

ปริญญาานิพนธ์

มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11
MULTI BOARD MICROCONTROLLER 68HC11



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ **ปีการศึกษา 2537** นั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วิทยาลัยการศึกษาศาสตร์วิศวกรรม
 คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ใบบัตรสอบปริิณญาณิพนธ์

ชื่อหัวข้อปริิณญาณิพนธ์ มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11

MULTI BOARD MICROCONTROLLER 68HC11

- ชื่อนักศึกษา
1. นายจรูญศักดิ์ สิริปัญญาพาน รหัสประจำตัว 36031302
 2. นายประพันธ์ อีณา รหัสประจำตัว 36031315
 3. นายสมชาติ ชินวานิชย์เจริญ รหัสประจำตัว 36031327
 4. นายสมนึก บุญประเสริฐ รหัสประจำตัว 36031328

อาจารย์ผู้ควบคุมปริิณญาณิพนธ์

1. อาจารย์กิติพงษ์ มะโน
2. อาจารย์วรวิทย์ สมหา
3. อาจารย์สันติ ตันตระกูล

คณะกรรมการสอบปริิณญาณิพนธ์	ลายมือ
1. อาจารย์กิติพงษ์ มะโน	
2. อาจารย์วิสุทธิ์ อภิพรกรรม	
3. อาจารย์วรวิทย์ สมหา	
4. อาจารย์สชิน อาจหาญ	
5. อาจารย์สันติ ตันตระกูล	

วัน/เดือน/ปี ที่สอบ 29 ตุลาคม 2537 เวลา 10.30 น. ถึง 11.30 น.

สถานที่สอบ ห้อง ค.301 คณะวิศวกรรมศาสตร์



ปริญญานิพนธ์

เรื่อง มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11

MULTI BOARD MICROCONTROLLER 68HC11

ผู้จัดทำ

- | | | | |
|-------------------|-----------------|--------------|----------|
| 1. นายจรรุญศักดิ์ | สิริปัญญาพาน | รหัสประจำตัว | 36031302 |
| 2. นายประพันธ์ | อินจา | รหัสประจำตัว | 36031315 |
| 3. นายสมชาติ | ชินวานิชย์เจริญ | รหัสประจำตัว | 36031327 |
| 4. นายสมนึก | บุญประเสริฐ | รหัสประจำตัว | 36031328 |

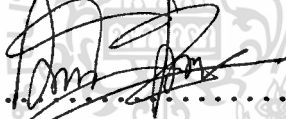
อาจารย์ที่ปรึกษา

ลงนาม



(อาจารย์ชิตพงษ์ มะโน)

ลงนาม



(อาจารย์วรวงศ์ สมหา)

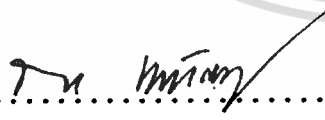
ลงนาม



(อาจารย์สันติ ตันตระกูล)

หัวหน้าภาควิชา

ลงนาม



(ผศ.ดร. ชีระพล เทพหัสติน ณ อรุณยา)

ปีการศึกษา 2537

ปริญญาานิพนธ์

เรื่อง มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 MULTI BOARD MICROCONTROLLER 68HC11

จุดประสงค์

1. เพื่อศึกษาถึงระบบการทำงานของไมโครคอนโทรลเลอร์ 68HC11
2. เพื่อศึกษาการใช้ภาษาแอสเซมบลีในการเขียนโปรแกรมควบคุมระบบให้ลิกซิ่งยังขึ้น
3. เพื่อวิเคราะห์เปรียบเทียบมัลติบอร์ดโดยใช้ไอซีเบอร์อื่น โดยศึกษาข้อดีและข้อเสีย
4. เพื่อสร้างชุดต้นแบบมัลติบอร์ดโดยใช้ไอซีไมโครคอนโทรลเลอร์ 68HC11
5. เพื่อเป็นแนวทางในการพัฒนาเครื่องต้นแบบให้มีประสิทธิภาพการทำงานที่สมบูรณ์ต่อไป

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจระบบการทำงานของไมโครคอนโทรลเลอร์ 68HC11
2. ได้ความรู้เกี่ยวกับเทคนิคการเขียนโปรแกรมภาษาแอสเซมบลีของ 68HC11 ตลอดจนวิธีการอินเตอร์เฟสระหว่างมัลติบอร์ดกับเครื่องคอมพิวเตอร์ โดยผ่านพอร์ต RS-232 ในการพัฒนาโปรแกรม
3. ได้มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 ไว้ใช้งาน 1 เครื่อง
4. เข้าใจถึงข้อดีและข้อเสียของการใช้ไอซีไมโครคอนโทรลเลอร์ 68HC11ในงานควบคุม
5. ได้แนวทางในการพัฒนามัลติบอร์ด 68HC11 สู่ VERSION ที่มีประสิทธิภาพสูงต่อไป

มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11

นายจรูญศักดิ์ สิริปัญญามาน
นายประพันธ์ อินจา
นายสมชาติ ชินวานิชย์เจริญ
นายสมนึก บุญประเสริฐ

อาจารย์ที่ปรึกษา

อ. กิติพงศ์ มะโน
อ. วรวิทย์ สมหา
อ. สันติ ตันตระกูล

ปีการศึกษา 2537

บทคัดย่อ

ในปฏิญานินพนธ์ฉบับนี้เสนอ มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 โดยสร้างเป็นชุดศึกษาการทำงานของไมโครคอนโทรลเลอร์ 68HC11 เพื่อให้ง่ายแก่การเรียนรู้และนำไปใช้งานได้อย่างมีประสิทธิภาพ การทำงานของมัลติบอร์ดนี้แบ่งเป็น 2 โหมด คือ

โหมด 1 ศึกษาการทำงานของไมโครคอนโทรลเลอร์โดยจะป้อนโปรแกรมที่อยู่ในรูปของภาษาเครื่องและจะประมวลผลโปรแกรมเพื่อศึกษาการทำงานในแต่ละส่วนบนมัลติบอร์ด ในโหมดนี้จะแสดงผลรีจิสเตอร์ เองอนไซและการเปลี่ยนแปลงสภาวะต่าง ๆ บนส่วนแสดงผลของมัลติบอร์ด

โหมด 2 ศึกษาการทำงานของไมโครคอนโทรลเลอร์โดยป้อนโปรแกรมที่เขียนอยู่ในรูปของภาษาแอสเซมบลีบนเครื่องคอมพิวเตอร์ ในโหมดนี้จะแอสเซมเบลอร์โปรแกรมด้วย AS11NEW และจะดาวน์โหลดโปรแกรมจากเครื่องคอมพิวเตอร์ผ่านพอร์ต RS-232C ลงไปสู่มัลติบอร์ด การแสดงผลของการเปลี่ยนแปลงสภาวะต่าง ๆ ภายในไมโครคอนโทรลเลอร์นั้นจะแสดงผลบนจอมอนิเตอร์ของเครื่องคอมพิวเตอร์

MULTI BOARD MICROCONTROLLER 68HC11

MR. JAROONSAK	SIRIPUNYACHAN
MR. PRAPUN	INJA
MR. SOMCHAT	CHINVANITCHAREUAN
MR. SOMNUK	BOONPRASERT

ADVISOR

MR. KITIPONG	MANO
MR. WORAWIT	SOMHA
MR. SUNTI	TONTRAKOOL

1994

Abstract :

This thesis presents a Multiboard Microcontroller 68HC11. It's created in the set for support to study the operating of the microcontroller 68HC11 and for good sufficiency used. Within this strumment has 2 operating mode.

Mode 1 For study the operating of microcontroller by keying the Machine Code program and running program to study in each part on Multiboard. This mode will show the condition register and any condition on display of Multiboard.

Mode 2 For study the operating of microcontroller by keying assembly program on personal computer. This mode will assembler the assembly program by AS11NEW and down load assembled program from personal computer pass RS-232c port to the multiboard. Any changed of condition in the microcontroller 68HC11 in this mode is show on monitor set of personal computer.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เพราะความช่วยเหลือและให้คำแนะนำจากอาจารย์ที่ปรึกษา ตลอดจนได้รับความช่วยเหลือในด้านเครื่องมือและอุปกรณ์ต่าง ๆ ในการทำปริญญานิพนธ์และการสร้างชิ้นงานจากภาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งทางคณะผู้จัดทำรู้สึกซาบซึ้งและขอบพระคุณเป็นอย่างสูง

ทั้งนี้ขอขอบคุณผู้ร่วมงานในกลุ่มที่ร่วมมือกันทำงานจนงานสำเร็จลุล่วงไปได้ด้วยดีและขอขอบคุณเพื่อน ๆ ที่ได้เอื้อเฟื้อในด้านเครื่องมือใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง

หน้า

บทคัดย่อภาษาไทย

บทคัดย่อภาษาอังกฤษ

กิตติกรรมประกาศ

สารบัญ

สารบัญตาราง

สารบัญภาพ

บทที่ 1 บทนำ

1

บทที่ 2 ทฤษฎี และหลักการ

2

2.1 68HC11 ไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่ใช้เทคโนโลยีไฮสปีดซีมอส

2

2.1.1 โครงสร้างของ 68HC11

2

2.1.2 คุณสมบัติของ 68HC11

4

2.1.3 ตระกูลของ 68HC11

4

2.1.4 โหมดการทำงานของ 68HC11

6

2.1.4.1 โหมดซิงเกิลชีพ

7

2.1.4.2 โหมดมัลติเพล็กซ์ขยาย

7

2.1.4.3 โหมดบุคคลตรีพิเศษ

10

2.1.4.4 โหมดทดสอบพิเศษ

11

2.1.5 ลักษณะของสัญญาณและการจัดขา

13

2.1.6 การจัดหน่วยความจำ

16

2.1.7 หน่วยความจำแรมภายในซีพียู

17

2.1.8 วงจรกำเนิดสัญญาณนาฬิกา

19

2.1.9 รีจิสเตอร์ของซีพียู

20

2.1.9.1 แอควิวเลเตอร์ A และ B

21

2.1.9.2 แอควิวเลเตอร์ D

21

2.1.9.3 รีจิสเตอร์อินเด็กซ์ IX

21

2.1.9.4 รีจิสเตอร์อินเด็กซ์ IY

21

2.1.9.5 รีจิสเตอร์ตัวชี้สแต็ค SP

22

2.1.9.6 รีจิสเตอร์โปรแกรมเคาน์เตอร์ PC

22

2.1.9.7 รีจิสเตอร์รหัสเงื่อนไข CCR

22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.10	วงจรวอตซ์ค็อก	23
2.1.11	วงจรถูกตี	23
2.1.12	ไทม์เมอร์	24
2.1.13	รีเลย์ไทม์	25
2.1.14	พัลส์แอกคิวมูเลเตอร์	25
2.1.15	รีเซ็ต	25
2.1.16	อินเตอร์รัพต์	26
2.1.17	พอร์ตอนุกรม	26
2.1.18	มัลติโปรเซสเซอร์	26
2.2	การอ้างแอดเดรสและชุดคำสั่งของ 68HC11	27
2.2.1	การอ้างแอดเดรสของ 68HC11	27
2.2.1.1	การอ้างแอดเดรสแบบทันทีทันใด	27
2.2.1.2	การอ้างแอดเดรสแบบโดยตรง	27
2.2.1.3	การอ้างแอดเดรสแบบขยาย	27
2.2.1.4	การอ้างแอดเดรสแบบอินเด็กซ์	28
2.2.1.5	การอ้างแอดเดรสแบบอินเฮียเรนต์	28
2.2.1.6	การอ้างแอดเดรสแบบสั้มพิทซ์	28
2.2.2	พรีไบต์ (Prebyte)	28
2.2.3	ชุดคำสั่งของ 68HC11	29
2.2.3.1	กลุ่มคำสั่งจัดการเกี่ยวกับข้อมูล	30
2.2.3.2	กลุ่มคำสั่งทางคณิตศาสตร์	32
2.2.3.3	กลุ่มคำสั่งทางลอจิก	33
2.2.3.4	กลุ่มคำสั่งการกระโดด	33
2.2.3.5	กลุ่มคำสั่งเปรียบเทียบและตรวจสอบข้อมูล	35
2.2.3.6	กลุ่มคำสั่งจัดการกับวีจีเอสเตอร์รหัสเงื่อนไข	35
2.2.3.7	กลุ่มคำสั่งควบคุม	36
2.2.4	ความแตกต่างของการ JUMP และ BRANCH	36
2.3	การเขียนโปรแกรมเพื่ออ้างแอดเดรส	37
2.3.1	โปรแกรมสำหรับการอ้างแอดเดรสแบบทันทีทันใด	37
2.3.2	โปรแกรมสำหรับการอ้างแอดเดรสแบบโดยตรง	37
2.3.3	โปรแกรมสำหรับการอ้างแอดเดรสแบบขยาย	38

เรื่อง	หน้า
2.3.4 โปรแกรมสำหรับการอ้างแอดเดรสแบบอินเด็กซ์	38
2.3.5 โปรแกรมสำหรับการอ้างแอดเดรสแบบอินเดี่ย เรนต์	38
2.3.6 โปรแกรมสำหรับการอ้างแอดเดรสแบบสัมพัทธ์	38
2.4 รูปแบบการเขียนโปรแกรมทั่วไป	39
2.5 โปรแกรมคณิตศาสตร์	40
2.5.1 โปรแกรมบวกลบเลข	40
2.5.2 โปรแกรมคูณเลข	41
2.5.3 โปรแกรมหารเลข	41
2.6 อินเตอร์เฟส RS-232 C	42
2.6.1 แบบจำลองของวงจรการสื่อสารแบบ RS-232 C	42
2.6.2 ข้อกำหนดทางไฟฟ้า	44
2.6.3 ขาเข้าต่อ RS-232 C และหน้าที่	45
2.6.4 การควบคุมคลื่นพาห์และการส่งข้อมูล	48
2.6.5 ช่องสัญญาณรอง	51
2.6.6 การตอบรับอย่างอัตโนมัติ	51
2.6.7 การให้จังหวะเวลาแก่วงจรซิงโครนิส	53
2.6.8 ข้อจำกัดของ RS-232 C	54
2.6.8.1 ระยะทาง	54
2.6.8.2 อัตราส่งข้อมูล	55
2.6.8.3 สัญญาณรบกวน	55
บทที่ 3 การออกแบบและการสร้าง	57
3.1 บล็อกไดอะแกรมของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	58
3.1.1 พัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์	58
3.1.2 พัฒนาโปรแกรมผ่านคีย์บอร์ดของมัลติบอร์ด	58
3.2 ส่วนประกอบโดยทั่วไป	58
3.2.1 ส่วนชุดควบคุม	58
3.2.2 ส่วนถอดรหัสพอร์ตภายนอก	58
3.2.3 ส่วนคีย์บอร์ด	58
3.2.4 ส่วนการแสดงผล 7 เซกเมนต์และแฟลกแอลอีดี	58
3.2.5 ส่วนการแสดงผลแอลซีดี	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6	ส่วนการแสดงผลทางลำโพง	59
3.2.7	ส่วนของการอินเตอร์เฟสกับอุปกรณ์ภายนอก	59
3.2.8	ส่วนของแหล่งจ่ายไฟ	59
3.2.9	ส่วนของบอร์ดทดลอง	59
3.3	การออกแบบและการทำงานในส่วนต่าง ๆ	59
3.3.1	ส่วนชุดควบคุม	59
3.3.1.1	การทำงานของชุดควบคุม	60
3.3.1.2	การจัดหน่วยความจำของบอร์ด	61
3.3.1.3	ระบบบัส	62
3.3.1.4	การเลือกหน่วยความจำแรม	62
3.3.1.5	การเลือกหน่วยความจำอีพ롬	62
3.3.1.6	แอดเดรสของพอร์ตอินพุตเอาต์พุต 8255	63
3.3.1.7	การเชื่อมต่อกับคีย์บอร์ด	63
3.3.1.8	แอดเดรสของพอร์ตเครื่องพิมพ์	64
3.3.1.9	การเชื่อมต่อกับจอแสดงผลแบบแอลซีดี	64
3.3.1.10	พอร์ตเชื่อมต่อวงจร A/D	65
3.3.1.11	พอร์ตตัวตั้งเวลาและตัวนับ	65
3.3.1.12	การเลือกโหมดการทำงาน	66
3.3.1.13	การนำสัญญาณนาฬิกาออกไปใช้งานภายนอก	66
3.3.1.14	การต่อจัมเปอร์ JOC5	66
3.3.1.15	การส่งข้อมูลอนุกรมและพอร์ตสื่อสารข้อมูลอนุกรม	67
3.3.1.16	พอร์ตสำหรับเชื่อมต่ออุปกรณ์อนุกรม (SPI)	68
3.3.1.17	การขยายพอร์ตเพิ่มเติมบนชุดควบคุม	68
3.3.1.18	การสร้าง	69
3.3.2	ส่วนถอดรหัสพอร์ตภายนอก	69
3.3.3	ส่วนคีย์บอร์ด	71
3.3.4	ส่วนแสดงผลบนแอลอีซี 7 เซกเมนต์และแอลอีซีแฟล็ก	71
3.3.5	ส่วนการแสดงผลแอลซีดี	72
3.3.6	ส่วนการแสดงผลทางลำโพง	73
3.3.7	ส่วนของการอินเตอร์เฟสกับอุปกรณ์ภายนอก	74
3.3.7.1	วงจรแสดงผลทางแอลอีซี	74

เรื่อง	หน้า
3.3.7.2 วงจรเชื่อมต่อเครื่องพิมพ์	75
3.3.7.3 วงจรวัดแรงดันไฟตรง 0-5 โวลต์ 8 ช่อง	76
3.3.8 ส่วนของแหล่งจ่ายไฟ	76
3.3.9 ส่วนของบอร์ดทดลอง	78
3.4 การสร้างมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	79
บทที่ 4 การทดลองและผลการทดลอง	82
4.1 การทดลองโปรแกรมทดสอบบนมัลติบอร์ด	82
4.2 การทดลองโปรแกรมทดสอบบนเครื่องคอมพิวเตอร์	87
4.3 สรุปเปรียบเทียบผลการทดลอง	95
บทที่ 5 บทวิจารณ์ สรุป และแนวทางในการพัฒนา	96
5.1 บทสรุป	96
5.2 ปัญหาและแนวทางแก้ไข	96
5.3 แนวทางในการพัฒนา	98
ภาคผนวก	100
ภาคผนวก ก วงจรส่วนต่าง ๆ ของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	100
ภาคผนวก ข การใช้งานมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	107
ภาคผนวก ค ลักษณะการวางอุปกรณ์และการจัดวางขาคอนเนคเตอร์	174
ภาคผนวก ง ฟีลด์ชาร์ทมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	182
ภาคผนวก จ โปรแกรมมอนิเตอร์มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	194
ภาคผนวก ฉ การจัดสรรหน่วยความจำของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	294
บรรณานุกรม	298

สารบัญตาราง

ตาราง		หน้า
ตารางที่ 2.1	ตระกูลของ 68HC11	5
ตารางที่ 2.2	การเลือกโหมดการทำงานของ 68HC11	7
ตารางที่ 2.3	อินเทอร์รัพต์เวคเตอร์ในโหมดการทำงานแบบบูตสแตร์ป	12
ตารางที่ 2.4	สรุปการใช้ขาของ 68HC11 ในโหมดการทำงานต่างๆ	15
ตารางที่ 2.5	รีจิสเตอร์ควบคุมและพอร์ตต่าง ๆ	19
ตารางที่ 2.6	การกำหนดช่องการทำงานและการเก็บข้อมูลของ วงจร A/D ใน 68HC11	24
ตารางที่ 2.7	กลุ่มคำสั่งเคลื่อนย้ายข้อมูล	30
ตารางที่ 2.8	กลุ่มคำสั่งเปลี่ยนแปลงและแก้ไขข้อมูล	31
ตารางที่ 2.9	กลุ่มคำสั่งการเลื่อนและหมุนข้อมูล	31
ตารางที่ 2.10	กลุ่มคำสั่งทางคณิตศาสตร์	32
ตารางที่ 2.11	กลุ่มคำสั่งทางลอจิก	33
ตารางที่ 2.12	กลุ่มคำสั่งการกระโดดแบบไม่มีเงื่อนไข	33
ตารางที่ 2.13	กลุ่มคำสั่งการกระโดดแบบมีเงื่อนไข	34
ตารางที่ 2.14	กลุ่มคำสั่งเปรียบเทียบและตรวจสอบข้อมูล	35
ตารางที่ 2.15	กลุ่มคำสั่งจัดการกับรีจิสเตอร์ที่สแต็ท	35
ตารางที่ 2.16	กลุ่มคำสั่งควบคุม	36
ตารางที่ 2.17	หน้าที่ของสายและความหมายของโวลต์เตจที่กำหนดให้	44
ตารางที่ 2.18	การกำหนดขาหัวต่อ RS-232 C	46
ตารางที่ 3.1	ตำแหน่งหน่วยความจำที่กำหนดที่สร้างสัญญาณ Chip Select ของ 8255 หมายเลข 1 และ 8255 หมายเลข 2	70
ตารางที่ 4.1	ผลการทดลองโปรแกรมทดสอบที่ 4.1	84
ตารางที่ 4.2	ผลการทดลองโปรแกรมทดสอบที่ 4.2	85
ตารางที่ 4.3	ผลการทดลองโปรแกรมทดสอบที่ 4.3	87
ตารางที่ 4.4	ผลการทดลองโปรแกรมทดสอบที่ 4.4	89
ตารางที่ 4.5	ผลการทดลองโปรแกรมทดสอบที่ 4.5	90
ตารางที่ 4.6	ผลการทดลองโปรแกรมทดสอบที่ 4.6	92
ตารางที่ 5.1	ปัญหาในการทำปริญญานิพนธ์และแนวทางการแก้ไข	97
ตารางที่ 5.2	ปัญหาในการทำปริญญานิพนธ์และแนวทางการแก้ไข	98

สารบัญรูปภาพ

	รูปภาพ	หน้า
รูปที่ 2.1	บล็อกไดอะแกรมของ 68HC11F1	3
รูปที่ 2.2	แบบตัวถังและการจัดขา	6
รูปที่ 2.3	วงจรการต่อ 68HC11 เมื่อทำงานในโหมดซิงเกิลชิพ	8
รูปที่ 2.4	วงจรตีลติเฟล็กซ์บัสแอดเดรสและบัสข้อมูลเมื่อกำหนดการทำงานในโหมดมัลติเฟล็กซ์ขยาย	9
รูปที่ 2.5	ตัวอย่างของวงจรการต่อใช้งาน 68HC11 เมื่อต้องการให้ทำงานในโหมดมัลติเฟล็กซ์ขยาย	10
รูปที่ 2.6	ไดอะแกรมการทำงานของ 68HC11 ในโหมดบุคคลตรีป	13
รูปที่ 2.7	การจัดหน่วยความจำในโหมดต่าง ๆ	16
รูปที่ 2.8	การต่อสัญญาณนาฬิกาให้กับ 68HC11	20
รูปที่ 2.9	การต่อสัญญาณนาฬิกาไมโครคอนโทรลเลอร์ตัวหนึ่งไปยังอีกตัวหนึ่ง	20
รูปที่ 2.10	รูปแบบการจัดบิตและความหมายรีจิสเตอร์ที่สเอนไทของซีพียู	21
รูปที่ 2.11	ไมโครคอนโทรลเลอร์ 68HC11 ที่ทำงานแบบมัลติโปรเซสเซอร์	26
รูปที่ 2.12	แบบจำลองของวงจรการสื่อสารที่ใช้ RS-232 C	42
รูปที่ 2.13	หัวต่อตัวเมียของ RS-232 C	42
รูปที่ 2.14	ระดับแรงดันของสัญญาณ RS-232 C ที่กำหนดให้ใช้งาน	44
รูปที่ 2.15	วงจรอินเตอร์เฟส RS-232 C	45
รูปที่ 2.16	การควบคุมคลื่นพาห์ในระบบฮาล์ฟดูเฟล็กซ์	48
รูปที่ 2.17	การควบคุมคลื่นพาห์ของระบบฟูลดูเฟล็กซ์	50
รูปที่ 2.18	การสลับสถานะแบบที่มีช่องย้อนกลับ	51
รูปที่ 2.19	DTE การควบคุมการตอบรับและหยุดการติดต่อการเรียกทางโทรศัพท์	52
รูปที่ 2.20	รูปแบบข้อมูลเนื้อความแอสกีแบบซิงโครนัส	53
รูปที่ 2.21	บล็อกไดอะแกรมของการส่งระบบซิงโครนัส	54
รูปที่ 2.22	การเลือกขา trans clock ของ ซิงโครนัสโมเด็ม	55
รูปที่ 3.1	บล็อกไดอะแกรมของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11	57
รูปที่ 3.2	วงจรเชื่อมต่อ 68HC11 กับหน่วยความจำภายนอก	60
รูปที่ 3.3	การจัดหน่วยความจำของบอร์ด	61
รูปที่ 3.4	การเลือกจัมเปอร์ของแรม	62
รูปที่ 3.5	การเลือกจัมเปอร์ของอีพ롬	63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	หน้า
รูปที่ 3.6 แอดเดรสของพอร์ตอินพุตเอาท์พุต 8255	63
รูปที่ 3.7 การเลือกลักษณะของคีย์บอร์ดโดยใช้ J5	64
รูปที่ 3.8 ลักษณะการเชื่อมต่อคอนเนคเตอร์ 20 ขา เข้ากับคอนเนคเตอร์ เซ็นทรอนิกส์ขนาด 36 ขา ของเครื่องพิมพ์	64
รูปที่ 3.9 การเลือก J6 เพื่อเลือกชนิดของโมดูลแอลซีดี	65
รูปที่ 3.10 การต่อแรงดันอ้างอิงโดย A/D โดย J9	65
รูปที่ 3.11 การเลือกโหมดการทำงานของ 68HC11	66
รูปที่ 3.12 การเลือกจัมเปอร์ J10	66
รูปที่ 3.13 การเลือกจัมเปอร์ J11	67
รูปที่ 3.14 การเชื่อมต่อ 68HC11 เพื่อส่งข้อมูลอนุกรม (SCI) โดยการต่อ พอร์ตอนุกรมกับคอนเนคเตอร์ DB25	67
รูปที่ 3.15 กราฟสำหรับการเลือกค่าความต้านทาน Rz	68
รูปที่ 3.16 จัมเปอร์ J7	68
รูปที่ 3.17 การขยายพอร์ตเพิ่มเติมบนชุดควบคุม	69
รูปที่ 3.18 วงจรที่ใช้เลือกไอซี 8255 หมายเลข 1 และหมายเลข 2	70
รูปที่ 3.19 การต่อวงจรส่วนคีย์บอร์ด	71
รูปที่ 3.20 วงจรแสดงผล 7 เซกเมนต์และแอลอีดีแฟล็ก	72
รูปที่ 3.21 วงจรเชื่อมต่อกับแอลซีดีขนาด 16 ตัวอักษร 1 แถว	72
รูปที่ 3.22 วงจรการแสดงผลทางลำโพง	74
รูปที่ 3.23 วงจรแสดงผลทางแอลอีดี	75
รูปที่ 3.24 วงจรเชื่อมต่อเครื่องพิมพ์	75
รูปที่ 3.25 วงจรวัดแรงดันไฟตรง 0-5 โวลต์ 8 ช่อง	76
รูปที่ 3.26 วงจรรักษาระดับแรงดันไฟตรงในบอร์ดควบคุม	76
รูปที่ 3.27 วงจรรักษาระดับแรงดันไฟตรงในบอร์ดแสดงผล	77
รูปที่ 3.28 วงจรแปลงไฟสลับเป็นไฟตรง	77
รูปที่ 3.29 วงจรหรีไฟควบคุมด้วยไมโครคอนโทรลเลอร์	78
รูปที่ 3.30 ชุดควบคุม CP-68HC11	79
รูปที่ 3.31 ชุดแสดงผลเอาท์พุตและคีย์บอร์ด	79
รูปที่ 3.32 ชุดแหล่งจ่ายไฟ	80
รูปที่ 3.33 ชุดอินเตอร์เฟซและถอดรหัสพอร์ต	80
รูปที่ 3.34 ชุดบอร์ดทดลอง	81

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปภาพ	หน้า
รูปที่ 3.35 ลักษณะโดยสมบูรณ์ของมัลติบอร์ด	81
รูปที่ 4.1 รูปของจอแสดงผลเมื่อประมวลผลโปรแกรมบนเครื่องคอมพิวเตอร์	88
รูปที่ 4.2 ตัวอักษรที่ปรากฏเมื่อจ่ายไฟให้กับมัลติบอร์ด	93
รูปที่ 4.3 ข้อความที่ให้เลือกโหมดการทำงาน	93
รูปที่ 4.4 การทำงานบนมัลติบอร์ด	94
รูปที่ 4.5 การเชื่อมต่อเพื่อทำงานบนเครื่องคอมพิวเตอร์	94



บทที่ 1

บทนำ

ปัจจุบันนี้จะเห็นว่าไมโครคอนโทรลเลอร์ถูกนำมาใช้ในอุปกรณ์เครื่องใช้ไฟฟ้าต่าง ๆ มากมาย ตลอดจนระบบควบคุมในงานอุตสาหกรรม เนื่องจากการใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานและการตัดสินใจในสภาวะต่าง ๆ ของวงจรอิเล็กทรอนิกส์นั้น จะเป็นระบบควบคุมที่สามารถยืดหยุ่นได้โดยการเปลี่ยนแปลงทางซอฟต์แวร์เท่านั้น นอกจากนี้ยังลดความยุ่งยากจากการออกแบบวงจรเป็นผลให้การเปลี่ยนแปลงการทำงานได้ง่ายขึ้น

จากที่กล่าวมาแล้วการที่เราจะสามารถนำไมโครคอนโทรลเลอร์ไปใช้ในการควบคุมดังกล่าวนั้น เราจะต้องมีความเข้าใจการทำงานของไมโครคอนโทรลเลอร์อย่างละเอียดเพื่อการนำไปประยุกต์ใช้งานได้อย่างมีประสิทธิภาพต่อไป

คณะผู้จัดทำเห็นว่าไมโครคอนโทรลเลอร์เบอร์ 68HC11 นั้นเป็นชิพที่มีประสิทธิภาพในการทำงานสูงสามารถประยุกต์ใช้งานต่าง ๆ ได้อย่างกว้างขวาง เนื่องจากโครงสร้างและคุณสมบัติของไมโครคอนโทรลเลอร์ 68HC11 เหมาะสำหรับระบบควบคุมโดยทั่วไปมากทางคณะผู้จัดทำโครงการจึงนำไมโครคอนโทรลเลอร์มาสร้างเป็นมัลติบอร์ด เพื่อศึกษาถึงการทำงานของ 68HC11 โดยจะศึกษาจากรายละเอียดในปริิฏยานิพนธ์ได้ต่อไป

ปริิฏยานิพนธ์มีเนื้อหาทั้งหมด 5 บท ดังต่อไปนี้

บทที่ 1 บทนำจะกล่าวถึงรายละเอียดของปริิฏยานิพนธ์ในแต่ละบทอย่างสังเขป

บทที่ 2 ทฤษฎีและหลักการจะนำเสนอเนื้คหาต่าง ๆ ที่นำมาใช้อ้างอิงและได้ใช้เป็นแนวทางในการออกแบบมัลติบอร์ดทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ การออกแบบและการสร้างจะอธิบายถึงขั้นตอนการออกแบบวงจรในแต่ละส่วนตลอดจนการทำงานของวงจร

บทที่ 3 การออกแบบและการสร้าง จะอธิบายถึงขั้นตอนในการออกแบบของวงจรในแต่ละส่วนตลอดจนการทำงานของวงจรของมัลติบอร์ด

บทที่ 4 การทดลองและผลการทดลอง ในบทนี้เป็นการนำเสนอโปรแกรมทดลองและเอาผลที่ได้จากการทดลองและให้มัลติบอร์ดมาบันทึกไว้เป็นตัวอย่าง

บทที่ 5 บทวิจารณ์ สรุป และแนวทางในการพัฒนา ในบทนี้มุ่งชี้แนะแนวทางในการแก้ปัญหาต่าง ๆ ที่เกิดขึ้นนับตั้งแต่เริ่มศึกษาปริิฏยานิพนธ์จนกระทั่งเสร็จสมบูรณ์เป็นชิ้นงานและปริิฏยานิพนธ์ พร้อมทั้งเสนอแนวทางพัฒนาเพื่อให้มัลติบอร์ดมีประสิทธิภาพสูงต่อไป

บทที่ 2
ทฤษฎี และหลักการ

2.1 68HC11 ไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่ใช้เทคโนโลยีไฮสปีดซีมอส

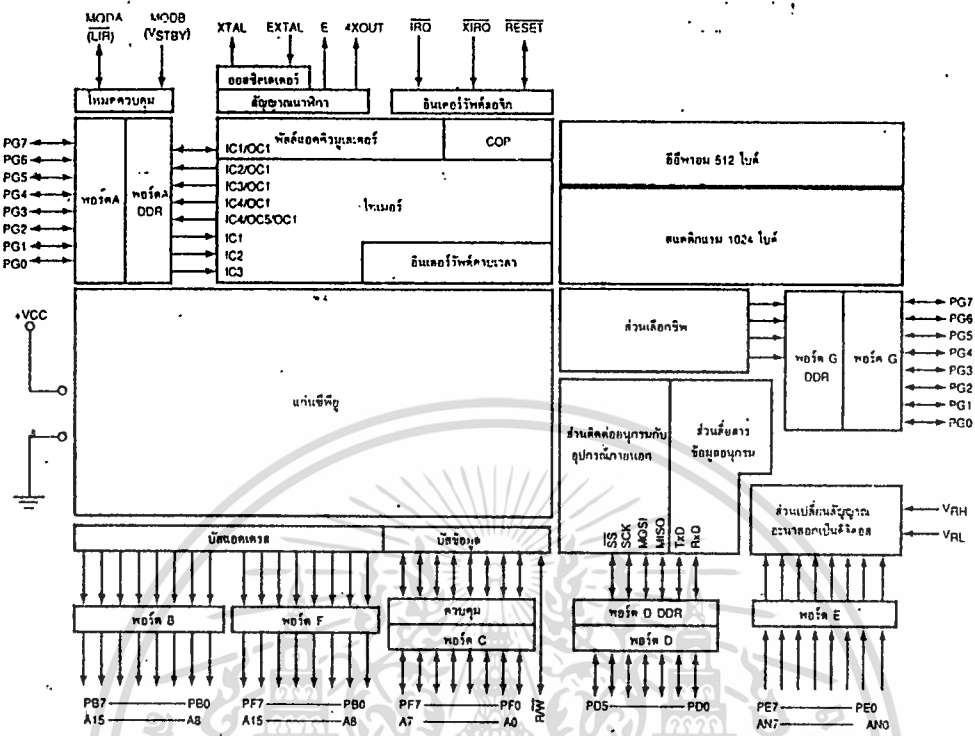
ในการออกแบบไมโครคอมพิวเตอร์เพื่อใช้ในงานควบคุม หรือใช้ในงานเฉพาะอย่าง โดยส่วนใหญ่จะใช้ไอซีประเภทไมโครคอนโทรลเลอร์ เพราะจะมีการทำงานที่รวดเร็วและเที่ยงตรงแม่นยำ นอกจากนี้ยังสามารถเปลี่ยนเงื่อนไขของงาน โดยเราเพียงแต่เปลี่ยนซอฟต์แวร์เท่านั้น เหตุผลอีกประการหนึ่งที่นิยมใช้ไอซีไมโครคอนโทรลเลอร์ คือ มีการรวมอุปกรณ์สนับสนุนทางด้านอินพุต เอาท์พุตและอินเตอร์เฟซต่าง ๆ ไว้ในตัวอย่างพร้อมมูล

68HC11 เป็นไมโครคอนโทรลเลอร์อีกตัวหนึ่งของบริษัทโมโตโรล่า ที่ถูกออกแบบมา โดยรวบรวมอุปกรณ์ที่จำเป็นในการใช้งานไว้ภายในตัวของมันเอง ได้แก่ อินพุต, เอาท์พุต, วงจรตั้งเวลา นอกจากนี้ยังมีวงจรมีวงจรรอสัญญาณแบบอนาล็อก ระบบอินเตอร์รัพต์แบบเวคเตอร์อินเตอร์รัพต์ และระบบป้องกันความผิดพลาดที่จะเกิดขึ้นจากโปรแกรม

2.1.1 โครงสร้างของ 68HC11

68HC11 เป็นชิพที่บริษัทโมโตโรล่าใช้เทคโนโลยีใหม่ผลิตขึ้นมา เพื่อลดการใช้กำลังไฟฟ้าและเพิ่มความหนาแน่นของจำนวนทรานซิสเตอร์ภายในพื้นที่ชิพสูงกว่าชิพเบอร์ 68000 ของโมโตโรล่าเดิม ไมโครคอนโทรลเลอร์ 68HC11 เป็นแบบ VLSI CMOS จัดอยู่ในกลุ่มชิพยูนิบัส 8 บิต รุ่นแรก ๆ ของ 68HC11 พัฒนาในรูปแบบของไมโครคอมพิวเตอร์ชิพเดี่ยวที่มีจำนวนขาเพียง 40 ขา โดยมีขาที่เป็นอินพุตเอาท์พุต 18 ขา การจัดสัญญาณที่ขายังเป็นระบบมัลติเพล็กซ์เพื่อให้สามารถเชื่อมต่อหลายระบบไปได้อีก โดยจะมีขีดความสามารถในการอ้างแอดเดรส 64 กิโลไบต์ หรือให้แอดเดรสบัสมีขนาด 16 เส้น บัสข้อมูลมีขนาด 8 เส้น

ชิพยูนิบัสของ 68HC11 พัฒนามาจาก 6801 จึงใช้รหัสคำสั่งทั้งหมดของ 6801 ได้โดยตรง สัญญาณต่าง ๆ ของระบบก็เหมือนของ MC6801 เมื่อนำ 6801 มาใช้ร่วมกับอุปกรณ์อื่น ๆ ภายในชิพก็ได้มีการพัฒนาให้สามารถใช้คำสั่งใหม่ ๆ เพิ่มขึ้นได้อีก 91 คำสั่ง การทำงานจะทำได้กับความถี่ของสัญญาณนาฬิกาตั้งแต่ 0 คือ ไฟตรงไปจนถึงสูงสุด 4 MHz และในปัจจุบันนี้ยังมีรุ่นที่สามารถทำงานได้สูงกว่านี้อีก โดยมีลักษณะโครงสร้างภายในของไมโครคอนโทรลเลอร์ 68HC11 แสดงได้เป็นบล็อกไดอะแกรมดังรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรมของ 68HC11F1

การที่โมโตโรล่าทำการพัฒนาจาก 6801 โดยมีการเพิ่มอินเด็กซ์รีจิสเตอร์ Y ขนาด 16 บิตเข้าไป ทำให้ได้คำสั่งใหม่เพิ่มขึ้นจากเดิมถึง 65 คำสั่ง และเพื่อให้ได้ลักษณะการทำงานที่เป็นไมโครคอนโทรลเลอร์ที่สมบูรณ์นั้น จึงมีการเพิ่มคำสั่งที่กระทำต่อบิต โดยให้มีความสามารถเซตหรือเคลียร์บิตใด ๆ ในหน่วยความจำขนาด 64 กิโลไบต์ที่แอดเดรสใดก็ได้ และยังจะสามารถตรวจสอบบิตใด ๆ ใน 64 กิโลไบต์นี้ เพื่อกำหนดเป็นเงื่อนไขในการกระโดดไปยังตำแหน่งต่าง ๆ

ขีดความสามารถของคำสั่งที่เพิ่มเติมจาก 6801 นั้นยังมีอีกหลายจุด เช่น การเพิ่มคำสั่งหารตัวเลข 16 บิต กับ 16 บิต เพื่อให้ได้ผลลัพธ์เป็นตัวเลขขนาด 16 บิต และเศษเหลือขนาด 16 บิตด้วย คำสั่งเหล่านี้กระทำการหารตัวเลขแบบจำนวนเต็ม คำสั่งหารมีขีดความสามารถพิเศษอยู่ บางคำสั่งใช้สำหรับการหารเลขเศษส่วน โดยตั้งข้อสมมติว่าตัวตั้งมีค่าน้อยกว่าตัวหาร ซึ่งผลลัพธ์ที่ได้จะมีค่าน้อยกว่า 1 ซึ่งนำมาใช้ในลักษณะเลขจุด ซึ่งมีค่าจาก 0.0 ถึง 0.99998 ตัวเลขที่เป็นจลศนิยมนั้นเหมาะที่จะนำมาใช้กับวงจร A/D และ D/A ซึ่งได้ตัวเลขในรูปแบบที่เป็นเปอร์เซ็นต์นั่นเอง

คำสั่งใหม่ใน 68HC11 ยังมีคำสั่งเกี่ยวกับการเปรียบเทียบตัวเลขขนาด 16 บิต ที่ใช้สำหรับการทดสอบและแลกเปลี่ยนข้อมูลในอินเด็กซ์รีจิสเตอร์ ซึ่งเป็นสิ่งที่มักพบสำหรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้เขียนโปรแกรมควบคุมอุปกรณ์ต่าง ๆ มาก โดยจากสถาปัตยกรรมที่ก้าวหน้าขึ้นทำให้ชิพยูนิต 68HC11 สามารถทำงานได้เร็วกว่า 6801 ถึงเท่าตัวในกรณีที่ใช้สัญญาณนาฬิกาเท่ากัน โดยเราสามารถสรุปคุณสมบัติต่าง ๆ ของ 68HC11 ได้ดังต่อไปนี้

2.1.2 คุณสมบัติของ 68HC11

- เป็นชิพยูนิตขนาด 8 บิต
- มีหน่วยความจำภายในที่ใช้สำหรับเก็บโปรแกรมชนิดรอมขนาดความจุ 4, 8 หรือ 12 กิโลไบต์
- มีหน่วยความจำภายในที่มีไว้สำหรับใช้เก็บโปรแกรมชนิดอีพีรอมขนาด 512 ไบต์ หรือ 2 กิโลไบต์
- มีหน่วยความจำภายในที่มีไว้สำหรับเก็บโปรแกรมชนิดแรมมีขนาด 192,256 หรือ 512 ไบต์
- มีวงจรตั้งเวลา/วงจรมับขนาด 16 บิต
- มีวงจรมับพัลส์ขนาด 8 บิต
- มีวงจรสำหรับการรับส่งข้อมูลอนุกรมได้สองทิศทาง (Universal Synchronous Receiver Transmitter : USRT)
- ขนาดที่ใช้ในการติดต่อสำหรับวงจรการส่งข้อมูลอนุกรมเป็นแบบมัลติโปรเซสเซอร์
- มีวงจรเปลี่ยนสัญญาณจากอนาล็อกเป็นดิจิทัล 8 ช่อง
- มีวงจรไทม์อินเตอร์รัพต์
- มีระบบวอตช์ด็อกที่ตั้งเวลาได้
- มีวงจรตรวจสอบสัญญาณนาฬิกาให้กับชิพยูนิต
- มีระบบอินเตอร์รัพต์ 2 ระดับจาก 21 แหล่ง
- สามารถติดต่อกับหน่วยความจำภายนอกได้ 64 กิโลไบต์

2.1.3 ตระกูลของ 68HC11

68HC11 เป็นไมโครคอนโทรลเลอร์ที่แบ่งออกเป็นหลายรุ่น โดยแบ่งตามลักษณะของหน่วยความจำที่อยู่ภายในชิพยูนิต ดังตารางที่ 2.1

เบอร์ชิพ	รวม	อีอีพรอม	แรม	วีซีเคอร์ CONFIG	หมายเหตุ
MC68HC11A8	8 k	512	256	\$0F	เป็นควมมาตรฐานของชิพตระกูลนี้
MC68HC11A1	0	512	256	\$0D	เหมือน MC68HC11A8 แต่ไม่มีรวม
MC68HC11A0	0	0	256	\$0C	เหมือน MC68HC11A8 แต่ไม่มีทั้งรวม และ อีอีพรอม
XC68HC11B8	8 k	512*	256	\$0F	เป็นรุ่นทดลองรุ่นแรก
XC68HC11B1	0	512*	256	\$CD	เหมือน XC68HC11B8 แต่ไม่มีรวม
XC68HC11B0	0	0	256	\$0C	เหมือน XC68HC11B8 แต่ไม่มีทั้ง ROM และ RAM
MC68HC11E9	12 k	512	512	\$0F	มี 4 แคนเปเจอร์อินพุต แรมขนาดใหญ่ และ รวม 12 กิโลไบต์
MC68HC11E1	0	512	512	\$0D	เหมือน MC68HC11E9 แต่ไม่มีรวม
MC68HC11E0	0	0	512	\$0C	เหมือน MC68HC11E9 แต่ไม่มีทั้งรวม และ อีอีพรอม
MC68HC11E2	0	2 k**	256	\$FF	ไม่สามารถเพิ่มรวมได้
MC68HC11D3	4 k	0	192	N/A	เป็นรุ่นประหยัดไม่มีเอทวูคเคอร์ และหน่วยความจำขนาดเล็ก

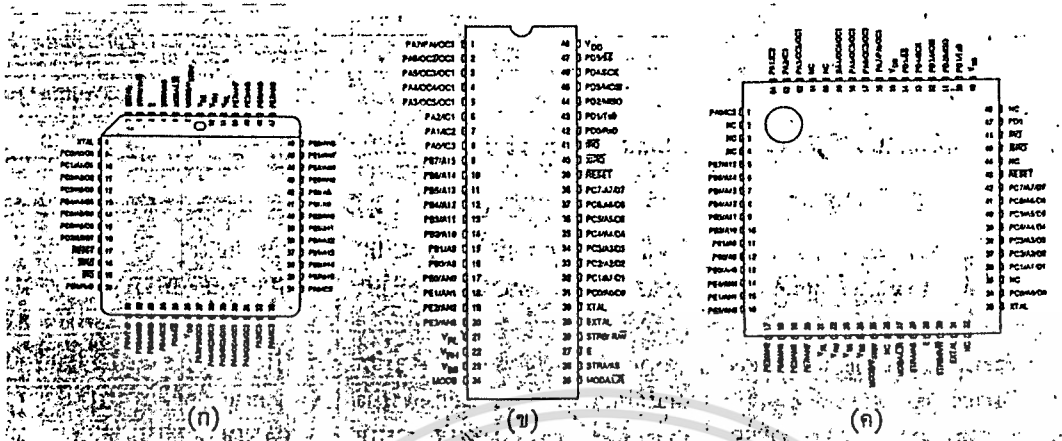
ตารางที่ 2.1 ตระกูลของ 68HC11

รูปแบบของชิพตระกูลนี้จะมีด้วยกัน 3 แบบ คือ

- PLCC (Plastic Leaded Chip Carrier) เป็นตัวถังรูปสี่เหลี่ยม 52 ขา
- DIP (Dual In-line Package) จะมีขนาด 48 ขา ในตัวถังแบบนี้จะไม่มีขา

สัญญาณของพอร์ต PE4-PE7

- QFP (Quad Flat Pack) เป็นตัวถังรูปสี่เหลี่ยมขนาด 64 ขา ดังรูปที่ 2.2



รูปที่ 2.2 แบบตัวถังและการจัดขา

- ก) ตัวถังแบบ PLCC
- ข) ตัวถังแบบ DIP
- ค) ตัวถังแบบ GFP

2.1.4 โหมดการทำงานของ 68HC11

68HC11 มีโหมดการทำงาน 4 โหมด ดังนี้

1. โหมดซิงเกิลชิป (Single chip operating mode)
2. โหมดมัลติเพล็กซ์ขยาย (Expanded multiplexed operating mode)
3. โหมดบูตสแตรปพิเศษ (Special bootstrap operating mode)
4. โหมดทดสอบพิเศษ (Special test operating mode)

โดยเราสามารถเลือกโหมดการทำงานได้โดยการกำหนดสถานะที่ขา MODA และ MODB ดังตารางที่ 2.2

MODB	MODA	โหมดการทำงาน
1	0	ซิงเกิลชิพ (Single chip)
1	1	มัลติเพล็กซ์ขยาย (Extended multiplexed)
0	0	บูตสแตร็ปพิเศษ (Special bootstrap)
0	1	ทดสอบพิเศษ (Special test)

ตารางที่ 2.2 การเลือกโหมดการทำงานของ 68HC11

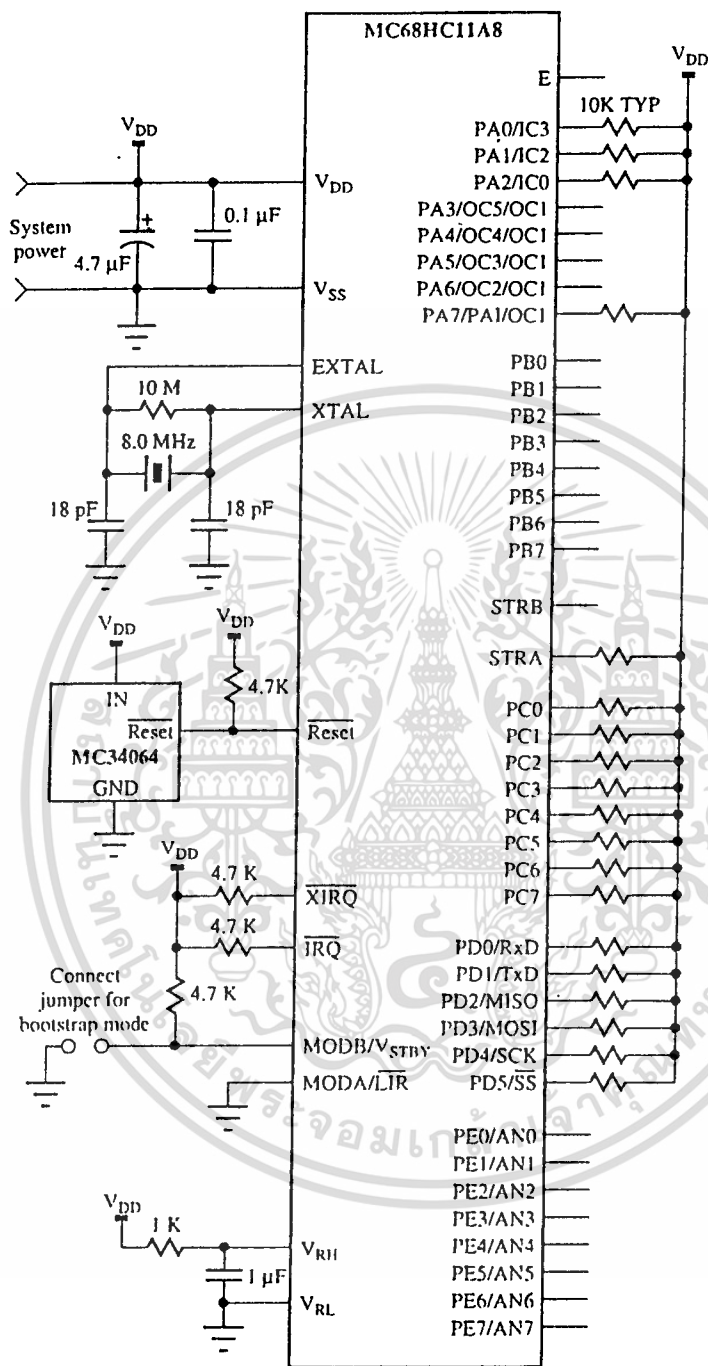
2.1.4.1 โหมดซิงเกิลชิพ

ในโหมดนี้ 68HC11 จะทำงานโดยลำพังตัวเดียวจากที่ได้กล่าวมาแล้วที่ว่าภายใน 68HC11 มีหน่วยความจำและพอร์ตถึง 5 พอร์ต ดังนั้นเมื่อทำงานโหมดซิงเกิลชิพตัวชิพก็จะทำการเรียกข้อมูลที่อยู่ในหน่วยความจำภายในออกมาใช้งาน และใช้พอร์ตที่มีอยู่รับหรือส่งค่าออกไป

ดังนั้นในโหมดนี้ 68HC11 จะไม่ต้องการหน่วยความจำภายนอก หรือชิพพอร์ตอินพุตเอาต์พุตภายนอก เมื่อใช้งานในโหมดนี้จะช่วยลดค่าใช้จ่ายลงได้มาก เพราะไม่ต้องมีชิพอื่น ๆ ภายนอกสับสนุนเลย และนอกจากนี้ยังมีความเชื่อถือสูง เพราะแทบจะไม่มีจุดต่อสัญญาณภายนอกเลย ทำให้โอกาสที่จะเกิดความผิดพลาดลดลง แต่อย่างไรก็ตามในโหมดนี้ 68HC11 จะใช้หน่วยความจำและพอร์ตภายในชิพ ทำให้ไม่สามารถเพิ่มหน่วยความจำหรือพอร์ตเพื่อขยายระบบได้ การต่อวงจรของ 68HC11 เมื่อทำงานในโหมดนี้แสดงดังรูปที่ 2.3

2.1.4.2 โหมดมัลติเพล็กซ์ขยาย

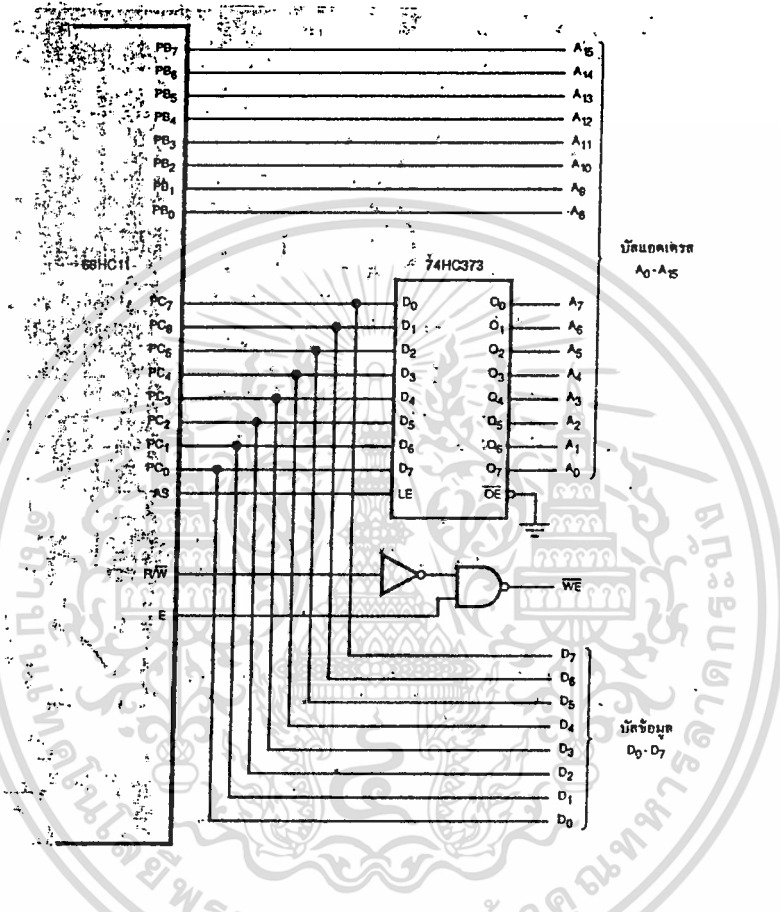
การทำงานของ 68HC11 ในโหมดนี้จะแตกต่างกับโหมดซิงเกิลชิพ โดยโปรแกรมการควบคุมการทำงานของชิพจะอยู่ในหน่วยความจำภายนอก โดยที่พอร์ต B และ C ของ 68HC11 จะทำหน้าที่เป็นบััสแอดเดรสและข้อมูล สิ่งเกิดได้จากบล็อกไดอะแกรมภายในของ 68HC11 ในรูปที่ 2.4 พอร์ต B จะทำหน้าที่เป็นบััสแอดเดรสตำแหน่งสูง และพอร์ต C ทำหน้าที่เป็นบััสแอดเดรสต่ำและบััสข้อมูล



รูปที่ 2.3 วงจรการต่อ 68HC11 เมื่อทำงานในโหมดซิงเกิลชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

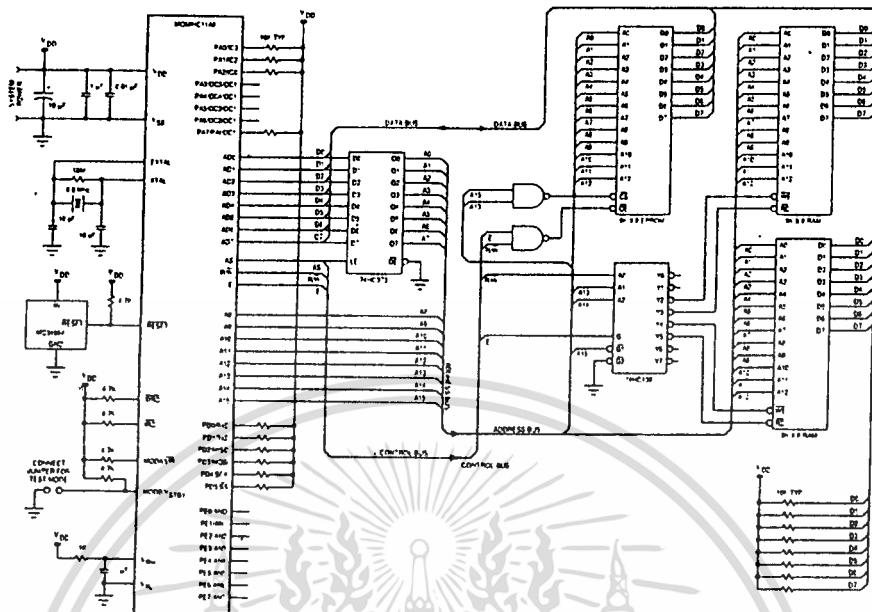
การที่พอร์ต C จะสามารถเป็นได้ทั้งบัสน์แอดเดรสและเป็นบัสน์ข้อมูลได้นั้น จะต้องใช้กระบวนการที่เรียกว่า การมัลติเพล็กซ์ (Multiplexed) การแยกสัญญาณแอดเดรสและข้อมูลมาใช้งานจะต้องต่อวงจรเพื่อทำการมัลติเพล็กซ์ไว้ที่ภายนอก ดังวงจรในรูปที่ 2.4



รูปที่ 2.4 วงจรมัลติเพล็กซ์บัสน์แอดเดรสและบัสน์ข้อมูลเมื่อกำหนดการทำงานในโหมดมัลติเพล็กซ์ขยาย

สัญญาณควบคุมที่ใช้มัลติเพล็กซ์สัญญาณแอดเดรสและข้อมูล ได้แก่สัญญาณ STROBE B/R/พื และสัญญาณ STROBE A/ADDRESS STROBE (STRA/AS) ในโหมดนี้ชิพ 68HC11 จะสามารถติดต่อกับหน่วยความจำภายนอกเพิ่มได้สูงสุดถึง 64 กิโลไบต์ แต่ก็มีข้อเสียคือ เราจะต้องต่ออุปกรณ์เพิ่มอีก เช่น หน่วยความจำ ชิพพอร์ตอินพุทเอาต์พุท ทำให้ค่าใช้จ่ายของระบบสูงขึ้น แต่อย่างไรก็ตามในโหมดนี้เป็นโหมดที่นิยมใช้งานมากที่สุด เพราะสามารถต่อขยายหน่วยความจำและพอร์ตได้มากมายนั่นเอง กรณีที่ต้องการต่อ 68HC11 ให้ทำงานในโหมดมัลติเพล็กซ์ขยายแสดงในรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ตัวอย่างของวงจรการต่อใช้งาน 68HC11 เมื่อต้องการให้ทำงานในโหมดมัลติเพล็กซ์ขยาย

2.1.4.3 โหมดบัสแคสเตร์พิเศษ

เป็นโหมดการทำงานที่คล้ายกับโหมดซิงเกิลชิพ คือตัวไมโครคอนโทรลเลอร์สามารถทำงานได้โดยลำพัง แต่จะมีข้อพิเศษกว่าแบบซิงเกิลชิพอยู่หลายประการ โดยสามารถเขียนโปรแกรมแล้วโหลดเข้าไปในแรมภายในชิพได้เลย โปรแกรมที่ใช้ในการบูตจะถูกเก็บไว้ในบัสแคสเตร์ประมาณจำนวน 192 ไบต์ รอมนี้จะสามารถเรียกออกมาใช้งานได้ก็ต่อเมื่อ 68HC11 ทำงานในโหมดบัสแคสเตร์เป็นเท่านั้น โดยจะโปรแกรมบูตนี้เก็บไว้ที่แอดเดรส \$BF40-\$BFFF เมื่อเรียกใช้งานโปรแกรมบูต ส่วน SCI (Serial Communication Interface) จะส่งข้อมูลโปรแกรมไปยังแรมภายในชิพที่แอดเดรส \$0000 - \$00FF หลังจากส่งข้อมูลที่แอดเดรส \$00FF เรียบร้อยแล้วชิพจะเริ่มทำงานที่แอดเดรส \$0000 ต่อไป

ในโหมดการทำงานแบบนี้ 68HC11 จะส่งผ่านข้อมูลผ่านพอร์ต SCI โดยหลังจากที่ทำการเซตชิพแล้ว ส่วน SCI จะทำงานที่สัญญาณนาฬิกา E/16 (หรือมีอัตราบอดเท่ากับ 7,812 สำหรับสัญญาณนาฬิกาที่มีความถี่ 2 MHz) โหมดนี้ยังมีลักษณะการทำงานที่ค่อนข้างพิเศษอีกประการหนึ่งคือ การป้องกันการคัดลอก โดยจะมีบิตป้องกัน (Security bit) เข้ามาเกี่ยวข้อง ถ้าหากบิตป้องกันนี้ถูกเซตเป็น "1" ที่เอาท์พุทของส่วน SCI จะส่งค่า \$FF ออกไป จะทำให้ข้อมูลในอีพ็รอมถูกลบ และจะทำให้เหมือนนั้นจนกว่าจะทำการลบข้อมูลในอีพ็รอมภายในชิพหมด แต่ในทางตรงข้ามถ้าไม่มีการเซตบิตป้องกันข้อมูลที่เป็นการหยุดการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้เฉพาะในท้องถิ่นเท่านั้น ไม่อนุญาตให้มีการนำออกจำหน่าย การค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงาน วงจรชีพจะถูกส่งออกไปภายนอกโดยจะผ่านส่วน SCI

ผู้ใช้งานต้องทำการดาวน์โหลดข้อมูลโปรแกรม 256 ไบต์ ไปยังแรมภายในโดยให้เริ่มทำงานที่แอดเดรส \$0000 ถ้าทุกอย่างเรียบร้อยก็จะไปเริ่มทำงานที่ตำแหน่ง \$0000 เพื่อเตรียมที่จะทำงาน

อินเตอร์รัพต์เวคเตอร์ของ 68HC11 เมื่อถูกใช้งานในโหมดนี้ จะสามารถเข้าถึงหน่วยความจำแรมได้เร็วขึ้น ซึ่งในการใช้อินเตอร์รัพต์จะทำให้ผู้ใช้งานสามารถกระโดดไปทำงานที่ส่วนอื่นได้ อินเตอร์รัพต์เวคเตอร์ที่มีใน 68HC11 สรุปได้ดังตารางที่ 2.3

ถ้าหากต้องการใช้งานอินเตอร์รัพต์ SWI ต้องระบุคำสั่งกระโดดเพื่อให้ชิพสามารถเรียกใช้งานได้ถูกลงในแรมที่ตำแหน่ง \$00F4, \$00F5 และ \$00F6 เมื่อเกิดอินเตอร์รัพต์แบบ SWI ขึ้น ค่าเวคเตอร์นี้เป็นตัวบอกให้ชิพทำการอ่านข้อมูลเพื่อประมวลผลที่ \$00F4 ในแรมและที่นั่นจะมีคำสั่ง JUMP เพื่อให้ชิพสามารถตอบสนองการอินเตอร์รัพต์ได้

ลักษณะพิเศษอีกอย่างหนึ่งที่ไม่ควรมองข้าม คือ สามารถที่จะเข้าไปทำงานในรอมที่แอดเดรสตำแหน่ง \$0000 ได้โดยไม่ต้องทำการดาวน์โหลดก่อน โดยวิธีการนี้จะใช้ได้กับเฉพาะเมื่อเราใช้สัญญาณนาฬิกา E/16 เริ่มด้วยการส่ง \$55 ไปแทนค่า \$FF ด้วยคำสั่งนี้จะทำให้ชิพกระโดดไปทำงานที่แอดเดรส \$0000 โดยที่ไม่ต้องผ่านการรวดโหลดได้ ลักษณะการทำงานโหมคนั้นแสดงในรูปที่ 2.6 แต่เนื่องจากในเบอร์ MC68HC11A1 ไม่มีรอมจึงไม่สามารถใช้งานในโหมคนั้นได้ สมบูรณ์ตามฟังก์ชันที่ระบุไว้ข้างต้น

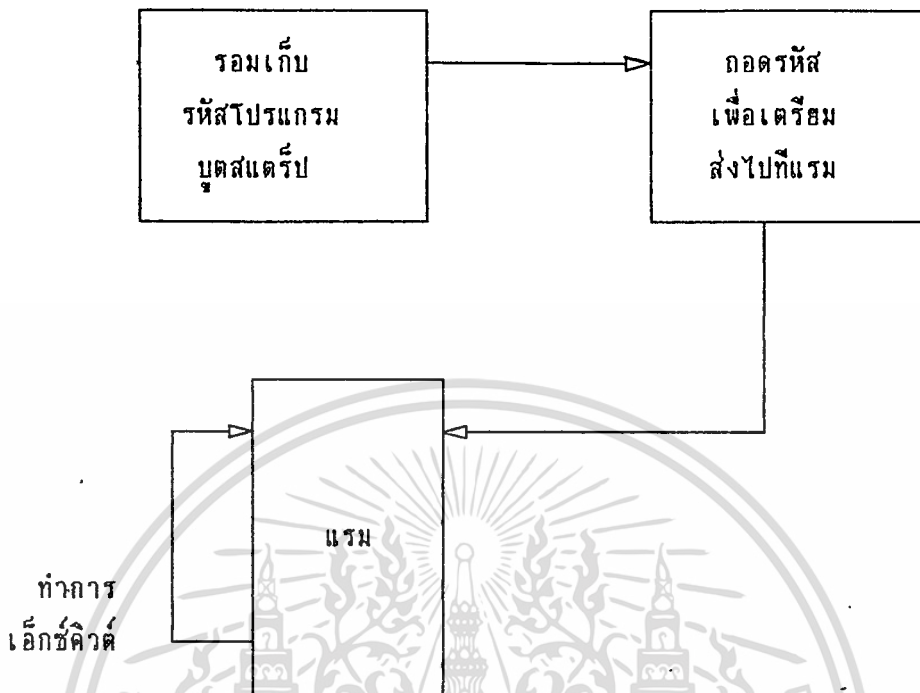
2.1.4.4 โหมคทดสอบพิเศษ

เป็นโหมคการทำงานที่โรงงานผู้ผลิตเป็นผู้กำหนด ซึ่งการทำงานในโหมคนี้จะคล้ายกับโหมคมีติเฟล็กซ์ชยา ซึ่งในการเซ็ตและอินเตอร์รัพต์เวคเตอร์จะได้รับการเพคท์จากหน่วยความจำภายนอกที่แอดเดรส \$BFC0-\$BFFF มากกว่าที่แอดเดรส \$FFC0-\$FFFF รีจิสเตอร์ TEST1 จะได้รับการอีนาเบิล เพื่อเป็นการบ่งบอกให้ทราบว่าจะขณะนี้ชิพพร้อมรับการตรวจสอบแล้ว โหมคการทำงานนี้เป็นโหมคที่ผู้ใช้งานปกติไม่ควรที่จะเลือกใช้เพราะจะทำให้ระบบป้องกันข้อมูลภายในชิพด้อยลงไป

แอดเดรส	เวคเตอร์
\$00C4	SCI
\$00C7	SPI
\$00CA	Pulse Accumulator Input Edge
\$00CD	Pulse Accumulator Overflow
\$00D0	Timer Overflow
\$00D3	Timer Output compare 5
\$00D6	Timer Output compare 4
\$00D9	Timer Output compare 3
\$00DC	Timer Output compare 2
\$00DF	Timer Output compare 1
\$00E2	Timer Input compare 3
\$00E5	Timer Input compare 2
\$00E8	Timer Input compare 1
\$00EB	Real Time Interrupt
\$00EE	IRQ
\$00F1	XIRQ
\$00F4	SWI
\$00F7	Illegal Opcode
\$00FA	COP Fail
\$00FD	Clock Monitor
\$BF40	Reset

ตารางที่ 2.3 อินเทอร์เน็ตเวคเตอร์ในโหมดการทำงานแบบบูตสแตร์ป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 โค้ดแรมการทำงานของ 68HC11 ในโหมดบูตส์แตร์ป

2.1.5 ลักษณะของสัญญาณและการจัดขา

$\overline{\text{RESET}}$ จะเป็นขาที่ทำงานสองทิศทางคือ เป็นอินพุตสำหรับการทำให้ชิพเริ่มต้นการทำงาน และเป็นเอาต์พุตแสดงถึงการทำงานภายในผิดพลาด ซึ่งจะเกิดขึ้นได้ 3 กรณี คือ ความผิดพลาดจากสัญญาณนาฬิกา, ความผิดพลาดที่จากโปรแกรมในการใช้ระบบวอลต์ดีคอก และการเอ็กซ์คิวต์คำสั่งผิดพลาด

XTAL, EXTAL เป็นขาอินพุตสำหรับต่อคริสตอลเพื่อผลิตสัญญาณนาฬิกาให้กับชิพหรืออาจจะใช้สัญญาณนาฬิกาจากภายนอกป้อนเข้าที่ขา EXTAL โดยตรงก็ได้ โดยที่มีตัวต้านทาน 10-100 กิโลโอห์ม ต่อลงกราวด์เพื่อป้องกันสัญญาณรบกวน

E เป็นเอาต์พุต โดยจะมีความถี่สัญญาณนาฬิกาน้อยกว่าความถี่ของสัญญาณนาฬิกาที่ขา EXTAL และขา XTAL 4 เท่า และยังเป็นขาที่แสดงการทำงานของชิพด้วย คือ ถ้ามีสัญญาณเป็นลอจิก "0" แสดงว่าชิพอยู่กำลังทำการประมวลผลภายในและหากเป็นลอจิก "1" หมายถึงขณะนี้ชิพกำลังทำกับข้อมูล เมื่ออยู่ใน STOP MODE จะไม่มีสัญญาณ E ออกมา

$\overline{\text{IRQ}}$ เป็นขาอินพุตทำงานที่ลอจิก "0" หรือขอบขาลง เป็นการขออินเทอร์รัพต์ชิพ โดยที่ขาที่ต้องต่อตัวต้านทานจากภายนอกมีค่า 4.7 กิโลโอห์ม กับไฟเลี้ยง +5 โวลต์ เพื่อให้สภาวะปกติมีค่าเป็นลอจิก "1"

XIRQ เป็นขาอินพุตจะมีระดับการทำงานที่ลอจิก "0" ซึ่งจะเป็นการขออินเตอร์รัพต์แบบนอนมาสเคเบิลอินเตอร์รัพต์

MODA, MODB เป็นขาที่ใช้เลือกโหมดการทำงานของ CPU ซึ่งแสดงการเลือกโหมดดังตารางที่ 2.2 หลังจากที่ยูนิทเลือกโหมดการทำงานแล้ว ขา MODA จะเป็นขาเอาต์พุตทำงานที่ลอจิก "0" เพื่อแสดงถึงการเพ็ชท์ของซีพียูที่จะใช้ในการทำให้ซีพียูทำงานที่ละคำสั่ง และขา MODB เป็นขาอินพุตสำหรับป้องกันไฟไปเลี้ยงหน่วยความจำภายใน เพื่อป้องกันไม่ให้ข้อมูลสูญหายหลังจากที่ไม่มีไฟเลี้ยงซีพียู

VRL, VRH เป็นขาอินพุตสำหรับป้องกันแรงดันอ้างอิงในวงจรแปลงสัญญาณอนาล็อกเป็นดิจิตอล

STROB/R/ \bar{P} เป็นขาเอาต์พุตที่แสดงผลการทำงานในโหมด SINGLE CHIP ขานี้เป็นเอาต์พุตแสดงผลการแฮนด์เชคข้อมูล และเมื่อทำงานในโหมด EXPANDED MULTIPLEXED ขานี้จะเป็นขาเอาต์พุตแสดงผลการอ่านเขียนข้อมูลของซีพียู โดยถ้าเป็นลอจิก "0" แสดงว่าเป็นการเขียนข้อมูล

STRA/AS เป็นทั้งขาอินพุตและเอาต์พุต ถ้าทำงานในโหมด SINGLE CHIP ขานี้จะเป็นขาอินพุตแสดงผลการแฮนด์เชคข้อมูล และเมื่อทำงานในโหมด EXPANDED MULTIPLEXED จะเป็นการแยกสัญญาณระหว่างแอดเดรส 8 บิตล่างกับสัญญาณข้อมูล

PORT A หน้าทีของพอร์ตนี้จะมีหลายแบบด้วยกัน ได้แก่ เป็นขาอินพุตเอาต์พุตปกติ, เป็นขาสัญญาณในระบบตัวตั้งเวลาหลักและแอกคิวเลเตอร์ เป็นต้น ที่พอร์ต A จะมีอยู่ 3 ขาสัญญาณเท่านั้นที่จะถูกกำหนดให้เป็นขาอินพุตได้ ส่วนที่ถูกกำหนดให้เป็นขาเอาต์พุตจะมีอยู่ 4 ขาที่เหลืออีก 1 ขาเป็นขาสัญญาณที่เป็นได้ทั้งอินพุตและเอาต์พุต ขาที่ว่านั้นคือ ขา PA7 ทิศทางการส่งผ่านข้อมูลของ PA7 จะถูกควบคุมโดยข้อมูลในบิตที่ 7 ของรีจิสเตอร์ควบคุมพัลส์แอกคิวเลเตอร์ (PACTL) โดยบิตที่ 7 นั้นมีชื่อว่า DDRA7 ข้อมูลที่ต้องการอ่านและเขียนของพอร์ต A จะถูกเก็บไว้ที่รีจิสเตอร์ PORTA

PORT B ถูกกำหนดให้เป็นพอร์ตเอาต์พุตเมื่อทำงานในโหมดซิงเกิลชิพและบูตสแตร์ปีพิเศษ ข้อมูลที่ต้องการส่งออกที่พอร์ตนี้ ต้องเขียนเข้าไปเก็บไว้ที่รีจิสเตอร์ PROT B และจะส่งออกไปได้ก็ต่อเมื่อมีสัญญาณ STRB เกิดขึ้น นั่นคือในการส่งข้อมูลพอร์ต B นี้จะมีการแฮนด์เชคระหว่างอุปกรณ์ที่ต่ออยู่กับพอร์ตและตัวไมโครคอนโทรลเลอร์ แต่ถ้าหาก 68HC11 ทำงานในโหมดมัลติเพล็กซ์หลายหรือโหมดทดสอบพิเศษ พอร์ต B ทำหน้าที่เป็นบัสแอดเดรสในไบต์สูง (A8-A15) เพื่อติดต่อกับหน่วยความจำภายนอก

PORT C เมื่อทำงานในโหมดซิงเกิลชิพและโหมดบูตสแตร์ปีพิเศษ พอร์ต C จะเป็นพอร์ตอินพุตเอาต์พุต สามารถส่งผ่านข้อมูลได้ 2 ทิศทาง โดยแต่ละขาของพอร์ต (หรือจะเรียกว่าแต่ละบิตก็ได้) จะเป็นอิสระต่อกันในด้านการกำหนดทิศทาง ในการกำหนดทิศทางของขาพอร์ต C แต่ละขานั้นเราสามารถกำหนดได้ที่รีจิสเตอร์ควบคุมทิศทางข้อมูลพอร์ต C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่อนำไปใช้โดยไม่ได้รับอนุญาตเป็นการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุด
คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

หรือที่เรียกว่า (Data Direction for Port C: DDRC) ถ้าบิตใดในรีจิสเตอร์ DDRC ถูกกำหนดให้เป็น "1" ขาของพอร์ต C ที่ตำแหน่งเดียวกันนั้นจะเป็นขาพอร์ตเอาต์พุต และถ้าต้องการให้เป็นอินพุตก็เขียนข้อมูล "0" ไปยังบิตที่ต้องการในรีจิสเตอร์ DDRC ถ้าหาก 68HC11 ทำงานในโหมดมัลติเพล็กซ์ขาหรือที่โหมดทดสอบพิเศษพอร์ต C จะถูกปรับให้ทำหน้าที่เป็นเข้าสัญญาณแอดเดรสไบต์ต่ำ (A0 - A7) และขาบัสข้อมูล (D0-D7) โดยจะสามารถแยกสัญญาณทั้งสองนี้ด้วยการมัลติเพล็กซ์

Port-Bit	Single-Chip and Bootstrap Mode	Expanded Multiplexed and Special Test Mode
A-0	PA0 IC3	PA0 IC3
A-1	PA1 IC2	PA1 IC2
A-2	PA2 IC1	PA2 IC1
A-3	PA3 OC5 and-or OC1	PA3 OC5 and-or OC1
A-4	PA4 OC4 and-or OC1	PA4 OC4 and-or OC1
A-5	PA5 OC3 and-or OC1	PA5 OC3 and-or OC1
A-6	PA6 OC2 and-or OC1	PA6 OC2 and-or OC1
A-7	PA7 PAI and-or OC1	PA7 PAI and-or OC1
B-0	PE0	A8
B-1	PE1	A9
B-2	PE2	A10
B-3	PE3	A11
B-4	PE4	A12
B-5	PE5	A13
B-6	PE6	A14
B-7	PE7	A15
C-0	PC0	A0 D0
C-1	PC1	A1 D1
C-2	PC2	A2 D2
C-3	PC3	A3 D3
C-4	PC4	A4 D4
C-5	PC5	A5 D5
C-6	PC6	A6 D6
C-7	PC7	A7 D7
D-0	PD0 RxD	PD0 RxD
D-1	PD1 TxD	PD1 TxD
D-2	PD2 MISO	PD2 MISO
D-3	PD3 MOSI	PD3 MOSI
D-4	PD4 SCK	PD4 SCK
D-5	PD5 SS	PD5 SS
	STR A	AS
	STR B	R W
E-0	PE0 AN0	PE0 AN0
E-1	PE1 AN1	PE1 AN1
E-2	PE2 AN2	PE2 AN2
E-3	PE3 AN3	PE3 AN3
E-4	PE4 AN4	PE4 AN4
E-5	PE5 AN5	PE5 AN5
E-6	PE6 AN6	PE6 AN6
E-7	PE7 AN7	PE7 AN7

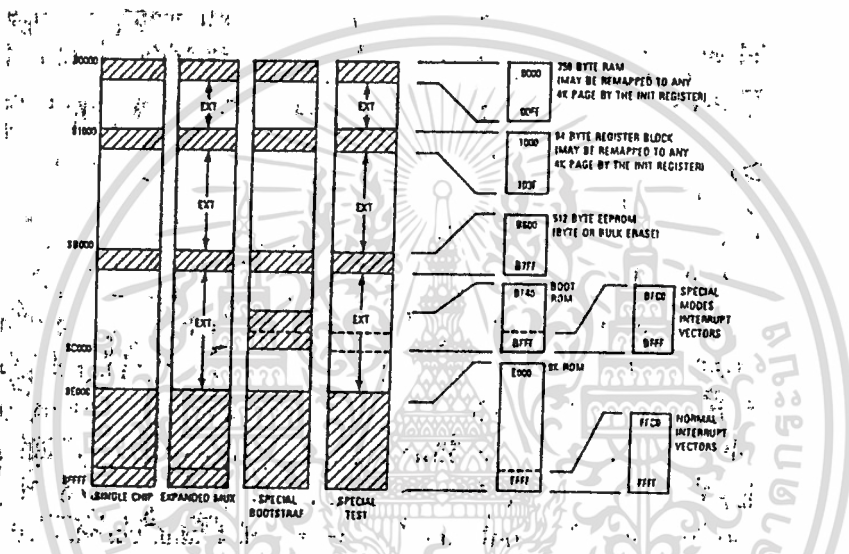
ตารางที่ 2.4 สรุปการใช้ขาของ 68HC11 ในโหมดการทำงานต่างๆ

PORT D เป็นพอร์ตอินพุตเอาต์พุตขนาด 6 บิต PD0 และ PD1 จะถูกกำหนดให้ทำหน้าที่เป็นขารับส่งข้อมูลอนุกรมเมื่อให้ 68HC11 ทำการสื่อสารข้อมูลอนุกรม (SCI) ส่วนอีก 4 ขาที่เหลือจะนำมาใช้ในการเชื่อมต่ออุปกรณ์อนุกรม (SPI) การกำหนดทิศทางการส่งผ่านข้อมูลของพอร์ต ทำได้โดยการเขียนข้อมูลไปรีจิสเตอร์ควบคุมทิศทางข้อมูลพอร์ต D (DDRD) ส่วนข้อมูลที่ต้องการจะส่งเข้ามาหรือส่งออกไปทางพอร์ต D ให้เขียนมาเก็บไว้ที่รีจิสเตอร์ PORTD

PORT E ถูกกำหนดให้เป็นพอร์ตอินพุตขนาด 8 บิต และใช้เป็นพอร์ตอินพุตของวงจรแปลงสัญญาณอะนาล็อกเป็นสัญญาณดิจิทัล (ถ้าเป็นรุ่น 48 ขา พอร์ต E จะมีเพียง 4 ขา)

2.1.6 การจัดหน่วยความจำ

68HC11 เป็นชิพที่มีการจัดหน่วยความจำได้อย่างมีประสิทธิภาพ โดยสามารถติดต่อกับหน่วยความจำได้ถึง 64 กิโลไบต์ รวมทั้งหน่วยความจำที่อยู่ภายในตัวชิพด้วย การจัดหน่วยความจำแบ่งออกได้เป็น 4 แบบตามโหมดการทำงาน ดังรูปที่ 2.7



รูปที่ 2.7 การจัดหน่วยความจำในโหมดต่างๆ

พื้นที่หน่วยความจำแอดเดรส 0000H - 00FFH เป็นพื้นที่ของหน่วยความจำแรมที่อยู่ภายในชิพของทุกโหมดการทำงาน แอดเดรส 0100H - 0FFFH จะเป็นพื้นที่หน่วยความจำภายนอกในโหมดการทำงาน EXPANDED MUX กับ SPECIAL TEST ที่ตำแหน่งแอดเดรส 1000H-103FH จะเป็นพื้นที่ของรีจิสเตอร์ภายในชิพของทุกโหมดการทำงาน โดยแอดเดรส 1040H-B5FFH เป็นพื้นที่หน่วยความจำภายนอกใน EXPANDED MUX กับ SPECIAL TEST ถัดมาที่แอดเดรส B600H-B7FFH เป็นพื้นที่ของหน่วยความจำอีพรอมภายในชิพ แอดเดรส BF40H-BFFFH เป็นพื้นที่บัสรวมในโหมด SPECIAL BOOT STRAP ส่วนตำแหน่งแอดเดรส B800H-FFFFH เป็นพื้นที่สำหรับหน่วยความจำภายนอกและที่แอดเดรส FFC0H ถึง FFFFH เป็นพื้นที่สำหรับเก็บเวกเตอร์การอินเตอร์รัพต์ของอินเตอร์รัพต์ทุกประเภท

2.1.7 หน่วยความจำแรมภายในชิพ

หน่วยความจำแรมภายในชิพแบ่งออกเป็น 2 ประเภท คือ

1. แรมที่ใช้ในการอ่านเขียนข้อมูลและรีจิสเตอร์ โดยตั้งแต่ที่แอดเดรส 0000H ถึง 00FFH (256 ไบต์) เป็นหน่วยความจำที่ใช้ในการเขียนอ่านข้อมูลทั่วไป ซึ่งหน่วยความจำบริเวณนี้สามารถเก็บข้อมูลได้แม้ไฟเลี้ยงของวงจรจะหายไป ข้อมูลแรมในส่วนนี้ในสภาวะปกติจะได้ไฟเลี้ยงจาก VDD แต่เมื่อ VDD หายไปจะได้ไฟเลี้ยงจากอีกทางหนึ่ง โดยก่อนที่ VDD จะหายไปต้องมีสัญญาณรีเซ็ต เพื่อให้ชิพหยุดการทำงานเพื่อป้องกันไม่ให้เกิดการเขียนข้อมูลในหน่วยความจำ และเมื่อ VDD เข้ามาก็จะเข้าไปจ่ายให้กับแรมส่วนนี้ตามเดิมทันที โดยที่ข้อมูลที่อยู่ในแรมส่วนนี้จะไม่มีการเปลี่ยนแปลง กระแสไฟฟ้าที่ใช้เลี้ยงแรมส่วนนี้จะใช้กระแสไฟน้อยมากจึงสามารถใช้แบตเตอรี่เล็ก ๆ เป็นไฟเลี้ยงให้กับแรมนี้ได้

2. แรมอีกส่วนที่ใช้เป็นรีจิสเตอร์จะถูกใช้ในการควบคุมการทำงานของต่าง ๆ ของชิพ โดยมีความแตกต่างกันออกไปดังนี้

DDRD รีจิสเตอร์ควบคุมทิศทางการติดต่อของพอร์ต D
 PORTE รีจิสเตอร์พอร์ต E ใช้สำหรับอ่านข้อมูลของพอร์ต E
 CFORC รีจิสเตอร์บอกผลของการเปรียบเทียบของไทม์เมอร์ต่าง ๆ
 OC1H รีจิสเตอร์ที่ใช้กำหนดบิตของข้อมูลในพอร์ต A ที่มีการเปรียบเทียบกับ OC1
 OC1D รีจิสเตอร์สำหรับเก็บข้อมูลของการเปรียบเทียบระหว่างพอร์ต A กับ OC1
 TOC1H รีจิสเตอร์ชี้ตผลของการเปรียบเทียบบิตต่อบิตระหว่างพอร์ต A กับ TOC1
 TCNT รีจิสเตอร์ 16 บิตใช้เป็นตัวฟรีเคาน์เตอร์ของระบบไทม์เมอร์ โดยจะทำการนับตั้งแต่ 0000H - FFFFH แล้วกลับมาเริ่มต้นใหม่

TIC1 รีจิสเตอร์ขนาด 16 บิตเป็นอินพุตของไทม์เมอร์ 1

TIC2 รีจิสเตอร์ขนาด 16 บิตเป็นอินพุตของไทม์เมอร์ 2

TIC3 รีจิสเตอร์ขนาด 16 บิตเป็นอินพุตของไทม์เมอร์ 3

TOC1 รีจิสเตอร์ขนาด 16 บิต ใช้ในการเปรียบเทียบกับไทม์เมอร์เคาน์เตอร์ 1

TOC2 รีจิสเตอร์ขนาด 16 บิต ใช้ในการเปรียบเทียบกับไทม์เมอร์เคาน์เตอร์ 2

TOC3 รีจิสเตอร์ขนาด 16 บิต ใช้ในการเปรียบเทียบกับไทม์เมอร์เคาน์เตอร์ 3

TOC4 รีจิสเตอร์ขนาด 16 บิต ใช้ในการเปรียบเทียบกับไทม์เมอร์เคาน์เตอร์ 4

TOC5 รีจิสเตอร์ขนาด 16 บิต ใช้ในการเปรียบเทียบกับไทม์เมอร์เคาน์เตอร์ 5

TCTL1 เป็นรีจิสเตอร์ที่ใช้ในการเลือกลักษณะเอาต์พุตจากการใช้ฟังก์ชันการเปรียบเทียบเอาต์พุต

TCTL2 เป็นรีจิสเตอร์ที่ใช้ในการเลือกลักษณะอินพุต เพื่อให้ฟังก์ชันอินพุตแคปเจอร์

TMSK1 เป็นรีจิสเตอร์ที่ใช้สำหรับอานาเบล หรือคิสเอเบิลอินเตอร์รัพต์ของตัวเปรียบเทียบเอาต์พุตกับตัวเปรียบเทียบอินพุต

TFLG1 เป็นรีจิสเตอร์ที่แสดงถึงการเท่ากันของไทม์เมอร์เคาท์เตอร์ กับ ตัวเปรียบเทียบเอาต์พุต กับ ตัวเปรียบเทียบอินพุต

TMSK2 เป็นรีจิสเตอร์ที่ใช้สำหรับอานาเบลหรือคิสเอเบิลการอินเตอร์รัพต์ของเรียลไทม์และเป็นตัวกำหนดตัวหารของเวลาเรียลไทม์

TFLG2 เป็นรีจิสเตอร์ที่ใช้แสดงการทำงานต่าง ๆ เป็นไทม์อินเตอร์รัพต์โอเวอร์โฟลว์

ACTL เป็นรีจิสเตอร์ที่ใช้ในการควบคุมพัลส์แอกคิวมูเลเตอร์

PACNT เป็นรีจิสเตอร์เคาน์เตอร์ของพัลส์แอกคิวมูเลเตอร์

AFCR เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานแบบ SPI (Serial Peripheral Interface)

SPSR เป็นรีจิสเตอร์ที่ใช้ในการบอกลักษณะการติดต่อแบบ SPI

BACD เป็นรีจิสเตอร์ที่ใช้กำหนดความเร็วในการส่งข้อมูลแบบ SCI (Serial - Communication Interface)

SCCR1 เป็นรีจิสเตอร์ที่ใช้ในการกำหนดรูปแบบในการส่งข้อมูลแบบ SCIS, CCR2 เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการอินเตอร์รัพต์

SCCR เป็นรีจิสเตอร์ที่ใช้ในการบ่งสถานะการรับส่งข้อมูลแบบ SCI

SCDR เป็นรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลจากการรับหรือส่งข้อมูลแบบ SCI

ADCT1 เป็นรีจิสเตอร์ที่ใช้ในการเลือกการอ่านข้อมูลแบบอะนาล็อก และแสดงสถานะการอ่านข้อมูล

ADR1 เป็นรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลที่ได้จากวงจร A/D 1

ADR2 เป็นรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลที่ได้จากวงจร A/D 2

ADR3 เป็นรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลที่ได้จากวงจร A/D 3

ADR4 เป็นรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลที่ได้จากวงจร A/D 4

OPTION เป็นรีจิสเตอร์ที่ใช้ในการกำหนดแบบของการใช้ระบบ COP (Computer Operating Properly) การเปิดสัญญาณนาฬิกาสำหรับการแซมปลิงในวงจร A/D

COPRST การเปิดสัญญาณนาฬิกาเลือกที่ขีดค่าเวลาในระบบ COP

PPROG เป็นรีจิสเตอร์ที่ใช้ในการกำหนดรูปแบบของการติดต่อกับอ็อพรอม

HPIRO เป็นรีจิสเตอร์ที่ใช้ในการจัดลำดับการอินเตอร์รัพต์

INIT เป็นรีจิสเตอร์ที่ใช้ในการกำหนดตำแหน่งของแรมและรีจิสเตอร์

TEST1 เป็นรีจิสเตอร์ที่ใช้ในการทดสอบของโรงงานผู้ผลิต

CONFIG เป็นรีจิสเตอร์ที่ใช้ในการบ่งบอกอุปกรณ์ภายในตัวชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

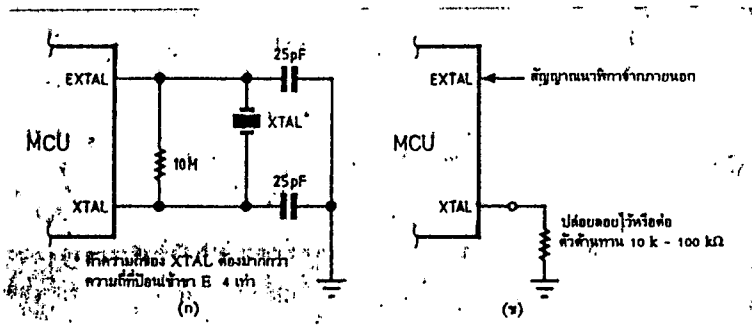
9000	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	PORTA	IO Port A
9001	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	Reserved	
9002	STAP	STAM	STW24	HWPS	SW	PLB	ESA	HWB	PORTC	Parallel IO Control Register
9003	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	PORTC	IO Port C
9004	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	PORTB	Output Port B
9005	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	PORTD	Alternate Latched Port C
9006	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	Reserved	
9007	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	DIRA	Data Direction for Port A
9008	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	DIRB	IO Port B
9009	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	DIRD	Data Direction for Port D
900A	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	DIRTE	Input Port E
900B	FRCA	FRCB	FRCC	FRCD	FRCE	FRCF	FRCG	FRCH	FRCHC	Compare Force Register
900C	OC1A0	OC1A1	OC1A2	OC1A3	OC1A4	OC1A5	OC1A6	OC1A7	OC1A8	OC1 Action Match Register
900D	OC1B0	OC1B1	OC1B2	OC1B3	OC1B4	OC1B5	OC1B6	OC1B7	OC1B8	OC1 Action Edge Register
900E	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT0	Timer Counter Register
900F	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT1	Input Capture 1 Register
9010	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT2	Input Capture 2 Register
9011	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT3	Input Capture 3 Register
9012	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT4	Input Capture 4 Register
9013	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT5	Output Compare 1 Register
9014	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT6	Output Compare 2 Register
9015	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT7	Output Compare 3 Register
9016	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT8	Output Compare 4 Register
9017	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	TCNT9	Output Compare 5 Register
9018	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	Reserved	
9019	ASPM	CSOL	MODE	RLV	CHG	CR	CM	SPTRM	SPTRM	System Configuration Register
901A	Ba 7	Ba 6	Ba 5	Ba 4	Ba 3	Ba 2	Ba 1	Ba 0	CPFRST	Anti-Pass CDP Timer Counter
901B	OOD	IVN	IVT	NOW	PLAB	ELAY	EEPM	SPRND	EEPROM Programming Control Register	
901C	REQ0	REQ1	REQ2	REQ3	REQ4	REQ5	REQ6	REQ7	REQ8	Highest Priority 8 Bit Int and Mask
901D	RAM2	RAMP1	RAMP0	RAMP3	RAMP2	RAMP1	RAMP0	RAMP3	RAMP2	RAM and SR Masking Register
901E	TRAP	GDCA	CTYP	WSP	FCM	ICOP	TCM	TEST	TEST	Factory TEST Control Register
901F	---	---	---	---	---	---	---	---	---	CDP, VCHL and EEPROM Erases

ตารางที่ 2.5 รีจิสเตอร์ควบคุมและพอร์ตต่าง ๆ

2.1.8 วงจรกำเนิดสัญญาณนาฬิกา

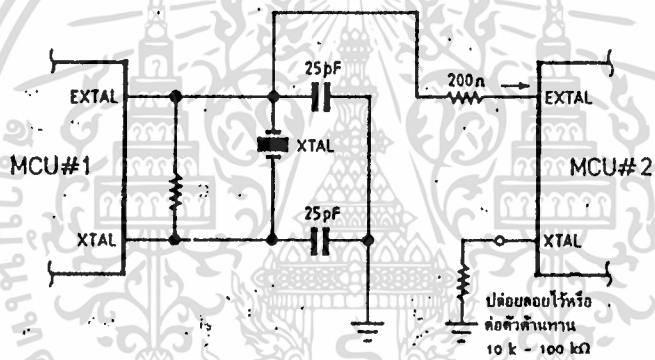
68HC11 สามารถต่อสัญญาณนาฬิกาได้สองรูปแบบคือ

- ใช้วงจรกำเนิดสัญญาณนาฬิกาภายใน โดยการต่อคริสตัลภายนอกกับตัวต้านทานและตัวเก็บประจุอีกสองตัว ดังรูปที่ 2.8 (ก)
- การต่อสัญญาณนาฬิกาจากภายนอกเข้าที่ขา XTAL และต่อตัวต้านทานเข้าที่ขา XTAL ลงกราวด์ ดังรูปที่ 2.8 (ข) นอกจากนี้สัญญาณนาฬิกาจากการที่ต่อ XTAL จากไมโครคอนโทรลเลอร์ตัวที่หนึ่งยังสามารถใช้เป็นสัญญาณนาฬิกาให้กับไมโครคอนโทรลเลอร์ตัวต่อไปได้อีก ดังรูปที่ 2.9 โดยสัญญาณนาฬิกาที่ให้กับซีพียูต้องมีความถี่สูงกว่าสัญญาณที่ขา E ประมาณ 4 เท่า



รูปที่ 2.8 การต่อสัญญาณนาฬิกาให้กับ 68HC11

- ก. สัญญาณนาฬิกาภายใน
- ข. สัญญาณนาฬิกาจากภายนอก



รูปที่ 2.9 การต่อสัญญาณนาฬิกาไมโครคอนโทรลเลอร์ตัวหนึ่งไปยังอีกตัวหนึ่ง

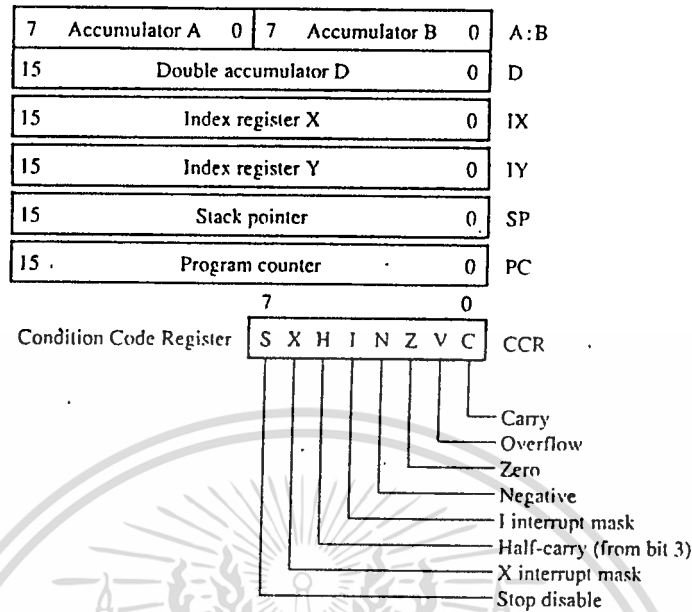
2.1.9 รีจิสเตอร์ของชิพชิพ

ชิพชิพของไมโครคอนโทรลเลอร์ 68HC11 จะมีรีจิสเตอร์ใช้งานอยู่ 7 ตัวอันได้แก่

1. แอควิวมูลเตอร์ A และ B
2. แอควิวมูลเตอร์ D
3. รีจิสเตอร์อินเด็กซ์ IX
4. รีจิสเตอร์อินเด็กซ์ IY
5. รีจิสเตอร์ตัวชี้สแต็ค (Stack Pointer : SP)
6. รีจิสเตอร์โปรแกรมเคาน์เตอร์ (Program Counter: PC)
7. รีจิสเตอร์รหัสเงื่อนไข (Condition Code Register : CCR)

โดยมีลักษณะของรูปแบบการจัดจำนวนบิตและความหมายดังรูปที่ 2.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 รูปแบบการจัดบิตและความหมายรีจิสเตอร์รหัสเงื่อนไขของชิพ 8080

2.1.9.1 แยกความเลแเคอร์ A และ B

เป็นรีจิสเตอร์ขนาด 8 บิต จะใช้ในการเก็บค่าโอเปอร์เรนด์และผลของการคำนวณทางคณิตศาสตร์ หรือผลของการจัดข้อมูลโดยชิพ ในการประมวลผลทางคณิตศาสตร์หรือลอจิก จะต้องนำข้อมูลเหล่านั้นมาเก็บในรีจิสเตอร์ทั้งสองนี้จึงจะสามารถประมวลผลได้

2.1.9.2 แยกความเลแเคอร์ D

เป็นรีจิสเตอร์ขนาด 16 บิต ที่ใช้ในการประมวลผลและใช้สำหรับเก็บค่าการคำนวณทางคณิตศาสตร์และลอจิก แยกความเลแเคอร์ D เกิดจากการรวมกันของแยกความเลแเคอร์ A และ B จึงทำให้มีขนาด 16 บิต

2.1.9.3 รีจิสเตอร์อินเด็กซ์ IX

เป็นรีจิสเตอร์ขนาด 16 บิตสำหรับใช้ในการชี้ตำแหน่งแอดเดรสเพื่อเข้าไปจัดการประมวลผลกับข้อมูลในแอดเดรสนั้น ๆ นอกจากนี้รีจิสเตอร์ IX ยังสามารถใช้เป็นตัวนับหรือรีจิสเตอร์เก็บข้อมูลชั่วคราวได้ด้วย

2.1.9.4 รีจิสเตอร์อินเด็กซ์ IY

เป็นรีจิสเตอร์ขนาด 16 บิต มีหน้าที่เหมือนกับ IX แต่แตกต่างกันที่ในทุกคำสั่งที่เกี่ยวข้องกับ IY จะต้องมีข้อมูลไปคิพิเศษของรหัสแมชชีน และใช้เคิลพิเศษของช่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่จะยึดเงื่อนไขการนำ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเอ็กซ์คิวต์ด้วย จึงทำให้คำสั่งที่มี IY ไปเกี่ยวข้องกับขนาดอย่างน้อย 2 ไบต์ขึ้นไปหรือที่เรียกว่า พรีไบต์ (Prebyte)

2.1.9.5 รีจิสเตอร์ตัวชี้สแต็ค SP

เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บแอดเดรสบนสแต็ค (Stack) โดยสแต็คบน 68HC11 นี้จะมีลักษณะการเก็บข้อมูลเข้าและการนำข้อมูลออกมาเป็นแบบ LIFO (Last In First Out) หรือข้อมูลที่นำเข้าไปเก็บในสแต็คหลังสุดเมื่อจะเรียกออกมาจะถูกเรียกออกมาก่อน สแต็คจะใช้เก็บข้อมูลของรีจิสเตอร์เมื่อต้องการนำรีจิสเตอร์ตัวนั้นไปทำงานในโปรแกรมย่อยอื่น ๆ หรือต้องไปใช้ในการตอบสนองการอินเทอร์รัพต์ เพื่อเป็นการป้องกันข้อมูลเดิมสูญหายต้องเก็บข้อมูลนั้นไว้ในหน่วยความจำสำรองก่อนก็คือ สแต็คนั่นเอง

ทุกครั้งที่มีการเก็บข้อมูลลงบนสแต็ค ค่าของ SP จะลดลง ในทางตรงข้ามถ้าเรียกข้อมูลออกมาจากสแต็ค ค่าของ SP ก็เพิ่มขึ้น

2.1.9.6 รีจิสเตอร์โปรแกรมเคาน์เตอร์ PC

เป็นรีจิสเตอร์ขนาด 16 บิต ใช้เก็บค่าแอดเดรสของคำสั่งถัดไปที่ซีพียูจะไปทำการเอ็กซ์คิวต์

2.1.9.7 รีจิสเตอร์รหัสเงื่อนไข CCR

เป็นรีจิสเตอร์ขนาด 8 บิต ในแต่ละบิตจะแสดงความหมายของผลจากการกระทำคำสั่งที่เพิ่งจะเอ็กซ์คิวต์ไปของซีพียู ดังแสดงในรูปที่ 2.10 แต่ละบิตของ CCR เป็นอิสระต่อกัน จึงสามารถตรวจสอบสถานะได้โดยโปรแกรม และสามารถนำผลของการตรวจสอบนี้ไปดำเนินการต่อได้ด้วย รายละเอียดแต่ละบิตใน CCR มีดังนี้

CARRY/BORROW (C) : เรียกว่า "บิตทด" บิตนี้จะเซตเมื่อซีพียูทำการประมวลผลทางคณิตศาสตร์แล้วเกิดการทดค่า (Carry) หรือยืมค่า (Borrow) บิตนี้สามารถที่จะใช้ทำงานร่วมกับคำสั่งการหมุน (Rotate) และเลื่อนข้อมูล (Shift) ได้

OVER FLOW (V) : บิตนี้จะเซตเป็น "1" เมื่อซีพียูได้ทำคำสั่งทางคณิตศาสตร์แล้วเกิดค่าที่เกินออกมา นอกเหนือจากเงื่อนไขดังกล่าวบิตนี้จะเป็น "0"

ZERO (Z) : บิตนี้จะเซตเมื่อผลของการทำงานทางคณิตศาสตร์และลอจิกหรือการประมวลผลข้อมูลแล้วทำให้เกิดเป็นค่าศูนย์ขึ้นมา

NEGATIVE (N) : ถ้าหากผลของการกระทำทางคณิตศาสตร์และลอจิก หรือการประมวลผลข้อมูลแล้วทำให้เกิดค่าเป็นลบ บิตนี้จะเซตผลลัพธ์ที่มีค่าเป็นลบสามารถสังเกตได้จากบิตที่มีนัยสำคัญสูงสุด (MSB) มีค่าเป็นลอจิก "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I-INTERRUPT MASK (I) : เราสามารถเซตบิตนี้ให้เป็น "1" ได้ 2 วิธี คือ 1. โดยวิธีทางฮาร์ดแวร์ และ 2. โดยการใช้คำสั่งที่ใช้ในการดีสแอสেমบลี การกำเนคอินเตอร์รัพต์แบบมาสเคเบิลทั้งภายในและภายนอก

HALF CARRY (H) : จะเซตเป็น "1" เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์ เช่น ADD, ABA หรือ ADC แล้วเกิดการทดเข้ามาจากบิตที่ 3 มายังบิตที่ 4

X-INTERRUPT MASK (X) : บิตนี้จะถูกเซตด้วยวิธีการทางฮาร์ดแวร์เท่านั้น โดยการใช้ปุ่มสัญญาณเข้าที่ขา RESET และขา XIRQ จะรีเซตบิตนี้ด้วยการใช้คำสั่ง TAP หรือ RTI เท่านั้น

STOP DISABLE (S) : บิตนี้จะถูกเซตเมื่อต้องการดีสแอสেমบลีคำสั่ง STOP และถ้ารีเซตบิตนี้จะเป็นการอานาเบิลคำสั่ง STOP บิตนี้ถูกควบคุมโดยโปรแกรมและเมื่อบิตนี้ถูกเซตจะทำให้คำสั่ง STOP มีผลเช่นเดียวกับคำสั่ง NOP

2.1.10 วงจรวอตซ์ดีค็อก

ในไมโครคอนโทรลเลอร์สมัยใหม่มักมีวงจรวอตซ์ดีค็อกอยู่ในชิพ โดยหน้าที่ของวงจรวอตซ์ดีค็อก คือ ทำตัวคอยตรวจสอบการทำงานของซีพียูว่ายังทำงานอยู่ในโปรแกรม หรือมีปัญหาก็เกิดขึ้นกับซอฟต์แวร์จนไม่สามารถทำงานต่อไปได้ถูกต้อง หลักการของวอตซ์ดีค็อกจะใช้หลักการง่าย ๆ ด้วยการเขียนรหัสอักษรมาเรียงตัวในรีจิสเตอร์ตัวใดตัวหนึ่ง ในลักษณะลำดับช่วงเวลาที่กำหนด หากไม่เขียนภายในเวลาที่กำหนด วงจรจะทำการกระตุ้นบอกซีพียูให้กลับมาเริ่มต้นโปรแกรมที่กำหนดใหม่เพื่อตั้งต้นการทำงานให้ถูกต้องตามเดิม

สำหรับ 88HC11 จะใช้วิธีการเขียนรหัส \$55 ลงในแอดเดรสที่กำหนดเพื่อบอกการทำงานและสั่งรหัส \$AA มาเขียนลงในแอดเดรสที่กำหนดเพื่อรีเซตวงจรตั้งเวลา ดังนั้นเมื่อวงจรตั้งเวลาไม่ได้รับการรีเซตภายในเวลาที่กำหนดไว้ ก็จะส่งสัญญาณไปรีเซตซีพียูให้เริ่มต้นใหม่

2.1.11 วงจรเอชดี

เป็นวงจรแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอลขนาด 8 บิต โดยสามารถรับอินพุตได้ถึง 8 ช่อง จะทำการรับครึ่งละช่องโดยการโปรแกรมเพื่อเลือกว่าจะทำการแปลงสัญญาณที่ช่องใด ผลลัพธ์ที่ได้จากการแปลงเป็นอัตราส่วนระหว่างสัญญาณอินพุตกับแหล่งจ่ายที่เป็นไฟอ้างอิงของวงจร A/D โดยแรงดันที่ใช้จะอยู่ในช่วง 2.5 - 5 โวลต์ เวลาที่ใช้ในการแปลงสัญญาณนั้นจะใช้สัญญาณนาฬิกา 32 ลูกต่อหนึ่งครั้งหรือที่เวลา 16 ไมโครวินาทีที่ความถี่ 8 MHz ผลลัพธ์ที่อ่านเข้ามาได้แล้วจะถูกเก็บอยู่ในรีจิสเตอร์ ADDR1 ถึง ADDR4 ดังตารางที่ 2.6

CD	CC	CB	CA	Channel Signal	Result in ADRx if MULLTL = 1
0	0	0	0	AN0	ADR1
0	0	0	1	AN1	ADR2
0	0	1	0	AN2	ADR3
0	0	1	1	AN3	ADR4
0	1	0	0	AN4	ADR1
0	1	0	1	AN5	ADR2
0	1	1	0	AN6	ADR3
0	1	1	1	AN7	ADR4
1	0	0	0	Reserved	ADR1
1	0	0	1	Reserved	ADR2
1	0	1	0	Reserved	ADR3
1	0	1	1	Reserved	ADR4
1	1	0	0	V _{RH} Pin	ADR1
1	1	0	1	V _{RL} Pin	ADR2
1	1	1	0	(V _{RH})/2	ADR3
1	1	1	1	Reserved	ADR4

ตารางที่ 2.6 การกำหนดช่องการทำงานและการเก็บข้อมูลของวงจร A/D ใน 68HC11

2.1.12 ไทม์เมอร์

68HC11 มีไทม์เมอร์ขนาด 16 บิต ที่ทำงานในลักษณะฟรีรันนิ่ง คือ จะทำการนับตั้งแต่ 0000H-FFFFH และกลับมาเริ่มต้นนับใหม่ โดยใช้สัญญาณนาฬิกาจากภายในตัวที่พื้ โดยไทม์เมอร์นี้จะเป็นตัวเปรียบเทียบกับรีจิสเตอร์ไทม์เมอร์อินพุตและค่าไทม์เมอร์เอาต์พุต รีจิสเตอร์ไทม์เมอร์อินพุต (input captuer) เป็นรีจิสเตอร์ที่อ่านได้อย่างเดียวมีทั้งหมด 3 ตัว ถ้าหากมีสัญญาณลอจิก "0" ที่ขาอินพุตของแต่ละตัวหลังจากที่ได้ทำการโหลดค่าจากฟรีรันไทม์เมอร์แล้ว สามารถที่จะทำการอินเตอร์รัพต์พื้ได้และยังสามารถนำค่าที่โหลดมาใช้ในการวิเคราะห์คาบเวลาได้ว่าเป็นความถี่เท่าไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ไทม์เมอร์เอาต์พุต (output compare) เป็นรีจิสเตอร์ที่เขียนและอ่านได้มีทั้งหมด 5 ตัว อยู่ที่ตำแหน่งตั้งแต่แอดเดรส 1016H-101FH เราสามารถเขียนค่าลงไปยังรีจิสเตอร์ไทม์เมอร์เอาต์พุต เพื่อที่จะนำไปเปรียบเทียบกับพรีรีนไทม์เมอร์ เมื่อค่าทั้งสองเท่ากันก็จะทำการอินเตอร์รัพต์ซีพียู หรือจะทำการหารความถี่จากสัญญาณนาฬิกาไปยังเอาต์พุตได้ทั้ง 5 ตัว ซึ่งเราสามารถควบคุมโดยการกำหนดค่าในรีจิสเตอร์ TMSK1 และ TCTL1

2.1.13 เร็ยลไทม์

เป็นวงจรส่วนหนึ่งทำหน้าที่เป็นฐานเวลาให้โปรแกรม โดยสามารถโปรแกรมเลือกเวลาไปยังรีจิสเตอร์ PACTL ที่บิต RTRO และ RTR1 บิต RTIF ในรีจิสเตอร์ TFLG2 จะเป็นตัวกำหนดว่าจะให้ทำการอินเตอร์รัพต์หรือไม่ ถ้าต้องการก็ควบคุมให้บิต RTOF มีค่าเท่ากับ "1"

2.1.14 ฟิลส์แอกคิวมูเลเตอร์

68HC11 มีรีจิสเตอร์สำหรับนับฟิลส์จากภายนอกขนาด 8 บิต โดยความถี่ที่สามารถนับได้สูงสุดจะเท่ากับความถี่ของสัญญาณนาฬิกาภายในตัวซีพียูหารด้วย 64 ลักษณะของการนับจะเป็นการนับขึ้นและถ้านับจนเกินค่าของรีจิสเตอร์แล้วจะทำการอินเตอร์รัพต์ซีพียูได้ โดยการเซตที่บิต PAOVI ของรีจิสเตอร์ TMSK2

2.1.15 รีเซ็ต

การรีเซ็ตของ 68HC11 จะแบ่งออกเป็น 3 ชนิด คือ

1. การรีเซ็ตจากภายนอกซึ่งจะเกิดจากการเปิดเครื่องครั้งแรกหรือสัญญาณรีเซ็ตจากภายนอก

2. การรีเซ็ตจากวงจรวอตช์ดีด็อก

3. การรีเซ็ตเนื่องจากการผิดพลาดของสัญญาณนาฬิกา

การรีเซ็ตจากการเปิดเครื่องจะใช้จำนวนฟิลส์ของสัญญาณนาฬิกา 4064 ลูกหลังจากที่สัญญาณนาฬิกาถูกรับทำงาน ถ้าใช้คริสตัลภายนอกที่ 8 MHz จะใช้เวลา 2 มิลลิวินาทีถ้าซีพียูทำงานในโหมดบัสสเตอร์ปหรือโหมดคอสตอบพิเศษ หลังจากซีพียูรีเซ็ตจะไปเฟลตซ์ข้อมูลจากเวคเตอร์แอดเดรสที่ FFFCH-FFFFH หรือ BFFEh-BFFFH ในกรณีที่มีการรีเซ็ตจากร่วงจรวอตช์ดีด็อกจะทำให้สัญญาณที่จะถูกรีเซ็ตเป็นลอจิก "0" ด้วยแล้วจะไปเฟลตซ์เอาข้อมูลทีเวคเตอร์แอดเดรส FFFAH-FFFBH และกรณีที่รีเซ็ตจากสัญญาณนาฬิกาจะไปเฟลตซ์ข้อมูลที่เวคเตอร์แอดเดรส FFFCH-FFFDH

2.1.16 อินเทอร์เน็ต

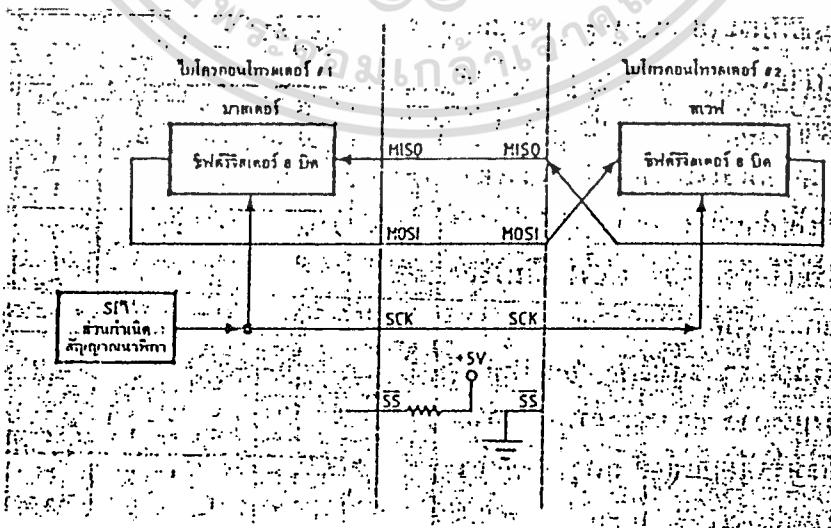
ลักษณะการอินเทอร์เน็ตนี้แบ่งเป็น 2 แบบ คือ ฮาร์ดแวร์และซอฟต์แวร์อินเทอร์เน็ต ฮาร์ดแวร์อินเทอร์เน็ต ซึ่งยังแบ่งออกได้เป็นการอินเทอร์เน็ตจากภายในและจากภายนอก สามารถที่จะลำดับความสำคัญของการขออินเทอร์เน็ตได้ด้วยการควบคุมวีจิสเตอร์ HPRIO การอินเทอร์เน็ตจะทำงานเป็นแบบเวคเตอร์อินเทอร์เน็ต

2.1.17 พอร์ตคอนกรม

พอร์ตคอนกรมของ 68HC11 มีลักษณะเหมือนกับพอร์ตคอนกรมในไมโครคอนโทรลเลอร์ โดยทั่วไป การรับส่งข้อมูลเป็นแบบฟูลดูเพล็กซ์ คือ สามารถรับส่งข้อมูลได้พร้อมกันในเวลา เดียวกันก่อนที่จะรับข้อมูลมายังวีจิสเตอร์จะต้องอ่านข้อมูลเดิมออกไปก่อน มิฉะนั้นข้อมูลที่จะส่ง มาใหม่จะทับข้อมูลเดิม แอดเรสของวีจิสเตอร์อยู่ที่ตำแหน่ง 102FH ในการรับส่งข้อมูลนี้ สามารถที่จะกำหนดการรับส่งข้อมูลได้ว่าเป็นแบบ 10 บิตหรือ 11 บิต แต่ลักษณะพิเศษของ พอร์ตคอนกรมสำหรับ 68HC11 นี้คือ สามารถที่จะทำการตรวจสอบสัญญาณรบกวนได้ โดย ความกว้างของสัญญาณรบกวนจะต้องไม่เกิน 1/16 ของอัตราความเร็วในการส่ง

2.1.18 มัลติโปรเซสเซอร์

นอกจากการติดต่อข้อมูลแบบพอร์ตอินพุต เอาท์พุต และพอร์ตคอนกรมแล้ว 68HC11 ยังมีการติดต่อแบบมัลติโปรเซสเซอร์ที่สามารถนำชิพอื่นมาต่อหลาย ๆ ตัวได้โดยมีชิพผู้ควบคุมหนึ่ง เป็นมาสเตอร์แล้วมีชิพผู้ควบคุมอื่น ๆ เป็นสเลฟจะใช้สัญญาณที่ขา SS, SCK, MISO, MOSI เป็นตัวติดต่อ ดังรูปที่ 2.11



รูปที่ 2.11 ไมโครคอนโทรลเลอร์ 68HC11 ที่ทำงานแบบมัลติโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นใบใช้ประโยชน์ดำเนินการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การอ้างแอดเดรสและชุดคำสั่งของ 68HC11

2.2.1 การอ้างแอดเดรสของ 68HC11

68HC11 มีกระบวนการอ้างแอดเดรส (Addressing) ทั้งสิ้น 6 โหมดด้วยกัน คือ

1. การอ้างแอดเดรสแบบทันทีทันใด (Immediate addressing)
2. การอ้างแอดเดรสแบบโดยตรง (Direct addressing)
3. การอ้างแอดเดรสแบบขยาย (Extended addressing)
4. การอ้างแอดเดรสแบบอินเด็กซ์ (Index addressing)
5. การอ้างแอดเดรสแบบอินเฮียเรนต์ (Inherent addressing)
6. การอ้างแอดเดรสแบบสัมพันธ์ (Relative addressing)

ในบางคำสั่งเมื่อถูกใช้ในการอ้างแอดเดรสบางแบบ จำเป็นต้องเพิ่มข้อมูลเข้าไปอีก 1 ไบต์ก่อนออพโค้ดเพื่อปรับให้ 68HC11 สามารถทำงานได้ในลักษณะมัลติเพจออพโค้ดแมป (Multi page op-code map) ได้เหมาะสมยิ่งขึ้น ข้อมูลไบต์นี้จะเรียกว่า (Prebyte)

2.2.1.1 การอ้างแอดเดรสแบบทันทีทันใด (Immediate Addressing)

ในการอ้างแอดเดรสแบบนี้เป็นการติดต่อเพื่อจัดการข้อมูลซึ่งเป็นค่าใด ๆ โดยตรงในทันทีทันใด จำนวนไบต์ของคำสั่งในการอ้างแอดเดรสแบบนี้จะมีขนาดตั้งแต่ 2 - 4 ไบต์ขึ้นอยู่กับขนาดของรีจิสเตอร์ที่เกี่ยวข้องด้วย เช่น ถ้าเป็นแอกคิวมูเลเตอร์ A ก็จะมีจำนวนไบต์ของคำสั่ง 2 ไบต์ (1 ไบต์แรกเป็นออพโค้ด อีก 1 ไบต์หลังเป็นขนาดของโอเปอร์แรนด์ ซึ่งเท่ากับขนาดของแอกคิวมูเลเตอร์ A)

รูปแบบคำสั่งที่มีการอ้างแอดเดรสแบบนี้ หลังจากคำสั่งต้องตามด้วยเครื่องหมาย # เสมอ เพื่อเป็นการบ่งบอกให้ชี้ปัญหาทราบว่า คำสั่งที่จะทำต่อไปนี้ใช้ในการอ้างแอดเดรสแบบทันทีทันใด

2.2.1.2 การอ้างแอดเดรสแบบโดยตรง (Direct Addressing)

เป็นการติดต่อเพื่อเป็นการประมวลผลข้อมูลขนาด 8 บิต ที่อยู่ในหน่วยความจำภายในชิพซึ่งมีอยู่ 256 ไบต์ โดยมีแอดเดรสตั้งแต่ \$0000 - \$00FF ดังนั้นค่าโอเปอร์แรนด์ที่ตามหลังออพโค้ดคำสั่งก็คือ ค่าแอดเดรสของแรมนั่นเอง

ในการอ้างแอดเดรสแบบนี้ บางทีเรียกว่า การอ้างแอดเดรสเพจศูนย์ (Zero page addressing)

2.2.1.3 การอ้างแอดเดรสแบบขยาย (Extended addressing)

ในการอ้างแอดเดรสแบบนี้ ข้อมูลในไบต์ที่ 2 และ 3 ที่ตามหลังออพโค้ด จะเก็บค่าแอดเดรสจริงของหน่วยความจำที่ต้องการนำข้อมูลในหน่วยความจำออกมาประมวลผล หรือเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จัดเก็บข้อมูลลงไปใหม่ ใช้การอ้างแอดเดรสแบบนี้ค่าสิ่งจะมีขนาด 3-4 ไบต์ โดยฮอปโค้ด ไบต์ที่ 1 หรือ 2 (กรณีถ้ามีขนาด 4 ไบต์) ส่วน 2 ไบต์หลังจะเป็นค่าแอดเดรสที่อ้างถึง ในหน่วยความจำ

2.2.1.4 การอ้างแอดเดรสแบบอินเด็กซ์ (Indexed Addressing)

การอ้างแอดเดรสแบบนี้จะมีการนำรีจิสเตอร์ที่ข้อมูล (Index register) ทั้ง 2 ตัว คือ IX และ IY มาใช้ในการคำนวณค่าแอดเดรสที่ต้องการเรียกใช้ข้อมูล ดังนั้นค่าแอดเดรสที่ต้องการใช้งานจะเปลี่ยนแปลงหรือไม่ ขึ้นอยู่กับองค์ประกอบหลัก 2 ประการ คือ

1. ค่าแอดเดรสที่ถูกกำหนดอยู่ในปัจจุบัน
2. ค่าออฟเซต 8 บิตที่บรรจุอยู่ในค่าสิ่งหรือค่าโอเปอร์แรนด์

ในการอ้างแอดเดรสแบบนี้ สามารถที่จะใช้หน่วยความจำที่ตำแหน่งใดก็ได้ในจำนวน 64 กิโลไบต์ เป็นจุดอ้างอิง ส่วนในด้านขนาดของค่าสิ่งในการอ้างแอดเดรสแบบนี้ ถ้าใช้รีจิสเตอร์ IX จะมีขนาด 2 ไบต์ โดยไบต์แรกเป็นฮอปโค้ด มีไบต์ที่สองเป็นค่า ออฟเซต ถ้าหากใช้รีจิสเตอร์ IY จะมีขนาด 3 ไบต์ ไบต์แรกคือฟรีไบต์ ตามด้วยฮอปโค้ด และค่าออฟเซตขนาด 8 บิต

2.2.1.5 การอ้างแอดเดรสแบบอินเฮริเอนต์ (Inherent Addressing)

การอ้างแอดเดรสแบบนี้จะไม่เกี่ยวข้องกับข้อมูลในหน่วยความจำแต่อย่างใด แต่เข้าไปจัดการในรีจิสเตอร์แทน ดังนั้นขนาดของค่าสิ่งในการอ้างแอดเดรสแบบนี้จะมีเพียง 1-2 ไบต์ โดยเป็นฮอปโค้ดทั้งสิ้นไม่มีโอเปอร์แรนด์

2.2.1.6 การอ้างแอดเดรสแบบสัมพัทธ์ (Relative Addressing)

การอ้างแอดเดรสแบบนี้จะใช้จุดค่าสิ่งกระโดด (Jump and branch instructions) ถ้าเงื่อนไขในการกระโดดเป็นจริงค่าออฟเซตขนาด 8 บิต ที่อยู่ตามหลังฮอปโค้ด ก็จะถูกบวกเข้าไปในรีจิสเตอร์โปรแกรมเคาน์เตอร์ (PC) เพื่อกำหนดแอดเดรสต่อไปที่จะข้ามไปทำงานของซีพียู ปกติแล้วขนาดของค่าสิ่งที่ใช้ในการอ้างแอดเดรสแบบนี้จะเท่ากับ 2 ไบต์ โดยไบต์แรกเป็นฮอปโค้ด ส่วนไบต์ที่สองเป็นค่าออฟเซตเพื่อบอกจำนวนที่จะเข้าไปเพิ่มใน PC

2.2.2 ฟรีไบต์ (Prebyte)

เป็นข้อมูลขนาด 8 บิต (1 ไบต์) ที่ใส่เข้าไปก่อนหน้าฮอปโค้ด เพื่อประโยชน์ในการบอกให้ซีพียูทราบถึงลักษณะของการจัดเพจฮอปโค้ด โดยถ้าหาก 68HC11 ทำงานในเอกสารนี้เป็นเอกสารที่ส่งงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจ 1 ข้อมูลพีไบต์ก็ไม่ต้องมี แต่ถ้าหากทำงานในเพจ 2 จะต้องเพิ่มค่าพีไบต์เท่ากับ \$18 เข้าไปก่อนออปปไคด์เสมอ แล้วตามด้วยออปปไคด์ ในกรณีเพจ 3 จะใช้ค่า \$1A และใช้ค่า \$CD สำหรับเพจ 4 ค่าพีไบต์จะเข้าไปเกี่ยวข้องกับเมื่อมีคำสั่งใด ๆ ให้รีจิสเตอร์ IY ทำงาน

2.2.3 ชุดคำสั่งของ 68HC11

รีพูภาษาในไมโครคอนโทรลเลอร์ 68HC11 จะมีลักษณะการทำงานที่คล้ายกับรีพู H6801 แต่ได้มีการเพิ่มเติมความสามารถของ 68HC11 ด้วยการเพิ่มรีจิสเตอร์และคำสั่งให้มากขึ้น เช่น เพิ่มรีจิสเตอร์อินเด็กซ์ขนาด 16 บิตอีก 1 ตัวคือ IY, เพิ่มคำสั่งเกี่ยวกับการหารเลข 16 บิตอีก 2 แบบ, การเพิ่มคำสั่งควบคุมและเพิ่มคำสั่งจัดการข้อมูลระดับบิต (Bit manipulation instruction) ในที่นี้จะแบ่งชุดคำสั่งของ 68HC11 ออกเป็น 7 กลุ่ม ดังต่อไปนี้

1. กลุ่มคำสั่งจัดการเกี่ยวกับข้อมูล (Data handling instructions)
2. กลุ่มคำสั่งทางคณิตศาสตร์ (Arithmetics instructions)
3. กลุ่มคำสั่งทางลอจิก (Logic instructions)
4. กลุ่มคำสั่งการกระโดด (Jump and Branch instructions)
5. กลุ่มคำสั่งเปรียบเทียบและตรวจสอบ (Data test instructions)
6. กลุ่มคำสั่งจัดการเกี่ยวกับรีจิสเตอร์รหัสเงื่อนไข (CCR instructions)
7. กลุ่มคำสั่งควบคุม (Control instructions)

2.2.3.1 กลุ่มคำสั่งจัดการเกี่ยวกับข้อมูล

(1) กลุ่มคำสั่งเคลื่อนย้ายข้อมูล

	รหัสคำสั่ง	การทำงาน	จำนวนโอแอนด์จำนวนโอเกิดของกฎควบคุม E				พื้นที่รีจิสเตอร์ที่เชื่อมโยง											
			แบบจับเส้นตรง	แบบทันทีทันใด	แบบกร	แบบขยาย	แบบอื่นใดก็ได้		S	X	H	I	N	Z	V	C		
							X	Y										
ระดับบิต	BSET	M ← M OR ๓๓			๒5		๒4	๒5										
	BCLR	M ← M AND (ลึกลับ)			๒6		๒4	๒5										
ระดับไบต์	LDAA (LDAB)	A ← M		22	23	34	24	35										
	STAA (STAB)	M ← A			23	34	24	35										
	TAB	B ← A	1/2															
	TBA	A ← B	1/2															
	TAP	CC ← A	1/2															
	TPA	A ← CC	1/2															
	CLRA (CLRB)	A ← 0	1/2										0	1	0	0		
	CLR	M ← 0					36	26	37					0	1	0	0	
	PSHA (PSHB)	Stack ← A	1/3															
PULA (PULB)	A ← Stack	1/4																
ระดับเวิร์ด	LDD	A ← M		33	24	35	25	36										
		B ← M+1																
	STD	M ← A			24	35	25	36										
		M+1 ← B																
	XGDX	D ↔ X	1/3															
	XGDY	D ↔ Y	2/4															
	LDX	X ← M,M+1		33	24	35	25	36										
	STX	M,M+1 ← X			24	35	25	36										
	LDY	Y ← M,M+1		44	35	46	36	36										
	STY	M,M+1 ← Y			35	46	36	36										
	LDS	S ← M,M+1		33	24	35	25	36										
	STS	M,M+1 ← S			24	35	25	36										
	PSHX	Stack ← X	1/4															
	PULX	X ← Stack	1/5															
	PSHY	Stack ← Y	2/5															
	PULY	Y ← Stack	2/6															
	TSX	X ← S-1	1/3															
	TXS	S ← X-1	1/3															
	TSY	Y ← S+1	2/4															
	TYS	Y ← Y-1	2/4															

ตารางที่ 2.7 กลุ่มคำสั่งเคลื่อนย้ายข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) กลุ่มคำสั่งเปลี่ยนแปลงและแก้ไขข้อมูล

	รหัสคำสั่ง	การคำนวณ	จำนวนไบต์ประมวลผลของวิบูลยควบคุม E				แพลตฟอร์มเวกซ์ลิงในใจ										
			แบบอิน-เอ็อบเทค	แบบกั้นที่ทับโต	แบบครา	แบบขยา	แบบอินเค็กร		S	X	H	I	N	Z	V	C	
							X	Y									
เติมค่า	INCA (INCB)	A ← A+1	1/2														
	INC	M ← M+1				3/6	2/6	3/7									
	INX	X ← X+1	1/3														
	INY	Y ← Y+1	2/4														
	IHS	S ← S+1	1/3														
ลบค่า	DECA (DECQ)	A ← A-1	1/2														
	DEC	M ← M-1				3/6	2/6	3/7									
	DEX	X ← X-1	1/3														
	DEY	Y ← Y-1	2/4														
	DES	S ← S-1	1/3														
แปลงค่า	COMA (COMB)	A ← FFH-A	1/2													0	1
	COM	M ← FFH-M				3/6	2/6	3/7								0	1

ตารางที่ 2.8 กลุ่มคำสั่งเปลี่ยนแปลงและแก้ไขข้อมูล

(3) กลุ่มคำสั่งเลื่อนและหมุนข้อมูล

	รหัสคำสั่ง	การทัววน	จำนวนไบต์ประมวลผลของวิบูลยควบคุม E				แพลตฟอร์มเวกซ์ลิงในใจ										
			แบบอิน-เอ็อบเทค	แบบกั้นที่ทับโต	แบบครา	แบบขยา	แบบอินเค็กร		S	X	H	I	N	Z	V	C	
							X	Y									
หมุนข้อมูล	ROLA (ROLB)	C A,B หรือ M	1/2														
	ROL					3/6	2/6	3/7									
	RORA (RORB)	C A,B หรือ M	1/2														
	ROR					3/6	2/6	3/7									
เลื่อนข้อมูลตามทิศทางชี้	ASLA (ASLB)	C A,B หรือ M 0	1/2														
	ASL					3/6	2/6	3/7									
	ASLD	C D ... 0	1/3														
เลื่อนข้อมูลรอบจึก	ASRA (ASRB)	C A,B หรือ M	1/2														
	ASR					3/6	2/6	3/7									
	LSLA (LSLB)	C A,B หรือ M 0	1/2														
	LSL					3/6	2/6	3/7									
	LSLD	C D ... 0	1/3														
	LSRA (LSRB)	C A,B หรือ M	1/2										0	1			
	LSR					3/6	2/6	3/7					0	1			
	LSRD	C D ... 0	1/3										0	1			

ตารางที่ 2.9 กลุ่มคำสั่งการเลื่อนและหมุนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกำรใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.2 กลุ่มคำสั่งทางคณิตศาสตร์

รหัสคำสั่ง	การทำงาน	จำนวนบิต/จำนวนไซเคิลของสัญญาณนาฬิกา E						แฟลทริจิสเตอร์หัวเงื่อนไข									
		แบบอิน- เอเจนต์	แบบทันทีที่ ทันใด	แบบตรง	แบบซาย	แบบอินเด็กซ์		S	X	H	I	N	Z	V	C		
						X	Y										
การบวก	ADDA (ADDB)	$A \leftarrow A+M$		2/2	2/3	3/4	2/4	3/5	.	.	↑	.	↑	↑	↑	↑	
	ABA	$A \leftarrow A+B$	1/2						.	.	↑	.	↑	↑	↑	↑	
	ADCA (ADCB)	$A \leftarrow A+M+C$		2/2	2/3	3/4	2/4	3/5	.	.	↑	.	↑	↑	↑	↑	
	DAA	Adjust binary sum in A to BCD	1/2						↑	↑	↑	↑	
	ADDD	$D \leftarrow D+(M+1)$		3/4	2/5	3/6	2/6	3/7	↑	↑	↑	↑	
	ABX	$X \leftarrow X \cdot 00B$	1/3					
	ABY	$Y \leftarrow Y \cdot 00B$	2/4					
การลบ	SUBA (SUBB)	$A \leftarrow A-M$		2/2	2/3	3/4	2/4	3/5	↑	↑	↑	↑	
	SBA	$A \leftarrow A-B$	1/2						↑	↑	↑	↑	
	SBCA (SBCB)	$A \leftarrow A-M-C$		2/2	2/3	3/4	2/4	3/5	↑	↑	↑	↑	
	NEGA (NEGB)	$A \leftarrow 0-A$	1/2						↑	↑	↑	↑	
	NEG	$M \leftarrow 0-M$				3/6	2/6	3/7	↑	↑	↑	↑	
	SUBD	$D \leftarrow D-(M+1)$		3/4	2/5	3/6	2/6	3/7	↑	↑	↑	↑	
การคูณ และการหาร	MUL	$D \leftarrow A \cdot B$	1/10						↑	
	IDIV	$X \leftarrow D/X;$ $D \leftarrow r$	1/41						↑	0	↑	
	FDIV	$X \leftarrow D/X;$ $D \leftarrow r$	1/41						↑	↑	↑

ตารางที่ 2.10 กลุ่มคำสั่งทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.3 กลุ่มคำสั่งทางลอจิก

รหัสคำสั่ง	การทำงาน	จำนวนไบต์/จำนวนใจเกิดของสัญญาณนาฬิกา E				แฟลกริจิสเตอร์ที่เงื่อนไข											
		แบบอิน-เอ็ชเรเนคต์	แบบทันทีทันใด	แบบตรง	แบบขยาย	แบบอินเด็กซ์		S	X	H	I	N	Z	V	C		
						X	Y										
ANDA (ANDB)	A ← A AND B		2/2	2/3	3/4	2/4	3/5	-	-	-	-	↑	↓	0	-		
ORAA (ORAB)	A ← A OR B		2/2	2/3	3/4	2/4	3/5	-	-	-	-	↑	↓	0	-		
EORA (EORB)	A ← A EOR B		2/2	2/3	3/4	2/4	3/5	-	-	-	-	↑	↓	0	-		

ตารางที่ 2.11 กลุ่มคำสั่งทางลอจิก

2.2.3.4 กลุ่มคำสั่งการกระโดด

(1) กลุ่มคำสั่งการกระโดดแบบไม่มีเงื่อนไข

รหัสคำสั่ง	การทำงาน	จำนวนไบต์/จำนวนใจเกิดของสัญญาณนาฬิกา E				แฟลกริจิสเตอร์ที่เงื่อนไข											
		แบบอิน-เอ็ชเรเนคต์	แบบทันทีทันใด	แบบตรง	แบบขยาย	แบบอินเด็กซ์		S	X	H	I	N	Z	V	C		
						X	Y										
JMP	PC ← M:M+1				3/3	2/3	3/4	-	-	-	-	-	-	-	-		
JSR	Stack ← PC; PC ← M:M+1			2/5	3/6	2/6	3/7	-	-	-	-	-	-	-	-		
BRA	PC ← PC+M		2/3					-	-	-	-	-	-	-	-		
BRN	PC ← PC+2		2/3					-	-	-	-	-	-	-	-		
BSR	Stack ← PC; PC ← PC+M		2/6					-	-	-	-	-	-	-	-		
RTS	PC ← Stack	1/5						-	-	-	-	-	-	-	-		
RTI	CC,B,A,X,Y,PC ← Stack	1/12						↓	↓	↓	↓	↓	↓	↓	↓		

ตารางที่ 2.12 กลุ่มคำสั่งการกระโดดแบบไม่มีเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(2) กลุ่มคำสั่งกระโดดแบบมีเงื่อนไข

	รหัสคำสั่ง	การทำงาน	จำนวนไบต์/จำนวนไซเคิลของอีลูเซชันรหัส E				แปลงรีจิสเตอร์ที่เลือกไป									
			แบบอิม-เลือกเรจิสเตอร์	แบบกับที่ที่ไบต์	แบบทวิ	แบบขยาย	แบบคันทันที		S	X	H	I	N	Z	V	C
							X	Y								
จุดต่อบิตไบต์	BRSET	Branch # (M) AND mm = 00H			4/6		47	58
	BRCLN	Branch # M AND mm = 00H			4.6		47	58
ทดสอบบิต	BCS			2/3				
	BCC			2/3				
	BEO			2/3				
	BNE			2/3				
	BLN			2/3				
	BPL			2/3				
	BVS			2/3				
	BVC			2/3				
เปรียบเทียบกับข้อมูลศูนย์	BLT			2/3				
	BLE			2/3				
	BEO			2/3				
	BGE			2/3				
	BGT			2/3				
	BHE			2/3				
เปรียบเทียบข้อมูล 2 ข้อมูล	BLO			2/3				
	BLS			2/3				
	BEO			2/3				
	BHS			2/3				
	BH			2/3				
	BHE			2/3				

ตารางที่ 2.13 กลุ่มคำสั่งการกระโดดแบบมีเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.5 กลุ่มคำสั่งเปรียบเทียบและตรวจสอบข้อมูล

	รหัสคำสั่ง	การทำงาน	จำนวนไบต์/จำนวนใจเกิดของสัญญาณนาฬิกา E						แฟลทรีจิสเตอร์รหัสเงื่อนไข											
			แบบอิน-เซอเรนด	แบบทันทีทันใด	แบบตรง	แบบขอย	แบบอินเด็กซ์		S	X	H	I	N	Z	V	C				
							X	Y												
ข้อมูลขนาด 8 บิต	CMPA (CMPB)	A - M		2/2	2/3	3/4	2/4	3/5	-	-	-	↑	↑	↑	↑					
	CBA	A B	1/2						-	-	-	↑	↑	↑	↑					
	TSTA (TSTB)	A - 0	1/2						-	-	-	↑	↑	0	0					
	TST	M - 0				3/6	2/6	3/7	-	-	-	↑	↑	0	0					
	BITA (BITB)	A AND M		2/2	2/3	3/4	2/4	3/5	-	-	-	↑	↑	0	-					
ข้อมูลขนาด 16 บิต	CPD	D - (M;M+1)		4/5	3/6	4/7	3/7	3/7	-	-	-	↑	↑	↑	↑					
	CPX	X - (M;M+1)		3/4	2/5	3/6	2/6	3/7	-	-	-	↑	↑	↑	↑					
	CPY	Y - (M;M+1)		4/5	3/6	4/7	3/7	3/7	-	-	-	↑	↑	↑	↑					

ตารางที่ 2.14 กลุ่มคำสั่งเปรียบเทียบและตรวจสอบข้อมูล

2.2.3.6 กลุ่มคำสั่งจัดการกับรีจิสเตอร์รหัสเงื่อนไข

รหัสคำสั่ง	การทำงาน	จำนวนไบต์/จำนวนใจเกิดของสัญญาณนาฬิกา E						แฟลทรีจิสเตอร์รหัสเงื่อนไข											
		แบบอิน-เซอเรนด	แบบทันทีทันใด	แบบตรง	แบบขอย	แบบอินเด็กซ์		S	X	H	I	N	Z	V	C				
						X	Y												
SEC	C ← 1	1/2							-	-	-	-	-	-	1				
CLC	C ← 0	1/2							-	-	-	-	-	-	0				
SEV	V ← 1	1/2							-	-	-	-	-	1	-				
CLV	V ← 0	1/2							-	-	-	-	-	0	-				
SEI	I ← 1 (disable interrupts)	1/2							-	-	-	1	-	-	-				
CLI	I ← 0 (enable interrupts)	1/2							-	-	-	0	-	-	-				

ตารางที่ 2.15 กลุ่มคำสั่งจัดการกับรีจิสเตอร์รหัสเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.3.7 กลุ่มคำสั่งควบคุม

รหัสคำสั่ง	การทำงาน	จำนวนไบต์/จำนวนไซเคิลของสัญญาณนาฬิกา E				แฟลกริเจิสเตอร์รหัสเงื่อนไข											
		แบบอิน-เฮอเรนซ์	แบบทันทีทันใด	แบบตรง	แบบขยาย	แบบอินเล็กซ์		S	X	H	I	N	Z	V	C		
						X	Y										
SWI	Stack ← PC,Y,X,A,B,CC; I ← I; PC ← M:M+1	1/14															
WAI	Stack ← PC,Y,X,A,B,CC; Halt	2/12															
STOP		1/2															
NOP	PC ← PC+1	1/2															

ตารางที่ 2.16 กลุ่มคำสั่งควบคุม

2.2.4 ความแตกต่างของการ JUMP และ BRANCH

คำสั่ง JUMP อันได้แก่ JMR, JSR จะสามารถกระโดดไปที่ใดในหน่วยความจำก็ได้ โดยกำหนดค่าแอดเดรสลงไปยังดัชนี ทำให้ต้องใช้หน่วยความจำสิ้นเปลือง แต่ถ้าเป็นคำสั่ง BRA หรือ BSR จะสามารถกระโดดไปได้ในขอบเขต +127 และ -128 ตำแหน่งจากจุดที่กระทำคำสั่ง ทำให้ใช้คำสั่งในการเขียนโปรแกรมน้อยกว่า และเขียนทำงานเร็วกว่า แต่มีข้อดีน้อยกว่าตรงที่มีขอบเขตของการกระโดดจำกัด

คำสั่งเกี่ยวกับการ BRANCH ถ้าจะเทียบกับไมโครโปรเซสเซอร์ Z80 ก็คือคำสั่ง Jump Relative หรือ JR นั่นเอง ซึ่งค่าสัมพัทธ์ (Relative) จะต้องมีการคำนวณด้วยคำสั่งสูตร

$$\text{ค่าสัมพัทธ์ (relative : rr)} = \text{PC ใหม่} - \text{PC เก่า} - 2$$

โดย rr จะอยู่ที่ค่า 8 บิตท้ายของผลที่ได้ เช่น PC ใหม่มีค่าเท่ากับ \$E009, PC เก่ามีค่าเท่ากับ \$E012 ดังนั้นค่า

$$rr = \$E009 - \$E012 - 2 = \$FFF5$$

นั่นคือเราจะได้ค่า rr ที่แท้จริงก็คือ \$F5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การเขียนโปรแกรมเพื่ออ้างแอดเดรส

68HC11 มีการอ้างแอดเดรส (Addressing) อยู่ด้วยกัน 6 แบบคือ

1. แบบทันทีทันใด (Immediate addressing)
2. แบบโดยตรง (Direct addressing)
3. แบบขยาย (Extended addressing)
4. แบบอินเด็กซ์ (Index addressing)
5. แบบอินเฮริเนต์ (Inherent addressing)
6. แบบสัมพันธ์ (Relative addressing)

ดังนั้นการเขียนโปรแกรมเพื่ออ้างแอดเดรสในแต่ละแบบก็จะแตกต่างกันไป

2.3.1 โปรแกรมสำหรับการอ้างแอดเดรสแบบทันทีทันใด

การเขียนโปรแกรมจะมีรูปแบบการเขียนจะเป็นดังนี้

คำสั่ง # $\&$ ค่าที่ต้องการ

$\&$ คือ ตัวอักษรที่ใช้บ่งบอกว่า ค่าที่ตามหลังนี้จะเป็นเลขฐานสิบหก
ตัวอย่างของโปรแกรมมีดังนี้

LDAA # $\&$ 5C

หมายถึง การนำค่า $\&$ 5C. ไปเก็บไว้ในแอกคิวมูเลเตอร์ A

LDX # $\&$ 3B27

หมายถึง เป็นการนำค่า $\&$ 3B27 ไปเก็บไว้ในรีจิสเตอร์อินเด็กซ์ X (IX) โดย $\&$ 3B จะถูกเก็บไว้ในไบต์สูงของ IX ส่วน $\&$ 27 จะเก็บไว้ในไบต์ต่ำของ IX

2.3.2 โปรแกรมสำหรับการอ้างแอดเดรสแบบโดยตรง

รูปแบบการเขียนโปรแกรมมีดังนี้

คำสั่ง $\&$ ค่าแอดเดรสของแรมที่อยู่ภายในชิพ (มีค่าได้ตั้งแต่ $\&$ 00- $\&$ FF)

ตัวอย่างของโปรแกรมมีดังนี้

LDAA $\&$ 1B

เป็นการนำค่าที่อยู่ในแรมแอดเดรสที่ $\&$ 1B มาเก็บไว้ในแอกคิวมูเลเตอร์ A ไม่ใช่เป็นการนำค่า $\&$ 1B มาเก็บไว้ในแอกคิวมูเลเตอร์ A

2.3.3 โปรแกรมสำหรับการอ้างอิงแอดเดรสแบบขยาย

รูปแบบการเขียนโปรแกรมเป็นดังนี้

คำสั่ง $\$$ ค่าแอดเดรสหน่วยความจำที่อยู่ภายนอกชิพ

ตัวอย่างของโปรแกรมนี้นี้

LDAB $\$2000$

เป็นการนำข้อมูลในหน่วยความจำแอดเดรส $\$2000$ มาเก็บในแอดคิวมูเลเตอร์ B

2.3.4 โปรแกรมสำหรับการอ้างอิงแอดเดรสแบบลินเด็ก์

รูปแบบการเขียนโปรแกรมเป็นดังนี้

คำสั่ง $\$$ ค่าออฟเซต, X (หรือ Y)

ตัวอย่างของโปรแกรมนี้นี้

STAA $\$05, X$

สมมติว่า IX มีแอดเดรสก่อนหน้าที่จะเอ็กซ์คิวต์คำสั่งนี้อยู่ที่ $\$2000$ เมื่อเอ็กซ์คิวต์คำสั่งในโปรแกรมนี้นี้ เป็นการนำค่าที่อยู่ในแอดคิวมูเลเตอร์ A ไปเก็บลงในหน่วยความจำที่มีแอดเดรสเท่ากับ $IX + 5$ นั่นคือ $\$2000 + \$5 = \$2005$

2.3.5 โปรแกรมสำหรับการอ้างอิงแอดเดรสแบบอินเฮียเรนด

เนื่องจากการอ้างอิงแอดเดรสแบบนี้ จะไม่ยุ่งเกี่ยวกับหน่วยความจำแต่อย่างใด ดังนั้นทุก ๆ คำสั่งเป็นการกระทำกับรีจิสเตอร์เท่านั้น จึงเป็นการอ้างอิงแอดเดรสแบบอินเฮียเรนดทั้งสิ้น ดังตัวอย่างต่อไปนี้

CLRA : เป็นการเคลียร์ค่าในแอดคิวมูเลเตอร์ A

XGDY : สลับค่าในแอดคิวมูเลเตอร์ D กับรีจิสเตอร์ IY

DEX : ลดค่าของ IX

2.3.6 โปรแกรมสำหรับการอ้างอิงแอดเดรสแบบสัมพัทธ์

รูปแบบการเขียนโปรแกรมเป็นดังนี้

คำสั่ง คำออฟเซตสัมพัทธ์ที่ต้องการ

เนื่องจากการอ้างอิงแอดเดรสแบบนี้ มักจะใช้ในโปรแกรมที่มีการกระโดด คำออฟเซตจึงเป็นค่าที่ใช้บอกให้ทราบว่า หลังจากทีเอ็กซ์คิวต์คำสั่งนี้แล้ว จะต้องกระโดดไปทำงานที่แอดเดรสใด ดังเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$1000      BEQ JMP
           .
           .
           .
$1007      JMP LDAA #$05

```

ที่บรรทัด \$1000 นั้นคือโปรแกรมที่เขียนขึ้นโดยอาศัยการอ้างแอดเดรสแบบสัมพันธ์ ค่าออฟเซตสัมพันธ์ที่ต้องการทราบคำนวณจาก ค่าแอดเดรสเก่า - ค่าแอดเดรสใหม่ - T โดย T จะมีค่าเท่ากับ 2 สำหรับคำสั่งปกติ จะเป็น 4 เมื่อใช้คำสั่ง BRSET และ BRCLR สำหรับ IX และเท่ากับ 5 เมื่อใช้คำสั่ง BRSET และ BRCLR สำหรับ IY จากตัวอย่าง จะได้ค่าออฟเซตสัมพันธ์เท่ากับ $\$1007 - \$1000 - \$2 = \05

2.4 รูปแบบการเขียนโปรแกรมทั่วไป

ทุกครั้งที่เริ่มต้นเขียนโปรแกรมภาษาแอสเซมบลีของ 68HC11 ต้องเริ่มต้นด้วย ORG แล้วตามด้วยแอดเดรสเริ่มต้นของโปรแกรมทุกครั้ง และเมื่อเขียนโปรแกรมจบแล้วจะต้องปิดท้ายด้วยคำสั่ง END เสมอ ดังตัวอย่าง

```

ORG $1000
LDAA #$5C
STAA $1007
END

```

ในการเขียนโปรแกรมด้วยภาษาแอสเซมบลีทุกครั้ง จะต้องนำโปรแกรมไปคอมไพล์ให้เป็นออพโค้ดเสียก่อน เพื่อให้ชิพภายใน 68HC11 เข้าใจคำสั่ง และนี่คือตัวอย่างของโปรแกรมที่ผ่านการคอมไพล์แล้ว

ออพโค้ดที่ผ่านการคอมไพล์แล้ว	โปรแกรมภาษาแอสเซมบลี
	ORG \$4000
4000 CE 8B 26	LDX #\$8B26
4003 BF	XGDX
4004 C0 12	SUBB #12

โปรแกรมที่ใช้คอมไพล์เรียกว่า ครอสแอสเซมเบลอร์ (Cross assembler)

ซึ่งนอกจากจะใช้แปลงภาษาแอสเซมบลีแล้ว ยังสามารถแปลงโปรแกรมที่เป็นภาษาซีได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาเบไซประเขยนดานการค้ำ

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำให้การพัฒนา 68HC11 ทำได้ง่ายและทรงประสิทธิภาพมากขึ้นเพราะการเขียนโปรแกรมด้วยภาษาซีจะทำได้ง่ายกว่า และจะประหยัดเวลาได้มากกว่าการเขียนโปรแกรมด้วยภาษาแอสเซมบลี ดังตัวอย่าง

time()

```
/* function time () increments sec, min, hour based on the 24
hour clock format */
```

```
{
  sec+=1;
  if(sec>=60)
    {sec=0;
    min+=1
    if(min>=60)
      {min=0;
      hour+=1;
      if(hour>=24)
        {hour=0}
      }
    }
}
```

นี่คือโปรแกรมนาฬิกา จะเห็นว่ามีเพียงไม่กี่บรรทัดเท่านั้น แต่ถ้าเป็นการเขียนด้วยภาษาแอสเซมบลี จะยาวกว่านี้มาก

2.5 โปรแกรมคณิตศาสตร์

2.5.1 โปรแกรมบวกเลข

ตัวอย่างของโปรแกรมทางคณิตศาสตร์อย่างง่าย ที่ยกมาเป็นตัวอย่างนี้เป็นโปรแกรมบวกเลขฐาน โดยตัวตั้งมีค่าเท่ากับ \$05 เก็บไว้ในหน่วยความจำแอดเดรส \$3000 ตัวบวกมีค่าเท่ากับ \$03 อยู่ในแอดเดรส \$3001 ตัวลบมีค่า \$02 อยู่ที่แอดเดรส \$3002 ผลลัพธ์จากการคำนวณให้มาเก็บไว้ที่ \$3003

```
ORG $2000
```

```
LDA $3000 ; A = $05
```

```
ADCA $3001 ; A = $05 + $03 = $08
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SBCA \$3002 ; A = \$08 - \$02 = \$06

STAA \$3003 ; A to \$3003

END

2.5.2 โปรแกรมคูณเลข

ต้องการคูณเลขฐานสิบระหว่าง \$20 กับ \$35

ORG \$2000

LDD ##2035 ; \$20 to A, \$35 to B

MUL ; A*B = 06A0 to D

ADCA ##00 ; adjust to 8 bit

END

2.5.3 โปรแกรมหารเลข

ต้องการหาร 6 ด้วย 4 คำตอบก็จะได้ 1 เศษ 2 สามารถเขียนโปรแกรมได้ดังนี้

ORG \$2000

LDD ##0006

LDX ##0004

IDIV

STX \$00

LDX ##0004

FDIV

END

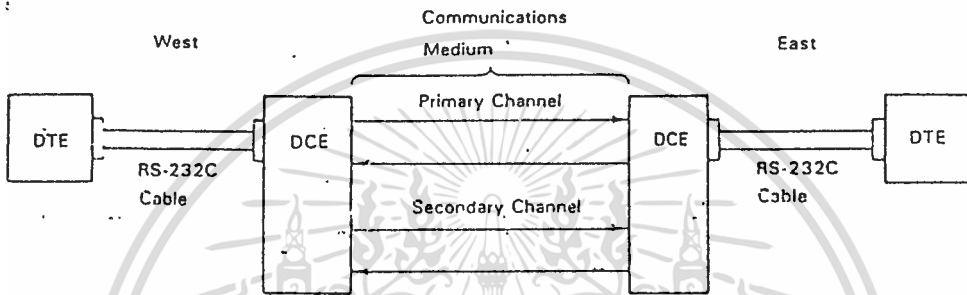
หลังจากกระทำคำสั่ง IDIV จะได้ผลการหารออกมาเป็น 1 เก็บไว้ในรีจิสเตอร์ IX ส่วนเศษ 2 จะถูกเก็บไว้ในรีจิสเตอร์ จากนั้นจะทำการหารเศษ 2 ด้วย 4 จึงต้องเก็บค่าของผลลัพธ์ในช่วงแรกไว้ใน IX จากนั้นก็จะมาทำงานด้วยคำสั่ง FDIV

หลังจากที่เอ็กซ์คิวต์คำสั่ง FDIV จะได้ผลการเท่ากับ 0.5 ถ้าแปลงเป็นเลขฐานสิบหก จะได้ค่า \$8000 เก็บในรีจิสเตอร์ IX เหตุผลที่เป็นค่า \$8000 เนื่องจากเป็นค่าครึ่งหนึ่งจาก \$0000 ถึง \$FFFF พอดี ดังนั้นผลการจึงมีค่าเท่ากับ \$1.8 หรือ 1.5 ในเลขฐานสิบ

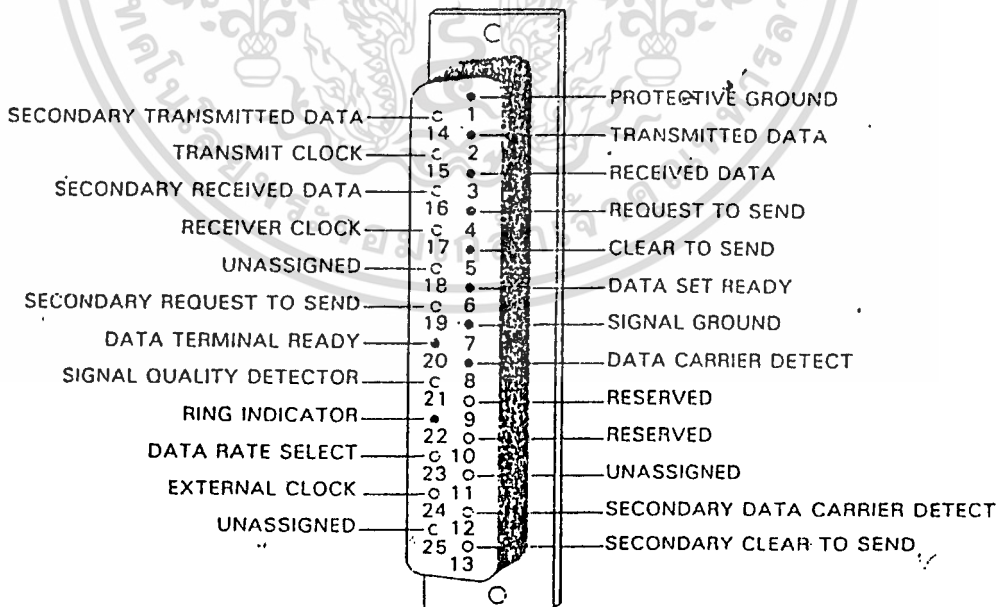
2.6 อินเทอร์เน็ต RS-232 C

2.6.1 แบบจำลองของวงจรการสื่อสารแบบ RS-232 C

ข้อกำหนดตาม RS-232 C นั้น บอกรับการเดินสายในเคเบิลที่ต่อระหว่าง DTE (Data Terminal Equipment) กับ DCE (Data Communication Equipment) ตามรูปที่ 2.12 เป็นเคเบิลที่ต่อกับปลั๊ก 25 ขา ที่เสียบเข้ากับคอนเนคเตอร์ "serial port" ที่ตั้ง PC หรืออุปกรณ์ปลายทางต่าง ๆ ปลั๊กนี้มีลักษณะดังรูปที่ 2.13



รูปที่ 2.12 แบบจำลองของวงจรการสื่อสารที่ใช้ RS-232 C



รูปที่ 2.13 หัวต่อตัวเมียของ RS-232 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในประการแรกข้อกำหนดจะกำหนดระดับแรงดันและคุณสมบัติทางไฟฟ้าอย่างอื่นของสายเคเบิลพร้อมทั้งอธิบายหน้าที่ของมันซึ่งจะได้กล่าวถึงในหัวข้อต่อไป และเพื่อที่จะให้เข้าใจ DTE เราจำเป็นต้องเข้าใจแบบจำลองหรือรูปแสดงแนวความคิดของตัวกลางการสื่อสารที่ใช้ ในข้อกำหนด RS-232 C ด้วย รูปแบบจำลองนี้แสดงในรูปที่ 2.12 ต่อไปเมื่อกล่าวถึงอุปกรณ์ที่อยู่ทางด้านซ้ายมือและด้านขวามือของตัวกลางการสื่อสาร เราจะอ้างว่าเป็นอุปกรณ์ที่อยู่ทางด้านซ้ายมือ และด้านขวามือของตัวกลางการสื่อสารเราจะอ้างว่าเป็นอุปกรณ์ด้านตะวันตก และด้านตะวันออก ตามลำดับ

ตัวกลางการสื่อสารประกอบด้วย ช่องสัญญาณหลัก (primary channel) และช่องสัญญาณรอง (secondary channel) ช่องสัญญาณนั้นเป็นช่องทาง (pipeline) ที่สัญญาณไหลผ่านเท่านั้นดังนั้นช่องสัญญาณหลักและช่องสัญญาณรองนี้อาจมีอยู่ในสายคู่เดียวกัน หรือคนละคู่ก็ได้ เกือบเนื่องกับช่องสัญญาณ ก็คือแนวความคิดเรื่อง คลื่นพาห์ (carrier) คลื่นพาห์คือ สัญญาณอนาล็อก (ปกติเป็นคลื่นรูปไซน์) ที่ความถี่ที่เหมาะสมที่จะเคลื่อนผ่านตัวกลางการสื่อสารได้ ตัวพาห์เป็นตัวบรรทุกข้อมูลที่เรียกว่ามอดูเลตข้อมูล เพื่อเอาข้อมูลผ่านตัวกลางการสื่อสารไปจุดหมายมีวิธีการมอดูเลตหลายวิธี

ในแต่ละช่องสัญญาณมีสัญญาณคลื่นพาห์ 2 สัญญาณ แต่ละสัญญาณใช้สำหรับการส่งแต่ละทิศทาง ถ้าคลื่นพาห์ทั้งสองสามารถใช้พร้อมกันได้ ช่องสัญญาณก็จะ เป็นชนิดฟูลดูเพล็กซ์ แต่ถ้าต้องใช้สลับเวลากัน ช่องสัญญาณนั้นก็ เป็นชนิด ฮาล์ฟดูเพล็กซ์

ช่องสัญญาณมีค่าความจุ ซึ่งมีค่าเท่ากับจำนวนบิตต่อวินาทีที่มากที่สุดที่ช่องสัญญาณยอมให้ผ่านตัวมันไปได้ ช่องสัญญาณรองไม่จำเป็นต้องมีไว้ แต่ถ้ามีอยู่แล้วค่าความจุ (ความเร็ว) ของมันจะน้อยกว่าค่าความจุของช่องสัญญาณหลัก เพราะถ้าไม่เป็นเช่นนั้นก็จะไม่มีแนวความแตกต่างระหว่างช่องสัญญาณหลักกับช่องสัญญาณรอง ข้อกำหนด RS-232 C กำหนดว่ามันไม่สามารถจะประยุกต์ใช้ได้กับตัวกลางการสื่อสารที่มีค่าสูงกว่า 200,000 บิตต่อวินาที

RS-232 C ได้ปล่อยทางเลือกหลายอย่างไว้ให้ขึ้นอยู่กับผู้ออกแบบ เป็นต้นว่า

- ช่องสัญญาณจะเป็น ฮาล์ฟ หรือ ฟูลดูเพล็กซ์
- ตัวกลางการสื่อสารจะเป็นแบบสวิตซ์หรือแบบเช่าเฉพาะ
- ช่องสัญญาณจะเป็นแบบหึ่งโครโซหรืออะซิงโครโซ
- ช่องสัญญาณรองจะมีหรือไม่

สิ่งเหล่านี้ก็ล้วนแล้วแต่ผู้ออกแบบว่าต้องการอย่างไร ผลก็คือจำนวนของสายที่อยู่ในเคเบิล RS-232 C ที่ใช้ควบคุม DCE จะถูกใช้ไม่เท่ากัน ขึ้นอยู่กับการออกแบบ บางทีอาจจะมีแบบหนึ่งแบบใดเลขก็ได้ที่ใช้สายครบหมดทุกสายตาม RS-232 C

2.6.2 ข้อกำหนดทางไฟฟ้า

มีรายละเอียดเกี่ยวกับคุณสมบัติของสัญญาณไฟฟ้าบนสายแต่ละสายในเคเบิล RS-232 มากมายแต่เพราะเราเกี่ยวข้องกับสัญญาณไบนารี ข้อจำกัดเกี่ยวกับแรงดันจึงถูกกำหนดลงในสองบริเวณ ดังแสดงในรูปที่ 2.14 บริเวณย่านบวก (positive region) อยู่ระหว่าง +3 โวลต์ ถึง +15 โวลต์ และบริเวณลบ (negative region) อยู่ระหว่าง -3 โวลต์ ถึง -15 โวลต์ บริเวณระหว่าง -3 โวลต์ ถึง 3 โวลต์ถือเป็นบริเวณเปลี่ยนถ่ายสถานะ (transition region) มีข้อกำหนดให้สัญญาณจะมีสถานะอยู่ในบริเวณนี้ได้ไม่เกิน 1 มิลิวินาที ในบริเวณเปลี่ยนถ่ายสถานะนี้จะไม่มีการกำหนดสภาพไฟฟ้าให้กับสัญญาณแต่อย่างใด



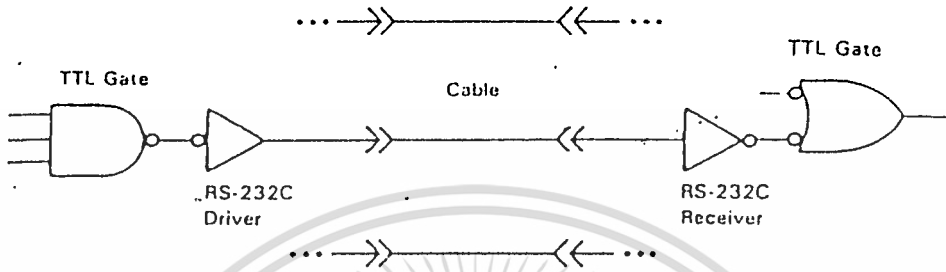
รูปที่ 2.14 ระดับแรงดันของสัญญาณ RS-232 C ที่กำหนดให้ใช้งาน

บริเวณเหล่านี้ใช้เกี่ยวข้องกับสถานะไบนารีของสัญญาณในลักษณะที่เป็นเอกเทศ การแปลความหมายของระดับแรงดันนี้ขึ้นอยู่กับหน้าที่ของสาย กำหนดได้ดังตารางที่ 2.17

WIRE FUNCTION	VOLTAGE LEVEL	
	Positive	Negative
Delta Modem Control & Timing	SPACE(0) on (asserted)	MARK(1) off (negated)

ตารางที่ 2.17 หน้าที่ของสายและความหมายของโวลต์เตจที่กำหนดให้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากระดับแรงดันดังกล่าวไม่พอเหมาะที่จะใช้กับ TTL ดังนั้นการอินเทอร์เฟส ไปสู่เคเบิล RS-232 C จึงต้องการวงจรภาคขับและภาครับเป็นพิเศษดังแสดงในรูปที่ 2.15 ภาคขับและภาครับนี้สามารถหาได้ในรูป IC ตามท้องตลาดทั่วไป



รูปที่ 2.15 วงจรอินเทอร์เฟส RS-232 C

RS-232 C นั้นจำกัดค่าความจุไฟฟ้าของสายสัญญาณโดยวัดเทียบกับชากรราวด์ อย่างมากที่สุดคือ 2,500 pF สำหรับสายเคเบิลที่มีฉนวนและช่องว่างระหว่างสายอย่างปกติทั่วไป ถ้าสายยาวประมาณ 50 ฟุต จะมีค่าความจุไฟฟ้าประมาณนั้น ดังนั้นถ้าไม่ใช่สายชนิดพิเศษแล้ว ระยะห่างมากที่สุดระหว่าง DTE และ DCE คือ 50 ฟุต หรือประมาณ 15 เมตร

2.6.3 ขาขั้วต่อ RS-232 C และหน้าที่

ตารางที่ 2.18 แสดงรายการขาขั้วต่อของ RS-232 C และชื่อสัญญาณที่ใช้ร่วมกับขาขั้วต่อนั้นคอลัมน์ขวามือสุดเป็นชื่อขั้วที่ใช้เรียกขานนั้น ๆ (ชื่อขั้วนี้ยังไม่ได้กำหนดมาตรฐานไว้เรียกโดยสากล) ในหัวขั้วนี้จะให้ข้อมูลเกี่ยวกับขาต่าง ๆ โดยจะจัดอธิบายเป็นกลุ่มตามประเภทของสัญญาณและหน้าที่ของมันโดยย่อ

ขาที่ 1 และขาที่ 7 เป็นชากรราวด์ โดยขาที่ 1 เป็นชากรราวด์เพื่อวัตถุประสงค์หลักในการป้องกันสัญญาณรบกวนโดยรอบและจะช่วยลดการสอดแทรกของสัญญาณอื่นจะเกิดขึ้นได้ ขาที่ 7 เป็นชากรราวด์ซึ่งใช้เพื่อต่อให้เกิดเส้นทางหรือจุดอ้างอิงร่วมกันของสัญญาณทุกชนิด ไม่ว่าจะเป็นข้อมูล, สัญญาณนาฬิกาหรือสัญญาณควบคุมต่าง ๆ ขา 7 จำเป็นต้องต่อระหว่าง DTE และ DCE เพื่อให้เครื่องทำงานร่วมกันได้ถูกต้อง การต่อกราวด์ขา 7 นี้ ควรระวังเรื่องความต่างศักย์อันเนื่องมาจากความต้านทานของสายด้วย ถ้ากราวด์ไม่ได้แล้วมักจะเป็นสาเหตุทำให้การรับข้อมูลมาตีความไม่ถูกต้องได้

Pin	Signal Name	Direction		Abbreviatio
		DTE	DCE	
1	PROTECTIVE (FRAME) GROUND			
2	TRANSMIT DATA		---->	XMT
3	RECEIVE DATA	<----		RCV
4	REQUEST TO SEND		---->	RTS
5	CLEAR TO SEND	<----		CTS
6	DATA SER READY	<----		DSR
7	SIGNAL GROUND (COMMON RETURN)			GRD
8	CARRIER DETECT	<----		CAR_DET
9	-			
10	-			
11	-			
12	SECONDARY CARRIER DETECT	<----		SEC_CAR_DET
13	SECONDARY CLEAR TO SEND	<----		SEC_CTS
14	SECONDARY TRANSMIT DATA		---->	SEC_XMT
15	TRANSMIT CLOCK (DCE SOURCE)	<----		XMT_CLK
16	SECONDARY RECEIVE DATA	<----		SEC_RCV
17	RECEIVE CLOCK	<----		RCV_CLK
18	-			
19	SECONDARY REQUEST TO SEND		---->	SEC_RTS
20	DATA TERMINAL READY		---->	DTR
21	SIGNAL QUALITY DETECTOR	<----		SQD
22	RING INDICATOR	<----		RI
23	DATA RATE SELECTOR		---->	DR_SEL
24	TRANSMIT CLOCK (DTE SOURCE)		---->	XMT_CLK
25	-			

ตารางที่ 2.18 การกำหนดขาหัวต่อ RS-232 C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่ 2 และขาที่ 3 เป็นขาสำหรับส่งและรับข้อมูลตามลำดับ คำว่าส่งหรือรับในที่นี้ให้ยึดเอาตัว CPU หรือ DTE เป็นหลักกว่าจะเป็นผู้ส่งหรือผู้รับ ตามเกณฑ์ของ RS-232 C DTE จะส่งข้อมูลออกมาที่ขา 2 และรับข้อมูลจากขา 3

ขาที่ 4 และ 5 คือขา RTS และ CTS สัญญาณบนขา 4 นี้ DTE ใช้แสดงต่อ DEC เมื่อประสงค์จะส่งข้อมูล สัญญาณ RTS นี้อาจจะใช้เพื่อเปิดเครื่อง DCE จะไม่ส่งข้อมูลจนกระทั่งได้รับสัญญาณ CTS บนขา 5 จาก DCE หรือ CTS เป็นสัญญาณตอบรับจาก DCE ว่าตัว DCE นี้พร้อมในการสื่อสารแล้ว ในกรณีที่ DCE มีความพร้อม และเตรียมคลื่นพาห์ที่จะใช้ในการส่งข้อมูลอยู่แล้วก็ไม่มีความจำเป็นจะต้องหน่วงเวลาระหว่าง RTS และ CTS

ขาที่ 6 และ 20 เป็นขา DSR และ DTR สัญญาณ DSR นี้ DCE ใช้แจ้ง DTE ให้รู้ว่าได้เปิดเครื่องรออยู่แล้ว และก็ได้ปฏิบัติตัวอยู่ในโหมดทดลอง (test mode) กล่าวคือ ชุดสื่อสาร (communication set) นี้เตรียมพร้อมอยู่แล้ว สัญญาณ DTR นี้ใช้เพื่อ DTE แจ้ง DCE ในการพร้อมที่จะตอบรับการสื่อสารที่มีฝ่ายเข้ามาแล้ว

ขาที่ 8 เป็นขาที่ใช้ในการตรวจจับการรับสัญญาณจากสาย บางครั้งอาจเรียกว่า Data Carrier Detect แทนคำว่า Carrier Detect ที่ใช้ในตารางที่ 2.18 การส่งข้อมูลโดยตรงระหว่าง DTE ต่อ DTE ขาที่ 8 นี้ปกติจะถูกต่อโดยตรงกับขาที่ 20

ขาที่ 22 เป็นขา RI (Ring Indicator) สัญญาณนี้เป็นสัญญาณที่ DEC บอก DTE ว่ามีการเรียกโทรศัพท์เข้ามาตามปกติ โหมดี้มจะถูกออกแบบให้เสมือนต่อโดยตรงอยู่กับสายโทรศัพท์ในกรณีที่โหมดี้มเป็นแบบตอบรับอัตโนมัติ โหมดี้มจะมีความสามารถในการตรวจรับสัญญาณเรียกเข้ามาทางโทรศัพท์ได้ และจะส่งสัญญาณ RI สู่ DTE ในขณะที่มีสัญญาณเรียก (ringing tone) เข้ามา และโหมดี้มจะทำการตอบรับโดยการจัดวงจรเสมือนมีการยกหูโทรศัพท์รับ เมื่อได้คำสั่งจาก DTE ซึ่งตามปกติ DTE จะสั่งให้โหมดี้มตอบรับการสื่อสารในโดยใช้สัญญาณ DTE ส่งผ่านขาที่ 20

ปกติสำหรับเครื่องคอมพิวเตอร์เล็ก ๆ ตามท้องตลาด มักจะเกี่ยวข้ออยู่กับขาหัวต่อ 10 ขาที่กล่าวมาแล้วข้างต้นเป็นสำคัญ ขาอื่น ๆ นั้นจะมีความสำคัญรองลงไป

ขาที่ 15, 17, 21 และ 24 จะให้สัญญาณเมื่อทำงานแบบซิงโครไนส์เพราะทางด้านส่งจะต้องส่งข้อมูลบางอย่างไป (0 หรือ 1) ที่แต่ละช่องเวลาบิต (bit time) โหมดี้มจะควบคุมจังหวะสัญญาณนาฬิกาจาก DTE และในทำนองเดียวกันโหมดี้มที่ทำหน้าที่ก็จะต้องส่งบิตข้อมูลและจังหวะสัญญาณนาฬิกาที่ร่วมมาด้วยกันออกมาด้วย ขาที่ 15 และ 17 ใช้สนองความต้องการเหล่านี้และในกรณีที่สัญญาณควบคุมผ่านขาที่ 21 นั้นจะใช้เพื่อแสดงว่าคลื่นพาห์ที่รับเข้ามานั้นมีคุณสมบัติเป็นไปตามเงื่อนไขที่ได้กำหนดไว้ก่อนแล้ว

ขาที่ 23 ใช้เพื่อส่งสัญญาณเลือกอัตราการส่งสัญญาณข้อมูล ในกรณีเป็นชนิดที่สามารถเปลี่ยนอัตราส่งข้อมูล ขานี้จะแจ้งให้ทั้ง DTE ทางด้านส่งและด้านรับให้บอก DTE ทางด้านรับอีกต่อหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

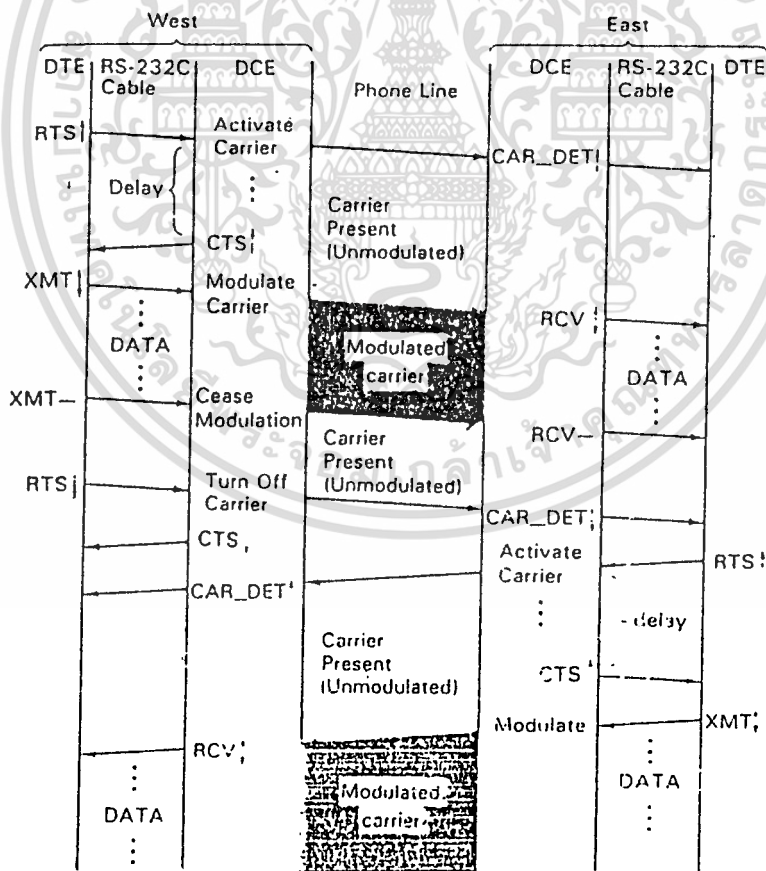
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาที่ 12, 13, 14, 16 และ 19 เป็นขาสัญญาณที่ใช้กับช่องสัญญาณรอง ในโมเด็ม บางเครื่องจะมีช่องสัญญาณใช้สองช่องคือช่องสัญญาณหลักและช่องสัญญาณรอง โดยขาสัญญาณ ทั้ง 5 ขาของช่องสัญญาณรองนั้น จะมีหน้าที่เหมือนกับหน้าที่ทางช่องสัญญาณหลักต่างกันแต่ว่า อัตราการส่งสัญญาณหลักต่างกัน แต่ว่าอัตราการส่งสัญญาณทางช่องสัญญาณรองนั้น ปกติมักจะ ช้ากว่าอัตราการส่งของช่องสัญญาณหลักและช่องสัญญาณรองนั้นจะมีทิศทางการส่งสัญญาณสวน กันกับทิศทางของช่องสัญญาณหลัก

ลักษณะของข้อมูลที่ส่งผ่านอินเทอร์เฟซ RS-232 C นั้นเป็นการส่งแบบลำดับอาจจะ เป็นแบบอะซิงโครไนส์หรือแบบซิงโครไนส์ก็ได้ขึ้นอยู่กับ DTE

2.6.4 การควบคุมคลื่นพาห์และการส่งข้อมูล

กระบวนการควบคุมคลื่นพาห์และการส่งข้อมูลสามารถอธิบายให้เข้าใจได้ง่ายโดยใช้ รูปที่ 2.16 ในรูปนี้ใช้เครื่องหมายลูกศรที่ขึ้นท้ายตัวอักษรย่อแสดงถึงการเปิดส่งสัญญาณ



รูปที่ 2.16 การควบคุมคลื่นพาห์ในระบบสวิตช์พรีเพล็กซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นและจะใช้เครื่องหมายลูกศรพุ่งลงแสดงการปิดเลิกสัญญาณ เครื่องหมายลูกศรสองหัวชี้ขึ้น และลงแสดงถึงการเริ่มรับส่งข้อมูลการส่งข่าวสาร และการตอบรับแสดงโดยลูกศรที่พุ่งผ่าน ข้ามช่องการสื่อสาร โดยเวลาที่ผ่านไปนั้นแสดงโดยการเคลื่อนลงแนวตั้งของรูปโคอะแกรม นั้น เครื่องหมายขีด(-)แสดงสถานะสงบ (quiescent)

เราเริ่มต้นด้วย DTE ด้านตะวันตกกระตุ่น DCE ด้วยสัญญาณ RST ทำให้ DCE ส่ง คลื่นพาห์ไปกระตุ่น CAR-DET ด้านตะวันออก ให้แจ้งให้ DTE ด้านตะวันออกรู้ตัว หลังจาก หนึ่งเวลาประมาณ 150 มิลลิวินาที DCE ด้านตะวันตกจะส่ง CTS ไปแจ้ง DTE ว่าพร้อม แล้วจึงได้มีการส่งข้อมูลเกิดขึ้น ซึ่งข้อมูลจะถูกมอดคูเลทกับคลื่นพาห์ส่งไปยังสถานีตะวันออก เมื่อสถานีตะวันตกส่งข้อมูลหมดก็จะปิดลด RTS ลง คลื่นพาห์ก็จะถูกปิดพร้อมกับ DCE ก็จะ ปิดตัด CAR-DET ลงด้วย ในตอนนี้ถ้า DTE ของสถานีตะวันออกต้องการจะส่งข้อมูลไปถึง สถานีนี้ด้านตะวันตกบ้าง จะต้องปฏิบัติกระบวนการเหมือนกับกระบวนการที่สถานีด้านตะวันตก ได้กระทำมาแล้วทุกประการ จะเห็นได้ว่าการสื่อสารแบบนี้ได้อธิบายอยู่ใน เป็นการสื่อสารที่ เรียกว่า ฮาล์ฟดูเพล็กซ์

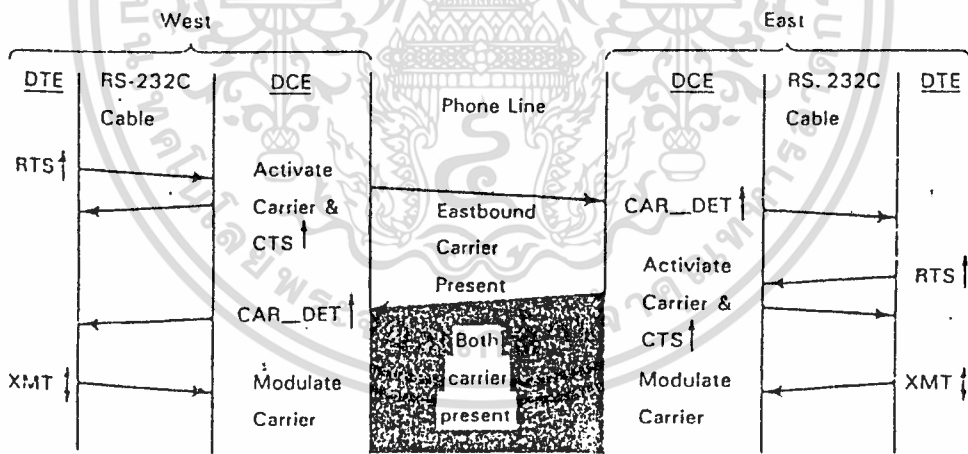
ในระบบฮาล์ฟดูเพล็กซ์ต้องมีการกลับทิศทางของการส่งสัญญาณและในการกลับทิศทางของ การสื่อสารนี้ คลื่นพาห์ในระบบจะต้องถูกจัดส่งใหม่ เพราะคลื่นพาห์ต้องออกจากเครื่องส่ง เสมอ ปรากฏการณ์สลับทิศทางของคลื่นพาห์นี้เมื่ศัพท์เฉพาะเรียกว่า "สายวกกลับ" (line-turn around) ซึ่งจะเกิดขึ้นในระบบฮาล์ฟดูเพล็กซ์เป็นปกติ ดังที่ได้อธิบายมาแล้วว่าการ ที่ DCE จะตอบรับสัญญาณ RTS จาก DTE ด้วย CTS นั้นจะต้องหนึ่งเวลาไว้ก่อนประมาณ 150 มิลลิวินาที ดังนั้นการวกกลับสายแต่ละครั้งต้องใช้เวลาไม่น้อยกว่า 150 มิลลิวินาที และในระบบสื่อสารข้อมูลระยะไกล (long-distance data call) จะเกิดการสะท้อน (echo) ของสัญญาณอยู่เป็นปกติ ในการที่จะวกกลับสายได้จะต้องรอให้คลื่นพาห์ที่เกิดจาก การสะท้อนลดหายไปก่อนด้วย ซึ่งจะต้องรอเวลาประมาณ 150 มิลลิวินาที ดังนั้นช่วงเวลา การวกกลับสายทำให้ประสิทธิภาพในการใช้สายลดลงเป็นอย่างมาก เวลา 150 มิลลิวินาที อาจดูเล็กน้อยแต่ลองพิจารณาถึงระบบที่ใช้อัตราส่งข้อมูล 9,600 บิต/วินาที ที่ใช้รหัสอักษร ซิงโครนัสที่มีความยาว 8 บิต เราจะพบว่าจำนวนอักษรที่ส่งต่อวินาที คือ

$$9,600/8 = 1,200 \quad \text{อักษร/วินาที}$$

ดังนั้นในเวลา 150 มิลลิวินาที จะเป็นเวลาที่ส่งตัวอักษรได้เท่ากับ

$$1,200 \times 0.15 = 180 \quad \text{ตัวอักษร}$$

จะเห็นว่าในช่วงเวลาการวอกกลับสายนี้ทำให้การส่งอักษรช้าลงไปถึง 180 ตัวอักษร ซึ่งปกติแล้วข้อมูล 180 ตัวอักษรนี้จะเป็ขนาดของบล็อกตัวอักษรที่ใช้กันในการสื่อสารข้อมูลธรรมดา ดังนั้นจึงนับว่าการวอกกลับสายนี้จะลดประสิทธิภาพในการใช้สายลงมากมาย ด้วยเหตุดังกล่าวนี้ในการส่งข้อมูลที่มีความเร็วสูงจึงนิยมทำการส่งในระบบฟูลดูเพล็กซ์ ในระบบฟูลดูเพล็กซ์ไม่มีการวอกกลับสาย เพราะว่าคลื่นพาห์ที่ใช้กันสถานีและควมถี่ จึงสามารถรักษาคลื่นพาห์ให้มียุ่บนสายการสื่อสารได้ตลอดเวลาการสื่อสาร เมื่อไม่มีการเปลี่ยแ่งคลื่นพาห์จึงไม่มีความจำเป็นในการสลับสาย ตัวอย่างการส่งข้อมูลระบบฟูลดูเพล็กซ์แบบหนึ่งมีลักษณะดังในรูปที่ 2.17 ตามรูปนี้สถานีด้านตะวันตกได้เริ่มทำการสื่อสารโดยการส่ง RST ไปก่อน DEC ด้านตะวันตกเมื่อรับสัญญาณ RST ก็ส่งคลื่นพาห์หลักไปสู่ DCE ด้านตะวันออก เมื่อตรวจพบคลื่นพาห์บนสาย DCE ก็จะส่ง CAR-DET ไปยัง DTE ด้านเดียวกัน DTE ทางด้านตะวันออกเมื่อได้รับ CAR-DET ก็จะส่ง RST กลับมายัง DCE ให้ส่งคลื่นพาห์รองไปสู่ DCE ด้านตะวันตกพร้อมกับส่ง CTS มาที่ DTE ด้านตะวันออกด้วย เมื่อ DCE ด้านตะวันตกตรวจจับคลื่นพาห์รองได้ ก็จะส่ง CAR-DET ไปบอก DTE ด้านตะวันตกว่ามีการจัดระบบคลื่นพาห์บนสายการสื่อสารเรียบร้อยแล้ว จากช่วงเวลานี้การสื่อสารข้อมูลก็จะเริ่มต้นได้พร้อมกันทั้งสองสถานี

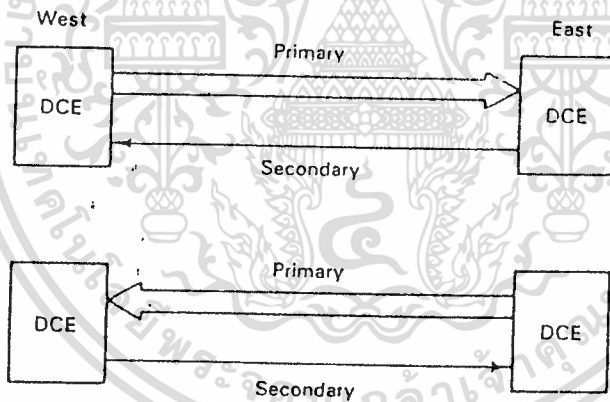


รูปที่ 2.17 การควบคุมคลื่นพาห์ของระบบฟูลดูเพล็กซ์

เพราะคลื่นพาห์นั้นจะปรากฏอยู่บนสายตลอดเวลาที่ทำการสื่อสารจึงไม่มีปรากฏการณ์สายวอกกลับที่เป็นเหตุให้ประสิทธิภาพในการใช้สายเสียไป จึงสามารถปรับปรุงเวลาการส่งผ่านข้อมูลเพิ่มขึ้นเป็นอย่างมาก

2.6.5 ช่องสัญญาณรอง

ดังที่ได้กล่าวมาแล้วว่าหน้าที่ของช่องสัญญาณรองจะคล้ายกับหน้าที่ของช่องสัญญาณหลัก ช่องสัญญาณรองนี้ปกติจะมีบอดเรท (baud rate) ต่ำกว่าบอดของช่องสัญญาณหลัก และใช้ในทิศทางสวนกันกับทิศทางของช่องสัญญาณหลัก ดังนั้นจึงมีชื่ออีกอย่างหนึ่งว่าช่องย้อนกลับ อาจทำการสลับเปลี่ยนการใช้ช่องสัญญาณได้สองแบบดังในรูปที่ 2.18 โดสที่ช่องสัญญาณหลัก (แสดงโดยลูกศรใหญ่) เป็นฮาล์ฟดูเพล็กซ์และถูกควบคุมด้วยวิธีการเหมือนที่ได้อธิบายมาแล้ว เครื่องส่งที่ใช้คลื่นพาห์รองจะถูกขัดสลับ (inter lock) กับคลื่นพาห์รองของสถานีนี้จะไม่มีโอกาสทำปฏิบัติการขึ้นมาได้เลยแม้ว่าสัญญาณร้องขอเพื่อการส่ง (request to sent) ของช่องสัญญาณรองจะถูกยืนยันขึ้นขึ้นมาก็ตาม ดังนั้นจึงเรียกการส่งสัญญาณในระบบตามรูปที่ 2.18 นี้ว่าระบบดูเพล็กซ์สามในสี่ (threequarter duplex) กล่าวคือ ระบบนี้มีการติดต่อกันได้ในสองทิศทางพร้อมกัน แต่ในทิศทางหนึ่งจะมีความจุของการสื่อสารในช่องสัญญาณรองเพียง 5 บอดเท่านั้น เพราะวัตถุประสงค์ของการใช้ช่องสัญญาณรองนี้ก็เพียงเพื่อใช้ในการตอบรับ การติดต่อเกี่ยวกับเรื่องการควบคุมการไหลของสัญญาณ (flow control) และการควบคุมความผิดพลาด (error control) หรือการส่งสัญญาณควบคุมเท่านั้น



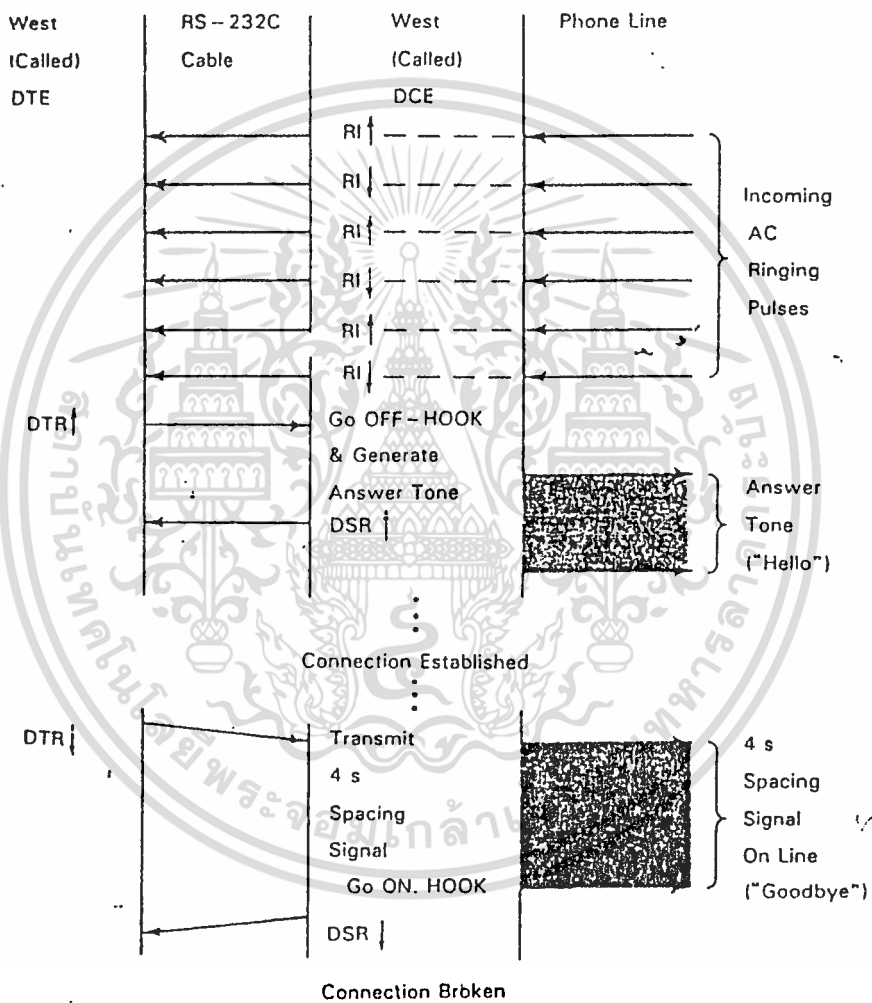
รูปที่ 2.18 การสลับสถานะแบบที่มีช่องย้อนกลับ

2.6.6 การตอบรับอย่างอัตโนมัติ

ในการตอบรับการติดต่ออย่างอัตโนมัติ เราอาจจัดระบบโดยการอาศัยสัญญาณเบรคา DSR, DTR และ RI มาใช้ประโยชน์ โดยมีแผนภูมิเวลาการทำงานแสดงรูปที่ 2.19 ในรูปนี้ เราสมมุติว่ามีการเรียกจากสถานีตะวันออกมายังสถานีตะวันตก สัญญาณกระตุ้นโทรศัพท์ก็จะทำให้ DCE ด้านตะวันตกตรวจจับสัญญาณกระตุ้นได้แล้วส่งสัญญาณ RI ไปหา DTE ทางด้านตะวันตกเพื่อบอกว่ามีการติดต่อเข้ามาเมื่อสัญญาณ RI ได้รับการยืนยันแล้ว 3 ครั้งถ้า DTE ด้านตะวันตกพร้อมจะรับข้อมูลก็จะตอบรับด้วยการยืนยันสัญญาณ DTE มายัง DCE ทางด้าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตะวันตก DCE จะติดต่อวงจรเพื่อเลียนแบบการยกหูโทรศัพท์ พร้อมกับการส่งสัญญาณตอบรับ ซึ่งปกติจะเป็นสัญญาณรูปไซน์ความถี่ประมาณ 2,000 เฮิรตซ์ ติดต่อกันประมาณ 2 วินาที ไป DCE ด้านตะวันออก หลังจากนั้นก็จะยืนยันด้วยสัญญาณ DSR กลับไปยังที่ DTE ด้านตะวันตก เพื่อบอกให้ DTE รู้ว่าสายการสื่อสารถูกต่อให้เรียบร้อยแล้ว จากนั้นจะเป็นกระบวนการติดต่อข้อมูลกันระหว่างสถานีด้านตะวันออกและด้านตะวันตก

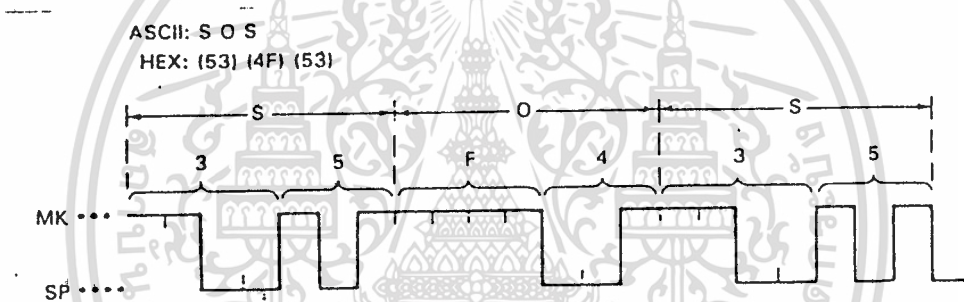


รูปที่ 2.19 DTE ที่รับสัญญาณการติดต่อบริบและหยุดการติดต่อการเรียกทางโทรศัพท์

ในกระบวนการที่จะยกเลิกการใช้ขาสาย DTE จะตัดสัญญาณ DTR ออก DCE ก็ส่งสัญญาณเป็นสเปซไป 4 วินาที เพื่อ"บอกลา" สถานีคู่สื่อสารว่าการติดต่อได้เสร็จสิ้นแล้ว และจะตัดสัญญาณ DSR ออกเป็นการแจ้งให้ DTE จงตัวเองรู้ว่าได้ยกเลิกสายการติดต่อเรียบร้อยแล้ว

2.6.7 การให้จังหวะเวลาแก่วงจรโครนิส

เราได้กล่าวถึงรูปแบบข้อมูลเชิงโครนิสมาบ้างแล้ว ข้อดีของระบบเชิงโครนิสก็คือได้ลดช่วงเวลาดำงระหว่างการส่งตัวอักษรลง ตัวอย่างส่วนหนึ่งของสัญญาณข้อมูลซึ่งตรงกับอักษร SOS ในระบบรหัสแอสกีมีดังแสดงในรูปที่ 2.20 จะเห็นได้ว่าเพราะไม่มีขีดแบ่งระหว่างรหัสตัวอักษรอย่างชัดเจน ดังนั้นถ้าหากข้อมูลติดต่อกันมายาว ๆ เช่น 200 ตัวอักษร ก็จะมีปัญหาเกิดขึ้น ทั้งนี้เพราะไม่มีการใช้บิตเริ่มต้นเหมือนอย่างในระบบอะซิงโครนิส



รูปที่ 2.20 รูปแบบข้อมูลเพื่อความแอสกีแบบเชิงโครนิส

ในระบบอะซิงโครนิส บิตเริ่มต้นจะมีหน้าที่ที่สำคัญ 2 ประการคือ

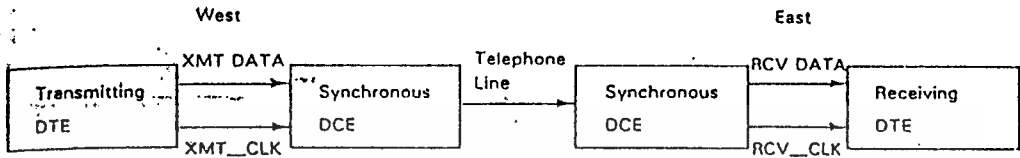
1. แยกตัวอักษรออกเป็นหน่วยที่บ่งชัดออกมา
2. ทำให้เกิดความสัมพันธ์ระหว่างสัญญาณนาฬิกาที่ใช้ในซีพริจิสเตอร์ของภาครับกับที่จุดกึ่งกลางบิตของข้อมูลที่มีเข้ามา

เพื่อให้ระบบเชิงโครนิสทำงานได้จะต้องมีการแก้ปัญหา 2 ประการที่เกิดจากการตัดบิตเริ่มต้นนั้นออกให้ได้ การแก้ปัญหาอันดับแรกเป็นเรื่องของการจัดระบบโปรโตคอลให้เหมาะสม ปัญหาที่สองจะเป็นเรื่องของเชิงโครนิสและ RS-232 C อินเตอร์เฟสเชิงโครนิส DTE จะมีอุปกรณ์ที่แปลงสัญญาณแบบลำดับเป็นสัญญาณแบบขนาน คล้ายกับ UART ในระบบเชิงโครนิส เพราะฉะนั้นจะต้องมีการอาศัยซีพริจิสเตอร์และสัญญาณนาฬิกาสำหรับเลื่อนข้อมูลในการส่งและรับข้อมูลตามตารางที่ 2.18 นั้นจะมีสายสัญญาณนาฬิกาภาคส่งและภาครับอยู่ สัญญาณนาฬิกาภาคส่งจาก DTE จะใช้เลื่อนข้อมูลจากเชิงโครนิส DCE ไปสู่สายการสื่อสารตามจังหวะนาฬิกานั้น และสัญญาณนาฬิกาภาครับจาก DCE จะต้องมีค่าเท่ากับสัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคส่งพอดี้ เพื่อเลื่อนข้อมูลเข้าสู่ DTE ภาครับได้อย่างถูกต้องที่จุดกึ่งกลางบิตพอดี้ ปัญหาที่เกิดขึ้นว่าจะมีการส่งสัญญาณนาฬิกาไปในสายการสื่อสารระหว่าง DCE ได้อย่างไร เพราะสายการสื่อสารมีอยู่คู่เดียว ดังในรูปที่ 2.21 ปัญหานี้ได้แก้โดยอาศัยเทคนิคการมอดคเคเลชั่นที่บรรจุทั้งสัญญาณข้อมูลและสัญญาณนาฬิกาบนคลื่นพาห้คลื่นเดียวกันจากเครื่องส่งไปเครื่องรับ



รูปที่ 2.21 บล็อกไดอะแกรมของการส่งระบบซิงโครนัส

เกี่ยวกับอินเตอร์เฟสดังที่กล่าวมาแล้ว จะเห็นได้ว่ามีรูปแบบให้จัดการได้อย่างหลากหลายตามรูปที่ 2.21 สัญญาณนาฬิกาภาคส่งบนอินเตอร์เฟส RS-232 C ระหว่าง DTE และ DCE ด้านตะวันตก ต้องเป็นสัญญาณที่ใช้กับข้อมูลภาคส่งใน DTE ด้านตะวันตก ตามรูปที่ 2.21 เราใช้สัญญาณนาฬิกาภายใน DTE ทำหน้าที่ดังกล่าว ดังนั้นจะส่งสัญญาณนาฬิกาในผ่านสายที่ต่อขาที่ 24 คือ transmit clock (DTE ด้านส่ง) ของ RS-232 C อย่างไรก็ตามในบางกรณีอาจใช้สัญญาณนาฬิกาจาก DCE ในการควบคุมการรับข้อมูลของ DTE ก็ได้ ดังนั้นในกรณีนี้เราอาจจะใช้ขา transmit clock (DTE ด้านส่ง) ซึ่งเป็นขาที่ 15 ของ RS-232 C แทนขาที่ 24 ตามปกติแล้วจะมีการต่อสวิตช์ดังในรูปที่ 2.22 เพื่อให้ผู้ใช้เลือกแหล่งกำเนิดสัญญาณนาฬิกาได้

การใช้สัญญาณนาฬิกาจาก DCE มีข้อได้เปรียบอยู่บ้างเล็กน้อยคือเราสามารถทำการเปลี่ยนสวิตช์ไปยังบอดเรทที่แตกต่างกันออกไปได้ โดยที่ DTE ไม่ต้องถูกจัดระบบใหม่

2.6.8 ขั้วจำกัดของ RS-232 C

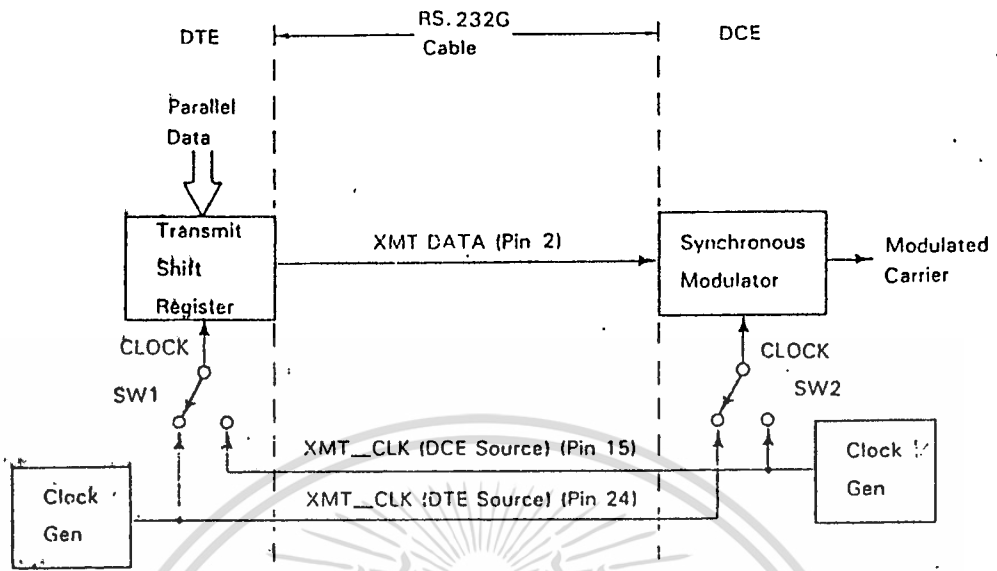
ตัวประกอบหลัก 3 ประการที่จำกัดการใช้งานของอินเตอร์เฟส RS-232 C นั้นก็คือ ระยะทาง อัตราส่งข้อมูล และ สัญญาณรบกวน

2.6.8.1 ระยะทาง

ในเคเบิล RS-232 C ค่าความจุไฟฟ้า ระหว่างสายสัญญาณต่าง ๆ กับสายกราวด์ เป็นพารามิเตอร์หลักที่จำกัดระยะทางการใช้เคเบิลนั้น ตามข้อกำหนดมีว่าค่าความจุไฟฟ้าที่มองไปจากวงจรขับ (drive) จะต้องไม่เกิน 2,500 pF ควรเพิ่มระยะทางของเคเบิลก็จะเพิ่มค่าความจุไฟฟ้าที่เป็นภาระของวงจรขับ ตามปกติเคเบิลที่ใช้อยู่จะมีความจุไฟฟ้า 2,500 pF ที่ความยาวประมาณ 50 ฟุต (15 เมตร)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 การเลือก trans clock ของ ซิงโครไนส์โมเด็ม

2.6.8.2 อัตราส่งข้อมูล

ข้อจำกัดที่ชัดเจนอีกอย่างหนึ่งก็คือ อัตราสัญญาณจะต้องไม่เกิน 20,000 บิต/วินาที ปัญหาที่เกี่ยวเนื่องกับค่าความจุไฟฟ้าของสายเคเบิลอีกเช่นกัน RS-232 C นั้นกำหนดค่าความต้านทานทานอินพุตของวงจรถอดรับไว้ค่อนข้างสูง (3,000-7,000 โอห์ม) เราจะเห็นว่าค่าความต้านทานนี้จำกัดอัตราการประจุ และการคายประจุของ เคเบิลคาปาซิเตอร์ ทำให้เกิดการรบกวน (round) ที่ขอบของสัญญาณและเกิดความผิดเพี้ยนที่อัตราข้อมูลสูง การมีขอบหน้า (leading edge) จะเป็นส่วนสำคัญที่ทำให้ค่าเวลาบิตผิดไป

ค่าจำกัดนี้แตกต่างไปจากทางปฏิบัติเล็กน้อย เพราะที่เร็วที่สุดนั้นจะทำงานในอัตรา 19,200 บิต/วินาที และส่วนใหญ่จะมีอัตราส่งสัญญาณต่ำกว่านี้ มีบางส่วนของสูงกว่าอัตราดังกล่าวนี้ แต่วงจร RS-232 C จะไม่สามารถใช้ได้ในการดังกล่าวนี้

2.6.8.3 สัญญาณรบกวน

สัญญาณรบกวนอาจทำให้สัญญาณข้อมูลผิดพลาดได้ มีสาเหตุจากแหล่งกำเนิด 2 แบบด้วยกัน คือ

1. แหล่งกำเนิดสัญญาณเกิดขึ้นได้แก่ จากมอเตอร์ไฟฟ้า คลื่นสัญญาณจากสถานีส่งวิทยุ หลอดฟลูออโรสเซนส์ เป็นต้น

2. แหล่งกำเนิดภายใน เช่น สายกราวด์สัญญาณรวม(common return signal) เกิดไม่เป็นตัวนำที่สมบูรณ์คือ ค่าความต้านทานไม่เป็นศูนย์ การกระจายของกระแสจากสายสัญญาณต่าง ๆ ที่ใช้สายกราวด์รวมกันจะทำให้เกิดศักย์ไฟฟ้าในลักษณะต่าง ๆ ตกคร่อมสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดสัญญาณรวมนี้และส่งผลไปที่ภาครับสัญญาณ ในลักษณะที่เป็นตัวก้อสัญญาณรบกวนเพิ่มขึ้น จากสี่าเหตุดังกล่าวทั้ง 2 ประการนี้ เมื่อสายเคเบิลยาวมากขึ้นจะก่อให้เกิดสัญญาณรบกวน มากขึ้น สายที่ยาวจะทำตัวคล้ายสายอากาศคอยตรวจจับสัญญาณรบกวนที่เข้ามาและจะเกิด ค่าความต้านทานสูงขึ้นทำให้เกิดสัญญาณรบกวนเพิ่มขึ้น ข้อกำหนดใหม่นี้จะได้รับการปรับปรุง ทำให้อัตราสัญญาณดีขึ้นถึง RS-232 C ลงไม่มากก็น้อย เช่น การใช้วงจรอินเทอร์เฟสสมดุลย์ เข้ามาแทนวงจรอินเทอร์เฟซไม่เฟสสมดุลย์ (unbalanced interface circuit)

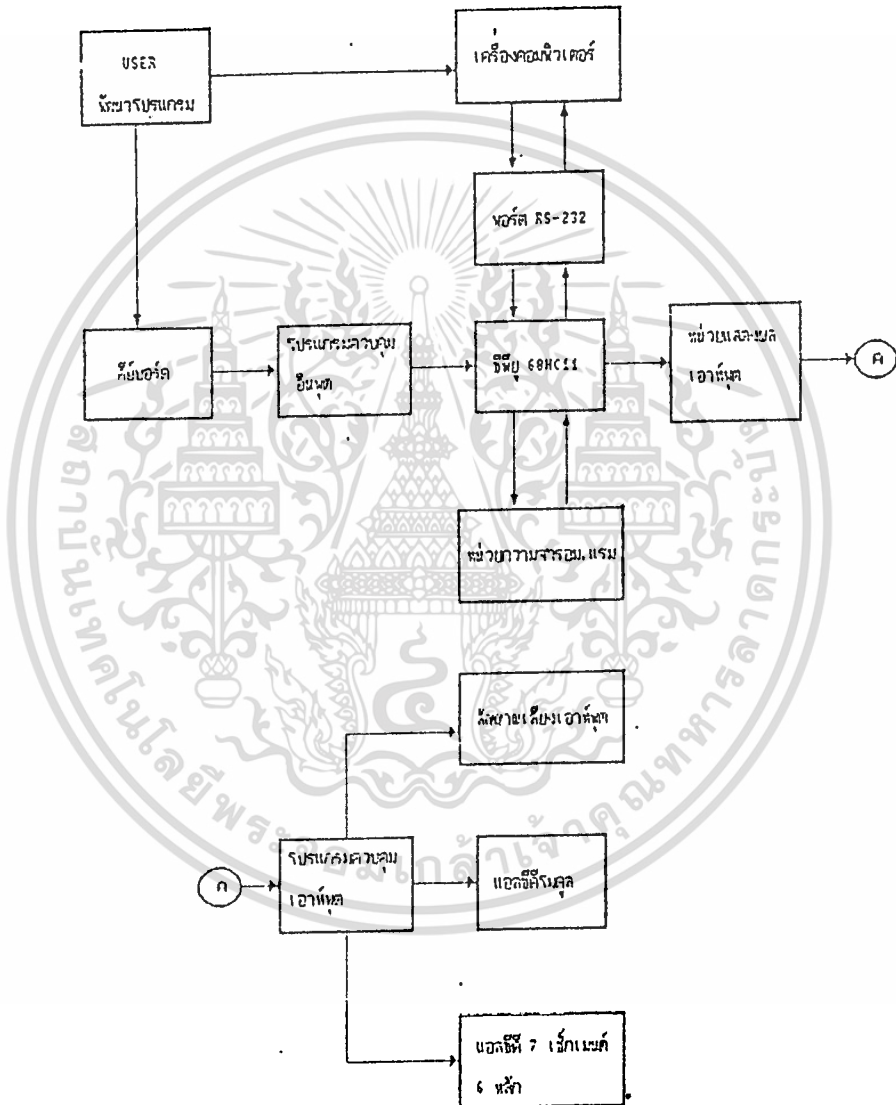
วงจรสมดุลย์นั้นใช้สายคู่ตีเกลียวสำหรับสัญญาณแต่ละสัญญาณและแนวของการออกแบบ วงจรภาคส่งและภาครับต่างไปจากวงจรรูปแบบไม่สมดุลย์ วงจรแบบไม่สมดุลย์นั้นใช้สายเพียง เส้นเดียวสำหรับแต่ละสัญญาณและใช้สายกราวด์ร่วมกัน



บทที่ 3

การลอกแบบและการสร้าง

3.1 บล็อกไดอะแกรมของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11



รูปที่ 3.1 บล็อกไดอะแกรมของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11

จากรูปที่ 3.1 จะเห็นว่ามัลติบอร์ดสามารถทดลองการทำงานหรือพัฒนาโปรแกรมได้ 2 ลักษณะ คือ การพัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์และพัฒนาโปรแกรมบนมัลติบอร์ด เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 พัฒนาโปรแกรมบนเครื่องคอมพิวเตอร์

จะประมวลผลโปรแกรมผ่านพอร์ต RS-232 โดยจะมี DEBUGGER 68HC11 ซึ่งใช้ PROCOMM PLUS ทำหน้าที่ติดต่อกับเครื่องคอมพิวเตอร์ในการสื่อสารกับมัลติบอร์ด โดยโอนโปรแกรมเป็นภาษาแอสเซมบลีและประมวลผลผ่านทางคีย์บอร์ดของเครื่องคอมพิวเตอร์ ในการแสดงผลของการประมวลผลโปรแกรมจะให้มอนิเตอร์ของเครื่องคอมพิวเตอร์

3.1.2 พัฒนาโปรแกรมผ่านคีย์บอร์ดของมัลติบอร์ด

การใช้งานต้องป้อนโปรแกรมในรูปของภาษาเครื่อง ซึ่งบนคีย์บอร์ดของมัลติบอร์ดจะประกอบด้วยคีย์ต่าง ๆ อันได้แก่ คีย์คำสั่ง, คีย์ข้อมูล และคีย์ฟังก์ชันอำนวยความสะดวกในการเขียนโปรแกรมอย่างพร้อมมูลอยู่ นอกจากนี้ยังมีโปรแกรมสำเร็จการประยุกต์ 68HC11 ทั่วไปใช้งาน ซึ่งจะอยู่ในรูปฟังก์ชันคีย์อีกด้วย

3.2 ส่วนประกอบโดยทั่วไป

มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 มีส่วนประกอบที่สำคัญ 9 ส่วนด้วยกัน คือ

3.2.1 ส่วนชุดควบคุม

เป็นส่วนที่ควบคุมการทำงานในส่วนต่าง ๆ ของมัลติบอร์ดตามโปรแกรมมอนิเตอร์ที่เขียนบนคีย์บอร์ด CP-68HC11

3.2.2 ส่วนทรานส์พอร์เตอร์ภาค

เป็นส่วนสร้างสัญญาณสำหรับการเลือกการติดต่อกับอุปกรณ์ภายนอกที่แยกแยะไปจากการสร้างสัญญาณเลือกอุปกรณ์บนชุดควบคุม

3.2.3 ส่วนคีย์บอร์ด

เป็นส่วนที่รับค่าหรือเป็นอินพุตที่สำคัญของมัลติบอร์ด โดยค่าที่รับนี้จะเป็น 2 รหัส คือ รหัสคำสั่งและรหัสข้อมูล

3.2.4 ส่วนการแสดงผล 7 เซกเมนต์และแฟล็กแอลอีดี

ทำหน้าที่ให้มอนิเตอร์ของมัลติบอร์ดและแฟล็กแอลอีดี จะแสดงค่าในบิตต่าง ๆ ของรีจิสเตอร์แฟล็กในขณะประมวลผลหรือประมวลผลแบบ STEP

3.2.5 ส่วนการแสดงผลแอลซีดี

เป็นส่วนที่ทวักเสวยมีการแสดงผลที่เกินขอบเขตของ 7 เซกเมนต์ โดยส่วนใหญ่มักจะใช้

เอกสารนี้แสดงในฟังก์ชันสำคัญของโปรแกรมตัวก่อน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 ส่วนการแสดงผลทางลำโพง

เป็นส่วนที่แสดงผลในรูปแบบของเสียงขณะทำงานแอมัลติบอร์ด

3.2.7 ส่วนของการอินเตอร์เฟซกับอุปกรณ์ภายนอก

เป็นส่วนที่ใช้เชื่อมต่อเพื่อรับสัญญาณหรือควบคุมอุปกรณ์ต่าง ๆ ที่เชื่อมต่อกับชุดควบคุม เพื่อให้การทำงานสมบูรณ์แบบยิ่งขึ้น

3.2.8 ส่วนของแหล่งจ่ายไฟ

ทำหน้าที่แปลงแรงดันไฟเอซี 220 โวลต์ เป็นแรงดันไฟกระแสตรง 12 โวลต์ แล้วผ่านวงจรเรกกูเลเตอร์จ่ายไฟกระแสตรง 5 โวลต์ส่งไปเลี้ยงวงจรในส่วนต่าง ๆ

3.2.9 ส่วนของบอร์ดทดลอง

เป็นส่วนที่จัดไว้สำหรับการทดลองต่าง ๆ มีแผ่นวงจรพิมพ์เอนกประสงค์เฟลทูกูโพลด์ และคอนเนคเตอร์ขนาด 40 ทา ในการเชื่อมต่อกับบอร์ด 8255 และสัญญาณควบคุมต่าง ๆ

3.3 การออกแบบและการทำงานในส่วนต่าง ๆ

3.3.1 ส่วนชุดควบคุม

คุณสมบัติทางเทคนิคของของมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 ได้แก่

ไมโครคอนโทรลเลอร์ : MC68HC11A1FN

หน่วยความจำ : แรม 6264 (8 กิโลไบต์) หรือ 62256 (32 กิโลไบต์)

ทำงานที่แอดเดรส \$2000-\$9FFF ให้อัฒรวม 2764 ขนาด 8 กิโลไบต์ หรือ 27256 (32 กิโลไบต์) แต่จะทำงานได้สูงสุด 24 กิโลไบต์ และมีช่วงแอดเดรสคือ \$A000-\$FFFF มีแรมภายใน 256 ไบต์และอีพีรวมขนาด 512 ไบต์

พอร์ตอินพุตเอาต์พุต : ขนาด 24 บิต โดยให้ชีพ 8255

พอร์ตเครื่องพิมพ์ : มีจำนวน 1 พอร์ต

พอร์ตเชื่อมต่อคีย์บอร์ด : แบบ Matrix ขนาด 3x4 หรือ 4x4 จำนวน 1 พอร์ต

พอร์ตแอลซีดี : มี 1 พอร์ต ใช้ได้ทั้งแบบตัวอักษรและแบบกราฟฟิก

พอร์ตตัวตั้งเวลา : มีพอร์ตอินพุต 3 ช่อง, พอร์ตเอาต์พุต 4 ช่อง และยังมีพอร์ตอินพุตเอาต์พุตอีก 4 ช่อง

พอร์ต A/D : มีพอร์ตแปลงสัญญาณอนาล็อกเป็นดิจิทัล 8 บิต 8 ช่อง

พอร์ตขนาน : มีพอร์ตขนาน 2 ช่อง แบบ SCI และ SPI

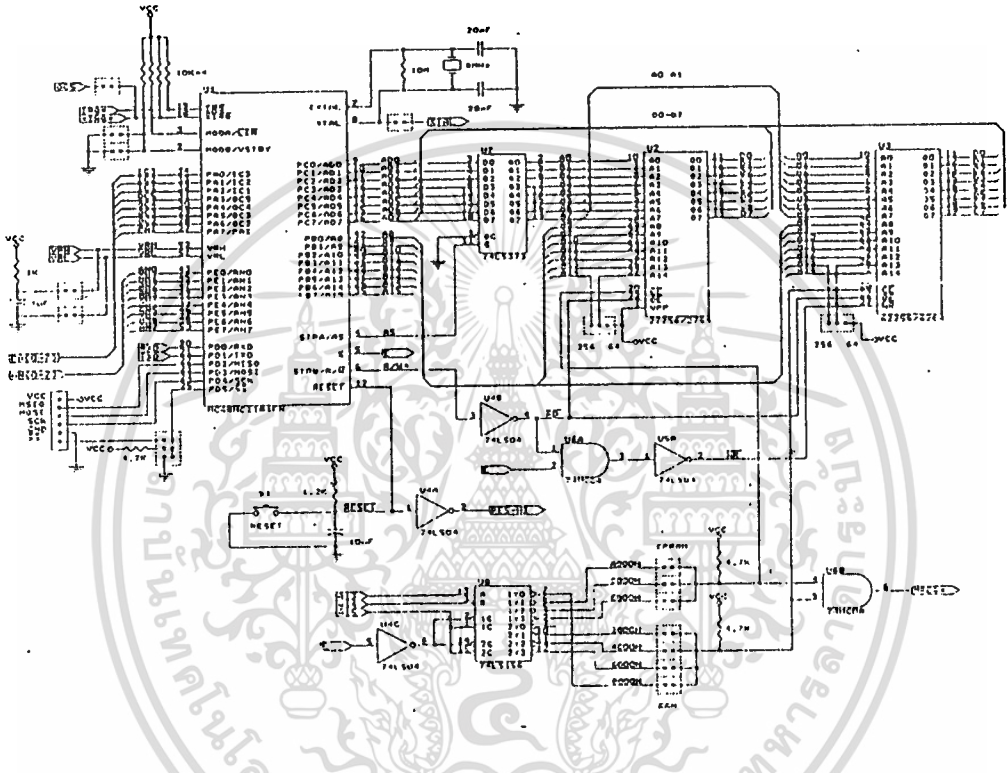
สัญญาณนาฬิกา : ให้สัญญาณนาฬิกา 8 MHz

เอกสารนี้เป็นเอกสารของชุดควบคุมกับภา : 0.3 x 14.1 เซนติเมตร ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.1 การทำงานของชุดควบคุม

หัวใจการทำงานการทำงานของบอร์ดขึ้นอยู่กับ U1 โดยไมโครคอนโทรลเลอร์ 68HC11 สามารถเลือกโหมดการทำงานให้แก่ U1 ได้โดยการเลือกจัมเปอร์ที่ขา 2 และ 3 ของ U1 ดังรูปที่ 3.2



รูปที่ 3.2 วงจรเชื่อมต่อ 68HC11 กับหน่วยความจำภายนอก

จากรูปที่ 3.2 ไมโครคอนโทรลเลอร์ 68HC11 นี้สามารถเชื่อมต่อกับหน่วยความจำทั้งอ็พรมและแรมได้โดยการเลือกโหมดการทำงานเป็นแบบมัลติเพล็กซ์ชยาย ในการเลือกโหมดนี้ขา PC0-PC7 ทำหน้าที่เป็นบัสแอดเดรสไบต์ต่ำสลับกับการเป็นบัสข้อมูล โดยมี U7 เบอร์ 74LS373 เป็นตัวแยกสัญญาณหรือเรียกว่าเป็นตัวดีมัลติเพล็กซ์ก็ได้ ซึ่งจะต้องทำงานร่วมกับสัญญาณ AS ที่ขา 4 ของ U1 ด้วย ส่วนขา PBO-PB7 จะเป็นขาของบัสแอดเดรสไบต์สูงคือ A8-A15

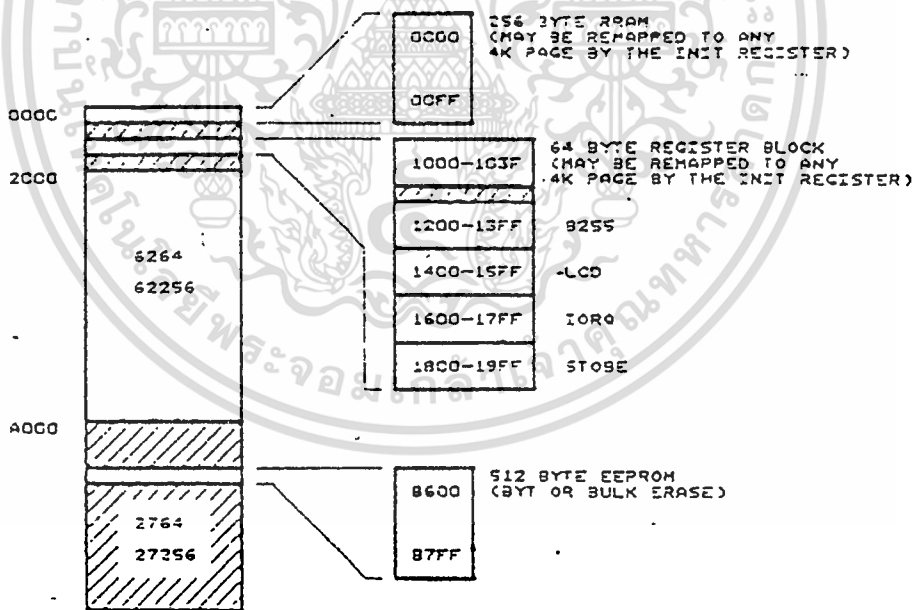
U4B, U6A และ U5A จะทำหน้าที่เป็นวงจรสร้างสัญญาณ WR เพื่อเขียนข้อมูลลงในหน่วยความจำแรม U3 ซึ่งสามารถเลือกได้ว่าเป็นแรมเบอร์ 6264 (8 กิโลไบต์) หรือว่าเบอร์ 62256 (32 กิโลไบต์) ก็ได้ เอาต์พุตจาก U4B ยังใช้เป็นสัญญาณ RD เพื่อทำการส่งผ่านข้อมูลจากอ็พรม U2 และแรม U3

ในวงจรถอดรหัสแอดเดรสของหน่วยความจำจะเป็นหน้าที่ของ U8 เบอร์ 74LS156 สามารถเลือกแอดเดรสได้ตั้งแต่ \$2000-\$9FFF นอกจากนี้ยังมีการนำสัญญาณจากที่ U8 มาจัดเป็นสัญญาณ MREQ สำหรับบัสของระบบเพื่อประโยชน์ในการเชื่อมต่อกับอุปกรณ์อื่น ๆ ของ Z-80

วงจรถ่ายทอดสัญญาณนาฬิกาของ U1 จะใช้พิกัดคริสตอลขนาด 8 MHz เป็นตัวกำเนิดความถี่ และสามารถนำสัญญาณนาฬิกาออกไปใช้งานกับฮาร์ดแวร์ภายนอกได้

3.3.1.2 การจัดหน่วยความจำของบอร์ด

ที่บอร์ด CP-68HC11 จะมีการจัดสรรหน่วยความจำ ดังแสดงในรูปที่ 3.3 ส่วนแรก มีขนาด 256 กิโลไบต์ ที่แอดเดรส \$0000-\$00FF จัดสรรไว้สำหรับแรมภายในชิพ โดยที่แอดเดรส \$1000-\$103F เป็นพื้นที่ของรีจิสเตอร์ควบคุมขนาด 64 กิโลไบต์ ส่วนที่ 3 แอดเดรสตั้งแต่ \$12000-\$13FF เป็นแอดเดรสของพอร์ต 8255 ใช้สำหรับแอดเดรสของพอร์ตแอลซีดีจัดสรรไว้ที่ 1400-\$15FF



MEMORY MAP

รูปที่ 3.3 การจัดหน่วยความจำของบอร์ด

ที่แอดเดรส \$1600-\$17FF เป็นของสัญญาณ IORQ และแอดเดรส \$1800-\$19FF

เป็นของสัญญาณสโตรบ ในแอดเดรสระหว่าง \$B600-\$B7FF เป็นหน่วยความจำอีพროม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่เสียค่าใช้จ่าย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในชิพ 68HC11 สดท้ายและแอดเดรส \$A000-\$FFFF เป็นพื้นที่สำหรับหน่วยความจำ อีพรอมภายนอก ซึ่งก็คือ U2 นี้เอง

3.3.1.3 ระบบบัส

เป็นคอนเนคเตอร์ขนาด 40 ขาใช้ในการขยายระบบ โดยมีการจัดขาสัญญาณต่าง ๆ ให้ใกล้เคียงกับบัส 40 ขาของ Z-80

3.3.1.4 การเลือกหน่วยความจำแรม

สามารถเลือกแรมได้ 2 เบอร์คือ 6264 ขนาด 8 กิโลไบต์ และ 62256 ขนาด 32 กิโลไบต์ โดยเลือกจัมเปอร์ J1 สำหรับการกำหนดตำแหน่งของหน่วยความจำสามารถเลือกได้ที่จัมเปอร์ J3 ซึ่งการใส่จัมเปอร์ 1 ตัว สามารถอ้างตำแหน่งในหน่วยความจำได้ 8 กิโลไบต์ ถ้าใส่ครบทั้ง 4 ตัวก็สามารถอ้างตำแหน่งได้ 32 กิโลไบต์ โดยที่ตำแหน่งของหน่วยความจำจะอยู่ในช่วง \$2000 - \$9FFF

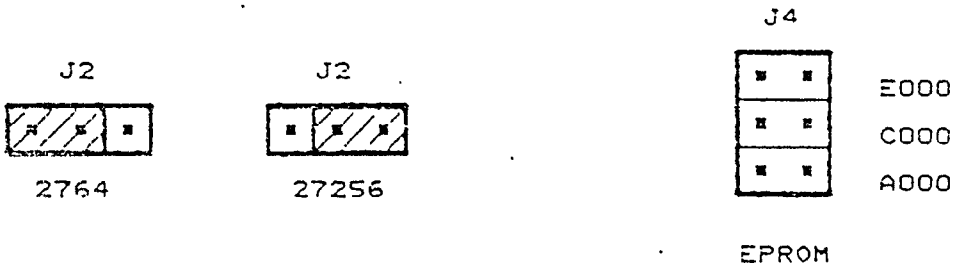


รูปที่ 3.4 การเลือกจัมเปอร์ของแรม

3.3.1.5 การเลือกหน่วยความจำอีพรอม

เราสามารถที่จะเลือกใช้อีพรอมได้ 2 เบอร์ คือเบอร์ 2764 (8 กิโลไบต์) และ เบอร์ 27256 (32 กิโลไบต์) โดยการเลือกที่จัมเปอร์ J2 สำหรับการกำหนดตำแหน่งของหน่วยความจำสามารถเลือกได้ที่จัมเปอร์ J4

ซึ่งการใส่จัมเปอร์ 1 ตัว สามารถอ้างตำแหน่งในหน่วยความจำได้ 8 กิโลไบต์ ถ้าใส่ครบทั้ง 3 ตัว จะสามารถอ้างตำแหน่งในหน่วยความจำได้ 24 กิโลไบต์ โดยที่ตำแหน่งหน่วยความจำจะอยู่ในช่วง \$A000-\$FFFF โดยการเลือกจัมเปอร์แสดง ดังรูปที่ 3.5



รูปที่ 3.5 การเลือกจัมเปอร์ของอีพรอม

3.3.1.6 แอดเดรสของพอร์ตอินพุตเอาต์พุต 8255

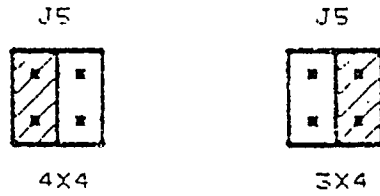
เป็นคอนเนคเตอร์ 34 ขา ที่ต่อออกมาจากพอร์ตของ 8255 สำหรับแอดเดรสติดต่อกับพอร์ตของ 8255 มีรายละเอียดดังรูปที่ 3.6

PORT A	\$1200
PORT B	\$1201
PORT C	\$1202
CONTROL PORT	\$1203

รูปที่ 3.6 แอดเดรสของพอร์ตอินพุตเอาต์พุต 8255

3.3.1.7 การเชื่อมต่อกับคีย์บอร์ด

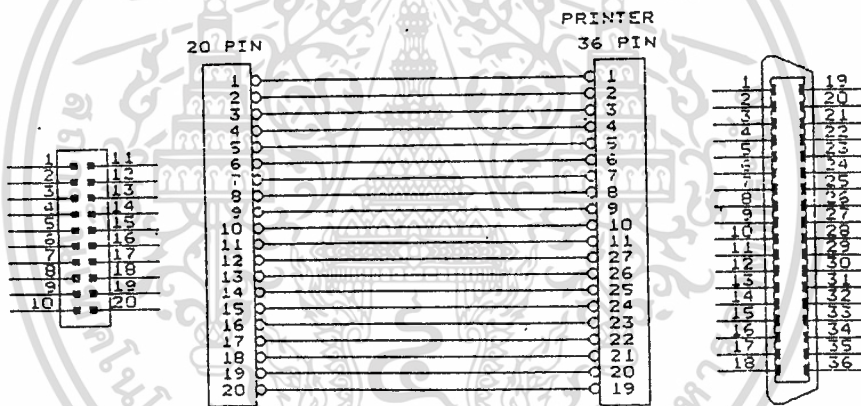
เป็นคอนเนคเตอร์ขนาด 10 ขา ที่ต่อมาจากพอร์ต C ของ 8255 สามารถต่อเข้ากับคีย์บอร์ดขนาด 4x4 หรือ 3x4 ได้โดยทำการเลือกจัมเปอร์ J5 ในกรณี que เลือกคีย์บอร์ดเป็นแบบ 3x4 ขา PC0 ของ 8255 จะถูกนำมาใช้เป็นสัญญาณ BUSY เพื่ออ่านสถานะของเครื่องพิมพ์ ส่วนในกรณีที่เลือกคีย์บอร์ดเป็นแบบ 4x4 จะไม่สามารถเชื่อมต่อกับเครื่องพิมพ์ได้ เนื่องจากขา PC0 ของ 8255 นี้จะถูกต่อเข้ากับคอนเนคเตอร์ 10 ขา ของคีย์บอร์ดแทนการเลือกจัมเปอร์ ดังแสดงในรูปที่ 3.7



รูปที่ 3.7 การเลือกลักษณะของคีย์บอร์ดโดยใช้ J5

3.3.1.8 แอดเดรสของพอร์ตเครื่องพิมพ์

เป็นคอนเนคเตอร์ขนาด 20 พิน ใช้ต่อกับเครื่องพิมพ์ธรรมดาทั่วไป สำหรับสัญญาณ STROBE นี้จะใช้พอร์ตเอาต์พุตของ 74HC138 ซึ่งได้จากการถอดรหัสแอดเดรสของพอร์ต โดยการส่งข้อมูลอะไรก็ได้ส่งออกไปที่พอร์ตแอดเดรสที่ \$1800 ก็จะเป็นการสวิตช์ข้อมูล สำหรับแอดเดรสที่ใช้ติดต่อกับเครื่องพิมพ์มีรายละเอียดดังนี้



รูปที่ 3.8 ลักษณะการเชื่อมต่อคอนเนคเตอร์ 20 พิน เข้ากับคอนเนคเตอร์ เซ็นเซอร์อิเล็กทรอนิกส์ขนาด 36 พิน ของเครื่องพิมพ์

3.3.1.9 การเชื่อมต่อกับจอแสดงผลแบบแอลซีดี

เป็นคอนเนคเตอร์ขนาด 20 พิน สามารถใช้กับโมดูลแอลซีดีได้ 2 แบบ คือ แบบตัวอักษรและแบบกราฟิกโดยเลือกที่จัมเปอร์ J6 ดังรูปที่ 3.9 และสามารถปรับความสว่างของจอแอลซีดีได้โดยการปรับที่ตัวต้านทานแปรค่าได้ 10 กิโลโอห์ม แอดเดรสที่ใช้ในการติดต่อกับแอลซีดีมีรายละเอียดสำหรับแอลซีดีแบบกราฟิกดังนี้

WRITE DATA INSTRUCTION	\$1400
READ BUSY FLAG ANND ADDRESS	\$1400
WRITE DATA	\$1401
READ DATA	\$1401

Page 1 อยู่ที่แอดเดรส \$1400-\$1401

Page 2 อยู่ที่แอดเดรส \$1402-\$1403



รูปที่ 3.9 การเลือก J6 เพื่อเลือกชนิดของโมดูลแอลซีดี

3.3.1.10 พอร์ตเชื่อมต่อวงจร A/D

มีคอนเนคเตอร์ขนาด 14 ขา ซึ่งต่อออกมาจากขา PE0-PE7 ของ 68HC11 เป็นอินพุตของวงจรแปลงสัญญาณอนาล็อกให้เป็นดิจิทัล (A/D) ขนาด 8 บิต 8 ช่อง โดยจะใช้แรงดันอ้างอิงภายในบอร์ด 5 โวลต์ สามารถใช้แรงดันอ้างอิงจากภายนอกได้ ให้แรงดันนี้อยู่ในช่วง 2.5-5 โวลต์โดยการถอดจัมเปอร์ J9



รูปที่ 3.10 การต่อแรงดันอ้างอิงโดย A/D โดย J9

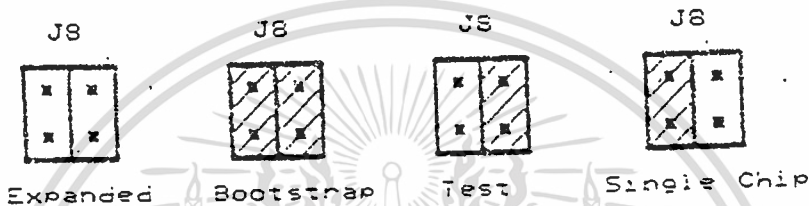
3.3.1.11 พอร์ตตัวตั้งเวลาและตัวนับ

เป็นคอนเนคเตอร์ขนาด 10 ขา ต่อออกมาจาก PA0-PA7 ของ 68HC11 สามารถนำไปประยุกต์ใช้งานอื่น ๆ ได้หลายอย่างเช่น การนับความถี่, นับจำนวนพัลส์คาบเวลา, กำเนิดสัญญาณนาฬิกาการป้อนให้เลี้ยวหรือจะใช้เป็นพอร์ตอินพุตเอาต์พุตก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.12 การเลือกโหมดการทำงาน

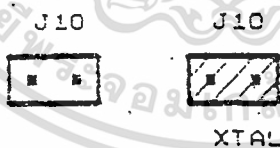
มีคอนเนคเตอร์ 4 ขา เพื่อที่จะใช้เลือกโหมดการทำงานของไมโครคอนโทรลเลอร์ ซึ่งสามารถกำหนดโหมดการทำงานได้ 4 โหมด คือ โหมดซิงเกิลชิพ (Single chip), โหมดมัลติเพล็กซ์ขยาย (Expanded multiplex), โหมดบูตสแตรปพิเศษ (Special-bootstrap) และโหมดทดสอบพิเศษ (Special test) แต่การทำงานโดยทั่วไปจะอยู่ที่ โหมดมัลติเพล็กซ์ขยายซึ่งเป็นการติดต่อกับหน่วยความจำภายนอก การเลือกโหมดจะกระทำที่จัมเปอร์ J8 ดังรูปที่ 3.11



รูปที่ 3.11 การเลือกโหมดการทำงานของ 68HC11

3.3.1.13 การนำสัญญาณนาฬิกาออกไปใช้งานภายนอก

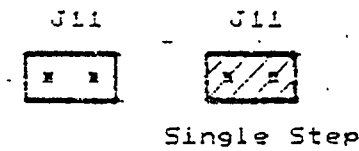
ในกรณีที่ต้องการสัญญาณความถี่ 8 MHz ออกไปยังคอนเนคเตอร์ 40 ขา เพื่อเชื่อมต่อบอร์ดวงจรภายนอกจะต้องลัดจัมเปอร์ J10 ด้วยการนำสัญญาณนาฬิกาออกไปใช้งานก็เพื่อให้ ระบบฮาร์ดแวร์ภายนอกกับบอร์ดมีจังหวะการทำงานที่สัมพันธ์สอดคล้องกัน



รูปที่ 3.12 การเลือกจัมเปอร์ J10

3.3.1.14 การลัดจัมเปอร์ JOC5

เป็นคอนเนคเตอร์ 2 ขา ซึ่งต่ออยู่ระหว่างขา OC5 กับขา XIRQ ของ 68HC11 ซึ่งจะถูกใช้ในกรณีที่ต้องการทำประมวลผลแบบ Single Step เป็นการทำให้ Single Step โดยฮาร์ดแวร์ ซึ่งจะทำให้เกิดอินเตอร์รัพต์ขึ้นที่ขา XIRQ ดังนั้นการนำขา OC5 ไปใช้งานอื่น ๆ ควรนำจัมเปอร์นี้ออกได้

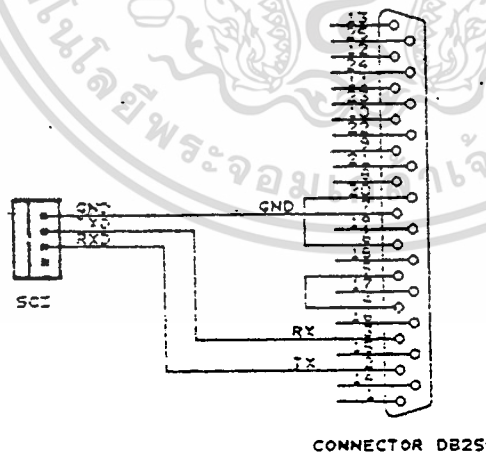


รูปที่ 3.13 การเลือกจัมเปอร์, J11

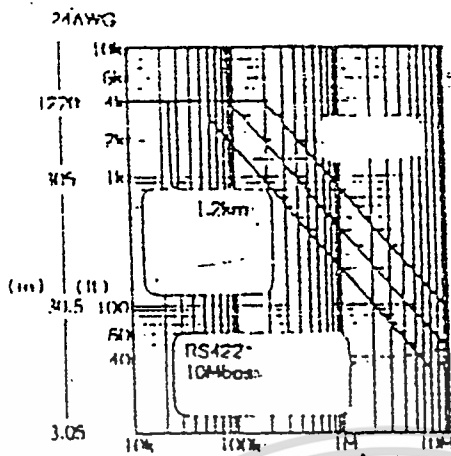
3:3.1.15 การส่งผ่านข้อมูลอนุกรมและพอร์ตสำหรับสื่อสารข้อมูลอนุกรม (SCI)

การเชื่อมต่อกับพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ผ่านทาง RS-232 จะใช้ U9 เบอร์ MAX232 เป็นทำหน้าที่นี้และถ้าต้องการเชื่อมต่อผ่านพอร์ต RS-232 ให้เปลี่ยน U9 จาก MAX232 เป็น 75176 แทน และต้องต่อให้ตัวต้านทาน R7 ด้วย โดยค่าของ R7 จะขึ้นอยู่กับความยาวของสายที่ต่อใช้งานกับพอร์ต RS-422 โดยปกติพอร์ต RS-422 สามารถจะใช้สายได้ยาวสูงสุด 4,000 ฟุต หรือประมาณ 1.2 กิโลเมตร และอัตราบอดในการส่งข้อมูลปกติแล้วจะใช้อัตราบอดเท่ากับ 9,600 บิตต่อวินาทีที่ระยะทาง 30.5 เมตร ค่า R7 จะอยู่ที่ 100 โอห์ม ถ้าระยะทางมากขึ้นค่า R7 ก็จะมีเพิ่มขึ้น ในอัตราบอดที่เท่าเดิม

คอนเนคเตอร์ขนาด 4 ขาที่ต่อเข้ากับ MAX232 เป็นการเชื่อมต่อกับพอร์ต RS-232 เพื่อใช้งานสื่อสารข้อมูลอนุกรม เราสามารถเชื่อมต่อกับพอร์ต RS-232 ได้ โดยการถอด MAX232 ออกแล้วใส่ 75176 แทนแล้วกำหนดค่า Rz ใหม่ ดังรูปที่ 3.15



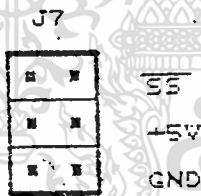
รูปที่ 3.14 การเชื่อมต่อ 68HC11 เพื่อส่งข้อมูลอนุกรม (SCI) โดยการต่อพอร์ตอนุกรมกับคอนเนคเตอร์ DB25



รูปที่ 3.15 กราฟสำหรับการเลือกค่าความต้านทาน Rz

3.3.1.16 พอร์ตสำหรับเชื่อมต่ออุปกรณ์อนุกรม (SPI)

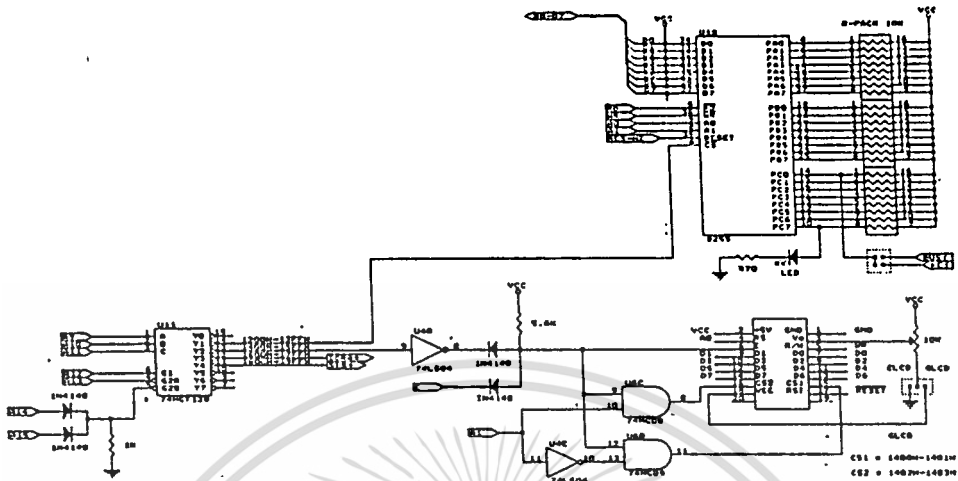
เป็นพอร์ตทำงานด้านเชื่อมต่อกับอุปกรณ์อนุกรม โดยจัมเปอร์ J7 เป็นตัวเลือกว่าจะใช้ต่อกับขา SS ของ 68HC11 ไปต่อเข้ากับแรงดัน +5 โวลต์, กราวด์ (GND) หรือต่อไปคอนเนคเตอร์



รูปที่ 3.16 จัมเปอร์ J7

3.3.1.17 การขยายพอร์ตเพิ่มเติมบนชุดควบคุม

ในส่วนของการขยายพอร์ตเพิ่มเติม U11 ในเบอร์ 74HCT138 จะทำหน้าที่ถอดรหัสแอดเดรสพอร์ตให้แก่ 68HC11 โดยพอร์ตเพิ่มเติมนั้นจะใช้ U12 เบอร์ 8255 นอกจากนั้น U11 ยังถอดรหัสแอดเดรสสำหรับพอร์ตที่เชื่อมต่อกับแอลซีดีโมดูลด้วย โดยจะทำงานร่วมกับ U4D, U4E, U6C และ U6D ความต้านทานปรับค่าได้ 10 กิโลโอห์ม จะปรับความสว่างของแอลซีดีโมดูล



รูปที่ 3.17 การขยายพอร์ตเพิ่มเติมบนชุดควบคุม

3.3.1.18 การสร้าง

บอร์ด CP-68HC11 ใช้แผ่นวงจรพิมพ์ 2 หน้า ชนิดเพลตทรูโฮลด์ ความสูงยักอยู่ที่ การติดตั้ง U1 68HC11 เพราะเป็นตัวถังแบบ PLCC ต้องติดตั้งซ็อกเก็ตก่อน ตำแหน่งและ ทิศทาง ดูดังภาคผนวก

การลงอุปกรณ์ควรวใส่อุปกรณ์ที่มีขนาดเล็กก่อนเช่น ตัวต้านทาน ไดโอด ตัวเก็บประจุ และใส่ให้ถูกขั้วด้วย สำหรับการใส่ซ็อกเก็ต 52 ขา PLCC พื้นของซ็อกเก็ตและที่ไอซีจะมี รอยบากอยู่ ให้รอยบากนี้อยู่ทางด้านซ้ายบนโดยจะใส่ไอซีให้รอยบากอยู่ทางด้านเดียวกัน ที่ ตัวไอซีจะมีจุดวงกลมเล็ก ๆ อยู่ ในรูปการจัดวางอุปกรณ์ในภาคผนวกจะแสดงตำแหน่งขา 1 ของไอซี สำหรับการใส่คอนเนคเตอร์ของพอร์ตอนุกรมทั้ง 2 ช่องนั้น ให้หันด้านพลาสติก เข้าด้านใน ด้านที่เป็นขั้วคอนเนคเตอร์จะหันออกด้านนอกแผ่นวงจรพิมพ์

3.3.2 ส่วนถอดรหัสพอร์ตภายนอก

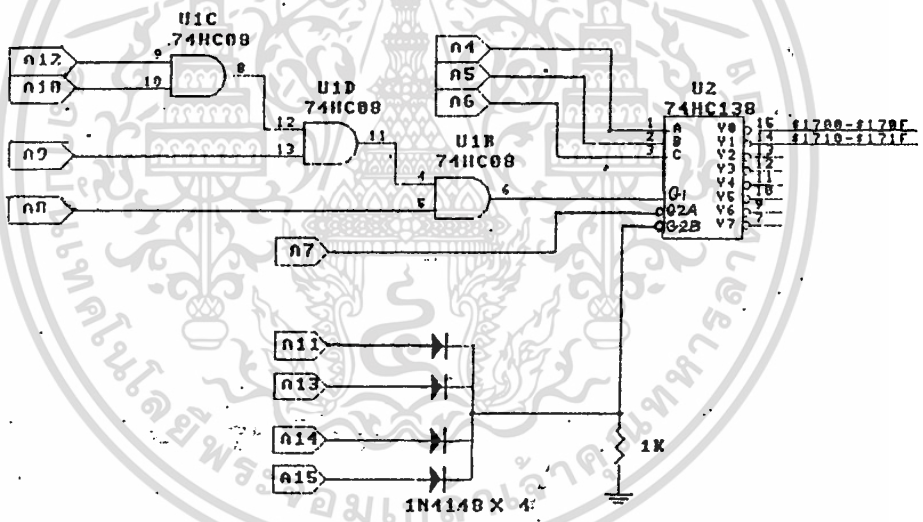
ในตารางที่ 3.1 เมื่อเราต้องการที่จะเปลี่ยนหมายเลขพอร์ตจาก \$1700-\$170F เป็นหมายเลขพอร์ตที่ \$1710-\$171F นั้น ลอจิกที่แอดเดรส A4 จะเปลี่ยนแปลงจาก "0" มาเป็น "1" เพราะฉะนั้นเราจึงต่อแอดเดรส A4, A5, A6 เข้าที่ขา A, B และ C ของไอซีมัลติเพล็กซ์เบอร์ 74138 ที่เป็นไอซีที่เรียกว่า 3 to 8 line demultiplex เพื่อทำการเปลี่ยนแปลงช่วงของพอร์ตทางเอาต์พุต Y ไอซีนี้มีสัญญาณควบคุมขั้ว 3 สัญญาณ คือ G1, G2A และ G2B สัญญาณ G1 จะทำงานที่ลอจิก "1" เราจึงนำขาแอดเดรส A12, A10, A9 และ A8 ที่ให้ลอจิกเป็น "1" ตลอดในการเลือกหมายเลขพอร์ตทั้งสองช่วงผ่าน แอนด์เกตเบอร์ 74HC08 แล้วป้อนลอจิกให้แก่ G1 ส่วนที่ขา G2A จะทำงานที่ลอจิก "0"

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้สำหรับบริการเชิงพาณิชย์เท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่ระบุชื่อเป็นการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุต	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	ตำแหน่ง
Y0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	S1710
	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	1	S1701
	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	S1702
	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	1	S1703
Y1	0	0	0	1	0	1	1	1	0	0	0	1	0	0	0	0	S1710
	0	0	0	1	0	1	1	1	0	0	0	1	0	0	0	1	S1711
	0	0	0	1	0	1	1	1	0	0	0	1	0	0	1	0	S1712
	0	0	0	1	0	1	1	1	0	0	0	1	0	0	1	1	S1713

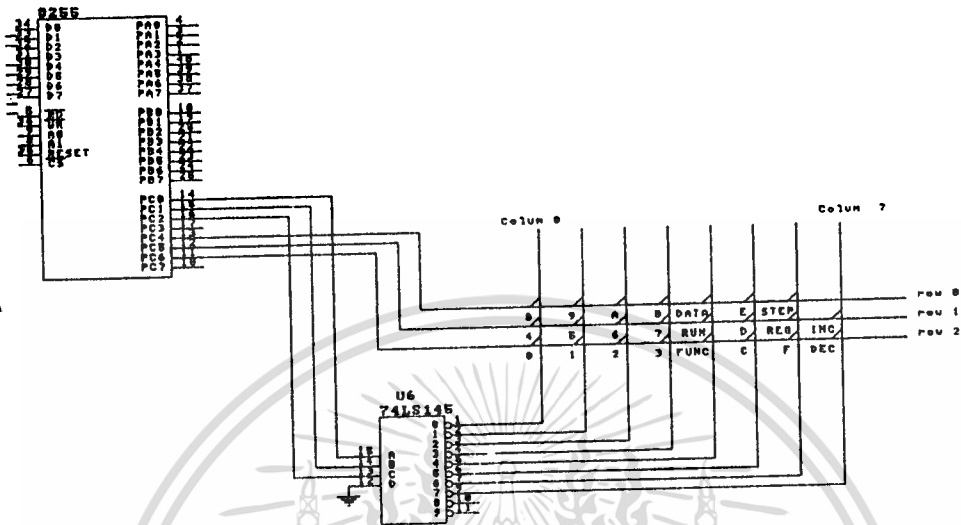
ตารางที่ 3.1 ตำแหน่งหน่วยความจำที่กำหนดสำหรับสร้างสัญญาณ Chip Select ของ 8255 หมายเลข 1 และ 8255 หมายเลข 2



รูปที่ 3.18 วงจรที่ใช้เลือกไอซี 8255 หมายเลข 1 และหมายเลข 2

จึงต่อเข้ากับ A7 ส่วนแอดเดรสควบคุมซึ่งที่เหลือคือ A11, A13, A14 และ A15 ให้ลอคจิกที่เป็น "0" ตลอด 2 ช่วงของหมายเลขพอร์ตดังกล่าวมาผ่านวงจรรอท์เกต ซึ่งประกอบด้วยไดโอด 1N4148 จำนวน 4 ตัว และตัวต้านทาน 1 กิโลโอห์ม 1 ตัว ดังรูปที่ 3.18 วงจรสมบูรณ์ของส่วนถอดรหัสพอร์ตภายนอกที่พิจารณาจากตารางที่ 3.1 สามารถเขียนได้ดังรูปที่ 3.18

3.3.3 ส่วนเคียบอร์ด



รูปที่ 3.19 การต่อวงจรส่วนเคียบอร์ด

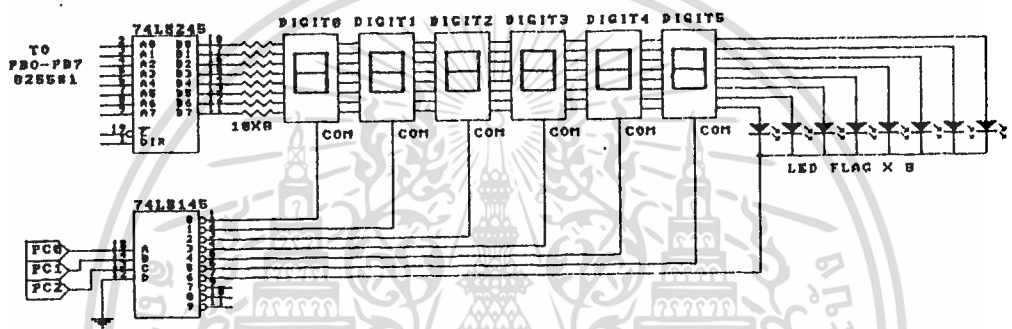
จากรูปที่ 3.19 เป็นการออกแบบส่วนเคียบอร์ด การทำงานจะให้พอร์ต C ค่า คือ ที่พอร์ต C ชุด 0-3 เป็นพอร์ตเอาต์พุต ส่วนพอร์ต C ชุด 4-7 ซึ่งเป็นพอร์ต C สูงบนนี้เป็นพอร์ตอินพุต การทำงานจะใช้งานร่วมกับการควบคุมทางด้านซอฟต์แวร์ ซึ่งสามารถดูได้จาก Flow chart ในภาคผนวก ในหัวข้อนี้จะกล่าวเน้นไปในด้านฮาร์ดแวร์

การทำงานของวงจรจะใช้วิธีการสแกน เพื่อหาว่าสวิตช์ใดที่ถูกกด โดยอาจเริ่มที่คอลัมน์ 0 โดยส่งค่าลอจิก "0" มาทางพอร์ตเอาต์พุตให้ U6 ซึ่งเป็นตัวสแกนคอลัมน์ เมื่อได้รับลอจิก "0" ทางอินพุตจึงทำให้ U6 ให้เอาต์พุต YO เป็นลอจิก "0" ส่วนคอลัมน์อื่น ๆ เป็นลอจิก "1" แล้วทำการตรวจสอบทีละแถว โดยจะเริ่มจากแถว 0 ตรวจสอบว่ามีระดับลอจิกเป็น "0" หรือไม่ ถ้ามีระดับลอจิกเป็น "0" แสดงว่าสวิตช์ตำแหน่งนั้นถูกกด ถ้าไม่มีก็เลื่อนไปแถวถัดไปแล้วทำการเลื่อนและตรวจสอบทั้งด้านคอลัมน์และแถวต่อไปเรื่อย ๆ จนกว่าจะหมดหรือจนกระทั่งพบสวิตช์ที่ถูกกด และเมื่อพบว่าสวิตช์ใดถูกกดแล้ว จะต้องทราบว่าสวิตช์นี้คือสวิตช์ตำแหน่งใด เพื่อจะได้ตีความหมายหรือสร้างรหัสขึ้นตามสวิตช์ตำแหน่งนั้น ๆ เพื่อนำรหัสนั้นไปใช้งานต่อไป

3.3.4 ส่วนแสดงผลบนแอลอีดี 7 เซกเมนต์และแอลอีดีแฟล็ก

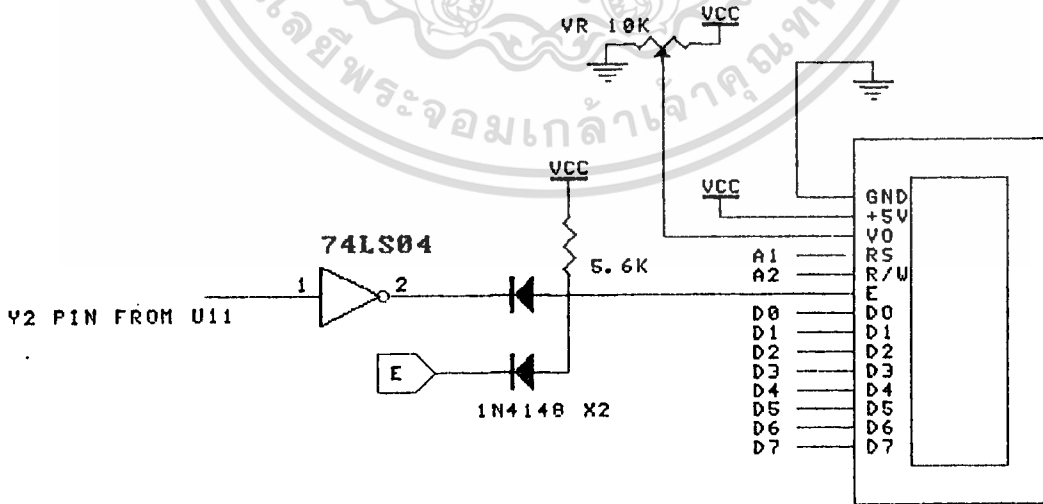
การออกแบบวงจรใช้วิธีการมัลติเพล็กซ์ แทนที่จะใช้จำนวนพอร์ตเท่ากับจำนวนของตัวแสดงผล คือ จะใช้พอร์ตหนึ่งต่อเข้ากับเซกเมนต์ของตัวแสดงผลทุกตัว และอีกพอร์ตหนึ่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อเข้ากับ U6 ซึ่งทำหน้าที่สแกนลอจิก "0" ของ 7 เซกเมนต์และแอลอีดีแฟล็ก การทำงาน เช่น เมื่อเราต้องการให้ 7 เซกเมนต์หลัก 0 ติด สามารถทำได้โดยส่งข้อมูลออกพอร์ต B และส่ง "000" ออกทางพอร์ต C ล่าง 3 บิต คือ PC0, PC1 และ PC2 เมื่อ U6 ได้รับ ลอจิก "0" ที่ขา A, B และ C ของ BCD ทางอินพุต จึงทำให้ส่งลอจิก "0" ออกให้ YO หลัก 0 ของ 7 เซกเมนต์จึงติด ส่วนเซกเมนต์หลักอื่นได้รับลอจิก "1" จึงดับ และเรา ทำการส่งข้อมูลให้เซกเมนต์ในหลักต่อ ๆ ไป พร้อมกับการส่งลอจิก "0" ให้เซกเมนต์ใน หลักนั้นก็ติดเรียงกันไป เมื่อครบรอบแล้วก็กลับมาเริ่มต้นที่หลัก 0 ใหม่ แต่ทำด้วยความ รวดเร็วมากจึงเห็นเซกเมนต์ทุกหลักติดพร้อมกัน ซึ่งมีวงจรใช้งานดังรูปที่ 3.20



รูปที่ 3.20 วงจรแสดงผล 7 เซกเมนต์และแอลอีดีแฟล็ก

3.3.5 ส่วนการแสดงผลแอลอีดี



รูปที่ 3.21 วงจรเชื่อมต่อกับแอลอีดีขนาด 16 ตัวอักษร 1 แดว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.21 เป็นการออกแบบการแสดงผลแอลซีดี แอลซีดีที่ใช้จะเป็นโมดูลแอลซีดี ขนาด 16 แถว 1 ตัวอักษร ซึ่งภายในโมดูลประกอบด้วย

1. DOT MATRIX LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการปิดและเปิดตัวเองกับแสงก็คือ ส่วนที่เป็นกระจกบรรจุผลึก

2. DRIVER เป็นตัวรับสัญญาณจากตัวควบคุมมาขับผลึกแอลซีดีอีกตัวหนึ่ง โดยมีเบอร์ที่นิยมใช้ในแอลซีดีโมดูล เช่น HD44100H และ MSM5259

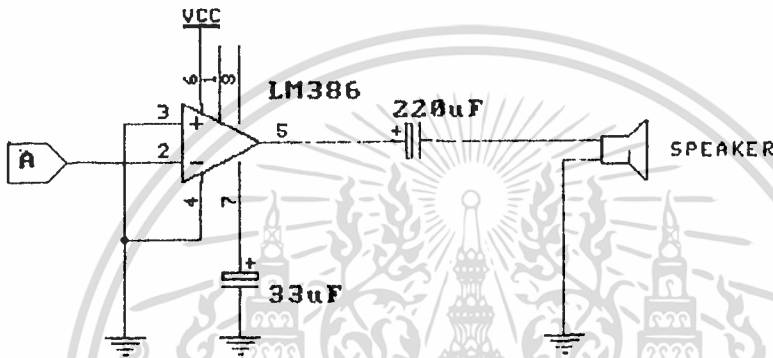
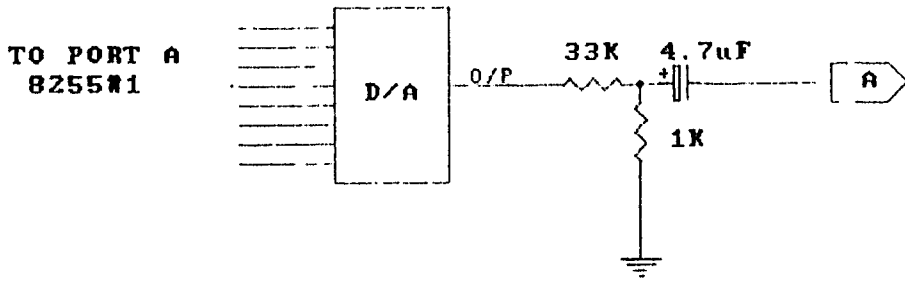
3. CONTROLLER เป็นตัวที่รับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุมแอลซีดีโมดูลให้ทำงานแสดงผลต่าง ๆ เช่น การลบจอภาพ การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ไอซีที่นิยมใช้กัน คือ HD44780 ซึ่งจะใช้แบบ CHARACTER แอลซีดี MODULE เป็นส่วนใหญ่

เพราะฉะนั้น ในการออกแบบเราเพียงแต่นำข้อมูลมาเชื่อมต่อกับ 68HC11 เพื่อส่งสัญญาณควบคุมและข้อมูลในการแสดงผลและการอ่านข้อมูลต่าง ๆ กับโมดูลเท่านั้น ในการเชื่อมต่อขา DO-D7 ซึ่งเป็นขาบัสข้อมูลของแอลซีดีสามารถที่จะต่อเข้ากับ PC0-PC7 ของ 68HC11 โดยตรง ขา RS และ R/W จะต่อเข้ากับแอดเดรส A1 และ A2 โดยขา E จะใช้เอาท์พุท Y2 ของ 74HC138 เป็นตัวเลือก โดยมีแอดเดรสของพอร์ตที่ใช้งานอยู่ในช่วง 1400H-15FFH การปรับระดับความสว่างของจอแอลซีดีอาศัยการปรับไบอัสให้ขา VO โดยต่อตัวต้านทานปรับค่าได้ 10 กิโลโอห์ม ปรับระดับการไบอัส

3.3.6 ส่วนการแสดงผลทางลำโพง

การออกแบบส่วนการแสดงผลทางลำโพง ในลักษณะของเสียงจะให้ที่พอร์ต A ของ 8255 หมายเลข 1 หมายเลขพอร์ต 81700 ใช้เป็นตัวสร้างสัญญาณความถี่เสียงผ่านวงจรขยายที่ประกอบด้วยขอปแอมป์ LM386 เป็นตัวขยายสัญญาณ การกำเนิดเสียงสามารถทำได้ โดยการส่งข้อมูลให้พอร์ต A ในรูปของสัญญาณดิจิทัล 8 บิต สัญญาณดิจิทัลจะถูกเป็นสัญญาณอนาล็อก โดย D/A CONVERTER ซึ่งรูปแบบของ D/A CONVERTER จะอาศัยตัวต้านทานทั้งหมด เพื่อเป็นการประหยัดและสะดวกให้ความสามารถในการแปลงสัญญาณเป็นที่น่าพอใจ การกำเนิดเสียงจะควบคุมโดยโปรแกรม ซึ่งจะประกอบด้วย

- ความดังของเสียง
- ระดับความดังของเสียง
- ช่วงเวลาในการกำเนิดเสียง



รูปที่ 3.22 วงจรการแสดงผลทางลำโพง

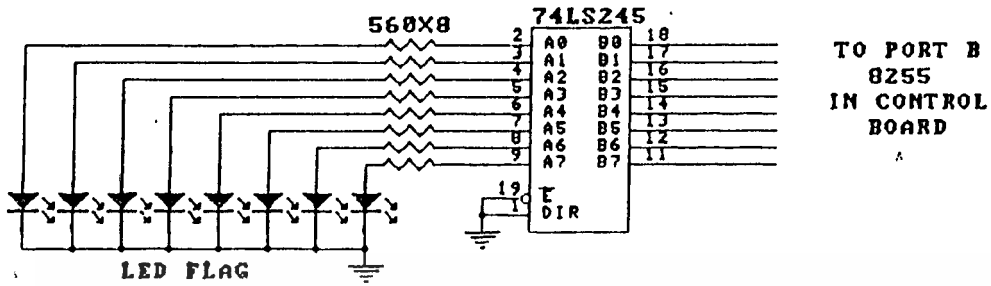
3.3.7 ส่วนของการอินเตอร์เฟสกับอุปกรณ์ภายนอก

ในส่วนนี้จะประกอบด้วยวงจร 3 ส่วน คือ

1. วงจรแสดงผลทางแอลอีดี
2. วงจรเชื่อมต่อเครื่องพิมพ์
3. วงจรวัดแรงดันไฟตรง 0-5 โวลต์ 8 ที่ลง

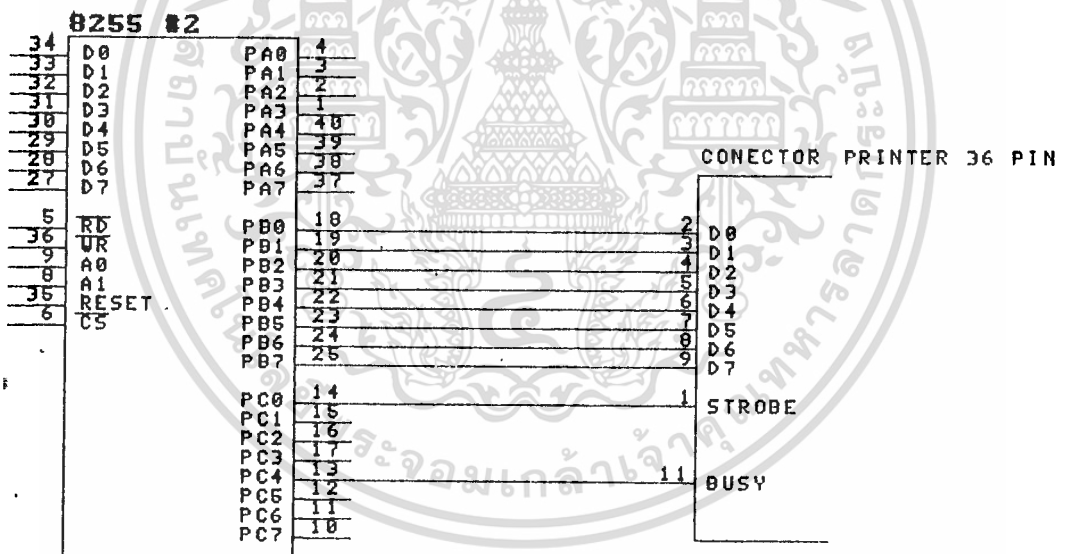
3.3.7.1 วงจรแสดงผลทางแอลอีดี

การออกแบบวงจรจะให้ไอซี 74LS245 เป็นตัวขับแอลอีดี 8 ดวง โดยมีตัวต้านทาน 580 โอห์ม 8 ตัว เป็นตัวจำกัดกระแส เมื่อเราต้องการจะให้แอลอีดีติด ก็ส่งค่าที่ต้องการออกไปยังพอร์ต B ของ 8255 ในชุดควบคุมซึ่งมีหมายเลขพอร์ต 4 อยู่ที่ตำแหน่ง \$1201\$ วงจรแสดงผลทางแอลอีดีมีวงจрдังรูปที่ 3.23



3.23 วงจรแสดงผลทางแอลอีดี

3.3.7.2 วงจรเชื่อมต่อเครื่องพิมพ์

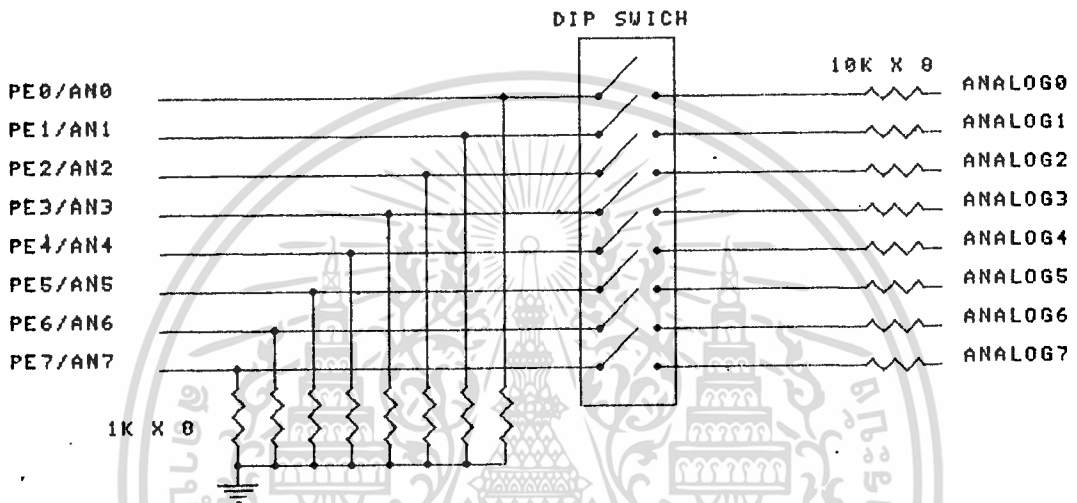


รูปที่ 3.24 วงจรเชื่อมต่อเครื่องพิมพ์

การออกแบบวงจรใช้ 8255 หมายเลข 2 เป็นตัวเชื่อมต่อกับเซนทรอนิกส์ 36 ทาใช้พอร์ต B ของ 8255 เป็นพอร์ตข้อมูล พอร์ต C บนจะกำหนดให้เป็นพอร์ตเอาท์พุท เราใช้ PC4 เป็นตัวส่งสัญญาณ STROBE ในการเปิดและปิดข้อมูลของเครื่องพิมพ์ และที่ในส่วนพอร์ต C ล่าง กำหนดให้เป็นพอร์ตอินพุท ใช้ในการอ่านสัญญาณ BUSY เมื่อเครื่องพิมพ์อยู่ในสภาวะพร้อม

3.3.7.3 วงจรวัดแรงดันไฟตรง 0-5 โวลต์ 8 ช่อง

การออกแบบภาคเชื่อมต่อในการนำวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลของ 68HC11 ที่ไปใช้วัดระดับแรงดันสัญญาณอนาล็อกที่เป็นไฟตรง 0-5 โวลต์ จะใช้ตัวต้านทาน 1 กิโลโอห์ม ต่อลงกราวด์เพื่อแก้สภาวะอิมพีแดนซ์สูงของอินพุต ในขณะที่ไม่มีสัญญาณอนาล็อกเข้ามา ดิพสวิตช์จะใช้เปิดหรือปิดในแต่ละช่องที่จะวัดสัญญาณอนาล็อก การต่อตัวต้านทานค่า 10 กิโลโอห์ม จะจำกัดกระแสเพื่อป้องกันความเสียหายของพอร์ตแปลงสัญญาณ 68HC11

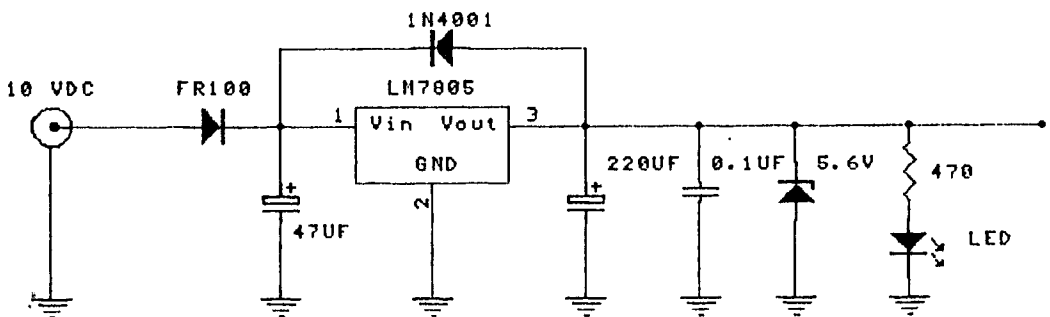


รูปที่ 3.25 วงจรวัดแรงดันไฟตรง 0-5 โวลต์ 8 ช่อง

3.3.8 ส่วนของแหล่งจ่ายไฟ

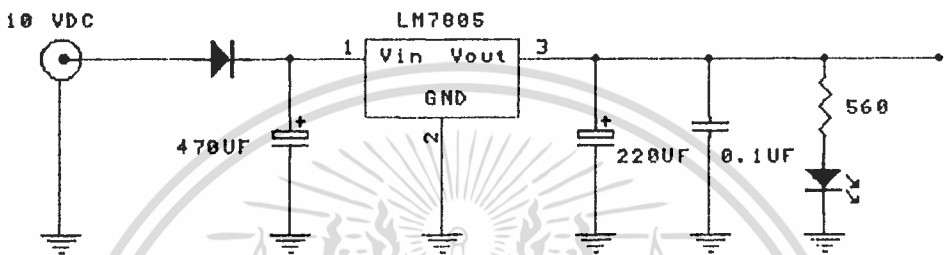
ส่วนของแหล่งจ่ายไฟประกอบด้วยวงจร 3 ส่วน คือ

1. วงจรรักษาระดับแรงดันไฟตรงในบอร์ดควบคุม
2. วงจรรักษาระดับแรงดันไฟตรงในบอร์ดแสดงผล
3. วงจรแปลงแรงดันไฟสลับเป็นไฟตรง



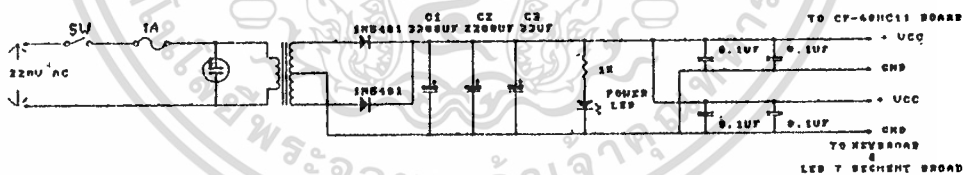
รูปที่ 3.26 วงจรรักษาระดับแรงดันไฟตรงในบอร์ดควบคุม

การออกแบบวงจรจะใช้ LM7805 เป็นตัวเรกกูเลเตอร์ไฟตรง 5 โวลต์ จะใช้ไดโอด FR100 เป็นอุปกรณ์ป้องกันการต่อแหล่งจ่ายไฟกลับขั้ว ตัวไดโอด 1N4001 ใช้บายพาสการคายประจุของตัวเก็บประจุไม่ให้คายประจุผ่านไอซีเรกกูเลต สำหรับซีเนอร์ไดโอดมีขนาด 5.6 โวลต์ ใช้ป้องกันการผิดพลาดของตัวเรกกูเลต จะจำกัดระดับแรงดันไฟเอาต์พุตไม่ให้เกิน 5.6 โวลต์ ตัวต้านทาน 470 โอห์ม นำมาใช้ลดแรงดันและจำกัดกระแสที่ไบอัสให้แก่แอลอีดีในสภาวะที่เปิดเครื่องอยู่



รูปที่ 3.27 วงจรรักษาระดับแรงดันไฟตรงในบอร์ดแสดงผล

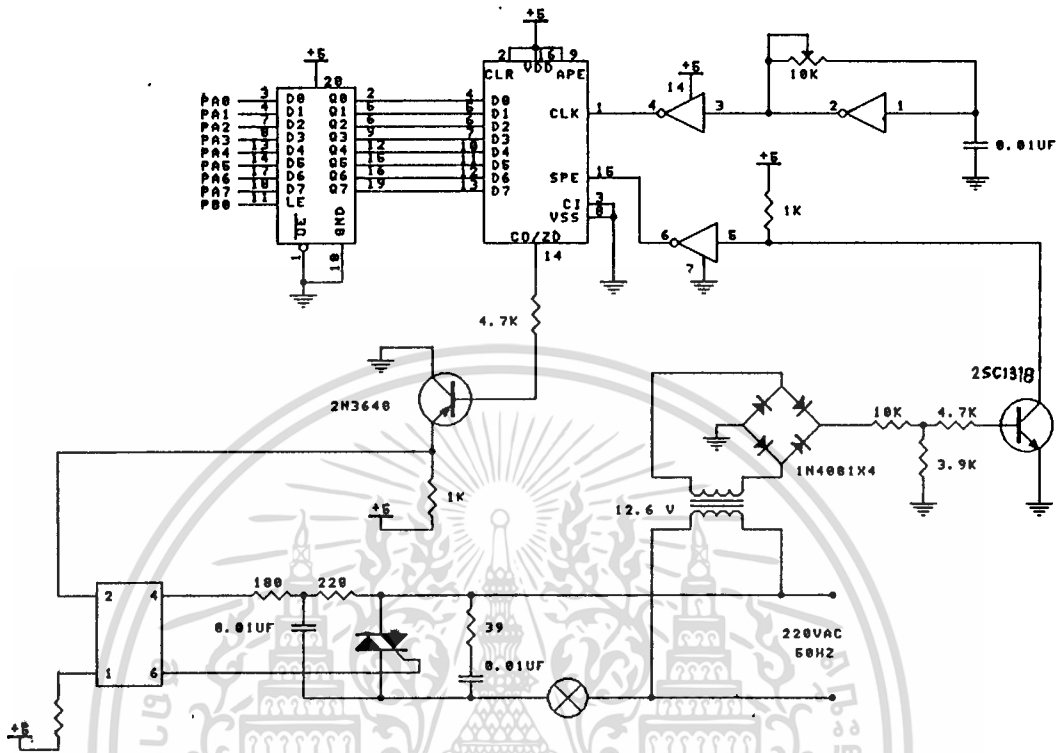
การออกแบบวงจรจะใช้ LM7805 เป็นตัวเรกกูเลเตอร์ไฟตรง 5 โวลต์ไปเลี้ยงวงจร โดยมีตัวเก็บประจุค่า 470uF และ 220uF ใช้เป็นตัวฟิลเตอร์แรงดันไฟตรงให้เรียบยิ่งขึ้น ส่วนตัวเก็บประจุ 0.1uF เป็นตัวฟิลเตอร์สัญญาณรบกวน



รูปที่ 3.28 วงจรแปลงไฟสลับเป็นไฟตรง

การออกแบบวงจรนี้จะใช้ไดโอด 1N5401 2 ตัว เป็นวงจรเรกติไฟร์แบบเต็มคลื่น ไฟตรงที่ได้จากการเรกติไฟร์จะถูกฟิลเตอร์และกรองสัญญาณรบกวน โดยตัวเก็บประจุได้รับแรงดันไฟตรงเอาต์พุตที่ฟิลเตอร์แล้วประมาณ 12 โวลต์

3.3.9 ส่วนของบอร์ดทดลอง



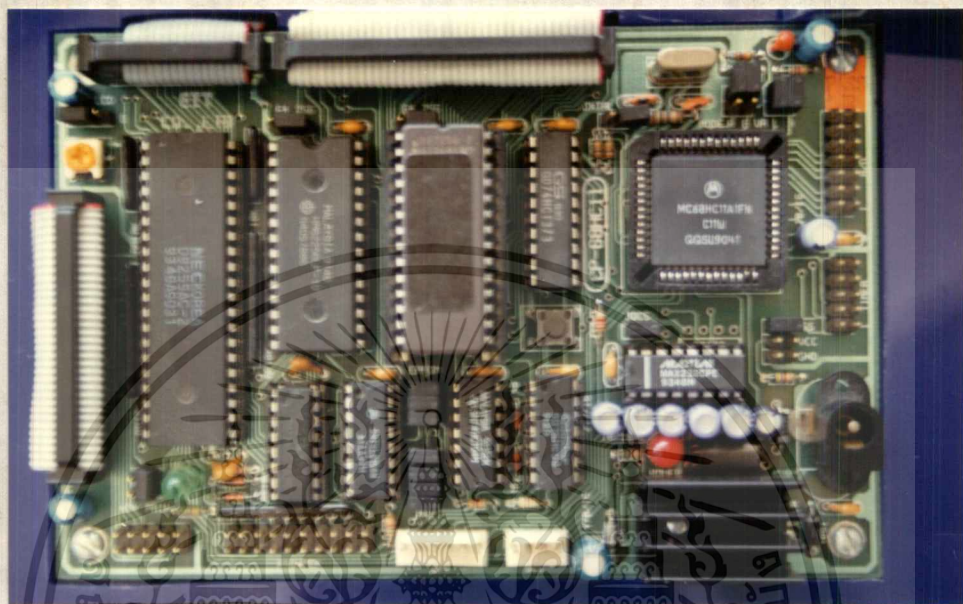
รูปที่ 3.29 วงจรหรีไฟควบคุมด้วยไมโครคอนโทรลเลอร์

บอร์ดทดลองเป็นส่วนที่เว้นไว้สำหรับการขยายระบบ บนมีลติบอร์ดจะมีคอนเนคเตอร์ขนาด 40 ขา ซึ่งต่อมาจาก 8255 หมายเลข 2 สำหรับนำไปทดลองการเชื่อมต่อกับวงจรจากภายนอกเพื่อศึกษาการควบคุมระบบต่าง ๆ โดยใช้ไมโครคอนโทรลเลอร์ ในส่วนของบอร์ดทดลองนี้ประกอบขึ้นมาเพื่อเป็นตัวอย่างสำหรับการทดลองควบคุมระบบภายนอก วงจรที่ประกอบเป็นวงจรหรีไฟควบคุมด้วยไมโครคอนโทรลเลอร์ ลักษณะของบอร์ดทดลองจะเป็นแผ่นวงจรพิมพ์ชนิดเพลตทรูโฮลด์มาตรฐาน ดังนั้นถ้าหากเราจะใช้บอร์ดนี้ไปประกอบวงจรควบคุมอื่น ๆ ก็สามารทำได้โดยการเปลี่ยนแผ่นวงจรพิมพ์ที่มีขนาดเท่ากับบอร์ดทดลองเดิมแล้วประกอบวงจรทดลองมาใส่แทนที่บอร์ดเดิม

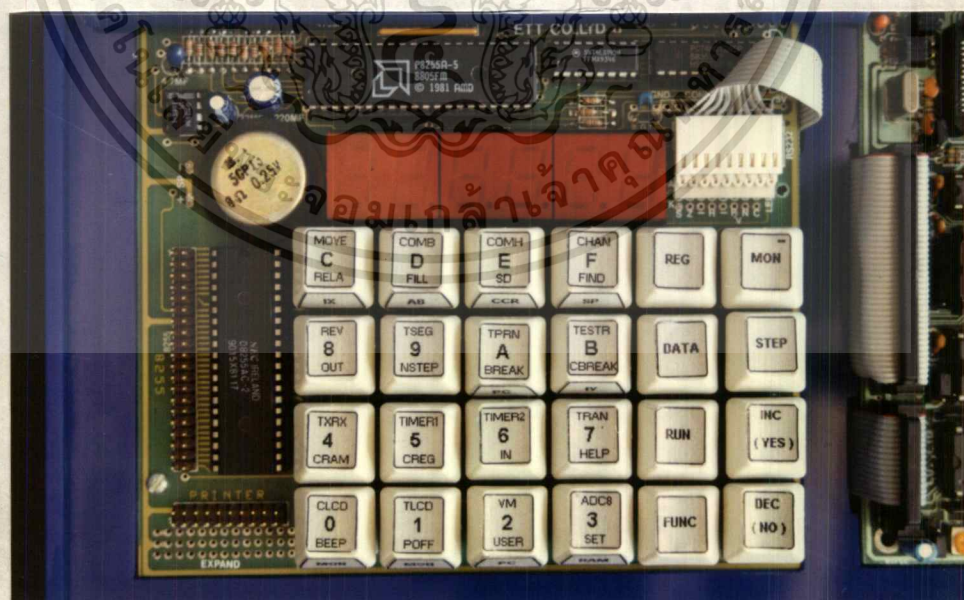
การออกแบบวงจรหรีไฟควบคุมด้วยไมโครคอนโทรลเลอร์นี้ ความสว่างของหลอดไฟจะปรับได้โดยการเปลี่ยนแปลงข้อมูลที่ส่งออกพอร์ต A ของ 8255 แลร์ 2 แต่ก่อนที่จะส่งข้อมูลออกพอร์ต A จะต้องอีน่าเบิลการส่งข้อมูลโดยการส่งข้อมูลออกพอร์ต B ให้พอร์ต B0 เป็นลอจิก "1" การแยกภาคไฟต่ำและไฟสูงจะให้ข้อป้ได้ค้เปเปอร์เพื่อป้องกันการลัดวงจรจากภาคไฟสูงมายังภาคไฟต่ำไม่ให้เกิดการช้ารุดของอุปกรณ์ รายละเอียดของส่วนนี้จะได้เพิ่มเติมไว้ในส่วนใบงาน

3.4 การสร้างมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11

การประกอบชุดต่าง ๆ ตลอดจนลักษณะโดยรวมของมัลติบอร์ดมีรูปแบบดังรูปต่อไปนี้

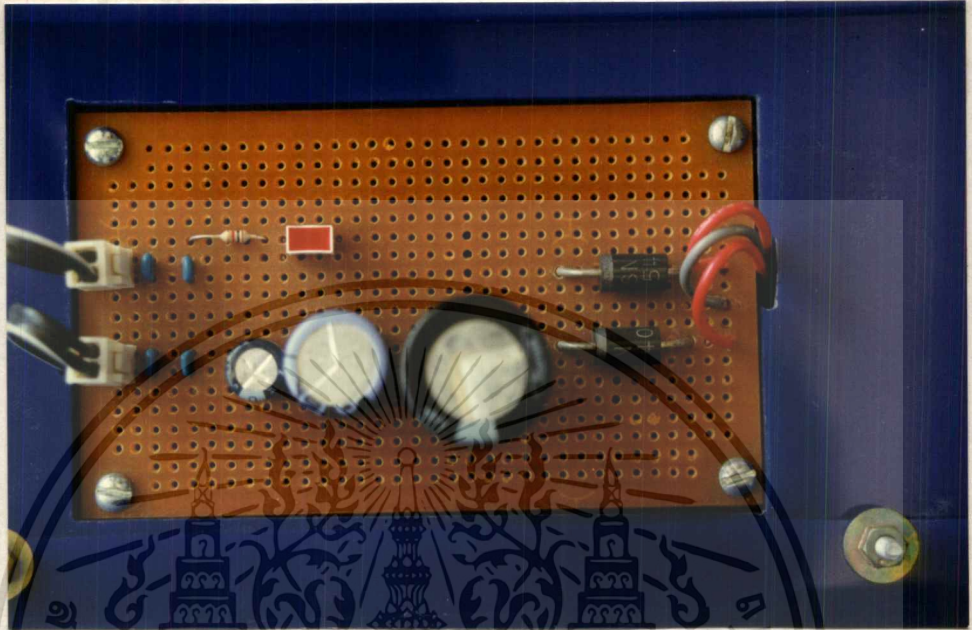


รูปที่ 3.30 ชุดควบคุม CP-68HC11

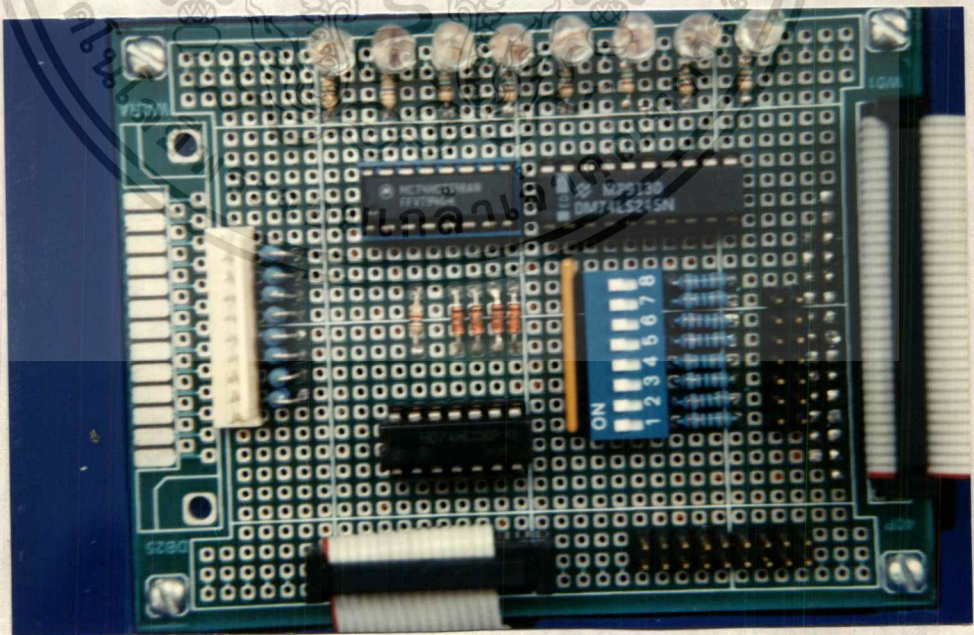


รูปที่ 3.31 ชุดแสดงผลเอาต์พุตและคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

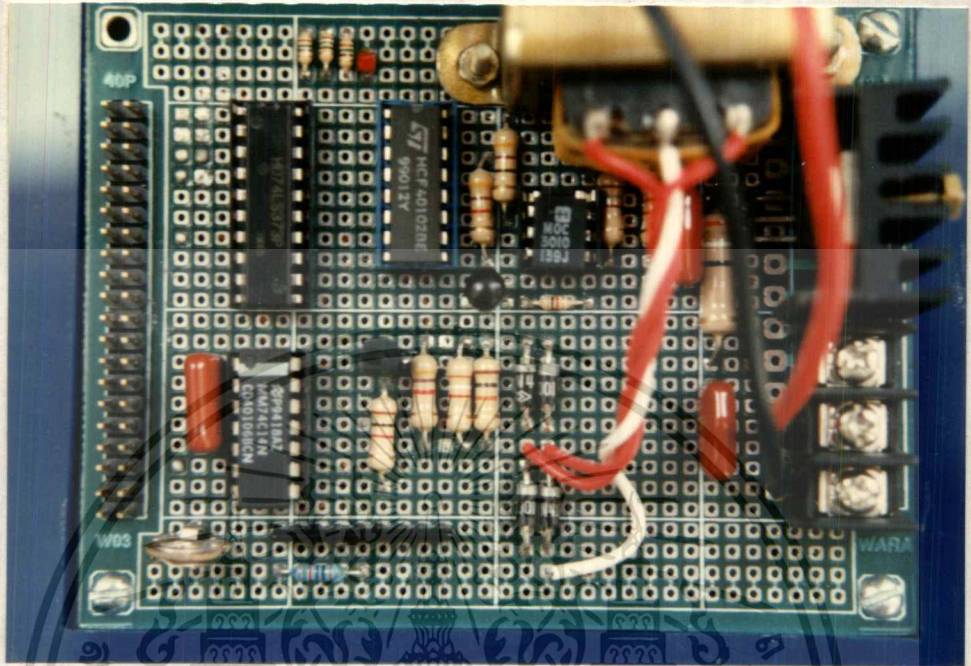


รูปที่ 3.32 ชุดแหล่งจ่ายไฟ

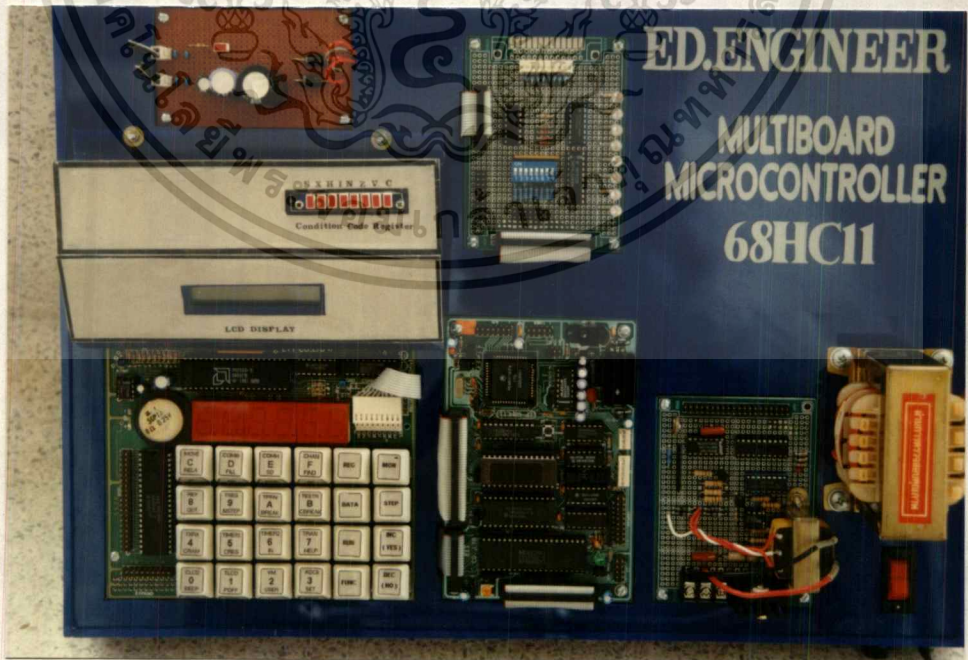


รูปที่ 3.33 ชุดอินเทอร์เฟซและถอดรหัสพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.34 ชุดบอร์ดทดลอง



รูปที่ 3.35 ลักษณะโดยสมบูรณ์ของมัลติบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 จะสามารถทำงานได้โดยการสั่งงานบนคีย์บอร์ดของเครื่องคอมพิวเตอร์ หรือบนตัวมัลติบอร์ดเอง เพราะฉะนั้นเราจึงแบ่งส่วนของการทดลองเป็น 2 ส่วนใหญ่ ๆ คือ

1. การทดลองโปรแกรมทดสอบบนมัลติบอร์ด
2. การทดลองโปรแกรมบนเครื่องคอมพิวเตอร์

โดยจะให้โปรแกรมทดสอบเดียวกัน แล้วนำผลการทดลองทั้งสองมาเปรียบเทียบกันว่าให้ผลการทดลองเหมือนกันหรือไม่

4.1 การทดลองโปรแกรมทดสอบบนมัลติบอร์ด

ลำดับที่การทดลอง

1. เปิดสวิตช์จ่ายไฟให้มัลติบอร์ด
2. จะปรากฏข้อความ ED.ID.1.0 วิ่งจากขอบซ้ายของจอแสดงผลมาหยุดกลางจอ หลังจากนั้นจะวิ่งหายไปทางขวาของจอแสดงผล พร้อมกับจะมีเสียงไตเติลซาวด์ 2 จังหวะดังขึ้น แล้วจอแอลซีดีจะขึ้นข้อความ " MULTIBOARD V1.0 "
3. รอจนปรากฏข้อความ S-012 บนจอแสดงผลให้กดคีย์เลข 1 จะขึ้นเครื่องหมาย Prompt วรรคเดียวบนหลัก 0 ของจอแสดงผล จอแอลซีดีจะขึ้น "AUTO RUN ON BOARD " แสดงการทำงานบนตัวบอร์ด
4. ตั้งแอด्रेसของบอร์ดที่ \$2300 แล้วป้อนโปรแกรมทดสอบที่ 4.1 อยู่ในรูปของภาษาเครื่อง เมื่อป้อนโปรแกรมเสร็จแล้วให้กดคีย์ MON จะมีเครื่องหมาย Prompt แสดงขึ้นมาที่จอแสดงผล
5. ตั้งแอด्रेसของโปรแกรมที่จะประมวลผลที่ \$2300 และใช้คีย์ STEP ทำการประมวลผลโปรแกรมทีละคำสั่ง เพื่อดูการทำงานในแต่ละส่วนและบันทึกค่ารีจิสเตอร์ลงในตารางที่ 4.1

โปรแกรมทดสอบที่ 4.1

Freeware assembler ASxx.EXE Ver 1.03.

```

0001          *****
0002          *      PROGRAM TEST MULTIBOARD      *
0003          *              ON BOARD              *
0004          *      [ LOAD REGISTER ]              *
0005          *****
0006 2300          ORG $2300
0007
0008 ce2b          RECALL EQU $CE2B ; System Call ADDR
0009
0010 2300 86 01          LDAA #$01 ; Load AccA
0011 2302 c6 23          LDAB #$23 ; Load AccB
0012 2304 ce 45 67          LDX #$4567 ; Load IX
0013 2307 18 ce 89 ab          LDY #$89AB ; Load IY
0014 230b 8e 22 00          LDS #$2200 ; Load SP
0015 230e 86 00          LDAA #$00 ; Goto Monitor
0016 2310 bd ce 2b          JSR RECALL
0017          END
Number of errors 0

```

Step	Assembly	Condition of Register						
		PC	IY	IX	A	B	CCR	SP
1	LDAA ##01	2302	0000	0000	01	00	90	201F
2	LDAB ##23	2304	0000	0000	01	23	90	201F
3	LDX ##4567	2307	0000	4567	01	23	90	201F
4	LDY ##89AB	230B	89AB	4567	01	23	98	201F
5	LDS ##2200	230E	89AB	4567	01	23	90	2200

ตารางที่ 4.1 ผลการทดลองโปรแกรมทดสอบที่ 4.1

6. ตั้งแอดเดรสของบอร์ดที่แอดเดรส \$2300 แล้วป้อนโปรแกรมทดสอบที่ 4.2 ในรูปของภาษาเครื่อง เมื่อป้อนโปรแกรมเสร็จแล้วให้กดคีย์ MON จะมีเครื่องหมาย Prompt ที่หน้าจอแสดงผล

7. ตั้งแอดเดรสของโปรแกรมที่จะประมวลผลที่ \$2300 แล้วกดคีย์ RUN บันทึกผลที่ได้ลงในตารางที่ 4.2

โปรแกรมทดสอบที่ 4.2

Freeware assembler ASxx.EXE Ver 1.03.

```

0001          *****
0002          *      PROGRAM TEST MULTIBOARD      *
0003          *                ON BOARD                *
0004          *      [ LOOP PROGRAM ]      *
0005          *****
0006 2300          ORG $2300
0007
0008 1701          SEGMENT EQU $1701 ; Data Display Port
0009 1702          DIGIT EQU $1702 ; Scan Digit Port
0010 1703          CTRL EQU $1703 ; Control Port

```

```

0011
0012 2300 86 88          LDAA  #$88  ; Control Word
0013 2302 b7 17 03      STAA  CTRL
0014 2305 86 66          LOOP  LDAA  #$66  ; Data Display
0015 2307 c6 00          LDAB  #$00  ; Start Counter
0016 2309 b7 17 01      LOOP1 STAA  SEGM ; Display Data
0017 230c f7 17 02      STAB  DIGIT ; Scan Digit
0018 230f 5c             INCB           ; Counter+1
0019 2310 c1 07          CMPB  #$07  ; Check Counter
0020 2312 26 f5          BNE  LOOP1 ; Next Digit Scan
0021 2314 20 ef          BRA  LOOP  ; Goto Loop
0022                      END
Number of errors 0
    
```

โปรแกรมทดสอบที่	การแสดงผล
4.2	จะส่งค่า \$66 ออกที่ส่วนแสดงผลหลักและ แอลอีดีแฟล็กบนจอแสดงผลจะปรากฏเป็นเลข 4 และติดค้างอยู่ตลอด เพราะโปรแกรมมีการวนลูป

ตารางที่ 4.2 ผลการทดลองโปรแกรมทดสอบที่ 4.2

8. ตั้งแอดเดรสของพอร์ตที่ \$2300 แล้วทำการป้อนโปรแกรมทดสอบที่ 4.3 ในรูปของภาษาเครื่อง เมื่อป้อนโปรแกรมเสร็จแล้วให้กดคีย์ MON จะมีเครื่องหมาย Prompt ปรากฏขึ้นบนจอแสดงผล

9. ตั้งแอดเดรสของโปรแกรมที่จะประมวลผลที่ \$2300 แล้วให้คีย์ RUN ทดสอบการทำงาน ของโปรแกรม บันทึกผลที่ได้ลงในตารางที่ 4.3

โปรแกรมทดสอบที่ 4.3

Freeware assembler ASxx.EXE Ver 1.03.

```

0001          *****
0002          *      PROGRAM TEST MULTIBOARD      *
0003          *              ON BOARD              *
0004          *      [ SYSTEM CALL ]              *
0005          *****
0006 2300          ORG $2300
0007
0008 ce2b          RECALL EQU $CE2b      ; System Call ADDR
0009 ff00          CALL EQU $FF00      ; System Call ADDR
0010
0011 2300 86 0c          LDAB #$0C      ; Call Subroutine
0012 2302 bd ff 00          JSR CALL      ; Clear LCD
0013 2305 86 0b          LDAA #$0B      ; Call Subroutine
0014 2307 bd ce 2b          JSR RECALL      ; READY
0015 230a 86 00          LDAA #$00      ; Goto Monitor
0016 230c bd ce 18          JSR RECALL
0017          END
Number of errors 0

```

โปรแกรมย่อยที่	โปรแกรม	การแสดงผล
1	LDAB ##OC JSR CALL	เคลียร์หน้าจอแอลซีดี
2	LDA A #\$OB JSR RECALL	จะแสดง rEAdY บนจอแสดงผล กดคีย์ INC จะขึ้นเครื่องหมาย Prompt ของบอร์ด
3	LDA A ##00 JSR RECALL	กลับสู่เมนูเอนเตอร์ของมัลติบอร์ด

ตารางที่ 4.3 ผลการทดลองโปรแกรมทดสอบที่ 4.3

สรุปผลการทดลอง

จากการทดลองในตารางที่ 4.1 นี้เป็นการทดลองประมวลผลโปรแกรมแบบ STEP เพื่อการเปลี่ยนแปลงในการไปคำสั่งไหลค่าให้รีจิสเตอร์ของ 68HC11 ผลการทดลองที่ได้นี้ให้ค่าถูกต้องตามเงื่อนไขการทำงานของคำสั่งไหล จึงสรุปว่ามัลติบอร์ดสามารถที่จะทำการประมวลผลแบบ STEP ได้อย่างสมบูรณ์

จากผลการทดลองในตารางที่ 4.2 แสดงว่ามัลติบอร์ดสามารถประมวลผลโปรแกรมที่เขียนโดยใช้คีย์ RUN ได้ถูกต้องตามโปรแกรมที่เขียนขึ้น จึงสามารถสรุปได้ว่ามัลติบอร์ดสามารถใช้คำสั่งประมวลผลให้ทำงานได้อย่างถูกต้อง

จากผลการทดลองในตารางที่ 4.3 เห็นว่ามัลติบอร์ดสามารถเรียกใช้โปรแกรมย่อยทั้งของตัวเองและในส่วนของ DEBUGGER ได้อย่างถูกต้องและสามารถกลับเข้าสู่เมนูเอนเตอร์ได้โดยให้ SYSTEM CALL ได้ จึงสรุปได้ว่าระบบ SYSTEM CALL ทำงานได้อย่างถูกต้อง

4.2 การทดลองโปรแกรมทดสอบบนเครื่องคอมพิวเตอร์

ลำดับขั้นการปฏิบัติการทดลอง

1. เปิดสวิตซ์แหล่งจ่ายไฟให้มัลติบอร์ด

2. จะปรากฏข้อความ Ed.Id.1.0 ซึ่งมาจากขอบข่ายของจอแสดงผลจะผลมาหยุดที่

กลางจอ แล้ววิ่งหายไปทางขวาของจอแสดงผล หลังจากนั้นจะมีข้อความเสียงไตเติลชาวด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 จังหวะ และบนจอจะแสดงข้อความ "MULTIBOARD V1.0"

3. รอจนกว่าจะปรากฏข้อความ S-012 บนจอแสดงผล ให้กดคีย์หมายเลข 2 จะปรากฏข้อความ "AUTORUN ON PC."

4. หน้าจอแสดงผลดับแล้วปรากฏข้อความดังรูปที่ 4.1 เป็นการเข้าสู่ DEBUGGER ก่อนที่จะทำการทดลองโปรแกรมทดสอบนี้ ต้องมีการประมวลผลโปรแกรม PROCOMM PLUS บนเครื่องคอมพิวเตอร์ก่อน

A013	86 00	LDA #00
A015	B7 10 24	STA \$1024
A018	86 00	LDA #00
A01A	B7 10 35	STA \$1035
A01D	8E 20 82	LDS #2082
A020	BD A3 56	JSR \$A356
A023	BD A3 35	JSR \$A335
A026	CE 20 50	LDX #2050
A029	FF 20 8C	STX \$208C
A02C	CE B6 00	LDX #B600
A02F	FF 20 90	STX \$2090
A032	CE B7 FF	LDX #B7FF
A035	FF 20 92	STX \$2092
A038	CE 00 00	LDX #0000
A03B	FF 20 A0	STX \$20A0
A03E	B6 10 20	LDA \$1020
A041	8A 03	ORA #03
A043	B7 10 20	STA \$1020
A046	86 D0	LDA #D0
A048	B7 20 8B	STA \$208B
A04B	CC 3F 0D	LDD #3F0D
A04E	FD 21 01	STD \$2101

รูปที่ 4.1 รูปของจอแสดงผลเมื่อประมวลผลโปรแกรมบนเครื่องคอมพิวเตอร์

5. ใ้ค้ำสั่ง A 2300 แล้วให้กด Enter เพื่อตั้งแอดเดรสเริ่มต้นโปรแกรมทดสอบที่ \$2300 แล้วเริ่มป้อนภาษาแอสเซมบลีดังโปรแกรมทดสอบที่ 4.4 ตามรูปแบบของวิธีการใน DEBUGGER ด้วย เมื่อจบการเขียนโปรแกรมแล้วกด Enter อีกครั้งจะขึ้นเครื่องหมาย Prompt บนหน้าจอแสดงผล

6. ตั้งค่าโปรแกรมเคาน์เตอร์ให้ที่แอดเดรส \$2300 และใช้ค้ำสั่ง T ประมวลผลโปรแกรมทีละค้ำสั่ง เพื่อดูการเปลี่ยนแปลงของค่ารีจิสเตอร์ แล้วบันทึกผลการทดลองลงในตารางที่ 4.4

โปรแกรมทดสอบที่ 4.4

```
*****
*   PROGRAM TEST MULTIBOARD   *
*       ON COMPUTER           *
*   [ LOAD REGISTER ]        *
*****
```

```
ORG $2300
```

```
LDAA #01 ; Load AccA
LDAB #23 ; Load AccB
LDX #4567 ; Load Index Register X
LDY #89AB ; Load Index Register Y
LDS #0020 ; Load Stack Pointer
LDAB #24 ; Goto Monitor
JSR $FF00
```

Step	Assembly	Condition of Register						
		PC	IY	IX	A	B	CCR	SP
1	LDAA ##01	2302	0000	0000	01	00	90	2050
2	LDAB ##23	2304	0000	0000	01	23	90	2050
3	LDX ##4567	2307	0000	4567	01	23	90	2050
4	LDY ##89AB	230B	89AB	4567	01	23	98	2050
5	LDS ##2200	230E	89AB	4567	01	23	90	2200

ตารางที่ 4.4 ผลการทดลองโปรแกรมทดสอบที่ 4.4

7. ตั้งแอดเดรสของโปรแกรมที่ \$2300 ป้อนโปรแกรมทดสอบที่ 4.5 ในรูปของภาษาแอสเซมบลี เมื่อป้อนแอสรีวแล้วให้กดคีย์ Enter อีกครั้งจะขึ้นเครื่องหมาย Prompt

8. ประมวลผลโปรแกรมที่แอดเดรส \$2300 โดยให้คำสั่ง G2300 ทำการบันทึกผลที่ได้ลงในตารางที่ 4.5

โปรแกรมทดสอบที่ 4.5

```
*****
*   PROGRAM TEST MULTIBOARD   *
*       ON COMPUTER           *
*   [ LOOP PROGRAM ]         *
*****
      ORG $2300
      LDAA #88      ; Control Word
      STAA 1703
LOOP   LDAA #66     ; Data Display
      LDAB #00     ; Start Counter
LOOP1  STAA 1701   ; Display Data
      STAB 1702   ; Scan Digit
      INCB      ; Counter+1
      CMPB #07   ; Check Counter
      BNE LOOP1 ; Next Digit Scan
      BRA LOOP  ; Goto Loop
```

โปรแกรมทดสอบที่	การแสดงผล
4.5	บนแอลอีดีแฟล็กจะติดสว่างเป็นรหัส \$66 ส่วนที่บนจอแสดงผลจะปรากฏเป็นเลข 4 ทุกหลัก และติดค้างอยู่ตลอด เพราะโปรแกรมมีการวนลูป

ตารางที่ 4.5 ผลการทดลองโปรแกรมทดสอบที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. ตั้งแอดเดรสของโปรแกรมที่ \$2300 จากนั้นป้อนโปรแกรมที่ 4.6 โดยการป้อนโปรแกรมจะอยู่ในรูปของภาษาแอสเซมบลี DEBUGGER 68HC11 เมื่อป้อนโปรแกรมเสร็จแล้วให้กดคีย์ Enter อีกครั้งจะปรากฏเครื่องหมาย Prompt ที่จอแสดงผล

10. ทำการประมวลผลโปรแกรมที่แอดเดรส \$2300 โดยใช้คำสั่ง G2300 บันทึกผลที่ได้จากการประมวลผลโปรแกรมลงในตารางที่ 4.6

โปรแกรมทดสอบที่ 4.6

```
*****
*   PROGRAM TEST MULTIBOARD   *
*   ON COMPUTER               *
*   [ SYSTEM CALL ]          *
*****
ORG $2300

LDAB #0C      ; System Call Subroutine Clear LCD
JSR  FF00
LDAA #0B      ; System Call Subroutine READY
JSR  CE2B
LDAB #24      ; Goto Monitor
JSR  FF00
```

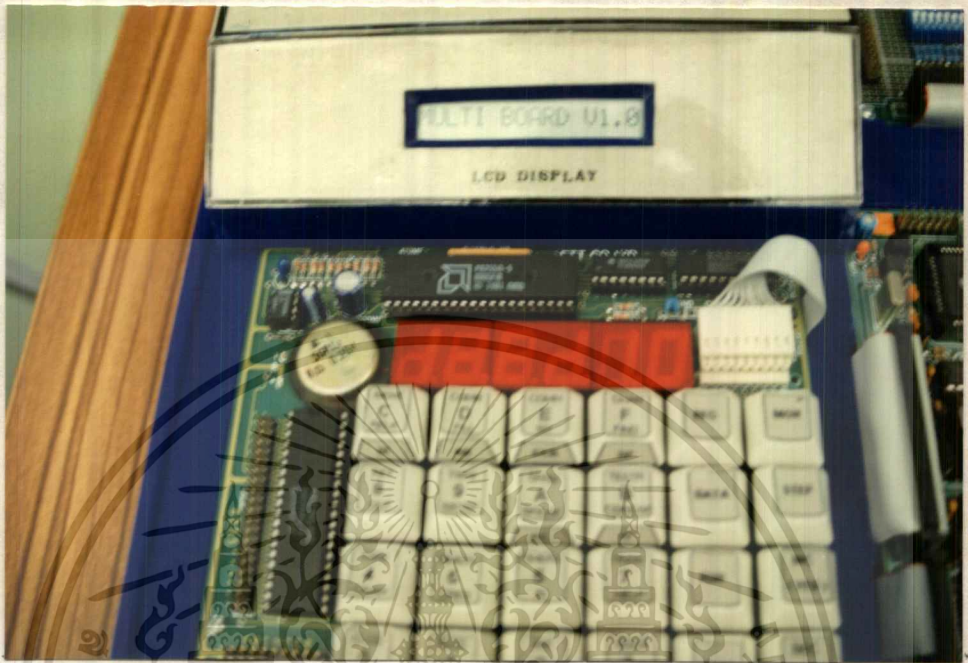
โปรแกรมย่อยที่	โปรแกรม	การแสดงผล
1	LDAB #\$0C JSR CALL	เคิลซ์ร์หน้าจอแอลซีดี
2	LDA A #\$0B JSR RECALL	จะแสดง rEADy ออกมาบนจอแสดงผล กดคีย์ INC จะเครื่องหมาย Prompt ของบอร์ด
3	LDA A #\$00 JSR RECALL	จะไม่กลับสู่โหมดดีบักเกอร์ของ DEBUGGER

ตารางที่ 4.6 ผลการทดลองโปรแกรมทดสอบที่ 4.6

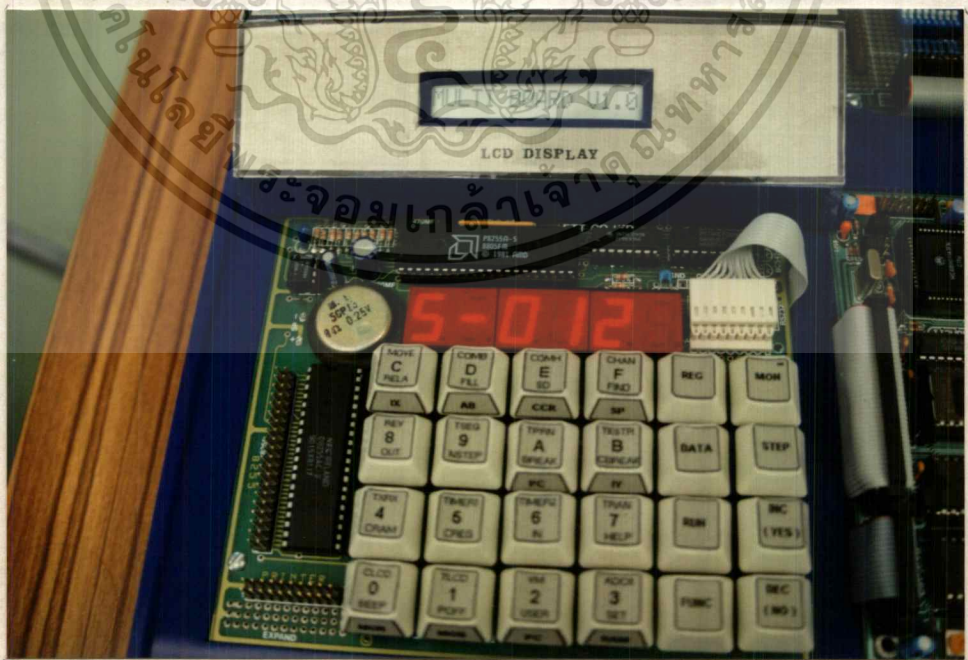
สรุปผลการทดลอง

จากผลการทดลองตารางที่ 4.4 และ 4.5 แสดงการทำงานของคำสั่ง T และ G ซึ่งใช้ในการประมวลผลโปรแกรมแบบ STEP และการประมวลผลโปรแกรมอย่างต่อเนื่องร่วมกับมัลติบอร์ดได้อย่างถูกต้อง

ในตารางที่ 4.6 จะเห็นว่า การเรียกใช้โปรแกรมย่อยที่ 1 และโปรแกรมย่อยที่ 2 ทำงานได้อย่างถูกต้อง แต่การเรียกใช้โปรแกรมย่อยเพื่อกลับสู่โหมดดีบักเกอร์ของ DEBUGGER ไม่สามารถทำงานได้ เนื่องจากคณะผู้จัดทำไม่มีโปรแกรมต้นฉบับ (Source files) ของโปรแกรมในส่วนของ DEBUGGER จึงเขียนโปรแกรม SYSTEM CALL มีโครงสร้างต่างกัน เมื่อจบการทำงานจากโปรแกรมย่อยแล้วโปรแกรมเคาน์เตอร์จึงหาค่าแห่งเอกซ์ควิต์คำสั่งต่อไปไม่เจอ และ DEBUGGER ขณะที่ประมวลผลโปรแกรมจะติดต่อกับพอร์ต RS-232 อยู่ตลอดเวลา ดังนั้นจึงทำให้การเรียกเข้าโหมดดีบักเกอร์ของ DEBUGGER เกิดทำงานผิดพลาด ดังการทดลอง ซึ่งส่วนนี้หากจะทำการแก้ไขจะต้องรู้โครงสร้างระบบ SYSTEM CALL ของ DEBUGGER ด้วยจึงจะสามารถแก้ไขได้รวดเร็วขึ้น

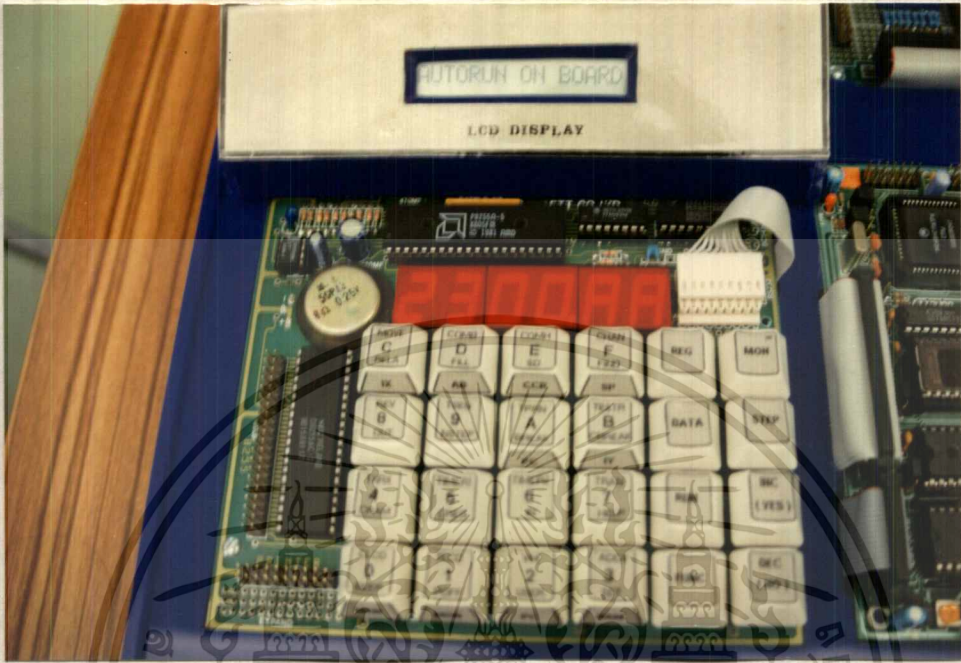


รูปที่ 4.2 ตัวอักษรที่ปรากฏเมื่อจ่ายไฟให้กับมัลติบอร์ด

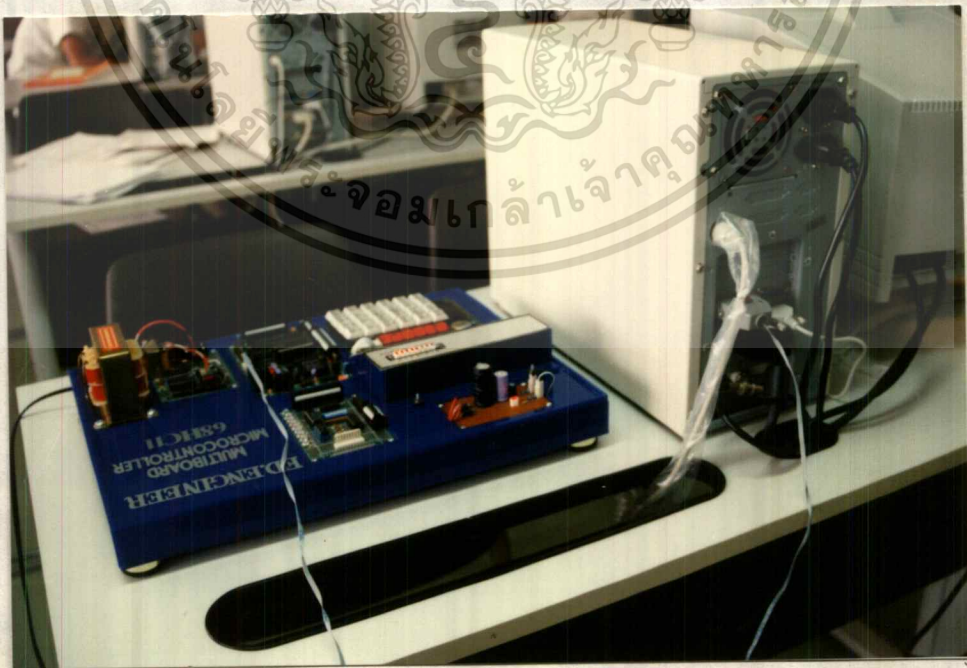


รูปที่ 4.3 ข้อความที่ให้เลือกโหมดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การทำงานบนบอร์ด



รูปที่ 4.5 การเชื่อมต่อเพื่อทำงานบนเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้กับเจ้าของเพื่อการค้าเท่านั้น เมื่อเผยแพร่ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 สรุปเปรียบเทียบเทียบบผลการทดลอง

เมื่อทำการประมวลผลโปรแกรมแบบ TRACE บนมัลติบอร์ดและทำการประมวลผลบนเครื่องคอมพิวเตอร์จะให้ผลการทดลองที่เหมือนกัน อธิบายเปรียบเทียบได้ดังนี้

ค่าของแอสต์คพอยน์เตอร์ (Stack Pointer) ที่ได้จากการประมวลผลโปรแกรมบนมัลติบอร์ดและการประมวลผลโปรแกรมบนเครื่องคอมพิวเตอร์จะแตกต่างกัน เนื่องจากว่าโปรแกรมมอนิเตอร์ของทั้งสองจะมีการไหลคอสต์คไปต่างกันและค่าโปรแกรมเคอร์เนลแอสต์คที่ละคำสั่ง ในสแต็คที่ 4 นั้นรีจิสเตอร์ที่แสดงสภาวะจะเป็น \$98 เพราะว่าถ้าบิตที่ 15 ของรีจิสเตอร์ IY เป็น "1" จะทำให้บิต N ในรีจิสเตอร์แสดงสภาวะเกิดการเซ็ต

การประมวลผลโปรแกรมแบบต่อเนื่องในโปรแกรมทดสอบจะมีการวนลปโปรแกรมอยู่ตลอดเวลาจะไม่มีมีการกลับสู่มอนิเตอร์ของมัลติบอร์ดหรือ DEBUGGER ดังนั้นจึงไม่ปรากฏเครื่องหมาย Prompt ของทั้งมัลติบอร์ดและ DEBUGGER และการประมวลผลบนมัลติบอร์ดกับการประมวลผลโปรแกรมบนเครื่องคอมพิวเตอร์จะให้ผลการประมวลผลที่เหมือนกัน

กรณีที่มีการประมวลผลโดยทดลองเรียกเป็นโปรแกรมย่อยผ่าน SYSTEM CALL ของมัลติบอร์ดผ่านเครื่องคอมพิวเตอร์ เมื่อจบจากการประมวลผลโปรแกรมย่อยแล้วจะสามารถกลับสู่มอนิเตอร์ของมัลติบอร์ดและขึ้นเครื่องหมาย Prompt ได้ แต่กรณีทำการประมวลผลบนเครื่องคอมพิวเตอร์นั้น หลังจากการเรียกใช้โปรแกรมย่อยของ DEBUGGER การทำงานโดยการเรียกใช้โปรแกรมย่อยที่ 2 ของมัลติบอร์ดจะมีการเรียกใช้งานถูกต้อง แต่หลังจากจบที่โปรแกรมแล้วจะไม่สามารถกลับสู่มอนิเตอร์ของ DEBUGGER บนเครื่องคอมพิวเตอร์ได้

เนื่องจากระบบ SYSTEM CALL ของมัลติบอร์ดและ DEBUGGER มีโครงสร้างของโปรแกรมที่ต่างกัน เมื่อนำโปรแกรมมอนิเตอร์ของ DEBUGGER ที่จะทำการประมวลผลบนเครื่องคอมพิวเตอร์เรียกโปรแกรมย่อยผ่านระบบ SYSTEM CALL ของมัลติบอร์ด มันจึงไม่สามารถหาตำแหน่งกลับของโปรแกรมย่อยได้ แต่สามารถเรียกโปรแกรมย่อยของมัลติบอร์ดได้โดยผ่านการเรียกเข้าสู่ตำแหน่งแอดเดรสของโปรแกรมย่อยโดยตรง

บทที่ 5

บทวิจารณ์ สรุป และแนวทางในการพัฒนา

5.1 บทสรุป

มัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11 ในปริศยานิพนธ์ฉบับนี้ สร้างขึ้นเพื่อจะนำไปใช้ศึกษาและเรียนรู้การทำงานด้านต่าง ๆ ของไมโครคอนโทรลเลอร์ 68HC11 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีประสิทธิภาพสูงตัวหนึ่ง ในการสร้างนั้นคณะผู้จัดทำได้วางขอบเขตการสร้างและการพัฒนาเพื่อเป็นแนวทางการออกแบบไว้ดังนี้

- ใช้ไมโครคอนโทรลเลอร์เบอร์ 68HC11 เป็นชิพ
- มีการแสดงผลทางเอาต์พุตบนจอแอลซีดีโมดูล 16 ตัวอักษร 1 แถว, พอร์ตเสียง และมีแอลอีดี 7 เซ็กเมนต์ 6 หลัก โดยอาศัยโปรแกรมควบคุมเอาต์พุตเป็นตัวทำการเลือกการแสดงผลและควบคุมการแสดงผลทางอุปกรณ์เอาต์พุต
- มอนิเตอร์ของมัลติบอร์ดใช้จอแอลซีดี 7 เซ็กเมนต์ 6 หลัก
- การพัฒนาโปรแกรมสามารถเขียนโปรแกรมบนเครื่องคอมพิวเตอร์ แล้วก็ส่งผ่านไปยังพอร์ต RS-232 เพื่อทำการประมวลผลโปรแกรมบนบอร์ด หรือจะพัฒนาโปรแกรมโดยแปลโปรแกรมเป็นเลขฐาน 16 แล้วป้อนผ่านเคย์บอร์ดของมัลติบอร์ดให้ชิพประมวลผล
- ใช้ฮาร์ดแวร์ขนาด 8 กิโลไบต์ หรือ 32 กิโลไบต์ เก็บโปรแกรมมอนิเตอร์ของบอร์ดและโปรแกรมตัวอย่าง เพื่อความเข้าใจการทำงานของชิพมากขึ้น ซึ่งในส่วนของเงื่อนไขและขอบเขตที่ได้เสนอนั้นก็สามารถทำให้ลุล่วงลงไปได้ด้วยดี และนอกจากนี้ก็ยังได้มีการเพิ่มขีดความสามารถของมัลติบอร์ด โดยมีฟังก์ชันอำนวยความสะดวกสำหรับการเขียนโปรแกรมและทดสอบการทำงานของโปรแกรมต่าง ๆ อันได้แก่ ฟังก์ชันรีเลย์, ฟังก์ชันเคลียร์แรม, ฟังก์ชันสำหรับเคลียร์รีจิสเตอร์ เป็นต้น และจะแยกการทำงานของมัลติบอร์ดเป็น 2 ส่วน คือ การประมวลผลบนเครื่องคอมพิวเตอร์ซึ่งติดต่อสื่อสารกับตัวบอร์ดด้วย PROCOMM PLUS ผ่านพอร์ต RS-232 หรือจะประมวลผลบนมัลติบอร์ดโดยตรงก็ได้

นอกจากนี้ก็ยังได้สร้างระบบ SYSTEM CALL โดยสามารถเรียกใช้โปรแกรมย่อยในโปรแกรมมอนิเตอร์ได้ จึงช่วยอำนวยความสะดวกในการพัฒนาโปรแกรม ตลอดจนได้บรรจุฟังก์ชันคีย์เพื่อแสดงการประยุกต์ใช้ 68HC11 ในงานทั่วไปเบื้องต้นและมีพอร์ตสำหรับใช้ในการขยายระบบ การทดลองหรือควบคุมระบบภายนอกอีกด้วย

5.2 ปัญหาและแนวทางแก้ไข

ในระหว่างการทำปริศยานิพนธ์ ทางคณะผู้จัดทำได้พบกับปัญหาและอุปสรรคต่าง ๆ ที่ทำให้เสียเวลาในการแก้ไข ซึ่งปัญหาบางอย่างอาจเล็กน้อย แต่ถ้าเรามองข้ามไปจะทำให้เสียเวลามากแก้ไขในภายหลัง ปัญหาและแนวทางการแก้ไขได้รวบรวมไว้ดังตารางที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และตารางที่ 5.2

ปัญหา	แนวทางการแก้ไข
<ul style="list-style-type: none"> - แอสเซมเบลอร์ C32 ไม่สามารถทำการแปลงคำสั่งของไอซี 68HC11 บางคำสั่งได้ เช่น BSET 0,X 50H และ BCLR 0,X 80H - รูปแบบของการเขียนโปรแกรมให้แอสเซมเบลอร์ C32 จะแตกต่างกับชุดคำสั่งของ MOTOROLA ที่นำมาใช้อ้างอิง อย่างเช่น เครื่องหมาย \$ นำหน้าคำสั่ง แสดงว่าเลขนั้นจะเป็นเลขฐาน 16 แต่ในรูปแบบของ C32 ต้องใช้ตัว H ตามหลังตัวเลข และการใช้รหัสเทียบที่แตกต่างกัน 	<ul style="list-style-type: none"> - ต้องหาแอสเซมเบลอร์ตัวใหม่มาใช้แทน โดยจะต้องเปลี่ยนมาใช้ AS11NEW.EXE เป็นแอสเซมเบลอร์ของบริษัท MOTOROLA
<ul style="list-style-type: none"> - เขียนโปรแกรมไปตามรูปแบบของ MOTOROLA แต่ว่าเมื่อเรานำไปผ่านแอสเซมเบลอร์แล้ว กลับรายงานผลการทำงานว่าเกิดข้อผิดพลาดในการเขียนโปรแกรม (ERROR) 	<ul style="list-style-type: none"> - เนื่องจากว่า LABEL ของชุดคำสั่งนั้นจะต้องเขียนติดกับคอลัมน์แรก ถ้าไปเขียนไว้ที่คอลัมน์ที่ไม่ติดเส้นเริ่มต้นจะแอสเซมเบลอร์ไม่ได้ก็จึงเลื่อน LABEL มาไว้ที่คอลัมน์ 1
<ul style="list-style-type: none"> - รีจิสเตอร์ใช้งานมีน้อยไม่เพียงพอต่อการเขียนโปรแกรม 	<ul style="list-style-type: none"> - จะต้องใช้พื้นที่หน่วยความจำจำลองเป็นรีจิสเตอร์เทียบและอาศัยการเก็บค่าลงสแต็ก

ตารางที่ 5.1 ปัญหาในการทำปริญญานิพนธ์และแนวทางการแก้ไข

ปัญหา	แนวทางการแก้ไข
<p>- อิมูเลเตอร์ไม่สามารถส่งผ่านข้อมูลที่ต้องการลงได้ โดยแอลอ์ดีดวงสีแดงกับสีเขียวติดพร้อมกัน</p>	<p>- เอาเศษตะกั่วที่ติดอยู่ในลายวงจรพิมพ์ออก และสายที่เชื่อมระหว่างอ็พรอมจำลองของชุดอิมูเลเตอร์กับอ็พรอมบนบอร์ดเกิดขาดภายใน จึงทำการเปลี่ยนสายเชื่อมใหม่เข้าไป</p> <p>- ให้ทำการประมวลผลโปรแกรมโดยใช้ชุด DEBUGGER 68HC11 อาศัยการดาวน์โหลดโปรแกรมลงพื้นที่แรมแทนการใช้อิมูเลเตอร์</p>
<p>- โปรแกรมแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัลที่เขียนบน DEBUGGER จะทำงานได้แต่เมื่อตั้ง AUTO START ให้ประมวลผลบนแรมโดยที่ไม่ผ่านชุด DEBUGGER แล้วแปลงสัญญาณไม่ได้</p>	<p>- โปรแกรมไม่ได้ทำการติดต่อกับรีจิสเตอร์บางตัวที่ทำหน้าที่การอ่านาเบิ้ลสัญญาณนาฬิกา ให้แก้ตัวแปลงสัญญาณ ดังนั้นสัญญาณนาฬิกาจึงถูกคัสเอเบิ้ลอยู่ แก้ไขโดยทำการติดต่อกับรีจิสเตอร์ตามที่ DEBUGGER ได้ตั้งค่าไว้</p>

ตารางที่ 5.2 ปัญหาในการทำปริญญาโทและแนวทางแก้ไข

5.3 แนวทางในการพัฒนา

ทางคณะผู้จัดทำได้พยายามที่จัดทำมัลติบอร์ดให้มีประสิทธิภาพมากที่สุดเท่าที่จะทำได้ โดยทำการพัฒนาตลอดภาคการศึกษาอย่างต่อเนื่อง แต่ด้วยระยะเวลาและงบประมาณที่มีอยู่อย่างจำกัด ตลอดจนยังเป็นไมโครคอนโทรลเลอร์ที่ใหม่และเป็นการเริ่มศึกษา ก็มีข้อจำกัดบางจุดที่ควรจะได้มีการพัฒนาให้มีประสิทธิภาพที่ดีขึ้นต่อไป

1. การป้อนโปรแกรม เราจะสามารถพัฒนาให้ป้อนโปรแกรมที่เป็นภาษาแอสเซมบลีบนเคียบอร์ดของมัลติบอร์ดได้โดยตรง ซึ่งในการป้อนโปรแกรมเป็นภาษาเครื่องนี้ ถ้าหากว่าเป็นการแปลงโปรแกรมด้วยมือ (MANUAL) จะใช้เวลาในการแปลงมาก แม้ว่ามัลติบอร์ดจะสามารถดาวน์โหลดโปรแกรมจากเครื่องคอมพิวเตอร์มาประมวลผลบนบอร์ดได้ แต่ก็ไม่ใช่เป็นการสะดวกในการเคลื่อนย้ายไปมาเท่าใดนัก เพราะจำเป็นต้องอาศัยเครื่องคอมพิวเตอร์เพื่อแอสเซมเบลอร์โปรแกรมที่เราเขียนขึ้นมาอยู่นั่นเอง

2. ในส่วนของ DEBUGGER 68HC11 บนเครื่องคอมพิวเตอร์สามารถเขียนขึ้นมาใช้

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการศึกษาดูงาน เมื่ออนุญาตเห็นชอบเรียบร้อยแล้ว

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราก็สามารถเปลี่ยนแปลงส่วนใด ๆ ของ DEBUGGER ได้ตามต้องการ

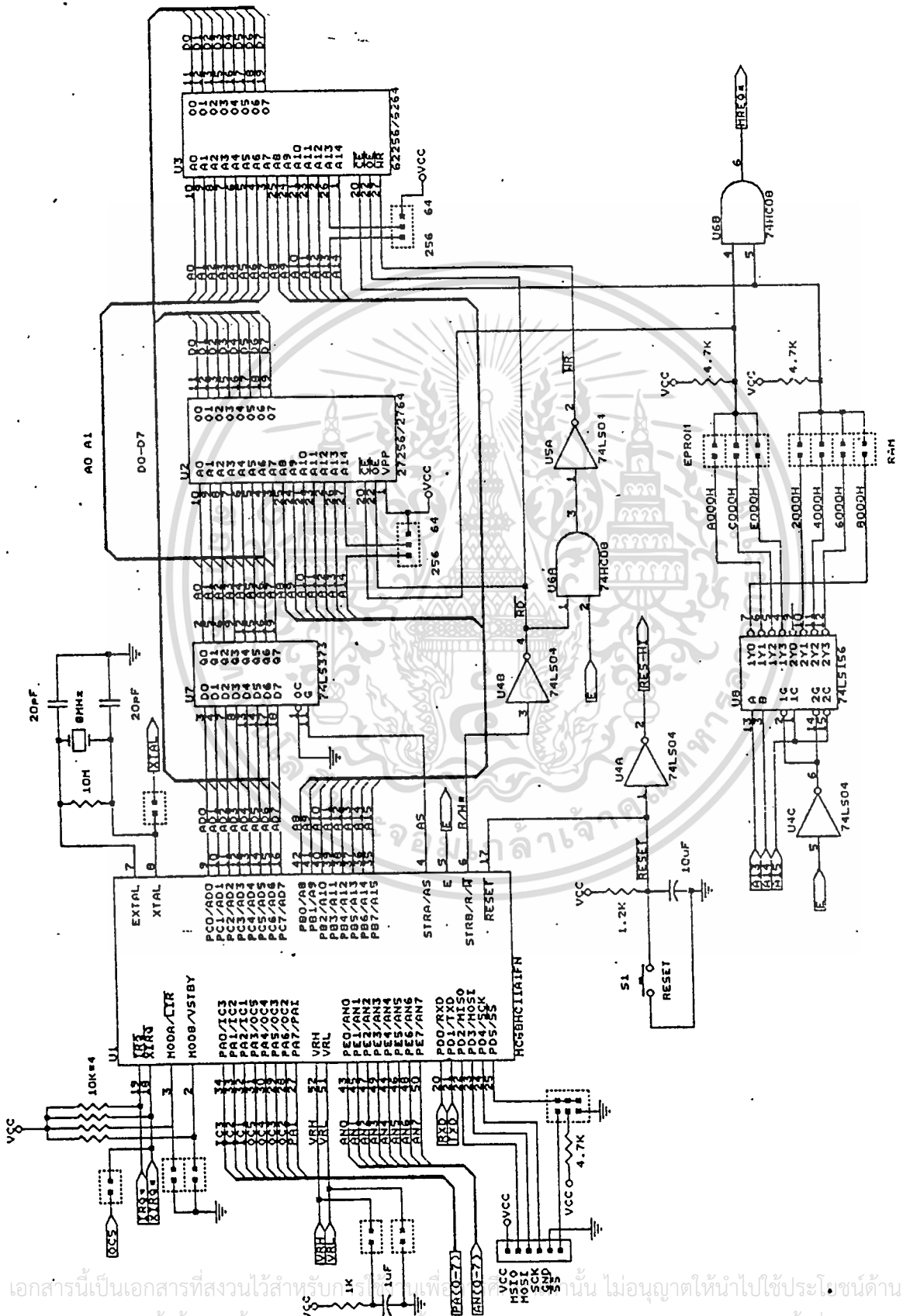
3. ขนาดของมัลติบอร์คยังมีขนาดใหญ่ ควรพัฒนาให้อยู่บนแผ่นวงจรพิมพ์เดียวกันเป็นซิงเกิ้ลบอร์คจะทำให้นำไปใช้ยิ่งขึ้น

4. การออกแบบวงจรหากออกแบบให้สมบูรณ์ โดยออกแบบวงจรเองทั้งหมดจะทำให้ลดการใช้อุปกรณ์การเชื่อมต่อกับระบบภายนอกให้น้อยลง ทำให้การทำงานของบอร์คแน่นอนและแม่นยำยิ่งขึ้น



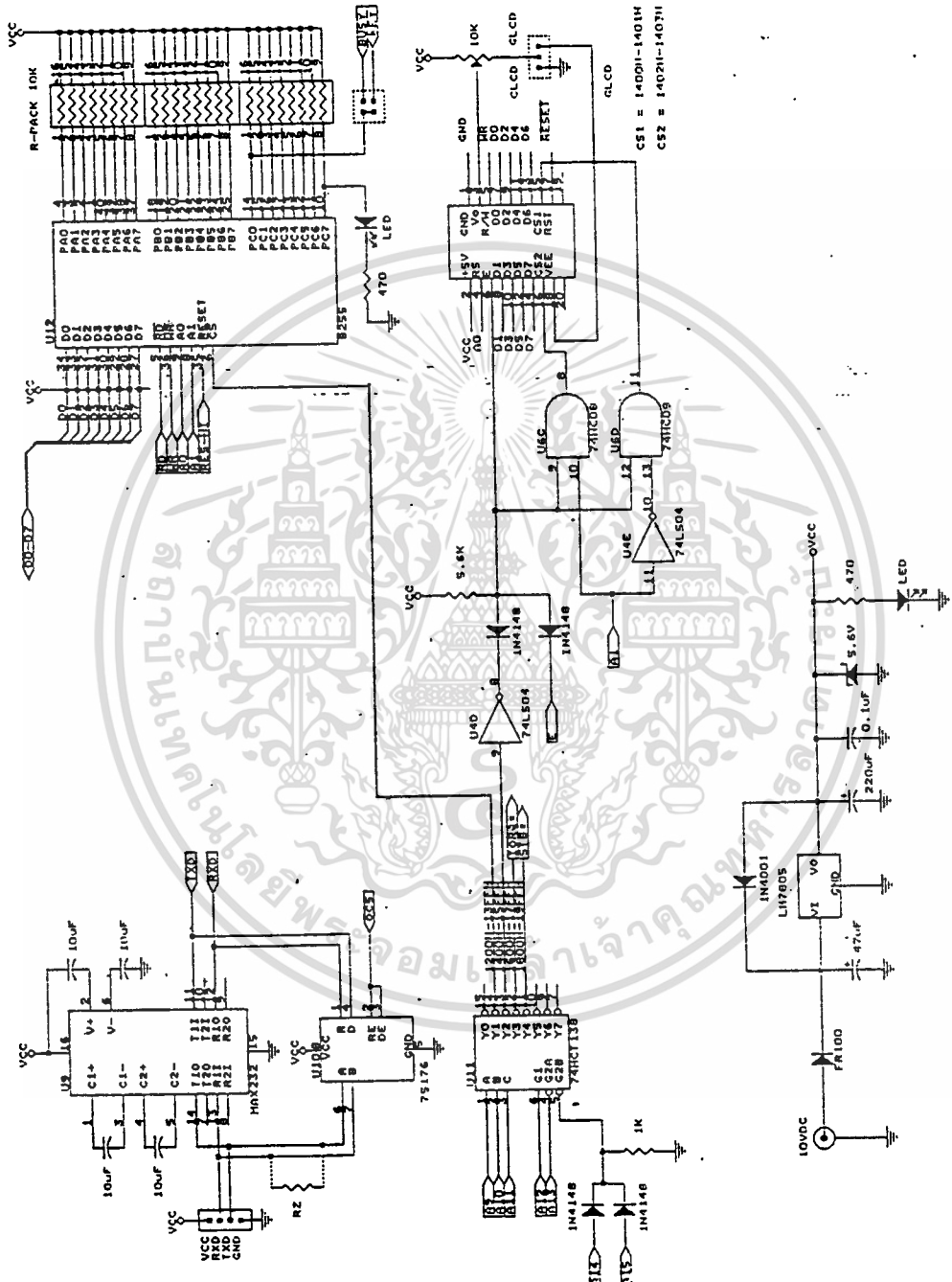


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



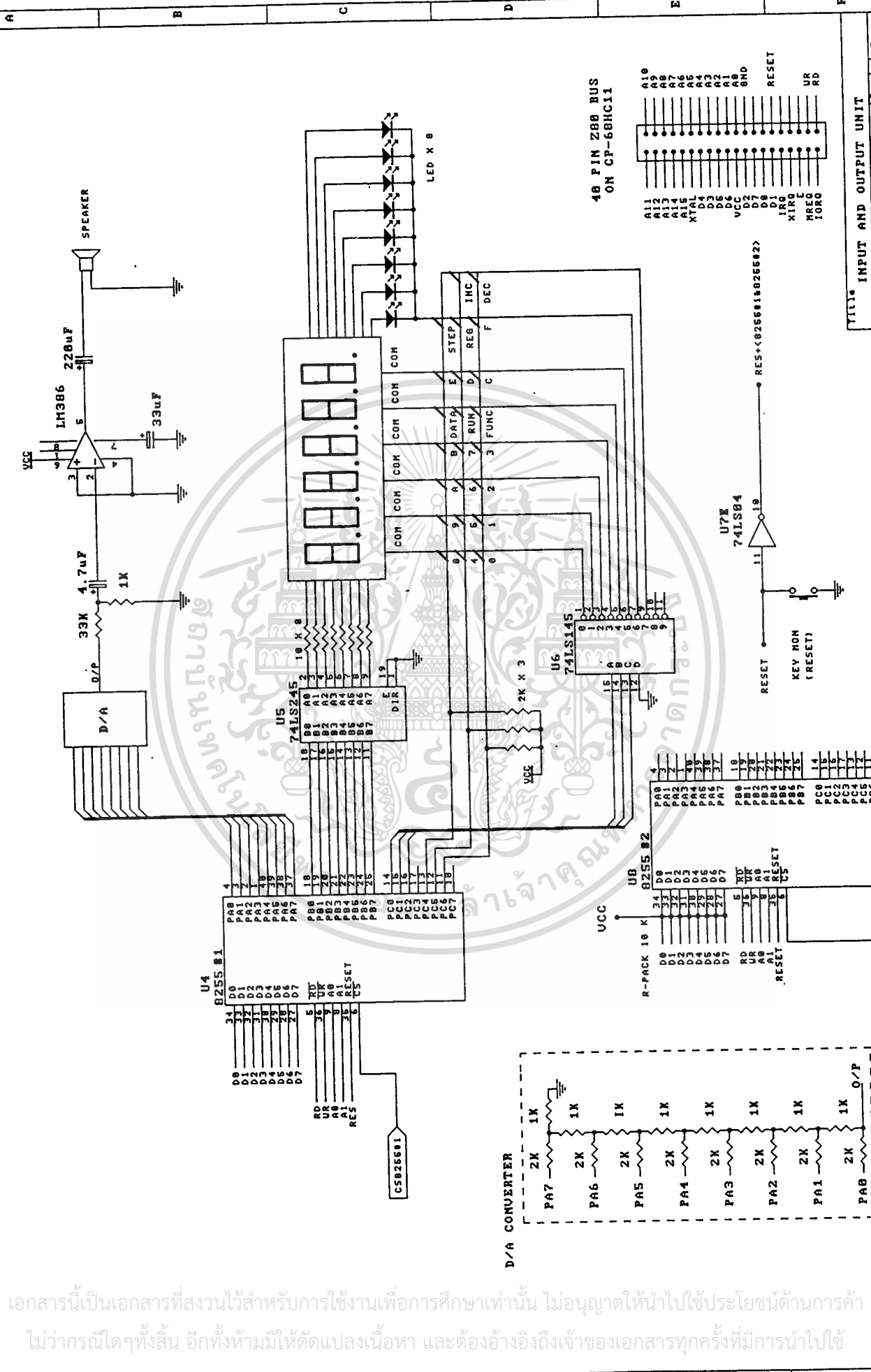
Title		MEMORY	
Size	Number	Revision	
A3	MULTIBORAD 68HC11		
File	22-00	Sheet	of
Date	6/11/87	Drawn By	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับก... ไม่นุญาติให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



TITLE		I/O DEVICE	
NUMBER	A3	REVISION	
MULTIBORAD 68HC11			
SHEET	1	OF	1
DATE	23-01-1994	DESIGNER	SM
FILE	68HC11	DR	SM

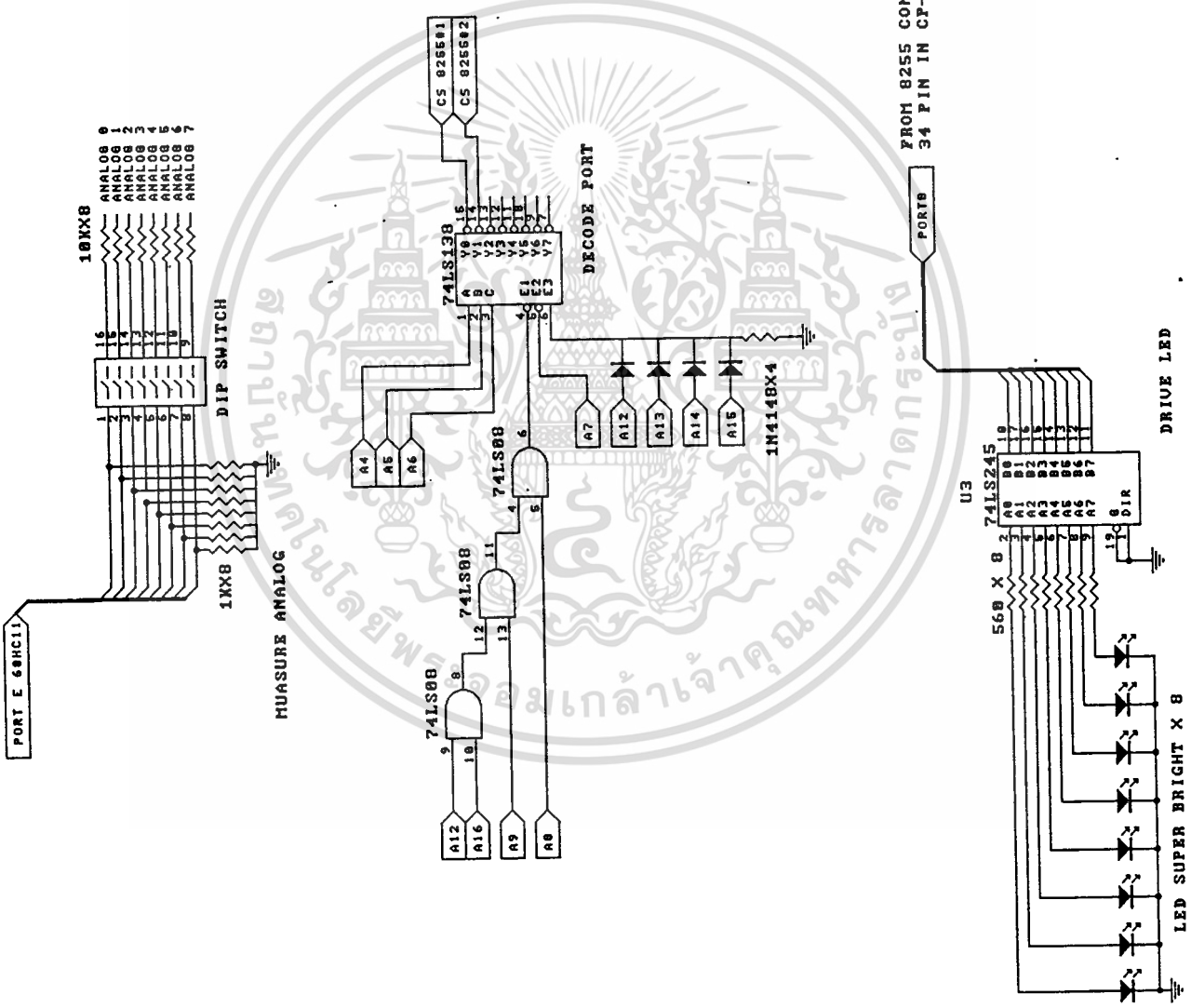
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CS 8365 #2

TITLE INPUT AND OUTPUT UNIT	
Size	Revision
A3	
MULTIBORAD 68HC11	
DATE: 23-01-83	Sheet of
FILE: C:\NEU\7\931	Drawn By:

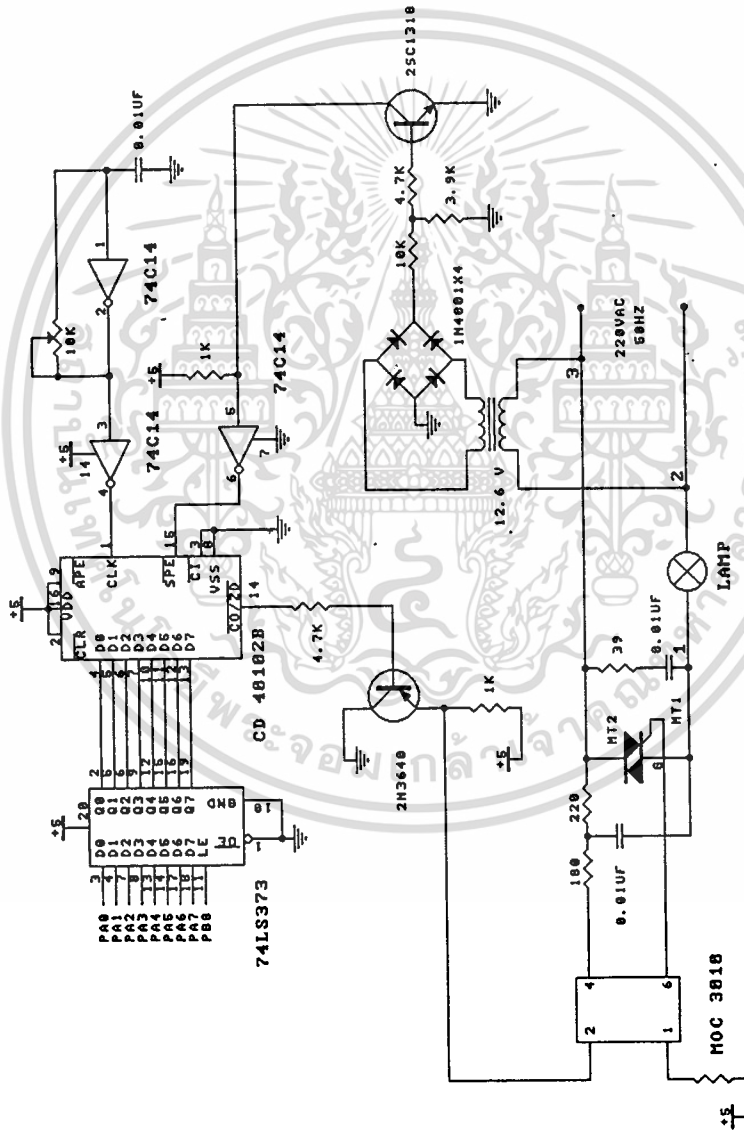
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FROM 8255 CONNECTOR
34 PIN IN CP-68HC11

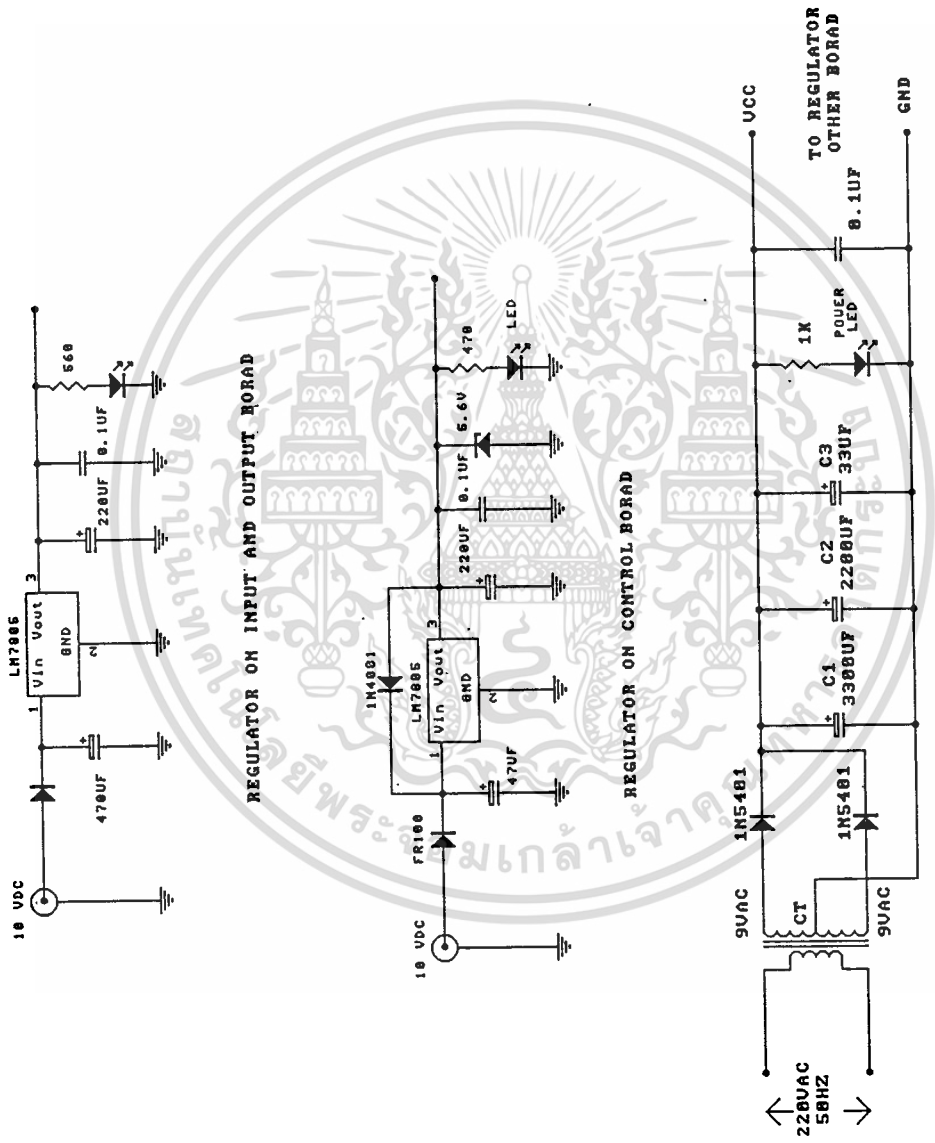
INTERFACE	
SIZE	NUMBER
A3	MULTIBOARD 68HC11
Revision	
FILE: 23-007	SHEET
FILE: 61XHEU77	FROM NO. 01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title Digital Right Intensity	
Size A3	Number MULTIBORAD 68HC11
Date 23-01-1994	Revision
File 68REV1	Sheet 1 of 1
Desain By	



RECTIFIER AND FILTER

TITLE		POWER SUPPLY	
Size	Number	Revision	Revision
A3	MULTIBORAD	68HC11	
DATE: 23-06-1991	DESIGNER: C. NEUVY	DRAWN BY:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของการแสดงผลบนแอลอีดี 7 เซกเมนต์

ตัวอักษร	ความหมาย
XXXX-A	บ่อนแอคเตอร์
XXXX-b	บ่อนแอคเตอร์ BREAK โปรแกรม
XXXX-S	บ่อนแอคเตอร์เริ่มต้น
XXXX-F	บ่อนแอคเตอร์สุดท้าย
XXXX-d	บ่อนแอคเตอร์ปลายทาง
XXXX-H	บ่อนเลขฐาน 16
XXXX-P	บ่อนหมายเลขพอร์ต
XXXX-n	บ่อนหมายเลขของเสตป
XXXX-L	บ่อนความยาวของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษร

ความหมาย

Ed. Id. 1.0

Education Industry เวอร์ชัน 1.0

Func. 1

ฟังก์ชันคีย์ 1

Func. 2

ฟังก์ชันคีย์ 2

Error

แสดงข้อผิดพลาด

On

เปิดเสียงออกลำโพง

OFF

ปิดเสียงที่ออกลำโพง

CLEAR

เคลียร์

SET-

เซ็คคีย์

USER

เลือกใช้คีย์

rEADy

ถามความพร้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษร

ความหมาย

S-012

เลือกโหมดการเริ่มต้นการทำงาน

LATCH

แลตช์ข้อมูล

Fron

จากข้อมูล

to

เป็นข้อมูล

XXXX-P

โต้คผ่าน

SELECT

ให้เลือกคีย์ INC หรือ DEC

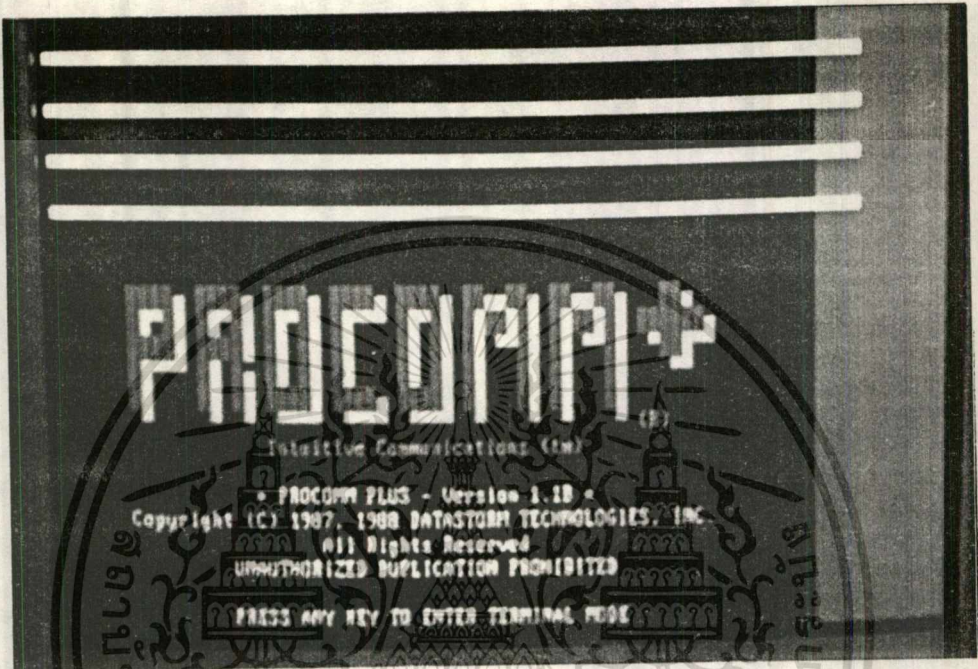
HELP

แสดงการทำงานใน HELP MENU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเรียกใช้โปรแกรม PROCOMM

นำแผ่นซอฟต์แวร์ PROCOMM ใส่ในดิสก์ไดรฟ์แล้วเรียก PCP จะมีข้อความขึ้นดังรูป

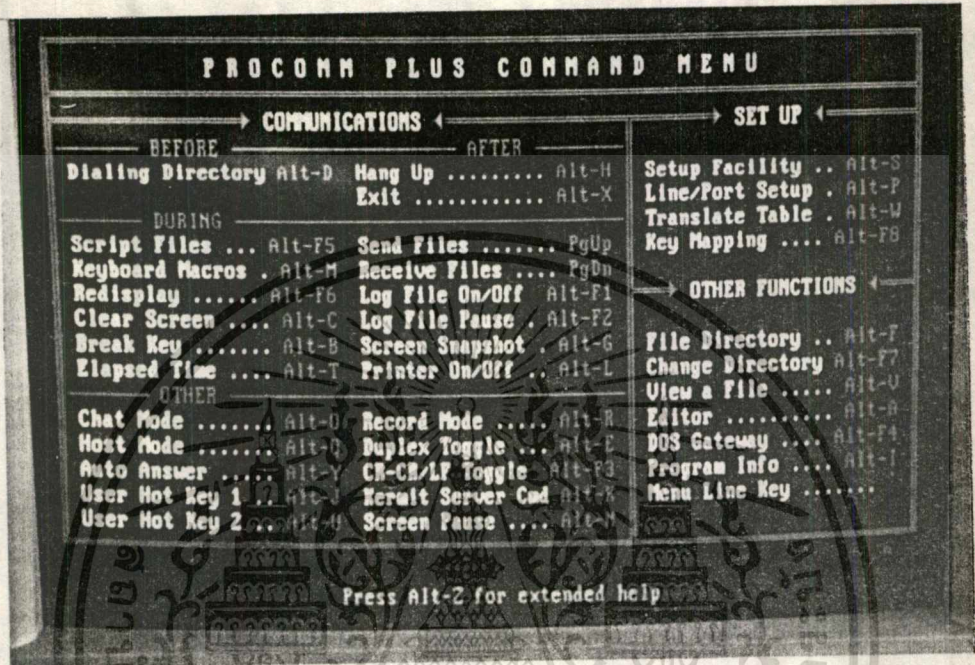


หลังจากนั้นให้กดคีย์ใด ๆ ก็เข้าสู่การทำงานของ PROCOMM เพื่อที่จะใช้ติดต่อกับ BOARD CP-68HC11 ภาษาใน PROCOMM จะมี HELP MEMU เพื่อช่วยอธิบายรายละเอียดอย่างคร่าว ๆ ของการใช้งานใน PROCOMM หลังจากที่เรากดคีย์ ALT Z ซึ่งเป็นการขอความช่วยเหลือ ใน HELP MENU จะบอกสถานะต่าง ๆ ดังต่อไปนี้

- การตั้งอัตราบอดใช้ ALT P เพื่อเข้าสู่การเลือกแล้วต้องการบันทึกไว้อย่างถาวรก็ใช้ ALT S
- การไหลลดข้อมูลจากเครื่องคอมพิวเตอร์ใช้คีย์ PAGE UP
- การไหลลดจาก มัลติบอร์ด ขึ้น เครื่องคอมพิวเตอร์ใช้คีย์ PAGE DOWN
- การออกจาก PROCOMM และเวลากลับเข้ามาใหม่ไม่ต้องเรียก PROCOMM อีกให้กดคีย์ ALT F4
- การเรียกดูโปรแกรมใช้คีย์ ALT V
- การเปิดหน้าจอเพื่อเก็บข้อมูลใช้คีย์ ALT F1
- การเปลี่ยน PATH ของการทำงานใช้คีย์ ALT F7
- การเคลียร์หน้าจอแสดงผลใช้คีย์ ALT C
- การเปิดเครื่องพิมพ์เพื่อนำสิ่งที่อยู่บนจอพิมพ์ออกทางเครื่องพิมพ์ใช้คีย์ ALT L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การใช้งานมัลติบอร์ดไมโครคอนโทรลเลอร์ 68HC11

มัลติบอร์ดในปริศยานี้จะแบ่งการทำงานเป็น 2 โหมด คือ โหมด 1 และ โหมด 2

โหมด 1 เป็นการทำงานบนมัลติบอร์ดทั้งหมด การป้อนโปรแกรมจะทำได้โดยให้ป้อนแมชชีนโค้ด (Machine code) ผ่านคีย์บอร์ดของตัวมัลติบอร์ด

โหมด 2 จะทำงานบนเครื่องคอมพิวเตอร์ สามารถแบ่งย่อยออกเป็น 2 ลักษณะ คือ

- 1) เขียนโปรแกรมภาษาแอสเซมบลีใน Editor แล้วแอสเซมเบลอร์โดยให้กดคีย์เลือกแฟ้มที่ชื่อ AS11NEW.EXE หลังจากเครื่องแสดงจำนวนที่ผิดพลาดเท่ากับ 0 แล้วก็จะทำการดาวน์โหลดข้อมูลจากเครื่องคอมพิวเตอร์ลงบนมัลติบอร์ดโดยผ่านพอร์ต RS-232C การประมวลผลโปรแกรมจะควบคุมโดยคีย์บอร์ดของมัลติบอร์ด

- 2) เขียนโปรแกรมในส่วนของ DEBUGGER และประมวลผลโปรแกรมผ่านคีย์บอร์ดของเครื่องคอมพิวเตอร์ การเปลี่ยนแปลงภายในไมโครคอนโทรลเลอร์จะแสดงผลแตกต่างจากที่กล่าวมาข้างต้น คือจะแสดงบนจอมอนิเตอร์ของเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดการใช้งาน

เมื่อจ่ายไฟให้มัลติบอร์ดโดยกดสวิตช์ POWER จะมีตัวอักษรดังรูป

Ed. Id. 1.0

ลักษณะของการวิ่ง โดยจะวิ่งจากจอแสดงผลด้านซ้ายมาหยุดที่กลางจอแล้ววิ่งหายไปทางขวาของจอแสดงผลพร้อมกับมีเสียงดัง 2 ระดับ หลังจากนั้นจะปรากฏตัวอักษรขึ้นที่จอแสดงผลอีกช่วงเวลาหนึ่งแล้วจะหายไป

S-012

หลังจากนั้นจะขึ้นเครื่องหมาย Prompt ทางขอบซ้ายของจอแสดงผลดังรูป

—

ขณะที่แสดงตัวอักษรให้เลือกโหมดการทำงาน ถ้าเราไม่กดคีย์ใด ๆ ที่จอแสดงผลจะขึ้นเครื่องหมาย Prompt หมายถึงว่าเป็นการเลือกโหมดแรก นั่นก็คือการเลือกโหมด 0 ของ SELECT MODE และถ้าระหว่างที่แสดงเครื่องหมายให้เลือกโหมดการทำงาน เราได้กดคีย์หมายเลข 1 จะเป็นการเลือกโหมด 1 ของ SELECT MODE ซึ่งเป็นการทำงานบนมอเตอร์โปรแกรมของมัลติบอร์ดเช่นเดียวกับกับการเลือกโหมด 0 ของ SELECT MODE แต่ถ้าหากเรากดคีย์ 2 จะเป็นการเลือกโหมด 2 ของ SELECT MODE ซึ่งเป็นการทำงานในโหมดของ DEBUGGER โดยจะทำการประมวลผลบนคีย์ของเครื่องคอมพิวเตอร์ ที่เชื่อมต่อกับตัวมัลติบอร์ดและควบคุมการสื่อสารข้อมูลโดย PROCOMM การทำงานส่วนนี้จะกล่าวถึงในภายหลัง บางครั้งตัวเลือกจะเพิ่มขึ้นอีก 1 ตัวดังรูป

S-0123

ตัวเลือกที่เพิ่มขึ้นมานั้นก็คือโหมด 3 ของ SELECT MODE นั้นเอง จะเกิดขึ้นในกรณีที่เราต้องการตั้ง AUTO START ตัวอย่างเช่น เราเขียนโปรแกรมมอนิเตอร์ของเราไว้ที่แอดเดรส \$2300 ซึ่งเป็นพื้นที่ของแรมขนาด 32 กิโลไบต์ เราสามารถตั้ง AUTO START โดยใส่ \$A3 ไว้ในแอดเดรสของแรมที่ตำแหน่ง \$9FFF ก็จะขึ้นตัวเลือกเพิ่มขึ้นมาดังกล่าว ดังนั้นถ้าเราจะให้การทำงานของมัลติบอร์ดมาเริ่มต้นที่โปรแกรมมอนิเตอร์ที่เขียนขึ้นระหว่างการแสดงผลของ SELECT MODE เราต้องกดคีย์หมายเลข 3 ถ้าหากเราไม่กดคีย์เพื่อเลือกโหมดใด ๆ ก็จะทำให้ผลดังที่กล่าวมาข้างต้น

รายละเอียดของคีย์บอร์ดและการใช้งาน

คีย์บอร์ดประกอบด้วยคีย์ตัวเลข 16 คีย์และคีย์คำสั่ง 8 คีย์อยู่ทางด้านขวาของบอร์ด ในคีย์เหล่านี้มีอยู่ 5 คีย์ที่เรียกว่า Instant keys คือคีย์ REG, DATA, RUN, FUNC และคีย์ STEP หมายถึงคีย์เหล่านี้จะกดได้ทันทีที่ต้องการใช้ไม่ว่าจะอยู่ในการทำงานส่วนใดอยู่ก็ตามที่เป็นการรอคีย์อยู่ รายละเอียดแต่ละคีย์คำสั่งจะอธิบายได้ดังนี้

คีย์ MON

คีย์นี้จะต่ออยู่กับขา RESET ของ 68HC11 หมายถึง เป็นการรีเซ็ตทางฮาร์ดแวร์ คีย์นี้จะสามารถกดได้ทุกกรณีไม่ว่ามัลติบอร์ดจะทำงานอยู่ในสภาวะใด เมื่อคีย์นี้ถูกกดจะเข้าสู่การทำงานของโปรแกรมมอนิเตอร์ทันที โดยมีเครื่องหมาย Prompt บนจอแสดงผล เป็นการเตรียมพร้อมที่จะทำงานตามคีย์อื่น ๆ ต่อไป

โปรแกรมมอนิเตอร์คือโปรแกรมที่บรรจุอยู่ในอ็พรอม ซึ่งจัดให้เป็นโปรแกรมเกี่ยวกับการใช้งานของมัลติบอร์ดทั้งหมด เครื่องหมาย Prompt เป็นการแจ้งให้ผู้ใช้ว่าต่อจากนี้จะเป็นการทำงานของโปรแกรมมอนิเตอร์นั่นเอง

คีย์ REG

เป็นคีย์ใช้สำหรับเขียนหรืออ่านค่าในรีจิสเตอร์ รีจิสเตอร์ที่กล่าวถึงนี้ไม่ใช่รีจิสเตอร์ที่ชิพผู้ใช้งานอยู่จริง ๆ แต่เป็นหน่วยความจำส่วนหนึ่งที่จัดไว้ให้ทำตัวเสมือนรีจิสเตอร์ เมื่อผู้ใช้ทำการประมวลผลโปรแกรมที่เขียนขึ้น โปรแกรมมอนิเตอร์จะไหลค่าในหน่วยความจำเหล่านี้มาเก็บในรีจิสเตอร์จริง ๆ แล้วจึงจะทำการประมวลผลโปรแกรมของผู้ใช้ต่อไป ให้กดคีย์ต่าง ๆ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์

การแสดงผล

MON

-

B

B

REG

0 0 0 0 I.Y

ใน 4 หลักแรกเป็นตัวเลขและ 2 หลักหลังเป็นตัวอักษร ส่วนหลักที่ 1-2 คือค่าของรีจิสเตอร์ IY บิตสูงคือบิตที่ 8-15 ส่วน 3-4 คือค่าของรีจิสเตอร์ IY บิตต่ำคือบิตที่ 0-7 จุดสว่างที่อักษร I แสดงว่าในขณะที่สามารถเปลี่ยนค่าในรีจิสเตอร์ IY บิตสูงได้ ให้ใส่ค่า \$99 ลงไป

คีย์

การแสดงผล

9

9 0 0 I.Y

9

9 9 0 0 I.Y

INC

9 9 0 0 I Y.

ค่า \$99 อยู่ในรีจิสเตอร์ IY บิตสูงและเมื่อคีย์ INC ถูกกดจุดสว่างจะเลื่อนไปอยู่ที่อักษร Y แสดงว่าสามารถทำการเปลี่ยนค่าในรีจิสเตอร์ IY บิตต่ำได้ เมื่อกด INC ต่อไปอีกก็จะเลื่อนค่าไปรีจิสเตอร์ตัวต่อไปเมื่อครบรอบแล้วจะวนกลับมาอีก รอบของค่ารีจิสเตอร์จะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์

การแสดงผล

PC

XXXXPC.

IY

XXXXIY.

IX

XXXXIX.

ACCA และ ACCB

XXXXAb.

CCR

00XXCr.

SP

XXXXSP.

คีย์ DEC ทำหน้าที่คล้ายกับคีย์ INC แต่จะสวนทางกันคือคีย์ DEC จะเป็นการลดค่า และคีย์ทั้งสองนี้เป็นคีย์ Auto Repeat คือสามารถกดค้างเอาไว้ได้และเพิ่มความเร็วได้ โดยกดคีย์ STEP ควบคู่กันไป ให้ลองเปลี่ยนค่าในรีจิสเตอร์ CCR เป็น ๕AA ดู

คีย์

การแสดงผล

CCR

0 0 0 0 C r.

คีย์	การแสดงผล
2	2
4	2 4
DATA	2 4 0 0 .X X

ในขณะที่อยู่ในฟังก์ชัน DATA ผู้ใช้สามารถเลื่อนไปดูค่าที่แอดเดรสต่อไปด้วยการกดคีย์ INC และทำนองเดียวกันก็จะสามารถดูแอดเดรสก่อนหน้าได้ด้วยคีย์ DEC ทั้งสองคีย์นี้จะเพิ่มและลดค่าแอดเดรสที่ละตำแหน่ง นอกจากนี้ผู้ใช้ยังสามารถกดค้างไว้ได้ ซึ่งจะทำให้แอดเดรสเพิ่มหรือลดค่าอย่างต่อเนื่องและเพิ่มความเร็วได้โดยให้กดคีย์ STEP ควบคู่กันไป เมื่อผู้ใช้ต้องการเขียนข้อมูลลงในหน่วยความจำ สามารถทำได้โดยการกดคีย์ตัวเลข ข้อมูล 2 หลักลงในขณะที่มีลิตบอร์คกำลังแสดงแอดเดรสนั้น ๆ สมมุติค่าต้องการใส่ค่า \$99 และ \$AA ลงที่แอดเดรส \$2400 และ \$24001 จะทำได้ดังนี้

คีย์	การแสดงผล
MON	-
2	2
4	2 4

คีย์	การแสดงผล
MON	-
2	2
3	2 3
0	2 3 0
0	2 3 0 0
RUN	-

เครื่องหมายขีดแสดงการกลับเข้าสู่โปรแกรมมอนิเตอร์ เกิดจากการที่ทาสโปรแกรมของผู้ใช้ใส่คำสั่ง JUMP เข้าหามอนิเตอร์ของมัลติบอร์ดจำเป็นต้องมีเสมอในทุกโปรแกรมที่เขียนลงบนมัลติบอร์ด ในการใช้คีย์ RUN นี้ มัลติบอร์ดจะทำการตรวจค่าใน แอดเดรสแรกที่จะประมวลผลก่อน ถ้ามีค่าเท่ากับ \$00 มัลติบอร์ดจะถือว่าไม่มีโปรแกรมอยู่ในแอดเดรสนั้น ๆ ซึ่งมัลติบอร์ดจะไม่ประมวลผลและส่งเสียงออกทางลำโพง เพื่อบอกให้ผู้ใช้ทราบทั้งนี้เพื่อป้องกันโอกาสผิดพลาดที่จะเกิดขึ้นได้จากการประมวลผล

คีย์ STEP

สำหรับการประมวลผลโปรแกรมที่ผู้ใช้เขียนขึ้นเช่นเดียวกับคีย์ RUN แต่จะกรณีนี้เป็น การประมวลผลทีละ 1 คำสั่ง เพื่อทดสอบการทำงานของโปรแกรม แต่ทั้งนี้ผู้ใช้จะสามารถ ตั้งให้ทำงานได้มากกว่า 1 คำสั่งก็ได้โดยการใช้ฟังก์ชัน NSTEP ซึ่งจะกล่าวถึงในตอนหลัง

ลักษณะการใช้คีย์ STEP จะเหมือนกับคีย์ RUN ทุกประการ เพียงแต่คีย์ STEP จะ

หยุดแสดงแอดเดรสหรือค่ารีจิสเตอร์ PC บนจอแสดงผลทุกคำสั่ง ผู้ใช้จะใช้งานคำสั่งอื่น ๆ ได้ตามต้องการ เมื่อต้องการให้ทำงานต่อก็ทำได้โดยกดคีย์ STEP อีก มัลติบอร์ดจะทำงานต่อจากแอดเดรสเดิม ให้กดคีย์ STEP นี้จะเป็น Auto Repeat เช่นกันคือสามารถกดค้างไว้ได้ มัลติบอร์ดจะประมวลผลอย่างต่อเนื่องกันไป โดยมีการแสดงแอดเดรสที่ประมวลผลบนจอแสดงผล ลักษณะนี้เรียกว่าการประมวลผลแบบ TRACE และเช่นเดียวกันผู้ใช้สามารถเพิ่มความเร็วได้โดยการกดคีย์ INC หรือ DEC ควบคู่กันไป

การใช้งานคีย์ STEP จะมีผลเมื่อแอดเดรสที่ใช้อยู่นอกเหนือจากโปรแกรมมอนิเตอร์ คือ \$A000-\$FFFF เท่านั้น ถ้าอยู่ในโปรแกรมมอนิเตอร์จะทำงานแบบเดียวกับคีย์ RUN

คีย์ FUNC

สำหรับการใช้งานฟังก์ชันต่าง ๆ ของอิทบอร์ด ซึ่งมีทั้งหมด 32 ฟังก์ชัน คีย์ FUNC จะมีสองระดับ โดยเมื่อผู้ใช้กดคีย์ FUNC ดังนี้

คีย์

การแสดงผล

FUNC

FUNC.1

มัลติบอร์ดจะแสดงคำว่า FUNC แล้วจะบอกระดับของฟังก์ชันด้วย ถ้าผู้ใช้กดคีย์ 0-F มัลติบอร์ดจะทำงานตามฟังก์ชันต่าง ๆ ที่พิมพ์ไว้ส่วนล่างของตัวเลข เช่น ถ้าผู้ใช้กดคีย์ 0 ก็ทำงานตามฟังก์ชัน BEEPทันที แต่ถ้าผู้ใช้ต้องการฟังก์ชันระดับสองให้กดคีย์ FUNC อีกทีดังนี้

คีย์

การแสดงผล

FUNC

FUNC.2

ในทำนองเดียวกันผู้ใช้จะใช้ฟังก์ชันตามที่พิมพ์ไว้ที่ส่วนบนของตัวเลข ถ้ากดคีย์ FUNC นี้ต่อไปเรื่อย ๆ มัลติบอร์ดก็จะเปลี่ยนระดับ 1 และ 2 สลับกันไป รายละเอียดของฟังก์ชัน

การใช้คีย์ฟังก์ชันต่าง ๆ ของมัลติบอร์ด คีย์ BREAK

ในบางครั้งโปรแกรมที่ผู้ใช้เขียนขึ้นมามีความยาวมากและยากที่จะทดสอบการทำงานของโปรแกรมทั้งหมด มัลติบอร์ดสามารถตั้งค่า BREAK หรือที่หยุดของโปรแกรมได้โดยให้หยุดระหว่างการประมวลผลโปรแกรมในแอดเดรสใดแอดเดรสหนึ่ง การใช้คีย์นี้มัลติบอร์ดจะถามแอดเดรส BREAK ของโปรแกรม โดยขึ้น -b ให้ใส่แอดเดรสที่ต้องการจะ BREAK ลงไป ถ้าเราใส่ค่าแอดเดรสผิดพลาดก็สามารถลบได้ด้วยการกดคีย์ DEC การกดคีย์ INC เสมือนเป็นการใส่ข้อมูลในการ BREAK ให้กับมัลติบอร์ดรับรู้และก็เช่นเดียวกับในคีย์ DATA ถ้าเราไม่ใส่ตัวเลขมัลติบอร์ดจะใส่ค่า 0 ให้ในหลักที่ไม่ใส่

เมื่อโปรแกรมมาหยุดในแอดเดรสที่เราตั้ง BREAK ไว้จะแสดงค่าของแอดเดรสบนหน่วยแสดงผล ขณะนี้เราก็สามารถตรวจสอบค่ารีจิสเตอร์และหน่วยความจำได้ตามต้องการ และเมื่อเรากดคีย์ RUN อีกทีโปรแกรมจะทำการประมวลผลต่อในแอดเดรสที่แสดงผลอยู่ต่อจนจบโปรแกรม

หมายเหตุ การ BREAK โปรแกรมจะทำได้ในส่วนที่เป็นรอม

คีย์ RELA

เป็นคีย์ RELATIVE CALCULATION ใช้สำหรับการคำนวณค่า RELATIVE ของคำสั่ง BRANCH ต่าง ๆ เช่น BCC, BCS, BNE, BEQ, และ BRA เป็นต้น

การใช้คีย์นี้มัลติบอร์ดจะถามแอดเดรสเริ่มต้นโดยขึ้น -s ที่จอแสดงผลและเมื่อเรากดคีย์ INC มัลติบอร์ดจะถามแอดเดรสปลายทางหรือจุดที่จะกระโดดไป โดยจะขึ้น -f เมื่อกดคีย์ INC อีกครั้งมัลติบอร์ดจะแสดงค่า RELATIVE ที่คำนวณได้พร้อมกับใส่ค่า RELATIVE ลงในหน่วยความจำที่ต่อจากออฟโค้ดของคำสั่ง BRANCH โดยอัตโนมัติ

ค่า RELATIVE มีข้อกำหนดโดยจะต้องอยู่ในค่า -128 ถึง +127 ของช่วงตำแหน่งจากจุดคำสั่ง RELATIVE นั้น ๆ ในกรณีที่เกินค่า RELATIVE มัลติบอร์ดจะตรวจสอบถ้าไม่อยู่ในช่วง -128 ถึง +127 แล้วมัลติบอร์ดจะแสดง ERROR ไปยังหน่วยแสดงผลและใส่ค่าตัวเลข 9FF ลงไปในหน่วยความจำที่หาค่า RELATIVE แทน

คีย์ FIND

เป็นคีย์ฟังก์ชันที่ใช้สำหรับการค้นหาข้อมูลในหน่วยความจำ โดยข้อมูลที่ค้นหาต้องมีขนาดตั้งแต่ 1 ไบต์ไปจนถึง 8 ไบต์ เมื่อใช้ฟังก์ชันนี้มัลติบอร์ดจะแสดงผลดังนี้

เอกสารนี้เป็นเอกสารที่หมายถึง ตำแหน่งแอดเดรสที่จะเริ่มหาข้อมูล เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

F หมายถึง ตำแหน่งสุดท้ายของช่วงที่จะให้หาข้อมูล

L หมายถึง ความยาวของข้อมูลว่ามีจำนวนกี่ไบต์

เมื่อป้อนค่าแต่ละอย่างแล้วกดคีย์ INC เป็นการแสดงให้เห็นให้มัลติบอร์ตรับค่านั้นเพื่อนำไปประมวลผล หลังจากป้อนความยาวข้อมูลและกดคีย์ INC มัลติบอร์ตจะให้ป้อนข้อมูลที่มีขนาดความยาวตั้งแต่ 1-8 ไบต์ ตามความยาวของข้อมูลที่ตั้งไว้ เมื่อป้อนเสร็จแต่ละไบต์ก็ให้กดคีย์ INC เมื่อป้อนไบต์ถัดไป

หลังจากป้อนค่าเสร็จแล้วมัลติบอร์ตจะทำการค้นหาข้อมูล หลังจากนั้นแสดงแอดเดรสที่พบข้อมูล ถ้ากด INC อีกก็จะทำการหาข้อมูลชุดนี้อีก ถ้าไม่มีข้อมูลอยู่จะแสดงเครื่องหมายขีดขึ้นมาที่จอแสดงผล

คีย์ FILL

ใช้สำหรับการใส่ข้อมูลค่าหนึ่งในหน่วยความจำช่วงหนึ่ง มัลติบอร์ตให้ป้อนแอดเดรสเริ่มต้นของข้อมูล, แอดเดรสสุดท้ายของข้อมูล และข้อมูลที่จะป้อนหลังจากการกดคีย์ INC มัลติบอร์ตจะทำการใส่ข้อมูลลงในช่วงหน่วยความจำที่เราต้องการ จากนั้นมีเสียงสัญญาณขึ้นเครื่องหมายขีด แสดงการป้อนข้อมูลลงหน่วยความจำเรียบร้อยแล้ว

คีย์ COMB

เป็นคีย์ที่ใช้สำหรับการเปรียบเทียบข้อมูลช่วงหนึ่งกับอีกช่วงหนึ่งในหน่วยความจำของมัลติบอร์ต โดยการใช้งานจะต้องกำหนดค่าดังนี้

START	ใส่ค่าแอดเดรสเริ่มต้นของช่วง
FINAL	ใส่ค่าแอดเดรสสุดท้ายของช่วง
DESTINATION	ใส่ค่าแอดเดรสเริ่มต้นของช่วงที่จะทำการเปรียบเทียบ

ในขณะที่ทำการเปรียบเทียบแล้วข้อมูลไม่เท่ากันมัลติบอร์ตจะแสดงแอดเดรสของจุดนั้นคือแอดเดรสของค่าแอดเดรสเริ่มต้นช่วงเปรียบเทียบที่ไม่เท่า เมื่อกดคีย์เรา INC อีกก็จะเปรียบเทียบต่อไปจนครบโดย ถ้าข้อมูลเท่ากันโดยตลอดก็จะแสดงเครื่องหมายขีดขึ้นมา

คีย์ COMH

เป็นคีย์ที่ใช้สำหรับเปรียบเทียบข้อมูลช่วงหนึ่งกับค่าของข้อมูล 1 ไบต์ การใช้งานจะกำหนดดังนี้

START	ใส่ค่าแอดเดรสเริ่มต้นของช่วง
FINAL	ใส่ค่าแอดเดรสสุดท้ายของช่วง
HEX	ใส่ค่าข้อมูลที่จะเปรียบเทียบ

การใช้งานก็ลักษณะเดียวกับในคีย์ฟังก์ชัน COMB โดยหน่วยแสดงผลแอดเดรสที่มีค่าไม่ตรงกับค่า เลขฐานสิบหก ที่เราใส่กำหนดไว้

คีย์ MOVE

ใช้สำหรับการคัดลอกหรือย้ายข้อมูลช่วงหนึ่งไปยังข้อมูลอีกช่วงหนึ่ง การใช้งานก็เป็นเช่นเดียวกับคีย์ COMB โดย

START	ใส่ค่าแอดเดรสเริ่มต้นของช่วง
FINAL	ใส่ค่าแอดเดรสสุดท้ายของช่วง
DESTINATION	ใส่ค่าแอดเดรสเริ่มต้นของช่วงที่ต้องการนำข้อมูลมาใส่

คีย์ CHAN

เป็นคีย์ที่ใช้สำหรับการเปลี่ยนค่าข้อมูล เราสามารถกำหนดจำนวนไบต์ในการเปลี่ยนได้ 8 ไบต์ ในช่วงหน่วยความจำช่วงหนึ่งให้เป็นค่าใหม่ตามจำนวน ไบต์ ที่จะเปลี่ยนมักจะใช้ในการเปลี่ยนค่ามากกว่า 1 ครั้ง การใช้ให้กำหนดค่าต่อไปนี้

START	ใส่ค่าแอดเดรสเริ่มต้นของช่วง
FINAL	ใส่ค่าแอดเดรสสุดท้ายของช่วง
LENGTH	ความยาวของข้อมูลว่ามีจำนวนกี่ไบต์

คีย์ NO.STEP

เป็นคีย์ที่ใช้สำหรับตั้งจำนวนครั้งของการทำ SINGLE STEP คือตั้งจำนวนคำสั่งที่จะประมวลผลจากการกดคีย์ STEP ว่าจะให้กดคีย์ STEP หนึ่งครั้งจะเท่ากับการประมวลผลกี่คำสั่ง การใช้งานจะกำหนดค่าจำนวนการ STEP ตามที่ต้องการ คือ 1 ถึง 255 คำ

หมายเหตุ เมื่อเปิดเครื่องมัลติบอร์ดจะตั้ง NO.STEP ให้มีค่าเท่ากับ 1

คีย์ BEEP

ใช้สำหรับการเปิดหรือปิดเสียงสัญญาณของการกดคีย์ โดยเมื่อกดคีย์จะมีเสียงสัญญาณดังออกที่ลำโพง ผู้ใช้สามารถเปิดหรือปิดเสียงได้ด้วยการคีย์ฟังก์ชันนี้ โดยการกดคีย์ BEEP มัลติบอร์ด จะแสดง OFF ที่หน่วยแสดงผลและถ้าหากต้องการเสียงอีก ก็กดคีย์ BEEP อีกที่ มัลติบอร์ดก็จะ ON หมายถึงได้เปิดเสียงแล้ว

คีย์ USER และ SET

เป็นคีย์ฟังก์ชันที่ต้องใช้ควบคู่กันในกรณีที่ผู้เขียนโปรแกรมขึ้นหลายโปรแกรม เมื่อเราสั่งให้ทำการประมวลผลแต่ละโปรแกรม จะต้องตั้งแอดเดรสในการประมวลผลอยู่เสมอ ๆ ทำให้ไม่สะดวกและเสียเวลา มัลติบอร์ดจึงมีฟังก์ชันพิเศษขึ้นมาโดยการตั้งค่าแอดเดรสที่จะประมวลผลของแต่ละโปรแกรมในฟังก์ชัน SET และประมวลผลโดยกดปุ่มฟังก์ชัน USER

SET หมายถึงคีย์ตัวเลข 0 ถึง F ที่เราจะนำมาตั้งเป็นคีย์ USER

A หมายถึงแอดเดรสที่ต้องการให้ประมวลผล

การประมวลผลแอดเดรสที่เราเซตไว้ ทำได้โดยกดคีย์ USER และกดคีย์ตัวเลขที่ตั้งไว้ให้ประมวลผลในแอดเดรสนั้น

คีย์ C.BREAK

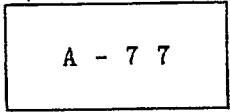
ใช้สำหรับเคลียร์หรือลบค่า BREAK ที่ตั้งไว้ โดยจะเคลียร์เป็นค่าศูนย์ทั้งหมด

คีย์ SEGM

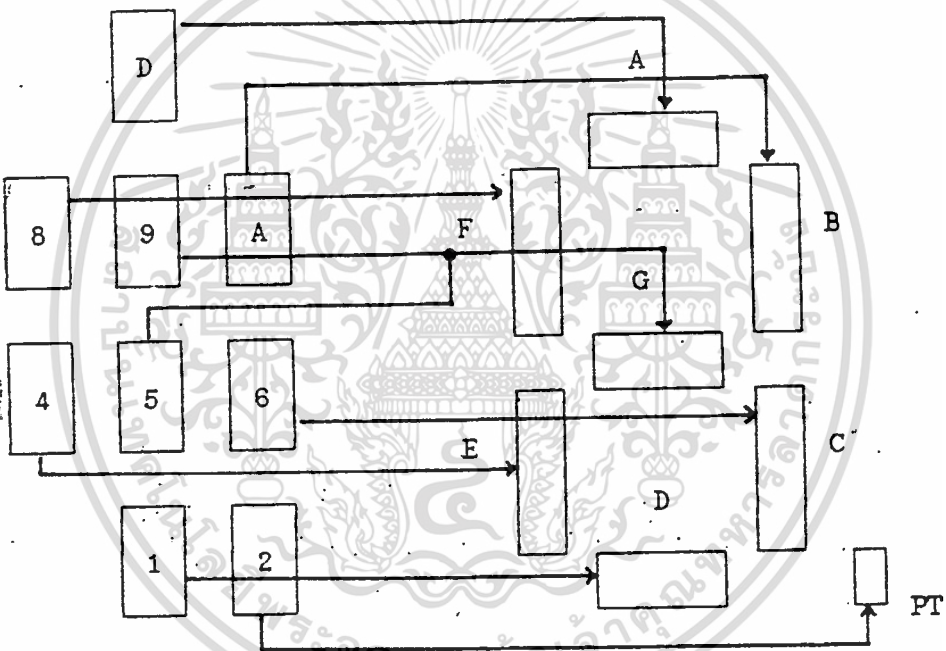
ใช้สำหรับการออกแบบตัวอักษรต่าง ๆ บนแอลอีดี 7 เซกเมนต์ ปกติถ้าผู้ใช้ต้องการสร้างอักษรบนจอแสดงผล ผู้ใช้ต้องร่างขีดตามต้องการแล้วจึงแปลงออกมาเป็นเลขฐาน 16 ซึ่งค่อนข้างเสียเวลา ในฟังก์ชันนี้ก็จะช่วยออกแบบอักษรนี้โดยเฉพาะ เมื่อเร้ากใช้ฟังก์ชันนี้ จะปรากฏค่าบนจอแสดงผลดังนี้

ซึ่งเมื่อกดครั้งแรกจะทำให้เกิดขีดนั้น ๆ บนหลักแรกของจอแสดงผลพร้อมทั้งจะแสดงเอกสารนี้เป็น เลขฐานสิบหกตามอักษรนั้น เมื่อกดซ้ำอีกจะทำให้ขีดนั้นหายไปด้วยวิธีการนี้จะทำให้ออกแบบไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษรได้ตามต้องการ เช่น กดให้แอลอัติ 7 เซกเมนต์แสดงตัวอักษร A ได้รหัสเป็น 77 โดยจะแสดงที่ จอแสดงผล ดังนี้



และขณะนี้ผู้ใช้สามารถกำหนดขีดต่าง ๆ ได้ตามต้องการ ตามคีย์ต่อไปนี้



คีย์ TESTR

ใช้สำหรับทดสอบแรมตามแอดเดรสที่เรากำหนดไว้ ผลการทดสอบแสดงที่จอแสดงผลเป็นช่วง ๆ ละ 256 ไบต์ ถ้าแรมเกิดบกพร่องก็จะแสดงช่วงที่ผิดนั้น แล้วจะรอการกดคีย์ ถ้าคีย์ที่กดเป็นคีย์ INC จะทำการตรวจสอบต่อไป แต่ถ้าเป็นคีย์ Instant อื่น จะกระโดดไปทำหน้าที่ตามคีย์นั้น เมื่อตรวจสอบแรมถูกต้องตามที่ตั้งไว้แอดเดรสที่ทำการตรวจสอบนั้น จะมี DATA = 00H เข้าไปแทน

คีย์ IN

ใช้สำหรับอ่านข้อมูลจากอุปกรณ์อินพุตหรือหน่วยความจำ คือ จะรับข้อมูลโดยการอ่านจากพอร์ตหรือหน่วยความจำเข้ามา ซึ่งสามารถกำหนดหมายเลขพอร์ตหรือหน่วยความจำที่เอกสารนี้เป็นเอกสารที่ลงวันไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะอ่านเข้ามาได้ โดยแสดงค่าข้อมูลที่อ่านได้ด้วยตัวเลข 2 หลัก เราสามารถกำหนดได้ว่า จะให้แลตซ์หรือไม่ คือให้ตัวเลข 2 ตัวท้ายแสดงค้างหรือไม่ โดยคีย์ INC คือ YES และ คีย์ DEC คือ NO ถ้าข้อมูลในพอร์ตหรือหน่วยความจำที่อ่านเข้ามามีค่าเปลี่ยนแปลง ตัวเลข ที่ 2 ตัวท้ายก็จะเปลี่ยนด้วยและสามารถอ่านข้อมูลในพอร์ตนั้นเป็นชุด ๆ ได้เมื่อพอร์ตอินพุต ส่งข้อมูลเข้าเรื่อย ๆ ถ้าไม่แลตซ์จะแสดงค่าอ่านจากพอร์ตอินพุตชั่วขณะ แล้วจะกลับไปถาม เบอร์ของพอร์ตหรือหน่วยความจำอีก เพื่อที่จะอ่านข้อมูลในพอร์ตหรือหน่วยความจำอื่นต่อไป

คีย์ OUT

ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์เอาต์พุตหรือหน่วยความจำ สามารถกำหนดหมายเลข ของพอร์ตหรือหน่วยความจำที่จะส่งข้อมูลออกไปได้

คีย์ POFF

ใช้สำหรับปิดเครื่องบนมัลติบอร์ดเวอร์ชัน 1 โดยคีย์นี้จะมีการทำงานถึง 3 ขั้นตอน คือ

1. ทำการเคลียร์ POWER CODE ทำให้เหมือนกับการเปิดเครื่องครั้งแรก และจะ HALT ระบบก่อนที่จะดึงแหล่งจ่ายไฟออกไป จุดประสงค์ของการ HALT ระบบก็เพื่อป้องกัน ข้อมูลสูญหายเนื่องจากบนมัลติบอร์ดไม่มีการป้องกันการเขียนทับ

2. ทำการสำรวจแรมไว้ และตรวจสอบข้อมูลว่ายังถูกต้องหรือไม่ด้วยการตรวจสอบ ผลรวมทั้งหมด โดยจะต้องมีแบตเตอรี่สำรองด้วย ถ้าการตรวจสอบผลรวมถูกต้องลำโพงจะ ให้กำเนิดเสียงสูง แต่ถ้าข้อมูลนั้นไม่เหมือนเดิมแสดงว่าการสำรองเกิดบกพร่อง ตรวจสอบ ผลรวมจะผิด ลำโพงจะให้กำเนิดเสียงต่ำ

3. ให้ทำการป้องกันการเขียนข้อมูลจากผู้อื่นได้ โดยจะให้มีการป้อนตัวเลข 4 หลัก ถ้ามีการเปิดมัลติบอร์ดครั้งใหม่ มัลติบอร์ดจะให้ตัวเลข 4 หลักนี้ ถ้าคีย์ไม่ถูกต้องแล้ว มัลติบอร์ดจะไม่ยอมให้ผ่านไปเขียนข้อมูลได้

แต่ถ้าเราไม่ทำการป้องกันการเขียนข้อมูลจากผู้อื่น หลังจากกดคีย์ POFF แล้วให้กด ที่คีย์ INC ได้เลยเมื่อเราตั้งรหัสของเราแล้ว พอเปิดมัลติบอร์ดใหม่มัลติบอร์ดจะถามรหัสนี้ เมื่อเราใส่ถูกต้องค่าตัวเลข 4 หลักที่เราใส่จะถูกเคลียร์ ดังนั้นเมื่อต้องการปิดมัลติบอร์ดก็ ไม่ควรดึงแหล่งจ่ายไฟออกโดยตรง ควรผ่าน POFF (POWER OFF) ก่อน เพื่อเป็นการบันทึก ข้อมูลที่เขียนขึ้น

คีย์ CRAM

ใช้สำหรับเคลียร์หรือลบค่าแรมในมัลติบอร์ด ให้มีค่าเป็นศูนย์หมดโดยในการใช้งาน จะแสดงคำว่า READY เพื่อถามความแน่ใจอีกครั้งหนึ่ง ถ้าต้องการเคลียร์จริงก็กดคีย์ INC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาดูเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

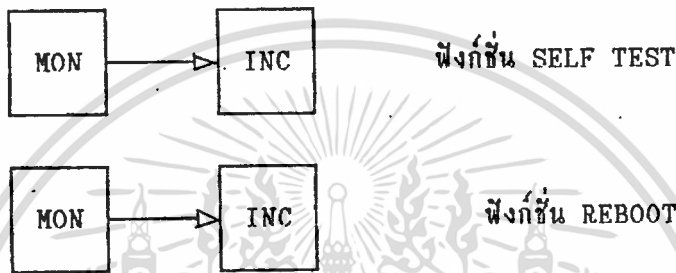
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์ CREG

ใช้สำหรับเคลียร์ค่าในรีจิสเตอร์ของผู้ใช้ทั้งหมดให้ค่าเป็น 00 ยกเว้น PC และ SP ที่จะไม่ถูกเคลียร์ค่า

การใช้ฟังก์ชันพิเศษ

มัลติบอร์คมีฟังก์ชันพิเศษอีก 2 ฟังก์ชัน โดยจะใช้คีย์ 2 คีย์ควบคู่กันไปดังนี้



วิธีการกดคีย์จะต้องกดคีย์ MON ค้างเอาไว้ก่อน จากนั้นให้กดคีย์ที่ต้องการตามและ ค้างไว้แล้วจึงปล่อยคีย์ MON และปล่อยคีย์ที่เหลือตามลำดับ รายละเอียดของฟังก์ชันทั้งสอง จะอธิบายได้ดังนี้

SELF TEST

ฟังก์ชันนี้ใช้สำหรับการทดสอบตัวเองของบอร์ดว่าอยู่ในสภาพสมบูรณ์หรือไม่ เมื่อเรา ใช้ฟังก์ชันนี้มัลติบอร์คจะแสดงที่จอแสดงผลดังนี้

YY MM LL

ซึ่งจะแสดงข้อมูลต่าง ๆ ในการผลิต YY ก็คือปี ค.ศ. (19YY) MM คือเลขเดือนที่ ผลิตและ LL คือรุ่นของการผลิต จากนั้นชั่วขณะหนึ่งมัลติบอร์คจะแสดงที่จอแสดงผลดังนี้

XX

ซึ่งจะเป็นการตรวจสอบคีย์บอร์ดทั้ง 23 คีย์ (ยกเว้น MON) โดยจะแสดงหมายเลขรหัสของคีย์ (รายละเอียดอยู่ในหัวข้อ DATA FORMAT) และให้ผู้ใช้กดคีย์ตามรหัสนั้น ๆ ถ้ามีเหตุผิดปกติมีลติบอร์คก็จะแสดง ERROR ขึ้นที่จอแสดงผลแต่ถ้าปกติครบทุกตัว มีลติบอร์คจะทำขบวนการต่อไป

8.8.8.8.8.8.

จะเป็นการตรวจสอบสภาพของตัวแสดงผลทั้ง 6 หลัก รวมทั้ง LED แสดงแฟล็กด้วย ถ้าแอลอีดีทุกตัวเรียบร้อยดีให้กดคีย์ใด ๆ ก็ได้เพื่อทำขบวนการต่อไป

2300

TEST RAM

9000

เป็นการตรวจสอบหน่วยความจำแรมตั้งแต่แอดเดรส \$2300 จนถึง \$9000 โดยจะทำการตรวจสอบทีละช่วง ๆ ละ 256 ไบต์ ถ้ามีช่วงใดบกพร่องก็จะแสดงค่าแอดเดรสนั้นค้างเอาไว้และจะทำงานต่อได้ด้วยคีย์ INC แต่ถ้าทุกอย่างเรียบร้อยดีหน้าจอแสดงผลจะดับเนื่องจากมีลติบอร์คเข้าสู่สภาวะ HALT และเป็นการสิ้นสุดการทดสอบตัวเองด้วย ถ้าผู้ใช้ต้องการใช้งานตามปกติก็ให้กด MON อีกครั้ง

ข้อควรระวัง การทดสอบตัวเองในส่วนที่ตารางทดสอบแรม มีลติบอร์คจะล้างข้อมูลในแรมทั้งหมดด้วย เพราะฉะนั้นต้องแน่ใจว่าข้อมูลที่อยู่ในแรมนั้นไม่ต้องการใช้แล้ว

REBOOT

ฟังก์ชันนี้ทำหน้าที่เสมือนกับการปิดและเปิดมีลติบอร์คใหม่ โดยที่ไม่จำเป็นต้องปิดและเปิดมีลติบอร์คจริงทั้งนี้เพื่อให้ผู้ใช้ประหยัด เวลาและขณะเดียวกันทำให้ข้อมูลในแรมยังคงจำอยู่ได้ด้วย

ฟังก์ชันคีย์โปรแกรมตัวอย่าง

ฟังก์ชัน CLCD

ทำหน้าที่ เคลียร์แอลซีดี

คำอธิบาย ใช้เคลียร์หน้าจอแอลซีดีและเคอร์เซอร์จะไปอยู่ที่คอลัมน์ที่ 1 กระพริบรอรับตัวอักษร

ฟังก์ชัน TLCD

ทำหน้าที่ ทดสอบแอลซีดี 16 ตัวอักษร 1 แถว

คำอธิบาย การทดสอบแอลซีดีจะใช้การส่ง ASCII ตั้งแต่ 20-7F โดยการส่ง ASCII แต่ละตัวจะเขียนลงบนแอลซีดีทั้ง 16 ตัวอักษร หลังจากทดสอบเสร็จแล้วจะทำการเคลียร์แอลซีดี

ฟังก์ชัน VM

ทำหน้าที่ แปลงสัญญาณดิจิทัลเป็นอนาล็อก 1 แชนแนล

คำอธิบาย จะนำสัญญาณอนาล็อกจาก AN0 มาแปลงเป็นสัญญาณดิจิทัล 4 หลักบนแอลซีดี แต่ระดับแรงดันของสัญญาณอนาล็อกนี้ต้องเป็นแรงดันไฟตรงที่ไม่เกิน 5 โวลท์

ฟังก์ชัน ADC 8

ทำหน้าที่ แปลงสัญญาณดิจิทัลเป็นอนาล็อก 8 แชนแนล

คำอธิบาย จะนำสัญญาณอนาล็อก 8 แชนแนลคือ AN0 จนถึง AN7 มาทำการแปลงเป็นสัญญาณดิจิทัลขนาด 4 หลักของในแต่ละแชนแนลไปแสดงผลบนจอมอนิเตอร์ของคอมพิวเตอร์ แต่ต้องมีการเรียกใช้ PROCMM PLUS ก่อน

ฟังก์ชัน TXRX

ทำหน้าที่ รับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับมัลติบอร์ด

คำอธิบาย ฟังก์ชันนี้จะแสดงการรับ-ส่งระหว่างมัลติบอร์ดกับเครื่องคอมพิวเตอร์ ถ้ากดคีย์บนคีย์บอร์ดของเครื่องคอมพิวเตอร์ มัลติบอร์ดก็จะรับเอาค่า ASCII ของคีย์ที่เรากดเข้ามาและส่งค่า ASCII นั้นกลับไปให้เครื่องคอมพิวเตอร์ โดยแสดงผลบนจอมอนิเตอร์ของเครื่องคอมพิวเตอร์

ฟังก์ชัน TIMER 1

ทำหน้าที่ วัดคาบเวลาของพัลส์

คำอธิบาย การวัดคาบเวลาของพัลส์ โดยการนำอินพุตแคปเจอร์ 1 มาประยุกต์ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้จัดทำเห็นใบใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจะอ่านขอบของสัญญาณสี่เหลี่ยมลูกแรกมาลบกับขอบของสัญญาณรูปสี่เหลี่ยมลูกที่สองของการแสดงผล 5 หลักรบนจอแอลซีดี

ฟังก์ชัน TIMER 2

ทำหน้าที่ กำเนิดสัญญาณสี่เหลี่ยม

คำอธิบาย กำเนิดสัญญาณสี่เหลี่ยมโดยการใช้อะไหล่ที่ 2 เอาท์พุทจะออกทางพอร์ต OC 2

ฟังก์ชัน TRAN

ทำหน้าที่ ส่งข้อมูลในบัฟเฟอร์ให้เครื่องคอมพิวเตอร์

คำอธิบาย จะทำหน้าที่ส่งข้อมูลเป็นชุดให้เครื่องคอมพิวเตอร์ และหยุดส่งเมื่อเจอรหัสจบคือ \$04 การใช้งานมีลิตบอร์คจะขึ้น XXXX-S เป็นการถามตำแหน่งที่เริ่มต้นของข้อมูลก่อนใช้งานฟังก์ชันนี้จะต้องเรียก PROCOMM PLUS ก่อน

ฟังก์ชัน REV

ทำหน้าที่ รับข้อมูลจากเครื่องคอมพิวเตอร์

คำอธิบาย ขณะที่เราอยู่ใน PROCOMM PLUS เมื่อคีย์บอร์ด ASCII ของคีย์บอร์ดจะมาปรากฏบนจอแอลซีดีของมีลิตบอร์ค เมื่อรับครบ 16 ตัวอักษรจะเคลียร์ค่าของแอลซีดีใหม่ เคอร์เซอร์จะไปกระพริบรับค่า ASCII ที่คอลัมน์ 1

ฟังก์ชัน TSEG

ทำหน้าที่ ทดสอบ 7 เซกเมนต์และแอลอีดีแฟล็ก

คำอธิบาย ทำการทดสอบ 7 เซกเมนต์และแอลอีดีแฟล็ก โดยจะส่งค่า \$FF ที่เป็นข้อมูลออกไปให้แอลอีดีทุกดวงติดและจะมีการเลื่อนหลักของจอแสดงผลไปเรื่อย ๆ เมื่อใช้ฟังก์ชันนี้จะให้เราเลือกการเลื่อนที่ของหลักจอแสดงผล ถ้ากดคีย์ INC จะปรากฏจอแสดงผลหลักแรกติดทางด้านซ้ายมือของจอแสดงผล แล้วก็ค่อย ๆ เลื่อนมาติดหลักถัดไปจนถึงแอลอีดีแฟล็กแล้ววนกลับไปยังจอแสดงผลหลักแรกอีกครั้งที่เรากดคีย์ DEC จะเลื่อนจากด้านขวา คือ แอลอีดีแฟล็กไปยังจอแสดงผลหลักแรก การเลือกทั้งสองคีย์จะมีผลให้เกิดการวนลูปตลอดจนการกดคีย์ MON

ฟังก์ชัน TPRN

ทำหน้าที่ ทดสอบการติดต่อกับเครื่องพิมพ์

คำอธิบาย เมื่อเรียกใช้ฟังก์ชันนี้จะให้ใส่ค่าเริ่มต้นโดยมีรูปแบบดังนี้ XXXX-S เมื่อใส่ค่าเอกสารนี้เป็นเอกสารที่ส่งวนเวียนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มต้นโปรแกรมจะนำข้อมูล ASCII ตั้งแต่ตำแหน่งนั้นส่งออกไปที่เครื่องพิมพ์ จนกว่าจะเจอรหัสจบคือ \$04 จึงกลับสู่มอนิเตอร์ของมัลติบอร์ด

ฟังกซ์ HELP

ทำหน้าที่ แสดงแอดเดรสของระบบไบท์เฟอ์ต่าง ๆ

คำอธิบาย การใช้งานมัลติบอร์ดนั้นจะมีหน่วยความจำบางส่วนสงวนไว้สำหรับการทำงานบนมอนิเตอร์โปรแกรม การเข้าไปแก้ไขข้อมูลในหน่วยความจำที่กล่าวจะมีผลต่อการทำงานของมัลติบอร์ด เพื่อจะได้ศึกษาการทำงานอย่างลึกซึ้งมากขึ้น จึงนำแอดเดรสเหล่านั้นมาแสดงไว้บน HELP ของบอร์ด เมื่อเราใช้ฟังกซ์นี้ จะแสดงแอดเดรสของระบบบนจอแอลซีดี แล้วให้เลือกว่าจะดูค่าอื่นต่อหรือไม่ ถ้ากดคีย์ INC จะเป็นการตอบ YES และถ้า DEC จะเป็นการตอบ NO

การทำงานในโหมด 2 บนเครื่องคอมพิวเตอร์

จะเป็นการทำงานใน DEBUGGER 68HC11 การป้อนโปรแกรมเป็นภาษาแอสเซมบลีตามข้อกำหนดของ DEBUGGER โดยป้อนโปรแกรมและควบคุมการประมวลผลโปรแกรมผ่านทางคีย์บอร์ดของเครื่องคอมพิวเตอร์ การใช้งานในส่วนนี้จะต้องมีการเตรียมพร้อมสำหรับการสื่อสารระหว่างมัลติบอร์ดกับเครื่องคอมพิวเตอร์ โดยการเรียกใช้โปรแกรม PROCOMM ที่จะทำการสื่อสารข้อมูลกัน

ข้อกำหนดต่าง ๆ ในการป้อนโปรแกรม

1. ในการป้อนคำสั่งผ่านเครื่องคอมพิวเตอร์นั้น แต่ละคำสั่งที่กระทำจะต้องคีย์ด้วยคีย์บอร์ดให้ถูกต้องตามรูปแบบที่กำหนดและกด Enter เสมอหลังจากจบคำสั่งแล้ว
2. ผู้ใช้สามารถใช้อักษรตัวใหญ่หรือตัวเล็กได้ตามต้องการ
3. จำนวนตัวเลขต่าง ๆ ที่ป้อนเข้านั้นจะถือว่าเป็นเลขฐาน 16 โดยไม่จำเป็นต้องใส่ # นำหน้า

4. ในกรณีที่ป้อนคำสั่งไม่ถูกรูปแบบเครื่องจะแสดงข้อความให้ทราบ เช่น

What ?

Command ?

Bad argument

5. ในกรณีที่การทำงานอยู่ในโหมดของการแอสเซมเบลอร์ทีละคำสั่ง การกำหนดค่าให้ดิบรีจิสเตอร์แบบกำหนดแอดเดรสโดยตรงจะต้องใส่ # นำหน้าข้อมูลนั้น

6. การกระทำต่าง ๆ ผ่านคีย์บอร์ดและจอภาพของเครื่องคอมพิวเตอร์นี้ เป็นส่วนที่

กระทำบนมัลติบอร์ดและเป็นสิ่งที่กระทำในขณะนั้นจริง ๆ (real time)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. การกดคีย์รีเซ็ตจะเป็นการเคลียร์ ค่าต่าง ๆ ใหม่หมด

8. การกระทำกับข้อมูลที่ต้องเป็นแรม หากมีการกระทำในพื้นที่ที่เป็นแรมที่ถูกป้องกันไว้จะมีข้อความบอกว่า ROM

คำสั่งการใช้งาน

คำสั่ง ?

ทำหน้าที่ แสดงคำสั่งที่มีใช้ใน ET-DEBUGGER 68HC11

รูปแบบ ?

คำอธิบาย แสดงคำสั่งที่มีใช้ใน ET-DEBUGGER 68HC11 พร้อมทั้งรูปแบบที่กำหนดและความหมายของคำสั่ง เพื่อให้เกิดความสะดวกในการใช้งานมากขึ้น

ตัวอย่าง

>?

A [<addr>] ASCII table/Assemble
 [=] Same addr [+] Next addr
 [-] Prev addr [CR] Next opcode
 [CTLA,.] Quit

B [-] [<addr>] breakpoint.

C <addr1><addr2> [<addr3>] Check ram/Compare block

D [<addr1> [<addr2>]] Memory dump

E [<addr>] Memory modify
 [=] Same addr [space] next addr
 [-] Prev addr [CR] Quit
 <addr>0 Compute relative offset

ET [<addr>] Edit Text

F <addr1><addr2> [<data>] Block fill memory

G <addr> Execute program

I <port> [<L>] In port latch

L [<offset>] Down load S-records

M <addr1><addr2><dest> Block move

O <port><data> Out port

P [<addr1><addr2>] Proceed/Up load

Q Quit

R [P,Y,X,A,B,C,S] Register modify

-----please any key-----

REG	Dump I/O Register
S	Test board
T [<n>]	Trace n instructions
U [<addr1>[<addr2>]]	disassemble
V [<offset>]	Verify S-records
VT	Show Interrupt Vector
BULK	Erase EEPROM
TOOLS	Tools for input capture
[CR]	Repeat last command
>	

คำสั่ง

A

ทำหน้าที่

ASCII table และ one line assemble

รูปแบบ

A ASCII table

A addr one line assemble

คำอธิบาย

ถ้าใช้คำสั่ง A อย่างเดียวจะแสดงรหัส ASCII

ถ้าใช้คำสั่ง A ตามด้วยแอดเดรสจะเป็นการแอสเซมบลีที่ละคำสั่ง ซึ่งในการเขียนแอสเซมบลีโดยใช้คำสั่งนี้จะไม่สามารถกำหนดค่าหรืออ้าง lable ต่าง ๆ ที่เป็นตัวแปรได้ ดังนั้นการใช้คำสั่งเกี่ยวกับการกระโดดต่าง ๆ ต้องกำหนดเป็นแอดเดรสที่จะกระโดดไป การใส่ข้อมูลที่เป็นตัวเลขฐาน 16 เราสามารถใส่ได้โดยตรงโดยไม่ต้องมี \$ นำหน้า ในกรณีที่ป้อนแอสเซมบลีผิดรูปแบบก็จะแสดงผล ERROR ให้เราทราบ

คำสั่งย่อย

= รวบรวมข้อมูลที่แอดเดรสเดิม

- ลดค่าแอดเดรสลง 1

+ เพิ่มค่าแอดเดรสขึ้น 1

Enter, Ctrl A หรือ . จะเป็นการออกจากคำสั่งนี้

ตัวอย่าง

>A 2200

2200 : LDAA #41 <Enter>

2202 : LDAB #00 <Enter>

2204 : JSR FF00 <Enter>

เอกสารนี้เป็นเอกสารที่ JSR FF00 ช่างงานเพื่อการศึกษานี้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2207 : LDAB #24 <Enter>

2209 : JSR FF00 <Enter>

2204 : .

>

>A

HEX	0	1	2	3	4	5	6	7
0	NUL	DLE	SPACE	0	@	P	'	P
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(8	H	X	h	x
9	HT	EM)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[k	{
C	FF	FS	'	<	L	\	l	
D	CR	GS	-	=	M]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

คำสั่ง	B	
ทำหน้าที่	ตั้งค่าแอดเดรสที่ทำการ BREAK โปรแกรมเพื่อดูค่ารีจิสเตอร์ต่าง ๆ ซึ่งจะมีรูปแบบต่าง ๆ ดังนี้	
รูปแบบ	B	แสดงค่า BREAK
	B address	ตั้งค่าแอดเดรสที่จะทำการ BREAK
	B -address	เคลียร์แอดเดรสที่จะทำการ BREAK เครื่องหมาย - และแอดเดรสจะต้องเขียนติดกัน
	B -	เคลียร์แอดเดรสที่ทำการ BREAK ทั้งหมด

คำอธิบาย สามารถตั้งค่าแอดเดรสที่ทำการ BREAK ได้สูงสุด 4 ค่า ใส่ค่ามากกว่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มัลติบอร์ดจะแสดงคำว่า Full และแสดงค่า BREAK เก้าให้ดูเมื่อครบเซต
ค่า BREAK ที่ตั้งไว้ทั้งหมดจะถูก เคลียร์

ข้อจำกัดในการ BREAK โปรแกรม

- ถ้าต้องการ BREAK โปรแกรมในส่วนที่เป็นโปรแกรมวนลูปต้องตั้ง BREAK อย่างน้อย 2 ค่า
- ไม่สามารถ BREAK โปรแกรมตรงตำแหน่งที่เริ่มประมวลผลโปรแกรมได้ เช่น ตั้ง BREAK ไว้ที่ 2200 แล้วประมวลผลโปรแกรมที่แอดเดรส 2200 ด้วยก็จะไม่สามารถ BREAK โปรแกรมในตำแหน่งที่ 2200 ได้
- ถ้าโปรแกรมมีการใช้คำสั่ง SWI และมีการตั้ง BREAK ในโปรแกรมนั้น จะต้องทำการใส่คำสั่ง JUMP ไปยังแอดเดรสที่จะทำโปรแกรมย่อยของ SWI ก่อนที่จะทำการประมวลผลโปรแกรม

ตัวอย่าง

- >B 2200 2204 เป็นการตั้งค่า BREAK ที่แอดเดรส \$2200 และ \$2204
- >B -2200 2206 เคลียร์ค่า BREAK ที่แอดเดรส \$2200 และตั้ง BREAK ใหม่ที่
แอดเดรส \$2206
- >B 2207 -2206 ตั้ง BREAK ที่แอดเดรส \$2207 และทำการเคลียร์ BREAK ที่
แอดเดรส \$2206
- >B -2200 เคลียร์ค่า BREAK ทั้งหมดแล้วตั้ง BREAK ที่แอดเดรส \$2200
- >

คำสั่ง

ทำหน้าที่

C ตรวจสอบแรมและเปรียบเทียบข้อมูลเป็นช่วง

รูปแบบ

C addr1 addr2 ตรวจสอบแรม

C addr1 addr2 addr3 เปรียบเทียบข้อมูลเป็นช่วง

คำอธิบาย

ในกรณีที่ทำการตรวจสอบแรมตามแอดเดรสที่กำหนดไว้ ถ้าพบว่าแอดเดรสใดเขียนข้อมูลลงไปแล้วอ่านข้อมูลนั้นกลับขึ้นมาไม่เท่ากันก็จะแสดงแอดเดรสนั้นให้ทราบโดยที่

addr1 = แอดเดรสเริ่มต้นที่จะทำการตรวจสอบ

addr2 = แอดเดรสสุดท้ายที่จะทำการตรวจสอบ

ถ้าข้อมูลไม่ตรงกันในกรณีที่ทำการเปรียบเทียบข้อมูลเป็นช่วง จะแสดง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งของข้อมูลและข้อมูลนั้นทั้งต้นทางและปลายทางให้ทราบโดย

addr1 = แอดเดรสเริ่มต้นของช่วงที่ 1

addr2 = แอดเดรสสุดท้ายของช่วงที่ 1

addr3 = แอดเดรสเริ่มต้นของช่วงที่ 2

ตัวอย่าง

```
>C 2200 2210 3000
```

```
2202: C6 3002: 00
```

```
2204: BD 3304: 00
```

```
2205: FF 3005: 00
```

```
2206: 00 3006: 04
```

```
2207: 7E 3007: 00
```

```
2208: 22 3008: 00
```

```
2210: 00 3010: 80
```

```
>C 2000 3FFF
```

```
-ram ok
```

```
>
```

คำสั่ง

D

ทำหน้าที่

แสดงข้อมูลที่อยู่ในแอดเดรสที่กำหนด

รูปแบบ

D

D addr1

D addr1 addr2

คำอธิบาย

- ในกรณีที่ใช้ D เพียงตัวเดียวก็จะแสดงข้อมูลโดยใช้แอดเดรสล่าสุดที่จำได้ โดยจะแสดงข้อมูลเป็นจำนวน 128 ไบต์

- ในกรณีใช้ D แล้วตามด้วยแอดเดรสก็จะแสดงข้อมูลตั้งแต่แอดเดรสนั้นเป็นจำนวน 128 ไบต์

- ในกรณีที่ใช้ D แล้วตามด้วยแอดเดรส 1 และแอดเดรส 2 จะแสดงข้อมูลในหน่วยความจำตั้งแต่แอดเดรส 1 ไปจนถึงแอดเดรส 2 โดยที่ addr 1 เป็นแอดเดรสเริ่มต้น addr 2 เป็นแอดเดรสสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งย่อย ctrl W, ctrl A หยดการถ่ายเทข้อมูลชั่วคราวและสามารถถ่ายเทข้อมูลต่อไปได้โดยการกดคีย์ใด ๆ
ctrl x หรือ DEL ยกเลิกการถ่ายเทข้อมูล

ตัวอย่าง

>D E300

```
E300 20 20 44 45 4C 0D 04 BD - A2 F4 26 0A CE E3 1D BD
E310 A3 F2 BD A3 E0 39 CE A4 - 95 BD A3 F2 39 20 41 64
E320 64 72 65 73 73 20 20 20 - 20 20 20 20 20 20 20
E330 20 20 56 65 63 74 6F 72 - 20 0D 20 20 32 30 43 34
E340 20 20 20 20 20 53 43 49 - 20 0D 20 20 32 30 43 37
E350 20 20 20 20 20 53 50 49 - 20 0D 20 20 32 30 43 41
E360 20 20 20 20 20 50 75 6C - 73 65 20 41 63 63 75 6D
E370 75 6C 61 74 6F 72 20 49 - 6E 70 75 74 20 45 64 67
```

>

คำสั่ง
ทำหน้าที่
รูปแบบ

E
แก้ไขข้อมูลในหน่วยความจำบางไบต์
E
E addr

คำอธิบาย

- ในกรณีที่ใช้ E เพียงตัวเดียวก็จะเป็นการแก้ไขข้อมูลโดยการใช้อัดเดรสล่าสุดที่จำได้
- ในกรณีใช้ E แล้วตามด้วยอัดเดรสก็จะเป็นการแก้ไขข้อมูลตามอัดเดรสที่กำหนด

เมื่อใช้คำสั่งนี้มีลคิบอร์คจะแสดงค่าอัดเดรสพร้อมทั้งข้อมูลเดิมที่มีอยู่ ถ้าผู้ใช้ต้องการแก้ไขข้อมูลในอัดเดรสนั้นก็เพียงแต่ใส่ข้อมูลเป็นเลขฐาน 16 จำนวน 1 ไบต์แล้วให้กดคีย์ Enter หรือกด space ข้อมูลจะถูกเขียนลงในอัดเดรสนั้นในกรณีที่ต้องการคิดค่า RELATIVE ก็เพียงแต่ใส่อัดเดรสที่จะกระโดดไปแล้วตามด้วย 0 ซึ่งจะต้องเขียนติดกับอัดเดรส มีลคิบอร์คจะทำการคำนวณค่า RELATIVE ให้และจะแสดงค่านี้ให้ทราบ แต่จะไม่เขียนลงในหน่วยความจำให้

คำสั่งย่อย
=
-
space

แอดเดรสเดิม
ลคค่าอัดเดรสลง 1
เพิ่มค่าอัดเดรสขึ้น 1

Enter ออกจากคำสั่งนี้ไปสู่เครื่องหมาย prompt
<addr>0 คำนวณค่า RELATIVE

ตัวอย่าง

```
>E 2200
2200 86 <space>
2201 41 7E <space>
2202 C6 22000 FD
2202 C6 =
2202 C6 -
2201 7E <Enter>
>
```

คำสั่ง ET
ทำหน้าที่ แก้ไขข้อมูลในหน่วยความจำซึ่งเป็นรหัส ASCII ที่เป็นตัวอักษร
รูปแบบ ET
 ET addr
คำอธิบาย - กรณีใช้ ET เพียงตัวเดียวนั้นจะเป็นการแก้ไขข้อมูลโดยใช้แอดเดรสล่าสุด
 ที่จำได้
 - ในกรณีที่ใช้ ET แล้วตามด้วยแอดเดรสเป็นการแก้ไขข้อมูลตามแอดเดรส
 ที่กำหนด

 เมื่อใช้คำสั่งนี้มีลติบอร์จะแสดงค่าแอดเดรสที่เริ่มต้นให้ทราบก่อนและ
 พร้อมจะรับตัวอักษรที่ผู้ใช้คีย์ลงไป ค่าแอดเดรสจะเพิ่มขึ้นเองโดยอัตโนมัติ
 นอกจากนี้เรายังสามารถใช้คีย์ Backspace ในการแก้ไขข้อมูลได้ ถ้าจะ
 ออกจากคำสั่งนี้ก็ให้กดคีย์ Enter แล้วมีลติบอร์จะแสดงค่าแอดเดรสของ
 ข้อมูลตัวอักษรตัวสุดท้ายให้ทราบ

ตัวอย่าง

```
>ET 2200
2200 >ABCDEFGHIJKLMNQRSTUWXYZ
-End data *2219
>
```

คำสั่ง F
ทำหน้าที่ ใส่อข้อมูล (Fill data) ลงในหน่วยความจำ
รูปแบบ F addr1 addr2
 F addr1 addr2 data
คำอธิบาย คำสั่งนี้จะทำการนำข้อมูลเลขฐาน 16 ขนาด 1 ไบต์เขียนลงในหน่วยความจำตามช่วงของแอดเดรสที่กำหนดโดยที่

addr1 = แอดเดรสเริ่มต้น

addr2 = แอดเดรสสุดท้าย

data = ข้อมูลขนาด 1 ไบต์ที่จะเขียนลงในหน่วยความจำ ถ้าไม่ใส่ข้อมูลลงไปมีลคิบอร์ดจะกำหนดค่าให้เป็น \$FF

ตัวอย่าง

```
>F 2200 2280 2E
```

```
>
```

คำสั่ง G

ทำหน้าที่ ประมวลผลโปรแกรมตามที่ได้ใช้เขียนขึ้น

รูปแบบ G ประมวลผลตามตำแหน่งที่ PC ที่อยู่ในขณะนั้น

G addr ประมวลผลตามแอดเดรส ที่กำหนด

คำอธิบาย ทำการประมวลผลโปรแกรมที่ผู้ใช้เขียนขึ้น โดยมัลติบอร์ดจะโหลดเอาค่าของรีจิสเตอร์ต่าง ๆ ที่จำไว้เข้าไปในตัวชิพชุดชิพ แต่ในการประมวลผลนี้ค่า PC หรือค่าแอดเดรส จะต้องไม่ใช่ตำแหน่งของรอมใน DEBUGGER

ตัวอย่าง

```
>A 2200
```

```
2200: LDAA #41
```

```
2202: LDAB #00
```

```
2204: JSR FF00
```

```
2207: LDAB #25
```

```
2209: JSR FF00
```

```
220C: .
```

```
>G 2200
```

```
A
```

```
>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง	I
ทำหน้าที่	อ่านข้อมูลจากพอร์ต
รูปแบบ	I port I port L
คำอธิบาย	ใช้สำหรับการอ่านข้อมูลจากพอร์ตตามหมายเลขพอร์ตที่กำหนด โดยแสดงค่าที่อ่านได้ในขณะนั้น ในกรณีที่ใช้คำสั่ง I port L ข้อมูลที่อ่านได้จะถูกแลตซ์ไว้จนกว่าข้อมูลที่รับเข้ามาใหม่มีการเปลี่ยนแปลงไปจากค่าเดิมก็จะแสดงค่าใหม่ให้ทราบ การออกจากคำสั่งทำได้โดยการกด ctrl A, ctrl X หรือ DEL

ตัวอย่าง

```
>I 16A0 L
F6
F7
F8
F9
F0
F1
F2
F3
F4
F5
>
```

คำสั่ง	L
ทำหน้าที่	ดาวน์โหลดข้อมูลจากเครื่องคอมพิวเตอร์ลงบนมัลติบอร์ด
รูปแบบ	L L offset
คำอธิบาย	ข้อมูลที่จะดาวน์โหลดลงมาจะต้องอยู่ในรูปแบบ Motorola hex format - ในกรณีที่ใช้คำสั่ง L เพียงอย่างเดียว ข้อมูลที่ดาวน์โหลดลงมาจะลงตามตำแหน่งที่ได้กำหนดค่าเริ่มต้นไว้ในโปรแกรม - ในกรณีให้คำสั่ง L แล้วตามด้วยค่าออฟเซตจะเป็นการดาวน์โหลดข้อมูลลงหน่วยความจำตำแหน่งที่ได้ตั้งค่าเริ่มต้นไว้บวกกับค่าออฟเซต เช่น ตั้งค่าเริ่มต้นโปรแกรมที่ \$2200 แล้วใช้คำสั่ง L 1000 ข้อมูลจะถูกดาวน์โหลดมาที่แอดเดรส \$3000 แต่ข้อมูลภายในโปรแกรมยังเป็นข้อมูลเดิม

ก็จะทำให้ข้อมูลที่ถูกดาวน์โหลดลงมา มีความถูกต้อง การกดเพื่อดาวน์โหลด สามารถทำได้โดยให้กด >L <Enter> จากนั้นให้กด page up แล้วเลือก เบอร์ 4 (ASCII) แล้วกด Enter ให้ใส่ชื่อไฟล์ที่จะทำการดาวน์โหลด แล้วกด Enter ข้อมูลจะถูกโหลดลงมาบนมัลติบอร์ด

คำสั่ง M
ทำหน้าที่ คัดลอกข้อมูลเป็นช่วง
รูปแบบ M addr1 addr2 addr3
คำอธิบาย ใช้ในการคัดลอกข้อมูลจากช่วงหนึ่งไปยังอีกช่วงหนึ่งโดยที่

addr1 = แอดเดรสเริ่มต้นของช่วงที่ 1

addr2 = แอดเดรสสุดท้ายของช่วงที่ 1

addr3 = แอดเดรสเริ่มต้นของช่วงปลายทาง

ตัวอย่าง

```
>M 2200 2200 3000
```

```
>
```

คำสั่ง O
ทำหน้าที่ ส่งข้อมูลออกไปยังพอร์ตที่กำหนด
รูปแบบ O port data
คำอธิบาย จะทำการส่งข้อมูลออกไปยังพอร์ตที่กำหนดโดยที่ โดยที่

port = เบอร์พอร์ตที่ต้องการส่งข้อมูลออกไป

data = ข้อมูลที่ต้องการส่งออกไป

ตัวอย่าง

```
>O 1203 80
```

```
>
```

คำสั่ง P
ทำหน้าที่ ประมวลผลโปรแกรมโดยไม่มีสนใจ BREAK ที่ตั้งไว้และจะอับโหลดข้อมูลจาก
มัลติบอร์ดไปยังเครื่องคอมพิวเตอร์

รูปแบบ P
P addr1 addr2

คำอธิบาย - กรณีที่ใช้คำสั่ง P อย่างเดียวจะเป็นการประมวลผลโปรแกรมโดยไม่มีสนใจ
จุด BREAK ที่ตั้งไว้ ซึ่งจะประมวลผลโปรแกรมตรงตำแหน่ง PC ที่อยู่
- ในกรณีที่ต้องการอับโหลดข้อมูลจากมัลติบอร์ดไปยังเครื่องคอมพิวเตอร์ ให้
ใช้คำสั่ง P แล้วตามด้วย addr1 และ addr2 โดยที่

addr1 = แอดเดรสเริ่มต้นที่จะทำการอับโหลด

addr2 = แอดเดรสสุดท้ายที่จะทำการอับโหลด

เป็นการนำข้อมูลตั้งแต่แอดเดรส 1 จนถึงแอดเดรส 2 ขึ้นไปเก็บเป็น
ไฟล์บนเครื่องคอมพิวเตอร์ในรูปแบบของ Motorola hex format ซึ่งจะ
ต้องใช้ฟังก์ชันของ PROCOMM เข้าช่วย

ตัวอย่าง

>P 2000 200F

ให้กดคีย์ page down แล้วเลือกเบอร์ 4 (ASCII) แล้วกด Enter
จากนั้นให้ใส่ชื่อไฟล์ที่ต้องการแล้วกด Enter อีกครั้ง ที่หน้าจอแสดงผลของ
เครื่องคอมพิวเตอร์จะแสดงข้อมูลที่ถูกล่วงไปเก็บ เมื่อสิ้นสุดการส่งไฟล์แล้วจะ
แสดงเครื่องหมาย prompt จากนั้นกด ESC เป็นคั่นเสร็จเรียบร้อย คือ

>P 2200 2220

S1132200841C600BDF00C625BDF0000000000DA

S11322100000000000000000000000000000000BA

S104222000B9

S9030000FC

>

คำสั่ง Q
ทำหน้าที่ ออกจากโปรแกรม DEBUGGER
รูปแบบ Q
คำอธิบาย ใช้ในการออกจากโปรแกรม DEBUGGER ซึ่งจะทำให้ที่พีชอยู่ STOP MODE คือ หยุดทำคำสั่งและอินพุต/เอาต์พุต ภายในหยุดการทำงานหมด

ตัวอย่าง

```
>Q
system stop
```

คำสั่ง R
ทำหน้าที่ แสดงค่ารีจิสเตอร์และกำหนดค่าให้กับรีจิสเตอร์
รูปแบบ R
R name
คำอธิบาย เมื่อใช้คำสั่งนี้ทั้ง 2 รูปแบบจะแสดงค่าในรีจิสเตอร์ทั้งหมดก่อน

- ในกรณีที่ใช้คำสั่ง R เพียงตัวเดียว บรรทัดถัดมาจะแสดงค่าในรีจิสเตอร์ และจะรอรับข้อมูลจากคีย์บอร์ด ถ้าต้องการเปลี่ยนแปลงค่าของรีจิสเตอร์ก็ใส่ค่าใหม่ลงไปแล้วกด Enter หรือ space bar
- กรณีที่ใช้คำสั่ง R name เมื่อ name คือชื่อของรีจิสเตอร์ที่ต้องการแก้ไขเพื่อใส่ค่าใหม่ลงไป ซึ่งมีชื่อดังต่อไปนี้

P = program counter
Y = index register Y
X = index register X
A = accumulator A
B = accumulator B
C = condition code
S = stack pointter

ถ้ากด Enter ก็จะกำหนดค่าให้กับรีจิสเตอร์แล้วออกจากคำสั่งนี้ ถ้าหากกด space bar จะกำหนดค่าให้กับรีจิสเตอร์แล้วแสดงค่าของรีจิสเตอร์ตัวถัดไป

ตัวอย่าง

```
>R
```

```
P-E7BF Y-BDFF X-FFFF A-FD B-EE C-DO S-2050
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

P-E7BF 2200 <space>
 Y-BDFF 0000 <space>
 X-FFFF 0000 <space>
 A-FD 01 <space>
 B-EE 45 <space>
 C-D0 <space>
 S-2050 <Enter>

>

คำสั่ง REG
 ทำหน้าที่ ดูค่ารีจิสเตอร์ภายในของ 68HC11
 รูปแบบ REG
 คำอธิบาย เพื่อดูค่ารีจิสเตอร์ภายในขณะนั้น รายละเอียดที่แสดงออกมาจะมีรูปแบบดังนี้

address : data1 data2 register

ในกรณีที่รีจิสเตอร์นั้นมีขนาด 8 บิต ก็จะมีเฉพาะ data1 อย่างเดียว ส่วนในกรณีที่รีจิสเตอร์นั้นมีขนาด 16 บิต data1 จะแสดงข้อมูลที่แอดเดรส นั้นซึ่งเป็นไบต์สูง ส่วน data2 เป็นข้อมูลของแอดเดรสถัดไปซึ่งเป็นไบต์ต่ำ

ตัวอย่าง

>REG

```

1000:8F ..PORTA 1001:00 ..      1002:F7 ..PIOC 1003:3E ..PORTC
1004:FF ..PORTB 1005:FD .PORTCL 1006:FF ..      1007:37 ...DDRC
1008:3F ..PORTD 1009:00 ...DDR  100A:00 ..PORTE 100B:00 ..CFORC
100C:00 ..OC1M  100D:00 ...OC1D  100E:81 F8 TCNT 1010:FF FF TIC1
1012:FF FF TIC2 1014:FF FF TIC3 1016:FF FF TOC1 1018:FF FF TOC2
101A:FF FF TOC3 101C:FF FF TOC4 101E:FF FF TOC5 1020:03 ..TCTL1
1021:00 ..TCTL2 1022:00 ..TMSK1 1023:F8 ..TFLG1 1024:00 ..TMSK2
1025:C0 ..TFLG2 1026:00 ..PACTL 1027:55 ..PACNT 1028:05 ...SPCR
1029:00 ...SPSR 102A:00 ...SPDR 102B:22 ...BUAD 102C:00 ..SCCR1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

102D:0C ..SCCR2  102E:10 ...SCSR  102F:0D ...SCDR  1030:80  ADCTAL
1031:00 ...ADR1  1032:00 ...ADR2  1033:00 ...ADR3  1034:00 ...ADR4
1035:00 ...      1036:00 ...      1037:00 ...      1038:00 ...
1039:93 OPTION  103A:00  COPRST  103B:00 ..PPROG  103C:25 ..HPRIO
103D:01 ..INIT  103E:00 ..TEST1  103F:0D  CONFIG

```

>

คำสั่ง S

ทำหน้าที่ ทดสอบการทำงานในส่วนต่าง ๆ

รูปแบบ S

คำอธิบาย ให้เลือกทดสอบการทำงานมัลติบอร์ดในส่วนต่าง ๆ ให้กดตัวเลขด้านหน้าที่จะทำการทดสอบ

ตัวอย่าง

>S

```

+-----+
| Test board CP-68HC11 |
+-----+

```

<Enter> to exit

- 1: Test LCD 16 Character
- 2: Test RAM 8K at addr 2000
- 3: Test RAM 32K test bank
- 4: Test Timer Input Capture 1
- 5: Test Timer Input Capture 2
- 6: Test Timer Input Capture 3
- 7: Test Timer Output Compare
- 8: Test A/D Channal AN0-AN3
- 9: Test A/D Channal AN4-AN7
- A: Test Printer Port
- B: Test Port 8255

?

หมายเหตุ การทดสอบแรม 8 กิโลไบต์ใช้พื้นที่ในหน่วยความจำภายในแอดเดรส \$00FF ในการเก็บข้อมูลชั่วคราว

คำสั่ง T

ทำหน้าที่ ทำชิงเกิลสแต็ป

รูปแบบ T

T n

คำอธิบาย ใช้ทดสอบการทำงานของโปรแกรม โดยเริ่มทดสอบแอดเดรสที่ PC ซอย์ไปที่ละ 1 คำสั่งสามารถตั้งจำนวนการสแต็ปใน 1 ครั้งว่าให้ทำสแต็ปละกี่คำสั่ง เมื่อให้คำสั่งนี้ มัลติบอร์ดจะแสดงคำสั่งและค่ารีจิสเตอร์ให้ทราบและสามารถทำสแต็ปต่อ ๆ ไปได้โดยการกด Enter คำสั่งนี้จะใช้ได้ก็ต่อเมื่อใส่สายจัมเข้าที่ JOC5 ซึ่งจะทำการต่อขา XIRQ เข้ากับ OC5 ในกรณีที่โปรแกรมนั้นมีการเรียกใช้ SYSTEM CALL เราก็จะถือว่าการเข้าไปในโปรแกรมข้อของ SYSTEM CALL เป็น 1 คำสั่ง

ตัวอย่าง

>R

P-FOFO Y-FOFO X-FOFO A-FO B-FO C-DO S-2050

P-FOFO 2200

>T

LDAA #\$41 P-2202 Y-FOFO X-FOFO A-41 B-FO C-90 S-2050

>

LDAB #\$00 P-2204 Y-FOFO X-FOFO A-41 B-00 C-94 S-2050

<

JSR \$FF00 P-FF00 Y-FOFO X-FOFO A-41 B-00 C-94 S-204E

>

A

BRA \$2200 P-2200 Y-FOFO X-FOFO A-41 B-00 C-90 S-2050

>

LDAA #\$41 P-2202 Y-FOFO X-FOFO A-41 B-00 C-90 S-2050

>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- คำสั่ง** U
- ทำหน้าที่** disassemble (DASM)
- รูปแบบ** U
U addr1
U addr1 addr2
- คำอธิบาย**
- ในกรณีที่ใช้ U เพียงตัวเดียวก็จะทำการ DASM โดยใช้แอดเดรสล่าสุดที่จำได้เป็นจำนวน 16 ไบต์
 - ในกรณีที่ใช้ U แล้วตามด้วยแอดเดรสก็จะทำการ DASM ตั้งแต่แอดเดรสนั้นเป็นจำนวน 16 ไบต์
 - กรณีที่ใช้ U แล้วตามด้วยแอดเดรส 1 และแอดเดรส 2 จะทำการ DASM ตั้งแต่แอดเดรส 1 ไปจนถึงแอดเดรส 2 โดยที่

addr1 เป็นแอดเดรสเริ่มต้น

addr2 เป็นแอดเดรสสุดท้าย

- คำสั่งย่อย** ctrl W, ctrl A = หยุดการ DASM ชั่วคราวและสามารถ DASM ต่อได้โดยการกดคีย์ใด ๆ ก็ได้
- ctrl X หรือ DEL = ยกเลิกการ DASM

ตัวอย่าง

```
>U 2200 221B
2200 86 30      LDAA #$30
2202 B7 10 2B  STAA $102B
2205 86 00      LDAA #$102C
2207 B7 10 2C  STAA $102C
220A 86 08      LDAA #$08
220C B7 10 2D  STAA $102D
220F B6 10 2E  LDAA $102E
2212 84 80      ANDA #$80
2214 27 F9      BEQ $220F
2216 86 41      LDAA #$41
2218 B7 10 2F  STAA $102F
221B 7E 22 0F  JMP $220F
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง V
 ทำหน้าที่ เปรียบเทียบข้อมูลในหน่วยความจำกับข้อมูลที่ทำกรดาวน์โหลด
 รูปแบบ V
 V offset
 คำอธิบาย คำสั่งนี้จะมีลักษณะคล้ายกับคำสั่ง L แต่จะไม่เขียนข้อมูลลงในหน่วยความจำ คือเปรียบเทียบข้อมูลที่จะโหลดลงมากับข้อมูลที่มีอยู่ในหน่วยความจำ ถ้าข้อมูลไม่ตรงกันก็จะแสดงค่าแอดเดรสที่มีข้อมูลไม่ตรงกันให้ทราบ

ตัวอย่าง

```
>V 2200
```

```
DONE
```

```
>
```

คำสั่ง VT
 ทำหน้าที่ แสดงอินเตอร์รัพต์เวคเตอร์
 รูปแบบ VT
 คำอธิบาย แสดงอินเตอร์รัพต์เวคเตอร์ให้ผู้ใช้ได้ทราบ ซึ่งได้ข้ายอินเตอร์รัพต์เวคเตอร์ให้มาอยู่ที่แรมตั้งแต่แอดเดรส \$20C4 ถึงแอดเดรส \$20FF แต่ละเวคเตอร์จะจองเนื้อที่ไว้ 3 ไบต์ผู้ใช้เพียงแต่ใส่คำสั่ง JUMP(7E) ตามด้วยแอดเดรสที่จะกระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัพต์อีก 2 ไบต์ ดังนั้นเมื่อเกิดการอินเตอร์รัพต์ขึ้นก็สามารถไปทำโปรแกรมตอบสนองการอินเตอร์รัพต์ของผู้ใช้ได้

ตัวอย่าง

```
>VT
```

Address	vector
20C4	SCI
20C7	SPI
20CA	Pulse Accumulator Input Edge
20CD	Pulse Accumulator Overflow
20D0	Timer Overflow
20D3	Timer Output Compare 5
20D6	Timer Output Compare 4
20D9	Timer Output Compare 3

20DC	Timer Output Compare 2
20DF	Timer Output Compare 1
20E2	Timer Input Capture 3
20E5	Timer Input Capture 2
20E8	Timer Input Capture 1
20EB	Real Time Interrupt
20EE	IRQ
20F1	XIRQ
20F4	SWI
20F7	Illegal Opcode
20FA	COP Fail
20FD	COP Clock Monitor Fail

คำสั่ง	BULK
ทำหน้าที่	ลบข้อมูลที่มีอยู่ในอ็พรอม โดยจะทำให้ข้อมูลที่มีอยู่ในอ็พรอมมีค่าเป็น \$FF
รูปแบบ	BULK
คำสั่ง	TOOLS
ทำหน้าที่	วัดความถี่, ความกว้างของพัลส์ และคาบเวลาของพัลส์
รูปแบบ	TOOLS
คำอธิบาย	เป็นการนำเอาอินพุตแคปเจอร์มาประยุกต์ใช้ในการวัดความถี่ วัดความกว้างของพัลส์และคาบเวลาของพัลส์

ตัวอย่าง

>TOOLS

-----Application for input capture 1 -----

1: Measures frequency

max 60,606Hz, min 30.5 Hz

2: Measures time between a rising edge and a falling edge

max 8.38 s, min 35 us

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3: Measures long time between two rising edges

max 32.768 ms

7

SYSTEM CALL

SYSTEM CALL คือการเรียกใช้โปรแกรมย่อยในรอมมอนิเตอร์ของมัลติบอร์ด ซึ่งจะ
ทำให้ผู้ใช้มีความสะดวกในการเขียนโปรแกรมหรือพัฒนาโปรแกรม โดยไม่เสียเวลาไปในการ
การเขียนโปรแกรมในส่วนนี้ โปรแกรมย่อยถูกเรียกใช้โดย SYSTEM CALL ของมัลติบอร์ด
นั้นอยู่ 2 ส่วน คือ

- (1) ส่วนของการทำงานบนมัลติบอร์ด
- (2) ส่วนของการทำงานบน DEBUGGER

การเรียกใช้ SYSTEM CALL ในส่วนของการทำงานบนมัลติบอร์ด

การเรียกใช้สามารถทำได้โดยการนำรหัสของโปรแกรมใน SYSTEM CALL ใส่ง
ไปในรีจิสเตอร์ A แล้วตามหลังด้วยคำสั่ง JSR RECALL แอดเดรสของ RECALL จะดูได้
จาก HELP MENU บนมัลติบอร์ด ซึ่งจะอยู่ในตำแหน่ง \$CE2B ตัวอย่างต้องการให้แสดงผล
ของการ ERROR บนแอลอีดี 7 เซกเมนต์ทำได้ดังนี้

```
LDA #10 ; SUBROUTINE SYSTEM CALL ERRST
JSR $CE2B
```

การเรียกใช้ SYSTEM CALL ในส่วนของการทำงานบน DEBUGGER

จะทำได้โดยการนำรหัสของโปรแกรมใน SYSTEM CALL มาเก็บไว้ที่รีจิสเตอร์ B
หลังจากนั้นก็ให้ใส่คำสั่ง JSR \$FF00 เช่น ต้องการส่งข้อมูล \$07 ออกจอสามารถทำได้
โดย

```
LDAB #01 ; SUBROUTINE SYSTEM CALL OUTH
LDA #07
JSR $FF00
LDAB #$24 ; SUBROUTINE SYSTEM CALL MAIN
JSR $FF00
```

สิ่งที่จำเป็นในการเรียกใช้ SYSTEM CALL สรุปได้ดังนี้

- 1) ใส่รหัสโปรแกรมของ SYSTEM CALL ที่เราจะใช้ให้กับรีจิสเตอร์ A และ B
- 2) ดูว่าโปรแกรม SYSTEM CALL นั้นต้องการอินพุตอะไรก็ใส่ลงไปตามนั้น
- 3) ดูว่าโปรแกรม SYSTEM CALL นั้นได้ใช้รีจิสเตอร์อะไรไปแล้วบ้าง ไม่ควรใช้รีจิสเตอร์ตัวนั้น ถ้าจะใช้ก็ให้ PUSH ค่าเก็บไว้ก่อน
- 4) ถ้าใช้คำสั่ง JSR \$FF00 หรือ JSR \$CE2B จะเป็นการเข้าสู่ SYSTEM CALL

หมายเหตุ

- ในการใช้โปรแกรมย่อยเกี่ยวกับการส่งข้อมูลออกจอแอลซีดีนั้น ผู้ใช้จะต้องคำนึงถึงแอดเดรสของไดนามิกแรมด้วยว่าเมื่อส่งข้อมูลออกไปแล้วข้อมูลจะอยู่ในตำแหน่งมองเห็นได้ที่จอแอลซีดี ซึ่งการจัดแอดเดรสของไดนามิกแรมของแอลซีดีแต่ละรุ่นจะไม่เหมือนกัน
- ก่อนที่จะเรียกใช้โปรแกรมย่อยที่เกี่ยวกับเครื่องพิมพ์ต้องสั่งชุดควบคุม 8255 ก่อน

โปรแกรมย่อยในส่วนของการทำงานบนมัลติบอร์ด

HON

รหัส \$00

ทำหน้าที่ ให้โปรแกรมกลับเข้าสู่โปรแกรมมอนิเตอร์โดยตรง ดังนั้นหากจะใช้โปรแกรมอีกครั้งต้องประมวลผลใหม่จึงไม่ต้องคำนึงถึงรีจิสเตอร์ใด

อินพุต -

เอาท์พุต -

SOUND

รหัส \$01

ทำหน้าที่ กำเนิดเสียงออกลำโพง

อินพุต ACCB ระดับความดังของเสียง

REGC ระดับความถี่ของเสียงถ้ามีค่ามากความถี่ต่ำและถ้าค่าน้อยความถี่สูง

IX ช่วงเวลาในการกำเนิดเสียง

เอาท์พุต -

SCAN

รหัส \$02

ทำหน้าที่ 1. นำค่าบัพเพอร์การแสดงผลไปแสดงที่แอลอีดี 7 เซกเมนต์

2. จะออกจากโปรแกรมย่อยนี้เมื่อมีการกดคีย์ โดยจะให้ค่ารหัสของคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ถ้าคีย์ที่กดนั้นเป็นถุกกดค้างมาก่อน โปรแกรมจะไม่ยอมรับคีย์นั้นจนกว่าจะปล่อยคีย์นั้นก่อน
4. จะตรวจคีย์ที่เป็น AUTO REPEAT ด้วย คือ จะออกไปจากโปรแกรมย่อยทันทีถ้าคีย์ที่ถุกกดเป็นคีย์ AUTO REPEAT และถึงกำหนดเวลาแล้ว คีย์ที่เป็น AUTO REPEAT มีอยู่ 3 คีย์ คือ STEP, INC และ DEC
5. จะกำเนิดเสียงดังสัญญาณถ้าคีย์ที่กดเป็นคีย์ 0-F จะกำเนิดเสียงต่ำและนอกนั้นจะกำเนิดเสียงสูง
6. ตรวจคีย์ INSTANT คือจะไปทำงานตามหน้าที่ที่ที่กดคีย์เหล่านี้ นั่นคือ คีย์ MON, DATA, STEP, RUN, และ FUNC แต่สามารถตัดการตรวจเช็คคีย์เหล่านี้ได้โดยการเซตแฟล็กบิตที่ 4 ให้เป็น "1"

อินพุต
เอาท์พุต

ค่าที่อยู่ในบัพเฟอร์ของการแสดงผล
ค่ารหัสคีย์ที่ A และในหน่วยความจำ KEYIN

CLEAR

รหัส

\$03

ทำหน้าที่

เคลียร์บัพเฟอร์การแสดงผลให้เป็น 0

อินพุต

เอาท์พุต

SCAND

รหัส

\$04

ทำหน้าที่

นำค่าบัพเฟอร์การแสดงผลไปแสดงที่แอลอีดี 7 เซกเมนต์ โดยจะหน่วงเวลาชั่วขณะด้วย ACCB

อินพุต

ACCB

เอาท์พุต

SCANK

รหัส

\$05

ทำหน้าที่

เคลียร์ค่า KEYIN ที่เก็บรหัสคีย์ไว้ให้มีค่าเป็น \$FF แล้วนำค่าในบัพเฟอร์การแสดงผลไปแสดงที่แอลอีดี 7 เซกเมนต์ โดยแสดงเพียง 1 รอบเท่านั้น ภายใน 1 รอบนี้จะมีการตรวจคีย์ด้วย ถ้าหากมีการกดคีย์ก็จะทำให้ได้ค่าคีย์ที่กดนั้นใน KEYIN

เลขที่ชุด KEYIN

PACK

รหัส \$06

ทำหน้าที่ ทำการเปลี่ยนข้อมูลที่อยู่ในบัพเฟอร์ของการแสดงผลที่เก็บค่ารหัสเซกเมนต์ให้เป็นเลขฐาน 16 และเก็บไว้ในบัพเฟอร์

อินพุต บัพเฟอร์การแสดงผล

เลขที่ชุด -

UNPK

รหัส \$07

ทำหน้าที่ เปลี่ยนเลขฐานสิบหกที่อยู่ในบัพเฟอร์ให้เป็นรหัสของเซกเมนต์และนำไปไว้ในบัพเฟอร์ของการแสดงผลเพื่อแสดงผลต่อไป

อินพุต บัพเฟอร์

เลขที่ชุด บัพเฟอร์การแสดงผล

IN2

รหัส \$08

ทำหน้าที่ รับข้อมูล 2 หลักจากคีย์บอร์ด และออกจากโปรแกรมย่อยเมื่อคีย์ INC ถูกกดหรือกดคีย์ที่เป็น INSTANT KEY

อินพุต ACCB - ค่ารหัสเซกเมนต์ที่แสดงผลหลักสุดท้าย

(IY) - ตำแหน่งที่จะนำข้อมูล 2 หลักที่ได้จากการกดคีย์ไปเก็บไว้ และยังสามารถกำหนดค่าข้อมูลเริ่มแรกไว้ที่ IY ได้โดยจะนำข้อมูลที่อยู่ในตำแหน่งนี้ไปท้อแสดงผล 2 หลักทางซ้าย ซึ่งถ้าไม่มีการกดคีย์ตัวเลขเปลี่ยนแปลงก็จะทำให้ได้ค่าที่กำหนดไว้เช่นเดิม

เลขที่ชุด A, (IY) ข้อมูลที่รับเข้ามา

IN4

รหัส \$09

ทำหน้าที่ รับข้อมูล 4 หลักจากคีย์บอร์ด โดยจะออกจากโปรแกรมย่อยเมื่อกดคีย์ INC หรือกดคีย์ที่เป็น INSTANT KEYS

อินพุต ACCB - รหัสเซกเมนต์ที่แสดงผลหลักสุดท้าย

IX - ค่าข้อมูลที่จะทำการอ่านเข้ามา คือ เป็นค่าที่จะให้จอแสดงผลตอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มแรก 4 หลักและถ้าไม่มีการกดตัวเลขใด ๆ ก็จะทำให้ได้ค่าที่ตั้งไว้แล้ว แต่ถ้ามีการกดตัวเลขจะเก็บไว้ที่ IX

เอากัณฑ์ IX

PRINT

รหัส \$0A

ทำหน้าที่ สำหรับแสดงข้อมูลบนจอแสดงผลในหน่วยความจำช่วงใด ๆ

อินพุต IY - แอดเดรสของส่วนที่จะนำข้อมูลไปแสดงผล

ACCB - เวลาในการทวนวง

REGC - จำนวนหลักที่แสดงผลต้องไม่เกิน 7 หลักโดยนับรวมทั้งแฟล็กด้วย

เอากัณฑ์ -

READY

รหัส \$0B

ทำหน้าที่ สำหรับแสดงตัวอักษร READY บนจอแสดงผล เพื่อเป็นการเตรียมพร้อมไปทำงานตามโปรแกรมต่อไป เราจะออกจากโปรแกรมย่อยได้เมื่อกดคีย์ INC หรือกดคีย์ที่เป็น INSTANT KEY เท่านั้น

อินพุต -

เอากัณฑ์ -

HBEEP

รหัส \$0C

ทำหน้าที่ กำหนดเสียงสัญญาณความถี่สูง

อินพุต -

เอากัณฑ์ -

DELAY

รหัส \$0D

ทำหน้าที่ หน่วงเวลา

อินพุต B - ช่วงเวลาในการทวนวง

เอากัณฑ์ -

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MON 1

รหัส	\$OE
ทำหน้าที่	กลับเข้าสู่โมไนเตอร์ โดยจะแสดงข้อมูลตามบัพเพอร์การแสดงผล
อินพุต	บัพเพอร์การแสดงผล
เอาต์พุต	-

POFFS

รหัส	\$OF
ทำหน้าที่	จะทำหน้าที่ถึง 2 อย่างคือ 1. ปิดมัลติบอร์ดด้วยการ HALT และเคลียร์ POWER CODE เพื่อให้เหมือนกับ การปิดมัลติบอร์ดครั้งแรกและข้อมูลจะไม่สูญหายถ้าเราปิดด้วยการ HALT แล้วดึงแหล่งจ่ายไฟออก เนื่องจากบนมัลติบอร์ดไม่มีส่วนป้องกัน การเขียน 2. ทำการสำรองแรมและตรวจสอบข้อมูลว่าถูกต้องหรือไม่ด้วยการตรวจสอบ ที่ผลรวม (SUM)

อินพุต -
เอาต์พุต -

ERRST

รหัส	\$10
ทำหน้าที่	หยุดการทำงานและแสดง ERROR บนจอแสดงผล ใช้สำหรับการแจ้ง ERROR ต่าง ๆ จากโปรแกรม

อินพุต -
เอาต์พุต -

SONGS

รหัส	\$11
ทำหน้าที่	กำเนิดเสียงดนตรีตามความยาวและตารางที่กำหนด
อินพุต	REGD - จำนวนตัวโน้ต IY - ตำแหน่งเริ่มต้นของตาราง
เอาต์พุต	-

TITLE

รหัส	§12
ทำหน้าที่	กำเนิดเสียงตัวโน้ตเป็นลักษณะ TITLE
อินพุต	-
เอาท์พุต	-

TABLE

รหัส	§13
ทำหน้าที่	หาค่าในตารางที่กำหนด
อินพุต	ACCB - ตัวชี้ค่าในตาราง IX - ตำแหน่งเริ่มต้นของตาราง
เอาท์พุต	IX - ค่าที่ได้จากตาราง

INSF

รหัส	§14
ทำหน้าที่	รับค่าตัวเลข 4 หลัก 2 ชุด โดยมีลักษณะดังนี้ XXXX - S หมายถึงชุดเริ่มต้น (START) XXXX - F หมายถึงชุดสุดท้าย (FINAL)
อินพุต	-
เอาท์พุต	(START) (FINAL)

IND

รหัส	§15
ทำหน้าที่	รับค่าตัวเลข 4 หลัก โดยมีลักษณะดังนี้ XXXX - d หมายถึง DESTINATION
อินพุต	-
เอาท์พุต	DEST

SOUNDI

รหัส	§16
ทำหน้าที่	กำเนิดเสียงโดยเริ่มจากเสียงเบาไปจนถึงเสียงดังที่สุด
อินพุต	REGC - ระดับความถี่ของเสียงถ้าค่ามากความถี่ต่ำและค่าน้อยความถี่สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตเห็นาไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	IX	- ช่วงเวลาในการกำเนิดเสียง
เลาก์พุด	-	
SOUNDD		
รหัส	\$17	
ทำหน้าที่		กำเนิดเสียงโดยเริ่มจากเสียงดังที่สุดไปจนถึงเสียงเบา
อินพุต	REGC	- ระดับความถี่เสียง
	IX	- ช่วงเวลาในการกำเนิดเสียง
เลาก์พุด	-	
LBEEP		
รหัส	\$18	
ทำหน้าที่		กำเนิดเสียงสัญญาณความถี่ต่ำ
อินพุต	-	
เลาก์พุด	-	
CHKSUM		
รหัส	\$19	
ทำหน้าที่		จะตรวจสอบข้อมูลที่อยู่ในแอดเดรสที่ต้องการตรวจสอบว่าข้อมูลยังเหมือนเดิมหรือเปล่า โดยเมื่อให้ CHKSUM จะทำให้ได้ตัวเลขเฉพาะมาค่า ๆ หนึ่ง
อินพุต	IX	- แอดเดรสเริ่มมาต้น
	IY	- ช่วงแอดเดรส
เลาก์พุด	ACCA, REGD	
DELAYL		
รหัส	\$1A	
ทำหน้าที่		หน่วงเวลาโดยการลดค่าสูงสุดของ IX ลงจนเป็น 0
อินพุต	-	
เลาก์พุด	-	
DELAY_L		
รหัส	\$1B	
ทำหน้าที่		หน่วงเวลาโดยให้ ACCA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุต -
เอาต์พุต -

TX_SUB

รหัส \$1C
ทำหน้าที่ ส่งข้อมูลในตำแหน่งที่ IX ขึ้นช่องพอร์ต RS-232 จนเจอรหัสจบ \$04
อินพุต IX - ตำแหน่งเริ่มต้นของข้อมูล
เอาต์พุต -

M_ASCII

รหัส \$1D
ทำหน้าที่ ย้ายค่ารหัส ASCII จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่ง เราสามารถออก
จากโปรแกรมได้เมื่อเจอรหัสจบ
อินพุต IX - ตำแหน่งเริ่มต้นของตาราง ASCII
IY - ตำแหน่งใหม่ของตาราง ASCII
เอาต์พุต -

INIT_LCD

รหัส \$1E
ทำหน้าที่ กำหนดค่าเริ่มต้นของแอลซีดีแบบ DOT MATRIX โดยจะเซตให้ D=1, C=1,
B=1, DL=1, N=1, F=0, I/D=1, S=0 และเคลียร์จอแสดงผลพร้อมที่จะ
รับข้อมูล
อินพุต -
เอาต์พุต -

CURSOR_L

รหัส \$1F
ทำหน้าที่ ทำการเลื่อนเคอร์เซอร์ไปยังตำแหน่งที่ต้องการ ซึ่งเป็นการเซตไคโนมิกกรม
โดยแอตทริบิวต์ของจอแอลซีดีแต่ละแบบจะไม่เท่ากันซึ่งคุณได้จากคู่มือ
อินพุต ACCA
เอาต์พุต -

CLEAR_L

รหัส	\$20
ทำหน้าที่	เคลียร์จอแสดงผลของแอลซีดี
อินพุต	-
เอาต์พุต	-

READY_L

รหัส	\$21
ทำหน้าที่	อ่านแฟล็กว่าง (BUSY FLAG) ของแอลซีดีว่าพร้อมจะรับข้อมูลหรือยัง
อินพุต	-
เอาต์พุต	-

WRP

รหัส	\$22
ทำหน้าที่	เขียนข้อมูลออกที่แอลซีดี 1 แถว
อินพุต	IX - ตำแหน่งเริ่มต้นของตารางที่จะส่ง
เอาต์พุต	-

ROW_LCD

รหัส	\$23
ทำหน้าที่	เหมือนกับ WRP แต่มีการกำหนดตำแหน่ง เริ่มต้นของแอลซีดีด้วย
อินพุต	IX - ตำแหน่งเริ่มต้นของตารางที่จะส่ง
เอาต์พุต	-

ROW_LCDL

รหัส	\$24
ทำหน้าที่	เหมือนกับ ROW_LCD แต่หลังจากแสดงผลแล้วจะเรียกใช้ DELAY_L 6 ครั้ง
อินพุต	IX - ตำแหน่งเริ่มต้นของตารางที่จะส่ง
เอาต์พุต	-

OUT_L

รหัส	\$25
ทำหน้าที่	ส่งข้อมูล ASCII ที่อยู่ใน ACCA ออกจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุต A
เอาท์พุต -

OUT_LINE

รหัส §26
ทำหน้าที่ เขียนข้อมูลที่รีจิสเตอร์ IX ที่อยู่แอดเดรสที่ 8 ตัว
อินพุต IX - ตำแหน่งเริ่มต้นของข้อมูล
เอาท์พุต -

SEND

รหัส §27
ทำหน้าที่ ส่งข้อมูลใน BUF_PRN ออกพอร์ตเครื่องพิมพ์ 1 ตัวอักษร
อินพุต BUF_PRN
เอาท์พุต -

BUSY

รหัส §28
ทำหน้าที่ รอรับสัญญาณว่างจากเครื่องพิมพ์
อินพุต -
เอาท์พุต -

ONFK

รหัส §29
ทำหน้าที่ หยุดการทำงานของโปรแกรมชั่วคราวและแสดงค่า IX บนจอแสดงผล ณ จุดที่หยุดนี้ถ้าเรากด INC จะก็ออกจากโปรแกรมย่อย แต่ถ้ากด INSTANT KEYS ก็จะไปทำงานตามหน้าที่นั้น ๆ
อินพุต IX
เอาท์พุต -

FIND_GL

รหัส §2A
ทำหน้าที่ รอรับข้อมูลเลข 1 ถึงเลข 8 จากคีย์บอร์ดไปเก็บใน IY
อินพุต IY - ที่ว่างหน่วยความจำ (LENGTH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขที่ชุด -

FIND_GB

รหัส \$2B

ทำหน้าที่ รับข้อมูลตามความยาวซึ่งกำหนดใน IY (LENGTH) ไปเก็บไว้ใน FIND_BUF

อินพุต IY - ี่แอดเดรสเริ่มต้นของ FIND_BUF

เลขที่ชุด (FIND_BUF)

CHKRAM

รหัส \$2C

ทำหน้าที่ ตรวจสอบแรมตามแอดเดรสที่เรากำหนดไว้ ซึ่งจะแสดงผลการตรวจสอบเป็นช่วง ๆ ละ 256 ไบต์ ถ้าผลการตรวจสอบแรมเป็นปกติข้อมูลของแอดเดรสที่เราตั้งไว้จะมีค่า 00 แต่ถ้าแรมเกิดบกพร่องก็จะแสดงช่วงที่ผิด แล้วรอการกดคีย์แบบ INSTANT ถ้าเป็นคีย์ INC ก็จะทำกรตรวจสอบต่อไป ถ้าไม่ใช้ก็จะทำงานตามคีย์ที่กดนั้น

อินพุต IX และ HL แอดเดรสเริ่มต้น

IY และ BC แอดเดรสสุดท้าย

เลขที่ชุด -

โปรแกรมย่อยในส่วนของการทำงานบน DEBUGGER

OUTA

รหัส \$00

ทำหน้าที่ ส่งรหัส ASCII 1 ไบต์ไปที่ SCI เช่น A=41 ที่จอของเครื่องคอมพิวเตอร์จะเป็น A

อินพุต A

เลขที่ชุด -

รีจิสเตอร์ -

OUTH

รหัส \$01

ทำหน้าที่ นำค่าตัวเลขฐาน 16 ที่อยู่ในรีจิสเตอร์ A เปลี่ยนเป็น ASCII แล้วส่งออกไปที่ SCI เช่น A=41 ที่จอของเครื่องคอมพิวเตอร์จะเป็น 41 ด้วยเช่นกัน

อินพุต A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอากัณฑ์ -
วีจิสเตอร์ -

OUT1BYT

รหัส \$02

ทำหน้าที่ นำค่าตัวเลขฐาน 16 ที่วีจิสเตอร์ IX ที่อยู่ขนาด 1 ไบต์เปลี่ยนเป็น ASCII แล้วส่งไปที่ SCI แล้วค่าของวีจิสเตอร์ IX จะชี้ไบต์ถัดไป แทน ข้อมูลที่ IX ที่อยู่มีค่าเป็น \$41 ที่จอของเครื่องคอมพิวเตอร์จะเป็น 41 ด้วย

อินพุต IX

เอากัณฑ์ -

วีจิสเตอร์ IX

OUT1BSP

รหัส \$03

ทำหน้าที่ เหมือนกับ OUT1BYT แต่จะส่งช่องว่าง (Space) ตามลงไปด้วย

อินพุต IX

เอากัณฑ์ -

วีจิสเตอร์ IX

OUT2BSP

รหัส \$04

ทำหน้าที่ เหมือนกับ OUT1BYT แต่จะส่งข้อมูล 2 ไบต์ แล้วส่งช่องว่าง (Space) ตามออกไปด้วย

อินพุต IX

เอากัณฑ์ -

วีจิสเตอร์ IX

OUTSTRGO

รหัส \$05

ทำหน้าที่ ส่งข้อมูลที่วีจิสเตอร์ IX ที่อยู่ซึ่งเป็นรหัส ASCII ออกไปที่ SCI จนกระทั่งเจอรหัส \$04 จึงจะหยุดส่ง ค่าของวีจิสเตอร์ IX จะมีค่าเท่ากับแอดเดรสของข้อมูล \$04

อินพุต IX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอากัด -
 รีจิสเตอร์ IX

OUTSTRG

รหัส \$06

ทำหน้าที่ จะเหมือนกับ OUTSTRGO แต่จะส่ง Carriage return และ Line feed ออกไปก่อน

อินพุต IX

เอากัด -
 รีจิสเตอร์ IX

OUTCRLF

รหัส \$07

ทำหน้าที่ ส่ง carriage return และ line feed ออกไปที่ SCI

อินพุต -

เอากัด -
 รีจิสเตอร์ -

INPUT

รหัส \$08

ทำหน้าที่ รับข้อมูลจากคีย์บอร์ดเป็นรหัส ASCII แล้วไปเก็บไว้ที่รีจิสเตอร์ A เช่น ถ้ามีการกดคีย์ A ก็จะได้ค่า \$41 ในรีจิสเตอร์ A แต่ถ้าไม่มีการกดคีย์ ค่าของรีจิสเตอร์ A จะเป็น \$00

อินพุต -

เอากัด A
 รีจิสเตอร์ -

INININ

รหัส \$09

ทำหน้าที่ รอรับข้อมูลจากคีย์บอร์ดจนกระทั่งมีการกดคีย์ ข้อมูลที่รับเข้ามานั้นจะเป็นรหัส ASCII และจะถูกเก็บไว้ที่รีจิสเตอร์ A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขที่ชุด A
รีจิสเตอร์ -

INCHAR

รหัส \$0A

ทำหน้าที่ เหมือนกับ INININ แต่จะส่งค่า ASCII ของตัวอักษรตัวนั้นจะไปปรากฏที่จอ เครื่องคอมพิวเตอร์ด้วย

อินพุต -

เลขที่ชุด A

รีจิสเตอร์ -

INIT_LCD

รหัส \$0b

ทำหน้าที่ ตั้งค่าเริ่มต้นของแอลซีดีแบบ DOT MATRIX โดยเซตให้ D=1, C=1, B=1, DL=1, N=1, F=0, I/D=1, S=0 และเคลียร์จอแสดงผลพร้อมจะรับข้อมูล

อินพุต -

เลขที่ชุด -

รีจิสเตอร์ -

CLEAR_L

รหัส \$0C

ทำหน้าที่ เคลียร์จอแสดงผลของแอลซีดี

อินพุต -

เลขที่ชุด -

รีจิสเตอร์ -

CURSOR_L

รหัส \$0d

ทำหน้าที่ เลื่อนเคอร์เซอร์ไปตำแหน่งที่ต้องการซึ่งเป็นการเซตไดนามิกแรม แอดเดรสของจอแอลซีดีแต่ละแบบจะไม่เท่ากัน ให้ดูในคู่มือแอลซีดีนั้นด้วย

อินพุต A

เลขที่ชุด -

รีจิสเตอร์ -

OUT_L

รหัส	\$0E
ทำหน้าที่	ส่งข้อมูล ASCII ที่อยู่ในรีจิสเตอร์ A ออกบนจอแอลซีดี
อินพุต	A
เอาท์พุต	-
รีจิสเตอร์	-

OUTH_L

รหัส	\$0F
ทำหน้าที่	ส่งข้อมูลที่อยู่ในรีจิสเตอร์ A ที่เป็นเลขเลขฐาน 16 ออกจอแอลซีดี
อินพุต	A
เอาท์พุต	-
รีจิสเตอร์	-

01BYT_L

รหัส	\$10
ทำหน้าที่	นำค่าตัวเลขฐาน 16 ที่รีจิสเตอร์ IX ที่อยู่ขนาด 1 ไบต์ออกจอแอลซีดีและค่าของรีจิสเตอร์ IX จะชี้ไปยังไบต์ถัดไป
อินพุต	IX
เอาท์พุต	-
รีจิสเตอร์	IX

01BSP_L

รหัส	\$11
ทำหน้าที่	เหมือนกับ 01BYT_L แต่จะส่งช่องว่าง (space) ตามออกไปด้วย
อินพุต	IX
เอาท์พุต	-
รีจิสเตอร์	IX

02BSP_L

รหัส	\$12
ทำหน้าที่	เหมือนกับ 01BYT_L แต่จะส่งข้อมูล 2 ไบต์แล้วส่งช่องว่างตามออกไปด้วย
อินพุต	IX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขที่ชุด -
 รัจิสเตอร์ IX

OSTRGO_L

รหัส \$13
 ทำหน้าที่ ส่งข้อมูลที่รัจิสเตอร์ IX ที่อยู่ซึ่งเป็นรหัส ASCII ออกจอแอลซีดีจนเจกว่า
 จะเจอรหัสค่า \$04 จึงจะหยุดส่งและค่าของรัจิสเตอร์ IX จะมีค่าเท่ากับ
 แอดเดรสข้อมูล \$04

อินพุต IX
 เลขที่ชุด -
 รัจิสเตอร์ IX

SHIFTR_L

รหัส \$14
 ทำหน้าที่ เลื่อนเคอร์เซอร์ไปทางขวา 1 ตำแหน่ง

อินพุต -
 เลขที่ชุด -
 รัจิสเตอร์ -

SHIFTL_L

รหัส \$15
 ทำหน้าที่ เลื่อนเคอร์เซอร์ไปทางซ้าย 1 ตำแหน่ง

อินพุต -
 เลขที่ชุด -
 รัจิสเตอร์ -

RADDR_L

รหัส \$16
 ทำหน้าที่ อ่านแอดเดรสตำแหน่งที่เคอร์เซอร์ที่อยู่แอดเดรสที่อ่านได้จะอยู่ที่รัจิสเตอร์ A
 อินพุต -
 เลขที่ชุด A
 รัจิสเตอร์ -

OUT_P

รหัส	\$17
ทำหน้าที่	ส่งข้อมูล ASCII ที่อยู่ในรีจิสเตอร์ A ออกไปที่เครื่องพิมพ์
อินพุต	A
เอาต์พุต	-
รีจิสเตอร์	-

OUTH_P

รหัส	\$18
ทำหน้าที่	ส่งข้อมูลที่อยู่ในรีจิสเตอร์ A ที่เป็นเลขเลขฐาน 16 ออกเครื่องพิมพ์
อินพุต	A
เอาต์พุต	-
รีจิสเตอร์	-

O1BYT_P

รหัส	\$19
ทำหน้าที่	นำค่าเลขฐาน 16 ที่รีจิสเตอร์ IX ที่อยู่ขนาด 1 ไบต์ออกไปยังเครื่องพิมพ์ และค่าของ IX จะชี้ไปยังไบต์ถัดไป
อินพุต	IX
เอาต์พุต	-
รีจิสเตอร์	IX

O1BSP_P

รหัส	\$1A
ทำหน้าที่	เหมือนกับ O1BYT_P แต่จะส่งช่องว่างตามออกไปด้วย
อินพุต	IX
เอาต์พุต	-
รีจิสเตอร์	IX

O2BSP_P

รหัส	\$1B
ทำหน้าที่	เหมือนกับ O1BYT_P แต่จะส่งข้อมูล 2 ไบต์แล้วส่งช่องว่างออกไปด้วย
อินพุต	IX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฮกซ์คูด -
รีจิสเตอร์ IX

OSTRGO_P

รหัส \$1C

ทำหน้าที่ ส่งข้อมูลที่รีจิสเตอร์ IX ที่อยู่ซึ่งเป็นรหัส ASCII ออกเครื่องพิมพ์จนกระทั่งเจอรหัส \$04 จึงจะหยุดส่ง ค่าของรีจิสเตอร์ IX จะมีค่าเท่ากับแอดเดรสของข้อมูล \$04

อินพุต IX

เฮกซ์คูด -
รีจิสเตอร์ IX

OSTRG_P

รหัส \$1D

ทำหน้าที่ เหมือนกับ OSTRGO_P แต่จะส่ง carriage return และ line feed ออกไปก่อน

อินพุต IX

เฮกซ์คูด -
รีจิสเตอร์ IX

OCRLF_P

รหัส \$1E

ทำหน้าที่ ส่ง carriage return และ line feed ออกไปยังเครื่องพิมพ์

อินพุต -

เฮกซ์คูด -

รีจิสเตอร์ -

CGLCD

รหัส \$1F

ทำหน้าที่ ตั้งค่าเริ่มต้นของแอลซีดีแบบกราฟฟิกขนาด 128x64 แล้วเคลียร์หน้าจอของแอลซีดี หลังจากเรียกใช้โปรแกรมย่อยแล้วผู้ใช้ต้องกำหนดค่าแอดเดรส X- และกำหนดแอดเดรส Y- ของแอลซีดีขึ้นมาใหม่

อินพุต -

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอากัดชุด -
 รัจิสเตอร์ -

UPCASE

รหัส \$20
 ทำหน้าที่ เปลี่ยนรหัส ASCII จากตัวอักษรตัวเล็กให้เป็นตัวใหญ่
 อินพุต A
 เอากัดชุด A
 รัจิสเตอร์ -

HEXBIN

รหัส \$21
 ทำหน้าที่ เปลี่ยนรหัส ASCII ของเลข 0-9 และ A-F หรือ a-f เป็นเลขฐาน 16
 ข้อมูลที่ได้จากการเปลี่ยนจะอยู่ที่แอดเดรส \$2145-\$2146 ซึ่งมีลักษณะดังนี้

			1
--	--	--	---

1 คือ ข้อมูลที่ได้จากการเปลี่ยนค่าแล้วครั้งที่ 1

		1	2
--	--	---	---

2 คือ ข้อมูลที่ได้จากการเปลี่ยนค่าแล้วครั้งที่ 2

	1	2	3
--	---	---	---

3 คือ ข้อมูลที่ได้จากการเปลี่ยนค่าแล้วครั้งที่ 3

1	2	3	4
---	---	---	---

4 คือ ข้อมูลที่ได้จากการเปลี่ยนค่าแล้วครั้งที่ 4

\$2145 \$2146

อินพุต A
 เอากัดชุด แอดเดรส \$2145 - \$2146
 รัจิสเตอร์ -

HTOD

รหัส	\$22
ทำหน้าที่	เปลี่ยนเลขฐาน 16 ขนาด 16 บิตไปเป็นเลขฐาน 10 ขนาด 5 หลักซึ่งค่าที่ได้จะเป็นรหัส ASCII ข้อมูลที่จะเปลี่ยนค่าอยู่ที่แอดเดรส \$2147-\$2148 ส่วนค่าที่ได้จะอยู่ที่แอดเดรส \$214A-\$214E หลังจากที่ใช้โปรแกรมย่อยแล้ว ข้อมูลที่อยู่ในแอดเดรส \$2147-\$2148 จะเปลี่ยนไป
อินพุต	แอดเดรส \$2147-\$2148
เอาต์พุต	แอดเดรส \$214A-\$214E
รีจิสเตอร์	-

HGTOD8

รหัส	\$23
ทำหน้าที่	เปลี่ยนเลขฐาน 16 ขนาด 24 บิตไปเป็นเลขฐาน 10 ขนาด 8 หลักซึ่งค่าที่ได้จะเป็นรหัส ASCII ข้อมูลที่เปลี่ยนจะอยู่ที่แอดเดรส \$2147-\$2149 ส่วนค่าที่ได้จะอยู่ที่แอดเดรส \$214A-\$2151 หลังจากการใช้โปรแกรมย่อยนี้แล้ว ข้อมูลที่แอดเดรส \$2147-\$2149 จะเปลี่ยนไป
อินพุต	แอดเดรส \$2147-\$2149
เอาต์พุต	แอดเดรส \$214A-\$2151
รีจิสเตอร์	-

MAIN

รหัส	\$24
ทำหน้าที่	ให้โปรแกรมที่ผู้ใช้เขียนขึ้นกลับเข้าสู่โปรแกรมหลักของมัลติบอร์ด โดยแสดงเครื่องหมาย prompt และพร้อมที่จะรับคำสั่งอื่น ๆ ได้ตามปกติ
อินพุต	-
เอาต์พุต	-
รีจิสเตอร์	-

EWRITE

รหัส	\$25
ทำหน้าที่	เขียนข้อมูลลงอีพีรอมของ 68HC11 ถ้าข้อมูลเดิมไม่ใช่ค่า \$FF จะทำการลบข้อมูลเดิมให้ก่อนแล้วจึงเขียนข้อมูลใหม่ลงไป ค่าของรีจิสเตอร์ A คือข้อมูลที่เขียนลงไปและค่าของรีจิสเตอร์ IX คือแอดเดรสที่จะเขียนข้อมูลลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินท A, IX
 เอกภพ -
 วีธีเตอร์ A

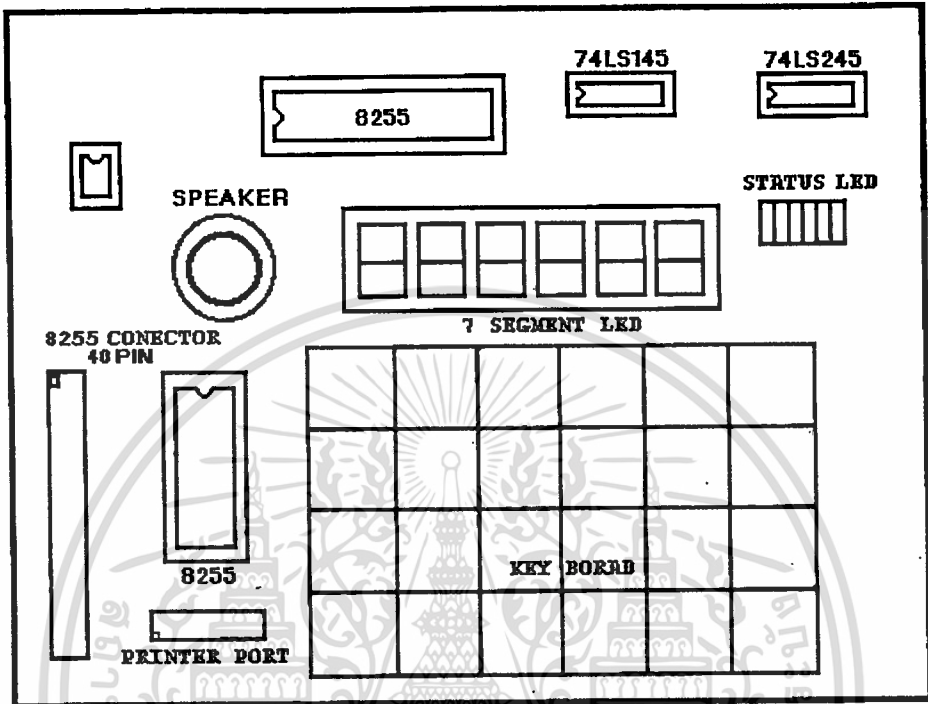


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

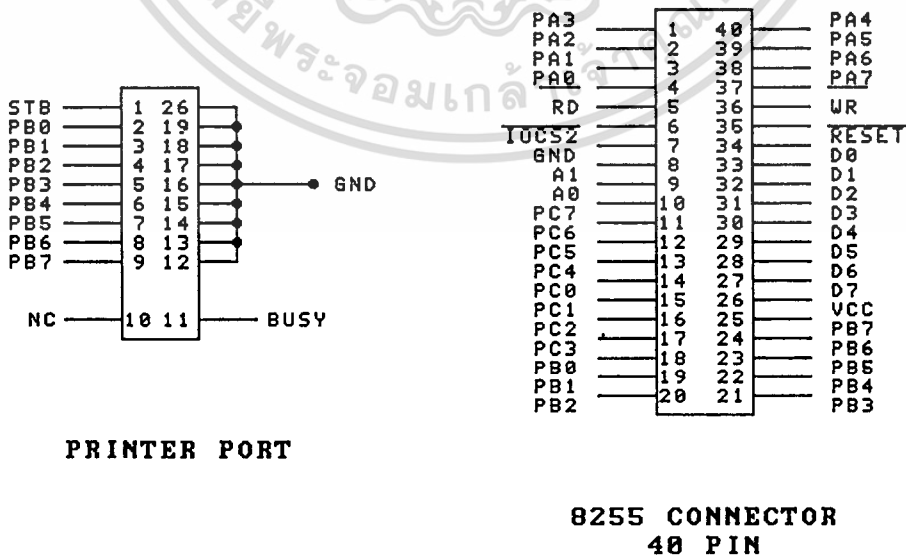


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการวางอุปกรณ์บอร์ดอินพุต/เอาต์พุต

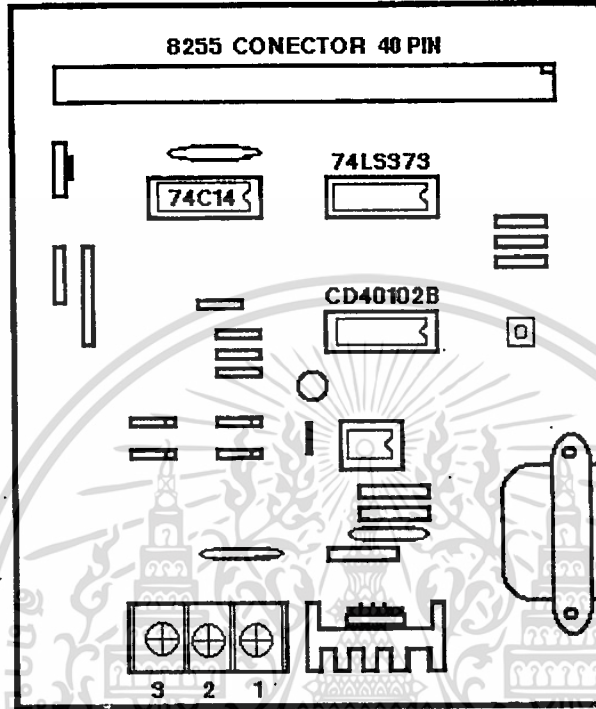


ลักษณะการวางขาคอนเน็คเตอร์บอร์ดอินพุต/เอาต์พุต

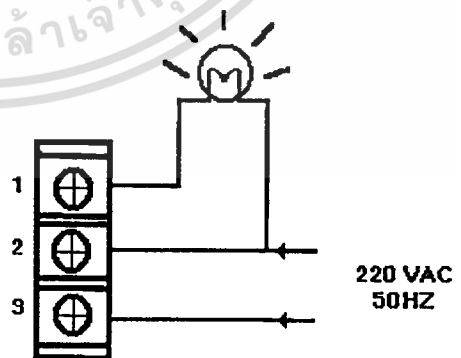
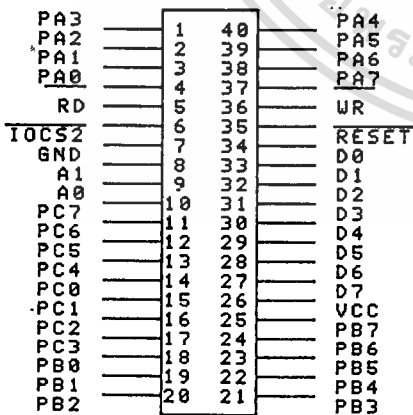


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการวางอุปกรณ์บอร์ดหรือไฟควบคุมด้วยไมโครคอนโทรลเลอร์



ลักษณะการจัดวางขาคอนเน็คเตอร์บอร์ดหรือไฟควบคุมด้วยไมโครคอนโทรลเลอร์

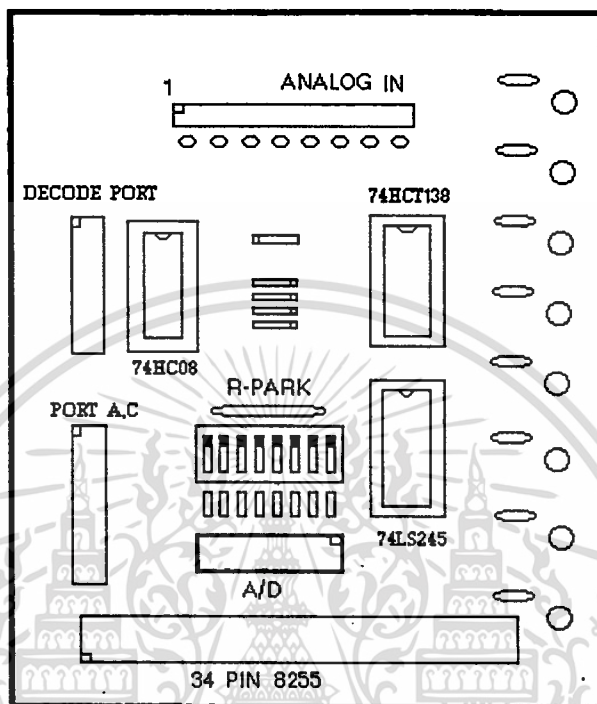


8255 CONNECTOR
40 PIN

รูปรายละเอียดของขาคอนเน็คเตอร์ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการวางอุปกรณ์บอร์ดคอมพิวเตอร์เฟส

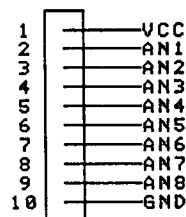
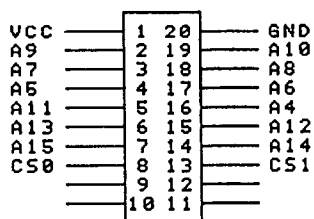


ลักษณะการจัดวางขาคอนเน็คเตอร์บอร์ดคอมพิวเตอร์เฟส



A-D CONNECTOR

PORT A,C CONNECTOR

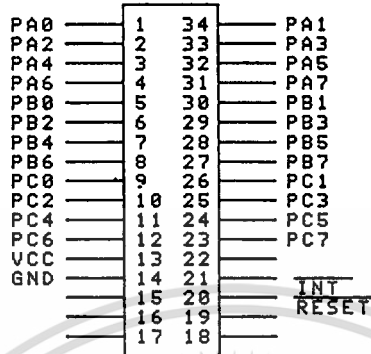


DECODE PORT CONNECTOR

ANALOG IN CONNECTOR

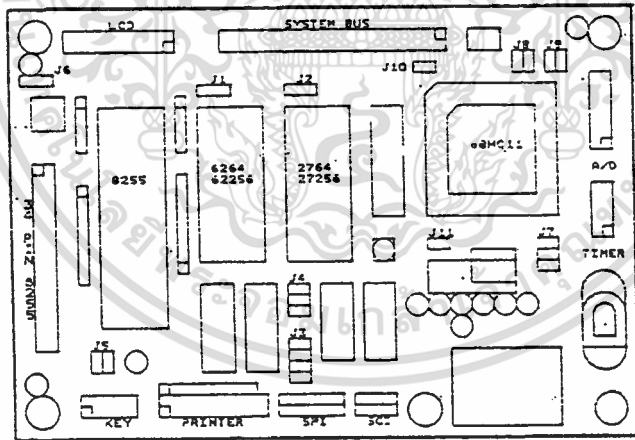
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการวางขาคอนเน็คเตอร์บอร์ดอินเตอร์เฟส (ต่อ)

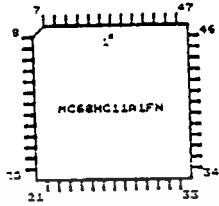


8255 CONNECTOR
34 PIN

ลักษณะการวางอุปกรณ์บอร์ดควบคุม

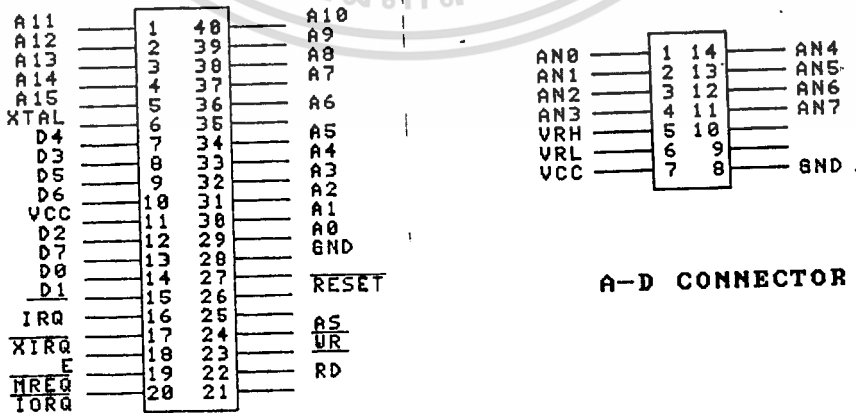
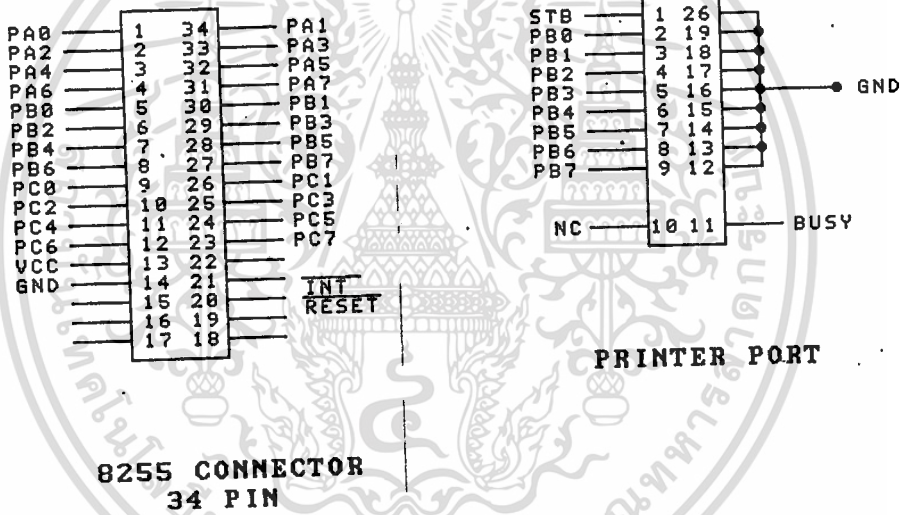


ลักษณะการจัดวางขาคอนเน็คเตอร์บอร์ดควบคุม



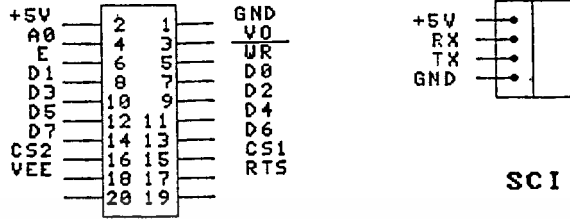
7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

SOCKET 52 PIN (TOP VIEW)

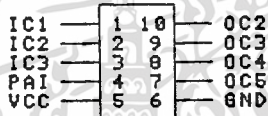


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

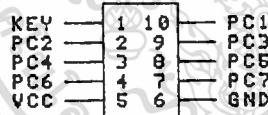
ลักษณะการจัดวางฮาร์ดแวร์บนบอร์ดควบคุม (ต่อ)



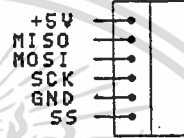
LCD



TIMER 68HC11

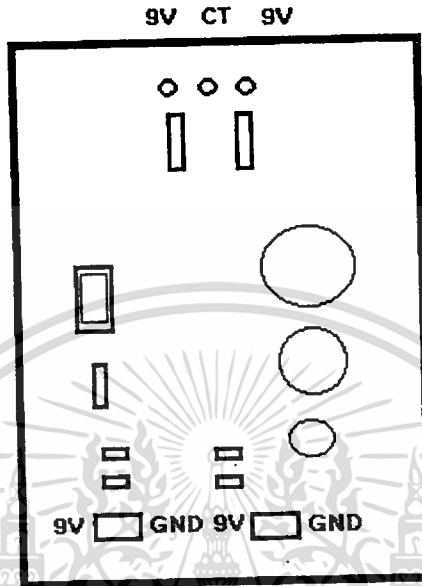


KEYBORAD 4X4 OR 3X4
8255



SPI

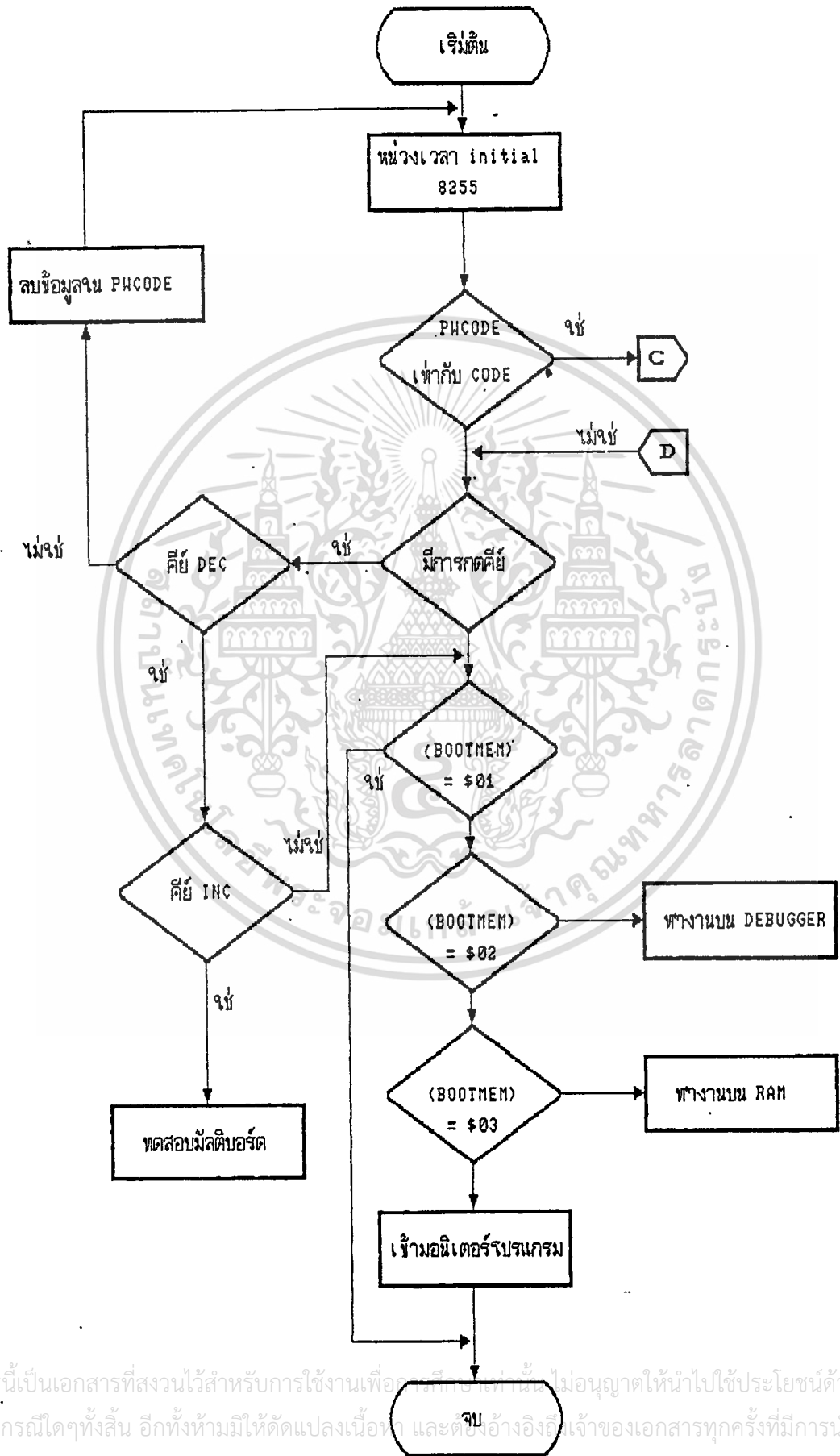
ลักษณะการวางอุปกรณ์บอร์ดเรดิไฟร์และฟิลเตอร์



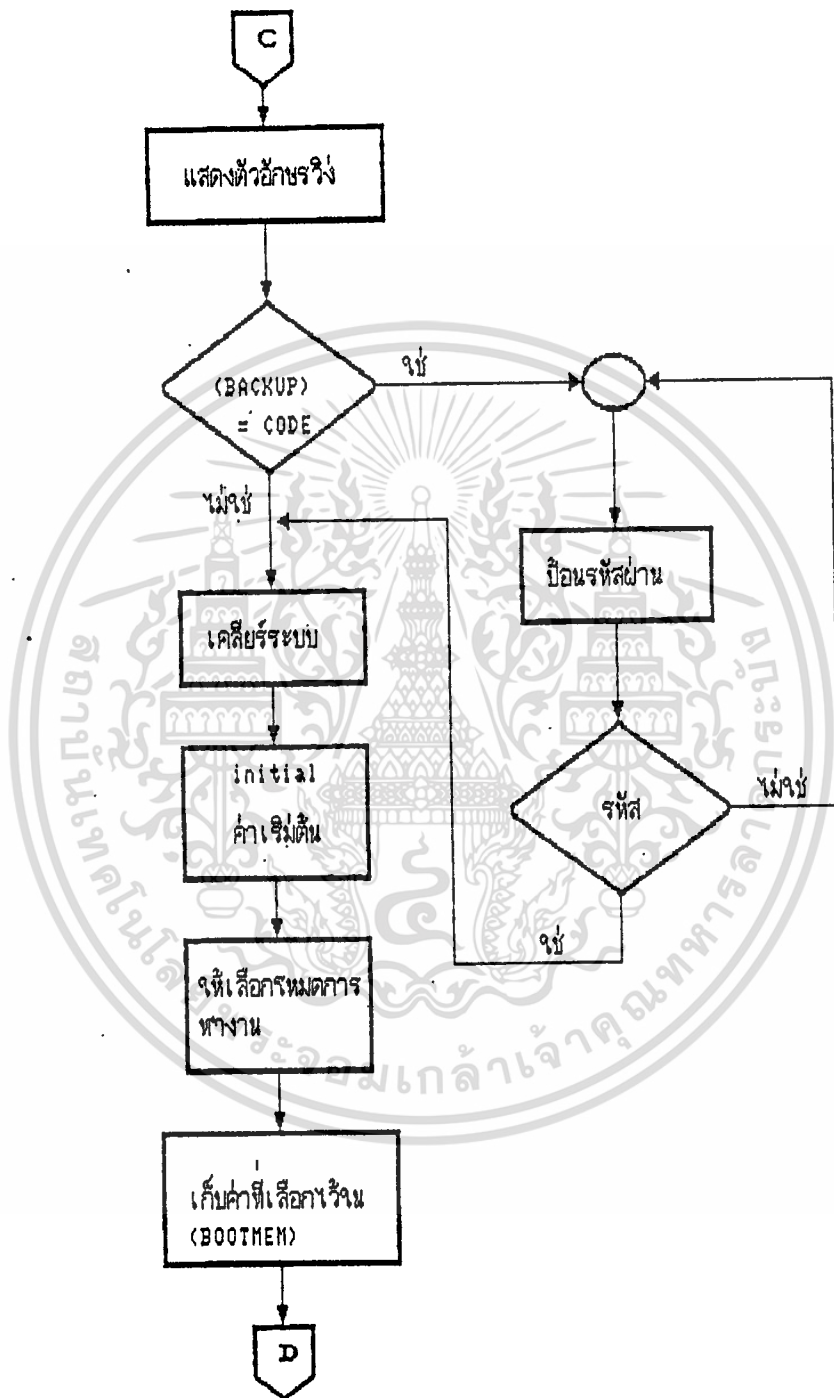
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

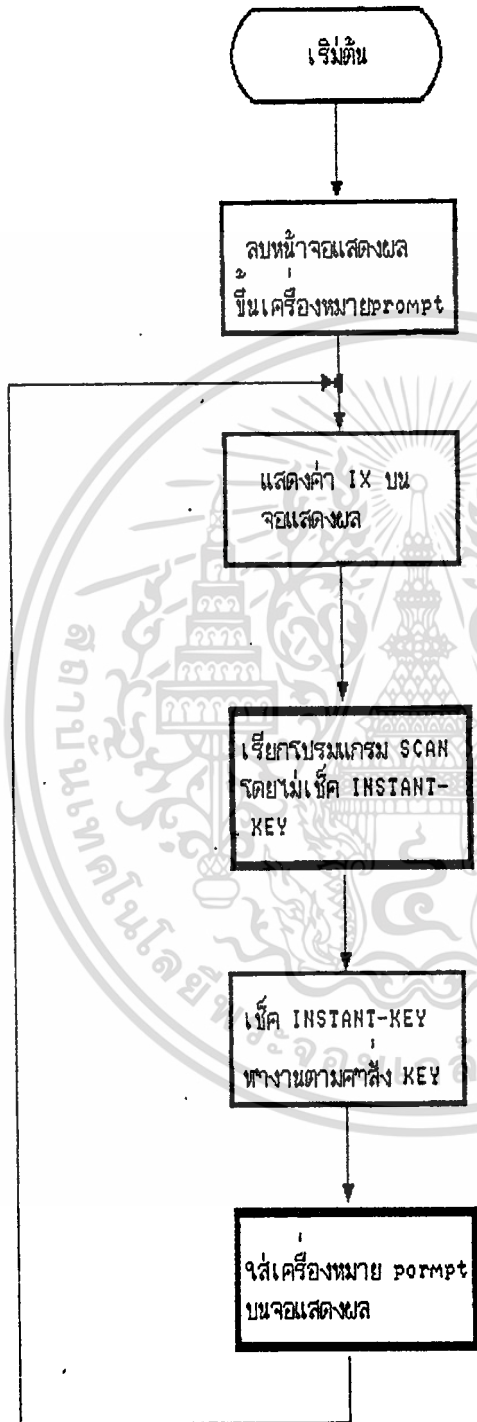


รูปลัหวารต์การทำงานของโปรแกรมขยายไฟงให้เครื่อง (ต่อ)



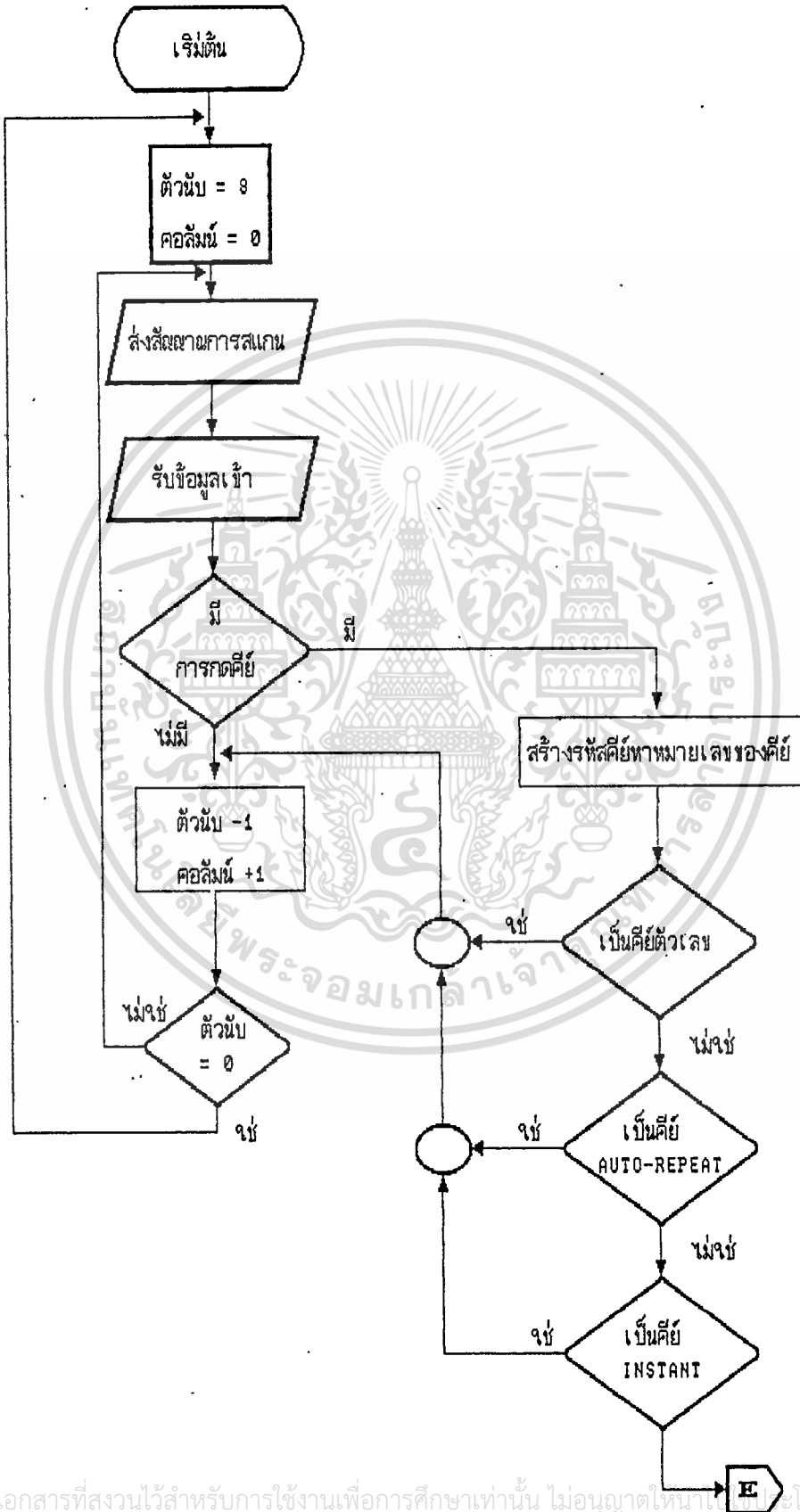
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปลัหวารต์การทางานของโปรแกรมหลัก

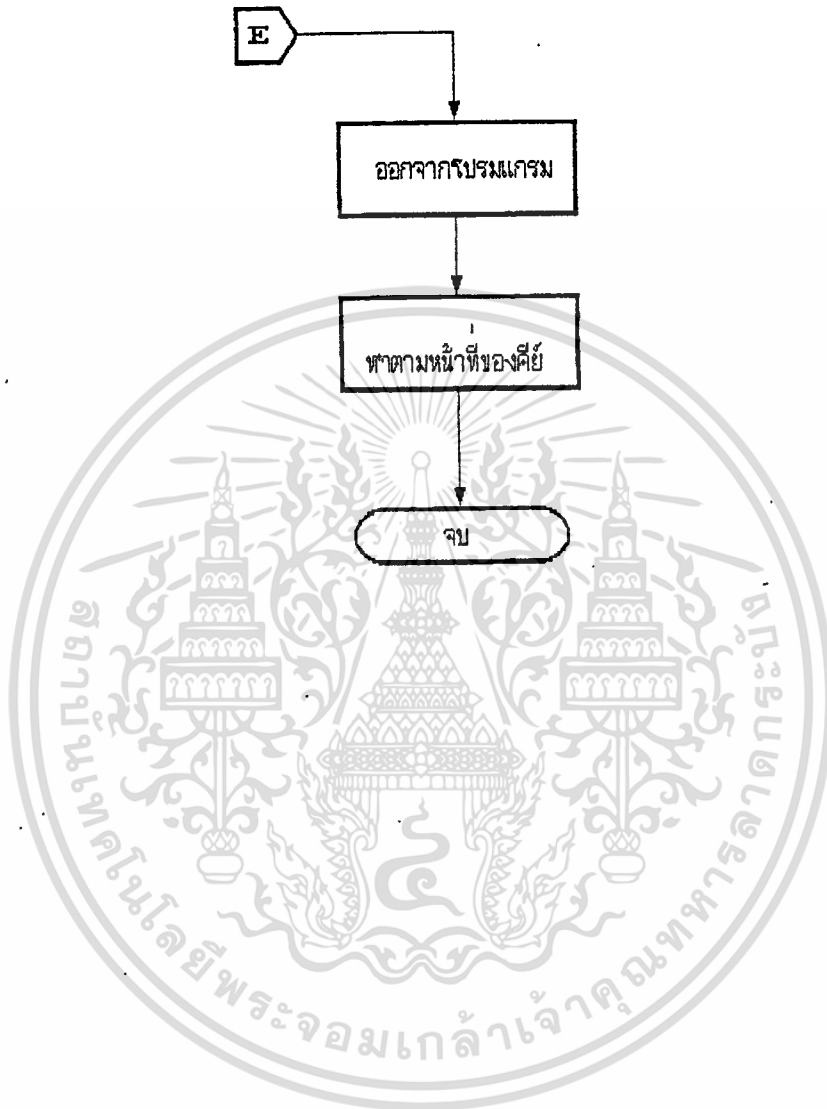


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รฟลวชาร์ตโปรแกรมสแกนคีย์บอร์ด

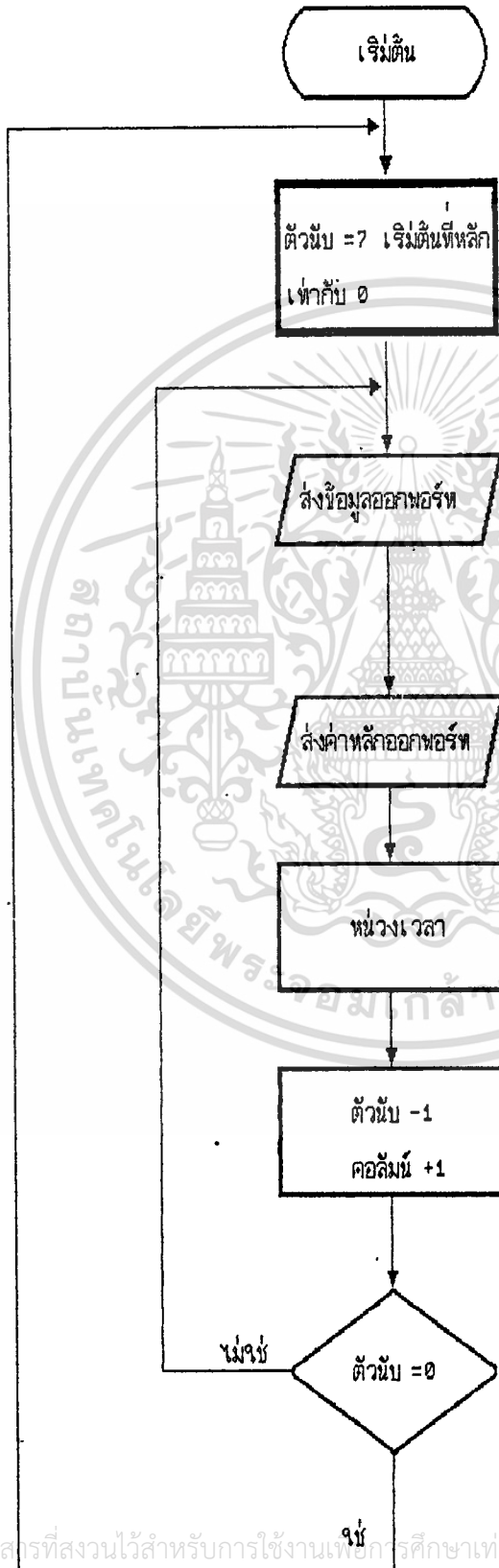


รหัสบาร์ตปรแกรมสแกนคิวบอर्ड (ต่อ)

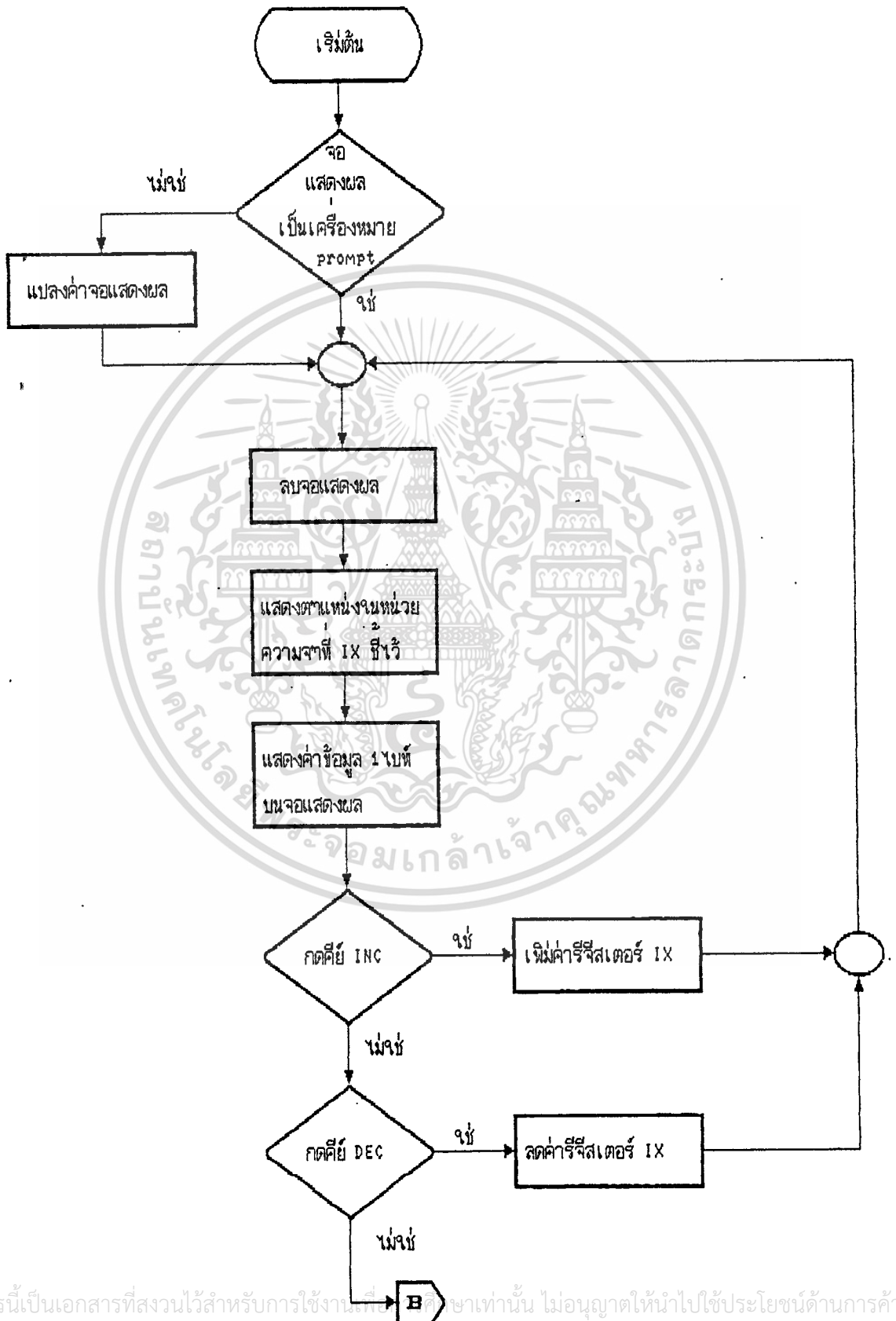


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสบาร์ตของโปรแกรม SCAN DISPLAY

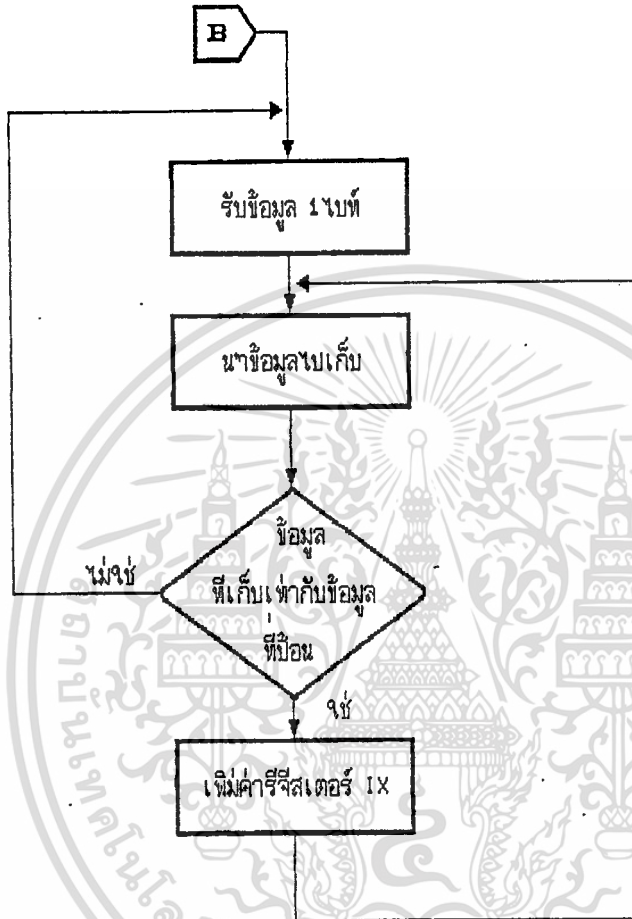


รหัสวาร์ตโปรแกรมขณะป้อน Machine Code

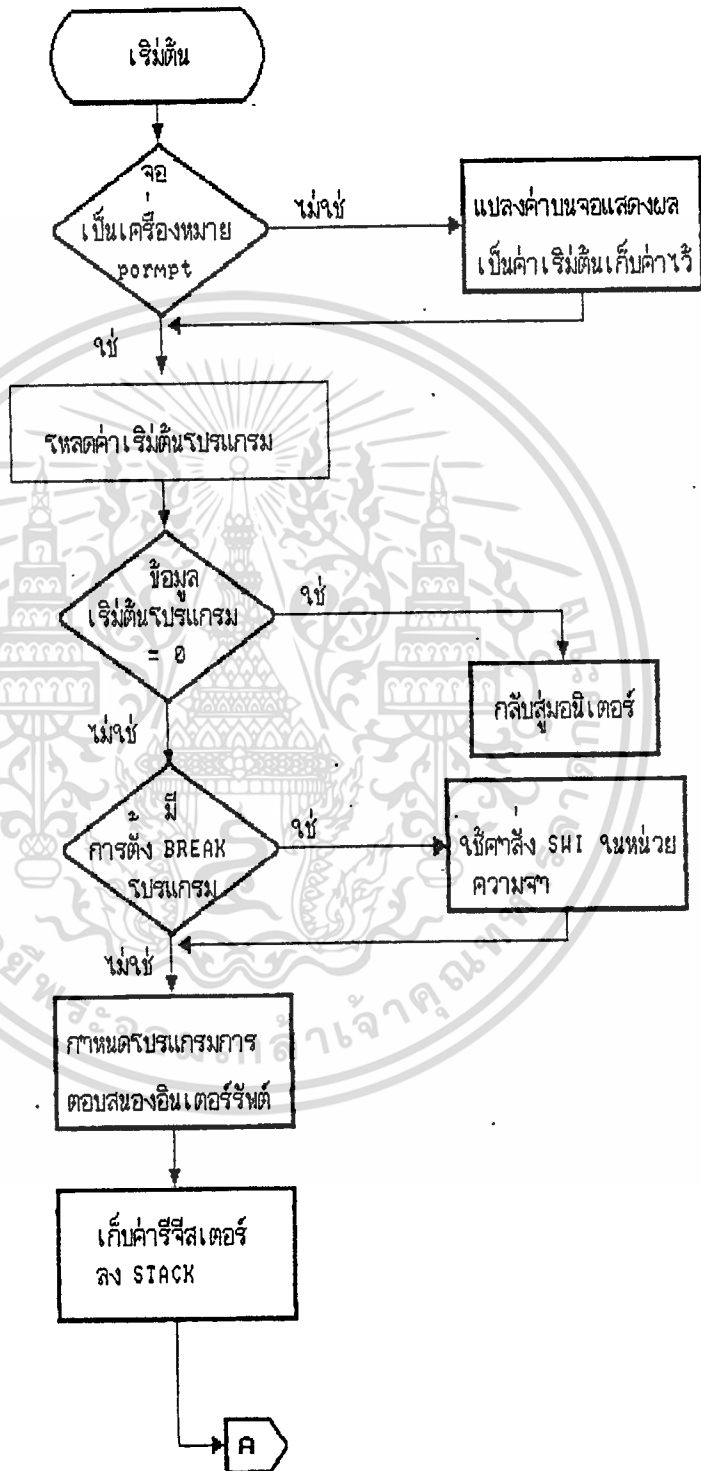


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสวาร์ตรับรแกรมขบะบ็อน Machine Code (ต่อบ)

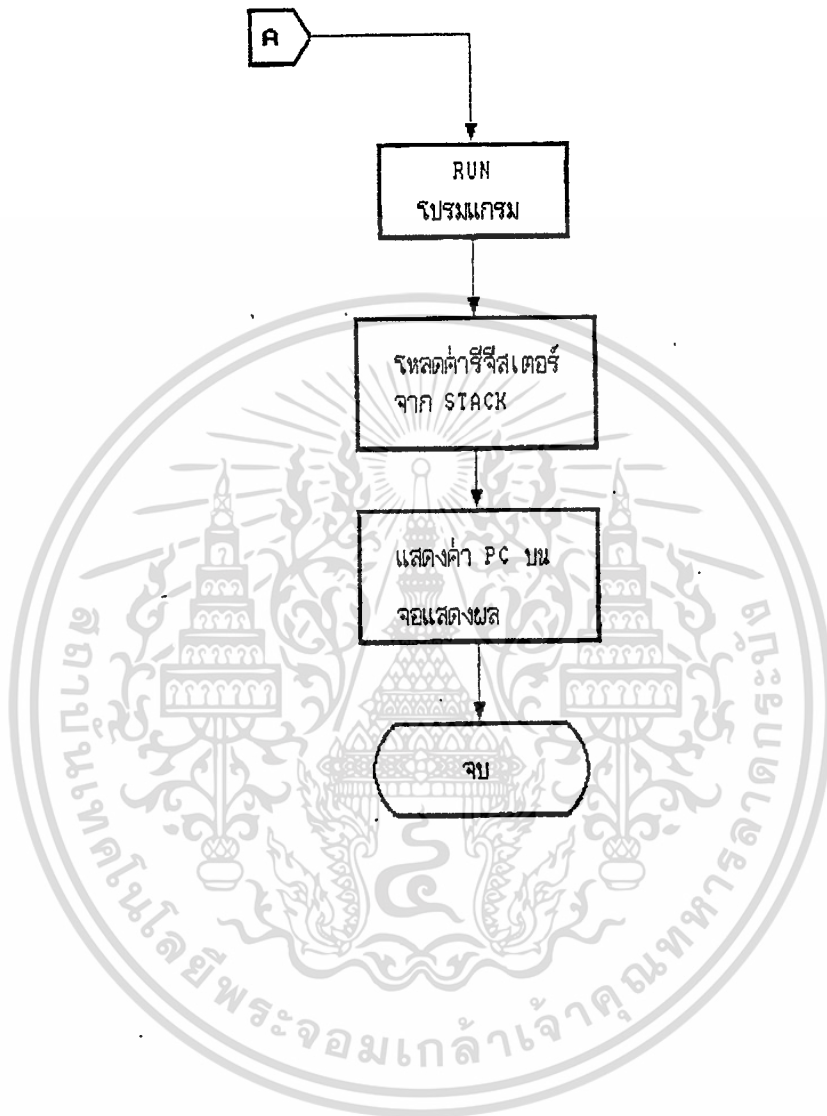


ฟลัชวาร์ตการ RUN โปรแกรม



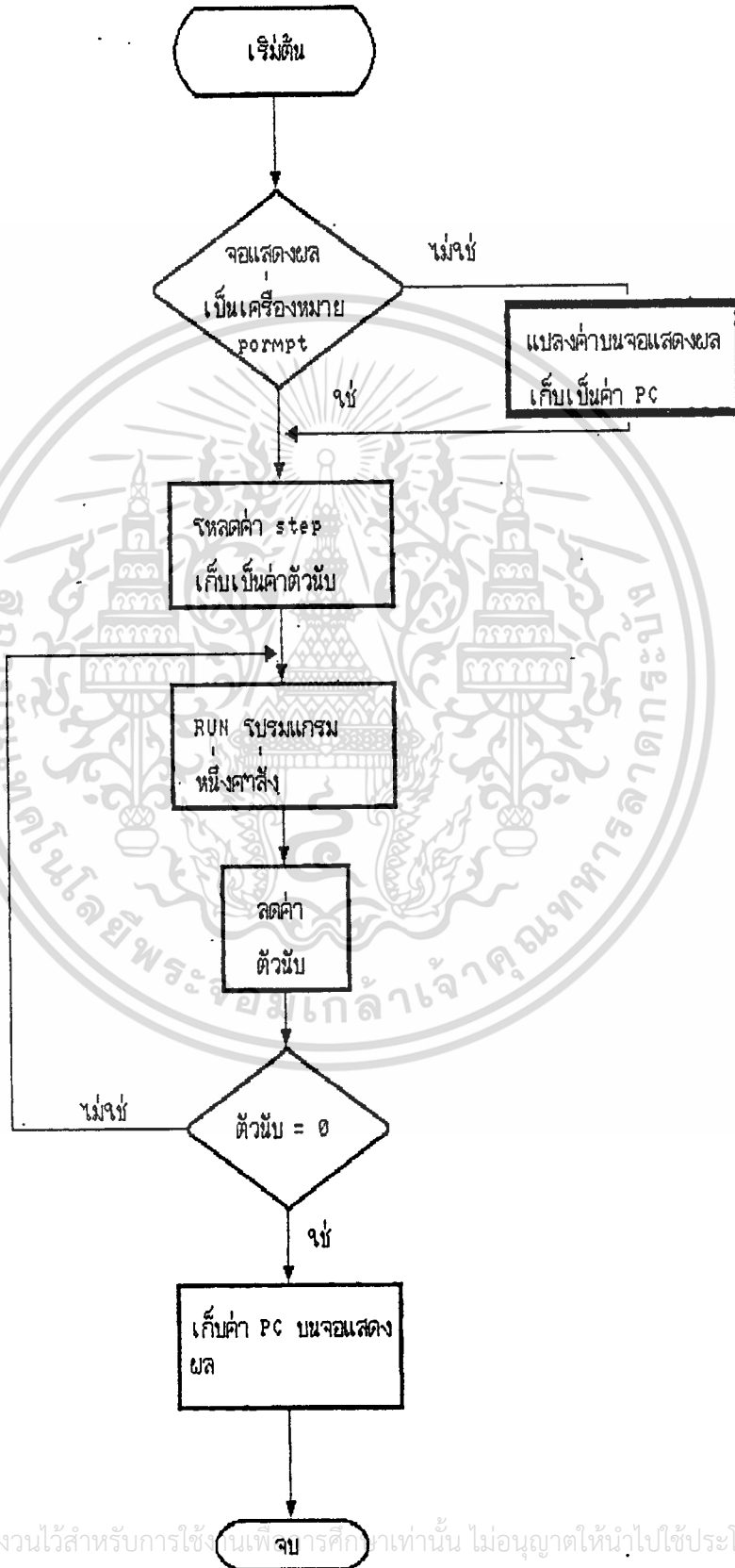
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสวิชา ๓๕๓๐๓๑๑ RUN โปรแกรม (ต่อ)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟลิวชาร์ตการ RUN แบบ TRACE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Freeware assembler ASxx.EXE Ver 1.03.

```

0001 *****
0002 *           MONITOR PROGRAM           *
0003 *       FOR   68HC11 MULTI BOARD       *
0004 *           MICRO COMPUTER             *
0005 *****
0006
0007 2000                               ORG   $2000
0008
0009 ***** INTERNAL RAM *****
0010
0011 2000          STRAM          RMB   30
0012 201e          USTACK        RMB   30
0013 203c          STACK         RMB   1
0014 203d          REGS          RMB  10
0015 2047          SP            RMB   2
0016 2049          PTR1         RMB   2
0017 204b          PTR2         RMB   2
0018 204d          TMP1         RMB   1
0019
0020 ***** VERSION 1.0 EXPANSION *****
0021
0022 204e          REGC          RMB   1
0023 204f          REGD          RMB   1
0024 2050          REGE          RMB   1
0025 2051          REGDS         RMB   1
0026 2052          REGES         RMB   1
0027 2053          REGH          RMB   1
0028 2054          REGL          RMB   1
0029 2055          PASDIG        RMB   2
0030 2057          CHKSUM        RMB   1
0031 2058          BACKUP        RMB   1
0032 2059          LENGTH        RMB   1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0033	205a	FIND_BUF	RMB	8
0034	2062	CHAN_BUF	RMB	8
0035	206a	BOOTMEM	RMB	1
0036	206b	BRKMEM	RMB	1
0037	206c	USRKEY	RMB	32
0038	208c	BC	RMB	2
0039	208e	HL	RMB	2
0040	2090	BRK	RMB	2
0041	2092	ADDR	RMB	2
0042	2094	START	RMB	2
0043	2096	FINAL	RMB	2
0044	2098	DEST	RMB	2
0045	209a	PORT	RMB	2
0046	209c	HEX	RMB	1
0047	209d	NSTEP	RMB	1
0048	209e	DISPY	RMB	8
0049	20a6	BUF	RMB	3
0050	20a9	CBUF	RMB	5
0051	20ae	MEMX	RMB	2
0052	20b0	MEMY	RMB	2
0053	20b2	KEYIN	RMB	1
0054	20b3	REPDLY	RMB	1
0055	20b4	SYSFAG	RMB	1
0056	20b5	PWCODE	RMB	1
0057	20b6	GAIN	RMB	1
0058	20b7	COUNT_L	RMB	1
0059	20b8	BUFF_T1	RMB	2
0060	20ba	BUF_PRN	RMB	1
0061	20bb	BUFF_ADC	RMB	4
0062	20bf	ENDRAM	RMB	1
0063				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0064 20c4          ORG  $20C4
0065
0066          ***** VECTOR JUMP TABLE *****
0067
0068 20c4          JSCI          RMB  3
0069 20c7          JSPI          RMB  3
0070 20ca          JPAIE        RMB  3
0071 20cd          JPAO         RMB  3
0072 20d0          JTOF         RMB  3
0073 20d3          JTOC5        RMB  3
0074 20d6          JTOC4        RMB  3
0075 20d9          JTOC3        RMB  3
0076 20dc          JTOC2        RMB  3
0077 20df          JTOC1        RMB  3
0078 20e2          JTIC3        RMB  3
0079 20e5          JTIC2        RMB  3
0080 20e8          JTIC1        RMB  3
0081 20eb          JRTI         RMB  3
0082 20ee          JIRQ         RMB  3
0083 20f1          JXIRQ        RMB  3
0084 20f4          JSWI         RMB  3
0085 20f7          JILLOP       RMB  3
0086 20fa          JCOP         RMB  3
0087 20fd          JCLM         RMB  3
0088
0089          **** ADDRESS USER PROGRAM IN RAM ****
0090
0091 2300          ORG  $2300
0092
0093 2300          RAM          RMB  1
0094
0095 9000          ORG  $9000

```

```

0096
0097 9000          END_PRAM      RMB      1
0098
0099 c000          ORG        $C000
0100
0101              ***** VARIABLE SET EQUATES *****
0102
0103 1700          ANALOG      EQU      $1700
0104 1701          SEGM        EQU      $1701
0105 1702          DIGIT       EQU      $1702
0106 1703          CTRL        EQU      $1703
0107 1711          PORT_DATA   EQU      $1711
0108 1712          PORT_STB    EQU      $1712
0109 1712          PORT_BUSY   EQU      $1712
0110 1713          CTRL1       EQU      $1713
0111 1200          PORTA       EQU      $1200
0112 1201          PORTB       EQU      $1201
0113 1202          PORTC       EQU      $1202
0114 1203          CTRL2       EQU      $1203
0115
0116 00a3          CODE        EQU      $A3
0117 0060          REP1        EQU      $60
0118 0016          REP2        EQU      $16
0119 0080          SLODLY     EQU      $80
0120 0030          MIDDLY     EQU      $30
0121 0015          FASDLY     EQU      $15
0122 0010          HIGFRE     EQU      $10
0123 0014          LOWFRE     EQU      $14
0124 0008          HBPFRE     EQU      $08
0125 0050          LBPFRE     EQU      $50
0126 00f0          DEGAIN     EQU      $F0
0127
0128 0000          RAMBS      EQU      $0000

```

0129	1000	REGBS	EQU	\$1000
0130	b600	STREE	EQU	\$B600
0131	b7ff	ENDEE	EQU	\$B7FF
0132	100a	PORTE	EQU	REGBS+\$0A
0133	100b	CFORC	EQU	REGBS+\$0B
0134	100e	TCNT	EQU	REGBS+\$0E
0135	0010	TIC1	EQU	\$10
0136	0018	TOC2	EQU	\$18
0137	101e	TOC5	EQU	REGBS+\$1E
0138	1020	TCTL1	EQU	REGBS+\$20
0139	0020	OTCTL1	EQU	\$20
0140	0021	TCTL2	EQU	\$21
0141	1022	TMSK1	EQU	REGBS+\$22
0142	0022	OTMSK1	EQU	\$22
0143	0023	TFLG1	EQU	\$23
0144	1024	TMSK2	EQU	REGBS+\$24
0145	102b	BAUD	EQU	REGBS+\$2B
0146	102c	SCCR1	EQU	REGBS+\$2C
0147	102d	SCCR2	EQU	REGBS+\$2D
0148	102e	SCSR	EQU	REGBS+\$2E
0149	102f	SCDAT	EQU	REGBS+\$2F
0150	0030	ADCTL	EQU	\$30
0151	0031	ADR1	EQU	\$31
0152	0032	ADR2	EQU	\$32
0153	0033	ADR3	EQU	\$33
0154	0034	ADR4	EQU	\$34
0155	1035	BPROT	EQU	REGBS+\$35
0156	1039	OPTION	EQU	REGBS+\$39
0157	103a	COPRST	EQU	REGBS+\$3A
0158	103b	PPROG	EQU	REGBS+\$3B
0159	103c	HPRIO	EQU	REGBS+\$3C
0160	103f	CONFIG	EQU	REGBS+\$3F
0161	003f	SWI	EQU	\$3F

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0162 1400          SET1          EQU    $1400
0163 1401          WR1LCD         EQU    $1401
0164 0004          EOT          EQU    $04
0165 ff00          CALL          EQU    $FF00
0166
0167              ***** POWER - UP *****
0168
0169 c000 5f          POWER          CLR B
0170 c001 5a          POWER1         DEC B
0171 c002 01          NOP
0172 c003 01          NOP
0173 c004 c1 00          CMPB    #$00
0174 c006 26 f9          BNE    POWER1
0175
0176              ***** SYSTEM RESET *****
0177
0178 c008 ce 60 00          RESET          LDX    $$6000
0179 c00b 09          RESETO         DEX
0180 c00c 01          NOP
0181 c00d 8c 00 00          CPX    $$0000
0182 c010 26 f9          BNE    RESETO
0183 c012 8e 20 3c          LDS    #STACK
0184 c015 86 88          LDAA   $$88
0185 c017 b7 17 03          STAA  CTRL
0186 c01a 86 91          LDAA   $$91
0187 c01c b7 17 13          STAA  CTRL1
0188 c01f 86 80          LDAA   $$80
0189 c021 b7 12 03          STAA  CTRL2
0190 c024 b6 20 b5          LDAA  PWCODE
0191 c027 81 a3          CMPA  #CODE
0192 c029 26 65          BNE   INIT
0193

```

```

0194          ***** BOOT & SELF TEST CHECK *****
0195
0196 c02b 86 07          RESET1          LDAA  #$07
0197 c02d b7 17 02          STAA  DIGIT
0198 c030 b6 17 02          LDAA  DIGIT
0199 c033 84 70          ANDA  #$70
0200 c035 81 70          CMPA  #$70
0201 c037 27 24          BEQ   RESET3
0202 c039 b7 20 4f          STAA  REGD
0203 c03c b6 17 02          RESET2          LDAA  DIGIT
0204 c03f 84 70          ANDA  #$70
0205 c041 81 70          CMPA  #$70
0206 c043 26 f7          BNE   RESET2
0207 c045 b6 20 4f          LDAA  REGD
0208 c048 81 60          CMPA  #$60
0209 c04a 26 0a          BNE   RESET22
0210 c04c 4f          CLRA
0211 c04d 4a          RESET21          DECA
0212 c04e 26 fd          BNE   RESET21
0213 c050 b7 20 b5          STAA  PWCODE
0214 c053 7e c0 00          JMP   POWER
0215 c056 81 50          RESET22          CMPA  #$50
0216 c058 26 03          BNE   RESET3
0217 c05a 7e c2 0a          JMP   STEST
0218
0219          ***** GOTO  MAIN PROGRAM *****
0220
0221 c05d b6 20 6a          RESET3          LDAA  BOOTMEM
0222 c060 81 01          CMPA  #$01
0223 c062 26 09          BNE   RESET31
0224 c064 ce d6 5f          LDX   #BOARD_TAB
0225 c067 bd cf bd          JSR   ROW_LCD
0226 c06a 7e c2 aa          JMP   MON

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0227 c06d 81 02          RESET31      CMPA  #$02
0228 c06f 26 09          BNE   RESET32
0229 c071 ce d6 6f          LDX  #PC_TAB
0230 c074 bd cf bd          JSR  ROW_LCD
0231 c077 7e a0 0e          JMP  #$A00E
0232 c07a 81 03          RESET32      CMPA  #$03
0233 c07c 26 0f          BNE   RESET33
0234 c07e ce d6 7f          LDX  #RAM_TAB
0235 c081 bd cf bd          JSR  ROW_LCD
0236 e084 7e 23 00          JMP  #$2300
0237 c087 ce d6 5f          LDX  #BOARD_TAB
0238 c08a bd cf bd          JSR  ROW_LCD
0239 c08d 7e c2 aa          RESET33      JMP  MON
0240
0241          ***** POWER - UP *****
0242          *   DISPLAY ID.ED. VERSION 1.0   *
0243
0244 c090 c6 00          INIT          LDAB  #$00
0245 c092 bd d3 df          JSR  DELAY
0246 e095 c6 36          LDAB  #$36
0247 c097 f7 20 4e          STAB  REGC
0248 c09a c6 ff          LDAB  #$FF
0249 c09c ce 40 00          LDX  #$4000
0250 c09f bd d3 2d          JSR  SOUND
0251 c0a2 c6 24          LDAB  #$24
0252 c0a4 f7 20 4e          STAB  REGC
0253 c0a7 c6 ff          LDAB  #$FF
0254 c0a9 ce 70 00          LDX  #$7000
0255 c0ac bd d3 2d          JSR  SOUND
0256 c0af ce d6 8f          LDX  #MULTI_TAB
0257 c0b2 bd cf bd          JSR  ROW_LCD
0258 c0b5 bd d3 d2          JSR  CLEAR
0259 c0b8 ce d5 1e          LDX  #HELTAB

```

0260	c0bb c6 06		LDAB	#\$06
0261	c0bd f7 20 4e		STAB	REGC
0262	c0c0 e6 00	INIT1	LDAB	0,X
0263	c0c2 f7 20 4f		STAB	REGD
0264	c0c5 bd c1 ed		JSR	INITS
0265	c0c8 08		INX	
0266	c0c9 7a 20 4e		DEC	REGC
0267	c0cc 26 f2		BNE	INIT1
0268	c0ce c6 80		LDAB	#\$SLODLY
0269	c0d0 bd d3 6f		JSR	SCAND
0270	c0d3 c6 06		LDAB	#\$06
0271	c0d5 f7 20 4e		STAB	REGC
0272	c0d8 c6 00	INIT2	LDAB	#\$00
0273	c0da f7 20 4f		STAB	REGD
0274	c0dd bd c1 ed		JSR	INITS
0275	c0e0 7a 20 4e		DEC	REGC
0276	c0e3 26 f3		BNE	INIT2
0277				
0278				
0279		****	CHECKSUM & PASS DIGIT	CHECK ****
0280				
0281	c0e5 b6 20 58		LDAA	BACKUP
0282	c0e8 81 a3		CMPA	#\$CODE
0283	c0ea 26 3e		BNE	INIT8
0284	c0ec ce 20 b4		LDX	#\$SYSFAG
0285	c0ef 1d 00 08		BCLR	0,X \$08
0286	c0f2 bd d3 fd		JSR	CHKSR
0287	c0f5 b6 20 57		LDAA	CHKSUM
0288	c0f8 b1 20 4f		CMPA	REGD
0289	c0fb 27 05		BEQ	INIT4
0290	c0fd bd d4 2c		JSR	LBEEP
0291	c100 20 03		BRA	INIT5
0292	c102 bd d4 33	INIT4	JSR	HBEEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0293 c105 fe 20 55      INIT5      LDX  PASDIG
0294 c108 8c 00 00      CPX   #$0000
0295 c10b 27 1d          BEQ   INIT8
0296 c10d ce 20 b4      LDX  #SYSFAG
0297 c110 1c 00 10      BSET 0,X $10
0298 c113 c6 73          INIT6      LDAB  #$73
0299 c115 ce 00 00      LDX  #$0000
0300 c118 bd d2 7b      JSR  IN4
0301 c11b 4f            CLRA
0302 c11c bc 20 55      CPX  PASDIG
0303 c11f 26 f2          BNE  INIT6
0304 c121 b6 17 02      INIT7      LDAA  DIGIT
0305 c124 84 70          ANDA  #$70
0306 c126 81 70          CMPA  #$70
0307 c128 26 f7          BNE  INIT7
0308
0309      ***** CLEAR WORKING AREA *****
0310
0311 c12a ce 20 00      INIT8      LDX  #STRAM
0312 c12d 4f            CLRA
0313 c12e a7 00      INIT81     STAA 0,X
0314 c130 08            INX
0315 c131 8c 20 c0      CPX  #ENDRAM+1
0316 c134 26 f8          BNE  INIT81
0317
0318      ***** SET  POWER-UP DEFAULT *****
0319
0320 c136 86 f0          LDAA  #DEGAIN
0321 c138 b7 20 b6      STAA  GAIN
0322 c13b ce 23 00      LDX  #RAM
0323 c13e ff 20 92      STX  ADDR
0324 c141 86 01          LDAA  #$01
0325 c143 b7 20 9d      STAA  NSTEP

```

0326 c146 ce 20 1e	LDX #USTACK
0327 c149 ff 20 47	STX SP
0328 c14c 86 93	LDAA ##93
0329 c14e b7 10 39	STAA OPTION
0330 c151 86 00	LDAA #\$00
0331 c153 b7 10 24	STAA TMSK2
0332 c156 86 00	LDAA #\$00
0333 c158 b7 10 35	STAA BPROT
0334 c15b b6 10 20	LDAA TCTL1
0335 c15e 8a 03	ORAA #\$03
0336 c160 b7 10 20	STAA TCTL1
0337 c163 86 d0	LDAA #\$D0
0338 c165 b7 20 46	STAA REGS+9
0339 c168 ce 23 00	LDX #RAM
0340 c16b ff 20 3d	STX REGS
0341	
0342	**** CHECK AUTO START & DECISION ****
0343	
0344 c16e bd d3 d2	JSR CLEAR
0345 c171 ce 6d 40	LDX ##6D40
0346 c174 ff 20 9e	STX DISPY
0347 c177 ce 20 a0	LDX #DISPY+2
0348 c17a 86 3f	LDAA #\$3F
0349 c17c a7 00	STAA 0,X
0350 c17e 08	INX
0351 c17f b6 df ff	LDAA \$DFFF
0352 c182 81 a3	CMPA #CODE
0353 c184 26 05	BNE INIT9
0354 c186 86 06	LDAA #\$06
0355 c188 a7 00	STAA 0,X
0356 c18a 08	INX
0357 c18b b6 df fe	LDAA \$DFFE
0358 c18e 81 a3	CMPA #CODE

0359	c190 26 05		BNE	INIT91
0360	c192 86 5b		LDA	#\$5B
0361	c194 a7 00		STAA	0,X
0362	c196 08		INX	
0363	c197 b6 21 ff	INIT91	LDA	\$21FF
0364	c19a 81 a3		CMPA	#CODE
0365	c19c 26 04		BNE	INIT92
0366	c19e 86 4f		LDA	#\$4F
0367	c1a0 a7 00		STAA	0,X
0368	c1a2 86 a1	INIT92	LDA	#DISPY+3
0369	c1a4 81 00		CMPA	#\$00
0370	c1a6 27 32		BEQ	INITB
0371				
0372	c1a8 ce 02 00	INITA	LDX	#\$0200
0373	c1ab 3c	INITA1	PSHX	
0374	c1ac bd d3 8f		JSR	SCANK
0375	c1af 38		PULX	
0376	c1b0 09		DEX	
0377	c1b1 8c 00 00		CPX	#\$0000
0378	c1b4 27 2c		BEQ	INITC
0379	c1b6 b6 20 b2		LDA	KEYIN
0380	c1b9 81 ff		CMPA	#\$FF
0381	c1bb 27 ee		BEQ	INITA1
0382	c1bd bd d1 e7		JSR	UNPKT
0383	c1c0 ce 20 a0		LDX	#DISPY+2
0384	c1c3 c6 04		LDAB	#\$04
0385	c1c5 a1 00	INITA2	CMPA	0,X
0386	c1c7 27 06		BEQ	INITA3
0387	c1c9 08		INX	
0388	c1ca 5a		DECB	
0389	c1cb 26 f8		BNE	INITA2
0390	c1cd 20 d9		BRA	INITA

0391
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0392 c1cf b6 20 b2          INITA3          LDAA  KEYIN
0393 c1d2 b7 20 6a          STAA  BOOTMEM
0394 c1d5 c6 00            LDAB  #$00
0395 c1d7 bd d3 df          JSR   DELAY
0396
0397 c1da 86 a3            INITB          LDAA  #CODE
0398 c1dc b7 20 b5          STAA  PWCODE
0399 c1df 7e c0 2b          JMP   RESET1
0400
0401                      ***** AUTO START (DELAY PASS) *****
0402
0403 c1e2 b6 20 a1          INITC          LDAA  DISPY+3
0404 c1e5 bd d1 a7          JSR   PACKT
0405 c1e8 b7 20 6a          STAA  BOOTMEM
0406 c1eb 20 ed            BRA   INITB
0407
0408                      ***** DISPLAY SHIFT SUB *****
0409
0410 c1ed 3c                INITS          PSHX
0411 c1ee ce 20 a3          LDX  #DISPY+5
0412 c1f1 c6 06            LDAB  #$06
0413 c1f3 a6 00            INITS1         LDAA  0,X
0414 c1f5 36                PSHA
0415 c1f6 b6 20 4f          LDAA  REGD
0416 c1f9 a7 00            STAA  0,X
0417 c1fb 32                PULA
0418 c1fc b7 20 4f          STAA  REGD
0419 c1ff 09                DEX
0420 c200 5a                DECB
0421 c201 26 f0            BNE  INITS1
0422 c203 c6 10            LDAB  #$10
0423 c205 bd d3 6f          JSR   SCAND
0424 c208 38                PULX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0425 c209 39                RTS
0426
0427                ***** SELF TEST *****
0428
0429 c20a b6 20 b5        STEST        LDAA  PWCODE
0430 c20d 81 a3                CMPA  #CODE
0431 c20f 27 0b                BEQ   STEST1
0432 c211 86 f0                LDAA  #DEGAIN
0433 c213 b7 20 b6        STAA  GAIN
0434 c216 ce 20 b4        LDX   #SYSFAG
0435 c219 1d 00 08        BCLR  0,X #08
0436
0437                ***** DISPLAY MONTH, YEAR, LOT *****
0438
0439 c21c bd d3 d2        STEST1        JSR   CLEAR
0440 c21f ce 20 b4        LDX   #SYSFAG
0441 c222 1c 00 10        BSET  0,X #10
0442 c225 86 94                LDAA  ##94
0443 c227 b7 20 a6        STAA  BUF
0444 c22a 86 07                LDAA  ##07
0445 c22c b7 20 a7        STAA  BUF+1
0446 c22f 86 01                LDAA  #01
0447 c231 b7 20 a8        STAA  BUF+2
0448 c234 bd d1 b6        JSR   UNPK
0449 c237 c6 80                LDAB  #SLODLY
0450 c239 bd d3 6f        JSR   SCAND
0451
0452                ***** KEY BOARD TEST *****
0453
0454 c23c ce 17 00                LDX  ##1700
0455 c23f ff 20 a6        STEST2        STX   BUF
0456 c242 3c                PSHX
0457 c243 bd d1 b6        JSR   UNPK

```

0458 c246 ce 00 00		LDX #0000
0459 c249 ff 20 9e		STX DISPY
0460 c24c ff 20 a2		STX DISPY+4
0461 c24f bd d0 50		JSR SCAN
0462 c252 38		PULX
0463 c253 36		PSHA
0464 c254 37		PSHB
0465 c255 8f		XGDX
0466 c256 b7 20 53		STAA REGH
0467 c259 f7 20 54		STAB REGL
0468 c25c 33		PULB
0469 c25d 32		PULA
0470 c25e b1 20 54		CMPA REGL
0471 c261 27 03		BEQ STEST21
0472 c263 7e d4 08		JMP ERRST
0473 c266 7c 20 54	STEST21	INC REGL
0474 c269 7a 20 53		DEC REGH
0475 c26c b6 20 53		LDAH REGH
0476 c26f f6 20 54		LDAB REGL
0477 c272 8f		XGDX
0478 c273 f6 20 53		LDAB REGH
0479 c276 c1 00		CMPB #00
0480 c278 26 c5		BNE STEST2
0481		
0482	*****	DISPLAY TEST *****
0483		
0484 c27a ce 20 9e		LDX #DISPY
0485 c27d c6 ff		LDAB #\$FF
0486 c27f f7 20 4e		STAB REGC
0487 c282 c6 07		LDAB #\$07
0488 c284 37	STEST3	PSHB
0489 c285 f6 20 4e		LDAB REGC
0490 c288 e7 00		STAB 0,X

```

0491 c28a 33          PULB
0492 c28b 08          INX
0493 c28c 5a          DECB
0494 c28d 26 f5      BNE STEST3
0495 c28f bd d0 50   JSR SCAN
0496
0497          ***** TEST RAM *****
0498
0499 c292 bd d3 d2   JSR CLEAR
0500 c295 ce 23 00   LDX #RAM
0501 c298 ff 20 8e   STX HL
0502 c29b 18 ce 90 00 LDY #END_PRAM
0503 c29f 18 ff 20 8c STY BC
0504 c2a3 bd ce 60   JSR CHKRAM
0505 c2a6 bd d4 33   JSR HBEEP
0506 c2a9 3e          WAI
0507
0508          ***** MON (RESET) *****
0509
0510 c2aa bd d3 d2   MON JSR CLEAR
0511 c2ad c6 15     LDAB #FASDLY
0512 c2af bd d3 df   JSR DELAY
0513 c2b2 86 40     LDAA #$40
0514 c2b4 b7 20 9e   STAA DISPY
0515 c2b7 20 0f     BRA MON1
0516
0517 c2b9 ff 20 a6   MONHL STX BUF
0518 c2bc bd d1 b6   JSR UNPK
0519 c2bf ce 00 00   LDX #$0000
0520 c2c2 ff 20 a2   STX DISPY+4
0521 c2c5 bd d4 33   JSR HBEEP

```

0522

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0523 c2c8 8e 20 3c      MON1      LDS  #STACK
0524 c2cb ce 20 b4      LDX  #SYSPAG
0525 c2ce 1c 00 10      BSET 0,X $10
0526 c2d1 bd d0 50      JSR  SCAN
0527 c2d4 1d 00 10      BCLR 0,X $10
0528 c2d7 81 10      CMPA  #$10
0529 c2d9 27 2f      BEQ  MONF
0530 c2db 81 12      CMPA  #$12
0531 c2dd 26 03      BNE  MON11
0532 c2df 7e c5 3c      JMP  RUN
0533 c2e2 81 14      MON11    CMPA  #$14
0534 c2e4 26 03      BNE  MON12
0535 c2e6 7e c3 6a      JMP  MEM
0536 c2e9 81 15      MON12    CMPA  #$15
0537 c2eb 26 03      BNE  MON13
0538 c2ed 7e c5 bc      JMP  STEP
0539 c2f0 81 16      MON13    CMPA  #$16
0540 c2f2 26 03      BNE  MON14
0541 c2f4 7e c3 d9      JMP  REG
0542
0543 c2f7 bd d2 0d      MON14    JSR  GET4
0544 c2fa 4f      CLRA
0545 c2fb f7 20 4e      STAB  REGC
0546 c2fe b1 20 4e      CMPA  REGC
0547 c301 26 c5      BNE  MON1
0548 c303 86 40      LDAA  #$40
0549 c305 b7 20 9e      STAA  DISPY
0550 c308 20 be      BRA  MON1
0551
0552
0553
0554 c30a 8e 20 3c      `MONF    LDS  #STACK
0555 c30d 18 ce d5 24      LDY  #FUNTAB

```

***** FUNCTION *****

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0556	c311 c6 05		LDAB	##05
0557	c313 f7 20 4e		STAB	REGC
0558	c316 c6 05		LDAB	##05
0559	c318 bd d3 57		JSR	PRINT
0560	c31b ce 20 b4	MONF1	LDX	#SYSFAG
0561	c31e 1c 00 10		BSET	0,X \$10
0562	c321 bd d0 50		JSR	SCAN
0563	c324 1d 00 10		BCLR	0,X \$10
0564				
0565	c327 81 10		CMPA	##10
0566	c329 27 22		BEQ	MONF2
0567	c32b 81 12		CMPA	##12
0568	c32d 26 03		BNE	MONF11
0569	c32f 7e c5 4c		JMP	RUN1
0570	c332 81 14	MONF11	CMPA	##14
0571	c334 27 44		BEQ	MEM1
0572	c336 81 15		CMPA	##15
0573	c338 26 03		BNE	MONF12
0574	c33a 7e c5 cc		JMP	STEP1
0575	c33d 81 16	MONF12	CMPA	##16
0576	c33f 26 03		BNE	MONF13
0577	c341 7e c4 06		JMP	REG1
0578				
0579	c344 81 11	MONF13	CMPA	##11
0580	c346 24 d3		BCC	MONF1
0581	c348 ce d4 d6		LDX	#FK1TAB
0582	c34b 20 0f		BRA	MONF3
0583				
0584	c34d 86 5b	MONF2	LDA	##5B
0585	c34f b7 20 a2		STAA	DISPY+4
0586	c352 bd d0 50		JSR	SCAN
0587	c355 81 10		CMPA	##10
0588	c357 24 f4		BCC	MONF2

```

0589 c359 ce d4 f6          LDX #FK2TAB
0590
0591 c35c 16                MONF3      TAB
0592 c35d bd d4 4f          JSR TABLE
0593 c360 8c 00 00          CPX #0000
0594 c363 26 03            BNE MON31
0595 c365 7e c2 aa          JMP MON
0596 c368 6e 00            MON31      JMP 0,X
0597
0598
0599          ***** DATA *****
0600
0601 c36a b6 20 9e          MEM      LDAA DISPY
0602 c36d 81 40            CMPA #40
0603 c36f 27 09            BEQ MEM1
0604 c371 bd d1 54          JSR PACK
0605 c374 fe 20 a6          LDX BUF
0606 c377 ff 20 92          STX ADDR
0607 c37a bd d3 d2          MEM1     JSR CLEAR
0608 c37d 8e 20 3c          LDS #STACK
0609 c380 18 fe 20 92          LDY ADDR
0610 c384 18 ff 20 a6          MEM2     STY BUF
0611 c388 18 a6 00          LDAA 0,Y
0612 c38b b7 20 a8          STAA BUF+2
0613 c38e 18 3c            PSHY
0614 c390 bd d1 b6          JSR UNPK
0615 c393 18 38            PULY
0616 c395 b6 20 a1          LDAA DISPY+3
0617 c398 8a 80            ORAA #80
0618 c39a b7 20 a1          STAA DISPY+3
0619
0620 c39d bd d0 50          MEM3     JSR SCAN
0621 c3a0 81 14            CMPA #14

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0622 c3a2 27 e0      BEQ  MEM2
0623 c3a4 81 11      CMPA  #$11
0624 c3a6 27 2d      BEQ  MEM5
0625 c3a8 81 13      CMPA  #$13
0626 c3aa 27 25      BEQ  MEM4
0627
0628 c3ac ce 20 a2    LDX  #DISPY+4
0629 c3af bd d2 5c    JSR  GET2
0630 c3b2 25 e9      BCS  MEM3
0631 c3b4 18 3c      PSHY
0632 c3b6 bd d1 54    JSR  PACK
0633 c3b9 18 38      PULY
0634 c3bb b6 20 a8    LDAA  BUF+2
0635 c3be 18 a7 00    STAA  0,Y
0636
0637                ***** DATA ENTRY OK ? *****
0638
0639 c3c1 b7 20 4f    STAA  REGD
0640 c3c4 18 a6 00    LDAA  0,Y
0641 c3c7 b1 20 4f    CMPA  REGD
0642 c3ca 26 b8      BNE  MEM2
0643 c3cc c6 15      LDAB  #FASDLY
0644 c3ce bd d3 6f    JSR  SCAND
0645
0646 c3d1 18 08      MEM4      INY
0647 c3d3 20 af      BRA  MEM2
0648 c3d5 18 09      MEM5      DEY
0649 c3d7 20 ab      BRA  MEM2
0650
0651                ***** REG *****
0652
0653 c3d9 86 00      REG      LDAA  #00
0654 c3db b7 20 45    STAA  REGS+8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0655 c3de b6 20 9e		LDA A DISP Y
0656 c3e1 81 40		CMP A # \$40
0657 c3e3 27 21		BEQ REG1
0658 c3e5 bd d1 54		JSR PACK
0659 c3e8 b6 20 a6		LDA A BUF
0660 c3eb 84 f0		ANDA # \$F0
0661 c3ed 44		LSRA
0662 c3ee 44		LSRA
0663 c3ef 44		LSRA
0664 c3f0 44		LSRA
0665 c3f1 81 0b		CMP A # \$0B
0666 c3f3 25 11		BCS REG1
0667 c3f5 80 0a		SUB A # \$0A
0668 c3f7 16		TAB
0669 c3f8 1b		ABA
0670 c3f9 16		TAB
0671 c3fa ce 20 3d		LDX #REGS
0672 c3fd 3a		ABX
0673 c3fe 18 ce d5 60		LDY #REGTAB
0674 c402 18 3a		ABY
0675 c404 20 0a		BRA REG2
0676		
0677 c406 8e 20 3c	REG1	LDS #STACK
0678 c409 ce 20 3d		LDX #REGS
0679 c40c 18 ce d5 60		LDY #REGTAB
0680 c410 b6 20 46	REG2	LDA A REGS+9
0681 c413 b7 20 a4		STAA DISPY+6
0682 c416 bd c5 08	REGL65	JSR REGLD
0683 c419 b6 20 54		LDA A REGL
0684 c41c 8a 80		ORAA # \$80
0685 c41e b7 20 54		STAA REGL
0686 c421 f8 20 53		LDAB REGH
0687 c424 f7 20 a3		STAB DISPY+5

0688 c427 f6 20 54		LDAB REGL
0689 c42a f7 20 a2		STAB DISPY+4
0690		
0691 c42d 3c	REGL1	PSHX
0692 c42e 18 3c		PSHY
0693 c430 bd d0 50		JSR SCAN
0694 c433 18 38		PULY
0695 c435 38		PULX
0696 c436 81 11		CMPA #\$11
0697 c438 27 1c		BEQ REGL2
0698 c43a 81 13		CMPA #\$13
0699 c43c 27 3f		BEQ REGR
0700 c43e 3c		PSHX
0701 c43f 18 3c		PSHY
0702 c441 ce 20 9e		LDX #DISPY
0703 c444 bd d2 5c		JSR GET2
0704 c447 24 05		BCC REGL11
0705 c449 18 38		PULY
0706 c44b 38		PULX
0707 c44c 20 df		BRA REGL1
0708 c44e 18 38	REGL11	PULY
0709 c450 38		PULX
0710 c451 bd c5 28		JSR REGGT
0711 c454 20 c0		BRA REGL65
0712 c456 3c	REGL2	PSHX
0713 c457 18 3c		PSHY
0714 c459 18 ce 20 3d		LDY #REGS
0715 c45d 18 ff 20 8c		STY BC
0716 c461 bc 20 8c		CPX BC
0717 c464 26 05		BNE REGL21
0718 c466 18 38		PULY
0719 c468 38		PULX
0720 c469 20 0b		BRA REGL3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0721 c46b 18 38	REGL21	PULY
0722 c46d 38		PULX
0723 c46e 09		DEX
0724 c46f 09		DEX
0725 c470 18 09		DEY
0726 c472 18 09		DEY
0727 c474 20 07		BRA REGR
0728		
0729 c476 ce 20 47	REGL3	LDX #SP
0730 c479 18 ce d5 6a		LDY #REGTAB+10
0731		
0732 c47d bd c5 08	REGR	JSR REGLD
0733 c480 b6 20 53		LDAA REGH
0734 c483 8a 80		ORAA #80
0735 c485 b7 20 53		STAA REGH
0736 c488 f6 20 53		LDAB REGH
0737 c48b f7 20 a3		STAB DISPY+5
0738 c48e f6 20 54		LDAB REGL
0739 c491 f7 20 a2		STAB DISPY+4
0740 c494 3c	REGR1	PSHX
0741 c495 18 3c		PSHY
0742 c497 bd d0 50		JSR SCAN
0743 c49a 18 38		PULY
0744 c49c 38		PULX
0745 c49d 81 11		CMPA #\$11
0746 c49f 26 03		BNE REGL651
0747 c4a1 7e c4 16		JMP REGL65
0748 c4a4 81 13	REGL651	CMPA #\$13
0749 c4a6 27 35		BEQ REGR2
0750 c4a8 3c		PSHX
0751 c4a9 18 3c		PSHY
0752 c4ab ce 20 a0		LDX #DISPY+2
0753 c4ae bd d2 5c		JSR GET2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0754 c4b1 24 05		BCC REGR11
0755 c4b3 18 38		PULY
0756 c4b5 38		PULX
0757 c4b6 20 dc		BRA REGR1
0758 c4b8 18 38	REGR11	PULY
0759 c4ba 38		PULX
0760 c4bb bd c5 28		JSR REGGT
0761 c4be 3c		PSHX
0762 c4bf 18 3c		PSHY
0763 c4c1 18 ce 20 45		LDY #REGS+8
0764 c4c5 18 ff 20 8c		STY BC
0765 c4c9 bc 20 8c		CPX BC
0766 c4cc 27 05		BEQ REG4
0767 c4ce 18 38		PULY
0768 c4d0 38		PULX
0769 c4d1 20 aa		BRA REGR
0770 c4d3 18 38	REG4	PULY
0771 c4d5 38		PULX
0772 c4d6 a6 01		LDAA 1,X
0773 c4d8 b7 20 a4		STAA DISPY+6
0774 c4db 20 a0		BRA REGR
0775 c4dd 3c	REGR2	PSHX
0776 c4de 18 3c		PSHY
0777 c4e0 18 ce 20 47		LDY #SP
0778 c4e4 18 ff 20 8c		STY BC
0779 c4e8 bc 20 8c		CPX BC
0780 c4eb 26 05		BNE REGR31
0781 c4ed 18 38		PULY
0782 c4ef 38		PULX
0783 c4f0 20 0c		BRA REGR3
0784 c4f2 18 38	REGR31	PULY
0785 c4f4 38		PULX
0786 c4f5 08		INX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0787	c4f6 08		INX
0788	c4f7 18 08		INY
0789	c4f9 18 08		INX
0790	c4fb 7e c4 16		JMP REGL65
0791			
0792	c4fe ce 20 3d	REGR3	LDX #REGS
0793	c501 18 ce d5 60		LDY #REGTAB
0794	c505 7e c4 10		JMP REG2
0795			
0796	c508 a6 00	REGLD	LDA 0,X
0797	c50a b7 20 a6		STAA BUF
0798	c50d a6 01		LDA 1,X
0799	c50f b7 20 a7		STAA BUF+1
0800	c512 18 3c		PSHY
0801	c514 3c		PSHX
0802	c515 bd d1 b6		JSR UNPK
0803	c518 38		PULX
0804	c519 18 38		PULY
0805	c51b 18 a6 00		LDA 0,Y
0806	c51e b7 20 54		STAA REGL
0807	c521 18 a6 01		LDA 1,Y
0808	c524 b7 20 53		STAA REGH
0809	c527 39		RTS
0810			
0811	c528 18 3c	REGGT	PSHY
0812	c52a 3c		PSHX
0813	c52b bd d1 54		JSR PACK
0814	c52e 38		PULX
0815	c52f 18 38		PULY
0816	c531 b6 20 a6		LDA 0, BUF
0817	c534 a7 00		STAA 0,X
0818	c536 b6 20 a7		LDA 0, BUF+1
0819	c539 a7 01		STAA 1,X

```

0820 c53b 39          RTS
0821
0822          ***** RUN PROGRAM *****
0823
0824 c53c b6 20 9e    RUN      LDAA  DISPY
0825 c53f 81 40          CMPA  #$40
0826 c541 27 09          BEQ   RUN1
0827 c543 bd d1 54          JSR  PACK
0828 c546 fe 20 a6          LDX  BUF
0829 c549 ff 20 3d          STX  REGS
0830
0831 c54c fe 20 3d    RUN1      LDX  REGS
0832 c54f a6 00          LDAA  0,X
0833 c551 81 00          CMPA  #00
0834 c553 26 06          BNE  RUN11
0835 c555 bd d4 2c          JSR  LBEEP
0836 c558 7e c2 aa          JMP  MON
0837
0838 c55b fe 20 3d    RUN11      LDX  REGS
0839 c55e 18 fe 20 90          LDY  BRK
0840 c562 18 ff 20 8c          STY  BC
0841 c566 bc 20 8c          CPX  BC
0842 c569 24 0c          BCC  RUN2
0843 c56b fe 20 90          LDX  BRK
0844 c56e a6 00          LDAA  0,X
0845 c570 b7 20 6b          STAA BRKMEM
0846 c573 86 3f          LDAA  #SWI
0847 c575 a7 00          STAA  0,X
0848
0849 c577 fe 20 f5    RUN2      LDX  JSWI+1
0850 c57a ff 20 4b          STX  PTR2
0851 c57d 86 7e          LDAA  #$7E
0852 c57f b7 20 f4          STAA JSWI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0853	c582 ce c5 8b		LDX #SWIIN
0854	c585 ff 20 f5		STX JSWI+1
0855	c588 7e c6 25		JMP RESTACK
0856			
0857	c58b 30	SWIIN	TSX
0858	c58c be 20 49		LDS PTR1
0859	c58f bd c6 45		JSR SAVSTACK
0860	c592 b6 20 6b		LDAA BRKMEM
0861	c595 fe 20 90		LDX BRK
0862	c598 a7 00		STAA 0,X
0863	c59a fe 20 4b		LDX PTR2
0864	c59d ff 20 f5		STX JSWI+1
0865	c5a0 fe 20 3d		LDX REGS
0866	c5a3 09		DEX
0867	c5a4 ff 20 3d		STX REGS
0868	c5a7 ff 20 a6		STX BUF
0869	c5aa bd d1 b6		JSR UNPK
0870	c5ad ce 00 00		LDX #0000
0871	c5b0 ff 20 a2		STX DISPY+4
0872	c5b3 b6 20 46		LDAA REGS+9
0873	c5b6 b7 20 a4		STAA DISPY+6
0874	c5b9 7e c2 c8		JMP MON1
0875			
0876			***** STEP *****
0877			
0878	c5bc b6 20 9e	STEP	LDAA DISPY
0879	c5bf 81 40		CMPA #40
0880	c5c1 27 09		BEQ STEP1
0881	c5c3 bd d1 54		JSR PACK
0882	c5c6 fe 20 a6		LDX BUF
0883	c5c9 ff 20 3d		STX REGS
0884			
0885	c5cc b6 20 9d	STEP1	LDAA NSTEP

0886	c5cf b7 20 d		STAA	TMP1
0887				
0888	c5d2 bd c5 f4	STEP2	JSR	RUNONE
0889	c5d5 7a 20 d		DEC	TMP1
0890	c5d8 27 02		BEQ	TRACDON
0891	c5da 20 f6		BRA	STEP2
0892				
0893	c5dc fe 20 3d	TRACDON	LDX	REGS
0894	c5df ff 20 a6		STX	BUF
0895	c5e2 bd d1 b6		JSR	UNPK
0896	c5e5 ce 00 00		LDX	#0000
0897	c5e8 ff 20 a2		STX	DISPY+4
0898	c5eb b6 20 46		LDAA	REGS+9
0899	c5ee b7 20 a4		STAA	DISPY+6
0900	c5f1 7e c2 c8		JMP	MON1
0901				
0902	c5f4 86 7e	RUNONE	LDAA	##7E
0903	c5f6 b7 20 f1		STAA	JXIRQ
0904	c5f9 ce c6 41		LDX	#XIRQIN
0905	c5fc ff 20 f2		STX	JXIRQ+1
0906	c5ff b6 10 20		LDAA	TCCTL1
0907	c602 8a 03		ORAA	##03
0908	c604 b7 10 20		STAA	TCCTL1
0909	c607 86 08		LDAA	##08
0910	c609 b7 10 0b		STAA	CFORC
0911	c60c b6 20 46		LDAA	REGS+9
0912	c60f 84 bf		ANDA	##BF
0913	c611 b7 20 46		STAA	REGS+9
0914	c614 c6 57		LDAB	#87
0915	c616 fe 10 0e		LDX	TCNT
0916	c619 3a		ABX	
0917	c61a ff 10 1e		STX	TOCS
0918	c61d b6 10 20		LDAA	TCCTL1

0919	c620 84 fe		ANDA #SFE
0920	c622 b7 10 20		STAA TCTL1
0921			
0922	c625 bf 20 49	RESTACK	STS PTR1
0923	c628 be 20 47		LDS SP
0924	c62b fe 20 3d		LDX REGS
0925	c62e 3c		PSHX
0926	c62f fe 20 3f		LDX REGS+2
0927	c632 3c		PSHX
0928	c633 fe 20 41		LDX REGS+4
0929	c636 3c		PSHX
0930	c637 fc 20 43		LDD REGS+6
0931	c63a 36		PSHA
0932	c63b 37		PSHB
0933	c63c b6 20 46		LDAA REGS+9
0934	c63f 36		PSHA
0935	c640 3b		RTI
0936			
0937	c641 30	XIRQIN	TSX
0938	c642 be 20 49		LDS PTR1
0939			
0940	c645 a6 00	SAVSTACK	LDAA 0,X
0941	c647 b7 20 46		STAA REGS+9
0942	c64a ec 01		LDD 1,X
0943	c64c b7 20 44		STAA REGS+7
0944	c64f f7 20 43		STAB REGS+6
0945	c652 ec 03		LDD 3,X
0946	c654 fd 20 41		STD REGS+4
0947	c657 ec 05		LDD 5,X
0948	c659 fd 20 3f		STD REGS+2
0949	c65c ec 07		LDD 7,X
0950	c65e fd 20 3d		STD REGS
0951	c661 c6 08		LDAB #8

```

0952 c663 3a ABX
0953 c664 ff 20 47 STX SP
0954 c667 b6 10 20 LDAA TCTL1
0955 c66a 8a 03 ORAA ##03
0956 c66c b7 10 20 STAA TCTL1
0957 c66f 86 08 LDAA ##08
0958 c671 b7 10 0b STAA CFORC
0959 c674 39 RTS
0960
0961 ***** HELP FUNCTION *****
0962
0963 c675 ce d6 9f HELP LDX #HELP_TAB
0964 c678 bd cf c4 JSR ROW_LCDL
0965 c67b bd d0 47 JSR DELAYL
0966 c67e ce d6 af LDX #ADDR_TAB
0967 c681 bd cf c4 JSR ROW_LCDL
0968 c684 bd d0 47 JSR DELAYL
0969 c687 c6 04 LDAB ##04
0970 c689 f7 20 4e STAB REGC
0971 c68c c6 10 LDAB ##10
0972 c68e 18 ce d5 5c LDY #SHELP_TAB
0973 c692 bd d3 57 JSR PRINT
0974 c695 c6 0a ST_HELP LDAB ##0A
0975 c697 ce d6 cf LDX #PW_TAB
0976 c69a ff 20 8e STX HL
0977 c69d fe 20 8e ST_HELP1 LDX HL
0978 c6a0 37 PSHB
0979 c6a1 bd cf c4 JSR ROW_LCDL
0980 c6a4 bd d0 47 JSR DELAYL
0981 c6a7 bd d0 47 JSR DELAYL
0982 c6aa ce d6 bf LDX #CON_TAB
0983 c6ad bd cf bd JSR ROW_LCD
0984 c6b0 bd d0 50 ANS_HELP JSR SCAN

```

```

0985 c6b3 81 13          CMPA  #$13
0986 c6b5 27 06          BEQ  ANS_HELP1
0987 c6b7 81 11          CMPA  #$11
0988 c6b9 27 13          BEQ  LAST_HELP
0989 c6bb 20 f3          BRA  ANS_HELP
0990 c6bd fe 20 8e      ANS_HELP1  LDX  HL
0991 c6c0 c6 10          LDAB #$10
0992 c6c2 3a            ABX
0993 c6c3 ff 20 8e      STX  HL
0994 c6c6 33            PULB
0995 c6c7 5a            DECB
0996 c6c8 26 d3          BNE  ST_HELP1
0997 c6ca 81 13          CMPA  #$13
0998 c6cc 27 c7          BEQ  ST_HELP
0999 c6ce bd d0 0a      LAST_HELP  JSR  CLEAR_L
1000 c6d1 7e c2 aa          JMP  MON
1001
1002          ***** BEEPT FUNCTION *****
1003
1004 c6d4 bd d3 d2      BEEPT  JSR  CLEAR
1005 c6d7 ce 20 b4      LDX  #SYSFAG
1006 c6da a6 00          LDAA 0,X
1007 c6dc 85 08          BITA  #$08
1008 c6de 27 09          BEQ  BEEPT1
1009
1010 c6e0 1d 00 08      BCLR 0,X $08
1011 c6e3 18 ce d5 31      LDY  #ONTAB
1012 c6e7 20 07          BRA  BEEPT2
1013
1014 c6e9 1c 00 08      EEPT1  BSET 0,X $08
1015 c6ec 18 ce d5 2e      LDY  #OFFTAB
1016 c6f0 86 03          EEPT2  LDAA  #03
1017 c6f2 b7 20 4e      STAA  REGC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1018 c6f5 c6 30          LDAB  ##30
1019 c6f7 bd d3 57      JSR  PRINT
1020 c6fa 7e c2 aa      JMP  MON
1021
1022                      ***** USER FUNCTION *****
1023
1024 c6fd 18 ce d5 39    USER          LDY  #USRTAB
1025 c701 c6 04          LDAB  ##04
1026 c703 f7 20 4e      STAB  REGC
1027 c706 c6 01          LDAB  #01
1028 c708 bd d3 57      JSR  PRINT
1029
1030 c70b bd d0 50      USER1          JSR  SCAN
1031 c70e 81 10          CMPA  ##10
1032 c710 24 f9          BCC  USER1
1033 c712 ce 20 6c      LDX  #USRKEY
1034 c715 16            TAB
1035 c716 bd d4 4f      JSR  TABLE
1036 c719 8c 00 00      CPX  #0000
1037 c71c 27 02          BEQ  USER2
1038 c71e 6e 00          JMP  0,X
1039
1040 c720 bd d4 2c      USER2          JSR  LBEEP
1041 c723 7e c2 aa      JMP  MON
1042
1043                      ***** SET FUNCTION *****
1044
1045 c726 18 ce d5 3d    SET           LDY  #SETTAB
1046 c72a c6 04          LDAB  #04
1047 c72c f7 20 4e      STAB  REGC
1048 c72f c6 01          LDAB  #01
1049 c731 bd d3 57      JSR  PRINT
1050

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1051	c734 bd d0 50	SET_1	JSR SCAN
1052	c737 81 10		CMPA ##10
1053	c739 24 f9		BCC SET_1
1054	c73b b7 20 4e		STAA REGC
1055	c73e bd d1 e7		JSR UNPKT
1056	c741 b7 20 a2		STAA DISPY+4
1057	c744 c6 30		LDAB #MIDDLY
1058	c746 bd d3 6f		JSR SCAND
1059	c749 ce 20 6c		LDX #USRKEY
1060	c74c b6 20 4e		LDA A REGC
1061	c74f 48		LSLA
1062	c750 16		TAB
1063	c751 3a		ABX
1064	c752 3c		PSHX
1065	c753 18 38		PULY
1066	c755 18 a6 00		LDA A 0,Y
1067	c758 18 e6 01		LDAB 1,Y
1068	c75b 8f		XGDX
1069	c75c c6 77		LDAB ##77
1070	c75e 18 3c		PSHY
1071	c760 bd d2 7b		JSR IN4
1072	c763 18 38		PULY
1073	c765 8f		XGDX
1074	c766 18 a7 00		STAA 0,Y
1075	c769 18 e7 01		STAB 1,Y
1076	c76c bd d4 33		JSR HBEEP
1077	c76f 7e c2 aa		JMP MON
1078			
1079		*****	RECEIVE FUNCTION *****
1080			
1081	c772 ce d5 ef	REV	LDX #RX_TAB
1082	c775 bd cf c4		JSR ROW_LCDL
1083	c778 86 30		LDA A ##30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1084	c77a b7 10 2b		STAA	BAUD
1085	c77d 7f 10 2c		CLR	SCCR1
1086	c780 86 0c		LDAA	##0C
1087	c782 b7 10 2d		STAA	SCCR2
1088				
1089	c785 bd d0 0a	REV1	JSR	CLEAR_L
1090	c788 86 00		LDAA	##00
1091	c78a bd d0 15		JSR	CURSOR_L
1092	c78d 86 08		LDAA	##08
1093	c78f b7 20 4e		STAA	REGC
1094	c792 b6 10 2e	REV2	LDAA	SCSR
1095	c795 85 20		BITA	##20
1096	c797 27 f9		BEQ	REV2
1097	c799 f6 10 2f		LDAB	SCDAT
1098	c79c bd d0 37		JSR	READY_L
1099	c79f f7 14 01		STAB	WR1LCD
1100	c7a2 7a 20 4e		DEC	REGC
1101	c7a5 26 eb		BNE	REV2
1102				
1103	c7a7 86 40		LDAA	##40
1104	c7a9 bd d0 15		JSR	CURSOR_L
1105	c7ac 86 08		LDAA	##08
1106	c7ae b7 20 4e		STAA	REGC
1107	c7b1 b6 10 2e	REV3	LDAA	SCSR
1108	c7b4 85 20		BITA	##20
1109	c7b6 27 f9		BEQ	REV3
1110	c7b8 f6 10 2f		LDAB	SCDAT
1111	c7bb bd d0 37		JSR	READY_L
1112	c7be f7 14 01		STAB	WR1LCD
1113	c7c1 7a 20 4e		DEC	REGC
1114	c7c4 26 eb		BNE	REV3
1115	c7c6 20 bd		BRA	REV1

1116

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1117          ***** TIMER EXAMPLE 1 FUNCTION *****
1118
1119 c7c8 ce d6 0f          TIMER1          LDX  #TIME_TAB
1120 c7cb bd cf c4          JSR  ROW_LCDL
1121 c7ce ce d5 a5          LDX  #CYC_TAB
1122 c7d1 18 ce 21 50       LDY  #$2150
1123 c7d5 bd cf ac          JSR  M_ASCII
1124 c7d8 ce 10 00         PERTOP          LDX  #REGBS
1125 c7db 86 10            LDAA  #%00010000
1126 c7dd a7 21            STAA  TCTL2,X
1127 c7df 86 04            LDAA  #$04
1128 c7e1 a7 23            STAA  TFLG1,X
1129
1130          ***** FIRST RISING EDGE *****
1131
1132 c7e3 1f 23 04 fc       BRCLR TFLG1,X $04      *
1133
1134          ***** FIRST EDGE DETECTED *****
1135
1136 c7e7 ec 10            LDD  TIC1,X
1137 c7e9 fd 20 b8         STD  BUFF_T1
1138 c7ec 86 04            LDAA  #$04
1139 c7ee a7 23            STAA  TFLG1,X
1140
1141          ***** SECOND RISING EDGE *****
1142
1143 c7f0 1f 23 04 fc       BRCLR TFLG1,X $04      *
1144
1145          ***** SECOND EDGE DETECTED *****
1146
1147 c7f4 ec 10            LDD  TIC1,X
1148 c7f6 b3 20 b8         SUBD  BUFF_T1
1149 c7f9 fd 21 47         STD  $2147

```

1150

1151

***** DISPLAY *****

1152

1153 c7fc c6 22

LDAB #\$22

1154 c7fe bd ff 00

JSR CALL

1155 c801 86 20

LDAA #\$20

1156 c803 b7 21 4f

STAA \$214F

1157 c806 ce 21 4a

LDX #\$214A

1158 c809 bd cf bd

JSR ROW_LCD

1159 c80c bd d0 47

JSR DELAYL

1160 c80f 7e c7 d8

JMP PERTOP

1161

1162

*** ANALOG TO DIGITAL 8 CH FUNCTION ***

1163

1164 c812 86 93

ADC_8CH

LDAA #\$93

1165 c814 b7 10 39

STAA OPTION

1166 c817 4f

CLRA

1167 c818 b7 10 24

STAA TMSK2

1168 c81b ce d5 bf

LDX #ADC8_TAB

1169 c81e bd cf c4

JSR ROW_LCDL

1170 c821 ce d7 84

LDX #ADC8CH_TAB

1171 c824 bd cf 8b

JSR TX_SUB

1172 c827 ce 10 00

SET_ADC1

LDX #REGBS

1173 c82a 86 02

LDAA #\$02

1174 c82c b7 20 4e

STAA REGC

1175 c82f 86 14

LDAA #\$14

1176 c831 a7 30

STAA ADCTL,X

1177 c833 1f 30 50 fc

LOOP

BRCLR ADCTL,X #80 LOOP

1178

1179 c837 ce 20 bb

RUN_ADC2

LDX #BUFF_ADC

1180 c83a 18 ce 10 31

LDY #\$1031

1181 c83e 18 a6 00

READ1

LDAA 0,Y

1182 c841 a7 00

STAA 0,X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1183 c843 18 08          INY
1184 c845 08            INX
1185 c846 18 8c 10 35   CPY  #$1035
1186 c84a 26 f2         BNE  READ1
1187
1188                    ***** DISPLAY *****
1189                    *   $FF = 255 = 5V   ;   1 = 0.0196V   *
1190                    *   ACT*196/10 = ACTUAL mv   *
1191
1192 c84c ce 20 bb        LDX  #BUFF_ADC
1193 c84f 3c              CH_NEXT  PSHX
1194 c850 a6 00          LDAA  0,X
1195 c852 c6 c4          LDAB  #196
1196 c854 3d            MUL
1197 c855 fd 21 47       STD  $2147
1198 c858 c6 22          LDAB  #$22
1199 c85a bd ff 00       JSR  CALL
1200
1201 c85d 86 04          LDAA  #$04
1202 c85f b7 21 4e       STAA  $214E
1203 c862 ce d7 7d       LDX  #MSG0
1204 c865 bd cf 8b       JSR  TX_SUB
1205 c868 ce 21 4a       LDX  #$214A
1206 c86b bd cf 8b       JSR  TX_SUB
1207 c86e ce d7 80       LDX  #MSG
1208 c871 bd cf 8b       JSR  TX_SUB
1209 c874 38            PULX
1210 c875 08            INX
1211 c876 8c 20 bf       CPX  #BUFF_ADC+4
1212 c879 26 d4          BNE  CH_NEXT
1213 c87b 7a 20 4e       DEC  REGC
1214 c87e 27 0d         BEQ  LADC_8CH
1215 c880 ce 10 00       LDX  #REGBS

```

```

1216 c883 86 10          LDAA  ##10
1217 c885 a7 30          STAA  ADCTL,X
1218 c887 1f 30 50 fc    LOOP1    BRCLR ADCTL,X #80 LOOP1
1219 c88b 20 aa          BRA   RUN_ADC2
1220 c88d ce d7 7b      LADC_8CH  LDX  #COL1_TAB
1221 c890 bd cf 8b      JSR   TX_SUB
1222 c893 20 92          BRA   SET_ADC1
1223
1224                      **** TRANSMIT & RECEIVE FUNCTION ****
1225
1226 c895 ce d5 ff      TX_RX    LDX  #TXRX_TAB
1227 c898 bd cf c4      JSR   ROW_LCDL
1228 c89b 86 30          LDAA  ##30
1229 c89d b7 10 2b      STAA  BAUD
1230 c8a0 7f 10 2c      CLR  SCCR1
1231 c8a3 86 0c          LDAA  ##0C
1232 c8a5 b7 10 2d      STAA  SCCR2
1233
1234 c8a8 b6 10 2e      RECEIVE  LDAA  SCSR
1235 c8ab 85 20          BITA  ##20
1236 c8ad 27 f9          BEQ  RECEIVE
1237
1238 c8af f6 10 2f      LDAB  SCDAT
1239 c8b2 b6 10 2e      TRANSMIT LDAA  SCSR
1240 c8b5 85 80          BITA  ##80
1241 c8b7 27 f9          BEQ  TRANSMIT
1242 c8b9 f7 10 2f      STAB  SCDAT
1243 c8bc 20 ea          BRA   RECEIVE
1244
1245
1246                      ***** VOLTS METER FUNCTION *****
1247
1248 c8be 86 93          VOLTS_METER LDAA  ##93

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1249 c8c0 b7 10 39          STAA  OPTION
1250 c8c3 4f                CLRA
1251 c8c4 b7 10 24          STAA  TMSK2
1252 c8c7 ce d5 cf          LDX  #VOLT_TAB
1253 c8ca bd cf c4          JSR  ROW_LCDL
1254 c8cd ce d7 6f          LDX  #MV_TAB
1255 c8d0 18 ce 21 4f       LDY  #$214F
1256 c8d4 bd cf ac          JSR  M_ASCII
1257 c8d7 ce 10 00          LDX  #REGBS
1258 c8da 86 20              LDAA  #$20
1259 c8dc a7 30              STAA  ADCTL,X
1260 c8de 1d 30 80          FLAG  BCLR  ADCTL,X $80
1261
1262 c8e1 ce 20 bb          LDX  #BUFF_ADC
1263 c8e4 18 ce 10 31       LDY  #$1031
1264 c8e8 18 a6 00          READ  LDAA  0,Y
1265 c8eb a7 00              STAA  0,X
1266 c8ed 18 08             INY
1267 c8ef 08                INX
1268 c8f0 18 8c 10 35       CPY  #$1035
1269 c8f4 26 f2             BNE  READ
1270
1271          ***** DISPLAY *****
1272          *   $FF = 255 = 5V   ;   1 = 0.0196V   *
1273          *           ACT*196/10 = ACTUAL mv       *
1274
1275 c8f6 ce 20 bb          LDX  #BUFF_ADC
1276 c8f9 a6 00              LDAA  0,X
1277 c8fb c6 c4              LDAB  #196
1278 c8fd 3d                MUL
1279
1280 c8fe fd 21 47          STD  $2147
1281 c901 c6 22              LDAB  #$22

```

```

1282 c903 bd ff 00      JSR   CALL
1283 c906 c6 20        LDAB  #$20
1284 c908 f7 21 4e     STAB  $214F
1285 c90b ce 21 4a     LDX  #$214A
1286 c90e bd cf bd     JSR  ROW_LCD
1287 c911 bd d0 47     JSR  DELAYL
1288 c914 bd d0 47     JSR  DELAYL
1289 c917 20 c5        BRA  FLAG
1290
1291                    ***** TIMER 2 FUNCTION *****
1292
1293 c919 ce d6 if      TIMER2  LDX  #SQW_TAB
1294 c91c bd cf c4     JSR  ROW_LCDL
1295 c91f 86 7e      TOP5    LDAA #7E
1296 c921 b7 20 dc     STAA JTOC2
1297 c924 ce c9 38     LDX  #INTOC2
1298 c927 ff 20 dd     STX  JTOC2+1
1299 c92a ce 10 00     LDX  #REGBS
1300 c92d 86 40      LDAA #01000000
1301 c92f a7 20      STAA OTCTL1,X
1302 c931 a7 23      STAA TFLG1,X
1303 c933 a7 22      STAA OTMSK1,X
1304 c935 0e        CLI
1305 c936 20 fe      LOOP_W  BRA  LOOP_W
1306
1307                    ***** INTERRUPT SERVICE ROUTINE *****
1308
1309 c938 cc ff ff      INTOC2  LDD  #$FFFF
1310 c93b e3 18        ADDD  TOC2,X
1311 c93d ed 18        STD  TOC2,X
1312 c93f 1d 23 bf     BCLR  TFLG1,X $BF
1313 c942 3b          RTI

```

1314
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1315                ***** TEST LCD FUNCTION *****
1316
1317 c943 ce d5 af      TEST_LCD      JDX  #TLCD_TAB
1318 c946 bd cf c4      JSR   ROW_LCDL
1319 c949 bd cf d5      JSR   INIT_LCD
1320 c94c bd d0 37      JSR   READY_L
1321 c94f 86 0c         LDAA  #$0C
1322 c951 b7 14 00      STAA  SET1
1323 c954 c6 06         LDAB  #$06
1324 c956 86 20         LDAA  #' '
1325 c958 b7 20 b7      STAA  COUNT_L
1326 c95b bd d0 20      TSLCD1 JSR  OUT_L
1327 c95e 5a            DECB
1328 c95f 26 fa         BNE  TSLCD1
1329 c961 86 40         LDAA  #$40
1330 c963 bd d0 15      JSR  CURSOR_L
1331 c966 c6 06         LDAB  #$06
1332 c968 b6 20 b7      LDAA  COUNT_L
1333 c96b bd d0 20      TSLCD2 JSR  OUT_L
1334 c96e 5a            DECB
1335 c96f 26 fa         BNE  TSLCD2
1336 c971 bd d0 47      JSR  DELAYL
1337 c974 bd d0 47      JSR  DELAYL
1338 c977 4f            CLRA
1339 c978 bd d0 15      JSR  CURSOR_L
1340 c97b 7c 20 b7      INC  COUNT_L
1341 c97e b6 20 b7      LDAA  COUNT_L
1342 c981 81 7f         CMPA  #$7F
1343 c983 26 d6         BNE  TSLCD1
1344 c985 bd d0 0a      JSR  CLEAR_L
1345 c988 bd d0 37      JSR  READY_L
1346 c98b 86 0f         LDAA  #$0F
1347 c98d b7 14 00      STAA  SET1

```

```

1348 c990 7e c2 aa          JMP  MON
1349
1350          ***** CLEAR REG. FUNCTION *****
1351
1352 c993 4f          CREG          CLRA
1353 c994 ce 20 47          LDX  #SP
1354 c997 c6 0a          LDAB  #$0A
1355 c999 09          CREG1         DEX
1356 c99a a7 00          STAA  0,X
1357 c99c 5a          DECB
1358 c99d 26 fa          BNE  CREG1
1359 c99f ce 20 1e          LDX  #USTACK
1360 c9a2 ff 20 47          STX  SP
1361 c9a5 ce 23 00          LDX  #RAM
1362 c9a8 ff 20 3d          STX  REGS
1363 c9ab c6 05          LDAB  #05
1364 c9ad f7 20 4e          STAB  REGC
1365 c9b0 c6 20          LDAB  #$20
1366 c9b2 18 ce d5 34          LDY  #CRTAB
1367 c9b6 bd d3 57          JSR  PRINT
1368 c9b9 bd d4 33          JSR  HBEEP
1369 c9bc 7e c2 aa          JMP  MON
1370
1371          ***** TESTR FUNCTION *****
1372
1373 c9bf bd d2 e2          TESTR        JSR  INSF
1374 c9c2 bd d3 05          JSR  READY
1375 c9c5 fe 20 94          LDX  START
1376 c9c8 ff 20 8e          STX  HL
1377 c9cb 18 fe 20 96          LDY  FINAL
1378 c9cf 18 ff 20 8c          STY  BC
1379 c9d3 bd ce 60          JSR  CHKRAM
1380 c9d6 bd d4 33          JSR  HBEEP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1381 c9d9 7e c2 aa          JMP  MON
1382
1383
1384          ***** CHANGE FUNCTION *****
1385
1386 c9dc bd d2 e2          CHAN          JSR  INSF
1387 c9df bd cf 5b          JSR  FIND_GL
1388 c9e2 18 ce d5 4b          LDY  #FRMTAB
1389 c9e6 c6 04          LDAB #04
1390 c9e8 f7 20 4e          STAB REGC
1391 c9eb c6 30          LDAB #30
1392 c9ed bd d3 57          JSR  PRINT
1393 c9f0 18 ce 20 5a          LDY  #FIND_BUF
1394 c9f4 bd cf 70          JSR  FIND_GB
1395 c9f7 f6 20 59          LDAB LENGTH
1396 c9fa ce 20 5a          LDX  #FIND_BUF
1397 c9fd 18 ce 20 62          LDY  #CHAN_BUF
1398 ca01 a6 00          M_CHAN          LDAA 0,X
1399 ca03 18 a7 00          STAA 0,Y
1400 ca06 08          INX
1401 ca07 18 08          INY
1402 ca09 5a          DECB
1403 ca0a 26 f5          BNE  M_CHAN
1404 ca0c 18 ce d5 4f          LDY  #TOTAB
1405 ca10 c6 02          LDAB #02
1406 ca12 f7 20 4e          STAB REGC
1407 ca15 c6 30          LDAB #30
1408 ca17 bd d3 57          JSR  PRINT
1409 ca1a 18 ce 20 62          LDY  #CHAN_BUF
1410 ca1e bd cf 70          JSR  FIND_GB
1411
1412 ca21 fe 20 96          LDX  FINAL
1413 ca24 ff 20 8c          STX  BC

```

1414	ca27 fe 20 94		LDX	START
1415	ca2a ff 20 8e		STX	HL
1416	ca2d 86 00		LDAA	#300
1417	ca2f b7 20 4f		STAA	REGD
1418	ca32 18 ce 20 5a	CHAN1	LDY	#FIND_BUF
1419	ca36 fe 20 8e		LDX	HL
1420	ca39 b6 20 59		LDAA	LENGTH
1421	ca3c b7 20 50		STAA	REGE
1422	ca3f 18 a6 00	CHAN2	LDAA	0,Y
1423	ca42 a1 00		CMPA	0,X
1424	ca44 26 25		BNE	CHAN4
1425	ca46 08		INX	
1426	ca47 18 08		INY	
1427	ca49 7a 20 50		DEC	REGE
1428	ca4c 26 f1		BNE	CHAN2
1429				
1430	ca4e 18 ce 20 62		LDY	#CHAN_BUF
1431	ca52 fe 20 8e		LDX	HL
1432	ca55 b6 20 59		LDAA	LENGTH
1433	ca58 b7 20 50		STAA	REGE
1434	ca5b 7c 20 4f		INC	REGD
1435	ca5e 18 a6 00	CHAN3	LDAA	0,Y
1436	ca61 a7 00		STAA	0,X
1437	ca63 08		INX	
1438	ca64 18 08		INY	
1439	ca66 7a 20 50		DEC	REGE
1440	ca69 26 f3		BNE	CHAN3
1441				
1442	ca6b bc 20 8c	CHAN4	CPX	BC
1443	ca6e 27 0c		BEQ	CHAN5
1444	ca70 18 fe 20 8e		LDY	HL
1445	ca74 18 08		INY	
1446	ca76 18 ff 20 8e		STY	HL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1447	ca7a 20 b6		BRA	CHAN1
1448	ca7c b6 20 4f	CHAN5	LDA	REGD
1449	ca7f b7 20 a8		STAA	BUF+2
1450	ca82 bd d1 b6		JSR	UNPK
1451	ca85 ce 00 00		LDX	#\$0000
1452	ca88 ff 20 9e		STX	DISPY
1453	ca8b ff 20 a0		STX	DISPY+2
1454	ca8e c6 80		LDAB	#SLODLY
1455	ca90 bd d3 6f		JSR	SCAND
1456	ca93 bd d4 33		JSR	HBEEP
1457	ca96 7e c2 aa		JMP	MON
1458				
1459			***** COMPARE HEX FUNCTION *****	
1460				
1461	ca99 bd d2 e2	COMPH	JSR	INSF
1462	ca9c c6 76		LDAB	#\$76
1463	ca9e 18 ce 20 9c		LDY	#HEX
1464	caa2 bd d2 9e		JSR	IN2
1465	caa5 bd d3 d2		JSR	CLEAR
1466	caa8 fe 20 94		LDX	START
1467	caab 18 fe 20 96		LDY	FINAL
1468	caaf 18 ff 20 8c		STY	BC
1469	cab3 b6 20 9c		LDA	HEX
1470	cab6 b7 20 4f		STAA	REGD
1471	cab9 a6 00	COMPH1	LDA	0,X
1472	cabb b1 20 4f		CMPA	REGD
1473	cabe 26 0e		BNE	COMPH4
1474	cac0 bc 20 8c	COMPH2	CPX	BC
1475	cac3 27 03		BEQ	COMPH3
1476	cac5 08		INX	
1477	cac6 20 f1		BRA	COMPH1
1478	cac8 bd d4 33	COMPH3	JSR	HBEEP
1479	cacb 7e c2 aa		JMP	MON

1480	cace 3c	COMP4	PSHX
1481	cacf bd cf 0b		JSR ONFK
1482	cad2 38		PULX
1483	cad3 b6 20 9c		LDAA HEX
1484	cad6 b7 20 4f		STAA REGD
1485	cad9 20 e5		BRA COMP2
1486			
1487		*****	COMPARE FUNCTION *****
1488			
1489	cadb bd d2 e2	COMP	JSR INSF
1490	cade bd d2 f9		JSR IND
1491	cae1 18 fe 20 96		LDY FINAL
1492	cae5 18 ff 20 8c		STY BC
1493	cae9 18 fe 20 94		LDY START
1494	caed fe 20 98		LDX DEST
1495	caf0 18 a6 00	COMP1	LDAA 0, Y
1496	caf3 a1 00		CMPA 0, X
1497	caf5 26 11		BNE COMP4
1498	caf7 18 bc 20 8c	COMP2	CPY BC
1499	cafb 27 05		BEQ COMP3
1500	cafd 08		INX
1501	cafe 18 08		INY
1502	cb00 20 ee		BRA COMP1
1503	cb02 bd d4 33	COMP3	JSR HBEEP
1504	cb05 7e c2 aa		JMP MON
1505	cb08 3c	COMP4	PSHX
1506	cb09 18 3c		PSHY
1507	cb0b bd cf 0b		JSR ONFK
1508	cb0e 18 38		PULY
1509	cb10 38		PULX
1510	cb11 20 e4		BRA COMP2
1511			

```

1512                ***** MOVE DATA FUNCTION *****
1513
1514 cb13 bd d2 e2      MOVE          JSR  INSF
1515 cb16 bd d2 f9                JSR  IND
1516 cb19 18 fe 20 96          LDY  FINAL
1517 cb1d 18 ff 20 8c          STY  BC
1518 cb21 fe 20 94      MOVE_INC     LDX  START
1519 cb24 18 fe 20 98          LDY  DEST
1520 cb28 a6 00          MOVE_INC1   LDAA 0,X
1521 cb2a 18 a7 00                STAA 0,Y
1522 cb2d bc 20 8c                CPX  BC
1523 cb30 27 05                BEQ  LAST_MOVE
1524 cb32 18 08                INY
1525 cb34 08                INX
1526 cb35 20 f1                BRA  MOVE_INC1
1527 cb37 bd d4 33      LAST_MOVE   JSR  HBEEP
1528 cb3a 7e c2 aa                JMP  MON
1529
1530                ***** FIND FUNCTION *****
1531
1532 cb3d bd d2 e2      FIND          JSR  INSF
1533 cb40 bd cf 5b                JSR  FIND_GL
1534 cb43 18 ce 20 5a          LDY  #FIND_BUF
1535 cb47 bd cf 70                JSR  FIND_GB
1536 cb4a fe 20 96          LDX  FINAL
1537 cb4d ff 20 8c          STX  BC
1538 cb50 fe 20 94          LDX  START
1539 cb53 ff 20 8e          STX  HL
1540 cb56 18 ce 20 5a      FIND2     LDY  #FIND_BUF
1541 cb5a fe 20 8e          LDX  HL
1542 cb5d b6 20 59                LDAA LENGTH
1543 cb60 b7 20 50                STAA REGE
1544 cb63 18 a6 00      FIND3     LDAA 0,Y

```

1545	cb66 a1 00		CMPA 0,X
1546	cb68 26 11		BNE FIND4
1547	cb6a 18 08		INY
1548	cb6c 08		INX
1549	cb6d 7a 20 50		DEC REGE
1550	cb70 26 f1		BNE FIND3
1551	cb72 fe 20 8e		LDX HL
1552	cb75 bd cf 0b		JSR ONFK
1553	cb78 fe 20 8e		LDX HL
1554	cb7b bc 20 8c	FIND4	CPX BC
1555	cb7e 27 0c		BEQ FIND5
1556	cb80 18 fe 20 8e		LDY HL
1557	cb84 18 08		INY
1558	cb86 18 ff 20 8e		STY HL
1559	cb8a 20 ca		BRA FIND2
1560	cb8c bd d4 33	FIND5	JSR HBEEP
1561	cb8f 7e c2 aa		JMP MON
1562			
1563		*****	FILL FUNCTION *****
1564			
1565	cb92 bd d2 e2	FILL	JSR INSF
1566	cb95 c6 76		LDAB #76
1567	cb97 18 ce 20 9c		LDY #HEX
1568	cb9b bd d2 9e		JSR IN2
1569	cb9e fe 20 96		LDX FINAL
1570	cba1 ff 20 8c		STX BC
1571	cba4 fe 20 94		LDX START
1572	cba7 b6 20 9c		LDAA HEX
1573	cbaa 09		DEX
1574			
1575	cbab 08	FILL1	INX
1576	cbac a7 00		STAA 0,X
1577	cbae bc 20 8c		CPX BC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1578 cbb1 26 f8          BNE  FILL1
1579 cbb3 bd d4 33      JSR  HBEEP
1580 cbb6 7e c2 aa      JMP  MON
1581
1582                    ***** BREAK FUNCTION *****
1583
1584 cbb9 fe 20 90      BREAK      LDX  BRK
1585 cbbc c6 7c          LDAB  #$7C
1586 cbbe bd d2 7b      JSR  IN4
1587 cbc1 ff 20 90      STX  BRK
1588 cbc4 bd d4 33      JSR  HBEEP
1589 cbc7 7e c2 aa      JMP  MON
1590
1591                    ***** OUT FUNCTION *****
1592
1593 cbca c6 73          OUT      LDAB  #$73
1594 cbcc fe 20 9a      LDX  PORT
1595 cbcf bd d2 7b      JSR  IN4
1596 cbd2 ff 20 9a      STX  PORT
1597 cbd5 c6 76          LDAB  #$76
1598 cbd7 18 ce 20 9c   LDY  #HEX
1599 cbdb bd d2 9e      JSR  IN2
1600 cbde 18 a6 00      LDAA  0,Y
1601 cbe1 fe 20 9a      LDX  PORT
1602 cbe4 a7 00          STAA  0,X
1603 cbe6 20 e2          BRA  OUT
1604
1605                    ***** #STEP FUNCTION *****
1606
1607 cbe8 c6 54          GNSTEP  LDAB  #$54
1608 cbea 18 ce 20 9d   LDY  #NSTEP
1609 cbee bd d2 9e      JSR  IN2
1610 cbf1 bd d4 33      JSR  HBEEP

```

```

1611 cbf4 7e c2 aa          JMP  MON
1612
1613          ***** C/BREAK FUNCTION *****
1614
1615 cbf7 ce 00 00          CBREAK      LDX  #0000
1616 cbfa ff 20 90          STX  BRK
1617 cbfd c6 05            LDAB  #$05
1618 cbff f7 20 4e          STAB  REGC
1619 cc02 c6 20            LDAB  #$20
1620 cc04 18 ce d5 34          LDY  #CRTAB
1621 cc08 bd d3 57          JSR  PRINT
1622 cc0b bd d4 33          JSR  HBEEP
1623 cc0e 7e c2 aa          JMP  MON
1624
1625          ***** TRANSMIT FUNCTION *****
1626
1627 cc11 ce d5 df          TRAN      LDX  #TX_TAB
1628 cc14 bd cf bd          JSR  ROW_LCD
1629 cc17 86 30            LDAA  #$30
1630 cc19 b7 10 2b          STAA  BAUD
1631 cc1c 7f 10 2c          CLR  SCCR1
1632 cc1f 86 0c            LDAA  #$0C
1633 cc21 b7 10 2d          STAA  SCCR2
1634 cc24 ce d9 03          LDX  #TRANSMIT_TAB
1635 cc27 bd cf 8b          JSR  TX_SUB
1636 cc2a fe 20 94          LDX  START
1637 cc2d c6 6d            LDAB  #$6D
1638 cc2f bd d2 7b          JSR  IN4
1639 cc32 ff 20 94          STX  START
1640
1641 cc35 b6 10 2e          TRAN1     LDAA  SCSR
1642 cc38 85 80            BITA  #$80
1643 cc3a 27 f9            BEQ   TRAN1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1644 cc3c e6 00          LDAB  0,X
1645 cc3e c1 04          CMPB  #EOT
1646 cc40 27 09          BEQ   LTRAN
1647 cc42 f7 10 2f       STAB  SCDAT
1648 cc45 bd d0 47       JSR   DELAYL
1649 cc48 08             INX
1650 cc49 20 ea          BRA   TRAN1
1651 cc4b 7e c2 aa       LTRAN JMP  MON
1652
1653          ***** TEST PRINTER FUNCTION *****
1654
1655 cc4e ce d6 3f       TES_PRN LDX  #PRN_TAB
1656 cc51 bd cf bd       JSR   ROW_LCD
1657 cc54 fe 20 94       LDX  START
1658 cc57 c6 6d          LDAB  ##6D
1659 cc59 bd d2 7b       JSR   IN4
1660 cc5c ff 20 94       STX  START
1661
1662 cc5f 86 10          LDAA  ##10
1663 cc61 b7 17 12       STAA  PORT_STB
1664 cc64 a6 00       TES_PRN1 LDAA  0,X
1665 cc66 81 04          CMPA  #EOT
1666 cc68 27 09          BEQ   END_TPRN
1667 cc6a b7 20 ba       STAA  BUF_PRN
1668 cc6d bd ce 44       JSR   SEND
1669 cc70 08             INX
1670 cc71 20 f1          BRA   TES_PRN1
1671
1672 cc73 86 0a       END_TPRN LDAA  ##0A
1673 cc75 b7 20 ba       STAA  BUF_PRN
1674 cc78 bd ce 44       JSR   SEND
1675 cc7b 86 0d          LDAA  ##0D
1676 cc7d b7 20 ba       STAA  BUF_PRN

```

```

1677 cc80 bd ce 44          JSR  SEND
1678 cc83 7e c2 aa          JMP  MON
1679
1680          ***** SEGM FUNCTION *****
1681
1682 cc86 4f          SD          CLRA
1683 cc87 b7 20 b0          STAA  MEMY
1684
1685 cc8a bd d3 d2          SD1         JSR  CLEAR
1686 cc8d b6 20 b0          LDAA  MEMY
1687 cc90 b7 20 a7          STAA  BUF+1
1688 cc93 bd d1 b6          JSR  UNPK
1689 cc96 b6 20 b0          LDAA  MEMY
1690 cc99 b7 20 9e          STAA  DISPY
1691 cc9c 86 40          LDAA  #$40
1692 cc9e b7 20 9f          STAA  DISPY+1
1693 cca1 ce 00 00          LDX  #0000
1694 cca4 ff 20 a2          STX  DISPY+4
1695
1696 cca7 bd d0 50          SD2         JSR  SCAN
1697 ccaa ce d5 6c          LDX  #SDTAB
1698 ccad c6 09          LDAB  #09
1699 ccaf a1 00          SD3         CMPA  0,X
1700 ccb1 27 07          BEQ  SD4
1701 ccb3 08          INX
1702 ccb4 08          INX
1703 ccb5 5a          DECB
1704 ccb6 26 f7          BNE  SD3
1705 ccb8 20 ed          BRA  SD2
1706
1707 ccba 08          SD4         INX
1708 ccbb b6 20 b0          LDAA  MEMY
1709 ccbe a8 00          EORA  0,X

```

```

1710 ccc0 b7 20 b0          STAA  MEMY
1711 ccc3 20 c5            BRA   SD1
1712
1713                      ***** TEST 7 SEGMENT *****
1714
1715 ccc5 ce d6 2f          TES_SEG  LDX  #TSEG_TAB
1716 ccc8 bd cf c4          JSR   ROW_LCDL
1717 cccb ce d6 4f          LDX  #SEL_TAB1
1718 ccee bd cf bd          JSR   ROW_LCD
1719 ccd1 bd d3 d2          JSR   CLEAR
1720 ccd4 18 ce d5 56        LDY  #SEL_TAB
1721 ccd8 c6 06             LDAB  #$06
1722 ccda f7 20 4e          STAB  REGC
1723 ccdd c6 01             LDAB  #$01
1724 ccdf bd d3 57          JSR   PRINT
1725 cce2 bd d0 50          GET_NEW JSR   SCAN
1726 cce5 81 13             CMPA  #$13
1727 cce7 27 18             BEQ  TEST_SEG2
1728 cce9 81 11             CMPA  #$11
1729 cceb 26 f5             BNE  GET_NEW
1730 cced c6 07             TEST_SEG1 LDAB  #$07
1731 ccef 86 ff             LDAA  #$FF
1732 ccf1 b7 17 01          T_SEG11  STAA  SEGM
1733 ccf4 f7 17 02          STAB  DIGIT
1734 ccf7 bd d0 47          JSR   DELAYL
1735 ccfa 5a                DECB
1736 ccfb c1 ff             CMPB  #$FF
1737 ccfd 26 f2             BNE  T_SEG11
1738 ccff 20 ec             BRA  TEST_SEG1
1739 cd01 c6 00             TRST_SEG2 LDAB  #$00
1740 cd03 86 ff             LDAA  #$FF
1741 cd05 b7 17 01          T_SEG22  STAA  SEGM
1742 cd08 f7 17 02          STAB  DIGIT

```

```

1743 cd0b bd d0 47      JSR  DELAYL
1744 cd0e 5c            INCB
1745 cd0f c1 07        CMPB  #$07
1746 cd11 26 f2        BNE  T_SEG22
1747 cd13 20 ec        BRA  TEST_SEG2

```

1748

1749

***** RELATIVE FUNCTION *****

1750

```

1751 cd15 bd d2 e2      RELA  JSR  INSF

```

```

1752 cd18 fe 20 94      LDX  START

```

```

1753 cd1b fc 20 96      LDD  FINAL

```

1754

```

1755 cd1e 08            INX

```

```

1756 cd1f 08            INX

```

```

1757 cd20 ff 20 8c      STX  BC

```

```

1758 cd23 b3 20 8c      SUBD BC

```

```

1759 cd26 f7 20 54      STAB REGL

```

```

1760 cd29 b7 20 53      STAA REGH

```

```

1761 cd2c 09            DEX

```

```

1762 cd2d 4f            CLRA

```

1763

```

1764 cd2e b1 20 53      CMPA REGH

```

```

1765 cd31 27 e2         BEQ  RELA

```

```

1766 cd33 4a           DECA

```

```

1767 cd34 b1 20 53      CMPA REGH

```

```

1768 cd37 27 12         BEQ  RELA3

```

1769

1770

***** RELATIVE ERROR *****

1771

```

1772 cd39 86 ff         RELA1 LDAA  #$FF

```

```

1773 cd3b a7 00         STAA  0,X

```

```

1774 cd3d bd d4 1a      JSR  ERR

```

```

1775 cd40 20 d3         BRA  RELA

```

1776

1777

***** JUMP TO HIGHER *****

1778

1779 cd42 86 7f

RELA2

LDAA #\$7F

1780 cd44 b1 20 54

CMPA REGL

1781 cd47 25 f0

BCS RELA1

1782 cd49 20 0c

BRA RELA4

1783

1784

***** JUMP TO LOWER *****

1785

1786 cd4b b1 20 54

RELA3

CMPA REGL

1787 cd4e 27 e9

BEQ RELA1

1788 cd50 86 7f

LDAA #\$7F

1789 cd52 b1 20 54

CMPA REGL

1790 cd55 24 e2

BCC RELA1

1791

1792

***** LOAD RELATIVE TO MEM. *****

1793

1794 cd57 b6 20 54

RELA4

LDAA REGL

1795 cd5a a7 00

STAA 0,X

1796 cd5c b7 20 a8

STAA BUF+2

1797 cd5f bd d1 b6

JSR UNPK

1798 cd62 ce 00 00

LDX #0000

1799 cd65 ff 20 9e

STX DISPY

1800 cd68 ff 20 a0

STX DISPY+2

1801 cd6b c6 30

LDAB #MIDDLY

1802 cd6d bd d3 6f

JSR SCAND

1803 cd70 bd d4 33

JSR HBEEP

1804 cd73 20 a0

BRA RELA

1805

1806

***** POFF FUNCTION *****

1807

1808 cd75 c6 73

POFF

LDAB #\$73

```

1809 cd77 ce 00 00          LDX  #$0000
1810 cd7a bd d2 7b          JSR  IN4
1811 cd7d ff 20 55          STX  PASDIG
1812 cd80 bd d3 fd          POFFS JSR  CHKSR
1813 cd83 b7 20 57          STAA CHKSUM
1814 cd86 86 a3             LDAA #CODE
1815 cd88 b7 20 58          STAA BACKUP
1816 cd8b 4f                CLRA
1817 cd8c b7 20 b5          STAA PWCODE
1818 cd8f bd d4 33          JSR  HBEEP
1819 cd92 3e                WAI
1820
1821
1822          ***** CLEAR DISPLAY LCD *****
1823
1824 cd93 bd d0 37          C_LCD JSR  READY_L
1825 cd96 36                PSHA
1826 cd97 86 01             LDAA ##01
1827 cd99 b7 14 00          STAA SET1
1828 cd9c 32                PULA
1829 cd9d 7e c2 aa          JMP  MON
1830
1831          ***** IN DATA PORT FUNCTION *****
1832
1833 cda0 18 ce d5 46          IN    LDY  #LATTAB
1834 cda4 c6 05                LDAB  #$05
1835 cda6 f7 20 4e          STAB  REGC
1836 cda9 c6 01             LDAB  #01
1837 cdab bd d3 57          JSR  PRINT
1838 cdae bd d0 50          IN1   JSR  SCAN
1839 cdb1 81 13             CMPA  #$13
1840 cdb3 27 2a             BEQ   IN15
1841 cdb5 81 11             CMPA  #$11

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1842	cdb7 26 f5		BNE IN1
1843			
1844	cdb9 c6 73	IN11	LDAB #\$73
1845	cdbb fe 20 9a		LDX PORT
1846	cdbe bd d2 7b		JSR IN4
1847	cdc1 ff 20 9a		STX PORT
1848	cdc4 a6 00		LDAA 0,X
1849	cdc6 b7 20 a8		STAA BUF+2
1850	cdc9 b7 12 01		STAA PORTB
1851	cdcc bd d1 b6		JSR UNPK
1852	cdcf ce 00 00		LDX #0000
1853	cdd2 ff 20 9e		STX DISPY
1854	cdd5 ff 20 a0		STX DISPY+2
1855	cdd8 c6 30		LDAB #MIDDLY
1856	cdda bd d3 6f		JSR SCAND
1857	dddd 20 da		BRA IN11
1858			
1859	cddf c6 73	IN15	LDAB #\$73
1860	cde1 fe 20 9a		LDX PORT
1861	cde4 bd d2 7b		JSR IN4
1862	cde7 ff 20 9a		STX PORT
1863	cdea a6 00	IN16	LDAA 0,X
1864	cdec b7 20 a8		STAA BUF+2
1865	cdef b7 12 01		STAA PORTB
1866	cdf2 bd d1 b6		JSR UNPK
1867	cdf5 ce 00 00		LDX #0000
1868	cdf8 ff 20 9e		STX DISPY
1869	cdfb ff 20 a0		STX DISPY+2
1870	cdfe c6 10		LDAB #\$10
1871	ce00 bd d3 6f		JSR SCAND
1872	ce03 fe 20 9a		LDX PORT
1873	ce06 20 e2		BRA IN16

1874 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1875          ***** CLEAR RAM FUNCTION *****
1876
1877 ce08 bd d3 05          CRAM          JSR  READY
1878 ce0b ce 23 00          LDY  #RAM
1879 ce0e 4f          CLRA
1880 ce0f a7 00          CRAM1         STAA 0,X
1881 ce11 08          INX
1882 ce12 8c 90 00          CPX  #END_PRAM
1883 ce15 26 f8          BNE  CRAM1
1884 ce17 c6 05          LDAB #05
1885 ce19 f7 20 4e          STAB REGC
1886 ce1c c6 80          LDAB #80
1887 ce1e 18 ce d5 34          LDY  #CRTAB
1888 ce22 bd d3 57          JSR  PRINT
1889 ce25 bd d4 33          JSR  HBEEP
1890 ce28 7e c2 aa          JMP  MON
1891
1892          ***** RECALL SUBROUTINE *****
1893
1894 ce2b 36          RECALL        PSHA
1895 ce2c 81 2d          CMPA #2D
1896 ce2e 25 02          BCS  RECALL1
1897 ce30 32          PULA
1898 ce31 39          RTS
1899 ce32 37          RECALL1       PSMB
1900 ce33 18 3c          PSHY
1901 ce35 ce d4 7c          LDY  #CALL_TAB
1902 ce38 16          TAB
1903 ce39 bd d4 4f          JSR  TABLE
1904 ce3c 18 38          PULY
1905 ce3e 33          PULB
1906 ce3f 32          PULA
1907 ce40 ad 00          JSR  0,X

```

```

1908 ce42 01          NOP
1909 ce43 39          RTS
1910
1911          ***** SEND DATA TO PRINTER SUB. *****
1912
1913 ce44 bd ce 58    SEND          JSR  BUSY
1914 ce47 b6 20 ba    LDAA  BUF_PRN
1915 ce4a b7 17 11    STAA  PORT_DATA
1916 ce4d 86 00       LDAA  #$00
1917 ce4f b7 17 12    STAA  PORT_STB
1918 ce52 86 10       LDAA  #$10
1919 ce54 b7 17 12    STAA  PORT_STB
1920 ce57 39          RTS
1921
1922          ***** READ BUSY FLAG SUB. *****
1923
1924 ce58 b6 17 12    BUSY          LDAA  PORT_BUSY
1925 ce5b 85 01       BITA  #$01
1926 ce5d 26 f9       BNE  BUSY
1927 ce5f 39          RTS
1928
1929          ***** CHKRAM SUB. *****
1930          *      IN  = IX = HL = START ADDR      *
1931          *      = IY FINAL ADDR                *
1932
1933 ce60 18 fe 20 8c  CHKRAM          LDY  BC
1934 ce64 c6 01       LDAB  #$01
1935 ce66 18 3a       ABY
1936 ce68 18 8f       XGDY
1937 ce6a b3 20 8e    SUEB  HL
1938 ce6d fd 20 8c    STD  BC
1939 ce70 18 ce 20 b4  LDY  #SYSFAG
1940 ce74 18 1d 00 40  BCLR 0,Y $40

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1941		
1942	ce78 ff 20 8e	CHKRAM1 STX HL
1943	ce7b ff 20 a6	STX BUF
1944	ce7e bd d1 b6	JSR UNPK
1945	ce81 18 ce 00 00	LDY # \$0000
1946	ce85 18 ff 20 a2	STY DISPY+4
1947	ce89 c6 20	LDAB # \$20
1948	ce8b bd d3 6f	JSR SCAND
1949	ce8e fe 20 8e	LDX HL
1950	ce91 4f	CLRA
1951	ce92 b7 20 50	STAA REGE
1952	ce95 18 ce 01 00	LDY #256
1953	ce99 18 8f	XGDY
1954	ce9b b3 20 8c	SUBD BC
1955	ce9e 25 0e	BCS CHKRAM2
1956	cea0 18 ce 20 b4	LDY #SYSFAG
1957	cea4 18 1c 00 40	BSET 0,Y \$40
1958	cea8 f6 20 8d	LDAB BC+1
1959	ceab f7 20 50	STAB REGE
1960		
1961	ceae f6 20 50	CHKRAM2 LDAB REGE
1962	ceb1 f7 20 4f	STAB REGD
1963	ceb4 4f	CLRA
1964		
1965	ceb5 ff 20 8e	STX HL
1966	ceb8 a7 00	CHKRAM3 STAA 0,X
1967	ceba 08	INX
1968	cebb 18 fe 20 8c	LDY BC
1969	cebf 18 09	DEY
1970	cec1 18 ff 20 8c	STY BC
1971	cec5 4c	INCA
1972	cec6 7a 20 4f	DEC REGD
1973	cec9 26 ed	BNE CHKRAM3

1974		
1975	cecb b6 20 50	LDAA REGE
1976	cece b7 20 4f	STAA REGD
1977	ced1 4f	CLRA
1978	ced2 fe 20 8e	LDX HL
1979	ced5 a1 00	CHKRAM4 CMPA 0,X
1980	ced7 26 1a	BNE CHKRAM8
1981	ced9 5f	CLRB
1982	ceda e7 00	STAB 0,X
1983	cedc 08	INX
1984	cedd 4c	INCA
1985	cede 7a 20 4f	DEC REGD
1986	cee1 26 f2	BNE CHKRAM4
1987		
1988	cee3 18 ce 20 b4	CHKRAM5 LDY #SYSFAG
1989	cee7 18 e6 00	LDAB 0,Y
1990	ceea c5 40	BITB #340
1991	ceec 27 8a	BEQ CHKRAM1
1992	cee 18 1d 00 40	BCLR 0,Y 340
1993	cef2 39	RTS
1994		
1995	cef3 3c	CHKRAM8 PSHX
1996	cef4 18 3c	PSHY
1997	cef6 fe 20 8e	LDX HL
1998	cef9 bd cf 0b	JSR ONFK
1999	cefc 18 fe 20 8e	LDY HL
2000	cf00 18 08	INY
2001	cf02 18 ff 20 8e	STY HL
2002	cf06 18 38	PUI.Y
2003	cf08 38	PULX
2004	cf09 20 d8	BRA CHKRAM5
2005		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2006                ***** ON FUNC.KY SUB. *****
2007                *   DISPLAY IX ON DISPLAY & WAIT   *
2008                *           FOR INC IN IX DATA     *
2009
2010 cf0b ff 20 a6      ONFK          STX  BUF
2011 cf0e bd d1 b6                JSR  UNPK
2012 cf11 ce 20 a1                LDX  #DISPY+3
2013 cf14 1c 00 80                BSET  0,X $80
2014 cf17 ce 00 00                LDX  #$0000
2015 cf1a ff 20 a2                STX  DISPY+4
2016
2017 cf1d ce 20 b4      ONFK1        LDX  #SYSFAG
2018 cf20 1c 00 10                BSET  0,X $10
2019 cf23 3c                PSHX
2020 cf24 bd d0 50                JSR  SCAN
2021 cf27 38                PULX
2022 cf28 1d 00 10                BCLR  0,X $10
2023
2024 cf2b 81 10                CMPA  #$10
2025 cf2d 26 03                BNE  ONFK2
2026 cf2f 7e c3 0a                JMP  MONF
2027 cf32 81 12      ONFK2        CMPA  #$12
2028 cf34 26 03                BNE  ONFK3
2029 cf36 7e c5 4c                JMP  RUN1
2030 cf39 81 14      ONFK3        CMPA  #$14
2031 cf3b 27 15                BEQ  ONFK7
2032 cf3d 81 15                CMPA  #$15
2033 cf3f 26 03                BNE  ONFK4
2034 cf41 7e c5 cc                JMP  STEP1
2035 cf44 81 16      ONFK4        CMPA  #$16
2036 cf46 26 03                BNE  ONFK5
2037 cf48 7e c2 aa                JMP  MON
2038 cf4b 81 13      ONFK5        CMPA  #$13

```

```

2039 cf4d 26 01          BNE  ONFK6
2040 cf4f 39            RTS
2041 cf50 20 cb          ONFK6   BRA  ONFK1
2042 cf52 ce 20 a1      ONFK7   LDX  #DISPY+3
2043 cf55 1d 00 80      BCLR  0,X $80
2044 cf58 7e c3 6a      JMP  MEM
2045
2046                    ***** GET LENGTH SUB. *****
2047                    *          GET DATA FROM 1 TO 8          *
2048
2049 cf5b c6 38          FIND_GL  LDAB  #$38
2050 cf5d 18 ce 20 59      LDY  #LENGTH
2051 cf61 bd d2 9e          JSR  IN2
2052 cf64 18 a6 00          LDAA  0,Y
2053 cf67 81 00          CMPA  #$00
2054 cf69 27 f0          BEQ  FIND_GL
2055 cf6b 81 09          CMPA  #$09
2056 cf6d 24 ec          BCC  FIND_GL
2057 cf6f 39            RTS
2058
2059                    ***** GET DATA BLOCK SUB. *****
2060                    *      IN = (LENGTH)                          *
2061                    *      = IY START BUFFER FIND DATA          *
2062                    *      OUT = (IY)                              *
2063
2064 cf70 86 01          FIND_GB  LDAA  #$01
2065 cf72 f6 20 59      LDAB  LENGTH
2066 cf75 36            FIND_GB1  PSHA
2067 cf76 37            PSHB
2068 cf77 18 3c          PSHY
2069 cf79 bd d1 e7      JSR  UNPKT
2070 cf7c 16            TAB
2071 cf7d 18 38          PULY

```

```

2072 cf7f bd d2 9e          JSR  IN2
2073 cf82 18 08            INY
2074 cf84 33              PULB
2075 cf85 32              PULA
2076 cf86 4c              INCA
2077 cf87 5a              DECB
2078 cf88 26 eb          BNE  FIND_GB1
2079 cf8a 39              RTS
2080
2081                      ***** TRANSMIT SUB. *****
2082
2083 cf8b 86 30          TX_SUB  LDAA  #$30
2084 cf8d b7 10 2b      STAA  BAUD
2085 cf90 7f 10 2c      CLR   SCCR1
2086 cf93 86 0c          LDAA  #$0C
2087 cf95 b7 10 2d      STAA  SCCR2
2088
2089 cf98 b6 10 2e      TX_SUB1 LDAA  SCSR
2090 cf9b 85 80          BITA  #$80
2091 cf9d 27 f9          BEQ   TX_SUB1
2092 cf9f e6 00          LDAB  0,X
2093 cfa1 c1 04          CMPB  #EOT
2094 cfa3 27 06          BEQ   LTX_SUB
2095 cfa5 f7 10 2f      STAB  SCDAT
2096 cfa8 08              INX
2097 cfa9 20 ed          BRA   TX_SUB1
2098 cfab 39              LTX_SUB RTS
2099
2100                      ***** MOVE ASCII TABLE *****
2101                      *      IX = START ASCII TABLE      *
2102                      *      IY = NEW ADDRESS      *
2103
2104 cfac 36          M_ASCII  PSHA

```

```

2105 cfad a6 00          M_ASCII1      LDAA  0,X
2106 cfaf 81 04          CMPA  #EOT
2107 cfb1 27 08          BEQ   LM_ASCII
2108 cfb3 18 a7 00      STAA  0,Y
2109 cfb6 08            INX
2110 cfb7 18 08          INY
2111 cfb9 20 f2          BRA   M_ASCII1
2112 cfbb 32            LM_ASCII  PULA
2113 cfbc 39            RTS
2114
2115          ***** WRITE CHARACTER FROM TABLE *****
2116          *                IX = ADDRESS (TABLE)                *
2117
2118 cfbd bd cf d5          ROW_LCD      JSR   INIT_LCD
2119 cfc0 bd cf f8          WRP          JSR   WRP
2120 cfc3 39          RTS
2121
2122          ***** WRITE LCD & DELAY *****
2123
2124 cfc4 37          ROW_LCDL     PSHB
2125 cfc5 c6 06          LDAB  #$06
2126 cfc7 bd cf d5          WRP          JSR   INIT_LCD
2127 cfca bd cf f8          WRP          JSR   WRP
2128 cfd d bd d0 47      ROW_LCDL1    JSR   DELAYL
2129 cfd0 5a          DECB
2130 cfd1 26 fa          BNE   ROW_LCDL1
2131 cfd3 33          PULB
2132 cfd4 39          RTS
2133
2134          ***** INITIAL LCD DISPLAY *****
2135
2136 cfd5 36          INIT_LCD     PSHA
2137 cfd6 86 0f          LDAA  #$0F

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2138 cfd8 b7 14 00          STAA  SET1
2139 cfdb bd d0 41          JSR   DELAY_L
2140 cfde 86 38              LDAA  #$38
2141 cfe0 b7 14 00          STAA  SET1
2142 cfe3 bd d0 41          JSR   DELAY_L
2143 cfe6 86 06              LDAA  #$06
2144 cfe8 b7 14 00          STAA  SET1
2145 cfef bd d0 41          JSR   DELAY_L
2146 cfef 86 01              LDAA  #$01
2147 cff0 b7 14 00          STAA  SET1
2148 cff3 bd d0 41          JSR   DELAY_L
2149 cff6 32                  PULA
2150 cff7 39                  RTS
2151
2152          ***** WRITE LCD 16 CHAR * 1 ROW *****
2153
2154 cff8 36          WRP          PSHA
2155 cff9 4f          CLRA
2156 cffa bd d0 15    JSR   CURSOR_L
2157 cffd bd d0 27    JSR   OUT_LINE
2158 d000 86 40       LDAA  #$40
2159 d002 bd d0 15    JSR   CURSOR_L
2160 d005 bd d0 27    JSR   OUT_LINE
2161 d008 32          PULA
2162 d009 39          RTS
2163
2164          ***** CLEAR DISPLAY LCD *****
2165
2166 d00a bd d0 37    CLEAR_L    JSR   READY_L
2167 d00d 36          PSHA
2168 d00e 86 01       LDAA  #$01
2169 d010 b7 14 00    STAA  SET1
2170 d013 32          PULA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2171 d014 39          RTS
2172
2173          ***** CURSOR LOCATION *****
2174          *          A = ADDRESS          *
2175
2176 d015 bd d0 37    CURSOR_L    JSR  READY_L
2177 d018 36          PSHA
2178 d019 8b 80          ADDA  #$80
2179 d01b b7 14 00    STAA  SET1
2180 d01e 32          PULA
2181 d01f 39          RTS
2182
2183          ***** WRITE BYTE OF LCD *****
2184          *          A = DATA (ASCII)      *
2185
2186 d020 bd d0 37    OUT_L    JSR  READY_L
2187 d023 b7 14 01    STAA  WR1LCD
2188 d026 39          RTS
2189
2190          ***** WRITE LINE OF LCD *****
2191          *          IX = TABLE OF DATA    *
2192
2193 d027 37          OUT_LINE    PSHB
2194 d028 36          PSHA
2195 d029 c6 08          LDAB  #$08
2196 d02b a6 00          OUT1_LINE    LDAA  0,X
2197 d02d bd d0 20    JSR  OUT_L
2198 d030 08          INX
2199 d031 5a          DECB
2200 d032 26 f7          BNE  OUT1_LINE
2201 d034 32          PULA
2202 d035 33          PULB
2203 d036 39          RTS

```

```

2204
2205          ***** READ BUSY FLAG *****
2206
2207 d037 36          READY_L          PSHA
2208 d038 b6 14 00    READY1_L         LDAA  SET1
2209 d03b 84 80          ANDA  #$80
2210 d03d 26 f9          BNE  READY1_L
2211 d03f 32          PULA
2212 d040 39          RTS
2213
2214          ***** DELAY *****
2215
2216 d041 86 00    DELAY_L          LDAA  #$00
2217 d043 4a    DELAY1_L         DECA
2218 d044 26 fd    BNE  DELAY1_L
2219 d046 39          RTS
2220
2221 d047 3c    DELAYL          PSHX
2222 d048 ce 00 00    LDX  #$0000
2223 d04b 09    DELAYL1         DEX
2224 d04c 26 fd    BNE  DELAYL1
2225 d04e 38          PULX
2226 d04f 39          RTS
2227
2228          ***** SCAN SUB. *****
2229          *IN = (DISPY)          *
2230          * = B4 OF (SYSFAG) 0=CHECK-INSTANT *
2231          * KEY 1=DON'T - CARE          *
2232          *OUT= A, (KEYIN)          *
2233
2234 d050 c6 08          SCAN          LDAB  #$08
2235 d052 f7 20 4e          STAB  REGC
2236 d055 86 00          LDAA  #$00

```

2237	d057 b7 20 50		STAA REGE
2238	d05a ce 20 9e		LDX #DISPY
2239			
2240	d05d bd d1 40	SCAN1	JSR SCANS
2241	d060 ff 20 ae		STX MEMX
2242	d063 b6 17 02		LDAA DIGIT
2243	d066 84 70		ANDA ##70
2244	d068 81 70		CMPA ##70
2245	d06a 26 1f		BNE SCAN3
2246			
2247	d06c 7a 20 4e		DEC REGC
2248	d06f 26 0f		BNE SCAN2
2249	d071 ce 20 b4		LDX #SYSFAG
2250	d074 1d 00 01		BCLR 0,X \$01
2251	d077 ce 20 b3		LDX #REPDLY
2252	d07a 36		PSHA
2253	d07b 86 60		LDAA #REP1
2254	d07d a7 00		STAA 0,X
2255	d07f 32		PULA
2256			
2257	d080 fe 20 ae	SCAN2	LDX MEMX
2258	d083 7c 20 50		INC REGE
2259	d086 5a		DECB
2260	d087 26 d4		BNE SCAN1
2261	d089 20 c5		BRA SCAN
2262			
2263			***** KEY PRESS *****
2264			
2265	d08b ba 20 50	SCAN3	ORAA REGE
2266	d08e 37		PSHB
2267	d08f ce d5 94		LDX #KEYTAB+22
2268	d092 c6 16		LDAB ##16
2269	d094 a1 00	SCAN31	CMPA 0,X

2270	d096 27 04		BEQ	SCAN32
2271	d098 09		DEX	
2272	d099 5a		DECB	
2273	d09a 26 f8		BNE	SCAN31
2274	d09c f7 20 4f	SCAN32	STAB	REGD
2275	d09f 33		PULB	
2276	d0a0 ce 20 b4		LDX	#SYSFAG
2277	d0a3 a6 00		LDAA	0,X
2278	d0a5 85 01		BITA	#\$01
2279	d0a7 27 25		BEQ	SCAN5
2280				
2281	d0a9 86 07		LDAA	#\$07
2282	d0ab b1 20 50		CMPA	REGE
2283	d0ae 27 08		BEQ	SCAN4
2284	d0b0 b6 20 4f		LDAA	REGD
2285	d0b3 bd d1 2f		JSR	SCANC
2286	d0b6 25 c8		BCS	SCAN2
2287	d0b8 b6 20 b2	SCAN4	LDAA	KEYIN
2288	d0bb bd d1 2f		JSR	SCANC
2289	d0be 25 c0		BCS	SCAN2
2290				
2291		*****	AUTO	REPEAT *****
2292				
2293	d0c0 ce 20 b3		LDX	#REPDLY
2294	d0c3 6a 00		DEC	0,X
2295	d0c5 26 b9		BNE	SCAN2
2296	d0c7 36		PSHA	
2297	d0c8 86 16		LDAA	#REP2
2298	d0ca a7 00		STAA	0,X
2299	d0cc 32		PULA	
2300	d0cd 39		RTS	
2301				

```

2302                ***** NEW KEY *****
2303
2304 d0ce b6 20 4f          SCAN5          LDAA  REGD
2305 d0d1 b7 20 b2                STAA  KEYIN
2306 d0d4 1c 00 01                BSET  0,X $01
2307 d0d7 e6 00                LDAB  0,X
2308 d0d9 c5 08                BITB  $$08
2309 d0db 26 20                BNE   SCAN7
2310
2311 d0dd ce 04 00                LDX   $$0400
2312 d0e0 b7 20 4e                STAA  REGC
2313 d0e3 b6 20 b6          LDAA  GAIN
2314 d0e6 16                TAB
2315 d0e7 b6 20 4e          LDAA  REGC
2316 d0ea 36                PSHA
2317 d0eb 86 10          LDAA  #HIGFRE
2318 d0ed b7 20 4e          STAA  REGC
2319 dof0 32                PULA
2320 dof1 81 10          CMPA  $$10
2321 dof3 24 05          BCC   SCAN6
2322 dof5 86 14          LDAA  #LOWFRE
2323 dof7 b7 20 4e          STAA  REGC
2324 dof8 bd d3 2d          SCAN6          JSR   SOUND
2325
2326                ***** INSTANT KEY CHECK *****
2327
2328 dof8 b6 20 b2          SCAN7          LDAA  KEYIN
2329 d100 ce 20 b4                LDX   #SYSFAG
2330 d103 e6 00                LDAB  0,X
2331 d105 c5 10                BITB  $$10
2332 d107 27 01                BEQ   SCAN71
2333 d109 39                RTS
2334 d10a 81 10          SCAN71          CMPA  $$10

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2335 d10c 26 03		BNE	SCAN711
2336 d10e 7e c3 0a		JMP	MONF
2337 d111 81 12	SCAN711	CMPA	#\$12
2338 d113 26 03		BNE	SCAN712
2339 d115 7e c5 4c		JMP	RUN1
2340 d118 81 14	SCAN712	CMPA	#\$14
2341 d11a 26 03		BNE	SCAN713
2342 d11c 7e c3 7a		JMP	MEM1
2343 d11f 81 15	SCAN713	CMPA	#\$15
2344 d121 26 03		BNE	SCAN714
2345 d123 7e c5 cc		JMP	STEP1
2346 d126 81 16	SCAN714	CMPA	#\$16
2347 d128 26 03		BNE	SCAN715
2348 d12a 7e c3 d9		JMP	REG
2349 d12d 01	SCAN715	NOP	
2350 d12e 39		RTS	
2351			
2352		***** AUTO-REPEAT-KEY CHECK SUB. *****	
2353			
2354 d12f 81 11	SCANC	CMPA	#\$11
2355 d131 26 01		BNE	SCANC1
2356 d133 39		RTS	
2357 d134 81 13	SCANC1	CMPA	#\$13
2358 d136 26 01		BNE	SCANC2
2359 d138 39		RTS	
2360 d139 81 15	SCANC2	CMPA	#\$15
2361 d13b 26 01		BNE	SCANC3
2362 d13d 39		RTS	
2363 d13e 0d	SCANC3	SEC	
2364 d13f 39		RTS	
2365			

```

2366                ***** SCAN DISPLAY SUB. *****
2367
2368 d140 b6 20 50          SCANS          LDAA REGE
2369 d143 b7 17 02          STAA DIGIT
2370 d146 a6 00            LDAA 0,X
2371 d148 b7 17 01          STAA SEGM
2372 d14b 4f                CLRA
2373 d14c 4a                SCANS1        DECA
2374 d14d 26 fd            BNE SCANS1
2375 d14f b7 17 01          STAA SEGM
2376 d152 08                INX
2377 d153 39                RTS
2378
2379                ***** PACK SUB. *****
2380                *          IN = (DISPY)          *
2381                *          OUT = (BUF)         *
2382
2383 d154 18 ce 20 9e        PACK          LDY #DISPY
2384 d158 ce 20 a6          LDX #BUF
2385 d15b bd d1 63          JSR PACKS
2386 d15e 08                INX
2387 d15f bd d1 63          JSR PACKS
2388 d162 08                INX
2389
2390 d163 86 02            PACKS          LDAA #$02
2391 d165 b7 20 4e          STAA REGC
2392 d168 18 a6 00          PACKS1        LDAA 0,Y
2393 d16b 3c                PSHX
2394 d16c bd d1 a7          JSR PACKT
2395 d16f 38                PULX
2396 d170 c6 04            LDAB #$04
2397 d172 f7 20 4f          STAB REGD
2398 d175 e6 00            LDAB 0,X

```

2399	d177 b7 20 53		STAA	REGH
2400	d17a f7 20 54		STAB	REGL
2401				
2402	d17d 05	PACKS11	LSLD	
2403	d17e 7a 20 4f		DEC	REGD
2404	d181 26 fa		BNE	PACKS11
2405	d183 36		PSHA	
2406	d184 37		PSHB	
2407	d185 b6 20 53		LDAA	REGH
2408	d188 84 f0		ANDA	#\$FO
2409	d18a f6 20 53		LDAB	REGH
2410	d18d c4 0f		ANDB	#\$OF
2411	d18f b7 20 53		STAA	REGH
2412	d192 f7 20 54		STAB	REGL
2413	d195 33		PULB	
2414	d196 32		PULA	
2415	d197 ba 20 53		ORAA	REGH
2416	d19a fa 20 54		ORAB	REGL
2417	d19d e7 00		STAB	0,X
2418	d19f 18 08		INY	
2419	d1a1 7a 20 4e		DEC	REGC
2420	d1a4 26 c2		BNE	PACKS1
2421	d1a6 39		RTS	
2422				
2423	d1a7 ce d5 a4	PACKT	LDX	#\$SEGTAB+15
2424	d1aa c6 0f		LDAB	#\$OF
2425				
2426	d1ac a1 00	PACKT1	CMPA	0,X
2427	d1ae 27 04		BEQ	PACKT2
2428	d1b0 09		DEX	
2429	d1b1 5a		DECB	
2430	d1b2 26 f8		BNE	PACKT1

2431

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2432 d1b4 17          PACKET2          TBA
2433 d1b5 39          RTS
2434
2435
2436          ***** UNPACK SUB.*****
2437          *              IN = (BUF)          *
2438          *              OUT = (DISPY)       *
2439
2440 d1b6 18 ce 20 9e  UNPK          LDY #DISPY
2441 d1ba ce 20 a6          LDX #BUF
2442 d1bd bd d1 c5          JSR UNPKS
2443 d1c0 08          INX
2444 d1c1 bd d1 c5          JSR UNPKS
2445 d1c4 08          INX
2446
2447 d1c5 c6 02          UNPKS          LDAB #$02
2448 d1c7 4f          UNPKS1          CLRA
2449 d1c8 37          PSHB
2450 d1c9 c6 04          LDAB #$04
2451 d1cb f7 20 4e          STAB REGC
2452 d1ce e6 00          LDAB 0,X
2453 d1d0 05          UNPK11          LSLD
2454 d1d1 7a 20 4e          DEC REGC
2455 d1d4 26 fa          BNE UNPK11
2456 d1d6 e7 00          STAB 0,X
2457 d1d8 33          PULB
2458 d1d9 3c          PSHX
2459 d1da bd d1 e7          JSR UNPKT
2460 d1dd 38          PULX
2461 d1de 18 a7 00          STAA 0,Y
2462 d1e1 18 08          INY
2463 d1e3 5a          DECB
2464 d1e4 26 e1          BNE UNPKS1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2465	d1e6 39		RTS
2466			
2467	d1e7 36	UNPKT	PSHA
2468	d1e8 37		PSHB
2469	d1e9 ce d5 95		LDX #SEGTAB
2470	d1ec 8f		XGDX
2471	d1ed b7 20 53		STAA REGH
2472	d1f0 f7 20 54		STAB REGL
2473	d1f3 33		PULB
2474	d1f4 32		PULA
2475	d1f5 bb 20 54		ADDA REGL
2476	d1f8 b7 20 54		STAA REGL
2477	d1fb 86 00		LDA #00
2478	d1fd b9 20 53		ADCA REGH
2479	d200 b7 20 53		STAA REGH
2480	d203 b6 20 53		LDA REGH
2481	d206 f6 20 54		LDAB REGL
2482	d209 8f		XGDX
2483	d20a a6 00		LDA 0,X
2484	d20c 39		RTS
2485			
2486		*****	GET FOUR SUB.*****
2487		*	GET 4 DIGIT CHECK *
2488		*	IN = (DISPY) *
2489			
2490	d20d 81 11	GET4	CMPA #\$11
2491	d20f 27 2f		BEQ GET43
2492	d211 81 10		CMPA #\$10
2493	d213 25 01		BCS GET401
2494	d215 39		RTS
2495			
2496	d216 bd d1 e7	GET401	JSR UNPKT
2497	d219 b7 20 4f		STAA REGD

2498	d21c ce 20 9e		LDX #DISPY
2499	d21f a6 00		LDAA 0,X
2500	d221 81 40		CMPA ##40
2501	d223 27 13		BEQ GET42
2502	d225 81 00		CMPA ##00
2503	d227 27 0f		BEQ GET42
2504	d229 c6 04		LDAB ##04
2505			
2506	d22b 08	GET41	INX
2507	d22c a6 00		LDAA 0,X
2508	d22e 5a		DECB
2509	d22f 26 03		BNE GET411
2510	d231 bd d3 d2		JSR CLEAR
2511			
2512	d234 81 00	GET411	CMPA ##00
2513	d236 26 f3		BNE GET41
2514	d238 f6 20 4f	GET42	LDAB REGD
2515	d23b e7 00		STAB 0,X
2516	d23d c6 ff		LDAB ##FF
2517	d23f 39		RTS
2518			
2519			***** DEC KEY *****
2520			
2521	d240 ce 20 a4	GET43	LDX #DISPY+6
2522	d243 c6 00		LDAB ##00
2523	d245 e7 00		STAB 0,X
2524	d247 09		DEX
2525	d248 09		DEX
2526	d249 c6 04		LDAB ##04
2527			
2528	d24b 09	GET44	DEX
2529	d24c a6 00		LDAA 0,X
2530	d24e 5a		DECB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2531 d24f 27 04          BEQ  GET45
2532 d251 81 00          CMPA  #$00
2533 d253 27 f6          BEQ  GET44
2534
2535 d255 36              GET45    PSHA
2536 d256 86 00          LDAA  #$00
2537 d258 a7 00          STAA  0,X
2538 d25a 32              PULA
2539 d25b 39              RTS
2540
2541                      ***** GET TWO SUB *****
2542                      *          GET 2. DIGIT CHECK          *
2543                      *          IN = (DISPY)                *
2544
2545 d25c 3c              GET2    PSHX
2546 d25d bd d1 e7       JSR  UNPKT
2547 d260 38              PULX
2548 d261 b7 20 4f      STAA  REGD
2549
2550 d264 08              INX
2551 d265 a6 00          LDAA  0,X
2552 d267 81 00          CMPA  #$00
2553 d269 27 08          BEQ  GET21
2554 d26b 36              PSHA
2555 d26c 86 00          LDAA  #$00
2556 d26e a7 00          STAA  0,X
2557 d270 32              PULA
2558 d271 09              DEX
2559 d272 0d              SEC
2560
2561 d273 36              GET21    PSHA
2562 d274 b6 20 4f      LDAA  REGD
2563 d277 a7 00          STAA  0,X

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2564 d279 32          PULA
2565 d27a 39          RTS
2566
2567          ***** INPUT/FOUR SUB.*****
2568          *           IN = B CHARACTOR           *
2569          *           = IX DATA                 *
2570          *           OUT = IX                   *
2571
2572 d27b ff 20 a6     IN4          STX  BUF
2573 d27e 86 40          LDAA  #$40
2574 d280 8f          XGDX
2575 d281 ff 20 b0     STX  MEMY
2576 d284 bd d1 b6     JSR  UNPK
2577
2578 d287 fe 20 b0     IN41         LDX  MEMY
2579 d28a ff 20 a2     STX  DISPY+4
2580 d28d bd d0 50     JSR  SCAN
2581 d290 bd d2 0d     JSR  GET4
2582 d293 81 13       CMPA  #$13
2583 d295 26 f0       BNE  IN41
2584
2585 d297 bd d1 54     JSR  PACK
2586 d29a fe 20 a6     LDX  BUF
2587 d29d 39          RTS
2588
2589          ***** INPUT/TWO SUB.*****
2590          *           IN = B CHARACTOR           *
2591          *           = (IY) DATA                 *
2592          *           OUT = (IY)                   *
2593
2594 d29e 37          IN2          PSHB
2595 d29f 18 a6 00     LDAA  0,Y
2596 d2a2 b7 20 a6     STAA  BUF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2597	d2a5 18 3c		PSHY
2598	d2a7 bd d1 b6		JSR UNPK
2599	d2aa 18 38		PULY
2600	d2ac 33		PULB
2601	d2ad 86 40		LDAA #\$40
2602	d2af 8f		XGDX
2603	d2b0 ff 20 a2		STX DISPY+4
2604	d2b3 ce 00 00		LDX #\$0000
2605	d2b6 ff 20 a0		STX DISPY+2
2606			
2607	d2b9 18 3c	IN21	PSHY
2608	d2bb bd d0 50		JSR SCAN
2609	d2be 18 38		PULY
2610	d2c0 81 11		CMPA #\$11
2611	d2c2 27 f5		BEQ IN21
2612	d2c4 81 13		CMPA #\$13
2613	d2c6 27 0c		BEQ IN22
2614			
2615	d2c8 ce 20 9e		LDX #DISPY
2616	d2cb 18 3c		PSHY
2617	d2cd bd d2 5c		JSR GET2
2618	d2d0 18 38		PULY
2619	d2d2 20 e5		BRA IN21
2620			
2621	d2d4 18 3c	IN22	PSHY
2622	d2d6 bd d1 54		JSR PACK
2623	d2d9 18 38		PULY
2624	d2db b6 20 a6		LDAA BUF
2625	d2de 18 a7 00		STAA 0,Y
2626	d2e1 39		RTS
2627			

```

2628                ***** INPUT/START,FINAL SUB.*****
2629                *           OUT = (START)           *
2630                *           = (FINAL)             *
2631
2632 d2e2 fe 20 94    INSF           LDX  START
2633 d2e5 c6 6d              LDAB  #$6D
2634 d2e7 bd d2 7b              JSR  IN4
2635 d2ea ff 20 94              STX  START
2636
2637 d2ed fe 20 96              LDX  FINAL
2638 d2f0 c6 71              LDAB  #$71
2639 d2f2 bd d2 7b              JSR  IN4
2640 d2f5 ff 20 96              STX  FINAL
2641 d2f8 39                  RTS
2642
2643                ***** INPUT/DESTINATION SUB.*****
2644                *           OUT = (DEST)           *
2645
2646 d2f9 fe 20 98    IND           LDX  DEST
2647 d2fc c6 5e              LDAB  #$5E
2648 d2fe bd d2 7b              JSR  IN4
2649 d301 ff 20 98              STX  DEST
2650 d304 39                  RTS
2651
2652                ***** READY SUB.*****
2653
2654 d305 18 ce d5 41    READY      LDY  #RDYTAB
2655 d309 c6 05              LDAB  #$05
2656 d30b f7 20 4e              STAB  REGC
2657 d30e c6 80              LDAB  #$80
2658 d310 bd d3 57              JSR  PRINT
2659 d313 bd d0 50          READY1   JSR  SCAN
2660 d316 81 13            CMPA  #$13

```

```

2661 d318 26 f9          BNE  READY1
2662 d31a 39            RTS
2663
2664          ***** SOUND INC SUB. *****
2665          *          IN = REGC FREQUENCY          *
2666          *          = IX LENGTH                    *
2667
2668 d31b c6 00          SOUNDI      LDAB  #00
2669 d31d bd d3 2d      SNDD1      JSR  SOUND
2670 d320 5c              INCB
2671 d321 26 fa          BNE  SNDD1
2672 d323 39            RTS
2673
2674          ***** SOUND DEC SUB. *****
2675          *          IN = REGC FREQUENCY          *
2676          *          = IX LENGTH                    *
2677
2678
2679 d324 c6 ff          SOUNDDD   LDAB  #$FF
2680 d326 bd d3 2d      SNDD1      JSR  SOUND
2681 d329 5a              DECB
2682 d32a 26 fa          BNE  SNDD1
2683 d32c 39            RTS
2684
2685          ***** SOUND SUB. *****
2686          *          IN = B LEVEL                    *
2687          *          = REGC FREQUENCY                *
2688          *          = IX LENGTH                    *
2689
2690 d32d 3c              SOUND      PSHX
2691 d32e 86 00          SOUND1     LDAA  #$00
2692 d330 b7 20 51      STAA  REGDS
2693 d333 17            TBA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2694 d334 bd d3 42		JSR SOUND2
2695 d337 4f		CLRA
2696 d338 bd d3 42		JSR SOUND2
2697 d33b 7a 20 51		DEC REGDS
2698 d33e 26 ee		BNE SOUND1
2699 d340 38		PULX
2700 d341 39		RTS
2701		
2702 d342 b7 17 00	SOUND2	STAA ANALOG
2703 d345 b6 20 4e		LDAA REGC
2704 d348 b7 20 52		STAA REGES
2705 d34b 09	SOUND3	DEX
2706 d34c 26 03		BNE SOUND4
2707 d34e 7c 20 51		INC REGDS
2708 d351 7a 20 52	SOUND4	DEC REGES
2709 d354 26 f5		BNE SOUND3
2710 d356 39		RTS
2711		
2712		***** PRINT SUB *****
2713	*	IN = B DELAY *
2714	*	= REGC DIGIT COUNT *
2715	*	= IY START TABLE *
2716		
2717 d357 37	PRINT	PSHB
2718 d358 bd d3 d2		JSR CLEAR
2719 d35b 18 a6 00	PRINT1	LDAA 0,Y
2720 d35e a7 00		STAA 0,X
2721 d360 7a 20 4e		DEC REGC
2722 d363 27 05		BEQ PRINT11
2723 d365 08		INX
2724 d366 18 08		INY
2725 d368 20 f1		BRA PRINT1
2726 d36a 33	PRINT11	PULB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2727 d36b bd d3 6f          JSR  SCAND
2728 d36e 39                RTS
2729
2730          ***** SCAND SUB. *****
2731          *          IN = B DELAY          *
2732          *          = (DISPY)          *
2733
2734 d36f 36                SCAND          PSHA
2735 d370 86 08            LDAA  #$08
2736 d372 b7 20 4f        STAA  REGD
2737 d375 86 00            LDAA  #$00
2738 d377 b7 20 50        STAA  REGE
2739 d37a 32              PULA
2740 d37b ce 20 9e        LDX   #DISPY
2741 d37e bd d1 40        SCAND1      JSR  SCANS
2742 d381 7c 20 50        INC  REGE
2743 d384 7a 20 4f        DEC  REGD
2744 d387 26 f5          BNE  SCAND1
2745 d389 5a              DECB
2746 d38a c1 00          CMPB  #$00
2747 d38c 26 e1          BNE  SCAND
2748 d38e 39                RTS
2749
2750          ***** SCANK SUB. *****
2751          *          IN = (DISPY)          *
2752          *          OUT = (KEYIN) $FF IS CLEAR      *
2753
2754 d38f 86 08            SCANK          LDAA  #$08
2755 d391 b7 20 4f        STAA  REGD
2756 d394 86 00            LDAA  #$00
2757 d396 b7 20 50        STAA  REGE
2758 d399 ce 20 9e        LDX   #DISPY
2759 d39c 86 ff          LDAA  #$FF

```

2760 d39e b7 20 b2		STAA KEYIN
2761		
2762 d3a1 bd d1 40	SCANK1	JSR SCANS
2763 d3a4 b6 17 02		LDAA DIGIT
2764 d3a7 84 70		ANDA # \$70
2765 d3a9 81 70		CMPA # \$70
2766 d3ab 27 1c		BEQ SCANK2
2767		
2768 d3ad ba 20 50		ORAA REGE
2769 d3b0 37		PSHB
2770 d3b1 18 3c		PSHY
2771 d3b3 3c		PSHX
2772 d3b4 ce d5 94		LDX #KEYTAB+22
2773 d3b7 c6 16		LDAB # \$16
2774 d3b9 a1 00	SCANK11	CMPA 0,X
2775 d3bb 27 04		BEQ SCANK12
2776 d3bd 09		DEX
2777 d3be 5a		DECB
2778 d3bf 26 f8		BNE SCANK11
2779 d3c1 17	SCANK12	TBA
2780 d3c2 38		PULX
2781 d3c3 18 38		PULY
2782 d3c5 33		PULB
2783 d3c6 b7 20 b2		STAA KEYIN
2784		
2785 d3c9 7c 20 50	SCANK2	INC REGE
2786 d3cc 7a 20 4f		DEC REGD
2787 d3cf 26 d0		BNE SCANK1
2788 d3d1 39		RTS
2789		

```

2790 ***** CLEAR SUB.*****
2791 * CLEAR DISPY & BUFFER *
2792
2793 d3d2 4f CLEAR CLRA
2794 d3d3 ce 20 a9 LDX #BUF+3
2795 d3d6 c6 0b LDAB #11
2796 d3d8 09 CLEAR1 DEX
2797 d3d9 a7 00 STAA 0,X
2798 d3db 5a DECB
2799 d3dc 26 fa. BNE CLEAR1
2800 d3de 39 RTS
2801
2802 ***** DELAY SUB.*****
2803 * IN = B DELAY *
2804
2805 d3df 4f DELAY CLRA
2806 d3e0 4a DELAY1 DECA
2807 d3e1 26 fd BNE DELAY1
2808 d3e3 5a DECB
2809 d3e4 26 f9 BNE DELAY
2810 d3e6 39 RTS
2811
2812 ***** CHKS SUB.*****
2813 * IN = IX START *
2814 * = IY LENGTH *
2815 * OUT = A,REGD *
2816
2817 d3e7 86 00 CHKS LDAA #00
2818 d3e9 b7 20 4f STAA REGD
2819 d3ec b6 20 4f CHKS1 LDAA REGD
2820 d3ef a8 00 EORA 0,X
2821 d3f1 b7 20 4f STAA REGD
2822 d3f4 08 INX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2823 d3f5 18 09          DEY
2824 d3f7 26 f3          BNE  CHKS1
2825 d3f9 73 20 4f       COM  REGD
2826 d3fc 39              RTS
2827
2828 d3fd ce 23 00        CHKSR  LDX  #RAM
2829 d400 18 ce 90 00     LDY  #END_PRAM - #RAM
2830 d404 bd d3 e7        JSR  CHKS
2831 d407 39              RTS
2832
2833          ***** ERRST SUB. *****
2834          *          DISPLAY ERROR AND STOP          *
2835
2836 d408 bd d4 2c        ERRST  JSR  LBEEP
2837 d40b c6 05          LDAB  #$05
2838 d40d f7 20 4e        STAB  REGC
2839 d410 c6 01          LDAB  #$01
2840 d412 bd d4 24        JSR  ERR1
2841 d415 bd d0 50        ERRST1 JSR  SCAN
2842 d418 20 fb          BRA  ERRST1
2843
2844          ***** ERROR DISPLAY *****
2845
2846 d41a bd d4 2c        ERR    JSR  LBEEP
2847 d41d c6 05          LDAB  #$05
2848 d41f f7 20 4e        STAB  REGC
2849 d422 c6 30          LDAB  #$30
2850 d424 18 ce d5 29     ERR1  LDY  #ERRTAB
2851 d428 bd d3 57        JSR  PRINT
2852 d42b 39              RTS
2853

```

```

2854                ***** L & H BEEP SUB. *****
2855
2856 d42c 86 50          LBEEP          LDAA #LBPFRE
2857 d42e b7 20 4e          STAA REGC
2858 d431 20 07          BRA BEEPS
2859
2860 d433 86 08          HBEEP          LDAA #HBPFRE
2861 d435 b7 20 4e          STAA REGC
2862 d438 20 00          BRA BEEPS
2863
2864 d43a ce 20 b4          BEEPS          LDX #SYSFAG
2865 d43d a6 00          LDAA 0,X
2866 d43f 85 08          BITA ##08
2867 d441 27 01          BEQ BEEPS1
2868 d443 39          RTS
2869 d444 ce 15 00          BEEPS1        LDX ##1500
2870 d447 b6 20 b6          LDAA GAIN
2871 d44a 16          TAB
2872 d44b bd d3 2d          JSR SOUND
2873 d44e 39          RTS
2874
2875                ***** TABLE SUB. *****
2876                *          LOOK-UP 2 BYTE TABLE          *
2877                *          IN = B NO.          *
2878                *          = IX START TABLE          *
2879                *          OUT = IX DATA FROM TABLE          *
2880
2881 d44f 58          TABLE          LSLB
2882 d450 3a          ABX
2883 d451 a6 00          LDAA 0,X
2884 d453 08          INX
2885 d454 e6 00          LDAB 0,X
2886 d456 8f          XGDY

```

```

2887 d457 39                RTS
2888
2889                ***** TITLE SONG SUB. *****
2890                *                PLAY TITLE SONG                *
2891
2892 d458 18 ce d5 16        TITLE                LDY  #SNGTAB
2893 d45c 86 04                LDAA  #04
2894 d45e b7 20 4f                STAA  REGD
2895
2896                ***** SONG SUB. *****
2897                *  IN = REGD NOTE COUNT                *
2898                *  = (IX) NOTE TABLE (2BYTE/NOTE) *
2899
2900 d461 c6 00                SONGS                LDAB  #00
2901 d463 18 a6 00                LDAA  0,Y
2902 d466 8f                XGDX
2903 d467 c6 ff                LDAB  #$FF
2904 d469 18 08                INY
2905 d46b 18 a6 00                LDAA  0,Y
2906 d46e b7 20 4e                STAA  REGC
2907 d471 bd d3 2d                JSR  SOUND
2908 d474 18 08                INY
2909 d476 7a 20 4f                DEC  REGD
2910 d479 26 e6                BNE  SONGS
2911 d47b 39                RTS
2912
2913                ** ADDRESS RECALL SUBROUTINE TABLE **
2914
2915 d47c c2 aa d3 2d        CALL_TAB                FDB  MON,SOUND
2916 d480 d0 50 d3 d2                FDB  SCAN,CLEAR
2917 d484 d3 6f d3 8f                FDB  SCAND,SCANK
2918 d488 d1 54 d1 b6                FDB  PACK,UNPK
2919 d48c d2 9e d2 7b                FDB  IN2,IN4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2920	d490 d3 57 d3 05	FDB	PRINT,READY
2921	d494 d4 33 d3 df	FDB	HBEEP,DELAY
2922	d498 c2 c8 cd 75	FDB	MON1,POFF
2923	d49c d4 08 d4 61	FDB	ERRST,SONGS
2924	d4a0 d4 58 d4 4f	FDB	TITLE,TABLE
2925	d4a4 d2 e2 d2 f9	FDB	INSF,IND
2926	d4a8 d3 1b d3 24	FDB	SOUNDI,SOUND
2927	d4ac d4 2c d3 e7	FDB	LBEEP,CHKS
2928	d4b0 d0 47 d0 41	FDB	DELAYL,DELAY_L
2929	d4b4 cf 8b cf ac	FDB	TX_SUB,M_ASCII
2930	d4b8 cf d5 d0 15	FDB	INIT_LCD,CURSOR_L
2931	d4bc d0 0a d0 37	FDB	CLEAR_L,READY_L
2932	d4c0 cf f8 cf bd	FDB	WRP,ROW_LCD
2933	d4c4 cf c4 d0 20	FDB	ROW_LCDL,OUT_L
2934	d4c8 d0 27 ce 44	FDB	OUT_LINE,SEND
2935	d4cc ce 58 cf 0b	FDB	BUSY,ONFK
2936	d4d0 cf 5b cf 70	FDB	FIND_GL,FIND_GB
2937	d4d4 ce 60	FDB	CHKRAM
2938			
2939		***** ADDRESS FUNCTION KEY TABLE *****	
2940			
2941	d4d6 c6 d4 cd 75	FK1TAB	FDB BEEPT,POFF
2942	d4da c6 fd c7 26		FDB USER,SET
2943	d4de ce 08 c9 93		FDB CRAM,CREG
2944	d4e2 cd a0 c6 75		FDB IN,HELP
2945	d4e6 cb ca cb e8		FDB OUT,GNSTEP
2946	d4ea cb b9 cb f7		FDB BREAK,CBREAK
2947	d4ec cd 15 cb 92		FDB RELA,FILL
2948	d4f2 cc 86 cb 3d		FDB SD,FIND
2949	d4f6 cd 93 c9 43	FK2TAB	FDB C_LCD,TEST_LCD
2950	d4fa c8 be c8 12		FDB VOLTS_METER,ADC_8CH
2951	d4fe c8 95 c7 c8		FDB TX_RX,TIMER1
2952	d502 c9 19 cc 11		FDB TIMER2,TRAN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2953	d506 c7 72 cc c5	FDB	REV, TES_SEG
2954	d50a cc 4e c9 bf	FDB	TES_PRN, TESTR
2955	d50e cb 13 ca db	FDB	MOVE, COMP
2956	d512 ca 99 c9 dc	FDB	COMPH, CHAN
2957			
2958		***** SOUND TABLE *****	
2959			
2960	d516 30 6d 30 57	SNGTAB	FCB \$30, \$6D, \$30, \$57
2961	d51a 30 49 30 36		FCB \$30, \$49, \$30, \$36
2962			
2963		***** SEVEN SEGMENT TABLE *****	
2964			
2965	d51e 06 de 79 de 86 3f	HELTAB	FCB \$06, \$DE, \$79, \$DE, \$86, \$3F
2966	d524 71 1c 54 d8 06	FUNTAB	FCB \$71, \$1C, \$54, \$D8, \$06
2967	d529 79 50 50 5c 50	ERRTAB	FCB \$79, \$50, \$50, \$5C, \$50
2968	d52e 3f 71 71	OFFTAB	FCB \$3F, \$71, \$71
2969	d531 3f 54 00	ONTAB	FCB \$3F, \$54, \$00
2970	d534 58 38 79 77 50	CRTAB	FCB \$58, \$38, \$79, \$77, \$50
2971	d539 3e 6d 79 50	USRTAB	FCB \$3E, \$6D, \$79, \$50
2972	d53d 6d 79 78 40	SETTAB	FCB \$6D, \$79, \$78, \$40
2973	d541 50 79 77 5e 6e	RDYTAB	FCB \$50, \$79, \$77, \$5E, \$6E
2974	d546 38 77 78 39 76	LATTAB	FCB \$38, \$77, \$78, \$39, \$76
2975	d54b 71 50 5c 54	FRMTAB	FCB \$71, \$50, \$5C, \$54
2976	d54f 78 5c	TOTAB	FCB \$78, \$5C
2977	d551 6e 79 64	YESTAB	FCB \$6E, \$79, \$64
2978	d554 54 3f	NOTAB	FCB \$54, \$3F
2979	d556 6d 79 38 79 58 78	SEL_TAB	FCB \$6D, \$79, \$38, \$79, \$58, \$78
2980	d55c 76 79 38 73	SHELP_TAB	FCB \$76, \$79, \$38, \$73
2981			
2982	d560 73 39 06 6e	REGTAB	FCB \$73, \$39, \$06, \$6E
2983	d564 06 70 77 7c		FCB \$06, \$70, \$77, \$7C
2984	d568 39 50 6d 73		FCB \$39, \$50, \$6D, \$73
2985			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2986 d56c 01 08 02 80          SDTAB  FCB $01,$08,$02,$80
2987 d570 04 10 05 40          FCB $04,$10,$05,$40
2988 d574 06 04 08 20          FCB $06,$04,$08,$20
2989 d578 09 40 0a 02          FCB $09,$40,$0A,$02
2990 d57c 0d 01                FCB $0D,$01
2991
2992 d57e 60 61 62 63          KEYTAB FCB $60,$61,$62,$63
2993 d582 50 51 52 53          FCB $50,$51,$52,$53
2994 d586 30 31 32 33          FCB $30,$31,$32,$33
2995 d58a 65 55 35 66          FCB $65,$55,$35,$66
2996 d58e 64 67 54 57          FCB $64,$67,$54,$57
2997 d592 34 36 56            FCB $34,$36,$56
2998
2999 d595 3f 06 5b 4f          SEGTAB FCB $3F,$06,$5B,$4F
3000 d599 66 6d 7d 07          FCB $66,$6D,$7D,$07
3001 d59d 7f 6f 77 7c          FCB $7F,$6F,$77,$7C
3002 d5a1 39 5e 79 71          FCB $39,$5E,$79,$71
3003
3004          ***** ASCII LCD TABLE *****
3005
3006 d5a5 20 20 43 79 63 6c    CYC_TAB  FCC   ' Cycles '
      65 73 20 20
3007 d5af 2a 54 45 53 54 20    TLCD_TAB FCC   '*TEST LCD 16CHR*'
      4c 43 44 20 31 36
      43 48 52 2a
3008 d5bf 3c 3c 41 4e 30 2d    ADC8_TAB  FCC   '<<ANO-AN7 DEMO>>'
      41 4e 37 20 44 45
      4d 4f 3e 3e
3009 d5cf 3c 3c 56 4f 4c 54    VOLT_TAB  FCC   '<<VOLTS METER>>'
      53 20 20 4d 45 54
      45 52 3e 3e
3010 d5df 3c 3c 3c 20 54 52    TX_TAB    FCC   '<<< TRANSMIT >>>'
      41 4e 53 4d 49 54

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20 3e 3e 3e
 3011 d5ef 3c 3c 3c 20 20 52 RX_TAB FCC '<<< RECEIVE >>>'
 45 43 45 49 56 45
 20 3e 3e 3e
 3012 d5ff 3c 3c 3c 20 54 58 TXRX_TAB FCC '<<< TX & RX >>>'
 20 20 26 20 52 58
 20 3e 3e 3e
 3013 d60f 3c 4d 55 53 55 52 TIME_TAB FCC '<MUSUREING TIME>'
 45 49 4e 47 20 54
 49 4d 45 3e
 3014 d61f 3c 20 53 51 55 41 SQW_TAB FCC '< SQUARE WAVE >'
 52 45 20 20 57 41
 56 45 20 3e
 3015 d62f 2a 20 54 45 53 54 TSEG_TAB FCC '* TEST SEGMENT *'
 20 53 45 47 4d 45
 4e 54 20 2a
 3016 d63f 3c 20 54 45 53 54 PRN_TAB FCC '< TEST PRINTER >'
 20 50 52 49 4e 54
 45 52 20 3e
 3017 d64f 5b 49 4e 43 3a 52 SEL_TAB1 FCC '[INC:R][DEC:L]'
 20 5d 5b 20 44 45
 43 3a 4c 5d
 3018 d65f 41 55 54 4f 52 55 BOARD_TAB FCC 'AUTORUN ON BOARD'
 4e 20 4f 4e 20 42
 4f 41 52 44
 3019 d66f 20 41 55 54 4f 52 PC_TAB FCC 'AUTORUN ON PC. '
 55 4e 20 4f 4e 20
 50 43 2e 20
 3020 d67f 20 41 55 54 4f 52 RAM_TAB FCC 'AUTORUN ON RAM '
 55 4e 20 4f 4e 20
 52 41 4d 20
 3021 d68f 4d-55 4c 54 49 20 MULTI_TAB FCC 'MULTI BOARD V1.0'
 42 4f 41 52 44 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

56 31 2e 30

3022

3023 d69f 48 45 4c 50 20 4d HELP_TAB FCC 'HELP MENU BOARD'
 45 4e 55 20 20 42
 4f 41 52 44

3024 d6af 41 44 44 52 45 53 ADDR_TAB FCC 'ADDRESS SYSTEM'
 53 20 20 53 59 53
 54 45 4d 20

3025 d6bf 43 4f 4e 54 49 4e CON_TAB FCC 'CONTINUE [Y/N]'
 55 45 20 20 5b 59
 2f 4e 5d 20

3026 d6cf 50 57 43 4f 44 45 PW_TAB FCC 'PWCODE : [20B5]'
 20 3a 20 5b 32 30
 42 35 5d 20

3027 d6df 52 45 43 41 4c 4c RST_TAB FCC 'RECALL : [CE2B]'
 20 3a 20 5b 43 45
 32 42 5d 20

3028 d6ef 42 4f 4f 54 4d 45 BOOT_TAB FCC 'BOOTMEM : [206A]'
 4d 20 3a 20 5b 32
 30 36 41 5d

3029 d6ff 50 41 53 53 44 49 PASS_TAB FCC 'PASSDIG : [2055]'
 47 20 3a 20 5b 32
 30 35 35 5d

3030 d70f 47 41 49 4e 20 20 GAIN_TAB FCC 'GAIN : [20B6]'
 3a 20 5b 32 30 42
 36 5d 20 20

3031 d71f 42 41 43 4b 55 50 BACK_TAB FCC 'BACKUP : [2058]'
 20 3a 20 5b 32 30
 35 38 5d 20

3032 d72f 53 59 53 46 4c 41 SYS_TAB FCC 'SYSFLAG : [20B4]'
 47 20 3a 20 5b 32
 30 42 34 5d

3033 d73f 53 54 2e 48 45 58 HBUF_TAB FCC 'ST.HEXBUF:[20A6]'

42 55 46 3a 5b 32

30 41 36 5d

3034 d74f 53 54 2e 44 49 53 DISPY_TAB FCC 'ST.DISPY :[209E]'

50 59 20 3a 5b 32

30 39 45 5d

3035 d75f 4d 4f 4e 49 54 4f MON_TAB FCC 'MONITOR : [C2AA]'

52 20 3a 20 5b 43

32 41 41 5d

3036

3037

***** ASCII ON MONITOR PC TAB *****

3038

3039 d76f 20 6d 69 6c 6c 69 MV_TAB FCC 'millivolts'

56 6f 6c 74 73

3040 d77a 04

FCB \$04

3041 d77b 0d 04

COL1_TAB FCB \$0D,\$04

3042 d77d 20 7c

MSGO FCC ' ' .

3043 d77f 04

FCB \$04

3044 d780 6d 56 7c

MSG FCC 'mV'

3045 d783 04

FCB \$04

3046

3047 d784 23 23 23 23 23 23 23 ADC8CH_TAB FCC '#####'

23 23 23 23 23 23

23 23 23 23 23 23

23

3048 d797 23 23 23 23 23 23

FCC '#####'

23 23 23 23 23 23

23 23 23 23 23 23

23

3049 d7aa 23 23 23 23 23 23

FCC '#####'

23 23 23 23 23 23

23 23 23 23 23 23

23

3050 d7bd 23 23 23 23 23 23

FCC '#####'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติเห็นาไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

23 23 23 23 23 23
 23 23 23.23 23
 3051 d7ce 0a 0d FCB \$0A,\$0D
 3052 d7d0 23 20 20 20 20 20 FCC '# << M'
 20 20 20 20 20 20
 20 20 20 3c 3c 20
 4d
 3053 d7e3 55 4c 54 49 20 20 FCC 'ULTI BOARD MICROC'
 42 4f 41 52 44 20
 20 4d 49 43 52 4f
 43
 3054 d7f6 4f 4e 54 52 4f 4c FCC 'ONTROLLER 68HC11 >'
 4c 45 52 20 20 36
 38 48 43 31 31 20
 3e
 3055 d809 3e 20 20 20 20 20 FCC '> #'
 20 20 20 20 20 20
 20 20 20 20 23
 3056 d81a 0a 0d FCB \$0A,\$0D
 3057 d81c 23 20 20 20 20 20 FCC '# D'
 20 20 20 20 20 20
 20 20 20 20 20 20
 44
 3058 d82f 45 4d 4f 53 54 41 FCC 'EMOSTATION VOLTS M'
 54 49 4f 4e 20 20
 56 4f 4c 54 53 20
 4d
 3059 d842 45 54 45 52 20 38 FCC 'ETER 8 CHANNEL '
 20 43 48 41 4e 4e
 45 4c 20 20 20 20
 20
 3060 d855 20 20 20 20 20 20 FCC ' #'
 20 20 20 20 20 20

20 20 20 20 23

3061 d866 0a 0d FCB \$0A,\$0D

3062 d868 23 23 23 23 23 23 FCC '#####'

23 23 23 23 23 23

23 23 23 23 23 23

23

3063 d87b 23 23 23 23 23 23 FCC '#####'

23 23 23 23 23 23

23 23 23 23 23 23

23

3064 d88e 23 23 23 23 23 23 FCC '#####'

23 23 23 23 23 23

23 23 23 23 23 23

23

3065 d8a1 23 23 23 23 23 23 FCC '#####'

23 23 23 23 23 23

23 23 23 23 23

3066 d8b2 0a 0d 0a 0d FCB \$0A,\$0D,\$0A,\$0D

3067 d8b6 20 20 5b 20 43 48 FCC ' [CH1] [CH2] '

31 20 5d 20 20 5b

20 43 48 32 20 5d

20

3068 d8c9 20 5b 20 43 48 33 FCC ' [CH3] [CH4] '

20 5d 20 20 5b 20

43 48 34 20 5d 20

20

3069 d8dc 5b 20 43 48 35 20 FCC ' [CH5] [CH6] ['

5d 20 20 5b 20 43

48 36 20 5d 20 20

5b

3070 d8ef 20 43 48 37 20 5d FCC ' CH7] [CH8] '

20 20 5b 20 43 48

38 20 5d 20 20

3071 d900 0a 0d 04

FCC \$0A,\$0D,\$04

3072

3073 d903 25 25 25 25 25 25 TRANSMIT_TAB FCC '%%'

25 25 25 25 25 25

25 25 25 25 25 25

25 25

3074 d917 25 25 25 25 25 25

FCC '%%'

25 25 25 25 25 25

25 25 25 25 25 25

25 25

3075 d92b 25 25 25 25 25 25

FCC '%%'

25 25 25 25 25 25

25 25 25 25 25 25

3076 d93d 0a 0d-

FCC \$0A,\$0D

3077 d93f 25 20 20 20 20 20

FCC '% << MULTI BOA'

20 3c 3c 20 4d 55

4c 54 49 20 20 42

4f 41

3078 d953 52 44 20 20 4d 49

FCC 'RD MICROCONTROLLER '

43 52 4f 43 4f 4e

54 52 4f 4c 4c 45

52 20

3079 d967 20 36 38 48 43 31

FCC ' 68HC11 >> %'

31 20 3e 3e 20 20

20 20 20 20 20 25

3080 d979 0a 0d

FCC \$0A,\$0D

3081 d97b 25 20 20 20 20 20

FCC '% DEMOS'

20 20 20 20 20 20

20 20 20 44 45 4d

4f 53

3082 d98f 54 41 54 49 4f 4e

FCC 'TATION TRANSMIT FUNC'

20 54 52 41 4e 53

4d 49 54 20 46 55

4e 43

3083 d9a3 54 49 4f 4e 20 20 FCC 'TION %'

20 20 20 20 20 20

20 20 20 20 20 25

3084 d9b5 0a 0d FCB \$0A,\$0D

3085 d9b7 25 25 25 25 25 FCC '%%%%%%%%%%'

. 25 25 25 25 25 25

25 25 25 25 25 25

25 25

3086 d9cb 25 25 25 25 25 25 FCC '%%%%%%%%%%'

25 25 25 25 25 25

25 25 25 25 25 25

25 25

3087 d9df 25 25 25 25 25 25 FCC '%%%%%%%%%%'

25 25 25 25 25 25

25 25 25 25 25 25

3088 d9f1 0a 0d 04 FCB \$0A,\$0D,\$04

3089

3090 dffe ORG \$DFFE

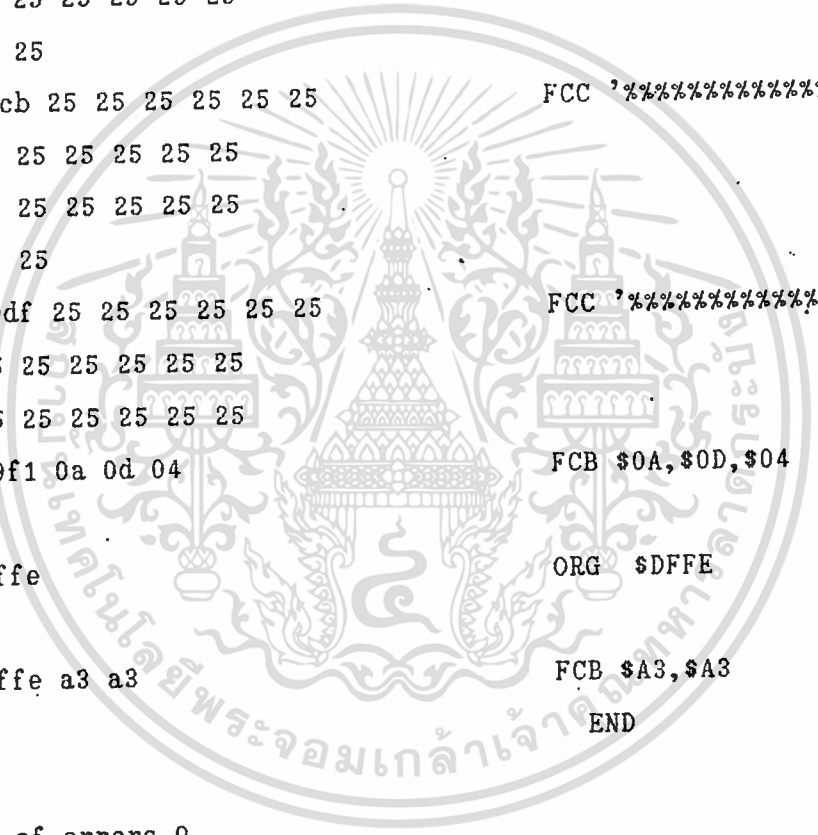
3091

3092 dffe a3 a3 FCB \$A3,\$A3

3093 END

3094

Number of errors 0





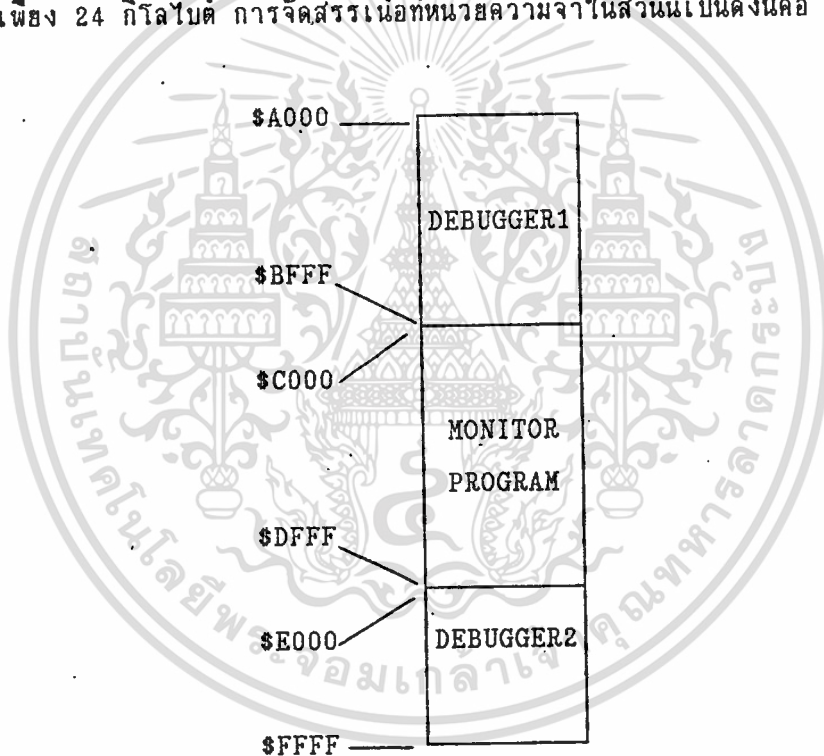
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดสรรเนื้อที่หน่วยความจำของมัลติบอร์ด

68HC11 สามารถอ้างหน่วยความจำและตำแหน่งของพอร์ตรวมกันได้ 64 กิโลไบต์ ในส่วนของพอร์ต 68HC11 นั้นจะมองเสมือนว่าเป็นตำแหน่งในหน่วยความจำ การจัดเนื้อที่ของหน่วยความจำในมัลติบอร์ดสามารถแบ่งเป็น 2 ส่วนคือ หน่วยความจำสำหรับโปรแกรม (Program memory) และหน่วยความจำสำหรับข้อมูล (Data memory)

หน่วยความจำสำหรับโปรแกรม

ในส่วนนี้ข้อมูลจะถูกเก็บอยู่ในอีพ롬ซึ่งจะเป็นส่วนโปรแกรมมอเนเตอร์ของมัลติบอร์ด การใช้งานจะอ่านได้อย่างเดียวอีพ롬ที่ใช้คือเบอร์ 27256 ขนาด 32 กิโลไบต์แต่ใช้งานจริงได้เพียง 24 กิโลไบต์ การจัดสรรเนื้อที่หน่วยความจำในส่วนนี้เป็นดังนี้คือ

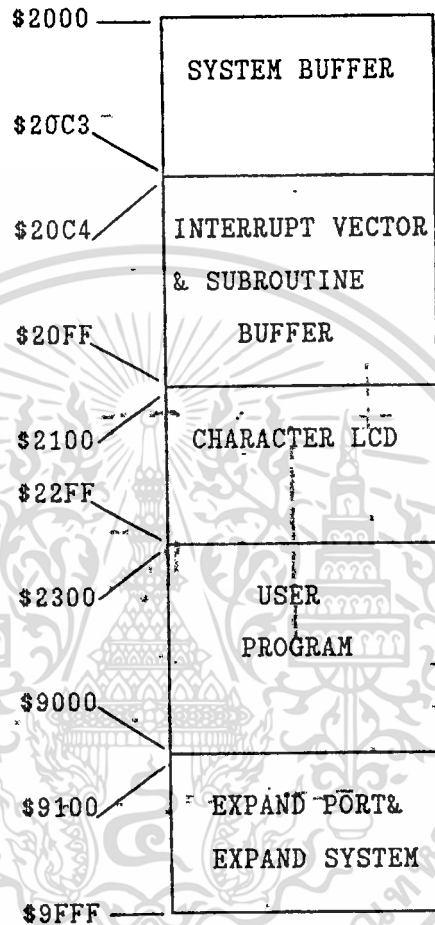


จากรูปจะเห็นว่า DEBUGGER ที่นำมาใช้สำหรับการทำงานบนเครื่องคอมพิวเตอร์นั้น จะมีพื้นที่ของโปรแกรมน้อยอยู่ 2 ช่วง คือ ช่วงตำแหน่งหน่วยความจำ \$A000-\$BFFF และ ช่วงตำแหน่ง \$E000-\$FFFF สำหรับโปรแกรมมอเนเตอร์ของมัลติบอร์ดในส่วนการทำงานบนมัลติบอร์ดนั้นจะโปรแกรมไว้ในหน่วยความจำช่วง \$C000-\$DFFF

หน่วยความจำสำหรับข้อมูล

หน่วยความจำในส่วนนี้จะใช้สำหรับการพักข้อมูลหรือเก็บข้อมูลชั่วคราว สามารถอ่านหรือเขียนได้ตามต้องการโดยจะใช้แรมมาทำหน้าที่ในส่วนนี้ แรมที่ใช้คือเบอร์ 62256 ซึ่งมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาดความจุ 32 กิโลไบต์ ตำแหน่งของแรมคือช่วงหน่วยความจำ \$2000-\$9FFF การจัด
เนื้อที่การใช้งานแรม ในช่วงดังกล่าวเป็นดังนี้คือ



หน่วยความจำในช่วงตำแหน่ง \$2000-\$20C3 ถูกจัดสรรไว้เป็น SYSTEM BUFFER นั่นคือจะใช้เป็นพื้นที่สแต็กและเก็บข้อมูลที่เกิดจากการจองเนื้อที่ของระบบในการทำงานของ โปรแกรมมอไนเตอร์ ซึ่งรายละเอียดของส่วนนี้จะดูได้ในส่วน LIST FILES ในภาคผนวก หน่วยความจำช่วงตำแหน่ง \$20C4-\$20FF จองเนื้อที่ไว้สำหรับเก็บอินเตอร์รัพต์เวคเตอร์ ของการอินเตอร์รัพต์แบบต่างๆ ที่มาจาก 21 แหล่ง ซึ่งหน่วยความจำในช่วงตำแหน่งตั้งแต่ \$2100-\$22FF ใช้พักข้อมูลที่แสดงผลบนจอแอลซีดีและการทำงานในโปรแกรมย่อย เช่น แปลงเลขฐานสิบหกเป็นฐานสิบเป็นต้น หน่วยความจำช่วงตำแหน่ง \$2300-\$9000 จะเป็น ส่วนสำหรับเก็บโปรแกรมที่ผู้ใช้ป้อนให้มัลติบอร์ดประมวลผลที่อยู่ในรูปของ machine code ส่วนหน่วยความจำตำแหน่งที่เหลือคือช่วง \$9001-\$9FFF จะเว้นไว้ในกรณีที่เราต้องการ ขยายพอร์ตภายนอกไม่ให้อัปเดตกับโปรแกรมที่ประมวลผลและในการสร้างระบบเพิ่มเติมในส่วน ของโปรแกรมมอไนเตอร์

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากที่ได้กล่าวถึงการจัดสรรพื้นที่หน่วยความจำของมัลติบอร์ดนั้นจะเป็นการจัดสรรโดยโปรแกรมมอนิเตอร์ที่เขียนขึ้นสำหรับการทำงานบนมัลติบอร์ด แต่กรณีที่เลือกการทำงานบนเครื่องคอมพิวเตอร์ใน DEBUGGER นั้นจะมีการจัดสรรหน่วยความจำส่วนที่แตกต่างไปจากที่กล่าวมา เนื่องจากไม่มี SOURCE FILES ของ DEBUGGER จึงไม่สามารถอธิบายการจัดสรรหน่วยความจำสำหรับข้อมูลเมื่อทำงานใน DEBUGGER ได้

นอกจากนี้แล้วก็ยังมีการจัดหน่วยความจำโดย 68HC11 เมื่อจัดโหมดการทำงานเป็นโหมดมัลติเพล็กซ์ชยาซึ่งได้กล่าวถึงไปแล้วในการออกแบบวงจรในส่วนของคุณสมบัติจึงจะไม่กล่าวซ้ำในที่นี้



บรรณานุกรม

- บริษัท ซีเอ็ดดูเคชั่นจำกัด. เซมิคอนดักเตอร์อิเล็กทรอนิกส์ฉบับที่ 127 พฤศจิกายน 2536,
ฉบับที่ 128 มิถุนายน 2536, ฉบับที่ 133 พฤศจิกายน 2536,
ฉบับที่ 135 มกราคม 2537, ฉบับที่ 136 กุมภาพันธ์ 2537,
ฉบับที่ 137 มีนาคม 2537, ฉบับที่ 138 เมษายน 2537,
ฉบับที่ 140 มิถุนายน 2537.--กรุงเทพมหานคร: OS PRINTING HOUSE.
- วิบูลย์ ชื่นแขก. ไมโครโปรเซสเซอร์. กรุงเทพมหานคร:
โรงพิมพ์สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, 2531.
- บริษัท อีทีทีจำกัด. ET-BORAD V.4.0 NEW GENERATION USER'S MANUAL.
กรุงเทพมหานคร: โรงพิมพ์พิษณุโลก, 2535.
- บริษัท อีทีทีจำกัด. ET-DEGGER 68HC11. กรุงเทพมหานคร: โรงพิมพ์พิษณุโลก,
2536.
- บริษัท อีทีทีจำกัด. Z80 ET-BORAD SINGLE BORAD MICROCOMPUTER.
กรุงเทพมหานคร: โรงพิมพ์พิษณุโลก, 2532.
- Joseph D. Greenfield. "The 68HC11 Microcontroller":
Saunders College Publishing, 1991.
- Motorola Inc. "M68HC11EVBU Universal Evaluation Board -
User's Manual": Motorola Inc., 1992.
- Ramakant Gayakwad, Lednard Sokoloff. "Analog And Digital -
Control Systems": Prentice-Hall Inc., 1988.