



ปริญญาานิพนธ์

## ชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180

Z80180 MICROPROCESSOR DEVELOPMENT AND TRAINING SET



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต

สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตรบัณฑิต

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาครุศาสตร์วิศวกรรม  
คณะครุศาสตร์อุตสาหกรรม  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาโท

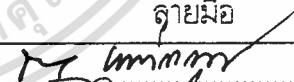


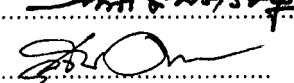

ชื่อหัวข้อปริญญาโท ชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180

Z80180 MICROPROCESSOR DEVELOPMENT AND TRAINING SET

ชื่อนักศึกษา	ชื่อ	รหัสนักศึกษา	เลขประจำตัว
1.	นายธราดล ไกรวิเศษฐ์	รหัสนักศึกษา	36031308
2.	นายนรินทร์ เลิศประเสริฐ	รหัสนักศึกษา	36031310
3.	นายพงศ์พันธุ์ ยรรยง	รหัสนักศึกษา	36031318
4.	นายเลิศพงศ์ ทองวินิจ	รหัสนักศึกษา	36031324

อาจารย์ผู้ควบคุมปริญญาโท

1. ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา
2. อ.กิติพงศ์ มะโน
3. อ.วรวิทย์ สมทา

คณะกรรมการสอบปริญญาโท	ลายมือ
1. ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา	
2. อ.กิติพงศ์ มะโน	
3. อ.วรวิทย์ สมทา	
4. อ.สันติ ตันตระกูล	
5. อ.สุชิน อาจหาญ	

วันเดือนปี ที่สอบ 25 ธันวาคม 2537 เวลา 10.00 น. ถึง 11.00

สถานที่สอบ ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม



(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

# ปริญญานิพนธ์

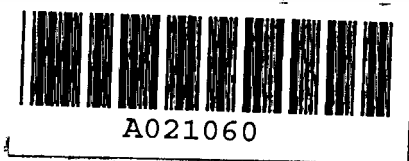
เรื่อง ชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180  
Z80180 MICROPROCESSOR DEVELOPMENT AND TRAINING SET

## จุดประสงค์

1. เพื่อใช้ประกอบการศึกษาเกี่ยวกับไมโครโปรเซสเซอร์ Z80180.
2. เพื่อใช้ประกอบการทดลองปฏิบัติการที่เกี่ยวกับ Z80180
3. เพื่อใช้ศึกษาเกี่ยวกับการเชื่อมต่อ Z80180 กับชิพสนับสนุนแบบต่างๆ

## ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจการทำงานของอุปกรณ์สนับสนุนต่างๆที่ใช้ในปริญญานิพนธ์นี้
2. เข้าใจรูปแบบการเขียนโปรแกรม ASSEMBLE เพื่อแปลคำสั่งให้กับ Z80180
3. ได้เครื่องต้นแบบชุดฝึก Z80180 ไว้ใช้ในการเรียนการสอน



เลขหมู่	.....
เลขทะเบียน	1292
วัน เดือน ปี	-3 WEI 2538

021060

# ชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180

นายธราดล ไกรวิเชษฐ์  
นายนรินทร์ เลิศประเสริฐ  
นายพงศ์พันธ์ ยรรยง  
นายเลิศพงศ์ ทองวินิจ

อาจารย์ที่ปรึกษา

ผศ.ดร. ชีระพล เทพหัสดิน ณ อยุธยา

อ.กิตติพงศ์ มะโน

อ.วรวิทย์ สมทา

ปีการศึกษา 2537

## บทคัดย่อ

ชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180 ได้ออกแบบขึ้นเพื่อใช้ในการศึกษาและทดลองการใช้ CPU Z80180 โดยออกแบบให้มีลักษณะเหมือน SINGLE BOARD ซึ่งผู้ใช้สามารถศึกษา ทำความเข้าใจ และพัฒนาโปรแกรมได้อย่างสะดวกและชัดเจน

โดยชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180 นี้ประกอบด้วย LCD DISPLAY ขนาด 16x1 ตัวอักษร , คีย์บอร์ดขนาด 4x8 คีย์ , ชุดแปลงสัญญาณดิจิทัลเป็นอนาล็อก และอนาล็อกเป็นดิจิทัล , หน่วยความจำขนาด 1 MB , แผงวงจร Z80180 จากระบบที่ออกแบบสามารถนำไปใช้ในการทดลอง และการเรียนได้อย่างเหมาะสม

## Z80180 MICROPROCESSOR DEVELOPMENT AND TRAINING SET

MR.TARADON            KRAIVICHET  
MR.NARIN              LERDPRASERT  
MR.PONGPUN          YANYONG  
MR.LERTPONG         THONGVINITH

ADVISOR

Assis.Prof.Dr. THRERAPHON THEPHASADIN NA AYUTHYA

MR. KITTIPONG MANO

MR. WORAWIT SOMHA

1994

### ABSTRACT

Z80180 DEVELOPMENT AND TRAINING SET IS DESIGNED FOR STUDYING AND EXPERIMENTING . THIS THESIS HAS DESIGNED LIKE A SINGLE BOARD SET THAT USERS CAN STUDY , UNDERSTAND AND DEVELOPE IT CONVINIENTLY AND CLEARLY.

THE CONTENT OF THIS THESIS IS ABOUT 16x1 CHARACTER LCD DISPLAY , 4x8 KEYBOARD , ANALOG TO DIGITAL AND DIGITAL TO ANALOG CONVERSION , MEMORY 1 MB , Z80180 BOARD . Z80180 DEVELOPMENT AND TRAINING SET IS COMPLETE AND SUITABLE FOR USING IN LABORATORY .

## กิตติกรรมประกาศ

ความสำเร็จคล่องของปริญญาณิพนธ์เล่มนี้ ส่วนหนึ่งเกิดจากท่านอาจารย์ใน  
ภาควิชาครุศาสตร์วิศวกรรมที่กรุณาให้คำแนะนำต่างๆ และอำนวยความสะดวกในการใช้  
เครื่องมือประกอบการสร้างและทดลอง อีกส่วนหนึ่งเกิดจากคุณเกรียงศักดิ์ บุญเสริมสูงวงศ์  
(บริษัท กิลาร์เสิร์ช) ที่ให้คำปรึกษาด้านซอฟต์แวร์ และคุณจักรวรรชัย สุริยจันทร์ (บริษัท อีทีที)  
ที่ให้คำปรึกษาทางด้านเทคนิค ขอขอบคุณ คุณอารยา พิณีจ นศ.ภาควิชาครุศาสตร์สถาปัตยกรรม  
ภายในที่ช่วยทำสติ๊กเกอร์บนแผ่นหน้าปัดของชุดฝึก ขอขอบคุณ คุณเฉลิมพล อภิวงศ์  
คุณพงษ์ทอง คำเลิศ คุณประทีป เตชชัย นศ.ภาควิชาครุศาสตร์สถาปัตยกรรม ที่ช่วยพ่นสี  
แทนเครื่อง ให้ยืมเครื่องมือในการสร้างแทนเครื่องและถ่ายภาพประกอบปริญญาณิพนธ์ และ  
ที่สำคัญที่สุดขอขอบพระคุณ บิดา-มารดา ที่ให้การส่งเสริมด้านการศึกษาเป็นอย่างดีตลอดมา  
คณะผู้จัดทำขอระลึกถึงไว้ ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ไมโครโปรเซสเซอร์ Z80180	2
2.1.1 ขบวนการใช้งาน	2
2.1.2 INTERNAL I/O REGISTOR	6
2.1.3 OPERATION MODE	7
2.1.4 เกี่ยวกับ TIMING	9
2.1.5 WAIT STATE GENERATOR	10
2.1.6 HALT และ LOW POWER MODE	11
2.1.7 MEMORY MANAGEMENT UNIT (MMU)	11
2.1.8 การอินเทอร์รัพท์	17
2.1.9 DYNAMIC RAM REFRESH CONTROL	19
2.1.10 DMA CONTROLLER ( DMAC)	20
2.1.11 CLOCK SERIAL I/O PORT (CSI/O)	29
2.1.12 PROGRAMMABLE RELOAD TIMER (PRT)	31
2.1.13 SECOUNDARY BUS INTERFACE	32
2.1.14 FREE RUNNING COUNTER	32
2.1.15 คำสั่งเพิ่มเติม 12 คำสั่ง	33
2.2 REAL TIME CLOCK	34
2.2.1 การจัดการและหน้าที่ของขาต่างๆ	35
2.2.2 รีจิสเตอร์ต่างๆ	36
2.3 LIQUID CRYSTAL DISPLAY	40
2.3.1 โครงสร้างของ LCD	41
2.3.2 LCD แบบนิวมดิกชนิดเกลียว	41
2.3.3 แบบต่างๆของการแสดงผล	43
2.3.4 สีสันของ LCD	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
2.3.5 คุณสมบัติทางแสง	45
2.3.6 มุมมอง	45
2.3.7 ความเร็วของการแสดงผล	46
2.3.8 การขับ LCD	46
2.4 มาตรฐาน RS 232	46
2.4.1 ลักษณะของสัญญาณ RS 232	48
2.4.2 การกำหนดจุดต่อของ RS 232	49
2.4.3 มาตรฐาน RS 232 กับ V.24	51
บทที่ 3 การออกแบบและการสร้าง	53
3.1 การออกแบบ การสร้างและการทำงานแต่ละส่วน	53
3.1.1 ด้านฮาร์ดแวร์	53
3.1.2 ด้านซอฟต์แวร์	54
3.2 รายละเอียดของบอร์ด JR-180	54
3.2.1 การจัดหน่วยความจำบน CP-JR180	55
3.2.2 การจัดการเกี่ยวกับพอร์ต	55
3.2.3 วงจรเพาเวอร์อนรีเซตและวอตช์ดีออก	56
3.3 รายละเอียดของบอร์ด MEMORY ขนาด 1 MB	57
3.3.1 การใช้พื้นที่หน่วยความจำในการพัฒนาโปรแกรม	61
3.3.2 การออกแบบการถอดรหัส MEMORY (DECODE)	63
3.4 LCD DISPLAY	63
3.5 วงจรแสดงผล ASCII CODE	63
3.6 USER PORT	64
3.7 คีย์บอร์ด	64
3.8 รายละเอียดของบอร์ด ANALOG TO DIGITAL และ DIGITAL TO ANALOG	64
บทที่ 4 ผลการทดลองและทดสอบ	69
4.1 การทดสอบ LED ASCII CODE และ USER PORT	69
4.1.1 ขั้นตอนการทดลอง	69
4.1.2 ผลการทดลอง	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
4.2 การทดสอบ MEMORY BOARD 1 MB	69
4.2.1 ขั้นตอนการทดลอง	69
4.2.2 ผลการทดลอง	70
4.3 การทดสอบจอแสดงผลแบบ LCD	70
4.3.1 ขั้นตอนการทดลอง	71
4.3.2 ผลการทดลอง	71
4.4 การทดสอบคีย์บอร์ด	71
4.4.1 ขั้นตอนการทดลอง	73
4.4.2 ผลการทดลอง	73
บทที่ 5 บทวิจารณ์สรุปและแนวทางพัฒนา	74
ภาคผนวก	76
ภาคผนวก ก โปรแกรมทดสอบการทำงาน	77
ภาคผนวก ข MEMORY MAP	98
ภาคผนวก ค การจัดหาไมโครโปรเซสเซอร์ Z80180	103
ภาคผนวก ง ASCII CODE TABLE	106
ภาคผนวก จ แผ่นวงจรพิมพ์	111
ภาคผนวก ฉ รายละเอียดของอุปกรณ์	121
ภาคผนวก ช CONTROL WORD 8255 MODE 0	143
บรรณานุกรม	

# สารบัญรูปภาพ

	หน้า
รูปที่ 2.1 แสดงบล็อกไดอะแกรมและโครงสร้างของ Z80180	3
รูปที่ 2.2 แสดง TIMING DIAGRAM สภาวะของ M1E	9
รูปที่ 2.3 แสดง TIMING DIAGRAM ของ IOC	9
รูปที่ 2.4 แสดงการจัดหน่วยความจำ	12
รูปที่ 2.5 บล็อกไดอะแกรมแสดงการทำ ASYNCHRONOUS SERIAL COMMUNICATION INTERFACE	25
รูปที่ 2.6 แสดงการจัดวางขาของ MSM 6242B	34
รูปที่ 2.7 โครงสร้างของ LCD	41
รูปที่ 2.8 แสดงโครงสร้างของโมเลกุลของผลึกเหลวทั้ง 3 แบบ(ก) นิเมติก (ข)โครสเทอริก (ค) สเมกติก	42
รูปที่ 2.9 การทำงานพื้นฐานของตัวแสดงผลแบบนิเมติกชนิดเกลียว (ก) ขณะไม่ทำงาน (ข) ขณะทำงาน	43
รูปที่ 2.10 การแสดงผลเชิงบวก	44
รูปที่ 2.11 ภาคแสดงผล 7 ส่วนแบบมีจุดทศนิยม (ก) ส่วนแสดงผล (ข) ส่วนจุดรวม	44
รูปที่ 2.12 แบบต่างๆของการแสดงผลของ LCD (ก) แบบส่งผ่าน(ข) แบบสะท้อน (ค) แบบส่งผ่าน / สะท้อน	44
รูปที่ 2.13 ความสัมพันธ์ระหว่างความเข้มของแสงกับแรงดัน	46
รูปที่ 2.14 วงจรสมมูลย์ของ LCD $R_1$ เป็นความต้านทานอนุกรมที่เกิดขึ้นจากตัวนำชนิดใส $R_2$ และ $C_1$ เป็นค่าที่เกิดจากผลึกเหลวที่อยู่ระหว่างตัวนำชนิดใส	47
รูปที่ 2.15 การใช้ RS 232C เชื่อมต่ออุปกรณ์	48
รูปที่ 2.16 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS 232C	49
รูปที่ 2.17 การกำหนดของขั้วต่อ RS-232	50
รูปที่ 2.18 แสดงขั้วต่อ RS-232 แบบ DB-9	51
รูปที่ 3.1 บล็อกไดอะแกรมแสดงระบบการทำงานของชุดฝึกและพัฒนา ไมโครโปรเซสเซอร์ Z80180	54
รูปที่ 3.2 แสดงชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180	55
รูปที่ 3.3 แสดงบอร์ด CP-JR180	56
รูปที่ 3.4 I/O MAP	57

	หน้า
รูปที่ 3.5 วงจรของบอร์ด MEMORY	58
รูปที่ 3.6 MEMORY DECODE	59
รูปที่ 3.7 วงจรของ CP-JR180	60
รูปที่ 3.8 บอร์ด MEMORY ขนาด 1 MB	61
รูปที่ 3.9 MEMORY MAP ที่ใช้สำหรับเขียนโปรแกรม	61
รูปที่ 3.10 PAGE MEMORY	62
รูปที่ 3.11 แสดงวงจร LED ASCII CODE	63
รูปที่ 3.12 แสดงวงจร USER PORT	64
รูปที่ 3.13 วงจร ANALOG TO DIGITAL CONVERSION และ DIGITAL TO ANALOG CONVERSION	65 68
รูปที่ 3.14 แสดง LCD DISPLAY , วงจรแสดงผล REGISTER FLAG และ USER PORT	66
รูปที่ 3.15 แสดงแป้น KEYBOARD	66
รูปที่ 3.16 บอร์ด ANALOG TO DIGITAL CONVERSION และ DIGITAL TO ANALOG CONVERSION	67
รูปที่ 4.1 แสดงผลการทดสอบ LED ASCII CODE และ USER PORT	70
รูปที่ 4.2 แสดงการวัดขา CHIP SELECT และแสดง LED FLAG	71
รูปที่ 4.3 แสดงผลการทดสอบจอ LCD	72
รูปที่ 4.4 แสดงการกดตัวอักษรบนแป้นพิมพ์และ LED ASCII CODE	72

## สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงสภาวะการทำงานโดยขา ST , HALT และ M1	5
ตารางที่ 2.2 แสดง INTERNAL I/O REGISTER	7-8
ตารางที่ 2.3 แสดงสภาวะการ INTERRUPT	18
ตารางที่ 2.4 แสดงการควบคุมการ REFRESH DYNAMIC RAM	20
ตารางที่ 2.5 แสดงสภาวะโหมดการทำงานของการทำ DMA	22
ตารางที่ 2.6 แสดงการใช้คำสั่ง DIM1 และ DIM2 กำหนด SOURCE และ DESTINATION	23
ตารางที่ 2.7 แสดงการทำ DMA CHANEL 0 กับ ASCII	24
ตารางที่ 2.8 แสดงการกำหนด BAUD RATE	29
ตารางที่ 2.9 แสดงการเลือกใช้ CLOCK	30
ตารางที่ 2.10 แสดงการใช้ TOC1.0 ควบคุมขาเอาต์พุต	32
ตารางที่ 2.11 แสดงข้อมูลของรีจิสเตอร์ W	36
ตารางที่ 2.12 แสดงคาบเวลาที่อินพุตคือ T1 และ T0	38
ตารางที่ 2.13 แสดง REGISTER TABLE	39
ตารางที่ 2.14 แสดงการกำหนดมาตรฐานแรงดันไฟฟ้า	48
ตารางที่ 2.15 คุณลักษณะโดยย่อของสัญญาณ RS 232C	52

# บทที่ 1

## บทนำ

โลกของระบบคอมพิวเตอร์ในปัจจุบันได้พัฒนาไปอย่างรวดเร็ว และมีส่วนเกี่ยวข้องในชีวิตประจำวันของมนุษย์มากขึ้นทุกที ระบบการควบคุมหลายๆอย่าง ถูกพัฒนาให้ทำงานภายใต้การควบคุมของคอมพิวเตอร์ ซึ่งจะได้ยินกันอยู่เสมอทั้งในบ้าน ในรถยนต์ สำนักงาน โรงงานอุตสาหกรรม ฯลฯ ระบบคอมพิวเตอร์ทำให้มนุษย์เราสามารถควบคุมการทำงานที่ซับซ้อน หรือการทำงานซ้ำๆกันได้ดี มีประสิทธิภาพสูง สิ่งสำคัญในระบบคอมพิวเตอร์ก็คือ ไมโครโปรเซสเซอร์ที่ควบคุมระบบนั้นๆ นับตั้งแต่จุดเริ่มต้นที่มนุษย์ได้ผลิตไมโครโปรเซสเซอร์ตัวแรกขึ้นมา ก็ได้มีการพัฒนาต่อมาเรื่อยๆ จนอาจจะแบ่งไมโครโปรเซสเซอร์เป็น 2 ส่วนคือไมโครโปรเซสเซอร์ที่ใช้ในการสร้างเครื่องคอมพิวเตอร์ส่วนบุคคลซึ่งจะเน้นการพัฒนาไปในด้านความเร็วของการประมวลผล หน่วยความจำที่สูงขึ้น และมีความซับซ้อนมากขึ้นด้วยเช่นกัน ส่วนอีกประเภทหนึ่งนั้นจะใช้ในการสร้างระบบควบคุมขนาดเล็กหรืออาจจะเรียกว่า ไมโครคอนโทรลเลอร์ ซึ่งจะเน้นการพัฒนาไปในด้านความแน่นอนในการทำงาน เนื่องจากส่วนใหญ่งานควบคุมจะอยู่ในสภาพแวดล้อมทางไฟฟ้าที่มีการรบกวนสูง และมีเนื้อที่จำกัด

Z80180 ก็เป็นไมโครโปรเซสเซอร์เบอร์หนึ่งที่น่าศึกษานำไปใช้งาน เนื่องจากระบบการทำงานไม่ซับซ้อนมากนัก ง่ายต่อการนำไปประยุกต์ใช้งาน มีคำสั่งในการใช้งานเหมือนกับ Z80 และยังเพิ่มคำสั่งขึ้นอีก 12 คำสั่ง เพื่อเพิ่มประสิทธิภาพการทำงานให้มากยิ่งขึ้น

## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 ไมโครโปรเซสเซอร์ Z80180

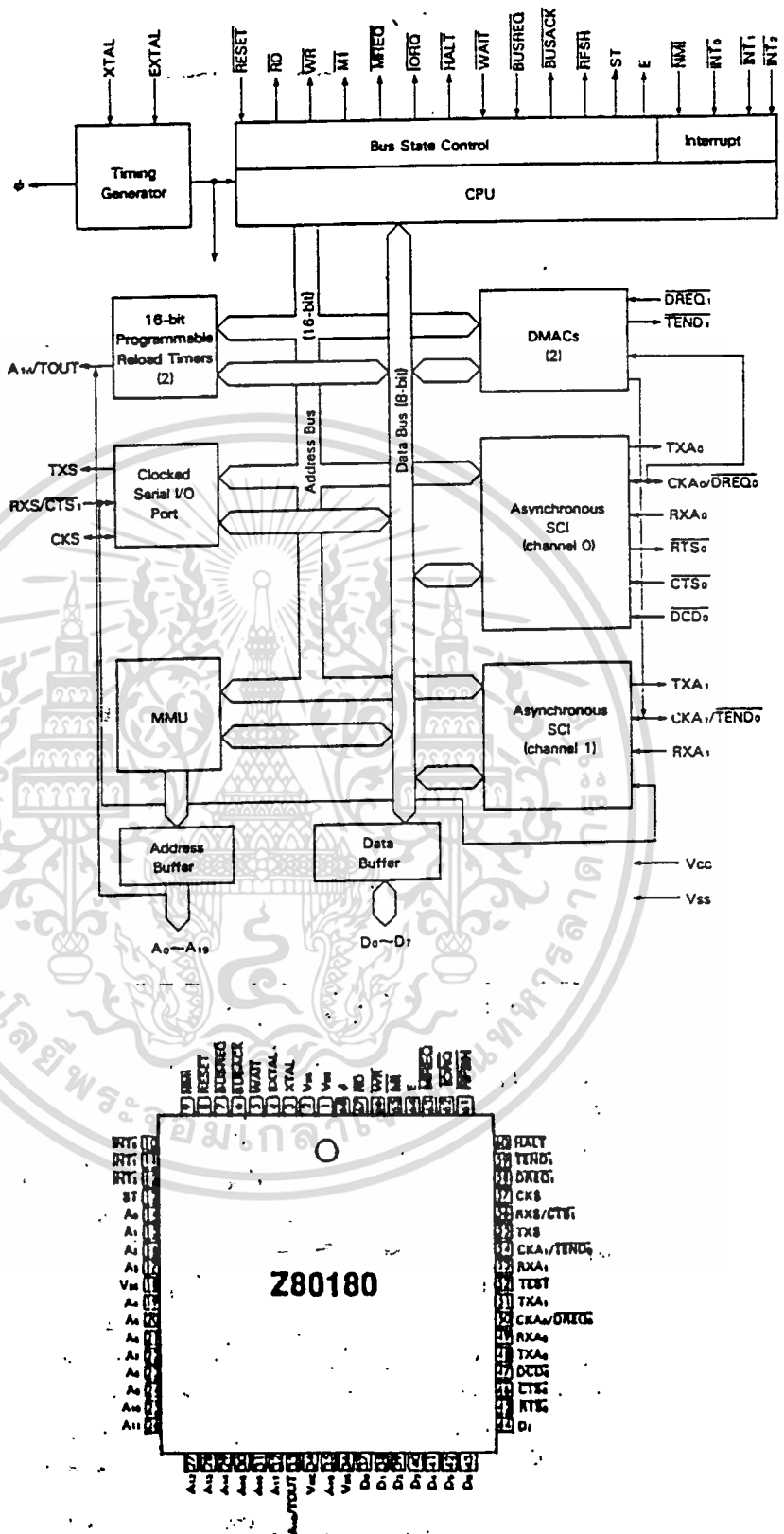
Z80180 เป็น CPU ที่มีความสามารถสูงที่ได้รับรวมชิพสำคัญอื่น ๆ ไว้ใน CPU ชิปเดียวจึงทำให้มีลักษณะคล้ายกับ CPU ที่ใช้ในงานควบคุมในจำพวก "ชิพเดี่ยว" แต่เนื่องจากชิพเดี่ยวมีข้อดี คือ เป็นระบบเล็กราคาถูก แต่ข้อเสียคือการโปรแกรมควบคุมค่อนข้างยากในตอนเริ่มต้นและกับระบบงานที่ใหญ่ขึ้น แต่ Z80180 ทางด้านโปรแกรมจะสะดวกอย่างมากเพราะคำสั่งที่ใช้มีมาก และตรงไปตรงมาทั้งคู่มือภาษาไทยและตัวอย่างการใช้งานอย่างมากเพราะ CPU Z80180 นี้เป็น SUPPER COMPAT Z80 คือคำสั่งทั้งหมดยังเป็น Z80 และได้เพิ่มชุดคำสั่งขึ้นมาเพื่อเพิ่มความสะดวกในการใช้งานขึ้นอีก

เมื่อมองดูระบบไมโครคอนโทรลเลอร์ชิพเดี่ยวแล้ว Z80180 จะดูดีกว่าตรงที่ไม่มี ROM , RAM และ PORT แต่ถ้าเป็นในระดับงานอุตสาหกรรมแล้วระบบของ Z80180 กับชิพไมโครคอนโทรลเลอร์แล้วจะไม่ต่างกันเลยเพราะความต้องการเนื้อหาในการเก็บข้อมูลมากและ PORT มากตามจึงทำให้ต้องต่อเพิ่มภายนอกขึ้นจึงทำให้ Z80180 ในระดับงานคอนโทรลอุตสาหกรรมค่อนข้างดีกว่ามากเพราะภายใน Z80180 ประกอบด้วย เป็น CMOS , OSCILLATOR ในตัวใช้งานที่ 10 MHZ , MMU ชิพ ย่างหน่วยความจำได้ ได้ 1 MBYTE , DMA 2 CHANEL , PORT สื่อสาร UART 2 CHANEL CLOCK SERIAL I/O , 16 BIT TIMER COUNTER และเกี่ยวกับ PORT สื่อสารสามารถทำ MULTI PROCESSOR COMMUNICATION ซึ่งโครงสร้างของชิพนี้จะเป็นดังรูปที่ 2.1

##### 2.1.1 ขาการใช้งาน

- AO-A19** ADDRESS BUS ระหว่าง RESET จะเป็น HIGH IMPEDANCE
- BUSAK** BUS ACKNOWLEDGE เป็นขา OUTPUT ACTIVE LOW ทำงานก็ต่อเมื่อ Z80180 ตอบสนองต่อการขอ BUS ของ BUSRQ และจะทำให้ BUS ข้อมูล BUS ADDRESS และสัญญาณ CONTROL บางเส้นเป็น HIGH IMPEDANCE
- BUSRQ** BUS REQUEST เป็นขา INPUT ACTIVE LOW ซึ่งจะมีความสำคัญสูงกว่า NMI โดยจะมีการตรวจสอบสัญญาณนี้ทุก ๆ การสิ้นสุดของ MACHINE CYCLE
- CXA0 , CXA1** ASYNCHRONOUS CLOCK และ 1 เป็นขาสัญญาณ CLOCK แบบ 2 ทิศทาง คือจะใช้เป็นขาอินพุทหรือเอาท์พุทก็ได้
- CXS** SERIAL CLOCK เป็นขา CLOCK 2 ทิศทางของ CSI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงบล็อกไดอะแกรมและโครงสร้างของ Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<b>CLOCK</b>	เป็นขาเอาต์พุทโดยจะเป็นครึ่งหนึ่งของคริสตอลหรือ CLOCK OUT เช่น คริสตอล 12 MHz Z80180 จะทำงานที่ 6 MHz
<b>CTS0-CTS1</b>	CLEAR TO SEND 0 และ 1 เป็นขา INPUT ACTIVE LOW ใช้ในการควบคุมโมเด็ม
<b>D0-D7</b>	DATA BUS เป็นแบบ 2 ทิศทาง
<b>DCDO</b>	DATA CARRIER DETECT 0 เป็นขา INPUT ACTIVE LOW ใช้ควบคุมในการติดต่อกับโมเด็มของ ASCII CHANNEL 0
<b>DREQ0-DREQ1</b>	DMA REQUEST 0 และ 1 เป็นขา INPUT ACTIVE LOW ใช้ในการขอ DMA และขานี้จะโปรแกรมได้เพื่อให้ตรวจสัญญาณที่ขอบหรือระดับได้
<b>E</b>	ENABLE CLOCK เป็นขา OUTPUT ACTIVE HIGH ซึ่งใช้ซิงค์การทำงานกับอุปกรณ์ภายนอกระหว่างการทำงานเกี่ยวกับ BUS และใช้เชื่อมต่อกับอุปกรณ์ภายในตระกูล 68XX และ 80XX
<b>HALT INTO</b>	เป็นขา OUTPUT ACTIVE LOW จะทำงานเมื่อคำสั่ง HALT หรือ SLP MASKABLE INTERRUPT 0 เป็นขา INPUT ACTIVE LOW สัญญาณที่ขานี้จะถูกตรวจทุก ๆ การสิ้นสุดของคำสั่ง
<b>INT1 , INT2</b>	เช่นเดียวกับ INTO แต่มีระดับความสำคัญรองลงมาตามลำดับ
<b>IORQ</b>	เป็นขาเอาต์พุทเพื่อบอกว่ากำลังติดต่อกับ I/O หรือขา IOE ใน 64180
<b>M1</b>	MACHINE CYCLE 1 เป็นขา OUTPUT ACTIVE LOW จะทำงานเมื่อ FETCH OP-CODE หรือเป็นขา LIR ของ 64180
<b>NMI</b>	NON MASKABLE INTERRUPT เป็นขา INPUT ACTIVE LOW ขานี้จะตอบรับการอินเทอร์รัพท์เสมอโดยไม่สามารถหยุดด้วยซอฟต์แวร์
<b>RD</b>	เป็นขาที่ใช้ทำการอ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์อินพุท/เอาต์พุท
<b>RFSH</b>	เป็นขาที่ให้ ADDRESS LOW (A0-A7) ไป REFRESH DYNAMIC RAM หรือขา REF ของ 64180
<b>RTS0</b>	REQUEST TO SEND เป็นขา OUTPUT ACTIVE LOW ขานี้ใช้โปรแกรมสัญญาณควบคุมโมเด็มของ ASCII CHANNEL 0
<b>RXA0 , RXA1</b>	RECEIVE DATA 0 และ 1 เป็นขารับสัญญาณจากพอร์ตอนุกรมของ ASCII
<b>RXS</b>	CLOCK SERIAL RECEIVE DATA เป็นขารับสัญญาณ SERIAL ของ CSI
<b>ST</b>	STATUS เป็นขา OUTPUT ACTIVE HIGH ใช้แสดงสถานะการทำงานของ CPU โดยร่วมกับ M1 และ HALT ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TEND0-TEND1** TRANSFER END 0 และ 1 เป็นขา OUTPUT ACTIVE LOW ใช้แสดงถึงว่าทำ DMA สิ้นสุดลงแล้ว

**TOUT** TIMER OUT ใช้กำเนิดพัลส์จาก PRT CHANEL 1

**TXA0 , TXA1** TRANSMIT DATA 0 และ 1 เป็นขาส่งข้อมูล SERIAL ของ ASCII

ST	HALT	M1	Operation
0	1	0	CPU operation (1st op-code fetch)
1	1	0	CPU operation (2nd op-code and 3rd op-code fetch)
1	1	1	CPU operation (MC except fo op-code fetch)
0	X	1	DMA operation
0	0	0	HALT mode
1	0	1	SLEEP mode (including SYSTEM STOP mode)

NOTE X: Don't care

MC: Machine cycle

ตารางที่ 2.1 แสดงสภาวะการทำงานโดยขา ST , HALT และ M1

**TXS** CLOCK SERIAL TRANSMIT DATA เป็นขาส่งข้อมูล SERIAL ของ CSIO

**WAIT** ขา INPUT ACTIVE LOW จะถูกตรวจที่ขอบขาของ CLOCK ลูกที่ 2 ของทุก ๆ MACHINE เพื่อเป็นการรอให้อุปกรณ์ภายนอกทำงานให้ทันกับการทำงานของ CPU

**WR** ใช้สำหรับการส่งข้อมูลไปยังอุปกรณ์อินพุท/เอาต์พุทหรือหน่วยความจำ

**X'TAL** เป็นขาที่ใช้ต่อกับคริสตอล

ขาที่ MULTIPLEX

**A18/TOUT** ระหว่างรีเซ็ตจะเป็น A18 แต่ถ้ามีการเลือก SET BIT TOC1 หรือ TOC0

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์กับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใน TIMER CONTROL REGISTOR (TCR) ก็จะทำหน้าที่เป็น TOUT  
**CKA0/DREQ0** ระหว่างรีเซ็ตขานี้จะเป็น CKA0 แต่ถ้า DM1 หรือ SM1 ใน DMA MODE  
 REGISTOR (DMODE) ถูกเซ็ตเป็น 1 จะเป็นขาค DREQ0  
**CKA1/TEND0** ระหว่างรีเซ็ตจะเป็นขาค CKA1 แต่ถ้า BIT CKA1D ใน ASCII ถูกเซ็ตจะเป็นขาค  
 TEND0  
**RXS/CTS1** ระหว่างรีเซ็ตขานี้จะเป็นขาค RXS ถ้า BIT CTIS1E ใน ASCII ถูกเซ็ตจะเป็นขาค  
 CTS1

### 2.1.2 INTERNAL I/O REGISTOR

ซึ่งมีด้วยกัน 64 I/O ADDRESS ดังแสดงในตารางที่ 2.2

จาก MAP I/O ภายในจะเห็นว่าโปรแกรม Z80 เก่าที่เรามีอยู่อาจจะมีการส่งพอร์ตเข้า  
 กับ I/O ภายใน ทำให้โปรแกรมเดิมทำงานไม่ได้ สามารถแก้ไขได้โดยการโปรแกรมย้าย MAP I/O  
 ภายใน โดยการ CONTROL BIT ใน REGISTOR I/O ADDRESS 3FH ซึ่งสามารถย้ายไปที่ใด  
 ก็ได้ภายใน 256 ตำแหน่ง ดังนี้



และการโปรแกรมจะเป็นดังนี้

IOA7	IOA6	ช่วง ADDRESS I/O
0	0	0000 - 003FH
0	1	0040 - 007FH
1	0	0080 - 00BFH
1	1	00C0 - 00FFH

เช่น ต้องการย้าย I/O ภายใน ไป I/O ADDRESS 80 H เป็นต้น จะโปรแกรมได้  
 เป็น - LD --A, 80H  
 OUT (3FH), A

ส่วน IOSTP : IOSTOP MODE BIT 5 เป็น 1 จะทำให้ I/O ภายในหยุด

ทำงานเมื่อรีเซ็ตบิตนี้จะเป็น 0

### 2.1.3 OPERATION MODE

Z80180 สามารถกำหนดการทำงานให้เหมือน 64180 ได้ SET BIT CONTROL MODE CONTROL REGISTOR (OMCR I/O ADDRESS 3EH)

**M1E (M1 ENABLE)** : ระหว่างรีเซ็ตบิตนี้จะเป็น 1 M1 เอาท์พุทจะเป็น LOW เมื่อ FETCH OP CODE และเนื่องจากการทำคำสั่ง RETI ของ Z80180 จะถูกกระทำ 2 ครั้งใน 1 คำสั่งจึงทำให้เกิด M1 ขึ้น 2 ครั้งด้วยอันอาจทำให้เกิดอินเตอร์รัพท์เข้ามาได้เมื่อยังทำไม่หมดคำสั่ง

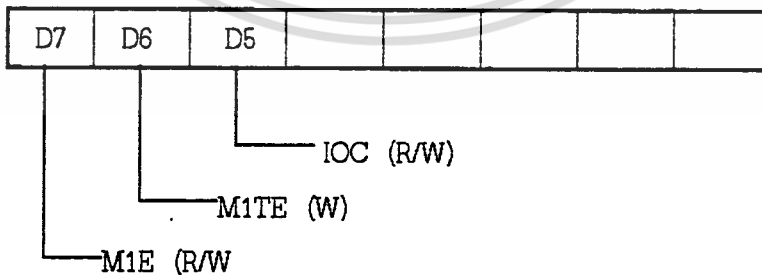
	Register	Mnemonic	Address	
			Binary	Hexadecimal
ASCII	ASCII Control Register A Ch 0	CNTLA0	XX000000	00H
	ASCII Control Register A Ch 1	CNTLA1	XX000001	01H
	ASCII Control Register B Ch 0	CNTLBO	XX000010	02H
	ASCII Control Register B Ch 1	CNTLB1	XX000011	03H
	ASCII Status Register Ch 0	STAT0	XX000100	04H
	ASCII Status Register Ch 1	STAT1	XX000101	05H
	ASCII Transmit Data Register Ch 0	TDR0	XX000110	06H
	ASCII Transmit Data Register Ch 1	TDR1	XX000111	07H
	ASCII Receive Data Register Ch 0	RDR0	XX001000	08H
	ASCII Receive Data Register Ch 1	RDR1	XX001001	09H
CSI/O	CSI/O Control Register	CNTR	XX001010	0AH
	CSI/O Transmit/Receive Data Register	TRDR	XX001011	0BH
Timer	Timer Data Register Ch 0L	TMDROL	XX001100	0CH
	Timer Data Register Ch 0H	TMDROH	XX001101	0DH
	Reload Register Ch 0L	RLDROL	XX001110	0EH
	Reload Register Ch 0H	RLDROH	XX001111	0FH
	Timer Control Register	TCR	XX010000	10H
	Reserved		XX010001	11H
			}	}
			XX010011	13H
	Timer Data Register Ch 1L	TMDR1L	XX010100	14H
	Timer Data Register Ch 1H	TMDR1H	XX010101	15H
Reload Register Ch 1L	RLDR1L	XX010110	16H	
Reload Register Ch 1H	RLDR1H	XX010111	17H	
Others	Free Running Counter	FRC	XX011000	18H
	Reserved		XX011001	19H
			}	}
			XX011111	1FH

### ตารางที่ 2.2 INTERNAL I/O REGISTOR

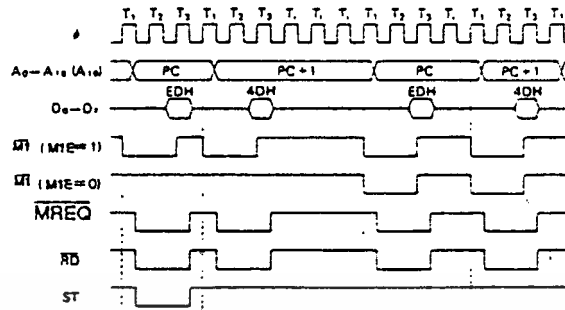
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Register	Mnemonic	Address	
			Binary	Hexadecimal
DMA	DMA Source Address Register Ch 0L	SAR0L	XX100000	20H
	DMA Source Address Register Ch 0H	SAR0H	XX100001	21H
	DMA Source Address Register Ch 0B	SAR0B	XX100010	22H
	DMA Destination Address Register Ch 0L	DAR0L	XX100011	23H
	DMA Destination Address Register Ch 0H	DAR0H	XX100100	24H
	DMA Destination Address Register Ch 0B	DAR0B	XX100101	25H
	DMA Byte Count Register Ch 0L	BCROL	XX100110	26H
	DMA Byte Count Register Ch 0H	BCROH	XX100111	27H
	DMA Memory Address Register Ch 1L	MAR1L	XX101000	28H
	DMA Memory Address Register Ch 1H	MAR1H	XX101001	29H
	DMA Memory Address Register Ch 1B	MAR1B	XX101010	2AH
	DMA I/O Address Register Ch 1L	IAR1L	XX101011	2BH
	DMA I/O Address Register Ch 1H	IAR1H	XX101100	2CH
	Reserved		XX101101	2DH
	DMA Byte Count Register Ch 1L	BCR1L	XX101110	2EH
	DMA Byte Count Register Ch 1H	BCR1H	XX101111	2FH
	DMA Status Register	DSTAT	XX110000	30H
	DMA Mode Register	DMODE	XX110001	31H
DMA/WAIT Control Register	DCNTL	XX110010	32H	
INT	IL Register (Interrupt Vector Low Register)	IL	XX110011	33H
	INT/TRAP Control Register	ITC	XX110100	34H
	Reserved		XX110101	35H
	Register	Mnemonic	Address	
			Binary	Hexadecimal
Refresh	Refresh Control Register	RCR	XX110110	36H
	Reserved		XX110111	37H
MMU	MMU Common Base Register	CBR	XX111000	38H
	MMU Bank Base Register	BBR	XX111001	39H
	MMU Common/Bank Area Register	CBAR	XX111010	3AH
I/O	Reserved		XX111011	3BH
			XX111101	3DH
	Operation Mode Control Register	OMCR	XX111110	3EH
	I/O Control Register	ICR	XX111111	3FH

ตารางที่ 2.2 INTERNAL I/O REGISTOR (ต่อ)



ด้วยเหตุนี้บิต M1E จะถูกเซ็ตเป็น 0 สำหรับ Z80180 เพื่อให้ M1 ถูกทำงานปกติคือเมื่อทำคำสั่ง RETI จะมี M1 เพียงครั้งเดียวดังรูป



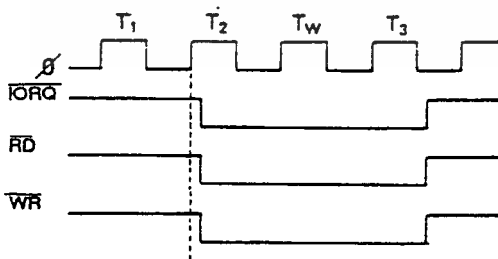
รูปที่ 2.2 TIMING DIAGRAM สภาวะของ M1E

**M1TE (M1 TEMPORARY ENABLE) :** ใช้กับการต่ออินเทอร์เฟซกับ Z80 PIO

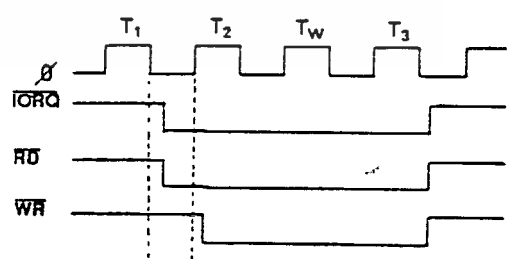
**IOC :** เป็นบิตใช้ควบคุม TIMING ของ IORQ และ RD ให้เหมือน Z80 หรือ 64180 โดยถ้าบิตนี้ถูกเซ็ตเป็น 1 TIMING จะเป็นของ 64180 คือ IORQ และ RD จะ ACTIVE ที่ขอบขาลงของ T1 แต่ถ้าบิตนี้เป็น 0 TIMING จะเป็นของ Z80 คือจะ ACTIVE ที่ขอบขาขึ้นของ T2 เพื่อให้ใช้อุปกรณ์สนับสนุนของ Z80 ได้ ระหว่างรีเซ็ตบิตนี้จะเป็น 1 ดังรูป

**2.1.4 เกี่ยวกับ TIMING**

Z80180 ใช้เวลาในการทำคำสั่งใน 1 MACHINE CYCLE น้อยกว่า Z80 อยู่ 1 T STATE คือใช้เวลาใน 1 MACHINE CYCLE เพียง 3 T STATE ในขณะที่ Z80 ใช้ 4 T STATE จะเห็นได้ว่าในขณะที่ให้ Z80180 ทำงานความถี่เดียวกันกับ Z80 CPU Z80180 ก็ยังให้ความเร็วกว่า Z80 ถึงอีก 25% แต่ในขณะเดียวกัน Z80180 ยังสามารถต่อ CLOCK สูงกว่า Z80 ได้มากกว่า 1 จึงทำให้ความเร็วในการทำงานของ Z80180 ดีกว่ามาก



I/O Read Write Cycle When IOC = 0



I/O Read Write Cycle When IOC = 1

รูปที่ 2.3 TIMING DIAGRAM ของ IO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.1.5 WAIT STATE GENERATOR

Z80180 ทำงานด้วยความถี่ที่สูงขึ้นจึงอาจทำให้หน่วยความจำหรือ I/O ทำงานไม่ทันจึงต้องมีสัญญาณมาเป็นตัวช่วยกำหนดความพร้อมระหว่าง CPU กับอุปกรณ์ภายนอกนั่นก็คือ สัญญาณ WAIT ซึ่ง STATE เพื่อเพิ่มเข้าไปในขณะที่ CPU ปฏิบัติคำสั่งหรือทำ DMA ด้วย

การโปรแกรมจะใช้ 4 BIT ของ DMA/WAIT CONTROL REGISTER (DCNTIL I/O ADDRESS 32H)

BIT	7	6	5	4	
	MWT1	MWIO	IWI1	IWIO	

BIT 7 , 6 MWI1 , MWIO (MEMORY WAIT INSERTION) จะทำการเพิ่มจาก 0-3 WAIT STATE ของการเข้าถึง MEMORY โดยการโปรแกรม

MWI1	MWIO	จำนวน WAIT STATE
0	0	0
0	1	1
1	0	2
1	1	3

BIT 5 , 4 IWI1 , IWIO (I/O WAIT INSERTION) จะทำการเพิ่ม WAIT STATE ให้กับ I/O ภายนอกจาก 1-6 ดังตาราง

IWI1	IWIO	I/O ภายนอก	INTO
0	0	1	2
0	1	2	4
1	0	3	5
1	1	4	6

จะเห็นว่า WAIT STATE ของ I/O มากกว่า MEMORY อยู่หนึ่ง T STATE เพราะขณะเข้าถึง I/O ปกติ WAIT จะถูกเพิ่มขึ้น 1 อยู่แล้ว ดังนั้นเมื่อเพิ่ม WAIT STATE เข้าไปก็จะรวมกับที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่ออยู่ปกติและส่วน INTO ก็เช่นเดียวกันขณะเกิด INTO ปกติจะมี WAIT STATE อยู่ 2 WAIT STATE และขณะที่รีเซ็ต บิต CONTROL WAIT STATE ทั้ง 4 จะเป็น 1 ทั้งหมดคือ อยู่ใน โหมดของ MAX WAIT STATE

ตัวอย่างเช่น เราต้องการเพิ่ม WAIT STATE ในการเข้าถึงหน่วยความจำ 2 WAIT STATE จะโปรแกรมดังนี้

```

IND  A  (32H)      ; IN ค่าในรีจิสเตอร์ DMA/WAIT
AND  0BFH         ; FILL เฉพาะบิต 7 และ 6 เท่านั้น
OUTC (32H) , A    ; ที่ใช้ IN แล้ว AND ก็เพราะว่ารีจิสเตอร์นี้มีการ
                  ; กำหนดเกี่ยวกับ DMA ดังนั้นเราจึง FILL เฉพาะบิต
                  ; ที่ต้องการโปรแกรม

```

### 2.1.6 HALT และ LOW POWER MODE

มีด้วยกัน 4 MODE คือ

**HALT MODE** โดยทำคำสั่ง 76H จะทำให้ CPU หยุดทำคำสั่ง แต่การทำงานต่าง ๆ ของ CPU ยังทำปกติการออกจาก HALT โดย RESET หรือ INTERRUPT

**SLEEP MODE** โดยการทำคำสั่ง SLP ซึ่ง CPU จะหยุด CLOCK ภายในทำให้ ADDRESS เป็น HIGH , DATA BUS เป็น TRISTATE , DRAM REFRESH , INTERNAL DMAC หยุดทำงานการออกจาก SLEEP MODE โดยการรีเซ็ตหรืออินเทอร์รัพท์

**IOSTOP MODE** ใช้หยุดการทำงานของชิพภายในคือ ASCII , CS/I/O และ PRT โดยการรีเซ็ตบิตใน I/O CONTROL REGISTOR (ICR I/O ADDRESS 3FH) เป็น 1 และจะให้ทำงานต่อก็ RESET หรือโปรแกรมให้บิตใน ICR เป็น 0

**SYSTEM STOP MODE** เป็นการรวมกันของ IOSTOP กับ SLEEP MODE โดยการรีเซ็ตบิตใน ICR แล้วตามด้วยคำสั่ง SLP จะทำให้ IO ภายในหยุดงานและ CPU หยุดทำงานเพื่อเป็นการประหยัดพลังงานซึ่งในโหมดนี้ CPU จะกินกระแสเพียง 7.5 MA ในขณะที่ปกติจะกินกระแสประมาณ 35 MA เมื่อจะออกจาก SYSTEM STOP MODE ก็โดยการรีเซ็ตหรืออินเทอร์รัพท์จากภายนอก

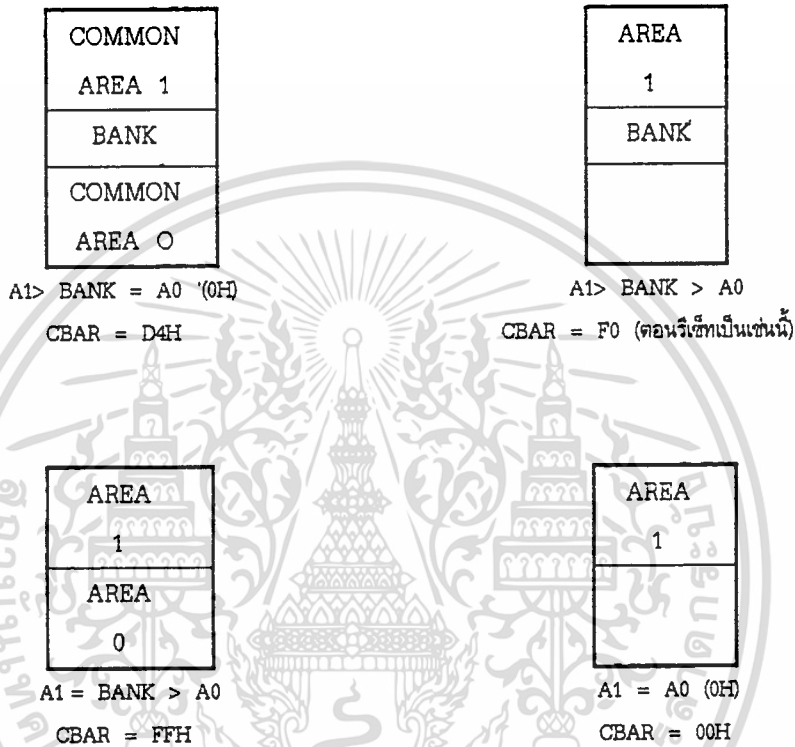
### 2.1.7 MEMORY MANAGEMENT UNIT (MMU)

ใช้เป็นตัวย้ายหน่วยความจำจาก 64 K (LOGICAL) เป็น 1 MBYTE (PHYSICAL) โดยการแบ่ง 64 K BYTE LOGICAL (คือ ADDRESS ปกติที่ใช้เช่นเดียวกับ Z80) เป็น 3 ส่วนในการใช้งานด้วยกัน คือ COMMON AREA 0 , BANK AREA และ COMMON AREA 1 โดยการกำหนดโปรแกรมจัด MAP LOGICAL ใน REGISTOR I/O CRAR (ADDRESS 3AH) ซึ่งในรีจิสเตอร์นี้จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ในการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบ่งเป็น 2 นิบเบิล คือ 4 บิตสูงและ 4 บิตต่ำ โดย 4 บิตสูงใช้โปรแกรมพื้นที่ของ COMMON AREA 1 และ 4 บิตต่ำใช้โปรแกรมส่วน BANK AREA ดังนั้นการโปรแกรมรีจิสเตอร์ CBAR นี้ก็จะจัด MAP ได้เป็น 2 คือ 4 รูปแบบ ดังรูป



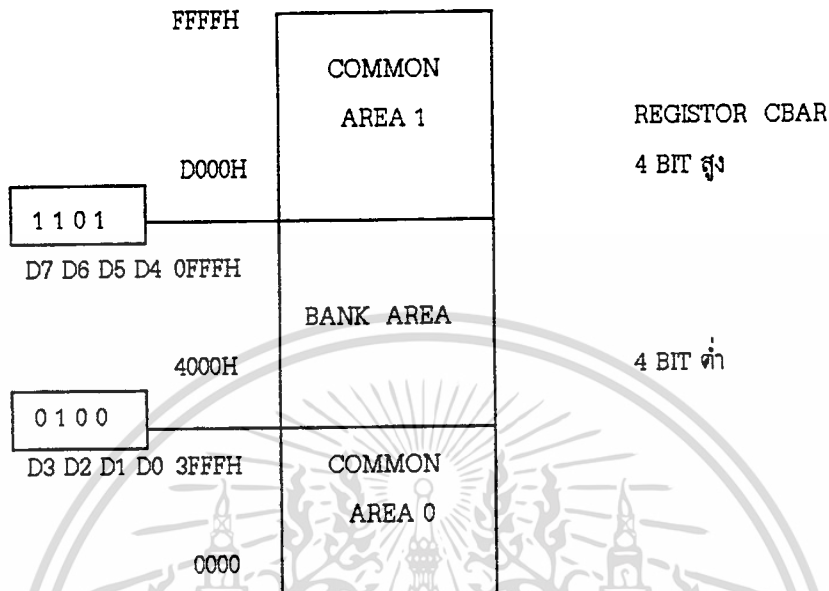
รูปที่ 2.4 แสดงการจัดหน่วยความจำ

จากรูปเป็นการโปรแกรมรีจิสเตอร์ CBAR ให้ MAP LOGICAL เป็น COMMON AREA 0 ตั้งแต่ ADDRESS 0000-3FFFH , BANK AREA ตั้งแต่ 4000H-CFFFH และ COMMON AREA ตั้งแต่ D000-FFFF ทั้งนี้เป็นไปตามค่าใน CBAR ทั้ง 2 นิบเบิล เพราะนิบเบิลสูงเป็นของ AREA 1 ซึ่งคือ 0DH ก็คือ AREA1 เริ่มแต่ D000H-FFFFH และนิบเบิลต่ำจะเป็นจุดสิ้นสุดของ BANK AREA ซึ่ง 04H ก็คือถัดจาก AREA 1 เป็นต้นไปจนถึง 4000H เป็น BANK ที่เหลือจึงเป็น AREA 0 นั่นเอง

จากค่าที่โปรแกรมใน CBAR จึงทำให้โปรแกรม COMMON AREA ทั้ง 2 และ BANK ได้ตั้งแต่ 4 KBYTE ขึ้นไปเช่น ให้นิบเบิลสูงของ CBAR = 0FH ก็คือ AREA 1 มีค่าตั้งแต่ F000-FFFFH (คือ 4 K อย่างต่ำนั่นเอง) และจุดที่น่าสังเกตจากการจัด MAP ทั้ง 4 รูปแบบนี้คือ COMMON 0 และ BANK สามารถมีตำแหน่งที่ทับซ้อนกันได้ (ตำแหน่งเดียวกัน) และ COMMON AREA 1 กับ BANK ก็สามารถโปรแกรมให้อยู่ที่ใดก็ได้ตัวอย่างอิสระตั้งแต่ 4 KBYTE ขึ้นไปของส่วน PHYSICAL ADDRESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## MAP LOGICAL ปกติ



(1 MBYTE โดยใช้ร่วมกับรีจิสเตอร์อีก 2 ตัว) แต่ส่วน COMMON AREA 0 แล้วจะเป็น BASED หรือ MONITOR ของระบบนั่นเอง

จากที่กล่าวมาเรายังไม่พูดถึงการขยายหน่วยความจำออกไปมากกว่า 64 K เพราะว่าการที่จะย้ายหน่วยความจำเกินกว่า 64 K นั้นจะต้องอิงส่วนของ LOGICAL ด้วย เนื่องด้วยคำสั่งของ Z80 ไม่สามารถอ้างหน่วยความจำเกินนี้ได้ ดังนั้นอ้างถึงหน่วยความจำทั้งหมดจึงยังเป็นส่วนของ LOGICAL แต่ข้อมูลที่ถูกระทำจริงจะเป็นส่วนของ PHYSICAL เช่น ในคำสั่งอาจเป็นดังนี้

```
LD A, (8000H)
```

ซึ่งดูจากคำสั่งนี่จะเป็นการทำกับตำแหน่ง 8000H (สมมุติในส่วนของ BANK AREA) แต่เราเห็น PHYSICAL AREA ไว้ที่ 10000H นั่นก็หมายความว่าการทำงานคำสั่งข้างบนนี้ข้อมูลจะถูกกระทำที่ ADDRESS 10000H นั่นเอง

### การคิด PHYSICAL ADDRESS

1) จะกระทำในส่วนของ BANK และ COMMON AREA 1 โดยผ่านทางรีจิสเตอร์ I/O CBR และ BBR คู่ด้วย 1000H แล้วนำค่าที่ได้บวกกับ LOGICAL ADDRESS ของส่วนนั้นๆ (BANK หรือ COMMON AREA 1)

2) การกระทำทั้งหมดเกิดขึ้นภายใน CPU เอง ดังนั้นการอ้าง ADDRESS ในโปรแกรมก็ยังเป็น 64 K คือตาม LOGICAL ที่กำหนดใน CBAR นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## REGISTER CONTROL

**CBAR** : COMMON/BANK AREA REGISTOR (I/O ADDRESS 3AH) ใช้กำหนดพื้นที่ของ LOGICAL ที่เป็น COMMON AREA 0 , BANK AREA และ COMMON AREA 1

BIT	7	6	5	4	3	2	1	0
	CA 3	CA 2	CA 1	CA 0	BA 3	BA 2	BA 1	BA 0

**CA 3 - CA 0** : เป็นตัวกำหนด ADDRESS เริ่มต้นของ COMMON AREA 1

**BA 3 - BA 0** : เป็นตัวกำหนดจุด ADDRESS สุดท้ายของพื้นที่ BANK AREA ที่ต่อจากจุดเริ่มต้นของ COMMON AREA 1

**CBR** : COMMON BASE REGISTOR (I/O ADDRESS 38H) เป็น REGISTOR I/O 8 บิตเพื่อใช้กำหนด PHYSICAL COMMON AREA 1

**BBR** : BANK BASE REGISTOR (I/O ADDRESS 39H) ใช้กำหนด PHYSICAL BANK AREA

ตัวอย่าง กำหนดให้มอนิเตอร์ที่ 0000H-7FFFH และ RAM ใช้งานที่ 10000H โดยมีพื้นที่ STACK มีเนื้อที่ 4 K นอกนั้นเป็น BANK จากนั้นก็หาค่าให้กับ BANK และ AREA 1 เช่น MAP LOGICAL กำหนดได้ดังนี้

หาค่าใส่กับ BBR และ CBR STACK ที่ 18000 H (PHYSICAL) ที่ LOGICAL เป็น F000 H ดังนั้นค่าที่ให้กับ CBR เป็น

FFFF	STACK
F000	RAM
EFFF	USER
8000	MONITOR
7FFF	
0000	

18000

- F000

09000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากที่ทราบแล้วว่า ค่าใน CBR จะคูณด้วย 1000 H ดังนั้นในทางกลับกันเมื่อจะนำค่า มาให้กับ CBR ก็ต้องทำการหารค่าผลต่างนั้นด้วย 1000 H ก็จะได้ค่าใน CBR = 09 H ส่วน RAMUSER (BANK) ก็เช่นเดียวกัน

10000

- 8000

08000

ที่ BBR = 08 H

ดังนั้นการโปรแกรมจากโจทย์ตัวอย่างก็จะเป็น

CBAR = 0F8H , BBR = 08 H และ CBR = 09 H

เมื่อคำนวณกลับจะได้ CBAR นิมเบิลต่ำ ADDRESS สุดท้ายของ BANK เป็น

8000 + (BBR = 8) X 1000 H = 10000 H

CBAR นิมเบิลสูง ADDRESS เริ่มต้นของ AREA 1 เป็น

0F000 + (CBR = 9) X 1000 H = 18000 H

การโปรแกรม

LD A , 0F8H

OUT0 (CBAR) , A

LD A , 8

OUT0 (BBR) , A

LD A , 9

OUT0 (CBR) , A

FFFFFH

FFFFFH

STACK

CBR = 9

CBAR 4 BIT สูง

1111

F000H

EFFFH

RAMUSER

BBR = 8

CBAR 4 BIT ต่ำ

1000

8000H

7FFFH

MONITOR

07FFFH

0000H

00000H

A1

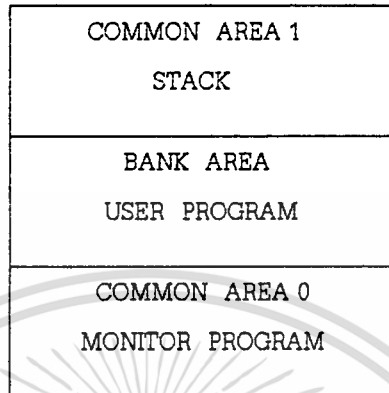
BANK

A0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ LOGICAL เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไป PHYSICAL

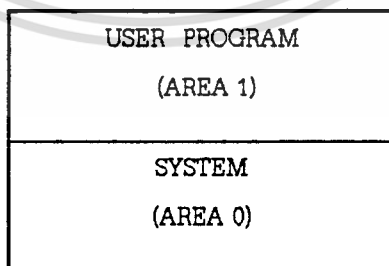
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อเรื่องนี้อ่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเราอาจจะกล่าวได้ว่า LOGICAL ส่วนมากจะถูกจัดเป็น



โดยให้ส่วนของ AREA 1 และ มอนิเตอร์ครึ่งที่ส่วนของ BANK ให้ย้ายไปที่ใดก็ได้ใน 1 MBYTE จะเป็นการขยายพื้นที่ของการใช้งานโดยใช้เนื้อที่ของ STACK เป็นการกระทำกับตัวแปรหรือ ข้อมูลอื่นใน 64K อื่นๆ เมื่อเรามอง LOGICAL 64K ออกเป็น PAGE ใดๆ ใน 1 MBYTE

แต่ข้อเสียในการจัดแบบนี้จะทำให้ใช้ BANK AREA ได้ไม่เต็มที่ เช่น เราต้องการใช้ RAM ถึง 32K เต็ม เช่น ให้ ROM MONITOR อยู่ที่ 0000-7FFFH และ RAM เริ่มตั้งแต่ 8000-FFFFH ซึ่งจะเห็นว่า RAM ในส่วนนี้จะต้องเป็น STACK ด้วยเมื่อเรายืด BANK ออกไปที่ PHYSICAL อื่นก็จะไม่สามารถใช้ได้ถึง 32K เช่นมี RAM ที่ตำแหน่ง 18000-1FFFFH อีกเราจะใช้ได้แค่ 24K เพราะพอเราอ้างที่ 0F000 แทนที่ข้อมูลจะถูกกระทำที่ 1F000H ก็จะมากกว่าที่ 0F000H แทนตามที่กำหนด AREA 1 ไว้ ใน LOGICAL เราจึงอาจแบ่ง MAP เป็นลักษณะกว้างๆดังนี้



CBAR = 80H

โดยกำหนดให้ AREA 1 เป็นส่วน USER PROGRAM ส่วนระบบเป็นของ COMMON AREA 0 ดังนั้น เมื่อเราให้ AREA 1 เริ่มที่ 8000H ก็จะใช้ RAM ได้ถึง 32 K เต็ม ส่วน SYSTEM ก็คือ ของ AREA 0 ซึ่งเป็นส่วนมอนิเตอร์แต่ในส่วนนี้เราได้กำหนดไว้ถึง 32 K คือ จาก 7FFF ลงไปถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

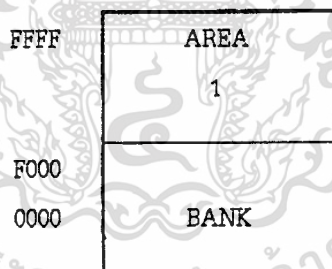
0000H ซึ่งในระบบเราอาจจะใส่ RAM ไว้ใน ADDRESS ช่วงนี้เพื่อเป็นเนื้อหาของ STACK ก็จะทำให้เราย้ายเนื้อหาของการใช้งานได้เต็ม

สรุป

- 1) ระหว่างรีเซ็ต LOGICAL ใน CBAR จะถูกกำหนดด้วยค่า 0F0H
- 2) ให้กำหนด MAP ADDRESS ของ LOGICAL ก่อนที่ CBAR (3AH)
- 3) BBR และ CBR จะเป็นตัวกำหนดตำแหน่งของข้อมูลในการใช้งานจริงในพื้นที่ 1 MBYTE (PHYSICAL ADDRESS)
- 4) การคิดค่า PHYSICAL ADDRESS คือ นำค่าใน BBR หรือ CBR คูณด้วย 1000H แล้วบวกด้วย LOGICAL ของพื้นที่นั้นๆ

ข้อสังเกต ในการจัดรูปแบบการ SET MAP ทั้ง 4 อย่าง ที่กล่าวในตอนต้น เช่น ตอนรีเซ็ต CBAR = F0 จึงเป็นดังรูป

ถ้าระบบของเรามี MONITOR ที่ 0-7FFFH RAM ที่ 8000-FFFFH โดยมี STACK ที่ FE00H ถ้าไม่มีการเข้าไป CONTROL REGISTOR ของ MMU ทั้ง 3 ตัวนี้ระบบของเรายังทำงานได้อยู่ แต่ถ้าเกิดเราโปรแกรมให้ BBR = 10 H ตอนนั้นระบบจะทำงานไม่ได้ทำไม่ถึงเป็นเช่นนั้น คำตอบก็คือ



เราไม่ได้โปรแกรม LOGICAL ให้มี AREA 0 แต่ที่ตอนแรกใช้งานได้เพราะ BANK AREA ทับซ้อน COMAREA 0 อยู่ แต่เมื่อโปรแกรม BBR = 10 H แล้ว BANK AREA เลยกกลายเป็น 10000H จึงทำให้ COMMON AREA 0 ไม่มีจึงทำให้ระบบทำงานไม่ได้

### 2.1.8 การอินเตอร์รัพท์

มีด้วยกัน 12 อินเตอร์รัพท์แบ่งเป็น 4 อินเตอร์รัพท์ภายนอก และ 8 อินเตอร์รัพท์ภายใน โดยมีลำดับความสำคัญจากมากไปหาน้อย ดังนั้น TRAP (ภายใน) , (ภายนอก) NMI ,INT0 , INT1 ,INT2 (ภายใน) TIMER 0 , TIMER 1 , DMA CHANAL 0 , DMA CHANAL 1 , CLOCK SERIAL , ASCI CHANAL 0 และ ASCI CHANAL 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### รีจิสเตอร์และ FLAG ที่ใช้ควบคุมการอินเทอร์รัพท์

อินเทอร์รัพท์ VECTOR LOW (IL) , อินเทอร์รัพท์ VECTOR HIGH (I) , อินเทอร์รัพท์ TRAP CONTROL (ITC) และ FLAG IEF1 , IEF2 โดยที่ FLAG IEF1 จะใช้ในการ ENABLE อินเทอร์รัพท์ภายในทั้งหมด ยกเว้น TRAP

#### อินเทอร์รัพท์ VECTOR LOW REGISTOR (IL I/O ADDRESS 33H)

ใช้เป็น VECTOR TABLE BYTE ค่า ของอินเทอร์รัพท์ภายนอก INT1 , INT2 และ อินเทอร์รัพท์ภายในทั้งหมดยกเว้น " TRAP " โดย 3 บิตสูงของ IL สามารถโปรแกรมได้ แต่ 5 BIT หลัง จะถูก FIX ดังตารางที่ 2.3

ดังนั้นการอินเทอร์รัพท์ส่วนใหญ่จะเป็น MODE 2 คือนำค่าใน I และ IL หรือจาก อุปกรณ์ที่ขออินเทอร์รัพท์ในกรณี INTO มาประกอบกันเป็น ADDRESS ที่จะเก็บข้อมูลที่จะกระโดดไป เช่น I = 10 H และ IL = 40 H และใน ADDRESS 1040H มีข้อมูล 00H, 60H ตามลำดับ เมื่อเกิด อินเทอร์รัพท์ขึ้นก็จะกระโดดไปทำโปรแกรมที่ตำแหน่ง 6000H นั้นเอง

Interrupt Source	Priority	IL			Fixed Code				
		b7	b6	b5	b4	b3	b2	b1	b0
INT 1	↑ Highest         ↓ Lowest	*	*	*	0	0	0	0	0
INT 2		*	*	*	0	0	0	1	0
PRT channel 0		*	*	*	0	0	1	0	0
PRT channel 1		*	*	*	0	0	1	1	0
DMA channel 0		*	*	*	0	1	0	0	0
DMA channel 1		*	*	*	0	1	0	1	0
CS/O		*	*	*	0	1	1	0	0
ASCI channel 0		*	*	*	0	1	1	1	0
ASCI channel 1		*	*	*	1	0	0	0	0

ตารางที่ 2.3 แสดงสภาวะการอินเทอร์รัพท์

#### INT/TRAP CONTROL REGISTOR (ITC ADDRESS I/O 34H)

BIT	7	6	5	4	3	2	1	0
	TRAP	UFO				ITE2	ITE1	ITE0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ITE2,1,0** : อินเทอร์รัพท์ ENABLE 2, 1, 0 ใช้ ENABLE และ DISABLE อินเทอร์รัพท์ภายนอก ถ้าเป็น 0 จะ DISABLE แต่บิตนี้จะไม่ทำให้เกิดอินเทอร์รัพท์ขึ้นทันทีจนกว่าจะทำคำสั่ง EI ดังนั้น INTO จะต่างกับ Z80 ตรงที่มีส่วนนี้ แต่เมื่อเกิดรีเซ็ต ITE0 จะถูกเซ็ตเป็น 1 โดยอัตโนมัติเพื่อให้อัปกับคำสั่ง EI หรือ DI อย่างเดียว เช่น Z80 แต่ ITE1 และ ITE2 จะเป็น 0

**TRAP** : จะเป็น 1 เมื่อทำคำสั่งที่ไม่มีใน Z80180 TRAP สามารถรีเซ็ตภายใต้โปรแกรมควบคุมได้ แต่ไม่สามารถเขียน 1 เข้าไปได้ระหว่างรีเซ็ตจะถูกเคลียร์

**UFO** : UNDEFINED FETCH OBJECT เมื่อ TRAP เกิดขึ้น UFO จะให้ค่าของ ตำแหน่งที่ผิดในคำสั่งนั้นไว้ใน STACK เนื่องจาก TRAP อาจเกิดขึ้นจาก OPCODE 2 หรือ 3 BYTE UFO จะปรับค่า PC ให้ คือ ถ้าเป็นคำสั่ง OPCODE 2 BYTE UFO จะเป็น 0 และจะทำให้ PC ของคำสั่งถัดไป จากคำสั่งที่ไม่ใช่ของ Z80180 ถูกลดลง 1 แต่ถ้า UFO = 1 คำสั่งที่ผิดจะมี OPCODE 3 BYTE และ PC จะถูกลดลง 2 ตำแหน่ง และค่า PC นี้จะถูกเก็บไว้ใน STACK เช่น

2000 ED 99

2002 PC ซ้ำคำสั่งถัดไป

เมื่อ CPU ทำงานมาพบข้อมูลที่ตำแหน่ง 2000 ก็เกิดอินเทอร์รัพท์ TRAP ขึ้น และรู้ด้วยว่าเป็นคำสั่ง 2 BYTE และ PC ก็ซ้ำคำสั่งถัดไปคือ ADDRESS 2002 แต่ FLAG UFO จะถูกทำให้เป็น 0 เพื่อปรับค่า PC นั้นด้วยการลดลง 1 เช่น ADDRESS 2001 ซึ่งก็คือตำแหน่งข้อมูลที่ผิดนั่นเอง

**TRAP INTERRUPT** : เป็นเหมือน NMI คือ ไม่สามารถหยุดได้เมื่อเกิดกระทำคำสั่งผิดขึ้นซึ่งเป็นตัวช่วยให้เกิดความน่าเชื่อถือทางด้าน SOFTWARE และอาจใช้เพิ่มคำสั่งได้อีกด้วยบิต TRAP ใน ITC จะถูกเซ็ตเป็น 1 และ UFO จะเซ็ตหรือไม่เซ็ตขึ้นอยู่กับว่าเป็นคำสั่ง 2 หรือ 3 ไบต์ และ FLAG UFO นี้ก็จะไปปรับ PC ให้ถูกต้อง และเก็บไว้ใน STACK แล้วกระโดดไป RUN ที่ ADDRESS 0000 H

### 2.1.9 DYNAMIC RAM REFRESH CONTROL

Z80180 ให้ ADDRESS A0-A7 สำหรับ DYNAMIC RAM และยังสามารถโปรแกรมเวลาในการ REFRESH โดยการโปรแกรมที่ RCR

**REFRESH CONTROL REGISTOR (RCR ADDRESS I/O 36H)**

BIT	7	6	5	4	3	2	1	0
	REFE	REFW	-	-	-	-	CYC1	CYC0

**REFE** : REFRESH ENABLE เมื่อเป็น 0 จะ DISABLE แต่ถ้าเป็น 1 จะให้สัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

REFRESH ระหว่างรีเซ็ตจะเป็น 1

**REFW** : REFRESH WAIT เป็น 0 จะให้สัญญาณ REFRESH ทุกๆ 2 CLOCK ถ้าเป็น 1 จะเพิ่ม REFRESH WAIT เข้าอีก 1 ระหว่างรีเซ็ตจะเป็น 1

**CYC1** , **CYC0** : CYCLE INTERVAL ใช้กำหนดช่วงเวลาในการ REFRESH เช่นกรณี DYNAMIC RAM จะต้อง REFRESH 128 ครั้ง ทุกๆ 2 ms (หรือ 256 ครั้ง ทุกๆ 4 ms) เพราะฉะนั้น สัญญาณ REFRESH แต่ละครั้งจะต้องไม่น้อยกว่าหรือเท่ากับ 15.625 us จากตาราง ค่าที่ขีดเส้นใต้เป็นค่าโปรแกรมที่เหมาะสมกับ CLOCK ที่ใช้ในระบบ

### 2.1.10 DMA CONTROLLER ( DMAC )

มีด้วยกัน 2 CHANNEL เพื่อเป็นการเพิ่มความเร็วในการ TRANSFER ข้อมูล โดยการกระทำไม่ต้องผ่าน CPU โดยมีความสามารถดังนี้

CYC1	CYC0	Insertion interval	Time interval			
			$\phi$ : 8 MHz	6 MHz	4 MHz	2.5 MHz
0	0	10 State	1.25 $\mu$ s	1.66 $\mu$ s	2.5 $\mu$ s	4.0 $\mu$ s
0	1	20 state	2.5 $\mu$ s	3.3 $\mu$ s	5.0 $\mu$ s	8.0 $\mu$ s
1	0	40 State	5.0 $\mu$ s	6.6 $\mu$ s	10.0 $\mu$ s	16.0 $\mu$ s
1	1	80 State	10.0 $\mu$ s	13.3 $\mu$ s	20.0 $\mu$ s	32.0 $\mu$ s

ตารางที่ 2.4 แสดงการควบคุมการ REFRESH DYNAMIC RAM

**MEMORY ADDRESS SPACE** โดยสามารถกำหนดตำแหน่งต้นทางและปลายทางที่ได้ก็ได้อีก 1024 K BYTE

**I/O ADDRESS SPACE** กำหนดที่ได้ก็ได้อีกใน 64 K BYTE ทั้งต้นทางและปลายทาง

**TRANSFER LENGTH** ใช้เป็น COUNTER ในการ TRANSFER ได้เป็น BLOCK ใดๆ 64 K BYTE

**DREQ** เป็นขาอินพุตจะตรวจจับที่ระดับหรือขอบของสัญญาณ

**TEND** เป็นขาเอาต์พุตเพื่อบอกกับอุปกรณ์ภายนอกว่าทำ DMA หมดบล็อกแล้ว

**TRANSFER RATE** การ TRANSFER แต่ละครั้งจะเกิดทุกๆ 6 CLOCK และ WAIT STATE สามารถเพิ่มเข้าไปใน DMA ได้ สำหรับ MEMORY หรือ I/O ที่ทำงานช้าที่ระบบ SYSTEM CLOCK ( $\phi$ ) = 6 MHz อัตราการ TRANSFER จะสูงถึง 1 M BYTE ใน 1 วินาที (ไม่มี WAIT STATE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ความสามารถของแต่ละ CHANEL

**CHANEL 0** สามารถ TRANSFER MEMORY MEMORY , MEMORY I/O , MEMORY MEMORY I/O MAP และสามารถให้ ADDRESS ในการ TRANSFER เพิ่ม, ลดหรือให้คงที่ได้ การ TRANSFER จะให้เป็นแบบ CYCLE STEAL (ขโมยเวลาเป็นช่วง) คือ เมื่อ TRANSFER ครบ 1 หรือ 2 BYTE ก็จะไปดึง BUS ให้ UP จนอุปกรณ์ก็จะ TRANSFER ข้อมูลต่อทำอย่างนี้สลับกันไปจนหมด BLOCK ใช้สำหรับอุปกรณ์ที่ทำงานช้า และ BURST (TRANSFER แบบต่อเนื่อง) คือ ทำจนจบ BLOCK จึงจะคืน BUS ให้ UP

**CHANEL 1** จะใช้กับ MEMORY I/O โดยMEMORY ADDRESS เพิ่มหรือลดได้

### DMAC REGISTER

**CHANEL 0** มี SARO (I/O ADDRESS 20H-22H) เป็นตัวกำหนด SOURCE ADDRESS ได้ถึง 1 M BYTE หรือ 64-K BYTE สำหรับ I/O

**DARO (I/O ADDRESS 23H-25H)** ใช้กำหนด DESTINATION เช่นเดียวกับ SARO

**BCRO (I/O ADDRESS 26H-27H)** ใช้กำหนดโบ้ทีในการ TRANSFER โดยสูงสุดได้ 64 K โบ้ที และเมื่อทำการ TRANSFER 1 BYTE REGISTER ตัวนี้จะลดลง 1

**CHANEL 1** มี MAR 1 (I/O ADDRESS 28H-2AH) ใช้กำหนด PHYSICAL ADDRESS ได้ถึง 1 M โบ้ทีโดยอาจจะให้เป็นต้นทางหรือปลายทางก็ได้

**IAR1 (I/O ADDRESS 2BH-2CH)** ใช้กำหนด ADDRESS ของ I/O โดยอาจจะให้เป็นต้นทางหรือปลายทางก็ได้

**BCR1 (I/O ADDRESS 2EH-2FH)** เช่นเดียวกับ BCRO

### REGISTER ที่เป็น COMMON

DMA STATUS REGISTER (DSTAT I/O ADDRESS 30H) ประกอบด้วย

BIT	7	6	5	4	3	2	1	0
	DE1	DE0	DWE1	DWE0	DIE1	DIE0	-	DME
	R/W	R/W	W	W	R/W	R/W		R

**DEL1 :** DMA-ENABLE CHANEL 1 เมื่อ DEL = 1 จะทำให้ DME = 1 DMA CHANEL 1 จะถูก ENABLE และเมื่อการ TRANSFER ลึ้นสุดลง (BCR1 = 0) เมื่อนั้น DE1 จะถูกเคลียร์เป็น 0 และถ้า DMA อินเตอร์รัพท์ถูก ENABLE (DIE1 = 1) CPU จะถูกอินเตอร์รัพท์ระหว่างรีเซ็ท DE1 จะถูกเคลียร์

**DE0 :** DMA ENABLE CHANEL 0 มีการทำงานลักษณะเดียวกับ CHANEL 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DWE1** : DE1 BIT WRITE ENABLE เมื่อมีการเขียนข้อมูลเข้าไปที่ DE1 ในขณะเดียวกันต้องเขียน DWE1 ด้วย 0 และค่านี้จะไม่คงอยู่ตลอดไป และถ้าอ่านจะเป็น 1 เสมอ

**DWE0** : DE0 BIT WRITE ENABLE เช่นเดียวกับ DWE1

**DIE1** : DMA INTERRUPT ENABLE CHANEL 1 เมื่อ DEL1 ถูกเซ็ทเป็น 1 และเมื่อการทำ DMA ลั้นสุดลง (เมื่อ DE1 = 0) ก็จะเกิดการอินเตอร์รัพท์ขึ้น แต่ถ้าบิตนี้ เป็น 0 จะเป็นการ DISABLE ระหว่างรีเซ็ทบิตนี้ จะเป็น 0

**DIE0** : DMA INTERRUPT ENABLE CHANEL 0 ลักษณะเช่นเดียวกับ DIE1

**DME** : DMA MAIN ENABLE เป็นบิตที่ใช้บอกการ ENABLE DMA เมื่อ DE BIT (DE0 , DE1) ถูกเซ็ทเมื่อนั้น DME BIT จะถูกเซ็ทเป็น 1 ซึ่งบิตนี้จะใช้อ่านอย่างเดียว DMA MODE REGISTOR (DMOD ADDRESS I/O 31H) ใช้กำหนด ADDRESS ในการ TRANSFER ของ CHANEL 0 ซึ่งประกอบด้วย

BIT	7	6	5	4	3	2	1	0
	-	-	DM1	DM0	SM1	SM0	MMOD	-

DM1	DM0	SM1	SM0	TRANSFER MODE	ADDRESS INCREMENT/DECREMENT
0	0	0	0	MEMORY TO MEMORY	SAR0+1 , DAR0+1
0	0	0	1	MEMORY TO MEMORY	SAR0-1 , DAR0+1
0	0	1	0	MEMORY* TO MEMORY	SAR0 FIXED , DAR0+1
0	0	1	1	I/O TO MEMORY	SAR0 FIXED , DAR0+1
0	1	0	0	MEMORY TO MEMORY	SAR0+1 , DAR0-1
0	1	0	1	MEMORY TO MEMORY	SAR0-1 , DAR0-1
0	1	1	0	MEMORY* TO MEMORY	SAR0 FIXED , DAR0-1
0	1	1	1	I/O TO MEMORY	SAR0 FIXED , DAR0-1
1	0	0	0	MEMORY TO MEMORY*	SAR0+1 , DAR0 FIXED
1	0	0	1	MEMORY TO MEMORY*	SAR0-1 , DAR0 FIXED
1	0	1	0	RESERVED	
1	0	1	1	RESERVED	
1	1	0	0	MEMORY TO I/O	SAR0+1 , DAR0 FIXED
1	1	0	1	MEMORY TO I/O	SAR0-1 , DAR0 FIXED
1	1	1	0	RESERVED	
1	1	1	1	RESERVED	

: INCLUDES MEMORY MAPPED I/O

## ตารางที่ 2.5 สภาวะโหมดการทำงานของการทำ DMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยบิตของ DM1 , DM0 เป็นตัวกำหนดเงื่อนไขปลายทางส่วน SM1 , SM0 ใช้กำหนดเงื่อนไขของต้นทางในระหว่างรีเซ็ต 4 บิตนี้จะเป็น 0 การโปรแกรมสามารถจัดได้ดังตารางที่ 2.5

**MMOD** : MEMORY MODE CHANEL 0 เมื่อ CHANEL 0 ถูกกำหนดการทำงานเป็น MEMORY ใช้อินพุทภายนอก DREQ0 จะไม่ถูกใช้ แต่จะถูกแทนด้วยการ TRANSFER อัตโนมัติ 2 แบบด้วยกัน คือ BURST MODE (MMOD = 1) คือ DMAC จะทำการ TRANSFER ข้อมูล

อย่างต่อเนื่องจนจบบล็อกที่กำหนด และ CYCLE STEAL MODE (MMOD = 0) คือ การ TRANSFER แบบขโมยเวลาของ CPU ซึ่งจะทำ DMA 1 ไบท์สลับกับ CPU ทำ 1 CYCLE (1 MACHINE) ไปเรื่อยๆจนกว่าจะจบบล็อกสำหรับ DMA กับ I/O ที่เป็นต้นทางหรือปลายทาง ใช้อินพุท DREQ0 จะถูกนำมาใช้และ MMODE จะไม่มีความหมาย ระหว่างรีเซ็ต MMOD จะถูกเคลียร์

**DMA/WAIT CONTROL REGISTOR (DCN1 I/O ADDRESS 32 H)** ประกอบด้วย

BIT	7	6	5	4	3	2	1	0
	MW11	MW10	IW11	IW10	DMS1	DMS0	DIM1	DIM0

**MW11 , MW10** : MEMORY WAIT STATE INSERTION ใช้กำหนดจำนวน WAIT STATE ของ CPU หรือ DMAC ดูหัวข้อ WAIT STATE GENERATOR

**IW11 , IW10** : I/O WAIT INSERTION กำหนด WAIT STATE I/O หรือ CPU ดูหัวข้อ WAIT STATE GENERATOR

**DMS1 , DMS0** : DMA REQUEST SENSE ใช้กำหนดการรับรู้ของขา INPUT DREQ0 , DREQ1 เมื่อเป็น 0 จะตรวจที่ LEVEL และถ้าเป็น 1 จะตรวจที่ EDGE ระหว่างรีเซ็ต 2 บิตนี้ จะเป็น 0

**DIM1 , DIM0** : DMA CHANEL 1 I/O และ MEMORY MODE ใช้กำหนดต้นทาง และ ปลายทางของ CHANEL 1 เมื่อ รีเซ็ต 2 บิตนี้จะเป็น 0 สามารถเห็นได้ดังตาราง

DIM 1	DIM0	TRANSFER MODE	ADDRESS INCREMENT/DECREMENT
0	0	MEMORY → I/O	MAR1+1 , IAR1 FIXED
0	1	MEMORY → I/O	MAR1-1 , IAR1 FIXED
1	0	I/O → MEMORY	IAR1 FIX , MAR1+1
1	1	I/O → MEMORY	IAR1 FIX , MAR1-1

ตารางที่ 2.6 การใช้คำสั่ง DIM1 และ DIM2 กำหนดต้นทางและปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การทำ DMA CHANNEL 0 กับ ASCII

ความสามารถพิเศษของ CHANNEL 0 คือ ทำ DMA กับ ASCII ได้ทั้ง 2 CHANNEL ในกรณีนี้ DREQ0 จะไม่ถูกใช้ แต่จะใช้ STATUS BIT ของ ASCII จะเป็นตัวกำหนด DREQ0 ภายในขึ้นโดยบิต TDRE และ RDRF สำหรับการส่งและรับ โดยให้กำหนดต้นทางและปลายทางที่ SAR0 และ DAR0 ให้เป็น ADDRESS ของ I/O ของการส่งและรับ (ASCII ADDRESS 6H-9H) และบิต A8-A15 ให้เป็น 0 และ บิต A17-A16 ต้องโปรแกรมดังตาราง

SAR18	SAR17	SAR16	DMA TRANSFER REQUEST
X	0	0	DREQ0
X	0	1	RDRF (ASCII CHANNEL 0)
X	1	0	RERF (ASCII CHANNEL 1)
X	1	1	RESERVED

DAR18	DAR17	DAR16	DMA TRANSFER REQUEST
X	0	0	DREQ0
X	0	1	TDRE (ASCII CHANNEL 0)
X	1	0	TDRE (ASCII CHANNEL 1)
X	1	1	RESERVED

ตารางที่ 2.7 การทำ DMA CHANNEL 0 กับ ASCII

จากนั้นกำหนดโหมดเป็น "EDGE SENSE" ส่วน ASCII ตอนเริ่มต้นต้องถูกกำหนดดังนี้ เวลาเริ่มต้น "EMPTY" คือ RDRF = 0 ส่วนการส่งต้อง "FULL" คือ TDRE = 0 (คือ BYTE แรกจะถูกกระทำโดย ASCII) ส่วนไบต์ถัดไป DMAC จะเป็นตัวจัดการ ขณะที่ DMA สามารถทำอินเตอร์รัพท์ NMI ได้ด้วย

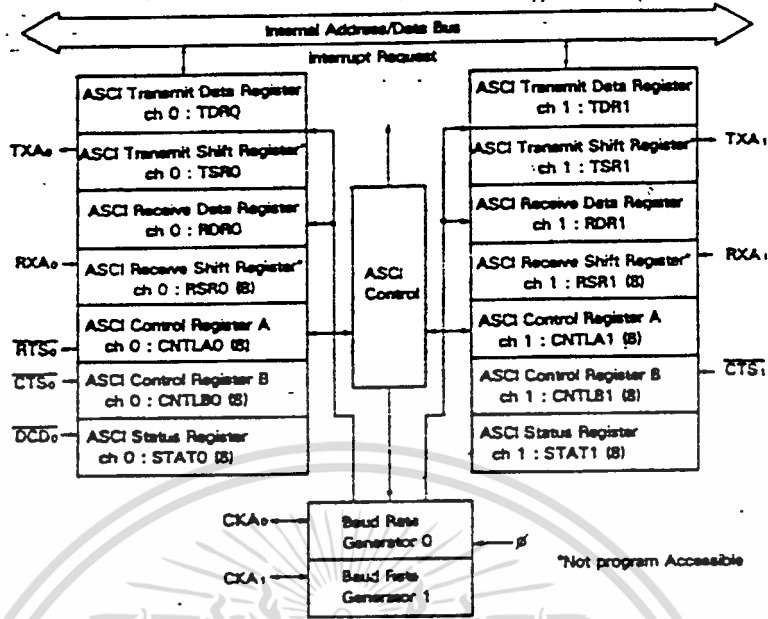
### ASYNCHRONOUS SERIAL COMMUNICATION INTERFACE (ASCII)

มีด้วยกัน 2 CHANNEL โดยมีบล็อกไดอะแกรมดังรูปที่ 2.4

**TSR 0, 1** : เป็นชิพที่รีจิสเตอร์ที่รับข้อมูลจาก TRANSMIT DATA REGISTOR (TDR) แล้วนำข้อมูลนั้นชิพที่ออกที่ขา TXA

**TDR 0, 1 (I/O ADDRESS 06H, 07H)** : เป็นรีจิสเตอร์ที่ใช้ส่งข้อมูลออกไปที่ขา TXA โดยการนำข้อมูลใน TDR ส่งไปที่ TSR เมื่อ TSR ว่างลง และสามารถที่จะเขียนข้อมูลเข้าไปที่ TDR ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 บล็อกไดอะแกรมแสดงการทำ ASYNCHRONOUS SERIAL COMMUNICATION INTERFACE

ในขณะที่ TSR กำลังชีพท์ข้อมูลออกไปที่ขา TXA

**RSR 0, 1 :** เป็นรีจิสเตอร์ที่รับข้อมูลจาก RXA PIN เมื่อรับเต็มบัพเพอร์แล้ว ก็ชีพท์ไปที่ RDR ถ้า RSR ไม่ว่างเมื่อมีการรับข้อมูลไบท์ต่อไปเข้าอีก จะเกิดข้อมูลทับซ้อนกันขึ้นจะทำให้การผิดพลาด และผลของการผิดพลาดก็จะแสดงที่รีจิสเตอร์สถานะรีจิสเตอร์นี้ไม่สามารถโปรแกรมได้

**RDR 0, 1 (I/O ADDRESS 08H, 09H) :** คือรีจิสเตอร์ที่ใช้เก็บข้อมูลที่รับเข้ามาจาก RXA PIN และในขณะที่ RDR กำลังบรรจุข้อมูลที่ได้รับเข้ามาจาก RSR ข้อมูลไบท์ถัดไปสามารถรับเข้ามาต่อได้

**STAT 0, 1 (I/O ADDRESS 04H, 05H) :** แต่ละ CHANNEL จะมีรีจิสเตอร์ใช้สำหรับตรวจสอบการสื่อสารเกี่ยวกับการผิดพลาด และสถานะสัญญาณควบคุมโมเด็มการ ENABLE และ DISABLE ASCII ดังรูป

STAT 0 : BIT

7	6	5	4	3	2	1	0
RDRF	OVRN	PE	FE	RIE	DCDO	TDRE	TIE

STAT 1 : BIT

7	6	5	4	3	2	1	0
RDRF	OVRN	PE	FE	RIE	CTSIE	TDRE	TIE

**RDRF** : RECEIVE DATA REGISTOR FULL จะถูกเซ็ตเป็น 1 เมื่อข้อมูลที่รับเข้ามาถูกเข้ามาที่ RDR เรียบร้อยแล้ว (ครบไบท์) แต่ถ้าการรับเกิด ERROR ขึ้น RDRF ก็จะถูกเซ็ตค้าง และข้อมูลที่ผิดนั้นก็จะถูกส่งมาที่ RDR และคงอยู่ ดังนั้นจะต้องทำการ CLEAR FLAG ERROR RDRF จะถูกเคลียร์เป็น 0 เมื่ออ่าน RDR , DCDO เป็น HIGH สำหรับ CHANEL 0 , IOSTOP และการรีเซ็ต

**OVRN** : OVERUN ERROR จะถูกเคลียร์ได้ เมื่อ EFR บิตใน CNTLA เป็น 0 , DCDO เป็น HIGH IOSTOP และรีเซ็ต

**PE** : PARITY ERROR เป็น 1 เมื่อข้อมูลที่รับเข้ามา PARITY ผิดและเคลียร์ได้ เช่นเดียวกับ OVERUN

**FE** : FRAMING ERROR เมื่อข้อมูลที่รับเข้ามารูปแบบผิดไปจากที่กำหนด BIT FE จะถูกเซ็ตเป็น 1 และการเคลียร์เช่นเดียวกับ OVERUN

**RIE** : RECEIVE INTERRUPT ENABLE เมื่อเป็น 1 จะอนุญาตให้ ASCII ทำการขออินเตอร์รัพท์ได้ เมื่อ RDRF , OVRN , PE หรือ FE ถูก SET เป็น 1 ด้วยเมื่อนั้น ASCII ก็จะทำให้สัญญาณอินเตอร์รัพท์สำหรับ ASCII CHANEL 0 อินเตอร์รัพท์สามารถเกิดขึ้น โดยการเปลี่ยนแปลงที่ขารับสัญญาณอินพุทภายนอกที่ขา DCDO จาก LOW เป็น HIGH และ RIE จะถูกเคลียร์เป็น 0 ระหว่างรีเซ็ต

**DCDO** : DATA CARRIER DETECT BIT นี้จะถูกเซ็ตเป็น 1 เมื่อขา INPUT DCDO เป็น HIGH และจะถูกเคลียร์เป็น 0 จากการอ่าน STAT 0 ครั้งแรก จากนั้นขาอินพุท DCDO จะถูกเปลี่ยนจาก HIGH เป็น LOW และระหว่างรีเซ็ตเมื่อ DCDO เป็น 1 ส่วนของภาครับจะไม่ทำงาน

**CTSIE** : CHANEL 1 CTS ENABLE ที่ CHANEL 1 มีขา INPUT CTS1 ภายนอก ซึ่ง MULTIPLEX กับ RXS เมื่อเซ็ตบิตนี้เป็น 1 จะถูกเลือกเป็นขา CTS1

**TDRE** : TRANSMIT DATD REGISTOR EMTPY เป็นตัวบอกว่าข้อมูลพร้อมที่จะส่งได้หรือไม่ ถ้าเป็น 1 คือ พร้อมที่จะส่งข้อมูลแล้วให้เขียนข้อมูลเข้าไปที่ TDR ได้ และเมื่อมีการเขียนข้อมูลเข้าไปที่ TDR ก็จะทำให้ TDRE เป็น 0 และข้อมูลใน TDR ก็จะถูกส่งให้ TSR จน TDR ว่างลง TDRE ก็จะถูกกลับ เป็น 1 อีกครั้ง

**TIE** : TRANSMIT INTERRUPT ENABLE เมื่อเป็น 1 จะอนุญาตให้ ASCII ใช้การส่งแบบอินเตอร์รัพท์ได้ โดยที่ TDRE ต้องเป็น 1 ด้วย TIE จะถูกเคลียร์เป็น 0 ระหว่างรีเซ็ต

**CNTLA 0 , 1 (I/O ADDRESS 00H-01H)** : เป็นรีจิสเตอร์กำหนดการทำงานประกอบด้วย

CNTLA 0 : BIT            7            6            5            4            3            2            1            0

MPE	RE	TE	RTSO	MPBR	MOD2	MOD1	MOD0
				EFR			

CNTLA 1 : BIT	7	6	5	4	3	2	1	0
	MPE	RE	TE	CKA1D	MPBR/ EFR	MOD2	MOD1	MOD0

**MPE** : MULTIPROCESSOR MODE ENABLE ใช้ ENABLE ในการสื่อสารแบบไมโครโปรเซสเซอร์ร่วมจากเมื่อมีการเลือกโหมดการสื่อสารแล้ว (MP = 1 ใน CNTLB) ในการสื่อสารแบบนี้ FORMAT ของการสื่อสารจะมีบิตพิเศษเพิ่มเข้ามาเรียกว่า MPB บิตซึ่งบิตนี้จะถูกใช้ในการตรวจสอบหรือใช้งาน เมื่อ ENABLE MPE ให้เป็น 1 และถ้า MPB = 1 เมื่อนั้นภาครับของ MULTIPROCESSOR จะทำงานคือ RDRF และ ERROR FLAG จะทำงาน และถ้า MPB = 0 ASCII จะไม่สนใจข้อมูลบิตนั้น ถ้า MPE = 0 จะไม่สามารถทำการสื่อสารแบบไมโครโปรเซสเซอร์ร่วมได้ แม้จะเซ็ท MP เป็น 1 แล้วก็ตาม

**RE** : RECEIVER ENABLE ถ้าเป็น 1 จะ ENABLE การรับของ ASCII แต่ถ้าเป็น 0 จะ DISABLE การรับ แต่ RDRF และ ERROR FLAG จะไม่ถูกรีเซ็ทตาม

**TE** : TRANSMIT ENABLE เป็น 1 จะ ENABLE การส่ง ถ้าเป็น 0 จะ DISABLE แต่ TDRE FLAG จะไม่ถูกรีเซ็ทตาม

**RISO** : REQUEST TO SEND CHANEL 0 เป็นบิตที่ให้ผลเช่นเดียวกับชาเอาร์ท์พุท RISO คือ ถ้าบิตนี้เป็น 1 ชาเอาร์ท์พุท RISO ก็จะเป็น 1 ถ้าบิตนี้เป็น 0 ชาเอาร์ท์พุทก็เป็น 0 RISO บิตนี้ จะถูกเซ็ทเป็น 1 ระหว่างรีเซ็ท

**CKA1D** : CKA1 CLOCK DISABLE ซึ่งชา CKA1 จะ MULTIPLAX กับ TENDO เมื่อบิตนี้เป็น 1 จะเลือกเป็นชา TENDO แต่ถ้าเป็น 0 ก็จะเป็นชา CLOCK ของ ASCII CHANEL 1 บิตนี้จะเป็น 0 ระหว่างรีเซ็ท

MPBR/EFR MULTIPROCESSOR BIT RECEIVE / ERROR FLAG เมื่อบิตนี้ถูกอ่านจะใช้ดู MPB บิตในกรณีที่พบไมโครโปรเซสเซอร์ที่จะทำการติดต่อกันแล้ว และอาจ DISABLE ไมโครโปรเซสเซอร์ตัวอื่นๆก็โดยการส่ง MPB บิตให้เป็น 0 เมื่ออ่านจะได้รู้ว่า MPB เป็น 0 จริง แต่ถ้าเขียน 0 ให้บิตนี้จะเป็นการรีเซ็ท ERROR FLAG ในการรับ

MOD 2, 1, 0 : ASCII DATA FORMAT MODE 2, 1, 0

โดย

MOD 2 = 0 7 BIT , 1 = 8 BIT

MOD 1 = 0 NOPARITY ,

1 = PARITY ENABLE

MOD 0 = 0 1 STOP BIT , 1 = 2 STOP BIT

สรุปได้ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOD 2	MOD 1	MOD 0	DATA FORMAT
0	0	0	START + 7 BIT DATA + 1 STOP
0	0	1	START + 7 BIT DATA + 2 STOP
0	1	0	START + 7 BIT DATA + PARITY + 1 STOP
0	1	1	START + 7 BIT DATA + PARITY + 2 STOP
1	0	0	START + 8 BIT DATA + 1 STOP
1	0	1	START + 7 BIT DATA + 2 STOP
1	1	0	START + 8 BIT DATA + PARITY + 1 STOP
1	1	1	START + 7 BIT DATA + PARITY + 2 STOP

รูปที่ 2.5 แสดงโหมด DATA FORMAT

ASCII CONTROL REGISTER B 0 , 1 (CNTLB 0 , 1 I/O ADDRESS 02H , 03H)

ประกอบด้วย

BIT	7	6	5	4	3	2	1	0
	MPBT	MP	CTS/PS	PEO	DR	SS2	SS1	SS0

**MPBT** : MULTIPROCESSOR BIT TRANSMIT ใช้ส่ง MPB บิตโดยถ้า MPBT = 1 เมื่อนั้น MPB BIT = 1 และ MPBT = 0 MPB ก็ = 0 ด้วย ระหว่างรีเซ็ตไม่สามารถกำหนดได้

**MP** : MULTIPROCESSOR MODE ถ้าเป็น 1 จะเป็นการเชื่อมต่อแบบไมโครโปรเซสเซอร์ร่วม โดยใช้ FORMAT ของ MOD 2 กับ MOD 0 โดยยกเว้น MOD 1 ดังนี้ START BIT + 7 หรือ 8 DATA บิต+ MPB บิต+ 1 หรือ 2 STOP บิตระหว่างรีเซ็ต MP จะเป็น 0

**CTS/PS** : CLEAR TO SEND / PRESCALE เมื่ออ่านบิตนี้จะใช้แสดงสถานะของขาอินพุท CTS ภายนอก ถ้าขา CTS เป็น HIGH ภาคส่งของ ACIS จะไม่ทำงาน แต่ถ้าเขียนเข้าไปที่บิตนี้จะเป็นการกำหนด BAUD RATE บิตนี้เป็น 0 ระหว่างรีเซ็ต

**PEO** : PARITY EVEN ODD บิตนี้จะไม่มีผลต่อการ ENABLE หรือ DISABLE ของ PARITY (MOD 1 ใน CNTLA) แต่จะใช้เลือกกว่าเมื่อมีการ ENABLE PARITY ใน MOD 1 จะให้ PARITY คู่หรือคี่ ถ้า PEO = 0 คือคู่ แต่ถ้า = 1 คือคี่

**DR** : DIVIDE RATIO ใช้กำหนด BAND RATE บิตนี้จะเป็น 0 ระหว่างรีเซ็ต

**SS2 , 1 , 0** : SOURCE/SPEED SELECT 2 , 1 , 0 ใช้กำหนด CLOCK ที่จะให้เป็นภายในหรือ

ภายนอก (โดยภายนอกคือ ขา CLOCK CKA) และเป็นตัวกำหนด BAUD RATE ด้วย ระหว่างรีเซ็ตทั้ง 3 บิตนี้จะเป็น 1 คือ เป็นการให้ CLOCK จากภายนอกนั้นเองซึ่งจากที่กล่าวมา ในการกำหนด BAUD RATE จึงมีด้วยกันหลายตัวสามารถสรุปเป็นตารางดังนี้

Prescaler		Sampling Rate		Baud Rate				General Divide Ratio	Baud Rate (Example) (BPS)			CKA	
PS	Divide Ratio	DR	Rate	SS2	SS1	SS0	Divide Ratio		$\phi=6.144$ MHz	$\phi=4.808$ MHz	$\phi=3.072$ MHz	I/O	Clock Frequency
0	$\phi+10$	0	16	0	0	0	+1	$\phi+160$	38400		19200		$\phi+10$
				0	0	1	2	320	19200	9600	20		
				0	1	0	4	640	9600	4800	40		
				0	1	1	8	1280	4800	2400	80		
				1	0	0	16	2560	2400	1200	160		
				1	0	1	32	5120	1200	600	320		
		1	1	0	64	10240	600	300	640				
		1	1	1	-	fc+16	-	-	fc				
		1	64	0	0	0	+1	$\phi+640$	9600		4800		$\phi+10$
				0	0	1	2	1280	4800	2400	20		
				0	1	0	4	2560	2400	1200	40		
				0	1	1	8	5120	1200	600	80		
1	0			0	16	10240	600	300	160				
1	0			1	32	20480	300	150	320				
1	1	0	64	40960	150	75	640						
1	1	1	-	fc+64	-	-	fc						
1	$\phi+30$	0	16	0	0	0	+1	$\phi+480$		9600		$\phi+30$	
				0	0	1	2	960	4800	2400	60		
				0	1	0	4	1920	2400	1200	120		
				0	1	1	8	3840	1200	600	240		
				1	0	0	16	7680	600	300	480		
				1	0	1	32	15360	300	150	960		
		1	1	0	64	30720	150	75	1920				
		1	1	1	-	fc+16	-	-	fc				
		1	64	0	0	0	+1	$\phi+1920$		2400		$\phi+30$	
				0	0	1	2	3840	1200	600	60		
				0	1	0	4	7680	600	300	120		
				0	1	1	8	15360	300	150	240		
1	0			0	16	30720	150	75	480				
1	0			1	32	61440	75	37.5	960				
1	1	0	64	122880	37.5	18.75	1920						
1	1	1	-	fc+64	-	-	fc						

ตารางที่ 2.8 แสดงการกำหนด BAUD RATE

### 2.1.11 CLOCK SERIAL I/O PORT (CSI/O)

มี 1 CHANEL ซึ่งเป็น SYNCHRONOUS SERIAL I/O พอร์ตโดยใช้ได้เฉพาะเป็น HALF-DUPLEX เท่านั้น และข้อมูลถูกกำหนดเป็น 8 บิตโดย CLOCK ที่ใช้ในการซิงค์เลือกได้ว่าจะใช้จาก SYSTEM CLOCK หรือ CLOCK ภายนอกที่ขา(CKS) ก็ได้ ซึ่ง CSI/O ประกอบไปด้วย 2 รีจิสเตอร์คือ

CSI/O TRANSMIT / RECEIVE DATA REGISTOR (TRDR I/O ADDRESS 0BH) ใช้ในการส่งและรับข้อมูล โดยระบบต้องเป็น HALF-DUPLEX (คือ การส่งและรับจะเกิดพร้อมกันไม่ได้)

CSI/O CONTROL / STATUS REGISTOR (CNTR I/O ADDRESS 0AH) เป็นตัวบอกสถานะและ CONTROL CSI/O ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT	7	6	5	4	3	2	1	0
	EF	EIE	RE	TE	-	SS2	SS1	SS0

**EF** : END FLAG เป็น 1 เมื่อ CS/I/O รับหรือส่งข้อมูลครบ 8 บิตแล้ว ซึ่งถ้า EIE ถูกเซ็ทเป็น 1 ไว้ ก็จะทำให้ CS/I/O ขออินเทอร์รัพท์ได้ ระหว่างรีเซ็ทบิตนี้จะเป็น 0 และใน IOSTOP โหมดด้วย

**EIE** : END INTERRUPT ENABLE เมื่อเป็น 1 จะเป็นการ ENABLE อินเทอร์รัพท์และจะอินเทอร์รัพท์เมื่อ EF = 1 ด้วย ระหว่างรีเซ็ท EIE = 0

**RE** : RECEIVE ENABLE ภาครับจะทำงานเมื่อ RE = 1 โดยข้อมูลจากขา RXS จะถูกชิพท์เข้ามาที่ TRDR โดยข้อมูลที่เข้าทาง RXS จะซิงค์กับสัญญาณ CLOCK ซึ่งจะเป็นภายในหรือภายนอก โดยการเลือก ถ้าเป็นภายในสัญญาณ CLOCK ที่ทำการหารแล้วก็จะออกที่ขา CKS ด้วย หรือถ้าเป็นภายนอกขา CKS ก็จะเป็นตัวรับสัญญาณ CLOCK เพื่อใช้ซิงค์นั่นเอง หลังจากที่ CS/I/O รับข้อมูลครบ 8 บิตแล้ว ก็จะ CLEAR RE โดย อัดโนมัติ และ EF จะถูกเซ็ทเป็น 1 ถ้าเกิดเซ็ท EIE ไว้ก็จะเกิดอินเทอร์รัพท์ขึ้นได้ และ RE กับ TE จะต้องไม่เป็น 1 พร้อมกัน

**TE** : TRANSMIT ENABLE ลักษณะการทำงานเช่นเดียวกับ RE แต่ TE นี้จะใช้ในการส่ง

**SS2, 1, 0** : SPEED SELECT 2, 1, 0 ใช้เลือก CLOCK ในการรับส่งดังรูป

SS2	SS1	SS0	DEVIDE RATIO	BAUD RATE
0	0	0	- 20	(200000)
0	0	1	- 40	(100000)
0	1	0	- 80	(50000)
0	1	1	- 160	(25000)
1	0	0	- 320	(12500)
1	0	1	- 640	(6250)
1	1	0	- 1280	(3125)
1	1	1	EXTENAL CLOCK INPUT (LESS THAN20)	

ตารางที่ 2.9 แสดงการเลือกใช้ CLOCK

( ) BAUD RATE ที่แสดง ที่ 0 = 4MHz

หลังจากรีเซ็ท CKS PIN จะถูกเลือกเป็นขาอินพุทเพราะบิต SS2, 1, 0 เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.12 PROGRAMMABLE RELOAD TIMER (PRT)

มีด้วยกัน 2 CHANEL เป็น 16 บิต PROGRAMMABLE RELOAD TIMER สำหรับ CHANEL 1 มีขาเอาต์พุตสามารถให้สัญญาณได้ทั้ง 2 CHANEL ประกอบด้วย

TIMER DATA REGISTOR (TMDR : I/O ADDRESS - CH0 ; 0DH ,0CH , CH1 ; 15H , 14H) เป็นรีจิสเตอร์ 16 บิตใช้กำหนด TIMER โดย ADDRESS I/O สูงเก็บค่า TIMER ค่าสูงระหว่างรีเซ็ต TMDR0 และ TMDR1 จะเป็น 0FFFFH โดย TMDR จะนับลง 1 ครั้งทุกๆ 20 CLOCK SYSTEM เมื่อ TMDR นับลงเป็น 0 ค่าใน RELOAD จะถูก LOAD มาให้ TMDR โดยอัตโนมัติ การอ่านค่าใน TMDR อ่านได้เลยโดยไม่ต้องหยุด PRT แต่ถ้าเป็นการเขียนต้องหยุด PRT ก่อน

TIMER RELOAD REGISTOR (RLDR : I/O ADDRESS - CH0 ; 0FH ,0EH , CH1 ; 17H , 16H) ใช้ LOAD ค่าที่อยู่ใน RLDR ไปให้ TMDR เมื่อ TMDR ลดลงเป็น 0

TIMER CONTROL REGISTOR (TCR : I/O ADDRESS 10H) เป็นรีจิสเตอร์ใช้แสดงสถานะและควบคุมดังนี้

BIT	7	6	5	4	3	2	1	0
	TIF 1	TIF 0	TIE 1	TIE 0	TOC 1	TOC 0	TDE 1	TDE 0

**TIF 1** : TIMER INTERRUPT FLAG 1 เมื่อ TMDR 1 ลดลงเป็น 0 TIF จะถูกเซ็ทเป็น 1 และถ้า TIE 1 = 1 ก็จะทำให้เกิด INTERRUPT ขึ้นได้ TIF จะถูกเคลียร์เป็น 0 ก็ต่อเมื่อการอ่านค่าในไบต์ HIGH หรือ LOW ของ TMDR 1 ระหว่างรีเซ็ต TIF = 0

**TIF 0** : TIMER INTERRUPT FLAG หลักการเช่นเดียวกับ TIF 1

**TIF 1 , 0** : TIMER INTERRUPT ENABLE 0 , 1 เมื่อเซ็ทเป็น 1 จะอนุญาตให้อินเตอร์รัพท์ได้ ระหว่างรีเซ็ต 2 บิตนี้จะเป็น 0

**TOC 1 , 0** : TIMER OUTPUT CONTROL 2 บิตนี้ใช้ควบคุมขาเอาต์พุตของ PRT 1 โดยถ้าทั้ง 2 บิตนี้ เป็น 0 จะเป็นการใช้งาน A18 นอกนั้นจะเป็นการกำหนดให้ TOUT เป็น HIGH , LOW หรือ TOGGLE ดังตารางระหว่างรีเซ็ต 2 บิตนี้จะเป็น 0

**TDE 1 , 0** : TIMER DOWN COUNT ENABLE เมื่อ SET เป็น 1 ก็คือให้เริ่มทำการนับ TMDR ได้ แต่ถ้าเป็น 0 การนับจะหยุดทำงาน ระหว่าง RESET BIT ทั้งคู่จะเป็น 0

การคำนวณเวลาเราทราบแล้วว่า TIMER จะนับลงทุกๆ 20 CLOCK ดังนั้นคริสตอลที่ใช้งาน คือ 12 MHz ความถี่ที่ทำงานบนบอร์ด = 6 MHz หากค่า TIMER คือ

$$T = 1/F = 0.1666 \text{ US ต่อ 1 CLOCK}$$

$$20 \text{ CLOCK} = 0.166 \times 20 = 3.333 \text{ uS}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOC 1	TOC 0	OUTPUT
0	0	ADDRESS A18
0	1	TOGGLE
1	0	0
0	1	1

ตารางที่ 2.10 แสดงการใช้ TOC1,0 ควบคุมเอาต์พุต

นั่นก็คือ TIMER นับลง 1 ครั้ง ทุกๆ 3.333  $\mu$ S ที่คริสตอล 12 MHz เช่นให้ TMDR มีค่า = 1 และเซ็ท FLAG อินเตอร์รัพท์ไว้ ก็จะทำให้ PRT เกิดการอินเตอร์รัพท์ทุกๆ 6.666  $\mu$ S เพราะการนับลงจะนับจากตัวเองลงก่อน คือ 1 แล้วก็ 0 จึงเท่ากับ 2 ครั้งนั่นเอง และถ้าให้กำเนิดสัญญาณสแควที่ TOUT ก็จะทำให้เกิดการ TOGGLE กันทุกๆจำนวนที่ให้นับ เช่น จากตัวอย่างข้างบนก็จะเป็น HIGH 6.666 US และ LOW 6.666 US ดังนั้น 1 ลูกสัญญาณ จะประมาณ 13  $\mu$ S หรือความถี่จะต่ำลงเท่าหนึ่งของค่าเวลาที่คิดจากจำนวนครั้งในการนับค่าก็ได้

### 2.1.13 SECONDARY BUS INTERFACE

E CLOCK OUTPUT TIMING เป็นสัญญาณบัสที่ 2 เพื่อใช้เชื่อมต่อกับอินเทอร์เฟซเป็นไปได้อย่างกับอุปกรณ์ PERIPHERAL ในตระกูลอื่นๆ เช่น 68XX และ 80XX และเป็นสัญญาณที่ทำให้ระบบเกิดความน่าเชื่อถือในการทำงาน เพราะจะติดต่อกับอุปกรณ์ก็ต่อเมื่อมีสัญญาณที่ขา MREQ หรือ IORQ จะเกิดขึ้นช่วงที่ T STATE แรกซึ่งยังไม่ใช่ช่วงของ DATA ที่อ่านหรือเขียนจึงทำให้อาจเกิดข้อมูลผิดพลาดกับอุปกรณ์ภายนอกแต่สัญญาณจะให้สัญญาณ ACTIVE HIGH เมื่อมีการจ่ายหรือรับข้อมูลเท่านั้น

### 2.1.14 FREE RUNNING COUNTER (18H)

เป็นรีจิสเตอร์ I/O ที่อ่านได้อย่างเดียวใช้สำหรับการ REFRESH DYNAMIC RAM ซึ่งเป็น COUNTER นับลง 8 บิต (A0 - A7) แบบอิสระโดยจะนับลง 1 ครั้ง ทุกๆ 10 CLOCK และถ้าเกิดการเขียนข้อมูลไปที่รีจิสเตอร์นี้ จะทำให้ช่วงเวลาของการ REFRESH DYNAMIC RAM , BAUD RATE ของ ASCII และ CS/IO ไม่นแน่นอน (คือจะไม่ถูกรับประกันว่าตรงตามที่กำหนดในคู่มือ)

ถึงแม้อยู่ใน IO STOP MODE ก็ตาม FREE RUNNING COUNTER นี้ก็ยังนับอยู่อย่างต่อเนื่องซึ่งในขณะรีเซ็ตจะมีค่าเป็น 0FFH

### 2.1.15 คำสั่งเพิ่มเติม 12 คำสั่ง

**SLP** : เมื่อใช้คำสั่งนี้ CPU จะหยุดทำงานบางอย่างทำให้ใช้กำลังงานต่ำ

**MLT** : MULTIPLY ใช้สำหรับคูณเลข 8 บิต 2 จำนวน โดยผลลัพธ์จะเป็น 16 บิต โดยรีจิสเตอร์ที่ใช้ในการคูณอาจจะเป็น BC , DE ,HL ,หรือ SP โดยผลลัพธ์จะได้ที่รีจิสเตอร์คู่นั้น

**OTIM , OTIMR , OTDM , OTDMR - BLOCK I/O** : เป็นคำสั่ง OUT PORT เป็นบล็อกของ PORT ADDRESS คำ A0 - A7 เท่านั้น คือ จะทำการ OUT ข้อมูลเป็นบล็อกโดยที่พอร์ตเพิ่มหรือลดตามจำนวนข้อมูล โดยใช้ HL เป็นตัวชี้ข้อมูลที่จะ OUT ออกไป และ C เป็น NUMBER พอร์ตในคำสั่ง OTIM และ OTDM ก็คือจะเพิ่มค่า HL ที่ชี้ขึ้นเป็นหนึ่งหรือลดลง 1 ตามด้วยพอร์ตเพิ่มขึ้นหรือลดลงด้วยและค่า B จะลดลง 1 ซึ่ง B จะเป็น COUNTER ในการส่งข้อมูล ส่วน OTIMR และ OTDMR จะมีลักษณะเช่นเดียวกับ OTIM และ OTDM เพียงแต่จะทำการส่งข้อมูลเพิ่มขึ้นหรือลง และพอร์ต NUMBER เพิ่มขึ้นหรือลดลงตามค่า B จนกระทั่ง B = 0

**TSTIO m** ใช้สำหรับ TEST I/O PORT คือจะทำการอ่านพอร์ตที่กำหนดโดยรีจิสเตอร์ C เข้ามาแล้วทำการ END กับข้อมูล 8 บิตที่ต้องการ โดยที่ค่าข้อมูลที่ IN เข้ามานั้นไม่เปลี่ยนแปลง แต่จะให้ผลที่ FLAG และพอร์ตที่ IN เข้ามาจะเป็นเฉพาะ ADDRESS คำ A0 -A7 เท่านั้น สามารถเปรียบเทียบเป็นโปรแกรมได้ดังนี้ :

```

XOR A          LD C , NUMBER PORT
IN A , (PORT)  TSTIO 70H
LD B , A       JP Z , OK
LD A , 70H
AND B
JP Z , OK

```

**TST g - TEST REGISTOR** . โดยค่าที่กำหนดในรีจิสเตอร์จะ AND กับ ACCUMULATOR ซึ่งจะทำให้มีผลต่อ FLAG ตามคำสั่ง AND แต่ค่าใน ACCUMULATOR และรีจิสเตอร์ ไม่เปลี่ยนแปลง ดังตัวอย่าง

```

LD A , 7          LD A , 7
LD C , A          TST B
AND B             JR Z , OK
LD A , C
JP Z , OK

```

TST m - TEST IMMEDIATE เช่นเดียวกับรีจิสเตอร์เพียงแต่ข้อมูลเป็น DATA โดยตรงที่ AND กับ ACCUMULATOR

TST (HL) - TEST MEMORY คือ จะนำค่าใน หน่วยความจำ ที่ถูกชี้โดย HL AND กับ ACCUMULATOR โดยค่าทั้ง 2 ไม่เปลี่ยนแปลงแต่ให้ผลการกระทำที่ FLAG

INO g (m) - INPUT , IMMEDIATE I/O IN ค่าจาก พอร์ต 8 บิต (A0 -A7) มายัง รีจิสเตอร์ ใดๆก็ได้ A , BC , DE , HL

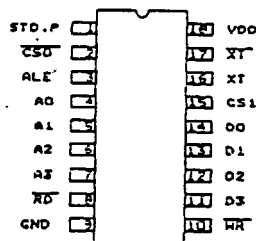
OUTO (m) , g - OUTPUT , IMMEDIATE I/O OUT ค่าจากรีจิสเตอร์ใดๆไปยัง พอร์ต 8 บิต (A0 - A7) รีจิสเตอร์ ก็มี A , BC , DE , HL

## 2.2 REAL TIME CLOCK

ในการนำไมโครโปรเซสเซอร์ไปใช้ในงานที่มีความเกี่ยวข้องกับเรื่องของเวลานั้น RTC เหมาะที่จะใช้กับงานลักษณะนี้เพราะสามารถบอกได้ทั้ง วัน,เดือน,ปี,วันในรอบสัปดาห์,ชั่วโมง,นาที,วินาที ในการติดต่อกับ RTC นั้นเปรียบเสมือนการติดต่อกับพอร์ตอินพุท/เอาต์พุท คือเราสามารถเขียนข้อมูลเกี่ยวกับเวลาไปที่เบอร์พอร์ทของ RTC ตัวนับเวลากายในก็จะเดินตามเวลาที่เรารตั้งให้ และเราก็สามารถอ่านข้อมูลจาก RTC ได้เช่นกัน

MSM 6242B สามารถที่จะปรับวันที่ให้ถูกต้องกับเดือนได้ไม่ว่าจะเป็นเดือนที่ลงท้ายด้วย"คม"ลงท้ายด้วย"ยน"หรือแม้กระทั่งเดือนกุมภาพันธ์ซึ่งปกติจะมี 28 วันแต่ในปีอธิกสุรทิน เดือนกุมภาพันธ์จะมี 29 วัน MSM 6242B ก็สามารถปรับวันที่ได้อย่างถูกต้อง

MSM 6462B เป็นไอซี REAL TIME CLOCK / CALENDAR ชนิด CMOS ใช้ต่อกับบัสของไมโครโปรเซสเซอร์ /ไมโครคอมพิวเตอร์ ได้โดยตรงมี ADDRESS BUS และ DATA BUS ขนาด 4 บิตมี CONTROL REGISTER ขนาด 4 บิต 3 ตัว คือ CD,CE,CF MSM 6242B โดยปกติจะทำงานที่ 5 V +10% ที่ -30 ถึง 25 C มี PACKAGE 3 แบบคือ 18 PIN PLASTIC DIP , 24 PIN PLASTIC FLAT PACKAGE และแบบ 12 PIN PLCC PACKAGE การจัดขาต่างๆดังแสดงในรูปที่ 2.6



รูปที่ 2.6 แสดงการจัดวางขาของ MSM 6242B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 18 PIN PLASTIC DIP PACKAGE

**A0-A3:** Address input

**D0-D3:** Data input/output

**CS0,CS1:** CHIP SELECT 0,1

**RD:** READ enable

**WR:** WRITE enable

**ALE:** Address latch enable

**STD.P:** Standard pulse output

**XT,XT:** XTAL oscillator input/output

**VDD:** +5V supply

**GND:** ground

### 2.2.1 การจัดหาและหน้าที่ของขาต่างๆ

MSM 6242B ได้ถูกออกแบบให้มาอินเทอร์เฟสเข้ากับ CPU ในตระกูล 8085 , MCS 48 ,Z80 และ Z80180 ได้ด้วย สำหรับหน้าที่ของขาต่างๆมีดังนี้

**D0-D3 (DATABUS)** เป็นบัสข้อมูลอินพุท/เอาต์พุทสามารถต่อเข้ากับบัสของไมโครคอนโทรลเลอร์ได้โดยตรงใช้ในการอ่าน และเขียนข้อมูลของรีจิสเตอร์ภายในที่เป็นนาฬิกา/ปฏิทินและรีจิสเตอร์ควบคุมโดย D0 = LSB และ D3 = MSB

**A0-A3 (ADDRESS BUS)** เป็นบัสแอดเดรสสำหรับติดต่อกับรีจิสเตอร์ภายในของ RTC เพื่อที่จะเขียนหรืออ่านข้อมูลในตำแหน่งนั้น ตำแหน่งของรีจิสเตอร์ต่างๆดูได้ในตารางที่ 2.10 A0-A3 จะใช้ร่วมกับ ALE สำหรับการอ้างตำแหน่งรีจิสเตอร์

**ALE (ADDRESS LATCH ENABLE)** เมื่อ CS0 = 0 และ ALE เปลี่ยนจากลอจิก "1" ไปเป็นลอจิก "0" แอดเดรสจะถูกแลทช์เอาไว้ในตัวของ RTC MICROCONTROLLER /MICROPROCESSORS ที่มีขา ALE เป็นเอาต์พุทควรจะต่อเข้ากับขา ALE ของ RTC ด้วย แต่ถ้าไม่มีขา ALE ให้ต่อขา ALE ของ MSM 6242B เข้ากับ VDD

**WR (WRITE ENABLE)** ใช้เขียนข้อมูลเข้าไปในรีจิสเตอร์ของ RTC แอคติฟที่ลอจิก "0" โดยที่ CS1 = 1 และ CS0 = 0

**RD (READ ENABLE)** ใช้อ่านข้อมูลจากรีจิสเตอร์ของ RTC แอคติฟที่ลอจิก "0" โดยที่ CS1 = 1 และ CS0 = 0 และ RD กับ WR จะต้องไม่แอคติฟพร้อมกัน

**CS0,CS1 (CHIP SELECT 0,1)** เป็น CHIP SELECT ทำหน้าที่ ENABLE/DISABLE การทำงานของ ALE, RD และ WR โดยที่ CS0 และ ALE จะทำงานร่วมกัน ส่วน CS1 กับ ALE จะ

เอกสารทำงานแยกกันที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**STD.P** (STANDARD PULSE OUTPUT) เป็นขาเอาต์พุตชนิด N-CH OPEN DRAIN ใช้ต่อเข้ากับขาอินเทอร์รัพท์ของ CPU

**XT,XT** (XTAL OSCILLATOR INPUT/OUTPUT) ต่อเข้ากับตัวคริสตอล 32.768 KHz ถ้าต้องการป้องกันความถี่จากภายนอก 32.768 KHz ทำได้โดยป้องกันความถี่เข้าที่ขา XT ถ้าความถี่มาจากเอาต์พุตของไอซี TTL ควรต่อ R PULL UP ไว้ด้วย ส่วนขา XT ควรปล่อยลอยไว้

**VDD** เป็นขา POWER SUPPLY +2 ~ +6V

**GND** ขา GROUND

## 2.2.2 รีจิสเตอร์ ต่างๆ

S1 , S10 , MI1 , MI10 , H1 , H10 , D1 , D10 , MO1 , MO10 , Y1 , Y10 , W

กลุ่มอักษรเหล่านี้เป็นชื่อย่อของรีจิสเตอร์ตามลำดับคือ SECOND1 , SECOND10 , MINUTE1 , MINUTE10 , HOUR1 , HOUR10 , DAY1 , DAY10 , MONTH1 , MONTH10 , YEAR1 , YEAR10 และ WEEK ในการกำหนดค่าให้รีจิสเตอร์เหล่านี้จะต้องให้เป็นรหัส BCD ตัวอย่างเช่น รีจิสเตอร์ S1 (S8,S4,S2,S1) = 1001 ซึ่งจะหมายถึง 9 วินาที

PM/AM,h20,h10 ในโหมด 24 ชม. บิต PM/AM จะไม่ใช้และจะอ่านบิตนี้ได้เป็น "0" ตลอดในขณะที่อยู่ในโหมด 12 ชม. บิต h20 จะถูกเซทในการอ่าน ถ้าบิต h0 ถูกเขียนด้วย "0" บิตนี้จะอ่านค่าได้เป็น "0" ตลอด ถ้าไม่มีการเขียน "1" เข้าไปที่บิตนี้

MSM 6242B ได้ถูกออกแบบมาสำหรับปีคริสต์ศักราช และยังสามารถจัดการเกี่ยวกับปีอธิกสุรทิน (LEAP YEAR) ได้อย่างอัตโนมัติ

ส่วนรีจิสเตอร์ W สามารถมีข้อมูลได้ตั้งแต่ 0-6

W4	W2	W1	DAY OF WEEK
0	0	0	SUNDAY
0	0	1	MONDAY
0	1	0	TUESDAY
0	1	1	WEDNESDAY
1	0	0	THURSDAY
1	0	1	FRIDAY
1	1	0	SATURDAY

### ตารางที่ 2.11 แสดงข้อมูลของรีจิสเตอร์ W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CD REGISTER (CONTROL D REGISTER)

- HOLD (D0) เมื่อเช็ทบิตนี้เป็น "1" สัญญาณ CLOCK 1 Hz ที่จะเข้ามาที่ S1 จะถูกหยุดไว้ ในเวลานี้เองบิต D1 (BUSY) ซึ่งเป็นบิตสถานะจะสามารถอ่านได้ เมื่อ BUSY เท่ากับ "0" รีจิสเตอร์ S1 ~ W สามารถอ่านหรือเขียนได้ ถ้าในช่วงเวลานี้มีตัวทวิตเกิดขึ้นที่วงจรรับของ S1 จะมิผลทำให้หลักหน่วยของวินาทีมีค่าเพิ่มขึ้นอีก 1 วินาที หลังจากที่ HOLD = 0 เมื่อ CS1 = 0 จะทำให้ HOLD = 0 โดยไม่สนใจสภาวะการอื่น ๆ

-BUSY(D1)เป็นบิตที่แสดงสถานะของการอินเตอร์เฟสกับไมโครคอนโทรลเลอร์/ไมโครโปรเซสเซอร์บิตนี้สามารถอ่านได้อย่างเดียวเท่านั้น เมื่อบิต BUSY = 0 หมายถึงพร้อมที่จะให้อ่านหรือเขียนกับรีจิสเตอร์ S1 W (ADDRESS 0 C)

- IRQ FLAG (D2) บิตสถานะนี้จะสัมพันธ์กับระดับสัญญาณของขา STD.P = 0 แล้ว IRQ FLAG จะเท่ากับ "0" IRQ FLAG จะเป็นตัวบอกไมโครคอมพิวเตอร์ว่า การอินเตอร์รัพท์เกิดจาก MSM 6242B (เมื่อ IRQ FLAG =1) บิตสถานะ IRQ FLAG จะทำงานร่วมกับรีจิสเตอร์ตัวอื่นอีกดังต่อไปนี้

- รีจิสเตอร์ CE บิต D0 (MASK) เมื่อ D0 (MASK) = 0 STD.P จะเปิด ในทางตรงกันข้ามถ้า D0 (MASK) = 0 STD.P เปลี่ยนสถานะตามเวลาที่กำหนดโดย D3 (T1) และ D2 (T0) ของรีจิสเตอร์ E

- เมื่อบิต D1 (INTRPT/STND) ของรีจิสเตอร์ E =1 (อินเตอร์รัพท์โหมด) เมื่อเกิดการอินเตอร์รัพท์เกิดขึ้น STD.P จะเป็น LOW จนกว่าบิต IRQ FLAG จะถูกเขียนด้วย "0" และถ้า IRQ FLAG เป็น "1" อยู่ และเกิดการอินเตอร์รัพท์ใหม่ขึ้นจะไม่มีผลต่อ STD.P (เป็น LOW เหมือนเดิม)

- เมื่อบิต D1 (INTRPT/STND) ของรีจิสเตอร์ E = 0 (STANDARD PULSE OUTPUT MODE) เมื่อเกิดการอินเตอร์รัพท์ STD.P จะยังคงเป็น LOW จนกระทั่งบิต IRQ FLAG ถูกเขียนด้วย "0" หรือเมื่อเวลาผ่านไป 7.8125 ms STD.P จะกลับเป็น HI โดยอัตโนมัติ

- เมื่อมีการเขียนไปที่บิต HOLD หรือ 30 SEC ADJUST ของรีจิสเตอร์ D จำเป็นจะต้องเขียน "1" ไปที่บิต IRQ FLAG ด้วย

- 30 ADJ (D3) ถ้าเช็ทบิตนี้ให้เป็น "1" ในขณะที่เราเช็ทเวลานั้น ถ้าหลักวินาทีนับไปได้ น้อยกว่า 30 วินาทีจะมีผลทำให้หลักวินาทีถูกเช็ทเป็น "00" วินาที แต่ถ้าเกิดว่าหลักวินาทีนับไปได้มากกว่า หรือเท่ากับ 30 วินาทีจะมีผลทำให้หลักวินาทีเพิ่มค่าขึ้นอีก 1 นาที แล้วหลักวินาทีจะถูกเช็ทให้เป็น "00" วินาที ในขณะที่เช็ทบิตนี้เป็น "1" นั้นไม่ควรที่จะอ่านหรือเขียนในเวลา 125 us หลังจากนั้นบิตนี้จะถูกเช็ทเป็น "1" มันจะเปลี่ยนกลับมากเป็น "0" อย่างอัตโนมัติหลังจากนั้นก็สามารอ่านหรือเขียนข้อมูลในรีจิสเตอร์ได้

**CE REGISTER (CONTROL E REGISTER)**

- MASK (D0) เป็นบิตที่ใช้ควบคุมเอาต์พุตของ STD.P เมื่อ MASK = 1 จะมีผลทำให้ STD.P = 1 (OPEN) คือ ไม่สามารถใช้บิตอื่นมาเปลี่ยนสถานะของ STD.P ได้ และเมื่อให้ MASK = 0 ก็จะทำให้ STD.P = OUTPUT MODE นั่นคือบิตอื่นๆสามารถควบคุมเอาต์พุตของ STD.P ได้ตามต้องการ ความสัมพันธ์ระหว่าง MASK บิตกับเอาต์พุตของ STD.P ดังแสดงในรูปที่ 4

- INTRPT/STND (D1) ใช้เป็นตัวเลือกสัญญาณเอาต์พุตของ STD.P ได้ 2 โหมดคือ อินเทอร์เน็ตร์รัฟท์กับ STANDARD TIMING WAVEFORMS (ผลิตพัลส์ออกมาด้วยคาบเวลาที่แน่นอน)

ถ้า INTRPT/STND = 1 และ MASK = 0 เมื่อเกิดการอินเทอร์เน็ตร์รัฟท์จาก RTC เอาต์พุตของ STD.P จะให้ลอจิก LOW จนกว่าจะเขียน "0" ไปที่ IRQ FLAG ในรีจิสเตอร์ C

ถ้า INTRPT/STND = 0 และ MASK = 0 จะส่งพัลส์ออกไปที่ขาเอาต์พุตของ STD.P โดยมี T1 และ T0 เป็นตัวกำหนดคาบเวลาในการอินเทอร์เน็ตร์รัฟท์ และมี LOW-LEVEL PULSE WIDTH ออกมาที่ขา STD.P ถ้าไม่มีการเขียนลอจิก "0" ไปที่ IRQ FLAG ความกว้างของพัลส์จะเท่ากับ 7.8125 ms

- T0 (D2) , T3 (D3) บิตทั้ง 2 นี้จะเป็นตัวกำหนดคาบเวลาของสัญญาณเอาต์พุตของ STD.P ของทั้ง 2 โหมด คือ INTERRUPT และ FIXED TIMING WAVEFORM ตารางข้างล่างจะแสดงถึงคาบเวลาซึ่งมี t0 และ t1 เป็นบิตอินเทอร์เน็ตร์รัฟท์ซึ่งมีส่วนเกี่ยวข้องกับ STD.P กับ INTRPT/STND

T1	T0	PERIOD	DUTY CYCLE OF "0" LEVEL WHEN INTRPT/STND BIT IS "0"
0	0	1/64 SECOUND	1/2
0	1	1 SECOND	1/128
1	0	1 MINUTE	1/7680
1	1	1 HOUR	1/460800

ตารางที่ 2.12 แสดงคาบเวลาอินเทอร์เน็ตร์รัฟท์คือ T1 และ T0

ถ้าเราให้บิต INTRPT/STND เป็น "1" ความสัมพันธ์ระหว่าง STD.P กับ MASK BIT จะเป็นดังนี้ คือ ถ้า STD.P เป็น "0" อยู่ก่อนแล้วเราเขียนให้ MASK BIT เป็น "1" ก็จะมีผลทำให้ STD.P เปลี่ยนเป็น "1" และในขณะที่ STD.P เป็น "1" อยู่ (โดยการ SET ให้ IRQ FLAG เป็น "0" ) แล้วเราให้ MASK BIT เป็น "1" ก่อนที่ STD.P จะเปลี่ยนมาเป็น "0" นั้นก็จะมีผลทำให้ STD.P เป็น "1"

ถ้าเราให้บิต INTRPT/STND เป็น "0" ความสัมพันธ์ระหว่าง STD.P กับ MASK BIT จะเป็นดังนี้คือ ถ้า STD.P เป็น "0" อยู่ก่อนแล้วเราเขียนให้ MASK BIT เป็น "1" ก็จะไม่มีการเกิดขึ้น และในขณะที่ STD.P เป็น "1" อยู่แล้วเราให้ MASK BIT เป็น "1" ก่อนที่ STD.P จะเปลี่ยนมาเป็น

REGISTER TABLE

Address Input	Address Input				Register Name	Data				Count value	Description
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>		D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>		
0	0	0	0	0	S <sub>1</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	0 ~ 9	1-second digit register
1	0	0	0	1	S <sub>10</sub>	S <sub>9</sub>	S <sub>8</sub>	S <sub>7</sub>	S <sub>6</sub>	0 ~ 5	10-second digit register
2	0	0	1	0	M <sub>1</sub>	m <sub>3</sub>	m <sub>2</sub>	m <sub>1</sub>	m <sub>0</sub>	0 ~ 9	1-minute digit register
3	0	0	1	1	M <sub>10</sub>	m <sub>9</sub>	m <sub>8</sub>	m <sub>7</sub>	m <sub>6</sub>	0 ~ 5	10-minute digit register
4	0	1	0	0	H <sub>1</sub>	h <sub>3</sub>	h <sub>2</sub>	h <sub>1</sub>	h <sub>0</sub>	0 ~ 9	1-hour digit register
5	0	1	0	1	H <sub>10</sub>	H	PM/AM	h <sub>10</sub>	h <sub>9</sub>	0 ~ 2 or 0 ~ 1	PM/AM, 10-hour digit register
6	0	1	1	0	D <sub>1</sub>	d <sub>3</sub>	d <sub>2</sub>	d <sub>1</sub>	d <sub>0</sub>	0 ~ 9	1-day digit register
7	0	1	1	1	D <sub>10</sub>	D	D	d <sub>10</sub>	d <sub>9</sub>	0 ~ 3	10-day digit register
8	1	0	0	0	MO <sub>1</sub>	mo <sub>3</sub>	mo <sub>2</sub>	mo <sub>1</sub>	mo <sub>0</sub>	0 ~ 9	1-month digit register
9	1	0	0	1	MO <sub>10</sub>	M	M	mo <sub>10</sub>	mo <sub>9</sub>	0 ~ 1	10-month digit register
A	1	0	1	0	Y <sub>1</sub>	y <sub>3</sub>	y <sub>2</sub>	y <sub>1</sub>	y <sub>0</sub>	0 ~ 9	1-year digit register
B	1	0	1	1	Y <sub>10</sub>	y <sub>9</sub>	y <sub>8</sub>	y <sub>7</sub>	y <sub>6</sub>	0 ~ 9	10-year digit register
C	1	1	0	0	W	W	w <sub>3</sub>	w <sub>2</sub>	w <sub>1</sub>	0 ~ 5	Week register
D	1	1	0	1	C <sub>3</sub>	30sec. ADJ	IRQ	BUSY	HOLD	-	Control register D
E	1	1	1	0	C <sub>5</sub>	l <sub>1</sub>	l <sub>0</sub>	INTRPT/STND	MASK	-	Control register E
F	1	1	1	1	C <sub>7</sub>	TEST	24/12	STOP	REST	-	Control register F

REST = RESET

INTRPT/STND = INTERRUPT/STANDARD

Note 1) - Bit E does not exit (unrecognized during a write and held at "0" during a read).

Note 2) - Be sure to mask the AM/PM bit when processing 10's of hour's data.

Note 3) - BUSY bit is read only. The IRQ FLAG bit can only be set to a "0". Setting the IRQ FLAG to a "1" is a

ตารางที่ 2.13 แสดง REGISTER TABLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0" นั่นก็จะมีผลทำให้ STD.P เป็น "1"

#### CF REGISTER (CONTROL F REGISTER)

- REST (D0) บิตนี้จะใช้ในการเคลียร์CLOCK ภายในที่ใช้ในการนับ/หารของวินาที เมื่อ REST = 1 จะทำให้ STD.P = 1 และวงจรมันภายในจะถูกรีเซ็ตและเมื่อต้องการให้วงจรมัน

ภายในทำงานต่อ (นอกจากการ RESET ) จำเป็นจะต้องให้ REST = 1 ถ้า CS1 = 1 ดังนั้น REST = 1 อย่างอัตโนมัติ

- STOP (D1) จะใช้ในการหยุดตัวหอดที่จะเข้าไปในวงจรความถี่ 8192 Hz และจะมีการหน่วงเวลาไป 122 us ก่อนที่เวลาจะทำการเดินหรือหยุดเดิน หลังจากที่มีการเปลี่ยนสถานะของ FLAG นี้ เป็น "1" = STOP / เป็น "0" = RUN ในขณะที่จะเซ็ทเวลาให้ RTC นั้น ควรให้บิตนี้เป็น "1" เพื่อไม่ให้ตัวหอดเข้ามาที่หลักวินาที หลังจากเซ็ทเวลาให้ RTC เสร็จแล้วจึงให้บิตนี้เป็น "0" เพราะว่าเป็นเวลาที่เราระเบิดเวลานั้นเกิดมีตัวหอดเข้ามาจะทำให้หลักวินาทีเพิ่มค่าขึ้นอีก 1 วินาที หลังจากที่เราเซ็ทเวลาเสร็จแล้ว

- 24/12 (D2) บิตนี้จะเป็นการเลือกว่าจะให้เวลาเดินแบบ 24 ชม. หรือ 12 ชม. (มี AM/PM) ถ้าเลือกโหมด 1-24 ชม. บิต PM/AM จะไม่ถูกนำมาใช้ (มีค่าเป็น "0") แต่ถ้าเลือกโหมด 0-12 ชม. บิต PM/AM จะมีการเปลี่ยนสถานะไปด้วยสำหรับการเซ็ทบิตมีขั้นตอนดังนี้

1) ต้องให้ REST BIT = 1

2) 24/12 HOUR BIT = 0 หรือ 1 ; ถ้าเป็น "0" หมายถึง โหมด 24 ชม. ถ้าเป็น "1"

หมายถึงโหมด 24 ชม.

3) REST BIT = 0

หมายเหตุ REST จะต้องเป็น "1" ถึงจะเขียนบิต 24/12 ได้

- TEST (D3) เมื่อบิตนี้เป็น "1" อินพุทของวงจรมันในหลักวินาทีจะมาจากวงจรมัน/หารแทนที่จะมาจากภาคหาร 15

ดังนั้นวงจรมันหลักวินาทีจะนับความถี่ที่ 5.4163 KHz แทน (ปกติจะนับที่ความถี่ 1 Hz) เมื่อ TEST = 1 (TEST MODE) บิต STOP และ บิต REST จะต้องไม่ถูกเซ็ท ในขณะที่อยู่ใน TEST MODE (TEST = 1) ถ้า HOLD = 1 วงจรมันภายในจะถูกหยุดไว้ แต่เมื่อ HOLD กลับมาเป็น "0" จะไม่รับรองว่าเวลาที่ได้จะถูกตั้ง

### 2.3 LIQUID CRYSTAL DISPLAY

LCD เป็นชื่อย่อของ Liquid Crystal Display หรือที่เรียกกันอย่างไม่เป็นทางการว่าตัวแสดงผลแบบผลึกเหลว

ผลึกเหลวเป็นสารที่รวมกันได้อย่างได้สัดส่วนระหว่างของเหลวกับผลึก จุดหลอมเหลวของสารชนิดนี้อยู่ในช่วงอุณหภูมิที่เรียกว่า เมโซเฟส (mesophase) ซึ่งโมเลกุลของสารสามารถเคลื่อนที่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

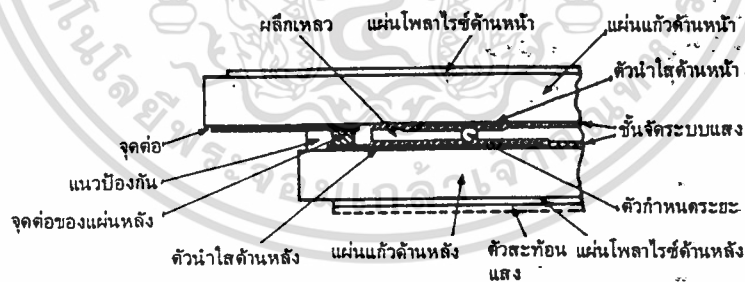
เหมือนดังของเหลว แต่สารชนิดนี้ถูกจัดอยู่ประเภทเดียวกับผลึกที่เป็นของแข็งทั่วไป

ในปี พ.ศ. 2513 ได้มีการค้นพบว่าแผ่นสารผลึกเหลวสามารถเปลี่ยนตัวเองจากใสกลายเป็นทึบแสง หรือจากทึบแสงกลายเป็นใสได้โดยการป้อนแรงดันเข้าไป คุณสมบัตินี้ก็คือหลักการพื้นฐานของ LCD ในปัจจุบันนั่นเอง

### 2.3.1 โครงสร้างของ LCD

ประกอบด้วยแผ่นแก้วสองแผ่นประกบกัน โดยเว้นช่องกลางไว้ประมาณ 6 ถึง 10 ไมโครเมตร ผิวด้านในของแผ่นแก้วเคลือบด้วยตัวนำไฟฟ้าชนิดไอทีโอไว้แสดงตัวอักษร สัญลักษณ์หรือเครื่องหมายต่าง ๆ มักทำมาจากสารอินเดียมทินออกไซด์ (Indium/Tin Oxide:ITO) ระหว่างตัวนำไฟฟ้าชนิดใสกับผลึกเหลวจะมีชั้นสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางของแสงที่ส่องมาชั้นสารนี้จึงเป็นที่รู้จักกันในนามของชั้นที่หักเหเข้าหาแสงหรือชั้นจัดระบบรับแสง (alignment layer) ระยะห่างระหว่างแผ่นทั้งสองถูกกำหนดโดยตัวจัดระยะ (สังเกตดูรูปที่ 2.7 จะเห็นเป็นวงกลมอยู่ระหว่างชั้น)

ชนิดของผลึกเหลวที่ใช้ โดยทั่วไปคือแบบนีเมติก (nematic) แสดงดังรูปที่ 2.8 ก โมเลกุลของผลึกเหลวแบบนีเมติกจะวางขนานกันไปเป็นแนวตรงคล้ายเส้นลวดยาว ถ้าหากวางกลับทิศจะทำให้คุณสมบัติของมันเปลี่ยนไป คริสตอลเหลวที่สามารถแสดงเครื่องหมายแสดงต่าง ๆ ได้คือแบบโคเรสเตอริก (cholesteric) รูปที่ 2.8 ข. และแบบสเมกติก (smectic) รูปที่ 2.8 ค.

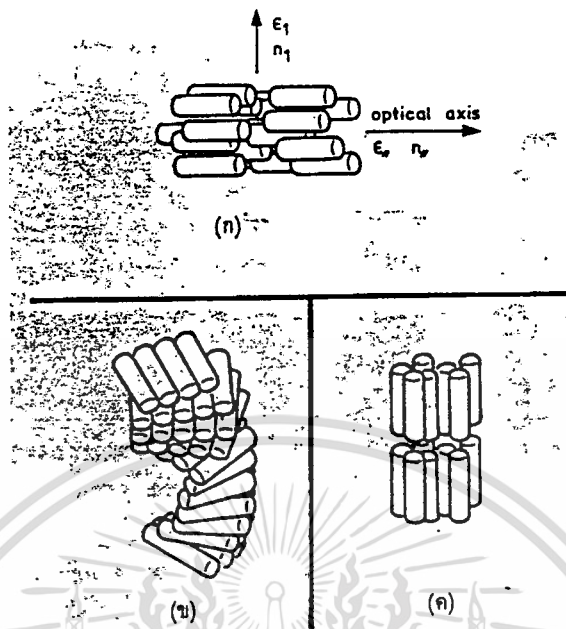


รูปที่ 2.7 โครงสร้างของ LCD

### 2.3.2 LCD แบบนีเมติกชนิดเกลียว

การทำงานเบื้องต้นของ LCD แบบนีเมติกชนิดเกลียว (Twisted Nematic:TN) แสดงในรูปที่ 2.9 โมเลกุลของผลึกเหลวนีเมติกจะถูกจัดไว้ในตำแหน่งโดยชั้นจัดระบบรับแสง

ตำแหน่งของการรับแสงระหว่างแผ่นล่างกับแผ่นบนจะต่างกัน 90 องศา โมเลกุลของผลึกเหลวจะบิดเป็นเกลียวต่างกัน 90 องศา เมื่อแสงจากด้านล่างผ่านโมเลกุลของผลึกเหลวที่บิดเป็นเกลียว เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงโครงสร้างของโมเลกุลของผลึกเหลวทั้ง 3 แบบ (ก) นีเมติก (ข) โครเรสเทอริก (ค) สเมกติก

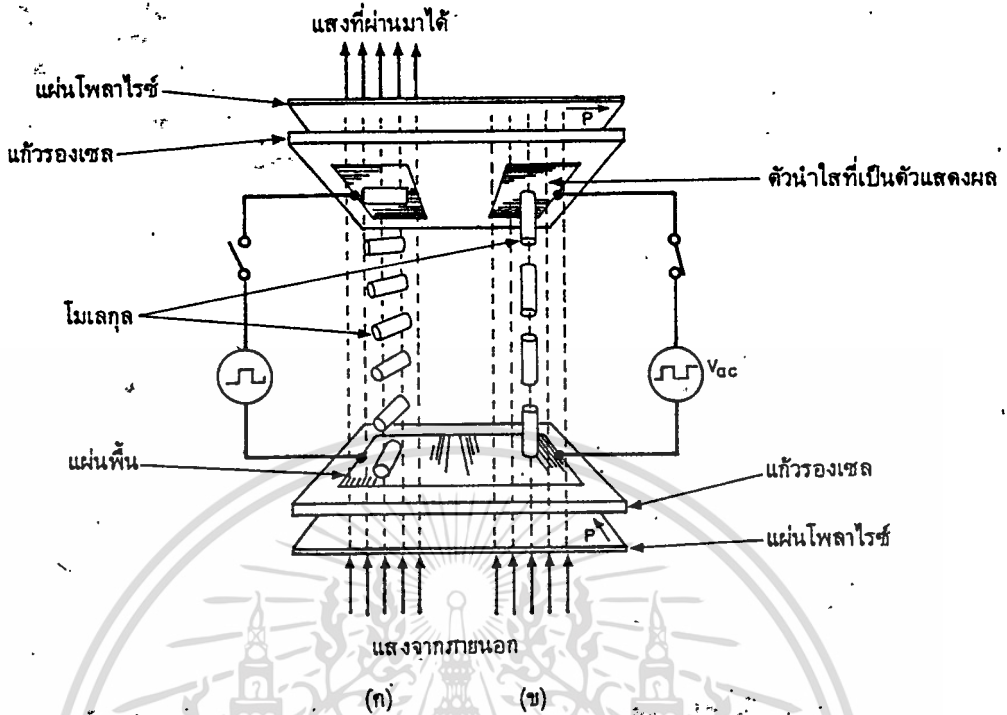
ขึ้นไปยังแผ่นบนทำให้ตำแหน่งของแสงหมุนไป 90 องศา ด้วยปรากฏการณ์นี้เกิดจากคุณสมบัติแอนนิโซโทรปี ทางแสงของโมเลกุลทำให้แสงลอดผ่านแผ่นโพลาริซ์ออกไปได้ แสดงดังรูปที่ 2.9 ก.

แอนนิโซโทรปี (anisotropy) หมายถึงค่าคงตัวของไดอิเล็กตริก (permittivity:  $E$ ) ของผลึกเหลวเปลี่ยนค่าไปตามตำแหน่งต่าง ๆ)

ถ้ามีแรงดันป้อนให้ที่ตัวนำไฟฟ้าชนิดใส คุณสมบัติแอนนิโซโทรปีทางไฟฟ้าของโมเลกุลจะทำให้การวางตัวของโมเลกุลเปลี่ยนไป (ดูรูปที่ 2.9 ข.) แสงจะผ่านผลึกเหลวด้วยทิศทางคงเดิม จึงไม่สามารถผ่านแผ่นโพลาริซ์ด้านบนออกไปได้

เมื่อเราป้อนแรงดัน โมเลกุลก็จะเรียงตัวดังเดิมอีกครั้งหนึ่ง (รูปที่ 2.9 ก.) แสงก็สามารถทะลุผ่านไปได้อีก การทำงานของ LCD แบบนี้เรียกว่า การแสดงผลเชิงบวก (positive image display) แสดงดังรูปที่ 2.10

หากเราวางแผ่นโพลาริซ์ให้หมุนไปมากกว่า 90 องศา ผลึกจะเกิดตรงกันข้ามคือ LCD จะมืดทึบเมื่อไม่มีแรงดันป้อน แต่จะเป็นแสงใสเมื่อป้อนแรงดัน เช่นนี้เรียกว่า การแสดงผลเชิงลบ (negative image display)



รูปที่ 2.9 การทำงานพื้นฐานของตัวแสดงผลแบบนิเมติกชนิดเกลียว (ก) ขณะไม่ทำงาน (ข) ขณะทำงาน

รูปที่ 2.11 เป็นการแสดงผลด้วยตัวเลข 7 ส่วน โดยการจ่ายแรงดันให้กับส่วนต่าง ๆ การแสดงผลจะสมบูรณ์ได้ก็ต่อเมื่อจ่ายแรงดันเลือกให้ทั้งที่เป็นส่วนแสดงผลและส่วนที่จัดรวมประกอบกัน

### 2.3.3 แบบต่าง ๆ ของการแสดงผล

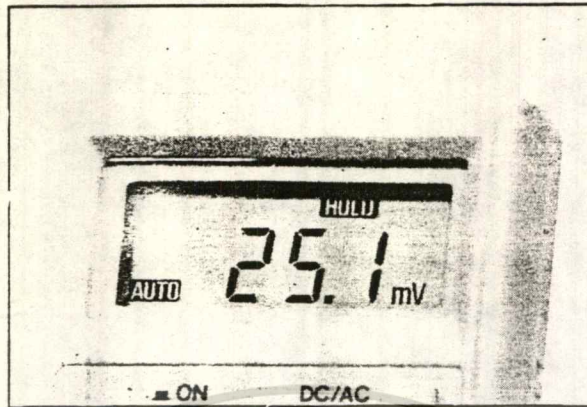
LCD สามารถแสดงผลให้เราเห็นได้โดยมีหลักการ 3 แบบ จะเลือกใช้แบบใดก็ขึ้นอยู่กับแสงสว่างโดยรอบ

1. แบบสะท้อน (reflective mode) จะมีสารประเภทโลหะเคลือบอยู่ที่แผ่นหลังของ LCD เช่น เคลือบด้วยอลูมิเนียมพอเพียงจะทำกรสะท้อนแสงจากภายนอกผ่านตัวแสดงผลไปยังตาของเรา (ดูรูปที่ 2.12 ข.) แบบนี้จะเหมาะกับที่มีแสงสว่างเพียงพอ ข้อดีคือ ไม่ต้องการแหล่งจ่ายแรงดันป้อนให้กับหลอดไฟใด ๆ อีก

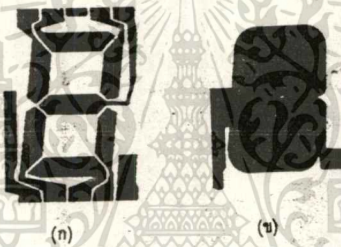
2. แบบส่งผ่าน (transmissive mode) มักใช้กับ LCD ที่มีการแสดงผลเชิงลบ (ซึ่งสภาวะการทำงานของตัวแสดงผลเชิงจะโปร่งใสในขณะที่มีพื้นเป็นสีทึบ) แบบนี้จะวางหลอดไฟไว้ด้านหลังทำให้อ่านค่าแสดงผลได้ชัดเจน แสดงดังรูปที่ 2.12 ก.

3. แบบส่งผ่าน/สะท้อน (transflective mode) เป็นการรวมระหว่าง 2 แบบที่กล่าวมาแล้ว ตัวแสดงผลอ่านได้จากการสะท้อนของแสงจากภายนอก และมีแสงส่องสว่างจากด้านหลังเมื่อต้องการอ่านค่าต่าง ๆ ในที่มืด แสดงดังรูปที่ 2.12 ค.

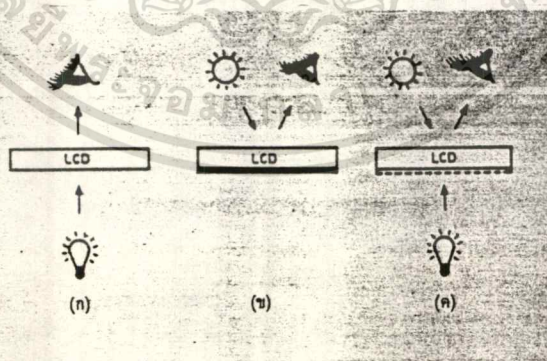
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 การแสดงผลเชิงบวก



รูปที่ 2.11 ภาคแสดงผล 7 ส่วนแบบมีจุดทศนิยม (ก) ส่วนแสดงผล (ข) ส่วนจุดรวม



รูปที่ 2.12 แบบต่างๆของการแสดงผลของ LCD

- (ก) แบบส่องผ่าน
- (ข) แบบสะท้อน
- (ค) แบบส่องผ่าน / สะท้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะวิธีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.4 สีสิ้นของ LCD แบบ TN

สีของ LCD แบบนี้เมติกชนิดเกลียว (TN) ได้มาจาก 3 สิ่ง คือ ตัวกระจายสี, ตัวกรองสี และแสงสีจากด้านหลัง ตัวกระจายสีจะให้ส่วนแสดงผลเป็นสีต่าง ๆ บนแผ่นพื้นไว้สีหรือตัวอักษรไว้สีบนแผ่นพื้นสีก็ได้ โดยการใช้แผ่นโพลาไรซ์ 2 สี เช่น สีแดงและเขียว อาจให้ตัวแสดงผลสีแดงอยู่บนพื้นสีเขียวหรือตัวแสดงผลสีเขียวอยู่บนพื้นสีแดงก็ได้ ตัวกรองสีจะแผ่นพอลียูรีเทนด้านหลังหรือเป็นสีขุ่น ๆ บนตัวแสดงผลก็ได้ มักใช้ใน LCD แบบส่งผ่าน เช่น ตัวอักษรสีแดงบนพื้นทึบแสง แสงสีจากด้านหลังทำให้ตัวแสดงผลเป็นสีดำบนแผ่นพื้นสีต่าง ๆ หรือตัวแสดงผลต่าง ๆ บนแผ่นพื้นสีดำ หากจะใช้ในแบบส่งผ่าน/สะท้อนควรคำนึงถึงด้วยว่าแสงจากด้านหลังจะถูกลดทอนลงอย่างมากโดยแสงจากภายนอก

### 2.3.5 คุณสมบัติทางแสง

ความชัดเจนในการอ่านขึ้นอยู่กับตัวแปรสำคัญคือ ความสว่าง (brightness) และความเข้มของแสง (contrast) ของ LCD ด้วย ความเข้มของแสงทำได้จากความสว่างของบริเวณไว้สีหารด้วย ความสว่างของบริเวณที่มีสีทึบ ใน LCD แบบ TN อัตราส่วนความเข้มของแสงจะอยู่ในย่าน 5 ถึง 50 แต่อัตราส่วนความเข้มสูงสุดที่ตามนุษย์สามารถจับได้มีค่าประมาณ 10 และความชัดเจนอย่างน้อยที่สุดประมาณ 2

ในตัวแสดงผลแบบที่มีแสงส่องจากด้านหลังต้องมีอัตราส่วนความเข้มของแสงสูงเป็นพิเศษเพราะเมื่อมองผ่านแสงที่สวนขึ้นมาตาของเราจะจับความเข้มได้น้อยลง ทั้งความสว่างและความเข้มขึ้นอยู่กับชนิดของแผ่นโพลาไรซ์ สำหรับตัวแสดงผลแบบสะท้อนเชิงบวกแผ่นโพลาไรซ์ประสิทธิภาพต่ำจะให้ความสว่างมากแต่ความเข้มน้อย ส่วนแผ่นโพลาไรซ์ประสิทธิภาพสูงจะให้ความเข้มสูงแต่ความสว่างลดลง

### 2.3.6 มุมมอง

รูปที่ 2.13 แสดงความสัมพันธ์ระหว่างความเข้มของแสงกับแรงดันของมุมมอง 3 มุมที่แรงดันต่ำมาก ๆ LCD ไม่สามารถแสดงผลให้เห็นชัดเจนได้ เมื่อ  $\theta$  คือมุมในแนวระดับของ LCD และ  $\alpha$  คือมุมที่อ้างอิงกับเส้นตั้งฉาก

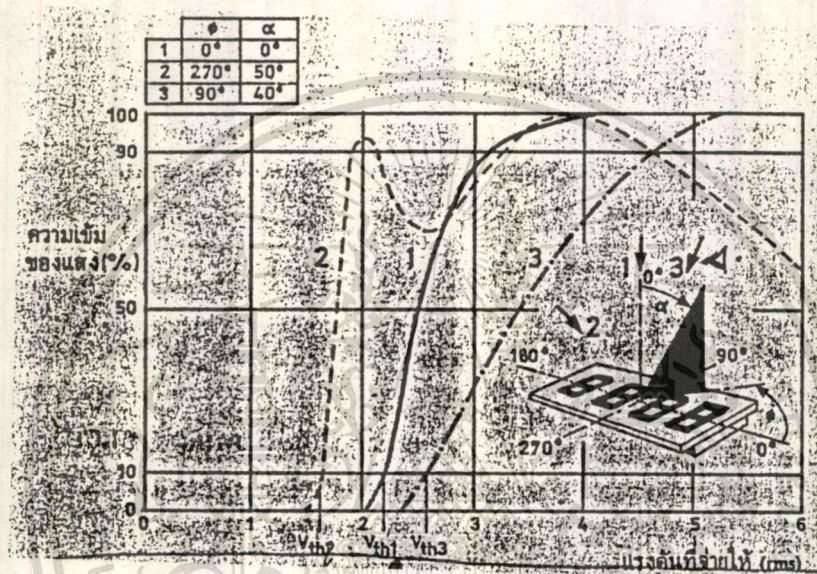
เมื่อแรงดันเพิ่มขึ้นมุมที่สามารถมองเห็นก็จะสูงตาม(มุมเงยต่ำ) ดังเส้นกราฟเส้นที่สอง หากแรงดันสูงขึ้นมุมเงยก็เพิ่มขึ้นตามลำดับ

แรงดันที่น้อยที่สุดที่พอจะทำให้มองเห็นได้ชัดเจน ( 10 เปอร์เซ็นต์ของแรงดันสูงสุด) เรียกว่า แรงดันเทอร์ชโฮล (threshold vottage:  $V_{th}$ ) แรงดันที่มีค่าถึง 90 เปอร์เซ็นต์ของแรงดันสูงสุด เรียกว่า แรงดันอิ่มตัว (saturtion vottage :  $v_{sat}$ )

ความสัมพันธ์ระหว่างแรงดันกับความเข้มของแสงจะเปลี่ยนไปตามส่วนผสมที่แตกต่างกันของผลึกเหลว อัตราส่วนผสมส่วนใหญ่มักมีสัมประสิทธิ์ของอุณหภูมิเป็นลบ ซึ่ง  $V_{th}$  จะลดลงเมื่ออุณหภูมิเพิ่มสูงขึ้น

### 2.3.7 ความเร็วของการแสดงผล

เวลาตอบสนองการทำงานอยู่ในช่วง 50 ถึง 100 มิลลิวินาที (ในอุณหภูมิห้อง 25 องศา) สิ่งสำคัญที่มีผลต่อช่วงเวลาตอบสนองคือความหนืดของผลึกเหลวซึ่งความหนืด (viscosity) ของมันจะเพิ่มขึ้นเมื่ออุณหภูมิลดลง เพราะโมเลกุลมีอิสระในการเคลื่อนที่น้อยลงทำให้การตอบสนองช้าลง เวลาตอบสนองยังขึ้นอยู่กับขนาดของแรงดันที่จ่ายให้วิธีป้อนแรงดันและความหนาของแผ่นผลึกเหลวด้วย



รูปที่ 2.13 ความสัมพันธ์ระหว่างความเข้มของแสงกับแรงดัน

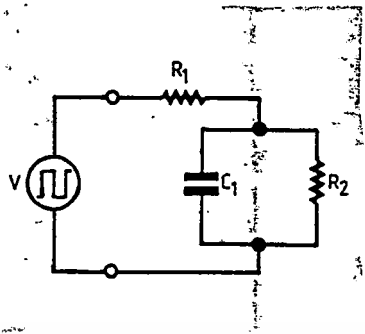
### 2.3.8 การขับ LCD

LCD แต่ละส่วน (เซกเมนต์) สามารถเขียนวงจรสมมูลย์ได้ดังรูปที่ 9 ประกอบด้วยตัวต้านทาน  $R_1$  ค่าต่ำต่ออนุกรมกับตัวเก็บประจุ  $C_1$  ซึ่งต่อขนานกับตัวต้านทาน  $R_2$  (มีค่าสูงกว่า  $R_1$ ) ค่าของตัวเก็บประจุจะขึ้นอยู่กับแรงดันที่ปรากฏคร่อม LCD ความถี่ของสัญญาณที่ป้อนเข้าไปต้องไม่ต่ำกว่า 30 เฮิรตซ์ เพื่อป้องกันการกระพริบของตัวแสดงผล กระแสที่ใช้กันทั่วไปคือ 1.5 ไมโครแอมป์ต่อตารางเซนติเมตรของแผ่น LCD คือจำนวนจุดรวมของแผ่นหลังหรือจำนวนของส่วนแสดงผลในแต่ละกลุ่ม

### 2.4 มาตรฐาน RS 232C

โดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นอนุกรม เรียกชื่อกันว่า RS 232C อยู่ในตัวเองอยู่แล้ว หลายเครื่องไม่มีมากับเครื่องอย่างเช่น IBM PC จำเป็นจะต้องมีการ์ดที่เรียกว่าอะซิงโครนัสอะแดปเตอร์ (Asynchronous Communication Adapter) มาเสียบใส่พอร์ต RS 232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 วงจรสมมูลของ LCD  $R_1$  เป็นความต้านทานอนุกรมที่เกิดขึ้นจากตัวนำชนิดใส  $R_2$  และ  $C_1$  เป็นค่าที่เกิดจากผลึกเหลวที่อยู่ระหว่างตัวนำชนิดใส

ทำหน้าที่รับและส่งข้อมูลในแบบอนุกรมเรียกว่า Universal Asynchronous Adapter เหตุที่มีชื่อเรียกว่า RS 232C ก็เนื่องจากสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของอเมริกาหรือ EAI ได้กำหนดมาตรฐานของอุปกรณ์ การสื่อสารแบบอนุกรมเอาไว้ภายใต้ชื่อว่า RS 232C ความจริงมาตรฐานของการส่งข้อมูลแบบอนุกรมมีหลายมาตรฐานแต่ที่นิยมกันมากที่สุดสำหรับไมโครคอมพิวเตอร์ก็คือ RS 232C

#### หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัสก็คือ

##### รับสัญญาณ

1. เปลี่ยนสัญญาณเข้ามาแบบอนุกรมให้เป็นแบบขนาน
2. ตรวจสอบความผิดพลาดของสัญญาณที่รับ
3. ตัดสตอปบิตและพาริตีบิตออก
4. ส่งสัญญาณให้ซีพียูรู้ว่ารับสัญญาณไว้แล้ว

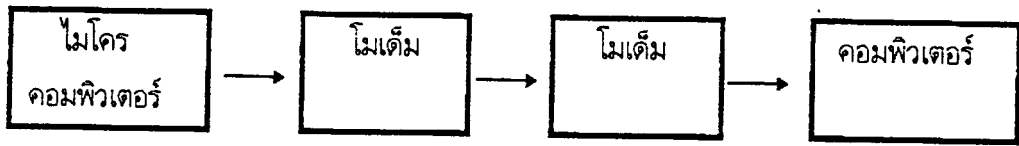
##### ส่งสัญญาณ

1. เปลี่ยนสัญญาณแบบขนานจากซีพียูค่อยทยอยส่งออกเป็นแบบอนุกรม
2. เพิ่มสตอปบิตและพาริตี
3. เพิ่มสัญญาณควบคุมโมเด็มที่ต่อเชื่อม(ถ้ามี)

มาตรฐาน RS 232C ได้จัดพิมพ์ขึ้นเมื่อ ปี ค.ศ. 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งสหรัฐอเมริกา RS ย่อมาจาก Recommended Standard ส่วน 232 เป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ C เป็นหมายเลขของฉบับท้ายสุดของมาตรฐานตัวนี้ จุดประสงค์ของมาตรฐานตัวนี้ก็เพื่อบรรยายคุณลักษณะของการการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment DTE) กับอุปกรณ์สื่อสารข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Data Communication Equipment DCE) สำหรับผู้ใช้ไมโครคอมพิวเตอร์ DTE ก็หมายถึงตัวไมโครคอมพิวเตอร์และ DCE ก็หมายถึง โมเด็ม อุปกรณ์อื่นๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบอนุกรมอาจจะเป็นได้ทั้ง DTE และ DCE



รูปที่ 2.15 การใช้ RS 232C เชื่อมต่ออุปกรณ์

ขึ้นอยู่กับผู้ผลิต ข้อแตกต่างของ DTE และ DCE จะเห็นได้จาก รูปที่ 2.15 จากรูปนี้เราจะเห็นได้ว่า RS 232C มีส่วนสำคัญอย่างใหญ่หลวงสำหรับการสื่อสารข้อมูลระหว่างไมโครคอมพิวเตอร์ ความจริงอีกประการหนึ่งของ RS 232C ก็คือ ความเร็วและระยะทางของการเชื่อมต่อ RS 232C สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้จาก 0-20,000 บิตต่อวินาที ซึ่งเพียงพอสำหรับไมโครคอมพิวเตอร์ที่มีขนาดอัตราบอด 110 ถึง 9600 บอด ความยาวของสายเชื่อมต่อโดยสัญญาณตามมาตรฐานของ RS 232 จำกัดอยู่แค่ 50 ฟุตซึ่งเพียงพอสำหรับการสื่อสารไมโครคอมพิวเตอร์กับอุปกรณ์รอบนอก

#### 2.4.1 ลักษณะของสัญญาณ RS 232C

เพื่อเป็นหลักประกันว่าข้อมูลถูกส่งออกไปอย่างถูกต้องและอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นจะต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้ มาตรฐาน RS 232C กำหนดย่านของแรงดันไฟฟ้าในสัญญาณเพื่อสนองจุดประสงค์ข้างบน ดังแสดงในตารางที่ 2.10 และรูป 2.16

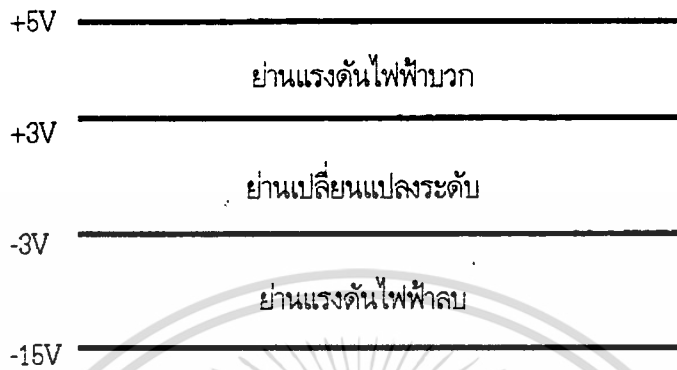
สำหรับเครื่องไมโครคอมพิวเตอร์บางเครื่อง ใช้แต่สัญญาณลอจิกออกมาเป็นสัญญาณของ RS 232C เลย อย่างเช่น อะซิงโครนัสอะแดปเตอร์ของ IBM PC ในกรณีเช่นนี้ระยะทางของสายที่เชื่อมต่ออาจจะไปได้สั้นกว่า 50 ฟุต ดังที่กล่าวเอาไว้เนื่องจากระดับของกราวด์เปลี่ยนแปลงไป อันเนื่องจากการสูญเสียไปในความต้านทานของสาย ผู้ที่เคยใช้ IBM PC อาจจะเคยประสบปัญหานี้มาแล้วว่า

มาตรฐานของการใช้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานะลอจิก	สถานะของสัญญาณ	ฟังก์ชันในการควบคุม
บวก	0	สเปซ	ออน
ลบ	1	มาร์ค	ออฟ

ตารางที่ 2.14 แสดงการกำหนดมาตรฐานแรงดันไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำไม่ต่อสัญญาณ RS 232C เกินกว่า 10 ฟุต แล้วใช้งานไม่ได้ แต่อย่างไรก็ตาม RS 232C ของ IBM PC ยังมีโอกาสให้เลือกใช้ 20 มิลลิแอมแปร์ กระแสกลับแรงดันไฟฟ้า



รูปที่ 2.16 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS 232C

#### 2.4.2 การกำหนดจุดต่อของ RS 232C

ในทางฟิสิกส์แล้ว มาตรฐานของ RS 232C กำหนดข้อต่อแบบ DB-25 แต่ละขาของข้อต่อกำหนดไว้ดังรูปที่ 2.17

อาจจะใช้ข้อต่อชนิดอื่นที่นอกเหนือไปจาก DB-25 ยกตัวอย่างเช่น Fujitsu F-8 , IBM AT , IBM Jr เป็นกันตัวเมียของข้อต่อควรอยู่ที่ตัวโมเด็ม ขณะที่ตัวผู้ควรอยู่ที่ Asynchronous Communication Adapter หรือที่ตัวไมโครคอมพิวเตอร์เอง อย่างไรก็ตามผู้ผลิตหลายรายไม่ได้ทำตามกฎเกณฑ์ที่ว่านี้

สัญญาณต่างๆ ถูกมอบหมายให้ทำหน้าที่ดังนี้

##### Transmit Data (TD ขาที่ 2)

เป็นสัญญาณที่ส่งออกจาก DTE (หรือตัวไมโครคอมพิวเตอร์) ไปยังโมเด็มหรือต่อเข้าโดยตรงกับไมโครคอมพิวเตอร์ตัวอื่น หรือเครื่องพิมพ์ เมื่อไม่มีสัญญาณส่งออกสถานะภาพของลอจิกที่ขา นี้จะมีค่าเท่ากับ "1" หรือเทียบเท่ากับสตีอปปิต

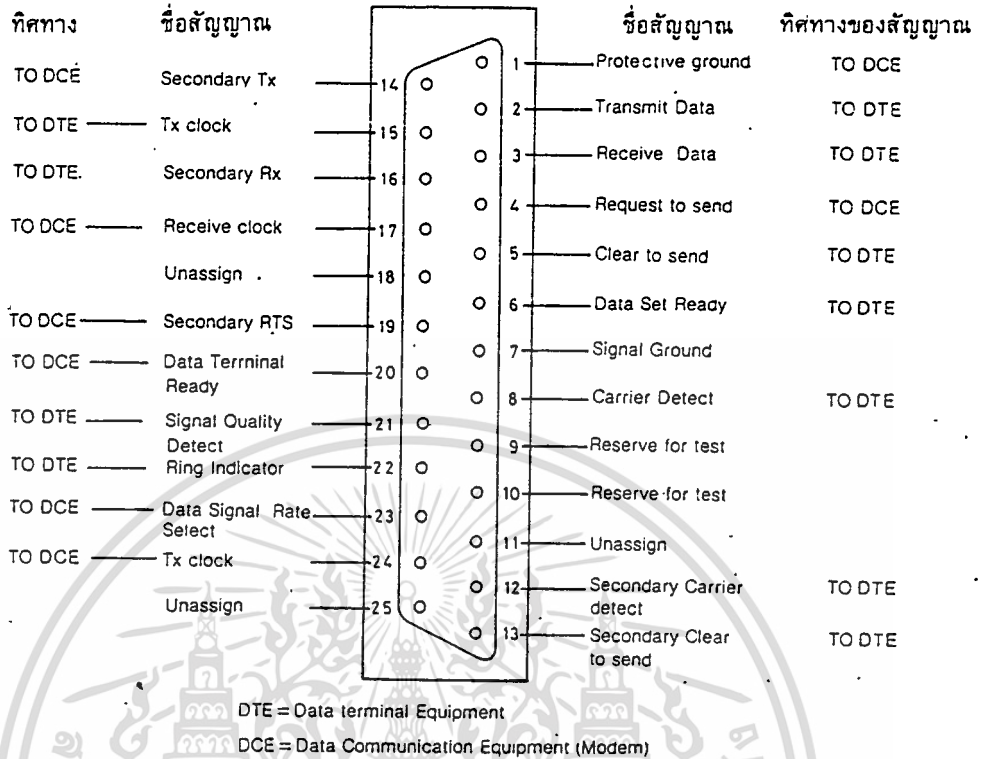
##### Receive Data (RD ขาที่ 3)

เป็นทางของสัญญาณเข้าไปยัง DTE หรือไมโครคอมพิวเตอร์เมื่อไม่มีสัญญาณรับเข้ามา ขานี้จะมีสถานะภาพทางลอจิกเป็น "1"

##### Request To Send (RTS ขาที่ 4)

ใช้สำหรับส่งสัญญาณไปยังโมเด็มหรือเครื่องพิมพ์เป็นการเรียกร้องที่จะส่งสัญญาณมาทางขา 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การกำหนดของหัวต่อ RS-232 แบบ DB-25

สัญญาณนี้ใช้คู่กับ CTS หรือ Clear to send อุปกรณ์รับหากได้รับสัญญาณ RTS จะตรวจสอบตัวเองว่าพร้อมจะรับสัญญาณได้หรือยัง หากพร้อมที่จะรับก็ส่งสัญญาณออกไปที่สาย CTS

#### Clear To Send (CTS ขาที่ 5)

ดังอธิบายไว้ใน RTS เมื่อสัญญาณนี้อยู่ในสถานะออฟ(negative voltage หรือลอจิก "1") หมายความว่า อุปกรณ์รับกำลังบอกว่าพร้อมที่จะรับข้อมูลแล้ว

#### Data Set Ready (DSR ขาที่ 6)

เมื่อสัญญาณสายนี้อยู่ในสถานะออน (หรือลอจิก 0) เป็นการบอกไมโครคอมพิวเตอร์ หรือฝ่ายส่งว่า โมเด็มต่อเข้ากับสายโทรศัพท์เรียบร้อยแล้วและพร้อมที่จะส่งได้แล้ว

#### Signal Ground (SG ขาที่ 7)

SG ทำหน้าที่เป็นระดับแรงดันอ้างอิงสำหรับทุกๆสายของสัญญาณ จะมีแรงดันเป็น "0" เมื่อเทียบกับสัญญาณตัวอื่น

#### Carrier Detect (CD ขาที่ 8)

โมเด็มจะส่งสัญญาณที่อยู่ในสถานะออน (ลอจิก "0") ไปบอกไมโครคอมพิวเตอร์ เมื่อได้รับสัญญาณจากโมเด็มของอีกฝ่ายหนึ่ง สัญญาณนี้จะนำไปจุด LED บอกว่าได้รับสัญญาณจากโมเด็มอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฝ่ายหนึ่งแล้วไฟ LED จะอยู่บนหน้าบัตรของโมเด็มเอง

#### Data Terminal Ready (DTR ขาที่ 20)

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

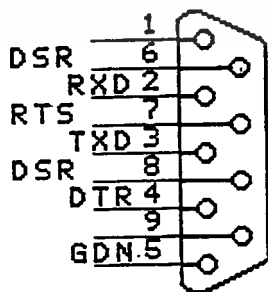
#### Ring Indicator (RI ขาที่ 22)

สัญญาณนี้ใช้ในโมเด็มที่เป็นระบบตอบได้อัตโนมัติ (Auto-answer) สัญญาณนี้จะออเมื่อมีสัญญาณกระดิ่งมาและออระหว่างเสียงดังของกระดิ่ง

บางทีเราอาจจะสับสนระหว่างสถานะภาพของลอจิกกับสถานะภาพของสัญญาณ โดยปกติเราจะคุ้นเคยอยู่กับความรู้สึกที่ว่า เมื่อแรงดันเป็นบวก หรือสัญญาณอนลอจิกน่าจะเป็น "1" สำหรับสัญญาณต่างๆที่กล่าวมานี้จะมีลักษณะตรงกันข้าม เพราะแต่เดิมนั้นการติดต่อกันทางโทรเลขการทำงานของสัญญาณจะต้องครบวงจรทั้งฝ่ายส่งและฝ่ายรับ เมื่อลอจิกเป็น "0" หรือขณะที่ไม่มีอะไรส่งควรรจะมีสัญญาณทางไฟฟ้าครบวงจรอยู่ตลอดเวลา จะได้ว่าว่างจรไม่ขาดระหว่างทางตรงไหน ควรรู้ว่าว่างจรครบอยู่ตลอดเวลา ก็โดยการให้ค่าแรงดันที่ฝ่ายส่ง ดังนั้นจึงถือกันว่าสัญญาณไฟบวกใช้เป็นลอจิก "0"

### 2.4.3 มาตรฐาน RS 232C กับ V.24

ได้กล่าวถึงสัญญาณตามสมาคมผู้ผลิตอุตสาหกรรมอิเล็กทรอนิกส์ ของสหรัฐอเมริกาหรือ RS 232C ไปแล้ว สหประชาชาติและกลุ่มของ CCITT (Comite Consultatif International Telephonique Telegraphique) ได้ออกมาตรฐานมาเหมือนกัน และก็หลายฉบับตั้งแต่การประชุมครั้งที่สองที่ กรุงนิวเดลีปี ค.ศ.1960 ออกมาเป็นสมุดปกแดง ครั้งที่สามปี ค.ศ.1964 ที่กรุงเจนีวา ออกมาเป็นสมุดปกสีน้ำเงิน จนกระทั่งครั้งที่ 6 เมื่อ ปี ค.ศ.1977 ที่กรุงเจนีวา อีกเหมือนกัน ออกมาเป็นสมุดปกสีส้ม ได้เป็นมาตรฐานออกมา 3 รูปแบบคือ



รูปที่ 2.18 แสดงหัวต่อ RS-232 แบบ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V.24 บรรยายถึงการเชื่อมต่ออุปกรณ์รับส่งข้อมูล (DTE) กับอุปกรณ์รับส่งข้อมูลปลายทาง

V.28 บรรยายถึงลักษณะทางไฟฟ้าสำหรับการใช้ Unbalance Double Current Interchange Circuit

Driver output logic levels with 3k to 7k load	$15V > 0_h > 5V$ $-5 > 0_l > -15$ โวลต์
Driver output voltage when open circuit	$V_o < 25$ โวลต์
Driver output impedance with Power off	$R_o > 300$ Ohms
Output short circuit current	$I_o < 0.5$ A
Driver slew rate	$dv/dt < 30$ V/s
Receiver input impedance	$7k > R_{in} > 3k$
Receiver input voltage	+15 compatible with driver
Receiver output with open circuit input	MARK
Receiver output with +3V input	SPACE
Receiver output with -3V input	MARK
+15	LOGIC 0 = SPACE =
+5	CONTROL ON
+5	Noise Margin
+3	
+3	Transition Region
-3	
-3	Noise Margin
-5	
-5	LOGIC 1 = MARK =
-15	CONTROL OFF

### ตารางที่ 2.15 คุณลักษณะโดยย่อของสัญญาณ RS 232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบและการสร้าง

ระบบของชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180 จะประกอบด้วยแผงวงจรหลายส่วนซึ่งทำหน้าที่ต่าง ๆ กัน ดังรูปที่ 3.1 และคำอธิบายประกอบการทำงานของระบบ

- DMA คอนโทรลเลอร์ ขนาด 2 CHANNEL

- พอร์ตสื่อสารอนุกรม ASYNCHRONOUS SERIAL COMMUNICATION (ASCI) เป็นอิสระต่อกัน ขนาด 2 CHANNEL แบบฟูลดูเพล็กซ์ซึ่งสามารถใช้งานได้กับพอร์ตสื่อสารอนุกรมแบบ UNIVERSAL ASYNCHRONOUS RECEIVER/TRANSMITTER (UART) ซึ่งภายในจะประกอบด้วย Z8440 SIO และ Z8530 SCC นอกจากนี้แล้ว พอร์ตนี้ที่ CHANNEL 0 จะมีขา DCD0,CTS0 และ RTS0 ไว้สำหรับติดต่อกับโมเด็มได้อีกด้วย

- มี 16 บิต PROGRAMMABLE RELOAD TIMERS ขนาด 2 CHANNEL

- มี MMU เพื่อจัดการกับหน่วยความจำขนาด 1 MB ได้

ชุดฝึกนี้สามารถรับส่งข้อมูลกับ PC ได้โดยใช้ REMOTE MONITOR ที่อยู่ใน 32 KB (LOW) ของ EPROM 27512 ที่อยู่บน MEMORY BOARD 1 MB และมีการติดต่อกับผู้ใช้โดยผ่านจอ LCD และมีคีย์บอร์ดขนาด 32 คีย์ เพื่อรับคำสั่งที่เป็นภาษาเอสเอ็มบลีได้โดยตรง โดยผู้ใช้ไม่ต้องเสียเวลาในการตาราง OPCODE นอกจากนี้ยังมี A/D ขนาด 1 ช่อง และ D/A ขนาด 1 ช่อง เพื่อให้ผู้ใช้ชุดฝึกสามารถกระทำกับสัญญาณอนาล็อกได้นำนามาประยุกต์เป็นชุดฝึกการประมวลผลสัญญาณดิจิทัล เพราะใน Z80180 มีคำสั่งคูณ (MLT) ให้ใช้งานเพื่อสะดวกในการทำ CONVOLUTION กับสัญญาณอนาล็อก ที่เหลือก็จะเป็น USER PORT 8 บิตเพื่อให้ผู้ใช้สามารถเชื่อมต่อชุดฝึกกับอุปกรณ์ภายนอกได้ ซึ่งสามารถนำชุดฝึกนี้ไปใช้ประโยชน์ในการเรียนการสอนได้เป็นอย่างดี

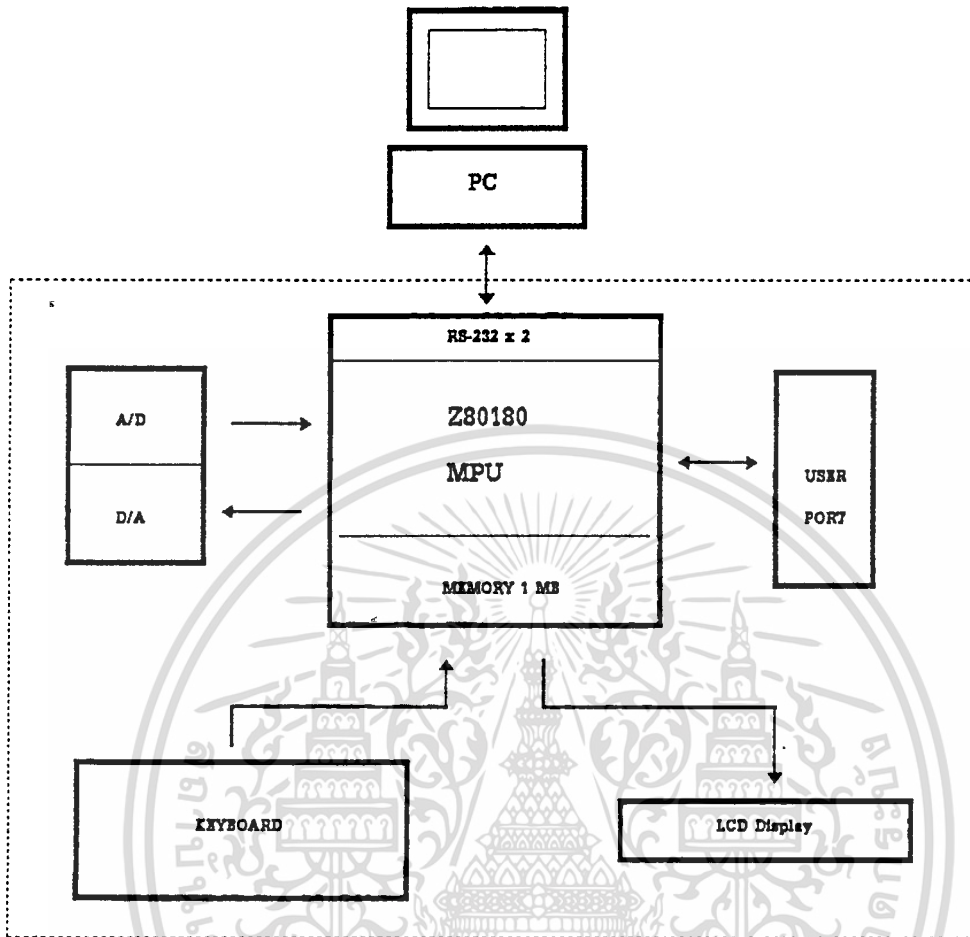
#### 3.1 การออกแบบ การสร้าง และการทำงานแต่ละส่วน

ขั้นตอนการออกแบบจะแบ่งเป็น 2 ส่วนคือ ทางด้านฮาร์ดแวร์ (Hardware) และทางด้านซอฟต์แวร์ (Software)

##### 3.1.2 ด้านฮาร์ดแวร์ ประกอบด้วย

- หน่วยความจำ 1 MB
- ADC 1 ช่อง และ DAC 1 ช่อง
- ชุดแสดงผลเพื่อติดต่อกับผู้ใช้โดยใช้จอ LCD
- RTC เบอร์ 6242

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 บล็อกไดอะแกรมแสดงระบบการทำงานของชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180

- คีย์บอร์ดขนาด 4x8 (32 คีย์)
- พอร์ตอนุกรม RS-232 2 ช่อง

### 3.2.2 ด้านซอฟต์แวร์ ประกอบด้วย

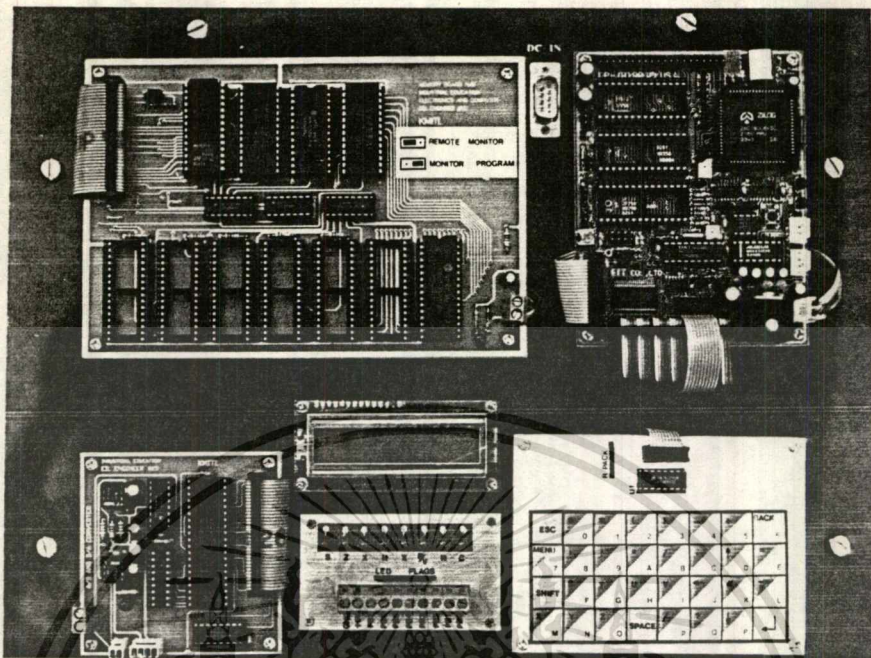
- โปรแกรม REMOTE เปลี่ยนการทำงานให้มาทำงานบน PC
- โปรแกรมมอนิเตอร์ สำหรับควบคุมให้บอร์ดทำงานโดยลำพัง (STAND ALONE)
- โปรแกรม ONE - LINE ASSEMBLER เพื่อรับคำสั่ง ASSEMBLY โดยตรง

### 3.2 รายละเอียดของบอร์ด JR-180

CPU เป็นบอร์ด Z80180 ของ ZILOG ซึ่งเป็น CPU ตระกูล Z80 สามารถใช้คำสั่งของ Z80 ได้ทั้งหมดและยังเพิ่มอีก 12 ชุดคำสั่งใช้งาน เช่น คำสั่งคูณ (MUL), TST g (คำสั่งตรวจสอบบิตในรีจิสเตอร์) เป็นต้น ในบอร์ด JR180 นี้เลือกใช้ Z80180 ความเร็วขนาด 6 เมกะเฮิร์ตซ์ แต่ก็สามารถใช้กับความถี่ 6.144 เมกะเฮิร์ตซ์ได้ด้วย ทำให้การทำงาน 1 คำสั่ง ใช้เวลาเพียง 0.48 ไมโครวินาทีเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180

ปัญหาที่ตามมาก็คือจำเป็นต้องใช้อีพ롬หรือแรมที่มีค่าแอสเซสใหม่ต่างๆเท่านั้น หรือซึ่งในข้อนี้ทาง ZILOG ได้ออกแบบแก้ไขจุดนี้ โดยสามารถตั้ง wait state ภายในตัว CPU ในการอ่านเขียนข้อมูลกับหน่วยความจำและพอร์ตนอกได้

### 3.2.1 การจัดหน่วยความจำบน CP-JR180

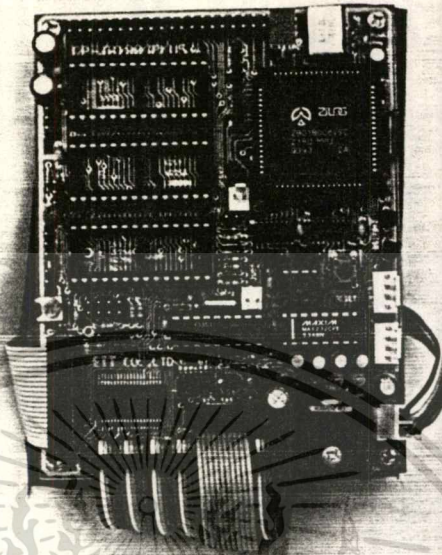
การจัดหน่วยความจำบน CP-JR180 นี้ ถ้าเรานำบอร์ดมาใช้งานในบริบทนิพนธ์เราจะต้องยกเลิกการ DECODE หน่วยความจำบนบอร์ดทั้งหมด โดยที่เราไม่ต้องสนใจ MEMORY MAP บนบอร์ด CP-JR180 และต้องตั้ง JUMPER บนบอร์ดออกให้หมดยกเว้น JUMPER ที่เกี่ยวกับ WATCH DOG ส่วน JUMPER สุดท้ายคือ J7 ซึ่งทำหน้าที่เลือกชนิดของจอ LCD ที่ต่ออยู่กับบอร์ด CP-JR180 ในบริบทนิพนธ์นี้ใช้จอ LCD แบบตัวอักษรจึงต้องเซ็ท JUMPER ไว้ที่ C แต่ถ้าเราต้องการใช้หน่วยความจำบนบอร์ด CP-JR180 ก็ให้ถอดสายแพที่เชื่อม SYSTEM BUS Z80180 ออกจาก MEMORY BOARD 1 MB แล้วใส่ JUMPER กลับคืนให้ตรงตามวัตถุประสงค์ของเราโดยดูคู่มือ CP-JR180 ประกอบ

### 3.2.2 การจัดการเกี่ยวกับพอร์ต

ในส่วนนี้ใช้ไอซี U7 74ALS138 ทำหน้าที่ถอดรหัสแอสแอดเดรสของพอร์ต โดยใช้สัญญาณควบคุม IORQ และ M<sub>1</sub> มาควบคุม เพื่อให้เกิดสัญญาณ CS ขึ้นเฉพาะในการติดต่อกับส่วนพอร์ต

อินพุตเอาท์พุตเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงบอร์ด CP-JR180

**พอร์ต 8255** เป็นพอร์ตอินพุตเอาต์พุตขนาด 8 บิต 3 พอร์ต มีแอดเดรสคือ พอร์ต A อยู่ที่แอดเดรส 80H, พอร์ต B อยู่ที่ 81H, พอร์ต C อยู่ที่ 82H และแอดเดรสของพอร์ตควบคุมอยู่ที่ 83H

**พอร์ตรีลไทม์คล็อก (RTC)** ใช้ไอซีรีลไทม์คล็อก/ปฏิทิน ชนิดซีมอสเบอร์ 6242 โดยต่อขาบั๊สแอดเดรส และข้อมูลขนาด 4 บิต เข้าโดยตรงกับซีพียู ซีพียูรีลไทม์คล็อกตัวนี้จะสามารถบอกได้ทั้งวัน, เดือน, ปี, วันในรอบสัปดาห์, ชั่วโมง, นาที, วินาที และบิอติกส์สุททินได้ด้วย ได้รับการจัดสรรแอดเดรสไว้ตั้งแต่ A0H-BFH

**พอร์ตเชื่อมต่อ LCD** สามารถต่อเข้ากับโมดูล LCD ได้ทั้งแบบตัวอักษรและกราฟิก โดยตรงไม่ต้องต่อ ผ่านพอร์ตอีก โดยออกแบบให้ต่อเข้ากับบั๊สของ Z80180 ได้โดยตรง ในบอร์ดนี้ใช้สัญญาณการถอดรหัสแอดเดรสจาก U7 ผ่านอินเวอร์เตอร์ U8B เพื่อกลับสถานะของสัญญาณ สร้างเป็นสัญญาณ E CLOCK ของ LCD ใช้ A<sub>1</sub> และ A<sub>2</sub> เป็นสัญญาณ RS และ R/W และใช้ U8C เป็นตัวกำหนดเพจ (page) การทำงานของ LCD ในแบบกราฟิกจากสัญญาณแอดเดรส A<sub>0</sub>

### 3.2.3 วงจรเพาเวอร์อนรีเซตและวอตช์ด็อก

ใช้ไอซีของ Dallas Semiconductor เบอร์ DS1232 โดยในส่วนของเพาเวอร์อนรีเซตนั้นจะทำการรีเซต CPU เมื่อแรงดันไฟเลี้ยงมากกว่าหรือต่ำกว่า 4.75 โวลท์และจะหน่วงเวลาในกรณีเพาเวอร์อนรีเซตประมาณ 250 มิลลิวินาทีถึง 1 วินาที สำหรับในส่วนของวอตช์ด็อกนั้นสามารถเลือกระยะเวลาในการทริกได้ด้วยจัมเปอร์ TD

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์กับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERNAL I/O REGISTERS		00H
		3FH
8255 (ADC,DAC)	PA = 40H PB = 41H PC = 42H PCC = 43H	40H
82C55 (SYSTEM)	PA = 80H PB = 81H PC=82H PCC =	80H
83H		9FH
RTC 6242		A0H
	A0H	
AFH		BFH
LCD PORT		COH
	WRITE INS = COH WRITE DATA =	
C2H	READ BUSY = C4H	DFH
WATCH DOG		EOH
	ENABLE = E0H	
		FFH

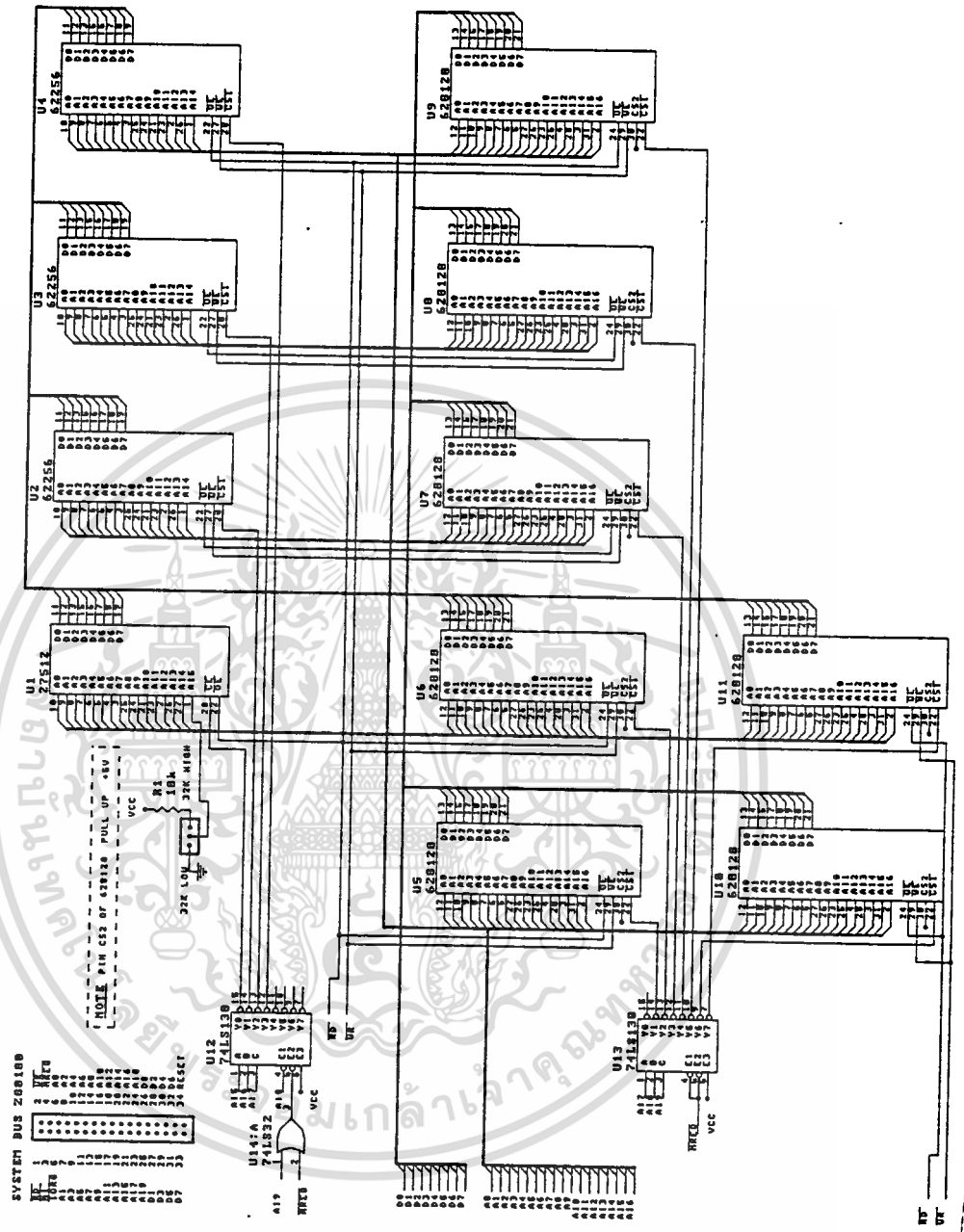
รูปที่ 3.4 แสดง I/O MAP

นอกจากนี้ยังสามารถตั้งให้วงจรวอตซ์ด็อกทำงานหรือไม่ทำงานได้ด้วยการเลือกที่จัมเปอร์ WDT และถ้ากำหนดให้วอตซ์ด็อกทำงานแล้วจะต้องทริกพอร์ตแอดเดรส E0H โดยจะใช้คำสั่ง IN หรือ OUT PORT ก็ได้ตามระยะเวลาที่เลือก เพื่อไม่ให้วงจรวอตซ์ด็อกนั้นทำการรีเซต CPU

### 3.3 รายละเอียดของบอร์ด MEMORY ขนาด 1MB

บอร์ด MEMORY ประกอบด้วย EPROM เบอร์ 27512 ขนาด 64 KB 1 ตัว, RAM เบอร์ 62256 ขนาด 32 KB 3 ตัว, RAM เบอร์ 628128 ขนาด 128 KB 7 ตัว จะได้หน่วยความจำทั้งหมด 1 MB, ไอซีถอดรหัส เบอร์ 74ALS138 2 ตัว, ไอซี OR GATE เบอร์ 7432 1 ตัว

EPROM เบอร์ 27512 (U<sub>1</sub>) จะเริ่มจากแอดเดรสที่ 00000H-07FFFH และจะแบ่งเป็น 2 ช่วง ช่วงละ 32 KB ซึ่ง 32 KB แรก (LOW) จะใช้เก็บโปรแกรม REMOTE ซึ่งเป็นโปรแกรมย้ายการทำงานส่วน INPUT (KEYBOARD) และส่วน OUTPUT (MONITOR หรือ DISPLAY) จากแผงสวิต มาใช้ในเครื่อง PC แต่การประมวลผลต่างๆจะทำในแผงสวิต ส่วน 32 KB หลัง (HIGH) จะใช้เก็บโปรแกรม



รูปที่ 3.5 วงจรของบอร์ด MEMORY.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32K	MONITOR PROGRAM	00000H
	U1	07FFFFH
32K	U2	08000H
		0FFFFFFH
32K	U3	10000H
		17FFFFH
32K	U4	18000H
		1FFFFFFH
		20000H
128K	U5	3FFFFFFH
		40000H
128K	U6	5FFFFFFH
		60000H
128K	U7	7FFFFFFH
		80000H
128K	U8	9FFFFFFH
		A0000H
128K	U9	BFFFFFFH
		C0000H
128K	U10	DFFFFFFH
		E0000H
128K	U11	FFFFFFH

รูปที่ 3.6 MEMORY DECODE

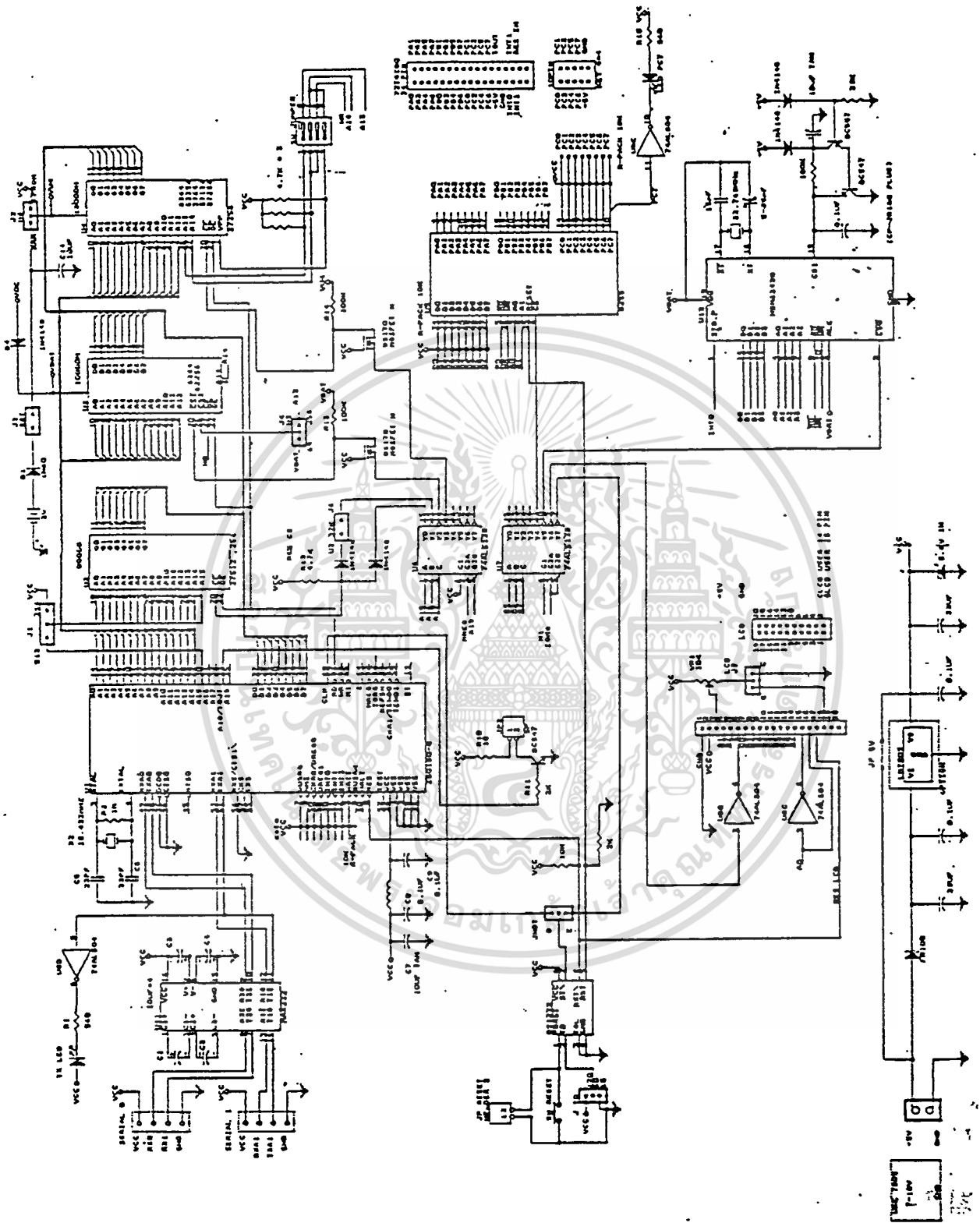
MONITOR ซึ่งเป็นโปรแกรมที่ใช้งานในแผงสวิตช์ทั้งหมด โดยมีจัมเปอร์สวิตช์ เป็นตัวเลือกซึ่งจะอยู่ด้านบนซ้ายของบอร์ด MEMORY ด้านซ้ายของสวิตช์จะเป็น 32 KB (LOW) ด้านขวาเป็น 32 KB (HIGH)

RAM เบอร์ 62256 (U<sub>2</sub>-U<sub>4</sub>) เป็น RAM ขนาด 32 KB ต่อเรียงกันจำนวน 3 ตัว โดยเริ่มจากแอดเดรสที่ 08000H-1FFFFFFH และถูกถอดรหัสแอดเดรสโดย 74ALS138

RAM เบอร์ 628128 (U<sub>5</sub>-U<sub>11</sub>) เป็น RAM ขนาด 128 KB ต่อเรียงกันจำนวน 7 ตัว โดยเริ่มจากแอดเดรสที่ 20000H-FFFFFFH และถูกถอดรหัสแอดเดรสโดย 74ALS138

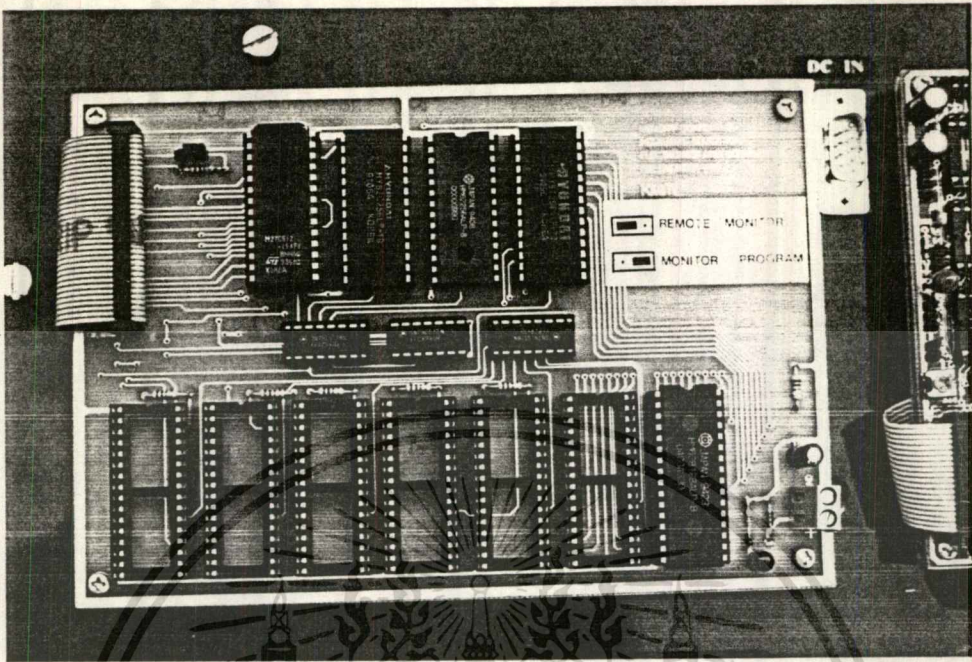
การเลือก USER PAGE ให้ดูที่ PAGE MEMORY ในภาคผนวกเป็นหลัก จะมี USER PAGE ให้ใช้ทั้งหมด 41 PAGE แบ่งเป็น PAGE ละ 24 KB ที่เหลืออีก 8 KB แบ่งเป็น USER BUFFER 2 KB, USER STACK 2 KB, SYSTEM BUFFER 2 KB และ SYSTEM STACK 2 KB เมื่อเปิดเครื่องใช้งาน MONITOR โปรแกรมจะเริ่มต้นที่ค่าใช้งานที่ PAGE 1 เราสามารถเลือก PAGE ที่ต้องการได้โดยใช้คำสั่ง X ใน HELP ซึ่งคำสั่ง X ใน HELP จะโปรแกรมค่าใน MMU ไปยัง PAGE ที่ต้องการโดยอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 วงจรของ CP-JR180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงบอร์ด MEMORY ขนาด 1 MB

### 3.3.1 การใช้พื้นที่ในหน่วยความจำในการพัฒนาโปรแกรม

ผู้ใช้ต้องดู MEMORY MAP ที่ใช้สำหรับการเขียนโปรแกรม ในภาคผนวกท้ายเล่ม ซึ่งจาก MAP จะเห็นว่า PAGE ที่เราเลือกนำมาใช้งานจะอยู่ในช่วง 08000H-0DFFFH ไม่ว่าจะผู้ใช้จะเลือก PAGE ใดจากทั้งหมด 41 PAGE ที่เลือกจะมาอยู่ในช่วง แอดเดรส ดังกล่าวโดยอัตโนมัติ ส่วน USER BUFFER จะอยู่ในช่วง 0E000H-0E7FFFH และ USER STACK จะอยู่ในช่วง 0E800-0EFFFFH

2K	SYSTEM STACK	FFFF
		F800
2K	SYSTEM BUFFER	F7FF
		F000
2K	USER STACK	FFFF
		E800
2K	USER BUFFER	E7FF
		E000
24K	PAGE ที่เลือกมาใช้งาน	DFFF
		8000
32K	MONITOR PROGRAM	7FFF
		0000

รูปที่ 3.9 MEMORY MAP ที่ใช้สำหรับเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32K	ROM MONITOR PROGRAM	00000H 07FFFFH	24K	PAGE 23	8C000H 91FFFFH
24K	PAGE 1	08000H 0DFFFFH	24K	PAGE 24	92000H 97FFFFH
24K	PAGE 2	0E000H 13FFFFH	24K	PAGE 25	98000H 9DFFFFH
24K	PAGE 3	14000H 19FFFFH	24K	PAGE 26	9B000H A3FFFFH
24K	PAGE 4	1A000H 1FFFFFH	24K	PAGE 27	A4000H A9FFFFH
24K	PAGE 5	20000H 25FFFFH	24K	PAGE 28	AA000H AFFFFFH
24K	PAGE 6	26000H 2BFFFFH	24K	PAGE 29	B0000H B5FFFFH
24K	PAGE 7	2C000H 31FFFFH	24K	PAGE 30	B6000H BBFFFFH
24K	PAGE 8	32000H 37FFFFH	24K	PAGE 31	BC000H C1FFFFH
24K	PAGE 9	38000H 3DFFFFH	24K	PAGE 32	C2000H C7FFFFH
24K	PAGE 10	3E000H 43FFFFH	24K	PAGE 33	C8000H CDFFFFH
24K	PAGE 11	44000H 49FFFFH	24K	PAGE 34	CE000H D3FFFFH
24K	PAGE 12	4A000H 4FFFFFH	24K	PAGE 35	D4000H D9FFFFH
24K	PAGE 13	50000H 55FFFFH	24K	PAGE 36	DA000H DFFFFFH
24K	PAGE 14	56000H 5BFFFFH	24K	PAGE 37	E0000H E5FFFFH
24K	PAGE 15	5C000H 61FFFFH	24K	PAGE 38	E6000H EBFFFFH
24K	PAGE 16	62000H 67FFFFH	24K	PAGE 39	EC000H F1FFFFH
24K	PAGE 17	68000H 6DFFFFH	24K	PAGE 40	F2000H F7FFFFH
24K	PAGE 18	6E000H 73FFFFH	24K	PAGE 41	F8000H FDFFFFH
24K	PAGE 19	74000H 79FFFFH	2K	USER BUFFER	FE000H FE7FFFH
24K	PAGE 20	7A000H 7FFFFFH	2K	USER STACK	FE800H FEPFFFH
24K	PAGE 21	80000H 85FFFFH	2K	SYSTEM BUFFER	FF000H FF7FFFH
24K	PAGE 22	86000H 8BFFFFH	2K	SYSTEM STACK	FF800H FFFFFFFH

รูปที่ 3.10 แสดงPAGE MEMORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแอดเดรสที่เหลือคือ F0000H-FFFFFFH จะสงวนไว้เป็น SYSTEM BUFFER และ SYSTEM STACK ผู้ใช้จะเข้าไปใช้งานในส่วนนี้ไม่ได้ เพราะจะทำให้มอนิเตอร์โปรแกรมทำงานผิดพลาด

### 3.3.2 การออกแบบการถอดรหัส MEMORY (DECODE)

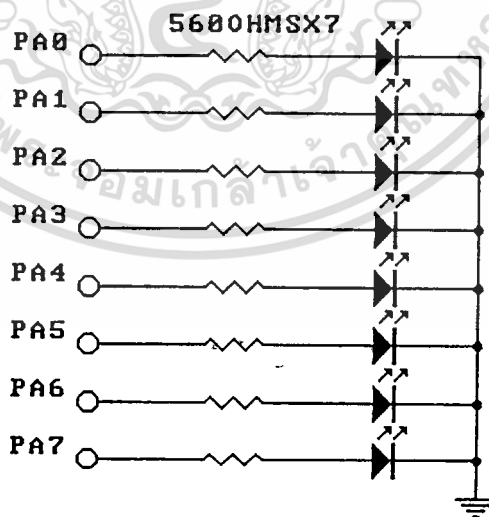
จากตารางที่ 3.1 จะสังเกตเห็นว่ามีมีการแยกการถอดรหัสอยู่ 2 ชุดโดยจะแยก RAM ขนาด 128 K ออกมาต่างหาก เป็นการถอดรหัสใช้งานเต็มตลอด 1 MB

### 3.4 LCD DISPLAY

เป็น DISPLAY แสดงผล ขนาด 16 ตัวอักษร 1 บรรทัด โดยต่อกับ แอดเดรสบัส 2 เส้น คือ A1 และ A2 และ บัสข้อมูลอีก 8 เส้น คือ D0-D7

### 3.5 วงจรแสดงผล ASCII CODE (LED 8 ดวง)

จะต่อมาจากพอร์ต A ของ 8255 ในบอร์ด JR-180 เป็นการแสดงค่ารหัสแอสกีขณะกำลังกดแป้นคีย์บอร์ดและแสดงค่ารหัสแอสกีที่รับเข้าทางพอร์ตอนุกรมของชุดฝึก Z80180 OUT ออกทางพอร์ต A ของ 8255 เพื่อเป็นการแสดงสถานะของ REGISTER



รูปที่ 3.11 แสดงวงจร LED ASCII CODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.6 USER PORT

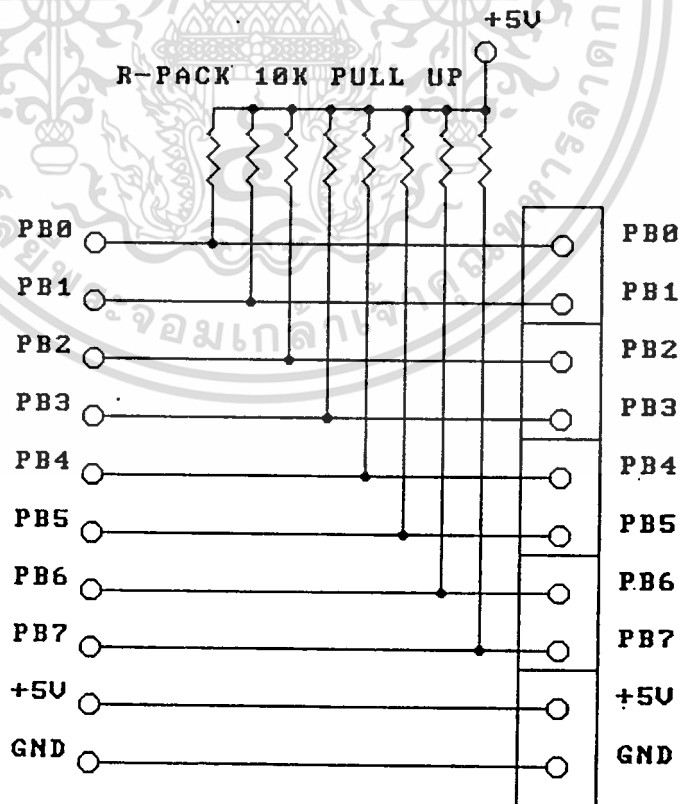
ต่อมาจากพอร์ต B ของ 8255 จากบอร์ด JR-180 และ USER PORT จะเป็นขั้วต่อไขน็อตสี่เหลี่ยม อยู่บนแผงวงจรเดียวกันกับ วงจรแสดงสถานะ REGISTER FLAG

USER PORT จะเป็น PORT ใช้งานของผู้ใช้แผงสาริตนี้ และได้ต่อความต้านทาน 10 Kohm PULL UP ไว้ทุกบิตแล้ว และยังมีขั้วต่อไฟ +5 V และ กราวนด์ เพื่อสะดวกต่อการใช้งาน

### 3.7 KEYBOARD

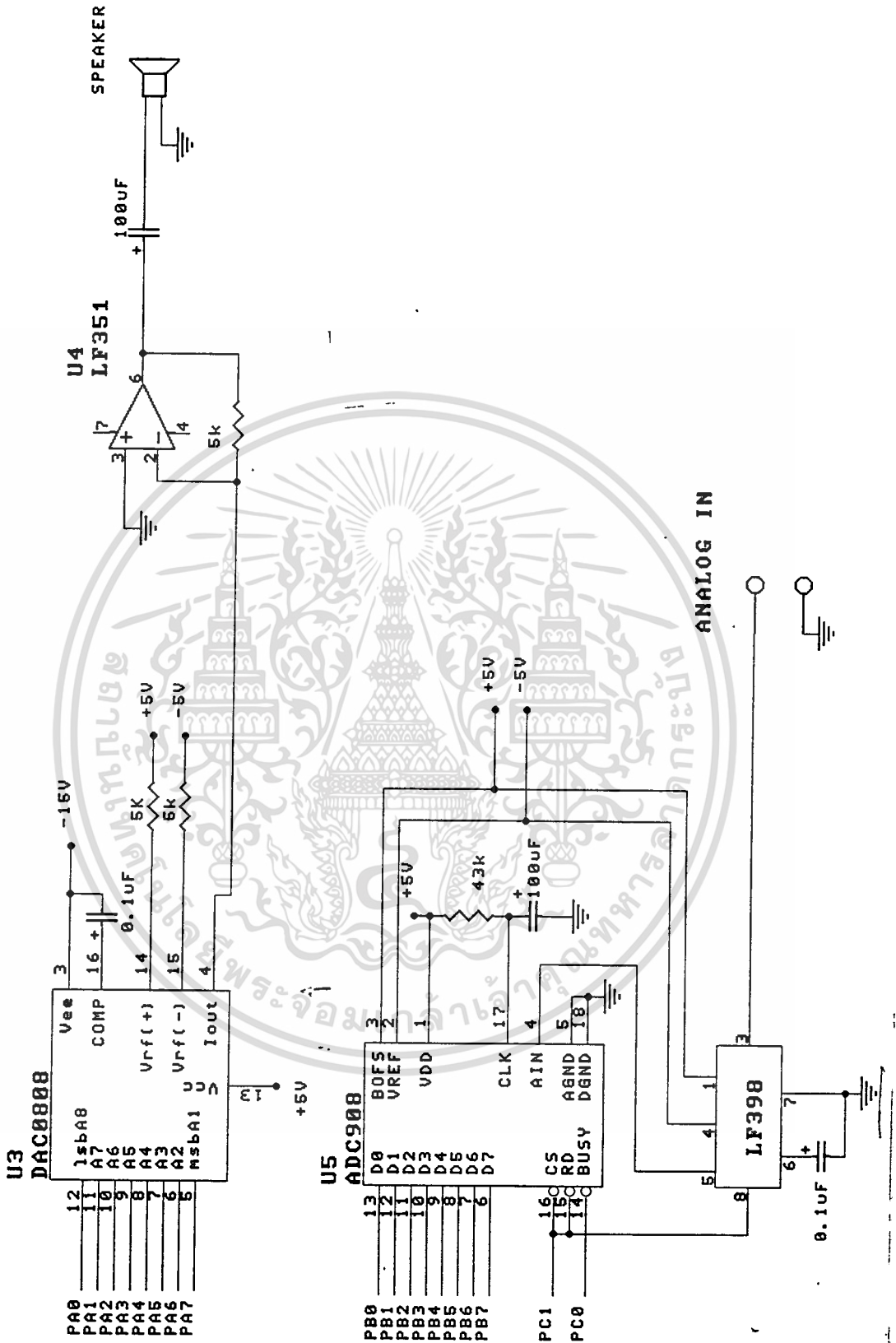
ต่อมาจากพอร์ต C ของ 8255 บนบอร์ด JR-180 ซึ่งพอร์ต C มีหน้าที่ควบคุมการ SCAN KEYBOARD เพื่อรับค่าจากการ SCAN KEYเข้ามาแล้วเปลี่ยนเป็นรหัส ASCII ยกเว้น KEY SHIFT และ KEY MENU ที่ไม่มีค่าในตาราง ASCII จึงต้องใช้รหัสพิเศษที่กำหนดขึ้นมาเอง

### 3.8 รายละเอียดของบอร์ด ANALOG TO DIGITAL CONVERSION และ DIGITAL TO ANALOG CONVERSION



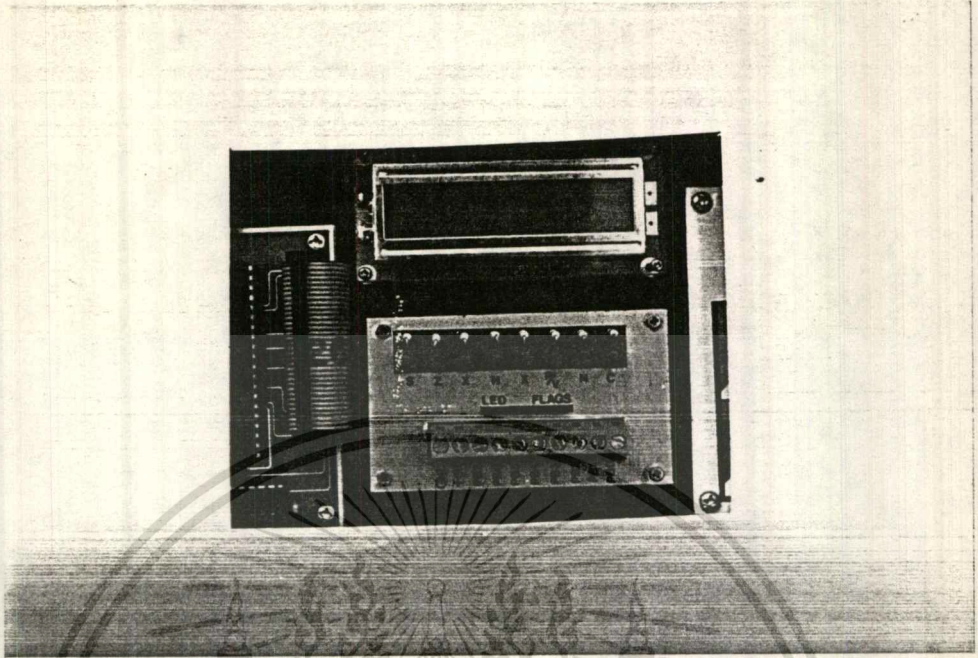
รูปที่ 3.12 แสดงวงจร USER PORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

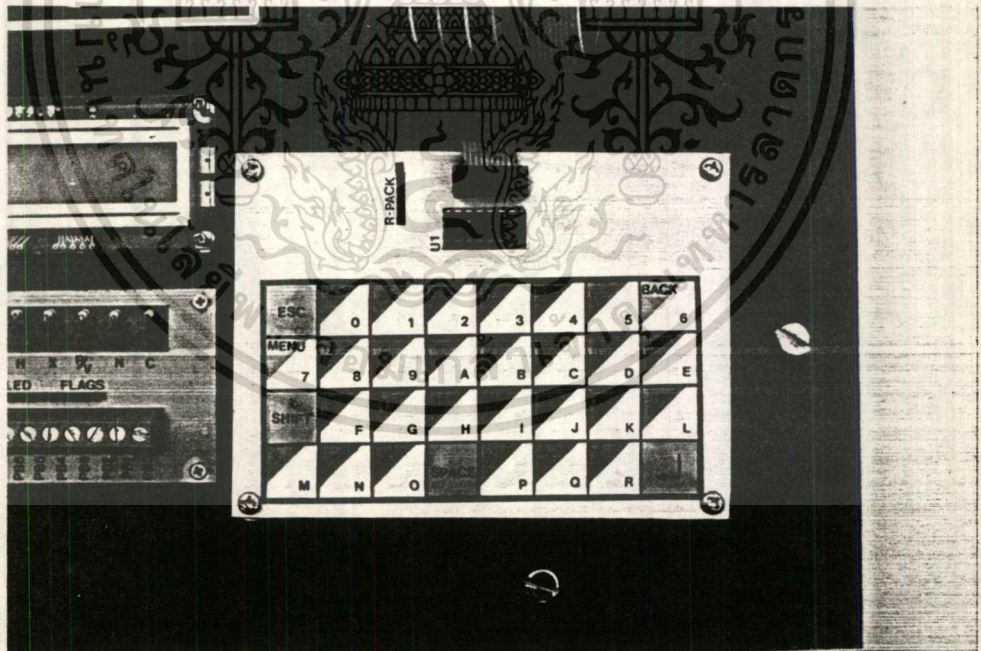


รูปที่ 3.13 วงจร ANALOG TO DIGITAL CONVERSION และ DIGITAL TO ANALOG CONVERSION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดง LCD DISPLAY , วงจรแสดงผล REGISTER FLAG และ USER PORT



รูปที่ 3.15 แสดงแป้น KEYBOARD

บนบอร์ด จะใช้ไอซีแปลง DIGITAL TO ANALOG เบอร์ DAC 0808 1 ตัว ขนาด 8 บิต, ไอซี ANALOG TO DIGITAL เบอร์ ADC 908 ขนาด 8 BIT, ไอซี อินพุต/เอาต์พุต พอร์ต เบอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

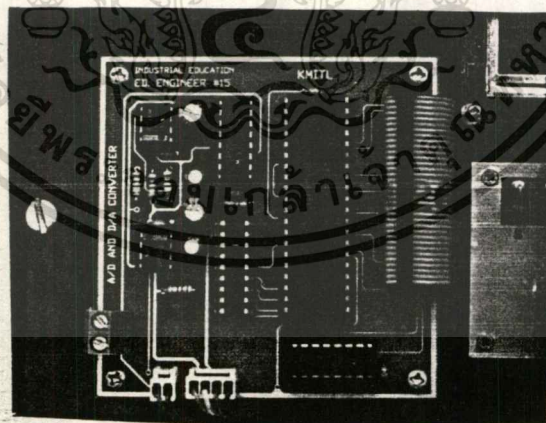
8255 1 ตัว, ไอซี SAMPLING AND HOLD เบอร์ LF 398 1ตัว, ไอซี OP-AMP (I/V CONVERTER) เบอร์ LF 351 1ตัว, ไอซีถอดรหัสแอดเดอเรส (DECODER) เบอร์ 74ALS138 และ ลำโพง 0.25 W 8 ohm 1ตัว

#### DAC 0808

ทำหน้าที่แปลงสัญญาณ DIGITAL เป็น ANALOG โดยที่ 8255 จะ OUT PORT จากพอร์ต A มาที่ขา อินพุต ของ DAC 0808 8 BIT คือ A1-A8 (ขา5-ขา12) DAC 0808 ก็จะทำการแปลงเป็นสัญญาณ ANALOG ออกที่ขา I<sub>OUT</sub> (ขา4) และจะมี LF 351 มีหน้าที่ทำให้กระแสที่ออกมาจากขา I<sub>OUT</sub> (ขา4) เป็นแรงดัน (I/V CONVERTER) มาขับลำโพงทำให้เกิดเสียง

#### ADC 908

ทำหน้าที่แปลงสัญญาณอนาล็อกเป็นดิจิตอล โดยที่จะมีการป้อนสัญญาณอนาล็อกมาทางช่อง ANALOG IN (ขั้วต่อไขน็อตสี่เหลี่ยม) และจะมี LF 398 เป็นตัว SAMPLING ค่าของสัญญาณอนาล็อก และเก็บค่าที่ SAMPLING ไว้รอให้ ADC 908 มาอ่านค่าที่ SAMPLING ไว้ เมื่อ ADC 908 อ่านค่าไปก็จะนำค่าที่ได้ไปทำการแปลงเป็นสัญญาณดิจิตอลแล้ว LF 398 ก็จะทำการ SAMPLING ใหม่อีก จะทำกระบวนการต่างๆนี้ไปเรื่อยๆจนกว่าสัญญาณอนาล็อกจะไม่มี



รูปที่ 3.16 บอร์ด ANALOG TO DIGITAL CONVERSION และ DIGITAL TO ANALOG CONVERSION

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADC 908 จะอ่านค่าที่ถูก SAMPLING จากขา 5 ของ LF 398 เข้ามาทางขา A<sub>IN</sub> (ขา 4) ของตัวเอง แล้วทำการแปลงเป็นสัญญาณดิจิตอลออกทาง เอาท์พุท คือ ขา D0-D7 (ขา6-ขา13) ไปยัง พอร์ต B ของ 8255 ส่วนสัญญาณ ความคุมตัว ADC 908 จะได้มาจากพอร์ต C ของ 8255

### 8255

จะถูก ถอดรหัสโดย 74ALS138 ซึ่งได้สัญญาณการถอดรหัสจาก แอดเดรสบัสของ CPU ขา A4,A5,A6 มาที่ขา A,B,C ตามลำดับของ 74ALS138 ได้สัญญาณจาก A7 ,IORQ ,M<sub>1</sub> ตาม CPU ขา A4,A5,A6 มาที่ขา A,B,C ตามลำดับของ 74ALS138 ได้สัญญาณจาก A7,IORQ,M1ตามลำดับ จาก CPU

พอร์ต A ของ 8255 ต่อกับขาอินพุทของ DAC 0808

พอร์ต B ของ 8255 ต่อกับขาเอาท์พุทของ ADC 908

พอร์ต C ของ 8255 จะใช้ PC0,PC1 (ขา 14,ขา 15) ต่อกับขา BUSY และ RD,CS ของ ADC 908

## บทที่ 4

### ผลการทดลองและทดสอบ

#### 4.1 การทดสอบ LED ASCII CODE และ USER PORT

โปรแกรม TEST1.ASM (ดูที่ภาคผนวก ก) เป็นโปรแกรมแสดงไฟรั้งจากขาไปซ้ายที่ละ 1 ดวง ที่ LED ASCII CODE ในขณะที่โปรแกรมกำลังทำงาน ค่า LOGIC ที่ USER PORT (พอร์ต B) จะมีค่าเท่ากับที่ LED ASCII CODE

##### 4.1.1 ขั้นตอนการทดลอง

- 1.จ่ายไฟให้กับชุดฝึกและพัฒนา Z80180 แล้วต่อ EPROM EMULATOR ไว้แทนที่ตำแหน่ง U1
- 2.เขียนโปรแกรมแล้วให้โปรแกรมเริ่มทำงานโดยใช้ ASSEMBLER C-32 เป็นตัวแปลคำสั่ง
- 3.สังเกตการติดสว่างของหลอด LED
- 4.ใช้ LOGIC PROBE วัดที่ USER PORT

##### 4.1.2 ผลการทดลอง

จะเห็นไฟรั้งจากขามาซ้ายที่ละดวงที่ LED ASCII CODE และขณะที่หลอด LED ดวงไหนของ LED ASCII CODE สว่าง ค่า LOGIC ที่ USER PORT บิตนั้นๆก็จะเป็น "1"ตามไปด้วย แสดงว่าชุด LED ASCII CODE และ USER PORT ทำงานได้สมบูรณ์

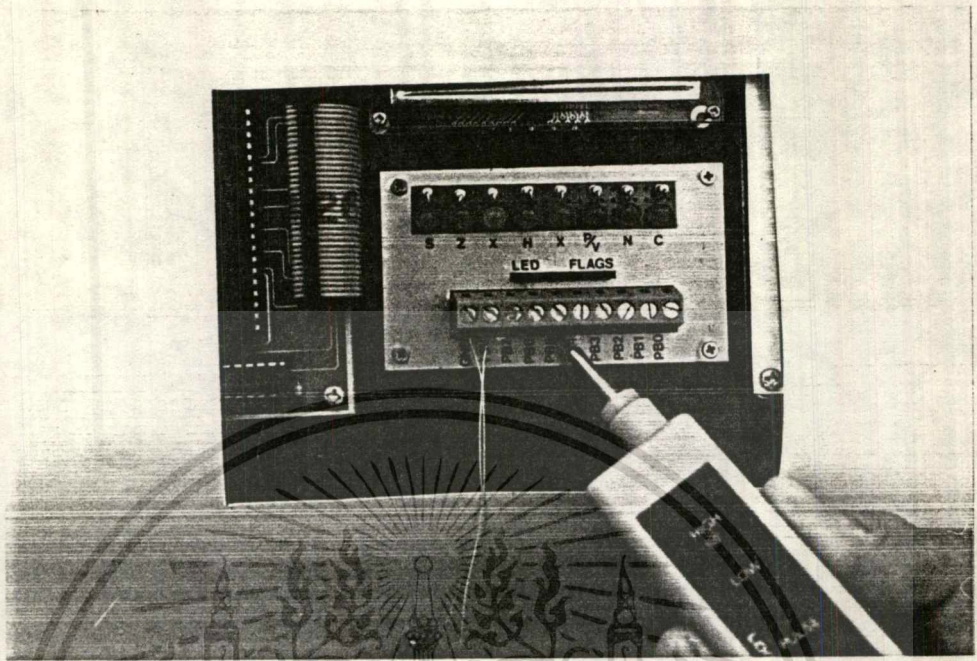
#### 4.2 การทดสอบ MEMORY BOARD 1MB

โปรแกรม TEST2.ASM (ดูที่ภาคผนวก ก) เป็นโปรแกรมที่นำค่า A5H ไปใส่ใน ADDRESS ที่เราต้องการโดยเราจะกำหนดค่า ADDRESS ลงในโปรแกรมและเลือก PAGE ของหน่วยความจำ โดยไปเปลี่ยนคำสั่ง CALL เช่น เลือก PAGE1 ก็ใช้ CALL PAGE1 เป็นต้น โปรแกรมที่จะตั้งค่าอัตโนมัติที่ PAGE1

##### 4.2.1 ขั้นตอนการทดลอง

- 1.จ่ายไฟให้กับชุดฝึกและพัฒนา Z80180 แล้วต่อ EPROM EMULATOR ไว้แทนที่ตำแหน่งของ U1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงผลการทดสอบ LED ASCII CODE และ USER PORT

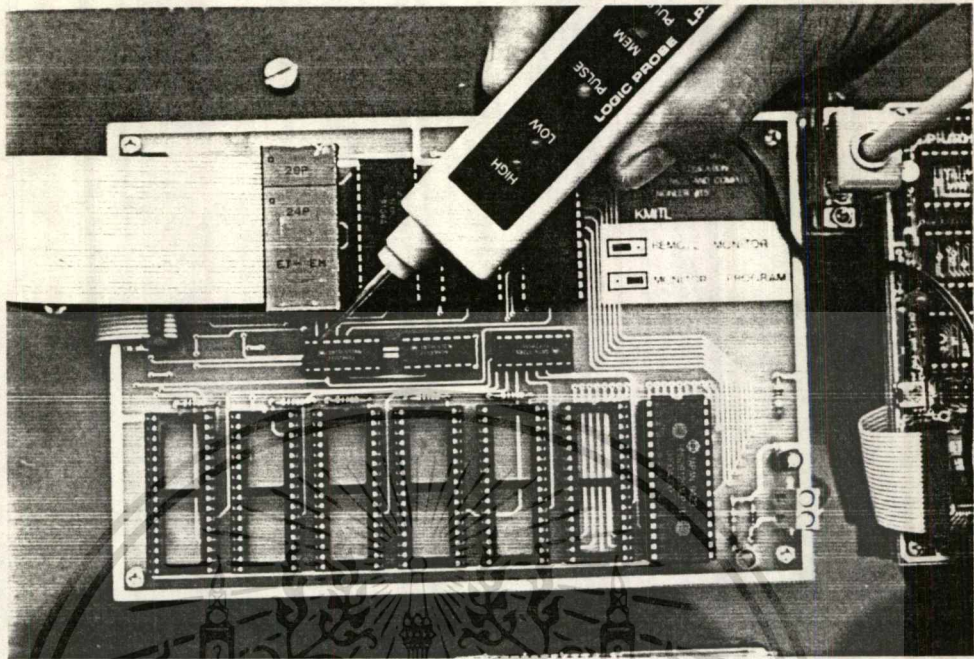
- 2.เขียนโปรแกรมแล้วให้โปรแกรมทำงานโดยใช้ ASSEMBLER C-32 เป็นตัวแปลคำสั่ง
- 3.นำ LOGIC PROBE วัดที่ขา Y ใดๆของ 74LS138 ให้สัมพันธ์กับช่วง ADDRESS ที่เรากำลังทดสอบ เช่น เราทดสอบ ADDRESS ที่ DFFFH เราก็วัดที่ Y1 ของ U12
- 4.สังเกตการติดสว่างที่ LED ASCII CODE

#### 4.2.2 ผลการทดลอง

จะเกิด LOGIC LOW ที่ขา Y ที่เลือก ช่วงหน่วยความจำที่กำลังทดสอบอยู่และที่ LED ASCII CODE จะได้สภาวะ LOGIC ดังนี้ " 10100101 " (A5) ซึ่งแสดงว่า หน่วยความจำเก็บค่า A5 ไว้ได้จริง ซึ่งแสดงว่าหน่วยความจำในช่วงที่เราทดสอบนั้นทำงานสมบูรณ์

#### 4.3 การสอบจอแสดงผลแบบ LCD

โปรแกรม TEST3.ASM เป็นโปรแกรมควบคุมจอแสดงผลแบบ LCD สร้างตัวอักษรคำว่า ED.ENGINEER #15



รูปที่ 4.2 แสดงการวัดค่า CHIP SELECT และแสดง LED ASCII CODE

#### 4.3.1 ขั้นตอนการทดลอง

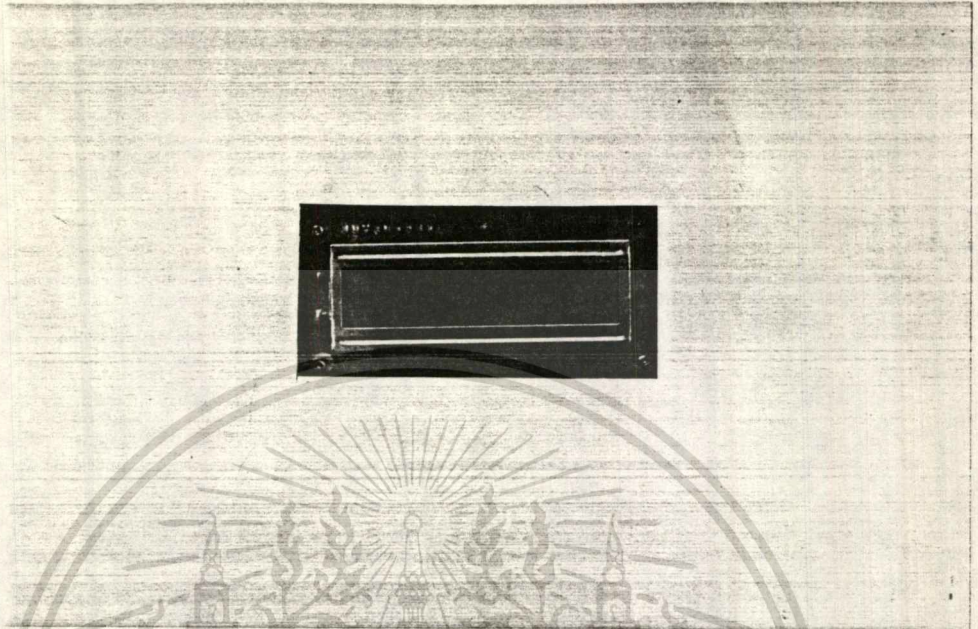
1. จ่ายไฟให้กับชุดฝึกและพัฒนา Z80180 แล้วต่อ EPROM EMULATOR ไว้แทนที่ตำแหน่งของ U1
2. เขียนโปรแกรมแล้วให้โปรแกรมทำงานโดยใช้ ASSEMBLER C-32 เป็นตัวแปลคำสั่ง
3. สังเกตที่หน้าจอ LCD

#### 4.3.2 ผลการทดลอง

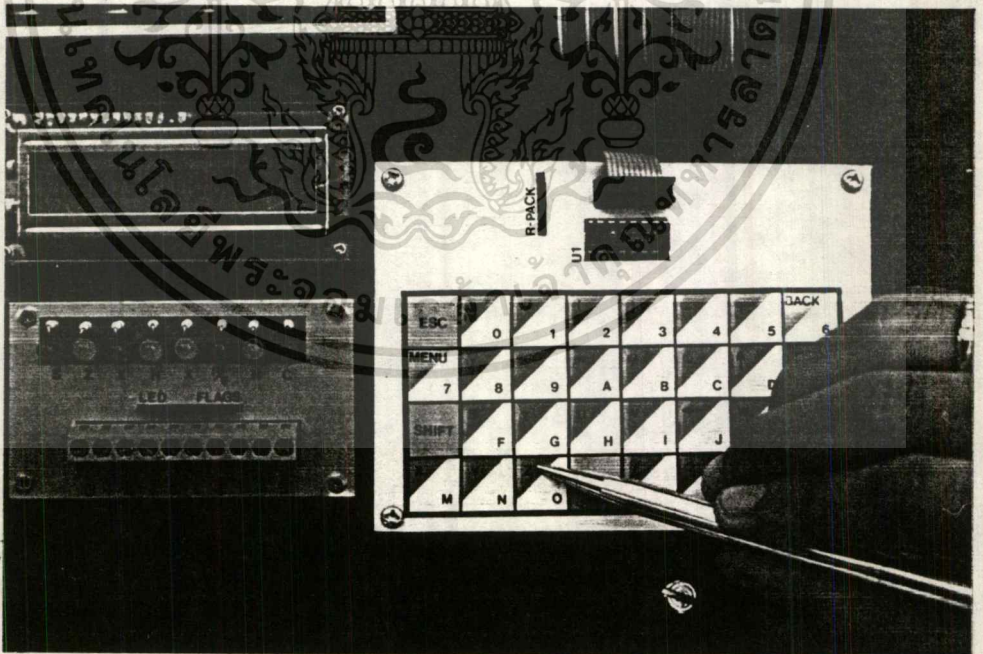
จะเกิดคำว่า ED.ENGINEER #15 ที่จอ LCD

#### 4.4 การทดสอบ KEYBOARD

โปรแกรม TEST4.ASM เป็นโปรแกรมที่รับค่า ตัวอักษรที่ถูกเรากดเปลี่ยนเป็นรหัส ASCII แล้วนำไปแสดงที่ LED ASCII CODE เช่น เรากดตัวอักษร "B" ที่ LED ASCII CODE จะเกิดค่าที่ตรงตามรหัส ASCII คือ 42H (0100 0010)



รูปที่ 4.3 แสดงผลการทดสอบจอ LCD



รูปที่ 4.4 แสดงการกดตัวอักษรบนแป้นพิมพ์และ LED ASCII CODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4.1 ขั้นตอนการทดลอง

- 1.จ่ายไฟให้กับชุดฝึกและพัฒนา Z80180 แล้วต่อ EPROM EMULATOR ไว้แทนที่ตำแหน่งของ U1
- 2.เขียนโปรแกรมแสดงให้โปรแกรมทำงาน โดยใช้ ASSEMBLER C-32 เป็นตัวแปลคำสั่ง
- 3.กดแป้นพิมพ์ แล้วสังเกตการเปลี่ยนแปลงที่ LED ASCII CODE

#### 4.4.2 ผลการทดลอง

LED ASCII CODE จะแสดงค่าตามรหัส ASCII ที่เรากดแป้นพิมพ์



## บทที่ 5

# บทวิจารณ์สรุปและแนวทางการพัฒนา

### บทสรุป

การสร้างชุดฝึกและพัฒนาไมโครโปรเซสเซอร์ Z80180 ของบริษัทยาโนพอนด์นี้  
จะได้ประโยชน์ในด้าน

- นำไปใช้ในการเรียนการสอนวิชาที่เกี่ยวข้องกับไมโครโปรเซสเซอร์
- นำไปเป็นเครื่องตัวอย่างในการประยุกต์ใช้งาน
- สามารถพัฒนาโปรแกรมบนเครื่อง PC แล้วส่งผ่านไปยังชุดฝึกและพัฒนา  
ไมโครโปรเซสเซอร์ Z80180 ได้ เพิ่มความสะดวกในการทำงานมากยิ่งขึ้น
- นำไปเป็นชุดสาธิตเบื้องต้นในการประมวลผลสัญญาณเชิงเลขได้

การใช้งานชุดฝึกนี้เป็นการฝึกการเขียนและทำความเข้าใจโปรแกรม เพื่อให้  
ไมโครโปรเซสเซอร์ไปทำงานควบคุมระบบตามต้องการได้อย่างง่าย และสะดวกในการทำงาน  
จากการมีอุปกรณ์สนับสนุนในการอำนวยความสะดวกในการทำโปรแกรมอย่างครบถ้วน  
ชุดฝึกนี้คณะผู้จัดทำตั้งใจทำขึ้นเพื่อใช้ประกอบการเรียนการสอนในการเขียน  
โปรแกรมในวิชาที่เกี่ยวกับไมโครโปรเซสเซอร์ หรือวิชาปฏิบัติการในห้องทดลอง เพื่อเสริมสร้าง  
ความรู้ ความเข้าใจ และความชำนาญในการเขียนโปรแกรมจากการศึกษาภาคทฤษฎีแล้ว ให้มี  
ความเข้าใจดียิ่งขึ้น และยังมีใบงานประกอบการทดลองเพื่อให้ผู้ที่สนใจได้ฝึกหัดเป็นแนวทางใน  
การเขียนโปรแกรมในระดับที่สูงขึ้นต่อไปได้

### ปัญหาและอุปสรรค

1. ADC908 ที่นำมาใช้ในวงจร A/D ในช่วงที่ทดลองโครงงานนี้หาซื้อได้ยาก จึง  
ทำให้การทดลองโครงงานเกิดการล่าช้า
2. ราคาของ RAM เบอร์ 628128 ใน MEMORY BOARD มีราคาค่อนข้างสูง
3. เนื่องจากแผ่นวงจรพิมพ์ได้ถูกออกแบบและสั่งทำใหม่เป็นส่วนมาก ทำให้ต้อง  
เสียเวลาในการออกแบบวงจร สายวงจร และ เวลาในการรอรับแผ่นวงจรพิมพ์จากทางร้าน ทำให้  
การเขียนโปรแกรมเพื่อทดสอบ HARDWARE ล่าช้าไปมาก
4. ในการเขียนและพัฒนาโปรแกรมจะต้องใช้ EPROM EMULATOR แทน  
EPROM จริงที่ใช้ในแผ่นวงจรพิมพ์ ทำให้ในการเคลื่อนย้ายต้องถอด EPROM EMULATOR  
ออกเสียก่อน เป็นผลให้ SOCKET ของ EPROM ในแผ่นวงจรพิมพ์ทวม

## แนวทางการพัฒนา

1. เนื่องจาก MAIN BOARD ในโครงการนี้ได้ใช้ของบริษัท ETT ในการพัฒนาต่อไป ควรออกแบบ MAIN BOARD ใหม่ โดยให้มีช่องเสียบ (SLOT) และทำแผงวงจรพิมพ์อื่นๆให้เป็น CARD สำหรับเสียบ เพื่อความสะดวกในการทดลองและแก้ไข
2. เนื่องจากโครงการนี้ได้ทำการแยก POWER SUPPLY และ ตัวโครงการ ออกจากกัน ในการพัฒนาต่อไป ควรออกแบบให้อยู่ในชุดเดียวกัน





**ภาคผนวก**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก

**โปรแกรมทดสอบการทำงาน**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมทดสอบการทำงานของ FLAG และ USER PORT TEST1.ASM

```
; FILENAME TEST1.ASM
; DESCRIPTION TEST LED FLAGS,USER PORT
; ASSEMBLER CROSS-32 V3.0
; KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
, FACULTY OF INDUSTRIAL EDUCATION
; MAJOR: ELECTRONIC AND COMPUTER
, ED.ENGINEER #15
```

```
CPU "Z180.TBL" ; CPU TABLE
HOF "INT8" ; HEX OUTPUT FORMAT
```

```
PA: EQU 80H
PB: EQU 81H
PC: EQU 82H
PCC: EQU 83H
```

```
ORG 0H
START: LD HL,2000H ; POWER UP DELAY
START1: DEC HL
LD A,H
OR L
JR NZ,START1
LD SP,0DFFFH ; STACK POINTER
LD A,80H
OUT (PCC),A
LD A,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ OUT (PC),A เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,1
L1: OUT (PA),A ; TEST LED FLAGS
OUT (PB),A ; TEST USER PORT
CALL DELAY
RLA
JR L1

```

```

;
DELAY: PUSH AF
PUSH BC
LD B,0FFH
DE1: LD C,0FFH
DE2: DEC C
JR NZ,DE2
DJNZ DE1
POP BC
POP AF
RET
;
END

```

## โปรแกรมทดสอบการทำงานของ MEMORY BOARD TEST2.ASM

```

; FILENAME TEST2.ASM
; DESCRIPTION TEST MEMORY 1MB (SEE PAGE MEMORY,
; MEMORY DECODE IN APPENDIX)
; ASSEMBLER CROSS-32 V3.0
; KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
; FACULTY OF INDUSTRIAL EDUCATION

```

เอกสารนี้ MAJOR: ELECTRONIC AND COMPUTER การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ED.ENGINEER #15
```

```
;
```

```
CPU "Z180.TBL" ; CPU TABLE
```

```
HOF "INT8" ; HEX OUTPUT FORMAT
```

```
;
```

```
PA: EQU 80H
```

```
PB: EQU 81H
```

```
PC: EQU 82H
```

```
PCC: EQU 83H
```

```
CBR: EQU 38H
```

```
BBR: EQU 39H
```

```
CBAR: EQU 3AH
```

```
OMCR: EQU 3EH
```

```
ADDRESS: EQU 8000H
```

```
;
```

```
ORG 0H
```

```
START: LD HL,2000H ;POWER UP DELAY
```

```
START1: DEC HL
```

```
LD A,H
```

```
OR L
```

```
JR NZ,START1
```

```
;
```

```
LD A,0 ;FOR Z80180 ONLY
```

```
OUT0 (OMCR),A
```

```
;
```

```
DEFAULT: LD A,0EEH ;MMU SETUP
```

```
OUT0 (CBAR),A
```

```
LD A,0F0H
```

```
OUT0 (BBR),A
```

```
OUT0 (CBR),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD SP,0F800H ;SYSTEM STACK

LD A,80H

OUT (PCC),A

CALL PAGE1 ;SET MMU SELECT YOUR PAGE

LOOP: LD HL,ADDRESS

LD A,0A5H ;1 0 1 0 0 1 0 1

LD (HL),A ;FILL A5 IN MEMORY

XOR A

LD A,(HL) ;GET A5 FROM MEMORY

OUT (PA),A ;DISPLAY A5 ON PORT A

JR LOOP

PAGE1: LD A,0EEH

OUT0 (CBAR),A

LD A,0F0H

OUT0 (CBR),A

OUT0 (BBR),A

RET

PAGE2: LD A,0E8H

OUT0 (CBAR),A

LD A,0F0H

OUT0 (CBR),A

LD A,06H

OUT0 (BBR),A

RET

PAGE3: LD A,0E8H

```

```

LD A,0F0H
OUT0 (CBR),A
LD A,24H
OUT0 (BBR),A
RET
PAGE8: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,2AH
OUT0 (BBR),A
RET
PAGE9: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,30H
OUT0 (BBR),A
RET
PAGE10: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,36H
OUT0 (BBR),A
RET
PAGE11: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTO (CBR),A

LD A,34H

OUTO (BBR),A

RET

PAGE12: LD A,0E8H  
OUTO (CBAR),A

LD A,0F0H

OUTO (CBR),A

LD A,42H

OUTO (BBR),A

RET

PAGE13: LD A,0E8H  
OUTO (CBAR),A

LD A,0F0H

OUTO (CBR),A

LD A,48H

OUTO (BBR),A

RET

PAGE14: LD A,0E8H  
OUTO (CBAR),A

LD A,0F0H

OUTO (CBR),A

LD A,4EH

OUTO (BBR),A

RET

PAGE15: LD A,0E8H  
OUTO (CBAR),A

LD A,0F0H

OUTO (CBR),A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,54H
OUT0 (BBR),A
RET
PAGE16: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,5AH
OUT0 (BBR),A
RET
PAGE17: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,60H
OUT0 (BBR),A
RET
PAGE18: LD A,0E8H
OUT0 (CBAR),A
LD A,CF0H
OUT0 (CBR),A
LD A,66H
OUT0 (BBR),A
RET
PAGE19: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,6CH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        OUT0 (BBR),A
        RET
PAGE20: LD  A,0E8H
        OUT0 (CBAR),A
        LD  A,0F0H
        OUT0 (CBR),A
        LD  A,72H
        OUT0 (BBR),A
        RET
PAGE21: LD  A,0E8H
        OUT0 (CBAR),A
        LD  A,0F0H
        OUT0 (CBR),A
        LD  A,78H
        OUT0 (BBR),A
        RET
PAGE22: LD  A,0E8H
        OUT0 (CBAR),A
        LD  A,0F0H
        OUT0 (CBR),A
        LD  A,7EH
        OUT0 (BBR),A
        RET
PAGE23: LD  A,0E8H
        OUT0 (CBAR),A
        LD  A,0F0H
        OUT0 (CBR),A
        LD  A,84H
        OUT0 (BBR),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
PAGE24: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,8AH
        OUT0 (BBR),A
        RET
PAGE25: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,90H
        OUT0 (BBR),A
        RET
PAGE26: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,96H
        OUT0 (BBR),A
        RET
PAGE27: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,9CH
        OUT0 (BBR),A
        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PAGE28: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,0A2H
        OUT0 (BBR),A
        RET

```

```

PAGE29: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,0A8H
        OUT0 (BBR),A
        RET

```

```

PAGE30: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,0AEH
        OUT0 (BBR),A
        RET

```

```

PAGE31: LD A,0E8H
        OUT0 (CBAR),A
        LD A,0F0H
        OUT0 (CBR),A
        LD A,0B4H
        OUT0 (BBR),A
        RET

```

```

PAGE32: LD A,0E8H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OUTO (CBAR),A
LD A,0F0H
OUTO (CBR),A
LD A,0BAH
OUTO (BBR),A
RET

```

```

PAGE33: LD A,0E8H
OUTO (CBAR),A

```

```

LD A,0F0H
OUTO (CBR),A
LD A,0C0H
OUTO (BBR),A
RET

```

```

PAGE34: LD A,0E8H
OUTO (CBAR),A

```

```

LD A,0F0H
OUTO (CBR),A
LD A,0C6H
OUTO (BBR),A
RET

```

```

PAGE35: LD A,0E8H
OUTO (CBAR),A

```

```

LD A,0F0H
OUTO (CBR),A
LD A,0CCH
OUTO (BBR),A
RET

```

```

PAGE36: LD A,0E8H
OUTO (CBAR),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD A,0F0H
OUT0 (CBR),A
LD A,0D2H
OUT0 (BBR),A
RET
```

```
PAGE37: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,0D8H
OUT0 (BBR),A
RET
```

```
PAGE38: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,0DEH
OUT0 (BBR),A
RET
```

```
PAGE39: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
OUT0 (CBR),A
LD A,0E4H
OUT0 (BBR),A
RET
```

```
- PAGE40: LD A,0E8H
OUT0 (CBAR),A
LD A,0F0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        OUTD (CBR),A
        LD  A,0EAH
        OUTD (BBR),A
        RET
PAGE41: LD  A,0E8H
        OUTD (CBAR),A
        LD  A,0F0H
        OUTD (CSR),A
        LD  A,0F0H
        OUTD (BBR),A
        RET
        END

```

## โปรแกรมทดสอบการทำงานของจอ LCD TEST3.ASM

```

; FILENAME  TEST3.ASM
; DESCRIPTION  TEST LCD 16 CHARACTERS * 1 LINE
; ASSEMBLER  CROSS-32 V3.0
; KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
; FACULTY OF INDUSTRIAL EDUCATION
; MAJOR: ELECTRONIC AND COMPUTER
; ED.ENGINEER #15
;

```

```

        CPU "Z180.TBL" ; CPU TABLE
        HOF "INT8"     ; HEX OUTPUT FORMAT
;

```

```

        WRITE_INS : EQU 0C0H

```

```

        WRITE_DATA : EQU 0C2H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
READ_BUSY: EQU 0C4H
```

```
ORG 0H
```

```
START: LD HL,5000H ; POWER UP DELAY
```

```
START1: DEC HL
```

```
LD A,H
```

```
OR L
```

```
JR NZ,START1
```

```
LD SP,0DFFFH ; STACK POINTER
```

```
TLCD : CALL INITLCD
```

```
LD HL,TABLE
```

```
CALL WRP
```

```
SLP
```

```
TABLE : DFB "ED. ENGINEER #15"
```

```
WRP : LD A,0
```

```
CALL GOTO
```

```
CALL WRLINE
```

```
LD A,40H
```

```
CALL GOTO
```

```
CALL WRLINE
```

```
RET
```

```
GOTO : SET 7,A
```

```
OUT (WRITE_INS),A
```

```
CALL READ
```

```
RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

READ : IN  A,(READ_BUSY)

      BIT 7,A

      JR  NZ,READ

      RET

```

```

WRLINE : LD  B,8

TEST11 : LD  A,(HL)

      LD  D,A

      CALL WRBYTE

      INC HL

      DJNZ TEST11

      RET

```

```

;*** INITIAL LCD ***
;

```

```

INITLCD: LD  A,00111000B ; FUNCTION SET 38H

      ; DL=1 8 BIT,N=1 1/16 DUTY,F=0 5*7

      OUT (WRITE_INS),A

      CALL DELAY1

      CALL DELAY1 ; DELAY > 4.1 ms

      LD  A,00001111B ; DISPLAY ON/OFF CONTROL

      ; D=1 OFF,C=1 CURSOR ON,B=1 BLINK

      OUT (WRITE_INS),A

      CALL DELAY1

      LD  A,00000110B ; ENTRY MODE SET

      ; I/D=1 INCREMENT, S=0 RIGHT

      OUT (WRITE_INS),A

      CALL DELAY1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;
;
        CLSLCD : LD  A,00000001B ; CLEAR ALL DISPLAY
                OUT (WRITE_INS),A
                CALL READ
                RET
;
;*** WRITE BYTE SUB ***
;
        WRBYTE : LD  A,D
                OUT (WRITE_DATA),A
                CALL READ
                RET
;
;**** READY SUB ****
;
        DELAY1 : PUSH BC
                PUSH AF
                LD   B,80H
        DELAY12 : DJNZ DELAY12
                POP AF
                POP BC
                RET
;
                END

```

## โปรแกรมทดสอบการทำงานของคีย์บอร์ด TEST4.ASM

PORTC: EQU 82H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ORG 8000H

KEYIN:      DFS 1
KEY_SHIFT:  DFS 1

ORG 0000H

LD A,88H
OUT (83H),A

SCAN_KEY:  LD B,08H
           LD E,00H
SCAN1:     LD A,E
           OUT (PORTC),A
           IN A,(PORTC)
           AND 0F0H
           CP 0F0H
           JR NZ,SCAN2
           INC E
           DJNZ SCAN1
           JR SCAN_KEY

SCAN2:     OR E
           LD (KEYIN),A

DEBOUND:  IN A,(PORTC)
           AND 0F0H
           CP 0F0H
           JR NZ,DEBOUND

LD A,(KEYIN)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 0B0H
JR NZ,KEYCODE
LD A,01H
LD (KEY_SHIFT),A
JR SCAN_KEY

KEY_CODE:
LD A,(KEY_SHIFT)
CP 01H
JR NZ,CODE1
LD HL,PCTAB
LD IX,SHIFTTAB
JR CODE2
CODE1:
LD HL,PCTAB
LD IX,NORMALTAB
CODE2:
LD A,(KEYIN)
LD B,1FH
CODE2_1:
CP (HL)
JR Z,CODE3
INC HL
INC IX
DJNZ CODE2_1
CODE3:
LD A,(IX+00H)
LD D,A
XOR A
LD (KEY_SHIFT),A
LD A,D
OUT (80H),A
JP SCAN_KEY

```

```

*****
PCTAB:      DFB  0E0H,0E1H,0E2H,0E3H,0E4H,0E5H,0E6H,0E7H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 0D0H,0D1H,0D2H,0D3H,0D4H,0D5H,0D6H,0D7H

DFB 0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H

DFB 070H,071H,072H,073H,074H,075H,076H,077H

NORMALTAB: DFB 1BH,30H,31H,32H,33H,34H,35H,36H

DFB 37H,38H,39H,41H,42H,43H,44H,45H

DFB 46H,47H,48H,49H,4AH,4BH,4CH

DFB 4DH,4EH,4FH,20H,50H,51H,52H,0DH

SHIFTTAB: DFB 1BH,23H,2AH,28H,29H,2DH,2BH,08H

DFB FFH,2CH,3AH,22H,27H,2EH,41H,3DH

DFB 53H,54H,55H,56H,44H,57H,43H

DFB 58H,59H,5AH,20H,2FH,3FH,42H,0DH

END.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32K	ROM MONITOR PROGRAM	00000H	24K	PAGE 23	8C000H
		07FFFH			91FFFH
24K	PAGE 1	08000H	24K	PAGE 24	92000H
		0DFFFH			97FFFH
24K	PAGE 2	0E000H	24K	PAGE 25	98000H
		13FFFH			9DFFFH
24K	PAGE 3	14000H	24K	PAGE 26	9E000H
		19FFFH			A3FFFH
24K	PAGE 4	1A000H	24K	PAGE 27	A4000H
		1FFFFH			A9FFFH
24K	PAGE 5	20000H	24K	PAGE 28	AA000H
		25FFFH			AFFFFH
24K	PAGE 6	26000H	24K	PAGE 29	B0000H
		2BFFFH			B5FFFH
24K	PAGE 7	2C000H	24K	PAGE 30	B6000H
		31FFFH			B8FFFH
24K	PAGE 8	32000H	24K	PAGE 31	BC000H
		37FFFH			C1FFFH
24K	PAGE 9	38000H	24K	PAGE 32	C2000H
		3DFFFH			C7FFFH
24K	PAGE 10	3E000H	24K	PAGE 33	C8000H
		43FFFH			CDFFFH
24K	PAGE 11	44000H	24K	PAGE 34	CE000H
		49FFFH			D3FFFH
24K	PAGE 12	4A000H	24K	PAGE 35	D4000H
		4FFFFH			D9FFFH
24K	PAGE 13	50000H	24K	PAGE 36	DA000H
		55FFFH			DFFFFH
24K	PAGE 14	56000H	24K	PAGE 37	E0000H
		5BFFFH			E5FFFH
24K	PAGE 15	5C000H	24K	PAGE 38	E6000H
		61FFFH			EBFFFH
24K	PAGE 16	62000H	24K	PAGE 39	EC000H
		67FFFH			F1FFFH
24K	PAGE 17	68000H	24K	PAGE 40	F2000H
		6DFFFH			F7FFFH
24K	PAGE 18	6E000H	24K	PAGE 41	F8000H
		73FFFH			FDFFFH
24K	PAGE 19	74000H	2K	USER BUFFER	FE000H
		79FFFH			FE7FFFH
24K	PAGE 20	7A000H	2K	USER STACK	FE800H
		7FFFFH			FEFFFFH
24K	PAGE 21	80000H	2K	SYSTEM BUFFER	FF000H
		85FFFH			FF7FFFH
24K	PAGE 22	86000H	2K	SYSTEM STACK	FF800H
		8BFFFH			FFFFFFH

## PAGE MEMORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

32K	MONITOR PROGRAM U1	00000H 07FFFFH
32K	U2	08000H 0FFFFH
32K	U3	10000H 17FFFFH
32K	U4	18000H 1FFFFH 20000H
128K	U5	3FFFFH 40000H
128K	U6	5FFFFH 60000H
128K	U7	7FFFFH 80000H
128K	U8	9FFFFH A0000H
128K	U9	BFFFFH C0000H
128K	U10	DFFFFH E0000H
128K	U11	FFFFFH

### MEMORY DECODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERNAL I/O REGISTERS		COH
		3FH
8255 (ADC,DAC)	PA = 40H PB = 41H PC = 42H PCC = 43H	40H
		80H
82C55 (SYSTEM)	PA = 80H PB = 81H PC=82H PCC = 83H	9FH
RTC 6242	AOH - AFH	AOH
		BFH
LCD PORT	WRITE INS = C0H WRITE DATA = C2H READ BUSY = C4H	COH
		DFH
WATCH DOG		EOH
ENABLE = E0H		
		FFH

### I/O MAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

E2 MRBQ OR	E1		C		BA1		A		< UI2 (74LS138)												
	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	
U1 32K LOW	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
32K HIGH	0	0	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
U2 32K	0	0	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
U3 32K	0	0	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
U4 32K	0	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	

\* 32K LOW = REMOTE, 32K HIGH = MONITOR PROGRAM

E1,E2 MRBQ	C		B		A		A16		A15		A14		A13		A12		A11		A10		A9		A8		A7		A6		A5		A4		A3		A2		A1		A0	
	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0																				
U5 128K	0	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
U6 128K	0	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
U7 128K	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
U8 128K	1	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
U9 128K	1	0	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
U10 128K	1	1	0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		
U11 128K	1	1	1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		

\* CHANGE ACCORDING TO OPERATION

### ตารางแสดงการถอดรหัสหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1	2	3	4	5	6	7	8	9	10	11	
A			9	7	5	3	1	67	65	63	61	
B		10	11	8	6	4	2	68	66	64	62	60
C		12	13								59	58
D		14	15								57	56
E		16	17								55	54
F		18	19								53	52
G		20	21								51	50
H		22	23								49	48
J		24	25								47	46
K		26	28	30	32	34	36	38	40	42	45	44
L		27	29	31	33	35	37	39	41	43		

## SOCKET Z80180 (มองจากด้านบน)

- บนตัว Z80180 จะจัดขาแบบ PLCC 68 PIN

- SOCKET ที่ลง PCB จะจัดขาแบบ PGA 68 PIN

PLCC	PGA	PLCC	PGA	PLCC	PGA	PLCC	PGA
1	A6	21	G2	41	L9	61	A10
2	B6	22	H1	42	K9	62	B10
3	A5	23	H2	43	L10	63	A9
4	B5	24	J1	44	K11	64	B9
5	A4	25	J2	45	K10	65	A8
6	B4	26	K1	46	J11	66	B8
7	A3	27	L2	47	J10	67	A7
8	B3	28	K2	48	H11	68	B7
9	A2	29	L3	49	H10		
10	B1	30	K3	50	G11		
11	B2	31	L4	51	G10		
12	C1	32	K4	52	F11		
13	C2	33	L5	53	F10		
14	D1	34	K5	54	E11		
15	D2	35	L6	55	E10		
16	E1	36	K6	56	D11		
17	E2	37	L7	57	D10		
18	F1	38	K7	58	C11		
19	F2	39	L8	59	C10		
20	G1	40	K8	60	B11		

## ตารางเปรียบเทียบขา PLCC 68 กับ PGA 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ง

**ASCII CODE TABLE**

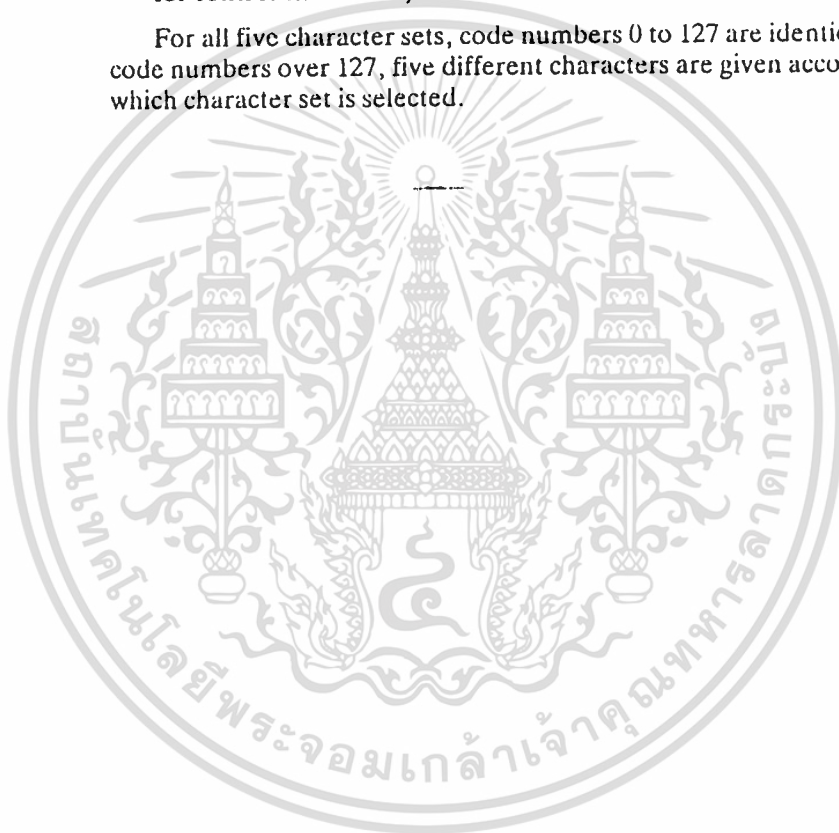
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ASCII code tables



- These tables list, for all the codes the printer can interpret:
- the code number in decimal
  - the code number in hexadecimal
  - the character that the code represents
  - for control characters, the sequence that you can type at the keyboard to enter the character
  - for control characters, the standard ASCII name for each character.

For all five character sets, code numbers 0 to 127 are identical. For code numbers over 127, five different characters are given according to which character set is selected.



decimal	hex	character	keyboard	ASCII name
0	00	<i>NUL</i>	Ctrl @	null
1	01	<i>SOH</i>	Ctrl A	start of heading
2	02	<i>STX</i>	Ctrl B	start of text
3	03	<i>ETX</i>	Ctrl C	end of text
4	04	<i>EOT</i>	Ctrl D	end of transmission
5	05	<i>ENQ</i>	Ctrl E	enquiry
6	06	<i>ACK</i>	Ctrl F	acknowledge
7	07	<i>BEL</i>	Ctrl G	bell
8	08	<i>BS</i>	Ctrl H	back space
9	09	<i>HT</i>	Ctrl I	tab horizontally
10	0A	<i>LF</i>	Ctrl J	line feed
11	0B	<i>VT</i>	Ctrl K	tab vertically
12	0C	<i>FF</i>	Ctrl L	form feed
13	0D	<i>CR</i>	Ctrl M	carriage return
14	0E	<i>SO</i>	Ctrl N	shift out
15	0F	<i>SI</i>	Ctrl O	shift in
16	10	<i>DLE</i>	Ctrl P	data link escape
17	11	<i>DC1</i>	Ctrl Q	device control 1
18	12	<i>DC2</i>	Ctrl R	device control 2
19	13	<i>DC3</i>	Ctrl S	device control 3
20	14	<i>DC4</i>	Ctrl T	device control 4
21	15	<i>NAK</i>	Ctrl U	negative acknowledge
22	16	<i>SYN</i>	Ctrl V	synchronous idle
23	17	<i>ETB</i>	Ctrl W	end of transmitted block
24	18	<i>CAN</i>	Ctrl X	cancel line
25	19	<i>EM</i>	Ctrl Y	end of medium
26	1A	<i>SUB</i>	Ctrl Z	substitute
27	1B	<i>ESC</i>	Ctrl [	escape
28	1C	<i>FS</i>	Ctrl \	file separator
29	1D	<i>GS</i>	Ctrl ]	group separator
30	1E	<i>RS</i>	Ctrl ^	record separator
31	1F	<i>US</i>	Ctrl _	unit separator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

decimal	hex	character	decimal	hex	character
32	20	SP	64	40	@
33	21	!	65	41	A
34	22	"	66	42	B
35	23	#	67	43	C
36	24	\$	68	44	D
37	25	%	69	45	E
38	26	&	70	46	F
39	27	'	71	47	G
40	28	(	72	48	H
41	29	)	73	49	I
42	2A	*	74	4A	J
43	2B	+	75	4B	K
44	2C	,	76	4C	L
45	2D	-	77	4D	M
46	2E	.	78	4E	N
47	2F	/	79	4F	O
48	30	0	80	50	P
49	31	1	81	51	Q
50	32	2	82	52	R
51	33	3	83	53	S
52	34	4	84	54	T
53	35	5	85	55	U
54	36	6	86	56	V
55	37	7	87	57	W
56	38	8	88	58	X
57	39	9	89	59	Y
58	3A	:	90	5A	Z
59	3B	;	91	5B	[
60	3C	<	92	5C	\
61	3D	=	93	5D	]
62	3E	>	94	5E	^
63	3F	?	95	5F	_

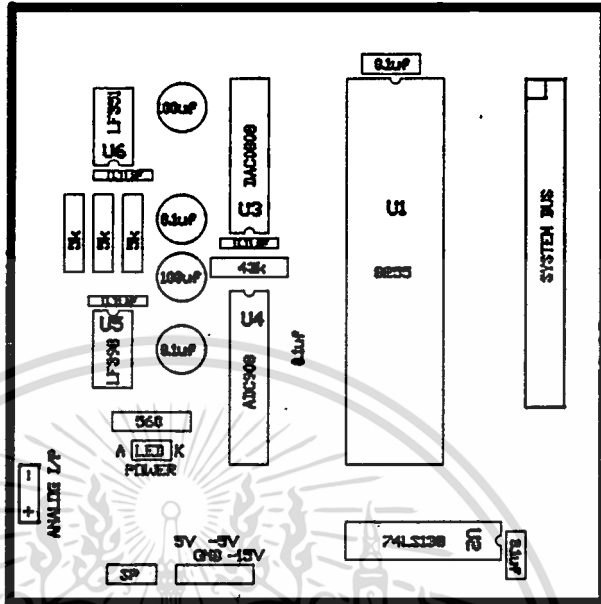
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

decimal	hex	character	decimal	hex	character
96	60	`	112	70	p
97	61	a	113	71	q
98	62	b	114	72	r
99	63	c	115	73	s
100	64	d	116	74	t
101	65	e	117	75	u
102	66	f	118	76	v
103	67	g	119	77	w
104	68	h	120	78	x
105	69	i	121	79	y
106	6A	j	122	7A	z
107	6B	k	123	7B	{
108	6C	l	124	7C	
109	6D	m	125	7D	}
110	6E	n	126	7E	~
111	6F	o	127	7F	DEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

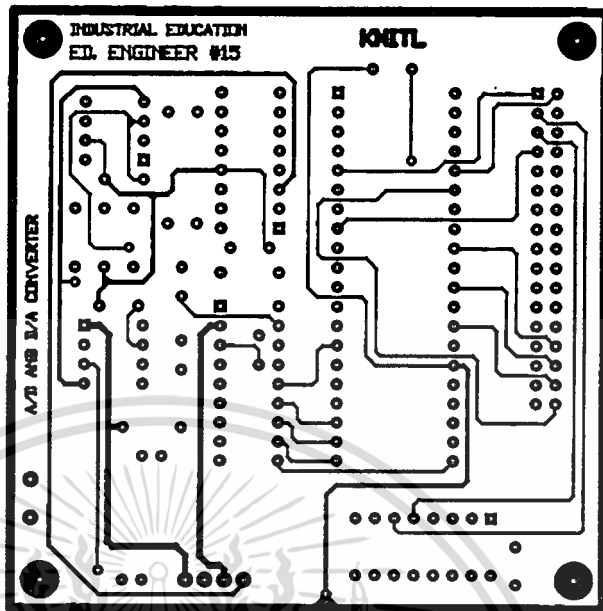


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



908 Top Overlay

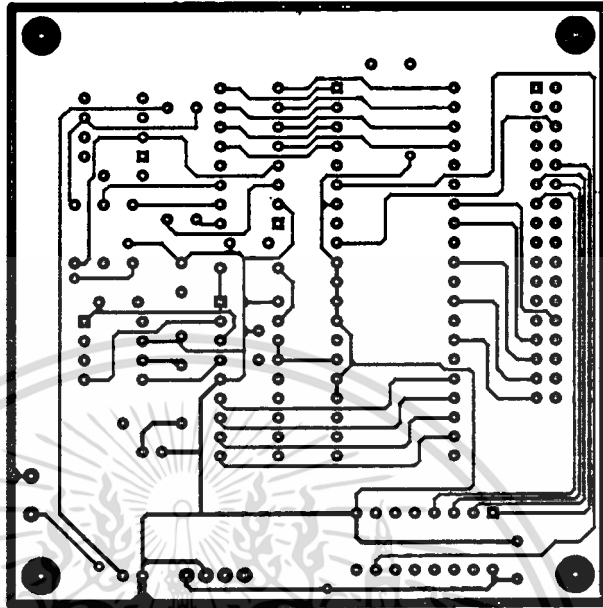
**แสดงการวางอุปกรณ์ของแผงวงจรแปลงสัญญาณ (A/D and D/A Board)**



908 Top Layer

แสดงลายวงจรด้านบนของแผ่นวงจรแปลงสัญญาณ (A/D and D/A Board)

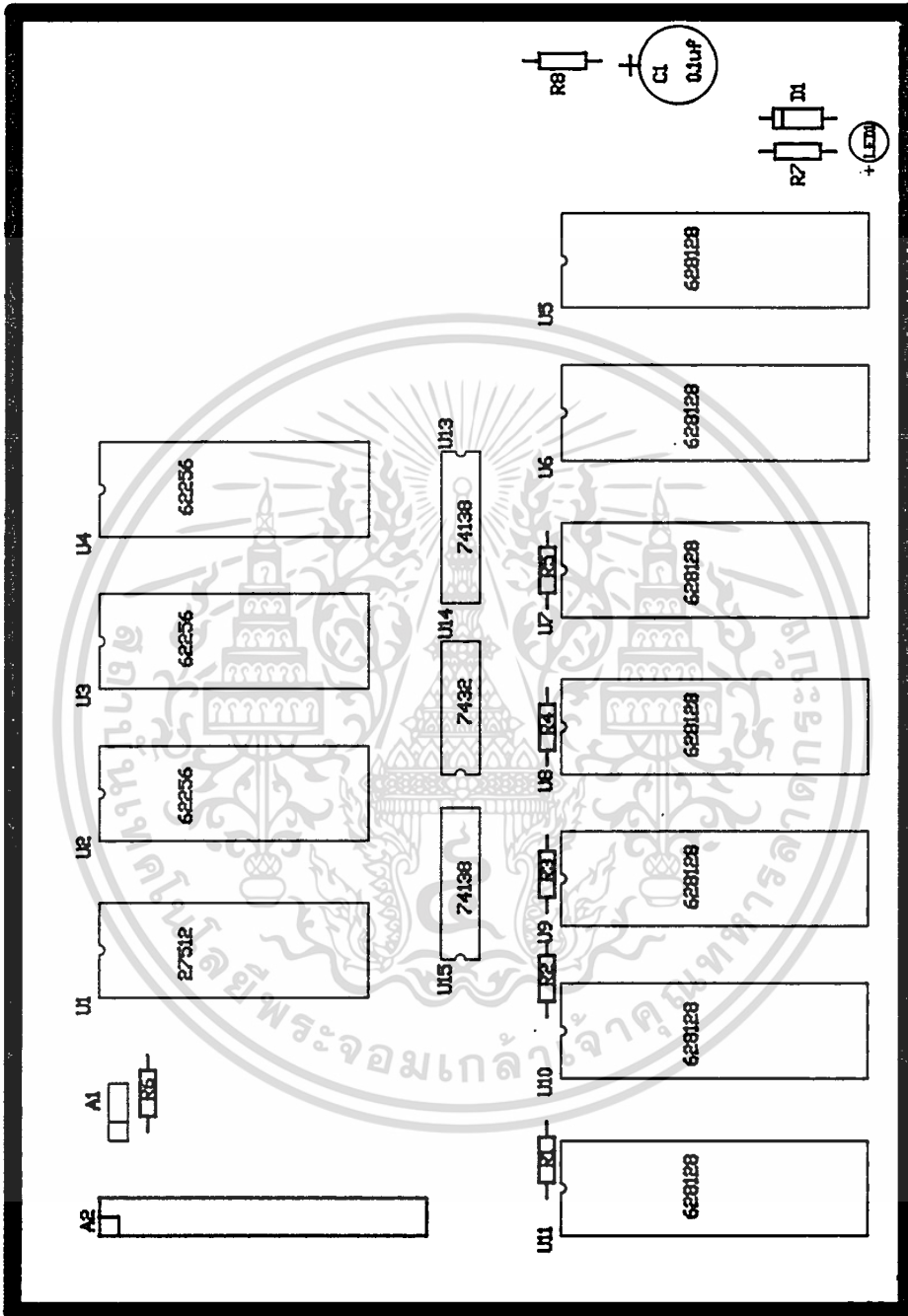
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



908 Bottom Layer

**แสดงลายวงจรด้านล่างของแผงวงจรแปลงสัญญาณ (A/D and D/A Board)**

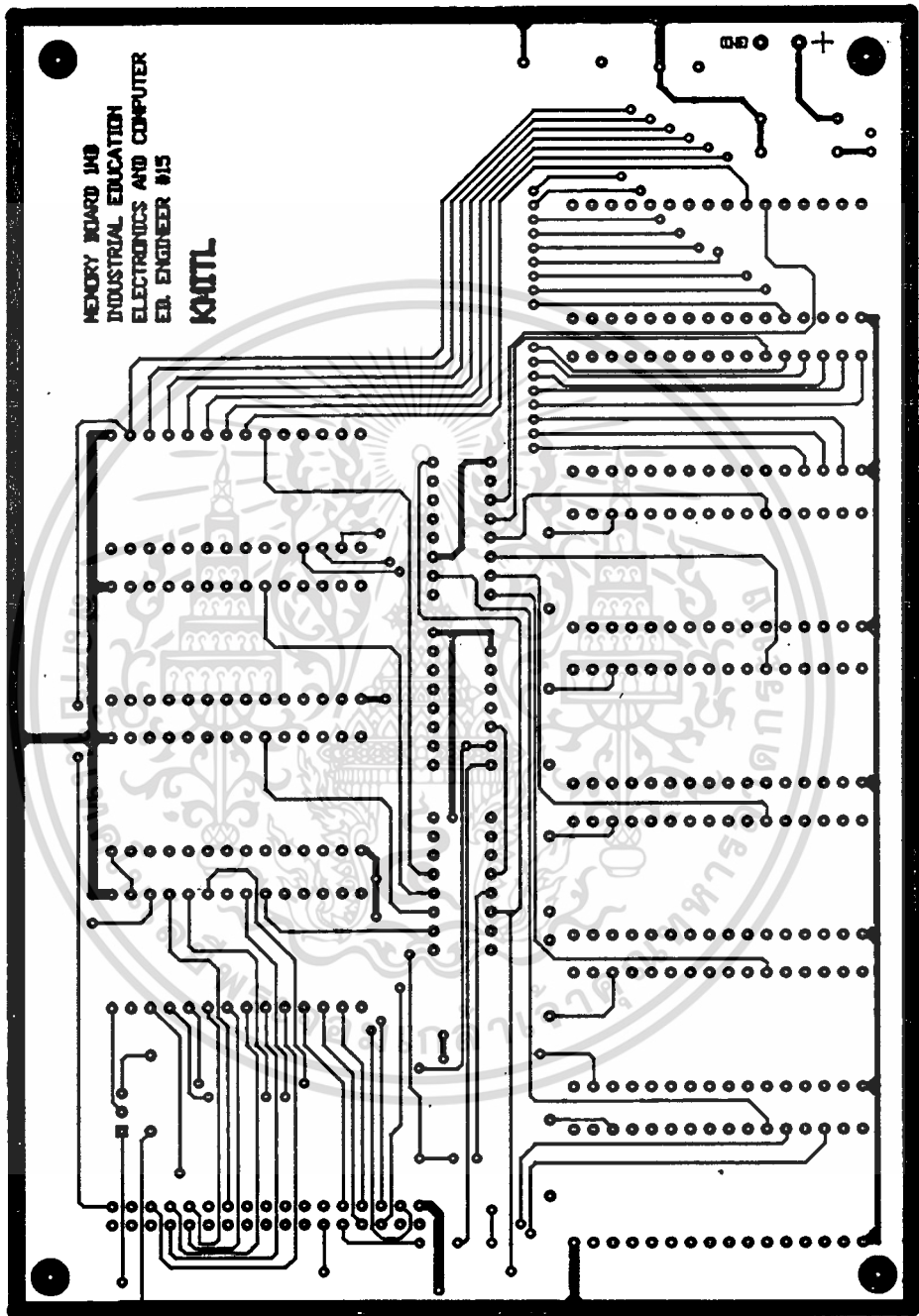
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MEMO Top Overlay

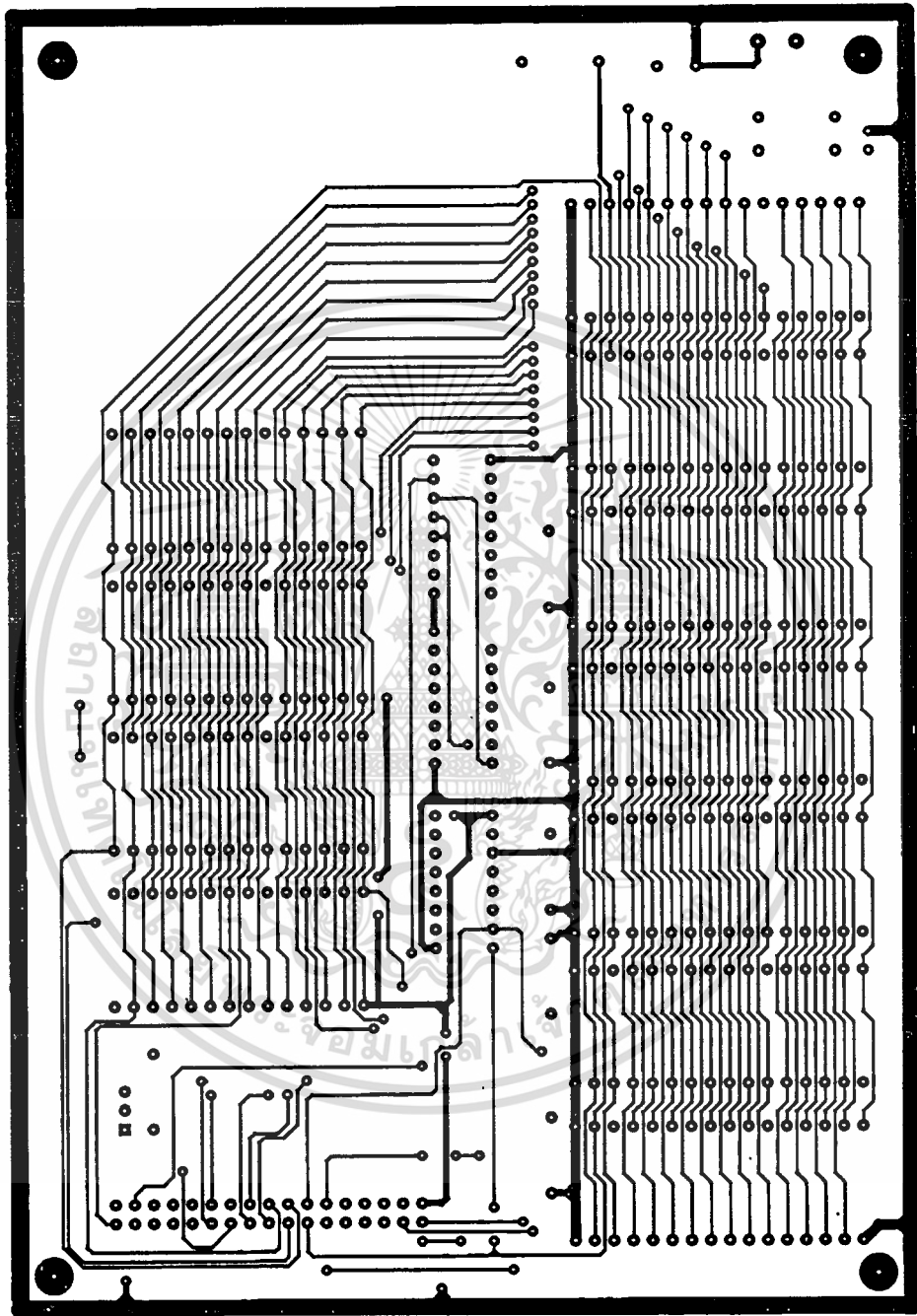
แสดงการวางอุปกรณ์ของแผ่นวงจรหน่วยความจำ (Memory Board)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แสดงลายวงจรด้านบนของแผ่นวงจรหน่วยความจำ (Memory Board)

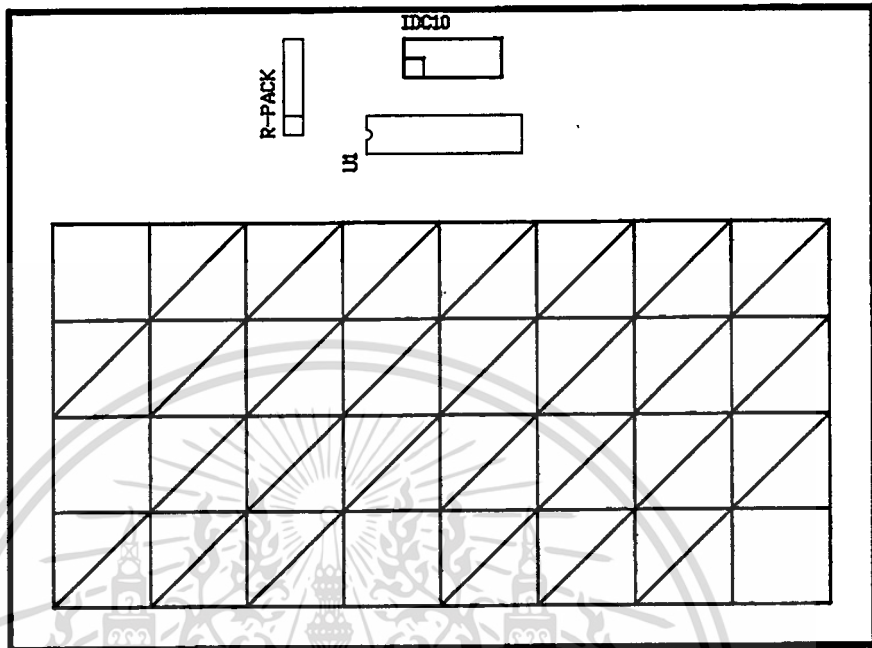
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MEMO Bottom Layer

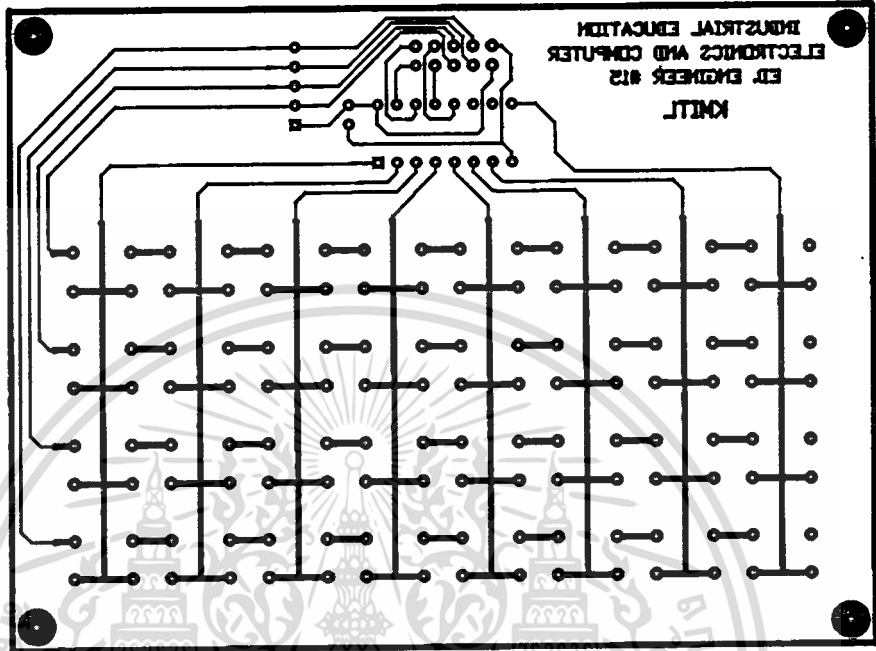
แสดงลายวงจรด้านล่างของแผ่นวงจรหน่วยความจำ (Memory Board)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



KEYBOARD Top Overlay

แสดงการวางอุปกรณ์ของแผงวงจรคีย์บอร์ด (Keyboard Board)



KEYBOARD Bottom Layer

แสดงลายวงจรด้านล่างของแผงวงจรคีย์บอร์ด (Keyboard Board)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## CHARACTER FONT TABLE

Address (Hex)	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
XXXX0000	CGRAM (1)												
XXXX0001	(2)	[Character patterns]											
XXXX0010	(3)	[Character patterns]											
XXXX0011	(4)	[Character patterns]											
XXXX0100	(5)	[Character patterns]											
XXXX0101	(6)	[Character patterns]											
XXXX0110	(7)	[Character patterns]											
XXXX0111	(8)	[Character patterns]											
XXXX1000	(9)	[Character patterns]											
XXXX1001	(10)	[Character patterns]											
XXXX1010	(11)	[Character patterns]											
XXXX1011	(12)	[Character patterns]											
XXXX1100	(13)	[Character patterns]											
XXXX1101	(14)	[Character patterns]											
XXXX1110	(15)	[Character patterns]											
XXXX1111	(16)	[Character patterns]											

NOTE: CGRAM is a CHARACTER GENERATOR RAM having a storage function of character pattern which enable to change freely by user's program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## MAXIMUM RATINGS

### Electric maximum ratings

Item	Symbol	Min.	Max.	Unit	Remarks
Power supply for logic	$V_{CC} - V_{CC}$			V	
Power supply for LCD drive	$V_{CD} - V_{CD}$			V	Refer to individual specifications
Input voltage	$V_i$			V	
Static electricity			100	V	See note

**Note** Electro-static discharge resistance is tested by charging a condenser with a capacity of 200pF and discharging it by contact with an interface connector pin.

### Environmental conditions

Item	Operating		Non-operating		Remarks
	Min.	Max.	Min.	Max.	
Ambient temperature	Refer to individual specifications				
Humidity	Note				
Vibration		4.9m/s <sup>2</sup> (0.5G)		19.6 m/s <sup>2</sup> (2G)	No dew
Shock		29.4 m/s <sup>2</sup> (3G)		490 m/s <sup>2</sup> (50G)	XYZ 3 directions
Corrosion gas	No corrosion gas				

**Note** Humidity conditions are as follows.

Number of dots	Under 128 x 240		128 x 240 or over	
	Ambient temperature (Ta)	95% RH max.		85% RH max
Ta ≤ 40°C	Below maximum absolute humidity of 40°C 95% RH		Below maximum absolute humidity of 40°C 85% RH	
Ta > 40°C (Below maximum temperature)				

## RELIABILITY CONDITIONS

LCD MODULE (Consumer Type)		
Item	Conditions	Evaluation
High Temperature Operation	Operating 96 - 100 Hrs at 50 ± 2°C surrounding temp.	No change is visible appearance nor function
Low Temperature Operation	Operating 96 - 100 Hrs at 0 ± 2°C surrounding temp.	
High Temperature Storage	Storage 96 - 100 Hrs at 60 ± 2°C surrounding temp. then storage 4 Hrs at normal condition (Power Off)	
Low Temperature Storage	Storage 96 - 100 Hrs at -20 ± 2°C surrounding temp. then storage 4 Hrs at normal condition (Power Off) No dew to be found	
Demo Proof	Storage 96 - 100 Hrs at 40 ± 2°C and 80 - 95% RH surrounding condition, then storage 4 Hrs at normal condition (Power Off) No dew to be found.	

**NOTE** The above condition is only representative, and may, differ in case of customized specifications.

## OPTICAL DATA

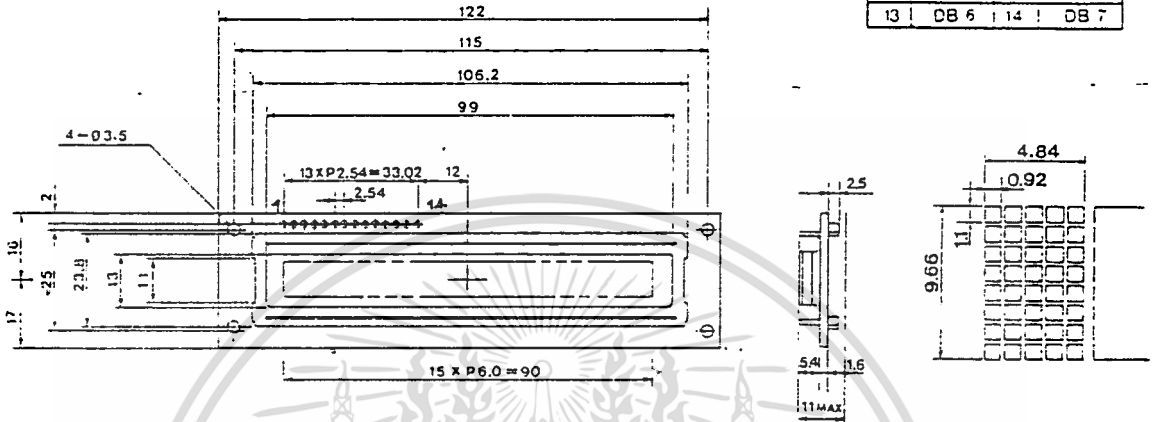
Ta = 25°C

Item	Symbol	Condition	Min.	Typ.	Max.	Unit	Notes to see
Viewing angle	$\phi 2 - \phi 1$	K = 1 ±	—	20	—	deg.	—
Contrast ratio	K	$\phi = 25^\circ$ $\theta = 0^\circ$	—	2	—	—	—
Response time (rise)	$t_r$	$\phi = 25^\circ$ $\theta = 0^\circ$	—	250	400	ms	—
Response time (fall)	$t_f$	$\phi = 25^\circ$ $\theta = 0^\circ$	—	150	250	ms	—
				250	400		
				150	250		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

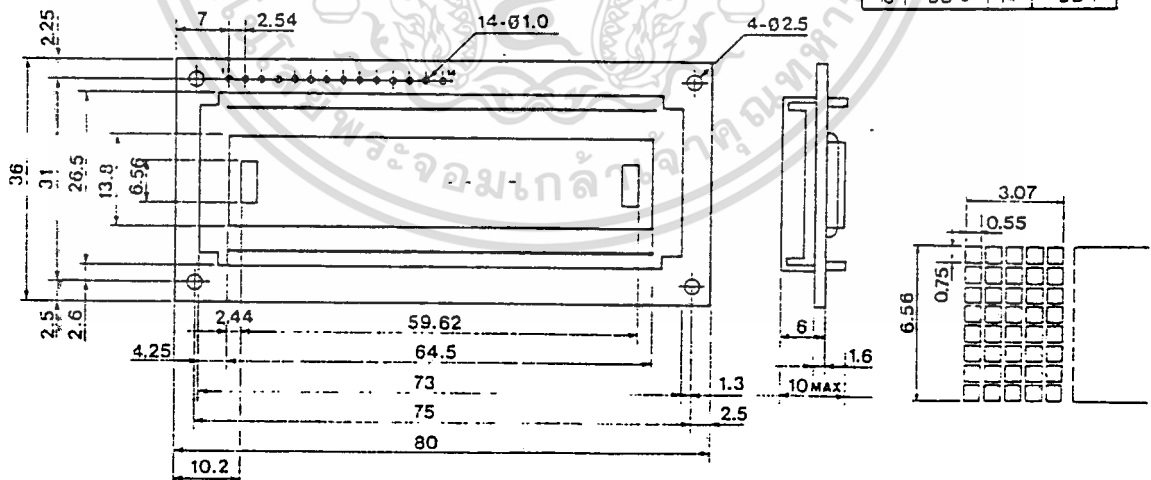
DV-1601B

No.	Signal	No.	Signal
1	VSS	2	VCC
3	VO	4	RS
5	R/W	6	E
7	DB 0	8	DB 1
9	DB 2	10	DB 3
11	DB 4	12	DB 5
13	DB 6	14	DB 7



DV-1601

No.	Signal	No.	Signal
1	VSS	2	VDD
3	VO	4	RS
5	R/W	6	E
7	DB 0	8	DB 1
9	DB 2	10	DB 3
11	DB 4	12	DB 5
13	DB 6	14	DB 7



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 27256 256K (32K x 8) PRODUCTION AND UV ERASABLE PROMS

- **New Quick-Pulse Programming™ Algorithm for Plastic P27256**
  - 4 Second Programming
  - Intelligent Programming™ Algorithm Compatible
- **Fast Access Time**
  - 170 ns D27256-1
  - 200 ns P27256-2
- **Intelligent Identifier™ Mode**
- **Plastic Production P27256 is Compatible with Auto-insertion Equipment**
- **Moisture Resistant**
- **Industry Standard Pinout ... JEDEC Approved ... 28 Lead Cerdip and Plastic Package**  
(See Packaging Spec. Order # 211009)

The Intel 27256 is a 5V only, 262,144-bit Ultraviolet Erasable (Cerdip)/plastic production (P27256) electrically programmable read-only memory (EPROM). Organized as 32K words by 8 bits, individual bytes can be accessed in less than 170 ns (27256-1). This is compatible with high performance microprocessors, such as the Intel iAPX 186, allowing full speed operation without the addition of performance-degrading WAIT states. The 27256 is also directly compatible with Intel's 8051 family of microcontrollers.

The Plastic P27256 is ideal for high volume production environments where code flexibility is crucial. Plastic packaging is also well-suited to auto-insertion equipment in cost-effective automated assembly lines. Intel's new Quick-Pulse Programming Algorithm enables the P27256 to be programmed within four seconds (plus programmer overhead). Programming equipment which takes advantage of this innovation will electronically identify the EPROM with the help of the Intelligent Identifier and rapidly program it using a superior programming method. The Intelligent Programming Algorithm may be utilized in the absence of such equipment.

The 27256 enables implementation of new, advanced systems with firmware-intensive architectures. The combination of the 27256's high-density, cost-effective EPROM storage, and new advanced microprocessors having megabit addressing capability provides designers with opportunities to engineer user-friendly, high reliability, high-performance systems.

The 27256's large storage capability of 32 K-bytes enables it to function as a high-density software carrier. Entire operating systems, diagnostics, high-level language programs and specialized application software can reside in a 27256 EPROM directly on a system's memory bus. This permits immediate microprocessor access and execution of software and eliminates the need for time-consuming disk accesses and downloads.

Two-line control and JEDEC-approved, 28-pin packaging are standard features of all Intel high-density EPROMs. This assures easy microprocessor interfacing and minimum design efforts when upgrading, adding, or choosing between nonvolatile memory alternatives.

The 27256 is manufactured using Intel's advanced HMOS®II-E technology.

\*HMOS is a patented process of Intel Corporation.

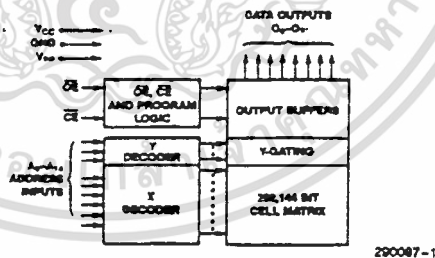


Figure 1. Block Diagram

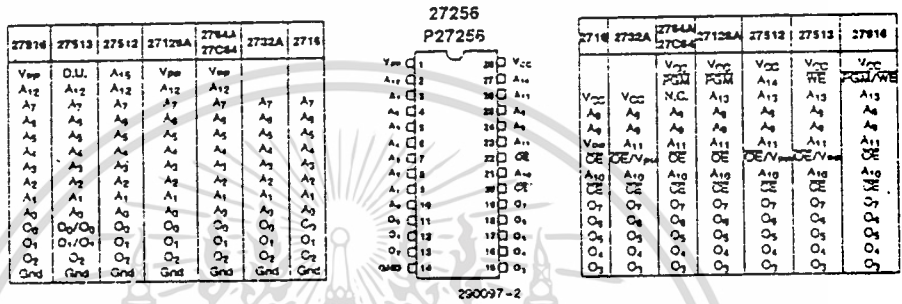
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



27256

Pin Names

A <sub>0</sub> -A <sub>14</sub>	Addresses
CE	Chip Enable
OE	Output Enable
O <sub>0</sub> -O <sub>7</sub>	Outputs
D.U.	Don't Use
WE	Write Enable



NOTE:  
Intel "Universal Site" Compatible EPROM pin configurations are shown in the blocks adjacent to the P27256 pins.  
Figure 2. Cardip/Plastic DIP Pin Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



National  
Semiconductor  
Corporation

## DAC0808, DAC0807, DAC0806 8-Bit D/A Converters

### General Description

The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with  $\pm 5V$  supplies. No reference current ( $I_{REF}$ ) trimming is required for most applications since the full scale output current is typically  $\pm 1$  LSB of  $255 I_{REF} / 256$ . Relative accuracies of better than  $\pm 0.19\%$  assure 8-bit monotonicity and linearity while zero level output current of less than  $4 \mu A$  provides 8-bit zero accuracy for  $I_{REF} \geq 2$  mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

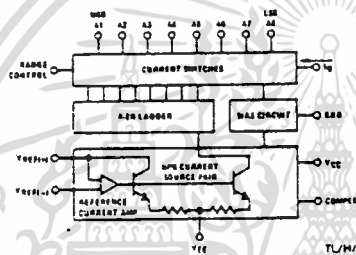
The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the

MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

### Features

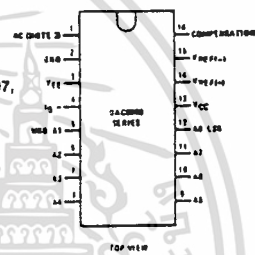
- Relative accuracy:  $\pm 0.19\%$  error maximum (DAC0808)
- Full scale current match:  $\pm 1$  LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying in/out slew rate: 8 mA/ $\mu s$
- Power supply voltage range:  $\pm 4.5V$  to  $\pm 18V$
- Low power consumption, 33 mW @  $\pm 5V$

### Block and Connection Diagrams

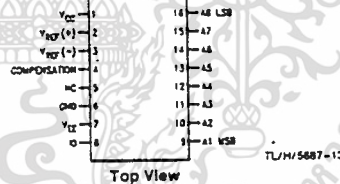


Order Number  
DAC0808, DAC0807,  
or DAC0806  
See NS Package  
Number J16A,  
M16A or N16A

#### Dual-In-Line Package



#### Small-Outline Package



### Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS		
		J PACKAGE (J16A)*	N PACKAGE (N16A)*	SO PACKAGE (M16A)
8-bit	$-55^{\circ}C \leq T_A \leq +125^{\circ}C$	DAC0808LJ	MC1508L8	
8-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$	DAC0808LCJ	MC1408L8	DAC0808LCM
7-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$	DAC0807LCJ	MC1408L7	DAC0807LCM
6-bit	$0^{\circ}C \leq T_A \leq +75^{\circ}C$	DAC0806LCJ	MC1408L6	DAC0806LCM

\*Note: Devices may be ordered by using either order number.



# LM386 Low Voltage Audio Power Amplifier

## General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value up to 200.

The inputs are ground referenced while the output is automatically biased to one half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion
- Eight pin dual-in-line package

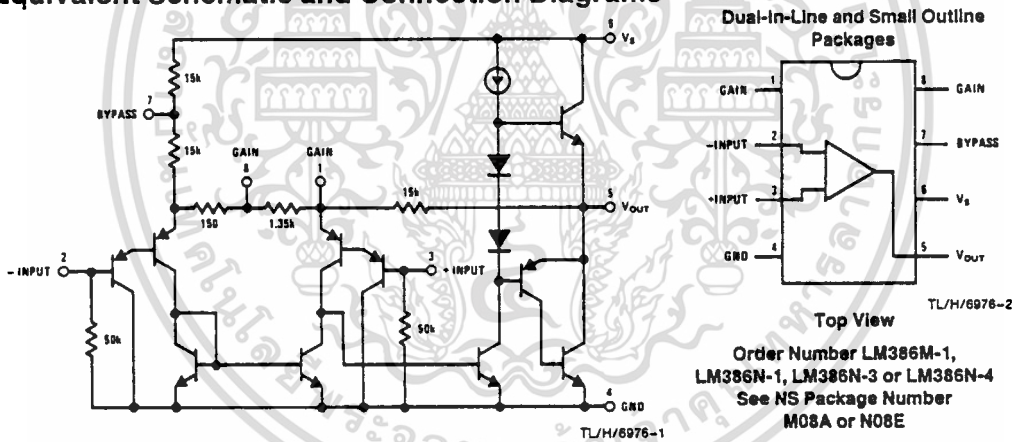
## Applications

- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

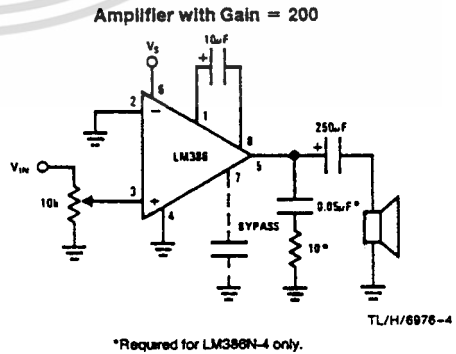
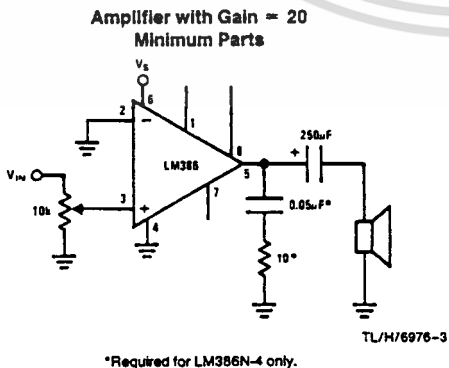
## Features

- Battery operation
- Minimum external parts
- Wide supply voltage range 4V-12V or 5V-18V
- Low quiescent current drain 4 mA

## Equivalent Schematic and Connection Diagrams



## Typical Applications



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Supply Voltage (LM386N-1, -3, LM386M-1)	15V
Supply Voltage (LM386N-4)	22V
Package Dissipation (Note 1) (LM386N-4)	1.25W
Input Voltage	±0.4V
Storage Temperature	-65°C to +150°C
Operating Temperature	0°C to +70°C

Junction Temperature	+150°C
Soldering Information	
Dual-In-Line Package	
Soldering (10 sec)	+260°C
Small Outline Package	
Vapor Phase (60 sec)	+215°C
Infrared (15 sec)	+220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

## Electrical Characteristics $T_A = 25^\circ\text{C}$

Parameter	Conditions	Min	Typ	Max	Units
Operating Supply Voltage ( $V_S$ )					
LM386N-1, -3, LM386M-1		4		12	V
LM386N-4		5		18	V
Quiescent Current ( $I_Q$ )	$V_S = 6V, V_{IN} = 0$		4	8	mA
Output Power ( $P_{OUT}$ )					
LM386N-1, LM386M-1	$V_S = 6V, R_L = 8\Omega, THD = 10\%$	250	325		mW
LM386N-3	$V_S = 9V, R_L = 8\Omega, THD = 10\%$	500	700		mW
LM386N-4	$V_S = 16V, R_L = 32\Omega, THD = 10\%$	700	1000		mW
Voltage Gain ( $A_V$ )	$V_S = 6V, f = 1\text{ kHz}$ $10\ \mu\text{F}$ from Pin 1 to 8		26	46	dB
Bandwidth (BW)	$V_S = 6V$ , Pins 1 and 8 Open		300		kHz
Total Harmonic Distortion (THD)	$V_S = 6V, R_L = 8\Omega, P_{OUT} = 125\text{ mW}$ $f = 1\text{ kHz}$ , Pins 1 and 8 Open		0.2		%
Power Supply Rejection Ratio (PSRR)	$V_S = 6V, f = 1\text{ kHz}, C_{BYPASS} = 10\ \mu\text{F}$ Pins 1 and 8 Open, Referred to Output		50		dB
Input Resistance ( $R_{IN}$ )			50		k $\Omega$
Input Bias Current ( $I_{BIAS}$ )	$V_S = 6V$ , Pins 2 and 3 Open		250		nA

Note 1: For operation in ambient temperatures above 25°C, the device must be derated based on a 150°C maximum junction temperature and 1) a thermal resistance of 80°C/W junction to ambient for the dual-in-line package and 2) a thermal resistance of 170°C/W for the small outline package.

## Application Hints

### GAIN CONTROL

To make the LM386 a more versatile amplifier, two pins (1 and 8) are provided for gain control. With pins 1 and 8 open the 1.35 k $\Omega$  resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35 k $\Omega$  resistor, the gain will go up to 200 (46 dB). If a resistor is placed in series with the capacitor, the gain can be set to any value from 20 to 200. Gain control can also be done by capacitively coupling a resistor (or FET) from pin 1 to ground.

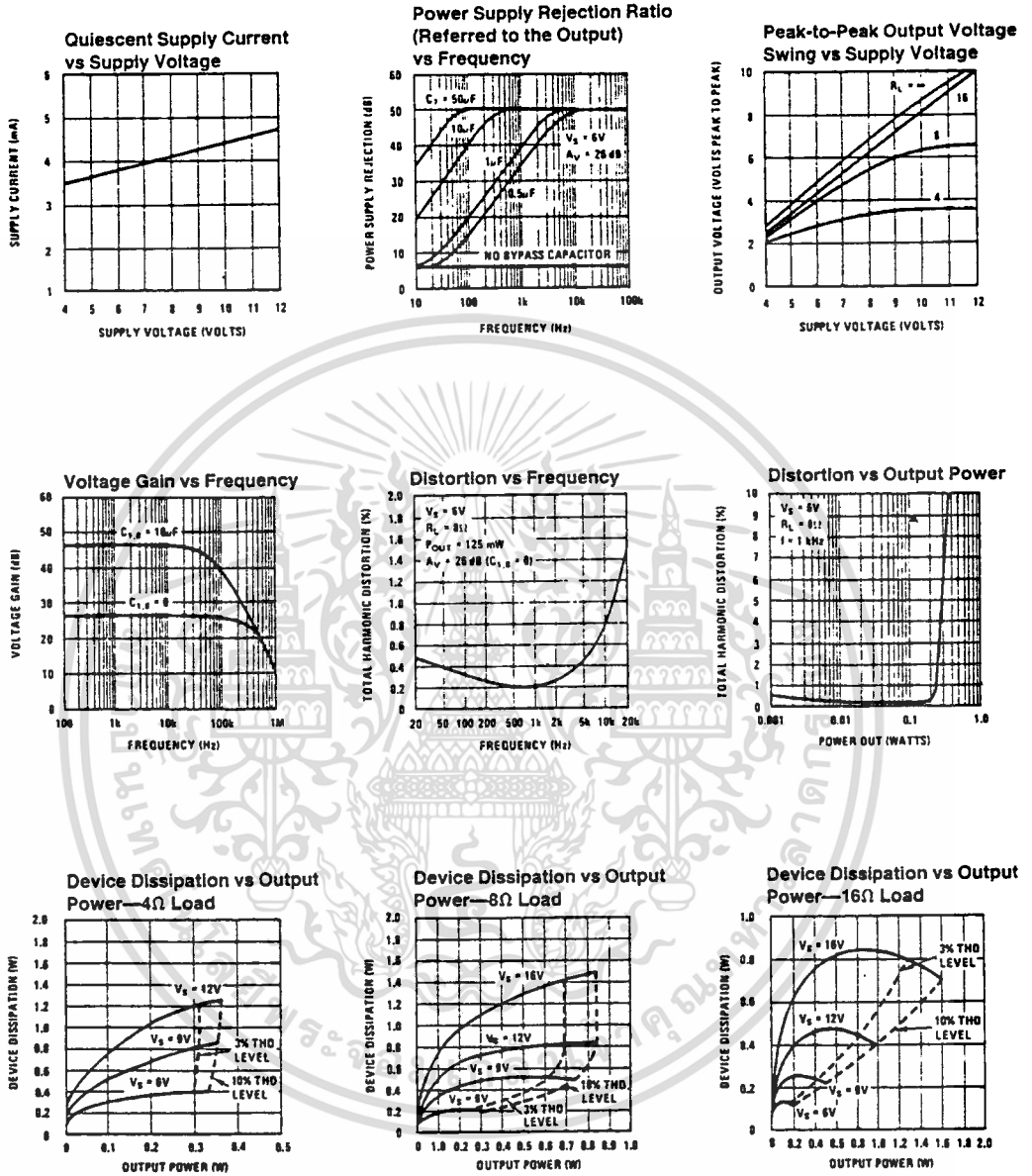
Additional external components can be placed in parallel with the internal feedback resistors to tailor the gain and frequency response for individual applications. For example, we can compensate poor speaker bass response by frequency shaping the feedback path. This is done with a series RC from pin 1 to 5 (paralleling the internal 15 k $\Omega$  resistor). For 6 dB effective bass boost:  $R = 15\text{ k}\Omega$ , the lowest value for good stable operation is  $R = 10\text{ k}\Omega$  if pin 8 is open. If pins 1 and 8 are bypassed then  $R$  as low as 2 k $\Omega$  can be used. This restriction is because the amplifier is only compensated for closed-loop gains greater than 9.

### INPUT BIASING

The schematic shows that both inputs are biased to ground with a 50 k $\Omega$  resistor. The base current of the input transistors is about 250 nA, so the inputs are at about 12.5 mV when left open. If the dc source resistance driving the LM386 is higher than 250 k $\Omega$  it will contribute very little additional offset (about 2.5 mV at the input, 50 mV at the output). If the dc source resistance is less than 10 k $\Omega$ , then shorting the unused input to ground will keep the offset low (about 2.5 mV at the input, 50 mV at the output). For dc source resistances between these values we can eliminate excess offset by putting a resistor from the unused input to ground, equal in value to the dc source resistance. Of course all offset problems are eliminated if the input is capacitively coupled.

When using the LM386 with higher gains (bypassing the 1.35 k $\Omega$  resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a 0.1  $\mu\text{F}$  capacitor or a short to ground depending on the dc source resistance on the driven input.

### Typical Performance Characteristics

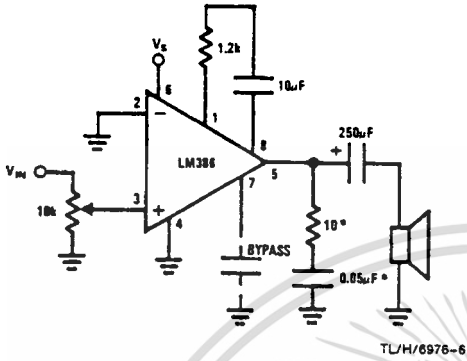


TL/H/6976-5

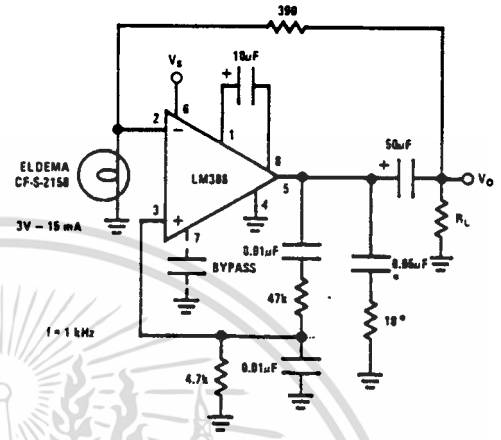
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

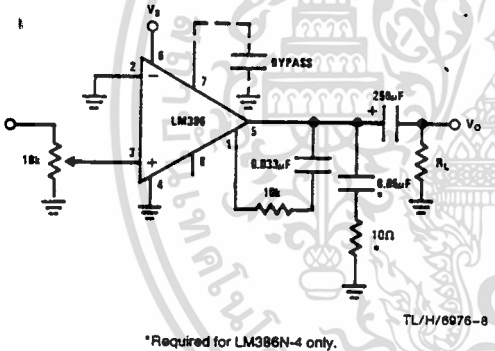
Amplifier with Gain = 50



Low Distortion Power Wienbridge Oscillator

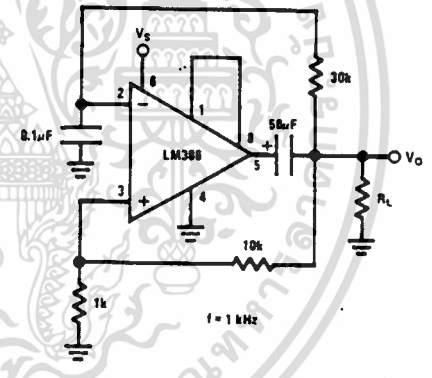


Amplifier with Bass Boost

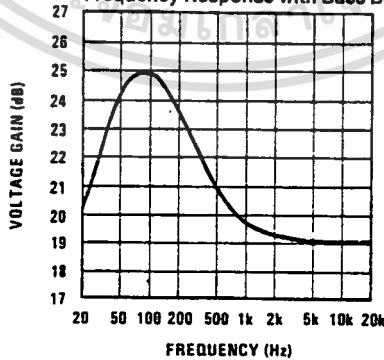


\*Required for LM386N-4 only.

Square Wave Oscillator



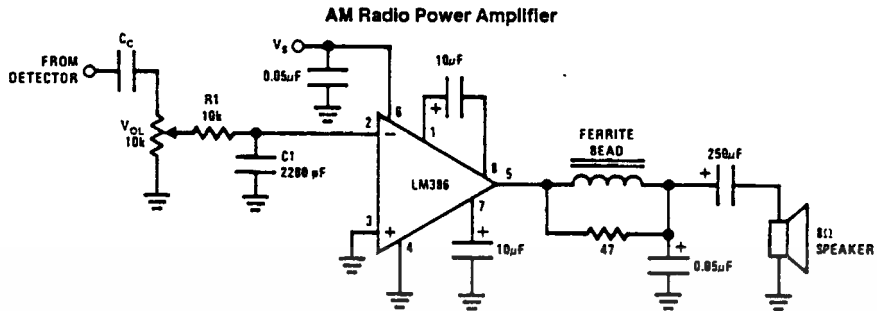
Frequency Response with Bass Boost



TL/H/6976-10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Typical Applications (Continued)



**Note 1:** Twist supply lead and supply ground very tightly.

**Note 2:** Twist speaker lead and ground very tightly.

**Note 3:** Ferrite bead is Ferroxcube K5-001-001/3B with 3 turns of wire.

**Note 4:** R1C1 band limits input signals.

**Note 5:** All components must be spaced very close to IC.

TL/H/6976-11



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MAXIM

## +5V Powered RS-232 Drivers/Receivers

### General Description

Maxim's family of line drivers/receivers are intended for all RS-232 and V.28/V.24 communications interfaces, and in particular, for those applications where  $\pm 12V$  is not available. The MAX230, MAX236, MAX240 and MAX241 are particularly useful in battery powered systems since their low power shutdown mode reduces power dissipation to less than  $5\mu W$ . The MAX233 and MAX235 use no external components and are recommended for applications where printed circuit board space is critical.

All members of the family except the MAX231 and MAX239 need only a single +5V supply for operation. The RS-232 drivers/receivers have on-board charge pump voltage converters which convert the +5V input power to the  $\pm 10V$  needed to generate the RS-232 output levels. The MAX231 and MAX239, designed to operate from +5V and +12V, contain a +12V to -12V charge pump voltage converter.

Since nearly all RS-232 applications need both line drivers and receivers, the family includes both receivers and drivers in one package. The wide variety of RS-232 applications require differing numbers of drivers and receivers. Maxim offers a wide selection of RS-232 driver/receiver combinations in order to minimize the package count (see table below).

Both the receivers and the line drivers (transmitters) meet all EIA RS-232C and CCITT V.28 specifications.

### Features

- ◆ Operates from Single 5V Power Supply (+5V and +12V — MAX231 and MAX239)
- ◆ Meets All RS-232C and V.28 Specifications
- ◆ Multiple Drivers and Receivers
- ◆ Onboard DC-DC Converters
- ◆  $\pm 9V$  Output Swing with +5V Supply
- ◆ Low Power Shutdown —  $< 1\mu A$  (typ)
- ◆ 3-State TTL/CMOS Receiver Outputs
- ◆  $\pm 30V$  Receiver Input Levels

### Applications

Computers  
Peripherals  
Modems  
Printers  
Instruments

### Selection Table

Part Number	Power Supply Voltage	No. of RS-232 Drivers	No. of RS-232 Receivers	External Components	Low Power Shutdown /TTL 3-State	No. of Pins
MAX230	+5V	5	0	4 capacitors	Yes/No	20
MAX231	+5V and +7.5V to 13.2V	2	2	2 capacitors	No/No	14
MAX232	+5V	2	2	4 capacitors	No/No	16
MAX233	+5V	2	2	None	No/No	20
MAX234	+5V	4	0	4 capacitors	No/No	16
MAX235	+5V	5	5	None	Yes/Yes	24
MAX236	+5V	4	3	4 capacitors	Yes/Yes	24
MAX237	+5V	5	3	4 capacitors	No/No	24
MAX238	+5V	4	4	4 capacitors	No/No	24
MAX239	+5V and +7.5V to 13.2V	3	5	2 capacitors	No/Yes	24
MAX240	+5V	5	5	4 capacitors	Yes/Yes	44
						(Flatpak)
MAX241	+5V	4	5	4 capacitors	Yes/Yes	28
						(Small Outline)

\* Patent Pending

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## +5V Powered RS-232 Drivers/Receivers

### ABSOLUTE MAXIMUM RATINGS

$V_{CC}$ .....	-0.3V to +6V
$V^*$ .....	( $V_{CC} - 0.3V$ ) to +14V
$V^-$ .....	+0.3V to -14V
Input Voltages	
$T_{IN}$ .....	-0.3 to ( $V_{CC} + 0.3V$ )
$R_{IN}$ .....	$\pm 30V$
Output Voltages	
$T_{OUT}$ .....	( $V^* + 0.3V$ ) to ( $V^- - 0.3V$ )
$R_{OUT}$ .....	-0.3V to ( $V_{CC} + 0.3V$ )

Short Circuit Duration	$T_{OUT}$ .....	continuous
Power Dissipation		
CERDIP .....		675mW
(derate 9.5mW/°C above +70°C)		
Plastic DIP .....		375mW
(derate 7mW/°C above +70°C)		
Small Outline (SO) .....		375mW
(derate 7mW/°C above +70°C)		
Lead Temperature (soldering 10 seconds) .....		+300°C
Storage Temperature .....		-65°C to +160°C

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions above those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

### ELECTRICAL CHARACTERISTICS

(MAX232, 234, 236, 237, 238, 240, 241  $V_{CC} = 5V \pm 10\%$ ; MAX233, 235  $V_{CC} = 5V \pm 5\%$  C1-C4 = 1.0 $\mu$ F; MAX231, 239  $V_{CC} = 5V \pm 10\%$ ,  $V^* = 7.5V$  to 13.2V;  $T_A$  = Operating Temperature Range, Figures 3-14, unless otherwise noted.)

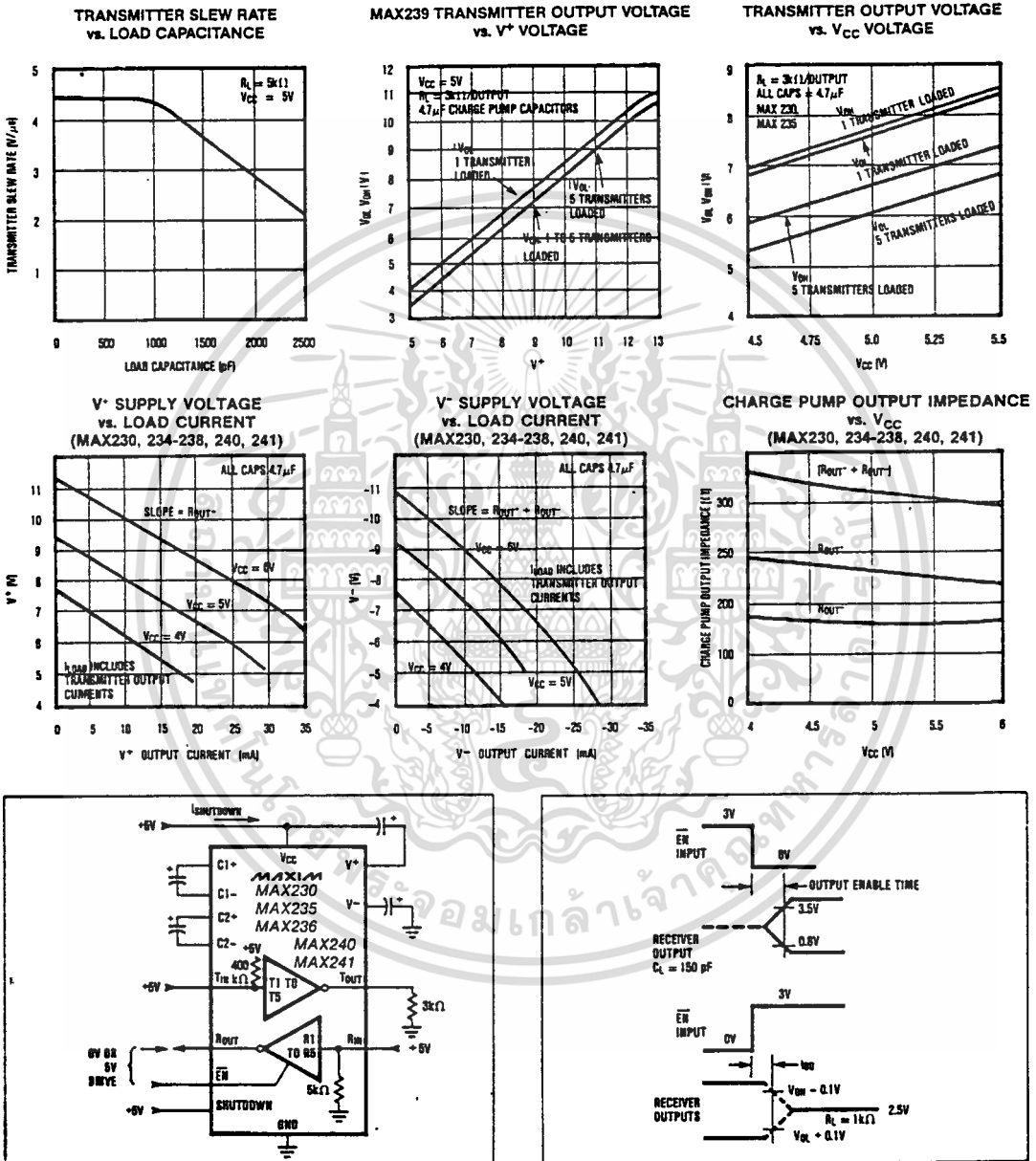
PARAMETER	CONDITIONS	MIN	TYP	MAX*	UNITS
Output Voltage Swing	All Transmitter Outputs loaded with 3k $\Omega$ to Ground	$\pm 5$	$\pm 9$		V
$V_{CC}$ Power Supply Current	No load, $T_A = +25^\circ C$ MAX232-MAX233		5	10	mA
	MAX230, MAX234-238, MAX240-MAX241		7	15	
	MAX231, MAX239		0.4	1	
$V^*$ Power Supply Current	No load, MAX231 and MAX239 only		1.8	5	mA
		MAX231	5	15	
Shutdown Supply Current	Figure 1, $T_A = +25^\circ C$		1	10	$\mu A$
Input Logic Threshold Low	$T_{IN}$ , $\overline{EN}$ , Shutdown			0.8	V
Input Logic Threshold High	$T_{IN}$	2.0			V
	$\overline{EN}$ , Shutdown	2.4			
Logic Pullup Current	$T_{IN} = 0V$		15	200	$\mu A$
RS-232 Input Voltage Operating Range		-30		+30	V
RS-232 Input Threshold Low	$V_{CC} = 5V$ , $T_A = +25^\circ C$ (MAX231, 239 $V^* = 0V$ )	0.8	1.2		V
RS-232 Input Threshold High	$V_{CC} = 5V$ , $T_A = +25^\circ C$ (MAX231, 239 $V^* = 12V$ )		1.7	2.4	V
RS-232 Input Hysteresis	$V_{CC} = 5V$	0.2	0.5	1.0	V
RS-232 Input Resistance	$T_A = +25^\circ C$ , $V_{CC} = 5V$	3	5	7	k $\Omega$
TTL/CMOS Output Voltage Low	$I_{OUT} = 1.6mA$ (MAX231-233, $I_{OUT} = 3.2mA$ )			0.4	V
TTL/CMOS Output Voltage High	$I_{OUT} = 1.0mA$	3.5			V
TTL/CMOS Output Leakage Current	$\overline{EN} = V_{CC}$ , $0V \leq R_{OUT} \leq V_{CC}$		0.05	$\pm 10$	$\mu A$
Output Enable Time (Figure 2)	MAX235, MAX236, MAX239, MAX240, MAX241		400		ns
Output Disable Time (Figure 2)	MAX235, MAX236, MAX239, MAX240, MAX241		250		ns
Propagation Delay	RS-232 to TTL		0.5		$\mu s$
Instantaneous Slew Rate	$C_L = 10pF$ , $R_L = 3-7k\Omega$ , $T_A = +25^\circ C$ (Note 1)			30	V/ $\mu s$
Transition Region Slew Rate	$R_L = 3k\Omega$ , $C_L = 2500pF$ , Measured from +3V to -3V or -3V to +3V		3		V/ $\mu s$
Output Resistance	$V_{CC} = V^* = V^- = 0V$ , $V_{OUT} = \pm 2V$	300			$\Omega$
RS-232 Output Short Circuit Current			$\pm 10$		mA

Note 1: Sample tested

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# +5V Powered RS-232 Drivers/Receivers

## Typical Operating Characteristics



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## +5V Powered RS-232 Drivers/Receivers

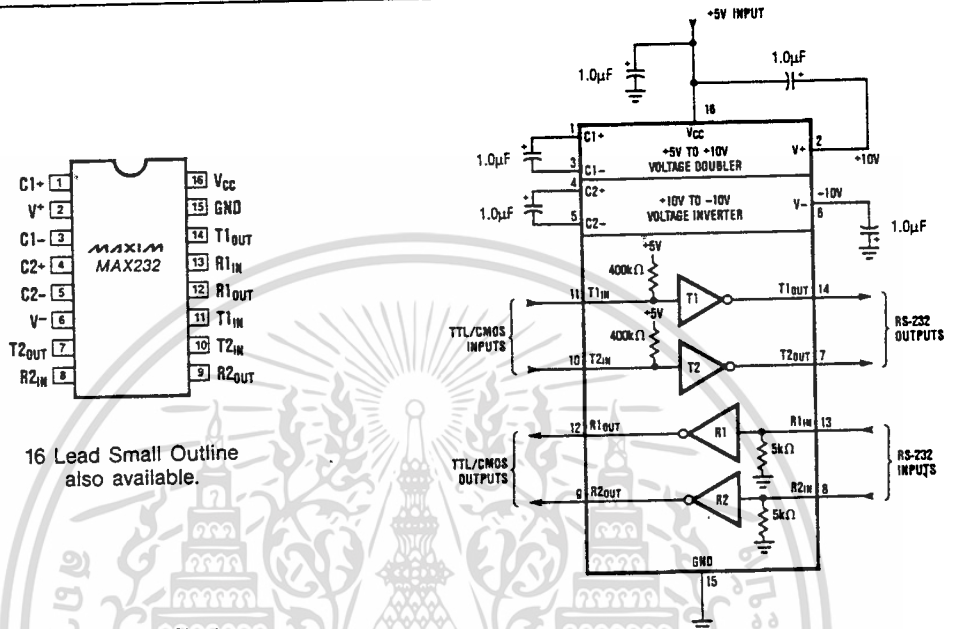


Figure 5. MAX232 Typical Operating Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## +5V Powered RS-232 Drivers/Receivers

### Typical Applications

Figures 3 through 14 show typical applications. The capacitor values are non-critical. Reducing the capacitors C1 and C2 to 1 $\mu$ F will slightly increase the impedance of the charge pump, lowering the RS-232 driver output voltages by about 100mV. Lower values of C3 and C4 increase the ripple on the V<sup>+</sup> and V<sup>-</sup> outputs.

If the power supply input to the device has a very fast rate-of-rise (as would occur if a PCB were to be plugged into a card cage with power already on), use the simple RC filter shown in Figure 15. This bypass network is not needed if the V<sub>CC</sub> rate-of-rise is below 1V/ $\mu$ s.

All receivers and drivers are inverting. The  $\overline{\text{ENable}}$  control of the MAX235, MAX236, MAX239, MAX240 and MAX241 enables the receiver TTL/CMOS outputs when it is at a low level, and places the TTL/CMOS outputs of the receivers into a high impedance state when it is at a high level.

When the Shutdown control of the MAX230, MAX235, MAX236, MAX240 and MAX241 is at a logic 1 the charge pump is turned off, the receiver outputs are put into the high impedance state, V<sup>+</sup> is pulled down to V<sub>CC</sub>, V<sup>-</sup> is pulled up to ground, and the transmitter outputs are disabled. The supply current drops to less than 10 $\mu$ A.

### Detailed Description

The following sections provide supplementary information for those designers with non-standard applications and for those with interest in the internal operation of the devices.

The devices consist of 3 sections: the transmitters, the receivers, and the charge pump DC-DC voltage converter.

### +5V to $\pm$ 10V

#### Dual Charge Pump Voltage Converter

All but the MAX231 and MAX239 convert +5V to  $\pm$ 10V. This conversion is performed by two charge pump voltage converters. The first uses capacitor C1 to double the +5V to +10V, storing the +10V on the V<sup>+</sup> output filter capacitor, C3. The second charge pump voltage converter uses capacitor C2 to invert the +10V to -10V, storing the -10V on the V<sup>-</sup> output filter capacitor, C4. The equivalent circuit of the charge pump section is shown in Figure 16.

A small amount of power may be drawn from the V<sup>+</sup> and V<sup>-</sup> outputs to power external circuitry. Two Typical Operating Characteristics graphs show typical output voltage versus load current for the MAX230, 234-238, and 241. Transmitter output current is included in these plots. The MAX231-233, which are not shown in the graphs, supply less output current, and are limited to 1 or 2mA of excess output load current.

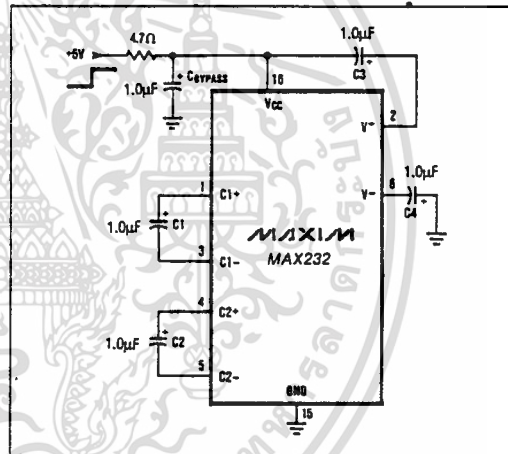


Figure 15. Protection from High  $\frac{dV}{dT}$

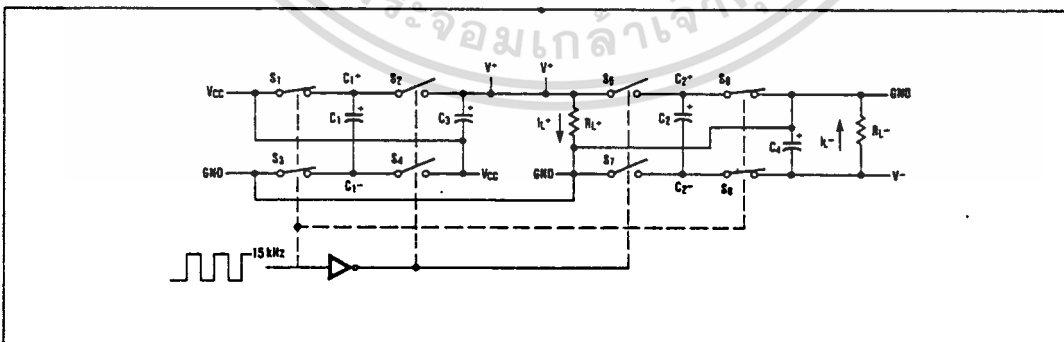


Figure 16. Charge Pump Diagram.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## +5V Powered RS-232 Drivers/Receivers

For applications needing only the +5V to  $\pm 10V$  charge pump voltage converter, the MAX680 is available.

The capacitor values for C1 through C4 are noncritical. At the 30kHz (MAX231-MAX233, 60kHz otherwise) typical switching frequency of the voltage converter, a 1 $\mu$ F capacitor has approximately 10 $\Omega$  impedance and replacing the 4.7 $\mu$ F and 10 $\mu$ F capacitors shown in the typical applications with 1 $\mu$ F for C1 and C2 will increase the output impedance of the  $V^+$  output by about 10 $\Omega$  and the output impedance of  $V^-$  by about 20 $\Omega$ . Lowering the value of C3 and C4 increases the ripple on the  $V^+$  and  $V^-$  outputs. Where operation to the upper temperature limit is not required, or  $V_{CC}$  will not go below 4.75V, C1 and C2 can be 1 $\mu$ F, and C3 and C4 can be 1 $\mu$ F per output channel (1 $\mu$ F if one transmitter is used, 5 $\mu$ F if five transmitters are used).

There are parasitic diodes which become forward biased if  $V^+$  goes below  $V_{CC}$  or  $V^-$  goes above ground. When in the shutdown mode (MAX230, MAX235, MAX236, MAX240 and MAX241 only),  $V^+$  is internally connected to  $V_{CC}$  by a 1k $\Omega$  pull-down, and  $V^-$  is internally connected to ground via a 1k $\Omega$  pull-up.

The MAX233 and MAX235 contain all charge pump components, including the capacitors, and operate with NO external components.

The MAX231 and MAX239 include only the  $V^+$  to  $V^-$  charge pump, and are intended for applications which have a +5V supply and either a +12V  $\pm 10\%$  supply or a 7.5V to 13.2V battery voltage. When operating with  $V^+$  greater than 8.0V, both capacitors can be 1 $\mu$ F.

### Driver (Transmitter) Section

The transmitters or line drivers are inverting level translators which convert the CMOS or TTL input levels to RS-232 or V.28 voltage levels. With +5V  $V_{CC}$ , the typical output voltage swing is  $\pm 9V$  when loaded with the nominal 5k $\Omega$  input resistance of an RS-232 receiver. The output swing is guaranteed to meet the RS-232/V.28 specification of  $\pm 5V$  minimum output swing under the worst case conditions of all transmitters driving the 3k $\Omega$  minimum allowable load impedance,  $V_{CC} = 4.5V$ , and maximum operating ambient temperature. The open circuit output voltage swing is from ( $V^+ - 0.6V$ ) to  $V^-$ .

The input thresholds are both CMOS and TTL compatible, with a logic threshold of about 25% of  $V_{CC}$ . The inputs of unused drivers sections can be left unconnected; an internal 400k $\Omega$  input pullup resistor to  $V_{CC}$  will pull the inputs high, forcing the unused transmitter outputs low. The input pullup resistors source about 12 $\mu$ A, and the driver inputs should be driven high or open circuited to minimize power supply current in the shutdown mode.

When in the low power shutdown mode, the driver outputs are turned off and their leakage current is less than 1 $\mu$ A with the driver output pulled to ground. The driver output leakage remains less than 1 $\mu$ A, even if the transmitter output is backdriven between 0V and ( $V_{CC} + 6V$ ). Below -0.5V the transmitter is diode clamped to ground with 1k $\Omega$  series impedance. The transmitter is also zener clamped to approximately  $V_{CC} + 6V$ , with a series impedance of 1k $\Omega$ . As required by the RS232 and V.28, the slew rate is limited to less than 30V/ $\mu$ s. This limits the maximum usable baud rate to 19,200 baud.

### Receiver Section

All but the MAX230 and MAX234 contain RS-232/V.28 receivers. These receivers convert the  $\pm 5V$  to  $\pm 15V$  RS-232 signals to 5V TTL/CMOS outputs. Since the RS-232C/V.28 specifications define a voltage level greater than +3V as a 0, the receivers are inverting. Maxim has set the guaranteed input thresholds of the receivers to 0.8V minimum and 2.4V maximum, which are significantly tighter than the -3.0V minimum and +3.0V maximum required by the RS-232 and V.28 specifications. This allows the receivers to respond both to RS-232/V.28 levels and TTL level inputs. The receivers are protected against input overvoltage up to  $\pm 30V$ .

The 0.8V guaranteed lower threshold is important to ensure that the receivers will have a logic 1 output if the receiver is not being driven because the equipment containing the line driver is turned off or disconnected, or if the connecting cable has an open circuit or short circuit. In other words, the receiver implements Type 1 interpretation of fault conditions (§7 of V.28, §2.5 of RS-232C). While a 0V or even a -3V receiver threshold would be acceptable for the data lines, these lower thresholds would not give proper indication on the control lines such as DTR and DSR. The receivers, on the other hand, have a full 0.8V noise margin for detecting the power-down or cable-disconnected states.

The receivers have a hysteresis of approximately 0.5V, with a minimum guaranteed hysteresis of 200mV. This aids in obtaining clean output transitions, even with slow rise and fall time input signals with moderate amounts of noise and ringing. The propagation delays of the receivers are 350ns for negative-going input signals, and 650ns for positive-going input signals (see Typical Characteristics graphs).

The MAX239 has a receiver 3-state control line, and the MAX235, MAX236, MAX240 and MAX241 have both a receiver 3-state control line and a low power shutdown control. The receiver TTL/CMOS outputs are in a high impedance 3-state mode whenever the 3-state  $\bar{E}$ Nable line is high, and are also high impedance whenever the Shutdown control line is high.

## +5V Powered RS-232 Drivers/Receivers

### Review of EIA Standard RS-232-C and CCITT

#### — Recommendations V.28 and V.24

The most common serial interface between electronic equipment is the "RS232" interface. This serial interface has been found to be particularly useful for the interface between units made by different manufacturers since the voltage levels are defined by the EIA Standard RS-232-C and CCITT Recommendation V.28. The RS-232 specification also contains signal circuit definitions and connector pin assignments, while CCITT circuit definitions are contained in a separate document, Recommendation V.24. Originally intended to interface modems to computers and terminals, these standards have many signals which are not used for computer-to-computer or computer-to-peripheral communication.

Serial interfaces can be used with a variety of transmission formats. The most popular by far is the asynchronous format, generally at one of the standard baud rates of 300, 600, 1200, etc. The maximum recommended baud rate for RS-232 and V.28 is 20,000 baud, and the fastest commonly used baud rate is 19,200 baud. Asynchronous serial links use a variety of combinations of the number of data bits, what type (if any) of parity bit, and the number of stop bits. A typical combination is 7 data bits, even parity, and 1 stop bit.

RS232/V.28 physical links are also suitable for synchronous transmission protocols. These higher level protocols often use the standard RS-232C/V.28 voltage levels. Note that one type of physical link (such as RS-232/V.28 voltage levels) can be used for a variety of higher level protocols. Table 2 summarizes the voltage levels and other requirements of V.28 and RS-232.

#### Comparison of RS-232C/V.28 with other Standards

The other two most common serial interface specifications are the EIA RS423 and RS422/RS485 (CCITT recommendations V.10 and V.11). While the RS-232 or V.28/V.24 interface is the most common interface for communication between equipment made by different manufacturers, the RS423/V.10 interface and RS422/V.11 interfaces can operate at higher baud rates. In addition, the RS485 interface can be used for low cost local area networks.

The RS423 and V.10 interfaces are unbalanced or "single-ended" interfaces which use a differential receiver. This standard is intended for data signaling rates up to 100 kbit/s (100 kilobaud). It achieves this higher baud rate through more precise requirements

on the waveshape of the transmitters and through the use of differential receivers to compensate for ground potential variations between the transmitting and receiving equipment. With certain limitations, this interface is compatible with RS-232 and V.28. The limitations are:

- 1) less than 20,000 baud rate,
- 2) maximum cable lengths determined by RS-232 performance,
- 3) RS423/V.10 DTE and DCE signal return paths must be connected to the the RS232/V.28 signal ground,
- 4) the RS-232 transmitter output voltages must be limited to  $\pm 12V$ , or additional protection must be provided for the RS423/V.10 receivers, and
- 5) not all RS232/V.28 receivers will show proper power-off detection of V.10 transmitter outputs.

Maxim's MAX230 and MAX232-MAX238, MAX240 and MAX241 meet restrictions 4 and 5 over the entire range of recommended operating conditions. The MAX231 and MAX239 meet restrictions 4 and 5 provided that the V<sup>+</sup> voltage is 12.5V or less.

The RS422, RS485, and V.11 interfaces are balanced double-current interchanges suitable for baud rates up to 10 Mbit/s. These interfaces are not compatible with RS-232 or V.28 voltage levels.

### Application Hints

#### Operation at High Baud Rates

V.28 states that "the time required for the signal to pass through the transition region during a change in state shall not exceed 1 millisecond or 3 percent of the nominal element period on the interchange circuit, whichever is less." RS-232C allows the transition time to be 4 percent of the duration of a signal element. At 19,200 baud, the "nominal element period" is approximately  $50\mu s$ , of which 3 percent is  $1.5\mu s$ . Since the transition region is from  $-3V$  to  $+3V$ , this means the V.28 slew rate would ideally be faster than  $6V/1.5\mu s = 4V/\mu s$  at 19.2 kbaud and  $2V/\mu s$  at 9600 baud. The RS-232 requirement is equivalent to  $3V/\mu s$  at 19.2 kbaud,  $1.5V/\mu s$  at 9600 baud, etc. The slew rate of the MAX230 series devices is about  $3V/\mu s$  with the maximum recommended load of 2500pF. In practice, the effect of less than optimum slew rate is a distortion of the recovered data, where the 1's and 0's no longer have equal width. This distortion generally has negligible effect and the devices can be reliably used for 19.2 kbaud serial links when the cable capacitance is kept below 2500pF. With very low capacitance loading, the MAX230 and MAX234-239, MAX240 and MAX241 may even be used at 38.4 kbaud, since the typical slew rate is  $5V/\mu s$  when loaded with 500pF in parallel with  $5k\Omega$ . Under no circumstance will the

## +5V Powered RS-232 Drivers/Receivers

### Non-Inverting Drivers and Receivers

Occasionally a non-inverting driver or receiver is needed instead of the inverting drivers and receivers of the family. Simply use one of the receivers as a TTL/CMOS inverter to get the desired operation (Figure 17). If the logic output driving the receiver input has less than 1mA of output source capability, then add the 2.2k $\Omega$  pullup resistor.

The receiver TTL outputs can directly drive the input of another receiver to form a non-inverting RS-232 receiver.

### Protection for Shorts to $\pm 15V$ Supplies

All driver outputs except on the MAX231, MAX232 and MAX233 are protected against short circuits to  $\pm 15V$ , which is the maximum allowable loaded output voltage of an RS-232/V.28 transmitter. The MAX231, MAX232, and MAX233 can be protected against short circuits to  $\pm 15V$  power supplies by the addition of a series 220 $\Omega$  resistor in each output. This protection is not needed to protect against short circuits to most RS-232 transmitters such as the 1488, since they have an internal short circuit current limit of 12mA.

The power dissipation of the MAX230 and MAX234-MAX239, MAX240 and MAX241 is about 200mW with all transmitters shorted to  $\pm 15V$ .

### Isolated RS-232 Interfaces

RS-232 and V.28 specifications require a common ground connection between the two units communicating via the RS-232/V.28 interface. In some cases, there may be large differences in ground potential between the two units, and in other cases it may be desired to avoid ground loop currents by isolating the two grounds. In other cases, a computer or control system must be protected against accidental connection of the RS-232/V.28 signal lines to 110/220VAC power lines. Figure 18 shows a circuit with this isolation. The power for the MAX233 is generated by a MAX635 DC-DC converter. When the MAX635 regulates point "A" to -5V, the isolated output at point "B" will be semi-regulated to +5V. The two optocouplers maintain isolation between the system ground and the RS-232 ground while transferring the data across the isolation barrier. While this circuit will not withstand 110VAC between the RS-232 ground and either the receiver or transmitter lines, the voltage difference between the two grounds is only limited by the optocoupler and DC-DC converter transformer breakdown ratings.

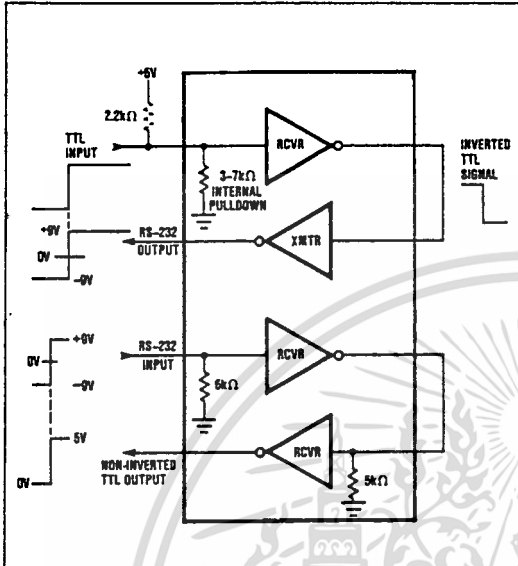


Figure 17. Non-inverting RS-232 Transmitters and Receivers.

slew rate exceed the RS-232/V.28 maximum spec of 30V/ $\mu$ s and, unlike the 1488 driver, no external compensation capacitors are needed under any load condition.

### Driving Long Cables

The RS-232 standard states that "The use of short cables (each less than approximately 50 feet or 15 meters) is recommended; however, longer cables are permissible, provided that the load capacitance . . . does not exceed 2500pF"

Baud rate and cable length can be traded off: use lower baud rates for long cables, use short cables if high baud rates are desired. For both long cables and high baud rates, use RS422/V.11. The maximum cable length for a given baud rate is determined by several factors, including the capacitance per meter of cable, the slew rate of the driver under high capacitive loading, the receiver threshold and hysteresis, and the acceptable bit error rate. The receivers have 0.5V of hysteresis, and the drivers are designed such that the slew rate reduction caused by capacitive loading is minimized (see Typical Characteristics).

## +5V Powered RS-232 Drivers/Receivers

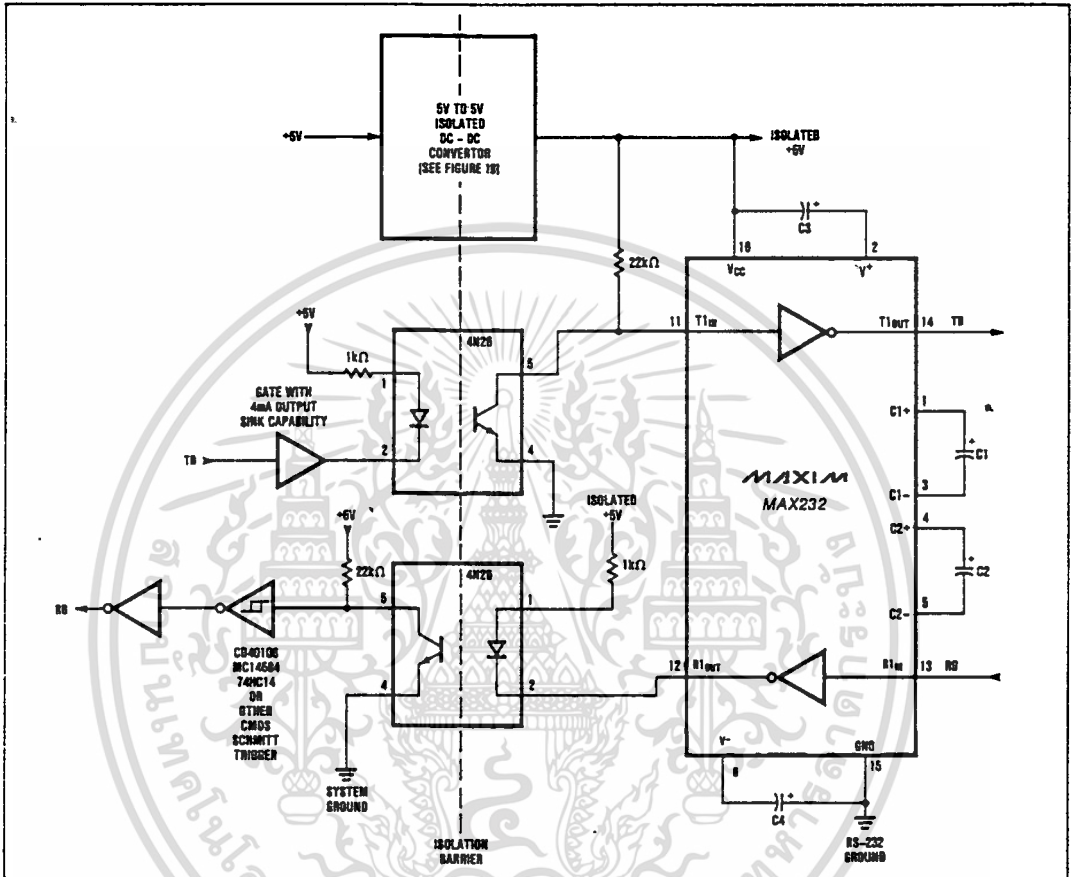


Figure 18. Optically isolated RS-232 Interface.

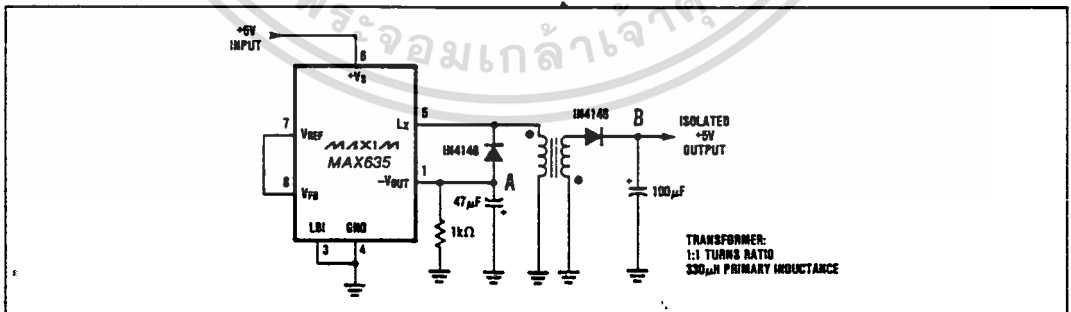


Figure 19. +5V Isolated Power Supply For Optically Isolated RS-232 Interface.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## +5V Powered RS-232 Drivers/Receivers

### Ordering Information

PART	TEMP. RANGE	PACKAGE
<b>MAX230</b>		<b>0.3" Wide</b>
MAX230CPP	0°C to +70°C	20 Lead Plastic DIP
MAX230CWP	0°C to +70°C	20 Lead Wide S.O.
MAX230C/D	0°C to +70°C	Dice
MAX230EPP	-40°C to +85°C	20 Lead Plastic DIP
MAX230EWP	-40°C to +85°C	20 Lead Wide S.O.
MAX230EJP	-40°C to +85°C	20 Lead CERDIP
MAX230MJP	-55°C to +125°C	20 Lead CERDIP
<b>MAX231</b>		<b>0.3" Wide</b>
MAX231CPD	0°C to +70°C	14 Lead Plastic DIP
MAX231CWE	0°C to +70°C	16 Lead Wide S.O.
MAX231C/D	0°C to +70°C	Dice
MAX231EPD	-40°C to +85°C	14 Lead Plastic DIP
MAX231EWE	-40°C to +85°C	16 Lead Wide S.O.
MAX231EJD	-40°C to +85°C	14 Lead CERDIP
MAX231MJD	-55°C to +125°C	14 Lead CERDIP
<b>MAX232</b>		<b>0.3" Wide</b>
MAX232CPE	0°C to +70°C	16 Lead Plastic DIP
MAX232CWE	0°C to +70°C	16 Lead Wide S.O.
MAX232C/D	0°C to +70°C	Dice
MAX232EPE	-40°C to +85°C	16 Lead Plastic DIP
MAX232EJE	-40°C to +85°C	16 Lead CERDIP
MAX232EWE	-40°C to +85°C	16 Lead Wide S.O.
MAX232MJE	-55°C to +125°C	16 Lead CERDIP
<b>MAX233</b>		<b>0.3" Wide</b>
MAX233CPP	0°C to +70°C	20 Lead Plastic DIP
MAX233EPP	-40°C to +85°C	20 Lead Plastic DIP
<b>MAX234</b>		<b>0.3" Wide</b>
MAX234CPE	0°C to +70°C	16 Lead Plastic DIP
MAX234CWE	0°C to +70°C	16 Lead Wide S.O.
MAX234C/D	0°C to +70°C	Dice
MAX234EPE	-40°C to +85°C	16 Lead Plastic DIP
MAX234EWE	-40°C to +85°C	16 Lead Wide S.O.
MAX234EJE	-40°C to +85°C	16 Lead CERDIP
MAX234MJE	-55°C to +125°C	16 Lead CERDIP
<b>MAX235</b>		<b>0.6" Wide</b>
MAX235CPG	0°C to +70°C	24 Lead Plastic DIP*
MAX235EPG	-40°C to +85°C	24 Lead Plastic DIP*
MAX235EDG	-40°C to +85°C	24 Lead Ceramic*
MAX235MDG	-55°C to +125°C	24 Lead Ceramic*

\* = 0.600" package

PART	TEMP. RANGE	PACKAGE
<b>MAX236</b>		<b>0.3" Wide</b>
MAX236CNG	0°C to +70°C	24 Lead Plastic DIP
MAX236CWG	0°C to +70°C	24 Lead Wide S.O.
MAX236C/D	0°C to +70°C	Dice
MAX236ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX236EWG	-40°C to +85°C	24 Lead Wide S.O.
MAX236ERG	-40°C to +85°C	24 Lead CERDIP
MAX236MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX237</b>		<b>0.3" Wide</b>
MAX237CNG	0°C to +70°C	24 Lead Plastic DIP
MAX237CWG	0°C to +70°C	24 Lead Wide S.O.
MAX237C/D	0°C to +70°C	Dice
MAX237ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX237EWG	-40°C to +85°C	24 Lead Wide S.O.
MAX237ERG	-40°C to +85°C	24 Lead CERDIP
MAX237MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX238</b>		<b>0.3" Wide</b>
MAX238CNG	0°C to +70°C	24 Lead Plastic DIP
MAX238CWG	0°C to +70°C	24 Lead Wide S.O.
MAX238C/D	0°C to +70°C	Dice
MAX238ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX238EWG	-40°C to +85°C	24 Lead Wide S.O.
MAX238ERG	-40°C to +85°C	24 Lead CERDIP
MAX238MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX239</b>		<b>0.3" Wide</b>
MAX239CNG	0°C to +70°C	24 Lead Plastic DIP
MAX239CWG	0°C to +70°C	24 Lead Wide S.O.
MAX239C/D	0°C to +70°C	Dice
MAX239ENG	-40°C to +85°C	24 Lead Plastic DIP
MAX239EWG	-40°C to +85°C	24 Lead Wide S.O.
MAX239ERG	-40°C to +85°C	24 Lead CERDIP
MAX239MRG	-55°C to +125°C	24 Lead CERDIP
<b>MAX240</b>		<b>Flatpak</b>
MAX240CMH	0°C to +70°C	44 Lead Flatpak
MAX240EMH	-40°C to +85°C	44 Lead Flatpak
<b>MAX241</b>		<b>0.3" Wide</b>
MAX241CWI	0°C to +70°C	28 Lead Wide S.O.
MAX241EWI	-40°C to +85°C	28 Lead Wide S.O.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## +5V Powered RS-232 Drivers/Receivers

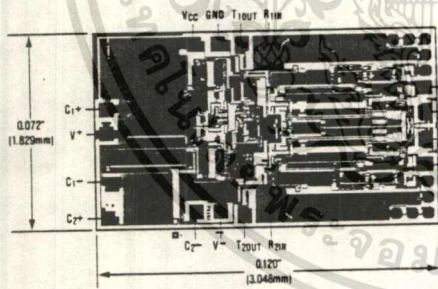
Table 1. Circuits Commonly Used for RS-232C and V.24 Asynchronous Interfaces

PIN	CIRCUIT	
1	Protective Ground	Connect to Earth Ground
2	Transmit Data (TD)	Data from DTE
3	Receive Data (RD)	Data from DCE
4	Request To Send (RTS)	Handshake from DTE
5	Clear to Send (CTS)	Handshake from DCE
6	Data Set ready (DSR)	Handshake from DCE
7	Signal Ground	Reference Point for Signals
8	Received Line Signal Detector (sometimes called Carrier Detect, DCD)	Handshake from DCE
11	Printer Busy Signal	Handshake from Printer
20	Data Terminal Ready	Handshake from DTE
22	Ring Indicator	Handshake from DCE

Table 2. Summary of RS-232C and V.28 Electrical Specifications

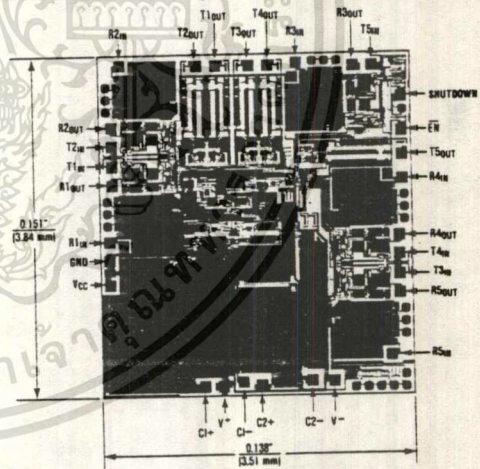
PARAMETER	SPECIFICATION	COMMENTS
Driver Output Voltage		
0 level	+5V to +15V	With 3-7kΩ load
1 level	-5V to -15V	With 3-7kΩ load
Max. output	±25V Max.	No Load
Receiver Input Thresholds (data and clock signals)		
0 level	+3V to +25V	
1 level	-3V to -25V	
Receiver Thresholds RTS, DSR, DTR		
On level	+3V to +25V	Detects Power Off Condition at Driver
Off level	Open Circuit or -3V to -25V	
Receiver Input Resistance	3kΩ to 7kΩ	
Driver Output Resistance, power off condition	300Ω Min.	V <sub>OUT</sub> < ±2V
Driver Slew Rate	30V/μs Max.	3kΩ < R <sub>L</sub> < 7kΩ; 0pF < C <sub>L</sub> < 2500pF
Signalling Rate	Up to 20kbits/sec.	
Cable Length	50'/15 m. Recommended Max. Length	Longer cables permissible, if C <sub>LOAD</sub> ≤ 2500pF

### Chip Topography



MAX231, MAX232 and MAX233

Note: Connect substrate to V\*.



MAX230 and MAX234-239, MAX240, MAX241

**Notes:**

1. Shutdown pin of MAX234, MAX237, MAX238, MAX239, MAX240 and MAX241 are internally connected to ground.
2. Connect substrate to V\*.

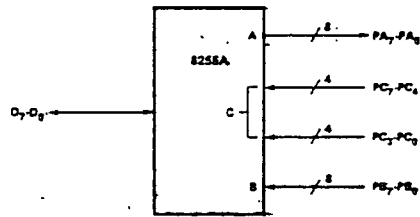
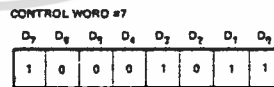
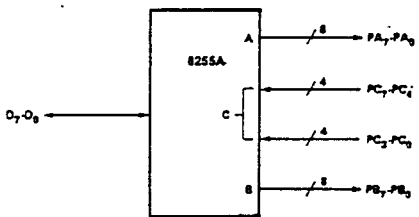
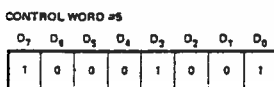
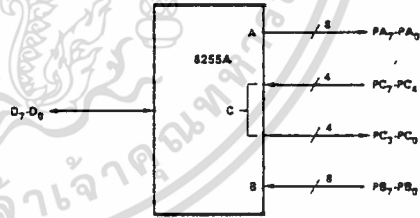
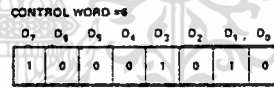
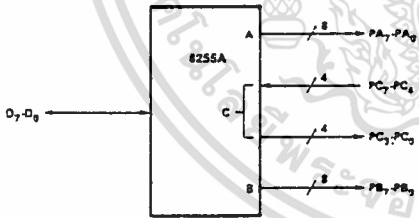
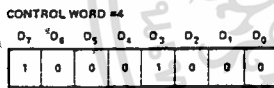
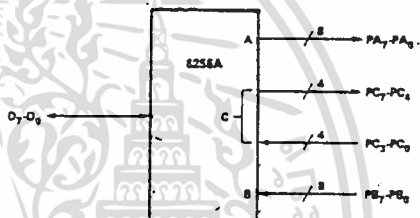
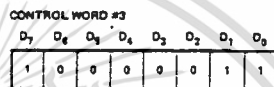
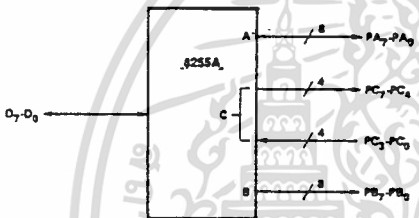
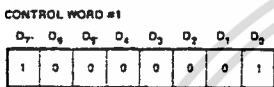
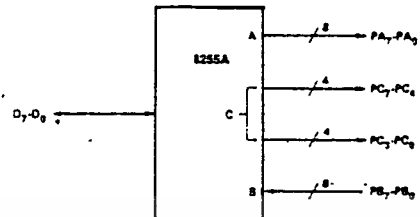
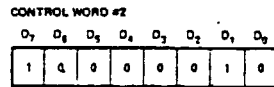
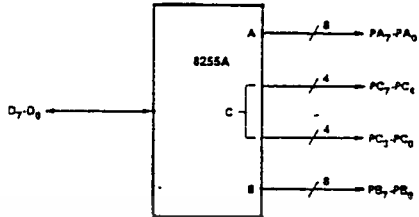
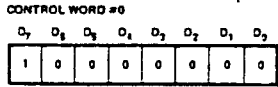


ภาคผนวก ช

**CONTROL WORD 8255 MODE 0**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

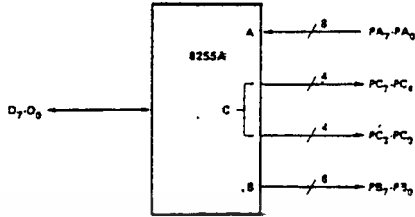
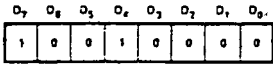
MODE 0 Configurations



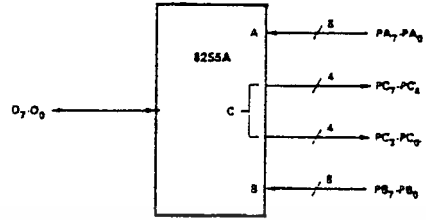
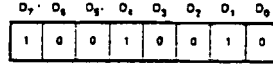
แสดงลักษณะต่างๆ ในการใช้งานในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

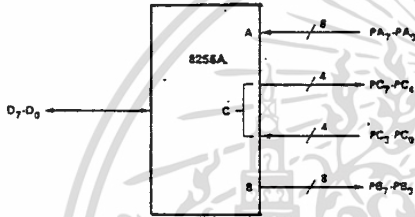
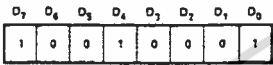
CONTROL WORD #8



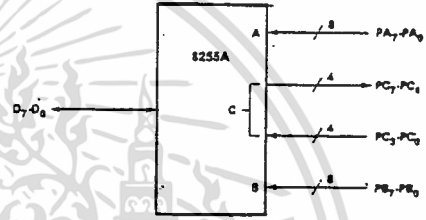
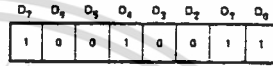
CONTROL WORD #10



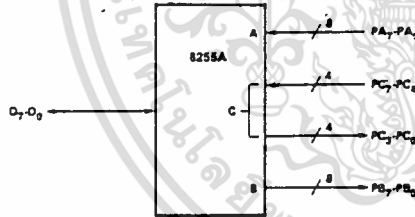
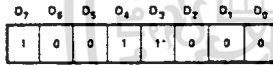
CONTROL WORD #9



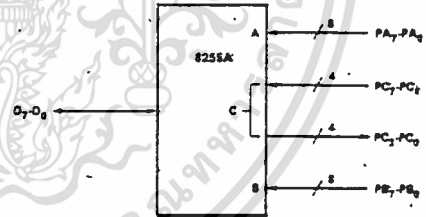
CONTROL WORD #11



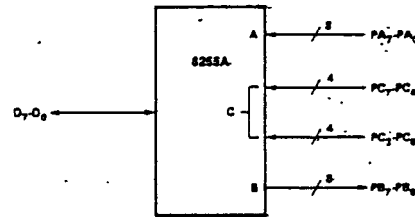
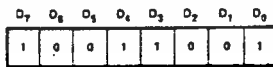
CONTROL WORD #12



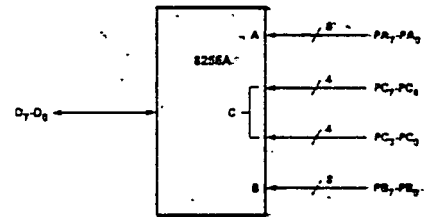
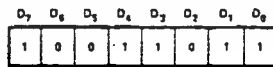
CONTROL WORD #14



CONTROL WORD #13



CONTROL WORD #18

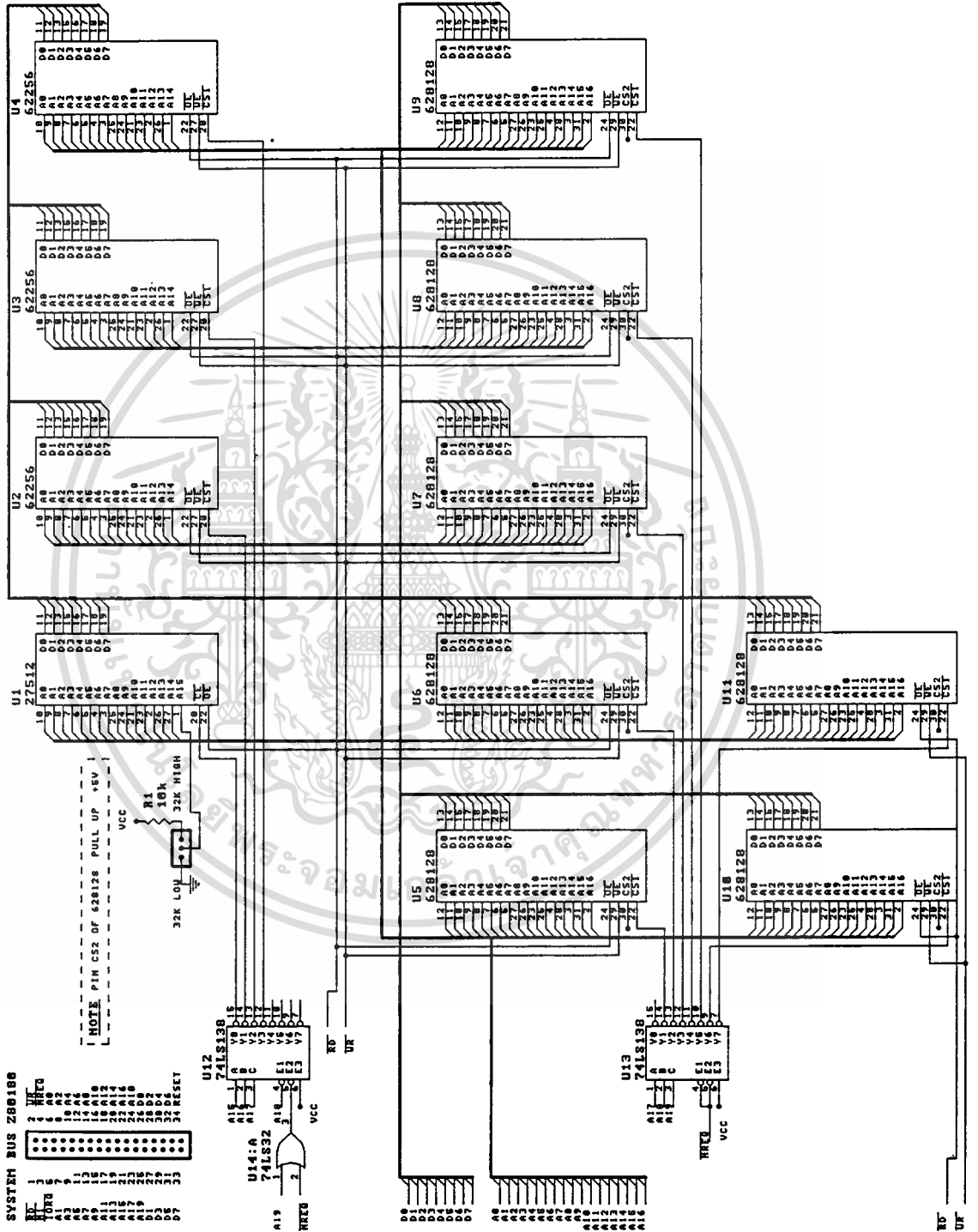


แสดงลักษณะต่างๆ ในการใช้งานในโหมด 0 (ต่อ)

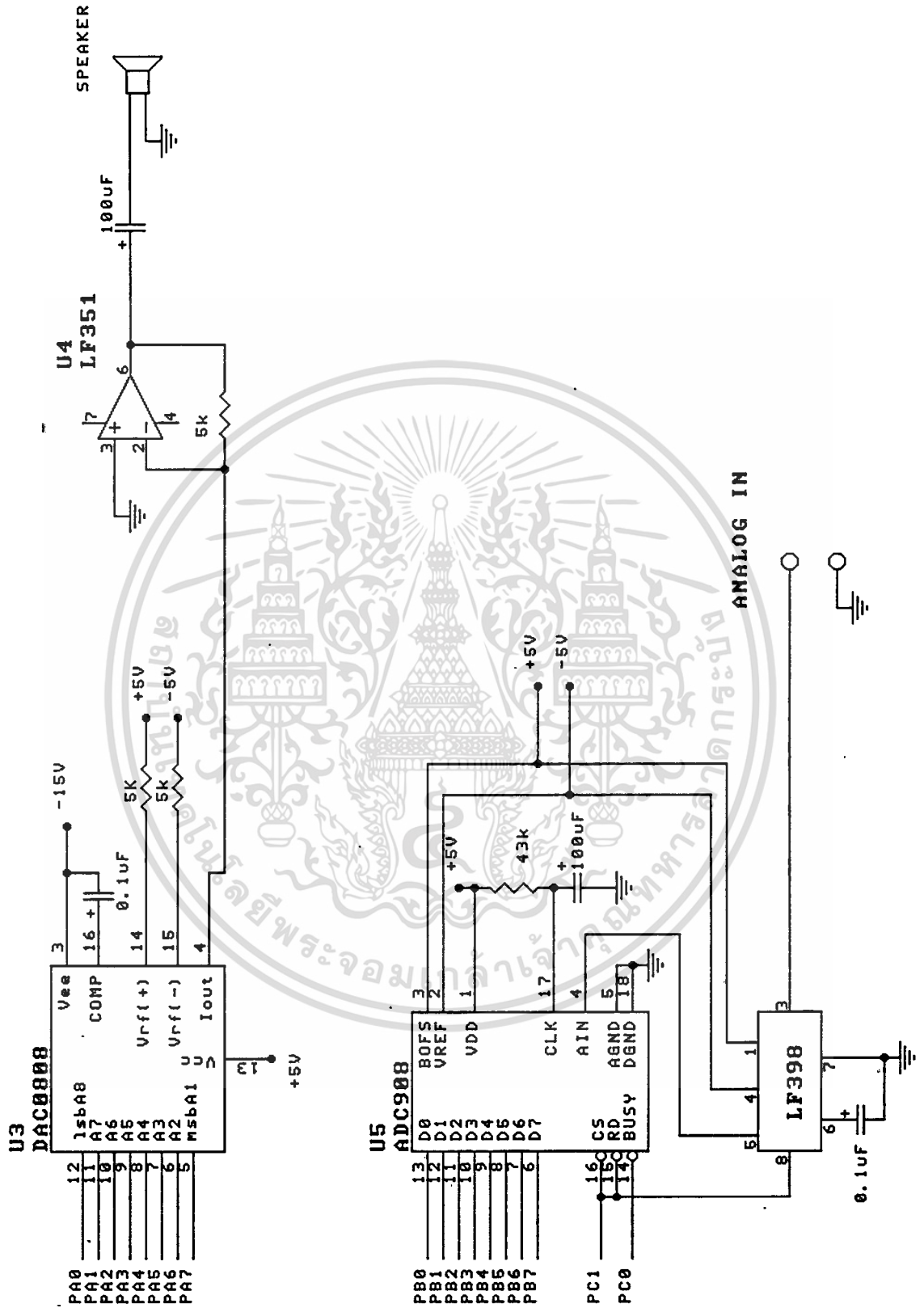
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

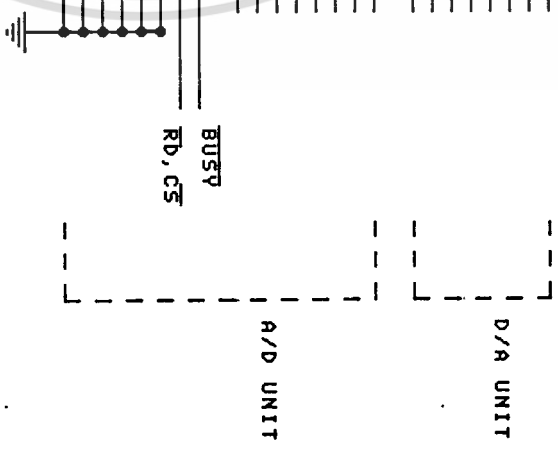
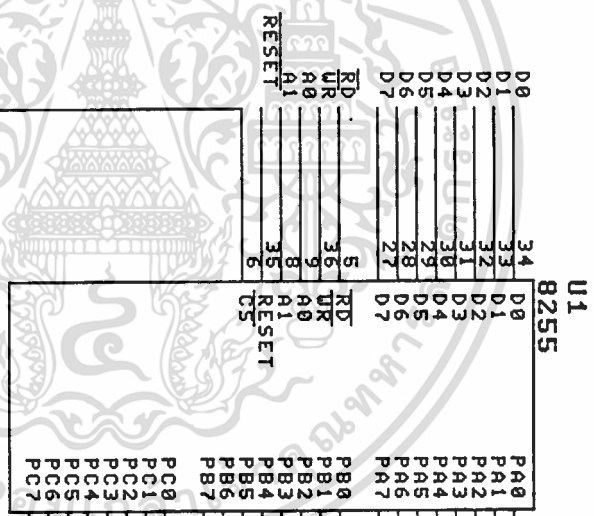
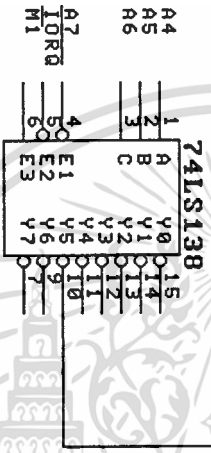
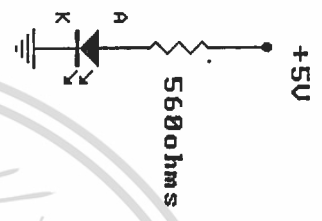
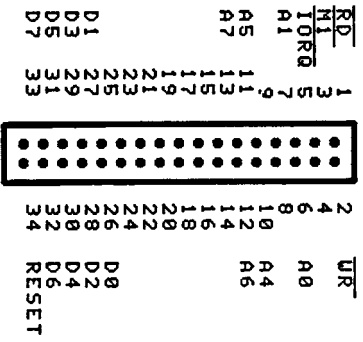


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

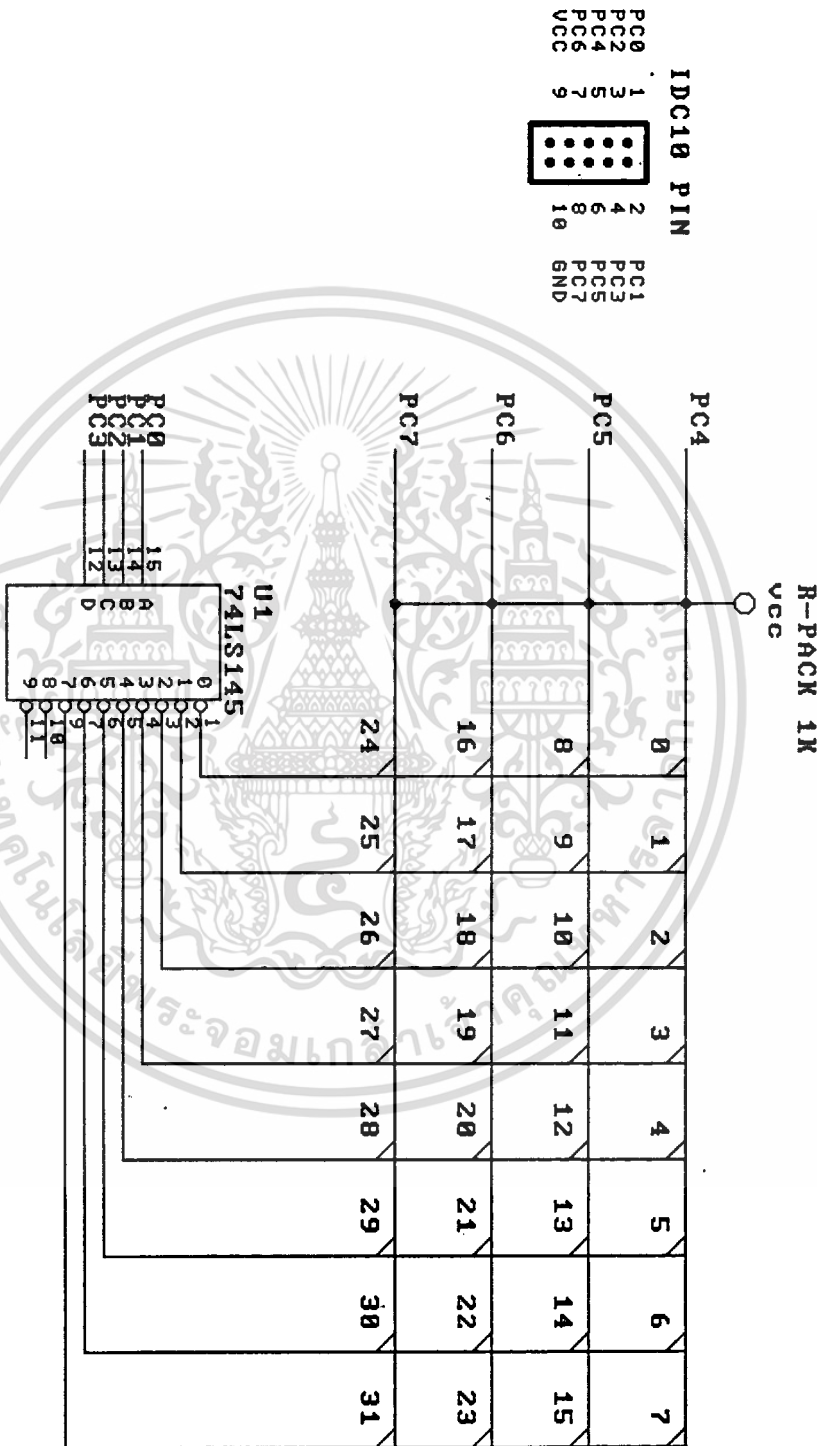


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าการณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

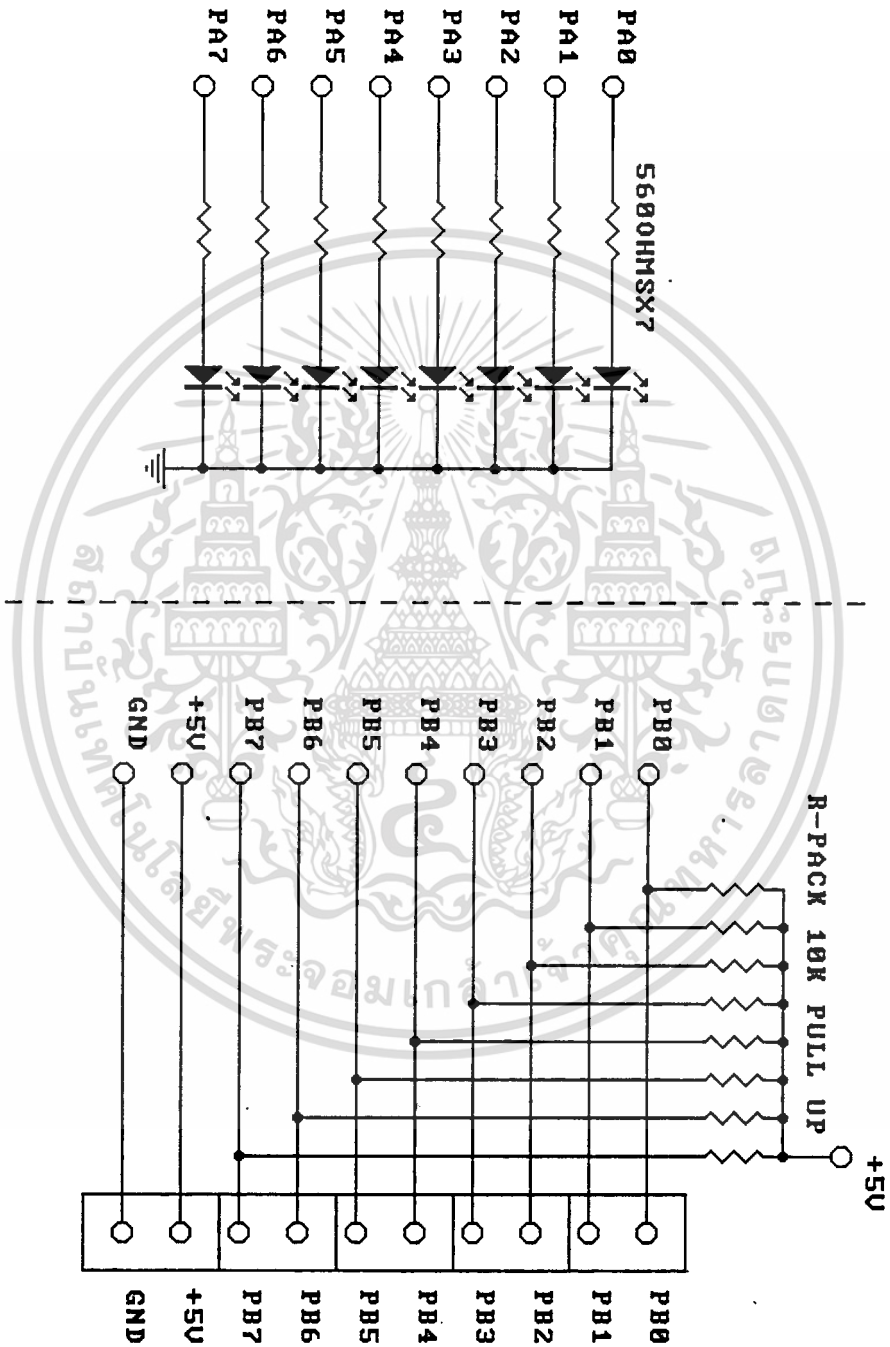
SYSTEM BUS Z80180



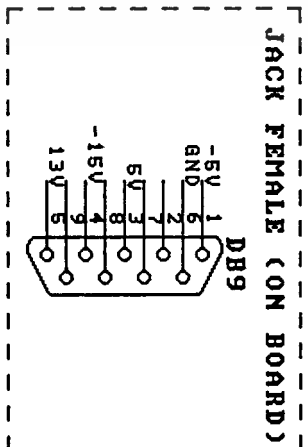
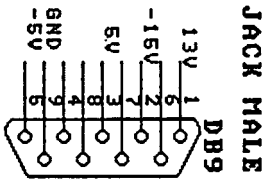
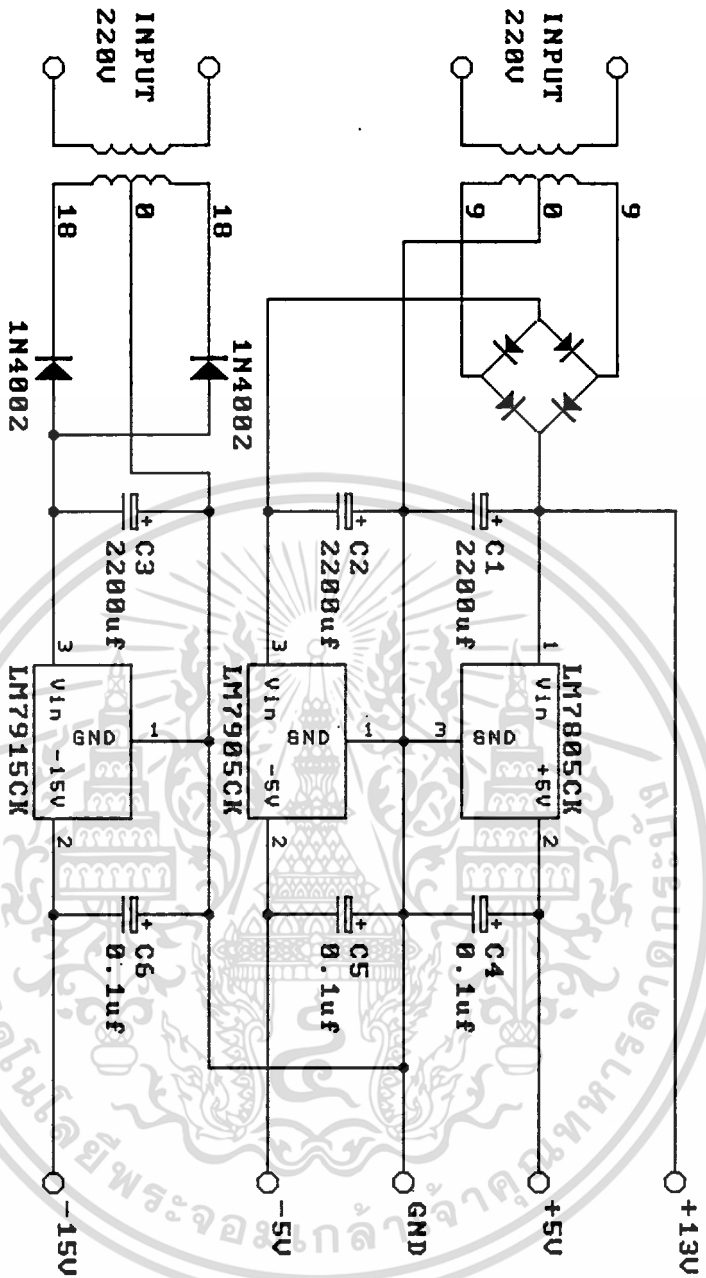
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; FILENAME      MON180.ASM
; DESCRIPTION   Z80180 MONITOR V1.0
; HARDWARE     FOR Z80180
; ASSEMBLER    CROSS-32 V3.0
; KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
; FACULTY OF INDUSTRIAL EDUCATION
; MAJOR: ELECTRONIC AND COMPUTER
; ED.ENGINEER #15

; THE FOLLOWING STATEMENTS DEFINE THE SPECIAL FUNCTION
; I/O REGISTERS USED BY THE Z180.

CNTLA0: EQU    00H    ;ASCII CONTROL REG A CH 0
CNTLA1: EQU    01H    ;ASCII CONTROL REG A CH 1
CNTLBO: EQU    02H    ;ASCII CONTROL REG B CH 0
CNTLB1: EQU    03H    ;ASCII CONTROL REG B CH 1
STAT0:  EQU    04H    ;ASCII STATUS  REG  CH 0
STAT1:  EQU    05H    ;ASCII STATUS  REG  CH 1
TDR0:   EQU    06H    ;ASCII XMIT DATA REG CH 0
TDR1:   EQU    07H    ;ASCII SMIT DATA REG CH 1
RDR0:   EQU    08H    ;ASCII RECV DATA REG CH 0
RDR1:   EQU    09H    ;ASCII RECV DATA REG CH 1

CNTR:   EQU    0AH    ;CSI/O CONTROL REGISTER

TRDR:   EQU    0BH    ;CSI/O XMIT/RECV DATA REG
TMDROL: EQU    0CH    ;TIMER  DATA REG CH 0L

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TMDROH: EQU 0DH ;TIMER DATA REG CH 0H  
 RLDR0L: EQU 0EH ;RELOAD REGISTER CH 0L  
 RLDR0H: EQU 0FH ;RELOAD REGISTER CH 0H  
 TCR: EQU 10H ;TIMER CONTROL REGISTER  
  
 TMDR1L: EQU 14H ;TIMER DATA REG CH 1L  
 TMDR1H: EQU 15H ;TIMER DATA REG CH 1H  
 RLDR1L: EQU 16H ;RELOAD REGISTER CH 1L  
 RLDR1H: EQU 17H ;RELOAD REGISTER CH 1H  
  
 SAR0L: EQU 20H ;DMA SOURCE ADDRESS REG CH 0L  
 SAR0H: EQU 21H ;DMA SOURCE ADDRESS REG CH 0H  
 SAR0B: EQU 22H ;DMA SOURCE ADDRESS REG CH 0B  
 DAR0L: EQU 23H ;DMA DSTNTN ADDRESS REG CH 0L  
 DAR0H: EQU 24H ;DMA DSTNTN ADDRESS REG CH 0H  
 DAR0B: EQU 25H ;DMA DSTNTN ADDRESS REG CH 0B  
 BCROL: EQU 26H ;DMA BYTE COUNT REGISTER CH 0L  
 BCROH: EQU 27H ;DMA BYTE COUNT REGISTER CH 0H  
 MAR1L: EQU 28H ;DMA MEMORY ADDRESS REG CH 1L  
 MAR1H: EQU 29H ;DMA MEMORY ADDRESS REG CH 1H  
 MAR1B: EQU 2AH ;DMA MEMORY ADDRESS REG CH 1B  
 IAR1L: EQU 2BH ;DMA DSTNTN I/O REG CH 1L  
 IAR1H: EQU 2CH ;DMA DSTNTN I/O REG CH 1H  
 BCR1L: EQU 2EH ;DMA BYTE COUNT REGISTER CH 1L  
 BCR1H: EQU 2FH ;DMA BYTE COUNT REGISTER CH 1H  
 DSTAT: EQU 30H ;DMA STATUS REGISTER  
 DMODE: EQU 31H ;DMA MODE REGISTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DCNTL: EQU    32H    ;DMA/ WAIT CONTROL REGISTER

IL:     EQU    33H    ;IL REGISTER (INT VECTOR REG)

ITC:    EQU    34H    ;INT/ TRAP CONTROL REGISTER

RCR:    EQU    36H    ;REFRESH CONTROL REGISTER

CBR:    EQU    38H    ;MMU COMMON BASE REGISTER

BBR:    EQU    39H    ;MMU BANK   BASE REGISTER

CBAR:   EQU    3AH    ;MMU COMMON/BANK AREA REGISTER

OMCR:   EQU    3EH    ;OPERATION MODE CONTROL REGISTER

ICR:    EQU    3FH    ;I/O CONTROL REGISTER

CPU"Z130.TBL"; CPU TABLE
HOF"INT8": HEX OUTPUT FORMAT

; ***** VARIABLE SET *****

SPA:    EQU    50H    ;DAC PORT

SPB:    EQU    51H    ;ADC PORT

SPC:    EQU    52H    ;BUSY RD,CS (ADC908) ACTIVE LOW

SPCC:   EQU    53H    ;CONTROL PORT

UPA:    EQU    80H    ;LED FLAG

UPB:    EQU    81H    ;USER PORT

UPC:    EQU    82H    ;KEYBOARD

UPCC:   EQU    83H    ;CONTROL PORT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRITE_INS: EQU 0C0H ;LCD PORT
WRITE_DATA: EQU 0C2H
READ_BUSY: EQU 0C4H
READ_DATA: EQU 0C6H
VECLOW: EQU 40H ;VECTOR LOW

```

```

; ***** RAM WORKING AREA *****

```

```

SYSSTK: EQU 0FFFFH ;SYSTEM STACK (2K)
WRKRAM: EQU 0E000H ;(8K) TOP MEMORY
USRSTK: EQU 0F000H ;USER STACK & USER BUFFER
ORG 0F000H ;SYSTEM BUFFER (2K)
RST10: DFS 2 ;RESTART MEMORY
RST18: DFS 2
RST20: DFS 2
RST28: DFS 2
RST30: DFS 2
RST38: DFS 2
RST66: DFS 2
USRPC: DFS 2 ;USER REG. MEMORY
USRSP: DFS 2
USRIF: DFS 2
USRIX: DFS 4 ;IX,IY
USRAF: DFS 8
USRAF1: DFS 6

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

USRH1:	DFS	2	
SEABUF:	DFS	8	:SEARCH DATA BUF
SEAMEM:	DFS	2	;SEARCH MEMORY
SEAFND:	DFS	2	:SEARCH FOUND MEMORY
SEACNT:	DFS	1	:SEARCH BYTE COUNT
ASMCOM:	DFS	8	:ASSEMBLER COMMAND BUF
ASMBUF:	DFS	3	:ASSEMBLER BUF
ASMOPC:	DFS	4	;OPCODE BUF
ASMDAT:	DFS	2	;N NN (N) (NN) DIS
ASMDATD:	DFS	1	:+D
UASMFAG:	DFS	1	:UNASM FLAG 0=ALL 1=NO ADDRESS.OP-CODE
UASMCNT:	DFS	1	:UNASM LINE COUNT
MONBUF:	DFS	61	:COMMAND BUF
MONINX:	DFS	1	;COMMAND BUF INDEX
MONDAT:	DFS	2	;16 BIT DATA FROM VAL SUB.
BAUD:	DFS	1	;BAUD RATE 12H,24H,48H.96H
BRKMEM:	DFS	1	:BREAK OLD DATA MEMORY
BRKFAG:	DFS	1	;BREAK FLAG (B0=BREAK B1=TRACE)
BRK:	DFS	2	;BREAK ADDRESS MEMORY
STPADD:	DFS	2	;TRACE ADDRESS MEMORY (SINGLE STEP)
START:	DFS	2	:START ADDRESS MEMORY
FINAL:	DFS	2	;FINAL ADDRESS MEMORY
DEST:	DFS	2	;DESTINATION ADDRESS MEMORY
POWMEM:	DFS	1	;POWER UP MEMORY (A5)
KEYIN:	DFS	1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

KEY_SHIFT: DFS 1
            DFS 8      ;RESERVE
BUFFER:           ;SPACE
IND:      DFS 1

```

```

; ***** POWER-UP *****

```

```

ORG 0000H
POWER: JP  RESET
      DFB  OFFH,OFFH,OFFH,OFFH,OFFH
; ***** RESTART *****
      PUSH HL          ;RST 08H FROM BREAK, TRACE
      LD   HL, BACK
      EX  (SP),HL
      RET
      DFB  OFFH,OFFH
      PUSH HL          ;RST 10H (USER)
      LD   HL, (RST10)
      EX  (SP),HL
      RET
      DFB  OFFH,OFFH

      PUSH HL          ;RST 18H (USER)
      LD   HL, (RST18)
      EX  (SP),HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
DFB OFFH,OFFH

PUSH HL ;RST 20H (USER)
LD HL,(RST20)
EX (SP),HL
RET
DFB OFFH,OFFH
PUSH HL ;RST 28H (USER)
LD HL,(RST28)
EX (SP),HL
RET
DFB OFFH,OFFH
PUSH HL ;RST 30H (USER)
LD HL,(RST30)
EX (SP),HL
RET
DFB OFFH,OFFH
PUSH HL ;RST 38H (USER)
LD HL,(RST38)
EX (SP),HL
RET
DFB OFFH,OFFH

```

; \*\* INTERNAL INTERRUPTS VECTOR TABLE \*\*

INT1: DWL ZZZ ;SLEEP

INT2: DWL ZZZ ;SLEEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PRT0:    DWL   ZZZ           :SLEEP
PRT1:    DWL   ZZZ           :SLEEP
DMA0:    DWL   ZZZ           :SLEEP
DMA1:    DWL   ZZZ           :SLEEP
CSIO:    DWL   ZZZ           :SLEEP
ASCII0:  DWL   ZZZ           :SLEEP
ASCII1:  DWL   RXBYTE1      :RX ASCII FORM RS-232 PORT

```

```

DFB   OFFH.OFFH.OFFH.OFFH.OFFH
DFB   OFFH.OFFH.OFFH.OFFH.OFFH
DFB   OFFH.OFFH.OFFH.OFFH.OFFH
DFB   OFFH.OFFH.OFFH.OFFH.OFFH
PUSH  HL           :NMI (USER)
LD    HL,(RST66)
EX    (SP),HL
RET

```

```
; ***** RESET *****
```

```

RESET:  LD    HL,2000H      :POWER UP DELAY
RESET1: DEC   HL
        LD    A,H
        OR   L
        JR   NZ,RESET1

```

```
; *****
```

```
; * INITIAL PARAMETER *
```

```
; *****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD A,0 ;FOR INSTRUCTION SET Z180 ONLY
OUT0 (OMCR),A
```

```
DFLT: LD A,0ESH ;MMU SETUP DEFAULT PAGE1
```

```
OUT0 (CBAR),A
```

```
LD A,0
```

```
OUT0 (BBR),A
```

```
LD A,0FOH
```

```
OUT0 (CBR),A
```

```
LD SP,SYSTK ;USER STACK
```

```
LD A,VELOW ;SET LOW VECTOR
```

```
OUT0 (IL),A
```

```
LD A,SAH ;(38H) SET CONTROL CP-JR180
```

```
OUT (UPCC),A
```

```
XOR A
```

```
OUT (UPA),A ;CLEAR LED FLAG
```

```
LD A,SBH ;SET CONTROL SIGNAL BOARD
```

```
OUT (SPCC),A
```

```
CALL INITLCD ;INITIAL LCD
```

```
LD HL,LCDTAB ;" Remote Monitor "
```

```
CALL WRP ;WRITE LCD
```

```
CALL DELY
```

```
CALL ASCI ;SET RS-232
```

```
LD A,(POWMEM) ;CHECK POWER UP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CP 0A5H
JR Z.RESET2
```

```
; **** POWER UP ****
```

```
: CLEAR WORKING AREA
```

```
XOR A
LD HL,WRKRAM
LD DE,WRKRAM+1
LD BC,SYSSTK-WRKRAM-1
LD (HL),A
LDIR
LD HL,USRSTK ;DEFAULT ADDRESS
LD (FINAL),HL
RESET2: CALL AUTOB ;AUTO BAUD RATE
LD B,24
RESET3: CALL TXLF ;CLEAR SCEEN
DJNZ RESET3
LD HL,MNOTAB
CALL TXBLOCK
LD HL,MN1TAB
CALL TXBLOCK
LD HL,MN2TAB
CALL TXBLOCK
LD HL,MN3TAB
CALL TXBLOCK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,MN4TAB
CALL TXBLOCK
LD HL,BAUTAB
CALL TXBLOCK
LD A,(BAUD)
LD D,A
CALL TXBYTE2
LD D,0
CALL TXBYTE2
LD HL,MN5TAB
CALL TXBLOCK
LD HL,MN6TAB
CALL TXBLOCK
LD HL,MN7TAB
CALL TXBLOCK
CALL TXLF
CALL TXLF

LD HL,USRSTK ;DEFAULT VAR
LD (USRPC),HL
LD HL,USRSTK
LD (USRSP),HL
LD HL,BRKFLAG
BIT 0,(HL) ;CHECK BREAK FLAG
JR Z,RESET4
RES 0,(HL) ;ON BREAK PROCESS
LD HL,(BRK)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD    A,(BRKMEM)
LD    (HL),A
JR    RESETS

RESET4: BIT    1,(HL)          ;CHECK TRACE FLAG
JR    Z.RESETS
RES   0,(HL)          ;ON TRACE PROCESS
LD    HL,(STPADD)
LD    A,(BRKMEM)
LD    (HL),A

RESETS: LD    A,(POWMEM)      ;POWER UP DISPLAY
CP    0A5H
JR    Z.RESET9
LD    HL,POWTAB          ;IS POWER UP
CALL  TXBLOCK
LD    A,0A5H
LD    (POWMEM),A

```

RESET9:

; \*\*\*\*\* MONITOR MAIN \*\*\*\*\*

```

MON:  LD    D,07H          ;BELL
CALL  TXBYTE
LD    D,5AH              ;Z
CALL  TXBYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD D,31H ;1
CALL TXBYTE
LD D,38H ;8
CALL TXBYTE
LD D,30H ;0
CALL TXBYTE
LD D,3EH ;>
CALL TXBYTE
CALL TMGETL
XOR A
LD (MONINX),A ;CLEAR INDEX
LD A,(MONBUF)
CP ODH
JR Z,MON
LD A,(MONBUF+1)
CP ""
JR Z,MON1
CP ODH
JP NZ,ERROR

```

```

MON1: LD A,(MONBUF) ;COMMAND

```

```

CALL UPPER

```

```

MON2: LD HL,MN8TAB

```

```

LD IX,MN9TAB

```

```

LD B,22

```

```

MON3: CP (HL)

```

```

JR Z,MON4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC HL

INC IX

INC IX

DJNZ MON3

JR ERROR ;NO COMMAND

MON4: CALL NEXT

LD H.(IX+1) ;JUMP

LD L.(IX+0)

JP (HL)

; **** SET ASCI CH1 TX.RX,S.N.1,9600 (X'TAL 12.488 Mhz) ****
ASCI: LD A,64H

OUT0 (CNTLA1),A ;TX.RX,SBIT,1 STOP BIT

LD A.2

OUT0 (CNTLB1),A

LD A.S ;RIE ENABLE TO KEYBOARD

OUT0 (STAT1),A

RET

; ***** LCD CONTROL SUB *****

; WRITE TEXT 1 LINE (16 CHARACTER) ON LCD

; IN = HL --> TABLE TEXT ( 16 BYTE ONLY! )

; REG = ABCDHL

WRP: LD A,0

CALL GOTO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL WRLINE

LD A,40H

CALL GOTO

CALL WRLINE

RET

GOTO: SET 7.A

OUT (WRITE_INS),A

CALL READ

RET

READ: IN A,(READ_BUSY)

BIT 7,A

JR NZ,READ

RET

WRLINE: LD B,S

TEST11: LD A,(HL)

LD D,A

CALL WRBYTE

INC HL

DJNZ TEST11

RET

HOME: LD A,02H

OUT (WRITE_DATA),A

CALL READ

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; \*\*\*\*\* INITIAL LCD DL=1 8 BIT,N=1 1/16 DUTY,F=0 5\*7 \*\*\*\*\*

```
INITLCD: LD A.00111000B :FUNCTION SET 38H
          :DL=1 8 BIT
          :N=1 1/16 DUTY F=0 5*7
          OUT (WRITE_INS),A
          CALL WAIT1
          CALL WAIT1 :DELAY > 4.1 ms
          LD A.00001111B :DISPLAY ON/OFF CONTROL
          :D=1 OFF , C=1 CURSOR ON
          :B=1 BLINK
          OUT (WRITE_INS),A
          CALL WAIT1
          LD A.00000110B :ENTRY MODE SET
          :I/D=1 INCREMENT,S=0 RIGHT
          OUT (WRITE_INS),A
          CALL WAIT1
          RET
```

;\*\*\*\*\* CLEAR SCREEN LCD \*\*\*\*\*

```
CLSLCD: LD A.00000001B ;CLEAR ALL DISPLAY
          OUT (WRITE_INS),A
          CALL READ
          RET
```

; \*\*\*\*\* WRITE BYTE SUB \*\*\*\*\*

; IN = D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

WRBYTE:  LD    A,D
          OUT   (WRITE_DATA),A
          CALL  READ
          RET

```

```

: ***** READY SUB *****

```

```

WAIT1:  PUSH  BC
          PUSH  AF
          LD    B,SOH

```

```

WAIT12: DJNZ  WAIT12
          POP   AF
          POP   BC
          RET

```

```

; *** END LCD CONTROL SUB ***

```

```

: ***** SYNTAX ERROR MAIN *****

```

```

ERROR:  LD    SP,SYSTK      ;RESET STACK

```

```

ERROR1: LD    D," "
          CALL TXBYTE
          LD    A,(MONINX)
          CP   0
          JR   Z,ERROR2
          DEC  A
          LD   (MONINX),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR ERROR1
ERROR2: LD HL,SYNTAB :SYNTAX ERROR
CALL TXBLOCK
JP MON

```

```

; IN = HL ERROR POSITION ON MONBUF

```

```

ERRORX: LD DE,MONBUF
XOR A
SBC HL,DE
LD A,L
LD (MONINX),A
JP ERROR

```

```

; ***** ASM ROUTINE *****

```

```

; ZSO ASSEMBLER

```

```

; A ADDR

```

```

ASM: CP 0
JR Z,ERROR
CALL VAL :ADDR
CALL NEXT
CP 0
JR NZ,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASM1:   LD   DE.(MONDAT)      ;DISPLAY ADDR

        CALL TXBYTE4

        LD   D.":"

        CALL TXBYTE

        CALL TMGETL

        XOR  A

        LD   (MONINX),A

        LD   A,(MONBUF)      :FIRST DIGIT

        CP   0DH

        JP   Z,MON           ;END ASM

        LD   HL,0

        LD   (ASMBUF+1),HL   ;CLEAR OPRAND INDEX

        LD   (ASMDAT),HL

        LD   HL,MONBUF       ;TO UPPER

ASM2:   LD   A,(HL)

        CP   0DH

        JR   Z,ASM3

        CALL UPPER

        LD   (HL),A

        INC  HL

        JR   ASM2

ASM3:   LD   B,4              ;MNEMONIC SEARCH

        LD   HL,MONBUF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL ASMMOV
LD IX,MNETAB
LD IY,ASMCOM
LD BC.4 ;LENGTH
LD HL.67
CALL SEAR
CP 0
JP Z,ASMME ;MNE NOT FOUND
LD (ASMBUF),A ;MNE INDEX
ASM4: CALL NEXT
CP 0
JR Z,ASM7 ;NO OPERAND
LD HL,MONBUF ;SEARCH ","
LD D,0
LD E,A
ADD HL,DE
ASM41: LD A,(HL)
CP ","
JR Z,ASM6
CP " "
JR Z,ASM5
CP ODH
JR Z,ASM5
INC HL
JR ASM41

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASM5:   CALL ASMOPR           :1 OPERAND
        CP   OFFH
        JR   Z,ASMOE
        LD   (ASMBUF+1),A    ;OPERAND INDEX
        JR   ASM7

```

```

ASM6:   LD   A," "           ;2 OPERAND
        LD   (HL),A          :CHANGE ", " TO BLANK
        CALL ASMOPR
        CP   OFFH
        JR   Z,ASMOE
        LD   (ASMBUF+1),A    ;OPERAND 1 INDEX
        CALL NEXT
        CP   0
        JR   Z,ASMOE
        CALL ASMOPR
        CP   OFFH
        JR   Z,ASMOE
        LD   (ASMBUF+2),A    ;OPERAND 2 INDEX

```

```

ASM7:   CALL ASMS           :SEARCH TABLE
        CP   0
        JR   Z,ASMCE        ;NO COMMAND
        LD   B,A            ;OP-CODE LENGHT
        PUSH BC
        CALL ASMREL         ;CHECK JR MNEMONICS
        POP  BC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP    OFFH

JP    Z.ASM1          :RELATIVE ERROR

LD    HL.(MONDAT)

LD    DE.ASMOPC

ASM71: LD    A.(DE)

LD    (HL),A          :LOAD OPCODE TO MEM

INC   HL

INC   DE

DJNZ  ASM71

LD    (MONDAT),HL

JP    ASM1            :BACK TO NEXT LINE

ASMME: LD    HL,ASMER1 :MNEMONIC ERROR

CALL  TXBLOCK

JP    ASM1

ASMOE: LD    HL,ASMER2 :OPERAND ERROR

CALL  TXBLOCK

JP    ASM1

ASMCE: LD    HL,ASMER3 :COMMAND ERROR

CALL  TXBLOCK

JP    ASM1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

: \*\*\*\* CHECK JR MNEMONIC SUB. \*\*\*\*

: FOR RELOCATE ADDRESS

: IN = (ASMDAT) DIS

: = (ASMBUF) TABLE INDEX

: = (MONDAT) OP-CODE ADDRESS

; OUT = (ASMOPC) OP-CODE

```
ASMREL: LD A,(ASMBUF) ;MNEMONICS INDEX
        CP 29 ;JR
        JR Z,ASMREL1
        CP 16 ;DJNZ
        RET NZ
ASMREL1: LD HL,(MONDAT) ;IS JR MNEMONICS
        LD DE,(ASMDAT)
        CALL ASMRELS ;CALCULATE AND LOAD TO REM
        RET
```

; JUMP RELATIVE CALCULATE SUB.

: IN = HL START

; = DE FINAL

: OUT = (HL+1)

; = A, FF=ERROR

```
ASMRELS: EX DE,HL
```

```
        INC DE
```

```
        INC DE
```

```
        XOR A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SBC HL,DE          ;FINAL-START+2
DEC DE
CP H
JR Z.ASMRELS2
DEC A              ;FF
CP H
JR Z.ASMRELS3

```

```

ASMRELS1: LD A,0FFH ;JUMP ERROR
LD (ASMOPC+1),A
LD HL,JUMPTAB
CALL INBLOCK
LD A,0FFH
RET
ASMRELS2: LD A,7FH ;JUMP TO HIGHER
CP L
JR C,ASMRELS1
JR ASMRELS4
ASMRELS3: CP L ;A=FF
JR Z,ASMRELS1
LD A,7FH
CP L
JR NC,ASMRELS1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASMRELS4: LD  A,L
          LD  (ASMOPC+1),A
          XOR A
          RET

```

```

: **** SEARCH OPERAND SUB. ****

```

```

; IN = (MONINX)

```

```

; OUT = A OPERAND INDEX FF=ERROR

```

```

: (ASMDATD) +D

```

```

; (ASMDAT) N NN (N) (NN) DIS

```

```

ASMOPR: LD  HL,MONBUF
        LD  A,(MONINX)
        LD  D,0
        LD  E,A
        ADD HL,DE
        LD  B,S ;MAX CHAR. OF OPERAND
        CALL ASMMOV

```

```

LD  HL,ASMCOM

```

```

LD  A,(HL)

```

```

CP  "("

```

```

JR  Z,ASMOPR1

```

```

CP  "0"

```

```

JR  C,ASMOPR2 ;NOT VAL

```

```

CP  "9"+1

```

```

JR  NC,ASMOPR2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD C,1 ;N NN DIS
JR ASMOPRV

ASMOPR1: INC HL ;CHECK "("
LD A,(HL)
CP "0"
JR C.ASMOPR12
CP "9"+1
JR NC.ASMOPR12
LD C,2 ;(N) (NN)
JR ASMOPRV
ASMOPR12: CP "I" ;CHECK IX+D IY+D
JR NZ.ASMOPR2 ;NOT IX IY
INC HL
LD D,(HL) ;X,Y
INC HL
LD A,(HL)
CP "+"
JR NZ.ASMOPR2 ;NOT +D
INC HL
LD A,D
LD C,3 ;(IX+D) INDEX
CP "X"
JR Z.ASMOPR13
LD C,10 ;(IY+D) INDEX
CP "Y"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*      JR      Z,ASMOPR13
      JR      ASMOPRE          ;NO + ERROR
*
ASMOPR13: EX  DE,HL
      LD      HL,ASMDATD
      LD      (HL),0
      LD      A,(DE)          ;CHECK NUMBER
      CP      "0"
      JR      C,ASMOPRE      ;NOT VAL
      CP      "9"+1
      JR      NC,ASMOPRE     ;NOT VAL
      LD      B,4            ;CHAR. LIMIT
ASMOPR14: LD      A,(DE)      ;+D
      CP      ")"
      JR      Z,ASMOPR15
      CALL   CHEX
      CP      OFFH
      JR      Z,ASMOPRE
      RLD
      INC    DE
      DJNZ   ASMOPR14
      JR      ASMOPRE
ASMOPR15: LD      A,C          ;IS (IX+D) (IY+D)
      RET
*
ASMOPR2: LD      IX,OPRTAB    ;OPERAND SEARCH
      LD      IY,ASMCOM
*

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD BC,4 ;TABLE LENGTH
LD HL,35
CALL SEAR
CP 0
RET NZ ;FOUND
ASMOPRE: LD A,0FFH ;OPERAND ERROR
RET
:
: VALUE N NN (N) (NN) DIS
:
ASMOPRV: EX DE,HL ;VAL N NN (N) (NN) DIS
LD B,7
ASMOPRV1: LD A,(DE)
CP " " ;END FIELD
JR Z,ASMOPRV9
CP ")" ;END FIELD
JR Z,ASMOPRV9
CP "H"
JR Z,ASMOPRV2
CALL CHEX
CP OFFH
JR Z,ASMOPRE
LD HL,ASMDAT
:
RLD
INC HL
RLD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASMOPRV2: INC DE
           DJNZ ASMOPRV1
           JR ASMOPRE
ASMOPRV9: LD A,C ;IS VAL
           RET

```

```

: **** MOVE TO ASMCOM SUB. ****

```

```

; IN = (HL) START FIELD

```

```

: B COUNT (MAX=8)

```

```

; END BY BLANK.CR OR B=0

```

```

; OUT = (ASMCOM)

```

```

ASMMOV: LD DE,2020H ;CLEAR BY BLANK
         LD (ASMCOM),DE
         LD (ASMCOM+2),DE
         LD (ASMCOM+4),DE
         LD (ASMCOM+6),DE
         LD DE,ASMCOM

```

```

ASMMOV1: LD A,(HL)
          CP " "
          RET Z
          CP 0DH
          RET Z
          LD (DE),A
          INC HL
          INC DE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DJNZ ASMMOV1
```

```
RET
```

```
; ***** BREAK ROUTINE *****
```

```
; SET/CLEAR BREAK
```

```
; B [ADDR]
```

```
BREAK: CP 0
```

```
JR Z,BREAKD
```

```
CALL VAL
```

```
CALL NEXT
```

```
CP 0
```

```
JP NZ,ERROR
```

```
LD HL,(MONDAT)
```

```
LD (BRK).HL
```

```
JP MON
```

```
BREAKD: LD DE.(BRK)
```

```
CALL TXBYTE4
```

```
CALL TXLF
```

```
JP MON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; \*\*\*\*\* COMPARE ROUTINE \*\*\*\*\*

; COMPARE MEMORY

; C RANGE ADDR

; RANGE H BYTE

COMPARE: CALL RANGE

CALL NEXT

CP 0

JP Z,ERROR

LD HL,MONBUF

LD B,0

LD A,(MONINX)

LD C,A

ADD HL,BC

LD A,(HL)

CALL UPPER

CP "H"

JR Z,COMPHEX

CALL VAL

CALL NEXT

CP 0

JP NZ,ERROR

CALL CLEN ;COMPARE BLOCK

LD IX,(MONDAT)

COMPARE1: LD A,(HL)

CP (IX+0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    Z,COMPARE2

PUSH HL                ;NOT EQUAL

POP  DE

CALL TXBYTE4          ;ADDR 1

LD   D,":"

CALL TXBYTE

LD   A,(HL)

LD   D,A

CALL TXBYTE2          ;DATA 1

LD   D," "

CALL TXBYTE

PUSH IX

POP  DE

CALL TXBYTE4          ;ADDR 2

LD   D,":"

CALL TXBYTE

LD   D,(IX+0)

CALL TXBYTE2          ;DATA 2

CALL TXLF

COMPARE2: INC  HL

        INC  IX

        DEC  BC

        LD   A,B

        OR   C

        JR   NZ,COMPARE1

        JP   MON

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

COMPHEX:  INC  HL

           LD   A,(HL)

           CP   " "

           JP   NZ.ERRORX

           CALL NEXT

           CP   0

           JP   Z.ERROR

           CALL VAL

           CALL NEXT

           CP   0

           JP   NZ.ERROR

           CALL CLEN

COMPHEX1: LD   A,(MONDAT)

           CP   (HL)

           JR   Z.COMPHEX2

           PUSH HL

           POP  DE

           CALL TXBYTE4

           LD   D,":"

           CALL TXBYTE

           LD   D,(HL)

           CALL TXBYTE2

           CALL TXLF

COMPHEX2: INC  HL

           DEC  BC

           LD   A,B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OR C
JR NZ,COMPHEX1
JP MON
```

```
; ***** DISPY ROUTINE *****
```

```
: DISPLAY MEMORY
```

```
; D [ADDR1] [ADDR2]
```

```
DISPY: LD HL,127
LD (DEST),HL
CALL AD2
LD IX,(START)
LD C,0
DISPY4: LD B,16 ;START LOOP
PUSH IX
POP DE
PUSH IX
POP IY
CALL TXBYTE4 ;DISPLAY ADDR
LD D,": "
CALL TXBYTE
DISPY5: LD D," "
CALL TXBYTE
LD D,(IX+0)
CALL TXBYTE2 ;DISPLAY DATA
PUSH BC ;CHECK END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD BC.(FINAL)
CALL COMP16
POP BC
JR NZ.DISPY51
LD C.1 ;END FLAG
DISPY51: INC IX
LD A.9
CP B
JR NZ.DISPY6
LD D." "
CALL TXBYTE
LD D."-"
CALL TXBYTE
DISPY6: DJNZ DISPY5
LD D." "
CALL TXBYTE
LD D." "
CALL TXBYTE
LD B.16
DISPY7: LD A,(IX+0)
AND 7FH
CP " "
JR NC,DISPY8
LD D,"."
CALL TXBYTE
JR DISPY9

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DISPY8: LD D,A
        CALL TXBYTE          ;DISPLAY CHAR.
DISPY9: INC IY
        DJNZ DISPY7
        CALL TXLF
        LD A,C
        CP 0
        JR Z,DISPY4
        LD (FINAL),IX
        JP MON
: ***** ENTER ROUTINE *****
; ENTER DATA
; E ADDR
ENTER: CP 0
        JP Z,ERROR
        CALL VAL
        CALL NEXT
        CP 0
        JP NZ,ERROR
        LD IX,(MONDAT)
ENTER1: PUSH IX
        POP DE
        CALL TXBYTE4
        LD D,":"
        CALL TXBYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD D,(IX+0)
CALL TXBYTE2
LD D," "
CALL TXBYTE
CALL TMGETL
LD A,(MONBUF) ;FIRST DIGIT
CP ODH
JP Z.MON
LD HL,MONDAT
LD (HL),0 ;CLEAR DATA
CALL CHEX
CP OFFH
JR Z.ENTERE
RLD
LD A,(MONBUF+1) ;SECOND DIGIT
CP ODH
JR Z,ENTER2
CALL CHEX
CP OFFH
JR Z.ENTERE
RLD
LD A,(MONBUF+2)
CP ODH
JR NZ,ENTERE

```

```

ENTER2: LD A,(MONDAT) ;ENTER
LD (IX+0),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC IX
JR ENTER1

ENTERE: LD D,07H          :ERROR
        CALL TXBYTE
        JR ENTER1

```

```

: ***** FILL ROUTINE *****

```

```

: FILL MEMORY

```

```

: F RANGE BYTE

```

```

FILL:   CALL RANGE
        CALL NEXT
        CP 0
        JP Z,ERROR
        CALL VAL
        CALL NEXT
        CP 0
        JP NZ,ERROR

        CALL CLEN

        LD A,(MONDAT)
        LD D,A

FILL1:  LD (HL),D

        INC HL
        DEC BC
        LD A,B
        OR C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    NZ.FILL1

JP    MON

; ***** GO ROUTINE *****

; RUN USER PROGRAM

; G [ADDR]

GO:   CP    0

      JR    Z.GO1

      CALL VAL

      CALL NEXT

      CP    0

      JP    NZ.ERROR

      LD    HL, (MONDAT)

      LD    (USRPC), HL

GO1:  LD    IX, (USRPC)

      LD    BC, (BRK)

      CALL COMP16

      JR    NC,EXIT

      LD    HL, BRKFAG

      SET  0, (HL)

      LD    HL, (BRK)

      LD    A, (HL)

      LD    (BRKMEM), A

      LD    (HL), 0CFH      ;RST 08H OP-CODE

EXIT: LD    HL, (USRIF)

      LD    A, H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD   I.A           ;LOAD I
DI
BIT  2.L
JR   Z.EXIT1
EI           ;LOAD IFF
EXIT1: LD   SP,USRIX
      POP  IX           ;LOAD IX
      POP  IY           ;LOAD IY
      POP  AF           ;LOAD AF
      POP  BC           ;LOAD BC
      POP  DE           ;LOAD DE
      POP  HL           ;LOAD HL
      EX  AF,AF'
      EXX
      POP  AF           ;LOAD AF'
      POP  BC           ;LOAD BC'
      POP  DE           ;LOAD DE'
      POP  HL           ;LOAD HL'
      EX  AF,AF'
      EXX
      LD   SP,(USRSP)   ;LOAD SP
      PUSH HL
      LD   HL,(USRPC)
      EX  (SP),HL
      RET           ;LOAD PC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; \*\*\*\*\* HELP ROUTINE \*\*\*\*\*

; USER HELP

‡ H

HELP: CP 0

JP NZ.ERROR

LD E.24 ;TAB

LD HL.HELPTB

HELP1: LD A.(HL)

CP 0

JP Z.MON :EXIT

INC HL

CP ODH

JR Z.HELPS

CP "\" :CHECK TAB

JR Z.HELPS

LD D.A

CALL TXBYTE

DEC E

JR HELP1

HELPS: CALL TXLF :LF

LD E.24

JR HELP1

HELPS8: DEC E ;PRINT TAB

JR Z,HELP1

LD D," "

CALL TXBYTE

JR HELPS8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; \*\*\*\*\* INPUT ROUTINE \*\*\*\*\*

; IN PORT

; I PORT

INPUT: CP 0

JP Z,ERROR

CALL VAL

CALL NEXT

CP 0

JP NZ,ERROR

LD A,(MONDAT)

LD C,A

IN D,(C)

CALL TXBYTE2

CALL TXLF

JP MON

; \*\*\*\*\* LOAD ROUTINE \*\*\*\*\*

; LOAD FROM TERMINAL (.HEX FILE)

; L [ADDR]

; ADDR IS OFFSET ADDRESS

LOAD: LD HL,0

LD (MONDAT),HL ;DEFAULT OFFSET

CP 0

JR Z,LOAD1

CALL VAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL NEXT

CP 0

JP NZ,ERROR

LOAD1: LD HL,LOADTAB2

CALL TXBLOCK

LD HL,LOADTAB3

CALL TXBLOCK

CALL RXBYTE ;START READ

CP ":"

JP NZ,LOADE

LD C,0 ;START CHECKSUM

CALL LOADS ;START LINE

LD B,D ;COUNTER

CALL LOADS ;ADDR

LD H,D

CALL LOADS

LD L,D

LD DE,(MONDAT)

ADD HL,DE ;OFFSET ADDR

LD (FINAL),HL

CALL LOADS ;00

LOAD2: CALL LOADS ;DATA

LD (HL),D

INC HL

DJNZ LOAD2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD   A,C           ;CHECKSUM
NEG
LD   C,A
CALL RXBYTE2
CP   C
JR   NZ,LOADE

CALL RXBYTE       ;CR
CALL RXBYTE       ;LF
CALL RXBYTE       ;NEW LINE
CP   ":"
JR   NZ,LOADE
LD   C,0
CALL LOADS
CP   0
JR   Z,LOAD3      ;LAST LINE
LD   B,A
CALL LOADS
LD   H,D
CALL LOADS
LD   L,D
LD   DE,(MONDAT)
ADD  HL,DE        ;OFFSET ADDR
CALL LOADS        ;00
JR   LOAD2
LOAD3: LD B,4      ;END DATA
LOAD31: CALL RXBYTE2 ;00
DJNZ LOAD31

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL RXBYTE          :CR
CALL RXBYTE          ;LF
LD  DE.(FINAL)
CALL TXBYTE4
CALL TXLF
JP  MON

LOADE:  LD  HL,LOADTAB      :LOAD ERROR
CALL TXBLOCK
LOADE1: CALL RXBYTE
CP  ODH
JR  NZ,LOADE1
JP  MON
LOADS:  CALL RXBYTE2      :2 BYTE LOAD SUB.
LD  A,C
ADD  A,D                :CHECKSUM
LD  C,A
RET

; ***** MOVE ROUTINE *****
; *MOVE MEMORY
; M RANGE ADDR

MQVE:  CALL RANGE
CALL NEXT
CP  0
JP  Z,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL VAL
CALL NEXT
CP 0
JP NZ.ERROR
LD BC.(MONDAT)
LD HL.(START)
LD DE,(FINAL)
CALL MOVES
JP MON
: ***** OUTPUT ROUTINE *****
: OUT PCRT
: O PORT BYTE
OUTPUT: CP 0
JP Z.ERROR
CALL VAL
LD A.(MONDAT)
LD C.A
PUSH BC
CALL NEXT
CP 0
JP Z.ERROR.
CALL VAL
CALL NEXT
CP 0
JP NZ.ERROR
POP BC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD A,(MONDAT)

OUT (C),A

JP MON

; \*\*\*\*\* UPLOAD ROUTINE \*\*\*\*\*

; UPLOAD TO TERMINAL (.HEX FILE)

; P RANGE

UPLOAD: CALL RANGE  
CALL NEXT  
CP 0  
JP NZ,ERROR  
LD HL,UPLTB1  
CALL TXBLOCK  
LD HL,UPLTB2  
CALL TXBLOCK  
LD HL,UPLTB3  
CALL TXBLOCK

UPLOAD0: CALL RXBYTE  
CP ODH  
JR NZ,UPLOAD0  
LD HL,(FINAL)  
LD DE,(START)  
XOR A  
SBC HL,DE  
INC HL  
PUSH HL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP IY                ;LENGTH
LD IX,(START)

UPLOAD1: PUSH IY
POP HL
LD DE.32+1
XOR A
SBC HL.DE
JR C.UPLOAD3         ;<=32
LD BC.2000H         ;32 LOOP
CALL UPLOADS
JR UPLOAD1
UPLOAD3: PUSH IY
POP BC
LD B.C
LD C,0
CALL UPLOADS        ;LAST LINE
LD HL,UPLHEX
CALL TXBLOCK
CALL RXBYTE         ;WAIT FOR ANY KEYS
JP MON

UPLOADS: LD D,":"    ;ONE LINE SUB.
CALL TXBYTE         ;START LINE
LD D,B
CALL UPLLOADX
CALL TXBYTE2        ;LINE LENGTH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH IX
POP DE
CALL UPL0ADX
LD D,E
CALL UPL0ADX
PUSH IX
POP DE
CALL TXBYTE4 ;ADDR
LD D,00H
CALL TXBYTE2 ;00
UPL0ADS1: LD D,(IX+0)
CALL UPL0ADX
CALL TXBYTE2
INC IX
DEC IY
DJNZ UPL0ADS1
LD A,C ;CHECKSUM
NEG
LD D,A
CALL TXBYTE2
CALL TXLF
RET

UPL0ADX: LD A,C ;CHECKSUM
ADD A,D
LD C,A
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

: \*\*\*\*\* BACK ROUTINE \*\*\*\*\*

: BACK BY BREAK ADDRESS OR INT KEY

```
BACK:  LD  (USRIX).IX      :SAVE IX
        POP  IX
        LD  (USRPC).IX    :SAVE PC
        LD  (USRSP),SP    :SAVE SP
        LD  SP.USRH1+2
        EXX
        PUSH HL           :SAVE HL
        PUSH DE           :SAVE DE
        PUSH BC           :SAVE BC
        EXX
        EX  AF.AF'
        PUSH AF           :SAVE AF
        EX  AF.AF'
        PUSH HL           :SAVE HL
        PUSH DE           :SAVE DE
        PUSH BC           :SAVE BC
        PUSH AF           :SAVE AF
        PUSH IY           :SAVE IY
        LD  A,I
        LD  (USRIF+1),A   ;SAVE I
        LD  A,0
        JP  PO,BACK1
        LD  A,4
BACK1:  LD  (USRIF),A     ;SAVE IFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD SP, SYSSTK
LD HL, BRKFAG
BIT 1, (HL)
JR Z, BACK4
RES 1, (HL) ; IS TRACE
LD HL, (USRPC)
DEC HL
LD (USRPC), HL
LD HL, (STPADD)
LD A, (BRKMEM)
LD (HL), A
JR BACK6
BACK4: BIT 0, (HL)
JR Z, BACK5
RES 0, (HL) ; IS BREAK
LD HL, (USRPC)
DEC HL
LD (USRPC), HL
LD HL, (BRK)
LD A, (BRKMEM)
LD (HL), A
JR BACK6

BACK5:
BACK6: LD DE, (USRPC)
CALL TXBYTE4
CALL TXLF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP MON

; \*\*\*\*\* REG ROUTINE \*\*\*\*\*

; REGISTOR DISPLAY & SET

; R [REG VALUE]

```
REG:    CP    0
        JP    Z,REGD          ;DISPLAY ONLY
        LD    HL,MONBUF
        LD    B,0
        LD    C,A
        ADD   HL,BC
        PUSH HL
        POP   IX              ;INDEX TO REG. NAME
        CALL NEXT
        CP    0
        JP    Z,ERROR
        PUSH AF
        CALL VAL              :VALUE
        CALL NEXT
        CP    0
        JP    NZ,ERROR       ;FIELD OVER
        POP   AF
        LD    (MONINX),A
        LD    A,(IX+0)       ;TO UPPER AND CHECK NO OF CHAR.
        CALL UPPER
        LD    (IX+0),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD  A,(IX+1)
CP  " "
JR  NZ,REG1
CALL REGX          ;1 CHAR. SEARCH 8 BIT
CP  OFFH
JR  Z,REGERR
JP  MON

```

```

REG1:  CALL UPPER
LD     (IX+1),A
LD     A,(IX+2)
CP     " "
JR     NZ,REG2
CALL  REGX          ;2 CHAR. SEARCH 8 , 16 BIT
CP     0
JP     Z,MON
CALL  REGXX
CP     0
JP     Z,MON
CALL  REGFAG
CP     OFFH
JR     Z,REGERR
JP     MON

```

```

REG2:  CALL UPPER
LD     (IX+2),A
LD     A,(IX+3)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP    " "

JR    NZ,REGERR      ;>3 CHAR.

CALL  REGXX          ;3 CHAR. SEARCH 16 BIT

CP    0

JP    Z.MON

REGERR:  PUSH IX          ;REG. NAME ERROR

        POP HL

        JP  ERRORX

; IN = (IX) REG. NAME
; OUT = A NOT FOUND FLAG

REGXX:   LD  IY,REGTB1    ;SEARCH 16 BIT REG. NAME

        LD  C,0

REGXX1:  PUSH IX

        POP HL

        PUSH IY

        POP DE

        LD  B,3

REGXX2:  LD  A,(DE)

        CP  (HL)

        JR  NZ,REGXX3

        INC HL

        INC DE

        DJNZ REGXX2

        LD  HL,USRPC     ;SEARCH FOUND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD HL,BC          ;B=0
LD DE,(MONDAT)
LD (HL),E
INC HL
LD (HL),D
XOR A
RET                ;FOUND EXIT

```

```

REGXX3: LD DE,3
ADD IY,DE
INC C
INC C
LD A,C
CP 1AH
JR C,REGXX1
LD A,OFFH
RET                ;NOT FOUND EXIT

```

```

; IN = (IX) REG. NAME

```

```

; OUT = A NOT FOUND FLAG

```

```

REGX: LD IY,REGTB3 ;SEARCH 8 BIT REG. NAME
LD BC,0
LD D,(IX+0)
LD E,(IX+1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

REGX:   LD   IY,REGTB3      ;SEARCH 8 BIT REG. NAME
        LD   BC,0
        LD   D,(IX+0)
        LD   E,(IX+1)
REGX1:  CALL REGCOMP
        JR   NZ,REGX2
        LD   HL,USRIF+1    ;SEARCH FOUND
        ADD  HL,BC         ;B=0
        LD   A,(MONDAT)
        LD   (HL),A
        XOR  A
        RET                ;FOUND EXIT
REGX2:  INC  IY
        INC  IY
        INC  C
        LD   A,C
        CP  12H
        JR  C,REGX1
        LD  A,OFFH
        RET                ;NOT FOUND EXIT

```

```

; IN = (IX) FLAG NAME
; OUT = A NOT FOUND FLAG

```

```

REGFAG: LD  D,(IX+0)
        LD  E,(IX+1)

```

```

LD IY,REGTB4
LD BC,087FH
REGFAG1: CALL REGCOMP
JR NZ,REGFAG2
CALL REGFAGD ;SEARCH FOUND
LD A,(USRAF)
AND C ;MARK
OR D ;SET
LD (USRAF),A
XOR A
RET ;FOUND EXIT
REGFAG2: INC IY
INC IY
RRC C
DJNZ REGFAG1
LD A,0FFH
RET ;NOT FOUND EXIT

REGFAGD: LD D,0 ;CHECK FLAG DATA
LD A,(MONDAT)
CP 0
RET Z
LD A,C
CPL
LD D,A ;DATA OUT SET BIT BY C
LD A,(MONDAT)
CP 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET Z
JP ERROR

; COMPARE DE & (IY)

REGCOMP: LD A,D
CP (IY+0)
RET NZ
LD A,E
CP (IY+1)
RET

; ***** REGD *****
REGD: LD IX,REGTB1
LD IY,USRPC
LD C,5 ;COUNT
CALL REGDR ;16BIT REG.
LD C,4
CALL REGDR ;MAIN REG.
LD C,4
CALL REGDR ;ALTERNATE REG.

LD IX,REGTB2 ;FLAG TABLE
LD A,(USRAF) ;FLAG
LD C,A
LD B,2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL REGDFX          ;S Z

SLA C

LD B,1

CALL REGDFX          ;H

SLA C

LD B,3

CALL REGDFX          :P/V N C

CALL TXLF

JP MON

REGDR: LD B,3 ;DISPLAY REG. ONE LINE
REGDR1: LD D,(IX+0)
CALL TXBYTE ;REG. NAME
INC IX
DJNZ REGDR1
LD D,"="
CALL TXBYTE
LD D,(IY+1)
LD E,(IY+0)
CALL TXBYTE4 ;REG. DATA
LD D," "
CALL TXBYTE
INC IY
INC IY
DEC C
JR NZ,REGDR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXLF
RET

; IN = IX TABLE

REGDF: LD D,(IX+0) ;DISPLAY FLAG NAME
CALL TXBYTE
INC IX
REGDF1: LD A,(IX+0)
CP " "
RET Z
INC IX
LD D,A
CALL TXBYTE
JR REGDF1

; IN = B LOOP
; C DATA

REGDFX: CALL REGDF ;DISPLAY FLAG & DATA
LD D,"1"
SLA C
JR C,REGDFX1
LD D,"0"
REGDFX1: CALL TXBYTE
DJNZ REGDFX
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; \*\*\*\*\* NEW ROUTINE \*\*\*\*\*

; CLEAR REGISTOR

; N

NEW: CP 0

JP NZ.ERROR

XOR A ;CLEAR

LD HL,USRH1+2

LD B,22

NEW1: DEC HL

LD (HL),A

DJNZ NEW1

LD HL,USRSTK

LD (USRSP),HL

LD HL,USRSTK

LD (USRPC),HL

LD HL,REGD ;TO DISPLAY

EX (SP),HL

RET

; \*\*\*\*\* SEARCH ROUTINE \*\*\*\*\*

; SEARCH DATA

; S RANGE BYTE ... ( 8 BYTE MAX )

\* SEARCH: CALL RANGE

CALL NEXT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 0
JP Z,ERROR
LD A,8
LD (SEACNT),A
LD IX,SEABUF
SEARCH1: CALL VAL ;STORE DATA TO SEABUF
LD A,(MONDAT)
LD (IX+0),A
INC IX
CALL NEXT
CP 0
JR Z,SEARCH2
LD A,(SEACNT)
DEC A
LD (SEACNT),A
JR NZ,SEARCH1
JP ERROR
SEARCH2: LD A,9
LD HL,SEACNT
SUB (HL)
LD (HL),A ;BYTE COUNT
CALL CLEN
SEARCH3: PUSH HL ;COMPARE SEABUF
PUSH BC
PUSH HL
POP IX
LD DE,SEABUF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A, (SEACNT)

LD B,A

SEARCH31: LD A, (DE)

CP (HL)

JR NZ,SEARCH32

INC HL

INC DE

DJNZ SEARCH31

PUSH IX ;SEARCH FOUND

POP DE

CALL TXBYTE4 ;DISPLAY FIRST ADDR

CALL TXLF

SEARCH32: POP BC

POP HL

INC HL

DEC BC

LD A,B

OR C

JR NZ,SEARCH3

JP MON

```

```

; ***** UNASM ROUTINE *****

```

```

; ZSO UNASSEMBLER

```

```

; U [N] [ADDR1] [ADDR2]

```

```

UNASM: PUSH AF

```

```

XOR A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (UASMFAG),A ;NO DISPLAY ADDRESS
POP AF
CP 0
JR Z,UNASMO
LD HL,MONBUF
LD B,0
LD A,(MONINX)
LD C,A
ADD HL,BC
LD A,(HL)
CALL UPPER
CP "N"
JR NZ,UNASMO
LD A,1
LD (UASMFAG),A ;NO DISPLAY ADDRESS
CALL NEXT
UNASMO: LD HL,16
LD (DEST),HL
CALL AD2
XOR A
LD (UASMCNT),A
LD A,D ;CHECK FINAL SET
CP 0
JR Z,UNASMO1 ;SET
LD A,16 ;END BY LINE COUNT
LD (UASMCNT),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UNASM01: LD IX,(START)

UNASM1: LD A,(UASMFAG)
        CP 0
        JR Z,UNASM11 ;BY PASS
        LD B,9 ;NO DISPLAY ADDRESS & OP-CODE

UNASM10: LD D," "
        CALL TXBYTE
        DJNZ UNASM10
        CALL UASM
        CP OFFH
        JP Z,UNASME
        JR UNASM31

UNASM11: PUSH IX
        POP DE
        CALL TXBYTE4 ;ADDR
        LD D,": "
        CALL TXBYTE
        LD D," "
        CALL TXBYTE
        CALL UASM
        CP OFFH
        JP Z,UNASME ;UASM ERROR
        LD C,5 ;BLANK COUNT (5*3)
        LD A,(IY+3) ;BYTE COUNT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND OFH
LD B,A
PUSH IX
POP HL
UNASM2: LD D,(HL)
CALL TXBYTE2 ;DISPLAY OPCODE
LD D," "
CALL TXBYTE
DEC C
INC HL
DJNZ UNASM2
UNASM3: LD B,3
CALL TXBLANK
DEC C
JR NZ,UNASM3
UNASM31: LD A,(IY+0) ;MNEMONIC INDEX
CALL MNESUB
LD D," "
CALL TXBYTE
LD A,(IY+1) ;OPERAND1 INDEX
CP 0
JR Z,UNASM4
CALL OPRSUB
LD A,(IY+2) ;OPERAND2 INDEX
CP 0
JR Z,UNASM4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PUSH AF
LD D,","
CALL TXBYTE
POP AF
CALL OPRSUB3
```

```
UNASM4: CALL TXLF
LD D,0
LD A,(IY+3)
AND 0FH
LD E,A
ADD IX,DE
LD A,(UASMCNT) ;CHECK END
CP 0
JR Z,UNASM5
DEC A ;END BY LINE COUNT
JR Z,UNASM6
LD (UASMCNT),A
JP UNASM1

UNASM5: LD BC,(FINAL) ;END BY ADDRESS
INC BC
CALL COMP16
JP C,UNASM1

UNASM6: LD (FINAL),IX
JP MON
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UNASME: LD HL,UNASMTAB ;UNASM ERROR

CALL TXBLOCK

JP MON

MNESUB: DEC A ;MNEMONIC PRINT

LD B,A

LD C,4

LD HL,MNETAB

CALL TABLE

LD B,4

MNESUB1: LD D,(HL)

CALL TXBYTE

INC HL

DJNZ MNESUB1

RET

OPRSUB: LD D,A

LD A,(IY+4) ;CHECK NUMBER

BIT 7,A

JR NZ,OPRSUB1

LD A,D

JR OPRSUB3

OPRSUB1: AND 7FH ;IS IM,BIT,RES,SET,RST

LD D,A

LD A,(IY+0)

CP 59

JR Z,OPRSUB2

LD A,D

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL HTOA

LD D,A

CALL TXBYTE

RET

OPRSUB2: CALL TXBYTE2      :RST

LD D,"H"

CALL TXBYTE

RET

OPRSUB3: CP 1 ;CHECK ≠ (#) (IX+D) (IY+D)
JR Z,OPR1
CP 2
JR Z,OPR2
CP 8
JR Z,OPRS
CP 10
JR Z,OPRA
DEC A
LD B,A
LD C,4
LD HL,OPRTAB
CALL TABLE
LD B,4
OPRSUB4: LD D,(HL)
LD A,D
CP " "
RET Z

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXBYTE

INC HL

DJNZ OPRSUB4

RET

*
OPR1: CALL OPRT          ;N NN DIS

RET

OPR2: LD  D,"("          ;(N) (NN)

CALL TXBYTE

CALL OPRT

LD  D,")"

CALL TXBYTE

RET

OPRS: LD  HL,OPRIX        ;(IX+D)

JR  OPRA1

OPRA: LD  HL,OPRIY       ;(IY+D)

OPRA1: CALL TXBLOCK

LD  D,(IX+2)

CALL OPBYTE2

LD  D,"H"

CALL TXBYTE

LD  D,")"

CALL TXBYTE

RET

*
OPRIX: DFB  "(IX+",00H

OPRIY: DFB  "(IY+",00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; DISPLAY TYPE 1,2,3,5.6

OPRT: LD A,(IX+3) ;N NN (N) (NN) DIS

AND OF0H

CP 10H

JR Z,OPRT1

CP 20H

JR Z,OPRT2

CP 30H

JR Z,OPRT3

CP 50H

JR Z,OPRT5

JR OPRT6

OPRT1: LD D,(IX+1) ;N (N)

JR OPRT51

OPRT5: LD D,(IX+3)

OPRT51: CALL OPBYTE2

LD D,"H"

CALL TXBYTE

RET

OPRT2: LD D,(IX+2) ;NN (NN)

LD E,(IX+1)

JR OPRT61

OPRT6: LD D,(IX+3)

LD E,(IX+2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

: OPRT61: CALL OPBYTE4
:
LD D,"H"
CALL TXBYTE
RET .

```

```

OPRT3: PUSH IX ;DIS

```

```
POP HL
```

```
LD D,0
```

```
LD E,(IX+1)
```

```
BIT 7,E
```

```
JR Z.OPRT31
```

```
LD D,OFFH
```

```
OPRT31: ADD HL,DE
```

```
INC HL
```

```
INC HL
```

```
PUSH HL
```

```
POP DE
```

```
CALL OPBYTE4
```

```
LD D,"H"
```

```
CALL TXBYTE
```

```
RET
```

```
; **** OPERAND BYTE2 & BYTE4 SUB. ****
```

```
; CHECK IF FIRST BYTE = A-F THEN ADD 0
```

```
OPBYTE2: LD A,D ;IN = D
```

```
CALL OPBYTES
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR C.OPBYTE21 ;NOT A-F
PUSH DE
LD D,"0"
CALL TXBYTE
POP DE
OPBYTE21: CALL TXBYTE2
RET

```

```

OPBYTE4: LD A,D ;IN = DE
CALL OPBYTES
JR C.OPBYTE41
PUSH DE
LD D,"0"
CALL TXBYTE
POP DE
OPBYTE41: CALL TXBYTE4
RET

```

```

OPBYTES: RRCA
RRCA
RRCA
RRCA
AND OFH
CP OAH
RET

```

```
; ***** VAR ROUTINE *****
```

```
; VARIABLE DISPLAY & SET
```

```
; V [VAR VALUE]
```

```
; VAR = 10,18,20,28,30,38,66
```

```
VAR:      CP    0
          JR    Z,VARD      ;DISPLAY ONLY
          CALL VAL
          LD    HL,(MONDAT)
          LD    A,H
          CP    0
          JP    NZ,ERROR
          LD    A,L      ;CHECK VAR
          LD    HL,VARTAB
          LD    BC,700H
VAR1:     CP    (HL)
          JR    Z,VAR2
          INC  HL
          INC  C
          INC  C
          DJNZ VAR1
          JP    ERROR

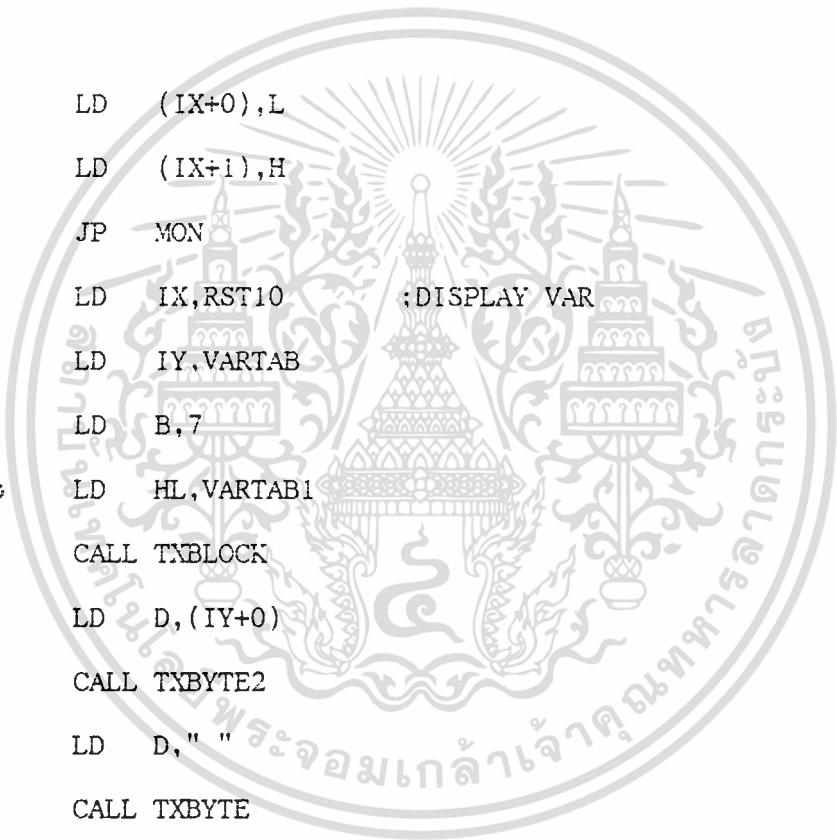
VAR2:     LD    B,0
          LD    IX,RST10
          ADD  IX,BC
          CALL NEXT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 0
JP Z,ERROR
CALL VAL ;VALUE
CALL NEXT
CP 0
JP NZ,ERROR
LD HL,(MONDAT)
LD (IX+0),L
LD (IX+1),H
JP MON
VARD: LD IX,RST10 ;DISPLAY VAR
LD IY,VARTAB
LD B,7
VARD1: LD HL,VARTAB1
CALL TXBLOCK
LD D,(IY+0)
CALL TXBYTE2
LD D," "
CALL TXBYTE
LD E,(IX+0)
LD D,(IX+1)
CALL TXBYTE4
CALL TXLF
INC IY
INC IX
INC IX

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DJNZ VARD1

JP MON

; \*\*\*\*\* TRACE ROUTINE \*\*\*\*\*

; TRACE USER PROGRAM (SENGLE STEP)

; T [ADDR]

TRACE: CP 0

JR Z,TRACE1

CALL VAL

CALL NEXT

CP 0

JP NZ.ERROR

LD HL,(MONDAT)

LD (USRPC),HL

TRACE1: LD IX,(USRPC)

CALL UASM

CP 0

JR Z,TRACE2

LD HL,TRATAB ;COMMAND ERROR

CALL TXBLOCK

JP MON

TRACE2: LD A,(IY+3) ;GO

AND OFH

LD E,A

LD D,0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,(USRPC)

ADD HL,DE

LD A,(HL)

LD (BRKMEM),A

LD (HL).OCFH :RST OSH OP-CODE

LD (STPADD).HL

LD HL,BRKFLAG

SET 1,(HL)

JP EXIT

; ***** QUIT ROUTINE *****
; QUIT (POWER OFF OR RESET)
; Q

QUIT: CP 0
JP NZ,ERROR
LD HL,QUITTB
CALL TXBLOCK
XOR A
LD (POWMEM),A

HALT

```

```

; ***** MOVES SUB. ***** ABCDEHL
; MOVE MEMORY
; IN = HL START ADDR
; DE FINAL ADDR
; BC DESTINATION ADDR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVES:  XOR  A
        EX   DE,HL
        SBC  HL,DE          ;FINAL-START
        PUSH HL            ;3
        PUSH BC            ;2
        PUSH BC            ;1
        POP  HL            ;1
        XOR  A
        SBC  HL,DE          ;DEST-START
        POP  HL            ;2
        EX   DE,HL          ;DEST,START
        POP  BC            ;3
        JR   NC,MOVES1
        INC  BC              ;LENGTH
        LDIR              ;MOVE INC
        RET
MOVES1: ADD  HL,BC          ;START
        EX   DE,HL
        ADD  HL,BC          ;DEST
        EX   DE,HL
        INC  BC
        LDDR
        RET

```

```

; ***** SEAR SUB. ***** ADEHLIX (SEAMEM) (SEAFND) (SEACNT)
; SEARCH STRING
; IN = B FLAG 0=NORMAL 1=BLANK-END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; C TABLE LENGTH
; HL NO. OF STRING IN TABLE
; IX START TABLE
; IY START SEARCH STRING
; OUT = HL NO. OF STRING 0=NOT FOUND
; HL=(SEAFND) , A=L
;

```

```

SEAR: LD (SEAMEM),HL
      LD HL,1
      LD (SEAFND),HL
SEAR1: LD A,C
      LD (SEACNT),A
      PUSH IX
      POP HL
      PUSH IY
      POP DE
SEAR2: LD A,(DE)
      CP (HL)
      JR NZ,SEAR4 ;NOT EQUAL
      INC HL
      INC DE
      LD A,B
      CP 0
      JR Z,SEAR3 ;NORMAL
      LD A,(DE) ;CHECK END BY BLANK
      CP " "
      JR Z,SEARF ;SEARCH FOUND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEAR3:  LD  A,(SEACNT)
        DEC A
        LD  (SEACNT),A
        JR  NZ,SEAR2      ;NEXT CHAR.
        JR  SEARF        ;SEARCH FOUND

```

```

SEAR4:  LD  D,0
        LD  E,C
        ADD IX,DE
        LD  HL,(SEAFND)
        INC HL
        LD  (SEAFND),HL
        LD  HL,(SEAMEM)
        DEC HL
        LD  (SEAMEM),HL
        LD  A,H
        OR  L
        JR  NZ,SEAR1      ;NEXT STRING
        LD  (SEAFND),HL
        RET                ;NOT FOUND

```

```

SEARF:  LD  HL,(SEAFND)    ;FOUND
        LD  A,L
        RET

```

```

; ***** TABLE SUB. ***** ABHL
; TABLE LOOKUP
; IN = B NO. 0 - 255

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;      C TABLE LENGTH
;      HL START TABLE
; OUT = (HL) FIRST DATA IN TABLE

```

```
TABLE:   PUSH DE
```

```
        PUSH HL
```

```
        INC  B
```

```
        LD   D,0
```

```
        LD   E,C
```

```
        LD   HL,0
```

```
TABLE1: DEC  B
```

```
        JR   Z, TABLE2
```

```
        ADD  HL, DE
```

```
        JR   TABLE1
```

```
TABLE2: EX  DE, HL
```

```
        POP  HL
```

```
        ADD  HL, DE
```

```
        POP  DE
```

```
        RET
```

```
; ***** D10M SUB. ***** AB
```

```
; DELAY 0.01 SECOND
```

```
; IN = B COUNT
```

```
D10M:   PUSH HL
```

```
D10M1:  LD   HL,1376
```

```
D10M2:  DEC  HL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD A,H  
OR L  
JR NZ,D10M2  
DJNZ D10M1  
POP HL  
RET

; \*\*\*\*\* DSEC SUB. \*\*\*\*\* AB  
: DELAY SECOND  
; IN = B SECOND COUNT

DSEC: PUSH HL  
DSEC1: LD HL,52640  
DSEC2: PUSH HL  
POP HL  
PUSH HL  
POP HL  
DEC HL  
LD A,H  
OR L  
JR NZ,DSEC2  
DJNZ DSEC1  
POP HL  
RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** COMP16 SUB. ***** A
; COMPARE IX-BC
; IN = IX BC
; OUT = FLAG

```

```
COMP16:  PUSH HL
```

```
        PUSH IX
```

```
        POP HL
```

```
        XOR A
```

```
        SBC HL,BC
```

```
        POP HL
```

```
        RET
```

```
; ***** HTOA SUB. ***** ADE
```

```
; HEX TO ASCII
```

```
; IN = A HEX
```

```
; OUT = DE ASCII (A=E)
```

```
; eg. 2A -> 32 41
```

```
HTOA:  PUSH AF
```

```
        AND OFOH
```

```
        RRCA
```

```
        RRCA
```

```
        RRCA
```

```
        RRCA
```

```
        CALL HTOAS
```

```
        LD  D,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
POP AF
AND 0FH
CALL HTOAS
LD E,A
RET
```

```
HTOAS: AND 0FH
CP 0AH
JR NC,HTOAS1
OR 30H
RET
```

```
HTOAS1: SUB 9
OR 40H
RET
```

```
; ***** ATOH SUB. ***** AD
; ASCII TO HEX
; IN = DE ASCII
; OUT = A HEX
```

```
ATOH: LD A,D
CALL ATOHS
RLCA
RLCA
RLCA
RLCA
LD D,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD A,E
CALL ATOHS
OR D
RET
```

```
; ***** ATOHS SUB. ***** A
```

```
; ASCII TO HEX ONE BYTE
```

```
; IN = A ASCII
```

```
; OUT = A HEX
```

```
ATOHS: CP "A"
JR NC,ATOHS1
AND OFH
RET
```

```
ATOHS1: SUB 7
AND OFH
RET
```

```
; ***** UPPER SUB. ***** A
```

```
; LOWER TO UPPER CHARECTER
```

```
; IN = A LOWER
```

```
; OUT =A UPPER
```

```
UPPER: CP "a"
RET C
CP "z"+1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET NC

SUB 32

RET

\*\*\*\*\* TMGETL SUB. \*\*\*\*\*

; GET LINE FROM TERMINAL ( END BY CR )

; OUT = (MONBUF) COMMAND BUF

; (MONINX) LENGTH

; REG = ABCDHL

TMGETL: XOR A

LD (MONINX),A ;CLEAR INDEX

TMGETL1: CALL RXBYTE

LD A,D ;CHECK KEY

CP 08H

JR Z, TMGETBS

CP 0DH

JR Z, TMGETL2

LD A, (MONINX) ;CHK MONBUF FULL

CP 60 ;AT 61 FOR CR ONLY

JR C, TMGETL2

LD D, 07H ;FULL

CALL TXBYTE

JR TMGETL1

TMGETL2: CALL TXBYTE ;ECHO

LD A, (MONINX) ;UP INDEX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD C,A
INC A
LD (MONINX),A
LD HL,MONBUF ;DATA TO BUF
LD B,0
ADD HL,BC
LD (HL),D
LD A,D
CP ODH
JR NZ,TMGETL1
LD D,0AH
CALL TXBYTE
RET ;EXIT
TMGETBS: LD A,(MONINX) ;BACKSPACE
CP 0
JR Z,TMGETL1
DEC A
LD (MONINX),A
LD D,0SH
CALL TXBYTE
LD D," "
CALL TXBYTE
LD D,0SH
CALL TXBYTE
JR TMGETL1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** RXBYTE SUB *****

; RECEIVE ONE BYTE FROM JACK

; OUT = D DATA (A=D)

; REG = ABDHLIX

ZZZ:          SLP

RXBYTE1:     NOP

RXBYTE:      LD      B,0SH

              LD      E,00H

SCAN1:      LD      A,E

              OUT     (UPC),A

              IN      A,(UPC)

              AND     OFOH

              CP      OFOH

              JR      NZ,SCAN2

              INC     E

              DJNZ   SCAN1

              JR      RXBYTE

SCAN2:      OR      E

              LD      (KEYIN),A

DEBOUND:    IN      A,(UPC)

              AND     OFOH

              CP      OFOH

              JR      NZ,DEBOUND

              LD      A,(KEYIN)

              CP      OBOH

              JR      NZ,KEYCODE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,01H
LD      (KEY_SHIFT),A
JR      RXBYTE

KEYCODE: LD      A,(KEY_SHIFT)
CP      01H
JR      NZ,CODE1
LD      HL,PCTAB
LD      IX,SHIFTTAB
JR      CODE2
CODE1:  LD      HL,PCTAB
LD      IX,NORMALTAB
CODE2:  LD      A,(KEYIN)
LD      B,1FH
CODE2_1: CP      (HL)
JR      Z,CODE3
INC     HL
INC     IX
DJNZ   CODE2_1
CODE3:  LD      A,(IX+0)
LD      D,A
XOR    A
LD      (KEY_SHIFT),A
LD      A,D
OUT    (UPA),A
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** RXBYTE2 SUB. ***** ADE
; RECEIVE ONE ASCII FROM COM PORT
; OUT = D DATA IN HEX (A=D)
RXBYTE2: CALL RXBYTE
        LD  E,D
        CALL RXBYTE
        LD  D,E
        LD  E,A
        CALL ATOH
        LD  D,A
        RET
; ***** TXBYTE SUB *****
; SEND 1 CHARECTER TO RS-232 PORT
; IN = D
; REG = E
TXBYTE:  PUSH DE
TXBYTE1: INO  E,(STAT1) ;CHECK FLAG SEND
        BIT  1,E
        JR   Z,TXBYTE1 ;FLAG TDRE ASCII=1?
        OUTO (TDR1),D ;YES,SEND TO DISPLAY
        POP  DE
        PUSH AF
        PUSH BC
        PUSH DE
        PUSH HL
        CALL DISPLAY
        POP  HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POP DE

POP BC

POP AF

RET

DISPLAY: LD A,0DH ;CR

CP D

JR NZ,DCON1

CALL HOME

RET

DCON1: LD A,0AH ;LF

CP D

JR NZ,DCON2

CALL CLSLCD

RET

DCON2: LD A,07H ;BELL

CP D

JR NZ,DCON3

CALL BELL

RET

DCON3: LD A,08H ;BACK SPACE

CP D

JR NZ,DCONWR

CALL READ

CP 40H

JR NZ,DCON31

LD A,08H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DCON31:  DEC A
          CALL GOTO
          LD D,20H          ;SPACE
          CALL WRBYTE
          CALL READ
          DEC A
          CALL GOTO
          RET

```

```

DCONWR:  CALL READ
          CP   08H
          JR   NZ,DCONWR1
          LD   A,40H
          CALL GOTO
DCONWR1: CP   48H          ;IF OVER
          JP   NZ,DCONWROK
          PUSH DE
          CALL DELY
          LD   H,1
          LD   B,S
DCONWR2: LD   A,H
          CALL GOTO
          CALL READ_DAT
          DEC  H
          LD   A,H
          CALL GOTO
          CALL WRBYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC H
INC H
DJNZ DCONWR2
LD A,40H
CALL GOTO
CALL READ_DAT
LD A,07
CALL GOTO
CALL WRBYTE
LD H,41H
LD B,8
DCONWR3: LD A,H
CALL GOTO
CALL READ_DAT
DEC H
LD A,H
CALL GOTO
CALL WRBYTE
INC H
INC H
DJNZ DCONWR3
LD A,47H
CALL GOTO
POP DE
CALL WRBYTE
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DCONWROK: CALL WRBYTE
           RET
READ_DAT: IN A,(READ_DATA)
           LD D,A
           CALL READ
           RET
DELY:     PUSH AF
           PUSH BC
           LD B,0SFH
DE1:     LD C,0FFH
DE2:     DEC C
           JR NZ,DE2
           DJNZ DE1
           POP BC
           POP AF
           RET
           LD B,01H
BELL1:   LD A,0FFH
BELL2:   CALL BEDEL
           OUT (SPA),A
           OUT (UPA),A
           DEC A
           CP 0
           JR NZ,BELL2
           DJNZ BELL1
           JR BRET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BEDEL: LD H,01H
BEDE1: LD L,S1H
BEDE2: DEC L
JR NZ,BEDE2
DEC H
JR NZ,BEDE1
RET

```

```

BRET:

```

```

POP HL
POP DE
POP BC
POP AF
RET

```

```

; ***** TXBYTE2 SUB. ***** ADE
; SEND ONE BYTE IN ASCII TO COM PORT
; IN = D DATA

```

```

TXBYTE2: LD A,D
CALL HTOA
CALL TXBYTE
LD D,E
CALL TXBYTE
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** TXBYTE4 SUB. ***** ADE
; SEND TWO BYTE IN ASCII TO COM PORT
; IN = BIT 0 (STSFAG)
; DE DATA

TXBYTE4: PUSH DE

CALL TXBYTE2

POP DE

LD D,E

CALL TXBYTE2

RET

; ***** TXBLOCK SUB. ***** ADHL
; SEND BLOCK (END BLOCK BY 00H OR 0DH) TO COM PORT
; IN = HL START BLOCK

TXBLOCK: LD A, (HL)

CP 0

RET Z

CP 0DH

JR Z, TXBLOCK1

LD D,A

CALL TXBYTE

INC HL

JR TXBLOCK

TXBLOCK1: CALL TXLF

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ***** TXBLANK SUB. ***** ABD B=0
; SEND BLANK TO COM PORT
; IN = B BLANK COUNTER
```

```
TXBLANK: LD D," "
          CALL TXBYTE
          DJNZ TXBLANK
          RET
```

```
; ***** TXLF SUB. ***** AD
; SEND CR LF TO COM PORT
```

```
TXLF: LD D,0DH
        CALL TXBYTE
        LD D,0AH
        CALL TXBYTE
        RET
```

```
; ***** ASMS SUB. *****
; Z80 INSTRUCTION TABLE SEARCH (FOR ASSEMBLER)
; IN = (ASMBUF) MNE,OPR1,OPR2
; (ASMDAT) N NN (N) (NN) DIS
; (ASMDATD) +D
; OUT = (ASMOPC) OPCODE
; A OPCODE COUNT (0=NOT FOUND)
```

```

ASMS:   LD   HL,0
        LD   (ASMOPC),HL
        LD   (ASMOPC+2),HL
        LD   IY,ASMOPC
        LD   IX,XX-1      :XX
        LD   BC,0

ASMS1:  CALL ASMSCOMP
        JR   NZ,ASMS12
        LD   (IY+0),C      :OPCODE
        CALL ASMSX
        RET  NZ

ASMS12: LD   DE,5
        ADD  IX,DE
        INC  C
        DJNZ ASMS1
        LD   A,0CBH      ;CB
        LD   (IY+0),A
        LD   BC,0

ASMS2:  CALL ASMSCOMP
        JR   NZ,ASMS2
        LD   (IY+1),C      ;OPCODE
        CALL ASMSX
        RET  NZ

ASMS22: LD   DE,5
        ADD  IX,DE
        INC  C
        DJNZ ASMS2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IX,DD ;DD
LD A,ODDH
LD (IY+0),A
LD B,39
ASMS3: CALL ASMSCOMP
JR NZ,ASMS31
CALL ASMS39
RET NZ
ASMS31: CALL ASMSUP
DJNZ ASMS3
LD IX,ED ;ED
LD A,OEDH
LD (IY+0),A
LD B,56
ASMS32: CALL ASMSCOMP
JR NZ,ASMS33
CALL ASMS39
RET NZ
ASMS33: CALL ASMSUP
DJNZ ASMS32
LD IX,FD ;FD
LD A,OFDH
LD (IY+0),A
LD B,39
ASMS34: CALL ASMSCOMP
JR NZ,ASMS35

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL ASMS39

RET NZ

ASMS35: CALL ASMSUP

DJNZ ASMS34

JR ASMS4

ASMS39: LD A, (IX+0) ;DD ED FD FOUND

LD (IY+1),A ;OPCODE BYTE 2

CALL ASMSX

*
RET

ASMS4: LD IX, DDCB ;DDCB

LD HL, ODDCBH

LD (IY+0),H

LD (IY+1),L

LD B, 31

ASMS41: CALL ASMSCOMP

JR NZ, ASMS42

CALL ASMS49

RET NZ

ASMS42: CALL ASMSUP

DJNZ ASMS41

LD IX, FDCB ;FDCB

LD HL, OFDCBH

LD (IY+0),H

LD (IY+1),L

LD B, 31

ASMS43: CALL ASMSCOMP

JR NZ, ASMS44

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL ASMS49
RET NZ
ASMS44: CALL ASMSUP
DJNZ ASMS43
XOR A ;COMMAND ERROR
RET
ASMS49: LD A,(IX+0) ;DDCB FDCB FOUND
LD (IY+3),A ;OPCODE BYTE 4
ASMSX: LD A,(IX+5) ;CHECK NUMBER
BIT 7,A
JR Z,ASMSX1
AND 7FH ;IS IM,BIT,RES,SET,RST
LD D,A
LD A,(ASMDAT)
CP D
JR Z,ASMSX3
XOR A ;NEXT SEARCH
RET
ASMSX1: LD A,(IX+4) ;CHECK TYPE
AND 0F0H
CP 10H
JR Z,ASMXT1
CP 20H
JR Z,ASMXT2
CP 30H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR Z,ASMXT1
CP 40H
JR Z,ASMXT4
CP 50H
JR Z,ASMXT5
CP 60H
JR Z,ASMXT6
ASMSX2: LD A,(IX+4) ;NO. OF BYTE
AND 0FH
RET ;EXIT
ASMSX3: LD A,(IX+4) ;CHECK BIT.RES.SET & +D
AND 0F0H
CP 40H
JR Z,ASMXT4
JR ASMSX2
ASMXT1: LD A,(ASMDAT)
LD (IY+1),A
JR ASMSX2
ASMXT2: LD HL,(ASMDAT)
LD (IY+1),L
LD (IY+2),H
JR ASMSX2
ASMXT4: LD A,(ASMDATD)
LD (IY+2),A
JR ASMSX2
ASMXT5: LD A,(ASMDATD)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (IY+2),A
LD A,(ASMDAT)
LD (IY+3),A
JR ASMSX2
ASMXT6: LD HL,(ASMDAT)
LD (IY+2),L
LD (IY+3),H
JR ASMSX2

ASMSCOMP: LD A,(ASMBUF) ;COMMAND COMPARE SUB.
CP (IX+1)
RET NZ
LD A,(ASMBUF+1)
CP (IX+2)
RET NZ
LD A,(ASMBUF+2)
CP (IX+3)
RET

ASMSUP: LD DE,6 ;TABLE UP SUB.
ADD IX,DE
RET

; ***** UASM SUB. *****
; Z80 INSTRUCTION TABLE SEARCH (FOR UNASSEMBLER)
; IN = (IX+0) OPCODE
; OUT = (IY+0) TABLE MNE,OPR1,OPR2,TYPE-BYTE,NUM
; A 0=OK FF=ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UASM:  LD  A, (IX+0)

        CP  OCBH

        JR  Z,UASMCB

        CP  ODDH

        JR  Z,UASMDD

        CP  OEDH

        JR  Z,UASMED

        CP  OFDH

        JR  Z,UASMPD

```

```

LD  HL,XX
UASM1: LD  B,A
        LD  C,5
        CALL TABLE
        PUSH HL
        POP  IY
        XOR  A
        RET

```

```

UASMCB: LD  A, (IX+1)      ;CB

        LD  HL,CB

        JR  UASM1

```

```

UASMDD: LD  A, (IX+1)      ;DD

        CP  OCBH

        JR  Z,UASMDDCB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,DD
UASMDD1: LD BC,6
UASMDD2: LD D,(HL)
CP D
JR Z,UASMDD3
JR C,UASMERR
ADD HL,BC
JR UASMDD2
UASMDD3: INC HL
PUSH HL
POP IY
XOR A
RET
UASMDDCB: LD A,(IX+3) ;DDCB
LD HL,DDCB
JR UASMDD1

UASMED: LD A,(IX+1) ;ED
LD HL,ED
JR UASMDD1

UASMFDB: LD A,(IX+1) ;FD
CP OCBH
JR Z,UASMFDCB
LD HL,FD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    UASMDD1

UASMFDCB: LD    A,(IX+3)      ;FDCB
        LD    HL,FDCB
        JR    UASMDD1

UASMERR: LD    A,OFFH        ;NOT FOUND OPCODE
        RET

```

```

; ***** NEXT SUB. *****
; SEARCH NEXT FIELD OF COMMAND LINE
; IN = (MONINX) START SEARCH
; OUT = (MONINX) A
; IF A=0 THEN NO NEXT FIELD & (MONINX) IS END

NEXT: LD    A,(MONINX)
      LD    C,A
      LD    B,0
NEXT1: INC    C          ;SEARCH BLANK
      LD    HL,MONBUF
      ADD   HL,BC
      LD    A,(HL)
      CP   0DH
      JR   Z,NEXTE
      CP   " "
      JR   Z,NEXT2      ;FOUND BLANK
      JR   NEXT1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NEXT2:   INC  C                ;SEARCH NOT BLANK

         LD   HL,MONBUF

         ADD  HL,BC

         LD   A,(HL)

         CP   0DH

         JR   Z,NEXTE

         CP   " "

         JR   Z,NEXT2

         LD   A,C                ;FOUND

         LD   (MONINX),A

         RET

NEXTE:   LD   A,C                ;NO NEXT FIELD

         LD   (MONINX),A

         XOR  A

         RET

; ***** VAL SUB. *****

; GET VALUE FROM COMMAND LINE

; IN = (MONINX) START FIELD

; OUT = (MONDAT)-(MONDAT+1) IN 8 BIT LOOK AT (MONDAT) ONLY

VAL:     LD   A,(MONINX)

         LD   HL,MONBUF

         LD   B,0

         LD   C,A

         ADD  HL,BC

         EX  DE,HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,0
LD (MONDAT),HL ;CLEAR

LD B,6
VAL1: LD A,(DE)
CP ODH
RET Z
CP " "
RET Z ;END FIELD - EXIT
CP "H"
JR Z,VAL2
CALL CHEX
CP OFFH
JP Z.ERROR
LD HL,MONDAT
RLD
INC HL
RLD
VAL2: INC DE
DJNZ VAL1 ;IF B=0 IS OVER DIGIT
JP ERROR

```

```

; ***** AD1 SUB. *****

```

```

; GET [ADDR]

```

```

; IN = (MONINX),A START FIELD

```

```

; OUT = (START)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AD1:      CP      0                ;CHECK ADDR
          RET     Z                ;DEFAULT
          CALL   VAL
          CALL   NEXT
          CP      0
          JP     NZ,ERROR
          LD     HL,(MONDAT)
          LD     (START),HL
          RET

```

```

; ***** AD2 SUB. *****
; GET [ADDR1] [ADDR2]
; IN = (MCNINX) START FIELD
;      (DEST) DEFAULT FINAL LENGTH
; OUT = (START)
;      (FINAL)
;      D DEFAULT USED 0=NO 1=YES

```

```

AD2:      CP      0                ;CHECK ADDR1
          JR     NZ,AD21
          LD     HL,(FINAL)        ;DEFAULT
          LD     (START),HL
          JR     AD22
AD21:     CALL   VAL                ;ADDR1
          CALL   NEXT
          LD     HL,(MONDAT)
          LD     (START),HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 0
JR Z,AD22
CALL VAL ;ADDR2
LD HL,(MONDAT)
LD (FINAL),HL
CALL NEXT
CP 0
JP NZ,ERROR ;FIELD OVER
LD D,0
RET
AD22: LD DE,(DEST) ;DEFAULT
ADD HL,DE
LD (FINAL),HL
LD D,1
RET

```

```

; ***** RANGE SUB. *****
; GET RANGE
; IN = (MONINX) START FIELD
; OUT = (START)
; (FINAL)

```

```

RANGE: CP 0
JP Z,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL VAL
LD HL,(MONDAT)
LD (START),HL
CALL NEXT
CP 0
JP Z.ERROR
CALL VAL
LD HL,(MONDAT)
LD (FINAL),HL
RET
; ***** CHEX SUB. *****
; CHECK HEX DIGIT AND CHANGE TO HEX
; IN = A ASCII DIGIT
; OUT = A HEX
; IF IN=0-9,A-F THEN OUT=HEX ELSE OUT=OFFH
CHEX: CP "0"
JR C,CHEXE
CP "9"+1
JR C,CHEX1 ;IS 0-9
CALL UPPER
CP "A"
JP C,CHEXE
CP "F"+1
JP NC,CHEXE
ADD A,9 ;IS A-F

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHEX1:  AND  OFH
        RET
```

```
CHEXE:  LD  A,OFFH          ;ERROR CODE
        RET
```

```
; ***** CLEN SUB. ***** ABCHL
```

```
; CAL LENGTH (FINAL)-(START)
```

```
; OUT = BC LENGTH
```

```
; HL START
```

```
CLEN:  PUSH DE
        LD  DE,(START)
        LD  HL,(FINAL)
        XOR  A
        SBC  HL,DE
        INC  HL          ;LENGTH
        EX  DE,HL
        PUSH DE
        POP  BC
        POP  DE
        RET
```

```
; ***** AUTOB SUB. *****
```

```
; OUT = (BAUDLY)
```

```
; REG = ALL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AUTOB:   LD   A,64H
          OUT0 (CNTLA1),A      ;TX,RX,Ş BIT,1 STOP
          LD   A,2             ;9600 BAUD AT X'TAL 12.48S Mhz
          OUT0 (CNTLB1),A
          LD   A,Ş            ;RIE ENABLE TO KEYBOARD
          OUT0 (STAT1),A
          LD   A,96H
          LD   (BAUD),A
          RET

; *****PAGE MEMORY ROUTINE*****
; SELECT PAGE MEMORY FOR PROGRAM MMU [SEE APPENDIC (PAGE MEMORY) ]
; X [NUMBER PAGE]

SPM:     CP 0
          JP Z,ERROR
          CALL VAL
          CALL NEXT
          CP 0
          JP NZ,ERROR
          LD A,(MONDAT)
          CP 0
          JP Z,ERROR
          CP 42H
          JP NC,ERROR
          LD HL,PXTAB
          LD IX,MMUTAB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD B,42
SPM1: CP (HL)
JR Z.SPM2
INC HL
INC IX
DJNZ SPM1
SPM2: CALL NEXT
LD A,(IX+0)
OUT0 (BBR),A
JP MON
; ***** ROM TABLE *****
LCDTAB: DFB " Monitor V1.0 "
MN0TAB: DFB " ",0DH
MN1TAB: DFB " Z80180 Monitor V1.0",0DH
MN2TAB: DFB " Industrial Education KMITL.",0DH
MN3TAB: DFB " Electronic & Computer",0DH
MN4TAB: DFB " ED.ENGINEER #15",0DH
BAUTAB: DFB " Baud Rate :",00H
MN5TAB: DFB " ",0DH
MN6TAB: DFB " ",0DH
MN7TAB: DFB " ",0DH
POWTAB: DFB "POWER UP !",0DH
SYNTAB: DFB "^ Syntax Error",0DH
MN8TAB: DFB "ABCDEFGHILMNOPQRSTUVWXYZ"
MN9TAB: DWL ASM,BREAK,COMPARE,DISPY,ENTER
DWL FILL,GO,HELP,INPUT,LOAD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DWL  MOVE,NEW,OUTPUT,UPLOAD,QUIT
DWL  REG,SEARCH,TRACE,UNASM,VAR,SPM
HELPTB: DFB  " A addr\Assembler",ODH
DFB  " B [addr]\Break",ODH
DFB  " C range addr\Compare block",ODH
DFB  "   range H byte\Compare byte",ODH
DFB  " D [addr1] [addr2]\Display memory",ODH
DFB  " E addr\Enter data",ODH
DFB  " F range byte\Fill memory",ODH
DFB  " G [addr]\Go (Run)",ODH
DFB  " H \Help",ODH
DFB  " I port\Input".ODH
DFB  " L [addr]\downLoad from terminal",ODH
DFB  " M range addr\Move",ODH
DFB  " N \New (Clear registor)",ODH
DFB  " O port byte\Output",ODH
DFB  " P range\upload to terminal",ODH
DFB  " Q \Quit/Halt (Power off or Reset)",ODH
DFB  " R [reg value]\Register",ODH
DFB  " S range byte ... (8)\Search",ODH
DFB  " T [addr]\Trace (Single step)",ODH
DFB  " U [N] [addr1] [addr2]\Unassembler",ODH
DFB  " V [var value]\Vector (RSTn)",ODH
DFB  " X [number page]\Select user page(1 - 41)",ODH,00H
UPLTB1: DFB  " Upload INTEL-HEX FILE ...",ODH
UPLTB2: DFB  " Use ^A and CA Command to Upload File (XTALK)",ODH
UPLTB3: DFB  " Type ENTER to Start and Finish",ODH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UPLHEX:  DFB  " :00000001FF".0DH
REGTB1:  DFB  "PC SP IF IX IY AF BC DE HL " ;16 bit reg. table
          DFB  "AF'BC'DE'HL'"
REGTB2:  DFB  "S= Z= H= PV= N= C= " ;flag print table
REGTB3:  DFB  "I          F A C B E D L H " ;8 bit reg. table
          DFB  "F'A'C'B'E'D'L'H'"
REGTB4:  DFB  "SFZF HF PFNFCF" ;flag table
LOADTAB: DFB  " Load Error ... Enter to Continue",0DH
LOADTAB2: DFB  " Download INTEL-HEX FILE ...",0DH
LOADTAB3: DFB  " Use ^A and SE Command to Send File (XTALK)",0DH
UNASMTAB: DFB  " Opcode Error",0DH
ASMER1:  DFB  " Mnemonic Error",0DH
ASMER2:  DFB  " Operand Error",0DH
ASMER3:  DFB  " Command Error".0DH
JUMPTAB: DFB  " Relative Error",0DH
VARTAB:  DFB  10H,18H,20H,28H,30H,38H,66H
VARTAB1: DFB  "RST",00H
TRATAB:  DFB  "Command Error",0DH
!QUITTB: DFB  "ZS0180 QUIT/HALT ! Power off or Reset ...".0DH

PCTAB:   DFB  0E0H,0E1H,0E2H,0E3H,0E4H,0E5H,0E6H,0E7H
          DFB  0D0H,0D1H,0D2H,0D3H,0D4H,0D5H,0D6H,0D7H
          DFB  0B1H,0B2H,0B3H,0B4H,0B5H,0B6H,0B7H
          DFB  070H,071H,072H,073H,074H,075H,076H,077H

NORMALTAB: DFB  1BH,30H,31H,32H,33H,34H,35H,36H
            DFB  37H,38H,39H,41H,42H,43H,44H,45H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DFB          46H,47H,48H,49H,4AH,4BH,4CH
DFB          4DH,4EH,4FH,20H,50H,51H,52H,0DH

SHIFTTAB:   DFB          1BH,23H,2AH,2SH,29H,2DH,2BH,08H
DFB          0FFH,2CH,3AH,22H,27H,2EH,41H,3DH
DFB          53H,54H,55H,56H,44H,57H,43H
DFB          58H,59H,5AH,20H,2FH,3FH,42H,0DH

PXTAB:      DFB          01H,02H,03H,04H,05H,06H,07H,08H
DFB          09H,10H,11H,12H,13H,14H,15H,16H
DFB          17H,18H,19H,20H,21H,22H,23H,24H
DFB          25H,26H,27H,28H,29H,30H,31H,32H
DFB          33H,34H,35H,36H,37H,38H,39H,40H
DFB          41H

MMUTAB:     DFB          00H,06H,0CH,12H,18H,1EH,24H,2AH
DFB          30H,36H,3CH,42H,48H,4EH,54H,5AH
DFB          60H,66H,6CH,72H,78H,7EH,84H,8AH
DFB          90H,96H,9CH,0A2H,0A8H,0AEH,0B4H
DFB          0C0H,0C6H,0CCH,0D2H,0D8H,0DEH
DFB          0E4H,0EAH,0FOH

```

; \*\*\*\*\* ZSO ROM TABLE \*\*\*\*\*

```

XX:         DFB          24H,00H,00H,01H,00H
DFB          1EH,10H,01H,23H,00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH,03H,0CH,01H,00H  
DFB 17H,10H,00H,01H,00H  
DFB 17H,0FH,00H,01H,00H  
DFB 0EH,0FH,00H,01H,00H  
DFB 1EH,0FH,01H,12H,00H  
DFB 34H,00H,00H,01H,00H  
DFB 12H,0DH,0EH,01H,00H  
DFB 02H,16H,10H,01H,00H  
DFB 1EH,0CH,03H,01H,00H  
DFB 0EH,10H,00H,01H,00H  
DFB 17H,11H,00H,01H,00H  
DFB 0EH,11H,00H,01H,00H  
DFB 1EH,11H,01H,12H,00H  
DFB 39H,00H,00H,01H,00H  
DFB 10H,01H,00H,32H,00H  
DFB 1EH,13H,01H,23H,00H  
DFB 1EH,05H,0CH,01H,00H  
DFB 17H,13H,00H,01H,00H  
DFB 17H,12H,00H,01H,00H  
DFB 0EH,12H,00H,01H,00H  
DFB 1EH,12H,01H,12H,00H  
DFB 32H,00H,00H,01H,00H  
DFB 1DH,01H,00H,32H,00H  
DFB 02H,16H,13H,01H,00H  
DFB 1EH,0CH,05H,01H,00H  
DFB 0EH,13H,00H,01H,00H  
DFB 17H,14H,00H,01H,00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 0EH, 14H, 00H, 01H, 00H  
DFB 1EH, 14H, 01H, 12H, 00H  
DFB 37H, 00H, 00H, 01H, 00H  
DFB 1DH, 1DH, 01H, 32H, 00H  
DFB 1EH, 16H, 01H, 23H, 00H  
DFB 1EH, 02H, 16H, 23H, 00H  
DFB 17H, 16H, 00H, 01H, 00H  
DFB 17H, 15H, 00H, 01H, 00H  
DFB 0EH, 15H, 00H, 01H, 00H  
DFB 1EH, 15H, 01H, 12H, 00H  
DFB 0DH, 00H, 00H, 01H, 00H  
DFB 1DH, 23H, 01H, 32H, 00H  
DFB 02H, 16H, 16H, 01H, 00H  
DFB 1EH, 16H, 02H, 23H, 00H  
DFB 0EH, 16H, 00H, 01H, 00H  
DFB 17H, 1AH, 00H, 01H, 00H  
DFB 0EH, 1AH, 00H, 01H, 00H  
DFB 1EH, 1AH, 01H, 12H, 00H  
DFB 0CH, 00H, 00H, 01H, 00H  
DFB 1DH, 1CH, 01H, 32H, 00H  
DFB 1EH, 22H, 01H, 23H, 00H  
DFB 1EH, 02H, 0CH, 23H, 00H  
DFB 17H, 22H, 00H, 01H, 00H  
DFB 17H, 06H, 00H, 01H, 00H  
DFB 0EH, 06H, 00H, 01H, 00H  
DFB 1EH, 06H, 01H, 12H, 00H  
DFB 3DH, 00H, 00H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1DH, 11H, 01H, 32H, 00H  
DFB 02H, 16H, 22H, 01H, 00H  
DFB 1EH, 0CH, 02H, 23H, 00H  
DFB 0EH, 22H, 00H, 01H, 00H  
DFB 17H, 0CH, 00H, 01H, 00H  
DFB 0EH, 0CH, 00H, 01H, 00H  
DFB 1EH, 0CH, 01H, 12H, 00H  
DFB 06H, 00H, 00H, 01H, 00H  
DFB 1EH, 0FH, 0FH, 01H, 00H  
DFB 1EH, 0FH, 11H, 01H, 00H  
DFB 1EH, 0FH, 12H, 01H, 00H  
DFB 1EH, 0FH, 14H, 01H, 00H  
DFB 1EH, 0FH, 15H, 01H, 00H  
DFB 1EH, 0FH, 1AH, 01H, 00H  
DFB 1EH, 0FH, 06H, 01H, 00H  
DFB 1EH, 0FH, 0CH, 01H, 00H  
DFB 1EH, 11H, 0FH, 01H, 00H  
DFB 1EH, 11H, 11H, 01H, 00H  
DFB 1EH, 11H, 12H, 01H, 00H  
DFB 1EH, 11H, 14H, 01H, 00H  
DFB 1EH, 11H, 15H, 01H, 00H  
DFB 1EH, 11H, 1AH, 01H, 00H  
DFB 1EH, 11H, 06H, 01H, 00H  
DFB 1EH, 11H, 0CH, 01H, 00H  
DFB 1EH, 12H, 0FH, 01H, 00H  
DFB 1EH, 12H, 11H, 01H, 00H  
DFB 1EH, 12H, 12H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH, 12H, 14H, 01H, 00H  
DFB 1EH, 12H, 15H, 01H, 00H  
DFB 1EH, 12H, 1AH, 01H, 00H  
DFB 1EH, 12H, 06H, 01H, 00H  
DFB 1EH, 12H, 0CH, 01H, 00H  
DFB 1EH, 14H, 0FH, 01H, 00H  
DFB 1EH, 14H, 11H, 01H, 00H  
DFB 1EH, 14H, 12H, 01H, 00H  
DFB 1EH, 14H, 14H, 01H, 00H  
DFB 1EH, 14H, 15H, 01H, 00H  
DFB 1EH, 14H, 1AH, 01H, 00H  
DFB 1EH, 14H, 06H, 01H, 00H  
DFB 1EH, 14H, 0CH, 01H, 00H  
DFB 1EH, 15H, 0FH, 01H, 00H  
DFB 1EH, 15H, 11H, 01H, 00H  
DFB 1EH, 15H, 12H, 01H, 00H  
DFB 1EH, 15H, 14H, 01H, 00H  
DFB 1EH, 15H, 15H, 01H, 00H  
DFB 1EH, 15H, 1AH, 01H, 00H  
DFB 1EH, 15H, 06H, 01H, 00H  
DFB 1EH, 15H, 0CH, 01H, 00H  
DFB 1EH, 1AH, 0FH, 01H, 00H  
DFB 1EH, 1AH, 11H, 01H, 00H  
DFB 1EH, 1AH, 12H, 01H, 00H  
DFB 1EH, 1AH, 14H, 01H, 00H  
DFB 1EH, 1AH, 15H, 01H, 00H  
DFB 1EH, 1AH, 1AH, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH, 1AH, 06H, 01H, 00H  
DFB 1EH, 1AH, 0CH, 01H, 00H  
DFB 1EH, 06H, 0FH, 01H, 00H  
DFB 1EH, 06H, 11H, 01H, 00H  
DFB 1EH, 06H, 12H, 01H, 00H  
DFB 1EH, 06H, 14H, 01H, 00H  
DFB 1EH, 06H, 15H, 01H, 00H  
DFB 1EH, 06H, 1AH, 01H, 00H  
DFB 14H, 00H, 00H, 01H, 00H  
DFB 1EH, 06H, 0CH, 01H, 00H  
DFB 1EH, 0CH, 0FH, 01H, 00H  
DFB 1EH, 0CH, 11H, 01H, 00H  
DFB 1EH, 0CH, 12H, 01H, 00H  
DFB 1EH, 0CH, 14H, 01H, 00H  
DFB 1EH, 0CH, 15H, 01H, 00H  
DFB 1EH, 0CH, 1AH, 01H, 00H  
DFB 1EH, 0CH, 06H, 01H, 00H  
DFB 1EH, 0CH, 0CH, 01H, 00H  
DFB 02H, 0CH, 0FH, 01H, 00H  
DFB 02H, 0CH, 11H, 01H, 00H  
DFB 02H, 0CH, 12H, 01H, 00H  
DFB 02H, 0CH, 14H, 01H, 00H  
DFB 02H, 0CH, 15H, 01H, 00H  
DFB 02H, 0CH, 1AH, 01H, 00H  
DFB 02H, 0CH, 06H, 01H, 00H  
DFB 02H, 0CH, 0CH, 01H, 00H  
DFB 01H, 0CH, 0FH, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 01H,0CH,11H,01H,00H  
DFB 01H,0CH,12H,01H,00H  
DFB 01H,0CH,14H,01H,00H  
DFB 01H,0CH,15H,01H,00H  
DFB 01H,0CH,1AH,01H,00H  
DFB 01H,0CH,06H,01H,00H  
DFB 01H,0CH,0CH,01H,00H  
DFB 42H,0FH,00H,01H,00H  
DFB 42H,11H,00H,01H,00H  
DFB 42H,12H,00H,01H,00H  
DFB 42H,14H,00H,01H,00H  
DFB 42H,15H,00H,01H,00H  
DFB 42H,1AH,00H,01H,00H  
DFB 42H,06H,00H,01H,00H  
DFB 42H,0CH,00H,01H,00H  
DFB 3CH,0CH,0FH,01H,00H  
DFB 3CH,0CH,11H,01H,00H  
DFB 3CH,0CH,12H,01H,00H  
DFB 3CH,0CH,14H,01H,00H  
DFB 3CH,0CH,15H,01H,00H  
DFB 3CH,0CH,1AH,01H,00H  
DFB 3CH,0CH,06H,01H,00H  
DFB 3CH,0CH,0CH,01H,00H  
DFB 03H,0FH,00H,01H,00H  
DFB 03H,11H,00H,01H,00H  
DFB 03H,12H,00H,01H,00H  
DFB 03H,14H,00H,01H,00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 03H,15H,00H,01H,00H  
DFB 03H,1AH,00H,01H,00H  
DFB 03H,06H,00H,01H,00H  
DFB 03H,0CH,00H,01H,00H  
DFB 43H,0FH,00H,01H,00H  
DFB 43H,11H,00H,01H,00H  
DFB 43H,12H,00H,01H,00H  
DFB 43H,14H,00H,01H,00H  
DFB 43H,15H,00H,01H,00H  
DFB 43H,1AH,00H,01H,00H  
DFB 43H,06H,00H,01H,00H  
DFB 43H,0CH,00H,01H,00H  
DFB 25H,0FH,00H,01H,00H  
DFB 25H,11H,00H,01H,00H  
DFB 25H,12H,00H,01H,00H  
DFB 25H,14H,00H,01H,00H  
DFB 25H,15H,00H,01H,00H  
DFB 25H,1AH,00H,01H,00H  
DFB 25H,06H,00H,01H,00H  
DFB 25H,0CH,00H,01H,00H  
DFB 07H,0FH,00H,01H,00H  
DFB 07H,11H,00H,01H,00H  
DFB 07H,12H,00H,01H,00H  
DFB 07H,14H,00H,01H,00H  
DFB 07H,15H,00H,01H,00H  
DFB 07H,1AH,00H,01H,00H  
DFB 07H,06H,00H,01H,00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 07H, 0CH, 00H, 01H, 00H  
DFB 2EH, 1DH, 00H, 01H, 00H  
DFB 2BH, 10H, 00H, 01H, 00H  
DFB 1CH, 1DH, 01H, 23H, 00H  
DFB 1CH, 01H, 00H, 23H, 00H  
DFB 05H, 1DH, 01H, 23H, 00H  
DFB 2CH, 10H, 00H, 01H, 00H  
DFB 02H, 0CH, 01H, 12H, 00H  
DFB 3BH, 01H, 00H, 01H, 50H  
DFB 2EH, 23H, 00H, 01H, 00H  
DFB 2EH, 00H, 00H, 01H, 00H  
DFB 1CH, 23H, 01H, 23H, 00H  
DFB 00H, 00H, 00H, 00H, 00H  
DFB 05H, 23H, 01H, 23H, 00H  
DFB 05H, 01H, 00H, 23H, 00H  
DFB 01H, 0CH, 01H, 12H, 00H  
DFB 3BH, 01H, 00H, 01H, 55H  
DFB 2EH, 1CH, 00H, 01H, 00H  
DFB 2BH, 13H, 00H, 01H, 00H  
DFB 1CH, 1CH, 01H, 23H, 00H  
DFB 28H, 02H, 0CH, 12H, 00H  
DFB 05H, 1CH, 01H, 23H, 00H  
DFB 2CH, 13H, 00H, 01H, 00H  
DFB 42H, 01H, 00H, 12H, 00H  
DFB 3BH, 01H, 00H, 01H, 90H  
DFB 2EH, 11H, 00H, 01H, 00H  
DFB 13H, 00H, 00H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1CH, 11H, 01H, 23H, 00H  
DFB 16H, 0CH, 02H, 12H, 00H  
DFB 05H, 11H, 01H, 23H, 00H  
DFB 00H, 00H, 00H, 00H, 00H  
DFB 3CH, 0CH, 01H, 12H, 00H  
DFB 3BH, 01H, 00H, 01H, 9SH  
DFB 2EH, 20H, 00H, 01H, 00H  
DFB 2BH, 16H, 00H, 01H, 00H  
DFB 1CH, 20H, 01H, 23H, 00H  
DFB 12H, 0BH, 16H, 01H, 00H  
DFB 05H, 20H, 01H, 23H, 00H  
DFB 2CH, 16H, 00H, 01H, 00H  
DFB 03H, 01H, 00H, 12H, 00H  
DFB 3BH, 01H, 00H, 01H, 0A0H  
DFB 2EH, 1FH, 00H, 01H, 00H  
DFB 1CH, 06H, 00H, 01H, 00H  
DFB 1CH, 1FH, 01H, 23H, 00H  
DFB 12H, 13H, 16H, 01H, 00H  
DFB 05H, 1FH, 01H, 23H, 00H  
DFB 00H, 00H, 00H, 00H, 00H  
DFB 43H, 01H, 00H, 12H, 00H  
DFB 3BH, 01H, 0QH, 01H, 0ASH  
DFB 2EH, 1EH, 00H, 01H, 00H  
DFB 2BH, 0DH, 00H, 01H, 00H  
DFB 1CH, 1EH, 01H, 23H, 00H  
DFB 0FH, 00H, 00H, 01H, 00H  
DFB 05H, 1EH, 01H, 23H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 2CH,0DH,00H,01H,00H  
DFB 25H,01H,00H,12H,00H  
DFB 3BH,01H,00H,01H,0BOH  
DFB 2EH,1BH,00H,01H,00H  
DFB 1EH,22H,16H,01H,00H  
DFB 1CH,1BH,01H,23H,00H  
DFB 11H,00H,00H,01H,00H  
DFB 05H,1BH,01H,23H,00H  
DFB 00H,00H,00H,00H,00H  
DFB 07H,01H,00H,12H,00H  
DFB 3BH,01H,00H,01H,0BSH  
DFB 33H,0FH,00H,02H,00H  
DFB 33H,11H,00H,02H,00H  
DFB 33H,12H,00H,02H,00H  
DFB 33H,14H,00H,02H,00H  
DFB 33H,15H,00H,02H,00H  
DFB 33H,1AH,00H,02H,00H  
DFB 33H,06H,00H,02H,00H  
DFB 33H,0CH,00H,02H,00H  
DFB 38H,0FH,00H,02H,00H  
DFB 38H,11H,00H,02H,00H  
DFB 38H,12H,00H,02H,00H  
DFB 38H,14H,00H,02H,00H  
DFB 38H,15H,00H,02H,00H  
DFB 38H,1AH,00H,02H,00H  
DFB 38H,06H,00H,02H,00H

CB:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 38H, 0CH, 00H, 02H, 00H  
DFB 31H, 0FH, 00H, 02H, 00H  
DFB 31H, 11H, 00H, 02H, 00H  
DFB 31H, 12H, 00H, 02H, 00H  
DFB 31H, 14H, 00H, 02H, 00H  
DFB 31H, 15H, 00H, 02H, 00H  
DFB 31H, 1AH, 00H, 02H, 00H  
DFB 31H, 06H, 00H, 02H, 00H  
DFB 31H, 0CH, 00H, 02H, 00H  
DFB 36H, 0FH, 00H, 02H, 00H  
DFB 36H, 11H, 00H, 02H, 00H  
DFB 36H, 12H, 00H, 02H, 00H  
DFB 36H, 14H, 00H, 02H, 00H  
DFB 36H, 15H, 00H, 02H, 00H  
DFB 36H, 1AH, 00H, 02H, 00H  
DFB 36H, 06H, 00H, 02H, 00H  
DFB 36H, 0CH, 00H, 02H, 00H  
DFB 3FH, 0FH, 00H, 02H, 00H  
DFB 3FH, 11H, 00H, 02H, 00H  
DFB 3FH, 12H, 00H, 02H, 00H  
DFB 3FH, 14H, 00H, 02H, 00H  
DFB 3FH, 15H, 00H, 02H, 00H  
DFB 3FH, 1AH, 00H, 02H, 00H  
DFB 3FH, 06H, 00H, 02H, 00H  
DFB 3FH, 0CH, 00H, 02H, 00H  
DFB 40H, 0FH, 00H, 02H, 00H  
DFB 40H, 11H, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 40H, 12H, 00H, 02H, 00H  
 DFB 40H, 14H, 00H, 02H, 00H  
 DFB 40H, 15H, 00H, 02H, 00H  
 DFB 40H, 1AH, 00H, 02H, 00H  
 DFB 40H, 06H, 00H, 02H, 00H  
 DFB 40H, 0CH, 00H, 02H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 41H, 0FH, 00H, 02H, 00H  
 DFB 41H, 11H, 00H, 02H, 00H  
 DFB 41H, 12H, 00H, 02H, 00H  
 DFB 41H, 14H, 00H, 02H, 00H  
 DFB 41H, 15H, 00H, 02H, 00H  
 DFB 41H, 1AH, 00H, 02H, 00H  
 DFB 41H, 06H, 00H, 02H, 00H  
 DFB 41H, 0CH, 00H, 02H, 00H  
 DFB 04H, 01H, 0FH, 02H, 80H  
 DFB 04H, 01H, 11H, 02H, 80H  
 DFB 04H, 01H, 12H, 02H, 80H  
 DFB 04H, 01H, 14H, 02H, 80H  
 DFB 04H, 01H, 15H, 02H, 80H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 04H,01H,1AH,02H,80H  
DFB 04H,01H,06H,02H,80H  
DFB 04H,01H,0CH,02H,80H  
DFB 04H,01H,0FH,02H,81H  
DFB 04H,01H,11H,02H,81H  
DFB 04H,01H,12H,02H,81H  
DFB 04H,01H,14H,02H,81H  
DFB 04H,01H,15H,02H,81H  
DFB 04H,01H,1AH,02H,81H  
DFB 04H,01H,06H,02H,81H  
DFB 04H,01H,0CH,02H,81H  
DFB 04H,01H,0FH,02H,82H  
DFB 04H,01H,11H,02H,82H  
DFB 04H,01H,12H,02H,82H  
DFB 04H,01H,14H,02H,82H  
DFB 04H,01H,15H,02H,82H  
DFB 04H,01H,1AH,02H,82H  
DFB 04H,01H,06H,02H,82H  
DFB 04H,01H,0CH,02H,82H  
DFB 04H,01H,0FH,02H,83H  
DFB 04H,01H,11H,02H,83H  
DFB 04H,01H,12H,02H,83H  
DFB 04H,01H,14H,02H,83H  
DFB 04H,01H,15H,02H,83H  
DFB 04H,01H,1AH,02H,83H  
DFB 04H,01H,06H,02H,83H  
DFB 04H,01H,0CH,02H,83H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 04H, 01H, 0FH, 02H, 84H  
DFB 04H, 01H, 11H, 02H, 84H  
DFB 04H, 01H, 12H, 02H, 84H  
DFB 04H, 01H, 14H, 02H, 84H  
DFB 04H, 01H, 15H, 02H, 84H  
DFB 04H, 01H, 1AH, 02H, 84H  
DFB 04H, 01H, 06H, 02H, 84H  
DFB 04H, 01H, 0CH, 02H, 84H  
DFB 04H, 01H, 0FH, 02H, 85H  
DFB 04H, 01H, 11H, 02H, 85H  
DFB 04H, 01H, 12H, 02H, 85H  
DFB 04H, 01H, 14H, 02H, 85H  
DFB 04H, 01H, 15H, 02H, 85H  
DFB 04H, 01H, 1AH, 02H, 85H  
DFB 04H, 01H, 06H, 02H, 85H  
DFB 04H, 01H, 0CH, 02H, 85H  
DFB 04H, 01H, 0FH, 02H, 86H  
DFB 04H, 01H, 11H, 02H, 86H  
DFB 04H, 01H, 12H, 02H, 86H  
DFB 04H, 01H, 14H, 02H, 86H  
DFB 04H, 01H, 15H, 02H, 86H  
DFB 04H, 01H, 1AH, 02H, 86H  
DFB 04H, 01H, 06H, 02H, 86H  
DFB 04H, 01H, 0CH, 02H, 86H  
DFB 04H, 01H, 0FH, 02H, 87H  
DFB 04H, 01H, 11H, 02H, 87H  
DFB 04H, 01H, 12H, 02H, 87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 04H,01H,14H,02H,87H  
DFB 04H,01H,15H,02H,87H  
DFB 04H,01H,1AH,02H,87H  
DFB 04H,01H,06H,02H,87H  
DFB 04H,01H,0CH,02H,87H  
DFB 2DH,01H,0FH,02H,80H  
DFB 2DH,01H,11H,02H,80H  
DFB 2DH,01H,12H,02H,80H  
DFB 2DH,01H,14H,02H,80H  
DFB 2DH,01H,15H,02H,80H  
DFB 2DH,01H,1AH,02H,80H  
DFB 2DH,01H,06H,02H,80H  
DFB 2DH,01H,0CH,02H,80H  
DFB 2DH,01H,0FH,02H,81H  
DFB 2DH,01H,11H,02H,81H  
DFB 2DH,01H,12H,02H,81H  
DFB 2DH,01H,14H,02H,81H  
DFB 2DH,01H,15H,02H,81H  
DFB 2DH,01H,1AH,02H,81H  
DFB 2DH,01H,06H,02H,81H  
DFB 2DH,01H,0CH,02H,81H  
DFB 2DH,01H,0FH,02H,82H  
DFB 2DH,01H,11H,02H,82H  
DFB 2DH,01H,12H,02H,82H  
DFB 2DH,01H,14H,02H,82H  
DFB 2DH,01H,15H,02H,82H  
DFB 2DH,01H,1AH,02H,82H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 2DH, 01H, 06H, 02H, S2H  
DFB 2DH, 01H, 0CH, 02H, S2H  
DFB 2DH, 01H, 0FH, 02H, S3H  
DFB 2DH, 01H, 11H, 02H, S3H  
DFB 2DH, 01H, 12H, 02H, S3H  
DFB 2DH, 01H, 14H, 02H, S3H  
DFB 2DH, 01H, 15H, 02H, S3H  
DFB 2DH, 01H, 1AH, 02H, S3H  
DFB 2DH, 01H, 06H, 02H, S3H  
DFB 2DH, 01H, 0CH, 02H, S3H  
DFB 2DH, 01H, 0FH, 02H, S4H  
DFB 2DH, 01H, 11H, 02H, S4H  
DFB 2DH, 01H, 12H, 02H, S4H  
DFB 2DH, 01H, 14H, 02H, S4H  
DFB 2DH, 01H, 15H, 02H, S4H  
DFB 2DH, 01H, 1AH, 02H, S4H  
DFB 2DH, 01H, 06H, 02H, S4H  
DFB 2DH, 01H, 0CH, 02H, S4H  
DFB 2DH, 01H, 0FH, 02H, S5H  
DFB 2DH, 01H, 11H, 02H, S5H  
DFB 2DH, 01H, 12H, 02H, S5H  
DFB 2DH, 01H, 14H, 02H, S5H  
DFB 2DH, 01H, 15H, 02H, S5H  
DFB 2DH, 01H, 1AH, 02H, S5H  
DFB 2DH, 01H, 06H, 02H, S5H  
DFB 2DH, 01H, 0CH, 02H, S5H  
DFB 2DH, 01H, 0FH, 02H, S6H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 2DH,01H,11H,02H,86H  
DFB 2DH,01H,12H,02H,86H  
DFB 2DH,01H,14H,02H,86H  
DFB 2DH,01H,15H,02H,86H  
DFB 2DH,01H,1AH,02H,86H  
DFB 2DH,01H,06H,02H,86H  
DFB 2DH,01H,0CH,02H,86H  
DFB 2DH,01H,0FH,02H,87H  
DFB 2DH,01H,11H,02H,87H  
DFB 2DH,01H,12H,02H,87H  
DFB 2DH,01H,14H,02H,87H  
DFB 2DH,01H,15H,02H,87H  
DFB 2DH,01H,1AH,02H,87H  
DFB 2DH,01H,06H,02H,87H  
DFB 2DH,01H,0CH,02H,87H  
DFB 3EH,01H,0FH,02H,80H  
DFB 3EH,01H,11H,02H,80H  
DFB 3EH,01H,12H,02H,80H  
DFB 3EH,01H,14H,02H,80H  
DFB 3EH,01H,15H,02H,80H  
DFB 3EH,01H,1AH,02H,80H  
DFB 3EH,01H,06H,02H,80H  
DFB 3EH,01H,0CH,02H,80H  
DFB 3EH,01H,0FH,02H,81H  
DFB 3EH,01H,11H,02H,81H  
DFB 3EH,01H,12H,02H,81H  
DFB 3EH,01H,14H,02H,81H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3EH,01H,15H,02H,S1H  
DFB 3EH,01H,1AH,02H,S1H  
DFB 3EH,01H,06H,02H,S1H  
DFB 3EH,01H,0CH,02H,S1H  
DFB 3EH,01H,0FH,02H,S2H  
DFB 3EH,01H,11H,02H,S2H  
DFB 3EH,01H,12H,02H,S2H  
DFB 3EH,01H,14H,02H,S2H  
DFB 3EH,01H,15H,02H,S2H  
DFB 3EH,01H,1AH,02H,S2H  
DFB 3EH,01H,06H,02H,S2H  
DFB 3EH,01H,0CH,02H,S2H  
DFB 3EH,01H,0FH,02H,S3H  
DFB 3EH,01H,11H,02H,S3H  
DFB 3EH,01H,12H,02H,S3H  
DFB 3EH,01H,14H,02H,S3H  
DFB 3EH,01H,15H,02H,S3H  
DFB 3EH,01H,1AH,02H,S3H  
DFB 3EH,01H,06H,02H,S3H  
DFB 3EH,01H,0CH,02H,S3H  
DFB 3EH,01H,0FH,02H,S4H  
DFB 3EH,01H,11H,02H,S4H  
DFB 3EH,01H,12H,02H,S4H  
DFB 3EH,01H,14H,02H,S4H  
DFB 3EH,01H,15H,02H,S4H  
DFB 3EH,01H,1AH,02H,S4H  
DFB 3EH,01H,06H,02H,S4H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3EH,01H,0CH,02H,84H  
DFB 3EH,01H,0FH,02H,85H  
DFB 3EH,01H,11H,02H,85H  
DFB 3EH,01H,12H,02H,85H  
DFB 3EH,01H,14H,02H,85H  
DFB 3EH,01H,15H,02H,85H  
DFB 3EH,01H,1AH,02H,85H  
DFB 3EH,01H,06H,02H,85H  
DFB 3EH,01H,0CH,02H,85H  
DFB 3EH,01H,0FH,02H,86H  
DFB 3EH,01H,11H,02H,86H  
DFB 3EH,01H,12H,02H,86H  
DFB 3EH,01H,14H,02H,86H  
DFB 3EH,01H,15H,02H,86H  
DFB 3EH,01H,1AH,02H,86H  
DFB 3EH,01H,06H,02H,86H  
DFB 3EH,01H,0CH,02H,86H  
DFB 3EH,01H,0FH,02H,87H  
DFB 3EH,01H,11H,02H,87H  
DFB 3EH,01H,12H,02H,87H  
DFB 3EH,01H,14H,02H,87H  
DFB 3EH,01H,15H,02H,87H  
DFB 3EH,01H,1AH,02H,87H  
DFB 3EH,01H,06H,02H,87H  
DFB 3EH,01H,0CH,02H,87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DD: DFB 09H, 02H, 1SH, 10H, 02H, 00H  
DFB 19H, 02H, 1SH, 13H, 02H, 00H  
DFB 21H, 1EH, 1SH, 01H, 64H, 00H  
DFB 22H, 1EH, 02H, 1SH, 64H, 00H  
DFB 23H, 17H, 1SH, 00H, 02H, 00H  
DFB 29H, 02H, 1SH, 1SH, 02H, 00H  
DFB 2AH, 1EH, 1SH, 02H, 64H, 00H  
DFB 2BH, 0EH, 1SH, 00H, 02H, 00H  
DFB 34H, 17H, 0SH, 00H, 43H, 00H  
DFB 35H, 0EH, 0SH, 00H, 43H, 00H  
DFB 36H, 1EH, 0SH, 01H, 54H, 00H  
DFB 39H, 02H, 1SH, 22H, 02H, 00H  
DFB 46H, 1EH, 0FH, 0SH, 43H, 00H  
DFB 4EH, 1EH, 11H, 0SH, 43H, 00H  
DFB 56H, 1EH, 12H, 0SH, 43H, 00H  
DFB 5EH, 1EH, 14H, 0SH, 43H, 00H  
DFB 66H, 1EH, 15H, 0SH, 43H, 00H  
DFB 6EH, 1EH, 1AH, 0SH, 43H, 00H  
DFB 70H, 1EH, 0SH, 0FH, 43H, 00H  
DFB 71H, 1EH, 0SH, 11H, 43H, 00H  
DFB 72H, 1EH, 0SH, 12H, 43H, 00H  
DFB 73H, 1EH, 0SH, 14H, 43H, 00H  
DFB 74H, 1EH, 0SH, 15H, 43H, 00H  
DFB 75H, 1EH, 0SH, 1AH, 43H, 00H  
DFB 77H, 1EH, 0SH, 0CH, 43H, 00H  
DFB 7EH, 1EH, 0CH, 0SH, 43H, 00H  
DFB 86H, 02H, 0CH, 0SH, 43H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB SEH, 01H, 0CH, 08H, 43H, 00H  
DFB 96H, 42H, 08H, 00H, 43H, 00H  
DFB 9EH, 3CH, 0CH, 08H, 43H, 00H  
DFB 0A6H, 03H, 08H, 00H, 43H, 00H  
DFB 0AEH, 43H, 08H, 00H, 43H, 00H  
DFB 0B6H, 25H, 08H, 00H, 43H, 00H  
DFB 0BEH, 07H, 08H, 00H, 43H, 00H  
DFB 0E1H, 2BH, 18H, 00H, 02H, 00H  
DFB 0E3H, 12H, 0BH, 18H, 02H, 00H  
DFB 0E5H, 2CH, 18H, 00H, 02H, 00H  
DFB 0E9H, 1CH, 07H, 00H, 02H, 00H  
DFB 0F9H, 1EH, 22H, 18H, 02H, 00H  
ED: DFB 40H, 16H, 0FH, 04H, 02H, 00H  
DFB 41H, 28H, 04H, 0FH, 02H, 00H  
DFB 42H, 3CH, 16H, 10H, 02H, 00H  
DFB 43H, 1EH, 02H, 10H, 64H, 00H  
DFB 44H, 23H, 00H, 00H, 02H, 00H  
DFB 45H, 30H, 00H, 00H, 02H, 00H  
DFB 46H, 15H, 01H, 00H, 02H, 80H  
DFB 47H, 1EH, 17H, 0CH, 02H, 00H  
DFB 48H, 16H, 11H, 04H, 02H, 00H  
DFB 49H, 28H, 04H, 11H, 02H, 00H  
DFB 4AH, 01H, 16H, 10H, 02H, 00H  
DFB 4BH, 1EH, 10H, 02H, 64H, 00H  
DFB 4DH, 2FH, 00H, 00H, 02H, 00H  
DFB 4FH, 1EH, 21H, 0CH, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 50H, 16H, 12H, 04H, 02H, 00H  
DFB 51H, 28H, 04H, 12H, 02H, 00H  
DFB 52H, 3CH, 16H, 13H, 02H, 00H  
DFB 53H, 1EH, 02H, 13H, 64H, 00H  
DFB 56H, 15H, 01H, 00H, 02H, 31H  
DFB 57H, 1EH, 0CH, 17H, 02H, 00H  
DFB 58H, 16H, 14H, 04H, 02H, 00H  
DFB 59H, 28H, 04H, 14H, 02H, 00H  
DFB 5AH, 01H, 16H, 13H, 02H, 00H  
DFB 5BH, 1EH, 13H, 02H, 64H, 00H  
DFB 5EH, 15H, 01H, 00H, 02H, 32H  
DFB 5FH, 1EH, 0CH, 21H, 02H, 00H  
DFB 60H, 16H, 15H, 04H, 02H, 00H  
DFB 61H, 28H, 04H, 15H, 02H, 00H  
DFB 62H, 3CH, 16H, 16H, 02H, 00H  
DFB 67H, 3AH, 00H, 00H, 02H, 00H  
DFB 68H, 16H, 1AH, 04H, 02H, 00H  
DFB 69H, 28H, 04H, 1AH, 02H, 00H  
DFB 6AH, 01H, 16H, 16H, 02H, 00H  
DFB 6FH, 35H, 00H, 00H, 02H, 00H  
DFB 72H, 3CH, 16H, 22H, 02H, 00H  
DFB 73H, 1EH, 02H, 22H, 64H, 00H  
DFB 78H, 16H, 0CH, 04H, 02H, 00H  
DFB 79H, 28H, 04H, 0CH, 02H, 00H  
DFB 7AH, 01H, 16H, 22H, 02H, 00H  
DFB 7BH, 1EH, 22H, 02H, 64H, 00H  
DFB 0A0H, 21H, 00H, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 0A1H, 0AH, 00H, 00H, 02H, 00H  
DFB 0A2H, 1AH, 00H, 00H, 02H, 00H  
DFB 0A3H, 2AH, 00H, 00H, 02H, 00H  
DFB 0ASH, 1FH, 00H, 00H, 02H, 00H  
DFB 0A9H, 08H, 00H, 00H, 02H, 00H  
DFB 0AAH, 1SH, 00H, 00H, 02H, 00H  
DFB 0ABH, 29H, 00H, 00H, 02H, 00H  
DFB 0B0H, 22H, 00H, 00H, 02H, 00H  
DFB 0B1H, 0BH, 00H, 00H, 02H, 00H  
DFB 0B2H, 1BH, 00H, 00H, 02H, 00H  
DFB 0B3H, 27H, 00H, 00H, 02H, 00H  
DFB 0BSH, 20H, 00H, 00H, 02H, 00H  
DFB 0B9H, 09H, 00H, 00H, 02H, 00H  
DFB 0BAH, 19H, 00H, 00H, 02H, 00H  
DFB 0BBH, 26H, 00H, 00H, 02H, 00H  
FD: DFB 09H, 02H, 19H, 10H, 02H, 00H  
DFB 19H, 02H, 19H, 13H, 02H, 00H  
DFB 21H, 1EH, 19H, 01H, 64H, 00H  
DFB 22H, 1EH, 02H, 19H, 64H, 00H  
DFB 23H, 17H, 19H, 00H, 02H, 00H  
DFB 29H, 02H, 19H, 19H, 02H, 00H  
DFB 2AH, 1EH, 19H, 02H, 64H, 00H  
DFB 2BH, 0EH, 19H, 00H, 02H, 00H  
DFB 34H, 17H, 0AH, 00H, 43H, 00H  
DFB 35H, 0EH, 0AH, 00H, 43H, 00H  
DFB 36H, 1EH, 0AH, 01H, 54H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 39H, 02H, 19H, 22H, 02H, 00H  
DFB 46H, 1EH, 0FH, 0AH, 43H, 00H  
DFB 4EH, 1EH, 11H, 0AH, 43H, 00H  
DFB 56H, 1EH, 12H, 0AH, 43H, 00H  
DFB 5EH, 1EH, 14H, 0AH, 43H, 00H  
DFB 66H, 1EH, 15H, 0AH, 43H, 00H  
DFB 6EH, 1EH, 1AH, 0AH, 43H, 00H  
DFB 70H, 1EH, 0AH, 0FH, 43H, 00H  
DFB 71H, 1EH, 0AH, 11H, 43H, 00H  
DFB 72H, 1EH, 0AH, 12H, 43H, 00H  
DFB 73H, 1EH, 0AH, 14H, 43H, 00H  
DFB 74H, 1EH, 0AH, 15H, 43H, 00H  
DFB 75H, 1EH, 0AH, 1AH, 43H, 00H  
DFB 77H, 1EH, 0AH, 0CH, 43H, 00H  
DFB 7EH, 1EH, 0CH, 0AH, 43H, 00H  
DFB 86H, 02H, 0CH, 0AH, 43H, 00H  
DFB 8EH, 01H, 0CH, 0AH, 43H, 00H  
DFB 96H, 42H, 0AH, 00H, 43H, 0CH  
DFB 9EH, 3CH, 0CH, 0AH, 43H, 00H  
DFB 0A6H, 03H, 0AH, 00H, 43H, 00H  
DFB 0AEH, 43H, 0AH, 00H, 43H, 00H  
DFB 0B6H, 25H, 0AH, 00H, 43H, 00H  
DFB 0BEH, 07H, 0AH, 00H, 43H, 00H  
DFB 0E1H, 2BH, 19H, 00H, 02H, 00H  
DFB 0E3H, 12H, 0BH, 19H, 02H, 00H  
DFB 0E5H, 2CH, 19H, 00H, 02H, 00H  
DFB 0E9H, 1CH, 09H, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 0F9H, 1EH, 22H, 19H, 02H, 00H

DDCB: DFB 06H, 33H, 0SH, 00H, 44H, 00H

DFB 0EH, 38H, 0SH, 00H, 44H, 00H

DFB 16H, 31H, 0SH, 00H, 44H, 00H

DFB 1EH, 36H, 0SH, 00H, 44H, 00H

DFB 26H, 3FH, 0SH, 00H, 44H, 00H

DFB 2EH, 40H, 0SH, 00H, 44H, 00H

DFB 3EH, 41H, 0SH, 00H, 44H, 00H

DFB 46H, 04H, 01H, 0SH, 44H, 80H

DFB 4EH, 04H, 01H, 0SH, 44H, 81H

DFB 56H, 04H, 01H, 0SH, 44H, 82H

DFB 5EH, 04H, 01H, 0SH, 44H, 83H

DFB 66H, 04H, 01H, 0SH, 44H, 84H

DFB 6EH, 04H, 01H, 0SH, 44H, 85H

DFB 76H, 04H, 01H, 0SH, 44H, 86H

DFB 7EH, 04H, 01H, 0SH, 44H, 87H

DFB 86H, 2DH, 01H, 0SH, 44H, 80H

DFB 8EH, 2DH, 01H, 0SH, 44H, 81H

DFB 96H, 2DH, 01H, 0SH, 44H, 82H

DFB 9EH, 2DH, 01H, 0SH, 44H, 83H

DFB 0A6H, 2DH, 01H, 0SH, 44H, 84H

DFB 0AEH, 2DH, 01H, 0SH, 44H, 85H

DFB 0B6H, 2DH, 01H, 0SH, 44H, 86H

DFB 0BEH, 2DH, 01H, 0SH, 44H, 87H

DFB 0C6H, 3EH, 01H, 0SH, 44H, 80H

DFB 0CEH, 3EH, 01H, 0SH, 44H, 81H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 0D6H, 3EH, 01H, 08H, 44H, 82H  
DFB 0DEH, 3EH, 01H, 08H, 44H, 83H  
DFB 0E6H, 3EH, 01H, 08H, 44H, 84H  
DFB 0EEH, 3EH, 01H, 08H, 44H, 85H  
DFB 0F6H, 3EH, 01H, 08H, 44H, 86H  
DFB 0FEH, 3EH, 01H, 08H, 44H, 87H

FDCB: DFB 06H, 33H, 0AH, 00H, 44H, 00H  
DFB 0EH, 3SH, 0AH, 00H, 44H, 00H  
DFB 16H, 31H, 0AH, 00H, 44H, 00H  
DFB 1EH, 36H, 0AH, 00H, 44H, 00H  
DFB 26H, 3FH, 0AH, 00H, 44H, 00H  
DFB 2EH, 40H, 0AH, 00H, 44H, 00H  
DFB 3EH, 41H, 0AH, 00H, 44H, 00H  
DFB 46H, 04H, 01H, 0AH, 44H, 80H  
DFB 4EH, 04H, 01H, 0AH, 44H, 81H  
DFB 56H, 04H, 01H, 0AH, 44H, 82H  
DFB 5EH, 04H, 01H, 0AH, 44H, 83H  
DFB 66H, 04H, 01H, 0AH, 44H, 84H  
DFB 6EH, 04H, 01H, 0AH, 44H, 85H  
DFB 76H, 04H, 01H, 0AH, 44H, 86H  
DFB 7EH, 04H, 01H, 0AH, 44H, 87H  
DFB 86H, 2DH, 01H, 0AH, 44H, 80H  
DFB 8EH, 2DH, 01H, 0AH, 44H, 81H  
DFB 96H, 2DH, 01H, 0AH, 44H, 82H  
DFB 9EH, 2DH, 01H, 0AH, 44H, 83H  
DFB 0A6H, 2DH, 01H, 0AH, 44H, 84H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 0AEH, 2DH, 01H, 0AH, 44H, 85H

DFB 0B6H, 2DH, 01H, 0AH, 44H, 86H

DFB 0BEH, 2DH, 01H, 0AH, 44H, 87H

DFB 0C6H, 3EH, 01H, 0AH, 44H, 80H

DFB 0CEH, 3EH, 01H, 0AH, 44H, 81H

DFB 0D6H, 3EH, 01H, 0AH, 44H, 82H

DFB 0DEH, 3EH, 01H, 0AH, 44H, 83H

DFB 0E6H, 3EH, 01H, 0AH, 44H, 84H

DFB 0EEH, 3EH, 01H, 0AH, 44H, 85H

DFB 0F6H, 3EH, 01H, 0AH, 44H, 86H

DFB 0FEH, 3EH, 01H, 0AH, 44H, 87H

MNETAB: DFB "ADC "

DFB "ADD "

DFB "AND "

DFB "BIT "

DFB "CALL "

DFB "CCF "

DFB "CP "

DFB "CPD "

DFB "CPDR "

DFB "CPI "

DFB "CPIR "

DFB "CPL "

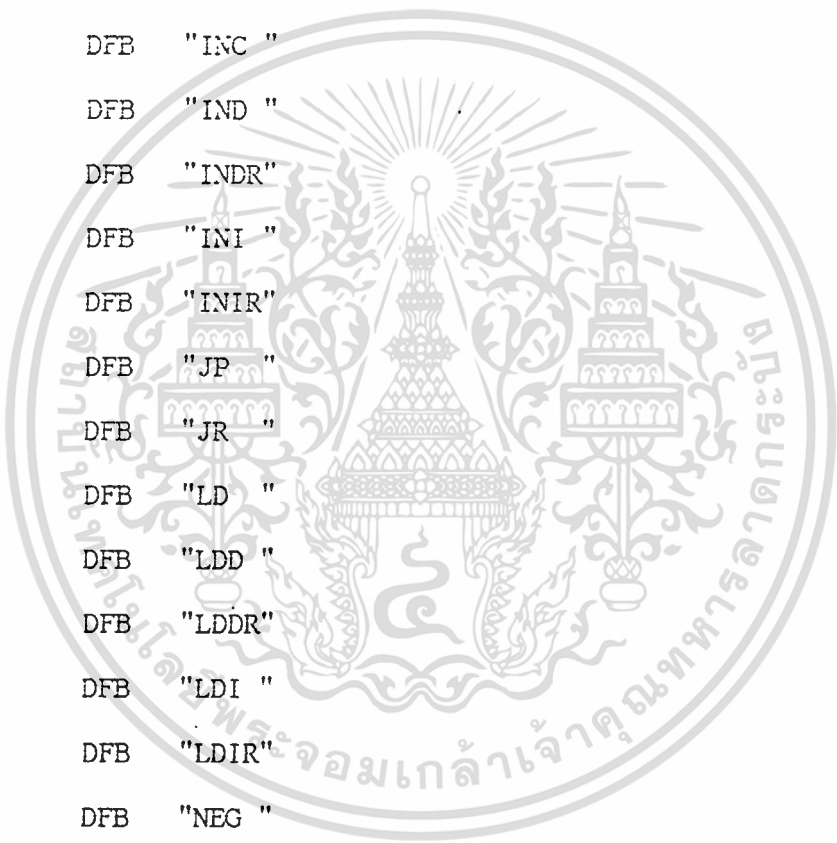
DFB "DAA "

DFB "DEC "

DFB "DI "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB "DJNZ"  
DFB "EI "  
DFB "EX "  
DFB "EXX "  
DFB "HALT"  
DFB "IM "  
DFB "IN "  
DFB "INC "  
DFB "IND "  
DFB "INDR"  
DFB "INI "  
DFB "INIR"  
DFB "JP "  
DFB "JR "  
DFB "LD "  
DFB "LDD "  
DFB "LDDR"  
DFB "LDI "  
DFB "LDIR"  
DFB "NEG "  
DFB "NOP "  
DFB "OR "  
DFB "OTDR"  
DFB "OTIR"  
DFB "OUT "  
DFB "OUTD"  
DFB "OUTI"



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB "POP "  
DFB "PUSH"  
DFB "RES "  
DFB "RET "  
DFB "RETI"  
DFB "RETN"  
DFB "RL "  
DFB "RLA "  
DFB "RLC "  
DFB "RLCA"  
DFB "RLD "  
DFB "RR "  
DFB "RRA "  
DFB "RRC "  
DFB "RRCA"  
DFB "RRD "  
DFB "RST "  
DFB "SBC "  
DFB "SCF "  
DFB "SET "  
DFB "SLA "  
DFB "SRA "  
DFB "SRL "  
DFB "SUB "  
DFB "XOR "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPRTAB: DFB "# "

DFB "(#) "

DFB "(BC)"

DFB "(C) "

DFB "(DE)"

DFB "(HL)"

DFB "(IX)"

DFB "(IX+"

DFB "(IY)"

DFB "(IY+"

DFB "(SP)"

DFB "A "

DFB "AF "

DFB "AF' "

DFB "B "

DFB "BC "

DFB "C "

DFB "D "

DFB "DE "

DFB "E "

DFB "H "

DFB "HL "

DFB "I "

DFB "IX "

DFB "IY "

DFB "L "

DFB "M "



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB "NC "  
DFB "NZ "  
DFB "P "  
DFB "PE "  
DFB "PO "  
DFB "R "  
DFB "SP "  
DFB "Z "  
END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

: FILENAME      REM180.ASM
; DESCRIPTION   ZS0180 REMOTE MONITOR V1.0
: HARDWARE     FOR ZS0180
; ASSEMBLER    CROSS-32 V3.0
: KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
; FACULTY OF INDUSTRIAL EDUCATION
; MAJOR: ELECTRONIC AND COMPUTER
: ED.ENGINEER #15

; THE FOLLOWING STATEMENTS DEFINE THE SPECIAL FUNCTION
; I/O REGISTERS USED BY THE Z180.

CNTLA0: EQU    00H      ;ASCII CONTROL REG A CH 0
CNTLA1: EQU    01H      ;ASCII CONTROL REG A CH 1
CNTLB0: EQU    02H      ;ASCII CONTROL REG B CH 0
CNTLB1: EQU    03H      ;ASCII CONTROL REG B CH 1
STAT0:  EQU    04H      ;ASCII STATUS  REG   CH 0
STAT1:  EQU    05H      ;ASCII STATUS  REG   CH 1
TDR0:   EQU    06H      ;ASCII XMIT DATA REG CH 0
TDR1:   EQU    07H      ;ASCII SMIT DATA REG CH 1
RDR0:   EQU    08H      ;ASCII RECV DATA REG CH 0
RDR1:   EQU    09H      ;ASCII RECV DATA REG CH 1
CNTR:   EQU    0AH      ;CSI/O CONTROL REGISTER
TRDR:   EQU    0BH      ;CSI/O XMIT/RECV DATA REG
TMDROL: EQU    0CH      ;TIMER  DATA REG CH 0L
TMDROH: EQU    0DH      ;TIMER  DATA REG CH 0H
RLDROL: EQU    0EH      ;RELOAD REGISTER CH 0L

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RLDR0H:	EQU	0FH	;RELOAD REGISTER CH 0H
TCR:	EQU	10H	;TIMER CONTROL REGISTER
TMDR1L:	EQU	14H	;TIMER DATA REG CH 1L
TMDR1H:	EQU	15H	;TIMER DATA REG CH 1H
RLDR1L:	EQU	16H	;RELOAD REGISTER CH 1L
RLDR1H:	EQU	17H	;RELOAD REGISTER CH 1H
SAR0L:	EQU	20H	;DMA SOURCE ADDRESS REG CH 0L
SAR0H:	EQU	21H	;DMA SOURCE ADDRESS REG CH 0H
SAR0B:	EQU	22H	;DMA SOURCE ADDRESS REG CH 0B
DAR0L:	EQU	23H	;DMA DSTNTN ADDRESS REG CH 0L
DAR0H:	EQU	24H	;DMA DSTNTN ADDRESS REG CH 0H
DAR0B:	EQU	25H	;DMA DSTNTN ADDRESS REG CH 0B
BCR0L:	EQU	26H	;DMA BYTE COUNT REGISTER CH 0L
BCR0H:	EQU	27H	;DMA BYTE COUNT REGISTER CH 0H
MAR1L:	EQU	28H	;DMA MEMORY ADDRESS REG CH 1L
MAR1H:	EQU	29H	;DMA MEMORY ADDRESS REG CH 1H
MAR1B:	EQU	2AH	;DMA MEMORY ADDRESS REG CH 1B
IAR1L:	EQU	2BH	;DMA DSTNTN I/O REG CH 1L
IAR1H:	EQU	2CH	;DMA DSTNTN I/O REG CH 1H
BCR1L:	EQU	2EH	;DMA BYTE COUNT REGISTER CH 1L
BCR1H:	EQU	2FH	;DMA BYTE COUNT REGISTER CH 1H
DSTAT:	EQU	30H	;DMA STATUS REGISTER
DMODE:	EQU	31H	;DMA MODE REGISTER
DCNTL:	EQU	32H	;DMA/ WAIT CONTROL REGISTER
IL:	EQU	33H	;IL REGISTER (INT VECTOR REG)
ITC:	EQU	34H	;INT/ TRAP CONTROL REGISTER
RCR:	EQU	36H	;REFRESH CONTROL REGISTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CBR:    EQU    38H    ;MMU COMMON BASE REGISTER
BBR:    EQU    39H    ;MMU BANK   BASE REGISTER
CBAR:   EQU    3AH    ;MMU COMMON/BANK AREA REGISTER
OMCR:   EQU    3EH    ;OPERATION MODE CONTROL REGISTER
ICR:    EQU    3FH    ;I/O CONTROL REGISTER

```

```

CPU"Z180.TBL"; CPU TABLE

```

```

HOF"INT8"; HEX OUTPUT FORMAT

```

```

; ***** VARIABLE SET *****

```

```

SPA:    EQU    50H    ;DAC PORT
SPB:    EQU    51H    ;ADC PORT
SPC:    EQU    52H    ;BUSY RD,CS (ADC908) ACTIVE LOW
SPCC:   EQU    53H    ;CONTROL PORT
UPA:    EQU    80H    ;LED FLAG
UPB:    EQU    81H    ;USER PORT
UPC:    EQU    82H    ;KEYBOARD
UPCC:   EQU    83H    ;CONTROL PORT
WRITE_INS: EQU    0C0H ;LCD PORT
WRITE_DATA: EQU    0C2H
READ_BUSY: EQU    0C4H
VELOW:   EQU    40H    ;VECTOR LOW

```

```

; ***** RAM WORKING AREA *****

```

```

SYSSTK: EQU    0FFFFH ;SYSTEM STACK (2K)
WRKRAM: EQU    0E000H ;(8K) TOP MEMORY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

USRSTK:   EQU   0F000H   ;USER STACK & USER BUFFER
          ORG   0F000H   ;SYSTEM BUFFER (2K)
RST10:    DFS   2       ;RESTART MEMORY
RST18:    DFS   2
RST20:    DFS   2
RST28:    DFS   2
RST30:    DFS   2
RST38:    DFS   2
RST66:    DFS   2
USRPC:    DFS   2       ;USER REG. MEMORY
USRSP:    DFS   2
USRIF:    DFS   2
USRIX:    DFS   4       ;IX,IY
USRAF:    DFS   8
USRAF1:   DFS   6
USRH1:    DFS   2
SEABUF:   DFS   8       ;SEARCH DATA BUF
SEAMEM:   DFS   2       ;SEARCH MEMORY
SEAFND:   DFS   2       ;SEARCH FOUND MEMORY
SEACNT:   DFS   1       ;SEARCH BYTE COUNT
ASMCOM:   DFS   8       ;ASSEMBLER COMMAND BUF
ASMBUF:   DFS   3       ;ASSEMBLER BUF
ASMOPC:   DFS   4       ;OPCODE BUF
ASMDAT:   DFS   2       ;N NN (N) (NN) DIS
ASMDATD:  DFS   1       ;+D
UASMFAG:  DFS   1       ;UNASM FLAG 0=ALL 1=NO ADDRESS,OP-CODE
UASMCNT:  DFS   1       ;UNASM LINE COUNT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MONBUF:   DFS   61       ;COMMAND BUF
MONINX:   DFS   1       ;COMMAND BUF INDEX
MONDAT:   DFS   2       ;16 BIT DATA FROM VAL SUB.
BAUD:     DFS   1       ;BAUD RATE           12H,24H,48H,96H
BRKMEM:   DFS   1       ;BREAK OLD DATA MEMORY
BRKFAG:   DFS   1       ;BREAK FLAG (B0=BREAK B1=TRACE)
BRK:      DFS   2       ;BREAK ADDRESS MEMORY
STPAD:    DFS   2       ;TRACE ADDRESS MEMORY (SINGLE STEP)
START:    DFS   2       ;START ADDRESS MEMORY
FINAL:    DFS   2       ;FINAL ADDRESS MEMORY
DEST:     DFS   2       ;DESTINATION ADDRESS MEMORY
POWMEM:   DFS   1       ;POWER UP MEMORY (A5)
           DFS   8       ;RESERVE
BUFFER:   ;SPACE

; ***** POWER-UP *****

ORG 0000H
POWER:    JP    RESET
           DFB  0FFH,0FFH,0FFH,0FFH,0FFH

```

```

; ***** RESTART *****

```

```

PUSH HL           ;RST 03H FROM BREAK,TRACE
LD HL,BACK
EX (SP),HL
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DFB OFFH,OFFH

PUSH HL ;RST 10H (USER)

LD HL,(RST10)

EX (SP),HL

RET

DFB OFFH,OFFH

PUSH HL ;RST 18H (USER)

LD HL,(RST18)

EX (SP),HL

RET

DFB OFFH,OFFH

PUSH HL ;RST 20H (USER)

LD HL,(RST20)

EX (SP),HL

RET

DFB OFFH,OFFH

PUSH HL ;RST 28H (USER)

LD HL,(RST28)

EX (SP),HL

RET

DFB OFFH,OFFH

PUSH HL ;RST 30H (USER)

LD HL,(RST30)

EX (SP),HL

RET

DFB OFFH,OFFH

PUSH HL ;RST 38H (USER)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD    HL, (RST38)
EX    (SP),HL
RET
DFB   0FFH,0FFH

```

```
; ** INTERNAL INTERRUPTS VECTOR TABLE **
```

```

INT1:   DWL   ZZZ           ;SLEEP
INT2:   DWL   ZZZ           ;SLEEP
PRT0:   DWL   ZZZ           ;SLEEP
PRT1:   DWL   ZZZ           ;SLEEP
DMA0:   DWL   ZZZ           ;SLEEP
DMA1:   DWL   ZZZ           ;SLEEP
CSIO:   DWL   ZZZ           ;SLEEP
ASCI0:  DWL   ZZZ           ;SLEEP
ASCI1:  DWL   RXBYTE1       ;RX ASCII FORM RS-232 PORT
        DFB   0FFH,0FFH,0FFH,0FFH,0FFH
        DFB   0FFH,0FFH,0FFH,0FFH,0FFH
        DFB   0FFH,0FFH,0FFH,0FFH,0FFH
        DFB   0FFH,0FFH,0FFH,0FFH,0FFH
        PUSH  HL             ;NMI (USER)
LD      HL, (RST66)
EX      (SP),HL
RET

```

```
; ***** RESET *****
```

```
RESET:  LD    HL,2000H      ;POWER UP DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RESET1:  DEC  HL
          LD   A,H
          OR   L
          JR   NZ,RESET1

; *****
; *  INITIAL PARAMETER  *
; *****

          LD   A,0           ;FOR INSTRUCTION SET Z180 ONLY
          OUTO (OMCR),A
DFLT:    LD   A,0ESH        ;MMU SETUP DEFAULT PAGE1
          OUTO (CBAR),A
          LD   A,0
          OUTO (BBR),A
          LD   A,0F0H
          OUTO (CBR),A
          LD   SP,SYSSTK    ;USER STACK
          LD   A,VECLOW     ;SET LOW VECTOR
          OUTO (IL),A
          LD   A,SAH        ;(88H) SET CONTROL    CP-JR180
          OUT  (UPCC),A
          XOR  A
          OUT  (UPA),A      ;CLEAR LED FLAG
          LD   A,SBH        ;SET CONTROL SIGNAL BOARD
          OUT  (SPCC),A
          CALL INITLCD      ;INITIAL LCD
          LD   HL,LCDTAB    ;" Remote Morniter "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL WRP                ;WRITE LCD
CALL ASCI               ;SET RS-232
LD  A, (POWMEM)        ;CHECK POWER UP
CP  0A5H
JR  Z.RESET2

; **** POWER UP ****
; CLEAR WORKING AREA

XOR  A
LD  HL, WRKRAM
LD  DE, WRKRAM+1
LD  BC, SYSSTK-WRKRAM-1
LD  (HL), A
LDIR
LD  HL, USRSTK          ;DEFAULT ADDRESS
LD  (FINAL), HL
RESET2: CALL AUTOB       ;AUTO BAUD RATE
LD  B, 24
RESET3: CALL TXLF        ;CLEAR SCEEN
DJNZ RESET3
LD  HL, MN0TAB
CALL TXBLOCK
LD  HL, MN1TAB
CALL TXBLOCK
LD  HL, MN2TAB
CALL TXBLOCK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,MN3TAB
CALL TXBLOCK
LD HL,MN4TAB
CALL TXBLOCK
LD HL,BAUTAB
CALL TXBLOCK
LD A,(BAUD)
LD D,A
CALL TXBYTE2
LD D.0
CALL TXBYTE2
LD HL,MN5TAB
CALL TXBLOCK
LD HL,MN6TAB
CALL TXBLOCK
LD HL,MN7TAB
CALL TXBLOCK
CALL TXLF
CALL TXLF
LD HL,USRSTK ;DEFAULT VAR
LD (USRPC),HL
LD HL,USRSTK
LD (USRSP),HL
LD HL,BRKFAG
BIT 0,(HL) ;CHECK.BREAK FLAG
JR Z,RESET4
RES 0,(HL) ;ON BREAK PROCESS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,(BRK)
LD A,(BRKMEM)
LD (HL),A
JR RESETS
RESET4: BIT 1,(HL) ;CHECK TRACE FLAG
JR Z,RESETS
RES 0,(HL) ;ON TRACE PROCESS
LD HL,(STPADD)
LD A,(BRKMEM)
LD (HL),A
RESETS: LD A,(POWMEM) ;POWER UP DISPLAY
CP 0A5H
JR Z,RESET9
LD HL,POWTAB ;IS POWER UP
CALL TXBLOCK
LD A,0A5H
LD (POWMEM),A
RESET9:
; ***** MONITOR MAIN *****

MON: LD D,07H ;BELL
CALL TXBYTE
LD D,5AH ;Z
CALL TXBYTE
LD D,31H ;1
CALL TXBYTE
LD D,38H ;8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXBYTE

LD D,30H ;0

CALL TXBYTE

LD D,3EH ;>

CALL TXBYTE

CALL TMGETL

XOR A

LD (MONINX),A ;CLEAR INDEX

LD A,(MONBUF)

CP ODH

JR Z,MON

LD A,(MONBUF+1)

CP " "

JR Z,MON1

CP ODH

JP NZ,ERROR

MON1: LD A,(MONBUF) ;COMMAND

CALL UPPER

MON2: LD HL,MNSTAB

LD IX,MN9TAB

LD B,22

MON3: CP (HL)

JR Z,MON4

INC HL

INC IX

INC IX

DJNZ MON3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JR    ERROR            ;NO COMMAND

MON4:          CALL NEXT

                LD    H,(IX+1)        ;JUMP

                LD    L,(IX+0)

                JP    (HL)

; **** SET ASCII CH1 TX,RX,S,N,1.9600 (X'TAL 12.488 Mhz) ****

ASCII:        LD    A,64H

                OUT0 (CNTLA1),A      ;TX,RX,SBIT,1 STOP BIT

                LD    A,2

                OUT0 (CNTLB1),A

                LD    A,8            ;RIE ENABLE TO KEYBOARD

                OUT0 (STAT1),A

                RET

; ***** LCD CONTROL SUB *****

; WRITE TEXT 1 LINE (16 CHARACTER) ON LCD

; IN = HL --> TABLE TEXT ( 16 BYTE ONLY! )

; REG = ABCDHL

WRP:          LD    A,0

                CALL GOTO

                CALL WRLINE

                LD    A,40H

                CALL GOTO

                CALL WRLINE

                RET

GOTO:         SET    7,A

                OUT   (WRITE_INS),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    READ
RET
READ:   IN    A, (READ_BUSY)
BIT     7,A
JR      NZ,READ
RET
WRLINE: LD    B,S
TEST11: LD    A,(HL)
LD      D,A
CALL    WRBYTE
INC     HL
DJNZ   TEST11
RET
; ***** INITIAL LCD    DL=1 8 BIT,N=1 1/16 DUTY,F=0 5*7 *****
INITLCD: LD    A,00111000B ;FUNCTION SET 38H
;DL=1 8 BIT
;N=1 1/16 DUTY    F=0 5*7
OUT     (WRITE_INS),A
CALL    WAIT1
CALL    WAIT1 ;DELAY > 4.1 ms
LD      A,00001111B ;DISPLAY ON/OFF CONTROL
;D=1 OFF , C=1 CURSOR ON
;B=1 BLINK
OUT     (WRITE_INS),A
CALL    WAIT1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,00000110B    ;ENTRY MODE SET
                                ;I/D=1 INCREMENT,S=0 RIGHT

OUT     (WRITE_INS),A

CALL    WAIT1

RET

:***** CLEAR SCREEN LCD *****

CLSLCD: LD      A,00000001B    ;CLEAR ALL DISPLAY

OUT     (WRITE_INS),A

CALL    READ

RET

; ***** WRITE BYTE SUB *****
; IN = D
WRBYTE: LD      A,D

OUT     (WRITE_DATA),A

CALL    READ

RET

; ***** READY SUB *****

WAIT1:  PUSH    BC

        PUSH    .AF

        LD      B,80H

WAIT12: DJNZ    WAIT12

        POP     AF

        POP     BC

        RET

; *** END LCD CONTROL SUB ***

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ***** SYNTAX ERROR MAIN *****
```

```
ERROR: LD SP, SYSSTK ; RESET STACK
```

```
ERROR1: LD D, " "
```

```
CALL TXBYTE
```

```
LD A, (MONINX)
```

```
CP 0
```

```
JR Z, ERROR2
```

```
DEC A
```

```
LD (MONINX), A
```

```
JR ERROR1
```

```
ERROR2: LD HL, SYNTAB ; SYNTAX ERROR
```

```
CALL TXBLOCK
```

```
JP MON
```

```
; IN = HL ERROR POSITION ON MONBUF
```

```
ERRORX: LD DE, MONBUF
```

```
XOR A
```

```
SBC HL, DE
```

```
LD A, L
```

```
LD (MONINX), A
```

```
JP ERROR
```

```
; ***** ASM ROUTINE *****
```

```
; Z80 ASSEMBLER
```

```
; A ADDR
```

```
ASM: CP 0
```

```
JR Z, ERROR
```

```
CALL VAL ; ADDR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL NEXT

CP 0

JR NZ,ERROR

ASM1: LD DE,(MONDAT) ;DISPLAY ADDR

CALL TXBYTE4

LD D,":"

CALL TXBYTE

CALL TMGETL

XOR A

LD (MONINX),A

LD A,(MONBUF) ;FIRST DIGIT

CP ODH

JP Z,MON ;END ASM

LD HL,0

LD (ASMBUF+1),HL ;CLEAR OPRAND INDEX

LD (ASMDAT),HL

LD HL,MONBUF ;TO UPPER

ASM2: LD A,(HL)

CP ODH

JR Z,ASM3

CALL UPPER

LD (HL),A

INC HL

JR ASM2

ASM3: LD B,4 ;MNEMONIC SEARCH

LD HL,MONBUF

CALL ASMMOV

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IX,MNETAB

LD IY,ASMCOM

LD BC,4 ;LENGTH

LD HL,67

CALL SEAR

CP 0

JP Z,ASMME ;MNE NOT FOUND

LD (ASMBUF),A ;MNE INDEX

ASM4: CALL NEXT

CP 0

JR Z,ASM7 ;NO OPERAND

LD HL,MONBUF ;SEARCH ", "

LD D,0

LD E,A

ADD HL,DE

ASM41: LD A,(HL)

CP ", "

JR Z,ASM6

CP " "

JR Z,ASM5

CP ODH

JR Z,ASM5

INC HL

JR ASM41

ASM5: CALL ASMOPR ;1 OPERAND

CP OFFH

JR Z,ASMOE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (ASMBUF+i),A ;OPERAND INDEX
JR ASM7
ASM6: LD A," " ;2 OPERAND
LD (HL),A ;CHANGE ", " TO BLANK
CALL ASMOPR
CP OFFH
JR Z,ASMOE
LD (ASMBUF+1),A ;OPERAND 1 INDEX
CALL NEXT
CP 0
JR Z,ASMOE
CALL ASMOPR
CP OFFH
JR Z,ASMOE
LD (ASMBUF+2),A ;OPERAND 2 INDEX
ASM7: CALL ASMS ;SEARCH TABLE
CP 0
JR Z,ASMCE ;NO COMMAND
LD B,A ;OP-CODE LENGHT
PUSH BC
CALL ASMREL ;CHECK JR MNEMONICS
POP BC
CP OFFH
JP Z,ASM1 ;RELATIVE ERROR
LD HL,(MONDAT)
LD DE,ASMOPC
ASM71: LD A,(DE)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (HL),A ;LOAD OPCODE TO MEM
INC HL
INC DE
DJNZ ASM71
LD (MONDAT),HL
JP ASM1 ;BACK TO NEXT LINE
ASMME: LD HL,ASMER1 ;MNEMONIC ERROR
CALL TXBLOCK
JP ASM1
ASMME: LD HL,ASMER2 ;OPERAND ERROR
CALL TXBLOCK
JP ASM1
ASMCE: LD HL,ASMER3 ;COMMAND ERROR
CALL TXBLOCK
JP ASM1
; **** CHECK JR MNEMONIC SUB. ****
; FOR RELOCATE ADDRESS
; IN = (ASMDAT) DIS
; = (ASMBUF) TABLE INDEX
; = (MONDAT) OP-CODE ADDRESS
; OUT = (ASMOPC) OP-CODE
ASMREL: LD A,(ASMBUF) ;MNEMONICS INDEX
CP 29 ;JR
JR Z,ASMREL1
CP 16 ;DJNZ
RET NZ

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASMREL1: LD HL,(MONDAT) ;IS JR MNEMONICS
          LD DE,(ASMDAT)
          CALL ASMRELS ;CALCULATE AND LOAD TO REM
          RET

```

```

; JUMP RELATIVE CALCULATE SUB.

```

```

; IN = HL START

```

```

; = DE FINAL

```

```

; OUT = (HL+1)

```

```

; = A FF=ERROR

```

```

ASMRELS: EX DE,HL
          INC DE
          INC DE
          XOR A
          SBC HL,DE ;FINAL-START+2
          DEC DE
          CP H
          JR Z,ASMRELS2
          DEC A ;FF
          CP H
          JR Z,ASMRELS3

```

```

ASMRELS1: LD A,OFFH ;JUMP ERROR
          LD (ASMOPC+1),A
          LD HL,JUMPTAB
          CALL TXBLOCK
          LD A,OFFH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
ASMRELS2: LD  A,7FH          ;JUMP TO HIGHER
          CP  L
          JR  C,ASMRELS1
          JR  ASMRELS4
ASMRELS3: CP  L              :A=FF
          JR  Z,ASMRELS1
          LD  A,7FH
          CP  L
          JR  NC,ASMRELS1
ASMRELS4: LD  A,L
          LD  (ASMOPC+1),A
          XOR A
          RET
; **** SEARCH OPERAND SUB. ****
; IN  = (MONINX)
; OUT = A OPERAND INDEX FF=ERROR
;      (ASMDATD) +D
;      (ASMDAT) N NN (N) (NN) DIS
ASMOPR:  LD  HL,MONBUF
          LD  A,(MONINX)
          LD  D,0
          LD  E,A
          ADD HL,DE
          LD  B,8          ;MAX CHAR. OF OPRERAND
          CALL ASMMOV

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,ASMCOM
LD A,(HL)
CP "("
JR Z,ASMOPR1
CP "0"
JR C,ASMOPR2 ;NOT VAL
CP "9"+1
JR NC,ASMOPR2
LD C,1 ;N NN DIS
JR ASMOPRV
ASMOPR1: INC HL ;CHECK "("
LD A,(HL)
CP "0"
JR C,ASMOPR12
CP "9"+1
JR NC,ASMOPR12
LD C,2 ;(N) (NN)
JR ASMOPRV
ASMOPR12: CP "I" ;CHECK IX+D IY+D
JR NZ,ASMOPR2 ;NOT IX IY
INC HL
LD D,(HL) ;X,Y
INC HL
LD A,(HL)
CP "+"
JR NZ,ASMOPR2 ;NOT +D
INC HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD   A,D
LD   C,S           ;(IX+D) INDEX
CP   "X"
JR   Z,ASMOPR13
LD   C,10          ;(IY+D) INDEX
CP   "Y"
JR   Z,ASMOPR13
JR   ASMOPRE      ;NC + ERROR
ASMOPR13: EX  DE,HL
LD   HL,ASMDATD
LD   (HL),0
LD   A,(DE)       ;CHECK NUMBER
CP   "0"
JR   C,ASMOPRE   ;NOT VAL
CP   "9"+1
JR   NC,ASMOPRE  ;NOT VAL
LD   B,4          ;CHAR. LIMIT
ASMOPR14: LD   A,(DE) ;+D
CP   ")"
JR   Z,ASMOPR15
CALL CHEX
CP   OFFH
JR   Z,ASMOPRE
RLD
INC  DE
DJNZ ASMOPR14
JR   ASMOPRE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ASMOPR15: LD  A,C                ;IS (IX+D) (IY+D)
          RET

ASMOPR2:  LD  IX,OPRTAB          ;OPERAND SEARCH
          LD  IY,ASMCOM
          LD  BC,4                ;TABLE LENGTH
          LD  HL,35
          CALL SEAR
          CP  0
          RET  NZ                ;FOUND
ASMOPRE:  LD  A,OFFH            ;OPERAND ERROR
          RET
; VALUE N NN (N) (NN) DIS
ASMOPRV:  EX  DE,HL             ;VAL N NN (N) (NN) DIS
          LD  B,7
ASMOPRV1: LD  A,(DE)
          CP  " "                ;END FIELD
          JR  Z,ASMOPRV9
          CP  ")"                ;END FIELD
          JR  Z,ASMOPRV9
          CP  "H"
          JR  Z,ASMOPRV2
          CALL CHEX
          CP  OFFH
          JR  Z,ASMOPRE
          LD  HL,ASMDAT
          RLD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        INC HL
        RLD
ASMOPRV2: INC DE
        DJNZ ASMOPRV1
        JR ASMOPRE
ASMOPRV9: LD A,C ;IS VAL
        RET

```

```

; **** MOVE TO ASMCOM SUB. ****
; IN = (HL) START FIELD
; B COUNT (MAX=8)
; END BY BLANK,CR OR B=0
; OUT = (ASMCOM)
ASMMOV: LD DE,2020H ;CLEAR BY BLANK
        LD (ASMCOM),DE
        LD (ASMCOM+2),DE
        LD (ASMCOM+4),DE
        LD (ASMCOM+6),DE
        LD DE,ASMCOM
ASMMOV1: LD A,(HL)
        CP " "
        RET Z
        CP ODH
        RET Z
        LD (DE),A
        INC HL
        INC DE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DJNZ ASMMOV1

RET

; \*\*\*\*\* BREAK ROUTINE \*\*\*\*\*

: SET/CLEAR BREAK

; B [ADDR]

BREAK: CP 0  
JR Z,BREAKD  
CALL VAL  
CALL NEXT  
CP 0  
JP NZ,ERROR  
LD HL,(MONDAT)  
LD (BRK),HL  
JP MON  
BREAKD: LD DE,(BRK)  
CALL TXBYTE4  
CALL TXLF  
JP MON

; \*\*\*\*\* COMPARE ROUTINE \*\*\*\*\*

; COMPARE MEMORY

; C RANGE ADDR

; RANGE H BYTE

COMPARE: CALL RANGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL NEXT
CP 0
JP Z,ERROR
LD HL,MONBUF
LD B,0
LD A,(MONINX)
LD C,A
ADD HL,BC
LD A,(HL)
CALL UPPER
CP "H"
JR Z,COMPHEX
CALL VAL
CALL NEXT
CP 0
JP NZ,ERROR
CALL CLEN ;COMPARE BLOCK
LD IX,(MONDAT)
COMPARE1: LD A,(HL)
CP (IX+0)
JR Z,COMPARE2
PUSH HL ;NOT EQUAL
POP DE
CALL TXBYTE4 ;ADDR 1
LD D,":"
CALL TXBYTE
LD A,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD D,A
CALL TXBYTE2 ;DATA 1
LD D," "
CALL TXBYTE
PUSH IX
POP DE
CALL TXBYTE4 ;ADDR 2
LD D,":"
CALL TXBYTE
LD D,(IX+0)
CALL TXBYTE2 ;DATA 2
CALL TXLF
COMPARE2: INC HL
INC IX
DEC BC
LD A,B
OR C
JR NZ,COMPARE1
JP MON
COMPHEX: INC HL
LD A,(HL)
CP " "
JP NZ,ERRORX
CALL NEXT
CP 0
JP Z,ERROR
CALL VAL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL NEXT
CP 0
JP NZ.ERROR
CALL CLEN
COMPHEX1: LD A,(MONDAT)
CP (HL)
JR Z,COMPHEX2
PUSH HL
POP DE
CALL TXBYTE4
LD D,":"
CALL TXBYTE
LD D,(HL)
CALL TXBYTE2
CALL TXLF
COMPHEX2: INC HL
DEC BC
LD A,B
OR C
JR NZ,COMPHEX1
JP MON

; ***** DISPY ROUTINE *****
; DISPLAY MEMORY
; D [ADDR1] [ADDR2]

DISPY: LD HL,127

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (DEST),HL
CALL AD2
LD IX,(START)
LD C,0
DISPY4: LD B,16 ;START LOOP
PUSH IX
POP DE
PUSH IX
POP IX
CALL TXBYTE4 ;DISPLAY ADDR
LD D,":"
CALL TXBYTE
DISPY5: LD D," "
CALL TXBYTE
LD D,(IX+0)
CALL TXBYTE2 ;DISPLAY DATA
PUSH BC ;CHECK END
LD BC,(FINAL)
CALL COMP16
POP BC
JR NZ,DISPY51
LD C,1 ;END FLAG
DISPY51: INC IX
LD A,9
CP B
JR NZ,DISPY6
LD D," "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXBYTE
LD D,"-"
CALL TXBYTE
DISPY6: DJNZ DISPY5
LD D," "
CALL TXBYTE
LD ,D," "
CALL TXBYTE
LD B,16
DISPY7: LD A,(IY+0)
AND 7FH
CP ""
JR NC,DISPYS
LD D,"."
CALL TXBYTE
JR DISPY9
DISPY8: LD D,A
CALL TXBYTE ;DISPLAY CHAR.
DISPY9: INC IY
DJNZ DISPY7
CALL TXLF
LD A,C
CP 0
JR Z,DISPY4
LD (FINAL),IX
JP MON

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** ENTER ROUTINE *****
; ENTER DATA
; E ADDR
ENTER:   CP   0
        JP   Z,ERROR
        CALL VAL
        CALL NEXT
        CP   0
        JP   NZ,ERROR
        LD   IX,(MONDAT)
ENTER1:  PUSH IX
        POP  DE
        CALL TXBYTE4
        LD   D,":"
        CALL TXBYTE
        LD   D,(IX+0)
        CALL TXBYTE2
        LD   D," "
        CALL TXBYTE
        CALL TMGETL
        LD   A,(MONBUF)      ;FIRST DIGIT
        CP   0DH
        JP   Z,MON
        LD   HL,MONDAT
        LD   (HL),0         ;CLEAR DATA
        CALL CHEX
        CP   0FFH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    Z,ENTERE

RLD

LD    A,(MONBUF+1)    ;SECOND DIGIT

CP    0DH

JR    Z,ENTER2

CALL  CHEX

CP    0FFH

JR    Z,ENTERE

RLD

LD    A,(MONBUF+2)

CP    0DH

JR    NZ,ENTERE

ENTER2: LD    A,(MONDAT)    :ENTER

LD    (IX+0),A

INC   IX

JR    ENTER1

ENTERE: LD    D,07H    ;ERROR

CALL  TXBYTE

JR    ENTER1

```

```

; ***** FILL ROUTINE *****

```

```

; FILL MEMORY

```

```

; F RANGE BYTE

```

```

FILL:  CALL RANGE

```

```

CALL NEXT

```

```

CP    0

```

```

JP    Z,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL VAL
CALL NEXT
CP 0
JP NZ,ERROR
CALL CLEN
LD A,(MONDAT)
LD D,A
FILL1: LD (HL),D
INC HL
DEC BC
LD A,B
OR C
JR NZ,FILL1
JP MON
; ***** GO ROUTINE *****
; RUN USER PROGRAM
; G [ADDR]
GO: CP 0
JR Z,GO1
CALL VAL
CALL NEXT
CP 0
JP NZ,ERROR
LD HL,(MONDAT)
LD (USRPC),HL
GO1: LD IX,(USRPC)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD BC,(BRK)

CALL COMP16

JR NC.EXIT

LD HL,BRKFA6

SET 0,(HL)

LD HL,(BRK)

LD A,(HL)

LD (BRKMEM),A

LD (HL),OCFH ;RST OSH OP-CODE

EXIT: LD HL,(USRIF)

LD A,H

LD I,A ;LOAD I

DI

BIT 2,L

JR Z,EXIT1

EI ;LOAD IFF

EXIT1: LD SP,USRIX

POP IX ;LOAD IX

POP IY ;LOAD IY

POP AF ;LOAD AF

POP BC ;LOAD BC

POP DE ;LOAD DE

POP HL ;LOAD HL

EX AF,AF'

EXX

POP AF ;LOAD AF'

POP BC ;LOAD BC'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP DE ;LOAD DE'
POP HL ;LOAD HL'
EX AF,AF'
EXX
LD SP,(^SRSP) ;LOAD SP
PUSH HL
LD HL,(^SRPC)
EX (SP),HL
RET ;LOAD PC
; ***** HELP ROUTINE *****
; USER HELP
; H
HELP: CP 0
JP NZ,ERROR
LD E,24 ;TAB
LD HL,HELPTB
HELP1: LD A,(HL)
CP 0
JP Z,MON ;EXIT
INC HL
CP 0DH
JR Z,HELPS
CP "\" ;CHECK TAB
JR Z,HELPS
LD D,A
CALL TXBYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC E

JR HELP1

HELP5: CALL TXLF          ;LF

LD E,24

JR HELP1

HELP8: DEC E              ;PRINT TAB

JR Z,HELP1

LD D," "

CALL TXBYTE

JR HELP8

; ***** INPUT ROUTINE *****
; IN PORT
; I PORT

INPUT: CP 0

JP Z,ERROR

CALL VAL

CALL NEXT

CP 0

JP NZ,ERROR

LD A,(MONDAT)

LD C,A

IN D,(C)

CALL TXBYTE2

CALL TXLF

JP MON

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** LOAD ROUTINE *****
; LOAD FROM TERMINAL (.HEX FILE)
; L [ADDR]
; ADDR IS OFFSET ADDRESS
LOAD:  LD  HL,0
      LD  (MONDAT),HL ;DEFAULT OFFSET
      CP  0
      JR  Z,LOAD1
      CALL VAL
      CALL NEXT
      CP  0
      JP  NZ,ERROR
LOAD1: LD  HL,LOADTAB2
      CALL TXBLOCK
      LD  HL,LOADTAB3
      CALL TXBLOCK
      CALL RXBYTE ;START READ
      CP  ":"
      JP  NZ,LOADE
      LD  C,0 ;START CHECKSUM
      CALL LOADS ;START LINE
      LD  B,D ;COUNTER
      CALL LOADS ;ADDR
      LD  H,D
      CALL LOADS
      LD  L,D
      LD  DE,(MONDAT)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD HL,DE ;OFFSET ADDR
LD (FINAL),HL
CALL LOADS ;00
LOAD2: CALL LOADS ;DATA
LD (HL),D
INC HL
DJNZ LOAD2
LD A,C ;CHECKSUM
NEG
LD C,A
CALL RXBYTE2
CP C
JR NZ,LOADE
CALL RXBYTE ;CR
CALL RXBYTE ;LF
CALL RXBYTE ;NEW LINE
CP ":"
JR NZ,LOADE
LD C,0
CALL LOADS
CP 0
JR Z,LOAD3 ;LAST LINE
LD B,A
CALL LOADS
LD H,D
CALL LOADS
LD L,D

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD DE,(MONDAT)

ADD HL,DE ;OFFSET ADDR

CALL LOADS ;00

JR LOAD2

LOAD3: LD B,4 ;END DATA

LOAD31: CALL RXBYTE2 ;00

DJNZ LOAD31

CALL RXBYTE ;CR

CALL RXBYTE ;LF

LD DE,(FINAL)

CALL TXBYTE4

CALL TXLF

JP MON

LOADE: LD HL,LOADTAB ;LOAD ERROR

CALL TXBLOCK

LOADE1: CALL RXBYTE

CP ODH

JR NZ,LOADE1

JP MON

LOADS: CALL RXBYTE2 ;2 BYTE LOAD SUB.

LD A,C

ADD A,D ;CHECKSUM

LD C,A

RET

```

```

; ***** MOVE ROUTINE *****

```

```

; MOVE MEMORY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; M RANGE ADDR
MOVE:   CALL RANGE
        CALL NEXT
        CP    0
        JP    Z,ERROR
        CALL VAL
        CALL NEXT
        CP    0
        JP    NZ,ERROR
        LD    BC,(MONDAT)
        LD    HL,(START)
        LD    DE,(FINAL)
        CALL MOVES
        JP    MON
; ***** OUTPUT ROUTINE *****
; OUT PORT
; O PORT BYTE
OUTPUT: CP    0
        JP    Z,ERROR
        CALL VAL
        LD    A,(MONDAT)
        LD    C,A
        PUSH BC
        CALL NEXT
        CP    0
        JP    Z,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL VAL
CALL NEXT
CP 0
JP NZ.ERROR
POP BC
LD A,(MONDAT)
OUT (C),A
JP MON
```

```
: ***** UPLOAD ROUTINE *****
```

```
; UPLOAD TO TERMINAL (.HEX FILE)
```

```
; P RANGE
```

```
UPLOAD: CALL RANGE
CALL NEXT
CP 0
JP NZ.ERROR
LD HL,UPLTB1
CALL TXBLOCK
LD HL,UPLTB2
CALL TXBLOCK
```

```
LD HL,UPLTB3
```

```
CALL TXBLOCK
```

```
UPLOAD0: CALL RXBYTE
```

```
CP 0DH
```

```
JR NZ,UPLOAD0
```

```
LD HL,(FINAL)
```

```
LD DE,(START)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XOR A
SBC HL,DE
INC HL
PUSH HL
POP IY          :LENGTH
LD IX,(START)

UPLOAD1: PUSH IY
          POP HL
          LD DE,32+1
          XOR A
          SBC HL,DE
          JR C,UPLOAD3 ;<=32
          LD BC,2000H ;32 LOOP
          CALL UPLOADS
          JR UPLOAD1

UPLOAD3:  PUSH IY
          POP BC
          LD B,C
          LD C,0
          CALL UPLOADS ;LAST LINE
          LD HL,UPLHEX
          CALL TXBLOCK
          CALL RXBYTE ;WAIT FOR ANY KEYS
          JP MON

UPLOADS: LD D,";" ;ONE LINE SUB.
          CALL TXBYTE ;START LINE
          LD D,B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL UPL0ADX
CALL TXBYTE2          ;LINE LENGTH
PUSH IX
POP DE
CALL UPL0ADX
LD D, E
CALL UPL0ADX
PUSH IX
POP DE
CALL TXBYTE4          ;ADDR
LD D, 00H
CALL TXBYTE2          ;00
UPLOADS1: LD D, (IX+0)
CALL UPL0ADX
CALL TXBYTE2
INC IX
DEC IY
DJNZ UPLOADS1
LD A, C                ;CHECKSUM
NEG
LD D, A
CALL TXBYTE2
CALL TXLF
RET
UPLOADX: LD A, C        ;CHECKSUM
ADD A, D
LD C, A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

: \*\*\*\*\* BACK ROUTINE \*\*\*\*\*

: BACK BY BREAK ADDRESS OR INT KEY

```
BACK: LD (USRIX), IX ;SAVE IX
      POP IX
      LD (USRPC), IX ;SAVE PC
      LD (USRSP), SP ;SAVE SP
      LD SP, USRH1+2
      EXX
      PUSH HL ;SAVE HL'
      PUSH DE ;SAVE DE'
      PUSH BC ;SAVE BC'
      EXX
      EX AF, AF'
      PUSH AF ;SAVE AF'
      EX AF, AF'
      PUSH HL ;SAVE HL
      PUSH DE ;SAVE DE
      PUSH BC ;SAVE BC
      PUSH AF ;SAVE AF
      PUSH IY ;SAVE IY
      LD A, I
      LD (USRIF+1), A ;SAVE I
      LD A, 0
      JP PO, BACK1
      LD A, 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BACK1:  LD  (USRIF),A      ;SAVE IFF
        LD  SP, SYSSTK
        LD  HL, BRKFAG
        BIT 1, (HL)
        JR  Z, BACK4
        RES 1, (HL)      ;IS TRACE
        LD  HL, (USRPC)
        DEC HL
        LD  (USRPC), HL
        LD  HL, (STPADD)
        LD  A, (BRKMEM)
        LD  (HL), A
        JR  BACK6
BACK4:  BIT 0, (HL)
        JR  Z, BACK5
        RES 0, (HL)     ;IS BREAK
        LD  HL, (USRPC)
        DEC HL
        LD  (USRPC), HL
        LD  HL, (BRK)
        LD  A, (BRKMEM)
        LD  (HL), A
        JR  BACK6
BACK5:
BACK6:  LD  DE, (USRPC)
        CALL TXBYTE4
        CALL TXLF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                JP    MON
; ***** REG ROUTINE *****
; REGISTOR DISPLAY & SET
; R [REG VALUE]
REG:           CP    0
                JP    Z,REGD           ;DISPLAY ONLY
                LD    HL,MONBUF
                LD    B,0
                LD    C,A
                ADD   HL,BC
                PUSH  HL
                POP   IX               ;INDEX TO REG. NAME
                CALL  NEXT
                CP    0
                JP    Z,ERROR
                PUSH  AF
                CALL  VAL               ;VALUE
                CALL  NEXT
                CP    0
                JP    NZ,ERROR         ;FIELD OVER
                POP   AF
                LD    (MONINX),A
                LD    A,(IX+0)         ;TO UPPER AND CHECK NO OF CHAR.
                CALL  UPPER
                LD    (IX+0),A
                LD    A,(IX+1)
                CP    " "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ทำกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JR NZ,REG1

CALL REGX ;1 CHAR. SEARCH 8 BIT

CP OFFH

JR Z,REGERR

JP MON

REG1: CALL UPPER

LD (IX+1),A

LD A,(IX+2)

CP " "

JR NZ,REG2

CALL REGX ;2 CHAR. SEARCH 8 , 16 BIT

CP 0

JP Z,MON

CALL REGXX

CP 0

JP Z,MON

CALL REGFAG

CP OFFH

JR Z,REGERR

JP MON

REG2: CALL UPPER

LD (IX+2),A

LD A,(IX+3)

CP " "

JR NZ,REGERR ;>3 CHAR.

CALL REGXX ;3 CHAR. SEARCH 16 BIT

CP 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JP    Z,MON
REGERR:  PUSH IX                ;REG. NAME ERROR
        POP  HL
        JP   ERRORX

; IN = (IX) REG. NAME
; OUT = A NOT FOUND FLAG

REGXX:   LD    IY,REGTB1        ;SEARCH 16 BIT REG. NAME
        LD    C,0
REGXX1:  PUSH IX
        POP  HL
        PUSH IY
        POP  DE
        LD   B,3
REGXX2:  LD   A,(DE)
        CP   (HL)
        JR   NZ,REGXX3
        INC  HL
        INC  DE
        DJNZ REGXX2
        LD   HL,USRPC           ;SEARCH FOUND
        ADD  HL,BC              ;B=0
        LD   DE,(MONDAT)
        LD   (HL),E
        INC  HL
        LD   (HL),D
        XOR  A
        RET                    ;FOUND EXIT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

REGXX3: LD DE,3
        ADD IY,DE
        INC C
        INC C
        LD A,C
        CP 1AH
        JR C,REGXX1
        LD A,0FFH
        RET ;NOT FOUND EXIT
; IN = (IX) REG. NAME
; OUT = A NOT FOUND FLAG
REGX: LD IY,REGTB3 ;SEARCH 8 BIT REG. NAME
      LD BC,0
      LD D,(IX+0)
      LD E,(IX+1)
REGX1: CALL REGCOMP
      JR NZ,REGX2
      LD HL,USRIF+1 ;SEARCH FOUND
      ADD HL,BC ;B=0
      LD A,(MONDAT)
      LD (HL),A
      XOR A
      RET ;FOUND EXIT
REGX2: INC IY
      INC IY
      INC C
      LD A,C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP    12H

JR    C.REGX1

LD    A,0FFH

RET                                ;NOT FOUND EXIT

; IN = (IX) FLAG NAME

; OUT = A NOT FOUND FLAG

REGFAG: LD    D,(IX+0)

LD    E,(IX+1)

LD    IY,REGTB4

LD    BC,087FH

REGFAG1: CALL REGCOMP

JR    NZ,REGFAG2

CALL REGFAGD                      ;SEARCH FOUND

LD    A,(USRAF)

AND   C                          ;MARK

OR    D                          ;SET

LD    (USRAF),A

XOR   A

RET                                ;FOUND EXIT

REGFAG2: INC  IY

INC   IY

RRC   C

DJNZ  REGFAG1

LD    A,0FFH

RET                                ;NOT FOUND EXIT

REGFAGD: LD    D,0                ;CHECK FLAG DATA

LD    A,(MONDAT)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP 0

RET Z

LD A,C

CPL

LD D,A          ;DATA OUT SET BIT BY C

LD A,(MONDAT)

CP 1

RET Z

JP ERROR

; COMPARE DE & (IY)
REGCOMP: LD A,D
CP (IY+0)
RET NZ
LD A,E
CP (IY+1)
RET

; ***** REGD *****

REGD: LD IX,REGTB1

LD IY,USRPC

LD C,5          ;COUNT

CALL REGDR     ;16BIT REG.

LD C,4

CALL REGDR     ;MAIN REG.

LD C,4

CALL REGDR     ;ALTERNATE REG.

LD IX,REGTB2   ;FLAG TABLE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,(USRAF) ;FLAG
LD C,A
LD B,2
CALL REGDFX ;S Z
SLA C
LD B,1
CALL REGDFX ;H
SLA C
LD B.3
CALL REGDFX ;P/V N C
CALL TXLF
JP MON
REGDR: LD B.3 ;DISPLAY REG. ONE LINE
REGDR1: LD D,(IX+0)
CALL TXBYTE ;REG. NAME
INC IX
DJNZ REGDR1
LD D,"="
CALL TXBYTE
LD D,(IY+1)
LD E,(IY+0)
CALL TXBYTE4 ;REG. DATA
LD D," "
CALL TXBYTE
INC IY
INC IY
DEC C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    NZ,REGDR

CALL TXLF

RET

; IN = IX TABLE

REGDF: LD    D,(IX+0)      ;DISPLAY FLAG NAME

CALL TXBYTE

INC   IX

REGDF1: LD   A,(IX+0)

CP    " "

RET   Z

INC   IX

LD    D,A

CALL TXBYTE

JR    REGDF1

; IN = B LOOP

;    C DATA

REGDFX: CALL REGDF      ;DISPLAY FLAG & DATA

LD    D,"1"

SLA  C

JR   C,REGDFX1

LD    D,"0"

REGDFX1: CALL TXBYTE

DJNZ REGDFX

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** NEW ROUTINE *****
; CLEAR REGISTOR
; N
NEW:   CP   0
        JP  NZ,ERROR
        XOR A                ;CLEAR
        LD  HL,USRH1+2
        LD  B,22
NEW1:  DEC  HL
        LD  (HL),A
        DJNZ NEW1
        LD  HL,USRSTK
        LD  (USRSP),HL
        LD  HL,USRSTK
        LD  (USRPC),HL
        LD  HL,REGD          ;TO DISPLAY
        EX  (SP),HL
        RET

```

```

; ***** SEARCH ROUTINE *****
; SEARCH DATA
; S RANGE BYTE ... ( 8 BYTE MAX )
SEARCH: CALL RANGE
        CALL NEXT
        CP   0
        JP  Z,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,8
LD (SEACNT),A
LD IX,SEABUF
SEARCH1: CALL VAL ;STORE DATA TO SEABUF
LD A,(MONDAT)
LD (IX+0),A
INC IX
CALL NEXT
CP 0
JR Z,SEARCH2
LD A,(SEACNT)
DEC A
LD (SEACNT),A
JR NZ,SEARCH1
JP ERROR
SEARCH2: LD A,9
LD HL,SEACNT
SUB (HL)
LD (HL),A ;BYTE COUNT
CALL CLEN
SEARCH3: PUSH HL ;COMPARE SEABUF
PUSH BC
PUSH HL
POP IX
LD DE,SEABUF
LD A,(SEACNT)
LD B,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEARCH31: LD  A,(DE)
          CP  (HL)
          JR  NZ,SEARCH32
          INC HL
          INC DE
          DJNZ SEARCH31
          PUSH IX          ;SEARCH FOUND
          POP  DE
          CALL TXBYTE4     ;DISPLAY FIRST ADDR
          CALL TXLF
SEARCH32: POP  BC
          POP  HL
          INC  HL
          DEC  BC
          LD   A,B
          OR   C
          JR   NZ,SEARCH3
          JP   MON

```

```
; ***** UNASM ROUTINE *****
```

```
; Z80 UNASSEMBLER
```

```
; U [N] [ADDR1] [ADDR2]
```

```

UNASM:   PUSH AF
          XOR  A
          LD   (UASMFAG),A  ;NO DISPLAY ADDRESS
          POP  AF
          CP  0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    Z,UNASMO
LD    HL,MONBUF
LD    B,0
LD    A,(MONINX)
LD    C,A
ADD   HL,BC
LD    A,(HL)
CALL  UPPER
CP    "N"
JR    NZ,UNASMO
LD    A,1
LD    (UASMFAG),A ;NO DISPLAY ADDRESS
CALL  NEXT
UNASMO: LD HL,16
LD    (DEST),HL
CALL  AD2
XOR   A
LD    (UASMCNT),A
LD    A,D ;CHECK FINAL SET
CP    0
JR    Z,UNASM01 ;SET
LD    A,16 ;END BY LINE COUNT
LD    (UASMCNT),A
UNASM01:- LD IX,(START)
-UNASM1: LD A,(UASMFAG)
CP    0
JR    Z,UNASM11 ;BY PASS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD B,9 ;NO DISPLAY ADDRESS & OP-CODE
UNASM10: LD D," "
CALL TXBYTE
DJNZ UNASM10
CALL UASM
CP OFFH
JP Z,UNASME
JR UNASM31
UNASM11: PUSH IX
POP DE
CALL TXBYTE4 ;ADDR
LD D,":"
CALL TXBYTE
LD D," "
CALL TXBYTE
CALL UASM
CP OFFH
JP Z,UNASME ;UASM ERROR
LD C,5 ;BLANK COUNT (5*3)
LD A,(IX+3) ;BYTE COUNT
AND OFFH
LD B,A
PUSH IX
POP HL
UNASM2: LD D,(HL)
CALL TXBYTE2 ;DISPLAY OPCODE
LD D," "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXBYTE

DEC C

INC HL

DJNZ UNASM2

UNASM3: LD B,3

CALL TXBLANK

DEC C

JR NZ,UNASM3

UNASM31: LD A,(IY+0) ;MNEMONIC INDEX

CALL MNESUB

LD D," "

CALL TXBYTE

LD A,(IY+1) ;OPERAND1 INDEX

CP 0

JR Z,UNASM4

CALL OPRSUB

LD A,(IY+2) ;OPERAND2 INDEX

CP 0

JR Z,UNASM4

PUSH AF

LD D," ,"

CALL TXBYTE

POP AF

CALL OPRSUB3

UNASM4: CALL TXLF

LD D,0

LD A,(IY+3)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

AND OFH
LD E,A
ADD IX,DE
LD A,(UASMCNT) ;CHECK END
CP 0
JR Z,UNASM5
DEC A ;END BY LINE COUNT
JR Z,UNASM6
LD (UASMCNT),A
JP UNASM1
UNASM5: LD BC,(FINAL) ;END BY ADDRESS
INC BC
CALL COMP16
JP C,UNASM1
UNASM6: LD (FINAL),IX
JP MON
UNASME: LD HL,UNASMTAB ;UNASM ERROR
CALL TXBLOCK
JP MON
MNESUB: DEC A ;MNEMONIC PRINT
LD B,A
LD C,4
LD HL,MNETAB
CALL TABLE
LD B,4
MNESUB1: LD D,(HL)
CALL TXBYTE

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC HL
DJNZ MNESUB1
RET
OPRSUB: LD D,A
LD A,(IY+4) ;CHECK NUMBER
BIT 7,A
JR NZ,OPRSUB1
LD A,D
JR OPRSUB3
OPRSUB1: AND 7FH ;IS IM,BIT,RES,SET,RST
LD D,A
LD A,(IY+0)
CP 59
JR Z,OPRSUB2
LD A,D
CALL HTOA
LD D,A
CALL TXBYTE
RET
OPRSUB2: CALL TXBYTE2 ;RST
LD D,"H"
CALL TXBYTE
RET
OPRSUB3: CP 1 ;CHECK # (#) (IX+D) (IY+D)
JR Z,OPR1
CP 2
JR Z,OPR2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      8
JR      Z,OPR8
CP      10
JR      Z,OPRA
DEC     A
LD      B,A
LD      C,4
LD      HL,OPRTAB
CALL    TABLE
LD      B,4
OPRSUB4: LD  D,(HL)
LD      A,D
CP      " "
RET     Z
CALL    TXBYTE
INC     HL
DJNZ   OPRSUB4
RET
OPR1:   CALL OPRT      ;N NN DIS
RET
OPR2:   LD  D,"("      ;(N) (NN)
CALL    TXBYTE
CALL    OPRT
LD      D,")"
CALL    TXBYTE
RET
OPR3:   LD  HL,OPRIX   ;(IX+D)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR OPRA1
OPRA: LD HL,OPRIY ;(IY+D)
OPRA1: CALL TXBLOCK
LD D,(IX+2)
CALL OPBYTE2
LD D,"H"
CALL TXBYTE
LD D,")"
CALL TXBYTE
RET
OPRIX: DFB "(IX+",00H
OPRIY: DFB "(IY+",00H
; DISPLAY TYPE 1,2,3,5,6
OPRT: LD A,(IY+3) ;N NN (N) (NN) DIS
AND OFOH
CP 10H
JR Z,OPRT1
CP 20H
JR Z,OPRT2
CP 30H
JR Z,OPRT3
CP 50H
JR Z,OPRT5
JR OPRT6
OPRT1: LD D,(IX+1) ;N (N)
JR OPRT51
OPRT5: LD D,(IX+3)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OPRT51:  CALL OPBYTE2
         LD  D,"H"
         CALL TXBYTE
         RET

OPRT2:   LD  D,(IX+2)      ;NN (NN)
         LD  E,(IX+1)
         JR  OPRT61

OPRT6:   LD  D,(IX+3)
         LD  E,(IX+2)

OPRT61:  CALL OPBYTE4
         LD  D,"H"
         CALL TXBYTE
         RET

OPRT3:   PUSH IX          :DIS
         POP  HL
         LD  D,0
         LD  E,(IX+1)
         BIT 7,E
         JR  Z,OPRT31
         LD  D,0FFH

OPRT31:  ADD  HL,DE
         INC -HL
         INC HL
         PUSH HL
         POP  DE
         CALL OPBYTE4
         LD  D,"H"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXBYTE
RET

; **** OPERAND BYTE2 & BYTE4 SUB. ****
; CHECK IF FIRST BYTE = A-F THEN ADD 0
OPBYTE2: LD  A,D          ;IN = D
          CALL OPBYTES
          JR  C,OPBYTE21  ;NOT A-F
          PUSH DE
          LD  D,"0"
          CALL TXBYTE
          POP DE
OPBYTE21: CALL TXBYTE2
          RET
OPBYTE4: LD  A,D          ;IN = DE
          CALL OPBYTES
          JR  C,OPBYTE41
          PUSH DE
          LD  D,"0"
          CALL TXBYTE
          POP DE
OPBYTE41: CALL TXBYTE4
          RET
OPBYTES: RRCA
          RRCA
          RRCA
          RRCA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AND 0FH

CP 0AH

RET

; \*\*\*\*\* VAR ROUTINE \*\*\*\*\*

; VARIABLE DISPLAY & SET

; V [VAR VALUE]

; VAR = 10,18,20,28,30,38,66

```
VAR:    CP    0
        JR    Z,VAR2    ;DISPLAY ONLY
        CALL VAL
        LD    HL,(MONDAT)
        LD    A,H
        CP    0
        JP    NZ,ERROR
        LD    A,L    ;CHECK VAR
        LD    HL,VRTAB
        LD    BC,700H
VAR1:   CP    (HL)
        JR    Z,VAR2
        INC  HL
        INC  C
        INC  C
        DJNZ VAR1
        JP   ERROR
VAR2:   LD    B,0
        LD    IX,RST10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD IX,BC
CALL NEXT
CP 0
JP Z,ERROR
CALL VAL ;VALUE
CALL NEXT
CP 0
JP NZ,ERROR
LD HL,(MONDAT)
LD (IX+0),L
LD (IX+1),H
JP MON
VARD: LD IX,RST10 ;DISPLAY VAR
LD IY,VARTAB
LD B,7
VARD1: LD HL,VARTAB1
CALL TXBLOCK
LD D,(IY+0)
CALL TXBYTE2
LD D," "
CALL TXBYTE
LD E,(IX+0)
LD D,(IX+1)
CALL TXBYTE4
CALL TXLF
INC IY
INC IX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INC IX
DJNZ VARD1
JP MON
```

```
: ***** TRACE ROUTINE *****
```

```
; TRACE USER PROGRAM (SENGLE STEP)
```

```
; T [ADDR]
```

```
TRACE: CP 0
JR Z,TRACE1
CALL VAL
CALL NEXT
CP 0
JP NZ,ERROR
LD HL,(MONDAT)
LD (USRPC),HL
TRACE1: LD IX,(USRPC)
CALL UASM
CP 0
JR Z,TRACE2
LD HL,TRATAB ;COMMAND ERROR
CALL TXBLOCK
JP MON
TRACE2: LD A,(IY+3) ;GO
AND OFH
LD E,A
LD D,0
LD HL,(USRPC)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD HL,DE

LD A,(HL)

LD (BRKMEM),A

LD (HL),0CFH ;RST 0SH OP-CODE

LD (STPADD).HL

LD HL,BRKFLAG

SET 1,(HL)

JP EXIT

```

```

; ***** QUIT ROUTINE *****

```

```

; QUIT (POWER OFF OR RESET)

```

```

; Q

```

```

QUIT: CP 0
      JP NZ,ERROR
      LD HL,QUITTB
      CALL TXBLOCK
      XOR A
      LD (POWMEM),A
      HALT

```

```

; ***** MOVES SUB. ***** ABCDEHL

```

```

; MOVE MEMORY

```

```

; IN = HL START ADDR

```

```

; DE FINAL ADDR

```

```

; BC DESTINATION ADDR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOVES:  XOR  A
        EX   DE,HL
        SBC  HL,DE          ;FINAL-START
        PUSH HL            ;3
        PUSH BC            ;2
        PUSH BC            ;1
        POP  HL            ;1
        XOR  A
        SBC  HL,DE          ;DEST-START
        POP  HL            ;2
        EX   DE,HL          ;DEST,START
        POP  BC            ;3
        JR   NC,MOVES1
        INC  BC              ;LENGTH
        LDIR              ;MOVE INC
        RET
MOVES1: ADD  HL,BC          ;START
        EX   DE,HL
        ADD  HL,BC          ;DEST
        EX   DE,HL
        INC  BC
        LDDR
        RET

```

```

; ***** SEAR SUB. ***** ADEHLIX (SEAMEM) (SEAFND) (SEACNT)
; SEARCH STRING
; IN = B FLAG 0=NORMAL 1=BLANK-END
; C TABLE LENGTH
; HL NO. OF STRING IN TABLE
; IX START TABLE
; IY START SEARCH STRING
; OUT = HL NO. OF STRING 0=NOT FOUND
; HL=(SEAFND) , A=L
SEAR: LD (SEAMEM),HL
      LD HL,1
      LD (SEAFND),HL
SEAR1: LD A,C
       LD (SEACNT),A
       PUSH IX
       POP HL
       PUSH IY
       POP DE
SEAR2: LD A,(DE)
       CP (HL)
       JR NZ,SEAR4 ;NOT EQUAL
       INC HL
       INC DE
       LD A,B
       CP 0
       JR Z,SEAR3 ;NORMAL
       LD A,(DE) ;CHECK END BY BLANK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP " "
JR Z,SEARF ;SEARCH FOUND
SEAR3: LD A,(SEACNT)
DEC A
LD (SEACNT),A
JR NZ,SEAR2 ;NEXT CHAR.
JR SEARF ;SEARCH FOUND
SEAR4: LD D,0
LD E,C
ADD IX,DE
LD HL,(SEAFND)
INC HL
LD (SEAFND),HL
LD HL,(SEAMEM)
DEC HL
LD (SEAMEM),HL
LD A,H
OR L
JR NZ,SEAR1 ;NEXT STRING
LD (SEAFND),HL
RET ;NOT FOUND
SEARF: LD HL,(SEAFND) ;FOUND
LD A,L
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** TABLE SUB. ***** ABHL
; TABLE LOOKUP
; IN = B NO. 0 - 255
; C TABLE LENGTH
: HL START TABLE
; OUT = (HL) FIRST DATA IN TABLE

```

```

TABLE:  PUSH DE
        PUSH HL
        INC  B
        LD  D,0
        LD  E,C
        LD  HL,0
TABLE1:  DEC  B
        JR  Z, TABLE2
        ADD HL, DE
        JR  TABLE1
TABLE2:  EX  DE, HL
        POP HL
        ADD HL, DE
        POP DE
        RET

```

```

; ***** D10M SUB. ***** AB
; DELAY 0.01 SECOND
; IN = B COUNT
D10M:   PUSH HL
D10M1:  LD   HL, 1376

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
D10M2: DEC HL
        LD A,H
        OR L
        JR NZ,D10M2
        DJNZ D10M1
        POP HL
        RET
```

```
: ***** DSEC SUB. ***** AB
; DELAY SECOND
; IN = B SECOND COUNT
DSEC:   PUSH HL
DSEC1:  LD HL,52640
DSEC2:  PUSH HL
        POP HL
        PUSH HL
        POP HL
        DEC HL
        LD A,H
        OR L
        JR NZ,DSEC2
        DJNZ DSEC1
        POP HL
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** COMP16 SUB. ***** A
; COMPARE IX-BC
; IN = IX BC
: OUT = FLAG
COMP16:  PUSH HL
        PUSH IX
        POP HL
        XOR  A
        SBC  HL,BC
        POP HL
        RET
: ***** HTOA SUB. ***** ADE
; HEX TO ASCII
; IN = A HEX
; OUT = DE ASCII (A=E)
; eg. 2A -> 32 41
HTOA:   PUSH AF
        AND  0F0H
        RRCA
        RRCA
        RRCA
        RRCA
        CALL HTOAS
        LD   D,A
        POP AF
        AND  0FH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL HTOAS

LD E,A

RET

HTOAS: AND 0FH

CP 0AH

JR NC,HTOAS1

OR 30H

RET

HTOAS1: SUB 9

OR 40H

RET

; ***** ATOH SUB. ***** AD
; ASCII TO HEX
; IN = DE ASCII
; OUT = A HEX
ATOH: LD A,D

CALL ATOHS

RLCA

RLCA

RLCA

RLCA

LD D,A

LD A,E

CALL ATOHS

OR D

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** ATOHS SUB. ***** A
; ASCII TO HEX ONE BYTE
; IN = A ASCII
; OUT = A HEX
ATOHS:   CP   "A"
         JR   NC,ATOHS1
         AND  0FH
         RET
ATOHS1:  SUB  7
         AND  0FH
         RET
; ***** UPPER SUB. ***** A
; LOWER TO UPPER CHARECTER
; IN = A LOWER
; OUT =A UPPER
UPPER:   CP   "a"
         RET  C
         CP   "z"+1
         RET  NC
         SUB  32
         RET
; ***** TMGETL SUB. *****
; GET LINE FROM TERMINAL ( END BY CR )
; OUT = (MONBUF) COMMAND BUF
;      (MONINX) LENGTH
; REG = ABCDHL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TMGETL:  XOR  A

          LD   (MONINX),A      ;CLEAR INDEX

TMGETL1: CALL RXBYTE

          LD   A,D              ;CHECK KEY

          CP   0SH

          JR   Z,TMGETBS

          CP   0DH

          JR   Z,TMGETL2

          LD   A,(MONINX)      ;CHK MONBUF FULL

          CP   60              ;AT 61 FOR CR ONLY

          JR   C,TMGETL2

          LD   D,07H           ;FULL

          CALL TXBYTE

          JR   TMGETL1

TMGETL2: CALL TXBYTE          ;ECHO

          LD   A,(MONINX)      ;UP INDEX

          LD   C,A

          INC  A

          LD   (MONINX),A

          LD   HL,MONBUF       ;DATA TO BUF

          LD   B,0

          ADD  HL,BC

          LD   (HL),D

          LD   A,D

          CP   0DH

          JR   NZ,TMGETL1

          LD   D,0AH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL TXBYTE

RET                                ;EXIT

TMGETBS: LD  A,(MONINX)            ;BACKSPACE

CP  0

JR  Z.TMGETL1

DEC  A

LD  (MONINX).A

LD  D,0SH

CALL TXBYTE

LD  D," "

CALL TXBYTE

LD  D,0SH

CALL TXBYTE

JR  TMGETL1

; ***** RXBYTE SUB *****
; RECEIVE ONE BYTE FROM JACK
; OUT = D DATA (A=D)
; REG = AD

RXBYTE:  EI

ZZZ:    SLP                        ;CPU SLEEP FOR WAIT ASCI INT

RET

RXBYTE1: DI

PUSH AF

IN0  A,(RDR1)                      ;CLEAR RDRF & READ KEYBOARD

OUT  (UPA),A                        ;SHOW ASCII ON LED FLAG

LD   D,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP AF

RETI ;RETURN FROM INTERRUPT

; ***** RXBYTE2 SUB. ***** ADE
; RECEIVE ONE ASCII FROM COM PORT
; OUT = D DATA IN HEX (A=D)
RXBYTE2: CALL RXBYTE
        LD E,D
        CALL RXBYTE
        LD D,E
        LD E,A
        CALL ATOH
        LD D,A
        RET

; ***** TXBYTE SUB *****
; SEND 1 CHARECTER TO RS-232 PORT
; IN = D
; REG = E
TXBYTE: PUSH DE
TXBYTE1: INO E,(STAT1) ;CHECK FLAG SEND
        BIT 1,E
        JR Z,TXBYTE1 ;FLAG TDRE ASCI=1?
        OUTO (TDR1),D ;YES,SEND TO DISPLAY
        POP DE
        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; ***** TXBYTE2 SUB. ***** ADE
```

```
; SEND ONE BYTE IN ASCII TO COM PORT
```

```
; IN = D DATA
```

```
TXBYTE2: LD A,D
```

```
CALL HTOA
```

```
CALL TXBYTE
```

```
LD D,E
```

```
CALL TXBYTE
```

```
RET
```

```
: ***** TXBYTE4 SUB. ***** ADE
```

```
; SEND TWO BYTE IN ASCII TO COM PORT
```

```
; IN = BIT 0 (STSFAG)
```

```
: DE DATA
```

```
TXBYTE4: PUSH DE
```

```
CALL TXBYTE2
```

```
POP DE
```

```
LD D,E
```

```
CALL TXBYTE2
```

```
RET
```

```
; ***** TXBLOCK SUB. ***** ADHL
```

```
; SEND BLOCK (END BLOCK BY 00H OR 0DH) TO COM PORT
```

```
; IN = HL START BLOCK
```

```
TXBLOCK: LD A,(HL)
```

```
CP 0
```

```
RET Z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP    0DH
JR    Z, TXBLOCK1

LD    D, A

CALL  TXBYTE

INC   HL

JR    TXBLOCK

TXBLOCK1: CALL TXLF

RET

; ***** TXBLANK SUB. ***** ABD B=0
; SEND BLANK TO COM PORT
; IN = B BLANK COUNTER
TXBLANK: LD    D, " "
          CALL TXBYTE
          DJNZ TXBLANK
          RET

; ***** TXLF SUB. ***** AD
; SEND CR LF TO COM PORT
TXLF:   LD    D, 0DH

          CALL TXBYTE

          LD    D, 0AH

          CALL TXBYTE

          RET

```

```

; ***** ASMS SUB. *****
; Z80 INSTRUCTION TABLE SEARCH (FOR ASSEMBLER)
; IN = (ASMBUF) MNE,OPR1,OPR2
;      (ASMDAT) N NN (N) (NN) DIS
;      (ASMDATD) +D
; OUT = (ASMOPC) OPCODE
;      A OPCODE COUNT (0=NOT FOUND)
ASMS:   LD   HL,0
        LD   (ASMOPC),HL
        LD   (ASMOPC+2),HL
        LD   IY,ASMOPC
        LD   IX,XX-1      ;XX
        LD   BC,0
ASMS1:  CALL ASMSCOMP
        JR   NZ,ASMS12
        LD   (IY+0),C      ;OPCODE
        CALL ASMSX
        RET  NZ
ASMS12: LD   DE,5
        ADD  IX,DE
        INC  C
        DJNZ ASMS1
        LD   A,0CBH      ;CB
        LD   (IY+0),A
        LD   BC,0
ASMS2:  CALL ASMSCOMP
        JR   NZ,ASMS22

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (IY+1),C ;OPCODE
CALL ASMSX
RET NZ
ASMS22: LD DE,5
ADD IX,DE
INC C
DJNZ ASMS2
LD IX,DD ;DD
LD A,ODDH
LD (IY+0),A
LD B,39
ASMS3: CALL ASMSCOMP
JR NZ,ASMS31
CALL ASMS39
RET NZ
ASMS31: CALL ASMSUP
DJNZ ASMS3
LD IX,ED ;ED
LD A,OEDH
LD (IY+0),A
LD B,56
ASMS32: CALL ASMSCOMP
JR NZ,ASMS33
CALL ASMS39
RET NZ
ASMS33: CALL ASMSUP
DJNZ ASMS32

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IX,FD ;FD
LD A,OFDH
LD (IY+0),A
LD B,39
ASMS34: CALL ASMSCOMP
JR NZ,ASMS35
CALL ASMS39
RET NZ
ASMS35: CALL ASMSUP
DJNZ ASMS34
JR ASMS4
ASMS39: LD A,(IX+0) ;DD ED FD FOUND
LD (IY+1),A ;OPCODE BYTE 2
CALL ASMSX
RET
ASMS4: LD IX,DDCB ;DDCB
LD HL,ODDCBH
LD (IY+0),H
LD (IY+1),L
LD B,31
ASMS41: CALL ASMSCOMP
JR NZ,ASMS42
CALL ASMS49
RET NZ
ASMS42: CALL ASMSUP
DJNZ ASMS41
LD IX,FDCB ;FDCB

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,0FDCBH
LD (IY+0),H
LD (IY+1),L
LD B,31
ASMS43: CALL ASMSCOMP
JR NZ,ASMS44
CALL ASMS49
RET NZ
ASMS44: CALL ASMSUP
DJNZ ASMS43
XOR A ;COMMAND ERROR
RET
ASMS49: LD A,(IX+0) ;DDCB FDCB FOUND
LD (IY+3),A ;OPCODE BYTE 4
ASMSX: LD A,(IX+5) ;CHECK NUMBER
BIT 7,A
JR Z,ASMSX1
AND 7FH ;IS IM.BIT,RES,SET,RST
LD D,A
LD A,(ASMDAT)
CP D
JR Z,ASMSX3
XOR A ;NEXT SEARCH
RET
ASMSX1: LD A,(IX+4) ;CHECK TYPE
AND 0FOH
CP 10H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR    Z,ASMXT1
CP    20H
JR    Z,ASMXT2
CP    30H
JR    Z,ASMXT1
CP    40H
JR    Z,ASMXT4
CP    50H
JR    Z,ASMXT5
CP    60H
JR    Z,ASMXT6
ASMSX2: LD    A, (IX++) ;NO. OF BYTE
        AND  0FH
        RET   ;EXIT
ASMSX3: LD    A, (IX++) ;CHECK BIT, RES, SET & +D
        AND  0F0H
        CP    40H
        JR    Z,ASMXT4
        JR    ASMSX2
ASMXT1: LD    A, (ASMDAT)
        LD    (IY+1),A
        JR    ASMSX2
ASMXT2: LD    HL, (ASMDAT)
        LD    (IY+1),L
        LD    (IY+2),H
        JR    ASMSX2
ASMXT4: LD    A, (ASMDATD)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (IY+2),A
JR ASMSX2
ASMXT5: LD A,(ASMDATD)
LD (IY+2),A
LD A.(ASMDAT)
LD (IY+3),A
JR ASMSX2
ASMXT6: LD HL,(ASMDAT)
LD (IY+2),L
LD (IY+3),H
JR ASMSX2
ASMSCOMP: LD A,(ASMBUF) ;COMMAND COMPARE SUB.
CP (IX+1)
RET NZ
LD A,(ASMBUF+1)
CP (IX+2)
RET NZ
LD A,(ASMBUF+2)
CP (IX+3)
RET
ASMSUP: LD DE,6 ;TABLE UP SUB.
ADD IX,DE
RET

```

; \*\*\*\*\* UASM SUB. \*\*\*\*\*

; Z80 INSTRUCTION TABLE SEARCH (FOR UNASSEMBLER)

; IN = (IX+0) OPCODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

: OUT = (IY+0) TABLE MNE,OPR1,OPR2,TYPE-BYTE,NUM
; A 0=OK FF=ERROR
UASM: LD A,(IX+0)
      CP 0CBH
      JR Z,UASMCB
      CP 0DDH
      JR Z,UASMDD
      CP 0EDH
      JR Z,UASMED
      CP 0FDH
      JR Z,UASMFDD
      LD HL,XX
UASM1: LD B,A
      LD C,5
      CALL TABLE
      PUSH HL
      POP IY
      XOR A
      RET
UASMCB: LD A,(IX+i) ;CB
        LD HL,CB
        JR UASM1
UASMDD: LD A,(IX+1) ;DD
        CP 0CBH
        JR Z,UASMDDCB
        LD HL,DD
UASMDD1: LD BC,6

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UASMDD2: LD D, (HL)

          CP D

          JR Z, UASMDD3

          JR C, UASMERR

          ADD HL, BC

          JR UASMDD2

UASMDD3: INC HL

          PUSH HL

          POP IY

          XOR A

          RET

UASMDDCB: LD A, (IX+3) ; DDCB

          LD HL, DDCB

          JR UASMDD1

UASMED: LD A, (IX+1) ; ED

          LD HL, ED

          JR UASMDD1

UASMFDCB: LD A, (IX+1) ; FD

          CP 0CBH

          JR Z, UASMFDCB

          LD HL, FD

          JR UASMDD1

UASMFDCB: LD A, (IX+3) ; FDCB

          LD HL, FDCB

          JR UASMDD1

UASMERR: LD A, OFFH ; NOT FOUND OPCODE

          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** NEXT SUB. *****
; SEARCH NEXT FIELD OF COMMAND LINE
; IN = (MONINX) START SEARCH
; OUT = (MONINX) A
:      IF A=0 THEN NO NEXT FIELD & (MONINX) IS END

```

```

NEXT:   LD  A,(MONINX)
        LD  C,A
        LD  B,0
NEXT1:  INC  C ;SEARCH BLANK
        LD  HL,MONBUF
        ADD HL,BC
        LD  A,(HL)
        CP  ODH
        JR  Z,NEXTE
        CP  " "
        JR  Z,NEXT2 ;FOUND BLANK
        JR  NEXT1
NEXT2:  INC  C ;SEARCH NOT BLANK
        LD  HL,MONBUF
        ADD HL,BC
        LD  A,(HL)
        CP  ODH
        JR  Z,NEXTE
        CP  " "
        JR  Z,NEXT2
        LD  A,C ;FOUND

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (MONINX),A
RET
NEXTE: LD A,C ;NO NEXT FIELD
LD (MONINX),A
XOR A
RET
; ***** VAL SUB. *****
: GET VALUE FROM COMMAND LINE
; IN = (MONINX) START FIELD
; OUT = (MONDAT)-(MONDAT+1) IN 8 BIT LOOK AT (MONDAT) ONLY
VAL: LD A,(MONINX)
LD HL,MONBUF
LD B,0
LD C,A
ADD HL,BC
EX DE,HL
LD HL,0
LD (MONDAT),HL ;CLEAR
LD B,6
VAL1: LD A,(DE)
CP ODH
RET Z
CP " "
RET Z ;END FIELD - EXIT
CP "H"
JR Z,VAL2
CALL CHEX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP    OFFH

JP    Z,ERROR

LD    HL,MONDAT

RLD

INC   HL

RLD

VAL2: INC   DE

      DJNZ VAL1          ;IF B=0 IS OVER DIGIT

      JP    ERROR

; ***** AD1 SUB. *****
; GET [ADDR]
; IN = (MONINX),A START FIELD
; OUT = (START)
AD1:  CP    0          ;CHECK ADDR
      RET   Z          ;DEFAULT
      CALL VAL
      CALL NEXT
      CP    0
      JP    NZ,ERROR
      LD    HL,(MONDAT)
      LD    (START),HL

      RET

; ***** AD2 SUB. *****
; GET [ADDR1] [ADDR2]
; IN = (MONINX) START FIELD
;      (DEST) DEFAULT FINAL LENGTH
; OUT = (START)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; (FINAL)
; D DEFAULT USED 0=NO 1=YES

AD2: CP 0 ;CHECK ADDR1
JR NZ,AD21
LD HL,(FINAL) ;DEFAULT
LD (START),HL
JR AD22
AD21: CALL VAL ;ADDR1
CALL NEXT
LD HL,(MONDAT)
LD (START),HL
CP 0
JR Z,AD22
CALL VAL ;ADDR2
LD HL,(MONDAT)
LD (FINAL),HL
CALL NEXT
CP 0
JP NZ,ERROR ;FIELD OVER
LD D,0
RET
AD22: LD DE,(DEST) ;DEFAULT
ADD HL,DE
LD (FINAL),HL
LD D,1
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** RANGE SUB. *****
; GET RANGE
; IN = (MONINX) START FIELD
; OUT = (START)
;      (FINAL)
RANGE:  CP  0
        JP  Z,ERROR
        CALL VAL
        LD  HL,(MONDAT)
        LD  (START),HL
        CALL NEXT
        CP  0
        JP  Z,ERROR
        CALL VAL
        LD  HL,(MONDAT)
        LD  (FINAL),HL
        RET

; ***** CHEX SUB. *****
; CHECK HEX DIGIT AND CHANGE TO HEX
; IN = A ASCII DIGIT
; OUT = A HEX
; IF IN=0-9,A-F THEN OUT=HEX ELSE OUT=0FFH
CHEX:   CP  "0"
        JR  C,CHEXE
        CP  "9"+1
        JR  C,CHEX1      ;IS 0-9

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL UPPER

CP  "A"

JP  C,CHEXE

CP  "F"+i

JP  NC,CHEXE

ADD  A,9                ;IS A-F

CHEX1:  AND  0FH

RET

CHEXE:  LD  A,0FFH      ;ERROR CODE

RET

; ***** CLEN SUB. ***** ABCHL
; CAL LENGTH (FINAL)-(START)
; OUT = BC LENGTH
; HL START
CLEN:   PUSH DE
        LD  DE,(START)
        LD  HL,(FINAL)
        XOR  A
        SBC  HL,DE

        INC  HL                ;LENGTH

        EX  DE,HL

        PUSH DE

        POP  BC

        POP  DE

RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

; ***** AUTOB SUB. *****
; OUT = (BAUDLY)
; REG = ALL
AUTOB:  LD  A,64H
        OUT0 (CNTLA1),A      ;TX,RX,S BIT,: STOP
        LD  A,2              ;9600 BAUD AT X'TAL 12.488 Mhz
        OUT0 (CNTLB1),A
        LD  A,S              ;RIE ENABLE TO KEYBOARD
        OUT0 (STAT1),A
        LD  A,96H
        LD  (BAUD),A
        RET
; *****PAGE MEMORY ROUTINE*****
; SELECT PAGE MEMORY FOR PROGRAM MMU [SEE APPENDIC (PAGE MEMORY) ]
; X [NUMBER PAGE]
SPM:    CP 0
        JP Z,ERROR
        CALL VAL
        CALL NEXT
        CP 0
        JP NZ,ERROR
        LD A,(MONDAT)
        CP 0
        JP Z,ERROR
        CP 42H
        JP NC,ERROR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD HL,PXTAB
LD IX,MMUTAB
LD B,42
SPM1: CP (HL)
JR Z,SPM2
INC HL
INC IX
DJNZ SPM1
SPM2: CALL NEXT
LD A,(IX+0)
OUT0 (BBR),A
JP MON
; ***** ROM TABLE *****
LCDTAB: DFB " Remote Monitor "
MNOTAB: DFB " ",ODH
MN1TAB: DFB " ZS01S0 Remote Monitor v1.0 ",ODH
MN2TAB: DFB " Industrial Education KM4TL. ",ODH
MN3TAB: DFB " Electronic & Computer ",ODH
MN4TAB: DFB " ED.ENGINEER #15 ",ODH
BAUTAB: DFB " Baud Rate :",ODH
MN5TAB: DFB " ",ODH
MN6TAB: DFB " ",ODH
MN7TAB: DFB " ",ODH
POWTAB: DFB "POWER UP !",ODH
SYNTAB: DFB "^ Syntax Error",ODH
MN8TAB: DFB "ABCDEFGHILMNOPQRSTUVWXYZ"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MN9TAB:   DWL   ASM,BREAK,COMPARE,DISPY,ENTER
          DWL   FILL,GO,HELP,INPUT,LOAD
          DWL   MOVE,NEW,OUTPUT,UPLOAD,QUIT
          DWL   REG,SEARCH,TRACE,UNASM,VAR,SPM
HELPTB:   DFB   "A addr\Assembler",0DH
          DFB   "B [addr]\Break",0DH
          DFB   "C range addr\Compare block",0DH
          DFB   "  range H byte\Compare byte",0DH
          DFB   "D [addr1] [addr2]\Display memory",0DH
          DFB   "E addr\Enter data",0DH
          DFB   "F range byte\Fill memory",0DH
          DFB   "G [addr]\Go (Run)",0DH
          DFB   "H \Help",0DH
          DFB   "I port\Input",0DH
          DFB   "L [addr]\downLoad from terminal",0DH
          DFB   "M range addr\Move",0DH
          DFB   "N \New (Clear registor)",0DH
          DFB   "O port byte\Output",0DH
          DFB   "P range\upload to terminal",0DH
          DFB   "Q \Quit/Halt (Power off or Reset)",0DH
          DFB   "R [reg value]\Register",0DH
          DFB   "S range byte ... (8)\Search",0DH
          DFB   "T [addr]\Trace (Single step)",0DH
          DFB   "U [N] [addr1] [addr2]\Unassembler",0DH
          DFB   "V [var value]\Vector (RSTn)",0DH
          DFB   "X [number page]\Select user page(1 - 41)",0DH,00H
UPLTB1:   DFB   "Upload INTEL-HEX FILE ...",0DH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UPLTB2: DFB "Use ^A and CA Command to Upload File (XTALK)",ODH
UPLTB3: DFB "Type ENTER to Start and Finish",ODH
UPLHEX: DFB ":00000001FF",ODH
REGTB1: DFB "PC SP IF IX IY AF BC DE HL " ;16 bit reg. table
DFB "AF'BC'DE'HL'"
RECTB2: DFB "S= Z= H= PV= N= C= " ;flag print table
REGTB3: DFB "I F A C B E D L H " ;8 bit reg. table
DFB "F'A'C'B'E'D'L'H'"
REGTB4: DFB "SFZF HF PNFNCF" ;flag table
LOADTAB: DFB "Load Error ... Enter to Continue",ODH
LOADTAB2: DFB "Download INTEL-HEX FILE ...",ODH
LOADTAB3: DFB "Use ^A and SE Command to Send File (XTALK)",ODH
UNASMTAB: DFB "Opcode Error",ODH
ASMER1: DFB "Mnemonic Error",ODH
ASMER2: DFB "Operand Error",ODH
ASMER3: DFB "Command Error",ODH
JUMPTAB: DFB "Relative Error",ODH
VARTAB: DFB 10H,18H,20H,28H,30H,38H,66H
VARTAB1: DFB "RST",00H
TRATAB: DFB "Command Error",ODH
QUITTB: DFB "Z80180 QUIT/HALT ! Power off or Reset ...",ODH
PXTAB: DFB 01H,02H,03H,04H,05H,06H,07H,08H
DFB 09H,10H,11H,12H,13H,14H,15H,16H
DFB 17H,18H,19H,20H,21H,22H,23H,24H
DFB 25H,26H,27H,28H,29H,30H,31H,32H
DFB 33H,34H,35H,36H,37H,38H,39H,40H
DFB 41H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MMUTAB:  DFB      00H,06H,0CH,12H,18H,1EH,24H,2AH
          DFB      30H,36H,3CH,42H,48H,4EH,54H,5AH
          DFB      60H,66H,6CH,72H,78H,7EH,84H,8AH
          DFB      90H,96H,9CH,0A2H,0ASH,0AEH,0B4H
          DFB      0C0H,0C6H,0CCH,0D2H,0D8H,0DEH
          DFB      0E4H,0EAH,0FOH

```

; \*\*\*\*\* Z80 ROM TABLE \*\*\*\*\*

```

XX:      DFB      24H,00H,00H,01H,00H
          DFB      1EH,10H,01H,23H,00H
          DFB      1EH,03H,0CH,01H,00H
          DFB      17H,10H,00H,01H,00H
          DFB      17H,0FH,00H,01H,00H
          DFB      0EH,0FH,00H,01H,00H
          DFB      1EH,0FH,01H,12H,00H
          DFB      34H,00H,00H,01H,00H
          DFB      12H,0DH,0EH,01H,00H
          DFB      02H,16H,10H,01H,00H
          DFB      1EH,0CH,03H,01H,00H
          DFB      0EH,10H,00H,01H,00H
          DFB      17H,11H,00H,01H,00H
          DFB      0EH,11H,00H,01H,00H
          DFB      1EH,11H,01H,12H,00H
          DFB      39H,00H,00H,01H,00H
          DFB      10H,01H,00H,32H,00H
          DFB      1EH,13H,01H,23H,00H
          DFB      1EH,05H,0CH,01H,00H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 17H, 13H, 00H, 01H, 00H  
DFB 17H, 12H, 00H, 01H, 00H  
DFB 0EH, 12H, 00H, 01H, 00H  
DFB 1EH, 12H, 01H, 12H, 00H  
DFB 32H, 00H, 00H, 01H, 00H  
DFB 1DH, 01H, 00H, 32H, 00H  
DFB 02H, 16H, 13H, 01H, 00H  
DFB 1EH, 0CH, 05H, 01H, 00H  
DFB 0EH, 13H, 00H, 01H, 00H  
DFB 17H, 14H, 00H, 01H, 00H  
DFB 0EH, 14H, 00H, 01H, 00H  
DFB 1EH, 14H, 01H, 12H, 00H  
DFB 37H, 00H, 00H, 01H, 00H  
DFB 1DH, 1DH, 01H, 32H, 00H  
DFB 1EH, 16H, 01H, 23H, 00H  
DFB 1EH, 02H, 16H, 23H, 00H  
DFB 17H, 16H, 00H, 01H, 00H  
DFB 17H, 15H, 00H, 01H, 00H  
DFB 0EH, 15H, 00H, 01H, 00H  
DFB 1EH, 15H, 01H, 12H, 00H  
DFB 0DH, 00H, 00H, 01H, 00H  
DFB 1DH, 23H, 01H, 32H, 00H  
DFB 02H, 16H, 16H, 01H, 00H  
DFB 1EH, 16H, 02H, 23H, 00H  
DFB 0EH, 16H, 00H, 01H, 00H  
DFB 17H, 1AH, 00H, 01H, 00H  
DFB 0EH, 1AH, 00H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH, 1AH, 01H, 12H, 00H  
DFB 0CH, 00H, 00H, 01H, 00H  
DFB 1DH, 1CH, 01H, 32H, 00H  
DFB 1EH, 22H, 01H, 23H, 00H  
DFB 1EH, 02H, 0CH, 23H, 00H  
DFB 17H, 22H, 00H, 01H, 00H  
DFB 17H, 06H, 00H, 01H, 00H  
DFB 0EH, 06H, 00H, 01H, 00H  
DFB 1EH, 06H, 01H, 12H, 00H  
DFB 3DH, 00H, 00H, 01H, 00H  
DFB 1DH, 11H, 01H, 32H, 00H  
DFB 02H, 16H, 22H, 01H, 00H  
DFB 1EH, 0CH, 02H, 23H, 00H  
DFB 0EH, 22H, 00H, 01H, 00H  
DFB 17H, 0CH, 00H, 01H, 00H  
DFB 0EH, 0CH, 00H, 01H, 00H  
DFB 1EH, 0CH, 01H, 12H, 00H  
DFB 06H, 00H, 00H, 01H, 00H  
DFB 1EH, 0FH, 0FH, 01H, 00H  
DFB 1EH, 0FH, 11H, 01H, 00H  
DFB 1EH, 0FH, 12H, 01H, 00H  
DFB 1EH, 0FH, 14H, 01H, 00H  
DFB 1EH, 0FH, 15H, 01H, 00H  
DFB 1EH, 0FH, 1AH, 01H, 00H  
DFB 1EH, 0FH, 06H, 01H, 00H  
DFB 1EH, 0FH, 0CH, 01H, 00H  
DFB 1EH, 11H, 0FH, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH, 11H, 11H, 01H, 00H  
DFB 1EH, 11H, 12H, 01H, 00H  
DFB 1EH, 11H, 14H, 01H, 00H  
DFB 1EH, 11H, 15H, 01H, 00H  
DFB 1EH, 11H, 1AH, 01H, 00H  
DFB 1EH, 11H, 06H, 01H, 00H  
DFB 1EH, 11H, 0CH, 01H, 00H  
DFB 1EH, 12H, 0FH, 01H, 00H  
DFB 1EH, 12H, 11H, 01H, 00H  
DFB 1EH, 12H, 12H, 01H, 00H  
DFB 1EH, 12H, 14H, 01H, 00H  
DFB 1EH, 12H, 15H, 01H, 00H  
DFB 1EH, 12H, 1AH, 01H, 00H  
DFB 1EH, 12H, 06H, 01H, 00H  
DFB 1EH, 12H, 0CH, 01H, 00H  
DFB 1EH, 14H, 0FH, 01H, 00H  
DFB 1EH, 14H, 11H, 01H, 00H  
DFB 1EH, 14H, 12H, 01H, 00H  
DFB 1EH, 14H, 14H, 01H, 00H  
DFB 1EH, 14H, 15H, 01H, 00H  
DFB 1EH, 14H, 1AH, 01H, 00H  
DFB 1EH, 14H, 06H, 01H, 00H  
DFB 1EH, 14H, 0CH, 01H, 00H  
DFB 1EH, 15H, 0FH, 01H, 00H  
DFB 1EH, 15H, 11H, 01H, 00H  
DFB 1EH, 15H, 12H, 01H, 00H  
DFB 1EH, 15H, 14H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH, 15H, 15H, 01H, 00H  
DFB 1EH, 15H, 1AH, 01H, 00H  
DFB 1EH, 15H, 06H, 01H, 00H  
DFB 1EH, 15H, 0CH, 01H, 00H  
DFB 1EH, 1AH, 0FH, 01H, 00H  
DFB 1EH, 1AH, 11H, 01H, 00H  
DFB 1EH, 1AH, 12H, 01H, 00H  
DFB 1EH, 1AH, 14H, 01H, 00H  
DFB 1EH, 1AH, 15H, 01H, 00H  
DFB 1EH, 1AH, 1AH, 01H, 00H  
DFB 1EH, 1AH, 06H, 01H, 00H  
DFB 1EH, 1AH, 0CH, 01H, 00H  
DFB 1EH, 06H, 0FH, 01H, 00H  
DFB 1EH, 06H, 11H, 01H, 00H  
DFB 1EH, 06H, 12H, 01H, 00H  
DFB 1EH, 06H, 14H, 01H, 00H  
DFB 1EH, 06H, 15H, 01H, 00H  
DFB 1EH, 06H, 1AH, 01H, 00H  
DFB 14H, 00H, 00H, 01H, 00H  
DFB 1EH, 06H, 0CH, 01H, 00H  
DFB 1EH, 0CH, 0FH, 01H, 00H  
DFB 1EH, 0CH, 11H, 01H, 00H  
DFB 1EH, 0CH, 12H, 01H, 00H  
DFB 1EH, 0CH, 14H, 01H, 00H  
DFB 1EH, 0CH, 15H, 01H, 00H  
DFB 1EH, 0CH, 1AH, 01H, 00H  
DFB 1EH, 0CH, 06H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 1EH,0CH,0CH,01H,00H  
DFB 02H,0CH,0FH,01H,00H  
DFB 02H,0CH,11H,01H,00H  
DFB 02H,0CH,12H,01H,00H  
DFB 02H,0CH,14H,01H,00H  
DFB 02H,0CH,15H,01H,00H  
DFB 02H,0CH,1AH,01H,00H  
DFB 02H,0CH,06H,01H,00H  
DFB 02H,0CH,0CH,01H,00H  
DFB 01H,0CH,0FH,01H,00H  
DFB 01H,0CH,11H,01H,00H  
DFB 01H,0CH,12H,01H,00H  
DFB 01H,0CH,14H,01H,00H  
DFB 01H,0CH,15H,01H,00H  
DFB 01H,0CH,1AH,01H,00H  
DFB 01H,0CH,06H,01H,00H  
DFB 01H,0CH,0CH,01H,00H  
DFB 42H,0FH,00H,01H,00H  
DFB 42H,11H,00H,01H,00H  
DFB 42H,12H,00H,01H,00H  
DFB 42H,14H,00H,01H,00H  
DFB 42H,15H,00H,01H,00H  
DFB 42H,1AH,00H,01H,00H  
DFB 42H,06H,00H,01H,00H  
DFB 42H,0CH,00H,01H,00H  
DFB 3CH,0CH,0FH,01H,00H  
DFB 3CH,0CH,11H,01H,00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3CH, 0CH, 12H, 01H, 00H  
 DFB 3CH, 0CH, 14H, 01H, 00H  
 DFB 3CH, 0CH, 15H, 01H, 00H  
 DFB 3CH, 0CH, 1AH, 01H, 00H  
 DFB 3CH, 0CH, 06H, 01H, 00H  
 DFB 3CH, 0CH, 0CH, 01H, 00H  
 DFB 03H, 0FH, 00H, 01H, 00H  
 DFB 03H, 11H, 00H, 01H, 00H  
 DFB 03H, 12H, 00H, 01H, 00H  
 DFB 03H, 14H, 00H, 01H, 00H  
 DFB 03H, 15H, 00H, 01H, 00H  
 DFB 03H, 1AH, 00H, 01H, 00H  
 DFB 03H, 06H, 00H, 01H, 00H  
 DFB 03H, 0CH, 00H, 01H, 00H  
 DFB 43H, 0FH, 00H, 01H, 00H  
 DFB 43H, 11H, 00H, 01H, 00H  
 DFB 43H, 12H, 00H, 01H, 00H  
 DFB 43H, 14H, 00H, 01H, 00H  
 DFB 43H, 15H, 00H, 01H, 00H  
 DFB 43H, 1AH, 00H, 01H, 00H  
 DFB 43H, 06H, 00H, 01H, 00H  
 DFB 43H, 0CH, 00H, 01H, 00H  
 DFB 25H, 0FH, 00H, 01H, 00H  
 DFB 25H, 11H, 00H, 01H, 00H  
 DFB 25H, 12H, 00H, 01H, 00H  
 DFB 25H, 14H, 00H, 01H, 00H  
 DFB 25H, 15H, 00H, 01H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 25H, 1AH, 00H, 01H, 00H

DFB 25H, 06H, 00H, 01H, 00H

DFB 25H, 0CH, 00H, 01H, 00H

DFB 07H, 0FH, 00H, 01H, 00H

DFB 07H, 11H, 00H, 01H, 00H

DFB 07H, 12H, 00H, 01H, 00H

DFB 07H, 14H, 00H, 01H, 00H

DFB 07H, 15H, 00H, 01H, 00H

DFB 07H, 1AH, 00H, 01H, 00H

DFB 07H, 06H, 00H, 01H, 00H

DFB 07H, 0CH, 00H, 01H, 00H

DFB 2EH, 1DH, 00H, 01H, 00H

DFB 2BH, 10H, 00H, 01H, 00H

DFB 1CH, 1DH, 01H, 23H, 00H

DFB 1CH, 01H, 00H, 23H, 00H

DFB 05H, 1DH, 01H, 23H, 00H

DFB 2CH, 10H, 00H, 01H, 00H

DFB 02H, 0CH, 01H, 12H, 00H

DFB 3BH, 01H, 00H, 01H, 30H

DFB 2EH, 23H, 00H, 01H, 00H

DFB 2EH, 00H, 00H, 01H, 00H

DFB 1CH, 23H, 01H, 23H, 00H

DFB 00H, 00H, 00H, 00H, 00H

DFB 05H, 23H, 01H, 23H, 00H

DFB 05H, 01H, 00H, 23H, 00H

DFB 01H, 0CH, 01H, 12H, 00H

DFB 3BH, 01H, 00H, 01H, 38H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 2EH, 1CH, 00H, 01H, 00H  
 DFB 2BH, 13H, 00H, 01H, 00H  
 DFB 1CH, 1CH, 01H, 23H, 00H  
 DFB 28H, 02H, 0CH, 12H, 00H  
 DFB 05H, 1CH, 01H, 23H, 00H  
 DFB 2CH, 13H, 00H, 01H, 00H  
 DFB 42H, 01H, 00H, 12H, 00H  
 DFB 3BH, 01H, 00H, 01H, 90H  
 DFB 2EH, 11H, 00H, 01H, 00H  
 DFB 13H, 00H, 00H, 01H, 00H  
 DFB 1CH, 11H, 01H, 23H, 00H  
 DFB 16H, 0CH, 02H, 12H, 00H  
 DFB 05H, 11H, 01H, 23H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 3CH, 0CH, 01H, 12H, 00H  
 DFB 3BH, 01H, 00H, 01H, 98H  
 DFB 2EH, 20H, 00H, 01H, 00H  
 DFB 2BH, 16H, 00H, 01H, 00H  
 DFB 1CH, 20H, 01H, 23H, 00H  
 DFB 12H, 0BH, 16H, 01H, 00H  
 DFB 05H, 20H, 01H, 23H, 00H  
 DFB 2CH, 16H, 00H, 01H, 00H  
 DFB 03H, 01H, 00H, 12H, 00H  
 DFB 3BH, 01H, 00H, 01H, 0A0H  
 DFB 2EH, 1FH, 00H, 01H, 00H  
 DFB 1CH, 06H, 00H, 01H, 00H  
 DFB 1CH, 1FH, 01H, 23H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 12H, 13H, 16H, 01H, 00H  
 DFB 05H, 1FH, 01H, 23H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 43H, 01H, 00H, 12H, 00H  
 DFB 3BH, 01H, 00H, 01H, 0ASH  
 DFB 2EH, 1EH, 00H, 01H, 00H  
 DFB 2BH, 0DH, 00H, 01H, 00H  
 DFB 1CH, 1EH, 01H, 23H, 00H  
 DFB 0FH, 00H, 00H, 01H, 00H  
 DFB 05H, 1EH, 01H, 23H, 00H  
 DFB 2CH, 0DH, 00H, 01H, 00H  
 DFB 25H, 01H, 00H, 12H, 00H  
 DFB 3BH, 01H, 00H, 01H, 0BOH  
 DFB 2EH, 1BH, 00H, 01H, 00H  
 DFB 1EH, 22H, 16H, 01H, 00H  
 DFB 1CH, 1BH, 01H, 23H, 00H  
 DFB 11H, 00H, 00H, 01H, 00H  
 DFB 05H, 1BH, 01H, 23H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 07H, 01H, 00H, 12H, 00H  
 DFB 3BH, 01H, 00H, 01H, 0BSH

CB: . DFB 33H, 0FH, 00H, 02H, 00H  
 DFB 33H, 11H, 00H, 02H, 00H  
 DFB 33H, 12H, 00H, 02H, 00H  
 DFB 33H, 14H, 00H, 02H, 00H  
 DFB 33H, 15H, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 33H, 1AH, 00H, 02H, 00H  
DFB 33H, 06H, 00H, 02H, 00H  
DFB 33H, 0CH, 00H, 02H, 00H  
DFB 33H, 0FH, 00H, 02H, 00H  
DFB 33H, 11H, 00H, 02H, 00H  
DFB 33H, 12H, 00H, 02H, 00H  
DFB 33H, 14H, 00H, 02H, 00H  
DFB 33H, 15H, 00H, 02H, 00H  
DFB 33H, 1AH, 00H, 02H, 00H  
DFB 33H, 06H, 00H, 02H, 00H  
DFB 33H, 0CH, 00H, 02H, 00H  
DFB 31H, 0FH, 00H, 02H, 00H  
DFB 31H, 11H, 00H, 02H, 00H  
DFB 31H, 12H, 00H, 02H, 00H  
DFB 31H, 14H, 00H, 02H, 00H  
DFB 31H, 15H, 00H, 02H, 00H  
DFB 31H, 1AH, 00H, 02H, 00H  
DFB 31H, 06H, 00H, 02H, 00H  
DFB 31H, 0CH, 00H, 02H, 00H  
DFB 36H, 0FH, 00H, 02H, 00H  
DFB 36H, 11H, 00H, 02H, 00H  
DFB 36H, 12H, 00H, 02H, 00H  
DFB 36H, 14H, 00H, 02H, 00H  
DFB 36H, 15H, 00H, 02H, 00H  
DFB 36H, 1AH, 00H, 02H, 00H  
DFB 36H, 06H, 00H, 02H, 00H  
DFB 36H, 0CH, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3FH, 0FH, 00H, 02H, 00H  
 DFB 3FH, 11H, 00H, 02H, 00H  
 DFB 3FH, 12H, 00H, 02H, 00H  
 DFB 3FH, 14H, 00H, 02H, 00H  
 DFB 3FH, 15H, 00H, 02H, 00H  
 DFB 3FH, 1AH, 00H, 02H, 00H  
 DFB 3FH, 06H, 00H, 02H, 00H  
 DFB 3FH, 0CH, 00H, 02H, 00H  
 DFB 40H, 0FH, 00H, 02H, 00H  
 DFB 40H, 11H, 00H, 02H, 00H  
 DFB 40H, 12H, 00H, 02H, 00H  
 DFB 40H, 14H, 00H, 02H, 00H  
 DFB 40H, 15H, 00H, 02H, 00H  
 DFB 40H, 1AH, 00H, 02H, 00H  
 DFB 40H, 06H, 00H, 02H, 00H  
 DFB 40H, 0CH, 00H, 02H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 00H, 00H, 00H, 00H, 00H  
 DFB 41H, 0FH, 00H, 02H, 00H  
 DFB 41H, 11H, 00H, 02H, 00H  
 DFB 41H, 12H, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 41H, 14H, 00H, 02H, 00H  
DFB 41H, 15H, 00H, 02H, 00H  
DFB 41H, 1AH, 00H, 02H, 00H  
DFB 41H, 06H, 00H, 02H, 00H  
DFB 41H, 0CH, 00H, 02H, 00H  
DFB 04H, 01H, 0FH, 02H, 00H  
DFB 04H, 01H, 11H, 02H, 00H  
DFB 04H, 01H, 12H, 02H, 00H  
DFB 04H, 01H, 14H, 02H, 00H  
DFB 04H, 01H, 15H, 02H, 00H  
DFB 04H, 01H, 1AH, 02H, 00H  
DFB 04H, 01H, 06H, 02H, 00H  
DFB 04H, 01H, 0CH, 02H, 00H  
DFB 04H, 01H, 0FH, 02H, 01H  
DFB 04H, 01H, 11H, 02H, 01H  
DFB 04H, 01H, 12H, 02H, 01H  
DFB 04H, 01H, 14H, 02H, 01H  
DFB 04H, 01H, 15H, 02H, 01H  
DFB 04H, 01H, 1AH, 02H, 01H  
DFB 04H, 01H, 06H, 02H, 01H  
DFB 04H, 01H, 0CH, 02H, 01H  
DFB 04H, 01H, 0FH, 02H, 02H  
DFB 04H, 01H, 11H, 02H, 02H  
DFB 04H, 01H, 12H, 02H, 02H  
DFB 04H, 01H, 14H, 02H, 02H  
DFB 04H, 01H, 15H, 02H, 02H  
DFB 04H, 01H, 1AH, 02H, 02H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 04H,01H,06H,02H,S2H  
DFB 04H,01H,0CH,02H,S2H  
DFB 04H,01H,0FH,02H,S3H  
DFB 04H,01H,11H,02H,S3H  
DFB 04H,01H,12H,02H,S3H  
DFB 04H,01H,14H,02H,S3H  
DFB 04H,01H,15H,02H,S3H  
DFB 04H,01H,1AH,02H,S3H  
DFB 04H,01H,06H,02H,S3H  
DFB 04H,01H,0CH,02H,S3H  
DFB 04H,01H,0FH,02H,S4H  
DFB 04H,01H,11H,02H,S4H  
DFB 04H,01H,12H,02H,S4H  
DFB 04H,01H,14H,02H,S4H  
DFB 04H,01H,15H,02H,S4H  
DFB 04H,01H,1AH,02H,S4H  
DFB 04H,01H,06H,02H,S4H  
DFB 04H,01H,0CH,02H,S4H  
DFB 04H,01H,0FH,02H,S5H  
DFB 04H,01H,11H,02H,S5H  
DFB 04H,01H,12H,02H,S5H  
DFB 04H,01H,14H,02H,S5H  
DFB 04H,01H,15H,02H,S5H  
DFB 04H,01H,1AH,02H,S5H  
DFB 04H,01H,06H,02H,S5H  
DFB 04H,01H,0CH,02H,S5H  
DFB 04H,01H,0FH,02H,S6H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 04H,01H,11H,02H,86H  
DFB 04H,01H,12H,02H,86H  
DFB 04H,01H,14H,02H,86H  
DFB 04H,01H,15H,02H,86H  
DFB 04H,01H,1AH,02H,86H  
DFB 04H,01H,06H,02H,86H  
DFB 04H,01H,0CH,02H,86H  
DFB 04H,01H,0FH,02H,87H  
DFB 04H,01H,11H,02H,87H  
DFB 04H,01H,12H,02H,87H  
DFB 04H,01H,14H,02H,87H  
DFB 04H,01H,15H,02H,87H  
DFB 04H,01H,1AH,02H,87H  
DFB 04H,01H,06H,02H,87H  
DFB 04H,01H,0CH,02H,87H  
DFB 2DH,01H,0FH,02H,80H  
DFB 2DH,01H,11H,02H,80H  
DFB 2DH,01H,12H,02H,80H  
DFB 2DH,01H,14H,02H,80H  
DFB 2DH,01H,15H,02H,80H  
DFB 2DH,01H,1AH,02H,80H  
DFB 2DH,01H,06H,02H,80H  
DFB 2DH,01H,0CH,02H,80H  
DFB 2DH,01H,0FH,02H,81H  
DFB 2DH,01H,11H,02H,81H  
DFB 2DH,01H,12H,02H,81H  
DFB 2DH,01H,14H,02H,81H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 2DH, 01H, 15H, 02H, 81H  
DFB 2DH, 01H, 1AH, 02H, 81H  
DFB 2DH, 01H, 06H, 02H, 81H  
DFB 2DH, 01H, 0CH, 02H, 81H  
DFB 2DH, 01H, 0FH, 02H, 82H  
DFB 2DH, 01H, 11H, 02H, 82H  
DFB 2DH, 01H, 12H, 02H, 82H  
DFB 2DH, 01H, 14H, 02H, 82H  
DFB 2DH, 01H, 15H, 02H, 82H  
DFB 2DH, 01H, 1AH, 02H, 82H  
DFB 2DH, 01H, 06H, 02H, 82H  
DFB 2DH, 01H, 0CH, 02H, 82H  
DFB 2DH, 01H, 0FH, 02H, 83H  
DFB 2DH, 01H, 11H, 02H, 83H  
DFB 2DH, 01H, 12H, 02H, 83H  
DFB 2DH, 01H, 14H, 02H, 83H  
DFB 2DH, 01H, 15H, 02H, 83H  
DFB 2DH, 01H, 1AH, 02H, 83H  
DFB 2DH, 01H, 06H, 02H, 83H  
DFB 2DH, 01H, 0CH, 02H, 83H  
DFB 2DH, 01H, 0FH, 02H, 84H  
DFB 2DH, 01H, 11H, 02H, 84H  
DFB 2DH, 01H, 12H, 02H, 84H  
DFB 2DH, 01H, 14H, 02H, 84H  
DFB 2DH, 01H, 15H, 02H, 84H  
DFB 2DH, 01H, 1AH, 02H, 84H  
DFB 2DH, 01H, 06H, 02H, 84H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 2DH,01H,0CH,02H,34H  
DFB 2DH,01H,0FH,02H,35H  
DFB 2DH,01H,11H,02H,35H  
DFB 2DH,01H,12H,02H,35H  
DFB 2DH,01H,14H,02H,35H  
DFB 2DH,01H,15H,02H,35H  
DFB 2DH,01H,1AH,02H,35H  
DFB 2DH,01H,06H,02H,35H  
DFB 2DH,01H,0CH,02H,35H  
DFB 2DH,01H,0FH,02H,36H  
DFB 2DH,01H,11H,02H,36H  
DFB 2DH,01H,12H,02H,36H  
DFB 2DH,01H,14H,02H,36H  
DFB 2DH,01H,15H,02H,36H  
DFB 2DH,01H,1AH,02H,36H  
DFB 2DH,01H,06H,02H,36H  
DFB 2DH,01H,0CH,02H,36H  
DFB 2DH,01H,0FH,02H,37H  
DFB 2DH,01H,11H,02H,37H  
DFB 2DH,01H,12H,02H,37H  
DFB 2DH,01H,14H,02H,37H  
DFB 2DH,01H,15H,02H,37H  
DFB 2DH,01H,1AH,02H,37H  
DFB 2DH,01H,06H,02H,37H  
DFB 2DH,01H,0CH,02H,37H  
DFB 3EH,01H,0FH,02H,80H  
DFB 3EH,01H,11H,02H,80H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3EH,01H,12H,02H,80H  
 DFB 3EH,01H,14H,02H,80H  
 DFB 3EH,01H,15H,02H,80H  
 DFB 3EH,01H,1AH,02H,80H  
 DFB 3EH,01H,06H,02H,80H  
 DFB 3EH,01H,0CH,02H,80H  
 DFB 3EH,01H,0FH,02H,81H  
 DFB 3EH,01H,11H,02H,81H  
 DFB 3EH,01H,12H,02H,81H  
 DFB 3EH,01H,14H,02H,81H  
 DFB 3EH,01H,15H,02H,81H  
 DFB 3EH,01H,1AH,02H,81H  
 DFB 3EH,01H,06H,02H,81H  
 DFB 3EH,01H,0CH,02H,81H  
 DFB 3EH,01H,0FH,02H,82H  
 DFB 3EH,01H,11H,02H,82H  
 DFB 3EH,01H,12H,02H,82H  
 DFB 3EH,01H,14H,02H,82H  
 DFB 3EH,01H,15H,02H,82H  
 DFB 3EH,01H,1AH,02H,82H  
 DFB 3EH,01H,06H,02H,82H  
 DFB 3EH,01H,0CH,02H,82H  
 DFB 3EH,01H,0FH,02H,83H  
 DFB 3EH,01H,11H,02H,83H  
 DFB 3EH,01H,12H,02H,83H  
 DFB 3EH,01H,14H,02H,83H  
 DFB 3EH,01H,15H,02H,83H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3EH,01H,1AH,02H,83H  
DFB 3EH,01H,06H,02H,83H  
DFB 3EH,01H,0CH,02H,83H  
DFB 3EH,01H,0FH,02H,84H  
DFB 3EH,01H,11H,02H,84H  
DFB 3EH,01H,12H,02H,84H  
DFB 3EH,01H,14H,02H,84H  
DFB 3EH,01H,15H,02H,84H  
DFB 3EH,01H,1AH,02H,84H  
DFB 3EH,01H,06H,02H,84H  
DFB 3EH,01H,0CH,02H,84H  
DFB 3EH,01H,0FH,02H,85H  
DFB 3EH,01H,11H,02H,85H  
DFB 3EH,01H,12H,02H,85H  
DFB 3EH,01H,14H,02H,85H  
DFB 3EH,01H,15H,02H,85H  
DFB 3EH,01H,1AH,02H,85H  
DFB 3EH,01H,06H,02H,85H  
DFB 3EH,01H,0CH,02H,85H  
DFB 3EH,01H,0FH,02H,86H  
DFB 3EH,01H,11H,02H,86H  
DFB 3EH,01H,12H,02H,86H  
DFB 3EH,01H,14H,02H,86H  
DFB 3EH,01H,15H,02H,86H  
DFB 3EH,01H,1AH,02H,86H  
DFB 3EH,01H,06H,02H,86H  
DFB 3EH,01H,0CH,02H,86H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 3EH,01H,0FH,02H,37H

DFB 3EH,01H,11H,02H,37H

DFB 3EH,01H,12H,02H,37H

DFB 3EH,01H,14H,02H,37H

DFB 3EH,01H,15H,02H,37H

DFB 3EH,01H,1AH,02H,37H

DFB 3EH,01H,06H,02H,37H

DFB 3EH,01H,0CH,02H,37H

DD: DFB 09H,02H,18H,10H,02H,00H

DFB 19H,02H,18H,13H,02H,00H

DFB 21H,1EH,18H,01H,64H,00H

DFB 22H,1EH,02H,18H,64H,00H

DFB 23H,17H,18H,00H,02H,00H

DFB 29H,02H,18H,18H,02H,00H

DFB 2AH,1EH,18H,02H,64H,00H

DFB 2BH,0EH,18H,00H,02H,00H

DFB 34H,17H,08H,00H,43H,00H

DFB 35H,0EH,08H,00H,43H,00H

DFB 36H,1EH,08H,01H,54H,00H

DFB 39H,02H,18H,22H,02H,00H

DFB 46H,1EH,0FH,08H,43H,00H

DFB 4EH,1EH,11H,08H,43H,00H

DFB 56H,1EH,12H,08H,43H,00H

DFB 5EH,1EH,14H,08H,43H,00H

DFB 66H,1EH,15H,08H,43H,00H

DFB 6EH,1EH,1AH,08H,43H,00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 70H, 1EH, 0SH, 0FH, 43H, 00H  
DFB 71H, 1EH, 0SH, 11H, 43H, 00H  
DFB 72H, 1EH, 0SH, 12H, 43H, 00H  
DFB 73H, 1EH, 0SH, 14H, 43H, 00H  
DFB 74H, 1EH, 0SH, 15H, 43H, 00H  
DFB 75H, 1EH, 0SH, 1AH, 43H, 00H  
DFB 77H, 1EH, 0SH, 0CH, 43H, 00H  
DFB 7EH, 1EH, 0CH, 0SH, 43H, 00H  
DFB 86H, 02H, 0CH, 0SH, 43H, 00H  
DFB 8EH, 01H, 0CH, 0SH, 43H, 00H  
DFB 96H, 42H, 0SH, 00H, 43H, 00H  
DFB 9EH, 3CH, 0CH, 0SH, 43H, 00H  
DFB 0A6H, 03H, 0SH, 00H, 43H, 00H  
DFB 0AEH, 43H, 0SH, 00H, 43H, 00H  
DFB 0B6H, 25H, 0SH, 00H, 43H, 00H  
DFB 0BEH, 07H, 0SH, 00H, 43H, 00H  
DFB 0E1H, 2BH, 18H, 00H, 02H, 00H  
DFB 0E3H, 12H, 0BH, 18H, 02H, 00H  
DFB 0E5H, 2CH, 18H, 00H, 02H, 00H  
DFB 0E9H, 1CH, 07H, 00H, 02H, 00H  
DFB 0F9H, 1EH, 22H, 18H, 02H, 00H

ED: DFB 40H, 16H, 0FH, 04H, 02H, 00H  
DFB 41H, 28H, 04H, 0FH, 02H, 00H  
DFB 42H, 3CH, 16H, 10H, 02H, 00H  
DFB 43H, 1EH, 02H, 10H, 64H, 00H  
DFB 44H, 23H, 00H, 00H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 45H, 30H, 00H, 00H, 02H, 00H  
DFB 46H, 15H, 01H, 00H, 02H, 80H  
DFB 47H, 1EH, 17H, 0CH, 02H, 00H  
DFB 48H, 16H, 11H, 04H, 02H, 00H  
DFB 49H, 28H, 04H, 11H, 02H, 00H  
DFB 4AH, 01H, 16H, 10H, 02H, 00H  
DFB 4BH, 1EH, 10H, 02H, 64H, 00H  
DFB 4DH, 2FH, 00H, 00H, 02H, 00H  
DFB 4FH, 1EH, 21H, 0CH, 02H, 00H  
DFB 50H, 16H, 12H, 04H, 02H, 00H  
DFB 51H, 28H, 04H, 12H, 02H, 00H  
DFB 52H, 3CH, 16H, 13H, 02H, 00H  
DFB 53H, 1EH, 02H, 13H, 64H, 00H  
DFB 56H, 15H, 01H, 00H, 02H, 81H  
DFB 57H, 1EH, 0CH, 17H, 02H, 00H  
DFB 58H, 16H, 14H, 04H, 02H, 00H  
DFB 59H, 28H, 04H, 14H, 02H, 00H  
DFB 5AH, 01H, 16H, 13H, 02H, 00H  
DFB 5BH, 1EH, 13H, 02H, 64H, 00H  
DFB 5EH, 15H, 01H, 00H, 02H, 82H  
DFB 5FH, 1EH, 0CH, 21H, 02H, 00H  
DFB 60H, 16H, 15H, 04H, 02H, 00H  
DFB 61H, 28H, 04H, 15H, 02H, 00H  
DFB 62H, 3CH, 16H, 16H, 02H, 00H  
DFB 67H, 3AH, 00H, 00H, 02H, 00H  
DFB 68H, 16H, 1AH, 04H, 02H, 00H  
DFB 69H, 28H, 04H, 1AH, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 6AH, 01H, 16H, 16H, 02H, 00H  
DFB 6FH, 35H, 00H, 00H, 02H, 00H  
DFB 72H, 3CH, 16H, 22H, 02H, 00H  
DFB 73H, 1EH, 02H, 22H, 64H, 00H  
DFB 78H, 16H, 0CH, 04H, 02H, 00H  
DFB 79H, 28H, 04H, 0CH, 02H, 00H  
DFB 7AH, 01H, 16H, 22H, 02H, 00H  
DFB 7BH, 1EH, 22H, 02H, 64H, 00H  
DFB 0A0H, 21H, 00H, 00H, 02H, 00H  
DFB 0A1H, 0AH, 00H, 00H, 02H, 00H  
DFB 0A2H, 1AH, 00H, 00H, 02H, 00H  
DFB 0A3H, 2AH, 00H, 00H, 02H, 00H  
DFB 0A8H, 1FH, 00H, 00H, 02H, 00H  
DFB 0A9H, 08H, 00H, 00H, 02H, 00H  
DFB 0AAH, 18H, 00H, 00H, 02H, 00H  
DFB 0ABH, 29H, 00H, 00H, 02H, 00H  
DFB 0B0H, 22H, 00H, 00H, 02H, 00H  
DFB 0B1H, 0BH, 00H, 00H, 02H, 00H  
DFB 0B2H, 1BH, 00H, 00H, 02H, 00H  
DFB 0B3H, 27H, 00H, 00H, 02H, 00H  
DFB 0B8H, 20H, 00H, 00H, 02H, 00H  
DFB 0B9H, 09H, 00H, 00H, 02H, 00H  
DFB 0BAH, 19H, 00H, 00H, 02H, 00H  
DFB 0BBH, 26H, 00H, 00H, 02H, 00H

FD: DFB 09H, 02H, 19H, 10H, 02H, 00H

DFB 19H, 02H, 19H, 13H, 02H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 21H, 1EH, 19H, 01H, 64H, 00H  
DFB 22H, 1EH, 02H, 19H, 64H, 00H  
DFB 23H, 17H, 19H, 00H, 02H, 00H  
DFB 29H, 02H, 19H, 19H, 02H, 00H  
DFB 2AH, 1EH, 19H, 02H, 64H, 00H  
DFB 2BH, 0EH, 19H, 00H, 02H, 00H  
DFB 34H, 17H, 0AH, 00H, 43H, 00H  
DFB 35H, 0EH, 0AH, 00H, 43H, 00H  
DFB 36H, 1EH, 0AH, 01H, 54H, 00H  
DFB 39H, 02H, 19H, 22H, 02H, 00H  
DFB 46H, 1EH, 0FH, 0AH, 43H, 00H  
DFB 4EH, 1EH, 11H, 0AH, 43H, 00H  
DFB 5GH, 1EH, 12H, 0AH, 43H, 00H  
DFB 5FH, 1EH, 14H, 0AH, 43H, 00H  
DFB 66H, 1EH, 15H, 0AH, 43H, 00H  
DFB 6EH, 1EH, 1AH, 0AH, 43H, 00H  
DFB 70H, 1EH, 0AH, 0FH, 43H, 00H  
DFB 71H, 1EH, 0AH, 11H, 43H, 00H  
DFB 72H, 1EH, 0AH, 12H, 43H, 00H  
DFB 73H, 1EH, 0AH, 14H, 43H, 00H  
DFB 74H, 1EH, 0AH, 15H, 43H, 00H  
DFB 75H, 1EH, 0AH, 1AH, 43H, 00H  
DFB 77H, 1EH, 0AH, 0CH, 43H, 00H  
DFB 7EH, 1EH, 0CH, 0AH, 43H, 00H  
DFB 86H, 02H, 0CH, 0AH, 43H, 00H  
DFB 8EH, 01H, 0CH, 0AH, 43H, 00H  
DFB 96H, 42H, 0AH, 00H, 43H, 00H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 9EH, 3CH, 0C1H, 0AH, 43H, 00H  
DFB 0A6H, 03H, 0AH, 00H, 43H, 00H  
DFB 0AEH, 43H, 0AH, 00H, 43H, 00H  
DFB 0B6H, 25H, 0AH, 00H, 43H, 00H  
DFB 0BEH, 07H, 0AH, 00H, 43H, 00H  
DFB 0E1H, 2BH, 19H, 00H, 02H, 00H  
DFB 0E3H, 12H, 0BH, 19H, 02H, 00H  
DFB 0E5H, 2CH, 19H, 00H, 02H, 00H  
DFB 0E9H, 1CH, 09H, 00H, 02H, 00H  
DFB 0F9H, 1EH, 22H, 19H, 02H, 00H  
DDCB: DFB 06H, 33H, 0SH, 00H, 44H, 00H  
DFB 0EH, 3SH, 0SH, 00H, 44H, 00H  
DFB 16H, 31H, 0SH, 00H, 44H, 00H  
DFB 1FH, 36H, 0SH, 00H, 44H, 00H  
DFB 26H, 3FH, 0SH, 00H, 44H, 00H  
DFB 2EH, 40H, 0SH, 00H, 44H, 00H  
DFB 3EH, 41H, 0SH, 00H, 44H, 00H  
DFB 46H, 04H, 01H, 0SH, 44H, 80H  
DFB 4FH, 04H, 01H, 0SH, 44H, 81H  
DFB 56H, 04H, 01H, 0SH, 44H, 82H  
DFB 5FH, 04H, 01H, 0SH, 44H, 83H  
DFB 66H, 04H, 01H, 0SH, 44H, 84H  
DFB 6EH, 04H, 01H, 0SH, 44H, 85H  
DFB 76H, 04H, 01H, 0SH, 44H, 86H  
DFB 7EH, 04H, 01H, 0SH, 44H, 87H  
DFB 86H, 2DH, 01H, 0SH, 44H, 80H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 8EH, 2DH, 01H, 0SH, 44H, S1H  
 DFB 96H, 2DH, 01H, 0SH, 44H, S2H  
 DFB 9EH, 2DH, 01H, 0SH, 44H, S3H  
 DFB 0A6H, 2DH, 01H, 0SH, 44H, S4H  
 DFB 0AEH, 2DH, 01H, 0SH, 44H, S5H  
 DFB 0B6H, 2DH, 01H, 0SH, 44H, S6H  
 DFB 0BEH, 2DH, 01H, 0SH, 44H, S7H  
 DFB 0C6H, 3EH, 01H, 0SH, 44H, S0H  
 DFB 0CEH, 3EH, 01H, 0SH, 44H, S1H  
 DFB 0D6H, 3EH, 01H, 0SH, 44H, S2H  
 DFB 0DEH, 3EH, 01H, 0SH, 44H, S3H  
 DFB 0E6H, 3EH, 01H, 0SH, 44H, S4H  
 DFB 0EEH, 3EH, 01H, 0SH, 44H, S5H  
 DFB 0F6H, 3EH, 01H, 0SH, 44H, S6H  
 DFB 0FEH, 3EH, 01H, 0SH, 44H, S7H

FDCB: DFB 06H, 33H, 0AH, 00H, 44H, 00H  
 DFB 0EH, 3SH, 0AH, 00H, 44H, 00H  
 DFB 16H, 31H, 0AH, 00H, 44H, 00H  
 DFB 1EH, 36H, 0AH, 00H, 44H, 00H  
 DFB 26H, 3FH, 0AH, 00H, 44H, 00H  
 DFB 2EH, 40H, 0AH, 00H, 44H, 00H  
 DFB 3EH, 41H, 0AH, 00H, 44H, 00H  
 DFB 46H, 04H, 01H, 0AH, 44H, S0H  
 DFB 4EH, 04H, 01H, 0AH, 44H, S1H  
 DFB 56H, 04H, 01H, 0AH, 44H, S2H  
 DFB 5EH, 04H, 01H, 0AH, 44H, S3H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB 66H,04H,01H,0AH,44H,84H  
 DFB 6EH,04H,01H,0AH,44H,85H  
 DFB 76H,04H,01H,0AH,44H,86H  
 DFB 7EH,04H,01H,0AH,44H,87H  
 DFB 86H,2DH,01H,0AH,44H,80H  
 DFB 8EH,2DH,01H,0AH,44H,81H  
 DFB 96H,2DH,01H,0AH,44H,82H  
 DFB 9EH,2DH,01H,0AH,44H,83H  
 DFB 0A6H,2DH,01H,0AH,44H,84H  
 DFB 0AEH,2DH,01H,0AH,44H,85H  
 DFB 0B6H,2DH,01H,0AH,44H,86H  
 DFB 0BEH,2DH,01H,0AH,44H,87H  
 DFB 0C6H,3EH,01H,0AH,44H,80H  
 DFB 0CEH,3EH,01H,0AH,44H,81H  
 DFB 0D6H,3EH,01H,0AH,44H,82H  
 DFB 0DEH,3EH,01H,0AH,44H,83H  
 DFB 0E6H,3EH,01H,0AH,44H,84H  
 DFB 0EEH,3EH,01H,0AH,44H,85H  
 DFB 0F6H,3EH,01H,0AH,44H,86H  
 DFB 0FEH,3EH,01H,0AH,44H,87H

MNETAB: DFB "ADC "  
 DFB "ADD "  
 DFB "AND "  
 DFB "BIT "  
 DFB "CALL"  
 DFB "CCF "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB "CP "  
DFB "CPD "  
DFB "CPDR"  
DFB "CPI "  
DFB "CPIR"  
DFB "CPL "  
DFB "DAA "  
DFB "DEC "  
DFB "DI "  
DFB "DJNZ"  
DFB "EI "  
DFB "EX "  
DFB "EXX "  
DFB "HALT"  
DFB "IM "  
DFB "IN "  
DFB "INC "  
DFB "IND "  
DFB "INDR"  
DFB "INI "  
DFB "INIR"  
DFB "JP "  
DFB "JR "  
DFB "LD "  
DFB "LDD "  
DFB "LDDR"  
DFB "LDI "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

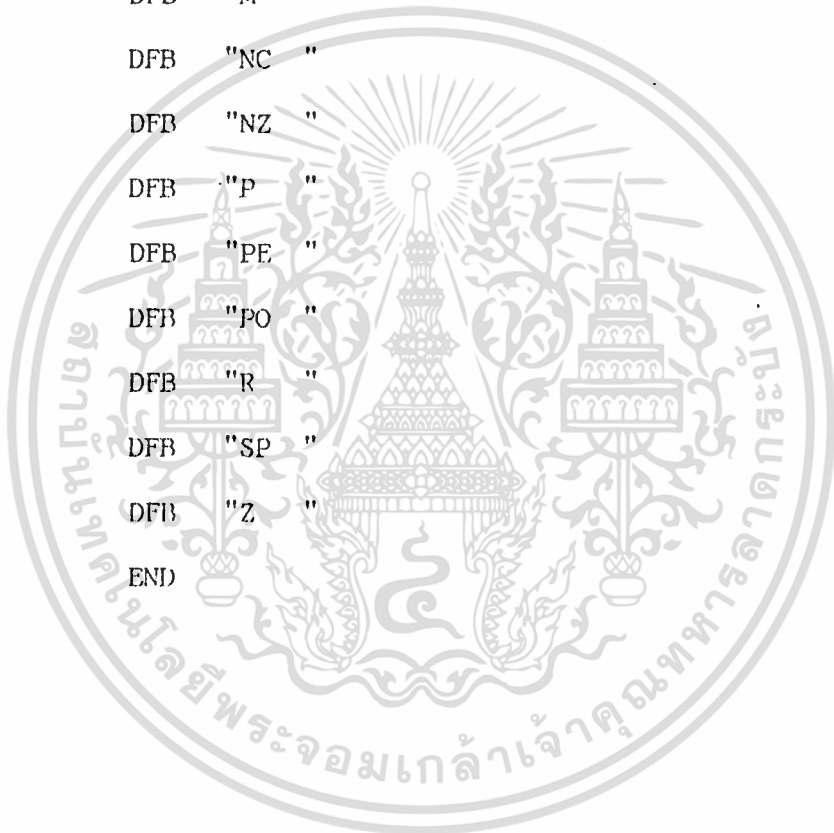
DFB "LDIR"  
DFB "NEG "  
DFB "NOP "  
DFB "OR "  
DFB "OTDR"  
DFB "OTIR"  
DFB "OUT "  
DFB "OUTD"  
DFB "OUTI"  
DFB "POP "  
DFB "PUSH"  
DFB "RES "  
DFB "RET "  
DFB "RJETI"  
DFB "RJETN"  
DFB "RL "  
DFB "RLA "  
DFB "RLC "  
DFB "RLCA"  
DFB "RLD "  
DFB "RR "  
DFB "RRA "  
DFB "RRC "  
DFB "RRCA"  
DFB "RRD "  
DFB "RST "  
DFB "SBC "



DFB "SCF "  
 DFB "SET "  
 DFB "SLA "  
 DFB "SRA "  
 DFB "SRL "  
 DFB "SUB "  
 DFB "XOR "  
 OPRTAB: DFB "# "  
 DFB "(#) "  
 DFB "(BC) "  
 DFB "(C) "  
 DFB "(DE) "  
 DFB "(HL) "  
 DFB "(IX) "  
 DFB "(IX+ "  
 DFB "(IY) "  
 DFB "(IY+ "  
 DFB "(SP) "  
 DFB "A "  
 DFB "AF "  
 DFB "AF' "  
 DFB "B "  
 DFB "BC "  
 DFB "C "  
 DFB "D "  
 DFB "DE "  
 DFB "E "

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DFB "H "  
DFB "HL "  
DFB "I "  
DFB "IX "  
DFB "IY "  
DFB "L "  
DFB "M "  
DFB "NC "  
DFB "NZ "  
DFB "P "  
DFB "PE "  
DFB "PO "  
DFB "R "  
DFB "SP "  
DFB "Z "  
END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. "RTC V.4 USER'S MANUAL" บริษัท อีทีที จำกัด
2. "DOT MATRIX LCD MODULE" บริษัท อีทีที จำกัด
3. "คู่มือไอซีไมโครโปรเซสเซอร์และไอซีที่เกี่ยวข้อง" บริษัท ซีเอ็ดดูเคชั่น จำกัด ,2536,เลขเอนการพิมพ์  
666 หน้า.
4. "คู่มือไอซีไมโครโปรเซสเซอร์ Z80180" ETT CO., LTD
5. "Z80180 MPU USER MANUAL" ETT CO., LTD
6. "ET- BOARD V.4 USER'S MANUAL" ETT CO., LTD
7. RODNEY ZAKS , "PROGRAMING THE Z80", SYBEX , THIRD EDITION.
8. LANCE A. LEVENTHAL AND WINTHROP SAVILLE , "Z80 ASSEMBLY  
LANGUAGE SUBROUTINES".
9. "MPF-1 MONITOR PROGRAM SOURCE LISTING" ,ACER INCORPORATED.  
PRELIMINARY TECHNICAL MANUAL ,ZILOG ,1988.
10. "CP-JR180 Z180 MICROPROCESSOR" ,ETT CO., LTD.