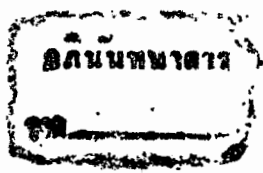




นาย คณิตเดช แซ่ลี เลขประจำตัว 281044
 นาย ณรงค์ศักดิ์ ลังขลังวัลย์ เลขประจำตัว 281072

อาจารย์ที่ปรึกษา

ผศ. อภินันท์ มัญญานท์
 อ. สุธัชชัย นพนาศิพงษ์



ปริญญาโทปีการศึกษา 2531

ภาควิชา โทคมมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่อง Simple Image Scanner

ผู้จัดทำ

1. นาย คณิงเดช แซ่ลี เลขประจำตัว 281044

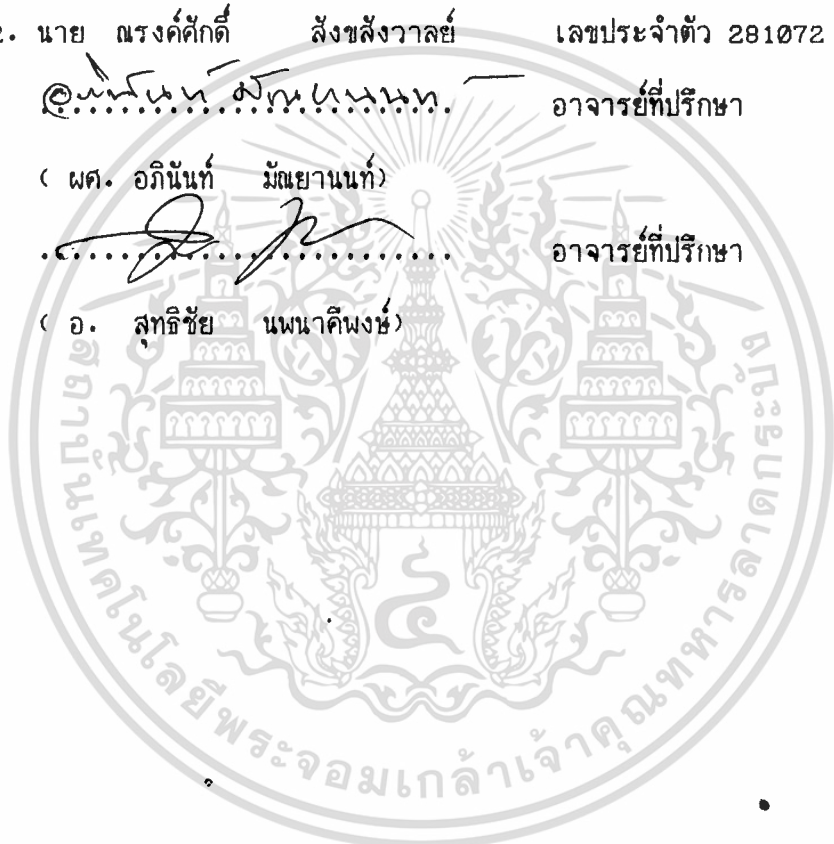
2. นาย ณรงค์ศักดิ์ สังขสังวาลย์ เลขประจำตัว 281072

อ.ณัฐพันธ์ ส.ท.บ.ร.น.ท. อาจารย์ที่ปรึกษา

(ผศ. อภินันท์ มัฒยานนท์)

..... อาจารย์ที่ปรึกษา

(อ. สุทธิชัย นพนาศิพงษ์)



สารบัญ

	บทคัดย่อ.....	1
	Abstract.....	2
บทที่ 1	บทนำ.....	3
บทที่ 2	Simple Image Scanner.....	4
	ส่วนรับสัญญาณภาพ.....	5
	ส่วนของวงจร a/d converter.....	6
	ส่วนของเครื่อง IBM PC และการ Interface.....	9
บทที่ 3	การสร้างส่วนต่าง ๆ ของเครื่องและการเขียนโปรแกรม.....	17
บทที่ 4	ผลการทดลอง.....	24
บทที่ 5	บทวิจารณ์และสรุปผล.....	32
	ภาคผนวก	33

เครื่อง SIMPLE IMAGE SCANNER

นาย คณิตเดช แซ่ลี

นาย แรงค์ศักดิ์ สังขลังวาลย์

ผศ. อภินันท์ มัชฌานนท์ อาจารย์ที่ปรึกษา

อ. สุธรัชชัย นพนาดีพงษ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2531

บทคัดย่อ

ในปัจจุบัน การใช้งานคอมพิวเตอร์ทางด้านกราฟิกต่าง ๆ ได้มีการพัฒนาไปมาก และเครื่องมืออีกชิ้นหนึ่งที่ใช้ในการสร้างภาพบนคอมพิวเตอร์ได้ก็คือ " IMAGE SCANNER " ซึ่งเราสามารถนำ IMAGE SCANNER นี้ไปประยุกต์ใช้งานต่าง ๆ ได้มากมาย เช่น การเก็บข้อมูลที่เป็น รูปถ่าย หรือ กราฟต่าง ๆ โดยจะใช้เครื่อง IMAGE SCANNER กับเครื่องคอมพิวเตอร์ IBM PC/XT

โดยปกติแล้วเครื่องคอมพิวเตอร์ในปัจจุบัน มักจะมี PORT สำหรับต่อ JOY STICK สำหรับเล่นเกมอยู่แล้ว ดังนั้นจึงได้ทำการใช้ PORT ของ JOY STICK นี้ ทำการเชื่อมต่อ (INTERFACE) ระหว่างเครื่อง IMAGE SCANNER และเครื่อง COMPUTER ทำให้เราไม่จำเป็นต้องเปลี่ยนแปลงตัวเครื่อง COMPUTER เลย ซึ่งจะสะดวกมากในการใช้งาน

SIMPLE IMAGE SCANNER

MR. Kanungdej Shairle

MR. Narongsak Sangkasangwan

Assistant Professor Apinan muntayanun Advisor

MR. Sutthichai nupnakepong Advisor

1988

Abstract

To develop Computer Graphic, Image Scanner is used for transfer the image, graph, other alphabets to show on computer screen.

This thesis is contain an overview of image scanner for IBM PC/XT and assembly language programming which control an interface to image scanner and computer PC by connecting joy stick port. This device is developed for a low cost for general uses

บทที่ 1

บทนำ

เครื่อง " IMAGE SCANNER " เป็นเครื่องมือที่ใช้ในการนำภาพ หรือ กราฟ
 ต่างๆ ที่เราต้องการให้ไปออกบนจอภาพของเครื่องคอมพิวเตอร์ โดยในที่นี้เครื่องคอมพิว
 เตอร์ที่ใช้เป็นเครื่องคอมพิวเตอร์ IBM/PC และจอที่ใช้เป็นจอภาพแบบ โมโนโครม
 (MONOCROME DISPLAY) ซึ่งมีความละเอียดในการแสดงภาพสูง และส่วน
 MAGCHANISM ที่ใช้ในการกวาดภาพนั้น จะใช้ PRINTER เป็นตัวกวาด โดยมี PHOTO
 SENSER เป็นตัวรับสัญญาณ

ในที่นี้ได้ทำการศึกษา และทำการทดลองเกี่ยวกับ การสร้าง PHOTO SENSER
 และวงจร A/D อย่างง่าย รวมถึงการใช้ SOFTWARE ในการควบคุมเครื่องคอมพิวเตอร์
 เพื่อแสดงผล

บทที่ 2

"simple image scanner"

เครื่อง "simple image scanner" นี้มีหลักการทำงานซึ่งไม่ยุ่งยากมากนัก เป็นการนำเอารูปภาพ กราฟ หรืออักษรต่างๆ มาแสดงออกที่ หน้าจอคอมพิวเตอร์ สำหรับคุณสมบัติของเครื่อง ที่ทำในโครงการนี้ มีข้อดีคือ

1. มีราคาถูกมาก
2. สามารถต่อเข้ากับเครื่องคอมพิวเตอร์ได้โดยใช้สายต่อเพียง 3 เส้น
3. ไม่ต้องปรับปรุงแก้ไข เครื่องพิมพ์เลย
4. สามารถปรับระดับสัญญาณได้

มีส่วนประกอบสำคัญอยู่เพียง 3 ส่วนเท่านั้น ซึ่งแยกออกได้ดังต่อไปนี้

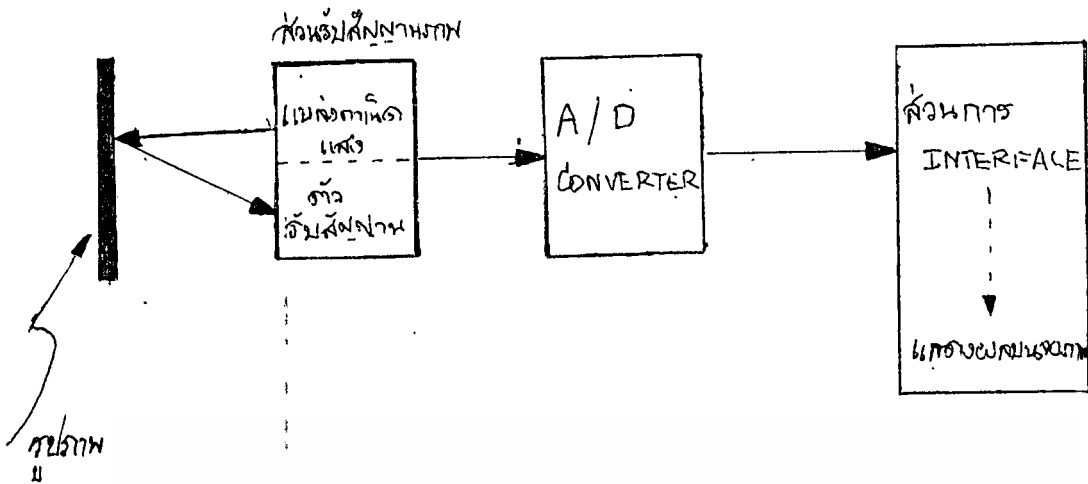
1. ส่วนรับสัญญาณภาพ (photo sensor) : เป็นส่วนที่ทำการรับสัญญาณภาพ ในรูปของสัญญาณแสงที่สะท้อนจากการส่องแสงไปกระทบภาพแล้วจะนำสัญญาณแสงที่ได้ไปเปลี่ยนให้เป็นสัญญาณทางไฟฟ้า (analog signal) เพื่อส่งต่อไปให้ส่วนต่อไป

2. วงจร "ANALOG TO DIGITAL CONVERTER" : เป็นส่วนที่ใช้ในการเปลี่ยนสัญญาณ "analog" ที่ได้จากส่วนของการรับสัญญาณภาพโดย photo sensor ให้เป็นสัญญาณ "digital" เพื่อให้สามารถทำการประมวลผลทางคอมพิวเตอร์ได้

3. ส่วนของเครื่อง IBM PC และการ interface : เป็นส่วนที่ใช้ในการรับ สัญญาณ "digital" จากวงจร A/D converter แล้วนำออกแสดงผลทางจอภาพของเครื่องคอมพิวเตอร์ โดยในที่นี้ใช้ joy stick port ของ game controll adapter เป็น port ในการเชื่อมต่อ

จากส่วนประกอบต่างๆ ที่ได้กล่าวมาสามารถนำมาเขียน block diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ได้ตั้งรูปที่ 1.
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1. แสดง block diagram ของเครื่อง simple image scanner

"ส่วนรับสัญญาณภาพ"

สำหรับส่วนรับสัญญาณภาพมีส่วนประกอบสำคัญอยู่ 2 ส่วนด้วยกัน คือ

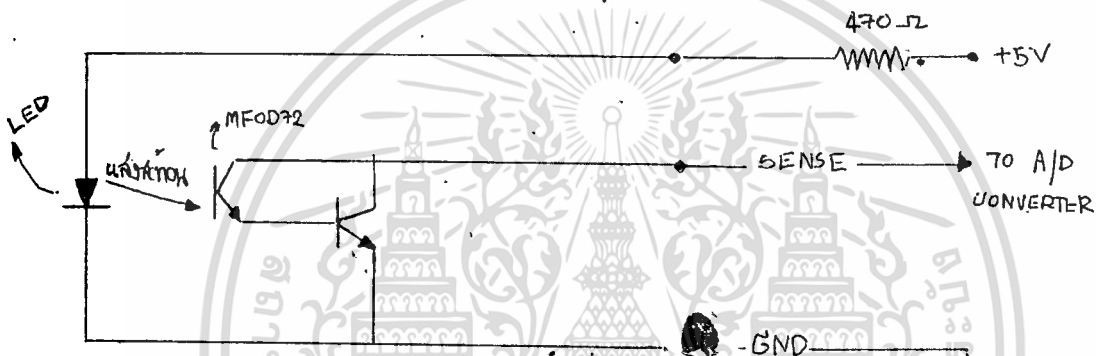
1. แหล่งกำเนิดแสง (light source) เป็นส่วนที่ปล่อยแสงออกไปกระทบกับภาพ โดยในที่นี้ใช้ LED (light emitting diode) สีแดงซึ่งจะปล่อยแสงที่มีความยาวคลื่นประมาณ 640 นาโนเมตร เป็น แหล่งกำเนิดแสง
2. ตัวรับสัญญาณแสง (photo detector) เป็นตัวรับสัญญาณแสงแล้ว เปลี่ยนเป็นสัญญาณไฟฟ้า ในที่นี้ใช้ photo transistor ที่มีช่วงของการตอบสนองสัญญาณสูงในช่วงความยาวคลื่นของแสงสีแดง ทำหน้าที่เป็นตัวรับสัญญาณ

สำหรับหลักการทำงานของ ส่วนรับสัญญาณภาพ (photo sensor) นี้มีอยู่ว่าเมื่อแสงจาก แหล่งกำเนิดแสง เดินทางไปกระทบกับภาพแล้วก็จะเกิดการสะท้อนกลับและสัญญาณแสงที่สะท้อนกลับมานี้ก็จะถูกตรวจจับด้วย ตัวรับสัญญาณแสง แต่เนื่องจากการดูดกลืนงานที่ไม่เท่ากันของ สีต่างขบภาพ ดังนั้นแสงที่สะท้อนกลับมาจากภาพก็จะมีขนาดความเข้มของแสงที่ต่างกันหากกระทบกับส่วนของภาพที่มีสีต่างกัน และเมื่อถูกตรวจจับได้โดย ตัวรับสัญญาณแสง ซึ่งในที่นี้คือ photo transistor ก็จะทำให้ได้ระดับของสัญญาณไฟฟ้าที่ต่างกันออกมาด้วย

แต่เนื่องจากระยะทางระหว่าง แหล่งกำเนิดแสง และ ตัวรับสัญญาณแสง

(noise) ที่มากขึ้นตามมา ดังนั้นเพื่อลดปัญหานี้จึงได้มีการใช้ สาย"เส้นใยแสง"มาทำการเชื่อมต่อระหว่าง แหล่งกำเนิดแสง และ ตัวรับสัญญาณแสง จากคุณสมบัติของสายเส้นใยแสงที่มีการสูญเสียของแสงภายในสายต่ำ จึงสามารถนำมาใช้เป็นทางเดินของแสงได้ดี

นอกจากนี้แล้ว สัญญาณที่ได้จาก photo transistor หากนำไปต่อกับโหลด ที่กิกะแอมป์ จะไม่สามารถขับกระแสได้พอ ดังนั้นจึงได้นำเอา transistor แบบ NPN อีกตัวหนึ่งมาต่อ แบบ darlington เอาไว้เพื่อให้สามารถขับกระแสได้มากขึ้น สำหรับวงจรของส่วนรับสัญญาณนี้แสดงดังรูปที่ 2



รูปที่ 2. แสดงส่วนรับสัญญาณภาพ

"ส่วนของวงจร A/D CONVERTER"

ในส่วนของวงจร A/D converter นี้ได้ใช้ op amp (operational amplifier) เบอร์ LM 339A ซึ่งเป็น วงจรรวมแบบ dip (dual in line package) มาทำเป็นวงจร comparator ซึ่ง out put ที่ได้ออกมาจาก วงจรนี้จะมีค่าเพียง 2 ค่า เท่านั้นคือ low หรือ high

ในที่นี้ได้ทำการออกแบบวงจร a/d converter ให้ใช้ได้ 2 mode การทำงานด้วยกัน คือ mode ที่ 1. จะเป็นโหมดที่ใช้ในการแสดงผลแบบหยาบ ส่วน mode ที่ 2. นั้น ใช้ในการแสดงผลแบบทีละเอียดยกกว่า

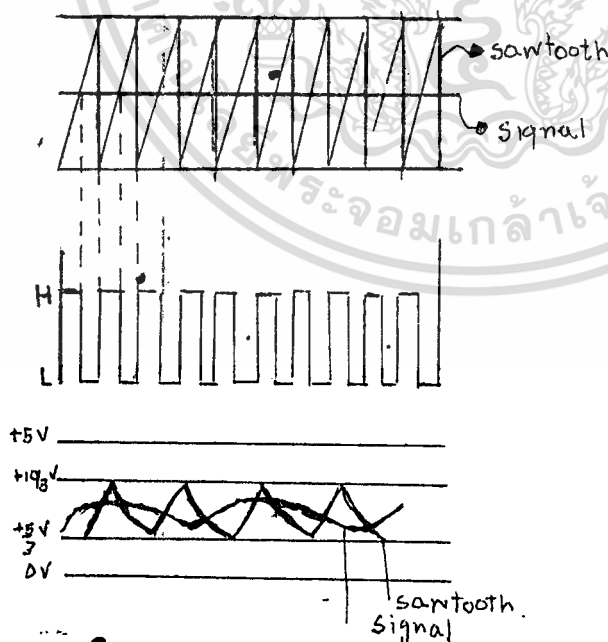
วงจร comparator นี้มี คุณสมบัติคือ หากระดับของ สัญญาณ input ที่ขา

จากวงจร astable multivibrator มาป้อนให้ที่ขา inverting ของ opamp แทนระดับสัญญาณ dc ธรรมดา ผลที่ได้ก็คือ out put ที่ออกมาจะอยู่ในรูปของ pulse ที่มีความกว้าง ต่างๆกัน โดยความกว้างของ pulse ขึ้นอยู่กับระดับของสัญญาณ input ที่ขา noninverting หรือ นั่นก็คือ amplitude ของสัญญาณถูกเปลี่ยนให้อยู่ในรูปความกว้างของ pulse แทน โดยการทำเช่นนี้จะทำให้สามารถ รับรู้ระดับสัญญาณได้มากกว่า 2 ระดับซึ่งนั่นก็คือ สีที่อยู่ระหว่าง ขาว และดำ สามารถจะรู้ได้ ดังนั้นในโหมดนี้จึงเหมาะที่จะใช้กับพวก รูปถ่าย

ส่วนของ วงจร astable multivibrator นั้นเราใช้ IC เบอร์ 555 ซึ่งเป็น IC ที่นิยมใช้กันมากอยู่แล้ว มาทำดังแสดงไว้ในรูปที่.3 สำหรับสูตรการคำนวณหา ค่าความถี่ ที่จะนำมาใช้งาน ของ IC 555 ในโหมด astable คือ

$$f = 1.44 / (R_1 + R_2) C$$

ในรูปที่ 4 ได้แสดงการทำ PWM ระหว่าง สัญญาณภาพ และ สัญญาณ saw tooth เอาไว้



รูปที่ 4. PWM

จากวงจร ในรูปที่ 3 จะเห็นว่า มีสายสัญญาณออก เพียง 3 เส้นเท่านั้น คือ +5 volt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ground และ สาย signal
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



"ส่วนของเครื่อง IBM/PC และการ

ในส่วนนี้ขอแบ่งออก เป็น 2 ส่วนด้วยกัน คือ 1 ส่วนการ interface และ 2 ส่วนของการ แสดงผลทางจอภาพ.

1. ส่วนของการ interface ในที่นี้จะใช้การ interface ผ่านทาง game controll adapter ของ เครื่อง IBM/PC ซึ่งมีลักษณะสำคัญข้อหนึ่งของ card นี้คือ สามารถรับข้อมูลได้อย่างเดียว โดยมี INPUT ที่รับสัญญาณ digital อยู่ 4 บิต และ INPUT อีก 4 บิตที่รับสัญญาณผ่านทางความต้านทาน เพื่อใช้เป็นค่าหน่วยเวลาให้กับวงจร ONE-SHORT ซึ่งในการการใช้งาน 4 บิตหลังนี้จะไม่กล่าวถึง

เมื่อทำการรับสัญญาณจาก port นี้เข้ามา 1 ไบท์ (byte) ข้อมูลของ บิต ต่างๆแสดงได้ดังต่อไปนี้

bit 7 bit 6 bit 5 bit 4 bit 3 bit 2 bit 1 bit 0

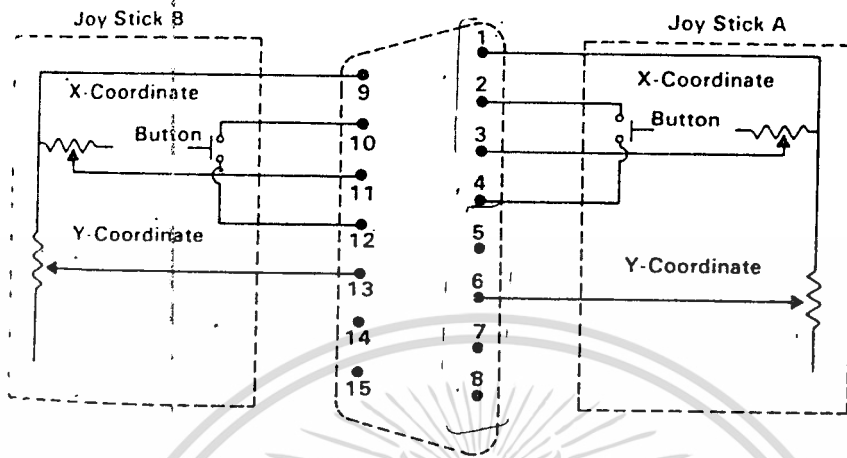
digital input

resistive input

จะเห็นว่า 4 บิต บนนั้นใช้ในการรับสัญญาณ digital ซึ่งในที่นี้เราต้องการ ใช้เพียงบิตเดียว เพื่อรับสัญญาณ และในที่นี้ได้เลือกเอา บิตที่ 4 มาใช้งาน

สำหรับ address ที่เครื่อง IBM PC ได้กำหนดเอาไว้สำหรับ game controll adapter card นี้ก็คือ address ที่ 201 ฐานสิบหก ซึ่งเราสามารถจะติดต่อกับ port นี้ได้โดยการใช้ คำสั่ง IN หรือ out กับ port address ที่ 201H โดย บิตที่สนใจคือบิตที่ 4 สำหรับการเชื่อมต่อกับ game controll adapter นี้ จะต่อผ่าน connector แบบ d type ดังแสดงดังรูปที่ 5 จากรูปจะเห็นว่า มี pin ที่ต่ออยู่ทั้งหมด 15 pin ซึ่งมีรายละเอียดของ pin ต่างๆดังรูปที่ 6 และจากรูปที่ 6 นี้ จะพอเห็นได้ว่า เราสามารถที่จะเลือกใช้สาย ในการรับสัญญาณซึ่งในที่นี้ต้องการเพียง 1 บิตได้ ถึง 4 ชุด

15-Pin Male D-Shell Connector



รูปที่ 5 แสดง D type conector ของ game controll adapter

At Standard TTL Levels

External Devices	Adapter Pin No.	Game Control Adapter
+5 Vdc	1	
Button 4	2	
Position 0	3	
Ground	4	
Ground	5	
Position 1	6	
Button 5	7	
+5 Vdc	8	
+5 Vdc	9	
Button 6	10	
Position 2	11	
Ground	12	
Position 3	13	
Button 7	14	
+5 Vdc	15	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของการแสดงผลออกทางจอภาพ ในที่นี้มีการแสดงผลบน IBM/PC ได้ 2 วิธี วิธีแรกคือ ทำการส่งข้อมูลต่างๆที่ต้องการแสดง ให้กับจอภาพในลักษณะเดียวกับอุปกรณ์ I/O ชนิดอื่นๆ โดยใช้การส่งคำสั่งต่างๆ ไปยังส่วนควบคุมการแสดงผล และอีกวิธีหนึ่ง ซึ่งเป็นที่นิยมใช้กันมาก เพราะสามารถสร้างรูปแบบการแสดงผลพิเศษได้มากกว่าวิธีแรก เรียกว่าวิธี MEMORY MAPPED

วิธีการของ MEMORY MAPPED นี้ จะมีลักษณะที่ CPU และส่วนแสดงผลใช้หน่วยความจำร่วมกัน ซึ่งในวิธีการนี้ส่วนแสดงผลจะต้องทำการอ่านข้อมูลจากหน่วยความจำ และแสดงผลออกบนจอภาพตามข้อมูลที่ปรากฏอยู่ในหน่วยความจำตลอดเวลา สำหรับวิธีการที่จะส่งข้อมูลออกไปยังตำแหน่งในจอภาพ เราก็สามารถที่จะทำได้โดยการส่งข้อมูลนั้นไปยังตำแหน่งของหน่วยความจำที่ตรงกับตำแหน่งบนจอภาพที่ต้องการ ซึ่งตำแหน่งต่าง ๆ บนจอภาพสามารถที่จะแทนได้โดยตำแหน่งของหน่วยความจำ ดังนั้นในกรณีที่ต้องการจะเขียนหรืออ่านข้อมูลจากจอภาพ เราก็จะสามารถทำได้โดยการเขียนหรืออ่าน จากหน่วยความจำในตำแหน่งที่ต้องการ

โดยทั่วไปแล้วในการแสดงผลบนจอภาพของระบบ IBM/PC จะขึ้นขึ้นอยู่กับลักษณะของการแสดงผล สำหรับวิธีของ Memory Mapped นั้นใช้สำหรับการแสดงผลข้อมูลทั่ว ๆ ไป ส่วนการใช้คำสั่งต่าง ๆ นั้นมักจะใช้ในการควบคุมรูปแบบต่าง ๆ ในการแสดงผลเช่น ความคมชัดของเคอร์เซอร์ หรือการเลือกโหมดในการแสดงผล เป็นต้น

สำหรับ Memory Mapped นี้มีข้อเสียอยู่ 2 ข้อคือ

1. เนื่องจาก Memory Mapped นี้ต้องใช้เนื้อที่ในหน่วยความจำด้วย ดังนั้น IBM/PC จึงแก้ไขโดยการให้ CPU ซึ่งสามารถที่จะ Access หน่วยความจำได้ถึง 1 Mbyte และกำหนดให้ส่วนของจอภาพมีหน่วยความจำภายใน ซึ่งโดยวิธีนี้จะทำให้ส่วนของจอภาพนั้นไม่จำเป็นต้องเข้าไปยุ่งเกี่ยวกับหน่วยความจำปกติที่ใช้สำหรับเก็บโปรแกรมและข้อมูลทั่วไปเลย

2. ส่วนแสดงผลจะต้องทำการอ่านข้อมูลในหน่วยความจำ เพื่อนำไปแสดง

กับ CPU แล้วก็จะเป็นการรบกวนการทำงานของ CPU เนื่องจาก CPU ต้องแบ่งเวลาในการ Access หน่วยแสดงผล ดังนั้นเพื่อแก้ปัญหานี้ IBM/PC จึงใช้วิธีการ Access หน่วยความจำแบบ Dual-port โดยวิธีการนี้จะช่วยให้การ Access หน่วยความจำของ CPU กับส่วนแสดงผลเกิดขึ้นได้อย่างรวดเร็วยิ่งขึ้น

สำหรับการ Access หน่วยความจำแบบ Dual-Port ระหว่าง CPU กับส่วนแสดงผลนี้ จะช่วยลดการรบกวนระหว่างจอภาพกับโปรแกรมที่ทำการส่งข้อมูลที่ต้องการจะแสดงให้กับจอภาพ แต่อย่างไรก็ตามในกรณีที่เราใช้งานโปรแกรมที่ทำการส่งข้อมูลออกไปยังส่วนแสดงผลได้อย่างรวดเร็ว เช่น โปรแกรม Word Processor ที่มีความเร็วสูง ก็มักเกิดเงาซ้อนบนจอภาพ ซึ่งเกิดจากการที่ส่วนควบคุมการแสดงผลและโปรแกรมที่เราใช้งานนั้นทำการอ่านหรือเขียนข้อมูลในเวลาเดียวกัน

การที่ IBM/PC. ใช้วิธี Memory Mapped การที่สร้างหน่วยความจำภายในให้กับส่วนแสดงผลและการ Access หน่วยความจำแบบ Dual-Port นั้นนอกจากจะเป็นการช่วยให้การทำงานของ CPU ร่วมกับการแสดงผลเป็นไปอย่างมีประสิทธิภาพแล้วยังเป็นการช่วยให้ IBM/PC สามารถที่จะสร้างรูปแบบการแสดงผลที่มีรายละเอียด สูงมาก

ในการแสดงผลแบบ Text Mode ในหน้าจอหนึ่ง ๆ จะแบ่งออกเป็น 80 * 25 บรรทัด ที่แถวที่ 1 ตัวที่ 1 จะตรงกับหน่วยความจำ ที่ B000:0000 และแถวที่ 1 ตัวที่ 2 จะตรงกับหน่วยความจำที่ B000: 0002 ดังนั้นหากเราใส่ค่า 41 และ 42 ลงในหน่วยความจำที่ B000: 0000 และ B000:0002 ตามลำดับแล้วตัวอักษร A และ B ก็จะไปอยู่ที่มุมบนทางซ้ายของจอภาพด้วย (41 และ 42 เป็นรหัส ASCII ของอักษร A และ B) จะเห็นได้ว่าข้ามหน่วยความจำที่ B000:0001 และ B000:0003 ไปทั้งนี้เนื่องจากในโหมดแสดงผลแบบตัวอักษร จะใช้หน่วยความจำ 2 Byte ต่อ 1 ตัวอักษร โดย Byte แรกเป็น ASCII ของตัวอักษรที่จะแสดงและ Byte ที่ 2 เป็น Attribute จะแสดงรายละเอียดของ Attribute แสดงดังรูปที่ 7

Attribute Function	Attribute Byte							
	7	6	5	4	3	2	1	0
	B	R	G	B	I	R	G	B
	FG	Background			Foreground			
Normal	B	0	0	0	I	1	1	1
Reverse Video	B	1	1	1	I	0	0	0
Nondisplay (Black)	B	0	0	0	I	0	0	0
Nondisplay (White)	B	1	1	1	I	1	1	1

I = Highlighted Foreground (Character)

B = Blinking Foreground (Character)

รูปที่ 7 แสดงรายละเอียดของ Attribute Byte

ในส่วนการแสดงผลกราฟฟิก จอ Monochrome สามารถแสดงได้ในแบบ hiresolution เท่านั้น คือใน 1 Byte ของหน่วยความจำบิตต่าง ๆ จะหมายถึง พิกซ์เซล แต่ละตัวบนจอภาพ ถ้าบิตใดเป็น 1 ก็จะทำให้พิกซ์เซลติด ถ้าเป็น 0 ก็จะทำให้พิกซ์เซลดับ โดยในโหมดการแสดงผลแบบกราฟฟิกนี้ จะไม่มี Attribute Byte แต่อย่างใด ในจอภาพแบบ Monochrome นี้มีความละเอียดจุดถึง $720 * 348$ ซึ่งมีค่าเท่ากับ $250,560$ จุดต่อ 1 จอภาพ ซึ่งในโครงงานนี้เราใช้ การ์ดเออร์คิวลิส มาเป็นส่วนที่ใช้ในการแสดงผลบนจอภาพ

การ์ดเออร์คิวลิส

การ์ดเออร์คิวลิสนี้หรือที่เรียกว่า การ์ดโมโนโครมกราฟฟิกนั้น ในเครื่องคอมพิวเตอร์พีซีนั้นการ์ดที่ใช้กับจอโมโนโครมส่วนมากก็จะเป็นการ์ดแบบนี้ เป็นการ์ดที่ออกแบบมาให้สร้างภาพบนจอภาพได้ทั้งในแบบคาแรกเตอร์โหมด ซึ่งในยุค 16 บิตนี้เรียกว่าเท็กซ์โหมด (Text Mode) และในแบบกราฟฟิกโหมด (Graphics Mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยจัดหน่วยความจำ RAM ไว้ 64 k ไบท์ ตั้งแต่แอดเดรสที่ \$B0000 ถึง \$BFFFF ดังนั้นหน่วยความจำ 32 k ไบท์หลังคือตั้งแต่แอดเดรสที่ \$B0000 เป็นต้นไปคือ หน่วยความจำที่ได้กำหนดไว้เป็นจอภาพสำหรับภาพกราฟฟิก จะซ้ำกับหน่วยความจำในการ์ด ลี (Color Graphics Adapter) นั่นคือเราจะใช้การ์ดเออร์คิวลิสร่วมกับการ์ดลีไม่ได้ หน่วยความจำทั้ง 64 k ไบท์นี้ ได้ถูกแบ่งออกเป็น 2 ส่วนคือ 32 k ไบท์ แรก เรียกว่า เพจ 0 (page 0 :PGE0) อีก 32 k ไบท์หลังเรียกว่าเพจ 1 (page 1:PGE1)

เมื่อเราเริ่มเปิดไฟเข้าเครื่อง จากการออกแบบของการ์ดเออร์คิวลิสจะตัด เพจ 1 ออกจากระบบ (คือไม่รบกวนกับการ์ดลี) หากต้องการให้ต่อจะต้องใช้โปรแกรม HGC.COM การ์ดลีของคอมพิวเตอร์ จะเป็นการใช้หน่วยความจำตั้งแต่แอดเดรสที่ \$B0000 ขึ้นไปคือตรงกับเพจ 1 ของการ์ดเออร์คิวลิสเท่านั้น ส่วนสำหรับการ์ดเออร์คิวลิสนั้นเราสามารถให้ทั้งเพจ 0 และเพจ 1 อันใดอันหนึ่งก็ได้ เพื่อใช้กับกราฟฟิก แต่เฉพาะใน high graphics mode เท่านั้น

โปรแกรม HGC.COM นี้มีโปรแกรมย่อยให้เราเลือกใช้งานคือ

A> HGC FULL

หมายถึงเราเลือกใช้ FULL การทำงานของโปรแกรมมีดังนี้

FULL ต่อเพจ 1 เข้ากับระบบ

HALF ตัดเพจ 1 ออกจากระบบ

DIAG เพื่อให้รันโปรแกรม Diagchic ของไอบีเอ็มได้คือตัด RAM ออกคงเหลือ เพียง 4 k ไบท์แรก

SAVE หากไม่ใช่คีย์บอร์ดนานเกิน 5 นาที จะงดการส่งสัญญาณออกจอ คือทำให้จอ ภาพมืด หากต้องการใช้จอภาพใหม่ ให้กดคีย์ Shift ทั้งนี้เพื่อถนอมจอภาพ

PRINT เมื่อเลือกโปรแกรมนี้แล้วจะฝังตัวอยู่ในหน่วยความจำ เมื่อจะเรียกใช้กด คีย์ดังนี้

(shift)-(PrtScr) 0 พิมพ์กราฟฟิกในเพจ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Shift)-(PrtScr) 1 พิมพ์กราฟฟิกในเพจ 1

(Shift)-(PrtScr) (Space) พิมพ์อักขระในเท็กซ์โหมด

ซอฟต์แวร์สวิตช์ (Soft switch)

เพื่อที่จะให้เราเปลี่ยนแปลงการแสดงผลจอภาพให้เป็นในแบบเท็กซ์โหมดหรือกราฟฟิกโหมดได้ การ์ดเออร์คิวลิสได้ทำเป็นซอฟต์แวร์สวิตช์ไว้ คือสวิตช์ที่สั่งให้ on หรือ off ได้ด้วยคำสั่ง out โดยจัดไว้เป็นพอร์ต 2 พอร์ต คือ

Configuration Switch

คือพอร์ต *03BF ที่ทำหน้าที่ตัด/ต่อ ระบบให้เป็น กราฟฟิกโหมดในการใช้งาน หากให้ค่า *03 แก่พอร์ตนี้แล้วทำการต่อเพจ 1 และให้สวิตช์เป็นกราฟฟิกโหมดได้ หากให้ค่า *02 จะตัดเพจ 1 และห้ามโปรแกรมเป็นกราฟฟิกโหมด

Display Mode Control Port

คือพอร์ต *03B8 ซึ่งเป็นพอร์ตขนาด 8 บิต แต่ใช้งานเป็นสวิตช์เพียง 4 บิตคือ

บิต 1 ค่า 0 = เลือกเท็กซ์โหมด

ค่า 1 = เลือกกราฟฟิกโหมด

บิต 3 ค่า 0 = ไม่นำภาพออกจอภาพ

ค่า 1 =ให้นำภาพออกจอภาพ

บิต 5 ค่า 0 = ไม่ให้อักขระกระพริบ (ในเท็กซ์โหมด)

ค่า 1 = ให้กระพริบได้ตามค่า attribute

บิต 7 ค่า 0 = เลือกเพจ 0

ค่า 1 = เลือกเพจ 1

โปรแกรม CRTIC

การ์ดเออร์คิวลิสใช้ CRTIC เบอร์ 6845 เป็นวงจรควบคุมจอภาพ

(Cathode Ray Tube Controller) เมื่อเรากำหนดให้แสดงผลภาพในแบบใดแล้วจะต้อง

โปรแกรม CRTIC ให้สอดคล้องกันด้วย

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และห้ามเผยแพร่โดยไม่อนุญาติจากศูนย์บริการคอมพิวเตอร์
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังแสดงในรูปที่ 8 ซึ่งการให้ค่ารีจิสเตอร์นี้ เราต้องให้ค่าผ่านพอร์ต 2 พอร์ต พอร์ตแรกให้ชื่อ reg คือพอร์ต #03B4 ค่าที่ให้คือหมายเลขรีจิสเตอร์ที่เราจะให้ค่า เสร็จแล้วเราต้องให้ค่าที่ต้องการให้แกรีจิสเตอร์นี้ คือค่าในรูปที่ 8 ผ่านอีกพอร์ตหนึ่งให้ชื่อว่า พอร์ต data คือพอร์ต #03B5

register	function	text	graphics
R0	Horizontal Total	97	53
R1	Characters / Row	80	45
R2	HSYNC Position	82	46
R3	HSYNC Width(-1)	15	7
R4	Vertical Total(-1)	25	91
R5	VSYNC Adjust	6	2
R6	Character Rows/Frame	25	87
R7	VSYNC Position	25	87
R8	Interlace Mode	2	2
R9	Scan Line/Row(-1)	13	13
R10	Cursor Start ScanLine	11	0
R11	Cursor Stop ScanLine	12	0

รูปที่ 8 ตารางแสดงค่าที่จะโปรแกรมให้แกรีจิสเตอร์ของ CRTC เบอร์ 6845 เมื่อแสดงภาพในเท็กซ์โหมดและกราฟฟิกโหมด

การเปลี่ยนโหมด

ในคู่มือการ์ดเออร์คิวลิสได้ให้คำแนะนำในการเปลี่ยนโหมดจากเท็กซ์โหมด

เป็นกราฟฟิกโหมด โดยให้ดำเนินการตามขั้นตอนดังนี้คือ

- 1 ต่อเพจ 1 เทียบได้กับ HCG FULL
- 2 เปลี่ยนเป็นกราฟฟิกโหมดแต่ยังไม่ให้ส่งสัญญาณออกจอภาพ
- 3 โปรแกรม CRTC
- 4 ลบจอภาพ (เพื่อป้องกันไม่ให้ภาพเด่นเป็นการยึดอายุจอภาพ)
- 5 ให้ส่งสัญญาณออกจอภาพได้

ซึ่งการดำเนินการตามขั้นตอนนี้ ให้เขียนด้วยภาษาเครื่อง ไม่ควรเขียนด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ภาษาสูง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การสร้างส่วนต่าง ๆ ของเครื่องและการเขียนโปรแกรม

1 การสร้างส่วนตัวรับสัญญาณภาพ

เราใช้ LED แสงสีแดงเป็นแหล่งกำเนิดแสง และใช้ photo transistor เป็นตัวรับสัญญาณแสงที่สะท้อนจากภาพ เพื่อให้แสงสะท้อนที่ได้เข้าไปใน photo transistor มากที่สุดจึงได้นำเอาเลนส์แสงมาทำการเชื่อมต่อระหว่าง LED กับ photo transistor โดยในส่วนของ LED เราใช้กาวยึดติดเข้ากับด้านข้างของ LED ส่วนทางด้าน photo transistor นั้นเราใช้เบอร์ MF0D72 ซึ่งเป็นแบบที่มีรูเสียบจึงนำเอาเลนส์แสงมาเสียบเข้าที่รูแล้วทำการยึดติดไว้ แล้วนำส่วนนี้ไปติดที่หัวของเครื่องพิมพ์ต่อไปเพื่อทำเป็นตัวกวาดสัญญาณ

2 ส่วนของวงจร a/d converter

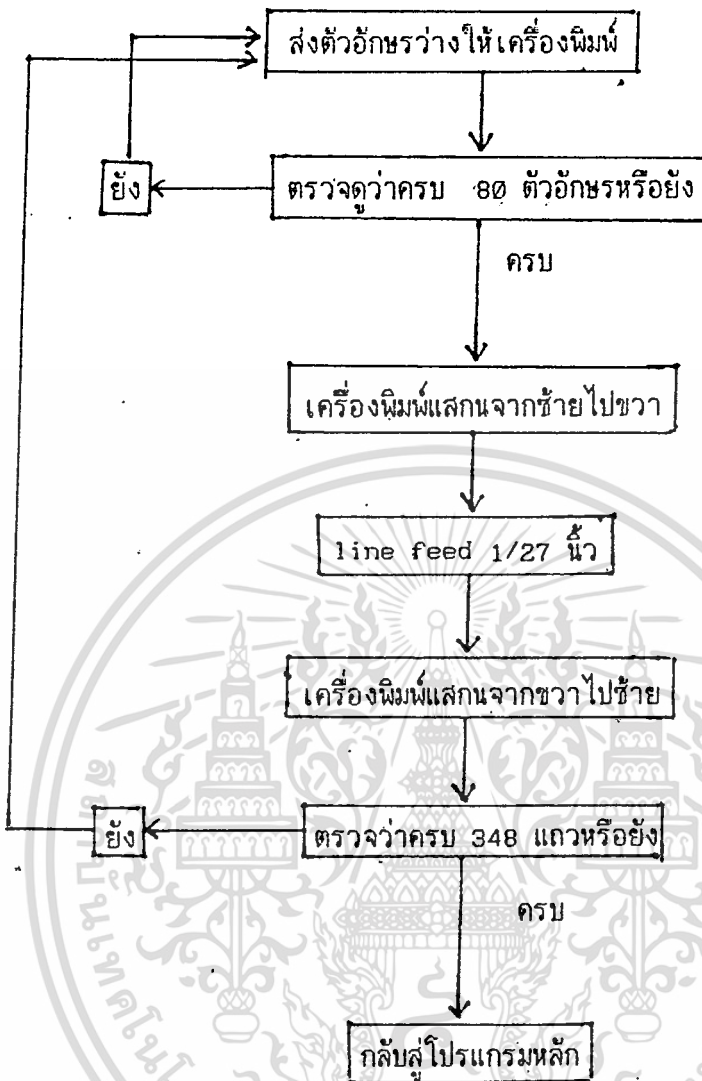
ในส่วนนี้ได้ทำการต่อวงจรดังรูปที่ 3 โดยใช้ IC LM339A เป็นวงจร comparator และสร้างสัญญาณไฟตรง 5/3 โวลต์ ใช้ IC เบอร์ 555 มาสร้างสัญญาณฟันเลื่อยเป็น clock เพื่อใช้ต่อเข้ากับในวงจร comparator ในโหมด 2 ในหลักการของ PWM

3 ส่วนของการเขียนโปรแกรม

ได้ทำการเขียนภาษา Assembly ความคมและแสดงผลบนจอภาพ ในส่วนโปรแกรมที่ใช้ในการควบคุมและการแสดงผลซึ่งเป็นการเขียนอธิบายโปรแกรมที่สำคัญเท่านั้นในส่วนโปรแกรมที่เขียนเป็นภาษา Assembly จะหาดูได้ในภาคผนวก

1 ส่วนของการกวาดสัญญาณภาพ

ในส่วนนี้เราได้นำเอา LED มาติดที่หัวเครื่องพิมพ์เป็นตัวกวาดสัญญาณ ดังที่ได้กล่าวมาแล้วในส่วนก่อน ๆ ซึ่งเราได้ทำการเขียนโปรแกรมในส่วนนี้โดยสั่งให้ เครื่องพิมพ์ทำการกวาดหัวของเครื่องพิมพ์จากซ้ายมาขวา พอเสร็จก็ให้กวาดกลับจากซ้ายมาขวาโดยมีการเลื่อนบรรทัดลงมาทีละ 1/27 นิ้ว โดยในส่วนนี้ได้ทำการเขียนไว้ในส่วนของโปรแกรมย่อยที่มีชื่อว่า SCAN ซึ่งแสดงได้ดังผังงานต่อไปนี้

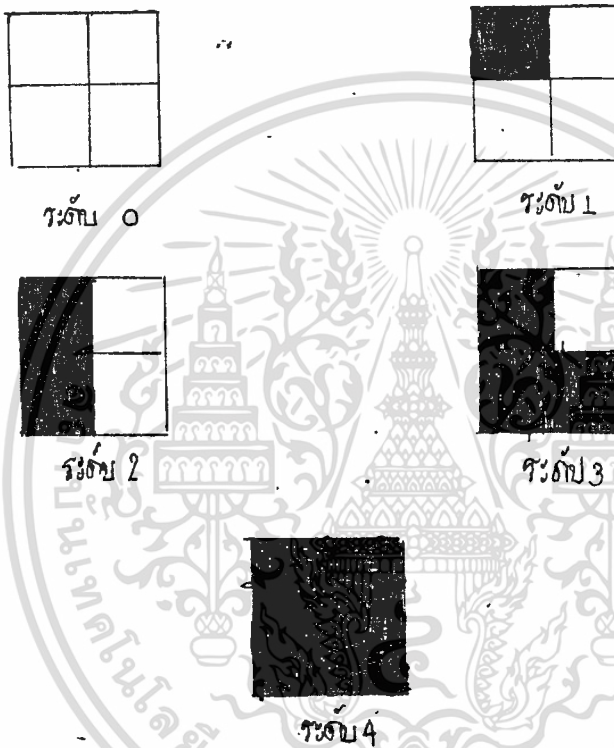


2 ส่วนของการรับสัญญาณภาพจากตัวรับสัญญาณ

เมื่อเราได้ทำการต่อตัวรับสัญญาณภาพเข้ากับส่วน a/d converter แล้วทำการ interface เข้ากับ joy stick port ของเครื่องคอมพิวเตอร์พีซี ซึ่งพอร์ทของ joy stick มีค่าเท่ากับ 201H เราทำการเขียนโปรแกรมให้รับค่าที่พอร์ทนี้เข้ามาโดยการรับค่าแบ่งออกเป็น 2 โหมดคือ โหมด 1 และ โหมด 2 โดยโหมด 1 จะแสดงสี ขาว ดำ 2 สีเท่านั้น กับในโหมด 2 แสดงระดับสีขาว ดำ ได้แตกต่างกัน 5 ระดับโหมด 1 นั้นเพียงแต่แสดงข้อมูล "0" หรือ "1" ที่เข้ามาเท่านั้น ในกรณีของเครื่อง simple image scanner นี้จะให้เอาไฟพุ่งออกมาเป็น "0" สำหรับส่วนที่สว่าง และเอาไฟพุ่ง เป็น

"1" สำหรับส่วนที่มืด ดังนั้นแสดงจุดเมื่อเป็น "0" และไม่แสดงจุดเมื่อเป็น "1" โยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโหมด 2 ในการที่จะแสดงหนึ่งจุดนั้นมีการเอาข้อมูล 40 ครั้ง โดยจะนับจำนวนครั้งของ "1" ที่เป็นข้อมูลเข้ามา ดังนั้นจำนวนครั้งจะอยู่ระหว่าง 0-39 และเพื่อที่จะทำให้สามารถแสดงค่าได้ 5. ระดับ จึงต้องหารด้วย 8 จึงได้ค่า 0 - 4 และในความเข้มของระดับ 0-4 นี้แสดงได้ดังรูปที่ 9



รูปที่ 9 แสดงค่าความเข้มของภาพในระดับ 0-4

3 ส่วนของการเปลี่ยนโหมดจอภาพบนการ์ดเออร์คิวลิส

จากที่ได้กล่าวมาแล้วในส่วนก่อน ๆ ถึงโครงสร้างและการเข้าถึง พอร์ตของการ์ดเออร์คิวลิส เราสามารถทำการเขียนโปรแกรมเปลี่ยนโหมดการแสดงผลของจอภาพได้ดังผังงานต่อไปนี้



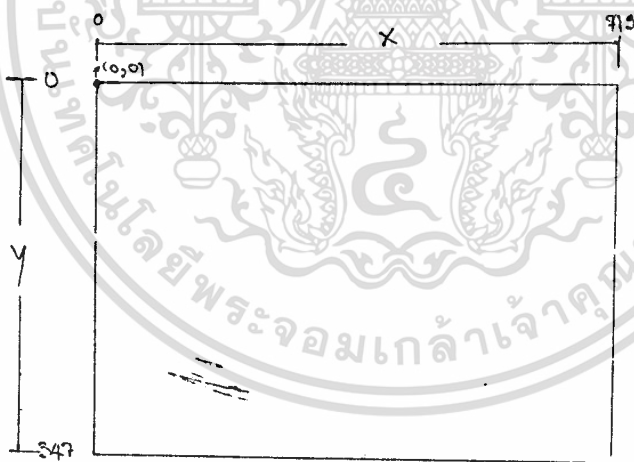
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 ส่วนของการเขียนจุดลงบนส่วนต่าง ๆ ของจอภาพ

เนื่องจากตำแหน่งความจำของการ์ดเออร์คิวลิสที่จัดไว้แบบต่อเนื่องและ ค่าของหน่วยความจำของแต่ละแถวบนจอภาพไม่ได้ยึดติดกัน ดังได้กล่าวมาแล้วในบทที่ 2 ในการที่เราจะทำการเขียนจุดใด ๆ ให้ติดหรือดับลงบนจอภาพในตำแหน่งที่ต้องการจำเป็นที่จะต้องรู้ค่าของตึกแห่งบนหน่วยความจำที่ตำแหน่งนั้น ๆ ก่อนจึงสามารถทำได้

สำหรับการ์ดเออร์คิวลิสแสดงจุดภาพในแนวนอนได้ 720 จุด และ 348 จุดภาพในแนวตั้งการกำหนดจุดภาพทำในหน่วยความจำที่เป็นจอภาพที่หน่วยความจำ 1 ไบท์คือจุดภาพมี 1 บิตต่อ 1 จุด ถ้าบิตใดมีค่า 1 จะสว่าง คือบังเกิดเป็นจุดภาพ ถ้ามีค่าเป็น 0 จะมืด

แต่การอ้างแอดเดรสถึงจุดภาพแต่ละจุดเป็นเรื่องยุ่งยากพอสมควร เพราะในโปรแกรมเราจะกำหนดจุดต่าง ๆ เหล่านี้เป็นค่า x, y โดยที่ x มีค่าระหว่าง 0-719 และ y มีค่าระหว่าง 0-347 โดยมีค่าจุด 0,0 อยู่ที่มุมบนซ้ายดังรูปที่ 10



รูปที่ 10 แสดงแกน x และแกน y ในการแสดงภาพแบบกราฟิกซึ่งจะมีจุดภาพทั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอรรถประโยชน์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 หมด $720 * 348 = 250,560$ จุดภาพ
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

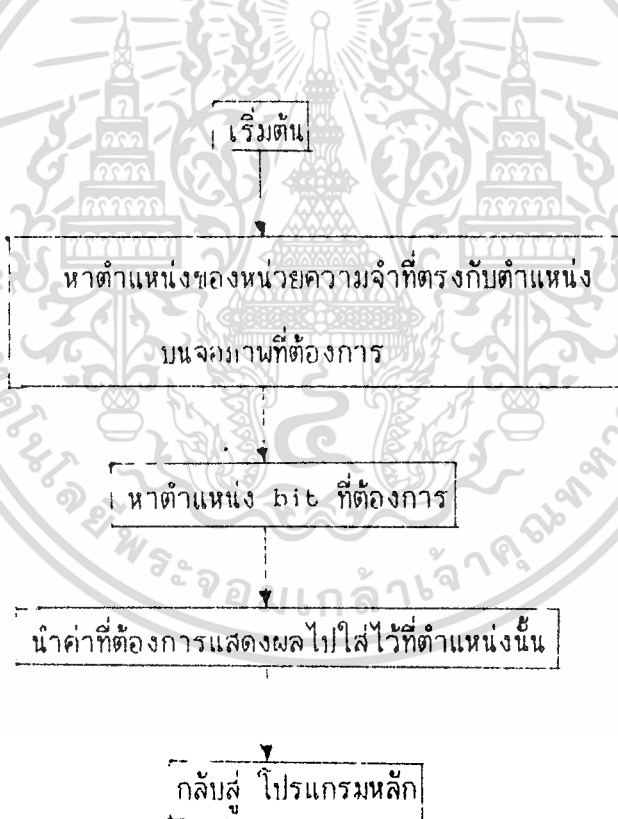
ในการหาที่อยู่ของหน่วยความจำสำหรับตำแหน่งบนจอภาพนั้นเราสามารถหาได้จากสูตรดังต่อไปนี้

```
offset := #2000 * (y mod 4) + (90 * (y div 4) ) +
(x div 8 );
```

```
bitnum := x mod 8 ;
```

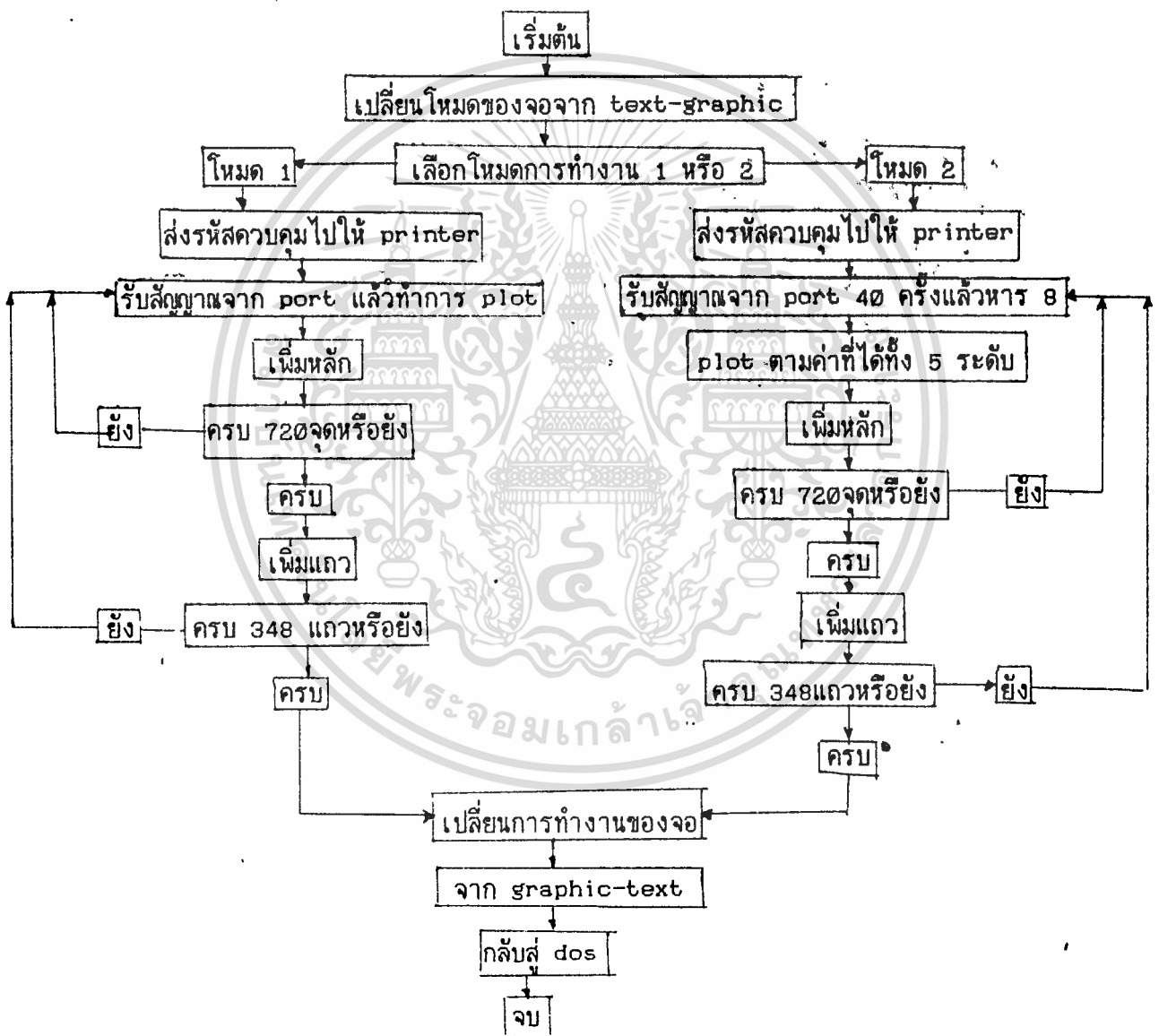
ในหน่วยความจำ 1 ไบต์ จะแทนค่าของจุด 8 จุด บนหน้าจอตั้งนั้นเมื่อทราบแอดเดรสแล้วก็จำเป็นจะต้องทราบค่าตำแหน่งของบิตในไบต์ นั้นด้วยซึ่งทำได้โดยการหาค่า bitnum

และจากที่กล่าวมาเราสามารถเขียนผังงานได้ดังนี้



5 ส่วนการอ่านจุดจากตำแหน่งบนจอภาพ

ในส่วนนี้มีหลักการคล้ายกับในส่วนวิธีเขียนจุดคือต้องหาตำแหน่ง offset ที่อยู่ของหน่วยความจำที่ตรงกับตำแหน่งบนจอเสียก่อน แล้วจึงทำการหาตำแหน่งบิตที่อยู่ในขณะนั้นหลังจากนั้นก็ค่อยนำค่าที่ได้ ส่งคืนโปรแกรมหลัก สำหรับโปรแกรมในการอ่านจุด แสดงไว้ในส่วนโปรแกรมย่อยที่มีชื่อว่า read_dot



ผังงานของ program ที่ใช้ในการรับสัญญาณและแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

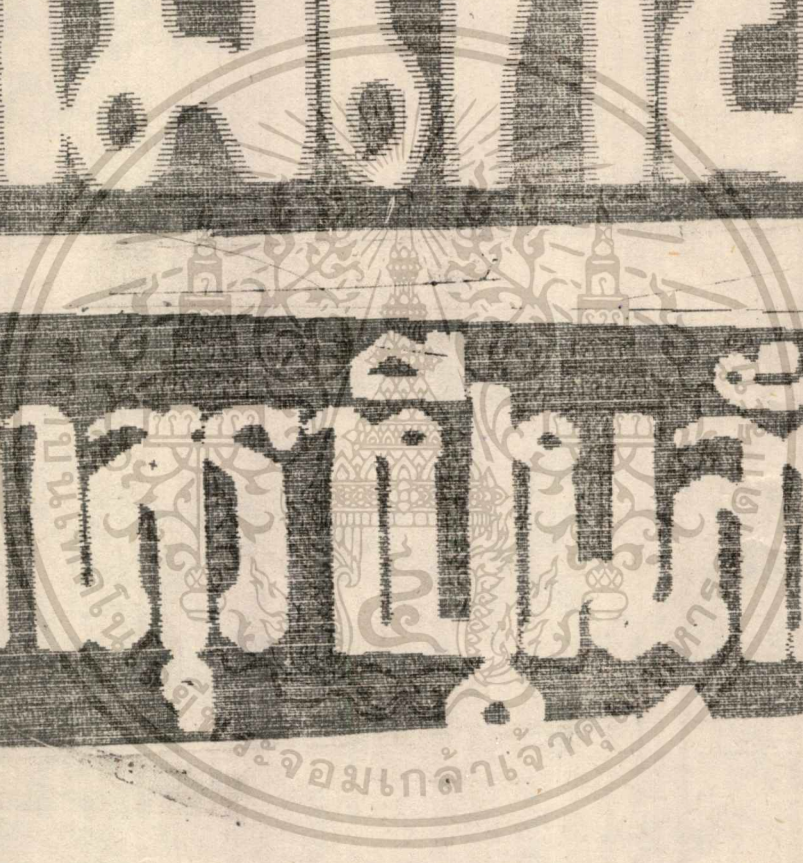
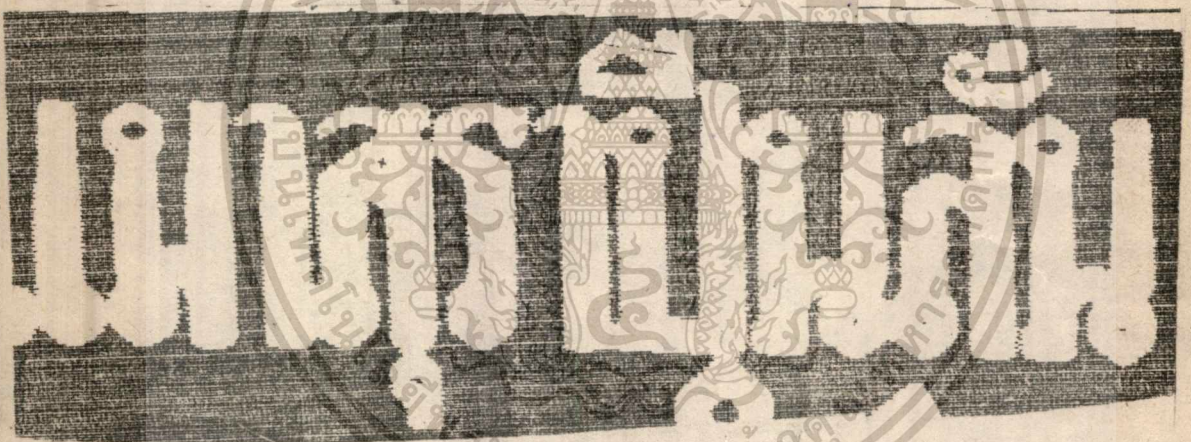
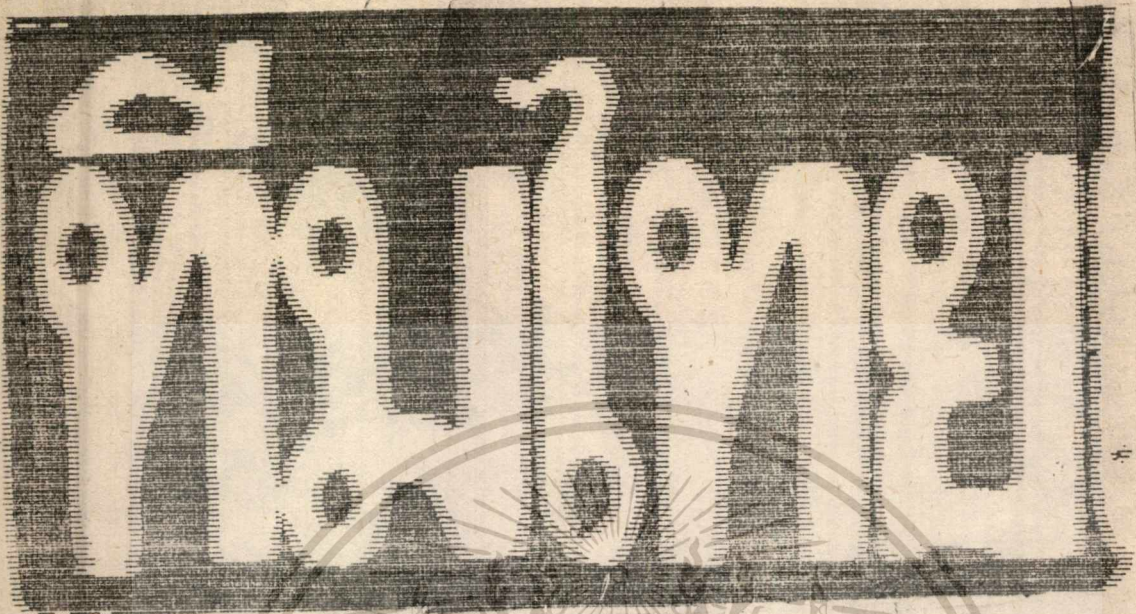
บทที่ 4

ผลการทดลอง

ในส่วนของตัวรับสัญญาณภาพนั้น เราได้นำไปติดที่หัวอ่านของเครื่องพิมพ์และทำการเสกภาพนำสัญญาณทางไฟฟ้าที่ได้ส่งเข้าไปยัง ส่วนวงจร a/d converter เพื่อเปลี่ยนให้เป็นสัญญาณทางดิจิตอลแล้วนำสัญญาณที่ได้ไปประมวลผลที่คอมพิวเตอร์ แล้วนำเอาผลที่ได้มาแสดงที่จอซึ่งได้ภาพออกมาคล้ายภาพที่นำมาเสกมาก จากนั้นได้ทำการเอาภาพที่ได้มาพิมพ์ออกทาง เครื่องพิมพ์ซึ่ง ได้ผลดีมาก ส่วนภาพตัวอย่างที่ได้ทดลองทำการเสกแล้วนั้น ได้นำมาแสดงที่หน้าถัดไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพผลการทดลองจากโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพผลการทดลองจากโหมต 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับภาพลักษณ์ที่ดลใจจากหม่อมตม ๒ อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพผลการทดลองจากโหมต 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



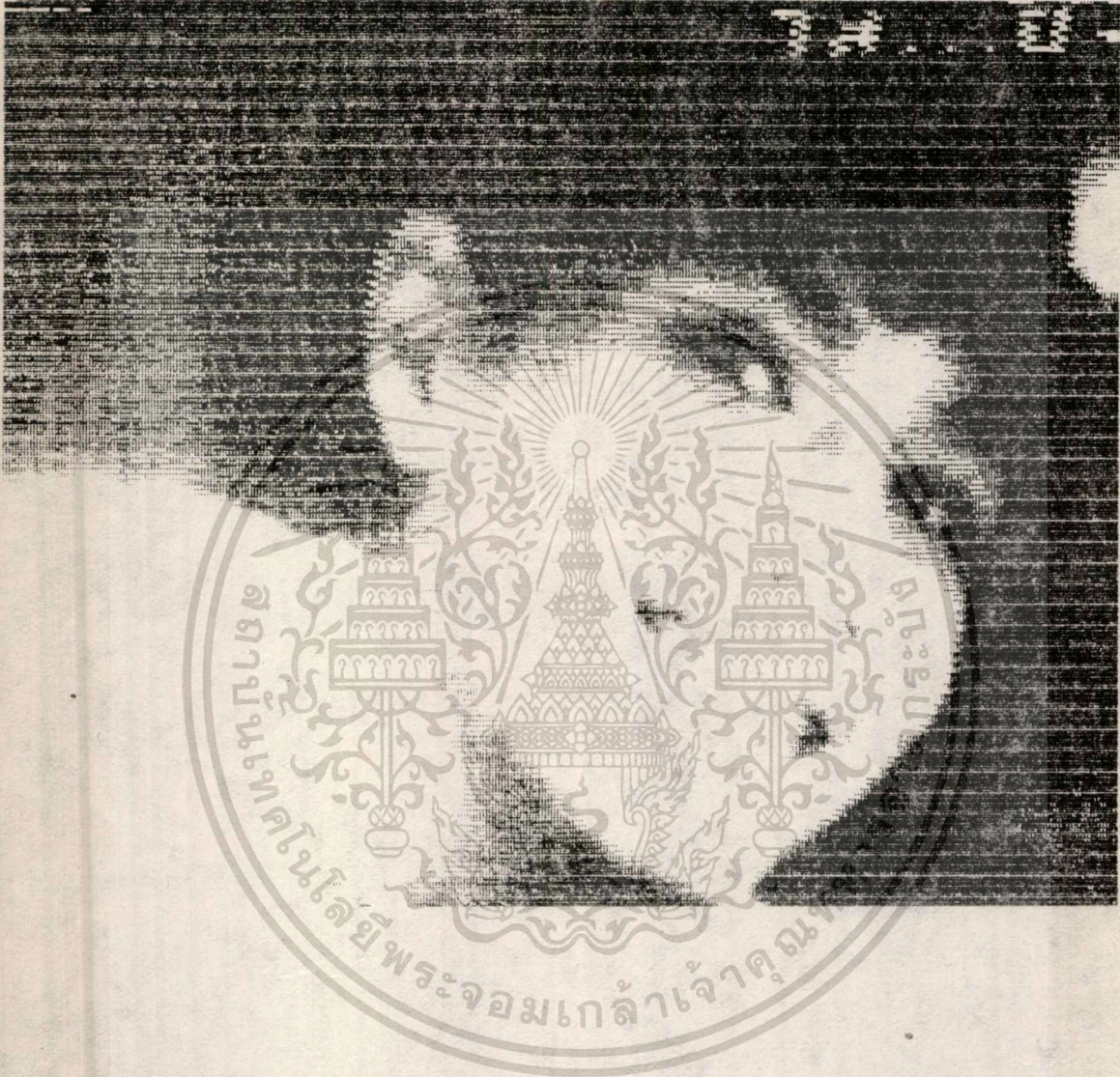
ภาพผลการทดลองจากโหมด 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพผลกวาดตลบจากไมโครฟิล์ม 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพผลการทดลองจากโหมด 2.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุปผล

ในโครงการนี้ในส่วนของตัว sensor อยู่ในเกณฑ์ที่ดีแต่อาจมีค่าผิดพลาดเกิดขึ้นได้เพราะเราใช้อุปกรณ์ที่หาได้ง่ายและราคาถูก ในส่วนของวงจร a/d converter ได้ผลดีในส่วนการต่อกันระหว่าง sensor และ a/d converter ก็ได้ผลที่ดี และในส่วนของการเขียนโปรแกรมนั้นสามารถใช้ความคุมส่วนต่าง ๆ ได้ตามจุดประสงค์ที่ต้องการ และในการต่อส่วนต่าง ๆ เข้าด้วยกันทั้งหมดก็ได้ผลที่ดีเป็นที่น่าพอใจมาก

ในโครงการนี้มีคุณสมบัติที่ดีคือ อุปกรณ์ที่ใช้มีราคาถูก และหาซื้อได้ง่ายและอุปกรณ์ที่ใช้เป็นตัวควบคุมส่วนนั้นก็คือเครื่องพิมพ์ซึ่งสามารถหามาใช้ได้ง่าย อีกทั้งยังไม่ต้องมีการปรับปรุงและแก้ไขเครื่องพิมพ์เลย และยังต่อเข้ากับเครื่องคอมพิวเตอร์ได้อย่างสะดวกเพราะมีสายที่ใช้ในการต่อเข้าเพียง 3 เส้นเท่านั้น และยังสามารถปรับระดับของสัญญาณได้อีกด้วย

แต่มีสิ่งที่ควรได้รับการปรับปรุงแก้ไขก็คือ ไม่สามารถที่จะรับภาพที่มีสีแดงได้ และสีอื่น ๆ ก็สามารถแสดงได้เพียงค่าระหว่างขาวกับดำ อีกทั้งความละเอียดของภาพที่ได้ยังไม่มากพอและในส่วนของการ scan เราใช้เครื่องพิมพ์เป็นตัวควบคุมสัญญาณจึงทำให้มีความล่าช้าในการแสดงภาพ

ภาคผนวก

page 132

```

;*****
;* Program      : image scanner      *
;* graphics mode: high resolution mode 720 x 348 dots *
;* written by   : Mr. kanungdej ,Mr. narongsukdi   *
;* use with     : hercules card ,momochrome graphic *
;* date        : december 1 , 1988      *
;*****
stack      segment      para stack 'stack'
           dw           64 dup(?)
stack      ends
;*****
code       segment      public
           assume      cs:code,ss:stack
           display     equ 2h
           doscall     equ 21h
           count       equ 49d
           g_mode      equ 0006h
           t_mode      equ 0007h
           esc         equ 27
           LF          equ 0ah
           CR          equ 0dh
data       segment      public
dummy     db           0

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gtable      db  35h,2dh,2eh,07h

             db  5bh,02h,57h,57h

             db  02h,03h,00h,00h

ttable      db  61h,50h,52h,0bh

             db  19h,06h,19h,19h

             db  02h,0dh,0dh,0ch

string      db  'image scanner ,please select'

             db  'mode of scan (1/2) ',0dh,0ah

extrn  row_p:word

extrn  row:word

extrn  column:word

data      ends

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

extrn  read_dot:near

extrn  g_screen:near

extrn  t_screen:near

extrn  plot_dot:near

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

main      proc  far

             jmp  short  start

             assume  ds:data

start:  ; _____ ;

             ; set screen to high resolution graphics mode ;

             ;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sub     ax,ax

push   ax

;-----;

;set 8/216-inch line spacing of printer

;

mov    dl,esc           ;send esc;3;8 to lpt0

mov    ah,5

int    21h             ;call dos

mov    dl,'3'

int    21h

mov    dl,8

int    21h

;-----;

;initail row&column

;

mov    row,0000

mov    column,0000

push  ds

mov    ah,6

mov    al,0

mov    ch,0

mov    cl,0

mov    dh,24

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    bh,7

int    10h

mov    ah,2

mov    dh,12

mov    dl,0

mov    bh,0

int    10h

mov    ax,data
mov    ds,ax
mov    cx,count
mov    bx,offset string

nextchr: mov    dl,[bx]
mov    ah,display
int    doscall
inc    bx

loop  nextchr

pop    ds

n:     mov    ah,0

int    16h

cmp    al,'1'

jz     mode_1

```

เอกสารนี้เป็นเอกสารที่ **comp** ไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    jz     mode2

    jmp    n

mode_1: call  g_screen

    mov    ax,0b800h

    mov    es,ax

    xor    di,di

    mov    cx,4000h

    mov    ax,0ffffh
    rep    stosw

    mov    ah,0

    int    16h

    mov    ax,0000

    xor    di,di

    mov    cx,4000h
    rep    stosw

```

```

start0: call  scan

    get:   sub    ax,ax

    mov    dx,201h

    in     al,dx

    and    al,10000000b

    rol    al,1

```

เอกสารนี้เป็นเอกสารที่ call ไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inc    column

cmp    column,720

jb     get

mov    column,0

inc    row

cmp    row,348

jb     start0

jmp    g_print

mode2: call  g_screen

mov    ax,0b800h

mov    es,ax

xor    di,di

mov    cx,4000h

mov    ax,0ffffh

rep    stosw

mov    ah,0

int    16h

xor    di,di

mov    ax,0000h

mov    cx,4000h

rep    stosw

```

```
start1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
call scan
```

```
get_it:
```

```
-----;
```

```
; get data from the photo sensor and change to 5 level ;
```

```
-----;
```

```
sub ax,ax
```

```
sub bx,bx
```

```
mov cx,40 ;39 time to get data ;
```

```
mov dx,201h ;port input. ;
```

```
loop: in al,dx ;in data from port ;
```

```
and al,10000000b ;
```

```
rol al,1 ;
```

```
add al,b1 ;
```

```
mov bl,al ;now input data is in bx ;
```

```
loop loop ;do 39 time ;
```

```
shr bl,1 ;
```

```
shr bl,1 ;divide by 8 ;
```

```
shr bl,1 ;
```

```
mov al,bl ;now signal level 0-4 is ;
```

```
; in ax ;
```

```
-----;
```

```
; check scanning line ;forward or reverse
```

```
;
```

```
push ax ;save signal level in ax
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการสื่อสารเท่านั้น ไปจนออกให้วงไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov  ax,row          ;get row
shr  ax,1
and  ax,1            ;is ti reverse scanning
jnz  reverse        ;yes
pop  ax              ;no

```

```

; plot data in ax onto screen
;

```

```

call ff_plot
call wait
inc  column
cmp  column,720      ;is it end of line?
jne  get_it         ;no,plot next dot
dec  column         ;yes,
jmp  inc_row

```

```
reverse:
```

```

pop  ax
call rew_plot
call wait
cmp  column,0       ;is it end of line?
je   inc_row       ;yes
dec  column        ;no
jmp  get_it        ;plot next dot

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

inc_row: ; mov    column,0

        inc    row

        inc    row

        cmp    row,348        ;is it compleat

        jnb   start1        ;if not goto start again

;-----;

; wait for a key stroke ;

; change screen back to text_mode ;

; return to dos ;

; ;

g_print: mov    ah,0

        int    16h        ;key board check

;-----;

; graphics print screen ;

;-----;

; initialize the starting row & column number

;

        mov    row_p,0000

        mov    column,0000

;-----;

        mov    al,esc

        call   print

        mov    al,'3'

        call   print

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov    al,24

call  print

;-----;

; set the printer into the bit image mode
;

nextRow: mov    al,esc

call  print

mov    al,'L'

call  print

mov    al,720 mod 256

call  print

mov    al,720/256

call  print

;-----;

mov    column,0000

;-----;

; set a bit-counter & clear a dummy byte
;

nextCol: mov    cx,8

mov    dummy,00

nextDot: push  cx

shl   dummy,1

call  read_dot

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

je    short notDot

or    dummy,01h

notDot: inc row_p

pop cx

loop nextDot

mov  al,dummy

call print

inc  column

cmp  column,720

jne  short nextCol

;-----
; mov the printer head to the new line
;
mov  al,CR
call print

mov  al,LF
call print

add  row_p,8

cmp  row_p,348

jbe  nextRow

;-----
; restore the 1/6 inch line space
;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้วนเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        call print
        mov  al,'2'
        call print

;-----

call  t_screen

;change to text mode

mov   ah,0

mov   al,0007h

int   10h

mov   ax,4c00h

int   21h      ;back to dos

;-----

main   endp

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

print  proc  near

        push  dx

        mov   ah,00

        mov   dx,00

        int   17h

        pop   dx

        ret

print  endp

;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

push    cx
push    dx
mov     cx,01h

loopw2:
mov     dx,7fh

loopw1:
dec     dx
jnz     loopw1
dec     cx
jnz     loopw2
pop     dx
pop     cx
ret

wait    endp

```

.....

```

scan    proc    near

```

-----;

; procedure to scan printer head ;

-----;

```

mov     ah,5

```

```

mov     dl,'.'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ 21h ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov     cx,78

loops:

mov     dl,' '      ; one blank line

int     21h

loop   loops

```

```

mov     dl,','

int     21h

mov     dl,LF      ; linefeed

int     21h

mov     dl,CR      ; carriage return

int     21h

ret

```

```

scan   endp

```

.....

```

ff_plot  proc      near

```

```

;-----;
```

```

; pocedure to write dot onto screen ;
```

```

;-----;
```

```

; ;
```

```

; change value in ax to number of 1 in ax ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov     cx,ax      ;get data in ax

mov     ax,0000h   ;initial ax

cmp     cx,0

je      npt

next_dt:

rol     ax,1

add     ax,1

dec     cx

jnz     next_dt

;-----;
;plot in 5 level 0-4 as the number of 1 in ax ;
;-----;

npt:    mov     dx,4

plot:   push    dx

        push   ax

        and    ax,0001

        call   plot_dot

        pop    ax

        ror    ax,1      ;next dot to write

        pop    dx

        dec    dx

        jz     return    ;if completed return

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        jl      right
        inc    row
        jmp    plot
right:   cmp    dx,2
        jl    up
        inc    column
        jmp    plot
up:     dec    row
        jmp    plot
return: ret
ff_plot endp
;.....;
rew_plot proc near
;-----;
;reverse period dot writting pocedure ;
;-----;
;change value in ax to number of 1 in ax ;
; ;
        mov   cx,ax           ;details as ff_plot
        mov   ax,0000h
        cmp   cx,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

next_dg:

    rol    ax,1

    add    ax,1

    dec    cx

    jnz    next_dg

rt:

    mov    dx,4                ;numbes of dot to plot

    ;-----;
;plotting dot in reverse scanning line ;
; ;

plot1:

    push  dx

    push  ax

    dec   dx

    jz    mask                ;is it last digit

    mov   cx,dx

    ror   ax,c1                ;rotate ax 3-1 times

mask:

    and   ax,0001h            ;now ax have shifted right

    call  plot_dot            ;and the digit to write is in al

                                ;now fist dot have written

    pop   ax

    pop   dx

    dec   dx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่ควรเผยแพร่ไปสาธารณะโดยไม่ได้รับอนุญาต
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cmp    dx,3
        jl     left
        inc    row            ;2nd dot
        jmp    plot1         ;plot under 1st dot
left:    cmp    dx,2
        jl     up
        dec    column        ;3rd dot
        jmp    plot1         ;plot left of 2nd dot
up1:    dec    row            ;4th dot
        jmp    plot1         ;plot above the 3rd dot
return_r:
        ret     ;return to main proc
rew_plot endp
;::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::;
;*****
code    ends
        end    main
;*****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

program test.asm

```
index equ 03b4h
datar equ 03b5h
cntrl equ 03b8h
consw equ 03bfh
graph equ 02h
text equ 20h
graphon equ 0ah
texton equ 28h
video equ 0b800h
;#####
stack segment para stack 'stack'
dw 64 dup(?)
stack ends
;#####
code segment public
public read_dot
public g_screen
public t_screen
public plot_dot
;#####
data segment public
public row,column,row_p
row_p dw 0
row dw 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

column    dw    0
ydiv4     dw    0
ymod4     dw    0
xdiv8     dw    0
xmod8     dw    0
offsets   dw    0

```

```
data      ends
```

```
#####
```

```
assume    cs:code,ds:data,ss:stack
```

```
plot_dot
```

```
proc      near
```

```
push     ax
```

```
push     bx
```

```
push     cx
```

```
push     dx
```

```
pushf
```

```
mov      si,ax
```

```
xor      dx,dx
```

```
mov      ax,column
```

```
mov      bx,8
```

```
div      bx
```

```
mov      xmod8,dx    ;store remainder in dx (x mod 8)
```

```
mov      xdiv8,ax    ;resalt in ax (x div 8)
```

```
xor      dx,dx
```

```
mov      ax,row
```

```
mov      bx,4    ;bx = 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวารใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

div    bx

mov    ymod4,dx          ;store (y mod 4)

mov    bx,90d

mul    bx                ;(y div 4)*90d result in ax

add    ax,xdiv8 ; ((y div 4)*90 + (x div 8))

mov    offsets,ax

mov    ax,ymod4 ;

mov    bx,2000h

mul    bx                ; 2000h*(y mod 4) result in ax

add    ax,offsets ; 2000h*(y mod 4)+((y div 4)*90

mov    di,ax             ; +(x div 8))

mov    bx,video

mov    es,bx

cmp    si,1

jne    nodot

mov    al,100000000b

jmp    ok

nodot: mov    al,0

ok:    mov    cx,xmod8

ror    al,cl

or     al,es:[di]

mov    es:[di],al

popf

pop    dx

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    pop    bx

    pop    ax

    ret

plot_dot    endp

```

;#####

```

g_screen    proc    near

```

```

    xor    ax,ax

```

```

    mov    dx,03bfh

```

```

    mov    al,03h

```

```

    out    dx,al

```

```

    mov    dx,03b8h

```

```

    mov    al,82h

```

```

    out    dx,al

```

```

    mov    dx,03b4h

```

```

    xor    ax,ax

```

```

    out    dx,al

```

```

    inc    dx

```

```

    mov    al,35h

```

```

    out    dx,al

```

```

    dec    dx

```

```

    mov    al,1

```

```

    out    dx,al

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mov    al,5
out    dx,al
inc    dx
mov    al,02h
out    dx,al
dec    dx
```

```
mov    al,6
out    dx,al
inc    dx
mov    al,57h
out    dx,al
dec    dx
```

```
mov    al,7
out    dx,al
inc    dx
```

```
mov    al,57h
out    dx,al
dec    dx
```

```
mov    al,8
out    dx,al
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mov    al,02h
```

```
out    dx,al
```

```
dec    dx
```

```
mov    al,9
```

```
out    dx,al
```

```
inc    dx
```

```
mov    al,03h
```

```
out    dx,al
```

```
dec    dx
```

```
mov    al,10
```

```
out    dx,al
```

```
inc    dx
```

```
mov    al,00h
```

```
out    dx,al
```

```
dec    dx
```

```
mov    al,11
```

```
out    dx,al
```

```
inc    dx
```

```
mov    al,00h
```

```
out    dx,al
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับศูนย์ 4000 ที่การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mov ax,0b800h
```

```
mov es,ax
```

```
xor di,di
```

```
mov ax,0000h
```

```
cld
```

```
rep stosw
```

```
mov dx,03b8h
```

```
mov al,8ah
```

```
out dx,al
```

```
mov cx,4000h
```

```
xor di,di
```

```
mov ax,0ffffh
```

```
rep stosw
```

```
mov ah,0
```

```
int 16h
```

```
ret
```

```
g_screen endp
```

```
#####
```

```
t_screen proc near
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out dx,a1
```

```
dec dx
```

```
mov al,2
```

```
out dx,a1
```

```
inc dx
```

```
mov al,52h
```

```
out dx,a1
```

```
dec dx
```

```
mov al,3
```

```
out dx,a1
```

```
inc dx
```

```
mov al,0fh
```

```
out dx,a1
```

```
dec dx
```

```
mov al,4
```

```
out dx,a1
```

```
inc dx
```

```
mov al,19h
```

```
out dx,a1
```

```
dec dx
```

```
mov al,5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
out dx,a1
inc dx
mov al,06h
out dx,a1
dec dx
```

```
mov al,6
out dx,a1
inc dx
mov al,19h
out dx,a1
dec dx
```

```
mov al,7
out dx,a1
inc dx
mov al,19h
```

```
out dx,a1
dec dx
```

```
mov al,8
out dx,a1
inc dx
```

```
mov al,02h
```

```
out dx,a1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dec dx  
  
mov al,9  
out dx,al  
inc dx  
mov al,0dh  
out dx,al  
dec dx
```

```
mov dx,10  
out dx,al  
inc dx  
mov al,0bh  
out dx,al  
dec dx
```

```
mov al,11  
out dx,al
```

```
inc dx  
mov al,0ch  
out dx,al
```

```
mov dx,03b8h
```

```
mov al,28h
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    popf

    pop    dx

    pop    cx

    pop    bx

    pop    ax

    ret

t_screen    endp

;#####

read_dot    proc    near

    push    bx
    push    cx
    push    dx
    pushf
    xor     dx,dx
    mov     ax,column
    mov     bx,8
    div     bx

    mov     xmod8,dx

    mov     xdiv8,ax

    xor     dx,dx

    mov     ax,row_p

    mov     bx,4

    div     bx

    mov     ymod4,dx

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mul     bx

add     ax,xdiv8

mov     offsets,ax

mov     ax,ymod4

mov     bx,2000h

mul     bx

add     ax,offsets

mov     di,ax

mov     bx,video

mov     es,bx

mov     al,10000000b

mov     cx,xmod8

ror     al,cl

and     al,es:[di]

rol     al,cl

ror     al,1

popf

pop     dx

pop     cx

pop     bx

ret

read_dot     endp

```

#####

เอกสารนี้เป็น code ที่สงวน ends หรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end

#####



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

QUAD COMPARATORS

LM139/139A, LM239/239A, LM339/339A

FEATURES

- Wide single supply
Voltage range – 2 VDC to 36 VDC
or dual supplies – ± 1 VDC to ± 18 VDC
- Very low supply current drain (0.8 mA) – independent of supply voltage (1 mW/comparator at +5 VDC)
- Low input biasing current – 35 nA
- Low input offset current – 3 nA
and offset voltage – 2 mV
- Input common-mode voltage range includes ground
- Differential input voltage range equal to the power supply voltage
- Low output saturation voltage – 1 mV at 5 μ A – 70 mV at 1 mA
- Output voltage compatible with TTL (fanout of 2), DTL, ECL, MOS and CMOS logic systems

ADVANTAGES

- Eliminates need for dual supplies
- Allows sensing near GND
- Compatible with all forms of logic
- Power drain suitable for battery operation

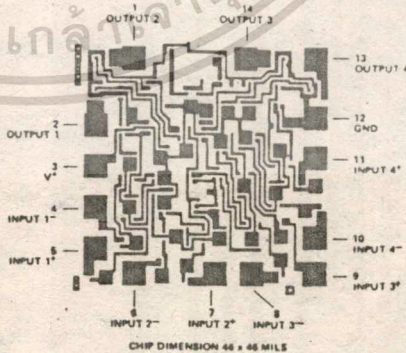
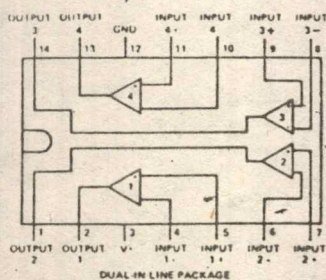
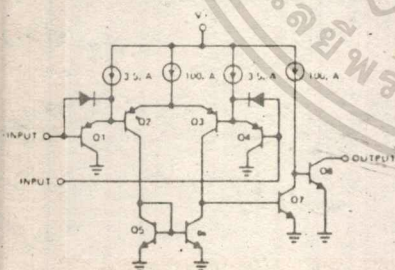
GENERAL DESCRIPTION

The LM139 series consists of four independent voltage comparators which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multi-vibrators and high voltage digital logic gates. The LM139 series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM339 will directly interface with MOS logic – where the low power drain of the LM339 is a distinct advantage over standard comparators.

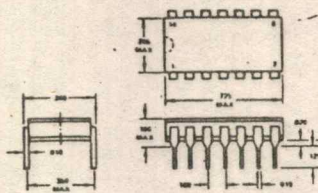
SCHEMATIC AND CONNECTION DIAGRAMS

CHIP TOPOGRAPHY

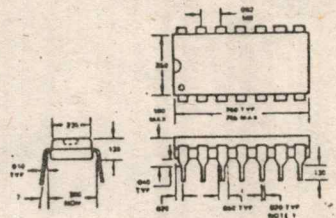


PACKAGE DIMENSIONS

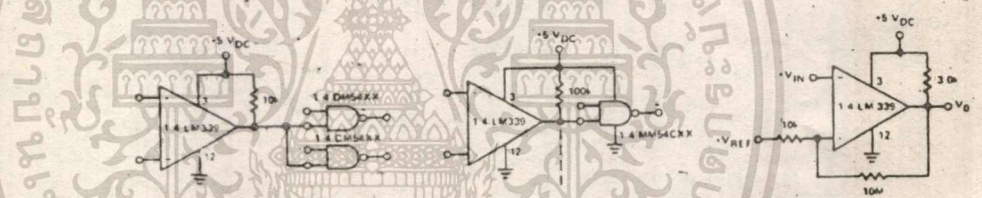
14 PIN CERDIP



14 PIN DIP



TYPICAL APPLICATIONS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำ

LINEAR INTEGRATED CIRCUITS

TYPES SN52555, SN72555 PRECISION TIMERS

BULLETIN NO. DL-S 7312053, SEPTEMBER 1973

- Timing from Microseconds to Hours
- Astable or Monostable Operation
- Adjustable Duty Cycle
- Up to 200-mA Sink or Source Output Current
- TTL Compatible Output
- Designed to be Interchangeable with Signetics SE555/NE555

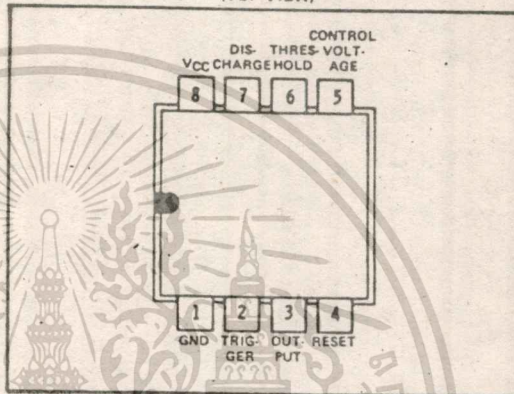
description

The SN52555 and SN72555 are monolithic timing circuits capable of producing accurate time delays or oscillation. In the time-delay or monostable mode of operation, the timed interval is controlled by a single external resistor and capacitor network. In the astable mode of operation, the frequency and duty cycle may be independently controlled with two external resistors and a single external capacitor.

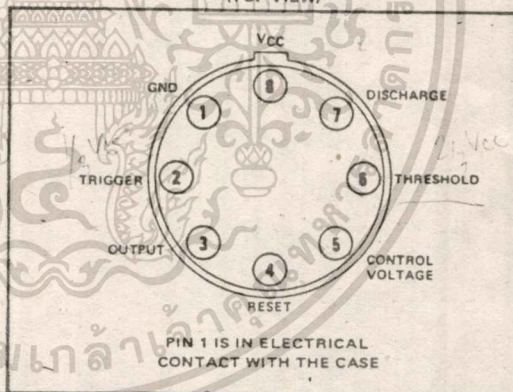
The threshold and trigger levels are normally two-thirds and one-third, respectively, of V_{CC} . These levels can be altered by use of the control voltage terminal. When the trigger input falls below the trigger level, the flip-flop is set and the output goes high. When the threshold input rises above the threshold level, the flip-flop is reset and the output goes low. The reset input can override all other inputs and can be used to initiate a new timing cycle. When the reset input goes low, the flip-flop is reset and the output goes low. When the output is low, a low-impedance path is provided between the discharge terminal and ground.

The output circuit is capable of sinking or sourcing current up to 200 milliamperes. Operation is specified for supplies of 5 to 15 volts. With a 5-volt supply, output levels are compatible with TTL inputs.

J_P OR P DUAL-IN-LINE PACKAGE (TOP VIEW)

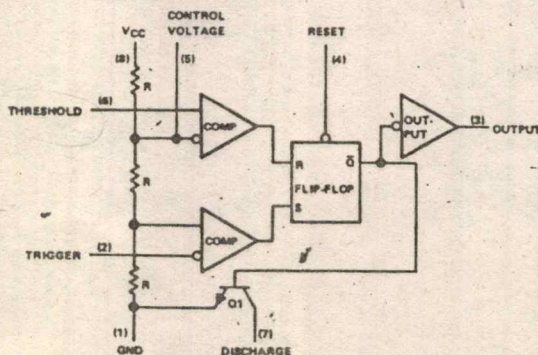


L PLUG-IN PACKAGE (TOP VIEW)



3

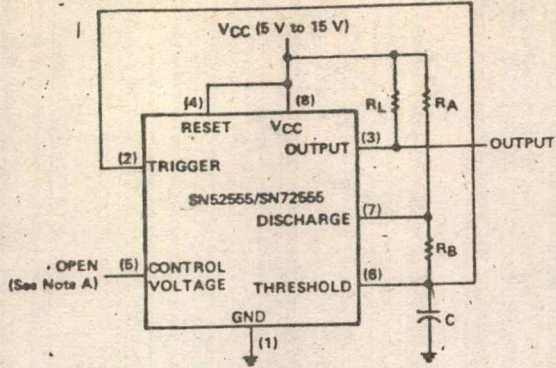
functional block diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA

astable operation



NOTE A: Decoupling the control voltage input (pin 5) to ground with a capacitor may improve operation. This should be evaluated for individual applications.

FIGURE 14—CIRCUIT FOR ASTABLE OPERATION

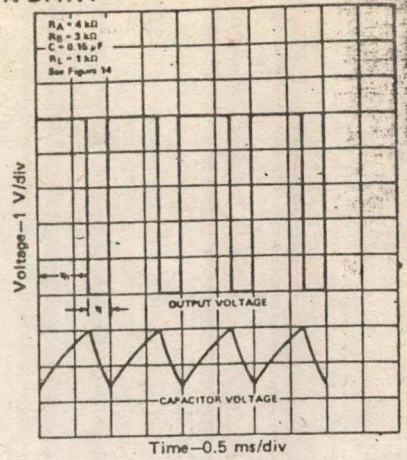


FIGURE 15—TYPICAL ASTABLE WAVEFORMS

Addition of a second resistor, R_B , to the circuit of Figure 11; as shown in Figure 14, and connection of the trigger input to the threshold input will cause the SN52555/SN72555 to self-trigger and run as a multivibrator. The capacitor C will charge through R_A and R_B then discharge through R_B only. The duty cycle may be controlled, therefore, by the values of R_A and R_B .

This astable connection results in capacitor C charging and discharging between the threshold-voltage level ($\approx 0.67 \cdot V_{CC}$) and the trigger-voltage level ($\approx 0.33 \cdot V_{CC}$). As in the monostable circuit, charge and discharge times (and therefore the frequency and duty cycle) are independent of the supply voltage.

Figure 15 shows typical waveforms generated during astable operation. The output high-level duration, t_h , is calculated as:

$$t_h = 0.693 (R_A + R_B)C,$$

output low-level duration, t_l , as:

$$t_l = 0.693 (R_B)C.$$

The total period is $T = t_h + t_l$ and frequency is

$$f = \frac{1}{T}, \text{ or } f = \frac{1.44}{(R_A + 2R_B)C}$$

The frequency of oscillation may be determined by referring to the chart shown in Figure 16, which relates free-running frequency, f , to the external resistors R_A and R_B and the external capacitor C . Duty cycle, D , is determined by the values selected for R_A and R_B and may be calculated as:

$$D = \frac{R_B}{R_A + R_B}$$

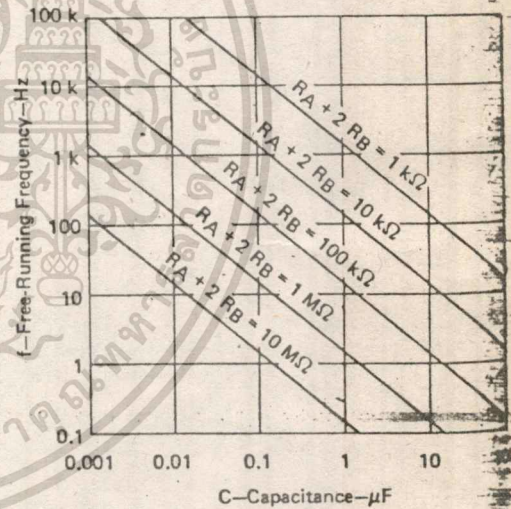


FIGURE 16—FREE-RUNNING FREQUENCY

TYPICAL APPLICATION DATA

missing-pulse detector

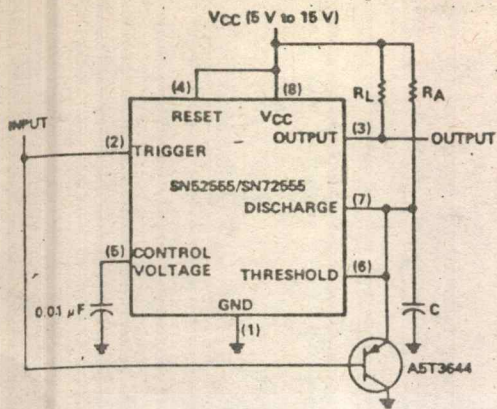


FIGURE 17—CIRCUIT FOR MISSING-PULSE DETECTOR

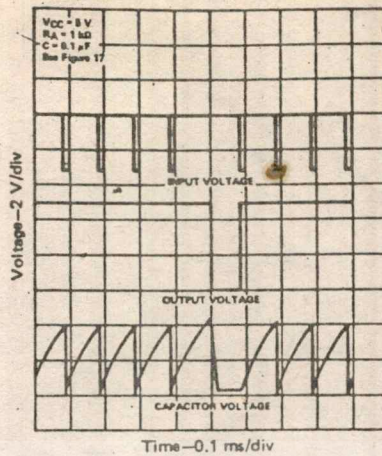


FIGURE 18—MISSING-PULSE-DETECTOR WAVEFORMS

The circuit shown in Figure 17 may be utilized to detect a missing pulse or abnormally long spacing between consecutive pulses in a train of pulses. The timing interval of the monostable circuit is continuously retriggered by the input pulse train as long as the pulse spacing is less than the timing interval. A longer pulse spacing, missing pulse, or terminated pulse train will permit the timing interval to be completed, thereby generating an output pulse as illustrated in Figure 18.

frequency divider

By adjusting the length of the timing cycle, the basic circuit of Figure 11 can be made to operate as a frequency divider. Figure 19 illustrates a divide-by-3 circuit that makes use of the fact that retriggering cannot occur during the timing cycle.

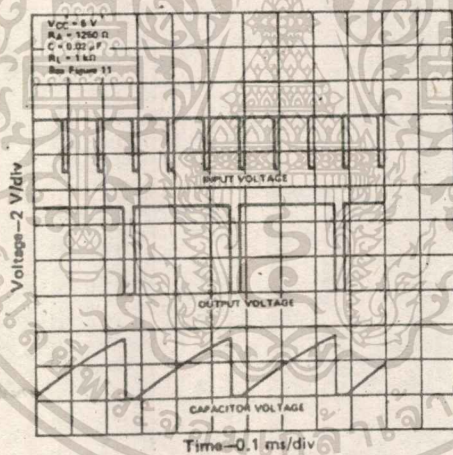
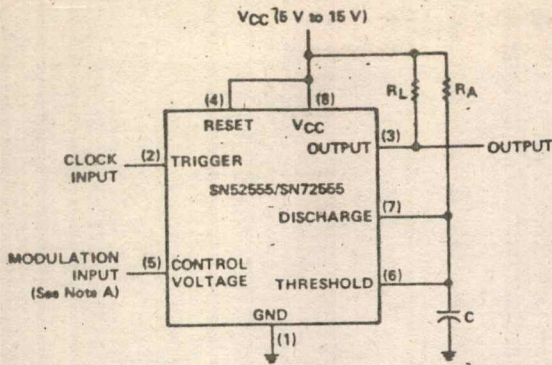


FIGURE 19—DIVIDE-BY-THREE CIRCUIT WAVEFORMS

TYPICAL APPLICATION DATA

pulse-width modulation



NOTE A: The modulating signal may be direct or capacitively coupled to the control voltage terminal. For direct coupling, the effects of modulation source voltage and impedance on the bias of the SN52555/SN72555 should be considered.

FIGURE 20—CIRCUIT FOR PULSE-WIDTH MODULATION

The operation of the timer may be modified by modulating the internal threshold and trigger voltages. This is accomplished by applying an external voltage (or current) to the control voltage pin. Figure 20 is a circuit for pulse-width modulation. The monostable circuit is triggered by a continuous input pulse train and the threshold voltage is modulated by a control signal. The resultant effect is a modulation of the output pulse width, as shown in Figure 21. A sine-wave modulation signal is illustrated, but any wave-shape could be used.

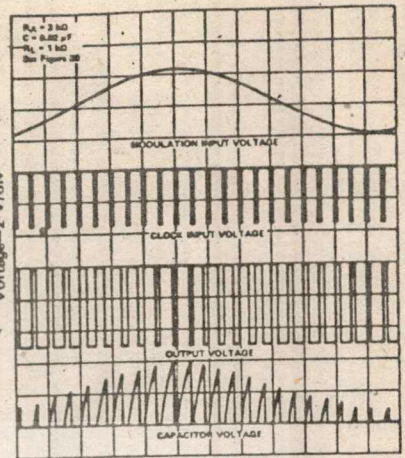
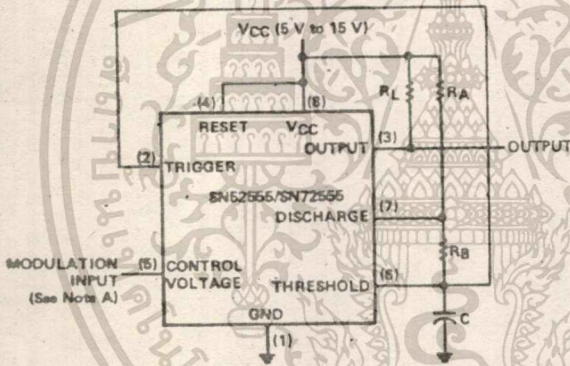


FIGURE 21—PULSE-WIDTH-MODULATION WAVEFORMS

pulse-position modulation



NOTE A: The modulating signal may be direct or capacitively coupled to the control voltage terminal. For direct coupling, the effects of modulation source voltage and impedance on the bias of the SN52555/SN72555 should be considered.

FIGURE 22—CIRCUIT FOR PULSE-POSITION MODULATION

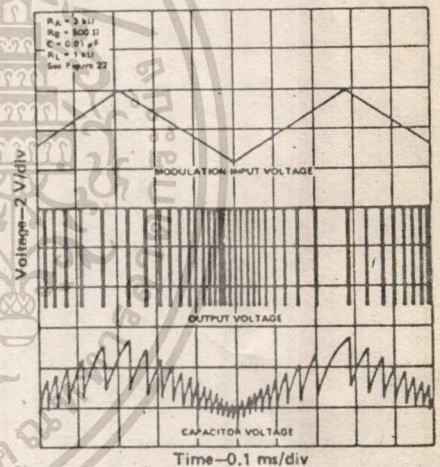


FIGURE 23—PULSE POSITION-MODULATION WAVEFORMS

The SN52555/SN72555 may be used as a pulse-position modulator as shown in Figure 22. In this application, threshold voltage, and thereby the time delay, of a free-running oscillator is modulated. Figure 23 shows such a circuit with a triangular-wave modulation signal, however, any modulating wave-shape could be used.

กิติกรรมประกาศ

ในโครงการนี้สามารถลุล่วงไปด้วยดี ก็เพราะได้รับคำแนะนำที่มีประโยชน์
จากอาจารย์ที่ปรึกษาทั้ง 2 ท่าน คือ ผศ. อภินันท์ มัชฌายานนท์ และ อ.สุทธิชัย นพนาศินพงษ์
จึงขอกราบขอบพระคุณเป็นอย่างสูง มา ณ.ที่นี้

ผู้จัดทำ

นาย คณิงเดช แซ่ลี

นาย ณรงค์ศักดิ์ สังขสังวาลย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. วารสาร "ไมโครคอมพิวเตอร์" ฉบับที่ 28
2. ทินกร ดูก, ชูชัย ธนสารตั้งเจริญ "อินไซด์ เดอะไอบีเอ็ม พีซี" 2529
3. JSK Group "แอสแซมบลี 8086/8088"
4. "IBM Technical Reference" 1983
5. Peter Abel "Assembler for IBM PC and PC/XT" 1983

