การแสดงผลภาพแบบด็อทเมตริกซ์

DOT MATRIX DISPLAY

โดย

นายโสภาค ศรีลำ

นายอนุสิษฐ์ วิทยสินธนา

นายอิทธิพงษ์ ลีวงค์วัฒน์

วัน เดือน ปี 30 ก.ค. 2540

เลขทะเบียน 036964

เลขเรียกหนังสือ T 38057 ส.๒๒ ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

036964

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2538

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การแสดงผลภาพแบบด็อทเมตริกซ์ (Dot Matrix Display)

จัดทำโดย

1. นายโสภาค ศรีลำ
2. นายอนุสิษฐ์ วิทยสินธนา
3. นายอิทธิพงษ์ ลีวงค์วัฒน์

.....อาจารย์ที่ปรึกษา

(อาจารย์เกียรติวรรณ ทรงสัตย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลภาพแบบด็อทเมตริกซ์
DOT MATRIX DISPLAY

โดย

นายโสภาค ศรีลำ

นายอนุสิษฐ์ วิทยสินธนา

นายอิทธิพงษ์ ลีวงค์วัฒน์

อาจารย์ที่ปรึกษา

อาจารย์เกียรติวรรณ ทรงสัตย์

บทคัดย่อ

การแสดงผลภาพแบบด็อทเมตริกซ์บนบอร์ดแสดงผลซึ่งใช้แอลอีดี หรือไดโอดปล่อยแสงชนิด 3 สี สามารถนำมาใช้แสดงข้อมูลได้ทั้งที่เป็นตัวอักษร และรูปภาพกราฟฟิก โดยใช้ไมโครคอนโทรลเลอร์ควบคุมการรับส่งข้อมูลที่จะออกแสดงผล โครงงานนี้ได้พัฒนาให้ไมโครคอนโทรลเลอร์สามารถรับข้อมูล และรูปแบบการแสดงผลผ่านทางคอมพิวเตอร์ส่วนบุคคล โดยการใช้เมาส์กำหนดตำแหน่ง และสีของแอลอีดีบนบอร์ดแสดงผลที่จำลองไว้บนหน้าจอของคอมพิวเตอร์ และได้พัฒนาให้การแสดงผลภาพเป็นได้หลายรูปแบบทั้งแบบภาพนิ่ง ข้อความต่อเนื่อง และรูปภาพเคลื่อนไหว

ABSTRACT

Dot matrix display on dot matrix display board comprising LEDs (or Light Emitting Diodes) of 3-colour type can be presented for both character and graphic display. For data transfer controlling, microcontroller is in use. This project presents facility in data receiving of the microcontroller. That is, the microcontroller can receive data from personal computer of which users can click mouse at any position of simulated display board on monitor. Each position of the simulated display board represents an LED that users can turn on or off and give one of three colours to. Besides, this project presents displaying of variable displays including fixed display, long-statement run and motion pictures.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับคำแนะนำ ความรู้ และความช่วยเหลือจากอาจารย์เกียรติวรรณ ทรงสัตย์ อาจารย์ที่ปรึกษา จึงขอขอบพระคุณมา ณ ที่นี้ด้วย

ขอขอบพระคุณคุณพ่อ คุณแม่ และพี่น้อง ที่ให้กำลังใจมาโดยตลอด

และขอขอบพระคุณเพื่อน ๆ ที่ให้ความช่วยเหลือ กำลังใจ และความเป็นเพื่อนที่ดีตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	i
กิตติกรรมประกาศ	ii
สารบัญ	iii
สารบัญตาราง	v
สารบัญภาพ	vi
บทที่ 1 บทนำ	1
บทที่ 2 บอร์ดแสดงข้อมูล	5
2.1 หลักการออกแบบ	5
2.2 คุณลักษณะของส่วนประกอบต่าง ๆ ของบอร์ดแสดงข้อมูล	8
2.2.1 ส่วนแสดงผล	8
2.2.2 ส่วนตัวขับ	16
2.2.3 ส่วนควบคุม และพักข้อมูล	20
บทที่ 3 ไมโครคอนโทรลเลอร์	31
หลักการทำงานของไมโครคอนโทรลเลอร์	35
3.1 ช่วงการรับข้อมูลจากคอมพิวเตอร์	35
3.2 ช่วงการส่งข้อมูลไปยังบอร์ดแสดงข้อมูล	38
บทที่ 4 คอมพิวเตอร์ส่วนบุคคล	41
4.1 รูปแบบของข้อมูลข่าวสารที่จะแสดงออกทางบอร์ดแสดงข้อมูล	41
4.2 รูปแบบในการแสดงข้อมูลข่าวสาร	42
4.2.1 การแสดงข้อมูลข่าวสารแบบภาพนิ่ง	42
4.2.2 การแสดงข้อมูลข่าวสารแบบเคลื่อนไหว	42
4.2.2.1 การแสดงข้อความต่อเนื่อง	42
4.2.2.2 การแสดงตัวอักษร และรูปภาพใน 4 ทิศทาง	43
4.2.2.3 การแสดงรูปภาพเคลื่อนไหว	45
สรุปผล และวิจารณ์	47
หนังสืออ้างอิง	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ
(ต่อ)

	หน้า
ภาคผนวก	50
ภาคผนวก ก. ข้อมูลทางเทคนิคของอุปกรณ์บนบอร์ดแสดงข้อมูล	51
ภาคผนวก ข. รีจิสเตอร์ที่สำคัญของไมโครคอนโทรลเลอร์ Z80180	62
ภาคผนวก ค. รายละเอียดโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์	69
ค-1 โปรแกรมรับข้อมูลจากคอมพิวเตอร์ (RECEIVE.ASM)	70
ค-2 โปรแกรมส่งข้อมูลไปแสดงผลที่บอร์ดแสดงผล (DISPLAY.ASM)	71
ภาคผนวก ง. รายละเอียดโปรแกรมควบคุมการทำงานของไมโครคอมพิวเตอร์	76
ง-1 PROJECT.EXE	77
ง-2 EDIT_FON.EXE	166
ภาคผนวก จ. วิธีการใช้โปรแกรมบนคอมพิวเตอร์	181
- วิธีการใช้โปรแกรม PROJECT.EXE	182
- วิธีการใช้โปรแกรม EDIT_FON.EXE	198

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 3.1	แสดงค่าบิตต่าง ๆ ที่ใช้กำหนดค่าแอดเดรสเริ่มต้นของรูทีนบริการการอินเทอร์รัพท์	38
ตารางที่ ข-1	แสดงรูปแบบต่าง ๆ ของข้อมูล เมื่อค่าที่บิต MOD2, 1, 0 แตกต่างกันไป	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

รูปที่		หน้า
1.1	แสดงส่วนประกอบที่สำคัญของโครงงาน	2
2.1	แสดงบล็อคไดอะแกรมของโครงสร้างทางฮาร์ดแวร์ และการส่งผ่านข้อมูลระหว่างส่วนประกอบต่าง ๆ ของโครงงาน	7
2.2	(ก) แสดงสัญลักษณ์ทางไฟฟ้าของแอลอีดี	9
	(ข) แสดงโครงสร้าง และการเกิดอิเล็กโตรลูมิเนสเซนส์ของแอลอีดี	9
2.3	(ก) แสดงรูปร่างของแอลอีดี	10
	(ข) แสดงค่าพิกัดของพารามิเตอร์ต่าง ๆ ของแอลอีดี	10
	(ค) แสดงคุณสมบัติของแอลอีดีทางไฟฟ้า และทางแสง	11
	(ง) แสดงกราฟความสัมพันธ์ของความเข้มแสงต่อความยาวคลื่น	11
	(จ) แสดงกราฟการเปลี่ยนแปลงของกระแสฟอร์เวิร์ดตามแรงดันฟอร์เวิร์ด	12
	(ฉ) แสดงกราฟความสัมพันธ์ของความเข้มแสงต่อกระแสฟอร์เวิร์ด	12
	(ช) แสดงกราฟการเปลี่ยนแปลงของประสิทธิภาพตามกระแสพีค	13
	(ซ) แสดงกราฟการเปลี่ยนแปลงของกระแสพีคสูงสุดตามความกว้างพัลซ์	13
	(ฌ) แสดงกราฟความสัมพันธ์ของการส่องแสงต่อมุมมอง	13
2.4	แสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านแถว	14
2.5	แสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านคอลัมน์	15
2.6	แสดงวงจรทางไฟฟ้าของตัวขับทางด้านแถวและคอลัมน์ ต่อ 1 จุด	17
2.7	แสดงวงจรและการเชื่อมต่อของการทำงานทางด้านคอลัมน์	19
2.8	แสดงวงจร Decode Port Latch ทั้ง 24 ตัว โดยใช้ 74LS139 และ 74LS138	25
2.9	แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 1	26
2.10	แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 2	27
2.11	แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 3	28
2.12	แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 4	29
2.13	แสดงวงจรทั้งหมดของส่วนวงจรควบคุมและพักข้อมูล	30
3.1	แสดงบล็อคไดอะแกรมของส่วนประกอบต่าง ๆ ของไมโครคอนโทรลเลอร์ Z80180	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ
(ต่อ)

รูปที่		หน้า
3.2	แสดงบล็อคไดอะแกรมของส่วนประกอบต่างๆ ของเอเอสซีไอ	36
3.3	แสดงวงจรเสมือนในการเกิดสัญญาณอินเทอร์รัพท์จากเอเอสซีไอ	37
3.4	แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการรับข้อมูลจากคอมพิวเตอร์	39
3.5	แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการส่งข้อมูลไปบอร์ดแสดงผล	40
4.1	แสดงผังงานการทำงานของโปรแกรมบนคอมพิวเตอร์	46
ข-1	แสดงบิตต่าง ๆ ของ ASCI Status Register	63
ข-2	แสดงบิตต่าง ๆ ของ ASCI Control Register A	66
จ-1	แสดงโหมดการทำงานเริ่มต้น หลังจากเรียกโปรแกรม PROJECT.EXE	184
จ-2	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อกดคีย์ Ctrl H	184
จ-3	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อกดคีย์ Ctrl M	185
จ-4	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อคลิกเมาส์ที่ Button 'Save'	185
จ-5	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อคลิกเมาส์ที่ Button 'Load'	186
จ-6	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อกดคีย์ Ctrl Z	186
จ-7	แสดงโหมดข้อความต่อเนื่อง จากการกดคีย์ Ctrl P ในโหมดการทำงานเริ่มต้น	188
จ-8	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดข้อความต่อเนื่อง เมื่อกดคีย์ K	188
จ-9	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดข้อความต่อเนื่อง เมื่อกดคีย์ S หรือ L	189
จ-10	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดข้อความต่อเนื่อง เมื่อกดคีย์ G	189
จ-11	แสดงโหมดภาพเคลื่อนไหว จากการกดคีย์ Ctrl F ในโหมดการทำงานเริ่มต้น	192
จ-12	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl H	192
จ-13	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl P	193
จ-14	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl S หรือ Ctrl L	193
จ-15	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl Z	194

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ
(ต่อ)

รูปที่		หน้า
จ-16	แสดงโหมดการแสดงข้อมูลใน 4 ทิศทาง จากการกดคีย์ Ctrl V ในโหมดการทำงานเริ่มต้น	196
จ-17	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการแสดงข้อมูลใน 4 ทิศทาง เมื่อกดคีย์ Ctrl H	196
จ-18	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการแสดงข้อมูลใน 4 ทิศทาง เมื่อกดคีย์ Ctrl L	197
จ-19	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการแสดงข้อมูลใน 4 ทิศทาง เมื่อกดคีย์ U หรือ คีย์ D หรือ คีย์ R หรือ คีย์ L หรือ คีย์ Ctrl G	197
จ-20	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ หลังจากเรียกโปรแกรม EDIT_FON.EXE	199
จ-21	แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโปรแกรม EDIT_FON.EXE หลังจากกดคีย์ S หรือ L	199

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

สำหรับโลกของข่าวสารข้อมูลในปัจจุบันที่ข้อมูลเป็นสิ่งสำคัญในการดำรงชีวิต ไม่ว่าจะเป็นข้อมูลทางด้านธุรกิจการค้า การโฆษณา ดังนั้นสื่อที่แสดงข้อมูลต่าง ๆ เหล่านี้ ทั้งในแบบรูปภาพและตัวอักษรต้องสามารถดึงดูดความสนใจและสื่อสารให้บุคคลได้เข้าใจในข้อมูลข่าวสารนั้นได้

ดังนั้นกระดาน หรือแผ่นป้ายแสดงข่าวสาร (Display Board) จึงเป็นสื่อชนิดหนึ่งที่นิยมใช้กันมากในปัจจุบันซึ่งสามารถพบเห็นได้ตามแหล่งชุมชน ย่านธุรกิจ ศูนย์การค้า โรงพยาบาล และบริษัทห้างร้านต่าง ๆ ซึ่งในอดีตยังคงใช้ระบบการแสดงข้อมูลในแบบเซเว่นเซกเม้นท์ (7-Segment Display) และต่อมาจึงมีการพัฒนาเป็นแบบจุด (Dot Matrix Display) ซึ่งจะมีความละเอียดของภาพมากขึ้น

ในตอนต้น การแสดงผลแบบจุดยังคงใช้แอลอีดี หรือไดโอดปล่อยแสง (LED หรือ Light Emitting Diode) ที่สามารถเปล่งแสงออกมาเป็นสีเดียว เป็นสีแดงบ้าง สีเขียวบ้าง ซึ่งยังไม่สามารถตอบสนองการแสดงผลออกมาเป็นรูปภาพได้เพราะมีสองสีเท่านั้น ต่อมาเนื่องจากเทคโนโลยีทางด้านอิเล็คทรอนิกส์ มีการพัฒนาคิดค้นผลิตแอลอีดีที่สามารถปล่อยแสงออกมาได้ถึง 3 สีในหลอดเดียวกัน คือ สีแดง สีเขียว หรือสีส้ม ดังนั้นสำหรับแอลอีดี 1 หลอด สามารถให้สีทั้งหมด 4 สี คือ เมื่อหลอดเปล่งแสงจะเกิด 3 สี (สีแดง สีเขียว หรือสีส้ม) และเมื่อหลอดไม่เปล่งแสงก็จะเกิดสีดำ

ดังนั้นในภาคการศึกษาต่าง ๆ ที่ผ่านมา จึงนำแอลอีดีชนิดดังกล่าวนี้มาประยุกต์ใช้ในการออกแบบผลิตเป็นบอร์ดแสดงข่าวสารแทนบอร์ดแสดงข่าวสารแบบเก่าซึ่งสามารถแสดงข้อมูลได้เพียงภาพขาวดำเท่านั้น

การแสดงผลมีทั้งแบบตัวอักษรภาษาไทย ภาษาอังกฤษ รวมทั้งรูปภาพกราฟฟิกต่าง ๆ นอกจากนี้แล้วยังสามารถดัดแปลงรูปแบบการแสดงผลแบบอินเวอร์ส (Inverse) โดยใช้สีพื้น (Background) เป็นสีต่าง ๆ แล้วให้ภาพที่ต้องการเป็นสีดำ หรือดับนั้นเอง

ดังนั้นในโครงงานนี้ เราจึงนำแอลอีดีชนิดนี้มาทำเป็นบอร์ดแสดงข้อมูล (Dot Matrix Display Board) ซึ่งสามารถแสดงผลภาพ หรือตัวอักษรแบบด็อทเมตริกซ์ (Dot Matrix) โดยใช้การควบคุมด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 1.1 แสดงส่วนประกอบที่สำคัญของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยส่วนประกอบที่สำคัญทั้งหมดของโครงงานมี 3 ส่วนด้วยกัน ซึ่งสามารถแสดงด้วยบล็อคไดอะแกรมอย่างง่าย ๆ ดังรูปที่ 1.1

1. บอร์ดแสดงข้อมูล (Dot Matrix Display Board) อันได้แก่

- ส่วนแสดงผล (Display Board)

ประกอบด้วยแผงแอลอีดีจำนวนมากเรียงต่อกันเป็นแผงแบบต่อเนื่องเพื่อให้สามารถแสดงข้อมูลได้ละเอียด และถูกต้องตามต้องการ

- ส่วนตัวขับ (Driver Circuit Board)

ประกอบด้วย 2 ส่วน คือ วงจรตัวขับทางแถว (Row Driver Circuit) และวงจรตัวขับทางคอลัมน์ (Column Driver Circuit) ซึ่งแยกกันคนละวงจร โดยจะทำหน้าที่ขับกระแสไปให้แอลอีดีตามตำแหน่งแถว และคอลัมน์ที่ต้องการให้หลอดเปล่งแสง

- ส่วนควบคุม และพักข้อมูล (Control / Latch Data Circuit Board)

เนื่องจากตัวไมโครคอนโทรลเลอร์ไม่สามารถส่งข้อมูลออกแสดงผลได้พร้อมกันในเวลาเดียวกัน จึงจำเป็นต้องมีการเก็บพักข้อมูลเพื่อให้ข้อมูลสามารถแสดงผลออกมาพร้อมกัน

2. ไมโครคอนโทรลเลอร์ (Microcontroller)

สำหรับไมโครคอนโทรลเลอร์ในการควบคุม ได้เปลี่ยนจากไมโครคอนโทรลเลอร์ เบอร์ 8032 หรือ แจ๊ส 32 (JAZZ 32) เป็น Z80180 เพื่อใช้ควบคุม และทำการประมวลผลเพื่อนำข้อมูลออกแสดงผลทางบอร์ดแสดงข้อมูลได้รวดเร็ว และสะดวกต่อผู้ใช้มากขึ้น โดยได้มีการใช้สัญญาณอินเทอร์รัพท์ของไมโครคอนโทรลเลอร์ Z80180 แทนการทำงานซิสเต็มคอล (System Call) ของแจ๊ส 32 ซึ่งทำให้การทำงานในส่วนของไมโครคอนโทรลเลอร์เป็นไปโดยอัตโนมัติอย่างอิสระ

3. คอมพิวเตอร์ส่วนบุคคล (Personal Computer หรือ PC)

ได้มีการพัฒนาในส่วนของโปรแกรมบนคอมพิวเตอร์โดยสร้างบอร์ดจำลองบนหน้าจอแสดงผล (Monitor) ของคอมพิวเตอร์โดยที่เราสามารถควบคุมการติด หรือดับ และสามารถกำหนดสีที่จะปล่อยออกมาของแอลอีดี โดยใช้เมาส์ (Mouse) เป็นเครื่องมือในการกำหนดตำแหน่ง และสี แล้วส่งข้อมูลไปให้ไมโครคอนโทรลเลอร์ ซึ่งทำให้สามารถกำหนดข้อมูลไม่ว่าจะเป็นตัวอักษร รูปภาพ หรือสัญลักษณ์ได้ตามต้องการ สำหรับข้อมูลที่เป็นตัวอักษรนั้น แทนที่ผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะต้องออกแบบที่บอร์ดจำลองบนหน้าจอคอมพิวเตอร์ คณะผู้จัดทำได้พัฒนาให้ผู้ใช้สามารถกำหนดข้อมูลที่เป็นอักษรนี้ผ่านทางแป้นพิมพ์ (Keyboard) ได้เลย

นอกจากนี้ยังได้พัฒนาโปรแกรมบนคอมพิวเตอร์เพื่อให้สามารถแสดงข้อมูลข่าวสารออกทางบอร์ดได้หลายรูปแบบ ไม่ว่าจะเป็นการแสดงข้อมูลแบบภาพนิ่ง การแสดงข้อความต่อเนื่อง และการแสดงข้อมูลแบบเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
บอร์ดแสดงข้อมูล
(Dot Matrix Display Board)

2.1 หลักการออกแบบ

เนื่องจากคณะผู้จัดทำต้องการออกแบบให้บอร์ดแสดงข้อมูล ซึ่งประกอบด้วยส่วนของวงจรที่ใช้ขับกระแสให้แก่แอลอีดี ซึ่งอยู่รวมกันเป็นแผงในส่วนแสดงผล และส่วนของวงจรควบคุมและพักข้อมูลให้มีขนาดกะทัดรัดที่สุด จึงได้ออกแบบให้ส่วนประกอบทั้ง 3 ส่วนแยกออกจากกัน โดยอยู่บนบอร์ดเดียวกัน และมีการเชื่อมต่อถึงกันเพื่อส่งข้อมูล โดยใช้สายแพส่งข้อมูลซึ่งข้อมูลจากคอมพิวเตอร์จะถูกส่งผ่านออกมาทางพอร์ตอนุกรมเข้าสู่ไมโครคอนโทรลเลอร์ และตัวไมโครคอนโทรลเลอร์ก็จะทำการประมวลผลข้อมูล และควบคุมการส่งข้อมูลไปสู่ส่วนควบคุม และพักข้อมูล ซึ่งเมื่อข้อมูลถูกพัก และพร้อมที่จะส่งออกไปแสดงผลแล้ว ก็จะถูกส่งออกไปยังส่วนตัวขับ ซึ่งเชื่อมต่ออยู่กับส่วนแสดงผล เพื่อแสดงผลข้อมูลข่าวสารตามต้องการ

เราได้ออกแบบบอร์ดแยกเป็น 2 ชั้น ชั้นบนเป็นชั้นที่ติดตั้งส่วนแสดงผล โดยด้านหลังของชั้นนี้ได้ติดตั้งวงจรตัวขับทางแถวไว้ และใช้สายแพเป็นสายส่งข้อมูลระหว่างส่วนตัวขับไปยังส่วนแสดงผลด้านหน้า ส่วนชั้นที่สองหรือชั้นหลังเป็นที่ติดตั้งของส่วนควบคุมและพักข้อมูล และวงจรตัวขับทางคอลัมน์ โดยใช้สายแพเป็นสายส่งข้อมูลระหว่างแต่ละบอร์ดเช่นเดียวกัน

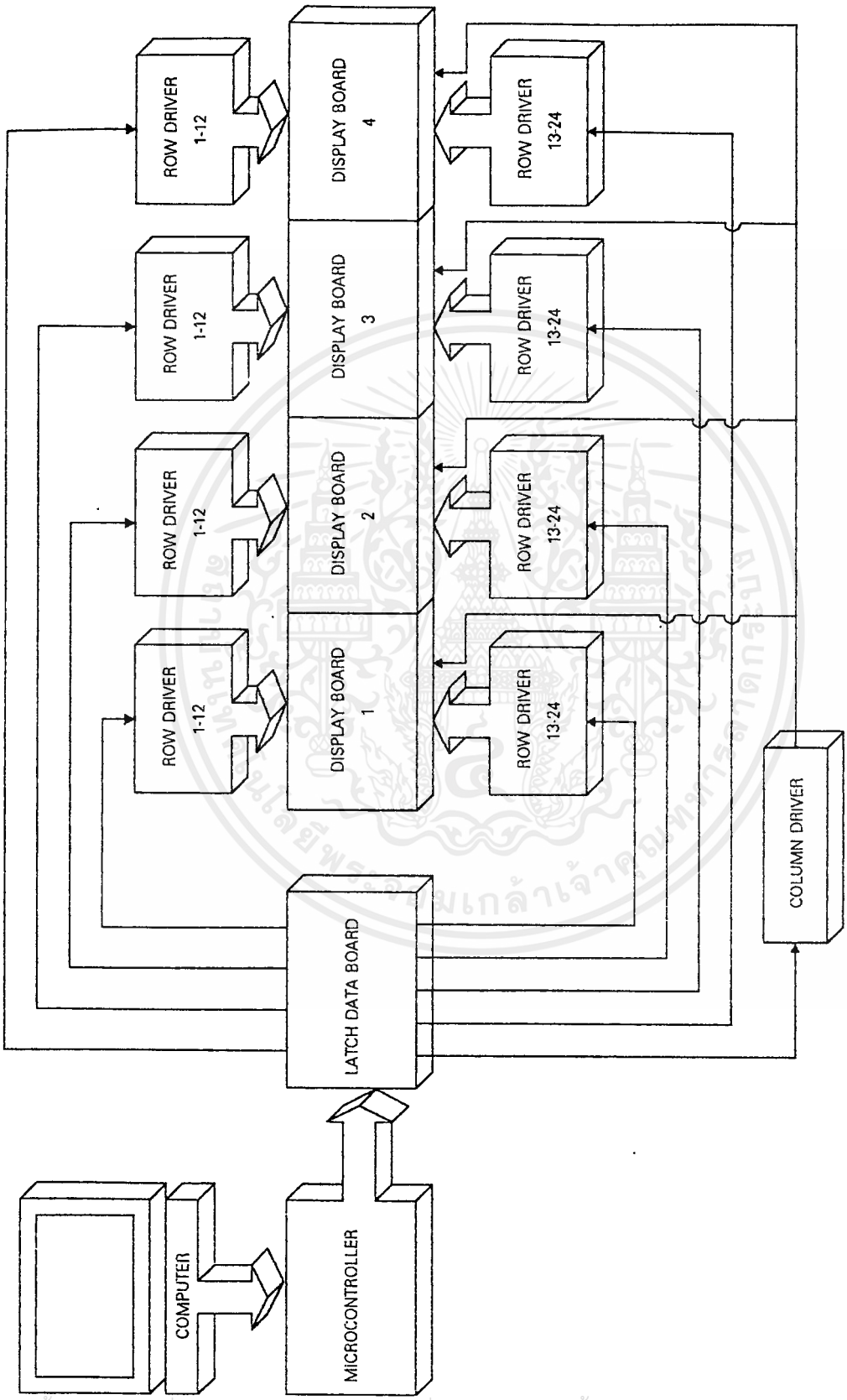
เราต้องการประหยัดเนื้อที่และพลังงาน เนื่องจากแหล่งจ่ายไฟ (Power Supply) ที่มีขนาดกระแสมาก จะมีราคาแพง ขนาดใหญ่ และสร้างได้ยาก ดังนั้นเราจึงไม่จ่ายไฟป้อนเข้าแผงแอลอีดีโดยตรงทุกดวงพร้อมกัน ซึ่งจะทำให้อายุการใช้งานของแอลอีดียาวขึ้นเพราะไม่ต้องทำงานตลอดเวลา

เราจึงออกแบบการทำงานเป็นแบบมัลติเพลกซ์ (Multiplex) โดยให้มีการทำงานทีละคอลัมน์สลับเรียงกันไปเรื่อย ๆ ด้วยความเร็วสูง เพื่อให้เห็นว่าทุกคอลัมน์ทำงานพร้อมกันหมด โดยใช้การสแกน (Scan) ทางด้านคอลัมน์ด้วยความเร็วสูง แต่ไม่มีการสแกนทางด้านแถวแต่อย่างใด ถ้าใช้แหล่งจ่ายไฟตัวเดียวจ่ายกระแสให้ทั้ง 4 ส่วนจะทำให้หลอดสว่างไม่เพียงพอ เพื่อทำให้หลอดสว่างมากยิ่งขึ้น จะต้องแยกจ่ายกระแสโดยใช้แหล่งจ่ายไฟ 2 แหล่งด้วยกัน โดยแต่ละแหล่งจะจ่ายกระแสให้กับ 2 ส่วน โดยต่อสายดินของทุกส่วนร่วมกัน ถ้าใช้วิธีนี้แต่ละส่วนจะสว่างมากขึ้น และทำให้แหล่งจ่ายไฟแต่ละแหล่งไม่ทำงานหนักเกินไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจ่ายไฟเลี้ยงขนาด 15 โวลท์ ให้กับแต่ละส่วนต้องทำแบบขนานเพื่อให้แต่ละส่วนได้รับไฟเลี้ยงเท่ากัน เนื่องจากถ้าเราจ่ายไฟเลี้ยงให้เพียงส่วนเดียว แล้วต่ออนุกรมแต่ละส่วน จะทำให้แต่ละส่วนได้รับไฟเลี้ยงไม่เท่ากันเนื่องจากมีศักดาไฟฟ้าตกคร่อมในแต่ละส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.1 แสดงบล็อกไดอะแกรมของโครงสร้างทางฮาร์ดแวร์ และการส่งผ่านข้อมูลระหว่างส่วนประกอบต่าง ๆ ของโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คุณลักษณะของส่วนประกอบต่าง ๆ ของบอร์ดแสดงข้อมูล (Dot Matrix Display Board)

2.2.1 ส่วนแสดงผล (Display Board)

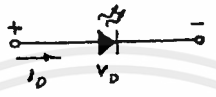
แผงแสดงผลอิเล็คทรอนิกส์ในโครงงานนี้ ส่วนแสดงผลจะประกอบด้วยแอลอีดีเป็นหัวใจสำคัญ ดังนั้นจึงต้องทำความเข้าใจเกี่ยวกับคุณสมบัติ และการทำงานของแอลอีดีให้เข้าใจเพื่อจะนำไปใช้ในการแบ่งระดับสีของแอลอีดี

คุณสมบัติของแอลอีดี หรือไดโอดปล่อยแสง (LED หรือ Light Emitting Diode)

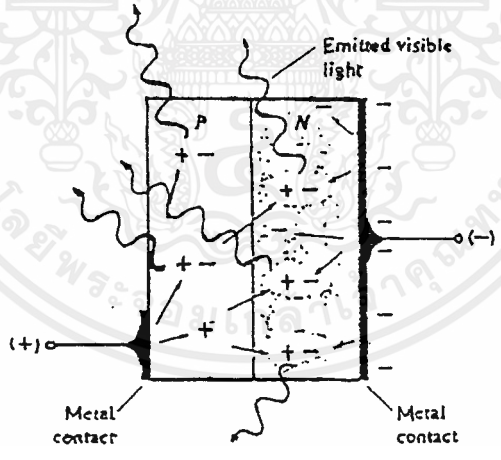
แอลอีดีเป็นไดโอดชนิดหนึ่งซึ่งสามารถปล่อยแสงได้ โดยโครงสร้างของแอลอีดีจะมีลักษณะเป็นสารกึ่งตัวนำชนิดพี (P-Type) และชนิดเอ็น (N-Type) ต่อกันเป็นพีเอ็นจังก์ชั่น (PN Junction) แอลอีดีจะปล่อยแสงออกมาได้ก็ต่อเมื่อมีการจ่ายกระแสฟอร์เวิร์ดไบอัส (Forward Bias) ให้กับมัน กระแสฟอร์เวิร์ดไบอัสนี้จะไปกระตุ้นอิเล็กตรอน (Electron) และโฮล (Hole) ข้ามพี-เอ็นจังก์ชั่นเพื่อมารวมกัน ในการรวมตัวกันนี้จะมีการปลดปล่อยพลังงานออกมาในรูปของโฟตอน (Photon) ซึ่งเป็นอนุภาคของแสง และคุณสมบัตินี้เองทำให้แอลอีดีต่างจากอุปกรณ์อย่างอื่น ๆ ที่ปลดปล่อยพลังงานออกมาในรูปของความร้อน สำหรับสารกึ่งตัวนำที่นิยมนำมาสร้างแอลอีดีจะใช้สารแกลเลี่ยมอาร์เซไนด์ฟอสไฟด์ (Gallium Arsenidephosphide : GaAsP) หรือแกลเลี่ยมฟอสไฟด์ (Gallium Phosphide : GaP) ซึ่งสารทั้งสองชนิดนี้จะใช้กระแสไฟฟ้าไม่มากในการไบอัสให้ปลดปล่อยโฟตอนออกมา การให้แสงของแอลอีดีโดยการจ่ายกระแสไฟฟ้า เรียกว่า อิเล็กโตรลูมิเนสเซนส์ (Electroluminescence) ดังรูปที่ 2.2 (ก) และ (ข) แสดงสัญลักษณ์ทางไฟฟ้าของแอลอีดี และแสดงโครงสร้าง และการเกิดอิเล็กโตรลูมิเนสเซนส์ของแอลอีดี ตามลำดับ ส่วนรูปที่ 2.3 (ก)-(ฌ) จะแสดงคุณสมบัติและกราฟแสดงคุณลักษณะของแอลอีดีของบริษัทฮิวเลทแพคการ์ดแบบไฮเอฟพีชิเอนซี (High Efficiency)

เนื่องจากเราได้แบ่งส่วนแสดงผลออกเป็น 4 ส่วน โดยแต่ละส่วนมีขนาดจำนวนจุดทางด้านแถวและด้านคอลัมน์เป็น 24x20 จุด ตามลำดับ ดังนั้นจะทำการอธิบายการออกแบบ และการทำงานเพียงส่วนเดียวเท่านั้น โดยวงจรในส่วนบอร์ดนั้นเราจะทำการเชื่อมขาของด็อทเมตริกซ์ (Dot Matrix) ที่ใช้ควบคุมด้านแถวเข้าด้วยกันในลักษณะขนาน โดยขาที่ใช้ควบคุมแถวเดียวกันจะถูกต่อเชื่อมถึงกันหมด ดังนั้นเมื่อจ่ายไฟเข้าที่ตัวใดตัวหนึ่งของแถว ก็จะทำให้หลอดแอลอีดีบนแถวนั้นติดทุกดวง และในคอลัมน์เดียวกันของแต่ละแถวจะถูกเชื่อมถึงกันทั้งหมดเพื่อลงกราวน์ที่จุดเดียวกัน โดยการต่อวงจรเป็นไปดังรูปที่ 2.4 และ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



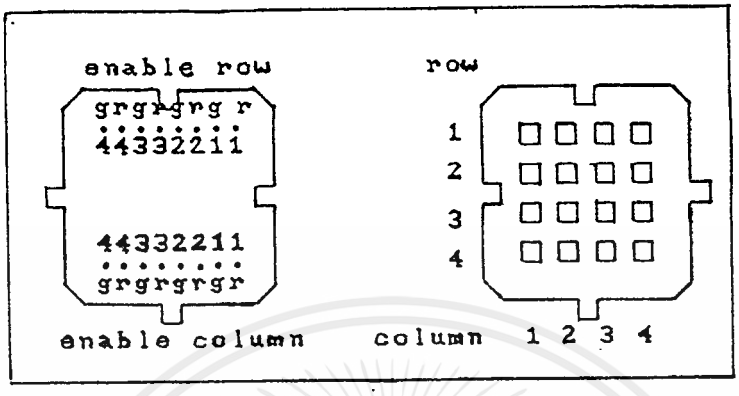
(ก)



(ข)

รูปที่ 2.2 (ก) แสดงสัญลักษณ์ทางไฟฟ้าของแอลอีดี
(ข) แสดงโครงสร้างและการเกิดอิเล็กโตร-
ลูมิเนสเซนส์ของแอลอีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) แสดงรูปร่างของแอลอีดี

Absolute Maximum Ratings at T_A = 25°C

Parameter	High Eff. Red 4160	Units
Power dissipation	120	mW
Average forward current	20[1]	mA
Peak forward current	60	mA
Operating and storage temperature range	-55°C to 100°C	
Lead soldering temperature [1.6 mm (0.063 in.) from body]	230°C for 3 seconds	

[1] Derate from 50°C at 0.2 mA/°C.

(ข) แสดงค่าพิกัดของพารามิเตอร์ต่าง ๆ ของแอลอีดี

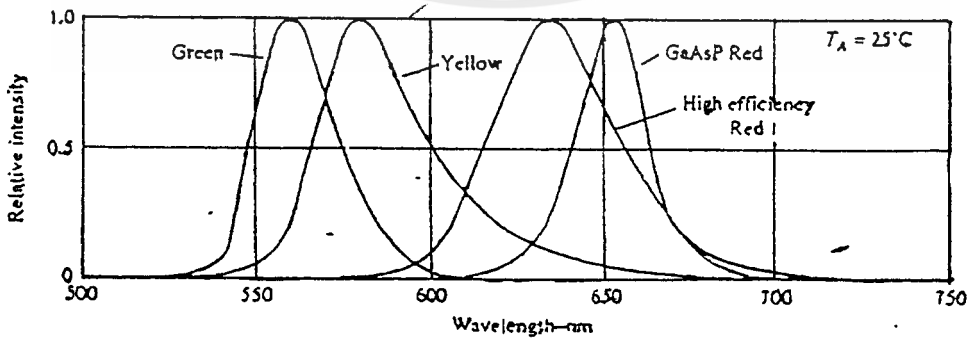
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical/Optical Characteristics at T_A = 25°C

Symbol	Description	High Eff Red 4160 Min.	Typ.	Max.	Units	Test Conditions
I_V	Axial luminous intensity	1.0	3.0		mcd	I_F = 10 mA
$2\theta_{1/2}$	Included angle between half luminious intensity points		80		deg.	Note 1
λ_{peak}	Peak wavelength		635		nm	Measurement at peak
λ_d	Dominant wavelength		628		nm	Note 2
τ_s	Speed of response		90		ns	
C	Capacitance		11		pF	V_F = 0; f = 1 Mhz
θ_{JC}	Thermal resistance		120		°C/W	Junction to cathode lead at 0.79 mm (.031 in) from body
V_F	Forward voltage		2.2	3.0	V	I_F = 10 mA
BV_R	Reverse breakdown voltage	5.0			V	I_R = 100 µA
η_1	Luminous efficacy		147		lm/W	Note 3

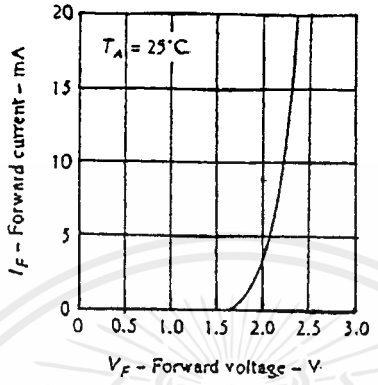
NOTES:
1. $\theta_{1/2}$ is the off-axis angle at which the luminous intensity is half the axial luminous intensity.
2. The dominant wavelength, λ_d is derived from the CIE chromaticity diagram and represents the single wavelength that defines the color of the device.
3. Radiant intensity, I_e, in watts/steradian, may be found from the equation $I_e = I_v/\eta_1$, where I_v is the luminous intensity in candelas and η_1 is the luminous efficacy in lumens/watt.

(ค) แสดงคุณสมบัติของแอลอีดีทางไฟฟ้า และทางแสง

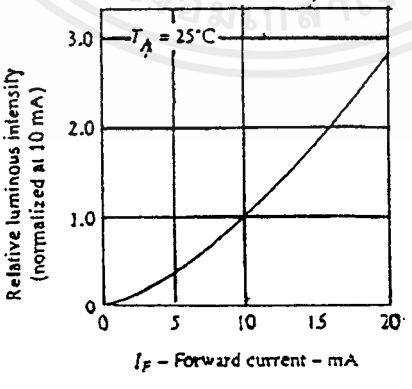


(ง) แสดงกราฟความสัมพันธ์ของความเข้มแสงต่อความยาวคลื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

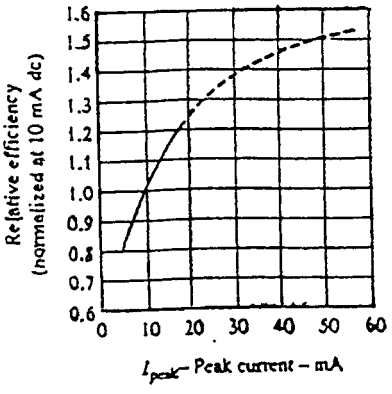


(จ) แสดงกราฟการเปลี่ยนแปลงของกระแสฟอร์เวิร์ดตามแรงดันฟอร์เวิร์ด

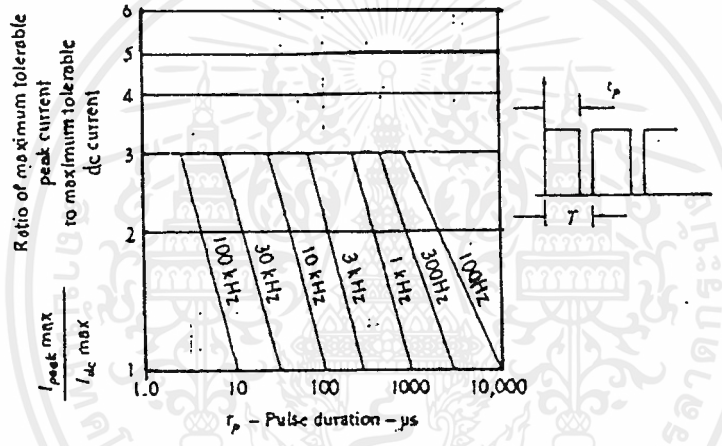


(ฉ) แสดงกราฟความสัมพันธ์ของความเข้มแสงต่อกระแสฟอร์เวิร์ด

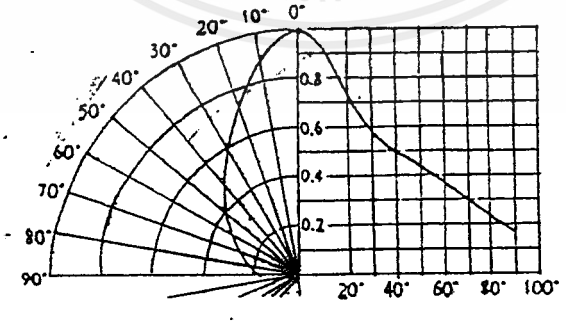
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ช) แสดงกราฟการเปลี่ยนแปลงของประสิทธิภาพตามกระแสพีค



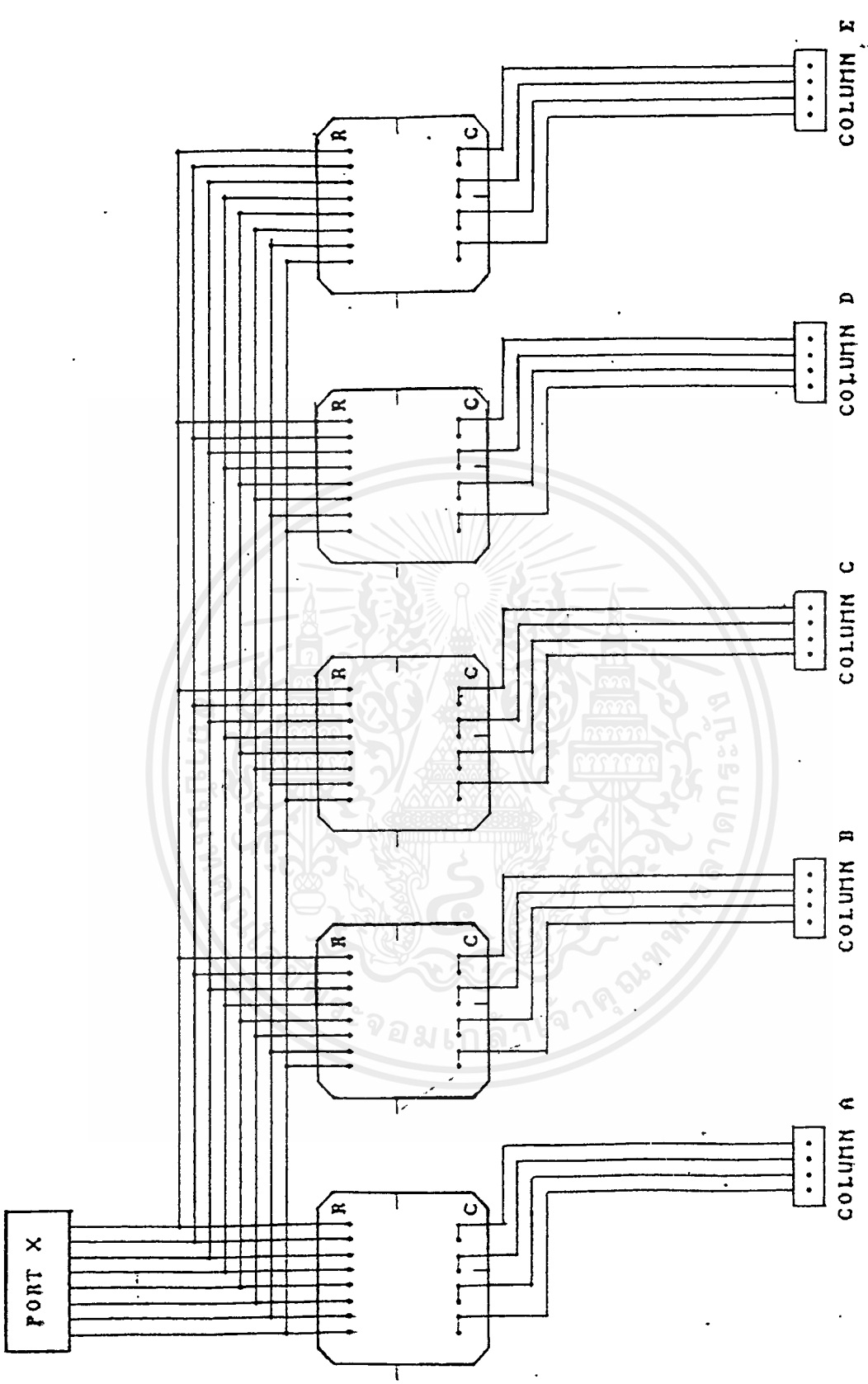
(ซ) แสดงกราฟการเปลี่ยนแปลงของกระแสพีคสูงสุดตามความกว้างของพัลซ์ (Pulse)



(ฌ) แสดงกราฟความสัมพันธ์ของการส่องแสงต่อมุมมอง

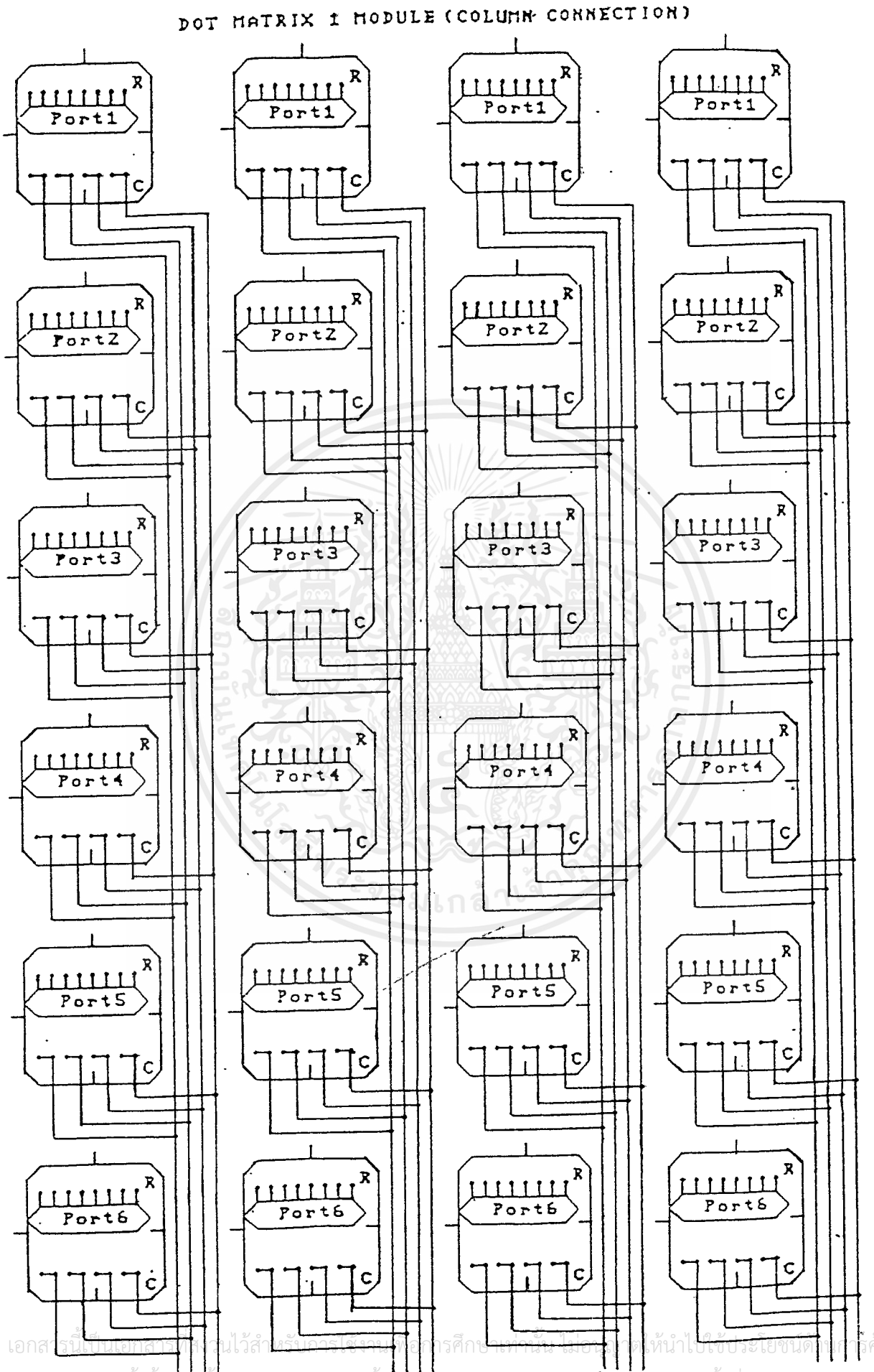
<u>รูปที่ 2.3 (ก) - (ฌ)</u> แสดงคุณสมบัติ และกราฟแสดงคุณลักษณะของแอลอีดีของ บริษัท ฮิเลทแพคการ์ต แบบไฮเอฟฟิชิเอนซี (High Efficiency)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงการเชื่อมต่อของบอร์ดแสดงผลทางด้านคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ส่วนตัวขับ (Driver Circuit Board)

เนื่องจากเราต้องจ่ายไฟให้แก่แผงแอลอีดีในแต่ละหลอดเพื่อควบคุมให้หลอดสว่างตามต้องการเพื่อแสดงข้อมูลเป็นตัวอักษร และรูปกราฟฟิกต่าง ๆ ดังนั้นเราจำเป็นต้องมีส่วนตัวขับที่ทำหน้าที่เป็นสวิตช์ปิด/เปิด โดยใช้ทรานซิสเตอร์ทำหน้าที่เป็นเกทควบคุมเกาจ่ายกระแสให้แก่แผง

ส่วนตัวขับแบ่งออกเป็น 2 วงจร ได้แก่

- วงจรตัวขับทางแถว (Row Driver Circuit)
- วงจรตัวขับทางคอลัมน์ (Column Driver Circuit)

วงจรตัวขับทางแถว (Row Driver Circuit)

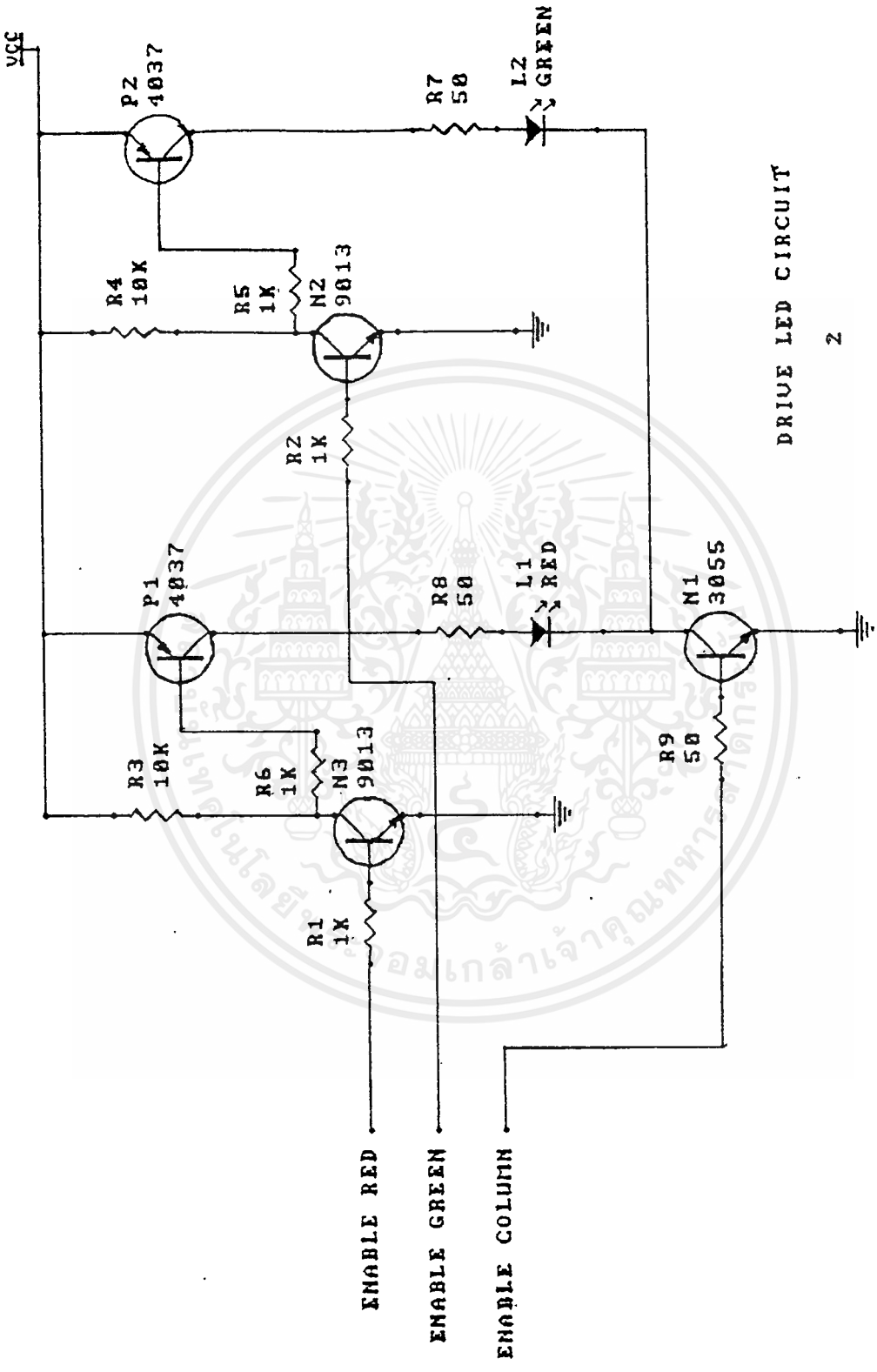
ทรานซิสเตอร์ที่ใช้ได้แก่ เบอร์ 4037 PNP และเบอร์ 9013 NPN ใช้ความต้านทานขนาด 1 กิโลโอห์ม, 10 กิโลโอห์ม และ 50 โอห์ม โดยจะแบ่งวงจรออกเป็น 2 ส่วน คือ ส่วนขับกระแสให้แก่แอลอีดีสีแดงและส่วนขับกระแสให้แก่แอลอีดีสีเขียว โดยในแต่ละส่วนมีการต่อวงจรแบบวงจรเสมือนสมมาตร (Quasi-complementary Symmetry) ซึ่งเป็นวงจรดาร์ลิงตัน (Darlington) ชนิดหนึ่ง ดังรูปที่ 2.6

โดยใช้ความต้านทานขนาด 50 โอห์ม (R7, R8) เป็นตัวจำกัดกระแสที่จะไหลเข้าสู่แอลอีดีแต่ละหลอด และใช้ความต้านทานขนาด 1 กิโลโอห์ม (R1, R2) เป็นตัวจำกัดกระแสเบสที่จะไหลเข้าไปไบอัสทรานซิสเตอร์เบอร์ 9013

นอกจากนี้ยังใช้ความต้านทานขนาด 10 กิโลโอห์ม (R3, R4) เป็นตัวจำกัดกระแสที่จะไหลเข้าสู่ทรานซิสเตอร์ เพราะถ้าไม่มี จะทำให้ทรานซิสเตอร์เบอร์ 9013 มีกระแสไหลผ่านมากเกินไป มีผลทำให้ทรานซิสเตอร์ร้อนจนเกินไปซึ่งจะทำให้ทรานซิสเตอร์เสียหายได้

เมื่อเราต้องการให้หลอดแอลอีดีสีใด และตำแหน่งใดสว่างก็จะต้องขับกระแสไปยังหลอดแอลอีดีนั้น ๆ โดยการส่งลอจิก (Logic) 1 ไปเปิดทาง (Enable) ที่ขาเบสของทรานซิสเตอร์ N2 และ/หรือ N3 โดยถ้าส่งลอจิก 1 ไปเปิดทางที่ขาเบสของทรานซิสเตอร์ N2 จะทำให้กระแสถูกขับไปยังแอลอีดี L2 ซึ่งจะให้แสงสีเขียวออกมา ในขณะที่ถ้าส่งลอจิก 1 ไปเปิดทางที่ขาเบสของ N3 จะทำให้กระแสถูกขับไปยังแอลอีดี L1 ซึ่งจะให้แสงสีแดงสว่างออกมา และถ้าส่งลอจิก 1 ไปเปิดทางที่ขาเบสของทรานซิสเตอร์ทั้งสองตัวพร้อมกัน ก็จะได้แสงสีส้มสว่างออกมา คือเป็นสีผสมระหว่างสีแดง และสีเขียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงวงวจรทางไฟฟ้าของตัวขับทางด้านแถวและคอลัมน์ต่อ 1 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราใช้วงจรตัวขับทางแถวนี้เพื่อควบคุมกระแสที่จะจ่ายเข้าที่ขาควบคุมด้านแถว (แถวบน) คือตั้งแต่ RR1-RR4 และ RG1-RG4 ถ้าในส่วนของวงจรนี้ประกอบไปด้วยทรานซิสเตอร์เพียงสองตัวเท่านั้น คือใช้ขับกระแสให้แก่หลอดแอลอีดีแต่ละตัว จะทำให้มีประสิทธิภาพไม่ดีเพราะไม่สามารถคัทออฟ (Cut Off) ทรานซิสเตอร์ได้ จึงต้องมีการพัฒนาวงจรขึ้นใหม่ดังกล่าว

วงจรตัวขับทางคอลัมน์ (Column Driver Circuit)

ทรานซิสเตอร์ที่ใช้ ได้แก่ เบอร์ 3055 NPN และใช้ความต้านทานขนาด 50 โอห์ม เพื่อควบคุมกระแสที่ขาเบสของทรานซิสเตอร์

โดยทรานซิสเตอร์นี้จะควบคุมกระแสที่จะไหลลงกราวน์โดยจะต่อออกจากขาควบคุมด้านคอลัมน์ (แถวล่าง) คือต่อกับขา CR1-CR4 และ CG1-CG4 ซึ่งโดยปกติเราจะเชื่อมต่อขา CR และ CG ที่ควบคุมแต่ละคอลัมน์เข้าด้วยกันอยู่แล้ว

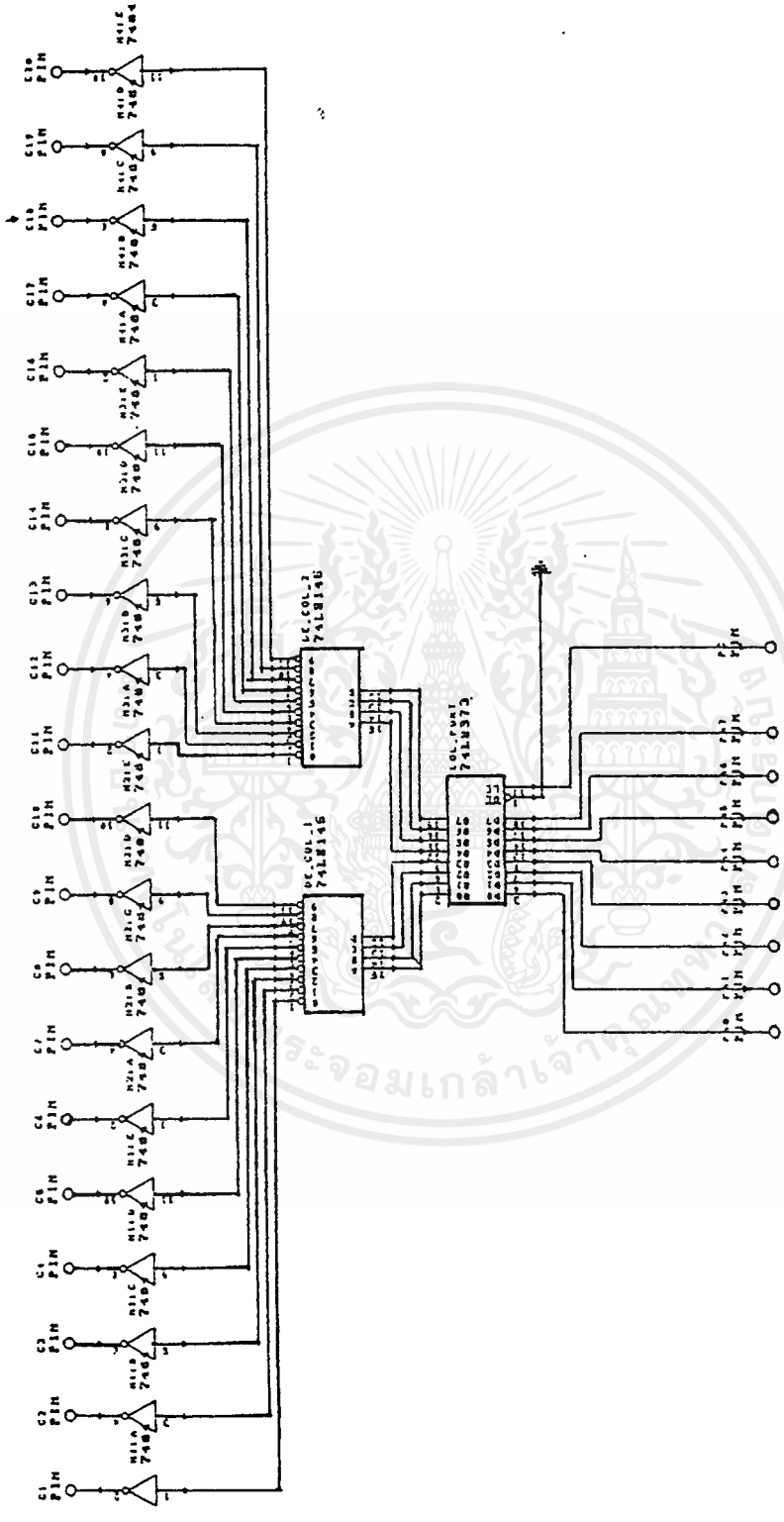
การควบคุมจะสามารถทำได้โดยการส่งลอจิก 1 ไปเปิดทางที่ขาเบสของทรานซิสเตอร์ เนื่องจากทรานซิสเตอร์ เบอร์ 4037 เป็นทรานซิสเตอร์ชนิด NPN

วงจรตัวขับทางคอลัมน์นี้จะรวมอยู่บนบอร์ดเพียงบอร์ดเดียว ใช้ควบคุมคอลัมน์ทั้งหมดทั้ง 80 คอลัมน์ โดยใช้ทรานซิสเตอร์ เบอร์ 3055 ทั้งหมดเพียง 20 ตัวเท่านั้น โดยใช้การต่อแบบขนาน ซึ่งคอลัมน์เดียวกันของแต่ละชุดจะสแกนไปพร้อม ๆ กัน ดังนั้นเราจึงใช้วงจรตัวขับทางคอลัมน์เพียงชุดเดียวก็พอ

การเชื่อมต่อระหว่างส่วนวงจรตัวขับ กับส่วนแสดงข้อมูลทั้งหมด เราได้ใช้สายแพ และตัวเชื่อม (Connector) เป็นตัวเชื่อมโยงส่งสัญญาณไปเปิดเกททรานซิสเตอร์ในส่วนตัวขับกระแสทั้งหมด เพื่อให้เกิดความยืดหยุ่น และสะดวกต่อการแยกส่วน ทั้งยังสามารถทำการตรวจสอบข้อผิดพลาดต่าง ๆ ที่เกิดขึ้นได้ง่ายอีกด้วย

รายละเอียดการต่อวงจรในส่วนของวงจรตัวขับทางคอลัมน์ แสดงไว้ดังรูปที่ 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงวงจรและการเชื่อมต่อของการทำงานทางด้านคอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

036964

2.2.3 <u>ส่วนควบคุม และพักข้อมูล</u> (Control / Latch Data Circuit Board)

เนื่องจากส่วนไมโครโปรเซสเซอร์ในไมโครคอมพิวเตอร์ส่งข้อมูลแบบอนุกรมมาเก็บไว้ที่หน่วยความจำของไมโครคอนโทรลเลอร์ เมื่อทำการเรียก (Run) โปรแกรมที่เก็บไว้มาใช้ ไมโครคอนโทรลเลอร์ก็จะส่งข้อมูลออกมาเข้าสู่ส่วนควบคุม และพักข้อมูล ซึ่งจะทำหน้าที่รับข้อมูลที่ส่งมา แล้วทำการพัก (Latch) ข้อมูลเอาไว้ให้ข้อมูลที่ถูกส่งเข้ามาครบทุกแถวก่อน

เนื่องจากข้อมูลที่ไมโครคอนโทรลเลอร์ส่งมานั้นส่งมาได้เพียงครั้งละ 8 บิตเท่านั้น โดยจะมีสัญญาณควบคุมการพักข้อมูลส่งออกมาจากตัวไมโครคอนโทรลเลอร์เพื่อควบคุมการพักข้อมูลให้ถูกต้อง เมื่อข้อมูลถูกส่งมาครบทุกแถวแล้ว ตัวไมโครคอนโทรลเลอร์ก็จะส่งสัญญาณออกมาควบคุมให้ไอซีพักข้อมูล (IC Latch หรือ Integrated Circuit Latch) ทุกตัว ส่งข้อมูลออกมาที่ขาเอาท์พุท (Output) พร้อมกัน เพื่อส่งเข้าไปยังส่วนแสดงผลเพื่อทำให้หลอดสว่างตามข้อมูลที่ส่งมา แต่หลอดจะยังไม่สว่าง โดยจะสว่างก็ต่อเมื่อไมโครคอนโทรลเลอร์ได้ส่งสัญญาณมาควบคุมให้วงจรตัวขับทางคอลัมน์ทำงานด้วย โดยส่งสัญญาณมาควบคุมเปิดเกทให้กระแสสามารถลงกราวน์ได้ จึงจะทำให้แอลอีดีสว่างตามต้องการได้ การส่งสัญญาณมาที่คอลัมน์ใดก็จะทำให้แอลอีดีในคอลัมน์นั้น ๆ สว่างตามข้อมูลที่ส่งมาในแต่ละแถว

องค์ประกอบที่สำคัญของวงจรควบคุมและพักข้อมูล ได้แก่

1. ไอซีพักข้อมูล (IC Latch)

ใช้ไอซีเบอร์ 74LS373 (เป็นแบบ Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops)

2. ดีโค้ดเดอร์ควบคุมการพักข้อมูล (Latch Decoder)

ใช้ไอซีเบอร์ 74LS138 (เป็นแบบ 3-Line-to-8-Line Decoder) ซึ่งจะมีสัญญาณเข้ามาควบคุม 3 ขา แล้วมีสัญญาณออกทั้งหมด 8 ขา โดยเราจะนำไปควบคุมการทำงานของไอซีพักข้อมูลทางด้านแถวทั้ง 6 ตัว แล้วที่เหลืออีก 2 ขาจะนำไปควบคุมการทำงานของไอซีพักข้อมูลทางด้านคอลัมน์

3. ดีโค้ดเดอร์ควบคุมส่วนตัวขับทางคอลัมน์ (Column Decoder)

ใช้ไอซีเบอร์ 74LS145 (เป็นแบบ BCD-to-Decimal Decoder/Driver with Open-Collector Output) ซึ่งจะมีสัญญาณเข้าควบคุม 4 ขา และมีสัญญาณออกทั้งหมด 10 ขา เราจะนำสัญญาณออกนี้ไปควบคุมการทำงานของวงจรตัวขับทางคอลัมน์โดยใช้ไอซีชนิดนี้ 2 ตัวเพื่อควบคุมทั้ง 20 คอลัมน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. อินเวอร์เตอร์ (Inverter)

ใช้ไอซีเบอร์ 74LS04 (เป็นแบบ Hex Inverter) เพื่อแปลงลอจิก ซึ่งจำเป็นในการควบคุมการทำงานในส่วนต่าง ๆ ซึ่งแต่ละองค์ประกอบมีข้อมูล และรายละเอียดดังจะกล่าวต่อไป

5. ดีโค้ดเดอร์ควบคุมส่วนแสดงผล (Board Decoder)

ใช้ไอซีเบอร์ 74LS139 (เป็น Dual 1-of-4 Decoder / Demultiplexer) ซึ่งมีสัญญาณเข้าทั้งหมด 2 ขา และมีสัญญาณออกทั้งหมด 4 ขา โดยสัญญาณที่นำมาเข้าจะนำมาจาก PB3 และ PB4 เพื่อถอดรหัสว่า จะให้บอร์ดใดรับข้อมูลไปพักเอาไว้ โดยสัญญาณเอาท์พุทที่ออกมาจะนำไปเปิดทางให้ไอซี 74LS138 ทำงาน

เมื่อไมโครโปรเซสเซอร์บนไมโครคอนโทรลเลอร์ทำการประมวลผลข้อมูลเรียบร้อยแล้ว ก็จะส่งสัญญาณออกทางพอร์ต โดยเป็นข้อมูลขนาด 8 บิต (คือสัญญาณข้อมูลที่จะส่งไปยังวงจรตัวขับทางแถวเพื่อควบคุมการเปิด/ปิดของทรานซิสเตอร์เกท) ข้อมูลก็จะถูกส่งมายังส่วนวงจรควบคุมและพักข้อมูล โดยถูกส่งไปที่ขาอินพุทของไอซีพักข้อมูลทั้ง 7 ตัว เพราะขาอินพุทของไอซีพักข้อมูลทุกตัวถูกต่อขนานกันเรียงตามลำดับตามขา หลังจากนั้นไมโครโปรเซสเซอร์ก็จะส่งสัญญาณออกมาที่ดีโค้ดเดอร์ควบคุมการพักข้อมูล จำนวน 3 บิต เพื่อถอดรหัสว่า ข้อมูลที่ส่งออกมาก่อนหน้านี้เป็นสัญญาณข้อมูลของแถวใด หรือต้องการให้พอร์ตใดรับสัญญาณข้อมูลเข้าไปพักเอาไว้

เมื่อข้อมูลถูกพักเอาไว้เรียบร้อยแล้ว ไมโครโปรเซสเซอร์ก็จะเริ่มส่งสัญญาณข้อมูล 8 บิต ถัดไปเพื่อเป็นข้อมูลการติด/ดับของแผงแอลอีดีในส่วนถัดมา (ตามแนวตั้ง) ทั้ง 4 แถวต่อไป และก็จะส่งสัญญาณออกมาให้พักข้อมูลนั้นไว้เช่นเดียวกับที่กล่าวมาแล้วจนกระทั่งครบทั้ง 6 พอร์ต ซึ่งสัญญาณข้อมูลของทั้ง 6 พอร์ตนี้เป็นสัญญาณข้อมูลที่ใช้ควบคุมการเปิด/ปิดทรานซิสเตอร์เกทในส่วนของวงจรตัวขับทางแถว เพื่อควบคุมการติด/ดับของแอลอีดีในแต่ละแถวอีกทีหนึ่ง

เมื่อสัญญาณข้อมูลของทั้ง 24 แถวถูกเก็บไว้เรียบร้อย ไมโครโปรเซสเซอร์ก็จะส่งสัญญาณอีก 8 บิต ซึ่งเป็นสัญญาณข้อมูลที่ใช้ควบคุมการสแกนทางคอลัมน์ โดยเมื่อพอร์ตพักข้อมูลเก็บข้อมูลไว้เรียบร้อยแล้ว ไมโครโปรเซสเซอร์ก็จะส่งสัญญาณมาเปิดทางที่ขา Output Enable ของพอร์ตพักข้อมูลทุกตัวพร้อมกัน โดยใช้การต่อขนานขา Output Enable ของทุกตัวเข้าด้วยกัน ทำให้พอร์ตพักข้อมูลทุกตัวส่งสัญญาณออกมาทางขาเอาท์พุทพร้อมกัน เพื่อไปควบคุมส่วนวงจรตัวขับทางแถวให้แสดงผลออกทางบอร์ดพร้อมกันทุกแถว และสัญญาณข้อมูลที่ออกจากพอร์ตพักข้อมูลพอร์ตที่ 7 (ซึ่งเป็นพอร์ตควบคุมการสแกนทางคอลัมน์) ก็จะถูกส่งไปยังดีโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมส่วนตัวขับทางคอลัมน์ (Column Decoder) ซึ่งมีอยู่ทั้งหมด 2 ตัว ๆ ละ 4 บิต เพื่อไปทำการถอดรหัสโดยอินพุทจะเป็นสัญญาณข้อมูลขนาด 4 บิต และสัญญาณข้อมูลเอาท์พุทมีขนาด 10 บิต โดยจะส่งสัญญาณออกมาเป็น 0 หรือเป็น Low บนขาเอาท์พุทที่มีค่าเท่ากับค่าสัญญาณข้อมูลที่อินพุท

เนื่องจากเราต้องใช้สัญญาณข้อมูล 1 หรือ High เพื่อไปเปิดทางที่ขาของทรานซิสเตอร์เบอร์ 3055 ที่อยู่ในส่วนของวงจรตัวขับทางคอลัมน์ แต่สัญญาณข้อมูลที่ได้จากตัวดีโค้ดเดอร์ที่ควบคุมตัวขับทางคอลัมน์ (Column Decoder) กลับออกมาเป็น 0 ดังนั้นเราจึงต้องใส่อินเวอร์เตอร์ (Inverter) เพื่อช่วยกลับสัญญาณข้อมูลจาก 0 ไปเป็น 1 แล้วจึงนำสัญญาณข้อมูลนี้ไปจ่ายให้แก่วงจรตัวขับทางคอลัมน์ เพื่อทำการสแกนคอลัมน์ต่อไป

การสแกนคอลัมน์จะมีสัญญาณข้อมูล 1 หรือ High ออกมาเพียงขาเดียวเท่านั้น ไล่จากตัวแรกไปเรื่อย ๆ จนกระทั่งถึงคอลัมน์ที่ 20 แล้วจึงวนกลับมาที่ตัวแรกเหมือนเดิม โดยจะวนไปเรื่อย ๆ และการสแกนจะเร็ว หรือช้า ขึ้นอยู่กับขีดความสามารถทางฮาร์ดแวร์ของตัวไมโครโปรเซสเซอร์บนไมโครคอนโทรลเลอร์ และขึ้นอยู่กับโปรแกรมซอฟท์แวร์ที่เขียนขึ้นด้วย

ความเร็ว หรือความถี่ในการสแกนคอลัมน์ต้องพอเหมาะ จึงจะทำให้แอลอีดีปล่อยแสงออกมาเต็มที่ เพราะถ้าสแกนเร็วเกินไป หรือความถี่สูงจนเกินไปจะทำให้แอลอีดีปล่อยแสงออกมาน้อย หลอดจะติดแบบหรี่ ๆ ทำให้มองไม่ชัด แต่ถ้าสแกนด้วยความถี่ต่ำจนเกินไป หรือสแกนช้าจนเกินไป จะทำให้แอลอีดีมีการติด/ดับช้าไปด้วย ทำให้สามารถสังเกตเห็นการสแกนที่เกิดขึ้นได้ นั่นคือจะเห็นตัวอักษร หรือภาพเกิดการสั่นได้

นอกจากนี้ถ้าเราสามารถรักษาดิวตี้ไซเคิล (Duty Cycle) ของคาบเวลาการติด และดับให้มีค่าคงที่ คือให้มีสัดส่วนของการติดต่อการดับเป็น 1 : 19 ไม่ว่าความถี่จะมากเท่าใดก็ตาม ความสว่างของแสงที่ออกมาจากแอลอีดีก็จะคงที่ด้วย เพราะเมื่อความถี่มากขึ้น จำนวนครั้งในการติดของแอลอีดีในหนึ่งหน่วยเวลาก็จะมากขึ้นตามไปด้วย

ตามความเป็นจริงแล้ว แอลอีดีไม่ได้ปล่อยแสงต่อเนื่องตลอดเวลา แต่จะติด/ดับด้วยความถี่ที่พอเหมาะ (ประมาณ 40-50 Hz) จนทำให้เรามองเห็นเหมือนกับว่า แอลอีดีติดตลอดเวลา แต่ถ้าเราจ่ายไฟให้แอลอีดีติดตลอดเวลา อาจต้องใช้แหล่งจ่ายไฟที่จ่ายกระแสสูงมาก และต้องทนกำลังไฟฟ้าได้สูงมากด้วย ซึ่งจะมีราคาแพง นอกจากนี้การที่หลอดติดตลอดเวลามีโอกาสทำให้หลอดขาดได้สูง เพราะจากการทดลองพบว่า แอลอีดีที่นำมาใช้นี้สามารถทนความต่างศักย์ได้เพียงประมาณ 5-9 โวลท์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ในการทำงานครั้งนี้ คณะผู้จัดทำได้จ่ายไฟจากแหล่งจ่ายไฟด้วยขนาดศักดาไฟฟ้าเท่ากับ 15 โวลท์ ซึ่งจะเห็นได้ว่ามีขนาดเกินความสามารถที่จะทนได้ของแอลอีดีที่นำมาใช้ ดังนั้นจึงได้ต่อความต้านทานขนาด 50 โอห์ม เข้าไปเพื่อจำกัดกระแสดังที่กล่าวไว้แล้วในตอนต้น จึงทำให้แอลอีดีปลอดภัยจากการขาด และชำรุด

สำหรับวงจรต่าง ๆ ของส่วนควบคุม และพักข้อมูลจะแสดงไว้ดังรูปที่ 2.8-2.13 ซึ่งจะเห็นได้ว่า ส่วนบอร์ดควบคุม (Control Board) ทำหน้าที่รับข้อมูลจากไมโครคอนโทรลเลอร์มาทำการพักข้อมูลไว้ที่ไอซีพักข้อมูล เบอร์ 74LS373 โดยในแต่ละบอร์ดจะต้องมีไอซีพักข้อมูลจำนวน 6 ตัว โดยที่ไอซีของพอร์ตที่ 1 ถึงพอร์ตที่ 6 ทำงานกับบอร์ดที่ 1 ส่วนไอซีของพอร์ตที่ 7 ถึงพอร์ตที่ 24 ก็จะทำงานกับบอร์ดที่ 2, 3 และ 4 ตามลำดับ เพราะแต่ละตัวทำหน้าที่พักข้อมูลไว้ได้ 8 บิต ซึ่งเท่ากับว่าสามารถพักข้อมูลที่จะสั่งให้แอลอีดีติดเป็นสีใด หรือให้ดับไว้ได้ทั้งสิ้น 4 แถวต่อไอซีพักข้อมูล 1 ตัว แต่บอร์ดแสดงผลมีความกว้างทั้งหมด 24 แถว ดังนั้นจึงต้องใช้ไอซีพักข้อมูล เบอร์ 74LS373 ทั้งสิ้น 6 ตัวต่อการพักข้อมูล 1 บอร์ด

ดังนั้นเมื่อบอร์ดแสดงข้อมูลมีทั้งหมด 4 บอร์ดย่อย จึงต้องมีชุดพักข้อมูลทั้งหมด 4 ชุด นั่นคือต้องมีพอร์ตพักข้อมูล (Port Latch) 24 พอร์ต โดยมีการส่งข้อมูลเข้ามาควบคุมลำดับการพักข้อมูลให้ข้อมูลไปพักไว้ตามพอร์ตพักข้อมูลต่าง ๆ ให้ถูกต้องสอดคล้องกับแถวที่ต้องการ โดยใช้ไอซี เบอร์ 74LS138 เป็นตัวถอดรหัสข้อมูลที่ส่งเข้ามา

โดยการรับข้อมูลเพื่อถอดรหัสนั้นได้รับมาจากพอร์ต B (PB0-PB2) ต่อเข้าที่ขา 1, 2 และ 3 (ขา A, B และ C) ของไอซี เบอร์ 74LS138 แบบขนานโดยต่อกับไอซี เบอร์ 74LS138 ทั้งหมด 4 ตัว เพราะมีพอร์ตพักข้อมูลทั้งหมด 4 ชุด โดยที่ขา Enable Input หรือขาที่ 4 และ 5 (G2A, G2B) จะถูกทำให้ทำงานโดยใช้ไอซี เบอร์ 74LS139 เป็นดีโค้ดเดอร์เลือกว่าจะส่งข้อมูลเข้าไปพักที่บอร์ดใดใน 4 บอร์ดย่อย หรือพักข้อมูลในพอร์ตพักข้อมูลชุดใดนั่นเอง โดยจะเลือกว่า จะให้ไอซี เบอร์ 74LS138 ตัวใดทำงานก่อน หรือหลัง โดยรับข้อมูลเข้าที่ขา 2 และ 3 (หรือขา A และ B) แล้วส่งสัญญาณออกที่ขา Y0 ถึง Y3 หลังจากนั้นต้องนำสัญญาณไปคอมพลีเมนท์ (Complement) (คือการแปลงค่าจาก 1 เป็น 0 หรือจาก 0 เป็น 1) โดยใช้อินเวอร์เตอร์ (Inverter) เบอร์ 7404 เพราะเอาท์พุทของไอซี เบอร์ 74LS138 จะทำงานที่ค่าสัญญาณเท่ากับ 0

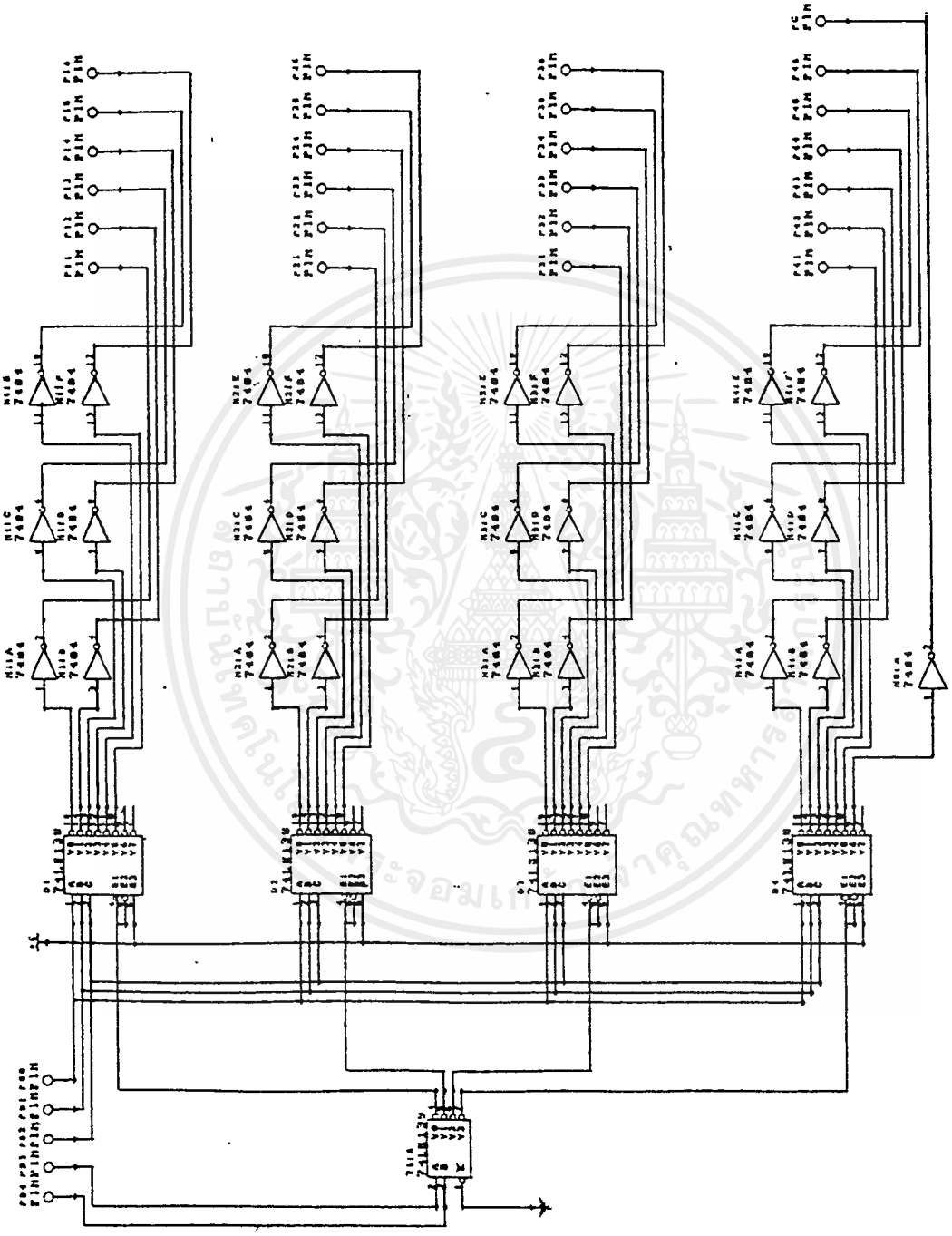
เมื่อไอซี เบอร์ 74LS138 (D1 ถึง D4) ตัวใดถูกเลือกให้ทำงาน ข้อมูลที่ถูกต่อที่ขาที่ 1, 2 และ 3 (หรือ A, B และ C) ก็จะถูกรับเข้าไปถอดรหัส แล้วส่งสัญญาณออกทางขา Y0 ถึง Y5 ไปเปิดทางที่ขา Latch Enable (ขา LE หรือขาที่ 11) ให้พอร์ตพักข้อมูลแต่ละตัวรับข้อมูลที่อยู่ที่ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุท (ขา D0 ถึง D7) ไปพักไว้ตามลำดับจนครบทุกพอร์ตของบอร์ดแรก ต่อจากนั้นก็จะทำงานเช่นเดียวกันนี้กับบอร์ดที่ 2 บอร์ดที่ 3 และบอร์ดที่ 4

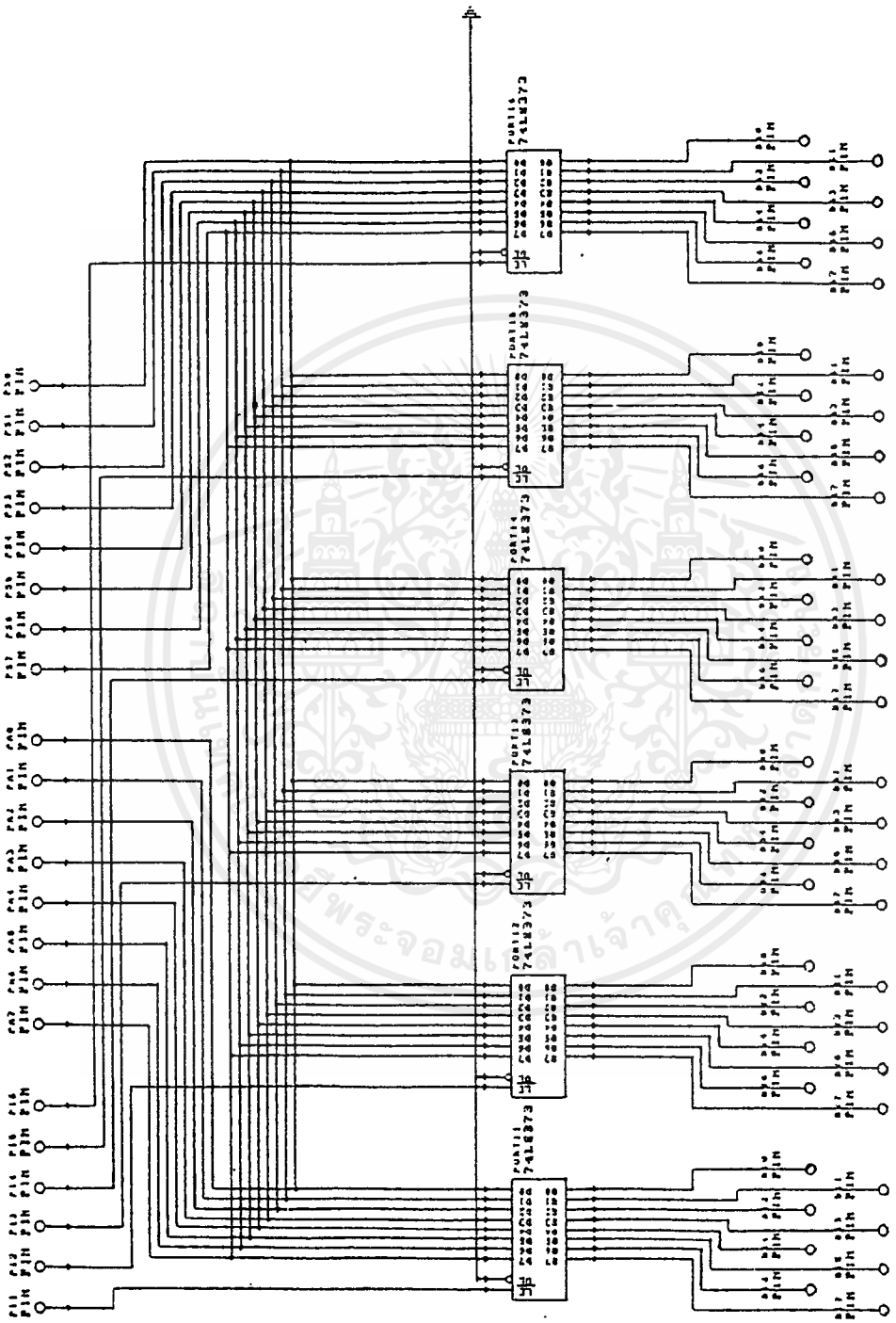
เมื่อส่งข้อมูลไปพักที่พอร์ตพักข้อมูลครบทั้ง 24 พอร์ตแล้ว ขา Y6 ของไอซี เบอร์ 74LS138 ก็จะส่งสัญญาณไปเปิดทางให้ไอซี เบอร์ 74LS373 COLPORT (Column Port) ทำการรับข้อมูลเข้าไปอีกชุด ซึ่งเป็นข้อมูลที่สั่งให้คอลัมน์ใดทำงาน ไอซีที่คอลัมน์พอร์ตจะมีขาเอาท์พุท คือ ขา Q0 ถึง Q3 ต่อกับไอซี เบอร์ 74LS145 DEC-COL-1 และเอาท์พุทที่ขา Q4 ถึง Q7 DEC-COL-2 ซึ่งเป็น ดีโค้ดเดอร์แบบเข้า 4 ออก 10 ไปทำการถอดรหัส แล้วส่งข้อมูลผ่านอินเวอร์เตอร์ (Inverter) เบอร์ 7404 เพื่อไปคอมพลีเมนท์ (Complement) สัญญาณ เนื่องจากเอาท์พุทของไอซี เบอร์ 74LS145 จะทำงานที่ค่าสัญญาณเป็น 0 สัญญาณที่ได้จะนำไปใช้ควบคุมคอลัมน์ทั้ง 20 คอลัมน์ของแต่ละบอร์ดให้บอร์ดสว่างตามคอลัมน์ที่ต้องการ โดยนำสัญญาณนี้ไปขับทรานซิสเตอร์ที่วงจรตัวขับทางคอลัมน์ให้กระแสสามารถลงกราวน์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



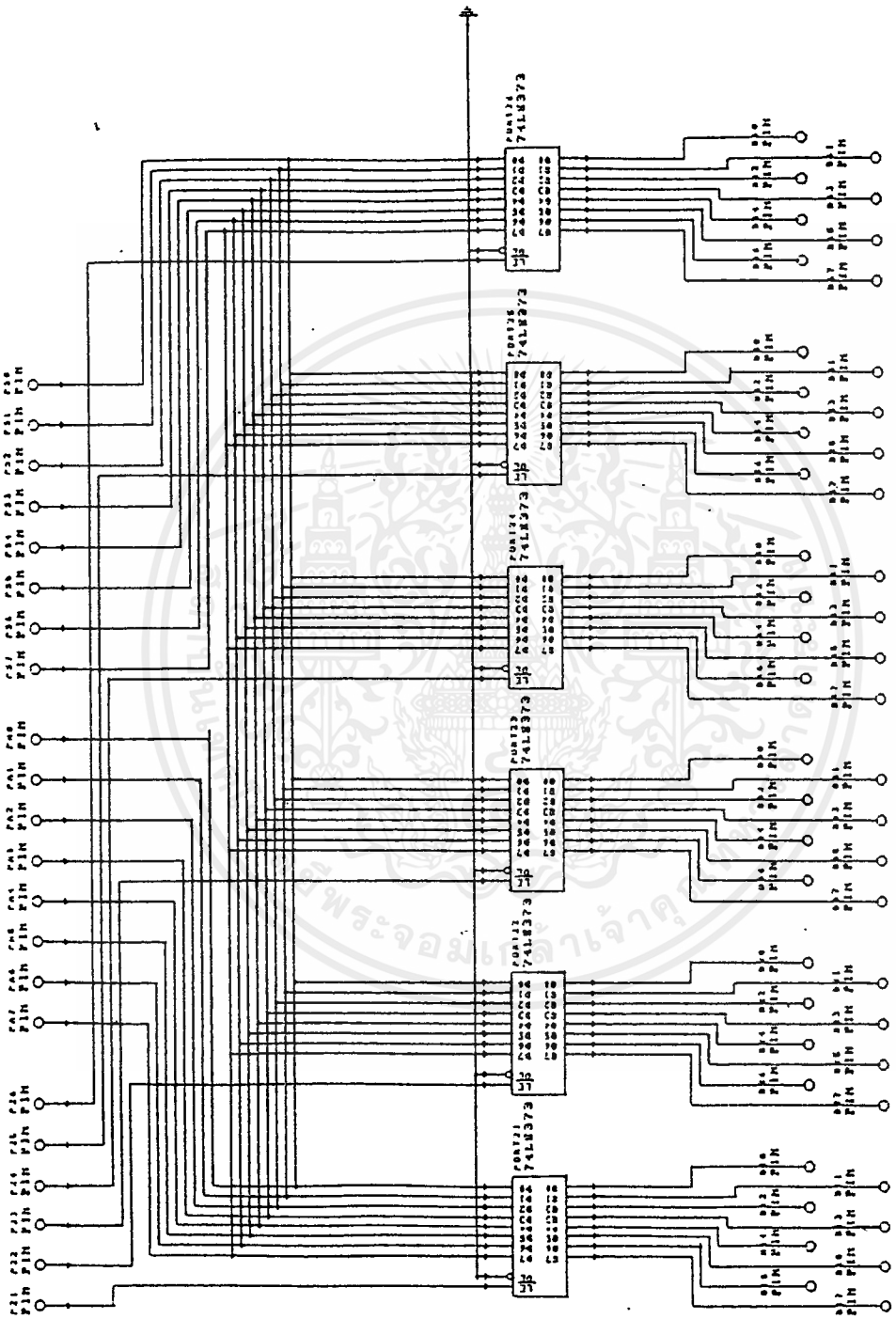
รูปที่ 2.8 แสดงวงจร Decode Port Latch ทั้ง 24 ตัว โดยใช้ 74LS139 และ 74LS138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



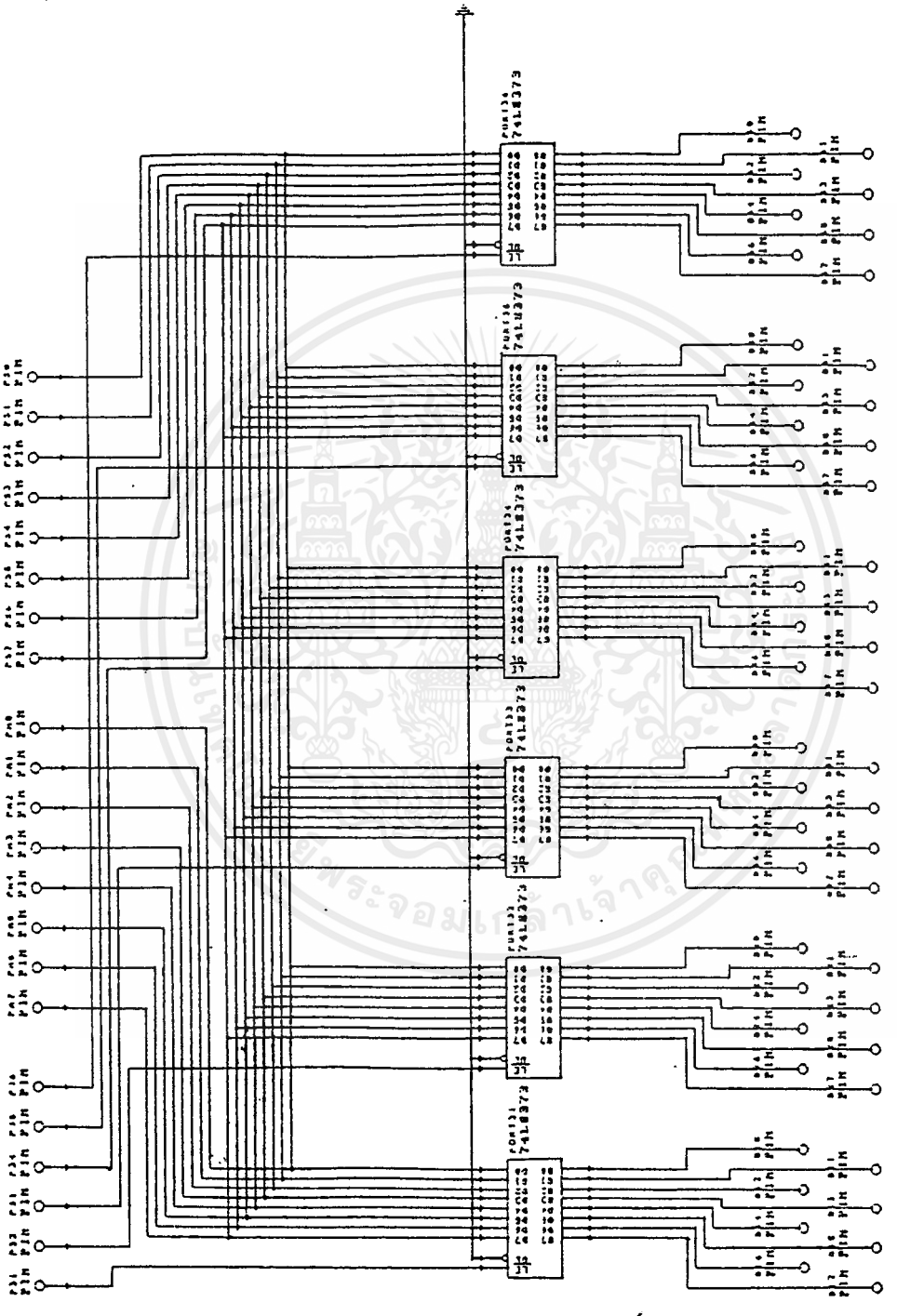
รูปที่ 2.9 แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



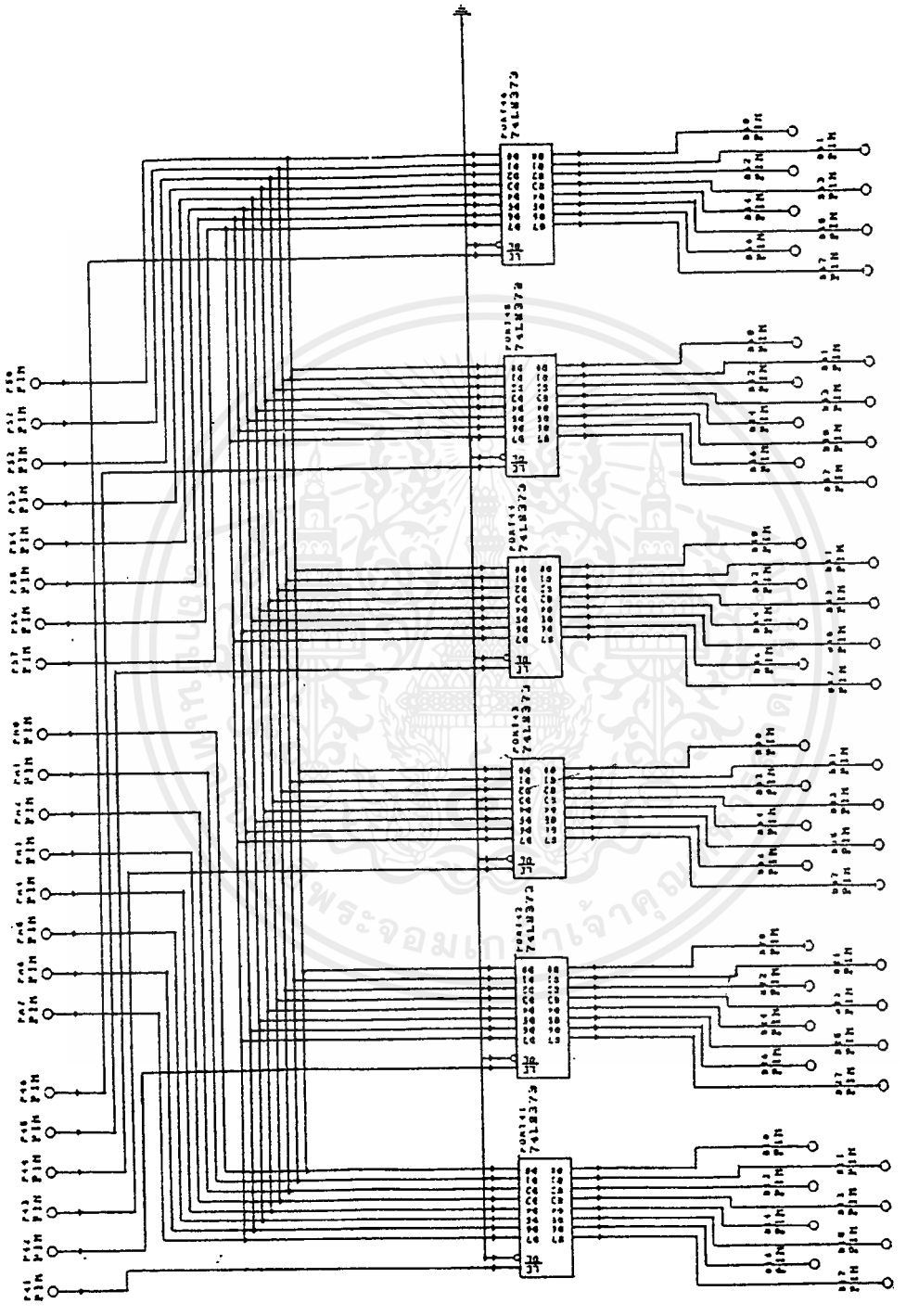
รูปที่ 2.10 แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



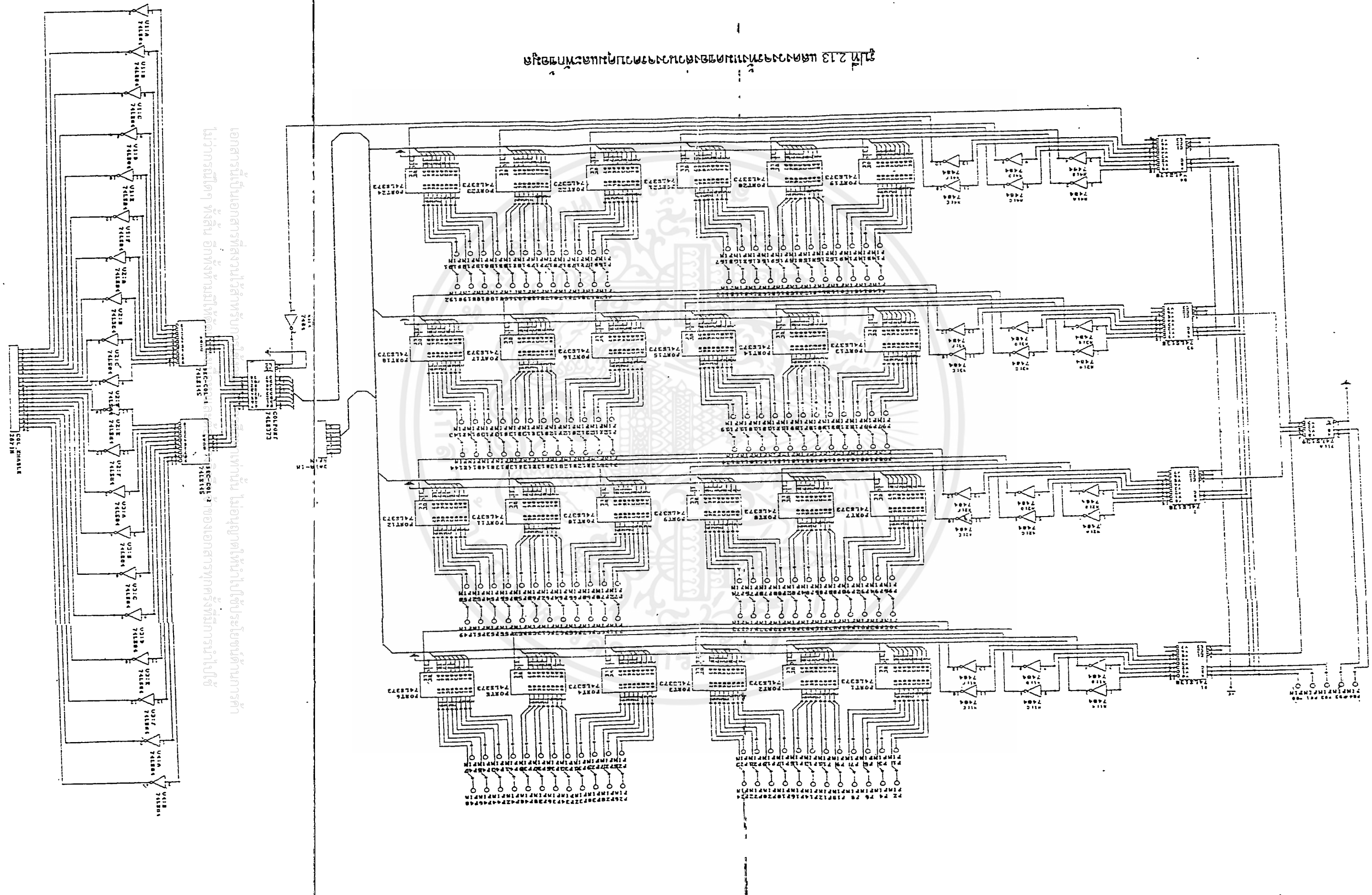
รูปที่ 2.11 แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



<u>รูปที่ 2.12</u> แสดงวงจร และการเชื่อมต่อพอร์ตพักข้อมูลของบอร์ดย่อยที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงการเชื่อมต่อวงจรเข้ากับบัสของระบบ

บทที่ 3
ไมโครคอนโทรลเลอร์
(Microcontroller)

สำหรับภาคการศึกษาที่ผ่านมา คณะผู้จัดทำได้ทดลองการทำงานบนไมโครคอนโทรลเลอร์ เบอร์ 8032 (JAZZ 32) ซึ่งอาศัยการทำงานซิสเต็มคอล (System Call) ในการเริ่มทำงาน จึงทำให้ไม่สามารถรับข้อมูลได้ด้วยตัวเอง นอกจากนั้นความเร็วในการทำงานยังไม่เป็นที่พอใจ

จากเหตุผลข้างต้นนี้ จึงนำไมโครคอนโทรลเลอร์ Z80180 มาประยุกต์ใช้ ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีรอบการทำงาน และความเร็วในการทำงานสูง และยังได้พัฒนาให้สามารถเริ่มทำงานได้โดยอัตโนมัติ โดยอาศัยสัญญาณอินเทอร์รัพท์ (Interrupt Signal)

ดังนั้นสามารถแสดงเหตุผลที่เลือกใช้ไมโครคอนโทรลเลอร์ Z80180 แทน แจ๊ส 32 ได้ดังนี้ คือ

1. <u>ความเร็วในการปฏิบัติตามชุดคำสั่ง</u> (Instruction Operation)

เนื่องจาก Z80180 แบ่ง 1 รอบทำงาน (Machine Cycle) ออกเป็น 3 สเตท (States) โดยจะใช้เวลาในแต่ละสเตทเท่ากับ 0.1 ไมโครวินาที ส่วน แจ๊ส 32 แบ่ง 1 รอบทำงานเป็น 6 สเตท โดยจะใช้เวลาในแต่ละสเตทเท่ากับ 1 ไมโครวินาที ซึ่งจะเห็นได้ว่า Z80180 ทำงานได้เร็วกว่า แจ๊ส 32 เมื่อทำงานตามคำสั่งที่มีจำนวนของรอบทำงานเท่ากัน ตัวอย่างเช่น คำสั่งการเคลื่อนย้ายข้อมูลผ่านแอคคิวมูเลเตอร์ (Accumulator) ใน Z80180 คือ

LD A, (BC)

จะใช้เวลาในการปฏิบัติตามคำสั่งนี้ 2 รอบทำงาน 6 สเตท ๆ ละ 0.1 ไมโครวินาที ดังนั้นจะใช้เวลาทั้งหมด 0.6 ไมโครวินาที

ในขณะที่ แจ๊ส 32 ใช้คำสั่งในการเคลื่อนย้ายข้อมูลจากรีจิสเตอร์ผ่านแอคคิวมูเลเตอร์เป็น

MOV A, Rn

แจ๊ส 32 จะใช้เวลาในการปฏิบัติตามคำสั่งนี้ 1 รอบทำงาน 12 สเตท ๆ ละ 1 ไมโครวินาที ดังนั้นจะใช้เวลาทั้งหมด 12 ไมโครวินาที

จะเห็นว่า Z80180 จะทำงานได้เร็วกว่า แจ๊ส 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การใช้การอินเทอร์รัพท์เพื่อเริ่มทำงาน

ดังอธิบายมาแล้วข้างต้น เราได้พัฒนาให้ Z80180 สามารถรับข้อมูลได้โดยอัตโนมัติ โดยอาศัยสัญญาณอินเทอร์รัพท์แทนการทำงานซิสเต็มคอล (System Call) จึงทำให้ Z80180 สามารถทำการรับคำสั่งจากคอมพิวเตอร์ได้โดยอัตโนมัติอย่างอิสระ

3. การอ้างอิงหน่วยความจำ (Memory Addressing)

Z80180 สามารถอ้างอิงหน่วยความจำได้ 64 กิโลไบต์ (16 บิต) ซึ่งเท่ากับการอ้างอิงข้อมูลในหน่วยความจำของแจ๊ส32 แต่ใน Z80180 มีอุปกรณ์ภายในที่สามารถทำการแปลง (Translation) หน่วยความจำจาก 64 กิโลไบต์ (16 บิต) ไปเป็น 1024 กิโลไบต์ (20 บิต) ซึ่งอุปกรณ์นั้นก็คือ หน่วยจัดการหน่วยความจำ (MMU หรือ Memory Management Unit)

4. อุปกรณ์ประกอบภายใน (On-Chip Peripheral Devices)

Z80180 มีอุปกรณ์ภายในมากกว่า แจ๊ส 32 ซึ่งสามารถนำมาประยุกต์ใช้งานได้กว้างขึ้น ซึ่งอุปกรณ์ภายใน Z80180 ที่สำคัญมีดังนี้

4.1 หน่วยจัดการหน่วยความจำ (MMU หรือ Memory Management Unit)

ใช้ในการขยายจำนวนแอดเดรสที่สามารถอ้างอิงได้ในหน่วยความจำ

4.2 ตัวควบคุมการเกิดอินเทอร์รัพท์ (Interrupt Controller)

Z80180 จะมีแหล่งสัญญาณอินเทอร์รัพท์ (Interrupt Source) 12 แหล่ง ซึ่งแบ่งออกเป็นจากภายนอก (External Interrupt Source) 4 แหล่ง และจากภายใน (Internal Interrupt Source) 8 แหล่ง แต่ แจ๊ส 32 จะมีแหล่งสัญญาณอินเทอร์รัพท์ (Interrupt Source) 5 แหล่ง (แบ่งเป็น Internal Interrupt Source 3 แหล่ง และ External Interrupt Source 2 แหล่ง) เท่านั้น ซึ่งทำให้ Z80180 สามารถใช้งานอินเทอร์รัพท์ได้มากขึ้น

4.3 ตัวจับเวลาขนาด 16 บิต (16-bit Timer)

สำหรับ Z80180 ก็จะมีตัวจับเวลาขนาด 16 บิต จำนวน 2 แชนเนล (Channel) ไว้ประยุกต์ใช้เช่นเดียวกันกับ แจ๊ส 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ตัวควบคุมการรับส่งข้อมูลแบบดีเอ็มเอ (DMAC หรือ Direct Memory Access Controller)

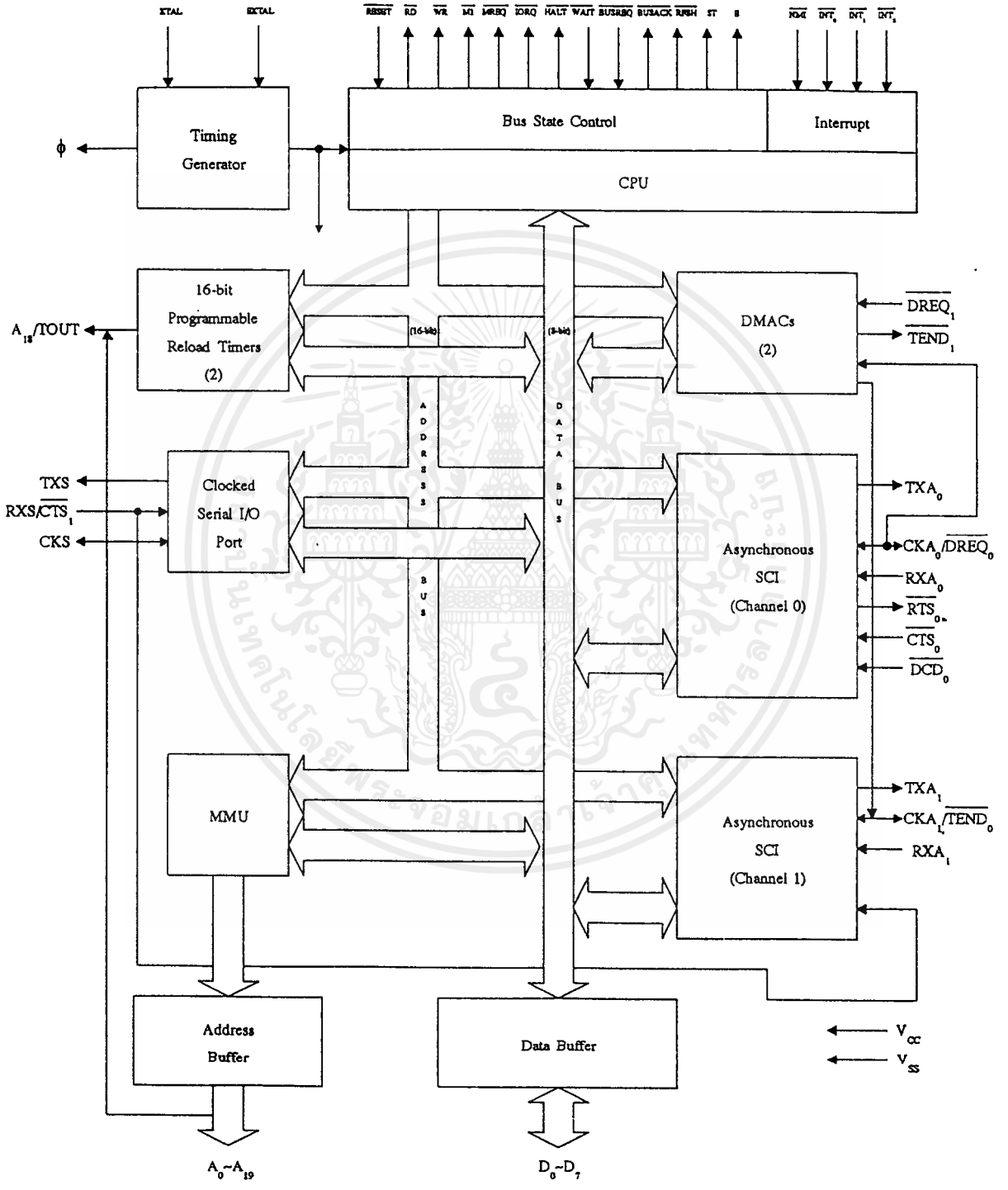
Z80180 สามารถใช้งานดีเอ็มเอ (DMA หรือ Direct Memory Access) ได้กล่าวคือ สามารถรับส่งข้อมูลโดยตรงจากหน่วยความจำ โดยไม่ต้องผ่านหน่วยประมวลผลกลาง หรือซีพียู (Central Processing Unit หรือ CPU) ซึ่งจะสามารถทำการรับคำสั่งจากคอมพิวเตอร์พร้อมกับโหลดข้อมูลที่จะแสดงบนบอร์ดได้พร้อมกัน โดยการรับส่งข้อมูลแบบนี้จะถูกควบคุมโดยตัวควบคุมดังกล่าวนี้

จะเห็นว่า Z80180 จะมีความสามารถโดยรวมสูงกว่า แจ๊ส 32 ดังนั้นจึงเลือกใช้ Z80180 มาประยุกต์ใช้กับโครงงานนี้เพื่อพัฒนาให้การทำงานรวดเร็วมากขึ้น

แต่อย่างไรก็ตาม เมื่อมีการเปลี่ยนแปลงคุณสมบัติทางกายภาพจากการเปลี่ยนไมโครคอนโทรลเลอร์ที่ใช้ จึงทำให้เกิดปัญหาขึ้น ซึ่งสามารถแก้ไขได้ดังต่อไปนี้

- สำหรับการส่งข้อมูลไปยังบอร์ดแสดงข้อมูล โดยที่พอร์ตในการรับข้อมูลจากไมโครคอนโทรลเลอร์ขนาดเป็น 26 ขา (Pin) แต่ไมโครคอนโทรลเลอร์ (ทั้ง JAZZ 32 และ Z80180) จะมีขนาดของพอร์ตที่ใช้ส่งข้อมูลไปยังบอร์ดแสดงข้อมูล เป็น 34 ขา จึงต้องทำตัวแปลงสัญญาณข้อมูลจากไมโครคอนโทรลเลอร์ ซึ่งขนาด 34 ขา ไปเป็น 26 ขา เพื่อให้ส่งข้อมูลไปยังบอร์ดแสดงข้อมูลได้
- เนื่องจาก Z80180 จะต้องใช้ไฟเลี้ยง (V_{cc}) ขนาด 5 โวลท์จึงจะสามารถทำงานได้ ถ้าจ่ายไฟเลี้ยงขาด หรือเกินจากนี้ Z80180 จะไม่ทำงาน จึงเป็นข้อควรระวังในการใช้งาน
- จากวงจรเดิม สายไฟที่ใช้จ่ายไฟเลี้ยงให้กับบอร์ดแสดงข้อมูลมีขนาดเล็กเกินไป ซึ่งทำให้ทนค่ากระแสไฟฟ้าได้ต่ำ จึงทำการเปลี่ยนขนาดสายไฟเลี้ยงนี้ให้มีขนาดใหญ่ขึ้นเพื่อให้ทนค่ากระแสไฟฟ้าได้สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แสดงบล็อคไดอะแกรมของส่วนประกอบต่าง ๆ ของไมโครคอนโทรลเลอร์ Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของไมโครคอนโทรลเลอร์

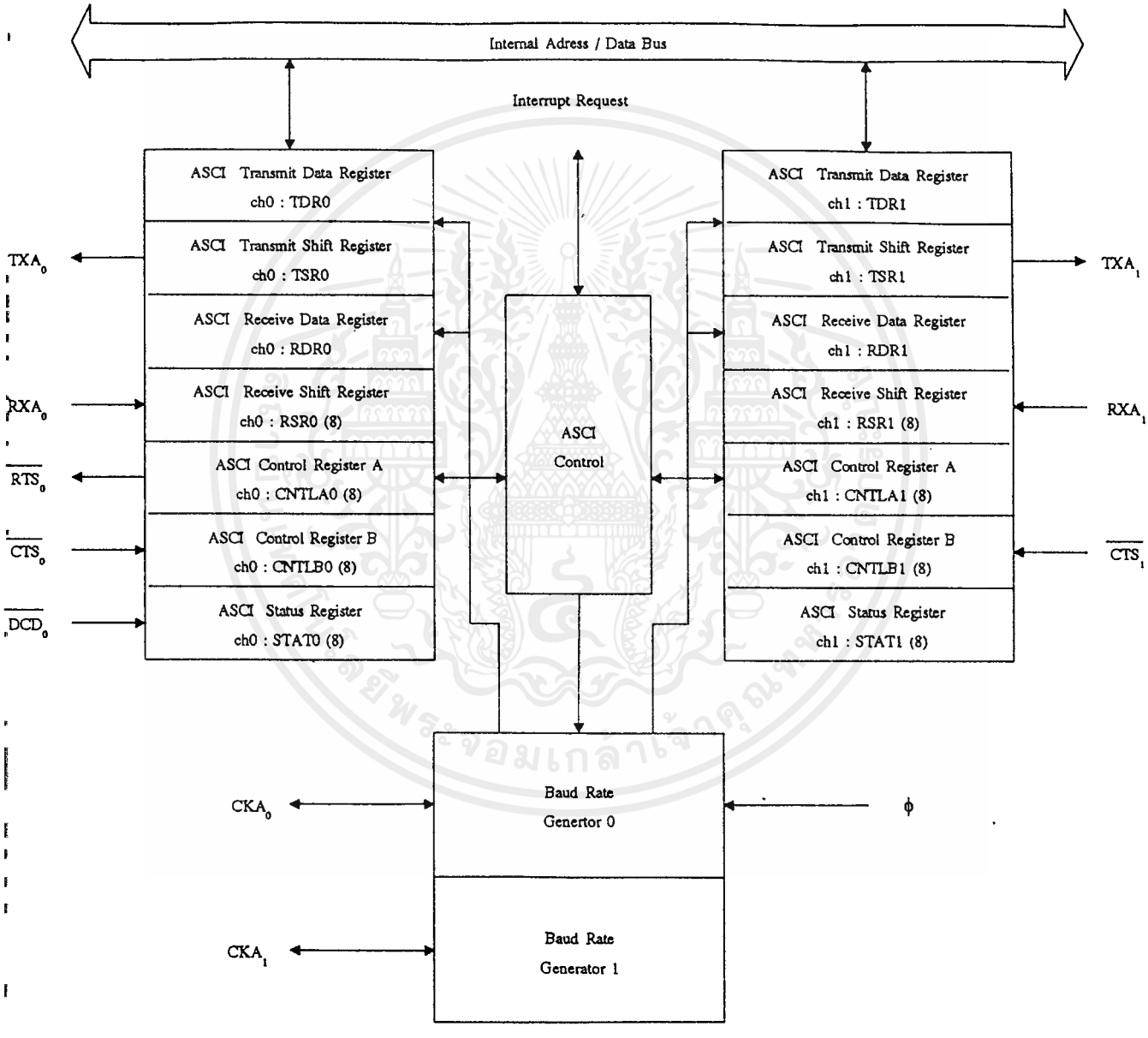
เราสามารถแบ่งการทำงานของไมโครคอนโทรลเลอร์ได้เป็น 2 ส่วน คือ

3.1 ช่วงการรับข้อมูลจากคอมพิวเตอร์

คอมพิวเตอร์จะส่งข้อมูลซึ่งเป็นแบบตัวอักษร หรือแบบรูปภาพไปยังไมโครคอนโทรลเลอร์ โดยผ่านเข้าทางพอร์ตอนุกรม (Serial Port) ของไมโครคอนโทรลเลอร์ ซึ่งพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ Z80180 ได้แก่ส่วนที่เรียกว่า เอเอสซีไอ (ASCI หรือ Asynchronous Serial Communication Interface) ดังแสดงในรูปที่ 3.1 และประกอบด้วย RS-232 ที่ควบคุมการรับส่งข้อมูลแบบอนุกรมเพื่อรับข้อมูลที่จะนำออกแสดงทางบอร์ดแสดงข้อมูลจากพอร์ตอนุกรมของคอมพิวเตอร์โดยเอเอสซีไอจะมีคุณสมบัติที่สำคัญดังนี้

- สามารถใช้รับส่งข้อมูลในแบบฟูลดูเพล็กซ์ (Full Duplex) ได้
- รับส่งข้อมูลได้คราวละ 7 บิต หรือ 8 บิต
- ใช้งานในโหมดของมัลติโปรเซสเซอร์ (Multiprocessor) ได้
- สามารถทำการเซ็ทให้มีสต๊อบบิต (Stop Bit) เป็นจำนวน 1 บิต หรือ 2 บิตก็ได้
- ใช้ตรวจสอบภาวะเสมอมูล (Parity) ได้
- สามารถจะโปรแกรมให้อัตรารับส่งข้อมูล (Baud Rate) อยู่ในโหมดของ $\frac{1}{16}$ และโหมดของ $\frac{1}{64}$ โดยถ้าหน่วยประมวลผลกลาง หรือซีพียู ทำงานที่ความถี่สูงสุด 6.144 เมกะเฮิรตซ์ และ Z80180 สามารถรับส่งข้อมูลได้ด้วยความเร็วสูงสุดถึง 38.4 กิโลบิตต่อวินาที (Kilobits per Second หรือ kbps)
- สามารถใช้งานกับโมเดม (Modem) ได้
- สามารถกำหนดเงื่อนไขการอินเทอร์รัพท์ได้
- ใช้ทำงานร่วมกับตัวควบคุมการรับส่งข้อมูลแบบดีเอ็มเอได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



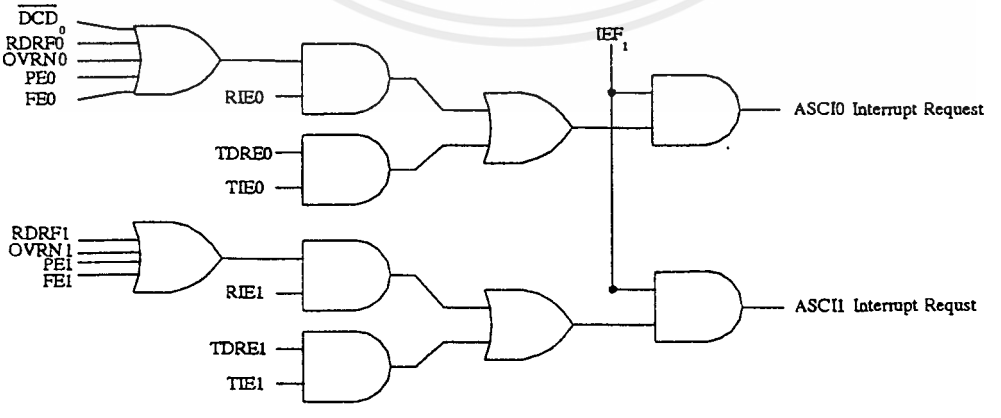
<u>รูปที่ 3.2</u> แสดงบล็อคไดอะแกรมของส่วนประกอบต่าง ๆ ของเอเอสซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเอสซีไอจะมีส่วนประกอบต่าง ๆ ดังแสดงในรูปที่ 3.2 ซึ่งจากรูป ในการรับข้อมูลจากคอมพิวเตอร์ ข้อมูลจะถูกส่งผ่านเข้าทางขา RXA ของเอเอสซีไอ แล้วถูกส่งเข้ามาเก็บในรีจิสเตอร์ RSR (ASCI Receive Shift Register) และเมื่อรับเข้ามาจนเต็ม (ครบ 1 ไบต์) แล้ว ข้อมูลทั้งหมดจะถูกส่งไปเก็บในรีจิสเตอร์ RDR (ASCI Receive Data Register) ถ้ารีจิสเตอร์ RDR ยังไม่เต็ม โดยในขณะที่ข้อมูลถูกส่งออกไปยังรีจิสเตอร์ RDR หากมีข้อมูลไบต์ถัดไปเข้ามาจะทำให้เกิดโอเวอร์รัน (Overrun Error) ขึ้น ดังนั้นข้อมูลไบต์ถัดไปจะส่งเข้ามาเก็บในรีจิสเตอร์ RSR ได้ก็ต่อเมื่อข้อมูลไบต์ก่อนหน้าถูกส่งไปเก็บในรีจิสเตอร์ RDR จนครบเรียบร้อยแล้ว โดยปกติจะไม่มีข้อมูลจากบัสข้อมูล (Data Bus) ส่งเข้ามาเก็บในรีจิสเตอร์ RDR เนื่องจากรีจิสเตอร์ RDR เป็นรีจิสเตอร์ที่ใช้อ่านได้อย่างเดียว (Read Only Register) แต่อย่างไรก็ตาม ในกรณีที่บิต RDRF (Receive Data Register Full) ในรีจิสเตอร์สถานะของเอเอสซีไอ (ASCI Status Register) มีค่าเป็น 0 เราก็จะสามารถทำการเขียนข้อมูลเก็บลงในรีจิสเตอร์ RDR ได้ และสามารถอ่านข้อมูลนั้นจากรีจิสเตอร์ RDR ได้เช่นกัน

หลักการรับข้อมูลจากคอมพิวเตอร์ของไมโครคอนโทรลเลอร์

คณะผู้จัดทำได้กำหนดให้ใช้สัญญาณอินเทอร์รัพท์เป็นสัญญาณกำหนดการรับข้อมูลของไมโครคอนโทรลเลอร์จากคอมพิวเตอร์ โดยสัญญาณอินเทอร์รัพท์จะถูกส่งไปยังหน่วยประมวลผลกลางของไมโครคอนโทรลเลอร์เพื่อขอรับการบริการ ถ้าได้รับการตอบรับ ระบบก็จะไปทำงานในรูทีนบริการการอินเทอร์รัพท์ (Interrupt Service Routine) ซึ่งก็คือโปรแกรมการรับข้อมูลจากคอมพิวเตอร์นั่นเอง



รูปที่ 3.3 แสดงวงจรเสมือนในการเกิดสัญญาณอินเทอร์รัพท์จากเอเอสซีไอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์รัพท์จะเกิดขึ้นได้โดยการกำหนดค่าให้กับรีจิสเตอร์ต่าง ๆ ของเอเอสซีไอดังที่ได้กล่าวมาแล้ว รวมทั้งค่าของแฟลก IEF_1 (Interrupt Enable Flag 1) จะต้องถูกกำหนดให้มีค่าเป็น 1 ด้วย ซึ่งสามารถแสดงเงื่อนไขของการเกิดสัญญาณอินเทอร์รัพท์จากเอเอสซีไอเป็นรูปวงจรทางไฟฟ้าได้ดังรูปที่ 3.3 ส่วนการกำหนดค่าแอดเดรสเริ่มต้นของรูทีนบริการการอินเทอร์รัพท์ของการรับข้อมูลจากวงจรที่ใช้งาน สามารถกำหนดได้ดังตารางต่อไปนี้

INTERRUPT SOURCE	IL			FIXED CODE				
	b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0
ASCI CHANNEL 0	.	.	.	0	1	1	1	0

ตารางที่ 3.1 แสดงค่าบิตต่าง ๆ ที่ใช้กำหนดค่าแอดเดรสเริ่มต้นของรูทีนบริการการอินเทอร์รัพท์ (Interrupt Service Routine)

บิต b_7, b_6 และ b_5 จะกำหนดได้ที่รีจิสเตอร์ IL (Interrupt Vector Low Register) ซึ่งโดยปกติแล้ว ค่าของรีจิสเตอร์ IL จะเป็นค่าใดก็ได้ ดังนั้นในการรับข้อมูลจากคอมพิวเตอร์นั้น เราสามารถที่จะกำหนดแอดเดรสเริ่มต้นของรูทีนบริการการอินเทอร์รัพท์ (Interrupt Service Routine) ให้กับไมโครคอนโทรลเลอร์ให้มีค่าเท่าใดก็ได้ และมีค่าคงที่

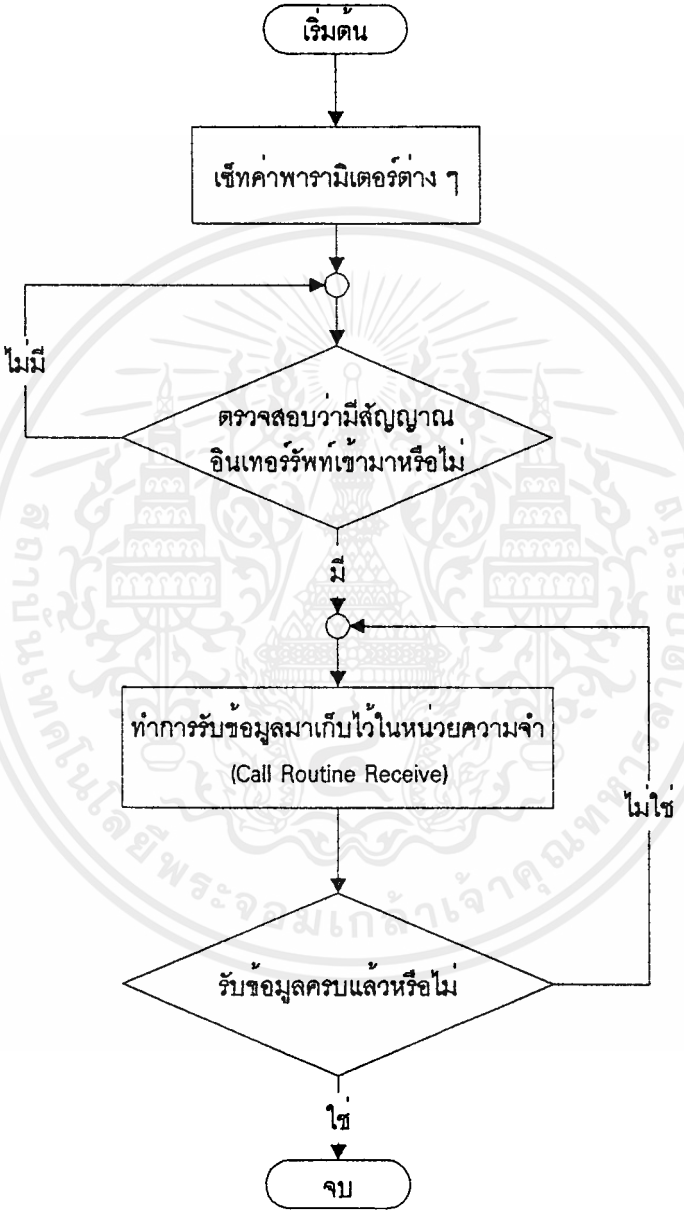
เมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูลจากคอมพิวเตอร์แล้ว ซีพียูของไมโครคอนโทรลเลอร์ก็จะกลับไปทำงานในรูทีนเดิมที่ทำอยู่ก่อนหน้าการเกิดการอินเทอร์รัพท์

3.2 ช่วงการส่งข้อมูลไปยังบอร์ดแสดงข้อมูล

หลักการส่งข้อมูลจากไมโครคอนโทรลเลอร์ Z80180 ไปยังบอร์ดแสดงข้อมูล ก็จะเป็นไปในลักษณะเดียวกันกับไมโครคอนโทรลเลอร์ JAZZ 32 กล่าวคือจะเชื่อมต่อกับบอร์ดแสดงข้อมูลทางไอซี เบอร์ 8255 ซึ่งเป็นพอร์ตที่สามารถทำการโปรแกรมได้ (Programmable Input/Output Port) เพื่อส่งข้อมูลออกไปให้แก่วงจรตัวขับของบอร์ดแสดงข้อมูล

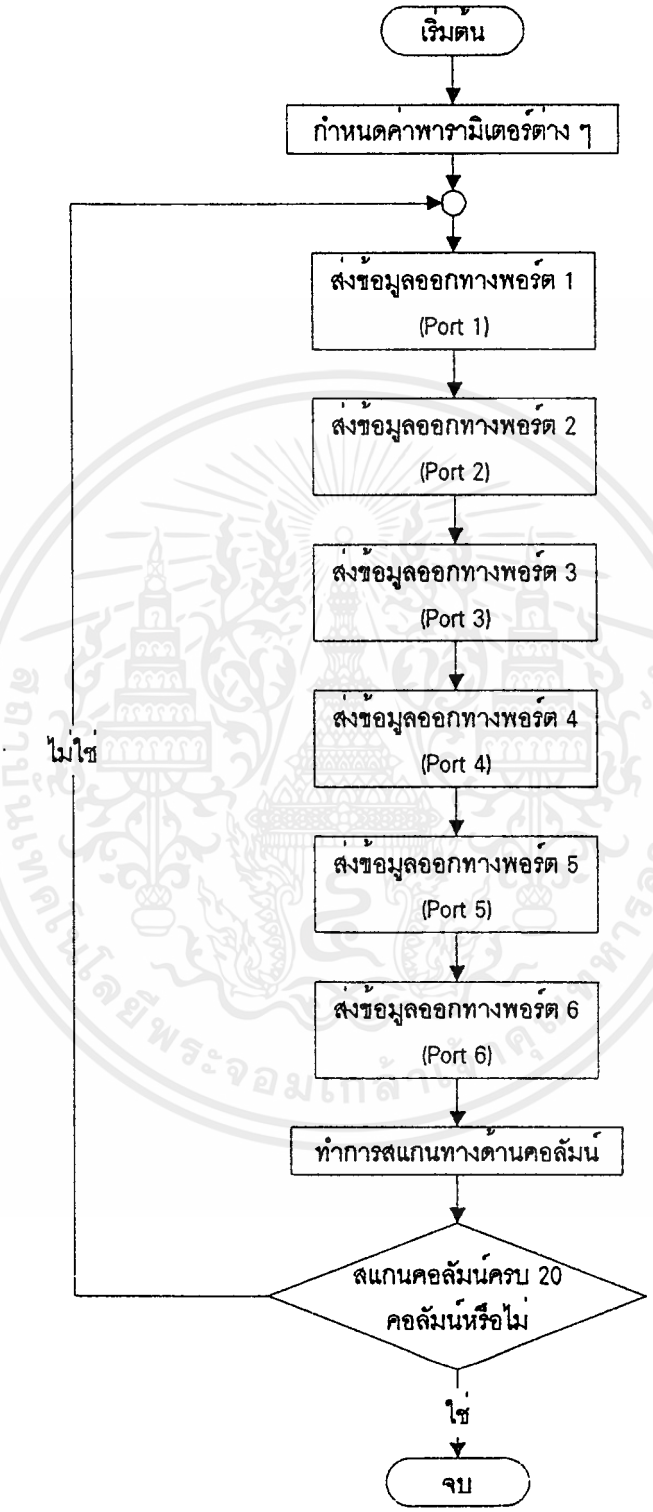
ส่วนโปรแกรมควบคุมการทำงานบนไมโครคอนโทรลเลอร์นี้ จะประกอบไปด้วยโปรแกรมการรับข้อมูลจากคอมพิวเตอร์ และโปรแกรมการส่งข้อมูลไปยังบอร์ดแสดงข้อมูล ซึ่งจะมี Source Code แสดงในภาคผนวก ค. และมีลักษณะการทำงานเป็นไปตามผังงาน (Flowchart) ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการรับข้อมูลจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงผังงานของโปรแกรมบนไมโครคอนโทรลเลอร์ในการส่งข้อมูลไปบอร์ดแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
คอมพิวเตอร์ส่วนบุคคล
(Personal Computer)

สำหรับโครงงานนี้ จะใช้คอมพิวเตอร์เป็นเครื่องมือในการกำหนดตำแหน่ง และสีของแอลอีดีแต่ละหลอด โดยได้มีการพัฒนาในส่วนของโปรแกรมบนคอมพิวเตอร์ให้สร้างบอร์ดจำลองบนหน้าจอ (Monitor) ของคอมพิวเตอร์โดยที่เราสามารถควบคุมการติด หรือดับ และสามารถกำหนดสีที่จะแสดงของแอลอีดีในตำแหน่งต่าง ๆ ได้ โดยใช้เมาส์ (Mouse) เป็นเครื่องมือในการกำหนดตำแหน่ง และสี แล้วส่งข้อมูลไปให้ไมโครคอนโทรลเลอร์ ซึ่งทำให้สามารถกำหนดข้อมูลไม่ว่าจะเป็นตัวอักษร หรือรูปสัญลักษณ์ได้ตามต้องการ นอกจากนั้นสำหรับข้อมูลที่เป็นตัวอักษรนั้น แทนที่ผู้ใช้จะต้องออกแบบที่บอร์ดจำลองบนหน้าจอคอมพิวเตอร์ คณะผู้จัดทำได้พัฒนาให้ผู้ใช้สามารถกำหนดข้อมูลที่เป็นอักษรนี้ผ่านทางแป้นพิมพ์ (Keyboard) ได้เลย

นอกจากนี้ยังได้พัฒนาโปรแกรมบนคอมพิวเตอร์เพื่อให้สามารถแสดงข้อมูลข่าวสารออกทางบอร์ดได้หลายรูปแบบ

4.1 รูปแบบของข้อมูลข่าวสารที่จะแสดงออกทางบอร์ดแสดงข้อมูล

ข้อมูลข่าวสารที่จะแสดงออกทางบอร์ดแสดงข้อมูล สามารถแบ่งได้เป็น 2 รูปแบบหลัก ๆ คือ

ก. ข้อมูลข่าวสารที่เป็นตัวอักษร

คณะผู้จัดทำได้พัฒนาโปรแกรมให้ผู้ใช้สามารถพิมพ์ตัวอักษรหรือข้อความที่ต้องการแสดงออกทางบอร์ดแสดงข้อมูลทางแป้นพิมพ์ได้เลย ข้อความที่พิมพ์จะเป็นข้อความภาษาไทยและ/หรือ ภาษาอังกฤษ ซึ่งจะต้องตรวจสอบว่า คีย์ที่กดนั้นต้องการให้เป็นภาษาไทย หรือภาษาอังกฤษ โดยใช้คีย์ [~ %] บนแป้นพิมพ์ เพื่อเป็นแฟลกบอกสถานะ โดยจะใช้แฟลกชนิดบูลีน (Boolean) กล่าวคือ เมื่อแฟลกมีสถานะเป็นจริง (True) ให้หมายความว่า คีย์ที่กดเป็นอักษรภาษาไทย และเมื่อแฟลกมีสถานะเป็นเท็จ (False) ให้หมายความว่า คีย์ที่กดเป็นภาษาอังกฤษ

จากนั้นก็ทำการตรวจสอบรหัสแอสกี (ASCII) ของคีย์ที่ถูกกด โดยในกรณีที่แฟลกมีสถานะเป็นจริง (คือต้องการให้ตัวอักษรเป็นภาษาไทย) ก็จะนำรหัสแอสกีของคีย์นั้นไปเทียบกับมาตรฐานรหัสแอสกีของคีย์ในแบบภาษาไทย (ซึ่งเราได้นำของเวิร์ดราชวิถีมาใช้) แต่ในกรณีที่แฟลกมีสถานะเป็นเท็จ (คือต้องการให้ตัวอักษรเป็นภาษาอังกฤษ) ก็สามารถนำเอารหัสแอสกีของคีย์นั้นไปใช้ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสแอสกีที่ได้นี้จะนำไปใช้ในการเรียกภาพของตัวอักษรที่มีรหัสแอสกีตรงกัน ซึ่งได้เก็บเป็นบิตแม็ปไว้ในหน่วยความจำของคอมพิวเตอร์ แล้วนำภาพนั้นแสดงออกทางบอร์ดแสดงข้อมูลได้ตามคีย์ที่ถูกกด และผู้ใช้สามารถจะเปลี่ยนรูปแบบของตัวอักษรได้ (โดยรูปแบบของตัวอักษรที่คณะผู้จัดทำได้กำหนดใช้เป็นของเวิร์ดราชวิถี) โดยการใช้รูปแบบของตัวอักษรจากมาตรฐานอื่น ๆ หรือการออกแบบเองบนโปรแกรม EDIT_FON.EXE ที่เราจัดทำไว้ให้

เนื่องจากบอร์ดต้องแสดงทั้งข้อความภาษาไทย และภาษาอังกฤษ ดังนั้นจึงต้องแบ่งบอร์ดแสดงผลออกเป็น 3 ส่วนในแนวตั้ง (3 บรรทัด) เพราะตามปกติคำในภาษาไทยจะประกอบด้วยพยัญชนะ วรรณยุกต์ และสระ เช่น คำว่า "นั้น" หรือ "กู้" เป็นต้น โดยบรรทัดที่ 1 จะใช้แสดง วรรณยุกต์ และ/หรือ สระ บรรทัดที่ 2 จะใช้แสดงพยัญชนะ และบรรทัดที่ 3 จะใช้แสดงสระ

ข. ข้อมูลข่าวสารที่เป็นรูปภาพ

สำหรับข้อมูลในแบบรูปภาพนั้นจะไม่มีความแน่นอนเหมือนกับตัวอักษรที่มีรหัสแอสกีของแต่ละคีย์ จึงไม่สามารถจะทำตารางเปรียบเทียบได้ ยังคงต้องใช้การออกแบบบนบอร์ดจำลองบนหน้าจอแสดงผลของคอมพิวเตอร์

4.2 รูปแบบในการแสดงข้อมูลข่าวสาร

ส่วนรูปแบบในการแสดงข้อมูลออกทางบอร์ดแสดงข้อมูลสามารถจะแบ่งได้เป็น 2 แบบหลัก ๆ คือ

4.2.1 การแสดงข้อมูลข่าวสารแบบภาพนิ่ง

เป็นการแสดงข้อมูลที่เป็นตัวอักษร และ/หรือ รูปภาพ ขนาด 24x80 จุด ออกทางบอร์ดแสดงข้อมูล โดยการส่งข้อมูลออกทางพอร์ตอนุกรม ผ่านพอร์ต A ของไอซี 8255 แล้วเข้าสู่เอเอสซีไอของไมโครคอนโทรลเลอร์ Z80180 แล้วออกแสดงทางบอร์ดแสดงข้อมูล

4.2.2 การแสดงข้อมูลข่าวสารแบบเคลื่อนไหว

สามารถแสดงข้อมูลออกทางบอร์ดแสดงข้อมูลแบบเคลื่อนไหวได้ 3 รูปแบบ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.1 <u>การแสดงข้อความต่อเนื่อง</u>

การแสดงข้อความต่อเนื่อง คือการแสดงข้อความออกทางบอร์ดแสดงข้อมูลจากทางขวาไปซ้าย โดยอาศัยการส่งข้อมูลทั้งหมดออกทางพอร์ตอนุกรม ผ่านพอร์ต A ของไอซี 8255 แล้วเข้าสู่เอเอสซีไอของไมโครคอนโทรลเลอร์ Z80180 จากนั้นทำการโปรแกรมให้ไมโครคอนโทรลเลอร์นั้นทำการแสดงข้อมูลที่ถูกส่งมาเก็บที่พอร์ต A

หลังจากการรับข้อมูลที่เป็นข้อความทางแป้นพิมพ์แล้ว ก็จะนำมาเก็บไว้ในบัฟเฟอร์ จากนั้นทำการส่งข้อมูลไปแสดงบนบอร์ด โดยจะแบ่งข้อมูลทั้งหมดจาก 24x80 จุด ออกเป็น 4 ช่วง ๆ ละ 24x20 จุด เพื่อประหยัดเนื้อที่บนไมโครคอนโทรลเลอร์ และทำให้ข้อมูลแสดงออกทางบอร์ดแบบต่อเนื่องได้อย่างรวดเร็วกว่าที่ใช้เป็น 24x80 จุด

เมื่อทำการแบ่งข้อมูลออกเป็นส่วนย่อย 4 ส่วนแล้ว เมื่อจะส่งข้อมูลก็จะใช้การพักข้อมูลโดยบอร์ดที่แบ่งเป็น 4 ส่วนนั้น ในแต่ละส่วนจะสามารถทำงานแยกจากกันได้อย่างอิสระหรือทำงานต่อเนื่องกันทั้งหมดได้ สำหรับข้อมูลที่ถูกแบ่งออกเป็น 4 ส่วน

เริ่มต้นโดยการส่งข้อมูลเข้าไปพักตามพอร์ตต่าง ๆ ที่คอลัมน์สุดท้ายของบอร์ดแสดงข้อมูล (เริ่มที่ส่วนที่สี่ของบอร์ดก่อน คือ ใช้ 24x20 จุดในส่วนสุดท้ายของบอร์ดก่อน ส่วนอื่นยังไม่ต้องใช้) ด้วยการใช้คอนโทรลเวิร์ด (Control Word) ผ่านทางพอร์ต B ของไอซี 8255 เพื่อทำให้พอร์ตสุดท้ายในส่วนสุดท้ายของบอร์ดทำการพักข้อมูล จากนั้นทำการส่งข้อมูลอีกหนึ่งชุดออกทางพอร์ต A เพื่อทำการสแกนคอลัมน์ ซึ่งจะทำให้ข้อมูลที่ถูกพักไว้ในตำแหน่งแอลอีดีของพอร์ตนั้นปล่อยแสงออกมา

จากนั้นทำการส่งข้อมูลอีกชุดเพื่อทำการดับหลอดทุกหลอดในส่วนสุดท้ายของบอร์ด แล้วส่งข้อมูลเดิมนี้ออกไปอีก แต่จะทำการพักข้อมูลไปที่ส่วนที่สามของบอร์ดแทน แล้วทำให้แอลอีดีในตำแหน่งบนส่วนที่สามของบอร์ดเปล่งแสงออกมาโดยใช้วิธีการดังที่กล่าวมาแล้ว หลังจากนั้นทำการดับหลอดแอลอีดี แล้วค่อยทำการส่งข้อมูลชุดใหม่ถัดจากข้อมูลที่ส่งไปยังส่วนสุดท้ายของบอร์ดแสดงข้อมูล แล้วทำให้หลอดสว่างด้วยวิธีการเดิม ทำเช่นนี้ไปเรื่อย ๆ จนถึงบอร์ดส่วนแรก ก็จะทำให้ข้อความที่แสดงออกมาทางบอร์ดแสดงข้อมูลเสมือนกับเป็นภาพต่อเนื่องที่เคลื่อนจากทางขวาไปทางซ้าย

โดยคณะผู้จัดทำได้กำหนดให้ข้อความที่ออกแสดงต่อเนื่อง สามารถมีจำนวนตัวอักษรเกินขนาดของบอร์ดที่จะแสดงได้ แต่ไม่เกิน 414 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 <u>การแสดงตัวอักษร และรูปภาพใน 4 ทิศทาง</u>

เป็นการแสดงข้อความ หรือรูปภาพ ขนาด 24x80 จุด (ขนาด 1 บอร์ด) ออกแสดงทางบอร์ด และให้เคลื่อนที่ได้ใน 4 ทิศทาง คือจากซ้ายไปขวา จากขวาไปซ้าย จากบนลงล่าง และจากล่างขึ้นบน

เมื่อต้องการแสดงข้อมูลให้เคลื่อนที่ได้ใน 4 ทิศทางดังกล่าว ก็จะเสมือนกับการแสดงภาพนิ่งของภาพหลาย ๆ เฟรม โดยแต่ละเฟรมจะเป็นภาพของแอลอีดีในตำแหน่งต่าง ๆ ของข้อมูลที่จะแสดง ซึ่งมีลักษณะต่าง ๆ ขึ้นอยู่กับรูปแบบการเคลื่อนที่ของข้อมูล ดังนี้

ก. การแสดงข้อมูลจากทางขวาไปซ้าย

เนื่องจากบอร์ดมีขนาด 24x80 จุด เมื่อต้องการแสดงข้อมูลจากขวาไปซ้ายก็เปรียบกับการแสดงข้อมูลภาพนิ่งทีละเฟรมทั้งหมด 80 เฟรม โดยเฟรมแรกก็คือภาพของแอลอีดีในคอลัมน์แรกของข้อมูลที่จะแสดง เฟรมที่ 2 จะเป็นภาพของแอลอีดีในสองคอลัมน์แรกของข้อมูลที่จะแสดง เฟรมที่ 3 ก็จะเป็นภาพของแอลอีดีในสามคอลัมน์แรกของข้อมูลที่จะแสดง เช่นนี้ไปเรื่อย ๆ จนถึงเฟรมที่ 80 ก็จะเป็นภาพของแอลอีดีในทุกคอลัมน์ของข้อมูลที่จะแสดง แล้วนำออกแสดงทีละเฟรมทางด้านขวาสุด (คอลัมน์ที่ 80) ของบอร์ด ด้วยระยะเวลาห่างในการแสดงภาพแต่ละเฟรมที่เหมาะสม ก็จะทำให้เห็นภาพข้อมูลเสมือนเคลื่อนที่จากขวาไปซ้ายได้

ข. การแสดงข้อมูลจากทางซ้ายไปขวา

เป็นการแสดงภาพนิ่ง 80 เฟรมเช่นเดียวกับในแบบแรก แต่เฟรมแรกจะเป็นภาพของแอลอีดีในคอลัมน์สุดท้ายของข้อมูลที่จะแสดง เฟรมที่ 2 จะเป็นภาพของแอลอีดีในสองคอลัมน์สุดท้ายของข้อมูลที่จะแสดง เฟรมที่ 3 ก็จะเป็นภาพของแอลอีดีในสามคอลัมน์สุดท้ายของข้อมูลที่จะแสดง เช่นนี้ไปเรื่อย ๆ จนถึงเฟรมที่ 80 ก็จะเป็นภาพของแอลอีดีในทุกคอลัมน์ของข้อมูลที่จะแสดง แล้วนำออกแสดงทีละเฟรมทางด้านซ้ายสุด (คอลัมน์แรก) ของบอร์ด ด้วยระยะเวลาห่างในการแสดงภาพแต่ละเฟรมที่เหมาะสม ก็จะทำให้เห็นภาพข้อมูลเสมือนเคลื่อนที่จากซ้ายไปขวาได้

ค. การแสดงข้อมูลจากบนลงล่าง

สำหรับการแสดงข้อความแบบนี้ ต้องทำการแบ่งข้อมูลเป็น 24 เฟรม โดยเฟรมแรกจะเป็นภาพของแอลอีดีในแถวล่างสุดของข้อมูลที่จะแสดง เฟรมที่ 2 จะเป็นภาพของแอลอีดีในสองแถวล่างสุดของข้อมูลที่จะแสดง เฟรมที่ 3 ก็จะเป็นภาพของแอลอีดีในสามแถวล่างสุดของข้อมูลที่จะแสดง เช่นนี้ไปเรื่อย ๆ จนถึงเฟรมที่ 24 ก็จะเป็นภาพของแอลอีดีในทุก ๆ แถวของข้อมูลที่จะแสดง จากนั้นก็นำออกแสดงทีละเฟรมทางด้านแถวบนสุด (แถวแรก) ของบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยช่วงเวลาห่างในการแสดงภาพแต่ละเฟรมที่เหมาะสม ก็จะทำให้เห็นภาพข้อมูลเสมือนเคลื่อนที่จากบนลงล่าง

ง. การแสดงข้อมูลจากล่างขึ้นบน

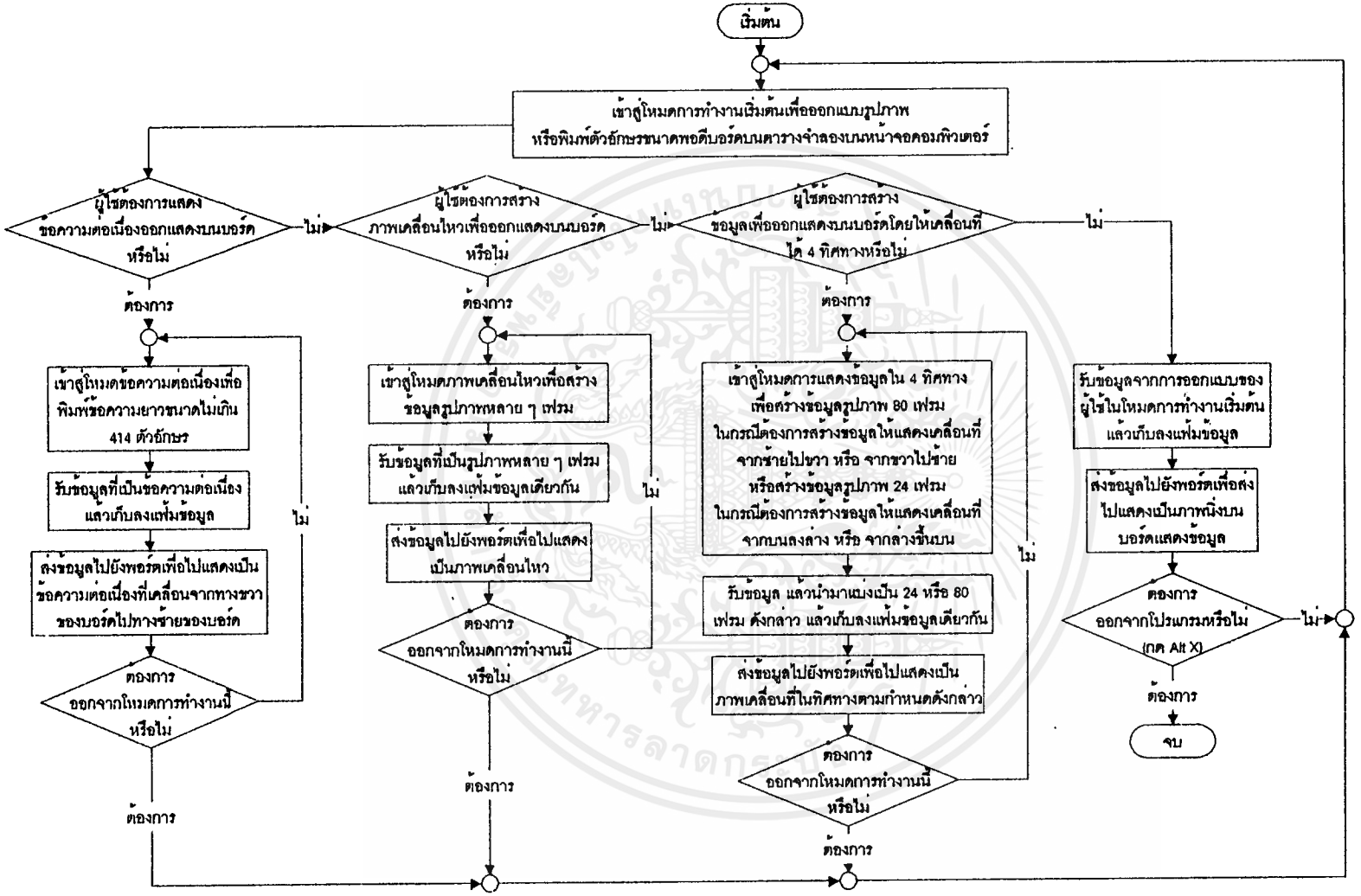
ยังคงต้องทำการแบ่งข้อมูลเป็น 24 เฟรมเช่นเดียวกับการแสดงข้อมูลจากบนลงล่าง แต่เฟรมแรกจะเป็นภาพของแอลอีดีในแถวบนสุดของข้อมูลที่จะแสดง เฟรมที่ 2 จะเป็นภาพของแอลอีดีในสองแถวบนสุดของข้อมูลที่จะแสดง เฟรมที่ 3 จะเป็นภาพของแอลอีดีในสามแถวบนสุดของข้อมูลที่จะแสดง เช่นนี้ไปเรื่อย ๆ จนถึงเฟรมที่ 24 ก็จะเป็นภาพของแอลอีดีในทุก ๆ แถวของข้อมูลที่จะแสดง จากนั้นก็นำออกแสดงทีละเฟรมทางด้านแถวล่างสุด (แถวที่ 24) ของบอร์ด ด้วยช่วงเวลาห่างในการแสดงภาพแต่ละเฟรมที่เหมาะสม ก็จะทำให้เห็นภาพข้อมูลเสมือนเคลื่อนที่จากล่างขึ้นบน

4.2.2.3 <u>การแสดงรูปภาพเคลื่อนไหว</u>

เป็นการแสดงภาพนิ่งหลาย ๆ เฟรมด้วยระยะเวลาห่างในการแสดงที่เหมาะสม โดยแต่ละเฟรมจะเป็นขั้นตอนการเคลื่อนไหวของภาพที่ต่อเนื่องกัน ผู้ใช้จะต้องออกแบบรูปภาพในแต่ละเฟรมบนตารางออกแบบข้อมูลบนหน้าจอแสดงผลของคอมพิวเตอร์โดยการใช้เมาส์กำหนดตำแหน่งต่าง ๆ และสีที่ตำแหน่งนั้นบนตาราง และต้องออกแบบรูปภาพให้เพียงพอเพื่อแสดงการเปลี่ยนแปลงที่สำคัญของภาพเคลื่อนไหว หลังจากนั้นเราจะทำการเชื่อมโยงทุกเฟรมให้เป็นไฟล์เดียวกัน เพื่อนำแต่ละเฟรมแสดงออกเป็นภาพนิ่งอย่างต่อเนื่อง ซึ่งจะทำให้เห็นรูปภาพเสมือนว่าเคลื่อนไหวได้

ส่วนโปรแกรมควบคุมการทำงานบนคอมพิวเตอร์จะมีรายละเอียดดังแสดงในภาคผนวก ง. ซึ่งจะมีลักษณะการทำงานเป็นไปตามผังงาน (Flowchart) ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดงผังงานการทำงานของโปรแกรมควบคุมการทำงานของคอมพิวเตอร์

สรุปผล และวิจารณ์

สำหรับการทำโครงงานที่ผ่านมา สามารถสรุปเป็นขั้นตอนที่สำคัญได้ดังนี้

1. เริ่มจากการศึกษาการทำงานของวงจรต่าง ๆ บนบอร์ดแสดงผล

เป็นการศึกษาการทำงานของวงจรบนบอร์ดแสดงผล ซึ่งบอร์ดแสดงผลได้รับการจัดสร้างและพัฒนามาก่อนแล้ว เราเพียงแต่ศึกษาถึงการนำข้อมูลที่ส่งมา แล้วทำให้แอลอีดีสว่าง หรือดับตามข้อมูลนั้น เพื่อใช้ในการเขียนโปรแกรมควบคุมการส่งข้อมูลของไมโครคอนโทรลเลอร์ ซึ่งรายละเอียดได้แสดงไว้แล้วในบทที่ 2

2. การศึกษาการทำงานของไมโครคอนโทรลเลอร์ Z80180 ในการรับส่งข้อมูล

เป็นการศึกษาลักษณะการทำงาน และส่วนประกอบที่สำคัญของไมโครคอนโทรลเลอร์ Z80180 ในการรับส่งข้อมูล เพื่อใช้ในการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ในการรับข้อมูลจากคอมพิวเตอร์ และการส่งข้อมูลไปแสดงบนบอร์ด แต่สำหรับโปรแกรมในส่วนนี้ เราเขียนเพียงเพื่อใช้ทดสอบว่า ข้อมูลจากคอมพิวเตอร์ได้ส่งมายังไมโครคอนโทรลเลอร์ และนำออกแสดงบนบอร์ดได้จริง

3. การพัฒนาโปรแกรมการทำงานของคอมพิวเตอร์

การพัฒนาโปรแกรมบนคอมพิวเตอร์ ก็เพื่อให้ผู้ใช้สามารถจัดการ ออกแบบ และแก้ไขข้อมูลรูปภาพ และตัวอักษร ที่จะนำออกแสดงบนบอร์ดแสดงข้อมูลได้ โดยได้สร้างบอร์ดแสดงข้อมูลจำลองบนหน้าจอคอมพิวเตอร์ไว้ ซึ่งสามารถสรุปการทำงานของโปรแกรมในส่วนนี้ได้ดังนี้

3.1 โหมดการทำงานเริ่มต้น

ใช้ในการออกแบบรูปภาพ หรือพิมพ์ตัวอักษรจำนวนไม่เกินขนาดความกว้างของบอร์ดแสดงผล และจัดเก็บเป็นแฟ้มข้อมูลเพื่อส่งไปยังไมโครคอนโทรลเลอร์ เพื่อนำออกแสดงเป็นภาพนิ่ง

3.2 โหมดข้อความต่อเนื่อง

ใช้ในการพิมพ์ข้อความยาว ๆ ที่มีตัวอักษรไม่เกิน 414 ตัวอักษร เพื่อจัดเก็บเป็นแฟ้มข้อมูล และส่งไปยังไมโครคอนโทรลเลอร์เพื่อนำออกแสดงโดยให้เคลื่อนที่จากทางขวาของบอร์ดไปทางซ้ายของบอร์ด ซึ่งหลักการได้อธิบายไว้ในบทที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โหมดภาพเคลื่อนไหว

ใช้ในการสร้างภาพจำนวนหลาย ๆ เฟรม เพื่อนำออกแสดงเป็นภาพนิ่งอย่างต่อเนื่องเพื่อให้เห็นเป็นภาพเคลื่อนไหว โดยสามารถจัดเก็บรูปภาพทุกเฟรมอยู่ในแฟ้มข้อมูลเดียวกันได้ แล้วทำการส่งไปยังไมโครคอนโทรลเลอร์ต่อไป

3.4 โหมดการแสดงข้อมูลใน 4 ทิศทาง

ใช้ในการนำข้อมูลที่ได้จัดทำไว้แล้วในโหมดการทำงานเริ่มต้น มาจัดทำให้เป็นข้อมูลรูปภาพหลาย ๆ เฟรมเพื่อจัดส่งไปยังไมโครคอนโทรลเลอร์ให้นำรูปภาพแต่ละเฟรมออกแสดงเป็นภาพนิ่งอย่างต่อเนื่อง ซึ่งจะทำให้เห็นข้อมูลเคลื่อนที่ได้ ซึ่งผู้ใช้สามารถกำหนดได้ว่า จะจัดทำข้อมูลเพื่อให้เห็นเคลื่อนที่บนบอร์ดได้ 4 ทิศทาง คือ จากบนลงล่าง จากล่างขึ้นบน จากซ้ายไปขวา และจากขวาไปซ้าย

โดยวิธีการใช้งานโปรแกรมนี้ ได้อธิบายไว้ในภาคผนวก จ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- Z80180 MICROPROCESSOR DATA BOOK TECHNICAL MANUAL, Zilog, 1988
- ET-DEBUGGER AT180, ETT Co.,Ltd.
- ธันวา ศรีประโมง, "การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม", พิมพ์ครั้งที่ 3, โครงงานตำราวิชาการ มหาวิทยาลัยเทคโนโลยีมหานคร, 2521
- ELECTRONICS DEVICES, Prentice Hall, 1988
- MOTOROLA FAST AND LS TTL DATA, Motorola Inc., 1986

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

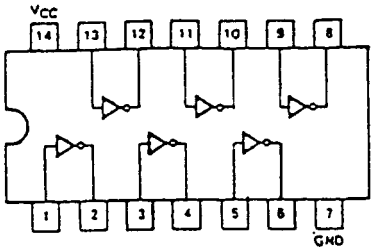
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก.
ข้อมูลทางเทคนิคของอุปกรณ์บนบอร์ดแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54LS04
SN74LS04



J Suffix — Case 632-07 (Ceramic)
N Suffix — Case 646-05 (Plastic)

HEX INVERTER

LOW POWER SCHOTTKY

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs
		74			0.8		
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table
		74	2.7	3.5		V	
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 4.0 mA; V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V
I_{OS}	Short Circuit Current		−20		−100	mA	V_{CC} = MAX
I_{CC}	Power Supply Current Total Output HIGH				2.4	mA	V_{CC} = MAX
	Total Output LOW				6.6		

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH}	Turn Off Delay, Input to Output		8.0	15	ns	V_{CC} = 5.0 V
t_{PHL}	Turn On Delay, Input to Output		10	15	ns	C_L = 15 pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54LS138
SN74LS138

1-OF-8-DECODER/ DEMULTIPLEXER

LOW POWER SCHOTTKY

DESCRIPTION — The LSTTL/MSI SN54LS/74LS138 is a high speed 1-of-8 Decoder/Demultiplexer. This device is ideally suited for high speed bipolar memory chip select address decoding. The multiple input enables allow parallel expansion to a 1-of-24 decoder using just three LS138 devices or to a 1-of-32 decoder using four LS138s and one inverter. The LS138 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

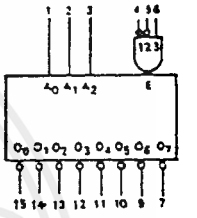
- DEMULTIPLEXING CAPABILITY
- MULTIPLE INPUT ENABLE FOR EASY EXPANSION
- TYPICAL POWER DISSIPATION OF 32 mW
- ACTIVE LOW MUTUALLY EXCLUSIVE OUTPUTS
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

PIN NAMES

		LOADING (Note a) HIGH	LOW
$A_0 - A_2$	Address Inputs	0.5 U.L.	0.25 U.L.
$\overline{E}_1, \overline{E}_2$	Enable (Active LOW) Inputs	0.5 U.L.	0.25 U.L.
E_3	Enable (Active HIGH) Input	0.5 U.L.	0.25 U.L.
$\overline{O}_0 - \overline{O}_7$	Active LOW Outputs (Note b)	10 U.L.	6.2.5 U.L.

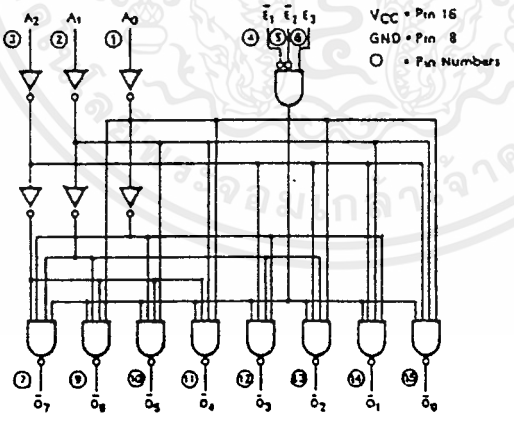
NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges

LOGIC SYMBOL

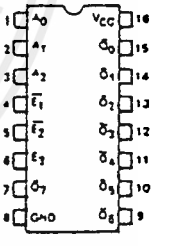


V_{CC} = Pin 16
GND = Pin 8

LOGIC DIAGRAM



CONNECTION DIAGRAM DIP (TOP VIEW)



J Suffix — Case 620-08 (Ceramic)
N Suffix — Case 648-05 (Plastic)

NOTE:
The Flatpak version has the same pinouts (Connection Diagram) as the Dual-In-Line Package

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTIONAL DESCRIPTION — The LS138 is a high speed 1-of-8 Decoder/Demultiplexer fabricated with the low power Schottky barrier diode process. The decoder accepts three binary weighted inputs (A_0, A_1, A_2) and when enabled provides eight mutually exclusive active LOW outputs ($\overline{O}_0$-$\overline{O}_7$). The LS138 features three Enable inputs, two active LOW ($\overline{E}_1$, $\overline{E}_2$) and one active HIGH (E_3). All outputs will be HIGH unless $\overline{E}_1$ and $\overline{E}_2$ are LOW and E_3 is HIGH. This multiple enable function allows easy parallel expansion of the device to a 1-of-32 (5 lines to 32 lines) decoder with just four LS138s and one inverter. (See Figure a.)

The LS138 can be used as an 8-output demultiplexer by using one of the active LOW Enable inputs as the data input and the other Enable inputs as strobes. The Enable inputs which are not used must be permanently tied to their appropriate active HIGH or active LOW state.

TRUTH TABLE

INPUTS						OUTPUTS							
$\overline{E}_1$	$\overline{E}_2$	E_3	A_0	A_1	A_2	$\overline{O}_0$	$\overline{O}_1$	$\overline{O}_2$	$\overline{O}_3$	$\overline{O}_4$	$\overline{O}_5$	$\overline{O}_6$	$\overline{O}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

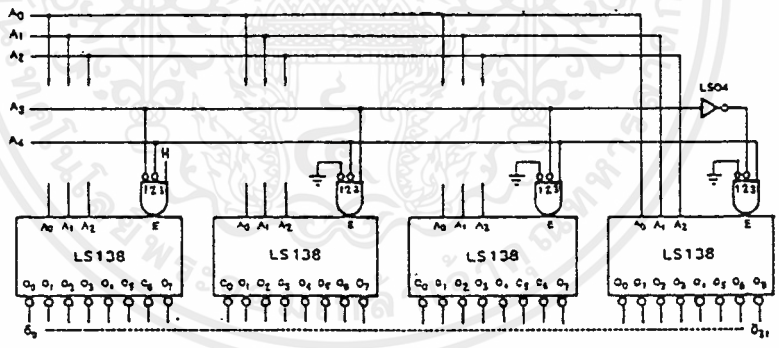


Fig. a.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS/74LS138

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.7	3.5		V		
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 4.0 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current		−20		−100	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				10	mA	V_{CC} = MAX	

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LEVEL OF DELAY	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH} t_{PHL}	Propagation Delay Address to Output	2 2		13 27	20 41	ns	V_{CC} = 5.0 V C_L = 15 pF
t_{PLH} t_{PHL}	Propagation Delay Address to Output	3 3		18 26	27 39	ns	
t_{PLH} t_{PHL}	Propagation Delay $\overline{E}_1$ or $\overline{E}_2$ Enable to Output	2 2		12 21	18 32	ns	
t_{PLH} t_{PHL}	Propagation Delay E_3 Enable to Output	3 3		17 25	26 38	ns	

AC WAVEFORMS

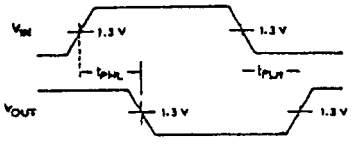


Fig. 1

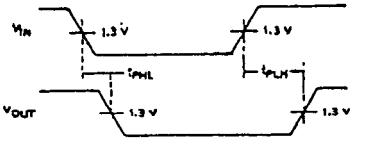


Fig. 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54LS139
SN74LS139

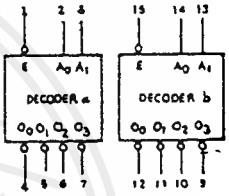
DUAL 1-OF-4 -DECODER/ DEMULTIPLEXER

LOW POWER SCHOTTKY

DESCRIPTION — The LSTTL/MSI SN54LS/74LS139 is a high speed Dual 1-of-4 Decoder/Demultiplexer. The device has two independent decoders, each accepting two inputs and providing four mutually exclusive active LOW Outputs. Each decoder has an active LOW Enable input which can be used as a data input for a 4-output demultiplexer. Each half of the LS139 can be used as a function generator providing all four minterms of two variables. The LS139 is fabricated with the Schottky barrier diode process for high speed and is completely compatible with all Motorola TTL families.

- SCHOTTKY PROCESS FOR HIGH SPEED
- MULTIFUNCTION CAPABILITY
- TWO COMPLETELY INDEPENDENT 1-OF-4 DECODERS
- ACTIVE LOW MUTUALLY EXCLUSIVE OUTPUTS
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

LOGIC SYMBOL



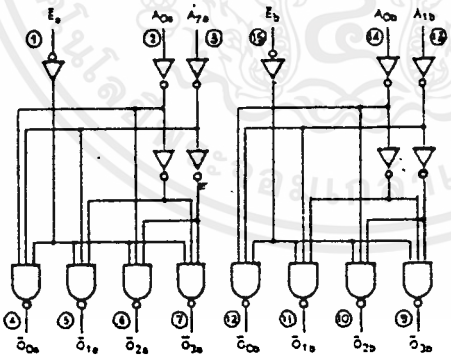
VCC = Pin 16
GND = Pin 8

PIN NAMES

		LOADING (Note a) HIGH	LOW
A_0, A_1	Address Inputs	0.5 U.L.	0.25 U.L.
$\overline{E}$	Enable (Active LOW) Input	0.5 U.L.	0.25 U.L.
$\overline{O}_0 - \overline{O}_3$	Active LOW Outputs (Note b)	10 U.L.	5 (2.5) U.L.

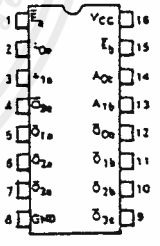
NOTES:
a. 1 TTL Unit Load (U.L.) = 40 µA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 2.5 U.L. for Military (54) and 5 U.L. for Commercial (74) Temperature Ranges.

LOGIC DIAGRAM



VCC = Pin 16
GND = Pin 8
O = Pin Numbers

CONNECTION DIAGRAM DIP (TOP VIEW)



J Suffix — Case 620-08 (Ceramic)
N Suffix — Case 648-05 (Plastic)

NOTE
The Flatpak version has the same pinouts (Connection Diagram) as the Dual-In-Line Package.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS/74LS139

FUNCTIONAL DESCRIPTION — The LS139 is a high speed dual 1-of-4 decoder demultiplexer fabricated with the Schottky barrier diode process. The device has two independent deposers each of which accept two binary weighted inputs (A_0, A_1) and provide four mutually exclusive active LOW outputs ($\bar{O}_0$-$\bar{O}_3$). Each decoder has an active LOW Enable ($\bar{E}$). When $\bar{E}$ is HIGH all outputs are forced HIGH. The enable can be used as the data input for a 4-output demultiplexer application.

Each half of the LS139 generates all four minterms of two variables. These four minterms are useful in some applications, replacing multiple gate functions as shown in Fig. a, and thereby reducing the number of packages required in a logic network.

TRUTH TABLE

INPUTS			OUTPUTS			
$\bar{E}$	A_0	A_1	$\bar{O}_0$	$\bar{O}_1$	$\bar{O}_2$	$\bar{O}_3$
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	H	L	H	L	H	H
L	L	H	H	H	L	H
L	H	H	H	H	H	L

H = HIGH Voltage Level
L = LOW Voltage Level
X = Don't Care

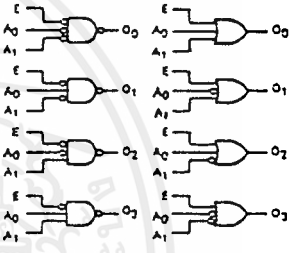


Fig. a

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−0.4	mA
I_{OL}	Output Current — Low	54 74			4.0 8.0	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS/74LS139

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54	2.5	3.5		V	V_{CC} = MIN, I_{OH} = MAX, V_{IN} = V_{IH} or V_{IL} per Truth Table	
		74	2.7	3.5		V		
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 4.0 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 8.0 mA	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Short Circuit Current		−20		−100	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current				11	mA	V_{CC} = MAX	

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LEVEL OF DELAY	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PLH} t_{PHL}	Propagation Delay Address to Output	2 2		13 22	20 33	ns	V_{CC} = 5.0 V C_L = 15 pF
t_{PLH} t_{PHL}	Propagation Delay Address to Output	3 3		18 25	29 38	ns	
t_{PLH} t_{PHL}	Propagation Delay Enable to Output	2 2		16 21	24 32	ns	

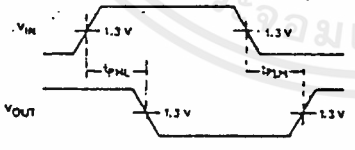


Fig. 1

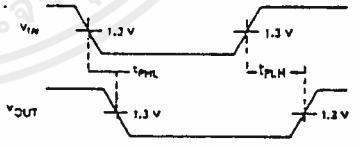


Fig. 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54LS145
SN74LS145

1-OF-10 DECODER/DRIVER OPEN-COLLECTOR
LOW POWER SCHOTTKY

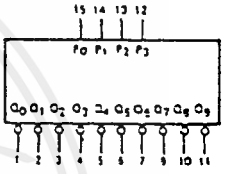
DESCRIPTION — The SN54LS/74LS145, 1-of-10 Decoder/Driver, is designed to accept BCD inputs and provide appropriate outputs to drive 10-digit incandescent displays. All outputs remain off for all invalid binary input conditions. It is designed for use as indicator/relay drivers or as an open-collector logic circuit driver. Each of the high breakdown output transistors will sink up to 80 mA of current. Typical power dissipation is 35 mW. This device is fully compatible with all TTL families.

- LOW POWER VERSION OF 54/74145
- INPUT CLAMP DIODES LIMIT HIGH SPEED TERMINATION EFFECTS

PIN NAMES		LOADING (Note a)	
		HIGH	LOW
P_0, P_1, P_2, P_3	BCD Inputs	0.5 U.L.	0.25 U.L.
$\overline{O}_0$ to $\overline{O}_9$	Outputs (Note b)	Open Collector	15 (7.5) U.L.

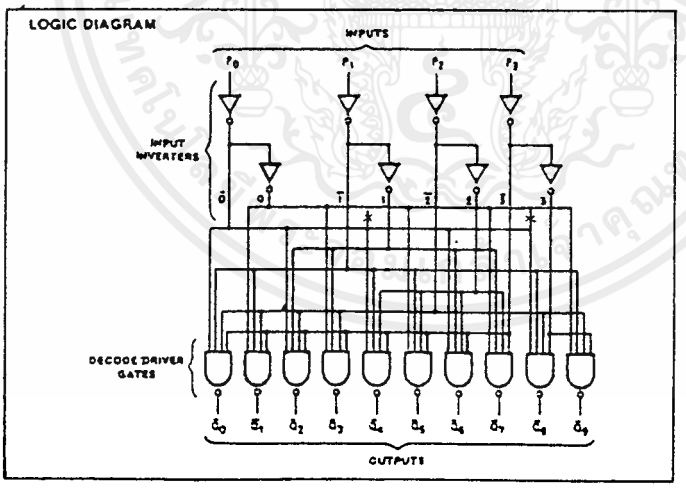
NOTES:
a. 1 TTL Unit Load (U.L.) = 40 μA HIGH/1.6 mA LOW.
b. The Output LOW drive factor is 7.5 U.L. for Military (54) and 15 U.L. for Commercial (74) Temperature Ranges

LOGIC SYMBOL

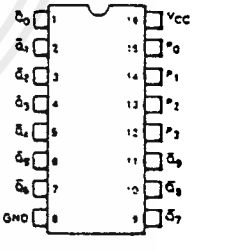


V_{CC} = Pin 16
GND = Pin 8

LOGIC DIAGRAM



CONNECTION DIAGRAM DIP (TOP VIEW)



J Suffix — Case 620-08 (Ceramic)
N Suffix — Case 648-05 (Plastic)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54LS/74LS145

TRUTH TABLE

INPUTS				OUTPUTS									
P_3	P_2	P_1	P_0	$\overline{Q}_0$	$\overline{Q}_1$	$\overline{Q}_2$	$\overline{Q}_3$	$\overline{Q}_4$	$\overline{Q}_5$	$\overline{Q}_6$	$\overline{Q}_7$	$\overline{Q}_8$	$\overline{Q}_9$
L	L	L	L	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	H	L	H	H	H	H	H	H	H
L	L	H	H	H	H	H	L	H	H	H	H	H	H
L	H	L	L	H	H	H	H	L	H	H	H	H	H
L	H	L	H	H	H	H	H	H	L	H	H	H	H
L	H	H	L	H	H	H	H	H	H	L	H	H	H
L	H	H	H	H	H	H	H	H	H	H	L	H	H
H	L	L	L	H	H	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L
H	L	H	L	H	H	H	H	H	H	H	H	H	H
H	L	H	H	H	H	H	H	H	H	H	H	H	H
H	H	L	L	H	H	H	H	H	H	H	H	H	H
H	H	L	H	H	H	H	H	H	H	H	H	H	H
H	H	H	L	H	H	H	H	H	H	H	H	H	H
H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = HIGH Voltage Level
L = LOW Voltage Level

GUARANTEED OPERATING RANGES

SYMBOL	PARAMETER		MIN	TYP	MAX	UNIT
V_{CC}	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T_A	Operating Ambient Temperature Range	54	−55	25	125	°C
		74	0	25	70	
V_{OH}	Output Voltage — High	54, 74			15	V
I_{OL}	Output Current — Low	54			12	mA
		74			24	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

SYMBOL	PARAMETER		LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
I_{OH}	Output HIGH Current	54,74			250	μA	V_{CC} = MIN, V_{OH} = MAX	
V_{OL}	Output LOW Voltage	54,74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
		54,74		2.3	3.0	V	I_{OL} = 80 mA	
I_{IH}	Input HIGH Current				20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
					0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
I_{IL}	Input LOW Current				−0.4	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{CC}	Power Supply Current				13	mA	V_{CC} = MAX, V_{IN} = GND	

AC CHARACTERISTICS: T_A = 25°C

SYMBOL	PARAMETER	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNITS	TEST CONDITIONS
t_{PHL} t_{PLH}	Propagation Delay P_n Input to Q_n Output			50 50	ns	V_{CC} = 5.0 V C_L = 45 pF

AC WAVEFORMS

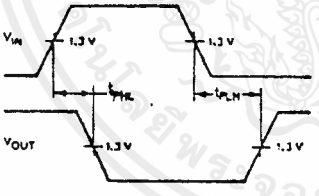


Fig. 1

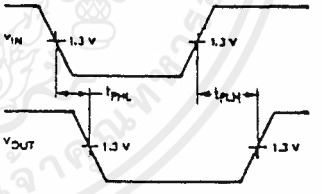


Fig. 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

รีจิสเตอร์ที่สำคัญของไมโครคอนโทรลเลอร์ Z80180

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อทางเอเอสซีไอของ Z80180 นั้น จะใช้รีจิสเตอร์ STAT (ASCI Status Register) เป็นรีจิสเตอร์เก็บค่าสถานะต่าง ๆ เช่น การรับส่งข้อมูลของรีจิสเตอร์ต่าง ๆ ของเอเอสซีไอ และการเปิดทาง (Enable) และปิดทาง (Disable) การอินเทอร์รัพท์ของเอเอสซีไอ เป็นต้น โดยบิตต่าง ๆ ของรีจิสเตอร์ STAT จะบ่งบอกถึงลักษณะต่าง ๆ ดังนี้

ASCI Status Register 0 (STAT0 : I/O Address = 04H)

bit	7	6	5	4	3	2	1	0
	RDRF	OVRN	PE	FE	RIE	$\overline{DCD}_0$	TDRE	TIE
	R	R	R	R	R/W	R	R	R/W

ASCI Status Register 1 (STAT1 : I/O Address = 05H)

bit	7	6	5	4	3	2	1	0
	RDRF	OVRN	PE	FE	RIE	CTS1E	TDRE	TIE
	R	R	R	R	R/W	R/W	R	R/W

รูปที่ ข-1 แสดงบิตต่าง ๆ ของ ASCI Status Register

RDRF : Receive Data Register Full (bit 7)

บิตนี้จะถูกเซ็ทเป็น 1 เมื่อมีข้อมูลถูกโหลดเข้ามาเก็บในรีจิสเตอร์ RDR แต่เมื่อต้องการจะอ่านข้อมูลจากรีจิสเตอร์ RDR บิตนี้จะถูกเคลียร์เป็น 0

OVRN : Overrun Error (bit 6)

บิตนี้จะถูกเซ็ทเป็น 1 เมื่อมีข้อมูลเข้ามาในขณะที่รีจิสเตอร์ RDR มีข้อมูลอยู่แล้ว และรีจิสเตอร์ RSR ก็มีข้อมูลอยู่แล้ว และบิตนี้จะถูกเคลียร์เป็น 0 ในช่วงการรีเซ็ท โดยที่บิต EFR (Error Flag Reset) ของรีจิสเตอร์ CNTLA (ASCI Control Register A) จะต้องถูกกำหนดให้มีค่าเป็น 0 และสัญญาณ DCD_0 เป็น 1 (High) ในขณะที่ Z80180 ทำงานในโหมด IOSTOP

PE : Parity Error (bit 5)

บิตนี้จะถูกเซ็ทเป็น 1 เมื่อเกิดความผิดพลาดภาวะเสมอมูล (Parity Error) กับข้อมูลที่กำลังเข้ามาในขณะที่ส่วนป้องกันความผิดพลาดชนิดนี้ (ASCI Parity Detection) ทำงาน (โดยการเซ็ทบิต MOD1 ของรีจิสเตอร์ CNTLA ให้มีค่าเป็น 1) และบิตนี้จะถูกเคลียร์เป็น 0 ในช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรีเซ็ท โดยที่บิต EFR ของรีจิสเตอร์ CNTLA จะถูกกำหนดให้มีค่าเป็น 0 และสัญญาณ DCD_0 เป็น 1 (High) ในขณะที่ Z80180 ทำงานในโหมด IOSTOP

FE : Framing Error (bit 4)

บิตนี้จะถูกเซ็ทเป็น 1 เมื่อไม่มีการจำกัดขนาดของข้อมูลที่รับเข้ามา จนทำให้ข้อมูลเกินขนาด (Framing Error) ซึ่งเกิดขึ้นจากสต็อบบิต (Stop Bit) ถูกกำหนดค่าผิด กล่าวคือ ถูกกำหนดเป็น 0 ทั้งที่ควรเป็น 1 และบิต FE นี้จะถูกเคลียร์เป็น 0 ในช่วงการรีเซ็ท โดยที่บิต EFR ของรีจิสเตอร์ CNTLA จะต้องถูกกำหนดให้มีค่าเป็น 0 และสัญญาณ DCD_0 เป็น 1 (high) ในขณะที่ Z80180 ทำงานในโหมด IOSTOP

RIE : Receive Interrupt Enable (bit 3)

บิตนี้จะถูกกำหนดให้มีค่าเป็น 1 เพื่อทำให้เอเอสซีไอสามารถสร้างสัญญาณอินเทอร์รัพท์ได้ โดยสัญญาณอินเทอร์รัพท์จะเกิดขึ้นได้ก็ต่อเมื่อ

(ก) บิตนี้ถูกเซ็ทเป็น 1 และบิตต่าง ๆ อันได้แก่ บิต RDRF, บิต OVRN, บิต PE หรือบิต FE ถูกเซ็ทเป็น 1

(ข) มีการเปลี่ยนแปลงค่าของสัญญาณอินพุทภายนอก DCD_0 จาก 0 (Low) เป็น 1 (High)

บิต RIE นี้จะถูกเคลียร์เป็น 0 ในช่วงการรีเซ็ท

$\overline{DCD}_0$: Data Carrier Detect (bit 2 STAT0)

สำหรับเอเอสซีไอ แชนเนล 0 นั้น จะมีสัญญาณอินพุท DCD_0 จากภายนอกเข้ามาที่ขา $\overline{DCD}_0$ นั่นคือบิต $\overline{DCD}_0$ จะมีค่าเป็น 1 เมื่อค่าที่ขา $\overline{DCD}_0$ เป็น 1 (High) ซึ่งก็คือ เมื่อสัญญาณอินพุท DCD_0 จากภายนอกมีค่าเป็น 0 (Low) และบิตนี้จะถูกเคลียร์เป็น 0 ในช่วงแรกของการอ่านค่าจากรีจิสเตอร์ STAT0 ในขณะที่ค่าที่ขา $\overline{DCD}_0$ เปลี่ยนจาก 1 (High) ไปเป็น 0 (Low) และในช่วงรีเซ็ท สำหรับค่าที่ขา $\overline{DCD}_0$ เป็น 1 (High) หรือสัญญาณอินพุท DCD_0 จากภายนอกมีค่าเป็น 0 (Low) จะทำให้หน่วยรับข้อมูลของเอเอสซีไอถูกรีเซ็ท ซึ่งทำให้ไม่สามารถรับข้อมูลจากคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CTS1E : Channel 1 $\overline{CTS}$ Enable (bit 2 STAT1)

เอเอสซีไอ แชนเนล 1 จะมีขา $\overline{CTS}_1$/RXS รับสัญญาณจากภายนอก โดยถ้า CTS1E ถูกเซ็ทเป็น 1 จะเป็นการเลือกใช้ฟังก์ชั่น $\overline{CTS}_1$ แต่ถ้าต้องการใช้ฟังก์ชั่น RXS จากขา $\overline{CTS}_1$/RXS ทำได้โดยการเคลียร์บิต CTS1E ให้เป็น 0

TDRE : Transmit Data Register Empty (bit 1)

เมื่อบิตนี้มีค่าเป็น 1 จะแสดงว่า ไม่มีข้อมูลอยู่ในรีจิสเตอร์ TDR (Transmit Data Register) และหลังจากนั้น เมื่อมีข้อมูลไบต์ถัดไปที่จะส่ง ข้อมูลนั้นก็จะถูกนำไปเก็บในรีจิสเตอร์ TDR และเมื่อไบต์ของข้อมูลถูกเขียนลงไปในรีจิสเตอร์ TDR แล้ว บิตนี้จะถูกเคลียร์เป็น 0 จนกระทั่งข้อมูลจากรีจิสเตอร์ TDR ถูกส่งไปยังรีจิสเตอร์ TSR (Transmit Shift Register) ดังนั้นบิตนี้ก็จะถูกเซ็ทเป็น 1 อีกครั้งหนึ่ง บิตนี้จะถูกเซ็ทเป็น 1 ในช่วงของการรีเซ็ท ในขณะที่ Z80180 ทำงานอยู่ในโหมด IOSTOP และจะถูกเซ็ทเป็น 0 ถ้าสัญญาณภายนอก $\overline{CTS}$ มีค่าเป็น 1 (High)

TIE : Transmit Interrupt Enable (bit 0)

บิตจะถูกเซ็ทเป็น 1 เพื่อทำให้เอเอสซีไอสามารถสร้างสัญญาณอินเทอร์รัพท์ในการส่งข้อมูลได้ โดยถ้าบิตนี้มีค่าเป็น 1 สัญญาณอินเทอร์รัพท์จะเกิดขึ้นเมื่อบิต TDRE มีค่าเป็น 1 ด้วย และบิต TIE นี้จะถูกเคลียร์เป็น 0 ในช่วงของการรีเซ็ท

นอกจากนั้นเอเอสซีไอยังแบ่งโหมดการทำงานต่าง ๆ เช่น โหมดของการเป็นตัวรับข้อมูล(Receiver) หรือโหมดการเป็นตัวส่งข้อมูล (Transmitter) ซึ่งโหมดการทำงานต่าง ๆ จะสามารถกำหนดได้ด้วยรีจิสเตอร์ CNTLA0, 1 (ASCI Control Register A 0, 1) ซึ่งมีบิตต่าง ๆ ดังแสดงในรูปที่ ข-2 และมีรายละเอียดของบิตต่าง ๆ ดังต่อไปนี้

MPE : Multiprocessor Mode Enable (bit 7)

การทำงานของโหมดนี้จะอาศัยบิตของข้อมูลพิเศษเพื่อช่วยเลือกหน่วยประมวลผล (Processor) ที่จะใช้งาน ในกรณีที่มีหน่วยประมวลผลหลายหน่วย (Multiprocessor) แต่ใช้บัสอนุกรม (Serial Bus) ร่วมกัน ถ้า MPE มีค่าเป็น 1 ในขณะที่บิต MPB (Multiprocessor Bit) มีค่าเป็น 1 จะทำให้ข้อมูลไบต์ต่าง ๆ ที่รับเข้ามาเท่านั้น ที่จะมีผลต่อบิต RDRF (Receive Data Register Full) และแฟลกแสดงข้อผิดพลาด (Error Flag) ต่าง ๆ แต่ถ้าบิต MPB มีค่าเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเอสซีไอจะไม่สนใจข้อมูลที่รับเข้ามาเหล่านั้น และบิต MPE นี้จะถูกเคลียร์เป็น 0 ในช่วงการรีเซ็ท

ASCI Control Register A 0 (CNTLA0 : I/O Address = 00H)

bit	7	6	5	4	3	2	1	0
	MPE	RE	TE	$\overline{RTS}_0$	MPBR/EFR	MOD2	MOD1	MOD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ASCI Control Register A 1 (CNTLA1 : I/O Address = 01H)

bit	7	6	5	4	3	2	1	0
	MPE	RE	TE	CKA1D	MPBR/EFR	MOD2	MOD1	MOD0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

รูปที่ ข-2 แสดงบิตต่าง ๆ ของ ASCI Control Register A

RE : Receiver Enable (bit 6)

บิตนี้จะถูกกำหนดค่าเป็น 1 เพื่อกำหนดให้เอเอสซีไอสามารถทำหน้าที่เป็นตัวรับข้อมูล (Receiver) ได้ แต่ถ้าบิตนี้ถูกรีเซ็ทเป็น 0 กระบวนการในการรับข้อมูลของเอเอสซีไอในทุกส่วนจะหยุดชะงัก แต่ข้อมูลที่บิต RDRF และค่าบิตแสดงข้อผิดพลาดต่าง ๆ จะยังคงเดิม บิต RE นี้จะถูกเคลียร์ในช่วงของการรีเซ็ท ในขณะที่ Z80180 ทำงานในโหมด IOSTOP

TE : Transmitter Enable (bit 5)

ถ้าบิตนี้มีค่าเป็น 1 จะทำให้เอเอสซีไอสามารถทำหน้าที่เป็นตัวส่งข้อมูล (Transmitter) ได้ แต่ถ้า TE ถูกรีเซ็ทเป็น 0 กระบวนการของการส่งข้อมูลในทุกส่วนของเอเอสซีไอจะหยุดลง แต่ข้อมูลที่บิต TDRE จะยังคงเดิม และบิต TE นี้จะถูกเคลียร์เป็น 0 ในช่วงของการรีเซ็ท ในขณะที่ Z80180 ทำงานในโหมดของ IOSTOP

$\overline{RTS}_0$: Request to Send Channel 0 (bit 4 in CNTLA0)

เมื่อ $\overline{RTS}_0$ ถูกรีเซ็ทเป็น 0 ค่าที่ขาเอาท์พุท $\overline{RTS}_0$ จะเป็น 0 (Low) แต่ถ้า $\overline{RTS}_0$ ถูกเซ็ทเป็น 1 ค่าที่ขาเอาท์พุท $\overline{RTS}_0$ จะเป็น 1 (High) โดยบิต $\overline{RTS}_0$ จะถูกเซ็ทเป็น 1 ในช่วงการรีเซ็ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CKA1D : CKA1 Clock Disable (bit 4 in CNTLA1)

บิตนี้ถูกเซ็ทเป็น 1 เพื่อกำหนดการใช้ฟังก์ชั่น $\overline{TEND}_0$ ของขา CKA1/$\overline{TEND}_0$ และในทางกลับกัน ถ้าบิตนี้ถูกรีเซ็ทเป็น 0 จะแสดงถึงการใช้ฟังก์ชั่น CKA1 ของขา CKA1/$\overline{TEND}_0$ ซึ่งเป็นฟังก์ชั่นของสัญญาณนาฬิกาของข้อมูลจากภายนอกสำหรับเอเอสซีไอ แชนเนล 1 โดยบิต CKA1D นี้จะถูกเคลียร์เป็น 0 ในช่วงการรีเซ็ท

MPBR/EFR : Multiprocessor Bit Receive / Error Flag Reset (bit 3)

ถ้ากำหนดให้มีค่าเป็น 1 จะทำให้ค่าของบิต MPB ที่ได้รับในช่วงของการรับข้อมูลครั้งล่าสุดถูกนำมาเก็บไว้ในบิตนี้ โดยที่เอเอสซีไอจะต้องทำงานในโหมดมัลติโปรเซสเซอร์ (Multiprocessor) ด้วย แต่ถ้าต้องการใช้ฟังก์ชั่น EFR เราจะต้องรีเซ็ทบิต MPBR/EFR ให้เป็น 0 ซึ่งจะเป็นการรีเซ็ทค่าบิตต่าง ๆ อันได้แก่ บิต OVRN, บิต FE หรือบิต PE ให้มีค่าเป็น 0 ส่วนช่วงของการรีเซ็ท ค่าของบิตนี้จะไม่แสดงความหมายใด (Undefined)

MOD2, 1, 0 : ASCI Data Format Mode 2, 1, 0 (bits 2-0)

ในแต่ละบิตจะบ่งบอกถึงรูปแบบของข้อมูลในรูปแบบต่าง ๆ ดังนี้

MOD2

MOD2 = 0 หมายถึง ข้อมูลมีขนาด 7 บิต

MOD2 = 1 หมายถึง ข้อมูลมีขนาด 8 บิต

MOD1

MOD1 = 0 หมายถึง ไบต์ข้อมูลจะไม่มีพาริตี้ (Parity)

MOD1 = 1 หมายถึง ไบต์ข้อมูลจะมีพาริตี้ (Parity)

MOD0

MOD0 = 0 หมายถึง ข้อมูลมีสต็อบบิต (Stop Bit) จำนวน 1 บิต

MOD0 = 1 หมายถึง ข้อมูลมีสต็อบบิต (Stop Bit) จำนวน 2 บิต

จากค่าต่าง ๆ ของบิต MOD2, MOD1 และ MOD0 จะทำให้ได้รูปแบบของข้อมูลทั้งหมดดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOD2	MOD1	MOD0	รูปแบบของข้อมูล (Data Format)
0	0	0	Start bit + 7 data bits + 1 stop bit
0	0	1	Start bit + 7 data bits + 2 stop bits
0	1	0	Start bit + 7 data bits + parity bit + 1 stop bit
0	1	1	Start bit + 7 data bits + parity bit + 2 stop bits
1	0	0	Start bit + 8 data bits + 1 stop bit
1	0	1	Start bit + 8 data bits + 2 stop bits
1	1	0	Start bit + 8 data bits + parity bit + 1 stop bit
1	1	1	Start bit + 8 data bits + parity bit + 2 stop bits

ตารางที่ ช-1 แสดงรูปแบบต่าง ๆ ของข้อมูล เมื่อค่าที่บิต MOD2, 1, 0 แตกต่างกันไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.
รายละเอียดโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์</u>

โปรแกรมในส่วนนี้จะเขียนด้วยภาษาแอสเซมบลี (Assembly) โดยแบ่งออกเป็น 2 โปรแกรมหลัก ๆ คือ โปรแกรมรับข้อมูลจากคอมพิวเตอร์ และโปรแกรมส่งข้อมูลไปแสดงผลที่บอร์ดแสดงข้อมูล ซึ่งจะมี Source Code ดังต่อไปนี้

ค-1. โปรแกรมรับข้อมูลจากคอมพิวเตอร์ (RECEIVE.ASM)

```
                .org    8000h
                .equ    conta1,01h
                .equ    contb1,03h
                .equ    status1,05h
                .equ    rdr1,09h
                .equ    il,33h
                ld      hl,8400h

; set Control Register B 1
                ld      a,21h           ; set BAUD Rate=9600
                out0    (contb1),a

; set Control Register A 1
                ld      a,44h           ; set RE=Enable
                                        ; set 1 Start,8 Data,1 Stop bit
                out0    (conta1),a

; set Status Register 1
                ld      a,08h           ; set interrupt enable
                out0    (status1),a

; set I Register
                ld      a,90h
                ld      i,a

; set IL Register
                ld      a,30h
                out0    (il),a

                EI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
main:           jp      main

; set Address Of Interrupt Service Routine
                .org    9030h
                DI
                in0     a,(rdr1)
                ld      (hl),a
                inc     hl
                EI
                reti

.end
```

ค-2. โปรแกรมส่งข้อมูลไปแสดงผลที่บอร์ดแสดงข้อมูล (DISPLAY.ASM)

```
                .org    9600h
        .db     0f0h,0f1h,0f2h,0f3h,0f4h,0f5h,0f6h,0f7h,0f8h,0f9h
        .db     00fh,01fh,02fh,03fh,04fh,05fh,06fh,07fh,08fh,09fh

                .org    8000h
                .equ    port_a,0e0h
                .equ    port_b,0e1h
                .equ    port_d,0e3h
                .equ    data0,8400h
                .equ    data1,8478h
                .equ    data2,84f0h
                .equ    data3,8568h

        ld      a,80h
        out     (port_d),a      ; set 8255 be output port

main:   ld      hl,9600h
        ld      a,01h
        ld      (9060h),a       ; check port latch
        ld      a,20d
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
loop:       ld    (9050h),a     ; check column

            ld    a,(9060h)
            ld    d,a           ; check 6 port
            ld    ix,data0
            call  board0        ; display board 0

            ld    a,(9060h)
            ld    d,a           ; check 6 port
            ld    ix,data1
            call  board1        ; display board 1

            ld    a,(9060h)
            ld    d,a           ; check 6 port
            ld    ix,data2
            call  board2        ; display board 2

            ld    a,(9060h)
            ld    d,a           ; check 6 port
            ld    ix,data3
            call  board3        ; display board 3

            ld    d,06h         ; increment data address
            ld    a,(9060h)
            add   a,d
            ld    (9060h),a

            call  scan

            ld    a,(9050h)
            dec   a
            jp    nz,loop
            jp    main

board0:     dec   d
            call  nz,address
            ld    b,00h
            ld    c,06h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
board0_loop:    ld      a,(ix+0)
                out     (port_a),a

                inc     ix
                ld      a,b
                out     (port_b),a
                inc     b

                ld      a,07h
                out     (port_b),a

                ld      a,b
                cp      c
                jp      nz,board0_loop
                ret

board1:         dec     d
                call    nz,address
                ld      b,08h
                ld      c,0eh

board1_loop:    ld      a,(ix+0)
                out     (port_a),a

                inc     ix
                ld      a,b
                out     (port_b),a
                inc     b

                ld      a,0fh
                out     (port_b),a

                ld      a,b
                cp      c
                jp      nz,board1_loop
                ret

board2:         dec     d
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                call    nz,address
                ld      b,10h
                ld      c,16h

board2_loop:    ld      a,(ix+0)
                out     (port_a),a

                inc     ix
                ld      a,b
                out     (port_b),a
                inc     b

                ld      a,17h
                out     (port_b),a

                ld      a,b
                cp      c
                jp      nz,board2_loop
                ret

board3:         dec     d
                call    nz,address
                ld      b,18h
                ld      c,1eh

board3_loop:    ld      a,(ix+0)
                out     (port_a),a

                inc     ix
                ld      a,b
                out     (port_b),a
                inc     b

                ld      a,1fh
                out     (port_b),a

                ld      a,b
                cp      c
                jp      nz,board3_loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ret

scan:           ld      a,(hl)
                out     (port_a),a
                inc     hl
                ld      a,1eh
                out     (port_b),a
                ld      a,1fh
                out     (port_b),a
                call    delay
                ld      a,0ffh
                out     (port_a),a
                ld      a,1eh
                out     (port_b),a
                ld      a,1fh
                out     (port_b),a
                ret

Delay:  ld      b,96h
        ld      a,00h
repeat: dec     a
        jp      nz,repeat
        ret

address:        inc     ix
                dec     d
                jp      nz,address
                ret

.end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง.
รายละเอียดโปรแกรมควบคุมการทำงานของไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>โปรแกรมควบคุมการทำงานของไมโครคอมพิวเตอร์</u>

โปรแกรมในส่วนนี้จะแบ่งเป็น 2 โปรแกรม โดยผู้จัดทำได้ใช้ภาษาซีในการพัฒนาโปรแกรมขึ้นมา แต่ละโปรแกรมจะมีรายละเอียดดังต่อไปนี้

ง-1. PROJECT.EXE

เป็นโปรแกรมที่ทำหน้าที่ควบคุมการทำงานในส่วนการติดต่อกับผู้ใช้เพื่อวาดภาพต่าง ๆ หรือพิมพ์ตัวอักษรจากแป้นพิมพ์ ทำหน้าที่จัดการข้อมูลให้สามารถนำออกแสดงได้หลายรูปแบบ ทำหน้าที่เก็บแฟ้มข้อมูลที่จะแสดงผล และทำหน้าที่ส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ทางพอร์ตอนุกรม โดยโปรแกรมนี้ถูกแปลมาจากโปรแกรม PROJECT.C ด้วยคอมไพเลอร์ของ Turbo C^{++} เวอร์ชั่น 3.0

โปรแกรมนี้จะใช้ควบคุมระบบทั้งหมด ซึ่งจะติดต่อกับผู้ใช้ในรูปแบบทางกราฟฟิก และผู้ใช้สามารถใช้เมาส์ในการควบคุมเป็นหลัก โดยได้จำลองแอลอีดีในทุกตำแหน่งบนบอร์ดแสดงข้อมูล ซึ่งเป็นการจำลองระบบทางฮาร์ดแวร์ จึงทำให้สามารถตรวจสอบการทำงาน และเห็นผลของการทำงานได้อย่างชัดเจน

โปรแกรมนี้สามารถแบ่งออกเป็นส่วนประกอบหลักได้ดังนี้

ง-1-1. ส่วนติดต่อกับผู้ใช้

เป็นส่วนที่ใช้แสดงหน้าจอ การรับคำสั่งจากผู้ใช้ และการทำงานตามคำสั่งนั้น ซึ่งจะประกอบไปด้วยโปรแกรมต่าง ๆ โดยเรากำหนดให้ใช้เมาส์ในการติดต่อกับผู้ใช้ ซึ่งฟังก์ชั่นต่าง ๆ ในการติดต่อกับผู้ใช้จะประกอบด้วยโปรแกรมย่อยต่าง ๆ ที่สำคัญดังนี้

- ฟังก์ชั่นในการติดต่อกับเมาส์

```
void cmouse( )
```

เป็นโปรแกรมหลักในการเรียกใช้เมาส์ ซึ่งจะเรียกใช้โปรแกรมย่อยต่าง ๆ ดังจะกล่าวต่อไปนี้

```
void mouse( )
```

ใช้สำหรับการกำหนดค่าต่าง ๆ ให้เมาส์ซึ่งจะทำการรีเซ็ทเมาส์และตรวจสอบระบบดูว่า มีการติดตั้งเมาส์ไดรเวอร์ (Mouse Driver) ซึ่งเป็นโปรแกรมที่ใช้ในการควบคุมการทำงานของเมาส์ โดยผู้ใช้จะต้องมีการติดตั้งเมาส์ไดรเวอร์ในหน่วยความจำของคอมพิวเตอร์เรียบร้อยแล้ว (MOUSE.COM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void ndelete( )

ใช้สำหรับการกำหนดให้การคลิกเมาส์แต่ละครั้ง เป็นการทำให้หลอดแอลอีดีบนตารางจำลองบนหน้าจอคอมพิวเตอร์ดับ (Blank)

void insert( )

ใช้สำหรับการกำหนดให้การคลิกเมาส์แต่ละครั้ง เป็นการทำให้หลอดแอลอีดีบนตารางจำลองบนหน้าจอคอมพิวเตอร์สว่างเป็นสีตามต้องการ (3 สี)

void enable_cursor( )

ใช้สำหรับการทำให้เมาส์เคอร์เซอร์ (Mouse Cursor) ปรากฏบนหน้าจอคอมพิวเตอร์

void disable_cursor( )

ใช้สำหรับการทำให้เมาส์เคอร์เซอร์ไม่ปรากฏบนหน้าจอคอมพิวเตอร์

void pos_x(int p_x) และ void pos_y(int p_y)

ใช้ตรวจสอบตำแหน่ง (Coordinate) ของเมาส์ เพื่อแสดงตำแหน่งของหลอดแอลอีดีบนตารางจำลองบนหน้าจอคอมพิวเตอร์ เมื่อผู้ใช้ทำการออกแบบรูปภาพ หรือพิมพ์ตัวอักษรบนตารางจำลองบนหน้าจอคอมพิวเตอร์

void x_pos(int x3) และ void y_pos(int y3)

ใช้ตรวจสอบตำแหน่งของเมาส์ เพื่อนำค่าตำแหน่งนั้นไปใช้ในการควบคุมการทำงาน เมื่อผู้ใช้ทำการออกแบบรูปภาพ หรือพิมพ์ตัวอักษรบนตารางจำลองบนหน้าจอคอมพิวเตอร์

void x_pos1(int x3, int *x1, int *x2) และ void y_pos1(int y3, int *y1, int *y2)

ใช้ตรวจสอบตำแหน่งของอักขระที่พิมพ์เป็นข้อความ เมื่อผู้ใช้ต้องการจะพิมพ์ข้อความต่อเนื่องออกแสดงบนบอร์ดแสดงข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ฟังก์ชั่นในการสร้างภาพกราฟฟิกเพื่อติดต่อกับผู้ใช้

void table( )

ทำหน้าที่ในการสร้างบอร์ดแสดงข้อมูลจำลองบนหน้าจอคอมพิวเตอร์ เพื่อให้ผู้ใช้กำหนดค่าของแอลอีดีที่ตำแหน่งต่าง ๆ บนบอร์ดแสดงข้อมูลผ่านเมาส์ หรือพิมพ์ตัวอักษรผ่านทางแป้นพิมพ์

void table1( )

ทำหน้าที่ในการสร้างบอร์ดแสดงข้อมูลจำลอง 2 บอร์ดบนหน้าจอคอมพิวเตอร์ เพื่อให้ผู้ใช้ทำงานในโหมดการแสดงตัวอักษร และรูปภาพใน 4 ทิศทาง

void screen( )

ทำหน้าที่ในการสร้างแผ่นกระดาษจำลองบนหน้าจอคอมพิวเตอร์ เพื่อให้ผู้ใช้พิมพ์ข้อความต่อเนื่องในโหมดการแสดงข้อความต่อเนื่อง

void pos_frame(init *na, init *pos) และ void call(init *p1, init *p2)

ทำหน้าที่ในการสร้างวินโดว์ (Window) เพื่อรับค่าตัวเลขที่เป็นจำนวนเฟรมของข้อมูลที่ผู้ใช้ต้องการนำออกแสดง

void help1( )

ทำหน้าที่ในการสร้างเมนูของคีย์ที่จะทำให้สามารถเข้าสู่โหมดการทำงานหลัก อันได้แก่ การแสดงข้อความต่อเนื่อง (Long-statement Run) การแสดงรูปภาพ และตัวอักษรใน 4 ทิศทาง (4-direction Display) และการแสดงภาพเคลื่อนไหว (Motion Pictures) รวมทั้งคีย์ที่ใช้ในการเปลี่ยนสีของแอลอีดี การเก็บและเรียกแฟ้มข้อมูล

void help2( )

ทำหน้าที่ในการสร้างเมนูของคีย์ที่จะทำให้สามารถเข้าสู่โหมดการทำงานย่อยของโหมดการแสดงรูปภาพ และตัวอักษรใน 4 ทิศทาง อันได้แก่การแสดงข้อมูลจากซ้ายไปขวา จากขวาไปซ้าย จากบนลงล่าง และจากล่างขึ้นบน รวมถึงคีย์ที่ใช้ในการเก็บแฟ้มข้อมูล การเรียกแฟ้มข้อมูล และการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void help3( )

ทำหน้าที่ในการสร้างเมนูของคีย์ที่เป็นฟังก์ชั่นการใช้งานในโหมดการทำงานของโหมดภาพเคลื่อนไหว

void menuon( )

ทำหน้าที่ในการสร้างเมนู Button เพื่อทำการเคลียร์ (ดับ) แอลอีดีในทุกตำแหน่งบนตารางจำลองบนหน้าจอคอมพิวเตอร์ เพื่อทำการเก็บ และเรียกแฟ้มข้อมูล เพื่อกำหนดให้การคลิกเมาส์แต่ละครั้งเป็นการติด หรือดับหลอดแอลอีดีบนตารางจำลองบนหน้าจอคอมพิวเตอร์ รวมทั้งเพื่อออกจากโปรแกรมการทำงาน (PROJECT.EXE)

void table_save(unsigned char name[8])

ทำหน้าที่ในการสร้างวินโดว์ (Window) เพื่อรับตัวอักษรที่เป็นชื่อแฟ้มข้อมูล เมื่อผู้ใช้ต้องการจะเก็บข้อมูลลงแฟ้มข้อมูล หรือเรียกข้อมูลขึ้นมาใช้

void plate1( )

ทำหน้าที่สร้างเมนูของ Button ที่ใช้เมาส์คลิก ในโหมดภาพเคลื่อนไหว ซึ่งจะเป็น Button ในการนำเฟรมรูปภาพที่ออกแบบไว้ ออกแสดงบนหน้าจอคอมพิวเตอร์

- ฟังก์ชั่นในการตรวจสอบ เพื่อตอบรับคำสั่งจากผู้ใช้

void menuoff( )

ทำหน้าที่ในการลบ (ไม่แสดง) เมนูตามฟังก์ชั่น menuon( ) เมื่อตรวจสอบพบว่าผู้ใช้กดคีย์ Esc

void clear( )

ทำหน้าที่เคลียร์ (ดับ) หลอดแอลอีดีในทุกตำแหน่งบนตารางจำลองบนหน้าจอคอมพิวเตอร์ เมื่อผู้ใช้ไม่ต้องการรูปภาพ หรือตัวอักษรที่อยู่บนตารางจำลองบนหน้าจอคอมพิวเตอร์ ซึ่งตรวจสอบได้เมื่อผู้ใช้คลิกเมาส์ที่บริเวณ Button 'CLEAR'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void paper( )

ทำหน้าที่เคลียร์ข้อความทั้งหมดที่อยู่บนแผ่นกระดาษจำลองในโหมดการแสดงข้อความต่อเนื่อง เมื่อตรวจสอบพบว่า ผู้ใช้ได้คลิกเมาส์ที่บริเวณ Button 'CLEAR'

void check(unsigned char *thai_font, int *upp)

ทำหน้าที่ตรวจสอบลำดับการพิมพ์สระซ้อน หรือวรรณยุกต์ซ้อนสระ เพื่อให้ลำดับการเขียนวรรณยุกต์ และสระถูกต้อง เมื่อผู้ใช้พิมพ์ข้อความ

void colour( )

ทำหน้าที่ตรวจสอบ และแสดงถึงสถานะของเมาส์ว่า ในการคลิกเมาส์แต่ละครั้งทำให้แอลอีดีปล่อยแสงสีใด (สีแดง สีเขียว หรือสีส้ม)

void text( )

ทำหน้าที่ตรวจสอบคีย์ว่า ผู้ใช้ต้องการพิมพ์ข้อมูลผ่านแป้นพิมพ์ (โดยการกดคีย์ ^T) หรือไม่ แล้วทำการเปลี่ยนการรับข้อมูลจากเมาส์มาเป็นรับจากแป้นพิมพ์แทน

void keyboard(int point, int point1)

ในโหมดการแสดงข้อความต่อเนื่อง จะใช้ตรวจสอบว่า ผู้ใช้ต้องการจะพิมพ์ข้อความ (โดยการกด ^K) หรือไม่ แล้วทำการไปรับข้อมูลจากแป้นพิมพ์

void load_font( )

ทำหน้าที่ตรวจสอบว่า คีย์ที่ผู้ใช้กดบนแป้นพิมพ์มีรหัสแอสกีตรงกับของรูปภาพแบบบิตแม็ปที่เก็บไว้ตัวใด แล้วทำการเรียกบิตแม็ปนั้นขึ้นมาแสดงบนหน้าจอ โดยรูปภาพแบบบิตแม็ปดังกล่าวนั้น จะถูกเรียกจากแฟ้มข้อมูลชื่อ FONT ดังนั้นเมื่อผู้ใช้ต้องการเปลี่ยนรูปแบบของตัวอักษรก็สามารถทำได้โดยนำแฟ้มข้อมูลของบิตแม็ปของตัวอักษรในรูปแบบที่ต้องการมาทำการเปลี่ยนชื่อ (Rename) ให้เป็น FONT หรืออาจจะทำการออกแบบเองที่โปรแกรม EDIT_FON.EXE ที่เราจัดทำไว้ให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void play( )

ทำหน้าที่ตรวจสอบว่า ถ้าผู้ใช้ได้คลิกเมาส์ที่ Button จะทำการแสดงภาพทุกเฟรมที่ได้ออกแบบไว้ในโหมดภาพเคลื่อนไหวออกทางหน้าจอหน้าคอมพิวเตอร์

void stop1( )

ทำหน้าที่ตรวจสอบว่า ถ้าผู้ใช้ได้คลิกเมาส์ที่ Button จะเป็นการหยุดการแสดงภาพจากการคลิกเมาส์ที่ Button โดยจะไปแสดงภาพในเฟรมสุดท้ายแทน

void step_dec( )

ทำหน้าที่ตรวจสอบว่า ถ้าผู้ใช้ได้คลิกเมาส์ที่ Button จะเป็นการย้อนดูภาพในเฟรมถัดจากเฟรมที่แสดงอยู่

void step_inc( )

ทำหน้าที่ตรวจสอบว่า ถ้าผู้ใช้ได้คลิกเมาส์ที่ Button จะเป็นการเลื่อนไปดูภาพในเฟรมถัดจากเฟรมที่แสดงอยู่

ง-1-2. ส่วนการเก็บข้อมูลลงแฟ้มข้อมูล

ทุกครั้งที่มีการเติมจุดสีลงบนบอร์ดแสดงผลจำลองบนหน้าจอคอมพิวเตอร์ ข้อมูลใน Array จะได้รับการ Update และเมื่อเติมจุดสีเรียบร้อยแล้ว ก็จะเก็บข้อมูลของสีต่าง ๆ ลงใน Array เพื่อเก็บข้อมูลลงแฟ้มข้อมูลต่อไป

void save( )

ทำหน้าที่เก็บข้อมูลลงแฟ้มข้อมูล โดยจะอ่านข้อมูลจาก Array ออกมาทีละไบต์ จากนั้นก็จะมาทำการแปลงข้อมูลให้อยู่ในรูปของเลขฐาน 16 แล้วเก็บข้อมูลลงไฟล์เรียงตามลำดับ

void save_text( )

เมื่อผู้ใช้ทำงานในโหมดการแสดงข้อความต่อเนื่อง และต้องการเก็บข้อมูลที่เป็นข้อความลงแฟ้มข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง-1-3. ส่วนการเรียกข้อมูลตามแฟ้มข้อมูล และการแสดงข้อมูล

เมื่อผู้ใช้ต้องการจะเรียกแฟ้มข้อมูลออกมาใช้งาน หรือแสดงบนหน้าจอคอมพิว-เตอร์ ซึ่งจะมีการเรียกใช้งานโปรแกรมย่อยดังต่อไปนี้

void load_data

เมื่อผู้ใช้ต้องการเรียกแฟ้มข้อมูลของข้อความยาว ๆ ที่ได้จัดเก็บในรูปของแฟ้มข้อมูลไว้แล้วในโหมดข้อความต่อเนื่อง

void not_found( )

ทำหน้าที่ตรวจสอบว่า แฟ้มข้อมูลที่ผู้ใช้ต้องการเรียกขึ้นมานั้น มีอยู่ในหน่วยความจำของคอมพิวเตอร์หรือไม่ โดยถ้าผู้ใช้พิมพ์ชื่อแฟ้มข้อมูลที่ไม่มีอยู่ในหน่วยความจำแล้ว โปรแกรมจะทำการแจ้งให้ผู้ใช้ทราบบนหน้าจอคอมพิวเตอร์ว่า 'File Not Found'

void not_match( )

ทำหน้าที่ตรวจสอบว่า แฟ้มข้อมูลที่ผู้ใช้เรียกขึ้นมานั้น เป็นข้อมูลที่ใช้งานในโหมดนั้นหรือไม่ เช่น การเรียกแฟ้มข้อมูลของข้อความยาวที่พิมพ์ไว้ในโหมดข้อความต่อเนื่อง ขึ้นมาแสดงบนโหมดการแสดงข้อมูลใน 4 ทิศทาง จะทำให้โปรแกรมแจ้งบนหน้าจอคอมพิวเตอร์ว่า 'File Not Match'

ง-1-4. ส่วนการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์

หลังจากเก็บข้อมูลในไฟล์เรียบร้อยแล้ว ก็จะทำการส่งข้อมูลออกทางพอร์ตอนุกรมเพื่อเก็บในหน่วยความจำของไมโครคอนโทรลเลอร์ต่อไป

void send(int p1, int p2)

ทำหน้าที่ในการส่งข้อมูลรูปภาพที่ได้ออกแบบไว้แล้วในโหมดการทำงานเริ่มต้น ไปยังไมโครคอนโทรลเลอร์เพื่อไปแสดงเป็นภาพนิ่งบนบอร์ดแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

void send_page( )

ทำหน้าที่ในการส่งข้อมูลของข้อความยาว ๆ ที่พิมพ์ไว้ในโหมดข้อความต่อเนื่องออกพอร์ตเพื่อส่งไปยังไมโครคอนโทรลเลอร์

ง-1-5. ส่วนการจัดการข้อมูลเพื่อให้แสดงข้อมูลใน 4 ทิศทาง และภาพเคลื่อนไหว

void left_right( ) void shift_left(int col)

เป็นฟังก์ชั่นในการทำให้ข้อมูลแบ่งเป็นหลายเฟรม แต่ไม่เกิน 80 เฟรม เพื่อให้สามารถแสดงข้อมูลจากทางซ้ายของบอร์ดไปทางขวาของบอร์ด

void right_left( ) void shift_right(int col)

เป็นฟังก์ชั่นในการทำให้ข้อมูลแบ่งเป็นหลายเฟรม แต่ไม่เกิน 80 เฟรม เพื่อให้สามารถแสดงข้อมูลจากทางขวาของบอร์ดไปทางซ้ายของบอร์ด

void up_to_down( ) void shift_data(int row)

เป็นฟังก์ชั่นในการทำให้ข้อมูลแบ่งเป็นหลายเฟรม แต่ไม่เกิน 24 เฟรม เพื่อให้สามารถแสดงข้อมูลจากด้านบนของบอร์ดไปด้านล่างของบอร์ด

void down_to_up( ) void shift_data1(int row)

เป็นฟังก์ชั่นในการทำให้ข้อมูลแบ่งเป็นหลายเฟรม แต่ไม่เกิน 24 เฟรม เพื่อให้สามารถแสดงข้อมูลจากด้านล่างของบอร์ดไปด้านบนของบอร์ด

void frame100( )

เป็นฟังก์ชั่นการทำงานในโหมดภาพเคลื่อนไหวในการเก็บข้อมูลลงแฟ้ม การเรียกข้อมูลมาใช้ รวมทั้งการแสดงภาพแต่ละเฟรมออกแสดงบนหน้าจอคอมพิวเตอร์อย่างต่อเนื่อง

สำหรับวิธีการใช้โปรแกรม PROJECT. EXE นี้ ได้อธิบายไว้แล้วในภาคผนวก จ. และ Source Code โปรแกรมภาษาซีของโปรแกรมนี้จะแสดงไว้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<dos.h>
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include<alloc.h>
#include<string.h>
#include<bios.h>

/* structure of data */
typedef struct {
unsigned char up,down;
unsigned char key;
unsigned char convert;
int color;
void *link,*up1;
} data;

typedef struct {
unsigned char data1[80][24];
int end;
void *link,*up;
} emu;

/* ASCII of thai code */
const  thai[95]   = {205,43,46,130,131,132,235,165,134,135,133,137,191,
                     162,211,187,166,206,47,158,190,180,215,219,163,179,169,196,
                     176,168,201,63,129,194,229,167,173,172,210,170,223,177,227,
                     198,197,63,228,214,171,128,175,164,182,226,203,34,41,222,40,
                     184,92,195,216,136,96,189,217,209,161,207,178,208,225,193,
                     224,206,199,181,220,183,192,213,188,200,204,218,202,212,185,
                     221,186,174,124,44,126};

/* Global Parameter */
#define COM2        1
#define DATA_READY 0x100
#define SETTINGS (_COM_9600 | _COM_CHR8 | _COM_STOP1 | _COM_NOPARITY)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int xa1,ya1,xa2,ya2,y2,x2,color,rcolor,chk_name=0,plate=0,plate2=0;
int check_delete,point=0,point1=0,frame=0,h=0,x1,y1,frame1=0,a=0,b=0,e=0;
int output[80][24],list;
char output_font[256][8][24];
emu *simu;
void *temp,*mark;
data *font;
int text_board=0,full=0;

/* Function Call */
void ndelete( );
void insert( );
void disable_cursor( );
void enable_cursor( );
void table_save(unsigned char name[8]);
void send(int p1,int p2);
void initial( );
void table( );
void pos_x(int p_x);
void pos_y(int p_y);
void x_pos(int x3);
void y_pos(int y3);
void menuon( );
void menuoff( );
void save( );
void load( );
void clear( );
void colour( );
void mouse( );
void text( );
void load_font( );
void up_to_down( );
void down_to_up( );
void shift_data(int row);
void shift_data1(int row);
void dis( );
void save_frame( );
void play_4( );
void table1( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void load_frame( );
void x_pos1(int x3,int *x1,int *x2);
void y_pos1(int y3,int *y1,int *y2);
void screen( );
void keyboard(int point,int point1);
void paper( );
void save_text( );
void show( );
void show1( );
void clear_pos(unsigned char ch,int point,int point1);
void clear_pos1(int point,int point1);
void save_page( );
void check(unsigned char *thai_font,int *upp);
void not_found( );
void not_match( );
void help1( );
void help2( );
void init( );
void pos_frame(int *na,int *pos);
void call(int *p1,int *p2);
void left_right( );
void shift_left(int col);
void right_left( );
void shift_right(int col);
void cmouse(int *stop);
void frame100( );
void plate1( );
void play( );
void stop1( );
void step_inc( );
void step_dec( );
void help3( );

/* Main Program */
main( )
{
int gd = DETECT, gm;
        _bios_serialcom(_COM_INIT, COM2, SETTINGS);
        load_font( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            chk_name=0;
            color=4;
            rcolor=4;
            initgraph(&gd,&gm,"egavga.bgi");
            initial( );
            mouse( );
            closegraph( );
return 0;
}

void table_save(unsigned char name[8])
{
char *n1;
char n;
int i=0;
disable_cursor();
setfillstyle(SOLID_FILL,15);
bar(170,130,464,190);
setcolor(5);
line(172,132,462,132);
line(172,132,172,188);
line(172,170,462,170);
line(172,188,462,188);
line(462,132,462,188);
setcolor(17);
outtextxy(350,175,"Enter = accept");
outtextxy(180,175,"Esc = abort");
setcolor(3);
outtextxy(180,148,"Enter your file's name : ");
enable_cursor( );
moverel(20,30);
name[0]='\0';
setcolor(1);
while (n!=13 && n!=27)
{
            n=getch();
            if (n==8 && i>0)
            {
                        i=i-;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                disable_cursor( );
                bar(379+8*i,146,389+8*i,155);
                enable_cursor( );
                name[i]='\0';
        }
        if (n==27) {chk_name=1;}
        if (n!=13 && n!=27 && i<8 && n!=8)
        {
                sprintf(n1,"%c",n);
                disable_cursor( );
                outtextxy(380+8*i,148,n1);
                enable_cursor( );
                strcat(name,n1);
                i=i++;
        }
        if (i>7) {sound(220);delay(33);nosound( );}
}
if (n==13) {chk_name=0;}
}

/* Disable Mouse Cursor */
void disable_cursor( )
{
        asm mov ax,02h;
        asm int 33h;
}

/* Enable Mouse Cursor */
void enable_cursor( )
{
        asm mov ax,01h;
        asm int 33h;
}

/* Show Delete */
void ndelete( )
{
disable_cursor( );
check_delete=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
rcolor=color;
color=3;
setfillstyle(SOLID_FILL,3);
bar(104,308,150,323);
setcolor(9);
outtextxy(104,310,"Delete.");
enable_cursor( );
}

/* Show Insert */
void insert( )
{
disable_cursor( );
color=rcolor;
check_delete=0;
setfillstyle(SOLID_FILL,3);
bar(104,308,150,323);
setcolor(9);
outtextxy(104,310,"Insert.");
enable_cursor( );
}

/* Send Data to Com Port */
void send(int p1,int p2)
{
unsigned char l;
int temp[4],point=0,temp1=0,i,j,k;
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
setfillstyle(SOLID_FILL,15);
bar(170,130,464,190);
setcolor(5);
line(172,132,462,132);
line(172,132,172,188);
line(172,188,462,188);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(462,132,462,188);
setcolor(25);
outtextxy(230,155,"Data is now SENDING ...");
enable_cursor( );
if (frame!=1)
{
        for (i=0;i<80;i++)
        {
                for (j=0;j<24;j++)
                {
                        point=fmod(j,4);
                        l=output[i][j];
                        if (l==3) {temp[point]=0;}
                        if (l==4) {temp[point]=1;}
                        if (l==10) {temp[point]=2;}
                        if (l==12) {temp[point]=3;}
                        if (point==3)
                        {
                                temp1=temp[3]*64+temp[2]*16+temp[1]*4+temp[0];
                                _bios_serialcom(_COM_SEND, COM2, temp1);
                        } //end if
                } //end loop j
        } //end loop i
}
else
{
        while (p1<=p2 && simu->end==0)
        {
                for (i=0;i<80;i++)
                {
                        for (j=0;j<24;j++)
                        {
                                l=simu->data1[i][j];
                                if (l==3) {temp[point]=0;}
                                if (l==4) {temp[point]=1;}
                                if (l==10) {temp[point]=2;}
                                if (l==12) {temp[point]=3;}
                                if (point==3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        {
                                                temp1=temp[3]*64+temp[2]*16+temp[1]*4+temp[0];
                                                _bios_serialcom(_COM_SEND, COM2, temp1);
                                        } //end if
                                }
                        }
                        simu=simu->link;
                        p1=p1++;
                }
}
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
enable_cursor( );
}

/* Set Initial Color */
void initial( )
{
int i,j;
for (i=0;i<80;++i)
for (j=0;j<24;++j)
output[i][j]=3;
}

/* Call Screen Function */
void table( )
{
int i;
setbkcolor(3);
for (i=0;i<81;i++)
{
        if (fmod(i,20)==0)
        {
                setcolor(9);
                line(37+7*i,88,37+7*i,280);
        }
        else
        {
                setcolor(14);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                line(37+7*i,88,37+7*i,280);
        }
}
for (i=0;i<25;i++)
{
        if (fmod(i,4)==0)
        {
                setcolor(9);
                line(37,88+8*i,597,88+8*i);
        }
        else
        {
                setcolor(14);
                line(37,88+8*i,597,88+8*i);
        }
}
setcolor(4);
outtextxy(105,295,"Red.");
setfillstyle(SOLID_FILL,15);
bar(0,0,639,20);
outtextxy(265,8,".LED Display.");
setfillstyle(SOLID_FILL,15);
bar(0,460,639,480);
setcolor(4);
outtextxy(127,468,".King Mongkut Institue of Technology Ladkrabang.");
setcolor(6);
outtextxy(480,310,"Row.");
outtextxy(480,295,"Column.");
outtextxy(45,295,"Colour.");
outtextxy(45,310,"Status.");
setfillstyle(SOLID_FILL,3);
bar(37,40,167,70);
setcolor(10);
rectangle(38,41,166,69);
setcolor(9);
outtextxy(48,52,"Ctrl H = Help.");
insert( );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Call Show X_Position Function */
void pos_x(int p_x)
{
int i=0;
char i1[4];
while (p_x>107)
{
        p_x=p_x-70;
        i=i+1;
}
i=i*10;
if (37<=p_x && p_x<44) {i=i+1;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (44<=p_x && p_x<51) {i=i+2;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (51<=p_x && p_x<58) {i=i+3;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (58<=p_x && p_x<65) {i=i+4;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (65<=p_x && p_x<72) {i=i+5;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (72<=p_x && p_x<79) {i=i+6;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (79<=p_x && p_x<86) {i=i+7;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (86<=p_x && p_x<93) {i=i+8;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (93<=p_x && p_x<100) {i=i+9;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
if (100<=p_x && p_x<=107) {i=i+10;x2=i;itoa(i,i1,10);outtextxy(550,295,i1);}
}

/* Call Show Y_Position Function */
void pos_y(int p_y)
{
int i=0;
char i1[4];
while (p_y>136)
{
        p_y=p_y-48;
        i=i+1;
}
i=i*6;
if (88<=p_y && p_y<96) {i=i+1;y2=i;itoa(i,i1,10);outtextxy(550,310,i1);}
if (96<=p_y && p_y<104) {i=i+2;y2=i;itoa(i,i1,10);outtextxy(550,310,i1);}
if (104<=p_y && p_y<112) {i=i+3;y2=i;itoa(i,i1,10);outtextxy(550,310,i1);}
if (112<=p_y && p_y<120) {i=i+4;y2=i;itoa(i,i1,10);outtextxy(550,310,i1);}
if (120<=p_y && p_y<128) {i=i+5;y2=i;itoa(i,i1,10);outtextxy(550,310,i1);}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (128<=p_y && p_y<=136) {i=i+6;y2=i;itoa(i,i1,10);outtextxy(550,310,i1);}
}

/* Display X Position */
void x_pos(int x3)
{
int i=0;
while (x3>10)
{
        x3=x3-10;
        i=i+1;
}
i=i*70;
if (x3==1) {xa1=38+i;xa2=42+i;}
if (x3==2) {xa1=45+i;xa2=49+i;}
if (x3==3) {xa1=52+i;xa2=56+i;}
if (x3==4) {xa1=59+i;xa2=63+i;}
if (x3==5) {xa1=66+i;xa2=70+i;}
if (x3==6) {xa1=73+i;xa2=77+i;}
if (x3==7) {xa1=80+i;xa2=84+i;}
if (x3==8) {xa1=87+i;xa2=91+i;}
if (x3==9) {xa1=94+i;xa2=98+i;}
if (x3==10) {xa1=101+i;xa2=105+i;}
}

/* Display Y Position */
void y_pos(int y3)
{
int i=0;
while (y3>6)
{
        y3=y3-6;
        i=i+1;
}
i=i*48;
if (y3==1) {ya1=89+i;ya2=94+i;}
if (y3==2) {ya1=97+i;ya2=102+i;}
if (y3==3) {ya1=105+i;ya2=110+i;}
if (y3==4) {ya1=113+i;ya2=118+i;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (y3==5) {ya1=121+i;ya2=126+i;}
if (y3==6) {ya1=129+i;ya2=135+i;}
}

/* Menu Function */
void menuon( )
{
/* Plate of Utilities ON */
disable_cursor( );
setfillstyle(SOLID_FILL,14);
bar(90,350,550,450);
setcolor(6);
rectangle(89,349,551,451);
setfillstyle(SOLID_FILL,10);
setcolor(10);
pieslice(115,375,0,360,5);
outtextxy(130,373,".Green.");
setfillstyle(SOLID_FILL,4);
setcolor(4);
pieslice(115,400,0,360,5);
outtextxy(130,398,".Red.");
setfillstyle(SOLID_FILL,12);
setcolor(12);
pieslice(115,425,0,360,5);
outtextxy(130,423,".Orange.");
setfillstyle(SOLID_FILL,9);
bar(220,370,300,390);
outtextxy(235,375,".Load.");
setfillstyle(SOLID_FILL,9);
bar(340,370,420,390);
outtextxy(353,375,".Clear.");
setfillstyle(SOLID_FILL,9);
bar(460,370,540,390);
outtextxy(469,375,".Insert.");
setfillstyle(SOLID_FILL,9);
bar(220,410,300,430);
outtextxy(235,415,".Save.");
setfillstyle(SOLID_FILL,9);
bar(340,410,420,430);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextxy(356,415,".Quit.");
bar(460,410,540,430);
outtextxy(469,415,".Delete.");
setfillstyle(SOLID_FILL,9);
enable_cursor( );
}

/* Plate of Utilities OFF */
void menuoff( )
{
disable_cursor( );
setfillstyle(SOLID_FILL,3);
bar(89,349,551,451);
enable_cursor( );
}

/* Save Data to File */
void save( )
{
FILE *fp;
char file_name[8];
int temp[4],point=0,temp1=0,i,j;
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
enable_cursor( );
if (chk_name!=1)
{
        fp=fopen(file_name,"wb");
        for (i=0;i<80;i++)
        {
                for (j=0;j<24;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        point=fmod(j,4);
                        if (output[i][j]==3) {temp[point]=0;}
                        if (output[i][j]==4) {temp[point]=1;}
                        if (output[i][j]==10) {temp[point]=2;}
                        if (output[i][j]==12) {temp[point]=3;}
                        if (point==3)
                        {
                                temp1=temp[3]*64+temp[2]*16+temp[1]*4+temp[0];
                                putc(temp1,fp);
                        } //end if
                } //end loop j
        } //end loop i
}
free(backup);
fclose(fp);
chk_name=0;
}

/* Load Data to Display */
void load( )
{
FILE *l;
char file_name[8];
unsigned char temp;
int point,x=0,y=0,decode[8],i=0,j=0,x1,x2,y1,y2;
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
enable_cursor( );
l=fopen(file_name,"rb");
back=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (chk_name!=1)
{
        if (l!=NULL)
        {
                while (!feof(l))
                {
                        if (back<481)
                        {
                                if (j==6) {j=0;i=i++;y=0;}
                                x=7;
                                temp=getc(l);
                                if (temp>=128) {decode[x]=1;temp=temp-128;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=64) {decode[x]=1;temp=temp-64;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=32) {decode[x]=1;temp=temp-32;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=16) {decode[x]=1;temp=temp-16;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=8) {decode[x]=1;temp=temp-8;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=4) {decode[x]=1;temp=temp-4;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=2) {decode[x]=1;temp=temp-2;x=x-;} else {decode[x]=0;x=x-;}
                                if (temp>=1) {decode[x]=1;} else {decode[x]=0;}
                                x=7;
                                for (point=0;point<8;point++)
                                {
                                        temp=decode[point]+decode[point+1];
                                        if (temp==0) {output[i][y]=3;}
                                        if (temp==1 && decode[point]==1) {output[i][y]=4;}
                                        if (temp==1 && decode[point]==0) {output[i][y]=10;}
                                        if (temp==2) {output[i][y]=12;}
                                        point=point+1;
                                        y=y++;
                                }
                                j=j++;
                        }
                        else
                        {
                                temp=getc(l);
                        }
                        back=back+1;
                } // end while
                if (back==481)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        disable_cursor( );
                        for (i=0;i<80;i++)
                        {
                                for (j=0;j<24;j++)
                                {
                                        setfillstyle(SOLID_FILL,output[i][j]);
                                        if (frame!=1)
                                        {
                                                x_pos(i+1);
                                                y_pos(j+1);
                                                bar(xa1,ya1,xa2,ya2);
                                        }
                                        else
                                        {
                                                x_pos1(i+1,&x1,&x2);
                                                y_pos1(j+1,&y1,&y2);
                                                bar(x1,222+y1,x2,222+y2);
                                        }
                                }
                        }
                        enable_cursor( );
                }
                else
                {
                        not_match( );
                }
        }
        else
        {
                not_found( );
        }
        fclose(f);
}
free(backup);
chk_name=0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Clear Screen */
void clear( )
{
int i,j;
disable_cursor( );
setfillstyle(SOLID_FILL,3);
for (i=0;i<80;i++)
for (j=0;j<24;j++)
{
        x_pos(i+1);
        y_pos(j+1);
        output[i][j]=3;
        if (frame1==1) {simu->data1[i][j]=3;}
        bar(xa1,ya1,xa2,ya2);
}
enable_cursor( );
}

/* Color of Led */
void colour( )
{
disable_cursor( );
if (color==4)
{
        setcolor(color);
        setfillstyle(SOLID_FILL,3);
        bar(99,290,160,305);
        outtextxy(105,295,"Red.");
}
if (color==10)
{
        setcolor(color);
        setfillstyle(SOLID_FILL,3);
        bar(99,290,160,305);
        outtextxy(105,295,"Green.");
}
if (color==12)
{
        setcolor(color);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setfillstyle(SOLID_FILL,3);
            bar(99,290,160,305);
            outtextxy(105,295,"Orange.");
}
            enable_cursor( );
}

/* Call Mouse Function */
void mouse( )
{
int open,open1,x,x1,y,y1,stop=0;
char x_str[5],y_str[5],key=0,scan=0;
            /* check mouse and mouse driver */
            asm mov ax,00h;
            asm int 33h;
            asm mov open,ax;
            asm mov ax,35h;
            asm mov open1,ax;
            /* Set mouse sensitivity and Interrupt rate */
            asm mov ax,1ah; // sentivity
            asm mov bx,50d;
            asm mov cx,50d;
            asm mov dx,50d;
            asm int 33h;
            asm mov ax,1ch; // interrupt rate
            asm mov bx,4d;
            asm int 33h;
            /* load mouse cursor and position */
if ((open!=0) && (open1!=0))
{
            table( );
            enable_cursor( );
            asm mov ax,07h;
            asm mov cx,00h;
            asm mov dx,278h;
            asm int 33h;
            asm mov ax,08h;
            asm mov cx,00h;
            asm mov dx,1d0h;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        asm int 33h;
do
{
        while (bioskey(1)==0 && stop==0)
        {
                cmouse(&stop);
        }
        if (stop!=1)
        {
                asm mov ah,0x00
                asm int 0x16
                asm mov key,al
                asm mov scan,ah
        }
        if (scan==45 && key==0 /* Alt X */) {stop=1;}
        if ((key==13 && scan==50 /* Ctrl M */) && plate==0)
        {
                menuon( );
                plate=1;
        }
        if (key==19 /* Ctrl S */) {save( );a=0;b=0;}
        if (key==12 /* Ctrl L */) {load( );a=0;b=0;}
        if (key==4 && color !=3 /* Ctrl D */) {ndelete( );a=0;b=0;}
        if (key==6 /* Ctrl F */) {frame1=1;frame100( );frame1=0;}
        if (key==9  /* Ctrl I */ && check_delete != 0) {insert( );a=0;b=0;}
        if (key==26 /* Ctrl Z */) {send(0,1);}
        if (key==20 /* Ctrl T */) {text( );}
        if (scan==25 /* Ctrl P */) {screen( );}
        if (key==22 /* Ctrl V */) {play_4( );}
        if (key==8 /* Ctrl H */) {help1( );}
        if (scan==46 /* Alt C */) {clear( );}
        if (check_delete!=1)
        {
                if (key==18 /* Ctrl R */)
                {
                        color=4;
                        colour( );
                        a=0;b=0;
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (key==7 /* Ctrl G */)
                {
                        color=10;
                        colour( );
                        a=0;b=0;
                }
                if (key==15 /* Ctrl O */)
                {
                        color=12;
                        colour( );
                        a=0;b=0;
                }
                rcolor=color;
        }
        if (key==27 && plate==1)
        {
                menuoff( );
                plate=0;
                a=0;b=0;
        }
}while (stop!=1);
} // end if (open!=0)
else
{
        outtextxy(220,200,("Mouse Driver is not Install!"));
        sound(2200);
        delay(33);
        nosound( );
        getch( );
}
}

/* Load Font */
void load_font( )
{
FILE *l;
int k,j;
unsigned char temp;
int point,decode[8],x=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
l=fopen("font1.csl","rb");
for (k=0;k<256;k++)
{
        for (j=0;j<24;j++)
        {
                temp=getc(l);
                if (temp>=128) {decode[x]=1;temp=temp-128;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=64) {decode[x]=1;temp=temp-64;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=32) {decode[x]=1;temp=temp-32;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=16) {decode[x]=1;temp=temp-16;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=8) {decode[x]=1;temp=temp-8;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=4) {decode[x]=1;temp=temp-4;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=2) {decode[x]=1;temp=temp-2;x=x++;} else {decode[x]=0;x=x++;}
                if (temp>=1) {decode[x]=1;} else {decode[x]=0;}
                x=0;
                for (point=0;point<8;point++)
                {
                        output_font[k][point][j]=decode[point];
                } //end of loop j
        } //end of for loop
} //end if
fclose(l);
}

void play_4( )
{
int i,j,q,p1=0,p2=0,x1,x2,y1,y2;
unsigned char key=0;
void *backup;
unsigned int back;
emu *p;
frame=1;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=farmalloc(back);
getimage(37,88,597,280,backup);
cleardevice( );
table1( );
enable_cursor( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
simu=(emu *)malloc(sizeof(emu));
simu->up=NULL;
simu->link=NULL;
simu->end=0;
temp=simu;
i=0;
while (i<79)
{
        p=(emu *)malloc(sizeof(emu));
        p->up=simu;
        simu->link=p;
        simu=p;
        simu->link=NULL;
        simu->end=0;
        i=i++;
}
while (key!=27)
{
        key=getch( );
        if (key==12)        {load( );}
        if (key=='u')
        {
                disable_cursor( );
                up_to_down( );
                dis( );
                enable_cursor( );
        }
        if (key=='d')
        {
                disable_cursor( );
                down_to_up( );
                dis( );
                enable_cursor( );
        }
        if (key=='r')
        {
                disable_cursor( );
                left_right( );
                dis( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                enable_cursor( );
        }
        if (key==7) {call(&p1,&p2);send(p1,p2);}
        if (key==8) {help2( );}
        if (key=='l')
        {
                disable_cursor( );
                right_left( );
                dis( );
                enable_cursor( );
        }
        if (key==27)
        {
                if (h==1)
                {
                        simu=temp;
                        while (simu!=NULL)
                        {
                                simu=simu->link;
                                farfree((emu *)simu->up);
                        }
                        simu->up=NULL;
                        simu->link=NULL;
                }
                h=0;
        }
}
disable_cursor( );
clearviewport( );
table( );
putimage(37,88,backup,COPY_PUT);
enable_cursor( );
farfree(backup);
frame=0;
}

void table1( )
{
int i,p=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setbkcolor(3);
while (p<=222)
{
        for (i=0;i<81;i++)
        {
                if (fmod(i,20)==0)
                {
                        setcolor(9);
                        line(37+7*i,38+p,37+7*i,230+p);
                }
                else
                {
                        setcolor(14);
                        line(37+7*i,38+p,37+7*i,230+p);
                }
        }
        for (i=0;i<25;i++)
        {
                if (fmod(i,4)==0)
                {
                        setcolor(9);
                        line(37,38+8*i+p,597,38+8*i+p);
                }
                else
                {
                        setcolor(14);
                        line(37,38+8*i+p,597,38+8*i+p);
                }
        }
        p=222+p;
} // end while loop
setcolor(5);
outtextxy(17,468,"Ctrl H = Help.");
}

void dis( )
{
int p1,p2,i,j,q,xa1,xa2,ya1,ya2;
call(&p1,&p2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
simu=temp;
while (simu!=NULL && simu->end==0 && p1<p2)
{
        for (j=0;j<80;j++)
        {
                for (q=0;q<24;q++)
                {
                        x_pos1(j+1,&xa1,&xa2);
                        y_pos1(q+1,&ya1,&ya2);
                        setfillstyle(SOLID_FILL,simu->data1[j][q]);
                        bar(xa1,ya1,xa2,ya2);
                }
        }
        mark=simu;
        simu=simu->link;
        p1=p1++;
} // end while loop
simu=mark;
}

void right_left( )
{
int i,j,m;
h=1;
simu=temp;
init( );
for (i=79;i>=0;i--)
{
        for (j=0;j<24;j++)
        {
                simu->data1[0][j]=output[i][j];
        }
        if (i>0) {shift_right(i);}
}
}

void shift_right(int col)
{
int i,j,point,temp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
point=79-col;
temp=point;
simu=simu->link;
init( );
for (i=0;i<80-col;i++)
{
        for (j=0;j<24;j++)
        {
                simu->data1[point+1][j]=((emu *)(simu->up))->data1[point][j];
        }
        temp=temp-;
        point=temp;
}
}

void left_right( )
{
int i,j,m;
h=1;
simu=temp;
init( );
for (i=0;i<80;i++)
{
        for (j=0;j<24;j++)
        {
                simu->data1[79][j]=output[i][j];
        }
        if (i<79) {shift_left(i);}
}
}

void shift_left(int col)
{
emu *p;
int i,j,point,temp;
point=79-col;
temp=point;
simu=simu->link;
init( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=0;i<col+1;i++)
{
        for (j=0;j<24;j++)
        {
                simu->data1[point-1][j]=((emu *)(simu->up))->data1[point][j];
        }
        temp=temp++;
        point=temp;
}
}

void init( )
{
int i,j;
for (i=0;i<80;i++)
{
        for (j=0;j<24;j++)
        {
                simu->data1[i][j]=3;
                simu->end=0;
        }
}
}

void up_to_down( )
{
int i,j,m;
h=1;
simu=temp;
init( );
for (i=0;i<24;i++)
{
        for (j=0;j<80;j++)
        {
                simu->data1[j][23]=output[j][i];
        }
        if (i<23) {shift_data(i);}
}
((emu *)(simu->link))->end=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void shift_data(int row)
{
int i,j,point,temp;
simu=simu->link;
init( );
point=23-row;
temp=point;
for (i=0;i<row+1;i++)
{
        for (j=0;j<80;j++)
        {
                simu->data1[j][point-1]=((emu *)(simu->up))->data1[j][point];
        }
        temp=temp++;
        point=temp;
}
}

void down_to_up( )
{
int i,j,m;
h=1;
simu=temp;
init( );
for (i=0;i<24;i++)
{
        m=23-i;
        for (j=0;j<80;j++)
        {
                simu->data1[j][0]=output[j][m];
        }
        if (i<23) {shift_data1(i);}
}
((emu *)(simu->link))->end=1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void shift_data1(int row)
{
int i,j,point,temp;
simu=simu->link;
init( );
point=23-row;
point=row;
temp=point;
for (i=0;i<row+1;i++)
{
        for (j=0;j<80;j++)
        {
                simu->data1[j][point+1]=((emu *)(simu->up))->data1[j][point];
        }
        temp=temp-;
        point=temp;
}
}

void save_frame( )
{
int point,temp1,i,j,k;
FILE *fp;
unsigned char file_name[8],temp2[4];
void *backup,*p;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
free(backup);
enable_cursor( );
fp=fopen(file_name,"wb");
simu=temp;
while (simu!=NULL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
        for (i=0;i<80;i++)
        {
                for (j=0;j<24;j++)
                {
                        point=fmod(j,4);
                        if (simu->data1[i][j]==3) {temp2[point]=0;}
                        if (simu->data1[i][j]==4) {temp2[point]=1;}
                        if (simu->data1[i][j]==10) {temp2[point]=2;}
                        if (simu->data1[i][j]==12) {temp2[point]=3;}
                        if (point==3)
                        {
                                temp1=temp2[3]*64+temp2[2]*16+temp2[1]*4+temp2[0];
                                putc(temp1,fp);
                        } //end if
                } //end loop j
        } //end loop i
        p=simu;
        simu=simu->link;
} //end while
simu=p;
fclose(fp);
}

void x_pos1(int x3,int *x1,int *x2)
{
int i=0;
while (x3>10)
{
        x3=x3-10;
        i=i+1;
}
i=i*70;
if (x3==1) {*x1=38+i;*x2=42+i;}
if (x3==2) {*x1=45+i;*x2=49+i;}
if (x3==3) {*x1=52+i;*x2=56+i;}
if (x3==4) {*x1=59+i;*x2=63+i;}
if (x3==5) {*x1=66+i;*x2=70+i;}
if (x3==6) {*x1=73+i;*x2=77+i;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (x3==7) {*x1=80+i;*x2=84+i;}
if (x3==8) {*x1=87+i;*x2=91+i;}
if (x3==9) {*x1=94+i;*x2=98+i;}
if (x3==10) {*x1=101+i;*x2=105+i;}
}

/* Display Y Position */
void y_pos1(int y3,int *y1,int *y2)
{
int i=0;
while (y3>6)
{
        y3=y3-6;
        i=i+1;
}
i=i*48;
if (y3==1) {*y1=39+i;*y2=44+i;}
if (y3==2) {*y1=47+i;*y2=52+i;}
if (y3==3) {*y1=55+i;*y2=60+i;}
if (y3==4) {*y1=63+i;*y2=68+i;}
if (y3==5) {*y1=71+i;*y2=76+i;}
if (y3==6) {*y1=79+i;*y2=85+i;}
}

void screen( )
{
unsigned char key=0;
void *backup;
unsigned int back;
int end;
        list=0;
        e=0;
        rcolor=color;
        disable_cursor( );
        back=imagesize(37,88,597,280);
        backup=malloc(back);
        getimage(37,88,597,280,backup);
        setfillstyle(SOLID_FILL,0);
        bar(0,0,640,480);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
enable_cursor( );
load_font( );
paper( );
font=(data *)malloc(sizeof(data));
font->key=' ';
font->up=0;
font->down=0;
font->convert=0;
font->up1=NULL;
font->link=NULL;
temp=font;
while (key!=27)
{
        key=getch( );
        if (key=='k')
        {
                setfillstyle(SOLID_FILL,3);
                bar(500,40,600,70);
                setcolor(10);
                rectangle(501,41,599,69);
                setcolor(9);
                outtextxy(510,52,"Key =");
                setcolor(5);
                outtextxy(560,52,"Eng.");
                keyboard(list,e);
                setfillstyle(SOLID_FILL,3);
                bar(500,40,600,70);
        }
        if (key=='s')
        {
                enable_cursor( );
                save_text( );
        }
        if (key=='l')
        {
                point=0;
                point1=0;
                enable_cursor( );
                show( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (chk_name!=1)
                {
                        setfillstyle(SOLID_FILL,3);
                        bar(500,40,600,70);
                        setcolor(10);
                        rectangle(501,41,599,69);
                        setcolor(9);
                        outtextxy(510,52,"Key =");
                        setcolor(5);
                        outtextxy(560,52,"Eng.");
                        setcolor(4);
                        keyboard(point,point1);
                        setfillstyle(SOLID_FILL,3);
                        bar(500,40,600,70);
                }
        }
        if (key=='c')
        {
                paper( );
                font=temp;
                while (font!=NULL)
                {
                        free((data *)font->up);
                        font=font->link;
                }
                font->up1=NULL;
                font->link=NULL;
        }
        if (key=='g')
        {
                save_page( );
        }
} // end while
font=(data *)temp;
while (font!=NULL)
{
        free((data *)font->up);
        font=font->link;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        free(font);
        font->up1=NULL;
        font->link=NULL;
        list=0;
        e=0;
        disable_cursor( );
        cleardevice( );
        table( );
        putimage(37,88,backup,COPY_PUT);
        enable_cursor( );
        free(backup);
        color=rcolor;
        frame1=0;
}

void keyboard(int point,int point1)
{
unsigned char key=0,temp1,scan,thai_font,temp2,mark2,mark3;
int i,j=0,sthai=0,upp=0;
data *p;
color=4;
while (key!=27 && key!=13)
{
        if (text_board!=1)
        {
                setcolor(5);
                outtextxy(40+point,98+point1,"_");
                setcolor(4);
        }

        if (bioskey(1)!=0)
        {
                asm mov ah,00h;
                asm int 16h;
                asm mov key,al;
                asm mov scan,ah;

                if (text_board!=1)
                {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
i=color;
if (scan==77)
{
        if (font->link!=NULL)
        {
                setcolor(15);
                outtextxy(40+point,98+point1,"_");
                color=font->color;
                if (font->convert==0)
                {
                        disable_cursor( );
                        clear_pos1(point,point1);
                        clear_pos(font->key,point,point1);
                        enable_cursor( );
                }
                else
                {
                        disable_cursor( );
                        clear_pos1(point,point1);
                        clear_pos(thai[font->key-32],point,point1);
                        if (font->up!=0 || font->down!=0)
                        {
                                check(&thai_font,&upp);
                                if (upp==1)
                                {
                                        clear_pos(thai_font,point,point1);
                                        upp=0;
                                }
                                else
                                {
                                        thai_font=thai[font->down-32];
                                        clear_pos(thai_font,point,point1);
                                        if (font->up!=0)
                                        {
                                                clear_pos(thai[font->up-32],
                                                point,point1);
                                        }
                                }
                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                    font=font->link;
                    enable_cursor( );
                    if (point==544) {if (point1<168) {point=0;point1=point1+28;}}
                    else {point=point+8;}
                    setcolor(5);
                }
} // end else

if (scan==75)
{
        disable_cursor();
        if (font->up1!=NULL)
        {
                setcolor(15);
                outtextxy(40+point,98+point1,"_");
                color=font->color;
                if (font->convert==0)
                {
                        clear_pos1(point,point1);
                        clear_pos(font->key,point,point1);
                }
                else
                {
                        clear_pos1(point,point1);
                        clear_pos(thai[font->key-32],point,point1);
                        if (font->up!=0 || font->down!=0)
                        {
                                check(&thai_font,&upp);
                                if (upp==1)
                                {
                                        clear_pos(thai_font,point,point1);
                                        upp=0;
                                }
                                else
                                {
                                        thai_font=thai[font->down-32];
                                        clear_pos(thai_font,point,point1);
                                        if (font->up!=0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                    {
                                                        clear_pos(thai[font->up-32],
                                                            point,point1);
                                                    }
                                                }
                                            }
                                        }
                                        font=(data *)font->up1;
                                        setcolor(5);
                                        if (point==0) {if (point1>=28) {point1=point1-28;point=544;} else
                                        {point=0;point1=0;}} else {point=point-8;}
                                    }
                                    enable_cursor( );
                                }
                                color=i;
} // text_board !=1

if (key==18 /* Ctrl R */)
{
        color=4;
}
if (key==7 /* Ctrl G */)
{
        color=10;
}
if (key==15 /* Ctrl O */)
{
        color=12;
}

if ((key==18 || key==7 || key==15) && text_board==1) {colour();}

if (key=='`')
{
        if (sthai==0)
        {
                sthai=1 ;
                setfillstyle(SOLID_FILL,3);
                bar(500,40,600,70);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                setcolor(10);
                rectangle(501,41,599,69);
                setcolor(9);
                outtextxy(510,52,"Key =");
                setcolor(5);
                outtextxy(560,52,"Thai.");
        }
        else
        {
                sthai=0 ;
                setfillstyle(SOLID_FILL,3);
                bar(500,40,600,70);
                setcolor(10);
                rectangle(501,41,599,69);
                setcolor(9);
                outtextxy(510,52,"Key =");
                setcolor(5);
                outtextxy(560,52,"Eng.");
        }
}

if (key==8)
{
        i=color;
        disable_cursor( );
        if (font->up1!=NULL )
        {
                font=font->up1;
                if (text_board!=1)
                {
                        setcolor(15);
                        outtextxy(40+point,98+point1,"_");
                        setcolor(5);
                        if (point==0) {if (point1>=28) {point1=point1-28;point=544;} else
                        {point=0;point1=0;}} else {point=point-8;}
                }
                else
                {
                        if (point==0) {point=0;} else {point=point-8;point1=0;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                if (font->convert==0 || (font->up==0 && font->down==0))
                {
                        list=point;
                        e=point1;
                        font=font->link;
                        ((data *)((data *)(font->up1))->up1)->link=font;
                        mark=((data *)((data *)(font->up1))->up1);
                        free((data *)font->up1);
                        font->up1=mark;
                        mark=font;
                        show1( );
                        font=mark;
                        j=1;
                        if (text_board==1) {clear_pos1(point,point1);}
                }

                if ((font->up!=0 || font->down!=0) && j!=1)
                {
                        clear_pos1(point,point1);
                        if (font->up!=0)
                        {
                                font->up=0;
                                color=font->color;
                                clear_pos(thai[font->key-32],point,point1);
                                clear_pos(thai[font->down-32],point,point1);
                        }
                        else
                        {
                                font->down=0;
                                color=font->color;
                                clear_pos(thai[font->key-32],point,point1);
                        }
                        temp2=font->down;
                        font=font->link;
                        point=point+8;
                }
                j=0;
        }//end if list!=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                else
                {
                        if (((data *)(font->up1))->up1==NULL)
                        {
                                if (text_board!=1)
                                {
                                        setcolor(15);
                                        outtextxy(40+point,98+point1,"_");
                                        setcolor(5);
                                        if (point==0) {if (point1>=28) {point1=point1-28;point=544;}
                                        else
                                        {point=0;point1=0;}} else {point=point-8;}
                                }
                                if (text_board==1) {point=0;clear_pos1(point,point1);}
                                list=point;
                                e=point1;
                                free(font->up1);
                                temp=font;
                                font->up1=NULL;
                                show1( );
                        }
                }
                enable_cursor( );
                color=i;
        }//end key==8

        if (key==13)
        {
                font=temp;
                while (font->link!=NULL)
                {
                        font=font->link;
                }
                font->link=NULL;
        }

        if (key==27)
        {
                font=temp;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while (font!=NULL)
{
        free((data *)font->up);
        font=font->link;
}
free(font);
font->up1=NULL;
font->link=NULL;
point=0;point1=0;
if (text_board!=1)
{
        paper( );
}
}

if (text_board==1)
{
 if (thai[key-32]<215)
 {
        if (point==80) {point1=168;}
 }
 if (key==8) {point1=0;}
}

if ((point1<168) && frame1<168)
{
 disable_cursor( );
 if (key>=32)
 {
        if (sthai==0 && key != '^' )
        {
                clear_pos1(point,point1);
                clear_pos(key,point,point1);
                if (font->link==NULL)
                {
                        font->convert=0;
                        font->key=key;
                        font->color=color;
                        font->up=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                font->down=0;
                p=(data *)malloc(sizeof(data));
                p->up1=font;
                font->link=p;
                font=p;
                font->link=NULL;
                font->key=' ';
                font->up=0;
                font->down=0;
                font->color=color;
                font->convert=0;
        }
        else
        {
                if (point==544) {if (point1<168) {point=0;point1=point1+28;}}
                else {point=point+8;}
                list=point;
                e=point1;
                p=(data *)malloc(sizeof(data));
                if (font->up1!=NULL)
                {
                        (data *)p->link=font;
                        (data *)((data *)(font->up1))->link=p;
                        p->up1=font->up1;
                        (data *)font->up1=p;
                        font=font->up1;
                        font->key=key;
                        font->convert=0;
                        font->up=0;
                        font->down=0;
                        font->color=color;
                        font=font->link;
                        mark=font;
                        i=color;
                        show1( );
                        color=i;
                        font=mark;
                }
                else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        temp=p;
                        p->link=font;
                        font->up1=p;
                        p->up1=NULL;
                        p->key=key;
                        p->convert=0;
                        p->up=0;
                        p->down=0;
                        p->color=color;
                        i=color;
                        show1( );
                        color=i;
                        font=p->link;
                }
                if (point==0) {point=544;point1=point1-28;} else {point=point-8;}
        }

        if (text_board!=1)
        {
                if (point==544) {if (point1<168) {point=0;point1=point1+28;}}
                else {point=point+8;}
        }
        else
        {
                if (point==80) {point=80;} else {point=point+8;}
        }
}
if (sthai==1 && key != '' )
{
        if (thai[key-32]<215)
        {
                clear_pos1(point,point1);
                clear_pos(thai[key-32],point,point1);
                if (font->link==NULL)
                {
                        font->convert=1;
                        font->key=key;
                        font->color=color;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
font->up=0;
font->down=0;
p=(data *)malloc(sizeof(data));
p->up1=font;
font->link=p;
font=p;
font->link=NULL;
font->key=' ';
font->up=0;
font->down=0;
font->color=color;
font->convert=0;
}
else
{
if (point==544) {if (point1<168) {point=0;point1=point1+28;}}
else {point=point+8;}
list=point;
e=point1;
p=(data *)malloc(sizeof(data));
if (font->up1!=NULL)
{
(data *)p->link=font;
(data *)((data *)(font->up1))->link=p;
p->up1=font->up1;
(data *)font->up1=p;
font=font->up1;
font->key=key;
font->convert=1;
font->up=0;
font->down=0;
font->color=color;
font=font->link;
mark=font;
i=color;
show1( );
color=i;
font=mark;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    }
                    else
                    {
                            temp=p;
                            p->link=font;
                            font->up1=p;
                            p->up1=NULL;
                            p->key=key;
                            p->convert=1;
                            p->up=0;
                            p->down=0;
                            p->color=color;
                            i=color;
                            show10;
                            color=i;
                            font=p->link;
                    }
                    if (point==0) {point=544;point1=point1-28;} else
                      {point=point-8;}
                }
                if (text_board!=1)
                {
                    if (point==544) {if (point1<168)    {point=0;point1=
                      point1+28;}}
                    else {point=point+8;}
                }
                else
                {
                    if (point==80) {point=80;} else  {point=point+8;}
                }
            }
            else // Vowel
            {
                temp1=key;
                mark2=font->up;
                mark3=font->down;
                if (thai[key-32]!=temp2 && ((data *)(font->up1))->convert==1
                  &&
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
(thai[((data *)(font->up1))->key-32]<204 || thai[((data *)(font->
   up1))->key-32]>214) &&
(thai[((data *)(font->up1))->key-32]>138))
{
        ((data *)(font->up1))->color=color;
        if (text_board!=1)
        {
            if (point==0) {if (point1>=28)
              {point1=point1-28;point=544;}
            else {point=0;point1=0;} } else
              {point=point-8;}
        }
        else
        {
            if (point==0) {point=0;} else
              {point=point-8;}
        }
        thai_font=thai[key-32];
        if (thai[key-32]-thai[temp2-32]>0)
        {
            font->down=temp2;
            font->up=key;
        }
        else
        {
            font->down=key;
            font->up=temp2;
            temp1=temp2;
        }
        check(&thai_font,&upp);
        if (238<=thai_font && thai_font<=242) //{font[list-
          1].down=98;}
        {((data *)(font->up1))->down=98;}
        if (243<=thai_font && thai_font<=246) //{font[list-
          1].down=117;}
        {((data *)(font->up1))->down=117;}
        if (247<=thai_font && thai_font<=250) //{font[list-
          1].down=55;}
        {((data *)(font->up1))->down=55;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (251<=thai_font && thai_font<=254) //{font[list-
   1].down=110;}
{((data *)(font->up1))->down=110;}
if (234<=thai_font && thai_font<=237) //{font[list-
   1].down=121;}
{((data *)(font->up1))->down=121;}
if (230<=thai_font && thai_font<=233) //{font[list-
   1].down=89;}
{((data *)(font->up1))->down=89;}

if (upp==1)
{
        upp=0;
        clear_pos1(point,point1);
        clear_pos(thai[((data *)(font->up1))->key-
          32],point,point1);
        clear_pos(thai_font,point,point1);
        ((data *)(font->up1))->up=temp1;
}
else
{
        if ((thai[((data *)(font->up1))->down-32]
          ==215 ||
        thai[((data *)(font->up1))->down]==216)
        && thai[key-32]==224
        && thai[key-32]==225)
        {
            if (thai[((data *)(font->up1))->
               down-32]==215) {((data *)
               (font->up1))->up=54;}
            if (thai[((data *)(font->up1))->
               down-32]==216) {((data *)
               (font->up1))->up=94;}
        }
if (((thai[((data *)(font->up1))->down-32]==215||thai[((data *)(font->up1))->down-32]==216) &&
(thai[key-32]!=224 && thai[key-32]!=225)) || ((thai[((data *)(font->up1))->down]!=225 &&
thai[((data *)(font->up1))->down]!=224) && (thai[key-32]==215 || thai[key-32]==216)))
        {
            ((data *)(font->up1))->down=key;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            thai_font=thai[((data *)(font->up1))->
              down-32];
            key=0;
          }

          clear_pos1(point,point1);
          clear_pos(thai_font,point,point1);
          clear_pos(thai[((data *)(font->up1))->key-
            32],point,point1);

          if (((data *)(font->up1))->down!=0)
          {
            clear_pos1(point,point1);

              clear_pos(thai_font,point,point1);
            clear_pos(thai[((data *)(font->up1))->
              key-32],point,point1);
            clear_pos(thai[((data *)(font->up1))->
              down-32],point,point1);
            ((data *)(font->up1))->up=key;
          }
          else
          {
            ((data *)(font->up1))->down=key;
          }
        }
        font->up=mark2;
        font->down=mark3;

        if (text_board!=1)
        {
              if (point==544) {if (point1<168)
                {point=0;point1=point1+28;}}
              else {point=point+8;}
        }
        else
        {
              if (point==80) {point=80;} else
                {point=point+8;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                                    }
                                                } // end if
                                            } // end else
                                            temp2=key;
                                        }
                                    } // end key>=32
                                    enable_cursor( );
                                } // end point<168
                            } // end if (kbhit( ))
} // end while loop
if (text_board!=1)
{
            setcolor(15);
            outtextxy(40+point,98+point1,"_");
            setcolor(4);
            list=point;
            e=point1;
}
}

void paper( )
{
disable_cursor( );
setfillstyle(SOLID_FILL,15);
bar(37,88,602,280);
setcolor(5);
rectangle(38,89,601,279);
bar(480,251,598,279);
outtextxy(488,272,"Esc to abort.");
setcolor(10);
outtextxy(230,330,"Key K = Use keyboard.");
outtextxy(230,350,"Key S = Save.");
outtextxy(230,370,"Key L = Load.");
outtextxy(230,390,"Key C = Clear.");
outtextxy(230,410,"Key Enter = Accept.");
outtextxy(230,430,"Key Esc = Abort.");
outtextxy(230,450,"Key G = Send to port.");
enable_cursor( );
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void save_text( )
{
FILE *fp;
unsigned char file_name[8];
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
free(backup);
enable_cursor( );
font=(data *)temp;
fp=fopen(file_name,"wb");
while (font->link!=NULL)
{
        putc(font->convert,fp);
        putc(font->color,fp);
        putc(font->key,fp);
        if (font->convert==1)
        {
                putc(font->up,fp);
                putc(font->down,fp);
        }
        font=font->link;
}
fclose(fp);
}

void show( )
{
FILE *fp;
unsigned char thai_font;
int upp;
unsigned char file_name[8];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void *backup;
unsigned int back;
int i=0;
data *p;
e=0;
list=0;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
free(backup);
enable_cursor( );
font=temp;
fp=fopen(file_name,"rb");
if (chk_name!=1)
{
        if (fp==NULL)
        {
                not_found( );
        }
        else
        {
                while (!feof(fp) && i<=414)
                {
                        font->convert=getc(fp);
                        font->color=getc(fp);
                        font->key=getc(fp);
                        if (font->convert==1)
                        {
                                font->up=getc(fp);
                                font->down=getc(fp);
                        }
                        else
                        {
                                font->up=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        font->down=0;
                }
                p=(data *)malloc(sizeof(data));
                p->up1=font;
                font->link=p;
                font=p;
                font->link=NULL;
                i=i++;
        }
        if (i<=415)
        {
                font=font->up1;
                font->key=' ';
                font->convert=0;
                font->color=color;
                font->up=0;
                font->down=0;
                free(font->link);
                font->link=NULL;
                font=temp;
                fclose(fp);
                disable_cursor( );
                paper( );
                show1( );
                enable_cursor( );
                printf(" %d",i);
        }
        else
        {
                not_match( );
        }
    } // end else fp!=NULL
} // end if chk_name!=1
if (point==0) {point=544;point1=point1-28;} else {point=point-8;}
}

void show1( )
{
unsigned char thai_font;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int upp=0;
void *mark1;
point=list;
point1=e;
                while (font!=NULL)
                {
                        color=font->color;
                        if (font->convert==0)
                        {
                                clear_pos1(point,point1);
                                clear_pos(font->key,point,point1);
                        }
                        if (font->convert==1)
                        {
                                clear_pos1(point,point1);
                                clear_pos(thai[font->key-32],point,point1);

                                if (font->up!=0 || font->down!=0)
                                {
                                        thai_font=0;
                                        check(&thai_font,&upp);

                                        if (upp==1)
                                        {
                                                upp=0;
                                                clear_pos(thai_font,point,point1);
                                        }
                                        else
                                        {
                                                clear_pos(thai[font->down-32],point,point1);
                                                if (font->up!=0)
                                                {
                                                        clear_pos(thai[font->up-32],point,point1);
                                                }
                                        }
                                }
                        }
                        mark1=font;
                        font=font->link;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    if (point==544) {point=0;point1=point1+28;} else {point=point+8;}
                }
                font=mark1;
                frame1=point1;
}

void check(unsigned char *thai_font,int *upp)
{
                if (thai[font->down-32]==217 && thai[font->up-32]==224)
                {*thai_font=238;*upp=1;}
                if (thai[font->down-32]==217 && thai[font->up-32]==225)
                {*thai_font=239;*upp=1;}
                if (thai[font->down-32]==217 && thai[font->up-32]==226)
                {*thai_font=240;*upp=1;}
                if (thai[font->down-32]==217 && thai[font->up-32]==227)
                {*thai_font=241;*upp=1;}
                if (thai[font->down-32]==217 && thai[font->up-32]==228)
                {*thai_font=242;*upp=1;}

                if (thai[font->down-32]==218 && thai[font->up-32]==224)
                {*thai_font=243;*upp=1;}
                if (thai[font->down-32]==218 && thai[font->up-32]==225)
                {*thai_font=244;*upp=1;}
                if (thai[font->down-32]==218 && thai[font->up-32]==226)
                {*thai_font=245;*upp=1;}
                if (thai[font->down-32]==218 && thai[font->up-32]==227)
                {*thai_font=246;*upp=1;}

                if (thai[font->down-32]==219 && thai[font->up-32]==224)
                {*thai_font=247;*upp=1;}
                if (thai[font->down-32]==219 && thai[font->up-32]==225)
                {*thai_font=248;*upp=1;}
                if (thai[font->down-32]==219 && thai[font->up-32]==226)
                {*thai_font=249;*upp=1;}
                if (thai[font->down-32]==219 && thai[font->up-32]==227)
                {*thai_font=250;*upp=1;}

                if (thai[font->down-32]==220 && thai[font->up-32]==224)
                {*thai_font=251;*upp=1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (thai[font->down-32]==220 && thai[font->up-32]==225)
                {*thai_font=252;*upp=1;}
                if (thai[font->down-32]==220 && thai[font->up-32]==226)
                {*thai_font=253;*upp=1;}
                if (thai[font->down-32]==220 && thai[font->up-32]==227)
                {*thai_font=254;*upp=1;}

                if (thai[font->down-32]==221 && thai[font->up-32]==224)
                {*thai_font=234;*upp=1;}
                if (thai[font->down-32]==221 && thai[font->up-32]==225)
                {*thai_font=235;*upp=1;}
                if (thai[font->down-32]==221 && thai[font->up-32]==226)
                {*thai_font=236;*upp=1;}
                if (thai[font->down-32]==221 && thai[font->up-32]==227)
                {*thai_font=237;*upp=1;}

                if (thai[font->down-32]==222 && thai[font->up-32]==224)
                {*thai_font=230;*upp=1;}
                if (thai[font->down-32]==222 && thai[font->up-32]==225)
                {*thai_font=231;*upp=1;}
                if (thai[font->down-32]==222 && thai[font->up-32]==226)
                {*thai_font=232;*upp=1;}
                if (thai[font->down-32]==222 && thai[font->up-32]==227)
                {*thai_font=233;*upp=1;}
}

void clear_pos(unsigned char ch,int point,int point1)
{
int i,j;
if (text_board!=1)
{
        for (i=0;i<24;i++)
        {
                for (j=0;j<8;j++)
                {
                        if (output_font[ch][j][i]==1)
                        {
                                putpixel(40+j+point,90+i+point1,color);
                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
        }
}
else
{
        for (i=0;i<8;i++)
        {
                for (j=0;j<24;j++)
                {
                        if (output_font[ch][i][j]==1)
                        {
                                x_pos(i+1+point);
                                y_pos(j+1);
                                output[i+point][j]=color;
                                setfillstyle(SOLID_FILL,color);
                                bar(xa1,ya1,xa2,ya2);
                        }
                }
        }
}
}

void clear_pos1(int point,int point1)
{
int i,j;
if (text_board!=1)
{
        for (i=0;i<24;i++)
        {
                for (j=0;j<8;j++)
                {
                        if (output_font[32][j][i]==0)
                        {
                                putpixel(40+j+point,90+i+point1,15);
                        }

                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else
{
        for (i=0;i<8;i++)
        {
                for (j=0;j<24;j++)
                {
                        if (output_font[32][i][j]==0)
                        {
                                x_pos(i+1+point);
                                y_pos(j+1);
                                output[i+point][j]=3;
                                setfillstyle(SOLID_FILL,3);
                                bar(xa1,ya1,xa2,ya2);
                        }
                }
        }
}
}

void text( )
{
text_board=1;
list=0;
point=0;
point1=0;
setfillstyle(SOLID_FILL,3);
bar(500,40,600,70);
setcolor(10);
rectangle(501,41,599,69);
setcolor(9);
outtextxy(510,52,"Key =");
setcolor(5);
outtextxy(560,52,"Eng.");
keyboard(0,0);
setfillstyle(SOLID_FILL,3);
bar(500,40,600,70);
text_board=0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void not_found( )
{
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
bar(170,130,464,190);
setcolor(5);
line(172,132,462,132);
line(172,132,172,188);
line(172,188,462,188);
line(462,132,462,188);
setcolor(17);
outtextxy(250,154,"File not found !");
setcolor(3);
delay(1500);
putimage(37,88,backup,COPY_PUT);
free(backup);
enable_cursor( );
}

void not_match( )
{
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
bar(170,130,464,190);
setcolor(5);
line(172,132,462,132);
line(172,132,172,188);
line(172,188,462,188);
line(462,132,462,188);
setcolor(17);
outtextxy(250,154,"File not correct !");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(3);
delay(1500);
putimage(37,88,backup,COPY_PUT);
free(backup);
enable_cursor( );
}

void help1( )
{
char key=0;
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(157,100,490,391);
backup=malloc(back);
getimage(157,100,490,391,backup);
setfillstyle(SOLID_FILL,15);
bar(157,100,490,391);
setcolor(8);
line(157,135,490,135);
rectangle(158,101,489,390);
settextstyle(0,0,2);
outtextxy(165,110,"   .Menu Command.");
settextstyle(0,0,0);
setcolor(9);
outtextxy(175,140,"Ctrl R = Red.");
outtextxy(175,158,"Ctrl G = Green.");
outtextxy(175,176,"Ctrl O = Orange.");
outtextxy(175,194,"Ctrl M = Plate.");
outtextxy(175,194+18,"Ctrl S = Save.");
outtextxy(175,194+36,"Ctrl L = Load.");
outtextxy(175,194+54,"Ctrl T = Text.");
outtextxy(175,194+72,"Ctrl D = Delete.");
outtextxy(175,194+90,"Ctrl I = Insert.");
outtextxy(175,194+108,"Ctrl P = Text Paper.");
outtextxy(175,194+126,"Ctrl Z = Send to port.");
outtextxy(175,194+144,"Ctrl F = Motion Pictures.");
outtextxy(175,194+162,"Ctrl V = Play 4 Directions.");
outtextxy(175,194+180,"Alt  X = Quit.");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
enable_cursor( );
while (key!=27)
{
        key=getch( );
}
disable_cursor( );
putimage(157,100,backup,COPY_PUT);
enable_cursor( );
free(backup);
}

void save_page( )
{
unsigned char ch,pege[4];
int i,j,color,pin=0;
unsigned char temp2[24];
void *backup;
unsigned int back;
disable_cursor( );
back=imagesize(170,330,464,390);
backup=malloc(back);
getimage(170,330,464,390,backup);
setfillstyle(SOLID_FILL,15);
bar(170,130+200,464,190+200);
setcolor(5);
line(172,132+200,462,132+200);
line(172,132+200,172,188+200);
line(172,188+200,462,188+200);
line(462,132+200,462,188+200);
setcolor(25);
outtextxy(230,155+200,"Data is now SENDING ...");
enable_cursor( );
point=0;
point1=0;
font=(data *)temp;
//while (temp1<list)
while(font->link!=NULL)
{
        for (i=0;i<8;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                for (j=0;j<24;j++)
                {
                        color=getpixel(40+i+point,90+j+point1);
                        if (color!=15)
                        {
                                temp2[j]=color;
                        }
                        else
                        {
                                temp2[j]=15;
                        }
                }
                for (j=0;j<24;j++)
                {
                        pin=fmod(j,4);
                        if (temp2[j]==15) {page[pin]=0;}
                        if (temp2[j]==4) {page[pin]=1;}
                        if (temp2[j]==10) {page[pin]=2;}
                        if (temp2[j]==12) {page[pin]=3;}
                        if (pin==3)
                        {
                                ch=page[3]*64+page[2]*16+page[1]*4+page[0];
                                _bios_serialcom(_COM_SEND, COM2, ch);
                        }
                }
        }
        if (point==544) {if (point1<168) {point=0;point1=point1+28;}}
        else {point=point+8;}
        font=font->link;
} // end while
disable_cursor( );
putimage(170,330,backup,COPY_PUT);
enable_cursor( );
}

void pos_frame(int *na,int *pos)
{
unsigned char *n1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char n;
int i=0,j=0,k,p,q;
*na=0;
while (n!=13 && n!=27)
{
        setcolor(5);
        outtextxy(270+8*i+*pos,154,"_");
        setcolor(4);

        n=getch( );

        if (n==8 && i>0)
        {
                i=i-;
                disable_cursor( );
                setfillstyle(SOLID_FILL,15);
                bar(270+8*i+*pos,152,290+8*i+*pos,169);
                setcolor(15);
                outtextxy(270+8*i+*pos,154,"_");
                enable_cursor( );
                setcolor(4);
                *na=q;
                if (i==0) {j=0;*na=0;}
        }
        if (i==0 && n==13) {n=0;}

        if (n==27) {e=1;}

        if (n!=13 && n!=27 && i<3 && n!=8)
        {
                if (n>=48 && n<=57)
                {
                        *na=(*na*j)+(n-48);
                        if (*na<10) {q=*na;}
                        if (*na<=100)
                        {
                                j=10;
                                setcolor(15);
                                disable_cursor( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        outtextxy(270+8*i+*pos,154,"_");
                                        setcolor(4);
                                        sprintf(n1,"%d",n-48);
                                        setcolor(8);
                                        outtextxy(270+8*i+*pos,154,n1);
                                        enable_cursor( );
                                        i=i++;
                                }
                                else
                                {
                                        *na=q;
                                }
                        }
                }
        if (i>3) {sound(220);delay(33);nosound( );}
}
setcolor(15);
disable_cursor( );
outtextxy(270+8*i+*pos,154,"_");
enable_cursor( );
setcolor(4);
if (n==13) {chk_name=0;}
}

void call(int *p1,int *p2)
{
void *backup;
unsigned int back;
int pos=0,pos1=0;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
setfillstyle(SOLID_FILL,15);
bar(170,130,464,190);
setcolor(5);
line(172,132,462,132);
line(172,132,172,188);
line(172,188,462,188);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(462,132,462,188);
setcolor(17);
outtextxy(200,154,"From :");
enable_cursor( );
setcolor(3);
pos_frame(&pos1,&pos);
setcolor(17);
*p1=pos1;
if (e!=1)
{
        outtextxy(290,154,"To ");
        pos=55;
        pos_frame(&pos1,&pos);
        *p2=pos1;
}
if (e==1) {*p1=0;*p2=0;}
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
free(backup);
enable_cursor( );
e=0;
}

void help2( )
{
void *backup;
unsigned int back;
unsigned char key=0;
disable_cursor( );
back=imagesize(200,150,450,320);
backup=malloc(back);
getimage(200,150,450,320,backup);
setfillstyle(SOLID_FILL,15);
bar(200,150,450,320);
setcolor(5);
rectangle(201,151,449,319);
line(201,177,449,177);
setcolor(8);
settextstyle(0,0,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextxy(215,155,".Menu Command.");
setcolor(9);
settextstyle(0,0,0);
outtextxy(230,190-5,"Ctrl L = Load.");
outtextxy(230,210-5,"Ctrl G = Send to port.");
outtextxy(230,230-5,"Key  U = Down to up.");
outtextxy(230,250-5,"Key  D = Up to down.");
outtextxy(230,270-5,"Key  R = Right to left.");
outtextxy(230,290-5,"Key  L = Left to right.");
outtextxy(230,310-5,"Key Esc = Quit.");
enable_cursor( );
while (key!=27)
{
        key=getch( );
}
disable_cursor( );
putimage(200,150,backup,COPY_PUT);
enable_cursor( );
free(backup);
}

void frame100( )
{
char key=0,*p1,scan=0;
int i,j,p3,p4;
emu *p;
setcolor(color);
disable_cursor( );
simu=(emu *)malloc(sizeof(emu));
simu->link=NULL;
simu->up=NULL;
temp=simu;
init( );
setcolor(8);
outtextxy(260,350,".Motion Picture.");
outtextxy(500,65,"Frame : ");
setcolor(6);
outtextxy(580,65,"0.");
enable_cursor( );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
e=0;
j=0;
i=0;
        while (key!=27)
        {
                while (bioskey(1)==0)
                {
                        cmouse(&j);
                }
                asm mov ah,00h;
                asm int 16h;
                asm mov key,al;
                asm mov scan,ah;
                if (scan==46) {clear( );}
                if (key==4) {ndelete( );a=0;b=0;}
                if (key==9) {insert( );a=0;b=0;}
                if (key==16) {plate1( );plate2=1;}
                if (key==19) {save_frame( );}
                if (key==12) {load_frame( );}
                if (key==8 && scan==35) {help3( );}
                if (key==26 && scan==44) {call(&p3,&p4);send(p3,p4);}
                if (scan==83)
                {
                        if (simu->link==NULL && simu->up!=NULL)
                        {
                                clear( );
                                simu=simu->up;
                                free(simu->link);
                                simu->link=NULL;
                                disable_cursor( );
                                for (i=0;i<80;i++)
                                {
                                        for (j=0;j<24;j++)
                                        {
                                                x_pos(i+1);
                                                y_pos(j+1);
                                                setfillstyle(SOLID_FILL,simu->data1[i][j]);
                                                bar(xa1,ya1,xa2,ya2);
                                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        }
                        enable_cursor( );
                }
                if (e>0) {e=e-1;}
                if (e>=0)
                {
                        itoa(e,p1,10);
                        if (e<100) {j=7;}
                        if (e<10) {j=0;}
                        setfillstyle(SOLID_FILL,3);
                        bar(560,60,587,72);
                        setcolor(6);
                        outtextxy(580-j,65,p1);
                }
}

if (check_delete!=1)
{
                if (key==18 /* Ctrl R */)
                {
                        color=4;
                        colour( );
                        a=0;b=0;
                }
                if (key==7 /* Ctrl G */)
                {
                        color=10;
                        colour( );
                        a=0;b=0;
                }
                if (key==15 /* Ctrl O */)
                {
                        color=12;
                        colour( );
                        a=0;b=0;
                }
                rcolor=color;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (scan==15 /* Tab */)
{
        disable_cursor( );
        if (e<100) {e=e++;}
        j=0;
        if (e<100)
        {
                itoa(e,p1,10);
                if (e>=10) {j=7;}
                if (e>=100) {j=15;}
                setfillstyle(SOLID_FILL,3);
                bar(560,60,587,72);
                setcolor(6);
                outtextxy(580-j,65,p1);
        }
        if (simu->link==NULL)
        {
                p=(emu *)malloc(sizeof(emu));
                (emu *)p->up=simu;
                (emu *)simu->link=p;
                simu=p;
                simu->link=NULL;
                init( );
                for (i=0;i<80;i++)
                {
                        for (j=0;j<24;j++)
                        {
                                simu->data1[i][j]=((emu *)(simu->up))->data1[i][j];
                        }
                }
        }
        else
        {
                simu=simu->link;
                for (i=0;i<80;i++)
                {
                        for (j=0;j<24;j++)
                        {
                                x_pos(i+1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        y_pos(j+1);
                                        setfillstyle(SOLID_FILL,simu->data1[i][j]);
                                        bar(xa1,ya1,xa2,ya2);
                                }
                        }
                }
                enable_cursor( );
}

if (scan==14 /* Backspace */)
{
                if (e>0) {e=e-;}
                j=0;
                setcolor(6);
                if (e>=0)
                {
                        itoa(e,p1,10);
                        if (e<200) {j=15;}
                        if (e<100) {j=7;}
                        if (e<10) {j=0;}
                        setfillstyle(SOLID_FILL,3);
                        bar(560,60,587,72);
                        outtextxy(580-j,65,p1);
                }
                if (simu->up!=NULL)
                {
                        frame1=0;
                        clear();
                        frame1=1;
                        simu=simu->up;
                        disable_cursor( );
                        for (i=0;i<80;i++)
                        {
                                for (j=0;j<24;j++)
                                {
                                        x_pos(i+1);
                                        y_pos(j+1);
                                        setfillstyle(SOLID_FILL,simu->data1[i][j]);
                                        bar(xa1,ya1,xa2,ya2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        }
                                    }
                                    enable_cursor( );
                                }
                            }
                        }

plate2=0;
simu=temp;
while (simu!=NULL)
{
        free(simu->up);
        simu=simu->link;
}
free(simu);
simu->up=NULL;
simu->link=NULL;
disable_cursor( );
cleardevice( );
table( );
enable_cursor( );
}

void load_frame( )
{
FILE *l;
unsigned char file_name[8],decode[8];
void *backup;
unsigned int back;
int x=0,y=0,point,i=0,j=0,temp2;
emu *p;
disable_cursor( );
back=imagesize(37,88,597,280);
backup=malloc(back);
getimage(37,88,597,280,backup);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,backup,COPY_PUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
free(backup);
enable_cursor( );
simu=temp;
init( );
l=fopen(file_name,"rb");
back=0;
if (chk_name!=1)
{
        if (l!=NULL)
        {
                while (!feof(l))
                {
                        if (j==6) {j=0;i=i++;y=0;}
                        if (i==80)
                        {
                                j=0;
                                i=0;
                                p=(emu*)malloc(sizeof(emu));
                                p->up=simu;
                                simu->link=p;
                                simu=p;
                                simu->link=NULL;
                                init( );
                        }
                        x=7;
                        temp2=getc(l);
                        if (temp2>=128) {decode[x]=1;temp2=temp2-128;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=64) {decode[x]=1;temp2=temp2-64;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=32) {decode[x]=1;temp2=temp2-32;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=16) {decode[x]=1;temp2=temp2-16;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=8) {decode[x]=1;temp2=temp2-8;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=4) {decode[x]=1;temp2=temp2-4;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=2) {decode[x]=1;temp2=temp2-2;x=x--;} else {decode[x]=0;x=x--;}
                        if (temp2>=1) {decode[x]=1;} else {decode[x]=0;}
                        for (point=0;point<8;point++)
                        {
                                temp2=decode[point]+decode[point+1];
                                if (temp2==0) {simu->data1[i][y]=3;}
                                if (temp2==1 && decode[point]==1) {simu->data1[i][y]=4;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        if (temp2==1 && decode[point]==0) {simu->data1[i][y]=10;}
                                        if (temp2==2) {simu->data1[i][y]=12;}
                                        point=point+1;
                                        y=y++;
                                }
                                back=back+1;
                                j=j++;
                        }
                }
                else
                {
                        not_found( );
                }
                fclose(l);
                simu=simu->up;.
                free(simu->link);
                simu->link=NULL;
                simu=temp;
                disable_cursor( );
                for (i=0;i<80;i++)
                {
                        for (j=0;j<24;j++)
                        {
                                x_pos(i+1);
                                y_pos(j+1);
                                setfillstyle(SOLID_FILL,simu->data1[i][j]);
                                bar(xa1,ya1,xa2,ya2);
                        }
                }
} // end chk_name!=1
enable_cursor( );
}

void cmouse(int *stop)
{
int x,y;
char x_str[5],y_str[5],button;

/* Read Mouse Position */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            asm mov ax,03h;
            asm int 33h;
            asm mov button,bl;
            asm mov x,cx;
            asm mov y,dx;
/* Display Position */
itoa(x,x_str,10);
itoa(y,y_str,10);
if (((x>36 && x<598) && (y>87 && y<281)))
{
            if (x-x1!=0 || y-y1!=0)
            {
                        setcolor(10);
                        pos_x(x);
                        pos_y(y);
                        setfillstyle(SOLID_FILL,3);
                        bar(541,290,575,320);
                        x1=x;
                        y1=y;
            }
            if (x-x1==0 && y-y1==0)
            {
                        setcolor(10);
                        pos_x(x);
                        pos_y(y);
            }

            /* Check Mouse Click */
            if (button==1 && (x2-a!=0 || y2-b!=0))
            {
                        a=x2;
                        b=y2;
                        if (frame1!=1)
                        {
                                    output[x2-1][y2-1]=color;
                        }
                        else
                        {
                                    simu->data1[x2-1][y2-1]=color;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }
                x_pos(x2);
                y_pos(y2);
                setfillstyle(SOLID_FILL,color);
                disable_cursor( );
                bar(xa1,ya1,xa2,ya2);
                enable_cursor( );
        }
}
/* Check Click in Plate Menu */
if ((button==1 && plate==1))
{
        if ((370<y && y<390))
        {
                if (220<x && x<300) {load();}
                if (340<x && x<420) {clear();}
                if (460<x && x<540) {insert();a=0;b=0;}
        }
        if ((110<x && x<120) && check_delete!=1)
        {
                if (370<y && y<380)
                {
                        color=10;
                        colour( );
                }
                if (395<y && y<405)
                {
                        color=4;
                        colour( );
                }
                if (420<y && y<430)
                {
                        color=12;
                        colour( );
                }
                rcolor=color;
        }
        if (410<y && y<430)
        {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (220<x && x<300) {save();}
                if (340<x && x<420) {*stop=1;}
                if (460<x && x<540) {if (color!=3) {ndelete();a=0;b=0;}}
        }
}/* End of loop */
/* Check Mouse Click in Plate of Motion Picture */
if (button==1 && plate2==1)
{
        if (415<y && y<435)
        {
                if (170<x && x<220)  {play( );}
                if (250<x && x<300) {stop1( );}
                if (330<x && x<380) {step_inc( );}
                if (410<x && x<460) {step_dec( );}
        }
}
}

void play( )
{
int i=0,j[8],q,click=0;
char *p1;
        e=-1;
        simu=temp;
        while (simu!=NULL)
        {
                disable_cursor( );
                for (i=0;i<80;i++)
                {
                        for (q=0;q<24;q++)
                        {
                                x_pos(i+1);
                                y_pos(q+1);
                                setfillstyle(SOLID_FILL,simu->data1[i][q]);
                                bar(xa1,ya1,xa2,ya2);
                        }
                }
                q=0;
                if (e<100) {e=e++;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if (e<100)
                {
                        itoa(e,p1,10);
                        if (e>=10) {q=7;}
                        setfillstyle(SOLID_FILL,3);
                        bar(560,60,587,72);
                        setcolor(6);
                        outtextxy(580-q,65,p1);
                }
                mark=simu;
                simu=simu->link;
                enable_cursor( );
                delay(200);
                cmouse(&click);
        } // end while loop
        simu=mark;
}

void stop1( )
{
int i,j=0;
char *p1;
while (simu->link!=NULL)
{
        simu=simu->link;
        if (e<100) {e=e++;}
        if (e<100)
        {
                itoa(e,p1,10);
                if (e<100 && e>=10) {j=7;}
                setfillstyle(SOLID_FILL,3);
                bar(560,60,587,72);
                setcolor(6);
                outtextxy(580-j,65,p1);
        }
}
for (i=0;i<80;i++)
{
        for (j=0;j<24;j++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
                        x_pos(i+1);
                        y_pos(j+1);
                        setfillstyle(SOLID_FILL,simu->data1[i][j]);
                        bar(xa1,ya1,xa2,ya2);
            }
}
}

void step_inc( )
{
int i,j=0;
char *p1;
if (simu->link!=NULL)
{
            simu=simu->link;
            if (e<100) {e=e+1;}
            if (e<100)
            {
                        itoa(e,p1,10);
                        if (e>=10) {j=7;}
                        setfillstyle(SOLID_FILL,3);
                        bar(560,60,587,72);
                        setcolor(6);
                        outtextxy(580-j,65,p1);
            }
}
for (i=0;i<80;i++)
{
            for (j=0;j<24;j++)
            {
                        x_pos(i+1);
                        y_pos(j+1);
                        setfillstyle(SOLID_FILL,simu->data1[i][j]);
                        bar(xa1,ya1,xa2,ya2);
            }
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void step_dec( )
{
int i,j=0;
char *p1;
if (simu->up!=NULL)
{
        simu=simu->up;
        if (e>0) {e=e-1;}
        if (e>=0)
        {
                itoa(e,p1,10);
                if (e<100) {j=7;}
                if (e<10) {j=0;}
                setfillstyle(SOLID_FILL,3);
                bar(560,60,587,72);
                setcolor(6);
                outtextxy(580-j,65,p1);
        }
}
for (i=0;i<80;i++)
{
        for (j=0;j<24;j++)
        {
                x_pos(i+1);
                y_pos(j+1);
                setfillstyle(SOLID_FILL,simu->data1[i][j]);
                bar(xa1,ya1,xa2,ya2);
        }
}
}

void plate1( )
{
int j[8],r=0;
disable_cursor( );
setfillstyle(SOLID_FILL,7);
bar(150,400,485,450);
setcolor(8);
rectangle(151,401,484,449);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
while (r<=240)
{
        setfillstyle(SOLID_FILL,8);
        bar(170+r,415,220+r,435);
        r=r+80;
}
        setcolor(9);
        setfillstyle(SOLID_FILL,9);
        j[0]=185;
        j[1]=419;
        j[2]=205;
        j[3]=424;
        j[4]=185;
        j[5]=429;
        j[6]=185;
        j[7]=419;
        fillpoly(4,j); // Play Button
        j[0]=185+78;
        j[1]=419;
        j[2]=205+78;
        j[3]=424;
        j[4]=185+78;
        j[5]=429;
        j[6]=185+78;
        j[7]=419;
        fillpoly(4,j); // Stop Button
        bar(286,419,289,429);
        j[0]=190+78*2;
        j[1]=419;
        j[2]=200+78*2;
        j[3]=424;
        j[4]=190+78*2;
        j[5]=429;
        j[6]=190+78*2;
        j[7]=419;
        fillpoly(4,j); // Increment Frame Button
        j[0]=200+78*2;
        j[1]=419;
        j[2]=210+78*2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        j[3]=424;
        j[4]=200+78*2;
        j[5]=429;
        j[6]=200+78*2;
        j[7]=419;
        fillpoly(4,j); // Increment Frame Button
        j[0]=200+78*3;
        j[1]=419;
        j[2]=190+78*3;
        j[3]=424;
        j[4]=200+78*3;
        j[5]=429;
        j[6]=200+78*3;
        j[7]=419;
        fillpoly(4,j); // Decrement Frame Button
        j[0]=210+78*3;
        j[1]=419;
        j[2]=200+78*3;
        j[3]=424;
        j[4]=210+78*3;
        j[5]=429;
        j[6]=210+78*3;
        j[7]=419;
        fillpoly(4,j); // Decrement Frame Button
        enable_cursor( );
}

void help3( )
{
void *backup;
unsigned int back;
unsigned char key=0;
disable_cursor( );
back=imagesize(187,100,460,371);
backup=malloc(back);
getimage(187,100,460,371,backup);
setfillstyle(SOLID_FILL,15);
bar(157+30,100,490-30,391-20);
setcolor(8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(157+31,135,490-31,135);
rectangle(158+30,101,489-30,390-20);
settextstyle(0,0,2);
outtextxy(165,110,"   .Menu Command.");
settextstyle(0,0,0);
setcolor(9);
outtextxy(175+40,140,"Ctrl R = Red.");
outtextxy(175+40,158,"Ctrl G = Green.");
outtextxy(175+40,176,"Ctrl O = Orange.");
outtextxy(175+40,194,"Ctrl P = Plate.");
outtextxy(175+40,194+18,"Ctrl S = Save.");
outtextxy(175+40,194+36,"Ctrl L = Load.");
outtextxy(175+40,194+54,"Ctrl I = Insert.");
outtextxy(175+40,194+72,"Ctrl D = Delete.");
outtextxy(175+40,194+90,"Ctrl Z = Send to Port.");
outtextxy(175+40,194+108,"Tab    = Increment.");
outtextxy(175+40,194+126,"Backspace = Decrement.");
outtextxy(175+40,194+144,"Alt  C = Clear.");
outtextxy(175+40,194+162,"Key Esc = Quit.");
enable_cursor( );
while (key!=27)
{
        key=getch( );
}
disable_cursor( );
putimage(187,100,backup,COPY_PUT);
enable_cursor( );
free(backup);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง-2. EDIT_FON.EXE

เป็นโปรแกรมเสริมที่คณะผู้จัดทำจัดทำไว้ให้ผู้ใช้ เพื่อใช้ในการแก้ไขรูปแบบของบิตแม็ปของตัวอักษร ซึ่งผู้ใช้จะต้องทำการออกแบบบิตแม็ปของตัวอักษรใหม่ตามค่ารหัสแอสกีของตัวอักษรแต่ละตัว

นอกจากวิธีนี้แล้ว เมื่อผู้ใช้ต้องการเปลี่ยนรูปแบบของตัวอักษร ซึ่งโดยปกติแฟ้มข้อมูลบิตแม็ปของตัวอักษรจะมีทำไว้แล้วเป็นมาตรฐาน เช่น บิตแม็ปของตัวอักษรของโปรแกรมเวิร์ดราชวิถี บิตแม็ปของตัวอักษรของโปรแกรมเวิร์ดจุฬา เป็นต้น โดยผู้ใช้สามารถนำมาใช้ได้เลยด้วยการเปลี่ยนชื่อแฟ้มข้อมูลบิตแม็ปของตัวอักษรที่นำมาดังกล่าว เป็นชื่อ 'FONT' เพราะโปรแกรมหลักที่เราได้เขียนขึ้นมานั้น จะเรียกบิตแม็ปของตัวอักษรจากแฟ้มข้อมูลชื่อ 'FONT' นี้มาใช้

สำหรับวิธีการใช้โปรแกรม EDIT_FON.EXE นี้ ได้อธิบายไว้แล้วในภาคผนวก จ. และSource Code ภาษาซีของโปรแกรมนี้จะแสดงไว้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include<string.h>
#include<alloc.h>
#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#include<graphics.h>
#include<conio.h>
#include<bios.h>
#include<dos.h>
struct charec{
int  up;
char output[8][24];
int  pointer;
} charecter[256];

void save( );
void load( );
void initial( );
void clear( );
void table_save(char name[8]);
void pos_x( );
void pos_y( );
void x_pos( );
void y_pos( );
void table( );
void mouse( );
void enable_cursor( );
void disable_cursor( );
void ndelete( );
void insert( );
void decrease( );
void increase( );
void far *backup;
int xa1,ya1,xa2,ya2,y2,x2,color,index2,chk_name,index=0;

main( )
{
int gd =DETECT,gm;
initgraph(&gd,&gm,"c:\\lang\\tc3\\bgi");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
index2=0;
color=5;
table( );
mouse( );
farfree(backup);
closegraph( );
return 0;
}

void x_pos(int x3)
{
int i=0;
if (x3==1) {xa1=88+i;xa2=98+i;}
if (x3==2) {xa1=101+i;xa2=111+i;}
if (x3==3) {xa1=114+i;xa2=124+i;}
if (x3==4) {xa1=127+i;xa2=137+i;}
if (x3==5) {xa1=140+i;xa2=150+i;}
if (x3==6) {xa1=153+i;xa2=163+i;}
if (x3==7) {xa1=166+i;xa2=176+i;}
if (x3==8) {xa1=179+i;xa2=189+i;}
}

/* Display Y Position */
void y_pos(int y3)
{
int i=0;
while (y3>6)
{
y3=y3-6;
i=i+1;
}
i=i*72;
if (y3==1) {ya1=89+i;ya2=98+i;}
if (y3==2) {ya1=101+i;ya2=110+i;}
if (y3==3) {ya1=113+i;ya2=122+i;}
if (y3==4) {ya1=125+i;ya2=134+i;}
if (y3==5) {ya1=137+i;ya2=146+i;}
if (y3==6) {ya1=149+i;ya2=158+i;}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void pos_x(int p_x)
{
int i=0;
char i1[4];
if (87<=p_x && p_x<100) {i=i+1;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (100<=p_x && p_x<113) {i=i+2;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (113<=p_x && p_x<126) {i=i+3;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (126<=p_x && p_x<139) {i=i+4;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (139<=p_x && p_x<152) {i=i+5;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (152<=p_x && p_x<165) {i=i+6;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (165<=p_x && p_x<178) {i=i+7;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
if (178<=p_x && p_x<191) {i=i+8;x2=i;itoa(i,i1,10);outtextxy(273,350,i1);}
}

/* Call Show Y_Position Function */
void pos_y(int p_y)
{
int i=0;
char i1[4];
while (p_y>160)
{
p_y=p_y-72;
i=i+1;
}
i=i*6;
if (88<=p_y && p_y<100) {i=i+1;y2=i;itoa(i,i1,10);outtextxy(273,365,i1);}
if (100<=p_y && p_y<112) {i=i+2;y2=i;itoa(i,i1,10);outtextxy(273,365,i1);}
if (112<=p_y && p_y<124) {i=i+3;y2=i;itoa(i,i1,10);outtextxy(273,365,i1);}
if (124<=p_y && p_y<136) {i=i+4;y2=i;itoa(i,i1,10);outtextxy(273,365,i1);}
if (136<=p_y && p_y<148) {i=i+5;y2=i;itoa(i,i1,10);outtextxy(273,365,i1);}
if (148<=p_y && p_y<=160) {i=i+6;y2=i;itoa(i,i1,10);outtextxy(273,365,i1);}
}

void disable_cursor( )
{
asm mov ax,02h;
asm int 33h;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void enable_cursor( )
{
asm mov ax,01h;
asm int 33h;
}

void mouse( )
{
int open,open1,button,x,x1,y,y1,stop,a,b;
char x_str[5],y_str[5],key;
        /* check mouse and mouse driver */
        asm mov ax,00h;
        asm int 33h;
        asm mov open,ax;
        asm mov ax,35h;
        asm mov open1,ax;
        /* Set mouse sensitivity and Interrupt rate */
        asm mov ax,1ah; // sentivity
        asm mov bx,50d;
        asm mov cx,50d;
        asm mov dx,50d;
        asm int 33h;
        asm mov ax,1ch; // interrupt rate
        asm mov bx,4d;
        asm int 33h;
        /* load mouse cursor and position */
if ((open!=0) && (open1!=0))
{
        enable_cursor( );
        asm mov ax,07h;
        asm mov cx,00h;
        asm mov dx,278h;
        asm int 33h;
        asm mov ax,08h;
        asm mov cx,00h;
        asm mov dx,1d0h;
        asm int 33h;
do
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* Read Mouse Position */
        asm mov ax,03h;
        asm int 33h;
        asm mov button,bx;
        asm mov x,cx;
        asm mov y,dx;
/* Display Position */
itoa(x,x_str,10);
itoa(y,y_str,10);
/* Check Keyboard Press */
if (kbhit())
{
        asm mov ax,00h;
        asm int 16h;
        asm mov key,al;
        if (key==17) {stop=1;}
        if (key=='d') {ndelete();a=0;b=0;}
        if (key=='i') {insert();a=0;b=0;}
        if (key=='s') {save();}
        if (key=='l') {load();}
        if (key=='c') {clear();}
        if (key==9) {increase();}
        if (key==8) {decrease();}
}

if (b<87 && x<191) && (y>87 && y<376))
{
if (x-x1!=0 || y-y1!=0)
{
setcolor(10);
pos_x(x);
pos_y(y);
setfillstyle(SOLID_FILL,3);
bar(265,345,297,375);
x1=x;
y1=y;
}
if (x-x1==0 && y-y1==0)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
setcolor(10);
pos_x(x);
pos_y(y);
}

/* Check Mouse Click */
if (button==1 && (x2-a!=0 || y2-b!=0))
{
a=x2;
b=y2;
if (color==5) {charecter[index].output[x2-1][y2-1]=1;}
else {charecter[index].output[x2-1][y2-1]=0;}
x_pos(x2);
y_pos(y2);
setfillstyle(SOLID_FILL,color);
disable_cursor( );
bar(xa1,ya1,xa2,ya2);
enable_cursor( );
}
}
/* End of loop */
}
while (stop!=1);
}
else
{
        outtextxy(220,200,"Mouse Driver is not Install!");
        sound(2200);
        delay(33);
        nosound( );
        getch( );
}
}

void table( )
{
int i;
unsigned int back;
setbkcolor(3);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=0;i<9;i++)
{
setcolor(14);
line(87+13*i,88,87+13*i,376);
}
for (i=0;i<25;i++)
{
line(87,88+12*i,191,88+12*i);
}
setcolor(8);
line(87,328,191,328);
line(87,220,191,220);
disable_cursor( );
back=imagesize(87,88,191,376);
backup=farmalloc(back);
getimage(87,88,191,376,backup);
enable_cursor( );
outtextxy(237,350,"Row.");
outtextxy(237,365,"Col.");
setcolor(10);
outtextxy(273,350,"0");
outtextxy(273,365,"0");
setcolor(15);
outtextxy(360,100,"Edit Font Mode.");
outtextxy(330,130,"Key (I) = Insert Mode.");
outtextxy(330,155,"Key (D) = Delete Mode.");
outtextxy(330,180,"Key (S) =  Save File.");
outtextxy(330,205,"Key (L) =  Load File.");
outtextxy(330,230,"Key (Ctrl q) =  Quit.");
outtextxy(330,255,"Key (Insert) =  Save.");
outtextxy(330,275,"Key (C) =  Clear.");
outtextxy(330,300,"Key (Space Bar = Insert Charecter.");
setcolor(17);
outtextxy(347,380,"Font Number: ");
outtextxy(457,380,"0.");
}

void clear( )
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i,j;
for (i=0;i<8;i++)
{
for (j=0;j<24;j++)
{
x_pos(i+1);
y_pos(j+1);
character[index].output[i][j]=0;
setfillstyle(SOLID_FILL,3);
bar(xa1,ya1,xa2,ya2);
}
}
}

void ndelete( )
{
disable_cursor( );
color=3;
enable_cursor( );
}

void insert( )
{
disable_cursor( );
color=5;
enable_cursor( );
}

void increase( )
{
int i,j;
char string[3];
disable_cursor( );
putimage(87,88,backup,COPY_PUT);
if (index<255)
{
index=index+1;
for (j=0;j<24;j++)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for (i=0;i<8;i++)
{
x_pos(i+1);
y_pos(j+1);
if (charecter[index].output[i][j]==1)
{
setfillstyle(SOLID_FILL,5);
bar(xa1,ya1,xa2,ya2);
}
}
}
index2=index2++;
itoa(index2,string,10);
setfillstyle(SOLID_FILL,3);
bar(456,375,485,395);
outtextxy(457,380,string);
enable_cursor( );
}
else
{
outtextxy(357,400,"Font is now Number 255.");
delay(2000);
setfillstyle(SOLID_FILL,3);
bar(356,397,550,450);
}
}

void decrease( )
{
int i,j;
char string[3];
if (index>0)
{
index=index-1;
disable_cursor( );
index2=index2--;
itoa(index2,string,10);
setfillstyle(SOLID_FILL,3);
bar(455,375,485,395);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outtextxy(457,380,string);
putimage(87,88,backup,COPY_PUT);
for (j=0;j<24;j++)
{
for (i=0;i<8;i++)
{
x_pos(i+1);
y_pos(j+1);
if (charecter[index].output[i][j]==1)
{
setfillstyle(SOLID_FILL,5);
bar(xa1,ya1,xa2,ya2);
}
}
}
enable_cursor( );
} // end if loop

else
{
outtextxy(357,400,"Font is now Number 0.");
delay(200);
setfillstyle(SOLID_FILL,3);
bar(356,397,550,450);
}
}

void save( )
{
FILE *fp;
char file_name[8],temp1;
int temp[8];
void far *back1;
unsigned int back;
int i,j,k;
disable_cursor( );
back=imagesize(37,88,579,280);
back1=farmalloc(back);
getimage(37,88,579,280,back1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,back1,COPY_PUT);
enable_cursor( );
if (chk_name!=1)
{
fp=fopen(file_name,"wb");
for (k=0;k<255;k++)
{
for (j=0;j<24;j++)
{
for (i=0;i<8;i++)
{
if (charecter[k].output[i][j]==0) {temp[i]=0;}
if (charecter[k].output[i][j]==1) {temp[i]=1;}
if (i==7)
{
temp1=temp[0]*128+temp[1]*64+temp[2]*32+temp[3]*16+temp[4]*8+temp[5]*4
      +temp[6]*2+temp[7];
putc(temp1,fp);
} //end if
} //end loop i
} //end loop j
} //end loop k
}
farfree(back1);
fclose(fp);
}

void table_save(char name[8])
{
char *n1;
char n;
int i=0;
setfillstyle(SOLID_FILL,15);
disable_cursor( );
bar(170,130,464,190);
setcolor(5);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
line(172,132,462,132);
line(172,132,172,188);
line(172,170,462,170);
line(172,188,462,188);
line(462,132,462,188);
setcolor(17);
outtextxy(360,175,"Enter = save");
outtextxy(180,175,"Esc = abort");
setcolor(3);
outtextxy(180,148,"Enter your file's name : ");
enable_cursor( );
moverel(20,30);
name[0]='\0';
setcolor(1);
while (n!=13 && n!=27)
{
n=getche( );

if (n==8 && i>0)
{
i=i-;
disable_cursor( );
bar(379+8*i,146,389+8*i,155);
enable_cursor();
name[i]='\0';
}
if (n==27) {chk_name=1;}

if (n!=13 && n!=27 && i<8 && n!=8)
{
sprintf(n1,"%c",n);
disable_cursor( );
outtextxy(380+8*i,148,n1);
enable_cursor( );
strcat(name,n1);
i=i++;
}
if (i>7) {sound(220);delay(33);nosound();}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void load( )
{
FILE *l;
char file_name[8];
unsigned char temp;
int point,decode[8],x=0;
void far *back1;
unsigned int back;
int i,k,j;
disable_cursor( );
back=imagesize(37,88,597,280);
back1=farmalloc(back);
getimage(37,88,597,280,back1);
enable_cursor( );
table_save(file_name);
disable_cursor( );
putimage(37,88,back1,COPY_PUT);
enable_cursor( );
if (chk_name!=1)
{
l=fopen(file_name,"rb");
for (k=0;k<256;k++)
{
for (j=0;j<24;j++)
{
temp=getc(l);
if (temp>=128) {decode[x]=1;temp=temp-128;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=64) {decode[x]=1;temp=temp-64;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=32) {decode[x]=1;temp=temp-32;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=16) {decode[x]=1;temp=temp-16;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=8) {decode[x]=1;temp=temp-8;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=4) {decode[x]=1;temp=temp-4;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=2) {decode[x]=1;temp=temp-2;x=x++;} else {decode[x]=0;x=x++;}
if (temp>=1) {decode[x]=1;} else {decode[x]=0;}
x=0;
for (point=0;point<8;point++)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
character[k].output[point][j]=decode[point];
}
} //end of loop j
} //end of for loop
} //end if
disable_cursor( );
for (j=0;j<24;j++)
{
for (i=0;i<8;i++)
{
x_pos(i+1);
y_pos(j+1);
if (character[0].output[i][j]==1)
{
setfillstyle(SOLID_FILL,5);
bar(xa1,ya1,xa2,ya2);
}
} // end of for loop
enable_cursor( );
}
farfree(back1);
fclose(l);
}

void initial( )
{
int i,j;
for (i=0;i<8;i++)
for (j=0;j<24;j++)
{
character[index].output[i][j]=0;
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ.
วิธีการใช้โปรแกรมบนคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้โปรแกรม PROJECT.EXE

ก่อนที่จะใช้โปรแกรมนี้ จะต้องทำการติดตั้งเมาส์ไดรเวอร์เสียก่อน ส่วนการใช้งานต่าง ๆ จะมีรายละเอียดดังต่อไปนี้

1. โหมดการทำงานเริ่มต้น

เมื่อเข้าสู่ DOS PROMPT พิมพ์คำว่า 'PROJECT' เพื่อเรียกแฟ้มข้อมูลชื่อ PROJECT.EXE

A:\>PROJECT

จะเข้าสู่โหมดการทำงานเริ่มต้นของโปรแกรม ที่ผู้ใช้สามารถออกแบบรูปภาพ หรือพิมพ์ตัวอักษรความยาวไม่เกินความกว้างของบอร์ดแสดงข้อมูลบนตารางจำลอง ดังแสดงในรูปที่ จ-1

เมื่อต้องการทราบคีย์ที่ใช้งานในโหมดการทำเริ่มต้นนั้น สามารถเรียกดูด้วยการกดคีย์ Ctrl H ดังแสดงในรูปที่ จ-2 ซึ่งแต่ละคีย์นั้นจะมีรายละเอียดอธิบายได้ดังนี้

Ctrl R เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีแดง

Ctrl G เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีเขียว

Ctrl O เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีส้ม

Ctrl M เป็นการเรียกวินโดว์ที่แสดง Button ต่าง ๆ ที่ใช้เมาส์คลิก ดังแสดงในรูปที่ จ-3 ซึ่งการคลิกที่ Button ต่าง ๆ นี้เพื่อใช้ในการสร้าง และจัดการข้อมูลดังจะแสดงรายละเอียดได้ดังนี้

Red การคลิกเมาส์ที่ Button นี้จะได้ผลเหมือนกับการกดคีย์ Ctrl R

Green การคลิกเมาส์ที่ Button นี้จะได้ผลเหมือนกับการกดคีย์ Ctrl G

Orange การคลิกเมาส์ที่ Button นี้จะได้ผลเหมือนกับการกดคีย์ Ctrl O

Save เมื่อผู้ใช้ต้องการเก็บข้อมูลรูปภาพ หรือตัวอักษรที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ ในรูปของแฟ้มข้อมูล โดยตั้งชื่อแฟ้มข้อมูลด้วยความยาวไม่เกิน 8 ตัวอักษร ซึ่งผลลัพธ์เมื่อคลิกเมาส์ที่ Button นี้ แสดงดังรูปที่ จ-4 และเมื่อพิมพ์ชื่อแฟ้มข้อมูลเสร็จแล้วก็กด Enter

Load เมื่อผู้ใช้ต้องการเรียกแฟ้มข้อมูลรูปภาพ หรือตัวอักษรที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ หลังจากได้มีการเก็บในรูปของแฟ้มข้อมูลไปแล้วดังกล่าว โดยรูปที่ จ-5 จะแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อคลิกเมาส์ที่ Button นี้

Insert เป็นการกำหนดสถานะให้กับเมาส์ว่า การคลิกเมาส์แต่ละครั้งแทนการทำให้แอลอีดีสว่าง ซึ่งจะสว่างเป็นสีใดนั้นก็กำหนดได้ด้วยการเลือกคลิกที่ Button 'Red' หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'Green' หรือ 'Orange' เพื่อให้สว่างเป็นสีแดง สีเขียว หรือสีส้ม ตามลำดับ หรืออาจใช้การกดคีย์บนแป้นพิมพ์ดังอธิบายมาแล้วก็ได้

Delete เป็นการกำหนดสถานะให้กับเมาส์ว่า การคลิกเมาส์แต่ละครั้งแทนการทำให้แอลอีดีดับ

Clear เมื่อคลิกเมาส์ที่ Button นี้ เป็นการดับแอลอีดีทุกดวงบนตารางจำลอง

Exit เมื่อผู้ใช้ต้องการออกจากโปรแกรม สามารถทำได้ด้วยการคลิกเมาส์ที่ Button นี้

Ctrl S การกดคีย์นี้จะได้ผลเหมือนกับการคลิกเมาส์ที่ Button 'Save'

Ctrl L การกดคีย์นี้จะได้ผลเหมือนกับการคลิกเมาส์ที่ Button 'Load'

Ctrl T เมื่อผู้ใช้ต้องการจะเปลี่ยนจากการใช้เมาส์ มาใช้การพิมพ์บนแป้นพิมพ์เพื่อพิมพ์ตัวอักษร โดยผู้ใช้สามารถพิมพ์ตัวอักษรได้ทั้งภาษาไทย และภาษาอังกฤษ แต่จะต้องมีความยาวไม่เกินความกว้างของบอร์ดแสดงผล โดยผู้ใช้สามารถเปลี่ยนฟังก์ชั่นคีย์ภาษาไทย หรือภาษาอังกฤษได้ด้วยการกดคีย์ [~ %] ซึ่งสถานะของคีย์จะถูกแสดงที่ตำแหน่งมุมขวาด้านบนของหน้าจอคอมพิวเตอร์

Ctrl C การกดคีย์นี้จะได้ผลเหมือนกับการคลิกเมาส์ที่ Button 'Clear'

Ctrl D การกดคีย์นี้จะได้ ผลเหมือนกับการคลิกเมาส์ที่ Button 'Delete'

Ctrl I การกดคีย์นี้จะได้ผลเหมือนกับการคลิกเมาส์ที่ Button 'Insert'

Ctrl Z เมื่อผู้ใช้ต้องการส่งข้อมูลรูปภาพ หรือตัวอักษรที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ออกพอร์ตเพื่อไปแสดงเป็นภาพนิ่งบนบอร์ดแสดงข้อมูล โดยรูปที่ จ-6 จะแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อกดคีย์นี้

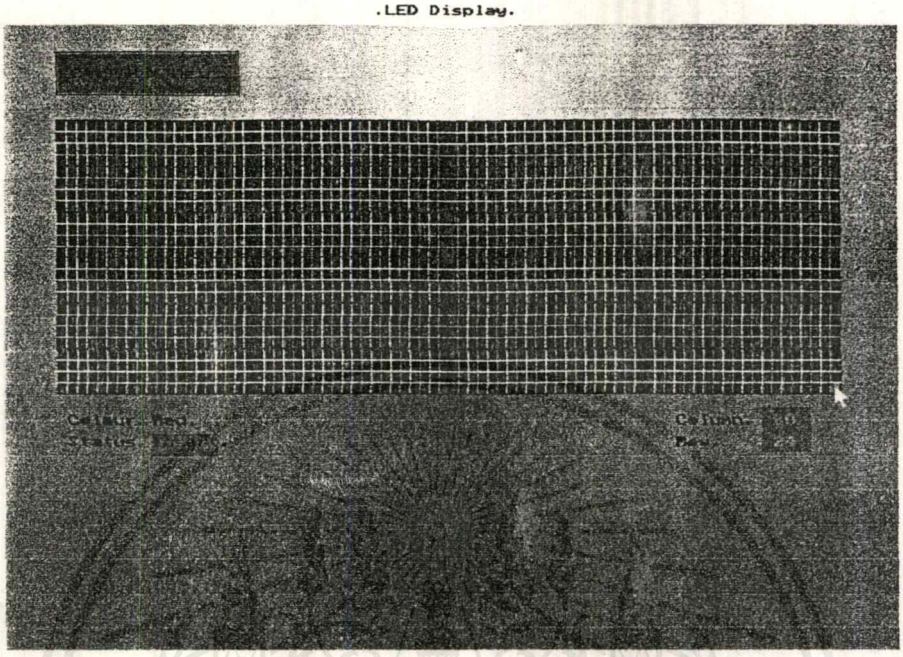
Ctrl P เมื่อผู้ใช้ต้องการเข้าไปทำงานในโหมดข้อความต่อเนื่อง ดังจะอธิบายวิธีการใช้ต่อไปในข้อ 2.

Ctrl F เมื่อผู้ใช้ต้องการเข้าไปทำงานในโหมดรูปภาพเคลื่อนไหว ดังจะอธิบายวิธีการใช้ต่อไปในข้อ 3.

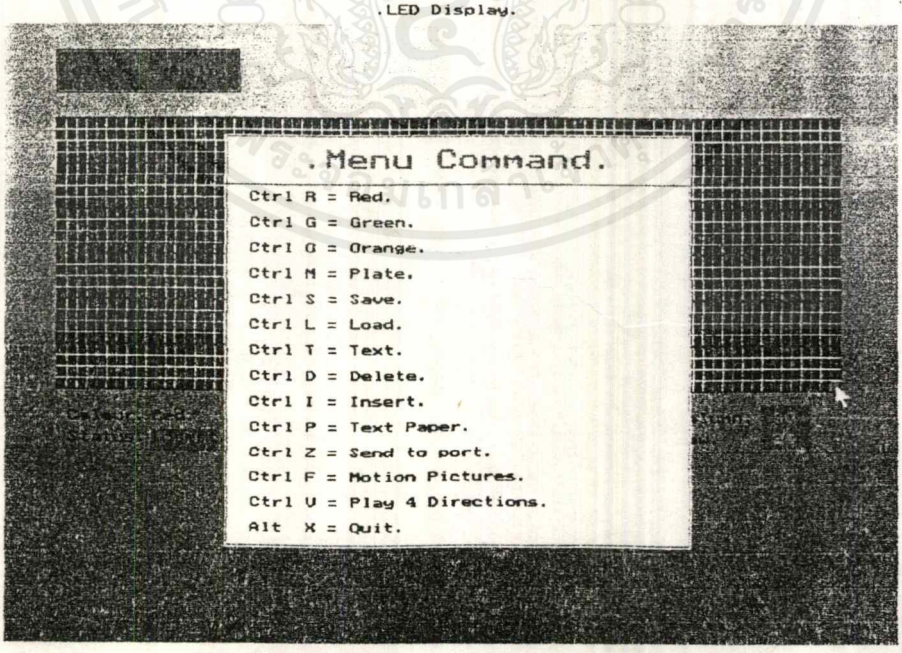
Ctrl V เมื่อผู้ใช้ต้องการเข้าไปทำงานในโหมดการแสดงข้อมูลใน 4 ทิศทาง ดังจะอธิบายวิธีการใช้ต่อไปในข้อ 4.

Alt X การกดคีย์นี้เป็นการออกจากโปรแกรม ซึ่งอาจทำได้ด้วยการคลิกเมาส์ที่ Button 'Exit'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

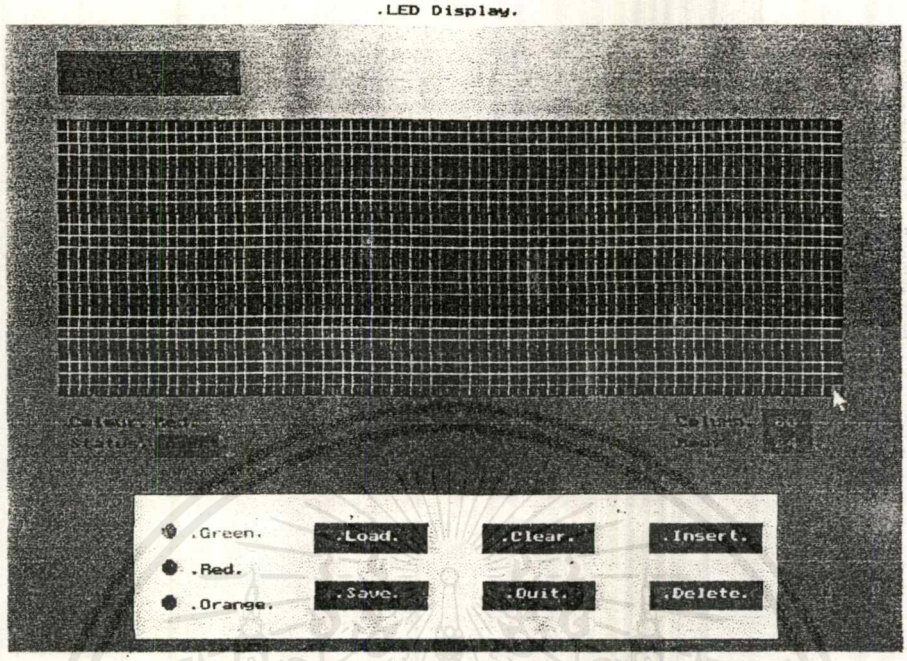


รูปที่ จ-1 แสดงโหมดการทำงานเริ่มต้น หลังจากเรียกโปรแกรม PROJECT.EXE

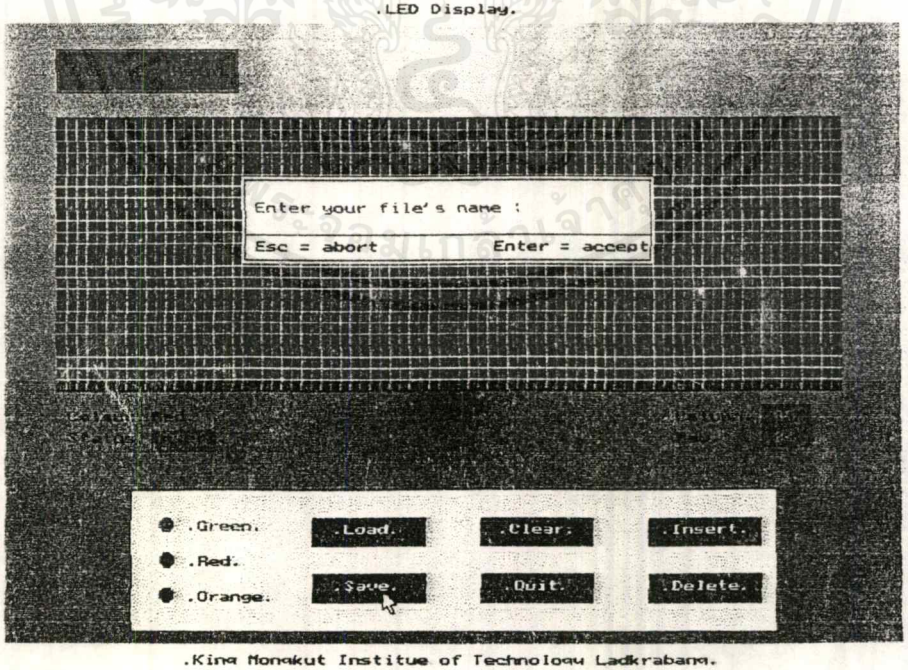


รูปที่ จ-2 แสดงผลลัพธ์หน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อกดคีย์ Ctrl H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

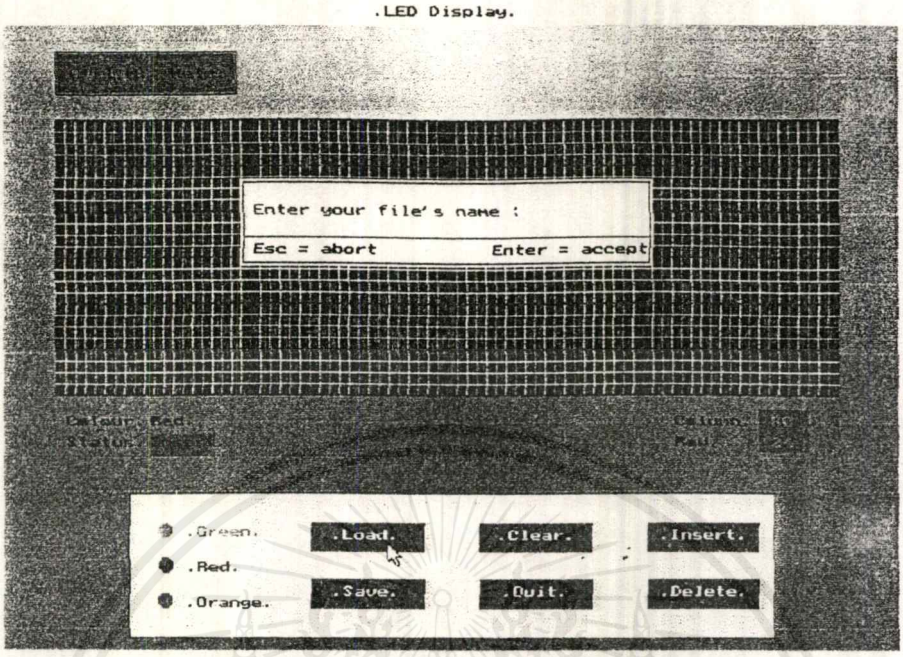


รูปที่ จ-3 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อกดคีย์ Ctrl M

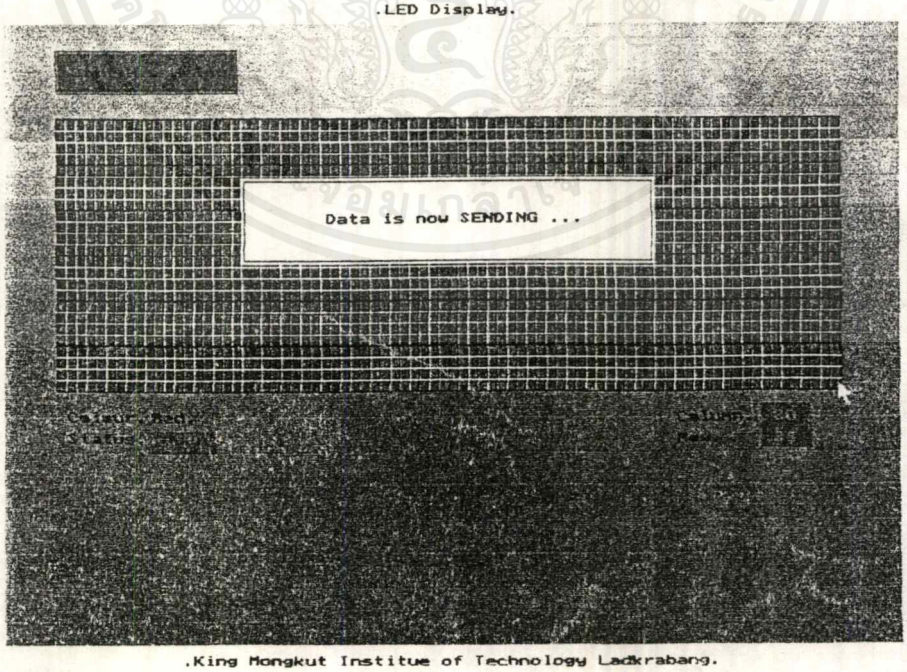


รูปที่ จ-4 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น
เมื่อคลิกเมาส์ที่ Button 'Save'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-5 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อคลิกเมาส์ที่ Button 'Load'



รูปที่ จ-6 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น เมื่อกดคีย์ Ctrl Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โหมดข้อความต่อเนื่อง

เมื่อผู้ใช้กดคีย์ Ctrl P ในโหมดการทำงานเริ่มต้น จะเป็นการเข้าสู่โหมดข้อความต่อเนื่อง โดยในโหมดการทำงานนี้ ผู้ใช้สามารถส่งข้อความขนาดยาว ๆ ออกแสดงผลได้ แต่ไม่เกิน 414 ตัวอักษร ดังแสดงในรูปที่ จ-7 ซึ่งจะประกอบไปด้วยแผ่นกระดาษจำลองที่ใช้พิมพ์ข้อความลงไป และคีย์ที่เป็นฟังก์ชั่นการใช้งานต่าง ๆ ซึ่งมีหน้าที่ดังนี้

Ctrl R เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีแดง

Ctrl G เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีเขียว

Ctrl O เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีส้ม

K กดคีย์นี้เมื่อต้องการเริ่มพิมพ์ข้อความ ดังแสดงในรูปที่ จ-8

S เมื่อผู้ใช้ต้องการเก็บข้อความที่พิมพ์บนแผ่นกระดาษจำลองบนหน้าจอคอมพิวเตอร์ ในรูปของแฟ้มข้อมูล โดยตั้งชื่อแฟ้มข้อมูลด้วยความยาวไม่เกิน 8 ตัวอักษร ซึ่งผลลัพธ์เมื่อกดคีย์นี้จะแสดงดังรูปที่ จ-9 และเมื่อพิมพ์ชื่อแฟ้มข้อมูลเสร็จแล้ว ให้กด Enter

L เมื่อผู้ใช้ต้องการเรียกแฟ้มข้อมูลของข้อความที่พิมพ์ไว้บนแผ่นกระดาษจำลองบนหน้าจอคอมพิวเตอร์ หลังจากได้มีการเก็บในรูปของแฟ้มข้อมูลไปแล้วดังกล่าว โดยรูปที่ จ-9 ได้แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อกดคีย์นี้

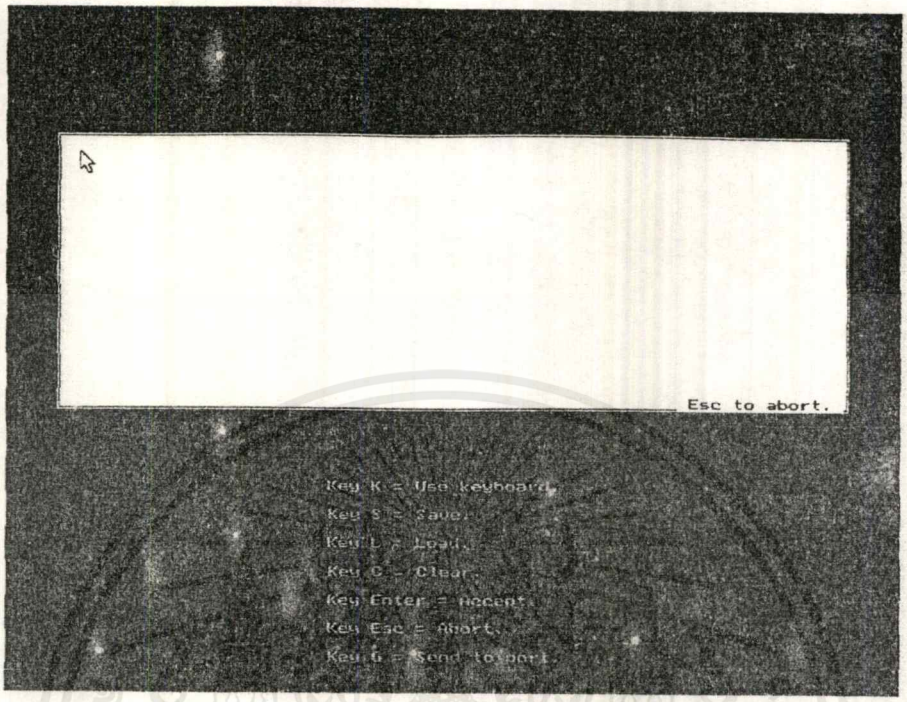
C กดคีย์นี้เมื่อต้องการลบข้อความทั้งหมดที่พิมพ์บนแผ่นกระดาษจำลอง

Enter หลังจากพิมพ์ข้อความเสร็จทุกครั้งให้กด Enter

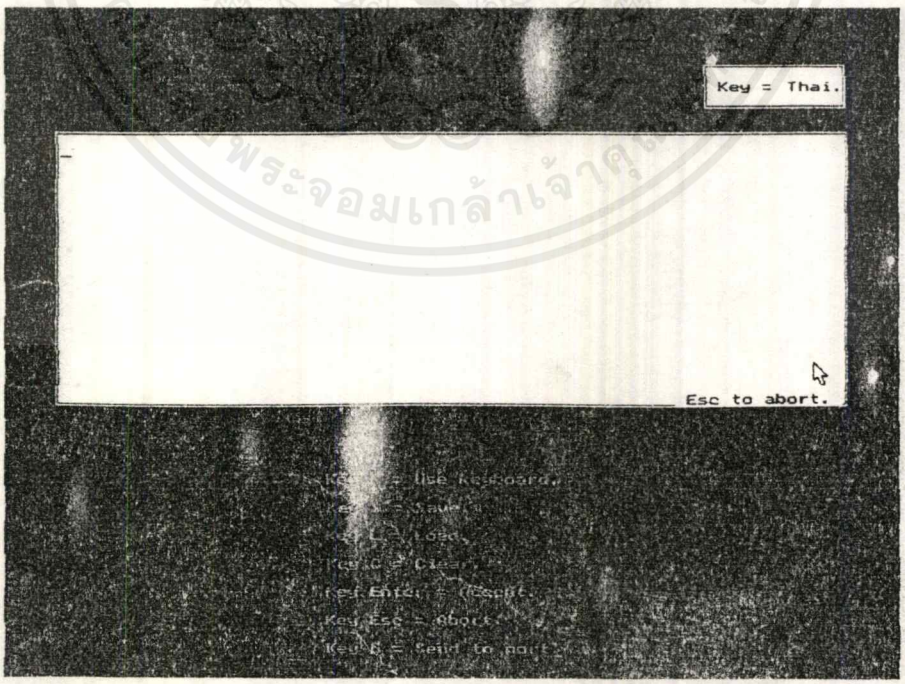
Esc กดคีย์นี้เมื่อต้องการออกจากโหมดการทำงานนี้ กลับเข้าสู่โหมดการทำงานเริ่มต้น

G เมื่อผู้ใช้ต้องการส่งข้อมูลของข้อความที่พิมพ์ไว้บนแผ่นกระดาษจำลองบนหน้าจอคอมพิวเตอร์ออกพอร์ตเพื่อไปแสดงเป็นข้อความต่อเนื่องบนบอร์ดแสดงข้อมูล โดยรูปที่ จ-10 จะแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อกดคีย์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

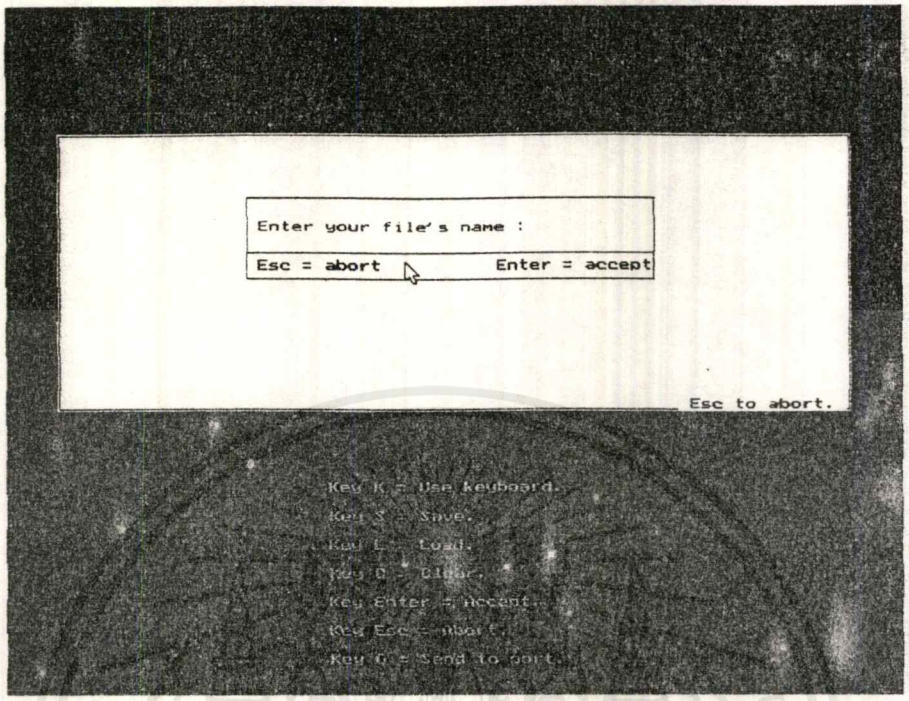


รูปที่ จ-7 แสดงโหมดข้อความต่อเนื่อง จากการกดคีย์ Ctrl P ในโหมดการทำงานเริ่มต้น

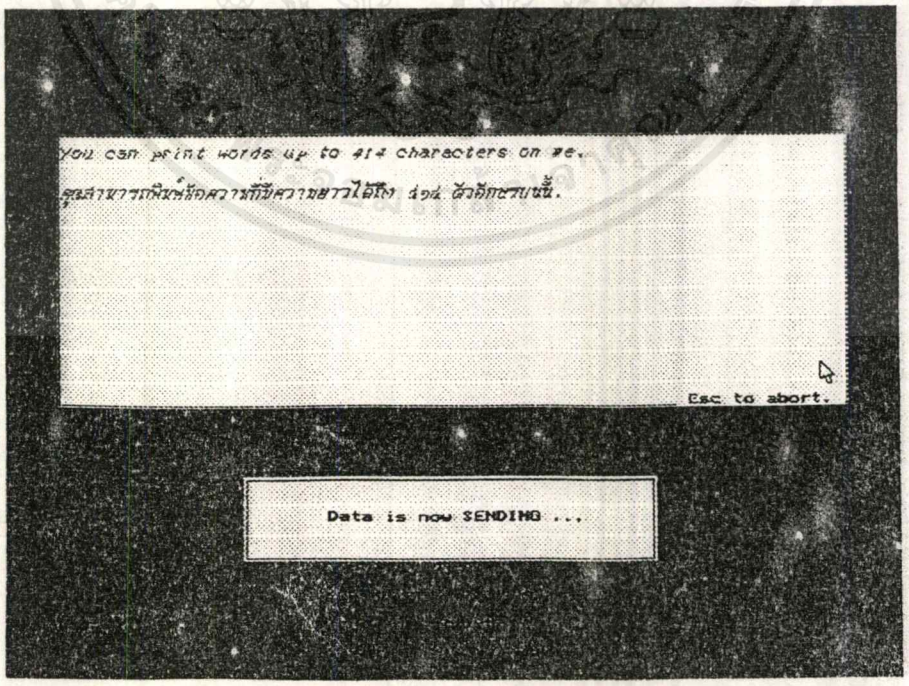


รูปที่ จ-8 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดข้อความต่อเนื่อง เมื่อกดคีย์ K

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-9 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดข้อความต่อเนื่อง เมื่อกดคีย์ S หรือ L



รูปที่ จ-10 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดข้อความต่อเนื่อง เมื่อกดคีย์ G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. <u>โหมดรูปภาพเคลื่อนไหว</u>

เมื่อผู้ใช้กดคีย์ Ctrl F ในโหมดการทำงานเริ่มต้น จะเป็นการเข้าสู่โหมดรูปภาพเคลื่อนไหว โดยในโหมดการทำงานนี้ ผู้ใช้สามารถสร้างรูปภาพหลาย ๆ เฟรมเพื่อออกแสดงต่อเนื่องกันเป็นรูปภาพเคลื่อนไหว ดังแสดงในรูปที่ จ-11 ซึ่งจะประกอบไปด้วยตารางจำลองที่ใช้ออกแบบรูปภาพลงไป

เมื่อต้องการทราบคีย์ที่ใช้งานในโหมดนี้ สามารถเรียกดูด้วยการกดคีย์ Ctrl H ดังแสดงในรูปที่ จ-12 ซึ่งแต่ละคีย์นั้นจะมีรายละเอียดอธิบายได้ดังนี้

Ctrl R เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีแดง

Ctrl G เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีเขียว

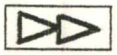
Ctrl O เป็นการกำหนดให้การคลิกเมาส์แต่ละครั้ง แทนการทำให้แอลอีดีสว่างเป็นสีส้ม

Ctrl P เป็นการเรียกวินโดว์ที่แสดง Button ต่าง ๆ ที่ใช้เมาส์คลิก ดังแสดงในรูปที่ จ-13 ซึ่งการคลิกที่ Button ต่าง ๆ นี้จะเป็นฟังก์ชั่นในการนำรูปภาพทุก ๆ เฟรมที่ผู้ใช้ออกแบบไว้ ออกแสดงบนหน้าจอคอมพิวเตอร์ ดังจะแสดงรายละเอียดได้ดังนี้

เมื่อคลิกเมาส์ที่ Button นี้ จะเป็นการสั่งให้แสดงภาพทุกเฟรมอย่างต่อเนื่องบนหน้าจอคอมพิวเตอร์

เมื่อคลิกเมาส์ที่ Button นี้ จะเป็นการสั่งให้หยุดแสดงภาพโดยจะหยุดที่ภาพในเฟรมสุดท้าย

เมื่อคลิกเมาส์ที่ Button นี้ จะเป็นการสั่งให้แสดงภาพก่อนหน้าภาพในเฟรมที่แสดงอยู่


เมื่อคลิกเมาส์ที่ Button นี้ จะเป็นการสั่งให้แสดงภาพถัดจากภาพในเฟรมที่แสดงอยู่

Ctrl S เมื่อผู้ใช้ต้องการเก็บข้อมูลรูปภาพทุกเฟรมที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ ในรูปของแฟ้มข้อมูล โดยตั้งชื่อแฟ้มข้อมูลด้วยความยาวไม่เกิน 8 ตัวอักษร ซึ่งผลลัพธ์เมื่อกดคีย์นี้แสดงดังรูปที่ จ-14 และเมื่อพิมพ์ชื่อแฟ้มข้อมูลเสร็จแล้ว ก็กด Enter

Ctrl L เมื่อผู้ใช้ต้องการเรียกแฟ้มข้อมูลรูปภาพทุกเฟรมที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ หลังจากได้มีการเก็บในรูปของแฟ้มข้อมูลไปแล้วดังกล่าว โดยรูปที่ จ-14 จะแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อกดคีย์นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ctrl I เป็นการกำหนดสถานะให้กับเมาส์ว่ า การคลิกเมาส์แต่ละครั้งแทนการทำให้แอลอีดีสว่าง ซึ่งจะสว่างเป็นสีใดนั้นก็กำหนดได้ด้วยการเลือกกดคีย์ Ctrl R หรือ Ctrl G หรือ Ctrl O เพื่อให้สว่างเป็นสีแดง สีเขียว หรือสีส้ม ตามลำดับ

Ctrl D เป็นการกำหนดสถานะให้กับเมาส์ว่า การคลิกเมาส์แต่ละครั้งแทนการทำให้แอลอีดีดับ

Alt C การกดคีย์นี้เป็นการดับหลอดแอลอีดีทุกดวงบนตารางจำลอง

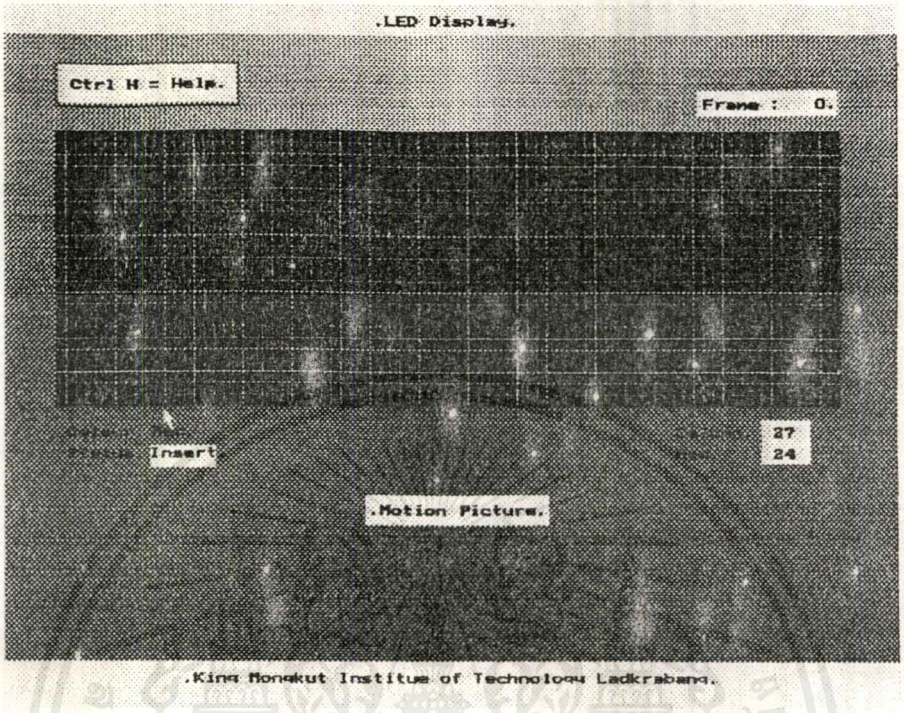
Ctrl Z เมื่อผู้ใช้ต้องการส่งข้อมูลรูปภาพที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ออกพอร์ตเพื่อไปแสดงอย่างต่อเนื่องเป็นภาพเคลื่อนไหวบนบอร์ดแสดงข้อมูล โดยผู้ใช้สามารถกำหนดจำนวนเฟรมรูปภาพที่จะส่งไปแสดงได้ โดยเมื่อกดคีย์นี้จะได้ผลลัพธ์บนหน้าจอคอมพิวเตอร์ดังแสดงในรูปที่ จ-15

Tab กดคีย์นี้เมื่อผู้ใช้ออกแบบรูปภาพในแต่ละเฟรมเสร็จแล้ว แล้วต้องการออกแบบรูปภาพในเฟรมถัดไป

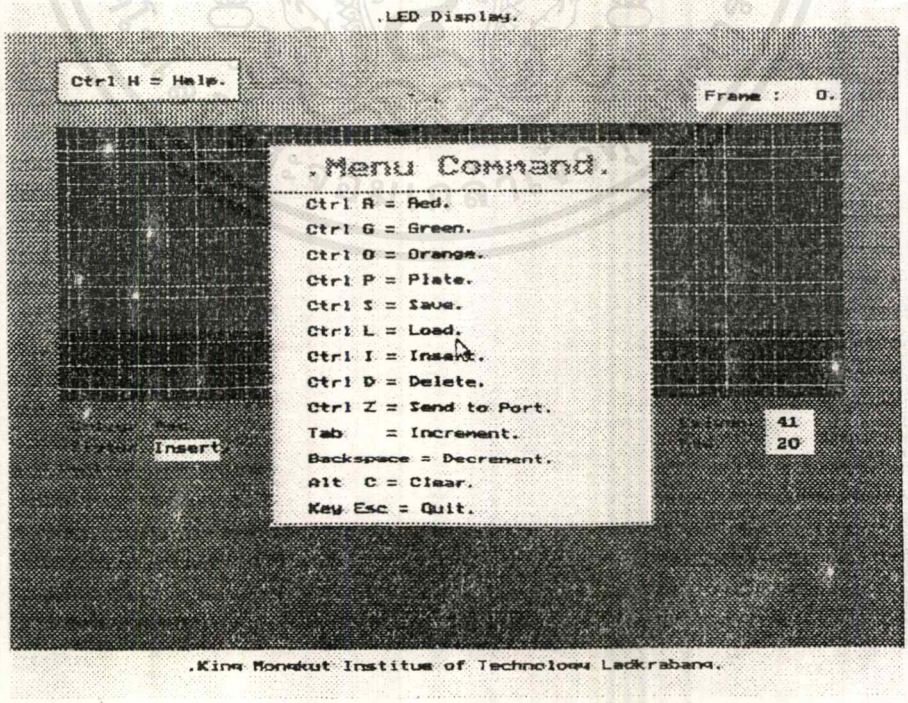
Backspace กดคีย์นี้เมื่อผู้ใช้ต้องการกลับไปแก้ไขรูปภาพในเฟรมก่อนหน้า

Esc กดคีย์นี้เมื่อต้องการออกจากโหมดการทำงานนี้ ซึ่งจะกลับเข้าสู่โหมดการทำงานเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

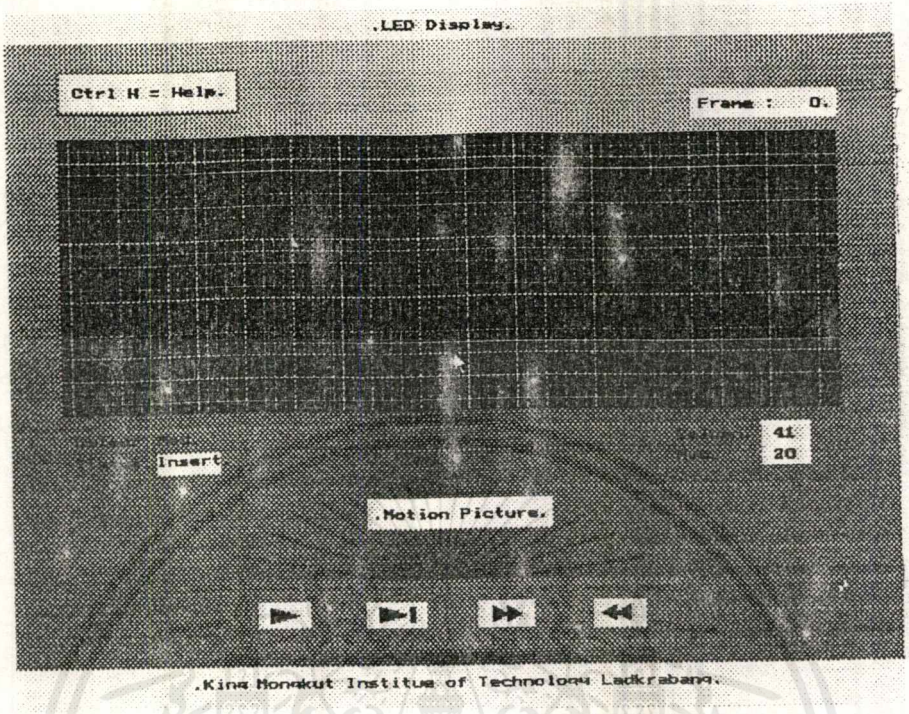


รูปที่ จ-11 แสดงโหมดภาพเคลื่อนไหว จากการกดคีย์ Ctrl F ในโหมดการทำงานเริ่มต้น

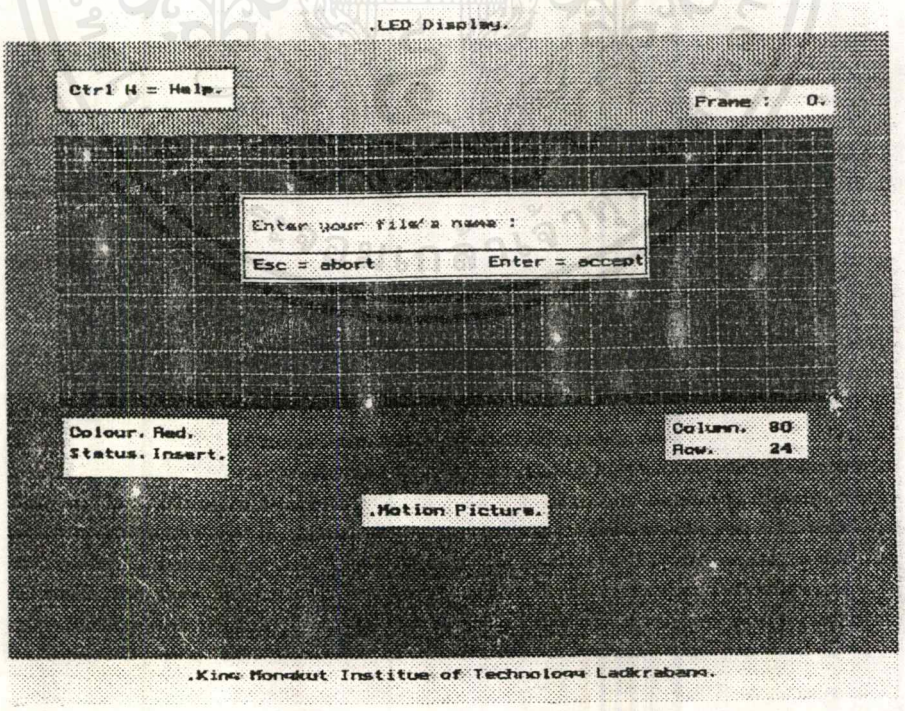


รูปที่ จ-12 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

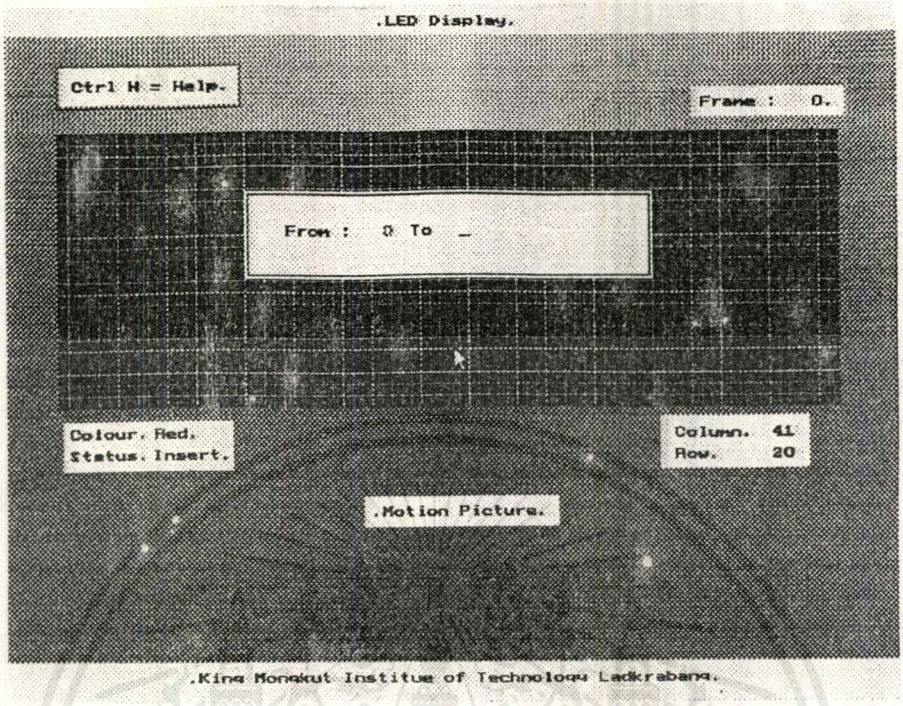


รูปที่ จ-13 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl P



รูปที่ จ-14 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว
เมื่อกดคีย์ Ctrl S หรือ Ctrl L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-15 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดภาพเคลื่อนไหว เมื่อกดคีย์ Ctrl Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โหมดการแสดงข้อมูลใน 4 ทิศทาง

เมื่อผู้ใช้กดคีย์ Ctrl V ในโหมดการทำงานเริ่มต้น จะเป็นการเข้าสู่โหมดการแสดงข้อมูลใน 4 ทิศทาง ดังแสดงในรูปที่ จ-16 โดยในโหมดการทำงานนี้ ผู้ใช้สามารถเรียกแฟ้มข้อมูลรูปภาพที่ได้ออกแบบไว้ในโหมดการทำงานเริ่มต้นเพื่อออกแสดงเป็นรูปภาพเคลื่อนที่ใน 4 ทิศทาง คือจากซ้ายไปขวา จากขวาไปซ้าย จากบนลงล่าง หรือจากล่างขึ้นบน

สำหรับคีย์ต่าง ๆ ที่เป็นฟังก์ชั่นใช้งานในโหมดการทำงานนี้ สามารถเรียกดูได้จากการกดคีย์ Ctrl H ดังแสดงในรูปที่ จ-17 ซึ่งมีรายละเอียดดังต่อไปนี้

Ctrl L เมื่อผู้ใช้ต้องการเรียกแฟ้มข้อมูลรูปภาพ หรือตัวอักษรที่ทำไว้บนตารางจำลองบนหน้าจอคอมพิวเตอร์ในโหมดการทำงานเริ่มต้น หลังจากได้มีการเก็บในรูปของแฟ้มข้อมูลไปแล้ว เพื่อนำออกแสดงเป็นข้อมูลเคลื่อนที่ใน 4 ทิศทางดังกล่าว โดยรูปที่ จ-18 จะแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อกดคีย์นี้

U เมื่อผู้ใช้ต้องการนำข้อมูลที่เรียกขึ้นมาแล้ว ให้ออกแสดงเคลื่อนที่จากล่างขึ้นบน

D เมื่อผู้ใช้ต้องการนำข้อมูลที่เรียกขึ้นมาแล้ว ให้ออกแสดงเคลื่อนที่จากบนลงล่าง

R เมื่อผู้ใช้ต้องการนำข้อมูลที่เรียกขึ้นมาแล้ว ให้ออกแสดงเคลื่อนที่จากขวาไปซ้าย

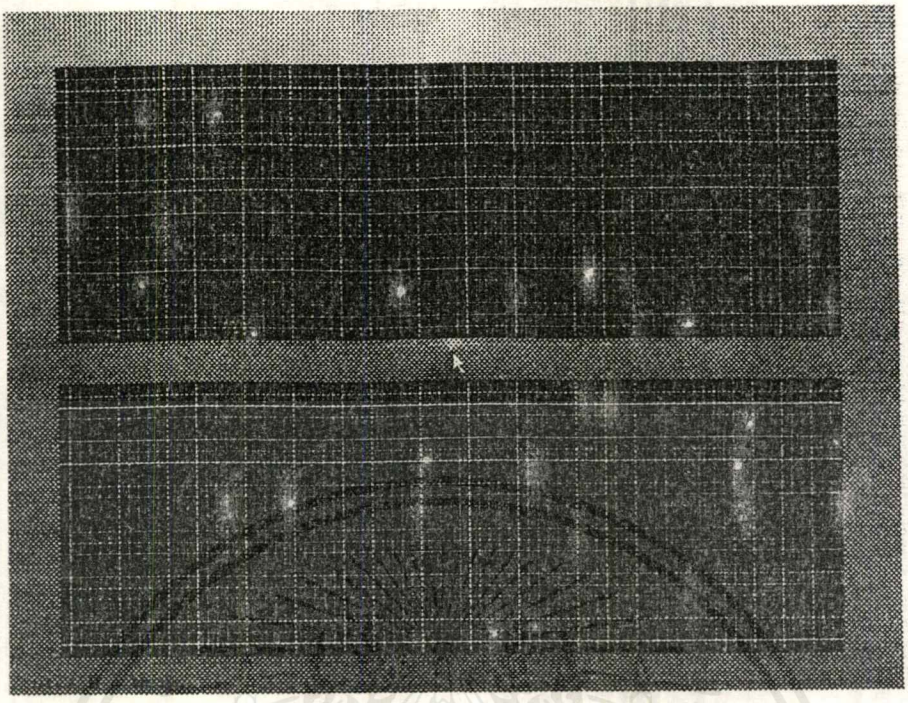
L เมื่อผู้ใช้ต้องการนำข้อมูลที่เรียกขึ้นมาแล้ว ให้ออกแสดงเคลื่อนที่จากซ้ายไปขวา

หลังจากกดคีย์ U หรือ D หรือ R หรือ L จ ะได้ผลลัพธ์ดังแสดงในรูปที่ จ-19 ผู้ใช้จะต้องใส่ตัวเลข ซึ่งเป็นการบ่งบอกจำนวนเฟรมรูปภาพจากการแบ่งข้อมูลเพื่อให้เกิดข้อมูลที่นำออกแสดงเป็นภาพเคลื่อนที่ได้ โดยในกรณีที่ต้องการให้ข้อมูลออกแสดงเคลื่อนที่จากล่างขึ้นบน (โดยการกดคีย์ U) หรือจากบนลงล่าง (โดยการกดคีย์ D) ผู้ใช้จะสามารถใส่จำนวนเฟรมได้ไม่เกิน 24 เฟรม (0 ถึง 24) ส่วนในกรณีที่ต้องการให้ข้อมูลออกแสดงเคลื่อนที่จากขวาไปซ้าย (โดยการกดคีย์ R) หรือจากซ้ายไปขวา (โดยการกดคีย์ L) ผู้ใช้จะสามารถใส่จำนวนเฟรมได้ไม่เกิน 80 เฟรม (0 ถึง 80) ที่เป็นเช่นนี้เนื่องมาจากบอร์ดแสดงข้อมูลมีขนาดเป็น 24x80 จุด

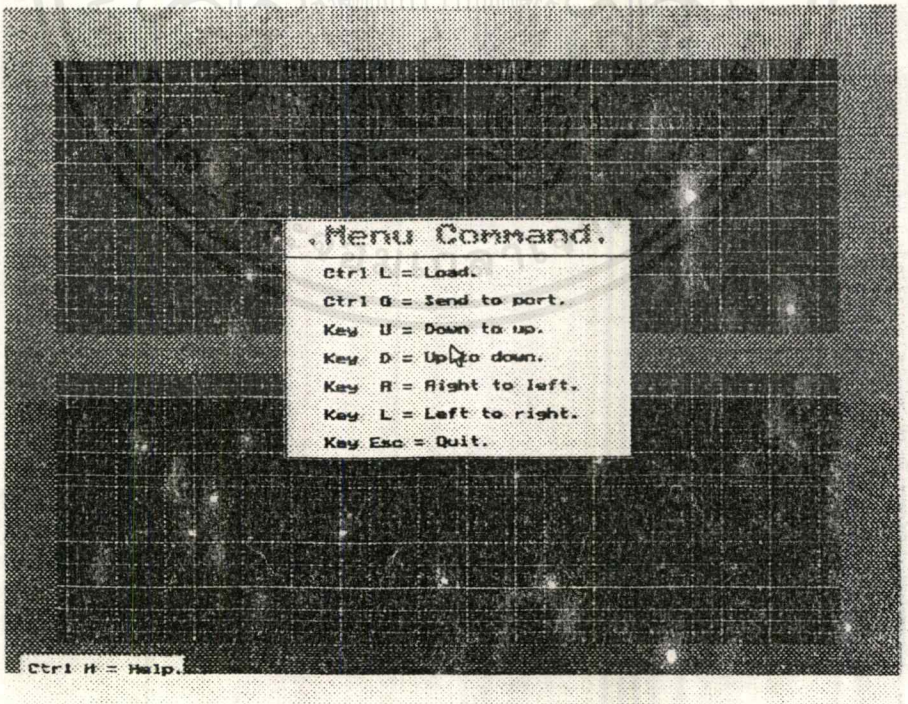
Ctrl G เมื่อผู้ใช้ต้องการส่งข้อมูลรูปภาพที่เรียกขึ้นมาแล้ว ซึ่งจะถูกทำเป็นหลาย ๆ เฟรม หลังจากกดคีย์ U หรือ D หรือ R หรือ L แล้ว เพื่อไปแสดงอย่างต่อเนื่องเป็นภาพเคลื่อนที่ในทิศทางต่าง ๆ ดังกล่าว ซึ่งผู้ใช้สามารถกำหนดจำนวนเฟรมรูปภาพที่จะส่งไปแสดงบนบอร์ดได้ด้วย โดยเมื่อกดคีย์จะได้ผลลัพธ์ดังแสดงในรูปที่ จ-19 เมื่อพิมพ์จำนวนเฟรมรูปภาพที่จะส่งไปเสร็จแล้ว ก็กด Enter ซึ่งจะมีข้อความแสดงว่าข้อมูลกำลังส่งไป หรือ 'Data is now sending'

Esc กดคีย์นี้เมื่อต้องการออกจากโหมดการทำงานนี้ ซึ่งจะเป็นการกลับเข้าสู่โหมดการทำงานเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

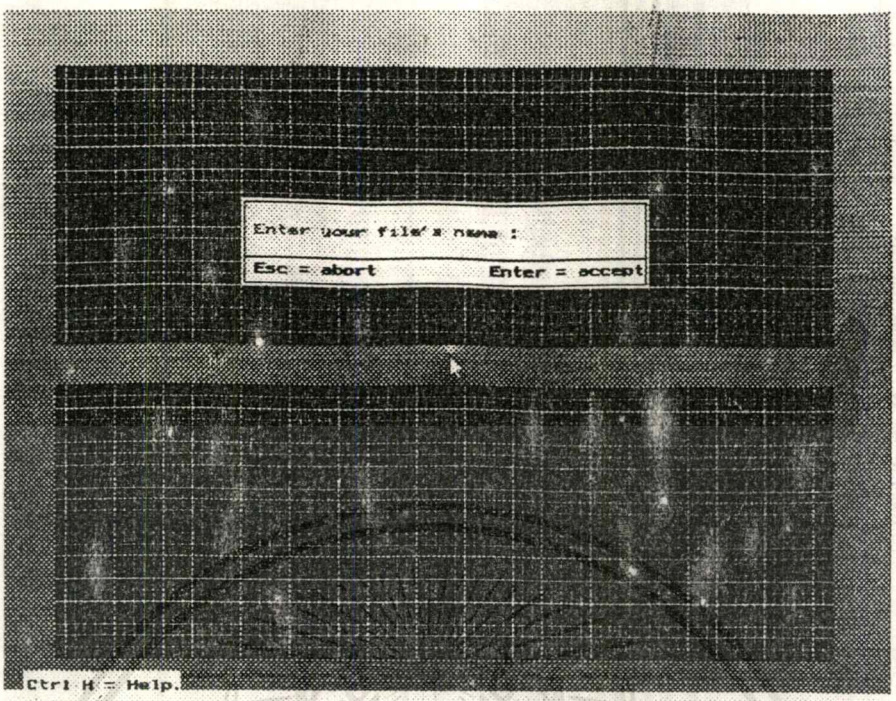


รูปที่ จ-16 แสดงโหมดการแสดงข้อมูลใน 4 ทิศทาง จากการกดคีย์ Ctrl V ในโหมดการทำงานเริ่มต้น

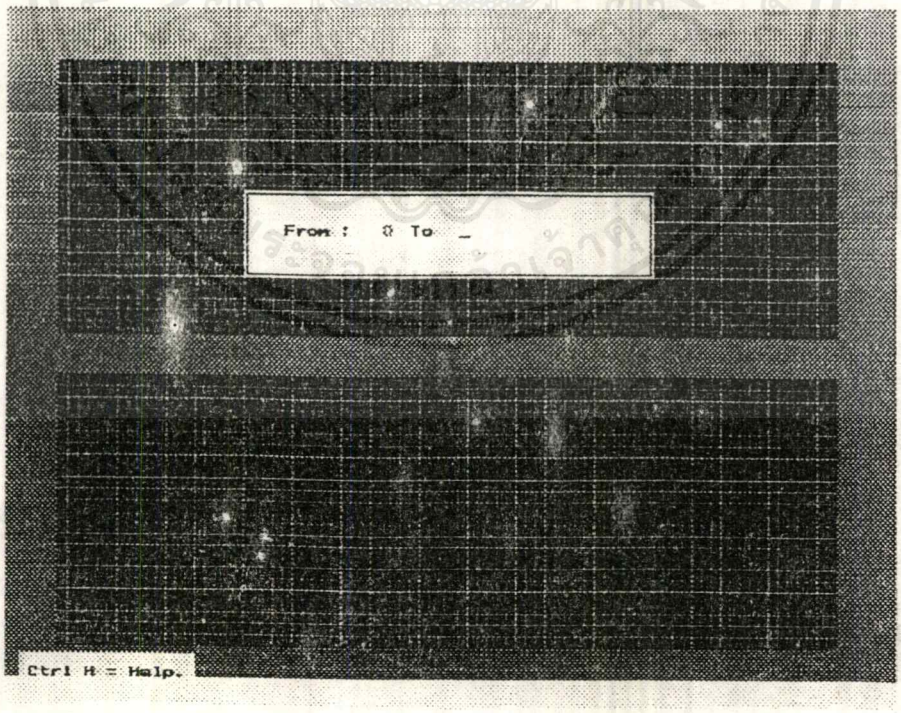


รูปที่ จ-17 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการแสดงข้อมูลใน 4 ทิศทาง เมื่อกดคีย์ Ctrl H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-18 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการแสดงข้อมูลใน 4 ทิศทาง เมื่อกดคีย์ Ctrl L



รูปที่ จ-19 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโหมดการแสดงข้อมูลใน 4 ทิศทาง เมื่อกดคีย์ U หรือ คีย์ D หรือ คีย์ R หรือ คีย์ L หรือ คีย์ Ctrl G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งานโปรแกรม EDIT_FON.EXE

เช่นเดียวกับการใช้งานโปรแกรม PROJECT.EXE จะต้องทำการติดตั้งเมาส์ไดรเวอร์ก่อนการใช้งาน ส่วนวิธีการใช้งานต่าง ๆ จะมีรายละเอียดดังต่อไปนี้

เมื่อเข้าสู่ DOS PROMPT พิมพ์คำว่า 'EDIT_FON' เพื่อเรียกแฟ้มข้อมูล EDIT_FON.EXE

A:\>EDIT_FON

จะเป็นการเข้าสู่การทำงานของโปรแกรมที่ผู้ใช้สามารถออกแบบรูปแบบบิตแม็ปของตัวอักษร ดังแสดงดังรูป จ-20 ซึ่งจะประกอบด้วยตารางในการออกแบบ และหน้าที่ของคีย์ต่าง ๆ ที่เป็นฟังก์ชั่นในการใช้งาน ซึ่งจะมีรายละเอียดดังนี้

C การกดคีย์นี้เป็นการลบข้อมูลทั้งหมดที่แสดงอยู่บนตารางออกแบบในขณะนั้น

I การกดคีย์นี้เป็นการกำหนดสถานะให้กับเมาส์ว่า การคลิกเมาส์แต่ละครั้งแทนการทำให้เกิดสีบนเซลล์ในตารางออกแบบ

D การกดคีย์นี้เป็นการกำหนดสถานะให้กับเมาส์ว่า การคลิกเมาส์แต่ละครั้งเป็นการลบสีบนเซลล์ในตารางออกแบบ

S เมื่อผู้ใช้ต้องการเก็บรูปแบบบิตแม็ปของตัวอักษรที่ได้ออกแบบไว้ ในรูปของแฟ้มข้อมูล โดยตั้งชื่อแฟ้มข้อมูลด้วยความยาวไม่เกิน 8 ตัวอักษร ซึ่งผลลัพธ์เมื่อกดคีย์นี้แสดงดังรูปที่ จ-21 และเมื่อพิมพ์ชื่อแฟ้มข้อมูลเสร็จแล้ว ก็กด Enter

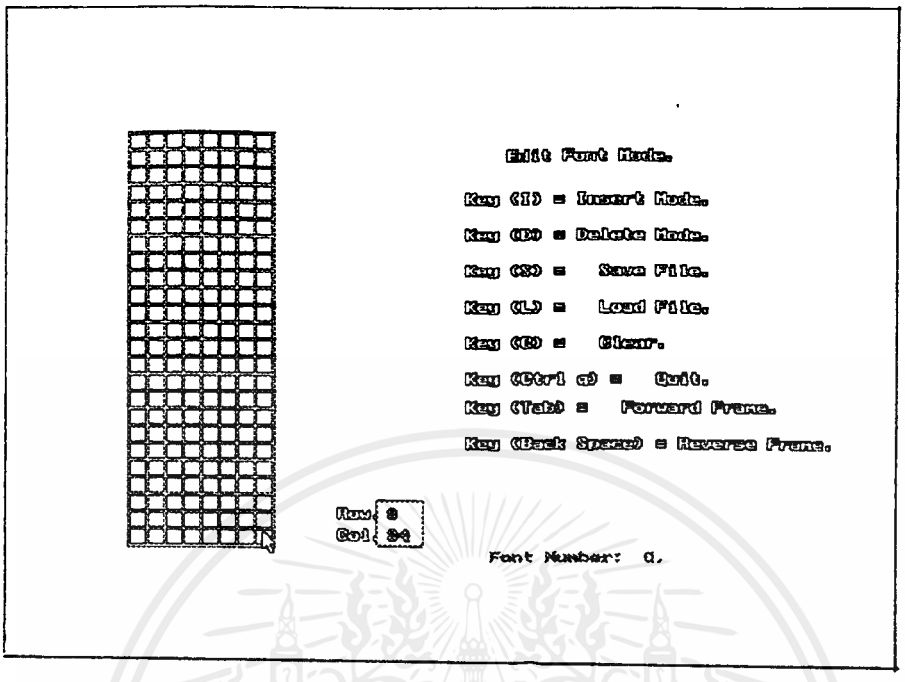
L เมื่อผู้ใช้ต้องการเรียกแฟ้มข้อมูลของรูปแบบบิตแม็ปของตัวอักษรที่ได้ออกแบบไว้หลังจากได้มีการเก็บในรูปของแฟ้มข้อมูลไปแล้ว โดยรูปที่ จ-21 จะแสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์เมื่อกดคีย์นี้ ซึ่งผู้ใช้จะต้องใส่ชื่อแฟ้มข้อมูลที่จะเรียกขึ้นมา

Tab กดคีย์นี้เมื่อผู้ใช้ออกแบบบิตแม็ปของตัวอักษรในแต่ละตำแหน่งของรหัสแอสกีเสร็จแล้ว และต้องการออกแบบรูปแบบตัวอักษรที่มีตำแหน่งของรหัสแอสกีถัดไป

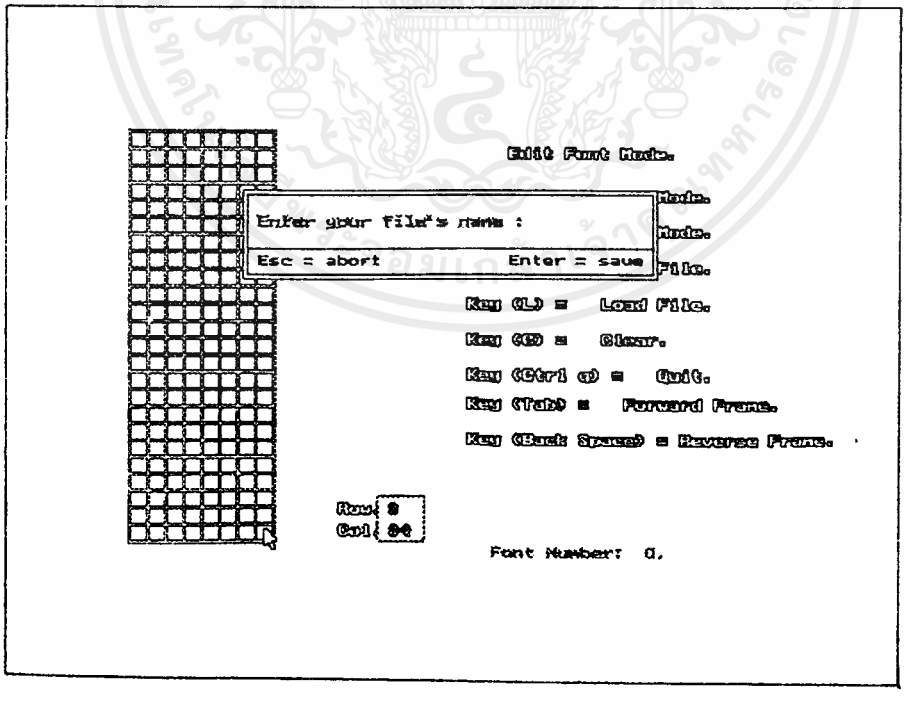
Backspace กดคีย์นี้เมื่อผู้ใช้ต้องการกลับไปแก้ไขรูปแบบตัวอักษรที่มีตำแหน่งของรหัสแอสกีก่อนหน้า

Ctrl Q กดคีย์นี้เมื่อผู้ใช้ต้องการออกจากโปรแกรมการทำงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ-20 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ หลังจากเรียกโปรแกรม EDIT_FON.EXE



รูปที่ จ-21 แสดงผลลัพธ์บนหน้าจอคอมพิวเตอร์ในโปรแกรม

EDIT_FON.EXE หลังจากกดคีย์ S หรือ L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้