ 0.5 ดังนั้นจะได้ค่าของคาปาซิเตอร์ C_1 ดังสมการที่ 6

$$C_1 = C_o/4 \qquad .....(6)$$

จากค่าของ C_o ตามขั้นตอนที่ 3 จะได้ค่าของ C_1 ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$C_1 = 58.8 \times 10^{-9} \times 0.25$

$= 1.47 \times 10^{-8}$

$= 14.7$ nF

6.คำนวณค่าของคาปาซิเตอร์ C_F ถ้ากำหนดตัวต้านทาน R_F มีขนาด 100 กิโลโอห์มและตัวต้านทาน R_B มีขนาด 510 กิโลโอห์ม ดังนั้นจะหาค่าของ C_F ได้จากสมการที่ 7

$C_F = 3 /$ Baud Rate uF .....(7)

เนื่องจากบ็อดเรทนั้นใช้เท่ากับ 1200 บิทต่อวินาทีดังนั้นจะได้ค่า C_F ดังนี้

$C_F = 3 / 1200$

$= 2.5$ nF

หมายเหตุ

สำหรับค่าของคาปาซิเตอร์ที่ใช้จริงในวงจรนั้นให้ใช้ค่าที่ใกล้เคียงกับค่าที่ได้จากการคำนวณมากที่สุด ส่วนค่าความต้านทานที่ใช้จริงในวงจรนั้นให้ใช้ค่าที่ใกล้เคียงที่น้อยกว่าต่ออนุกรมกับความต้านทานที่ปรับค่าเพื่อให้สามารถปรับแต่งให้ได้คุณสมบัติของวงจรที่ดีที่สุด

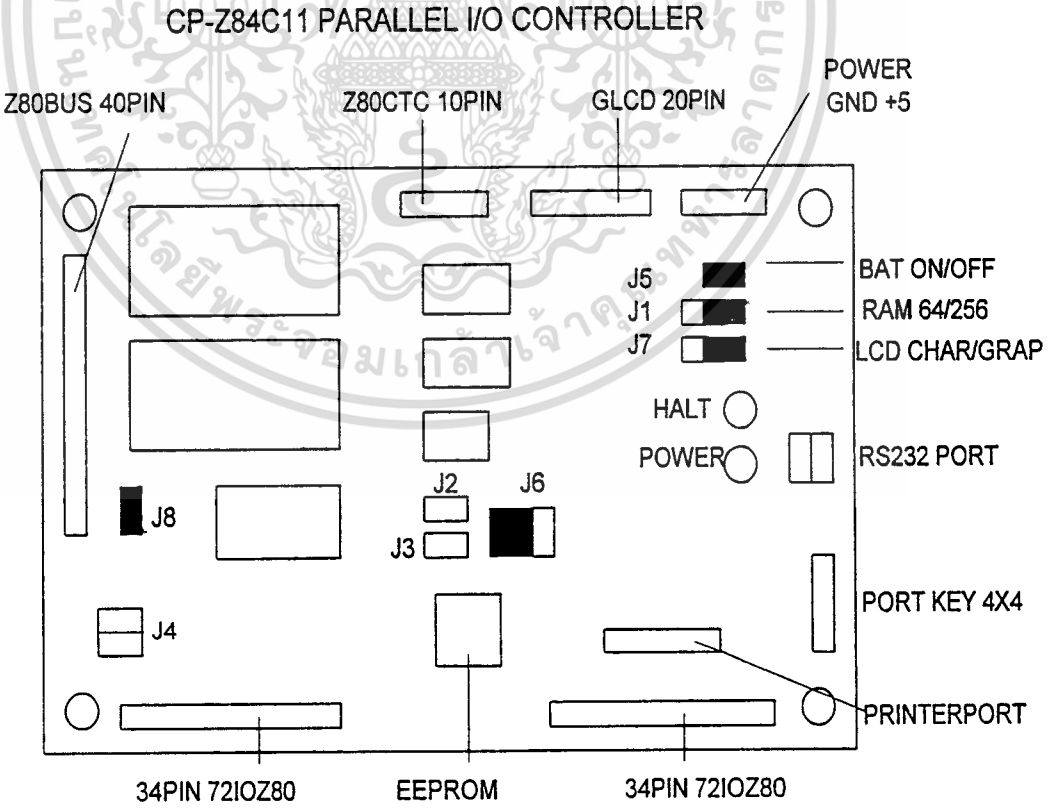
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

037134

บทที่ 3
ข้อมูลของบอร์ด CP-Z84C11

3.1 ข้อมูลของบอร์ด CP-Z84C11

Z84C11 ของบริษัท ไซล็อก โดยมีการทำงานด้วยความเร็วสูง (HIGH SPEED OPERATION) โดยมีให้เลือก 2 รุ่น คือ ทำงาน ที่ความถี่ 6 Mhz. , และที่ ซีมอส 10 Mhz. ในบอร์ดนี้จะใช้ CPU รุ่น ความถี่ 10 Mhz CPU เบอร์นี้ของ ไซล็อก จะเป็นแบบ ซีมอส ทำให้กินกระแสต่ำมากในขณะที่ สภาวะไม่ทำงานอยู่ใน(STOP MODE) จะกินกระแสเพียง 50 ไมโครแอม ข้อดีอีกอย่างหนึ่งของ CPU ตัวนี้ก็คือ ถึงแม้จะ ทำงานที่ความถี่ 10 Mhz ก็ตามก็สามารถกำหนดให้ทำงานที่ความถี่ 5 Mhz ในกรณีที่ใช้ แรม (RAM) หรือ Read Only Memory ที่มีการทำงานความเร็วต่ำ ๆ โดยเมื่อ เริ่มจ่ายไฟ แล้ว CPU จะทำงาน เพียง 5 Mhz เท่านั้นตอนเริ่มต้น และเมื่อต้องการทำงาน 10 Mhz ก็สามารถตั้งค่า ได้ด้วยโปรแกรม รูปแบบลักษณะของ CPU จะเป็น แพคเกจ แบบ 100 PIN QFP



รูปที่ 3.1 ลักษณะทั่วๆไปของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ (MEMORY)

CP-Z84C11 สามารถใส่หน่วยความจำได้สูงสุด 64 กิโลไบท์ โดยแบ่งเป็น

SOCKET 1 อีพรอม สามารถใส่ อีพรอม ขนาด 32 กิโลไบท์ ได้โดยเป็นเบอร์ 27256 มีหน่วยความจำเริ่มจาก ตำแหน่ง 0000H ถึง ตำแหน่ง 7FFFH

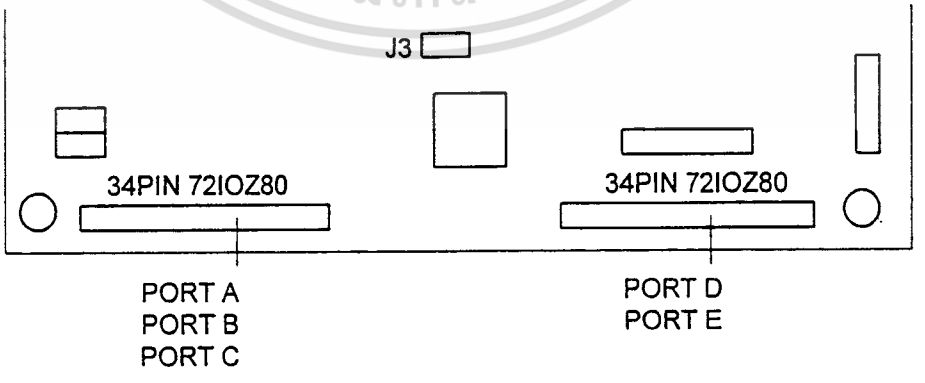
SOCKET 2 แรม สามารถใส่ แรม ขนาด 8-32 กิโลไบท์ ได้โดยเป็นเบอร์ 6264 หรือ 62256 มีหน่วยความจำเริ่มตำแหน่ง 8000H ถึง FFFFH โดย เซท ตำแหน่ง จัมเปอร์ ในการกำหนดเบอร์ไอซี ดังรูป 3.2

รูปที่ 3.2 แสดงการ เซท ตำแหน่ง จัมเปอร์

หน่วยความจำ แรม ส่วนนี้เราสามารถต่อ แบตเตอรี่ 3.6V. ใช้ แบคอัพ ข้อมูลของหน่วยความจำนี้ได้ด้วย

พอร์ทสื่อสาร (PORT)

CP- Z84 C11 จะมีพอร์ทใช้งาน 40 บิท สำหรับใช้เป็น อินพุท เอาท์พุท พอร์ท หรือรวม 5 พอร์ท ด้วยกัน โดยทั้ง 5 พอร์ทนี้จะเป็นพอร์ทในตัว CPU โดยมีตำแหน่งต่อออกมาดังรูปที่3.3



รูปที่ 3.3 แสดงพอร์ทใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ท ที่ต่อออกมานี้จะอยู่เป็น ขั้วต่อ 34 ขา (72IOZ80) สามารถต่อร่วมใช้กับอุปกรณ์สนับสนุนต่าง ๆ

ช่องทางของสัญญาณ	ตำแหน่งของพอร์ท
พอร์ทข้อมูล A	50 H
พอร์ทข้อมูล B	51 H
พอร์ทข้อมูล C	52H
พอร์ทข้อมูล D	30 H
พอร์ทข้อมูล E	40 H

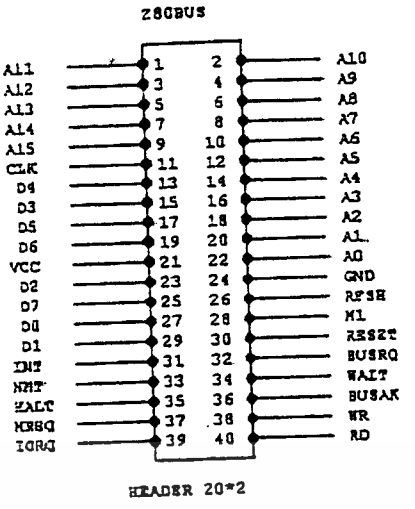
ช่องควบคุมของสัญญาณ	ตำแหน่งพอร์ท
พอร์ทกำหนดสถานะ A	54 H
พอร์ทกำหนดสถานะ B	55 H
พอร์ทกำหนดสถานะ C	56 H
พอร์ทกำหนดสถานะ D	34 H
พอร์ทกำหนดสถานะ E	44 H

สามารถสั่งให้พอร์ทของ Z84C11 เป็นอินพุท หรือ เอาท์พุท ได้โดยอิสระ บิท ต่อ บิท โดยถ้าเซ็ตค่าออกที่พอร์ทกำหนดสถานะ (PORT DATA DIRECTION REGISTER) ถ้าให้บิทใดเป็น "1" ก็คือให้พอร์ทข้อมูลของบิทนั้น ๆ เป็น เอาท์พุทพอร์ท (ถ้า SET PORT นั้นเป็นพอร์ท เอาท์พุท : ค่าเริ่มต้นจะเป็นศูนย์) และถ้าต้องการให้เป็นพอร์ทอินพุท ก็ SET ค่าออกที่พอร์ทกำหนดสถานะเป็นค่า "0"

ขาสัญญาณของ Z80

CP-Z84C11 จะมีส่วนขยายระบบได้ทาง Z80 บัส โดยมีลักษณะขาเหมือนกับ CPU Z80 40 ขา CPU Z84C11 นี้มีส่วนวงจร รีเซทเมื่อจ่ายไฟ ในตัวอยู่แล้วไม่จำเป็นต้องใช้วงจรรีเซทเมื่อจ่ายไฟ ต่อจากภายนอก เช่น CPU เบอร์อื่น ๆ ไปต่อร่วมกับ Z80 บัส อื่น ๆ ที่มีตัวต้านทาน , ตัวเก็บประจุ ต่ออยู่แล้วอาจจะทำให้ CPU Z84C11 นี้อยู่ในสภาพ รีเซท ตลอดเวลาได้ บอร์ด CP-Z84C11 จึงมี จัมเปอร์ ในการเลือกไม่ให้ต่อ รีเซท จากขาภายนอกได้ โดยถ้าทีการต่อตัวต้านทาน , ตัวเก็บประจุ ที่ ที่ขารีเซท ภายนอกก็ให้ถอด จัมเปอร์รีเซท ออก ดังรูปที่ 3.4

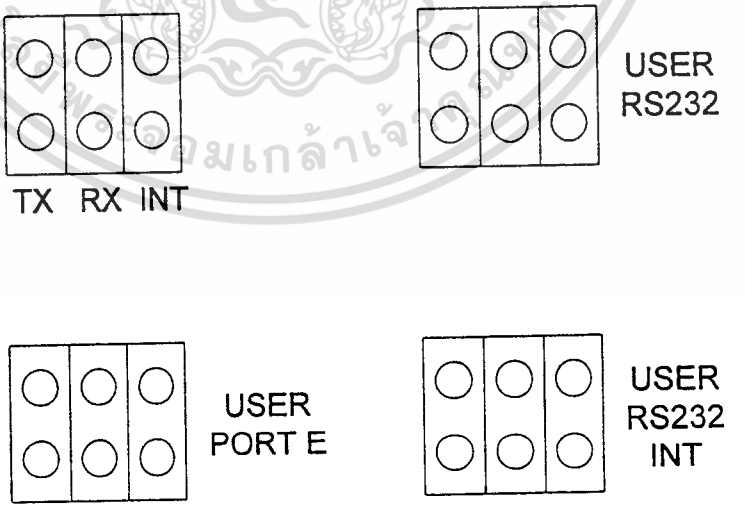
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 ตำแหน่ง Z80 บัส

การใช้งานพอร์ทอนุกรม

เป็นจุดต่อ (CONNECTOR) ที่ต่อจากวงจร RS-232 ที่ใช้เปลี่ยนระดับสัญญาณต่อเข้าพอร์ท PE 6 , PE 5 , และเข้าขา INT ของ Z84C11 โดยถ้าไม่ใช้ พอร์ท RS-232 ก็ใช้ถอด จัมเปอร์ออกเพื่อจะใช้ PE 6 , PE 5 ได้อย่างอิสระ และอีกส่วนหนึ่งคือ จัมเปอร์ ของ INT นั้นเราสามารถเซท ให้ พอร์ท RS-232 รับข้อมูลในรูปแบบของขบวนการขัดจังหวะ (INTERRUPT) เช่นเดียวกับการขัดจังหวะ จาก RS-232 พอร์ท ได้

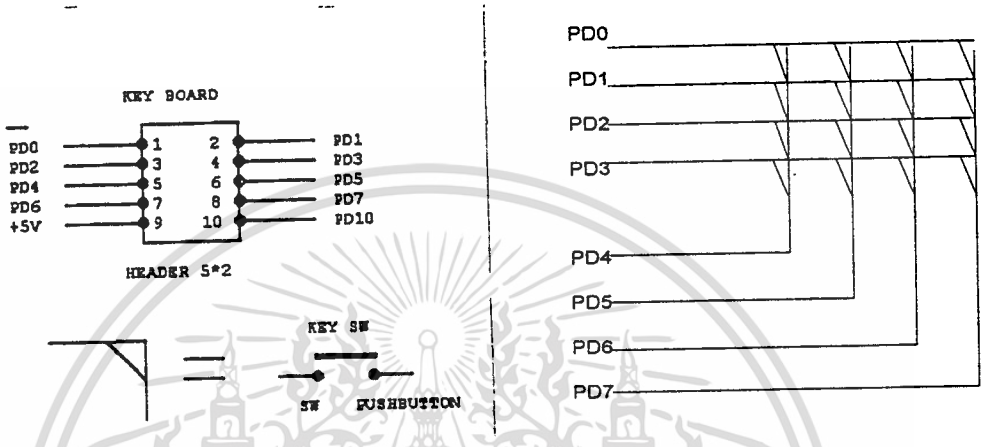


รูปที่ 3.5 การ เซทพอร์ท RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์บอร์ด (KEYBOARD)

เป็นจุดต่อขนาด 10 PIN ต่อจาก PORT PD โดยมี R-PULL-UP 10K ต่ออยู่ด้วย โดยจุดต่อ 10 PIN นี้จะสามารถต่อเป็นรูปแบบคีย์บอร์ด ขนาด 4x4 คือ 16 คีย์ โดยเขียนโปรแกรมอ่านค่าคีย์บอร์ดขึ้นหรือจะนำไปใช้เป็นพอร์ตอิสระในงานอื่น ๆ ก็ได้



รูปที่ 3.6 การต่อกับคีย์บอร์ด

LCD J7

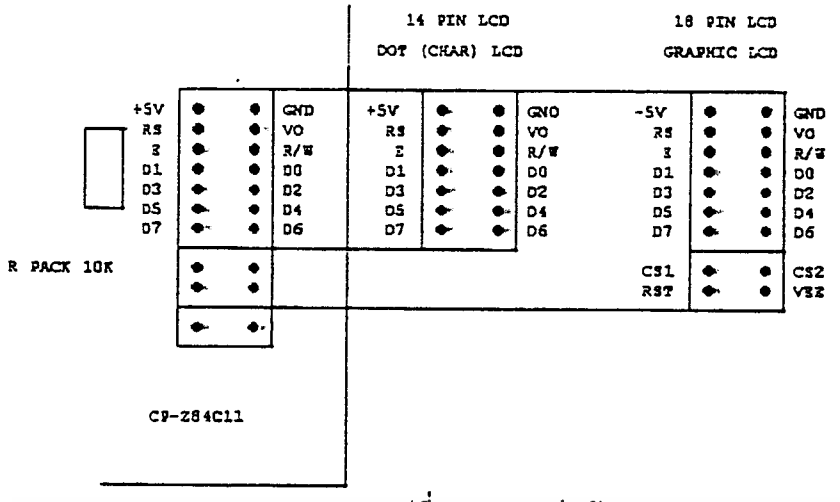
เป็น จุดต่อ 3 ขา ใช้เลือกว่าจะต่อ LCD โมดูล ประเภทใดคือ LCD โมดูล แบบตัวอักษร หรือ LCD โมดูล แบบ กราฟฟิค

รูปที่ 3.7 การเลือก LCD

GLCD

เป็น จุดต่อขนาด 20 ขา โดยสามารถต่อ LCD ได้ 2 แบบคือ แบบจุด หรือ แบบกราฟฟิค โดยเราเลือกต่อได้ถ้าเป็น ชนิดจุด คือ ตัวอักษรก็จะใช้ 14 ขา ต่อได้ดังรูปที่ 3.8 หรือถ้าจะเลือกต่อแบบ กราฟฟิค ก็จะต่อใช้งาน 18 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 การต่อกับ LCD

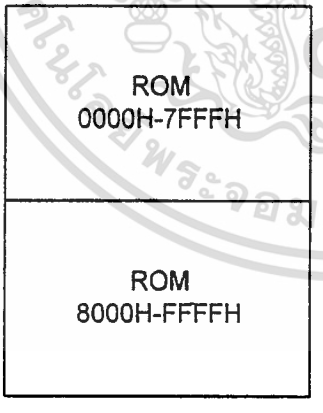
ตำแหน่งของพอร์ท LCD

ช่องสัญญาณ	ตำแหน่งพอร์ท
เขียนข้อมูลเริ่มต้น	80 H
เขียนข้อมูลลงหน่วยความจำ	82 H
อ่านสถานะและตำแหน่ง	84 H

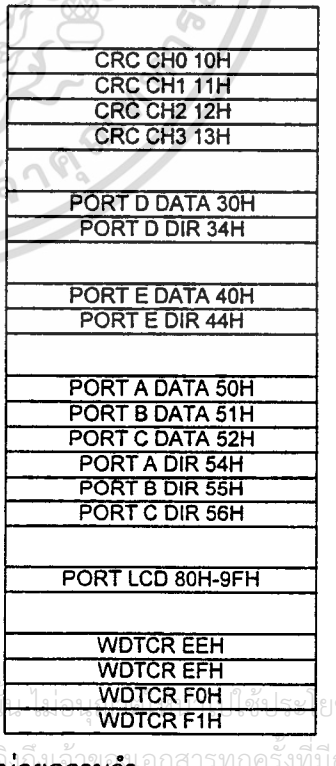
แผนผังการใช้หน่วยความจำ

CP-Z84C11 จัดหน่วยความจำ ออกเป็น 2 ส่วน คือ

MEMMORY CP-Z84C11

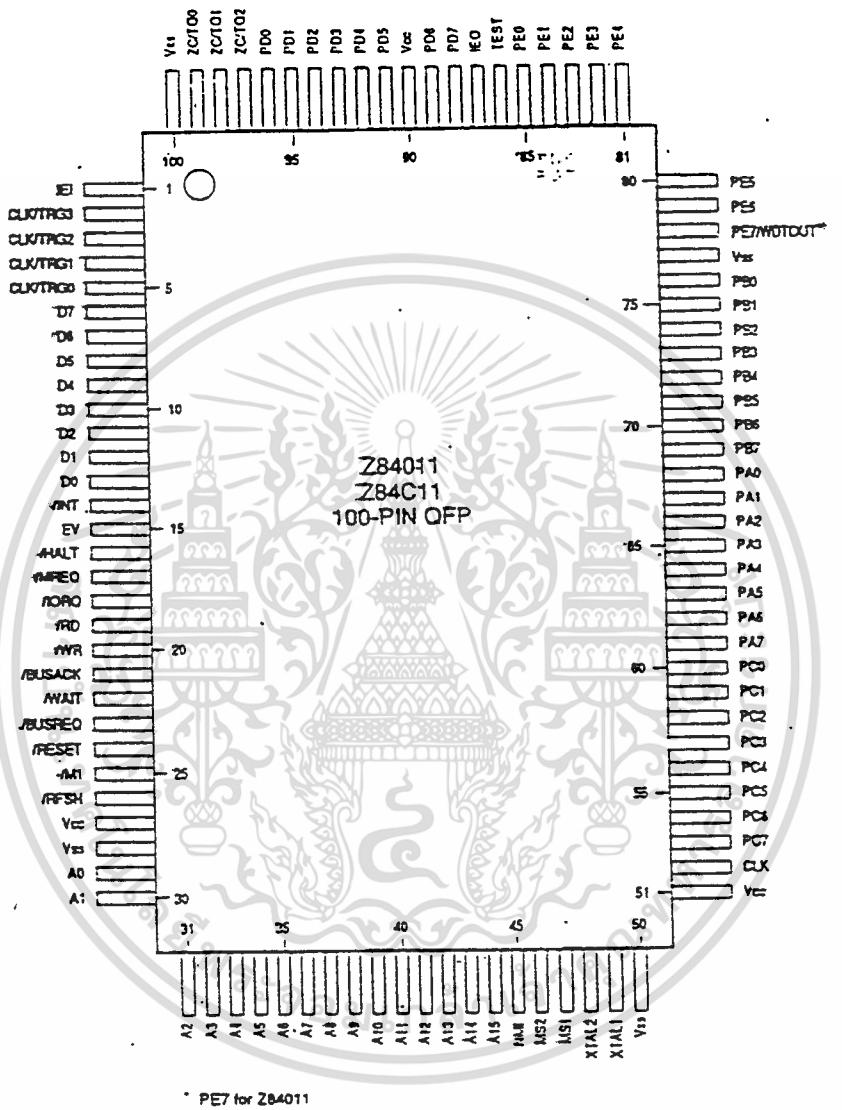


I/O PORT CP-Z84C11



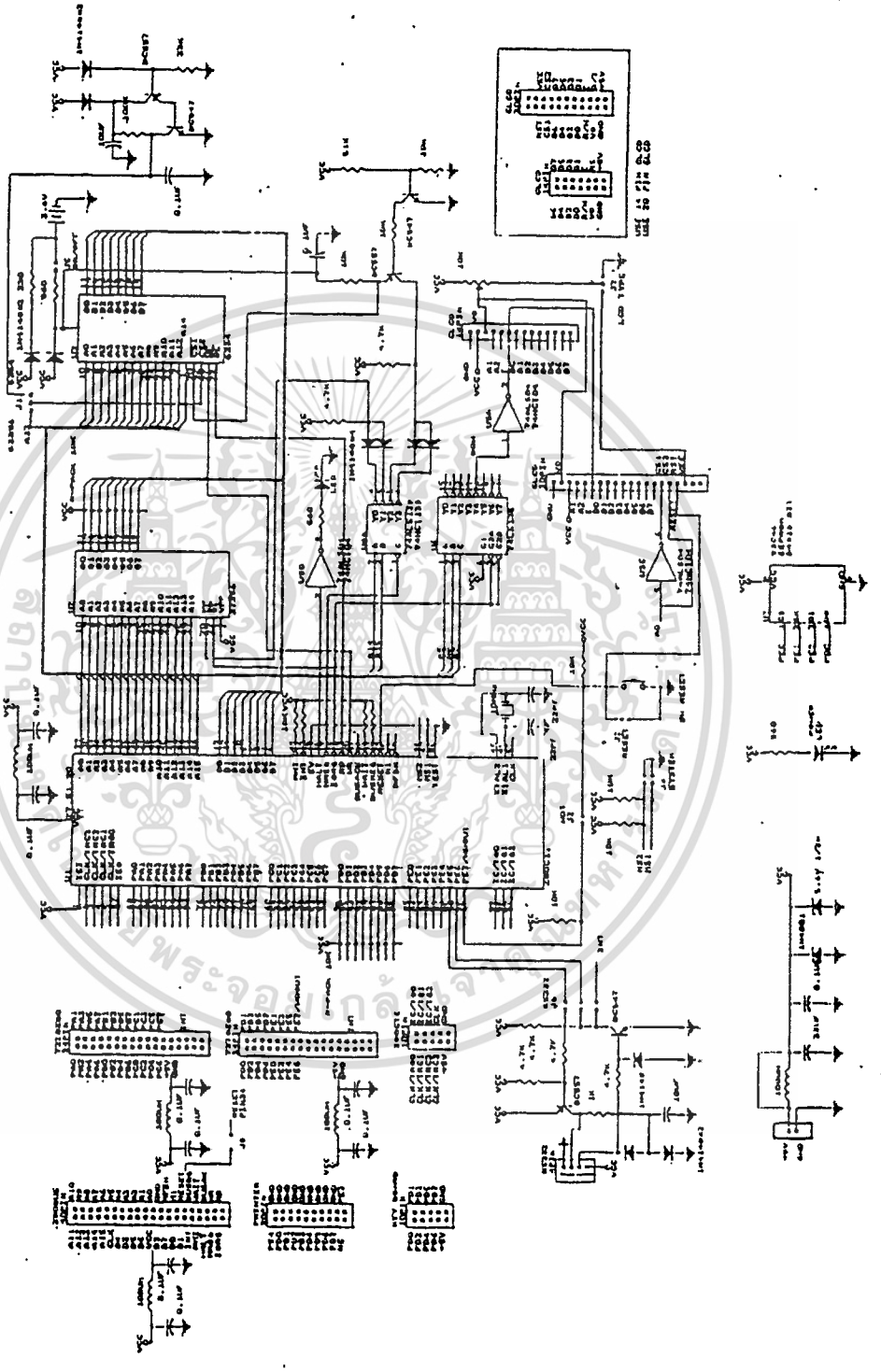
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.9 แสดงแผนผังของหน่วยความจำ



รูปที่3.10 แสดงรายละเอียดของขา Z84C11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงวงจรโดยละเอียดของบอร์ด CP-Z84C11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การควบคุมแอลซีดีกราฟฟิค (LCD GRAPHIC)

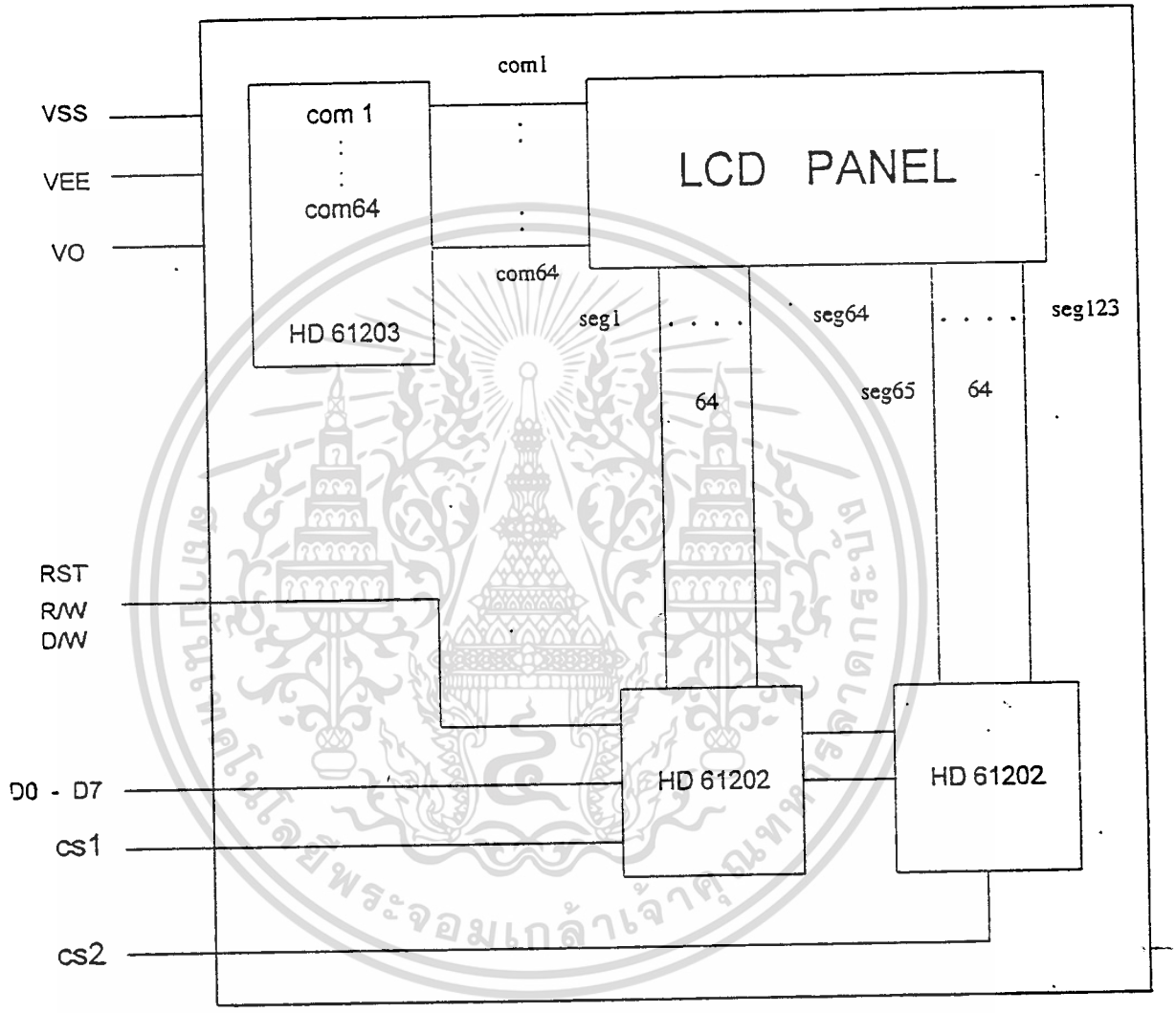
	ตำแหน่ง	ข้อมูล
1) แสดงผล เปิด/ปิด	80-81	(3E,ON) (3F,OFF)
2) บรรทัดแรก	80-81	(CO-F7)
3) เลือกหน้าจอ	80-81	(B8-B7)
4) เลือกคอลัมส์	80-81	(40-7F)
5) สถานะของพอร์ท LCD	84-85	
6) พอร์ทอ่านข้อมูล	82-83	(ข้อมูล)
7) พอร์ทเขียนข้อมูล	86-87	

ขาที่ใช้เชื่อมต่อ

ท	สัญลักษณ์	ระดับแรงดัน	หน้าที่การทำงาน
1 2 3	VSS VDD VO		กราวด์ แรงไฟสำหรับวงจรลอจิก แรงไฟสำหรับขับแอลซีดี
4	D/I	H/L	H ; ข้อมูลอินพุท L ; โครงสร้างรหัสอินพุท
5	R/W	H/L	H ; อ่านข้อมูล L ; เขียนข้อมูล
6	E	H, H-H-L	สัญญาณอินาเบิ้ล
7-14	DB0-DB7	H/L	เส้นทางเดินข้อมูลของรหัส
15	CS1	H	เลือก IC1
16	CS2	H	เลือก IC2
17	RST	L	รีเซ็ท
18	VEE		แรงดันไฟลบ
19-20	N.C		

ตารางที่ 3.1 แสดงขาที่ใช้เชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 บล็อคไดอะแกรมและแหล่งจ่ายไฟ

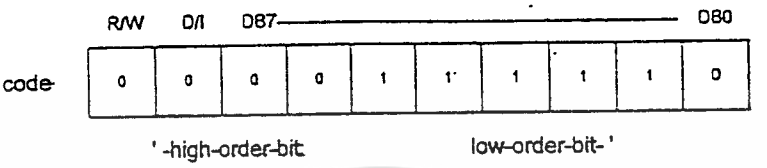
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทขยายการรหัสควบคุม (DETAILED EXPLANTION)

1. **เปิด/ปิด หน้าจอ** เป็นรหัสคำสั่งควบคุมการเปิด/ปิดของจอแอลซีดี

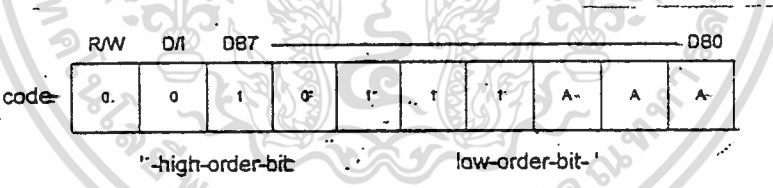
โดยที่ บิท D = 0 จะเป็นรหัสเปิด

บิท D = 1 จะเป็นรหัสปิด



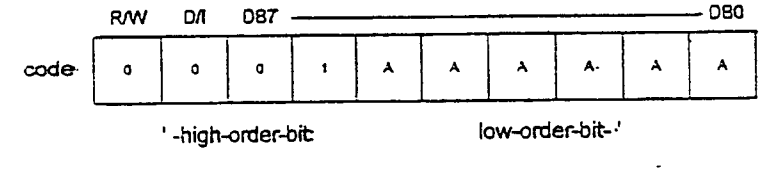
2. **กำหนดตำแหน่งของเส้นเริ่มต้น** ตำแหน่งจะถูกกำหนดโดย AAAAAA (**เลขชี้กำลัง**) จะเป็นการกำหนดค่าเป็นบรรทัดเริ่มต้นการทำงาน

3. **กำหนดตำแหน่งของบรรทัด** จอแอลซีดีจะแบ่งได้เป็น 8 บรรทัด กำหนดโดย AAA (เลขฐาน 2)



4. **กำหนดตำแหน่งพอยเตอร์ของหน่วยความจำ**

เป็นการกำหนดตำแหน่งของหน่วยความจำของส่วนแสดงผล **กำหนดโดย** AAAAAA (เลขฐาน 2) โดยค่าที่กำหนดจะเพิ่มเองโดยอัตโนมัติทุก ๆ ครั้งที่มีการอ่านหรือเขียนข้อมูล



5. ไบท์แสดงสถานะของระบบ

บิท BUSY แสดงความพร้อมของจอแอลซีดี เมื่อมีการทำงานอื่นอยู่

เซ็ต "0" แสดงความพร้อมที่จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิท ON/OFF แสดงสถานะเปิด/ปิด ของจอแอลซีดี

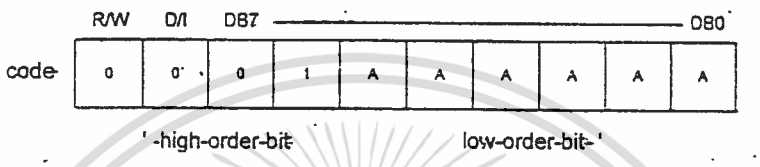
เซ็ต “1” เมื่ออยู่ในสถานะปิด

เซ็ต “0” เมื่ออยู่ในสถานะเปิด

บิท RESET เริ่มระบบใหม่

เซ็ต “1” แสดงระบบว่ามีการเริ่มทำงานอยู่

เซ็ต “0” แสดงว่าระบบกำลังทำงานอยู่



6. ไบท์สำหรับอ่านเขียนข้อมูล

เขียน (WRITE) - กำหนดตำแหน่งของหน่วยความจำ ที่ DDDDDDDD และจะเพิ่มขึ้น1 ทุก ๆ ครั้งที่มีการเขียน

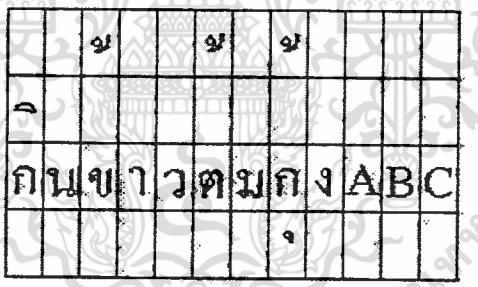
อ่าน (READ) - การทำงานเช่นเดียวกับการเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การแสดงตัวอักษรภาษาไทย

4.1 การแสดงตัวอักษรภาษาไทย 4 ระดับ

การแสดงผลภาษาไทยมีความยุ่งยากพอสมควร ภาษาไทยไม่สามารถพิมพ์เป็นแถวต่อเนื่องกันมีสระบางชนิดอยู่บนตัวอักษร บางชนิดอยู่ข้างล่างตัวอักษรแถมยังมีวรรณยุกต์หรือพวกไม้เอก ไม้โท..... อยู่เหนือขึ้นไปอีกหากลองดูดีดีอาจแยกเป็น 4 ระดับหรือ 4 บรรทัด ถ้าจะไล่ระดับจากข้างบนลงข้างล่างจะประกอบด้วยบรรทัดของไม้เอก-ไม้โท บรรทัดของสระอิ-สระอุ บรรทัดของตัวอักษร ก-ข-คและบรรทัดของ สระอุ-สระอูดังนั้นก่อนที่จะพิมพ์ภาษาไทยได้เราต้องแยกข้อมูล(string) ออกเป็น 4 แถวเสียก่อนจัดวางตำแหน่งให้ถูกต้อง ข้อมูลชื่อคนซึ่งยาว 30 ตัวอักษรอาจจัดให้หดสั้นเหลือแค่ 20 คอลัมน์แต่ปัญหาเกิดขึ้นที่ว่า ถ้าพิมพ์ด้วยระยะห่าง 6 บรรทัดต่อ 1 นิ้วเราจะได้ภาษาไทยที่รูปร่างสูงโย่ง ถึงแม้จะบีบลงมาเป็น 8 บรรทัดต่อ 1 นิ้วก็ยังอ่านไม่ค่อยสะดวก พวกไม้เอกไม้โทจะขึ้นไปลอยอยู่ไกลสุดขอบฟ้าเลย ดูตัวอย่างในรูปที่ 4.1



รูปที่ 4.1 แสดงภาษาไทยเมื่อพิมพ์ช่องละ 1 อักขระ

สาเหตุที่ทำให้ตัวอักษรดูสูงโย่งก็เพราะแต่ละกรอบ จะมีขนาดกว้าง 1 ต่อ 10 นิ้วและสูง 1/16นิ้วซึ่งสูงผิดปกติเมื่อจะต้องแยกเป็น 4 แถวอีกเราเลยได้ข้อความตัวอักษรที่สูงโย่ง ดังที่เห็นด้านบน

วิธีแก้ปัญหาในเรื่องนี้เราสามารถทำได้โดยการแบ่ง 1 ช่องที่เคยพิมพ์ตัว A,B,Cนั้นควรแยกออกเป็น 3 ส่วน ส่วนบนใช้พิมพ์สระอิ-สระอี ส่วนล่างใช้พิมพ์ตัวอักษร ก-ข-ค การแบ่งนี้จะเป็นสัดส่วนประมาณ1/3และ 2/3 ตามลำดับด้วยวิธีนี้จะทำให้ตัวอักษร ก-ข-ศ ลดความสูงลงมาเป็น 1/6 คูณ2/3=1/9นิ้ว ซึ่งเมื่อเทียบกับความกว้าง 1/10นิ้วแล้วทำให้ความสูงได้ระดับพอดี

นอกจากนี้ตำแหน่งที่ตั้งของฟอนต์ จะต้องจัดให้อยู่ด้านบนและล่างดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ตำแหน่งที่ตั้งของฟอนต์ จะต้องจัดให้อยู่ด้านบนและล่างดังนี้

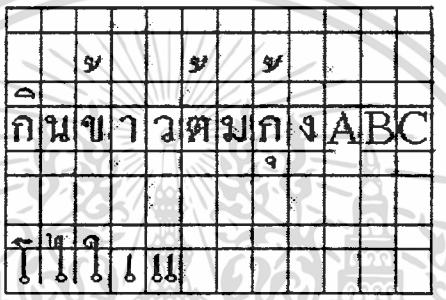
1. ไม้เอก-ไม้โท อยู่ด้านล่าง

2. สระอิ-สระอี อยู่ด้านบน

3.ก-ข-ค อยู่ด้านล่าง

4.สระอุ-สระอู อยู่ด้านบน

การจัดแบ่งหน้าของฟอนต์ออกเป็น 3 ส่วนยังช่วยทำให้เราแกะรูป สระไอ-สระโอ ให้มีขนาดสูงเต็มที่ด้วย ถ้าเราพิมพ์แยกเป็น 4 แถวแบบเดิมเราต้องตัดสระพวกนี้ออกเป็น 2 ส่วนเรียกว่า "สระเอต่อหัว"เวลาผลแสดงจะเชื่อมต่อกันเป็นแนวเดียวได้สนิท



รูปที่ 4.2 แสดงภาษาไทยแบ่งช่องพิมพ์ออกเป็น 2 ส่วนเมื่อลดความสูงลงแล้ว

อันว่ารูปแบบของตัวอักษรที่ปรากฏให้เห็นบนหน้าจอหรือบนกระดาษพิมพ์นั้น เขาเรียกว่าฟอนต์(FRONT)เป็นการนำเสนอภาษาไทยบนจอและเครื่องพิมพ์

4.2 บิทแม็พกับเวคเตอร์

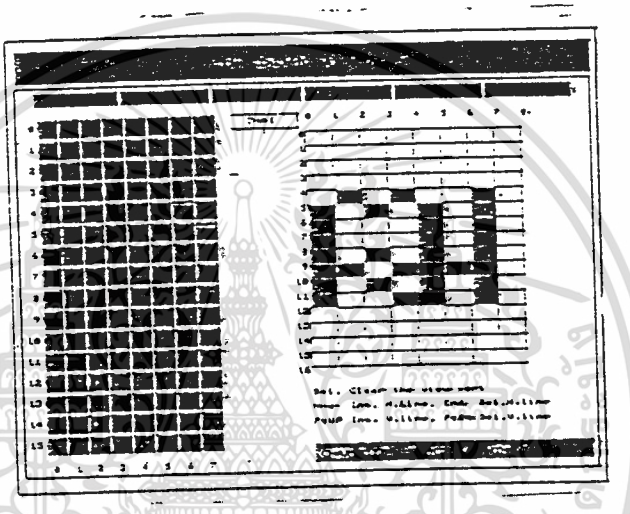
ฟอนต์ที่มีรูปแบบจุดประกอบกันในลักษณะของตารางสองมิติ โดยกำหนดจุดในตารางสองมิติให้เป็นขาวหรือดำ เพื่อต่อกันเป็นรูปแบบของตัวอักขระเรียกว่าฟอนต์แบบมิตแม็พ เพราะเป็นการเก็บรูปแบบของตัวอักขระเหล่านี้โดยการกำหนดให้ทุกจุดบนตารางเกี่ยวโยงกับหน่วยความจำ 1 บิต ฟอนต์แบบนี้ใช้ในการแสดงผลบนจอ ซีอาร์ที ที่เราเห็นโดยทั่วไป เช่นกัน รูปลักษณะของอักขระโรมันนั้น สามารถใช้ตาราง 8คูณ8 จุดแทนได้โดยคุณภาพของอักขระที่เห็นนับว่าอยู่ที่ยอมรับได้

ฟอนต์ที่มีลักษณะเป็นคำสั่งในการลากเส้น แบบที่ใช้สั่งเครื่องพล็อตเตอร์นั้นแตกต่างไปจากบิตแม็พโดยสิ้นเชิง การเก็บฟอนต์แบบนี้คือการเก็บคำสั่งในการลากเส้นของเครื่อง อาจจะเป็นการลากเส้นอย่างเดียวหรือมีคำสั่งอื่นๆมาผสมเช่นเส้นโค้ง วงกลมหรือครึ่งวงกลมหรือส่วนของ

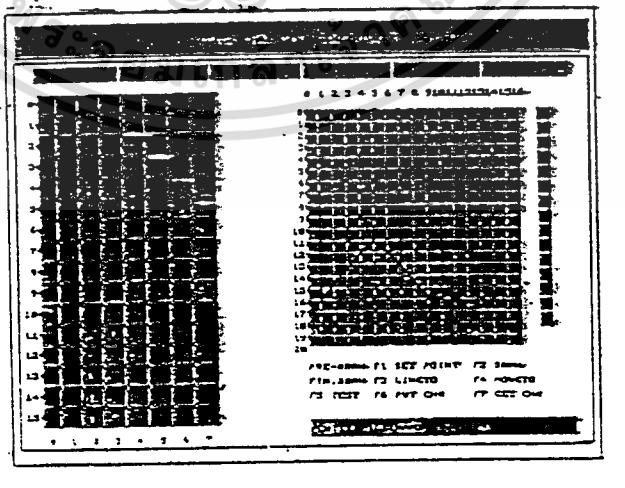
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นโค้งเพื่อทำให้รูปแบบตัวอักษรออกมาสวยงามเสมือนหนึ่งว่าเป็นการกำหนดทิศทางของการเขียนเอาไว้

การแสดงผลบนจอแทบทุกรายการเป็นการแสดงผลแบบบิทแม็พ เนื่องจากง่ายและแสดงผลได้รวดเร็วกว่าวิธีการใช้ฟอนต์แบบเวคเตอร์ โปรแกรมทางด้านกราฟฟิคหลายตัว ใช้เวคเตอร์ฟอนต์สำหรับการแสดงผล แม้ว่าการแสดงผลจะช้ากว่าการใช้ฟอนต์แบบบิทแม็พก็ตาม สาเหตุของการใช้เวลาในการเขียนบนจอมากกว่าบิทแม็พก็เนื่องจากว่า ซีพียู จะต้องเสียเวลาในการแปลงเวคเตอร์ฟอนต์เขียนออกบนจอ



รูปที่ 4.3 การสร้างฟอนต์แบบบิทแม็พขนาด 8 คูณ 16

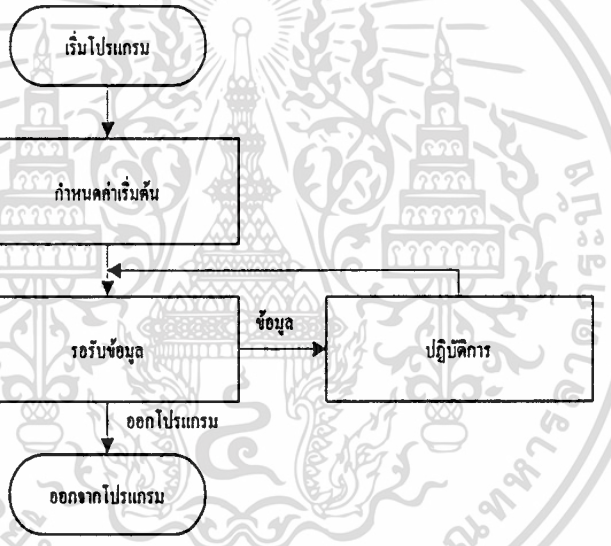


รูปที่ 4.4 การสร้างฟอนต์แบบเวคเตอร์ขนาด 8 คูณ 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
หลักการและการออกแบบโปรแกรมบนเครื่องไมโครคอมพิวเตอร์

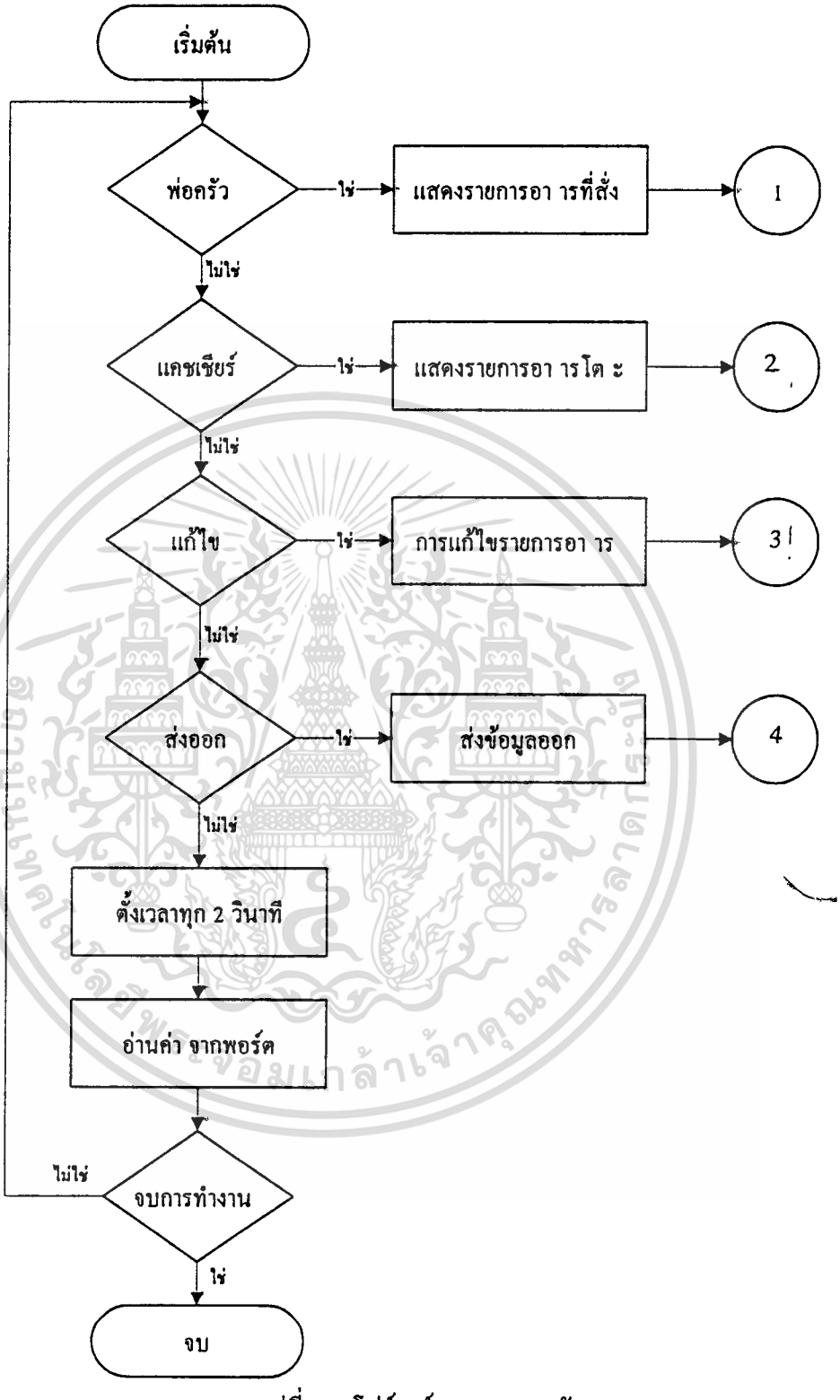
แนวทางในการเขียนโปรแกรมรับข้อมูลจากคอมพอร์ท(Communication Port)จะต้องทำการเขียนโปรแกรมแบบเรียวไทม์(Real Time)คือจะทำการไปอ่านข้อมูลมาจากพอร์ทถ้าไม่มีก็ไปทำงานอย่างอื่นต่อ ถ้าไปทำการอ่านจากพอร์ทแล้วเจอข้อมูลก็จะทำการอ่านข้อมูลแล้วจึงไปทำงานอย่างอื่นต่อเมื่อทำการเรียกโปรแกรมให้ทำงาน จะให้ทำการเลือกค่าต่างๆดังนี้ พอร์ทความเร็ว บิทข้อมูล บิทปิดท้ายและพาริตี้บิท เมื่อกดปุ่มตกลง ก็จะเข้าสู่เมนูหลักของโปรแกรม ดังโฟร์ชาร์ทรูปที่ 5.1



รูปที่ 5.1 โฟร์ชาร์ทการทำงาน

จากโปรแกรม(Window Function) จะคอยเป็นสวิทซ์เลือกข้อมูลในการรับข้อมูล คือการคลิกส์เม้าส์เลือกรายการโดยแสดงเป็นปุ่มต่างๆดังนี้ พ่อครัว แคชเชียร์ แก้ไข ส่งออกและจบการทำงาน ดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

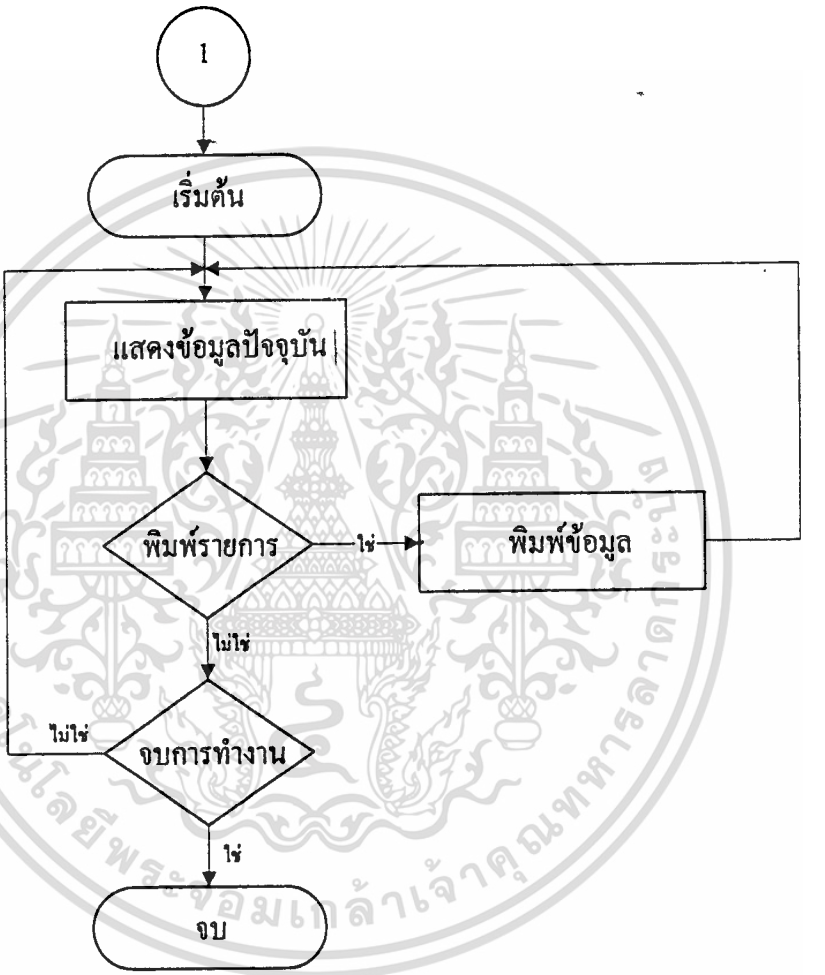


รูปที่ 5.2 โฟร์ชาร์ทแสดงเมนู ลัก

เมื่อคลิกส์ปุ่มเม้าส์ไปที่ปุ่มพ่อครัวจะแสดงเมนูปุ่มของรายการอา ารตามที่สั่งทั้ง มด โดย จะแสดงข้อมูลปัจจุบันที่ทำการอ่านข้อมูลมาจากพอร์ทไว้ที่ file project log โดยการทำงานเรียก

เอกสารนี้เป็นเอกสาร ไม่ว่ากรณีใดๆ

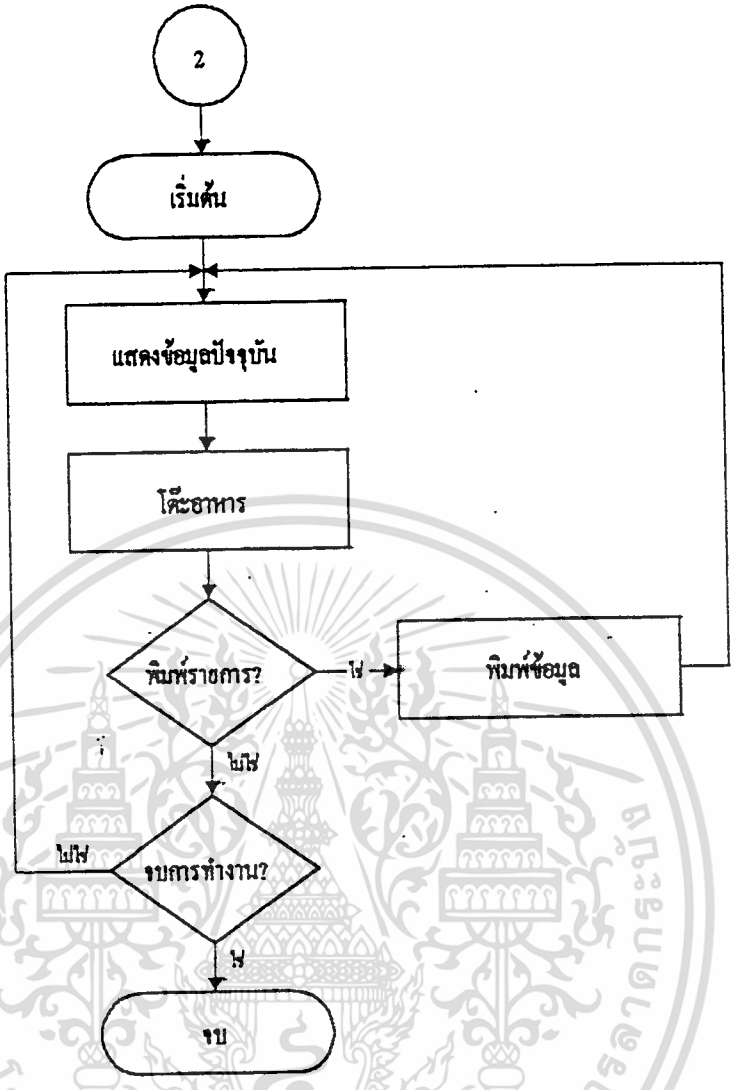
ข้อมูลจาก file project log มาไว้ที่หน่วยความจำส่วนหนึ่ง เมื่อทำการคลิกส์ปุ่มเมาส์ไปที่พิมพ์รายการ ข้อมูลในเมนูรายการอาหารตามที่สั่งจะถูกเคลียร์ ไปเก็บไว้ในหน่วยความจำอีกส่วนในเมนูแสดงรายการอาหารตามโต๊ะ แล้วข้อมูลของรายการทั้งหมดจะถูกพิมพ์ออกมาและเมื่อทำการคลิกส์ปุ่มเมาท์ไปที่จบการทำงาน ก็จะออกไปสู่เมนูหลัก



รูปที่ 5.3 โฟร์ชาร์ทแสดงเมนูพ่อครัว

เมื่อคลิกส์ปุ่มเมาส์ไปที่แคชเชียร์ก็จะทำการแสดงข้อมูลรายการอาหารตามโต๊ะ แต่ละโต๊ะพร้อมทั้งคิดเงินรวมของราคาอาหารทั้งหมดของแต่ละโต๊ะ เมื่อคลิกส์ปุ่มเมาส์ไปที่ปุ่มพิมพ์ใบเสร็จก็จะทำการพิมพ์รายการทั้งหมดของโต๊ะนั้นพร้อมคิดราคารวมและจะทำการเคลียร์ข้อมูลของโต๊ะนั้น เมื่อคลิกส์ปุ่มเมาท์ไปที่จบการทำงานก็จะออกไปสู่เมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

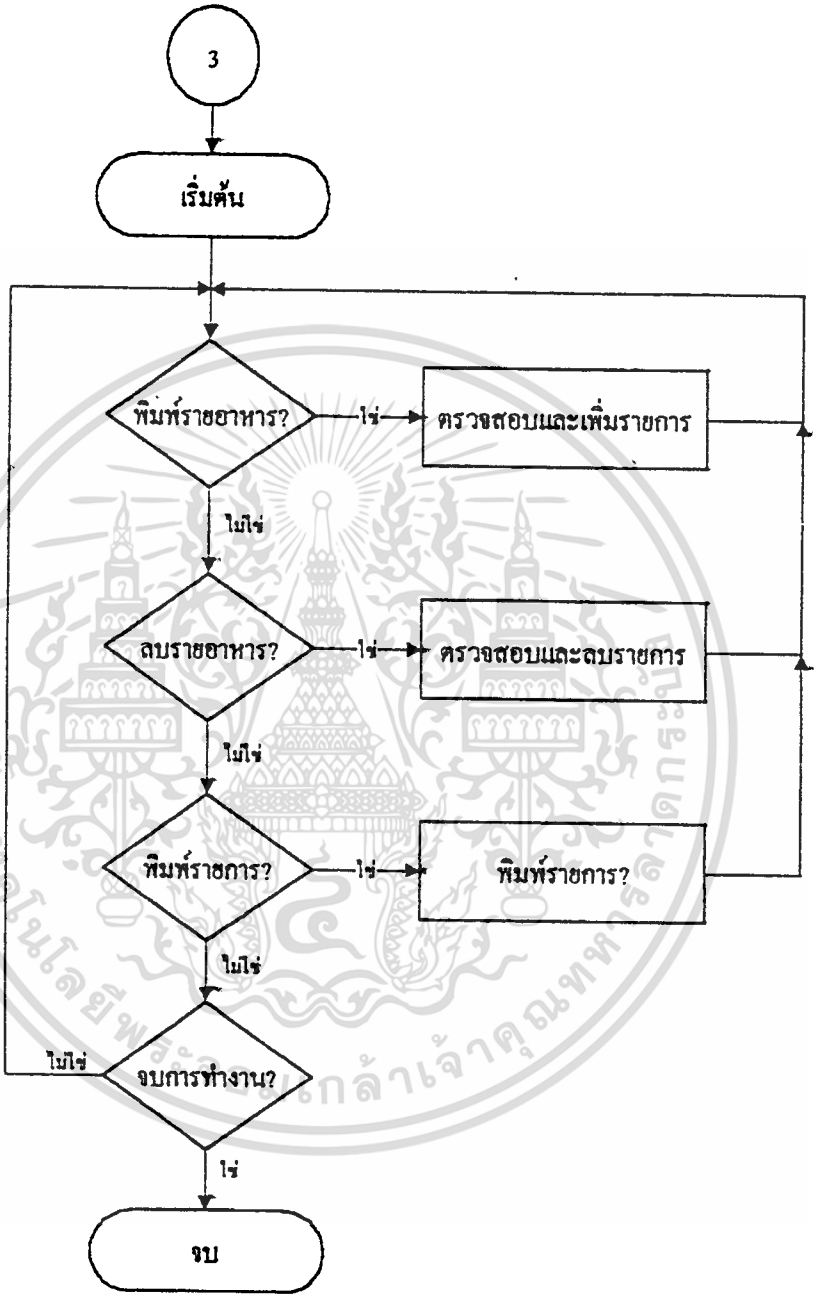


รูปที่ 5.4 โฟร์ชาร์ทแสดงเมนูแคชเชียร์

เมื่อคลิกส์ปุ่มเมาส์ไปที่แก้ไขก็จะแสดงเมนูแก้ไขรายการอาหาร จะแสดงรายการอาหารที่เป็นฐานข้อมูลทั้งหมดได้แก่ ประเภท รหัส ราคาและชื่อรายการอาหาร เมื่อคลิกส์เมาท์ไปที่ปุ่มเพิ่มรายการก็จะทำการให้ใส่ ประเภทอาหารที่เป็นตัวเลขแล้ววทำการตรวจสอบ ถ้าถูกต้องก็จะทำการเก็บค่าลงไป ถ้าเก็บค่าไม่ถูกต้องก็จะฟ้องบอกว่าให้ใส่ค่าใหม่ แล้ก็จะรับรหัสอาหาร ชื่อรายการอาหารและราคา

เมื่อคลิกส์ปุ่มเมาท์ไปที่พิมพ์รายการก็จะพิมพ์รายการอาหารทั้งหมด และเมื่อคลิกส์ปุ่มเมาท์ไปที่จบการทำงานก็จะออกมาสู่เมนูหลักข้อมูลรายการอาหารที่เป็นฐานข้อมูลจะถูกเก็บไว้ในไฟล์ชื่อ project dot เมื่อคลิกส์ปุ่มเมาท์ไปที่ลบรายการก็จะให้เลือกรายการอาหารที่ต้องการให้ลบแล้วทำการลบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



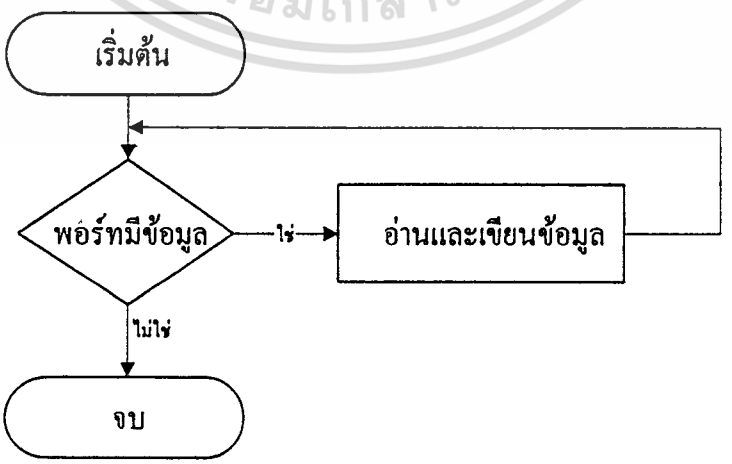
รูปที่ 5.5 โฟร์ชาร์ทแสดงเมนูแก้ไข

เมื่อคลิกส์เมาท์ไปที่ปุ่มส่งออก ข้อมูลของรายการอา ารที่เป็นฐานข้อมูลก็จะถูกส่งออกไปที่พอร์ทเพื่อที่จะทำการเก็บฐานข้อมูลลงในRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.6 โฟร์ชาร์ทแสดงการส่งข้อมูลออก

ส่วนโฟร์ชาร์ทแสดงการตั้งเวลา 2 วินาทีในการอ่านข้อมูลจาก พอร์ทถ้ามีข้อมูลก็จะทำการเขียนข้อมูลลงในไฟล์ project dat



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

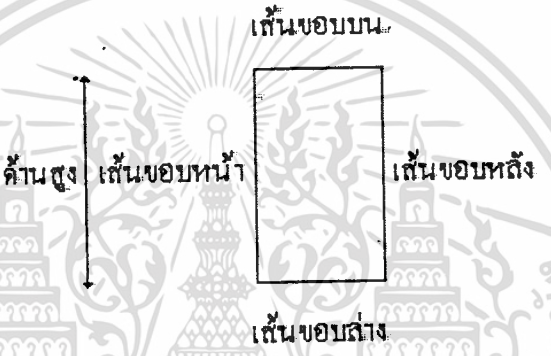
รูปที่ 5.7 โฟร์ชาร์ทการตั้งเวลา

บทที่ 6

หลักการและการออกแบบโปรแกรมบนบอร์ดCP-784C11

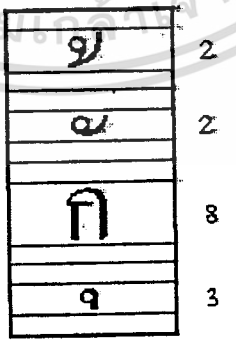
6.1 การแบ่งหน้าจอและการออกแบบตัวอักษร

หลักการทำงานของโปรแกรมจะมีการแสดงผล เป็นแบบบิตแม็พโดยกำหนดให้มีการแสดงผลทีละ column จากซ้ายไปขวาโปรแกรมจะทำการแบ่งจอ LCD ออกเป็น 4 แถว(ตามแนวนอน)16 column (ตามแนวตั้ง)โดย 1 column จะมีกรอบอยู่ดังรูป



รูปที่ 6.1 แสดงตำแหน่งจอ LCD

และเนื่องจากตัวอักษรไทย มีการวางสระ วรรณยุกต์เป็น 4 ระดับฉะนั้นจึงต้องมีการแบ่งกรอบออกเป็น 4 ระดับตามรูปที่ 8มีอัตราส่วนจากบนลงล่างดังนี้ คือ2,2,8,3 ของเส้น ดอท



รูปที่ 6.2 การแสดงตัวอักษรภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากต้องการใช้ประโยชน์จากจอ LCD ให้แสดงผลให้มากที่สุด ในขณะที่จอ LCD มีขนาดจำกัด

การออกแบบโปรแกรม

จากการพิจารณา Hardware ของจอ LCD กราฟฟิคเพื่อสะดวกในการควบคุมจึงแบ่ง Memmory ส่วนหนึ่งใน Ramให้เป็นVedio Ram โดยการแสดงผลทั้งหมดที่ปรากฏบนหน้าจอจะนำข้อมูลจาก Vedio Ram มาแสดงผลทั้งหมดจะมีการนำประโยค (SIRING)จากชุดคำสั่งนำไปเรียงไว้ใน Vedio Ramทุกครั้งก่อนการแสดงผล เพื่อสะดวกในการพัฒนาโปรแกรมการทำงานต่อไปในอนาคต

รูปที่ 6.3 การทำงานของแต่ละกระบวนการของจอ LCD

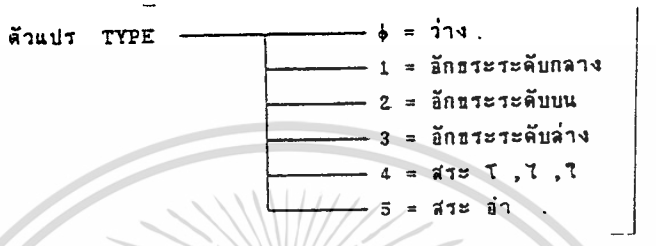
การทำงานของแต่ละ process จะอธิบายในตอนต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

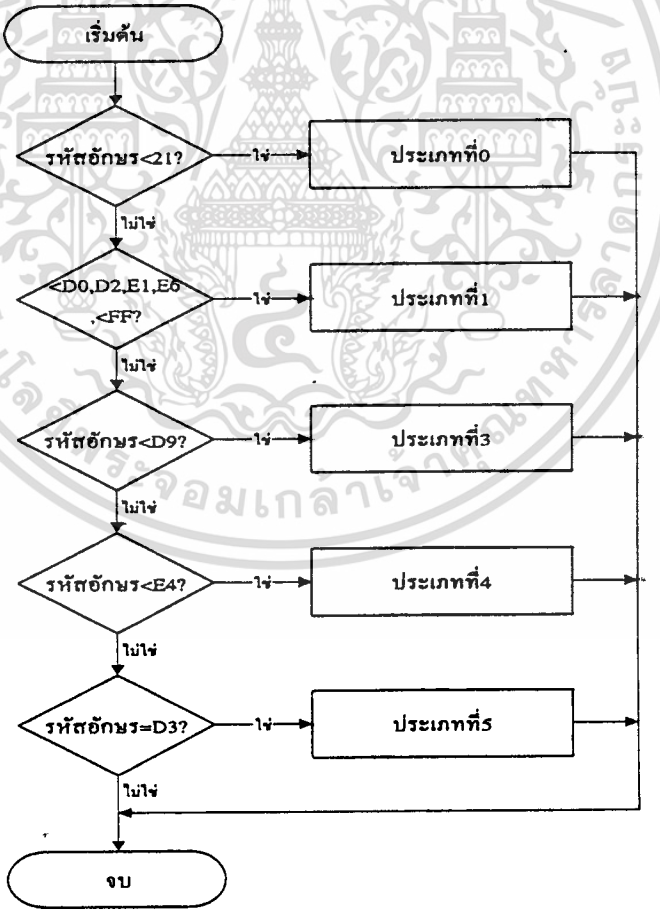
การแยกชนิดสระและวรรณยุกต์

การออกแบบโปรแกรม จัดรูปแบบภาษาไทยเป็นเรื่องที่ยุ่งยากมาก เนื่องจากมีหลายระดับและมีอักษรพิเศษค่อนข้างมากเช่น สระโอ,ไอ,อา ซึ่งสระเหล่านี้ไม่สามารถที่จะจัดลงตาราง 8คูณ8 dotได้ทำให้ต้องเพิ่มเงื่อนไขของโปรแกรมสำหรับอักขระพิเศษบางตัว

จากโปรแกรมจะมีการกำหนดตัวอักษรดังนี้

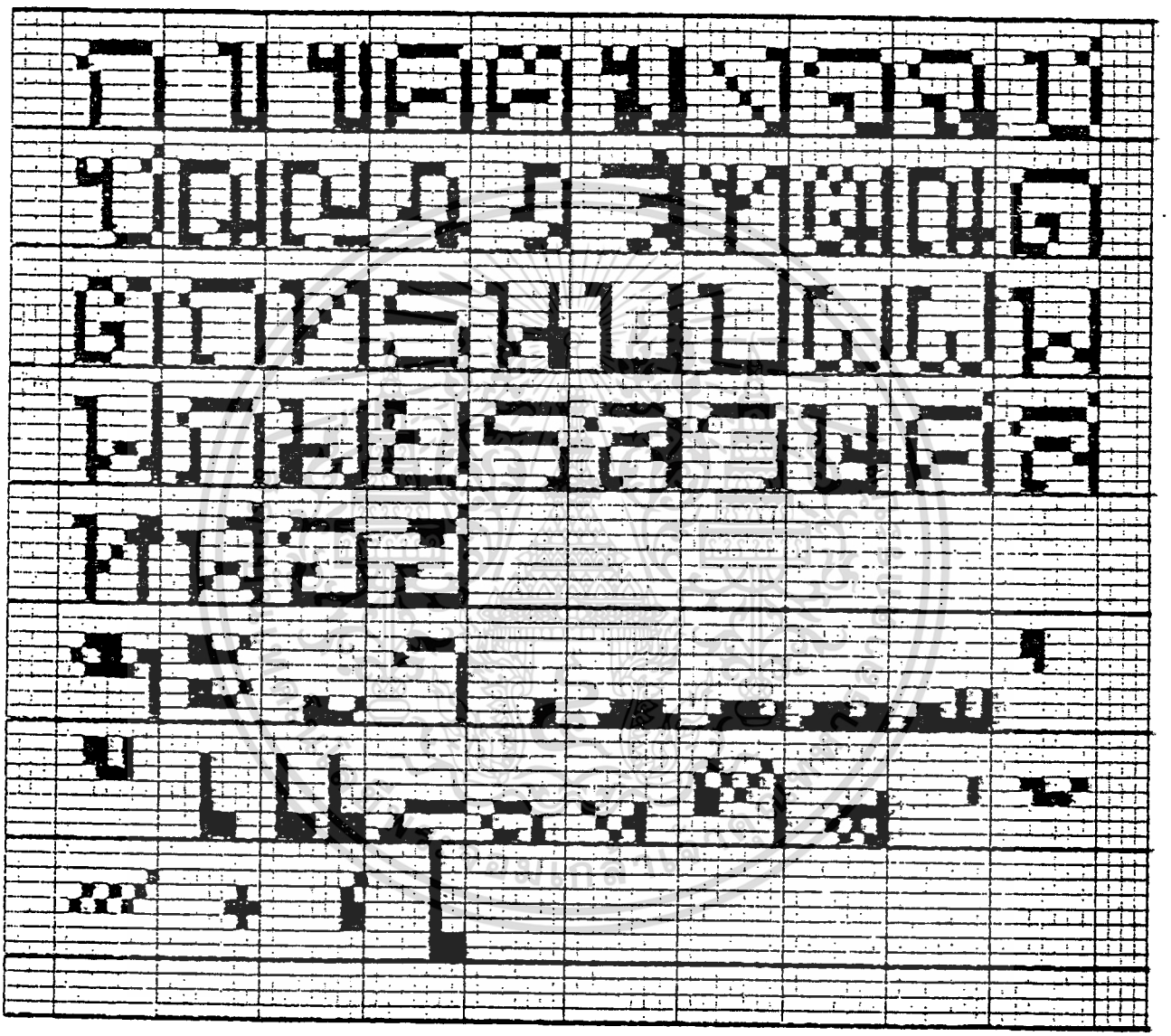


รูปที่ 6.4 แสดงประเภทอักขระต่างๆ



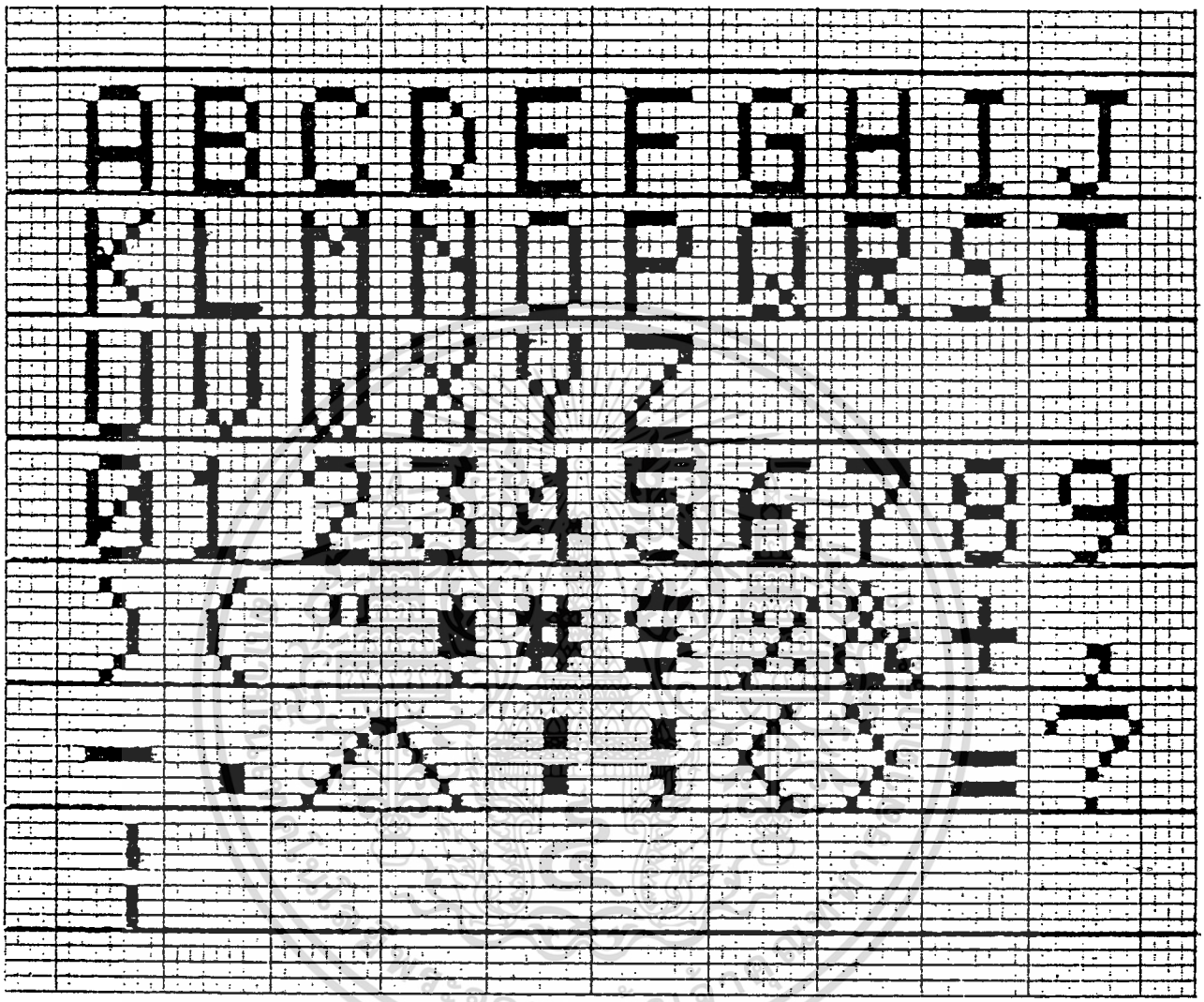
รูปที่ 6.5 โฟร์ชาร์ทการแยกประเภทอักขระต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 รูปแบบตัวอักษรภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



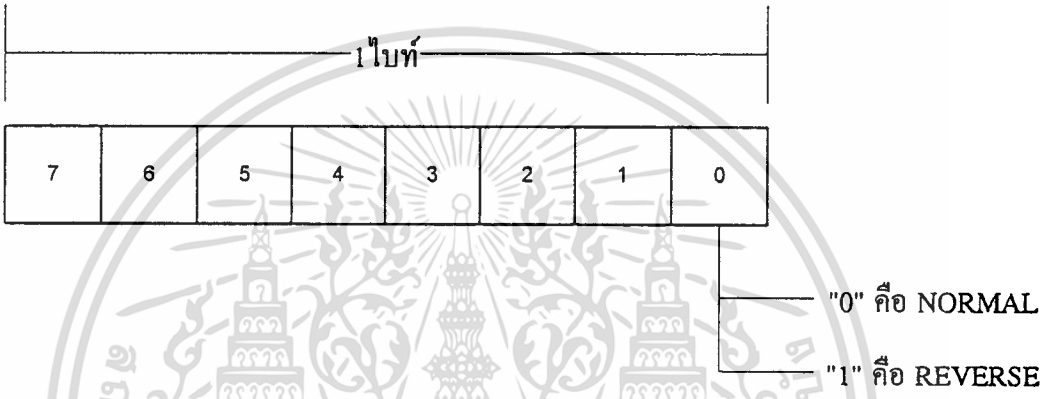
รูปที่ 6.7 รูปแบบตัวอักษรภาษาอังกฤษและตัวแปรต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแบ่งส่วนของหน่วยความจำสำหรับใช้ในการแสดงผล

จากทฤษฎีเบื้องต้นการแสดงผลภาษาไทยที่เป็นแบบ 4 ระดับถ้าเราสังเกตุให้ดีจะพบว่าการวางสระวรรณยุกต์ จะมีการใช้อักษรเพียงแค่ 3 ตัวทุกครั้งใน 1 column เราจะใช้หน่วยความจำ 4 byte โดยตัวอักษร 1 ตัว/byte 3ตัวรวมกับค่าแอททริบิว อีก 1 byte

แอตทริบิวไบท์จะใช้เพียงแค่หนึ่งบิทหลังเท่านั้น(ส่วนอื่นใช้ในการพัฒนาโปรแกรมต่อไปในอนาคต)โดยมีการกำหนดดังนี้



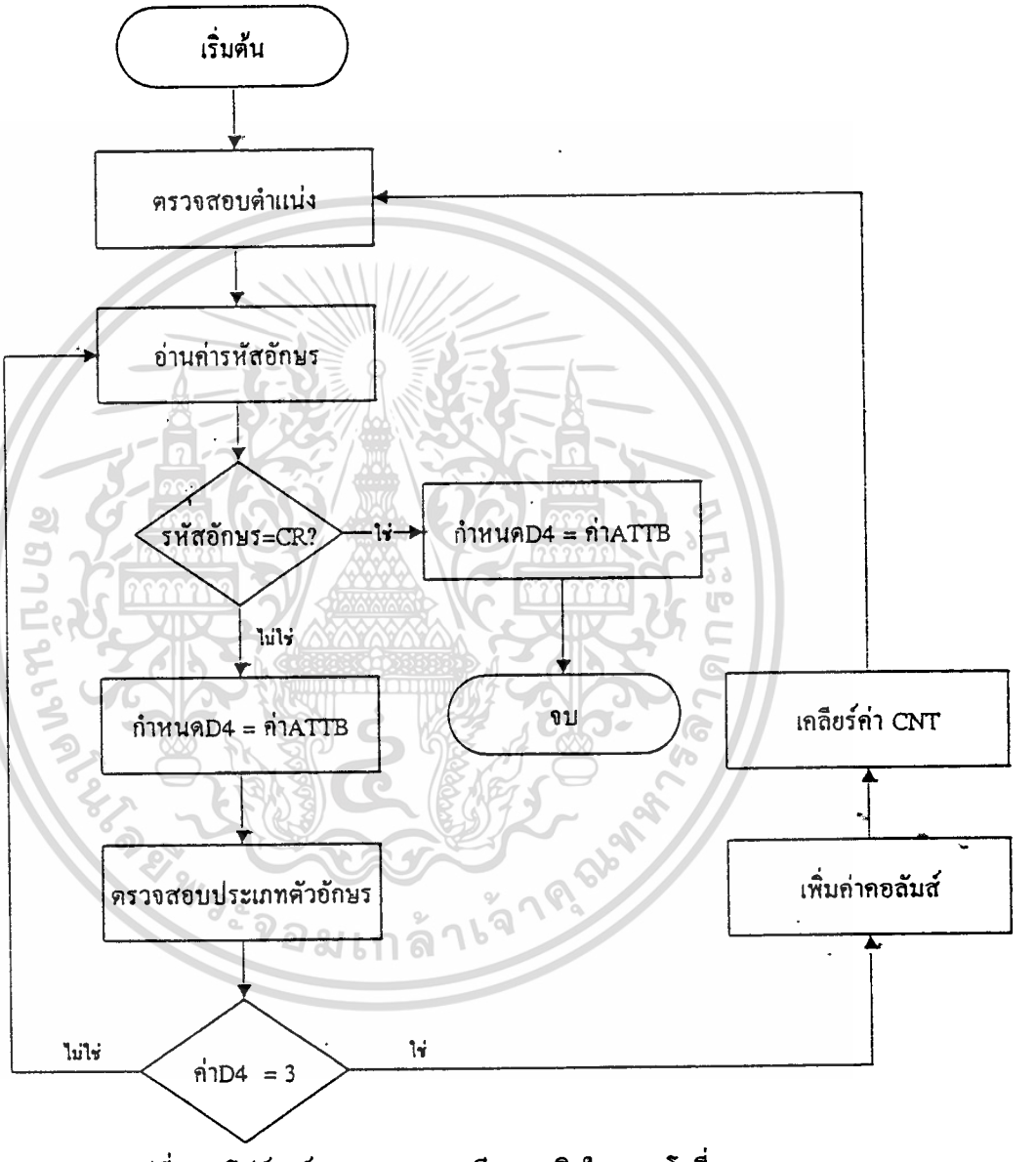
รูปที่ 6.8 การใช้งานไบท์กำหนดแอททริบิว

จากเรื่องการแบ่งจอในหัวข้อที่แล้วเราจะพบว่าเราแบ่งจอLCD ออกเป็น 4แถว 16 คอลัมส์ ทั้งหมด 4คูณ16 เท่ากับ 64 คอลัมส์ โดย 1คอลัมส์ใช้หน่วยความจำ 4 ไบท์เพราะฉะนั้นวีดีโอแรมจะต้องมีขนาดรวม 64คูณ4 เท่ากับ 256ไบท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำอักขระไปเรียงในหน่วยความจำสำหรับใช้ในการแสดงผล

การนำอักขระภาษาไทยไปเรียงไว้ในวีดีโอแรมแมมโมรี่โดยจะมีการตรวจสอบชนิดของอักขระทุกครั้งก่อนทำการเรียงอักษรลงในหน่วยความจำของแต่ละคอลัมภ์รูทีนมีการทำงานดังนี้



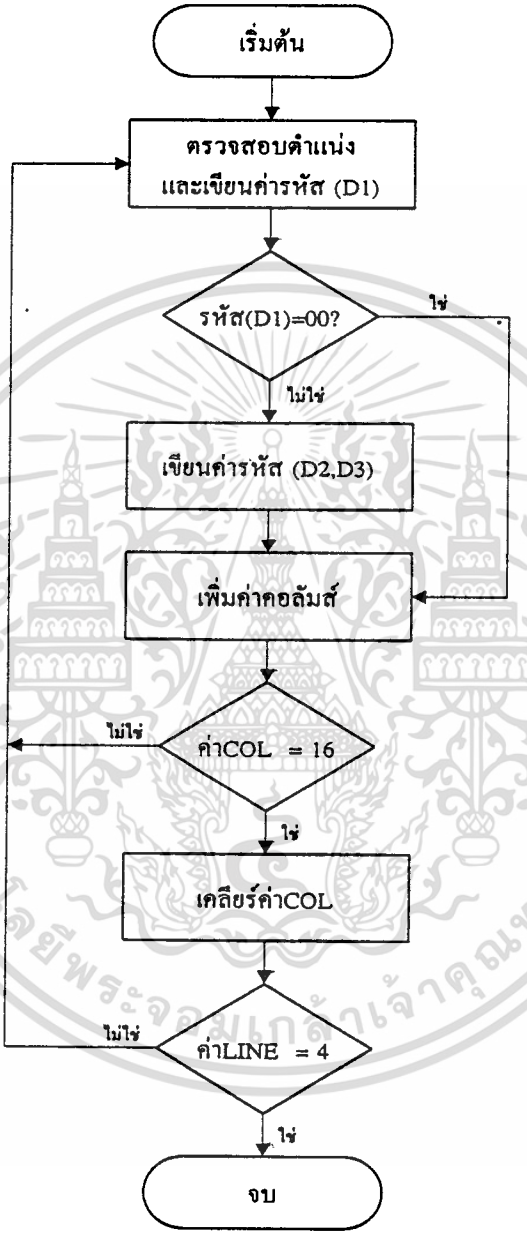
รูปที่ 6.9 โฟร์ชาร์ทกระบวนการเขียนสตริงในแมมโมรี่

ตรวจสอบตำแหน่งที่ต้องการในวีดีโอแรมอ่านรหัสอักษรตรวจสอบว่าใช่เครื่องหมายจบประโยคหรือไม่ตรวจสอบชนิดตัวอักษรแล้วใส่ค่าลงสู่หน่วยความจำตรวจสอบว่าเต็มคอลัมน์หรือยังถ้ายังไม่เต็มให้อ่านอักษรตัวใหม่ ถ้าเต็มแล้วให้เริ่มคอลัมน์เพื่อเปลี่ยนตำแหน่งของหน่วยความจำแล้วทำการอ่านข้อมูลตัวใหม่มารอการเรียงลงสู่หน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำข้อมูลในหน่วยความจำออกแสดงผล

โปรแกรมการทำงานในส่วนนี้ค่อนข้างตายตัว แต่จะต้องมีการเคลียร์หน้าจอก่อนทุกครั้งก่อนที่จะทำการเขียนลงบนจอใหม่มีการทำงานดังนี้



รูปที่ 6.10 โฟร์ชาร์ทนำค่าหน่วยความจำออกแสดงผล

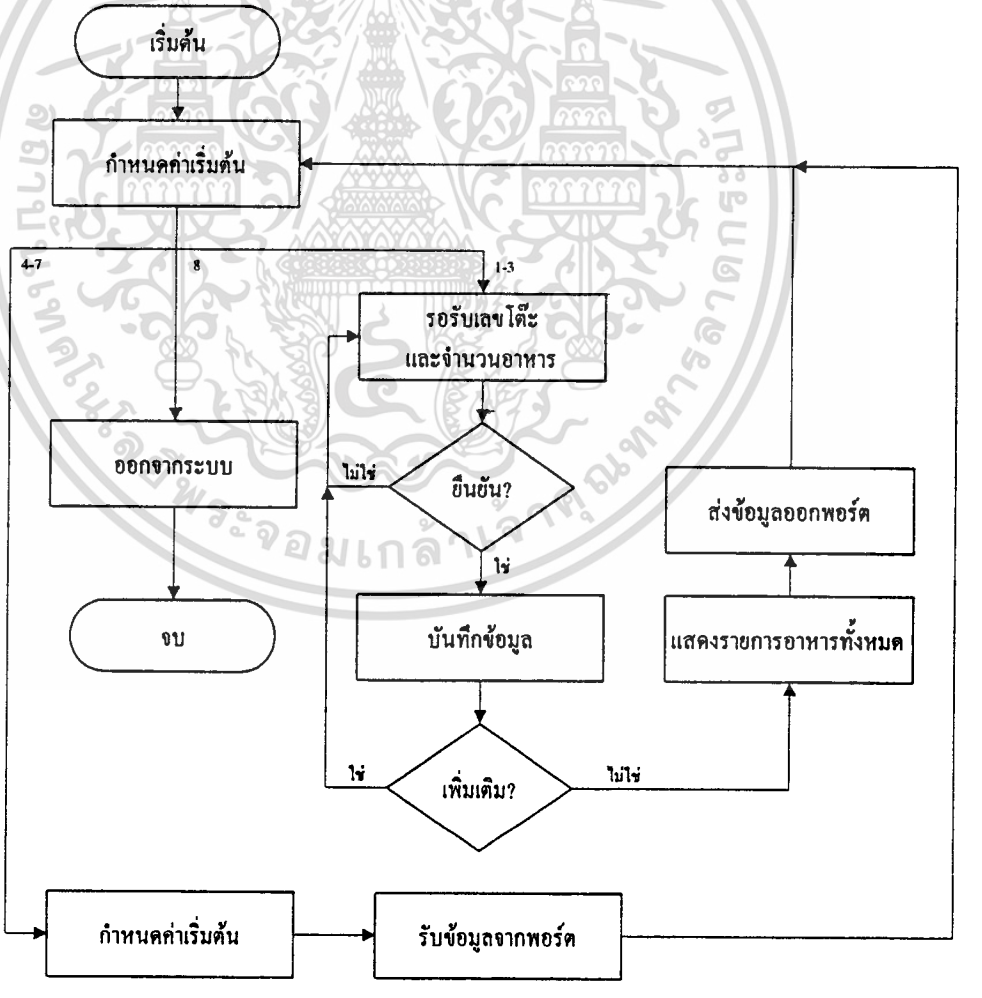
ข้อมูลในหนึ่งคอลัมน์จะมีอยู่ 4 ไบท์ดังรูปเริ่มต้นโปรแกรมจะทำการตรวจสอบตำแหน่งที่ส่งมาให้แล้วอ่านค่าแอทริบิวทำการแสดงอักขระตัวแรกออกจอแสดงผล ตรวจดูว่าเป็นโค๊ดว่างหรือไม่ ถ้าใช่ก็ทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2 การใช้งานและการออกแบบโปรแกรมจัดระบบ

เนื่องจากโครงการเป็นอุปกรณ์ใช้งานอำนวยความสะดวกภายในร้านอาหารซึ่งผู้ใช้งานไม่จำเป็นต้องมีพื้นฐานเกี่ยวกับระบบอิเล็กทรอนิกส์เลยจะต้องสามารถใช้งานได้ เพราะฉะนั้นระบบควรมีการใช้งานที่เข้าใจง่ายและสะดวก เพื่อเหมาะกับผู้ใช้งานในทุกระดับ

โดยที่ตัวเครื่องจะมีโปรแกรมการทำงานสั่งงานที่เมนูหลัก ที่มีลักษณะแถบสีเลือก(score bar) โดยผู้ใช้งานจะเลือกแถบสีโดยการกดคีย์ลูกศร(arrow key)ขึ้น-ลง เพื่อเลือกการทำงานตามที่เครื่องกำหนดไว้ เข้าทำงานที่เมนูย่อย(sub-manu)ตามลำดับ โดยแต่ละขั้นตอนจะมีข้อความ (message)กำกับ การใช้งานในทุกๆเมนูย่อย และมีการให้ยืนยัน (Confirm) ในขั้นตอนที่สำคัญ โดยมีลำดับการทำงานตาม โฟร์ชาร์ทดังนี้

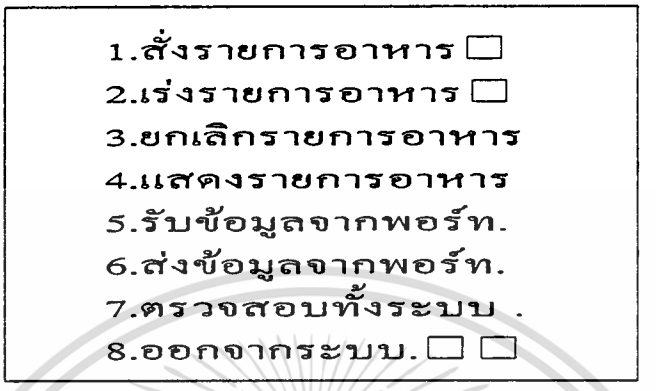


รูปที่ 6.11 โฟร์ชาร์ทการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรมจัดการ

เริ่มต้น เลือกการทำงานจากเมนูหลักโดยมีตัวเลือกที่กำหนดไว้ดังนี้



รูปที่ 6.12 ตัวเลือกเมนูหลัก

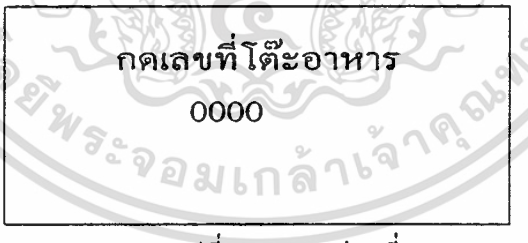
ตัวเลือกที่ 1-3 จะมีการทำงานเหมือนกันแต่จะแตกต่างกันที่รหัสของคำสั่งที่ใช้ส่ง

รหัส 00 สั่งอาหาร

01 เร่งอาหาร

02 ยกเลิกอาหาร

เมื่อกำหนดเลือกที่จอ แอลซีดี จะปรากฏ เมนูย่อยที่1 สำหรับรับค่าเลขที่โต๊ะอาหารตามรูปที่ 2 ดังนี้



รูปที่ 6.13 เมนูย่อยที่ 1

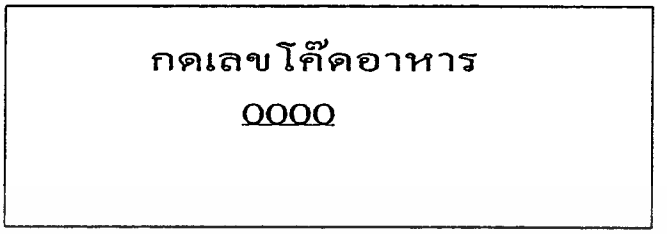
โดยเลขที่โต๊ะอาหารในขั้นต้นจะถูกจำกัดค่าไว้ที่ 0-255 โต๊ะกดคีย์ตั้งแต่ 1-6 การป้อนค่าจะมีลักษณะเลขจากด้านหลังไปที่ด้านหน้าฉะนั้นจะไม่มีการลบข้อมูล เพียงแต่กดค่าตัวเลขที่ปรากฏบนจอแอลซีดี มีค่าถูกต้องก็เพียงพอแล้วทำการกดคีย์ enter เป็นการยืนยันเช่น

ต้องการใส่ค่าโต๊ะที่ 51 ดังนี้

เริ่มต้น	0000
กด5	0005
กด1	0051
กดยืนยัน(enter)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อยืนยันแล้วจะปรากฏเมนูย่อยที่ 2 สำหรับใส่รหัสอาหาร มีขนาด 4 ไบต์คือ 0000-FFFF ในเลขฐานสิบหกหรือ 0-65535 รายการในเลขฐานสิบ เมนูย่อยที่ 2 มีลักษณะตามรูป ใส่ค่าเป็นรหัส 4 ตัวมีการใส่เหมือนกับเมนูที่ 1



รูปที่ 6.14 เมนูย่อยที่ 2

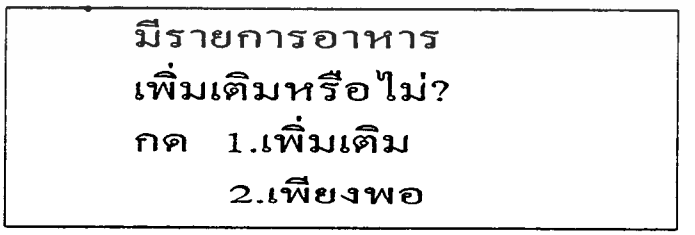
เมื่อกดยืนยันแล้วเครื่องจะนำรหัสอาหารไปค้นหาในฐานข้อมูลเพื่อนำชื่ออาหารมาแสดงที่เมนูย่อยที่ 3 เป็นการใส่จำนวนหน่วยของอาหารจะปรากฏตัวเลือกขึ้นตามรูป

รูปที่ 6.15 เมนูย่อยที่ 3

โดยถ้ากดคีย์ 1จะเป็นการยืนรายการและจำนวนอาหารที่ปรากฏในจอแอลซีดี

คีย์ 2จะเป็นการยกเลิกรายการและจำนวนอาหารที่ปรากฏในจอแอลซีดี

หลังจากนั้นจะปรากฏเมนูย่อยที่ 4 ดังรูป ตามความต้องการสั่งรายการอาหารเพิ่มเติมหรือไม่



รูปที่ 6.16 เมนูย่อยที่ 4

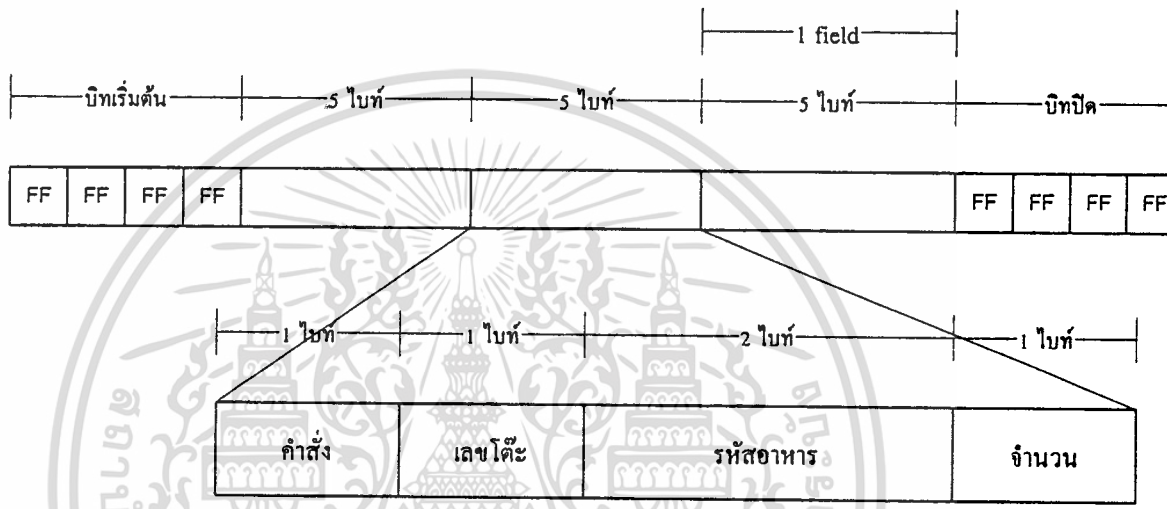
กด 1 จะเพิ่มเติมรายการอาหารโดยกลับไปทำที่เมนูย่อยที่ 2 หลังเริ่มทำลำดับใหม่ตั้งแต่ต้น

กด 2 ไม่ต้องการสั่งเพิ่มเครื่องจะทำการแสดงรายชื่ออาหารทั้งหมดออกมา แล้วทำการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 รูปแบบเฟรมข้อมูลของระบบ

ในการส่งข้อมูลเป็นกลุ่มอักขระที่แทนด้วยรหัสไบนารี่(binary coded character)โดยระบบของโครงงานนี้จะมีการกำหนดรูปแบบของข้อมูลเองตามรูปที่ xx.โดยจะมีการส่งข้อมูลแบบพื้นฐานรับหรือส่งทิศทางเดียว(Simplex)



รูปที่ 6.17 ส่วนประกอบของเฟรมข้อมูล

รหัสคำสั่ง ประกอบด้วยข้อมูลขนาด 1 ไบต์ (00-02)

รหัส 00 สั่งอาหาร

01 เร่งอาหาร

02 ยกเลิกอาหาร

เลขที่โต๊ะอาหาร ข้อมูลมีขนาด 1 ไบต์(00-FF)

เลขรหัสอาหาร ข้อมูลมีขนาด 2 ไบต์ (0000-FFFF)

โดยแบ่งเป็นไบต์แรกเป็นประเภทของอาหาร ไบต์สองเป็นชื่ออาหาร

รหัสบอกจำนวนอาหาร ข้อมูลมีขนาด 1 ไบต์ (00-FF)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

ในส่วนของฮาร์ดแวร์ทดสอบการมอดดูเลทและการดีมอดดูเลท

ขั้นตอนที่ 1 FSKโดยต่อวงจรที่ใช้ IC XR 2206 แล้วป้อนไฟเข้าพร้อมจับสัญญาณเอาท์พุทที่ขา 2 การส่งข้อมูลมี 2 สภาวะคือ 0 และ 1 ดังนั้นก่อนจะทดสอบต้องกำหนดสภาวะก่อนว่า "0" หรือ "1" ถ้าต่ออินพุทขา 9 ลงกราวด์เสมือนลอจิก "0" แล้วนำเครื่องวัดความถี่มาจับแล้วค่อยๆปรับ VR ที่ต่ออยู่ที่ 8 ค่อยๆ ปรับให้ได้มาตรฐานที่อัตราการส่งที่ 300 บิทต่อวินาที จะต้องได้เท่ากับ 2100 Hz เป็นรูปสัญญาณซายด์ การกำหนดสัญญาณ เอาท์พุท สามารถปรับได้ที่ตัวต้านทานที่ขา 13 และ 14 ปรับให้ได้สัญญาณซายด์สมบูรณ์ที่สุด

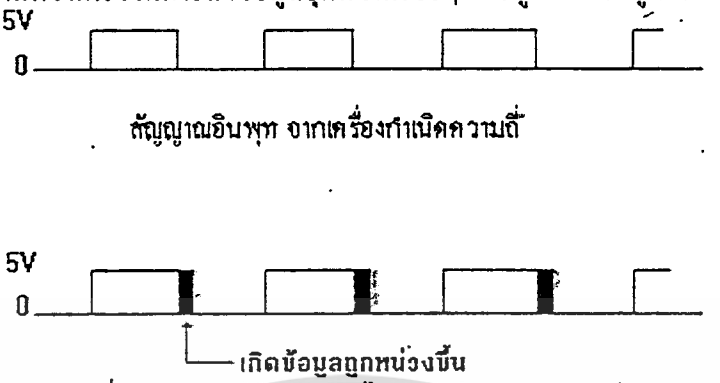
ขั้นตอนที่ 2 นำวงจรดีมอดูเลทที่ต่อเสร็จมาต่อกับภาคมอดดูเลทโดยในตอนแรกต่อกันแบบตรง แล้วกำหนดสัญญาณบิทอินพุทเป็น 0 แล้วค่อยๆ ปรับค่า RX จนได้สัญญาณที่ เอาท์พุทที่ขา 7 ของ ICXR 2211 ต้องได้ออกมาเป็นลอจิก "0" คือมีค่าต่ำกว่า 2 โวลท์ ผลจากการทดสอบทั้งบิท 0 และ บิท 1 เป็นไปตามอย่างทฤษฎี เมื่อลองใช้เครื่องกำเนิดความถี่ผลิตสัญญาณพัลส์ที่พัลส์สมมาตรที่ 300 ลูกต่อวินาที แล้วทดสอบเพิ่มความเร็วขึ้นเรื่อยๆ แล้วตรวจสอบเอาท์พุทของ ไอซี 2211ว่าสามารถจับสัญญาณได้หรือไม่ดูตามตารางผลดังนี้

ลำดับ	ความเร็วในการส่ง (บิท/วินาที)	ขนาดสัญญาณที่ดีมอด (Vpeak)
1	100 bps	5 Vp
2	200 bps	5 Vp
3	300 bps	4.9 Vp
4	400 bps	4.9 Vp
5	800 bps	4.8 Vp
6	1600 bps	ดีมอดไม่ได้ จับสัญญาณไม่ได้

ตารางที่ 7.1 ตารางแสดงผลการทดลอง

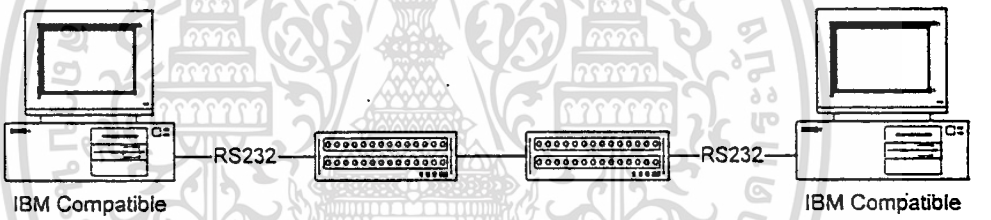
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพิ่มความเร็วในการส่งข้อมูลชุดนี้ขึ้นเรื่อยๆ ข้อมูลที่ดีมอดดูเลทจะเกิดปัญหาที่ดีมอด



รูปที่ 7.1 แสดงรูปข้อมูลที่ถูกหน่วงเวลาจากการป้อนสัญญาณที่อินพุท

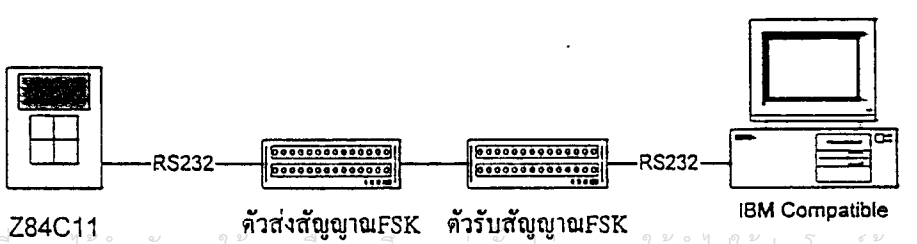
ขั้นตอนที่ 3 เป็นการทดสอบฮาร์ดแวร์สำเร็จรูปโดยใช้คอมพิวเตอร์ 2 เครื่อง ต่อเข้าฮาร์ดแวร์ของแต่ละข้างแล้วทดสอบการส่งข้อมูลผ่าน RS 232 โดยใช้โปรแกรม Procomm Plus ทดสอบดังรูป



รูปที่ 7.2 แสดงการทดสอบขั้นที่ 3

จากรูปมีสาย 3 เส้นที่เชื่อมต่อระหว่างวงจรทดสอบ 1 กับวงจรทดสอบ 2 คือ RX, TX, GND จากผลการทดสอบโดยใช้โปรแกรมProcomm Plus โดยตั้งค่าความเร็วให้เท่ากันและกำหนดข้อมูลบิท 8 บิท แล้วส่งตัวอักษรตัว A ผ่านวงจรทดสอบ 1 และดูที่จอ PC ที่ 2 ว่ารับตัว A หรือไม่ ปรากฏว่ารับได้สมบูรณ์

ขั้นตอนที่ 4 เป็นการเอาบอร์ด Z84C11 ต่อเข้าที่ตัวส่งสัญญาณ S และนำPCต่อเข้ากับ ตัวรับสัญญาณ S แล้วทำการส่งรหัสออกจากบอร์ด Z84C11แล้วดูที่จอ PC ดูรูปการต่อแบบทดสอบ



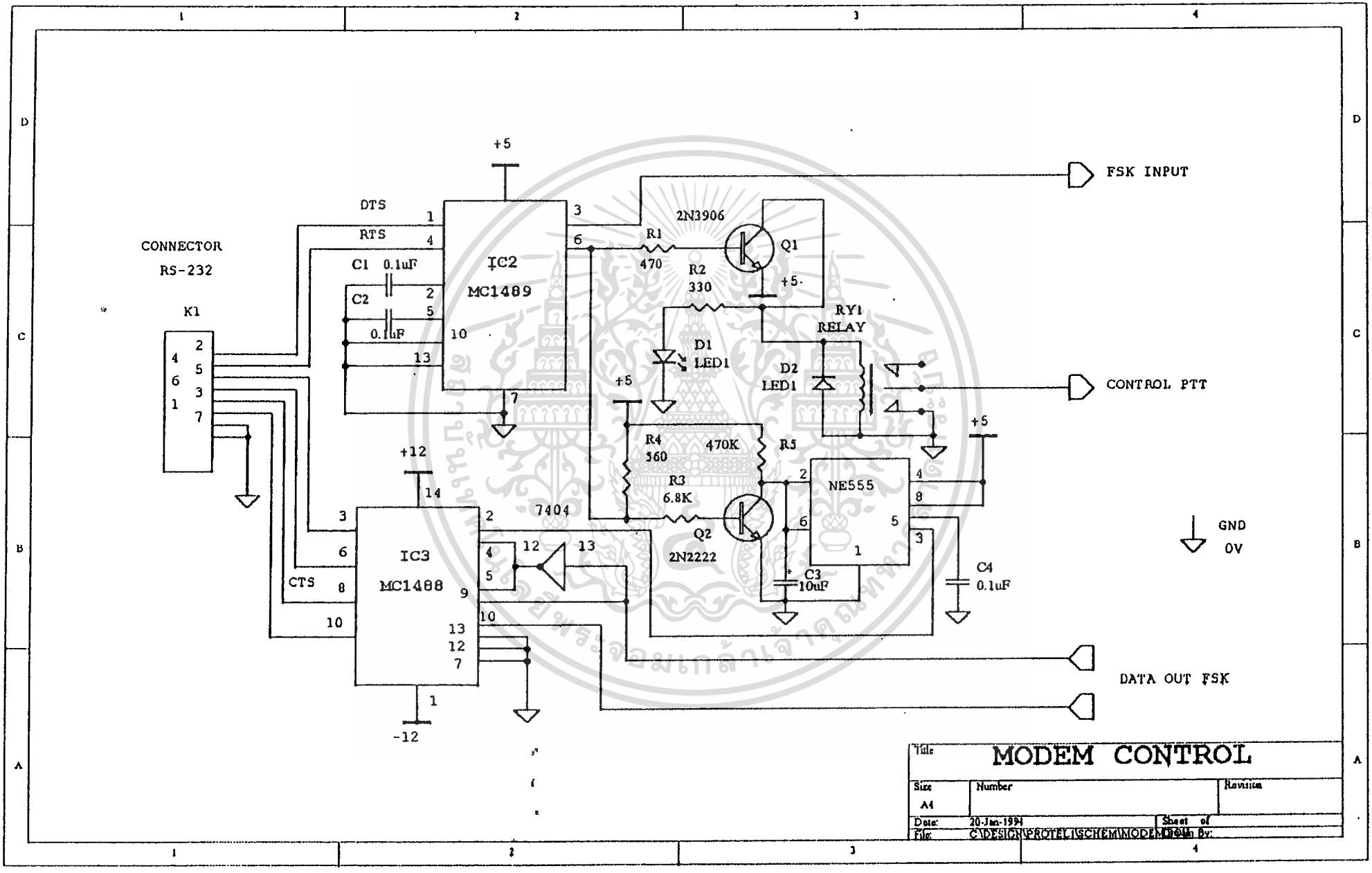
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.3 แสดงการทดสอบขั้นที่ 4

ผลการทดสอบโดยกการกด 001 ซึ่ง 001 เป็นข้าวผัดจะขึ้นมาที่หน้าจอ LCD จากนั้นส่งข้อมูลออกไปมาเป็นโค๊ด 001 ที่จอ PC จะต้องขึ้น 001 แล้วข้อมูล 001 ที่ออกมาแปลว่า เป็นอาหารชนิด ผลจากการทดสอบสามารถทำการรับส่งข้อมูลได้ แต่ข้อมูลที่จะรับส่งต้องมีอยู่ในฐานข้อมูลจึงจะสามารถที่จะแปลเป็นภาษาไทยแสดงออกมาที่หน้าจอได้

แล้วจากนั้นต่อผ่านเครื่องส่งวิทยุแทนการต่อตรง ๆ แล้วทำการส่งออกอากาศแล้วลองทดสอบที่ระยะ 10 เมตร ปรากฏว่าสามารถส่งและรับข้อมูลได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


CONNECTOR
RS-232
K1
DTS
RTS
CTS
IC2
MC1489
C1 0.1uF
C2
0.1uF
IC3
MC1488
7404
+5
+12
-12
R1
470
R2
330
2N3906
Q1
D1
LED1
D2
LED1
RY1
RELAY
R4
560
470K
R5
R3
6.8K
Q2
2N2222
NE555
C3
10uF
C4
0.1uF
FSK INPUT
CONTROL PTT
GND
0V
DATA OUT FSK
MODEM CONTROL
Size
A4
Number
Revision
Date: 20-Jan-1994
Sheet of
File: C:\DESIGN\PROTEL\SCHEM\MODEM.SCH
Drawn By:

บทที่ 8
บทสรุปและวิจารณ์

จากโครงงานที่ทำขึ้นปรากฏว่าสามารถรับส่งข้อมูลได้จริงแต่ยังมีปัญหาที่เกี่ยวกับสัญญาณรบกวนจากคอมพิวเตอร์ซึ่งก็เข้าไปรบกวนในระบบวิทยุสื่อสารทำให้การรับส่งอาจมีผิดพลาดบ้าง อีกปัญหาใหญ่ คือ การควบคุมการรับส่งของเครื่องวิทยุผ่านบอร์ดคอนโทรเลอร์ ซึ่งก็มีผลเสียก็คือ เปลืองพลังงานแบตเตอรี่ทำให้หมดเร็ว ดังนั้นแบตเตอรี่จะต้องมีขนาดใหญ่เพื่อจะจ่ายพลังงานได้นาน เพราะการสื่อสารที่ดีคือต้องรับข้อมูลได้ถูกต้องและสิ้นเปลืองพลังงานน้อยที่สุด อีกอย่างหนึ่งคือโปรแกรมของ Z 84 C 11 ที่เขียนขึ้นพอใช้งานจริงต้องมีคนเขียนโปรแกรมคอยบอกว่า ขั้นตอนต่อไปทำอะไรจึงต้องแก้ไขอีก ปัญหาที่พบบ่อยๆในโครงงานนี้คือ การเกิดการหน่วงเวลาของสัญญาณที่ดีมอดทำให้ช่วงกว้างของพัลส์ผิดไปจึงไม่สามารถที่จะส่งด้วยอัตราเร็วที่สูงขึ้นได้

เนื่องจากโครงงานนี้เป็นเครื่องต้นแบบเพื่อทดสอบ แนวคิดที่จะอำนวยความสะดวกในร้านอาหารใหญ่ๆ จึงอาจไม่สมบูรณ์แบบและยังมีสิ่งที่ต้องพัฒนาอีกหลายจุด เพื่อสามารถใช้งานได้อย่างมีประสิทธิภาพคือ

1. จะต้องดัดแปลงให้สามารถส่งข้อมูลแบบโต้ตอบไปกลับได้
2. สามารถใช้งานได้ที่ความเร็วของข้อมูลสูง
3. ประหยัดพลังงาน
4. มีวิธีการตรวจสอบข้อมูลที่ผิดพลาด
5. การลดสัญญาณรบกวน
6. การวางโปรแกรมยังไม่ดีพอ
7. สามารถส่งข้อมูลได้ในระยะไกล
8. ขนาดเล็กเหมาะสมกับการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include "project.h"
int PASCAL WinMain(HANDLE hInstance, HANDLE hPrevInstance, LPSTR lpszCmdLine,
int nCmdShow)
{
  MSG msg;
  if(!hPrevInstance){
    if(!InitFirstInstance(hInstance, hPrevInstance, nCmdShow))
      return NULL;
  }else{
    SetActiveWindow(FindWindow(szApplName, "เครื่องต้นแบบการส่งข้อมูลระยะไกล"));
    return NULL;
  }
  hInst = hInstance;
  hwMain = CreateWindow(szApplName,              //Window class name
                        "เครื่องต้นแบบการส่งข้อมูลระยะไกล",  //Title
                       WS_OVERLAPPED|WS_SYSMENU|WS_MINIMIZEBOX,      //
Window style
                        0,          //X position
                        0,          //Y position
                        MaxX = GetSystemMetrics(SM_CXSCREEN), //X size   525
                        MaxY = GetSystemMetrics(SM_CYSCREEN), //Y size   130
                       NULL,                  //Parent window
                        LoadMenu(hInstance, "MENU_ENG"), //Menu
                       hInstance,             //Application instance
                        NULL);                 //Creation parameter
  if(!hwMain)
    return NULL;
  SendMessage(hwMain, WM_COMMAND, FILE_COMM_SET, 0L);
  if(InitCommPort() < 0){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MessageBox(hwMain, "ไม่สามารถควบคุมการทำงานของพอร์ตอนุกรม 2 ได้!", "พอร์ตอนุ
กรม", MB_ICONSTOP);
    PostQuitMessage(0);
   }
  if(!SetTimer(hwMain, 1, 2000, NULL)){
    MessageBox(hwMain, "ไม่สามารถติดตั้งวงจรเวลาได้!", "วงจรเวลา", MB_ICONSTOP);
    CloseComm(CommID);          //Close communication port before exit
    PostQuitMessage(0);
   }
  if(InitDataFile() < 0){
    MessageBox(hwMain, "ไม่สามารถเปิดแฟ้มข้อมูลได้!", "แฟ้มข้อมูล", MB_ICONSTOP);
    CloseComm(CommID);          //Close communication port before exit
    KillTimer(hwMain, 1);       //Kill timer before exit
    PostQuitMessage(0);
   }
  ShowWindow(hwMain, nCmdShow);
  UpdateWindow(hwMain);
  while(GetMessage((LPMSG) &msg, NULL, 0, 0)){
    TranslateMessage(&msg);
    DispatchMessage(&msg);
   }
  return(msg.wParam);
}
int InitDataFile()
{
  char buffer[100];
  int  nread,
       nor,
       i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    DataFd = open(DATAFILE, O_CREAT| O_RDWR| O_BINARY);
    if(DataFd < 0)
      return -1;
    read(DataFd, buffer, 4);
    NoOfRec = 0;
    nor = atoi(buffer);
    for(i=0; i<nor; i++){
      nread = read(DataFd, &DataArray[i], sizeof(DataArray[0]));
      if(nread != sizeof(DataArray[0]))
        break;
      NoOfRec++;
    }
    LogWrFd = open(LOGFILE, O_CREAT| O_TRUNC| O_WRONLY| O_BINARY);
    if(LogWrFd < 0)
      return -1;
    lseek(LogWrFd, 0L, SEEK_END);
    LogRdFd = open(LOGFILE, O_RDONLY| O_BINARY);
    if(LogRdFd < 0)
      return -1;
    return 1;
}
int InitCommPort()
{
  char CommSetting[20];
    sprintf(CommSetting, "%s:%s,%c,%c,%c", CommPort, CommSpeed,
            tolower(CommParity[0]), CommDataBit[0], CommStopBit[0]);
    CommID = OpenComm(CommPort, 1024, 1024);
    if(CommID < 0){
      MessageBox(hwMain, "Error from OpenComm().", "Comm Error", MB_ICONSTOP);
      return -1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    if(BuildCommDCB(CommSetting, &dcb) != 0){
      MessageBox(hwMain, "Error from BuildCommDCB().", "Comm Error", MB_ICONSTOP);
      return -1;
    }
    if(SetCommState(&dcb) != 0){
      MessageBox(hwMain, "Error from SetCommState().", "Comm Error", MB_ICONSTOP);
      return -1;
    }
    return 1;
}
int InitFirstInstance(HANDLE hInstance, HANDLE hPrevInstance, int CmdShow)
{
  WNDCLASS wc;
  wc.lpszClassName = szApplName;       //Window class name
  wc.hInstance     = hInstance;        //Instance
  wc.lpfnWndProc   = WinMainProc;      //Window function
  wc.style         = CS_HREDRAW|CS_VREDRAW;  //Window style
  wc.lpszMenuName  = (LPSTR)NULL;  //Menu name
  wc.hCursor       = LoadCursor(NULL, IDC_ARROW);
  wc.hIcon         = LoadIcon(hInstance, MAKEINTRESOURCE(IDICON));
  wc.hbrBackground = GetStockObject(LTGRAY_BRUSH);
  wc.cbClsExtra    = 0;                //No extra byte
  wc.cbWndExtra    = 0;                //No extra byte
  return(RegisterClass(&wc));
}
LRESULT CALLBACK WinMainProc(HWND hWnd, unsigned msg, WORD wParam, LONG
lParam)
{
  DLGPROC DlgProc;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
switch(msg){
  case WM_CREATE:
      GetProfileString("windows", "device", (LPSTR)"", (LPSTR)CookPrinterString, sizeof
(CookPrinterString));
      GetProfileString("windows", "device", (LPSTR)"", (LPSTR)CashPrinterString, sizeof
(CashPrinterString));
      hIns      = GetWindowWord(hwMain, GWW_HINSTANCE);
      hbCook    = CreateWindow("BUTTON", "พ่อครัว",
                                WS_CHILD| WS_VISIBLE|
                                BS_PUSHBUTTON| WS_TABSTOP| WS_GROUP,
                                15, 10, (MaxX/4)-20, (MaxY/2)-25,
                               hWnd, ID_COOK_BT,
                               hIns, NULL);
      hbCashier = CreateWindow("BUTTON", "แคชเชียร์",
                                WS_CHILD| WS_VISIBLE|
                                BS_PUSHBUTTON| WS_TABSTOP| WS_GROUP,
                                (MaxX/4)+10, 10, (MaxX/4)-20, (MaxY/2)-25,
                               hWnd, ID_CASHIER_BT,
                                hIns, NULL);
      hbFedit   = CreateWindow("BUTTON", "แก้ไข",
                                WS_CHILD| WS_VISIBLE|
                                BS_PUSHBUTTON| WS_TABSTOP| WS_GROUP,
                                (MaxX/2)+5, 10, (MaxX/4)-20, (MaxY/2)-25,
                              hWnd, ID_FEDIT_BT,
                                hIns, NULL);
      hbFupdate = CreateWindow("BUTTON", "ส่งออก",
                                WS_CHILD| WS_VISIBLE|
                                BS_PUSHBUTTON| WS_TABSTOP| WS_GROUP,
                                3*(MaxX/4), 10, (MaxX/4)-20, (MaxY/2)-25,
                               hWnd, ID_FUPDATE_BT,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                    hIns, NULL);
        hbExit    = CreateWindow("BUTTON", "จบ",
                                    WS_CHILD| WS_VISIBLE|
                                    BS_PUSHBUTTON| WS_TABSTOP| WS_GROUP,
                                    15, (MaxY/2), MaxX-35, (MaxY/2)-55,
                                  hWnd, ID_EXIT_BT,
                                    hIns, NULL);
        SetFocus(hbCook);
        break;
    case WM_CLOSE:
        PostMessage(hwMain, WM_COMMAND, FILE_EXIT, 0L);
        break;
    case WM_DESTROY:
        PostQuitMessage(0);
        break;
    case WM_TIMER:
        CommCheck();
        break;
    case UM_GETORDER:
        GetOrder();
        break;
    case UM_UPDATE:
        UpdateView(UPDATE_COOK);
        UpdateView(UPDATE_CASHIER);
      break;
    case WM_COMMAND:
        switch(wParam){
          case FILE_COOK:
          case ID_COOK_BT:
            DlgProc = (DLGPROC) MakeProcInstance(CookViewProc, hInst);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DialogBox(hInst, "COOK_DLG", hwMain, DlgProc);
            FreeProcInstance((FARPROC)DlgProc);
            break;
          case FILE_CASHIER:
        case ID_CASHIER_BT:
            DlgProc = (DLGPROC) MakeProcInstance(CashierViewProc, hInst);
            DialogBox(hInst, "CASHIER_DLG", hwMain, DlgProc);
            FreeProcInstance((FARPROC)DlgProc);
            break;
          case FILE_COOKPRN_SET:
            DlgProc = (DLGPROC) MakeProcInstance(CookPrnSetupProc, hInst);
            DialogBox(hInst, "COOKPRN_DLG", hwMain, DlgProc);
            FreeProcInstance((FARPROC)DlgProc);
            break;
          case FILE_CASHIERPRN_SET:
            DlgProc = (DLGPROC) MakeProcInstance(CashPrnSetupProc, hInst);
            DialogBox(hInst, "CASHPRN_DLG", hwMain, DlgProc);
            FreeProcInstance((FARPROC)DlgProc);
          break;
          case FILE_COMM_SET:
            DlgProc = (DLGPROC) MakeProcInstance(CommSetupProc, hInst);
            DialogBox(hInst, "COMM_DLG", hwMain, DlgProc);
            FreeProcInstance((FARPROC)DlgProc);
            break;
          case FILE_EXIT:
        case ID_EXIT_BT:
            if(MessageBox(hwMain, "ต้องการออกจากโปรแกรม?", "Exit",
MB_YESNO|MB_ICONEXCLAMATION) == IDYES){
              CloseComm(CommID);        //Close communication port before exit
              KillTimer(hwMain, 1);     //Kill timer before exit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BOOL FAR PASCAL AboutProc(HWND hDlg, unsigned msg, WORD wParam, LONG
lParam)
{
  switch(msg){
    case WM_INITDIALOG:
      return TRUE;
    case WM_CLOSE:
      PostMessage(hDlg, WM_COMMAND, ABOUT_DLG_OK, 0L);
      break;
    case WM_COMMAND:
      switch(wParam){
          case ABOUT_DLG_OK:
            EndDialog(hDlg, TRUE);
            break;
          default:
            return FALSE;
      }
      break;
    default:
      return FALSE;
  }
  return TRUE;
}
BOOL FAR PASCAL CookViewProc(HWND hDlg, unsigned msg, WORD wParam, LONG
lParam)
{
  int  i, j;
  char buffer[100];
  Order_t tOrder;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  switch(msg){
    case WM_INITDIALOG:
      hCookDlg = hDlg;
      GetOrder();
      return TRUE;
    case WM_CLOSE:
      PostMessage(hDlg, WM_COMMAND, COOK_EXIT_BT, 0L);
      break;
    case WM_COMMAND:
      switch(wParam){
          case COOK_EXIT_BT:
            hCookDlg = 0;
            EndDialog(hDlg, TRUE);
            break;
          case COOK_PRINT_BT:
          CookPrint();
            break;
            return FALSE;
      }
      break;
    default:
      return FALSE;
  }
  return TRUE;
}

BOOL FAR PASCAL CashierViewProc(HWND hDlg, unsigned msg, WORD wParam, LONG
lParam)
{
  char buffer[100];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
switch(msg){
  case WM_INITDIALOG:
    hCashierDlg = hDlg;
    PostMessage(hwMain, UM_UPDATE, 0, 0L);
    return TRUE;

  case WM_CLOSE:
    PostMessage(hDlg, WM_COMMAND, CASHIER_EXIT_BT, 0L);
    break;
  case WM_COMMAND:
    switch(wParam){
      case CASHIER_LEFT_BT:
        if(CashCurrentTable == 1)
          CashCurrentTable = 100;
        CashCurrentTable--;
        sprintf(buffer, "%2d", CashCurrentTable);
        SendDlgItemMessage(hCashierDlg, CASHIER_TABLE, EM_SETSEL, 0, 9L);
        SendDlgItemMessage(hCashierDlg, CASHIER_TABLE, EM_REPLACESEL, 0,
(LONG)(LPSTR)buffer);
        PostMessage(hwMain, UM_UPDATE, 0, 0L);
        break;
      case CASHIER_RIGHT_BT:
        if(CashCurrentTable == 99)
          CashCurrentTable = 0;
        CashCurrentTable++;
        sprintf(buffer, "%2d", CashCurrentTable);
        SendDlgItemMessage(hCashierDlg, CASHIER_TABLE, EM_SETSEL, 0, 9L);
        SendDlgItemMessage(hCashierDlg, CASHIER_TABLE, EM_REPLACESEL, 0,
(LONG)(LPSTR)buffer);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PostMessage(hwMain, UM_UPDATE, 0, 0L);
            break;
          case CASHIER_PRINT_BT:
          CashPrint();
            break;
          case CASHIER_EXIT_BT:
            hCashierDlg = 0;
            EndDialog(hDlg, TRUE);
            break;
          default:
            return FALSE;
        }
        break;
      default:
        return FALSE;
    }
    return TRUE;
}
BOOL FAR PASCAL CookPrnSetupProc(HWND hDlg, unsigned msg, WORD wParam,
LONG lParam)
{
    int  i, j;
    char buffer[300], temp[50], temp2[60], *PrnPtr[15];
    switch(msg){
      case WM_INITDIALOG:
        if((i = GetProfileString("devices", NULL, (LPSTR)"", (LPSTR)buffer, sizeof(buffer))) <
0)
            return FALSE;
        j = 0;
        for(i=0; i<15; i++){          //Max 15 printer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if(strlen(&buffer[j]) < 1)
              break;
            PrnPtr[i] = &buffer[j];
            j += strlen(&buffer[j]) + 1;
            GetProfileString("devices", PrnPtr[i], (LPSTR)"", (LPSTR)temp, sizeof(temp));
          sprintf(temp2, "%s,%s", PrnPtr[i], temp);
            SendDlgItemMessage(hDlg, COOKPRN_COMBO, CB_ADDSTRING, 0, (LONG)
(LPSTR)temp2);
        }
        SendDlgItemMessage(hDlg, COOKPRN_COMBO, CB_SELECTSTRING, 0, (LONG)
(LPSTR)CookPrinterString);
        return TRUE;
      case WM_CLOSE:
        PostMessage(hDlg, WM_COMMAND, COOKPRN_CANCEL, 0L);
        break;
      case WM_COMMAND:
        switch(wParam){
            case COOKPRN_CANCEL:
              EndDialog(hDlg, TRUE);
              break;
            case COOKPRN_OK:
              i = SendDlgItemMessage(hDlg, COOKPRN_COMBO, CB_GETCURSEL, 0, 0L);
              if(i != CB_ERR){
                SendDlgItemMessage(hDlg, COOKPRN_COMBO, CB_GETLBTEXT, i, (LONG)
(LPSTR)CookPrinterString);
              }
              EndDialog(hDlg, TRUE);
              break;
            default:
              return FALSE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        break;
      default:
        return FALSE;
    }
    return TRUE;
  }
  BOOL FAR PASCAL CashPrnSetupProc(HWND hDlg, unsigned msg, WORD wParam,
  LONG lParam)
  {
    int  i, j;
    char buffer[300], temp[50], temp2[60], *PrnPtr[15];
    switch(msg){
      case WM_INITDIALOG:
        if((i = GetProfileString("devices", NULL, (LPSTR)"", (LPSTR)buffer, sizeof(buffer))) <
  0)
            return FALSE;
        j = 0;
        for(i=0; i<15; i++){          //Max 15 printer
            if(strlen(&buffer[j]) < 1)
              break;
            PrnPtr[i] = &buffer[j];
            j += strlen(&buffer[j]) + 1;
            GetProfileString("devices", PrnPtr[i], (LPSTR)"", (LPSTR)temp, sizeof(temp));
          sprintf(temp2, "%s,%s", PrnPtr[i], temp);
            SendDlgItemMessage(hDlg, CASHPRN_COMBO, CB_ADDSTRING, 0, (LONG)
  (LPSTR)temp2);
        }
        SendDlgItemMessage(hDlg, CASHPRN_COMBO, CB_SELECTSTRING, 0, (LONG)
  (LPSTR)CashPrinterString);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return TRUE;
      case WM_CLOSE:
        PostMessage(hDlg, WM_COMMAND, CASHPRN_CANCEL, 0L);
        break;
    case WM_COMMAND:
        switch(wParam){
            case CASHPRN_CANCEL:
              EndDialog(hDlg, TRUE);
              break;
            case CASHPRN_OK:
              i = SendDlgItemMessage(hDlg, CASHPRN_COMBO, CB_GETCURSEL, 0, 0L);
              if(i != CB_ERR){
                SendDlgItemMessage(hDlg, CASHPRN_COMBO, CB_GETLBTEXT, i, (LONG)
(LPSTR)CashPrinterString);
              }
              EndDialog(hDlg, TRUE);
              break;
            default:
              return FALSE;
        }
        break;
      default:
        return FALSE;
    }
    return TRUE;
}

BOOL FAR PASCAL CommSetupProc(HWND hDlg, unsigned msg, WORD wParam, LONG
lParam)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int  i, j, Index;
  char buffer[100];
 switch(msg){
    case WM_INITDIALOG:
  SendDlgItemMessage(hDlg, COMM_PORT, CB_ADDSTRING, 0, (LONG)(LPSTR)"
COM1");
     SendDlgItemMessage(hDlg, COMM_PORT, CB_ADDSTRING, 0, (LONG)(LPSTR)"
COM2");
     SendDlgItemMessage(hDlg, COMM_PORT, CB_ADDSTRING, 0, (LONG)(LPSTR)"
COM3");
     SendDlgItemMessage(hDlg, COMM_PORT, CB_ADDSTRING, 0, (LONG)(LPSTR)"
COM4");
     SendDlgItemMessage(hDlg, COMM_PORT, CB_SELECTSTRING, 0, (LONG)(LPSTR)
CommPort);
     SendDlgItemMessage(hDlg, COMM_SPEED, CB_ADDSTRING, 0, (LONG)(LPSTR)
"9600");
     SendDlgItemMessage(hDlg, COMM_SPEED, CB_ADDSTRING, 0, (LONG)(LPSTR)
"4800");
     SendDlgItemMessage(hDlg, COMM_SPEED, CB_ADDSTRING, 0, (LONG)(LPSTR)
"2400");
     SendDlgItemMessage(hDlg, COMM_SPEED, CB_ADDSTRING, 0, (LONG)(LPSTR)
"1200");
     SendDlgItemMessage(hDlg, COMM_SPEED, CB_SELECTSTRING, 0, (LONG)
(LPSTR)CommSpeed);
     SendDlgItemMessage(hDlg, COMM_DATA, CB_ADDSTRING, 0, (LONG)(LPSTR)"7
บิท");
     SendDlgItemMessage(hDlg, COMM_DATA, CB_ADDSTRING, 0, (LONG)(LPSTR)"8
บิท");
     SendDlgItemMessage(hDlg, COMM_DATA, CB_SELECTSTRING, 0, (LONG)(LPSTR)
CommDataBit);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SendDlgItemMessage(hDlg, COMM_STOP, CB_ADDSTRING, 0, (LONG)(LPSTR)"1
บิท");
        SendDlgItemMessage(hDlg, COMM_STOP, CB_ADDSTRING, 0, (LONG)(LPSTR)"2
บิท");
        SendDlgItemMessage(hDlg, COMM_STOP, CB_SELECTSTRING, 0, (LONG)(LPSTR)
CommStopBit);
        SendDlgItemMessage(hDlg, COMM_PARITY, CB_ADDSTRING, 0, (LONG)(LPSTR)"
None");
        SendDlgItemMessage(hDlg, COMM_PARITY, CB_ADDSTRING, 0, (LONG)(LPSTR)"
Odd");
        SendDlgItemMessage(hDlg, COMM_PARITY, CB_ADDSTRING, 0, (LONG)(LPSTR)"
Even");
        SendDlgItemMessage(hDlg, COMM_PARITY, CB_SELECTSTRING, 0, (LONG)
(LPSTR)CommParity);
        return TRUE;
      case WM_CLOSE:
        PostMessage(hDlg, WM_COMMAND, COMM_CANCEL, 0L);
        break;
      case WM_COMMAND:
        switch(wParam){
          case COMM_CANCEL:
            EndDialog(hDlg, TRUE);
            break;
          case COMM_OK:
            Index = SendDlgItemMessage(hDlg, COMM_PORT, CB_GETCURSEL, 0, 0L);
            SendDlgItemMessage(hDlg, COMM_PORT, CB_GETLBTEXT, Index, (LONG)
(LPSTR)CommPort);
            Index = SendDlgItemMessage(hDlg, COMM_SPEED, CB_GETCURSEL, 0, 0L);
            SendDlgItemMessage(hDlg, COMM_SPEED, CB_GETLBTEXT, Index, (LONG)
(LPSTR)CommSpeed);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Index = SendDlgItemMessage(hDlg, COMM_DATA, CB_GETCURSEL, 0, 0L);
            SendDlgItemMessage(hDlg, COMM_DATA, CB_GETLBTEXT, Index, (LONG)
(LPSTR)CommDataBit);
            Index = SendDlgItemMessage(hDlg, COMM_STOP, CB_GETCURSEL, 0, 0L);
            SendDlgItemMessage(hDlg, COMM_STOP, CB_GETLBTEXT, Index, (LONG)
(LPSTR)CommStopBit);
            Index = SendDlgItemMessage(hDlg, COMM_PARITY, CB_GETCURSEL, 0, 0L);
            SendDlgItemMessage(hDlg, COMM_PARITY, CB_GETLBTEXT, Index, (LONG)
(LPSTR)CommParity);
            EndDialog(hDlg, TRUE);
            break;
          default:
            return FALSE;
        }
        break;
      default:
        return FALSE;
    }
    return TRUE;
}
BOOL FAR PASCAL EditFoodProc(HWND hDlg, unsigned msg, WORD wParam, LONG
lParam)
{
    char szType[3],
         szCode[3],
        szPrice[5],
         szName[NAMELEN],
         szListBoxText[NAMELEN+50],
         buffer[100];
    int i,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            nor,
            nType,
            nCode,
           nPrice,
            nIndex;
    switch(msg){
     case WM_INITDIALOG:   //First time when dialog was loaded
 //Display all available food lists
       hEditDlg = hDlg;
       for(i=0; i<NoOfRec; i++){  //Copy from DataArray to ListBox
          sprintf(szListBoxText, "%5c%02d%2c%02d%5c%04d%5c%s",
                                 ' ', DataArray[i].nType,
                                 ' ', DataArray[i].nCode,
                                 ' ', DataArray[i].nPrice,
                                 ' ', DataArray[i].szName);
          SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_ADDSTRING, 0,
                          (LONG)(LPSTR)szListBoxText);
       }
       return TRUE;
     case WM_CLOSE:  //When double click at system box
       PostMessage(hDlg, WM_COMMAND, EDF_EXIT, 0L);
       break;
     case WM_COMMAND:
       switch(wParam){
          case EDF_ADD:          //ADD button was press
 //Get value of EDF_INTYPE field and convert to integer
            GetDlgItemText(hDlg, EDF_INTYPE, szType, sizeof(szType));
            if(!strlen(szType) || !IsNumber(szType)){
              MessageBox(hDlg, "ชนิดอาหารต้องเป็นตัวเลข", "Add Item",
                              MB_OK | MB_ICONINFORMATION);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SetFocus(GetDlgItem(hDlg, EDF_INTYPE));
            break;
          }
          nType = atoi(szType);
- //Get value of EDF_INCODE field and convert to integer
          GetDlgItemText(hDlg, EDF_INCODE, szCode, sizeof(szCode));
          if(!strlen(szCode) || !IsNumber(szCode)){
            MessageBox(hDlg, "รหัสอาหารต้องเป็นตัวเลข", "Add Item",
                    MB_OK | MB_ICONINFORMATION);
            SetFocus(GetDlgItem(hDlg, EDF_INCODE));
          break;
          }
          nCode = atoi(szCode);
 //Get value of EDF_INNAME field
          GetDlgItemText(hDlg, EDF_INNAME, szName, 20);
          if(szName[0] == '\0'){
           MessageBox(hDlg, "ชื่อรายการอาหารจะต้องมีค่า", "Add Item",
                    MB_OK | MB_ICONINFORMATION);
            SetFocus(GetDlgItem(hDlg, EDF_INNAME));
            break;
          }
 //Get value of EDF_INPRICE field and convert to integer
          GetDlgItemText(hDlg, EDF_INPRICE, szPrice, sizeof(szPrice));
          if(!strlen(szPrice) || !IsNumber(szPrice)){
            MessageBox(hDlg, "ราคาอาหารต้องเป็นตัวเลข", "Add Item",
                    MB_OK | MB_ICONINFORMATION);
            SetFocus(GetDlgItem(hDlg, EDF_INPRICE));
          break;
          }
          nPrice = atoi(szPrice);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//Check for duplication of primary key
            for(i=0; i<NoOfRec; i++){
              if((nType == DataArray[i].nType) &&
                  (nCode == DataArray[i].nCode)){
                MessageBox(hDlg, "ชนิดและรหัสรายการอาหารซ้ำ", "Add Item",
                        MB_OK| MB_ICONINFORMATION);
                SetFocus(GetDlgItem(hDlg, EDF_INTYPE));
                return FALSE;
              }
            }
            sprintf(szListBoxText, "%5c%02d%2c%02d%5c%04d%5c%s",
                                ' ', nType,
                                ' ', nCode,
                                ' ', nPrice,
                                ' ', szName);
            SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_ADDSTRING, 0,
                        (LONG)(LPSTR)szListBoxText);
            SetDlgItemText(hDlg, EDF_INTYPE, "");
            SetDlgItemText(hDlg, EDF_INCODE, "");
            SetDlgItemText(hDlg, EDF_INPRICE, "");
            SetDlgItemText(hDlg, EDF_INNAME, "");
            SetFocus(GetDlgItem(hDlg, EDF_INTYPE));
            DataArray[NoOfRec].nType  = nType;
            DataArray[NoOfRec].nCode  = nCode;
            DataArray[NoOfRec].nPrice = nPrice;
            strcpy(DataArray[NoOfRec].szName, szName);
            NoOfRec++;
            break;
          case EDF_REMOVE:
//Get current selection list
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            nIndex = SendDlgItemMessage(hDlg, EDF_LISTBOX,
                                        LB_GETCURSEL, 0, 0L);
            if(nIndex == LB_ERR){ //error if not available or not selected
              MessageBox(hDlg, "ไม่มีรายการหรือไม่ได้เลือกรายการที่จะลบ",
                        "Delete Item", MB_OK| MB_ICONEXCLAMATION);
              SetFocus(GetDlgItem(hDlg, EDF_LISTBOX));
            return FALSE;
            }else{
//If no error then delete the selected list and shift others list
              SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_DELETESTRING,
                                 nIndex, 0L);
              SetFocus(GetDlgItem(hDlg, EDF_LISTBOX));
              if((nIndex == SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_GETCOUNT, 0, 0
L))
                && (nIndex != 0))
                nIndex--;
              SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_SETCURSEL, nIndex, 0L);
              NoOfRec = SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_GETCOUNT, 0, 0
L);
//Update sequence of list in structure
              for(i=0; i<NoOfRec; i++){
                SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_GETTEXT, i,
                                   (LONG)(LPSTR)szListBoxText);
                memcpy(buffer, &szListBoxText[5], 2);
                buffer[2] = '\0';
                DataArray[i].nType = atoi(buffer);
                memcpy(buffer, &szListBoxText[5+2+2], 2);
                buffer[2] = '\0';
                DataArray[i].nCode = atoi(buffer);
                memcpy(buffer, &szListBoxText[5+2+2+2+5], 4);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                buffer[4] = '\0';
                DataArray[i].nPrice = atoi(buffer);
                strcpy(DataArray[i].szName, &szListBoxText[5+2+2+2+5+4+5]);
              }
            }
            break;
          case EDF_PRINT:
            FoodListPrint();
            break;
          case EDF_EXIT:
//            NoOfRec = SendDlgItemMessage(hDlg, EDF_LISTBOX, LB_GETCOUNT, 0, 0L);
          lseek(DataFd, 0L, SEEK_SET); //Seek file pointer to begin of file
            sprintf(buffer, "%04d", NoOfRec);
            write(DataFd, buffer, 4);        //Write first 4 byte (Number of records)
//For each record , fill values into structure and write it to file
            for(i=0; i<NoOfRec; i++){
              write(DataFd, (LPSTR)&DataArray[i], sizeof(DataArray[0]));
            }
            hEditDlg = 0;
            EndDialog(hDlg, TRUE);
            break;
          default:
            return FALSE;
        }
        break;
      default:
        return FALSE;
    }
    return TRUE;
  }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int SendFoodList(void){
  int  i;
  char temp[100];
  sprintf(temp, "%c%c%c%c", PRECODE, PRECODE, PRECODE, PRECODE);
  if(WriteComm(CommID, (LPSTR)temp, 4) != 4)
    MessageBox(GetActiveWindow(), "Write error", "cx", MB_OK);
  for(i=0; i<NoOfRec; i++){  //For each food record
    sprintf(temp, "%c%c%s                    ", DataArray[i].nType,
            DataArray[i].nCode, DataArray[i].szName);
    temp[22] = '\0';  //copy only 22 byte
    if(WriteComm(CommID, (LPSTR)temp, 22) != 22)     //Write 22 bytes
      MessageBox(GetActiveWindow(), "Write error1", "cx", MB_OK);
  }
  sprintf(temp, "%c%c%c%c", POSTCODE, POSTCODE, POSTCODE, EOF);
  WriteComm(CommID, (LPSTR)temp, 4);
  MessageBox(GetActiveWindow(), "Send food list OK.", "Sendfood",
MB_OK|MB_ICONINFORMATION);
  return TRUE;
}
int CookPrint(void){
  int  i, tx, ty;
  char temp[100], PrnName[30], PrnDriver[15], PrnInterface[10], *p;
  char szName[] = "Order";
  HDC  hPrnDC;
  DOCINFO di;
  TEXTMETRIC tm;
  HFONT  hFont,
         hFontOld;
//Get printer information from config.
  strcpy(temp, CookPrinterString);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
p = strtok(temp, ",");
strcpy(PrnName, p);
p = strtok(NULL, ",");
strcpy(PrnDriver, p);
p = strtok(NULL, ",");
strcpy(PrnInterface, p);
//Create DC
hPrnDC = CreateDC(PrnDriver, PrnName, PrnInterface, NULL);
if(hPrnDC == NULL){
  MessageBox(GetActiveWindow(), "Error to initialize printer.", "Error", MB_OK|
MB_ICONSTOP);
  return FALSE;
}
di.cbSize      = sizeof(DOCINFO);
di.lpszDocName = szName;
di.lpszOutput  = NULL;
hFont = CreateFont(-35,0,0,0,0,0,0,0,0,0,0,0,0,0,NULL);
hFontOld = SelectObject(hPrnDC, hFont);
GetTextMetrics(hPrnDC, &tm);
tx = tm.tmAveCharWidth;
ty = tm.tmHeight;
i = StartDoc(hPrnDC, &di);
if(!i){
  MessageBox(GetActiveWindow(), "Error to StartDoc().", "Error", MB_OK|
MB_ICONSTOP);
  DeleteDC(hPrnDC);
  return FALSE;
}
StartPage(hPrnDC);
hFont = CreateFont(-40,0,0,0,FW_BOLD,0,1,0,0,0,0,0,0,0,NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  SelectObject(hPrnDC, hFont);
  strcpy(temp, "รายการอาหารสั่งทำ");
  TextOut(hPrnDC, 40*tx, 2*ty, temp, strlen(temp));
  hFont = CreateFont(-35,0,0,0,FW_BOLD,0,0,0,0,0,0,0,0,NULL);
  SelectObject(hPrnDC, hFont);
  strcpy(temp, " เร่ง  โต๊ะ   จำนวน      ชื่ออาหาร");
  TextOut(hPrnDC, 10*tx, 4*ty, temp, strlen(temp));
  hFont = CreateFont(-35,0,0,0,0,0,0,0,0,0,0,0,0,NULL);
  SelectObject(hPrnDC, hFont);
  for(i=0; i<NoOfOrder; i++){
    Order[i].bPrint = TRUE;
    Order[i].nPrice = Order[i].nQty * GetPrice(Order[i].nType, Order[i].nCode);
    sprintf(temp, "        %2d          %2d          %s", Order[i].nTable, Order[i].nQty,
                  Order[i].szName);
    if(Order[i].bFast)
      temp[3] = '*';
    TextOut(hPrnDC, 10*tx, (i+5)*ty, temp, strlen(temp));
  }
  EndPage(hPrnDC);
  EndDoc(hPrnDC);
  DeleteDC(hPrnDC);
  if(hFontOld != NULL)
    SelectObject(hPrnDC, hFontOld);
  PostMessage(hwMain, UM_GETORDER, 0, 0L);
  return TRUE;
}
int CashPrint(void){
  int  i, j, n, tx, ty, nTotalPrice;
  char temp[100], PrnName[30], PrnDriver[15], PrnInterface[10], *p;
  char szName[] = "Reciept";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  HDC  hPrnDC;
  DOCINFO di;
  TEXTMETRIC tm;
  HFONT  hFont,
         hFontOld;
//Get printer information from config.
  strcpy(temp, CashPrinterString);
  p = strtok(temp, ",");
  strcpy(PrnName, p);
  p = strtok(NULL, ",");
  strcpy(PrnDriver, p);
  p = strtok(NULL, ",");
  strcpy(PrnInterface, p);
//Create DC
  hPrnDC = CreateDC(PrnDriver, PrnName, PrnInterface, NULL);
  if(hPrnDC == NULL){
    MessageBox(GetActiveWindow(), "Error to initialize printer.", "Error", MB_OKI
MB_ICONSTOP);
    return FALSE;
  }
  di.cbSize      = sizeof(DOCINFO);
  di.lpszDocName = szName;
  di.lpszOutput  = NULL;
  hFont = CreateFont(-35,0,0,0,0,0,0,0,0,0,0,0,0,NULL);
  hFontOld = SelectObject(hPrnDC, hFont);
  GetTextMetrics(hPrnDC, &tm);
  tx = tm.tmAveCharWidth;
  ty = tm.tmHeight;
  i = StartDoc(hPrnDC, &di);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MessageBox(GetActiveWindow(), "Error to StartDoc().", "Error", MB_OK|
MB_ICONSTOP);
    DeleteDC(hPrnDC);
    return FALSE;
   }
   StartPage(hPrnDC);
   hFont = CreateFont(-40,0,0,0,FW_BOLD,0,1,0,0,0,0,0,0,0,NULL);
   SelectObject(hPrnDC, hFont);
   strcpy(temp, "ใบเสร็จรับเงิน");
   TextOut(hPrnDC, 40*tx, 2*ty, temp, strlen(temp));

   hFont = CreateFont(-35,0,0,0,FW_BOLD,0,0,0,0,0,0,0,0,0,NULL);
   SelectObject(hPrnDC, hFont);
   strcpy(temp, " จำนวน  ราคา        ชื่ออาหาร");
   TextOut(hPrnDC, 10*tx, 5*ty, temp, strlen(temp));
   hFont = CreateFont(-35,0,0,0,0,0,0,0,0,0,0,0,0,0,NULL);
   SelectObject(hPrnDC, hFont);
   sprintf(temp, " โต๊ะ  %2d", CashCurrentTable);
   TextOut(hPrnDC, 10*tx, 4*ty, temp, strlen(temp));

//Print all items of current table and delete them
   nTotalPrice = 0;
   j = 0;
   for(i=0; i<NoOfOrder; i++){
    if((Order[i].nTable == CashCurrentTable) && (Order[i].bPrint)){
     sprintf(temp, "      %2d          %2d          %s", Order[i].nQty,
                Order[i].nPrice, Order[i].szName);
     TextOut(hPrnDC, 10*tx, (j+6)*ty, temp, strlen(temp));
     j++;
     nTotalPrice += Order[i].nPrice;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      for(n=i; n<NoOfOrder-1; n++){       //Shift orders up
        CopyOrder(&Order[n], &Order[n+1]);
      }
      i--;    //Repeat this item again
      NoOfOrder--;
    }
  }
  sprintf(temp, "                    รวมเงิน : %d  บาท.", nTotalPrice);
  TextOut(hPrnDC, 10*tx, (j+7)*ty, temp, strlen(temp));
  EndPage(hPrnDC);
  EndDoc(hPrnDC);
  DeleteDC(hPrnDC);
  if(hFontOld != NULL)
    SelectObject(hPrnDC, hFontOld);
  PostMessage(hwMain, UM_UPDATE, 0, 0L);
  return TRUE;
}
int FoodListPrint(void){
  TEXTMETRIC tm;
  HFONT  hFont,
         hFontOld;
  HDC  hPrnDC;
  char temp[100], PrnName[30], PrnDriver[15], PrnInterface[10], *p;
  char szName[] = "FoodList";
  DOCINFO di;
  int  tx, ty, i;
//Get printer information from config.
  strcpy(temp, CashPrinterString);
  p = strtok(temp, ",");
  strcpy(PrnName, p);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    p = strtok(NULL, ",");
    strcpy(PrnDriver, p);
    p = strtok(NULL, ",");
    strcpy(PrnInterface, p);
  //Create DC
    hPrnDC = CreateDC(PrnDriver, PrnName, PrnInterface, NULL);
    if(hPrnDC == NULL){
      MessageBox(GetActiveWindow(), "Error to initialize printer.", "Error", MB_OKI
  MB_ICONSTOP);
      return FALSE;
    }
    di.cbSize      = sizeof(DOCINFO);
    di.lpszDocName = szName;
    di.lpszOutput  = NULL;
    hFont = CreateFont(-35,0,0,0,0,0,0,0,0,0,0,0,0,NULL);
    hFontOld = SelectObject(hPrnDC, hFont);
    GetTextMetrics(hPrnDC, &tm);
    tx = tm.tmAveCharWidth;
    ty = tm.tmHeight;
    i = StartDoc(hPrnDC, &di);
    if(!i){
      MessageBox(GetActiveWindow(), "Error to StartDoc().", "Error", MB_OKI
  MB_ICONSTOP);
      DeleteDC(hPrnDC);
      return FALSE;
    }
    StartPage(hPrnDC);
    hFont = CreateFont(-40,0,0,0,FW_BOLD,0,1,0,0,0,0,0,0,NULL);
    SelectObject(hPrnDC, hFont);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  strcpy(temp, "รายการอาหาร");
  TextOut(hPrnDC, 40*tx, 2*ty, temp, strlen(temp));
  hFont = CreateFont(-35,0,0,0,FW_BOLD,0,0,0,0,0,0,0,0,NULL);
  SelectObject(hPrnDC, hFont);
  strcpy(temp, "ประเภท รหัส     ราคา          ชื่ออาหาร      ");
  TextOut(hPrnDC, 10*tx, 4*ty, temp, strlen(temp));
  hFont = CreateFont(-35,0,0,0,0,0,0,0,0,0,0,0,0,NULL);
  SelectObject(hPrnDC, hFont);
  for(i=0; i<NoOfRec; i++){
    sprintf(temp, "     %02d        %02d        %4d        %s", DataArray[i].nType,
          DataArray[i].nCode, DataArray[i].nPrice, DataArray[i].szName);
    TextOut(hPrnDC, 10*tx, (5+i)*ty, temp, strlen(temp));
  }
  EndPage(hPrnDC);
  EndDoc(hPrnDC);
  DeleteDC(hPrnDC);
  if(hFontOld != NULL)
    SelectObject(hPrnDC, hFontOld);
  return TRUE;
}
int  GetPrice(int nType, int nCode){
  int i;

  for(i=0; i<NoOfRec; i++){
    if((DataArray[i].nType == nType) && (DataArray[i].nCode == nCode))
      return(DataArray[i].nPrice);
  }
  return(0);
}
int CommCheck(void){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int  i,
       n;
  static int Inused = 0;
  char buffer[100];
  COMSTAT Comstat;
  if(Inused)
    return -1;
  else
    Inused++;
  GetCommError(CommID, (COMSTAT FAR *) &Comstat);
  n = 1;
  while((Comstat.cbInQue > 0) && (n > 0)){
    n = ReadComm(CommID, (LPSTR)buffer, sizeof(buffer));
    write(LogWrFd, buffer, n);
    GetCommError(CommID, (COMSTAT FAR *) &Comstat);
    SendMessage(hwMain, UM_GETORDER, 0, 0L);
  PostMessage(hwMain, UM_UPDATE, 0, 0L);
  Inused = 0;
  return 1;
}
int IsNumber(char *str){
   int i,
       len;
   len = strlen(str);
   for(i=0; i<len; i++){
     if(!isdigit(str[i]))
       return 0;
   }
   return 1;
int ReadOrder(char *OrderStr){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  int  n,
       OldP;
  char Buffer[10];
  OldP = LogRdFd;       //Save old value of data file pointer
  do{
    n = read(LogRdFd, Buffer, 1);
  }while((n > 0) && ((Buffer[0] == (char)PRECODE) || (Buffer[0] == (char)POSTCODE)));
//PRECODE 'FE' or 'FF'
  if(n < 1){
    LogRdFd = OldP;     //Restore old value and error
    return -1;
  }
  n = read(LogRdFd, &Buffer[1], 4);
  if(n < 4){
    LogRdFd = OldP;
    return -1;
  }
  memcpy(OrderStr, Buffer, 5);  //Copy 5 bytes back
  return 5;
}
int IsDup(int nType, int nCode){
  int  i;
  for(i=0; i<NoOfRec; i++){
    if((DataArray[i].nType == nType) && (DataArray[i].nCode == nCode)){
      return i;
    }
  }
  return -1;
int GetOrder(void){       //Get order from log file
  static int bUsed=FALSE;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int  i, j, n,
     nCmd,
     nType,
     nCode,
     nTable,
     nQty;
char szName[NAMELEN],
     Buffer[20],
     Buf[10];
BOOL bFound;
if(bUsed)
  return(0);
bUsed = TRUE;
bFound = FALSE;
for(;;){      //Exit when can't get any more order
  if((i = ReadOrder(Buffer)) < 0){
    bUsed = FALSE;
    break;                          //No order available
  }
  bFound = TRUE;
  nCmd   = (int)Buffer[0];          //Convert to integer
  nTable = (int)Buffer[1];          //Convert to integer
  nType  = (int)Buffer[2];          //Convert to integer
  nCode  = (int)Buffer[3];          //Convert to integer
  nQty   = (int)Buffer[4];          //Convert to integer

  switch(nCmd){
    case ORDERCMD:
      n = IsDup(nType, nCode);  //Is has food Type and Code in table
      if(n < 0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;                  //Error
      strcpy(szName, DataArray[n].szName);  //Copy name of food

      for(n=0; n<NoOfOrder; n++){ //Find record in Order[]
        if((Order[n].nType  == nType)  && (Order[n].nCode == nCode) &&
          (Order[n].nTable == nTable) && (!Order[n].bPrint)){
          Order[n].nQty += nQty; //If exist then add it
          break;          //exit for() loop
        }
      if(n >= NoOfOrder){       //If not found then add new order
        Order[NoOfOrder].bFast  = FALSE;    //Not fast
        Order[NoOfOrder].nTable = nTable;   //Table
      Order[NoOfOrder].nType  = nType;
        Order[NoOfOrder].nCode  = nCode;
        Order[NoOfOrder].nQty   = nQty;
      Order[NoOfOrder].bPrint = FALSE;
        strcpy(Order[NoOfOrder].szName, szName);
        NoOfOrder++;
      }
      break;
    case FASTCMD:
     for(n=0; n<NoOfOrder; n++){  //Find match record in Order[]
        if((Order[n].nType  == nType) &&
          (Order[n].nCode  == nCode) && (!Order[n].bPrint)){
         Order[n].bFast = TRUE;
        }
      }
     break;
    case CANCELCMD:
     for(n=0; n<NoOfOrder; n++){  //Find match record
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if((Order[n].nTable == nTable) && (Order[n].nType  == nType) &&
            (Order[n].nCode  == nCode)  && (!Order[n].bPrint)){
           if((nQty > 0) && (nQty < Order[n].nQty)){    //O = All , others = delete nQty from
Order
           Order[n].nQty -= nQty;
           }else{
//Replace current record with the last record
           CopyOrder(&Order[n], &Order[NoOfOrder-1]);
             NoOfOrder--;
          }
          break;
        }
        break;
    }
   PostMessage(hwMain, UM_UPDATE, 0, 0L); //call for update view if changed
  return(1);
}
int UpdateView(int nCmd){
int  i,
     nTotal=0;   //Total price
char buffer[100];
BOOL bFound;
  switch(nCmd){
    case UPDATE_COOK:     //Update cook view command
      if(hCookDlg != 0){  //If cook view is displaying then show lists
//Clear all items in list box and then update all items from Order[]
        SendDlgItemMessage(hCookDlg, COOK_LIST, LB_RESETCONTENT, 0, 0L);
        for(i=0; i<NoOfOrder; i++){
          if(!Order[i].bPrint){
            if(Order[i].bFast){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                sprintf(buffer, "   *  %02d%5c%02d%5c%s", Order[i].nTable, ' ', Order[i].nQty  , '
', Order[i].szName);
              }else{
                sprintf(buffer, "      %02d%5c%02d%5c%s", Order[i].nTable, ' ', Order[i].nQty  , '
', Order[i].szName);
              SendDlgItemMessage(hCookDlg, COOK_LIST, LB_ADDSTRING, 0, (LONG)
(LPSTR)buffer);
            }
          }
      }
      break;
    case UPDATE_CASHIER:
      if(hCashierDlg != 0){
        for(i=0; i<NoOfOrder; i++){
          if((Order[i].nTable == CashCurrentTable) && (Order[i].bPrint)){
          bFound = TRUE;
            break;
          }
        }
        if(!bFound){
          MessageBox(hCashierDlg, "No order for this table", "Error", MB_OK |
MB_ICONINFORMATION);
          return(-1);
        }
        sprintf(buffer, "%2d", CashCurrentTable);
        SendDlgItemMessage(hCashierDlg, CASHIER_TABLE, EM_SETSEL, 0, 9L);
        SendDlgItemMessage(hCashierDlg, CASHIER_TABLE, EM_REPLACESEL, 0,
(LONG)(LPSTR)buffer);
        SendDlgItemMessage(hCashierDlg, CASHIER_LIST, LB_RESETCONTENT, 0, 0L);
        for(i=0; i<NoOfOrder; i++){
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if((Order[i].nTable == CashCurrentTable) && (Order[i].bPrint)){
              sprintf(buffer, "%5c%02d%5c%04d%5c%s", ' ', Order[i].nQty,
                                  ' ', Order[i].nPrice, ' ',Order[i].szName);
              SendDlgItemMessage(hCashierDlg, CASHIER_LIST, LB_ADDSTRING, 0,
(LONG)(LPSTR)buffer);
              nTotal += Order[i].nPrice;
            }
            }
            sprintf(buffer, "%9d", nTotal);
            SendDlgItemMessage(hCashierDlg, CASHIER_TOTAL, EM_SETSEL, 0, 9L);
            SendDlgItemMessage(hCashierDlg, CASHIER_TOTAL, EM_REPLACESEL, 0,
(LONG)(LPSTR)buffer);
        }
        break;
    }
    return 1;
}

void CopyOrder(Order_t *DestOrder, Order_t *SrcOrder)
{
  DestOrder->nTable = SrcOrder->nTable;
  DestOrder->nType  = SrcOrder->nType;
  DestOrder->nCode  = SrcOrder->nCode;
  DestOrder->nQty   = SrcOrder->nQty;
  DestOrder->nPrice = SrcOrder->nPrice;
  DestOrder->bFast  = SrcOrder->bFast;
  DestOrder->bPrint = SrcOrder->bPrint;
  strcpy(DestOrder->szName, SrcOrder->szName);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;       * * * * * * * * * * * * * * * * * * * **
;       *        >>>>  PROJECT  <<<<              *
;       *    SOFTWARE FOR DATA-COMMUNICATION     *
;       *     AND DISPLAY THAI CHARACTER         *
;       *          ON LCD.....GRAPHIC            *
;       *    DEVELOPE BY.......                  *
;       *                   MR.ANUROJ SEAKHOW    *
;       *         ELECTRONIC ENGINEERING         *
;       *         KMIT. LADKRABANG               *
;       * * * * * * * * * * * * * * * * * * * **
            CPU  "Z80.TBL"
            HOF  "INT8"
;
          ORG  8000H
;       * * * * * * * *
;       *    MAIN()    *
;       * * * * * * * *
            CALL MENU1
            RST  18H
 MN1:       DFB   "1.แสดงรายการอาหาร",0DH
            DFB   "2.เรียกการสั่งอาหาร",0DH
            DFB   "3.ยกเลิกอาหาร",0DH
            DFB   "4.แสดงรายการพิเศษ",0DH
            DFB   "5.รวมยอดขายจากพอร์ต",0DH
            DFB   "6.สั่งยอดขายออกพอร์ต",0DH
            DFB   "7.ตรวจสอบหน่วยระบบ",0DH
            DFB   "8.ออกจากระบบ",0DH
; Define Parameter for program
       DP01:      EQU   81H
       DP02:      EQU   80H
       LDP01:     EQU   81H
       LDP02:     EQU   80H
       LP01:      EQU   81H
       LP02:      EQU   80H
       CP01:      EQU   81H
       CP02:      EQU   80H
       SRP01:     EQU   85H
       SRP02:     EQU   84H
       WP01:      EQU   83H
       WP02:      EQU   82H
       RP01:      EQU   87H
       RP02:      EQU   86H
       STD:       EQU   0FCH          ; STD =0C0H
       STL:       EQU   0B8H
       STC:       EQU   40H
       Pctrl:     EQU   34h           ; Port of keybroad
       Pdout:     EQU   30h
       Wctrl:     EQU   0F0h
       Wstrt:     EQU   01110111b
;  End define parameter
       LD   SP,STACK
;      * * * * * * * * * * * * * * * * *
;      *        Display ON state        *
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * * * * * * * * * * * * * * * * *
  DISON: LD   B,3FH              ; data
         LD   C,DP01             ; On Display 1.
         CALL OPDC1
         LD   C,DP02             ; On Display 2.
         CALL OPDC2
         RET
;     * * * * * * * * * * * * * * * * *
;     *     Display OFF State          *
;     * * * * * * * * * * * * * * * * *
 DISOFF: LD   B,3EH              ; data
         LD   C,DP01             ; Off Display 1.
         CALL OPDC1
         LD   C,DP02             ; Off Display 2.
         CALL OPDC2
         RET
         ORG 8100H
;     * * * * * * * * * * * * * * * * *
;     *         Clear  Display         *
;     * * * * * * * * * * * * * * * * *
  CLRD:  CALL SSRT               ; set start status on display
         LD  H,08H               ; counter of line
         LD  E,STL               ; set start line
  CL1:   LD  L,40H               ; counter of column
         LD  D,STC               ; set start column
         LD  A,00H               ; minimum limit is 00h
  CL2:   LD  B,D                 ; Start out DATA 00h
         LD  C,CP01
         CALL OPDC1              ; set column display 1.
         LD  C,CP02
         CALL OPDC2              ; set column display 2.
         LD  B,00H               ; data for display 00h
         LD  C,WP01
         CALL OPDC1              ; write data D1.
         LD  C,WP02
         CALL OPDC2              ; write data D2.
         INC D                   ; increat column
         DEC  L
         CP   L                  ; compare counter column
         JR   NZ,CL2
         INC E                   ; increat line
         LD  B,E
         LD  C,LP01
         CALL OPDC1
         LD  C,LP02
         CALL OPDC2
         DEC H
         CP  H                   ; compare counter line
         JR  NZ,CL1
         CALL CLSBUF
         RET
;     * * * * * * * * * * * * * * * * *
;     *       Reverse  Display         *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * * * * * * * * * * * * * * * * *
  REVD:  CALL SSRT
         LD  H,08H
         LD  E,STL
  RE1:   LD  L,40H
         LD  D,STC
         LD  A,00H
  RE2:   LD  B,D
         LD  C,CP01
         CALL OPDC1
         LD  C,CP02
         CALL OPDC2
         LD  B,0FFH
         LD  C,WP01
         CALL OPDC1
         LD  C,WP02
         CALL OPDC2
         LD  C,LP02
         CALL OPDC2
         DEC H
         CP  H
         JR  NZ,RE1
         CALL DISOFF
         RST 18H
;     * * * * * * * * * * * * * * * * *
;     *     Delay  time             *
;     * * * * * * * * * * * * * * * * *
  DELAY: PUSH AF
         PUSH BC
         LD C,0FH
         LD A,00H
  DL1:   LD B,0FFH
  DL2:   NOP
         DJNZ DL2
         DEC C
         CP C
         JR NZ,DL1
         POP BC
         POP BC
         RET
;     * * * * * * * * * * * * * * * * *
;     *     LONG Delay  time          *
;     * * * * * * * * * * * * * * * * *
 LDELAY: PUSH AF
         PUSH BC
         LD C,0FFH
         LD A,00H
 LDL1:   LD B,0FFH
 LDL2:   NOP
         DJNZ LDL2
         DEC C
         CP C
         JR NZ,LDL1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          POP BC
          POP BC
          RET
;     * * * * * * * * * * * * * * * *
;     *     Set start calue          *
;     * * * * * * * * * * * * * * * *
  SSRT:  PUSH BC
         LD  B,STD          ; start line in dot
         LD  C,LDP01
         CALL OPDC1
         LD  C,LDP02
         CALL OPDC2
         LD  B,STL          ; start line
         LD  C,LP01
         CALL OPDC1
         LD  C,LP02
         CALL OPDC2
         LD  B,STC          ; start column
         RET
;     * * * * * * * * * * * * * * * * * *
;     * Check select and run sub'func *
;     * I/P = (SLM1)                   *
;     * * * * * * * * * * * * * * * * * *
 RUNCHOICE:LD  A,(SLM1)    ;get select'menu1->Reg. A
         CP  04H           ;compare four?
         JR  NC,M3         ;if select'menu<4,Call func'ORDER
         CALL ORDER
         LD  A,(SLM1)
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
;      M1:  CP  02H
;           JR  NZ,M2     ;if select'menu=2,Call func'TRIGGER
;           CALL ORDER
;           LD  A,(SLM1)
;      M2:  CP  03H
;           JR  NZ,M3      ;if select'menu=3,Call func'CANCEL
;           CALL CANCEL
;           LD  A,(SLM1)
;- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     M3:  CP  04H
          JR  NZ,M4        ;if select'menu=4,Call func'PRESENT
          CALL PRESENT
          LD  A,(SLM1)
     M4:  CP  05H
          JR  NZ,M5        ;if select'menu=5,Call func'PORTTOMEM
          CALL PORTTOMEM
          LD  A,(SLM1)
     M5:  CP  06H
          JR  NZ,M6        ;if select'menu=6,Call func'MEMTOPORT
          CALL MEMTOPORT
          LD  A,(SLM1)
     M6:  CP  07H
          JR  NZ,M7        ;if select'menu=7,Call func'SAFTTEST
          CALL SAFTTEST
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD   A,(SLM1)
       M7:  CP   08H
            JR   NZ,M8          ;if select'menu=8,Call func'EXIT_PRO
            CALL EXIT_PRO
            LD   A,(SLM1)
       M8:  RET
;      * * * * * * * * * * * * * * * *
;      * Call 1..Order code of food  *
;      * I/P = A(CODE OF COMMAND)    *
;      * O/P = Send data to computer *
;      * * * * * * * * * * * * * * * *
 ORDER:      CALL CLRVALUE     ;func clear all value
             CALL SAVECOMM     ;func save command code
             CALL SAVETAB      ;func input+save No.table value
 SAVE_F:     CALL SAVEFOOD     ;func input+save No.food code
             CALL SAVEACCT     ;func input+save account of food
             CALL CONFIRMF     ;func confirm food+account
             CALL EXIT_YN      ;func Your want to exit?(y/n)
             LD   A,(TMP3)     ;check return value
             CP   01H          ;bit check set ?
             JR   Z,SAVE_F     ;jump to func save food
             CALL SENDDATA     ;func send data to port
;            CALL LISTORDER    ;func list all order'food
             RET
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ;
;            CALL CLRVALUE     ;func clear all value                      ;
;            CALL SAVETAB      ;func input+save No.table value            ;
; LISTORDER:CALL SAVEFOOD      ;func input+save code'food value           ;
;            CALL SAVEACCT     ;func input+save account  value            ;
;            CALL EXIT_YN      ;func You want to exit?(y/n)               ;
;            CALL LISTORDER    ;func list all order'food                  ;
;            CALL SENDDATA     ;func send data to port                    ;
;            CALL DISPCOMPL    ;func display compete                      ;
;            LD   A,(SLM1)                                                ;
;            RET                                                          ;
; * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * ;
;      * * * * * * * * * * * * * *
;      * SEND DATA TO SERAIL PORT  *
;      * I/P = (DB_FOOD)           *
;      * O/P = PORT-E 5            *
;      * * * * * * * * * * * * * * *
 SENDDATA:   LD   A,00
             LD   (TMP1),A       ; reset variable
             LD   (TMP2),A       ; reset variable
             CALL DP_SEND
             CALL S_DATA
             RET
 S_DATA:     CALL SCANKY         ; scan key and get code
             CP   0CH            ; press to key #
             JR   NZ,DA_A        ; exit
             LD   A,01           ; set attribute
             LD   (TMP1),A       ; save attribute
             CALL SEND_PORT      ; func send data to serial port
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JR    SSSS          ; jump func ccc
DA_A:       CP    0AH           ; press to key *
            JR    NZ,D_TA       ; exit
            LD    A,01          ; set attribute
            LD    (TMP2),A      ; save attribute
            JR    SSSS          ; jump func ccc
D_TA:       JR    S_DATA        ; loopback
SSSS:       CALL  DP_SEND       ; restore display
            RET
SEND_PORT:
            RET
DP_SEND:    CALL  DISOFF
            CALL  CLRD
            LD    HL,MN1_S20        ;pointer of string
            LD    B,00H             ;line
            LD    C,00H             ;column
            LD    E,00H             ;attrib if 1=rev 0=normal
            CALL  STRTORAM
            LD    HL,MN1_S21        ;pointer of string
            LD    B,01H             ;line
            LD    C,02H             ;column
            LD    E,00H             ;attrib if 1=rev 0=normal
            CALL  STRTORAM
            CALL  STRTORAM
            LD    HL,MN1_S13        ;pointer of string
            LD    B,02H             ;line
            LD    C,00H             ;column
            LD    E,00H             ;attrib if 1=rev 0=normal
            CALL  STRTORAM
            LD    HL,MN1_S22        ;pointer of string
            LD    B,02H             ;line
            LD    C,03H             ;column
            LD    A,(TMP1)
            LD    E,A               ;attrib if 1=rev 0=normal
            CALL  STRTORAM
            LD    HL,MN1_S23        ;pointer of string
            LD    B,03H             ;line
            LD    C,03H             ;column
            LD    A,(TMP2)
            LD    E,A               ;attrib if 1=rev 0=normal
            CALL  STRTORAM
            CALL  RAMTODIS
            CALL  SHIFTLIN
            CALL  DISON
            CALL  LDELAY
            CALL  LDELAY
            RET
SAVECOMM:   LD    A,(SLM1)
            LD    (COM_V),A  ;save command code
            RET
SAVETAB:    CALL  DPTABLE     ;display intro word
            CALL  T_FROMKEY   ;input table from key
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SAVEFOOD:   CALL CLRINTMP
            CALL DPFOOD         ;display intro word
            CALL F_FROMKEY      ;input table from key
            RET
SAVEACCT:   CALL CLRINTMP
            CALL DPACCOUNT      ;display intro word
            CALL A_FROMKEY      ;input account from key
            RET
;     * * * * * * * * * * * * * * * * *
;     * CONFIRM FOOD AND ACCOUNT     *
;     * I/P = KEYBROAD (Yes or No)   *
;     * * * * * * * * * * * * * * * * *
CONFIRMF:   CALL CONFFOOD
            CALL FIRFL
            RET
FIRFL:      LD   A,00
            LD   (TMP1),A       ; reset variable
            LD   (TMP2),A       ; reset variable
            CALL SCANKY         ; scan key and get code
            CP   01H            ; press to key 1
            JR   NZ,C_ONFI      ; confirm
            LD   A,01           ; set attribute
            LD   (TMP1),A       ; save attribute
            CALL S_FRAME        ; func save frame type
            JR   CCCC           ; jump func ccc
C_ONFI:     CP   02H            ; press to key 2
            JR   NZ,C_ANCE      ; cancel
            LD   A,01           ; set attribute
            LD   (TMP2),A       ; save attribute
            JR   CCCC           ; jump func ccc
C_ANCE:     JR   FIRFL          ; loopback
CCCC:       CALL CONF
            RET
CONF:       CALL DISOFF
            CALL CLRD
            LD   HL,MN1_SUB12       ;pointer of string
            LD   B,01H              ;line
            LD   C,00H              ;column
            LD   E,00H              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            LD   HL,MN1_S13         ;pointer of string
            LD   B,02H              ;line
            LD   C,00H              ;column
            LD   E,00H              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            LD   HL,MN1_S14         ;pointer of string
            LD   B,02H              ;line
            LD   C,03H              ;column
            LD   A,(TMP1)
            LD   E,A                ;attrib if 1=rev 0=normal
            CALL STRTORAM
            LD   HL,MN1_S15         ;pointer of string
            LD   B,03H              ;line
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD    C,03H             ;column
                LD    A,(TMP2)
                LD    E,A               ;attrib if 1=rev 0=normal
                CALL  STRTORAM
                CALL  SEARCHFOOD        ;search name from code
                CALL  STRTORAM
                CALL  V_TORAM           ;Get TMP_V-memory to VDOBUF
                CALL  RAMTODIS
                CALL  SHIFTLIN
                CALL  DISON
                CALL  LDELAY
                CALL  LDELAY
                RET
CONFFOOD:       CALL  DISOFF
                CALL  CLRD
                LD    HL,MN1_SUB12      ;pointer of string
                LD    B,01H             ;line
                LD    C,00H             ;column
                LD    E,00H             ;attrib if 1=rev 0=normal
                CALL  STRTORAM
                LD    HL,MN1_S13        ;pointer of string
                LD    B,02H             ;line
                LD    C,00H             ;column
                LD    E,00H             ;attrib if 1=rev 0=normal
                CALL  STRTORAM
                LD    HL,MN1_S14        ;pointer of string
                LD    B,02H             ;line
                LD    C,03H             ;column
                LD    E,00H             ;attrib if 1=rev 0=normal
                CALL  STRTORAM
                LD    HL,MN1_S15        ;pointer of string
                LD    B,03H             ;line
                LD    C,03H             ;column
                LD    E,00H             ;attrib if 1=rev 0=normal
                CALL  STRTORAM
                CALL  SEARCHFOOD        ;search name from code
                CALL  STRTORAM
                CALL  V_TORAM           ;Get TMP_V-memory to VDOBUF
                CALL  RAMTODIS
                CALL  SHIFTLIN
                CALL  DISON
                RET
;     * * * * * * * * * * * * * * * *
;     * Your want to exit program?  *
;     *          (y/n)               *
;     * O/P = A <00=y,01=n>          *
;     * * * * * * * * * * * * * * * *
EXIT_YN:        LD    A,00
                LD    (TMP1),A      ; reset variable
                LD    (TMP2),A      ; reset variable
                LD    (TMP3),A      ; reset variable
                CALL  DP_EXIT
                CALL  T_YN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
T_YN:       CALL SCANKY         ; scan key and get code
            CP   01H            ; press to key 1
            JR   NZ,EX_T        ; exit
            LD   A,01           ; set attribute
            LD   (TMP1),A       ; save attribute
            LD   (TMP3),A       ; return value
            JR   TTTT           ; jump func ccc
EX_T:       CP   02H            ; press to key 2
            JR   NZ,E_IT        ; exit
            LD   A,01           ; set attribute
            LD   (TMP2),A       ; save attribute
            LD   IY,(DB_TMP)    ; load pointer
            LD   (IY+0),0DDH    ; end of database
            JR   TTTT           ; jump func ccc
E_IT:       JR   T_YN           ; loopback
TTTT:       CALL DP_EXIT        ; restore display
            RET
DP_EXIT:    CALL DISOFF
            CALL CLRD
            LD   HL,MN1_S16         ;pointer of string
            LD   B,00H              ;line
            LD   C,00H              ;column
            LD   E,00H              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            LD   HL,MN1_S17         ;pointer of string
            LD   B,01H              ;line
            LD   C,00H              ;column
            LD   E,00H              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            CALL STRTORAM
            LD   HL,MN1_S13         ;pointer of string
            LD   B,02H              ;line
            LD   C,00H              ;column
            LD   E,00H              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            LD   HL,MN1_S18         ;pointer of string
            LD   B,02H              ;line
            LD   C,03H              ;column
            LD   A,(TMP1)
            LD   E,A                ;attrib if 1=rev 0=normal
            CALL STRTORAM
            LD   HL,MN1_S19         ;pointer of string
            LD   B,03H              ;line
            LD   C,03H              ;column
            LD   A,(TMP2)
            LD   E,A                ;attrib if 1=rev 0=normal
            CALL STRTORAM
            CALL RAMTODIS
            CALL SHIFTLIN
            CALL DISON
            CALL LDELAY
            CALL LDELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
;     * * * * * * * * * * * * * * * * *
;     * SAVE FRAME OF DATABASE       *
;     * I/P = (TMP3)                 *
;     * O/P = DB_FOOD                *
;     * * * * * * * * * * * * * * * * *
  S_FRAME: LD  IY,(DB_TMP)
           LD  A,(COM_V)
           LD  (IY+0),A
           INC IY
           LD  A,(TAB_V)
           LD  (IY+0),A
           INC IY
           LD  A,(F_CODE2)
           LD  (IY+0),A
           INC IY
           LD  A,(F_CODE1)
           LD  (IY+0),A
           INC IY
           LD  A,(ACC_V)
           LD  (IY+0),A
           INC IY
           LD  (DB_TMP),IY
           RET
;     * * * * * * * * * * * * * * * * *
;     * SEARCH NAME OF FOOD          *
;     * I/P = F_CODE?                *
;     * O/P = HL -> STRING           *
;     * * * * * * * * * * * * * * * * *
 SEARCHFOOD:LD HL,FOODLST    ;set index of food
           LD A,(F_CODE2)    ;check group of food
 SEARCHF:  LD A,(HL)         ;load Reg.A <- dat.mem
           CP OFFH           ;compare by end-of-file(FFH)
           JR NZ,S_CHK1      ;if !=eof goto next state
           LD HL,NFOUND      ;display "not found"
           JR S_CHK5
 S_CHK1:   LD A,(F_CODE2)
           CP (HL)           ;compare and search group
           JR Z,S_CHK2       ;if ==grp goto next state
 S_C1:     INC HL            ;increate pointer
           LD A,(HL)         ;load Reg.A <- dat.mem
           CP OFFH           ;compare by end-of-frame(FFH)
           JR NZ,S_C1        ;if !=eoframe goto S_C1:
           INC HL
           JR SEARCHF
 S_CHK2:   INC HL
           LD A,(HL)         ;load Reg.A <- dat.mem
           CP OFFH           ;compare by end-of-file(FFH)
           JR NZ,S_CHK4      ;if !=eof goto next state
           LD HL,NFOUND      ;display "not found"
           JR S_CHK5
 S_CHK4:   LD  A,(F_CODE1)
           CP (HL)           ;compare and search code
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JR Z,S_CHK3       ;if ==code goto next state
S_C2:       INC HL            ;increate pointer
            LD A,(HL)         ;load Reg.A <- dat.mem
            CP ODH            ;compare by end-of-string(ODH)
            JR NZ,S_C2        ;if !=eostring goto S_C2:
            JR S_CHK2
S_CHK3:     INC HL            ;pointer of string
S_CHK5:     LD   B,OOH        ;line
            LD   C,OOH        ;column
            LD   E,OOH        ;attrib if 1=rev 0=normal
            RET
DPACCOUNT:  CALL DISOFF
            CALL CLRD
            LD   HL,MN1_SUB12       ;pointer of string
            LD   B,01H              ;line
            LD   C,OOH              ;column
            LD   E,OOH              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            CALL SEARCHFOOD         ;search name from code
            CALL STRTORAM
            CALL V_TORAM            ;Get TMP_V-memory to VDOBUF
            CALL RAMTODIS
            CALL SHIFTLIN
            CALL DISON
            RET
DPFOOD:     CALL DISOFF
            CALL CLRD
            LD   HL,MN1_SUB11       ;pointer of string
            LD   B,OOH              ;line
            LD   C,01H              ;column
            LD   E,OOH              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            CALL V_TORAM            ;Get TMP_V-memory to VDOBUF
            CALL RAMTODIS
            CALL SHIFTLIN
            CALL DISON
            RET
DPTABLE:    CALL DISOFF
            CALL CLRD
            LD   HL,MN1_SUB1        ;pointer of string
            LD   B,OOH              ;line
            LD   C,01H              ;column
            LD   E,OOH              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            CALL V_TORAM            ;Get TMP_V-memory to VDOBUF
            CALL RAMTODIS
            CALL SHIFTLIN
            CALL DISON
            RET
;     * * * * * * * * * * * * * * * *
;     * INPUT ACCOUNT FROM KEYBROAD *
;     * I/P = KEYBROAD              *
;     * O/P = ACC_V                 *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * * * * * * * * * * * * * * * *
  A_FROMKEY: LD    A,00H
             LD    (COUNT),A
  A_FROM:    CALL  SCANKY        ; call function scan key
             ADD   A,30H         ; convert heximal to ascii-code
             LD    B,A           ; save Reg.A -> Reg.B
             CP    3BH           ; if ascii==3Bh <key value>
             JR    NZ,A_KEY
             LD    A,30H         ; convert 3Bh to 30h
  A_KEY:     LD    (TEMPV),A     ; write ascii-code to memory
             LD    A,B
             SUB   30H           ; convert ascii-code to heximal
             CP    01H            ; press to No.1
             JR    Z,KEY_1_12
             CP    02H           ; press to No.2
             JR    Z,KEY_2_12
             CP    03H           ; press to No.3
             JR    Z,KEY_3_12
             CP    04H           ; press to No.4
             JR    Z,KEY_4_12
             CP    05H           ; press to No.5
             JR    Z,KEY_5_12
             CP    06H           ; press to No.6
             JR    Z,KEY_6_12
             CP    07H           ; press to No.7
             JR    Z,KEY_7_12
             CP    08H           ; press to No.8
             JR    Z,KEY_8_12
             CP    09H           ; press to No.9
             JR    Z,KEY_9_12
             CP    0CH           ; press to key'enter
             JR    Z,KEY_12_12
             CP    0BH           ; press to No.11
             JR    Z,KEY_11_12
;         CP    0AH             ; press to No.10
;         JR    Z,KEY_10_12
;         CP    09H             ; press to No.12
;         JR    Z,KEY_9_12
             JR    A_FROM
 KEY_1_12:
 KEY_2_12:
 KEY_3_12:
 KEY_4_12:
 KEY_5_12:
 KEY_6_12:
 KEY_7_12:
 KEY_8_12:
 KEY_9_12:
 KEY_11_12:  LD    A,(COUNT)
             CP    00H           ; compare which start data
             LD    IX,TMP_V      ; not shift data
             JR    Z,SUB_11_12
             LD    A,(IX+4)      ; shift data ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LD   (IX+6),A      ;    in      ;
                LD   A,(IX+2)      ;            ;
                LD   (IX+4),A      ; tempolary  ;
                LD   A,(IX+0)      ;    of      ;
                LD   (IX+2),A      ;video buffer;
 SUB_11_12:     LD   A,(TEMPV)
                LD   (IX+0),A      ; write data to tmp_buf (D1)
                LD   A,(COUNT)
                CP   00H
                JR   NZ,A_DISPLAY  ;if flag-start==0 Call Change flag
                INC  A             ;call function set flag
                LD   (COUNT),A
 A_DISPLAY:     CALL DPACCOUNT
                JR   A_FROM
 KEY_12_12:     CALL  DPACCOUNT    ;if press Enter-key
                CALL  WRTOBYTE     ;Call f'n write to variable BY?_V
                LD   A,(BY1_V)
                LD   (ACC_V),A
                RET
; KEY_10_12:
;     * * * * * * * * * * * * * * * * *
;     * INPUT CFOOD FROM KEYBROAD     *
;     * I/P = KEYBROAD                *
;     * O/P = FOO_V                   *
;     * * * * * * * * * * * * * * * * *
  F_FROMKEY: LD    A,00H
             LD    (COUNT),A
  F_FROM:    CALL  SCANKY       ; call function scan key
             ADD   A,30H        ; convert heximal to ascii-code
             LD    B,A          ; save Reg.A -> Reg.B
             CP    3BH          ; if ascii==3Bh <key value>
             JR    NZ,F_KEY
             LD    A,30H        ; convert 3Bh to 30h
  F_KEY:     LD    (TEMPV),A    ; write ascii-code to memory
             LD    A,B
             SUB   30H          ; convert ascii-code to heximal
             CP    01H           ; press to No.1
             JR    Z,KEY_1_11
             CP    02H          ; press to No.2
             JR    Z,KEY_2_11
             CP    03H          ; press to No.3
             JR    Z,KEY_3_11
             CP    04H          ; press to No.4
             JR    Z,KEY_4_11
             CP    05H          ; press to No.5
             JR    Z,KEY_5_11
             CP    06H          ; press to No.6
             JR    Z,KEY_6_11
             CP    07H          ; press to No.7
             JR    Z,KEY_7_11
             CP    08H          ; press to No.8
             JR    Z,KEY_8_11
             CP    09H          ; press to No.9
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR    Z,KEY_9_11
                CP    0CH               ; press to key'enter
                JR    Z,KEY_12_11
                CP    0BH               ; press to No.11
                JR    Z,KEY_11_11
;          CP    0AH                    ; press to No.10
;          JR    Z,KEY_10_11
;          CP    09H                    ; press to No.12
;          JR    Z,KEY_9_11
                JR    F_FROM
 KEY_1_11:
 KEY_2_11:
 KEY_3_11:
 KEY_4_11:
 KEY_5_11:
 KEY_6_11:
 KEY_7_11:
 KEY_8_11:
 KEY_9_11:
 KEY_11_11:   LD    A,(COUNT)
              CP    00H               ; compare which start data
              LD    IX,TMP_V          ; not shift data
              JR    Z,SUB_11_11
              LD    A,(IX+4)          ; shift data ;
              LD    (IX+6),A          ;    in      ;
              LD    A,(IX+2)          ;            ;
              LD    (IX+4),A          ; tempolary  ;
              LD    A,(IX+0)          ;    of      ;
              LD    (IX+2),A          ;video buffer;
 SUB_11_11:   LD    A,(TEMPV)
              LD    (IX+0),A          ; write data to tmp_buf (D1)
              LD    A,(COUNT)
              CP    00H
              JR    NZ,F_DISPLAY      ;if flag-start==0 Call Change flag
              INC   A                 ;call function set flag
              LD    (COUNT),A
 F_DISPLAY:   CALL  DPF00D
              JR    F_FROM
 KEY_12_11:   CALL   DPF00D           ;if press Enter-key
              CALL   WRTOF_C          ;Call f'n write to variable F_CODE?
              RET
; KEY_10_11:
;     * * * * * * * * * * * * * * * *
;     * INPUT TABLE FROM KEYBROAD    *
;     * I/P = KEYBROAD               *
;     * O/P = TAB_V                  *
;     * * * * * * * * * * * * * * * *
  T_FROMKEY: LD     A,00H
             LD     (COUNT),A
  T_FROM:    CALL   SCANKY           ; call function scan key
             ADD    A,30H            ; convert heximal to ascii-code
             LD     B,A              ; save Reg.A -> Reg.B
             CP     3BH              ; if ascii==3Bh <key value>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JR    NZ,T_KEY
                LD    A,30H         ; convert 3Bh to 30h
    T_KEY:      LD    (TEMPV),A     ; write ascii-code to memory
                LD    A,B
                SUB   30H           ; convert ascii-code to heximal
                CP   01H            ; press to No.1
                JR   Z,KEY_1_1
                CP   02H            ; press to No.2
                JR   Z,KEY_2_1
                CP   03H            ; press to No.3
                JR   Z,KEY_3_1
                CP   04H            ; press to No.4
                JR   Z,KEY_4_1
                CP   05H            ; press to No.5
                JR   Z,KEY_5_1
                CP   06H            ; press to No.6
                JR   Z,KEY_6_1
                CP   07H            ; press to No.7
                JR   Z,KEY_7_1
                CP   08H            ; press to No.8
                JR   Z,KEY_8_1
                CP   09H            ; press to No.9
                JR   Z,KEY_9_1
                CP   0CH            ; press to key'enter
                JR   Z,KEY_12_1
                CP   0BH            ; press to No.11
                JR   Z,KEY_11_1
;           CP   0AH                ; press to No.10
;           JR   Z,KEY_10_1
;           CP   09H                ; press to No.12
;           JR   Z,KEY_9_1
                JR   T_FROM
 KEY_1_1:
 KEY_2_1:
 KEY_3_1:
 KEY_4_1:
 KEY_5_1:
 KEY_6_1:
 KEY_7_1:
 KEY_8_1:
 KEY_9_1:
 KEY_11_1:    LD   A,(COUNT)
 T_DISPLAY:   CALL DPTABLE
              JR   T_FROM
 KEY_12_1:    CALL  DPTABLE       ;if press Enter-key
              CALL  WRTOBYTE      ;Call f'n write to variable BY?_V
              CALL  DTOH          ;Convert (BY1_V) to (TAB_V)<decimal>
              RET
; KEY_10_1:
;-----------------------------------;
;     * * * * * * * * * * * * * * * *
;     *     TMP_V to VDOBUF         *
;     * I/P = (TMP_V) ->ascii code  *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * O/P = VDOBUF                    *
;     * * * * * * * * * * * * * * * * *
 V_TORAM:    LD    A,03H              ; Counter=4
             LD    B,01H              ; line
             LD    C,09H              ; column
             LD    E,00H              ; attrib if 1=rev 0=normal
             LD    HL,TMP_V           ; pointer of string
  TORAM:     CALL  STRTORAM
             CP    00H
             JR    Z,V_RET
             INC   HL
             INC   HL
             DEC   C
             DEC   A
             JR    TORAM
  V_RET:     RET
;     * * * * * * * * * * * * * * * * *
;     *     Write to byte              *
;     *     I/P = (TMP_V)->ascii code  *
;     *     O/P = (BY1_V),(BY2_V)      *
;     * * * * * * * * * * * * * * * * *
  WRTOBYTE:  LD   IX,TMP_V
             LD   HL,BY1_V        ; set pointer ->BY1_V
             LD   A,(IX+0)
             RRBCD
             LD   A,(IX+2)
             RRBCD
             LD   HL,BY2_V        ; set pointer ->BY2_V
             LD   A,(IX+4)
             RRBCD
             LD   A,(IX+6)
             RRBCD
             RET
;     * * * * * * * * * * * * * * * * *
;     *  Write to food information   *
;     *  I/P = (TMP_V)->ascii code   *
;     *  O/P = (F_CODE1),(F_CODE2)   *
;     * * * * * * * * * * * * * * * * *
  WRTOF_C:   LD   IX,TMP_V
             LD   HL,F_CODE1        ; set pointer ->F_CODE1
             LD   A,(IX+0)
             RRBCD
             LD   A,(IX+2)
             RRBCD
             LD   HL,F_CODE2        ; set pointer ->F_CODE2
             LD   A,(IX+4)
             RRBCD
             LD   A,(IX+6)
             RRBCD
             RET
;     * * * * * * * * * * * * * * * * *
;     * Clear TMP_V  value            *
;     * I/P = (TMP_V) ->ascii code    *
;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * * * * * * * * * * * * * * * *
 CLRINTMP:  LD   A,30H
            LD   IX,TMP_V
            LD   (IX+0),A
            LD   (IX+2),A
            LD   (IX+4),A
            LD   (IX+6),A
            LD   A,0DH
            LD   IX,TMP_V
            LD   (IX+1),A
            LD   (IX+3),A
            LD   (IX+5),A
            LD   (IX+7),A
            RET
;     * * * * * * * * * * * * * * * *
;     *    Clear all value          *
;     *    I/P =                    *
;     * * * * * * * * * * * * * * * *
 CLRVALUE:  CALL CLRINTMP
            LD   A,00H
            LD   (COM_V),A
            LD   (TAB_V),A
            LD   (ACC_V),A
            LD   IX,PNT_V
            LD   (IX+0),A
            LD   (IX+1),A
            LD   IX,F00_V
            LD   (IX+0),A
            LD   (IX+1),A
            LD   IY,DB_F00D  ; startup database
            LD   (DB_TMP),IY ; save pointers
            RET
;     * * * * * * * * * * * * * * * *
;     * Call 4..Present  food       *
;     * I/P = A(CODE OF COMMAND)    *
;     * O/P = Send data to computer *
;     * * * * * * * * * * * * * * * *
 PRESENT:   CALL DISOFF
            CALL CLRD
            LD   HL,MN1_SUB4        ;poiter of string
            LD   B,00H              ;line
            LD   C,00H              ;column
            LD   E,00H              ;attrib if 1=rev 0=normal
            CALL STRTORAM
            CALL RAMTODIS
            CALL SHIFTLIN
            CALL DISON
            CALL LDELAY
            CALL LDELAY
            CALL LDELAY
            RET
 EXIT_PRO:  CALL DISOFF
            RST  18H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MN1_SUB1:   DFB   "กดเลขหน้าโต๊ะอาหาร",0DH
MN1_SUB11:  DFB   " กดโต๊ะอาหาร",0DH
MN1_SUB12:  DFB   "จำนวน        หน่วย",0DH
MN1_S13:    DFB   "กด",0DH
MN1_S14:    DFB   "1. ยืนยัน",0DH
MN1_S15:    DFB   "2. ยกเลิก",0DH
MN1_S16:    DFB   "มีรายการอาหาร",0DH
MN1_S17:    DFB   "เฟ้นมเต็มหรือไม่?",0DH
MN1_S18:    DFB   "1. เฟ้นมเต็ม",0DH
MN1_S19:    DFB   "2. เฟ้นยังพอ",0DH
MN1_S20:    DFB   "ข้อมูลพร้อมส่ง",0DH
MN1_S21:    DFB   "ออกพอร์ต",0DH
MN1_S22:    DFB   "#. ส่งข้อมูล",0DH
MN1_S23:    DFB   "*. ยกเลิก",0DH
MN1_SUB2:   DFB   "  HELLO MENU2",0DH
MN1_SUB3:   DFB   "  HELLO MENU3",0DH
MN1_SUB4:   DFB   "  HELLO MENU4",0DH
MN1_SUB5:   DFB   "  HELLO MENU5",0DH
MN1_SUB6:   DFB   "  HELLO MENU6",0DH
MN1_SUB7:   DFB   "  HELLO MENU7",0DH
MN1_SUB8:   DFB   "  HELLO MENU8",0DH
NFOUND:     DFB   "**ค้นหาข้อมูลไม่พบ**",0DH
;    * * * * * * * * * * * * * * *
PAGE3:  CALL DISOFF
        CALL CLRD
        LD   HL,LINE1
        LD   B,00H
        LD   C,00H
        LD   E,00H
        CALL STRTORAM
        LD   HL,LINE2
        LD   B,01H
        LD   C,00H
        LD   E,00H
        CALL STRTORAM
        LD   HL,LINE3
        LD   B,02H
        LD   C,00H
        LD   E,00H
        CALL STRTORAM
        LD   HL,LINE4
        LD   B,03H
        LD   C,00H
        LD   E,01H
        CALL STRTORAM
        LD   HL,LIN1
        LD   B,02H              ;test test
        LD   C,0CH
        LD   E,00H
        CALL STRTORAM
        LD   HL,LIN2
        LD   B,02H              ;test test
        LD   C,0EH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
         LD   E,00H
         CALL STRTORAM
         CALL RAMTODIS
         CALL SHIFTLIN
         CALL DISON
         RET
LINE1:    DFB   "กขคงจฉชซฌญฎฐฒณ",0DH
LINE2:    DFB   "ดตถทธนบปผฝพฟภมยร",0DH
LINE3:    DFB   "ลวศษสหฬอฮ ไอ",0DH
LINE4:    DFB   "ะัาิีึืุูเแโใไ่้๊๋",0DH
LINE9:    DFB   "กขคงจฉชซฌญฎฐฒณ",0DH
LINE10:   DFB   "ดตถทธนบปผฝพฟภมยร",0DH
LINE11:   DFB   "ลวษศสหฬอฮ",0DH
LINE12:   DFB   "ๆ.,;ยำไใโ-:ะแๆ",0DH
LIN1:     DFB   "ไอ",0DH
LIN2:     DFB   "ไอ",0DH
;     * * * * * * * * * * * * * * * * *
;     * * * *  MENU 1 * * * * * * * * *
;     *   I/P = ()                     *
;     *   O/P = HL                     *
;     *   USE = ABCDEHL                *
;     * * * * * * * * * * * * * * * * *
MENU1:   LD   A,01H
         LD   (SLMN),A
         LD   A,00H
         LD   (STMENU),A
         LD   A,09H
         LD   (CAMN),A
         CALL WRT_MN
         LD   A,01H          ; start menu,Only 1.
         LD   (STM1),A       ; set start'menu1 value
         LD   (SLM1),A       ; set select'menu1 value
         LD   A,09H          ; and menu,Only all'menu+1
         LD   (CAM1),A       ; set all'menu value(or cnt'menu)
MNLOOP:  CALL M_N1           ; Call function M_N1
         JR   MNLOOP
         RET
 M_N1:   CALL SCANKY         ; Scan keybroad
         CP   08H            ; press to key'up
         JR   Z,KEY_UP
         CP   0BH            ; press to key'down
         JR   Z,KEY_DW
         CP   0CH            ; press to key'enter
         JR   Z,KEY_EN
         JR   M_N1           ; Call function write'menu
KEY_UP:  LD   A,(SLM1)
         DEC  A              ; if press'up decrease select'menu1
         JR   CHK_SL         ; and call function check'select
KEY_DW:  LD   A,(SLM1)
         INC  A              ; if press'down increase select'menu1
         JR   CHK_SL         ; and call function check'select
;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
KEY_EN:  CALL SAVESCR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CALL RUNCHOICE      ; if press enter get select'menu1
                              ; and call function your'select
          CALL RESTSCR
          RET
;_ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ _
 CHK_SL:  LD   B,A
          LD   A,(CAM1)
          LD   D,A            ; Load cnt'menu1 ->Reg.D
          LD   A,B            ; Load select ->Reg.A
          CP   D              ; Comp'select and cnt'menu1
          JR   NZ,CHK_S1      ; if select!=cnt'menu1,Call CHK_S1
          CALL CHK_1          ; else ,Call CHK_1
 CHK_S1:  CP   00H            ; Comp'select and zero
          JR   NZ,CHK_S2      ; if select!=0,Call CHK_S2
          CALL CHK_2          ; else ,Call CHK_2
 CHK_S2:  LD   B,A            ; Load select ->Reg.B
          LD   A,(STM1)       ; Load start_disp ->Reg.A
          CP   B              ; Comp'select and start_disp
          JR   C,CHK_S3       ; if select>start_disp,Call
          CALL CHK_3          ; else,Call
 CHK_S3:  ADD  A,04H          ; OLD 03
          CP   B              ; Comp'select and start_disp+3
          JR   NC,CHK_S4      ; if select<start_disp+3,Call
          CALL CHK_4          ; else,Call
 CHK_S4:  LD   A,B            ; Load select ->Reg.A
          LD   (SLM1),A
          LD   A,(CAM1)       ; Load cnt'menu1 ->Reg.A
          LD   (CAMN),A       ; Load Reg.A ->cnt'menu
          LD   A,(STM1)       ; Load start'menu1 ->Reg.A
          LD   (STMENU),A     ; Load Reg.A ->start'menu
          LD   A,(SLM1)       ; Load select'menu1 ->Reg.A
          LD   (SLMN),A       ; Load Reg.A ->select'menu
          CALL WRT_MN         ; Call function write_menu
          RET
 CHK_1:   DEC  D              ; decrease select value
          LD   A,D
          LD   (SLM1),A       ; and Load to select'menu1
          RET
 CHK_2:   INC  A              ; if select=0 must increase
          LD   (SLM1),A       ; and Load to select'menu1
          RET
 CHK_3:   DEC  A              ; decrease select value
          LD   (STM1),A       ; and Load to select'menu1
          RET
 CHK_4:   SUB  04H            ; decrease end_disp  OLD 03
          INC  A              ; and start_disp+1
          LD   (STM1),A       ; and Load to start_menu1
          RET
;     * * * * * * * * * * * * * * * *
;     *    Scan Keybroad 16 key     *
;     *  I/P = STANDARD INPUT       *
;     *  O/P = A                    *
;     *  USE = ABCE                 *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * * * * * * * * * * * * * * * *
 SCANKY:LD   A,Wctrl
        OUT  (Pctrl),A       ;control port-d
        LD   A,Wstrt
        LD   B,A
 SCAN:  LD   A,B
        RLC  A               ;rotate start word
        LD   B,A
        AND  0F0h            ;mask bit
        OUT  (Pdout),A
        LD   A,Wctrl
        OUT  (Pctrl),A
        IN   A,(Pdout)
        AND  0Eh
        CP   0Eh
        JR   Z,SCAN
                             ; Check Row -> Reg.E
        BIT  4,B
        JR   Z,ROW1
        BIT  5,B
        JR   Z,ROW2
        BIT  6,B
        JR   Z,ROW3
        BIT  7,B
        JR   Z,ROW4
  ROW1: LD   E,00H
        JR   ENDD
  ROW2: LD   E,01H
        JR   ENDD
  ROW3: LD   E,02H
        JR   ENDD
  ROW4: LD   E,03H
        JR   ENDD
                             ;Check Column -> Reg.C
  ENDD: BIT  3,A
        JR   Z,COL1
        BIT  2,A
        JR   Z,COL2
        BIT  1,A
        JR   Z,COL3
        BIT  0,A
        JR   Z,COL4
  COL1: LD   C,01H
        JR   ENDE
  COL2: LD   C,02H
        JR   ENDE
  COL3: LD   C,03H
        JR   ENDE
  COL4: LD   C,04H
        JR   ENDE
  ENDE: LD   B,C
  ENDN: LD   A,E
        CP   00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JR   Z,ENDER
        DEC  E
        LD   A,B
        ADD  A,03H
        LD   B,A
        JR   ENDN
ENDER:  LD   A,B
        RET
;    * * * * * * * * * * * * * * * * *
;    *       Find Start menu         *
;    *   I/P = A[counter], HL[prt]   *
;    *   O/P = HL                    *
;    *   USE = ABCDEHL               *
;    * * * * * * * * * * * * * * * * *
STFIND: CP   00H
        JR   Z,ENDSTF          ;if -eq 0 goto end of start find
        LD   E,A
        LD   A,(HL)            ;load ascii code
        CP   0DH               ;compare end string code
        JR   Z,STF1
        INC  HL
        LD   A,E
        JR   STFIND
STF1:   INC  HL
        DEC  E
        LD   A,E
        JR   STFIND
ENDSTF: RET
;    * * * * * * * * * * * * * * * * *
;    * Convert Decimal to Hexacimal*
;    *   I/P = (BY1_V),(BY2_V)     *
;    *   O/P = (TAB_V)             *
;    *   USE = ABCDEHL             *
;    * * * * * * * * * * * * * * * * *
  DTOH: XOR A             ; Clear Reg. A
        LD   HL,TAB_V     ; Load address ->Reg.HL
        LD   (HL),A       ; Clear Reg.HL
        LD   A,(BY1_V)    ; Load data from BY1_V
        LD   B,A
        SRL B             ;  Shift for  ;
        SRL B             ;    get      ;
        SRL B             ;  MSB(BY1_V) ;
        SRL B             ;             ;
        RLBCD             ; Get LSB(BY1_V) to Reg.HL
DH_LOOP:LD   A,B
        CP   00H
        JR   Z,DH_RET     ;if msb==0 goto end
        LD   A,0AH        ;sum incread 10d
        ADD A,(HL)
        LD   (HL),A
        DEC B
        JR   DH_LOOP
DH_RET: RET               ;return (TAB_V)
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     * * * * * * * * * * * * * * * *
;     *       Shift line of Dot       *
;     * * * * * * * * * * * * * * * *
 SHIFTLIN: LD  B,0FBH
           LD  C,81H
           CALL OPDC1
           LD  C,80H
           CALL OPDC2
           RET
;     * * * * * * * * * * * * * * * *
;     *   Read Vdo.buf to Display   *
;     *   I/P = (VDOBUF)            *
;     *   O/P = []   USE = ABCDE,IX *
;     * * * * * * * * * * * * * * * *
 RAMTODIS:  LD  BC,00H
            LD  A,00H
            LD  (LINE),A
            LD  (COLU),A
     RAMT:  PUSH BC
            CALL CHKPOINT
            POP  BC
            LD  IX,(POINTER)
            LD  E,(IX+3)
            LD  D,(IX+0)
            CALL WRTCHAR
            LD  A,D
            CP  00H
            JR  Z,INCCOL
            LD  D,(IX+1)
            LD  A,D
            CP  00H
            JR  Z,INCCOL
            CALL WRTCHAR
            LD  D,(IX+2)
            LD  A,D
            CP  00H
            JR  Z,INCCOL
            CALL WRTCHAR
  INCCOL:   INC C
            LD  A,C
            LD  (COLU),A
            CP  10H
            JR  NZ,RAMT
            LD  C,00H
            LD  A,C
            LD  (COLU),A
            INC B
            LD  A,B
            LD  (LINE),A
            CP  04H
            JR  NZ,RAMT
            RET
;     * * * * * * * * * * * * * * * *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     *   Set Position of Character *
;     *   I/P = (TLINE),(TCOLU)     *
;     *   O/P = ()                  *
;     *   USE = ABC                 *
;     * * * * * * * * * * * * * * * *
 SETPOS: LD   A,(TLINE)
         LD   B,A
         LD   A,(LPO)
         LD   C,A
         CALL OPDC
         LD   A,(TCOLU)
         LD   B,A
         LD   A,(CPO)
         LD   C,A
         CALL OPDC
         LD   B,STD
         LD   A,(LDPO)
         LD   C,A
         CALL OPDC
         RET
;     * * * * * * * * * * * * * * * *
;     *          TEMP               *
;     *  Set Position of Character  *
;     *   I/P = (TMPL),(TMPC)       *
;     *   O/P = ()                  *
;     *   USE = ABC                 *
;     * * * * * * * * * * * * * * * *
 SETPOST: EXX
          LD   A,(TMPL)
          LD   B,A
          LD   A,(LPO)
          LD   C,A
          CALL OPDC
          LD   A,(TMPC)
          LD   B,A
          LD   A,(CPO)
          LD   C,A
          CALL OPDC
          LD   A,(LDPO)
          LD   C,A
          CALL OPDC
          EXX
          RET
;     * * * * * * * * * * * * * * * *
;     *  Write Character to Display *
;     *   I/P = (ASCIIC)            *
;     *   O/P = ()                  *
;     *   USE = ABCDEHL             *
;     * * * * * * * * * * * * * * * *
 WRTCHR:  LD  A,(TYPCHR)
          CP  00H
          JR  Z,CHRTO
          CP  01H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          JR   Z,CHRT1
          CP   02H
          JR   Z,CHRT2
          CP   03H
          JR   Z,CHRT3
          CP   04H
          JR   Z,CHRT4
          CP   05H
          JR   Z,CHRT2
CHRT0:    LD   A,(TLINE)          ; write 'space'
          LD   B,A
          LD   A,(TCOLU)
          LD   D,A
          LD   A,B
          LD   (TMPL),A
          LD   A,D
          LD   (TMPC),A
          CALL SETPOST
          CALL SETFLAG1
          LD   A,01H              ; '**********'
          LD   (ASCIIC),A
          CALL WRITE
          RET
CHRT1:    LD   A,(TLINE)          ; write 'center'
          LD   B,A
          LD   A,(TCOLU)
          LD   D,A
          LD   A,B
          LD   (TMPL),A
          LD   A,D
          LD   (TMPC),A
          CALL SETPOST
          CALL SETFLAG1
          CALL WRITE
          RET
CHRT2:    LD   A,(TLINE)          ; write 'over'
          LD   B,A
          LD   A,(TCOLU)
          LD   D,A
          CALL DECLINE
          LD   A,B
          LD   (TMPL),A
          LD   A,D
          LD   (TMPC),A
          CALL SETPOST
          CALL SETFLAG0
          CALL WRITE
          RET
CHRT3:    LD   A,(TLINE)          ; write 'under'
          LD   B,A
          LD   A,(TCOLU)
          LD   D,A
          CALL INCLINE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD   A,B
            LD   (TMPL),A
            LD   A,D
            LD   (TMPC),A
            CALL SETPOST
            CALL SETFLAGO
            CALL WRITE
            RET
--  CHRT4:  CALL CHRT2              ; write 'over' & 'center'
            LD   A,0EEH
            LD   (ASCIIC),A
            CALL CHRT1
            RET
;  CHRT5:   CALL CHRT2              ; write 'over' & 'center'
;           LD   A,(TCOLU)
;           INC  A
;           LD   (TCOLU),A
;           LD   A,0D2H                 ; '++++++++++'
;           LD   (ASCIIC),A
;           CALL CHRT1
;           RET
;     * * * * * * * * * * * * * * * * *
;     *  Get data from Ascii-Table  *
;     *   I/P = (ASCIIC)            *
;     *   O/P = ()                  *
;     *   USE = ABCDEHL             *
;     * * * * * * * * * * * * * * * * *
   WRITE:   LD   A,(ASCIIC)         ; load ascii code
            LD   C,A
            LD   E,8                ; A*8 -> A
            CALL MUL                ; call multiplier function
            LD   BC,TABLE           ; pointer 'HL' by TABLE
            ADD  HL,BC              ; sum bett. ascii+table ->'HL'
            LD   E,08H              ; counter eight
   RESET:   RES  0,A
    LDAB:   LD   B,A
    WRC2:   LD   A,B
            CALL SETPOST            ; set position
;           LD   B,A
            LD   A,(WP0)
            LD   C,A
            CALL OPDC               ; out data to port
            DEC  E                  ; decreat counter
            INC  HL                 ; increat pointer
            LD   A,(TMPC)           ; increat column of data
            INC  A                  ;     .     .     .
            LD   (TMPC),A           ;     .     .     .
            LD   A,E
            CP   00H
            JR   NZ,WRC1            ; check end of data
            RET
;     * * * * * * * * * * * * * * * * *
;     *  Multoplier  Program        *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;     *    I/P = C(INT1),E(INT2)      *
;     *    O/P = HL                   *
;     *    USE = ABCDEHL              *
;     * * * * * * * * * * * * * * * *
   MUL:    LD    B,08H            ; counter =8
           LD    D,00H            ; clear d REG.
           LD    HL,00H           ; clear result
   MULT:   SRL   C                ; shift right one bit
           JR    NC,NOADD         ; check zero bit and jump
           ADD   HL,DE            ; result =result + constva
;     * * * * * * * * * * * * * * * *
;     *    Decreat Line              *
;     * * * * * * * * * * * * * * * *
  DECLINE: LD   A,B
           CP   STL
           JR   Z,DLIN
           DEC B
           RET
  DLIN:    ADD A,07H
           LD   B,A
           RET
;     * * * * * * * * * * * * * * * *
;     *    Increat Line              *
;     * * * * * * * * * * * * * * * *
  INCLINE: LD   A,B
           CP   STL+07H
           JR   Z,ILIN
           INC B
           RET
  ILIN:    LD B,STL
           RET
;     * * * * * * * * * * * * * * * *
;     *    Check flag of Attribute   *
;     *         Character            *
;     *    I/P = (REV)               *
;     *    O/P = (FREV) USE = A      *
;     * * * * * * * * * * * * * * * *
 SETFLAG1: LD   A,(ATTB)
           CP   01H
           JR   Z,FLG1
 SETFLAG0: LD   A,00H
           LD   (FREV),A
           RET
   FLG1:   LD   A,01H
           LD   (FREV),A
           RET
;     * * * * * * * * * * * * * * * *
;     *  Out data and Check busy     *
;     *    I/P = B(DATA),C(PORT)     *
;     *    O/P = ()    USE = ABC     *
;     * * * * * * * * * * * * * * * *
 OPDC:    LD    A,(SRP0)
          LD    D,C
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LD   C,A
          IN   A,(C)
          RLC  A
          JR   C,OPDC
          LD   A,B
          LD   C,D
          OUT  (C),A
          RET
;     * * * * * * * * * * * * * * * * *
;     *  Input data and Check busy  *
;     *   I/P = ()                  *
;     *   O/P = A    USE = ABC      *
;     * * * * * * * * * * * * * * * * *
 IPDC:    LD   A,(SRP0)
          LD   C,A
          IN   A,(C)
          RLC  A
          JR   C,IPDC
          LD   A,(RP0)
          LD   C,A
          IN   A,(C)
          RET
;     * * * * * * * * * * * * * * * * *
;     *   Out data and Check busy   *
;     *   I/P = B(DATA),C(PORT)     *
;     *   O/P = ()   USE = ABC      *
;     * * * * * * * * * * * * * * * * *
 OPDC1:   PUSH AF
          PUSH BC
          PUSH DE
 OPC1:    IN   A,(SRP01)
          RLC  A
          JR   C,OPC1
          LD   A,B
          OUT  (C),A
          POP  DE
          POP  BC
          POP  AF
          RET
;     * * * * * * * * * * * * * * * * *
;     *   Out data and Check busy   *
;     *        for display 2        *
;     *   I/P = B(DATA),C(PORT)     *
;     *   O/P = ()   USE = ABC      *
;     * * * * * * * * * * * * * * * * *
 OPDC2:   PUSH AF
          PUSH BC
          PUSH DE
 OPC2:    IN   A,(SRP02)
          RLC  A
          JR   C,OPC2
          LD   A,B
          OUT  (C),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          POP  DE
          POP  BC
          POP  AF
          RET
;     * * * * * * * * * * * * * * * *
;     *       Check Ascii Code        *
;     * I/P = (ASCIIC)                *
;     * O/P = (TYPCHAR) USE = ABC     *
;     * * * * * * * * * * * * * * * *
CHKASCII:  LD  A,(ASCIIC)
           CP  00H
           JR  Z,TYPE1
           CP  22H
           JR  C,TYPE0
           CP  0D1H
           JR  C,TYPE1
           CP  0D1H
           JR  Z,TYPE2
           CP  0D2H
           JR  Z,TYPE1
           CP  0D3H
           JR  Z,TYPE5
           CP  0D8H
           JR  C,TYPE2
           CP  0DAH
           JR  C,TYPE3
           CP  0E2H
           JR  C,TYPE1
           CP  0E5H
           JR  C,TYPE4
           CP  0E7H
           JR  C,TYPE1
           CP  0EEH
           JR  C,TYPE2
           CP  0FFH
           JR  C,TYPE1
           JR  Z,TYPE1
TYPE0:     LD  A,00
           LD  (TYPCHR),A
           RET
TYPE1:     LD  A,01
           LD  (TYPCHR),A
           RET
TYPE2:     LD  A,02
           LD  (TYPCHR),A
           RET
TYPE3:     LD  A,03
           LD  (TYPCHR),A
           RET
TYPE4:     LD  A,04
           LD  (TYPCHR),A
           RET
TYPE5:     LD  A,05
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD  (TYPCHR),A
            RET
;     * * * * * * * * * * * * * * * *
;     *       Check Line              *
;     * I/P = (LINE)   USE = A       *
;     * O/P = TRUE LINE BUFFER       *
;     * * * * * * * * * * * * * * * *
 CHKLIN:    LD  A,(LINE)
            SLA A               ; A*2 -> A
            LD  B,STL           ; load start line
            ADD A,B
            LD  (TLINE),A       ; stay in ture line
            RET
;     * * * * * * * * * * * * * * * *
;     *       Check Line              *
;     * I/P = (COLUMN)   USE = A     *
;     * O/P = Port buf.and Col.buf. *
;     * * * * * * * * * * * * * * * *
 CHKCOL:    LD  A,(COLU)
            CP  08H
            JR  C,MEMPORT1
 MEMPORT2:  SUB 08H
            SLA A
            SLA A
            SLA A
            LD  B,STC           ; load start column
            ADD A,B
            LD  (TCOLU),A       ; load ture column
            LD  A,WP02          ; write port
            LD  (WP0),A
            LD  A,SRP02         ; status read port
            LD  (SRP0),A
            LD  A,LP02          ; line port
            LD  (LP0),A
            LD  A,CP02          ; column port
            LD  (CP0),A
            LD  A,RP02          ; read port
            LD  (RP0),A
            LD  A,LDP02         ; line dot port
            LD  (LDP0),A
           RET
;     * * * * * * * * * * * * * * * *
;     * Write String to video buffer*
;     * I/P = B(LINE),C(COLUMN)     *
;     *        E(ATTRIB),HL(PRTDATA) *
;     * O/P =(LONGSTR) USE =ABCDEHL *
;     * * * * * * * * * * * * * * * *
STRTORAM:  PUSH AF
           PUSH BC
           PUSH DE
           PUSH HL
           LD  A,00
           LD  (CNT),A          ; set start counter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LD   (LONGSTR),A
            LD   A,B
            LD   (LINE),A          ; load line
            LD   A,C
            LD   (COLU),A          ; load column
            LD   A,E
            LD   (ATTB),A          ; load attribute
            CALL CHKPOINT
            LD    IX,(POINTER)     ; load pointer of vdo. buffer
RDASCII:    LD    A,(HL)
            CP    0DH              ; compare CR ?
            LD    (ASCIIC),A       ; load ascii code to memory
            JR    Z,ROOTOUT        ; if ascii same as CR jump to
            LD    A,(LONGSTR)
            INC   A
            LD    (LONGSTR),A
            LD    A,(ATTB)         ; set attribute of position
            LD    (IX+3),A
            CALL CHKASCII          ; call check ascii type
            CALL SMEMO             ; call write ascii to memory
            INC   HL
            LD    A,(CNT)
            INC   A
            LD    (CNT),A
            CP    03H
            JR    NZ,RDASCII
            CALL INCCOLU
            JP    RDASCII
; + + + + + + + + + + + + + + + + + + + +
 ROOTOUT:    LD   A,(CNT)
             CP   00H
             JR   Z,ENDPRG
             LD   A,(ATTB)
             LD   (IX+3),A
  ENDPRG:    POP HL
             POP DE
             POP BC
             POP AF
             RET
SMEMO:      LD   A,(TYPCHR)
            CP   00H
            JR   Z,SCHRT0
            CP   01H
            JR   Z,SCHRT1
            CP   02H
            JR   Z,SCHRT2
            CP   03H
            JR   Z,SCHRT3
            CP   04H
            JR   Z,SCHRT4
            CP   05H
            JR   Z,SCHRT5
 SCHRT0:    CALL INCCOLU
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LD  A,00H
          LD  (IX+0),A
          LD  (IX+1),A
          LD  (IX+2),A
          RET
SCHRT1:   LD  A,(CNT)
          CP  00H
          JR  Z,TD10
          CP  01H
          JR  Z,TD11
          CP  02H
          JR  Z,TD12
          RET
SCHRT2:   CALL ASCTOMEM
          RET
SCHRT3:   CALL ASCTOMEM
          RET
SCHRT4:   LD  A,(CNT)
          CP  00H
          JR  Z,TD40
          CP  01H
          JR  Z,TD41
          CP  02H
          JR  Z,TD42
          RET
SCHRT5:   CALL ASCTOMEM
          LD  A,01H
          LD  (CHKCHK),A
          RET
 CHECKER: LD  A,(CHKCHK)
          CP  01H
          JR  Z,CHECKT5
          RET
 CHECKT5: LD  A,0D2H
          LD  (IX+0),A
          LD  A,00H
          LD  (IX+1),A
          LD  (IX+2),A
          LD  (CHKCHK),A
          LD  A,(ATTB)
          LD  (IX+3),A
          CALL INCCOLU
          RET
;     * * * * * * * * * * * * * * * * *
;     *    Increat Column             *
;     *    I/P = A,(COLU),(POINTER)   *
;     *    O/P = ()  USE = A,IX       *
;     * * * * * * * * * * * * * * * * *
 INCCOLU: LD  A,(COLU)
          INC A
          LD  (COLU),A
          LD  A,00H
          LD  (CNT),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL CHKPOINT
            LD   IX,(POINTER)
            CALL CHECKER
            RET
;     * * * * * * * * * * * * * * * * *
;     *    Save Screen                 *
;     *    I/P = (VDOBUF),SP           *
;     *    O/P = (TMPBUF),TMPSP        *
;     * * * * * * * * * * * * * * * * *
  SAVESCR: EXX
            LD  BC,00FFH
            LD  DE,TMPBUF
            LD  HL,VDOBUF
            LDIR
            RET
;     * * * * * * * * * * * * * * * * *
;     *    Restore Screen              *
;     *    I/P = (TMPBUF),TMPSP        *
;     *    O/P = (VDOBUF),SP           *
;     * * * * * * * * * * * * * * * * *
  RESTSCR: CALL CLRD              ; clear video buffer
            LD  BC,00FFH          ; transfer tmp_buffer
            LD  DE,VDOBUF         ; to video buffer
            LD  HL,TMPBUF
            LDIR
            CALL DISOFF           ; start
            CALL RAMTODIS
            CALL SHIFTLIN
            CALL DISON
            EXX
            RET
;     * * * * * * * * * * * * * * * * *
;     *  Write Ascii to video buf.    *
;     *  I/P = IX,(CNT),(ASCIIC)      *
;     *  O/P = ()  USE = AB,IX        *
;     * * * * * * * * * * * * * * * * *
 ASCTOMEM: LD  A,(ASCIIC)
            LD  B,A
            LD  A,(CNT)
            CP  00H
            JR  Z,CT0
            CP  01H
            JR  Z,CT1
            CP  02H
            JR  Z,CT2
  CT0:      LD  (IX+0),B
            RET
  CT1:      LD  (IX+1),B
            RET
  CT2:      LD  (IX+2),B
            RET
;     * * * * * * * * * * * * * * * * *
;     *    ASCII TABLE OF PROGRAM     *
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;      *      THAI CHARACTERISTIC      *
;      * * * * * * * * * * * * * * * * *
  TABLE: DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 00D
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 10D
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 20D
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 30D
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,00H,38H,0BCH,38H,00H   ; ASCII 33D
         DFB     00H,00H,00H,00H,0CH,00H,0CH,00H
         DFB     00H,00H,28H,7CH,28H,7CH,28H,00H
         DFB     00H,00H,48H,54H,0D6H,54H,24H,00H
         DFB     00H,00H,80H,48H,20H,10H,48H,04H
         DFB     00H,60H,94H,8AH,94H,60H,50H,80H
         DFB     00H,00H,00H,00H,00H,00H,00H,00H
         DFB     00H,00H,00H,38H,44H,82H,00H,00H    ; ASCII 40D
         DFB     00H,00H,00H,82H,44H,38H,00H,00H
         DFB     00H,00H,54H,38H,7CH,38H,54H,00H
         DFB     00H,00H,10H,10H,7CH,10H,10H,00H
         DFB     00H,00H,00H,00H,0A0H,60H,00H,00H
         DFB     00H,00H,10H,10H,10H,10H,10H,00H
         DFB     00H,00H,00H,00H,60H,60H,00H,00H
         DFB     00H,00H,80H,40H,20H,10H,08H,04H
         DFB     00H,00H,7CH,0A2H,92H,8AH,7CH,00H   ; ASCII 48D
         DFB     00H,00H,84H,0FEH,80H,00H,00H,00H
         DFB     00H,00H,84H,0C2H,0A2H,92H,8CH,00H  ; ASCII 50D
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DFB     00H,00H,42H,82H,8AH,96H,62H,00H
DFB     00H,00H,30H,28H,24H,0FEH,20H,00H
DFB     00H,00H,4EH,8AH,8AH,8AH,72H,00H
DFB     00H,00H,78H,94H,92H,92H,62H,00H
DFB     00H,00H,02H,0E2H,12H,0AH,06H,00H
DFB     00H,00H,6CH,92H,92H,92H,6CH,00H
DFB     00H,00H,0CH,92H,92H,52H,3CH,00H    ; ASCII 57D
DFB     00H,00H,00H,00H,6CH,6CH,00H,00H
DFB     00H,00H,00H,00H,4CH,6CH,00H,00H
DFB     00H,00H,10H,28H,44H,82H,00H,00H    ; ASCII 60D
DFB     00H,00H,50H,50H,50H,50H,50H,00H
DFB     00H,00H,82H,44H,28H,10H,00H,00H
DFB     00H,00H,04H,0AH,0A2H,12H,0AH,04H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,0FCH,22H,22H,22H,0FCH,00H ; ASCII 65D
DFB     00H,00H,0FEH,92H,92H,92H,6CH,00H
DFB     00H,00H,7CH,82H,82H,82H,44H,00H
DFB     00H,00H,0FEH,82H,82H,44H,38H,00H
DFB     00H,00H,0FEH,92H,92H,92H,82H,00H
DFB     00H,00H,0FEH,12H,12H,12H,02H,00H ; ASCII 70D
DFB     00H,00H,0FEH,82H,92H,92H,0F4H,00H
DFB     00H,00H,0FEH,10H,10H,10H,0FEH,00H
DFB     00H,00H,00H,82H,0FEH,82H,00H,00H
DFB     00H,00H,40H,80H,82H,7EH,02H,00H
DFB     00H,00H,0FEH,10H,28H,44H,82H,00H
DFB     00H,00H,0FEH,80H,80H,80H,80H,00H
DFB     00H,00H,0FEH,04H,08H,04H,0FEH,00H
DFB     00H,00H,0FEH,08H,10H,20H,0FEH,00H
DFB     00H,00H,7CH,82H,82H,82H,7CH,00H
DFB     00H,00H,0FEH,12H,12H,12H,0CH,00H   ; ASCII 80D
DFB     00H,00H,7CH,82H,0A2H,42H,0BCH,00H
DFB     00H,00H,0FEH,12H,32H,52H,8CH,00H
DFB     00H,00H,8CH,92H,92H,62H,00H,00H
DFB     00H,00H,02H,02H,0FEH,02H,02H,00H
DFB     00H,00H,7EH,80H,80H,80H,7EH,00H
DFB     00H,00H,3EH,40H,80H,40H,3EH,00H
DFB     00H,00H,7EH,80H,70H,80H,7EH,00H
DFB     00H,00H,0C6H,28H,10H,28H,0C6H,00H
DFB     00H,00H,06H,08H,0F0H,08H,06H,00H
DFB     00H,00H,0C2H,0A2H,92H,8AH,86H,00H  ; ASCII 90D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,04H,08H,10H,20H,40H,80H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H     ; ASCII 100D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
```

เอกสารนี้เป็นเอกสารสงวนไว้ [illegible] ญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 110D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 120D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,0EEH,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 130D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 140D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 150D
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 160D
DFB     00H,08H,0F4H,04H,04H,04H,04H,0F8H
DFB     00H,00H,00H,00H,04H,0FCH,80H,0FCH
DFB     00H,00H,00H,0CH,08H,0FCH,80H,0FCH
DFB     00H,0F8H,44H,24H,24H,24H,04H,0F8H
DFB     00H,0F8H,44H,24H,28H,24H,04H,0F8H
DFB     00H,00H,0CH,0E8H,0FCH,40H,80H,0FCH
DFB     00H,00H,08H,10H,20H,40H,84H,0FCH
DFB     00H,08H,04H,24H,64H,84H,84H,0F8H
DFB     00H,08H,24H,0E4H,84H,44H,0F8H,0C0H
DFB     00H,00H,04H,7CH,80H,80H,84H,7AH   ; ASCII 170D
DFB     00H,0CH,08H,7CH,80H,80H,84H,7AH
DFB     00H,0FCH,84H,04H,44H,0F8H,40H,0FCH
DFB     00H,0FCH,84H,04H,38H,0A0H,0A0H,0BCH
DFB     00H,00H,20H,3CH,04H,0C4H,0F8H,40H
DFB     00H,00H,00H,20H,0BCH,84H,44H,0F8H
DFB     00H,00H,04H,04H,84H,94H,56H,0FCH
DFB     00H,0CH,08H,0FCH,10H,08H,04H,0FCH
DFB     00H,78H,84H,0A8H,44H,0F8H,40H,0FCH
DFB     00H,0FCH,84H,04H,0F8H,40H,0FCH,40H
DFB     00H,0F8H,84H,0A4H,64H,04H,04H,0F8H  ; ASCII 180D
DFB     00H,0F8H,84H,0A4H,68H,04H,04H,0F8H
DFB     00H,08H,0F4H,84H,04H,04H,04H,0F8H
DFB     00H,04H,0FCH,20H,10H,08H,04H,0FCH
DFB     00H,58H,94H,94H,94H,94H,94H,64H
DFB     00H,04H,0FCH,40H,20H,20H,0FCH,0E0H
DFB     00H,04H,0FCH,80H,80H,80H,80H,0FCH
DFB     00H,04H,0FCH,80H,80H,80H,80H,0FEH
DFB     00H,7CH,84H,80H,60H,80H,80H,7CH
DFB     00H,7CH,84H,80H,60H,80H,80H,7EH
DFB     00H,04H,0FCH,40H,20H,20H,40H,0FCH  ; ASCII 190D
DFB     00H,04H,0FCH,40H,20H,20H,40H,0FEH
DFB     00H,88H,0F4H,04H,04H,04H,04H,0F8H
DFB     00H,0E4H,0FCH,20H,20H,40H,80H,0FCH
DFB     00H,6CH,94H,90H,80H,80H,80H,7CH
DFB     00H,18H,14H,14H,14H,14H,94H,0E4H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,08H,0E4H,94H,14H,14H,24H,0F8H
DFB     00H,00H,00H,00H,00H,00H,00H,00H
DFB     00H,08H,04H,04H,04H,04H,84H,0F8H
DFB     00H,0F8H,44H,24H,24H,24H,04H,0FAH  ; ASCII 200D
DFB     00H,04H,0FCH,80H,0B0H,0A0H,0FCH,20H; ASCII 201D
DFB     00H,08H,0E4H,94H,14H,14H,24H,0FAH  ; ASCII 202D
DFB     00H,04H,0FCH,20H,10H,08H,0FCH,0CH
DFB     00H,08H,0F8H,40H,20H,24H,44H,0FEH    ;ASCII 204D
DFB     00H,08H,0E4H,0A4H,84H,84H,84H,0F8H   ;ASCII 205D
DFB     00H,00H,0E0H,0A4H,084H,84H,84H,0FAH  ;ASCII 206D
DFB     00H,08H,14H,1CH,1CH,10H,08H,0F8H
DFB     00H,00H,6CH,6CH,48H,48H,24H,00H
DFB     00H,00H,0C0H,0C0H,80H,80H,40H,00H
```

เอกสารนี้เป็นเอกสารสงวนไว้ [illegible] ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DFB     00H,00H,08H,04H,04H,04H,04H,0F8H    ; ASCII 210D
        DFB     00H,00H,00H,00H,0C0H,0C0H,00H,00H
        DFB     00H,00H,80H,0C0H,0C0H,0C0H,80H,00H ; ASCII 212D
        DFB     00H,00H,80H,0C0H,0C0H,0C0H,80H,0C0H
        DFB     00H,80H,0C0H,0C0H,0C0H,80H,0C0H,0C0H
        DFB     00H,80H,0C0H,0C0H,80H,0C0H,80H,0C0H
        DFB     00H,00H,00H,02H,06H,00H,00H,00H
        DFB     00H,00H,02H,06H,04H,06H,00H,00H
        DFB     00H,00H,00H,0C0H,0C0H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H   ; ASCII 220D
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 222D
        DFB     00H,00H,7CH,54H,0FEH,54H,28H,00H
        DFB     00H,00H,00H,0FCH,0C0H,0C0H,00H,00H
        DFB     00H,0FCH,0C0H,0C0H,00H,0FCH,0C0H,0C0H
        DFB     00H,80H,0C0H,0A0H,0A0H,20H,20H,20H
        DFB     00H,40H,0A0H,0A0H,20H,0C0H,00H,00H
        DFB     00H,20H,40H,80H,40H,0E0H,00H,00H
        DFB     00H,00H,08H,04H,04H,04H,04H,0F8H
        DFB     00H,38H,34H,08H,10H,08H,04H,0F8H    ; ASCII 230D
        DFB     00H,00H,00H,40H,0A0H,60H,0A0H,0FDH
        DFB     00H,00H,00H,00H,00H,00H,18H,00H    ; ASCII 232D
        DFB     00H,00H,08H,18H,10H,10H,08H,04H
        DFB     00H,10H,08H,10H,08H,10H,08H,04H
        DFB     00H,00H,00H,00H,00H,10H,38H,10H
        DFB     00H,00H,00H,00H,00H,00H,38H,34H
        DFB     00H,00H,00H,00H,0C0H,0C0H,00H,00H
        DFB     00H,00H,00H,00H,00H,0FFH,0C0H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H     ; ASCII 240D
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 242D
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 250D
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 252D
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H
        DFB     00H,00H,00H,00H,00H,00H,00H,00H    ; ASCII 255D

;           ORG  D000H
;     * * * * * * * * * * * * * * *
;     *   VARIABLE OF PROGRAM     *
;     *         RAM ARRAY         *
;     * * * * * * * * * * * * * * *
      COLU:  DFS  1       ; column in thai display
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LINE:     DFS   1          ; line in thai display
TCOLU:    DFS   1          ; ture column after convert
TLINE:    DFS   1          ; ture line after convert
TMPC:     DFS   1          ; temp column
TMPL:     DFS   1          ; temp line
LPO:      DFS   1          ; common line port
CPO:      DFS   1          ; common column port
LDPO:     DFS   1          ; common line dot port
SRPO:     DFS   1          ;   ..   status read port
WPO:      DFS   1          ;   ..   write port
RPO:      DFS   1          ;   ..   read port
ASCIIC:   DFS   1          ; acsii code buffer
TYPCHR:   DFS   1          ; type of character
ATTB:     DFS   1          ; common attribute
FREV:     DFS   1          ; flag reverse
CNT:      DFS   1          ; counter size of column
CHKCHK:   DFS   1          ; check of code 'D3'
LONGSTR:  DFS   1          ; long of string to ram
STACK:    DFS   100        ; array of stack
VDOBUF:   DFS   256        ; video buffer
TMPBUF:   DFS   256        ; tempolary video buffer
TMPSP:    DFS   1          ; tempolary stack
POINTER:  DFS   1          ; pointer of video buffer
;     * * * * * * * * * * * * * * * * *
;     *    Variable for M1 menu      *
;     * * * * * * * * * * * * * * * * *
STMN:     DFS   1          ; value start of menu
SLMN:     DFS   1          ; value select of menu
CAMN:     DFS   1          ; value counter all of menu
CNMN:     DFS   1          ; value counter temp of menu
VAR1:     DFS   2
VAR2:     DFS   2
TMP1:     DFS   1          ; value tempolary of menu 1
TMP2:     DFS   1          ; value tempolary of menu 2
TMP3:     DFS   1          ; value tempolary of menu 3
STM1:     DFS   1          ; value start of menu 1
SLM1:     DFS   1          ; value select of menu 1
CAM1:     DFS   1          ; value counter all of menu 1
STMENU:   DFS   1          ; value start of menu
DB_TMP:   DFS   2          ; pointer of Database
DB_FOOD:  DFS   100        ; Database which want to send
;     * * * * * * * * * * * * * * * * *
;     *    Variable for format data *
;     * * * * * * * * * * * * * * * * *
COM_V:    DFS   1          ; value code of command
TAB_V:    DFS   1          ; value No.table
FOO_V:    DFS   2          ; value code of food
PNT_V:    DFS   2          ; value pointer of data
ACC_V:    DFS   1          ; value account of food
TMP_V:    DFS   8          ; value tempolary,In key
COUNT:    DFS   1          ; value conter
TEMPV:    DFS   1          ; value tempolary
BY1_V:    DFS   1          ; value byte1 for save Keyin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      BY2_V:   DFS   1          ; value byte2 for save Keyin
      F_CODE2: DFS   1          ; value group of food
      F_CODE1: DFS   1          ; value type of food
;     * * * * * * * * * * * * * * * * *
      ENDRAM:
;     * * * * * * * * * * * * * * * * *
;     *   Data Base  Code of food   *
;     * * * * * * * * * * * * * * * * *
 FOODLST: DFB   00H,01H,"น้ำพริกปลาทู",0DH
          DFB        02H,"น้ำพริกตาแดง",0DH
          DFB        03H,"น้ำพริกลงเรือ",0DH
          DFB        04H,"น้ำพริกหนุ่ม",0DH
          DFB        05H,"น้ำพริกกะปิ",0DH,0FFH
          DFB   02H,00H,"ยำหมูยอทรงเครื่อง",0DH
          DFB        01H,"ยำปลาหมึกย่าง",0DH
          DFB        02H,"ยำปลาดุกฟู",0DH
          DFB        03H,"ตำมะม่วงปูเค็มสด",0DH
          DFB        04H,"ตำมะม่วงกุ้งสดวิตามิน",0DH
          DFB        05H,"ตำมะม่วงทะเล",0DH,0FFH
          DFB   05H,00H,"น้ำพริกเผาตราสิงห์",0DH
          DFB        01H,"น้ำพริกเชียงใหม่",0DH
          DFB        02H,"เนื้อวัวสิงห์(ใหญ่)",0DH
          DFB        03H,"เนื้อวัวสิงห์(เล็ก)",0DH
          DFB        04H,"น้ำสมุนไพรสด",0DH
          DFB        05H,"น้ำมะพร้าว",0DH,0FFH,0FFH
;     END.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            close(DataFd);              //Close opened file before exit
            close(LogRdFd);
            close(LogWrFd);
            DestroyWindow(hwMain);
        }
          break;

        case FOOD_UPDATE:
      case ID_FUPDATE_BT:
        SendFoodList();          //Send food list to serial port
          break;
        case FOOD_EDIT:
      case ID_FEDIT_BT:
        DlgProc = (DLGPROC) MakeProcInstance(EditFoodProc, hInst);
        DialogBox(hInst, "EDITFOOD_DLG", hwMain, DlgProc);
        FreeProcInstance((FARPROC)DlgProc);
        break;
            case HELP_ABOUT:
        DlgProc = (DLGPROC) MakeProcInstance(AboutProc, hInst);
        DialogBox(hInst, "ABOUT_DLG", hwMain, DlgProc);
        FreeProcInstance((FARPROC)DlgProc);
        break;
        default:
          return(DefWindowProc(hWnd, msg, wParam, lParam));
      }
      break;
  default:      return(DefWindowProc(hWnd, msg, wParam, lParam));
 }
 return 0L;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Monolithic Function Generator

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 0.01 Hz to more than 1 MHz.

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

FEATURES

Low-Sine Wave Distortion	.5%, Typical
Excellent Temperature Stability	20 ppm/°C, Typical
Wide Sweep Range	2000:1, Typical
Low-Supply Sensitivity	0.01%V, Typical
Linear Amplitude Modulation	
TTL Compatible FSK Controls	
Wide Supply Range	10V to 26V
Adjustable Duty Cycle	1% to 99%

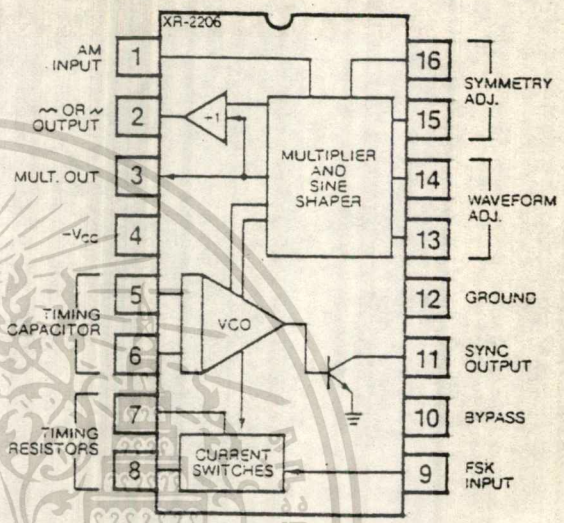
APPLICATIONS

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

ABSOLUTE MAXIMUM RATINGS

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

FUNCTIONAL BLOCK DIAGRAM



ORDERING INFORMATION

Part Number	Package	Operating Temperature
XR-2206M	Ceramic	-55°C to +125°C
XR-2206N	Ceramic	0°C to +70°C
XR-2206P	Plastic	0°C to +70°C
XR-2206CN	Ceramic	0°C to +70°C
XR-2206CP	Plastic	0°C to +70°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 actually produces an output frequency proportional to an input current, which is produced by a resistor from the timing terminals to ground. The current switches route one of the timing pins current to the VCO controlled by an FSK input pin, to produce an output frequency. With two timing pins, two discrete output frequencies can be independently produced for FSK Generation Applications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS

Test Conditions: Test Circuit of Figure 1, $V^+ = 12V$, $T_A = 25°$, $C = 0.01\ \mu F$, $R_1 = 100\ k\Omega$, $R_2 = 10\ k\Omega$, $R_3 = 25\ k\Omega$ unless otherwise specified. S_1 open for triangle, closed for sine wave.

PARAMETER	XR-2206M			XR-2206C			UNIT	CONDITIONS
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.		
GENERAL CHARACTERISTCS								
Single Supply Voltage	10		26	10		26	V	
Split-Supply Voltage	±5		±13	±5		±13	V	
Supply Current		12	17		14	20	mA	$R_1 \geq 10\ k\Omega$
OSCILLATOR SECTION								
Max. Operating Frequency	0.5	1		0.5	1		MHz	$C = 1000\ pF$, $R_1 = 1\ k\Omega$
Lowest Practical Frequency		0.01			0.01		Hz	$C = 50\ \mu F$, $R_1 = 2\ M\Omega$
Frequency Accuracy		±1	±4		±2		% of f_o	$f_o = 1/R_1C$
Temperature Stability		±10	±50		±20		ppm/°C	$0°C \leq T_A \leq 75°C$, $R_1 = R_2 = 20\ k\Omega$
Supply Sensitivity		0.01	0.1		0.01		%/V	$V_{LOW} = 10V$, $V_{HIGH} = 20V$, $R_1 = R_2 = 20\ k\Omega$
Sweep Range	1000:1	2000:1			2000:1		$f_H = f_L$	f_H @ $R_1 = 1\ k\Omega$, f_L @ $R_1 = 2\ M\Omega$
Sweep Linearity								
10:1 Sweep		2			2		%	$f_L = 1$ kHz, $f_H = 10$ kHz
1000:1 Sweep		8			8		%	$f_L = 100$ Hz, $f_H = 100$ kHz
FM Distortion		0.1			0.1		%	±10% Deviation
Recommended Timing Components								
Timing Capacitor: C	0.001		100	0.001		100	µF	See Figure 4.
Timing Resistors: R_1 & R_2	1		2000	1		2000	kΩ	
Triangle Sine Wave Output								See Note 1, Figure 2.
Triangle Amplitude		160			160		mV/kΩ	Figure 1, S_1 Open
Sine Wave Amplitude	40	60	80		60		mV/kΩ	Figure 1, S_1 Closed
Max. Output Swing		6			6		Vp-p	
Output Impedance		600			600		Ω	
Triangle Linearity		1			1		%	
Amplitude Stability		0.5			0.5		dB	For 1000:1 Sweep
Sine Wave Amplitude Stability		4800			4800		ppm/°C	See Note 2.
Sine Wave Distortion								
Without Adjustment		2.5			2.5		%	$R_1 = 30\ k\Omega$
With Adjustment		0.4	1.0		0.5	1.5	%	See Figures 6 and 7.
Amplitude Modulation								
Input Impedance	50	100		50	100		kΩ	
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		nsec	$C_L = 10\ pF$
Fall Time		50			50		nsec	$C_L = 10\ pF$
Saturation Voltage		0.2	0.4		0.2	0.6	V	$I_L = 2\ mA$
Leakage Current		0.1	20		0.1	100	µA	$V_{11} = 26V$
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.

Note 1: Output amplitude is directly proportional to the resistance, R_3, on Pin 3. See Figure 2.

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

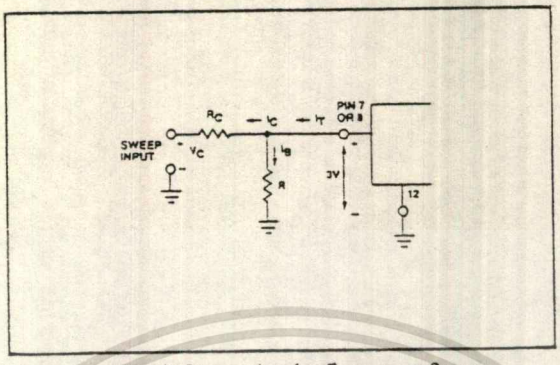


Figure 9: Circuit Connection for Frequency Sweep.

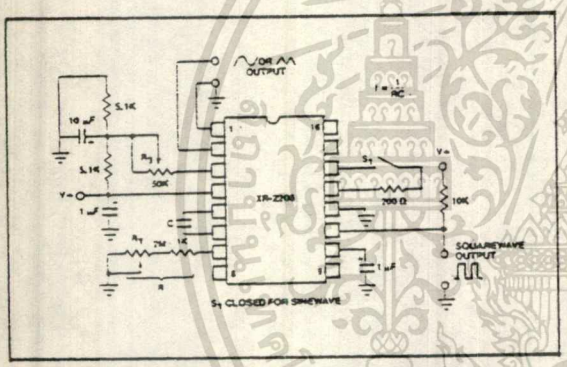


Figure 10: Circuit for Sine Wave Generation without External Adjustment. (See Figure 2 for Choice of R_3.)

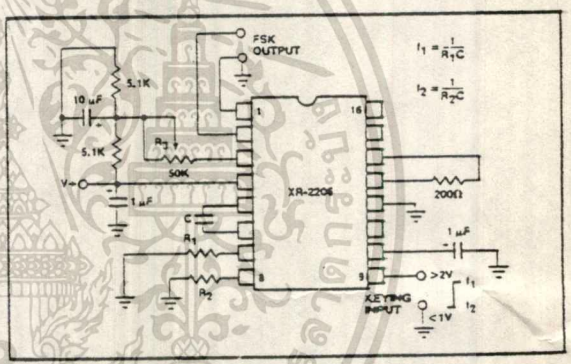


Figure 12: Sinusoidal FSK Generator.

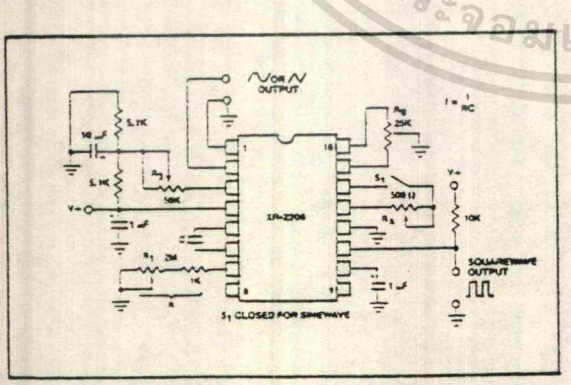


Figure 11: Circuit for Sine Wave Generation with Minimum Harmonic Distortion. (R_3 Determines Output Swing — See Figure 2.)

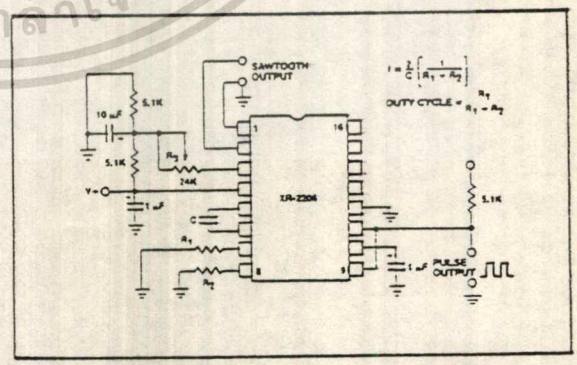


Figure 13: Circuit for Pulse and Ramp Generation.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 12. 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 Figures 10, 11 and 12, Pin 3 is biased midway between V^+ and ground, to give an output dc level of $\approx V^+/2$.

APPLICATIONS INFORMATION

Sine Wave Generation

Without External Adjustment:

Figure 10 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 10 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.

With External Adjustment:

The harmonic content of sinusoidal output can be reduced to $\approx$0.5% by additional adjustments as shown in Figure 11. 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 Figures 10 and 11 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 12 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 13 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 1 kΩ to 2 MΩ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

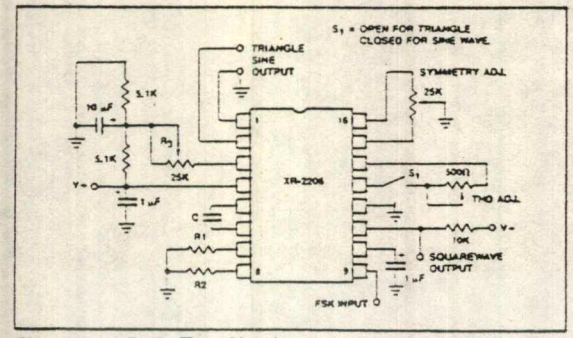


Figure 1: Basic Test Circuit.

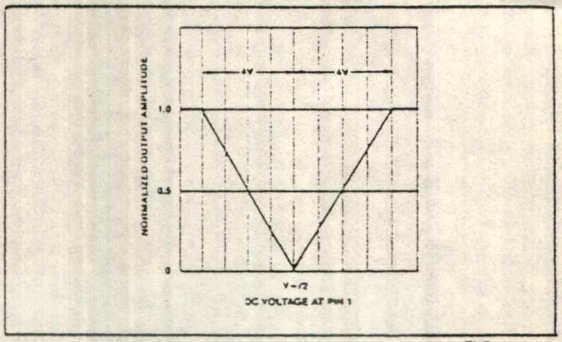


Figure 5: Normalized Output Amplitude versus DC Bias at AM Input (Pin 1).

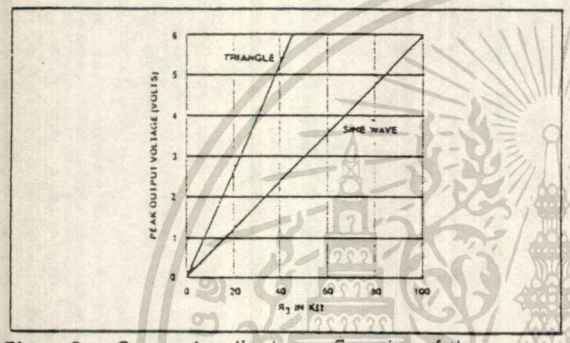


Figure 2: Output Amplitude as a Function of the Resistor, R_3, at Pin 3.

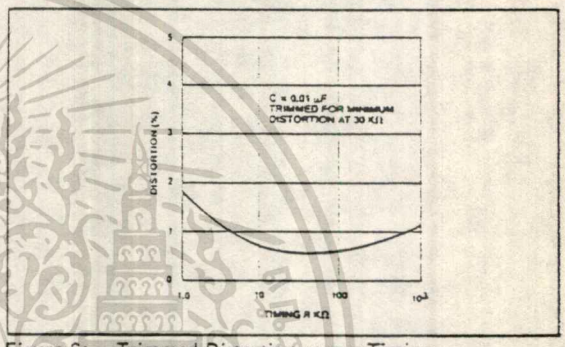


Figure 6: Trimmed Distortion versus Timing Resistor.

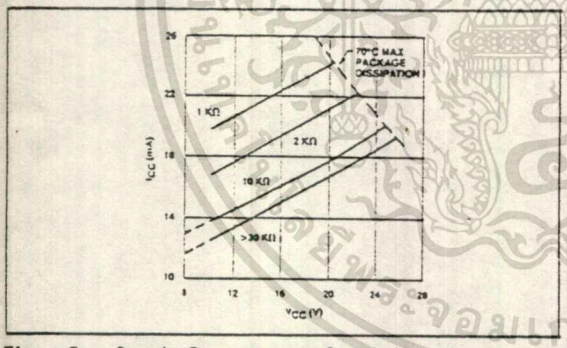


Figure 3: Supply Current versus Supply Voltage, Timing, R.

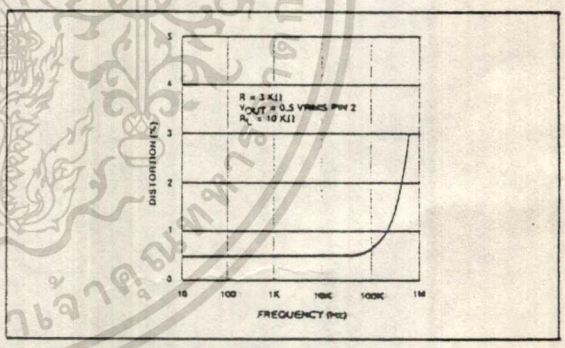


Figure 7: Sine Wave Distortion versus Operating Frequency with Timing Capacitors Varied.

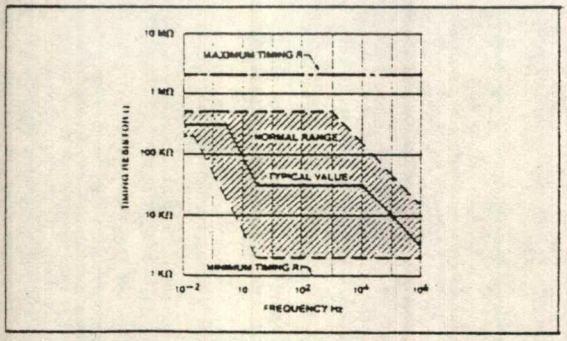


Figure 4: R versus Oscillation Frequency.

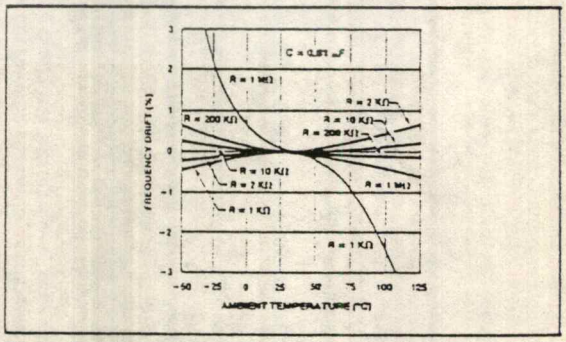


Figure 8: Frequency Drift versus Temperature.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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, are shown in Figure 4. Temperature stability is optimum for 4 kΩ < R < 200 kΩ. Recommended values of C are from 1000 pF to 100 μF.

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

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

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 μA to 3 mA. The frequency can be controlled by applying a control voltage, V_C, to the activated timing pin as shown in Figure 9. The frequency of oscillation is related to V_C as:

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

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

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

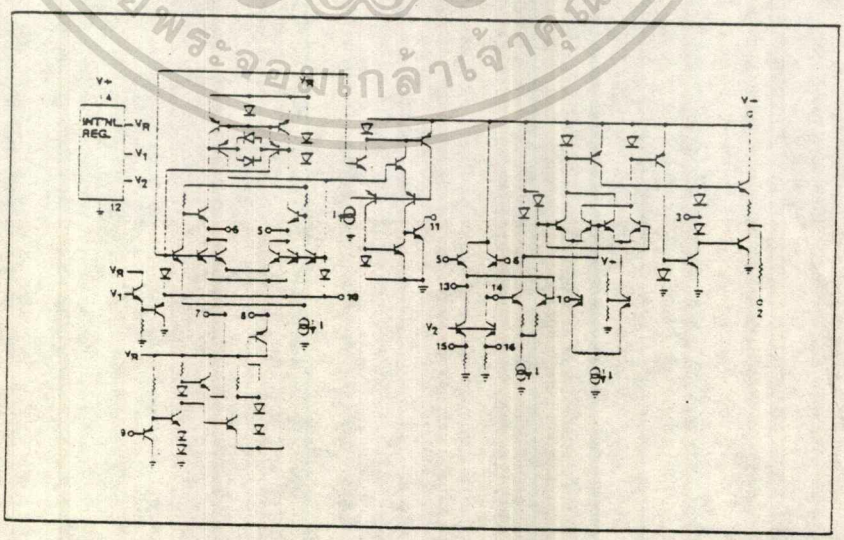
CAUTION: For safe operation of the circuit, I_T should be limited to ≤3 mA.

Output Amplitude:
Maximum output amplitude is inversely proportional to the external resistor, R_3, connected to Pin 3 (see Figure 2). For sine wave output, amplitude is approximately 60 mV peak per kΩ of R_3; for triangle, the peak amplitude is approximately 160 mV peak per kΩ of R_3. Thus, for example, R_3 = 50 kΩ would produce approximately ±3V 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 100 kΩ. Output amplitude varies linearly with the applied voltage at Pin 1, for values of dc bias at this pin, within ±4 volts of $V^+/2$ as shown in Figure 5. As this bias level approaches $V^+/2$, 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 55 dB.

CAUTION: AM control must be used in conjunction with a well-regulated supply, since the output amplitude now becomes a function of V^+.

EQUIVALENT SCHEMATIC DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HIGH QUALITY FUNCTION GENERATOR SYSTEM WITH THE XR-2206

January 1982

INTRODUCTION

Waveform or function generators capable of producing am/fm modulated sine wave outputs find a wide range of applications in electrical measurement and laboratory instrumentation. This application note describes the design, construction and the performance of such a complete function generator system suitable for laboratory usage or hobbyist applications. The entire function generator is comprised of a single XR-2206 monolithic IC and a limited number of passive circuit components. It provides the engineer, student, or hoobyist with a highly versatile laboratory instrument for waveform generation at a very small fraction of the cost of conventional function generators available today.

GENERAL DESCRIPTION

The basic circuit configuration and the external components necessary for the high-quality function generator system is shown in Figure 1. The circuit shown in the figure is designed to operate with either a 12 V single power supply, or with ±6 V split supplies. For most applications, split-supply operation is preferred since it results in an output dc level which is nearly at ground potential.

The circuit configuration of Figure 1 provides three basic waveforms: sine, triangle and square wave. There are four overlapping frequency ranges which give an overall frequency range of 1 Hz to 100 kHz. In each range, the frequency may be varied over a 100:1 tuning range.

The sine or triangle output can be varied from 0 to over 6 V (peak to peak) from a 600 ohm source at the output terminal.

A squarewave output is available at the sync output terminal for oscilloscope synchronizing or driving logic circuits.

TYPICAL PERFORMANCE CHARACTERISTICS

The performance characteristics listed below are not guaranteed or warranted by Exar. However, they represent the typical performance characteristics measured by Exar's application engineers during the laboratory evaluation of the function generator system shown in Figure 1. The typical performance specifications listed below apply only when all of the recommended assembly instructions and adjustment procedures are followed:

(a) **Frequency Ranges:** The function generator system is designed to operate over four overlapping frequency ranges:

1 Hz to 100 Hz
10 Hz to 1 kHz
100 Hz to 10 kHz
1 kHz to 100 kHz

The range selection is made by switching in different timing capacitors.

(b) **Frequency Setting:** At any range setting, frequency can be varied over a 100:1 tuning range with a potentiometer (see R_{13} of Figure 1).

(c) **Frequency Accuracy:** Frequency accuracy of the XR-2206 is set by the timing resistor R and the timing capacitor C, and is given as:

$$f = 1/RC$$

The above expression is accurate to within ±5 % at any range setting. The timing resistor R is the series combination of resistors R_4 and R_{13} of Figure 1. The timing capacitor C is any one of the capacitors C_3 through C_6, shown in the figure.

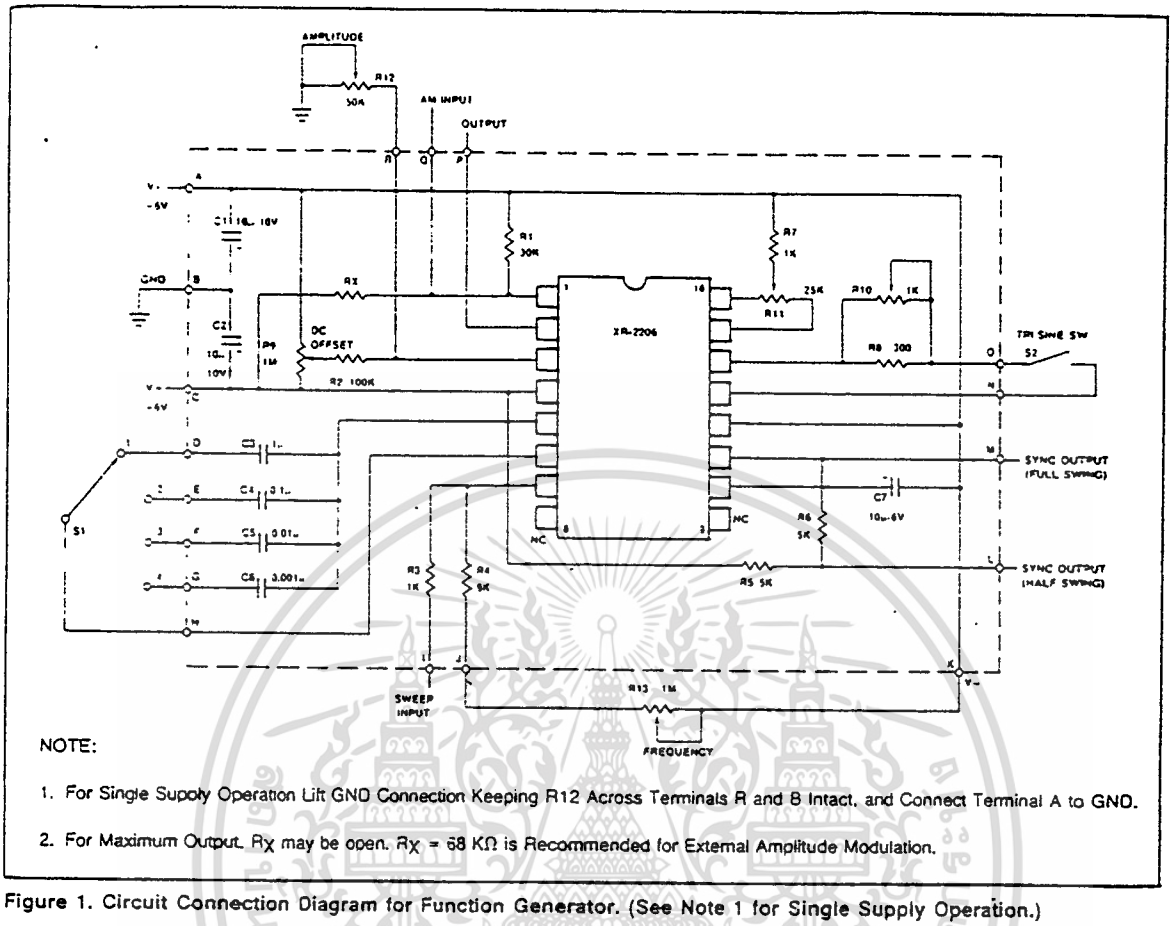
(d) **Sine and Triangle Output:** The sine and triangle output amplitudes are variable from 0 V to 6 V_{pp}. The amplitude is set by an external potentiometer, R_{12} of Figure 1. At any given amplitude setting, the triangle output amplitude is approximately twice as high as the sinewave output. The internal impedance of the output is 600 Ω.

(e) **Sinewave Distortion:** The total harmonic distortion of sinewave is less than 1 % from 10 Hz to 10 kHz and less than 3 % over the entire frequency range. The selection of a waveform is made by the triangle/sine selector switch, S_2.

(f) **Sync Output:** The sync output provides a 50 % duty cycle pulse output with either full swing or upper half swing of the supply voltage depending on the choice of sync output terminals on the printed circuit board (see Figure 1).

EXAR INTEGRATED SYSTEMS, INC.
750 Palomar Avenue, Sunnyvale, CA 94088 (408) 732-7970, TWX 910-339-9233

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



NOTE:

1. For Single Supply Operation Lift GND Connection Keeping R12 Across Terminals A and B Intact, and Connect Terminal A to GND.

2. For Maximum Output, R_X may be open. R_X = 68 KΩ is Recommended for External Amplitude Modulation.

Figure 1. Circuit Connection Diagram for Function Generator. (See Note 1 for Single Supply Operation.)

(g) **Frequency Modulation (External Sweep):** Frequency can be modulated or swept by applying an external control voltage to sweep terminal (Terminal I of Figure 1). When not used, this terminal should be left open-circuited. The open circuit voltage at this terminal is approximately 3 V above the negative supply voltage and its impedance is approximately 1000 ohms.

(h) **Amplitude Modulation:** The output amplitude varies linearly with modulation voltage applied to am input (terminal Q of Figure 1). The output amplitude reaches its minimum as the am control voltage approaches the half of the total power supply voltage. The phase of the output signal reverses as the amplitude goes through its minimum value. The total dynamic range is approximately 55 dB, with am control voltage range of 4 V referenced to the half of the total supply voltage. When not used, am terminal should be left open-circuited.

(i) **Power Source:** Split supplies: ±6 V, or single supply: +12 V. Supply Current 15 mA (see Figure 3).

EXPLANATION OF CIRCUIT CONTROLS:

Switches

Range Select Switch, S1: Selects the frequency range of operation for the function generator. The frequency is inversely proportional to the timing capacitor connected across Pins 5 and 6 of the XR-2206 circuit. Nominal capacitance values and frequency ranges corresponding to switch positions of S1 are as follows:

Position	Nominal Range	Timing Capacitance
1	1 Hz to 100 Hz	1 µF
2	10 Hz to 1 kHz	0.1 µF
3	100 Hz to 10 kHz	0.01 µF
4	1 Hz to 100 kHz	0.001 µF

If additional frequency ranges are needed, they can be added by introducing additional switch positions.

Triangle/Sine Waveform Switch, S2: Selects the triangle or sine output waveform.

Trimmers and Potentiometers

Dc Offset Adjustment, R9: The potentiometer used for adjusting the dc offset level of the triangle or sine output waveform.

Sinewave Distortion Adjustment, R10: Adjusted to minimize the harmonic content of sinewave output.

Sinewave Symmetry Adjustment, R11: Adjusted to optimize the symmetry of the sinewave output.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Amplitude Control, R12: Sets the amplitude of the triangle or sinewave output.

Frequency Adjust, R13: Sets the oscillator frequency for any range setting of S1. Thus, R13 serves as a frequency dial on a conventional waveform generator and varies the frequency of the oscillator over an approximate 100 to 1 range.

Terminals

A. Negative Supply −6 V
B. Ground
C. Positive Supply +6 V
D. Range 1, timing capacitor terminal
E. Range 2, timing capacitor terminal
F. Range 3, timing capacitor terminal
G. Range 4, timing capacitor terminal
H. Timing capacitor common terminal
I. Sweep Input
J. Frequency adjust potentiometer terminal
K. Frequency adjust potentiometer negative supply terminal
L. Sync output (1/2 swing)
M. Sync output (full swing)
N. Triangle/sine waveform switch terminals
O. Triangle/sine, waveform switch terminals
P. Triangle or sinewave output
Q. AM input
R. Amplitude control terminal

PARTS LIST

The following is a list of external circuit components necessary to provide the circuit interconnections shown in Figure 1.

Capacitors:

C1, C2, C7	Electrolytic, 10 μF, 10 V
C3	Mylar, 1 μF, nonpolar, 10 %
C4	Mylar, 0.1 μF, 10 %
C5	Mylar, 0.01 μF, 10 %
C6	Mylar, 1000 pF, 10 %

Resistors:

R1	30 KΩ, 1/4 W, 10 %
R2	100 KΩ, 1/4 W, 10 %
R3, R7	1 KΩ, 1/4 W, 10 %
R4	9 KΩ, 1/4 W, 10 %
R5, R6	5 KΩ, 1/4 W, 10 %
R8	300 Ω, 1/4 W, 10 %
RX	62 KΩ, 1/4 W, 10 % (RX can be eliminated for maximum output)

Potentiometers:

R9	Trim, 1 MΩ, 1/4 W
R10	Trim, 1 KΩ, 1/4 W
R11	Trim, 25 KΩ, 1/4 W

The following additional items are recommended to convert the circuit of Figure 1 to a complete laboratory instrument:

Potentiometers:

R12	Amplitude control, linear, 50 KΩ
R13	Frequency control, audio taper, 1 MΩ

Switches:

S1	Rotary switch, 1-pole, 4 positions
S2	Toggle or slide, SPST

Case:

7" x 4" x 4" (approx.) Metal or Plastic
(See Figures 4(a) and 4(b).)

Power Supply:

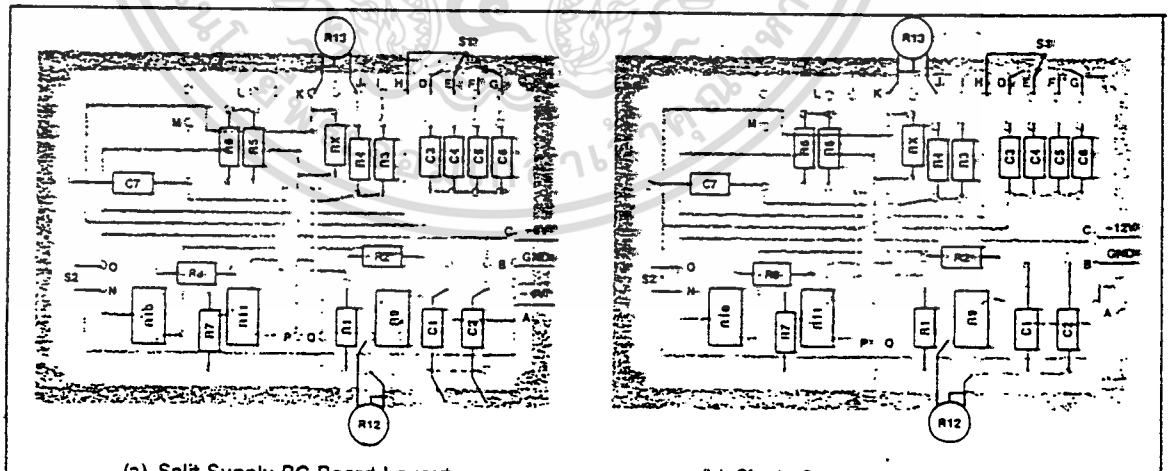
Dual supplies ±6 V or single +12 V
Batteries or power supply unit
(See Figures 3(a) and 3(b).)

Miscellaneous:

Knobs, solder, wires, terminals, etc.

BOARD LAYOUT

Figures 2(a) and 2(b) show the recommended printed-circuit board layout for the function generator circuit of Figure 1.



(a) Split Supply PC Board Layout

(b) Single Supply PC Board Layout

Figure 2. Recommended PC Board Layout for Function Generator Circuit of Figure 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RECOMMENDED ASSEMBLY PROCEDURE

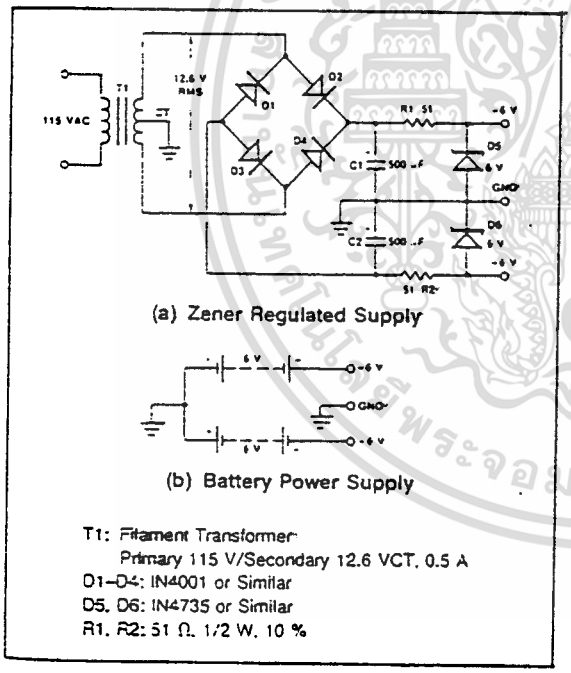
The following instructions and recommendations for the assembly of the function generator assume that the basic PC board layout of Figure 2(a) or 2(b) is used in the circuit assembly.

All the parts of the generator, with the exception of frequency adjust potentiometer, amplitude control potentiometer, triangle/sine switch and frequency range select switch, are mounted on the circuit board.

Install and solder all resistors, capacitors and trimmer resistors on the PC board first. Be sure to observe the polarity of capacitors C1, C2, C7. The timing capacitors C3, C4, C5 and C6 must be non-polar type. Now install IC1 on the board. We recommend the use of an IC socket to prevent possible damage to the IC during soldering and to provide for easy replacement in case of a malfunction.

The entire generator board along with power supply or batteries and several switches and potentiometers will fit into a case of the type readily available at electronic hobby shops. It will be necessary to obtain either output jacks or terminals for the outputs and am and frequency sweep inputs.

Install the frequency adjust pot, the frequency range select switch, the output amplitude control pot, the power switch, and the triangle/sine switch on the case. Next, install the PC board in the case, along with a power supply.



(a) Zener Regulated Supply

(b) Battery Power Supply

T1: Filament Transformer
Primary 115 V/Secondary 12.6 VCT, 0.5 A
D1–D4: IN4001 or Similar
D5, D6: IN4735 or Similar
R1, R2: 51 Ω, 1/2 W, 10 %

Figure 3. Recommended Power Supply Configurations.

Any simple power supply having reasonable regulation may be used. Figure 3 gives some recommended power supply configurations.

Precaution: Keep the lead lengths small for the range selector switch.

ADJUSTMENT PROCEDURE

When assembly is completed and you are ready to put the function generator into operation, make sure that the polarity of power supply and the orientation of the IC unit are correct. Then apply the dc power to the unit.

To adjust for minimum distortion, connect the scope probe to the triangle/sine output. Close S2 and adjust the amplitude control to give non-clipping maximum swing. Then adjust R10 and R11 alternately for minimum distortion by oberving the sinusoidal waveform. If a distortion meter is available, you may use it as a final check on the setting of sine-shaping trimmers. The minimum distortion obtained in this manner is typically less than 1 % from 1 Hz to 10 kHz and less than 3 % over the entire frequency range.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TIMER

SE/NE555/SE555C

DESCRIPTION

The 555 monolithic timing circuit is a highly stable controller capable of producing accurate time delays, or oscillation. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For a stable operation as an oscillator, the free running frequency and the duty cycle are both accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output structure can source or sink up to 200mA.

FEATURES

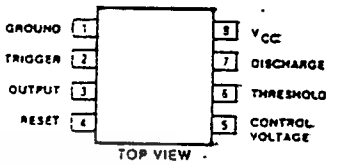
- Turn off time less than 2μs
- Maximum operating frequency greater than 500kHz
- Timing from microseconds to hours
- Operates in both astable and monostable modes
- High output current
- Adjustable duty cycle
- TTL compatible
- Temperature stability of 0.005% per °C

APPLICATIONS

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Missing pulse detector

PIN CONFIGURATIONS

D, N, FE PACKAGE



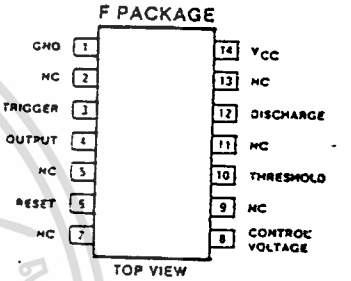
TOP VIEW

ORDER NUMBERS

SE/NE555N,FE NE555D

SE555CN,CFE

F PACKAGE



TOP VIEW

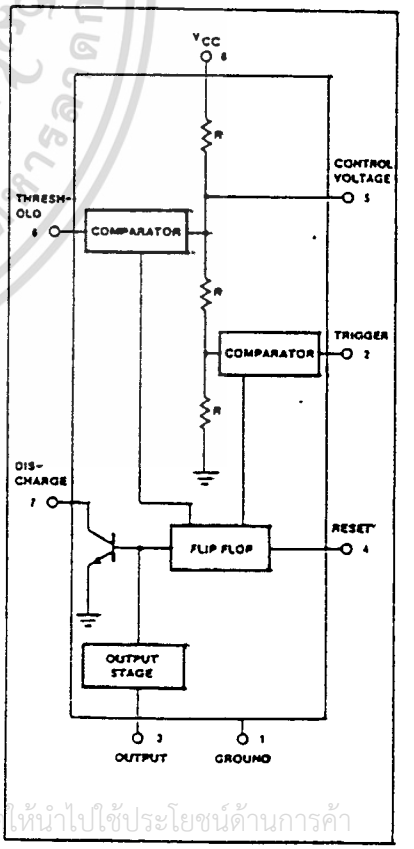
ORDER NUMBERS

SE/NE555F SE555CF

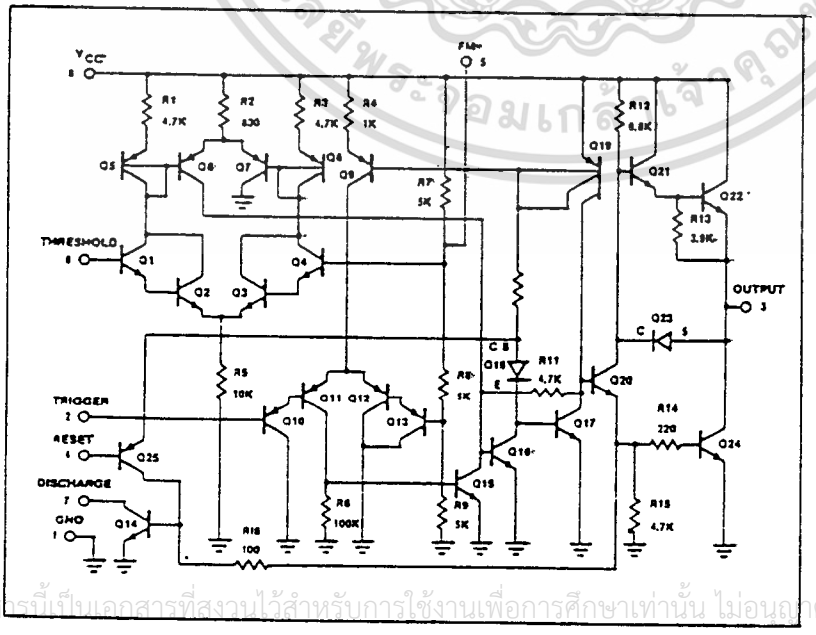
ABSOLUTE MAXIMUM RATINGS

PARAMETER	RATING	UNIT
Supply voltage		
SE555	+18	V
NE555, SE555C	+16	V
Power dissipation	600	mW
Operating temperature range		
NE555	0 to +70	°C
SE555, SE555C	-55 to +125	°C
Storage temperature range	-65 to +150	°C
Lead temperature (soldering, 60sec)	300	°C

BLOCK DIAGRAM



EQUIVALENT SCHEMATIC



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC ELECTRICAL CHARACTERISTICS $T_A = 25°C$, $V_{CC} = +5V$ to $+15$ unless otherwise specified.

PARAMETER	TEST CONDITIONS	SE555			NE555/SE555C			UNIT
		Min	Typ	Max	Min	Typ	Max	
Supply voltage		4.5		18	4.5		16	V
Supply current (low state)[1]	$V_{CC} = 5V$ $R_L = \infty$		3	5		3	6	mA
	$V_{CC} = 15V$ $R_L = \infty$		10	12		10	15	mA
Timing error (monostable)	$R_A = 2K\Omega$ to $100K\Omega$							
Initial accuracy[2]	$C = 0.1\mu F$		0.5	2.0		1.0	3.0	%
Drift with temperature			30	100		50	150	ppm/°C
Drift with supply voltage			0.05	0.2		0.1	0.5	%/V
Timing error (astable)	R_A, $R_B = 1k\Omega$ to $100k\Omega$							
Initial accuracy[2]	$C = 0.1\mu F$		4	6		5	13	%
Drift with temperature	$V_{CC} = 15V$			500			500	ppm/°C
Drift with supply voltage			0.15	0.6		0.3	1	%/V
Control voltage level	$V_{CC} = 15V$	9.6	10.0	10.4	9.0	10.0	11.0	V
	$V_{CC} = 5V$	2.9	3.33	3.8	2.6	3.33	4.0	V
Threshold voltage	$V_{CC} = 15V$	9.4	10.0	10.6	8.8	10.0	11.2	V
	$V_{CC} = 5V$	2.7	3.33	4.0	2.4	3.33	4.2	V
Threshold current[3]			0.1	0.25		0.1	0.25	μA
Trigger voltage	$V_{CC} = 15V$	4.8	5.0	5.2	4.5	5.0	5.6	V
	$V_{CC} = 5V$	1.45	1.67	1.9	1.1	1.67	2.2	V
Trigger current	$V_{TRIG} = 0V$		0.5	0.9		0.5	2.0	μA
Reset voltage[4]		0.3		1.0	0.3		1.0	V
Reset current			0.1	0.4		0.1	0.4	mA
Reset current	$V_{RESET} = 0V$		0.4	1.0		0.4	1.5	mA
Output voltage (low)	$V_{CC} = 15V$							
	$I_{SINK} = 10mA$		0.1	0.15		0.1	0.25	V
	$I_{SINK} = 50mA$		0.4	0.5		0.4	0.75	V
	$I_{SINK} = 100mA$		2.0	2.2		2.0	2.5	V
	$I_{SINK} = 200mA$		2.5			2.5		V
	$V_{CC} = 5V$							
	$I_{SINK} = 8mA$		0.1	0.25		0.3	0.4	V
	$I_{SINK} = 5mA$		0.05	0.2		0.25	0.35	V
Output voltage (high)	$V_{CC} = 15V$							
	$I_{SOURCE} = 200mA$		12.5			12.5		V
	$I_{SOURCE} = 100mA$	13.0	13.3		12.75	13.3		V
	$V_{CC} = 5V$							
	$I_{SOURCE} = 100mA$	3.0	3.3		2.75	3.3		V
Turn off time[5]	$V_{RESET} = V_{CC}$		0.5	2.0		0.5	2.0	μs
Rise time of output			100	200		100	300	ns
Fall time of output			100	200		100	300	ns
Discharge leakage current			20	100		20	100	na

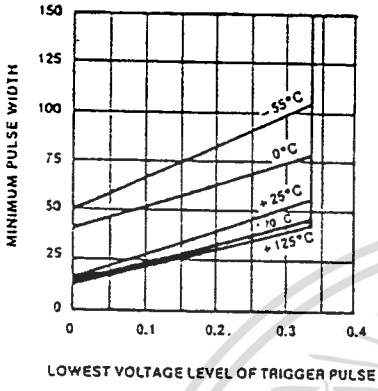
NOTES

1. Supply current when output high typically 1mA less.
2. Tested at $V_{CC} = 5V$ and $V_{CC} = 15V$.
3. This will determine the maximum value of $R_A + R_B$, for 15V operation, the max total R = 10 megohm, and for 5V operation, the max total R = 3.4 megohm.
4. Specified with trigger input high.
5. Time measured from a positive going input pulse from 0 to 0.8 x V_{CC} into the threshold to the drop from high to low of the output. Trigger is tied to threshold.

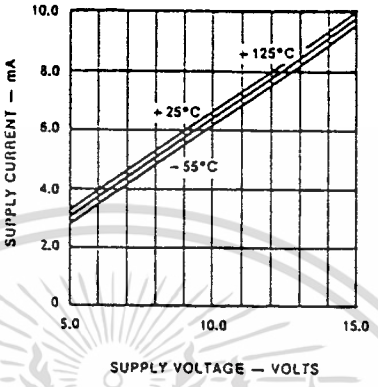
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL PERFORMANCE CHARACTERISTICS

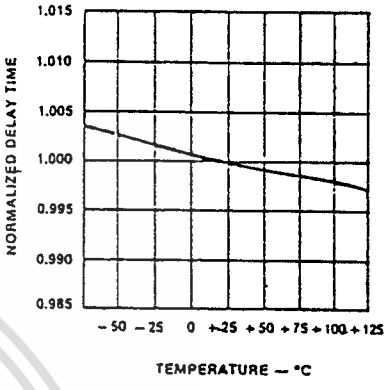
MINIMUM PULSE WIDTH REQUIRED FOR TRIGGERING



SUPPLY CURRENT vs SUPPLY VOLTAGE

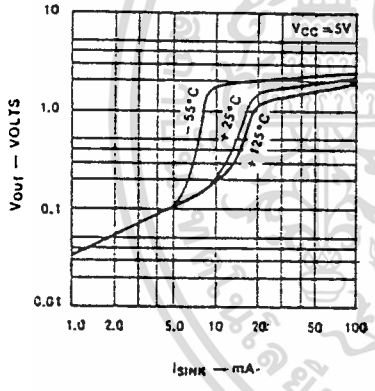


DELAY TIME vs TEMPERATURE

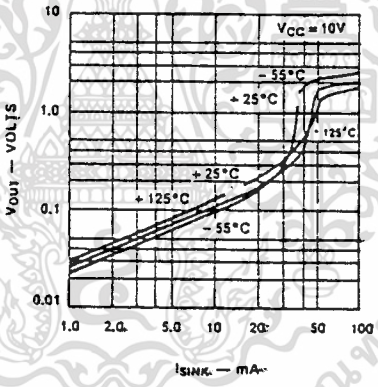


5

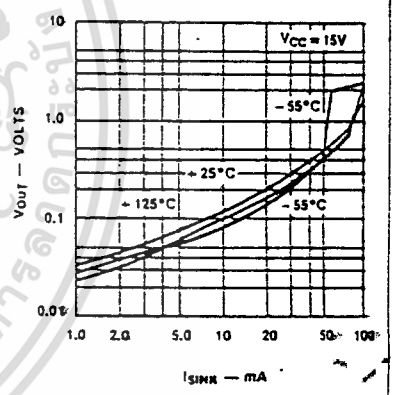
LOW OUTPUT VOLTAGE vs OUTPUT SINK CURRENT



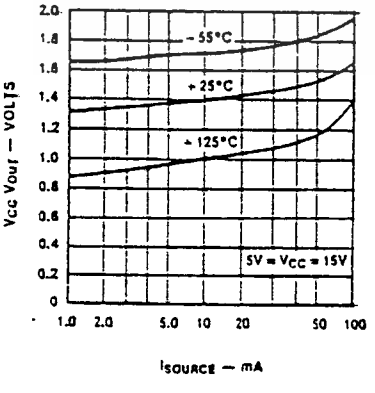
LOW OUTPUT VOLTAGE vs OUTPUT SINK CURRENT



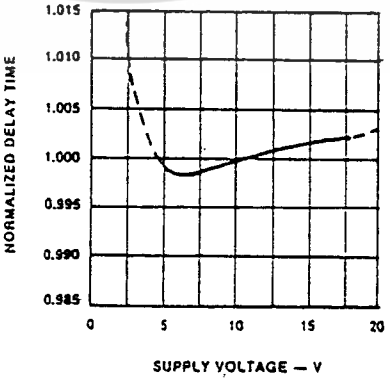
LOW OUTPUT VOLTAGE vs OUTPUT SINK CURRENT



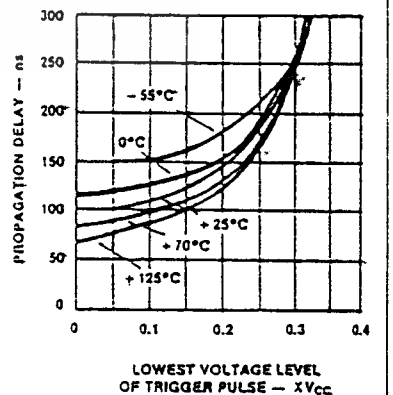
HIGH OUTPUT VOLTAGE DROP vs OUTPUT SOURCE CURRENT



DELAY TIME vs SUPPLY VOLTAGE



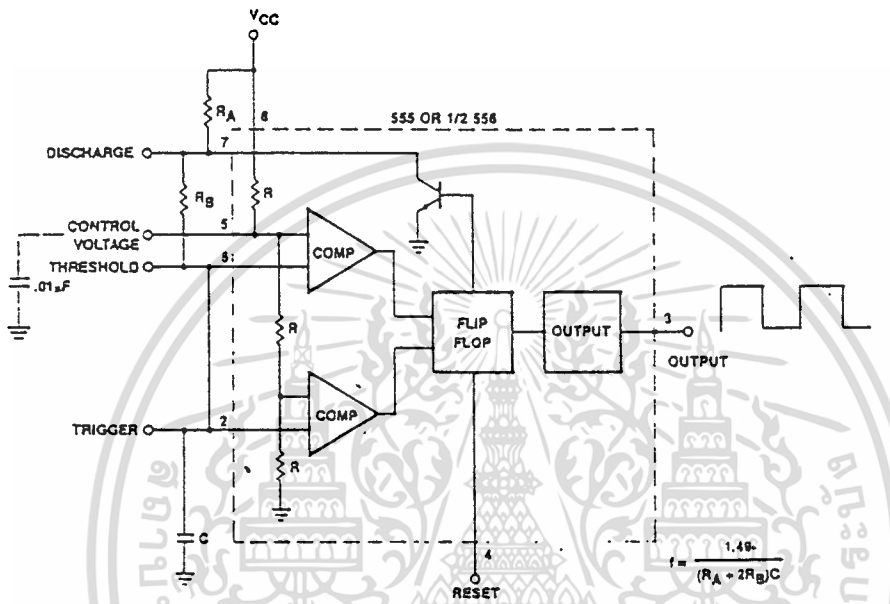
PROPAGATION DELAY vs VOLTAGE LEVEL OF TRIGGER PULSE



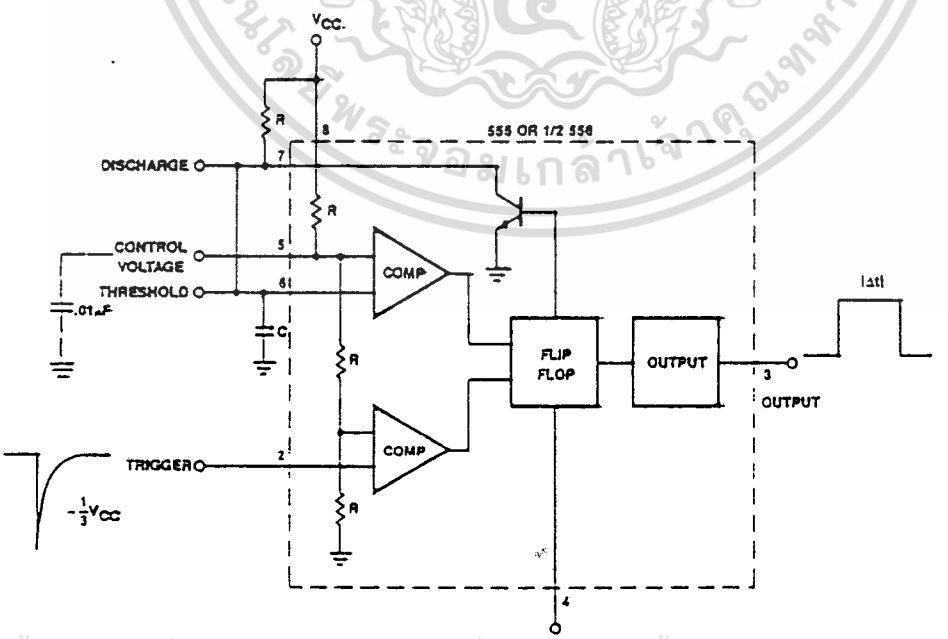
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATIONS

ASTABLE OPERATION



MONOSTABLE OPERATION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Trigger Pulse Width Requirements and Time Delays

Due to the nature of the trigger circuitry, the timer will trigger on the negative going edge of the input pulse. For the device to time out properly, it is necessary that the trigger voltage level be returned to some voltage greater than one third of the supply before the time out period. This can be achieved by making either the trigger pulse sufficiently short or by AC coupling into the trigger. By AC coupling the trigger, see Figure 1, a short negative going pulse is achieved when the trigger signal goes to ground. AC coupling is most frequently used in conjunction with a switch or a signal that goes to ground which initiates the timing cycle. Should the trigger be held low, without AC coupling, for a longer duration than the timing cycle the output will remain in a high state for the duration of the low trigger signal, without regard to the threshold comparator state. This is due to the predominance of Q_{15} on the base of Q_{16}, controlling the state of the bistable flip-flop. When the trigger signal then returns to a high level, the output will fall immediately. Thus, the output signal will follow the trigger signal in this case.

TYPICAL APPLICATIONS

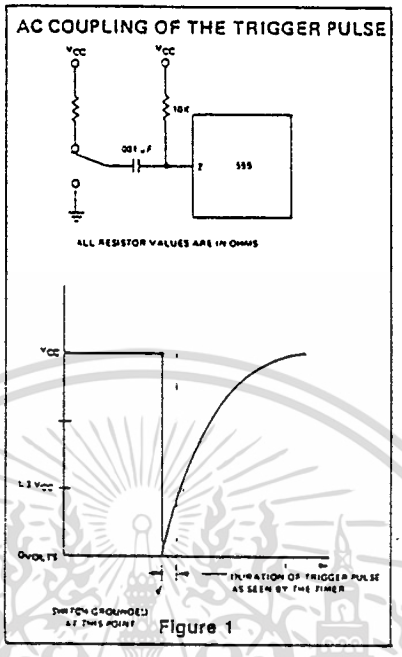


Figure 1

Another consideration is the "turn off time". This is the measurement of the amount of time required after the threshold reaches 2/3 V_{CC} to turn the output low. To explain further, Q_1 at the threshold input turns on after reaching 2/3 V_{CC}, which then turns on Q_5, which turns on Q_6. Current from Q_6 turns on Q_{16} which turns Q_{17} off. This allows current from Q_{19} to turn on Q_{20} and Q_{24} to give an output low. These steps cause the 2μs maximum delay as stated in the data sheet.

Also, a delay comparable to the turn off time is the trigger release time. When the trigger is low, Q_{10} is on and turns on Q_{11} which turns on Q_{15}. Q_{15} turns off Q_{16} and allows Q_{17} to turn on. This turns off current to Q_{20} and Q_{24}, which results in output high. When the trigger is released, Q_{10} and Q_{11} shut off, Q_{15} turns off, Q_{16} turns on and the circuit then follows the same path and time delay explained as "turn off time". This trigger release time is very important in designing the trigger pulse width so as not to interfere with the output signal as explained previously.

5

SCHEMATIC 555 OR 1/2 556 DUAL TIMER

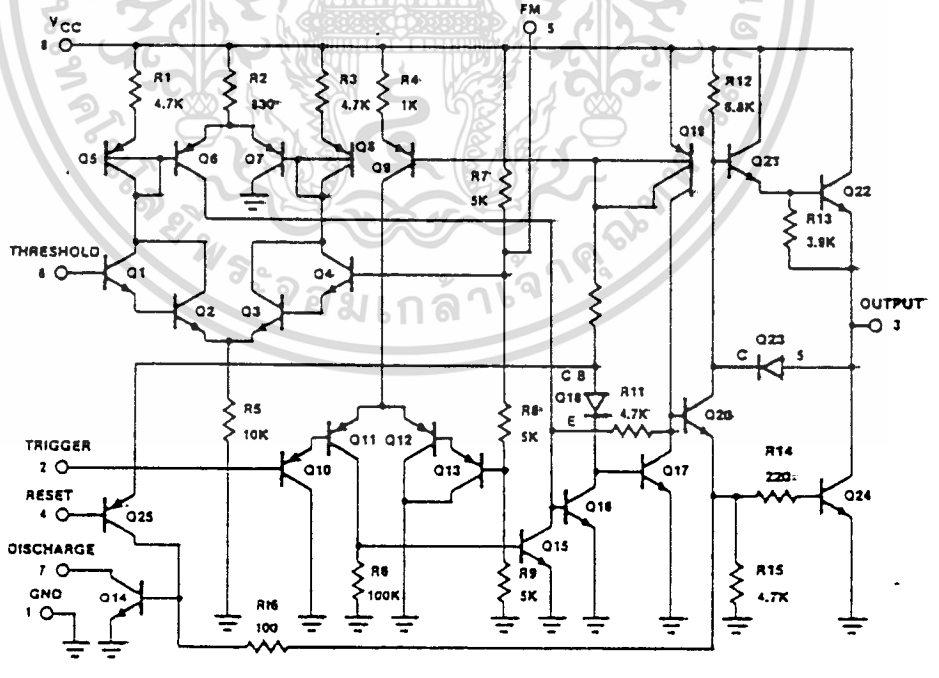


Figure 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ เรื่อง เครื่องสั่งอาหารระยะไกล สามารถสำเร็จลง ได้ด้วยดี โดยได้รับคำแนะนำ และความช่วยเหลือเป็นอย่างดี จากท่านอาจารย์รัตติกร วรากูลศิริพันธุ์ , อาจารย์พรชัยยศ ศรีปัญญาพงศ์ ,รวมทั้งคณะอาจารย์ ตลอดจนเจ้าหน้าที่ในภาควิชาอิเล็กทรอนิกส์ทุกท่าน คณะผู้จัดทำขอขอบพระคุณทุกท่านเป็นอย่างสูง และขอขอบคุณเพื่อนๆ ร่วมรุ่นที่ให้คำแนะนำ ช่วยเหลือด้วยดีตลอดมา และที่ขาดไม่ได้ก็คือ บิดา มารดา ผู้ให้กำลังใจและกำลังทรัพย์ในการศึกษาเล่าเรียนตลอดมา ขอกราบขอบพระคุณท่านมา ณ โอกาสนี้ด้วย

นายเฉลิมภัณฑ์ ฟองสมุทร
นายอนุโรจน์ แซ่โค้ว
นายอมรพงษ์ ชุมสาย ณ อยุธยา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. กำธร สถิรกุล ; " ลายสือไทย 700 ปี (ฉบับปรับปรุง)" , องค์การคุรุสภา , พ.ศ. 2527 หน้า 101-118
2. ศาสวัติ เกตุมและคณะ ; " ออกแบบอักษร " , อาร์ตไดเร็กโทรพับริเคชั่น , พ.ศ. 2529 หน้า 58-69
3. ชูชีพ ประเสริฐกุล ; " อักษรบนจอ LCD ฉบับที่ 67" , พ.ศ. 2535 , หน้า 32-48
4. ประเสริฐ คำดี ; " เรียนรู้เกี่ยวกับโมเด็ม" ,โดยฟิสิกส์เซ็นเตอร์ , พ. ศ. 2536 หน้า 67-75
5. น.ต. ฉัตรชัย สุมามาลย์ ; " การสื่อสารข้อมูลคอมพิวเตอร์และระบบเครือข่าย " พิมพ์ครั้งที่ 2 , พ.ศ. 2534 หน้า 25-36
6. H.S. Hou , Digital Document Processing , John- Wiley & Sons , 1983 Page 189-201
7. ANSI X3 , 122-1986 , Computer Graphics- Metafile for the Storage and Transfer of Picture Description Institute 1986 Page 120-135

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้