



ผศ.ดร.รัตติกร วรากุลศิริพันธ์

เลขหมาย	T. 33134 14
เลขทะเบียน	097967
วัน, เดือน, ปี	31 ก.ค. 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ (0) 27967

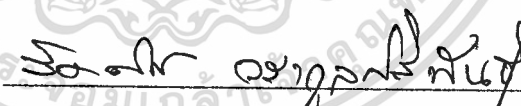
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ภาควิชาการศึกษาที่ 2 ปีการศึกษา 2533

เรื่อง เครื่องสแกนภาพ (IMAGE SCANNER)

ผู้จัดทำ

1. นาย กัมบนาท เรืองรายวัน 301010
2. นาย เกร็ดคณิต ศิริสภิตย์กุล 301033
3. นาย พงษ์ศักดิ์ พร้อมวงศ์ 301166


(ผศ. ดร. รัตติกอร์ วราภูลศิริพันธ์)

อาจารย์ที่ปรึกษา

ชื่อภาษาไทย เครื่องสแกนภาพ
ชื่อภาษาอังกฤษ IMAGE SCANNER

นายกัมปนาท เรืองรายวัน 301010
นายเกรือคณิต ศิริสถิตย์กุล 301033
นายพงษ์ศักดิ์ พร้อมวงศ์ 301166

อาจารย์ที่ปรึกษา

ผศ.ดร.รัตติกร วรากุลศิริพันธุ์

บทคัดย่อ

เครื่องอ่านภาพ (IMAGE SCANNER) เป็นเครื่องมือสำหรับอ่านข้อมูลภาพเข้าสู่คอมพิวเตอร์ เพื่อใช้ในการแสดงผลและประมวลผล ซึ่งมีประโยชน์ในการเก็บข้อมูลที่ต้องการภาพประกอบ ในปัจจุบันเครื่องชนิดนี้เริ่มใช้กันอย่างแพร่หลาย แต่ยังมีราคาสูงอยู่ ครงงานนี้จึงประดิษฐ์ เครื่องอ่านภาพที่มีประสิทธิภาพพอสมควร ราคาถูกและใช้ง่าย เครื่องนี้ได้มีการออกแบบแทนสแกน และฮาร์ดแวร์ขึ้น โดยใช้ประกอบกับซอฟต์แวร์ ซึ่งทำเป็นเมนูเพื่อความสะดวกในการทำงาน เครื่องนี้สามารถอ่านข้อมูลภาพได้จากกระดาษหลายชนิด เก็บไว้บนแฟ้มข้อมูลในแผ่นดิสค์ แสดงผลบนจอมอนิเตอร์ และพิมพ์ออกทางเครื่องพิมพ์ได้ โดยสามารถปรับความเร็วและตำแหน่งที่จะเริ่มสแกนได้อีกด้วย ปัญหาของเครื่องคือ การอ่านภาพยังไม่ชัดเจนเท่าที่ควร อาจแก้ปัญหาได้โดยการเปลี่ยนหัวอ่าน แต่หัวอ่านที่ใช้ในครงงานนี้ก็อยู่ในขั้นที่พอจะใช้งานได้

IMAGE SCANNER

Kampanart Ruangraiwaan 301010

Khruakhanit Sirisathitgul 301033

Pongsak prompwong 301166

Dr.Rattikorn Warakulsiripan Advisor

1990

ABSTRACT

Image Scanner is an instrument for scanning picture to store in computer and convert it to digital signal for display and process. This instrument is very useful to which make use of the picture of image processing or the database which the picture is the important factor. Since the present, the image scanner is very expensive. This project is determined to invent a low cost, convenient and efficient image scanner.

From the experiment, this instrument can read picture from many kind of paper. This instrument can select area and speed. There are two type of storing picture and can print out from printer.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ	i
abstract	ii
บทหน้า	1
บทที่ 1 ส่วนประกอบของเครื่องสแกนภาพ	2
บทที่ 2 ทฤษฎีการทำงานของวงจรฮาร์ดแวร์	9
2.1 บล็อกไดอะแกรม	9
2.2 วงจรอินเตอร์เฟส	13
2.3 วงจรควบคุม	15
2.4 วงจรขับมอเตอร์	19
2.5 วงจรควบคุมมอเตอร์	21
บทที่ 3 ลักษณะการทำงานของหัวอ่าน	22
3.1 การทำงานของพอร์ตเกม	23
3.2 หลักการทำงานของหัวอ่าน	24
บทที่ 4 โปรแกรมควบคุมการทำงานของเครื่องสแกนภาพ	27
4.1 Menu File	27
4.2 Menu Display	27
4.3 Menu Scan	28
4.4 Menu Print	28
4.5 Menu Parameter	28
บทที่ 5 โปรแกรมควบคุมฮาร์ดแวร์และการแสดงผล	31
5.1 โปรแกรมควบคุมการสแกน	31
5.2 โปรแกรมพิมพ์ออกทางเครื่องพิมพ์	32
5.3 โปรแกรมการแสดงผลบนจอภาพ	34
5.4 โปรแกรม Demo	37

บทที่ 6 การทดลอง	52
บทที่ 7 สรุปผลการทดลองและวิจารณ์	56
ภาคผนวก	
ตัวโปรแกรมทั้งหมด	58
กิตติกรรมประกาศ	105
บรรณานุกรม	106



สารบัญรูปภาพ

รูป 1.1	แบบของโครงที่เคลื่อนที่ในแกน x	4
รูป 1.2	แบบของตัวเคลื่อนที่ในแกน x	5
รูป 1.3	ภาพด้านข้างของแท่นสแกน	6
รูป 1.4	ภาพด้านข้างของแท่นสแกน	7
รูป 1.5	ภาพด้านบนของแท่นสแกน	7
รูป 1.6	ภาพส่วนประกอบของโครงที่เคลื่อนที่ในแกน x	8
รูป 2.1	แผนผังวงจรส่วนฮาร์ดแวร์	10
รูป 2.2	แผนผังส่วน Home	11
รูป 2.3	บล็อกไดอะแกรมของเครื่องสแกนภาพ	12
รูป 2.4	แผนภูมิเวลาของสัญญาณควบคุมมอเตอร์	16
รูป 2.5	แผนภูมิเวลาของสัญญาณสแกน	17
รูป 2.6	แผนภูมิเวลาของการแลตช์ข้อมูล	18
รูป 2.7	บล็อกไดอะแกรมแสดงการต่อวงจรนับ	18
รูป 2.8	วงจรขับมอเตอร์	19
รูป 2.9	วงจรขับกระแส	20
รูป 2.10	วงจรซีเคานเซอร์ที่ใช้ในการกระตุ้นสองเฟส	21
รูป 3.1	แสดงลักษณะของออฟต์ดีเท็คเตอร์ที่ใช้เป็นหัวอ่าน	22
รูป 3.2	แสดงวงจรขยายกระแสจากหัวอ่าน	23
รูป 3.3	แสดงสัญญาณจากการอ่านเพื่อแปลงเป็นระดับ	24
รูป 3.4	แสดงวงจรของพอร์ทเกม	25
รูป 4.1	แสดงเมนูที่ใช้ควบคุมเครื่องสแกนภาพ	26
รูป 5.1	แสดงการวางตัวของเข็มของเครื่องพิมพ์	35
รูป 5.2	แสดงแพทเทิร์นที่ออกแบบ	36
รูป 5.3	แสดงการแทนค่าแพทเทิร์นด้วยตัวเลข	36
รูป 5.4	แสดงภาพที่ได้จากการสแกน	51
รูป 6.1	แสดงระยะระหว่างหัวอ่านกับภาพ	53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง 2.1 แสดงการอ้างพอร์ท	14
ตาราง 2.2 แสดงหน้าที่ของพอร์ทต่างๆ	14
ตาราง 2.3 แสดงหน้าที่ของสัญญาณต่างๆ	15
ตาราง 5.1 แสดงโหมดต่างๆในการพิมพ์ของ เครื่องพิมพ์	33



บทนำ

ในปัจจุบันที่ระบบคอมพิวเตอร์ได้เข้ามามีบทบาท และใช้งานกันอย่างแพร่หลายในธุรกิจและอุตสาหกรรมแขนงต่างๆเป็นอย่างมาก ซึ่งนับวันจะมีราคาถูกลงและมีขีดความสามารถสูงขึ้น ยิ่งในยุคที่ธุรกิจและอุตสาหกรรมกำลังขยายตัวอย่างรวดเร็ว ข่าวสารข้อมูลต่างๆจึงเป็นสิ่งที่มีความสำคัญยิ่ง ซึ่งระบบคอมพิวเตอร์ได้เข้ามามีบทบาทในการประมวลผลข้อมูล ซึ่งมีการประมวลข้อความ (Text processing) และได้มีการพัฒนามาสู่การประมวลผลด้วยรูปภาพ (Image processing) เนื่องจากสามารถสื่อความหมายได้ชัดเจนและง่ายต่อการทำความเข้าใจ และข้อมูลบางอย่างต้องการภาพประกอบเพื่อให้เกิดความสมบูรณ์ เช่น ระบบข้อมูลทะเบียนประวัติของพนักงาน , logo เครื่องหมายการค้า , ข้อมูลทางด้านสถิติ , ข้อมูลการวิจัยตลาด , ข้อมูลทางด้านแผนที่และโครงสร้างของสิ่งต่างๆ เป็นต้น ดังนั้นเครื่องอ่านภาพ (Image scanner) จึงมีความสำคัญและมีบทบาทมากยิ่งขึ้น วิธีที่มีประสิทธิภาพสูงที่สุดในขณะนี้ คือการใช้ CCD (Charge couple device) เป็นตัวอ่านข้อมูล แต่อุปกรณ์ชนิดนี้มีราคาแพงและมีความซับซ้อนของระบบมาก ดังนั้นจึงเลือกใช้ ทรานซิสเตอร์ และโปรโตไทป์ในการอ่านภาพ ซึ่งจะมีราคาถูกกว่ามาก และมีคุณภาพพอใช้ได้

คณะผู้จัดทำ

บทที่ 1

ส่วนประกอบของ เครื่องสแกนภาพ

เครื่องสแกนภาพ (IMAGE SCANNER) เป็นอุปกรณ์ต่อพ่วงกับคอมพิวเตอร์ มีหน้าที่สำหรับอ่านข้อมูลภาพเข้ามาเก็บไว้ในเครื่องคอมพิวเตอร์ เพื่อจะนำไปใช้ประโยชน์ในด้านต่างๆ เช่น การประมวลผลข้อมูลรูปภาพ ประกอบด้วยส่วนต่างๆ ดังนี้

1. แท่นสแกน
2. หัวอ่านภาพ
3. วงจรอินเตอร์เฟสกับคอมพิวเตอร์
4. วงจรควบคุม
5. วงจรขับมอเตอร์
6. วงจรจ่ายไฟเลี้ยง

จากที่กล่าวมาทั้งหมดนี้เป็นส่วนประกอบทางฮาร์ดแวร์ในส่วนของซอฟต์แวร์ ใช้พูลดาวน์เมนู (PULL DOWN MENU) เพื่อความสะดวกในการใช้งาน ประกอบด้วย

1. ส่วนที่จัดการเกี่ยวกับไฟล์ (FILE)
2. ส่วนที่จัดการเกี่ยวกับการสแกนภาพ
3. ส่วนที่จัดการเกี่ยวกับหน้าจอ
4. ส่วนที่จัดการเกี่ยวกับค่าตัวแปรของการสแกน
5. ส่วนที่จัดการเกี่ยวกับการพิมพ์

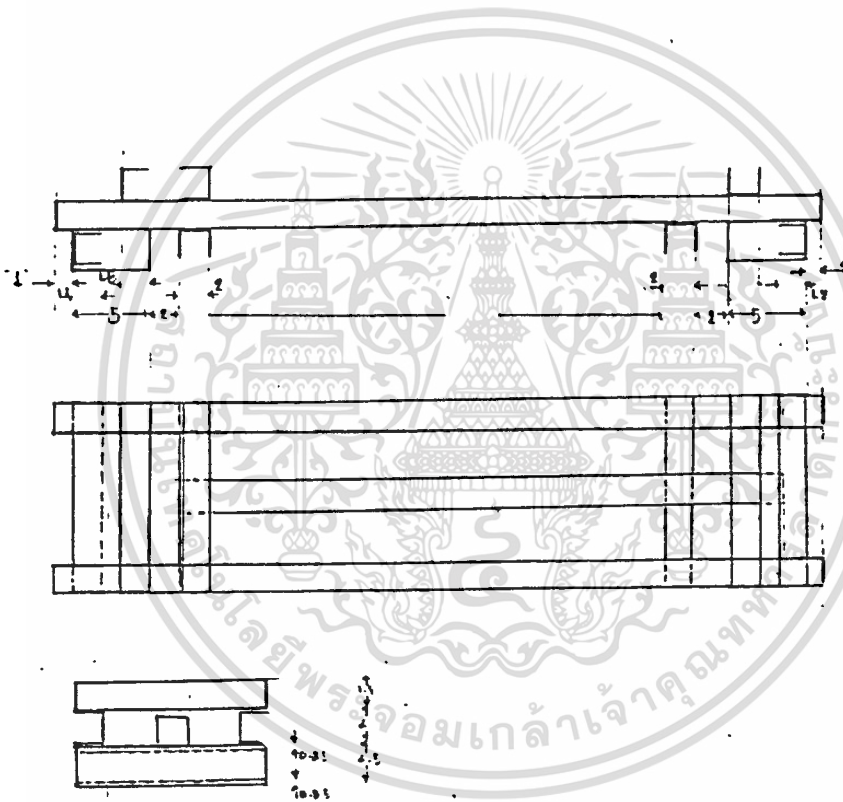
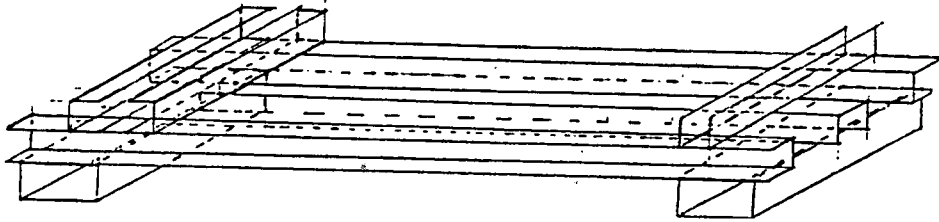
สำหรับโครงงานนี้ในเทอมแรกเราจะทำแท่นสแกนขึ้นมาก่อน และทำในส่วนฮาร์ดแวร์ (HARDWARE) ทั้งหมด และทดลองทำการควบคุมแท่นสแกนด้วยซิงเกิลบอร์ด (SINGLE BOARD) เนื่องจากควบคุมได้ง่าย สะดวก และเห็นการทำงานของส่วนประกอบต่างๆได้ชัดเจน ในเทอมที่สองจะทำในส่วนหัวอ่าน ทำการอ่านภาพและโปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- งานชิ้นแรก เราได้ทำการออกแบบแทนสแกน โดยมีขั้นตอนดังต่อไปนี้
- 1.ศึกษาถึงแนวทาง รูปแบบและปัญหาที่อาจจะเกิดขึ้น
 - 2.ออกแบบโมเดล (Model) ของแทน โดยลองใช้กระดาษแข็งมาทำ
 - 3.ทำการเลือกซื้ออุปกรณ์
 - 4.ทำการประกอบแทนสแกน
 - 5.ทำการทดสอบการทำงานของแทนและแก้ไขปัญหาต่างๆที่เกิดขึ้น

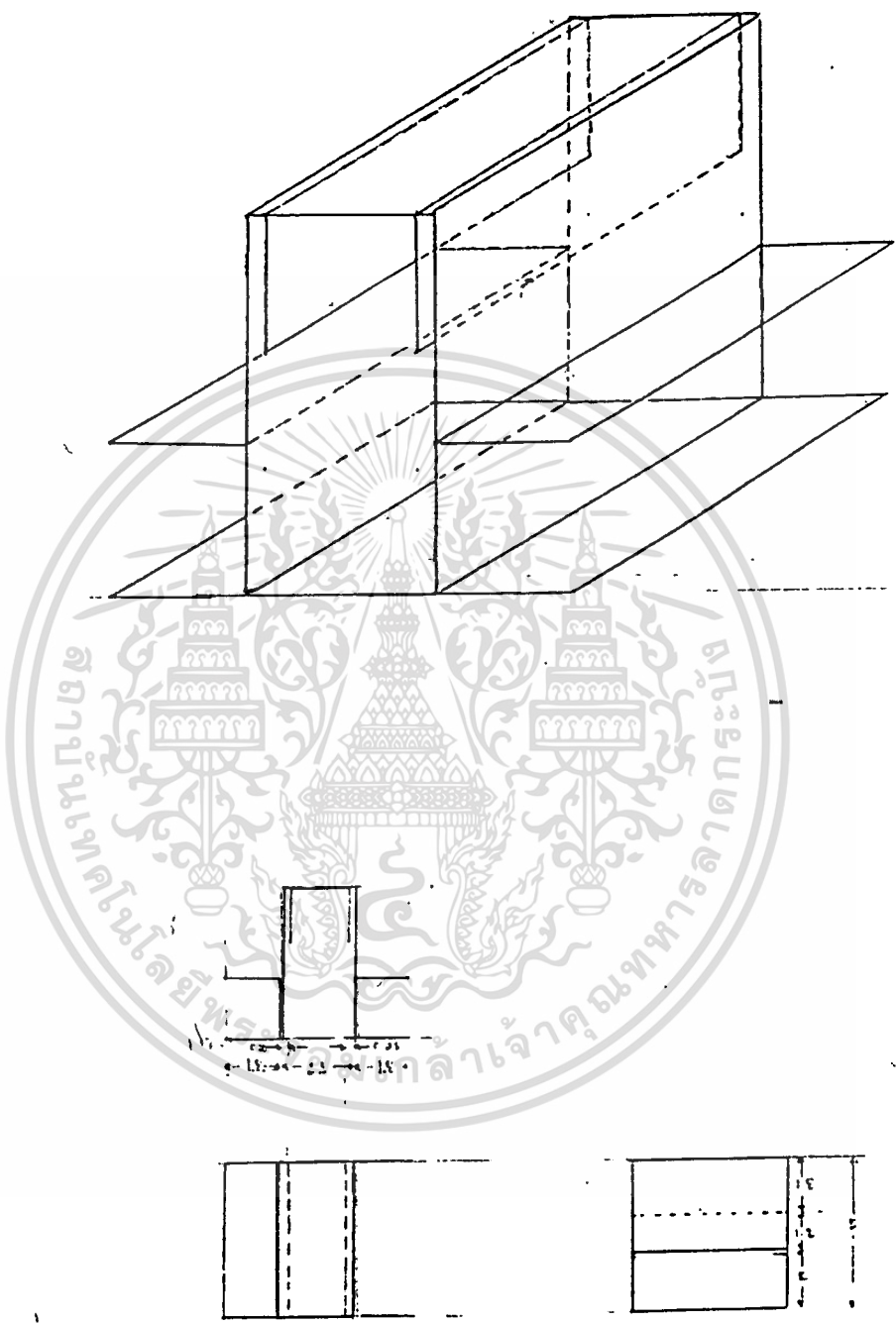
ต่อไปนี้ เป็นแบบที่ได้ทำการออกแบบขึ้นมา เพื่อสร้างแทนสแกนดังรูปต่อไปนี้





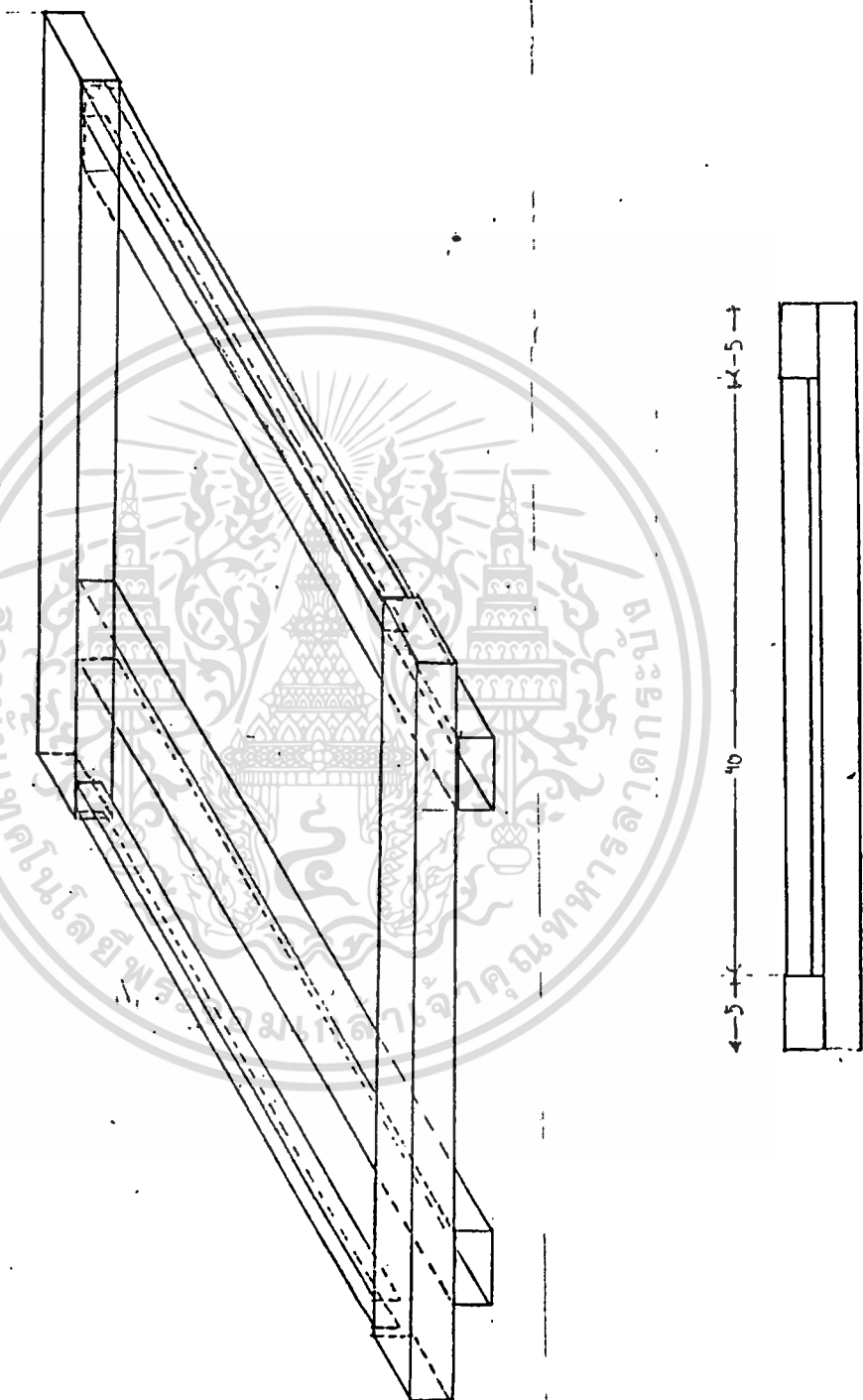
รูป 1.1 แบบของโครงที่เคลื่อนที่แกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



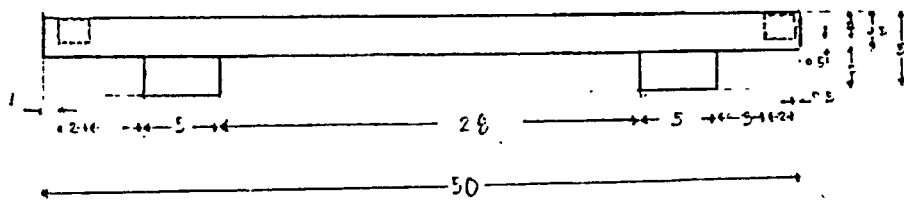
รูป 1.2 แบบของตัวเคลื่อนที่ในแกน x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

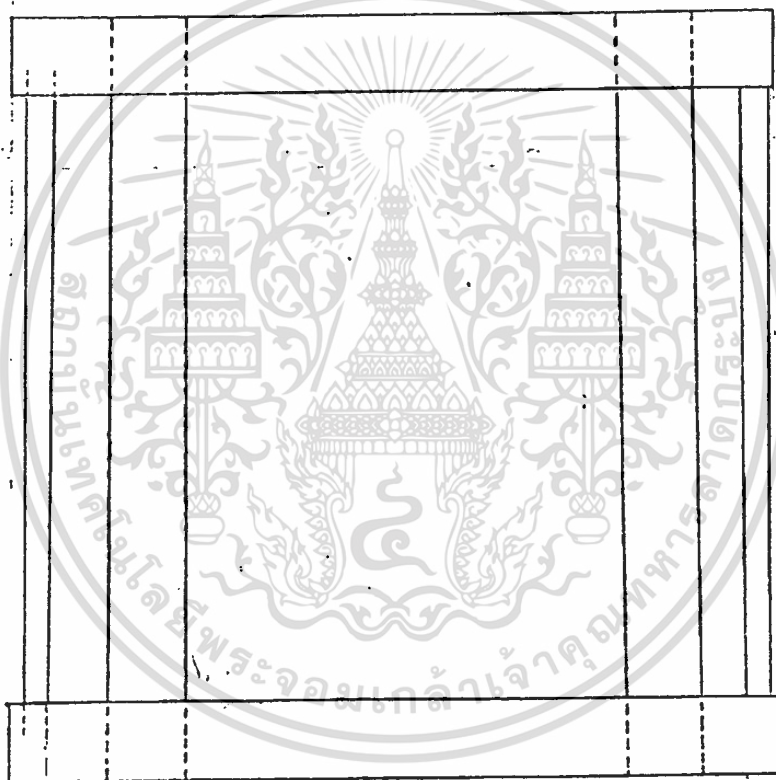


รูป 1.3 ภาพด้านข้างของแท่นสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

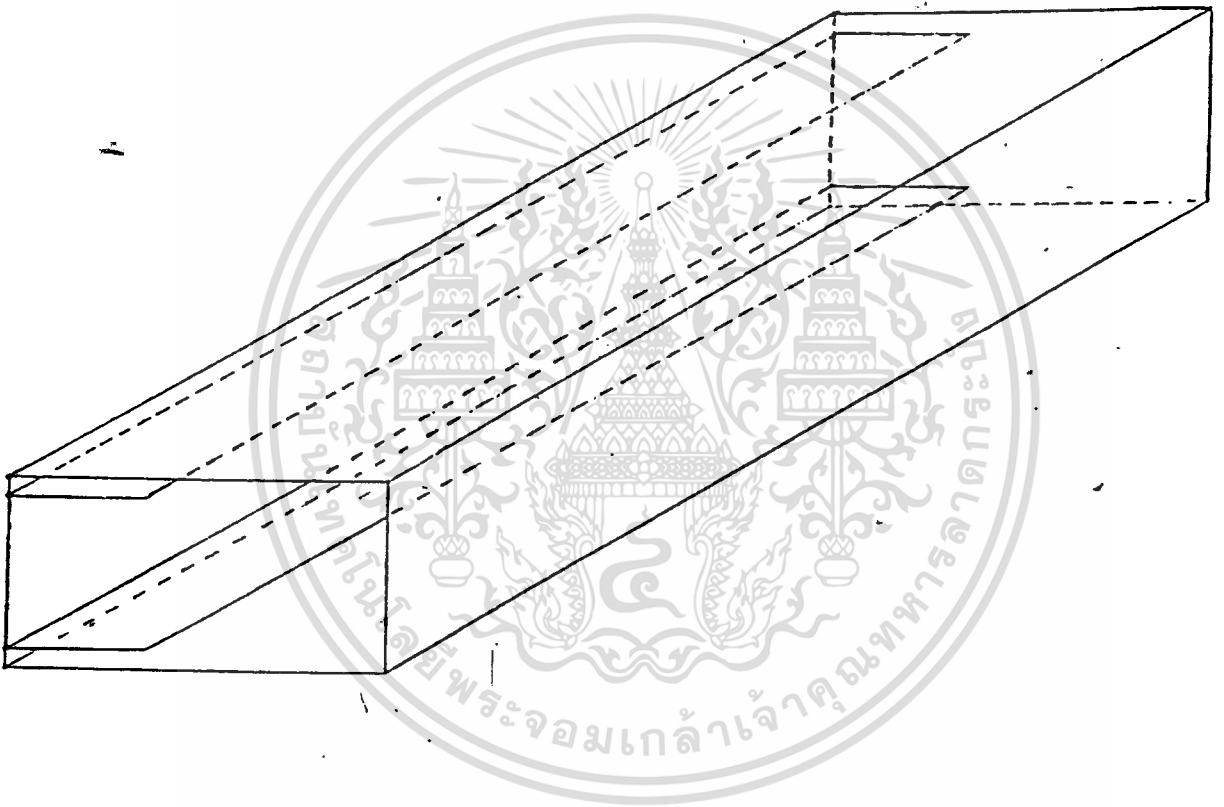


รูป 1.4 ภาพด้านข้างของแท่นสแกน



รูป 1.5 ภาพด้านบนของแท่นสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1.6 ภาพส่วนประกอบของโครงที่เคลื่อนขนานแกน x

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ทฤษฎีการทำงานของวงจร

2.1 บล็อกไดอะแกรม

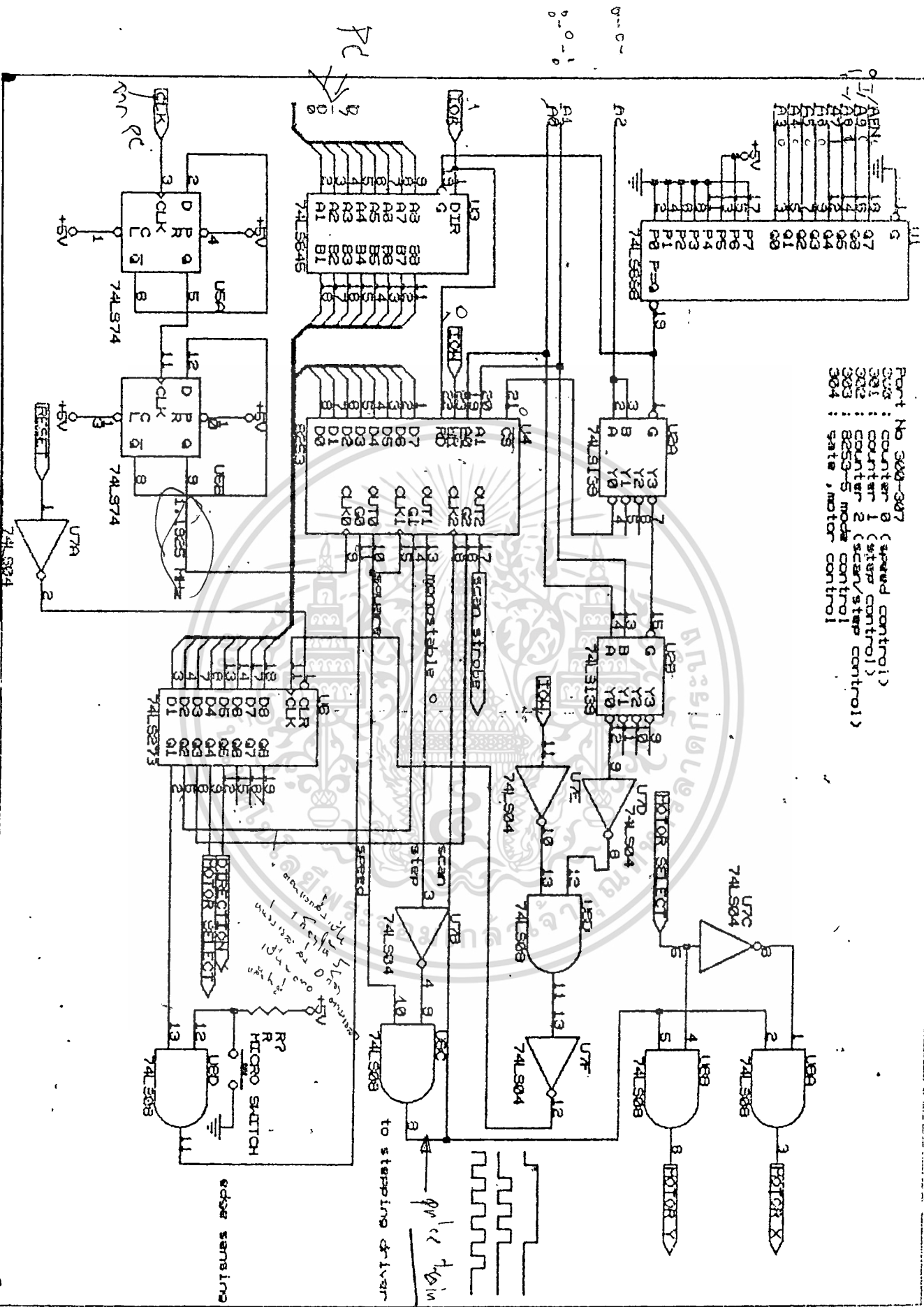
ประกอบด้วยส่วนที่อินเตอร์เฟสกับคอมพิวเตอร์ ส่วนที่ควบคุมการหมุนของมอเตอร์ และส่วนที่ทำหน้าที่อ่านข้อมูลมาเก็บไว้ในหน่วยความจำ และส่วนแสดงผล

บล็อกไดอะแกรมจากสล็อต (Slot) ของคอมพิวเตอร์ จะส่งสัญญาณมาที่วงจรมอเตอร์เฟส 3 ชุด คือ แอดเดรส ข้อมูล และ อินเตอร์เฟสประกอบด้วยวงจรถอดรหัส และแลทช์ ในส่วนควบคุมทั้งความเร็ว สเตปหรือความละเอียดของการสแกนในแกน x มีวงจรมับ ส่วนการควบคุมมอเตอร์จะสามารถควบคุมได้ทั้งเฟสและทิศทางการหมุน และมีสัญญาณเลือกมอเตอร์ (motor select) เนื่องจากการสแกนมอเตอร์จะสลับกับการหมุน คือมอเตอร์ในแกน y จะหมุน 1 สเตปเพื่อเลื่อนบรรทัด และมอเตอร์ในแกน x จึงหมุนเพื่อสแกนบรรทัดต่อไป สัญญาณที่ได้จากวงจรถามุมจะต้องผ่านวงจรมอเตอร์ซีเควนเซียล (Sequentcial Circuit) เพื่อจัดลำดับของการกระตุ้นให้มอเตอร์ทำงาน และต้องผ่านวงจรมอเตอร์ขับเคลื่อน (Driver Circuit) เพื่อขับกระแสให้สูงขึ้น เพื่อที่จะทำให้ มอเตอร์มีแรงบิดสูง พอที่จะขับส่วนต่างๆของแท่นสแกนให้เคลื่อนที่ไปตามที่เราต้องการ

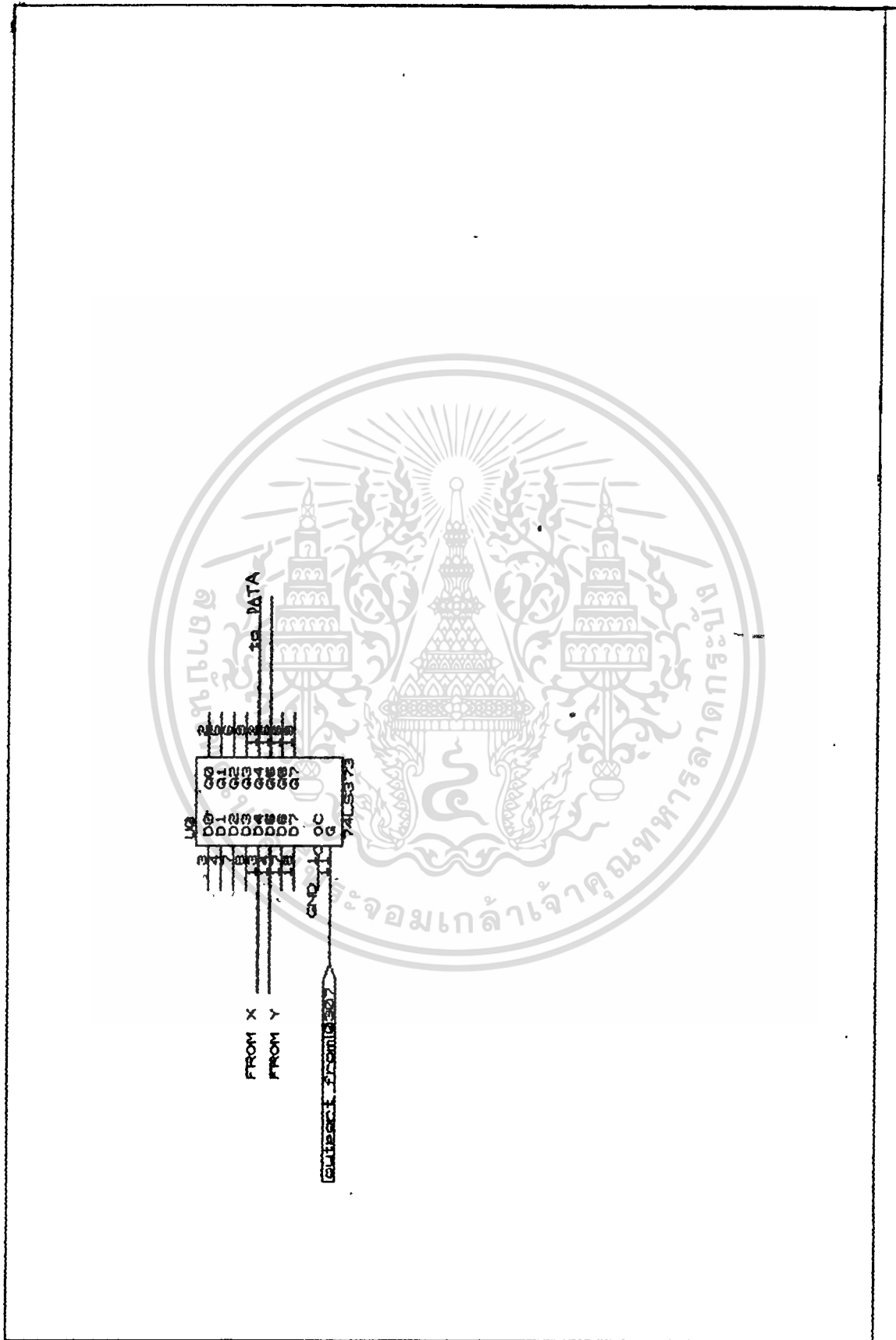
ในส่วนของหัวสแกนจะแปลงสัญญาณภาพ มาอยู่ในรูปของกระแสไฟฟ้าและผ่าน วงจรแปลงอนาล็อกเป็นดิจิตอล (A/D Circuit) เก็บสัญญาณไว้ใน RAM และส่งกลับไปยังจอภาพ

027967

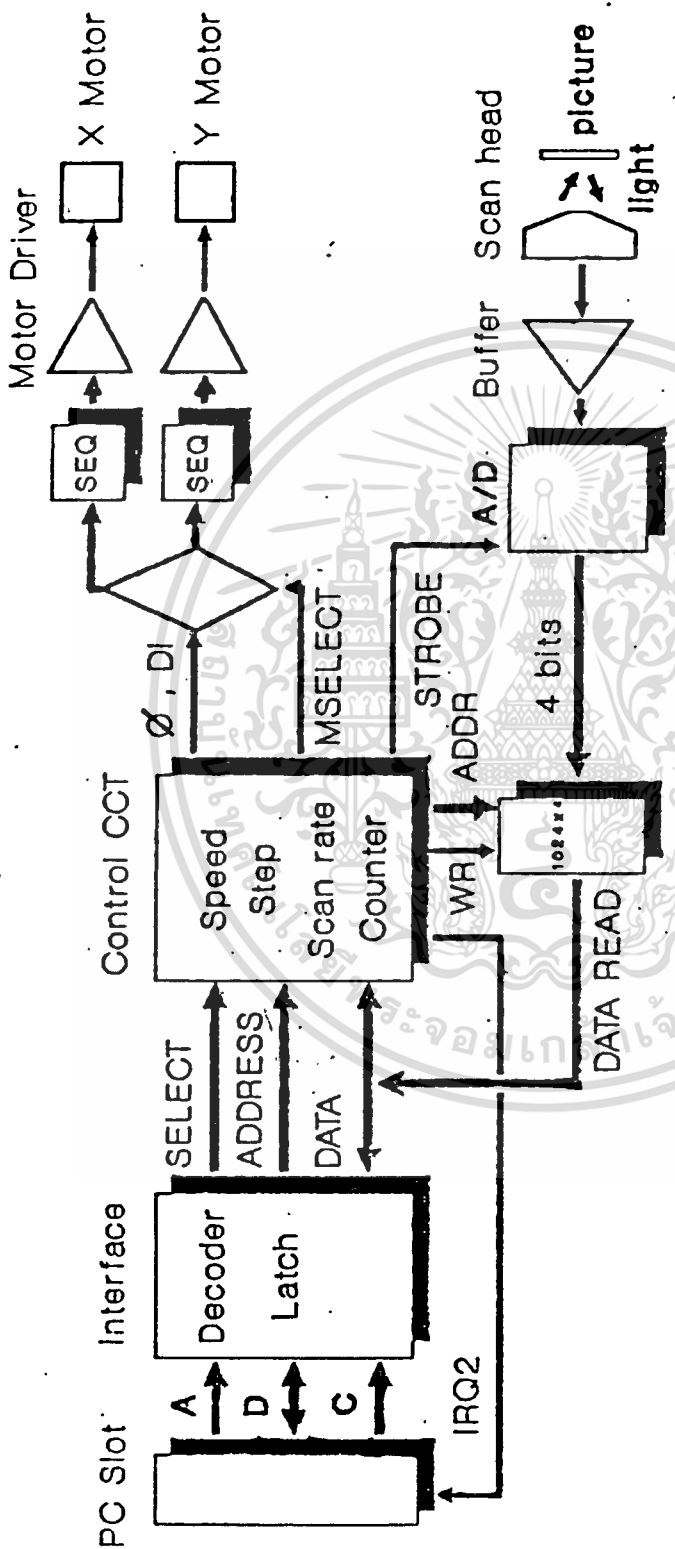
Port No 300-307
 300 : counter 0 (speed control)
 301 : counter 1 (step control)
 302 : counter 2 (scan/step control)
 303 : 8255-5 mode control
 304 : state, motor control



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรณีศึกษาเท่านั้น ไม่ควรนำมาใช้โดยไม่ได้รับอนุญาต
 รูป 2.1 แผนผังวงจรส่วนฮาร์ดแวร์ อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในห้องเรียนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูป 2.2 แผนผังส่วน Home
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Scanner block diagram

รูป 2.3 บล็อกไดอะแกรมของเครื่องสแกนภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านภาพใช้ทิวสแกนที่เป็นอุปกรณ์ ประเภทออปโตคัปเปิลเลอร์ (Optocoupler) ซึ่งประกอบด้วยโฟโตไดโอด และโฟโตทรานซิสเตอร์ ซึ่งโฟโตไดโอดจะส่งแสงอินฟราเรดมายังภาพและโฟโตทรานซิสเตอร์จะเป็นตัวรับแสงที่สะท้อนจากภาพ ปริมาณแสงสะท้อนขึ้นอยู่กับ

1. พื้นผิวของกระดาษ ถ้าขรุขระมาก ปริมาณแสงสะท้อนจะน้อยเมื่อเทียบกับผิวเรียบ
2. ระยะระหว่างภาพกับออปโตคัปเปิลเลอร์ ปริมาณแสงจะได้สูงสุดเมื่อภาพอยู่ห่างจากทิวสแกนเท่ากับระยะโฟกัส
3. สีของภาพ พื้นผิวสีขาวจะสะท้อนแสงได้มากกว่าสีดำ

2.2 วงจรอินเทอร์เฟส

เป็นวงจรเชื่อมต่อระหว่างไมโครคอมพิวเตอร์ สำหรับการติดต่อกับอุปกรณ์ภายนอกกับ ไมโครคอมพิวเตอร์ใช้แอดเดรส 10 เส้น โดย a9 จะต้องเป็น 1 เมื่อต้องการให้เป็นเอาต์พุตพอร์ท (อินพุตพอร์ทจะไม่ขึ้นกับ a9) ดังนั้นจึงต้องเลือกพอร์ทเบอร์ 300H-307H และเพื่อความสะดวกในการออกแบบวงจร และเขียนโปรแกรมวงจรนี้ประกอบด้วย u1, u2, u3 โดย u1 (74LS688) ทำหน้าที่เป็นตัวกำหนดพอร์ทใน 2 หลักแรก คือ 30 (0110 0000) และ u2, u3 จะเป็นตัวแปลงรหัส เพื่อเลือกเบอร์พอร์ทในหลักสุดท้ายคือ 0 ถึง 7 (0000 0111)

การทำงานของ 74LS688 คือเมื่อ A9-A3 (Q7-Q0)=P7-P0 จะให้ลอจิก "0" ซึ่งในวงจรนี้ ถ้าต้องการให้เอาต์พุตเป็นลอจิก "0" จะต้องมีลอจิกดังนี้

AEN	A9	A8	A7	A6	A5	A4	A3
0	1	1	0	0	0	0	0

A_2, A_1, A_0

ตาราง 2.1 แสดงการอ้างพอร์ต

ซึ่งก็คือการอ้างพอร์ตใน 2 หลักการแรกเป็น 30 ดังที่กล่าวมาแล้ว สำหรับ u2 (74LS139) เป็นวงจรถอดรหัส โดยขา A และ B จะต่อกับ A2ซึ่งหมายความว่าถ้า A2 เป็น 0 (A=0,B=0) Y0 จะมีลอจิกต่ำ เมื่อ A2 เป็น 1 Y3จะถูกเลือกแทนคือมีลอจิกต่ำ และไป Enable U2B 1ที่ถอดรหัส A1 และ A0 ออกเป็น 4พอร์ตอีกที โดยพอร์ตต่างๆ จะถูกใช้งานดังต่อไปนี้

PORT No	FUNCTION
300H	วงจรรับ 0 (Speed control)
301H	วงจรรับ 1 (Step control)
302H	วงจรรับ 2 (Scan Rate Control)
303H	8253-5 MODE CONTROL
304H	Motor Selected ,Direction

output 6 (0x300, 0x000)
" (0x300, 0x000)
" output 6 (0x301, 0x000)
" (0x301, 0x000)


ตาราง 2.2 แสดงหน้าที่ของพอร์ตต่างๆ

ส่วนพอร์ตเบอร์ 305-307 จะไม่ได้อ้างใช้

- ส่วนข้อมูลที่โปรแกรมจากไมโครคอมพิวเตอร์จะถูกส่งมายัง U3(74LS645) ซึ่งเป็น บัฟเฟอร์ซึ่งจะกำหนดทิศทางของข้อมูลได้ โดยถ้าขา 1 เป็น 0 ข้อมูลส่งจาก B ไปยัง A ถ้าเป็น 1 จะส่งจาก A ไปยัง B านกรณีนี้ขา IOR เป็น 1 ขา G เป็นขา (Gate) จะมีลจิจดเป็น 0

2.3 วงจรควบคุม

วงจรวางส่วนนี้ประกอบด้วย U4,U5,U6,U7 และ U8 โดยจะเริ่มกล่าวถึง U6 ซึ่งเป็นแลตช์ แบบ Positive-edge triggered ก่อน โดยมีแผนภูมิเวลา ดังรูป เมื่อมีการ Out มาที่ Port 304H ขา Y0 ของ U2B เป็น 0 และจะทำให้ขา 12 ของ U7F หรือขา CLK ของ U6 เป็น 1 ซึ่งทำให้เกิดสัญญาณขอบขาขึ้น ไป LATCH ข้อมูลใน U3 ไว้ ซึ่งแต่ละบิตจะนำไปควบคุมส่วนต่างๆดังนี้



สัญญาณ	หน้าที่
Q1	จะต่อไปยัง G0 ซึ่งจะควบคุมวงจรมุม 0 ของ 8253
Q2	จะต่อไปยัง G1 ซึ่งจะควบคุมวงจรมุม 1 ของ 8253
Q3	จะต่อไปยัง G2 ซึ่งจะควบคุมวงจรมุม 2 ของ 8253
Q4	เป็นสัญญาณเลือกมอเตอร์ ถ้าเป็น 0 จะเลือกมอเตอร์ X ถ้าเป็น 1 จะเลือกมอเตอร์ Y
Q5	Direction ใช้กำหนดทิศทางของมอเตอร์ ถ้าเป็น 0 จะหมุนตามเข็มนาฬิกา เป็น 1 จะหมุนทวนเข็มนาฬิกา
Q6	Unused
Q7	Unused
Q8	Unused

ตาราง 2.3 แสดงหน้าที่ของสัญญาณต่างๆ !

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

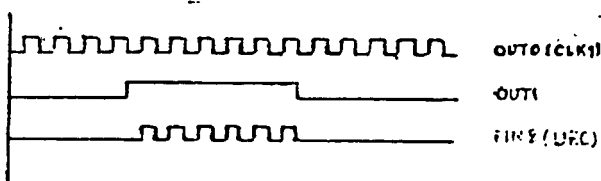
ส่วนสำคัญที่สุดของวงจรควบคุมนี้ จะเกี่ยวกับวงจรนับซึ่งใช้ 8253-5

PROGRAMMABLE INTERVAL TIMER ซึ่งมีวงจรนับ 16 บิต 3 ตัว สามารถโปรแกรมได้อิสระถึง 6 โหมด รายละเอียดของไอซีเบอร์นี้จะอยู่ในภาคผนวก ความถี่ที่ใช้งานสูงสุดประมาณ 2.6 MHz แต่สัญญาณ CLK สูงสุดของสล๊อตเท่ากับ 10 MHz จะใช้ D-FLIP FLOP 2 ตัวต่อเป็น TOGGLE 2 ภาค ซึ่งใช้หารความถี่จาก PC SLOT แล้วไปเข้าที่ขา CLK0 ของ 8253 เพื่อเป็นความถี่มูลฐานของวงจรนับต่อไป

2.5 MHz

วงจรนับเบอร์ 0 ของ 8253 จะถูกโปรแกรมโดย D7-D0 ให้เป็น SQUARE WAVE GENERATOR (MODE 3) ให้ความถี่ที่ขา OUT0 ออกมาและจะต่อไปยัง CLK1 เพื่อเป็น CLOCK ให้กับวงจรนับ 1 ซึ่งถูกโปรแกรมเป็น MONOSTABLE (MODE 1) โดยความยาวของ MODE 1 จะโปรแกรมได้เช่นกัน ดังนั้นจะเสมือนกับการนับพัลส์ที่ส่งมาจากวงจรนับ 0 โดยเริ่มนับที่ขอบขาขึ้นของ Q1 ซึ่งเอาท์พุท OUT1 จะไป AND กับ OUT0 จะได้ PULSE TRAIN ที่มีความถี่และจำนวนพัลส์ที่ต้องการ

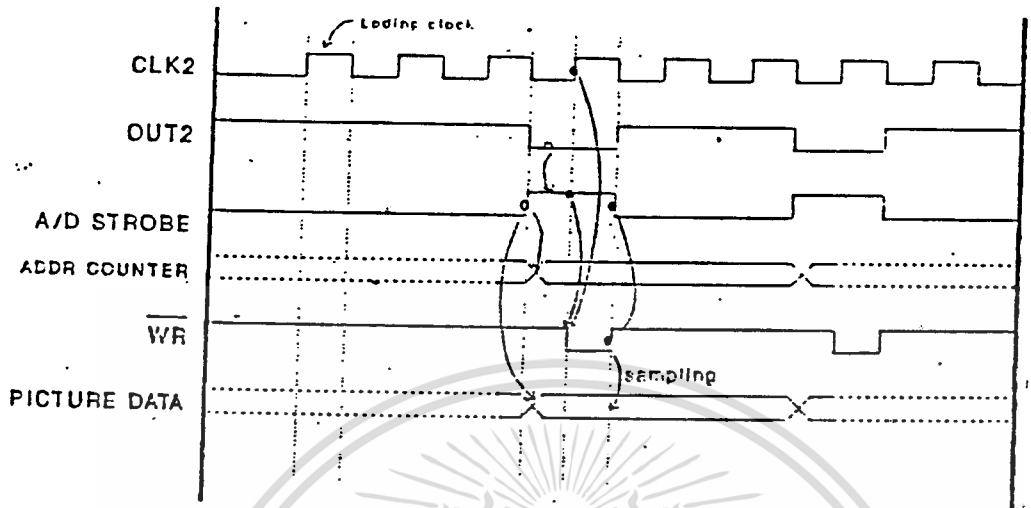
สัญญาณจาก B8C จะต่อไปยังวงจรเลือก โดยเมื่อสัญญาณเลือกมอเตอร์เป็น "0" จะผ่านสัญญาณจาก B8C นี้ไปยังวงจรซีเควนเซอร์ของมอเตอร์ X สัญญาณเลือกมอเตอร์ เป็น "1" จะผ่านสัญญาณนี้ไปยังวงจรซีเควนเซอร์ของแกน Y สำหรับสัญญาณจากวงจรนับ 0 จะเป็นความเร็วของมอเตอร์ และสัญญาณจากวงจรนับ 1 จะเป็นตัวควบคุมสเตรป ส่วนวงจรนับ 2 จะถูกโปรแกรมให้อยู่ใน MODE 2 (RATE GENERATOR) ซึ่งรับสัญญาณ CLOCK จากสัญญาณทรริกของวงจรซีเควนเซอร์ ซึ่งแสดงได้ดังรูป



รูป 2.4 แผนภูมิเวลาของสัญญาณควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

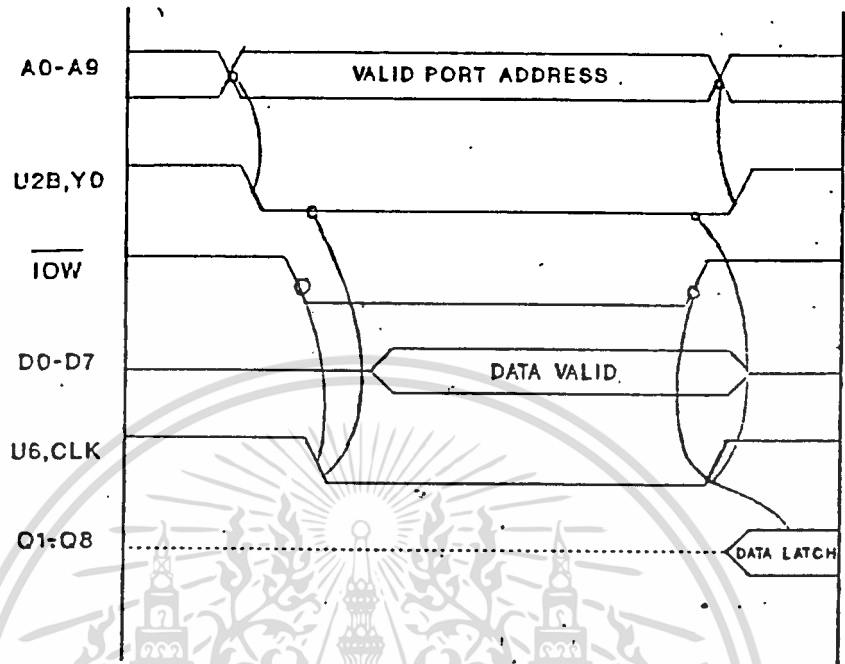
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Scan Strobe Timing Diagram

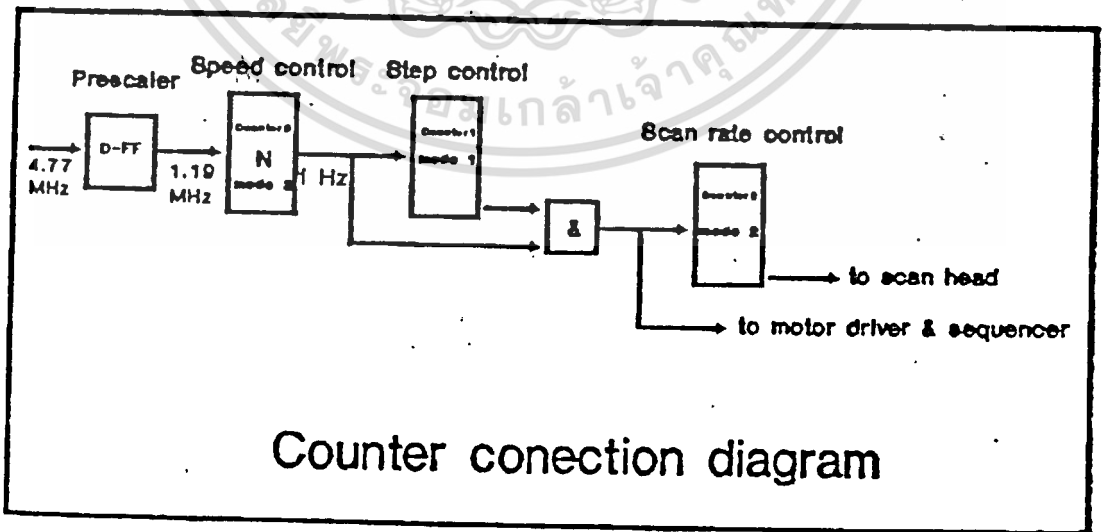
รูป 2.5 แสดงแผนภูมิเวลาของการสแกน

สำหรับความจุของหน่วยความจำจะขึ้นกับความละเอียดของภาพใน 1 บรรทัด เช่นที่ความละเอียด 1000 จุดต่อบรรทัด หรือ $8.5 \times 1000 = 850$ จุดต่อบรรทัด ดังนั้นจึงใช้หน่วยความจำขนาด $1K \times 4 = 4 \text{ Kbyte}$ ก็พอ ถ้าต้องการเพิ่มความละเอียดมากกว่านี้ก็ต้องหน่วยความจำเพิ่มให้ได้ความจุตามต้องการ



Data Latch Timing Diagram

2.6 แผนภูมิเวลาแห่งการแลกรหัสข้อมูล



Counter connection diagram

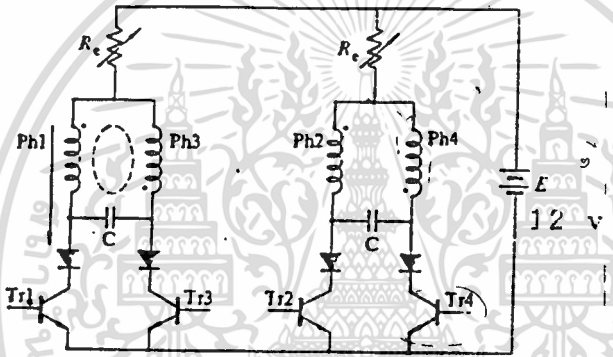
2.7 มัลติเพลกซ์ไดอะแกรมแห่งการต่อวงจรหม้อของ R253

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 วงจรขับสแต็ปปีงมอเตอร์ (STEPPING MOTOR)

สแต็ปปีงมอเตอร์ (STEPPING MOTOR) ที่ใช้นี้ต้องการไฟเลี้ยง 12โวลท์ ความต้านทานขดลวด 20 โอห์ม และมีมุมตอสแต็ปเท่ากับ 1.8 องศา โดยทำการกระตุ้นแบบสองเฟส เพื่อลดการสั่น (damp) และเพิ่มแรงบิดเพื่อความแน่นอนทางตำแหน่ง

DRIVE SYSTEM AND CIRCUITRY



รูป 2.8 แสดงวงจรขับมอเตอร์

จากวงจรดังรูป กำหนดค่าให้

$E = 12 \text{ v}$

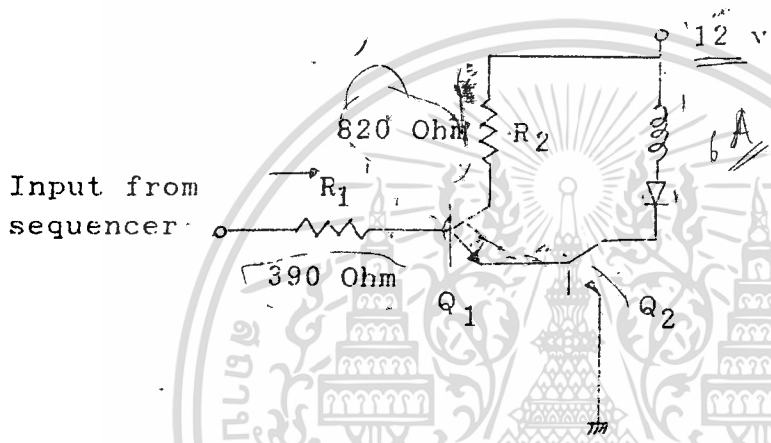
$R_e = 0.2 \text{ Ohm}$

$C = 2.22 \text{ } \mu\text{F}$

Diode ใช้เบอร์ 1N4001

Tr ใช้เบอร์ TIP 31

สำหรับตัวเก็บประจุ (capacitor) ที่ต่อกับ PH1 และ PH3 จะเป็นตัวป้องกันทรานซิสเตอร์ (transistor) ในช่วงที่มันไม่ทำงาน (off) โดยจะเป็นตัวดูดคลื่นกระแสจากขดลวด และยังทำหน้าที่เป็นอิเล็กทรอนิกส์แดมเปอร์ (electrical damper) ำให้กระแสไหลวนลูป (loop) และเกิดจูลฮีท (Jule heat) ในขดลวด ค่าของตัวเก็บประจุน้อยแรงบิดจะยิ่งมาก ในโครงการนี้จะใช้ค่าประมาณ 0.22 ไมโครฟารัด , 100 โวลท์



รูป 2.9 แสดงวงจรขับกระแส

กระแสที่ต้องการคือ 0.6 A
 Q1 2N2222 มีค่า $B_1=100$
 Q2 TIP 31 มีค่า $B_2=20$

I_{c2} คือค่ากระแสที่เราต้องการ

$$I_{b2} = 0.6/20 = 30 \text{ mA}$$

$$I_{b2} = b_1 * I_{b1} = I_{c1}$$

$$I_{b1} = 30/100 = 0.3 \text{ mA}$$

I_{b1} เป็นกระแสสูงสุดที่ได้จากวงจรควบคุมมอเตอร์

ถ้า $V_{on} = V_{max}$ ที่ได้จากวงจรควบคุม ซึ่งประมาณ 2.4 V

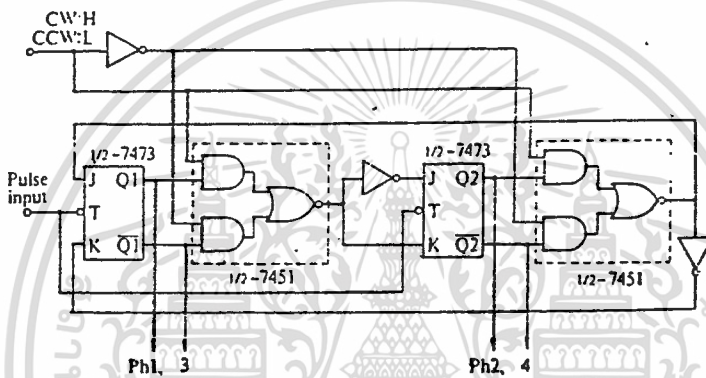
$$(2:4) = 0.3R_1 + 0.6 + 0.6$$

$R_1 = 400$ โอห์ม ดังนั้นจะใช้ R1 เท่ากับ 390 โอห์ม



2.5 วงจรควบคุมมอเตอร์ (Sequencer)

วงจรมีหน้าที่ในการจัดลำดับการกระตุ้นสี่เฟสของมอเตอร์ ให้หมุนไปในทิศทางที่ต้องการ ด้วยโหมดการกระตุ้นที่กำหนด ในโครงงานนี้ใช้การตรึงแบบสองเฟสแสดงวงจรดังรูป



A bidirectional two-phase-on logic sequencer for a four-phase motor.

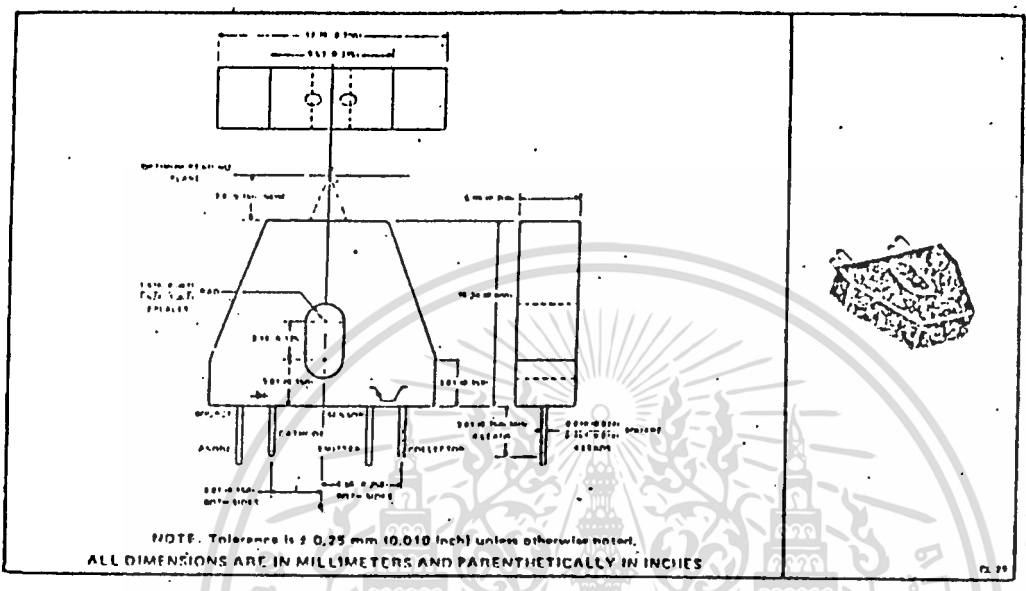
รูป 2.10 แสดงวงจร Sequencer ที่ใช้ในการกระตุ้นสองเฟส

สำหรับพัลส์อินพุท (Pulse input) ได้จาก U8 และสัญญาณการหมุนได้จากขา 12 ของไอซีเบอร์ 74273 เอาท์พุท (output) ของวงจรมีเป็นจะลอจิก (logic) 1 ครั้งละสองเฟส เรียงกันไปจาก PH1 และ PH2, PH2 และ PH3, PH3 และ PH4, PH4 และ PH1 ไปเรื่อยๆ หรือกลับทิศก็ขึ้นอยู่กับที่กำหนดทิศทาง ความเร็วในการเปลี่ยนเฟสขึ้นอยู่กับความถี่อินพุทที่ป้อน แต่ต้องไม่เกินความถี่ที่มอเตอร์จะตอบสนองได้ เอาท์พุททั้งสี่ที่ได้จะนำไปต่อกับวงจรขั้วมอเตอร์ต่อไป

บทที่ 3

ลักษณะการทำงานของหัวอ่าน

หัวอ่านมีลักษณะดังรูปข้างล่างนี้



รูป 3.1 แสดงลักษณะของออปโตดีเท็คเตอร์ที่ใช้เป็นหัวอ่าน

หัวอ่านของสแกนเนอร์ได้ตัดแปลงมาจาก source and detector infrared เบอร์ TIL 139 ซึ่งมีทั้งตัวรับและตัวส่งอยู่ในตัวเดียวกันซึ่งได้นำมาประกบตัวใช้ โดยอาศัยหลักการของการสะท้อนแสงซึ่งกระดาษแต่ละชนิดจะสะท้อนแสงต่างกันและสีต่างๆจะสะท้อนแสงและดูดกลืนแสงต่างกัน โดยหลักการดังกล่าวเมื่อนำไปต่อเข้ากับพอร์ตเกมและใช้ร่วมกับซอฟต์แวร์ก็สามารถใช้ เป็นหัวอ่านได้

3.1 การทำงานของพอร์ตเกม

พอร์ตเกมจะมี 8 อินพุต แบ่งออกเป็นอินพุตสำหรับสัญญาณล็อกอินพุตและสำหรับสัญญาณดิจิทัล 4 อินพุต แต่ที่เรานำมาใช้งานจะเป็นอินพุตสำหรับสัญญาณล็อก เมื่อเราใช้งานร่วมกับซอฟต์แวร์โดยใช้คำสั่ง เอาท์พอร์ทมายังแอดเดรสของพอร์ตเกมไอซี เบอร์ 558 ซึ่งเป็นวงจรมอนิเตอร์เบิ้ลภายในพอร์ตเกมจะเริ่มทำงาน พัลส์ที่ได้จากวงจรมอนิเตอร์เบิ้ลจะมีความกว้างของพัลส์ขึ้นอยู่กับความต้านทานของโพเทนชิโอมิเตอร์ที่ต่ออยู่ระหว่างไฟเลี้ยงกับอินพุตตามรูป จากนั้นจึงทำการอินพอร์ทเข้ามาเพื่อเช็คสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งความต้านทานของโพเทนชิโอมิเตอร์ จะมีผลต่อเวลาในการชาร์จตัวเก็บประจุตามค่าคงตัวของเวลา RC ซึ่งสามารถคำนวณเวลาที่อยู่ในสถานะ "1" โดยใช้แรงดันค่าหนึ่งได้จากสมการ

$$T = 24.2 + (0.011 * R) \quad \text{ไมโครวินาที}$$

ซึ่งสูตรนี้จะใช้ได้กับกรณีที่แรงดันคงที่เท่านั้น เช่น กรณีของจอยสติ๊ก แต่ในโครงการนี้เราใช้กระแสในการชาร์จประจุ จึงใช้สูตรคำนวณที่ต่างกัน

3.2 หลักการทำงานของหัวอ่าน

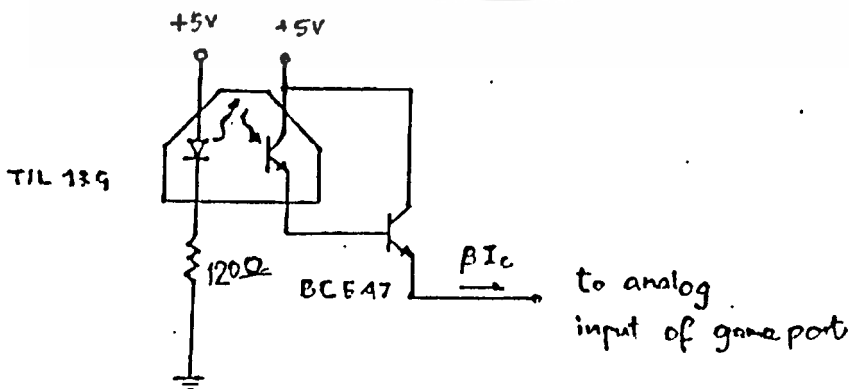
วงจรอิเล็กทรอนิกส์ที่ประกอบกันเป็นหัวอ่านประกอบด้วย

3.2.1 ภาคเปลี่ยนสัญญาณแสงเป็นสัญญาณไฟฟ้า

ภาคนี้ประกอบด้วยออปติคัลเปลวเบอร์เบอร์ TIL139 โดยไดโอดที่เป็นตัวส่งจะต่ออยู่กับตัวต้านทาน 120 โอห์ม ส่วนตัวรับซึ่งเป็นโฟโตทรานซิสเตอร์ จะถูกต่ออยู่กับภาคขยายกระแส

3.2.2 ภาคขยายกระแส

เนื่องจากสัญญาณไฟฟ้าที่ได้จากโฟโตทรานซิสเตอร์มีขนาดต่ำมาก ไม่สะดวกที่จะนำไปใช้ จึงต้องผ่านวงจรขยายกระแสก่อน ในโครงการนี้ใช้ทรานซิสเตอร์สองตัวต่อกันแบบคาร์ลิ่งตัน ซึ่งในโครงการนี้ใช้ทรานซิสเตอร์ เบอร์ BC 547 เมื่อขยายกระแสแล้วจะต่อเข้าเกมพอร์ทต่อไป



รูป 3.2 แสดงวงจรขยายกระแสจากหัวอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

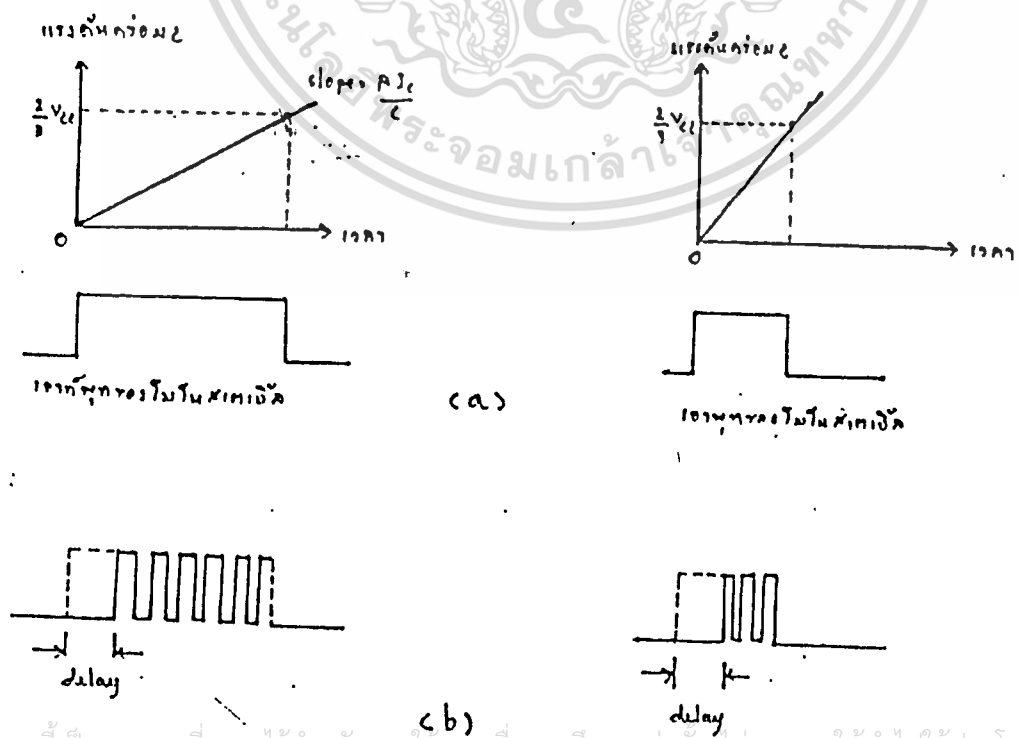
3.2.3 ภาคแยกระดับสัญญาณ

จากการศึกษาการทำงานของ เกมพอร์ท เมื่อใช้อินพุตสำหรับสัญญาณอนาล็อกพบว่า สามารถใช้วิธีอื่น ในการชาร์จตัวเก็บประจุที่ต่อกับไอซี 555 ซึ่งสามารถตรวจวัดและรับข้อมูล ผ่านอนาล็อกบิทของพอร์ทเกมได้เช่นเดียวกัน โดยยังต้องรักษาเงื่อนไขเดิมว่าอัตราการ ชาร์จตัวเก็บประจุ จะต้องแปรผันตามความเข้มแสงที่สะท้อนกลับมาจากจุดบนภาพ

การนำเอาพรีดีคิป์เปิ้ลมาใช้กับอินพุตสำหรับสัญญาณอนาล็อกจะมีความเหมาะสมมาก เพราะ อัตราการชาร์จตัวเก็บประจุ จะแปรผันตามกระแส I_C ตามสมการ

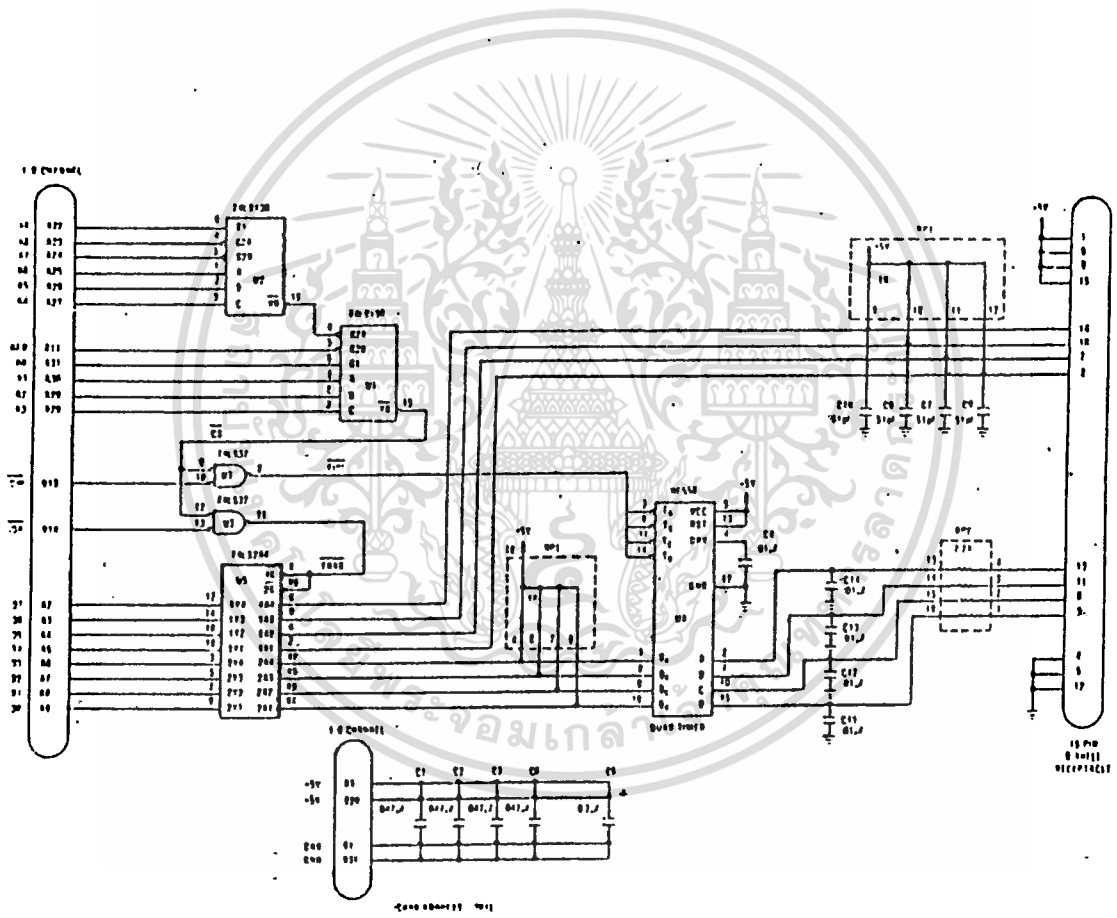
$$I_C = C \, dv/dt$$

และกระแส I_C จะแปรผันตามความเข้มของแสงสะท้อนตามจุดต่างๆบนภาพ กล่าวคือ เมื่อจุดนั้นมีสีดำ แสงจะถูกดูดกลืนมาก ทำให้ I_C มีค่าต่ำ อัตราการชาร์จประจุก็จะต่ำ ทำให้เวลาในการที่ระดับแรงดันของตัวเก็บประจุจะมีค่าถึง $2/3 V_{CC}$ นั้นจะกินเวลานานกว่าจุด ที่มีสีขาว ทำให้เวลาหน่วงของพัลส์ที่ได้จากเอาต์พุตของโมโนสเตเบิลของจุดสีดำมีค่ามาก ดังรูป



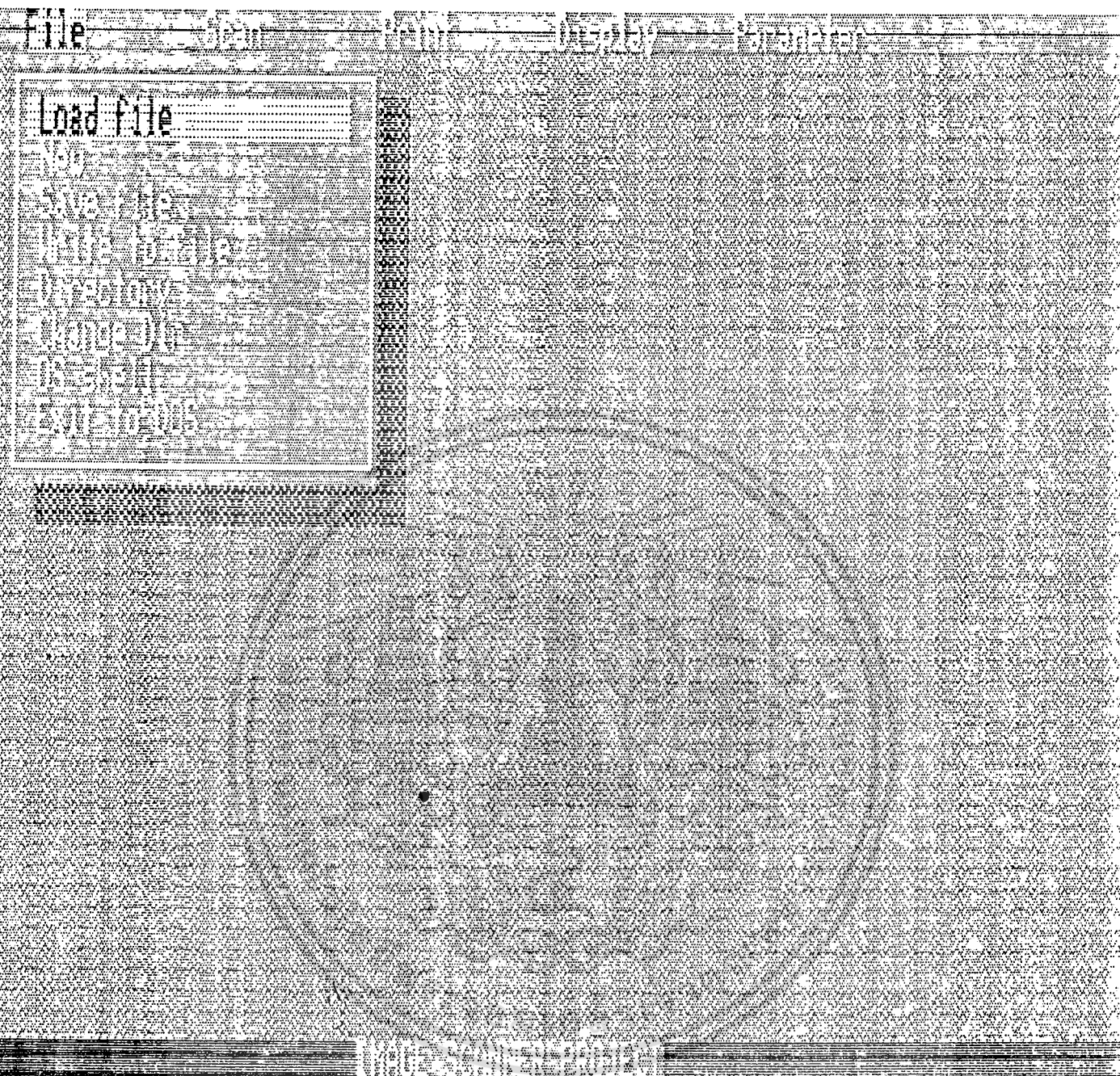
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูป 3.3 แสดงสัญญาณจากการอ่านเพื่อแปลงเป็นระดับ
 'ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่นโดยไม่ได้รับอนุญาตจากทางออกสารทุกครั้งที่มีการนำไปใช้'

เมื่อมีการส่งสัญญาณเอาต์พอร์ทและวงจรมอนสเตอร์เบิ้ลสร้างพัลส์ออกมาแล้วจึงส่งคำสั่งอินพุทไปยังแอดเดรสของพอร์ทเกม เพื่อตรวจสอบสถานะของพัลส์และเพิ่มค่าตัวแปรไปเรื่อยๆ จนกว่าจะมีการเปลี่ยนลอจิก (เมื่อระดับแรงดันของตัวเก็บประจุเท่ากับ $2/3V_{CC}$) ก็จะหยุดเพิ่มค่า เพราะฉะนั้นวิธีนี้ก็จะใช้แทนการนับเวลาได้ ถ้าจุดที่อ่านได้เป็นสีดํา ค่าตัวแปรจะมีความมากกว่าสีขาว และจะนำค่าที่อ่านได้ไปเปรียบเทียบกับทอนสีต่อไป



รูป 3.4 แสดงวงจรของพอร์ทเกม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.1 แสดงเมนูที่ผู้ใช้ควบคุมเครื่องสแกนภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

โปรแกรมควบคุมการทำงานของ Image Scanner

ในส่วนของซอฟต์แวร์ จะทำหน้าที่ควบคุมทั้งฮาร์ดแวร์และแท่นสแกนโดยมีการใช้คำสั่งเอาต์พอร์ทออกควบคุมพอร์ทที่ได้ออกแบบไว้แล้ว โดยผู้ใช้งานจะมีการควบคุมโดยผ่านทางเมนู ซึ่งได้ออกแบบมาให้ใช้งานได้ง่ายอยู่แล้ว ในโครงงานนี้ได้เปลี่ยนจากเมนูใน graphic ไปใช้ใน text mode เพราะว่าการเขียนโปรแกรมทำให้ง่ายกว่า ความเร็วสูงกว่า และใช้หน่วยความจำน้อยกว่า โปรแกรมจะเขียนด้วยภาษาซีทั้งหมด เมนูที่เขียนมีลักษณะเป็น Pull down Menu ซึ่งมีฟังก์ชันต่างๆดังนี้

- File
- Scan
- Print
- Display
- Parameter

4.1 Menu File

มีหน้าที่จัดการเกี่ยวกับไฟล์ทั้งหมด โดยมีฟังก์ชันในเมนูดังนี้

- Load file มีหน้าที่ในการ load file ที่มีอยู่แล้ว เพื่อทำการ print หรือ display
- New ทำหน้าที่คล้ายกับในเมนูของ turbo C คือใช้ในการ load file ใหม่
- Save file ทำหน้าที่เก็บไฟล์จากหน่วยความจำลงไปยังแผ่นดิสก์
- Write to file ทำหน้าที่คล้ายกับในเมนูของ turbo C คือเปลี่ยนชื่อไฟล์หรือ Save จากหน่วยความจำลงไปในดิสก์
- Directory ทำหน้าที่ในการดูไฟล์ในไดเรกทอรีปัจจุบัน
- Change dir ทำหน้าที่ในการเปลี่ยนไดเรกทอรี
- Os shell ทำหน้าที่ในการออกไปที่ DOS ชั่วคราวและสามารถกลับเข้ามาได้โดยพิมพ์ Exit
- Exit to Dos ใช้ในการออกจากโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 Menu Display

มีหน้าที่เกี่ยวกับการแสดงภาพที่อ่านได้บนจอมอนิเตอร์ มีฟังก์ชันในเมนูดังนี้

- Display picture ใช้เมื่อมีการ load ภาพจากแผ่นดิสก์ หรือเมื่อสแกนเสร็จแล้ว ต้องการจะดูภาพที่สแกนได้
- X-scale ใช้เมื่อต้องการย่อหรือขยายภาพในแนวแกน X
- Y-scale ใช้เมื่อต้องการย่อหรือขยายภาพในแนวแกน Y
- Scanner Demo ใช้เมื่อต้องการสแกนพร้อมกับแสดงผลไปด้วยซึ่งการสแกนด้วยค่า parameter จะต่างไปจากฟังก์ชัน scan
- Reload Demo ใช้ในการเรียกภาพที่ได้จาก scanner demo กลับมา

4.3 Menu scan

มีหน้าที่สั่งให้เริ่มการสแกน เมื่อกด enter แท่นสแกนจะเริ่มทำงานทันที และจะหยุดได้โดยกด Esc ในการหยุดสแกนจะหยุดที่ตำแหน่งซ้ายสุดเสมอ

4.4 Menu print

มีหน้าที่ในการสั่งพิมพ์ภาพออกทาง เครื่องพิมพ์ มีฟังก์ชันดังนี้

- Print มีหน้าที่ในการสั่งพิมพ์ภาพออกทาง เครื่องพิมพ์ ซึ่งใช้ได้เฉพาะเครื่องแบบ dot matrix เท่านั้น
- Left Margin ใช้ตั้งขอบซ้าย
- Right Margin ใช้ตั้งขอบขวา

4.5 Menu Parameter

มีหน้าที่ในการตั้งค่า parameter ต่างๆทั้งหมดของแท่นสแกน มีฟังก์ชันต่างๆดังนี้

- Speed ใช้ set ความเร็วของการสแกน ควรมีค่าระหว่าง 100-700 Hz
- Width ใช้ set ความกว้างของกระดาษที่จะสแกน โดยค่าสูงสุดไม่ควร

เกิน 10 นิ้ว

- Delay เป็นการ set delay time ของการอ่านภาพ ซึ่งค่านี้มี
ความสำคัญมาก ถ้าตั้งค่านี้ไม่ตึจะทาให้ภาพที่อ่านมาผิดพลาด
หรืออ่านไม่ได้เลย
- position ใช้ในการ set ระยะห่างระหว่างตำแหน่งที่เริ่มต้นสแกน กับ
ตำแหน่ง home โดยมีหน่วยเป็นนิ้ว
- Hor size ใช้ตั้งขนาดของภาพในแนวนอนว่าจะให้อ่านกี่จุด
- Ver size ใช้ตั้งขนาดของภาพในแนวตั้ง
- A type ใช้เป็น parameter ของหัวอ่านในการอ่านกระดาษ (ด้าน)
- B type ใช้เป็น parameter ของหัวอ่านในการอ่านกระดาษ (มีน)
- C type ใช้เป็น parameter ของหัวอ่านในการอ่านกระดาษโดยทั่วไป

เมื่อเราได้ทราบถึงหน้าที่ของฟังก์ชันแล้ว ในการใช้ เครื่องมีขั้นตอนดังนี้

1. เมื่อเข้าไปเมนู ครั้งแรกแถบสว่างจะติดอยู่ที่ load file ควรเลื่อนแถบไปยัง

Parameter ก่อนโดย Arrow key

2. ค่า Parameter ต่างๆใน Menu จะมีค่า Default ไว้ดังนี้

speed = 200 Hz

Width = 8 inch

delay = 8. msec

position = 1 inch

Hor size = 256

ver size = 256

และ ถ้าไม่มีการตั้ง Parameter A, B หรือ C จะตั้งไว้เป็น type C

ในการตั้งค่าต่างๆเมื่อเปลี่ยนค่าแล้ว ให้ใช้ key enter เท่านั้น

3. เมื่อตั้งค่าเสร็จแล้ว เลื่อนแถบไปที่ scan เพื่อเริ่มสแกน เมื่อเริ่มสแกนจะ
เคลื่อนที่ไปยัง home ก่อนจากนั้นจึงเลื่อนลงมา ณ ตำแหน่งที่ setไว้ให้เริ่มสแกน แล้วจึง

เริ่มสแกนด้วยความเร็วที่ set ไว้ สามารถหยุดได้โดยกดปุ่ม Esc เมื่อหยุดการทำงานก่อน ก็ยังสามารถดูภาพที่ scan ผ่านมาแล้วได้โดยการเลือกเมนู display แต่ถ้าปล่อยไว้สแกนจนเสร็จ จะมีการถามชื่อไฟล์ เมื่อตั้งเสร็จแล้วจะ save ลงแผ่นดิสก์

4. เมื่อ Save แล้ว ก็จะแสดงผล หลังจากนั้นจะสั่งพิมพ์ได้ทันที

5. เมื่อต้องการออกจากโปรแกรมชั่วคราวสามารถทำได้ โดยใช้ฟังก์ชัน OS shell แต่การรใช้ฟังก์ชันนี้ จะต้องมี command.com อยู่ใน Drive A และมีเนื้อที่ของหน่วยความจำเพียงพอ แต่ถ้าต้องการออกจากโปรแกรมก็ใช้ Exit to Dos



บทที่ 5

โปรแกรมควบคุมฮาร์ดแวร์และการแสดงผล

ในบทที่แล้วเราได้ทราบถึงวิธีใช้โปรแกรมควบคุมแทนสแกนแล้ว ในบทนี้จะกล่าวถึงการเขียนโปรแกรมควบคุมฮาร์ดแวร์และการแสดงผล ทั้งออกทางเครื่องพิมพ์และจอภาพซึ่งเป็นส่วนหลักของโปรแกรมทั้งหมด

5.1 โปรแกรมควบคุมการสแกน

ในส่วนที่ควบคุมการสแกนจะอยู่ในฟังก์ชัน scan เริ่มการทำงานก็จะทำการเอาท์พอร์ตไปยัง Counter0 โดยจะเอาท์พอร์ต low byte ออกไปก่อน แล้วจึงเอาท์พอร์ต high byte จากนั้นก็เรียกฟังก์ชัน Home ฟังก์ชันนี้มีหน้าที่ set ตำแหน่งเริ่มต้นเมื่อเริ่มการสแกนทุกครั้ง ความเร็วของการเคลื่อนที่ขณะเข้าสู่ตำแหน่ง home จะช้ากว่าปกติ เมื่อเคลื่อนไปกระทบ microswitch จะหยุดเคลื่อนที่ในแกน x และเคลื่อนในแกน y ต่อไปจนกระทบ microswitch อีกตัวหนึ่ง ซึ่งตำแหน่งนั้นจะเป็นตำแหน่ง home จากนั้นจึงสั่งให้เคลื่อนที่ไปแกน y จนถึงตำแหน่งต้องการจะเริ่มสแกน จากนั้นจึงเริ่มสแกนและเก็บภาพ โดยการเรียกฟังก์ชัน motor, steptomove, move เป็นคำสั่งที่เกี่ยวกับการโปรแกรม 8253 เพื่อทำงานตามต้องการ โดยฟังก์ชันต่างๆ มีหน้าที่ดังนี้

- motor ฟังก์ชันนี้จะมี Argument เป็น ver หรือ hor นั่นคือ ใช้ในการเลือกมอเตอร์นั่นเอง

ver จะเป็นการเลือกให้มอเตอร์แกน y หมุน

hor จะเป็นการเลือกให้มอเตอร์แกน x หมุน

- Steptomove ใช้กำหนดจำนวนสเตปที่จะให้มอเตอร์เคลื่อน มี argument เป็นตัวแปรซึ่งเก็บจำนวนเต็มฐาน 16 ซึ่งเป็นจำนวนสเตปที่จะให้มอเตอร์เคลื่อน

- move ใช้ในการเริ่มให้มอเตอร์เคลื่อนที่

เมื่อเรียกใช้ฟังก์ชันเหล่านี้แล้ว ก็ทำการวนลูบการอ่านภาพ โดยให้แกน y เป็นวนลูบ y-dimension ครั้ง และแกน x ให้ทำการอ่านภาพ x-dimension ครั้ง

ค่า x-dimension ,y-dimension สามารถsetได้ จากเมนู

ในส่วนของการอ่านภาพ จะเริ่มจากการเอาท์พอร์ทไปที่พอร์ท 0201 ซึ่งเป็น Game port เพื่อเป็นการclear port จากนั้นจึงเริ่มเช็คการอินพอร์ท เมื่อสัญญาณเป็น "1"อยู่ จะเพิ่มค่าตัวแปร img ไปเรื่อยๆ จนกระทั่งสัญญาณที่อินพอร์ทเข้ามาเป็น "0"หรือ อยู่ในขอบขาลง ก็จะนำค่าimg ไปเปิดตาราง แล้วเก็บไว้ในหน่วยความจำ ค่าที่เก็บไว้จะเป็นตัวเลข 0-7 ซึ่งเป็นค่าโทนสี ดังนั้นภาพหนึ่งจุดจะเก็บไว้ในหน่วยความจำขนาด 1 Byte และภายในลูบจะมีคำสั่งหน่วงเวลา เพราะถ้าไม่มีการ

หน่วงเวลา หัวอ่านจะอ่านภาพไปเรื่อยๆจนครบจำนวนครั้งที่ตั้งไว้แล้วจะหยุดทันที โดยหัวอ่านอาจยังเคลื่อนที่ไม่เสร็จ ทำให้หัวอ่านได้ในบางบริเวณเท่านั้น ดังนั้นเพื่อให้อ่านภาพดีขึ้น จึงต้องมีการหน่วงเวลาไว้ค่าหนึ่ง

เมื่อออกจากลูบของการอ่านภาพแล้ว จะสั่งให้มอเตอร์แกน y เคลื่อนที่ 1ครั้ง และหน่วงเวลาไว้ค่าหนึ่งเพื่อให้หยุดเคลื่อนที่จริงๆเสียก่อน แล้วจึงสั่งให้หัวสแกนเคลื่อนกลับ ในการเคลื่อนกลับจะไม่มีการอ่านภาพ เพราะภาพที่ได้จากการสแกนไปและกลับจะต่อกันไม่สนิท ทำให้ได้ภาพผิดเพี้ยนไป และจะเคลื่อนที่จนครบตามที่กำหนดไว้ หลังจากนั้นจะมีการถามชื่อไฟล์ แล้วนำชื่อนี้ไปเปิดไฟล์ และเขียนลงในดิสค์ในชื่อที่ตั้งไว้ จากนั้นก็จะปิดไฟล์ และจบการทำงานของโปรแกรม

5.2 โปรแกรมพิมพ์ออกทางเครื่องพิมพ์

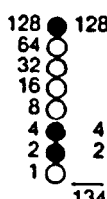
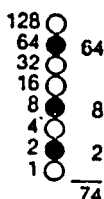
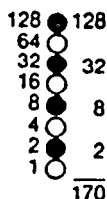
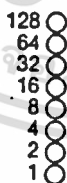
ในโครงการนี้ใช้เครื่องพิมพ์ Epson Dot Matrix เมื่อ operate ใน Graphic modeจะมี8 mode ดังนี้

Mode	Density	Description	speed
0	single	60 dpi	16 inch/sec
1	low speed double	120 dpi	8 inch/sec
2	high speed double	120 dpi	16 inch/sec
3	Quadruple	240 dpi	8 inch/sec
4	crtI	80dpi	8 inch/sec
5	one-to-one	72dpi	12 inch/sec
6	crtII	90dpi	8 inch/sec
7	Dual Density	144dpi	3 inch/sec

ตาราง 5.1 แสดงโหมดต่างๆในการพิมพ์ของเครื่องพิมพ์

โหมด 0-6 จะใช้ 8 เข็ม ซึ่งจะให้ภาพที่ดีและควบคุมได้สะดวก เพราะในการสั่งพิมพ์ 1 ครั้ง จะใช้ character ขนาด 8 บิตควบคุมเข็มพิมพ์ ซึ่งจะอธิบายได้ดังนี้

พิจารณา 1 คอลัมน์มี 8 เข็ม ดังรูป การให้ค่า character ที่มีรหัสเท่ากับรหัสของเข็มแต่ละเข็ม ทว่าให้เข็มนั้นถูกยิงออกไปกุดผ้าหมึกติดบนกระดาษ ดังนั้นถ้าต้องการให้เข็มบนสุดพิมพ์จะต้องส่ง 128 เป็นต้น



รูป 5.1 แสดงการวางตัวของเข็มของเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

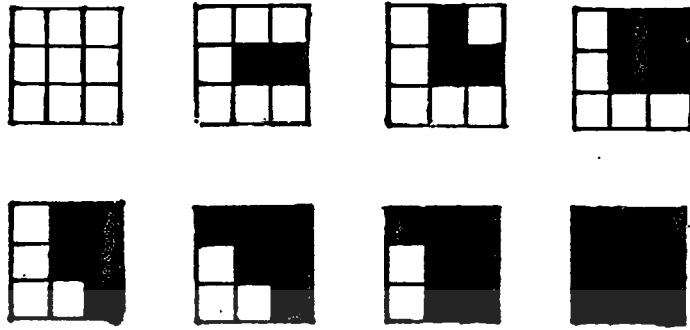
เมื่อทราบถึงหลักการโดยทั่วไปแล้ว จะกล่าวถึงโปรแกรมที่ใช้ในโครงการนี้อยู่ในฟังก์ชัน `prnimg` โปรแกรมนี้ใช้กับ Epson dot matrix ได้ทุกรุ่น รวมทั้ง NEC Pinwriter ด้วย

เริ่มแรกจะมีการเรียกฟังก์ชัน `initlut` ซึ่งมีหน้าที่ให้ค่าแก่ตารางที่ใช้เปรียบเทียบค่าที่อ่านได้จากหัวอ่าน การเปรียบเทียบค่าในการพิมพ์กับการแสดงผลใช้วิธีการต่างกัน โดยในการแสดงผลจะเปรียบเทียบดูว่าอยู่ในโหมด แล้วแสดงผลออกมาทีละ 1 แพทเทิร์น ส่วนการพิมพ์จะเปรียบเทียบกับค่าในตารางขนาด 4×4 โดยเปรียบเทียบทีละจุด ถ้ามากกว่าก็จะพิมพ์ออกมา 1 จุด ถ้าน้อยกว่าจะไม่พิมพ์

หลังจากเรียกฟังก์ชันนี้แล้วก็เรียกฟังก์ชัน `wait` เพื่อตรวจสอบความพร้อมของเครื่องพิมพ์ เมื่อพร้อมแล้วจึงเริ่มการพิมพ์ โดยการส่งรหัสควบคุมเพื่อ `set` เครื่องพิมพ์ด้วยรหัส `Esc A 8` เป็นการ `set linespace` เท่ากับ $8/60$ นิ้ว ตามด้วย `Esc *,0,0,1` เป็นการ `set` ให้พิมพ์ใน Graphic mode โดยให้พิมพ์ใน mode 0 คือ 8 เข็ม 60 dpi single density และให้พิมพ์ 256 คอลัมน์ โดย $n_1=0, n_2=1$ เมื่อ `set` เรียบร้อยแล้ว จึงเริ่มส่งพิมพ์ เมื่อพิมพ์เสร็จจะ `reset` เครื่องพิมพ์และ `clear` บรุตปัจจุบัน ก็จะสิ้นสุดการทำงานของโปรแกรมพิมพ์

5.3 โปรแกรมแสดงผลบนจอภาพ

จะกล่าวถึงการเก็บภาพและแสดงผลเป็นอันดับแรก เนื่องจากการแสดงผลต้องการให้ภาพมีความลึก โดยทำให้แบ่งโทนสีได้ 8 ระดับ แต่จอมอนิเตอร์ที่ใช้ในโครงการนี้เป็นจอโมโนโครม ใช้การ์ดเฮอรัลด์ จึงสามารถแสดงสีได้เพียง 2 สีเท่านั้น คือมีดกับสว่าง ดังนั้นเพื่อแบ่งระดับเป็น 8 ระดับจึงต้องใช้เทคนิคของ Image Processing มาช่วยเรียกว่า ดิทเทอร์ (Dither) กล่าวโดยง่ายคือการนำเอาจุดหลายๆจุดมาแสดงแทน 1 จุด เพื่อให้สามารถแบ่งระดับสีได้มากขึ้น ในโครงการนี้ใช้ 9 จุดแทน 1 จุด สามารถแบ่งความเข้มได้ 8 ระดับ โดยออกแบบแพทเทิร์นไว้ 8 แพทเทิร์นคือแพทเทิร์น 0-7 แพทเทิร์น 0 จะมีความเข้มน้อยที่สุด และจะเพิ่มขึ้นจนถึงแพทเทิร์น 7 จะเข้มที่สุด แพทเทิร์นที่ออกแบบไว้เป็นดังรูป



รูป 5.2 แสดงแพทเทิร์นที่ออกแบบ

ซึ่งหลักในการออกแบบแพทเทิร์นคือต้องให้มีการเรียงกันดูแล้วกลมกลืน และ
 แยกระดับความเข้มได้ชัดเจน ซึ่งการออกแบบแพทเทิร์นหนึ่งๆอาจให้ภาพที่ดีหรือไม่ดีก็ได้
 ดังนั้นแพทเทิร์นที่ดีที่สุดจะได้รับการทดลองหลายๆแบบ เมื่อได้แพทเทิร์นแล้วจะนำมา
 แปลงเป็นเลขฐานสิบหก เพื่อแทนแพทเทิร์นต่างๆ โดยในโครงการนี้ได้จัดดังนี้

1	2	4
8	16	32
64	128	256

รูป 5.3 แสดงการแทนค่าแพทเทิร์นด้วยตัวเลข

ถ้าใน คิทเทอร์ | ไม่มีจุดสว่างเลยจะแทนแพทเทิร์นด้วย 0 ถ้ามีจุดสว่างที่ 1 กับ 16
 ก็แทนด้วย 17 เป็นต้น จากนั้นก็ทำการทดลองให้หัวอ่านลองอ่านค่าสีต่างๆ เมื่อได้ข้อมูล
 มาแล้วก็จัดเป็น 8 กลุ่ม แบ่งออกเป็น 8 ระดับ จากนั้นก็ทำตารางขึ้นมาเพื่อใช้เปรียบเทียบ
 ค่าที่อ่านได้ในโปรแกรม เมื่ออ่านภาพ 1 จุด จะนำมาเปรียบเทียบในตารางแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เก็บเป็น character ขนาด 8 บิต เมื่อแสดงผลก็จะเอาค่า character ที่เก็บไว้ไปเรียกแพทเทิร์นแล้วแสดงบนหน้าจอ 1 แพทเทิร์นคือ 9 จุด แต่เนื่องจากเซอร์คูลิสการ์ตัน Graphic mode จะแสดงผลได้ 720*348 จุด ภาพที่ได้จากการสแกนอาจแสดงได้ไม่หมด จึงต้องมีการย่อภาพลง โดยการข้ามการแสดงผลภาพบางจุดไป ทำให้ภาพมีขนาดเล็กลงพอที่จะแสดงผลทั้งหมดบนจอได้ โดยที่ภาพไม่ผิดส่วนมากนัก

กล่าวโดยสรุป การอ่านภาพ 1 จุด จากแผ่นกระดาษ จะเก็บค่าทอนสีไว้ในหน่วยความจำและลงไฟล์ขนาด 1 ไบท์ ในการแสดงผลบนจอภาพจะแสดง 9 จุดแทนภาพจริง 1 จุด เพื่อให้ภาพมีความลึก และชัดเจนยิ่งขึ้น แต่เมื่อพิมพ์ออกมาจะพิมพ์ 1 จุด ต่อภาพจริง 1 จุด ถ้าพิมพ์ 9 จุดแทน 1 จุดจะพิมพ์ได้ไม่หมดเพราะภาพมีขนาดใหญ่เกินไป



5.4 โปรแกรม Demo

โปรแกรม DEMO เป็นโปรแกรมที่เขียนขึ้นอย่างคร่าวๆ เพื่อใช้ในการทดลองหาค่าต่างๆ และเพื่อแสดงให้เห็นหลักการทางานและฟังก์ชันพื้นฐานของเครื่องสแกนภาพ ซึ่งมีฟังก์ชัน HOME ทาหน้าที่ในการเลื่อนหัวอ่านมายังจุดเริ่มต้นของการสแกนภาพ ฟังก์ชันในการควบคุมการเคลื่อนที่ของหัวอ่านให้เคลื่อนที่ตามทิศทางที่ต้องการ และทาการเก็บข้อมูลภาพและแสดงภาพออกจอภาพ โดยสั่งให้หัวอ่านเคลื่อนที่ทีละบรรทัดและให้ทาการเก็บข้อมูลและแสดงภาพออกจอไปพร้อมๆกัน ในการอ่านข้อมูลดังกล่าวความเร็วในการอ่านจะเร็วกว่าการเลื่อนของหัวอ่าน จึงต้องใช้เวลาท่วงที่เหมาะสม เพื่อให้ได้ภาพที่มีสัดส่วนถูกต้อง เวลาท่วงที่เหมาะสมหาได้จากทดลอง เมื่ออ่านจบหนึ่งบรรทัดก็จะให้หัวอ่านเลื่อนไปที่ต้นบรรทัดถัดไป เพื่อความเที่ยงตรงทางคาแหน่งมากขึ้น ฟังก์ชัน SAVE ใช้ในการเก็บข้อมูลบนจอภาพลงบนแผ่นดิสค์ (disk) โดยเก็บข้อมูลจากหน่วยความจำจอภาพ (video ram) ลงบนแผ่นดิสค์โดยตรง ฟังก์ชัน LOAD ใช้แสดงภาพจากแผ่นดิสค์ลงบนจอภาพโดยตรง ฟังก์ชัน PRINT ใช้ในการพิมพ์ภาพจากจอภาพออก printer ซึ่งโปรแกรมนี้อยู่ในฟังก์ชัน Demoscanner และ Reloadscanner แล้ว

แสดงโปรแกรม demo

```

#include <dos.h>
#include <graphics.h>
#include <conio.h>
#include <stdio.h>

#define HOR    0
#define VER    1
#define CW     0
#define CCW    1
#define STOP   0
#define START  1
#define CNT0   0x300
#define CNT1   0x301
#define CNT2   0x302
#define CNT    0x303
#define M_CTRL 0x304
#define MAX_STEP 1400
#define CLK    4772726.6667

unsigned bit;
void motor(int);
void direction(int);
void steptomove(unsigned);

```

```

void move(unsigned);
void main(void)
{
    char want ;
    int drive = DETECT, mode,x,y;
    initgraph (&drive, &mode,"");

```

```

    home();

```

```

    outportb(CNT,0x36);

```

```

    outportb(CNT0,0x74);

```

```

    outportb(CNT0,0x36);

```

```

    for(y=0;y<240;y++)
    {

```

```

        motor(HOR);

```

```

        direction(CW);

```

```

        steptomove(0x0200);

```

```

        move(START);delay(100);

```

```

        picture(y); delay(2500);

```

```

        motor(VER);

```

```

        direction(CW);

```

```

        steptomove(0x0001);

```

```

        move(START);      delay(20);

```

```

        motor(HOR);

```

```

        direction(CCW);

```

```

        steptomove(0x0200);ล

```

3674

00110110 (CONTROL WORD)

CNT 0, - READ/LOAD

- MODE 3, - 2

ได้รหัสออกมา CNT0 → 3674H

คำสั่ง outportb 129 ครั้ง

```

    move(START);    delay(3500);
    }
do
{ if((want = getch()) == 's')
    save();
  else if((want =getch()) == 'p')
    print(0,0);}
while((want = getch()) != 'n');
closegraph();
}
/*-----*/
void motor(int m)
{ if(m == HOR){ bit &= 0xF7; outportb(M_CTRL,bit); }
  else {bit |= 0x08;outportb(M_CTRL,bit);}
}
/*-----*/

void direction(int d)
{ if(d == CW){ bit &= 0xEF; outportb(M_CTRL,bit);}
  else { bit |= 0x10; outportb(M_CTRL,bit);}
}
/*-----*/
void steptomove(unsigned s)
{int t,a;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((s>0) && (s<MAX_STEP))
{
  outportb(CNT,0x72);t=(s & 0x00FF);outportb(CNT1,t);
  a=s>>8;outportb(CNT1,a);}
}
/*-----*/

```

```

void move(unsigned s)

```

```

{
  if(s == START)

```

```

  {
    bit |= 0x0001;outportb(M_CTRL,bit & 0xFD);

```

```

    outportb(M_CTRL,bit | 0x02);}

```

```

  else {
    bit &= 0xFE;outportb(M_CTRL,bit);}

```

```

  }

```

```

/*-----*/

```

```

#include <stdio.h>

```

```

#include <dos.h>

```

```

#include <conio.h>

```

```

#include <bios.h>

```

```

void home(void);

```

```

void home(void)

```

```

{

```

```

  int n;

```

```

  outportb(0x303,0x36);

```

```

  outportb(0x300,0x00);

```

กำหนด MODE มาที่งานของ CNT0 ให้เป็น MODE 3 และให้
 bit 0 เป็น 1

ให้ bit 0 เป็น 1 0400 H

0600

```

outputb(0x300,0x04);
outputb(0x303,0x72);
outputb(0x301,0x01);
outputb(0x301,0x00);
do {
  n = inportb(0x307);
  n &= 0x30;
  outputb(0x304,0x00);
  outputb(0x304,0x13);
} while(n != 0x20); beep();
do {
  n = inportb(0x307);
  n &= 0x30;
  outputb(0x304,0x01);
  outputb(0x304,0x1B);
} while(n != 0x00); beep();
}

/*-----*/
beep()
{ sound(1500);delay(40);sound(100);delay(20);nosound();}
/*-----*/

```

micro MODE ใช้ CNF1 เป็น MODE 1, 1111
ใช้ micro MODE

ใช้โหมด motor 0001

do { ... } ... 20H ถึง 00H ...

um n = 20H ...

n = inportb(0x307); ...

n &= 0x30; ... AND

outputb(0x304,0x00); ...

outputb(0x304,0x13); delay(50);

} while(n != 0x20); beep();

do { ... 00100000

n = inportb(0x307); ... 00100000

n &= 0x30;

outputb(0x304,0x01);

outputb(0x304,0x1B); delay(50);

} while(n != 0x00); beep();

}

/*-----*/

beep()

{ sound(1500);delay(40);sound(100);delay(20);nosound();}

/*-----*/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <stdio.h>
#include <graphics.h>
#include <dos.h>
#include <bios.h>
void picture(int);
void picture(yy)
int yy;
{ int a,x,y,t;
  y=yy;
  for(x=0;x<1024;x +=2) {
    t=0;
    outportb(0x0201,00);
do {
  a=inportb(0x0201);
  a &=0x01;
  t++;
}while(a==1);
delay(5);
if(t<65)
  ptone1(x,y);
else if(t>=65 && t<=66)
  ptone2(x,y);
else if(t>66)

```

```

    ptone3(x,y);
}
}

ptone1(int x,int y)
{
    putpixel(x,y,1);
    putpixel(x+1,y,1);
}

ptone2(int x,int y)
{
    putpixel(x,y,1);
    putpixel(x+1,y,0);
}

ptone3(int x,int y)
{
    putpixel(x,y,0);
    putpixel(x+1,y,0);
}

```

```

#include <stdio.h>
#include <graphics.h>
#include <io.h>
#include <alloc.h>
#include <stdlib.h>

main()
{
    int drive = DETECT,mode;
    char filename[20];
    unsigned xsize,ysize,size;
    void *temp;
    FILE *f1;
    printf("Enter filename :");
    gets(filename);
    f1= fopen(filename,"r");
    initgraph(&drive,&mode,"");
    xsize = fgetc(f1)|(fgetc(f1)<<8);
    ysize = fgetc(f1)|(fgetc(f1)<<8);
    size = imagesize(0,0,xsize,ysize);
    temp =(void *)malloc(size);
    rewind(f1);fread((void *)temp,size,1,f1);
    fclose(f1);
    putimage(0,0,temp,COPY_PUT);
    getch(); closegraph();}

```

```

    fprintf(stdprn, "\x1B*%c%c%c", Mode, Lsb, Msb);
    for(i=0; i<=MaxX; i++)
    {
        m=0;
        for(k=0; k<8; k++)
        {
            m<<=1;
            if(getpixel(i, j*8+k)) m++;
        }
        fprintf(stdprn, "%c", m);
    }
    fprintf(stdprn, "\x0D\x0A");
}
}
case LANDSCAPE:
{
    Lsb=MaxY&0x00FF;
    Msb=MaxY>>8;
    for(j=0; j<MaxX; j+=8)
    {
        fprintf(stdprn, "\x1B*%c%c%c", 1, Lsb, Msb);
        for(i=MaxY; i>=0; i--)
        {
            m=0;

```

```

#include <stdio.h>

#include <stdlib.h>

#include <graphics.h>

#define PORTRAIT 0

#define LANDSCAPE 1

void print(int Mode,char Direction)
{
    char m;

    int i,j,k,Msb,LSb;

    int MaxX,MaxY;

    MaxX =getmaxx();
    MaxY =getmaxy();

    setviewport(0,0,MaxX,MaxY,0);
    fprintf(stdprn,"%c%c%c",27,51,25);
    switch(Direction)
    {
        case PORTRAIT:
        {
            Lsb=MaxX&0x00FF;

            Msb=MaxX>>8;

            for(j=0;j<=MaxY/8;j++)
            {

```

```

    for(k=0;k<8;k++)
    {
        m<<=1;
        if(getpixel(j+k,i)) m++;
    }
    fprintf(stdprn,"%c",m);
    }
    fprintf(stdprn,"\x0D\x0A");
}
}
fprintf(stdprn,"\f");
}
}

#include <stdio.h>
#include <graphics.h>
#include <alloc.h>
#include <conio.h>
#include <stdlib.h>

void *S(int,int,int,int,unsigned*);
void F(void*,unsigned,char*);
void *flash ,*flash2;
void save(void)

```

```

{
char filename[20];
unsigned size=0;
printf("Enter name");
gets(filename);
flash = S(0,0,700,340,&size);
F(flash,size,filename);
}

void *S(int left,int top,int right,int bottom,unsigned *size)
{
void *image;
*size = imagesize(left,top,right,bottom);
printf("%u",size);
image = malloc(*size);
getimage(left,top,right,bottom,(void *)image);
putimage(left,top,(void *)image,COPY_PUT);
return(image);
}

void F(void *image,unsigned size,char *filename)
{
FILE *f1 = fopen(filename,"w");
fwrite((void *)image,size,1,f1);
fflush(f1);
fclose(f1);
}

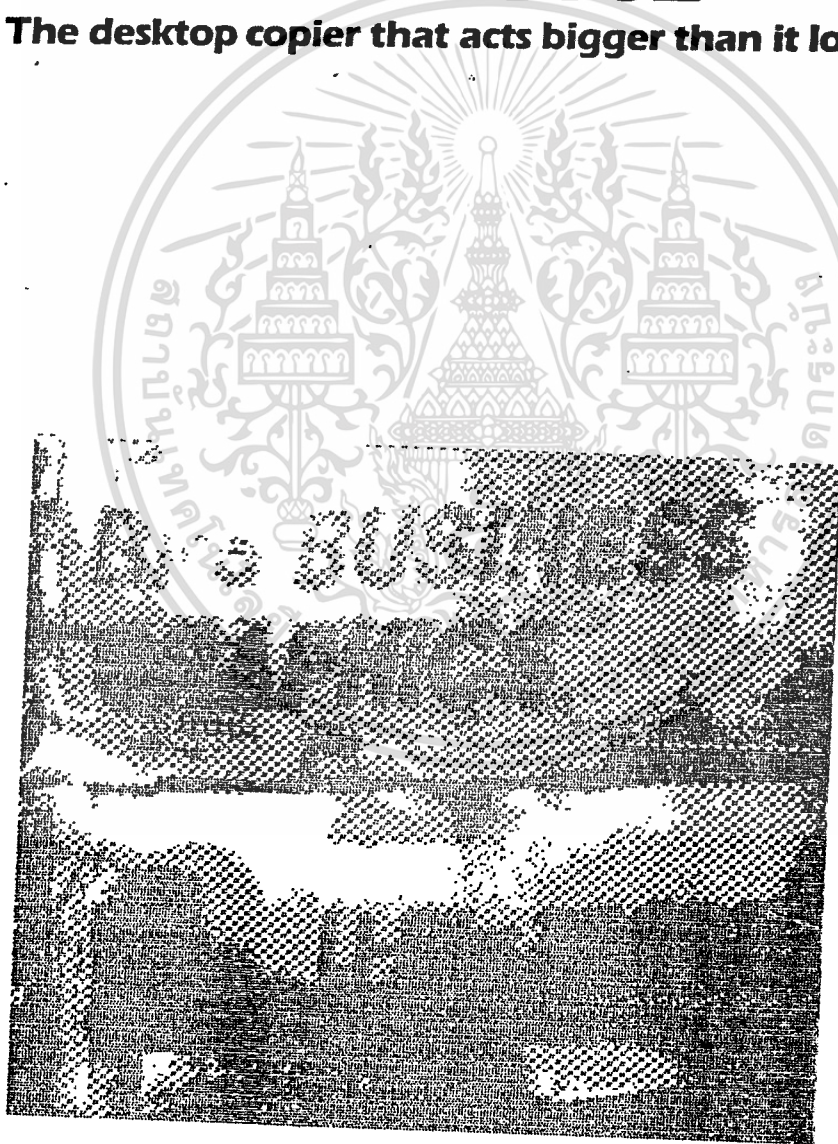
```

BPS BUSINESS GRAPHICS™

NEC

U-Bix 2800Z

The desktop copier that acts bigger than it looks



รูป 5.4 แสดงภาพที่ได้จากการสแกนเฟรมที่พิมพ์กับภาพจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

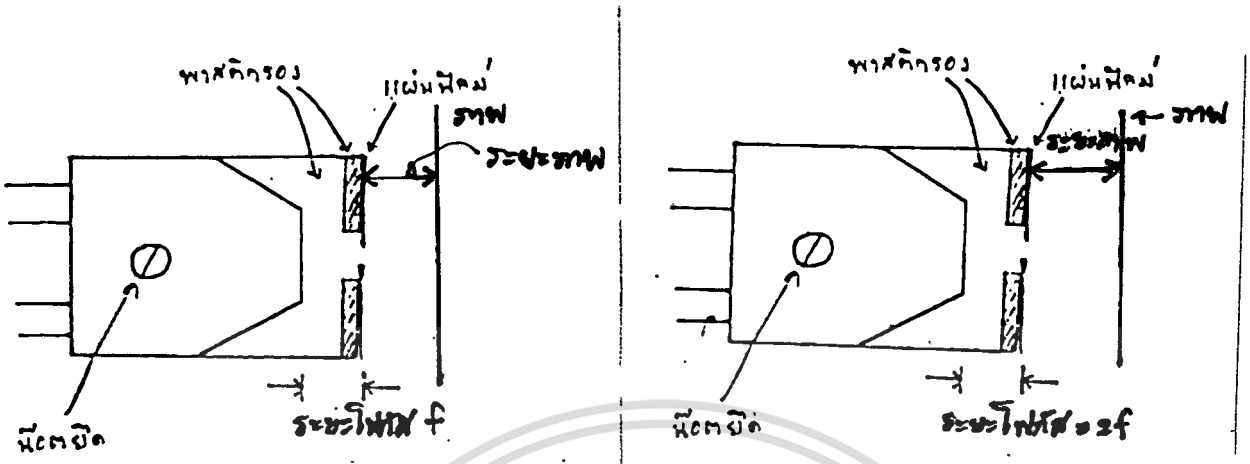
การทดลอง

จากการทำงานของ Scanner มีกลไกการทำงานที่ซับซ้อนพอสมควรซึ่งแต่ละจุดจะมีผลต่อการทำงาน ความแตกต่างของแต่ละจุดจะมีผลต่อภาพ (ขนาด, ความละเอียด) และ เวลาที่ใช้ในการทำงานสามารถที่จะกำหนดค่าให้เป็น parameter ที่สามารถกำหนดขึ้นมา เพื่อให้ได้ลักษณะตามที่ต้องการ ซึ่งเราสามารถโปรแกรมค่าพารามิเตอร์ (Parameter) ได้โดยการสั่งงานทางเมนู ซึ่งได้อธิบายในบทก่อนหน้าแล้ว

จากการทดลองแบ่งพิจารณาปัจจัยภายนอกออกเป็นสองแบบ คือ

1. สภาวะต่างๆที่เปลี่ยนแปลงได้ของพารามิเตอร์ที่ได้จากการโปรแกรม
2. สภาวะที่บังคับไม่ได้ เช่น ปริมาณของแสงสว่าง ซึ่งความสว่างของแสงที่ตกกระทบ จะมีผลต่อการอ่านค่า

ก่อนที่จะอธิบายถึง เรื่องสภาวะทั้งสองสภาวะ จะขออธิบายเรื่องที่เป็นมาตรฐานสำหรับสองสภาวะที่สามารถจะทำได้ คือ เรื่องการปรับและตรวจสอบระยะห่างระหว่างหัวอ่านกับภาพอันเนื่องมาจากความผิดพลาดของระยะทางระหว่างหัวอ่านกับภาพแต่ละจุดจะไม่เท่ากันทำให้การอ่านผิดพลาดไป ถึงแม้ว่าจะเป็นสีเดียวกันก็ตามจึงต้องมีการปรับระยะ เพื่อให้ได้การอ่านภาพมีความผิดพลาดน้อยที่สุด ณ จุดที่ระยะทางระหว่างหัวอ่านกับภาพแตกต่างกันมากก็ตาม ซึ่งจะขึ้นอยู่กับ การปรับปัจจัยต่างๆ เช่น การปรับระยะ โฟกัสของออปโตดีเทคเตอร์ (optodetector) กับรูรับแสงของหัวอ่าน ซึ่งจากการทดลองสามารถตั้งไว้ได้



รูป 6.1 แสดงระยะระหว่างหัวอ่านกับภาพ

รูปที่ 1

รูปที่ 2

ในการทดลองแรกได้ตั้งค่าระยะต่างๆตามรูปที่ 1 ซึ่งจากการทดลองจะเห็นว่าเมื่อระยะระหว่างภาพกับหัวอ่านของแต่ละจุดมีความแตกต่างกันจะทำให้อ่านค่าได้ผิดพลาด ทำให้ภาพที่อ่านได้ผิดพลาด และพบว่าแสงจากภายนอกมีผลต่อการอ่านด้วย ถ้าระยะต่างๆเป็นไปตามรูปที่ 1 อยู่

ในการทดลองที่สองได้ตั้งระยะต่างๆตามรูปที่ 2 ซึ่งจากการทดลองพบว่าสามารถลดความผิดพลาดลงได้มาก เนื่องจากผลของแสงตกกระทบจากภายนอกลดลง

ส่วนการสั่นของแท่นติดหัวอ่านและหลอดสียง เป็นปรากฏการณ์ที่เกิดขึ้นอย่างสุ่มทำให้ไม่สามารถที่จะแก้ไขให้หมดไปได้ แต่พอที่จะป้องกันให้ไม่มีผลต่อการทำงาน โดยการกดน้ำหนักลงบนตัวแท่น หรือใช้ความเสียดทานเข้าช่วย ส่วนการสั่นของหลอดสียงก็ต้องมีการปรับให้มีความตึงที่พอดี เพื่อไม่ให้เกิดการกระชาก ซึ่งเป็นการปรับโดยอาศัยความชำนาญ ที่ไม่สามารถวัดได้ ซึ่งความผิดพลาดจะลดลง แต่อาจเกิดเสียงดังเวลาสแกน

ส่วนที่จะกล่าวถึงต่อไปเป็นผลมาจากปัจจัยภายนอก คือ

1. ส่วนที่โปรแกรมได้ คือ

1.1 ค่าตารางสำหรับกระดาษแต่ละชนิด ซึ่งกระดาษแต่ละชนิด มีการสะท้อนแสงไม่เท่ากัน เนื่องจากมีความเรียบความมันและการดูดกลืนแสงต่างกัน จึงต้องมีการทดลองเพื่อให้ได้ค่าตารางที่ดีที่สุดสำหรับกระดาษแต่ละชนิด ซึ่งในการทดลองใช้กระดาษสี่แบบ ทำให้ได้ตารางที่เหมาะสม 3 ตารางซึ่งได้อธิบายไว้แล้ว

1.2 ค่าตัวหน่วงเวลาสำหรับการย่อและขยายภาพ เป็นตัวกำหนดช่วงเวลาในการอ่านภาพแต่ละจุด เป็นการกำหนดระยะระหว่างจุดซึ่งจะแปรผันตรงกับความละเอียดของภาพ ถ้าค่าหน่วงเวลาน้อยจะทำให้ได้ระยะระหว่างจุดน้อยตาม ทำให้ได้ความละเอียดเพิ่ม ภาพที่ปรากฏบนจอจะขยาย ซึ่งจากการทดลองจะได้ว่าค่าหน่วงเวลาจะต้องสัมพันธ์กับความเร็วมอเตอร์

1.3 ระยะทาง เวลาและความเร็ว จากการทดลองสามารถเขียนความสัมพันธ์ ได้โดยที่ 1 สเติบการหมุนของมอเตอร์ทำให้ได้ระยะทาง $1/64$ นิ้ว ซึ่งสามารถกำหนดเป็นระยะทางที่จะทำการสแกนได้ มีสูตรว่า

$$\text{จำนวนสเติบของมอเตอร์} = 64 * \text{ระยะทาง (นิ้ว)}$$

ในการสั่งการทำงานจะต้องรู้ค่าความถี่ที่จะป้อนให้มอเตอร์ ซึ่งจะมีความสัมพันธ์ดังสูตร

$$F = CLK / (4 * f)$$

F: ค่าตัวหารความถี่คล็อกที่จะต้องโปรแกรมเข้าที่ 8253

CLK: ค่าความถี่คล็อกของคอมพิวเตอร์

f: ค่าความถี่ของมอเตอร์

เนื่องจากรู้ค่าความหน่วงเวลา ค่าความถี่และจำนวนสเติบที่ต้องหมุน สามารถที่จะนำมาแสดงเป็นสูตรของเวลาที่ใช้น้อยที่สุดลดค่าเวลาที่เกิดจากการทำงานไม่สัมพันธ์กันลง

2. ปัจจัยที่ควบคุมไม่ได้ เช่นแสงจากภายนอก เนื่องจากการวางตำแหน่งของหัวอ่านจะมีผลต่อการทำงานของหัวอ่าน แต่จากการทดลองได้พยายามปรับเพื่อให้เกิดผลนี้น้อยที่สุดแล้ว ได้ทำการทดลอง 3 กรณี คือ

2.1 แสงสว่างจากภายนอกเท่านั้น

2.2 แสงสว่างภายนอกพร้อมกับแสงสว่างในห้อง

2.3 แสงสว่างภายนอกพร้อมกับแสงสว่างในห้องร่วมกับโคมไฟ

พบว่า 1 กับ 2 แตกต่างกันน้อยมาก ส่วน 3 จะทำให้ภาพเพี้ยนไป



บทที่ 7

สรุปและวิจารณ์ผลการทดลอง

เครื่องสแกนภาพที่ประดิษฐ์ขึ้น สามารถทำงานได้ตามที่ต้องการได้เป็นที่น่าพอใจ ซึ่งสามารถที่จะโปรแกรมได้เพื่อให้ได้พารามิเตอร์เหมาะสมกับการทำงานดังนี้

- 1) กระจก ที่ใช้ในการสแกนมีความแตกต่างกัน ในด้านคุณสมบัติ ทำให้ต้องมีค่าตารางสำหรับกระจกแต่ละชนิด เพื่อให้ได้ภาพที่ดีที่สุด
- 2) ค่าหน่วย เวลา เพื่อให้ได้ภาพที่มีความละเอียดหรือมีการขยายของภาพ
- 3) ค่าความถี่ ใช้ได้ไม่เกินที่ 700 Hz
- 4) ค่าอื่นๆที่จะทำให้สแกนทำงานได้อย่างถูกต้อง ซึ่งสามารถจะแสดงเป็นความสัมพันธ์เป็นสูตรทางคณิตศาสตร์ได้

ภาพที่ได้จะไม่ได้เหมือนกันทุกประการกับภาพต้นแบบ คือมีความเพี้ยนบนแนวอนไปบ้าง ซึ่งอาจเกิดขึ้นจากขบวนการอ่าน หรือเป็นเพราะการแสดงผลบนหน้าจอผิดเพี้ยนไปตามการทำงานของจอเอง แต่ภาพที่ได้ก็ออกมาที่มีความชัดเจนดี มีความคล้ายกันแต่อาจมีการขยายขนาดบ้าง ซึ่งสามารถยอมรับได้ จากตัวอย่างที่ได้มาจากโปรแกรมพิมพ์ที่มีการย่อแล้ว

ส่วนเวลาที่ใช้ในการสแกนแต่ละครั้งประมาณ 30-35 นาทีต่อรูป (กระดาษ A4) ซึ่งเป็นเวลาที่ใช้ที่สั้นที่สุดโดยให้ความละเอียดของภาพแนวตั้งมากที่สุด ถ้าต้องการให้ได้เวลาที่ลดลง ~~สามารถกำหนด~~กำหนดให้ความละเอียดลดลง

วิจารณ์การทดลอง

เครื่องสแกนภาพที่ประดิษฐ์ขึ้น มีประสิทธิภาพน่าพอใจ เพราะสามารถเปลี่ยนความถี่ได้ตามที่ต้องการ และได้ทำการทดลองหาค่าโตนส์ที่เหมาะสมกับการอ่านภาพจากกระดาษหลายชนิด สามารถเลือกพื้นที่ที่ต้องการสแกนได้ รวมทั้งสามารถพิมพ์ และย่อขยายขนาดภาพได้อีกด้วย ทำให้สะดวกแก่การใช้งานมาก แต่ก็ยังมีข้อผิดพลาดอยู่บ้าง ข้อสำคัญคือหัวอ่านยังแยกสีได้ไม่ดี และทำงานได้ช้าถ้าต้องการปรับปรุงก็ทำได้โดยการเปลี่ยนหัวอ่านที่สามารถแยกสีได้ดีกว่า ถ้าต้องการความละเอียดสูงขึ้นก็ทำได้โดยเปลี่ยนมอเตอร์ให้มีสเปคที่ละเอียดขึ้น และถ้าต้องการให้การทำงานเป็นอิสระ จะทำได้โดยการมีไมโครโปรเซสเซอร์ควบคุมโดยอิสระ ทำให้สามารถทำงานอื่นไปพร้อมกันได้



```
/* ----- scanner.c ----- */
```

```
#define ARRAY(p, y, x) (*(p)+((y)*X_DIMENSION)+(x))
#define STATUS 2
#define PORTNUM 0
#define PORTRAIT 0
#define LANDSCAPE 1
```

```
#include <math.h>
#include <bios.h>
#include <ctype.h>
#include <stdio.h>
#include <process.h>
#include <string.h>
#include <conio.h>
#include <stdlib.h>
#include <dos.h>
#include <mem.h>
#include <alloc.h>
#include <graphics.h>
#include "window.h"
#include "menu.h"
#include "cursor.h"
#include "keys.h"
#define READ_MODE 0
#define WRITE_MODE 1
#define FALSE 0
#define TRUE 1
```

```
static char currentdir[81];
static int X_DIMENSION =256 ;
static int Y_DIMENSION =256;
static int Y_SKIP = 2;
static int X_SKIP = 1;
static int HOR = 0;
static int VER = 1;
static int CW = 0;
static int CCW = 1;
static int STOP = 0;
static int START = 1;
static int CNT0 = 0x300;
static int CNT1 = 0x301 ;
static int CNT2 = 0x302;
static int CNT = 0x303;
static int M_CTRL = 0x304;
static int MAX_STEP= 1400;
static int LUT[2000];
static unsigned char dit2[4][4]={{ 15,35, 35,15},
                                   {35, 15,15,35},
                                   {35,15, 15,35},
                                   {15,35,35,15}};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

static unsigned char lut[256];
static int i,j,k,m,t;
static unsigned char d;
static unsigned char dit[4][4]={{ 15,135, 45,165},
                                  {195, 75,225,105},
                                  { 60,180, 30,150},
                                  {240,120,210, 90}};

static int l=8;
static int d1=3000;
static int d2=200;
static int d3=4000;
static int pattern[8] = {
    0,16,48,176,432,500,0x01bf,511
};
static char far *buffer, far *p;
static char fname[80];
static FILE *fp;
static int s,m1,d9,s1;
static int ipx=1,ipy=2;
static int h=256;
static int v=256;
static int c=0x01E0 ;
static int fq=200;
static long t1;
static unsigned int devide;
static unsigned int lb;
static unsigned int hb;
static int nuvo=0x0040;
static int pun=1;
static int stp;
static float del1,del3;
static int stm=8;
static float time1;
static float time2;
void *S(int,int,int,int,unsigned*);
void F(void*,unsigned,char*);
void *flash ,*flash2;
int quit(void);
int info(void);
int shell(void);
int load(void);
int new(void);
int scn(void);
int dir(void);
int writto(void);
int chgdir(void);
int save(void);
int freq(void);
int dely(void);
int step(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int dimx(void);
int dimy(void);
int prning(void);
int type1(void);
int type2(void);
int type3(void);
int dsplay(void);
int skipx(void);
int skipy(void);
int pos(void);
int test(void);
void home(void);
void beep(void);
void picture(int);
void sve(void);
void ptone1(int x,int y);
void ptone2(int x,int y);
void ptone3(int x,int y);
void print2(int Mode,char Direction);
int reload(void);
int display_image(char far *buff);
int initlut(unsigned char lut[256]);
int initdither(unsigned char dit[4][4]);
int wait (void);
int getname(char *message, char *st, char *ext, int mode);
char *get_string(char *s,int maxlen);
static unsigned bit;
static unsigned bit2;
void motor(int);
void direction(int);
void steptomove(unsigned);
void move(unsigned);

void mo_tor(int);
void di_rection(int);
void step_tomove(unsigned);
void m_ove(unsigned);

char *file[] = {
    "load file",
    "New",
    "Save file",
    "Write to file",
    "Directory",
    "Change Dir",
    "OS shell",
    "Exit to DOS",
    NULL
};
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    "Display picture *    ",
    "X Scale              ",
    "Y Scale              ",
    "Scanner demo        ",
    "Reload demo         ",
    NULL
};

char *scan[] = {
    "Regin scan          ",
    NULL
};

char *print[] = {
    "Print                ",
    "Left margin          ",
    "Right margin         ",
    NULL
};

char *config[] = {
    "Speed                ",
    "Width                ",
    "Delay                ",
    "Position             ",
    "Hor size             ",
    "Ver size             ",
    "A type               ",
    "B type               ",
    "C type               ",
    NULL
};

int (*filefnc[])()={load,new,save,writto,dir,chgdir,shell,quit,NULL};
int (*scanfnc[])()={scn,NULL};
int (*disp[])()={dsplay,skipx,skipy,test,reload,NULL};
int (*prnfn[])(int)={prnimg,dimx,dimy,NULL};
int (*cfg[])()={freq,step,dely,pos,dimx,dimy,type1,type2,type3,NULL};

static MENU mn[] = {
    {"File",      file,      filefnc},
    {"Scan",     scan,      scanfnc},
    {"Print",    print,    prnfn[] },
    {"Display",  display, disp  },
    {"Parameter", config,  cfg    },
    {NULL}
};

void initLUT(int a1,int a2,int a3,int a4,int a5,int a6,int a7,int a8)
{
    register int i;
    for (i=0;i<a1;i++) LUT[i]=7;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    for (i=a1;i<a2;i++) LUT[i]=6;
    for (i=a2;i<a3;i++) LUT[i]=5;
    for (i=a3;i<a4;i++) LUT[i]=4;
    for (i=a4;i<a5;i++) LUT[i]=3;
    for (i=a5;i<a6;i++) LUT[i]=2;
    for (i=a6;i<a7;i++) LUT[i]=1;
    for (i=a7;i<a8;i++) LUT[i]=0;
}

unsigned getint(int maxdigit, int col, int row)
{
char str[20], *p;
char ca;
register int i=0;

p = str;
savecursor();
unhidecursor();
gotoxy(col, row);
while((ca = (char) bioskey(0)) != '\r')
{
    if (ca==27)
    {
        restorecursor();
        hidecursor();
        return -1;
    }
    if (isdigit(ca) && i<maxdigit)
    {
        *p++ = ca;
        i++;
        putchar(ca);
    }
    else if (ca=='\b')
    {
        if (p>str)
        {
            p--;
            i--;
            putchar(8);
            putchar(' ');
            putchar(8);
        }
    }
}
*p = '\0';
hidecursor();
restorecursor();
return atoi(str);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int freq ()
{
    open_window(35,10,70,12,BLACK,WHITE,1,1,35);
    cprintf("SPEED: %d Hz [Not above 700]",fq) ;
    fq = getint(4,8,1);
    close_window();
    t1=fq*4;
    devide = 10000000/t1;
    lb=devide&0x00FF;
    hb=(devide>>8);
    open_window(15,10,70,12,BLACK,WHITE,1,1,35);
    cprintf(" LOW BYTE IS %x   and HIGH BYTE IS %x  ",lb,hb) ;
    getch();
    close_window();
    return 0;
}
int step()
{
    open_window(30,12,70,14,BLACK,WHITE,1,1,35);
    cprintf("Enter paper width:%d  ",stm);
    stm=getint(3,19,1);
    close_window();
    stp=stm*64;
    c=stp;
    open_window(30,15,70,17,BLACK,WHITE,1,1,35);
    cprintf("Your step to move is %d  step",c);
    getch();
    close_window();
    return 0;
}
int dely()
{
    open_window(40,10,60,12,BLACK,WHITE,1,1,35);
    cprintf("DELAY:%d  ms",l);
    l=getint(4,7,1);
    close_window();

    time1=256*(0.03+l);
    time2= (c/fq)*1000;

    if(time2<=time1)
        del1=time1;
    else
    {
        del1=time2-time1+500;
    }
    del3=time2+800;

    d1=del1;
    d3=del3;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

open_window(10,15,70,17,BLACK,WHITE,1,1,35);
cprintf(" DELAY1=%d ms,DELAY2=%d ms,DELAY3=%d ms",d1,d2
,d3);
getch();

close_window();
return 0;
}
int pos(void)
{
open_window(20,12,65,14,BLACK,WHITE,1,1,35);
cprintf("Enter your distance from home:%d inch ",pun);
pun=getint(3,31,1);
close_window();
nuvo=64*pun;

open_window(20,12,55,14,BLACK,WHITE,1,1,35);
cprintf("Your step to move:%x ",nuvo);
getch();
close_window();
return 0;
}

int dimx()
{
open_window(50,12,75,14,BLACK,WHITE,1,1,35);
cprintf("X_DIM: %d ", h);
h=getint(3,8,1);
X_DIMENSION=h;
close_window();
return 0;
}

int dimy()
{
open_window(50,12,75,14,BLACK,WHITE,1,1,35);
cprintf("Y_DIM: %d ", v);
v=getint(3,8,1);
Y_DIMENSION=v;
close_window();
return 0;
}

int skipx()
{
open_window(45,10,70,12,BLACK,WHITE,1,1,35);
cprintf("X_SKIP: %d ", ipx);
ipx=getint(3,9,1);
X_SKIP=ipx;
close_window();
return 0;
}

int skipy()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    open_window(45,10,70,12,BLACK,WHITE,1,1,35);
    cprintf("Y_SKIP: %d      ", ipy);
    ipy=getint(3,9,1);
    Y_SKIP=ipy;
    close_window();
    return 0;
}

int type1()
{
    initLUT(240,252,264,270,282,290,300,600);
    return 0;
}

int type2()
{
    initLUT(320,340,360,370,375,380,384,600);
    return 0;
}

int type3()
{
    initLUT(348,352,354,360,366,372,380,600);
    return 0;
}

int dsplay(void)
{
    int loop,img;
    register x;
    char *textbuff;
    open_window(1,1,80,25,7,0,0,0,0);
    fread((void *) buffer, X_DIMENSION*(Y_DIMENSION/2), 2, fp);
    fclose(fp);
    textbuff=(char *)malloc(4066);
    gettext(1,1,80,25,textbuff);

    loop = DETECT;
    initgraph(&loop, &img, "");
    x = graphresult();
    if (x!=grOk)
    {
        printf("Error: %s\n", grapherrormsg(x));
        farfree(buffer);
        exit(0);
    }
    display_image(buffer);
    if (!getch()) (void) getch();
    cleardevice();
    closegraph();
    puttext(1,1,80,25,textbuff);
    free(textbuff);
    close_window();
    hidecursor();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    return 0;
}

int prnimg(void)
{
    open_window(30,10,65,13,BLACK,WHITE,1,1,35);
    cprintf(" IMAGE SCANNER PRINT Version1.0");
    getkey();
    close_window();

    initlut(lut);

    initdither(dit);

    if (wait()==1) return 0;
    for (i=0;i<32;i++) /* line # */
    {
        /* print bw image */
        fprintf(stdprn,"%c%c%c",27,'A',8); /* set 8/72 line space */
        /* set graphic 256 dots */
        fprintf(stdprn,"%c%c%c%c",27,'*',0,0,1);
        /* column # */
        for (j=0;j<256;j++)
        { d=0;
            /* dot in line */
            for (k=0;k<8;k++)
            {
                d<<=1;
                /*if ((8*lut[buffer[i*8+k][j]]) < dit[(i*8+k)%4][j%4]) d++;*/
                if ((8*lut[ARRAY(buffer, i*8+k, j)]) < dit[(i*8+k)%4][j%4])
                d++;
            }
            fprintf(stdprn,"%c",d);
        }
        fprintf(stdprn,"\n");
    }
    fprintf(stdprn,"%c%c",27,64);
    return 0;
}

int quit(void)
{
    if (yesno(" Quit, are you sure ?"))
        return 1;
    else
        return 0;
}

int load(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{ . . .
char temp[80], tmp[80];
strcpy(temp, ".*");
enable_selection(mn, 1, 3);
enable_selection(mn, 1, 4);
enable_selection(mn, 3, 1);
enable_selection(mn, 4, 1);
open_window(8, 4, 73, 6, YELLOW, BLUE, 1, 1, 30);
cprintf("Load file: ");
textcolor(WHITE);
unhidecursor();
get_string(temp, 50);
close_window();
hidecursor();
getfname(temp, tmp);
strcpy(fname, tmp);
if ((fp = fopen(fname, "rb"))!=NULL)
{
open_window(10,8,40,10,BLACK,WHITE, 1, 1, 30);
cprintf("Can't open file ...");
getkey();
close_window();
return 0;
}
fread((void *) buffer, X_DIMENSION*(Y_DIMENSION/2), 2, fp);
fclose(fp);
return 0;
}

int shell(void)
{
open_window(1, 1, 80, 25, 7, 0, 0, 0, 35);
textattr(7);
clrscr();
unhidecursor();
cprintf("\n\rType EXIT to return to SCANNER . . .\n\r");
textcolor(WHITE);
cprintf("\n\rBe careful to use of any TSR program");
if (system("")== -1) ;
close_window();
hidecursor();
return 0;
}

int new(void)
{
if (yesno(" Warning! Previous data may be lost, try ?"))
{
disable_selection(mn, 1, 3);
disable_selection(mn, 1, 4);
}
}

```



เอกสารนี้เป็นเอกสารที่สวอนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    .disable_selection(mn, 3, 1);
    .disable_selection(mn, 4, 1);
}
return 0;
}
int save(void)
{
char temp[80], tmp[80];
strcpy(temp, ".*");
open_window(7, 6, 72, 8, YELLOW, BLUE, 1, 1, 30);
cprintf("Save to file: ");
textcolor(WHITE);
unhidecursor();
get_string(temp, 50);
hidecursor();
close_window();
strcpy(fname, temp);
if ((fp = fopen(fname, "wb"))==NULL)
{
open_window(20,5,60,7,BLACK,WHITE,1,1,35);
cprintf("Can\'t open file ...Press any key");
getkey();
close_window();
return(0);
}
open_window(20,5,70,7,BLACK,WHITE,1,1,35);
cprintf(" Saving output file, please wait ...");
fwrite((void *) buffer, X_DIMENSION*(Y_DIMENSION/2), 2, fp);
fclose(fp);
close_window();
return 0;
}
void draw_pattern(int xpos,int ypos,int patt)
{
register int x,y;
int color,temp ;

color=pattern[patt];
temp =color&1;
for(y=0;y<3;y++)
for(x=0;x<3;x++)
{
if(temp) putpixel(xpos+x,ypos+y,1);
color >>=1;
temp=color&1;
}
}
int display_image(char far *buff)
{
register int x, y, yy;
char far *p;

```

```

    p = buff;
    for (y=0, yy=0; y<Y_DIMENSION*3; y+=3)
    {
        p = buff + yy*X_DIMENSION;
        yy += Y_SKIP;
        for (x=0; x<X_DIMENSION*3; x+=3)
        {
            draw_pattern(x, y, *p);
            p += X_SKIP;
        }
    }
}

void motor(int m)
{ if(m == HOR){ bit &= 0xF0; }
  else {bit |= 0x08;
        bit &= 0x08; }
}
/*-----*/

void direction(int d)
{ if(d == CW){ bit &= 0xEC; }
  else { bit |= 0x10; }
}
/*-----*/

void steptomove(unsigned s)
{int t,a;
  if((s>0) && (s<MAX_STEP))
  { outportb(CNT,0x72);t=(s & 0x00FF);outportb(CNT1,t);a=s>>8;
    outportb(CNT1,a);}
}

/*-----*/

void move(unsigned s)
{ if(s == START)
  {bit |= 0x0001;
   outportb(M_CTRL,bit |= 0x05);
   outportb(M_CTRL,bit |= 0x02);}
  else { bit &= 0xFE;outportb(M_CTRL,bit);}
}
/*-----*/

int scn(void)
{
    register int x,y;
    int loop,img;
    char temp[80],tmp[80];
    strcpy(temp,"");
    enable_selection(mn, 1, 3);
    enable_selection(mn, 1, 4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

enable_selection(mn, 3, 1);
enable_selection(mn, 4, 1);

open_window(25,15,50,20,BLACK,WHITE,1,1,35);
hidecursor();
printf("  SCANNING IN PROGRESS  ");
printf("  ");
printf("  Press ESC to Quit  ");

home();

outportb(CNT,0x36);
outportb(CNT0,lb);
outportb(CNT0,hb);

motor(VER);
direction(CW);
steptomove(nuvo);
move(START); delay(2000);

for(y=0;y<Y_DIMENSION;y++)
{
  motor(HOR);
  direction(CW);
  steptomove(c);
  move(START);

  for(x=0;x<X_DIMENSION ;x++)
  {
    img=0;
    outportb(0x0201,0);
    do {
      loop = 1&inportb(0x0201);
      img+=6;
    } while(loop);
    *p++ =LUT[img];
    delay(1);
  }delay(d1);
  motor(VER);
  direction(CW);
  steptomove(0x0001);
  move(START);      delay(d2);

  motor(HOR);
  direction(CCW);
  steptomove(c);
  move(START);      delay(d3);

  if (kbhit())
  {
    if(getch()==27)
    {

```

```

        close_window();
    ...return 0;
    }
}
close_window();
sound(2000);delay(1000);nosound();

open_window(25,7,75,9,YELLOW,BLUE,1,1,35);
cprintf("Enter filename:");
unhidecursor();
get_string(temp,50);
hidecursor();
close_window();
strcpy(fname,temp);
if ((fp = fopen(fname, "wb"))==NULL)
{
    open_window(20,5,60,7,BLACK,WHITE,1,1,35);
    cprintf("Can\'t open file ...Press any key");
    getkey();
    close_window();
    return(0);
}
open_window(20,5,70,7,BLACK,WHITE,1,1,35);
cprintf(" Saving output file, please wait ...");
fwrite((void *) buffer, X_DIMENSION*(Y_DIMENSION/2), 2, fp);
fclose(fp);
close_window();
return 0;
}

int writto(void)
{
char temp[80], tmp[80];
strcpy(temp, "*");
open_window(8, 7, 73, 9, YELLOW, BLUE, 1, 1, 30);
cprintf("Write to file: ");
textcolor(WHITE);
unhidecursor();
get_string(temp,50);
hidecursor();
close_window();
strcpy(fname,temp);
if ((fp = fopen(fname, "wb"))==NULL)
{
    open_window(20,5,60,7,BLACK,WHITE,1,1,35);
    cprintf("Can\'t open file ...Press any key");
    getkey();
    close_window();
    return(0);
}
open_window(20,5,70,7,BLACK,WHITE,1,1,35);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    cprintf(" Saving output file, please wait ...");
    fwrite((void *) buffer, X_DIMENSION*(Y_DIMENSION/2), 2, fp);
    fclose(fp);
    close_window();

    return 0;
}

int chgdir(void)
{
char tmp[80];
    strcpy(tmp, currentdir);
    open_window(8, 9, 73, 11, YELLOW, BLUE, 1, 1, 30);
    cprintf("New Dir: ");
    unhidecursor();
    get_string(tmp, 50);
    hidecursor();
    if (tmp[0]!=' ')
        if (chdir(tmp)!=0)
            error_message(" Path not found !");
        else strcpy(currentdir, strupr(tmp));
    close_window();
    return 0;
}

int wait(void)
{
    int status,abyte=0;
    open_window(8,10,48,12,BLACK,WHITE,1,1,35);
    hidecursor();
    cprintf("ON LINE PRINTER...then press any key");
    getch();
    close_window();
    status=biosprint(STATUS,abyte,PORTNUM);
    if(status & 0x08)
    {
        open_window(20,10,50,12,BLACK,WHITE,1,1,35);
        hidecursor();
        cprintf("I/O error ...Press ESC ");
        while(getch()!=27);
        close_window();
        return 1;
    }
    if(status & 0x20)
    {
        open_window(20,10,50,12,BLACK,WHITE,1,1,35);
        hidecursor();
        cprintf(" Out of Paper...Press ESC ");
        while(getch()!=27);
        close_window();
        return 1;
    }
    if (status & 0x10) return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

    printf("IMAGE SCANNER PROJECT");
    restorecursor();
    hidecursor();

    executive(mn);

    close_window();
    unhidecursor();
}

```

```

int test(void)
{
    int drive = DETECT, mode,x,y;
    char *textbuff;
    textbuff=(char *)malloc(4066);
    gettext(1,1,80,25,textbuff);

    initgraph (&drive, &mode, "");
    home();
    outportb(CNT,0x36);
    outportb(CNT0,0x74);
    outportb(CNT0,0x36);

    for(y=0;y<20;y++)
    {
        motor(HOR);
        direction(CW);
        steptomove(0x0200);
        move(START);delay(100);
        picture(y);delay(2500);

        motor(VER);
        direction(CW);
        steptomove(0x0001);
        move(START); delay(20);

        motor(HOR);
        direction(CCW);
        steptomove(0x0200);

        move(START); delay(3500);
        if (kbhit())
        {
            if(getch()==27)
            {
                closegraph();
                puttext(1,1,80,25,textbuff);
                hidecursor();
                free(textbuff);
                return 0;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

do
{
if(getch() == 's')
    sve();
else if(getch() == 'p')
    print2(0,0);
}
    while(getch() != 'n');
    closegraph();
    hidecursor();
    puttext(1,1,80,25,textbuff);
    free(textbuff);
    return 0;
}
/*-----*/

void mo_tor(int m1)
{ if(m1 == HOR){ bit2 &= 0xF7; outportb(M_CTRL,bit2); }
  else {bit2 |= 0x08;outportb(M_CTRL,bit2);}
}
/*-----*/

void di_rection(int d9)
{ if(d9 == CW){ bit2 &= 0xEF; outportb(M_CTRL,bit2);}
  else { bit2 |= 0x10; outportb(M_CTRL,bit2);}
}
/*-----*/

void step_tomove(unsigned s1)
{int t,a;
  if((s1>0) && (s1<MAX_STEP))
  { outportb(CNT,0x72);t=(s1 & 0x00FF);outportb(CNT1,t);
    a=s1>>8;outportb(CNT1,a);}
}
/*-----*/

void m_ove(unsigned s1)
{ if(s1 == START)
  {bit2 |= 0x0001;outportb(M_CTRL,bit2 & 0xFD);outportb(M_CTRL,bit2 |
  0x02);}
  else { bit2 &= 0xFE;outportb(M_CTRL,bit2);}
}
/*-----*/

void home(void)
{
  int n;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outportb(0x303,0x36);
outportb(0x300,0x00);
outportb(0x300,0x04);

outportb(0x303,0x72);
outportb(0x301,0x01);
outportb(0x301,0x00);
do {

    n = inportb(0x307);
    n &= 0x30;
    outportb (0x304,0x00);
    outportb(0x304,0x13); delay(50);
    }while(n != 0x20); beep();
do {
    n = inportb(0x307);
    n &=0x30;
    outportb(0x304,0x01);
    outportb(0x304,0x1B); delay(50);
    }while(n != 0x00); beep();
}

/*-----*/
void beep()
{
sound(1500);delay(40);sound(100);delay(20);nosound();
}
/*-----*/

void picture(yyy)
int yyy;
{int a,x,y,t;
y=yyy;
for(x=0;x<1024;x +=2) {
t=0;
outportb(0x0201,00);
do {
a=inportb(0x0201);
a &=0x01;
t++;
}while(a==1);
delay(3);
if(t<45)
ptone1(x,y);
else if(t>=45 && t<=65)
ptone2(x,y);
else if(t>65)
ptone3(x,y);
}
}

void ptone1(int x,int y)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    putpixel(x,y,1);
    putpixel(x+1,y,1);
}

void ptone2(int x,int y)
{
    putpixel(x,y,1);
    putpixel(x+1,y,0);
}

void ptone3(int x,int y)
{
    putpixel(x,y,0);
    putpixel(x+1,y,0);
}

void sve(void)
{
    char filename[20];
    unsigned size=0;
    printf("Enter name");
    gets(filename);
    flash = S(0,0,700,340,&size);
    F(flash,size,filename);
}

void *S(int left,int top,int right,int bottom,unsigned *size)
{
    void *image;
    *size = imagesize(left,top,right,bottom);
    printf("%u",size);
    image = malloc(*size);
    getimage(left,top,right,bottom,(void *)image);
    putimage(left,top,(void *)image,COPY_PUT);
    return(image);
}

void F(void *image,unsigned size,char *filename)
{
    FILE *f1 = fopen(filename,"w");
    fwrite((void *)image,size,1,f1);
    fflush(f1);
    fclose(f1);
}

void print2(int Mode,char Direction)
{
    char m;
    int i,j,k,Msb,LSb;
    int MaxX,MaxY;
    MaxX =getmaxx();
    MaxY =getmaxy();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setviewport(0,0,MaxX,MaxY,0);
fprintf(stdprn,"%c%c%c",27,51,25);
switch(Direction)
{
  case PORTRAIT:
  {
    Lsb=MaxX&0x00FF;
    Msb=MaxX>>8;
    for(j=0;j<=MaxY/8;j++)
    {
      fprintf(stdprn,"\x1B*%c%c%c",Mode,Lsb,Msb);
      for(i=0;i<=MaxX;i++)
      {
        m=0;
        for(k=0;k<8;k++)
        {
          m<<=1;
          if(getpixel(i,j*8+k)) m++;
        }
        fprintf(stdprn,"%c",m);
      }
      fprintf(stdprn,"\x0D\x0A");
    }
  }
  case LANDSCAPE:
  {
    Lsb=MaxY&0x00FF;
    Msb=MaxY>>8;
    for(j=0;j<MaxX;j+=8)
    {
      fprintf(stdprn,"\x1B*%c%c%c",1,Lsb,Msb);
      for(i=MaxY;i>=0;i--)
      {
        m=0;
        for(k=0;k<8;k++)
        {
          m<<=1;
          if(getpixel(j+k,i)) m++;
        }
        fprintf(stdprn,"%c",m);
      }
      fprintf(stdprn,"\x0D\x0A");
    }
  }
}
fprintf(stdprn,"\f");
}

int reload(void)
{
  int drive = DETECT,mode;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char filename[20];
unsigned x_size,y_size,size;
void *temps;
FILE *f1;
char *textbuff;
register x;

open_window(20,10,50,12,BLACK,WHITE,1,1,35);
cprintf("Enter filename :");
unhidecursor();
gets(filename);
hidecursor();
close_window();
f1= fopen(filename,"r");
textbuff=(char *)malloc(4066);
gettext(1,1,80,25,textbuff);

initgraph(&drive,&mode,"");
x = graphresult();
if (x!=grOk)
{
printf("Error: %s\n", grapherrormsg(x));
exit(0);
}

x_size = fgetc(f1)|(fgetc(f1)<<8);
y_size = fgetc(f1)|(fgetc(f1)<<8);
size = imagesize(0,0,x_size,y_size);
temps =(void *)malloc(size);
rewind(f1);
fread((void *)temps,size,1,f1);
fclose(f1);
putimage(0,0,temps,COPY_PUT);
getch();

closegraph();
puttext(1,1,80,25,textbuff);
hidecursor();
free(textbuff);
return 0;
}

```

CGETS.C

Tuesday, March 26, 1991 6:52 am

```
#include <conio.h>
#include <ctype.h>
#include "keys.h"
```

```
int cgets_returnkey;
```

```
char *cgets(char *s)
{
char *bf = s+2;
int c = 0;
```

```
while ((c = getkey()) != '\r')
{
if (c == BS)
c = '\b';
if (c == '\b')
{
if (bf > s+2)
{
--bf;
cputs("\b \b");
}
}
else if (isprint(c) && bf < s + *s + 1)
{
putch(c);
*bf++ = c;
}
else if (c==ESC || c & 0x80) break;
else putch(BELL);
}
*bf = '\0';
s[1] = bf - s - 1;
cgets_returnkey = c;
return s+2;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURSOR.C

Tuesday, March 26, 1991 6:56 am

```

/* ----- cursor.c ----- */
#include <dos.h>
#include <conio.h>
#include "cursor.h"
#include "interrupt.h"

static void interrupt far (*oldkb)(void);
static void interrupt far newkb(IREGS);
static void invert_block(void);

/* ----- keyboard BIOS (0x16) functions ----- */
#define READKB 0
#define KBSTAT 1

/* ----- video BIOS (0x10) functions ----- */
#define SETCURSOR 2
#define READCURSOR 3
#define READATTRCHAR 8
#define WRITEATTRCHAR 9
#define HIDECURSOR 0x20

static int cursorpos;
static int cursorshape;

/*
 * Low-level get cursor shape and position
 */
static void getcursor(void)
{
    _AH = READCURSOR;
    _BH = 0;
    geninterrupt(VIDEO);
}

/*
 * Save the current cursor configuration
 */
void savecursor(void)
{
    getcursor();
    cursorshape = _CX;
    cursorpos = _DX;
}

/*
 * Restore the saved cursor configuration
 */
void restorecursor(void)
{
    _AH = SETCURSOR;
    _BH = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURSOR.C

Tuesday, March 26, 1991 6:56 am

```

    _DX = cursorpos;
    geninterrupt(VIDEO);
    _AH = SETCURSOR;
    _CX = cursorshape;
    geninterrupt(VIDEO);
}

/*
 * Make a normal cursor
 */
void normalcursor(void)
{
    _AH = SETCURSOR;
    _CX = 0x0607;
    geninterrupt(VIDEO);
}

/*
 * Hide the cursor
 */
void hidecursor(void)
{
    getcursor();
    _CH |= HIDECURSOR;
    _AH = SETCURSOR;
    geninterrupt(VIDEO);
}

/*
 * Unhide the cursor
 */
void unhidecursor(void)
{
    getcursor();
    _CH &= ~HIDECURSOR;
    _AH = SETCURSOR;
    geninterrupt(VIDEO);
}

/*
 * Make a block cursor
 */
void block_cursor(void)
{
    hidecursor();
    oldkb = getvect(KEYBOARD);
    setvect(KEYBOARD, newkb);
}

/*
 * Return to the default blinking cursor
 */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURSOR.C

Tuesday, March 26, 1991 6:56 am

```

void default_cursor(void)
{
    unhidecursor();
    setvect(KEYBOARD,oldkb);
}

/*
 * Interrupt Function to display and clear the block cursor
 */
void interrupt far newkb(IREGS ir)
{
    static int cset = 0;
    static char kfunc;

    kfunc = _AH;
    if (kfunc == READKB || kfunc == KBSTAT)
    {
        if (cset == 0)
        {
            cset = 1;
            invert_block();
        }
        _AX = ir.ax;
        (*oldkb)();
        ir.ax = _AX;
        ir.fl = _FLAGS;
        if (kfunc == READKB && cset == 1)
        {
            cset = 0;
            invert_block();
        }
    }
}

/*
 * Invert the video attribute at the current cursor position
 */
static void invert_block(void)
{
    static int c, at;

    _BH = 0;
    _AH = READATTRCHAR;
    geninterrupt(VIDEO);
    c = _AX;

    at = (c >> 8) & 0xff;
    at = ((at >> 4) & 7) | ((at << 4) & 0x70) | (at & 0x88);

    _CX = 1;
    _BH = 0;
    _BL = at;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CURSOR.C

Tuesday, March 26, 1991 6:56 am

```
_AL = 'c' & 0xff;  
_AH = WRJTEATTRCHAR;  
geninterrupt(VIDEO);  
}
```



DIR.C

Tuesday, January 1, 1980 12:13 am

Page 1

```

#define BG      CYAN
#define FG      YELLOW
#define REVBG   WHITE
#define REVFG   BLACK

#include <io.h>
#include <dos.h>
#include <dir.h>
#include <mem.h>
#include <bios.h>
#include "keys.h"
#include <ctype.h>
#include <alloc.h>
#include <stdio.h>
#include <stdarg.h>
#include <stdlib.h>
#include <string.h>
#include <conio.h>
#include "window.h"
#include "cursor.h"

void getfname(char *mask, char *output);
struct dirinfo { /* Doubly Linked list control structure */
    char fname[13]; /* file name */
    char attr; /* file attribute */
    struct dirinfo *prev; /* previous file */
    struct dirinfo *next; /* next file */
};

struct dirinfo parent;
struct dirinfo *first, *current, *last, *wnode;
int filenum = 0;

int XPOS, YPOS;

#define DIR_SEARCH 0
#define FIL_SEARCH 1

#define ALLFILE (FA_ARCH|FA_HIDDEN|FA_RDONLY|FA_SYSTEM)

void freeall(void)
{
    struct dirinfo *temp;

    while (first)
    {
        temp = first->next;
        free(first);
        first = temp;
    }
    filenum = 0;
    first = current = last = NULL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

void insert(struct dirinfo **start, struct dirinfo *d, struct dirinfo
**end)
{
struct dirinfo *old, *p;

if (*end==NULL) /* if the first file of list */
{
d->next = NULL;
d->prev = NULL;
*start = *end = d;
return;
}
p = *start;
old = NULL;

while (p)
{
if (((p->attr)&FA_DIREC)==0 && ((d->attr)&FA_DIREC)!=0)
{
old = p;
p = p->next;
}
else if (strcmp(p->fname, d->fname)<0)
{
old = p;
p = p->next;
}
else if (strcmp(d->fname, "..")==0)
{
old = p;
p = p->next;
}
else
{
if (p->prev) /* in the middle of file list
*/
{
p->prev->next = d;
d->next = p;
d->prev = p->prev;
p->prev = d;
return;
}
d->next = p; /* new first of file list */
d->prev = NULL;
p->prev = d;
*start = d;
return;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    old->next = d;                               /* end of file list */
    d->next = NULL;
    d->prev = old;
    *end = d;
}

struct dirinfo * search(char firstchar, char attribute)
{
    struct dirinfo *p;
    char keyflag;

    keyflag = (attribute==DIR_SEARCH) ? 0 : FA_DIREC;
    p = current;
    if (p->next) p = p->next;
    else p = NULL;
    while (p)                                     /* search forward */
    {
        if (*(p->fname)==firstchar && ((p->attr)&FA_DIREC)!=keyflag) return p;
        p = p->next;
    }
    p = first;
    while (p)                                     /* search backward */
    {
        if (*(p->fname)==firstchar && ((p->attr)&FA_DIREC)!=keyflag) return p;
        p = p->next;
        if (p==current) break;
    }
    return NULL;
}

unsigned dlltell(struct dirinfo *start, struct dirinfo *curr)
{
    register unsigned i;
    struct dirinfo *p;

    p = start;
    for (i=0; p!=curr && p!=NULL; i++, p = p->next)
        ;
    if (p) return i;
    return 0;
}

struct dirinfo * dllseek(unsigned offset, int direct, struct dirinfo *
from)
{
    struct dirinfo *d;
    register int i;

    d = from;
    for (i=0; i<offset && d!=NULL; i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        if (direct==FWD) d = d->next;
        else d = d->prev;
    }
    return d;
}

void reperr(void)
{
    gotoxy(11, 18);
    cprintf(" To many files ");
}

void scandir(char *mask)
{
    char temp[MAXPATH];
    struct fblk fffile;
    struct dirinfo *p;
    int done;
    char far *olddta;

    / olddta = getdta(); /* save previous DTA */
    strcpy(temp, mask);
    setmem(1+strchr(temp, '\\'), 1, '\0');
    strcat(temp, ".*");
    done = findfirst(mask, &fffile, ALLFILE);
    while (done==0)
    {
        filenum++;
        p = (struct dirinfo *) malloc (sizeof(struct dirinfo));
        if (p==NULL || filenum>=300)
        {
            reperr();
            setdta(olddta);
            return;
        }
        strcpy(p->fname, fffile.ff_name);
        p->attr = fffile.ff_attrib;
        insert(&first, p, &last); /* put into list */
        done = findnext(&fffile);
    }
    done = findfirst(temp, &fffile, ALLFILE|FA_DIREC);
    while (done==0)
    {
        if (((fffile.ff_attrib)&FA_DIREC)!=0)
        {
            if (strcmp(fffile.ff_name, ".")==0 || strcmp(fffile.ff_name, "..")=0)
            {
                done = findnext(&fffile);
                continue;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    filenum++;
    p = (struct dirinfo *) malloc (sizeof(struct dirinfo));
    if (p==NULL)
    {
        reperr();
        setdta(olddta);
        return;
    }
    strcpy(p->fname, fffile.ff_name);
    p->attr = fffile.ff_attrib;
    insert(&first, p, &last);
    done = findnext(&fffile);
}
else done = findnext(&fffile);
}
if (strchr(mask, '\\')!=mask+2)
{
    p = (struct dirinfo *) malloc (sizeof(struct dirinfo));
    if (p==NULL)
    {
        reperr();
        setdta(olddta);
        return;
    }
    strcpy(p->fname, ".");
    p->attr = FA_DIRC;
    filenum++;
    insert(&first, p, &last);
}
setdta(olddta); /* restore previous DTA */
}

void dispfname(struct dirinfo *p)
{
    register int x, y;
    struct dirinfo *d;

    if (filenum==0)
    {
        gotoxy(11, 9);
        cprintf("Can't find file");
        getkey();
        gotoxy(11, 9);
        cprintf(" ");
        return;
    }
    d = p;
    for (y=9; y<18; y++)
        for (x=11; x<70; x+=15)
            gotoxy(x+1, y);

```

```

        cprintf("        ");
        if (d)
        {
            gotoxy(x+1, y);
            cprintf(d->fname);
            if ((d->attr)&FA_DIREC)
                putchar('\\');
            d = d->next;
        }
    }
}

void listdir(char *mask, struct dirinfo *f, char *output)
{
    struct dirinfo *p;
    int c, cc, len;

    open_window(10, 8, 72, 18, FG, BG, 1, 1, 30);
    window(1, 1, 80, 25);
    gotoxy(11, 9);
    cprintf("Scanning, please wait ...");
    freeall();
    scandir(mask);
    gotoxy(11, 9);
    cprintf("        ");

    current = wnode = first;
    dispfname(wnode);

    XPOS = YPOS = 0;
    if (filenum>0)
    {
        do {
            textbackground(REVBG);
            textcolor(REVFG);
            gotoxy(XPOS*15+11, -YPOS+9);
            putchar(' ');
            cprintf(current->fname);
            if ((current->attr)&FA_DIREC) putchar('\\');
            len = 13-strlen(current->fname);
            while (len-->0) putchar(' ');
            c = getkey();
            gotoxy(XPOS*15+11, YPOS+9);
            textbackground(BG);
            textcolor(FG);
            putchar(' ');
            cprintf(current->fname);
            if ((current->attr)&FA_DIREC) putchar('\\');
            len = 13-strlen(current->fname);
            while (len-->0) putchar(' ');
            switch (c)
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 27:
    *(f->fname) = '\0';
    f->attr = 0;
    *output = '\0';
    close_window();
    freeall();
    return;
case '\r':
    strcpy(output, mask);
    *(strrchr(output, '\\')+1) = '\0';
    strcat(output, current->fname);
    strcpy(f->fname, current->fname);
    f->attr = current->attr;
    close_window();
    freeall();
    return;
case BS:
    if (current==first) break;
    if (XPOS==0 && YPOS==0)
    {
        wnode = dllseek(4, BS, wnode);
        current = dllseek(1, BS, current);
        dispfname(wnode);
        XPOS = 3;
    }
    else
    {
        if (--XPOS<0)
        {
            XPOS = 3;
            if (--YPOS<0) YPOS = 0;
        }
        current = dllseek(1, BS, current);
    }
    break;
case UP:
    if (current==first) break;
    if (--YPOS<0)
    {
        YPOS = 0;
        if ((p=dllseek(4, BS, wnode))==NULL) break;
        else
        {
            wnode = p;
            current = dllseek(4, BS, current);
            dispfname(wnode);
        }
    }
    else current = dllseek(4, BS, current);
    break;
case FWD:
    if (current==last) break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (XPOS==3 && YPOS==8)
{
  wnode = dllseek(4, FWD, wnode);
  current = dllseek(1, FWD, current);
  dispfname(wnode);
  XPOS = 0;
}
else
{
  if (++XPOS>3)
  {
    XPOS = 0;
    if (++YPOS>8) YPOS = 8;
  }
  current = dllseek(1, FWD, current);
}
break;
case DN:
  if (current==last) break;
  if (dlltell(wnode, last)<36)
  {
    cc = ((filenum%4)==0) ? 4 : (filenum%4);
    if ((p=dllseek(4, FWD, current))==NULL) break;
    /*current = dllseek(4-XPOS+cc-1, BWD, last);*/
    else current = p;
    YPOS = dlltell(wnode, current)/4;
  }
  else
  {
    if (++YPOS>8)
    {
      YPOS = 8;
      wnode = dllseek(4, FWD, wnode);
      dispfname(wnode);
    }
    if ((p=dllseek(4, FWD, current))==NULL)
    {
      YPOS = dlltell(wnode, last)/4;
      cc = ((filenum%4)==0) ? 4 : (filenum%4);
      if (XPOS>=cc)
      {
        current = dllseek(4-XPOS+cc-1, BS, last);
        YPOS = dlltell(wnode, current)/4;
      }
      else current = dllseek(cc-XPOS-1, BS, last);
    }
    else current = p;
  }
  break;
case PGUP:
  if (wnode==first)
  {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        current = dllseek(XPOS, FWD, first);
        YPOS = 0;
    }
    else
    {
        if ((wnode=dllseek(36, BS, wnode))!=NULL) wnode = first;
        if ((current=dllseek(36, BS, current))!=NULL)
            current = dllseek(XPOS, FWD, wnode);
        dispfname(wnode);
        YPOS = dlltell(wnode, current)/4;
        /* XPOS = dlltell(wnode, current)%4; */
    }
    break;
case PGDN:
    if (current==last) break;
    if ((cc=dlltell(wnode, last))<36)
    {
        YPOS = cc/4;
        cc = ((filenum%4)==0) ? 4 : (filenum%4);
        if (XPOS>=cc)
        {
            current = dllseek(4-XPOS+cc-1, BS, last);
            YPOS = dlltell(wnode, current)/4;
        }
        else
            current = dllseek(cc-XPOS-1, BS, last);
    }
    else
    {
        wnode = dllseek(36, FWD, wnode);
        if ((current=dllseek(36, FWD, current))!=NULL)
        {
            wnode = dllseek(4, BS, wnode);
            YPOS = dlltell(wnode, last)/4;
            cc = ((filenum%4)==0) ? 4 : (filenum%4);
            if (XPOS>=cc)
            {
                current = dllseek(4-XPOS+cc-1, BS, last);
                YPOS = dlltell(wnode, current)/4;
            }
            else
                current = dllseek(cc-XPOS-1, BS, last);
        }
        dispfname(wnode);
    }
    break;
case HOME:
    if (wnode==first)
    {
        current = first;
        XPOS = YPOS = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    else
    {
        wnode = current = first;
        dispfname(wnode);
        XPOS = YPOS = 0;
    }
    break;
case END:
    if (current==last) break;
    if (dlltell(wnode, last)<36)
    {
        current = last;
        YPOS = dlltell(wnode, current)/4;
        XPOS = dlltell(wnode, current)%4;
    }
    else
    {
        cc = ((filenum%4)==0) ? 4 : (filenum%4);
        wnode = dllseek(31+cc, BS, last);
        current = last;
        dispfname(wnode);
        XPOS = cc-1;
        YPOS = 8;
    }
} while (1);
}
close_window();
*(f->fname) = '\0';
f->attr = 0;
*output = '\0';
freeall();
}

void getlfname(char *mask, char *output)
{
    char olddir[MAXPATH], drive[MAXDRIVE], dir[MAXDIR];
    char name[MAXFILE], ext[MAXEXT], path[MAXPATH];
    int flag;
    struct dirinfo f;

    if (*mask==' ') return;
    strcpy(path, mask);
    strupr(path);

    flag = fnsplit(path, drive, dir, name, ext);
    if ((flag&DRIVE)==0)
    {
        strcpy(drive, "A:");
        drive[0] = getdisk() + 'A';
    }
    fnmerge(path, drive, dir, name, ext);
}

```

```

if ((flag&WILDCARDS)==0)
{
    *olddir = drive[0];
    *(olddir+1) = ':';
    *(olddir+2) = '\\';
    getcurdir(drive[0]+1-'A', olddir+3);
    if (chdir(path)!=-1)
    if (*(path+strlen(path)-1)!='\\')
        strcat(path, "\\");
    chdir(olddir);
    strcpy(dir, path+2);
}

flag = fnsplit(path, drive, dir, name, ext);
if ((flag&DRIVE)==0)
{
    strcpy(drive, "A:");
    drive[0] = getdisk() + 'A';
}
if ((flag&DIRECTORY)==0)
{
    *dir = '\\';
    getcurdir(1 + drive[0] - 'A', (char *) dir+1);
    if (*(dir+strlen(dir))!='\\' && *(dir+1)!='\0')
        strcat(dir, "\\");
}
if ((flag&FILENAME)==0) strcpy(name, "*");
if ((flag&EXTENSION)==0) strcpy(ext, ".*");

fnmerge(path, drive, dir, name, ext);

do {
    listdir(path, &f, output);
    flag = (f.attr)&FA_DIREC;
    switch (flag)
    {
        case 0:
            return;
        default:
            strcpy(path, output);
            if (strcmp(f.fname, "..")==0)
            {
                *(strchr(path, '\\')) = '\0';
                *(strchr(path, '\\')+1) = '\0';
            }
            else strcat(path, "\\");
    }
    strcat(path, name);
    strcat(path, ext);
} while (((f.attr)&FA_DIREC)!=0);
}

```

```
int dir(void)
{
char temp[80];
getfname("*.*", temp);
return 0;
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WINDOW C

Tuesday, March 26, 1991 6:49 am

```

/* ----- window.c ----- */

#include <dos.h>
#include <stdlib.h>
#include <conio.h>
#include <string.h>
#include <bios.h>
#include <ctype.h>
#include "window.h"
#include "keys.h"
#include "cursor.h"

/* ----- window definition structures ----- */
struct text_info windows [MAX_WINDOWS];
struct text_info wkw; /* working window */

/* ----- window dressing ----- */
struct {
    int frame; /* true if the window has a frame */
    int shadow; /* 0 if the window has no shadow
                /* 1 if the window has a transparent shadow
                /* 2 if the window has an opaque shadow
    char *wsave; /* points to the video memory save buffer
} dressing [MAX_WINDOWS];

int curr_wnd = 1;
static void clear_window(int);
static int post_message(char *, int, int, int);

/* ----- open a new window ----- */
void open_window(left,top,right,bottom,foreg,backg,frame,shadow,
{
int bFSIZE;
int sinc = 0;

if (shadow && right < 79 && bottom < 24) sinc = 2;
bFSIZE = (bottom-top+1+sinc) * (right-left+1+sinc) * 2;
if (curr_wnd < MAX_WINDOWS)
{
if (curr_wnd == 1)
    gettextinfo(windows);
else
{
windows[curr_wnd-1].curx = wherex();
windows[curr_wnd-1].cury = wherey();
}
if ((dressing[curr_wnd].wsave=malloc(bFSIZE))!=NULL)
{
gettext(left, top, right+sinc, bottom+sinc,
dressing[curr_wnd].wsave);
window(left,top,right,bottom);
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WINDOW.C

Tuesday, March 26, 1991 6:49 am

```

    textcolor(foreg);
    textbackground(backg);
    gettextinfo(&wkw);
    dressing[curr_wnd].frame = frame;
    dressing[curr_wnd].shadow = shadow;
    clear_window(delaytime);
    windows[curr_wnd++] = wkw;
}
}
}

/* ----- window frame characters ----- */
#define NW (dressing[curr_wnd].frame == 1 ? '\xda' : '\xc9')
#define NE (dressing[curr_wnd].frame == 1 ? '\xbf' : '\xbb')
#define SE (dressing[curr_wnd].frame == 1 ? '\xd9' : '\xbc')
#define SW (dressing[curr_wnd].frame == 1 ? '\xc0' : '\xc8')
#define SIDE (dressing[curr_wnd].frame == 1 ? '\xb3' : '\xba')
#define LINE (dressing[curr_wnd].frame == 1 ? '\xc4' : '\xcd')

/* ----- blank the window and draw its frame ----- */
static void clear_window(int delaytime)
{
int x, y, c, d;
int ht = wkw.winbottom - wkw.wintop + 1;
int wd = wkw.winright - wkw.winleft + 1;
char line[81];

if(wd>10)
{
for(c = wd/2,d=ht/2;c < wd-4;c+=6,d+=2)
{
x = c+wkw.winleft ;
y = d+wkw.wintop ;
if(y<wkw.winbottom)
window(x-2*(c-wd/2),y-2*(d-ht/2),x,y);
else window(x-2*(c-wd/2),wkw.wintop,x,wkw.winbottom);
clrscr();
delay(delaytime);
}
}
window(wkw.winleft,wkw.wintop,wkw.winright,wkw.winbottom);

clrscr();
if (dressing[curr_wnd].shadow)
{
textcolor(LIGHTGRAY);
textbackground(BLACK);
window(wkw.winleft,wkw.wintop,wkw.winright+2, wkw.winbottom+2);
for (y = 2; y < ht+1; y++)
{
gotoxy(wd+1,y);
putch(dressing[curr_wnd].shadow == 2 ? ' ' :

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        *(dressing[curr_wnd].wsave+((wd+2)*y-1)*2));
    putchar(dressing[curr_wnd].shadow == 2 ? ' ' :
        *(dressing[curr_wnd].wsave+((wd+2)*y-1)*2));
}
gotoxy(3, ht+1);
for (x = 1; x <= wd; x++)
    putchar(dressing[curr_wnd].shadow == 2 ? ' ' :
        *(dressing[curr_wnd].wsave+(((wd+2)*ht+1)+x)*2));
window(wkw.winleft,wkw.wintop,wkw.winright,wkw.winbottom);
textattr(wkw.attribute);
}
if (dressing[curr_wnd].frame)
{
    window(wkw.winleft,wkw.wintop,wkw.winright, wkw.winbottom+1);
    memset(line+1,LINE,wd-2);
    line[0] = NW;
    line[wd-1] = NE;
    line[wd] = '\0';
    cputs(line);
    for (y = 2; y < ht; y++)
    {
        gotoxy(1,y);
        putchar(SIDE);
        gotoxy(wd,y);
        putchar(SIDE);
    }
    line[0] = SW;
    line[wd-1] = SE;
    cputs(line);
    window(wkw.winleft+1,wkw.wintop+1,wkw.winright-1,wkw.winbottom-1);
    wkw.curx = wkw.cury = 1;
    gotoxy(1, 1);
}
}

/* ----- close a window ----- */
void close_window(void)
{
    int sinc = 0;

    if (dressing[curr_wnd-1].shadow) `sinc = 2;
    if (curr_wnd > 1)
    {
        puttext(wkw.winleft,wkw.wintop,wkw.winright+sinc,wkw.winbottom+sinc,
            dressing[curr_wnd-1].wsave);
        free(dressing[curr_wnd-1].wsave);
        wkw = windows[--curr_wnd]-1;
        textattr(wkw.attribute);
        if (dressing[curr_wnd-1].frame)
            window(wkw.winleft+1,wkw.wintop+1,
                wkw.winright-1,wkw.winbottom-1);
        else

```

WINDOW.C

Tuesday, March 26, 1991 6:49 am

```

        window(wkw.winleft,wkw.winlop,wkw.winright,wkw.winbottom);
        gotoxy(wkw.curx, wkw.cury);
    }
}

/* ----- write an error message ----- */
void error_message(char *errmsg)
{
    post_message(errmsg, YELLOW, 128|RED, FALSE);
}

/* ----- write a notice ----- */
void notice(char *note)
{
    post_message(note, WHITE, GREEN, FALSE);
}

/* ----- ask a question ----- */
int yesno(char *ques)
{
    return post_message(ques, WHITE, LIGHTCYAN, TRUE);
}

/* ----- post a message ----- */
static int post_message(char *s,int foreg,int backg,int test)
{
    int c;
    int lf = (80-strlen(s)+2)/2;
    int rt = lf+max(strlen(s)+2,15);
    open_window(lf, 11, rt, 14, foreg, backg, 1, 1, 60);
    cputs(s);
    if (backg == RED)  putch(BELL);
    cputs(test ? "\r\n (Y/N) ..." : "\r\n Any key ...");
    unhidecursor();
    do {
        c = bioskey(0) & 255;
    } while (test && tolower(c) != 'y' && tolower(c) != 'n');
    hidecursor();
    close_window();
    return (test && tolower(c) == 'y');
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MENU.C

Tuesday, March 26, 1991 6:45 am

```

/* ----- menu.c ----- */

#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <string.h>
#include <ctype.h>
#include "window.h"
#include "menu.h"
#include "keys.h"
#include "cursor.h"

#define ON 1
#define OFF 0

static int getvmn(MENU *, int *);
static void dimension(char **,int *,int *);
static void light(MENU *, int, int);
static void vlight(MENU *, int, int, int);

static int lastvsel[] = {1,1,1,1,1,1};

/* ----- display & process a menu ----- */
void executive(MENU *mn)
{
extern struct text_info wkw;
int hsel=1, vsel, frtn = FALSE;
int i = 0;

open_window(1,1,80,1,MENUG,MEUBG,0,0,50);
cprintf(" ");
while ((mn+i)->mname) cprintf(" %-10.10s ", (mn+i++)->mname);
light(mn, hsel, ON);
while (!frtn && ((vsel = getvmn(mn, &hsel)) != 0))
{
vlight(mn, hsel, vsel, OFF);
window(1, 1, 80, i);
light(mn, hsel, OFF);
window(wkw.winleft, wkw.wintop, wkw.winright, wkw.winbottom);
frtn = (*(mn+hsel-1)->func [vsel-1])();
close_window();
window(1, 1, 80, 1);
light(mn, hsel, ON);
window(wkw.winleft, wkw.wintop, wkw.winright, wkw.winbottom);
}
close_window();
}

/* ----- pop down a vertical menu ----- */
static int getvmn(MENU *mn, int *hsel)
{
int ht, wd, vx, sel = 0, vsel, vs;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MENU.C

Tuesday, March 26, 1991 6:45 am

```

char **selcs;

while (/*sel != ESC &&*/ sel != '\r')
{
    dimension((mn+*hsel-1)->m selcs, &ht, &wd);
    vx = 5+(*hsel-1)*12;
    open_window(vx, 2, vx+wd+1, ht+3, MENUFG, MENUFG,1,1,60);
    selcs = (mn+*hsel-1)->m selcs;
    for (vsel = 1; *selcs; vsel++)
    {
        textcolor(enabled(mn,*hsel,vsel)?MENUFG:DISABLEFG);
        cprintf(vsel == 1 ? " %s" : "\r\n %s",*selcs++);
    }
    textcolor(MENUFG);
    sel = 0;
    vsel = lastvsel[*hsel-1];
    while (sel != ESC && sel != '\r' && sel != FWD && sel != BS)
    {
        vlight(mn, *hsel, vsel, ON);
        sel = enabled(mn, *hsel, vsel) ? getkey() : (sel == UP ? UP
);
        while (sel==ESC)
            sel = enabled(mn, *hsel, vsel) ? getkey() : (sel == UP ? UP
DN);
        selcs = (mn+*hsel-1)->m selcs;
        for (vs = 1; *selcs; vs++, selcs++)
        {
            if (enabled(mn,*hsel,vs) && tolower(**selcs) == tolower(sel)
)
            {
                vlight(mn, *hsel, vsel, OFF);
                vsel = vs;
                vlight(mn, *hsel, vsel, ON);
                sel = '\r';
                break;
            }
        }
    }
    switch (sel)
    {
        case UP:
            vlight(mn, *hsel, vsel, OFF);
            if (vsel > 1) --vsel;
            else vsel = ht;
            vlight(mn, *hsel, vsel, ON);
            break;
        case DN:
            vlight(mn, *hsel, vsel, OFF);
            if (vsel < ht) vsel++;
            else vsel = 1;
            vlight(mn, *hsel, vsel, ON);
            break;
        case '\r':
            if ((mn+*hsel-1)->func [vsel-1] == NULL)

```

เอกสารนี้เป็นเอกสารทลงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MENU.C

Tuesday, March 26, 1991 6:45 am

```

    {
        if (test_option(mn,*hsel,vsel)) clear_option(mn,*hsel
);
        else set_option(mn,*hsel,vsel);
        vlight(mn, *hsel, vsel, ON);
        sel = 0;
    }
    /*else close_window();*/
    break;
case FWD:
    close_window();
    light(mn, *hsel, OFF);
    if ((mn+*hsel)->mname) (*hsel)++;
    else *hsel = 1;
    light(mn, *hsel, ON);
    break;
case BS:
    close_window();
    light(mn, *hsel, OFF);
    if (*hsel != 1)
        while ((mn+*hsel)->mname) (*hsel)++;
    else --(*hsel);
    light(mn, *hsel, ON);
    break;
case ESC:
    close_window();
    break;
}
}
}
return sel == ESC ? 0 : vsel;
}

/* ----- compute a menu's height & width ----- */
static void dimension(char *sl[], int *ht, int *wd)
{
    *ht = *wd = 0;
    while (sl && sl [*ht])
    {
        *wd = max(*wd, (unsigned) strlen(sl [*ht]) + 2);
        (*ht)++;
    }
}

/* ----- accent a horizontal menu selection ----- */
static void light(MENU *mn, int hsel, int onoff)
{
    textcolor(onoff ? SELECTFG : MENUFG);
    textbackground(onoff ? SELECTBG : MENUBG);
    gotoxy((hsel-1)*12+6, 1);
    cprintf((mn+hsel-1)->mname);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MENU.C

Tuesday, March 26, 1991 6:45 am

```

}

/* ----- accent a vertical menu selection ----- */
static void vlight(MENU *mn, int hsel, int vsel, int onoff)
{
    textcolor(onoff ? SELECTFG : enabled(mn, hsel), vsel) ? MENUFG :
    EFG);
    textbackground(onoff ? SELECTBG : MENUBG);
    gotoxy(1, vsel);
    cprintf(" %s", *((mn+hsel-1)->mselfs+vsel-1));
    lastvsel[hsel-1] = vsel;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ รัตติกร วรากุลศิริพันธุ์ และ นักศึกษาปริญญาโท เป็น
อย่างสูงที่ช่วยให้คาบปรึกษา จนโครงการนี้สำเร็จลุล่วงไปด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. International bussiness Matchines, "IBM TECHNICAL REFERENCE.
2. Kenjo, Takashi, "Stepping motors and thier micro processor controls, Clarendon press ,Oxford 1984
3. Barkakati Nabajyati, "The Wait group's TURBO C Bible, Howard W. Sam & Son Company, 1989, 921 p.
4. Nelson Johnson, "Advance Graphics in C", Osborne Mcgraw-Hill, Berkeley, CA, 1987, 670 p.
5. H. Miller Lawrence, "The Official Borland TURBO C, Servival Guide", John Wiley & Sons 1989 , 320 p.