



แฟงแสดงผลแบบ 3 สี

3 COLORS LED DISPLAY BOARD



โดย

นายชัยวัฒน์ ชัชวาลย์

นายสุรนาท กิ่งไทร

นายสุรัตน์ ศรีเมธาวดี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2537

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แผลงแสดงผลแบบ 3 ดี

ผู้จัดทำ

นายชัยวัฒน์ ชัชวาลย์

นายสุรนาท กิ่งไทร

นายสุรัตน์ ศรีเมธาวดี



(ดร. กิติพล ชิตสกุล)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แผงแสดงผลแบบ 3 ที

นายชัยวัฒน์ ชัชวาลย์  
นายสุรนาท กิ่งไทร  
นายสุรัตน์ ศรีเมธาวดี

ดร. กิตติพล ชิตสกุล อาจารย์ที่ปรึกษา  
ปีการศึกษา 2537

## บทคัดย่อ

โครงการแผงแสดงผลแบบ 3 ที เป็นการนำเสนอรูปแบบสื่อของการโฆษณาอีกรูปแบบหนึ่ง ซึ่งสามารถนำเสนอข้อมูลได้ทั้งตัวอักษรภาษาไทยและภาษาอังกฤษ ตลอดจนรูปภาพกราฟฟิกต่าง ๆ ลักษณะของการแสดงผลสามารถแสดงผลได้ 3 ที คือ สีแดง สีเขียว และสีส้ม โดยทำการแสดงผลบนแผง Matrix LED ที่มีขนาด 16X96 ดวง โดยใช้ CPU เบอร์ Z80180 เป็นอุปกรณ์ควบคุมการทำงานทั้งหมด สามารถส่งข้อมูลจากเครื่อง IBM PC ลงในหน่วยความจำของแผงควบคุมให้สามารถทำการ RUN เพื่อแสดงข้อความต่าง ๆ ได้

ส่วนของซอฟต์แวร์ จะมีการคำนวณการสแกนแผงแสดงผลด้วยค่าเวลาคงที่, การจัดการระบบตัวอักษร, การเลื่อนภาพบนจอแสดงผล, การรับข้อมูลทางพอร์ทอนุกรมกับหน่วยความจำ, และการมีระบบเก็บรักษาข้อมูล

### 3 COLOR LED DISPLAY BOARD

Mr. Chaiwat Chatchawan

Mr. Suranart Kingsai

Mr. Surat Sriramaethawadee

Dr. Kitipol Chitsakul

Adviser 1994

#### ABSTRACT

LED display color board is an exhibit in form of advertising. It can be presented in Thai and English language and also graphic pictures in three colors , red , green , and orange. The board is construction by LED size in matrix form of 16X96 dots. A Z80180 CPU is used for control system and it can be transmitted/received data by IBM PC to/from memory of control board to change the display patterns.

The supporting software determines the period of display scanning , processing of character , directions of moving patterns and controlling of transmitting and receiving data via a serial port.

## กิตติกรรมประกาศ

ขอกราบขอบพระคุณ อาจารย์ กิติพล ชิตสกุล อาจารย์ภาควิชาอิเล็กทรอนิกส์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการนี้ ผู้ให้ความช่วยเหลือและคำปรึกษา ตลอดจนคำแนะนำที่ทำให้โครงการนี้สำเร็จได้ด้วยดี

ขอขอบพระคุณคณาจารย์และเจ้าหน้าที่ภาควิชาสถิติประยุกต์ คณะวิทยาศาสตร์ ทุกท่านที่เอื้อเพื่อสถานที่และอุปกรณ์คอมพิวเตอร์

ขอขอบคุณ คุณนิรันดร์ ไชยมภู นักศึกษาภาควิชาคอมพิวเตอร์ที่เอื้อเพื่อ Power Supply ในการทดลองทำโครงการ

ขอขอบคุณเพื่อน ๆ และน้อง ๆ ทุกคนที่ทำให้กำลังใจ คอยติดตามความก้าวหน้าของโครงการและให้ความช่วยเหลือเสมอมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย .....	ก
บทคัดย่อภาษาอังกฤษ .....	ข
กิตติกรรมประกาศ .....	ค
บทที่ 1 บทนำ .....	1
บทที่ 2 สถาปัตยกรรม Z80180 .....	2
ลักษณะโดยทั่วไป .....	4
ขาต่าง ๆ ของ Z80180 .....	5
Multiplex Pin Descriptions .....	8
Internal I/O Registers .....	9
Interrupts .....	12
การโปรแกรมตัวตั้งเวลา (Programmable Reload Timer : PRT) .....	18
บทที่ 3 หลักการทำงานของวงจร .....	22
อธิบายบล็อกไดอะแกรม .....	22
การทำงานของวงจร Scan .....	25
การทำงานของภาค Decode Port Latch DATA .....	27
การออกแบบวงจร Drive LED .....	29
บทที่ 4 การทดลองและผลการทดลอง .....	32
การทดลองภาค Scan .....	32
การทดลองภาค Decode Port Latch DATA .....	33
การทดลองวงจรภาค Drive LED .....	34
บทที่ 5 บทสรุปและวิจารณ์ผล .....	35
ภาคผนวก ก. รูปถ่ายชิ้นงาน แผงแสดงผลแบบ 3 สี .....	36
ภาคผนวก ข. Source Code .....	39
บรรณานุกรม .....	148

# บทที่ 1

## บทนำ

ในปัจจุบัน มีสื่อโฆษณาชนิดหนึ่งที่กำลังได้รับความนิยมกันอย่างแพร่หลาย สามารถพบเห็นได้ตามย่านธุรกิจ , ศูนย์การค้า , โรงงาน , โรงพยาบาล , หน่วยราชการบางแห่ง นั่นคือ แผงแสดงผลแบบ LED 3 สี ( 3 Color LED Display Board ) ซึ่งการทำงานของมันนั้นจะแสดงข้อความหรือรูปภาพตามที่กำหนด แล้วให้ข้อความนั้นเคลื่อนที่ไป เช่น ซ้ายไปขวา, ขวาไปซ้าย, บนลงล่าง, เลื่อนเข้าหากันตรงกลาง, แยกจากกัน, กระพริบ เป็นต้น

จากจุดประสงค์และการทำงานของแผงแสดงผลแบบ LED 3 สี เราจึงออกแบบโครงงานนี้ขึ้นมา โดยแบ่งออกเป็น 2 ส่วน คือ ฮาร์ดแวร์และซอฟต์แวร์ ส่วนของฮาร์ดแวร์ประกอบด้วย แผง LED ขนาด 1536 ดวง ( สูง X กว้าง = 16X96 ), ภาควิทยาศาสตร์ Decode port, ภาควิทยาศาสตร์ควบคุม ( CP-JR180 ), ภาควิทยาศาสตร์จ่ายไฟ หัวใจหลักของฮาร์ดแวร์ คือ ภาควิทยาศาสตร์ควบคุม ( CP-JR180 ) ซึ่งใช้ CPU เบอร์ Z80180 เหตุที่ใช้ CPU ตัวนี้เพราะว่ามีความเข้ากันได้กับ Z80 ( Code Compatible with Zilog Z80 CPU ) แต่มีความเร็วสูงกว่า, สามารถติดต่อกับหน่วยความจำภายนอกได้ถึง 1 Megabyte, ใช้พลังงานต่ำเนื่องจากทำด้วย CMOS VLSI และอื่น ๆ อีกมาก ทำให้สะดวกในการใช้งานและการเขียนโปรแกรมสนับสนุนการทำงาน ส่วนที่สองคือ ซอฟต์แวร์ ซึ่งเป็นภาษา Assembly ของ Z180 ซึ่งส่วนประกอบของโปรแกรมประกอบด้วย การ Latch ค้างข้อมูลทาง Column และการสแกนจอแสดงผลทาง Row ด้วยค่าเวลาที่ เหมือนกับการสแกนจอภาพโทรทัศน์, การจัดการเกี่ยวกับตัวอักษร ใช้รหัสประจำตัวอักษรเพื่อกำหนดความกว้าง - ความสูงของกลุ่มตัวอักษรที่มีขนาดเท่ากัน, การย่อขยายขนาดตัวอักษรบน Font เพียงชุดเดียว, การเลื่อนภาพบนจอแสดงผลจะจัดการกับข้อมูลในหน่วยความจำที่เรียกว่า Display Buffer เท่านั้น โดยนำข้อความถัดไปมาเก็บไว้ใน Display Buffer สำรอง ก่อนที่จะนำมาแทนที่ Display Buffer หลัก การแทนที่ของข้อมูลในแบบต่างๆ ก็จะได้การเลื่อนข้อความในรูปแบบทิศทางต่างๆกัน, การรับส่งข้อมูลทางพอร์ตอนุกรม พร้อมระบบตรวจสอบความผิดพลาด ( Check Sum ), การมีระบบเก็บรักษาข้อมูล ซึ่งทั้งหมดนี้เป็นส่วนประกอบหลักของ ซอฟต์แวร์ ที่เขียนขึ้นในโครงงานนี้

## บทที่ 2

### สถาปัตยกรรม Z80180

Z80180 รวมเอาคุณสมบัติหลัก ๆ ของ CPU ในระบบต่าง ๆ และทรัพยากร input/output ซึ่งมีประโยชน์ในการประยุกต์ใช้งานอย่างกว้างขวาง แกนหลักของ CPU ประกอบด้วยบล็อกการทำงาน 5 บล็อก ดังนี้  
ผลิตสัญญาณนาฬิกา (clock generator) , ควบคุมสัญญาณ BUS (bus state controller) ร่วมด้วยสัญญาณ refresh Dynamic RAM , ควบคุมการอินเทอร์รัพท์ (interrupt controller) , หน่วยจัดการหน่วยความจำ (Memory management unit, MMU) , และหน่วยประมวลผลกลาง (CPU) ทรัพยากร input/output ประกอบด้วยบล็อกการทำงานดังนี้ DMA 2 ช่อง , PRT 2 ช่อง , และช่องสัญญาณอนุกรมคล็อกอีก 1 ช่อง

ผลิตสัญญาณนาฬิกา (Clock generator) สัญญาณนาฬิกาของระบบได้มาจากคริสตอลภายนอกหรือ Clock ที่ป้อนเป็นอินพุตให้กับ CPU สัญญาณนาฬิกาภายนอกจะถูกหารด้วย 2 เป็นฐานเวลาให้กับอุปกรณ์ภายในและภายนอก CPU

ควบคุมสัญญาณ BUS (Bus state controller) เป็นสัญญาณสถานะทั้งหมดของระบบและควบคุมการทำงานร่วมกันทั้ง CPU และ internal I/O ที่ร่วมด้วย wait state timing , วงรอบการ Reset , DRAM refresh , และการทำ DMA

ควบคุมการอินเทอร์รัพท์ (Interrupt controller) ทำหน้าที่ตอบรับและจัดลำดับความสำคัญของการขออินเทอร์รัพท์ภายในและภายนอกแบบต่าง ๆ เพื่อจะได้ผลการตอบสนองที่ถูกต้องจาก CPU อินเทอร์รัพท์ทั้ง 3 แบบที่มีใน Z80 ยังคงมีใน Z80180 เพื่อที่จะให้คอมแพททิเบิลกับ Z80

หน่วยจัดการหน่วยความจำ (Memory Management Unit) MMU อนุญาตให้ผู้ใช้สามารถ Map หน่วยความจำขนาด 64 KB ไปอยู่ที่ใดก็ได้ใน 1 MB ทำให้คำสั่งต่าง ๆ ของ Z180 สามารถใช้กับหน่วยความจำที่เพิ่มขึ้นได้โดยใช้การแบ่งหน่วยความจำแบบ Common area - Bank area.

หน่วยประมวลผลกลาง (Central Processing Unit) คำสั่งต่าง ๆ ของ Z80 สามารถใช้ได้กับ CPU Z80180 ทุกคำสั่ง พร้อมทั้งยังเพิ่มคำสั่งการคูณ , หาร แบบ 8 บิต อีกด้วย ทำให้การคำนวณแบบคูณ.หารใช้เวลา น้อยลง

ควบคุมการ DMA (DMA Controller) หน่วยควบคุม DMA ทำหน้าที่ถ่ายข้อมูลระหว่างหน่วยความจำกับอุปกรณ์อื่น ๆ ด้วยความเร็วสูง การถ่ายข้อมูลสามารถกระทำได้ระหว่างหน่วยความจำกับหน่วยความจำ , หน่วยความจำกับอุปกรณ์ I/O , อุปกรณ์ I/O กับ อุปกรณ์ I/O Mode ของการถ่ายข้อมูลมี burst Mode และ steal Mode การถ่ายข้อมูลทำได้ในขอบเขต 1 MB บล็อกละ 64 KB

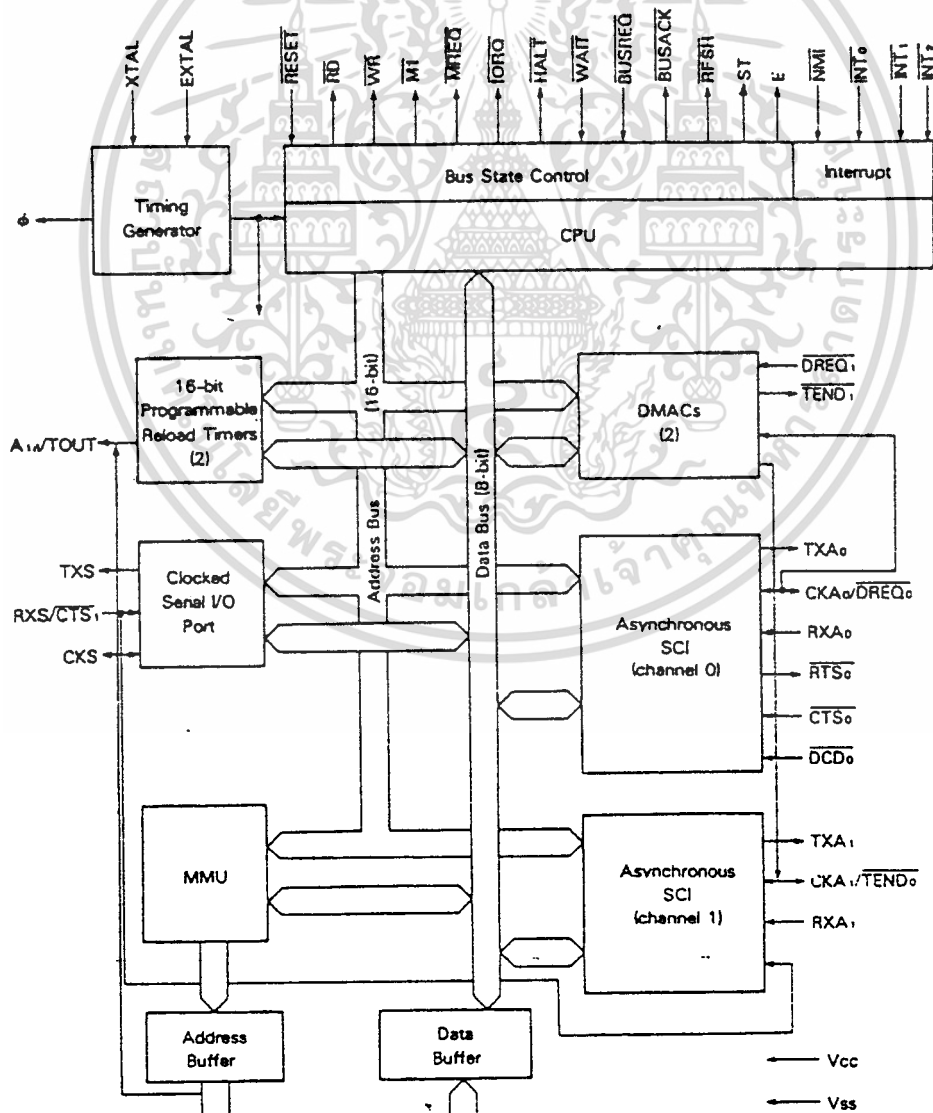
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อสื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส (Asynchronous Serial Communication Interface) มีช่อง ASCI แบบ full-duplex UARTs 2 ช่องแยกจากกัน แต่ละช่องประกอบด้วย programmable baud rate generator และสัญญาณควบคุมโมเด็ม ช่อง ASCI นี้ สนับสนุนการทำงานแบบ Multiprocessor ด้วย

ตัวตั้งเวลาแบบโปรแกรมได้ [Programmable Reload Timer (PRT)] มี 2 ช่อง แต่ละช่องประกอบด้วย counter 16 bit (timer) และรีจิสเตอร์โหนดค่าเวลา ค่าฐานเวลาที่ป้อนให้กับ counter คำนวณจากสัญญาณนาฬิกาของระบบหารด้วย 20

Clock Serial I/O (CSIO) ทำหน้าที่ส่งและรับข้อมูลอนุกรมแบบ half-duplex ช่องสัญญาณนี้ใช้สำหรับติดต่อข้อมูลด้วยความเร็วสูงกับไมโครโปรเซสเซอร์ หรือกับไมโครคอมพิวเตอร์เครื่องอื่น

บล็อกไดอะแกรมภายใน CPU Z80180



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คุณลักษณะ Z80180 หรือ Z180 MPU

- ทำงานที่ความถี่ 10 MHz
- หน่วยจัดการหน่วยความจำอยู่บนชิป เพื่อขยายหน่วยความจำ
- ช่อง DMA 2 ช่อง
- Wait State Gen บนชิป
- ช่อง UARTs 2 ช่อง
- ช่อง Timer 16 bit 2 ช่อง
- Interrupt Controller บนชิป
- Clock OSC/GEN บนชิป
- พอร์ต อนุกรม
- คำสั่ง Compat กับ Z80 CPU
- มีคำสั่งเพิ่มเติมจาก Z80

### ลักษณะโดยทั่วไป

ตั้งอยู่บนพื้นฐานของหน่วยประมวลผลทั่วไป ใช้เทคโนโลยีการผลิตแบบ CMOS คือไมโครโปรเซสเซอร์ชนิด 8 บิต ซึ่งมีประโยชน์ในการลดต้นทุนของระบบ และทำงานโดยใช้กำลังงานต่ำ ขณะที่ให้คุณภาพที่สูงกว่าและยังเขียนซอฟต์แวร์เข้ากันได้กับ Zilog Z80 CPU

คุณภาพที่เหนือกว่าก็เนื่องจากการทำงานที่ความถี่สูงกว่า ทำให้ลดค่าเวลาในการปฏิบัติคำสั่งลง , รวมทั้งยังมีชุดคำสั่งที่เพิ่มขึ้นอีก และมีหน่วยจัดการหน่วยความจำ (MMU) พร้อมทั้งติดต่อกับหน่วยความจำได้สูงถึง 1 MB

การลดราคาของระบบลงก็เนื่องจากการรวมเข้าไว้ด้วยกันของ Key function ต่าง ๆ อยู่ใน CPU , Key function เหล่านี้ประกอบด้วยอุปกรณ์ทาง I/O เช่น DMA, UART และ Timer channel รวมทั้ง Glue function เช่น Dynamic RAM refresh control , Wait state generator , Clock Oscillator , และ Interrupt controller

ไม่เพียงแต่ Z80180 จะใช้กำลังงานในการทำงานภาวะปกติเท่านั้น แต่มันยังกำหนดโหมดการทำงานที่ออกแบบมาเพื่อลดกำลังงานสิ้นเปลือง มี 2 โหมด คือ

- Sleep Mode ลดกำลังงานโดยให้ CPU เข้าสู่สถานะ " Stopped " เพื่อให้ใช้กระแสน้อยลง ขณะที่ on-chip I/O device ยังทำงานอยู่

- System Stop Mode คือทั้ง CPU และ on-chip peripherals เข้าสู่สถานะ " Stopped " ดังนั้นจะลดกำลังงานสิ้นเปลืองได้ดียิ่งขึ้นทีเดียว

เมื่อต่อร่วมกับ memory และอุปกรณ์อื่น ๆ ที่ใช้เทคโนโลยีการผลิตประเภท CMOS VLSI Z80180 จะให้ผลตอบแทนที่ดีเลิศกับระบบที่ต้องการคุณภาพสูงและใช้กำลังงานต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ขาต่าง ๆ ของ Z180

- A0 - A19** Address Bus (Output, active High, 3 state) คือขา Address ขนาด 20 บิตซึ่งจะกำหนดตำแหน่งของหน่วยความจำเพื่อแลกเปลี่ยนข้อมูล ขนาดหน่วยความจำสูงสุดเท่ากับ 1 MB และขนาด I/O Port เท่ากับ 64K Address bus จะมีสถานะเป็น High Impedance เมื่อรีเซ็ตและเมื่ออุปกรณ์ภายนอกต้องการขอใช้ Bus A18 จะมีลติเฟล็กซ์กับสัญญาณเอาต์พุตของ PRT Channel 1 (Programmable Reload Timer) หรือ TOUT ในสถานะรีเซ็ตขา A18 จะเป็น Address Bus และ A19 จะมีเฉพาะ Z80180 รุ่น PLCC
- $\overline{\text{BUSACK}}$**  Bus Acknowledge (Output, active Low) ตอบรับถึงความพร้อมที่จะยก Bus ต่าง ๆ ให้กับอุปกรณ์อื่น ๆ Bus เหล่านี้เช่น Address Bus, DATA Bus สัญญาณควบคุมบางขาจะมีสถานะเป็น High Impedance
- $\overline{\text{BUSREQ}}$**  Bus Request (Input, active Low) ขานี้ถูกใช้โดยอุปกรณ์ภายนอก (เช่น DMA Controller) เมื่อต้องการขอใช้ Bus ของระบบ ความต้องการนี้มีความสำคัญสูงกว่า  $\overline{\text{NMI}}$  และ CPU จะคอยตรวจสอบสัญญาณนี้ที่ machine cycle สุดท้ายของ cycle ปัจจุบันเสมอ สัญญาณนี้จะหยุด CPU จากการ execute คำสั่ง และให้ Address Bus, DATA Bus และสัญญาณควบคุมอื่น ๆ เข้าสู่สถานะ High Impedance
- $\text{CKA}_0, \text{CKA}_1$**  Asynchronous Clock 0 and 1 (Bidirectional, active High) ขาสองขานี้จะส่งและรับสัญญาณนาฬิกาสำหรับ Synchronous Channels  $\text{CKA}_0$  มัลติเฟล็กซ์กับ  $\overline{\text{DREQ}_0}$  และ  $\text{CKA}_1$  มัลติเฟล็กซ์กับ  $\overline{\text{TEND}_0}$
- CKS** Serial Clock (Bidirectional, active High) เป็น clock สำหรับ CSIO Channel
- CLOCK** System Clock (Output, active High) เป็นเอาต์พุต clock ใช้สำหรับอ้างอิง (Reference clock) ให้กับ MPU และระบบภายนอก ความถี่ขาออกนี้เท่ากับ  $\frac{1}{2}$  ของคริสตอลหรือ input clock
- $\overline{\text{CTS}_0} - \overline{\text{CTS}_1}$**  Clear to Send 0 and 1 (Input, active Low) ขาเหล่านี้คือสัญญาณควบคุมโมเด็มสำหรับ ASCII Channel  $\overline{\text{CTS}_1}$  ถูกมัลติเฟล็กซ์กับ RXS
- D0 - D7** DATA Bus (Bidirectional, active High, 3-state) D0-D7 ประกอบขึ้นเป็น 8 บิต บัส 2 ทิศทาง ใช้สำหรับเป็นทางเดินของข้อมูลจาก I/O และ Memory DATA Bus จะเข้าสู่สถานะ High Impedance ในช่วงรีเซ็ตและช่วงตอบรับการขอใช้ Bus จากอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{\text{DCD0}}$	DATA Carrier Detect 0 (Input, active Low) เป็นขา programmable modem control signal สำหรับ ASCII Channel 0
$\overline{\text{DREQ0}}, \overline{\text{DREQ1}}$	DMA Request 0 and 1 (Input, active Low) $\overline{\text{DREQ}}$ ถูกใช้สำหรับร้องขอการทำ DMA ที่อยู่บนชิปทั้งหมด 2 ช่อง DMA Channel Monitor ใช้ขาเหล่านี้เพื่อบ่งบอกเมื่ออุปกรณ์ภายนอกพร้อมที่จะอ่านหรือเขียนข้อมูล ขา Input เหล่านี้สามารถโปรแกรมให้เป็น Level 0 หรือ 1 หรือที่ขอบสัญญาณก็ได้ $\overline{\text{DREQ0}}$ ถูกมัลติเพล็กซ์กับ $\text{CKA}_0$
E	Enable Clock (Output, active High) เป็นขา clock ที่ทำงานเข้าจังหวะกับ Machine cycle ระหว่างการจัดการเกี่ยวกับ Bus
EXTAL	External Clock/Crystal (Input, active High) เป็นจุดต่อคริสตอล หรือ clock ภายนอกสามารถป้อนเข้าสู่ Z80180 ที่ขานี้เมื่อเราไม่ใช้คริสตอล ขานี้มีลักษณะเป็น Schmitt Trigger
$\overline{\text{HALT}}$	Halt/Sleep Status (Output, active Low) ขานี้จะ active หลังจาก CPU ปฏิบัติคำสั่ง HALT หรือ SLP ซึ่งจะตรวจสอบการ non-maskable หรือ maskable interrupt ก่อนทำคำสั่งนี้เสมอ และยังใช้ร่วมกับ $\overline{\text{MI}}$ และ ST เพื่อ Decode สถานะการทำงานของ CPU
$\overline{\text{INT0}}$	Maskable Interrupt Request 0 (input, active Low) สัญญาณนี้กำเนิดมาจากอุปกรณ์ภายนอก CPU จะตรวจสอบการอินเทอร์รัพท์ที่จุดสิ้นสุดของการปฏิบัติคำสั่งปัจจุบัน ขณะที่ $\overline{\text{NMI}}$ และ $\overline{\text{BUSREQ}}$ ไม่ active เมื่อ CPU รับรู้แล้วจะเข้าสู่วงรอบของการอินเทอร์รัพท์ ช่วงไซเคิลนี้ทั้ง $\overline{\text{MI}}$ และ $\overline{\text{IORQ}}$ จะไม่ active
$\overline{\text{INT1}}, \overline{\text{INT2}}$	Maskable Interrupt Request 1 and 2 (Input, active Low) สัญญาณนี้กำเนิดมาจากอุปกรณ์ภายนอก CPU จะตรวจสอบการอินเทอร์รัพท์ที่จุดสิ้นสุดของการปฏิบัติคำสั่งปัจจุบัน ขณะที่ $\overline{\text{NMI}}$ , $\overline{\text{BUSREQ}}$ และ $\overline{\text{INT0}}$ ไม่ active เมื่อ CPU รับรู้แล้วจะเข้าสู่วงรอบของการตอบรับการอินเทอร์รัพท์ซึ่งจะไม่เหมือนกับการตอบรับของ $\overline{\text{INT0}}$ ช่วงไซเคิลนี้ทั้ง $\overline{\text{MI}}$ และ $\overline{\text{IORQ}}$ จะไม่ active
$\overline{\text{IORQ}}$	I/O Request (Output, active Low, 3-state) $\overline{\text{IORQ}}$ เป็นสัญญาณบ่งบอกว่าขณะนี้สาย Address Bus แสดงตำแหน่งของอุปกรณ์ I/O เพื่อที่จะอ่านหรือเขียนกับอุปกรณ์เหล่านั้น $\overline{\text{IORQ}}$ จะ active พร้อมกับ $\overline{\text{MI}}$ ระหว่างการตอบรับการอินเทอร์รัพท์ $\overline{\text{INT0}}$ ซึ่งจะต้องป้อนข้อมูล interrupt vector เข้าที่ DATA Bus

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- $\overline{M1}$**  Machine Cycle 1 (Output, active Low) ใช้ร่วมกับ  $\overline{MREQ}$  เพื่อบ่งบอกช่วงจังหวะการ fetch ข้อมูลจากหน่วยความจำ, ใช้ร่วมกับ  $\overline{IORQ}$  เพื่อบ่งบอกสภาวะการตอบรับการอินเทอร์รัพท์, ใช้ประกอบกับสัญญาณ  $\overline{HALT}$  และ ST เพื่อ decode สถานะของ CPU
- $\overline{MREQ}$**  Memory Request (Output, active Low, 3-state)  $\overline{MREQ}$  บ่งบอกให้รู้ว่าข้อมูลใน Address Bus นั้นเป็นตำแหน่งของหน่วยความจำที่จะอ่านหรือเขียนข้อมูลลงไป
- $\overline{NMI}$**  Non-Maskable Interrupt (Input, Negative edge trigger)  $\overline{NMI}$  มีความสำคัญสูงกว่า  $\overline{INT}$  และจะตรวจสอบ  $\overline{NMI}$  ทุก ๆ state สุดท้ายของแต่ละคำสั่ง และไม่คำนึงถึงค่าใน Interrupt Enable flip-flop เมื่อมีสัญญาณนี้ CPU จะไปเริ่มปฏิบัติคำสั่งที่ Address 0066H
- $\overline{RD}$**  Read (Output, active Low, 3-state)  $\overline{RD}$  เป็นขาที่บ่งบอกว่า CPU ต้องการอ่านข้อมูลจาก หน่วยความจำหรืออุปกรณ์ I/O หน่วยความจำหรืออุปกรณ์ I/O ควรใช้ขานี้เป็นสัญญาณเปิด เกทเข้าสู่บัสข้อมูลของ CPU
- $\overline{RFSH}$**  Refresh (Output, active Low) ใช้ร่วมกับ  $\overline{MREQ}$  เพื่อบ่งบอกว่าขณะนี้ CPU กำลังใช้ 8 บิต-ล่างของ Address Bus ซึ่งไปยังหน่วยความจำแบบ Dynamic เพื่อทำการ Refresh
- $\overline{RTS0}$**  Request to Send 0 (Output, active Low) เป็นขาสัญญาณควบคุมโมเด็มสำหรับ ASCII ช่อง 0
- RXA0, RXA1** Receive Data 0 and 1 (Input, active High) ขานี้ใช้สำหรับรับข้อมูลเข้าไปใน ASCII
- RXS** Clock Serial Receive Data (Input, active High) ขานี้ใช้สำหรับรับข้อมูลเข้าไปใน CSIO
- RXS** มัลติเพล็กซ์กับ  $\overline{CTS1}$  (ของ ASCII ช่อง 1)
- ST** Status (Output, active High) สัญญาณนี้ใช้ร่วมกับ  $\overline{M1}$  และ  $\overline{HALT}$  เพื่อ decode สถานะของ CPU ดังตารางที่ 1

ST	$\overline{HALT}$	$\overline{M1}$	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 for 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 1 แสดงสภาวะการทำงานของ CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{TEND0}$ , $\overline{TEND1}$	Transfer End 0 and 1 (Output, active Low) ขานี้จะ active ระหว่าง cycle สุดท้ายของการทำ DMA มันใช้สำหรับบ่งบอกจุดสิ้นสุดการถ่ายข้อมูลชุดนั้น $\overline{TEND0}$ มัลติเพล็กซ์กับ $CKA1$
TOUT	Timer Out (Output, active High) TOUT คือสัญญาณพัลส์ที่ออกจาก PRT ช่อง 1 ขานี้มัลติเพล็กซ์กับ A18
TXA0, TXA1	Transmit Data 0 and 1 (Output, active High) ขาเหล่านี้ใช้สำหรับส่งข้อมูลออกจาก ASCII Channels
TXS	Clocked Serial Transmit Data (Output, active High) ขานี้ใช้สำหรับส่งข้อมูลออกจาก CSIO Channel
$\overline{WAIT}$	Wait (Input, active Low) $\overline{WAIT}$ ใช้แจ้ง CPU ว่าหน่วยความจำหรืออุปกรณ์ I/O ไม่พร้อมในการรับส่งข้อมูล ขาอินพุตนี้จะต้องป้อนสัญญาณเข้าไปในช่วงของไอเคิลปัจจุบัน สัญญาณ $\overline{WAIT}$ นี้จะถูกตรวจสอบที่ขอบขาของ T2 (ต่อมาก็คือจะเป็นช่วง Wait state) ถ้าอินพุตเป็น Low สถานะ Wait state จะถูกรวมเข้าไว้ในไอเคิลนั้น จนกระทั่งอินพุตเป็น High คำสั่งนั้นจะถูกปฏิบัติต่อไป
$\overline{WR}$	Write (Output, active Low, 3-state) $\overline{WR}$ บ่งบอกว่า CPU มีข้อมูลอยู่บนสาย Data Bus เพื่อที่จะเก็บลงในตำแหน่งของหน่วยความจำหรืออุปกรณ์ I/O
XTAL	Crystal (Input, active High) เป็นจุดต่อผลึกคริสตอล ถ้าเราใช้สัญญาณนาฬิกาภายนอกแล้ว (external clock) ก็ไม่ต้องต่อคริสตอล
<b>Multiplexed pin descriptions</b>	
A18/ $\overline{TOUT}$	ระหว่างรีเซ็ตขานี้จะเป็น A18 ถ้า TOC1 หรือ TOC0 บิตของ Timer Control Register (TCR) ถูกเซ็ตเป็น 1 ขานี้จะเป็น TOUT ถ้า TOC1 และ TOC0 ถูก Clear เป็น 0 ขานี้จะเป็น A18
CKA0/ $\overline{DREQ0}$	ระหว่างรีเซ็ตขานี้จะเป็น CKA0 ถ้า DM1 หรือ SM1 ใน DMA Mode Register (DMODE) ถูกเซ็ตเป็น 1 ขานี้จะเป็น $\overline{DREQ0}$
CKA1/ $\overline{TEND0}$	ระหว่างรีเซ็ตขานี้จะเป็น CKA1 ถ้าบิต CKA1D ใน ASCII Control Register ช่อง 1 (CNTLA1) ถูกเซ็ตเป็น 1 ขานี้จะเป็น $\overline{TEND0}$ ถ้าบิต CKA1D ถูก Clear เป็น 0 ขานี้จะเป็น CKA1
RXS/ $\overline{CTS1}$	ระหว่างรีเซ็ตขานี้จะเป็น RXS ถ้าบิต CTS1E ใน ASCII Status Register ช่อง 1 (STAT1) ถูกเซ็ตเป็น 1 ขานี้จะเป็น $\overline{CTS1}$ ถ้า CTS1E ถูก Clear เป็น 0 ขานี้จะเป็น RXS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

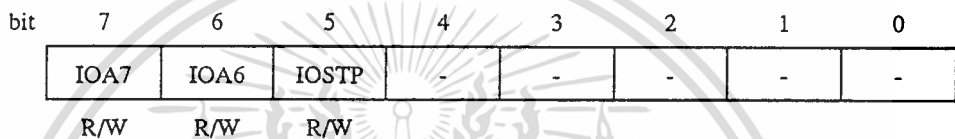


## Internal I/O Registers

I/O register ภายใน Z80180 มีทั้งหมด 64 แอดเดรส (รวมทั้งที่ reserve ไว้ด้วย) แบ่งเป็น Internal I/O Modules (เช่นn ASCII, CSI/O, PRT) และ Control function (DMAC, DRAM refresh, interrupts, wait state generator, MMU, และ I/O relocation)

เพื่อที่จะหลีกเลี่ยงไม่ให้ I/O address ภายในและภายนอก มีตำแหน่งที่ทับซ้อนกัน I/O register ภายใน Z80180 สามารถเคลื่อนย้ายกลุ่มของ internal I/O ภายใน 256 byte ล่างสุดของพื้นที่ I/O ทั้งหมด 64 Kbyte.

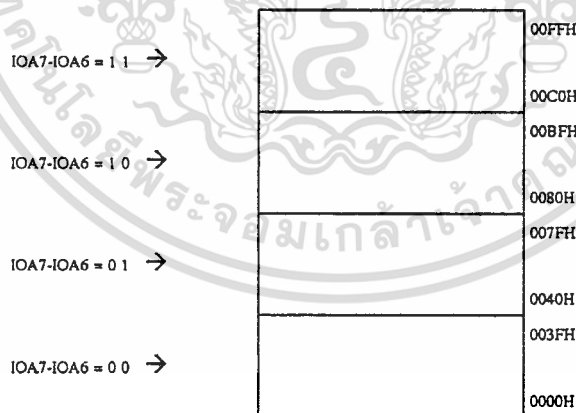
I/O Control register (ICR): ICR ใช้สำหรับเคลื่อนย้าย internal I/O address พร้อมทั้งสามารถควบคุมการ enable/disable IOSTOP โหมดได้



รูปที่ 1 I/O Control Register (ICR : I/O Address = 3FH)

IOA7,6: Address Relocation (Bits 7,6)

IOA7 และ IOA6 ใช้ย้ายตำแหน่งของ internal I/O ดังรูปที่ 2 ข้อสังเกต 8 bit บนของ I/O address จะเป็น "0" เสมอ สภาวะ reset IOA7, IOA6 จะเคลียร์เป็น "0"



รูปที่ 2 I/O Address Relocation

IOSTP : IOSTOP Mode (bit 5) ถ้าเซ็ทบิตนี้เป็น 1 IOSTOP จะ enable ปกติแล้วถ้าจะใช้งาน internal I/O จะ Reset IOSTOP เป็น "0" สภาวะ Reset IOSTOP จะเป็น "0"

internal I/O Register Address Map ตำแหน่งของ internal I/O แสดงดังตารางที่ 2 ตำแหน่ง I/O เหล่านี้เป็นตำแหน่งสัมพันธ์กับ base address ที่กำหนดไว้ใน ICR

I/O Address Notes : ตำแหน่งของ internal I/O register จะอยู่ในพื้นที่ระหว่าง 0000H-00FFH (16 บิต I/O address) ดังนั้นในการออกคำสั่งเกี่ยวกับ Internal I/O 8 บิตบนของ 16 บิต I/O address ต้องเป็น "0" คำสั่งเกี่ยวกับ I/O โดยทั่วไป (OUT (M),A / In A,(M) / OUTI / INI / etc.) ได้ใช้ 8 บิตบนของ 16 บิต I/O address ในการกำหนดตำแหน่งของ I/O ด้วย ดังนั้นจึงเป็นการยากที่จะใช้งาน Internal I/O registers.

เพื่อที่จะให้มีประสิทธิภาพในการใช้งาน Internal I/O register จึงได้เพิ่มคำสั่งใหม่เข้าไปเพื่อทำให้ 8 บิตบนของ 16 บิต I/O address เป็น "0" คำสั่งเหล่านี้คือ IN0, OUT0, OTIMR, OTIM, OTDM, OTDMR และ TSTIO

เมื่อเขียนข้อมูลลงบน internal I/O register, I/O write cycle ก็จะปรากฏบน external bus ด้วย ปกติแล้ว I/O ภายนอกควรจะ decode address ให้ไม่ตรงกับตำแหน่งของ internal I/O เพื่อหลีกเลี่ยงการทำงานที่พร้อมกันทั้ง Internal และ External I/O

	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	CNTLBI	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	TMDR0L	XX001100	0CH
	Timer Data Register Ch 0H	TMDR0H	XX001101	0DH
	Reload Register Ch 0L	RLDR0L	XX001110	0EH
	Reload Register Ch 0H	RLDR0H	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	

ตารางที่ 2 I/O Address Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Register	Mnemonic	Address	
			Binary	Hexadecimal
Others	Free Running Counter	FRC	XX011000	18H
	Reserved		XX011001	19H
			↓	↓
			XX011111	1FH
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	BCR0L	XX100110	26H
	DMA Byte Count Register Ch 0H	BCR0H	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
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 I/O Address Map (continue)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Interrupts.

CPU Z80180 มี 12 interrupt แบ่งเป็น ภายนอก 4 , ภายใน 8 พร้อมกับลำดับความสำคัญของ Interrupt ไว้แล้ว (ดูรูป 3 ประกอบ)

Higher	(1) TRAP(Undefine Op-code Trap).....Internal Interrupt	
Priority	(2) NMI (Non Maskable Interrupt)	
	(3) INT <sub>0</sub> (Maskable Interrupt Level 0)	External Interrupt
	(4) INT <sub>1</sub> (Maskable Interrupt Level 1)	
	(5) INT <sub>2</sub> (Maskable Interrupt Level 2)	
	(6) Timer 0	
	(7) Timer 1	
	(8) DMA channel 0	
	(9) DMA channel 1	Internal Interrupt
	(10) Clocked Serial I/O Port	
Lower	(11) Asynchronous SCI channel 0	
Priority	(12) Asynchronous SCI channel 1	

รูปที่ 3 แสดงแหล่งกำเนิดสัญญาณอินเทอร์รัพท์ต่าง ๆ

ในส่วนนี้จะอธิบายถึง CPU register ต่าง ๆ ที่ทำงานร่วมกับการ interrupt, TRAP interrupt, โหมดการตอบสนองการ interrupt และ interrupt ภายนอก รายละเอียดของแหล่งกำเนิดสัญญาณ interrupt ภายใน (ยกเว้น TRAP) จะต้องทำงานร่วมกับส่วนของ Hard ware อย่างเหมาะสมด้วย (เช่น PRT, DMAC, ASCI, and CSI/O)

Interrupt Control Registers และ Flags: Z80180 มี 3 registers และ 2 Flags ซึ่งใช้งานร่วมกับกระบวนการ Interrupt.

Function	Name	Access Method
(1) Interrupt Vector High	I	LD A,I และ LD I,A
(2) Interrupt Vector Low	IL	I/O instruction (addr=33H)
(3) Interrupt TRAP Control	ITC	I/O instruction (addr=34H)
(4) Interrupt enable Flag 1,2	IEF1,IEF2	EI และ DI

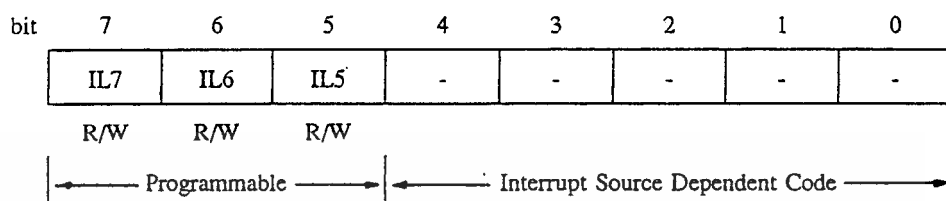
### Interrupt Vector Register ( I )

INT<sub>0</sub> Mode 2 , INT<sub>1</sub> และ INT<sub>2</sub> (interrupt ภายนอก) และ interrupt ภายในทั้งหมด (ยกเว้น TRAP) ใช้เทคนิคการโปรแกรมเวคเตอร์ เพื่อคำนวณตำแหน่งเริ่มต้นของโปรแกรม interrupt ในการตอบสนองการ interrupt จะผลิต address ขนาด 16 บิต มาประกอบกับตารางเวคเตอร์ในหน่วยความจำ ซึ่งจะได้รับตำแหน่งเริ่มต้นของโปรแกรม interrupt นั้น

คำสั่งเกี่ยวกับ Interrupt Vector register จะใช้ LD A,I และ LD I,A มากกว่าคำสั่ง IN/OUT

### Interrupt vector low register

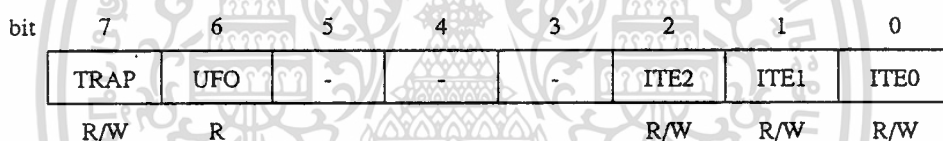
Register นี้ใช้ 3 บิตบน ในการคำนวณตำแหน่ง interrupt vector table สำหรับ interrupt ภายนอก  $INT_1, INT_2$  และ Interrupt ภายในทั้งหมด (ยกเว้น TRAP) 5 บิตล่าง จะถูกกำหนดไว้สำหรับแต่ละแหล่งกำเนิด Interrupt โดยการโปรแกรม IL register ตาราง vector สามารถเคลื่อนย้ายได้เป็นกลุ่ม กลุ่มละ 32 byte ในสถานะ RESET IL จะถูกรีเซ็ตเป็น 00H



รูปที่ 4 Interrupt Vector Low Register (IL : I/O Address = 33H)

### INT/TRAP control register ( ITC )

ITC ใช้สำหรับควบคุมการ TRAP interrupt และ enable/disable interrupt ภายนอก เฉพาะ Maskable interrupt  $INT_0, INT_1, INT_2$



รูปที่ 5 INT/TRAP Control Register (ITC : I/O Address = 34H)

### TRAP ( bit7 )

บิตนี้จะถูกเซ็ตเป็น "1" เมื่อพีทซ์ op-code ที่ไม่สามารถรับรู้ได้ ผู้โปรแกรมสามารถเขียนบิตนี้ได้ด้วย "0" แต่ไม่สามารถเขียนได้ด้วย "1" TRAP จะเป็น "0" ในสถานะ reset

### UFO:Undefined Fetch object ( bit6 )

เมื่อเกิด TRAP interrupt ขึ้น ค่าใน UFO สามารถบอกตำแหน่งเริ่มต้นของ undefined op-code นั้นได้ ถ้า UFO=0 op-code เริ่มต้นจะอยู่ที่ตำแหน่ง pc-1 ถ้า UFO=1 op-code เริ่มต้นจะอยู่ที่ stack pc-2 UFO นั้น อ่านได้อย่างเดียว

### ITE2,1,0: Interrupt Enable 2,1,0 ( bits 2-0 )

ITE2,ITE1,ITE0 ใช้ enable/disable interrupt input  $INT_2, INT_1, INT_0$  ตามลำดับ ถ้า ITE รีเซ็ตเป็น "0" การ interrupt จะไม่ตอบรับ สถานะรีเซ็ต ITE0=1 ขณะที่ ITE1 และ ITE2=0

### Interrupt enable flag 1,2 (IEF1,IEF2)

IEF1 ควบคุม enable/disable interrupt ภายใน,ภายนอกทั้งหมด(ยกเว้น NMI,TRAP) ถ้า IEF1=0 maskable interrupt ทั้งหมดจะไม่ตอบรับ โดยใช้คำสั่ง DI และเซ็ทเป็น "1" โดยคำสั่ง EI

IEF2 ใช้สำหรับการ NMI ในช่วง NMI ลำดับความสำคัญของการ interrupt จะถูก save และ maskable interrupt ทั้งหมดจะถูก disable (IEF1 copied to IEF2 และ IEF1 เคลียร์เป็น"0") ที่คำสั่งสุดท้ายของ NMI คือ RETN (return from non-maskable interrupt) จะนำค่าสถานะความสำคัญของ maskable interrupt กลับคืน (IE2 copied to IE1) IEF2 สามารถมีผลต่อ P/V bit ของ cpu status register โดยคำสั่ง LD A,I หรือ LD A,R

CPU Operation	IEF1	IEF2	REMARKS
RESET	0	0	Inhibits the interrupt except NMI and TRAP.
NMI	0	IEF1	Copies the contents of IEF1 to IEF2.
RETN	IEF2	not affected	Return from the NMI service routine.
Interrupt except NMI and TRAP	0	0	Inhibits the interrupt except NMI and TRAP
RETI	not affected	not affected	
TRAP	not affected	not affected	
EI	1	1	
DI	0	0	
LD A,I	not affected	not affected	Transfers the contents of IEF2 to P/V flag.
LD A,R	not affected	not affected	Transfers the contents of IEF2 to P/V flag.

ตารางที่ 3 แสดงสถานะของ IEF1 และ IEF2

External Interrupt Z80180 มี external interrupt อยู่ 4 ชนิด

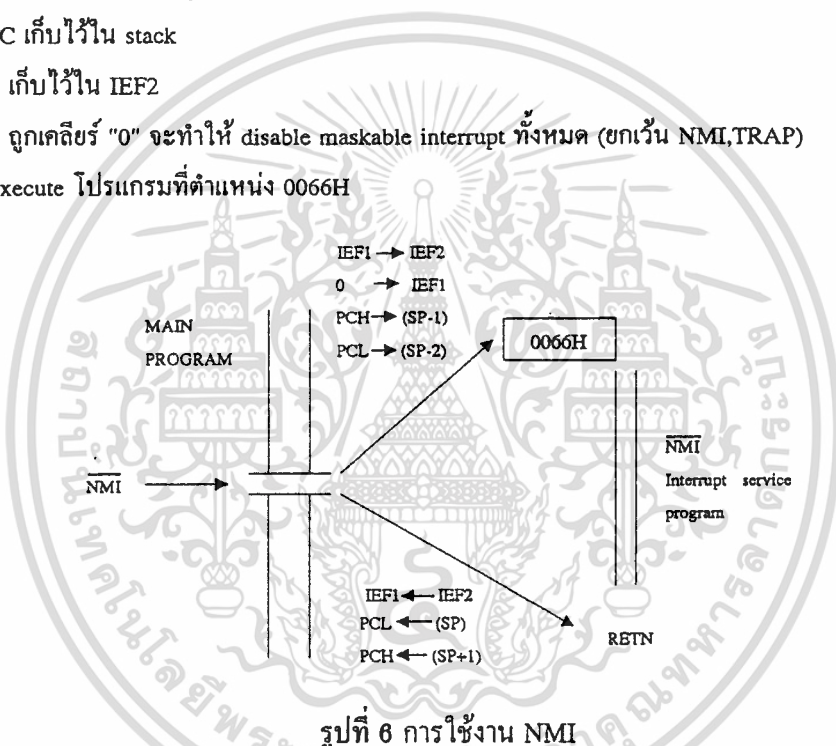
1. NMI :Non-maskable interrupt
2.  $INT_0$ :Maskable interrupt level 0
3.  $INT_1$ :Maskable interrupt level 1
4.  $INT_2$ :Maskable interrupt level 2

NMI,  $INT_1$ ,  $INT_2$  มี Mode การทำงานที่เฉพาะตัว

$INT_0$  มี 3 Mode การทำงาน สามารถที่จะโปรแกรมได้ด้วย Software

NMI (Non-maskable interrupt) สัญญาณ NMI input จะ detect ที่ขอบของสัญญาณและไม่สามารถยับยั้งการตอบรับได้ด้วย Software เมื่อ NMI เกิดขึ้น Z80180 จะปฏิบัติดังนี้

1. กระบวนการ DMA จะถูกพักไว้ชั่วคราวโดยเปลี่ยนบิต DME (DMA Main Enable) ใน DCNTL register
2. ค่า PC เก็บไว้ใน stack
3. IEF1 เก็บไว้ใน IEF2
4. IEF1 ถูกเคลียร์ "0" จะทำให้ disable maskable interrupt ทั้งหมด (ยกเว้น NMI,TRAP)
5. ไป execute โปรแกรมที่ตำแหน่ง 0066H



คำสั่งสุดท้ายของโปรแกรม NMI ควรเป็น RETN ซึ่งนอกจากจะเป็นการนำค่า stack คืนสู่ PC แล้ว จะนำค่า IEF2 กลับมายัง IEF1 ด้วย

$INT_0$  Maskable interrupt level 0

interrupt ภายนอกที่มีความสำคัญต่อจาก NMI คือ  $INT_0$ ,  $INT_0$  จะถูกตรวจสอบทุก ๆ ขอบขาลงของ Clock T3 หรือ T1 ของ Machine Cycle ล่าสุด ถ้า  $INT_0=0$   $INT_0$  จะถูกตอบรับหรือไม่นั้นขึ้นอยู่กับ IE1 flag หรือ ITE0 bit (interrupt enable 0) ใน ITC ในสภาวะหลังจาก reset จะเป็นดังนี้

1. IEF1=0 ดังนั้น  $INT_0$  จะไม่ทำงาน
2. ITE0=1 ดังนั้น  $INT_0$  จะ enable โดยใช้คำสั่ง IE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INT<sub>0</sub> สามารถโปรแกรมให้ทำงานได้ 3 Mode คือ Mode0, Mode1, Mode2 การกำหนดโหมดทำได้ด้วยคำสั่ง IM0, IM1 และ IM2 (set interrupt mode) สภาวะ reset Z80180 กำหนดให้ใช้ Mode0 interrupt ทั้ง 3 mode มีดังนี้

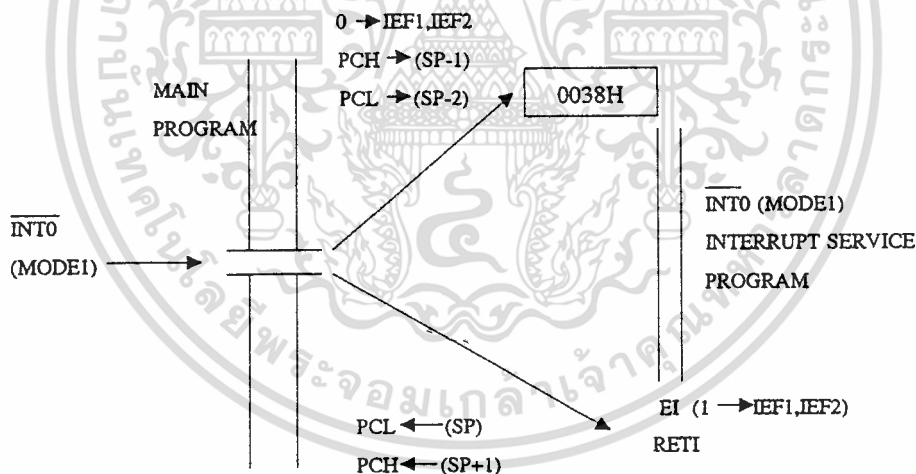
1. Mode0 : คำสั่งจะถูก Fetch จาก data bus
2. Mode1 : reset ที่ address 0038H
3. Mode2 : Low byte vector table จะ Fetch จาก data bus

#### INT<sub>0</sub> Mode 0.

ในช่วงไซเคิลตอบรับการ interrupt คำสั่งจะถูก Fetch จาก data bus (D0-D7) ที่ขอบขาขึ้นของ T3 คำสั่งเหล่านี้ ได้แก่ คำสั่ง RST (restart) ซึ่งจะเก็บค่า PC และเริ่ม restart โปรแกรมที่ตำแหน่งตายตัว

#### INT<sub>0</sub> Mode 1

เมื่อได้รับ INT<sub>0</sub> เข้ามา PC จะถูกเก็บไว้ใน Stack ที่ address 0038H ทั้ง IEF1 และ IEF2 ถูกเคลียร์เป็น "0" เพื่อ disable maskable interrupt ทั้งหมด โปรแกรม interrupt service routine ควรจะสิ้นสุดด้วยคำสั่ง EI และ RETI ตามลำดับ

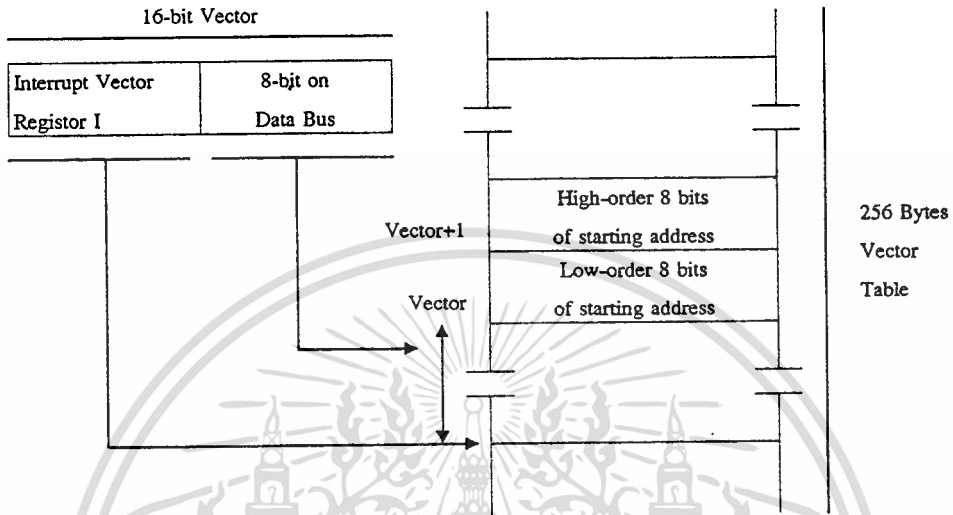


รูปที่ 7 INT<sub>0</sub> Mode 1 Interrupt sequence

### INT<sub>0</sub> Mode 2

วิธีนี้จะหาค่า restart address โดยอ่านจากตารางใน Memory ตาราง vector นั้น ประกอบด้วย 128 two-byte restart address เก็บเป็น low byte และ high byte ตามลำดับ

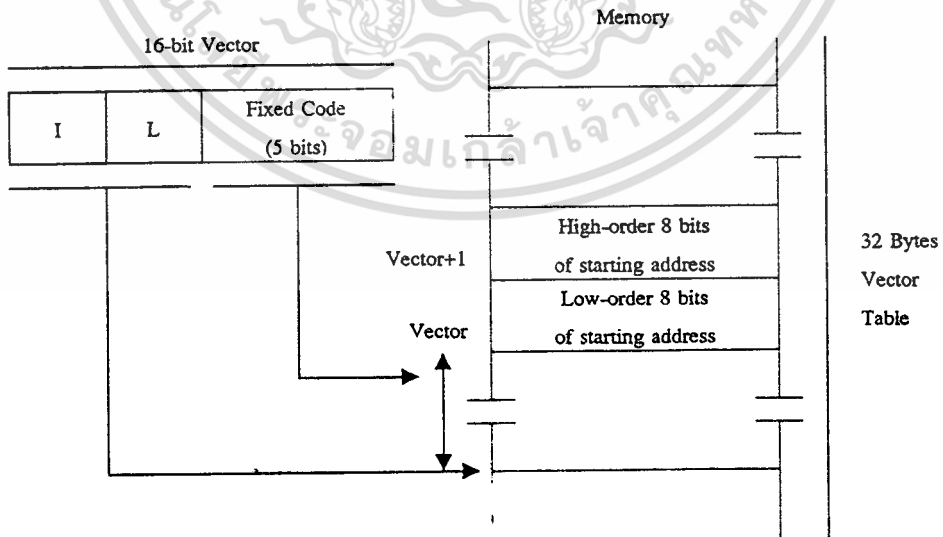
ตาราง vector จะมีขนาด 256 byte ซึ่งจะอยู่ที่ใดก็ได้ใน 64 Kbyte ซึ่งสามารถโปรแกรมได้ที่ interrupt vector register (I)



รูปที่ 8 INT<sub>0</sub> Mode 2 vector Acquisition

### INT<sub>1</sub>, INT<sub>2</sub>

การทำงานของ external interrupt INT<sub>1</sub> และ INT<sub>2</sub> จะใช้วิธี vector table mode คล้ายกับ INT<sub>0</sub> - mode 2 ที่แตกต่างกันคือ INT<sub>1</sub> และ INT<sub>2</sub> จะผลิต low order byte ของ vector table โดยใช้ IL (interrupt vector low) register มากกว่าที่จะ fetch จาก data bus โดยตรง



รูปที่ 9 INT<sub>1</sub>, INT<sub>2</sub> vector Acquisition

INT<sub>1</sub> และ INT<sub>2</sub> จะไม่ตอบรับถ้า IEF1=0 และสามารถโปรแกรมแต่ละตัวทำได้โดยเคลียร์ ITE1

และ ITE2 (bit 1,2) ของ INT/TRAP control register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Internal interrupt

(ยกเว้น TRAP) ใช้ตาราง vector คล้ายๆกับ  $INT_1, INT_2$  Internal interrupt ทั้งหมดจะไม่ตอบรับ ถ้า  $IEF1=0$  หรือจะไม่ตอบรับเพียงบางตัวได้โดย เซทที่แต่ละ I/O control register (PRT,DMAC,CSI/O,ASCI)

Interrupt Source	Priority	II			Fixed Code				
		b7	b6	b5	b4	b3	b2	b1	b0
INT1	Highest	*	*	*	0	0	0	0	0
INT2		*	*	*	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
CSI/O		*	*	*	0	1	1	0	0
ASCI channel 0		*	*	*	0	1	1	1	0
ASCI channel 1		Lowest	*	*	*	1	0	0	0

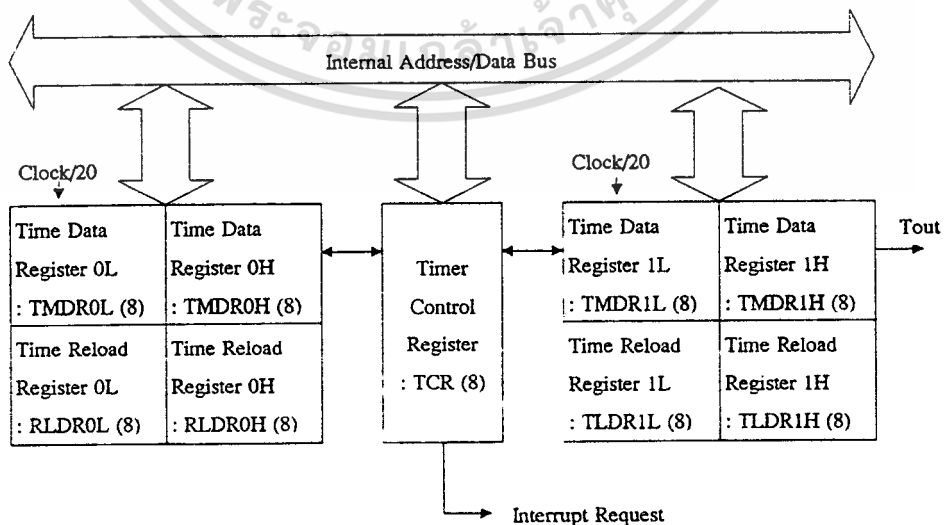
\*Programmable

ตารางที่ 4 Vector Table

## การโปรแกรมตัวตั้งเวลา (Programmable Reload Time : PRT)

Z80180 ประกอบด้วย PRT จำนวน 2 ช่องด้วยกัน แต่ละช่องประกอบด้วย 16 บิต down counter และ 16 บิต reload register สำหรับ down counter นั้นสามารถเขียนและอ่านได้โดยตรง และยังกำหนดให้เกิดสัญญาณ interrupt ไปยัง CPU ได้เมื่อนับลงถึง 0 แล้ว ดังนั้น PRT ช่อง 1 จึงมีขา TOUT (ขา 31 มัลติเพล็กซ์กับ A18) ซึ่ง set ให้เป็นสภาวะ High, Low หรือ Toggle ก็ได้

PRT block diagram แสดงดังรูปที่ 10 ทั้ง 2 ช่องมีค่าฐานเวลา และรีโวลด์รีจิสเตอร์ของแต่ละช่องโดยเฉพาะ และใช้ counter register ร่วมกัน สัญญาณ Clock ที่ป้อนให้กับ Timer มาจาก System Clock หารด้วย 20



รูปที่ 10 PRT Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คำบรรยาย PRT รีจิสเตอร์ต่าง ๆ

Timer data register (TMDR : I/O address ช่อง 0 :0DH,0CH ;ช่อง 1:15H,14H) PRT0 และ PRT1 แต่ละช่องจะมี Timer register ขนาด 16 บิต (TMDR) ช่อง 0 คือ TMDR0 , ช่อง 1 คือ TMDR1 โดยแบ่ง เป็น รีจิสเตอร์ขนาด 8 บิต 2 ตัว คือ Low กับ High (TMDR0H,TMDR0L และ TMDR1H,TMDR1L) สภาวะ reset ทั้ง TMDR0 และ TMDR1 ถูกเซ็ทเป็น FFFFH

TMDR จะลดลง 1 ครั้งทุก ๆ 20 สัญญาณนาฬิกาของระบบ เมื่อ TMDR นับลงถึง 0 แล้วจะโหลดค่าฐานเวลาจาก reload register ให้โดยอัตโนมัติ

TMDR อ่านและเขียนโดยใช้วิธีดังต่อไปนี้ การอ่านค่าเวลาจาก PRT จะไม่ทำให้ PRT หยุดการนับลง แต่การเขียนค่าฐานเวลา PRT จะหยุดนับลง สำหรับการอ่าน (ปราศจากการหยุด Timer) TMDR ถูกอ่านจาก Low byte, Higher byte (TMDEnL,TMDRnH)ตามลำดับ วิธีการนี้ทำให้แน่ใจได้เลยว่าจะไม่มีปัญหาของการ update ของค่าฐานเวลา 16 บิตระหว่างการอ่านทีละ 8 บิต ถ้าอ่านค่า TMDR จาก Higher byte และ Low byte ตามหลังอาจจะได้ค่าที่ผิดพลาด โดยปกติแล้วในการโปรแกรมการอ่านค่าจาก TMDR ควรจะปฏิบัติทีเดียวทั้ง Lower และ Higher byte ตามลำดับ สำหรับการเขียน TMDR จะต้องยับยั้ง Timer โดยใช้ TDE บิต (time down count enable) ใน TCR (time control register) ดังนั้น ทั้ง Higher และ Lower byte สามารถโปรแกรมได้โดยอิสระ

Timer reload register (RLDR : I/O Address ช่อง 0:0EH,0FH ช่อง 1:16H,17H) PRT0 และ PRT1 จะมี 16 บิต time reload register (RLDR) ของตัวเอง RLDR0 และ RLDR1 แบ่งเป็น รีจิสเตอร์ 8 บิต 2 ตัว คือ Low และ High byte register (RLDR0H, RLDR0L, RLDR1H, RLDR1L) สภาวะ reset RLDR0 และ RLDR1 จะถูก set เป็น FFFFH

Note :เมื่อ TMDR นับลงถึง 0 มันจะโหลดค่าฐานเวลาจาก RLDR ให้โดยอัตโนมัติ

Timer control register (TCR) ใช้แสดงสถานะของทั้ง 2 ช่อง (PRT0, PRT1) ทั้งยังสามารถเริ่มหรือหยุดการนับ และให้มีหรือไม่มีสัญญาณ Interrupt ออกไปที่ CPU และที่ขา A18/TOUT สำหรับช่อง PRT1 ได้

bit	7	6	5	4	3	2	1	0
	TIF1	TIF0	TIE1	TIE0	TOC1	TOC0	TDE1	TDE0
	R	R	R/W	R/W	R/W	R/W	R/W	R/W

รูปที่ 11 Timer Control Register (TCR : I/O Address = 10H)

TIF1 : Timer Interrupt Flag 1 (bit 7) เมื่อ TMDR1 ลดลงถึง 0 TIF1 จะเซ็ทเป็น "1" และกำเนิดสัญญาณ interrupt ออกไปถ้า TIE1=1 TIF1 จะรีเซ็ทเป็น "0" เมื่อ TCR ถูกอ่าน และ Higher หรือ Lower byte ของ TMDR1 ถูกอ่าน ในสภาวะ reset TIF1จะถูกเคลียร์เป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TIF0 : Timer Interrupt Flag 0 (bit 6)** เมื่อ TMDR0 ลดลงถึง 0 TIF0 จะเซ็ทเป็น "1" และกำเนิดสัญญาณ interrupt ออกไปถ้า TIE0=1 TIF0 จะรีเซ็ทเป็น "0" เมื่อ TCR ถูกอ่าน และ Higher หรือ Lower byte ของ TMDR0 ถูกอ่าน ในสถานะ Reset TIF0 ถูกเคลียร์เป็น "0"

**TIE1 : Timer Interrupt Enable 1 (bit 5)** เมื่อ TIE1 ถูกเซ็ทเป็น "1" TIF1=1 จะกำเนิดสัญญาณ Interrupt ไปยัง CPU เมื่อ TIE1 ถูก reset เป็น "0" สัญญาณการร้องขอการ Interrupt จะถูกยับยั้งไว้ ในสถานะรีเซ็ท TIE1 จะถูกเคลียร์เป็น "0"

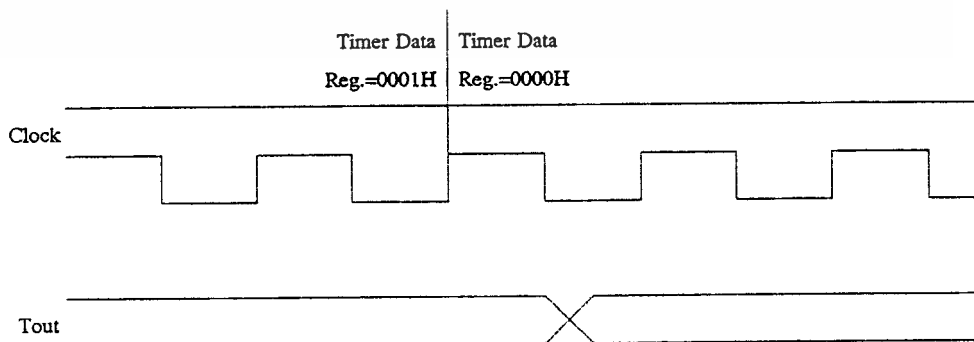
**TIE0 : Timer Interrupt Enable 0 (bit 4)** เมื่อ TIE0 ถูกเซ็ทเป็น "1" TIF0=1 จะกำเนิดสัญญาณ Interrupt ไปยัง CPU เมื่อ TIE0 ถูก reset เป็น "0" สัญญาณการร้องขอการ Interrupt จะถูกยับยั้งไว้ ในสถานะรีเซ็ท TIE0 จะถูกเคลียร์เป็น "0"

**TOC1,0 : Timer Output Control (bit 3,2)** TOC1 และ TOC0 ควบคุมสัญญาณเอาต์พุตจาก PRT1 ทางขา A18/TOUT ดังตารางที่ 5 ในสถานะรีเซ็ท TOC1 และ TOC0 ถูกเคลียร์เป็น "0" ดังนั้นขา A18/TOUT จะเป็นขา address โดยการโปรแกรม TOC1 และ TOC0 ขา A18/TOUT สามารถตั้งให้เป็น High , Low หรือ Toggled ก็ได้เมื่อ TMDR1 ลดลงถึง 0

TOC1	TOC0	OUTPUT
0	0	Inhibited (A <sub>18</sub> /TOUT pin is selected as an address output function)
0	1	toggled* (A <sub>18</sub> /TOUT pin is selected as a
1	0	0 PRT1 output function)
1	1	1

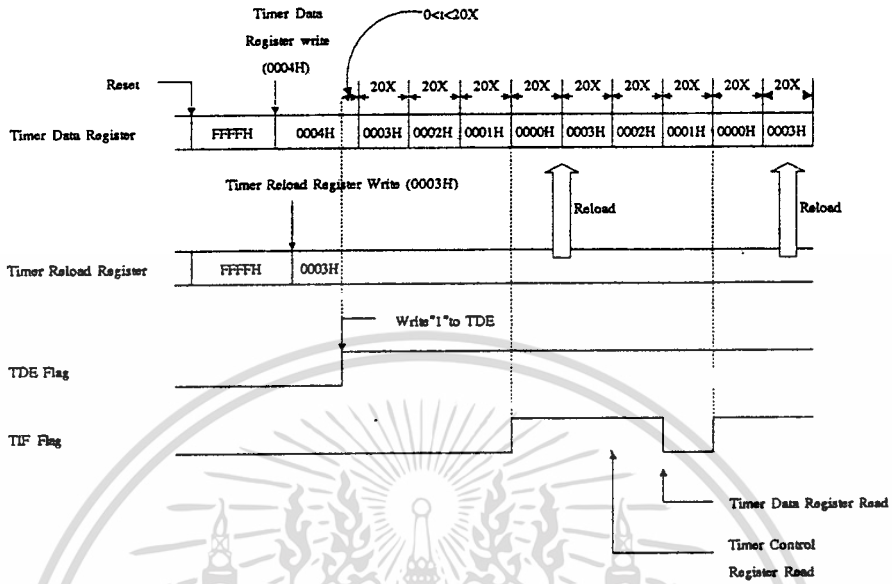
ตารางที่ 5 Timer Output Control

**TDE1,0 : Timer Down Count Enable (bit 1,0)** TDE1 และ TDE0 ทำหน้าที่เป็นแฟล็ก ON/OFF การนับสำหรับ TMDR1 และ TMDR0 โดยลำดับ เมื่อ TDE<sub>n</sub> (n=0,1) ถูกเซ็ทเป็น "1" TMDR<sub>n</sub> ก็จะเริ่มนับลง เมื่อ TDE<sub>n</sub> ถูกรีเซ็ทเป็น "0" TMDR<sub>n</sub> จะหยุดนับ TMDR<sub>n</sub> จะไม่ลดลงจนกว่า TDE<sub>n</sub> ถูกเซ็ทเป็น "1" ในสถานะ reset TDE1 และ TDE0 จะถูกเคลียร์เป็น "0"

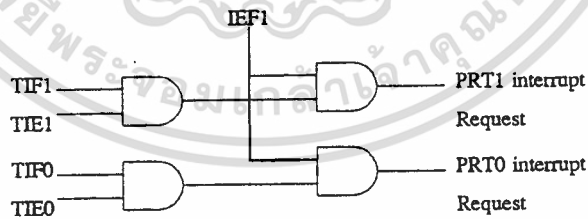


รูปที่ 12 สัญญาณ Output ที่ขา A18/TOUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13 แสดงสถานะเริ่มต้นของ Timer , การนับลงและการรีโหลด



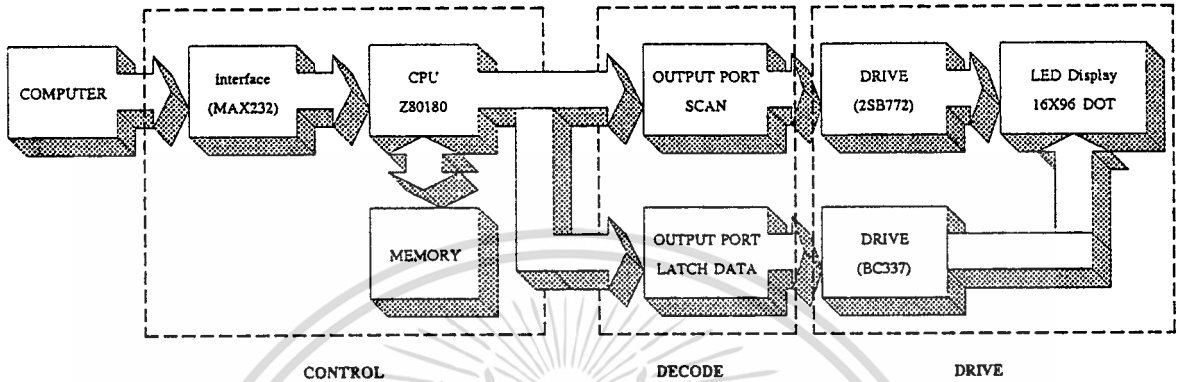
รูปที่ 14 แสดงวงจรสร้างสัญญาณ Interrupt

PRT และ reset ในสถานะ reset บิตต่าง ๆ ใน TCR จะถูกตั้งค่าเริ่มต้นตามคำบรรยายที่กล่าวมา , การนับลงจะถูกหยุด , TMDR และ RLDR ถูกตั้งค่าเป็น FFFFH , ขา A18/TOUT มีสถานะเป็นขา Address

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### หลักการทํางานของวงจร



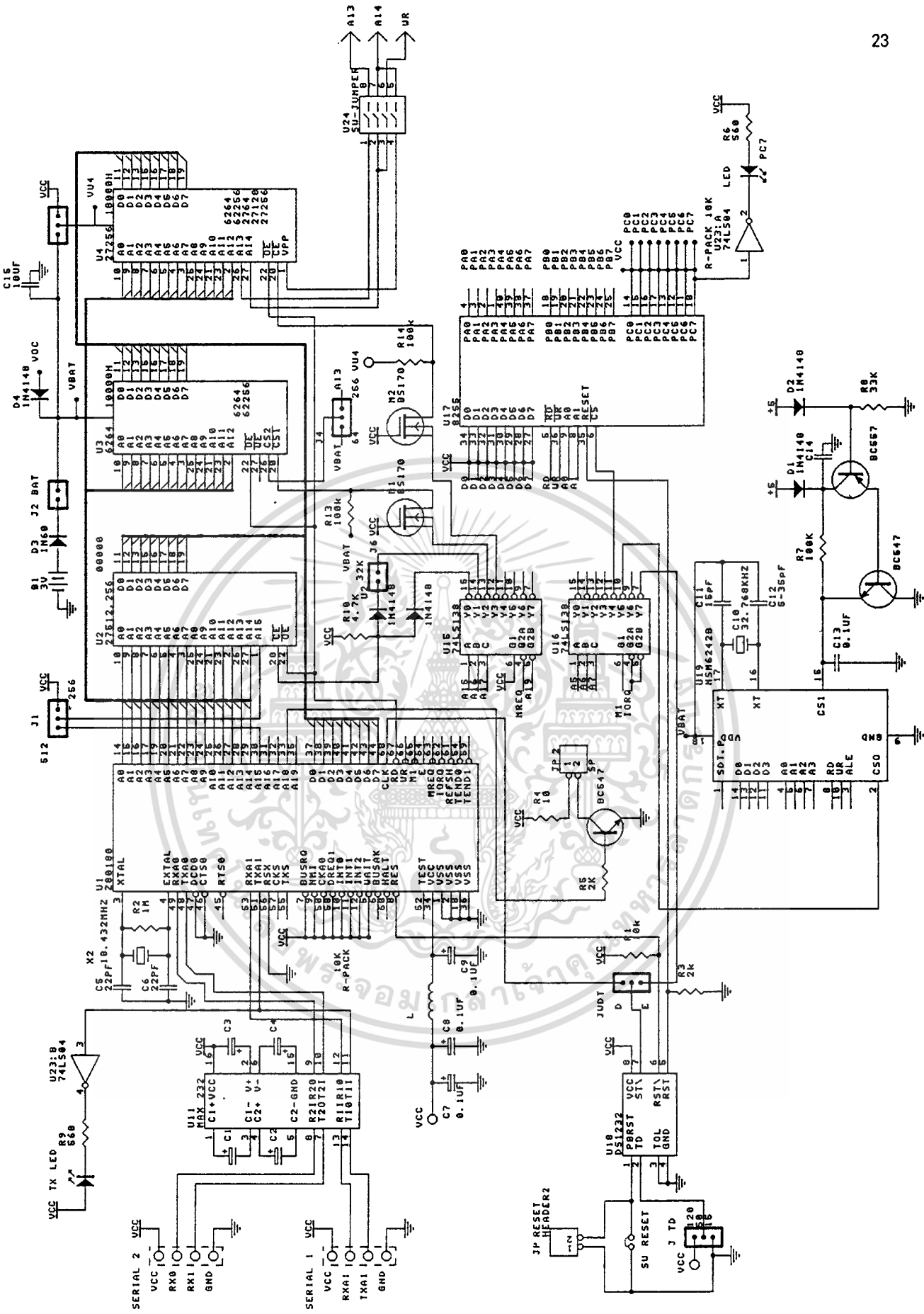
รูปที่ 15 แสดงบล็อกโคอะแกรมของ LED display board

#### อธิบายบล็อกโคอะแกรม

ภาค CONTROL ประกอบด้วย CPU , หน่วยความจำ (ROM/RAM) , อุปกรณ์เชื่อมต่อกับคอมพิวเตอร์ทางพอร์ตอนุกรม

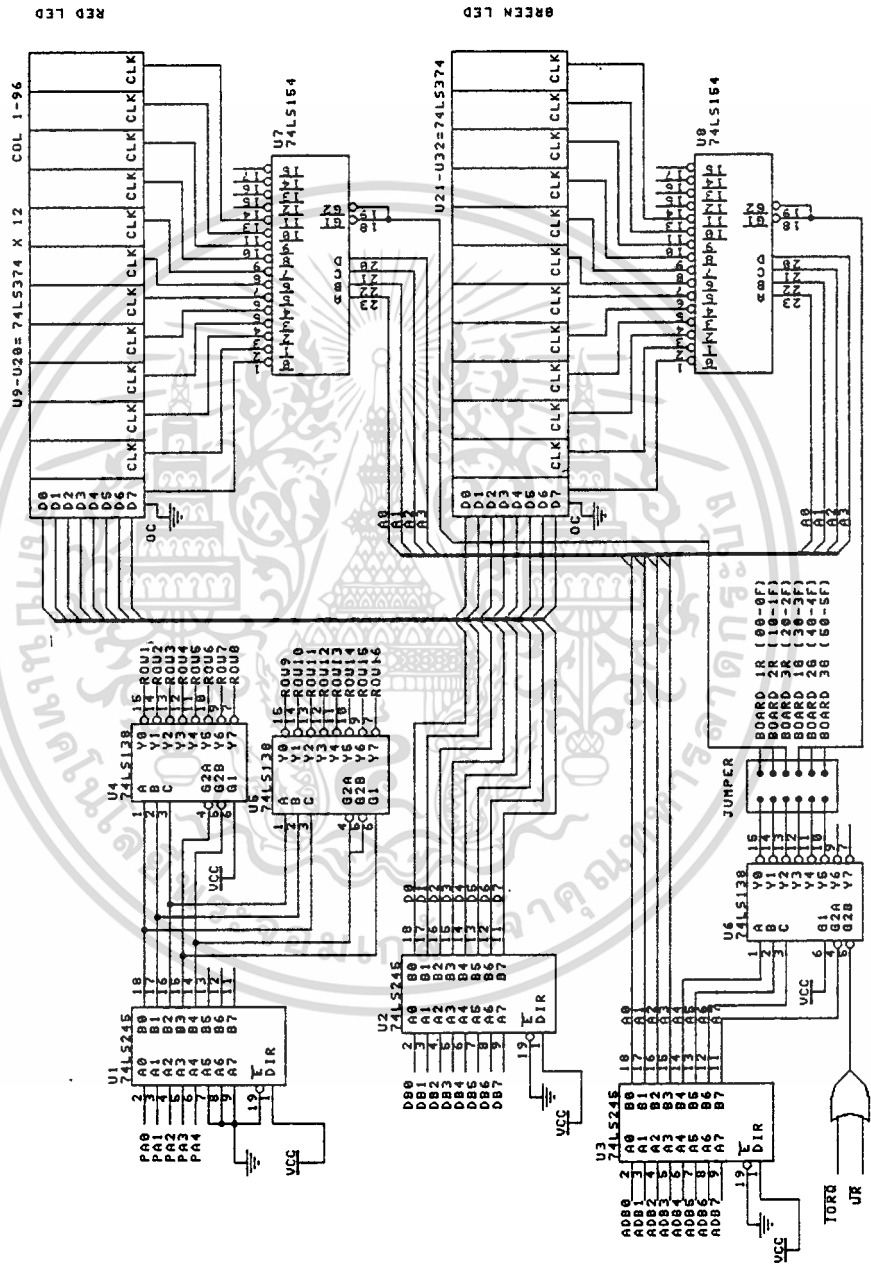
ภาค DECODE ประกอบด้วย Output Port Scan , Output Port Latch Data ทำหน้าที่รับข้อมูลจากภาค CONTROL แล้วถอดรหัส (decode) ตำแหน่งของพอร์ต Scan และ พอร์ต Latch Data

ภาค DRIVE ประกอบด้วย ภาค Drive ด้านแถว (ใช้ทรานซิสเตอร์ 2SB772) , ภาค Drive ด้านคอลัมน์ (ใช้ทรานซิสเตอร์ BC337) และ ภาค LED Display 16X96 dot ทั้งหมดนี้ทำหน้าที่ขับสัญญาณที่มาจากภาค DECODE เพื่อให้เพียงพอดต่อการไบอัส LED ให้สว่างตามต้องการ



รูปที่ 16 วงจรภาคควบคุม (CPU Z80180)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 17 วงจร Decode เบอร์ Port และวงจรถนแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การทำงานของวงจรสแกน

ประกอบไปด้วย U1 ซึ่งเป็น Buffer เบอร์ 74LS245 U4 และ U5 จะเป็นส่วน decode 3 to 8 (IC เบอร์ 74LS138 โดย U4 จะสแกนจากแถว 1 - แถว 8 และ U5 สแกนจากแถว 9 - แถว 16 ซึ่งสัญญาณสแกนแถว 1 ถึง แถว 16 นี้ จะถูกส่งไปยังขาเบสของทรานซิสเตอร์ชนิด PNP เบอร์ 2SB772 เพื่อให้ทรานซิสเตอร์ทำงานและส่งกระแสไบอัสให้กับ LED ต่อไป

การสแกนจะทำการสแกนทีละแถว คือเริ่มจากแถวที่ 1 ก่อน (แถวบนสุด) แล้วไล่ถัดไปยัง แถวที่ 2 , 3 , 4 , .... , 16 จากนั้นก็จะวนกลับมายังแถวที่ 1 อีก และจะสแกนอย่างนี้ไปเรื่อย ๆ สัญญาณ PA0 - PA4 ได้มาจากพอร์ท A ของ IC #8255 โดยการถอดรหัส (Decode) สัญญาณ PA0 - PA4 นี้ จะได้สัญญาณสแกนแถว 1 - แถว 16 ดัง function table ของ U4 และ U5 ดังต่อไปนี้

ตารางที่ 6 Function Table ของ U4

INPUT						OUTPUT							
ENABLE			SELECT			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Vcc	Pa4	Pa3	Pa2	Pa1	Pa0								
G1	G2b	G2a	C	B	A								
1	0	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	1	1	1	0	1	1
1	0	0	1	1	0	1	1	1	1	1	1	0	1
1	0	0	1	1	1	1	1	1	1	1	1	1	0

ตำแหน่ง Input Port

ตำแหน่ง Output Scan

00H

Y0 (Row 1)

01H

Y1 (Row 2)

02H

Y2 (Row 3)

03H

Y3 (Row 4)

04H

Y4 (Row 5)

05H

Y5 (Row 6)

06H

Y6 (Row 7)

07H

Y7 (Row 8)

ตารางที่ 7 Function Table ของ U5

INPUT					OUTPUT							
ENABLE		SELECT			Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
Pa4 G2a+G2b	Pa3 G2a	Pa2 C	Pa1 B	Pa0 A								
0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	1	1	0	1	1	1	1	1	1
0	0	0	1	0	1	1	0	1	1	1	1	1
0	0	0	1	1	1	1	1	0	1	1	1	1
0	0	1	0	0	1	1	1	1	0	1	1	1
0	0	1	0	1	1	1	1	1	1	0	1	1
0	0	1	1	0	1	1	1	1	1	1	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0

ตำแหน่ง Input Port                      ตำแหน่ง Output Scan

- 08H    Y0 (Row 9)
- 09H    Y1 (Row 10)
- 0AH    Y2 (Row 11)
- 0BH    Y3 (Row 12)
- 0CH    Y4 (Row 13)
- 0DH    Y5 (Row 14)
- 0EH    Y6 (Row 15)
- 0FH    Y7 (Row 16)

DATA

1-----16 17-----32 33-----48 49-----64 65-----80 81-----96



ROW1	BLOCK1	BLOCK2	BLOCK3	BLOCK4	BLOCK5	BLOCK6
ROW8						
ROW9	BLOCK7	BLOCK8	BLOCK9	BLOCK10	BLOCK11	BLOCK12
ROW16						



รูปที่ 18 แสดงการวางแผง LED ขนาดแผงละ 8X16 จำนวน 12 แผง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

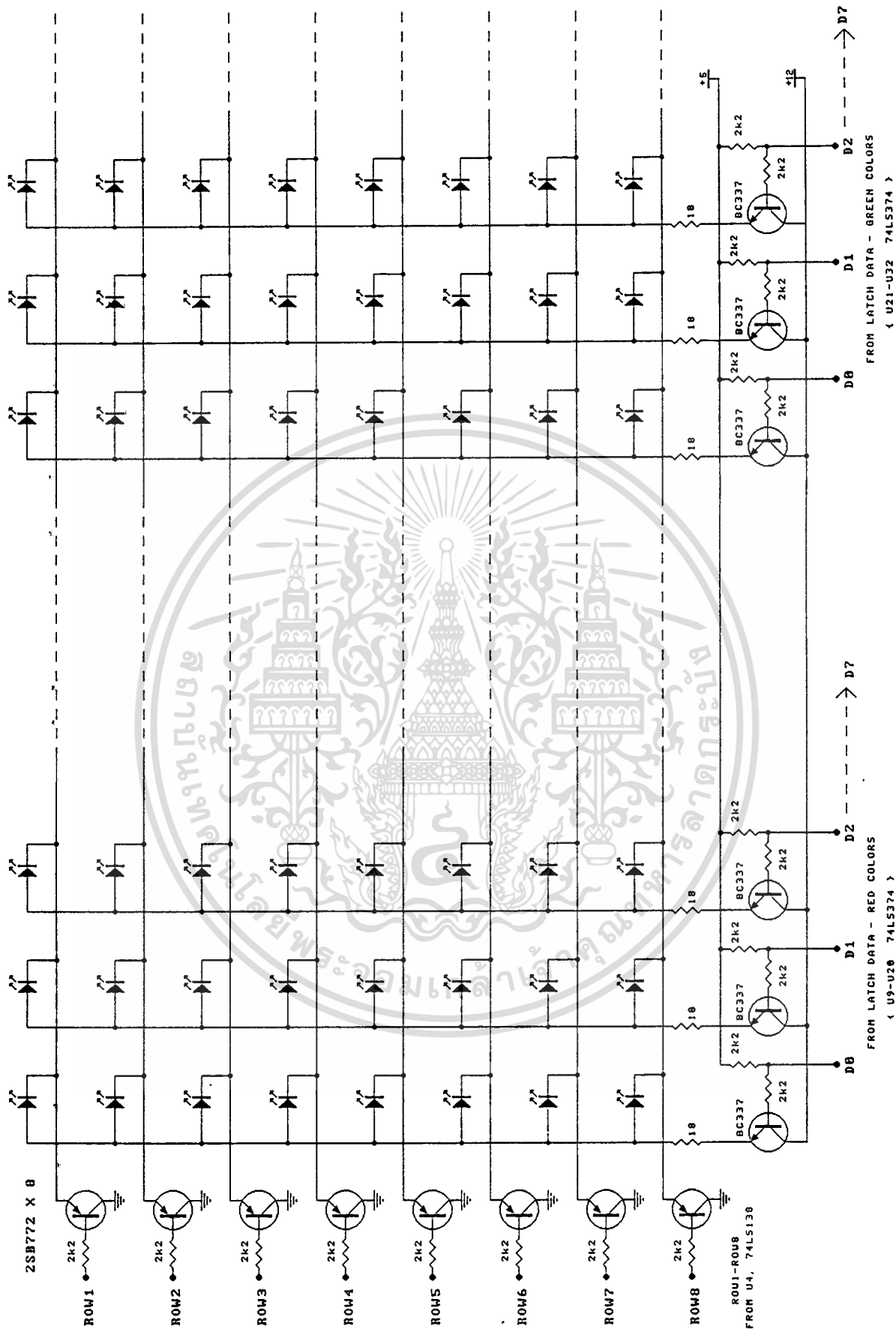
ในการสแกนแต่ละ ROW ถ้าใช้ทรานซิสเตอร์ตัวเดียวรับภาระจ่ายกระแสให้ LED 96 ตัว อาจทำให้กระแสที่จ่ายให้ LED มีไม่เพียงพอ LED ก็จะสว่างไม่เต็มที่ เพราะฉะนั้นในแต่ละ Block จึงให้ ทรานซิสเตอร์รับภาระ LED เพียง 16 ตัว และในแต่ละ Block ก็จะมีทรานซิสเตอร์สำหรับขยายสัญญาณ ด้าน ROW 8 ตัว เพื่อจะได้มีกระแสจ่ายเพียงพอให้กับ LED ในแต่ละ Block ก็จะต่อสัญญาณ ROW ขนานกันคือ Block1-Block6 ต่อขนานกัน ส่วน Block7-Block12 ก็ต่อขนานเช่นเดียวกัน จึงทำให้มี การสแกน LED ทาง ROW 96 ตัวพร้อมกัน ตั้งแต่ ROW1 จนกระทั่งถึง ROW16 และกลับไปยัง ROW1 ทำการสแกนซ้ำไปเรื่อยๆ

#### การทำงานของภาค Decode Port Latch DATA

สาย Data bus และ Address bus จาก CPU จะผ่านเข้ามายัง Buffer U2 และ U3 (74LS245) ตามลำดับ ในช่วงคำสั่ง OUT (#Port) สัญญาณ  $\overline{IORQ}$  กับ  $\overline{WR}$  นำมา OR กันเพื่อ Enable U6(74LS138) Address bus 4 บิตล่างจะถูกนำไปเข้า U7,U8 (74LS154) ส่วน Address bus 4 บิตบนจะนำมาเข้า U6 และ นำ output ของ U6 มา Enable U7,U8 จุดประสงค์ของการนำ 4 บิตบนมาเข้า U6 (Decode 3 To 8) ก็เพื่อ ต้องการเลือกช่วงเบอร์ Port Latch Data (U9-U32) ซึ่งหมายเลข Port ของ U7 ที่จะเป็นไปได้คือ 00H-0FH, 10H-1FH, 20H-2FH และของ U8 หมายเลข Port ที่จะเป็นไปได้คือ 30H-3FH, 40H-4FH, 50H-5FH

U7 จะ Decode port latch DATA ของ LED สีแดง (RED) ดังนั้น LED สีแดง มีหมายเลข Port คือ 00H-0FH, 10H-1FH, 20H-2FH โครงการนี้มี LED แนวตั้ง 16 ตัว เพราะฉะนั้นจึงใช้ U7 2 ตัว หมายเลข Port 00H-0FH, 10H-1FH ซึ่งจะเหลือหมายเลข Port 20H-2FH เอาไว้สำหรับกรณีที่ต้องการ ขยายจำนวน LED ในแนวตั้งเป็น 24 ดวง ก็จะใช้ U7 อีก 1 ตัว

U8 decode port latch DATA ของ LED สีเขียว (GREEN) ซึ่งลักษณะการจัดหมายเลข Port ก็เช่นเดียวกับกรณีของ U7

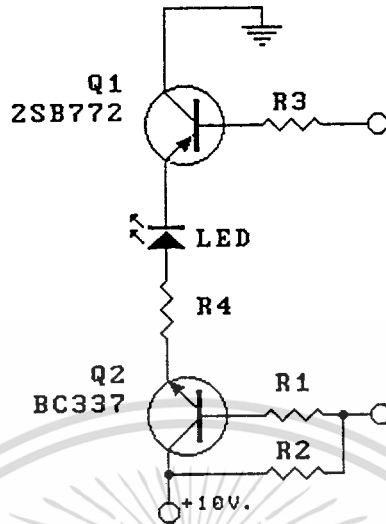


รูปที่ 19 วงจร Drive LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบวงจร drive LED

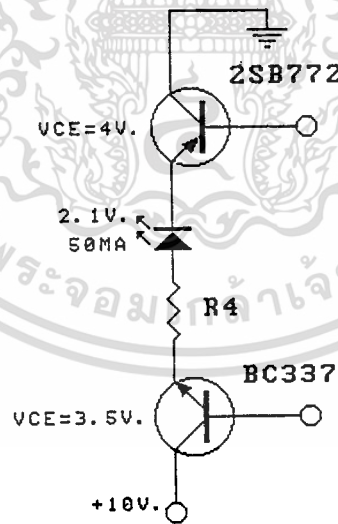
จากรูปวงจร drive LED ขนาด 8 X 8 dot เราตัดเอามาเพียง 1 ส่วน จะเหลือวงจรดังรูปที่ 20



รูปที่ 20 วงจร Drive LED ที่ตัดออกมาเพียง 1 ส่วน

หา R<sub>4</sub> ก่อนโดยไล่ loop output

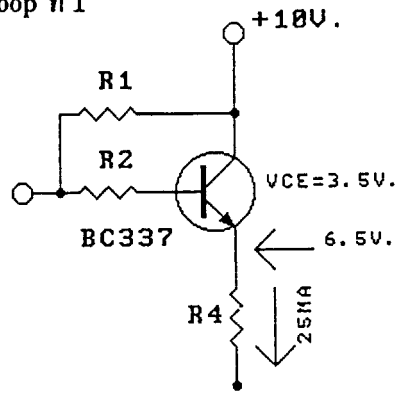
กำหนด voltage ตกคร่อม LED V<sub>F</sub> = 2.1 Volt (Red or Green)



รูปที่ 21 วงจรที่ใช้ในการคำนวณหาค่า R<sub>4</sub>

$$R_4 = \frac{10 - 3.5 - 4 - 2.1}{25\text{mA}} = 16 \Omega \quad \# \text{ (ใช้ } I_F = 25 \text{ mA)}$$

หา  $R_1, R_2$  โดยใช้ loop ที่ 1



รูปที่ 22 วงจรที่ใช้ในการคำนวณหาค่า  $R_1, R_2$

$$10 - VR_2 - VR_1 - 0.7 = 6.5$$

$$VR_1 + VR_2 = 10 - 6.5 - 0.7$$

$$R_1 + R_2 = \frac{2.8}{IB} ; IB = \frac{IC}{B} = \frac{25mA}{70} = 0.35 mA$$

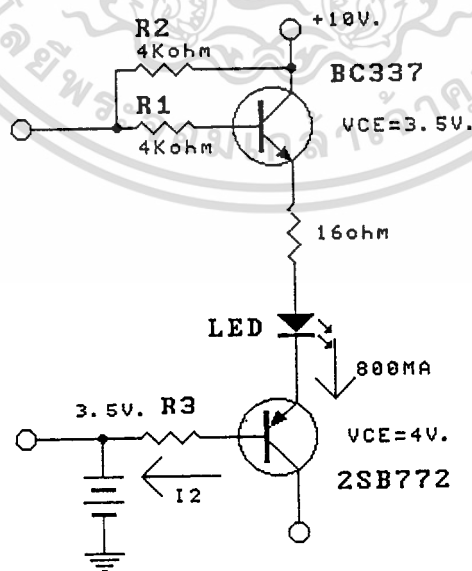
$$= \frac{2.8}{0.35mA}$$

$$= 8 \text{ K}\Omega$$

กำหนด  $R_1=R_2 \therefore R_1=R_2=4 \text{ K}\Omega$  #

หา  $R_3$  โดยใช้วงจรดังรูปที่ 23

กำหนด  $\beta=600 \therefore I_2 = \frac{800mA}{600} = 1.33mA$  (2SB772)



รูปที่ 23 วงจรที่ใช้ในการคำนวณหาค่า  $R_3$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$IR_2 = IB = 0.35 \text{ mA}$$

$$VR_2 = IB \times R_2 = 0.35 \text{ mA} \times 4 \text{ K} = 1.4 \text{ V}$$

$$\therefore V_1 = 10 - 1.4 = 8.6 \text{ V}$$

loop1

$$V_1 - IBR_1 - 0.7 - 0.4 - 2.1 - 0.7 - I_2R_3 = 0.35$$

$$\begin{aligned} R_3 &= \frac{V_1 - IBR_1 - 0.7 - 0.4 - 2.1 - 0.7 - 0.35}{I_2} \\ &= \frac{8.6 - 1.4 - 0.7 - 0.4 - 2.1 - 0.7 - 0.35}{1.33 \text{ mA}} \\ &= \frac{2.95}{1.33 \text{ mA}} = 2218 \text{ } \Omega \quad \# \end{aligned}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

#### การทดลองภาค scan

1. ต่อวงจร scan ที่ประกอบด้วย U1,U4,U5
2. ป้อน logic เข้าที่ขา PA0-PA4 ดังตารางที่ 8 แล้ววัดสัญญาณ output ของ U4 และ U5 ตารางที่ 8

HEX	INPUT					OUTPUT U4								OUTPUT U5							
	PA4	PA3	PA2	PA1	PA0	row1	row2	row3	row4	row5	row6	row7	row8	row9	row10	row11	row12	row13	row14	row15	row16
0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
4	0	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
5	0	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
6	0	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
7	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
8	0	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
9	0	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
A	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
B	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
C	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
D	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
E	0	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
F	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

สถานะ "1" คือแรงดัน 5V. (High level)

สถานะ "0" คือแรงดัน 0V. (Low level)

จากการทดลองนี้ output ROW1 - ROW16 จะได้ สัญญาณสแกนเป็น Low Level หรือ Logic "0" เพราะฉะนั้นจึงนำสัญญาณ Low level นี้ไป Bias ให้แก่ Transistor ชนิด PNP ต่อไป

### การทดลองภาค Decode Port Latch DATA

1. ต่อวงจรทดลองที่ประกอบด้วย U3,U6 (ดูรูปวงจร Decode Display ประกอบ)
2. ป้อน Logic เข้าที่ขา ADB4-ADB7 ดังตารางที่ 9 แล้ววัดสัญญาณ output ของ U6 (74LS138)  
ตารางที่ 9

HEX	INPUT						OUTPUT U6 (74LS138)							
	ADB7	ADB6	ADB5	ADB4	IORQ	WR	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
00-0F	0	0	0	0	0	0	0	1	1	1	1	1	1	1
10-1F	0	0	0	1	0	0	1	0	1	1	1	1	1	1
20-2F	0	0	1	0	0	0	1	1	0	1	1	1	1	1
30-3F	0	0	1	1	0	0	1	1	1	0	1	1	1	1
40-4F	0	1	0	0	0	0	1	1	1	1	0	1	1	1
50-5F	0	1	0	1	0	0	1	1	1	1	1	0	1	1
60-6F	0	1	1	0	0	0	1	1	1	1	1	1	0	1
70-7F	0	1	1	1	0	0	1	1	1	1	1	1	1	0
80-8F	1	0	0	0	0	0	1	1	1	1	1	1	1	1
90-9F	1	0	0	1	0	0	1	1	1	1	1	1	1	1
A0-AF	1	0	1	0	0	0	1	1	1	1	1	1	1	1
B0-BF	1	0	1	1	0	0	1	1	1	1	1	1	1	1
C0-CF	1	1	0	0	0	0	1	1	1	1	1	1	1	1
D0-DF	1	1	0	1	0	0	1	1	1	1	1	1	1	1
E0-EF	1	1	1	0	0	0	1	1	1	1	1	1	1	1
F0-FF	1	1	1	1	0	0	1	1	1	1	1	1	1	1

สถานะ "1" คือแรงดัน 5V. (High level)

สถานะ "0" คือแรงดัน 0V. (Low level)

3. ต่อวงจรทดลองที่ประกอบด้วย U3, U7 (หรือ U8)
4. ป้อน Logic เข้าที่ขา ADB0-ADB3 ดังตารางที่ 10 แล้ววัดสัญญาณ output ของ U7 (หรือ U8)  
#74LS154 โดยต่อขา G1, G2 ลงกราวด์ไว้

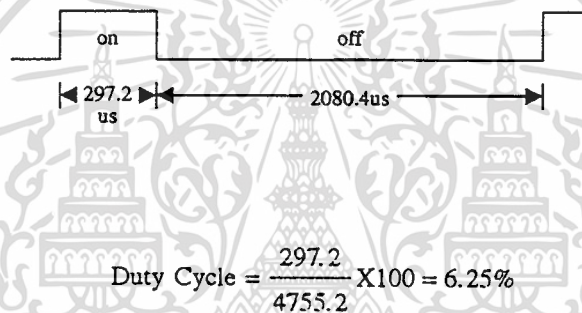
ตารางที่ 10

HEX	INPUT				OUTPUT U7 (74LS154)																
	ADB3	ADB2	ADB1	ADB0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12	Y13	Y14	Y15	
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	0	1	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	0	0	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
5	0	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1
6	0	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1	1
7	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1
A	1	0	1	0	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
B	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
C	1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1
D	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1
E	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### การทดลองวงจรภาค Drive LED

- 1.) ป้อน DATA "1" ที่ input latch DATA (74LS374) ทุกขา
- 2.) นำสัญญาณพัลส์ตั้งรูปป้อนเข้าที่วงจร Drive LED (ด้าน scan row)
- 3.) สังเกตความสว่างของ LED ปรากฏว่ายังไม่สว่างเท่าที่ต้องการ ดังนั้นจึงเปลี่ยนค่า R ที่ไบอัสทรานซิสเตอร์เป็นดังนี้ R1, R2 จากเดิม 4 K $\Omega$  เป็น 2.2 K $\Omega$
- 4.) จากนั้นทำการทดลองซ้ำข้อ 1.) และ 2.) แล้วสังเกตความสว่างของ LED ปรากฏว่าได้ความสว่างเป็นที่น่าพอใจ
- 5.) วัดกระแสที่จ่ายเข้าไปใน Board ขนาด 16X16 DOT (ให้สว่างทั้งสองสี จะได้เป็นสีส้ม) กระแสที่วัดได้มีค่าเท่ากับ 0.38 Amp



รูปที่ 24 รูปสัญญาณพัลส์ที่ใช้ในการทดลอง

## บทที่ 5

### บทสรุปและวิจารณ์ผล

ในโครงการนี้มีปัญหาเกิดขึ้นหลายอย่างด้วยกันคือ ปัญหาแรก Board JR-180 ที่นำมาใช้นั้น DATA ของ D6 มีค่าเป็น "1" ตลอด ทำให้ DATA Error จึงทำการตรวจสอบพบว่า ขา VCC และ D6 สลับกัน เนื่องจากคู่มือการใช้งาน Z180 บอกรหัสผิดจึงทำการสลับขากลับคืน

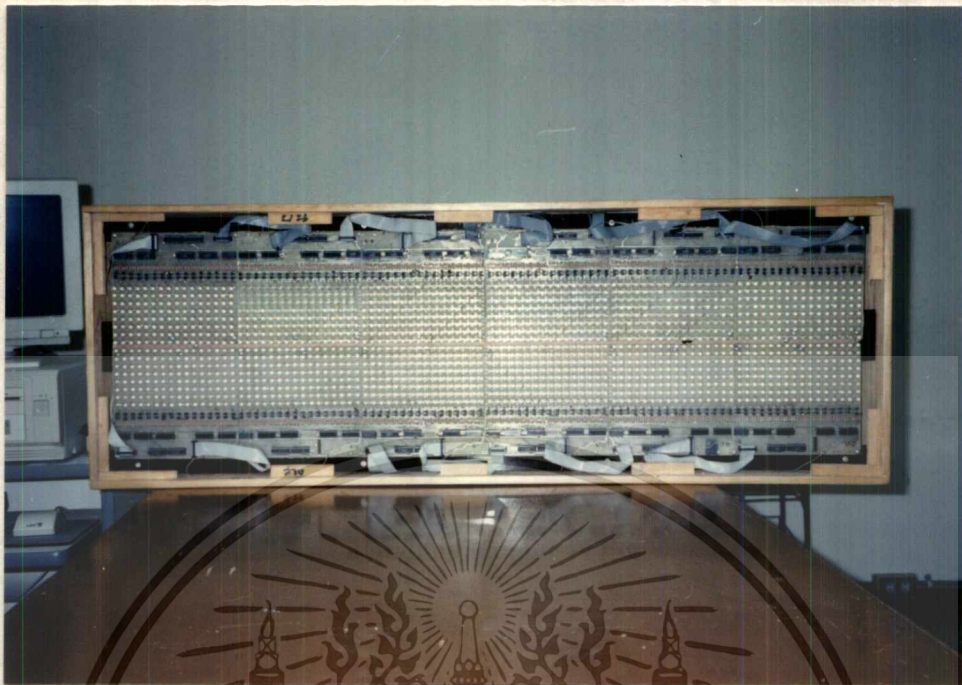
คาร์ซีสเตอร์ในภาค Drive (ขณะยังไม่มี R2) ที่ได้จากการคำนวณนั้นจะทำให้ LED สว่างไม่เต็มที่ แก้ไขโดย การเปลี่ยนคาร์ซีสเตอร์ไปอีตัวใหม่ โดยลดค่า R1 และ R2 เพื่อให้มีกระแสเบสเพิ่มขึ้นอีก

การออกแบบแผง LED ส่วนบน (ขนาด 8X16) และส่วนล่าง (ขนาด 8X16) โดย LED และภาคขยายกำลังอยู่บนแผ่นปริ้นท์เดียวกัน แล้วนำมาต่อกันเป็นขนาด 16X16 นั้น จะมีปัญหาในการจัดวางแผงให้หลอด LED ตรงกัน อาจแก้ไขโดยให้ LED อยู่บนแผงเดียวกัน (ขนาด 16X16) แล้วแยกภาคขยายออกมาต่างหาก แต่วิธีนี้จะทำให้เกิดความยุ่งยากในการเชื่อมต่อสายไฟระหว่างแผง LED และแผงภาคขยาย

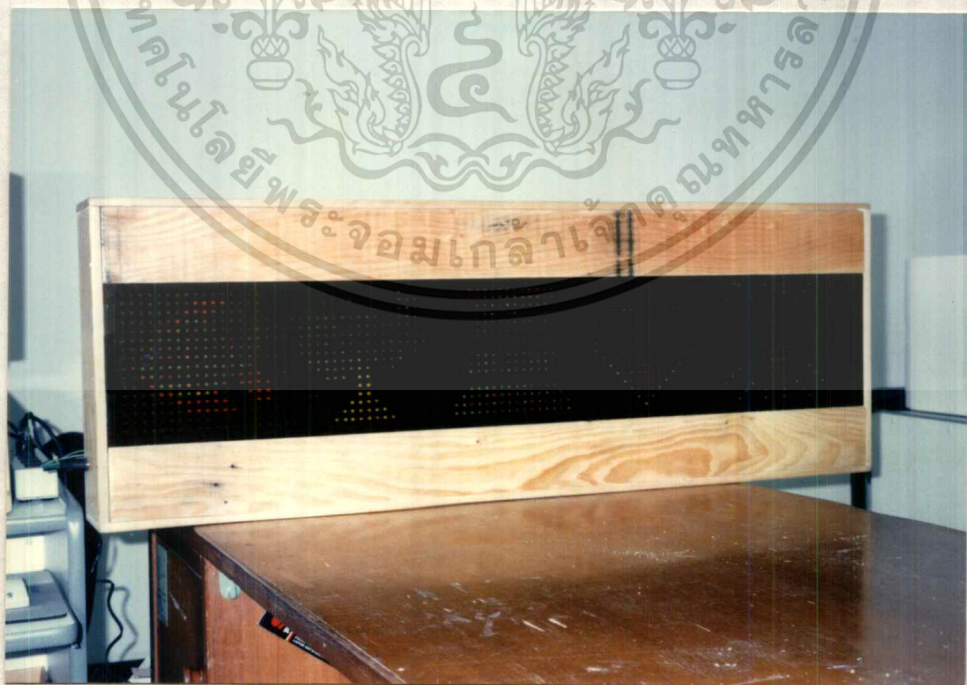
Program Monitor ที่เขียนขึ้นมานั้น ยังมีจุดเสีย (bug) อีกหลายจุดด้วยกัน จึงอยากให้ผู้ที่สนใจโครงการนี้มาพัฒนา Software เพิ่มเติมอีกต่อไป สำหรับจุดที่ควรจะพัฒนาต่อก็คือ ขยาย Port ต่อ Keyboard IBM, เพิ่มตัวอักษร กราฟฟิก, เพิ่มรูปแบบการเลื่อนของข้อความ, ปรับปรุงความสว่างของ LED โดยออกแบบการ Bias ใหม่, มีระบบ Backup ข้อมูลที่ยาวนานขึ้น โดยใช้ถ่าน Rechargeable Battery, ออกแบบระบบลดสัญญาณรบกวนที่ออกไปสู่ภายนอก ซึ่งจะทำให้โครงการนี้มีความสามารถเพียงพอที่นำเข้าไปในเชิงธุรกิจได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

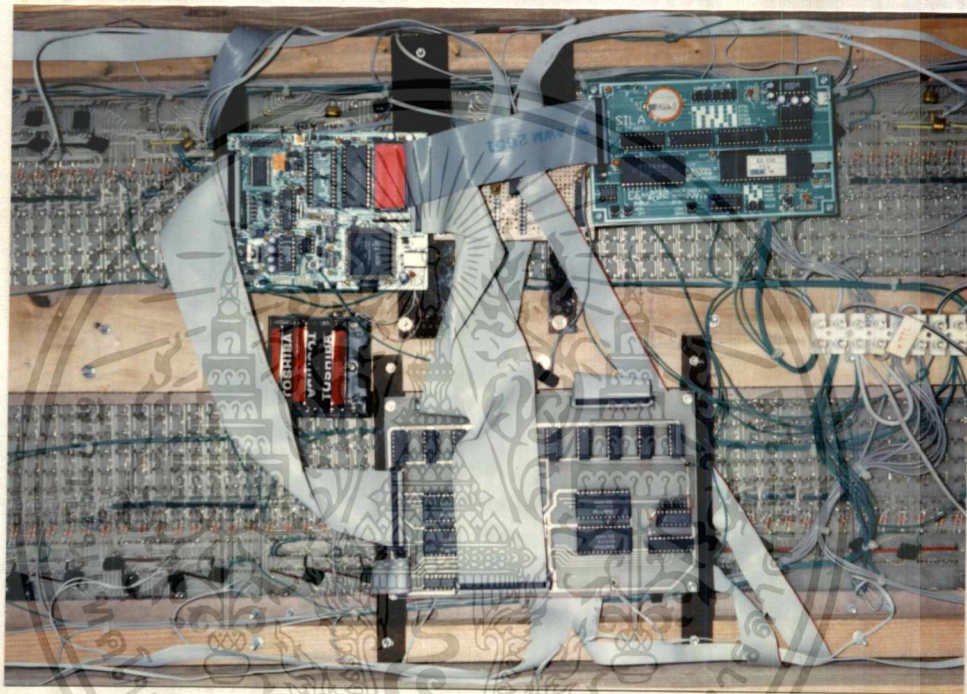


รูปที่ 25 แผงแสดงผลแบบ 3 สี (ด้านหน้า)



รูปที่ 26 แผงแสดงผลแบบ 3 สี (หลังจากปิดแผ่นกรองแสงแล้ว)

เอกสารนี้เป็นเอกสารที่สวทช. ได้รับความไว้วางใจให้ดำเนินการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 27 แผงวงจรควบคุมใช้ CPU Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น. อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:FILENAME:      01.ASM
:DESCRIPTON:   Monitor Program for 3 Color LED Display Board (16X96 dot)
:HARDWARE:     JR-180 (CPU Z80180 12.288MHz)
:ASM:          Cross-Assembler Level II (a180z.exe)
:START_DATE:   22 November 1994

.command +ai +l
;(+l Turn on off Listing File)

```

```

.opdef EQU,.EQU
.opdef DS,.RS
.opdef DB,.DB
.opdef ORG,.ORG
.opdef DRW,.DRW

```

```

EQU NUMROW ,16
EQU NUMCOL ,12
EQU NUMPAGE ,10

```

```

EQU GEN,05000H ;NOTE RAM 32K = 8000h-FFFFh
EQU RAM1,08000H
EQU COMMON1,0E000H

```

```

EQU TDR0 ,06H+40H ;ASCII Channel 0

```

```

EQU RDR0 ,08H+40H

```

```

EQU STAT0 ,04H+40H

```

```

EQU CNTLA0 ,00H+40H

```

```

EQU CNTLB0 ,02H+40H

```

```

EQU TDR1 ,07H+40H ;ASCII Channel 1

```

```

EQU RDR1 ,09H+40H

```

```

EQU STAT1 ,05H+40H

```

```

EQU CNTLA1 ,01H+40H

```

```

EQU CNTLB1 ,03H+40H

```

```

EQU TMDROL ,0CH+40H ;Programmable Reload Timer

```

```

EQU TMDROH ,0DH+40H

```

```

EQU RLDR0L ,0EH+40H

```

```

EQU RLDR0H ,0FH+40H

```

```

EQU TCR ,10H+40H

```

```

EQU TMDR1L ,14H+40H

```

```

EQU TMDR1H ,15H+40H

```

```

EQU RLDR1L ,16H+40H

```

```

EQU RLDR1H ,17H+40H

```

```

EQU SAR0L ,20H+40H ;DMA

```

```

EQU SAR0H ,21H+40H

```

```

EQU SAR0B ,22H+40H

```

```

EQU DAR0L ,23H+40H

```

```

EQU DAR0H ,24H+40H

```

```

EQU DAR0B ,25H+40H

```

```

EQU BCR0L ,26H+40H

```

```

EQU BCR0H ,27H+40H

```

```

EQU DSTAT ,30H+40H

```

```

EQU DMODE ,31H+40H

```

```

EQU DCNTL ,32H+40H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EQU IL      , 33H+40H    ;Interrupt Vector Low Register
EQU ITC     , 34H+40H    ;INT/TRAP Control Register
EQU OMCR    , 3EH+40H    ;Operation Mode Control Register
EQU RCR     , 36H+40H    ;Refresh Control Register
EQU CBAR    , 3AH+40H    ;COMMON Bank Area Register
EQU BBR     , 39H+40H    ;BANK Base Register
EQU CBR     , 38H+40H    ;COMMON Base Register
EQU ICR     , 3FH        ;I/O Control Register

```

```
;english font width
```

```

EQU WFONT0  , 10
EQU WFONT1  , 17
EQU WFONT2  , 10
EQU WFONT3  , 17
EQU WFONT4  , 11
EQU WFONT5  , 21
EQU WFONT6  , 11
EQU WFONT7  , 21

```

```
;thai font width
```

```

EQU WFONT8  , 10
EQU WFONT9  , 18
EQU WFONTA  , 10
EQU WFONTB  , 18
EQU WFONTC  , 11
EQU WFONTD  , 22
EQU WFUNTE  , 11
EQU WFONTF  , 22

```

```
;graphic font width
```

```

EQU WFONT10 , 24
EQU WFONT11 , 24
EQU WFONT12 , 24

```

```
;command font width
```

```
EQU WFONT13 , 8
```

```
EQU STACK , 0FFF0H ;if RAM=32K,then STACK=0FFF0h
```

```
EQU VELOW , 80H
```

```
EQU PORTA , 80H
```

```
EQU PCTRL , 83H
```

```
EQU RED1 , 020H
```

```
EQU RED2 , 030H
```

```
EQU GRN1 , 000H
```

```
EQU GRN2 , 010H
```

```
EQU TIMELO , 050H ;:060H
```

```
EQU TIMEHO , 001H ;:001H
```

```
EQU TIMEL1 , 0
```

```
EQU TIMEH1 , 0
```

```
EQU DISPAUSE , 6
```

```
;Origin generate fon is define space for use as variable only
```

```
ORG GEN
```

```
ENGFONT1: DS 24*96 ; = (24 byte) * 95 character
```

```
ENGFONT2: DS 48*96 ; = (48 byte) * 95 character
```

```
THAIFON1: DS 24*75 ; = (24 byte) * 68 character
```

```
GRAPFON1: DS 64*16
```

```
ORG RAM1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EDBUF: DS NUMROW\*NUMCOL\*NUMPAGE  
 PAPER: DS 100  
 PAUSEN: DS 1  
 DSPCOLR: DS 2  
 TKN\_BUF: DS 200  
 TKN\_END:  
 EQU TXTBUF,EDBUF

ORG COMMON1  
 DSPBUFR: DS NUMROW\*NUMCOL  
 DSPBUFG: DS NUMROW\*NUMCOL  
 MEMBUF: DS NUMROW\*NUMCOL  
 PREDSPR: DS NUMROW\*NUMCOL  
 PREDSPG: DS NUMROW\*NUMCOL  
 GEN\_BUF1: DS NUMROW\*3  
 GEN\_BUF2: DS NUMROW\*3  
 DSPBUF2: DS NUMROW\*NUMCOL  
 PREDSP2: DS NUMROW\*NUMCOL\*2  
 DSPSIZE: DS 2  
 X\_STDSP: DS 1  
 HALFCOL: DS 1  
 MAXCOL: DS 1  
 MAXDOT: DS 2  
 CHKLOCK: DS 1  
 ADDSCANR: DS 2  
 ADDSCANG: DS 2  
 ROW: DS 1  
 PAGELP: DS 1  
 BLANK: DS 1  
 LENBUF: DS 1  
 NAMEBUF: DS 2  
 NAMEGEN: DS 2  
 X\_STPLT: DS 2  
 Y\_STPLT: DS 1  
 Y\_START: DS 1  
 YPOS: DS 1  
 X0\_FLAG: DS 1  
 X\_START: DS 2  
 XPOS: DS 2  
 PL\_DATA: DS 1  
 PL\_COLR: DS 2  
 CL\_BUF: DS 1  
 ASCBUF: DS 8  
 CH\_ASC: DS 1  
 CH\_TYPE: DS 1  
 CH\_ROW: DS 1  
 CH\_COL: DS 1  
 CH\_MODE: DS 1  
 CH\_FONT: DS 1  
 PL\_LOOP: DS 1  
 PL\_PTR: DS 2  
 MODE: DS 1  
 DISDLY: DS 1  
 RUN\_CMD: DS 1  
 RUN\_PTR: DS 1  
 T\_ASC: DS 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

G_ASC:    DS 1
SLONG:    DS 1
SHFBYTE:  DS 1
TKN_PTR:  DS 2
MOVEBLOCK: DS 2
END:

```

```

ORG 00H
XOR A     ;***** POWER UP DELAY *****
FIRST: DEC A
JR NZ,FIRST
JP START

```

```

:ignore RESTART and NMI

```

```

ORG 08H
RETI
ORG 010H
RETI
ORG 018H
RETI
ORG 020H
RETI
ORG 028H
RETI
ORG 030H
RETI
ORG 038H
RETI
ORG 066H
RETN

```

```

ORG VELOW
INT1: DRW EMPTY
INT2: DRW EMPTY
PRT0: DRW PRT0INT
PRT1: DRW EMPTY ;PRT1INT
DMA0: DRW EMPTY
DMA1: DRW EMPTY
CSIO: DRW EMPTY
ASCII0: DRW EMPTY ;ASCII0INT
ASCII1: DRW EMPTY

```

```

EMPTY: EI
RETI     ;ignore all internal interrupt except PRT.ASCII

```

```

;***** START PROGRAM BY Off LED *****

```

```

START: LD  A,80H    ;set #8255
        OUT (PCTRL),A
        LD  A,17    ;OFF row port by send over
        OUT (PORTA),A

```

```

;**** INITIAL PARAMETER ****

```

```

INIT:  IN0 A,(ICR)  ;Relocate I/O from 00-3F to 40-7F
        OR  01000000B
        OUT0 (ICR),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INO  A,(ITC)
AND  01111000B ;clear trap after reset
OUT0 (ITC),A

DI
INO  A,(DCNTL) ;mem wait state = 0
AND  00001111B ;I/O wait state = 1
OUT0 (DCNTL),A

LD   A,0F8H
OUT0 (CBAR),A ;set Logical address
LD   A,0 ;select Page #1
OUT0 (BBR),A ;physical bank 8000H
OUT0 (CBR),A ;physical common1 area F000H

LD   SP,STACK ;set stack address FFF0H

LD   A,VEFLOW ;set LOW VECTOR
OUT0 (IL),A
CALL ASCOSET

LD   A,40H ;for Z80180 operation
OUT0 (OMCR),A

LD   A,0
OUT0 (RCR),A ;not refresh dynamic ram

LD   A,0 ;TIF is clear to 'V' during RESET
OUT0 (TCR),A ;clear timer PRT0,PRT1

LD   HL,COMMON1 ;clear COMMON1 to ENDCOMMON1
LD   BC,END-COMMON1
INIT_11: LD  A,0
        LD  (HL),A
        INC HL
        DEC BC
        LD  A,B
        OR  C
        JR  NZ,INIT_11

;Clear GREEN port for test program on hardware scan 3 color
LD   B,12
LD   C,GRN1
LD   A,0
INIT_12: OUT (C),A
        INC C
        DJNZ INIT_12
LD   B,12
LD   C,GRN2

INIT_13: OUT (C),A
        INC C
        DJNZ INIT_13

;Clear RED port for test program on hardware scan 3 color
LD   B,12
LD   C,RED1
LD   A,0

```

```

INIT_14: OUT (C),A
         INC C
         DJNZ INIT_14
         LD B,12
         LD C,RED2
INIT_15: OUT (C),A
         INC C
         DJNZ INIT_15

INIT_3:  LD A,6
         LD (HALFCOL),A
         LD A,12
         LD (MAXCOL),A
         LD A,0
         LD (X_STDSP),A
         LD HL,96
         LD (MAXDOT),HL

INIT_5:  LD BC,NUMROW*NUMCOL*2 ;TWO COLOR cause to multiply by 2
         LD (DPSIZE),BC

         EI
INIT_6:  CALL SETPRTO

;MAIN PROGRAM
CALL CLRDSP
CALL ONSCAN

MAIN:   LD A,22H
         LD (RUN_CMD),A
         CALL VELOCITY

         LD A,(MAXCOL)
         LD (LENBUF),A

         LD IX,TKN_BUF
         INC IX
         INC IX
         INC IX
         INC IX
         LD (RUN_PTR),IX
         LD (PL_PTR),IX

PLAY:

         CALL LDPRE4
         CALL LDDSPBUF
         CALL PAUSE
         CALL CRPREDSP

         CALL LDPRE0
         CALL SHIFT_S
         CALL CRPREDSP

         LD A,28H
         LD (RUN_CMD),A
         CALL PAUSECON

```

เอกสารนี้เป็นเอกสารที่ สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL PAUSECON

CALL LDPRE8

CALL SHIFT\_N

CALL CRPREDSP

CALL LDPRE6

CALL SHIFT\_W

CALL CRPREDSP

CALL LDPRE9

CALL SHIFT\_E

CALL CRPREDSP

CALL LDPRES

CALL ROTIN\_N

CALL CRPREDSP

CALL LDPRE11

CALL ROTIN\_W

CALL CRPREDSP

CALL LDPRE7

CALL ROTIN\_E

CALL CRPREDSP

CALL LDPRE10

CALL ROTIN\_S

CALL CRPREDSP

CALL LDPRE12

CALL OPEN\_H

CALL CRPREDSP

CALL LDPRE13

CALL CLOSE\_H

CALL CRPREDSP

CALL LDPRE1

CALL SHIFT\_W

CALL CRPREDSP

CALL LDPRE2

CALL OPEN\_H

CALL CRPREDSP

CALL LDPRE15

CALL SHIFT\_N

CALL CRPREDSP

CALL LDPRE16

CALL CLOSE\_H

CALL CRPREDSP

CALL LDPRE17

CALL RANDOM

CALL CRPREDSP

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL LDPREI8
CALL MIXT_E
CALL CRPREDSPP
```

```
CALL LDPREI4
CALL RANDOM
CALL CRPREDSPP
```

```
CALL LDPRE3
CALL MIXT_E
CALL CRPREDSPP
```

```
JP PLAY
```

```
LDPREDSPP: PUSH HL
            PUSH BC
            PUSH DE
            LD HL,DSPBUFR
            LD DE,PREDSPR
            LD BC,NUMROW*NUMCOL*2
            LDIR
            POP DE
            POP BC
            POP HL
            RET

LDDSPBUFR: PUSH HL
            PUSH BC
            PUSH DE
            LD HL,PREDSPR
            LD DE,DSPBUFR
            LD BC,NUMROW*NUMCOL*2
            LDIR
            POP DE
            POP BC
            POP HL
            RET
```

```
CRPREDSPP: PUSH HL
            PUSH BC
            PUSH DE
            LD HL,PREDSPR
            LD DE,PREDSPR+1
            LD A,0
            LD (HL),A
            LD BC,NUMROW*NUMCOL
            LDIR
            JR CLRPRE_1
```

```
CLRPREG:  PUSH HL
            PUSH BC
            PUSH DE

CLRPRE_1: LD HL,PREDSPG
            LD DE,PREDSPG+1
            LD A,0
            LD (HL),A
            LD BC,NUMROW*NUMCOL
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LDIR
JR CLRGEN_1

CLRGEN: PUSH HL
        PUSH BC
        PUSH DE
CLRGEN_1: LD HL,GEN_BUF1
          LD DE,GEN_BUF1+1
          LD A,0
          LD (HL),A
          LD BC,NUMROW*3*2
          LDIR
          POP DE
          POP BC
          POP HL
          RET

DELAY:  LD D,10
DELAY1: LD BC,20000
DELAY_1: DEC BC
        LD A,B
        OR C
        JR NZ,DELAY_1
        DEC D
        JR NZ,DELAY1
        RET

;*****
; GETGEN : read data from charactor gen. send to GEN_BUF
; INPUT  : CH_ASC,CH_TYPE
; OUTPUT : CH_MODE 0=ENG,1=THAI,2=GRAPHIC,3=SUPER GRAPH,4=COMMAND
;         CH_FONT 0=8X8,1=8X16,2=16X2,3=16X16,4=21X24 (HIGH X WIDTH)
;         S=24X24GRAPHIC,6=24X8COM

GETGEN: PUSH BC
        PUSH DE
        PUSH HL
        PUSH IX

        LD A,I
        LD (PL_LOOP),A

        LD A,(CH_TYPE)
        CP 07H
        JP Z,GETG_E ;ENGLISH = 0-7
        JP C,GETG_E
        CP 0FH
        JP Z,GETG_T ;THAI = 8-0F
        JP C,GETG_T
        CP 12H
        JP Z,GETG_GR ;GRAPH = 10-12
        JP C,GETG_GR

GETG_E: LD A,0
        LD (CH_MODE),A ;CH_MODE=ENGLISH
        LD A,(CH_TYPE) ;load char type

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR Z,GETG_E1
JR C,GETG_E1
LD HL,ENGFONT2
SUB 04H
LD (CH_FONT),A ;CH_FONT=4 -> 7
JR GETG_E3
GETG_E1: LD (CH_FONT),A ;CH_FONT=0 -> 3
LD HL,ENGFONT1
GETG_E2: LD A,(CH_ASC) ;find offset address of that char
SUB 20H
LD B,A
LD C,24
MLT BC ;save offset address in BC
ADD HL,BC ;point to start address of char gen
LD BC,24
CALL LDCHAR
JP GETG_END

```

```

GETG_E3: LD A,(CH_ASC) ;find offset address of that char
SUB 20H
LD B,A
LD C,48
MLT BC ;save offset address in BC
ADD HL,BC ;point to start address of char gen
LD BC,48
CALL LDCHAR
JP GETG_END

```

```

GETG_T: LD A,1
LD (CH_MODE),A ;CH_MODE=THAI
LD A,(CH_TYPE)
SUB 08H
CP 3
JR Z,GETG_F1
JR C,GETG_F1
SUB 04H

```

```

GETG_F1: LD (CH_FONT),A ;CH_FONT=4 -> 7
LD HL,THAIFONT1 ;CH_FONT=0 -> 3

```

```

GETG_F2: LD A,(CH_ASC) ;convert to thai ascII
SUB 0A0H
LD (T_ASC),A

```

```

CP 2FH ;SARA SURM
JP Z,GETG_T1
CP 32H ;SARA O
JP Z,GETG_T2
CP 33H ;SARA I MIMUAN
CP Z,GETG_T2
CP 34H ;SARA I MIMALAI
JP Z,GETG_T2
CP 08H ;YOR YING
JP Z,GETG_T3
CP 0CH ;DOOR CHADA
CP Z,GETG_T3
CP 0DH ;TOR PATAK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JP Z.GETG_T3
CP 0EH ;THOR SONTAN
CP Z.GETG_T3
CP' 22H ;ROR RUE
JP Z.GETG_T4
CP 37H ;SARA U
JP Z.GETG_T5
CP 38H ;SARA UU
JP Z.GETG_T5
JP NC.GETG_T6 ;>38H up CHAR (> 217 decimal)
JP GETG_T9

```

GETG\_T1: PUSH HL

```

LD A,2EH ;sara R
LD (T_ASC),A
CALL GETG_TL2
POP HL
LD A,3EH ;sara um
LD (T_ASC),A
CALL GETG_TL1
LD A,7
LD (PL_LOOP),A
JP GETG_END

```

GETG\_T2: PUSH HL

```

CALL GETG_TL2 ;SARA I,O
POP HL
LD A,045H ;243 decimal
LD (T_ASC),A
CALL GETG_TL1
LD A,2 ;UP
LD (PL_LOOP),A
JP GETG_END

```

GETG\_T3: PUSH HL

```

CALL GETG_TL ;YO YING,DOOR CHADA,TOR PATAK,THOR SONTAN
LD A,(T_ASC)
ADD A,59
LD (T_ASC),A
POP HL
CALL GETG_TL2
LD A,3
LD (PL_LOOP),A
JP GETG_END

```

GETG\_T4: CALL GETG\_TL1 ;ROR RUE

```

LD A,04AH
LD (T_ASC),A
CALL GETG_TL2
LD A,3
LD (PL_LOOP),A
JP GETG_END

```

GETG\_T5: CALL GETG\_TL1 ;SARA U,UU

```

LD A,4
LD (PL_LOOP),A
JP GETG_END

```

```

GETG_T6: CALL GETG_TL1      ;UPPER CHAR

GETG_T65: LD  IX,(RUN_PTR)  ;if RUN mode ,IX point to run token pointer

GETG_T60: LD  A,(T_ASC)
          CP  046H
          JR  Z,GETG_T62    ;>=224 then check old char = upper char1?
          JR  NC,GETG_T62

GETG_T61: LD  A,5           ;upper char 1 st
          LD  (PL_LOOP),A
          JP  GETG_END

GETG_T62: LD  A,(IX-3)     ;check type is thai
          AND 00001111B
          CP  08H         ;>=4 continue check
          JR  Z,GETG_T63
          JR  NC,GETG_T63
          JR  GETG_T61

GETG_T63: CP  10H         ;>=9 then upper char 1 ;>=8
          JR  Z,GETG_T61
          JR  NC,GETG_T61
          LD  A,(IX-4)     ;< 9 then continue check
          CP  39H         ;>=217 then adjust for upper char 2
          JR  Z,GETG_T64
          JR  NC,GETG_T64
          JR  GETG_T61

GETG_T64: LD  A,6
          LD  (PL_LOOP),A
          JP  GETG_END

GETG_T9:  CALL GETG_TL1
          JP  GETG_END

GETG_TL1: LD  A,(T_ASC)
          LD  B,A
          LD  C,24
          MLT BC          ;save offset address in BC
          ADD HL,BC       ;point to start address of char gen
          LD  DE,GEN_BUF1
          LD  BC,24
          LDIR           ;load data to GEN_BUF
          RET

GETG_TL2: LD  A,(T_ASC)
          LD  B,A
          LD  C,24
          MLT BC          ;save offset address in BC
          ADD HL,BC       ;point to start address of char gen
          LD  DE,GEN_BUF2
          LD  BC,24
          LDIR           ;load data to GEN_BUF
          RET

GETG_GR:  LD  A,2
          LD  (CH_MODE),A ;CH_MODE=GRAPH
          LD  A,4
          LD  (CH_FONT),A ;CH_FONT=4

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,(CH_TYPE)
CP 10H
JR C,GETG_END
LD HL,GRAPFON1

GETG_GR3: LD A,(CH_ASC)
CP 90H
JR NC,GETG_END
CP 80H
JR C,GETG_END
SUB 80H
LD (G_ASC),A
LD A,9
LD (PL_LOOP),A
CALL GETG_GL
JR GETG_END

GETG_GL: LD A,(G_ASC)
LD B,A
LD C,64
MLT BC ;save offset address in BC
ADD HL,BC ;point to start address of char gen
LD DE,GEN_BUF1
LD BC,32
LDIR ;load data RED to GEN_BUF1

LD DE,GEN_BUF2
LD BC,32
LDIR ;load data GREEN to GEN_BUF2
RET

GETG_END: POP IX
POP HL
POP DE
POP BC
RET

include(01z.asm)
include(01s.asm)
include(01u.asm)
include(01r.asm)
include(01m.asm)
.END

```

```

:FILE NAME: 01Z.ASM
:DESCRIPTION: Simulate RUN Program,Process of ASCII character
:START DATE: 18 MARCH 1995

```

```

LDPRE0: LD A,0 ;LED 3 Color
LD (X_STPLT),A

```

```

LD HL,PREDSPR
LD (NAMEBUF),HL
LD IY,(PL_PTR)

```

```

LD (IY+0),04CH ;CH_ASC
LD (IY+1),002H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),10 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (IY+0),045H ;CH_ASC
LD (IY+1),002H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),10 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (IY+0),044H ;CH_ASC
LD (IY+1),002H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),10 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (IY+0),020H ;CH_ASC
LD (IY+1),002H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),5 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (IY+0),033H ;CH_ASC
LD (IY+1),002H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),9 ;CH_COL
CALL RUNPLOT

```

```

LD (IY+0),020H ;CH_ASC
LD (IY+1),002H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),5 ;CH_COL
CALL RUNPLOT

```

```

LD (IY+0),043H ;CH_ASC
LD (IY+1),002H ;CH_TYPE

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือสงวนลิขสิทธิ์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (1Y+2),1 :CH_ROW
LD (1Y+3),10 :CH_COL
CALL RUNPLOT
```

```
LD (1Y+0),06FH :CH_ASC
LD (1Y+1),002H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),9 :CH_COL
CALL RUNPLOT
```

```
LD (1Y+0),06CH :CH_ASC
LD (1Y+1),002H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),9 :CH_COL
CALL RUNPLOT
```

```
LD (1Y+0),06FH :CH_ASC
LD (1Y+1),002H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),9 :CH_COL
CALL RUNPLOT
```

```
LD (1Y+0),072H :CH_ASC
LD (1Y+1),002H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),8 :CH_COL
CALL RUNPLOT
```

```
RET
```

```
LDPREL:
```

```
:KMITL
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,0
LD (X_STPLT),A
LD (Y_PL_PTR)
```

```
LD (1Y+0),04BH :CH_ASC
LD (1Y+1),003H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),21 :CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (1Y+0),04DH :CH_ASC
LD (1Y+1),003H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),16 :CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (1Y+0),049H :CH_ASC
LD (1Y+1),003H :CH_TYPE
LD (1Y+2),1 :CH_ROW
LD (1Y+3),15 :CH_COL
LD A,0
LD (PL_COLR),A
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL RUNPLOT
```

```
LD (Y+0),054H ;CH_ASC
LD (Y+1),003H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),14 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),027H ;CH_ASC
LD (Y+1),003H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),04CH ;CH_ASC
LD (Y+1),003H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),15 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
RET
```

```
LDPRE2:
```

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,0
LD (X_STPLT),A
LD (Y,(PL_PTR))

LD (Y+0),065H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),06CH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),065H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),10 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

เอกสารนี้เป็นเอกสารที่สวจนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (Y+0),063H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),074H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),072H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),06FH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),10 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),06EH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),069H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),7 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),063H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (Y+0),073H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT

```

```
RET
```

```
LDPRE3: ;glass,tree,clock,tree,fish
```

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,3
LD (X_STPLT),A
LD Y,(PL_PTR)

```

```
LD (Y+0),08DH ;CH_ASC
LD (Y+1),012H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),17 ;CH_COL
CALL RUNPLOT

```

```
LD (Y+0),086H ;CH_ASC
LD (Y+1),012H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),18 ;CH_COL
CALL RUNPLOT

```

```
LD (Y+0),082H ;CH_ASC
LD (Y+1),012H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),19 ;CH_COL
CALL RUNPLOT

```

```
LD (Y+0),08CH ;CH_ASC
LD (Y+1),012H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),17 ;CH_COL
CALL RUNPLOT

```

```
LD (Y+0),085H ;CH_ASC
LD (Y+1),012H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),17 ;CH_COL
CALL RUNPLOT

```

```
LD HL,GEN_BUF1
LD DE,GEN_BUF1+1
LD A,0
LD (HL),A
LD BC,NUMROW*3*2
LDIR

```

```
RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD HL,PREDSPR
LD A,0
LD (X_STPLT),A
LD (Y,(PL_PTR)

LD (Y+0),080H ;CH_ASC
LD (Y+1),010H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),17 ;CH_COL
CALL RUNPLOT
```

```
LD (Y+0),0C7H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0C4H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0DDH ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0C7H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0B2H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0DAH ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
```

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL RUNPLOT

```
LD (Y+0),020H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),3 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0A3H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0C1H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0DDH ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0B8H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),11 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),021H ;CH_ASC
LD (Y+1),001H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),7 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

RET

```
LDPRES: ;SURAT,APPLE
LD HL,PRESR
LD (NAMEBUF),HL
LD A,4
LD (X_STPLT),A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD IY,(PL_PTR)
```

```
LD (IY+0),0C7H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),1 ;CH_ROW
LD (IY+3),17 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0D7H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),17 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0C1H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),18 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0DDH ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),18 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0B3H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),17 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0B7H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),19 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0E4H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),19 ;CH_COL
LD A,1
```

```
LD (PL_COLR),A
CALL RUNPLOT
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (Y+0),084H ;CH_ASC
LD (Y+1),010H ;CH_TYPE
LD (Y+2),0 ;CH_ROW
LD (Y+3),17 ;CH_COL
CALL RUNPLOT

```

```
RET
```

```

LDPRE6: ;CHAIWAT
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,4
LD (X_STPLT),A
LD Y,(PL_PTR)

```

```

LD (Y+0),0A8H ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),1 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0DDH ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0C0H ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0C4H ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0DDH ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

```

```
LD (Y+0),0B0H ;CH_ASC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),18 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0B7H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),18 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),0E4H ;CH_ASC
LD (IY+1),009H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),17 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
RET
```

```
LDPRE7: ;SURANART
```

```
LD HL,PREDSPR
```

```
LD (NAMEBUF),HL
```

```
LD A,5
```

```
LD (X_STPLT),A
```

```
LD IY,(PL_PTR)
```

```
LD (IY+0),0C7H ;CH_ASC
```

```
LD (IY+1),009H ;CH_TYPE
```

```
LD (IY+2),1 ;CH_ROW
```

```
LD (IY+3),17 ;CH_COL
```

```
LD A,2
```

```
LD (PL_COLR),A
```

```
CALL RUNPLOT
```

```
LD (IY+0),0D7H ;CH_ASC
```

```
LD (IY+1),009H ;CH_TYPE
```

```
LD (IY+2),2 ;CH_ROW
```

```
LD (IY+3),17 ;CH_COL
```

```
LD A,2
```

```
LD (PL_COLR),A
```

```
CALL RUNPLOT
```

```
LD (IY+0),0C1H ;CH_ASC
```

```
LD (IY+1),009H ;CH_TYPE
```

```
LD (IY+2),2 ;CH_ROW
```

```
LD (IY+3),18 ;CH_COL
```

```
LD A,2
```

```
LD (PL_COLR),A
```

```
CALL RUNPLOT
```

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),16 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0CEH ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),15 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),0B5H ;CH_ASC
LD (Y+1),009H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
RET
```

```
LDPRE3:
```

```
;JUD THUM DOI
```

```
LD HL,PREDSPR
```

```
LD (NAMEBUF),HL
```

```
LD A,7
```

```
LD (X_STPLT),A
```

```
LD Y,(PL_PTR)
```

```
LD (Y+0),0A6H ;CH_ASC
```

```
LD (Y+1),008H ;CH_TYPE
```

```
LD (Y+2),2 ;CH_ROW
```

```
LD (Y+3),9 ;CH_COL
```

```
LD A,1
```

```
LD (PL_COLR),A
```

```
CALL RUNPLOT
```

```
LD (Y+0),0DDH ;CH_ASC
```

```
LD (Y+1),008H ;CH_TYPE
```

```
LD (Y+2),2 ;CH_ROW
```

```
LD (Y+3),9 ;CH_COL
```

```
LD A,1
```

```
LD (PL_COLR),A
```

```
CALL RUNPLOT
```

```
LD (Y+0),0B2H ;CH_ASC
```

```
LD (Y+1),008H ;CH_TYPE
```

```
LD (Y+2),2 ;CH_ROW
```

```
LD (Y+3),9 ;CH_COL
```

```
LD A,1
```

```
LD (PL_COLR),A
```

```
CALL RUNPLOT
```

```
LD (Y+0),0B5H ;CH_ASC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0CFH ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),6 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0D2H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0B2H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),0C0H ;CH_ASC
LD (Y+1),008H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),020H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),5 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT

```

```

LD (Y+0),08EH ;CH_ASC
LD (Y+1),010H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),16 ;CH_COL
CALL RUNPLOT

```

```
RET
```

LDPRE9: :NUMBER\_35103178

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (NAMEBUF),HL
LD A,5
LD (X_STPLT),A
LD (Y,(PL_PTR)
```

```
LD (Y+0),033H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),052H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),020H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),5 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),033H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),035H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),031H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),6 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),030H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
```

```
LD (Y+3),9 ;CH_COL
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,0
LD (PL_COLR),A
CALL RUNPLOT

LD (IY+0),033H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

LD (IY+0),031H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),6 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

LD (IY+0),037H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

LD (IY+0),038H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT

RET

LDPRE10: ;NUMBER 35103214
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,5
LD (X_STPLT),A
LD IY,(PL_PTR)

LD (IY+0),033H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),9 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT

LD (IY+0),052H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),8 ;CH_COL

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD A,2
LD (PL_COLR),A
CALL RUNPLOT

LD (Y+0),020H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),5 ;CH_COL

LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),033H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL

LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),035H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL

LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),031H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),6 ;CH_COL

LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),030H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL

LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),033H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL

LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),032H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
```

LD A,2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),031H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),6 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),034H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),9 ;CH_COL
LD A,2
LD (PL_COLR),A
CALL RUNPLOT
```

```
RET
```

```
LDPRE11: ;NUMBER 35103215
```

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,5
LD (X_STPLT),A
LD IY,(PL_PTR)
LD (IY+0),033H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),052H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),8 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),020H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),5 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (IY+0),033H ;CH_ASC
LD (IY+1),000H ;CH_TYPE
LD (IY+2),2 ;CH_ROW
LD (IY+3),9 ;CH_COL
```

```
LD A,1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),035H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),9 :CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),031H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),6 :CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),030H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),9 :CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),033H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),8 :CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),032H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),9 :CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),031H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),6 :CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),035H :CH_ASC
LD (TY+1),000H :CH_TYPE
LD (TY+2),2 :CH_ROW
LD (TY+3),9 :CH_COL
LD A,I
```

```
LD (PL_COLR),A
```

เอกสารนี้เป็นเอกสารที่เผยแพร่ให้รับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL RUNPLOT
```

```
RET
```

```
LDPRE12: ;Advisor by
```

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,5
LD (X_STPLT),A
LD (Y,(PL_PTR)
```

```
LD (Y+0),041H ;CH_ASC
LD (Y+1),002H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),064H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),10 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),076H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),7 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),069H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),7 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),073H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),065H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,0
LD (PL_COLR),A
```

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL RUNPLOT

LD (IY+0),072H ;CH\_ASC  
 LD (IY+1),000H ;CH\_TYPE  
 LD (IY+2),2 ;CH\_ROW  
 LD (IY+3),8 ;CH\_COL  
 LD A,0  
 LD (PL\_COLR),A  
 CALL RUNPLOT

LD (IY+0),020H ;CH\_ASC  
 LD (IY+1),000H ;CH\_TYPE  
 LD (IY+2),2 ;CH\_ROW  
 LD (IY+3),8 ;CH\_COL  
 LD A,0  
 LD (PL\_COLR),A  
 CALL RUNPLOT

LD (IY+0),062H ;CH\_ASC  
 LD (IY+1),000H ;CH\_TYPE  
 LD (IY+2),2 ;CH\_ROW  
 LD (IY+3),10 ;CH\_COL  
 LD A,2  
 LD (PL\_COLR),A  
 CALL RUNPLOT

LD (IY+0),079H ;CH\_ASC  
 LD (IY+1),000H ;CH\_TYPE  
 LD (IY+2),3 ;CH\_ROW  
 LD (IY+3),10 ;CH\_COL  
 LD A,2  
 LD (PL\_COLR),A  
 CALL RUNPLOT  
 RET

LDPRE13: ;Dr. KITIPHOL

LD HL,PREDSPR  
 LD (NAMEBUF),HL  
 LD A,1  
 LD (X\_STPLT),A  
 LD IY,(PL\_PTR)

LD (IY+0),044H ;CH\_ASC  
 LD (IY+1),002H ;CH\_TYPE  
 LD (IY+2),2 ;CH\_ROW  
 LD (IY+3),10 ;CH\_COL  
 LD A,2  
 LD (PL\_COLR),A  
 CALL RUNPLOT

LD (IY+0),072H ;CH\_ASC  
 LD (IY+1),000H ;CH\_TYPE  
 LD (IY+2),2 ;CH\_ROW  
 LD (IY+3),6 ;CH\_COL  
 LD A,2  
 LD (PL\_COLR),A

CALL RUNPLOT

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (Y+0),02EH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),5 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),020H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),5 ;CH_COL
LD A,0
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),04BH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),049H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),7 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),054H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),049H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),7 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),050H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,1
LD (PL_COLR),A
CALL RUNPLOT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (Y+0),048H ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),04FH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),9 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),04CH ;CH_ASC
LD (Y+1),000H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),8 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
RET
```

LDPRE14: ;I LOVE YOU

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,0
LD (X_STPLT),A
LD (Y,(PL_PTR)
```

```
LD (Y+0),049H ;CH_ASC
LD (Y+1),003H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),19 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),088H ;CH_ASC
LD (Y+1),010H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
CALL RUNPLOT
CALL CLRGEN
```

```
LD (Y+0),059H ;CH_ASC
LD (Y+1),003H ;CH_TYPE
LD (Y+2),2 ;CH_ROW
LD (Y+3),17 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (Y+0),04FH ;CH_ASC
LD (Y+1),003H ;CH_TYPE
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (TY+2),2 ;CH_ROW
LD (TY+3),17 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
```

```
LD (TY+0),055H ;CH_ASC
LD (TY+1),003H ;CH_TYPE
LD (TY+2),2 ;CH_ROW
LD (TY+3),19 ;CH_COL
LD A,I
LD (PL_COLR),A
CALL RUNPLOT
RET
```

```
LDPRE15: ;BOAD,ICECREAM,CLOCK,CAR,APPLE
```

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,I
LD (X_STPLT),A
LD (Y_PL_PTR)
```

```
LD (TY+0),080H ;CH_ASC
LD (TY+1),010H ;CH_TYPE
LD (TY+2),0 ;CH_ROW
LD (TY+3),19 ;CH_COL
CALL RUNPLOT
```

```
LD (TY+0),081H ;CH_ASC
LD (TY+1),010H ;CH_TYPE
LD (TY+2),0 ;CH_ROW
LD (TY+3),19 ;CH_COL
CALL RUNPLOT
```

```
LD (TY+0),082H ;CH_ASC
LD (TY+1),010H ;CH_TYPE
LD (TY+2),0 ;CH_ROW
LD (TY+3),19 ;CH_COL
CALL RUNPLOT
```

```
LD (TY+0),083H ;CH_ASC
LD (TY+1),010H ;CH_TYPE
LD (TY+2),0 ;CH_ROW
LD (TY+3),19 ;CH_COL
CALL RUNPLOT
```

```
LD (TY+0),084H ;CH_ASC
LD (TY+1),010H ;CH_TYPE
LD (TY+2),0 ;CH_ROW
LD (TY+3),19 ;CH_COL
CALL RUNPLOT
RET
```

```
LDPRE16: ;FISH,TREE,COMPUTER,HEARTH,SUN
```

```
LD HL,PREDSPR
LD (NAMEBUF),HL
LD A,I
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:FILENAME 01S.ASM
:DESCRIPTION PLOT other buffers to NAMEBUF
:START DATE 14 JANUARY 1905

;*****
; RUNPLOT: plot 1 token pointer to TXT_BUF
; INPUT : RUN_PTR
; OUTPUT : plot to TXT_BUF,increment X_STPLT by char column length
;
RUNPLOT: PUSH BC
        PUSH DE
        PUSH HL
        LD A,(DX+1)
        CP 08H
        JR Z,RUN_S1
        JR NC,RUN_S1

LD A,(DX-4)
CP 'S'
JR NZ,RUN_S1
LD A,(DX+0)
CP 'H'
JR Z,RUN_S0
CP 'M'
JR Z,RUN_S0
CP 'S'
JR Z,RUN_S0
CP 'D'
JR Z,RUN_S0
CP 'N'
JR Z,RUN_S0
CP 'Y'
JR Z,RUN_S0
JR RUN_S1

RUN_S0: INC IY
        INC IY
        INC IY
        INC IY

RUN_S1: LD A,(IY+0)
        LD (CH_ASC),A
        LD A,(IY+1)
        LD (CH_TYPE),A
        LD A,(IY+2)
        LD (CH_ROW),A
        LD A,(IY+3)
        LD (CH_COL),A

        LD A,(CH_ROW)
        ADD A,3
        LD (Y_STPLT),A

        CALL GETGEN

        LD HL,GEN_BUF1
        LD (NAMEGEN),HL

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,(PL_LOOP)
CP 4
JP Z,SARA_UR
CP 5
JP Z,UP_CHAR1
CP 6
JP Z,UP_CHAR2
CP 7
JP Z,SARASURM
CP 9
JP Z,GP_COLOR

RUN_S1_1:LD A,(CH_FONT)
CP 0
JP NZ,RUN_S2
LD A,(CH_TYPE)
CP 4
JP NC,RUN_S1_2
CALL PL8X8
JP RUN_S5
RUN_S1_2:CALL PL8X8
JP RUN_S5

RUN_S2: CP 1
JP NZ,RUN_S3
LD A,(CH_TYPE)
CP 4
JP NC,RUN_S2_1
CALL PL8X16
JP RUN_S5
RUN_S2_1:CALL PL8X16
JP RUN_S5

RUN_S3: CP 2
JP NZ,RUN_S4
LD A,(CH_TYPE)
CP 4
JP NC,RUN_S3_1
CALL PL16X8
JP RUN_S5
RUN_S3_1:CALL PL16X8
JP RUN_S5

RUN_S4: CP 3
JP NZ,RUN_S4_2
LD A,(CH_TYPE)
CP 4
JP NC,RUN_S4_1
CALL PL16X16
JP RUN_S5
RUN_S4_1:CALL PL16X16
JP RUN_S5

RUN_S4_2:CP 4
JP NZ,RUN_S5
CALL PL8X8

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JP RUN_S5

RUN_S4_3:LD A,0
LD (X0_FLAG),A
JP RUN_S5

RUN_S5: LD HL,GEN_BUF2
LD (NAMEGEN),HL
LD A,(PL_LOOP)
CP 1
JP Z,RUN_S10
CP 2
JP NZ,RUN_S6
CALL SALA_IR
JP RUN_S1_1

RUN_S6: CP 3
JP NZ,RUN_S7
CALL YOYINGR
JP RUN_S1_1

RUN_S7: CP 4
JP NZ,RUN_S8
LD A,1
LD (PL_LOOP),A
JP RUN_S10

RUN_S8: CP 5
JP NZ,RUN_S9
LD A,1
LD (PL_LOOP),A
JP RUN_S10

RUN_S9: CP 6
JP NZ,RUN_S11
LD A,1
LD (PL_LOOP),A

RUN_S11: CP 8
JR NZ,RUN_S12
JP SARA_RR

RUN_S12: CP 9
JR NZ,RUN_S13
LD A,10
LD (PL_LOOP),A
LD A,1
LD (PL_COLR),A ;Then: Plot Green Color
JP RUN_S1_1

RUN_S13: CP 10
JR NZ,RUN_S10
LD A,(CL_BUF)
LD (PL_COLR),A ;Return Color to PL_COLR
LD A,1
LD (PL_LOOP),A
JR RUN_S10

RUN_S10: LD B,0
LD A,(CH_COL)
LD C,A
LD HL,(X_STPLT)
ADD HL,BC
LD (X_STPLT),HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP HL
POP DE
POP BC
RET

```

```

;*** SUBROUTINE FOR RUNPLOT ***

```

```

SALA_IR: LD  A,(Y_STPLT) ;mix char upper
        SUB  7
        LD  (Y_STPLT),A
        LD  A,I
        LD  (PL_LOOP),A
        RET

```

```

YOYINGR: LD  A,(Y_STPLT) ;mix char under
        ADD A,8
        LD  (Y_STPLT),A
        LD  A,I
        LD  (PL_LOOP),A
        RET

```

```

SARA_UR:LD A,(Y_STPLT)
        ADD A,3
        LD  (Y_STPLT),A

        LD  B,0
        LD  A,(CH_COL)
        LD  C,A
        LD  HL,(X_STPLT)
        SCF
        CCF
        SBC HL,BC
        LD  (X_STPLT),HL
        JP  RUN_S1_I

```

```

UP_CHAR1: LD  A,(Y_STPLT) ;upper char 1 st
        SUB  8
        LD  (Y_STPLT),A

        LD  B,0
        LD  A,(CH_COL)
        LD  C,A
        LD  HL,(X_STPLT)
        SCF
        CCF
        SBC HL,BC
        LD  (X_STPLT),HL
        JP  RUN_S1_I

```

```

UP_CHAR2: LD  A,(Y_STPLT) ;upper char 2 nd
        SUB  9
        LD  (Y_STPLT),A

        LD  B,0
        LD  A,(CH_COL)
        LD  C,A
        LD  HL,(X_STPLT)
        SCF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CCF
SBC HL,BC
LD (X_STPLT),HL
JP RUN_S1_I

```

```

SARASURM: LD A,(Y_STPLT) ;upper char 2 nd
SUB 7
LD (Y_STPLT),A

LD A,8
LD (PL_LOOP),A
JP RUN_S1_I

```

```

SARA_RR: LD A,(Y_STPLT) ;upper char 2 nd
ADD A,7
LD (Y_STPLT),A
LD A,1
LD (PL_LOOP),A
JP RUN_S1_I

```

```

GP_COLOR: LD A,(PL_COLR) ;save current color to CL_BUF
LD (CL_BUF),A
LD A,0
LD (PL_COLR),A ;First: Plot RED Color
LD (Y_STPLT),A ;Start Y=0
JP RUN_S1_I

```

```

;-----
ONSCAN: PUSH HL
LD A,0
LD (ROW),A
LD HL,DSPBUFR
LD (ADDSCANR),HL
LD HL,DSPBUFG
LD (ADDSCANG),HL
LD A,11H
OUT0 (TCR),A
POP HL
RET

```

```

;-----
OFFSCAN: PUSH BC
OFFSCAN1: LD A,(ROW)
CP 15
JR NZ,OFFSCAN1
LD A,17 ;OFF scan by send over
OUT (PORTA),A
LD A,0 ;clear timer interrupt enable
OUT0 (TCR),A
POP BC
RET

```

```

CLRDSP: CALL CLRDSPR
CALL CLRDSPG
RET

```

```

CLRDSPR: PUSH HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH BC
PUSH DE
LD HL,DSPBUFR
LD DE,DSPBUFR+1
LD A,0
LD (HL),A
LD BC,NUMROW*NUMCOL
LDIR
POP DE
POP BC
POP HL
RET

```

```

CLRDSPG: PUSH HL
          PUSH BC
          PUSH DE
          LD HL,DSPBUFG
          LD DE,DSPBUFG+1
          LD A,0
          LD (HL),A
          LD BC,NUMROW*NUMCOL
          LDIR
          POP DE
          POP BC
          POP HL
          RET

:***** DMA SECTION *****

:*** MOVE DMA ***
:Input: HL=source address.DSPSIZE=byte count
:Output: PREDSP
:*****

MOVDMA:  OUT0 (SAR0L),L ;set source address
          OUT0 (SAR0H),H
          LD  A,0
          OUT0 (SAR0B),A

          LD  HL,PREDSPR ;set destination address
          OUT0 (DAR0L),L
          OUT0 (DAR0H),H
          LD  A,0
          OUT0 (DAR0B),A

          LD  A,0000010B ;dest=memory+ ,source=memory+,burst mode
          OUT0 (DMODE),A

          LD  HL,(DSPSIZE)
          OUT0 (BCR0L),C
          OUT0 (BCR0H),B

          LD  A,01000000B ;DMA0 enable
          OUT0 (DSTAT),A
          RET

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ฟรีสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:Input: HL=source address,DE=destination address,BC=byte count
:*****

DMATRAN: OUT0 (SAR0L),L ;set source address
          OUT0 (SAR0H),H
          LD  A,0
          OUT0 (SAR0B),A

          OUT0 (DAR0L),E ;set destination address
          OUT0 (DAR0H),D
          LD  A,0
          OUT0 (DAR0B),A

          :LD  A,00000010B ;dest=memory+ ,source=memory+,burst mode
          LD  A,00000000B ;cycle steal mode
          OUT0 (DMODE),A

          OUT0 (BCR0L),C
          OUT0 (BCR0H),B

          LD  A,01000000B ;DMA0 enable
          OUT0 (DSTAT),A
          RET

:*** LOAD CHARACTER ***
:Input: 0HL=source address,BC=byte count
:Output: GEN_BUF1
:*****

LDCHAR: OUT0 (SAR0L),L ;set source address
          OUT0 (SAR0H),H
          LD  A,00000000B ;0XXXX
          OUT0 (SAR0B),A

          LD  HL,GEN_BUF1 ;set destination address
          OUT0 (DAR0L),L
          OUT0 (DAR0H),H
          LD  A,0
          OUT0 (DAR0B),A

          LD  A,00000010B ;dest=memory+ ,source=memory+,burst mode
          OUT0 (DMODE),A

          OUT0 (BCR0L),C
          OUT0 (BCR0H),B

          LD  A,01000000B ;DMA0 enable
          OUT0 (DSTAT),A
          RET

:*** CLEAR DISPLAY BUFFER ***
:Input: NONE
:*****

CRDSPDMA:PUSH AF
          PUSH BC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD BC,BLANK
OUT0 (SAR0L),C ;set source address
OUT0 (SAR0H),B
XOR A
OUT0 (SAR0B),A

LD BC,DSPBUFR ;set destination address
OUT0 (DAR0L),C
OUT0 (DAR0H),B
XOR A
OUT0 (DAR0B),A

LD A,11001011B ;dest=memory+ ,source=memory fix,burst mode
OUT0 (DMODE),A

LD BC,NUMCOL*NUMROW*2
OUT0 (BCR0L),C
OUT0 (BCR0H),B

LD A,0100000B ;DMA0 enable
OUT0 (DSTAT),A

POP BC
POP AF
RET

;*****
; PL8X8: PLOT CHARACTER GEN. 8X8 to display buffer
; INPUT: PL_COLR 0=RED, 1=GREEN, 2=YELLOW
;*****

PL8X8: PUSH BC
      PUSH DE
      PUSH HL
      LD HL,(NAMEGEN)

PL8X8_3: PUSH HL
      PUSH DE
      LD A,(Y_STPLT)
      LD (Y_START),A
      LD (YPOS),A
      LD A,0
      LD (X0_FLAG),A

      PUSH HL
      LD HL,(X_STPLT)
      LD (X_START),HL
      LD (XPOS),HL
      POP HL

      LD E,8 ;21 line
PL8X8_0: LD C,3
PL8X8_1: LD A,(HL)
      LD (PL_DATA),A

LD B,4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD D,1000000B
PL8X8_2: LD A,(PL_DATA)
AND D
CALL NZ,PL8X8_P ;bit = 1 PLOT
RRC D
RRC D
PUSH HL
LD HL,(XPOS) ;X_POS=X_POS+1
INC HL
LD (XPOS),HL
POP HL
DJNZ PL8X8_2 ;until finish 8 bit

INC HL ;increment BUFFER pointer
DEC C
JP NZ,PL8X8_1
PUSH HL
LD HL,(X_START)
LD (XPOS),HL ;XPOS=X_START
POP HL
LD A,(YPOS)
INC A ;YPOS=YPOS+1
LD (YPOS),A
DEC E
JP NZ,PL8X8_0
POP DE
POP HL
POP HL
POP DE
POP BC
RET

PL8X8_P: PUSH HL
LD HL,(XPOS)
LD A,H
BIT 7,A
JR Z,PL8X8_P1
LD A,1
LD (X0_FLAG),A
POP HL
RET

PL8X8_P1: POP HL
LD A,(YPOS)
AND 1000000B ;<0 then no plot
JR NZ,PL8X8_NP
LD A,(YPOS)
CP NUMROW ;0FFH
JR Z,PL8X8_NP ;ypos >= 24 then no plot
JR NC,PL8X8_NP

LD A,(PL_COLR)
CP 0 ;check RED
JR Z,PL8X8_PR
CP 1
JR Z,PL8X8_PG

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PL8X8_PY:CALL PLOTY
      RET
PL8X8_PR:CALL PLOTR
      RET
PL8X8_PG:CALL PLOTG
      RET
PL8X8_NP:RET

;*****
; PL8X16: PLOT CHARACTER GEN. 8X16 to display buffer
; INPUT: PL_COLR 0=RED, 1=GREEN, 2=YELLOW
;
;*****

```

```

PL8X16: PUSH BC
      PUSH DE
      PUSH HL
      LD HL,(NAMEGEN)

PL8X16_3: PUSH HL
      PUSH DE
      LD A,(Y_STPLT)
      LD (Y_START),A
      LD (YPOS),A
      LD A,0
      LD (X0_FLAG),A

      PUSH HL
      LD HL,(X_STPLT)
      DEC HL
      DEC HL
      DEC HL
      LD (X_START),HL
      LD (XPOS),HL
      POP HL

      LD E,8 ;21 line
PL8X16_0: LD C,3
PL8X16_1: LD A,(HL)
      LD (PL_DATA),A
      LD B,8
      LD D,10000000B
PL8X16_2: LD A,(PL_DATA)
      AND D
      CALL NZ,PL8X16_P ;bit = 1 PLOT
      RRC D
      PUSH HL
      LD HL,(XPOS) ;X_POS=X_POS+1
      INC HL
      LD (XPOS),HL
      POP HL
      DJNZ PL8X16_2 ;until finish 8 bit
      INC HL ;increment BUFFER pointer
      DEC C
      JP NZ,PL8X16_1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH HL
LD HL,(X_START)
LD (XPOS),HL ;XPOS=X_START
POP HL
LD A,(YPOS)
INC A ;YPOS=YPOS+1
LD (YPOS),A
DEC E
JP NZ,PL8X16_0
POP DE
POP HL

```

```

POP HL
POP DE
POP BC
RET

```

```

PL8X16_P: PUSH HL

```

```

LD HL,(XPOS)
LD A,H
BIT 7,A
JR Z,PL8X16P1
LD A,1
LD (X0_FLAG),A
POP HL
RET

```

```

PL8X16P1: POP HL

```

```

LD A,(YPOS)
AND 1000000B ;<0 then no plot
JR NZ,PL8X16NP
LD A,(YPOS)
CP NUMROW ;:OFFH
JR Z,PL8X16NP ;ypos >= 24 then no plot
JR NC,PL8X16NP

LD A,(PL_COLR)
CP 0 ;check RED
JR Z,PL8X16PR
CP 1
JR Z,PL8X16PG

```

```

PL8X16PY: CALL PLOTY

```

```

RET

```

```

PL8X16PR: CALL PLOTR

```

```

RET

```

```

PL8X16PG: CALL PLOTG

```

```

RET

```

```

PL8X16NP: RET

```

```

;.....
; PL16X8: PLOT CHARACTER GEN. 16X8 to display buffer

```

```

; INPUT: PL_COLR 0=RED, 1=GREEN, 2=YELLOW

```

```

;
;.....

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH DE
PUSH HL
LD HL,(NAMEGEN)
LD A,(Y_STPLT)
LD (Y_START),A
LD (YPOS),A
LD A,0
LD (X0_FLAG),A

PUSH HL
LD HL,(X_STPLT)
LD (X_START),HL
LD (XPOS),HL
POP HL

LD E,16 ;:21 line
PL16X8_0: LD C,3
PL16X8_1: LD A,(HL)
LD (PL_DATA),A
LD B,4
LD D,1000000B
PL16X8_2: LD A,(PL_DATA)
AND D
CALL NZ,PL16X8_P ;bit = 1 PLOT
RRC D
RRC D
PUSH HL
LD HL,(XPOS) ;X_POS=X_POS+1
INC HL
LD (XPOS),HL
POP HL
DJNZ PL16X8_2 ;until finish 8 bit
INC HL ;increment BUFFER pointer
DEC C
JP NZ,PL16X8_1

PUSH HL
LD HL,(X_START)
LD (XPOS),HL ;XPOS=X_START
POP HL
LD A,(YPOS)
INC A ;YPOS=YPOS+1
LD (YPOS),A
DEC E
JP NZ,PL16X8_0

POP HL
POP DE
POP BC
RET

```

```

PL16X8_P: PUSH HL
LD HL,(XPOS)
LD A,H
BIT 7,A
JR Z,PL16X8P1
LD A,L

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (X0_FLAG),A
POP HL
RET
PL16X8P1: POP HL
LD A,(YPOS)
AND 1000000B ;<0 then no plot
JR NZ,PL16X8NP
LD A,(YPOS)
CP NUMROW ;0FFH
JR Z,PL16X8NP ;ypos >= 24 then no plot
JR NC,PL16X8NP

LD A,(PL_COLR)
CP 0 ;check RED
JR Z,PL16X8PR
CP 1
JR Z,PL16X8PG

```

```
PL16X8PY: CALL PLOTY
```

```
RET
```

```
PL16X8PR: CALL PLOTR
```

```
RET
```

```
PL16X8PG: CALL PLOTG
```

```
RET
```

```
PL16X8NP: RET
```

```

;*****
; PL16X16: PLOT CHARACTER GEN. 16X16 (H x W) to display buffer
; INPUT: PL_COLR 0=RED, 1=GREEN, 2=YELLOW
; X_STPLT,Y_STPLT
;*****

```

```
PL16X16: PUSH BC
```

```
PUSH DE
```

```
PUSH HL
```

```
LD HL,(NAMEGEN)
```

```
LD A,(Y_STPLT)
```

```
LD (Y_START),A
```

```
LD (YPOS),A
```

```
LD A,0
```

```
LD (X0_FLAG),A
```

```
PUSH HL
```

```
LD HL,(X_STPLT)
```

```
LD (X_START),HL
```

```
LD (XPOS),HL
```

```
POP HL
```

```
LD E,16 ;16 line
```

```
PL16X160: LD C,3
```

```
PL16X161: LD A,(HL)
```

```
LD (PL_DATA),A
```

```
LD B,8
```

```
LD D,1000000B
```

```
PL16X162: LD A,(PL_DATA)
```

```
AND D
```

```
CALL NZ,PL16X16P ;bit = 1 PLOT
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RRC D
PUSH HL
LD HL,(XPOS) ;X_POS=X_POS+1
INC HL
LD (XPOS),HL
POP HL
DJNZ PL16X162 ;until finish 8 bit
INC HL ;increment BUFFER pointer
DEC C
JP NZ,PL16X161

PUSH HL
LD HL,(X_START)
LD (XPOS),HL ;XPOS=X_START
POP HL
LD A,(YPOS)
INC A ;YPOS=YPOS+1
LD (YPOS),A
DEC E
JP NZ,PL16X160

POP HL
POP DE
POP BC
RET

PL16X16P: PUSH HL
LD HL,(XPOS)
LD A,H
BIT 7,A
JR Z,PL1616P1
LD A,L
LD (X0_FLAG),A
POP HL
RET

PL1616P1: POP HL
LD A,(YPOS)
AND 1000000B ;<0 then no plot
JR NZ,PL1616NP
LD A,(YPOS)
CP NUMROW ;:0FFH
JR Z,PL1616NP ;ypos >= 24 then no plot
JR NC,PL1616NP

LD A,(PL_COLR)
CP 0 ;check RED
JR Z,PL1616PR
CP 1
JR Z,PL1616PG

PL1616PY: CALL PLOTY
RET
PL1616PR: CALL PLOTR
RET
PL1616PG: CALL PLOTG
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PL1616NP: RET

```

:-----
: PLGRAPH: PLOT CHARACTER GEN. 16X16 (H x W)to display buffer
: INPUT: PL_COLR 0=RED, 1=GREEN, 2=YELLOW
: X_STPLT,Y_STPLT
:-----

```

```

PLGRAPH: PUSH BC
        PUSH DE
        PUSH HL

```

```

PLGRAPH0: LD  A,0
          LD  (Y_START),A
          LD  (YPOS),A
          LD  A,0
          LD  (X0_FLAG),A

          LD  HL,(X_STPLT)
          LD  (X_START),HL
          LD  (XPOS),HL

          LD  HL,(NAMEGEN)
          LD  E,16      ;16 line
PLGRAPH1: LD  C,2
PLGRAPH2: LD  A,(HL)
          LD  (PL_DATA),A
          LD  B,8
          LD  D,10000000B
PLGRAPH3: LD  A,(PL_DATA)
          AND  D
          CALL NZ,PLGRAPHP ;bit = 1 PLOT
          RRC  D

          PUSH HL
          LD  HL,(XPOS) ;X_POS=X_POS+1
          INC HL
          LD  (XPOS),HL
          POP HL

          DJNZ PLGRAPH3 ;until finish 8 bit
          INC HL ;increment BUFFER pointer
          DEC C
          JP  NZ,PLGRAPH2

          PUSH HL
          LD  HL,(X_START)
          LD  (XPOS),HL ;XPOS=X_START
          POP HL

          LD  A,(YPOS)
          INC A ;YPOS=YPOS+1
          LD  (YPOS),A
          DEC E
          JP  NZ,PLGRAPH1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP HL
POP DE
POP BC
RET

```

```

PLGRAPHP: PUSH HL
LD HL,(XPOS)
LD A,H
BIT 7,A
JR Z,PLGRAPPI
LD A,I
LD (X0_FLAG),A
POP HL
RET

```

```

PLGRAPPI: POP HL
LD A,(YPOS)
AND 1000000B ;<0 then no plot
JR NZ,PLGRAPNP
LD A,(YPOS)
CP NUMROW ;0FFH
JR Z,PLGRAPNP ;ypos >= 24 then no plot
JR NC,PLGRAPNP
LD A,(PL_COLR)
CP 0 ;check RED
JR Z,PLGRAPPR
CP 1
JR Z,PLGRAPPG
PLGRAPPY: CALL PLOTY
RET
PLGRAPPR: CALL PLOTR
RET
PLGRAPPG: CALL PLOTG
RET
PLGRAPNP: RET

```

```

;*****
; SUB_NAME: PLOTR
; DESCRIPTION PLOT 1 POINT TO DSPBUF
; INPUT NAMEBUF = buffer name etc. DSPBUFR
; LENBUF = buffer lenght
; XPOS = X POSITION (0 - 95)
; YPOS = Y POSITION (0 - 15)
;*****

```

```

PLOTR: PUSH BC
PUSH DE
PUSH HL
LD HL,(NAMEBUF)
LD A,(LENBUF) ;ROW ADD.= start add.of buffer + (LENBUF*YPOS)
LD B,A
LD A,(YPOS)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;find row address of BUFFER to be plot,put it in HL

```
LD C,A
MLT BC ;BC = LENBUF*YPOS
ADD HL,BC ;HL = ROW ADDRESS
```

;find OFFSET address of BUFFER to be plot,put it in BC

;find bit to be plot,put it in D

PLOTR\_1: PUSH HL

```
LD HL,(XPOS)
LD BC,NUMCOL*8
SCF
CCF
SBC HL,BC
JP C,PLOTR_11 ;XPOS < NUMCOL*8 then jump
POP HL ;XPOS >= NUMCOL*8 then end PLOTR
JP PLOTR_E
```

PLOTR\_11:POP HL

```
LD BC,(XPOS)

LD A,C
AND 00001111B
LD EA ; E = lo nibble of C
AND 00000111B
LD DA ; D = bit contents

LD A,C ;find OFFSET ADD.
RRCA
RRCA
RRCA
RRCA
AND 00001111B
LD C,A

LD A,B
RLCA
RLCA
RLCA
RLCA
AND 11110000B
OR C
ADD A,A
LD C,A
```

BIT 3,E

JR Z,PLOTR\_2

INC C

PLOTR\_2: ;C = OFFSET ADDRESS

```
LD B,0 ;BC = OFFSET address of buffer
ADD HL,BC ;HL = address of buffer to be plot
```

LD B,D ;B = bit contents

LD A,00000001B

PLOTR\_3: RLCA

DJNZ PLOTR\_3

OR (HL) ;load data of buffer to be plot

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (HL),A      ;plot in this line
PLOTTR_E: POP HL
              POP DE
              POP BC
              RET

BLNKR:  PUSH BC
        PUSH DE
        PUSH HL
        LD HL,(NAMEBUF)
        LD A,(LENBUF) ;ROW ADD.= start add.of buffer + (LENBUF*YPOS)
        LD B,A
        LD A,(YPOS)

;find row address of BUFFER to be plot,put it in HL
        LD CA
        MLT BC      ;BC = LENBUF*YPOS
        ADD HL,BC   ;HL = ROW ADDRESS

;find OFFSET address of BUFFER to be plot,put it in BC
;find bit to be plot,put it in D
BLNKR_1:
        PUSH HL
        LD HL,(XPOS)
        LD BC,NUMCOL*8
        SCF
        CCF
        SBC HL,BC
        JP C,BLNKR_11 ;XPOS < NUMCOL*8 then jump
        POP HL      ;XPOS >= NUMCOL*8 then no plot
        JP BLNKR_E

BLNKR_11:POP HL
        LD BC,(XPOS)

        LD A,C
        AND 00001111B
        LD E,A      ; E = lo nibble of C
        AND 00001111B
        LD D,A      ; D = bit contents

        LD A,C      ;find OFFSET ADD.
        RRCA
        RRCA
        RRCA
        RRCA
        AND 00001111B
        LD CA

        LD A,B
        RLCA
        RLCA
        RLCA
        RLCA
        AND 11110000B
        OR C
        ADD A,A
        LD CA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    BIT 3,E
    JR  Z,BLNKR_2
    INC C
BLNKR_2:      ;C = OFFSET ADDRESS
    LD  B,0      ;BC = OFFSET address of buffer
    ADD HL,BC    ;HL = address of buffer to be plot

    LD  B,D      ;B = bit contents
    LD  A,1111110B
BLNKR_3: RLCA
    DJNZ BLNKR_3

    AND (HL)     ;load data of buffer to be plot
    LD  (HL),A   ;plot in this line
BLNKR_E: POP HL
    POP DE
    POP BC
    RET

;.....
; SUB_NAME: PLOTG
; DESCRIPTION PLOT 1 POINT TO DSPBUF
; INPUT NAMEBUF = buffer name etc. DSPBUF
; LENBUF = buffer leaght
; XPOS = X POSITION (0 - 95)
; YPOS = Y POSITION (0 - 15)
;.....
PLOTG: PUSH BC
    PUSH DE
    PUSH HL
    LD  HL,(NAMEBUF)
    LD  A,(LENBUF) ;ROW ADD.= start add.of buffer + (LENBUF*YPOS)
    LD  B,A
    LD  A,(YPOS)
;find row address of BUFFER to be plot,put it in HL
    LD  C,A
    MLT BC      ;BC = LENBUF*YPOS
    ADD HL,BC   ;HL = ROW ADDRESS

;find OFFSET address of BUFFER to be plot,put it in BC
;find bit to be plot,put it in D
PLOTG_1: PUSH HL
    LD  HL,(XPOS)
    LD  BC,NUMCOL*8
    SCF
    CCF
    SBC HL,BC
    JP  C,PLOTG_11 ;XPOS < NUMCOL*8 then jump
    POP HL        ;XPOS >= NUMCOL*8 then end PLOTG
    JP  PLOTG_E

PLOTG_11:POP HL
    LD  BC,(XPOS)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,C
AND 00001111B
LD E,A      ; E = lo nibble of C
AND 00001111B
LD D,A      ; D = bit contents

LD A,C      ;find OFFSET ADD.
RRCA
RRCA
RRCA
RRCA
AND 00001111B
LD C,A

LD A,B
RLCA
RLCA
RLCA
RLCA
AND 11110000B
OR C
ADD A,A
LD C,A

BIT 3,E
JR Z,PLOTG_2
INC C
PLOTG_2: ;C = OFFSET ADDRESS
LD B,0     ;BC = OFFSET address of buffer
ADD HL,BC  ;HL = address of buffer to be plot

LD B,D     ;B = bit contents
LD A,0000001B
PLOTG_3: RLCA
DJNZ PLOTG_3

LD BC,NUMROW*NUMCOL
ADD HL,BC
OR (HL)    ;load data of buffer to be plot
LD (HL),A  ;plot in this line
PLOTG_E: POP HL
POP DE
POP BC
RET

BLNKG: PUSH BC
PUSH DE
PUSH HL
LD HL,(NAMEBUF)
LD A,(LENBUF) ;ROW ADD.= start add.of buffer + (LENBUF*YPOS)
LD B,A
LD A,(YPOS)
;find row address of BUFFER to be plot.put it in HL
LD C,A
MLT BC     ;BC = LENBUF*YPOS
ADD HL,BC  ;HL = ROW ADDRESS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:find OFFSET address of BUFFER to be plot,put it in BC
:find bit to be plot,put it in D
BLNKG_1:
    PUSH HL
    LD HL,(XPOS)
    LD BC,NUMCOL*8
    SCF
    CCF
    SBC HL,BC
    JP C,BLNKG_11 ;XPOS < NUMCOL*8 then jump
    POP HL ;XPOS >= NUMCOL*8 then no plot
    JP BLNKG_E

```

```

BLNKG_11:POP HL
    LD BC,(XPOS)

    LD A,C
    AND 00001111B
    LD E,A ; E = lo nibble of C
    AND 00001111B
    LD D,A ; D = bit contents

    LD A,C ;find OFFSET ADD.
    RRCA
    RRCA
    RRCA
    RRCA
    AND 00001111B
    LD C,A
    LD A,B
    RLCA
    RLCA
    RLCA
    RLCA
    AND 11110000B
    OR C
    ADD A,A
    LD C,A

    BIT 3,E
    JR Z,BLNKG_2
    INC C

BLNKG_2: ;C = OFFSET ADDRESS
    LD B,0 ;BC = OFFSET address of buffer
    ADD HL,BC ;HL = address of buffer to be plot

    LD B,D ;B = bit contents
    LD A,11111110B

BLNKG_3: RLCA
    DJNZ BLNKG_3

    LD BC,NUMROW*NUMCOL
    ADD HL,BC
    AND (HL) ;load data of buffer to be plot
    LD (HL),A ;plot in this line

BLNKG_E:POP HL

```

```

POP DE
POP BC
RET

```

```

;-----
; SUB_NAME:  PLOTY
; DESCRIPTION  PLOT 1 POINT TO DSPBUFR,G
; INPUT      NAMEBUF = buffer name etc. DSPBUFR,G
;           LENBUF = buffer length
;           XPOS = X POSITION (0 - 95)
;           YPOS = Y POSITION (0 - 15)
;-----

```

```

PLOTY:  PUSH BC
        PUSH DE
        PUSH HL

```

```

        LD HL,(NAMEBUF)
        LD A,(LENBUF)  ;ROW ADD.= start add.of buffer + (LENBUF*YPOS)
        LD B,A
        LD A,(YPOS)
;find row address of BUFFER to be plot,put it in HL
        LD C,A
        MLT BC        ;BC = LENBUF*YPOS
        ADD HL,BC     ;HL = ROW ADDRESS

```

```

;find OFFSET address of BUFFER to be plot,put it in BC
;find bit to be plot,put it in D

```

```

PLOTY_1: PUSH HL
        LD HL,(XPOS)
        LD BC,NUMCOL*8
        SCF
        CCF
        SBC HL,BC
        JP C,PLOTY_11 ;XPOS < NUMCOL*8 then jump
        POP HL        ;XPOS >= NUMCOL*8 then end PLOTY
        JP PLOTY_E

```

```

PLOTY_11:POP HL

```

```

        LD BC,(XPOS)

        LD A,C
        AND 00001111B
        LD E,A        ; E = lo nibble of C
        AND 00000111B
        LD D,A        ; D = bit contents

```

```

        LD A,C        ;find OFFSET ADD.
        RRCA
        RRCA
        RRCA
        RRCA
        AND 00001111B
        LD C,A

```

```

        LD A,B
        RLCA

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RLCA
RLCA
RLCA
AND 11110000B
OR C
ADD A,A
LD CA

BIT 3,E
JR Z,PLOTY_2
INC C

PLOTY_2:      ;C = OFFSET ADDRESS
LD B,0        ;BC = OFFSET address of buffer
ADD HL,BC     ;HL = address of buffer to be plot

LD B,D        ;B = bit contents
LD A,0000001B

PLOTY_3: RLCA
DJNZ PLOTY_3

OR (HL)       ;load data of buffer to be plot
LD (HL),A     ;plot in this line
LD BC,NUMROW*NUMCOL
ADD HL,BC
LD (HL),A

PLOTY_E: POP HL
POP DE
POP BC
RET

```

```

:FILENAME:      01U.ASM
:DESCRIPTION:   Program communication ASCII and Program Timer PRT
:START_DATE:   11 JANUARY 1985

```

```

;*** SET ASCII-0 AT XTAL 18.432 ***

```

```

; 8 data,no parity,1 stop bit

```

```

;
ASC0SET: LD   A,64H
          OUT0 (CNTLA0),A
;         LD   A,00100001b ;9600
;         LD   A,00100010B ;4800
;         LD   A,00100011B ;2400
          LD   A,00100100B ;1200
          OUT0 (CNTLB0),A
          LD   A,8      ;interrupt
          OUT0 (STAT0),A
          RET

```

```

;*** Set timer counter channel 0 ***

```

```

SETPRT0: LD   A,TIMEL0
          OUT0 (TMDROL),A
          OUT0 (RLDROL),A
          LD   A,TIMEH0
          OUT0 (TMDROH),A
          OUT0 (RLDROH),A
          RET

```

```

START_T0: IN0  A,(TCR)
          OR   00010001B
          OUT0 (TCR),A
          RET

```

```

STOP_T0: IN0  A,(TCR)
          AND  11101110B
          OUT0 (TCR),A
          RET

```

```

;*** Set timer counter channel 1 ***

```

```

SETPRT1: LD   A,TIMEL1
          OUT0 (TMDR1L),A
          OUT0 (RLDR1L),A
          LD   A,TIMEH1
          OUT0 (TMDR1H),A
          OUT0 (RLDR1H),A
          RET

```

```

START_T1: IN0  A,(TCR)
          OR   00100010B
          OUT0 (TCR),A
          RET

```

```

STOP_T1: IN0  A,(TCR)
          AND  11011101B
          OUT0 (TCR),A
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;***** SUBROUTINE *****
;Timer0 interrupt service routine
;DESCRIPTION: send data to latch column port on current row
;
;      increment ROW and ADDSCAN
;
;      if ROW=NUMROW then ROW=0,ADDSCAN=DSPBUF

PRT0INT:  DI
          PUSH AF
          PUSH BC
          PUSH DE
          PUSH HL
          PUSH IX
          PUSH IY

PRT0INT1: LD A,17
          OUT (PORTA),A      ;off LED by send over

          LD A,(PAGE1P)
          DEC A
          LD (PAGE1P),A
          LD A,(CHKLOCK)
          CP 0
          JP NZ,PRTMEM

          LD A,(ROW)
          CP 8
          JR Z,PRT0INT4
          JR NC,PRT0INT4      ;>=8 scan lower board
                              ;< 8 scan upper board

          LD B,12
          LD C,RED1          ;latch RED upper board
          LD HL,(ADDSCANR)

PRT0INT2: LD A,(HL)
          OUT (C),A
          INC HL
          INC C
          DJNZ PRT0INT2
          LD (ADDSCANR),HL

PRT0INT3: LD B,12          ;latch GREEN upper board
          LD C,GRN1
          LD HL,(ADDSCANG)

PRT0I3_1: LD A,(HL)
          OUT (C),A
          INC HL
          INC C
          DJNZ PRT0I3_1
          LD (ADDSCANG),HL
          JR PRT0INT6

PRT0INT4: LD B,12          ;latch RED lower board
          LD C,RED2
          LD HL,(ADDSCANR)

PRT0I4_1: LD A,(HL)
          OUT (C),A
          INC HL
          INC C
          DJNZ PRT0I4_1
          LD (ADDSCANR),HL
          JR PRT0INT6

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DJNZ PRT0I4_1
LD (ADDSCANR),HL

PRT0INT5: LD B,12 ;latch GREEN lower board
           LD C,GRN2
           LD HL,(ADDSCANG)

PRT0I5_1: LD A,(HL)
           OUT (C),A
           INC HL
           INC C
           DJNZ PRT0I5_1
           LD (ADDSCANG),HL

PRTMEM:
PRT0INT6: LD A,(ROW)
           OUT (PORTA),A ;data show on current row

PUSH AF
LD BC,150
TIME1: DEC BC
        LD A,B
        OR C
        JR NZ,TIME1
        POP AF

INC A
LD (ROW),A ;ROW=ROW+1
CP NUMROW
JR NZ,PRT0INT7
XOR A
LD (ROW),A
LD HL,DSPBUFR
LD (ADDSCANR),HL
LD HL,DSPBUIFG
LD (ADDSCANG),HL

PRT0INT7: IN0 A,(TCR) ;TIFO flag enable ?
           BIT 6,A
           JR Z,PRT0INT7
           IN0 A,(TMDROL) ;Yes, clear TIFO flag
           POP IY
           POP IX
           POP HL
           POP DE
           POP BC
           POP AF
           EI
           RETI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:FILENAME  01R.ASM
:DESCRIPTION  SUB-ROUTINE FOR RUN PROGRAM
:START DATE  6 FEBRUARY 1995

```

```

:-----
: NAME:    VELOCITY
: DESCRIPTION: control speed of display moving
: DIRECTION: 1,2,3 (use VELOCON)
:-----

```

```

VELOCITY: LD  A,(RUN_CMD)  ;VELOCITY
          SUB  20H
          INC  A
          CP  01H
          JR  Z,VELO_1
          CP  02H
          JR  Z,VELO_2
          CP  03H
          JR  Z,VELO_3

```

```

VELO_1: LD  A,80H
        LD  (DISDLY),A
        RET

```

```

VELO_2: LD  A,40H
        LD  (DISDLY),A
        RET

```

```

VELO_3: LD  A,00H
        LD  (DISDLY),A
        RET

```

```

:-----
: NAME:    MOVDLY
:-----

```

```

: DESCRIPTION: this routine is control speed of display moving
:-----

```

```

MOVDLY: LD  A,30H  ;030H
        LD  (PAGE LP),A
LP4:    LD  A,(PAGE LP)
        CP  0
        JR  NZ,LP4
        RET

```

```

MOVDLY2: LD  A,1  ;DELAY-4 ;001H
         LD  (PAGE LP),A

```

```

DLY2_1: LD  A,(PAGE LP)
        CP  0
        JR  NZ,DLY2_1
        RET

```

```

MOVDLY3: LD  A,0FFH  ;0FFH
         LD  (PAGE LP),A

```

```

DLY3_1: LD  A,(PAGE LP)
        CP  0
        JR  NZ,DLY3_1
        RET

```

```

MOVDLY4: LD  A,(DISDLY)  ;DISDLY

```

```

        CP  0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JR Z,DLY4_2
        LD (PAGEP),A
DLY4_1: LD A,(PAGEP)
        CP 0
        JR NZ,DLY4_1
DLY4_2: RET

MOV DLY5: PUSH BC
        LD A,(DISDLY)      ;DISDLY + 10H
        ADD A,010H
        LD (PAGEP),A
DLY5_1: LD A,(PAGEP)
        CP 0
        JR NZ,DLY5_1
DLY5_2: POP BC
        RET

```

```

;-----
; NAME:      PAUSECON
; DESCRIPTION: break moving of display 0.1 - 0.9 second
;            use PAUSECON is variable contents
; DIRECTION: NOP
;-----

```

```

PAUSECON: PUSH AF
          PUSH BC
          PUSH HL
          LD A,(RUN_CMD) ;PAUSECON
          SUB 23H
          INC A
          JR PAUSE_1

```

```

PAUSE:   PUSH AF
          PUSH BC
          PUSH HL
          CALL DSPDELAY
          LD A,(PAUSECON)
PAUSE_1: CP 09H
          JR Z,PAUSE_2 ;if PAUSECON = 9 jump
          JR C,PAUSE_2 ;if PAUSECON < 9 jump
          JR PAUSE_5   ;if PAUSECON < 9 return

```

```

PAUSE_2: LD B,A
PAUSE_21: LD C,2
PAUSE_3: LD HL,08FFFH ;DELAY 0.5 SEC
PAUSE_4: DEC HL
          NOP
          NOP
          LD A,H
          OR L
          JP NZ,PAUSE_4
          DEC C
          JR NZ,PAUSE_3
          DJNZ PAUSE_21

```

```

PAUSE_5: POP HL
          POP BC
          POP AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

;-----
SHFLONG: LD IX,(RUN_PTR) ;TEXT PROCESS
          PUSH IX
          CALL INC_RPTR
          XOR A
          LD (SLONG),A
          LD B,255
SHFLONG1: LD IX,(RUN_PTR)
          LD A,(IX+1) ;CHAR TYPE
          CP 13H
          JR Z,SHFLONG2
          CALL INC_RPTR
          DJNZ SHFLONG1
          JR SHFLONG3

SHFLONG2: LD A,(IX+0)
          CP 48H
          JR NZ,SHFLONG3
          LD A,1
          LD (SLONG),A

SHFLONG3: POP IX
          LD (RUN_PTR),IX
          LD HL,TEXTBUF
          LD A,(SHFBYTE)
          LD B,A
SHFLNG_1: CALL ROTBYTE
          INC HL
          DJNZ SHFLNG_1
          LD A,(SLONG)
          CP 1
          JR NZ,SHFLONG4
          XOR A
          LD (SLONG),A
          RET

SHFLONG4: LD A,(MAXCOL)
          LD B,A
SHFLNG_2: PUSH BC
          LD E,08H ;ROT 8 BIT
SHFLNG_3: LD IY,DSPBUFR ;address of display buffer
          LD D,NUMROW ;NUMBER OF ROW
          PUSH DE
          LD E,00H
SHFLNG_4: PUSH IY
          SCF
          CCF
          CALL ROTBIT
          POP IY
          LD A,(MAXCOL) ;point to next row of DSPBUF
          LD B,0
          LD C,A
          ADD IY,BC
          CALL MOVDLY2
          DEC D
          JR NZ,SHFLNG_4
          CALL MOVDLY4 ;page loop

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POP DE
DEC E
JR NZ,SHFLNG_3
POP BC
DJNZ SHFLNG_2
XOR A
LD (SLONG),A
RET

```

```

-----
; NAME: ROTBYTE
; DESCRIPTION this routine is used by SHFLONG routine
; don't worry about how did it work...
; DIRECTION: NOP
-----

```

```

ROTBYTE: PUSH BC
LD E,08H ;ROT 8 BIT
LP1: PUSH HL
LD IY,DSPBUFR ;address of display buffer
LD D,NUMROW ;NUMBER OF ROW
PUSH DE
LD E,00H
LP2: PUSH IY
RLC (HL)
CALL ROTBIT
POP IY
LD A,(MAXCOL) ;point to next row of DSPBUF
LD B,0
LD C,A
ADD IY,BC
LD BC,(MOVEBLOCK) ;point to next row of TXTBUF
ADD HL,BC ;(10 page)
CALL MOVDLY2
DEC D
JR NZ,LP2
CALL MOVDLY4
POP DE
POP HL
DEC E
JR NZ,LP1
POP BC
RET

```

```

-----
; NAME: ROTBIT
; DESCRIPTION this routine is used by ROTBYTE routine
; don't worry about him...
; DIRECTION: NOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ROTBIT: PUSH AF
        LD A,(MAXCOL)
        LD B,0
        LD C,A
        ADD IY,BC
        DEC IY
        POP AF
        LD A,(MAXCOL)
        LD B,A ;ROT COL PORT
LP3:   LD A,(IY+0)
        RL A
        LD (IY+0),A
        DEC IY
        DJNZ LP3
        RET

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

:FILENAME      01M.ASM
:DESCRIPTION   SUB ROUTINE FOR MOVING DISPLAY
:START DATE    6 FEBRUARY 1995

```

```

DSPDELAY: PUSH AF
          PUSH BC
          LD   A,(DISDLY)
          ADD  A,80
          LD   (PAGELP),A

```

```

DSPDLY_1: LD  A,(PAGELP)
          CP  0
          JR  NZ,DSPDLY_1
          POP BC
          POP AF
          RET

```

```

CHKCOLR: LD  A,(PL_COLR)
          CP  0
          JR  Z,CHKCOL_R
          CP  1
          JR  Z,CHKCOL_G

```

```

CHKCOL_R:LD  BC,DSPBUFR
          LD  (DSPCOLR),BC
          RET

```

```

CHKCOL_G:LD  BC,DSPBUFG
          LD  (DSPCOLR),BC
          RET

```

```

;-----
; NAME:      SHIFT_S
; DESCRIPTION: shift PREDSP to DSPBUF
; DIRECTION: north to south
;-----

```

```

SHIFT_S: LD  IX,PREDSPR
          LD  IY,DSPBUFR
          LD  D,NUMROW-2 ;find start add. of last row-1
          LD  A,(MAXCOL)
          LD  B,0
          LD  C,A

```

```

SHF_S1:  ADD IY,BC
          ADD IX,BC
          DEC D
          JR  NZ,SHF_S1
          ADD IX,BC
          LD  (PAPER+0),IX ;start add. of predsp (lastrow)
          LD  (PAPER+2),IY ;start add. of dspbuf (lastrow-1)
          LD  (PAPER+12H),IY
          ADD IY,BC
          LD  (PAPER+4),IY ;start add. of dspbuf (lastrow)
          LD  (PAPER+14H),IY
          LD  E,NUMROW

```

```

SHF_S2: LD  D,NUMROW-1

```

```

SHF_S3: LD  IX,(PAPER+2)
          LD  IY,(PAPER+4)

```

```

LD A,(MAXCOL)
LD B,A
SHF_S4: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_S4

PUSH IX
PUSH IY
LD IX,(PAPER+2)
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL
ADD IY,BC
ADD IX,BC
LD A,(MAXCOL)
LD B,A
SHF_S41: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_S41
POP IY
POP IX

LD A,(MAXCOL) ;load BC with MAXCOL+MAXCOL
ADD A,A
LD B,0
LD C,A
;clear A and CARRY
PUSH IX ;load HL with IX
POP HL
SBC HL,BC
PUSH HL
POP IX ;load IX with HL
LD (PAPER+2),IX
XOR A
PUSH IY
POP HL
SBC HL,BC
PUSH HL
POP IY
LD (PAPER+4),IY
DEC D
JR NZ,SHF_S3
LD IX,(PAPER+0) ;PREDSP
LD IY,(PAPER+4) ;DSPBUF
LD A,(MAXCOL)
LD B,A
SHF_S5: LD A,(IX+0) ;insert PREDSP to DSPBUF
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_S5

PUSH IX
PUSH IY

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IX,(PAPER+0)
LD IY,(PAPER+4)
*LD BC,NUMROW*NUMCOL
ADD IY,BC
ADD IX,BC
LD A,(MAXCOL)
LD B,A
SHF_S51: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_S51
POP IY
POP IX

LD A,(MAXCOL) ;load BC with MAXCOL+MAXCOL
ADD A,A
LD B,0
LD C,A
XOR A ;clear A and CARRY
PUSH IX ;load HL with IX
POP HL
SBC HL,BC
PUSH HL
POP IX ;load IX with HL
LD (PAPER+0),IX ;point to PREDSP next row
LD IX,(PAPER+12H)
LD (PAPER+2H),IX
LD IY,(PAPER+14H)
LD (PAPER+4H),IY

CALL DSPDELAY
DEC E
JP NZ,SHF_S2

LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

```

```

-----
; NAME: SHIFT_N
; DESCRIPTION: shift PREDSP to DSPBUF
; DIRECTION: south to north
-----

```

```

SHIFT_N: LD IX,PREDSPR
LD IY,DSPBUFR
LD A,(MAXCOL)
LD B,0
LD C,A
LD (PAPER+0),IX ;start add. of predsp (row 0)
LD (PAPER+4),IY ;start add. of dspbuf (firstrow)
LD (PAPER+14H),IY
ADD IY,BC
LD (PAPER+2),IY ;start add. of dspbuf (firstrow+1)
LD (PAPER+12H),IX

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD E,NUMROW
SHF_N2: LD D,NUMROW-1
SHF_N3: LD IX,(PAPER+2)
LD IY,(PAPER+4)
LD A,(MAXCOL)
LD B,A
SHF_N4: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_N4

PUSH IX
PUSH IY
LD IX,(PAPER+2)
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
LD A,(MAXCOL)
LD B,A
SHF_N41: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_N41
POP IY
POP IX

LD (PAPER+2),IX
LD (PAPER+4),IY
DEC D
JR NZ,SHF_N3

LD IX,(PAPER+0) ;PREDSP
LD IY,(PAPER+4) ;DSPBUF
LD A,(MAXCOL)
LD B,A
SHF_N5: LD A,(IX+0) ;insert PREDSP to DSPBUF
LD (IY+0),A
INC IX
INC IY
DJNZ SHF_N5

PUSH IX
PUSH IY
LD IX,(PAPER+0)
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
LD A,(MAXCOL)
LD B,A
SHF_N51: LD A,(IX+0)
LD (IY+0),A
INC IX

```

เอกสารนี้เป็นเอกสารต้นฉบับไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

D/JZ SHF_N51
POP IY
POP IX

LD (PAPER+0),IX ;point to PREDSP next row
LD IX,(PAPER+12H)
LD (PAPER+2H),IX
LD IY,(PAPER+14H)
LD (PAPER+4H),IY
CALL DSPDELAY
DEC E
JP NZ,SHF_N2

LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

```

```

;-----
; NAME: SHIFTE
; DESCRIPTION: shift PREDSP to DSPBUF
; DIRECTION: left to right
;-----

```

```

SHIFTE: LD A,(MAXCOL)
LD B,0
LD C,A
LD HL,PREDSPR
ADD HL,BC
DEC HL
LD B,A
SHF_E1: CALL SHFROT_E
DEC HL
D/JZ SHF_E1

LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

```

```

SHFROT_E: PUSH BC
LD E,08H ;ROT 8 BIT
SHFRT_E1: PUSH HL
LD IY,DSPBUFR ;address of display buffer
LD D,NUMROW ;NUMBER OF ROW
PUSH DE
LD E,00H
SHFRT_E2: PUSH IY ;process RED
RLC (HL)
CALL ROTBIT_E
POP IY

PUSH IY
PUSH HL
LD BC,NUMROW*NUMCOL
ADD HL,BC

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RLC (HL)      ;process GREEN
CALL ROTBIT_E
POP HL
POP IY

LD A,(MAXCOL) ;point to next row of DSPBUF
LD B,0
LD C,A
ADD IY,BC
ADD HL,BC
CALL MOVDLY2
DEC D
JR NZ,SHFRT_E2
CALL MOVDLY4
POP DE
POP HL
DEC E
JR NZ,SHFRT_E1
POP BC
RET

```

```

ROTBIT_E: LD A,(MAXCOL)
LD B,A      ;ROT COL PORT
ROTBIT_E1: LD A,(IY+0)
RL A
LD (IY+0),A
INC IY
DJNZ ROTBIT_E1
RET

```

```

-----
: NAME:      SHIFT_W
: DESCRIPTION: shift PREDSP to DSPBUF
: DIRECTION: right to left
-----

```

```

SHIFT_W: LD A,(MAXCOL)
LD B,A
LD HL,PREDSPR
SHF_W1: CALL SHFROT_W
INC HL
DJNZ SHF_W1

```

```

LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

```

```

SHFROT_W: PUSH BC
LD E,08H      ;ROT 8 BIT
SHFRT_W1: PUSH HL
LD IY,DSPBUFR ;address of display buffer
LD A,(MAXCOL)
LD C,A
LD B,0
ADD IY,BC
DEC IY

```

```

LD D,NUMROW ;NUMBER OF ROW
PUSH DE
LD E,00H
SHFRT_W2: PUSH IY ;process RED
RRC (HL)
CALL ROTBIT_W
POP IY

PUSH IY
PUSH HL
LD BC,NUMROW*NUMCOL
ADD IY,BC
ADD HL,BC
RRC (HL) ;process GREEN
CALL ROTBIT_W
POP HL
POP IY

LD A,(MAXCOL) ;point to next row of DSPBUF
LD B,0
LD C,A
ADD IY,BC
ADD HL,BC
CALL MOVDLY2
DEC D
JR NZ,SHFRT_W2
CALL MOVDLY4
POP DE
POP HL
DEC E
JR NZ,SHFRT_W1
POP BC
RET

ROTBIT_W: LD A,(MAXCOL)
LD B,A ;ROT COL PORT
RTBIT_W1: LD A,(IY+0)
RR A
LD (IY+0),A
DEC IY
DJNZ RTBIT_W1
RET

```

```

;-----
; NAME: ROTIN_S
; DESCRIPTION: shift DSPBUF and instead by PREDSP
; DIRECTION: north to south
;-----

```

```

ROTN_S: LD IX,PREDSPR
LD IY,DSPBUFR
LD D,NUMROW-2 ;find start add. of last row-1
LD A,(MAXCOL)
LD B,0
LD C,A

```

เอกสารนี้เป็น ROTIN\_S1: ADD IY,BC สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาติให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JR NZ,RTIN_S1
LD (PAPER+0),IX ;start add. of predsp (row 0)
LD (PAPER+2),IY ;start add. of dspbuf (lastrow-1)
LD (PAPER+12H),IY
ADD IY,BC
LD (PAPER+4),IY ;start add. of dspbuf (lastrow)
LD (PAPER+14H),IY
LD E,NUMROW-1
LD A,NUMROW-1
LD (PAPER+6),A ;shift loop
RTIN_S2: LD A,(PAPER+6)
LD D,A
RTIN_S3: LD IX,(PAPER+2) ;process RED color
LD IY,(PAPER+4)
LD A,(MAXCOL)
LD B,A
RTIN_S4: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_S4

PUSH IX
PUSH IY
LD IX,(PAPER+2) ;process GREEN color
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
LD A,(MAXCOL)
LD B,A
RTIN_S41:LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_S41
POP IY
POP IX

LD A,(MAXCOL) ;load BC with MAXCOL+MAXCOL
ADD A,A
LD B,0
LD C,A'
XOR A' ;clear A and CARRY
PUSH IX ;load HL with IX
POP HL
SBC HL,BC
PUSH HL
POP IX ;load IX with HL
LD (PAPER+2),IX
XOR A
PUSH IY
POP HL
SBC HL,BC
PUSH HL
POP IY
LD (PAPER+4),IY

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEC D
JR NZ,RTIN_S3
CALL MOVDLY
LD IX,(PAPER+0) ;PREDSP
LD IY,(PAPER+4) ;DSPBUF
LD A,(MAXCOL)
LD B,A
RTIN_S5: LD A,(IX+0) ;insert PREDSPP to DSPBUFR
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_S5

```

```

PUSH IX
PUSH IY
LD IX,(PAPER+0) ;PREDSP
LD IY,(PAPER+4) ;DSPBUF
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
LD A,(MAXCOL)
LD B,A
RTIN_S51:LD A,(IX+0) ;insert PREDSPP to DSPBUFR
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_S51
POP IY
POP IX

LD (PAPER+0),IX ;point to PREDSP next row
LD IX,(PAPER+12H)
LD (PAPER+2H),IX
LD IY,(PAPER+14H)
LD (PAPER+4H),IY
LD A,(PAPER+6) ;decrement shift loop
DEC A
LD (PAPER+6),A
CALL MOVDLY4
DEC E
JP NZ,RTIN_S2

```

```

LD IX,(PAPER+0) ;process RED color
LD IY,(PAPER+4)
LD A,(MAXCOL)
LD B,A
RTIN_S6: LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_S6

```

```

LD IX,(PAPER+0) ;process GREEN color
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL

```

```
ADD IX,BC
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD IY,BC
LD A,(MAXCOL)
LD B,A
RTIN_S61:LD A,(IX+0)
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_S61

LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

```

```

-----
; NAME:      ROTIN_N
; DESCRIPTION: shift DSPBUF and instead by PREDSP
; DIRECTION:  south to north
-----

```

```

ROTIN_N:LD IX,PREDSPR
LD IY,DSPBUFR
LD D,NUMROW-1 ;find start add. of last row-1
LD A,(MAXCOL)
LD B,0
LD C,A
RTIN_N1: ADD IX,BC
DEC D
JR NZ,RTIN_N1
LD (PAPER+0),IX ;start add. of predsp (row 0)
LD (PAPER+4),IY ;start add. of dspbuf (firstrow)
LD (PAPER+14H),IY
ADD IY,BC
LD (PAPER+2),IY ;start add. of dspbuf (firstrow+1)
LD (PAPER+12H),IY
LD E,NUMROW-1
LD A,NUMROW-1
LD (PAPER+6),A ;shift loop
RTIN_N2: LD A,(PAPER+6)
LD D,A
RTIN_N3: LD IX,(PAPER+2)
LD IY,(PAPER+4)
LD A,(MAXCOL)
LD B,A
RTIN_N4: LD A,(IX+0) ;process RED color
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_N4

PUSH IX
PUSH IY
LD IX,(PAPER+2)
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD A,(MAXCOL)
LD B,A
RTN_N41:LD A,(IX+0) ;process GREEN color
LD (IY+0),A
INC IX
INC IY
DJNZ RTN_N41
POP IY
POP IX

LD (PAPER+2),IX
LD (PAPER+4),IY
DEC D
JR NZ,RTN_N3
CALL MOVDLY
LD IX,(PAPER+0) ;PREDSPPR
LD IY,(PAPER+4) ;DSPBUFR
LD A,(MAXCOL)
LD B,A
RTN_N5: LD A,(IX+0) ;insert PREDSPPR to DSPBUFR
LD (IY+0),A
INC IX
INC IY
DJNZ RTN_N5

PUSH IX
PUSH IY
LD IX,(PAPER+0) ;PREDSPPG
LD IY,(PAPER+4) ;DSPBUFG
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
LD A,(MAXCOL)
LD B,A
RTN_N51:LD A,(IX+0) ;insert PREDSPPG to DSPBUFG
LD (IY+0),A
INC IX
INC IY
DJNZ RTN_N51
POP IY
POP IX

LD A,(MAXCOL) ;load BC with MAXCOL+MAXCOL
ADD A,A
LD B,0
LD C,A
XOR A ;clear A and CARRY
PUSH IX ;load HL with IX
POP HL
SBC HL,BC
PUSH HL
POP IX ;load IX with HL
LD (PAPER+0),IX ;point to PREDSPP next row
LD IX,(PAPER+12H)
LD (PAPER+2H),IX
LD IY,(PAPER+14H)
LD (PAPER+4H),IY

```

```

LD A,(PAPER+6)      ;decrement shift loop
DEC A
LD (PAPER+6),A
CALL MOVDLY4
DEC E
JP NZ,RTIN_N2
LD IX,(PAPER+0)
LD IY,(PAPER+4)
LD A,(MAXCOL)
LD B,A
RTIN_N6: LD A,(IX+0)  ;process RED color
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_N6

LD IX,(PAPER+0)
LD IY,(PAPER+4)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
LD A,(MAXCOL)
LD B,A
RTIN_N61:LD A,(IX+0) ;process GREEN color
LD (IY+0),A
INC IX
INC IY
DJNZ RTIN_N61

LD A,DISPAUSE
LD (PAUSEN),A
CALL PAUSE
RET

;-----
; NAME:      ROTIN_E
; DESCRIPTION: shift DSPBUF and instead by PREDSP
; DIRECTION: left to right
;-----

ROTIN_E: LD IX,PREDSPR
LD A,(MAXCOL)
LD B,A -
LD (PAPER),A
LD IY,DSPBUFR      ;IY POINT TO DISPLAY BUFFER RAM
RTIN_E1: PUSH BC      ;SAVE ROT CON. TO STACK
LD H,0FEH          ;AND,OR CONTENTS
LD L,8              ;LOOP 8 TIME (1 BYTE)
LD (PAPER+3),IY
LD (PAPER+1),IX
RTIN_E2: LD D,NUMROW ;LOOP NUMROW TIME

RTIN_E3: PUSH IY
CALL RTIN_E4      ;process RED color
POP IY

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PUSH IY
LD BC,NUMROW*NUMCOL
ADD IY,BC
ADD IX,BC
CALL RTIN_E4 ;process GREEN color
POP IY
POP IX

LD A,(MAXCOL)
LD B,0
LD CA
ADD IY,BC
ADD IX,BC
DEC D
JR NZ,RTIN_E3

CALL MOVDLY5
LD IX,(PAPER+1)
LD IY,(PAPER+3)
SLA H
DEC L
JR NZ,RTIN_E2

POP BC
INC IY
INC IX
LD A,(PAPER)
DEC A
LD (PAPER),A
DJNZ RTIN_E1

LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

RTIN_E4: PUSH IY
LD A,(PAPER) ;LOAD ROT CONTROL
LD B,A
RTIN_E41:RL (IY+0) ;ROT DISPLAY BUF B BYTE
INC IY
DJNZ RTIN_E41
POP IY
LD A,(IY+0) ;LOAD DISPLAY BUF
AND H
LD B,A
LD A,H
CPL
AND (IX+0)
OR B
LD (IY+0),B
RET

```

```

; NAME: ROTIN_W
; DESCRIPTION: shift DSPBUF and instead by PREDSP
; DIRECTION: right to left

```

```

RTIN_W:LD A,(MAXCOL)
LD B,0
LD C,A
LD IX,PREDSPR
ADD IX,BC
DEC IX
LD IY,DSPBUFR
ADD IY,BC
DEC IY
LD A,(MAXCOL)
LD B,A
LD (PAPER),A

RTIN_W1:PUSH BC ;SAVE ROT CON. TO STACK
LD H,07FH ;AND,OR CONTENTS
LD L,8 ;LOOP 8 TIME (1 BYTE)
LD (PAPER+3),IY ;1703-1704 SAVE IY
LD (PAPER+1),IX ;1701-1702 SAVE IX

RTIN_W2:LD D,NUMROW ;LOOP NUMROW TIME

RTIN_W3: PUSH IY
CALL RTIN_W4 ;process RED color
POP IY

PUSH IX
PUSH IY
LD BC,NUMROW*NUMCOL
ADD IY,BC
ADD IX,BC
CALL RTIN_W4 ;process GREEN color
POP IY
POP IX

LD A,(MAXCOL)
LD B,0
LD C,A
ADD IY,BC
ADD IX,BC
DEC D
JR NZ,RTIN_W3

CALL MOVDLY5
LD IX,(PAPER+1)
LD IY,(PAPER+3)
SRL H
DEC L
JR NZ,RTIN_W2

POP BC
DEC IY
DEC IX
LD A,(PAPER)
DEC A
LD (PAPER),A
DJNZ RTIN_W1

LD A,DISPAUSE
LD (PAUSE),A

```

```
CALL PAUSE
RET
```

```
RTN_W4: PUSH IY
LD A,(PAPER) ;LOAD ROT CONTROL
LD B,A
RTN_W41: RR (IY+0) ;ROT DISPLAY BUF 8 BYTE
DEC IY
DJNZ RTN_W41
POP IY
LD A,(IY+0) ;LOAD DISPLAY BUF
AND H
LD B,A
LD A,H
CPL
AND (IX+0)
OR B
LD (IY+0),A
RET
```

```
;-----
; NAME: OPEN_H
; DESCRIPTION: shift DSPBUF out of center in hor.direction
; and insert PREDSP
;-----
OPEN_H: LD B,0 ;load BC with half of column
LD A,(HALFCOL)
LD C,A
LD IX,PREDSPR
ADD IX,BC ;load IX with PREDSP+HALFCOL
LD IY,DSPBUFR
ADD IY,BC
LD A,(HALFCOL)
LD B,A
LD (PAPER),A
LD (PAPER+03H),IY
LD (PAPER+13H),IY
LD (PAPER+05H),IX
LD (PAPER+15H),IX
DEC IY
DEC IX
LD (PAPER+09H),IY
LD (PAPER+19H),IY
LD (PAPER+0BH),IX
LD (PAPER+1BH),IX
OPEN_H1: PUSH BC
LD A,0FEH
LD (PAPER+01H),A
LD A,07FH
LD (PAPER+02H),A
LD L,8
OPEN_H2: LD D,NUMROW
```

```
OPEN_H3: LD IY,(PAPER+19H)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD IX,(PAPER+1BH)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
PUSH IY
LD A,(PAPER+00H)
LD B,A

```

```

OPEN_H4: RR (IY+0)
DEC IY
DJNZ OPEN_H4
POP IY
LD A,(PAPER+02H)
LD B,A
CALL PREIN
LD IY,(PAPER+19H)
LD IX,(PAPER+1BH)
PUSH IY
LD A,(PAPER+00H)
LD B,A

```

```

OPEN_H41:RR (IY+0)
DEC IY
DJNZ OPEN_H41
POP IY
LD A,(PAPER+02H)
LD B,A
CALL PREIN
LD (PAPER+19H),IY
LD (PAPER+1BH),IX
LD IY,(PAPER+13H)
LD IX,(PAPER+15H)
LD BC,NUMROW*NUMCOL
ADD IX,BC
ADD IY,BC
PUSH IY
LD A,(PAPER+00H)
LD B,A

```

```

OPEN_H5: RL (IY+0)
INC IY
DJNZ OPEN_H5
POP IY
LD A,(PAPER+01H)
LD B,A
CALL PREIN

```

```

LD IY,(PAPER+13H)
LD IX,(PAPER+15H)
LD BC,NUMROW*NUMCOL
PUSH IY
LD A,(PAPER+00H)
LD B,A

```

```

OPEN_H51:RL (IY+0)
INC IY
DJNZ OPEN_H51
POP IY
LD A,(PAPER+01H)
LD B,A

```

เอกสารนี้เป็นเอกสารที่จัดทำไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL PREIN  
 LD (PAPER+13H),IY  
 LD (PAPER+15H),IX

CALL MOVDLY2  
 DEC D  
 JP NZ,OPEN\_H3

CALL MOVDLY5  
 LD IY,(PAPER+03H)  
 LD IX,(PAPER+05H)  
 LD (PAPER+13H),IY  
 LD (PAPER+15H),IX  
 LD IY,(PAPER+09H)  
 LD IX,(PAPER+0BH)  
 LD (PAPER+19H),IY  
 LD (PAPER+1BH),IX  
 LD A,(PAPER+01H)

SLA A  
 LD (PAPER+01H),A  
 LD A,(PAPER+02H)  
 SRL A  
 LD (PAPER+02H),A  
 DEC L

JP NZ,OPEN\_H2  
 POP BC  
 DEC IY  
 DEC IX  
 LD (PAPER+09H),IY  
 LD (PAPER+19H),IY  
 LD (PAPER+0BH),IX  
 LD (PAPER+1BH),IX  
 LD IY,(PAPER+03H)  
 LD IX,(PAPER+05H)

INC IY  
 INC IX  
 LD (PAPER+03H),IY  
 LD (PAPER+13H),IY  
 LD (PAPER+05H),IX  
 LD (PAPER+15H),IX  
 LD A,(PAPER+00H)  
 DEC A  
 LD (PAPER+00H),A  
 DEC B  
 JP NZ,OPEN\_H1

LD A,DISPAUSE  
 LD (PAUSE),A  
 CALL PAUSE  
 RET

PREIN: LD A,(IY+0)  
 AND B  
 LD H,A  
 LD A,B  
 CPL

AND (IX+0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OR H
LD (Y+0),A
LD A,(MAXCOL)
LD B,0
LD C,A
ADD Y,BC
ADD X,BC
RET

```

```

-----
; NAME:      CLOSE_H
; DESCRIPTION: shift DSPBUF to center in hor.direction
;            and insert PREDSP
-----

```

```

CLOSE_H: LD  A,(MAXCOL)
          LD  B,0
          LD  C,A
          LD  IX,PREDSPR
          ADD IX,BC
          DEC IX
          LD  IY,DSPBUFR
          ADD IY,BC
          DEC IY
          LD  A,(HALPCOL)
          LD  B,A
          LD  (PAPER+00H),A
          LD  (PAPER+03H),IY ;start right DSPBUF
          LD  (PAPER+13H),IY ;work right DSPBUF
          LD  (PAPER+05H),IX ;start right PREDSP
          LD  (PAPER+15H),IX ;work right PREDSP
          LD  IY,DSPBUFR
          LD  IX,PREDSPR
          LD  (PAPER+09H),IY ;start left DSPBUF
          LD  (PAPER+19H),IY ;work left DSPBUF
          LD  (PAPER+0BH),IX ;start left PREDSP
          LD  (PAPER+1BH),IX ;work left PREDSP

```

```

CLOSE_H1: PUSH BC
          LD  A,07FH
          LD  (PAPER+01H),A ;AND right contents
          LD  A,0FEH
          LD  (PAPER+02H),A ;AND left contents
          LD  L,8

```

```

CLOSE_H2: LD  D,NUMROW

```

```

CLOSE_H3: LD  IY,(PAPER+13H)
          LD  IX,(PAPER+15H)
          LD  BC,NUMROW*NUMCOL ;process GREEN color
          ADD IY,BC
          ADD IX,BC
          CALL CLOSE_H5
          LD  IY,(PAPER+13H)
          LD  IX,(PAPER+15H)
          CALL CLOSE_H5 ;process RED color
          LD  (PAPER+13H),IY
          LD  (PAPER+15H),IX

```

```

LD IY,(PAPER+19H)
LD IX,(PAPER+1BH)
LD BC,NUMROW*NUMCOL ;process GREEN color
ADD IY,BC
ADD IX,BC
CALL CLOSE_H4
LD IY,(PAPER+19H)
LD IX,(PAPER+1BH)
CALL CLOSE_H4 ;process RED color
LD (PAPER+19H),IY
LD (PAPER+1BH),IX

```

```

DEC D
JP NZ,CLOSE_H3

```

```

CALL MOVDLY5
LD IY,(PAPER+03H)
LD IX,(PAPER+05H)
LD (PAPER+13H),IY
LD (PAPER+15H),IX
LD IY,(PAPER+09H)
LD IX,(PAPER+0BH)
LD (PAPER+19H),IY
LD (PAPER+1BH),IX
LD A,(PAPER+01H) ;AND right contents

```

```

SRL A
LD (PAPER+01H),A
LD A,(PAPER+02H) ;AND left contents
SLA A
LD (PAPER+02H),A
DEC L
JP NZ,CLOSE_H2

```

```

POP BC
INC IY
INC IX
LD (PAPER+09H),IY
LD (PAPER+19H),IY
LD (PAPER+0BH),IX
LD (PAPER+1BH),IX
LD IY,(PAPER+03H)
LD IX,(PAPER+05H)

```

```

DEC IY
DEC IX
LD (PAPER+03H),IY
LD (PAPER+13H),IY
LD (PAPER+05H),IX
LD (PAPER+15H),IX
LD A,(PAPER+00H)
DEC A
LD (PAPER+00H),A
DEC B
JP NZ,CLOSE_H1

```

```

LD A,DISPAUSE
LD (PAUSEN),A

```

CALL PAUSE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

CLOSE\_H4:PUSH IY

LD A,(PAPER+00H) ;rotate contents

LD B,A

CLOSEH41:RL (IY+0) ;rotate left side

INC IY

DJNZ CLOSEH41

POP IY

LD A,(PAPER+02H)

LD B,A

CALL PREIN

RET

CLOSE\_H5:PUSH IY

LD A,(PAPER+00H)

LD B,A

CLOSEH51:RR (IY+0) ;rotate right side

DEC IY

DJNZ CLOSEH51

POP IY

LD A,(PAPER+01H)

LD B,A

CALL PREIN

RET

```

:-----
: NAME:    RANDOM
: DESCRIPTION:  PREDSP random to DSPBUF
: DIRECTION:  full screen
:-----

```

```

RANDOM: CALL CLRDSP
      CALL DSPDELAY

```

LD IY,DSPBUFR

LD IX,PREDSPR

LD (PAPER+0),IX ;start add. of predsp (row 0)

LD (PAPER+4),IY ;start add. of dspbuf (row 0)

LD A,NUMROW

LD (PAPER+6),A ;row counter

LD A,0 -

LD (PAPER+8),A

LD B,8

RAND\_0: PUSH BC

LD IX,(PAPER+0) ;PREDSP

LD IY,(PAPER+4) ;DSPBUF

LD D,2

RAND\_1: LD HL,RANDTAB

LD A,(PAPER+8)

LD B,0

LD C,A

ADD HL,BC

LD E,8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RAND_2: PUSH IX
        PUSH IY
        LD BC,NUMROW*NUMCOL
        ADD IX,BC
        ADD IY,BC
        LD A,(MAXCOL)
        LD B,A
RAND_3: LD A,(IX+0)      ;insert PREDSP to DSPBUF
        AND (HL)
        OR (IY+0)
        LD (IY+0),A
        INC IX
        INC IY
        DJNZ RAND_3      ;until one row complete
        POP IY
        POP IX

        LD A,(MAXCOL)
        LD B,A
RAND_4: LD A,(IX+0)      ;insert PREDSP to DSPBUF
        AND (HL)
        OR (IY+0)
        LD (IY+0),A
        INC IX
        INC IY
        DJNZ RAND_4      ;until one row complete

        LD B,0
        LD C,8
        ADD HL,BC
        DEC E
        JR NZ,RAND_2      ;until 8 row complete
        DEC D
        JR NZ,RAND_1      ;until 16 row complete (8*3)
        CALL DSPDELAY
        CALL DSPDELAY
        CALL DSPDELAY
        CALL DSPDELAY
        LD A,(PAPER+8)
        INC A
        LD (PAPER+8),A    ;point to next column of random table
        POP BC
        DJNZ RAND_0

RANDOM_E:
        LD A,DISPAUSE
        LD (PAUSEN),A
        CALL PAUSE
        RET

```

RANDTAB:

```

DB 0000001B,00100000B,0000100B,1000000B,00001000B,00000010B,00010000B,01000000B
DB 00001000B,01000000B,00010000B,00000010B,00100000B,10000000B,00000001B,00000100B
DB 00100000B,00000100B,00000001B,00001000B,01000000B,00010000B,00000010B,10000000B
DB 00000010B,01000000B,00010000B,10000000B,00000100B,00000001B,00100000B,00001000B
DB 00000001B,00000100B,10000000B,00100000B,00000010B,00001000B,00010000B,01000000B
DB 00010000B,01000000B,00001000B,00000001B,10000000B,00000010B,00100000B,00000100B
DB 10000000B,00000001B,00100000B,00000100B,00010000B,01000000B,00000010B,00001000B
DB 00100000B,00000100B,10000000B,00001000B,00000010B,00010000B,01000000B,00000001B

```

```

-----
; NAME:      MIXT_E
; DESCRIPTION: shift PREDSP to DSPBUF
; DIRECTION: left to right
-----

```

MIXT\_E: CALL CLRDSF

```

LD A,(MAXCOL)
LD B,0
LD C,A
LD HL,PREDSPR
ADD HL,BC
DEC HL
LD B,A
MIX_EI: CALL MIXROT_E
DEC HL
DJNZ MIX_EI
LD A,(MAXCOL)
LD B,A
LD HL,PREDSPG
LD B,A
MIX_WI: CALL MIXROT_W
INC HL
DJNZ MIX_WI
LD A,DISPAUSE
LD (PAUSE),A
CALL PAUSE
RET

```

MIXROT\_E:PUSH BC

```
LD E,08H ;ROT 8 BIT
```

MIXRT\_E1:PUSH HL

```
LD IY,DSPBUFR ;address of display buffer
LD D,NUMROW ;NUMBER OF ROW
PUSH DE

```

MIXRT\_E2:PUSH IY ;process RED

```
RLC (HL)
CALL ROTBIT_E
POP IY

```

```
LD A,(MAXCOL) ;point to next row of DSPBUF
```

```
LD B,0
```

```
LD C,A
```

```
ADD IY,BC
```

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADD HL,BC
CALL MOVDLY2
DEC D
JR NZ,MIXRT_E2
CALL MOVDLY4
POP DE
POP HL
DEC E
JR NZ,MIXRT_E1
POP BC
RET

```

```

MIXROT_W: PUSH BC
          LD  E,08H      ;ROT 8 BIT
MIXRT_W1: PUSH HL
          LD  IY,DSPBUF  ;address of display buffer
          LD  A,(MAXCOL)
          LD  C,A
          LD  B,0
          ADD IY,BC
          DEC IY
          LD  D,NUMROW  ;NUMBER OF ROW
          PUSH DE
          LD  E,00H
MIXRT_W2: PUSH IY      ;process RED
          RRC (HL)
          CALL ROTBIT_W
          POP  IY
          LD  A,(MAXCOL) ;point to next row of DSPBUF
          LD  B,0
          LD  C,A
          ADD IY,BC
          ADD HL,BC
          CALL MOVDLY2
          DEC D
          JR  NZ,MIXRT_W2
          CALL MOVDLY4
          POP DE
          POP HL
          DEC E
          JR  NZ,MIXRT_W1
          POP BC
          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

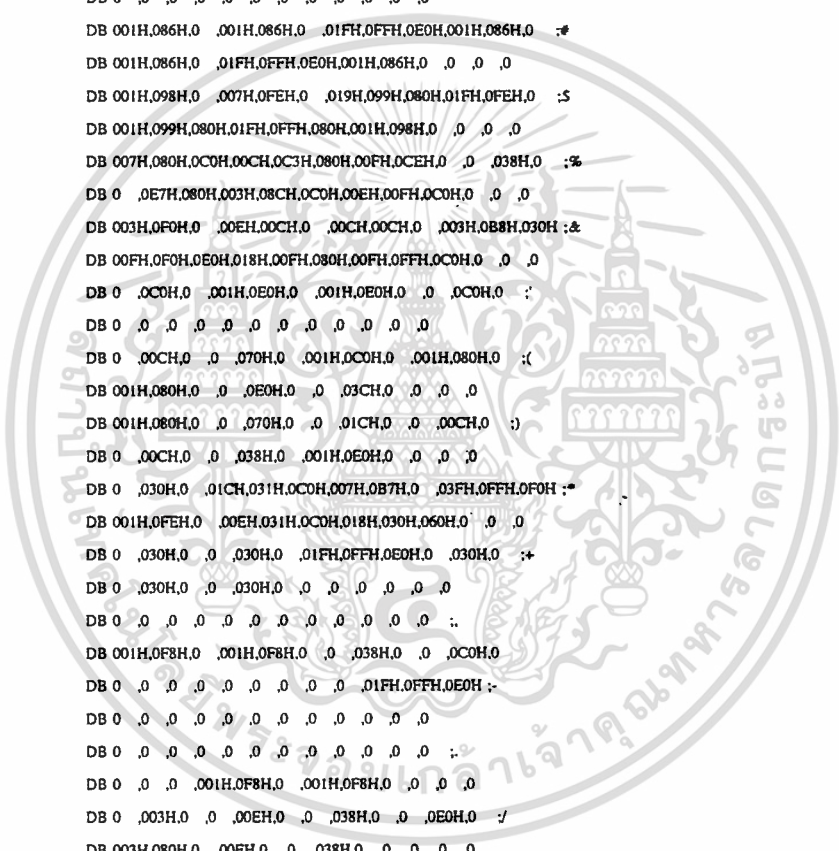
```

:FILENAME GEN.FON
:DESCRIPTION this file have the ENGLISH,THAI font
:START DATE 28 DECEMBER 1994
.command -l
.opdef DB,.DB
.opdef DS,.RS

.ORG GEN

:Under this line is ENGFONT1: (fon size=8X8, 8X16)
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;space
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,060H,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ;!
DB 0 ,060H,0 ,0 ,0 ,0 ,0 ,0 ,060H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 006H,006H,0 ,00FH,00FH,0 ,00FH,00FH,0 ,006H,006H,0 ,0 ,0 ,0 ,0 ,0 ;"
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 001H,086H,0 ,001H,086H,0 ,01FH,0FFH,0E0H,001H,086H,0 ,0 ,0 ,0 ,0 ,0 ;#
DB 001H,086H,0 ,01FH,0FFH,0E0H,001H,086H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 001H,098H,0 ,007H,0FEH,0 ,019H,099H,080H,01FH,0FEH,0 ,0 ,0 ,0 ,0 ,0 ;$
DB 001H,099H,080H,01FH,0FFH,080H,001H,098H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 007H,080H,0C0H,00CH,0C3H,080H,00FH,0CEH,0 ,0 ,038H,0 ,0 ,0 ,0 ,0 ;%
DB 0 ,0E7H,080H,003H,08CH,0C0H,00EH,00FH,0C0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 003H,0F0H,0 ,00EH,00CH,0 ,00CH,00CH,0 ,003H,0B8H,030H ,0 ,0 ,0 ,0 ,0 ,0 ;&
DB 00FH,0F0H,0E0H,018H,00FH,080H,00FH,0FFH,0C0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0C0H,0 ,001H,0E0H,0 ,001H,0E0H,0 ,0 ,0C0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ;
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,00CH,0 ,0 ,070H,0 ,001H,0C0H,0 ,001H,080H,0 ,0 ,0 ,0 ,0 ,0 ;(
DB 001H,080H,0 ,0 ,0E0H,0 ,0 ,03CH,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 001H,080H,0 ,0 ,070H,0 ,0 ,01CH,0 ,0 ,00CH,0 ,0 ,0 ,0 ,0 ,0 ,0 ;)
DB 0 ,00CH,0 ,0 ,038H,0 ,001H,0E0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,030H,0 ,01CH,031H,0C0H,007H,0B7H,0 ,03FH,0FFH,0F0H ,0 ,0 ,0 ,0 ,0 ,0 ;*
DB 001H,0FEH,0 ,00EH,031H,0C0H,018H,030H,060H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,030H,0 ,0 ,030H,0 ,01FH,0FFH,0E0H,0 ,030H,0 ,0 ,0 ,0 ,0 ,0 ,0 ;+
DB 0 ,030H,0 ,0 ,030H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;,
DB 001H,0F8H,0 ,001H,0F8H,0 ,0 ,038H,0 ,0 ,0C0H,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;-
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;.
DB 0 ,0 ,0 ,001H,0F8H,0 ,001H,0F8H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,003H,0 ,0 ,00EH,0 ,0 ,038H,0 ,0 ,0E0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ;/
DB 003H,080H,0 ,00EH,0 ,0 ,038H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 001H,0FEH,0 ,00EH,001H,080H,018H,0 ,060H,018H,0 ,060H ,0 ,0 ,0 ,0 ,0 ,0
DB 018H,0 ,060H,01CH,0 ,0E0H,007H,0FFH,080H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,030H,0 ,003H,0F0H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,0 ,0 ,0 ,0 ;:
DB 0 ,030H,0 ,0 ,030H,0 ,003H,0FFH,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 003H,0FEH,0 ,01CH,001H,0C0H,01CH,001H,0C0H,0 ,03FH,0 ,0 ,0 ,0 ,0 ,0 ,0 ;:
DB 003H,0C0H,0 ,01CH,0 ,060H,01FH,0FFH,0E0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 007H,0FEH,0 ,01CH,001H,0C0H,0 ,0 ,0C0H,0 ,0FFH,080H ,0 ,0 ,0 ,0 ,0 ,0 ;:
DB 0 ,0 ,0C0H,01CH,001H,0C0H,007H,0FFH,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,00FH,0 ,0 ,03BH,0 ,0 ,0E3H,0 ,003H,083H,0 ,0 ,0 ,0 ,0 ,0 ,0 ;:
DB 00EH,003H,0 ,01FH,0FFH,0E0H,0 ,003H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 007H,0FFH,0C0H,006H,0 ,0 ,006H,07EH,0 ,007H,081H,0C0H ,0 ,0 ,0 ,0 ,0 ,0 ;:
DB 0 ,0 ,060H,00CH,0 ,0E0H,007H,0FFH,080H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,07FH,080H,003H,081H,0E0H,00EH,0 ,0 ,00DH,0FFH,080H ,0 ,0 ,0 ,0 ,0 ,0 ;:
DB 00FH,0 ,0E0H,00EH,0 ,060H,003H,0FFH,0C0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 00FH,0FFH,0C0H,00CH,001H,0C0H,0 ,007H,0 ,0 ,01CH,0 ,0 ,0 ,0 ,0 ,0 ,0 ;:

```

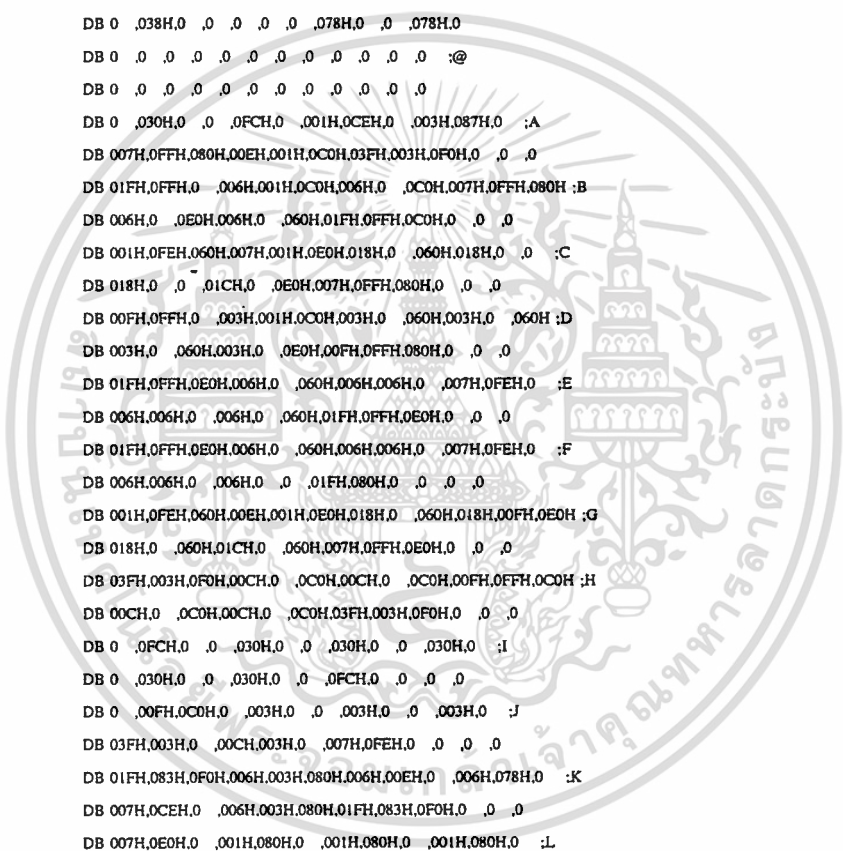


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 0 ,030H,0 ,0 ,060H,0 ,0 ,060H,0 ,0 ,0 ,0
DB 001H,0FEH,0 ,00EH,001H,0C0H,00CH,0 ,0C0H,007H,0FFH,080H :8
DB 00EH,001H,0C0H,00CH,0 ,0C0H,007H,0FFH,0 ,0 ,0 ,0
DB 003H,0FFH,0 ,01CH,001H,080H,018H,0 ,0C0H,01CH,003H,0C0H :9
DB 003H,0FCH,0C0H,0 ,0 ,080H,007H,0FFH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0 ,0 ::
DB 0 ,0 ,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0 ,0
DB 001H,0F8H,0 ,001H,0F8H,0 ,0 ,0 ,0 ,001H,0F8H,0 ::
DB 001H,0F8H,0 ,0 ,038H,0 ,0 ,0C0H,0 ,0 ,0 ,0
DB 0 ,001H,080H,0 ,01EH,0 ,001H,0E0H,0 ,01EH,0 ,0 ,0 :<
DB 007H,080H,0 ,0 ,078H,0 ,0 ,007H,080H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,01FH,0FFH,0E0H,0 ,0 ,0 ,0 :=-
DB 01FH,0FFH,0E0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 006H,0 ,0 ,001H,0E0H,0 ,0 ,01EH,0 ,0 ,001H,0E0H :>
DB 0 ,007H,080H,0 ,078H,0 ,007H,080H,0 ,0 ,0 ,0
DB 003H,0FFH,0 ,00EH,0 ,0E0H,00EH,0 ,060H,0 ,007H,080H :?
DB 0 ,038H,0 ,0 ,0 ,0 ,0 ,078H,0 ,0 ,078H,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 :@
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,030H,0 ,0 ,0FCH,0 ,001H,0CEH,0 ,003H,087H,0 :A
DB 007H,0FFH,080H,00EH,001H,0C0H,03FH,003H,0F0H,0 ,0 ,0
DB 01FH,0FFH,0 ,006H,001H,0C0H,006H,0 ,0C0H,007H,0FFH,080H :B
DB 006H,0 ,0E0H,006H,0 ,060H,01FH,0FFH,0C0H,0 ,0 ,0
DB 001H,0FEH,060H,007H,001H,0E0H,018H,0 ,060H,018H,0 ,0 :C
DB 018H,0 ,0 ,01CH,0 ,0E0H,007H,0FFH,080H,0 ,0 ,0
DB 00FH,0FFH,0 ,003H,001H,0C0H,003H,0 ,060H,003H,0 ,060H :D
DB 003H,0 ,060H,003H,0 ,0E0H,00FH,0FFH,080H,0 ,0 ,0
DB 01FH,0FFH,0E0H,006H,0 ,060H,006H,006H,0 ,007H,0FEH,0 :E
DB 006H,006H,0 ,006H,0 ,060H,01FH,0FFH,0E0H,0 ,0 ,0
DB 01FH,0FFH,0E0H,006H,0 ,060H,006H,006H,0 ,007H,0FEH,0 :F
DB 006H,006H,0 ,006H,0 ,0 ,01FH,080H,0 ,0 ,0 ,0
DB 001H,0FEH,060H,00EH,001H,0E0H,018H,0 ,060H,018H,00FH,0E0H :G
DB 018H,0 ,060H,01CH,0 ,060H,007H,0FFH,0E0H,0 ,0 ,0
DB 03FH,003H,0F0H,00CH,0 ,0C0H,00CH,0 ,0C0H,00FH,0FFH,0C0H :H
DB 00CH,0 ,0C0H,00CH,0 ,0C0H,03FH,003H,0F0H,0 ,0 ,0
DB 0 ,0FCH,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 :I
DB 0 ,030H,0 ,0 ,030H,0 ,0 ,0FCH,0 ,0 ,0 ,0
DB 0 ,00FH,0C0H,0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0 :J
DB 03FH,003H,0 ,00CH,003H,0 ,007H,0FEH,0 ,0 ,0 ,0
DB 01FH,083H,0F0H,006H,003H,080H,006H,00EH,0 ,006H,078H,0 :K
DB 007H,0CEH,0 ,006H,003H,080H,01FH,083H,0F0H,0 ,0 ,0
DB 007H,0E0H,0 ,001H,080H,0 ,001H,080H,0 ,001H,080H,0 :L
DB 001H,080H,0 ,001H,080H,060H,007H,0FFH,0E0H,0 ,0 ,0
DB 03CH,0 ,0F0H,01EH,001H,0E0H,01BH,087H,060H,019H,0CEH,060H :M
DB 018H,0FCH,060H,018H,030H,060H,03FH,003H,0F0H,0 ,0 ,0
DB 03CH,003H,0F0H,00FH,0 ,0C0H,00DH,0C0H,0C0H,00CH,070H,0C0H :N
DB 00CH,01CH,0C0H,00CH,007H,0C0H,03FH,001H,0F0H,0 ,0 ,0
DB 001H,0FEH,0 ,007H,001H,0C0H,01CH,0 ,060H,018H,0 ,060H :O
DB 018H,0 ,060H,01CH,0 ,0E0H,007H,0FFH,080H,0 ,0 ,0
DB 01FH,0FEH,0 ,006H,001H,0C0H,006H,0 ,0C0H,006H,001H,0C0H :P
DB 007H,0FEH,0 ,006H,0 ,0 ,01FH,080H,0 ,0 ,0 ,0
DB 001H,0FEH,0 ,006H,001H,0C0H,018H,0 ,060H,018H,0 ,060H :Q
DB 018H,030H,060H,01CH,00FH,0E0H,007H,0FFH,0E0H,0 ,0 ,030H
DB 01FH,0FFH,0 ,006H,001H,0C0H,006H,0 ,0C0H,007H,0FFH,080H :R
DB 006H,01CH,0 ,006H,007H,0 ,01FH,083H,0F0H,0 ,0 ,0
DB 003H,0FCH,0 ,01CH,003H,080H,01CH,0 ,0 ,007H,0F0H,0 :S
DB 0 ,01FH,0 ,018H,001H,080H,01FH,0FFH,0 ,0 ,0 ,0

```

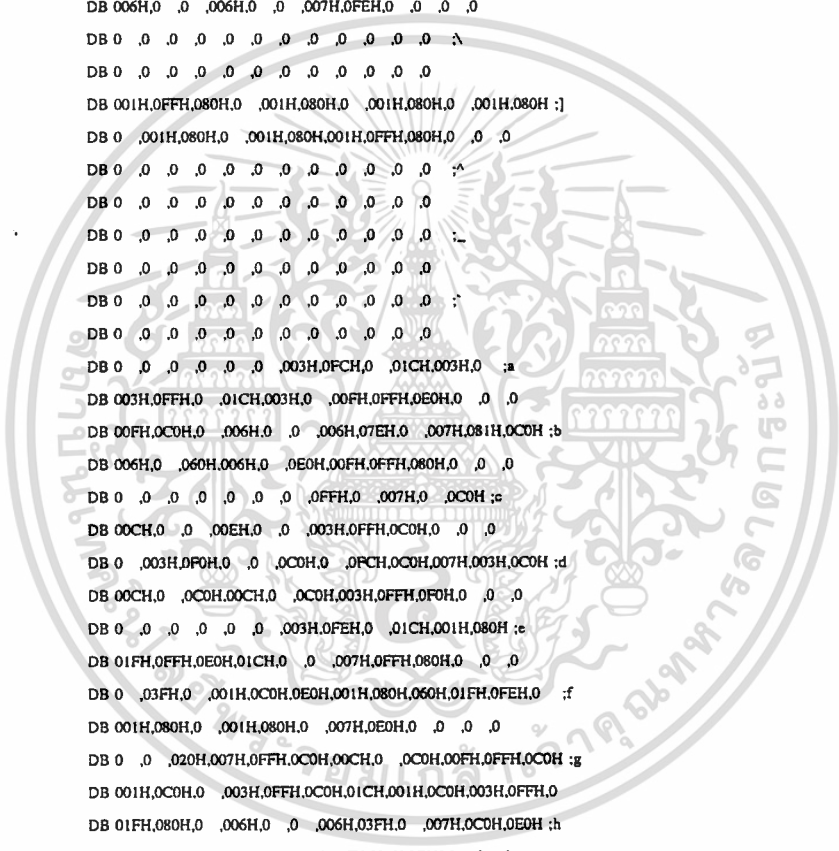


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 01FH,0FFH,0E0H,018H,030H,060H,0 ,030H,0 ,0 ,030H,0 :T
DB 0 ,030H,0 ,0 ,030H,0 ,0 ,0FCH,0 ,0 ,0 ,0
DB 03FH,001H,0F0H,00CH,0 ,060H,00CH,0 ,060H,00CH,0 ,060H :U
DB 00CH,0 ,060H,00EH,0 ,0E0H,003H,0FFH,080H,0 ,0 ,0
DB 03FH,003H,0F0H,00CH,0 ,0C0H,006H,001H,080H,003H,003H,0 :V
DB 001H,086H,0 ,0 ,0CCH,0 ,0 ,078H,0 ,0 ,0 ,0
DB 03CH,0 ,0F0H,018H,0 ,060H,018H,030H,060H,00CH,078H,0C0H :W
DB 00FH,0CFH,0C0H,007H,087H,080H,003H,003H,0 ,0 ,0 ,0
DB 01FH,083H,0F0H,006H,001H,0C0H,001H,087H,0 ,0 ,07CH,0 :X
DB 0 ,0EEH,0 ,003H,083H,080H,01FH,003H,0F0H,0 ,0 ,0
DB 01FH,083H,0F0H,007H,001H,0C0H,001H,0C7H,0 ,0 ,07CH,0 :Y
DB 0 ,030H,0 ,0 ,030H,0 ,0 ,0FCH,0 ,0 ,0 ,0
DB 00FH,0FFH,0C0H,00CH,003H,080H,0 ,00EH,0 ,0 ,038H,0 :Z
DB 0 ,0E0H,0 ,003H,080H,0C0H,00FH,0FFH,0C0H,0 ,0 ,0
DB 007H,0FEH,0 ,006H,0 ,0 ,006H,0 ,0 ,006H,0 ,0 :{
DB 006H,0 ,0 ,006H,0 ,0 ,007H,0FEH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 001H,0FFH,080H,0 ,001H,080H,0 ,001H,080H,0 ,001H,080H :}
DB 0 ,001H,080H,0 ,001H,080H,001H,0FFH,080H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,0FCH,0 ,01CH,003H,0 :a
DB 003H,0FFH,0 ,01CH,003H,0 ,00FH,0FFH,0E0H,0 ,0 ,0
DB 00FH,0C0H,0 ,006H,0 ,0 ,006H,07EH,0 ,007H,081H,0C0H :b
DB 006H,0 ,060H,006H,0 ,0E0H,00FH,0FFH,080H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0FFH,0 ,007H,0 ,0C0H :c
DB 00CH,0 ,0 ,00EH,0 ,0 ,003H,0FFH,0C0H,0 ,0 ,0
DB 0 ,003H,0F0H,0 ,0 ,0C0H,0 ,0FCH,0C0H,007H,003H,0C0H :d
DB 00CH,0 ,0C0H,00CH,0 ,0C0H,003H,0FFH,0F0H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,0FEH,0 ,01CH,001H,080H :e
DB 01FH,0FFH,0E0H,01CH,0 ,0 ,007H,0FFH,080H,0 ,0 ,0
DB 0 ,03FH,0 ,001H,0C0H,0E0H,001H,080H,060H,01FH,0FEH,0 :f
DB 001H,080H,0 ,001H,080H,0 ,007H,0E0H,0 ,0 ,0 ,0
DB 0 ,0 ,020H,007H,0FFH,0C0H,00CH,0 ,0C0H,00FH,0FFH,0C0H :g
DB 001H,0C0H,0 ,003H,0FFH,0C0H,01CH,001H,0C0H,003H,0FFH,0
DB 01FH,080H,0 ,006H,0 ,0 ,006H,03FH,0 ,007H,0C0H,0E0H :h
DB 006H,0 ,060H,006H,0 ,060H,01FH,081H,0F0H,0 ,0 ,0
DB 0 ,03CH,0 ,0 ,03CH,0 ,0 ,0FCH,0 ,0 ,00CH,0 :i
DB 0 ,00CH,0 ,0 ,00CH,0 ,0 ,03FH,0 ,0 ,0 ,0
DB 0 ,007H,080H,0 ,007H,080H,0 ,01FH,080H,0 ,001H,080H :j
DB 0 ,001H,080H,003H,001H,080H,003H,003H,080H,0 ,0FCH,0
DB 01FH,080H,0 ,006H,0 ,0 ,006H,00FH,0C0H,006H,01EH,0 :k
DB 007H,0E0H,0 ,006H,03CH,0 ,01FH,007H,0E0H,0 ,0 ,0
DB 001H,0F8H,0 ,0 ,018H,0 ,0 ,018H,0 ,0 ,018H,0 :l
DB 0 ,018H,0 ,0 ,018H,0 ,0 ,0FFH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,033H,0CFH,0C0H,01CH,030H,070H :m
DB 018H,030H,030H,018H,030H,030H,03CH,078H,070H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,038H,0FFH,0 ,00FH,0 ,0E0H :n
DB 006H,0 ,060H,006H,0 ,060H,01FH,081H,0F0H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0FFH,0 ,007H,001H,0C0H :o
DB 018H,0 ,030H,01CH,0 ,070H,007H,0FFH,080H,0 ,0 ,0
DB 0 ,0 ,0 ,00FH,03FH,0 ,003H,0C0H,0C0H,003H,0 ,060H :p

```



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 003H,0 ,0E0H,003H,0FFH,0C0H,003H,0 ,0 ,00FH,0C0H,0
DB 0 ,0 ,0 ,0 ,0FCH,0F0H,007H,003H,0C0H,00CH,0 ,0C0H ;q
DB 00CH,0 ,0C0H,003H,0FFH,0C0H,0 ,0 ,0C0H,0 ,003H,0F0H
DB 0 ,0 ,0 ,0 ,0 ,00FH,01FH,0 ,003H,0E1H,0C0H ;r
DB 003H,0 ,0C0H,003H,0 ,0 ,00FH,0C0H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,001H,0FEH,0 ,006H,001H,080H ;s
DB 003H,0FEH,0 ,0 ,001H,0C0H,00FH,0FFH,0C0H,0 ,0 ,0
DB 003H,0 ,0 ,003H,0 ,0 ,03FH,0F0H,0 ,003H,0 ,0 ;t
DB 003H,0 ,0 ,003H,001H,080H,001H,0FFH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,03CH,007H,0E0H,00CH,001H,080H ;u
DB 00CH,001H,080H,00CH,001H,080H,007H,0FFH,0F0H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,03FH,003H,0F0H,00EH,001H,0C0H ;v
DB 003H,006H,0 ,001H,0DCH,0 ,0 ,070H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,038H,078H,070H,038H,078H,070H ;w
DB 01CH,0FCH,0E0H,00FH,0CFH,0C0H,007H,087H,080H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,03FH,00FH,0C0H,00FH,00FH,0 ;x
DB 001H,0F8H,0 ,003H,09CH,0 ,03FH,007H,0E0H,0 ,0 ,0
DB 0 ,0 ,0 ,03FH,003H,0F0H,00EH,001H,0C0H,003H,083H,0 ;y
DB 0 ,0EEH,0 ,0 ,018H,0 ,038H,070H,0 ,00FH,0C0H,0
DB 0 ,0 ,0 ,0 ,0 ,00FH,0FFH,0C0H,00CH,007H,080H ;z
DB 0 ,038H,0 ,001H,0C0H,0 ,00FH,0FFH,0C0H,0 ,0 ,0
DB 0 ,03FH,0 ,001H,0C0H,0 ,001H,0E0H,0 ,00FH,0C0H,0 ;{
DB 0 ,0E0H,0 ,001H,080H,0 ,0 ,0FFH,0 ,0 ,0 ,0
DB 0 ,060H,0 ,0 ,060H,0 ,0 ,060H,0 ,0 ,0 ,0 ;|
DB 0 ,060H,0 ,0 ,060H,0 ,0 ,060H,0 ,0 ,0 ,0
DB 01FH,0C0H,0 ,0 ,038H,0 ,0 ,078H,0 ,0 ,03FH,0 ;}
DB 0 ,070H,0 ,0 ,00CH,0 ,01FH,0F0H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;~
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;_
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;del
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0

```

under this line is ENGFONT2 : (FONT 16X8,16X16)

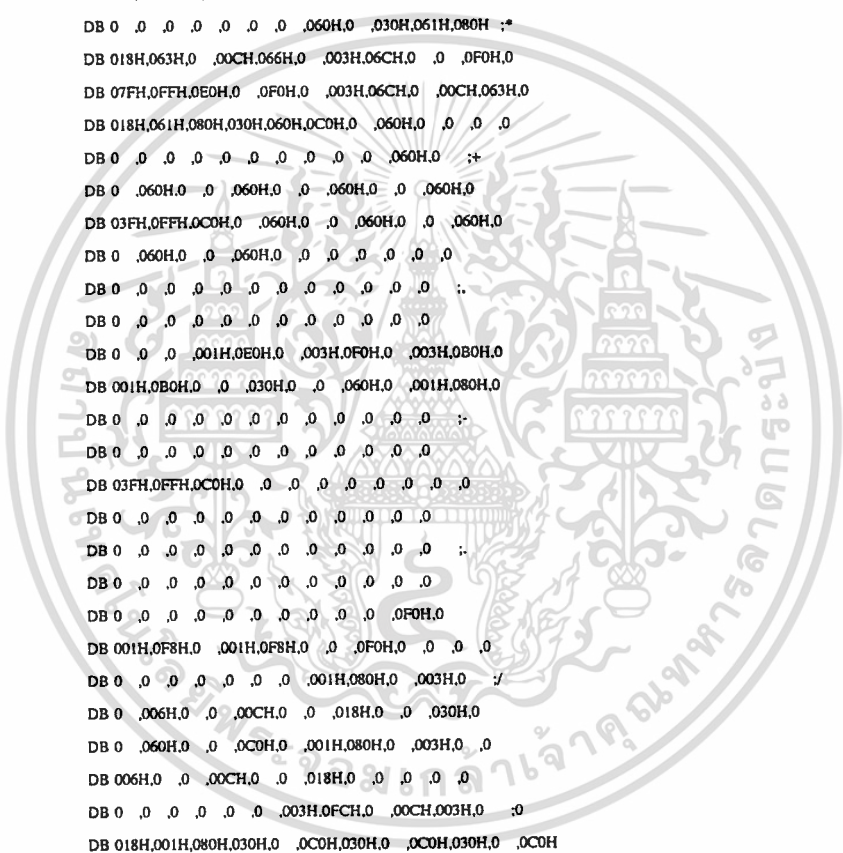
```

DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;space
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,060H,0 ,0 ,0F0H,0 ;!
DB 0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0F0H,0 ,0 ,0F0H,0
DB 0 ,0F0H,0 ,0 ,060H,0 ,0 ,060H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,060H,0 ,0 ,060H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,006H,006H,0 ,00FH,00FH,0 ;"
DB 00FH,00FH,0 ,00FH,00FH,0 ,006H,006H,0 ,006H,006H,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,00CH,0 ,003H,00CH,0 ;#
DB 003H,00CH,0 ,003H,00CH,0 ,03FH,0FFH,0C0H,003H,00CH,0
DB 003H,00CH,0 ,003H,00CH,0 ,003H,00CH,0 ,03FH,0FFH,0C0H
DB 003H,00CH,0 ,003H,00CH,0 ,003H,00CH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,001H,098H,0 ,001H,098H,0 ;$
DB 007H,0FEH,0 ,019H,099H,080H,019H,098H,0 ,019H,098H,0
DB 007H,0FEH,0 ,001H,099H,080H,001H,099H,080H,019H,099H,080H
DB 007H,0FEH,0 ,001H,098H,0 ,001H,098H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,00FH,001H,080H,00DH,083H,0 ;%
DB 00DH,086H,0 ,00DH,08CH,0 ,00FH,018H,0 ,0 ,030H,0
DB 0 ,060H,0 ,0 ,0C0H,0 ,001H,08FH,0 ,003H,019H,080H
DB 006H,019H,080H,00CH,019H,080H,018H,00FH,0 ,0 ,0 ,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB 0 .0 .0 .0 .0 .0 .007H,0E0H,0 .00CH,018H,0 ;&  
DB 018H,018H,0 .018H,018H,0 .018H,018H,0 .006H,030H,0  
DB 003H,060H,060H,007H,080H,0C0H,018H,061H,080H,030H,01BH,0  
DB 030H,00FH,0 .018H,01EH,0 .007H,0E1H,080H,0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .001H,080H,0 .003H,0C0H,0 ;'  
DB 003H,0C0H,0 .003H,0C0H,0 .001H,080H,0 .001H,080H,0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .018H,0 .0 .060H,0 ;(  
DB 0 .0C0H,0 .001H,080H,0 .003H,0 .0 .003H,0 .0  
DB 003H,0 .0 .003H,0 .0 .003H,0 .0 .001H,080H,0  
DB 0 .0C0H,0 .0 .060H,0 .0 .018H,0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .003H,0 .0 .0 .0C0H,0 ;)  
DB 0 .060H,0 .0 .030H,0 .0 .018H,0 .0 .018H,0  
DB 0 .018H,0 .0 .018H,0 .0 .018H,0 .0 .030H,0  
DB 0 .060H,0 .0 .0C0H,0 .003H,0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .060H,0 .030H,061H,080H ;\*  
DB 018H,063H,0 .00CH,066H,0 .003H,06CH,0 .0 .0F0H,0  
DB 07FH,0FFH,0E0H,0 .0F0H,0 .003H,06CH,0 .00CH,063H,0  
DB 018H,061H,080H,030H,060H,0C0H,0 .060H,0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .060H,0 ;+  
DB 0 .060H,0 .0 .060H,0 .0 .060H,0 .0 .060H,0  
DB 03FH,0FFH,0C0H,0 .060H,0 .0 .060H,0 .0 .060H,0  
DB 0 .060H,0 .0 .060H,0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;.  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .001H,0E0H,0 .003H,0F0H,0 .003H,0B0H,0  
DB 001H,0B0H,0 .0 .030H,0 .0 .060H,0 .001H,080H,0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;-  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 03FH,0FFH,0C0H,0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 001H,0F8H,0 .001H,0F8H,0 .0 .0F0H,0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .001H,080H,0 .003H,0 ;/  
DB 0 .006H,0 .0 .00CH,0 .0 .018H,0 .0 .030H,0  
DB 0 .060H,0 .0 .0C0H,0 .001H,080H,0 .003H,0 .0  
DB 006H,0 .0 .00CH,0 .0 .018H,0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .003H,0FCH,0 .00CH,003H,0 ;0  
DB 018H,001H,080H,030H,0 .0C0H,030H,0 .0C0H,030H,0 .0C0H  
DB 030H,0 .0C0H,030H,0 .0C0H,030H,0 .0C0H,030H,0 .0C0H  
DB 018H,001H,080H,00CH,003H,0 .003H,0FCH,0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .030H,0 .0 .0F0H,0 ;1  
DB 003H,0F0H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0  
DB 0 .030H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0  
DB 0 .030H,0 .0 .030H,0 .003H,0FFH,0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .007H,0FCH,0 .00CH,003H,0 ;2  
DB 018H,001H,080H,018H,001H,080H,0 .003H,0 .0 .006H,0  
DB 0 .078H,0 .001H,080H,0 .006H,0 .0 .018H,0 .0  
DB 030H,0 .0C0H,030H,0 .0C0H,03FH,0FFH,0C0H,0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .007H,0FEH,0 .018H,003H,0 ;3  
DB 030H,001H,080H,030H,001H,080H,0 .001H,080H,0 .003H,0  
DB 001H,0FEH,0 .0 .003H,0 .0 .001H,080H,030H,001H,080H  
DB 030H,001H,080H,018H,003H,0 .007H,0FEH,0 .0 .0 .0

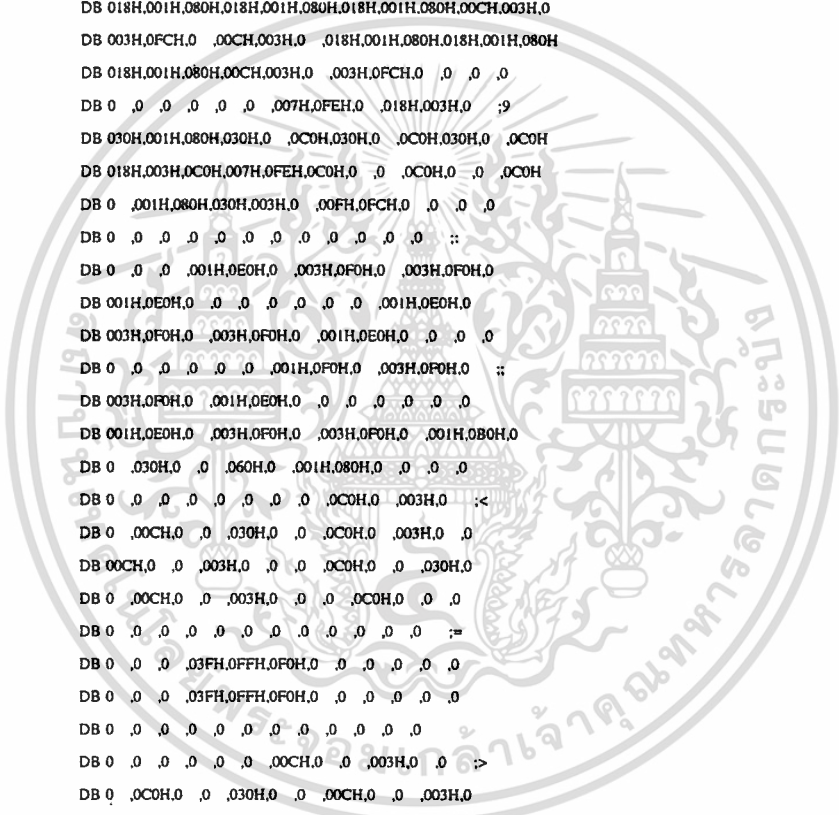


เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 0 .066H,0 ,0 .0C6H,0 .001H,086H,0 .003H,006H,0
DB 006H,006H,0 .00CH,006H,0 .018H,006H,0 .03FH,0FFH,0C0H
DB 0 .006H,0 ,0 .006H,0 ,0 .006H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 .00FH,0FFH,080H,00CH,0 ,0 ,0 ;5
DB 00CH,0 ,0 .00CH,0 ,0 .00CH,0FCH,0 .00FH,003H,0
DB 00CH,001H,080H,0 ,0 .0C0H,0 ,0 .0C0H,0 ,0 .0C0H,0
DB 018H,001H,080H,00CH,003H,0 .003H,0FCH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 .0FFH,0 .003H,003H,0C0H ;6
DB 006H,0 ,0 .00CH,0 ,0 .018H,0 ,0 .018H,0FCH,0
DB 018H,003H,0 .01EH,001H,080H,018H,0 .0C0H,018H,0 .0C0H
DB 00CH,0 .0C0H,006H,001H,080H,003H,0FEH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 .03FH,0FFH,080H,030H,001H,080H ;7
DB 030H,003H,0 ,0 .006H,0 ,0 .00CH,0 ,0 .018H,0
DB 0 .030H,0 ,0 .060H,0 ,0 .060H,0 ,0 .0C0H,0
DB 0 .0C0H,0 ,0 .0C0H,0 ,0 .0C0H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 .003H,0FCH,0 .00CH,003H,0 ;8
DB 018H,001H,080H,018H,001H,080H,018H,001H,080H,00CH,003H,0
DB 003H,0FCH,0 .00CH,003H,0 .018H,001H,080H,018H,001H,080H
DB 018H,001H,080H,00CH,003H,0 .003H,0FCH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 .007H,0FEH,0 .018H,003H,0 ;9
DB 030H,001H,080H,030H,0 .0C0H,030H,0 .0C0H,030H,0 .0C0H
DB 018H,003H,0C0H,007H,0FEH,0C0H,0 ,0 .0C0H,0 ,0 .0C0H
DB 0 .001H,080H,030H,003H,0 .00FH,0FCH,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ::
DB 0 ,0 ,0 .001H,0E0H,0 .003H,0F0H,0 .003H,0F0H,0
DB 001H,0E0H,0 ,0 ,0 ,0 ,0 ,0 .001H,0E0H,0
DB 003H,0F0H,0 .003H,0F0H,0 .001H,0E0H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 .001H,0F0H,0 .003H,0F0H,0 ::
DB 003H,0F0H,0 .001H,0E0H,0 ,0 ,0 ,0 ,0 ,0
DB 001H,0E0H,0 .003H,0F0H,0 .003H,0F0H,0 .001H,0B0H,0
DB 0 .030H,0 ,0 .060H,0 .001H,080H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 .0C0H,0 .003H,0 ;<
DB 0 .00CH,0 ,0 .030H,0 ,0 .0C0H,0 .003H,0 ,0
DB 00CH,0 ,0 .003H,0 ,0 ,0 .0C0H,0 ,0 .030H,0
DB 0 .00CH,0 ,0 .003H,0 ,0 ,0 .0C0H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;=
DB 0 ,0 ,0 .03FH,0FFH,0F0H,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 .03FH,0FFH,0F0H,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 .00CH,0 ,0 .003H,0 ,0 ;>
DB 0 .0C0H,0 ,0 .030H,0 ,0 .00CH,0 ,0 .003H,0
DB 0 ,0 .0C0H,0 .003H,0 ,0 .00CH,0 ,0 .030H,0
DB 0 .0C0H,0 .003H,0 ,0 .00CH,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 .007H,0FEH,0 .00CH,001H,080H ;?
DB 018H,0 .0C0H,018H,0 .0C0H,00CH,0 .0C0H,0 .003H,0
DB 0 .00CH,0 ,0 .030H,0 ,0 .060H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 .0F0H,0 ,0 .0F0H,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;@
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 .060H,0 ,0 .0F0H,0 ;A
DB 001H,098H,0 .001H,098H,0 .003H,00CH,0 .003H,00CH,0
DB 006H,006H,0 .006H,006H,0 .00FH,0FFH,0 .00CH,003H,0
DB 018H,001H,080H,018H,001H,080H,07EH,007H,0E0H,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 .03FH,0FEH,0 .00CH,003H,0 ;B
DB 00CH,001H,080H,00CH,001H,080H,00CH,001H,080H,00CH,003H,0

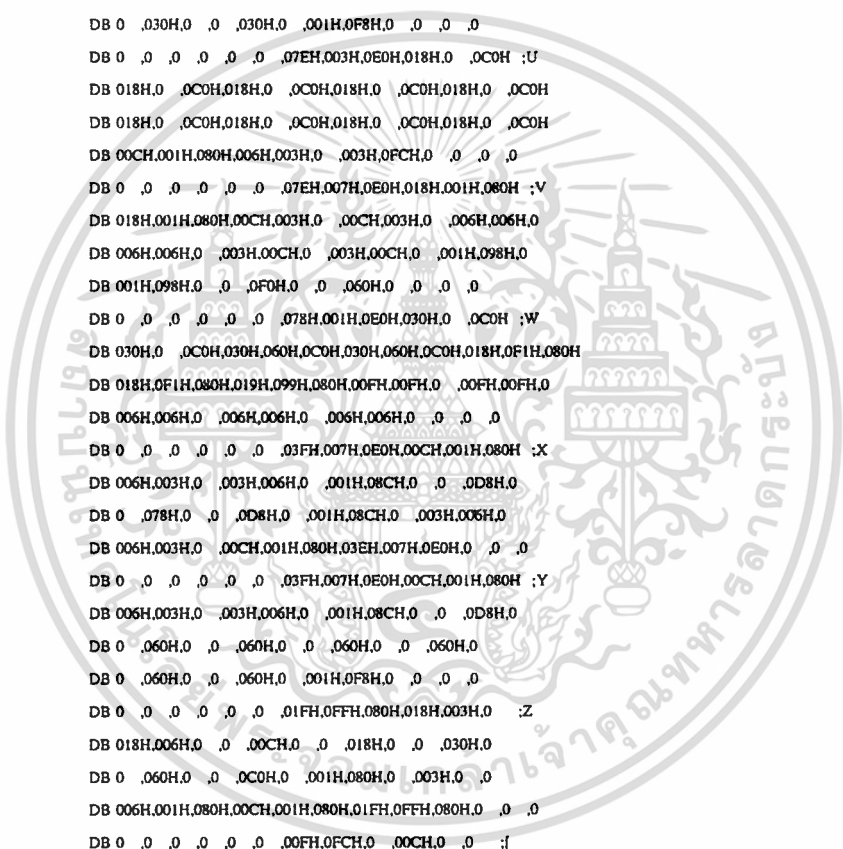
```



DB 00FH,0FEH,0 ,00CH,001H,080H,00CH,0 ,060H,00CH,0 ,060H  
 DB 00CH,0 ,060H,00CH,001H,080H,03FH,0FEH,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,003H,0FCH,0COH,00CH,003H,0COH ;C  
 DB 018H,0 ,0COH,030H,0 ,0COH,030H,0 ,0 ,030H,0 ,0  
 DB 030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0COH  
 DB 018H,001H,080H,00CH,003H,0 ,003H,0FCH,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,003H,0FCH,0COH,00CH,003H,0COH ;D  
 DB 018H,0 ,0COH,030H,0 ,0COH,030H,0 ,0 ,030H,0 ,0  
 DB 030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0COH  
 DB 018H,001H,080H,00CH,003H,0 ,003H,0FCH,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,03FH,0FFH,0COH,00CH,0 ,0COH ;E  
 DB 00CH,0 ,0COH,00CH,0 ,0 ,00CH,00CH,0 ,00CH,00CH,0  
 DB 00FH,0FCH,0 ,00CH,00CH,0 ,00CH,00CH,0 ,00CH,0 ,0  
 DB 00CH,0 ,0COH,00CH,0 ,0COH,03FH,0FFH,0COH,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,0FFH,0COH,00CH,0 ,0COH ;F  
 DB 00CH,0 ,0COH,00CH,0 ,0 ,00CH,00CH,0 ,00CH,00CH,0  
 DB 00FH,0FCH,0 ,00CH,00CH,0 ,00CH,00CH,0 ,00CH,0 ,0  
 DB 00CH,0 ,0 ,00CH,0 ,0 ,03FH,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,0FCH,0COH,00CH,003H,0COH ;G  
 DB 018H,0 ,0COH,030H,0 ,0COH,030H,0 ,0 ,030H,0 ,0  
 DB 030H,01FH,0COH,030H,0 ,0COH,030H,0 ,0COH,030H,0 ,0COH  
 DB 018H,0 ,0COH,00CH,003H,0COH,003H,0FCH,0COH,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,003H,0F0H,00CH,0 ,0COH ;H  
 DB 00CH,0 ,0COH,00CH,0 ,0COH,00CH,0 ,0COH,00CH,0 ,0COH  
 DB 00FH,0FFH,0COH,00CH,0 ,0COH,00CH,0 ,0COH,00CH,0 ,0COH  
 DB 00CH,0 ,0COH,00CH,0 ,0COH,03FH,003H,0F0H,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0FCH,0 ,0 ,030H,0 ,0 ;I  
 DB 0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0  
 DB 0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0  
 DB 0 ,030H,0 ,0 ,030H,0 ,0 ,0FCH,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,01FH,080H,0 ,006H,0 ,0 ;J  
 DB 0 ,006H,0 ,0 ,006H,0 ,0 ,006H,0 ,0 ,006H,0  
 DB 0 ,006H,0 ,0 ,006H,0 ,07EH,006H,0 ,018H,006H,0  
 DB 018H,006H,0 ,00CH,00CH,0 ,003H,0F0H,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,007H,0E0H,00CH,003H,0 ,0 ;K  
 DB 00CH,006H,0 ,00CH,00CH,0 ,00CH,018H,0 ,00CH,030H,0  
 DB 00CH,0F0H,0 ,00DH,098H,0 ,00FH,00CH,0 ,00CH,006H,0  
 DB 00CH,003H,0 ,00CH,001H,080H,03FH,007H,0E0H,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,00FH,0COH,0 ,003H,0 ,0 ,0 ;L  
 DB 003H,0 ,0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0 ,0  
 DB 003H,0 ,0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0 ,0  
 DB 003H,0 ,0COH,003H,0 ,0COH,00FH,0FFH,0COH,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,078H,001H,0E0H,03CH,003H,0COH ;M  
 DB 03CH,003H,0COH,036H,006H,0COH,033H,006H,0COH,033H,00CH,0COH  
 DB 031H,098H,0COH,031H,098H,0COH,030H,0F0H,0COH,030H,060H,0COH  
 DB 030H,060H,0COH,030H,0 ,0COH,07EH,007H,0E0H,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,078H,007H,0E0H,01CH,001H,080H ;N  
 DB 01EH,001H,080H,01BH,001H,080H,019H,081H,080H,018H,0C1H,080H  
 DB 018H,061H,080H,018H,031H,080H,018H,019H,080H,018H,00DH,080H  
 DB 018H,007H,080H,018H,003H,080H,07EH,001H,0E0H,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,0FCH,0 ,00CH,003H,0 ,0 ;O  
 DB 018H,001H,080H,030H,0 ,0COH,030H,0 ,0COH,030H,0 ,0COH  
 DB 030H,0 ,0COH,030H,0 ,0COH,030H,0 ,0COH,030H,0 ,0COH  
 DB 018H,001H,080H,00CH,003H,0 ,003H,0FCH,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,0FCH,0 ,00CH,003H,0 ,0 ;P  
 DB 00CH,001H,080H,00CH,001H,080H,00CH,001H,080H,00CH,001H,080H  
 DB 00CH,003H,0 ,00FH,0FCH,0 ,00CH,0 ,0 ,00CH,0 ,0

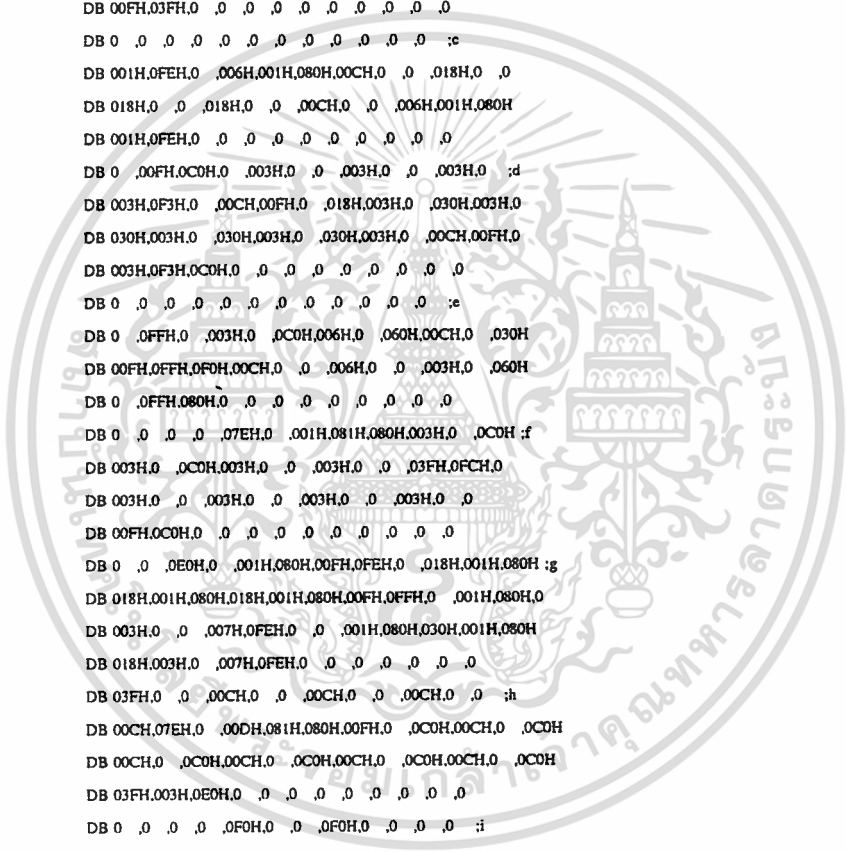
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB 00CH,0 ,0 ,00CH,0 ,0 ,03FH,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,0FCH,0 ,00CH,003H,0 ;Q  
DB 018H,001H,080H,030H,0 ,0COH,030H,0 ,0COH,030H,0 ,0COH  
DB 030H,0 ,0COH,030H,0 ,0COH,030H,060H,0COH,030H,018H,0COH  
DB 018H,007H,080H,00CH,003H,0 ,003H,0FCH,0COH,0 ,0 ,060H  
DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,0FCH,0 ,00CH,003H,0 ;R  
DB 00CH,001H,080H,00CH,001H,080H,00CH,001H,080H,00CH,003H,0  
DB 00FH,0FCH,0 ,00CH,030H,0 ,00CH,018H,0 ,00CH,00CH,0  
DB 00CH,006H,0 ,00CH,003H,0 ,03FH,007H,0E0H,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,007H,0F8H,0 ,018H,006H,0 ;S  
DB 030H,003H,0 ,030H,0 ,0 ,018H,0 ,0 ,00FH,0 ,0  
DB 001H,0E0H,0 ,0 ,03CH,0 ,0 ,006H,0 ,0 ,003H,0  
DB 030H,003H,0 ,018H,006H,0 ,007H,0F8H,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,0FFH,0COH,030H,030H,0COH ;T  
DB 030H,030H,0COH,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0  
DB 0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0 ,0 ,030H,0  
DB 0 ,030H,0 ,0 ,030H,0 ,001H,0F8H,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,07EH,003H,0E0H,018H,0 ,0COH ;U  
DB 018H,0 ,0COH,018H,0 ,0COH,018H,0 ,0COH,018H,0 ,0COH  
DB 018H,0 ,0COH,018H,0 ,0COH,018H,0 ,0COH,018H,0 ,0COH  
DB 00CH,001H,080H,006H,003H,0 ,003H,0FCH,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,07EH,007H,0E0H,018H,001H,080H ;V  
DB 018H,001H,080H,00CH,003H,0 ,00CH,003H,0 ,006H,006H,0  
DB 006H,006H,0 ,003H,00CH,0 ,003H,00CH,0 ,001H,098H,0  
DB 001H,098H,0 ,0 ,0F0H,0 ,0 ,060H,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,078H,001H,0E0H,030H,0 ,0COH ;W  
DB 030H,0 ,0COH,030H,060H,0COH,030H,060H,0COH,018H,0F1H,080H  
DB 018H,0F1H,080H,019H,099H,080H,00FH,00FH,0 ,00FH,00FH,0  
DB 006H,006H,0 ,006H,006H,0 ,006H,006H,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,007H,0E0H,00CH,001H,080H ;X  
DB 006H,003H,0 ,003H,006H,0 ,001H,08CH,0 ,0 ,0D8H,0  
DB 0 ,078H,0 ,0 ,0D8H,0 ,001H,08CH,0 ,003H,006H,0  
DB 006H,003H,0 ,00CH,001H,080H,03EH,007H,0E0H,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,03FH,007H,0E0H,00CH,001H,080H ;Y  
DB 006H,003H,0 ,003H,006H,0 ,001H,08CH,0 ,0 ,0D8H,0  
DB 0 ,060H,0 ,0 ,060H,0 ,0 ,060H,0 ,0 ,060H,0  
DB 0 ,060H,0 ,0 ,060H,0 ,001H,0F8H,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,01FH,0FFH,080H,018H,003H,0 ;Z  
DB 018H,006H,0 ,0 ,00CH,0 ,0 ,018H,0 ,0 ,030H,0  
DB 0 ,060H,0 ,0 ,0COH,0 ,001H,080H,0 ,003H,0 ,0  
DB 006H,001H,080H,00CH,001H,080H,01FH,0FFH,080H,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,00FH,0FCH,0 ,00CH,0 ,0 ;[  
DB 00CH,0 ,0 ,00CH,0 ,0 ,00CH,0 ,0 ,00CH,0 ,0  
DB 00CH,0 ,0 ,00CH,0 ,0 ,00CH,0 ,0 ,00CH,0 ,0  
DB 00CH,0 ,0 ,00CH,0 ,0 ,00FH,0FCH,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 \\  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,003H,0FFH,0 ,0 ,003H,0 ;]  
DB 0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0  
DB 0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0 ,0 ,003H,0  
DB 0 ,003H,0 ,0 ,003H,0 ,003H,0FFH,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;^  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 :  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 :  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 :a  
DB 001H,0FEH,0 .006H,001H,080H,00CH,001H,080H,0 .001H,080H  
DB 001H,0FFH,080H,006H,001H,080H,00CH,001H,080H,00CH,001H,0B0H  
DB 003H,0FFH,060H,0 .0 .0 .0 .0 .0 .0 .0  
DB 00FH,0COH,0 .003H,0 .0 .003H,0 .0 .003H,0 .0 :b  
DB 003H,03FH,0 .003H,0COH,0COH,003H,0 .060H,003H,0 .030H  
DB 003H,0 .030H,003H,0 .030H,003H,0 .060H,003H,0COH,0COH  
DB 00FH,03FH,0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 :c  
DB 001H,0FEH,0 .006H,001H,080H,00CH,0 .0 .018H,0 .0  
DB 018H,0 .0 .018H,0 .0 .00CH,0 .0 .006H,001H,080H  
DB 001H,0FEH,0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .00FH,0COH,0 .003H,0 .0 .003H,0 .0 .003H,0 :d  
DB 003H,0F3H,0 .00CH,00FH,0 .018H,003H,0 .030H,003H,0  
DB 030H,003H,0 .030H,003H,0 .030H,003H,0 .00CH,00FH,0  
DB 003H,0F3H,0COH,0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 :e  
DB 0 .0FFH,0 .003H,0 .0COH,006H,0 .060H,00CH,0 .030H  
DB 00FH,0FFH,0F0H,00CH,0 .0 .006H,0 .0 .003H,0 .060H  
DB 0 .0FFH,080H,0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .07EH,0 .001H,081H,080H,003H,0 .0COH :f  
DB 003H,0 .0COH,003H,0 .0 .003H,0 .0 .03FH,0FCH,0  
DB 003H,0 .0 .003H,0 .0 .003H,0 .0 .003H,0 .0  
DB 00FH,0COH,0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0E0H,0 .001H,080H,00FH,0FEH,0 .018H,001H,080H :g  
DB 018H,001H,080H,018H,001H,080H,00FH,0FFH,0 .001H,080H,0  
DB 003H,0 .0 .007H,0FEH,0 .0 .001H,080H,030H,001H,080H  
DB 018H,003H,0 .007H,0FEH,0 .0 .0 .0 .0 .0 .0  
DB 03FH,0 .0 .00CH,0 .0 .00CH,0 .0 .00CH,0 .0 :h  
DB 00CH,07EH,0 .00CH,081H,080H,00FH,0 .0COH,00CH,0 .0COH  
DB 00CH,0 .0COH,00CH,0 .0COH,00CH,0 .0COH,00CH,0 .0COH  
DB 03FH,003H,0E0H,0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .0 .0 .0 .0F0H,0 .0 .0F0H,0 .0 .0 .0 :i  
DB 003H,0F0H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0  
DB 0 .030H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0  
DB 0 .0FCH,0 .0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 0 .00FH,0 .0 .00FH,0 .0 .0 .0 .0 .03FH,0 :j  
DB 0 .003H,0 .0 .003H,0 .0 .003H,0 .0 .003H,0  
DB 0 .003H,0 .0 .003H,0 .00CH,003H,0 .00CH,003H,0  
DB 006H,006H,0 .001H,0F8H,0 .0 .0 .0 .0 .0 .0  
DB 03FH,0 .0 .00CH,0 .0 .00CH,0 .0 .00CH,0 .0 :k  
DB 00CH,01FH,080H,00CH,00CH,0 .00CH,030H,0 .00CH,0COH,0  
DB 00FH,0COH,0 .00CH,060H,0 .00CH,018H,0 .00CH,006H,0  
DB 01EH,00FH,0COH,0 .0 .0 .0 .0 .0 .0 .0 .0  
DB 003H,0F0H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0 :l  
DB 0 .030H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0  
DB 0 .030H,0 .0 .030H,0 .0 .030H,0 .0 .030H,0  
DB 001H,0FEH,0 .0 .0 .0 .0 .0 .0 .0 .0 .0



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ภายใต้กฎหมายว่าด้วยการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB 067H,09FH,080H,038H,060H,0C0H,030H,060H,060H,030H,060H,060H  
 DB 030H,060H,060H,030H,060H,060H,030H,060H,060H,030H,060H,060H  
 DB 078H,0F0H,0E0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,071H,0FEH,0 ;n  
 DB 01EH,001H,080H,00CH,0 ,0C0H,00CH,0 ,0C0H,00CH,0 ,0C0H  
 DB 00CH,0 ,0C0H,00CH,0 ,0C0H,00CH,0 ,0C0H,00CH,0 ,0C0H  
 DB 03FH,003H,0E0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;o  
 DB 003H,0FCH,0 ,00CH,003H,0 ,018H,001H,080H,030H,0 ,0C0H  
 DB 030H,0 ,0C0H,030H,0 ,0C0H,030H,001H,080H,00CH,003H,0  
 DB 003H,0FCH,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;p  
 DB 03CH,0FCH,0 ,00FH,003H,0 ,00CH,001H,080H,00CH,0 ,0C0H  
 DB 00CH,0 ,0C0H,00CH,0 ,0C0H,00CH,001H,080H,00FH,003H,0  
 DB 00CH,0FCH,0 ,00CH,0 ,0 ,00CH,0 ,0 ,03FH,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;q  
 DB 003H,0F3H,0C0H,00CH,00FH,0 ,018H,00CH,0 ,030H,003H,0  
 DB 030H,003H,0 ,030H,003H,0 ,018H,003H,0 ,00CH,00FH,0  
 DB 003H,0F3H,0 ,0 ,003H,0 ,0 ,003H,0 ,0 ,00FH,0C0H  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;r  
 DB 01EH,03CH,0 ,006H,0C3H,0 ,007H,081H,080H,006H,001H,080H  
 DB 006H,0 ,0 ,006H,0 ,0 ,006H,0 ,0 ,006H,0 ,0  
 DB 01FH,080H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;s  
 DB 003H,0FCH,0 ,00CH,003H,0 ,0C0H,0 ,0 ,007H,080H,0  
 DB 0 ,0FCH,0 ,0 ,003H,0 ,0 ,001H,080H,018H,001H,080H  
 DB 007H,0FEH,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 006H,0 ,0 ,006H,0 ,0 ,006H,0 ,0 ,006H,0 ,0 ;t  
 DB 006H,0 ,0 ,07FH,0F0H,0 ,006H,0 ,0 ,006H,0 ,0  
 DB 006H,0 ,0 ,006H,001H,080H,006H,001H,080H,003H,003H,0  
 DB 0 ,0FCH,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;u  
 DB 078H,00FH,0C0H,018H,003H,0 ,018H,003H,0 ,018H,003H,0  
 DB 018H,003H,0 ,018H,003H,0 ,018H,003H,0 ,00CH,00FH,020H  
 DB 003H,0F0H,0C0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;v  
 DB 03EH,003H,0E0H,018H,001H,080H,00CH,003H,0 ,006H,006H,0  
 DB 006H,006H,0 ,003H,00CH,0 ,001H,098H,0 ,0 ,0F0H,0  
 DB 0 ,060H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;w  
 DB 070H,0F0H,0E0H,060H,060H,060H,030H,0F0H,0C0H,030H,0F0H,0C0H  
 DB 019H,099H,080H,019H,099H,080H,00FH,00FH,0 ,00FH,00FH,0  
 DB 006H,006H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;x  
 DB 01FH,083H,0F0H,006H,0 ,0C0H,001H,083H,0 ,0 ,0CCH,0  
 DB 0 ,078H,0 ,0 ,0CCH,0 ,001H,083H,0 ,006H,0 ,0C0H  
 DB 01FH,083H,0F0H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;y  
 DB 037H,007H,0E0H,018H,001H,080H,00CH,003H,0 ,006H,0 ,0  
 DB 003H,006H,0 ,001H,098H,0 ,0 ,0D8H,0 ,0 ,030H,0  
 DB 0 ,030H,0 ,060H,060H,0 ,030H,0C0H,0 ,01FH,080H,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ;z  
 DB 01FH,0FFH,080H,018H,003H,0 ,0 ,00CH,0 ,0 ,030H,0  
 DB 0 ,060H,0 ,001H,080H,0 ,003H,0 ,0 ,00CH,001H,080H  
 DB 01FH,0FFH,080H,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 ,0  
 DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,07EH,0 ,001H,080H,0 ;{  
 DB 003H,0 ,0 ,003H,0 ,0 ,0 ,0C0H,0 ,001H,080H,0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ; ำ

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0

DB 0 .0C0H.0 .0 .0C0H.0 .0 .0C0H.0 .0 .0C0H.0 .0 .0C0H.0 .0 .0C0H.0 ; ำ

DB 0 .0C0H.0 .0 .0FCH.0 .0 .0FCH.0 .0 .0FCH.0 .0 .0FCH.0 .0 .0FCH.0 ; ำ

DB 00CH.00CH.0 .00CH.00CH.0 .00CH.00CH.0 .00CH.00CH.0 .00CH.00CH.0 .00CH.00CH.0 ; ำ

DB 00CH.00CH.0 .00DH.0CDH.0C0H.00FH.0CFH.0C0H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .003H.0FFH.030H .: ำ (บน)

DB 01CH.0 .0C0H.01FH.0FFH.0 .0 .006H.0 .0 .006H.0

DB 0 .0 .0 .0 .0 .0 .003H.0F0H.0 .01CH.00EH.0 .: ำ (บน)

DB 01FH.0C6H.0 .01FH.0C6H.0 .0 .006H.0 .0 .006H.0

DB 0 .0 .0 .0 .0 .0 .006H.0 .038H.01EH.0 .: ำ (บน)

DB 00EH.076H.0 .006H.0C6H.0 .0 .006H.0 .0 .006H.0

DB 001H.0E3H.0C0H.007H.01EH.030H.006H.0E0H.030H.006H.0E0H.030H .; ำ

DB 0 .0 .030H.0 .0 .070H.0 .001H.0C0H.0 .006H.0

DB 0 .0 .0 .01FH.083H.0C0H.01BH.087H.0C0H.00FH.0FCH.0C0H .: ำ

DB 0 .0 .0C0H.0 .0 .0C0H.0 .0 .0C0H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .0 .007H.080H.0 .001H.080H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .007H.0FEH.0 .01FH.0FFH.080H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .007H.0F9H.080H.01FH.0FFH.080H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .003H.0C0H.007H.0F6H.060H.01FH.0FFH.0E0H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .007H.0F6H.060H.01FH.0FFH.0E0H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .001H.0F0H.0E0H.001H.0FFH.0C0H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ; ำ (จุดของสระอ้อ)

DB 0 .0 .0 .07CH.0 .0 .07CH.0 .0 .0 .0 .0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .001H.080H .;

DB 001H.0FEH.0 .003H.018H.0E0H.001H.0E7H.0C0H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .0 .001H.080H.0 .001H.080H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .0 .0 .0 .0F9H.0E0H.0 .03FH.080H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .022H.020H.0 .0FEH.0E0H.0 .0E7H.080H.0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 ;

DB 0 .003H.0 .0 .01FH.0E0H.0 .003H.0 .0 .0 .0

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .001H.0C0H .;

DB 0 .007H.0 .0 .01FH.0 .0 .00FH.0 .0 .0 .0

DB 0 .006H.0 .0 .006H.0 .0 .006H.0 .0 .006H.0 .0 .006H.0 .: ำ (ของสระอี)

DB 0 .006H.0 .0 .007H.0E0H.0 .007H.0E0H.0 .0 .0

DB 0 .066H.070H.0 .03FH.0C0H.0 .0 .0 .0 .0 .0 .0 .0 .: ำ (ล่าง)

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0

DB 007H.0FEH.060H.007H.0E1H.0E0H.0 .0 .0 .0 .0 .0 .0 .: ำ (ล่าง)

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0

DB 01FH.0E6H.060H.01FH.0F9H.0E0H.0 .0 .0 .0 .0 .0 .0 .: ำ (ล่าง)

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0

DB 01FH.0D9H.0E0H.01FH.0FFH.0E0H.0 .0 .0 .0 .0 .0 .0 .: ำ (ล่าง)

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0

DB 0 .0 .0C0H.0 .0 .0C0H.0 .0 .0 .0 .0 .0 .0 .: ำ (ล่าง)

DB 0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0 .0

:Under This Line GRAPHICS FONT size (16X16)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 0 .030H,0 .038H,0 .03CH,0 .03EH
DB 0 .03FH,0 .020H,0 .020H,0 .02FH
DB 0FFH,0AFH,06AH,0F6H,03FH,0FCH,01FH,0F8H
DB 0 .0 .040H,0 .0C0H,001H,0C0H ;BOAT_(GREEN)
DB 003H,0D0H,007H,0D8H,00FH,0DCH,01FH,0DEH
DB 03FH,0DFH,07FH,0C0H,0FFH,0 .0 .0
DB 0 .0 .015H,008H,0 .0 .0 .0
DB 001H,081H,0 .003H,0 .006H,004H,06CH ;ICECREAM_(RED)
DB 00EH,0F8H,01FH,0F8H,0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .0
DB 0 .001H,001H,0C3H,003H,0C6H,007H,08CH ;ICECREAM_(GREEN)
DB 00FH,008H,01FH,0 .03FH,0FCH,03FH,0FCH
DB 01FH,0F8H,007H,0E0H,003H,0C0H,001H,080H
DB 001H,080H,001H,080H,003H,0C0H,007H,0E0H
DB 0 .0 .001H,080H,008H,010H,001H,080H ;CLOCK_(RED)
DB 021H,084H,001H,080H,001H,080H,047H,082H
DB 047H,082H,0 .0 .0 .020H,004H
DB 0 .0 .008H,010H,001H,080H,0 .0
DB 007H,0E0H,01EH,078H,037H,0ECH,07EH,07EH ;CLOCK_(GREEN)
DB 05EH,07AH,0FEH,07FH,0FEH,07FH,0B9H,0FDH
DB 0B9H,0FDH,0FFH,0FFH,0FFH,0FFH,05FH,0FAH
DB 07FH,0FEH,037H,0ECH,01EH,0F8H,007H,0E0H
DB 0 .0 .0 .0 .0 .0 .0 ;CAR_(RED)
DB 0 .0 .0 .0DCH,001H,0DEH,003H,0DEH
DB 0 .0 .0 .018H,018H,03CH,03CH
DB 03CH,03CH,018H,018H,0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .0 ;CAR_(GREEN)
DB 0 .0FCH,001H,0FEH,003H,0FFH,007H,0FFH
DB 03FH,0FFH,07FH,0FFH,0E7H,0EFH,0C3H,0C3H
DB 0 .0 .0 .0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .01EH,038H ;APPLE_(RED)
DB 03FH,0FCH,07FH,0CEH,0F7H,0CFH,0FFH,0E7H
DB 0FFH,0F3H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH
DB 07FH,0FFH,03FH,0FEH,01FH,0FCH,00FH,038H
DB 0 .038H,0 .060H,0 .0C0H,0 .0C0H ;APPLE_(GREEN)
DB 0 .0 .0 .030H,0 .030H,0 .018H
DB 0 .00CH,0 .0 .0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .0
DB 0 .0 .001H,0E0H,003H,0C0H,004H,081H ;FISH_(RED)
DB 01CH,043H,022H,027H,05AH,01EH,09BH,084H
DB 084H,004H,058H,01EH,02AH,02FH,019H,047H
DB 007H,0E3H,003H,0F0H,0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .003H,0 ;FISH_(GREEN)
DB 007H,080H,01FH,0C0H,027H,0E0H,067H,0F8H
DB 07FH,0F8H,027H,0E0H,01FH,0C0H,007H,080H
DB 0 .0 .0 .0 .0 .0 .0
DB 0 .0 .002H,0 .0 .040H,004H,058H ;TREE_(RED)
DB 036H,0 .030H,098H,011H,050H,007H,020H
DB 003H,040H,001H,080H,001H,080H,001H,080H
DB 001H,080H,003H,0C0H,007H,0E0H,00FH,0F0H
DB 01FH,0F8H,03DH,0FEH,07FH,0BFH,0FFH,0E6H ;TREE_(GREEN)
DB 0E9H,0FEH,0FFH,0E7H,0FFH,0BFH,0FDH,0FAH
DB 07FH,0F8H,03FH,090H,01DH,080H,001H,080H
DB 001H,080H,003H,0C0H,007H,0E0H,00FH,0F0H
DB 03FH,0FCH,03FH,0FCH,03FH,0FCH,03FH,0FCH ;COMPUTER_(RED)
DB 03FH,0FCH,03FH,0FCH,03FH,034H,007H,0E0H

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 0 .0 .008H,038H,014H,0 .0 .0
DB 0 .0 .015H,054H,02AH,0A8H,007H,0E0H
DB 0 .0 .00FH,0F0H,01FH,0F8H,01FH,0F8H ;COMPUTER_(GREEN)
DB 01FH,0F8H,01FH,0F8H,003H,0C8H,0 .0
DB 03FH,0FCH,03FH,0FCH,02BH,0FCH,0 .0
DB 03FH,0FCH,06AH,0AAH,0D5H,057H,0F8H,01FH
DB 0 .0 .0 .0 .0 .03CH,078H ;HEART_(RED)
DB 07EH,0FCH,0FFH,0FEH,0FFH,0FEH,0FFH,0FEH
DB 0FFH,0FEH,07FH,0FCH,03FH,0F8H,01FH,0F0H
DB 00FH,0E0H,007H,0C0H,003H,080H,001H,0
DB 0 .0 .0 .0 .0 .03CH,078H ;HEART_(GREEN)
DB 046H,0C4H,083H,082H,081H,002H,080H,002H
DB 080H,002H,040H,004H,020H,008H,010H,010H
DB 008H,020H,004H,040H,002H,080H,001H,0
DB 022H,044H,012H,048H,088H,011H,047H,0E2H ;SUN_(RED)
DB 02FH,0F4H,01FH,0F8H,0D9H,09BH,019H,098H
DB 01FH,0F8H,0DFH,0FBH,01FH,0F8H,02FH,0F4H
DB 047H,0E2H,088H,011H,012H,048H,022H,044H
DB 022H,044H,012H,048H,088H,011H,040H,002H ;SUN_(GREEN)
DB 020H,004H,0 .0 .0C6H,063H,006H,060H
DB 0 .0 .0C7H,0E3H,003H,0C0H,020H,004H
DB 040H,002H,088H,011H,012H,048H,022H,044H
DB 0 .0 .07FH,0FEH,07FH,0FEH,07FH,0FCH ;DISKET_(RED)
DB 07FH,0FCH,07FH,0FEH,07EH,07EH,07CH,03EH
DB 07CH,03EH,07EH,07EH,07FH,0FEH,07EH,0FEH
DB 07DH,07EH,07DH,07EH,07EH,0FEH,0 .0
DB 0 .0 .0 .0 .03FH,0 .03FH,0 ;DISKET_(GREEN)
DB 03FH,0 .0 .0 .001H,080H,002H,040H
DB 002H,040H,001H,090H,0 .0 .001H,0
DB 003H,080H,003H,080H,001H,0 .0 .0
DB 0 .0 .01FH,0F8H,03FH,0FCH,07FH,0FEH ;FACE_(RED)
DB 073H,0CEH,069H,0A6H,061H,086H,073H,0CEH
DB 07FH,0FEH,07FH,0FEH,07FH,0FEH,07FH,0FEH
DB 07FH,0FEH,03FH,0FCH,01FH,0F8H,0 .0
DB 0 .0 .0 .0 .0 .03FH,0FCH ;FACE_(GREEN)
DB 03FH,0FCH,033H,0CCH,033H,0CCH,03FH,0FCH
DB 03FH,0FCH,03FH,0FCH,030H,00CH,038H,01CH
DB 03CH,03CH,01FH,0F8H,0 .0 .0 .0
DB 0 .0FCH,003F,0F0H,00FH,0E0H,03FH,0C0H ;MOON_(RED)
DB 07FH,080H,0F8H,080H,0F7H,0 .0F7H,0
DB 0F8H,0 .0FFH,0 .0FFH,0 .07FH,0C0H
DB 03FH,0C0H,00FH,0E0H,003H,0F0H,0 .0FCH
DB 0 .0FCH,003H,0F0H,00FH,0E0H,03FH,0C0H ;MOON_(GREEN)
DB 07FH,080H,0FFH,080H,0F8H,080H,0F8H,080H
DB 0FFH,0 .0FFH,0 .0FFH,0 .07CH,0
DB 03CH,0 .00FH,0E0H,003H,0F0H,0 .0FCH
DB 0 .0 .0 .0 .0 .0 .0 .0 ;GLASS_(RED)
DB 0 .0 .0 .0 .0 .0 .0 .0
DB 030H,060H,078H,0F0H,0FDH,0F8H,0FDH,0F8H
DB 078H,0F0H,030H,060H,0 .0 .0 .0
DB 0 .0 .0 .0 .0 .0 .0 .0 ;GLASS_(GREEN)
DB 018H,003H,028H,005H,020H,004H,040H,008H
DB 0CFH,098H,0B7H,068H,078H,0F0H,078H,0F0H
DB 030H,060H,0 .0 .0 .0 .0 .0
DB 0 .07FH,0 .0FEH,001H,0FCH,003H,0F0H ;LIGHTNING_(RED)
DB 007H,0C0H,00FH,080H,01FH,0F8H,03FH,0F0H
DB 001F,0E0H,003H,0C0H,007H,080H,00EH,0

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DB 01CH,0 ,038H,0 ,060H,0 ,0C0H,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 :LIGHTNING_(GREEN)
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,00FH,0FEH ;VAN_(RED)
DB 018H,0CDH,028H,0CDH,04CH,089H,0FFH,0FFH
DB 0FFH,0FFH,0FFH,0FFH,0E7H,0E7H,0C3H,0C3H
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0
DB 0 ,0 ,0 ,0 ,0 ,0 ,0 ,0 :VAN_(GREEN)
DB 0 ,044H,004H,044H,006H,0 ,0 ,0
DB 0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,03CH,03CH
DB 03CH,03CH,018H,018H,0 ,0 ,0 ,0

```

.END



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. ชื่น ภู่วรรณ , วัฒนา เชียงกุล "ไมโครคอมพิวเตอร์" โรงพิมพ์ หก. เอช-เอน การพิมพ์ ,  
กรุงเทพมหานคร , 2532
2. บัณฑิต จามรภูติ , "คู่มือการใช้โปรแกรม orcad/sdt & protel" โรงพิมพ์ หก. เอช-เอน การพิมพ์ ,  
กรุงเทพมหานคร , 2536
3. ETT CO.,LTD , "Microprocessor data book - Technical manual(Z80180)" , ETT CO.,LTD ,  
Bangkok , 1989
4. Science engineering & education CO.,LTD "The work TTL data & Cross - reference guide"  
หก. เอช-เอน การพิมพ์ , กรุงเทพมหานคร , 2532
5. ทินกร ดีก , ธงชัย อุดมกิจโกศล , ชูชัย ธนสารตั้งเจริญ , คณิศ แสงสุริยศิลป์ , ธาณินทร์ ถาวรศาสนวงศ์  
"การใช้งาน Z80" , หก. สำนักพิมพ์ ฟิสิกส์เซ็นเตอร์ , กรุงเทพมหานคร , 2528