



โปรแกรมช่วยออกแบบตัวอักษร
FONT GENERATOR PROGRAM

โดย

นายเมธา รุ่งพัฒนพันธ์
นายโสธร จ้ายเจริญ

วัน เดือน ปี... 1 ต.ค 2541
เลขทะเบียน... 038356
เลขเรียกหนังสือ... T.343.7.6 2541

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์ ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2539

038356

กรรมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรรมการ

หัวข้อปริญญานิพนธ์ โปรแกรมช่วยออกแบบตัวอักษร

โดย

นายเมธา รุ่งพัฒนพันธ์ เลขประจำตัว 37013351

นายโสธร จ้ายเจริญ เลขประจำตัว 37013365

อาจารย์ที่ปรึกษา อาจารย์ไพศาล สิทธิโยภาสกุล

ระดับการศึกษา อุตสาหกรรมศาสตรบัณฑิต

ภาควิชา เทคนิคอุตสาหกรรม

ปีการศึกษา 2539

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบและเขียนโปรแกรม เพื่อนำมาช่วยออกแบบและสร้างตัวอักษรในลักษณะต่างๆ หรือเป็นฟอนต์เจนเนอเรเตอร์ (ชนิดบิตแมปฟอนต์) โดยผู้ใช้งานสามารถเลือกหรือกำหนดขนาดของฟอนต์ได้ตามต้องการ และสามารถแก้ไขขนาดและรายละเอียดโดยโปรแกรมออกแบบการใช้งานเป็นแบบ Pop-Up และ Pull-Down Menu ให้มีความสะดวกในการใช้งาน

โปรแกรมนี้ทำงานบนจอภาพ VGA สามารถเก็บข้อมูลที่ออกแบบลงบน Memory และนำออกมาแก้ไขใหม่ได้, สามารถแก้ไขฟอนต์ของ WORD จูท , และสามารถพิมพ์ฟอนต์ที่ออกแบบออกทางเครื่องพิมพ์ โปรแกรมนี้พัฒนาด้วยโปรแกรมภาษา C โดยใช้คอมไพเลอร์ของ Borland C

PROJECT FONT GENERATOR PROGRAM

BY MR. METHA RUNGPATTANAPHAN

MR. SOTHORN JAYCHAREON

ADVISOR MR. PHAISAL SITIYOPASAKUL

LEVEL OF STUDY BACHELOR'S DEGREE IN INDUSTRIAL TECHNOLOGY

ACADEMIC YEAR 1996

ABSTRACT

This Project present the way to create program used to design font or it font generator (bitmap font). User can select font size,operate program in pop-up and pull-down menu,it has many menu to support design font

Program run on VGA display after design font can save data to memory and restore for design font again,can design word CU font,can print font to printer. The program development by c language (Referance to borland turbo C version 2).

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความสำเร็จด้วยความเรียบร้อย ด้วยการให้ความสนับสนุนจากผู้มีพระคุณดังต่อไปนี้

1. อาจารย์ไพศาล สิทธิโยภาสกุล
2. คุณชาमित นิลอร่าม
3. เพื่อนๆ พี่ๆ ทุกคนในแผนก INFORMATION SYSTEM (DIGITAL EQUIPMENT CORPORATION THAILAND) ที่ช่วยประสิทธิ์ประสาทให้ความรู้ในระบบคอมพิวเตอร์ในสมัยที่ยังทำงานจบจนปัจจุบัน

ทางผู้จัดทำขอขอบคุณมา ณ. ที่นี้ และขอขอบคุณบิดา มารดา และเพื่อนๆ ที่ช่วยเป็นกำลังใจและให้ความช่วยเหลือในการจัดทำปริญญานิพนธ์ฉบับนี้

นายเมธา รุ่งพัฒน์พันธ์
นายโสธร จ้ายเจริญ

สารบัญ	หน้า
บทคัดย่อ	ก
ABSTRACT	ข
กิตติกรรมประกาศ	ค
บทที่ 1 บทนำ	1
1.1 แนวความคิดในการทำปริญญานิพนธ์	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขอบเขตของโครงการ	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 เทคนิคการแสดงกราฟิกบนจอภาพ	3
2.2 อุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์กราฟิก	5
2.3 VGA ฮาร์ดแวร์	5
2.4 โครงสร้างของการ์ด VGA	6
2.5 การโปรแกรมใช้การ์ด VGA	18
2.6 หน่วยความจำแสดงผลในแต่ละโหมด	29
บทที่ 3 การออกแบบฟอนต์	33
3.1 ความรู้ทั่วไปเกี่ยวกับฟอนต์	33
3.2 การออกแบบฟอนต์แบบบิตแมป	38
3.3 การเก็บฟอนต์โดยใช้รหัสแอสกี	43
บทที่ 4 ผังงานของการทำงานในส่วนต่างๆ	46
บทที่ 5 วิธีการใช้งานโปรแกรม	56
5.1 ทรัพยากรและส่วนประกอบที่จำเป็นในการใช้งานโปรแกรม	56
5.2 แนะนำวิธีการใช้งานโปรแกรม	56
5.3 รายละเอียดขององค์ประกอบจอภาพของ โปรแกรม	58
บทที่ 6 บทวิจารณ์ สรุปและแนวทางในการพัฒนา	65
6.1 บทสรุป	65
6.2 แนวทางในการพัฒนา	65
หนังสืออ้างอิง	67

บทที่ 1

บทนำ

1. กล่าวนำ

ในปัจจุบันการนำเครื่องคอมพิวเตอร์มาใช้งานนั้นได้รับความนิยมเพิ่มขึ้นอย่างต่อเนื่อง ทั้งเพื่อใช้สำหรับอำนวยความสะดวกในด้านธุรกิจการงานต่างๆ ด้านการศึกษาและการวิจัย ฯลฯ ดังนั้นจะเห็นว่าเครื่องคอมพิวเตอร์จะเป็นสิ่งที่มีความจำเป็นอันจะเป็นปัจจัยที่มีความสำคัญอย่างมากทั้งในปัจจุบันและอนาคต ในปัจจุบันจึงทำให้ระบบคอมพิวเตอร์ ได้พัฒนาองค์ประกอบต่างๆ และได้มีการปรับปรุงทั้งด้าน SOFTWARE และ HARDWARE โดยเฉพาะประเทศที่พัฒนาแล้ว มักจะมีหน่วยงานวิจัยและพัฒนา ระบบคอมพิวเตอร์ทั้งสองด้านดังกล่าวควบคู่กันไป แต่สำหรับในโครงการนี้เป็นการศึกษาในทางซอฟต์แวร์หรือพัฒนาโปรแกรมจึงได้ออกแบบโปรแกรมเพื่อให้สามารถก้าวตามฮาร์ดแวร์ได้ทัน และมีความสามารถในการใช้งานมากที่สุด จอภาพและการ์ดสำหรับใช้ควบคุมการแสดงผลบนจอภาพในปัจจุบันมีมากมายหลายประเภทโปรแกรมในโครงการนี้ออกแบบให้ใช้งานกับ จอภาพแบบ VGA ซึ่งเป็นจอภาพที่นิยมกันมากในปัจจุบัน

1.1 แนวความคิดในการทำปริญญานิพนธ์

เนื่องจากการออกแบบตัวอักษรหรือตัวอักขระต่างๆในอดีตใช้การออกแบบบนกระดาษกราฟ ทำให้ไม่ สะดวกเกิดความยุ่งยากและล่าช้า ต่อมาเมื่อมีการนำเครื่องคอมพิวเตอร์มาใช้งาน ได้มีหน่วยงานเอกชนต่างประเทศได้พัฒนาโปรแกรมที่สามารถสร้างหรือออกแบบตัวอักษร ซึ่งผู้จัดทำเห็นว่า โปรแกรมดังกล่าวนี้ มีข้อจำกัดและราคาแพงจึงได้จัดทำโครงการนี้ขึ้น เพื่อเป็นแนวทางและใช้โปรแกรมนี้เป็นเครื่องมือในการออกแบบตัวอักษร โดยโครงการนี้ออกแบบเพื่อการพัฒนาตัวอักษรแบบบิตแมปโดยเฉพาะ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาลักษณะโครงสร้างของภาษา C ทั้งใน TEXT MODE และ GRAPHICS MODE
2. เพื่อศึกษาการทำงาน และความสามารถของการ์ดแสดงผล VGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เพื่อให้สามารถเขียนโปรแกรมใช้ในการออกแบบ และเก็บตัวอักษรและให้ตัวโปรแกรมมี MBNU ช่วยอำนวยความสะดวกในการออกแบบตัวอักษรให้มากที่สุด
4. เพื่อเป็นต้นแบบในการพัฒนาโปรแกรมที่ใช้ในการออกแบบตัวอักษรต่อไปในอนาคต

1.3 ขอบเขตของโครงการ

1. ใช้เครื่องคอมพิวเตอร์ส่วนบุคคลในการพัฒนาและใช้งาน
2. ในส่วนของการศึกษาการใช้งานของการ์ดแสดงผล VGA นั้นจะเป็นการศึกษาความสามารถของการ์ดแสดงผล VGA เป็นส่วนใหญ่ และรวมทั้งการให้สนับสนุนจอภาพแบบอื่น
3. ตัวโปรแกรมเขียนขึ้นจากภาษาซีโดยใช้ C FUNCTION STANDARD เพื่อให้ง่ายในการนำไปพัฒนาบนภาษาโครงสร้างอื่นๆ
4. โปรแกรมที่ใช้งานได้นี้เป็นแบบ POP-UP และ PULL-DOWN MENU เพื่ออำนวยความสะดวกในการใช้งาน โดยตัวโปรแกรมนี้ทำงานบน DOS
5. การออกแบบหรือสร้างฟอนต์ใหม่ผู้ใช้สามารถกำหนดขนาดของฟอนต์ที่ต้องการได้
6. สามารถแก้ไขฟอนต์ของ จูฬาได้

บทที่ 2

ทฤษฎีและหลักการ

2.1 การแสดงกราฟิกบนจอ

คอมพิวเตอร์กราฟิกหมายถึงการสร้างและการจัดการกับภาพกราฟิกโดยใช้คอมพิวเตอร์ คำจำกัดความสั้นๆ นี้ไม่ได้อธิบายถึงเหตุผลของการพัฒนาคอมพิวเตอร์กราฟิกซึ่งเริ่มต้นมาจากการเป็นเทคนิคอย่างหนึ่ง ในการแสดงข้อมูลตัวเลขจำนวนมากๆ ให้อยู่ในรูปที่ชัดเจนกว่าเดิมและทำความเข้าใจได้ง่ายกว่าเดิม เช่นข้อมูลอาจแสดงได้ในรูปของ เส้นกราฟ แผนภาพ แผนภูมิ แทนที่จะเป็นตารางของตัวเลข เป็นต้น ก่อนหน้าที่จะเข้าใจการแสดงกราฟิกบนจอ เราควรเข้าใจ เพื่อให้ทำความเข้าใจเรื่องกราฟิกที่แสดงบนจอให้ดียิ่งขึ้นจะขอกล่าวอย่างย่อ ๆ ถึงวิธีการนำเสนอแสดงผลกราฟิกบนหน้าจอ เทคนิคการแสดงผลกราฟิกอาจแบ่งได้เป็น 2 วิธีใหญ่ ๆ คือ

1. Bit Mapped
2. Vector

วิธีการของบิตแมปก็คือทุกจุดบนหน้าจอ ซึ่งเรียกกันว่าพิกเซลนั้นมีความสัมพันธ์โดยตรงกับหน่วยความจำแบบหนึ่งต่อหนึ่ง แต่ก็มีได้หมายความว่า 1 พิกเซลจะเป็น 1 บิต ในหน่วยความจำเสมอไปอาจจะเป็น 4 บิต หรือ 8 บิต แล้วแต่จะทำให้ 1 พิกเซลสามารถแสดงสีหรือระดับความเข้มขาวดำมากน้อยแค่ไหน ข้อสำคัญก็คือจากตำแหน่งของพิกเซลบนหน้าจอเราสามารถมองหาค่าของพิกเซลในหน่วยความจำได้โดยตรง หรือการเปลี่ยนแปลงหน่วยความจำมีผลโดยตรงต่อการแสดงผลบนหน้าจอเป็นที่เรียกกันติดปากว่าหน่วยความจำที่ใช้ในการแสดงผลคือ วิดีโอแรม

ดังนั้นเทคนิคการแสดงผลกราฟิกโดยวิธีบิตแมป ก็คือการแก้ไขเปลี่ยนแปลงหน่วยความจำของการแสดงผลโดยตรงข้อมูลภาพกราฟิกก็คือภาพข้อมูลที่น่าไปเขียนบนหน่วยความจำแสดงผล วิธีการแสดงผลกราฟิกแบบเวกเตอร์นั้นแตกต่างไปจากวิธีการของบิตแมป คือถ้าข้อมูลของภาพไม่ได้อยู่ในรูปที่จะนำไปแสดงบนจอโดยตรงแต่จะอยู่ในรูปของคำสั่งในการลากเส้น เขียนวงกลม เขียนรูปหลายเหลี่ยม และการระบายสีแทน ภาพที่ได้จากการสร้างโดยวิธีของเวกเตอร์จึงเรียกว่า Object - Based Image คือบ่งบอกลักษณะของภาพมากกว่าที่จะเป็นตัวภาพเอง

จอภาพของการแสดงผลที่ใช้ในไมโครคอมพิวเตอร์ในปัจจุบันจะเป็น Raster Scan คือแสดงในลักษณะกวาดแสดงไปบนจอติดต่อกันจากซ้ายไปขวา โดยใช้หลักการเดียวกันกับทีวี

จำนวนเส้นที่กวาดแสดงผลนั้นเป็นตัวกำหนดขีดความสามารถของวิดีโอการ์ด ซึ่งจะมีตั้งแต่ 200,000 เส้นขึ้นไป ไม่่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

350, 480 จนถึง 800 เส้น แต่เดิมนั้นจอภาพที่แสดงสี CGA สามารถรับสัญญาณภาพให้แสดงผลได้ 200 เส้นใน 1 ภาพหรือที่เรียกว่า Frame Monochrome หรือ Hercules Graphic Adapter ก็ 350 เส้น จะเห็นว่าจำนวนเส้นของการแสดงจะใกล้เคียงกับระบบทีวีที่มีอยู่มาก คือระบบทีวีที่มีอยู่ในบ้านเราใช้ 625 เส้นใน 2 Frame ทุกจุดที่ปรากฏบนจอจะต้องมีหน่วยความจำเก็บค่าของพิกเซลเอาไว้ นั่นก็คือ การมีหน่วยความจำสำหรับเก็บภาพแบบบิตแมปนั่นเอง กราฟิกโดยวิธีการบิตแมปสามารถแสดงบนหน้าจอได้ทันที ส่วนกราฟิกแบบเวกเตอร์นั้นจะต้องให้ซีพียูแปลงออกมาเป็นบิตแมปในหน่วยความจำของการแสดงผลเสียก่อน จึงจะเห็นภาพปรากฏบนหน้าจอได้

ถ้าเปรียบเทียบให้เห็นชัดถึงความแตกต่างระหว่างบิตแมปและเวกเตอร์ ก็คงจะพอประมาณได้กับงานศิลปะ 2 วิธีคือ การวาดภาพจิตรกรรม และงานปฏิมากรรม อย่างแรกเป็นการสร้างภาพขึ้นมาเปรียบได้กับกราฟิกชนิดบิตแมป อย่างที่สองเป็นการสร้างงานศิลปะขึ้นมาสามารถที่จะมองในแง่มุมที่แตกต่างกันได้ ซึ่งเปรียบได้กับกราฟิกแบบเวกเตอร์

งานกราฟิกที่ใช้ทางด้านสถาปนิก และวิศวกรรมจะต้องเป็นชนิดเวกเตอร์ก็จะต้องมีทั้ง 3 มิติและสามารถใช้ขีดความสามารถของซีพียูแปลงเวกเตอร์ของกราฟิกให้ปรากฏออกมาบนจอได้ในแง่มุมที่แตกต่างกัน โปรแกรม Auto CAD เป็นตัวอย่างหนึ่งของการสร้างภาพแบบเวกเตอร์ ซึ่งสามารถกำหนดแง่มุมของการมองได้ไม่เพียงเท่านั้นแต่กราฟิกในรูปแบบเวกเตอร์สามารถที่จะนำมาแสดงในลักษณะย่อหรือขยาย ในอัตราส่วนเท่าใดก็ได้ โดยที่ภาพที่แสดงออกมายังคงรูปร่างที่เหมือนกับความเป็นจริง

เพื่อให้ผู้อ่านได้เห็นภาพชัดเจน ระหว่างบิตแมปและเวกเตอร์ลองมาดูการแสดงตัวหนังสือบนจอกราฟิก โดยมากการแสดงอักขระบนหน้าจอจะเป็นแบบบิตแมป อักษรแต่ละตัวถูกกำหนดให้มีหน้าตาต่างกัน และเข้าไปครอบครองเนื้อที่ในหน่วยความจำเท่าๆกัน การนำอักขระเหล่านี้มาแสดงจึงไม่น่าจะคิดที่จะเรียกว่า การพิมพ์อักขระบนจอ ถ้าเราจะขยายการพิมพ์ให้ใหญ่ขึ้นก็จะต้องขยายให้เป็นอัตราส่วนที่เป็นเลขจำนวนเต็มเท่านั้น ภาพของอักษรที่ปรากฏบนหน้าจอไม่ได้มีความละเอียดเพิ่มขึ้น แต่กลับจะลดถอยลงไปมองเห็นรอยยัก ๆ บนตัวอักษรได้ ในทางตรงกันข้ามการแสดงอักขระบนจอแบบเวกเตอร์นั้น เป็นการบอกวิธีการเขียนตัวอักษร บนจอ เช่น จากจุดเริ่มต้นลากเส้นขึ้นข้างบน 4 หน่วย ลากไปทางขวา 3 หน่วย ลากลง 4 หน่วย ดังกล่าวเป็นต้น ลักษณะเช่นนี้จึงไม่คิดที่จะเรียกว่าเป็นการสอนคอมพิวเตอร์ให้รู้จักเขียนหนังสือ ข้อดีของการเขียนแบบเวกเตอร์ก็คือ สามารถขยายโดยอัตราส่วนเท่าใดก็ได้โดยไม่เสียความละเอียดของแง่มุม การใช้งานของวิธีการสร้างภาพกราฟิกบนจออาจจะนำมาใช้ผสมกันทั้งสองแบบ คือ บิตแมป และเวกเตอร์ แต่เวลาเก็บผลงาน คือ ภาพที่ได้จะเอาไปเก็บแบบบิตแมป วิธีการเช่นนี้ใช้

เอก ในโปรแกรมสำเร็จรูปในการสร้างกราฟิกเพื่อการนำเสนออย่างเช่น PC Paint ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 อุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์กราฟิก

1. ตัวไมโครคอมพิวเตอร์ ซึ่งบางทีก็ใช้คำว่า Graphic Engine
2. อุปกรณ์แสดงผลภาพ อันได้แก่ การ์ดแสดงผลและตัวจอด้วย
3. อุปกรณ์พิมพ์รูปกราฟิกบนกระดาษ เช่น เครื่องพิมพ์, เครื่องวาด (Plotter)
4. อุปกรณ์นำภาพจากภายนอกเข้าไปยังคอมพิวเตอร์ เช่น เครื่อง Scanner
5. อุปกรณ์ช่วยในการเปลี่ยนแปลงแก้ไขภาพ

2.3 VGA ฮาร์ดแวร์

เทคโนโลยีการแสดงผลบนเครื่องคอมพิวเตอร์ส่วนบุคคลนับว่ามีการพัฒนาขึ้นมามากทีเดียวจากเดิมซึ่งเป็นการแสดงผลบนจอโมโนโครม ที่ใช้การ์ดโมโนโครมอะแดปเตอร์หรือการ์ดเซอร์คิวลิสแล้วพัฒนาต่อมาเป็นการการแสดงผลบนจอสีที่เรียกว่า Color Graphic Adapter (CGA) และ Enhanced Graphic Adapter (EGA) ซึ่งล้วนแล้วแต่เป็นระบบแสดงผลที่เป็นสัญญาณดิจิทัล

นั่นคือสัญญาณที่ออกจากการ์ดแสดงผลที่จะไปควบคุมจอภาพนั้นเป็นสัญญาณดิจิทัลในระบบดิจิทัลนี้การ์ด EGA จะมีความละเอียดสูงสุดคือจำนวนจุด (Pixel) เท่ากับ 640x350 จุด และสามารถแสดงสีได้ 16 สีพร้อมกันจากจำนวนสีทั้งหมด 64 สีต่อมาไอบีเอ็มได้พัฒนาระบบแสดงผลแบบใหม่ขึ้นมาอีกเรียกว่า Video Graphic Array (VGA) ซึ่งใช้สัญญาณอะนาลอกควบคุมจอแสดงผล ทำให้แสดงจำนวนสีได้มากขึ้นเพราะว่าคอมพิวเตอร์ของสัญญาณอะนาลอกมีมากกว่าสัญญาณดิจิทัล ระบบ VGA จึงสามารถแสดงภาพสีได้ถึง 256 สีพร้อมกันจากจำนวนสีที่เป็นไปได้ทั้งหมด 256x1024 สี และยังมีความละเอียดสูงขึ้นถึง 640x480 จุด ทำให้สามารถแสดงภาพเหมือนจริงได้

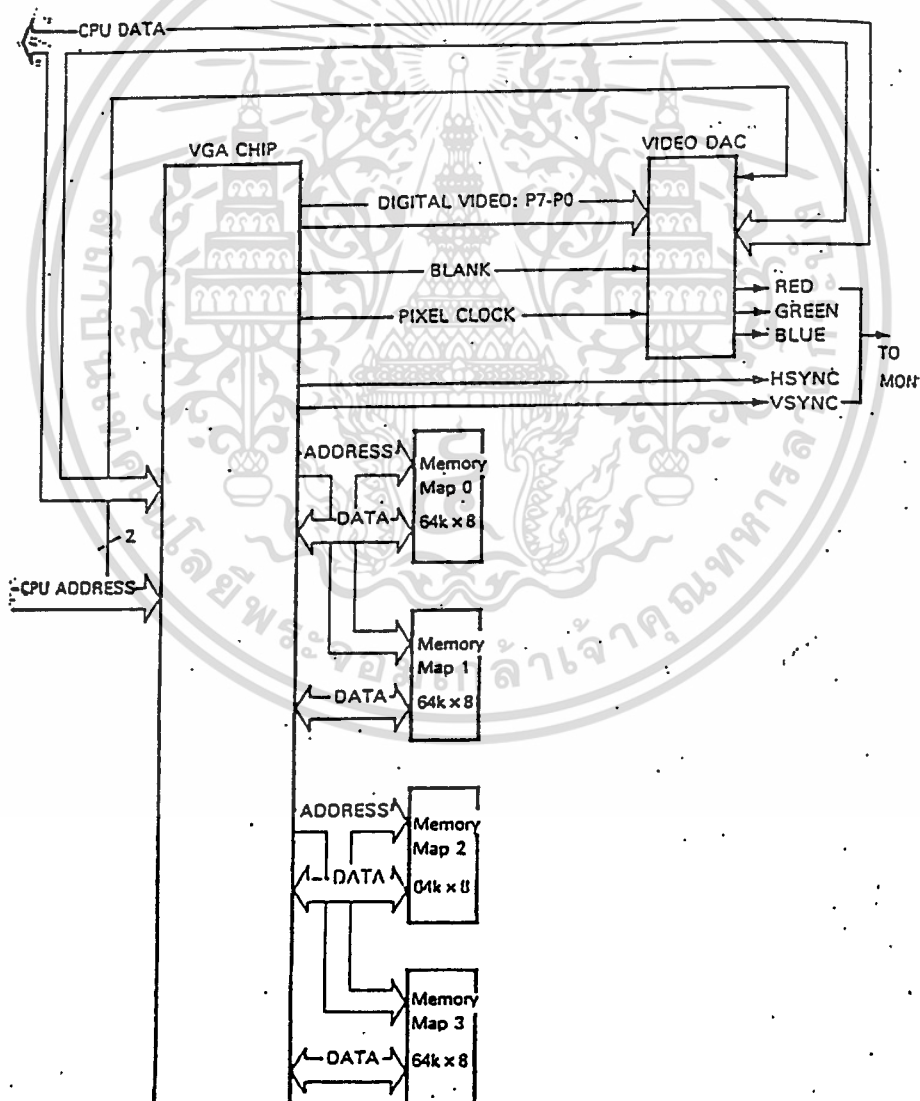
ซึ่งมีประโยชน์อย่างมากในการใช้คอมพิวเตอร์กับงานด้านต่าง ๆ อาทิ เช่น งานออกแบบด้านอุตสาหกรรม (CAD/CAM) งานทางด้านการแพทย์งานทางด้านการประมวลผลทางรูปภาพ ซึ่งสามารถประยุกต์ใช้กับงานต่าง ๆ ได้อย่างมากมาย เนื่องจากข้อดีของระบบแสดงผลแบบ VGA นี้เอง ไอบีเอ็มจึงได้รวมการ์ดแสดงผลระบบ VGA ลงบนเมนบอร์ดของเครื่อง IBM PS/2 เช่น PS/2 model 50,60,70 จึงถือได้ว่าเป็นมาตรฐานการแสดงผลของเครื่อง IBM PS/2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โครงสร้างของการ์ด VGA

ระบบแสดงผลแบบ VGA หรือที่เราเรียกกันดีว่าการ์ด VGA นั้นประกอบด้วยส่วนสำคัญอยู่ 3 ประการด้วยกันคือ

1. หน่วยความจำของส่วนแสดงผล (Video Memory)
2. ชิพ VGA
3. เปลี่ยนสัญญาณดิจิทัลให้เป็นสัญญาณอะนาลอก (Video DAC)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.3 ส่วนประกอบหลักของการ์ด VGA ตัดให้หน้าไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 หน่วยความจำแสดงผล

หน่วยความจำของการ์ด VGA ตามมาตรฐานของไอบีเอ็มจะมีจำนวน 256 กิโลไบต์ เป็นไดนามิกแรมที่ถูกแบ่งออกเป็น 4 plane ข้อมูลที่ถูกใช้แสดงผลจะถูกเก็บอยู่ในหน่วยความจำนี้ ซึ่งรูปแบบของการเก็บจะแตกต่างกันไปขึ้นกับโหมดของการแสดงผล ตัวอย่าง เช่นการแสดงผลในเท็กซ์โหมด (Text Mode) เฟรม 0 ซึ่งเป็นหน่วยความจำแอดเดรสคู่ ซึ่งจะเก็บข้อมูลที่เป็นรหัสแอสกีหน่วยความจำแอดเดรสคี่ ซึ่งอยู่ในเฟรม 1 จะเก็บค่าแอดทริบิวต์ ส่วนหน่วยความจำเฟรม 2 จะสำรองไว้ใช้สำหรับเก็บค่าแคแรคเตอร์เรนเนอเรเตอร์ หน่วยความจำเฟรม 3 จะไม่ถูกใช้ ส่วนในโหมดกราฟิกก็จะแตกต่างกันไป ในปัจจุบันการ์ด VGA ที่ถูกผลิตขึ้นมักมีความละเอียดในการแสดงผลสูงกว่ามาตรฐานของไอบีเอ็มและต้องใช้หน่วยความจำเพิ่มสูงขึ้นด้วย

หน่วยความจำในเท็กซ์โหมด

การแสดงผลในเท็กซ์โหมด (Text Mode) นั้นจะมีความซับซ้อนน้อยกว่าการแสดงผลในโหมดกราฟิกมาก เพราะว่าเป็นการจัดการกับรหัสแอสกี มิใช่จัดการกับจุดใดจุดหนึ่งบนจอภาพ มาตรฐานของเท็กซ์โหมดแบ่งเป็น 80 คอลัมน์ต่อบรรทัดในกรณีที่เป็น 80 คอลัมน์ต่อบรรทัด ใน 1 จอภาพสามารถสามารถแสดงตัวอักษรได้ทั้งสิ้น 2,000 ตัวแต่การแสดงผลของตัวอักษรละ 1 ตัว ต้องใช้หน่วยความจำ 2 ไบต์

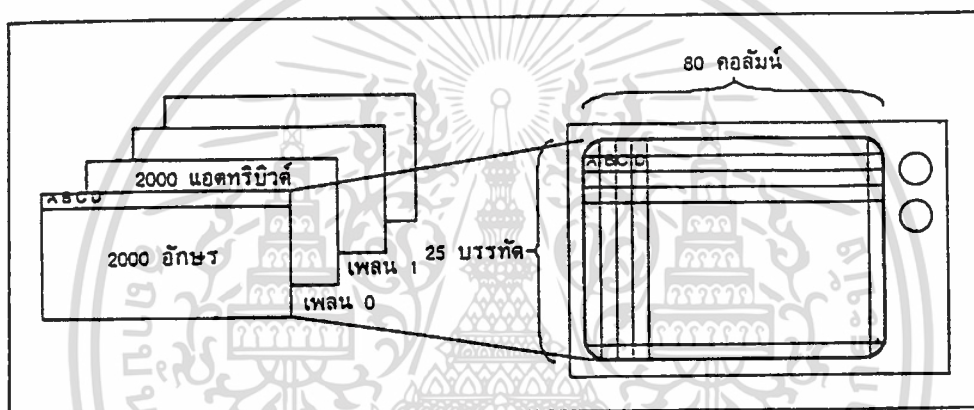
ดังนั้นในหนึ่งจอภาพจะต้องใช้หน่วยความจำทั้งสิ้น 4,000 ไบต์ แต่หน่วยความจำแสดงผลแบ่งออกเป็นเพจ ๆ ละ 4096 ไบต์ ซึ่งจะเหลือที่ว่าง 96 ไบต์ ที่ไม่ถูกใช้ ในการเปลี่ยนรหัสแอสกีไปเป็นจุดที่เรียงกันเป็นตัวอักษรบนจอภาพนั้นจะต้องใช้ตารางการแปลงที่เรียกว่าแคแรคเตอร์เรนเนอเรเตอร์ในการแสดงผลระบบเดิม เช่น การ์ดโมโนโครมตารางการเปลี่ยนแปลงนี้จะเก็บอยู่ในหน่วยความจำที่เป็นชนิด ROM (อ่านได้อย่างเดียว) จึงไม่สามารถแก้ไขรูปแบบของตัวอักษรได้โดยง่าย แต่ใน VGA หรือ EGA ตารางการแปลงนี้จะถูกโหลดลงใน เฟรมที่ 2 ของหน่วยความจำ (ซึ่งเป็นไดนามิกแรม) ทำให้สามารถแก้ไขรูปแบบตัวอักษรได้ง่าย ใน EGA จะมีตารางแปลงนี้ได้ถึง 4 ชุด ส่วนใน VGA มีได้ถึง 8 ชุด แต่ละชุดเก็บได้ถึง 256 ตัว รูปแบบการจัดหน่วยความจำในเท็กซ์โหมดแสดงดังรูปที่ 2.2 (ก)

มาตรฐานของตัวอักษรในตารางแปลงนี้ จะมีจำนวนจุด (pixel) ต่างกันไปขึ้นอยู่กับโหมดของการแสดงผล เช่น ในโหมด CGA (การ์ด VGA สามารถทำงานในโหมดที่มีความละเอียดต่ำกว่าได้) ตัวอักษรจะมีขนาด 8x8 (กว้างxยาว) จุดใน EGA จะเป็น 8x14 จุดและ 8x16 จุด สำหรับ VGA ค่าในตารางแปลงนี้จะถูกโหลดลงใหม่ทุกครั้งโดยไบออสบนการ์ด เมื่อมีการเปลี่ยนโหมดของการทำงาน

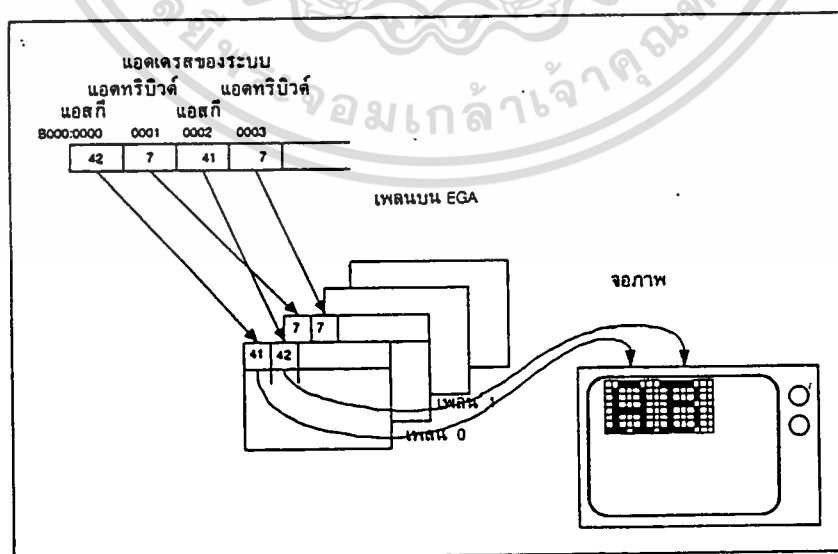
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่วาระณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำในกราฟิกโหมด

ในโหมดกราฟิกจุดๆ หนึ่งบนจอจะแทนด้วยข้อมูลที่มีจำนวนบิตต่าง ๆ กัน เช่นใน โหมด CGA 2 สี (mode 6) จะใช้ข้อมูลจำนวน 1 บิตในการแสดงจุดหนึ่งจุด (8 จุดต่อไบต์) ในที่นี้จะขอกล่าวถึงโหมด Enhanced Color Graphics (mode 10H) เท่านั้นในโหมด 10H นี้เป็น ที่นิยมสำหรับการใช้กับงานต่าง ๆ เพราะมีความละเอียดสูงถึง 640x350 จุด และแสดงสีได้ 16 สีพร้อมกันในการแทนจุดๆหนึ่งบนจอภาพจะใช้ข้อมูล 4 บิต โดยที่แต่ละบิตมาจากแต่ละเฟลน ของหน่วยความจำ

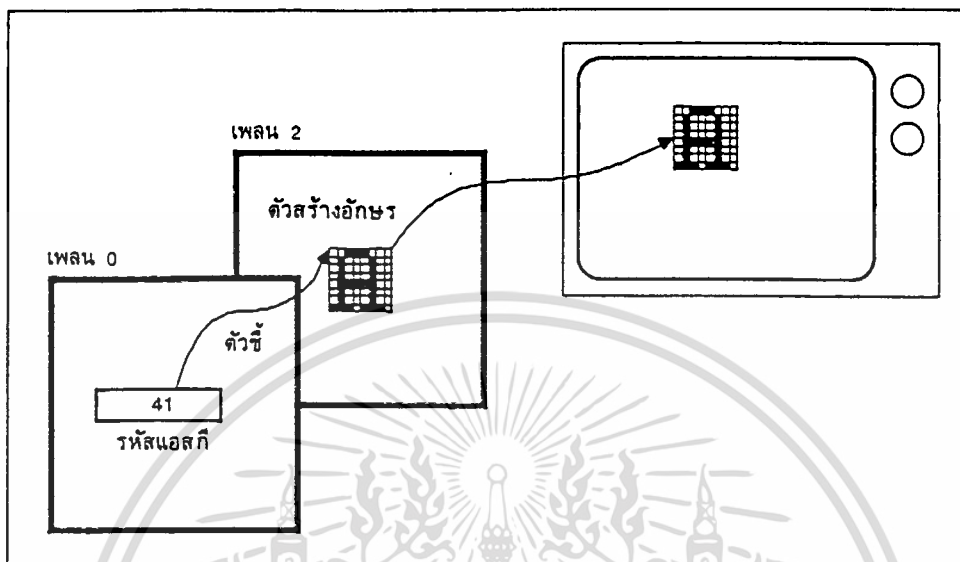


รูปที่ 2.2 (ก) รูปแบบการจัดหน่วยความจำในเท็กซ์โหมด

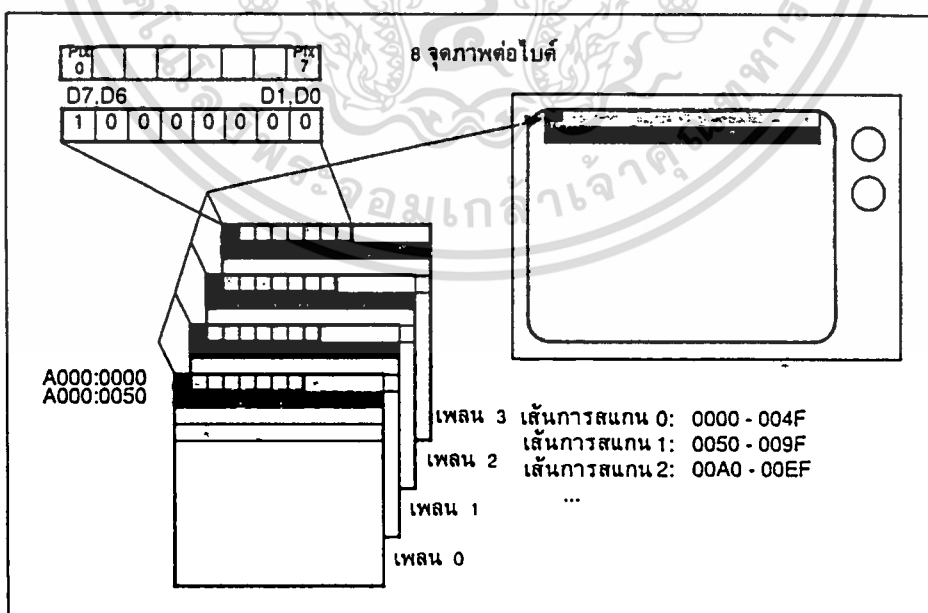


รูปที่ 2.2 (ข) ความสัมพันธ์ระหว่างแอดเดรสกับเฟลนของหน่วยความจำ

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 ขั้นตอนการทำให้เกิดตัวอักษรบนจอภาพ



รูปที่ 2.4 รูปแบบการเก็บข้อมูลในหน่วยความจำสำหรับโหมดกราฟิก

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับโครงการวิจัยในเชิงวิชาการเท่านั้น ไม่อนุญาตให้นำไปใช้โดยไม่ได้รับอนุญาตจากเจ้าของลิขสิทธิ์ หากมีผู้ใดฝ่าฝืนจะดำเนินการตามกฎหมายต่อไป

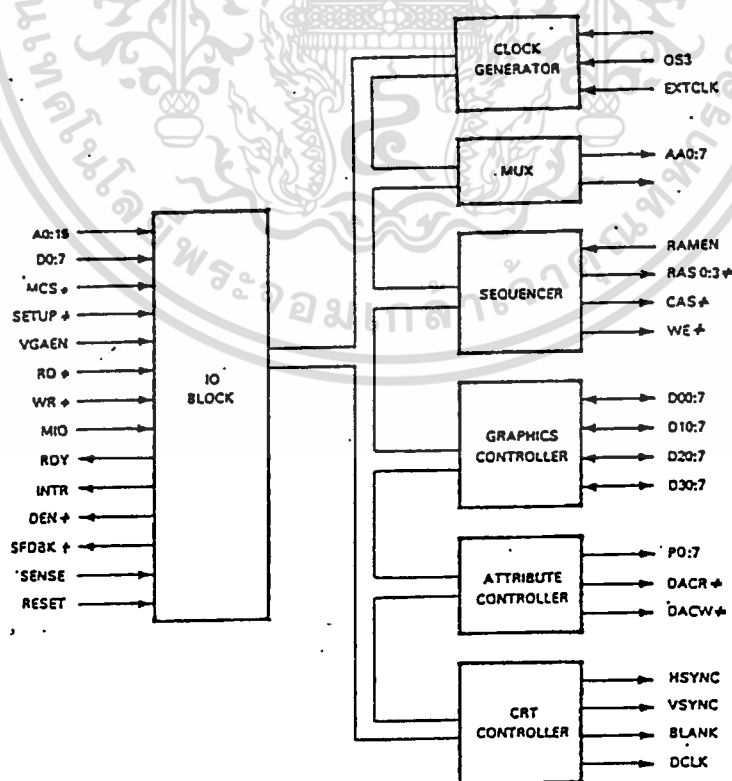
2.4.2 จีพ VGA

ระบบแสดงผลแบบ VGA มาตรฐานของไอบีเอ็มนั้นใช้ชิพ 82706 Video Graphic Array ซึ่งคอมแพคติเบิ้ลระดับไบออสกับระบบเดิมคือ EGA CGA และ MDA

82706 VGA controller เป็นตัวอินเทอร์เฟสระหว่างซีพียู ซึ่งในที่นี้อาจจะเป็น 8088, 80286 หรือ 80386 กับหน่วยความจำของการแสดงผล (video memory) และเป็นตัวส่งข้อมูลของภาพไปยัง video DAC (Digital to Analog Converter) ทำการแปลงสัญญาณดิจิทัลไปเป็นสัญญาณอะนาลอก เพื่อส่งให้จอแสดงผลต่อไป

การแอ็กเซสกับหน่วยความจำของการแสดงผลจะต้องผ่านตัว 82706 เสมอ ซึ่งมีข้อดีก็คือ ซีพียูสามารถแอ็กเซสกับหน่วยความจำขณะที่ทำการรีเฟรชหน่วยความจำ ซึ่งทำให้ซีพียูเขียน หรืออ่านจากหน่วยความจำได้โดยไม่จำเป็นต้องรอให้ถึงช่วงเวลาของการรีเฟรช (retrace) ของจอภาพ

82706 ใช้กับหน่วยความจำได้ 256 กิโลไบต์ โดยสามารถโปรแกรมแอดเดรสเริ่มต้นได้ 3 ตำแหน่ง ซึ่งโดยทั่วไปมักจะขึ้นกับโหมดการแสดงผลจาก รูปที่ 2.5 จะเห็นว่าภายในชิพ 82706 จะยังมีตัวควบคุมการทำงานหลักอยู่ 4 ตัว คือ



รูปที่ 2.5 บล็อกไดอะแกรมของจีพ VGA

1. CRT controller

อุปกรณ์ CRT controller ทำหน้าที่กำเนิดสัญญาณที่ใช้ควบคุมการทำงานของจอภาพ วิธีการเหมือนในเครื่อง TV เช่น สัญญาณซิงค์ตามแนวนอน สัญญาณซิงค์ตามแนวตั้ง สัญญาณแบล็กกิ้งและแอดเดรส สำหรับการรีเฟรชหน่วยความจำในระบบนี้ CRT controller นั้นจะใช้จำนวนรีจิสเตอร์ถึง 25 ตัว ซึ่งมีบางตัวที่คอมแพคติเบิ้ลกับ 6845 CRT controller ที่มีอยู่บนการ์ดแสดงผลแบบโมโนโครม รีจิสเตอร์ต่างๆ ใน CRT controller แสดงดังรูปที่ 2.6

2. Graphic Controller

เป็นตัวกลางเชื่อมทางเดินข้อมูลระหว่างหน่วยความจำของการแสดงผลกับตัวโปรเซสเซอร์หลักกับแอดดริวิตคอนโทรลเลอร์ ในสภาวะปกติข้อมูลที่ส่งมาจากตัวโปรเซสเซอร์จะสามารถถูกส่งไปยังหน่วยความจำโดยทราฟฟิกคอนโทรลเลอร์ได้โดยตรง สำหรับในกรณีอื่น กราฟฟิกคอนโทรลเลอร์ มีฟังก์ชันช่วยในการวาดรูปกราฟิก โดยนำข้อมูลที่ผ่านตัวมันมากระทำฟังก์ชันทางลอจิก ก่อนที่จะเขียนไปยังหน่วยความจำ ภายในกราฟิกคอนโทรลเลอร์มีรีจิสเตอร์อยู่ 9 รีจิสเตอร์ ดังแสดงในตารางรูปที่ 2.7 ซึ่งสามารถสรุปการทำงานได้ดังรูปที่ 2.8 จากรูปจะเห็นว่า มีข้อมูลที่เรียกว่า Latched Data อยู่ 4 ไบต์ (แต่ละไบต์มาจากแต่ละเฟรม) ข้อมูลเหล่านี้จะถูกเก็บอยู่ในที่ที่หนึ่ง ซึ่งจะเกิดขึ้นทุกครั้งที่มีการอ่านข้อมูลจากโปรเซสเซอร์ (Processor Data) จะผ่านการ ROTATE ซึ่งอาจจะเป็นการ ROTATE ตั้งแต่ 0 บิต (ไม่มีการ ROTATE) ถึง 7 บิต ทั้งนี้ขึ้นกับค่าในรีจิสเตอร์ Data Rotate (บิตที่ 0-2) จากนั้นจะถูกนำมากระทำฟังก์ชันไค้นขึ้นกับค่าในรีจิสเตอร์ Data Rotate (บิตที่ 3-4)

จากนั้นข้อมูลจะผ่านรีจิสเตอร์ 2 ตัวคือ SET/RESET และ SET/RESET ENABLE แต่ละรีจิสเตอร์จะใช้เพียง 4 บิต แต่ละบิตแทนแต่ละเฟรม จากรูปบิตที่ 2 และที่ 3 ในรีจิสเตอร์ SET/RESET ENABLE เป็น 0 ดังนั้น ข้อมูลเฟรมที่ 2 และ 3 จะผ่านไปโดยตรงส่วนบิตที่ 0 และ 1 มีค่าเป็น 1 ต้องพิจารณาในรีจิสเตอร์ SET/RESET เช่นบิตที่ 0 มีค่าเป็น 1 ดังนั้นเฟรมที่ 0 จึงมีค่าเป็น 1 ทั้ง 8 บิต จากนั้นข้อมูลจะผ่านรีจิสเตอร์ Bit Mask ถ้าค่าในรีจิสเตอร์เป็น 1 ก็จะไม่มีการเปลี่ยนแปลงค่าของข้อมูล แต่ถ้าค่าในรีจิสเตอร์เป็น 0 ข้อมูลในบิตนั้น ๆ จะถูกนำมาจาก Latched Data แทนที่จะเป็นข้อมูลเดิม รีจิสเตอร์ Bit Mask เป็นรีจิสเตอร์สุดท้ายของกราฟิกคอนโทรลเลอร์ที่ทำให้ข้อมูลมีการเปลี่ยนแปลง หลังจากนั้นแล้วข้อมูลยังมิได้ถูกเขียนลงไปยังหน่วยความจำโดยตรง จะต้องผ่านตัวควบคุมตัวหนึ่งที่เรียกว่า Sequencer ซึ่งจะได้อธิบายต่อไปตัวอย่างของการกระทำฟังก์ชันทางลอจิกของกราฟิกคอนโทรลเลอร์แสดงดังรูปที่ 2.9 จากรูปจะเห็นว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 อักษรเดิมคือ A และ B เราสามารถทำให้เกิดอักษรใหม่ขึ้นมาได้
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ต่อผู้อื่น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Name	R/W	Index	Read Port	Write Port
CRT Controller Address	R/W		0374	0374
Horizontal Total	R/W	00	0375	0375
Horizontal Display Enable	R/W	01	0375	0375
Start Horizontal Blanking	R/W	02	0375	0375
End Horizontal Blanking	R/W	03	0375	0375
Start Horizontal Retrace Plus	R/W	04	0375	0375
End Horizontal Retrace	R/W	05	0375	0375
Vertical Total	R/W	06	0375	0375
Overflow	R/W	07	0375	0375
Preset RowScan	R/W	08	0375	0375
Maximum Scan Line	R/W	09	0375	0375
Cursor Start	R/W	0A	0375	0375
Cursor End	R/W	0B	0375	0375
Start Address High	R/W	0C	0375	0375
Start Address Low	R/W	0D	0375	0375
Cursor Location High	R/W	0E	0375	0375
Cursor Location Low	R/W	0F	0375	0375
Vertical Retrace Start	R/W	10	0375	0375
Vertical Retrace End	R/W	11	0375	0375
Vertical Display Enable End	R/W	12	0375	0375
Offset	R/W	13	0375	0375
Underline Location	R/W	14	0375	0375
Start Vertical Blank	R/W	15	0375	0375
End Vertical Blank	R/W	16	0375	0375
CRTC Mode Control	R/W	17	0375	0375
Line Compare	R/W	18	0375	0375

NOTE:

? = B in Monochrome Emulation Mode

? = D in Color Emulation Mode

All address are given in Hex

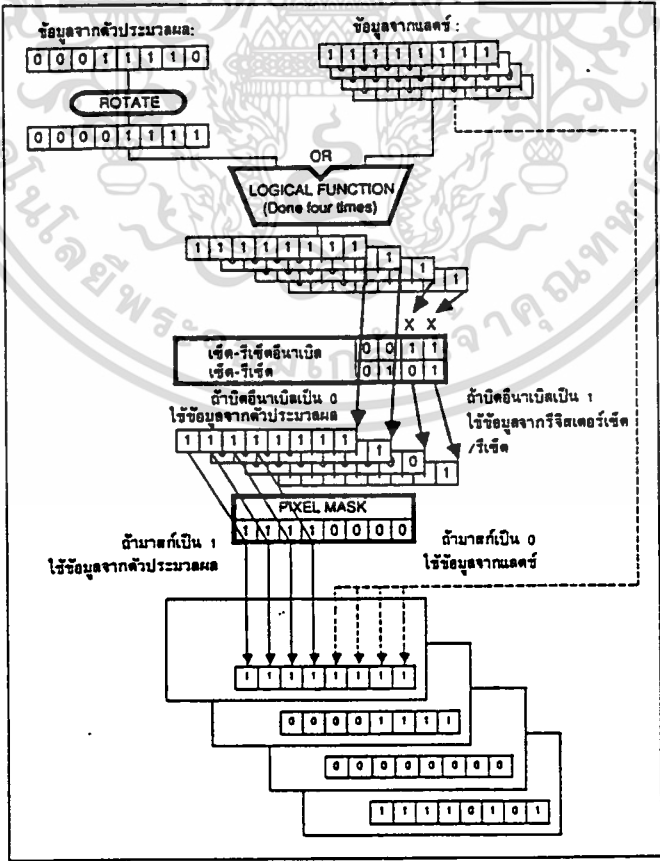
รูปที่ 2.6 ตารางแสดงรีจิสเตอร์ต่าง ๆ ใน CTR Controller

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

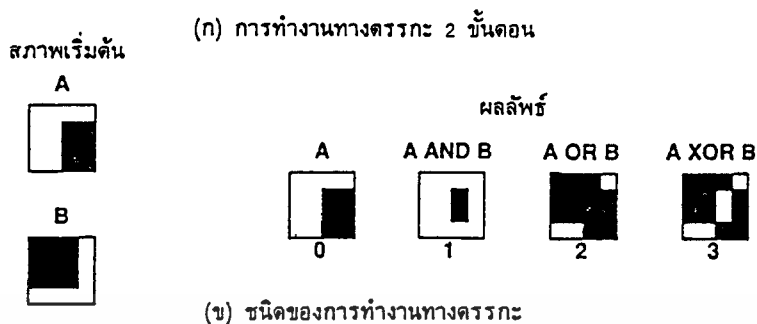
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Name	R/W	Index	Read Port	Write Port
Graphics Address	R/W		03CE	03CE
Set/Reset	R/W	00	03CE	03CE
Enable Set/Reset	R/W	01	03CE	03CE
Color Compare	R/W	02	03CE	03CE
Data Rotate	R/W	03	03CE	03CE
Read Map Select	R/W	04	03CE	03CE
Graphics Mode	R/W	05	03CE	03CE
Miscellaneous	R/W	06	03CE	03CE
Color Don't Care	R/W	07	03CE	03CE
Bit Mask	R/W	08	03CE	03CE

รูปที่ 2.7 รีจิสเตอร์ในกราฟิกคอนโทรลเลอร์

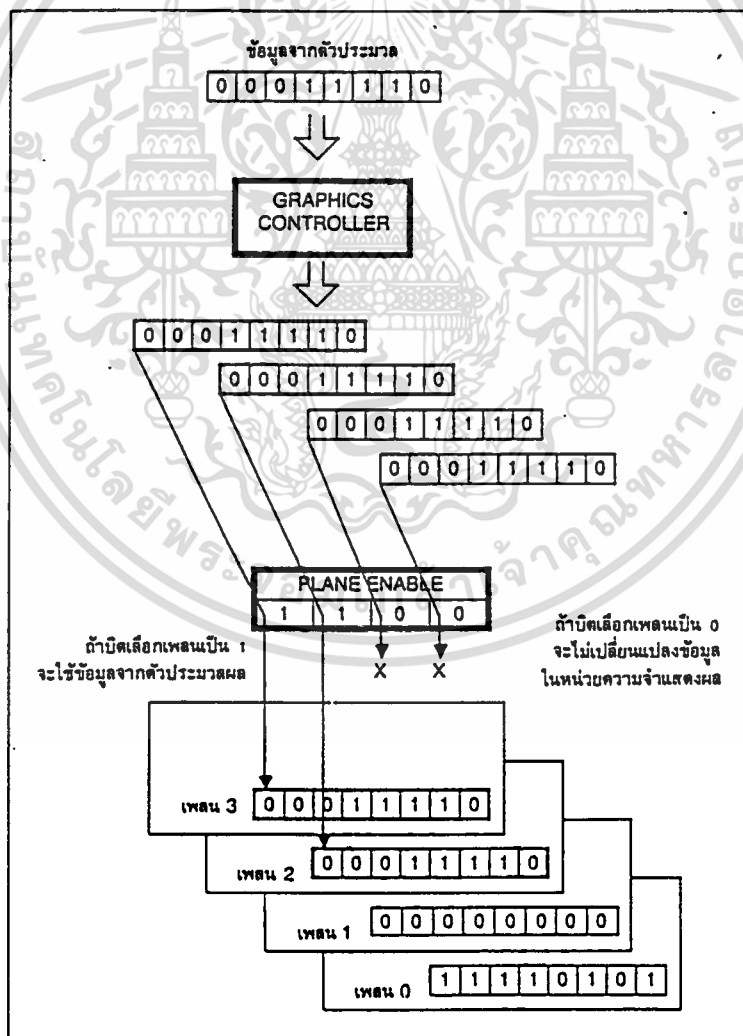


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.8 บล็อกไดอะแกรมของกราฟิกคอนโทรลเลอร์
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่เนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ข) ชนิดของการทำงานทางตรรกะ

รูปที่ 2.9 การกระทำฟังก์ชันทางลอจิกระหว่างข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการเรียนการสอนเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Register Name	R/W	Index	Read Port	Write Port
Sequencer Address	R/W		03C4	03C4
Reset	R/W	00	03C5	03C5
Clocking Mode	R/W	01	03C5	03C5
Map Mask	R/W	02	03C5	03C5
Character Map Select	R/W	03	03C5	03C5
Memory Mode	R/W	04	03C5	03C5

รูปที่ 2.11 ตารางแสดงรีจิสเตอร์ใน Sequencer

3. Sequencer

ทำหน้าที่กำเนิดสัญญาณนาฬิกาที่ควบคุมการรีเฟรชหน่วยความจำควบคุมช่วงเวลา ในการเขียนและอ่านกับหน่วยความจำ และยังมีส่วนตรวจสอบการยอมหรือไม่ยอมให้โปรเซสเซอร์กระทำกับหน่วยความจำเฟลนใดเฟลนหนึ่ง ซึ่งแสดงดังรูปที่ 2.10 และตารางในรูปที่ 2.11 แสดงรีจิสเตอร์ต่าง ๆ ของ Sequencer จากรูปจะเห็นว่าถ้าค่าในรีจิสเตอร์ Map mask (plane enable) เป็น 0 (D0 แทน 0) เฟลนนั้นจะไม่สามารถถูกเปลี่ยนข้อมูลได้

4. Attribute Controller

ตัวควบคุมแอตทริบิวต์ (Attribute Controller) ทำหน้าที่กำหนดสีในการแสดงผล โหมดข้อความและโหมดกราฟิก โดยจะใช้ตารางค้นหาสีทำการแปลงข้อมูลขนาด 4 บิตบนหน่วยความจำแสดงผลเป็น 6 บิตในการแสดงผลแบบ EGA และ 12 บิต ในการแสดงผลแบบ VGA ตารางค้นหาสีจะกำหนดสีเริ่มต้นให้เหมาะสมกับโหมดการทำงานนั้น โดยในโหมด EGA และ VGA ตารางค้นหาสีจะสนับสนุนการแสดงผลได้มากขึ้น โดยสามารถโปรแกรมเลือกชุดสีที่จะใช้แสดงผลที่สามารถแสดงได้ สำหรับการควบคุมแอตทริบิวต์ของการแสดงผล เช่น แอตทริบิวต์ที่เป็นสีต่าง ๆ แอตทริบิวต์ที่แสดงการกะพริบ (blinking) หรือการขีดเส้นใต้ (underline) แอตทริบิวต์คอนโทรลเลอร์ประกอบด้วยรีจิสเตอร์ 20 รีจิสเตอร์ดังแสดงในตารางในรูปที่ 2.12 ในการกระทำ (เขียนหรืออ่าน) กับรีจิสเตอร์ของ VGA นั้นจะใช้คำสั่งที่กระทำกับพอร์ตคือ คำสั่ง In และคำสั่ง Out แต่ละกลุ่มของรีจิสเตอร์จะมีแอดเดรสพอร์ตประจำแต่ละกลุ่ม เช่นกราฟิกคอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โทรลเลอร์จะมีแอดเดรสพอร์ตที่ 3CB เป็นอินเด็กซ์รีจิสเตอร์ละ 3CF เป็นดาต้ารีจิสเตอร์ ไปที่พอร์ต 3CE(03 เป็น อินเด็กซ์ของ Data Rotate) จากนั้นจึงเอา Out ค่า 3CH ไปที่พอร์ต 3CF

สำหรับรีจิสเตอร์อื่นก็ทำนองเดียวกัน แต่สำหรับแอดทริบิวต์คอนโทรลเลอร์ซึ่งมีพอร์ตสำหรับการเขียนอยู่แอดเดรสเดียวกันไหลเกลของการเขียนจะทำให้มีการเปลี่ยนไปมาระหว่างอินเด็กซ์รีจิสเตอร์กับดาต้ารีจิสเตอร์ ซึ่งสามารถขึ้นต้นด้วยคำสั่ง In ที่พอร์ต 3DH หลังจากนั้นคำสั่งแรกที่ Out ไปที่พอร์ต 3C0 จะถูกส่งไปที่อินเด็กซ์รีจิสเตอร์ และคำสั่ง Out คำสั่งต่อไปจะถูกส่งไปที่ดาต้ารีจิสเตอร์ ส่วนสำคัญของแอดทริบิวต์คอนโทรลเลอร์อยู่ที่ตารางค้นหาสี (Color Look Up Table) ซึ่งจะทำการแปลงข้อมูลขนาด 4 บิต ไปเป็นข้อมูลของสีขนาด 6 บิต

Register Name	R/W	Index	Read Port	Write Port
Address	R/W		03C0	03C0
Palette Register	R/W	00-0F	03C1	03C0
Attribute Mode Control	R/W	10	03C1	03C0
Overscan Color	R/W	11	03C1	03C0
Color Plane Enable	R/W	12	03C1	03C0
Horizontal PEL panning	R/W	13	03C1	03C0
Color Select	R/W	14	03C1	03C0

รูปที่ 2.12 ตารางแสดงรีจิสเตอร์ในแอดทริบิวต์คอนโทรลเลอร์

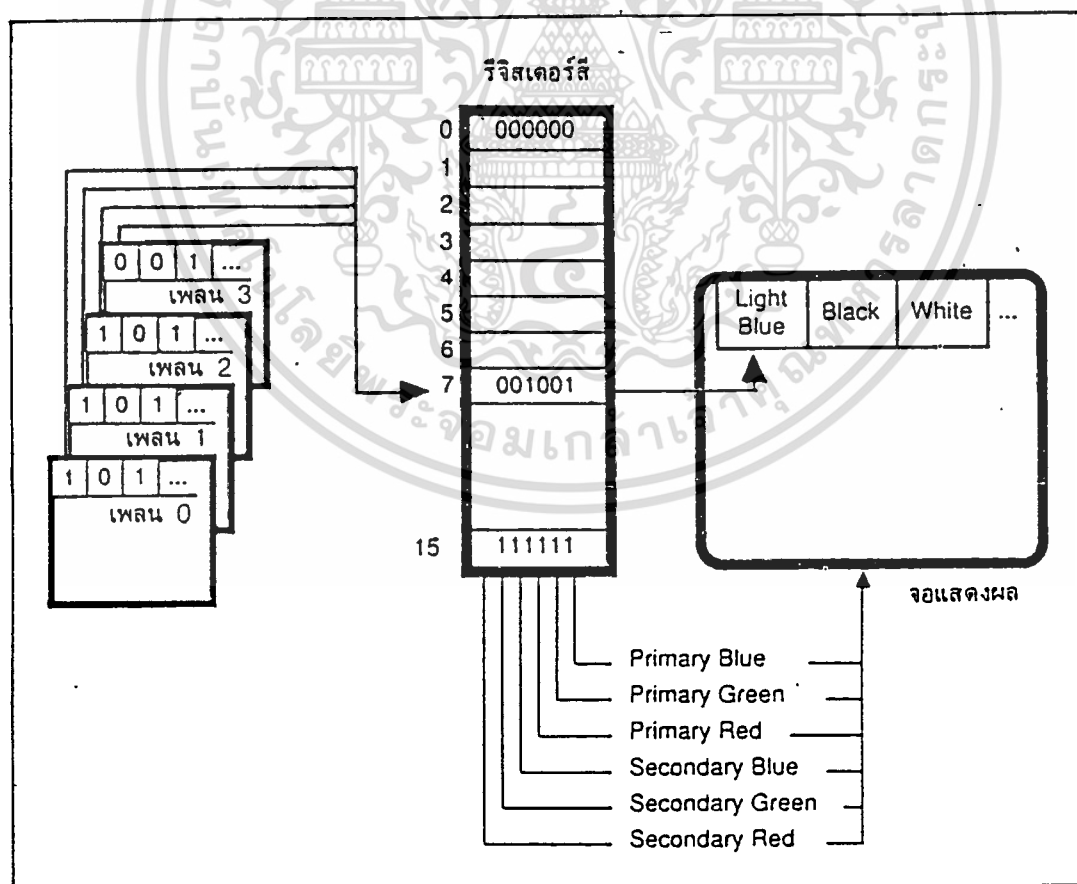
จากรีจิสเตอร์ Color Select อีก 2 บิตรวมเป็น 8 บิต ที่จะส่งไปให้กับ Video DAC ต่อไป แต่สำหรับ EGA ข้อมูลสี 6 บิตจะถูกส่งไปยังจอแสดงผลโดยตรง รูปที่ 2.13 แสดงตารางค้นหาสีของแอดทริบิวต์คอนโทรลเลอร์ในรูป (ก) นั้นเป็นการแสดงในโหมดกราฟิก ค่าของสีจุด ๆ หนึ่ง (pixel) มีค่า 0111 (เท่ากับ 7) ค่าของสีนี้จะถูกใช้เป็นแอดเดรสชี้ไปที่รีจิสเตอร์ที่ 7 ของตารางค้นหาสีซึ่งอยู่ภายในมีค่าเป็น 001001 บิตที่ 0 และบิตที่ 3 แทนสีฟ้าที่มีความเข้มแตกต่างกัน ดังนั้นจุดของภาพ (pixel) จุดแรกจึงมีสีฟ้า ส่วนจุดต่อไปจะมีค่าเป็น 0000 ซึ่งชี้ไปที่รีจิสเตอร์ 0 และค่าภายในเป็น 0 ดังนั้นจุด (pixel) นี้จึงเป็นสีดำในรูปที่ 2.13 (ข) เป็นการแสดงในโหมดตัวอักษรจะเห็นว่า เฟลน 0 เก็บค่า 41 ซึ่งเป็นรหัสแอสกีของตัวอักษร A เฟลน 1 เก็บค่า 07 ซึ่งเป็นแอดทริบิวต์ของตัวอักษร A 4 บิตบนแอดทริบิวต์เป็นตัวกำหนดสีของแบคกราวนด์และ 4 บิตล่างเป็นตัวกำหนดสีของฟรอนท์กราวนด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

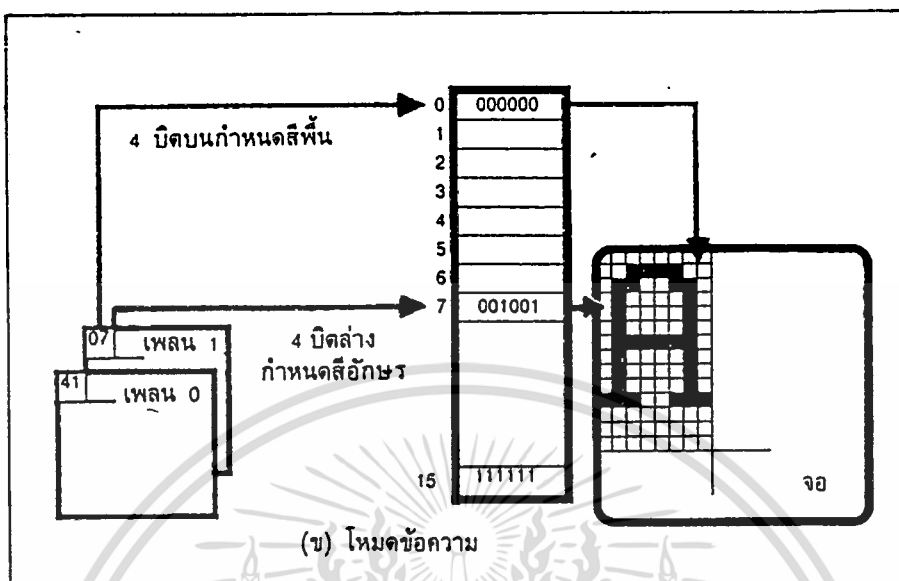
2.4.3 ตัวเปลี่ยนสัญญาณดิจิทัลให้เป็นอะนาล็อก (VIDEO DAC)

ส่วนประกอบหลักตัวสุดท้ายของระบบแสดงผล VGA นี้ก็คือ VIDEO DAC (Digital to Analog Converter) ตัวแปลงข้อมูลดิจิทัลให้เป็นสัญญาณอะนาล็อกเพื่อขับจอแสดงผลไอบีเอ็ม ใช้ VIDEO DAC ของบริษัท Immos เบอร์ IMSC-171 ภายในประกอบด้วยตัวแปลงสัญญาณ 3 ชุด สำหรับ 3 สีคือ แดง เขียว น้ำเงิน และตารางค้นหาสี (Color look-up table)

โดยที่ตารางค้นหาสีทำหน้าที่ที่รับข้อมูลขนาด 8 บิตจากแอดทรีวิวด์คอนโทรลเลอร์มาเป็นตัวรีจิสเตอร์ขนาด 18 บิต 1 ใน 256 รีจิสเตอร์ เพื่อเลือกข้อมูลที่จะนำไปแปลงเป็นสัญญาณอะนาล็อกเพื่อใช้ในการขับสัญญาณส่งให้กับจอภาพ ตารางค้นหาสีเป็นตัวแทนที่มีความสำคัญในขบวนการเกี่ยวกับสี ใน VIDEO DAC รีจิสเตอร์แต่ละตัวทำหน้าที่กำหนดสีแต่ละสี เราพบว่าใน VIDEO DAC เป็นรีจิสเตอร์ขนาด 18 บิตเพราะฉะนั้น จากรีจิสเตอร์จำนวน 256 รีจิสเตอร์ ทำให้สามารถแสดงสีได้พร้อมกัน 256 สี



เอกสารนี้เป็นรูปที่ 2.13 (ก) แสดงการแปลงข้อมูลจากหน่วยความจำไปเป็นรูปจุดในโหมดกราฟิก
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 (ข) การแปลงรหัสไปเป็นตัวอักษรพร้อมทั้งแอตทริบิวต์

แต่ภายใน 1 รีจิสเตอร์ ซึ่งมีขนาด 18 บิตจะมีสีที่แตกต่างกันได้ทั้งสิ้น $2^{18} = 256 \text{ K}$ สี และนี่คือเหตุผลที่ว่าระบบ VGA สามารถแสดงสีได้ 256 สีพร้อมกัน จากจำนวนสีทั้งสิ้น 256 K

ที่กล่าวมาทั้งหมดคือโครงสร้างทางฮาร์ดแวร์ของการ์ด VGA ซึ่งจะเห็นว่าภายใน VGA มีรีจิสเตอร์ต่างๆ มากมายซึ่งแต่ละตัวก็มีหน้าที่แตกต่างกันไป การที่จะแก้ไขค่าในรีจิสเตอร์นั้นควรมีการศึกษาก่อนว่ารีจิสเตอร์ตัวนั้นถ้าแก้ไขแล้วจะทำให้เกิดความเสียหายหรือไม่ อย่างเช่นรีจิสเตอร์บางตัวของ CRT Controller ถ้ามีค่าไม่เหมาะสมจะทำให้จอภาพเสียหายได้ การใช้ฟังก์ชันต่าง ๆ ของการ์ด VGA นอกจากจะทำกับรีจิสเตอร์โดยตรงแล้วยังสามารถทำได้โดยเรียกใช้บริการของรอมไบออสบนการ์ด ฟังก์ชันต่าง ๆ เหล่านี้ได้แก่ ฟังก์ชันเซตโหมดของการแสดงผล ฟังก์ชันอ่านโหมดของการแสดงผล ฟังก์ชันเซตพาร์เล็ลรีจิสเตอร์และฟังก์ชันโหลดค่าลงในตารางแปลงอักขระ เป็นต้น

2.5 การโปรแกรมใช้การ์ด VGA

หลังจากที่เราทราบโครงสร้างพื้นฐานของการ์ด VGA แล้ว การโปรแกรมเพื่อที่จะใช้งานการ์ด VGA เป็นที่สำคัญที่ควรจะทราบในลำดับถัดไป เนื่องจากการ์ด VGA มีความสลับซับซ้อนมากการใช้งานก็สามารถทำได้หลายรูปแบบดังนั้นในบทความนี้จะกล่าวถึงวิธีการ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



โปรแกรมใช้งานการ์ด VGA เบื้องต้น ก่อนอื่นใดเราควรทำความรู้จักกับโหมดการแสดงผลที่เป็นมาตรฐาน ซึ่งการ์ด VGA สามารถแสดงได้ในตารางรูปที่ 2.14 จะแสดงให้เห็นโหมดต่าง ๆ ที่ระบบ VGA สนับสนุนโหมดการแสดงผลของการ์ดแสดงผล MDA และ CGA ในระบบเก่าด้วยต่อไปจะขออธิบายรายละเอียดของแต่ละโหมดการแสดงผลดังต่อไปนี้

2.5.1 การแสดงผลโหมด 0 และ 1 (สำหรับข้อความมีสี)

สำหรับในโหมดการแสดงผลโหมดนี้นั้นเป็นของการ์ด CGA บนการ์ด VGA ฟังก์ชันการทำงานทุกอย่างเหมือนกับการ์ด CGA ทุกประการ เพียงแต่ไม่มีขั้วต่อสำหรับสัญญาณวีดีโอคอมโพสิต จอภาพที่ใช้ในโหมดนี้จะเป็นจอ Color Display (CD), Enhanced Color Display (ECD), จอ VGA หรือจอหลายความถี่ (multifrequency) ที่มีความละเอียด 40 ตัวอักษร 25 บรรทัด ฟอนต์ตัวอักษรในโหมดนี้มีขนาด 3 x 8 จุดโหมดนี้เป็นโหมดข้อความที่มีรายละเอียดต่ำสุด การ์ด VGA ในโหมด 0 และหนึ่งนี้ มี 8 เพจ Start Address register แสดงดังรูปที่ 2.15

สำหรับซอฟต์แวร์ที่ทำงานบนระบบ CGA เดิมจะสามารถทำงานบนการ์ด VGA ได้ แต่จะต้องเป็นซอฟต์แวร์ที่ใช้ฟังก์ชันบริการของไบออสซอฟต์แวร์ที่กระทำโดยตรงต่อหน่วยความจำแสดงผลหรือติดต่อโดยตรงกับรีจิสเตอร์อินพุท/เอาต์พุทของการ์ดแสดงผล การ์ด VGA ในโหมด 0 และ 1 นี้มี 8 เพจ การเลือกเพจสำหรับแสดงผลอาจจะควบคุมผ่านไบออสหรือผ่านทาง Start Address register ใน CRT controller สำหรับเรื่องสีก็เป็นไปตามมาตรฐานคือ จะประกอบด้วย 8 สีตามปกติ และอีก 8 สี เมื่อเพิ่มความเข้ม (intensity) สำหรับจอภาพ VGA ที่มีความละเอียดสูงเมื่อแสดงผลในโหมดนี้ VGA จะใช้เทคนิคพิเศษในการจัดให้อัตราส่วนของภาพในแนวตั้งใกล้เคียงกับจอ CGA ซึ่งจอ CGA มีความละเอียดต่ำคือ 200 เส้นสแกน แต่ถ้เป็นจอ VGA ใน 200 เส้นจะถูกสแกน 2 ครั้ง เป็น 400 เส้นสแกน เรียกเทคนิคนี้ว่าเป็น “Double Scanning” เทคนิคนี้จะใช้ในโหมด 1,2,3,4,5,6,D, และ E

2.5.2 การแสดงผลโหมด 0' และ 1' (โหมดข้อความสี)

โหมด 0' และ 1' เป็นโหมดที่เหนือขึ้นมาจากโหมด 0 และ 1 การแสดงผลยังเป็นขนาด 40 ตัวอักษร 25 บรรทัด แต่ขนาดของตัวอักษรซึ่งในโหมดที่ 0 และ 1 มีขนาด 8x8 จุดก็จะเป็นขนาด 8x14 จุดแทน ดังนั้นในโหมดนี้จอภาพ CD จึงใช้ไม่ได้ โหมดนี้จะใช้ได้กับจอ ECD, VGA หรือจอหลายความถี่ (Multifrequency) การปรับขนาดของเคอร์เซอร์และการขีดเส้นได้เช่นเดียวกับโหมด 0 และ 1 โดยจะมี 8 เพจ การแสดงผลในแต่ละแต่ละเพจก็จะมีตำแหน่งเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า หน่วยความจำดังรูปที่ 2.18

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode	Type	Color	Resolution	Compatible Display
0, 1	Color text	16	40x25 8x8 char cell	CD, ED, VGA Multifrequency
0°, 1°	Color text	16	40x25 8x14 char cell	ED, VGA Multifrequency
0+ , 1+	Color text	16	40x25 9x16 char cell	VGA, Multifrequency
2, 3 →	Color text	16	80x25 8x8 char cell	CD, ED, VGA Multifrequency
2°, 3° →	Color text	16	80x25 8x14 char cell	ED, VGA Multifrequency
2+, 3+	Color text	16	80x25 9x16 char cell	VGA Multifrequency
4, 5	Color graphics	4	320x200	CD, ED, VGA Multifrequency
6	Color graphics	2	640x200	CD, ED, VGA Multifrequency
7 →	Monochrome text	2	80x25 8x14 char cell	Monochrome VGA
7	Monochrome text		80x25 9x16 char cell	VGA only
8, 9, A	Pos only			
D →	Color graphics	16	320x200	CD, ED, VGA Multifrequency
B →	Color graphics	16	640x200	CD, ED, VGA Multifrequency
F →	Mono graphics		640x350	Monochrome VGA
10 →	Color graphics	16	640x350	ED, VGA Multifrequency
11	Color graphics	2	640x480	VGA Multifrequency
12	Color graphics	16	640x480	VGA Multifrequency
13	Color graphics	256	320x200	VGA Multifrequency

Most multifrequency display are VGA compatible

CD = Color Display

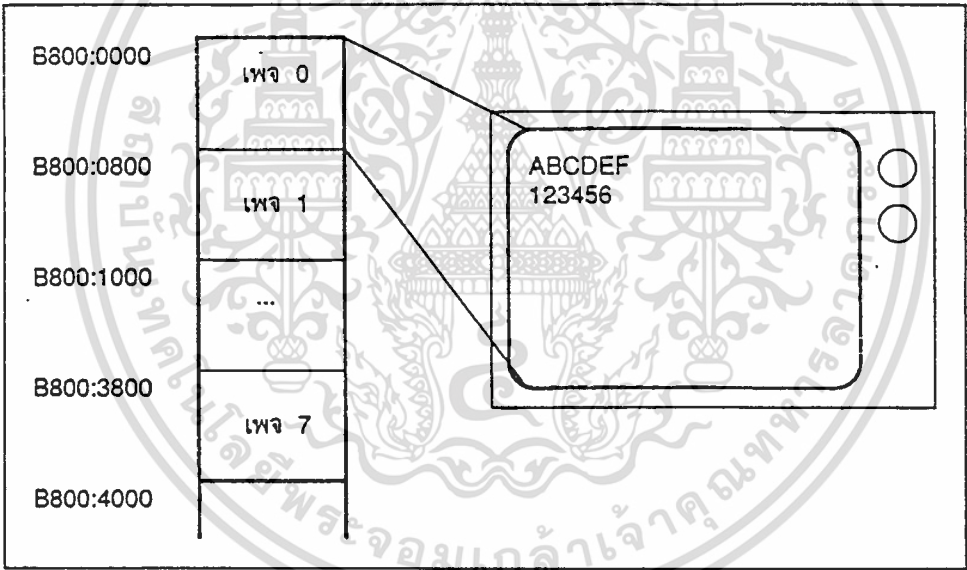
The original NEC Multisync is not

ED = Enhanced Color Display

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้ในการเรียนเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.14 ตารางแสดงโหมดแสดงผลมาตรฐานของระบบวิดีโอ IBM
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพลง 0 ตำแหน่ง B800:0000	เพลง 4 ตำแหน่ง B800:2000
เพลง 1 ตำแหน่ง B800:0800	เพลง 5 ตำแหน่ง B800:2800
เพลง 2 ตำแหน่ง B800:1000	เพลง 6 ตำแหน่ง B800:3000
เพลง 3 ตำแหน่ง B800:1800	เพลง 7 ตำแหน่ง B800:3800

รูปที่ 2.15



รูปที่ 2.16 แสดงหน่วยความจำแต่ละเพลงในโหมด 0 และ 1

Attribute	Standard Color	Intensified Color
0 0 0	Black	Gray
0 0 1	Blue	Light Blue
0 1 0	Green	Light Green
0 1 1	Cyan	Light Cyan
1 0 0	Red	Light Red
1 0 1	Magenta	Light Magenta
1 1 0	Brown	Yellow
1 1 1	Gray	White

รูปที่ 2.17 ตารางแสดงสีของอักขรในโหมดข้อความที่มีสี

เพจ 0 ตำแหน่ง B800:0000	เพจ 4 ตำแหน่ง B800:2000
เพจ 1 ตำแหน่ง B800:0800	เพจ 5 ตำแหน่ง B800:2800
เพจ 2 ตำแหน่ง B800:1000	เพจ 6 ตำแหน่ง B800:3000
เพจ 3 ตำแหน่ง B800:1800	เพจ 7 ตำแหน่ง B800:3800

รูปที่ 2.18

2.5.3 การแสดงผลโหมด 0+ และ 1+ (โหมดข้อความ สี)

สำหรับโหมดนี้ก็จะเป็โหมดที่เพิ่มขึ้นมาจากการแสดงผลในโหมด 0 และ 1 ของ CGA ซึ่งมีความละเอียดของจอจะเป็น 40 ตัวอักษรต่อคอลัมน์และ 25 บรรทัด แต่ชุดอักขรที่ใช้ในโหมดของ CGA ที่เคยมีขนาด 8x8 จุดก็จะใช้ชุดตัวอักษรของ VGA แทนซึ่งมีขนาด 9x16 จุด

2.5.4 การแสดงผลโหมด 2 และ 3 (โหมดข้อความสี)

เช่นเดียวกันในโหมด 0 กับโหมด 1 ก็ทั้งโหมด 2 กับโหมด 3 การทำงานทั้งสองโหมดเหมือนกันทุกประการในโหมดนี้ก็จะมีความละเอียด 80 คอลัมน์ 25 บรรทัด และ สำหรับจอภาพที่ใช้ในโหมดนี้คือ CD,BCD,VGA หรือจอภาพหลายความถี่ ตัวอักษรที่ปรากฏจะมีขนาด 8x8 จุดต่อตัวอักษร การแสดงผลแต่ละเพจจะมีหน่วยความจำต่างกัันดังรูปที่ 2.19

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานในหน่วยงานราชการ ไม่ควรเผยแพร่ไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจ 0 ตำแหน่ง B800:0000	เพจ 4 ตำแหน่ง B800:4000
เพจ 1 ตำแหน่ง B800:1000	เพจ 5 ตำแหน่ง B800:5000
เพจ 2 ตำแหน่ง B800:2000	เพจ 6 ตำแหน่ง B800:6000
เพจ 3 ตำแหน่ง B800:3000	เพจ 7 ตำแหน่ง B800:7000

รูปที่ 2.19

2.5.5 การแสดงผลโหมด 2' และ 3' (โหมดข้อความสี)

โหมด 2' และ 3' ต่างจากโหมด 2 และ 3 คือ ขนาดของตัวอักษรที่ใช้แสดงจะเป็น 8 x14 จุดต่อตัวอักษร ซึ่งจะทำให้ไม่สามารถใช้กับจอภาพ เช่น CD ได้ แต่ใช้จอภาพแบบ EDC, VGA และจอภาพหลายความถี่และเนื่องจากความละเอียดของอักษรมากกว่าในโหมด 2 และ 3 ดังนั้น ฟังก์ชันเกี่ยวกับการเปลี่ยนแปลงขนาดของเคอร์เซอร์จึงแตกต่างกันจำนวนเพจสำหรับแสดงผลก็มี 8 เพจ เช่นเดียวกับโหมด 2 และ 3 ตำแหน่งของหน่วยความจำแต่ละเพจก็ตรงกับ 2 และ 3

2.5.6 การแสดงผลโหมด 2+ และ 3+ (โหมดข้อความสี)

โหมดนี้ก็เช่นเดียวกับโหมดการแสดงผลเพิ่มขึ้นมาจาก 2 และ 3 ของมาตรฐานระบบ CGA ก็คือมี 80 ตัวอักษรคอลัมน์และ 25 บรรทัด แต่ชุดตัวอักษรจะเป็นชุดของ VGA แทน ซึ่งตัวอักษร 1 ตัวมีขนาดกว้าง 9 จุดและสูง 16 จุด

2.5.7 การแสดงผลโหมด 4 และ 5 (โหมดกราฟิก 4 สี ความละเอียด 320x200 จุด)

ในโหมด 4 และ 5 เป็นโหมดแสดงกราฟิกของระบบ CGA ที่นิยมมากที่สุด ดังนั้นจึงไม่ต้องสงสัยเลยว่าในระบบ EGA หรือ VGA จะมีการแสดงผลโหมดนี้ด้วยความละเอียดของระบบนี้คือ 320 จุดในแนวนอน และ 200 จุดในแนวตั้ง จอภาพที่ใช้กับโหมดนี้คือจอ CD, ECD, VGA และจอภาพหลายความถี่

ในโหมด 4 และ 5 สีที่ใช้แสดงมีอยู่ด้วยกัน 4 สี ซึ่งอาจจะมาจากกลุ่มสี 2 กลุ่มดังในตารางในรูป 2.20 สำหรับซอฟต์แวร์บนระบบ CGA เดิมที่มีการสั่งงานหรือควบคุมจอภาพผ่านทางรีจิสเตอร์อินพุต/เอาต์พุตของการ์ด CGA เมื่อนำมาใช้งานในโหมดนี้บนการ์ดแสดงผล EGA และ VGA อาจจะทำงานได้ไม่ถูกต้อง แต่เป็นซอฟต์แวร์ที่ควบคุมภาพทางไบออสจะทำงานได้ทั้งไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี่เนื่องจากการ์ดแสดงผล EGA และ VGA ไม่คอมแพตติเบิลกับจอ CGA ทั้งหมด ในโหมดนี้จะมีเพจสำหรับแสดงผลเพียงอย่างเดียว โดยเริ่มที่ตำแหน่งหน่วยความจำ B800:0000 ข้อมูลของจุดบนจอจะเหมือนในระบบ CGA เดิมทุกประการคือหนึ่งจุดจะแทนด้วย 2 บิตการ์ด CGA รุ่นเก่าซึ่งใช้ชิพของโมโตโรล่า 6845 การสแกนในแนวดิ่งจะถูกจำกัดอยู่ที่ 128 เส้นสแกน เพราะจำนวนเส้นสแกนที่น้อยภาพที่ได้จึงมีความคมชัดของภาพที่ลดน้อยลง

Standard Color	Alternate Color
Black	Black
Light Cyan	Green
Light Magenta	Red
White	Brown

รูปที่ 2.20 ตารางแสดงสีมาตรฐานในโหมด 4 และ 5

ดังนั้นเพื่อให้ความละเอียดในแนวดิ่งสูงถึง 200 เส้น จึงต้องโปรแกรมให้ CRT controller ทำงานในโหมดข้อความโดยมี 100 แถว แต่สำหรับจำนวนจุดในความสูงของตัวอักษรจะมีเพียง 2 จุดเท่านั้น ดังนั้นสิ่งที่แสดงบนจอภาพจึงต้องอาศัยภาพที่ได้มาจากรูปร่างของตัวอักษรซึ่งทำให้ตำแหน่งหน่วยความจำไม่ต่อเนื่องกันเหมือนตำแหน่งของจุดบนจอภาพจึงต้องใช้การคำนวณ

2.5.8 การแสดงผลโหมด 6 (โหมดกราฟิก 2 สี ความละเอียด 640x200จุด)

โหมด 6 เป็นการแสดงผลที่มีความละเอียดสูงสุดในโหมดกราฟิกของระบบ CGA โดยมีความละเอียด 640 จุดในแนวนอน และ 200 จุดในแนวดิ่งจอภาพที่ใช้ในโหมดนี้คือ จอภาพ CD, ECD, VGA และจอภาพหลายความถี่ เช่นเดียวกับโหมดกราฟิกก่อนหน้านี้คือซอฟต์แวร์ที่ใช้งานบน CGA อาจจะทำไม่ได้ไม่ถูกต้องถ้ามาทำงานบนการ์ด EGA หรือ VGA ในกรณีที่ซอฟต์แวร์ตัวนั้นไม่ได้ควบคุมจอภาพผ่านทางไบออส

ดังอธิบายในโหมด 4 และ 5 แล้วตำแหน่งของหน่วยความจำของโหมด 6 จึงเหมือนกับโหมด 4 และ 5 คือตำแหน่งหน่วยความจำจะไม่เรียงต่อกัน และในโหมดนี้จะแสดงผลเพียงเพจเดียวคือที่ตำแหน่ง B800:0000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.9 การแสดงผลในโหมด 7 (โหมดข้อความของโมโนโครม)

ในโหมด 7 เป็นโหมดของข้อความซึ่งอยู่ในระบบ Monochrome Display Adapter (MDA) จอภาพที่ต้องใช้กับโหมดนี้คือจอโมโนโครม หรือจอ VGA ที่มีความละเอียดของตัวอักษร 80 ตัวอักษรต่อคอลัมน์ 25 แถว หนึ่งตัวอักษรประกอบด้วยจุด 9x14 จุด (เพื่อให้ได้จำนวนจุดในแนวนอนเป็น 720 จุด) โดยที่บิตที่ 8 หรือบิตสุดท้ายของข้อมูลจะถูกขยายออกไปเป็นบิตที่เก้า ซึ่งช่วยให้การเขียนตัวอักษรที่ใช้สำหรับทำกรอบสี่เหลี่ยม (ตั้งแต่รหัสแอสกี C0 hex ถึง DF hex) มีรอยต่อเชื่อมสนิท อย่างไรก็ตามการขยายบิตที่ 9 สามารถจะควบคุมได้

โหมดนี้มีเพจสำหรับแสดงผล 8 เพจ (ยกเว้นกรณีการ์ด EGA ของไอบีเอ็มซึ่งมีหน่วยความจำ 64 กิโลไบต์ จะมี 4 เพจ) โดยที่แต่ละเพจมีตำแหน่งหน่วยความจำดังแสดงดังรูปที่ 2.21 ส่วนแอตทริบิวต์ของตัวอักษรก็ยังคงเป็นเช่นเดิม

เพจ 0 ตำแหน่ง B800:0000	เพจ 4 ตำแหน่ง B000:4000
เพจ 1 ตำแหน่ง B800:1000	เพจ 5 ตำแหน่ง B000:5000
เพจ 2 ตำแหน่ง B800:2000	เพจ 6 ตำแหน่ง B000:6000
เพจ 3 ตำแหน่ง B800:3000	เพจ 7 ตำแหน่ง B000:7000

รูปที่ 2.21

2.5.10 การแสดงผลโหมด 7+ (โหมดข้อความโมโนโครม)

โหมดนี้เป็นโหมดเพิ่มขึ้นจากโหมดข้อความของ MDA มาตรฐานโดยชุดตัวอักษรของ VGA ขนาด 9x16 จุด

2.5.11 การแสดงผลโหมด D (โหมดกราฟิก 16 สี 320x200 จุด)

ในโหมด D นี้ไม่เหมือนการแสดงผลโหมดต่างๆ ที่ผ่านมา เพราะว่าไม่ใช่ทำเพื่อความคมชัดแต่เบิ้ลกับระบบแสดงผลเก่า ถึงแม้ความละเอียดจะเท่ากับโหมด 4 ของ CGA แต่ในโหมด D สามารถแสดงสีได้ถึง 16 สีและเพราะความละเอียดของจอภาพที่จำกัดไว้ที่ 300x200 จุด จึงทำให้ซอฟต์แวร์จำนวนน้อยมากที่ทำงานบนโหมดนี้ หน่วยความจำ 256 ,128 และ 64 กิโลไบต์จะมีเพจแสดงผลได้ 8 เพจ 4 เพจและ 2 เพจตามลำดับ แต่ละเพจอยู่ที่ตำแหน่งหน่วยความจำแสดงดัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 2.22
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจ 0 ตำแหน่ง A000:0000	เพจ 4 ตำแหน่ง A000:4000
เพจ 1 ตำแหน่ง A000:1000	เพจ 5 ตำแหน่ง A000:5000
เพจ 2 ตำแหน่ง A000:20.0	เพจ 6 ตำแหน่ง A000:6000
เพจ 3 ตำแหน่ง A000:3000	เพจ 7 ตำแหน่ง A000:7000

รูปที่ 2.22

ในโหมดนี้ตำแหน่งของจุดกับตำแหน่งหน่วยความจำจะเรียงต่อกันเรื่อยไป และมีสีใช้งาน 16 สีอยู่ในตารางรูปที่ 2.24

2.5.12 การแสดงผลโหมด E (กราฟิก 16 สี ขนาด 640 x 200 จุด)

โหมด E กับโหมด 6 จะเป็นกราฟิกความละเอียดเท่ากันต่างกันว่าโหมด E สามารถแสดงสี 16 สี และเช่นเดียวกันโหมด D เนื่องจากความละเอียดยังไม่ดีพอ ซอฟต์แวร์สมัยใหม่จึงไม่นิยมทำงานในโหมด E นี้ สำหรับการ์ด EGA ที่มีหน่วยความจำ 256 กิโลไบต์สามารถจะแสดงผลในโหมดนี้ได้ถึง 4 เพจ ถ้ามีหน่วยความจำ 128,64 กิโลไบต์จะแสดงได้ 2 และ 1 เพจตามลำดับ ตำแหน่งหน่วยความจำแต่ละเพจจะเป็นดังรูปที่ 2.23 ในโหมดนี้นั้นตำแหน่งของจุดกับตำแหน่งหน่วยความจำจะเรียงต่อกันและสีที่ใช้งานจะอยู่ในตารางที่ 2.24

เพจ 0	ตำแหน่ง	A000:0000
เพจ 1	ตำแหน่ง	A000:4000
เพจ 2	ตำแหน่ง	A000:8000
เพจ 3	ตำแหน่ง	A000:C000

รูปที่ 2.23

2.5.13 การแสดงผลโหมด F (โหมดกราฟิก ขาวดำ 640 x 350)

เฉพาะการ์ด EGA และ VGA เท่านั้นที่สามารถแสดงผลในโหมดนี้เพราะว่าไม่ใช่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้แก้ไขโดยไม่ขออนุญาตจากเจ้าของลิขสิทธิ์
โหมดการแสดงผลที่สร้างขึ้นเพื่อให้คอมพิวเตอร์เชื่อมต่อกับการ์ดใด ๆ จอภาพที่ใช้กับโหมดนี้คือ จอ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูล และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Monochrome VGA ซึ่งต้องมีความละเอียดในแนวนอน 640 จุด และความละเอียดในแนวตั้ง 350 จุด (จะพบว่าความละเอียดน้อยกว่าโหมดของเซอร์คิวลิสที่มีความละเอียด 720 x 380)

ในโหมด F ตำแหน่งของจุดบนจุดถึงตำแหน่งของหน่วยความจำจะเรียงต่อกัน ไม่เหมือนกับโหมดกราฟิกของเซอร์คิวลิส การแสดงผลจะมีเพียง 2 เพจ ยกเว้นการ์ดแสดงผล EGA ของไอบีเอ็มที่มีหน่วยความจำแสดงผลเพียง 64 กิโลไบต์จะมีการแสดงผลได้เพียง 1 เพจ ตำแหน่งหน่วยความจำของแต่ละเพจจะเป็นดังรูปที่ 2.25

Plane	Full (128k+) Color	Fame (64KB) Color
0 0 0 0	Black	Black
0 0 0 1	Blue	Blue
0 0 1 0	Green	Black
0 0 1 1	Cyan	Blue
0 1 0 0	Red	Red
0 1 0 1	Magenta	White
0 1 1 0	Brown	Red
0 1 1 1	White	White
1 0 0 0	Dark Gray	Black
1 0 0 1	Light Blue	Blue
1 0 1 0	Light Green	Back
1 0 1 1	Light Cyan	Blue
1 1 0 0	Light Red	Red
1 1 0 1	Light Magenta	White
1 1 1 0	Yello	Red
1 1 1 1	Intens White	White

รูปที่ 2.24 ตารางแสดงสีมาตรฐาน 16 สี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพจ	0	ตำแหน่ง	A000:0000
เพจ	1	ตำแหน่ง	A000:8000

รูปที่ 2.25

ส่วนเรื่องของสีนั้นจะเกิดมาจากหน่วยความจำเฟลนสี 2 เฟลน ซึ่งจะช่วยให้จุดเด่นแต่ละจุดมีแอดทริบิวต์ได้ดังรูปที่ 2.26

00	:	ดำ
01	:	ขาว
10	:	กระพริบ
11	:	มีความเข้ม

รูปที่ 2.26

2.5.14 การแสดงผลโหมด 10 H (โหมดกราฟิกส์แบบ Enhanced ขนาด 640x350 จุด)

ในโหมด 10 H เป็นที่เพิ่มขึ้นใน EGA และ VGA โหมดนี้เป็นโหมดที่นิยมใช้กับโปรแกรมกราฟิกทั่วไป ความละเอียดของโหมดนี้จะมีถึง 640 จุดในแนวนอน และ 350 จุดในแนวตั้ง โหมดนี้จะใช้กับจอภาพชนิด CD ไม่ได้ต้องใช้กับจอภาพ ECD, VGA หรือจอหลายความถี่ การควบคุมสีใช้เฟลนสีถึง 4 เฟลน ดังนั้น สามารถแสดงผลได้ถึง 16 สีในเวลาเดียวกัน ยกเว้นการ์ด EGA ของไอบีเอ็มที่มีหน่วยความจำเพียง 64 กิโลไบต์ ซึ่งจะใช้เฟลนสีได้แค่ 2 เฟลน ในโหมด 10 H จะแสดงผลได้ 2 เพจ โดยหน่วยความจำแต่ละเพจดังรูปที่ 2.27

เพจ	0	ตำแหน่ง	A000:0000
เพจ	1	ตำแหน่ง	A000:8000

รูปที่ 2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สีที่ใช้ในโหมด 10 H อยู่ในตารางรูปที่ 2.24 โหมดการแสดงผลต่อไปนี้เป็นโหมดการแสดงผลที่มีเพิ่มขึ้นเฉพาะในระบบ VGA เท่านั้น

2.5.15 การแสดงผลโหมด 11 H (โหมดกราฟิก 2 สี 640x480 จุด)

เป็นโหมดการแสดงผลกราฟิกที่มีความละเอียดสูงสุดแต่แสดงได้เพียง 2 สี ในโหมดนี้สามารถใช้แสดงข้อความได้ถึง 30 แถว แถวละ 80 คอลัมน์ หน่วยความจำแสดงผลเริ่มต้นที่ A000:0000

2.5.16 การแสดงผลโหมด 12 H (โหมดกราฟิก 16 สี 640x480 จุด)

เป็นโหมดการแสดงผลกราฟิกที่มีความละเอียดและแสดงสีได้ถึง 16 สีพร้อมกันดังในตารางรูปที่ 2.24 หน่วยความจำแสดงผลเริ่มต้นที่ A000:0000

2.5.17 การแสดงผลโหมด 13 (โหมดกราฟิก 256 สี 320x200 จุด)

ในกราฟิกโหมดนี้สามารถแสดงสีพร้อมกันได้ถึง 256 สี แต่ความละเอียดจะลดลงเหลือเพียง 320 x 200 จุด หน่วยความจำแสดงผลเริ่มต้นที่ตำแหน่ง A000:0000

2.6 หน่วยความจำแสดงผลแต่ละโหมด

เนื่องจากในโครงงานนี้เป็นโครงงานที่ใช้กับจอภาพชนิด VGA ขึ้นไปจึงจะขออธิบายหลักการของหน่วยความจำแสดงผลให้สามารถเปรียบเทียบกันได้ตั้งแต่โหมด F ขึ้นไปดังนี้

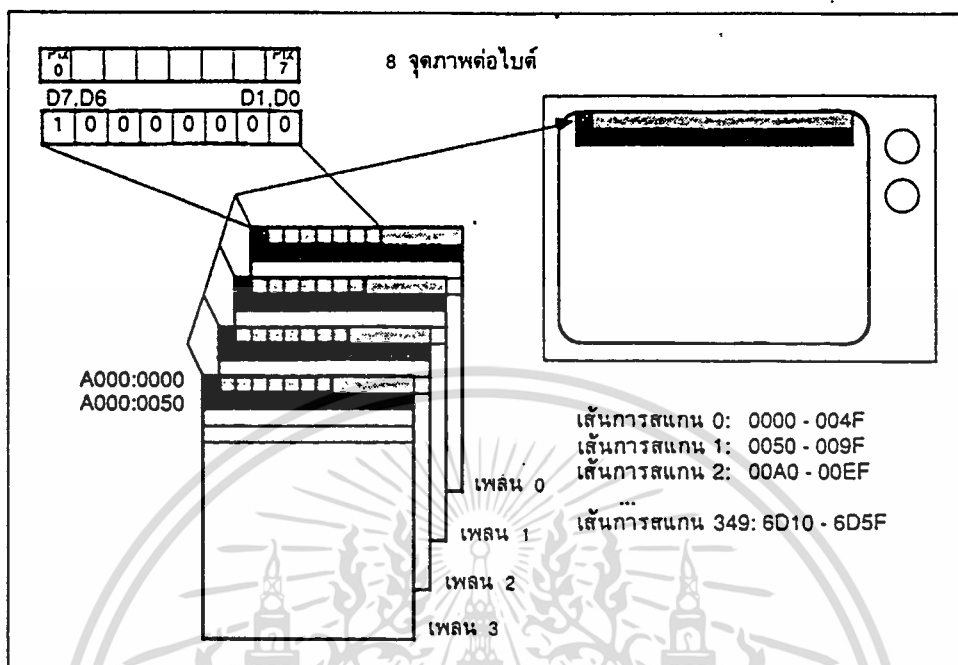
2.6.1 หน่วยความจำแสดงผลในโหมด F (กราฟิกของโมโนโครม)

ในโหมดนี้การจัดเรียงหน่วยความจำแสดงผลจะต่อเนื่องกันความละเอียด 640x350 จุด หนึ่งจุดบนจอจะถูกแทนด้วยบิตบนเพลนสี (Color Plane) คือเพลน 0 และเพลน 1 ดังนั้น ข้อมูลใน 1 ไบต์จะแทนได้ 8 จุด สีจะถูกควบคุมด้วยบิตเพลน 0 และเพลน 1 ซึ่งจะได้ 4 สี คือ ดำ สว่าง สว่างเข้มและกระพริบทั้งสองเพลนสามารถควบคุมได้ทาง Color Write Enable Register ของ Sequencer ในรูปที่ 2.28 แสดงหน่วยความจำที่สัมพันธ์กับจุดบนจอภาพและการหาตำแหน่งหน่วยความจำจะหาได้จากสูตร (อ้างอิงจากจุดบนซ้ายมือ 0, 0 และจุดล่างขวามือ 639, 349)

$$\text{ตำแหน่งไบต์} = (y \times 80) + (x/8)$$

$$\text{ตำแหน่งบิต} = 7 - (X \text{ modulo } 8)$$

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 การจัดเรียงหน่วยความจำในโหมดกราฟิก F

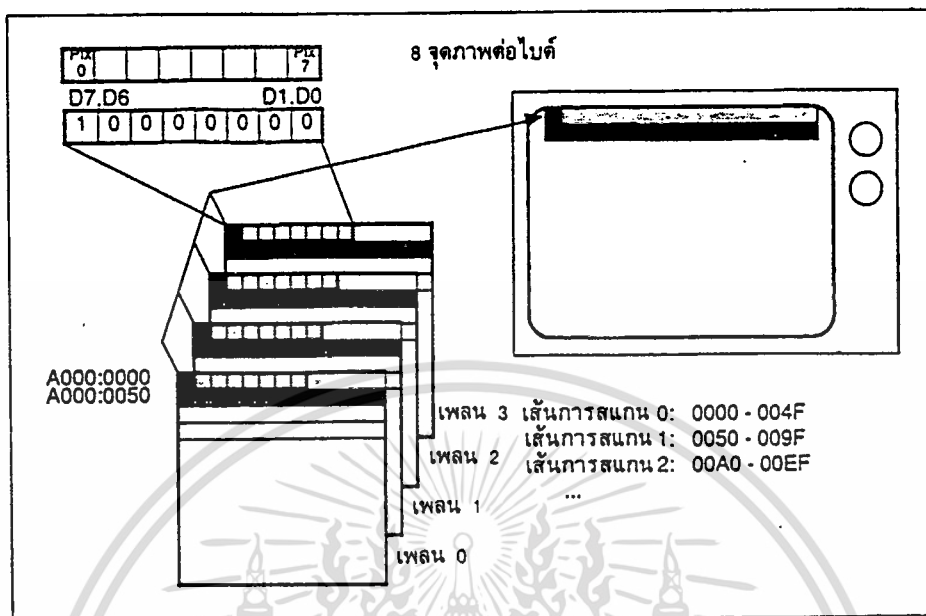
2.6.2 หน่วยความจำแสดงผลโหมด 10 H (กราฟิกสี Enhanced)

มีซอฟต์แวร์จำนวนมากที่ทำงานบนโหมดความละเอียดของจอภาพมีขนาด 640 x 350 จุด ข้อมูลที่แทนจุดบนจอภาพอยู่เฟลนสีทั้ง 4 เฟลน หนึ่งบิตในแต่ละเฟลนแทนจุดหนึ่ง ซึ่งสีของจุดจะถูกควบคุมโดยข้อมูลแต่ละบิตทั้ง 4 เฟลน สีที่ตรงกัน ดังนั้น จะสามารถแสดงสีได้ 16 สีในรูปที่ 2.29 แสดงหน่วยความจำบนเฟลนสีทั้งที่สัมพันธ์กับจุดบนจอภาพ การหาตำแหน่งของหน่วยความจำจากจุดบนจอภาพทำได้จาก

$$\begin{aligned}\text{ตำแหน่งไบต์} &= Y * 80 + X / 8 \\ \text{ตำแหน่งบิต} &= 7 - (X \text{ modulo } 8)\end{aligned}$$

2.6.3 หน่วยความจำแสดงผลโหมด D และ E (กราฟิก 16 สี)

โหมด D และ E จะคล้ายกับโหมด 10 H ต่างกันที่จำนวนจุด หรือความละเอียดของจอภาพ คือ โหมด D มีความละเอียด 320 x 200 จุด ส่วนโหมด E มีความละเอียด 640 x 200 ทั้งสองโหมดมีซอฟต์แวร์ที่ทำงานบนทั้งสองโหมดน้อย เนื่องจากความละเอียดน้อยกว่าโหมด 10 ค่าความคมชัดของรูปจึงน้อยกว่าโหมด 10 H



รูปที่ 2.29 การจัดเรียงหน่วยความจำโหมดกราฟิก D, E, 10H, 12H

2.6.4 หน่วยความจำแสดงผลโหมด 11 (กราฟิก 2 สี)

โหมดนี้มีอยู่บน VGA เท่านั้นโดยมีความละเอียดเป็น 640x480 จุด มีความสามารถแสดงสีได้เพียง 2 สี ข้อมูลบนหน่วยความจำ 1 บิตจะแทน 1 จุดบนจอภาพ และถูกเก็บอยู่บนเฟลน 0 เท่านั้น ดังรูปที่ 2.30 การหาตำแหน่งของหน่วยความจำหาได้จากสูตร

$$\text{ตำแหน่งไบต์} = (Y \times 80) + (X/8)$$

$$\text{ตำแหน่งบิต} = 7 - (X \bmod 8)$$

2.6.5 หน่วยความจำแสดงผลโหมด 12 H (กราฟิก 16 สี)

โหมด 12H มีลักษณะคล้ายโหมด 10H ต่างกันที่ความละเอียดในแนวตั้งจะเป็น 480 จุด แทน 350 จุด ข้อมูลจะเก็บลงทั้ง 4 เฟลน เหมือนโหมด 10H แสดงสีได้ 16 สีพร้อมกัน การจัดเรียงหน่วยความจำคล้ายรูป 2.29

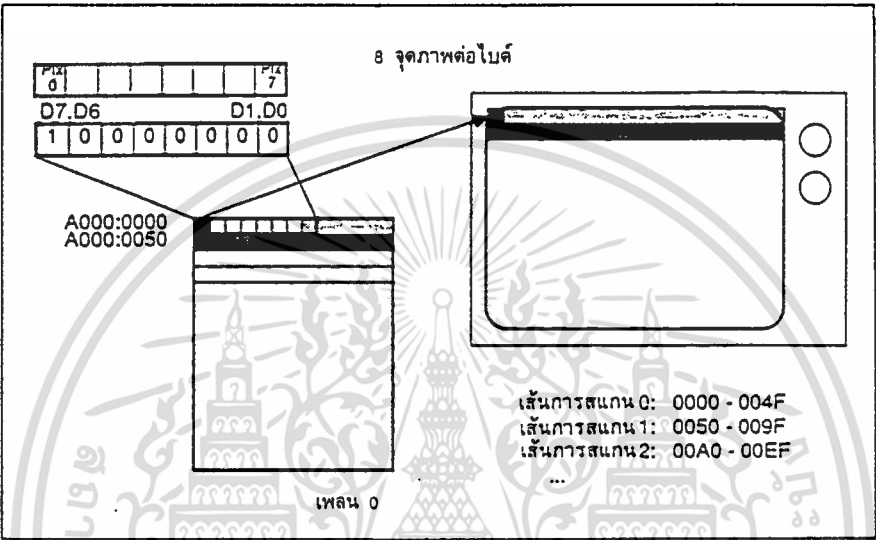
2.6.6 หน่วยความจำแสดงผลโหมด 13 H (กราฟิก 256 สี)

ในโหมด 13 H นี้สามารถแสดงสีได้ถึง 256 สีพร้อมกัน แต่ความละเอียดจะต่ำคือ 320 x 200 จุด การจัดเรียงหน่วยความจำอยู่ในรูปที่ 2.31 หนึ่งจุดบนจอภาพจะแทนด้วยข้อมูล

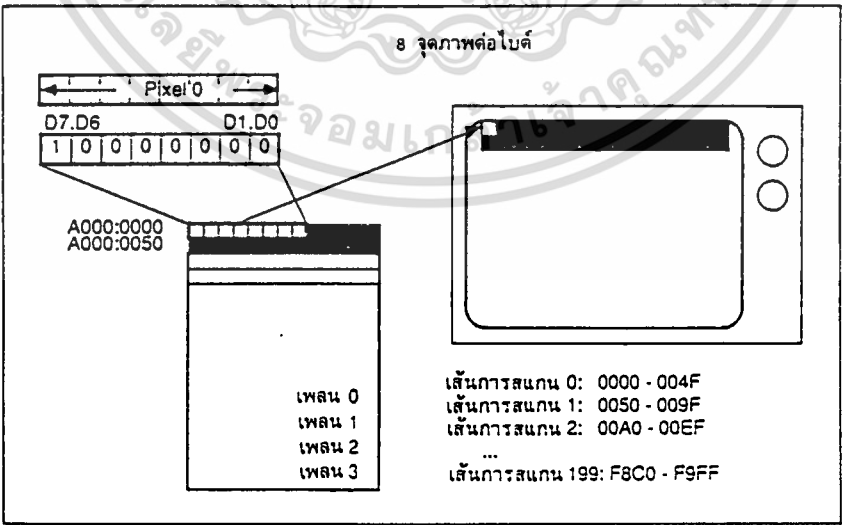
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาด 1 ไบต์ ซึ่งจัดเก็บอยู่บนทั้ง 4 เฟลน การหาตำแหน่งของหน่วยความจำจากจุดบนจอภาพได้จากสูตร

ตำแหน่งไบต์ = (Y*320) + X



รูปที่ 2.30 การจัดเรียงหน่วยความจำโหมดกราฟิก 11 H



รูปที่ 2.31 การจัดเรียงหน่วยความจำโหมดกราฟิก 13 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบฟอนต์

3.1 ความรู้ทั่วไปเกี่ยวกับฟอนต์

ในปัจจุบันการสร้างโปรแกรมที่ทำงานในโหมดกราฟิกเป็นที่ยอมรับกันมากเพราะในโหมดกราฟิกสามารถลดข้อจำกัดในหลาย ๆ ประการของการทำงานในโหมดข้อความ (text mode) ได้ แต่สิ่งหนึ่งที่เป็นพื้นฐานของโปรแกรมก็คือ การแสดงข้อความเพื่อติดต่อระหว่างโปรแกรมกับผู้ใช้ ดังนั้น ตัวอักษรจึงเป็นสิ่งจำเป็นพื้นฐานของโปรแกรมที่ทำงานในโหมดกราฟิกและในปัจจุบันชนิดของฟอนต์ (font type) ก็มีอยู่ด้วยกันมากมาย โครงการนี้จึงได้เสนอแนวความคิดออกแบบโปรแกรมเพื่อใช้ในการช่วยออกแบบตัวอักษรลักษณะต่างๆ โดยเขียนโปรแกรมขึ้นจากภาษาซี ฟอนต์โดยลักษณะใหญ่ ๆ แล้วแบ่งเป็นสองประเภทคือชนิดที่สามารถขยายได้กับที่ไม่สามารถขยายได้ โครงการนี้เป็นโปรแกรมช่วยออกแบบฟอนต์ชนิดบิตแมป ซึ่งฟอนต์แบบนี้ไม่สามารถขยายได้ ก่อนหน้าที่จะเข้าใจถึงการออกแบบฟอนต์ เราควรเข้าใจถึงคำจำกัดความของฟอนต์, การแบ่งฟอนต์ตามลักษณะและชนิดต่าง ๆ รวมถึงการบอกขนาดของฟอนต์ ซึ่งจะช่วยให้เราเข้าใจฟอนต์ได้ดียิ่งขึ้น

3.1.1 คำจำกัดความทั่วไปของฟอนต์

ฟอนต์เป็นเทคโนโลยีใหม่ของคอมพิวเตอร์เพราะในปัจจุบันโปรแกรมระดับแนวหน้า เช่น Windows, PageMaker เป็นต้นเป็นโปรแกรมที่ทำงานในโหมดกราฟิก ดังนั้นฟอนต์จึงจำเป็นมากสำหรับโปรแกรกดังกล่าว ในปัจจุบันจึงได้มีการพัฒนาฟอนต์ขึ้นมาหลาย ๆ รูปแบบโดยหลาย ๆ บริษัทผู้ผลิตซอฟต์แวร์ดังนั้นคำจำกัดความต่าง ๆ สำหรับใช้ในการเรียกฟอนต์จึงเกิดขึ้นและเพื่อความเข้าใจ ในหัวข้อนี้จึงขอแสดงคำอธิบายของคำต่าง ๆ ดังนี้

Font ถ้าหากมีการใช้เพียงลำพังโดยไม่มีการนำไปขยายคำใด ๆ จะหมายถึง **Font Character** , **Font Style** , **Font Format**, **Font Data** โดยทั่วไปคำว่า **Font** มักจะถูกนำไปใช้ร่วมกับคำอื่น ๆ เพื่อแสดงความหมายที่เฉพาะอย่าง ตามตัวอย่างข้างต้น

Font Set หมายถึง กลุ่มของตัวอักษรที่ได้รับการออกแบบและถูกจัดเก็บเอาไว้ในไฟล์เดียวกัน เช่น ผู้เขียนออกแบบตัวอักษร Roman ซึ่งประกอบด้วยกลุ่มของตัวอักษรต่าง ๆ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

!"#\$%&'()*+,-./0123456789:;<=>?@
 ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]Z~
 abcdefghijklmnopqrstuvwxyz{|}~

สำหรับตัวอย่างของ FONT SET สามารถดูได้ในรูปที่ 3.1 แสดง FONT SET ต่างๆ

Font Character หมายถึง ตัวอักษรหรือตัวอักขระตัวใดตัวหนึ่งใน Font Set เช่น A,B,C,1,{ เป็นต้น

Font Style หมายถึง ตัวอักษรที่ออกแบบ เช่น Helvetica, Modern, Times, Roman เป็นต้น

Roman

!"#\$%&'()*x+,-./0123456789: ; < = >
 @ABCDEFGHIJKLMN O P Q R S T U V W X
 Y Z [\] ^ _ ' ` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~

Modern

!"#\$%&'()*x+,-./0123456789: ; < = >
 @ABCDEFGHIJKLMN O P Q R S T U V W
 X Y Z [/] ^ _ ' ` a b c d e f g h i j k l m n o p q r s t u v w x y z { | } ~

Angsana UPC

!"#\$%&'()*x +,- / 0123456789:;<=>@

ABCDEFGHIJKLMN O P Q R S T U V W

X Y Z [/] ^ _ ' ` a b c d e f g h i j k l m n o p q r s t u v w x y z

{ | } ~

รูปที่ 3.1 แสดง Font Set ต่างๆ

Font Families หมายถึงตระกูลของฟอนต์ โดยที่ฟอนต์นี้จะรักษาลักษณะที่เป็นเอกลักษณ์ของตนเองไว้ในแต่ละตระกูล เช่น Regular Font , Bold Font , Italic Font , Bold Italic

Font เป็นต้น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Font Data หมายถึง ข้อมูลของแต่ละตัวอักษรที่ออกแบบของฟอนต์แต่ละชนิด โดยทั่ว ๆ ไปจะมีความหมายถึงบิตหรือไบต์ (bit or byte)

Font Header หมายถึง ข้อมูลส่วนหัวของไฟล์เก็บตัวอักษร โดยที่ข้อมูลในส่วนนี้จะใช้ในการกำหนดหรือระบุรูปแบบหรือวิธีการจัดเก็บของตัวอักษรในไฟล์ เช่น เก็บชื่อของฟอนต์ ขนาดของตัวอักษรเป็น ดัน

Font File หมายถึง ไฟล์ที่จัดเก็บข้อมูล (font header, font data) ของทุก ๆ ตัวอักษรที่ออกแบบ ในกลุ่มเดียวกัน

Font Format หมายถึง ลักษณะหรือวิธีที่โปรแกรมเมอร์แต่ละคนจัดเก็บข้อมูลของทุก ๆ ตัวอักษรในไฟล์ ซึ่งฟอนต์ต่างประเภทกันจะมีวิธีการจัดเก็บข้อมูลของตัวอักษรที่ออกแบบลงไฟล์ต่างกัน

Font Name หมายถึง ชื่อของแฟ้มที่ออกแบบโดยที่ font style เดียวกันเราอาจกำหนดให้มี font name ที่ต่างๆ กันได้ เพื่อที่จะใช้แยกฟอนต์ที่เราออกแบบให้ต่างไปจาก font style เดียวกัน แต่ออกแบบโดยที่อื่นๆ เช่น Helve86, Roman12 เป็นต้น

3.1.2 ฟอนต์ที่แบ่งตามสื่อจัดเก็บ

ในปัจจุบันฟอนต์จะถูกบรรจุในสื่อต่าง ๆ เพื่อเสนอให้กับผู้ใช้คอมพิวเตอร์ในลักษณะที่ต่างกัน และมาตรฐานโดยทั่ว ๆ ไปสามารถแบ่งฟอนต์ตามชนิดของสื่อ (media) ที่จัดเก็บได้ 3 ประเภทใหญ่ ๆ ดังนี้

Font Diskette หมายถึง ฟอนต์ที่จัดเก็บในแผ่นดิสก์ ซึ่งสามารถเป็นได้ทั้งแผ่นขนาด 5 1/4 นิ้ว หรือ 3 1/2 นิ้ว ก็ได้หรือเก็บไว้ในหน่วยความจำของคอมพิวเตอร์ที่เรียกว่า ฮาร์ดดิสก์ก็ได้ ในการนำเสนอฟอนต์ในรูปแบบนี้มักพบกันมากในปัจจุบัน เช่น Adobe Type Manager เป็นต้น เพราะวิธีนี้เป็นวิธีที่สะดวกและราคาค่อนข้างถูกสามารถนำฟอนต์เข้ามาแก้ไขใหม่หรือให้มีประสิทธิภาพเพิ่มขึ้น (version upgrade) ได้ง่ายกว่าฟอนต์ที่เก็บในสื่อชนิดอื่น ๆ และฟอนต์ที่จัดเก็บในสื่อประเภทนี้ เรามักเรียกชื่อกันว่า soft font

Font Cartridge หมายถึง ฟอนต์ที่จัดเก็บในตลับซึ่งใช้เสียบลงในช่อง เช่น ฟอนต์สำหรับเครื่องพิมพ์ เป็นต้น

RAM/ROM/EPROM Card หมายถึง ฟอนต์ที่มีในเครื่องรุ่นใหม่ ๆ เช่นเครื่องพิมพ์ชนิดเลเซอร์ มักนิยมใช้วิธีจัดเก็บฟอนต์ชนิดนี้ โดยที่ขนาดของหน่วยความจำโดยทั่วไปจะเท่ากับ 62,128 หรือ 256 กิโลไบต์ เป็นต้น

3.1.3 รูปแบบของฟอนต์

ในความหมายทั่ว ๆ ไปรูปแบบของฟอนต์ (font format) สามารถแบ่งได้เป็น 2 ชนิด ดังนี้

1. Graphic Font

เป็นฟอนต์แบบกราฟิก หมายถึง ฟอนต์ที่ถูกออกแบบเพื่อให้สามารถจัดเก็บในสื่อประเภท แผ่นดิสก์ ฮาร์ดดิสก์หรือตลับสียิป (cartidge) หรือที่เรียกกันว่า soft media ดังนั้น ฟอนต์ชนิดนี้จะไม่ขึ้นกับฮาร์ดแวร์และจะถูกเรียกใช้โดยซอฟต์แวร์หรือ โปรแกรมที่มาพร้อมกับฟอนต์แต่ละประเภท

2. Text Font

เป็นฟอนต์แบบข้อความ หมายถึง ฟอนต์ที่ถูกออกแบบเพื่อให้สามารถจัดเก็บในสื่อ ประเภท ROM และฟอนต์ชนิดนี้จะขึ้นกับฮาร์ดแวร์จะถูกเรียกใช้โดยโปรแกรมที่มาพร้อมกับฮาร์ดแวร์ของเครื่องคอมพิวเตอร์ เช่น ฟอนต์ที่มาพร้อมกับเครื่องคอมพิวเตอร์ เป็นต้น แต่ถ้าเราจะแบ่งรูปแบบของฟอนต์ตามวิธีที่ใช้สร้างตัวอักษร ก็สามารถแบ่งได้เป็น 4 ชนิดดังนี้

1. **Vector or Stroke Font** หมายถึง ฟอนต์แบบเวกเตอร์หรือสโตรก เป็นการออกแบบตัวอักษรในโหมดกราฟิกโดยอาศัยหลักการของการเชื่อมโยงเส้นตรงระหว่างจุดสองจุดที่อยู่ติดกันซึ่งตัวอักษรในลักษณะนี้ การเปลี่ยนขนาดของตัวอักษรสามารถทำได้ง่ายโดยไม่มีความจำเป็นต้องสร้างตัวอักษรไว้ทุกขนาด เพราะในการเปลี่ยนแปลงขนาดของตัวอักษรก็สามารถทำได้ง่ายโดยอาศัยสมการทางคณิตศาสตร์ช่วย

2. **Outline Font** เป็นฟอนต์ที่สร้างโดยอาศัยสมการคณิตศาสตร์ คล้ายกับฟอนต์ชนิดแรก คือ มีการรวมข้อดีของฟอนต์แบบบิตแมปและเวกเตอร์เข้าด้วยกันนั่นคือจะเก็บโครงร่างของฟอนต์ไว้ในรูปของเวกเตอร์แทนที่จะเก็บไว้ในรูปของจุด เมื่อมีการนำมาแสดงผลก็เพียงแต่เติมจุดสีเข้าไปในพื้นที่ภายในตัวอักษรเท่านั้น ดังนั้นฟอนต์ชนิดนี้จึงสามารถขยายหรือลดขนาดได้ (scaling) โดยที่คุณภาพตัวอักษรยังคงเดิม

3. **Bitmap Font** หมายถึง ฟอนต์ที่มีการจัดเก็บข้อมูลของตัวอักษรแต่ละตัวในรูปแบบของจุด (dots หรือ pixel) ดังนั้น ฟอนต์ชนิดนี้จึงมีขนาดคงที่ไม่สามารถขยายหรือลดขนาดลงได้แต่ฟอนต์ชนิดนี้ก็สามารถเพิ่มขนาด (expand) ได้ แต่ถ้าหากเพิ่มขนาดมากเกินไปก็จะทำให้ตัวอักษรดูไม่สวยงาม

4. **Text Font** ฟอนต์ชนิดนี้เป็นฟอนต์แบบบิตแมป (bitmap) ชนิดพิเศษโดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นด้านการค้าจะมีขนาดคงที่ เช่น ฟอนต์ที่เราเห็นบนจอของเครื่องคอมพิวเตอร์ในตอนเริ่มเปิดเครื่อง เป็นต้น

3.1.4 การบอกขนาดของฟอนต์

ในการบอกขนาดของฟอนต์มีอยู่ด้วยกันหลายวิธี และมีหน่วยในการวัดขนาดของฟอนต์มากมายดังนี้คือ

- **Point Size** บริษัทผู้ผลิตฟอนต์โดยทั่วไปจะบอกขนาดของฟอนต์ในหน่วยของพอยต์ (point) โดยสามารถวัดค่าออกมาในหน่วยการวัดสากลดังนี้ 1 พอยต์ = $1/72$ หรือ 0.1389 นิ้ว สำหรับวิธีที่ใช้วัดขนาดของตัวอักษร มีอยู่ 2 วิธีคือ บอกขนาดของฟอนต์โดยใช้ความสูงของกรอบที่ให้ออกแบบฟอนต์ และอีกวิธีคือ บอกขนาดของฟอนต์โดยใช้ความสูงของอักษรพิมพ์ใหญ่เท่านั้นดังในรูป 3.2



รูปที่ 3.2 แสดงถึงการบอกขนาดของฟอนต์

- **Inch Size** เป็นการบอกขนาดของฟอนต์โดยใช้หน่วยการวัดเป็นนิ้วในโปรแกรมช่วย ช่วยการพิมพ์ (page layout) เช่น PageMaker หรือ Ventura จะมีการบอกขนาดของฟอนต์ทั้ง ในหน่วยของ พอยต์ นิ้ว และ เซนติเมตร โดยในการแปลงหน่วยไปมาระหว่างกันก็สามารถทำได้โดยใช้สูตรข้างต้น

- **Pixel Size** การบอกขนาดของฟอนต์โดยใช้หน่วยนี้จะป็นวิธีที่บอกเพื่อให้ตรงกับรายละเอียดของจอภาพ เพราะจอภาพจะมีหน่วยแสดงผลที่เล็กที่สุดเรียกว่าจุด และสำหรับจอภาพโดยมาตรฐานจะมีขนาดเท่ากับ 11 x 8 1/2 นิ้ว (ฐาน x สูง) แต่สำหรับรายละเอียดของจอภาพแต่ละชนิดมีความแตกต่างกัน ดังนี้

จอภาพ	ความละเอียด
CGA	320 x 200, 640 x 200
Hercules	720 x 348
EGA	640 x 350
VGA	640 x 480
SuperVGA	800 x 600, 1024 x 768

3.2 การออกแบบฟอนต์

ในปัจจุบันเทคนิคการออกแบบตัวอักษร (font design technique) มีอยู่ด้วยกันหลายวิธี แต่สำหรับโครงการนี้ผู้เขียนขอเสนอเทคนิคการสร้างตัวอักษรแบบบิตแมป (bitmap font) ซึ่งเป็นเทคนิคการสร้างตัวอักษรที่ง่ายและตรงไปตรงมาที่สุด โดยจะอาศัยหลักการของการแสดงผลของกราฟิกแบบจุดต่อจุด ดังนั้นถ้าหากผู้อ่านต้องการตัวอักษรที่มีขนาดต่างไปจากเดิมผู้อ่านจะต้องทำการสร้างแบบของตัว อักษรขึ้นมาใหม่

3.2.1 การออกแบบฟอนต์แบบบิตแมป

สำหรับตัวอักษรอังกฤษชุดของตัวอักษรประกอบด้วย

1. พยัญชนะ และสระ (A-Z, a-z)
2. ตัวเลข (0-9)
3. เครื่องหมายต่าง ๆ เช่น (!, #, %)

ในการออกแบบตัวอักษรจะทำการออกแบบในกรอบสี่เหลี่ยม ซึ่งมาตรฐานการออกแบบตัวอักษรโดยทั่วไปจะมีลักษณะเป็นกรอบสี่เหลี่ยมจะมีรูปแบบดังในรูปที่ 3.3 โดยที่ระยะระหว่างเส้นต่าง ๆ จะมีความหมายดังนี้

ความกว้างของฟิมพ์ (Cell Width) หมายถึง ความกว้างสูงสุดของตัวอักษรที่ออกแบบซึ่งเป็นความกว้างของตัวอักษรรวมกับระยะช่องไฟระหว่างตัวอักษร ซึ่งเป็นระยะตั้งแต่เส้นขอบหน้าถึงเส้นขอบหลัง

ความสูงของฟิมพ์ (Cell Height) หมายถึง ความสูงสูงสุดของตัวอักษร ที่ออกแบบซึ่งเป็นระยะตั้งแต่เส้นขอบล่างถึงเส้นขอบบน

ความกว้างของตัวอักษร (Character Width) หมายถึง ความกว้างสูงสุดของตัวอักษรที่ออกแบบ ซึ่งเป็นระยะตั้งแต่เส้นก้นหน้าถึงเส้นก้นหลัง

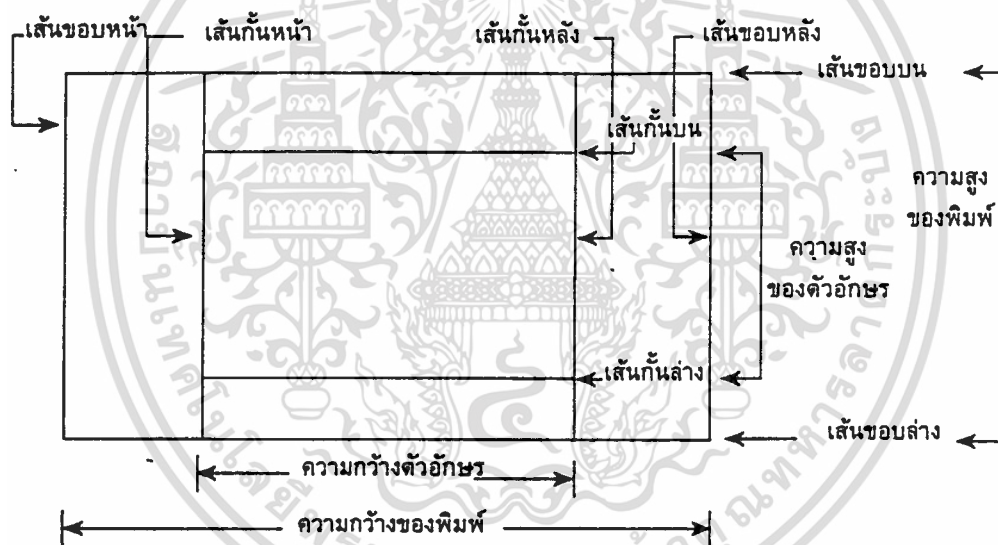
ความสูงของตัวอักษร (Character Height) หมายถึง ระยะห่างตั้งแต่เส้นก้นล่างถึงเส้นก้นบน

ช่องไฟระหว่างตัวอักษร (Proportional) หมายถึง ระยะห่างของแต่ละตัวอักษรที่อยู่ติดกัน ซึ่งเป็นระยะตั้งแต่เส้นขอบหน้าถึงเส้นก้นหน้า และระยะตั้งแต่เส้นก้นหน้าถึงเส้นขอบหลัง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่องไฟระหว่างบรรทัด (Line Gap) ระยะห่างของบรรทัดที่อยู่ติดกัน ซึ่งเป็นระยะตั้งแต่เส้นกั้นบนถึงเส้นขอบบน และตั้งแต่เส้นกั้นล่างถึงเส้นขอบล่าง

เนื่องจากตัวอักษรีมีระดับการวางตัวอักษรที่แตกต่างกัน เช่น อักษร A, g, a, h จะเห็นว่าระดับของหาง (descend) ของตัวอักษรเล็ก จะอยู่ในระดับต่ำลงมาจากระดับฐานของตัวอักษรพิมพ์ใหญ่ (A-Z) และความสูงของตัวอักษรพิมพ์เล็กจะมีค่าน้อยกว่าความสูงของตัวอักษรพิมพ์ใหญ่และ ตัวเลข ดังนั้นจากรูป 3.3 เพื่อให้สามารถนำมาใช้ออกแบบตัวอักษรอังกฤษเราจะเห็นว่าการวางตัวอักษรจะต่างไปจากตัวอักษรพิมพ์ใหญ่



รูปที่ 3.3 แสดงกรอบและเส้นกำกับของตัวอักษร

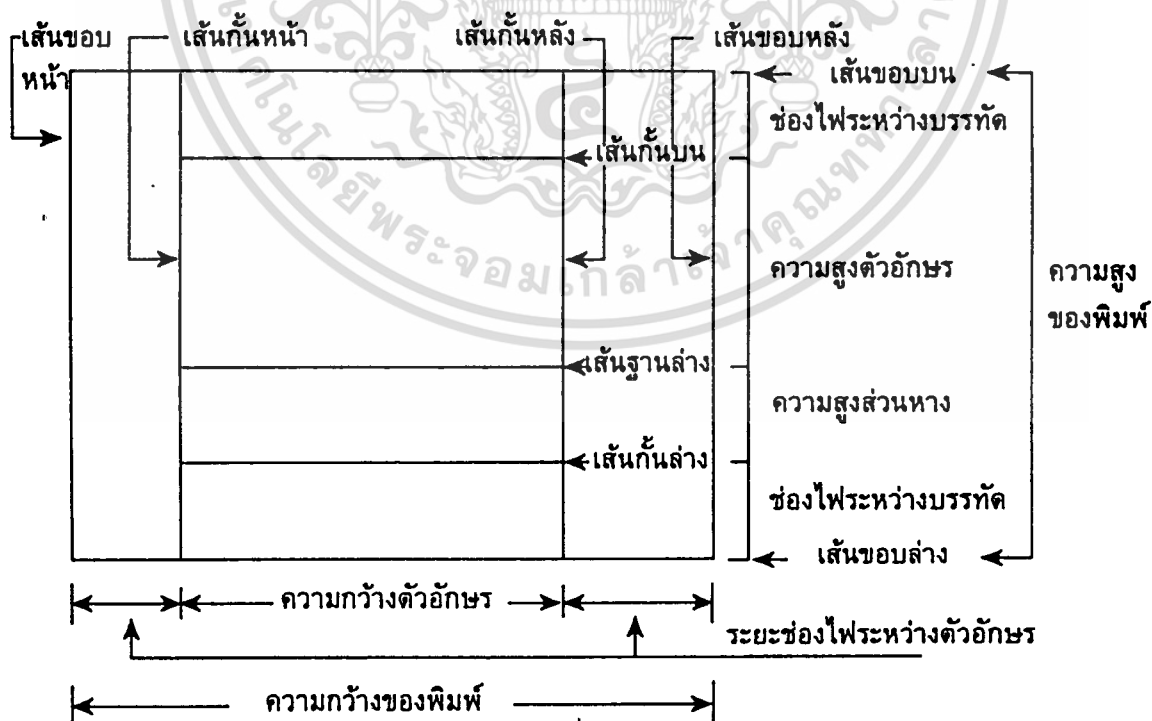
สำหรับเส้นกำกับที่ใช้ในการออกแบบตัวอักษรดังในรูป 3.3 แต่ละเส้นมีความหมายดังนี้

เส้นขอบหน้า หมายถึง เส้นที่ใช้บอกขอบเขตของส่วนหน้าสุดของพิมพ์หรือกรอบออกแบบตัวอักษร

เส้นกั้นหน้า หมายถึง เส้นที่ใช้บอกขอบเขตของส่วนหน้าสุดของตัวอักษรที่ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นกันหลัง	หมายถึง	เส้นที่ใช้บอกขอบเขตของส่วนหลังสุดของตัวอักษรที่ ออกแบบ
เส้นขอบหลัง	หมายถึง	เส้นที่ใช้บอกขอบเขตของส่วนหลังสุดของพิมพ์หรือ กรอบออกแบบตัวอักษร
เส้นขอบบน	หมายถึง	เส้นที่ใช้บอกขอบเขตของส่วนบนสุดของพิมพ์หรือ กรอบออกแบบตัวอักษร
เส้นกันบน	หมายถึง	เส้นที่ใช้บอกขอบเขตของส่วนบนสุดของตัวอักษรที่ ออกแบบ
เส้นกันล่าง	หมายถึง	เส้นที่ใช้บอกขอบเขตของส่วนล่างสุดของตัวอักษรที่ ออกแบบ
เส้นขอบล่าง	หมายถึง	เส้นที่ใช้บอกขอบเขตของส่วนล่างสุดของพิมพ์หรือ กรอบออกแบบตัวอักษร
เส้นฐานล่าง	หมายถึง	เส้นที่ใช้กำหนดระดับแนวฐานล่างของอักษรพิมพ์ใหญ่



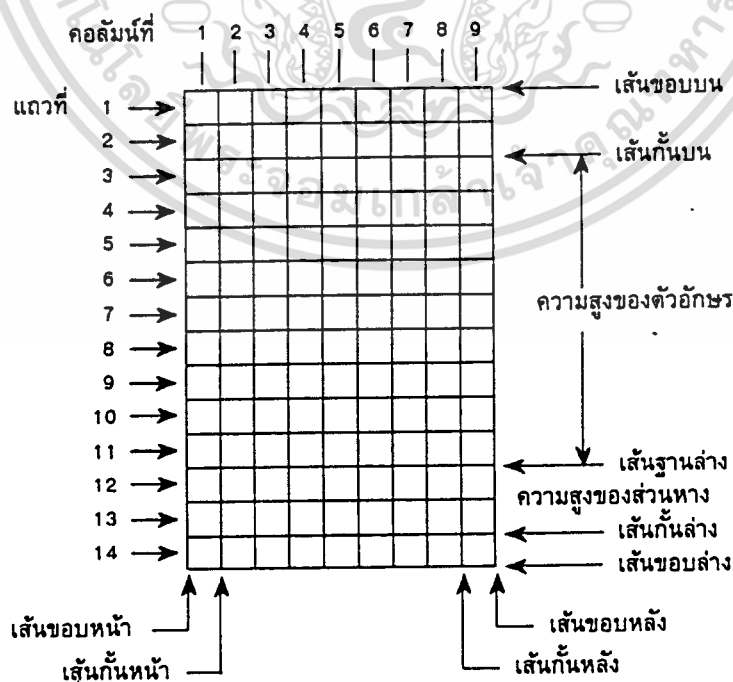
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเนื้อหาที่ละเอียดและถูกต้อง ซึ่งอาจมีการแก้ไขปรับปรุงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 แสดงเส้นกำกับระดับของตัวอักษรอังกฤษ

หลังจากที่ได้กรอบสี่เหลี่ยมสำหรับออกแบบตัวอักษรอังกฤษแล้ว เนื่องจากตัวอักษรที่ออกแบบผู้เขียนได้เลือกใช้วิธีการแสดงผลตัวอักษรแบบบิตแมป ดังนั้นจึงต้องทำการนำกรอบที่ได้ในรูปที่ 2.2 มาแบ่งออกเป็นช่องเล็ก ๆ ซึ่งเรียกว่า pixel หรือ grid โดยที่หนึ่งช่องเท่ากับหนึ่งจุดของการแสดงผล(pixel) ข้อตกลงสำหรับการบอกขนาดของฟอนต์สำหรับหนังสือเล่มนี้ จะบอกขนาดของกว้างคูณสูง (XxY) โดยมีหน่วยเป็นจุดดังเช่น ถ้าหากต้องการออกแบบตัวอักษรขนาด 9x14 จุด เราก็นำกรอบสี่เหลี่ยมในรูปที่ 3.4 มาทำการแบ่งออกเป็นช่อง ๆ โดยให้มีจำนวนช่องเท่ากับขนาดของตัวอักษรที่ต้องการออกแบบ ดังรูปที่ 3.5

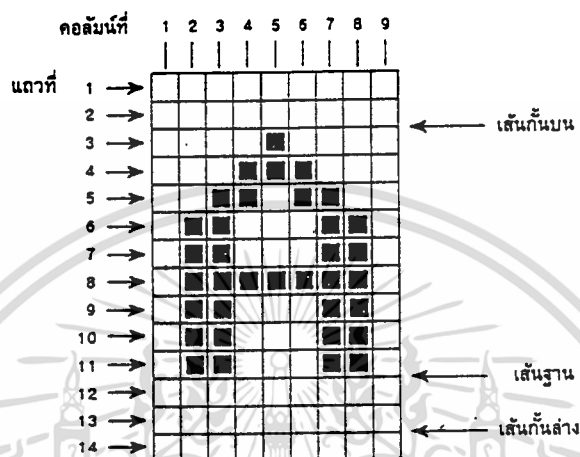
สำหรับความกว้างของตัวอักษรแต่ละตัวสามารถปรับให้มีค่าแตกต่างกันได้ เพราะตัวอักษรอังกฤษบางตัวจะมีความกว้างที่แตกต่างกัน เช่น ตัว A จะกว้างกว่าตัว I เป็นต้น แต่สำหรับตัวอย่างนี้ผู้เขียนกำหนดให้ความสูงของตัวอักษรเท่ากับ 9 จุด และความสูงของส่วนหางเท่ากับ 2 จุด

ทั้งนี้การแก้ไขระยะต่าง ๆ สามารถทำได้โดยให้ขึ้นกับความพอใจในการออกแบบของผู้อ่าน เพื่อให้ได้แบบของตัวอักษรที่ดีที่สุด หลังจากที่ได้ตารางในรูปที่ 3.5 ก็ให้ผู้อ่านทำการออกแบบตัวอักษร สมมติว่าผู้เขียนต้องการออกแบบตัวอักษร A ก็จะได้ตัวอย่างของอักษร A ดังที่ปรากฏในรูปที่ 3.6 โดยระบายในช่องที่ต้องการแสดงจุด และไม่ระบายในช่องที่ไม่ต้องการแสดงจุด



รูปที่ 3.5 แสดงตารางออกแบบตัวอักษรขนาด 9x14

เราสามารถใช้เลขฐานสองแทนความหมายโดยการเปลี่ยนจากจุดในตารางให้อยู่ในรูปแบบของเลขฐานสอง (0 และ 1) ดังในตารางที่รูป 3.7



รูปที่ 3.6 แสดงแบบของตัวอักษร

COL	1	2	3	4	5	6	7	8	9
R1									
R2									
R3					1				
R4				1	1	1			
R5			1				1		
R6		1	1				1	1	
R7		1	1				1	1	
R8		1	1	1	1	1	1	1	
R9		1	1				1	1	
R10		1	1				1	1	
R11		1	1				1	1	
R12									
R13									
R14									

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.7 แสดงตารางเลขฐานสองแทนข้อมูลของตัวอักษร A
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การเก็บฟอนต์โดยใช้รหัสแอสกี

จากรูปตารางที่ 3.7 ถ้าเราใช้เลขจำนวนเต็ม integer ในการแทนเลข 0 และ 1 จึงเห็นว่าเป็นการไม่เหมาะสมนักเพราะข้อมูลแต่ละตัวจะใช้เนื้อที่ในการเก็บ 2 ไบต์ อักษร (A) จะใช้เนื้อที่ในการเก็บเท่ากับ $9 \times 14 \times 2 = 252$ ไบต์ และถ้าเราออกแบบตัวอักษรทั้ง 256 ตัวจะต้องใช้เนื้อที่ในการเก็บมากซึ่งจะเป็นการไม่ประหยัดและไม่นิยม จึงควรมีวิธีที่ดีกว่านี้ใน โครงการนี้การจัดเก็บข้อมูลตัวอักษรจึงแทนข้อมูลแต่ละตัว (0 หรือ 1) ให้เป็นบิตบิตหนึ่งในเลขฐานสอง โดยข้อมูล 8 ตัว จะใช้เนื้อที่ในการเก็บเพียง 1 ไบต์ (ข้อมูลชนิดสตริง) เท่านั้น

แต่สำหรับวิธีการแปลงข้อมูลชนิดเลขฐานสองข้างต้นให้เป็นรหัสแอสกีผู้เขียนขอเสนอ 2 วิธีเพื่อเป็นแนวทางสำหรับผู้อ่านดังนี้

1. Row major conversion
2. Column major conversion

1. วิธี Row major conversion

สำหรับวิธีนี้หมายถึง ในการแปลงข้อมูลชนิดเลขฐานสอง 8 บิต ให้เป็นรหัสแอสกี 1 ตัว จะเริ่มนับจากแถวที่ 1, คอลัมน์ที่ 1, 2, 3, ..., 9 และแถวที่ 2 คอลัมน์ที่ 1, 2, 3, ..., 9 และแถวที่ 3, 4, ..., 14 ตามลำดับ ดังนั้นในแต่ละแถวจะสามารถแปลงให้อยู่ในรูปแบบของรหัสแอสกีขนาด 2 ไบต์ และสำหรับตัวอักษร 1 ตัว จะต้องใช้เนื้อที่ในการเก็บขนาด $2 \times 14 = 28$ ไบต์ จากรูปตารางที่ 3.7 เราสามารถนำมาแปลงให้อยู่ในรูปแบบรหัสแอสกีได้ ดังรูปตารางที่ 3.9 สำหรับข้อดีของวิธีนี้คือ ในการแสดงผลตัวอักษรออกทางจอภาพสามารถทำได้เร็ว สำหรับกรณีที่ต้องการแสดงผลแต่ละจุด สำหรับโครงการนี้เลือกใช้วิธีที่ 1 Row major conversion ในการเก็บตัวอักษร

วิธีการแสดงผลในโหมดกราฟิกสามารถทำได้ 2 ลักษณะ ดังนี้

1. แสดงผลโดยใช้บริการจากไบออสหมายเลข 10 H ฟังก์ชันหมายเลข 0CH (Write Graphics Pixel) สำหรับวิธีนี้ สามารถทำได้ง่ายเพราะสำหรับจอภาพทุก ๆ ชนิด เราสามารถใช้ฟังก์ชันหมายเลข 0CH แสดงผลได้ทันที แต่การแสดงผลจะค่อนข้างช้ากว่าวิธีที่ 2

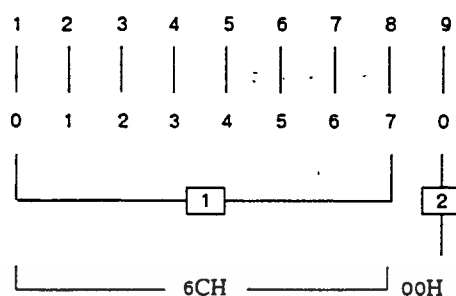
2. แสดงผลโดยการเข้าถึงหน่วยความจำสำหรับการแสดงผลออกจากจอภาพ โดยตรงสำหรับวิธีหลัง สามารถทำได้ยากกว่าวิธีแรกมาก เพราะสำหรับจอภาพแต่ละชนิดจะอ้างตำแหน่งของหน่วยความจำที่แตกต่างกัน และจำนวนบิตที่ใช้สำหรับแต่ละจุดของจอภาพก็แตกต่างกันสำหรับแต่ละโหมดของกราฟิก แต่วิธีนี้การแสดงผลทางจอภาพเร็วกว่าวิธีแรก

ตัวอย่างการแปลงแถวที่ 5 คอลัมน์ที่-

บิตที่

ไบต์ที่

รหัสแอสกีในรูปแบบเลขฐานสิบหก



รูปที่ 3.8

คอลัมน์ที่	1	2	3	4	5	6	7	8	9	ไบต์ที่	ไบต์ที่
แถวที่ 1	0	0	0	0	0	0	0	0	0	1 00H	2 00H
แถวที่ 2	0	0	0	0	0	0	0	0	0	3 00H	4 00H
แถวที่ 3	0	0	0	0	1	0	0	0	0	5 10H	6 00H
แถวที่ 4	0	0	0	1	1	1	0	0	0	7 38H	8 00H
แถวที่ 5	0	0	1	1	0	1	1	0	0	9 6CH	10 00H
แถวที่ 6	0	1	1	0	0	0	1	1	0	11 C6H	12 00H
แถวที่ 7	0	1	1	0	0	0	1	1	0	13 C6H	14 00H
แถวที่ 8	0	1	1	1	1	1	1	1	0	15 FEH	16 00H
แถวที่ 9	0	1	1	0	0	0	1	1	0	17 C6H	18 00H
แถวที่ 10	0	1	1	0	0	0	1	1	0	19 C6H	20 00H
แถวที่ 11	0	1	1	0	0	0	1	1	0	21 C6H	22 00H
แถวที่ 12	0	0	0	0	0	0	0	0	0	23 00H	24 00H
แถวที่ 13	0	0	0	0	0	0	0	0	0	25 00H	26 00H
แถวที่ 14	0	0	0	0	0	0	0	0	0	27 00H	28 00H

รูปที่ 3.9 ตารางแสดงวิธี Row major conversion

2. วิธี Column major conversion

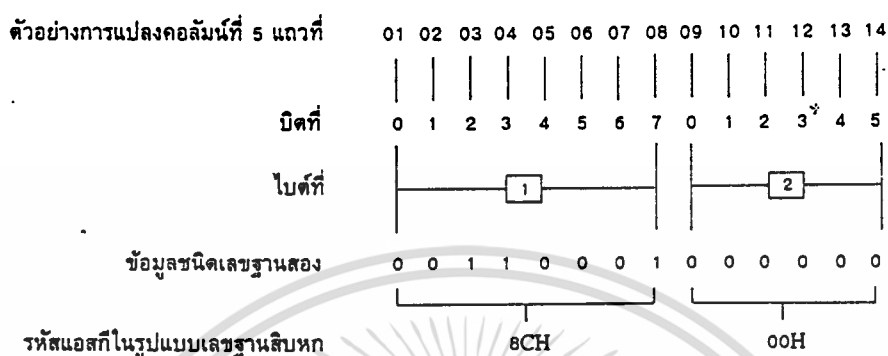
สำหรับวิธีนี้หมายถึง ในการแปลงข้อมูลชนิดเลขฐานสอง 8 บิต ให้เป็นรหัสแอสกี 1

ตัว จะเริ่มนับจากคอลัมน์ที่ 1 , แถวที่ 1, 2, 3, ..., 14 และคอลัมน์ที่ 2 แถวที่ 1, 2, 3, ..., 14 และ

คอลัมน์ที่ 3, 4, ..., 9 ตามลำดับ ดังนั้นในแต่ละคอลัมน์จะสามารถแปลงให้อยู่ในรูปแบบของรหัสค่า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอสกีขนาด 2 ไบต์ และสำหรับตัวอักษร 1 ตัว จะต้องใช้เนื้อที่ในการเก็บขนาด $2 \times 9 = 18$ ไบต์
จากรูปตารางที่ 3.7 เราสามารถนำมาแปลงให้อยู่ในรูปแบบรหัสก็ได้ ดังรูปตารางที่ 3.11



รูปที่ 3.10

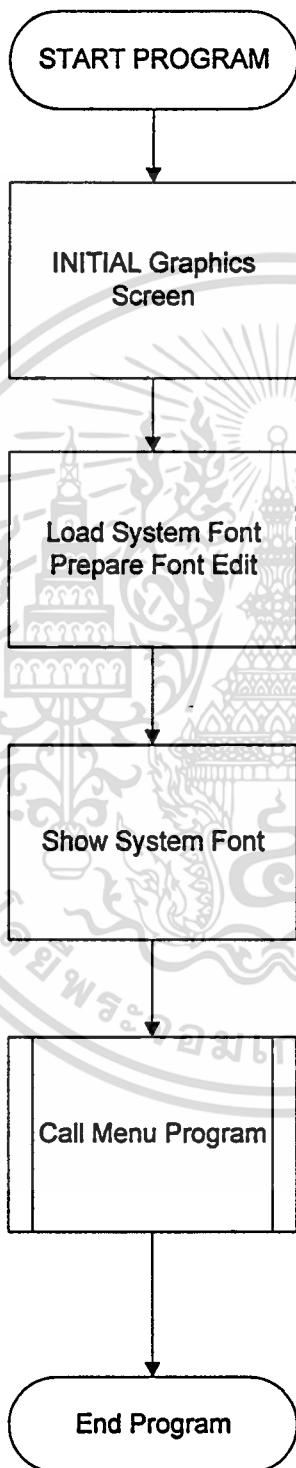
คอลัมน์ที่	1	2	3	4	5	6	7	8	9
แถวที่ 1	0	0	0	0	0	0	0	0	0
แถวที่ 2	0	0	0	0	0	0	0	0	0
แถวที่ 3	0	0	0	0	1	0	0	0	0
แถวที่ 4	0	0	0	1	1	1	0	0	0
แถวที่ 5	0	0	1	1	0	1	1	0	0
แถวที่ 6	0	1	1	0	0	0	1	1	0
แถวที่ 7	0	1	1	0	0	0	1	1	0
แถวที่ 8	0	1	1	1	1	1	1	1	0
แถวที่ 9	0	1	1	0	0	0	1	1	0
แถวที่ 10	0	1	1	0	0	0	1	1	0
แถวที่ 11	0	1	1	0	0	0	1	1	0
แถวที่ 12	0	0	0	0	0	0	0	0	0
แถวที่ 13	0	0	0	0	0	0	0	0	0
แถวที่ 14	0	0	0	0	0	0	0	0	0
ไบต์ที่	1	3	5	7	9	11	13	15	17
	00H	E0H	F0H	98H	8CH	98H	F0H	E0H	00H
	2	4	6	8	10	12	14	16	18
	00H	07H	07H	00H	00H	00H	07H	07H	00H

รูปที่ 3.11 ตารางแสดงวิธี Column major conversion

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า

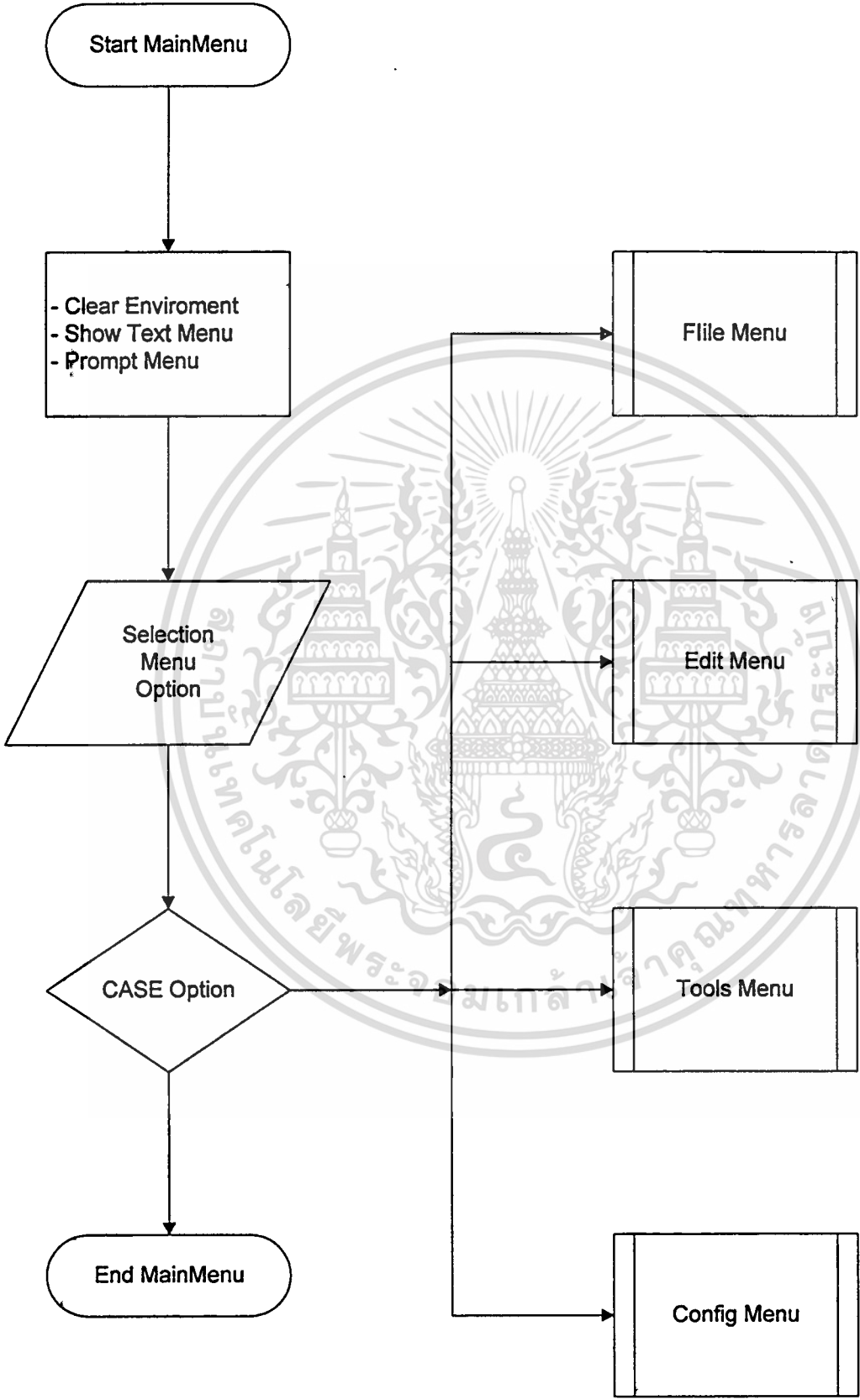
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผังงานของการทำงานในส่วนต่าง ๆ



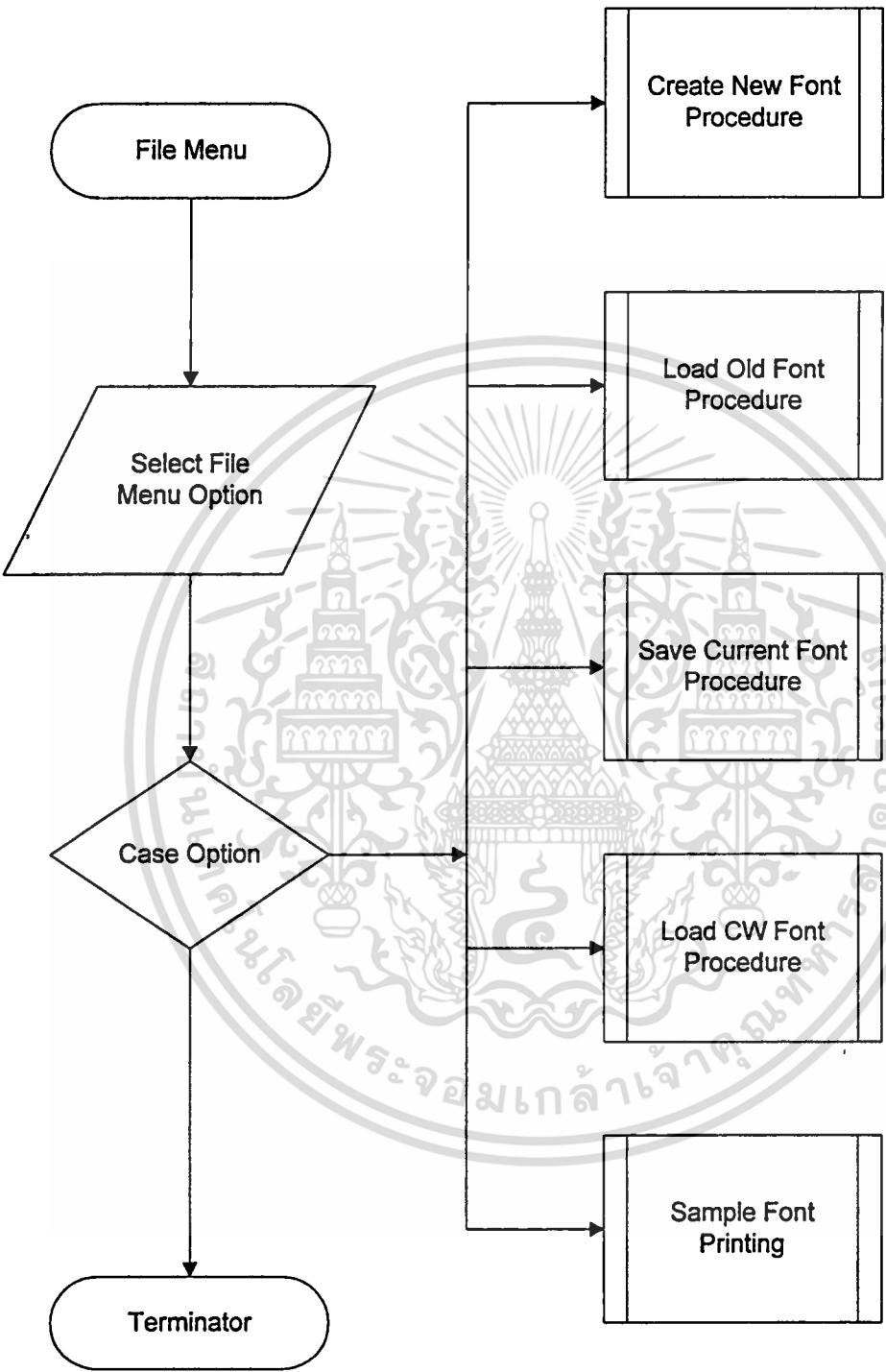
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับจรรยาบรรณเพื่อตรวจสอบเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

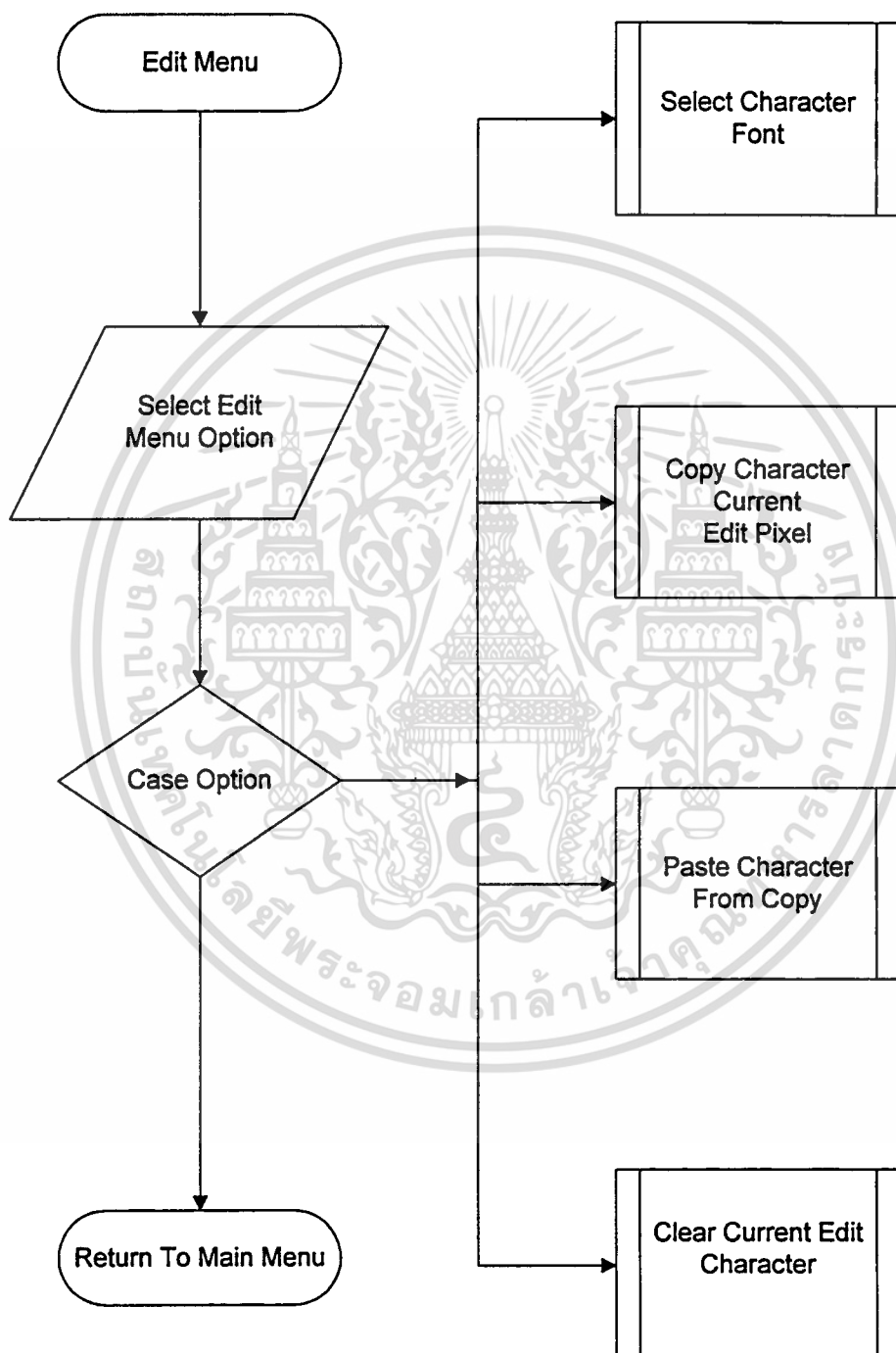
Main Program Flow



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

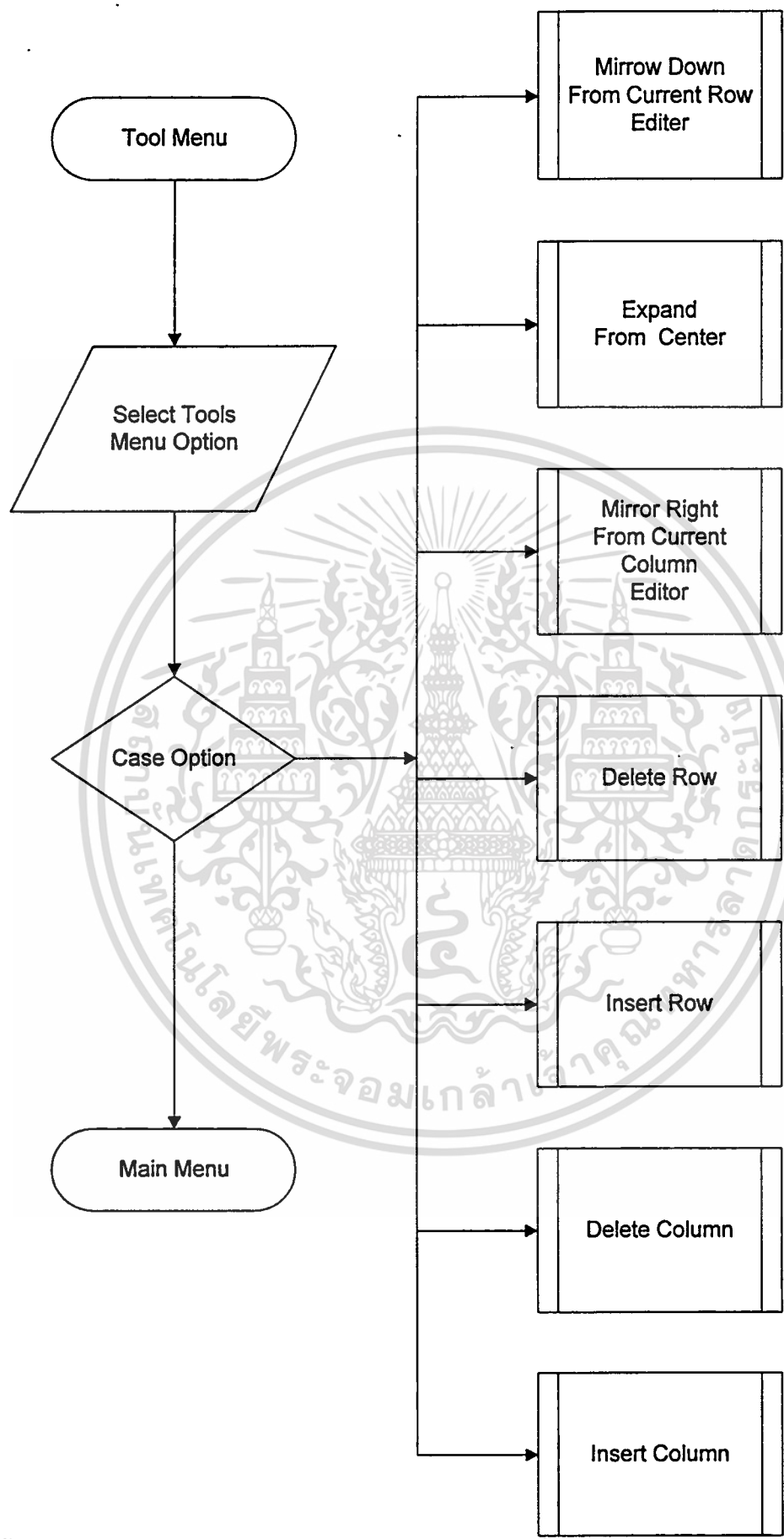
Main Menu Flow





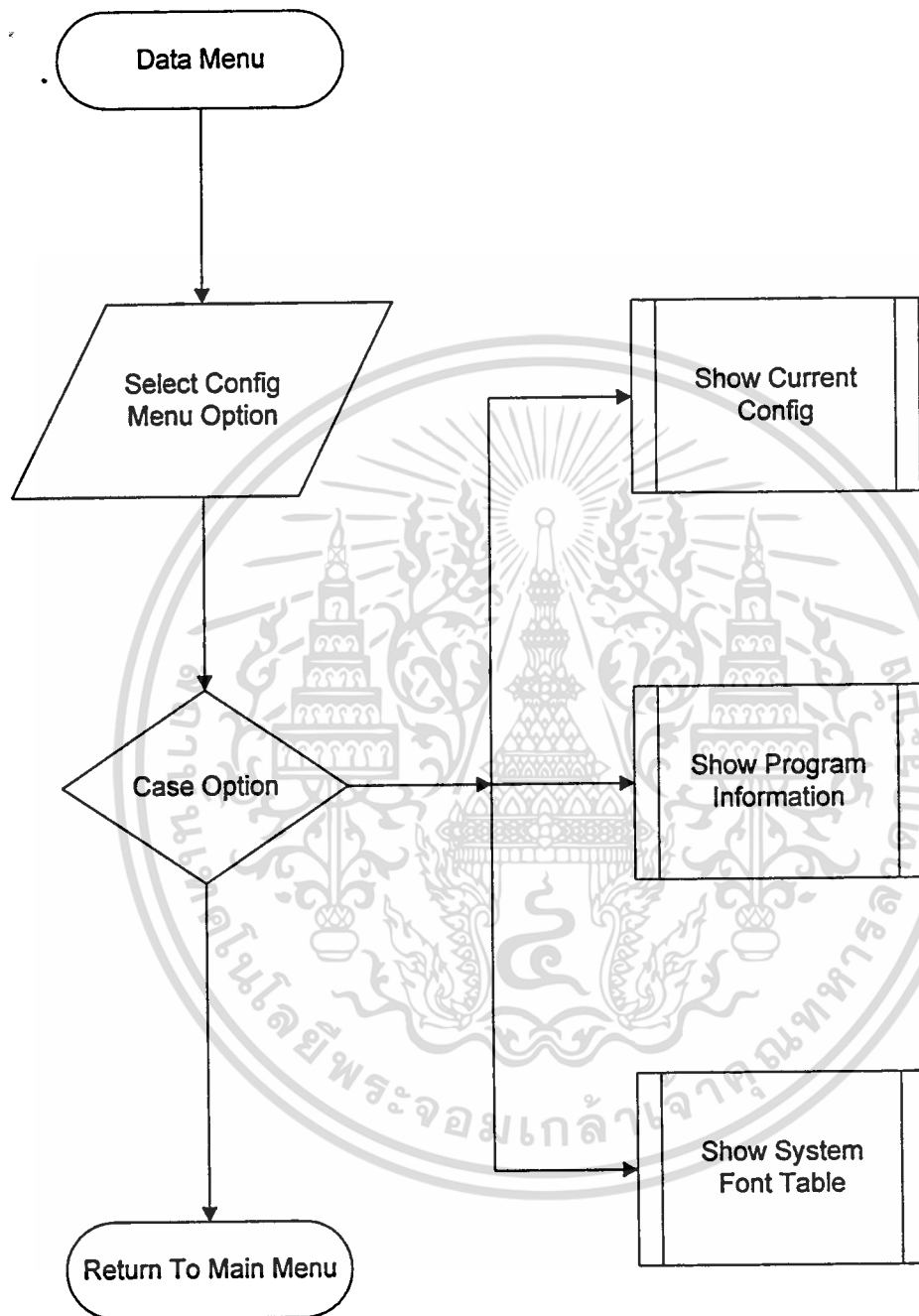
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรณีการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Edit Menu Flow



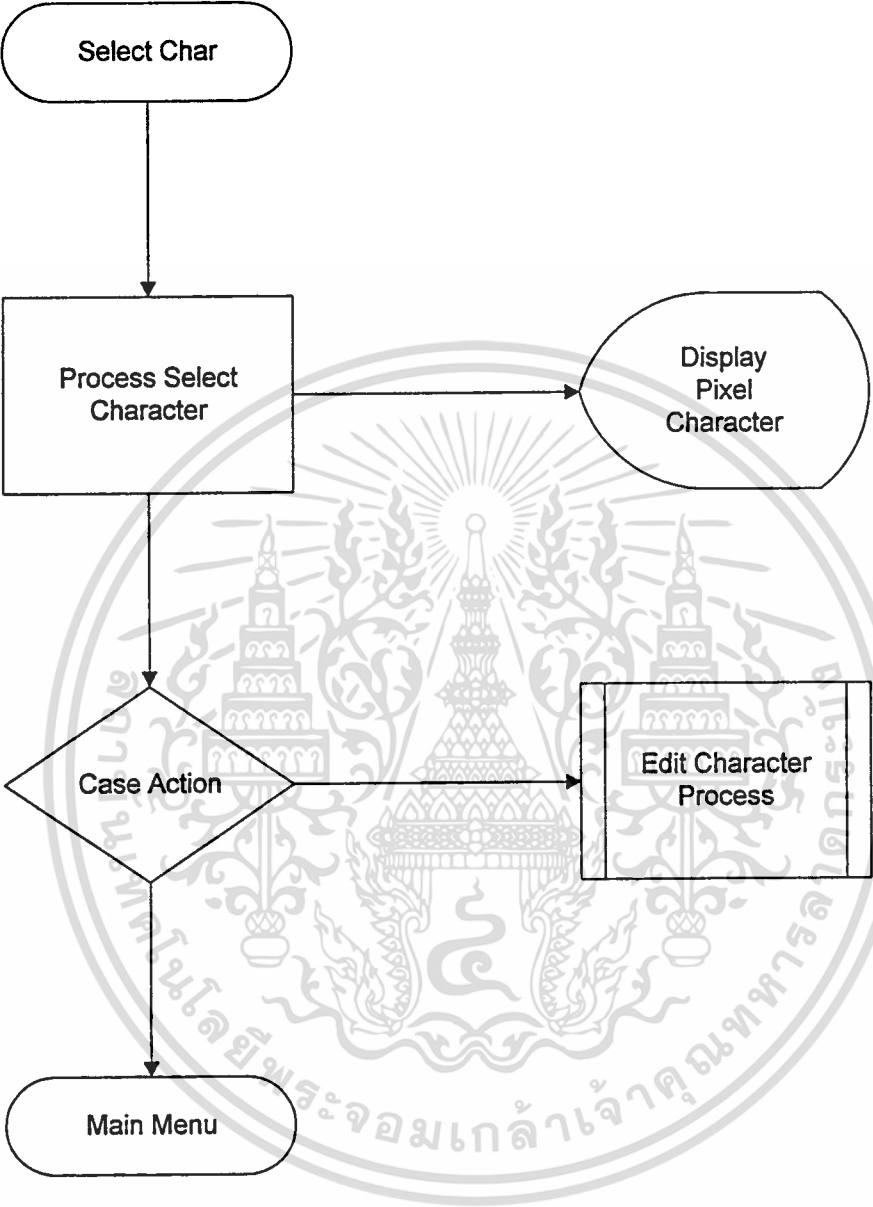
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาของเอกสารทุกครั้งที่มีการนำไปใช้

Tools Menu Flow



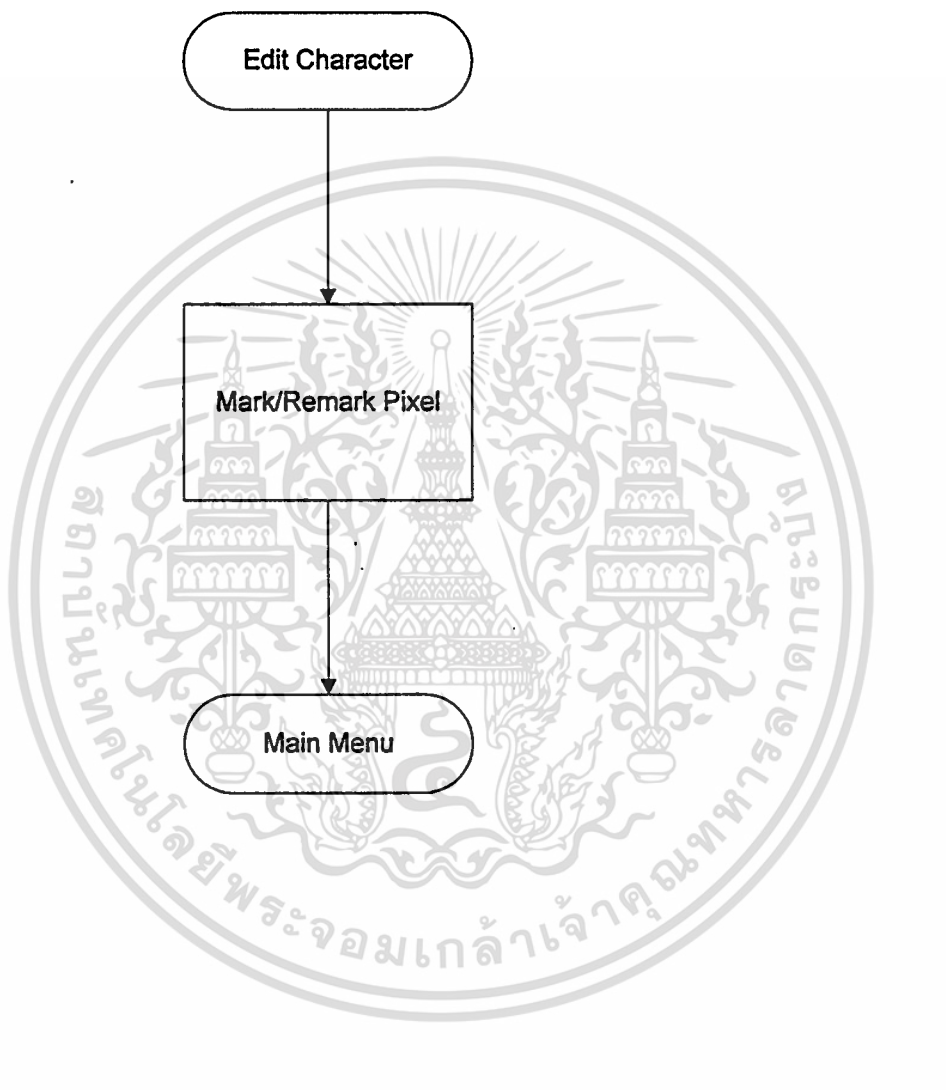
Data Menu Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



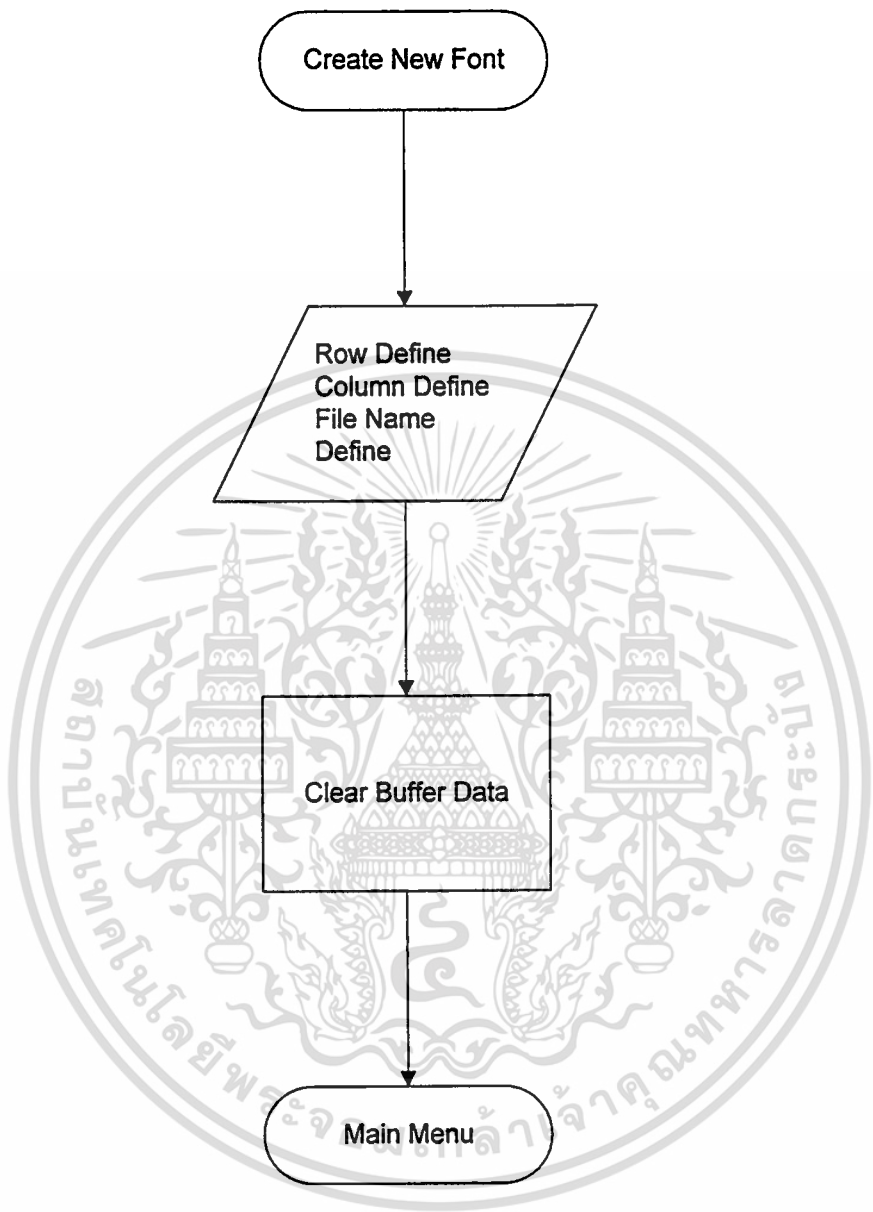
Select Character Procedure Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



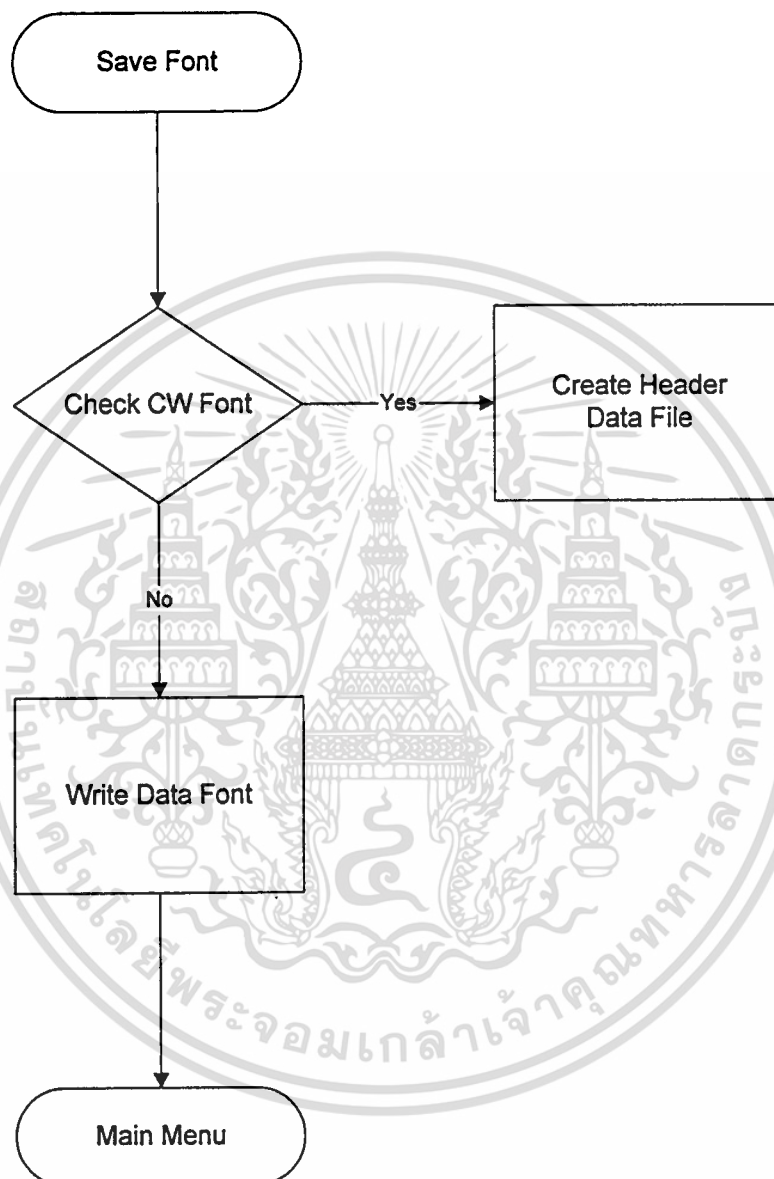
Edit Pixel Character Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Create New Font Procedure Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Save Font Procedure Flow

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิธีการใช้งานโปรแกรม

5.1 ทรัพยากรและส่วนประกอบที่จำเป็นในการใช้งานโปรแกรม

- ทรัพยากรฮาร์ดแวร์ โปรแกรมนี้ต้องการติดตั้งบน เครื่องคอมพิวเตอร์ส่วนบุคคลที่มีความสามารถดังต่อไปนี้

อุปกรณ์	ความต้องการ	หมายเหตุ
MICROPROCESSOR	INTEL 386 UP	
MONITOR CARD	VGA OR HIGHER	
MEMORY	640 K (min)	
KEYBOARD	83 or 101	
DISK SPACE	200,000 Byte	

- ระบบ เพิ่มข้อมูลของ โปรแกรมที่จำเป็น มีดังนี้

- FONTMAIN.EXE เป็นโปรแกรมหลักของ ช่วยออกแบบตัวอักษร
- EGAVGA.BGI เป็นโปรแกรมสำหรับ ควบคุมและให้บริการโปรแกรมหลักในการใช้งานการแสดงผลแบบ VGA
- SYSFONT.BGI เป็นโปรแกรมเก็บตัวอักษรที่ใช้ในการแสดงข้อความต่างๆ บนโปรแกรมหลัก

5.2 การใช้งานโปรแกรม ช่วยออกแบบตัวอักษร

5.2.1 เข้าสู่โปรแกรม ช่วยออกแบบตัวอักษร

C:\FG\FONTMAIN [Press Enter]

โปรแกรมจะทำการ ปรับ รูปแบบการแสดงผลเป็น กราฟิกและทำการ LOAD ข้อมูลระบบที่จำเป็น และแสดงการทำงานดังรูป 5.1

แฟ้มข้อมูล	แก้ไข	เครื่องมือ	ข้อมูล	จบงาน
ชื่อแฟ้ม : CWI1.FON			!"#%&'()*+,-./ 0123456789:;<=>? @ABCDEFGHI JKLMNO PQRSTUvwxyz[\]^_ 'abcdefghijklmno pqrstuvwxyzc1~ αβγδ-ΣΔφπ ρσ μτ ΗΓΓ-ΙΚ Π ρ ·กขคฌงจฉซฌฌฌ รุทฌเตตททฌนปฌฝฝ ฌมยรฤทวตฌสฌหือช ะ ำ ึ ๅ ๆ ็ ่ แ ใ ใ ใ ใ ใ ใ ใ ๐๑๒๓๔๕๖๗๘๙๐๑ NONE 8 20	

รูปที่ 5.2 หน้าโปรแกรมการ LOAD แฟ้มตัวอักษร

5.2.3 องค์ประกอบต่าง ๆ ของโปรแกรม

โปรแกรมถูกออกแบบให้ม้องค์ประกอบหลักอยู่ 4 ส่วนผู้ใช้สามารถใช้ KEYBOARD ในการเปลี่ยนการทำงานไปยังองค์ประกอบต่าง ๆ ซึ่งองค์ประกอบที่เป็นองค์ประกอบปัจจุบัน จะมีขอบของหน้าต่างเป็นสีเหลือง

5.3 รายละเอียดขององค์ประกอบจอภาพของโปรแกรมมีดังนี้

5.3.1. ส่วนของ เมนูทำหน้าที่ในการเลือกทำงาน ต่าง ๆ มีลักษณะและรายละเอียดคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟ้มข้อมูล	แก้ไข	เครื่องมือ	ข้อมูล	จบงาน
สร้างใหม่				
เปิดแฟ้มเก่า				
จัดเก็บข้อมูล				
แก้ไขอักษร CW				
พิมพ์ตัวอักษร				



รูปที่ 5.3 เมนุย่อย เพิ่มข้อมูล

- เมนู เพิ่มข้อมูล มีความสามารถดังต่อไปนี้
 - สร้างตัวอักษรใหม่ ใช้สำหรับออกแบบตัวอักษรใหม่
 - แก้ไขข้อมูลเก่า ใช้สำหรับนำตัวอักษรที่สร้างแล้วมาแก้ไข
 - บันทึกข้อมูล ใช้ในการจัดเก็บตัวอักษรที่ใช้งานและแก้ไขอยู่
 - แก้ไขตัวอักษรของ CW ใช้สำหรับแก้ไขข้อมูลตัวอักษรของ จุฬาโดยเฉพาะ
 - พิมพ์ตัวอักษรที่ออกแบบ ทางเครื่องพิมพ์

แฟ้มข้อมูล	แก้ไข	เครื่องมือ	ข้อมูล	จบงาน
		สะท้อนด้านล่าง ขยายตัวอักษร สะท้อนด้านขวา แทรกบรรทัด ลบบรรทัด แทรกแนวตั้ง ลบแนวตั้ง	! " # % & ' () * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H I J K L M N O P Q R S T U V W X Y Z [\] ^ _ ' a b c d e f g h i j k l m n o p q r s t u v w x y z { } ~ α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω α β γ δ ε ζ η θ ι κ λ μ ν ξ ο π ρ σ τ υ φ χ ψ ω Γ Δ Ε Ζ Η Θ Ι Κ Λ Μ Ν Ξ Ο Π Ρ Σ Τ Υ Φ Χ Ψ Ω	NONE 8 20

รูปที่ 5.5 เมนูเครื่องมือ

- เมนู เครื่องมือ มีความสามารถดังต่อไปนี้

- สะท้อนล่าง ใช้ในการบันทึกสะท้อนจากตำแหน่ง แฉกของจุดในปัจจุบัน
- สะท้อนขวา ใช้ในการบันทึกสะท้อนจากตำแหน่งแนวตั้งของจุดในปัจจุบัน
- ขยายตัวอักษร ใช้ในการขยายตัวอักษรจาก กึ่งกลางความกว้างและความสูงของ

ตัวอักษร

- เพิ่มบรรทัด ใช้ในการเพิ่มแฉกของตัวอักษร จากตำแหน่ง แฉกของจุดในปัจจุบัน
- ลบบรรทัด ใช้ในการลบแฉกของตัวอักษร จากตำแหน่ง แฉกของจุดในปัจจุบัน
- เพิ่มแนวตั้ง ใช้ในการเพิ่มแนวของตัวอักษร จากตำแหน่ง แนวของจุดในปัจจุบัน
- ลบแนวตั้ง ใช้ในการลบแนวของตัวอักษร จากตำแหน่ง แนวของจุดในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

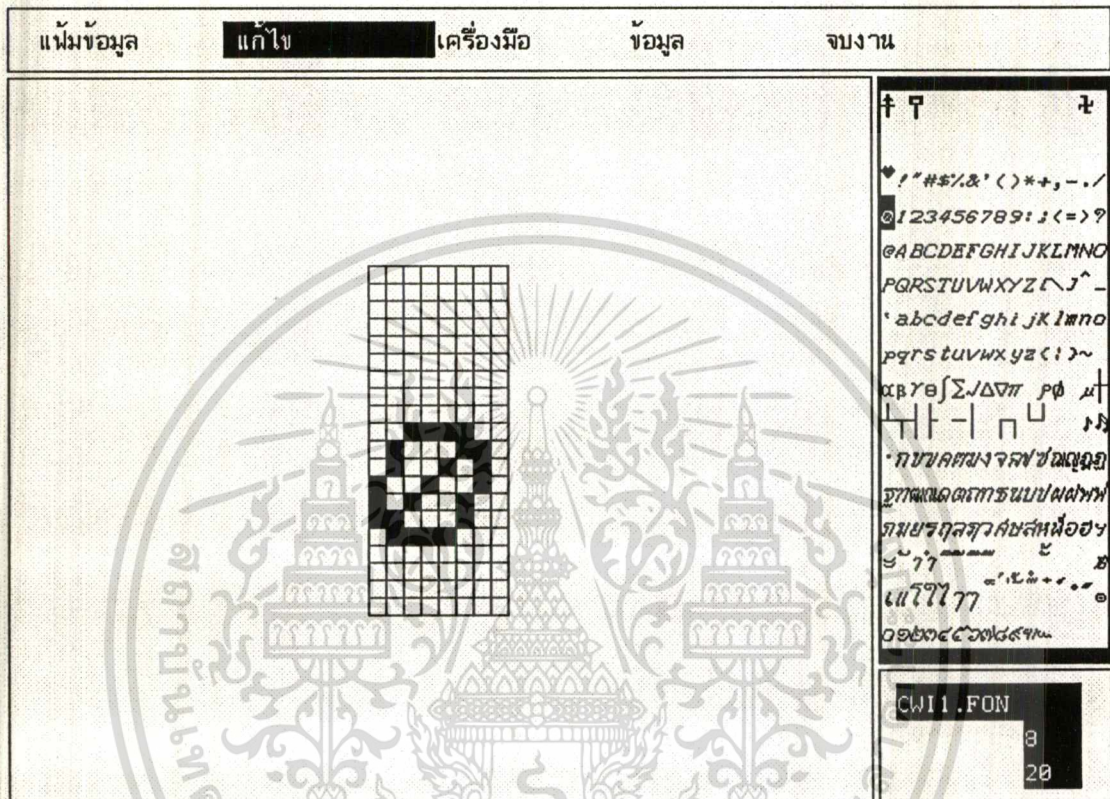
[illegible]

รูปที่ 5.6 เมนูข้อมูล

- เมนู ข้อมูล มีความสามารถดังต่อไปนี้
 - ปรับขนาดตัวอักษรปัจจุบัน
 - แสดง ตัวอักษรของระบบ
 - แสดงตัวอักษรของแฟ้ม
 - แสดงรายละเอียดของโปรแกรม

5.3.2. ส่วนของ FONT VIEW WINDOW

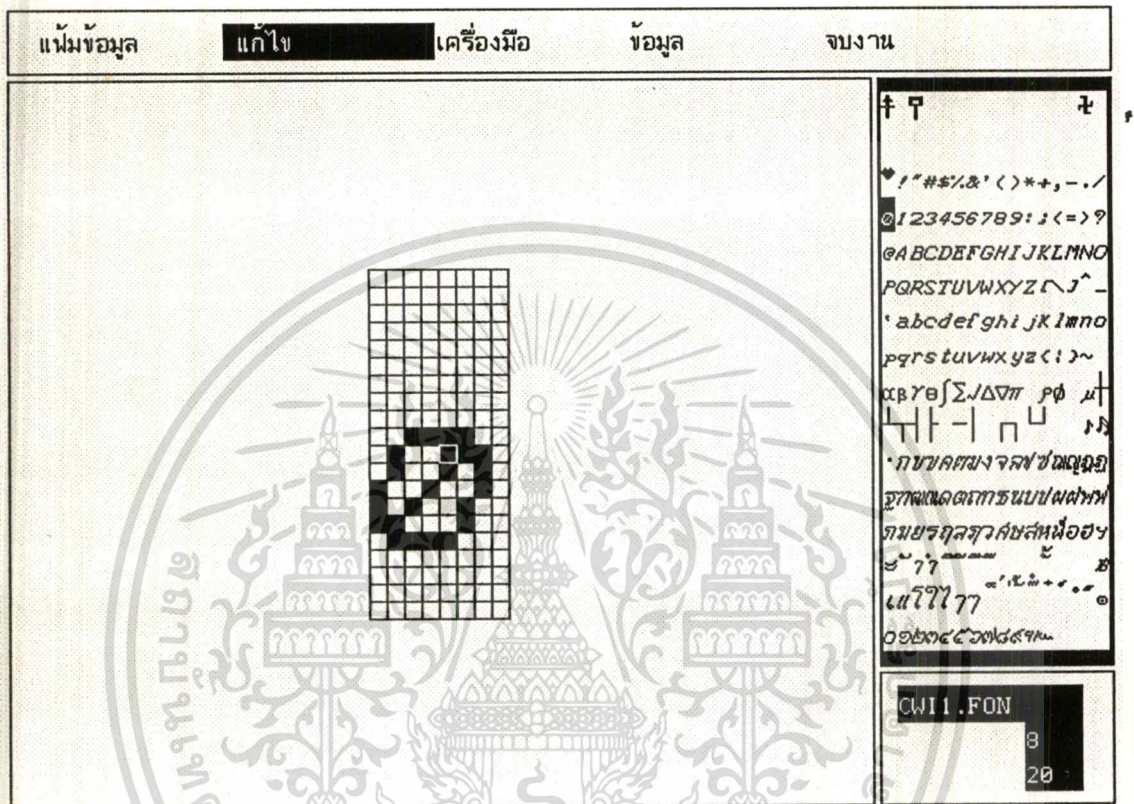
ในส่วนนี้จะทำงานเมื่อ เลือกเมนูแก้ไขจาก เมนูหลัก เลือกเมนูเลือกตัวอักษร ผู้ใช้สามารถ
ใช้ คีย์ลูกศรในการเลือกตัวอักษรที่ต้องการแก้ไข ในกรณีที่เลือกให้กด Return ในกรณีที่ต้องการข
เลิกให้กด Esc แสดงการเลือกในรูปที่ 5.7



รูปที่ 5.7 เลือกตัวอักษร

5.3.3. ส่วนของ EDITOR WINDOW

ส่วนนี้จะทำงานเมื่อมีการเลือกตัวอักษรเท่านั้น ผู้ใช้สามารถใช้ คีย์ลูกศรในการเลือก จุดที่ต้องการ แก้ไขในกรณีจุดเดิมไม่ทึบ จุดปัจจุบันจะแสดงสีขาว ในกรณีที่จุดนั้นทึบ จุดปัจจุบันจะแสดงสีแดง ผู้ใช้สามารถเปลี่ยนค่าได้โดยการกด SPACE BAR



รูปที่ 5.8 แก้ไขตัวอักษร

5.3.4. ส่วนของ INFORMATION WINDOW

ในส่วนนี้จะบอกสถานะของ

- ชื่อแฟ้มปัจจุบัน
- ขนาดของแถว หน่วยเป็นจุด
- ขนาดของความกว้าง หน่วยเป็นจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทวิจารณ์ รูป และแนวทางในการพัฒนา

6.1 บทสรุป

โปรแกรมช่วยออกแบบตัวอักษรนี้ ทางผู้จัดทำได้วางขอบเขตความสามารถของโปรแกรมไว้ดังต่อไปนี้

- ผู้ใช้สามารถกำหนดขนาดของฟอนต์ที่ต้องการออกแบบได้เอง
- สามารถเก็บข้อมูลที่อยู่บนหน้าจอไว้ในแฟ้มที่เราสร้างขึ้น ซึ่งจะอยู่ในไฟล์นามสกุล .FON
- สามารถโหลดข้อมูลจากแฟ้มข้อมูลขึ้นมาแสดง หรือทำการแก้ไขข้อมูลเก่าได้
- สามารถพิมพ์ฟอนต์ที่ออกแบบออกทางเครื่องพิมพ์ได้

จากการที่ได้ทดลองโปรแกรมนี้แล้ว ปรากฏผลที่ได้เป็นที่น่าพอใจในระดับหนึ่ง ข้อจำกัดของโปรแกรมนี้คือ ต้องใช้กับจอภาพ VGA ขึ้นไปความละเอียด 640x480 จุดจนถึงจอภาพแบบ Super VGA ในปัจจุบันนี้โปรแกรมช่วยออกแบบตัวอักษรก็พอมีอยู่บ้างแต่ที่สามารถกำหนดขนาดของฟอนต์ได้เองยังมีน้อยมาก นอกจากนี้โปรแกรมนี้ยังสามารถแก้ไขฟอนต์ของ WORD จุฬา ได้ และโครงการนี้ยังได้สร้างฟอนต์ตัวอย่างเกี่ยวกับสัญลักษณ์พิเศษต่าง ๆ เอาไว้ให้ผู้ใช้เพื่ออำนวยความสะดวกในการใช้งาน

6.2 แนวทางในการพัฒนา

ทางผู้จัดทำพยายามที่จะจัดทำโปรแกรมนี้ให้เป็นโปรแกรมช่วยออกแบบตัวอักษรที่มีประสิทธิภาพมากที่สุดเท่าที่จะทำได้ในระยะเวลาที่จำกัด และนอกเหนือจากนี้เรายังสามารถที่จะพัฒนาโปรแกรมให้ดีขึ้น เช่น

- สามารถให้โปรแกรมนี้สนับสนุนทุกๆ จอภาพที่มีความละเอียดน้อยกว่าจอภาพ VGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
สามารถให้ใช้ MOUSE กับ โปรแกรมช่วยออกแบบตัวอักษรนี้ได้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่โดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ให้สามารถมีคำอธิบาย แนะนำในทุกกรณีเมื่อมีปัญหาในการออกแบบตัวอักษร

โครงการนี้ใช้ภาษาซีในการสร้างโปรแกรม เพราะภาษาซีมีความยืดหยุ่นสูง แต่ในขณะเดียวกันการเรียนรู้ภาษาซีจำเป็นต้องใช้เวลามาก ในส่วนของฟังก์ชันการทำงานที่มีความสลับซับซ้อน สำหรับบุคคลที่ยังไม่มีทักษะ ยังต้องใช้เวลาเพิ่มขึ้นในการศึกษาค้นคว้า ในส่วนของ โครงการนี้จึงสามารถแสดงผลงานได้ดังที่เห็นแต่เพียงเท่านั้น



หนังสืออ้างอิง

1. มัณฑนา ปราการสมุทร, การเขียนชุดคำสั่งภาษาซี, กรุงเทพมหานคร:ไฮเทคพริ้นติ้ง, 2534
2. สุทธิศักดิ์ พงศ์ธนาพาณิชย์, บิตแมปฟอนต์ สำหรับ QUICK Basic, กรุงเทพมหานคร, ซีเอ็ดยุคเข็่น, 2536
3. เจนวิทย์ เหลืองอร่าม, การใช้ TURBO C++ เขียนโปรแกรมภาษา C, กรุงเทพมหานคร, สุขภาพใจ, 2538
4. สมพัฒน์ รุ่งตะวันเรืองศรี, เรียนรู้คอมพิวเตอร์กราฟิก 2 มิติด้วยภาษาซี, กรุงเทพมหานคร, ซีเอ็ดยุคเข็่น, 2535
5. ศิววัฒน์ ศิวบรร, การประยุกต์ใช้งานภาษาซี, กรุงเทพมหานคร, ซีเอ็ดยุคเข็่น, 2535
6. ธันวา ศรีประโม่ง, การเขียนโปรแกรมภาษาซีสำหรับวิศวกรรม, กรุงเทพมหานคร :
โครงการตำราวิชาการมหาวิทยาลัยเทคโนโลยีมหานคร
7. Borland International ,Inc , “Borland C++ Libary Reference” ,1991
8. Borland International ,Inc , “Borland C++ Programming Guide” ,1991

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/
/* Program : FONTMAIN.C */
/* Descrip : Main Program Call All Procedure / Function */
/* < Listing > */
/* Init_Program() */
/* Init_Var() */
/* main() */
/* ----- */

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

#define MAX_COLUMN 80 /* Maximum Column */
#define MAX_ROW 34 /* Maximum Row */
#define MAX_CHAR_WIDTH 40 /* Program Char Width 9 */
#define MAX_CHAR_HIGH 40 /* Program Char High 14 */
#define MAX_EDIT_ROW 30 /* Maximum For Edit Block*/
#define MAX_EDIT_COL 16

```

```

#define SYS_FONT_W      8
#define SYS_FONT        "SYSFONT.FNT"
#define SYS_FONT_VBG    MAGENTA
#define SYS_FONT_VFG    WHITE
#define SYS_FONT_EBG    BLUE
#define SYS_FONT_EFG    YELLOW

#define FONT_HIGH       20
#define FONT_WIDTH      8
#define POINT_WIDTH     10

#define CHAR_SLIP       3      /* Char Slip Hor */
#define CHAR_SPACE      1      /* Space Between Char */
#define BAR_REV_C       LIGHTGRAY /* Color Bar Revers */
#define BAR_NOR_C       BLUE     /* Color Bar Normal */
#define TEXT_NOR_C      WHITE    /* Text Color Normal */
#define TEXT_REV_C      RED      /* Text Revers Color */
#define BORDER_ACT      YELLOW   /* Default Border Color */
#define BORDER_NOR      LIGHTGRAY
#define TEXT_NBG        BLUE
#define TEXT_NFG        WHITE
#define TEXT_RBG        LIGHTGRAY
#define TEXT_RFG        RED

```

```

#define F1_key          59      /* Define Scan Code F1 */
#define F2_key          60      /* Define Scan Code F1 */
#define F3_key          61      /* Define Scan Code F1 */
#define F4_key          62      /* Define Scan Code F1 */
#define F5_key          63      /* Define Scan Code F1 */

```

```

#define F6_key      64          /* Define Scan Code F1 */
#define F7_key      65          /* Define Scan Code F1 */
#define F8_key      66          /* Define Scan Code F1 */
#define F9_key      67          /* Define Scan Code F1 */
#define F10_key     68          /* Define Scan Code F1 */
#define Home_key    71          /* Define Scan Code F1 */
#define Up_key      72          /* Define Scan Code F1 */
#define Pgup_key    73          /* Define Scan Code F1 */
#define Left_key    75          /* Define Scan Code F1 */
#define Right_key   77          /* Define Scan Code F1 */
#define End_key     79          /* Define Scan Code F1 */
#define Down_key    80          /* Define Scan Code F1 */
#define Pgdn_key    81          /* Define Scan Code F1 */
#define SpaceBar_key ' '        /* Define Space Keyboard */
#define Return_key  0x0d        /* Define Return Key */
#define Esc_key     0x1b        /* Define Esc Key */
#define Backspace_key 0x08
#define Thueda_key  0x60

#define Massage_Lenght 30      /* Maximum of Message Lenght */

#define MOUSE_HOLD  0
#define MOUSE_RIGHT 1
#define MOUSE_LEFT  2
#define MOUSE_UP    3
#define MOUSE_DOWN  4

```

```

unsigned char SMOTable[96] =

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0xf6,0xf7,0xf5,0xf9,0xc1,0xa2,0xe3,0xbd,
0xa8,0xc5,0x2f,0x2d,0xc0,0xb6,0xd8,0xd6,
0xa4,0xb5,0xab,0xc7,0xb2,0xaa,0xcc,0x3f,
0xf1,0xc4,0x25,0xa9,0xaf,0xae,0xe2,0xac,
0xe7,0xb3,0xeb,0xc9,0xc8,0x3f,0xec,0xcf,

0xad,0xf0,0xb1,0xa6,0xb8,0xea,0xce,0x22,
0x29,0xeb,0x28,0xba,0x5c,0xc5,0xd9,0xf8,
0x60,0xbf,0xd4,0xe1,0xa1,0xd3,0xb4,0xe0,

0xe9,0xc3,0xe8,0xd2,0xca,0xb7,0xd7,0xb9,
0xc2,0xe6,0xbe,0xcb,0xd0,0xd5,0xcd,0xb4,
0xbb,0xd1,0xbc,0xb0,0x7c,0x2c,0x7e,0x7f };

```
char SysFont[256][20];  
int SysFontAdr_X[16][16];  
int SysFontAdr_Y[16][16];  
int SysFontNumber[16][16];  
char DataFont[256][60];  
char DescDat[20];  
char Fontname[12] = "ITALIC.FON";  
char Curr_Fname[12],Msg[12];  
int Curr_BG,Curr_FG;  
int Curr_width,Curr_high,CurrView_X,CurrView_Y;  
int Curr_SubMenu,Curr_MainMenu;  
int Curr_Erow,Curr_Ecol,Curr_char;  
int Conner_Col,Conner_Row;  
int EditValue[MAX_EDIT_ROW][MAX_EDIT_COL];  
int EditAdr_X[MAX_EDIT_ROW][MAX_EDIT_COL];  
int EditAdr_Y[MAX_EDIT_ROW][MAX_EDIT_COL];
```

```
int TempValue[MAX_EDIT_ROW][MAX_EDIT_COL];
int Mirror_Dat[MAX_EDIT_ROW][MAX_EDIT_COL];
int CWFONT,Byte_size;
```

```
#include "fontlib.c"
#include "fontscr.c"
#include "fontmenu.c"
#include "fontmous.c"
#include "fontutl.c"
#include "fontfile.c"
#include "fontprm.c"
```

```
/******
```

```
/* Funtion : Initail_Program */
```

```
/* Descrip : Init Font Program */
```

```
/******
```

```
Init_Program()
```

```
{
```

```
Init_mouse();
```

```
SolidBorder(62,2,79,19,MAGENTA);
```

```
Active_Border(0);
```

```
Curr_width = 8;
```

```
Curr_high = 20;
```

```
strcpy(Curr_Fname,"NONE");
```

```
if(LoadSysFont(SYS_FONT))
```

```
{
```

```
ViewSysFont();
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Init_Var();
}

/*****
/* Funtion : Init_Var                                     */
/* Desc   : Set up Variable Enviroment                     */
*****/

Init_Var()
{
    int ix,iy;
    div_t ispec;
    char col_txt[3],row_txt[3],by_txt[3];

    Conner_Col = ((( 40 - Curr_width ) / 2 ) * POINT_WIDTH )+ 50;
    Conner_Row = ((( 40 - Curr_high ) / 2 ) * POINT_WIDTH )+ 60;
    CurrView_X = 0;
    CurrView_Y = 0;

    for(ix=0;ix<3;ix++);
    {
        col_txt[ix] = 0x00;
        row_txt[ix] = 0x00;
    }

    Byte_size = Curr_width / 8;
    if( Curr_width > ( Byte_size * 8 ))
        Byte_size++;

    iy = Curr_high;
    ix = Curr_width;

```

```

itoa(ix,col_txt,10);
itoa(iy,row_txt,10);
itoa(Byte_size,by_txt,10);

```

```

Revers_Bar(64,20,12,GREEN);
Revers_Bar(73,21,3,GREEN);
Revers_Bar(73,22,3,GREEN);
Printxy(64,20,Curr_Fname);
Printxy(64,21,"COLUMN :");
Printxy(73,21,col_txt);
Printxy(64,22,"ROW   :");
Printxy(73,22,row_txt);

```

```

/* Check Byte Size */
/*
Revers_Bar(78,21,0);
Printxy(78,21,by_txt);
*/
}

```

```

/*****

```

```

/* Main Program */

```

```

*****/

```

```

main()

```

```

{

```

```

    char action;

```

```

    int ix;

```

```

    clrscr();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (Open_Graph())
{
    Init_Program();    /* Initail Enviroment Data */
    Main_Menu();       /* Main Menu Function Call */
    Close_Graph();
}
else
{
    clrscr();
    gotoxy(25,12);
    printf(" Program Can Not Initial Graphics Mode \n");
    getch();
}
}

```

□

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/
/* Program : fontlib.C */
/* Descrip : Program Font Library */
/* Listing */
/* -----*/
/* void DrawSysFont(int x,int y,int inchar,int emode) */
/* int SelectSysChar() */
/* ViewSysFont() */
/* int SelectDataChar() */
/* int sqrt2(int data) */
/* KeepFont() */
/* DrawBigFont2(int inchar) */
/* CheckEdit() */
/* DrawDataFont(int x,int y,int inchar,int emode) */
/* EditChar() */
/* RefreshChar() */
/* ClearChar() */
/* ClearDataFont() */
/* CopyChar() */
/* PasteChar() */
/*=====*/
/

```

```

/*-----*/
/* Draw System Font Function */
/* Update : 12/06/1996 */
/* Test : Complete */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void DrawSysFont(int x,int y,int inchar,int emode)
```

```
{
```

```
    int i,j;
```

```
    for(i=0;i<20;i++)
```

```
    {
```

```
        for(j=0;j<8;j++)
```

```
        {
```

```
            if( ( (SysFont[inchar][i] >> ( 7-j ) ) &1 ) )
```

```
                putpixel(x+j,y+i,Curr_FG);
```

```
            else
```

```
            {
```

```
                if(emode)
```

```
                    putpixel(x+j,y+i,Curr_BG);
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

```
/*-----*/
```

```
/* Display System Char To View Window */
```

```
/* Update : 14/06/1996 */
```

```
/* Test : Complete */
```

```
/*-----*/
```

```
ViewSysFont()
```

```
{
```

```
    int ix,iy,ikk,chridx,Frow,Fcol,xx,yy;
```

```
    SolidBorder(62,2,79,19,MAGENTA);
```

```
    Frow = Goto_Y(3);
```

เอกสารนี้เป็นเอกสารที่สํานักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Fcol = Goto_X(63);
chridx = 0;
Curr_FG = SYS_FONT_VFG;
Curr_BG = SYS_FONT_VBG;

for(ix=0;ix<16;ix++)
{
for(iy=0;iy<16;iy++)
{
xx = ( iy * SYS_FONT_W ) + Fcol;
yy = ( ix * SYS_FONT_H ) + Frow;

DrawSysFont(xx ,yy,chridx,1 );

chridx++;
SysFontAdr_X[iy][ix] = xx;
SysFontAdr_Y[iy][ix] = yy;
SysFontNumber[iy][ix] = chridx - 1;
}
}
}

/*-----*/
/* Select System Char From View Window */
/* Update : 14/06/1996 */
/* Test : Complete */
/*-----*/

int SelectSysChar()
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int cur_row,cur_col,ix,iy,chridx;
```

```
int ExitSel,Exitcode;
```

```
Active_Border(2);
```

```
cur_row = CurrView_Y;
```

```
cur_col = CurrView_X;
```

```
ExitSel = 0;
```

```
Exitcode = 0;
```

```
chridx = SysFontNumber[cur_col][cur_row];
```

```
ix = SysFontAdr_X[cur_col][cur_row];
```

```
iy = SysFontAdr_Y[cur_col][cur_row];
```

```
Curr_FG = SYS_FONT_EFG;
```

```
Curr_BG = SYS_FONT_EBG;
```

```
DrawSysFont(ix,iy,chridx,1);
```

```
while (!ExitSel)
```

```
{
```

```
key = Readkey();
```

```
switch(key)
```

```
{
```

```
case Left_key:
```

```
{
```

```
chridx = SysFontNumber[cur_col][cur_row];
```

```
ix = SysFontAdr_X[cur_col][cur_row];
```

```
iy = SysFontAdr_Y[cur_col][cur_row];
```

```
Curr_FG = SYS_FONT_VFG;
```

```
Curr_BG = SYS_FONT_VBG;
```

เอกสารนี้เป็น DrawSysFont(ix,iy,chridx,1);งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(cur_col > 0)
    cur_col--;
else
    cur_col = 15;

    chridx = SysFontNumber[cur_col][cur_row];
    ix = SysFontAdr_X[cur_col][cur_row];
    iy = SysFontAdr_Y[cur_col][cur_row];
    Curr_FG = SYS_FONT_EFG;
    Curr_BG = SYS_FONT_EBG;
    DrawSysFont(ix,iy,chridx,1);

    break;
}
case Right_key:
{
    chridx = SysFontNumber[cur_col][cur_row];
    ix = SysFontAdr_X[cur_col][cur_row];
    iy = SysFontAdr_Y[cur_col][cur_row];
    Curr_FG = SYS_FONT_VFG;
    Curr_BG = SYS_FONT_VBG;
    DrawSysFont(ix,iy,chridx,1);

    if(cur_col < 15)
        cur_col++;
    else
        cur_col = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนลิขสิทธิ์ไว้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ix      = SysFontAdr_X[cur_col][cur_row];
iy      = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_EFG;
Curr_BG = SYS_FONT_EBG;
DrawSysFont(ix,iy,chridx,1);
break;
}

case Up_key:
{
chridx  = SysFontNumber[cur_col][cur_row];
ix      = SysFontAdr_X[cur_col][cur_row];
iy      = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_VFG;
Curr_BG = SYS_FONT_VBG;
DrawSysFont(ix,iy,chridx,1);

if(cur_row > 0)
    cur_row--;
else
    cur_row = 15;

chridx  = SysFontNumber[cur_col][cur_row];
ix      = SysFontAdr_X[cur_col][cur_row];
iy      = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_EFG;
Curr_BG = SYS_FONT_EBG;
DrawSysFont(ix,iy,chridx,1);
break;
}

```

เอกสารนี้ **case Down_key:** เป็นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    chridx  = SysFontNumber[cur_col][cur_row];
    ix      = SysFontAdr_X[cur_col][cur_row];
    iy      = SysFontAdr_Y[cur_col][cur_row];
    Curr_FG = SYS_FONT_VFG;
    Curr_BG = SYS_FONT_VBG;
    DrawSysFont(ix,iy,chridx,1);

    if(cur_row < 15)
        cur_row++;
    else
        cur_row = 0;

    chridx  = SysFontNumber[cur_col][cur_row];
    ix      = SysFontAdr_X[cur_col][cur_row];
    iy      = SysFontAdr_Y[cur_col][cur_row];
    Curr_FG = SYS_FONT_EFG;
    Curr_BG = SYS_FONT_EBG;
    DrawSysFont(ix,iy,chridx,1);
    break;
}

```

case Return_key :

```

{
    CurrView_X = cur_col;
    CurrView_Y = cur_row;
    ExitSel    = 1;
    Exitcode   = 1;
    break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case Esc_key):
{
    CurrView_X = cur_col;
    CurrView_Y = cur_row;
    ExitSel = 1;
    Exitcode = 0;
    break;
}
} /* Switch */
Curr_char = chridx;
DrawBigFont2(chridx);
} /* While */
return(Exitcode);
} /* End Select Font */

```

```

/*-----*/
/* Calulate Bin Value */
/* Update : 6/06/1996 */
/* Test : Complete */
/*-----*/

```

```

int sqrt2(int data)
{
    int results;
    int ix;

    results = 1;
    for(ix=0;ix < data ;ix++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 results = results * 2;
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
printf("%3i",results);
*/
}
return results;
}

```

```

/*-----*/
/* Keep Data Edit Char */
/* Update : 6/06/1996 */
/* Test : Complete */
/*-----*/

```

```

KeepFont()
{
int i,j,k,row,col;
int xx,yy;
char dummy;

for(i=0;i< Curr_high;i++)
{
for(xx=0;xx<Byte_size;xx++)
{
k = 0;
yy = ( i * Byte_size ) + xx;
for(j=0;j<8;j++)
{
if(EditValue[i][j+(xx*8)])
{
k = k + sqrt2(7-j);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    DataFont[Curr_char][yy] = (char)k;
} /* Byte_size */
}
}

```

```

/*-----*/
/* Check Edit Correct (Test) */
/* Update : 6/06/1996 */
/* Test : Complete */
/*-----*/

```

```

CheckEdit()
{
    int ix,iy;
    int x1,y1,x2,y2;

    for(iy=0;iy<Curr_high;iy++)
        for(ix=0;ix<Curr_width;ix++)
        {
            x1 = ix * 10 + 50;
            y1 = iy * 10 + 60;
            x2 = x1 + POINT_WIDTH;
            y2 = y1 + POINT_WIDTH;

            if(EditValue[iy][ix])
            {

```

```

                setfillstyle(SOLID_FILL,RED);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 bar(x1+1,y1+1,x2-1,y2-1);

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
else
{
    setcolor(YELLOW);
    rectangle(x1,y1,x2,y2);
    setfillstyle(SOLID_FILL,TEXT_NFG);
    bar(x1+1,y1+1,x2-1,y2-1);
}
}
}

```

```

/*-----*/
/* Copy Char */
/* Update : 6/06/1996 */
/* Test : Complete */
/*-----*/

```

```
CopyChar()
```

```

{
    int ix,iy;

    for(iy=0;iy<Curr_high;iy++)
        for(ix=0;ix<Curr_width;ix++)
            TempValue[iy][ix] = EditValue[iy][ix];
}

```

```
/*-----*/
```

```
/* Paste Char From Copy */
```

```
*/
```

```

/* Update : 6/06/1996                                     */
/* Test   : Complete                                       */
/*-----*/
PasteChar()
{
    int ix,iy,x1,y1,x2,y2;
    for(iy=0;iy<Curr_high;iy++)
        for(ix=0;ix<Curr_width;ix++)
        {
            x1 = Conner_Col + (ix * POINT_WIDTH );
            y1 = Conner_Row + (iy * POINT_WIDTH );
            x2 = x1 + POINT_WIDTH;
            y2 = y1 + POINT_WIDTH;

            EditValue[iy][ix] = TempValue[iy][ix];
            if(EditValue[iy][ix])
            {
                setfillstyle(SOLID_FILL,YELLOW);
                bar(x1+1,y1+1,x2-1,y2-1);
            }
            _else
            {
                setcolor(YELLOW);
                rectangle(x1,y1,x2,y2);
                setfillstyle(SOLID_FILL,BLUE);
                bar(x1+1,y1+1,x2-1,y2-1);
            }
        } /* For ix */
}

```

KeepFont();

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ViewDataFont();

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
/*-----*/
```

```
/* Clear Big Char */
```

```
/* Update : 6/06/1996 */
```

```
/* Test : Complete */
```

```
/*-----*/
```

```
ClearChar()
```

```
{
```

```
int ix,iy,x1,y1,x2,y2;
```

```
for(iy=0;iy<Curr_high;iy++)
```

```
for(ix=0;ix<Curr_width;ix++)
```

```
{
```

```
x1 = Conner_Col + (ix * POINT_WIDTH);
```

```
y1 = Conner_Row + (iy * POINT_WIDTH);
```

```
x2 = x1 + POINT_WIDTH;
```

```
y2 = y1 + POINT_WIDTH;
```

```
EditValue[iy][ix] = 0;
```

```
setcolor(YELLOW);
```

```
rectangle(x1,y1,x2,y2);
```

```
setfillstyle(SOLID_FILL,BLUE);
```

```
bar(x1+1,y1+1,x2-1,y2-1);
```

```
}
```

```
KeepFont();
```

```
}
```

```
/*-----*/
```

```
/* refresh Big Char */
```

```
/* Update : 6/06/1996 */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Test : Complete */
/*-----*/

RefreshChar()
{
    int ix,iy,x1,y1,x2,y2;
    for(iy=0;iy<Curr_high;iy++)
        for(ix=0;ix<Curr_width;ix++)
        {
            x1 = Conner_Col + (ix * POINT_WIDTH );
            y1 = Conner_Row + (iy * POINT_WIDTH );
            x2 = x1 + POINT_WIDTH;
            y2 = y1 + POINT_WIDTH;

            if(EditValue[iy][ix])
            {
                setfillstyle(SOLID_FILL,YELLOW);
                bar(x1+1,y1+1,x2-1,y2-1);
            }
            else
            {
                setcolor(YELLOW);
                rectangle(x1,y1,x2,y2);
                setfillstyle(SOLID_FILL,BLUE);
                bar(x1+1,y1+1,x2-1,y2-1);
            }
        } /* For ix */
    KeepFont();
    ViewDataFont();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/* Select Data Character */
/* Update : 6/06/1996 */
/* Test : Review */
/*-----*/

```

```
int SelectDataChar()
```

```

{
    char key;
    int cur_row,cur_col,ix,iy,chridx;
    int ExitSel,Exitcode;

    Active_Border(2);

    cur_row = CurrView_Y;
    cur_col = CurrView_X;
    ExitSel = 0;
    Exitcode = 0;

    chridx = SysFontNumber[cur_col][cur_row];
    ix = SysFontAdr_X[cur_col][cur_row];
    iy = SysFontAdr_Y[cur_col][cur_row];
    Curr_FG = SYS_FONT_EFG;
    Curr_BG = SYS_FONT_EBG;

```

```
    DrawDataFont(ix,iy,chridx,1);
```

```
    while (!ExitSel)
```

```
    {
```

```
        key = Readkey();
```

```
        switch(key)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
case Left_key:
{
chridx = SysFontNumber[cur_col][cur_row];
ix      = SysFontAdr_X[cur_col][cur_row];
iy      = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_VFG;
Curr_BG = SYS_FONT_VBG;
DrawDataFont(ix,iy,chridx,1);

if(cur_col > 0)
cur_col--;
else
cur_col = 15;

chridx = SysFontNumber[cur_col][cur_row];
ix      = SysFontAdr_X[cur_col][cur_row];
iy      = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_EFG;
Curr_BG = SYS_FONT_EBG;
DrawDataFont(ix,iy,chridx,1);

break;
}

```

```

case Right_key:

```

```

{
chridx = SysFontNumber[cur_col][cur_row];
ix      = SysFontAdr_X[cur_col][cur_row];
iy      = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_VFG;

```

```

Curr_BG = SYS_FONT_VBG;
DrawDataFont(ix,iy,chridx,1);

if(cur_col < 15)
    cur_col++;
else
    cur_col = 0;

chridx = SysFontNumber[cur_col][cur_row];
ix = SysFontAdr_X[cur_col][cur_row];
iy = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_EFG;
Curr_BG = SYS_FONT_EBG;
DrawDataFont(ix,iy,chridx,1);
break;
}
case Up_key:
{
chridx = SysFontNumber[cur_col][cur_row];
ix = SysFontAdr_X[cur_col][cur_row];
iy = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_VFG;
Curr_BG = SYS_FONT_VBG;
DrawDataFont(ix,iy,chridx,1);

```

```

if(cur_row > 0)

```

```

    cur_row--;

```

```

else

```

```

    cur_row = 15;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

chridx = SysFontNumber[cur_col][cur_row];
ix     = SysFontAdr_X[cur_col][cur_row];
iy     = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_EFG;
Curr_BG = SYS_FONT_EBG;
DrawDataFont(ix,iy,chridx,1);
break;
}
case Down_key:
{
chridx = SysFontNumber[cur_col][cur_row];
ix     = SysFontAdr_X[cur_col][cur_row];
iy     = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_VFG;
Curr_BG = SYS_FONT_VBG;
DrawDataFont(ix,iy,chridx,1);

if(cur_row < 15)
cur_row++;
else
cur_row = 0;

chridx = SysFontNumber[cur_col][cur_row];
ix     = SysFontAdr_X[cur_col][cur_row];
iy     = SysFontAdr_Y[cur_col][cur_row];
Curr_FG = SYS_FONT_EFG;
Curr_BG = SYS_FONT_EBG;
DrawDataFont(ix,iy,chridx,1);
break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
case Return_key :
{
    CurrView_X = cur_col;
    CurrView_Y = cur_row;
    ExitSel  = 1;
    Exitcode = 1;
    break;
}
case Esc_key :
{
    CurrView_X = cur_col;
    CurrView_Y = cur_row;
    ExitSel  = 1;
    Exitcode = 0;
    break;
}
} /* Switch */
Curr_char = chridx;
DrawBigFont2(chridx);
} /* While */
return(Exitcode);
} /* End Select Data Font */

```

```

/*-----*/

```

```

/* Display Data Char To View Window */

```

```

/* Update : 14/06/1996 */

```

```

/* Test : Complete */

```

```

/*-----*/

```

```

ViewDataFont()

```

```

{

int ix,iy,ikk,chridx,Frow,Fcol,xx,yy;

SolidBorder(62,2,79,19,MAGENTA);

Frow = Goto_Y(3);

Fcol = Goto_X(63);

chridx = 0;

Curr_FG = SYS_FONT_VFG;

Curr_BG = SYS_FONT_VBG;

for(ix=0;ix<16;ix++)
{
for(iy=0;iy<16;iy++)
{
xx = ( iy * Curr_width ) + Fcol;
yy = ( ix * Curr_high ) + Frow;

DrawDataFont(xx ,yy,chridx,1 );

chridx++;

SysFontAdr_X[iy][ix] = xx;

SysFontAdr_Y[iy][ix] = yy;

SysFontNumber[iy][ix] = chridx - 1;
}
}
}

```

```

/*-----*/

```

```

/* Draw Data To Edit Win

```

```

*/

```

```

/* Update : 6/06/1996

```

```

*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Test : Review */
/*-----*/

```

```

DrawDataFont(int lx,int ly,int inchar,int emode)

```

```

{
    int i,j,yy,xx;
    int x1,y1;

    for(i=0;i< Curr_high ;i++)
    {
        for(xx=0;xx<Byte_size;xx++)
        {
            yy = ( i * Byte_size ) + xx;
            for(j=0;j < 8;j++)
            {
                x1 = lx + ( j + (xx * 8 ) );
                y1 = ly + i;
                if( ( ( xx * 8 ) + j ) < Curr_width )
                {
                    if( ( (DataFont[inchar][yy] >> ( 7-j ) ) &1 ) )
                        putpixel(x1,y1,Curr_FG);
                    else
                    {
                        if(emode)
                            putpixel(x1,y1,Curr_BG);
                    }
                }
            } /* If < Curr_width */
        }
    } /* xx */
} /* i */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
```

```
/*-----*/
```

```
/* Edit Char View */
```

```
/* Update : 6/06/1996 */
```

```
/* Test : Complete */
```

```
/*-----*/
```

```
EditChar()
```

```
{
```

```
char key;
```

```
int cur_row,cur_col,chrldx,ix,iy;
```

```
int ExitSel;
```

```
int x1,y1,x2,y2;
```

```
cur_row = Curr_high / 2;
```

```
cur_col = Curr_width / 2;
```

```
ExitSel = 0;
```

```
Active_Border(3);
```

```
/* First Edit */
```

```
x1 = EditAdr_X[cur_row][cur_col];
```

```
y1 = EditAdr_Y[cur_row][cur_col];
```

```
x2 = x1 + POINT_WIDTH;
```

```
y2 = y1 + POINT_WIDTH;
```

```
if(EditValue[cur_row][cur_col])
```

```
{
```

```
setfillstyle(SOLID_FILL,YELLOW);
```

```
bar(x1+1,y1+1,x2-1,y2-1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else
{
    setcolor(YELLOW);
    rectangle(x1,y1,x2,y2);
    setfillstyle(SOLID_FILL,BLUE);
    bar(x1+1,y1+1,x2-1,y2-1);
}
/* Process Edit */
while (!ExitSel)
{
    Curr_Erow = cur_row;
    Curr_Ecol = cur_col;
    key = Readkey();
    switch(key)
    {
    case '-':
    {
        DelRow();
        break;
    }
    case '+':
    {
        InsRow();
        break;
    }
    case '*':
    {

```

```

        ExpandChar();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

```

        break;

```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

case Backspace_key :
{
    DelCol();
    break;
}

case '/':
{
    InsCol();
    break;
}

case Left_key:
{
    x1 = EditAdr_X[cur_row][cur_col];
    y1 = EditAdr_Y[cur_row][cur_col];
    x2 = x1 + POINT_WIDTH;
    y2 = y1 + POINT_WIDTH;

    if(EditValue[cur_row][cur_col])
    {
        setfillstyle(SOLID_FILL,YELLOW);
        bar(x1+1,y1+1,x2-1,y2-1);
    }
    else
    {
        setcolor(YELLOW);
        rectangle(x1,y1,x2,y2);
        setfillstyle(SOLID_FILL,BLUE);
        bar(x1+1,y1+1,x2-1,y2-1);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if(cur_col > 0)
            cur_col--;
        else
            cur_col = Curr_width - 1;
        x1 = EditAdr_X[cur_row][cur_col];
        y1 = EditAdr_Y[cur_row][cur_col];
        x2 = x1 + POINT_WIDTH;
        y2 = y1 + POINT_WIDTH;

        if(EditValue[cur_row][cur_col])
        {
            setfillstyle(SOLID_FILL,RED);
            bar(x1+1,y1+1,x2-1,y2-1);
        }
        else
        {
            setcolor(YELLOW);
            rectangle(x1,y1,x2,y2);
            setfillstyle(SOLID_FILL,TEXT_NFG);
            bar(x1+1,y1+1,x2-1,y2-1);
        }
        break;
    }

case Right_key:
{
    x1 = EditAdr_X[cur_row][cur_col];
    y1 = EditAdr_Y[cur_row][cur_col];
    x2 = x1 + POINT_WIDTH;
    y2 = y1 + POINT_WIDTH;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(EditValue[cur_row][cur_col])
{
    setfillstyle(SOLID_FILL,YELLOW);
    bar(x1+1,y1+1,x2-1,y2-1);
}
else
{
    setcolor(YELLOW);
    rectangle(x1,y1,x2,y2);
    setfillstyle(SOLID_FILL,BLUE);
    bar(x1+1,y1+1,x2-1,y2-1);
}
if(cur_col < (Curr_width - 1 ))
    cur_col++;
else
    cur_col = 0;
x1 = EditAdr_X[cur_row][cur_col];
y1 = EditAdr_Y[cur_row][cur_col];
x2 = x1 + POINT_WIDTH;
y2 = y1 + POINT_WIDTH;

```

```

if(EditValue[cur_row][cur_col])
{
    setfillstyle(SOLID_FILL,RED);
    bar(x1+1,y1+1,x2-1,y2-1);
}
else
{

```

```

    setcolor(YELLOW);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        setfillstyle(SOLID_FILL,TEXT_NFG);
        bar(x1+1,y1+1,x2-1,y2-1);
    }
    break;
}

case Up_key:
{
    x1 = EditAdr_X[cur_row][cur_col];
    y1 = EditAdr_Y[cur_row][cur_col];
    x2 = x1 + POINT_WIDTH;
    y2 = y1 + POINT_WIDTH;

    if(EditValue[cur_row][cur_col])
    {
        setfillstyle(SOLID_FILL,YELLOW);
        bar(x1+1,y1+1,x2-1,y2-1);
    }
    else
    {
        setcolor(YELLOW);
        rectangle(x1,y1,x2,y2);
        setfillstyle(SOLID_FILL,BLUE);
        bar(x1+1,y1+1,x2-1,y2-1);
    }
}

if(cur_row > 0)
    cur_row--;

else
    cur_row = Curr_high - 1;

    x1 = EditAdr_X[cur_row][cur_col];
    y1 = EditAdr_Y[cur_row][cur_col];

```

```

x2 = x1 + POINT_WIDTH;
y2 = y1 + POINT_WIDTH;

if(EditValue[cur_row][cur_col])
{
    setfillstyle(SOLID_FILL,RED);
    bar(x1+1,y1+1,x2-1,y2-1);
}
else
{
    setcolor(YELLOW);
    rectangle(x1,y1,x2,y2);
    setfillstyle(SOLID_FILL,TEXT_NFG);
    bar(x1+1,y1+1,x2-1,y2-1);
}
break;
}
case Down_key:
{
    x1 = EditAdr_X[cur_row][cur_col];
    y1 = EditAdr_Y[cur_row][cur_col];
    x2 = x1 + POINT_WIDTH;
    y2 = y1 + POINT_WIDTH;

    if(EditValue[cur_row][cur_col])
    {
        setfillstyle(SOLID_FILL,YELLOW);
        bar(x1+1,y1+1,x2-1,y2-1);
    }
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    setcolor(YELLOW);
    rectangle(x1,y1,x2,y2);
    setfillstyle(SOLID_FILL,BLUE);
    bar(x1+1,y1+1,x2-1,y2-1);
}

if(cur_row < ( Curr_high - 1 ))
    cur_row++;
else
    cur_row = 0;

x1 = EditAdr_X[cur_row][cur_col];
y1 = EditAdr_Y[cur_row][cur_col];
x2 = x1 + POINT_WIDTH;
y2 = y1 + POINT_WIDTH;

if(EditValue[cur_row][cur_col])
{
    setfillstyle(SOLID_FILL,RED);
    bar(x1+1,y1+1,x2-1,y2-1);
}
else
{
    setcolor(YELLOW);
    rectangle(x1,y1,x2,y2);
    setfillstyle(SOLID_FILL,TEXT_NFG);
    bar(x1+1,y1+1,x2-1,y2-1);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
    case SpaceBar_key :
    {
        EditValue[cur_row][cur_col] = !(EditValue[cur_row][cur_col]);
        x1 = EditAdr_X[cur_row][cur_col];
        y1 = EditAdr_Y[cur_row][cur_col];
        x2 = x1 + POINT_WIDTH;
        y2 = y1 + POINT_WIDTH;

        if(EditValue[cur_row][cur_col])
        {
            setfillstyle(SOLID_FILL,RED);
            bar(x1+1,y1+1,x2-1,y2-1);
        }
        else
        {
            setcolor(YELLOW);
            rectangle(x1,y1,x2,y2);
            setfillstyle(SOLID_FILL,TEXT_NFG);
            bar(x1+1,y1+1,x2-1,y2-1);
        }
        break;
    }
    case Esc_key :
    {
        ExitSel = 1;
    }
} /* Switch */
} /* While */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

KeepFont();
ViewDataFont();
} /* Edit Char */

```

```

/*-----*/
/* Draw Big Font View */
/* Update : 6/06/1996 */
/* Test : Complete */
/*-----*/

```

```

DrawBigFont2(int inchar)
{
    int ij,ik,kk;
    int x1,y1,x2,y2,ix,iy,xx,yy;

    for(i=0;i< Curr_high ;i++)
    {
        for(xx=0;xx<Byte_size;xx++)
        {
            yy = ( i * Byte_size ) + xx;
            for(j=0;j < 8;j++)
            {
                x1 = Conner_Col + ( ( j + (xx * 8 ) ) * POINT_WIDTH );
                y1 = Conner_Row + ( i * POINT_WIDTH );
                x2 = x1 + POINT_WIDTH;
                y2 = y1 + POINT_WIDTH;

                if( ( ( xx * 8 ) + j ) < Curr_width )
                {
                    setcolor(YELLOW);

```

```

EditAdr_X[i][j+(xx * 8)] = x1;
EditAdr_Y[i][j+(xx * 8)] = y1;
if( ( (DataFont[inchar][yy] >> ( 7-j ) ) &1 ) )
{
    setfillstyle(SOLID_FILL,YELLOW);
    bar(x1+1,y1+1,x2-1,y2-1);
    EditValue[i][7-(7-j)+(xx*8)] = 1;
}
else
{
    setfillstyle(SOLID_FILL,BAR_NOR_C);
    bar(x1+1,y1+1,x2-1,y2-1);
    EditValue[i][7-(7-j)+(xx*8)] = 0; /* yy Replace by i 0000 */
}
} /* If < Curr_width */
}
} /* xx */
} /* i */
}

/*-----*/
/* Clear Data Font */
/* Update : 6/06/1996 */
/* Test : Complete */
/*-----*/

ClearDataFont()
{
    int ix,iy;

    for(iy=0;iy < 256;iy++)
        for(ix=0;ix < 60;ix++)

```

DataFont[iy][ix] = 0x00;

}

□



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/
/* Program : FONTMENU.C */
/* Descrip : Program Pulldown Menu */
/* Listing */
/* -----*/
/* Information() */
/* CharRun() */
/* FileAction() */
/* EditAction() */
/* ToolAction() */
/* Init_Conf() */
/* Information() */
/* HelpActive() */
/* Popup(Locate_x,Locate_y,Width,Choice,sub_txt,exit_code) */
/* Main_Menu() */
/* -----*/

/* -----## POPUP ##-----*/
Popup(Locate_x,Locate_y,Width,Choice,sub_txt,exit_code)
int Locate_x,Locate_y,Width,Choice,*exit_code;
char *sub_txt[];
{
    int left,top,right,bottom,ix;
    int x1,y1,x2,y2;
    int curr_sub,exit_sub;
    char key;

    void *Scr_save;
    int size;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

curr_sub = 0;
exit_sub = 0;
left     = Locate_x + 1;
top      = Locate_y;
right    = Locate_x + Width + 1;
bottom   = Locate_y + Choice + 1;

x1 = Goto_X(Locate_x -1);
y1 = Goto_Y(Locate_y -1);
x2 = Goto_X(right + 2 );
y2 = Goto_Y(bottom + 2 );
size = imagesize(x1, y1, x2, y2);
Scr_save = malloc( size );
getimage(x1, y1, x2, y2, Scr_save);

SolidBorder(left-1,top-1,right+1,bottom,BAR_NOR_C);

setcolor(BORDER_ACT);
Border(left-1,top-1,right+1,bottom);

for ( ix = 0 ; ix < Choice + 1 ; ix++)
    Printxy( left , top + ix ,sub_txt[ix]);

Revers_Bar( left , top + curr_sub , Width);
Revers_Text();
Printxy( left , top + curr_sub ,sub_txt[curr_sub] );

```

```

while ( exit_sub == 0 )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

key = Readkey();
switch(key)
{
    case Down_key :
    {
        Normal_Bar( left , top + curr_sub ,Width);
        Normal_Text();
        Printxy( left , top + curr_sub ,sub_txt[curr_sub] );
        if( curr_sub < Choice )
            curr_sub++;
        else
            curr_sub = 0;
        Revers_Bar( left , top + curr_sub ,Width);
        Revers_Text();
        Printxy( left , top + curr_sub ,sub_txt[curr_sub] );
    }
    break;
    case Up_key :
    {
        Normal_Bar( left , top + curr_sub ,Width);
        Normal_Text();
        Printxy( left , top + curr_sub ,sub_txt[curr_sub] );
        if( curr_sub > 0 )
            curr_sub--;
        else
            curr_sub = Choice;
        Revers_Bar( left , top + curr_sub ,Width);
        Revers_Text();
        Printxy( left , top + curr_sub ,sub_txt[curr_sub] );
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    case Right_key :
    {
        putimage(x1 ,y1 , Scr_save, COPY_PUT);
        free(Scr_save);
        exit_code = 1;
        Curr_SubMenu = -1;
        return;
    }
    case Left_key :
    {
        putimage(x1 ,y1 , Scr_save, COPY_PUT);
        free(Scr_save);
        exit_code = -1;
        Curr_SubMenu = -2;
        return;
    }
    case Return_key :
    {
        putimage(x1 ,y1 , Scr_save, COPY_PUT);
        free(Scr_save);
        exit_sub = 1;
        Curr_SubMenu = curr_sub;
        return;
    }
} /* End Switch */
} /* End While */

```

```

} /* End Popup */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/* ----- ## MENU ## ----- */
```

```
Main_Menu()
```

```
{
```

```
int ix,iy,max_main,curr_menu,curr_popup,sub_exit_code;
```

```
char key,exit_menu,popup_sts;
```

```
int Main_Width = 14;
```

```
int Max_choice[] = { 5,4,7,3 };
```

```
int Width_choice[] = { 17,17,17,17 };
```

```
char *menu_txt[10] =
```

```
    {"เพิ่มข้อมูล",
```

```
    "แก้ไข",
```

```
    "เครื่องมือ",
```

```
    "ข้อมูล",
```

```
    "จบงาน"};
```

```
char *File_bar[17] =
```

```
    {"สร้างใหม่",
```

```
    "เปิดเพิ่มเก่า",
```

```
    "จัดเก็บข้อมูล",
```

```
    "แก้ไขอักษร CW",
```

```
    "Print Data"};
```

```
char *Edit_bar[17] =
```

```
    {"เลือกตัวอักษร",
```

```
    "จำตัวอักษร",
```

```
    "ลอกตัวอักษร",
```

```
    "ลบตัวอักษร"};
```

```
char *Utility_bar[17] =
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{" สะท้อนด้านล่าง ",
  " ขยายตัวอักษร  ",
  " สะท้อนด้านขวา ",
  " แทรกบรรทัด  ",
  " ลบบรรทัด  ",
  " แทรกแนวตั้ง  ",
  " ลบแนวตั้ง  "};

/*
char *Help_bar[19] =
{ " ปรับขนาดตัวอักษร  ",
  " แสดงอักษรของระบบ  ",
  " แสดงอักษรของแฟ้ม  ",
  " รายละเอียดโปรแกรม "};

*/
char *Help_bar[19] =
{ " แสดงอักษรของระบบ  ",
  " แสดงอักษรของแฟ้ม  ",
  " รายละเอียดโปรแกรม "};

char Main_Menu_Mess[ ][Message_Length]
= {"File Maintenance",
  "Font Edit Menu",
  "Utility Menu",
  "Program Help",
  "Quit Program"};

```

```
max_main = 4;
```

```
curr_menu = 0;
```

```
curr_popup = 0;
```

```
exit_menu = "N";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
popup_sts = "N";  
sub_exit_code = 0;
```

```
Active_Border(1);
```

```
Normal_Text();
```

```
for ( ix = 0 ; ix < max_main + 1 ; ix++)
```

```
    Printxy( ( ix * Main_Width) + 2 , 1 ,menu_txt[ix]);
```

```
Revers_Bar((curr_menu * Main_Width) + 2, 1 ,Main_Width);
```

```
Revers_Text();
```

```
Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
```

```
Information();
```

```
while ( exit_menu != "Y")
```

```
{
```

```
    Active_Border(1);
```

```
    if ( (curr_menu==3)&(Curr_SubMenu==(-1) ) )
```

```
        Curr_SubMenu = 0;
```

```
    if ( (curr_menu==0)&(Curr_SubMenu==(-2) ) )
```

```
        Curr_SubMenu = 0;
```

```
    if (Curr_SubMenu < 0)
```

```
    {
```

```
        if ( Curr_SubMenu == -1 )
```

```
            { /* Rigth Active */
```

```
                Normal_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
```

```
                Normal_Text();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
if( curr_menu < max_main )
curr_menu++;
else
curr_menu = 0;
Revers_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
Revers_Text();
Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
}
if ( Curr_SubMenu == -2 )
{
Normal_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
Normal_Text();
Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
if( curr_menu > 0 )
curr_menu--;
else
curr_menu = max_main;
Revers_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
Revers_Text();
Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
}
if ( curr_menu != 4 )
key = Return_key;
}
else
{
key = Readkey();
popup_sts = "N";

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(key)
{
    case Right_key :
    {
        Normal_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
        Normal_Text();
        Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
        if( curr_menu < max_main )
            curr_menu++;
        else
            curr_menu = 0;
        Revers_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
        Revers_Text();
        Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
    }
    break;
    case Left_key :
    {
        Normal_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
        Normal_Text();
        Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
        if( curr_menu > 0 )
            curr_menu--;
        else
            curr_menu = max_main;
        Revers_Bar( (curr_menu * Main_Width ) + 2, 1,Main_Width);
        Revers_Text();
        Printxy( ( curr_menu * Main_Width ) + 2, 1 ,menu_txt[curr_menu] );
    }
    break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

case Return_key :

```
{  
if ( popup_sts == "N")  
    popup_sts = "Y";  
switch(curr_menu)
```

```
{
```

case 0 :

```
{
```

```
/* Message(" File Manu Active "); */
```

```
Popup((Main_Width * curr_menu + 1),3,
```

```
Width_choice[curr_menu],
```

```
Max_choice[curr_menu] - 1,
```

```
File_bar,sub_exit_code);
```

```
FileAction();
```

```
break;
```

```
}
```

case 1 :

```
{
```

```
/* Message(" Edit Menu Active "); */
```

```
Popup((Main_Width * curr_menu + 1),3,
```

```
Width_choice[curr_menu],
```

```
Max_choice[curr_menu] - 1,
```

```
Edit_bar,sub_exit_code);
```

```
EditAction();
```

```
break;
```

```
}
```

case 2 :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        /* Message(" Utility Menu Active "); */
        Popup((Main_Width * curr_menu + 1),3,
        Width_choice[curr_menu],
        Max_choice[curr_menu] - 1,
        Utility_bar,sub_exit_code);
        ToolAction();
        break;
    }
case 3 :
    {
        /* Message(" Help Active "); */
        Popup((Main_Width * curr_menu + 1),3,
        Width_choice[curr_menu],
        Max_choice[curr_menu] - 1,
        Help_bar,sub_exit_code);
        HelpActive();
        break;
    }
case 4 :
    {
        exit_menu = "Y";
        return;
    }
} /* End Of Switch Current Manu Enter */

}

break;

} /* End Switch Select key At Main Menu */

/* Message(Main_Menu_Mess[curr_menu]); */

key = " ";
} /* End While Loop */

```

```
} /* End Main Menu Function */
```

```
FileAction()
```

```
{
```

```
switch(Curr_SubMenu)
```

```
{
```

```
case 0 : /* New File Procedure */
```

```
{
```

```
ClearEditWin();
```

```
ClearDataFont();
```

```
ViewDataFont();
```

```
ReadString(2,3,TEXT_RFG,12,Curr_Fname,"ชื่อเพิ่ม :");
```

```
CWFONT=0;
```

```
Init_Conf();
```

```
CreateFont(Curr_Fname);
```

```
Init_Var();
```

```
Init_Table();
```

```
break;
```

```
}
```

```
case 1 : /* Load Old File */
```

```
{
```

```
CWFONT = 0;
```

```
ReadString(2,3,TEXT_RFG,12,Curr_Fname,"ชื่อเพิ่ม :");
```

```
if(!(LoadDataFont(Curr_Fname)))
```

```
    Message(" ไม่พบแฟ้มข้อมูล ");
```

```
SolidBorder(62,2,79,19,MAGENTA);
```

```
Active_Border(2);
```

```
ViewDataFont();
```

```
Init_Var();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CWFONT = 0;

break;

}

case 2 : /* Save Procedure */
{
    if(!(SaveDataFont(Curr_Fname)))
        Message(" ไม่สามารถบันทึก เพิ่มข้อมูลได้ ");
    SolidBorder(62,2,79,19,MAGENTA);
    Active_Border(2);
    ViewSysFont();
    ClearEditWin();
    CWFONT = 0;
    break;
}

case 3 : /* Load Cw File */
{
    CWFONT = 1;
    ReadString(2,3,TEXT_RFG,12,Curr_Fname,"ชื่อเพิ่ม :");
    if(!(LoadDataFont(Curr_Fname)))
        Message(" ไม่พบเพิ่มข้อมูลตัวอักษร ");
    SolidBorder(62,2,79,19,MAGENTA);
    Active_Border(2);
    ViewDataFont();
    Init_Var();
    CWFONT = 1;
    break;
}

case 4 : /* Print Data File */
{
    PrintFile();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
} /* Curr_SubMenu */

}

```

```

EditAction()
{
    switch(Curr_SubMenu)
    {
        case 0 :
        {
            if(SelectDataChar())
                EditChar();
            break;
        } /* Case 0 */
        case 1:
        {
            CopyChar();
            break;
        }
        case 2:
        {
            PasteChar();
            break;
        }
        case 3:
        {
            ClearChar();
            break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    } /* switch */  
}
```

ToolAction()

```
{  
    switch(Curr_SubMenu)  
    {  
        case 0 :  
        {  
            DownMirror();  
            break;  
        }  
        case 1 :  
        {  
            ExpandChar();  
            break;  
        }  
        case 2 :  
        {  
            RightMirror();  
            break;  
        }  
        case 3 :  
        {  
            InsRow();  
            break;  
        }  
        case 4 :  
        {  
            DelRow();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
case 5 :
    {
        InsCol();
        break;
    }
case 6 :
    {
        DelCol();
        break;
    }
}

Init_Conf()
{
    Curr_high = ReadIntData(3,2,3,"Set High");
    Curr_width = ReadIntData(3,2,3,"Set Width");
    if(Curr_width > MAX_EDIT_COL)
    {
        Message(" โปรแกรมจำกัดความกว้างสูงสุด 40 Point");
        Curr_width = MAX_EDIT_COL;
    }
    if(Curr_width < 8)
    {
        Message(" ความกว้างถูกตั้งเป็น 8 Point");
        Curr_width = 8;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if(Curr_high > MAX_EDIT_ROW)
{
    Message(" โปรแกรมจำกัดบรรทัดสูงสุด 40 Point");
    Curr_high = MAX_EDIT_ROW;
}
if(Curr_high < 8)
{
    Message(" บรรทัดถูกตั้งเป็น 8 Point");
    Curr_high = 8;
}
CWFONT = 0;
return;
}

CharRun()
{
    int ix;
    char st;

    for(ix=0;ix<256;ix++)
    {
        DrawBigFont2(ix);
        delay(300);
        itoa(ix,st,10);
        Printxy(3,3,st);
    }
}

```

HelpActive()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 {
 ไม่ว่าจะเป็นใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char newfname[12];
```

```
switch(Curr_SubMenu)
```

```
{ /*
```

```
case 0 : Set System Config
```

```
{
```

```
    If(strcmp(Curr_Fname,"NONE"))
```

```
    {
```

```
        Init_Conf();
```

```
        Init_Var();
```

```
        Init_Table();
```

```
    }
```

```
    else
```

```
        Message(" ยังไม่มีเพิ่มข้อมูล ");
```

```
        break;
```

```
    }
```

```
*/
```

```
case 0 : /* Show System Font */
```

```
{
```

```
    SolidBorder(62,2,79,19,MAGENTA);
```

```
    Active_Border(2);
```

```
    ViewSysFont();
```

```
    break;
```

```
}
```

```
case 1 : /* Show Data Font */
```

```
{
```

```
    /*
```

```
    If(strcmp(Curr_Fname,"NONE"))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    SolidBorder(62,2,79,19,MAGENTA);
    Active_Border(2);
    ViewDataFont();
}
else
    Massage(" ยังไม่มีเพิ่มข้อมูล ");
*/
SolidBorder(62,2,79,19,MAGENTA);
Active_Border(2);
ViewDataFont();
break;
}
case 2 : /* Program Help Information */
{
    Information();
    break;
}
} /* Switch */
}

```

Information()

```

{
    ClearEditWin();
    Active_Border(3);
    /*
    Border(16,5,24,12);
    */
    Revers_Bar(15,6,30);
    Revers_Bar(15,7,30);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Revers_Bar(15,8,30);  
Revers_Bar(15,9,30);  
Revers_Bar(15,10,30);  
Revers_Bar(15,11,30);  
Revers_Bar(15,12,30);  
Printxy(17,8," โปรแกรม Font Generator ");  
Printxy(17,9," พัฒนาโดย นาย โสธร จ้ายเจริญ ");  
Printxy(17,10," นาย เมธา รุ่งพัฒน์พันธ์ ");  
getch();  
ClearEditWin();
```

}



```

/*-----*/
/* Print Font View                                */
/* Update : 6/11/1996                             */
/* Test  : .....                                */
/*-----*/

```

CopyByte(char byte)

```

{
    biosprint(0,byte,0);
}

```

InitPrinter()

```

{
    int n1,n2,data;

    CopyByte(27);CopyByte(42);

    n1 = ( Curr_width * 16 ) % 256;
    n2 = ( Curr_width * 16 ) / 256;

    CopyByte(27);CopyByte(51);CopyByte(24); /* Line Space */
    CopyByte(27);CopyByte(108);CopyByte(5); /* Left Margin */

```

```

    CopyByte(27);CopyByte(42);CopyByte(0);CopyByte(n1);CopyByte(n2);

```

```

    /* 0 For Normal Mode, 1 For Condend */

```

```

}

```

FormFeed()

```

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CopyByte(12); /* Form Feed */
}

LineFeed()
{
    CopyByte(27);CopyByte(51);CopyByte(24);
    CopyByte(10); /* New Line */
}

LineSkip()
{
    CopyByte(27);CopyByte(0);
    CopyByte(10);
}

PrintChar(int inchar,int level)
{
    int i,j,ik,kk;
    int x1,y1,x2,y2,ix,iy,xx,yy,y_dif;
    int bit_val[8]={ 128,64,32,16,8,4,2,1 };
    int byte_val;

    for(i=0;i< Curr_high ;i++) /* Line Control */
    {
        for(xx=0;xx<Byte_size;xx++)
        {
            yy = ( i * Byte_size ) + xx;
            for(j=0;j < 8;j++)
            {
                if( ( ( xx * 8 ) + j ) < Curr_width )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if( ( (DataFont[inchar][yy] >> ( 7-j ) ) &1 ) )
        EditValue[i][7-(7-j)+(xx*8)] = 1;
    else
        EditValue[i][7-(7-j)+(xx*8)] = 0;
    } /* If < Curr_width */
} /* for j */
} /* xx */
} /* i */

```

```

CheckEdit();

```

```

if(level == 0)

```

```

{ y1 = 0;

```

```

  y2 = 8;

```

```

}

```

```

if(level' == 1)

```

```

{

```

```

  y1 = 8;

```

```

  if( Curr_high < 16)

```

```

    y2 = Curr_high + 1;

```

```

  else

```

```

    y2 = 16;

```

```

}

```

```

if(level == 2)

```

```

{

```

```

  y1 = 16;

```

```

  if( Curr_high <= 24)

```

```

    y2 = Curr_high + 1;

```

```

  else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        y2 = 24;
    }

/*
y1 = (level - 1) * 8;
y2 = y1 + 8;
if( y2 > Curr_high )
    y2 = ( Curr_high - y1 ) + y1;
*/

for(ix=0;ix<Curr_width;ix++)
{
    byte_val = 0;
    kk = 0;
    for(iy=y1 ; iy<y2 ;iy++)
    {
        if(EditValue[iy][ix]==1)
        {
            byte_val = byte_val + bit_val[kk];
        }
        kk++;
    }
    CopyByte(byte_val);
    /* delay(500); */
}

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*-----*/
/* Print File                               */
/* Update : 6/11/1996                      */
/* Test : .....                          */
/*-----*/

```

```
PrintFile()
```

```
{
```

```
int ix,miy,level,ik,im,chridx;
```

```
level = 0;
```

```
ik = 0;
```

```
level = Curr_high / 8;
```

```
ik = Curr_high % 8;
```

```
if(ik > 0)
```

```
level = level + 1;
```

```
if(level < 1)
```

```
level = 1;
```

```
chridx = 0;
```

```
ik = 0;
```

```
for(miy=0;miy<16;miy++)
```

```
{
```

```
ik = miy * 16 ;
```

```
for(im=0;im<level; im++)
```

```
{
```

```
InitPrinter();
```

```
chridx = 0;
```

```
for(ix=0; ix< 16 ;ix++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    chridx = ix + ik;
    PrintChar(chridx,im);
}
LineFeed();

```

```

LineSkip();

```

```

}

```

```

}

```

```

□

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/
/* Program : FONTUTL.C */
/* Descrip : Program Font Tools Utility */
/* Listing */
/* RightMirror() */
/* DownMirror() */
/* InsRow() */
/* DelRow() */
/* InsCol() */
/* InsCol() */
/* ExpandChar() */
/* ----- */
/* ----- */
/* Down Mirror Character */
/* Update : 6/07/1996 */
/* Test : Complete */
/* ----- */
DownMirror()
{
    int Start_row,End_row;
    int st1;
    int ik,ix,iy;
    int ij,x1,x2,y1,y2;

    for(iy=0;iy< Curr_high;iy++)
        for(ix=0;ix<Curr_width;ix++)
            TempValue[iy][ix] = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Start_row = Curr_Erow + 1;
```

```
for(iy=0;iy< Start_row;iy++)
```

```
    for(ix=0;ix<Curr_width;ix++)
```

```
    {
```

```
        TempValue[iy][ix] = EditValue[iy][ix];
```

```
    }
```

```
if((Start_row * 2) < Curr_high)
```

```
    End_row = Start_row * 2;
```

```
else
```

```
    End_row = Curr_high;
```

```
ik = Start_row;
```

```
for(iy=Start_row;iy < End_row;iy++)
```

```
    for(ix=0;ix<Curr_width;ix++)
```

```
    {
```

```
        EditValue[iy][ix] = TempValue[End_row - iy -1 ][ix];
```

```
    }
```

```
RefreshChar();
```

```
}
```

```
/*-----*/
```

```
/* Insert Current Row */
```

```
/* Update : 8/07/1996 */
```

```
/* Test : Complete */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-----*/
```

```
InsRow()
```

```
{
```

```
int Start_row,Begin_row;
```

```
int st1;
```

```
int ik,ix,iy,dist;
```

```
int i,j,x1,x2,y1,y2;
```

```
for(iy=0;iy<Curr_high;iy++)
```

```
for(ix=0;ix<Curr_width;ix++)
```

```
TempValue[iy][ix] = EditValue[iy][ix];
```

```
Start_row = Curr_Erow;
```

```
for(iy=Start_row;iy < Curr_high - 1;iy++)
```

```
for(ix=0;ix<Curr_width;ix++)
```

```
{
```

```
EditValue[iy+1][ix] = TempValue[iy][ix];
```

```
}
```

```
RefreshChar();
```

```
}
```

```
/*-----*/
```

```
/* Delete Current Row */
```

```
/* Update : 8/07/1996 */
```

```
/* Test : Complete */
```

```
/*-----*/
```

```
DelRow()
```

```
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int Start_row,Begin_row;

int st1;

int ik,ix,iy,dist;

int i,j,x1,x2,y1,y2;


for(iy=0;iy<Curr_high;iy++)
    for(ix=0;ix<Curr_width;ix++)
        TempValue[iy][ix] = EditValue[iy][ix];


Start_row = Curr_Erow;


for(iy=Start_row;iy < Curr_high - 1;iy++)
    for(ix=0;ix<Curr_width;ix++)
    {
        EditValue[iy][ix] = TempValue[iy+1][ix];
    }


RefreshChar();
}


/*-----*/
/* Delete Current Column */
/* Update : 12/07/1996 */
/* Test : Complete */
/*-----*/

DelCol()
{
    int Start_row,Begin_row;

    int Start_col,Begin_col;

    int st1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int ik,ix,iy,dist;
```

```
int i,j,x1,x2,y1,y2;
```

```
for(iy=0;iy<Curr_high;iy++)
```

```
    for(ix=0;ix<Curr_width;ix++)
```

```
        TempValue[iy][ix] = EditValue[iy][ix];
```

```
Start_col = Curr_Ecol;
```

```
for(iy=0;iy < Curr_high;iy++)
```

```
    for(ix=Start_col;ix < Curr_width -1;ix++)
```

```
    {
```

```
        EditValue[iy][ix] = TempValue[iy][ix+1];
```

```
    }
```

```
/* Clear Last Column */
```

```
for(iy=0;iy<Curr_high;iy++)
```

```
    EditValue[iy][Curr_width - 1] = 0;
```

```
RefreshChar();
```

```
}
```

```
/*-----*/
```

```
/* Insert Current Column */
```

```
/* Update : 12/07/1996 */
```

```
/* Test : Complete */
```

```
/*-----*/
```

```
InsCol()
```

```
{
```

```
    int Start_row,Begin_row;
```

```
    int Start_col,Begin_col;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int st1;  
int ik,ix,iy,dist;  
int i,j,x1,x2,y1,y2;
```

```
for(iy=0;iy<Curr_high;iy++)  
    for(ix=0;ix<Curr_width;ix++)  
        TempValue[iy][ix] = EditValue[iy][ix];
```

```
Start_col = Curr_Ecol;
```

```
for(iy=0;iy < Curr_high;iy++)  
    for(ix=Start_col;ix < Curr_width -1;ix++)  
    {  
        EditValue[iy][ix+1] = TempValue[iy][ix];  
    }
```

```
RefreshChar();  
}
```

```
/*-----*/  
/* Right Mirror Datat Current Column */  
/* Update : 15/07/1996 */  
/* Test : Complete */  
/*-----*/
```

```
RightMirror()
```

```
{  
    int Start_col,End_col;  
    int st1;  
    int ik,ix,iy;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i,j,x1,x2,y1,y2;
```

```
for(iy=0;iy< Curr_high;iy++)  
    for(ix=0;ix<Curr_width;ix++)  
        TempValue[iy][ix] = 0;
```

```
Start_col = Curr_Ecol + 1;
```

```
for(iy=0;iy< Curr_high;iy++)  
    for(ix=0;ix<Start_col;ix++)  
    {  
        TempValue[iy][ix] = EditValue[iy][ix];  
    }
```

```
if((Start_col * 2) < Curr_width)  
    End_col = Start_col * 2;  
else  
    End_col = Curr_width;
```

```
ik = Start_col;  
for(iy=0;iy < Curr_high;iy++)  
    for(ix=Start_col;ix<End_col;ix++)  
    {  
        EditValue[iy][ix] = TempValue[iy][End_col - ix - 1];  
    }
```

```
RefreshChar();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/*-----*/
/* Expand From Center */
/* Update : 15/07/1996 */
/* Test : Complete */
/*-----*/

ExpandChar()
{
    Curr_Ecol = Curr_width / 2;
    Curr_Erow = Curr_high / 2;
    InsRow();
    InsCol();
    Curr_Ecol = 0;
    Curr_Erow = 0;
    DelRow();
    DelCol();
    /* Reset */
    Curr_Ecol = Curr_width / 2;
    Curr_Erow = Curr_high / 2;
}

/*-----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/* Program : FONTSCR.C                               */
/* Descrip : Program Screen Function Library          */
/* Listing                                           */
/* -----*/
/* Open_Graph()                                     */
/* Close_Graph()                                    */
/* Readkey()                                         */
/* Goto_X(grid_x)                                    */
/* Goto_Y(grid_y)                                    */
/* int Bottom_Y(grid_y)                             */
/* int Right_X(grid_x)                              */
/* int Top_Y(grid_y)                                */
/* int Left_X(grid_x)                               */
/* Border(Start_x,Start_y,End_x,End_y)              */
/* Revers_Bar(grid_x,grid_y,column_width)           */
/* Normal_Bar(grid_x,grid_y,column_width)           */
/* Revers_Text()                                    */
/* Normal_Text()                                    */
/* Printxy(int x,int y,unsigned char St[])           */
/* PrintDataxy(int x,int y,unsigned char St[])       */
/* Active_Border(int act)                           */
/* ClearEditWin()                                    */
/* SolidBorder(Start_x,Start_y,End_x,End_y,color)    */
/* ReadString(int x,int y,int color,int length,      */
/*      unsigned char *strin,char *Msgdata)         */
/* Message(char *Msgdata)                           */
/* ReadIntData(int x,int y,int length,char *Msgdata) */
/* Init_Table()                                      */
/*-----*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*=====*/
/
Open_Graph()
{
int driver = VGA, crtmode = VGAHI , ix , iy ;

initgraph(&driver , &crtmode , "");
if ( graphresult() == grOk )
{
setfillstyle(SOLID_FILL,BAR_NOR_C);
bar(0,0,639,479);
Normal_Text(); /* Initial Text color */
return(1);
}
else
return(0);
}

Close_Graph()
{
closegraph();
}

Readkey()
{
char key,ext;

/* Normal KeyBoard Keep in Variable (Key) */
/* Extended KeyBoard Keep in Variable (Ext) */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

key = getch();
if ( key == 0 )
{
    ext = getch();
    return(ext);
}
else
    return(key);
}

```

```

Goto_X(grid_x)
int grid_x;
{
    int point_x;

    if ( grid_x == 0 )
        grid_x = 1;
    if ( grid_x >= MAX_COLUM)
        grid_x = MAX_COLUM;
    point_x = grid_x * SYS_FONT_W;
    return(point_x);
}

```

```

Goto_Y(grid_y)
int grid_y;
{
    int point_y;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ( grid_y  $\neq$  0 )
    grid_y = 1;
if ( grid_y >= MAX_ROW)
    grid_y = MAX_ROW;
point_y = grid_y * SYS_FONT_H;
return(point_y);
}

```

```

int Bottom_Y(grid_y)
int grid_y;
{
    int bottom_point;
    bottom_point = Goto_Y(grid_y) + ( SYS_FONT_H / 2 );
    return(bottom_point);
}

```

```

int Right_X(grid_x)
int grid_x;
{
    int right_point;
    right_point = Goto_X(grid_x) + ( SYS_FONT_W / 2 );
    return(right_point);
}

```

```

int Top_Y(grid_y)
int grid_y;
{
    int top_point;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

top_point = Goto_Y(grid_y) + ( SYS_FONT_H / 2 );
return(top_point);
}

```

```

int Left_X(grid_x)
int grid_x;
{
    int left_point;
    left_point = Goto_X(grid_x) + ( SYS_FONT_W / 2 );
    return(left_point);
}

```

```

Border(Start_x,Start_y,End_x,End_y)
int Start_x,Start_y,End_x,End_y;
{
    int left,top,right,bottom;
    int side_shift,down_shift;

    side_shift = ( SYS_FONT_W / 5 ) * 2;
    down_shift = ( SYS_FONT_H / 5 ) * 2;

    left  = Goto_X(Start_x + 1 ) - side_shift;
    right = Goto_X(End_x ) + side_shift ;
    top   = Goto_Y(Start_y + 1 ) - down_shift;
    bottom = Goto_Y(End_y ) + down_shift;

    rectangle( left,top,right,bottom );
    Normal_Text();
}

```

```

Revers_Bar(grid_x,grid_y,column_width)
int grid_x,grid_y,column_width;
{
    int left,top,right,bottom;
    int side_shift,down_shift;
    int x1,x2,y1,y2;

    x1 = grid_x;
    x2 = grid_x + column_width;
    y1 = grid_y;
    y2 = grid_y + 1;
    /*
    side_shift = ( SYS_FONT_W / 5 ) * 2;
    down_shift = ( SYS_FONT_H / 5 ) * 2;

    left  = Goto_X(x1 + 1 ) - side_shift + 2;
    right = Goto_X(x2 ) + side_shift - 2 ;
    top   = Goto_Y(y1 ) - down_shift + 2 ;
    bottom = Goto_Y(y2 ) + down_shift - 2 ;
    */

    left  = Goto_X(x1);
    right = Goto_X(x2 + 1 );
    top   = Goto_Y(y1);
    bottom = Goto_Y(y2);

    setfillstyle(SOLID_FILL,BAR_REV_C);
    bar(left,top,right,bottom);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Normal_Bar(grid_x,grid_y,column_width)

int grid_x,grid_y,column_width;

{

int left,top,right,bottom;

int side_shift,down_shift;

int x1,x2,y1,y2;

x1 = grid_x;

x2 = grid_x + column_width;

y1 = grid_y;

y2 = grid_y + 1;

/*

side_shift = (SYS_FONT_W / 5) * 2;

down_shift = (SYS_FONT_H / 5) * 2;

left = Goto_X(x1 + 1) - side_shift + 2;

right = Goto_X(x2) + side_shift - 2 ;

top = Goto_Y(y1) - down_shift + 2 ;

bottom = Goto_Y(y2) + down_shift - 2 ;

*/

left = Goto_X(x1);

right = Goto_X(x2 + 1);

top = Goto_Y(y1);

bottom = Goto_Y(y2);

setfillstyle(SOLID_FILL,BAR_NOR_C);

bar(left,top,right,bottom);

}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Revers_Text()
```

```
{  
    Curr_FG = TEXT_RFG;  
    Curr_BG = TEXT_RBG;  
}
```

```
Normal_Text()
```

```
{  
    Curr_FG = TEXT_NFG;  
    Curr_BG = TEXT_NBG;  
}
```

```
Printxy(int x,int y,unsigned char St[])
```

```
{  
    int count=0;  
    int a=0;  
    int test;  
    int ixrow,ixcol;  
    int color;  
  
    color = TEXT_NOR_C;  
    ixrow = Goto_Y(y);  
    ixcol = Goto_X(x);  
  
    while((St[count] != 0) & (count<128))  
    {  
        test = (int)St[count];  
        if( (test==209) ||
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

((test>211)&&(test<220)) ||
((test>230)&&(test<239)) )
DrawSysFont(ixcol+a-SYS_FONT_W,ixrow,test,0);
else
{
DrawSysFont(ixcol+a, ixrow,test,0);
a+=SYS_FONT_W;
}
count++;
}
}

```

```

PrintDataxy(int x,int y,unsigned char St[])

```

```

{
int count=0;
int a=0;
int test;
int ixrow,ixcol;
int color;

color = TEXT_NOR_C;
ixrow = Goto_Y(y);
ixcol = Goto_X(x);

```

```

while((St[count] != 0) & (count<128))

```

```

{
test = (int)St[count];
if( (test==209) ||
((test>211)&&(test<220)) ||
((test>230)&&(test<239)) )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DrawDataFont(ixcol+a-SYS_FONT_W,ixrow,test,0);

else
{
DrawDataFont(ixcol+a, ixrow,test,0);
a+=SYS_FONT_W;
}

count++;
}
}

```

```

Active_Border(int act)
{
setcolor(BORDER_NOR);
Border(0,0,79,2);          /* Main Menu Border */
Border(62,2,79,19);        /* Border Of Font View */
Border(0,2,62,23);          /* Border Of Editor */
Border(62,19,79,23);        /* Border Of Status */
switch(act)
{
case 1:
{
setcolor(BORDER_ACT);
Border(0,0,79,2);          /* Main Menu Border */
break;
}
case 2:
{
setcolor(BORDER_ACT);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Border(62,2,79,19);      /* Border Of Font View */
    break;
}
case 3:
{
    setcolor(BORDER_ACT);
    Border(0,2,62,23);      /* Border Of Editor */
    break;
}
case 4:
{
    setcolor(BORDER_ACT);
    Border(62,19,79,23);    /* Border Of Status */
    break;
}
}
}

ClearEditWin()
{
    SolidBorder(1,3,61,22,BAR_NOR_C);    /* Border Of Editor */
}

```

SolidBorder(Start_x,Start_y,End_x,End_y,color)

int Start_x,Start_y,End_x,End_y,color;

```

{
    int left,top,right,bottom;
    int side_shift,down_shift;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

side_shift = ( SYS_FONT_W / 5 ) * 2;
down_shift = ( SYS_FONT_H / 5 ) * 2;

left  = Goto_X(Start_x + 1 ) - side_shift;
right = Goto_X(End_x ) + side_shift ;
top   = Goto_Y(Start_y + 1 ) - down_shift;
bottom = Goto_Y(End_y ) + down_shift;

setfillstyle(SOLID_FILL,color);
bar( left,top,right,bottom );
}

```

```

ReadString(int x,int y,int color,int length,
           unsigned char *strin,char *Msgdata)
{
    unsigned char oldchar[80],dummy[80],
    dummy2[10],readin,subreadin;
    int      cnt,a,b,xx,xy,slong,curinv;
    char far  *bkarea;
    int      THAI = 0;
    void* Scrtxtsave;
    int  size,Msgwidth;
    int  x1,y1,x2,y2;

```

```

xx = 0;
cnt = 0;
Msgwidth = 10;

```

```

x1 = Goto_X(x - 1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

y1 = Goto_Y(y - 1);
x2 = Goto_X(x + length + 3 + Msgwidth );
y2 = Goto_Y(y + 3);
size = imagesize(x1, y1, x2, y2);
Scrxtsave = malloc(size);
getimage(x1, y1, x2, y2, Scrxtsave);

```

```

Revers_Bar(x + 1 + Msgwidth,y + 1 ,length );

```

```

setcolor(BORDER_ACT);
Border(x,y,x + length + 2 + Msgwidth ,y+2);
Printxy(x + 1,y + 1, Msgdata );

```

```

while(cnt < length + 1 )

```

```

{
    strin[cnt] = 0x00;
    cnt++;
}

```

```

do

```

```

{
    Revers_Bar(x + 1 + Msgwidth,y + 1, length );
    Printxy(x + 1 + Msgwidth ,y + 1, strin );

```

```

curinv = 0;

```

```

for(a=0;a<xx;a++)

```

```

    if((*strin+a)==209) ||

```

```

    ((*strin+a) > 211) && ((*strin+a) < 220)) ||

```

```

    ((*strin+a) > 230) && ((*strin+a) < 239)))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    curinv++;

curinv = xx - curinv + Msgwidth;

setcolor(WHITE);
setfillstyle(SOLID_FILL,YELLOW);
if (THAI)
bar(Goto_X( x + 1 + curinv ) , Goto_Y( y + 1 ),
    Goto_X( x + 2 + curinv ) , Goto_Y( y + 2 ) );
else
bar(Goto_X( x + 1 + curinv ) , Goto_Y( y + 2 ) - ( Curr_high / 4 ),
    Goto_X( x + 2 + curinv ) , Goto_Y( y + 2 ) );
readin = Readkey();
switch(readin)
{
case Left_key :
{
    if(xx>0)
        xx--;
    break;
}
case Right_key :
{
    if(xx<length)
        xx++;
    break;
}
case Esc_key :
{
    for(a=0;a<length;a++)
        *(strin+a) = 0x00;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        readin = Return_key;

        break;
    }

    case Backspace_key :
    {
        if (xx>0)
            xx--;

        for (a=xx;a <length -1;a++)
        {
            *(strin+a)= *(strin+a+1);
            *(strin+length-1) = 0x0;
        }

        break;
    }

    case '^' :
    {
        THAI='!THAI;
        break;
    }

} /* Switch */

if((readin>=0x20)&(readin!=0x60))
{
    if(THAI)
        readin = SMOTable[readin-0x20];

    /*
    for(a=(length-2);a>=xx;a--)

        *(strin+1)= *(strin+a);

    */

    for(a=0;a<xx;a++)
        if(*(strin+a) < 0x20)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
*(strin+a) =0x20;
```

```
*(strin+xx) = readin;
```

```
if(xx<(length-1))
```

```
xx++;
```

```
}
```

```
} while(readin!=Return_key);
```

```
putimage(x1, y1, Sctrxtsave , COPY_PUT );
```

```
free(Sctrxtsave);
```

```
return 1;
```

```
}
```

```
Message(char *Msgdata)
```

```
{
```

```
void* Msgsave;
```

```
int size;
```

```
int x1,y1,x2,y2;
```

```
x1 = Goto_X(10);
```

```
y1 = Goto_Y(9);
```

```
x2 = Goto_X(51);
```

```
y2 = Goto_Y(12);
```

```
size = imagesize(x1, y1, x2, y2);
```

```
Msgsave = malloc(size);
```

```
getimage(x1, y1, x2, y2, Msgsave);
```

```
Revers_Bar(x1,y1,38);
```

```
setcolor(BORDER_ACT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Border(10,9,50,11);

Printxy( 11,10,Msgdata );


getch();

putimage(x1,y1, Msgsave ,COPY_PUT );

free(Msgsave);

}

ReadIntData(int x,int y,int length,char *Msgdata)
{
    char readin;
    char strin[10];
    int  cnt,a,b,xx,xy,slong,curinv;
    void* Scrxtsave;
    int  size,Msgwidth;
    int  x1,y1,x2,y2;
    int  idata;


    xx = 0;
    cnt = 0;
    Msgwidth = 10;


    x1 = Goto_X(x - 1);
    y1 = Goto_Y(y - 1);
    x2 = Goto_X(x + length + 3 + Msgwidth );
    y2 = Goto_Y(y + 3);
    size = image_size(x1, y1, x2, y2);
    Scrxtsave = malloc(size);
    getimage(x1, y1, x2, y2, Scrxtsave);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Revers_Bar(x + 1 + Msgwidth,y + 1 ,length );
```

```
setcolor(BORDER_ACT);
```

```
Border(x,y,x + length + 2 + Msgwidth ,y+2);
```

```
Printxy(x + 1,y + 1, Msgdata );
```

```
while(cnt < length + 1 )
```

```
{  
    strin[cnt] = 0x00;  
    cnt++;  
}
```

```
do
```

```
{  
    Revers_Bar(x + 1 + Msgwidth,y + 1, length );  
    Printxy(x + 1 + Msgwidth ,y + 1, strin );
```

```
    curinv = 0;
```

```
    for(a=0;a<xx;a++)
```

```
        if((*strin+a)==209) ||
```

```
        ((*strin+a) > 211) && ((*strin+a) < 220)) ||
```

```
        ((*strin+a) > 230) && ((*strin+a) < 239)))
```

```
            curinv++;
```

```
    curinv = xx - curinv + Msgwidth;
```

```
    readin = Readkey();
```

```
    switch(readin)
```

```
    {
```

```
        case Esc_key :
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    {
        for(a=0;a<length;a++)
            *(strin+a) = 0x00;
        readin = Return_key;
        break;
    }

```

case Backspace_key :

```

    {
        if (xx>0)
            xx--;
        for (a=xx;a <length -1;a++)
        {
            *(strin+a)= *(strin+a+1);
            *(strin+length-1) = 0x0;
        }
        break;
    }

```

} /* Switch */

```

    if((readin>=0x30)&(readin<=0x39))
    {

```

```

        for(a=0;a<xx;a++)
            if(*(strin+a) < 0x20)
                *(strin+a) =0x20;

```

```

        *(strin+xx) = readin;

```

```

        if(xx<(length-1))

```

```

            xx++;

```

```

    }

```

```

} while(readin!=Return_key);

```

```

idata = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        idata = atoi(strin);

        putimage(x1, y1, Sctrxtsave , COPY_PUT );
        free(Sctrxtsave);
        return(idata);
    }

```

Init_Table()

```

{
    int ix,iy,mx,my;

    ix = 0;
    iy = 0;

    setcolor(GREEN);
    for (my=0 ; my < Curr_high ; my++)
    {
        for (mx=0 ; mx < Curr_width ; mx++)
        {
            iy = Conner_Row + (my * POINT_WIDTH);
            ix = Conner_Col + (mx * POINT_WIDTH);
            rectangle(ix,iy,ix + POINT_WIDTH ,iy + POINT_WIDTH );
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

