

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

บอร์ดแสดงผลโดย LED
LED DISPLAY BOARD



ปริญญาโทสำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน..... 36880
วัน, เดือน, ปี..... 29 ส.ค. 2543

เอกสารนี้สงวนลิขสิทธิ์ไว้เพื่อใช้ในการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดก็ตาม หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อเจ้าหน้าที่หอสมุดกลาง

ปริญญาานิพนธ์ ปีการศึกษา 2542

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง บอร์ดแสดงผลโดย LED

ผู้จัดทำ

1. นายวีระชาติ สุนทรวงค์ เลขประจำตัว 39014491
2. นายวีระชาติ ภูวนาท เลขประจำตัว 39014493
3. นางสาวสรณีย์ นัยวัฒน์ เลขประจำตัว 39014548



ลงชื่อ.....*ท. ส.*.....แทนอาจารย์ที่ปรึกษา
(อาจารย์เทอดศักดิ์ ลีวาททอง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดแสดงผลโดย LED

นายวีระชาติ สุนทรวงศ์

นายวีระชาติ ภูวนาถ

นางสาวสรญา นัยวัฒน์

ถ. ชินภัทร นันทจิวากรชัย อาจารย์ที่ปรึกษา

ปีการศึกษา 2542

บทคัดย่อ

ในปฏิญานีพนธ์ฉบับนี้ เป็นการศึกษาเกี่ยวกับการแสดงผล ตัวอักษร รูปภาพ ผ่านทาง LEDบอร์ด โดยพัฒนารูปแบบของการแสดงผล คือ ให้ไมโครคอนโทรลเลอร์ รับข้อมูลภาพจากคอมพิวเตอร์ส่วนบุคคล ซึ่งข้อมูลภาพจะเป็นเพียงคำสั่งในการสร้างภาพ หรือ ตัวอักษร และรูปแบบการแสดงผลภาพ ไมโครคอนโทรลเลอร์จะต้องทำการประมวลผลคำสั่งที่ได้รับมาสร้างเป็นภาพเพื่อนำไปแสดงผลตามรูปแบบที่ได้รับมา และยังสามารถแสดงตัวอักษรที่เกิดจากการกดคีย์บอร์ดโดยตรง ไม่ต้องผ่านคอมพิวเตอร์ส่วนบุคคล ซึ่ง LEDบอร์ด จะประกอบด้วยวงจรต่างๆ คือ วงจรสร้างสัญญาณนาฬิกา วงจรนับ วงจรถอดรหัส วงจรพักข้อมูล วงจรเก็บข้อมูลภาพ วงจรจับกระแส วงจรควบคุมการเปลี่ยนจากสัญญาณข้อมูลแบบอนุกรมให้เป็นสัญญาณแบบขนาน และส่วนแสดงผลประกอบด้วย LED ขนาด 60 x 91 ต่อกันแบบ เมตริก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED DISPLAY BOARD

Weerachart Soontornwong

Weerachart Puwanart

Soraya Naiyawat

Chinnapat Nantajivagornchai Advisor

1999

ABSTRACT

This thesis is to study and develop a graphic or text display of the LED's Board. The microcontroller is used to receive graphic data from a personal computer which graphic data is instruction to display graphic or text. The microcontroller is used these instruction to create graphic for display. A graphic and text got by pressing keyboard can be display by the microcontroller without a personal computer. The LED's Board circuit consists of many parts. The important part of this thesis consist of a clock generator, a counter circuit, a decoder circuit, latch buffer circuit, a drive circuit, a control change serial data input to parallel data output and a switch LED display board consists of LEDS 60 x 91 dot matrix.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทคัดย่อ	I
ABSTRACT	II
สารบัญ	III
สารบัญรูป	VI
บทที่ 1 บทนำ	1
1.1 ความมุ่งหมายของโครงการ	1
บทที่ 2 ทฤษฎี	3
2.1 คุณสมบัติของหลอดไฟโอดนำแสง	3
2.2 การสแกน	6
2.2.1 หลักการของการสแกน	7
2.2.2 การสแกนทางคอลัมน์	7
2.2.3 การสแกนทางโรวี	7
2.2.4 หลักการสแกนโดยใช้ฮาร์ดแวร์ช่วย	8
2.3 วงจรสร้างสัญญาณนาฬิกา	8
บทที่ 3 ส่วนประกอบของโครงการ	12
3.1 วงจรสร้างสัญญาณนาฬิกา	12
3.2 วงจรนับ	13
3.3 วงจรถอดรหัส	14
3.3.1 ส่วนควบคุมการสแกนในแนวโรวี	14
3.3.2 ส่วนควบคุมการสแกนในแนวคอลัมน์	15
3.4 วงจรรักษาระดับข้อมูลโดยใช้ไอซีบัฟเฟอร์	16
3.5 หน่วยความจำ	18
3.6 วงจรขับกระแส	20
3.7 ส่วนแสดงผล	22
3.8 ส่วนไมโครคอนโทรลเลอร์	22
บทที่ 4 คีย์บอร์ดเบื้องต้น	25
4.1 โครงสร้างคีย์บอร์ด	25
4.2 การเชื่อมต่อกับคีย์บอร์ด	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
4.3 การออกแบบบิตบอร์ดอินเตอร์เฟซ	28
4.4 การทำงานของวงจร	28
บทที่ 5 ส่วนโปรแกรม	30
5.1 โปรแกรมไมโครคอนโทรลเลอร์	30
5.1.1 โปรแกรมหลัก	31
5.1.2 โปรแกรมย่อยสำหรับการเลื่อนภาพขึ้น	32
5.1.3 โปรแกรมย่อยสำหรับการเลื่อนภาพลง	33
5.1.4 โปรแกรมย่อยสำหรับการเลื่อนภาพไปทางขวา	34
5.1.5 โปรแกรมย่อยสำหรับการเลื่อนภาพไปทางซ้าย	35
5.1.6 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงขึ้นขวา	36
5.1.7 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงลงขวา	37
5.1.8 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงขึ้นซ้าย	38
5.1.9 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงลงซ้าย	39
5.1.10 โปรแกรมย่อยสำหรับการเพิ่มขนาดภาพ	40
5.1.11 โปรแกรมย่อยสำหรับการลดขนาดภาพ	41
5.1.12 โปรแกรมย่อยสำหรับการแสดงภาพกระพริบ	42
5.1.13 โปรแกรมย่อยสำหรับการแสดงภาพนิ่ง	43
5.1.14 โปรแกรมย่อยสำหรับการสร้างรูปเส้นตรง	43
5.1.15 โปรแกรมย่อยสำหรับการสร้างรูปสี่เหลี่ยม	45
5.1.16 โปรแกรมย่อยสำหรับการเช็ทค่าในหน่วยความจำ	46
5.1.17 โปรแกรมย่อยสำหรับการรับค่าจากคีย์บอร์ด	47
5.1.18 โปรแกรมย่อยสำหรับการแสดงรูปภาพ	49
5.1.19 โปรแกรมย่อยสำหรับการแสดงตัวอักษร	52
5.2 โปรแกรมเคลฟไล์	53
5.2.1 โปรแกรมหลักเคลฟไล์	56
5.2.2 โปรแกรมการจัดลำดับภาพและตัวอักษร	57
5.2.3 โปรแกรมการเลือกรูปแบบการแสดงผล	58
5.2.4 โปรแกรมการเก็บข้อมูลรูปภาพที่วาด	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
5.2.5 โปรแกรมการบันทึกข้อมูลภาพ	60
5.2.6 โปรแกรมการบันทึกข้อมูลตัวอักษร	61
5.2.7 รูปแบบข้อมูลเฮกซ์ไฟล์ (Hex File)จากโปรแกรมเดลไฟล์	62
บทที่ 6 ผลการทดลอง	63
6.1 ส่วนโปรแกรมไมโครคอนโทรลเลอร์	63
6.2 ส่วนโปรแกรมเดลไฟล์	63
6.3 ส่วนคีย์บอร์ด	63
บทที่ 7 สรุปและวิจารณ์	64
7.1 สรุปและวิจารณ์ผลการทดลอง	64
ภาคผนวก	66
กิตติกรรมประกาศ	139
เอกสารอ้างอิง	140



	หน้าที่
รูปที่ 4.2 แสดงวงจรภายในคีย์บอร์ด (ต่อ)	26
รูปที่ 4.3 โครงสร้างการเชื่อมโยงคีย์บอร์ดกับเมนบอร์ด	27
รูปที่ 4.4 แสดงหัวต่อแบบ DIN (5 ขา) ของคีย์บอร์ดและสัญญาณของแต่ละขา	27
รูปที่ 4.5 แสดงค่าสแกนโค้ดของแป้นตัวอักษร A	28
รูปที่ 4.6 วงจรที่ใช้เชื่อมต่อกับคีย์บอร์ดกับบอร์ด ANT-31	29
บทที่ 5	
รูปที่ 5.1 แผนภาพแสดงการทำงานของโปรแกรมหลัก	31
รูปที่ 5.2 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพขึ้น	32
รูปที่ 5.3 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพลง	33
รูปที่ 5.4 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพไปทางขวา	34
รูปที่ 5.5 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพไปทางซ้าย	35
รูปที่ 5.6 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงขึ้นขวา	36
รูปที่ 5.7 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงลงขวา	37
รูปที่ 5.8 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงขึ้นซ้าย	38
รูปที่ 5.9 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงลงซ้าย	39
รูปที่ 5.10 แผนภาพแสดงการทำงานของโปรแกรมการเพิ่มขนาดภาพ	40
รูปที่ 5.11 แผนภาพแสดงการทำงานของโปรแกรมการลดขนาดภาพ	41
รูปที่ 5.12 แผนภาพแสดงการทำงานของโปรแกรมแสดงภาพกระพริบ	42
รูปที่ 5.13 แผนภาพแสดงการทำงานของโปรแกรมแสดงภาพนิ่ง	43
รูปที่ 5.14 แผนภาพแสดงการทำงานของโปรแกรมสร้างรูปเส้นตรง	44
รูปที่ 5.15 แผนภาพแสดงการทำงานของโปรแกรมสร้างรูปสี่เหลี่ยม	45
รูปที่ 5.16 แผนภาพแสดงการทำงานของโปรแกรมเช็ทค่าในหน่วยความจำ	46
รูปที่ 5.17 แผนภาพแสดงการทำงานของโปรแกรมรับค่าจากคีย์บอร์ด	47
รูปที่ 5.18 แผนภาพแสดงการทำงานของโปรแกรมแสดงรูปภาพ	49
รูปที่ 5.19 แผนภาพแสดงการทำงานของโปรแกรมแสดงตัวอักษร	52
รูปที่ 5.20 รูปแสดงการใช้งานโปรแกรมเดลไฟล์	55
รูปที่ 5.21 แผนภาพแสดงการทำงานของโปรแกรมหลักเดลไฟล์	56
รูปที่ 5.22 แผนภาพแสดงการทำงานของโปรแกรมจัดลำดับภาพและตัวอักษร	57
รูปที่ 5.23 แผนภาพแสดงการทำงานของโปรแกรมเลือกรูปแบบการแสดงผล	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้าที่
บทที่ 1	
รูปที่ 1.1 บล็อกไดอะแกรม (Block Diagram) ของวงจร โคขรวม	2
บทที่ 2	
รูปที่ 2.1 แสดงระดับพลังงานของการรวมตัวของอิเล็กตรอน	3
รูปที่ 2.2 โครงสร้างทั่วไปของ LED	4
รูปที่ 2.3 กราฟแสดงคุณลักษณะของ LED	4
รูปที่ 2.4 ภาพตัดขวางของรอยต่อพี-เอ็น ของ LED	5
รูปที่ 2.5 แสดงแถบพลังงานของLEDเมื่อไม่มีการป้อนศักดาไฟฟ้าให้LED	5
รูปที่ 2.6 แสดงแถบพลังงานของLEDเมื่อมีการจ่ายศักดาไฟฟ้าให้LED	5
รูปที่ 2.7 แผนภาพแสดงการกระจายของประจุพาหะส่วนมากและส่วนน้อย	6
รูปที่ 2.8 โครงสร้างภายในของ LM555	8
รูปที่ 2.9 วงจรสร้างสัญญาณแบบ Astable	9
รูปที่ 2.10 ลักษณะสัญญาณที่จุดต่างๆ	10
รูปที่ 2.11 วงจรการต่อ LM555 แบบ Astable	10
บทที่ 3	
รูปที่ 3.1 การต่อวงจรสร้างสัญญาณนาฬิกาโดยใช้ไอซี LM555	12
รูปที่ 3.2 แสดงลักษณะของขาสัญญาณของไอซีเบอร์ 74HC4040	13
รูปที่ 3.3 แสดงลักษณะของขาสัญญาณของไอซีเบอร์ 74HC4514	14
รูปที่ 3.4 แสดงลักษณะการต่อวงจรถอดรหัสในแนวคอลัมน์	15
รูปที่ 3.5 การต่อวงจรรักษาระดับข้อมูลโดยใช้ไอซีบัฟเฟอร์ 2 ชุด	17
รูปที่ 3.6 ลักษณะของขาสัญญาณแรมเบอร์ 6116	18
รูปที่ 3.7 บล็อกไดอะแกรม (Block Diagram) แสดงการสลับแรม	19
รูปที่ 3.8 วงจรที่ช่วยในการสลับข้อมูลของแรม 2 ตัว	20
รูปที่ 3.9 วงจรในส่วนขับกระแสของบอร์ดแสดงผล	21
รูปที่ 3.10 วงจรส่วนไมโครคอนโทรลเลอร์	23
รูปที่ 3.11 วงจรรวม	24
บทที่ 4	
รูปที่ 4.1 แสดงวงจรภายในคีย์บอร์ด	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้าที่
รูปที่ 5.24 แผนภาพแสดงการทำงานของโปรแกรมเก็บค่าข้อมูลรูปภาพที่วาด	59
รูปที่ 5.25 แผนภาพแสดงการทำงานของโปรแกรมบันทึกข้อมูลภาพ	60
รูปที่ 5.26 แผนภาพแสดงการทำงานของโปรแกรมบันทึกข้อมูลตัวอักษร	61
ภาคผนวก	
รูปวงจรรวม	67



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันซึ่งเป็นโลกของข้อมูลข่าวสาร ที่เป็นหนึ่งเดียวกันทั่วโลก เพราะการติดต่อสื่อสารกันสามารถทำได้ง่ายด้วยเทคโนโลยีของอุปกรณ์สื่อสารที่พัฒนาขึ้นเรื่อยๆ เพื่ออำนวยความสะดวกในการติดต่อสื่อสาร สิ่งเหล่านี้ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์มากขึ้น โดยเฉพาะด้านธุรกิจโฆษณา ที่ต้องการสื่อที่สามารถแสดงข่าวสารข้อมูลได้อย่างครบถ้วน และมีรูปแบบที่ดึงดูดความสนใจพร้อมทั้งสามารถสื่อสารให้ทุกคนเข้าใจในข้อมูล ได้อย่างชัดเจน

แผ่นป้ายแสดงข่าวสาร (Display Board) เป็นอีกสื่อหนึ่งที่ได้รับคามนิยมใช้กันมากในปัจจุบัน ซึ่งได้พัฒนาจากระบบการแสดงผลแบบเป็นเซเวนเซกเมนต์ (7 - Segment) มาเป็นการแสดงผลแบบจุดแสดงผล (Dot Matrix Display) ซึ่งให้ความละเอียดของภาพมากกว่าเดิม และสามารถแสดงผลได้ทั้งตัวอักษรภาษาไทย , ภาษาอังกฤษ , รูปภาพกราฟฟิกต่างๆ และรูปแบบการแสดงผล เช่น ภาพเคลื่อนไหว การเลื่อนภาพขึ้นหรือลง เป็นต้น และสามารถปรับเปลี่ยนข้อมูลที่จะแสดงผลได้ตลอดเวลาตามต้องการ ทั้งยังป้ายแสดงข่าวสารนี้จะมีขนาดเล็ก , ใหญ่ ตามต้องการ สามารถมองเห็นได้ชัดเจน จึงเป็นที่นิยมใช้ในปัจจุบันมากดังจะเห็นได้จากแหล่งชุมชน , ย่านธุรกิจ , ศูนย์การค้า และห้างสรรพสินค้า

1.1 ความมุ่งหมายของโครงการ

ในโครงการนี้จะเป็นการพัฒนารูปแบบของการแสดงผล บนบอร์ดแสดงผลแบบจุด (Dot Matrix Display Board) ซึ่งใช้ LED ชนิด Dot Matrix ขนาด 5 x 7 มาใช้ในการสร้างบอร์ดแสดงผลซึ่งมีขนาดความละเอียด 60 x 91 จุด จากเดิมที่การแสดงผลบนบอร์ดจะใช้ คอมพิวเตอร์ส่วนบุคคล (Personal Computer หรือ PC) ส่งข้อมูลภาพและรูปแบบที่ต้องการแสดงผลไปยัง ไมโครคอนโทรลเลอร์ โดยข้อมูลภาพที่ส่งไปจะมีลักษณะเป็นบิตแมปไฟล์ (Bitmap File) คือเป็นไฟล์ตระกูลบิตแมปชนิดสีเป็นขาว - ดำ มีขนาดความละเอียดของภาพเป็น 91 x 60 จุด ซึ่งจะทำให้ต้องใช้หน่วยความจำ (Memory) ในการเก็บข้อมูลภาพที่ส่งมามากเป็นการสิ้นเปลืองเมื่อภาพมีขนาดเล็กแต่ต้องเก็บทั้งขนาด 91 x 60 จุด ส่วนไมโครคอนโทรลเลอร์จะทำการประมวลรูปแบบการแสดงผล และควบคุมการแสดงผลอีกขั้นตอนหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

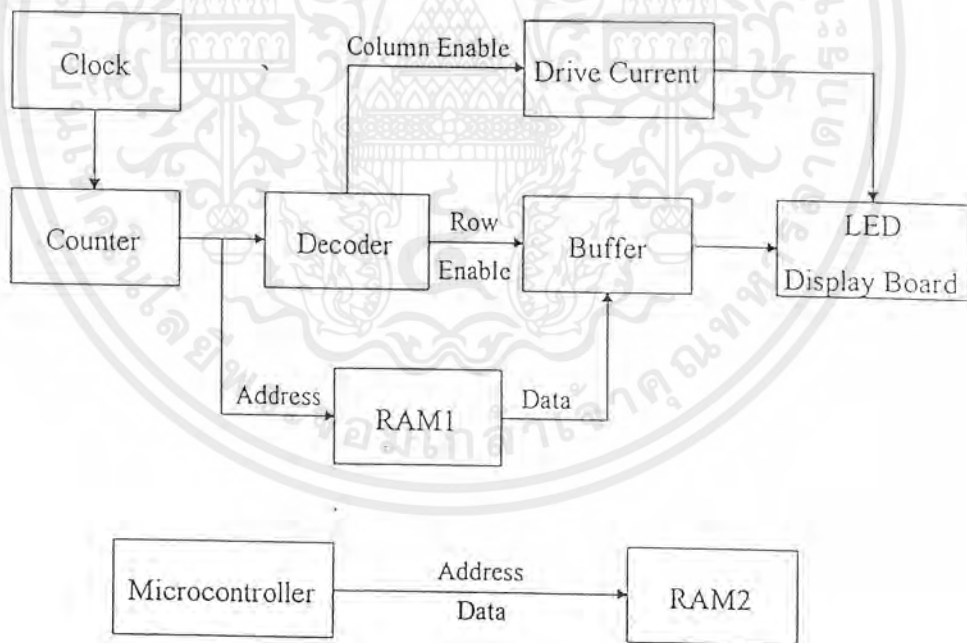
ส่วนในโครงการนี้จะรับข้อมูลมาจาก คอมพิวเตอร์ส่วนบุคคล โดยข้อมูลที่ส่งมาจะเป็นเพียงคำสั่งในการสร้างรูปภาพและรูปแบบการแสดงผล และสามารถส่งข้อมูลจากคีย์บอร์ด มาให้ ไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์จะเป็นตัวประมวลคำสั่งที่ได้รับมา แล้วนำมาสร้างเป็น ข้อความ , รูปภาพ และรูปแบบการแสดงผลตามที่เราต้องการ และควบคุมการแสดงผล จะเห็นได้ว่าจะใช้หน่วยความจำในการเก็บข้อมูลที่มาจากคอมพิวเตอร์ส่วนบุคคล และคีย์บอร์ด น้อยลง

ส่วนประกอบทางฮาร์ดแวร์ที่สำคัญของโครงการนี้ ได้แก่

ส่วนแสดงผล (Display Board)

ส่วนตัวขับ (Driver Board)

ส่วนควบคุมและพักข้อมูล (Control / Latch Data Board)



รูปที่ 1.1 บล็อกไดอะแกรม (Block Diagram) ของวงจรโดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

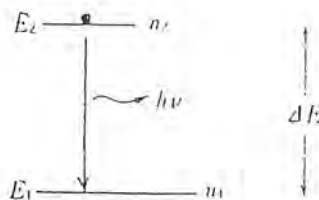
ในโครงการนี้ประกอบด้วย ส่วนควบคุมการแสดงผลและส่วนแสดงผล โดยส่วนควบคุมการแสดงผลจะมี Microcontroller (ไมโครคอนโทรลเลอร์) ตระกูล MCS-51 เป็นส่วนประกอบที่สำคัญ ส่วนแสดงผลก็จะมี LED เป็นส่วนประกอบสำคัญ ดังนั้นจึงต้องทำความเข้าใจเกี่ยวกับคุณสมบัติและการทำงานของ ทั้งสองตัวนี้ให้เข้าใจ และศึกษาการแสดงผลโดย LED นี้

2.1 คุณสมบัติของหลอดไดโอดนำแสง (LED)

LED เป็นไดโอดชนิดหนึ่งซึ่งสามารถเปล่งแสงได้ โดยโครงสร้างของ LED จะมีลักษณะเป็นสารกึ่งตัวนำชนิดพี (P-Type) และชนิดเอ็น (N-Type) ต่อกันเป็นรอยต่อ พี-เอ็น (P-N Junction) LED จะเปล่งแสงออกมาได้ก็ต่อเมื่อมีการจ่ายกระแสไบอัสตรง (Forward Bias) ให้กับมัน กระแสไบอัสตรงนี้จะไปกระตุ้นอิเล็กตรอน (Electron) และ โฮล (Hole) ข้ามรอยต่อ พี-เอ็น เพื่อมารวมตัวกัน ในการรวมตัวกันนี้จะมีการปลดปล่อยพลังงานออกมาในรูปของโฟตอน (Photon) ซึ่งเป็นอนุภาคของแสง ซึ่งต่างจากอุปกรณ์อย่างอื่น ๆ ที่ปลดปล่อยพลังงานออกมาในรูปของความร้อน สำหรับสารกึ่งตัวนำที่นิยมนำมาสร้าง LED จะใช้แกเลียมอาร์เซไนด์ฟอสไฟด์ (Gallium Arsenide Phosphide : GaAsP) ซึ่งสารทั้งสองชนิดนี้จะใช้กระแสไฟฟ้าไม่มากในการไบอัสเพื่อให้เกิดการปลดปล่อยโฟตอนออกมากการให้แสงของ LED โดยการจ่ายกระแสไฟฟ้าเรียกว่าอิเล็กโตรลูมิเนสเซนส์ (Electroluminescence)

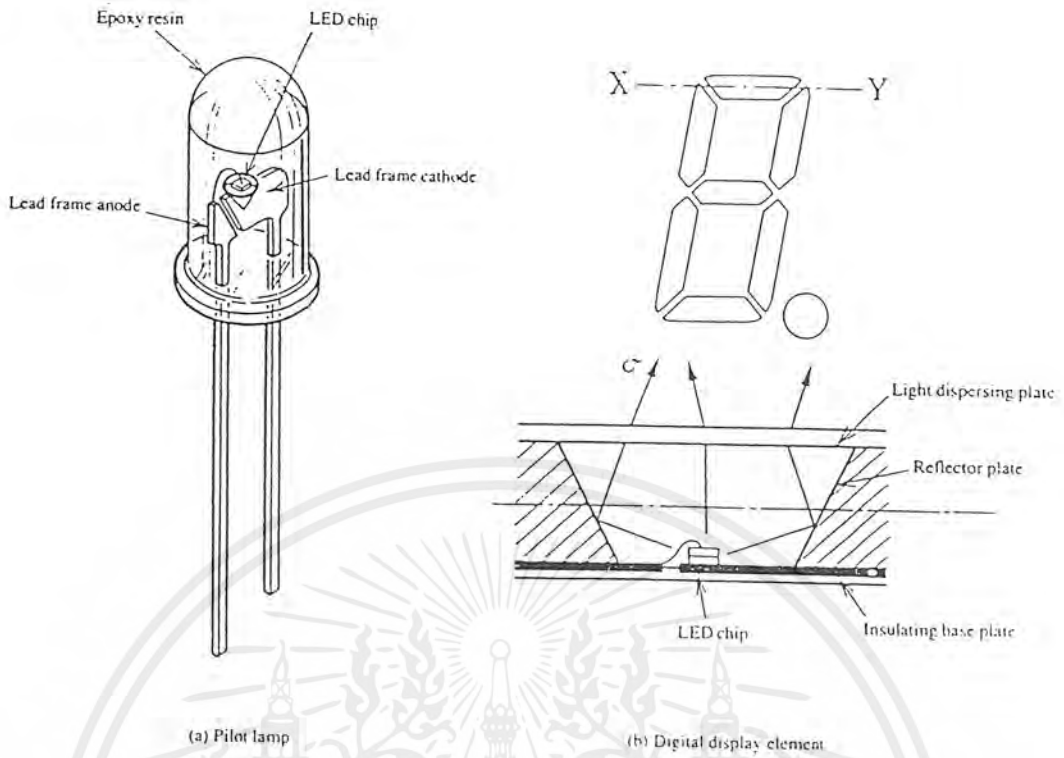
คุณสมบัติเด่นของ LED คือ

1. เป็นแหล่งกำเนิดแสงขนาดเล็กแต่มีความทนทานสูง
2. ใช้แรงดันที่มาไบอัส LED ต่ำ (ประมาณ 2 โวลต์)
3. สามารถนำมาใช้งานในการมอดดูเลชัน (Modulation) ความเร็วสูงได้

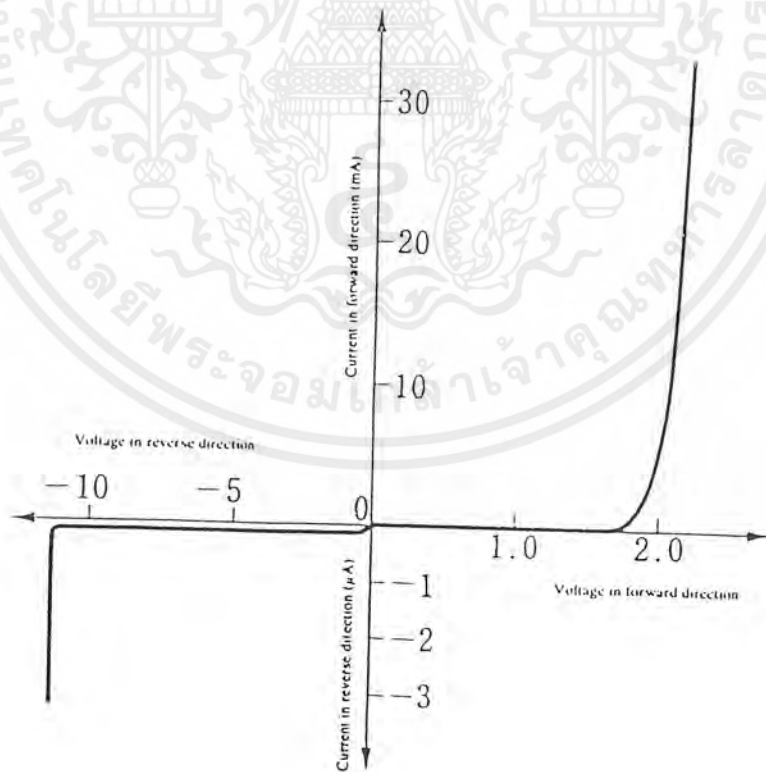


รูปที่ 2.1 แสดงระดับพลังงานของการรวมตัวของอิเล็กตรอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

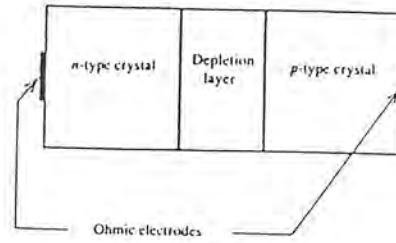


รูปที่ 2.2 โครงสร้างทั่วไปของ LED



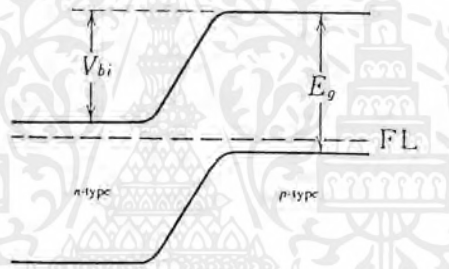
รูปที่ 2.3 กราฟแสดงคุณลักษณะของ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



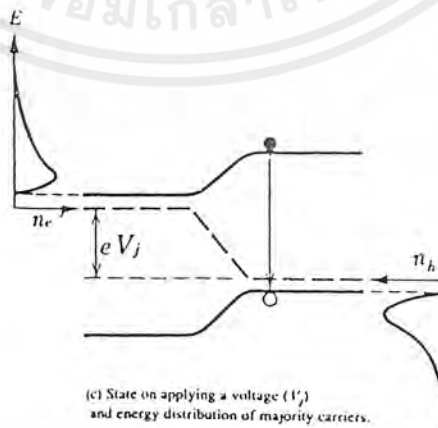
(a) Cross-section of diode structure

รูปที่ 2.4 ภาพตัดขวางของรอยต่อ พี-เอ็น ของ LED



(b) Band structure when no voltage has been applied

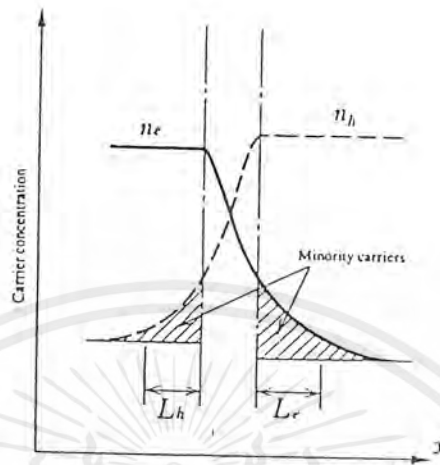
รูปที่ 2.5 แสดงแถบพลังงานของ LED เมื่อไม่มีการป้อนศักดาไฟฟ้าให้ LED



(c) State on applying a voltage (V_j) and energy distribution of majority carriers.

รูปที่ 2.6 แสดงแถบพลังงานของ LED เมื่อมีการจ่ายศักดาไฟฟ้าให้ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แผนภาพแสดงการกระจายของประจุพาหะส่วนมากและส่วนน้อย

หมายเหตุ :

E_g : Forbidden bandwidth

V_{bi} : Internal potential difference at junction

n_c : Electron density

n_h : Positive hole density

L_h : Positive hole diffusion length

L_e : Electron diffusion length

FL : Fermi level

E : Energy

V_j : Voltage applied to junction

2.2 การสแกน (Scanning)

ภาพที่เราเห็นหรือตัวอักษรต่างๆ ในป้ายโฆษณา นั้น จะประกอบไปด้วยจุดเล็กๆ จำนวนหนึ่งที่ทำให้เกิดขึ้นโดยเส้นแนวนอนและเส้นแนวตั้ง โดยจะแบ่งออกเป็นส่วนๆ ส่วนเล็กๆ กัน จุดเหล่านี้ก็คือ LED หนึ่งดวงนั่นเอง การที่เราจะมองเห็นว่าเป็นภาพได้ ก็คือการควบคุมให้ LED สว่างติดตามต้องการ แต่เราไม่สามารถทำให้ LED เหล่านี้สว่างติดตลอดเวลาได้ ดังนั้นต้องใช้เทคนิคในการสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่ยุ่งยากอันหนึ่งคือเราจำเป็นต้องควบคุมการติดดับของ LED ให้สามารถมองเห็นเป็นภาพนิ่งหรือภาพที่เคลื่อนไหวได้ โดยปกติแล้ว LED ที่ใช้กันอยู่จะมีความเข้มของการส่องสว่างในระดับ มิลลิคาเดลลา (mcd) ซึ่งเป็นหน่วยการวัดความเข้มของการส่องสว่าง เช่น LED สีแดงให้ความส่องสว่าง 20 mcd ความส่องสว่างเอาท์พุทเมื่อป้อนกระแสตรง 10 มิลลิแอมป์ (mA) จะได้เพียง 0.7 mcd ในขณะที่ ถ้าป้อนเป็นพัลส์ที่มีกระแสสูงสุด 100 mA คิวดีไซเคิล $d/T = 1/10$ ซึ่งก็ได้กระแสเฉลี่ย 10 mA เช่นกัน จะได้ความส่องสว่างเอาท์พุทเฉลี่ย 2 mcd ซึ่งจะเห็นได้ว่าถ้าใช้กระแสพัลส์ จะได้กำลังแสงเอาท์พุทเพิ่มขึ้นเป็นสามเท่า เมื่อใช้กระแสเท่ากัน และการทำให้มองเห็นเป็นแสงต่อเนื่องได้ จะต้องป้อนกระแสพัลส์ที่มีความถี่มากกว่า 30 Hz ตามที่เราจึงมองเห็นแสงเอาท์พุท LED เป็นแสงต่อเนื่อง แต่ถ้าเราต้องการให้เห็นเป็นภาพต่อเนื่องหรือตัวอักษรวิ่งที่ไม่สามารถสังเกตเห็นการกระพริบได้เลย เราควรป้อนพัลส์ที่มีความถี่ประมาณ 50 - 60 Hz จึงจะไม่เห็นภาพเกิดการพริ้ว และหน้าที่ของการสแกนก็คือการเลือก LED ที่สว่างติดให้ได้ภาพที่ชัดเจนตามต้องการ การสแกนหมายถึง จำนวนเส้นการสแกนต่อหนึ่งภาพ และจำนวนภาพที่ส่งออกไปต่อวินาทีมากมายเท่าไร การกระพริบของภาพก็จะลดลงเท่านั้น

2.2.1 หลักการของการสแกน

ในการทำให้วิ่งมีหลักการอยู่ที่การสแกน ซึ่งการสแกนนี้สามารถทำได้ 2 วิธี คือ สแกนทางคอลัมน์ (Column) การสแกนทางโรว์ (Row)

2.2.2 การสแกนทางคอลัมน์

การสแกนทางคอลัมน์จะทำการส่งข้อมูลออกไปทางโรว์โดยส่งข้อมูลตัวที่ 1 ออกไปแล้วให้ คอลัมน์ที่ 1 แอกทีฟ (Active) จากนั้นก็ทำการส่งข้อมูลตัวที่ 2 ออกไป แล้วให้คอลัมน์ที่ 2 แอกทีฟ ทำเช่นนี้ไปจนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกคอลัมน์ก็จะเป็นการสแกนครบ 1 รอบ ดังนั้นถ้าจำนวนหลักที่จะแสดงผลออกมาเป็นตัวอักษรที่มีจำนวนหลายหลัก วิธีนี้จะไม่เหมาะสมนักที่จะนำมาใช้งาน เพราะว่าเมื่อให้ LED ในคอลัมน์ที่ 1 ติด กว่าที่ LED ที่คอลัมน์สุดท้ายจะติดต้องใช้เวลาานาน

2.2.3 การสแกนทางโรว์

ส่วนการสแกนทางโรว์จะทำการส่งข้อมูลออกไปครบทุกหลักก่อนแล้วให้โรว์ที่ 1 แอกทีฟ จากนั้นก็ทำการส่งข้อมูลชุดถัดไปออกไปจนครบหมดทุกหลัก แล้วให้โรว์ที่ 2 แอกทีฟ ทำเช่นนี้จนกระทั่งข้อมูลถูกส่งออกไปครบหมดทุกโรว์ ก็จะเป็นการสแกนครบ 1 รอบ วิธีนี้มีข้อดีคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถแสดงผลเป็นตัวอักษรพร้อมกันได้หลายหลักและถ้าจัดเวลาให้เหมาะสมแล้ว เวลาทำการสแกนจะไม่เกิดอาการฟริ้ว แต่มีข้อเสียคือ การเขียนโปรแกรมควบคุมให้ตัวอักษรเลื่อนทำได้ยากกว่าแบบแรก

2.2.4 หลักการสแกนโดยใช้ฮาร์ดแวร์ช่วย

โดยทั่วไปในการสแกนไม่ว่าจะเป็นการสแกนทางโร้วหรือการสแกนทางคอลัมน์ เราต้องควบคุมการสแกน และข้อมูลที่ส่งไปเองทุกอย่าง จึงทำให้เกิดปัญหาในกรณีที่เรากำลังต้องการแสดงผลที่มีความละเอียดสูง จะทำให้จำนวนคอลัมน์และโร้วมีค่ามากทั้งคู่ ทำให้การแสดงผลตั้งแต่แถวแรกจนถึงแถวสุดท้ายใช้เวลานาน นอกจากนี้ส่วนควบคุมยังต้องใช้เวลามากขึ้น ในการเคลื่อนย้ายข้อมูลเพื่อทำให้ภาพเคลื่อนไหว ทำให้ช่วงนี้ไม่สามารถควบคุมการแสดงผลได้ดี

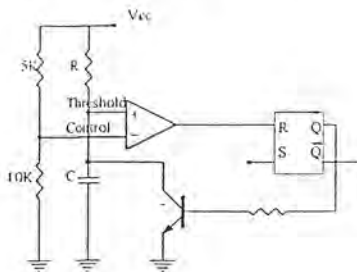
ดังนั้นจึงมีการใช้วงจรทางฮาร์ดแวร์เข้ามาช่วย โดยใช้ฮาร์ดแวร์นี้ทำการเก็บข้อมูลที่แสดงผลและทำการสแกนออกมาโดยอัตโนมัติ โดยไม่ต้องใช้หน่วยควบคุมเข้ามาช่วย ทำให้หน่วยควบคุมนี้มีเวลาที่จะเคลื่อนย้ายข้อมูลเพื่อทำภาพเคลื่อนไหวได้ โดยไม่ส่งผลกระทบต่อภาพที่กำลังแสดงอยู่ นอกจากนี้เนื่องจากการใช้ฮาร์ดแวร์เข้ามาช่วยเป็นการสแกนโดยอัตโนมัติ เราจึงสามารถที่จะแบ่งการสแกนผลที่มีความละเอียดสูงออกเป็นส่วนย่อยๆ เพื่อทำการสแกนด้วยตัวมันเอง ทำให้ลดช่องว่างของเวลาในการแสดงผลระหว่างแถวแรกถึงแถวสุดท้าย ทำให้ภาพที่ได้มีคุณภาพดีขึ้น

ตัวอย่างการสแกนโดยใช้ฮาร์ดแวร์เข้ามาช่วย อย่างเช่น การใช้แรมแยกออกต่างหาก แล้วทำการส่งข้อมูลออกมาโดยใช้ไอซีเคาท์เตอร์ (IC Counter) ช่วยในการแสดง ซึ่งเป็นวิธีที่ใช้ในปริณูณานิพนธ์เล่มนี้ ดังจะกล่าวต่อไป

2.3 วงจรสร้างสัญญาณนาฬิกา

วงจรสร้างสัญญาณนาฬิกาจะใช้ไอซีเบอร์ LM555 ซึ่งเป็น ไอซี 8 ขา ซึ่งจะเอาไว้ต่อกับอุปกรณ์ภายนอกเพื่อที่จะให้ไอซีทำงานในแบบต่างๆ เช่น Monostable หรือ Astable

โครงสร้างภายในของไอซี LM555 เป็นดังรูป

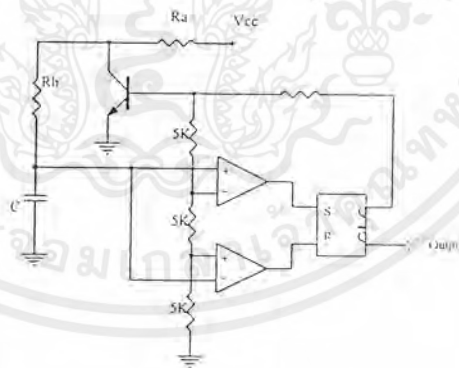


รูปที่ 2.8 โครงสร้างภายในของ LM555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.8 สามารถอธิบายการทำงานของวงจรได้คือ ในขั้นแรกเราสมมุติให้เอาต์พุต Q มีสถานะเป็นสูง (High "+Vcc") ซึ่งจะทำให้ทรานซิสเตอร์อิ่มตัว (Saturate) ดังนั้นตัวเก็บประจุ C ก็จะถูกช้อตลงกราวด์ ค่าของความต่างศักย์ที่ตัวเก็บประจุก็จะเป็น 0 V เนื่องจากตัวเก็บประจุไม่สามารถถูกชาร์ต (Charge) ได้ เราจะนิยามค่าของความต่างศักย์ที่ขานอนอินเวอร์ตติ้ง (Noninverting Input) ของออปแอมป์ ในวงจรดังรูปเป็นค่า แรงดันขีดเริ่ม (Threshold Voltage) และในทำนองเดียวกัน ค่าความต่างศักย์ที่ขา อินเวอร์ตติ้ง (Inverting Input) เราจะนิยามว่า แรงดันควบคุม (Control Voltage) เราก็คจะสามารถกล่าวได้ว่าเมื่อ RS-FF ถูกเซ็ต ทรานซิสเตอร์จะอิ่มตัว ซึ่งก็จะทำให้ค่าแรงดันขีดเริ่มมีค่า 0 V ค่าของแรงดันควบคุมซึ่งมีค่าเท่ากับ 10 V จากการแบ่งแรงดัน ก็จะทำให้เอาต์พุตของออปแอมป์ มีค่าเป็น 0 V ถ้าเราป้อนลอจิกสูง (High) เข้าที่ขา R ซึ่งก็จะเป็นการรีเซ็ต RS-FF ซึ่งเอาต์พุต Q ก็จะมีสถานะเป็นต่ำ (Low) และทรานซิสเตอร์ก็จะคัทออฟ (Cutoff) ตัวเก็บประจุ C ก็จะถูกชาร์ต ผ่านตัวต้านทาน R ค่าของแรงดันขีดเริ่มก็จะมีค่าเพิ่มขึ้น เมื่อค่าของแรงดันขีดเริ่มเพิ่มขึ้นจนมากกว่าค่าของแรงดันควบคุม (+10 V) นิดหน่อย เอาต์พุตของออปแอมป์ ก็จะมีสถานะเป็นสูง (+15 V) ซึ่งก็จะทำให้ RS-FF ถูกเซ็ต เอาต์พุต Q ก็จะเป็นสูง ทรานซิสเตอร์ก็จะอิ่มตัว ตัวเก็บประจุ C ก็จะถูกดิสชาร์ต (Discharge) ผ่านทรานซิสเตอร์นั่นเอง ในการใช้งานทั่วไปขาสัญญาณควบคุมนี้มักจะไม่ได้ใช้

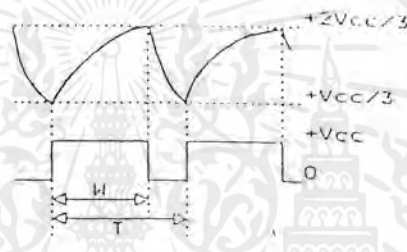
Astable Operation



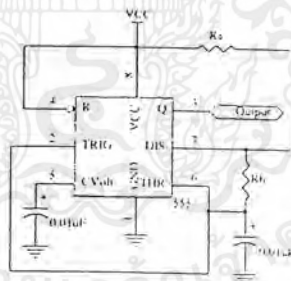
รูปที่ 2.9 วงจรสร้างสัญญาณแบบ Astable

จากรูปที่ 2.9 เป็นการแสดงถึงการต่อไอซี LM555 ทำงานในแบบ Astable หรือ free-running เอาต์พุตที่ได้ของวงจรจะเป็นสัญญาณคลื่นสี่เหลี่ยมที่ต่อเนื่องกันออกมา เราจะสามารถอธิบายการทำงานได้คือ เมื่อเอาต์พุต Q มีสถานะเป็นต่ำ ทรานซิสเตอร์ก็จะคัทออฟ ตัวเก็บประจุ C จะถูกชาร์ตผ่านตัวต้านทานรวม $R_A + R_B$ ดังนั้นค่าเวลาคงตัว (Time constant) ก็จะเป็น $(R_A + R_B)C$ เมื่อตัวเก็บประจุถูกชาร์ต ค่าแรงดันขีดเริ่มก็จะมีค่าเพิ่มขึ้น เมื่อค่าแรงดันขีดเริ่มมีค่าเพิ่ม เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นจนมากกว่าค่า $+2V_{cc}/3$ ออปแอมป์ตัวบนก็จะมีเอาต์พุตเป็นสูง ทรานซิสเตอร์ก็จะอิมิตัว ซึ่งก็เหมือนกับการช้อตขา 7 ของไอซีลจกราวด์น ขณะนี้ตัวเก็บประจุ C ก็จะถูกดิสชาร์จ โดยผ่านทาง R_B ดังนั้นค่า เวลาคงตัวของการดิสชาร์จก็คือ $R_B C$ เมื่อค่าของ แรงดันตกคร่อมตัวเก็บประจุ C มีค่าลดลงจนน้อยกว่า $+V_{cc}/3$ ออปแอมป์ตัวล่างก็จะมีเอาต์พุตเป็นสูงซึ่งจะ รีเซ็ต RS-FF จากรูปที่ 2.10 จะแสดงถึงลักษณะสัญญาณที่จุดต่างๆจะเห็น ได้ว่าการชาร์ตและดิสชาร์จ ของตัวเก็บประจุจะเป็นแบบเอ็กโปเนนเชียล และสัญญาณเอาต์พุตที่ได้จะเป็นคลื่นสี่เหลี่ยม แต่เนื่อง จากค่าเวลาคงตัวของการชาร์ตนั้นมีค่ามากกว่าค่าเวลาคงตัวของการดิสชาร์จ สัญญาณเอาต์พุตที่ได้จึงไม่สมมาตร ซึ่งค่าคาบเวลาของสัญญาณในขณะที่มีสถานะสูง จะยาวนานกว่าในขณะที่มีสถานะต่ำดังรูป



รูปที่ 2.10 ลักษณะสัญญาณที่จุดต่างๆ



รูปที่ 2.11 วงจรการต่อ LM555 แบบ Astable

ในการที่จะกำหนดความไม่สมมาตรของสัญญาณเอาต์พุต เราจะนิยามคำว่า duty cycle ซึ่ง จะกำหนดโดย

$$D = (W / T) * 100\%$$

ยกตัวอย่างเช่นถ้า $W = 2 \text{ ms}$ และ $T = 2.5 \text{ ms}$ ดังนั้นค่า duty cycle คือ

$$D = (2 \text{ ms} / 2.5 \text{ ms}) * 100\% = 80\%$$

ค่า duty cycle ของวงจรมีค่าอยู่ระหว่าง 50-100% ทั้งนี้ขึ้นอยู่กับค่าของ R_A และ R_B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการซาร์ทและคิซาร์ท เราสามารถคำนวณหาค่าของความถี่ของสัญญาณเอาต์พุต คือ

$$f = 1.44 / (R_A + 2R_B) * C$$

และค่าของ duty cycle จะคำนวณได้จาก

$$D = (R_A + R_B) / (R_A + 2R_B) * 100\%$$

จากสูตรจะเห็นได้ว่าถ้าค่า R_A มีค่าน้อยกว่า R_B มาก ๆ แล้วค่า duty cycle จะประมาณ 50% และจากรูปที่ 2.11 จะเป็นรูปวงจรถ่ายที่นิยมเขียนโดยทั่วไปของวงจรถ่าย Astable 555 โดยเราต้องต่อขา 4 ของไอซีเข้ากับ +Vcc และเช่นกัน ขอสัญญาณควบคุม ก็ควรต่อตัวเก็บประจุค่าน้อย (โดยมากใช้ 0.01 F) เอาไว้ดังรูป



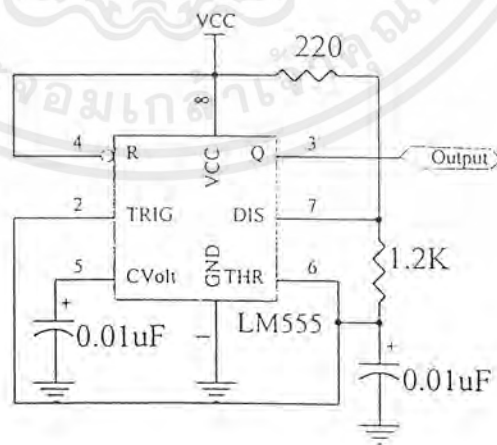
บทที่ 3

ส่วนประกอบของโครงการ

ในบทนี้เราจะมาพิจารณาส่วนประกอบของวงจรทั้งหมด โดยจะทำการแยกพิจารณาทีละส่วนเพื่อให้เข้าใจในการทำ ความเข้าใจ และการปรับปรุงเปลี่ยนแปลง ซึ่งในโครงการนี้ได้เลือกใช้ไอซีตระกูลซี-มอส (C-MOS) เป็นส่วนใหญ่ เนื่องจากในโครงการนี้มีความจำเป็นต้องใช้ไอซีเป็นจำนวนมากและไอซีในตระกูลนี้ใช้กระแสไฟฟ้าในการทำงานน้อยมาก นอกจากนี้แรงดันที่ได้จากไอซีตระกูลนี้ก็มีค่าค่อนข้างคงที่ไม่เปลี่ยนแปลงมาก ซึ่งการรักษาระดับแรงดันนี้มีความจำเป็นในการควบคุมวงจรในส่วนขับเคลื่อน และในปัจจุบัน ไอซีในตระกูลซี-มอสก็มีราคาไม่แตกต่างจากไอซีตระกูลอื่นมากนัก และยังมีให้เลือกใช้มากมายหลายแบบอีกด้วย โดยเราจะพิจารณาแยกเป็นส่วนๆ ดังนี้

3.1 วงจรสร้างสัญญาณพิก้า

วงจรสร้างสัญญาณพิก้า จะใช้ไอซีเบอร์ LM555 เพื่อความสะดวกในการทำงานและออกแบบได้ง่าย รวมทั้งสามารถปรับแต่งความถี่ที่ต้องการได้ตามต้องการโดยง่าย โดยปรับค่าความต้านทานในวงจรเท่านั้น ดังรูปที่ 3.1



รูปที่ 3.1 การต่อวงจรสร้างสัญญาณพิก้าโดยใช้ไอซีเบอร์ LM555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการแสดงภาพนิ่งหรือภาพเคลื่อนไหวจะต้องการใช้ความถี่ในการแสดงภาพในช่วง 50-60 ครั้งต่อวินาที (50-60 Hz) ตามที่ได้กล่าวไว้ในทฤษฎี เพื่อที่จะให้ภาพที่ปรากฏในบอร์ดแสดงผลไม่เกิดการสั่นพริ้ว หรือเกิดการกระพริบได้

จากรูปที่ 3.1 วงจรแผงไฟจะเห็นว่าขนาดของบอร์ดแสดงผลมีความละเอียดขนาด 91 x 60 จุด ถ้าทำการสแกนภาพตามแนวคอลัมน์และในแต่ละแถวใช้บัพเฟอร์ 12 ตัว (12 x 8 = 96 บิต) และใช้วงจรถอดรหัสขนาด 64 x 16 เพราะฉะนั้นจะสามารถคำนวณ สัญญาณนาฬิกาเพื่อให้เกิดความถี่ตามต้องการได้เป็น

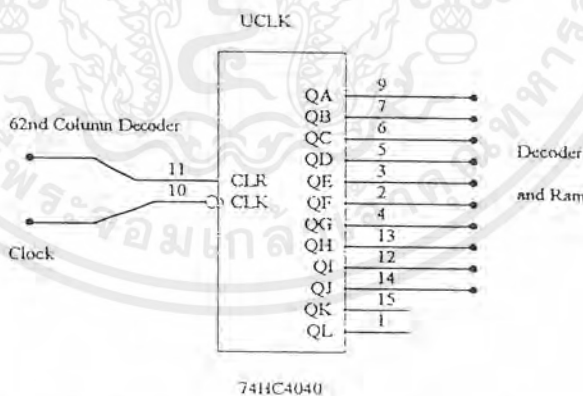
$$\text{Min} = 64 \times 16 \times 50 = 51200 \text{ Hz}$$

$$\text{Max} = 64 \times 16 \times 60 = 61440 \text{ Hz}$$

ความถี่ที่สร้างขึ้นจะต้องมีค่าอยู่ในช่วง Min และ Max จึงจะทำให้มนุษย์สามารถสังเกตเห็นภาพนิ่งได้ และสัญญาณความถี่ที่ได้นี้ก็จะเป็นอินพุตให้กับวงจรนับต่อไป

3.2 วงจรนับ

วงจรมานี้ใช้ไอซีเบอร์ 74HC4040 ซึ่งเป็นไอซีเคาท์เตอร์ขนาด 12 บิต และมีลักษณะของขาสัญญาณดังรูปที่ 3.2



รูปที่ 3.2 แสดงลักษณะของขาสัญญาณของไอซีเบอร์ 74HC4040

ซึ่งไอซีนี้จะใช้นับสัญญาณนาฬิกาจากวงจรสร้างสัญญาณนาฬิกาซึ่งป้อนเข้าที่ขา CLK ของไอซี โดยจะทำการนับที่ขอบขาลงของสัญญาณนาฬิกา ให้ออกมาเป็นไบนารีขนาด 12 บิต ที่ขา Q_A - Q_B ตามลำดับ โดยเมื่อสัญญาณได้ทำการนับจนถึงค่าสุดท้าย (111111111111) ก็จะถูกกลับมาที่ค่าเริ่มต้นอีกครั้ง หรือใช้สัญญาณป้อนเข้าที่ขา CLR เพื่อให้การนับกลับมาที่ค่าเริ่มต้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไม่ต้องรอให้การนับถึงค่าสุดท้าย ซึ่งในโครงการนี้ใช้ขาสัญญาณจากวงจรถอครหัสที่ควบคุมในแนวคอลัมน์เส้นที่ 62 มาใช้ในการเคลียร์การนับเพื่อที่จะให้วงจรสแกนไม่เสียเวลาในการสแกนคอลัมน์ที่ 63 และ 64 ซึ่งเป็นคอลัมน์ที่ไม่ได้ใช้งาน

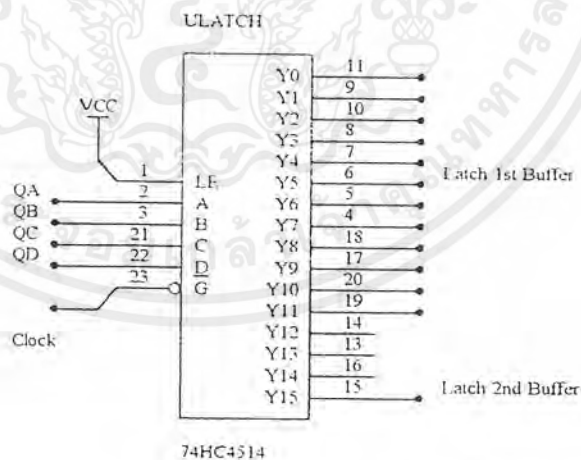
ค่าที่ได้จากวงจรมันนี้จะนำไปใช้ในการกำหนดแอดเดรสให้กับแรมตัวที่จะส่งข้อมูลไปควบคุมการติดดับของ LED ในขณะที่เดียวกันค่าที่ได้จากวงจรมันนี้ ก็จะส่งค่าไปควบคุมวงจรรักษา ระดับข้อมูล และวงจรถอครหัส ให้ทำการเก็บข้อมูลจากแรมตัวที่ส่งข้อมูลออกมาและทำการสแกนทางแนวคอลัมน์อย่างอัตโนมัติ ให้สัมพันธ์กับเวลาที่ข้อมูลถูกส่งออกมาจากแรม

3.3 วงจรถอครหัส

วงจรถอครหัสนี้แบ่งออกเป็น 2 ส่วน คือ ส่วนที่ใช้สำหรับควบคุมการสแกนในแนวคอลัมน์และส่วนที่ใช้ควบคุมการสแกนในแนวโรว์

3.3.1 ส่วนควบคุมการสแกนในแนวโรว์

ในส่วนนี้สัญญาณจาก 4 บิตต่างของวงจรมัน (ขา $Q_A - Q_D$) มาป้อนให้กับไอซีถอครหัสเบอร์ 74HC4514 ซึ่งเป็นไอซีถอครหัสเข้า 4 ออก 16 และทำงานที่สถานะสูง (IC 4-16 Decoder Active High) ที่ขา A-D ตามลำดับ และมีลักษณะของขาสัญญาณดังรูปที่ 3.3



รูปที่ 3.3 แสดงลักษณะของขาสัญญาณของไอซีเบอร์ 74HC4514

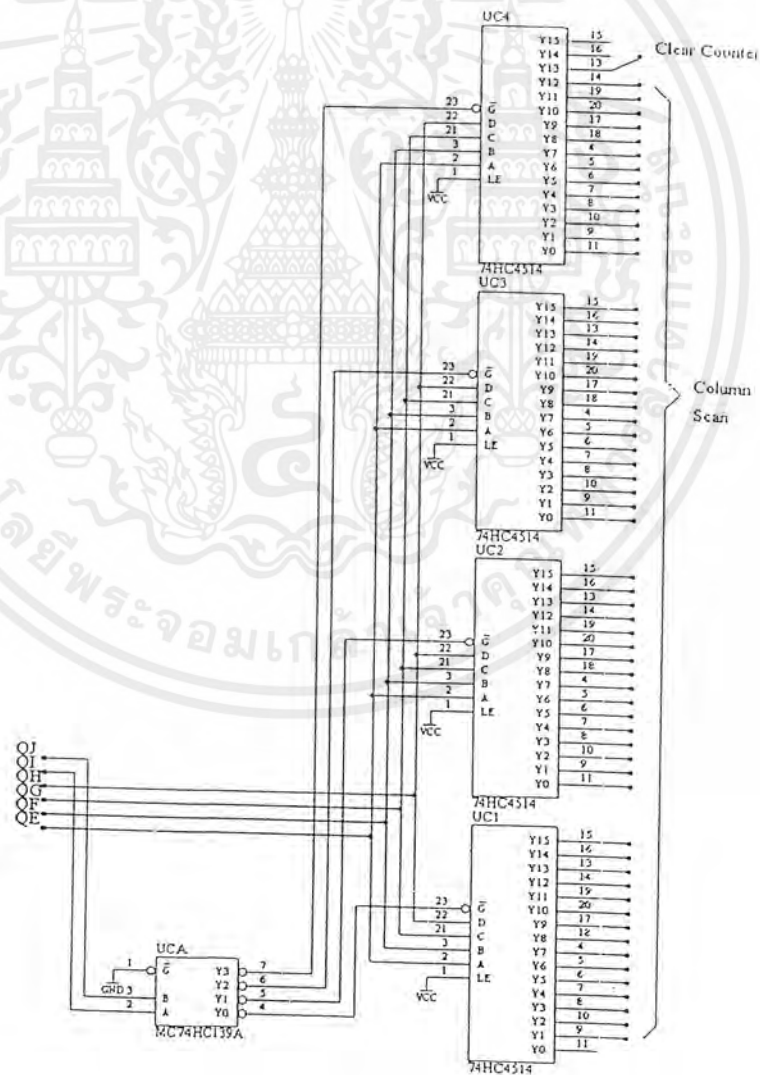
ในวงจรมันนี้ ต่อขาที่ 1 ลงไฟเลี้ยง เพื่อให้สัญญาณออกเปลี่ยนไปตามค่าสัญญาณอินพุตตลอดเวลา และต่อขาที่ 23 กับสัญญาณนาฬิกา เพื่อให้ไอซีทำงานในช่วงครึ่งของการแลทซ์ข้อมูล เพื่อป้องกันการหน่วงเวลาที่ไม่ว่ากันของข้อมูลที่ออกมาจากแรม และสัญญาณที่ใช้ในการแลทซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไอซีบัพ-เฟอร์ โดยขาของสัญญาณเอาต์พุตที่ได้นี้ขา Y0 - Y11 จะนำไปใช้ในการแลตซ์ข้อมูลของไอซีบัพ-เฟอร์ชุดที่หนึ่ง และจะต่อขาสัญญาณ Y15 ในการแลตซ์ข้อมูลของบัพ-เฟอร์ชุดที่สอง คั้งจะได้กล่าวในส่วนของวงจรรักษาระดับข้อมูล

3.3.2 ส่วนควบคุมการสแกนในแนวคอลัมน์

ในการสแกนในแนวนี้ จะใช้สัญญาณจากวงจรนับใน 4 บิตถัดไป ($Q_E - Q_H$) มาป้อนเป็นอินพุตให้กับไอซีเบอร์ 74HC4514 จำนวน 4 ตัว โดยต่อร่วมกับไอซีเบอร์ 74HC139A ที่ได้รับอินพุตจากวงจรถอดรหัส 2 บิตถัดไป (ขา $Q_i - Q_j$) โดยไอซีนี้เป็นไอซีถอดรหัสเข้า 2 ออก 4 จำนวน 2 ชุดทำงานที่สถานะต่ำ (IC Dual 2-4 Decoder active Low) แต่เราใช้เพียงชุดเดียวเท่านั้น และมีลักษณะการต่อร่วมกันดังรูปที่ 3.4



รูปที่ 3.4 แสดงลักษณะการต่อของวงจรถอดรหัสในแนวคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการต่อวงจรดังรูปที่ 3.4 ทำให้ได้วงจรถอดรหัสที่ทำกรขยายเป็นวงจรถอดรหัสเข้า 6 ออก 64 วงจรส่วนที่ได้นี้จะเป็นวงจรสแกนในแนวคอลัมน์ที่สแกนคอลลัมน์เลื่อนไปเรื่อยๆ โดยจะเลื่อนทุกๆ ครั้งที่การสแกนในแนวโร้วทำการสแกนครบ 16 ครั้ง

เนื่องจากสัญญาณที่ได้จากวงจรนี้ จะออกมาจากไอซีโดยตรง ทำให้ไม่สามารถที่จะจ่ายกระแสจำนวนมากได้ เราจึงจำเป็นต้องนำสัญญาณเอาต์พุตที่ได้นี้ไปต่อร่วมกับวงจรขับกระแส โดยการต่อจะใช้คอลลัมน์ที่ 2 - 61 ในการต่อกับวงจรขับกระแส เนื่องจากในขณะที่ทำการสแกนในคอลลัมน์ที่ 1 ข้อมูลจากวงจรรักษาระดับข้อมูลจะยังไม่ป้อนเข้ามา และส่วนของวงจรขับกระแสจะได้อธิบายในส่วนต่อไป

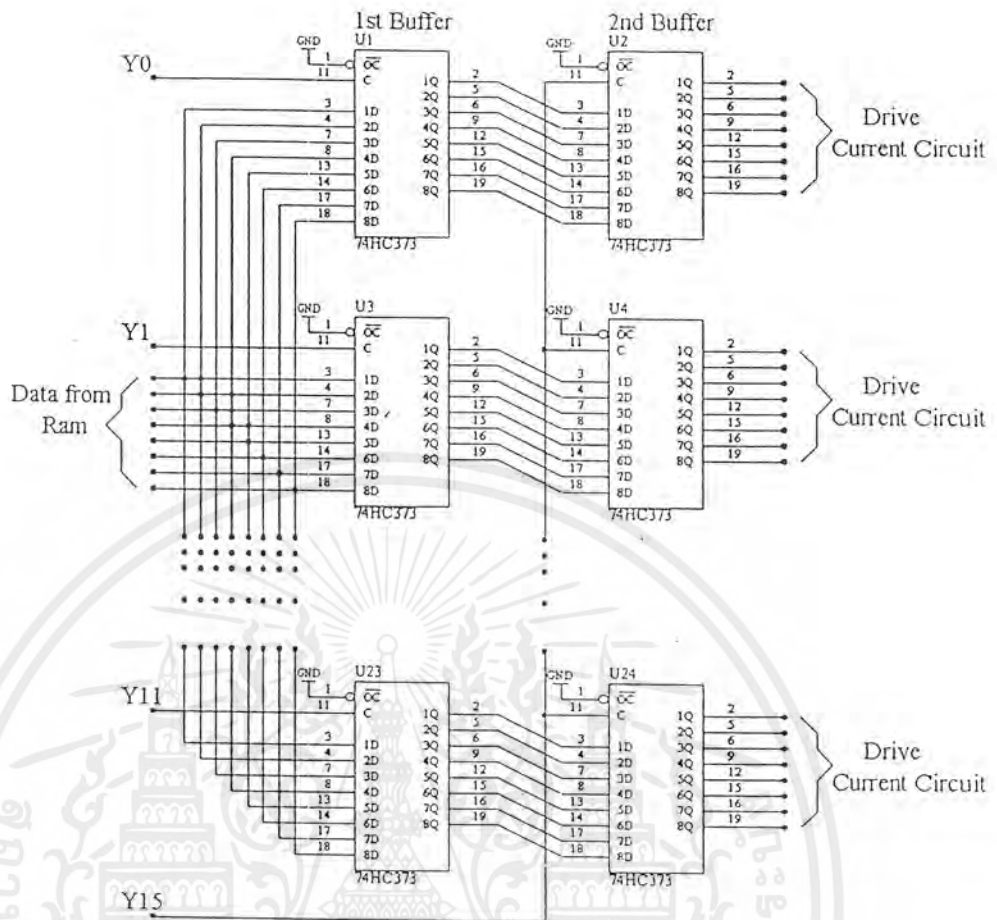
ในวงจรที่ได้นี้เราสามารถที่ใช้นำมาขับกระแสของ LED ได้ถึง 64 คอลลัมน์ แต่ในโครงการนี้ใช้เพียงแค่ 61 คอลลัมน์เท่านั้น (คอลลัมน์ที่ 1 ไม่ได้ใช้งาน) ดังนั้นเราจึงนำสัญญาณที่ได้จากคอลลัมน์ที่ 62 ไปใช้ในการเคลียร์วงจรรัน เพื่อให้วงจรรันเริ่มนับใหม่และวงจรสแกนก็จะเริ่มทำการสแกนในแนวคอลลัมน์ที่ 1 และ โร้วที่ 1 ใหม่ ทำให้วงจรมองไม่ต้องเสียเวลาในการสแกนคอลลัมน์ที่ 63 และ 64 โดยเปล่าประโยชน์

3.4 วงจรรักษาระดับข้อมูลโดยใช้ไอซีบัฟเฟอร์

ในการแสดงภาพแต่ละภาพ ถ้าเราใช้ตัวเก็บข้อมูลหรือรักษาข้อมูลทุกจุด จะเป็นการสิ้นเปลืองอุปกรณ์มาก เราจึงใช้ไอซีบัฟเฟอร์แบบมีแลตช์ (IC Buffer Noninverting with Latch) เพื่อรักษาข้อมูลเอาไว้ตัวละ 8 บิต เพื่อรักษาข้อมูลที่ได้รับมาจากแรมเอาไว้ ซึ่งในโครงการนี้ได้เลือกใช้ไอซีเบอร์ 74HC373 เนื่องจากเป็นไอซีที่สามารถหาได้ง่ายทั่วไป และมีราคาไม่แพง

ในการใช้ไอซีบัฟเฟอร์เพื่อรักษาข้อมูลเอาไว้ นั้น ถ้าหากมีไอซีบัฟเฟอร์เพียง 1 ชุด (12 ตัว) ไอซีบัฟเฟอร์ ที่ได้รับข้อมูลจากแรมก่อนก็สามารถส่งข้อมูลให้ LED ติดสว่างได้นานกว่าไอซีบัฟเฟอร์ ที่ได้รับข้อมูลที่หลัง เนื่องจากเมื่อแรมส่งข้อมูลครบ 16 ครั้ง วงจรก็จะเริ่มทำการสแกนในแนวคอลลัมน์ถัดไปทันที ทำให้ไอซีบัฟเฟอร์ที่ได้รับข้อมูลที่หลังสามารถส่งข้อมูลได้เพียงชั่วครู่เท่านั้น ทำให้เกิดปัญหาว่า LED จะสว่างไม่เท่ากันในแต่ละแถว นอกจากนี้หากเราทำการแลตช์ข้อมูลเอาไว้ในขณะที่ไอซีบัฟเฟอร์ตัวแรกๆ ทำการแลตช์ข้อมูลในคอลลัมน์ปัจจุบันอยู่ ไอซีบัฟเฟอร์ตัวหลังจะยังมีข้อมูลของคอลลัมน์ที่แล้วอยู่จะทำให้ข้อมูลที่ส่งออกมาเกิดการผิดพลาดได้

จากปัญหาที่ได้กล่าวมา เราจึงใช้ไอซีบัฟเฟอร์เพิ่มอีกหนึ่งชุดเพื่อใช้ในการรักษาระดับข้อมูล ดังแสดงในรูปที่ 3.5

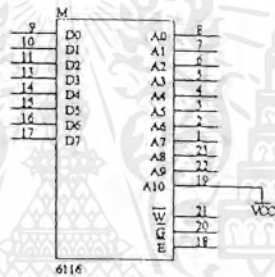


รูปที่ 3.5 การต่อวงจรรักษาระดับข้อมูลโดยใช้ไอซีบัฟเฟอร์สองชุด

ในวงจรส่วนนี้ขาที่ 1 ของไอซีบัฟเฟอร์ทุกตัวถูกต่อลงกราวด์ตลอดเวลาเพื่อให้ไอซีทำงานตลอดเวลา โดยสามารถแบ่งส่วนของวงจรนี้ได้เป็นไอซีบัฟเฟอร์ชุดที่ 1 จะทำหน้าที่รับข้อมูลจากแรม โดยข้อมูลที่รับมาจะเป็นข้อมูลที่ใช้ทำการแสดงในแนวคอลัมน์ถัดไปโดยขาแลตซ์ (ขาที่ 11) ของไอซีบัฟเฟอร์แต่ละตัวจะได้รับสัญญาณมาจากวงจรถอดรหัสในแนวโรว์ จากขา Y0 - Y12 ตามลำดับ เพื่อจะเป็นการรับข้อมูลจากแรมในแอดเดรสที่ xx0H ถึง xxCH ส่วนไอซีบัฟเฟอร์ชุดที่ 2 จะทำหน้าที่รักษาระดับข้อมูลที่กำลังแสดงอยู่ในแนวคอลัมน์ขณะนั้น โดยขาแลตซ์ของไอซีชุดที่ 2 ทุกตัวจะต่ออยู่กับวงจรถอดรหัสในแนวโรว์ จากขา Y16 เพื่อรับข้อมูลมาจากไอซีบัฟเฟอร์ชุดที่ 1 ที่ได้รับข้อมูลจากแรมครบทั้ง 12 ตัวแล้ว ดังนั้นข้อมูลในไอซีบัฟเฟอร์ชุดที่ 2 จะเปลี่ยนไปทุกๆ ครั้งที่วงจรถอดรหัสเส้นที่ 16 ทำงาน ทำให้ LED สว่างได้นานเท่ากันและยังสว่างได้นานกว่าการใช้ไอซีบัฟเฟอร์ เพียงชุดเดียวอีกด้วย ซึ่งไอซีบัฟเฟอร์ชุดที่หนึ่งนั้น จะรับข้อมูลจากหน่วยความจำแรมขนาด 2K x 8 bit โดยการส่งข้อมูลจากหน่วยความจำจะอธิบายในหัวข้อต่อไป

3.5 หน่วยความจำ

ในโครงการนี้เราได้ออกแบบให้บอร์ดแสดงผลแต่ละบอร์ดมีความละเอียด 91x60 จุดและใช้วงจรถอดรหัสในแนวโรว์และคอลัมน์เป็น 16x64 ดังนั้นเราจำเป็นต้องใช้หน่วยความจำสำหรับเก็บข้อมูลภาพเป็น $16 \times 64 = 1024$ ไบต์ (Byte) ซึ่งเป็นขนาดของหน่วยความจำที่เราจำเป็นต้องใช้แต่ใช้ได้ในปัจจุบันไอซีแรมที่มีขนาด 1 กิโลไบต์ หาได้ยากมากในท้องตลาดและมีราคาไม่ต่างจากไอซีแรมขนาด 2 กิโลไบต์ มากนัก เราจึงเลือกใช้ไอซีแรมเบอร์ 6116 ซึ่งมีขนาด 2 กิโลไบต์ แทน โดยทำการต่อขาแอดเดรส A10 ลงกราวด์นั้นเพื่อใช้หน่วยความจำเพียงแค่ 1 กิโลไบต์ แทน ดังรูปที่ 3.6



รูปที่ 3.6 ลักษณะของขาสัญญาณแรมเบอร์ 6116

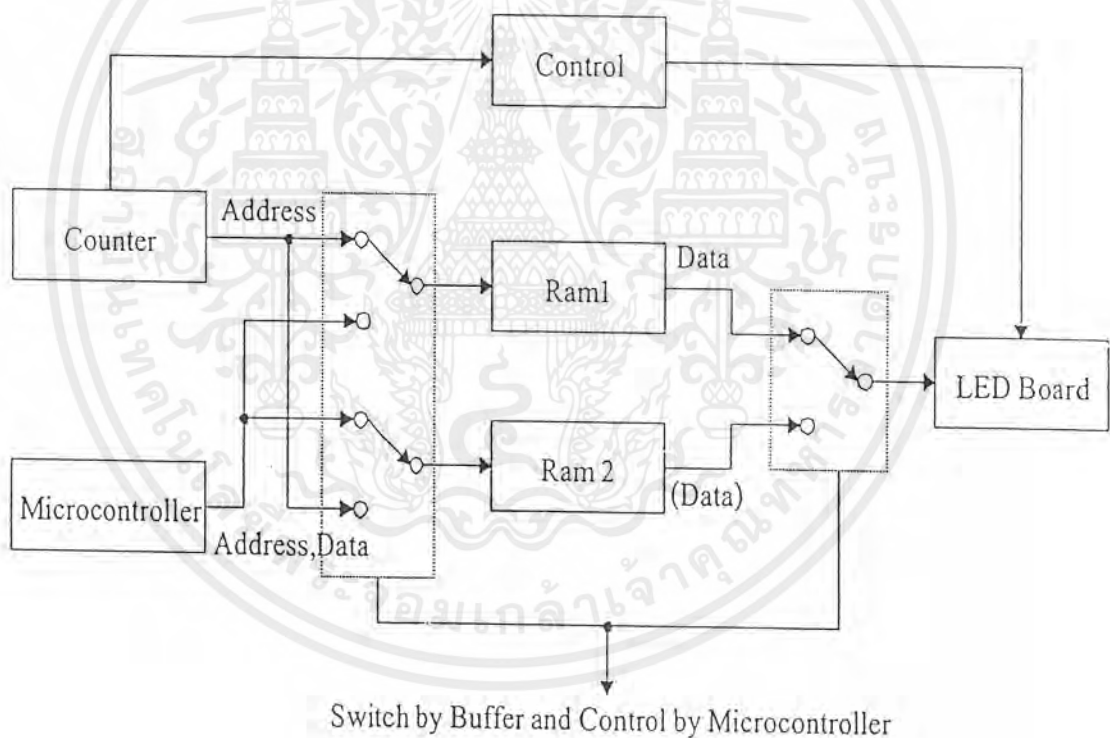
วงจรถวนนี้เราจะแบ่งหน่วยความจำออกเป็น 2 ตัว คือ

1. หน่วยความจำ ขณะส่งข้อมูลภาพปัจจุบัน
2. หน่วยความจำ รับข้อมูลของภาพถัดไป จากไมโครคอนโทรลเลอร์

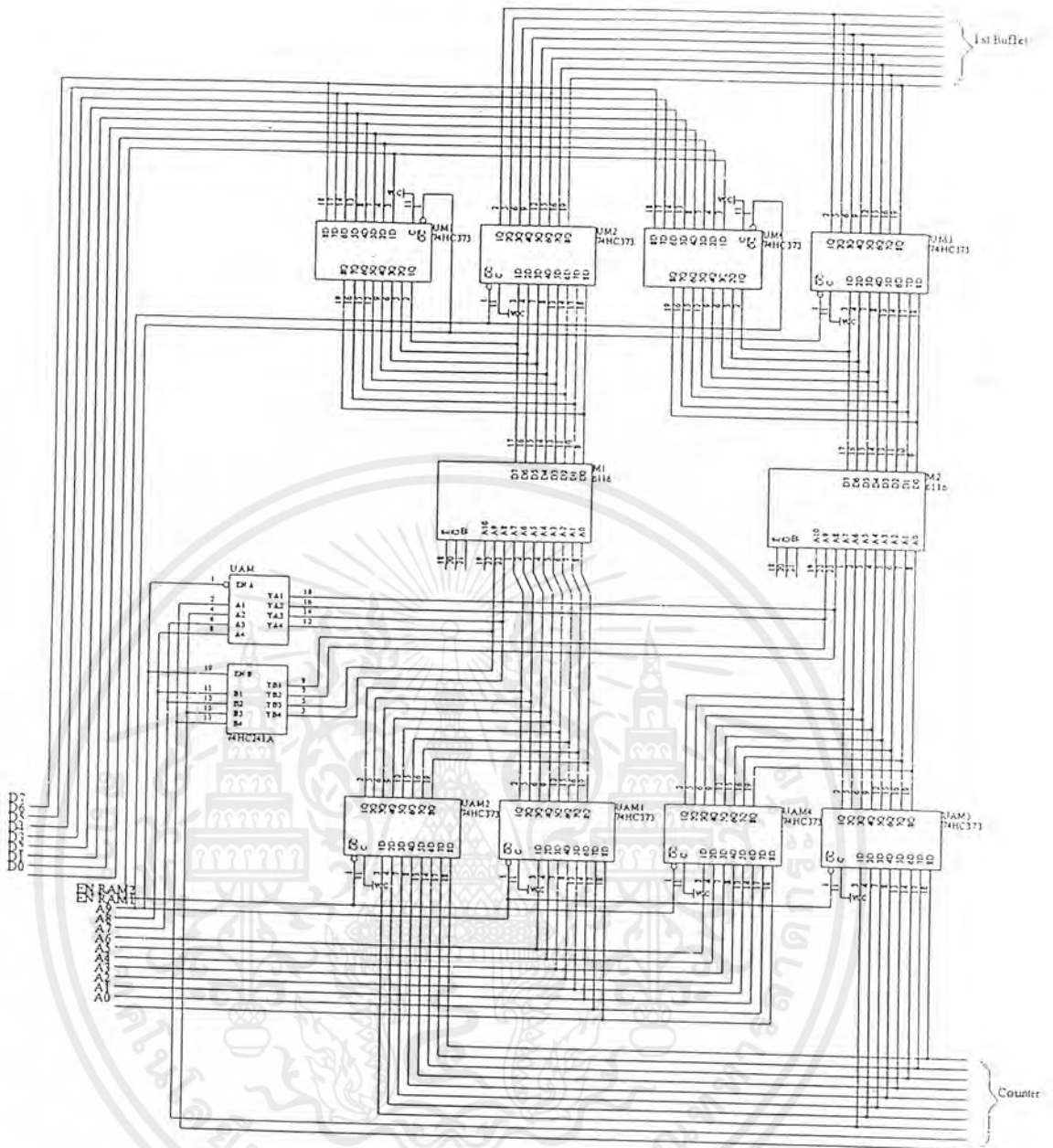
หน่วยความจำทั้งสองส่วนจะทำหน้าที่สลับกันไปในารแสดงภาพแต่ละครั้ง โดยที่ในขณะที่หน่วยความจำตัวหนึ่งอยู่ในช่วงการอ่านข้อมูล และจะส่งข้อมูลให้กับ Buffer เพื่อทำการแสดงผลหน่วยความจำตัวนี้จะได้รับแอดเดรสจากวงจรถวนที่มีค่าเพิ่มขึ้นตามสัญญาณนาฬิกา ทำให้ข้อมูลจากแรมตัวนี้มีค่าเปลี่ยนไปตามแอดเดรสที่ได้รับ ซึ่งสัมพันธ์กับการ Latch Buffer ที่ควบคุมโดยวงจรถอดรหัส ทำให้มีการแสดงผลเป็นภาพที่ถูกเก็บไว้ในหน่วยความจำตัวนี้ ในขณะที่เดียวกันหน่วยความจำอีกตัวหนึ่งจะอยู่ในช่วงการเขียนข้อมูลจากไมโครคอนโทรลเลอร์โดยไมโครคอนโทรลเลอร์ จะทำการส่งทั้งข้อมูลและแอดเดรสให้กับแรมตัวนี้ เพื่อเขียนข้อมูลภาพที่ต้องการจะแสดงในลำดับต่อไปลงไปหน่วยความจำ เมื่อไมโครคอนโทรลเลอร์ได้ทำการเขียนข้อมูลลงไปแรมตัวนี้เสร็จแล้วแรมตัวแรกได้ทำการแสดงผลไปเป็นระยะเวลาที่ได้กำหนดไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ก็จะทำการสลับให้แรมตัวที่สอง ทำการส่งข้อมูลออกไปแสดงผลเป็นภาพที่ได้เขียนข้อมูลเอาไว้และทำการเขียนข้อมูลที่ต้องการแสดงผลต่อไปลงในแรมตัวแรกแทน ทำอย่างนี้สลับกันไปก็จะได้ภาพที่สามารถเปลี่ยนแปลงไปได้เรื่อยๆ ซึ่งการจะทำการสลับแรมไปมานี้จำเป็นต้องใช้ Buffer เพื่อที่จะช่วยกำหนดทิศทางของแรมแต่ละตัวว่าต้องการให้แรมตัวไหนติดต่อกับไมโครคอนโทรลเลอร์และแรมตัวไหนทำการส่งข้อมูลไปแสดงผล โดยในวงจรนี้จะใช้ไอซีเบอร์ 74HC241A เข้าช่วยด้วย ซึ่งเป็นไอซีบัฟเฟอร์ขนาด 4 บิตของชุด ในตัวเดียวกัน ในการสลับแรมที่จะส่งข้อมูลไปแสดงผลและทำการเขียนข้อมูล สามารถแสดงเป็นบล็อกไดอะแกรมการสลับแรมได้ดังรูปที่ 3.7 และรูปวงจรถ่ายสลับแรมรูปที่ 3.8



รูปที่ 3.7 บล็อกไดอะแกรม (Block diagram) แสดงการสลับแรม



รูปที่ 3.8 วงจรที่ช่วยในการสลับข้อมูลของแรมสองตัว

3.6 วงจรขับกระแส

ในวงจรส่วนนี้จะแบ่งออกเป็นสองการขับกระแสตามแนวโรว์ และส่วนสแกนตามแนวคอลัมน์ ส่วนขับกระแสตามแนวโรว์จะต่อจากไอซีขับไฟเฟอร์ที่มาจากวงจรรักษาระดับข้อมูล โดยถ้าเอาต์พุตของไอซีลอจิก 1 หมายถึงหลอดไฟไม่สว่าง และถ้าเอาต์พุตออกเป็น 0 หลอดไฟจะสว่างโดยวงจรนี้ได้ออกแบบกระแสไว้มากที่สุด 100 มิลลิแอมป์ ต่อ LED 1 หลอด โดยจะต้องใช้กระแสไฟขนาด $0.1 \times 91 = 9.1$ แอมป์ ในการจ่ายไฟทั้งหมดต่อ 1 แถว ได้ออกแบบโดยใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์เบอร์ BC558B ในการจ่ายกระแสไฟให้ LED แต่ละดวง และจากวงจรจะต้องกำหนดค่าความต้านทาน เพื่อป้องกันความเสียหายต่อทรานซิสเตอร์โดยคำนวณจากค่าอัตราขยาย (Beta) ทรานซิสเตอร์คือ 200 , กระแสไฟที่กำหนดไว้มีค่าเท่ากับ $I_c = 100$ มิลลิแอมป์ โดยใช้ $I_b = 500$ มิลลิแอมป์ โดยที่ทรานซิสเตอร์ เบอร์ BC558B นี้สามารถทนกระแสไฟสูงสุด 100 มิลลิแอมป์ และต้องใช้

$$\begin{aligned} R_{bias} &= (V_{cc} - 0.7) / I_b \\ &= (5 - 0.7) / 500 \cdot 100 \text{ ยกกำลัง } (-6) \\ &= 8600 \text{ โอห์ม} \end{aligned}$$

ส่วนการขับกระแสตามแนวตั้ง จะรับสัญญาณจากวงจรทรานซิสเตอร์ในแนวคอลัมน์ (6 - 64 Output Active High) เพื่อมาขับทรานซิสเตอร์ เบอร์ TIP 100 ซึ่งมีค่าอัตราขยายขั้นต่ำ 750 โดยต้องใช้

$$\begin{aligned} R_{bias} &= (5 \text{ V} - 0.7) * h_{ic} / I_c \\ &= (5 - 0.7) * 750 / 9.1 \\ &= 340 \text{ โอห์ม} \end{aligned}$$

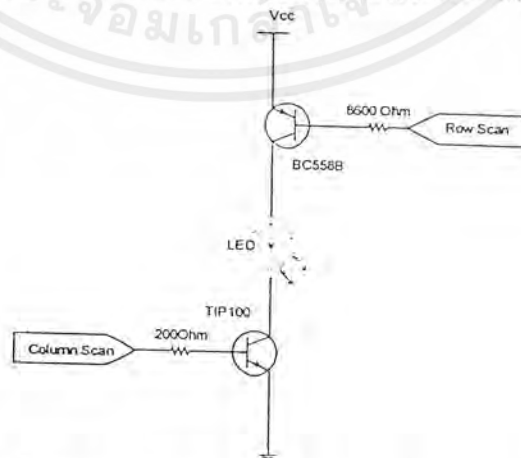
ค่า V_{cc} จะคำนวณได้จาก

$$\begin{aligned} V_{cc} &= V_{CE1} + V_{CE2} + V_{LED \text{ ที่ } 100\text{mA}} \\ &= 0.2 + 2.5 + 2.8 \\ &= 5.5 \text{ V} \end{aligned}$$

โดยที่สามารถคำนวณได้จากวงจรขับกระแส ไม่เช่นนั้นทรานซิสเตอร์จะทำงานตลอดเวลา

$$V_{cc} < 5 \text{ V} + 0.7 = 5.7 \text{ V}$$

ซึ่งจากวงจรและค่าความต้านทานที่คำนวณได้ทั้งหมด สามารถแสดงได้ดังรูปที่ 3.8



รูปที่ 3.9 วงจรในส่วนขับกระแสของบอร์ดแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 ส่วนแสดงผล

จากขนาดของบอร์ดเป็น 91 x 60 จุด ประมาณความกว้างต่อความยาว มีสัดส่วนเป็นสองต่อสามและต่อกันแบบเมตริกซ์ โดยในโครงการนี้เลือกใช้ LED สำเร็จรูปขนาด 5x7 จุดที่ต่อกันในลักษณะเมตริกซ์ในตัวของมันเอง มีขนาดเล็ก และสะดวกในการเข้าถึงข้อมูล โดยออกแบบลายวงจรเพียงหน้าเดียว เพราะเนื้อที่ในการเขียนลายวงจรมีขนาดจำกัด และอีกด้านหนึ่งของเมตริกซ์นี้จะมีกระแสไหลผ่านสูงมาก (ประมาณ 9.5 แอมป์ ตามที่ได้จากการคำนวณ) จึงจำเป็นต้องใช้สายไฟเชื่อมเข้าช่วยเพื่อนำกระแสและยังเป็นการต่อลายวงจรส่วนที่เหลืออีกด้วย

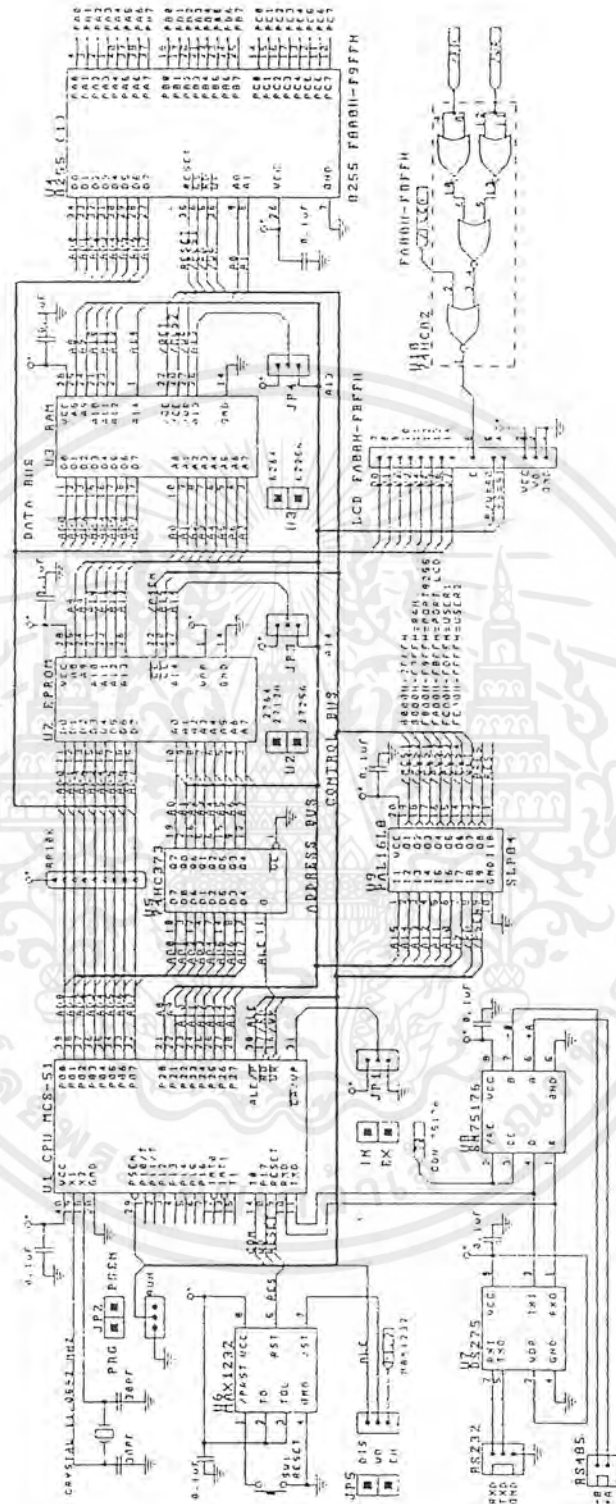
3.8 ส่วนไมโครคอนโทรลเลอร์

เป็นวงจรควบคุมการแสดงผลของบอร์ดแสดงผลไฟวิ่งแบบต่างๆ เช่น เลื่อนภาพขึ้น – ลง เลื่อนภาพไปทางซ้าย – ขวา แสดงการกระพริบของภาพ และแสดงนิ่ง โดยมีการเชื่อมต่อกับส่วนแสดงผล LED บอร์ด ด้วยการต่อพอร์ต 40 ขา ซึ่งประกอบไปด้วยขาต่างๆ เช่น ขาแอดเดรส (ADDRESS) ขาข้อมูล (DATA) ขาควบคุม (CONTROL)

ซึ่งการเชื่อมต่อจะผ่านไอซีเบอร์ 8255 ทำให้ต้องต่อขา ไอซี 8255 จำนวน 2 ตัว โดยมีการกำหนดขาพอร์ตต่างๆ ดังนี้

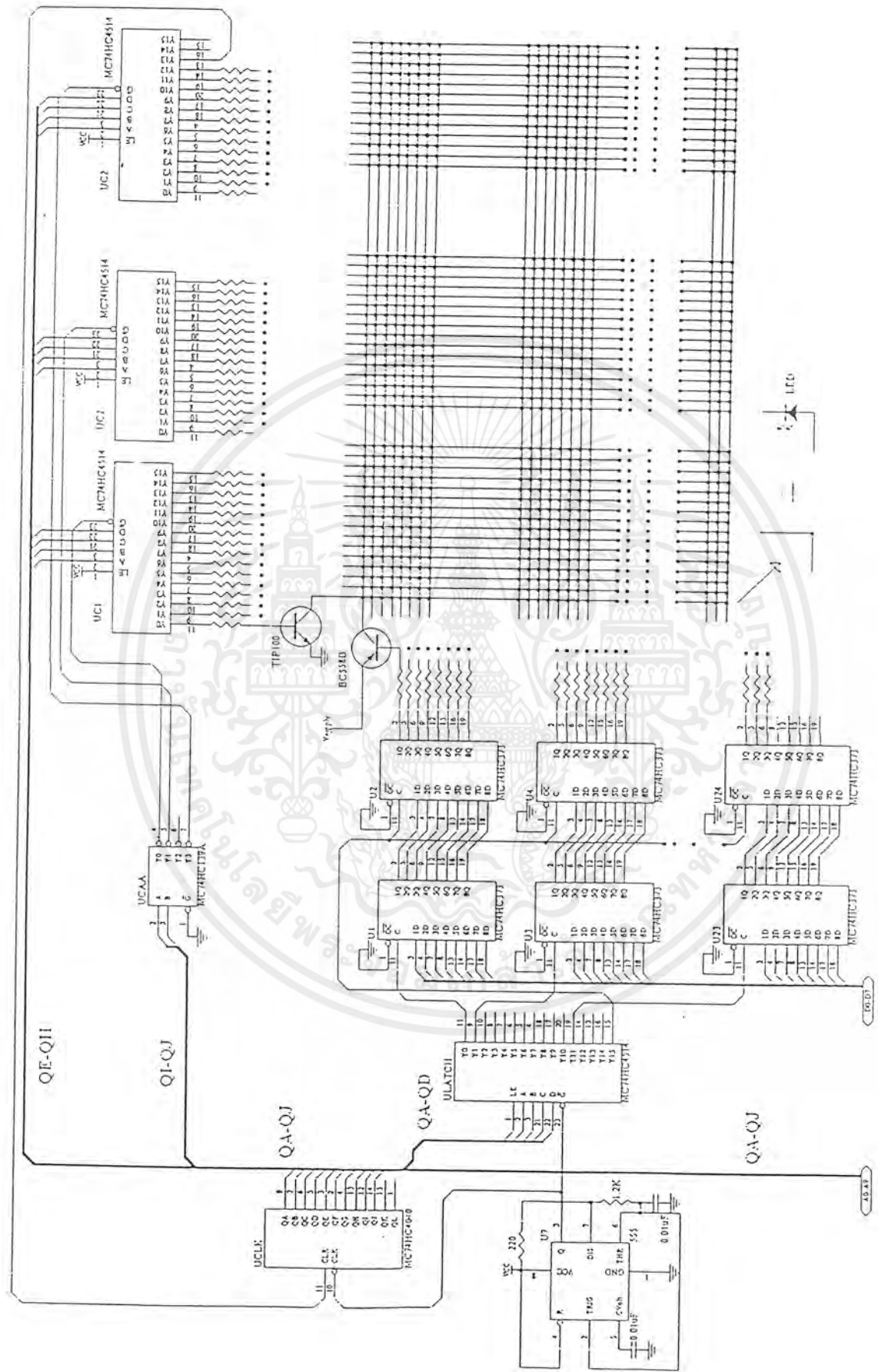
- พอร์ต P1 ของขา 8255 ตัวที่ 1 กำหนดเป็น ขาแอดเดรส A0 - A7
- พอร์ต P2 ของขา 8255 ตัวที่ 1 กำหนดเป็น ขาข้อมูล D0 - D7
- พอร์ต P3 ของขา 8255 ตัวที่ 1 กำหนดเป็น ขาแอดเดรส A8 - A9
- พอร์ต P1 ของขา 8255 ตัวที่ 2 กำหนดเป็น อ่านและเขียนข้อมูลแรม 1 และแรม 2
- พอร์ต P2 ของขา 8255 ตัวที่ 2 กำหนดเป็น ขาคัดต่อแรม 1 และแรม 2

โดยการทำงานได้ใช้ อีพรอมอีมูเลเตอร์ (EPROM EMULATOR) ในการรับส่งข้อมูลของภาพมาจากโปรแกรมทางคอมพิวเตอร์ส่วนบุคคล โดยใช้การเขียนโปรแกรมแล้วรับส่งข้อมูลผ่านทางพอร์ตอนุกรม



รูปที่ 3.10 วงจรส่วนไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 วงจรรวม

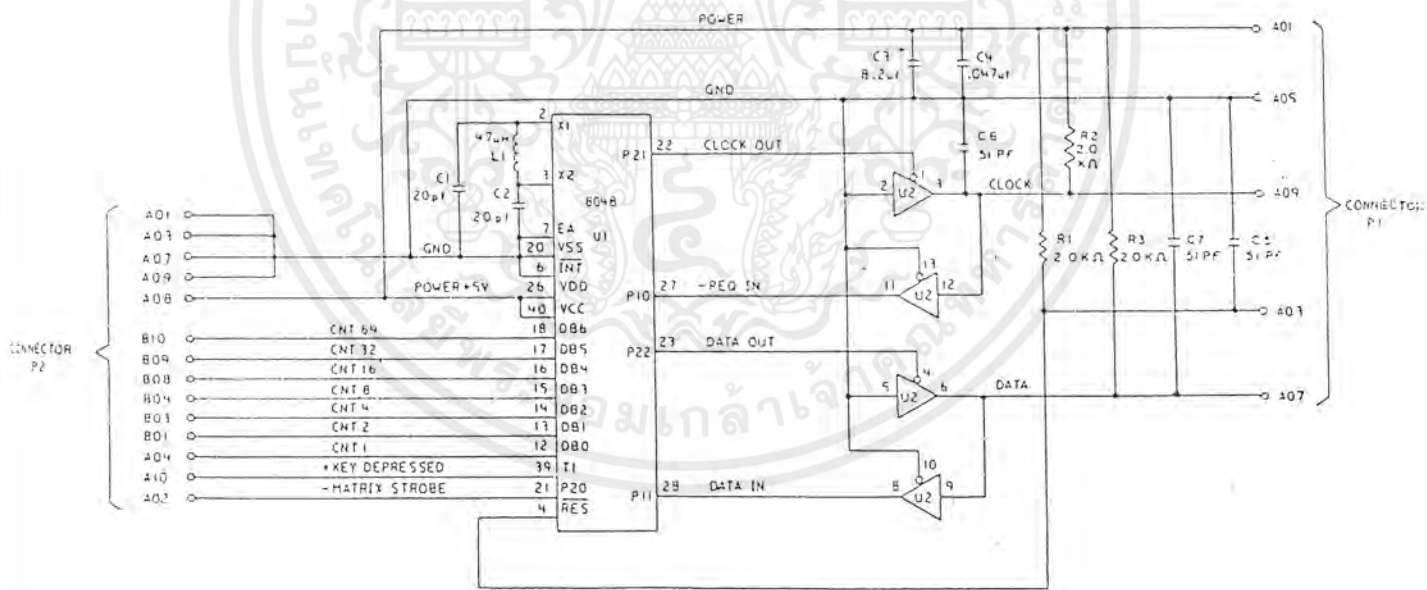
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 คีย์บอร์ดเบื้องต้น

4.1 โครงสร้างคีย์บอร์ด

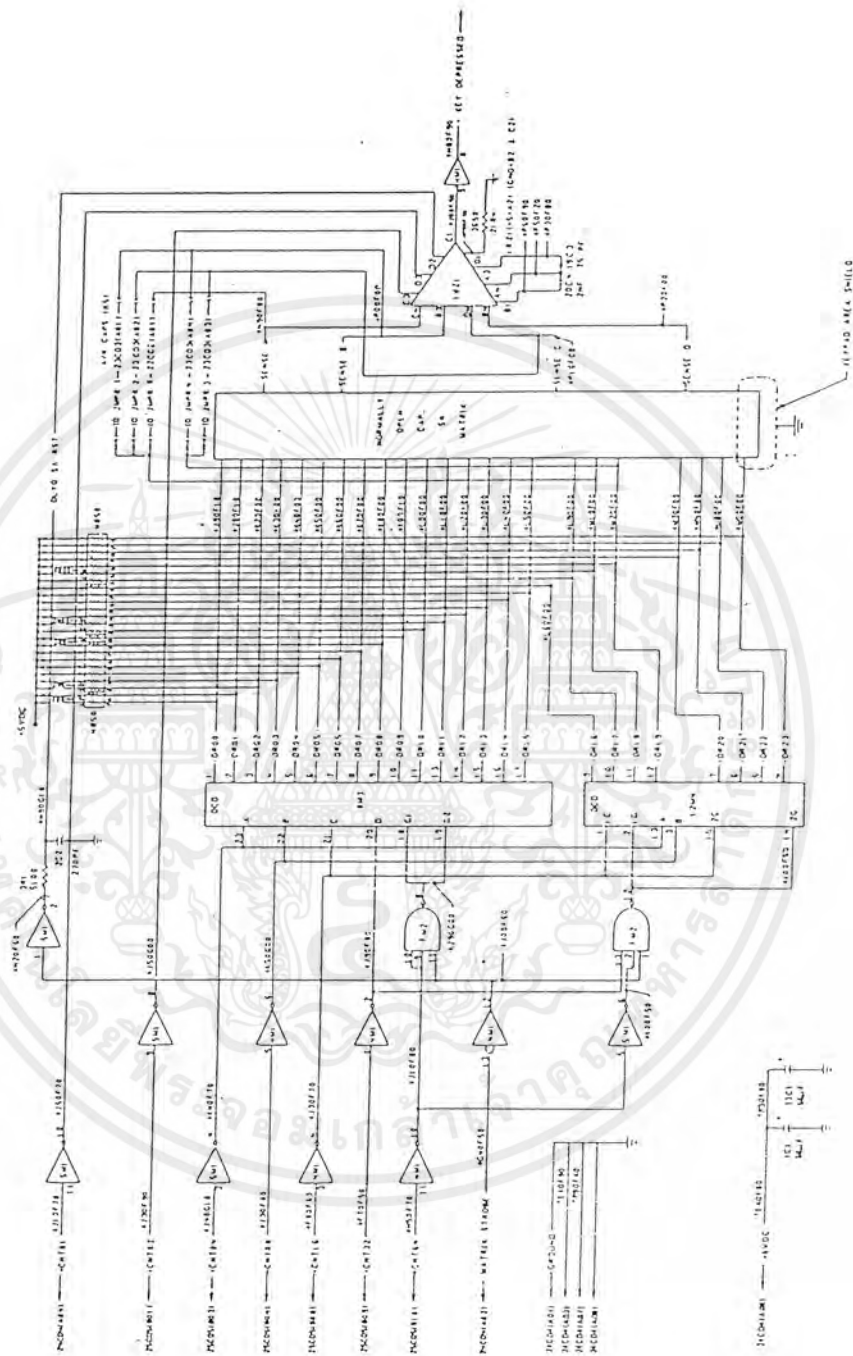
ตัวคีย์บอร์ดในที่นี้ใช้ขนาด 101 คีย์ บนคีย์บอร์ดจะมีไมโครโปรเซสเซอร์ 8048 1 ตัว ทำหน้าที่ในการสแกนคีย์บอร์ดและส่งค่ารหัสสแกน (Key Scan Code) มายังเมนบอร์ด (Main Board) โดยการทำงานเมื่อมีการกดคีย์ ไมโครโปรเซสเซอร์ก็จะรับรู้และส่งข้อมูลออกมาตามค่ารหัสสแกนคีย์นั้นๆ

การส่งข้อมูลออกมานี้จะส่งออกทางพอร์ต 2 ขา 2 และส่งสัญญาณนาฬิกา (Clock) ออกทางพอร์ต 2 ขา 1 ตามค่ารหัสสแกนคีย์ที่ได้รับจากการกดคีย์ โดยจะอ่านเข้าทางขาสัญญาณข้อมูล (Data Bus) ดังแสดงในรูปที่ 4.1 และโครงสร้างของคีย์บอร์ดแสดงในรูปที่ 4.2



รูปที่ 4.1 แสดงวงจรภายในคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

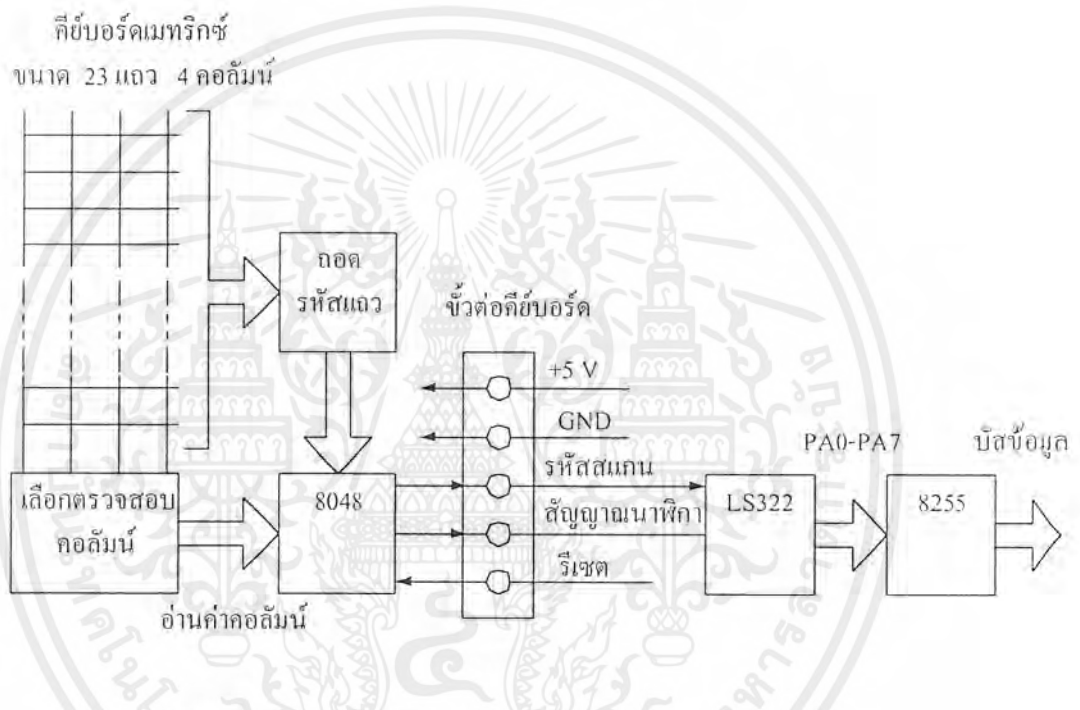


รูปที่ 4.2 แสดงวงจรภายในคีย์บอร์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การเชื่อมต่อคีย์บอร์ด

ในการเชื่อมต่อคีย์บอร์ดนั้น โครงสร้างทางฮาร์ดแวร์ (Hardware) มีการทำงานในลักษณะที่ซับซ้อนพอสมควร กล่าวคือ โครงสร้างของอินพุตจากคีย์บอร์ดนี้ทำงานร่วมกับอินเทอร์รัพท์ (Interrupt) ด้วย โครงสร้างของคีย์บอร์ดที่ต่อกับเมนบอร์ด (Main Board) จะใช้สายเชื่อมต่อ 5 เส้น โดยมีรายละเอียดบล็อกไดอะแกรมของโครงสร้างการเชื่อมต่อได้จากรูปที่ 4.3



รูปที่ 4.3 โครงสร้างการเชื่อมต่อคีย์บอร์ดกับเมนบอร์ด

สังเกตว่าคีย์บอร์ดของไมโครคอมพิวเตอร์ 16 บิต นี้มีสายที่สำคัญคือ ไฟบวก 5 โวลต์ , กราวด์ , สายสัญญาณข้อมูล (Data Bus) , สายสัญญาณนาฬิกา (Clock) , สัญญาณรีเซ็ต (Reset) ลักษณะของหัวต่อจะเป็นหัวต่อแบบ DIN โดยมีไดอะแกรมของหัวต่อดังในรูปที่ 4.4



คอนเนคเตอร์สำหรับคีย์บอร์ดแบบ DIN (5 ขา)

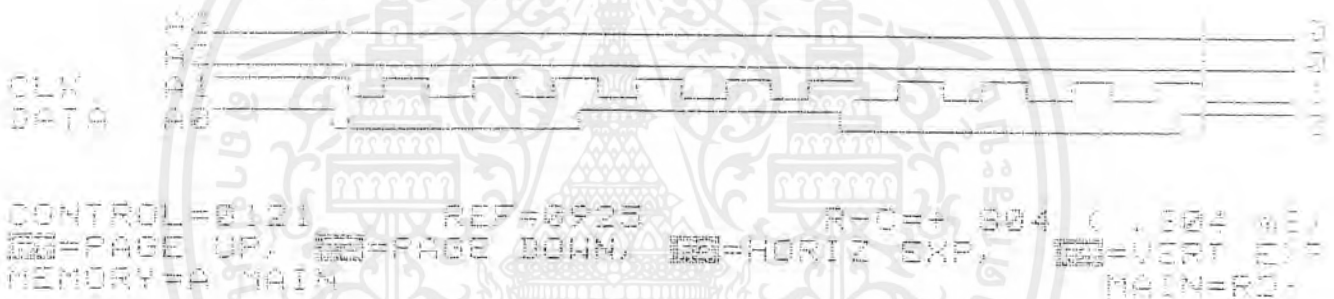
- ขาที่ 1 : สัญญาณนาฬิกาคีย์บอร์ด (Keyboard Clock)
- ขาที่ 2 : สัญญาณข้อมูลคีย์บอร์ด (Keyboard Data)
- ขาที่ 3 : สัญญาณรีเซ็ตคีย์บอร์ด (Keyboard Reset)
- ขาที่ 4 : กราวด์ (Ground)
- ขาที่ 5 : ไฟบวก 5 โวลต์ (+5 Volts)

รูปที่ 4.4 แสดงหัวต่อแบบ DIN (5 ขา) ของคีย์บอร์ดและสัญญาณของแต่ละขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การออกแบบคีย์บอร์ดอินเตอร์เฟซ

ในการออกแบบวงจรส่วนนี้เป็นการออกแบบเพื่อดึงค่าของข้อมูลออกมาจากตัวคีย์บอร์ดซึ่งค่าข้อมูลที่ส่งออกมาจากตัวคีย์บอร์ดนั้นจะเป็นการส่งข้อมูลแบบอนุกรมที่ใช้สัญญาณนาฬิกาเป็นตัวทำให้เกิดการซิงโครนัส (Synchronous) ซึ่งข้อมูลหนึ่งจะถูกจะเท่ากับสัญญาณนาฬิกาหนึ่งลูกด้วยเช่นกัน ซึ่งการส่งข้อมูลของตัวคีย์บอร์ดนั้นเมื่อเราทำการกดตัวอักษร แป้นคีย์บอร์ดก็จะส่งค่าสแกนไค้ดออกมาตามค่าของแป้นนั้นๆ ซึ่งมีค่า 8 บิต แต่เมื่อมีการปล่อยแป้นออกจากคีย์บอร์ดจะทำการเซ็ทค่าบิตแรกของข้อมูลให้เป็นลอจิกหนึ่ง "1" เพราะฉะนั้นค่าสแกนไค้ดที่ส่งออกมาจะเป็น 9 บิต ดังตัวอย่างของค่าสแกนไค้ดของตัวอักษรของแป้นตัว A ที่มีค่าสแกนไค้ดประจำแป้นของมันคือ E3H ซึ่งมีค่าไบนารีคือ 11100011B ซึ่งผลที่ได้จะแสดงในรูปที่ 4.5 ซึ่งได้จากการทำการวัดโดยใช้ลอจิกอานาไลเซอร์



รูปที่ 4.5 แสดงค่าสแกนไค้ดของแป้นตัวอักษร A

เพื่อให้ได้ค่าสแกนไค้ดตามที่ทำการวัดเราจึงนำไอซีเบอร์ 74LS322 ซึ่งโครงสร้างภายในประกอบด้วยดีฟลิปฟลอป (D-F/F) 8 บิต ซึ่งทำหน้าที่เป็น 8 Bit Shift Registers With Sign Exten ซึ่งบิตที่ 9 ของข้อมูลมันสามารถนำมาทำการค้างของข้อมูลได้เมื่อข้อมูลที่เป็นสแกนไค้ดเข้ามาครบ 8 บิต

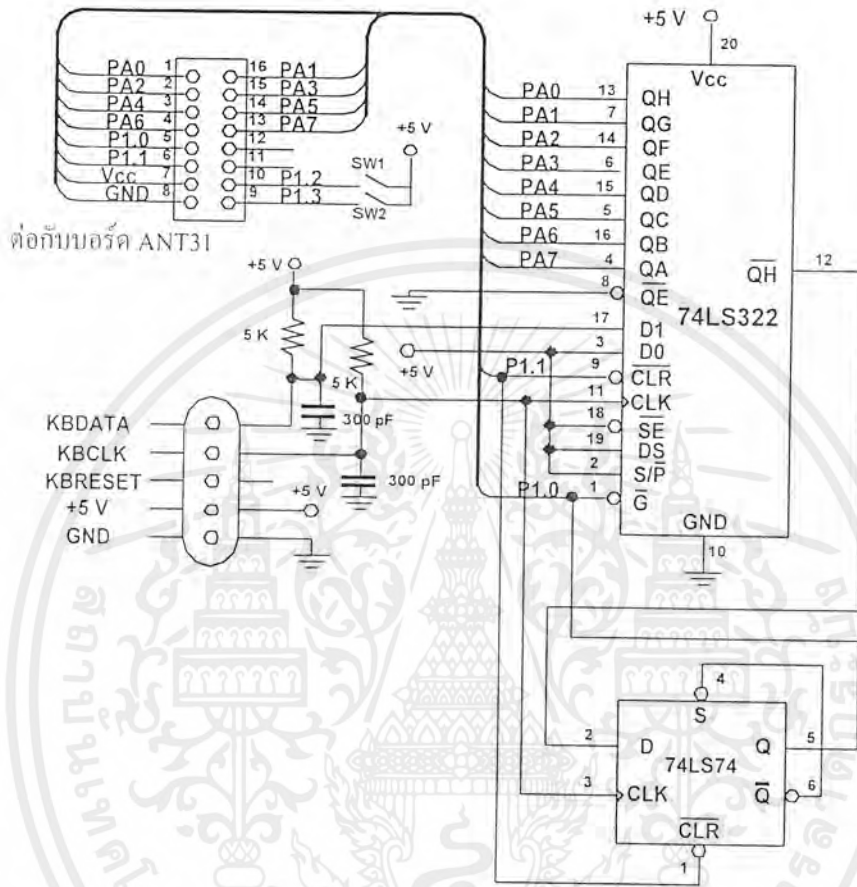
4.4 การทำงานของวงจร

จากวงจรจะเห็นได้ว่าสัญญาณของคีย์บอร์ดที่นำมาใช้งานมีอยู่ 2 สายสัญญาณ คือ สายสัญญาณข้อมูล (data) และสายสัญญาณนาฬิกา (clock) เพื่อนำสัญญาณทั้งสองมาควบคุมการเปลี่ยนจากสัญญาณข้อมูลแบบอนุกรมให้เป็นสัญญาณแบบขนาน

คีย์บอร์ดจะส่งข้อมูลเป็นระดับลอจิกคือ "0" หรือ "1" สัมพันธ์กับการส่งสัญญาณนาฬิกา 1 พัลส์ต่อข้อมูล 1 บิต จากวงจรใช้อุปกรณ์ 2 ตัว คือ IC เบอร์ 74LS322 ซึ่งเป็นชิพท์รีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Shift Register) ขนาด 8 บิต แบบรับอินพุตแบบอนุกรมและส่งออกที่พุกแบบขนาน (serial in parallel out) และ IC เบอร์ 74LS74 มาต่อวงจรกัน



รูปที่ 4.6 วงจรที่ใช้เชื่อมต่อคีย์บอร์ดกับบอร์ด ANT-31

เมื่อมีการกดคีย์บอร์ด คีย์บอร์ดจะส่งข้อมูลออกมาตามคีย์ที่มีการกด โดยขาสัญญาณข้อมูลจากคีย์บอร์ดจะต่ออยู่ที่ขา D1 และขาสัญญาณนาฬิกาจากคีย์บอร์ดจะต่อเข้าที่ขา CLK ของ IC1 เพื่อทำการเลื่อนข้อมูลแบบอนุกรมให้เป็นข้อมูลแบบขนาน สำหรับ IC2 จะนำเอาสัญญาณข้อมูลจากขา Q/H ของ IC1 มาตรวจสอบว่ามีการกดคีย์หรือไม่ นั่นคือถ้ามีการกดคีย์ขา Q/H จะเป็นลอจิก “1” เสมอ และขา CLK ของ IC2 ได้รับสัญญาณนาฬิกานั้นคือบอกได้ทันทีว่ามีการกดคีย์เกิดขึ้นแล้ว โดยส่งเข้าที่ขา P1.0 บอกให้ไมโครคอนโทรลเลอร์รู้ และทำการอ่านค่ารหัสของคีย์ได้แล้ว เมื่อทำการอ่านค่ารหัสของคีย์เรียบร้อยแล้วต้องทำการเคลียร์ค่าของข้อมูลเดิม โดยส่งพัลส์ลอจิก “0” ไปที่ขา CLR ของ IC1 และ IC2 เพื่อที่จะได้รับรับการกดคีย์ของตัวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ส่วนโปรแกรม

สำหรับโครงงานนี้ได้ทำการแบ่งส่วนของโปรแกรมออกเป็น 2 ส่วน คือ

1. ส่วนโปรแกรมไมโครคอนโทรลเลอร์
2. ส่วนโปรแกรมเดสก์ท็อป

5.1 โปรแกรมไมโครคอนโทรลเลอร์

ส่วนโปรแกรมไมโครคอนโทรลเลอร์ คือส่วนที่ทำการรับข้อมูลของภาพและรูปแบบของการแสดงผลมาทำการประมวลผลให้ได้ตามที่กำหนดไว้ในส่วนของโปรแกรมเดสก์ท็อป และทำการรับข้อมูลตัวอักษรจากคีย์บอร์ดนำมาแสดงบนบอร์ดLCDและบอร์ดแสดงผลตามคีย์ที่กดไปนั้น

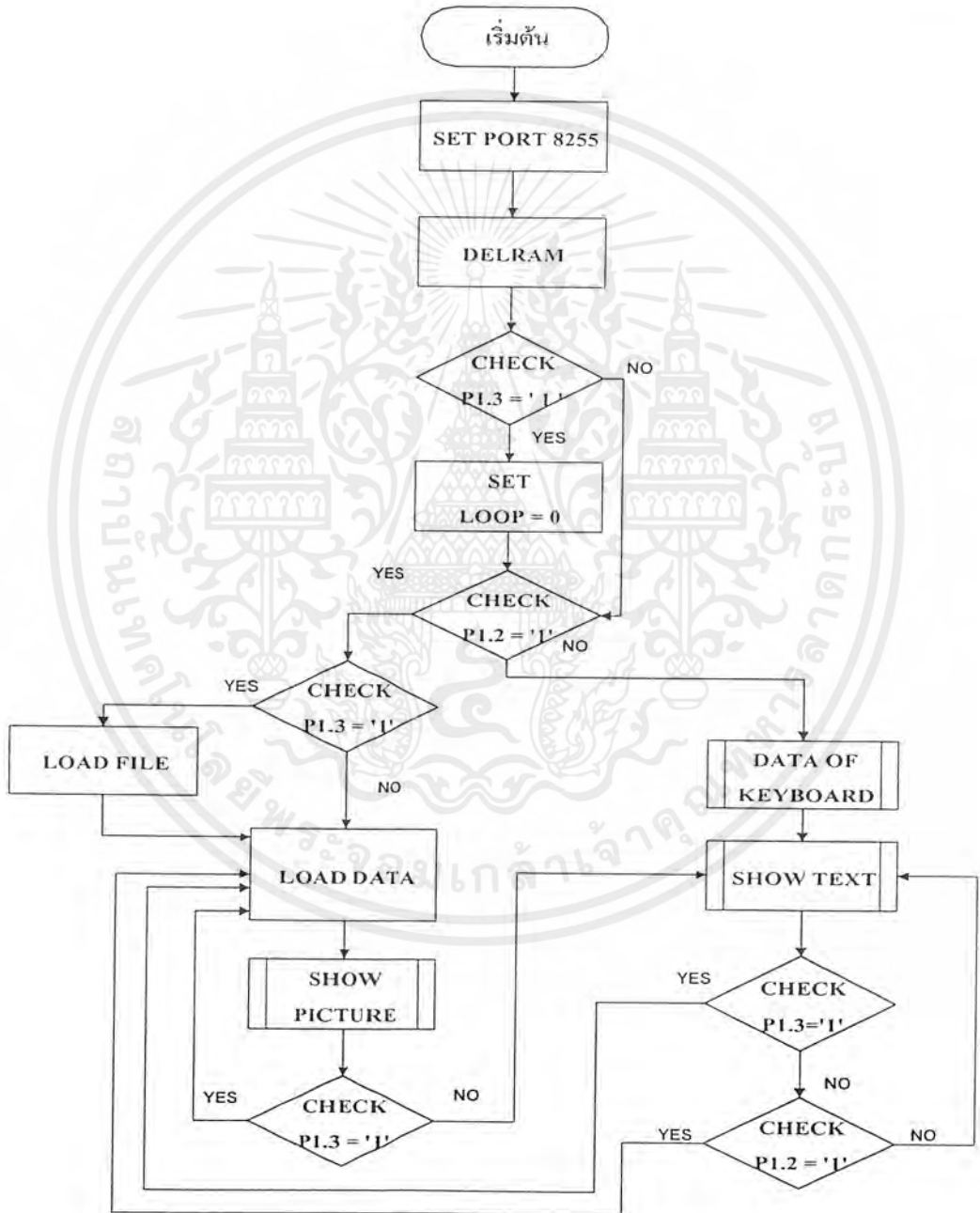
สามารถแบ่งส่วนโปรแกรมไมโครคอนโทรลเลอร์ได้เป็นส่วนๆ ตามหน้าที่ดังนี้

1. โปรแกรมหลัก
2. โปรแกรมย่อยสำหรับการเลื่อนภาพขึ้น
3. โปรแกรมย่อยสำหรับการเลื่อนภาพลง
4. โปรแกรมย่อยสำหรับการเลื่อนภาพไปทางขวา
5. โปรแกรมย่อยสำหรับการเลื่อนภาพไปทางซ้าย
6. โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงขึ้นขวา
7. โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงลงขวา
8. โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงขึ้นซ้าย
9. โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงลงซ้าย
10. โปรแกรมย่อยสำหรับการเพิ่มขนาดภาพ
11. โปรแกรมย่อยสำหรับการลดขนาดภาพ
12. โปรแกรมย่อยสำหรับการแสดงภาพกระพริบ
13. โปรแกรมย่อยสำหรับการแสดงภาพนิ่ง
14. โปรแกรมย่อยสำหรับการสร้างรูปเส้นตรง
15. โปรแกรมย่อยสำหรับการสร้างรูปสี่เหลี่ยม
16. โปรแกรมย่อยสำหรับการเซ็ท (SET) ค่าในหน่วยความจำ
17. โปรแกรมย่อยสำหรับการรับค่าจากคีย์บอร์ด
18. โปรแกรมย่อยสำหรับการแสดงรูปภาพ
19. โปรแกรมย่อยสำหรับการแสดงตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.1 โปรแกรมหลัก

ทำหน้าที่เชื่อมโยงข้อมูลที่ได้รับจากส่วนโปรแกรมเคลฟไลต์ และทำการตรวจสอบรูปแบบของการแสดงผลจากนั้นจึงทำการส่งที่อยู่ของข้อมูลภาพและค่าเวลาในการแสดงผลไปยังโปรแกรมย่อยต่างๆ ตามรูปแบบการแสดงผลของข้อมูลภาพนั้นๆ

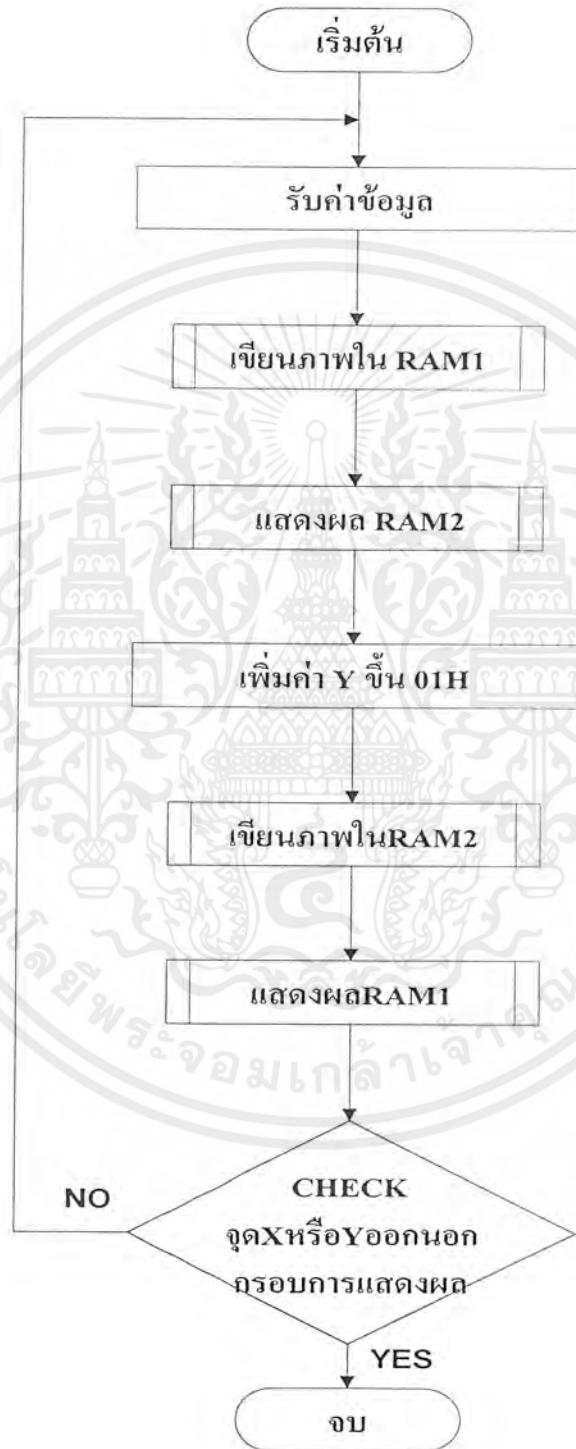


รูปที่ 5.1 แผนภาพแสดงการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 โปรแกรมย่อยสำหรับการเลื่อนภาพขึ้น

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพขึ้นจากด้านล่าง



รูปที่ 5.2 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 โปรแกรมย่อยสำหรับการเลื่อนภาพลง

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพลงจากด้านบน

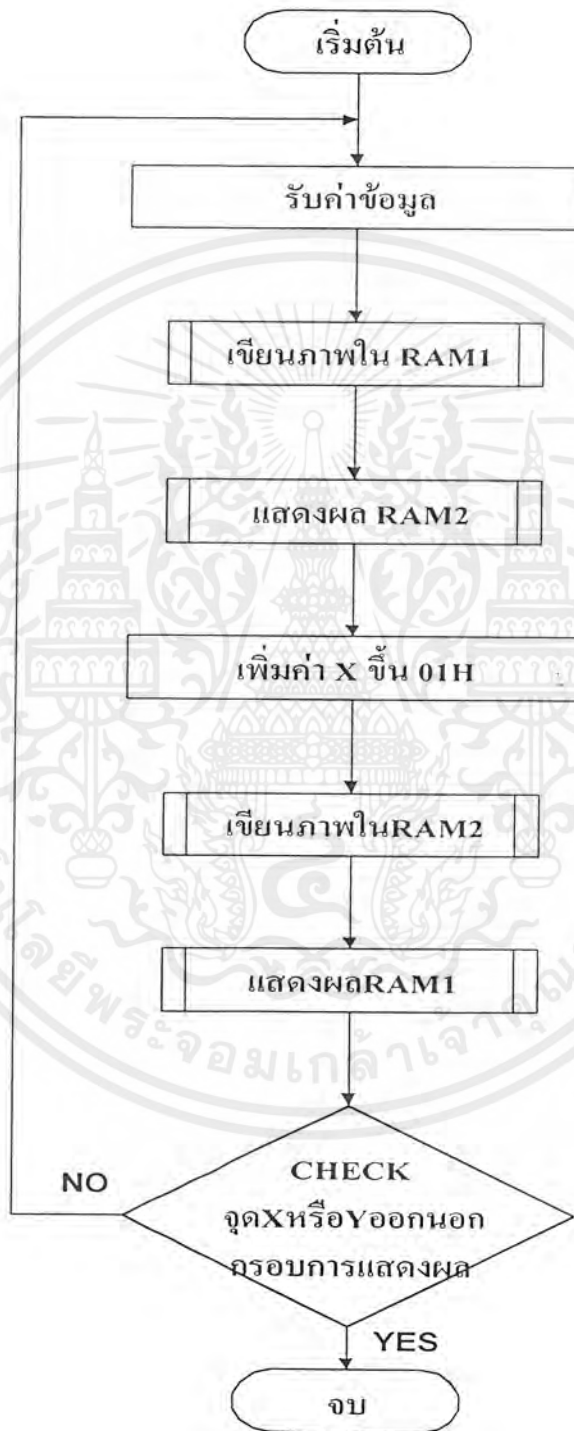


รูปที่ 5.3 แผนภาพแสดงการทำงานของ โปรแกรมเลื่อนภาพลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.4 โปรแกรมย่อยสำหรับการเลื่อนภาพไปทางขวา

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพไปทางขวา

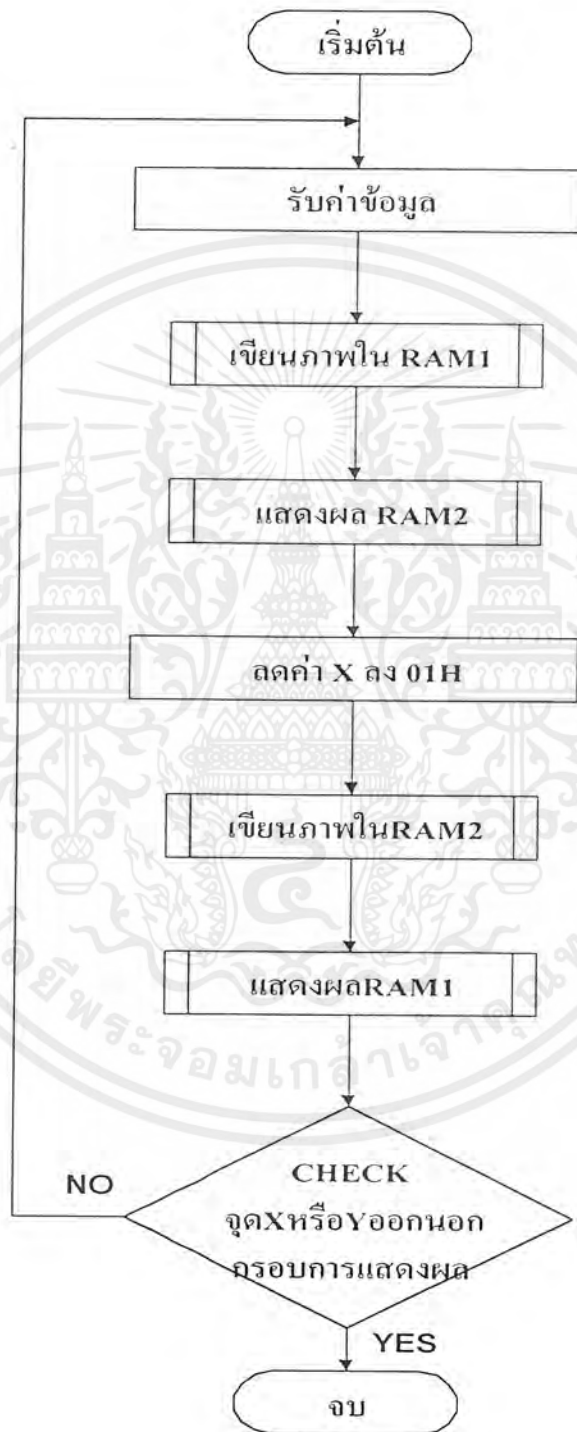


รูปที่ 5.4 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพไปทางขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.5 โปรแกรมย่อยสำหรับการเลื่อนภาพไปทางซ้าย

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับ ให้แสดงผลในลักษณะเลื่อนภาพไปทางซ้าย

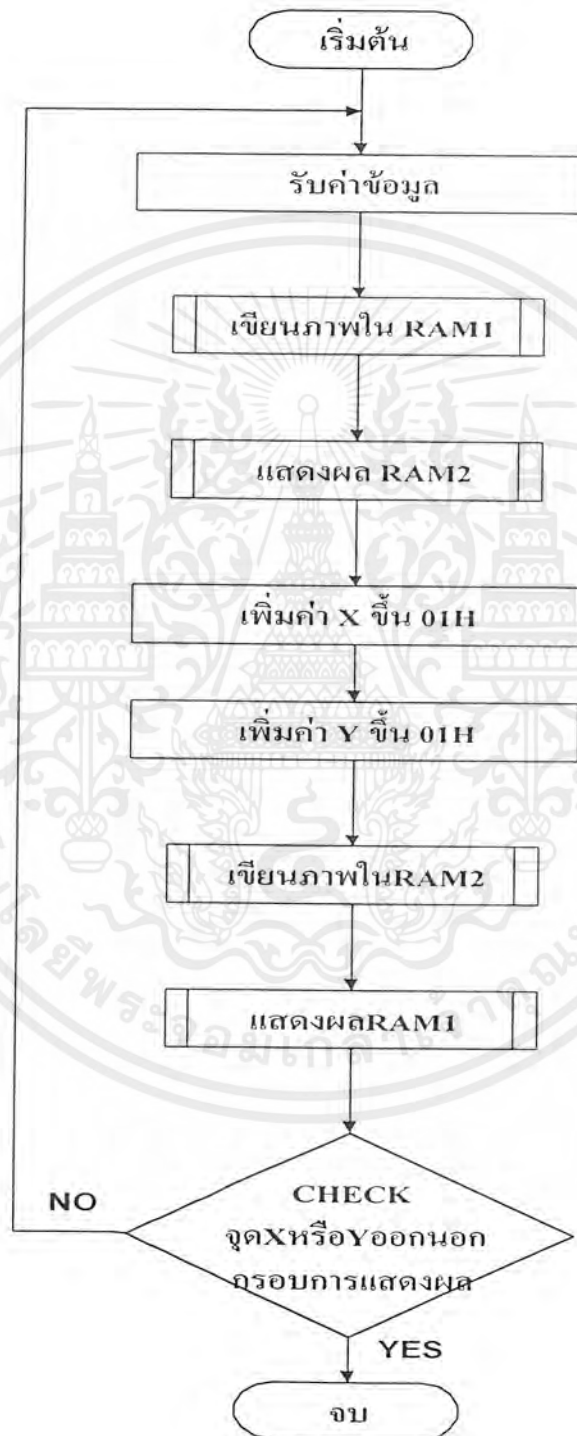


รูปที่ 5.5 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพไปทางซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.6 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงขึ้นขวา

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพเฉียงขึ้นขวา

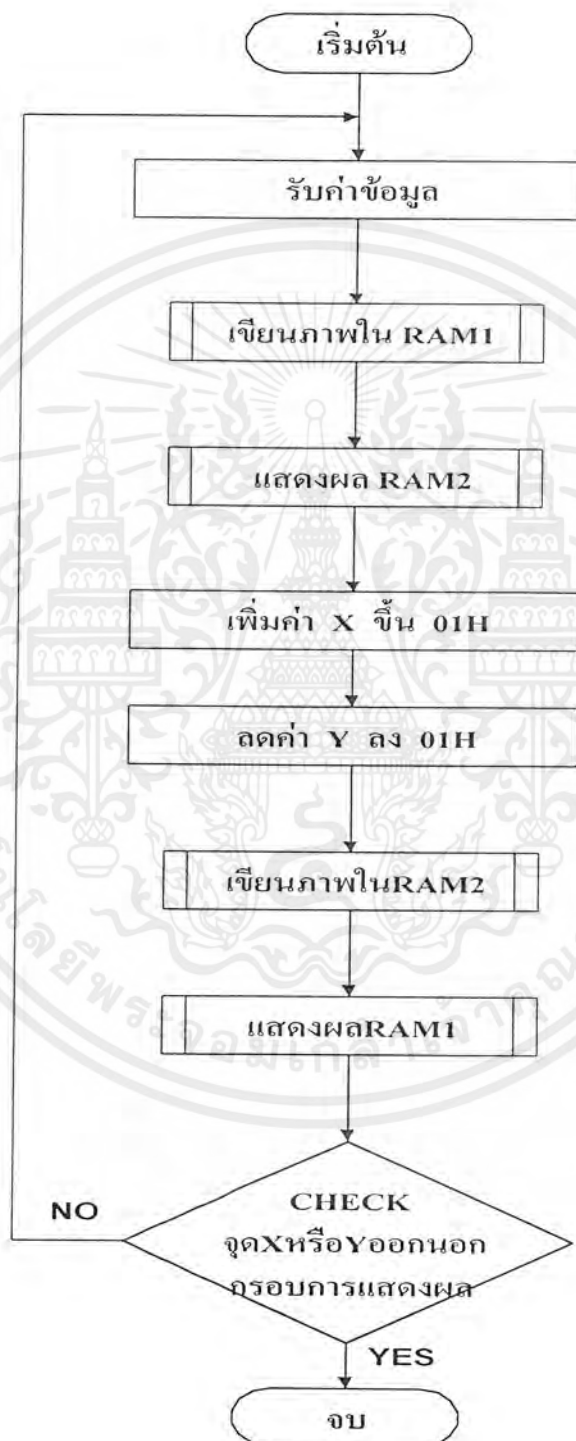


รูปที่ 5.6 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงขึ้นขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.7 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงลงขวา

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพเฉียงลงขวา

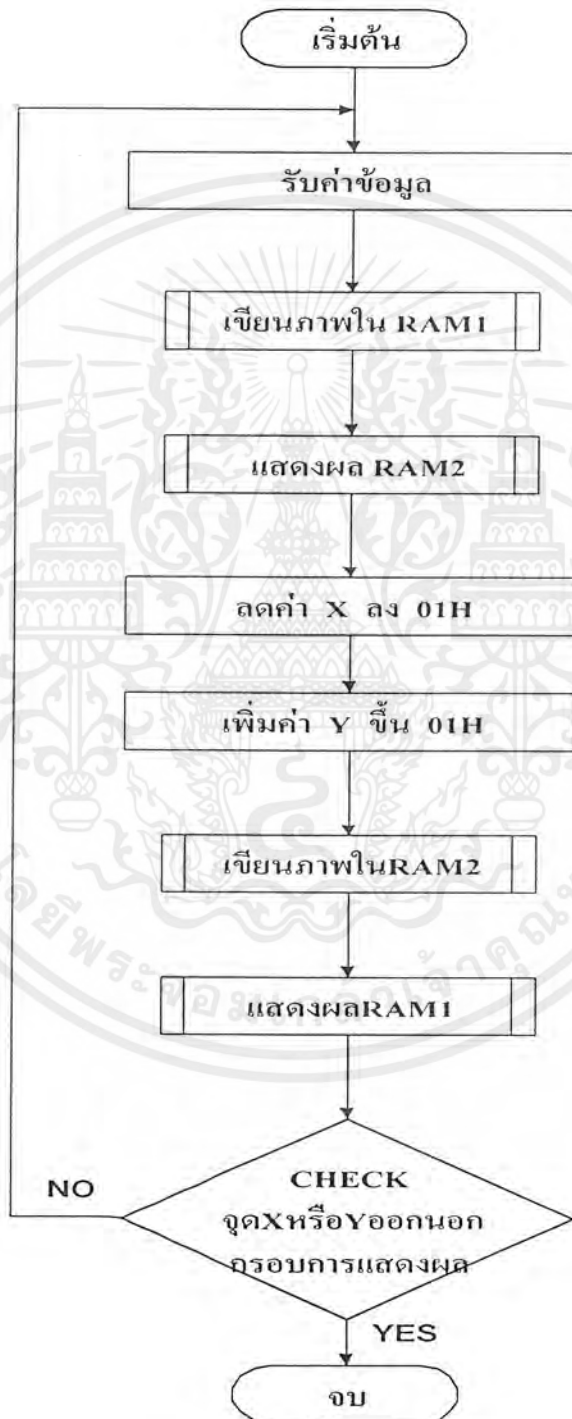


รูปที่ 5.7 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงลงขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.8 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงขึ้นซ้าย

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพเฉียงขึ้นซ้าย

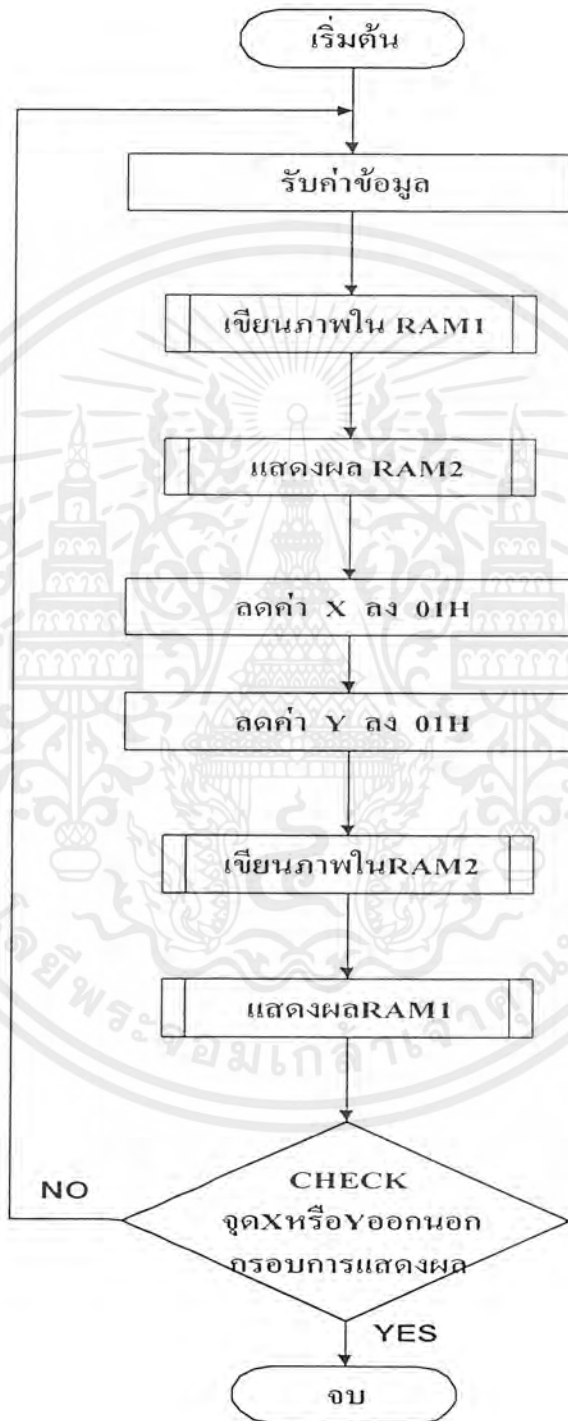


รูปที่ 5.8 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงขึ้นซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.9 โปรแกรมย่อยสำหรับการเลื่อนภาพเฉียงลงซ้าย

ทำหน้าที่ประมวลผลข้อมูลภาพที่ได้รับให้แสดงผลในลักษณะเลื่อนภาพเฉียงลงซ้าย

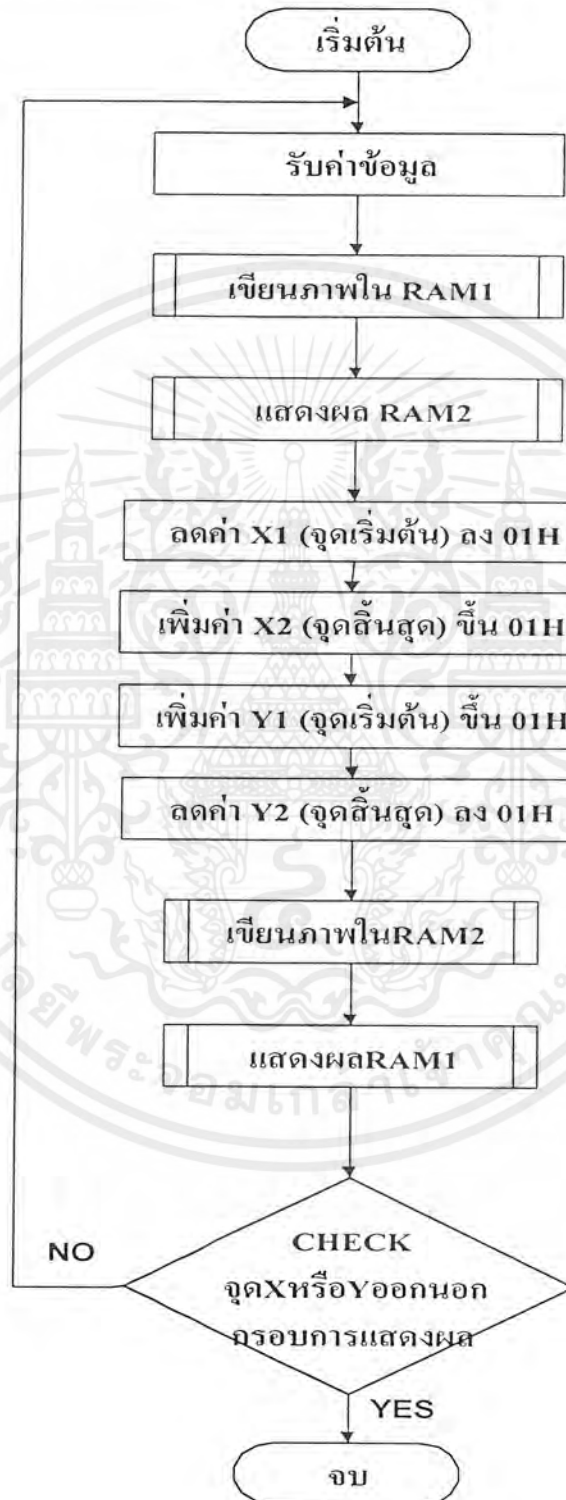


รูปที่ 5.9 แผนภาพแสดงการทำงานของโปรแกรมเลื่อนภาพเฉียงลงซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.10 โปรแกรมย่อยสำหรับการเพิ่มขนาดภาพ

ทำหน้าที่ขยายขนาดของภาพให้มีขนาดเพิ่มขึ้นเรื่อยๆ ตามจำนวนครั้งที่กำหนด

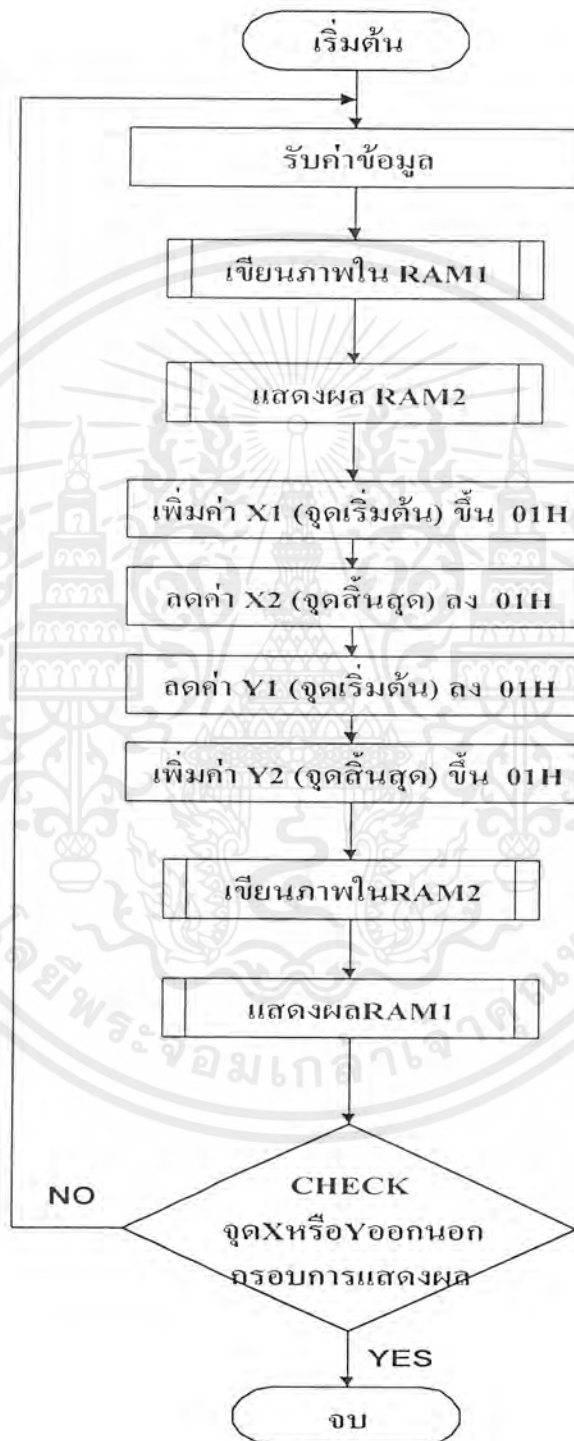


รูปที่ 5.10 แผนภาพแสดงการทำงานของโปรแกรมการเพิ่มขนาดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.11 โปรแกรมย่อยสำหรับการลดขนาดภาพ

ทำหน้าที่ลดขนาดของภาพให้มีขนาดลดลงเรื่อยๆ ตามจำนวนครั้งที่กำหนด

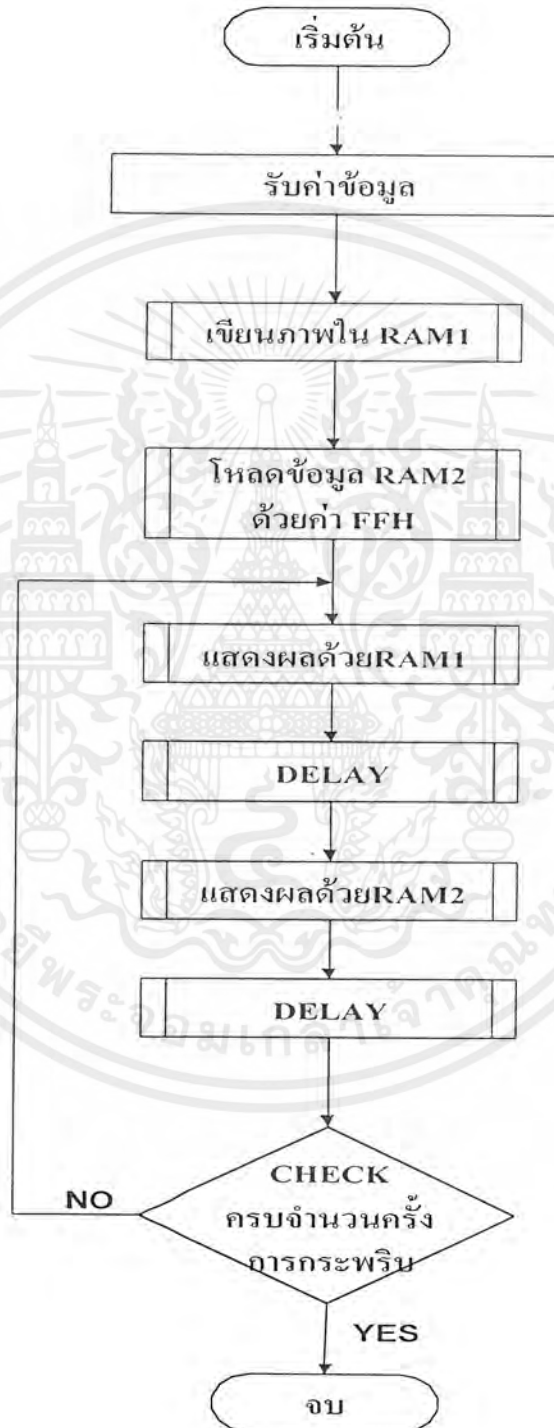


รูปที่ 5.11 แผนภาพแสดงการทำงานของโปรแกรมการลดขนาดภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.12 โปรแกรมย่อยสำหรับการแสดงภาพกระพริบ

ทำหน้าที่แสดงข้อมูลภาพที่ได้รับเป็นระยะเวลาที่ได้กำหนดสลับกับไม่แสดงผลตามจำนวนครั้งที่กำหนด

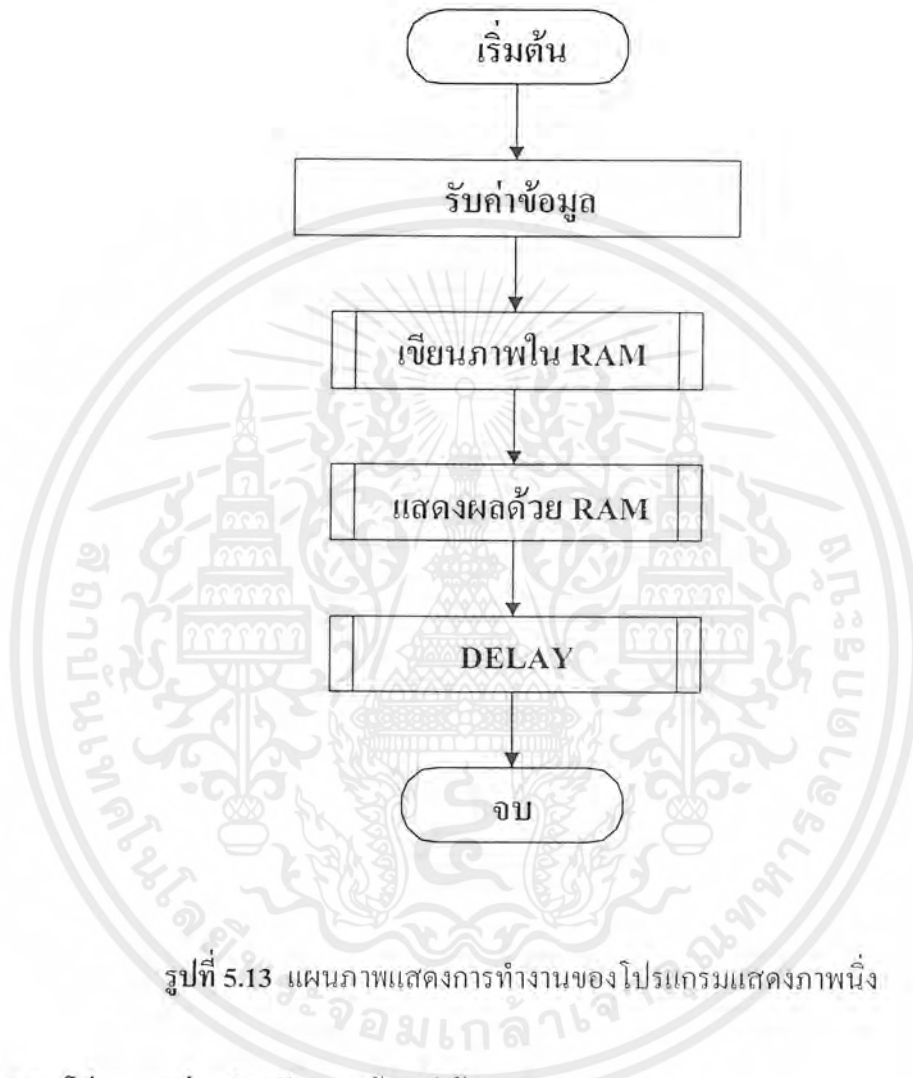


รูปที่ 5.12 แผนภาพแสดงการทำงานของโปรแกรมแสดงภาพกระพริบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.13 โปรแกรมย่อยสำหรับการแสดงภาพนิ่ง

ทำหน้าที่แสดงข้อมูลภาพที่ได้รับเป็นระยะเวลาที่ได้กำหนดไว้



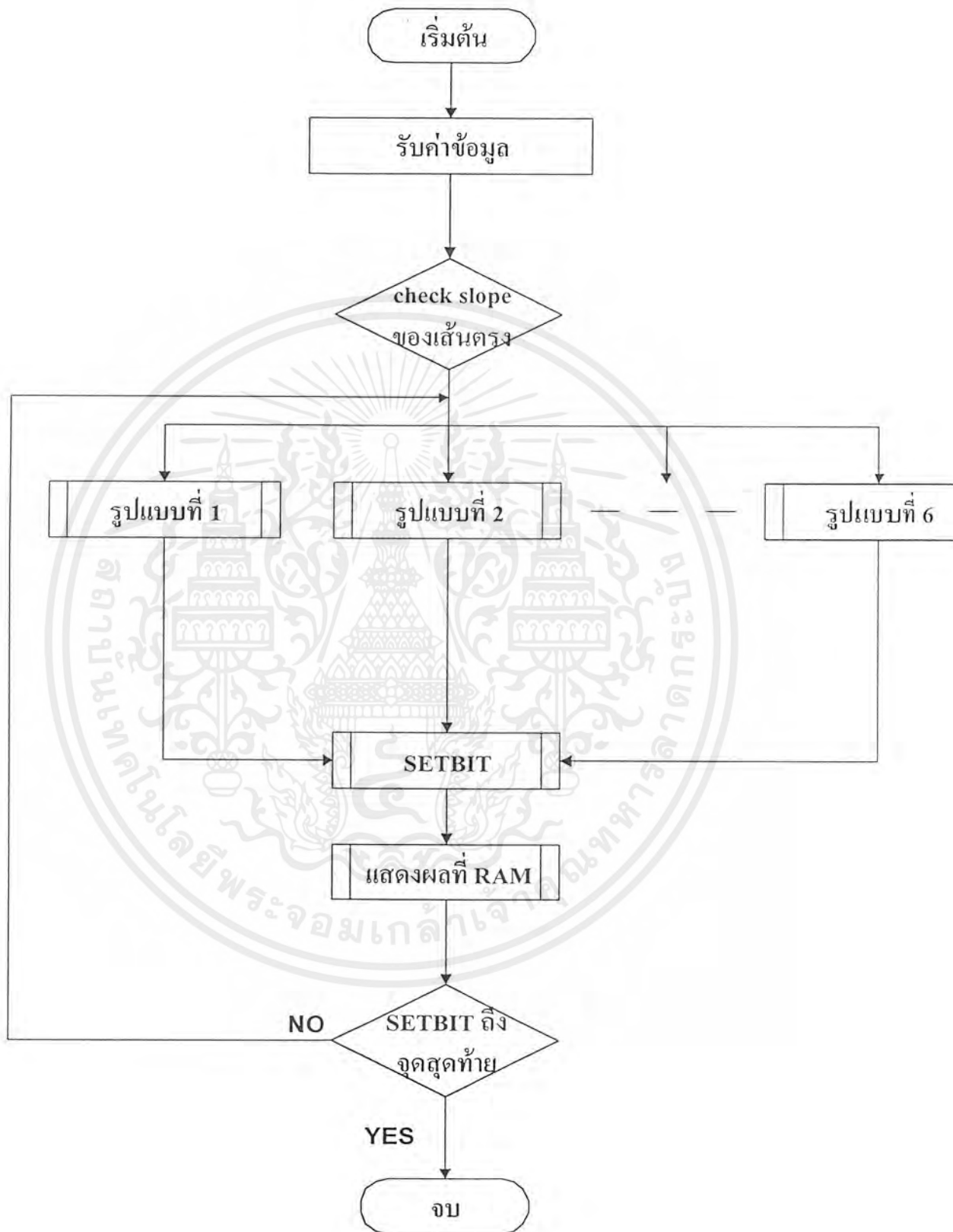
รูปที่ 5.13 แผนภาพแสดงการทำงานของโปรแกรมแสดงภาพนิ่ง

5.1.14 โปรแกรมย่อยสำหรับการสร้างรูปเส้นตรง

ทำหน้าที่ในการสร้างรูปเส้นตรงจากค่าที่กำหนดมาโดยแบ่งเป็นเส้นตรงรูปแบบต่างๆ คือ

- รูปแบบที่ 1 เส้นตรงที่มีความชันเท่ากับ 1
- รูปแบบที่ 2 เส้นตรงที่มีความชันเท่ากับ -1
- รูปแบบที่ 3 เส้นตรงที่มีความชันน้อยกว่า 1 และเท่ากับ 0
- รูปแบบที่ 4 เส้นตรงที่มีความชันมากกว่า 1 และเป็นอนันต์
- รูปแบบที่ 5 เส้นตรงที่มีความชันน้อยกว่า -1
- รูปแบบที่ 6 เส้นตรงที่มีความชันมากกว่า -1

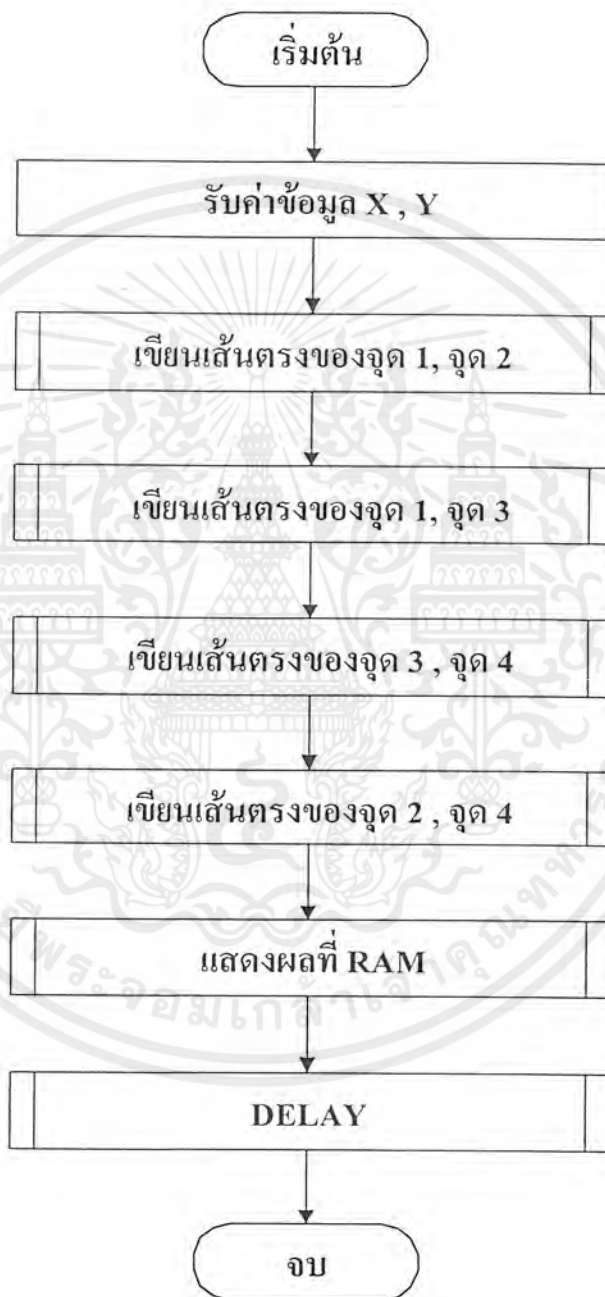
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.14 แผนภาพแสดงการทำงานของโปรแกรมสร้างรูปเส้นตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.15 โปรแกรมย่อยสำหรับการสร้างรูปสี่เหลี่ยม
ทำหน้าที่ในการสร้างรูปสี่เหลี่ยมจากค่าที่กำหนดมา

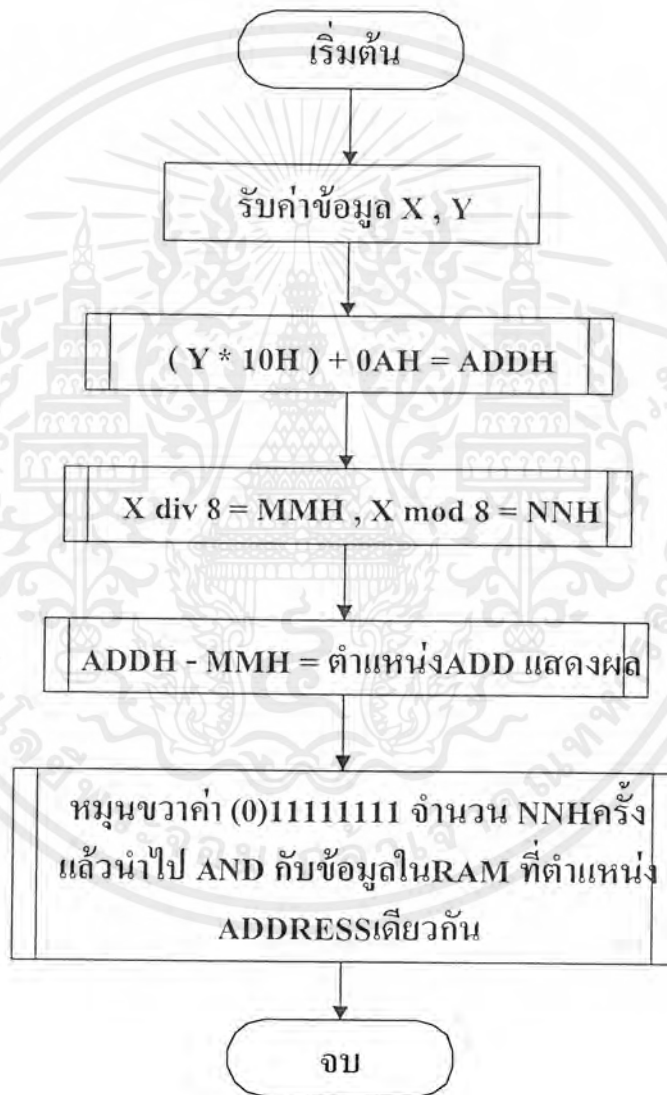


รูปที่ 5.15 แผนภาพแสดงการทำงานของโปรแกรมสร้างรูปสี่เหลี่ยม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.16 โปรแกรมย่อยสำหรับการเช็คค่าในหน่วยความจำ

ทำหน้าที่เช็คค่า 0 หรือ 1 ในหน่วยความจำเพื่อให้เกิดภาพตามที่ไมโครคอนโทรลเลอร์สร้างจากค่าที่ได้รับมาเมื่อนำไปแสดงผล



รูปที่ 5.16 แผนภาพแสดงการทำงานของโปรแกรมเช็คค่าในหน่วยความจำ

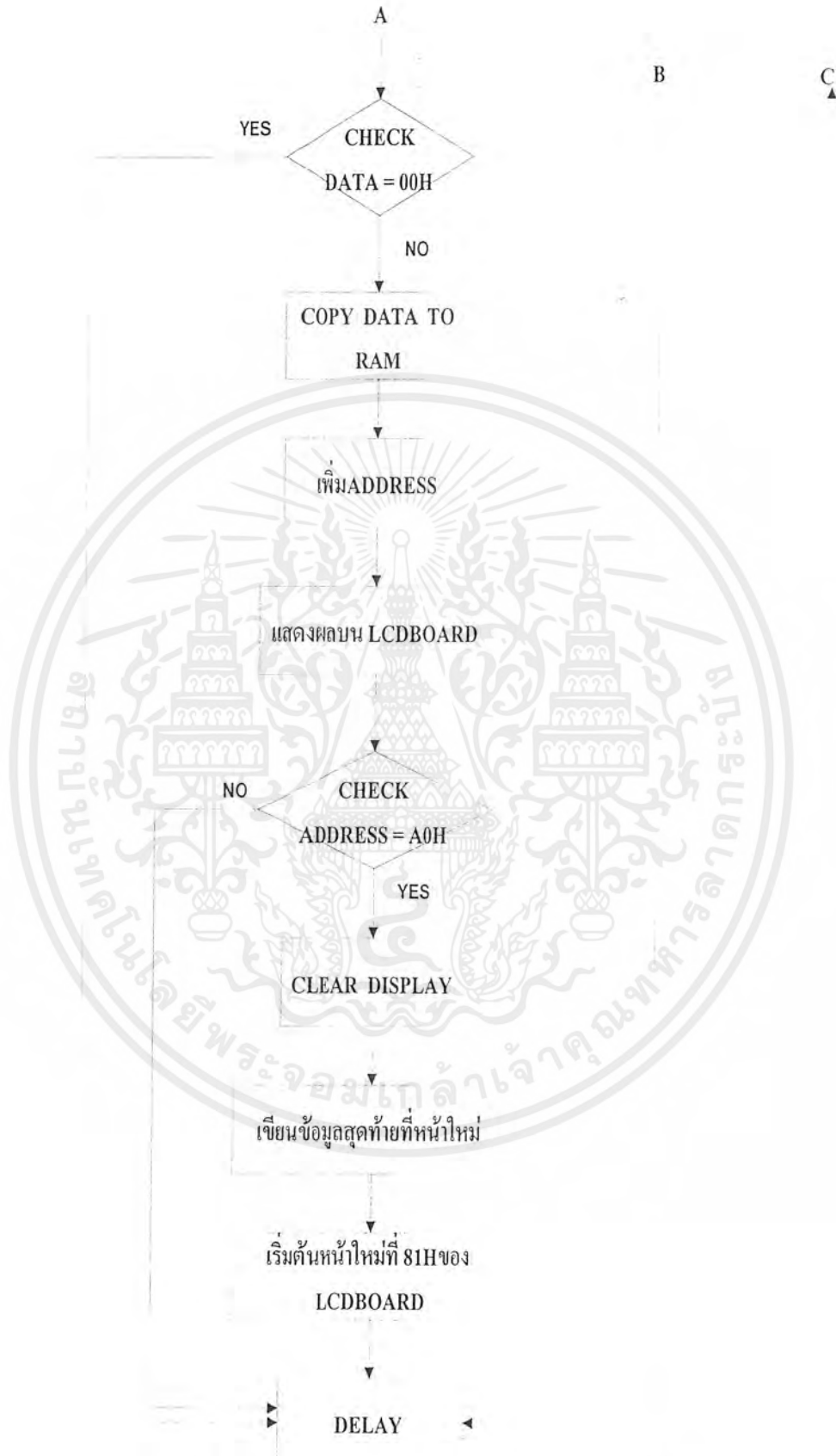
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.17 โปรแกรมย่อยสำหรับการรับค่าจากคีย์บอร์ด

ทำหน้าที่ในการรับค่าข้อมูลตัวอักษรจากคีย์บอร์ดนำมาแสดงบนบอร์ดLCDและบอร์ดแสดงผลตามคีย์ที่กด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

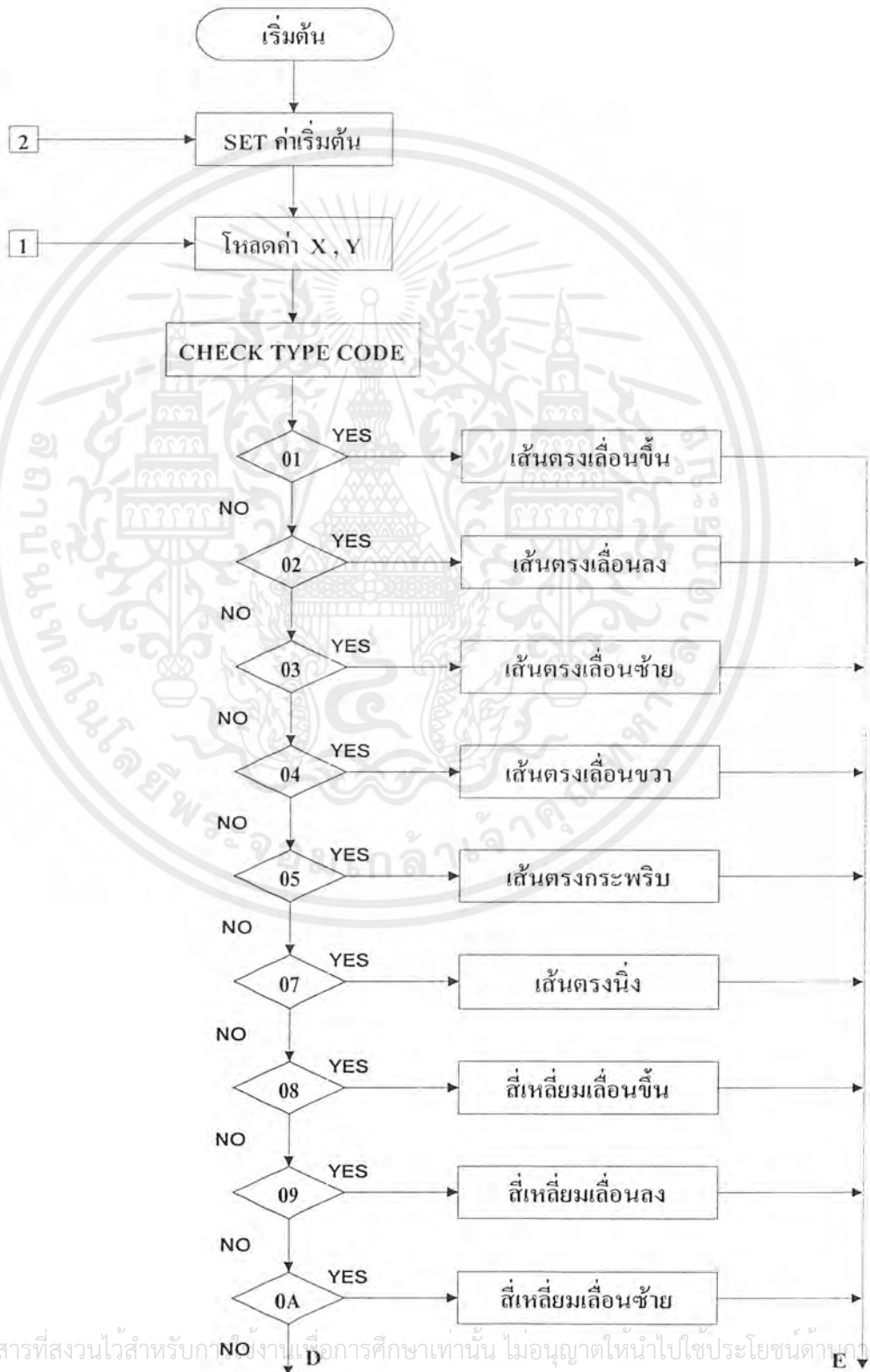


รูปที่ 5.17 แผนภาพแสดงการทำงานของโปรแกรมรับค่าจากคีย์บอร์ด

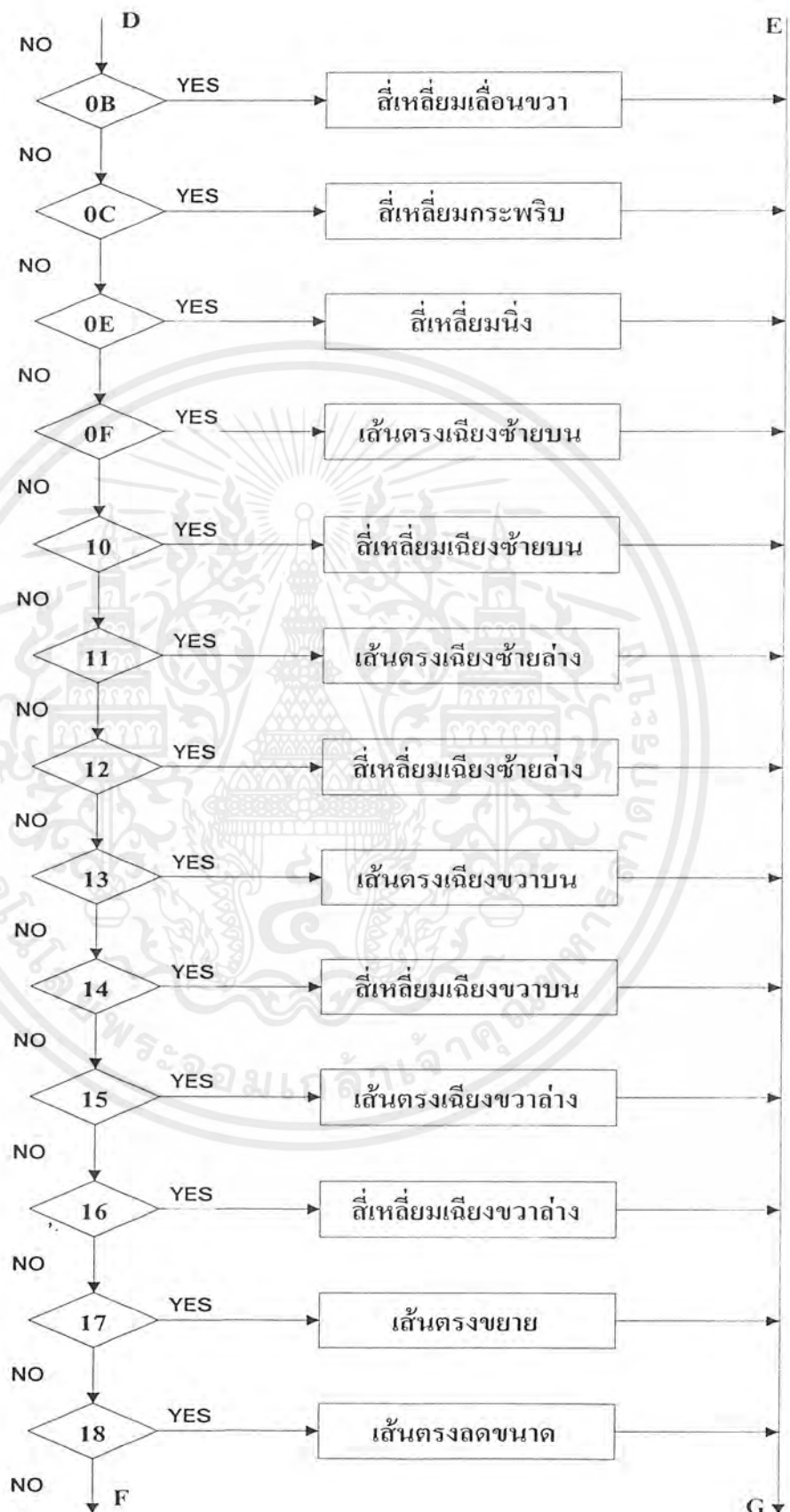
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.18 โปรแกรมย่อยสำหรับการแสดงรูปภาพ

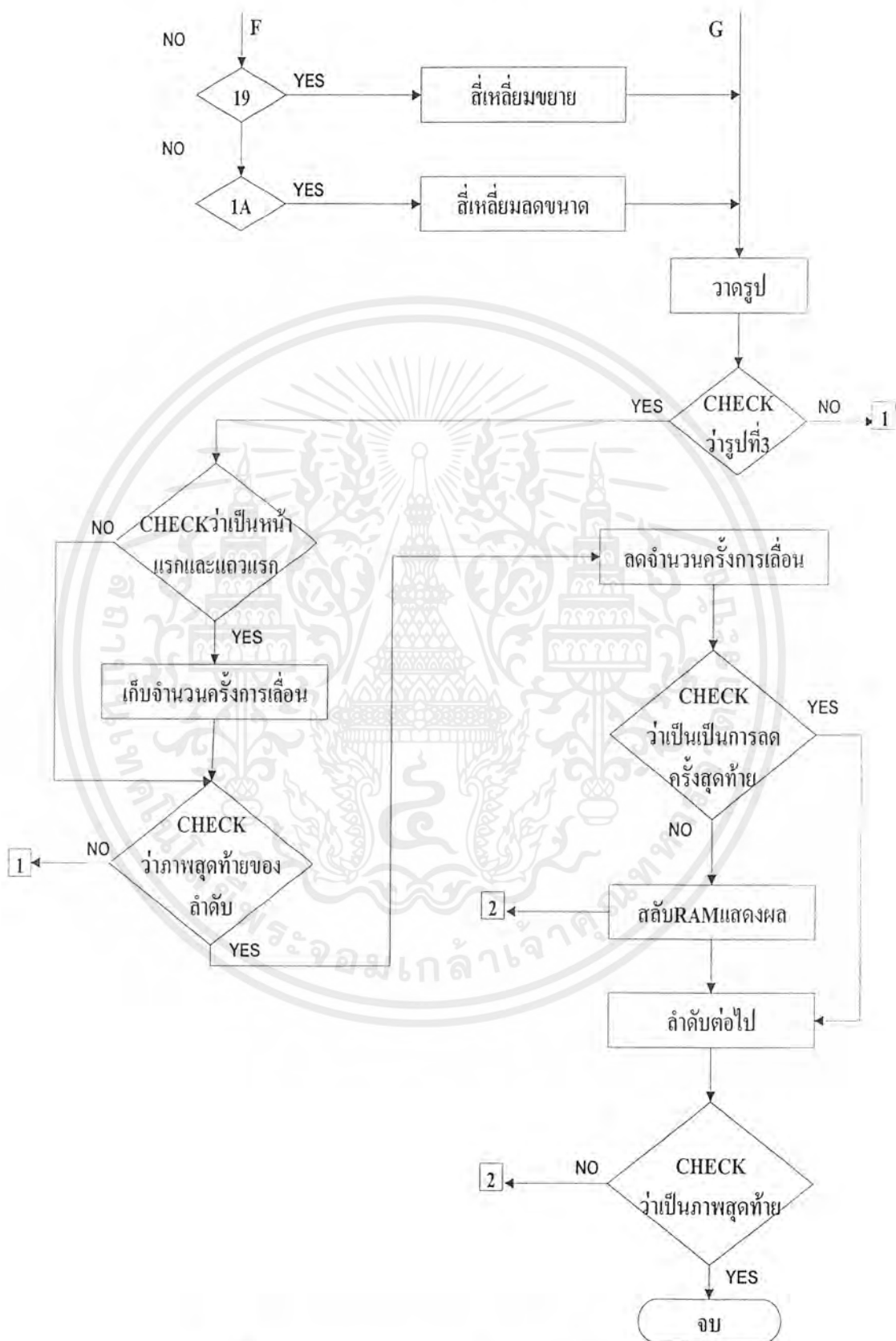
ทำหน้าที่นำข้อมูลภาพมาแสดงบนบอร์ดแสดงผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับงานวิชาการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางธุรกิจ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

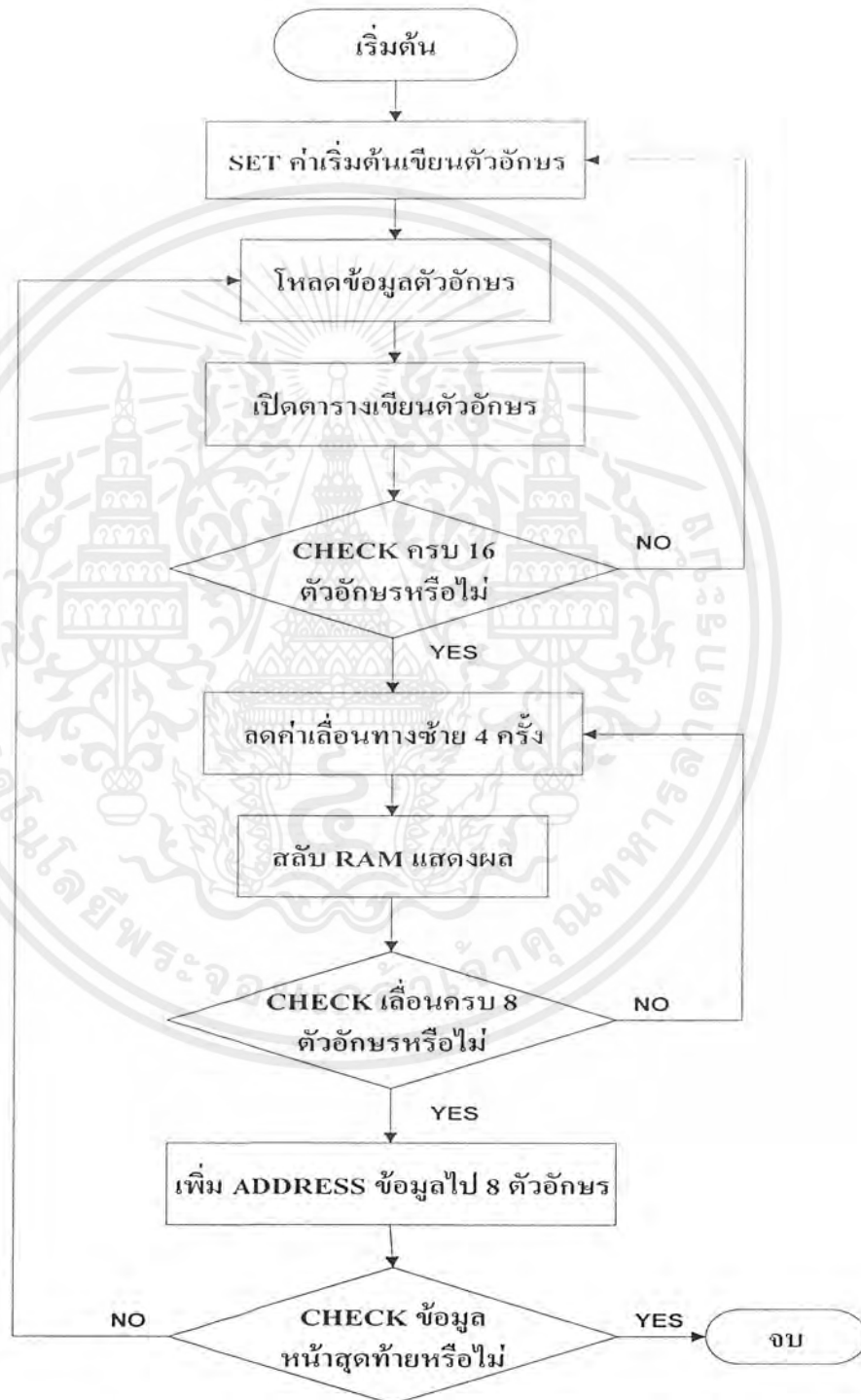


รูปที่ 5.18 แผนภาพแสดงการทำงานของโปรแกรมแสดงรูปภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.19 โปรแกรมย่อยสำหรับการแสดงตัวอักษร

ทำหน้าที่นำข้อมูลตัวอักษรมาแสดงบนบอร์ดแสดงผล



รูปที่ 5.19 แผนภาพแสดงการทำงานของโปรแกรมแสดงตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 โปรแกรมเคลฟไฟล์

ในส่วนของโปรแกรมเคลฟไฟล์จะเป็นส่วนที่ติดต่อกับผู้ใช้โดยตรง โดยเป็นส่วนที่ทำการกำหนดรูปแบบภาพที่ต้องการแสดงผล ซึ่งได้ทำการออกแบบโปรแกรมให้มีลักษณะการใช้งานดังรูปที่ 5.20

จากการออกแบบ สามารถเลือกการแสดงเส้นตรง , สีเหลี่ยม , และตัวอักษรทั้งภาษาไทยและภาษาอังกฤษ โดยการแสดงเส้นตรงและสีเหลี่ยม จะต้องทำการกำหนดข้อมูลของภาพด้วย เช่น ถ้าเป็นเส้นตรงจะกำหนดค่าจุดเริ่มต้นและจุดสิ้นสุด ถ้าเป็นรูปสีเหลี่ยมก็กำหนดค่าจุดที่มุมตรงข้ามกัน 2 มุม ซึ่งจะดูได้จากช่องแสดงตำแหน่ง

ในแต่ละรูปสามารถที่จะเลือกรูปแบบการแสดงผลได้ 12 รูปแบบ ดังนี้

1. เลื่อนขึ้น
2. เลื่อนลง
3. เลื่อนซ้าย
4. เลื่อนขวา
5. เลื่อนเฉียงขึ้นขวา
6. เลื่อนเฉียงลงขวา
7. เลื่อนเฉียงขึ้นซ้าย
8. เลื่อนเฉียงลงซ้าย
9. เพิ่มขยายขนาด
10. ลดขนาด
11. กระจก
12. สว่างค้ำง

ทั้งสามารถกำหนดจำนวนครั้งการกระจกและจำนวนครั้งการเลื่อนได้ จำนวนเส้นตรงและสีเหลี่ยมที่สามารถเขียนได้ในหนึ่งลำดับภาพเท่ากับ 48 เส้น โปรแกรมสามารถทำการโหลดรูปภาพมาเป็นต้นแบบในการวาดภาพเป็นรูปภาพต่างๆ ได้

ในหนึ่งไฟล์ สามารถบรรจุข้อมูลรูปภาพได้ 9 ลำดับ และข้อความทั้งภาษาไทยและภาษาอังกฤษ เมื่อกำหนดค่าต่างๆ ของภาพตามที่เราต้องการแล้ว ก็ทำการ “ตกลง” และทำการบันทึก โปรแกรมจะทำการแปลงข้อมูลภาพ , ข้อความและรูปแบบการแสดงผลเป็นรูปเฮกซ์ไฟล์ (Hex file) เพื่อใช้ในการส่งข้อมูลเข้าสู่ส่วนของไมโครคอนโทรลเลอร์ต่อไป โดยขั้นตอนนี้เราต้องกำหนดชื่อไฟล์ที่ต้องบันทึก การส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อแสดงผลในครั้งเดียวกันสามารถส่งได้สูงสุด 5 ไฟล์ โดยจะแสดงภาพไฟล์ข้อมูลตามลำดับในการส่งไฟล์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการส่งข้อมูลภาพเข้าสู่บอร์ดไมโครคอนโทรลเลอร์ จะใช้วิธีส่งข้อมูลภาพในรูปแบบเฮกซ์ไฟล์ที่ได้จากส่วนโปรแกรมเซลล์ไฟล์ โดยใช้การส่งข้อมูลแบบ RS – 232 เข้าสู่บอร์ดไมโครคอนโทรลเลอร์ ข้อมูลที่ส่งไปนี้ไมโครคอนโทรลเลอร์จะนำมาทำการประมวลผลและแสดงภาพตามที่ได้กำหนดไว้

โปรแกรมเซลล์ไฟล์สามารถแบ่งเป็นส่วนๆ ตามหน้าที่ได้ดังนี้

1. โปรแกรมหลักเซลล์ไฟล์
2. โปรแกรมการจัดลำดับภาพและตัวอักษร
3. โปรแกรมการเลือกรูปแบบการแสดงผล
4. โปรแกรมการเก็บข้อมูลรูปภาพที่วาด
5. โปรแกรมการบันทึกข้อมูลภาพ
6. โปรแกรมการบันทึกข้อมูลตัวอักษร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Form1 _ 6 _ X

เพิ่มข้อมูล

เลื่อนขึ้น เลื่อนซ้าย เที่ยงขึ้นขวา กระพริบ ขยาย สดขนาด
 เลื่อนลง เลื่อนขวา เที่ยงลงขวา แสดงผล

เส้นตรง

สีเทาเข้ม

5.27

ตกลง

สำคัญมาก
 สำคัญสูง
 สำคัญสาม
 สำคัญสี่
 สำคัญห้า
 สำคัญหก
 สำคัญเจ็ด
 สำคัญแปด
 สำคัญเก้า
 ตัวอักษร

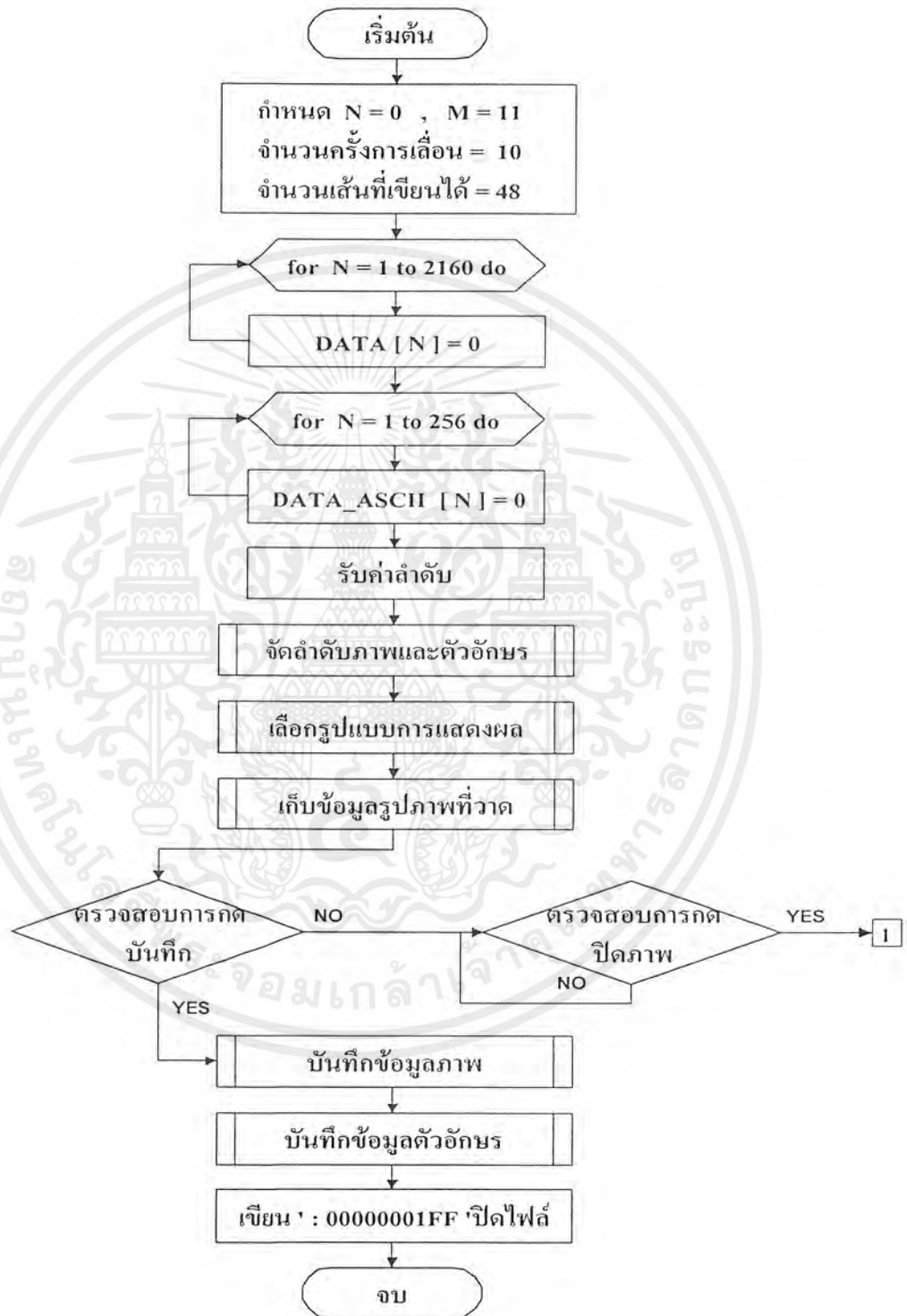
จำนวนเส้นที่เขียนได้ 48

จำนวนครั้งที่เขียน 10

รูปที่ 5.20 รูปแสดงการใช้งานโปรแกรมเดสท็อป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

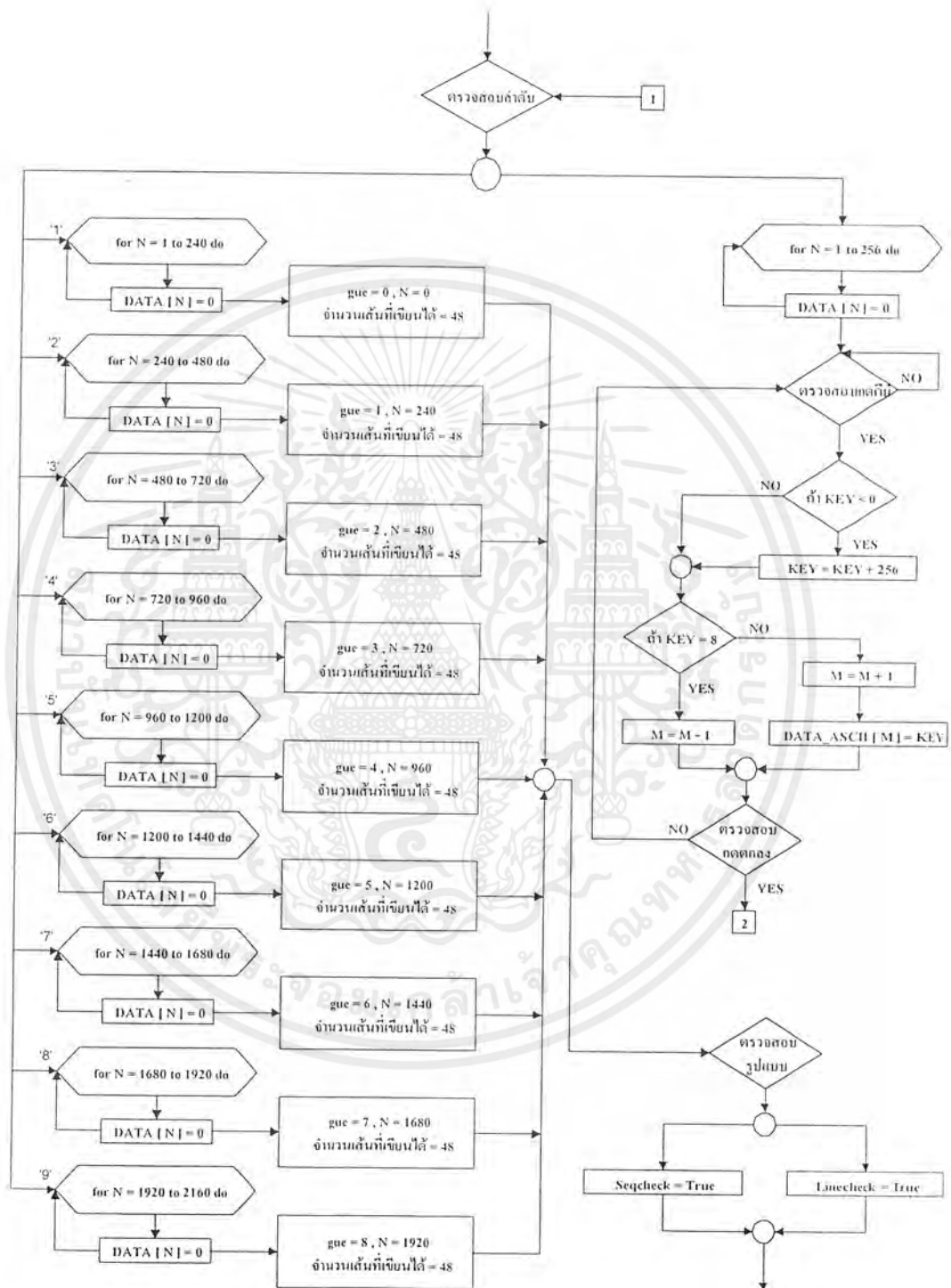
5.2.1 โปรแกรมหลักเคลไฟล์



รูปที่ 5.21 แผนภาพแสดงการทำงานของโปรแกรมหลักเคลไฟล์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

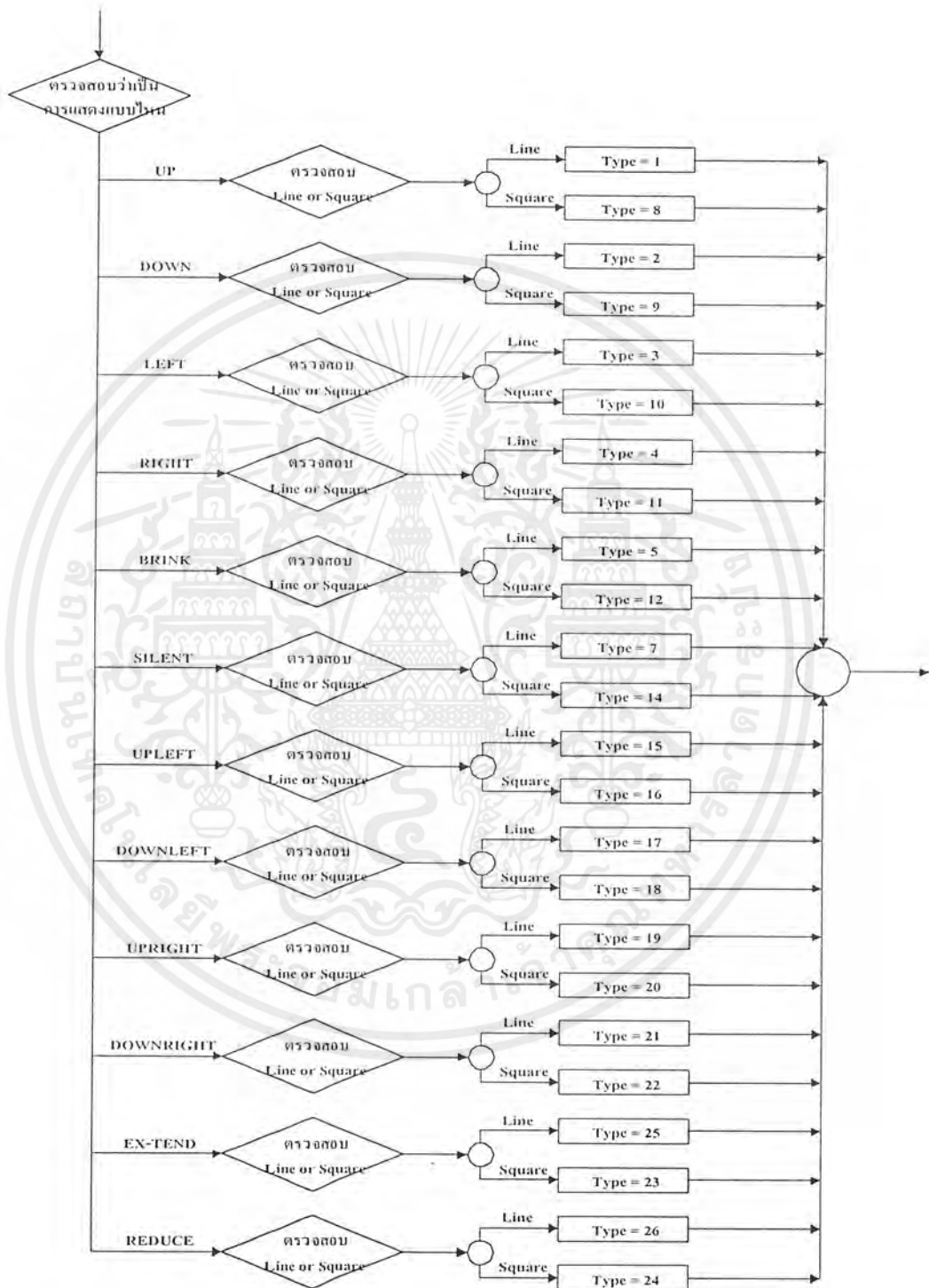
5.2.2 โปรแกรมการจัดลำดับภาพและตัวอักษร



รูปที่ 5.22 แผนภาพแสดงการทำงานของโปรแกรมจัดลำดับภาพและตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

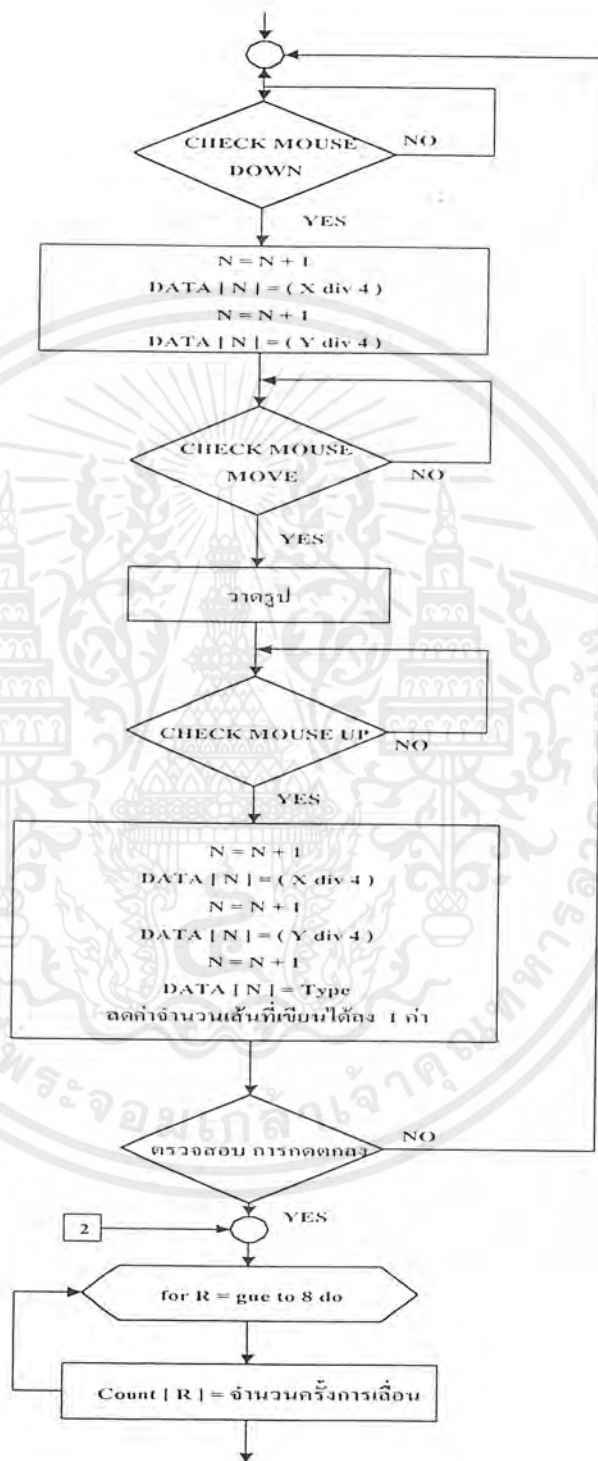
5.2.3 โปรแกรมการเลือกรูปแบบการแสดงผล



รูปที่ 5.23 แผนภาพแสดงการทำงานของโปรแกรมเลือกรูปแบบการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

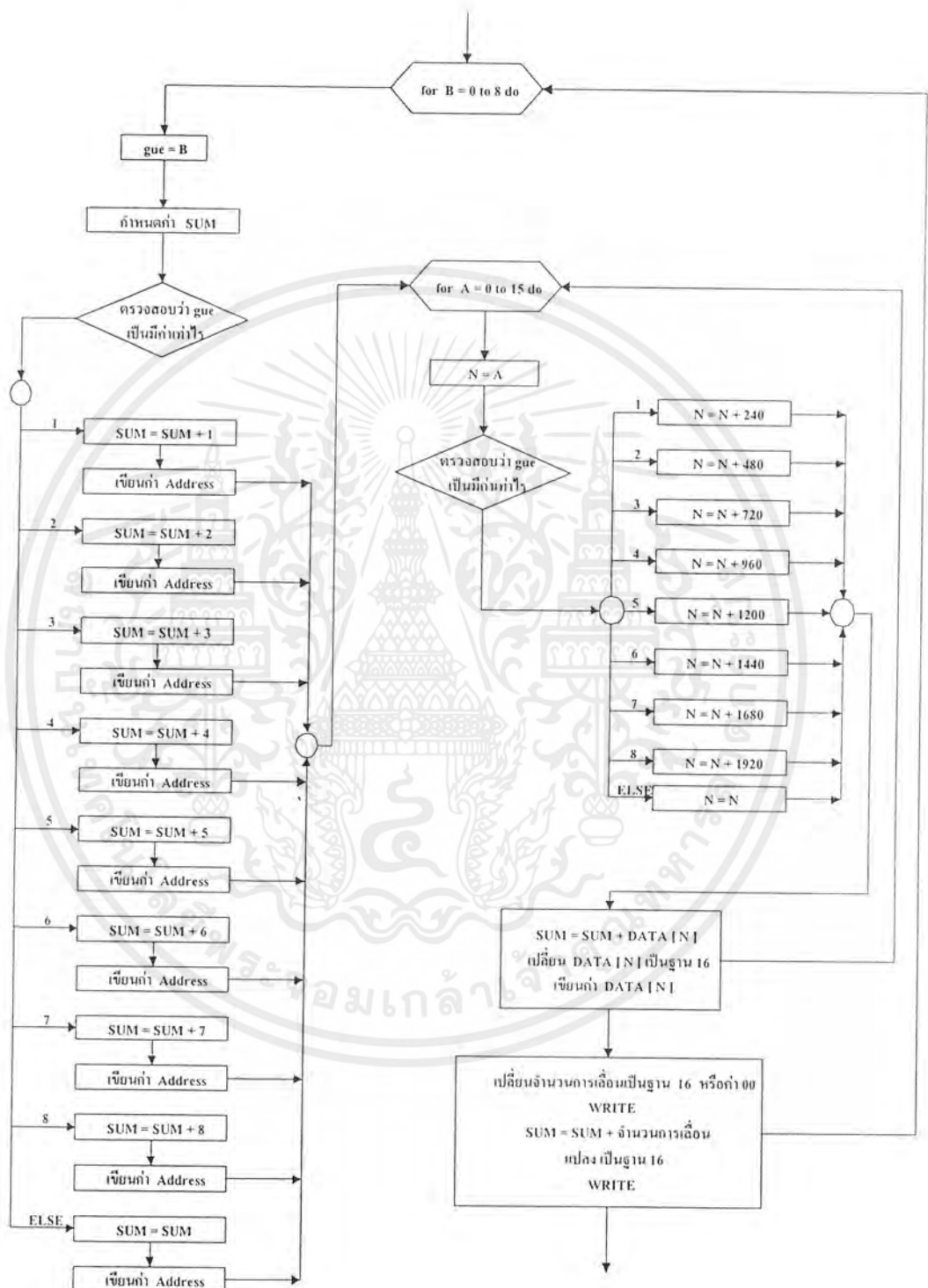
5.2.4 โปรแกรมการเก็บข้อมูลรูปภาพที่วาด



รูปที่ 5.24 แผนภาพแสดงการทำงานของโปรแกรมเก็บค่าข้อมูลรูปภาพที่วาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

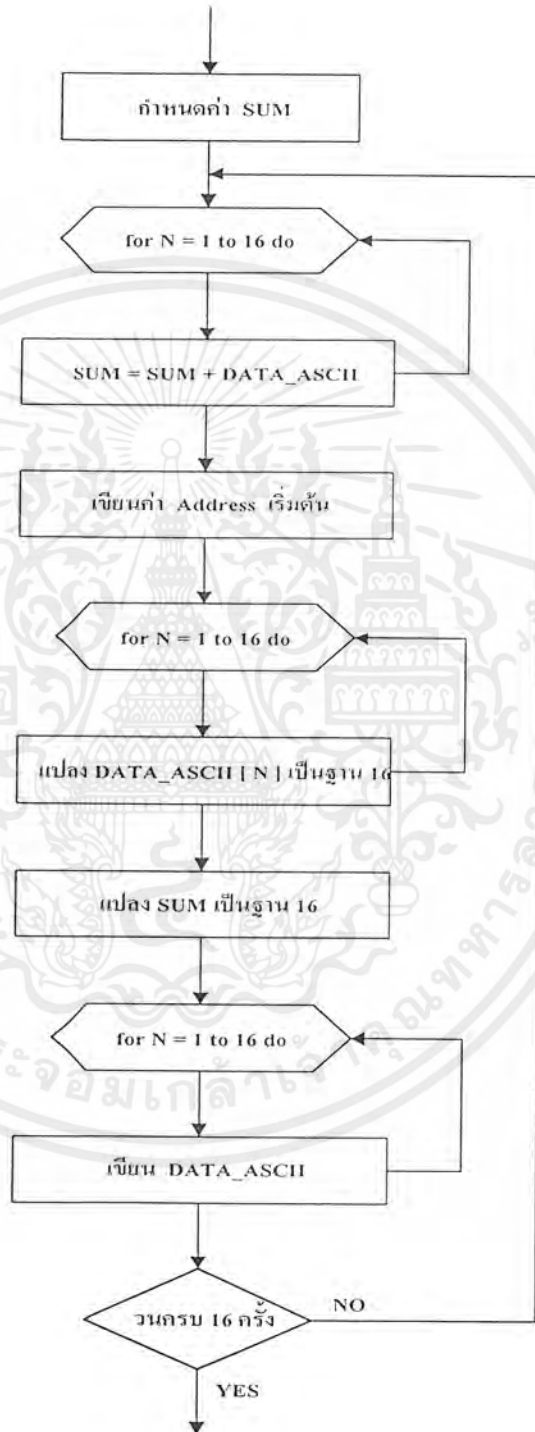
5.2.5 โปรแกรมการบันทึกข้อมูลภาพ



รูปที่ 5.25 แผนภาพแสดงการทำงานของโปรแกรมบันทึกข้อมูลภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.6 โปรแกรมการบันทึกข้อมูลตัวอักษร



รูปที่ 5.26 แผนภาพแสดงการทำงานของโปรแกรมบันทึกข้อมูลตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.7 รูปแบบข้อมูลเฮกซ์ไฟล์ (Hex File) จากโปรแกรมเดลไฟล์

รูปแบบข้อมูลเฮกซ์ไฟล์ที่ได้จากโปรแกรมเดลไฟล์จัดเป็นมาตรฐานแบบ อินเทลเฮกซ์ไฟล์ (Intel Hex File) ที่นิยมใช้กันมาก สังเกตได้จากโปรแกรม COMPILER หรือ ASSEMBLER ต่างๆ มักจะกำหนดให้เลือกเอาที่พูด ที่ต้องการเป็นแบบอินเทลเฮกซ์ไฟล์ได้ ทั้งนี้เนื่องจากอินเทลเฮกซ์ไฟล์ มีรูปแบบที่เหมาะสมหลายประการ คือ มีระบบเช็คซั่ม (Check Sum) ซึ่งสามารถตรวจสอบความถูกต้องของข้อมูลได้ในแต่ละบรรทัด รวมทั้งมีการกำหนดแอดเดรส ของข้อมูลได้ และที่สำคัญคือเป็น ไฟล์แบบ ASCII ก็สามารถใช้ในการส่งออกทางพอร์ตสื่อสารต่างๆ ได้ และยังสามารถใช้กับโปรแกรม EDITOR ทั่วๆ ไปเพื่อการแก้ไขได้ (ไฟล์ที่จะใช้ส่งออกทางพอร์ตสื่อสาร จะต้องเป็นไฟล์แบบ ASCII เท่านั้น ทั้งนี้เนื่องจากการสื่อสาร จำเป็นจะต้องมีรหัสเพื่อการสื่อสารเองอยู่แล้วจึงไม่สามารถใช้ไฟล์แบบ OBJECT ได้ เพราะข้อมูลอาจจะไปเข้ากับรหัสของการสื่อสารได้) รายละเอียดของอินเทลเฮกซ์ไฟล์ ในแต่ละบรรทัดเป็นดังนี้

:BCAAAATTHH.....HHCC

:	คือ Start Character (Colon)
BC	คือ จำนวนไบต์ ของข้อมูลในบรรทัด มีค่าเป็นเลขฐาน 16 (Hex) ถ้า BC = 0 จะเป็น End - Of - File Record
AAAA	คือ แอดเดรส ของข้อมูลในไบต์แรก
TT	คือ ชนิดของข้อมูลในบรรทัดนั้นๆ ถ้า TT = 0 เป็น Data Record ถ้า TT = 1 เป็น End - Of - File Record
HH	คือ ข้อมูลแต่ละไบต์
CC	คือ ค่าเช็คซั่ม ของบรรทัดนั้นๆ โดยจะเป็นค่า Two's complement ของผลบวกของข้อมูลทุกๆ ไบต์ ในบรรทัด ซึ่งรวมทั้งค่า BC , AAAA และ TT ด้วย

บทที่ 6

ผลการทดลอง

ผลการทดลองจะมีในส่วนของซอฟต์แวร์ สามารถแบ่งออกได้เป็น 3 ส่วน คือ

6.1 ส่วนโปรแกรมไมโครคอนโทรลเลอร์

โปรแกรมไมโครคอนโทรลเลอร์ จะแบ่งเป็นโปรแกรมหลักและโปรแกรมย่อยต่างๆ เราจึงสามารถทำการทดลองที่แต่ละส่วนได้ ทำให้ง่ายต่อการตรวจเช็คและจากการทดลองพบว่าในส่วนโปรแกรมหลักซึ่งทำหน้าที่ตรวจว่าจะรับข้อมูลจากส่วนโปรแกรมเซลล์ไฟล์ (คอมพิวเตอร์) พร้อมรับข้อความจากคีย์บอร์ด ทำการตรวจสอบรูปแบบเพื่อนำไปประมวลผลแล้วส่งไปยังบอร์ดควบคุมการแสดงผลเพื่อทำการแสดงนั้นๆ ที่บอร์ดแสดงผล LED ก็พบว่าสามารถทำงานได้จริง

ส่วนโปรแกรมย่อยต่างๆ ที่ทำหน้าที่ประมวลรูปแบบของการแสดงผลแบบต่างๆ สามารถทำการแสดงผลในรูปแบบต่างๆ ที่กำหนดได้

6.2 ส่วนโปรแกรมเซลล์ไฟล์

ส่วนโปรแกรมเซลล์ไฟล์สามารถทำการโหลดข้อมูลรูปภาพที่ต้องการให้แสดงผลและเลือกรูปแบบการแสดงผลได้ โดยจากการออกแบบได้กำหนดให้สามารถทำการกำหนดรูปแบบรูปภาพและขนาดจากการป้อนค่า ซึ่งมีรูปเส้นตรง , สีเหลี่ยม , และออกแบบให้เลือกรูปแบบการแสดงผลได้ปรากฏว่าโปรแกรมสามารถทำการแปลงข้อมูลภาพและรูปแบบการแสดงผลแบบต่างๆ เป็นรูปแบบเฮกซ์ไฟล์ได้อย่างถูกต้อง สำหรับการตรวจสอบการแปลงข้อมูล ได้ทำการตรวจสอบจากรูปแบบเฮกซ์ไฟล์ที่ทำการแปลงแล้วว่าถูกต้องหรือไม่ โดยหาความสัมพันธ์ของรูปภาพ , รูปแบบภาพ และรูปแบบเฮกซ์ไฟล์ที่ได้

6.3 ส่วนคีย์บอร์ด

สามารถรับข้อมูลโดยตรงจากคีย์บอร์ด ให้แสดงผลที่บอร์ดLEDได้ โดยไม่ต้องพิมพ์ข้อมูลผ่านคอมพิวเตอร์ จากการที่ได้ออกแบบไว้เมื่อพิมพ์ข้อมูลที่คีย์บอร์ดข้อมูลจะไปแสดงที่บอร์ดLCD และที่บอร์ดLED

บทที่ 7

สรุปผลและวิจารณ์ผลการทดลอง

7.1 สรุปผลและวิจารณ์ผลการทดลอง

จากการทดลองทำการแสดงผลด้วยบอร์ด LED พบว่าสามารถแสดงผลได้ตามที่ต้องการ ความสว่างที่ได้ค่อนข้างสว่างแต่ภาพที่ได้มีการสั่นพริ้วเล็กน้อย คาดว่าเกิดจากความถี่ที่ใช้ต่ำเกินไป และ LED แต่ละดวงจะสว่างไม่เท่ากันเกิดจาก LED ที่ใช้มีคุณภาพไม่ดีมากนักทำให้มีศักดาตกคร่อม LED แต่ละดวงไม่เท่ากันเป็นเหตุให้กระแสที่จ่ายให้ LED แต่ละดวงไม่เท่ากัน และการขับกระแสไม่ได้ใช้เป็น Current Mode

LED หลอดที่ใกล้เคียงกับหลอดที่สว่างอยู่จะมีความสว่างเกิดขึ้นแต่น้อยมาก เป็นเพราะช่วงเวลาในการสแกนทาง ไร้วีและคอลัมน์ไม่เท่ากันและภาพที่ได้จากกำหนดค่าบางค่าจะทำให้รูปที่ได้เกิดการบิดเบี้ยวไป ซึ่งเกิดจากความละเอียดของบอร์ดไม่มากพอและโปรแกรมที่ใช้ยังใช้อัลกอริทึมที่ไม่ละเอียดในการสร้างภาพทำให้มีความละเอียดไม่พอ ในบางครั้งมีการเสียหายของ LED เกิดขึ้นในระหว่างการทดลอง แต่เนื่องจากบอร์ดมีขนาดจำกัดทำให้การสับเปลี่ยน LED ตัวใหม่เป็นไปอย่างยากลำบากเพราะจะทำความเสียหายให้กับ LED ตัวข้างเคียง

เนื่องจากบอร์ดแสดงผล ไม่ได้สร้างขึ้นมาเอง ทำให้ในบางครั้งเมื่อมีปัญหาที่ตัวบอร์ดก็จะหาข้อผิดพลาดและแก้ไขได้ยากขณะทำการทดลอง ถ้าใช้บอร์ดแสดงผลนานๆ บอร์ดมักจะเกิดการรวนอยู่บ่อยๆ ไม่สามารถหาสาเหตุได้ จะใช้การพักบอร์ดแสดงผลโดยปิดสวิทซ์ไฟ ทำให้การทำงานไม่ราบรื่นการทำงานไปได้ช้า


การดำเนินงานในส่วนของซอฟต์แวร์ โดยแบ่งเป็น โปรแกรมไมโครคอนโทรลเลอร์ซึ่งทำหน้าที่รับข้อมูลต่างๆ จากคอมพิวเตอร์ แล้วทำการประมวลผลเพื่อสร้างภาพ, ตัวอักษรและรูปแบบการแสดงผล โดยมีการแสดงผลได้หลายแบบและสามารถจัดลำดับข้อมูลเป็นหลายๆ หน้าสำหรับสลับการแสดงผลเพื่อแสดงเป็นภาพเคลื่อนไหวได้

โปรแกรมเคลไฟล์ ใช้ในการติดต่อสื่อสารระหว่างไมโครคอนโทรลเลอร์และคอมพิวเตอร์ โดยจะทำหน้าที่รับข้อมูลต่างๆ จากผู้ใช้ เช่น รูปภาพ, ตัวอักษร, รูปแบบ และลำดับการแสดงผลและส่งข้อมูลเหล่านี้ไปยังบอร์ดควบคุมการแสดงผลผ่าน RS-232 โดยสามารถส่งไฟล์ได้สูงสุด 5 ไฟล์ และแสดงผลตามลำดับการส่ง

การดำเนินงานส่วนฮาร์ดแวร์ ได้สร้างวงจรรับข้อมูลจากคีย์บอร์ดเพื่อแสดงผลบนบอร์ด โดยตรงไม่ต้องส่งข้อมูลผ่านคอมพิวเตอร์โดยการเปลี่ยนสัญญาณข้อมูลแบบอนุกรมเป็นสัญญาณข้อมูลแบบขนาน แล้วให้ไมโครคอนโทรลเลอร์ทำการอ่านค่ารหัสที่ได้ (Scan Key - Code) แปลงเป็นรหัสแอสกี (ASCII) เพื่อแสดงผลเป็นตัวอักษรตามที่ได้รับมา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก

1. รูปวงจรรวม
2. แผนภาพการใช้ MEMORY MAP ของ ANT-31
3. โปรแกรมเตลไฟล์
4. โปรแกรมไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แผนภาพการใช้ MEMORY MAP ของ ANT - 31

0000H	PROGRAM MEMORY
7FFFH 8000H	ที่พักข้อมูลรูปภาพ
8FFFH 9000H	ใช้ในการ SETBIT
93FFH 9400H	ที่พักข้อมูลชนิดตัวอักษรจากเดสไฟล์ หรือที่เก็บข้อมูลที่รับโดยตรงจากคีย์บอร์ด
94FFH 9500H	
9FFFH A000H	ที่เก็บข้อมูลรูปภาพไฟล์ที่ 1
AFFFH B000H	ที่เก็บข้อมูลรูปภาพไฟล์ที่ 2
BFFFH C000H	ที่เก็บข้อมูลรูปภาพไฟล์ที่ 3
CFFFH D000H	ที่เก็บข้อมูลรูปภาพไฟล์ที่ 4
DFFFH E000H	ที่เก็บข้อมูลรูปภาพไฟล์ที่ 5
EFFFH F000H	ที่เก็บข้อมูลชนิดตัวอักษรไฟล์ที่ 1
F0FFH F100H	ที่เก็บข้อมูลชนิดตัวอักษรไฟล์ที่ 2
F1FFH F200H	ที่เก็บข้อมูลชนิดตัวอักษรไฟล์ที่ 3
F2FFH F300H	ที่เก็บข้อมูลชนิดตัวอักษรไฟล์ที่ 4
F3FFH F400H	ที่เก็บข้อมูลชนิดตัวอักษรไฟล์ที่ 5
F4FFH	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unit Unit1;

interface
uses
  Windows, Messages, SysUtils, Classes, Graphics,
  Controls, Forms, Dialogs,
  ExtCtrls, Buttons, StdCtrls, ComCtrls, Menus,
  ExtDlgs, ToolWin;

type
  TForm1 = class(TForm)
    TEXTEDIT_BOARD: TPanel;
    CONTROL_BOARD: TPanel;
    SEQUENCE_BOARD: TPanel;
    TYPE_BOARD: TPanel;
    LEFT: TRadioButton;
    RIGHT: TRadioButton;
    UP: TRadioButton;
    DOWN: TRadioButton;
    BLINK: TRadioButton;
    SHOW: TRadioButton;
    Line: TSpeedButton;
    Rectangle: TSpeedButton;
    CURSOR: TEdit;
    TEXTEDIT: TRichEdit;
    UPLEFT: TRadioButton;
    DOWNLEFT: TRadioButton;
    UPRIGHT: TRadioButton;
    DOWNRIGHT: TRadioButton;
    MainMenu1: TMainMenu;
    File1: TMenuItem;
    Exit1: TMenuItem;
    EXTEND: TRadioButton;
    SMALL: TRadioButton;
    ONE: TRadioButton;
    TWO: TRadioButton;
    THREE: TRadioButton;
    FOUR: TRadioButton;
    FIVE: TRadioButton;
    SIX: TRadioButton;
    SEVEN: TRadioButton;
    EIGHT: TRadioButton;
    NINE: TRadioButton;
    SaveDialog1: TSaveDialog;
    TEXT: TRadioButton;
    OK: TButton;
    OpenPictureDialog1: TOpenPictureDialog;
    Open1: TMenuItem;
    Close1: TMenuItem;
    NEW: TMenuItem;
    savefile: TMenuItem;
    Edit3: TEdit;
    NUMBER_OF_LINE: TLabel;
    NUMBER_OF_SHIFT: TLabel;
    Edit2: TEdit;
    newquick: TSpeedButton;
    openquick: TSpeedButton;
    closequick: TSpeedButton;
    savequick: TSpeedButton;
    exitquick: TSpeedButton;
    Bevel1: TBevel;
    Image1: TImage;

    procedure
    DrawShape(Start:TPoint;Stop:TPoint;AMode:TPE
    nMode);

    procedure LineClick(Sender: TObject);
    procedure RectangleClick(Sender: TObject);
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure LEFTClick(Sender: TObject);
procedure RIGHTClick(Sender: TObject);
procedure UPClick(Sender: TObject);
procedure DOWNClick(Sender: TObject);
procedure BLINKClick(Sender: TObject);
procedure TEXTClick(Sender: TObject);
procedure SHOWClick(Sender: TObject);
procedure TEXTEDITKeyPress(Sender:
TObject; var Key: Char);
procedure UPLEFTClick(Sender: TObject);
procedure DOWNLEFTClick(Sender: TObject);
procedure UPRIGHTClick(Sender: TObject);
procedure DOWNRIGHTClick(Sender:
TObject);
procedure Exit1Click(Sender: TObject);
procedure EXTENDClick(Sender: TObject);
procedure SMALLClick(Sender: TObject);
procedure TWOClick(Sender: TObject);
procedure ONEClick(Sender: TObject);
procedure THREEClick(Sender: TObject);
procedure FOURClick(Sender: TObject);
procedure FIVEClick(Sender: TObject);
procedure SIXClick(Sender: TObject);
procedure SEVENClick(Sender: TObject);
procedure EIGHTClick(Sender: TObject);
procedure NINEClick(Sender: TObject);
procedure Image1MouseDown(Sender:
TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
procedure Image1MouseMove(Sender: TObject;
Shift: TShiftState; X, Y: Integer);
procedure Image1MouseUp(Sender: TObject;
Button: TMouseButton; Shift: TShiftState; X, Y:
Integer);
procedure OKClick(Sender: TObject);
procedure Open1Click(Sender: TObject);
procedure Close1Click(Sender: TObject);
procedure NEWClick(Sender: TObject);
procedure savefileClick(Sender: TObject);
procedure newquickClick(Sender: TObject);
procedure openquickClick(Sender: TObject);
procedure closequickClick(Sender: TObject);
procedure savequickClick(Sender: TObject);
procedure exitquickClick(Sender: TObject);
private
FUNCTION TOHEX1(NUM:BYTE):CHAR;
} Private declarations }
public
} Public declarations }
end;
var
Form1: TForm1;
implementation
{SR *.DFM}
var
data:array[1..2160]of BYTE;
data_ascii:array [1..256]of byte;
m,n,A,B:integer;
Output_File:textfile;
Output_Filename:string[60];
count:array[0..8]of string;
typecode,num:BYTE;
TOHEX1:string;
TEXT1,TEXT2:STRING; que:BYTE;
Drawing,linecheck,sqcheck,CHECK:boolean;
StartPt,EndPt,MovePt:TPoint;
type TDrawTool=(dtLine,dtRectangle);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
var DrawTool:TDrawTool;
```

```
procedure TForm1.LineClick(Sender: TObject);
```

```
begin
```

```
  new.Enabled:=false;
```

```
  newquick.Enabled:=false;
```

```
  DrawTool:=dtLine;
```

```
  linecheck:=true;
```

```
  squcheck:=false;
```

```
  CHECK:=FALSE;
```

```
  UP.Checked:=FALSE;
```

```
  DOWN.Checked:=FALSE;
```

```
  LEFT.Checked:=FALSE;
```

```
  RIGHT.Checked:=FALSE;
```

```
  BLINK.Checked:=FALSE;
```

```
  SMALL.Checked:=FALSE;
```

```
  EXTEND.Checked:=FALSE;
```

```
  UPRIGHT.Checked:=FALSE;
```

```
  UPLEFT.Checked:=FALSE;
```

```
  DOWNRIGHT.Checked:=FALSE;
```

```
  DOWNLEFT.Checked:=FALSE;
```

```
  SHOW.Checked:=FALSE;
```

```
  UP.enabled:=true;
```

```
  DOWN.enabled:=true;
```

```
  LEFT.enabled:=true;
```

```
  RIGHT.enabled:=true;
```

```
  BLINK.enabled:=true;
```

```
  SMALL.enabled:=true;
```

```
  EXTEND.enabled:=true;
```

```
  UPRIGHT.enabled:=true;
```

```
  UPLEFT.enabled:=true;
```

```
  DOWNRIGHT.enabled:=true;
```

```
  DOWNLEFT.enabled:=true,
```

```
  SHOW.enabled:=true;
```

```
end;
```

```
procedure TForm1.RectangleClick(Sender:
```

```
TObject);
```

```
begin
```

```
  new.Enabled:=false;
```

```
  newquick.Enabled:=false;
```

```
  Canvas.Brush.Style:=bsClear;
```

```
  DrawTool:=dtRectangle;
```

```
  squcheck:=true;
```

```
  linecheck:=false;
```

```
  CHECK:=FALSE;
```

```
  UP.Checked:=FALSE;
```

```
  DOWN.Checked:=FALSE;
```

```
  LEFT.Checked:=FALSE;
```

```
  RIGHT.Checked:=FALSE;
```

```
  BLINK.Checked:=FALSE;
```

```
  SMALL.Checked:=FALSE;
```

```
  EXTEND.Checked:=FALSE;
```

```
  UPRIGHT.Checked:=FALSE;
```

```
  UPLEFT.Checked:=FALSE;
```

```
  DOWNRIGHT.Checked:=FALSE;
```

```
  DOWNLEFT.Checked:=FALSE;
```

```
  SHOW.Checked:=FALSE;
```

```
  UP.enabled:=true;
```

```
  DOWN.enabled:=true;
```

```
  LEFT.enabled:=true;
```

```
  RIGHT.enabled:=true;
```

```
  BLINK.enabled:=true;
```

```
  SMALL.enabled:=true;
```

```
  EXTEND.enabled:=true;
```

```
  UPRIGHT.enabled:=true;
```

```
  UPLEFT.enabled:=true;
```

```
  DOWNRIGHT.enabled:=true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DOWNLEFT.enabled:=true;
SHOW.enabled:=true;
end;

FUNCTION TForm1.TOHEX1
(NUM:BYTE):CHAR;
BEGIN
  case Num of
    0:ToHex1:='0';
    1:ToHex1:='1';
    2:ToHex1:='2';
    3:ToHex1:='3';
    4:ToHex1:='4';
    5:ToHex1:='5';
    6:ToHex1:='6';
    7:ToHex1:='7';
    8:ToHex1:='8';
    9:ToHex1:='9';
    10:ToHex1:='A';
    11:ToHex1:='B';
    12:ToHex1:='C';
    13:ToHex1:='D';
    14:ToHex1:='E';
    15:ToHex1:='F';
  end;
end;

procedure TForm1.LEFTClick(Sender: TObject);
begin
  CHECK:=TRUE;
  if linecheck=true then typecode:=03 else
typecode:=10;
end;

procedure TForm1.RIGHTClick(Sender: TObject);
begin
  CHECK:=TRUE;
  if linecheck=true then typecode:=04 else
typecode:=11;
end;

procedure TForm1.UPClick(Sender: TObject);
begin
  CHECK:=TRUE;
  if linecheck=true then typecode:=01 else
typecode:=08;
end;

procedure TForm1.DOWNClick(Sender:
TObject);
begin
  CHECK:=TRUE;
  if linecheck=true then typecode:=02 else
typecode:=09;
end;

procedure TForm1.BLINKClick(Sender: TObject);
begin
  CHECK:=TRUE;
  if linecheck=true then typecode:=05 else
typecode:=12;
end;

procedure TForm1.SHOWClick(Sender: TObject);
begin
  CHECK:=TRUE;
  if linecheck=true then typecode:=07 else
typecode:=14;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.UPLEFTClick(Sender:
TObject);
begin
    CHECK:=TRUE;
    if linecheck=true then typecode:=15 else
typecode:=16;
end;

procedure TForm1.DOWNLEFTClick(Sender:
TObject);
begin
    CHECK:=TRUE;
    if linecheck=true then typecode:=17 else
typecode:=18;
end;

procedure TForm1.UPRIGHTClick(Sender:
TObject);
begin
    CHECK:=TRUE;
    if linecheck=true then typecode:=19 else
typecode:=20;
end;

procedure TForm1.DOWNRIGHTClick(Sender:
TObject);
begin
    CHECK:=TRUE;
    if linecheck=true then typecode:=21 else
typecode:=22;
end;

procedure TForm1.EXTENDClick(Sender:
TObject);
begin
    CHECK:=TRUE;
    if linecheck=true then typecode:=25 else
typecode:=23;
end;

procedure TForm1.SMALLClick(Sender:
TObject);
begin
    CHECK:=TRUE;
    if linecheck=true then typecode:=26 else
typecode:=24;
end;

procedure TForm1.TEXTEDITKeyPress(Sender:
TObject; var Key: Char);
var KeyNo:integer;
begin
    KeyNo:=ord(Key);
    if KeyNo < 0 then KeyNo:=KeyNo+256;
    IF KEYNO=8 THEN M:=M-1
    ELSE
    BEGIN
        m:=m+1;
        data_ascii[m]:=keyNo;
    END;
end;

procedure TForm1.Exit1Click(Sender: TObject);
begin
    application.Terminate;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.ONEClick(Sender: TObject);
begin
  FOR N:=1 TO 240 DO
  BEGIN
    DATA[N]:=0;
  END;

  new.Enabled:=true;
newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=0;N:=0;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;
  SMALL.Checked:=FALSE;
  EXTEND.Checked:=FALSE;
  UPRIGHT.Checked:=FALSE;
  UPLEFT.Checked:=FALSE;
  DOWNRIGHT.Checked:=FALSE;
  DOWNLEFT.Checked:=FALSE;
  SHOW.Checked:=FALSE;
  UP.enabled:=false;
  DOWN.enabled:=false;
  LEFT.enabled:=false;
  RIGHT.enabled:=false;
  BLINK.enabled:=false;
  SMALL.enabled:=false;
  EXTEND.enabled:=false;
  UPRIGHT.enabled:=false;
  UPLEFT.enabled:=false;
  DOWNRIGHT.enabled:=false;
  DOWNLEFT.enabled:=false;
  SHOW.enabled:=false;
  Two.Enabled:=false;
  Three.Enabled:=false;
  Four.Enabled:=false;
  Five.Enabled:=false;
  Six.Enabled:=false;
  Seven.Enabled:=false;
  Eight.Enabled:=false;
  Nine.Enabled:=false;
  Text.Enabled:=false;
end;

procedure TForm1.TWOClick(Sender: TObject);
begin
  FOR N:=240 TO 240*2 DO
  BEGIN
    DATA[N]:=0;
  END;

  new.Enabled:=true;
newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=1; N:=240;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SMALL.Checked:=FALSE;
EXTEND.Checked:=FALSE;
UPRIGHT.Checked:=FALSE;
UPLEFT.Checked:=FALSE;
DOWNRIGHT.Checked:=FALSE;
DOWNLEFT.Checked:=FALSE;
SHOW.Checked:=FALSE;
UP.enabled:=false;
DOWN.enabled:=false;
LEFT.enabled:=false;
RIGHT.enabled:=false;
BLINK.enabled:=false;
SMALL.enabled:=false;
EXTEND.enabled:=false;
UPRIGHT.enabled:=false;
UPLEFT.enabled:=false;
DOWNRIGHT.enabled:=false;
DOWNLEFT.enabled:=false;
SHOW.enabled:=false;
One.Enabled:=false;
Three.Enabled:=false;
Four.Enabled:=false;
Five.Enabled:=false;
Six.Enabled:=false;
Seven.Enabled:=false;
Eight.Enabled:=false;
Nine.Enabled:=false;
Text.Enabled:=false;
end;

procedure TForm1.THREEClick(Sender:
TObject);
begin
    FOR N:=240*2 TO 240*3 DO
        BEGIN
            DATA[N]:=0;
            END;
            new.Enabled:=true;
            newquick.Enabled:=true;
            Line.Enabled:=true;
            Rectangle.Enabled:=true;
            CHECK:=FALSE;
            que:=2; N:=240*2;
            edit3.text:='48';
            UP.Checked:=FALSE;
            DOWN.Checked:=FALSE;
            LEFT.Checked:=FALSE;
            RIGHT.Checked:=FALSE;
            BLINK.Checked:=FALSE;
            SMALL.Checked:=FALSE;
            EXTEND.Checked:=FALSE;
            UPRIGHT.Checked:=FALSE;
            UPLEFT.Checked:=FALSE;
            DOWNRIGHT.Checked:=FALSE;
            DOWNLEFT.Checked:=FALSE;
            SHOW.Checked:=FALSE;
            UP.enabled:=false;
            DOWN.enabled:=false;
            LEFT.enabled:=false;
            RIGHT.enabled:=false;
            BLINK.enabled:=false;
            SMALL.enabled:=false;
            EXTEND.enabled:=false;
            UPRIGHT.enabled:=false;
            UPLEFT.enabled:=false;
            DOWNRIGHT.enabled:=false;
            DOWNLEFT.enabled:=false;
            SHOW.enabled:=false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

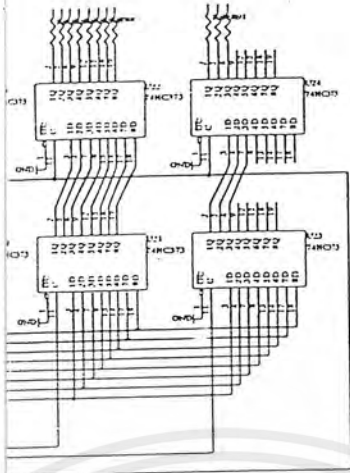
```

One.Enabled:=false;
Two.Enabled:=false;
Four.Enabled:=false;
Five.Enabled:=false;
Six.Enabled:=false;
Seven.Enabled:=false;
Eight.Enabled:=false;
Nine.Enabled:=false;
Text.Enabled:=false;
end;

procedure TForm1.FOURClick(Sender: TObject);
begin
  FOR N:=240*3 TO 240*4 DO
  BEGIN
    DATA[N]:=0;
  END;
  new.Enabled:=true;
  newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=3; N:=240*3;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;
  SMALL.Checked:=FALSE;
  EXTEND.Checked:=FALSE;
  UPRIGHT.Checked:=FALSE;
  UPLEFT.Checked:=FALSE;
  DOWNRIGHT.Checked:=FALSE;
  DOWNLEFT.Checked:=FALSE;
  SHOW.Checked:=FALSE;
  UP.enabled:=false;
  DOWN.enabled:=false;
  LEFT.enabled:=false;
  RIGHT.enabled:=false;
  BLINK.enabled:=false;
  SMALL.enabled:=false;
  EXTEND.enabled:=false;
  UPRIGHT.enabled:=false;
  UPLEFT.enabled:=false;
  DOWNRIGHT.enabled:=false;
  DOWNLEFT.enabled:=false;
  SHOW.enabled:=false;
  One.Enabled:=false;
  Two.Enabled:=false;
  Three.Enabled:=false;
  Five.Enabled:=false;
  Six.Enabled:=false;
  Seven.Enabled:=false;
  Eight.Enabled:=false;
  Nine.Enabled:=false;
  Text.Enabled:=false;
end;
procedure TForm1.FIVEClick(Sender: TObject);
begin
  FOR N:=240*4 TO 240*5 DO
  BEGIN
    DATA[N]:=0;
  END;
  new.Enabled:=true;
  newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ร.บ.จ.บ.ร.ม.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHECK:=FALSE;
que:=4; N:=240*4;
edit3.text:='48';
UP.Checked:=FALSE;
DOWN.Checked:=FALSE;
LEFT.Checked:=FALSE;
RIGHT.Checked:=FALSE;
BLINK.Checked:=FALSE;
SMALL.Checked:=FALSE;
EXTEND.Checked:=FALSE;
UPRIGHT.Checked:=FALSE;
UPLEFT.Checked:=FALSE;
DOWNRIGHT.Checked:=FALSE;
DOWNLEFT.Checked:=FALSE;
SHOW.Checked:=FALSE;
UP.enabled:=false;
DOWN.enabled:=false;
LEFT.enabled:=false;
RIGHT.enabled:=false;
BLINK.enabled:=false;
SMALL.enabled:=false;
EXTEND.enabled:=false;
UPRIGHT.enabled:=false;
UPLEFT.enabled:=false;
DOWNRIGHT.enabled:=false;
DOWNLEFT.enabled:=false;
SHOW.enabled:=false;
One.Enabled:=false;
Two.Enabled:=false;
Three.Enabled:=false;
Four.Enabled:=false;
Six.Enabled:=false;
Seven.Enabled:=false;
Eight.Enabled:=false;
Nine.Enabled:=false;
Text.Enabled:=false;
end;
procedure TForm1.SIXClick(Sender: TObject);
begin
  FOR N:=240*5 TO 240*6 DO
  BEGIN
    DATA[N]:=0;
  END;
  new.Enabled:=true;
  newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=5; N:=240*5;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;
  SMALL.Checked:=FALSE;
  EXTEND.Checked:=FALSE;
  UPRIGHT.Checked:=FALSE;
  UPLEFT.Checked:=FALSE;
  DOWNRIGHT.Checked:=FALSE;
  DOWNLEFT.Checked:=FALSE;
  SHOW.Checked:=FALSE;
  UP.enabled:=false;
  DOWN.enabled:=false;
  LEFT.enabled:=false;
  RIGHT.enabled:=false;
  BLINK.enabled:=false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SMALL.Enabled:=false;
EXTEND.Enabled:=false;
UPRIGHT.Enabled:=false;
UPLEFT.Enabled:=false;
DOWNRIGHT.Enabled:=false;
DOWNLEFT.Enabled:=false;
SHOW.Enabled:=false;
One.Enabled:=false;
Two.Enabled:=false;
Three.Enabled:=false;
Four.Enabled:=false;
Five.Enabled:=false;
Seven.Enabled:=false;
Eight.Enabled:=false;
Nine.Enabled:=false;
Text.Enabled:=false;
end;

procedure TForm1.SEVENClick(Sender:
TObject);
begin
  FOR N:=240*6 TO 240*7 DO
  BEGIN
    DATA[N]:=0;
  END;
  new.Enabled:=true;
  newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=6; N:=240*6;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;
  SMALL.Checked:=FALSE;
  EXTEND.Checked:=FALSE;
  UPRIGHT.Checked:=FALSE;
  UPLEFT.Checked:=FALSE;
  DOWNRIGHT.Checked:=FALSE;
  DOWNLEFT.Checked:=FALSE;
  SHOW.Checked:=FALSE;
  UP.Enabled:=false;
  DOWN.Enabled:=false;
  LEFT.Enabled:=false;
  RIGHT.Enabled:=false;
  BLINK.Enabled:=false;
  SMALL.Enabled:=false;
  EXTEND.Enabled:=false;
  UPRIGHT.Enabled:=false;
  UPLEFT.Enabled:=false;
  DOWNRIGHT.Enabled:=false;
  DOWNLEFT.Enabled:=false;
  SHOW.Enabled:=false;
  One.Enabled:=false;
  Two.Enabled:=false;
  Three.Enabled:=false;
  Four.Enabled:=false;
  Five.Enabled:=false;
  Six.Enabled:=false;
  Eight.Enabled:=false;
  Nine.Enabled:=false;
  Text.Enabled:=false;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.EIGHTClick(Sender: TObject);
begin
  FOR N:=240*7 TO 240*8 DO
  BEGIN
    DATA[N]:=0;
  END;
  new.Enabled:=true;
  newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=7; N:=240*7;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;
  SMALL.Checked:=FALSE;
  EXTEND.Checked:=FALSE;
  UPRIGHT.Checked:=FALSE;
  UPLEFT.Checked:=FALSE;
  DOWNRIGHT.Checked:=FALSE;
  DOWNLEFT.Checked:=FALSE;
  SHOW.Checked:=FALSE;
  UP.enabled:=false;
  DOWN.enabled:=false;
  LEFT.enabled:=false;
  RIGHT.enabled:=false;
  BLINK.enabled:=false;
  SMALL.enabled:=false;
  EXTEND.enabled:=false;
  UPRIGHT.enabled:=false;
  UPLEFT.enabled:=false;
  DOWNRIGHT.enabled:=false;
  DOWNLEFT.enabled:=false;
  SHOW.enabled:=false;
  One.Enabled:=false;
  Two.Enabled:=false;
  Three.Enabled:=false;
  Four.Enabled:=false;
  Five.Enabled:=false;
  Six.Enabled:=false;
  Seven.Enabled:=false;
  Nine.Enabled:=false;
  Text.Enabled:=false;
end;
procedure TForm1.NINEClick(Sender: TObject);
begin
  FOR N:=240*8 TO 240*9 DO
  BEGIN
    DATA[N]:=0;
  END;
  new.Enabled:=true;
  newquick.Enabled:=true;
  Line.Enabled:=true;
  Rectangle.Enabled:=true;
  CHECK:=FALSE;
  que:=8; N:=240*8;
  edit3.text:='48';
  UP.Checked:=FALSE;
  DOWN.Checked:=FALSE;
  LEFT.Checked:=FALSE;
  RIGHT.Checked:=FALSE;
  BLINK.Checked:=FALSE;
  SMALL.Checked:=FALSE;
  EXTEND.Checked:=FALSE;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

UPRIGHT.Checked:=FALSE;
UPLEFT.Checked:=FALSE;
DOWNRIGHT.Checked:=FALSE;
DOWNLEFT.Checked:=FALSE;
SHOW.Checked:=FALSE;
UP.enabled:=false;
DOWN.enabled:=false;
LEFT.enabled:=false;
RIGHT.enabled:=false;
BLINK.enabled:=false;
SMALL.enabled:=false;
EXTEND.enabled:=false;
UPRIGHT.enabled:=false;
UPLEFT.enabled:=false;
DOWNRIGHT.enabled:=false;
DOWNLEFT.enabled:=false;
SHOW.enabled:=false;
One.Enabled:=false;
Two.Enabled:=false;
Three.Enabled:=false;
Four.Enabled:=false;
Five.Enabled:=false;
Six.Enabled:=false;
Seven.Enabled:=false;
Eight.Enabled:=false;
nine.Enabled:=false;
CHECK:=FALSE;
TEXTedit.Visible:=true;
FOR N=1 TO 256 DO
BEGIN
DATA_ASCII[N]:=0;
end;
end;
procedure TForm1.TEXTClick(Sender: TObject);
begin
new.Enabled:=true;
newquick.Enabled:=true;
Line.Enabled:=false;
Rectangle.Enabled:=false;
CHECK:=FALSE;
UP.enabled:=false;
DOWN.enabled:=false;
LEFT.enabled:=false;
RIGHT.enabled:=false;
BLINK.enabled:=false;
SMALL.enabled:=false;
EXTEND.enabled:=false;
UPRIGHT.enabled:=false;
UPLEFT.enabled:=false;
DOWNRIGHT.enabled:=false;
DOWNLEFT.enabled:=false;
SHOW.enabled:=false;
One.Enabled:=false;
Two.Enabled:=false;
Three.Enabled:=false;
Four.Enabled:=false;
Five.Enabled:=false;
Six.Enabled:=false;
Seven.Enabled:=false;
Eight.Enabled:=false;
nine.Enabled:=false;
CHECK:=FALSE;
TEXTedit.Visible:=true;
FOR N=1 TO 256 DO
BEGIN
DATA_ASCII[N]:=0;
end;
end;
procedure TForm1.Image1MouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
begin
if check=true then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
    Drawing:=True;
    Canvas.MoveTo(X,Y);
    StartPt:=Point(X,Y);
    MovePt:=StartPt;
    n:=n+1;
    data[n]:=X div 4;
    n:=n+1;
    data[n]:=60-(Y div 4);
end
else Drawing:=False;
end;

procedure TForm1.Image1MouseMove(Sender:
TObject; Shift: TShiftState; X, Y: Integer);
var xx,yy:integer; XStr,YStr:string[4];
begin
    if Drawing then
        begin
            DrawShape(StartPt,MovePt,pmNOTXOR);
            MovePt:=Point(X,Y);
            DrawShape(StartPt,MovePt,pmNOTXOR);
        end;
        xx:=X div 4;
        Yy:=Y div 4;
        str(Xx,XStr);
        str(Yy,YStr);
        CURSOR.Text:=XStr+' '+YStr;
end;

procedure TForm1.Image1MouseUp(Sender:
TObject; Button: TMouseButton; Shift:
TShiftState; X, Y: Integer);
begin
    if Drawing then
        begin
            n:=n+1;
            data[n]:=X div 4;
            n:=n+1;
            data[n]:=60-(Y div 4);
            DrawShape(StartPt,Point(X,Y),pmCopy);
            Drawing:=False;
            n:=n+1;
            data[n]:=typecode;
            edit3.text:=inttostr(strtoint(edit3.text)-1);
        end;
    end;
end;

procedure TForm1.DrawShape(Start,Stop:TPoint;AMode:TP
cnMode);
VAR I,J:INTEGER;
begin
    IMAGE1.Canvas.PEN.Mode:=AMODE;
    case DrawTool of dtLine:begin
        IMAGE1.Canvas.MoveTo
            (Start.X,Start.Y);
        IMAGE1.Canvas.LineTo
            (Stop.X,Stop.Y);
        end;
    dtRectangle:
        BEGIN
            FOR I:=START.X TO STOP.X DO
                BEGIN
                    IMAGE1.CANVAS.Pixels
                        [I,START.Y]:=CLBLACK;
                    IMAGE1.CANVAS.PIXELS
                        [I,STOP.Y]:=CLBLACK;
                end;
            end;
        end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

END;
FOR I:=START.X DOWNT0 STOP.x DO
BEGIN
IMAGE1.CANVAS.PIXELS
[I,STOP.Y]:=CLBLACK;
IMAGE1.CANVAS.Pixels
[I,START.Y]:=CLBLACK;
END;

FOR J:=START.Y TO STOP.Y DO
BEGIN
IMAGE1.CANVAS.Pixels
[START.X,J]:=CLBLACK;
IMAGE1.CANVAS.PIXELS
[STOP.X,J]:=CLBLACK;
END;
FOR J:=START.Y DOWNT0 STOP.Y
DO
BEGIN
IMAGE1.CANVAS.PIXELS
[STOP.X,J]:=CLBLACK;
IMAGE1.CANVAS.Pixels
[START.X,J]:=CLBLACK;
END;
END;
end;
end;

procedure TForm1.OKClick(Sender: TObject);
VAR R:INTEGER;
begin
FOR R:=QUE TO 8 DO
BEGIN
count[R]:=edit2.text;
END;
end;

procedure TForm1.Open1Click(Sender: TObject);
begin
IF OPENPICTUREDIALOG1.EXECUTE THEN
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IMAGE1.PICTURE.ILOADFROMFILE
(OPENPICTUREDIALOG1.FILENAME);
  open1.Enabled:=false;
  Openquick.Enabled:=false;
  UP.Enabled:=false;
  DOWN.Enabled:=false;
  LEFT.Enabled:=false;
  RIGHT.Enabled:=false;
  BLINK.Enabled:=false;
  SMALL.Enabled:=false;
  EXTEND.Enabled:=false;
  UPRIGHT.Enabled:=false;
  UPLEFT.Enabled:=false;
  DOWNRIGHT.Enabled:=false;
  DOWNLEFT.Enabled:=false;
  SHOW.Enabled:=false;
end;
end;
procedure TForm1.Close1Click(Sender: TObject);
begin
  TEXTedit.lines.Clear;
  textedit.Visible:=false;
  check:=false;
  IMAGE1.CANVAS.Rectangle(0,0,370,250);
  Open1.Enabled:=true;
  Openquick.Enabled:=true;
  UP.Enabled:=false;
  DOWN.Enabled:=false;
  LEFT.Enabled:=false;
  RIGHT.Enabled:=false;
  BLINK.Enabled:=false;
  SMALL.Enabled:=false;
  EXTEND.Enabled:=false;
  UPRIGHT.Enabled:=false;
  UPLEFT.Enabled:=false;
  DOWNRIGHT.Enabled:=false;
  DOWNLEFT.Enabled:=false;
  SHOW.Enabled:=false;
  One.Enabled:=True;
  Two.Enabled:=True;
  Three.Enabled:=True;
  Four.Enabled:=True;
  Five.Enabled:=True;
  Six.Enabled:=True;
  Seven.Enabled:=True;
  Eight.Enabled:=True;
  Nine.Enabled:=True;
  Text.Enabled:=True;
  One.Checked:=false;
  Two.Checked:=false;
  Three.Checked:=false;
  Four.Checked:=false;
  Five.Checked:=false;
  Six.Checked:=false;
  Seven.Checked:=false;
  Eight.Checked:=false;
  Nine.Checked:=false;
  Text.Checked:=false;
end;
procedure TForm1.NEWClick(Sender: TObject);
begin
  IMAGE1.CANVAS.Rectangle(0,0,370,250);
  closequick.Enabled:=true;
  close1.Enabled:=true;
  new.Enabled:=false;
  newquick.Enabled:=false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TEXTedit.lines.Clear;
textedit.Visible:=false;
n:=0; m:=11;QUE:=0;
EDIT2.TEXT:='10';
edit3.text:='48';
ok.Enabled:=true;
One.Enabled:=True;
Two.Enabled:=True;
Three.Enabled:=True;
Four.Enabled:=True;
Five.Enabled:=True;
Six.Enabled:=True;
Seven.Enabled:=True;
Eight.Enabled:=True;
Nine.Enabled:=True;
Text.Enabled:=True;
One.checked:=false;
Two.checked:=false;
Three.checked:=false;
Four.checked:=false;
Five.checked:=false;
Six.checked:=false;
Seven.checked:=false;
Eight.checked:=false;
Nine.checked:=false;
Text.checked:=false;
FOR N:=1 TO 2160 DO
BEGIN
DATA[N]:=0;
END;
FOR N:=1 TO 256 DO
BEGIN
DATA_ASCII[N]:=0;
end;

line.Enabled:=false;
rectangle.Enabled:=false;
end;

procedure TForm1.savefileClick(Sender:
TObject);
var
DATA_TEXT:ARRAY[1..2160]OF STRING;
TYPECODE_TEXT,CHECKCODE_TEXT,CHE
CKSUM_TEXT:STRING[2];CHECKSUM:BYTE
;
begin
// one.Checked:=true;
// Two.Enabled:=false;
// Three.Enabled:=false;
// Four.Enabled:=false;
// Five.Enabled:=false;
// Six.Enabled:=false;
// Seven.Enabled:=false;
// Eight.Enabled:=false;
// nine.Enabled:=false;
// Text.Enabled:=false;
// line.Enabled:=false;
// rectangle.Enabled:=false;
// UP.enabled:=false;
// DOWN.enabled:=false;
// LEFT.enabled:=false;
// RIGHT.enabled:=false;
// BLINK.enabled:=false;
// SMALL.enabled:=false;
// EXTEND.enabled:=false;
// UPRIGHT.enabled:=false;
// UPLEFT.enabled:=false;
// DOWNRIGHT.enabled:=false;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// DOWNLEFT.enabled:=false;
// SHOW.enabled:=false;
// savefile.Enabled:=false;
// savequick.Enabled:=false;
if SaveDialog1.Execute then
    Output_Filename:=SaveDialog1.FileName;
    assignfile(Output_File,Output_Filename);
    Rewrite(Output_File);
FOR B:=0 TO 8 DO
BEGIN
    QUE:=B;
    writeln(Output_File);
    CHECKSUM:=16+128;
    CASE QUE OF
        1:CHECKSUM:=CHECKSUM+1;
        2:CHECKSUM:=CHECKSUM+2;
        3:CHECKSUM:=CHECKSUM+3;
        4:CHECKSUM:=CHECKSUM+4;
        5:CHECKSUM:=CHECKSUM+5;
        6:CHECKSUM:=CHECKSUM+6;
        7:CHECKSUM:=CHECKSUM+7;
        8:CHECKSUM:=CHECKSUM+8;
    ELSE CHECKSUM:=CHECKSUM;
    end;
    CASE QUE OF
        1:write(Output_File,'10810000');
        2:write(Output_File,'10820000');
        3:write(Output_File,'10830000');
        4:write(Output_File,'10840000');
        5:write(Output_File,'10850000');
        6:write(Output_File,'10860000');
        7:write(Output_File,'10870000');
        8:write(Output_File,'10880000');
    ELSE write(Output_File,'10800000');
end;
end;
FOR A:=1 TO 15 DO
BEGIN
    N:=A;
    CASE QUE OF
        1:N:=N+240;
        2:N:=N+240*2;
        3:N:=N+240*3;
        4:N:=N+240*4;
        5:N:=N+240*5;
        6:N:=N+240*6;
        7:N:=N+240*7;
        8:N:=N+240*8;
    ELSE N:=N;
    end;
    CHECKSUM:=CHECKSUM+DATA[N];
    NUM:=DATA[N] MOD 16;
    TEXT1:=TOHEX1(NUM);
    NUM:=DATA[N] DIV 16;
    TEXT2:=TOHEX1(NUM);
    DATA_TEXT[N]:=TEXT2+TEXT1;
    write(Output_File,DATA_TEXT[N]);
END;
CHECKSUM:=256-(CHECKSUM+strtoint
(count[que]) MOD 16);
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
NUM:=strtoint(count[que]) MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=strtoint(count[que]) DIV 16;
TEXT2:=TOHEX1(NUM);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHECKCODE_TEXT:=TEXT2+TEXT1;
write(Output_File,CHECKCODE_TEXT);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+16;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
  1:write(Output_File,':10811000');
  2:write(Output_File,':10821000');
  3:write(Output_File,':10831000');
  4:write(Output_File,':10841000');
  5:write(Output_File,':10851000');
  6:write(Output_File,':10861000');
  7:write(Output_File,':10871000');
  8:write(Output_File,':10881000');
ELSE write(Output_File,':10801000');
end;
FOR A:=16 TO 30 DO
BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;
    3:N:=N+240*3;
    4:N:=N+240*4;
    5:N:=N+240*5;
    6:N:=N+240*6;
    7:N:=N+240*7;
    8:N:=N+240*8;
    ELSE N:=N;
  end;
  CHECKSUM:=CHECKSUM+DATA[N];
  NUM:=DATA[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
  write(Output_File,DATA_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+32;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
1:write(Output_File,'10812000');
2:write(Output_File,'10822000');
3:write(Output_File,'10832000');
4:write(Output_File,'10842000');
5:write(Output_File,'10852000');
6:write(Output_File,'10862000');
7:write(Output_File,'10872000');
8:write(Output_File,'10882000');
ELSE write(Output_File,'10802000');
end;
FOR A:=31 TO 45 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N+240;
2:N:=N+240*2;
3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+48;
CASE QUE OF
1:CHECKSUM:=CHECKSUM+1;
2:CHECKSUM:=CHECKSUM+2;
3:CHECKSUM:=CHECKSUM+3;
4:CHECKSUM:=CHECKSUM+4;
5:CHECKSUM:=CHECKSUM+5;
6:CHECKSUM:=CHECKSUM+6;
7:CHECKSUM:=CHECKSUM+7;
8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
1:write(Output_File,'10813000');
2:write(Output_File,'10823000');
3:write(Output_File,'10833000');
4:write(Output_File,'10843000');
5:write(Output_File,'10853000');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

6:write(Output_File,'10863000');
7:write(Output_File,'10873000');
8:write(Output_File,'10883000');
ELSE write(Output_File,'10803000');
end;
FOR A:=46 TO 60 DO
BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;
    3:N:=N+240*3;
    4:N:=N+240*4;
    5:N:=N+240*5;
    6:N:=N+240*6;
    7:N:=N+240*7;
    8:N:=N+240*8;
  ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,data_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+64;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
  1:write(Output_File,'10814000');
  2:write(Output_File,'10824000');
  3:write(Output_File,'10834000');
  4:write(Output_File,'10844000');
  5:write(Output_File,'10854000');
  6:write(Output_File,'10864000');
  7:write(Output_File,'10874000');
  8:write(Output_File,'10884000');
ELSE write(Output_File,'10804000');
end;
FOR A:=61 TO 75 DO
BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+80;
CASE QUE OF
1:CHECKSUM:=CHECKSUM+1;
2:CHECKSUM:=CHECKSUM+2;
3:CHECKSUM:=CHECKSUM+3;
4:CHECKSUM:=CHECKSUM+4;
5:CHECKSUM:=CHECKSUM+5;
6:CHECKSUM:=CHECKSUM+6;
7:CHECKSUM:=CHECKSUM+7;
8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
1:write(Output_File,'10815000');
2:write(Output_File,'10825000');
3:write(Output_File,'10835000');
4:write(Output_File,'10845000');
5:write(Output_File,'10855000');
6:write(Output_File,'10865000');
7:write(Output_File,'10875000');
8:write(Output_File,'10885000');
ELSE write(Output_File,'10805000');
end;
FOR A:=76 TO 90 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N+240;
2:N:=N+240*2;
3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+128+96;
CASE QUE OF
1:CHECKSUM:=CHECKSUM+1;
2:CHECKSUM:=CHECKSUM+2;
3:CHECKSUM:=CHECKSUM+3;
4:CHECKSUM:=CHECKSUM+4;
5:CHECKSUM:=CHECKSUM+5;
6:CHECKSUM:=CHECKSUM+6;
7:CHECKSUM:=CHECKSUM+7;
8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
1:write(Output_File,':10816000');
2:write(Output_File,':10826000');
3:write(Output_File,':10836000');
4:write(Output_File,':10846000');
5:write(Output_File,':10856000');
6:write(Output_File,':10866000');
7:write(Output_File,':10876000');

8:write(Output_File,':10886000');
ELSE write(Output_File,':10806000');
end;
FOR A:=91 TO 105 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N+240;
2:N:=N+240*2;
3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,0);
write(Output_File,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+128+112-256;

CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
  1:write(Output_File,':10817000');
  2:write(Output_File,':10827000');
  3:write(Output_File,':10837000');
  4:write(Output_File,':10847000');
  5:write(Output_File,':10857000');
  6:write(Output_File,':10867000');
  7:write(Output_File,':10877000');
  8:write(Output_File,':10887000');
ELSE write(Output_File,':10807000');
end;
FOR A:=106 TO 120 DO
  BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;
    3:N:=N+240*3;
    4:N:=N+240*4;
    5:N:=N+240*5;
    6:N:=N+240*6;
    7:N:=N+240*7;
    8:N:=N+240*8;
  ELSE N:=N;
  end;
  CHECKSUM:=CHECKSUM+DATA[N];
  NUM:=DATA[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
  write(Output_File,data_TEXT[n]);
  end;
  CHECKSUM:=256-CHECKSUM;
  NUM:=CHECKSUM MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=CHECKSUM DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;
  write(Output_File,0);
  write(Output_File,0);
  write(Output_File,CHECKSUM_TEXT);
  writeln(Output_File);
  CHECKSUM:=16+128+128-256;
  CASE QUE OF
    1:CHECKSUM:=CHECKSUM+1;
    2:CHECKSUM:=CHECKSUM+2;
    3:CHECKSUM:=CHECKSUM+3;
    4:CHECKSUM:=CHECKSUM+4;
    5:CHECKSUM:=CHECKSUM+5;
    6:CHECKSUM:=CHECKSUM+6;
    7:CHECKSUM:=CHECKSUM+7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;

CASE QUE OF
  1:write(Output_File,'10818000');
  2:write(Output_File,'10828000');
  3:write(Output_File,'10838000');
  4:write(Output_File,'10848000');
  5:write(Output_File,'10858000');
  6:write(Output_File,'10868000');
  7:write(Output_File,'10878000');
  8:write(Output_File,'10888000');
ELSE write(Output_File,'10808000');
end;
FOR A:=121 TO 135 DO
BEGIN
N:=A;
CASE QUE OF
  1:N:=N+240;
  2:N:=N+240*2;
  3:N:=N+240*3;
  4:N:=N+240*4;
  5:N:=N+240*5;
  6:N:=N+240*6;
  7:N:=N+240*7;
  8:N:=N+240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[N]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+144-256;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
  1:write(Output_File,'10819000');
  2:write(Output_File,'10829000');
  3:write(Output_File,'10839000');
  4:write(Output_File,'10849000');
  5:write(Output_File,'10859000');
  6:write(Output_File,'10869000');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

7:write(Output_File,'10879000');
8:write(Output_File,'10889000');
ELSE write(Output_File,'10809000');
end;
FOR A:=136 TO 150 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N+240;
2:N:=N+240*2;
3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+160-256;
CASE QUE OF
1:CHECKSUM:=CHECKSUM+1;
2:CHECKSUM:=CHECKSUM+2;
3:CHECKSUM:=CHECKSUM+3;
4:CHECKSUM:=CHECKSUM+4;
5:CHECKSUM:=CHECKSUM+5;
6:CHECKSUM:=CHECKSUM+6;
7:CHECKSUM:=CHECKSUM+7;
8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
1:write(Output_File,'1081A000');
2:write(Output_File,'1082A000');
3:write(Output_File,'1083A000');
4:write(Output_File,'1084A000');
5:write(Output_File,'1085A000');
6:write(Output_File,'1086A000');
7:write(Output_File,'1087A000');
8:write(Output_File,'1088A000');
ELSE write(Output_File,'1080A000');
end;
FOR A:=151 TO 165 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N+240;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2:N:=N+240*2;
3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;
end;

```

CASE QUE OF

```

CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[N]);
end;

CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+128+176-256;
CASE QUE OF
1:CHECKSUM:=CHECKSUM+1;
2:CHECKSUM:=CHECKSUM+2;
3:CHECKSUM:=CHECKSUM+3;
4:CHECKSUM:=CHECKSUM+4;
5:CHECKSUM:=CHECKSUM+5;
6:CHECKSUM:=CHECKSUM+6;
7:CHECKSUM:=CHECKSUM+7;
8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
end;

1:write(Output_File,'1081B000');
2:write(Output_File,'1082B000');
3:write(Output_File,'1083B000');
4:write(Output_File,'1084B000');
5:write(Output_File,'1085B000');
6:write(Output_File,'1086B000');
7:write(Output_File,'1087B000');
8:write(Output_File,'1088B000');
ELSE write(Output_File,'1080B000');
end;

FOR A:=166 TO 180 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N+240;
2:N:=N+240*2;
3:N:=N+240*3;
4:N:=N+240*4;
5:N:=N+240*5;
6:N:=N+240*6;
7:N:=N+240*7;
8:N:=N+240*8;
ELSE N:=N;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,data_TEXT[n]);
end;

CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+128+192-256;
CASE QUE OF
1:CHECKSUM:=CHECKSUM+1;
2:CHECKSUM:=CHECKSUM+2;
3:CHECKSUM:=CHECKSUM+3;
4:CHECKSUM:=CHECKSUM+4;
5:CHECKSUM:=CHECKSUM+5;
6:CHECKSUM:=CHECKSUM+6;
7:CHECKSUM:=CHECKSUM+7;
8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
1:write(Output_File,'1081C000');
2:write(Output_File,'1082C000');
3:write(Output_File,'1083C000');
4:write(Output_File,'1084C000');
5:write(Output_File,'1085C000');
6:write(Output_File,'1086C000');
7:write(Output_File,'1087C000');
8:write(Output_File,'1088C000');
ELSE write(Output_File,'1080C000');
end;

FOR A:=181 TO 195 DO
BEGIN
N:=A;
CASE QUE OF
1:N:=N-240;
2:N:=N-240*2;
3:N:=N-240*3;
4:N:=N-240*4;
5:N:=N-240*5;
6:N:=N-240*6;
7:N:=N-240*7;
8:N:=N-240*8;
ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,DATA_TEXT[n]);
end;

CHECKSUM:=256-CHECKSUM;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+128+208-256;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;

CASE QUE OF
  1:write(Output_File,'1081D000');
  2:write(Output_File,'1082D000');
  3:write(Output_File,'1083D000');
  4:write(Output_File,'1084D000');
  5:write(Output_File,'1085D000');
  6:write(Output_File,'1086D000');
  7:write(Output_File,'1087D000');
  8:write(Output_File,'1088D000');
ELSE write(Output_File,'1080D000');
end;

FOR A:=196 TO 210 DO
  BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;
    3:N:=N+240*3;
    4:N:=N+240*4;
    5:N:=N+240*5;
    6:N:=N+240*6;
    7:N:=N+240*7;
    8:N:=N+240*8;
  ELSE N:=N;
  end;
  CHECKSUM:=CHECKSUM+DATA[N];
  NUM:=DATA[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
  write(Output_File,DATA_TEXT[N]);
end;

CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHECKSUM:=16+128+224-256;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;
end;
CASE QUE OF
  1:write(Output_File,'1081E000');
  2:write(Output_File,'1082E000');
  3:write(Output_File,'1083E000');
  4:write(Output_File,'1084E000');
  5:write(Output_File,'1085E000');
  6:write(Output_File,'1086E000');
  7:write(Output_File,'1087E000');
  8:write(Output_File,'1088E000');
ELSE write(Output_File,'1080E000');
end;
FOR A:=211 TO 225 DO
BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;
    3:N:=N+240*3;
    4:N:=N+240*4;
    5:N:=N+240*5;
    6:N:=N+240*6;
    7:N:=N+240*7;
    8:N:=N+240*8;
    ELSE N:=N;
  end;
  CHECKSUM:=CHECKSUM+DATA[N];
  NUM:=DATA[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
  write(Output_File,DATA_TEXT[N]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,0);
write(Output_File,0);
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
CHECKSUM:=16+128+240+170-256-256;
CASE QUE OF
  1:CHECKSUM:=CHECKSUM+1;
  2:CHECKSUM:=CHECKSUM+2;
  3:CHECKSUM:=CHECKSUM+3;
  4:CHECKSUM:=CHECKSUM+4;
  5:CHECKSUM:=CHECKSUM+5;
  6:CHECKSUM:=CHECKSUM+6;
  7:CHECKSUM:=CHECKSUM+7;
  8:CHECKSUM:=CHECKSUM+8;
ELSE CHECKSUM:=CHECKSUM;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end;
CASE QUE OF
  1:write(Output_File,'1081F000');
  2:write(Output_File,'1082F000');
  3:write(Output_File,'1083F000');
  4:write(Output_File,'1084F000');
  5:write(Output_File,'1085F000');
  6:write(Output_File,'1086F000');
  7:write(Output_File,'1087F000');
  8:write(Output_File,'1088F000');
ELSE write(Output_File,'1080F000');
end;
FOR A:=226 TO 240 DO
BEGIN
  N:=A;
  CASE QUE OF
    1:N:=N+240;
    2:N:=N+240*2;
    3:N:=N+240*3;
    4:N:=N+240*4;
    5:N:=N+240*5;
    6:N:=N+240*6;
    7:N:=N+240*7;
    8:N:=N+240*8;
  ELSE N:=N;
end;
CHECKSUM:=CHECKSUM+DATA[N];
NUM:=DATA[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
write(Output_File,data_TEXT[n]);
end;
CHECKSUM:=256-CHECKSUM;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,'AA');
write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
END;
////////////////////2////////////////////
CHECKSUM:=16+148;
FOR N:=1 TO 16 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=1 TO 16 DO
BEGIN
  NUM:=DATA_ASCII[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA_ASCII[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
END;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,'10940000');
for n:=1 to 16 do
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+16;
FOR N:=17 TO 32 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=17 TO 32 DO
BEGIN
NUM:=DATA_ASCII[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA_ASCII[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
END;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,'10941000');
for n:=17 to 32 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+48;
FOR N:=49 TO 64 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=49 TO 64 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=33 TO 48 DO
BEGIN
NUM:=DATA_ASCII[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA_ASCII[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
END;
NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,'10942000');
for n:=33 to 48 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+48;
FOR N:=49 TO 64 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=49 TO 64 DO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

BEGIN
    NUM:=DATA_ASCII[N] MOD 16;
    TEXT1:=TOHEX1(NUM);
    NUM:=DATA_ASCII[N] DIV 16;
    TEXT2:=TOHEX1(NUM);
    DATA_TEXT[N]:=TEXT2+TEXT1;
END;

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,'10943000');
for n:=49 to 64 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
    CHECKSUM:=16+148+64;
FOR N:=65 TO 80 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=65 TO 80 DO
BEGIN
    NUM:=DATA_ASCII[N] MOD 16;
    TEXT1:=TOHEX1(NUM);
    NUM:=DATA_ASCII[N] DIV 16;
    TEXT2:=TOHEX1(NUM);
    DATA_TEXT[N]:=TEXT2+TEXT1;
END;

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,'10944000');
for n:=65 to 80 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);
    CHECKSUM:=16+148+80;
FOR N:=81 TO 96 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=81 TO 96 DO
BEGIN
    NUM:=DATA_ASCII[N] MOD 16;
    TEXT1:=TOHEX1(NUM);
    NUM:=DATA_ASCII[N] DIV 16;
    TEXT2:=TOHEX1(NUM);
    DATA_TEXT[N]:=TEXT2+TEXT1;
END;

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

write(Output_File,':10945000');
  for n:=81 to 96 do
  begin
  write(Output_File,data_TEXT[n]);
  end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+96-256;
FOR N:=97 TO 112 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=97 TO 112 DO
BEGIN
  NUM:=DATA_ASCII[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA_ASCII[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
END;
  NUM:=CHECKSUM MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=CHECKSUM DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,':10946000');
for n:=97 to 112 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+128-256;
FOR N:=129 TO 144 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=129 TO 144 DO
BEGIN
  NUM:=DATA_ASCII[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA_ASCII[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;
END;
write(Output_File);
CHECKSUM:=16+148+112-256;
FOR N:=113 TO 128 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=113 TO 128 DO
BEGIN
  NUM:=DATA_ASCII[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA_ASCII[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
END;
  NUM:=CHECKSUM MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=CHECKSUM DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;
write(Output_File,':10947000');
for n:=113 to 128 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+128-256;
FOR N:=129 TO 144 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;
FOR N:=129 TO 144 DO
BEGIN
  NUM:=DATA_ASCII[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=CHECKSUM DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;
END;
write(Output_File);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CHECKSUM:=256-CHECKSUM;
FOR N:=129 TO 144 DO
BEGIN
  NUM:=DATA_ASCII[N] MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=DATA_ASCII[N] DIV 16;
  TEXT2:=TOHEX1(NUM);
  DATA_TEXT[N]:=TEXT2+TEXT1;
END;

  NUM:=CHECKSUM MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=CHECKSUM DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,'10948000');
for n:=129 to 144 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+144-256;
FOR N:=145 TO 160 DO
  BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
  END;
CHECKSUM:=256-CHECKSUM;
FOR N:=145 TO 160 DO
  BEGIN
NUM:=DATA_ASCII[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA_ASCII[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
  END;
  NUM:=CHECKSUM MOD 16;
  TEXT1:=TOHEX1(NUM);
  NUM:=CHECKSUM DIV 16;
  TEXT2:=TOHEX1(NUM);
  CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,'10949000');
for n:=145 to 160 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+160-256;
FOR N:=161 TO 176 DO
  BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
  END;
CHECKSUM:=256-CHECKSUM;
FOR N:=161 TO 176 DO
  BEGIN
NUM:=DATA_ASCII[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA_ASCII[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
  END;
  NUM:=CHECKSUM MOD 16;
  TEXT1:=TOHEX1(NUM);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,':1094A000');
for n:=161 to 176 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+176-256;
FOR N:=177 TO 192 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;

FOR N:=177 TO 192 DO
BEGIN
NUM:=DATA_ASCII[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA_ASCII[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
END;

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,':1094B000');
for n:=177 to 192 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);

begin
write(Output_File,data_TEXT[n]);
end;

Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

CHECKSUM:=16+148+192-256;
FOR N:=193 TO 208 DO
BEGIN
CHECKSUM:=CHECKSUM+DATA_ASCII[N];
END;
CHECKSUM:=256-CHECKSUM;

FOR N:=193 TO 208 DO
BEGIN
NUM:=DATA_ASCII[N] MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=DATA_ASCII[N] DIV 16;
TEXT2:=TOHEX1(NUM);
DATA_TEXT[N]:=TEXT2+TEXT1;
END;

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,':1094C000');
for n:=193 to 208 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

BEGIN
    NUM:=DATA_ASCII[N] MOD 16;
    TEXT1:=TOHEX1(NUM);
    NUM:=DATA_ASCII[N] DIV 16;
    TEXT2:=TOHEX1(NUM);
    DATA_TEXT[N]:=TEXT2+TEXT1;
END;

NUM:=CHECKSUM MOD 16;
TEXT1:=TOHEX1(NUM);
NUM:=CHECKSUM DIV 16;
TEXT2:=TOHEX1(NUM);
CHECKSUM_TEXT:=TEXT2+TEXT1;

write(Output_File,':1094F000');
for n:=241 to 256 do
begin
write(Output_File,data_TEXT[n]);
end;
Write(Output_File,CHECKSUM_TEXT);
writeln(Output_File);

//////////B//////////

write(Output_File,':00000001FF');
closefile(Output_File);
end;

procedure TForm1.newquickClick(Sender:
TObject);
begin
    newclick(sender);
end;

procedure TForm1.openquickClick(Sender:
TObject);
begin
    openlick(sender);
end;

procedure TForm1.closequickClick(Sender:
TObject);
begin
    Closelick(sender);
end;

procedure TForm1.savequickClick(Sender:
TObject);
begin
    savefileclick(sender);
end;

procedure TForm1.exitquickClick(Sender:
TObject);
begin
    application.Terminate;
end;

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

                ORG 0000H
ADDRESS_LOW EQU 0F800H
ADDRESS_HIGH EQU 0F802H
DATAPORT EQU 0F801H
READ_WRITE EQU 0FC00H
CONTROL_PORT EQU 0FC01H
LOOP EQU 40H
TOTA EQU 41H
FIR EQU 42H
COUNTLOOP EQU 43H
CONSTANTLOOP EQU 45H
ENTER EQU 46H
DATA_X1 EQU 49H
DATA_X2 EQU 4AH
GUESS EQU 4BH
DELTA_X EQU 4CH
DELTA_Y EQU 4DH
D_ADDRESSL EQU 4EH
D_ADDRESSH EQU 4FH
D_BIT EQU 50H
D_COUNTER EQU 51H
DELAYTIME EQU 52H
LINE_X EQU 53H
LINE_XX EQU 54H
SAID EQU 55H
CON_X EQU 56H
CONT_X2 EQU 57H
CON_Y EQU 58H
CONT_Y2 EQU 59H
ALL1 EQU 60H
CUP EQU 61H
CHCK EQU 62H
DATA_Y1 EQU 63H
DATA_Y2 EQU 64H
DATA_Z1 EQU 65H
DATA_Z2 EQU 66H
DATA_ASCII EQU 67H
WIDE EQU 68H
DRIFT EQU 69H
SLIDE EQU 70H
DATA_X5 EQU 71H
STR EQU 72H
SPACE EQU 73H
TYPECODE EQU 74H
CON_M EQU 75H
CON_N EQU 76H
DATA_X4 EQU 77H
DATA_A EQU 78H
BEGIN:
MOV R0,#0FFH ;;delay:;
START0: MOV R1,#00H
DJNZ R1,$
DJNZ R0,START0
MOV DPTR,#0F803H ;;set port:;
MOV A,#80H
MOVX @DPTR,A
MOV DPTR,#0FC03H
MOVX @DPTR,A
MOV DPTR,#READ_WRITE
MOV A,#03H
MOVX @DPTR,A
LCALL DELRAM2
LCALL DELRAM0
LCALL DELRAM1
CLR P1.3
JNB P1.3,STL
MOV LOOP,#00H
STL:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CLR    P1.2
JNB    P1.2,KB
JMP    START_LOAD

KB:
MOV    A,#9BH
MOV    DPTR,#0FC03H
MOVX   @DPTR,A
MOV    ALL1,#00H
LCALL  DELRAMKEY
MOV    ALL1,#0FFH
MOV    A,#00110000B
        ;SET FUNCTION LCD
LCALL  LCDSET
LCALL  DELAYLCD
MOV    A,#00001111B
        ;DISPLAY CONTROL
LCALL  LCDSET
LCALL  DELAYLCD
MOV    A,#00010100B
        ;CURSOR SHIFT
LCALL  LCDSET
LCALL  DELAYLCD
MOV    A,#00000001B
        ;CLEAR DISPLAY
LCALL  LCDSET
LCALL  DELAYLCD
MOV    DRIFT,#0EH
MOV    WIDE,#80H

LCDCON:
MOV    A,WIDE
        ;BEGIN ADDRESS LCD
LCALL  LCDSET
LCALL  DELAYLCD
LCALL  SCAN_KEY

LCALL  DATA_OUT
LCALL  BACK
LCALL  CONVERT_VALUE
CJNE   A,#00H,LCDNO
JMP    NV

LCDNO:
MOV    DATA_ASCII,A
        ;;KEEP DATA FOR NEW DISPLAY;;
LCALL  COPY
        ;;copy to ram;;
INC    DRIFT
INC    WIDE
LCALL  LCDWRITE
LCALL  DELAYLCD
MOV    A,WIDE
DEC    A
CJNE   A,#0A0H,NV
        ;;no ,write continue;;
MOV    A,#00000001B
        ;;yes ,clear display;;
LCALL  LCDSET
LCALL  DELAYLCD
MOV    A,DATA_ASCII
        ;;WRITE DATA IN NEW DISPLAY;;
LCALL  LCDWRITE
LCALL  DELAYLCD
MOV    WIDE,#81H

NV:
CLR    P1.1
JB     P1.0,NV
MOV    R0,#0FFH    ;;delay;;
DEELCD1: MOV    R1,#00H
DJNZ   R1,S
DJNZ   R0,DEELCD1
MOV    R0,#0FFH    ;;delay;;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DEELCD2: MOV R1,#00H                                LCALL DELAYLCD
          DJNZ R1,$                                  RET
          DJNZ R0,DEELCD2                          ;;;;;;SUB FUNCTION DELAY ;;;;;;
          JMP LCDCON                                DELAYLCD: MOV R0,#0FFH
          ;;;;;;BACK; ;;;;;;                        LOOPDELAY:
BACK:                                          NOP
          MOV A,DATA_ASCII                          NOP
          CJNE A,#0EH,RET_BACK                      DJNZ R0,LOOPDELAY
          ;;check back pressed;;                 RET
          DEC DRIFT                                ;;;;;; DATA_OUT ;;;;;;
          DEC WIDE                                  DATA_OUT: MOV A,DATA_ASCII
          JMP NV                                    CJNE A,#1CH,DATA_IN
RET_BACK: ;;check OUT_DATA pressed;;
          RET                                        MOV A,#0DH
          ;;;;;;SCAN_KEY; ;;;;;;                  LCALL COPY
SCAN_KEY: CLR P1.0                                LCALL DELAYLCD
          SETB P1.1                                MOV R0,#0FFH ;;delay;;
KEY:      JNB P1.0,KEY                            START1: MOV R1,#00H
          MOV R0,#0FFH ;;delay;;                 DJNZ R1,$
DELCD1:  MOV R1,#00H                                DJNZ R0,START1
          DJNZ R1,$                                MOV DPTR,#0F803H
          DJNZ R0,DELCD1                          ;;set port;;
          MOV DPTR,#0FC02H                          MOV A,#80H
          MOVX A,@DPTR                              MOVX @DPTR,A
          MOV DATA_ASCII,A                        MOV DPTR,#0FC03H
          RET                                        MOVX @DPTR,A
          ;;;;;; SUB FUNCTION LCDSET ;;;;;;        MOV DPTR,#READ_WRITE
LCDSET:  MOV DPTR,#0FA00H                          MOV A,#03H
          MOVX @DPTR,A                              MOVX @DPTR,A
          LCALL DELAYLCD                          JMP YEAR
          RET                                        DATA_IN: RET
          ;;;;;; SUB FUNCTION LCDWRITE ;;;;;;      ;;;;;;COPY; ;;;;;;
LCDWRITE: MOV DPTR,#0FA02H                        COPY:
          MOVX @DPTR,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    DPH,#94H ;;copy to ram;;
MOV    DPL,DRIFT
MOVX   @DPTR,A
RET

;;;;;;;;;; CONVERT_VALUE ;;;;;;
CONVERT_VALUE:
MOV    A,DATA_ASCII
CJNE   A,#3AH,CONVERT
CA:    CLR    PI.1
JB     PI.0,CA
MOV    R0,#0FFH ;;delay;;
DEELCD3: MOV  R1,#00H
DJNZ   R1,$
DJNZ   R0,DEELCD3
MOV    R0,#0FFH ;;delay;;
DEELCD4: MOV  R1,#00H
DJNZ   R1,$
DJNZ   R0,DEELCD4
LCALL  SCAN_KEY
LCALL  DATA_OUT
LCALL  BACK
LCALL  CONVERT_CAP
CJNE   A,#00H,LCDVALE
JMP    NV
LCDVALE: JMP  LCDNO
RET

;;;;;;;;;; CONVERT ;;;;;;;;;;
NUMLCD:JMP  CHARLCD
CONVERT: MOV  DPTR,#TABLELCD
CLR    C
MOV    A,DATA_ASCII
SUBB   A,#01H
JC     NUMLCD
MOV    DATA_X4,A
MOV    DPH,#2DH ;;;
RET
MOV    A,#3DH ;;;
RET
MOV    A,#1BH ;;BACK SPACE;;
RET
MOV    A,#00H ;;TAB;;
RET
MOV    A,#30H ;;0;;
RET
MOV    A,#39H ;;9;;
RET
MOV    A,#37H ;;7;;
RET
MOV    A,#36H ;;6;;
RET
MOV    A,#35H ;;5;;
RET
MOV    A,#34H ;;4;;
RET
MOV    A,#33H ;;3;;
RET
MOV    A,#32H ;;2;;
RET
MOV    A,#'I' ;;1;;
RET
MOV    A,#00H ;;ESC;;
RET
RL     A
CLR    C
ADDC   A,DATA_X4
JMP    @A+DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV	A,#51H	::Q::	MOV	A,#46H	::F::
RET			RET		
MOV	A,#57H	::W::	MOV	A,#47H	::G::
RET			RET		
MOV	A,#45H	::E::	MOV	A,#48H	::H::
RET			RET		
MOV	A,#52H	::R::	MOV	A,#4AH	::J::
RET			RET		
MOV	A,#54H	::T::	MOV	A,#4BH	::K::
RET			RET		
MOV	A,#59H	::Y::	MOV	A,#4CH	::L::
RET			RET		
MOV	A,#55H	::U::	MOV	A,#3BH	::I::
RET			RET		
MOV	A,#49H	::I::	MOV	A,#27H	::G::
RET			RET		
MOV	A,#4FH	::O::	MOV	A,#00H	::N::
RET			RET		
MOV	A,#50H	::P::	MOV	A,#00H	::SHIFT::
RET			RET		
MOV	A,#5BH	::J::	MOV	A,#5CH	::K::
RET			RET		
MOV	A,#5DH	::L::	MOV	A,#5AH	::Z::
RET			RET		
MOV	A,#0DH	::ENTER::	MOV	A,#58H	::X::
RET			RET		
MOV	A,#00H	::CTRL::	MOV	A,#43H	::C::
RET			RET		
MOV	A,#'A'	::A::	MOV	A,#56H	::V::
RET			RET		
MOV	A,#53H	::S::	MOV	A,#42H	::B::
RET			RET		
MOV	A,#44H	::D::	MOV	A,#4EH	::N::
RET			RET		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV	A,#4DH	::M::	MOV	A,#69H	::F9::
RET			RET		
MOV	A,#2CH	::::	MOV	A,#6AH	::F10::
RET			RET		
MOV	A,#2EH	::::	MOV	A,#00H	::NUM::
RET			RET		
MOV	A,#2FH	::/::	MOV	A,#00H	::NO::
RET			RET		
MOV	A,#00H	::SHIFT::	MOV	A,#37H	::7::
RET			RET		
MOV	A,#2AH	::*::	MOV	A,#38H	::8::
RET			RET		
MOV	A,#00H	::ALT::	MOV	A,#39H	::9::
RET			RET		
MOV	A,#20H	::SPACE::	MOV	A,#2DH	::-::
RET			RET		
MOV	A,#00H	::CAPS LOCK::	MOV	A,#34H	::4::
RET			RET		
MOV	A,#61H	::F1::	MOV	A,#35H	::5::
RET			RET		
MOV	A,#62H	::F2::	MOV	A,#36H	::6::
RET			RET		
MOV	A,#63H	::F3::	MOV	A,#2BH	::+::
RET			RET		
MOV	A,#64H	::F4::	MOV	A,#31H	::1::
RET			RET		
MOV	A,#65H	::F5::	MOV	A,#32H	::2::
RET			RET		
MOV	A,#66H	::F6::	MOV	A,#33H	::3::
RET			RET		
MOV	A,#67H	::F7::	MOV	A,#30H	::0::
RET			RET		
MOV	A,#68H	::F8::	MOV	A,#2EH	::,::
RET			RET		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#6BH ;;F11;;
RET
MOV A,#6CH ;;F12;;
RET
CHARLCD: RET
          ;;;;;;;;;;CONVERT_CAP;;;;;;;;;
NUMLCD_CAP: JMP CHARLCD
CONVERT_CAP:
MOV DPTR,#TABLELCD_CAP
CLR C
MOV A,DATA_ASCII
SUBB A,#01H
JC NUMLCD_CAP
MOV DATA_X4,A
RL A
CLR C
ADDC A,DATA_X4
JMP @A+DPTR
TABLELCD_CAP: MOV A,#00H ;;ESC;;
RET
MOV A,#21H ;;!;;
RET
MOV A,#40H ;;@;;
RET
MOV A,#23H ;;#;;
RET
MOV A,#24H ;;$;;
RET
MOV A,#25H ;;%;;
RET
MOV A,#00H ;;^;;
RET
MOV A,#26H ;;&;
RET
MOV A,#2AH ;;*;;
RET
MOV A,#28H ;;(;;
RET
MOV A,#29H ;;);;
RET
MOV A,#5FH ;;_;;
RET
MOV A,#2BH ;;+;;
RET
MOV A,#1BH ;;BACK SPACE;;
RET
MOV A,#00H ;;TAB;;
RET
MOV A,#70H ;;Q;;
RET
MOV A,#77H ;;W;;
RET
MOV A,#65H ;;E;;
RET
MOV A,#72H ;;R;;
RET
MOV A,#74H ;;T;;
RET
MOV A,#79H ;;Y;;
RET
MOV A,#75H ;;U;;
RET
MOV A,#69H ;;I;;
RET
MOV A,#6FH ;;O;;
RET
MOV A,#70H ;;P;;
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV A,#7BH ;;t;;
RET
MOV A,#7DH ;;l;;
RET
MOV A,#0DH ;;ENTER;;
RET
MOV A,#00H ;;CTRL;;
RET
MOV A,#61H ;;A;;
RET
MOV A,#73H ;;S;;
RET
MOV A,#64H ;;D;;
RET
MOV A,#66H ;;F;;
RET
MOV A,#67H ;;G;;
RET
MOV A,#68H ;;H;;
RET
MOV A,#6AH ;;J;;
RET
MOV A,#6BH ;;K;;
RET
MOV A,#6CH ;;L;;
RET
MOV A,#3AH ;; ;;;
RET
MOV A,#22H ;;";;
RET
MOV A,#00H ;;~;;
RET
MOV A,#00H ;;SHIFT;;
RET
MOV A,#00H ;;;
RET
MOV A,#7AH ;;Z;;
RET
MOV A,#78H ;;X;;
RET
MOV A,#63H ;;C;;
RET
MOV A,#76H ;;V;;
RET
MOV A,#62H ;;B;;
RET
MOV A,#6EH ;;N;;
RET
MOV A,#6DH ;;M;;
RET
MOV A,#3CH ;;<;
RET
MOV A,#3EH ;;>;
RET
MOV A,#3FH ;;?;;
RET
MOV A,#00H ;;SHIFT;;
RET
MOV A,#2AH ;;*;;
RET
MOV A,#00H ;;ALT;;
RET
MOV A,#20H ;;SPACE;;
RET
MOV A,#00H
;;PRESS CAPS_LOCK AGAIN;;
RET
MOV A,#61H ;;F1;;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET		RET	
MOV	A,#62H ;;F2;;	MOV	A,#36H ;;6;;
RET		RET	
MOV	A,#63H ;;F3;;	MOV	A,#2BH ;;7;;
RET		RET	
MOV	A,#64H ;;F4;;	MOV	A,#31H ;;1;;
RET		RET	
MOV	A,#65H ;;F5;;	MOV	A,#32H ;;2;;
RET		RET	
MOV	A,#66H ;;F6;;	MOV	A,#33H ;;3;;
RET		RET	
MOV	A,#67H ;;F7;;	MOV	A,#30H ;;0;;
RET		RET	
MOV	A,#68H ;;F8;;	MOV	A,#2EH ;;;;
RET		RET	
MOV	A,#69H ;;F9;;	MOV	A,#6BH ;;F11;;
RET		RET	
MOV	A,#6AH ;;F10;;	MOV	A,#6CH ;;F12;;
RET		RET	
MOV	A,#00H ;;NUM;;	RET	
RET		RET	
MOV	A,#00H ;;NO;;	START_LOAD:	COMPUTER
RET		MOV	A,#37H ;;7;;
MOV	A,#37H ;;7;;	CLR	P1.3
RET			;;P1.3=DATA NOT END;;
MOV	A,#38H ;;8;;	JNB	P1.3,BEGIN_SHOW
RET		CALL	LOAD
MOV	A,#39H ;;9;;		
RET			
MOV	A,#2DH ;;;;		DATA_PICTURE
RET		DATA_PIC:	MOV R0,#80H
MOV	A,#34H ;;4;;		;;ADDRESS AT PROGRAM OPD;;
RET		MOV	R1,#00H
MOV	A,#35H ;;5;;	MOV	R2,#90H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;ADDRESS-10 AT BUFFER DATA;;
MOV R3,#09H
MOV R4,LOOP
ADUP: MOV A,#10H
ADD A,R2
MOV R2,A
DJNZ R4,ADUP
LEOPIC: MOV DPH,R0
MOV DPL,R1
MOVX A,@DPTR
MOV DPH,R2
MOVX @DPTR,A
DJNZ R1,LEOPIC
INC R0
INC R2
DJNZ R3,LEOPIC
::: DATA_TEXT :::
DATA_TEXT: MOV R0,#94H
;;ADDRESS AT PROGRAM OPD;;
MOV R1,#00H
MOV R2,#0EFH
;;ADDRESS-10 AT BUFFER DATA;;
MOV R4,LOOP
ADUPTXT: INC R2
DJNZ R4,ADUPTXT
LEOTEXT: MOV DPH,R0
MOV DPL,R1
MOVX A,@DPTR
MOV DPH,R2
MOVX @DPTR,A
DJNZ R1,LEOTEXT
JMP CAL_LOAD
::: DATA_OPD_PIC :::
BEGIN_SHOW:
MOV CONSTANTLOOP,COUNTLOOP
MOV COUNTLOOP,#01H
NOT_LOAD: MOV A,COUNTLOOP
DEC A
CJNE A,CONSTANTLOOP,DATA_OPD_PIC
MOV COUNTLOOP,#01H
DATA_OPD_PIC: MOV R0,#80H
;;ADDRESS AT PROGRAM OPD;;
MOV R1,#00H
MOV R2,#90H
;;ADDRESS-10 AT BUFFER DATA;;
MOV R3,#09H
MOV R4,COUNTLOOP
ADUPOP: MOV A,#10H
ADD A,R2
MOV R2,A
DJNZ R4,ADUPOP
LEOPICOPD: MOV DPH,R2
MOV DPL,R1
MOVX A,@DPTR
MOV DPH,R0
MOVX @DPTR,A
DJNZ R1,LEOPICOPD
INC R0
INC R2
DJNZ R3,LEOPICOPD
::: DATA_OPD_TEXT :::
DATA_OPD_TEXT: MOV R0,#94H
;;ADDRESS AT PROGRAM OPD;;
MOV R1,#00H
MOV R2,#0EFH
;;ADDRESS-10 AT BUFFER DATA;;
MOV R3,#01H
MOV R4,COUNTLOOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ADUPTEXTOPD:
    INC    R2
    DJNZ   R4,ADUPTEXTOPD
LEOTEXTOPD:
    MOV    DPH,R2
    MOV    DPL,R1
    MOVX   A,@DPTR
    MOV    DPH,R0
    MOVX   @DPTR,A
    DJNZ   R1,LEOTEXTOPD
    DJNZ   R3,LEOTEXTOPD
    INC    COUNTLOOP
    ;;;;;;PICTURE DISPLAY;;;;;;
    MOV    CHCK,#01H
FOREVER:
    LCALL  DELRAM0
    MOV    A,CHCK
    CJNE   A,#01H,DINO3
    LCALL  DELRAM1
    JMP    DINO4
DINO3:   LCALL  DELRAM2
DINO4:   MOV    ALL1,#0FFH
    ;;set initial variable;;
    MOV    LOOP,#01H
    MOV    FIR,#80H
    MOV    STR,#00H
    MOV    WIDE,#00H
    MOV    SLIDE,#00H
    MOV    DRIFT,#00H
    MOV    TOTA,#11H
LPG:     MOV    DPH,FIR
    ;;write picture 1;;
    MOV    DPL,STR
    MOVX   A,@DPTR
    MOV    DATA_X1,A
    INC    DPL
    MOVX   A,@DPTR
    MOV    DATA_Y1,A
    INC    DPL
    MOVX   A,@DPTR
    MOV    DATA_X2,A
    INC    DPL
    MOVX   A,@DPTR
    MOV    DATA_Y2,A
    INC    DPL
    MOVX   A,@DPTR
    MOV    TYPECODE,A
    MOV    STR,DPL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL NEW
MOV DPH,FIR;;write picture 3;;
MOV DPL,STR
INC DPL
MOVX A,@DPTR
MOV DATA_X1,A
INC DPL
MOVX A,@DPTR
MOV DATA_Y1,A
INC DPL
MOVX A,@DPTR
MOV DATA_X2,A
INC DPL
MOVX A,@DPTR
MOV DATA_Y2,A
INC DPL
MOVX A,@DPTR
MOV TYPECODE,A
MOV STR,DPL
LCALL NEW
INC STR
MOV A,LOOP
CJNE A,#01H,DOY
;;if loop 1 and row 1 keep data to cup;;
MOV A,TOTA
CJNE A,#11H,DOY
MOV DPH,FIR
MOV DPL,STR
MOVX A,@DPTR
CJNE A,#0AAH,COOL
;;if data=AA jump to next picture;;
JMP COO
COOL: MOV CUP,A
INC CUP ;INC CUP
DOY: INC STR
DINZ TOTA,LDPG
LDPG: JMP LPG
COO: LCALL DELRAM0
;;next shift/any picture;;
INC LOOP ;;add round;;
MOV TOTA,#11H
;;set new variable;;
MOV STR,#00H
MOV A,CHCK
CJNE A,#01H,CK
;;swap ram 1/2;;
MOV CHCK,#02H
LCALL DELRAM2
JMP CLK2
CK: MOV CHCK,#01H
LCALL DELRAM
CLK2: DJNZ CUP,FI
JMP SEC
FI: MOV A,CHCK
CJNE A,#01H,SHOW2
LCALL DISPLAY2
JMP LPG
SHOW2: LCALL DISPLAY1
JMP LPG
SEC: LCALL DELRAM0
MOV A,CHCK
CJNE A,#01H,DINO1
LCALL DELRAM1
JMP DINO2
DINO1: LCALL DELRAM2
DINO2: MOV LOOP,#01H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;;set initial variable next sequence;;
MOV STR,#00H
MOV TOTA,#11H
INC FIR ;;next sequence;;
MOV A,FIR
CJNE A,#89H,LDPG
;;check last sequence;;
MOV DATA_X5,#32H
CLR P1.3
JNB P1.3,OLD_DATA
JMP NOT_LOAD
;;;;;;;;;TEXT DISPLAY;;;;;;;;;
YEAR: MOV LOOP,#01H
MOV CHCK,#01H
MOV GUESS,#00H
LCALL DELRAM2
LCALL DELRAM0
LCALL DELRAM1
MOV ENTER,#00H
GUE: MOV SPACE,#01H
TONG: MOV WIDE,#00H
;;set initial variable;;
MOV SLIDE,#00H
MOV DRIFT,#00H
MOV CUP,#00H
TXTX: MOV STR,#00H
MOV FIR,#10H
;;FIR is fixed variable;;
MOV DATA_X5,LOOP
MOV A,DATA_X5
CJNE A,#01H,KAN
;;check loop if =1 jump to write text;;
MOV DATA_X5,#08H
JMP TXT

KAN: DEC DATA_X5
KANTI: MOV A,STR ;;next text suit;;
ADD A,#08H
MOV STR,A
DJNZ DATA_X5,KANTI
MOV DATA_X5,#08H
TXT: MOV DPH,#94H ;;write text;;
MOV DPL,STR
MOVX A,@DPTR
MOV DATA_ASCII,A
CJNE A,#0DH,NOHAVE
MOV ENTER,#0DH
NOHAVE: LCALL SETTEXT
INC STR
DJNZ DATA_X5,TXT
INC DATA_X5
MOV A,CUP
;;cup is last position;;
CJNE A,#00H,NINEY
MOV CUP,WIDE
NINEY: DJNZ FIR,TXT
;;check write text=16;;
LCALL DELRAM0
MOV A,CHCK
CJNE A,#02H,OTT
LCALL DISPLAY2
LCALL DELRAM1
DEC DRIFT ;;shif 4 column;;
DEC DRIFT
DEC DRIFT
DEC DRIFT
MOV WIDE,#00H
MOV CHCK,#01H
DJNZ CUP,BUFFER

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        INC     CUP                MOV     SPACE,#02H
BUFFER1:                JMP     TON
        DJNZ   CUP,BUFFER3        OUT:    MOV     ENTER,#00H
        INC     CUP                MOV     SPACE,#01H
BUFFER3:                CLR     P1.3
        DJNZ   CUP,BUFFER5        JNB    P1.3,OLD_TEXT
        INC     CUP                JMP     NOT_LOAD
BUFFER5:                OLD_TEXT:
        DJNZ   CUP,TXTX          MOV     A,GUESS
        JMP     YUT                MOV     GUESS,#05H
OTT:    LCALL  DISPLAY1          CJNE   A,#05H,SUBGUE
        LCALL  DELRAM2          CLR     P1.2
        DEC    DRIFT    ;;shift 4 column;; JNB    P1.2,SUBYEAR
        DEC    DRIFT                JMP     NOT_LOAD    ;;yes;;
        DEC    DRIFT                SUBGUE: JMP     GUE
        DEC    DRIFT                SUBYEAR:
        MOV    WIDE,#00H          JMP     YEAR
        MOV    CHCK,#02H        TON:    JMP     TONG    ;;no;;
        DJNZ   CUP,BUFFER2        OIL:   JMP     TXTX
        INC     CUP                ;;::::::::::SETTEXT::::::::::
BUFFER2:                RASA:   JMP     TITI
        DJNZ   CUP,BUFFER4        SETTEXT:
        INC     CUP                MOV     DPTR,#SARA
BUFFER4:                CLR     C
        DJNZ   CUP,BUFFER6        MOV     A,DATA_ASCII
        INC     CUP                SUBB   A,#0E0H
BUFFER6:                JC     RASA
        DJNZ   CUP,OIL          RL     A    ;;*2;;
YUT:    INC     LOOP            MOV     DATA_X4,A
        MOV    A,SPACE          ADD    A,DATA_X4
        CJNE  A,#02H,ENT        JMP    @A+DPTR
        JMP    OUT                SARA:  LCALL  TH46    ;;E0=A;;
ENT:    MOV    A,ENTER          RET
        CJNE  A,#0dH,TON        LCALL  TH47    ;;E1=AA;;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
LCALL TH49 ;;E2=0;;
RET
LCALL TH50 ;;E3=I_MON;;
RET
LCALL TH48 ;;E4=I_MARA;;
RET
LCALL TH69 ;;E5=R_LONG;;
RET
LCALL TH65 ;;E6=YAMUG;;
RET
LCALL TH61 ;;E7=TAIKU;;
RET
LCALL TH62 ;;E8=EAK;;
RET
LCALL TH63 ;;E9=TO;;
RET
LCALL TH60 ;;EA=TEE;;
RET
LCALL TH59 ;;EB=JATTAVA;;
RET
LCALL TH64 ;;EC=KARAN;;
RET
LAND: JMP TABPIC
TITI: MOV DPTR,#THI
CLR C
MOV A,DATA_ASCII
SUBB A,#0A1H
JC LAND
RL A
MOV DATA_X4,A
ADD A,DATA_X4
JMP @A+DPTR
THI: LCALL TH01
RET
LCALL TH02
RET
LJMP CHAR
RET
LCALL TH03
RET
LCALL TH04
RET
LCALL TH05
RET
LCALL TH06
RET
LCALL TH07
RET
LCALL TH08
RET
LCALL TH09
RET
LCALL TH10
RET
LCALL TH11
RET
LCALL TH12
RET
LCALL TH13
RET
LCALL TH14
RET
LCALL TH15
RET
LCALL TH16
RET
LCALL TH17

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET		RET	
LCALL	TH18	LCALL	TH67
RET		RET	
LCALL	TH19	LCALL	TH35
RET		RET	
LCALL	TH20	LCALL	TH68
RET		RET	
LCALL	TH21	LCALL	TH36
RET		RET	
LCALL	TH22	LCALL	TH37
RET		RET	
LCALL	TH23	LCALL	TH38
RET		RET	
LCALL	TH24	LCALL	TH39
RET		RET	
LCALL	TH25	LCALL	TH40
RET		RET	
LCALL	TH26	LCALL	TH41
RET		RET	
LCALL	TH27	LCALL	TH42
RET		RET	
LCALL	TH28	LCALL	TH43
RET		RET	
LCALL	TH29	LCALL	TH66
RET		RET	
LCALL	TH30	LCALL	TH45
RET		RET	
LCALL	TH31	LCALL	TH58
RET		RET	
LCALL	TH32	LCALL	TH44
RET		RET	
LCALL	TH33	LCALL	TH51
RET		RET	
LCALL	TH34	LCALL	TH52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

RET
LCALL TH53
RET
LCALL TH54
RET
LCALL TH55
RET
LCALL TH56
RET
LCALL TH57
RET
NUMBER: JMP TAB
TABPIC: MOV DPTR,#TABLEPIC
CLR C
MOV A,DATA_ASCII
SUBB A,#61H
JC NUMBER
RL A
MOV DATA_X4,A
CLR C
ADDC A,DATA_X4
JMP @A+DPTR
TABLEPIC:
LCALL MK24
RET
LCALL MK25
RET
LCALL MK26
RET
LCALL MK27
RET
LCALL MK28
RET
LCALL MK29
RET
LCALL MK30
RET
LCALL MK31
RET
LCALL MK32
RET
LCALL MK33
RET
LCALL MK34
RET
LCALL MK35
RET
LCALL MK36
RET
LCALL MK37
RET
NUMTE: JMP CHAR
TAB: MOV DPTR,#TABLE
CLR C
MOV A,DATA_ASCII
SUBB A,#20H
JC NUMTE
RL A
MOV DATA_X4,A
CLR C
ADDC A,DATA_X4
JMP @A+DPTR
TABLE:
LCALL ABC
RET
LCALL MK1
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCALL MK20	LCALL N30
RET	RET
LCALL MK12	LCALL N40
RET	RET
LCALL MK22	LCALL N50
RET	RET
LCALL MK11	LCALL N60
RET	RET
LCALL MK21	LCALL N70
RET	RET
LCALL MK18	LCALL N80
RET	RET
LCALL MK9	LCALL N90
RET	RET
LCALL MK10	LCALL MK16
RET	RET
LCALL MK23	LCALL MK15
RET	RET
LCALL MK4	LCALL MK8
RET	RET
LCALL MK14	LCALL MK6
RET	RET
LCALL MK5	LCALL MK7
RET	RET
LCALL MK17	LCALL MK2
RET	RET
LCALL MK3	LCALL MK26 @@
RET	RET
LCALL N00	LCALL AA0
RET	RET
LCALL N10	LCALL BB0
RET	RET
LCALL N20	LCALL CC0
RET	RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCALL DD0	LCALL UU0
RET	RET
LCALL EE0	LCALL VV0
RET	RET
LCALL FF0	LCALL WW0
RET	RET
LCALL GG0	LCALL XX0
RET	RET
LCALL HH0	LCALL YY0
RET	RET
LCALL II0	LCALL ZZ0
RET	RET
LCALL JJ0	CHAR: LCALL ABC
RET	RET
LCALL KK0	NEW: NEW
RET	NEW: MOV A,TYPECODE
LCALL LL0	LNE: CJNE A,#01H,DON
RET	LCALL UP
LCALL MM0	JMP SH
RET	DON: CJNE A,#02H,LET
LCALL NN0	LCALL DOWN
RET	JMP SH
LCALL OO0	LET: CJNE A,#03H,RIT
RET	LCALL LEFT
LCALL PP0	JMP SH
RET	RIT: CJNE A,#04H,NOR
LCALL QQ0	LCALL RIGHT
RET	JMP SH
LCALL RR0	NOR: CJNE A,#07H,BLK
RET	LCALL LINE
LCALL SS0	JMP SH
RET	BLK: CJNE A,#05H,ONE
LCALL TT0	LCALL BLINK
RET	JMP SH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ONE:   CJNE   A,#13H,TWO           LCALL  X_Y_2
       LCALL  X_Y_
       JMP    SH
TWO:   CJNE   A,#0FH,THREE        LCALL  X_Y_2
       LCALL  X_Y_
       JMP    SH
THREE: CJNE   A,#11H,FOUR        LCALL  SQUARE
       LCALL  X_Y_
       JMP    SH
FOUR:  CJNE   A,#15H,LNE2        LCALL  EXTEND
       LCALL  X_Y_
       JMP    SH
LNE2:  CJNE   A,#08H,DON2        LCALL  SMALL
       LCALL  UP2
       JMP    SH
DON2:  CJNE   A,#09H,LET2        LCALL  EXTENDSQU
       LCALL  DOWN2
       JMP    SH
LET2:  CJNE   A,#0AH,RIT2        LCALL  SMALLSQU
       LCALL  LEFT2
       JMP    SH
RIT2:  CJNE   A,#0BH,BLK2        LCALL  RIGHT2
       LCALL  RIGHT2
       JMP    SH
BLK2:  CJNE   A,#0CH,ONE2        LCALL  BLINK2
       LCALL  BLINK2
       JMP    SH
ONE2:  CJNE   A,#14H,TWO2        LCALL  X_Y_2
       LCALL  X_Y_2
       JMP    SH
TWO2:  CJNE   A,#10H,THREE2      LCALL  X_Y_2
       LCALL  X_Y_2
       JMP    SH
THREE2: CJNE   A,#12H,FOUR2      LCALL  SQUARE
       LCALL  SQUARE
       JMP    SH
FOUR2: CJNE   A,#16H,SQU2       LCALL  X_Y_2
       LCALL  X_Y_2
       JMP    SH
SQU2:  CJNE   A,#0EH,EX         LCALL  SQUARE
       LCALL  SQUARE
       JMP    SH
EX:    CJNE   A,#17H,SMA        LCALL  EXTEND
       LCALL  EXTEND
       JMP    SH
SMA:   CJNE   A,#18H,EXSQ       LCALL  SMALL
       LCALL  SMALL
       JMP    SH
EXSQ:  CJNE   A,#19H,SMASQ      LCALL  EXTENDSQU
       LCALL  EXTENDSQU
       JMP    SH
SMASQ: CJNE   A,#1AH,SH        LCALL  SMALLSQU
       LCALL  SMALLSQU
SH:    RET
      ;;;;;;;;;;EXTEND;;;;;;;;;;
EXTEND: MOV    R2,LOOP
DG1:   DEC    DATA_X1
      INC    DATA_X2
      INC    DATA_Y1
      DEC    DATA_Y2
      MOV    A,DATA_X1
      JZ    DOG1
      MOV    A,DATA_Y2
      JZ    DOG1
      DJNZ  R2,DG1
      LCALL  SQUARE
DOG1:  RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

          ;;;;;;;;;SMALL;;;;;;;;;
SMALL:  MOV    R2,LOOP
DG2:    INC    DATA_X1
        DEC    DATA_X2
        DEC    DATA_Y1
        INC    DATA_Y2
        MOV    A,DATA_X2
        JZ     DOG2
        MOV    A,DATA_Y1
        JZ     DOG2
        DJNZ  R2,DG2
        LCALL SQUARE
DOG2:   RET
          ;;;;;;;;;EXTENDSQU;;;;;;;;;
EXTENDSQU:
        MOV    R2,LOOP
DG3:    DEC    DATA_X1
        INC    DATA_X2
        INC    DATA_Y1
        DEC    DATA_Y2
        MOV    A,DATA_X1
        JZ     DOG3
        MOV    A,DATA_Y2
        JZ     DOG3
        DJNZ  R2,DG3
        LCALL LINE
DOG3:   RET
          ;;;;;;;;;SMALLSQU;;;;;;;;;
SMALLSQU:
        MOV    R2,LOOP
D4:     INC    DATA_X1
        DEC    DATA_X2
        DEC    DATA_Y1
        INC    DATA_Y2
        MOV    A,DATA_X2
        JZ     DOG4
        MOV    A,DATA_Y1
        JZ     DOG4
        DJNZ  R2,D4
        LCALL LINE
DOG4:   RET
          ;;;;;;;;;BLINK;;;;;;;;;
BLINK:  MOV    A,CHCK
        CJNE  A,#01H,KK1
        MOV    ALL1,#00H
        LCALL LINE
        MOV    ALL1,#0FFH
        RET
KK1:    LCALL LINE
        RET
          ;;;;;;;;;X+Y;;;;;;;;;
X_Y_:   MOV    R2,LOOP
BG1:    INC    DATA_X1
        INC    DATA_X2
        INC    DATA_Y1
        INC    DATA_Y2
        DJNZ  R2,BG1
        LCALL LINE
        RET
          ;;;;;;;;;X+Y-;;;;;;;;;
X_Y_-:  MOV    R2,LOOP
BG2:    INC    DATA_X1
        INC    DATA_X2
        DEC    DATA_Y1
        DEC    DATA_Y2
        MOV    A,DATA_Y1
        JZ     DOG5
        MOV    A,DATA_Y2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

JZ      DOG5
DGNZ   R2,BG2
LCALL  LINE
DOG5:  RET
.....X-Y+.....
X_Y_:  MOV    R2,LOOP
BG3:   DEC    DATA_X1
DEC    DATA_X2
INC    DATA_Y1
INC    DATA_Y2
MOV    A,DATA_X1
JZ     DOG6
MOV    A,DATA_X2
JZ     DOG6
DGNZ   R2,BG3
LCALL  LINE
DOG6:  RET
.....X-Y.....
X_Y_:  MOV    R2,LOOP
BG4:   DEC    DATA_X1
DEC    DATA_X2
DEC    DATA_Y1
DEC    DATA_Y2
MOV    A,DATA_X1
JZ     DOG7
MOV    A,DATA_Y2
JZ     DOG7
MOV    A,DATA_X2
JZ     DOG7
MOV    A,DATA_Y1
JZ     DOG7
DGNZ   R2,BG4
LCALL  LINE
DOG7:  RET
JZ     DOG5
DGNZ   R2,BG2
LCALL  LINE
UP:    MOV    R2,LOOP
UL:    INC    DATA_Y1
INC    DATA_Y2
DGNZ   R2,UL
LCALL  LINE
RET
.....DOWN.....
DOWN:  MOV    R2,LOOP
DWN:   DEC    DATA_Y1
DEC    DATA_Y2
MOV    A,DATA_Y1
JZ     DOG8
MOV    A,DATA_Y2
JZ     DOG8
DGNZ   R2,DWN
LCALL  LINE
DOG8:  RET
.....RIGHT.....
RIGHT: MOV    R2,LOOP
RTG:   INC    DATA_X1
INC    DATA_X2
DGNZ   R2,RTG
LCALL  LINE
RET
.....LEFT.....
LEFT:  MOV    R2,LOOP
LTF:   DEC    DATA_X1
MOV    A,DATA_X1
JZ     DOG9
DEC    DATA_X2
MOV    A,DATA_X2
JZ     DOG9
DGNZ   R2,LTF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL  LINE
DOG9:  RET
        .....BLINK2;.....
BLINK2: MOV  A,CHCK
        CJNE  A,#01H,KK2
        MOV  ALL1,#00H
        LCALL SQUARE
        MOV  ALL1,#0FFH
        RET
KK2:   LCALL  SQUARE
        RET
        .....X+Y+2;.....
X_Y_2: MOV  R2,LOOP
CG1:   INC   DATA_X1
        INC   DATA_X2
        INC   DATA_Y1
        INC   DATA_Y2
        DJNZ  R2,CG1
        LCALL SQUARE
        RET
        .....X+Y-2;.....
X_Y_2: MOV  R2,LOOP
CG2:   INC   DATA_X1
        INC   DATA_X2
        DEC   DATA_Y1
        DEC   DATA_Y2
        MOV  A,DATA_Y1
        JZ   DOG10
        MOV  A,DATA_Y2
        JZ   DOG10
        DJNZ  R2,CG2
        LCALL SQUARE
DOG10: RET
        .....X-Y+2;.....
X_Y_2: MOV  R2,LOOP
CG3:   DEC   DATA_X1
        DEC   DATA_X2
        INC   DATA_Y1
        INC   DATA_Y2
        MOV  A,DATA_X1
        JZ   DOG11
        MOV  A,DATA_X2
        JZ   DOG11
        DJNZ  R2,CG3
        LCALL  SQUARE
DOG11: RET
        .....X-Y-2;.....
X_Y_2: MOV  R2,LOOP
CG4:   DEC   DATA_X1
        DEC   DATA_X2
        DEC   DATA_Y1
        DEC   DATA_Y2
        MOV  A,DATA_X1
        JZ   DOG12
        MOV  A,DATA_Y2
        JZ   DOG12
        MOV  A,DATA_X2
        JZ   DOG12
        MOV  A,DATA_Y1
        JZ   DOG12
        DJNZ  R2,C4
        LCALL  SQUARE
DOG12: RET
        .....UP2;.....
UP2:   MOV  R2,LOOP
CG5:   INC   DATA_Y1
        INC   DATA_Y2
        DJNZ  R2,C5

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        LCALL  SQUARE                SUBB   A,DATA_X2
        RET                               JC    LI
        .....DOWN2.....
DOWN2: MOV    R2,LOOP
C6:   DEC    DATA_Y1
      DEC    DATA_Y2
      MOV    A,DATA_Y1
      JZ     DOG13
      MOV    A,DATA_Y2
      JZ     DOG13
      DJNZ   R2,C6
      LCALL  SQUARE
DOG13: RET
        .....RIGHT2.....
RIGHT2: MOV    R2,LOOP
C7:   INC    DATA_X1
      INC    DATA_X2
      DJNZ   R2,C7
      LCALL  SQUARE
      RET
        .....LEFT2.....
LEFT2: MOV    R2,LOOP
C8:   DEC    DATA_X1
      DEC    DATA_X2
      MOV    A,DATA_X1
      JZ     DOG14
      MOV    A,DATA_X2
      JZ     DOG14
      DJNZ   R2,C8
      LCALL  SQUARE
DOG14: RET
        .....LINE .....
LINE:  CLR    C
      MOV    A,DATA_X1
        SUBB   A,DATA_X2
        JC    LI
        MOV    DATA_Z1,DATA_X1
        MOV    DATA_Z2,DATA_Y1
        MOV    DATA_X1,DATA_X2
        MOV    DATA_Y1,DATA_Y2
        MOV    DATA_X2,DATA_Z1
        MOV    DATA_Y2,DATA_Z2
        LI:   CLR    C
        MOV    A,DATA_Y2
        SUBB   A,DATA_Y1
        JC    NEG
        CLR    C
        MOV    A,DATA_X2    ;X2-X1
        SUBB   A,DATA_X1
        MOV    DELTA_X,A
        CLR    C
        MOV    A,DATA_Y2    ;Y2-Y1
        SUBB   A,DATA_Y1
        MOV    DELTA_Y,A
        CLR    C
        MOV    A,DELTA_Y
        SUBB   A,DELTA_X
        JZ     LINE1
        JC    LINE3
        JMP    LINE4
        NEG:  CLR    C
        MOV    A,DATA_X2    ;X2-X1
        SUBB   A,DATA_X1
        MOV    DELTA_X,A
        CLR    C
        MOV    A,DATA_Y1    ;Y2-Y1
        SUBB   A,DATA_Y2
        MOV    DELTA_Y,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

      CLR      C
      MOV      A,DELTA_Y
      CJNE     A,DELTA_X,NN
      JMP      LINE2
NN:    JC       LINE5
      JMP      LINE6

      ;;;;;;;;;;;;;;;;; LINE |M| < 1 ;;;;;;;;;;;;;;;;;
LINE5: MOV      R1,DATA_X1
      MOV      R2,DATA_Y1
      CALL     SETBIT
      INC      DELTA_Y
      MOV      A,DELTA_Y
      MOV      B,A
      MOV      A,DELTA_X
      DIV      AB
      MOV      SAID,B
      MOV      LINE_XX,A
      MOV      LINE_X,A
AC1:   INC      R1
      CALL     SETBIT
      DJNZ     LINE_XX,AC1
      DEC      R2
      MOV      LINE_XX,LINE_X
      DJNZ     DELTA_Y,AC1
      INC      SAID
      DEC      R1
      INC      R2
AC2:   INC      R1
      CALL     SETBIT
      DJNZ     SAID,AC2
      RET

      ;;;;;;;;;;;;;;;;; LINE M=1 ;;;;;;;;;;;;;;;;;
LINE1: MOV      R1,DATA_X1
      MOV      R2,DATA_Y1
      INC      DELTA_X
      LIN1:   LCALL  SETBIT
      INC      R1
      INC      R2
      DJNZ     DELTA_X,LIN1
      RET

      ;;;;;;;;;;;;;;;;; LINE M=-1 ;;;;;;;;;;;;;;;;;
LINE2: MOV      R1,DATA_X1
      MOV      R2,DATA_Y1
      INC      DELTA_X
      LIN2:   LCALL  SETBIT
      INC      R1
      DEC      R2
      DJNZ     DELTA_X,LIN2
      RET

      ;;;;;;;;;;;;;;;;; LINE M<=0 ;;;;;;;;;;;;;;;;;
LINE3: MOV      R1,DATA_X1
      MOV      R2,DATA_Y1
      CALL     SETBIT
      INC      DELTA_Y
      MOV      A,DELTA_Y
      MOV      B,A
      MOV      A,DELTA_X
      DIV      AB
      MOV      SAID,B
      MOV      LINE_XX,A
      MOV      LINE_X,A
ABI:   INC      R1
      CALL     SETBIT
      DJNZ     LINE_XX,ABI
      INC      R2
      MOV      LINE_XX,LINE_X
      DJNZ     DELTA_Y,ABI
      INC      SAID

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    DEC    R1
    DEC    R2
AB2: INC    R1
    CALL  SETBIT
    DJNZ  SAID,AB2
    RET

    ;;;;LINE M>1 M= infinity ;;;;;;
LINE4: MOV    R1,DATA_X1
    MOV    R2,DATA_Y1
    CALL  SETBIT
    INC    DELTA_X
    MOV    A,DELTA_X
    MOV    B,A
    MOV    A,DELTA_Y
    DIV   AB
    MOV    SAID,B
    MOV    LINE_XX,A
    MOV    LINE_X,A
AB3: INC    R2
    CALL  SETBIT
    DJNZ  LINE_XX,AB3
    INC    R1
    MOV    LINE_XX,LINE_X
    DJNZ  DELTA_X,AB3
    INC    SAID
    DEC    R2
    DEC    R1
AB4: INC    R2
    CALL  SETBIT
    DJNZ  SAID,AB4
    RET

    ;;;;LINE |M| > 1 ;;;;;;
LINE6: MOV    R1,DATA_X1
    MOV    R2,DATA_Y1
    CALL  SETBIT
    INC    DELTA_X
    MOV    A,DELTA_X
    MOV    B,A
    MOV    A,DELTA_Y
    DIV   AB
    MOV    SAID,B
    MOV    LINE_XX,A
    MOV    LINE_X,A
    INC    R2
    DEC    R1
    DEC    R2
    CALL  SETBIT
    DJNZ  LINE_XX,AC3
    INC    R1
    MOV    LINE_XX,LINE_X
    DJNZ  DELTA_X,AC3
    INC    SAID
    INC    R2
    DEC    R1
    DEC    R2
    CALL  SETBIT
    DJNZ  SAID,AC4
    RET

    ;;;; SQUARE ;;;;;;
SQUARE: CLR    C
    MOV    A,DATA_X1
    SUBB   A,DATA_X2
    JC     SQ0
    MOV    DATA_Z1,DATA_X1
    MOV    DATA_Z2,DATA_Y1
    MOV    DATA_X1,DATA_X2
    MOV    DATA_Y1,DATA_Y2
    MOV    DATA_X2,DATA_Z1
    MOV    DATA_Y2,DATA_Z2
    MOV    C,DATA_Z1
    MOV    C,DATA_Z2
SQ0: CLR    C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

INC      R0
DJNZ    R3,LEO
RET

;;;;;;;;; DEL RAM1 ;;;;;;;;;;
DELRAM1:
MOV     DPTR,#DATAPORT
MOV     A,ALL1
MOVX    @DPTR,A
MOV     DPTR,#CONTROL_PORT
MOV     A,#05H
MOVX    @DPTR,A
MOV     R0,#04H
MOV     R1,#00H
FILL5:  MOV     DPTR,#ADDRESS_HIGH
MOV     A,R0
MOVX    @DPTR,A
FILL6:  MOV     DPTR,#ADDRESS_LOW
MOV     A,R1
MOVX    @DPTR,A
CALL    WRITE1
DJNZ    R1,FILL6
DJNZ    R0,FILL5
RET

;;;;;;;;; DEL RAM2 ;;;;;;;;;;
DELRAM2:
MOV     DPTR,#CONTROL_PORT
MOV     A,#0AH
MOVX    @DPTR,A
MOV     DPTR,#DATAPORT
MOV     A,ALL1
MOVX    @DPTR,A
MOV     R0,#04H
MOV     R1,#00H
FILL3:  MOV     DPTR,#ADDRESS_HIGH
MOV     A,R0
MOVX    @DPTR,A
MOV     DPTR,#CONTROL_PORT
MOV     A,#0aH
MOVX    @DPTR,A
MOV     R2,DELAYTIME
CALL    DELAY
RET

;;;;;;;;; DISPLAY1 ;;;;;;;;;;
DISPLAY1:
MOV     DPTR,#CONTROL_PORT
MOV     A,#0aH
MOVX    @DPTR,A
MOV     R2,DELAYTIME
CALL    DELAY
RET

;;;;;;;;; DISPLAY2 ;;;;;;;;;;
DISPLAY2:
MOV     DPTR,#CONTROL_PORT
MOV     A,#05H
MOVX    @DPTR,A
MOV     R2,DELAYTIME
CALL    DELAY
RET

;;;;;;;;; WRITE RAM ;;;;;;;;;;
WRITE2:  MOV     DPTR,#READ_WRITE
MOV     A,#01H
MOVX    @DPTR,A
MOV     A,#03H
MOVX    @DPTR,A
RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

..... WRITE RAM .....
WRITE1: MOV DPTR,#READ_WRITE
        MOV A,#02H
        MOVX @DPTR,A
        MOV A,#03H
        MOVX @DPTR,A
        RET

..... DELAY TIME .....
DELAY: MOV R2,DELAYTIME
        MOV R3,DELAYTIME
        MOV R4,DELAYTIME
        MOV R5,DELAYTIME
DELAY1: DJNZ R3,DELAY1
DELAY2: DJNZ R4,DELAY1
        DJNZ R5,DELAY2
        RET

.....SETBIT.....
SETBIT: CLR C
        MOV A,#5BH
        SUBB A,R1
        JC NOTE
        MOV A,#3CH
        SUBB A,R2
        JC NOTE
        JMP SETxy
NOTE: JMP NOTE2
SETxy: CLR C
        MOV A,R1
        MOV B,#08H
        DIV AB
        MOV R3,A
        MOV B,#08H
        MUL AB
        MOV R4,A
        MOV A,R1
        SUBB A,R4
        MOV R4,A
        CLR C
        MOV A,R2
        MOV R3,#10H
        MUL AB
        ADD A,#0AH
        MOV R5,A
        MOV R6,B
        MOV A,R5
        SUBB A,R3
        MOV D_ADDRESSL,A
        MOV A,R6
        CLR C
        ADDC A,#90H
        MOV D_ADDRESSH,A
        MOV A,ALL1
        CJNE A,#0FFH,CLRBIT
        CLR C
        INC R4
        MOV R7,#0FFH
        MOV A,R7
        RLC A
        DJNZ R4,PP
        MOV R7,A
        MOV DPL,D_ADDRESSL
        MOV DPH,D_ADDRESSH
        MOVX A,@DPTR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ANL    A,R7
MOVX   @DPTR,A
SJMP   CONSET
CLRBIT: SETB   C
        ;;CLEAR BIT OF ADDRESS 9000;;
INC     R4
MOV     R7,#00H
MOV     A,R7
CC:     RLC     A
        DJNZ   R4,CC
MOV     R7,A
MOV     DPL,D_ADDRESSL
MOV     DPH,D_ADDRESSH
MOVX   A,@DPTR
ORL    A,R7
MOVX   @DPTR,A
CONSET: CLR    C
MOV     A,R1
MOV     B,#08H
DIV    AB
MOV     R3,A
MOV     B,#08H
MUL   AB
MOV     R4,A
MOV     A,R1
SUBB   A,R4
MOV     R4,A
MOV     A,R2
MOV     B,#10H
MUL   AB
ADD    A,#0AH
MOV     R5,A
MOV     R6,B
MOV     A,R5
SUBB   A,R3
MOV     D_ADDRESSL,A
MOV     A,R6
CLR    C
ADDC   A,#00H
MOV     R7,A
AA:     MOV     DPTR,#ADDRESS_LOW
        ;;OUTPUT TO 8255;;
MOV     A,D_ADDRESSL
MOVX   @DPTR,A
MOV     DPTR,#ADDRESS_HIGH
MOV     A,R7
MOVX   @DPTR,A
MOV     DPL,D_ADDRESSL
MOV     DPH,D_ADDRESSH
MOVX   A,@DPTR
MOV     DPTR,#DATAPORT
        ;;DATA FROM 9000+;;
MOVX   @DPTR,A
MOV     A,CHCK
CJNE   A,#01H,WR2
CALL   WRITE1
RET
WR2:   CALL   WRITE2
NOTE2: RET
        ;;;;;;;;;;LOAD;;;;;;;;;;;;;
LOAD:  INC     LOOP
        MOV     COUNTLOOP,LOOP
STARTLOAD:
MOV     PCON,#00H    ;sm0d=0
MOV     SCON,#050H
        ;serial (mode 1) ren = 1
MOV     TMOD,#20H
        ;timer1 mode2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MOV    TH1,#0FDH                MOV    R5,DPL
    ;9600 baud                    CJNE   R5,#0H,INC_DPH
SETB   TR1                       INC    DPH
    ;start timer 1                INC_DPH:
SETB   EA                         JMP    READ_T
RE_RX: CLR    ES                    CHECK_SUM:
CLR    TI                          CALL   DECODE_CH
CLR    RI                          JMP    RECEIVE
RECEIVE:                           END_LOAD:
JNB    RI,RECEIVE                 END_LOAD1:
MOV    A,SBUF                      RET
CLR    RI                          DECODE_CH:
CJNE   A,#3Ah,RECEIVE             JNB    RI,$
    ; check colon code           MOV    A,SBUF
CALL   DECODE_CH                  CLR    RI
MOV    R6,A    ; get counter      CALL   DEC_CHAR
CALL   DECODE_CH                  MOV    R4,A
    ; get hi_address             JNB    RI,$
MOV    DPH,A                       MOV    A,SBUF
CALL   DECODE_CH                  CLR    RI
    ; get lo_address            CALL   DEC_CHAR
MOV    DPL,A                       MOV    R3,A
CALL   DECODE_CH                  MOV    A,R4
    ; check end                 SWAP  A
CJNE   A,#1H,READ_DATA            ORL   A,R3
JMP    END_LOAD                   RET
READ_DATA:                          DEC_CHAR:
    INC    R6                      CLR    C
READ_T: DJNZ   R6,READ              SUBB  A,#30H
    ;get data from computer to ram  MOV    R2,A
JMP    CHECK_SUM                  CLR    C
READ:  CALL   DECODE_CH            SUBB  A,#0AH
MOVX   @DPTR,A                    JC    EXIT_DEC
INC    DPL                         MOV    A,R2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SUBB    A,#7H
        RET
EXIT_DEC:
        MOV     A,R2
        RET
NDD:    NOP
        END
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำขอขอบคุณ อาจารย์ชินภัทร นันทจิวงกรชัย ที่ได้ให้คำแนะนำและชี้แนะแนวทางต่างๆ ในการทำงานจนสามารถแก้ไขปัญหาคงโครงการสำเร็จลุล่วงไปได้ด้วยดี และขอขอบคุณผู้มีพระคุณอีกหลายท่านที่ช่วยเหลือและเอื้อเฟื้อต่อผู้จัดทำ จนทำให้โครงการสำเร็จไปได้ด้วยดี

.....
(นายวีรชาติ สุนทรวงศ์)

.....
(นายวีระชาติ ภูวนาด)

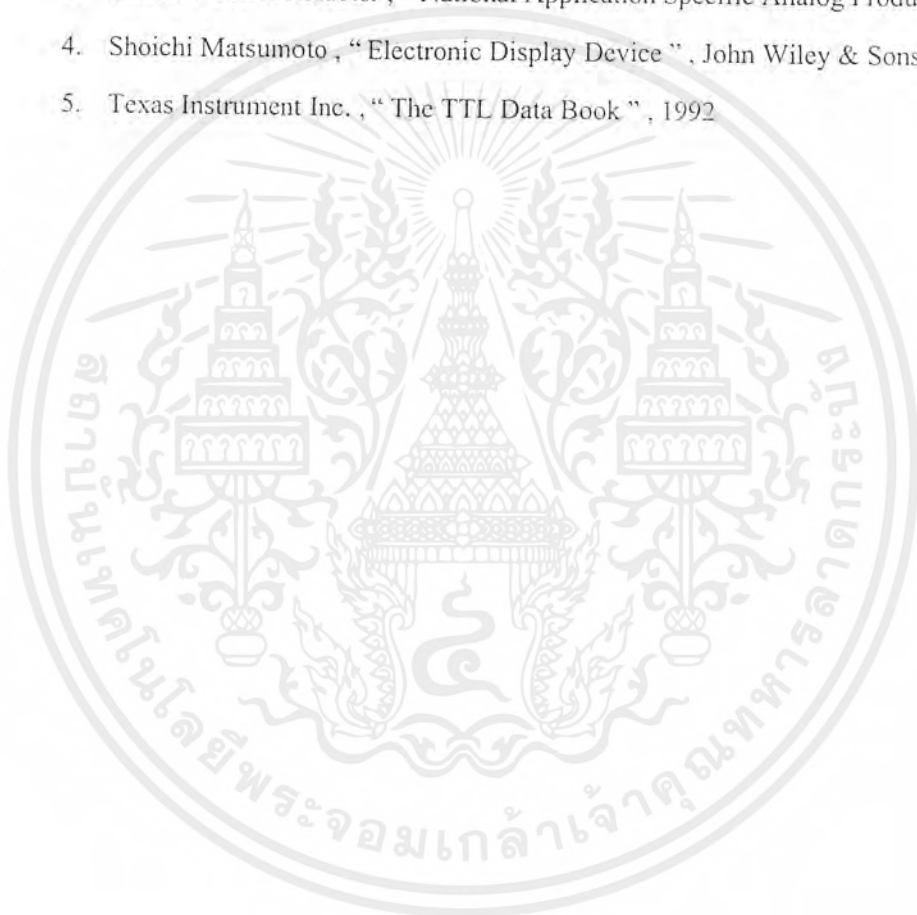
.....
(สรญา นัยวัฒน์)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. นิพนธ์ ศิริรัตน์, ไมโครคอนโทรลเลอร์ แชนเนล : เชื่อมต่อกับบอร์ดกับ Jazz-31 .
“เซมิคอนดักเตอร์ อิเล็กทรอนิกส์” , ฉบับที่ 150 , ปี 2538 , หน้า 54-57
2. Motorola , Inc. , “ High – Speed CMOS Data ” . 1996
3. National Semiconductor , “ National Application Specific Analog Products ” . 1995
4. Shoichi Matsumoto , “ Electronic Display Device ” , John Wiley & Sons 273p , 1993
5. Texas Instrument Inc. , “ The TTL Data Book ” , 1992



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้