สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบช่วยในการสำรวจหนังสือของสำนักหอสมุดกลาง

HANDY BOOK CHECKING

T037061

โดย

นาย พิพัฒน์ จินตามฤทธิ์

นาย มงคลเดช ราชภักดี

อาจารย์ที่ปรึกษา

อาจารย์ ธนา หงษ์สุวรรณ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 37061
วัน, เดือน, ปี 31 ส.ค. 2543

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบช่วยในการสำหนังสือของสำนักหอสมุดกลาง

HANDY BOOK CHECKING

ผู้จัดทำ

1. นาย พิพัฒน์ จินตามฤทธิ์ รหัส 39014368
2. นาย มงคลเดช ราชภักดี รหัส 39014396

อาจารย์ที่ปรึกษา

(อาจารย์ ธนา หงษ์สุวรรณ)

ระบบช่วยในการสำรวจหนังสือของสำนักหอสมุดกลาง

นายพิพัฒน์ จินตามฤทธิ์ รหัส 39014368
นาย มงคลเดช ราชภักดี รหัส 39014396
อาจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษา
ปีการศึกษา 2542

บทคัดย่อ

ในปัจจุบันการสำรวจหนังสือของสำนักหอสมุดกลางนั้นจะทำโดยการนำคอมพิวเตอร์เข้าไปสำรวจหนังสือตามชั้นหนังสือทีละชั้นโดยใช้เครื่องอ่านบาร์โค้ด จนครบทุกเล่ม ซึ่งเป็นวิธีการที่ยุ่งยากมากเพราะต้องต่อสายไฟฟ้าเข้าไปตามชั้นหนังสือที่เราจะเข้าไปสำรวจ และไม่สามารถกระทำการสำรวจหนังสือไปพร้อมๆกันทีละหลายๆเครื่องได้ จึงทำให้ต้องเสียเวลาในการสำรวจหนังสือในแต่ละปีเป็นเวลานานมาก จนถึงกับต้องปิดบริการห้องสมุดเป็นเวลาหลายวัน

โครงงานนี้เป็นโครงงานที่เกิดขึ้นมาจากความต้องการใช้งานของผู้ใช้ เพื่อให้การสำรวจหนังสือเป็นไปโดยสะดวก รวดเร็ว โดยการสร้างเครื่องสำรวจหนังสือโดยใช้ไมโครคอนโทรลเลอร์ 8051 เพื่อเก็บข้อมูลจากเครื่องอ่านบาร์โค้ด แล้วนำมาตรวจสอบโดยแอปพลิเคชันโปรแกรมซึ่งเขียนโดยใช้ เดลไฟล์ (Delphi4) เพื่อตรวจสอบหนังสือที่สำรวจไม่พบ โดยการรับข้อมูลจากเครื่องสำรวจหนังสือมาตรวจสอบกับข้อมูลในฐานข้อมูล และแสดงรายงานการสำรวจหนังสือออกด้วย

HANDY BOOK CHECKING

Pipat Jintamalit

Mongkoldech Lajapakdee

Assoc. Prof. Thana Hongsuwan

ABSTRACT

Once a year, all of the books in the central library of KMITL have to be checked to see how many of them are vanished from the library. To check the books the staffs have to carry a computer and a barcode reader along to the book shelves to read the barcode on every single book, which causes some difficulties due to the wiring problem and the fact that the staffs can use only one set of machine at a time, therefor it takes an unnecessarily long time to complete the task, and sometimes causes the library to be closed for several days.

This project is originated from user requirement to eliminate the difficulties mentioned. Therefor a handy book checker is designed for the project, by using an MCS-51 microcontroller to store data from the barcode reader. These data are then processed by an application program written in Delphi 4 to check for vanished books. The data from the book checker are compared to the data in the database, and after these processes are done the book checking report is built.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้คงไม่อาจเสร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือ และร่วมมือจากหลาย ๆ ฝ่ายด้วยกัน บุคคลแรกที่ต้องกล่าวถึงเพราะเป็นส่วนสำคัญที่ทำให้วิทยานิพนธ์นี้เสร็จลงได้ก็คือ อาจารย์ ธนา หงษ์สุวรรณ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่ให้ความเอาใจใส่ แนะนำ และช่วยเหลือเสมอมา ซึ่งต้องขอขอบพระคุณเป็นอย่างมาก

ขอขอบคุณเจ้าหน้าที่ห้องสมุดที่ให้ความร่วมมือในการสอบถามข้อมูลของทางห้องสมุด และให้ยืมเครื่องยิงบาร์โค้ดมาใช้ทดลอง

ขอขอบคุณพี่ๆห้อง ESL ที่ช่วยตอบคำถามข้อข้องใจทางด้านฮาร์ดแวร์มาโดยตลอด

ขอขอบคุณทุกๆคนที่ช่วยพิมพ์รายงานฉบับนี้

ขอขอบคุณเพื่อนๆทุกคนที่คอยเป็นกำลังใจให้กันและกันโดยตลอดมาในยามที่ท้อใจกับการทำโปรเจค

สุดท้ายนี้ขอพระคุณอย่างสูง สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ได้พบกับสิ่งที่ดีๆในชีวิต

พิพัฒน์ จินตามฤทธิ์

มงคลเดช ราชภักดี

สารบัญ

หน้า

บทที่1 บทนำ

ความเป็นมาของโครงงาน 1
วัตถุประสงค์ของโครงงาน 1
ขอบเขตของโครงงาน 1
วิธีการดำเนินงาน 1

บทที่ 2 ทฤษฎีและหลักการ

ไมโครคอนโทรลเลอร์ตระกูล 8051 3
ลักษณะพื้นฐานของ 8051 3
หน่วยความจำโปรแกรมของ 8051 5
หน่วยความจำข้อมูลของ 8051 6
พอร์ตอินพุตและเอาต์พุต 15
รหัสแถบ 17
หลักการของรหัสแถบ 17
ชนิดของรหัสแถบ 18
ลักษณะของรหัสแถบที่ดี 32
เครื่องอ่านรหัสแถบชนิดไม่สัมผัสกับรหัสแถบ (Non-contact Scanner) 32
การประยุกต์ใช้ส่วนแสดงผลชนิด LCD module กับ MCS-51 33
คำสั่งที่ใช้ควบคุมการทำงาน 36
รายละเอียดของคำสั่ง HD44780 38
ทฤษีการออกแบบฐานข้อมูลแบบสัมพันธ์โดยวิธีไนแอม (The NIAM Conceptual Schema Model) 42
การออกแบบฐานข้อมูลด้วยวิธีไนแอม 42
ส่วนประกอบพื้นฐานของไนแอม 42

บทที่ 3 การออกแบบ

ปัญหาของระบบเดิม 44
ความต้องการของระบบ 44
ภาพรวมของระบบ 44
ส่วนของฮาร์ดแวร์ 45
ส่วนประกอบของฮาร์ดแวร์ 45
การรับข้อมูลจากเครื่องอ่านบาร์โค้ด 46

การแปลงค่าแสกนโค้ดเป็นแอสกี้โค้ด 46
การนำค่าแอสกี้โค้ดที่ได้ไปเก็บไว้ในหน่วยความจำ 47
การแสดงผลบนจอ LCD 47
การโอนถ่ายข้อมูลไปยังเครื่องคอมพิวเตอร์ 48
ความสามารถในการเก็บค่าในเมโมรี่เมื่อหยุดจ่ายไฟเลี้ยง 48
การเคลียร์ข้อมูลทั้งหมดในหน่วยความจำ 48
ความสามารถในการใช้แหล่งจ่ายไฟจากแบตเตอรี่ 12 โวลท์ 49
วงจรรวมของฮาร์ดแวร์ 49
ส่วนของแอปพลิเคชั่นโปรแกรม 49
อุปกรณ์ที่ใช้ในการพัฒนาซอฟทแวร์ของระบบ 49
ความต้องการของระบบ 50
การออกแบบฐานข้อมูล 50
ตารางของฐานข้อมูลที่มีการใช้งานที่ใช้งาน 51
ฟังก์ชันการทำงานของซอฟทแวร์ 53

บทที่ 4 การใช้งาน

การใช้งานในส่วนของฮาร์ดแวร์ 59
การเริ่มต้นการใช้งาน 59
การโอนถ่ายข้อมูลไปยังเครื่องคอมพิวเตอร์ 59
การเคลียร์ค่าในหน่วยความจำ 59
การเลิกใช้งาน 59
การใช้งานในส่วนของซอฟท์แวร์ 60
การรับข้อมูลและการตรวจสอบข้อมูลจากเครื่องสำรวจหนังสือ 60
การค้นหาข้อมูลและการแก้ไขข้อมูล 60
การแสดงรายงานการสำรวจหนังสือ 61
การอัพเดตฐานข้อมูลที่ใช้ในงานสำรวจหนังสือ 61

บทที่ 5 บทสรุปและวิจารณ์

ข้อดีของระบบ 62
ข้อจำกัดของระบบที่พัฒนา 62
ปัญหาที่พบในการทำโครงงาน 62
บทวิจารณ์ผล 63
ข้อเสนอแนะ 63

ภาคผนวก ก. 64

ภาคผนวก ข. 70

ภาคผนวก ค. 72

บรรณานุกรม 78

รูปที่ 3.7 ออกแบบฐานข้อมูลโดยวิธีไนแอม 51
รูปที่ 3.8 ฐานข้อมูลของระบบที่ได้ทำการออกแบบ ใช้กับการสำรวจหนังสือ 52
รูปที่ 3.9 ส่วนติดต่อผู้ใช้ หน้าจอหลัก 53
รูปที่ 3.10 ส่วนติดต่อผู้ใช้ ส่วนแสดงข้อมูลในฐานข้อมูล 54
รูปที่ 3.11 ส่วนติดต่อผู้ใช้ การค้นหาข้อมูลในฐานข้อมูล 55
รูปที่ 3.12 ส่วนติดต่อผู้ใช้ การแสดงรายงานหนังสือที่สำรวจไม่พบ 56
รูปที่ 3.13 ส่วนติดต่อผู้ใช้ การอัพเดตฐานข้อมูล 57
รูปที่ 3.14 ส่วนติดต่อผู้ใช้ การรับข้อมูลจากเครื่องสำรวจหนังสือ 58

สารบัญตาราง

ตารางที่ 2.1 ตารางรหัสเลขฐานสองของรหัส 2 ใน 5 ทั้งสองแบบ 19
ตารางที่ 2.2 ตารางแสดงเลขฐานสองของรหัสแถบ 3 ใน 9 21
ตารางที่ 2.3 ตารางแสดงการแทนรหัสและค่าเลขประจำตัวของรหัส 39 23
ตารางที่ 2.4 ตารางแสดงเลขฐานสองของรหัสโคดาบาร์ 25
ตารางที่ 2.5 แสดงการแทนรหัสแถบยูพีซี 27
ตารางที่ 2.6 ตารางแสดงการแทนตัวเลขบอกลักษณะงาน 27
ตารางที่ 2.7 แสดงการจัดเรียงของรหัส 28
ตารางที่ 2.8 ตารางแสดงอักขระของรหัสเอียน 29
ตารางที่ 2.9 ตารางแสดงการจัดเรียงของข้อมูลเพื่อหาค่ารหัสเติมหน้า 31
ตารางที่ 2.10 แสดงการทำงานของสัญญาณ E 36
ตารางที่ 2.11 ตารางคำสั่งที่ใช้ควบคุมกรทำงานชิปคอนโทรลเลอร์ HD44780 37
ตารางที่ 3.1 ตารางแสดงลักษณะของฟิลด์ข้อมูล 52

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่1
บทนำ

ความเป็นมาของโครงงาน

เนื่องจากระบบสำรวจหนังสือประจำปีของหอสมุดกลางของสถาบันเราในปัจจุบันนี้ จะทำโดยการนำคอมพิวเตอร์เข้าไปสำรวจหนังสือตามชั้นหนังสือทีละชั้นจนครบทุกเล่ม ซึ่งเป็นวิธีการที่ยุ่งยากมาก เพราะต้องต่อสายไฟฟ้าเข้าไปตามชั้นหนังสือที่เราจะเข้าไปสำรวจ และไม่สามารถกระทำการสำรวจหนังสือไปพร้อมๆกันทีละหลายๆเครื่องได้ จึงทำให้ต้องเสียเวลาในการสำรวจหนังสือในแต่ละปีเป็นเวลานานมาก จนถึงกับต้องปิดบริการห้องสมุดเป็นเวลาหลายวัน

โครงงานชิ้นนี้จึงจัดทำขึ้นมาเพื่อช่วยให้ระบบสำรวจหนังสือมีความสะดวกสบายมากขึ้น และใช้เวลาในการสำรวจน้อยลง โดยการสร้างอุปกรณ์ที่ช่วยในการสำรวจหนังสือขึ้นมา ซึ่งทำหน้าที่เป็นหน่วยความจำชั่วคราวในการเก็บบาร์โค้ดของหนังสือ โดยเป็นอุปกรณ์แบบพกพา มีแหล่งจ่ายพลังงานในตัว และสามารถทำการสำรวจหนังสือไปพร้อมๆกันทีละหลายๆเครื่องได้

วัตถุประสงค์ของโครงงาน

- เพื่อพัฒนาระบบสำรวจหนังสือของห้องสมุด
- เพื่อช่วยให้การสำรวจหนังสือของห้องสมุดมีความสะดวกมากขึ้น
- เพื่อให้สามารถนำไปประยุกต์ใช้กับงานอื่นๆที่มีลักษณะคล้ายกันได้ เช่นการตรวจเช็คสินค้าคงเหลือของห้างสรรพสินค้า

ขอบเขตของโครงงาน

สำหรับขอบเขตของโครงงานที่ได้พัฒนา เพื่อให้ระบบสามารถทำงานได้จริงในการทำงานต่างๆที่ได้ออกแบบไว้ ซึ่งการทำงานดังกล่าว ได้แก่

1. การสำรวจหนังสือของหอสมุดกลางของสถาบันในแต่ละปี
2. การให้ข้อมูลของการสำรวจหนังสือ และการจัดการกับฐานข้อมูลของหอสมุดกลาง
3. ติดต่อฐานข้อมูลและ แก้ไขข้อมูลต่างๆ ได้อย่างถูกต้อง
4. สามารถนำระบบไปประยุกต์ใช้งานทางด้านอื่นๆ ต่อไปได้

วิธีการดำเนินงาน

การพัฒนาระบบสำรวจหนังสือของห้องสมุด สามารถแสดงเป็นขั้นตอนหลักๆ ดังนี้

1. เริ่มต้นจากการศึกษาและออกแบบความต้องการของระบบ โดยอ้างอิงถึง เครื่องมือหรือโปรแกรมที่ใช้ และระยะเวลาในการพัฒนาโครงงาน เป็นหลัก เพื่อทำการกำหนดฟังก์ชันการทำงานที่ระบบสามารถทำได้ ได้อย่างครบถ้วน

2. ทำการศึกษาทฤษฎีพื้นฐานต่างๆที่เกี่ยวข้องกับโครงงาน อันได้แก่

- ศึกษาขั้นตอนการสำรวจหนังสือของระบบห้องสมุด
- ศึกษาการออกแบบระบบ และ ฐานข้อมูล
- ศึกษาการทำงานของบาร์โค้ด และฮาร์ดแวร์ที่ใช้
- การใช้งานซอฟทแวร์ที่ใช้พัฒนาโปรแกรมประยุกต์ต่างๆ

3. นำความรู้ที่ได้จากการศึกษามาทำการออกแบบ โครงร่างของระบบ ทั้ง รูปแบบฐานข้อมูล แบบฟอร์มการทำงาน และ ส่วนติดต่อผู้ใช้ ซึ่งการออกแบบดังกล่าวจะต้องครอบคลุมการทำงานในฟังก์ชันต่างๆ

4. เมื่อทำการออกแบบในรายละเอียดต่างๆ แล้ว จะเริ่มทำการพัฒนาระบบตามที่ได้ออกแบบไว้ อันได้แก่ การสร้างฐานข้อมูล เพื่อจัดเก็บและบันทึกข้อมูลที่เกี่ยวข้อง การสร้างแบบฟอร์มการทำงานหรือส่วนติดต่อผู้ใช้ โดยในส่วนนี้จำเป็นจะต้องมีความรู้ในด้านโครงสร้างคำสั่งต่างๆ ของโปรแกรมที่ใช้เป็นเครื่องมือในการพัฒนาโปรแกรมเป็นอย่างดี ซึ่งจำเป็นจะต้องทำการศึกษาการใช้งานโปรแกรม ควบคู่ไปด้วย

5.ในระหว่างการพัฒนาระบบจำเป็นจะต้องมีการตรวจสอบความถูกต้องของระบบว่า ใช้งานได้หรือไม่ มีข้อผิดพลาดในส่วนใด โดยจะต้องมีการทำงานในส่วนนี้ควบคู่ไปกับการพัฒนาระบบ เพื่อให้ง่ายต่อการพัฒนาระบบ

6.หลังจากที่ได้ตรวจสอบระบบว่าสามารถทำงานได้ถูกต้องและมีฟังก์ชันการทำงานครบถ้วนตามที่ได้ออกแบบไว้ ถือว่า การพัฒนาระบบได้เสร็จสิ้น

7. ขั้นตอนสุดท้าย คือ การสรุปผลการพัฒนาโครงงาน และจัดทำรายงานประกอบการพัฒนา ซึ่งจะมีรายละเอียดต่างๆ ที่เกี่ยวข้อง

บทที่ 2

ทฤษฎีและหลักการ

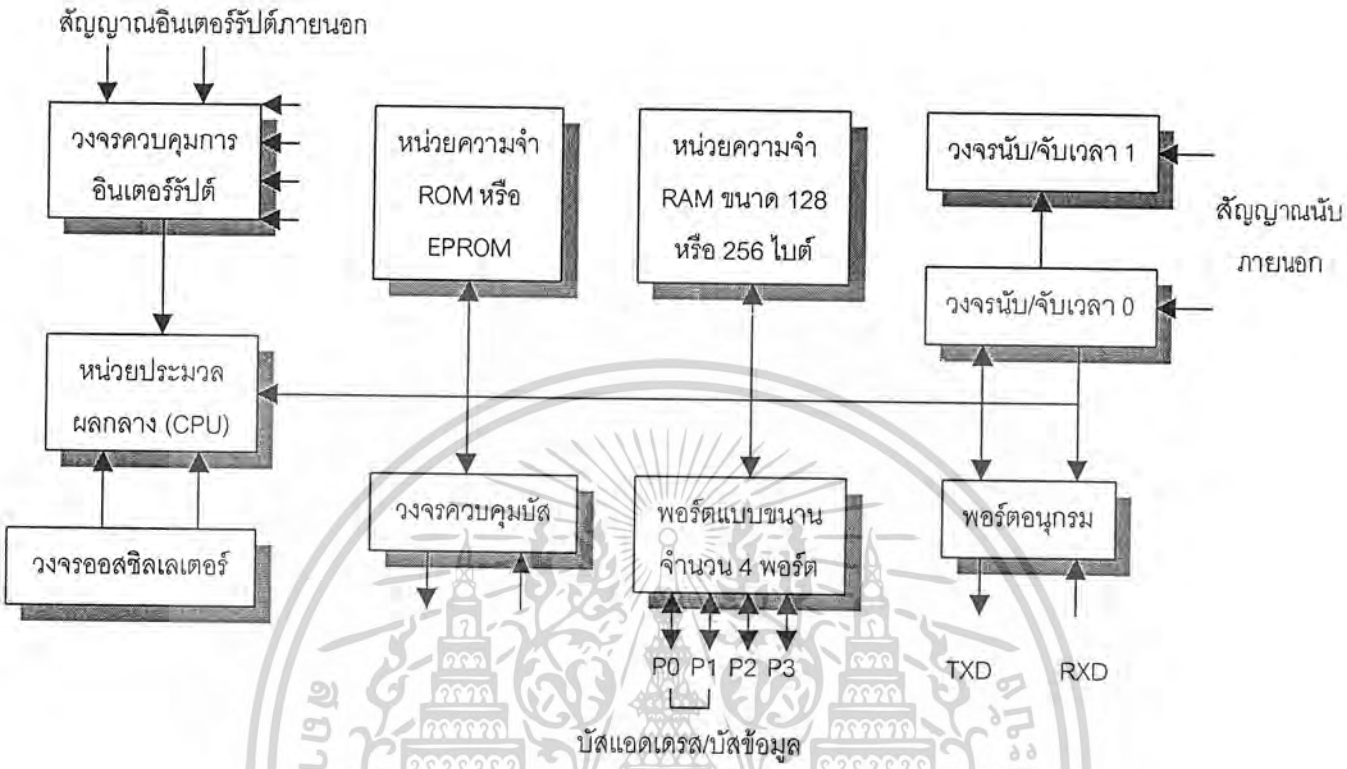
ไมโครคอนโทรลเลอร์ตระกูล 8051

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ประกอบด้วยไมโครคอนโทรลเลอร์หลายรุ่น ซึ่งมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหรือจำนวนของหน่วยทำงานภายในที่แตกต่างกันออกไป เพื่อความเหมาะสมในการประยุกต์ต่างๆ ตามความต้องการ โดยทีทั้งลักษณะที่ใช้เทคโนโลยีการผลิตไอซีวงจรรวมความจุสูงมาก (LSI) แบบ HMOS หรือ CHMOS ซึ่งมีลักษณะที่สูงมากขึ้น และสิ้นเปลืองกำลังไฟฟ้าน้อยกว่ามาก

ลักษณะพื้นฐานของ 8051

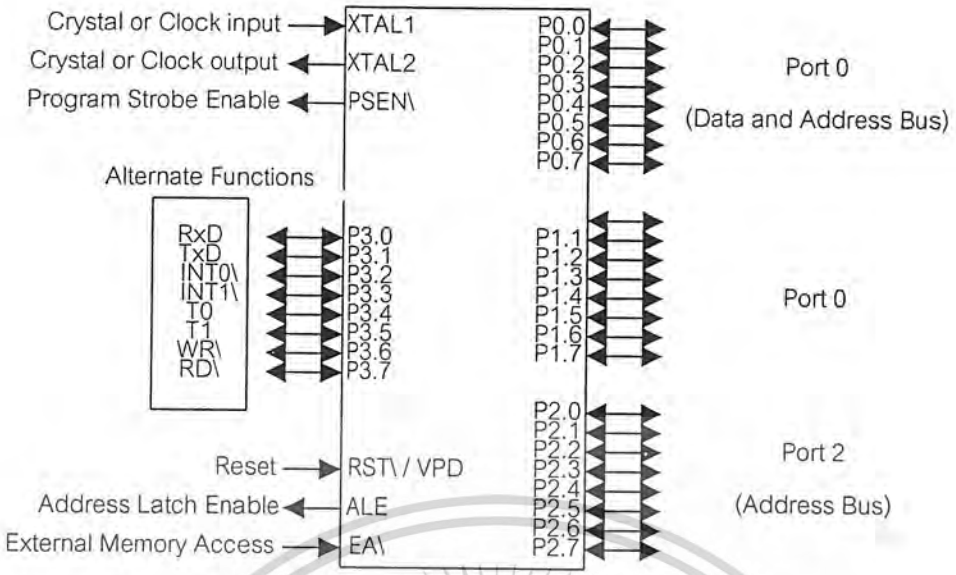
จากแผนภาพในรูปที่ 2.1 แสดงให้เป็นถึงหน่วยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่างๆ ที่จัดอยู่ภายในตระกูล MCS-51 นี้ ประกอบด้วย

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (Boolean Processor)
- ความสามารถในการอ้างถึงตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างถึงตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ EPROM (เบอร์ 8751) หรือแบบ ROM (เบอร์ 8051)
- หน่วยความจำแบบ RAM ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวน 2 วงจร
- วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ (Full Duplex)
- วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนดลำดับความสำคัญได้สองระดับ
- วงจรออสซิลเลเตอร์ภายใน



รูปที่ 2.1 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของ MCS-51

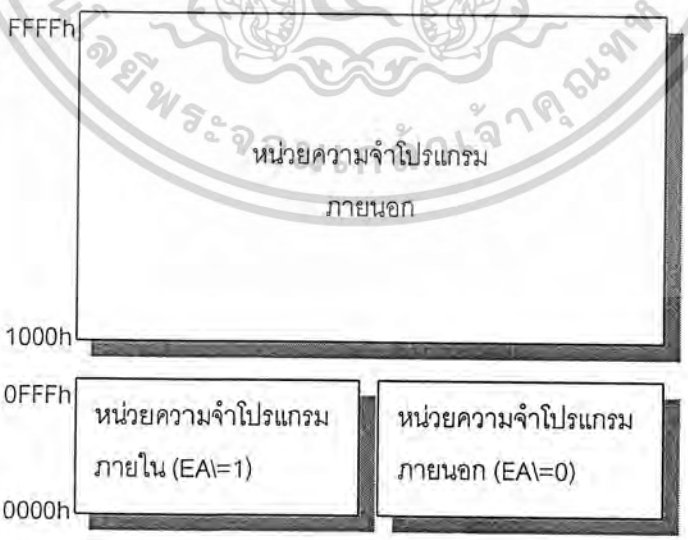
โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา ดังแผนภาพในรูปที่ 2.2 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณ ที่อาจมีหน้าที่ได้มากกว่าหนึ่งอย่าง ซึ่งไม่สามารถใช้งานในเวลาเดียวกันได้ การกำหนดว่าจะทำงานในลักษณะใดก็ขึ้นอยู่กับการเชื่อมต่อวงจรเข้ากับขาสัญญาณ และโปรแกรมควบคุมของระบบนั้น



รูปที่ 2.2 การกำหนดหน้าที่ขาสัญญาณของไอซี 8051

หน่วยความจำโปรแกรมของ 8051

หน่วยความจำโปรแกรมของ 8051 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูล และคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบ ข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซี หน่วยความจำประเภทต่างๆ เช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory) เป็นต้น



รูปที่ 2.3 การจัดพื้นที่หน่วยความจำโปรแกรมสำหรับไมโครคอนโทรลเลอร์ 8051

8051 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซีไมโครคอนโทรลเลอร์เอง และหน่วยความจำโปรแกรมภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

หน่วยความจำโปรแกรมภายใน

ไมโครคอนโทรลเลอร์เบอร์ต่างๆ ที่จัดอยู่ภายในตระกูล 8051 นี้ มีขนาดของหน่วยความจำโปรแกรมภายในแตกต่างกันออกไป เพื่อความเหมาะสมกับการนำไปใช้งานในลักษณะต่างๆ กัน ดังนี้

8051 และ 8052 มีหน่วยความจำแบบ ROM ขนาด 4 และ 8 กิโลไบต์ ตามลำดับ ประกอบอยู่ภายในไอซี และมีความเหมาะสมกับการนำไปใช้ในวงจรทางอุตสาหกรรมที่มีจำนวนการผลิตมาก เนื่องจากจะมีผลทำให้ต้นทุนค่าใช้จ่ายการผลิตต่อหน่วยลดลงได้มาก

8751 มีหน่วยความจำแบบ EPROM ขนาด 4 กิโลไบต์อยู่ภายในไอซี ข้อมูลที่จัดเก็บอยู่ภายในนี้สามารถใช้แสงอัลตราไวโอเลตลบ และนำกลับไปบรรจุโปรแกรมใหม่ได้อีกครั้งหนึ่ง

8031 และ 8032 ไม่มีหน่วยความจำโปรแกรมอยู่ภายในตัวไอซีเลย ดังนั้นในการนำไปใช้งานจึงจำเป็นต้องอาศัยหน่วยความจำภายนอกเสมอ ซึ่งมีผลทำให้ต้องเสียความสามารถบางประการเกี่ยวกับพอร์ตอินพุต/เอาท์พุตของไมโครคอนโทรลเลอร์ไป เนื่องจากต้องนำไปใช้เป็นสัญญาณควบคุม เกี่ยวกับการจัดการติดต่อหน่วยความจำภายนอกแทน

หน่วยความจำข้อมูลของ 8051

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูล หรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว เป็นหน่วยความจำ RAM แบบสแตติก พื้นที่หน่วยความจำข้อมูลของ 8051 สามารถมีได้สูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจรลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

หน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายในของ 8051 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะ คือ พื้นที่เฉพาะสำหรับตัวประมวลผลกลางใช้งานเท่านั้น ซึ่งเรามักนะเรียกกันอีกชื่อหนึ่งว่า รีจีสเตอร์ (Register) และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา

นั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำข้อมูลภายในปกติด้วยการอ้างถึงหมายเลขของแอดเดรสนั้นๆ โดยตรง

บริเวณแอดเดรส 20H-2FH จำนวน 16 กิโลไบต์ บริเวณพื้นที่เป็นส่วนสำหรับผู้ใช้ซึ่งจะมีความพิเศษต่างไปจากหน่วยความจำส่วนอื่นๆ เนื่องจากผู้ใช้อาจสามารถอ้างถึงหน่วยความจำบริเวณนี้ได้ทั้งในลักษณะของไบต์ข้อมูลเช่นปกติ หรืออาจจะเป็นบิตข้อมูลได้โดยตรง (ซึ่งถือเป็นการใช้งาน 8051 อย่างเต็มประสิทธิภาพ เนื่องจากว่า 8051 ได้รับการออกแบบมาสำหรับงานควบคุมเป็นพื้นฐานอยู่แล้ว ซึ่งส่วนมากจะเป็นการนำเข้าข้อมูลก็มักจะเป็นเพียงการอ่านค่าสถานะลอจิกของเส้นสัญญาณ หรือกรณีการส่งออกข้อมูลก็จะเป็นการกำหนดสภาวะลอจิกให้กับวงจรภายนอกผ่านทางบิตใดบิตหนึ่งอยู่แล้ว ก็จะมีพื้นที่ตัวแปรแบบบิตให้ใช้งานได้มากถึง 128 บิต โดยตำแหน่งแรกของบิตจะเป็นบิตซึ่งเริ่มต้นนับจากบิตนัยสำคัญต่ำสุด (LSB) ของแอดเดรส 20H เรื่อยไปจนกระทั่งถึงบิตที่ 127 ซึ่งเป็นบิตนัยสำคัญสูงสุด (MSB) ของแอดเดรส 2FH

บริเวณแอดเดรส 20H-7FH เป็นบริเวณที่สามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถอ้างถึงได้เฉพาะในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

(2) หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่บริเวณตั้งแต่ 80H-FFH เป็นบริเวณของหน่วยความจำข้อมูลภายในสำหรับการใช้งานเพิ่มมากขึ้นกว่าเบอร์อื่นๆ เช่น 8031 หรือ 8751 อีก 128 ไบต์ โดยจะอยู่บริเวณช่วงแอดเดรส 80H ถึง FFH เช่นกัน ซึ่งแม้ว่าจะเป็นพื้นที่ที่มีหมายเลขแอดเดรสเดียวกับส่วนของรีจิสเตอร์หน้าที่พิเศษ แต่ในความเป็นจริงแล้วจะเป็นพื้นที่หน่วยความจำอีกบริเวณหนึ่ง ซึ่งมีการซ้อนเกยกันให้อยู่ในบริเวณแอดเดรสส่วนนี้ ซึ่งหากว่าผู้ใช้งานต้องการจะเก็บข้อมูลในพื้นที่บริเวณนี้ก็จะต้องใช้การอ้างถึงหน่วยความจำแบบโดยอ้อม (Indirect Addressing) เท่านั้น

รีจิสเตอร์หน้าที่พิเศษ

รีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่ และการทำงานของอุปกรณ์หรือพอร์ตของ 8051 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

ชื่อรีจีสเตอร์	คำจำกัดความ	ความสามารถในการอ้างถึงแบบบิต
ACC	Accumulator	ได้
B	B Register	ได้
PSW	Program Stack Word	ได้
SP	Stack Pointer	ได้
DPTR	Data Pointer (DPH & DPL)	ได้
P0	Port 0	ได้
P1	Port 1	ได้
P2	Port 2	ได้
P3	Port 3	ได้
IP	Interrupt Priority	ได้
IE	Interrupt Enable	ได้
TMOD	Timer / counter mode	ไม่ได้
TCON	Timer / counter control	ได้
TH0	Timer / counter 0 (high)	ไม่ได้
TL0	Timer / counter 0 (low)	ไม่ได้
TH1	Timer / counter 1 (high)	ไม่ได้
TL1	Timer / counter 0 (low)	ไม่ได้
SCON	Serial Control	ไม่ได้
SBUF	Serial Data Buffer	ไม่ได้
PCON	Power Control	ไม่ได้

รูปที่ 2.5 รีจีสเตอร์ใช้งานพิเศษ (Special Function Register หรือ SFR)

จากรูปที่ 2.5 แสดงให้เห็นลักษณะการจัดพื้นที่หน่วยความจำสำหรับรีจีสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่ารีจีสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่เป็นจำนวนทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือ แอดเดรส 80H, 88H, 90H, 98H, A0H, A8H, B0H, B8H, D0H, E0H และ F0H)

แอกคิวมูเลเตอร์ (Accumulator) หรือ ACC เป็นรีจีสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียู และเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานของรีจีสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานภายในโปรแกรมจะเรียกว่า รีจีสเตอร์ A

รีจีสเตอร์ บี (B Register) เป็นรีจีสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณ และหารตัวเลข ในกรณีที่ไม่ใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจีสเตอร์ทั่วไปได้

โปรแกรมเคาน์เตอร์ (Program Counter) เป็นรีจีสเตอร์ที่ใช้ในการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานภายในโปรแกรมจะเรียกว่า รีจีสเตอร์ PC

สแต็กพอยน์เตอร์ (Stack Pointer) เป็นรีจีสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งขอตัวชี้ หรือพอยน์เตอร์ของบริเวณสแต็กสำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจีสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม โดยปกติแล้วเมื่อทำการเริ่มต้นระบบใหม่ภายหลังจากการเริ่มจ่ายไฟฟ้า หรือมีการรีเซตเกิดขึ้น ค่าภายในสแต็กพอยน์เตอร์จะมีค่า 07H ซึ่งเป็นตำแหน่งแอดเดรสภายในบริเวณเนื้อที่ 128 ไบต์แรกขอองหน่วยความจำข้อมูลภายใน การใช้งานภายในโปรแกรมจะเรียกว่า รีจีสเตอร์ SP

ตัวชี้ข้อมูล หรือดาต้าพอยน์เตอร์ (Data Pointer) เป็นรีจีสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจีสเตอร์ SPTR และสามารถใช้งานแยกออกเป็นรีจีสเตอร์ขนาด 8 บิตสองตัว คือ รีจีสเตอร์ DPH และ DPL เพื่อเก็บค่าของแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจจะเป็นแอดเดรสของอุปกรณ์ภายนอก ซึ่งกำหนดให้ติดต่อกันโดยใช้ตำแหน่งของหน่วยความจำนั้นภายในโปรแกรม

โปรแกรมสเตตัสเวิร์ด (PSW) รีจีสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์ของรีจีสเตอร์ที่ใช้งานด้วย ดังแสดงเป็นแผนภาพในรูปที่ 2.6

รีจีสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register) รีจีสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาต์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจีสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุต หรือการเอาต์พุตได้ การดำเนินการใดๆ ที่เกี่ยวข้องกับพอร์ตทั้งสี่นี้จะมีผลทำให้จะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรม หรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรส 8 บิตบนของหน่วยความจำ ส่วนพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือส่งข้อมูลกับหน่วยอุปกรณ์ภายนอก สำหรับพอร์ต P3 นั้น นอกเหนือจากจะใช้ในฐานะของพอร์ตอินพุต/เอาต์พุตเช่นปกติแล้ว ยังนำมาใช้ในฐานะบัสควบคุมเกี่ยวกับสัญญาณอินเตอร์รัปต์ได้อีกด้วย

รีจีสเตอร์ SBUF เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์นี้มีอยู่ด้วยกันสองชุด และแยกออกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดยซีพียูจะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

รีจีสเตอร์ PCON เป็นรีจีสเตอร์ที่ใช้ในการควบคุมหน้าที่การงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโพรเซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟล็กสภาวะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

- บิต PD (Power Down) เป็นการกำหนดให้ลดกำลังไฟฟ้าที่จ่ายให้กับส่วนของโพรเซสเซอร์ภายในลง โดยยังคงมีกำลังไฟฟ้าจ่ายให้กับส่วนหน่วยความจำข้อมูลภายในผ่านทางขาสัญญาณ RST วิธีการนี้มักนำมาใช้ในกรณีที่มีการตรวจสอบการไม่มีกำลังไฟฟ้า โดยวงจรตรวจสอบภายนอกจะต้องมีการอินเตอร์รัปต์เข้ามา เพื่อทำการเก็บข้อมูลที่กำลังประมวลผลอยู่ก่อน และเมื่อมีกระแสไฟฟ้าจ่ายให้เป็นปกติแล้ว จึงค่อยนำข้อมูลนั้นมาประมวลผลต่อไป
- บิต IDL (Idle Mode) เป็นการกำหนดให้โพรเซสเซอร์หยุดการทำงานชั่วขณะ (Sleep) และจะกลับมาอยู่ในสภาพปกติอีกครั้งเมื่อทำการรีเซตทางฮาร์ดแวร์ หรือมีการอินเตอร์รัปต์อย่างใด

อย่างหนึ่งเกิดขึ้น การทำงานในลักษณะนี้สามารถเกิดขึ้นได้ก็เนื่องจากว่า สภาวะการหยุดทำงานชั่วขณะนั้น เป็นเพียงการห้ามไม่ให้มีสัญญาณนาฬิกาจ่ายให้ส่วนของโพรเซสเซอร์เท่านั้น ส่วนของวงจรการอินเตอร์รัปต์พอร์ตอนุกรม และวงจรนับ/จับเวลายังคงมีสัญญาณนาฬิกาอยู่เป็นปกติ

รีจีสเตอร์ IP, IE, TMOD, SCON เป็นกลุ่มของรีจีสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของการอินเตอร์รัปต์ต่างๆ ของ 8051

ชื่อบิต: PSW ตำแหน่ง: D0h ค่าบิตเริ่มต้น: 0000 0111

CY	AC	F0	RS1	RS0	OV	.	P

ชื่อบิต	ตำแหน่ง	ความหมาย
CY	PSW.7	Carry Flag
AC	PSW.6	Auxiliary Carry Flag
F0	PSW.5	Flag 0
RS1	PSW.4	บิตสำหรับเลือกรีจีสเตอร์แบงก์ บิต 1
RS0	PSW.3	บิตสำหรับเลือกรีจีสเตอร์แบงก์ บิต 0
OV	PSW.2	Overflow Flag
.	PSW.1	
P	PSW.0	Parity Flag

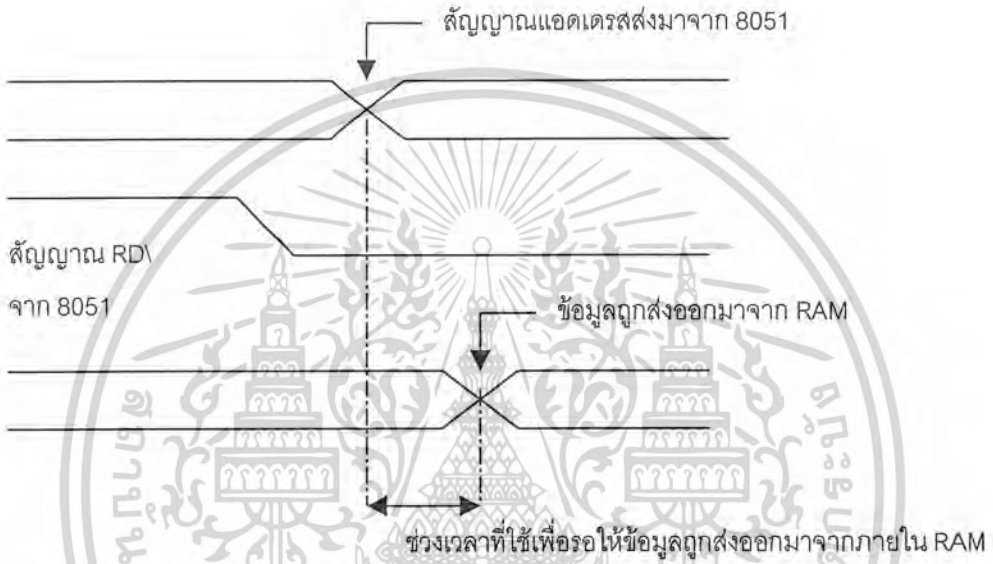
รูปที่ 2.6 บิตต่างๆ ภายในรีจีสเตอร์ PSW (Program Status Word)

หน่วยความจำข้อมูลภายนอกของ 8051

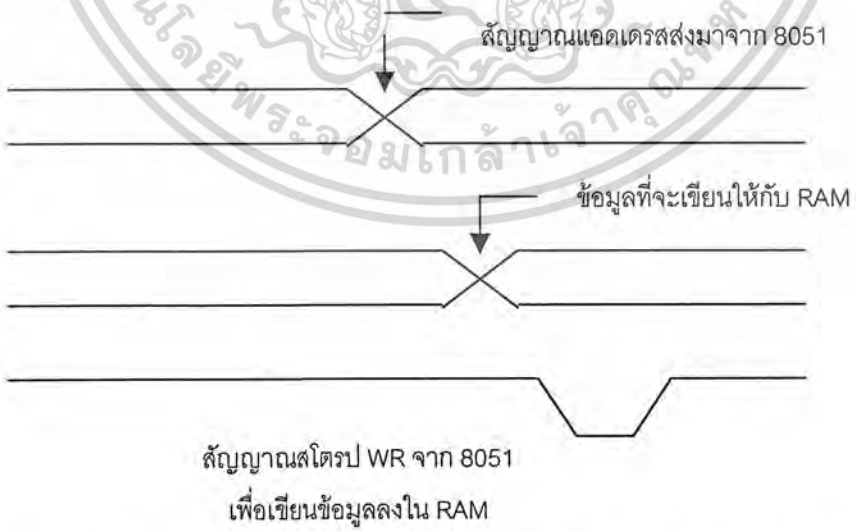
การใช้หน่วยความจำข้อมูลภายนอกเป็นวิธีการแก้ปัญหาอย่างหนึ่ง ในกรณีที่มีความต้องการหน่วยความจำสำหรับการเก็บข้อมูลชั่วคราว หรือตัวแปรของโปรแกรมมากเกินกว่าขนาดของหน่วยความจำข้อมูลภายใน ซึ่งมีขนาดเพียง 128 หรือ 256 ไบต์เท่านั้น บางครั้งการใช้หน่วยความจำข้อมูลยนอกยังเหมาะกับงานประยุกต์บางอย่างที่จำเป็น ต้องมีการเก็บสำรองข้อมูลบางอย่างไว้ไม่ให้สูญหาย แม้ว่าจะไม่มีการจ่ายไฟให้กับระบบก็สามารถทำได้โดยการใช้ไอซีหน่วยความจำ RAM พร้อมแบตเตอรี่สำรองประเภทลิเทียมหรือนิเกิลแคดเมียมเป็นตัวเก็บข้อมูลเหล่านี้ไว้แทน อย่างไรก็ตาม ไม่ว่าสาเหตุการนำไอซีหน่วยความจำภายนอกมาใช้งานจะเป็นอะไรจะมีผลทำให้พอร์ตอินพุต/เอาต์พุตข้อมูลของ 8051 ถูกนำไปใช้เพื่อให้ติดต่อกับหน่วยความจำเหล่านี้แทน ดังนั้นอาจจำเป็นต้องมีการใช้วงจรประกอบอื่นๆ เพื่อชดเชยความสามารถเหล่านั้นของ 8051 แทน

ลักษณะสมบัติของหน่วยความจำ RAM แบบสแตติก

กระบวนการนำข้อมูลเข้าไปจัดเก็บไว้ภายในหน่วยความจำ RAM แบบสแตติก จะเรียกว่า การเขียนไปยังหน่วยความจำ และเมื่อมีการนำข้อมูลที่จัดเก็บไว้นั้นออกมาก็จะเรียกว่า การอ่านจากหน่วยความจำ ซึ่งกระบวนการทั้งสองนี้จะต้องกระทำภายในช่วงรอบเวลาที่กำหนดไว้เช่นกัน สำหรับลักษณะสมบัติของหน่วยความจำ RAM มีความคล้ายคลึงกับหน่วยความจำ EPROM มาก รวมทั้งประเภทของสัญญาณติดต่อ คือ บัสแอดเดรส บัสข้อมูล และกลุ่มสัญญาณควบคุมทั้งหมดเพียงแต่มีการเพิ่มสัญญาณ WR เพื่อระบุถึงการนำข้อมูลจากบัสข้อมูลลงเก็บไปยังหน่วยความจำตำแหน่งที่ได้ระบุแอดเดรสมาเท่านั้น



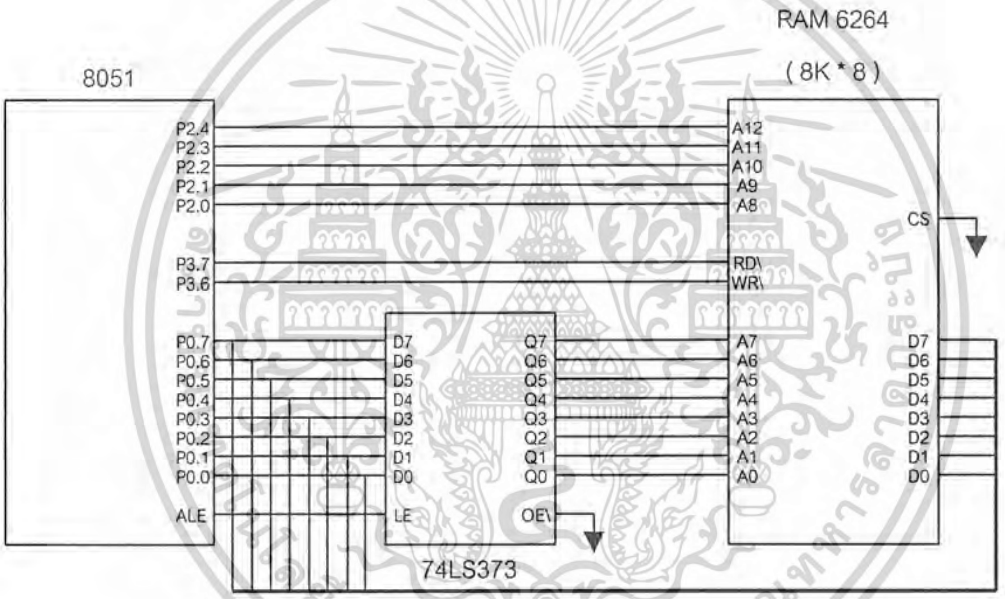
รูปที่ 2.7 แผนภาพเวลาแสดงลำดับเหตุการณ์การอ่านข้อมูลจากหน่วยความจำ RAM



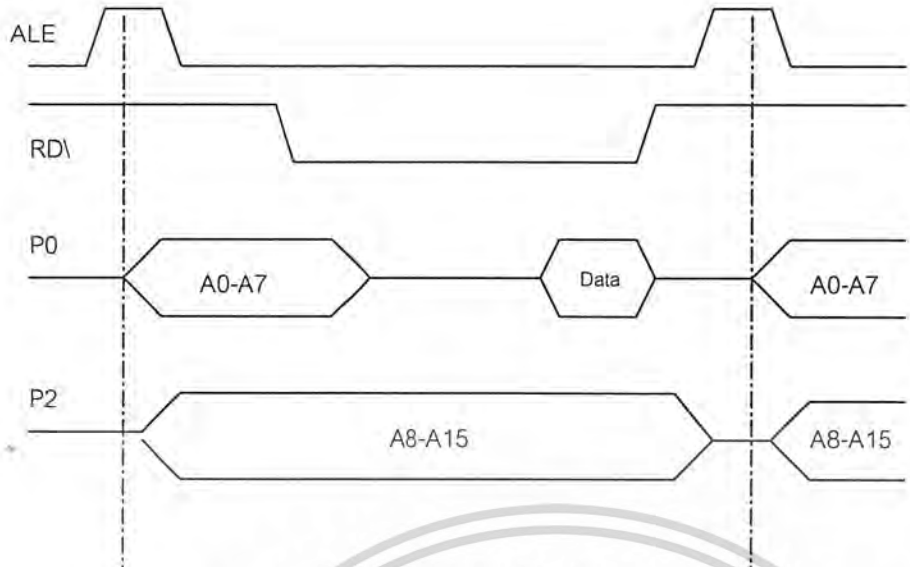
รูปที่ 2.8 แผนภาพเวลาแสดงลำดับเหตุการณ์การเขียนข้อมูลจากหน่วยความจำ RAM

การเชื่อมต่อกับหน่วยความจำข้อมูลภายนอก

การเชื่อมต่อกับหน่วยความจำ RAM เข้ากับระบบของไมโครคอนโทรลเลอร์ 8051 จะใช้วิธีการเหมือนกับหน่วยความจำ EPROM โดยในรูปที่ 2.9 แสดงให้เห็นถึงวงจรของการเชื่อมต่อเข้ากับหน่วยความจำ RAM แบบสแตติกเบอร์ 6264 ขนาด 8K*8 ไบต์ มีหลักการทำงานตามที่ได้กล่าวไว้ในหัวข้อที่ผ่านมา ไอซี 74LS373 ทำหน้าที่ในการค้างหรือแลตช์ค่าแอดเดรสให้กับอินพุตของหน่วยความจำ RAM ซึ่งมีอยู่เพียงตัวเดียว สำหรับขาสัญญาณ OE (หรือ RD) จะต่อเข้ากับขาสัญญาณ RD และขาสัญญาณ WR ของ 8051 ในที่นี้ เนื่องจากว่าภายในวงจรนี้มีไอซีหน่วยความจำเพียงตัวเดียว จึงได้ทำการต่อสัญญาณ CS กับสัญญาณกราวด์โดยตรง เพื่อทำการเลือกให้ไอซีทำงานตลอดเวลา



รูปที่ 2.9 วงจรตัวอย่างแสดงการเชื่อมต่อหน่วยความจำข้อมูลภายนอกกับ 8051



(a) วัฏจักรของการอ่านข้อมูลจากหน่วยความจำ RAM

(b) วัฏจักรการเขียนข้อมูลในหน่วยความจำ RAM

รูปที่ 2.10 แผนภาพเวลาแสดงการติดต่อกับหน่วยความจำ RAM ของ 8051

ในรูปที่ 2.10 แสดงให้เห็นถึงแผนภาพเวลาของสัญญาณที่เกี่ยวข้องดังนี้ เมื่อเริ่มการติดต่อกับหน่วยความจำข้อมูลภายนอกจะมีการส่งค่าแอดเดรสไบต์ต่ำ (A0-A7) ของตำแหน่งหน่วยความจำที่ต้องการออกมายังพอร์ต 0 ในราวก่อนช่วงเวลาของขาลงของสัญญาณ ALE ซึ่งจะนำไปใช้ในการควบคุมไอซีแลตซ์เพื่อทำการค้างค่าแอดเดรสไบต์ต่ำนี้ไว้ ดังนั้นในเวลาต่อมาพอร์ต 0 ก็จะว่าง และพร้อมที่จะนำไปใช้ในฐานะบัสข้อมูล ส่วนค่าของแอดเดรสไบต์สูง (A8-A15) ก็จะถูกส่งออกมาทางพอร์ต 2 ราวช่วง

เวลาประมาณกึ่งกลางระหว่างที่สัญญาณ ALE เป็นระดับลอจิกสูง เมื่อ 8051 ทำการขับสัญญาณ RD หรือ WR ให้เป็นระดับลอจิกต่ำก็จะมีผลทำให้เกิดการส่งและรับข้อมูลระหว่างหน่วยความจำ RAM กับบัสข้อมูลขึ้นได้

พอร์ตอินพุตและเอาต์พุต

พอร์ต หมายถึง แอดเดรสหนึ่งที่ได้รับการกำหนดไว้เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูลเมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก ดังนั้นการนำเข้าข้อมูลจากวงจรภายนอกจึงเรียกว่า การอินพุต และในกรณีตรงกันข้ามเพื่อส่งออกข้อมูลก็จะเรียกว่า การเอาต์พุต

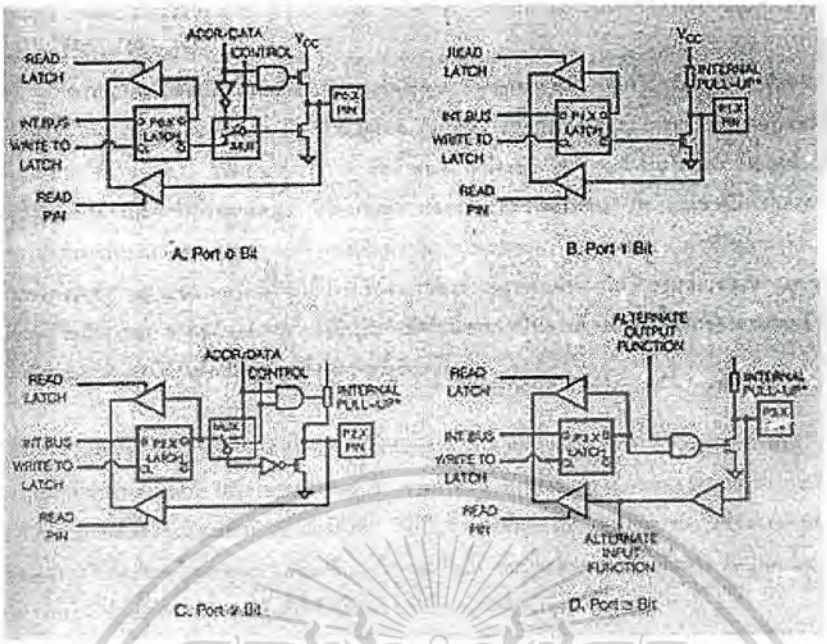
เมื่อพิจารณาถึงการส่งข้อมูลภายในพอร์ตจะสามารถแยกประเภทของพอร์ตออกได้เป็นสองลักษณะ คือ พอร์ตแบบขนาน ซึ่งทำการส่งจำนวนบิตข้อมูลทั้งหมดออกมา หรือนำเข้าไปพร้อมกันในคราวเดียว และพอร์ตแบบอนุกรมซึ่งทำการโอนย้ายข้อมูลคราวละบิตๆ จนครบจำนวน

พอร์ตแบบขนานของ 8051

8051 มีโครงสร้างของพอร์ตที่สามารถใช้งานแบบขนานได้จำนวนทั้งหมดสี่พอร์ต เรียกชื่อเรียงตามลำดับว่า พอร์ต 0, 1, 2 และ 3 และเป็นพอร์ตขนาน 8 บิตทั้งหมด การใช้งานพอร์ตสามารถทำได้ทั้งในลักษณะของเส้นสัญญาณเดี่ยวๆ หรือกลุ่มของสัญญาณได้ นอกจากนี้พอร์ต 0, 2 และ 3 ยังสามารถนำไปใช้งานอื่นๆ ที่ไม่ใช่เป็นพอร์ตอินพุต/เอาต์พุตได้ โดยพอร์ต 0 จะทำหน้าที่มัลติเพล็กซ์ระหว่างบัสแอดเดรสไบต์ต่ำ และบัสข้อมูล สำหรับการติดต่อกับวงจรประกอบร่วมกับข้อมูลบัสแอดเดรสไบต์สูงซึ่งจะส่งออกมาทางพอร์ต 2 สำหรับพอร์ต 3 นั้น นอกเหนือไปจากความสามารถเช่นพอร์ตปกติแล้ว สามารถนำไปเป็นขาสัญญาณขอองการอินเตอร์รัปต์ต่างๆ ซึ่งรวมทั้งการสร้างสัญญาณการสร้างสัญญาณควบคุม RD\ และ WR\ เพื่อทำหน้าที่อ่าน หรือเขียนข้อมูลหน่วยความจำข้อมูลภายนอกด้วย การใช้งานพอร์ตลักษณะงานแบบอื่นๆ ที่ไม่ใช่พอร์ตแบบอินพุต/เอาต์พุตนี้จะดำเนินการโดย 8051 เองโดยอัตโนมัติ

โครงสร้างการทำงานของพอร์ต 8051

จากลักษณะโครงสร้างของแต่ละบิตภายในพอร์ตของ 8051 ซึ่งได้แสดงไว้ในรูปที่ 2.11 นั้น จะเห็นว่ามีความคล้ายคลึงกันตามลักษณะโครงสร้าง ที่เรียกว่า Quasi-bidirectional port ยกเว้นพอร์ต 0 ซึ่งเพียงแต่ไม่มีตัวต้านทานทำหน้าที่ Pull-up สัญญาณไว้ภายในเท่านั้น วงจรประกอบอื่นภายในยังมีฟลิปฟลอปเฉพาะของพอร์ต 0 และพอร์ต 2 จะมีโครงสร้างที่ทำหน้าที่คล้ายสวิตช์เพิ่มเติมขึ้น เพื่อควบคุมให้เอาต์พุตนี้ต่อเข้ากับส่วนของทรานซิสเตอร์ในระหว่างที่ไม่ได้มีการทำงานในลักษณะของบัสแอดเดรส หรือบัสข้อมูลด้วย สำหรับบัฟเฟอร์จำนวนสองตัวของทุกบิตในพอร์ตนั้นมีการทำงานแยกกันโดยอิสระ โดยตัวที่อยู่ทางด้านบนจะยอมให้สัญญาณผ่านได้ก็ต่อเมื่อมีการอ่านค่าข้อมูลที่ค้างไว้ ส่วนอีกตัวหนึ่งซึ่งอยู่ทางด้านล่างจะถูกใช้งานเฉพาะ เมื่อได้มีการอ่านสถานะของขาสัญญาณเท่านั้น



รูปที่ 2.11 โครงสร้างของแต่ละบิตภายในพอร์ตอินพุต/เอาท์พุตของ 8051

1. การใช้งานพอร์ตเป็นการอินพุต

การใช้งานพอร์ตเป็นการอินพุตข้อมูลจะต้องเริ่มด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตนั้นก่อนเป็นลำดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทาน ซึ่งทำหน้าที่ Pull-up ภายใน ซึ่งมีผลทำให้บิตนั้นๆ ของพอร์ต 1, 2 และ3 เป็นสภาวะของลอจิกสูง ตัวต้านทานนี้มีค่าประมาณ 50 กิโลโอห์ม ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับของพอร์ต 0 นั้น แม้ว่าจะมีหลักการทำงานที่คล้ายคลึงกันกับบิตของพอร์ตอื่นๆ แต่เนื่องจากการที่ไม่มีตัวต้านทานทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ทำหน้าที่ขับสัญญาณเอาต์พุตนั้นหยุดการทำงานก็จะเป็นผลให้ขาสัญญาณนี้อยู่ในสภาวะอิมพีแดนซ์สูงแทน

2. การใช้งานพอร์ตเป็นการเอาต์พุต

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับฟลิปฟลอปซึ่งจะค้างค่านี้ไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาต์พุตนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสภาวะลอจิกเป็นลอจิกต่ำด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1 หรือ 2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเออาต์พุตนั้นหยุดการทำงาน มีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการทำงานในแต่ละบิตทางพอร์ต 0 นั้นจะมีผลที่แตกต่างออกไป โดยขาสัญญาณจะเป็นสภาวะอิมพีแดนซ์สูง เนื่องจากไม่มีตัวต้าน

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

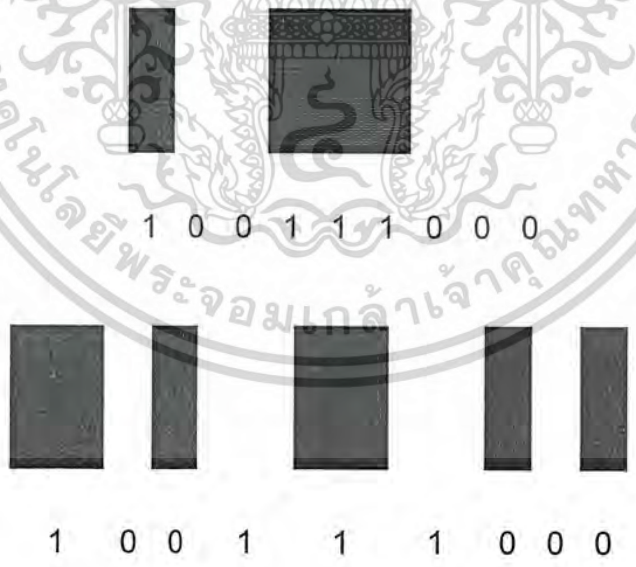
ทานภายในเชื่อมต่ออยู่นั่นเอง ดังนั้นในการใช้งานพอร์ต 0 เป็นการเอาต์พุตข้อมูล จึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทน

ความสามารถอีกประการหนึ่งเกี่ยวกับพอร์ตอินพุต/เอาต์พุตของ 8051 เป็นวิธีการอ่านค่าลอจิกจากพอร์ตซึ่งมีได้สองวิธี คือ การอ่านค่าลอจิกที่ขาสัญญาณ (Port pin) และการอ่านค่าลอจิกของการแลตช์ที่พอร์ตดังจะสังเกตได้จากรูปที่ 2.11 วิธีการอ่านค่าจากพอร์ตทั้งสองแบบนี้จะช่วยให้ระบบทำงานได้ด้วยความถูกต้องมากยิ่งขึ้น ยกตัวอย่างเช่น หากว่าพอร์ตถูกนำไปต่อกับขาเบสของทรานซิสเตอร์และ NPN และขาอิมิเตอร์ต่อกับกราวน์ของระบบ เมื่อมีการส่งค่า 1 ออกไปจะมีผลทำให้ทรานซิสเตอร์ทำงาน ในขณะนั้นถ้าซีพียูมีการอ่านค่าลอจิกจากขาสัญญาณของพอร์ตนี้ก็จะได้ค่าลอจิกต่ำเนื่องจากมองเห็นค่าศักย์ไฟฟ้าระหว่างขาเบส และขาอิมิเตอร์ ซึ่งมีค่าประมาณ 0.6 โวลต์แทน ดังนั้นในกรณีเช่นนี้หากว่าเป็นการอ่านค่าจากลอจิกของการแลตช์ ก็จะได้รับค่าระดับลอจิกสูง ซึ่งเป็นค่าที่ถูกต้องสภาพที่เป็นจริง

รหัสแถบ

หลักการของรหัสแถบ

รหัสแถบ คือ ข้อมูลประจำตัวที่กำหนดขึ้นแล้วมาจัดเรียงเข้ารหัสในรูปของแถบสีดำและแถบสีขาว วางเรียงขนานสลับกันด้วยชุดความกว้างของแถบ หรือจำนวนของแถบต่างๆ กันไปขึ้นอยู่กับข้อมูลที่ต่างกัน และชนิดของรหัสแถบที่เลือกใช้ต่างๆ กันด้วย



Encoding of binary string 100111000

Delta Code (top) and Width Code (bottom)

รูปที่ 2.12 แสดงรูปแบบการเข้ารหัส

37061

การเข้ารหัสของรหัสแถบ (Bar Code) แบ่งออกเป็น 2 วิธีการคือ แบบแรกจะใช้สีของแถบ นำมาเข้ารหัส โดยใช้แถบสีดำแทน "1" และแถบสีขาวแทน "0" ลักษณะเช่นนี้เรียกว่า"เดลต้าโคด" (Delta Code) อีกรูปแบบหนึ่งจะใช้ความกว้างของแถบนำมาเข้ารหัส ที่เรียกว่า "วิดท์โคด" (Width Code) โดยถ้าเป็นแถบกว้างจะแทน "1" และแถบแคบจะแทน "0" การเข้ารหัสเช่นนี้จะไม่สนใจสีของแถบเลย

ชนิดของรหัสแถบ

ปัจจุบันรหัสแถบที่ใช้มีอยู่ด้วยกันหลายชนิด รหัสที่นิยมใช้แพร่หลายแบ่งได้เป็น

ชนิดรหัส 2 ใน 5 (2 of 5 code)

ชนิดรหัส 2 ใน 5 แบบแทรกสอด (Interleaved 2 of 5 code)

ชนิดรหัส 3 ใน 9 (3 of 9 or 39 code)

ชนิดรหัสโคดาบาร์ (Codabar)

ชนิดรหัสยูพีซี (UPC = Universal Product Code)

ชนิดรหัสเอียน (EAN = European Article Numbering)

ชนิดรหัส 2 ใน 5 (2 of 5 code)

เป็นรหัสที่มีใช้ตั้งแต่ ค.ศ. 1960 เป็นแบบที่ใช้งานง่ายสุดในการใช้งาน การที่ชื่อเรียกว่า 2 ใน 5 เพราะใน 1 รหัสจะประกอบไปด้วยแถบ 5 แถบ (5 บิต) แต่จะมีแถบกว้างที่มีค่าเป็น 1 (แถบกว้าง) เพียง 2 แถบ (2 บิต) เท่านั้น ส่วนบิตที่เหลือเป็น 0 ทั้งหมด คือการแทนด้วยแถบแคบ (Narrow Bar) 3 แถบ โดยไม่นำส่วนที่เป็นช่องว่างมาใช้เลย

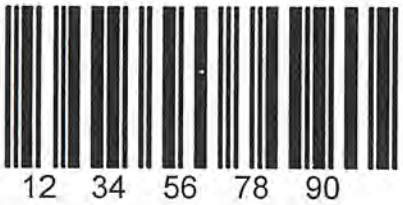
รหัส 2 ใน 5 นี้เป็นรหัสที่ใช้แทนข้อมูลได้เฉพาะตัวเลข 0 ถึง 9 เพียงแค่ 10 รหัสเท่านั้น เริ่มด้วยรหัสเริ่มต้น (Start Code) 3 บิต คือ 110 (แถบกว้าง 2 และแถบแคบ 1) และปิดท้ายด้วยรหัสปิดท้าย (Stop Code) 3 บิต คือ 101 ส่วนรหัสทั้ง 5 บิต ที่แทนเลข 0 ถึง 9 ดูได้จากในตารางในรหัส 2 ใน 5 แบบแทรกสอด

รูปที่ 2.13 แสดงรหัสแถบชนิด 2 ใน 5

ชนิดรหัส 2 ใน 5 แบบแทรกสอด (Interleaved 2 of 5 code)

รหัสแบบนี้คล้ายคลึงกับแบบแรกมาก เพราะพัฒนามาจากรหัสแบบแรกเนื่องจากรหัส 2 ใน 5 ไม่ได้นำส่วนที่เป็นช่องว่างกว้างและช่องว่างแคบมาใช้ คงใช้เพียงแต่แถบกว้างและแถบแคบ จึงทำให้ความหนาแน่นของข้อมูลน้อยลงนั่นคือ เมื่อต้องบรรจุข้อมูลต่อเนื่องหลายตัวเลข จะต้องใช้แถบที่มีความกว้างมากขึ้น

รหัส 2 ใน 5 แบบแทรกสอด ได้ดัดแปลงนำส่วนที่เป็นช่องว่างทั้ง 2 ชนิดมาใช้งานด้วย โดยการสอดแทรกรหัสลงไปอีก 1 รหัสทุกๆ ช่วง 5 แถบของรหัสปกติที่เป็นแถบสีดำ แต่ก็ยังแทนรหัสตัวเลข 0 ถึง 9 ได้เพียง 10 รหัสเท่านั้น



รูปที่ 2.14 แสดงรหัสแถบชนิด 2 ใน 5 แบบแทรกสอด

จากรูปที่ 2.14 ใช้แทนรหัสตัวเลข 1234...ตามลำดับ การใช้งานของรหัส 2 ใน 5 แบบแทรกสอด จะเริ่มต้นด้วยส่วนที่เป็นรหัสเริ่มต้นทางด้านซ้ายประกอบด้วยแถบแคบ 2 แถบ และช่องว่างแคบ 2 แถว สลับกัน ส่วนทางด้านขวาเป็นรหัสปิดท้ายประกอบด้วยแถบกว้าง 1 แถบ ช่องว่างแคบ 1 แถบและแถบแคบ 1 แถบ ตามลำดับ

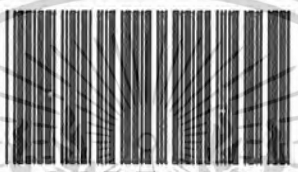
ภายในระหว่างรหัสเริ่มต้นและรหัสปิดท้ายแบ่งเป็น 2 ส่วน ส่วนแรกคือ ส่วนที่เป็นแถบดำกว้างและแคบ จะใช้แทนรหัสเหมือน 2 ใน 5 ขณะเดียวกันในส่วนของแถบรหัสเหล่านี้จะมีแถบช่องว่างสีขาวกว้างและแคบแทนได้ เช่นเดียวกันรหัส 2 ใน 5 ปกติ จากตัวอย่าง 5 แถบแรกที่เป็นสีดำแทนค่าได้เท่ากับ 1 แถบขาวในช่วงเดียวกันเท่ากัน 2 สีดำช่วงต่อมา 5 แถบแทนได้เท่ากับ 3 และ 5 แถบขาวต่อมาเท่ากับ 4 เช่นนี้ตลอดไปจนหมด รวมเป็นค่าที่อ่านได้เท่ากับ 1234...

ตารางที่ 2.1 ตารางรหัสเลขฐานสองของรหัส 2 ใน 5 ทั้งสองแบบ

ตัวเลข	เลขฐานสองทั้ง 2 แบบ				
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0

ชนิดรหัส 3 ใน 9 (3 of 9 or 39 code)

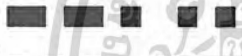
รหัสแถบที่ได้กล่าวมาข้างต้นยังมีข้อเสียที่สำคัญอยู่ประการหนึ่ง คือ สามารถแทนรหัสได้เฉพาะตัวเลขเท่านั้น จึงได้มีการพัฒนารหัสแถบชนิดใหม่ที่สามารถแทนได้ทั้งตัวเลขและตัวอักษรซึ่งรหัสแถบชนิดแรกที่สามารถทำได้คือ รหัส 39 ซึ่งพัฒนาขึ้นมาในปี ค.ศ. 1974 โดย Dr. David C. Allas และ Mr. Ray Stevens ชาวสหรัฐอเมริกา โดยรหัส 39 นี้เป็นรหัสที่ได้รับความนิยมอย่างมากโดยเฉพาะอย่างยิ่งในวงการอุตสาหกรรม ข้อดีของรหัสชนิดนี้คือใช้งานได้กว้างขวางมากขึ้น เพราะสามารถใช้ตัวเลขปนกับตัวอักษร และเครื่องหมายต่างๆ ได้ ซึ่งรหัสทั้งหมดแทนได้ตามตารางที่ 2.2



รูปที่ 2.15 แสดงรหัสแถบชนิด 39

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ตารางที่ 2.2 ตารางแสดงเลขฐานสองของรหัสแถบ 3 ใน 9

อักระ	แพทเทอร์น	แถบ	ช่องว่าง	อักระ	แพทเทอร์น	แถบ	ช่องว่าง
1		10001	0100	M		11000	0001
2		01001	0100	N		00101	0001
3		11000	0100	O		10100	0001
4		00101	0100	P		01100	0001
5		10100	0100	Q		00011	0001
6		01100	0100	R		10010	0001
7		00011	0100	S		01010	0001
8		10010	0100	T		00110	0001
9		01010	0100	U		10001	1000
0		00110	0100	V		01001	1000
A		10001	0010	W		11000	1000
B		01001	0010	X		00101	1000
C		11000	0010	Y		10100	1000
D		00101	0010	Z		01100	1000
E		10100	0010	_		00011	1000
F		01100	0010	.		10010	1000
G		00011	0010	SPACE		01010	1000
H		10010	0010	*		00110	1000
I		01010	0010	$		00000	1110
J		00110	0010	/		00000	1101
K		10001	0001	+		00000	1011
L		01001	0001	%		00000	0111

รหัส 39 มีส่วนประกอบที่สำคัญ 3 ส่วน คือ รหัสเริ่มต้นและรหัสปิดท้าย, รหัสข้อมูล และรหัสตรวจสอบ โดยมีข้อกำหนดต่างๆ กันในการแทนรหัสดังนี้

1. สามารถแทนรหัสได้ทั้งหมด 44 ตัว ตัวเลข 0 ถึง 9 พยัญชนะ A ถึง B และอักขระพิเศษอีก 8 ตัว คือ * , - , . , $, / , + , % และช่องว่าง (Space) โดย * นั้นจะใช้เป็นรหัสเริ่มต้นและสิ้นสุดเท่านั้น

2. ขนาดความกว้างของรหัสจะมีเพียง 2 ขนาด คือ แถบกว้าง (Width Bar) และแถบแคบ (Narrow Bar) และการแทนแถบจะใช้เลขฐานสอง 1 บิต โดยให้เลขฐานสอง “1” แทนแถบกว้างและเลขฐานสอง “0” แทนแถบแคบ

3. ในการแทนรหัสหนึ่งตัวจะใช้แถบดำ 2 แถบ สลับกับแถบขาว 4 แถบ รวมเป็น 9 แถบซึ่งประกอบด้วยแถบกว้าง 3 แถบและแถบแคบ 6 แถบ โดยไม่สนใจว่าจะเป็นแถบดำหรือแถบขาว

4. การจัดเรียงรหัสแถบ 39 จะเริ่มต้นด้วยรหัสเริ่มต้น แล้วตามด้วยรหัสข้อมูลและสิ้นสุดด้วยรหัสปิดท้าย โดยรหัสข้อมูลแต่ละตัวจะถูกแยกด้วยแถบขาวแคบ 1 แถบ และรหัส 39 นี้ก็ไม่ได้มีการกำหนดจำนวนข้อมูลไว้เป็นมาตรฐาน จึงสามารถมีข้อมูลได้มากน้อยตามต้องการ

จากข้อกำหนดที่กล่าวมานั้นยังมีข้อกำหนดพิเศษที่รหัส 39 สามารถจะเลือกได้ว่าจะใช้หรือไม่ใช้ก็ได้ คือการกำหนดค่ารหัสตรวจสอบ (Check-Sum) ซึ่งจะวางไว้ที่ตำแหน่งก่อนที่จะถึงรหัสสิ้นสุดโดยวิธีการหาค่ารหัสตรวจสอบทำได้ดังนี้

1. นำค่าตัวเลขประจำตัวของรหัสแต่ละตัวมาบวกกัน
2. นำผลบวกที่ได้ไปหารด้วย 43
3. นำค่าตัวเลขเศษของผลหารที่ได้ไปเทียบหารหัสตรวจสอบจากตารางที่ 2.2

ตัวอย่างการเข้ารหัส 39

กำหนดให้รหัสของข้อความเป็น 98PQ

- ขั้นแรกหาผลรวมของค่าตรวจสอบของอักขระทุกๆ ตัวในข้อความนั้นคือ 98PQ (จะได้ 9+8+25+26 = 68)
- หารผลรวมที่ได้ด้วย 43 (68/43) ก็จะได้เท่ากับ 1 กับเศษที่เหลืออีก 25
- ต่อมาให้ไปดูที่ตาราง 2.2 ว่าอักขระใดมีค่า Check-Sum เท่ากับ 25 ซึ่งอักขระที่ได้ก็คือตัว P
- ดังนั้นข้อความก็จะถูกแปลงเป็นรหัส 39 รวมทั้ง Check-Sum และอักขระเริ่มต้นและสิ้นสุดด้วย ดังนี้ *98PQ*
- จากนั้นนำข้อความที่ได้มาแปลงเป็นเลขฐานสองจะได้ดังนี้

98PQ => 010010100/0/001100100/0/100100100/0/001010010/0/000000111/0/010010100

ตารางที่ 2.3 ตารางแสดงการแทนรหัสและค่าเลขประจำตัวของรหัส 39

รหัส	การแทนรหัส	ค่าประจำตัว	รหัส	การแทนรหัส	ค่าประจำตัว
0	000110100	0	M	101000010	22
1	100100001	1	N	000010011	23
2	001100001	2	O	100010010	24
3	101100000	3	P	001010010	25
4	000110001	4	Q	000000111	26
5	100110000	5	R	100000110	27
6	001110000	6	S	001000110	28
7	000100101	7	T	000010110	29
8	100100100	8	U	110000001	30
9	001100100	9	V	011000001	31
A	100001001	10	W	111000000	32
B	001001001	11	X	010010001	33
C	101001000	12	Y	110010000	34
D	000011001	13	Z	011010000	35
E	100011000	14	-	010000101	36
F	001011000	15	.	110000100	37
G	000001101	16	(Space)	011000100	38
H	100001100	17	*	010010100	-
I	001001100	18	$	010101000	39
J	000011100	19	/	010100010	40
K	100000011	20	+	010001010	41
L	001000011	21	%	000101010	42

- แล้วนำเลขฐานสองที่ได้มาแทนด้วยแถบเส้นหรือช่องว่างโดยให้เลขฐานสอง "0" แทนด้วยแถบเส้นหรือช่องว่างที่แคบ (narrow) และเลขฐานสอง "1" แทนด้วยส่วนที่กว้าง (wide) ก็จะได้รหัสแถบตามความต้องการ

หมายเหตุ

อักขระแต่ละตัวในข้อความหนึ่งๆ จะถูกแยกออกจากกันด้วยช่องว่างแคบๆ (narrow space) ซึ่งมีค่าลอจิกเป็น "0"

ชนิดรหัสโคดาบาร์ (Codabar)

รหัสโคดาบาร์ประกอบด้วย 7 บิต โดย 4 บิตเป็นแถบดำและ 3 บิตเป็นช่องว่าง ใช้แทนตัวเลข 0 ถึง 9 เครื่องหมาย - , $, : , / , . , + , A , B , C , D

รหัสโคดาบาร์ที่สมบูรณ์จะต้องมีรหัสที่ใช้แทนตัวอักษร A B C หรือ D (เช่น A = 0011010) เป็นส่วนเริ่มต้นหรือสิ้นสุด ภายในประกอบด้วยรหัสของโคดาบาร์ที่เป็นตัวเลขและเครื่องหมายซึ่งทำให้มีความยาวไม่แน่นอนเพราะ 12 รหัสแรกมีบิตที่เป็น 1 อยู่ 2 บิต 4 รหัส ต่อมีบิต 1 อยู่ 3 บิต (โคดาบาร์ใช้ทั้งแถบดำและขาวแทนข้อมูลใน 1 รหัส) และ 4 รหัสสุดท้ายเป็นรหัสของ A , B , C , D กำหนดขึ้นมาเพื่อใช้เป็นรหัสเริ่มต้นและปิดท้าย

การที่มี 20 ตัวอักขระที่ใช้งานจากที่เป็นไปได้ 128 ตัวนั้น ทำให้รหัสชนิดนี้มีการตรวจสอบตัวเองเป็นลักษณะประจำตัว และไม่มีการกำหนดค่าของเช็คซัม (Check-Sum)

ข้อความหนึ่งๆ สามารถจะเข้ารหัสแบบโคดาบาร์ โดยจะประกอบด้วยอักขระเริ่มต้นและสิ้นสุด 4 ตัว (A , B , C หรือ D) ตัวใดตัวหนึ่ง ตัวอย่างเช่น ข้อความ 2345 ใช้ A เป็นอักขระเริ่มต้นและปิดท้าย ดังนั้นจะได้รหัสโคดาบาร์เป็น A2345A ซึ่งแปลงเป็นเลขฐานสองได้ดังนี้

0011010/0/0001001/0/1100000/0/0010010/0/1000010/0/0011010

โดยเริ่มต้นมีช่องว่างแคบๆ เป็นตัวแบ่งแยกอักขระแต่ละตัว ซึ่งก็คือเลขฐานสอง "0" ABC Codabar หรือรู้จักกันในชื่อ NW7 และค่อนข้างใช้งานกันอย่างกว้างขวาง ABC เป็นชื่อย่อมาจาก American Blood Commission และรหัสชนิดนี้เป็นมาตรฐานที่ยอมรับของนานาชาติ ซึ่งใช้เป็นบาร์โค้ดในงานที่เกี่ยวกับการถ่ายโลหิต และส่วนใหญ่แล้วจะใช้กันในด้านเวชกรรม

รูปที่ 2.16 แสดงรหัสแถบชนิดโคดาบาร์

ตารางที่ 2.4 ตารางแสดงเลขฐานสองของรหัสโคดาบาร์

หมายเลข	แพทเทอร์น	รหัส 7 บิต	อักขระ	แพทเทอร์น	รหัส 7 บิต
0		0000011	-		0001100
1		0000110	$		0011000
2		0001001	:		1000101
3		1100000	/		1010001
4		0010010	.		1010100
5		1000010	+		0010101
6		0100001	A		0011010
7		0100100	B		0101001
8		0110000	C		0001011
9		1001000	D		0001110

ชนิดรหัสยูพีซี (Universal Product Code, UPC)

รหัสแถบยูพีซี (Universal Product Code, UPC) เป็นรหัสที่ได้รับการพัฒนาและนำมาใช้งานในครั้งแรกในปี ค.ศ. 1949 โดย Mr. Norm Woodland และ Mr. Barnard Silvers และได้มีการทดสอบและปรับปรุงให้ใช้งานได้อย่างสมบูรณ์ในปี ค.ศ. 1973 โดย Uniform Code Council (UCC) เพื่อใช้ในสินค้าอุปโภคและบริโภคในประเทศสหรัฐอเมริกา และเป็นรหัสที่ได้ความนิยมใช้งานอย่างมากใน 2 ประเทศเท่านั้น คือประเทศสหรัฐอเมริกาและแคนาดา ซึ่งจนกระทั่งปัจจุบันได้มีการพัฒนาไปในอีกหลายรูปแบบตามความต้องการของผู้ใช้งาน เช่น รหัสยูพีซี-เอ (เป็นรหัสพื้นฐานของยูพีซี) , รหัสยูพีซี-บี , รหัสยูพีซี-ซี , รหัสยูพีซี-ดี , รหัสยูพีซี-อี เป็นต้น

รหัสยูพีซี-เอ (UPC-A)

เป็นรหัสพื้นฐานของรหัสยูพีซีที่ได้ถูกสร้างขึ้นเป็นแบบแรก จึงมีโครงสร้างที่เป็นพื้นฐานของรหัสยูพีซีแบบอื่นๆ โดยโครงสร้างพื้นฐานของรหัสยูพีซีจะมีส่วนประกอบที่สำคัญ ดังนี้

1. ส่วนของแถบข้อมูล (Character Bar) ประกอบด้วย แถบข้อมูลซ้ายและแถบข้อมูลขวา
2. ส่วนของแถบคุม (Guard Bar) ประกอบด้วย แถบกั้นซ้าย แถบกั้นขวา และแถบกั้นกลาง
3. บริเวณขอบเผื่อ (Quiet Zone) เป็นบริเวณที่มีไว้เพื่อเป็นพื้นที่ให้หัวอ่านวางเพื่อเริ่มต้นหรือสิ้นสุดการอ่าน ซึ่งบางกรณีรหัสแถบจะละส่วนนี้ไว้ไม่แสดงให้เห็น

นอกจากส่วนประกอบของรหัสยูพีซีนี้แล้ว ในการแทนรหัสยูพีซียังมีข้อกำหนดที่สำคัญที่ใช้เป็นมาตรฐานของรหัสยูพีซีดังนี้

1. สามารถแทนข้อมูลตัวเลข 0 ถึง 9 เท่านั้น โดยที่ข้อมูลด้านขวาและด้านซ้ายจะต้องมีจำนวนเท่ากัน และข้อมูลแต่ละตัวจะแทนด้วยแถบ 4 แถบ คือ แถบดำและแถบขาวอย่างละ 2 แถบ

2. ความกว้างของแถบแต่ละแถบจะมีทั้งหมด 4 ขนาดที่เป็นสัดส่วนกัน คือ ขนาดเล็กสุดจะมีขนาดเป็น 1 และที่เหลือจะมีขนาดเป็น 2 , 3 และ 4 เท่าของขนาดเล็กสุด ซึ่งค่าขนาดเล็กสุดนี้จะเป็นค่ามาตรฐานที่กำหนดให้มีค่าเท่ากับ 1 ส่วนย่อย

3. ความกว้างของแถบที่ใช้แทนรหัสข้อมูล (ตัวเลข) แต่ละตัวจะมีความกว้างเท่ากันทุกตัว คือ 7 ส่วนย่อยจะแทนด้วยเลขฐานสอง 1 บิต โดยเลขฐานสอง "0" แทนแถบขาว ส่วนเลขฐานสอง "1" แทนแถบดำ

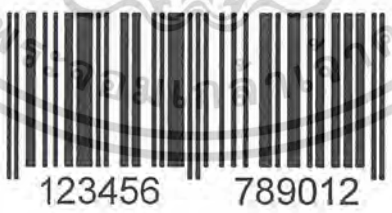
4. แถบข้อมูลด้านซ้ายและด้านขวา จะมีลักษณะการแทนด้วยรหัสฐานสองที่เป็นตรงข้าม หรือคอมพลีเมนต์ (Complement) สำหรับข้อมูลตัวเดียวกันแต่ขนาดความกว้างยังคงเหมือนกัน

5. แถบคุมซ้ายและแถบคุมขวาจะมีความกว้าง 3 หน่วยย่อย คือ แถบดำ 2 แถบ และแถบขาว 1 แถบ โดยแต่ละแถบกว้าง 1 ส่วนย่อย และจัดเรียงเป็น ดำ-ขาว-ดำ สำหรับแถบคุมกลางจะมีความกว้าง 5 ส่วนย่อย คือ แถบดำ 3 แถบและแถบขาว 2 แถบ ซึ่งจัดเรียงเป็น ดำ-ขาว-ดำ-ขาว-ดำ

รหัสยูพีซีชนิดนี้มีจำนวนของรหัสที่แน่นอนคือ 12 ตัว คือรหัสข้อมูลซ้าย 6 ตัว รหัสข้อมูลขวา 5 ตัว และรหัสตรวจสอบอีก 1 ตัว โดยการแทนรหัสของรหัสยูพีซีจะใช้ค่าดังแสดงในตารางที่ 2.5 และตัวอย่างรหัสยูพีซีในรูปที่ 2.17 ซึ่งนอกจากส่วนประกอบที่กล่าวมาแล้วยังมีข้อกำหนดพิเศษที่ไม่ได้กำหนดเป็นมาตรฐานแต่มีการกำหนดขึ้นมาใช้งานดังนี้ คือ

1. ข้อมูลด้านซ้ายตัวแรกจะเป็นตัวเลขที่บอกถึงลักษณะงานที่นำรหัสแถบยูพีซีนี้ไปใช้จะเรียกข้อมูลหรือตัวเลขนี้ว่า ตัวเลขบอกชนิดของสินค้า (Product Code) โดยจะแสดงการแทนค่าในตารางที่ 2.5

2. ข้อมูลด้านขวาตัวสุดท้ายจะเป็นตัวเลขรหัสตรวจสอบ (Check Digit) ซึ่งไม่ได้มีการกำหนดวิธีการหาค่ารหัสตรวจสอบนี้ว่าเป็นอย่างไร แต่ที่มีใช้งานอยู่คือ นำค่าผลรวมของตัวเลขรหัสข้อมูลทุกตัวมาบวกกับรหัสตรวจสอบแล้วจะต้องได้ผลลัพธ์ของตัวเลขหลักหน่วยเป็นศูนย์ เช่น ถ้าค่าผลรวมของตัวเลขรหัสข้อมูลเป็น 35 รหัสตรวจสอบจะต้องเป็นตัวเลข 5 เพื่อให้ได้ผลลัพธ์เป็น 40 ซึ่งเลขหลักหน่วยเป็น 0



รูปที่ 2.17 แสดงรหัสแถบยูพีซี-เอ

ตารางที่ 2.5 แสดงการแทนรหัสแถบยูพีซี

รหัส	ขนาดความกว้างแถบ	ข้อมูลด้านซ้าย	ข้อมูลด้านขวา
0	3-2-1-1	0001101	1110010
1	2-2-2-1	0011001	1100110
2	2-1-2-2	0010011	1101100
3	1-4-1-1	0111101	1000010
4	1-1-3-2	0100011	1011100
5	1-2-3-1	0110001	1001110
6	1-1-1-4	0101111	1010000
7	1-3-1-2	0111011	1000100
8	1-2-1-3	0110111	1001000
9	3-1-1-2	0001011	1110100
การ์ดซ้าย	1-1-1	101	----
การ์ดขวา	1-1-1	----	101
การ์ดกลาง	1-1-1-1-1	----	----

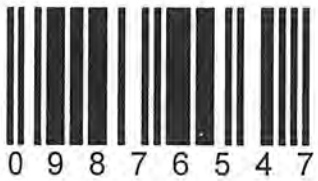
ตารางที่ 2.6 ตารางแสดงการแทนตัวเลขบอกลักษณะงาน

รหัสตัวเลข	ลักษณะการใช้งาน
0	รหัสยูพีซีทั่วไป
2	อาหารหรือสิ่งของที่มีน้ำหนักไม่แน่นอน
3	ยาและสิ่งที่เกี่ยวข้องกับสาธารณสุข
4	รายการที่ไม่เกี่ยวข้องกับอาหาร
5	สำหรับคูปอง
อื่นๆ	เตรียมไว้สำหรับอนาคต

รหัสยูพีซี-อี (UPC-E)

เป็นรหัสที่ใช้การแทนค่ารหัสเหมือนกับรหัสยูพีซี-เอ แต่โครงสร้างจะต่างกันเล็กน้อยคือ รหัสยูพีซี-อีจะมีจำนวนข้อมูลเพียง 6 ตัว ซึ่งเสมือนการตัดเอาเฉพาะข้อมูลด้านซ้ายของรหัสยูพีซี-เอ มาใช้ คือ มีเฉพาะแถบคุมซ้าย แถบข้อมูล และแถบคุมกลาง ดังแสดงตัวอย่างรหัสยูพีซี-อี ในรูปที่ 2.18 ในการใช้งานแล้ว รหัสยูพีซี-อี เป็นรหัสที่ได้รับความนิยมไม่น้อยไปกว่ารหัสยูพีซี-เอ เลย เพราะ ความจริงข้อมูลก็ยังคงมีเท่าเดิมแต่รหัสยูพีซี-อี นี้เป็นแบบที่ตัดตัวเลข 0 ออกเพื่อให้สามารถใช้ในสินค้าที่มีพื้นที่ในการติดรหัสแถบน้อย เช่น ซองบุหรี่ ซึ่งโดยโครงสร้างของข้อมูลจะประกอบด้วยรหัสตัวเลขบอกชนิดของสินค้า 1 ตัว

และรหัสข้อมูลอีก 5 ตัว ซึ่งเป็นข้อมูลที่ได้มาจากข้อมูลจริง 10 ตัวที่ตัดเลข 0 ออก ดังตัวอย่าง เช่น ข้อมูล 5680000021 ก็ได้เป็น 56821 เป็นต้น



รูปที่ 2.18 แสดงรหัสแถบยูพีซี-อี

รหัสยูพีซี-บี (UPC-B)

เป็นรหัสยูพีซีแบบที่พัฒนามาจากรหัสยูพีซี-เอ เพื่อใช้ในงานด้านยาและสาธารณสุขแห่งชาติของประเทศสหรัฐอเมริกา โดยโครงสร้างของรหัสที่แตกต่างก็เพียงแค่รหัสยูพีซี-บี จะไม่มีรหัสตรวจสอบ คือ รหัสตัวสุดท้ายของข้อมูลด้านขวามือจะไม่ใช่รหัสตรวจสอบแต่จะเป็นรหัสข้อมูล

รหัสยูพีซี-ซี (UPC-C)

เป็นรหัสยูพีซีแบบที่พัฒนาขึ้นมาเพื่อใช้ในโรงงานอุตสาหรรมเนื่องจากเดิมโรงงานอุตสาหกรรมยังมิได้มีการนำรหัสแถบไปใช้งาน (ใช้ตัวเลขธรรมดา) จึงได้มีการพัฒนารหัสยูพีซี-ซี ขึ้นมารองรับความต้องการ โดยโครงสร้างของรหัสที่แตกต่างจากรหัสยูพีซีแบบมาตรฐาน คือ จะมีรหัสข้อมูล 12 ตัวกับรหัสตรวจสอบและรหัสบอกชนิดสินค้า รวมทั้งหมดเป็น 14 ตัว โดยการเรียงลำดับของรหัสจะแสดงในตาราง 2.5

รหัสยูพีซี-ดี (UPC-D)

เป็นรหัสยูพีซีแบบที่มีจำนวนข้อมูลไม่แน่นอนโดยโครงสร้าง แล้วจะมีการจัดเรียงรหัสและจำนวนข้อมูลในส่วนแรกเหมือนหรือเท่ากับรหัสยูพีซี-เอ แต่จะมีรหัสข้อมูลที่ตามหลังเพิ่มขึ้นมา ซึ่งจำนวนของรหัสข้อมูลที่ตามหลังนี้ไม่ได้มีการกำหนดจำนวนที่แน่นอนไว้ จึงทำให้มีจำนวนเท่าไรก็ได้ตามความต้องการของงานที่จะนำรหัสนี้ไปใช้

ตารางที่ 2.7 แสดงการจัดเรียงของรหัส

รหัสยูพีซีแบบ	รูปแบบการจัดเรียงข้อมูล
A	PXXXXXXXXXXC
B	PXXXXXXXXXXX
C	XPXXXXXXXXXXCX
D	PXXXXXXXXXXCXX
E	XXXXX

หมายเหตุ X หมายถึง รหัสข้อมูล

P หมายถึง รหัสบอกชนิดสินค้า

C หมายถึง รหัสตรวจสอบ

ชนิดรหัสเอียน (EAN = European Article Numbering)

เป็นรหัสที่พัฒนามาจากพื้นฐานของรหัสยูพีซี โดยพัฒนาขึ้นมาเมื่อปี ค.ศ. 1976 เพื่อให้เป็นมาตรฐานของรหัสแถบสำหรับสินค้าที่ใช้ในประเทศแถบยุโรป แต่ต่อมามีการนำมาใช้งานอย่างกว้างขวางและได้มีการกำหนดมาตรฐานที่ทำให้สามารถใช้รหัสแถบได้ทั่วโลก ยกเว้นประเทศสหรัฐอเมริกาและแคนาดาที่ใช้รหัสยูพีซี โดยรหัสเอียนจะมีการกำหนดรหัสประเทศต่างๆ ที่ไม่ซ้ำกันทำให้สามารถระบุได้ว่าสินค้าที่ติดรหัสแถบนั้นเป็นสินค้าของประเทศใด รหัสแถบชนิดนี้จะใช้จำนวนตัวเลข 5 หรือ 10 หลัก โดยมีเพียงอักขระ 0 ถึง 9 เท่านั้นที่ใช้และแทนด้วยรหัสไบนารี่ รหัสเอียนนี้มี 4 ระดับความกว้าง คือ ในแต่ละแถบเส้นหรือช่องว่างจะมีระดับความกว้าง 1 , 2 , 3 หรือ 4 โดยให้แถบเส้นแทนด้วยไบนารี่ "1" ในขณะที่ช่องว่างแทนด้วยไบนารี่ "0" ตัวอย่างเช่น รหัส 00011 ก็จะถูกแทนด้วยช่องว่างที่มีความกว้าง 3 ส่วนและตามด้วยแถบเส้นที่มีความกว้าง 2 ส่วน

อักขระแต่ละตัวถูกสร้างขึ้นด้วยเลขไบนารี่ 7 บิต โดยรหัสแถบหนึ่งๆ จะประกอบด้วยรหัสกั้นหน้า รหัสกั้นกลางและรหัสกั้นหลัง (Start , Center and End Guard Bar) รหัสที่อยู่ท้ายด้านซ้ายของรหัสกั้นกลางถูกเข้ารหัสโดยการใช้คอลัมน์ด้านซ้ายมือดังแสดงในตารางที่ 2.7 ดังนั้นอักขระที่อยู่ทางขวามือของรหัสกั้นกลางก็จะเข้ารหัสโดยคอลัมน์ด้านขวามือ ความแตกต่างระหว่าง 2 คอลัมน์ทางด้านซ้ายมือนั่นคือ A จะใช้เข้ารหัสข้อมูลกับข้อความที่มีจำนวนเป็นคี่ (Odd Parity) แล้วจำนวนเป็นคู่ (Even Parity) ก็จะใช้คอลัมน์ B

ตารางที่ 2.8 ตารางแสดงอักขระของรหัสเอียน

เลขที่	ซ้ายมือ	ซ้ายมือ	ขวามือ
0	0001101	0100111	1110010
1	0011001	0110011	1100110
2	0010011	0011011	1101100
3	0111101	0100001	1000010
4	0100011	0011101	1011100
5	0110001	0111001	1001110
6	0101111	0000101	1010000
7	0111011	0010001	1000100
8	0110111	0001001	1001000
9	0001011	0010111	1110100

ซึ่งรหัสเอียนได้มีการกำหนดจำนวนข้อมูลที่แน่นอนไว้เป็น 2 แบบ โดยมีรายละเอียดและโครงสร้างเป็นดังนี้

รหัสเอียน-13

รหัสเอียนเป็นรหัสที่มีพื้นฐานการแทนรหัสคล้ายกับรหัสยูพีซี แต่มีข้อกำหนดพิเศษที่เพิ่มขึ้นมา โดยโครงสร้างการการแทนรหัสนั้นจะแทนเช่นเดียวกับรหัสยูพีซี แต่ที่พิเศษคือข้อมูลด้านซ้ายจะสามารถแทนได้ 2 ลักษณะ คือ แบบคู่ (Even) และแบบคี่ (Odd) ดังแสดงในตารางที่ 2.7 ซึ่งรหัสข้อมูลที่แทนในรหัสเอียนแบบนี้จะมีเพียง 12 ตัวและรหัสตรวจสอบอีก 1 ตัว โดยในรหัสข้อมูลจะแทนเป็นรหัสแถบเพียง 11 ตัว ส่วนรหัสตัวที่เหลือจะเป็นรหัสเติมหน้า (Flag Digit) ที่กำหนดขึ้นมาจากการจัดเรียง ลักษณะการแทนข้อมูลในส่วนข้อมูลด้านซ้าย ซึ่งเดิมถ้าเป็นรหัสยูพีซีแล้วข้อมูลด้านซ้ายจะแทนในลักษณะมีพาริตี้เป็นแบบคู่เท่านั้น แต่สำหรับรหัสเอียนข้อมูลด้านซ้ายจะมีการแทนในลักษณะพิเศษที่จัดเรียงตามตารางที่ 2.7 เพื่อใช้กำหนดค่ารหัสเติมหน้า

รหัสเอียน-13 เป็นรหัสที่แทนตัวเลข 13 ตัวดังแสดงในรูปที่ 2.19 ได้แบ่งรหัสข้อมูลออกเป็น 4 ส่วน ดังมีรายละเอียดแต่ละส่วน ดังนี้

1. รหัส 3 ตัวแรก คือ รหัสประเทศของผู้จดทะเบียน หรือผู้ผลิตสินค้า (Country Code) ดังแสดงการแทนรหัสของประเทศต่างๆ จากตัวอย่างรหัสเอียนในรูปที่ 2.19 เป็นรหัส 855 หมายถึงประเทศไทย

2. รหัส 4 ตัวต่อมา คือ รหัสทะเบียนของโรงงานผู้ผลิตสินค้า (Manufacture Identify Code) ซึ่งจะเป็นรหัสที่ใช้บอกว่าสอนค้าที่ติดรหัสแถบนั้นเป็นสินค้าที่ผลิตมาจากโรงงานใด ซึ่งหมายเลขทะเบียนนี้ในแต่ละประเทศจะต้องมีการไปขอจดทะเบียนหรือขอเป็นสมาชิกขององค์กรที่จัดการรหัสแถบภายในประเทศนั้นๆ สำหรับรหัสแถบของประเทศไทย สภาอุตสาหกรรมแห่งประเทศไทยเป็นนายทะเบียนที่ทำหน้าที่ผู้ควบคุมดูแล จากรูปที่ 2.19 คือ รหัส 2047

3. รหัส 5 ตัวถัดมา คือ รหัสของสินค้า (Product Code) เป็นรหัสที่ใช้บอกรายละเอียดของสินค้า เช่น วัน เดือน ปีที่ผลิต สี ชนิด รุ่น เป็นต้น โดยรหัสนี้ทางโรงงานผู้ผลิตจะเป็นผู้กำหนดเอง แต่จะต้องแจ้งให้นายทะเบียนของแต่ละประเทศทราบ

4. รหัสตัวสุดท้าย คือ รหัสตรวจสอบ (Check Code) เป็นรหัสที่ใช้ในการตรวจสอบเพื่อให้เกิดความถูกต้องในการอ่านรหัสแถบ ซึ่งค่านี้จะใช้เป็นตัวตรวจสอบรหัส 12 ตัวก่อนหน้านี้ เพราะถ้ารหัสตรวจสอบที่อ่านได้ผิดพลาดก็แสดงว่ารหัสที่อ่านได้ทั้งหมดผิดไม่สามารถนำมาใช้สื่อความหมายได้ จากรูปที่ 2.19 โดยรหัสตรวจสอบนั้นมีวิธีในการหาค่าดังนี้ คือ

1. หาผลรวมของรหัสข้อมูลหลักคี่

2. หาผลรวมของรหัสข้อมูลหลักคู่คูณด้วย 3

3. หาผลรวมจากข้อ 1 และข้อ 2

4. ค่ารหัสตรวจสอบ ได้จากการนำค่าตัวเลขที่น้อยที่สุดไปบวกกับผลลัพธ์ในข้อ 3 แล้วได้หลักหน่วยเป็นเลข 0

ตารางที่ 2.9 ตารางแสดงการจัดเรียงของข้อมูลเพื่อหาค่ารหัสเติมหน้า

รหัสเติมหน้า	การจัดเรียงรหัสข้อมูล
0	OOOOOO
1	OOEOEE
2	OOEEOE
3	OOEEEO
4	OEOOEE
5	OEEOOE
6	OEEEOO
7	OEOEOE
8	OEOEEO
9	OEEOEO

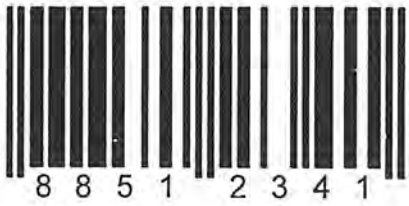
หมายเหตุ O หมายถึง รหัสข้อมูลที่มีพาริตี้เป็นคี่

E หมายถึง รหัสข้อมูลที่มีพาริตี้เป็นคู่

รูปที่ 2.19 แสดงรหัสเอียน-13

รหัสเอียน-8

สำหรับรหัสเอียน-8 จะมีลักษณะการแทนรหัสเหมือนกับรหัสเอียน-13 แต่ที่ต่างจากรหัสเอียน-13 คือ รหัสเอียน-8 จะไม่มีรหัสเติมหน้า โดยรหัสเอียน-8 เป็นรหัสที่มีจำนวนข้อมูล 8 ตัว เป็นข้อมูลด้านซ้ายและขวาอย่างละ 4 ตัว ซึ่งรหัสเอียน-8 เป็นรหัสที่เหมาะสำหรับใช้ในธุรกิจขนาดเล็ก หรือใช้ในสินค้าที่มีพื้นที่ในการติดรหัสแถบน้อย ซึ่งในรหัสเอียน-8 จะยังคงมีการกำหนดรหัสประเทศไว้ในแถบรหัส ดังแสดงในรูปที่ 2.20



รูปที่ 2.20 แสดงรหัสเอียน-8

ลักษณะของรหัสแถบที่ดี

รหัสแถบที่ดีควรที่จะมีคุณสมบัติดังต่อไปนี้

1. สามารถตรวจสอบความถูกต้องภายในรหัสได้
2. ความกว้างและจำนวนของแถบต่อรหัสควรจะคงที่
3. สามารถใช้แทนตัวเลขปนตัวอักษรได้ครบ
4. มีโครงสร้างแบบง่ายๆ
5. การอ่านด้วยความเร็วที่ต่างกัน ควรได้ค่าถูกต้องเสมอ
6. มีความหนาแน่นของข้อมูลต่อแถบความกว้างสูง

เครื่องอ่านรหัสแถบชนิดไม่สัมผัสกับรหัสแถบ (Non-contact Scanner)

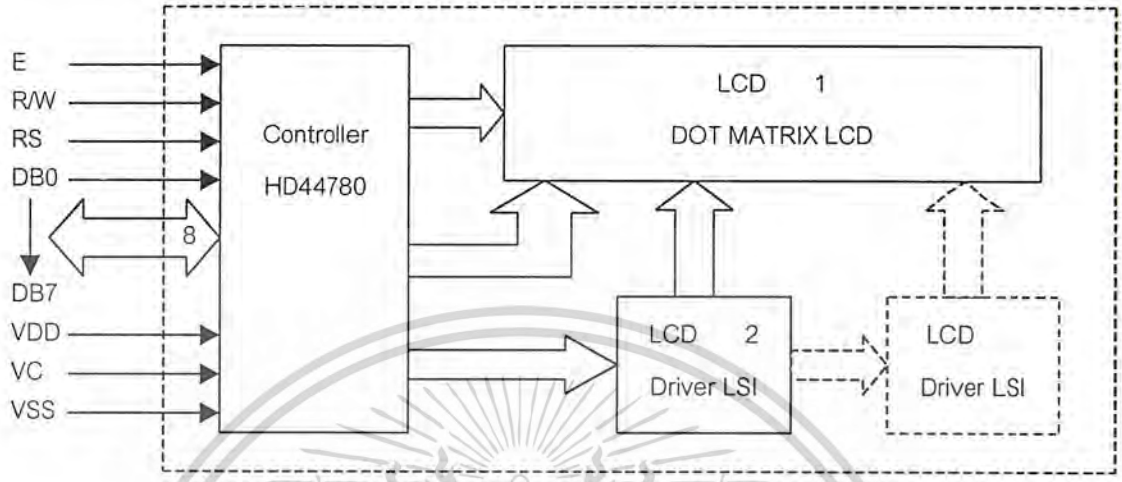
เครื่องอ่านรหัสแถบประเภทนี้เป็นเครื่องอ่านประเภทที่สามารถอ่านรหัสได้โดยหัวอ่านอยู่ห่างจากแถบรหัส และยังจะเป็นการลดข้อผิดพลาดจากผลของระยะโฟกัสของลำแสงอีกด้วย นอกจากนี้ยังลดปัญหาจากการรูดหัวอ่านไม่สม่ำเสมอ ซึ่งเกิดในเครื่องอ่านรหัสแถบชนิดสัมผัสโดยตรง (Contact Scanner)

เครื่องอ่านชนิดไม่สัมผัสกับรหัสแถบยังมีอีก 2 ประเภท คือ

- แบบแอกทีฟ (Active Non-contact Scanner) ซึ่งใช้หลักการของแสงเลเซอร์ (Laser)
- แบบพาสสีฟ (Passive Non-contact Scanner) ซึ่งใช้อุปกรณ์ตรวจวัดแสงชนิดกล้อง CCD ขนาดเล็ก หลักการคือจะใช้แสงแฟลชฉายลงบนรหัสแถบ จะทำให้เกิดการสะท้อนกลับไปยังส่วนรับแสง การที่เครื่องอ่านชนิดนี้ใช้แสงแฟลชเนื่องจากต้องการความคมชัดและลึกของภาพให้มีช่องกว้างมากขึ้น หรือเพื่อเพิ่มระยะห่างในการอ่านรหัสแถบให้มีช่วงกว้างขึ้น เครื่องอ่านแบบนี้จะมีข้อจำกัดที่ความกว้างของแถบรหัส

การประยุกต์ใช้ส่วนแสดงผลชนิด LCD module กับ MCS-51

ปัจจุบัน LCD ที่มีขายในท้องตลาดส่วนใหญ่จะประกอบเป็นโมดูลเพื่อให้สะดวกในการใช้งาน โดยจะมีส่วนประกอบทั่วไปดังในรูปที่ 2.21



รูปที่ 2.21 ส่วนประกอบของ LCD module

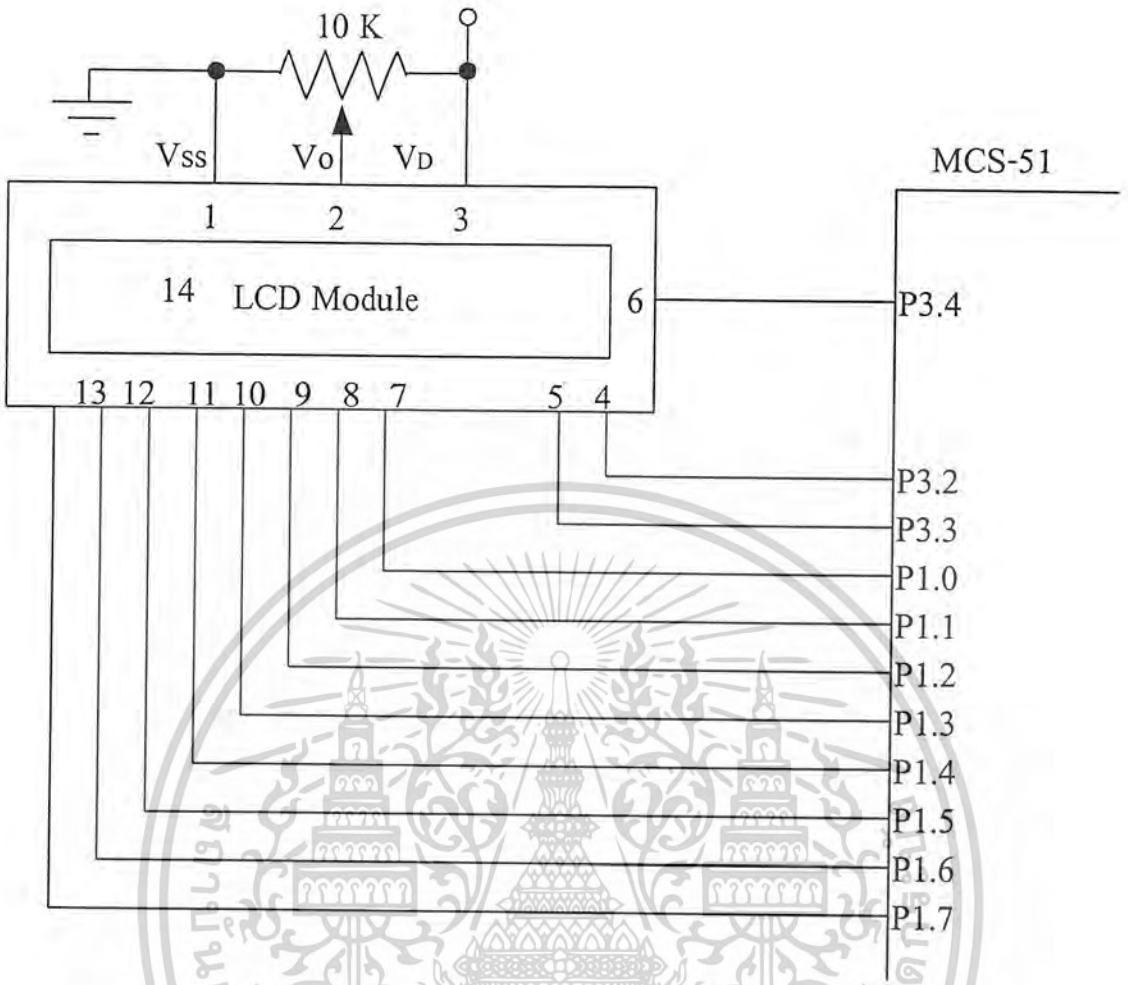
LCD module ที่จะกล่าวต่อไปนี้ จะกล่าวถึงเฉพาะ character LCD module ซึ่งจะเรียกย่อๆว่า LCM โดยมีส่วนประกอบที่สำคัญดังนี้

1. Dot Matrix LCD : เป็นส่วนที่ทำหน้าที่แสดงผล ซึ่งใช้หลักการหักเหของแสงผ่านผลึกโดยจะประกอบไปด้วยจุด (pixel) จำนวนมากที่สามารถบังคับให้ติดหรือดับได้ทุกจุด
2. Driver : เป็นวงจรที่ใช้ขับ LCD ส่วนใหญ่จะใช้ชิปเบอร์ HD44110H
3. Controller : เป็นส่วนที่ใช้ควบคุมการทำงานทั้งหมดของ LCD module โดยจะรับข้อมูลจากภายนอกมาจัดการให้ LCD แสดงผลในรูปแบบต่างๆ ส่วนใหญ่จะใช้ชิปเบอร์ HD44780 ซึ่งมีใช้งานในแบบ character LCD module

การใช้ LCD module ผู้ใช้เพียงแต่ศึกษาและทำความเข้าใจในส่วนคอนโทรลเลอร์ของ LCM เท่านั้น เพราะส่วนนี้เป็นส่วนที่รับข้อมูลที่ต้องการแสดงผลจากวงจรภายนอก และควบคุมการทำงานทั้งหมดของ LCM โดยจะกล่าวถึงเฉพาะชิปที่เป็นคอนโทรลเลอร์เบอร์ HD44780 เท่านั้น ส่วนชิปคอนโทรลเลอร์เบอร์อื่นๆนั้น ส่วนใหญ่ก็จะมีการใช้งานที่คล้ายกับเบอร์นี้มาก

ชิปคอนโทรลเลอร์เบอร์ HD44780 เป็นชิปของบริษัท Hitachi สามารถต่อใช้งานเพื่อควบคุม LCM กับไมโครคอนโทรลเลอร์หรือไมโครโปรเซสเซอร์ได้ทั้งแบบ 4 บิต และ 8 บิต

เนื่องจากไมโครคอนโทรเลอร์ตระกูล MCS-51 มีดาต้าบัสขนาด 8 บิต ดังนั้นเราจะกล่าวถึงเฉพาะการติดต่อในแบบ 8 บิต เท่านั้น ตัวอย่างวงจรการอินเทอร์เฟส ระหว่าง MCS-51กับ LCM มีแสดงในรูปที่ 2.22



รูปที่ 2.22 การอินเทอร์เฟส ระหว่าง MCS-51กับ LCM

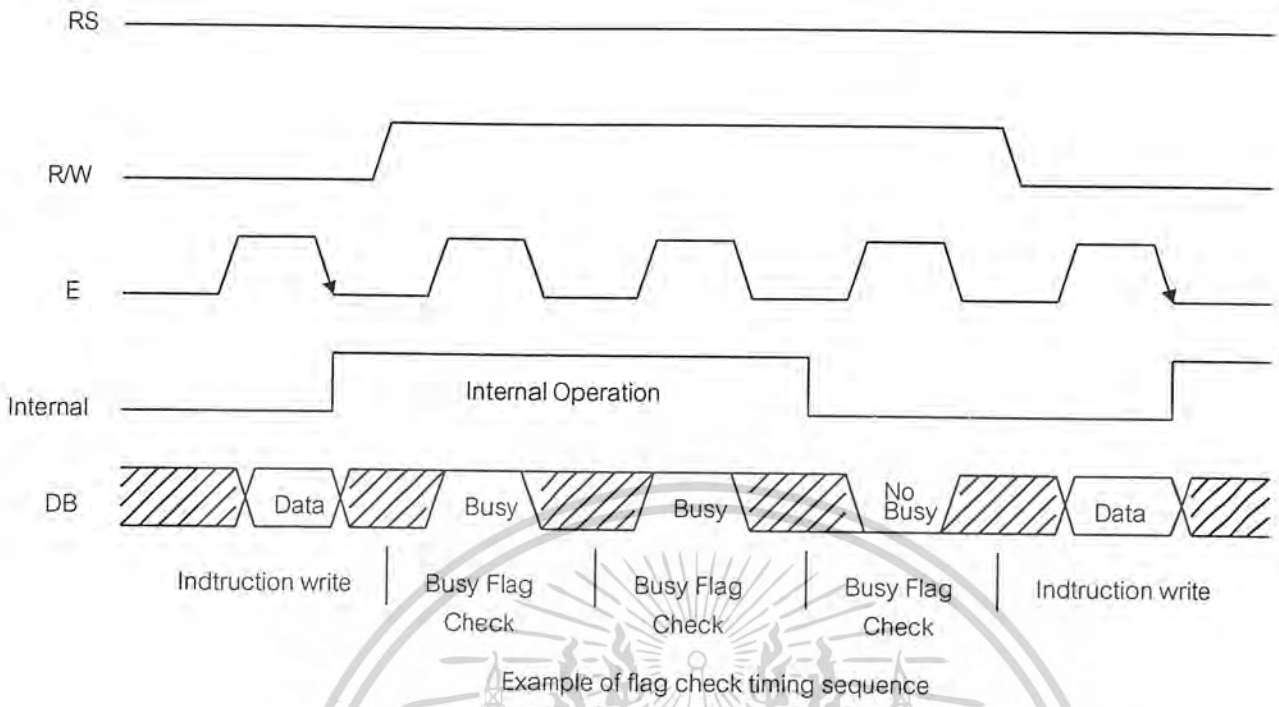
จากในรูป จะเห็นว่า LCM ติดต่อกับ MCS-51 โดย

- ใช้ขา P1.0 – P1.7 เป็นดาต้าบัส (DB0-DB7) ในการติดต่อ
- ใช้ขา P3.2 เป็นสัญญาณ RS
- ใช้ขา P3.3 เป็นสัญญาณ R/W
- ใช้ขา P3.3 เป็นสัญญาณ EN (E)

การทำความเข้าใจการใช้งาน LCM จำเป็นที่จะต้องทราบรายละเอียดดังต่อไปนี้เสียก่อน คือ

1. LCM มีหลายขนาด แต่ทุกขนาดจะมีคำสั่งที่ใช้ในการควบคุมเหมือนกัน ต่างกันเพียงขนาดของหน่วยความจำในการแสดงผล หรือ DDRAM (Data Display RAM) เท่านั้น
2. แผนผังเวลา (timing diagram) ในการติดต่อกับ LCM
3. คำสั่งในการควบคุม LCM

แผนผังเวลาในการติดต่อกับ LCM มีแสดงดังในรูปที่ 2.23



รูปที่ 2.23แผนผังเวลาในการติดต่อกับ LCM

จากในรูปที่ 2.23 แสดงเฉพาะในช่วงที่สัญญาณ RS เป็น 0 เท่านั้น ส่วนในช่วงที่ สัญญาณ RS เป็น 1 นั้น จะมีแผนผังเวลาเหมือนกัน รายละเอียดของแต่ละสัญญาณมีดังนี้

1. RS : เนื่องจากในชิปคอนโทรเลอร์มีรีจิสเตอร์อยู่ 2 ประเภท คือ command register หรือ instruction register และ data register โดยรีจิสเตอร์ทั้งสองจะถูกเลือกโดยสัญญาณ RS ดังนี้

   สัญญาณ RS = 0 หมายถึงเลือกใช้ data register

   สัญญาณ RS = 1 หมายถึงเลือกใช้ instruction register

2. R/W (Read/Write) เป็นสัญญาณที่ใช้เลือกว่าจะทำการเขียนหรืออ่านข้อมูลจาก LCM โดย

   สัญญาณ R/W = 0 หมายถึงต้องการอ่านข้อมูลจาก LCM

   สัญญาณ R/W = 0 หมายถึงต้องการเขียนข้อมูลไปยัง LCM

3. E (Enable) มีรายละเอียดดังแสดงในตารางที่ 2.10

ตารางที่ 2.10 แสดงการทำงานของสัญญาณ E

RS	R/W	E	OPERATION
0	0		Write instruction code
0	1		Read busy flag and address counter
1	0		Write data
1	1		Read data

จากแผนผังเวลาในการตรวจสอบ busy flag และจากตารางจะเห็นว่าในการเขียนรหัสคำสั่ง (instruction code) ทุกครั้ง

- RS และ RW ต้องมีค่าเป็น 0 และส่งข้อมูลไปในขณะที่สัญญาณ E เปลี่ยนจาก 1 เป็น 0 ในการเขียนข้อมูลทุกครั้ง

- RS = 1 และ RW = 0 และส่งข้อมูลไปขณะที่สัญญาณ E เปลี่ยนจาก 1 เป็น 0 ในการอ่าน busy flag และ address counter ทุกครั้ง

- RS = 0 , RW = 1 และรับข้อมูลเข้ามาในขณะที่สัญญาณ E เป็น 1 ในการอ่านข้อมูลทุกครั้ง

- RS = 1 , RW = 1 และรับข้อมูลเข้ามาในขณะที่สัญญาณ E เป็น 1

จากแผนผังเวลา (ในรูปที่ 2.23) จะเห็นว่า DB0-DB7 จะมีสถานะเป็น high impedance เมื่อสัญญาณ E มีสถานะเป็น 0 ดังนั้นในการใช้งานจริงเมื่อเราเลิกติดต่อกับ LCM ควรจัดการส่งสัญญาณ E ให้มีค่าเป็น 0 เพื่อให้ P1.0-P1.7 ของ MCS-51 มีสถานะเป็น high impedance ด้วยทั้งนี้เพื่อเราจะได้ใช้งาน P1.0-P1.7 อย่างอื่นได้ด้วย และเนื่องจากเวลาในการทำงานคำสั่งต่าง ๆ ของ HD44780 ไม่เท่ากัน เราจึงควรที่จะตรวจสอบสัญญาณ busy flag ทุก ๆ ครั้งก่อนที่จะทำการเขียนข้อมูลใด ๆ ลงไป ทั้งนี้เพื่อป้องกันการเขียนข้อมูลทับนั้นเอง

คำสั่งที่ใช้ควบคุมการทำงาน

ชิปคอนโทรลเลอร์ HD44780 มีการทำงานได้หลายอย่างขึ้นกับคำสั่งที่ได้รับเข้ามา โดคำสั่งทั้งหมดมีดังแสดงในตารางที่ 2.11

รายละเอียดของคำสั่ง HD44780

CLEAR DISPLAY

คำสั่งนี้เป็นการเขียนช่องว่างหรือ space (ASCII 20H) เข้าไปใน DD RAM (display data RAM) ทั้งหมด และทำการเซต DD RAM แอดเดรสให้เป็นศูนย์ตัวเคอร์เซอร์จะกลับไอยู่งตำแหน่งมุมบนซ้ายมือของจอภาพ เซต I/D = 1 , S ไม่มีการเปลี่ยน

RETURN HOME

คำสั่งนี้จะทำการเซต DD RAM แอดเดรสให้เป็นศูนย์ ตัวเคอร์เซอร์จะกลับไปอยู่งตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจอภาพไม่เปลี่ยน

ENTRY MODE SET

BIT I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้DD RAM แอดเดรสเพิ่มขึ้นหนึ่งหรือลดลงหนึ่ง โดย 1 = เพิ่ม , 0 = ลดลงหนึ่ง

BIT S : เป็นตัวกำหนดการแสดงผล ถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัวเคอร์เซอร์อยู่กับที่ ข้อมูลถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่กับที่ตัวเคอร์เซอร์จะถูกดันไปทางขวามือ

DISPLAY ON/OFF CONTROL

BIT D : เป็นบิตที่ใช้ควบคุมการเปิดปิดหน้าจอ โดยถ้า D = 1 จะ ON และ D = 0 จะ OFF

BIT C : ใช้แสดเคอร์เซอร์เมื่อให้บิต C = 1 และถ้าไม่ต้องการแสดงเคอร์เซอร์ก็ให้บิต C = 0 โดยตัวเคอร์เซอร์จะอยู่ที่แถวที่ 8 ในแบบ 5x7 จุด และจะอยู่แถวที่ 11 ในแบบ 5x10 จุด

BIT B : เป็นบิตที่ใช้เซตการกะริบของเคอร์เซอร์ โดย B = 1 มีการกระพริบ B = 0 ไม่มีการะพริบมีระยะเวลาการกะพริบประมาณ 379.2 ms

CURSOR OF DISPLAY SHIFT

เป็นคำสั่งกำหนดให้ตำแหน่งเคอร์เซอร์หรือข้อมูลไปปรากฏทางซ้ายหรือขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่านโดย

S/C	R/L	
0	0	ทำการย้ายเคอร์เซอร์ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้ายเคอร์เซอร์ไปจากตำแหน่งเดิมไปทางขวามือ 1 ตำแหน่ง
1	0	เป็นการดันตัวอักษรที่ปรากฏไปทางซ้าย
1	1	เป็นการดันตัวอักษรที่ปรากฏไปทางขวามือ

FUCTION SET

BIT DL : เป็นการเซตการติดต่อว่าจะให้เป็นแบบ 8 บิต หรือ 4 บิต โดยถ้าต้องการติดต่อ 4 บิต DL= 0 และ 8 บิต DL = 1

N : เป็นการเซตบรรทัดการแสดงผล N = 0 แสดง 1 บรรทัด N = 1 แสดง 2 บรรทัด ในกรณีมากกว่า 2 บรรทัด ก็ให้เซต N = 1

F : เป็นการเซตขนาดจุดของการแสดงผลเป็นแบบ 5x7 หรือ 5x10 โดย F = 0 เป็นแบบ 5x7 และ F= 1 เป็นแบบ 5x10

SET CG RA ADDRESS

ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ display data RAM (DD RAM) จำหน่าย 80x8 บิต และ character generator RAM (CG RAM) จำนวน 512 บิต และ 7200 บิต คำสั่งนี้จะเป็นการเซตแอดเดรสใน CG RAM โดยต้องทำการเซตแอดเดรสก่อนเขียนหรืออ่านข้อมูลจาก CG RAM ด้วย

SET DD RAM ADDESS

คำสั่งเซตค่าแอดเดรสใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM (DD RAM คือ ส่วนที่แสดงผลหน้าจอ LCD) โดยจำนวนแอดเดรสที่จะเกิดขึ้นบนจอ LCD ขึ้นอยู่กับการเซนค่า N ด้วย

ถ้า N = 0 (1 บรรทัด) แอดเดรสจะอยู่ 00H-4FH

ถ้า N = 1 (2 บรรทัด) แอดเดรสจะอยู่ 00H-27H สำหรับบรรทัดที่ 1 40-67 สำหรับบรรทัดที่ 2

ตัวอย่างการจัดแอดเดรสของ DD RAM หน้าจอ LCD แบบ 16 ตัวอักษร 4 บรรทัด และ 0 ตัวอักษร 2 บรรทัดของ HDM-16416H , HDM-20216H

READ BUSY FLAG AND ADRESS

เป็นคำสั่งอ่านค่า busy flag ซึ่งจะเป็นตัวบอกว่าตัว DH44780 นี้อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูลแอดเดรสของ CG RAM หรือ DD RAM ด้วย

WRITE DATA TO CG หรือ DD RAM

เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและแอดเดรสจะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่เซตใน Entry Mode ข้อกำหนดที่จะรู้ว่าเป็นการเขียนข้อมูลของ CG RAM หรือ DD RAM ทำได้โดยการเซตแอดเดรสของ CG RAM หรือ DD RAM ขึ้นมาก่อนจะเขียนข้อมูล

READ DATA FROM CC OR DD RAM

เป็นคำสั่งอ่านค่าข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรจะใช้คำสั่งเซตแอดเดรสก่อนเพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD หรือ CG RAM

จากตารางการทำงานจะเห็นว่าการใช้งาน LCD โมดูลนั้นง่าย เพียงแต่เราส่งคำสั่งเริ่มแรกและเซตความต้องการขนาดตัวอักษร, เคอร์เซอร์ หลังจากนั้นก็สามารถเขียนตัวอักษรเข้าไปใน DD RAM ตามตารางตัวอักษรที่ให้มานั้น ก็จะเกิดอักษรในจอ LCD สามารถกำหนดตำแหน่งอักษรที่จะให้เกิดบนจอได้โดยการเซต DD RAM แอดเดรส ตามตารางที่ให้มาในหัวข้อ SET DD RAM ADDRESS ขอให้ทดสอบทำความเข้าใจกับตัวโปรแกรม จะเห็นว่ามีส่วนเริ่มต้นก็คือ ส่วนการอินนิเชียล LCD เพื่อกำหนดหน้าที่การทำงานต่าง ๆ

การเริ่มต้นใช้งาน LCM เพื่อให้ทำงานได้ถูกต้องจำเป็นต้องมีการรีเซตเสียก่อนเช่นเดียวกับชิปไมโครโปรเซสเซอร์ การรีเซตชิปคอนโทรลเลอร์ HD4780 มีด้วยกันสองวิธีดังนี้

1. เริ่มต้นใช้งานโดยวงจรที่ทำหน้าที่รีเซตภายใน ชิปคอนโทรลเลอร์ HD44780 สามารถรีเซตวงจรภายในได้เองในขณะเริ่มต้นใช้งาน โดยมันจะเริ่มทำงานด้วยคำสั่งดังจะได้กล่าวต่อ

ไป ในขณะทำงานคำสั่งเมื่อมีการรีเซต สัญญาณ busy flag (BF) จะอยู่ในสถานะ busy จนกระทั่งกระบวนการรีเซตภายในเสร็จสิ้น สัญญาณ busy flag จะมีค่าเป็น 1 เป็นเวลา 10 มิลลิวินาที ภายหลังจากที่ไฟเลี้ยงชิปมีค่าเป็น 4.5 โวลต์ คำสั่งที่ทำงานในช่วง busy หรือในช่วงรีเซตมีดังต่อไปนี้

- Display Clear
- Function Set DL = 1 : 8 Bit Long Interface Data

  N = 0 : 1 Line Display

  F = 0 : 5 * 7 Character Font
- Display ON/OFF

  Control D = 0 : Display Off

  C = 0 : Cursor Off

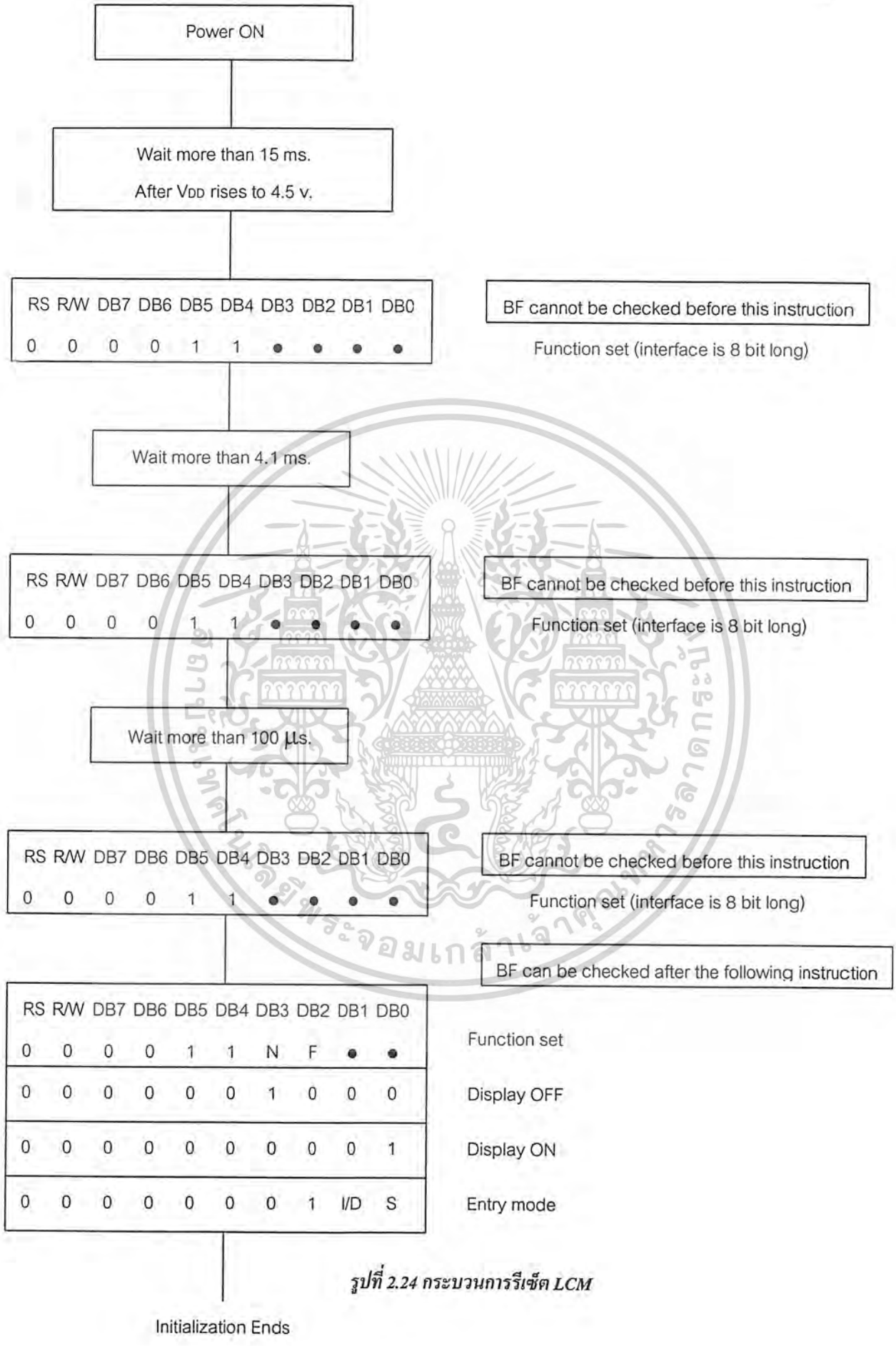
  B = 0 : Blink Off
- Entry Mode Set I/D = 1 : +1 Increment

  S = 0 : No Shift
- หลังจากช่วงนี้ผ่านไป เริ่มต้นใช้งานได้

หาก rise time ของแหล่งจ่าย (ช่วงเวลาที่แรงดันมีค่าเปลี่ยนจาก 0.2 โวลต์เป็น 4.5 โวลต์) มีค่าอยู่นอกช่วง 0.1-10 มิลลิวินาที หรือเมื่อไฟเลี้ยงที่จ่ายให้ LCM มีค่าตกลงมาต่ำกว่า 0.2 โวลต์ น้อยกว่า 1 มิลลิวินาที วงจรที่ทำหน้าที่รีเซตโดยอัตโนมัติภายในชิปจะไม่สามารถทำงานได้ถูกต้อง

ในกรณีที่การรีเซตของวงจรภายในชิปทำงานผิดพลาดอันเนื่องมาจากสาเหตุใดก็ตาม จำเป็นที่วงจรภายนอกจะต้องทำการรีเซต LCM เองเสียก่อน เพื่อให้การใช้งานไม่เกิดข้อผิดพลาด ทั้งนี้โดยการส่งคำสั่งไปควบคุม LCM เอง

2. การเริ่มต้นใช้งานโดยการส่งคำสั่งไปควบคุม LCM เอง เพื่อป้องกันมิให้เกิดการรีเซตผิดพลาดหรือเพื่อให้แน่ใจว่า LCM จะทำงานได้ถูกต้องแน่นอน วงจรภายนอกควรจะทำการรีเซต LCM เอง การรีเซต LCM ด้วยคำสั่งมีกระบวนการดังแสดงในแผงผัง ในรูปที่ 2.24



รูปที่ 2.24 กระบวนการรีเซ็ต LCM

ทฤษีการออกแบบฐานข้อมูลแบบสัมพันธ์โดยวิธีไนแอม (The NIAM Conceptual Schema Model)

แบบจำลองข้อมูลไนแอมถูกคิดค้นขึ้นโดย Prof. G.M. Nijssen โดยไนแอมเป็นแบบจำลองข้อมูลระดับแนวคิด (Conceptual Model) ซึ่งมีพื้นฐานมาจากภาษาธรรมชาติแบบโครงสร้างลึก

การออกแบบฐานข้อมูลด้วยวิธีไนแอม

ไนแอมมีขั้นตอนการอกแบบอยู่ 9 ขั้นตอนดังนี้

ขั้นตอนที่ 1 : กำหนดขอบเขตของงาน (Universe Of Discourse : UOD) และความจริงที่เกิดขึ้นภายในขอบเขตของงานที่กำหนดไว้

ขั้นตอนที่ 2 : วาด Conceptual Schema Diagram โดยคร่าวๆจากความจริงในขอบเขตของงาน

ขั้นตอนที่ 3 : จัดรูปของ Schema ให้เป็นระเบียบ และ หาชนิดความจริงที่ได้รับข้อมูลมาจากชนิดความจริงอื่น

ขั้นตอนที่ 4 : เติมสัญลักษณ์แสดง Uniqueness constraints

ขั้นตอนที่ 5 : ตรวจสอบความถูกต้องของชนิดความจริง

ขั้นตอนที่ 6 : เติมสัญลักษณ์แสดง Lexical ,Mandatory Role ,Subtype constrains

ขั้นตอนที่ 7 : ตรวจสอบ Unique Identifier ของแต่ละชนิดเอนติตี

ขั้นตอนที่ 8 : เติมสัญลักษณ์แสดง Equality ,Exclusion ,Subset constraints

ขั้นตอนที่ 9 : ตรวจสอบความบูรณ์ของ Conceptual Schema ว่าสอดคล้องกับตัวอย่างข้อมูล และไม่มีความซ้ำซ้อนของข้อมูล

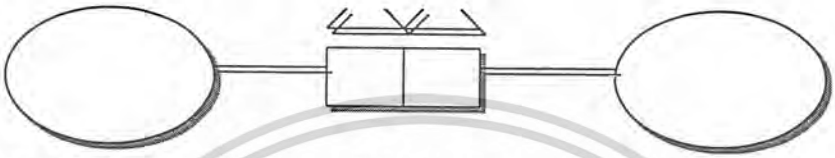
ส่วนประกอบพื้นฐานของไนแอม

ไนแอมมีส่วนประกอบพื้นฐานดังต่อไปนี้

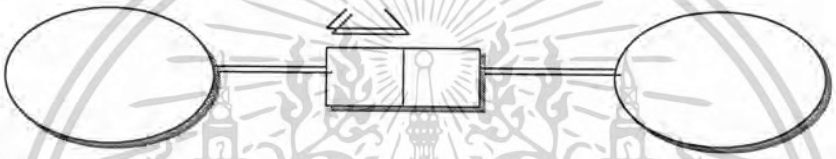
1. ชนิดเอนติตี (Entity Type)
2. ชนิดเลเบิล (Label Type)
3. ชนิดความจริง (Fact Type)
4. ชนิดอ้างอิง (Reference Type)
5. ข้อจำกัดเพื่อความถูกต้อง (Integrity Constraint)

ชนิดเอนติตี(Entity Type)

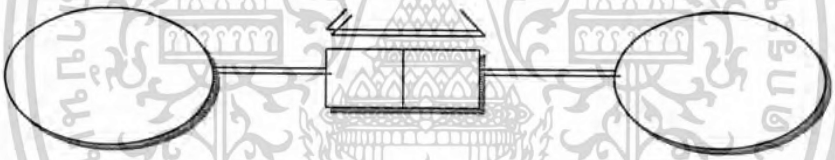
ชนิดเลเบิล (Label Type)



แสดงความสัมพันธ์อ้างอิงแบบ 1 : 1 (one to one)



แสดงความจริงแบบ m : 1 (many to one)



แสดงความจริงแบบ m : 1 (many to one)

รูปที่ 2.25 สัญลักษณ์ของส่วนประกอบพื้นฐานของแบบจำลองไนแอม

ชนิดเอนติตี เป็นเซตซึ่งมีสมาชิกเป็นตัวอย่างเอนติตี (Entity Instance) และ เครื่องหมายความสัมพันธ์ที่เป็นส่วนเชื่อมโยงระหว่างชนิดเอนติตี และ ชนิดเอนติตี หรือชนิดเลเบิลนั้น เรียกว่าบทบาท (role)

บทที่ 3

การออกแบบ

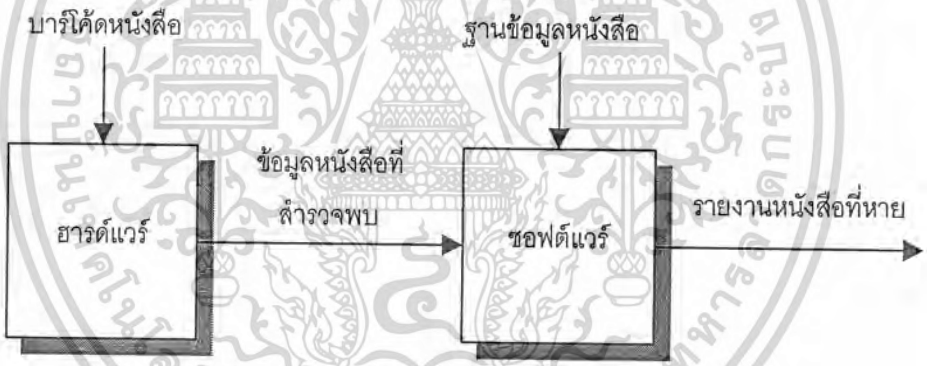
ปัญหาของระบบเดิม

เนื่องจากระบบสำรวจหนังสือประจำปีของหอสมุดกลางของสถาบันเราในปัจจุบันนี้ จะทำโดยการนำคอมพิวเตอร์เข้าไปสำรวจหนังสือตามชั้นหนังสือทีละชั้นจนครบทุกเล่ม ซึ่งเป็นวิธีการที่ยุ่งยากมาก เพราะต้องต่อสายไฟฟ้าเข้าไปตามชั้นหนังสือที่เราจะเข้าไปสำรวจ และไม่สามารถกระทำการสำรวจหนังสือไปพร้อมๆกันทีละหลายๆเครื่องได้ จึงทำให้ต้องเสียเวลาในการสำรวจหนังสือในแต่ละปีเป็นเวลานานมาก จนถึงกับต้องปิดบริการห้องสมุดเป็นเวลาหลายวัน

ความต้องการของระบบ

สามารถทำให้การสำรวจหนังสือเป็นไปด้วยความสะดวกสบายและรวดเร็ว

ภาพรวมของระบบ



รูปที่ 3.1 ภาพรวมของระบบ

โครงงานนี้จะประกอบด้วยส่วนสองส่วนคือ

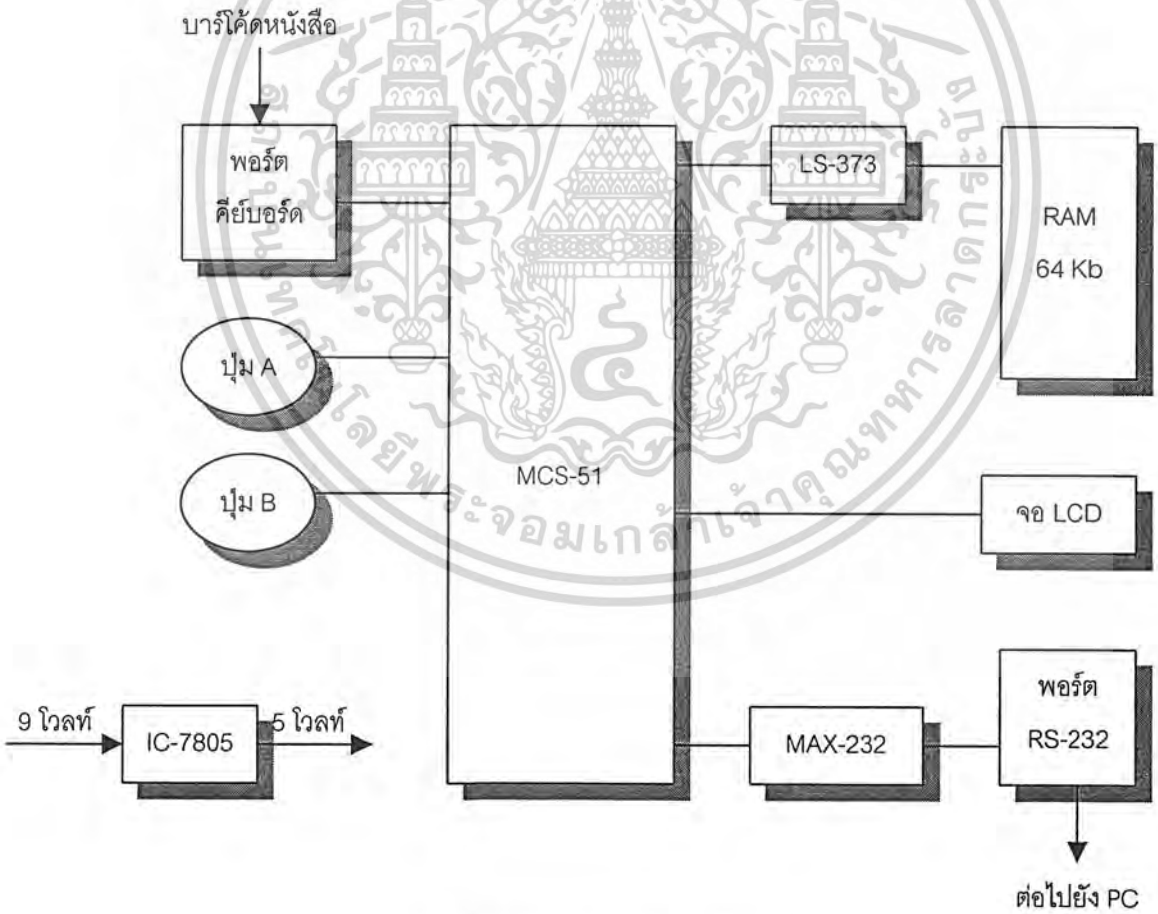
1. ส่วนของฮาร์ดแวร์ (เครื่องสำรวจหนังสือแบบพกพา)
2. ส่วนของซอฟต์แวร์ (โปรแกรมจัดการฐานข้อมูลหนังสือ)

ส่วนของฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์จะมีฟังก์ชันในการทำงานดังนี้

- รับข้อมูลบาร์โค้ดจากเครื่องอ่านบาร์โค้ดโดยผ่านทางพอร์ตคีย์บอร์ด
- นำข้อมูลบาร์โค้ดที่ได้มาทำการแปลงเป็นแอสกีโค้ด แล้วนำไปเก็บไว้ในหน่วยความจำ
- มีจอแอลซีดีแสดงสถานะของหน่วยความจำ และแสดงข้อมูลของหนังสือเล่มล่าสุดที่ทำการสำรวจ
- โอนถ่ายข้อมูลแอสกีโค้ดของหนังสือที่สำรวจมาได้ทั้งหมดจากหน่วยความจำ ไปยังเครื่องคอมพิวเตอร์ที่มีโปรแกรมจัดการฐานข้อมูลหนังสืออยู่
- สามารถเก็บข้อมูลของหนังสือที่ทำการสำรวจไปแล้วได้ แม้ว่าจะหยุดจ่ายไฟเลี้ยงแล้วก็ตาม
- สามารถเคลียร์ข้อมูลหนังสือที่ทำการสำรวจมาทั้งหมดในหน่วยความจำได้
- ใช้แหล่งจ่ายไฟจากแบตเตอรี่ 12 โวลท์

ส่วนประกอบของฮาร์ดแวร์



รูปที่ 3.2 ส่วนประกอบของฮาร์ดแวร์

การใช้งานของปุ่ม A และ ปุ่ม B

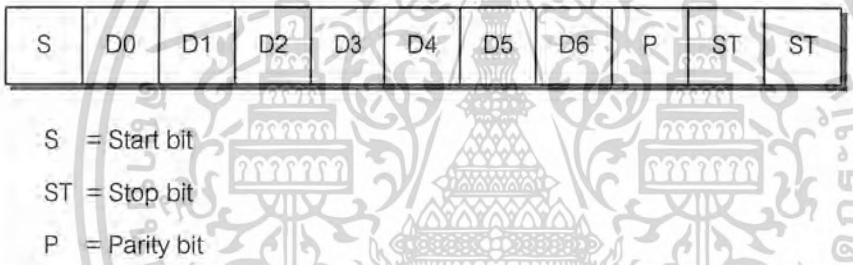
- เมื่อกดปุ่ม B จะเป็นการเริ่มโอนถ่ายข้อมูล
- เมื่อกดปุ่ม A ค้างไว้แล้วกดปุ่ม B จะเป็นการเคลียร์ข้อมูลในหน่วยความจำ

ส่วนรายละเอียดของการทำงานนั้นจะอธิบายในภายหลัง

การรับข้อมูลจากเครื่องอ่านบาร์โค้ด

ในการอ่านบาร์โค้ด 1 ครั้งนั้น เครื่องยิงบาร์โค้ดจะส่งค่าแสกนโค้ดของลำดับตัวอักษรที่อ่านได้ออกมา ซึ่งเป็นมาตรฐานเดียวกันกับคีย์บอร์ด โดยจะมีการเพิ่มแสกนโค้ดของคีย์ Enter ขึ้นมาต่อท้าย

เมื่อกดคีย์บอร์ดหนึ่งครั้งนั้น จะมีการส่งข้อมูลค่าของแสกนโค้ดของคีย์ที่กดออกมา ซึ่งมีรูปแบบเป็นข้อมูล 11 บิตที่มาพร้อมกับสัญญาณนาฬิกาจำนวน 11 ลูก โดยบิตแรกจะเป็น สตาร์ทบิต บิตที่ 2-8 นั้นจะเป็นค่าของแสกนโค้ดที่เราต้องการ บิตต่อมาจะเป็นพาริตี้บิตของค่าแสกนโค้ด ส่วนสองบิตสุดท้ายนั้นจะเป็นสต้อปบิต และเมื่อมีการปล่อยคีย์ที่กดนั้นจะมีการส่งแสกนโค้ดมาอีกสองค่า โดยค่าแรกคือ 0E ส่วนค่าที่สองคือค่าแสกนโค้ดของคีย์ที่ปล่อยนั่นเอง



รูปที่ 3.3 ลักษณะการส่งข้อมูลของพอร์ทคีย์บอร์ด

ในการรับค่าแสกนโค้ดจากพอร์ตคีย์บอร์ดนั้น ใช้วิธีการเรียกโปรแกรมอินเทอร์รัปต์ตามสัญญาณนาฬิกาที่ส่งมาจากคีย์บอร์ด โดยจะทำการต่อขาสัญญาณนาฬิกาของคีย์บอร์ดเข้ากับขา INT0\ ของ MCS-51 คือเมื่อมีการรับค่าแสกนโค้ดแต่ละค่านั้น จะมีการเรียกอินเทอร์รัปต์ทั้งหมด 11 ครั้ง โดยจะมีการรับค่าในการอินเทอร์รัปต์ครั้งที่ 2 ถึงครั้งที่ 8 นั่นเอง

การแปลงค่าแสกนโค้ดเป็นแอสกีโค้ด

ใช้วิธีการอ้างตาราง โดยจะมีตัวแปรตัวหนึ่งเก็บค่าตำแหน่งเริ่มต้นของตารางไว้ (ซึ่งมีค่าแสกนโค้ดเป็น 0) แล้วเมื่อเราต้องการจะแปลงค่าแสกนโค้ดที่รับมาได้ให้เป็นค่าแอสกี้โค้ดนั้น ทำได้โดยนำค่าตัวแปรตำแหน่งเริ่มต้นตารางบวกกับค่าแสกนโค้ดที่รับมา ก็จะได้ตำแหน่งที่เก็บค่าของรหัสแอสกีที่ตรงกับค่าของแสกนโค้ดนั้น

การนำค่าแอสกีโค้ดที่ได้ไปเก็บไว้ในหน่วยความจำ

หน่วยความจำในที่นี้เราจะใช้หน่วยความจำภายนอก เพราะหน่วยความจำภายนอกนั้นมีขนาดใหญ่กว่าหน่วยความจำภายในมาก ทำให้สามารถเก็บข้อมูลได้มากกว่า โดยเราจะใช้หน่วยความจำขนาด 32 กิโลไบต์ ทำให้สามารถเก็บข้อมูลบาร์โค้ดของหนังสือได้ประมาณสี่พันเล่ม (หนังสือ 1 เล่ม จะใช้หน่วยความจำ 8 ไบต์)

การแสดงผลบนจอ LCD

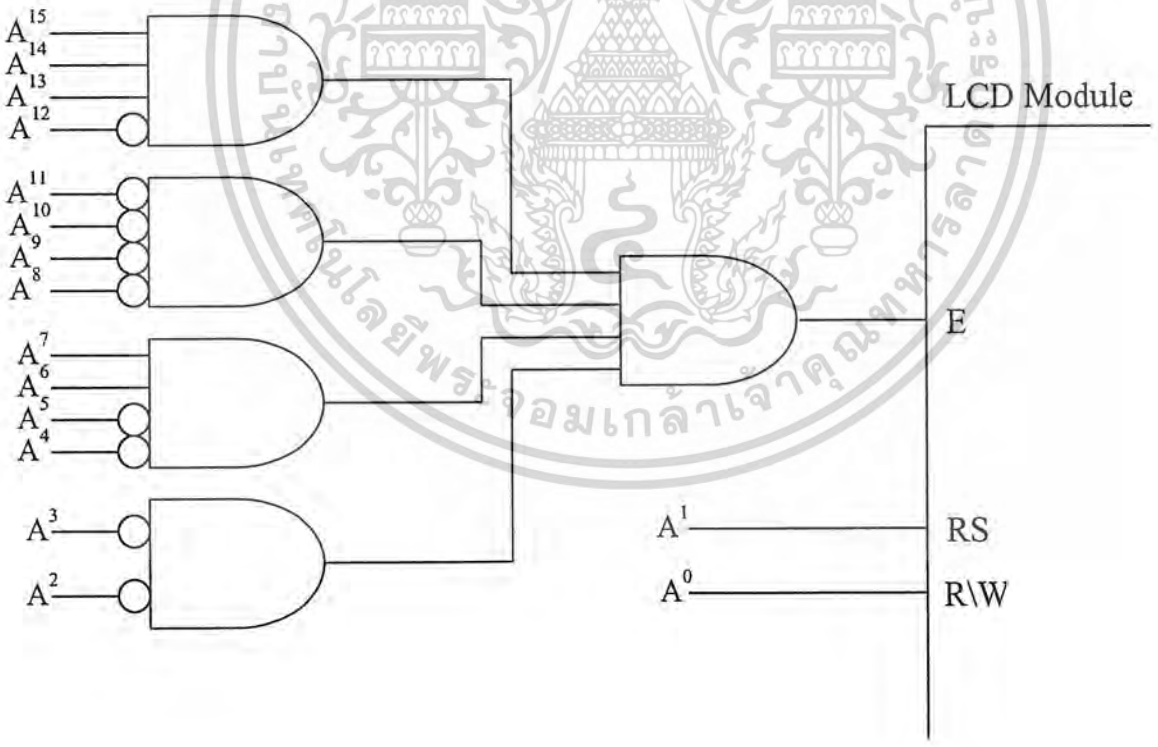
การเชื่อมต่อกับจอ LCD นั้น จะใช้วิธี เมโมรี่แมป คือจะทำการดีโค้ดแอดเดรสของตำแหน่งเมโมรี่ที่ต้องการมาใช้ในการติดต่อกับ LCD ตำแหน่งที่ใช้ในการติดต่อมีดังนี้

ตำแหน่ง E0C0h ใช้ในการเขียนคำสั่งไปยัง LCD

ตำแหน่ง E0C1h ใช้ในการอ่านค่า busy แฟล็ก จาก LCD

ตำแหน่ง E0C2h ใช้ในการเขียนข้อมูลไปยัง LCD

ตำแหน่ง E0C3h ใช้ในการอ่านข้อมูลมาจาก LCD



รูปที่ 3.4 การดีโค้ดพอร์ทแอดเดรส เพื่อใช้กับ LCD module

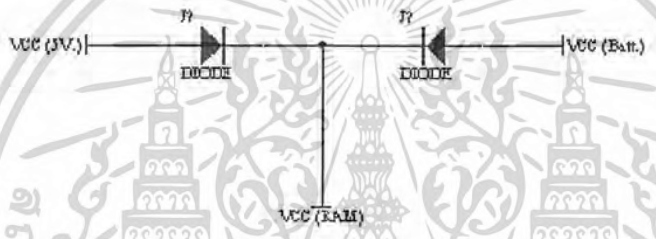
ส่วนในการแสดงผลนั้น จะแสดงค่าบาร์โค้ดของหนังสือเล่มล่าสุดที่ทำการสำรวจ กับค่าจำนวนเล่มของหนังสือที่ทำการสำรวจมาทั้งหมด

การโอนถ่ายข้อมูลไปยังเครื่องคอมพิวเตอร์

การโอนถ่ายข้อมูลไปยังเครื่องคอมพิวเตอร์นั้น จะใช้มาตรฐาน RS-232 โดยจะมีสวิทช์ปุ่ม B เป็นปุ่มที่ใช้ในการเรียกอินเทอร์รัปต์เพื่อเริ่มการโอนถ่ายข้อมูล โดยสวิทช์นี้จะต่ออยู่กับขา INT1\ ของ MCS-51

ความสามารถในการเก็บค่าในเมโมรี่เมื่อหยุดจ่ายไฟเลี้ยง

ใช้วิธีการแบ็คอัพเมโมรี่ โดยการใช้แบตเตอรี่ ซึ่งมีวงจรดังนี้



รูปที่ 3.5 วงจรแบตเตอรี่แบ็คอัพหน่วยความจำ

การเคลียร์ข้อมูลทั้งหมดในหน่วยความจำ

ใช้ สวิทช์ปุ่ม A และ สวิทช์ปุ่ม B ในการควบคุม โดยจะต้องกดสวิทช์ปุ่ม A ค้างไว้ แล้วกดสวิทช์ปุ่ม B

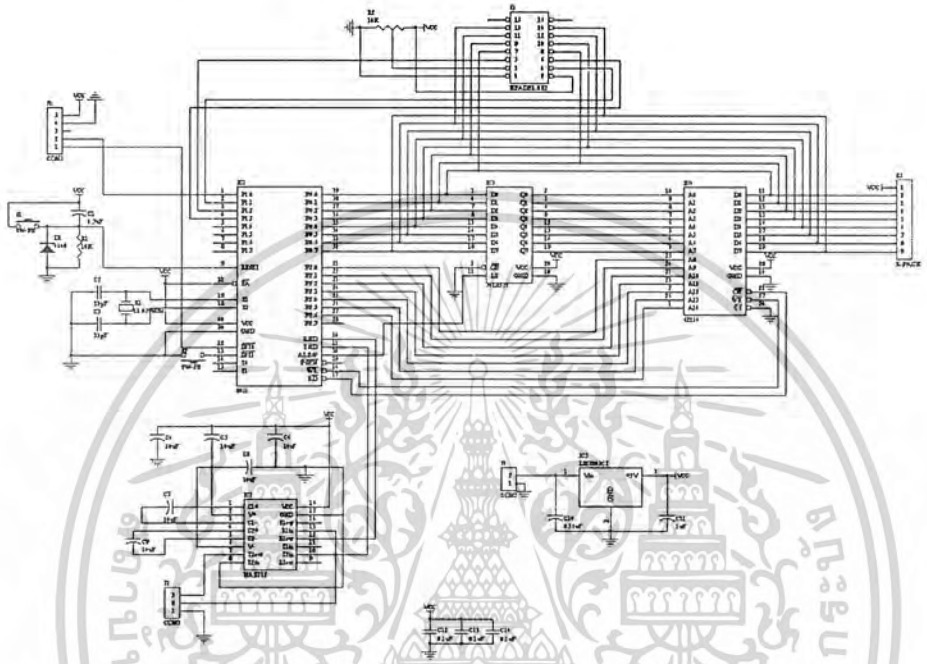
สาเหตุที่ต้องใช้ 2 ปุ่มในการควบคุมเนื่องจาก MCS-51 นั้นมีการตอบสนองต่อสัญญาณอินเทอร์รัปต์จากภายนอกได้เพียงสองสัญญาณ โดยสัญญาณ INT0\ นั้น ใช้กับการรับข้อมูลของพอร์ทคีย์บอร์ดไปแล้ว จึงเหลืออีกเพียงหนึ่งสัญญาณ คือ INT1\ แต่ต้องทำการตอบสนองต่อสัญญาณภายนอกอีก 2 สัญญาณ คือสัญญาณเริ่มโอนถ่ายข้อมูล และสัญญาณการเคลียร์ข้อมูลในหน่วยความจำ ทำให้ต้องแบ่งการทำงานของสัญญาณ INT1\ ให้สามารถตอบสนองต่อสัญญาณทั้งสองได้ โดยนำ 1 บิตของพอร์ต 3 (P3.7) เข้ามาช่วย โดยเมื่อปล่อยปุ่ม A อยู่นั้น พอร์ต P3.7 จะมีค่า เป็น 1 และเมื่อกดปุ่ม A พอร์ต P3.7 จะถูกต่อลงกราวน์ทำให้มีค่าเป็น 0

จึงสามารถนำปุ่ม A และ ปุ่ม B มาใช้กำหนดการตอบสนองต่อสัญญาณภายนอกได้โดย เมื่อมีการเรียกอินเทอร์รัปต์ INT1\ นั้น จะมีการตรวจสอบค่าของพอร์ต P3.7 โดยถ้าเป็น 1 นั้น จะไปทำการเรียกรูทีนของการโอนถ่ายข้อมูล แต่ถ้าเป็น 0 นั้น จะทำการเรียกรูทีนการเคลียร์ค่าข้อมูลในหน่วยความจำ

ความสามารถในการใช้แหล่งจ่ายไฟจากแบตเตอรี่ 12 โวลท์

เนื่องจากวงจรของ MCS-51 นั้นต้องการแหล่งจ่ายแรงดัน 5 โวลท์ จึงต้องใช้ IC เบอร์ 7805 ซึ่งเป็น IC regulator ในการแปลงไฟจาก 9-12 โวลท์ ไปเป็น 5 โวลท์

วงจรรวมของฮาร์ดแวร์



รูปที่ 3.6 วงจรรวมของฮาร์ดแวร์

ส่วนของแอปพลิเคชั่นโปรแกรม

อุปกรณ์ที่ใช้ในการพัฒนาซอฟทแวร์ของระบบ

- เครื่องคอมพิวเตอร์
- โปรแกรมจัดการการกำหนดคุณสมบัติต่างๆในการติดต่อฐานข้อมูล(Boland Database Engine)
- โปรแกรมพัฒนาแอปพลิเคชั่น (Rapid Application Development : RAD) ที่ใช้ภาษา ออปเจคปาสคาล (Object Pascal) บนระบบวินโดวส์ (Windows) ซึ่งในที่นี้ใช้โปรแกรม เดลไฟล์ 4 (Delphi 4)
- คอมโพเนนต์ในการติดต่อสื่อสารผ่านพอร์ตอนุกรมซึ่งในที่นี้ใช้ MSCOMM32.OCX ซึ่งเป็นคอมโพเนนต์ของวิชชวลเบสิค (Visual Basic)

ความต้องการของระบบ

จากการศึกษาความต้องการของระบบจากผู้ใช้คือเจ้าหน้าที่ของหอสมุดกลาง ได้ความต้องการดังต่อไปนี้

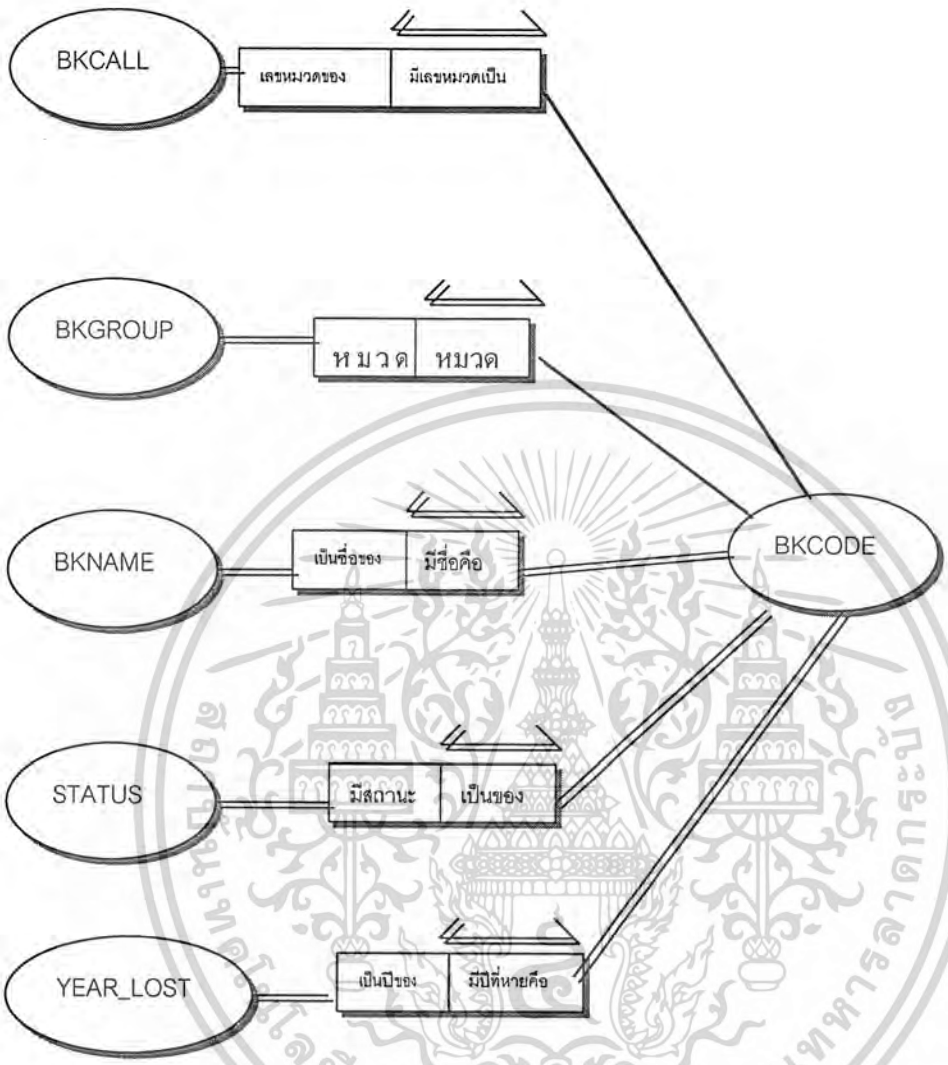
- ทำการสำรวจหนังสือทั้งหมดในหอสมุดกลางของสถาบันโดยการใช้บาร์โค้ดในการสำรวจแล้วจึงนำมาตรวจสอบกับฐานข้อมูล
- สามารถเลือกสถานะของหนังสือบางชนิดที่ไม่ต้องการนำออกมาแสดงโดยเลือกที่สถานะ "ไม่แสดง"
- ต้องการรายงานการสำรวจหนังสือ ว่ามีหนังสือเล่มใดบ้างที่หายโดยต้องการรายงาน 4 ชนิดคือ
  - รายงานหนังสือที่หายเรียงตามหมวดหนังสือ
  - รายงานหนังสือที่หายเรียงตามปีที่หาย
  - รายงานสรุปหนังสือที่หายเรียงตามหมวดหนังสือ
  - รายงานสรุปหนังสือที่หายเรียงตามปีที่หาย และหมวด
- ในระบบเก่านั้น การป้อนข้อมูลในเลขหมวดนั้นมีการใส่ช่องว่าง (Space) ซึ่งเมื่อทำการเรียงลำดับข้อมูลในการทำรายงานนั้นจะเกิดปัญหา มีการผิดพลาดเกิดขึ้น ในระบบใหม่จึงต้องมีการแก้ไขข้อมูลของเลขหมวดของหนังสือแต่ละเล่มไม่ให้มีช่องว่างเหลืออยู่โดยจะทำการแก้ไขในขั้นตอนการอัพเดตข้อมูล

การออกแบบฐานข้อมูล

จากการศึกษาและรวบรวมความต้องการของระบบที่ได้ทำการพัฒนาจะได้ข้อมูลดังนี้

- ฐานข้อมูลที่ใช้งานต้องเก็บรายละเอียดของหนังสือคือ รหัสหนังสือ เลขหมวดของหนังสือ ชื่อหนังสือ และหมวดของหนังสือ ซึ่งได้มาจากส่วนหนึ่งของฐานข้อมูลของระบบเก่า
- ฐานข้อมูลที่ใช้งานต้องมีการเพิ่มข้อมูลอื่นๆคือ สถานะของหนังสือมี 3 สถานะคือ หาย ไม่หาย และไม่แสดงซึ่งได้จากการศึกษาความต้องการของผู้ใช้
- ฐานข้อมูลที่ใช้งานต้องมีการเพิ่มข้อมูลปีที่สำรวจหนังสือเล่มนั้นๆไม่พบ

จากความต้องการที่ได้จะทำการออกแบบฐานข้อมูลโดยวิธีไนแอมได้ดังนี้



รูปที่ 3.7 ออกแบบฐานข้อมูลโดยวิธีไนแอม

ตารางของฐานข้อมูลที่มีการใช้งานที่ใช้งาน

- ฐานข้อมูลของข้อมูลหนังสือในหอสมุดกลางที่ใช้กับระบบเก่า เป็นฐานข้อมูลที่ได้มีการป้อนข้อมูลของหนังสือทั้งหมดในหอสมุดกลาง จะมีการนำไปใช้ในขั้นตอนการอัพเดตฐานข้อมูลเพื่อใช้ในการสำรวจหนังสือ
- ฐานข้อมูลที่ได้สร้างไว้เพื่อใช้ในงานที่ได้ออกแบบไว้

ตาราง LIBBK2

BKCODE	BKCALL	BKGROUP	BKNAME	STATUS	YEAR_LOST

รูปที่ 3.8 ฐานข้อมูลของระบบที่ได้ทำการออกแบบ ใช้กับการสำรวจหนังสือ

BKCODE คือ รหัสหนังสือ

BKCALL คือ เลขหมวดของหนังสือ

BKGROUP คือ หมวดของหนังสือ

BKNAME คือ ชื่อหนังสือ

STATUS คือ สถานะของหนังสือ

YEAR_LOST คือ ปีที่สำรวจหนังสือเล่มนั้นไม่พบ

ตารางที่ 3.1 ตารางแสดงลักษณะของฟิลด์ข้อมูล

Field	Type	null	Key
BKCODE	CHARACTOR	NO	Primary Key
BKCALL	CHARACTOR	YES	
BKGROUP	CHARACTOR	YES	
BKNAME	CHARACTOR	YES	
STATUS	CHARACTOR	YES	
YEAR_LOST	NUMBER	YES	

ฟังก์ชันการทำงานของซอฟทแวร์

จากการศึกษาความต้องการของผู้ใช้ระบบ และปัญหาของระบบเดิม ในระบบนี้จึงได้มีการจัดเตรียมฟังก์ชันการทำงานต่างๆดังต่อไปนี้

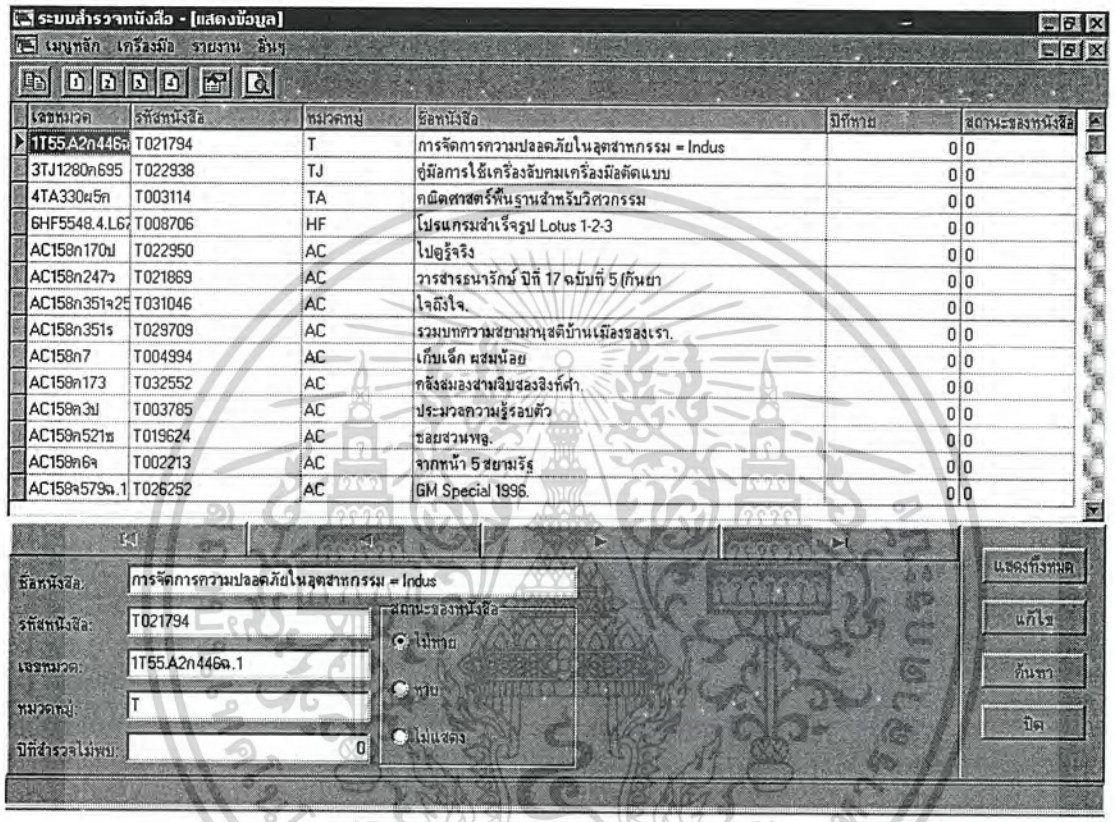
1) ส่วนหน้าจอหลักของโปรแกรม

ในส่วนนี้จะเป็นหน้าจอหลักของโปรแกรมที่ได้พัฒนาขึ้นมาโดยสร้างขึ้นมาในลักษณะของ MDI แอปพลิเคชั่น (Multiple Document Interface) โดยส่วนนี้จะเป็นส่วนที่ทำการเชื่อมต่อกับฟังก์ชันการทำงานอื่นๆ ทั้งหมดของโปรแกรม

รูปที่ 3.9 ส่วนติดต่อผู้ใช้ หน้าจอหลัก

2) ส่วนการแสดงข้อมูลในฐานข้อมูล

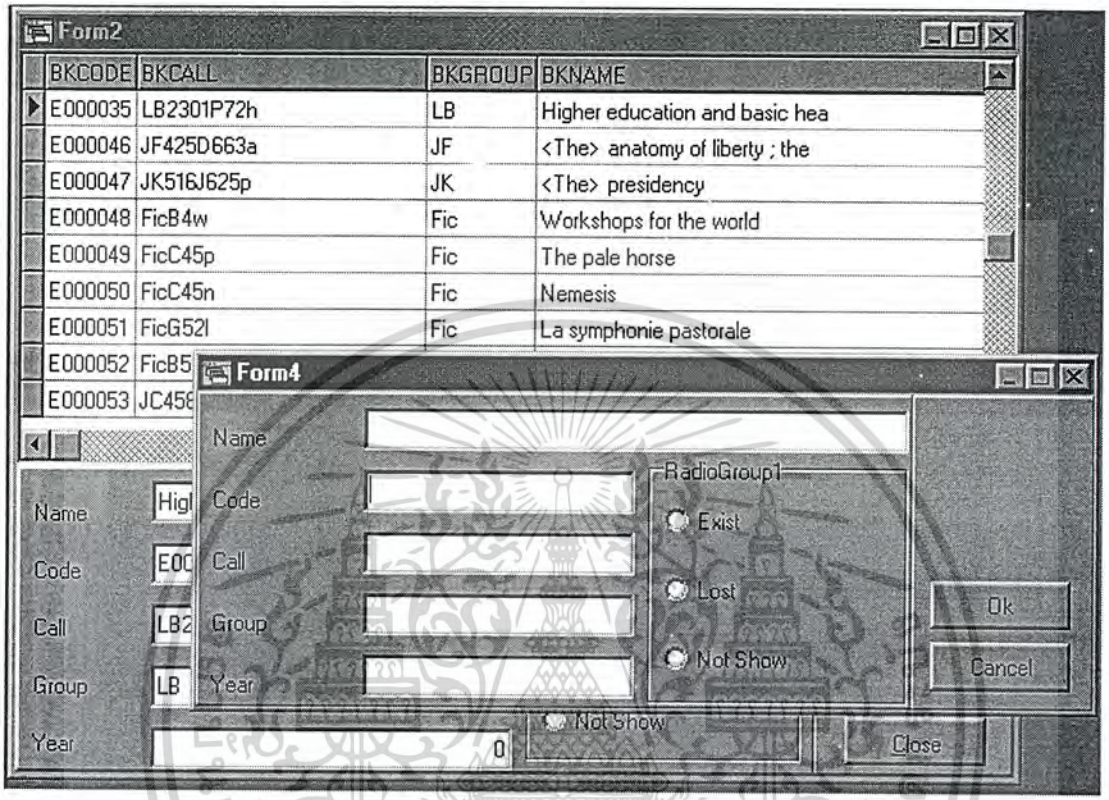
ในส่วนนี้จะเป็นส่วนของการแสดงข้อมูลของหนังสือทั้งหมดในฐานข้อมูล โดยจะแสดงในรูปของตารางแสดงข้อมูล และสามารถแสดงข้อมูลของหนังสือที่ถูกเลือกขึ้นมาได้เพื่อให้ผู้ใช้สามารถดูรายละเอียดของหนังสือได้อย่างชัดเจน



รูปที่ 3.10 ส่วนติดต่อผู้ใช้ ส่วนแสดงข้อมูลในฐานข้อมูล

ในส่วนนี้ยังประกอบไปด้วยฟังก์ชันย่อยๆอีกสองฟังก์ชันคือ

1 ฟังก์ชันการค้นหาข้อมูล โดยเมื่อต้องการค้นหาข้อมูลของหนังสือที่ต้องการดูข้อมูล ก็สามารถทำได้โดยการเลือกที่ปุ่ม ค้นหา จะมีหน้าจอให้ป้อนข้อมูลที่จะค้นหาปรากฏขึ้นมา โดยผู้ใช้ต้องป้อนข้อมูลที่ต้องการค้นหาก็จะแสดงข้อมูลที่ต้องการออกมา



รูปที่ 3.11 ส่วนติดต่อผู้ใช้ การค้นหาข้อมูลในฐานข้อมูล

ในส่วนของการค้นหาข้อมูลนี้ มีการใช้คำสั่งของ SQL คือคำสั่ง SELECT ในการเลือกข้อมูลที่ผู้ใช้ต้องการค้นหา

2 ฟังก์ชันการแก้ไขข้อมูล จากความต้องการของผู้ใช้นั้น ต้องการเปลี่ยนแแปลงสถานะของหนังสือที่ไม่ต้องการให้แสดงได้ โดยการเลือกสถานะของหนังสือที่ต้องการเป็น " ไม่แสดง " โดยในส่วนนี้จะใช้คำสั่ง SQL คือ UPDATE ในการแก้ไขสถานะของหนังสือที่ต้องการ

3) ส่วนของการแสดงรายงาน

จากความต้องการของผู้ใช้นั้น ต้องการให้มีการแสดงรายงานจากการสำรวจหนังสือว่ามีหนังสือเล่มใดบ้างที่สำรวจไม่พบ หรือหายไป โดยรายงานที่ผู้ใช้ต้องการนั้นมี 4 ชนิดคือ

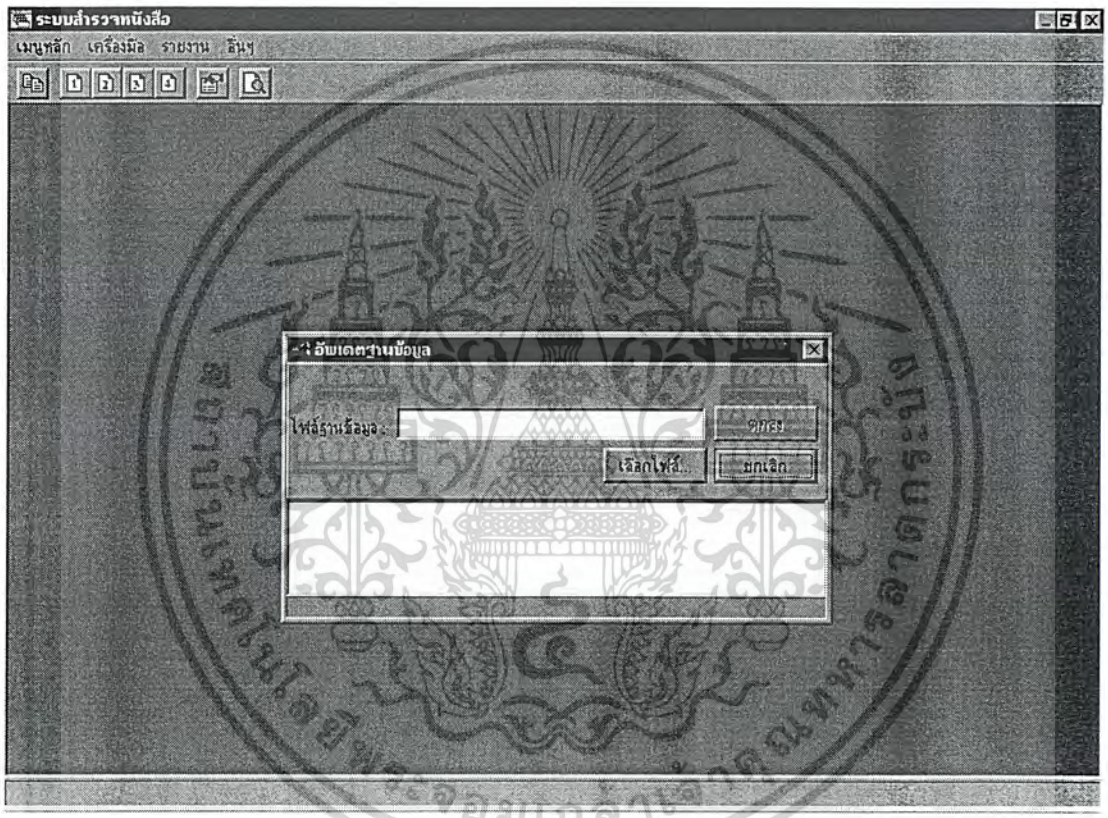
- รายงานหนังสือที่หายเรียงตามหมวดหนังสือ
- รายงานหนังสือที่หายเรียงตามปีที่หาย
- รายงานสรุปหนังสือที่หายเรียงตามหมวดหนังสือ
- รายงานสรุปหนังสือที่หายเรียงตามปีที่หาย และหมวด

รูปที่ 3.12 ส่วนติดต่อผู้ใช้ การแสดงรายงานหนังสือที่สำรวจไม่พบ

4) ส่วนของการอัพเดตข้อมูลในฐานข้อมูล

ในส่วนนี้จะเป็นการอัพเดตข้อมูลในฐานข้อมูลเนื่องจากในแต่ละปีที่มีการซื้อหนังสือใหม่เข้ามาที่หอสมุดกลางจะต้องมีการเพิ่มข้อมูลในฐานข้อมูลของหอสมุดกลาง เมื่อจะทำการสำรวจหนังสือในแต่ละปีนั้นจึงต้องมีการอัพเดตข้อมูลให้กับฐานข้อมูลที่ใช้งาน โดยในส่วนนี้จะใช้การค้นหาข้อมูลในฐานข้อมูลจากฟิลด์ที่เป็น คีย์หลัก (Primary Key) ว่าซ้ำกับที่มีอยู่แล้วหรือไม่ถ้าหาไม่เจอจะทำการเพิ่มข้อมูลลงในฐานข้อมูลที่ใช้งานทันที

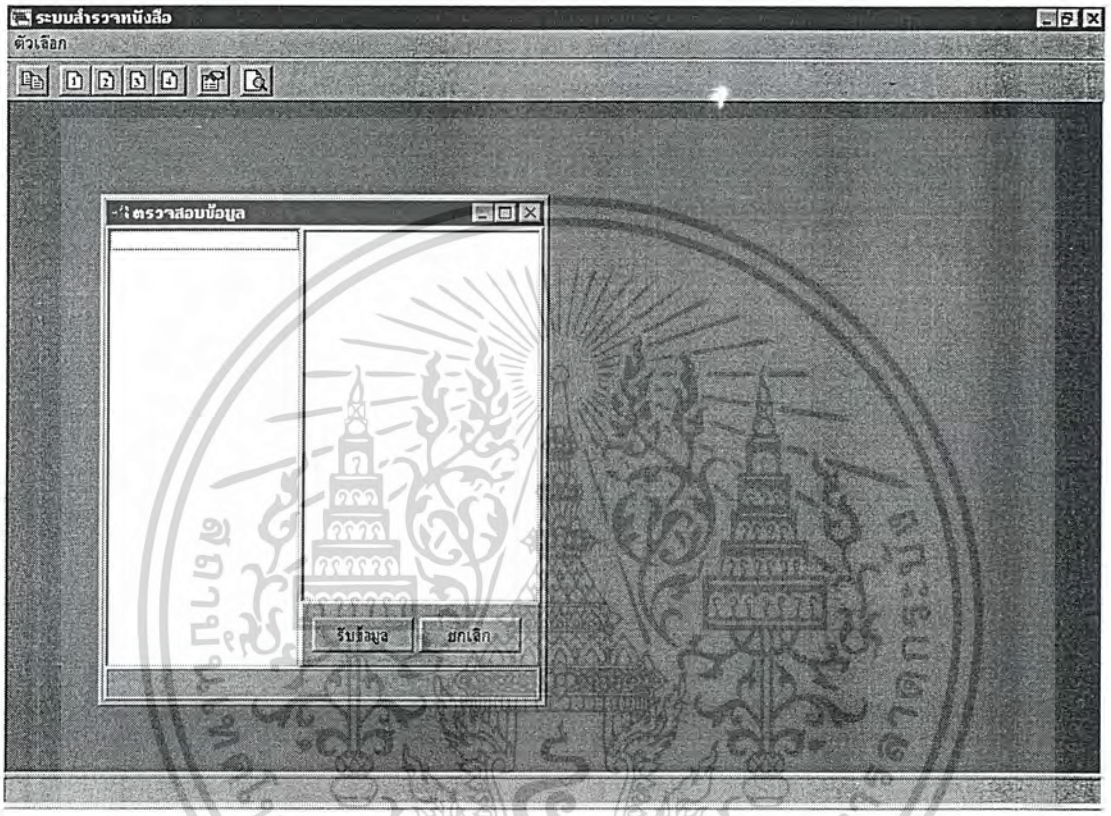
การใช้งานในส่วนนี้ขั้นแรกต้องมีการเลือกฐานข้อมูลที่มีการป้อนข้อมูลเพิ่มเติม เพื่อมาอัพเดตข้อมูลในฐานข้อมูลที่ใช้งาน



รูปที่ 3.13 ส่วนติดต่อผู้ใช้ การอัพเดตฐานข้อมูล

5) ส่วนการรับข้อมูลจากเครื่องสำรวจหนังสือและตรวจสอบข้อมูล

ในส่วนนี้จะเป็นการรับข้อมูลจากเครื่องสำรวจหนังสือที่ทำการสำรวจมาจากชั้นหนังสือเรียบร้อยแล้ว ก็จะทำการส่งข้อมูลโดยผ่านพอร์ตอนุกรมจะได้ข้อมูลเป็นรหัสแอสกี (ASCII) ของรหัสหนังสือแล้วจึงนำมาตรวจสอบกับรหัสหนังสือในฐานข้อมูล ถ้าตรงกับในฐานข้อมูลก็จะเปลี่ยนสถานะของหนังสือเป็น “ยังอยู่” และเมื่อสำรวจเสร็จสิ้นแล้วแล้ว หนังสือที่สำรวจไม่พบก็จะมีสถานะเป็น “หาย”



รูปที่ 3.14 ส่วนติดต่อผู้ใช้ การรับข้อมูลจากเครื่องสำรวจหนังสือ

บทที่ 4
การใช้งาน

การใช้งานในส่วนของฮาร์ดแวร์

ในส่วนของฮาร์ดแวร์นั้นจะมีการใช้งานดังนี้

การเริ่มต้นการใช้งาน

- ใส่แบตเตอรี่ขนาด เอเอ จำนวน 8 ก้อน
- ต่อเครื่องอ่านบาร์โค้ดเข้ากับพอร์ทคีย์บอร์ดของเครื่องสำรวจ
- เปิดสวิทช์ เริ่มการทำงาน
- กดเคลียร์ค่าในหน่วยความจำ โดยกดปุ่ม A ค้างไว้แล้วจึงกดปุ่ม B
- ทดลองอ่านค่าบาร์โค้ดดู หน้าจอ LCD จะแสดง ค่าของบาร์โค้ดของหนังสือที่อ่านได้ และจะแสดงจำนวนเล่มของหนังสือที่ได้ทำการอ่านไปแล้ว
- กดเคลียร์ค่าในหน่วยความจำอีกครั้งเพื่อเริ่มต้นการทำงานใหม่

การโอนถ่ายข้อมูลไปยังเครื่องคอมพิวเตอร์

- ต่อสาย serial port จากเครื่องสำรวจไปยังพอร์ทอนุกรมของเครื่องคอมพิวเตอร์
- กดปุ่ม B เพื่อเริ่มทำการโอนถ่ายข้อมูล

การเคลียรค่าในหน่วยความจำ

- กดเคลียร์ค่าในหน่วยความจำ โดยกดปุ่ม A ค้างไว้แล้วจึงกดปุ่ม B หน้าจอ LCD จะขึ้นข้อความว่า " READY " และจะแสดงจำนวนเล่มของหนังสือที่อ่านได้เป็น 0

การเลิกใช้งาน

- ปิดสวิทช์เลิกการทำงาน
- ถอดสายเครื่องอ่านบาร์โค้ดและสาย serial port ออก
- ควรถอดแบตเตอรี่ เอเอ ออกด้วย

การใช้งานในส่วนของซอฟท์แวร์

การรับข้อมูลและการตรวจสอบข้อมูลจากเครื่องสำรวจหนังสือ

หลังจากที่ได้ทำการสำรวจหนังสือด้วยเครื่องสำรวจหนังสือเรียบร้อยแล้ว ขั้นตอนต่อไปในการทำงานของระบบคือการนำข้อมูลที่ได้สำรวจมาแล้วมาโอนถ่ายลงเครื่องคอมพิวเตอร์เพื่อตรวจสอบหนังสือที่หายโดยมีขั้นตอนดังนี้

1. นำเครื่องสำรวจหนังสือมาเชื่อมต่อกับเครื่องคอมพิวเตอร์โดยใช้ พอร์ตอนุกรมโดยการต่อสายสัญญาณส่งข้อมูลกับเครื่องสำรวจหนังสือเข้ากับพอร์ตอนุกรม
2. เริ่มการทำการของโปรแกรมสำรวจหนังสือ โดยการเลือกที่ไอคอนของโปรแกรมดังกล่าว
3. ในหน้าจอหลักของโปรแกรม เลือกที่ "เครื่องมือ" และ "โอนถ่ายข้อมูล" โปรแกรมจะแสดงหน้าจอการโอนถ่ายข้อมูลขึ้นมา
4. เมื่อตรวจสอบและแน่ใจว่าไม่มีปัญหาในการเชื่อมต่อกับเครื่องสำรวจหนังสือแล้วจึงเริ่มทำการส่งได้โดยการเลือกที่ปุ่ม "รับข้อมูล" โปรแกรมจะทำการเปิดพอร์ตเพื่อรอรับการส่งข้อมูลจากเครื่องสำรวจหนังสือ
5. เมื่อโปรแกรมรอรับข้อมูลแล้วก็เริ่มส่งข้อมูลได้โดยการกดปุ่ม "ส่งข้อมูล" ที่เครื่องสำรวจหนังสือ โปรแกรมจะเริ่มรับข้อมูลโดยจะแสดงข้อมูลที่รับได้ในหน้าจอการรับข้อมูลด้วย
6. เมื่อรับข้อมูลเรียบร้อยแล้ว โปรแกรมจะแสดงผลการสำรวจหนังสือที่ได้ออกมาว่ารับข้อมูลได้กี่เล่ม และมีหนังสือที่สำรวจไม่พบกี่เล่ม

การค้นหาข้อมูลและการแก้ไขข้อมูล

ในการค้นหาข้อมูลและการแก้ไขข้อมูลของหนังสือนั้นจะทำได้ดังนี้

1. ในหน้าจอหลักของโปรแกรมเลือกที่ "เครื่องมือ" และ "แสดงข้อมูล" โปรแกรมจะแสดงหน้าจอการแสดงข้อมูลในฐานข้อมูล
2. ถ้าต้องการดูรายละเอียดของข้อมูลหนังสือทั้งหมด เลือก "แสดงข้อมูล" จะแสดงข้อมูลของหนังสือทั้งหมดในฐานข้อมูล
3. ในการค้นหาข้อมูลนั้นเลือก "ค้นหา" โปรแกรมจะแสดงหน้าจอการค้นหาข้อมูลโดยให้ป้อนข้อมูลที่ต้องการค้นหา แล้วเลือก "ตกลง" โปรแกรมจะแสดงข้อมูลของหนังสือที่มีข้อมูลตรงกับข้อมูลที่ต้องการค้นหาออกมา
4. ในการแก้ไขข้อมูลนั้น ผู้ใช้สามารถแก้ไขข้อมูลได้เฉพาะในส่วนของ "สถานะ" ของหนังสือเท่านั้น โดยการเลือกที่หนังสือที่ต้องการแก้ไขข้อมูล แล้วเลือกที่

สถานะที่ต้องการของหนังสือเล่มนั้นๆ จากนั้นจึงเลือกปุ่ม "แก้ไข" ข้อมูลจะถูกแก้ไขเป็นสถานะที่ได้เลือกไว้

การแสดงรายงานการสำรวจหนังสือ

เมื่อทำการสำรวจหนังสือเสร็จเรียบร้อยแล้ว ผู้ใช้อาจต้องการรายงานของหนังสือที่สำรวจไม่พบในการสำรวจ เพื่อนำไปตรวจสอบในขั้นตอนอื่นๆ โดยโปรแกรมจะแสดงรายงานได้ดังนี้

1. ในหน้าจอหลักของโปรแกรม เลือกที่ "รายงาน" จะมีรายงายงานให้เลือก 4 ชนิดคือ
   - รายงานหนังสือที่หายเรียงตามหมวดหนังสือ
   - รายงานหนังสือที่หายเรียงตามปีที่หาย
   - รายงานสรุปหนังสือที่หายเรียงตามหมวดหนังสือ
   - รายงานสรุปหนังสือที่หายเรียงตามปีที่หาย และหมวด

โดยจะแสดงรายงานในรูปแบบของ ควิกรีพอร์ต(Quick Report) ซึ่งเป็นรายงานที่สร้างได้ง่ายในโปรแกรมเดลไฟล์ ซึ่งสะดวกในการใช้งานอย่างมาก โดยมีฟังก์ชันสำหรับการพิมพ์รายงาน การแสดงรายงานในรูปแบบต่างๆ รวมทั้งการ set up เครื่องพิมพ์ที่ต้องการใช้อีกด้วย

การอัพเดตฐานข้อมูลที่ใช้ในงานสำรวจหนังสือ

ในการอัพเดตฐานข้อมูลที่มีการเพิ่มเติมหนังสือที่ซื้อมาใหม่จากส่วนของการเพิ่มหนังสือในฐานข้อมูลของหอสมุดกลาง โดยจะทำได้ดังนี้

1. ในหน้าจอหลักเลือก "เครื่องมือ" และ "อัพเดตฐานข้อมูล" โปรแกรมจะแสดงหน้าจอในการอัพเดตฐานข้อมูล
2. เลือก "Browse..." เพื่อเลือกไฟล์ฐานข้อมูลที่ต้องการแล้วเลือก "Ok"
3. โปรแกรมจะแสดงชื่อไฟล์และไดเรคทอรี่ที่ไฟล์นั้นๆถูกเก็บอยู่ เมื่อตรวจสอบว่าใช่ไฟล์ที่ต้องการแล้ว เลือก "ตกลง"
4. โปรแกรมจะเริ่มทำการตรวจสอบฐานข้อมูลที่นำมาอัพเดต โดยตรวจสอบจาก รหัสหนังสือ ที่เป็นคีย์หลักของฐานข้อมูล ถ้าหากไม่พบในฐานข้อมูลที่ใช้งานในโปรแกรมแล้ว โปรแกรมจะทำการเพิ่มข้อมูลของหนังสือเล่มนั้น ลงในฐานข้อมูลของโปรแกรม
5. เมื่ออัพเดตเสร็จแล้ว โปรแกรมจะแสดงจำนวนของหนังสือที่ได้มีการเพิ่มลงไปในฐานข้อมูลพร้อมทั้งรหัสหนังสือของหนังสือนั้นๆด้วย

บทที่ 5
บทสรุปและวิจารณ์

ข้อดีของระบบ

1. สามารถช่วยให้การสำรวจหนังสือของสำนักหอสมุดกลางเป็นไปอย่างสะดวก รวดเร็วมากขึ้น
2. สามารถสืบค้นข้อมูลจากระบบได้อย่างรวดเร็ว
3. สามารถออกรายงานต่าง ๆ ที่มีประโยชน์ต่อผู้ใช้
4. สามารถพัฒนาต่อไปในอนาคตได้ง่าย
5. สามารถช่วยลดเวลาในการทำงานได้มากขึ้น
6. สามารถทำงานได้หลายคนในเวลาเดียวกัน

ข้อจำกัดของระบบที่พัฒนา

1 ข้อจำกัดทางฮาร์ดแวร์

ข้อจำกัดทางด้านฮาร์ดแวร์คือ ถ้าใช้แบตเตอรี่แบบ เอเอ แล้วจะทำให้ใช้งานได้เป็นเวลาจำกัด เพราะเครื่องอ่านบาร์โค้ดที่ต่อกับเครื่องสำรวจนั้นกินพลังงานพอสมควร และเมื่อใช้งานตอนแบตเตอรี่อ่อนนั้น อาจทำให้เกิดข้อผิดพลาดได้ เช่น อ่านข้อมูลผิดพลาด หรืออ่านข้อมูลไม่ได้เป็นต้น

แต่ถ้าใช้แบตเตอรี่ขนาดใหญ่ที่มีความจุมากๆ ก็จะทำให้ไม่สะดวกในการพกพาเพราะแบตเตอรี่จะมีขนาดใหญ่ และมีน้ำหนักมาก

2 ข้อจำกัดทางซอฟท์แวร์

ในการใช้โปรแกรมต้องมีการใช้ BDE (Boland Database Engine) ที่ใช้ในการจัดการฐานข้อมูลต่างๆ ดังนั้นจึงต้องมี BDE เตรียมไว้เสมอ

ปัญหาที่พบในการทำโครงงาน

1 ปัญหาในขั้นหาความต้องการระบบ

ในขั้นตอนการศึกษาและหาความต้องการของระบบนั้นมีปัญหาโดยสรุปเป็นข้อ ๆ คือ

1. ผู้ใช้ยังมีการใช้ฟังก์ชันบางอย่างของระบบเดิมอยู่
2. ผู้ใช้แต่ละคนมีความเข้าใจในงานไม่เหมือนกัน
3. มีความสับสนในความต้องการของผู้ใช้กับผู้วิเคราะห์ระบบ

2 ปัญหาในการพัฒนาระบบ

1. การใช้งานอัพเดตฐานข้อมูล มีการใช้ฐานข้อมูลของระบบเก่าซึ่งจะหา Driver ภาษาไทยไม่ได้

2. ในการใช้เดลไฟล์ติดต่อกับพอร์ตอนุกรมนั้น ต้องใช้คอมโพเนนต์ของ วิชชวลเบสิก คือ MSCOMM32.OCX เนื่องจากเดลไฟล์ไม่มีคอมโพเนนต์สำหรับติดต่อกับพอร์ตอนุกรม

บทวิจารณ์ผล

สำหรับการพัฒนาโครงงานนี้ ซึ่งได้กล่าวในวัตถุประสงค์ว่า เพื่อสร้างระบบที่อำนวยความสะดวกและให้ประโยชน์อย่างมากในการสำรวจหนังสือของสำนักหอสมุดกลาง ซึ่งโครงงานนี้จะประสบผลสำเร็จและให้ประโยชน์สูงสุดได้ จำเป็นต้องขึ้นอยู่กับปัจจัยในหลายๆด้าน ไม่ว่าจะเป็น ประสิทธิภาพของโปรแกรม ความต้องการของผู้ใช้ เทคโนโลยีที่นำมาสนับสนุน เป็นต้น ซึ่งถ้าปัจจัยต่างๆ เหล่านี้สนับสนุนต่อระบบงาน ทางผู้พัฒนาก็คาดว่า ระบบนี้ต้องได้รับการสนับสนุนในอนาคต และอาจนำไปพัฒนาเพื่อใช้ในงานอื่นๆ ที่มีการทำงานในรูปแบบใกล้เคียงกันได้

ข้อเสนอแนะ

การพัฒนาโครงงานนี้ นอกจากจะทำให้ได้ชิ้นโครงงานตามที่ต้องการแล้ว ยังทำให้ผู้พัฒนาได้ความรู้เพิ่มเติมในส่วนที่เกี่ยวข้องตามทฤษฎีที่ได้ศึกษาไปแล้ว และโครงงานนี้ยังสามารถนำไปใช้งานจริงได้ในการสำรวจหนังสือของสำนักหอสมุดกลางอีกด้วย ทำให้ผู้พัฒนาสามารถมองเห็นแนวทาง และ วิธีการในการพัฒนาระบบต่อไปในอนาคต

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคผนวก ก

การเขียนโปรแกรมโดยใช้เดลไฟล์ (Delphi4)

เดลไฟล์ (Delphi) เป็นเครื่องมือสำหรับการพัฒนาแอปพลิเคชั่นบนวินโดวส์ (Windows) ภาษาพื้นฐานที่ใช้คือภาษาปาสคาล (Object Pascal) ในการเขียนโปรแกรมและยังสนับสนุนการพัฒนาโปรแกรมแบบ วิชชวล โปรแกรม (Visual Programming) ที่ผู้พัฒนาระบบสามารถเลือกคอมโพเนนต์ที่ต้องการมาวางบนฟอร์ม แล้วกำหนดคุณสมบัติบางอย่างของของคอมโพเนนต์เหล่านั้น รวมทั้งอาจมีการเขียนโปรแกรมควบคุมการทำงานของคอมโพเนนต์เหล่านั้นด้วย

เดลไฟล์ เป็นเครื่องมือที่สร้างแอปพลิเคชั่นต่างๆได้หลากหลายซึ่งมีความสามารถหลักๆดังนี้

1. พัฒนาแอปพลิเคชั่นทั่งไปที่รันบนระบบปฏิบัติการวินโดวส์ โดยเราสารถสร้างโปรแกรมด้านกราฟฟิก โปรแกรมการคำนวณต่างๆตามความต้องการได้
2. พัฒนาแอปพลิเคชั่นด้านฐานข้อมูล ซึ่งเดลไฟล์มีจุดเด่นอย่างมากในการสร้างแอปพลิเคชั่นด้านนี้ไม่ว่าไม่ว่าจะเป็นความง่ายในการพัฒนา ความเร็วของแอปพลิเคชั่นที่พัฒนาหรือความพร้อมของเครื่องมือที่ใช้ในการพัฒนาแอปพลิเคชั่น เราสามารถติดต่อกับไฟล์ฐานข้อมูลทั่วไป เช่น ดีเบส(Dbase) ,พาราดอกส์(Paradox) และ แอกเซส(Access) หรือติดต่อกับฐานข้อมูลบนเครื่องเซอเวอร์(Server) หรือฐานข้อมูลประเภทอื่นๆได้ นอกจากนั้นยังมีเครื่องมือในการสร้างรายงานคือ ควิกรีพอร์ต (Quick Report) ซึ่งช่วยในการสร้างรายงานได้อย่างง่ายดาย
3. แอกทีพเอ็กซ์ (ActiveX) แบบต่างๆได้แก่แอกทีพเอ็กซ์คอนโทรล ซึ่งคือเครื่องมือที่ช่วยให้สามารถนำส่วนของโปรแกรมที่เราสร้างไว้ใน เดลไฟล์ ไปใช้งานในโปรแกรมอื่นๆได้

ความต้องการของระบบสำหรับ Delphi 4

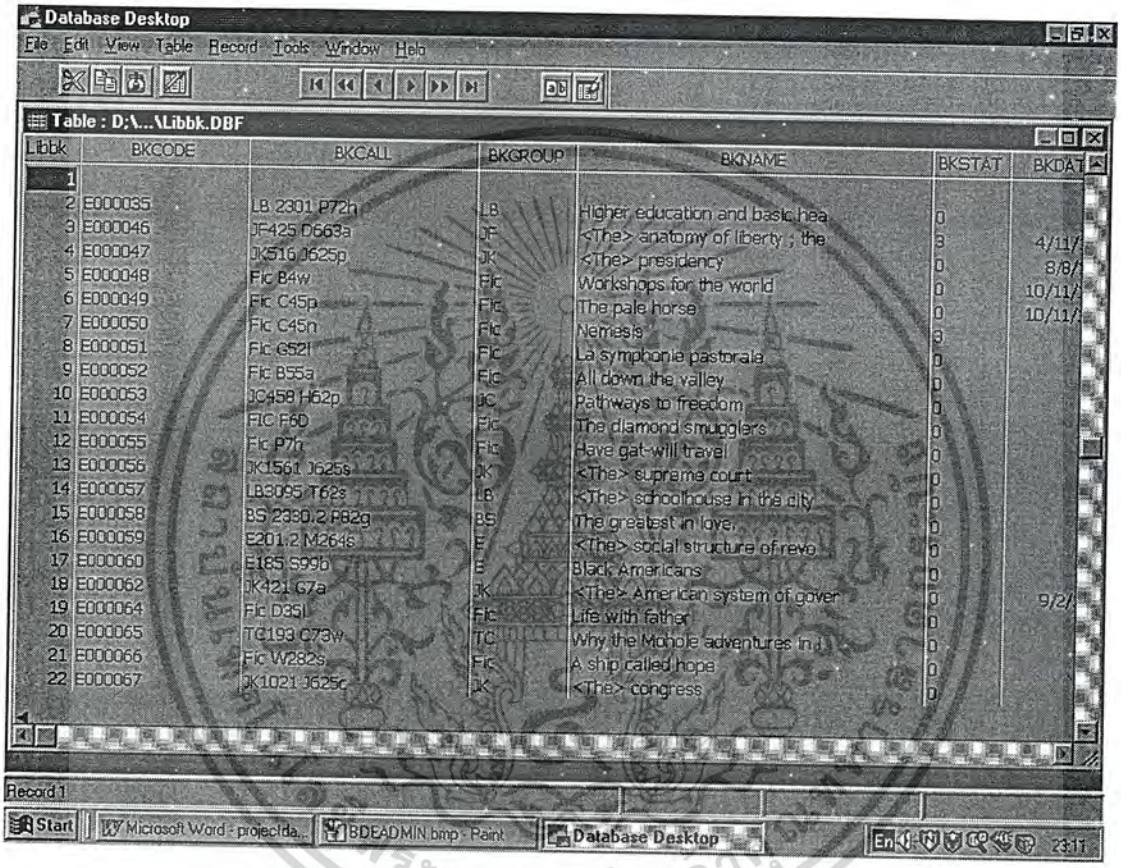
สำหรับเครื่องที่จะใช้งาน Delphi 4 ได้ จำเป็นจะต้องมีส่วนประกอบดังนี้เป็นอย่างน้อย

- Windows 95/98 หรือ Windows NT
- CPU 80486 หรือ Pentium หรือสูงกว่า
- หน่วยความจำ 8 MB(16 MB สำหรับ Windows 95/98 และ 32 MB สำหรับWindows NT)
- จอภาพ VGA หรือจอภาพที่มีรายละเอียดสูงกว่า
- เครื่องอ่าน CD-ROM
- Mouse
- เนื้อที่บนฮาร์ดดิสก์ที่ต้องการขึ้นอยู่กับรูปแบบการติดตั้ง และเวอร์ชั่นของ Delphi

โปรแกรมและเครื่องมือต่างๆของ เดลไฟล์ 4 (Delphi 4)

1. ดาต้าเบสเดสก์ทอป (Database Desktop)

เป็นโปรแกรมสำหรับจัดการเกี่ยวกับงานทางด้านฐานข้อมูลต่างๆ ไม่ว่าจะเป็นการสร้าง แสดง และการแก้ไขตารางฐานข้อมูลต่างๆรวมทั้งการจัดการเกี่ยวหับอินเด็กซ์ของฐานข้อมูล โดยสามารถสร้างอินเด็กซ์ได้



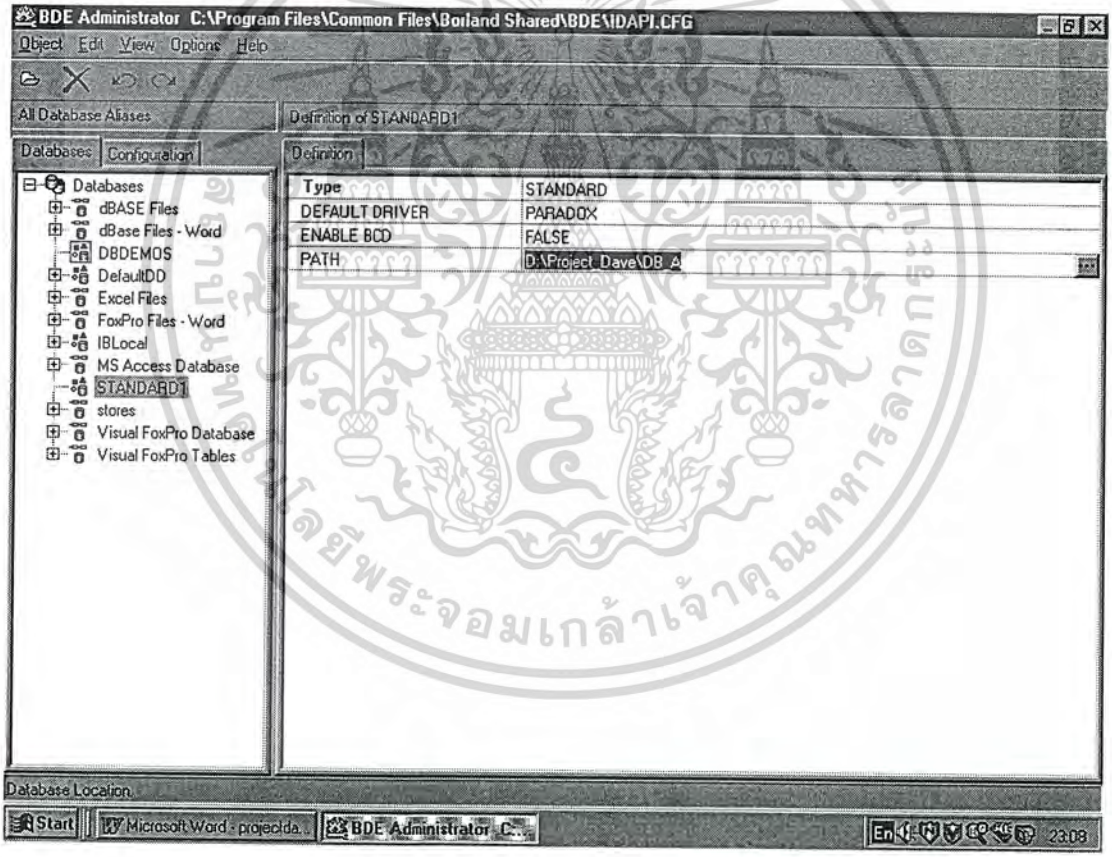
Libbk	BKCODE	BKCALL	BKGROUP	BKNAME	BKSTAT	BKDAT
1						
2	E000035	LB 2301 P72h	LB	Higher education and basic hea	0	
3	E000046	JF425 D663a	JF	<The> anatomy of liberty ; the	3	4/11/
4	E000047	JK516 J625p	JK	<The> presidency	0	8/8/
5	E000048	Fic B4w	Fic	Workshops for the world	0	10/11/
6	E000049	Fic C45p	Fic	The pale horse	0	10/11/
7	E000050	Fic C45n	Fic	Nemesis	3	
8	E000051	Fic G52l	Fic	La symphonie pastorale	0	
9	E000052	Fic B55a	Fic	All down the valley	0	
10	E000053	JC458 H62p	JC	Pathways to freedom	0	
11	E000054	FIC F6D	Fic	The diamond smugglers	0	
12	E000055	Fic P7h	Fic	Have gat-will travel	0	
13	E000056	JK1561 J625s	JK	<The> supreme court	0	
14	E000057	LB3095 T62s	LB	<The> schoolhouse in the city	0	
15	E000058	BS 2330.2 P82g	BS	The greatest in love,	0	
16	E000059	E201.2 M264s	E	<The> social structure of revo	0	
17	E000060	E185 S99b	E	Black Americans	0	
18	E000062	JK421 G7a	JK	<The> American system of gover	0	9/2/
19	E000064	Fic D35l	Fic	Life with father	0	
20	E000065	TC193 C73w	TC	Why the Monole adventures in t	0	
21	E000066	Fic W282s	Fic	A ship called hope	0	
22	E000067	JK1021 J625c	JK	<The> congress	0	

2. บีดีอี แอดมินิสเตรเตอร์ (BDE Administrator)

เนื่องจากเดลไฟล์ใช้ บอร์แลนด์ดาต้าเบสเอนจิน (Borland Database Engine : BDE) เป็นเครื่องมือสำหรับกำหนดคุณสมบัติต่างๆ ที่ใช้ในการติดต่อกับฐานข้อมูล ดังนั้นเดลไฟล์จึงสร้างส่วนนี้เพื่อจัดการ BDE อีกที

หน้าที่การทำงานของ บีดีอี แอดมินิสเตรเตอร์ (BDE Administrator)

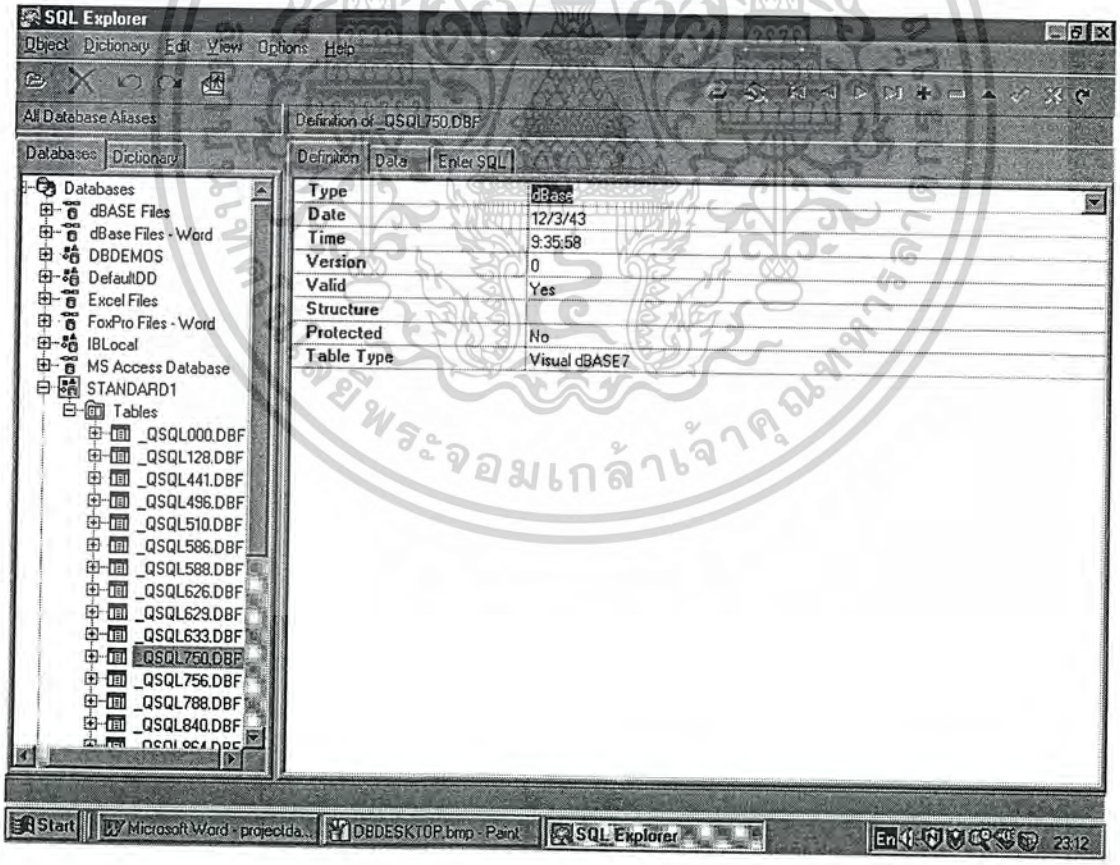
- กำหนดคุณสมบัติต่างๆของ บีดีอี(BDE)
- กำหนดคุณสมบัติของสแตนดาร์ด (Paradox และ Dbase/Foxpro) ,SQL,แอกเซส (Access), และ ODBC ไดร์ฟเวอร์
- สร้าง และลบ ODBC ไดร์ฟเวอร์
- สร้าง และแก้ไข ดาต้าเบส Alias



3. ดาต้าเบสเอ็กซ์โพรเลอร์ (Database Explorer)

เป็นโปรแกรมที่ใช้ในการแสดงฐานข้อมูล โดยจะสามารถแก้ไขข้อมูลได้ด้วยโดยจะมีการทำงานที่แตกต่างกันตามเวอร์ชันดังนี้

เวอร์ชันของ Delphi	ชื่อ Database Explorer	หมายเหตุ
Client/Server Suit และ Enterprice Edition	SQL Explorer	■ สามารถเขียน SQL Query ได้ ■ เข้าถึง SQL Database ที่อยู่บนเครื่อง Server ■ มี Data Dictionary
Developer	Database Explorer	■ เข้าถึงข้อมูลใน Local Database และ Local InterBase Server (LIBS) ■ สามารถเข้าถึงฐานข้อมูลผ่านทาง ODBC ได้ ■ มี Data Dictionary
Desktop	Database Explorer	■ เข้าถึง Local Database ได้เท่านั้น

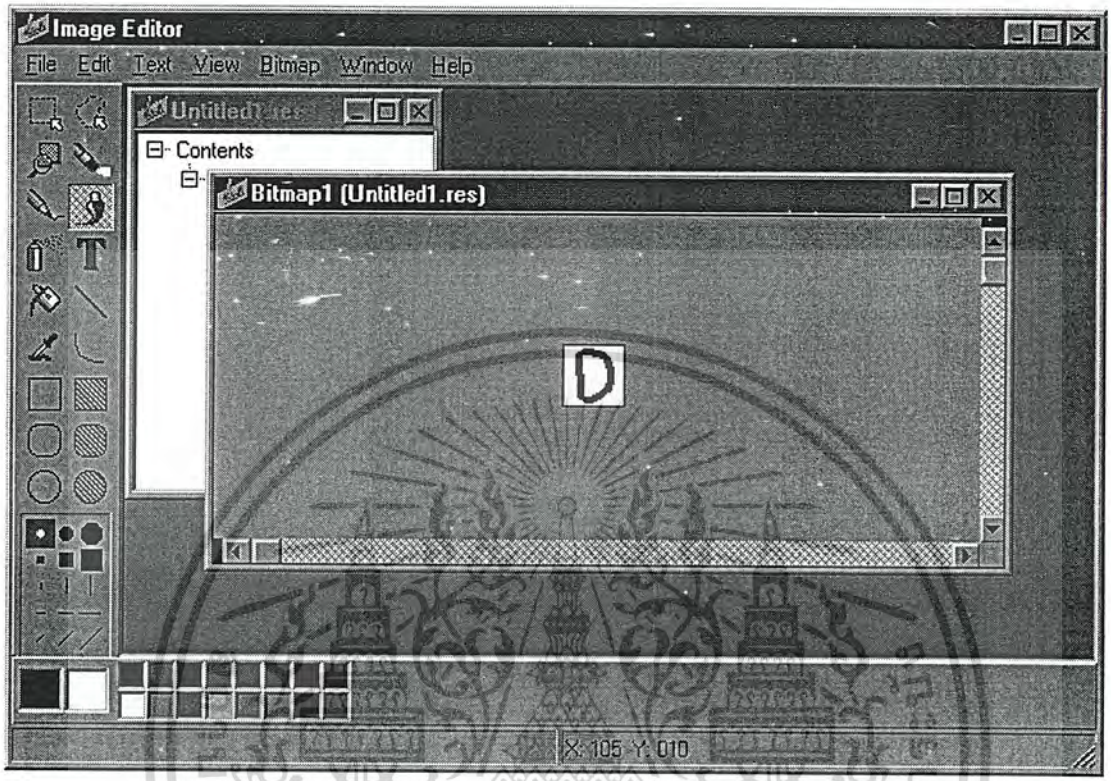


4. Winsight32

เป็นเครื่องมือที่ช่วยตรวจสอบการทำงานของโปรแกรม (Debug) เพื่อใช้ค้นหาว่ามีข้อผิดพลาดในการทำงานร่วมกับ วินโดวส์ หรือไม่ โดยจะมีข้อมูลที่แสดงอยู่ 3 ชนิดคือ ข้อมูลแสดงวินโดวส์(Windows) ,คลาส(Class) และ แมสเซส (Message) ซึ่งปกติจะแสดงข้อมูลเพียง 2 ประเภทคือ วินโดวส์ และ แมสเซส

5. อิมเมจอิดิตเตอร์ (Image Editor)

ใช้ในการสร้าง และแก้ไขรูปภาพต่างๆ ที่จำเป็นต้องใช้ในการพัฒนาแอปพลิเคชั่น



6. คาต้าปั้ม (Data Pump)

เป็นโปรแกรมที่ใช้สำหรับการโอนย้ายฐานข้อมูล ซึ่งอาจเป็นการโอนย้ายจากฐานข้อมูลประเภทหนึ่งไปเป็นอีกประเภทหนึ่ง เช่น ย้ายข้อมูลจากฐานข้อมูลของ Paradox ไปเป็นข้อมูลในฐานข้อมูลของ MS Access

Data Pump สามารถโอนย้ายฐานข้อมูลได้ทุกประเภทที่ Borland Database Engine สามารถเข้าถึง

ภาคผนวก ข

การเขียนโปรแกรมสื่อสารผ่านพอร์ตอนุกรมโดยใช้ MSCOMM Control

MSCOMM Control ใช้สำหรับการติดต่อกับพอร์ตอนุกรมทั้งการรับและการส่งข้อมูลให้กับพอร์ตอนุกรมซึ่งจะมีการติดต่อได้ 2 แบบคือ

การสื่อสารแบบกระตุ้นด้วยเหตุการณ์ (Even-driven communications) หรือเป็นการเขียนโปรแกรมแบบกระตุ้นด้วยอินเทอร์รัพต์นั่นเอง โดยเมื่อมีข้อมูลเข้ามาก็จะเกิด เหตุการณ์ (CommEvent) ขึ้นกับ OnComm Events

การสื่อสารแบบโพลลิ่ง(Polling)โดยการวนรอบตรวจสอบข้อมูลจากพอร์ตอนุกรมตลอดเวลา

MSCOMM Control ประกอบไปด้วย อีเวนท์(Event) และ พรอพโพรตี(Property) ดังนี้

1. อีเวนท์ของ MSCOMM Control จะมีเพียงอีเวนท์เดียวเท่านั้นคือ OnComm ซึ่งประกอบด้วยพรอพโพรตีดังต่อไปนี้

เมื่อเกิดเหตุผิดพลาดในการสื่อสารจะแสดงใน CommEvent ดังนี้

CommEvent Property	ความหมาย
ComEventBreak	ได้รับสัญญาณเบรก
CommEventCDTO	เกิดไทม์เอาต์ขณะที่กำลังรอสัญญาณ CD (Carrier Detect)
ComEventCTSCO	เกิดไทม์เอาต์ขณะที่กำลังรอสัญญาณ CTS (Clear To Send)
ComEventDSRTO	เกิดไทม์เอาต์ขณะที่กำลังรอสัญญาณ DSR (Data Set Ready)
ComEventFrame	เกิดความผิดพลาดทางเฟรม คือ ไม่พบบิตจบตามที่ควรจะเป็น
CommEventOverrun	เกิดความผิดพลาดโอเวอร์รัน คือ การที่รับข้อมูลไม่ทันในการประมวลผล
ComEventCRxOver	บัฟเฟอร์รับข้อมูลเกิดโอเวอร์โฟวล์ หรือ รับตัวอักษรหลังการรับ EOF Char
ComEventRxParity	เกิดความผิดพลาดทางพาริตี้ คือ ตัวอักษรที่รับได้มีพาริตี้ไม่ถูกต้อง
ComEventTxFull	บัฟเฟอร์ส่งข้อมูลเต็ม
CommEventDCB	ความผิดพลาดที่เกิดขึ้นโดยไม่ได้คาดถึง

เมื่อเกิดสถานะการสื่อสารขึ้น จะแสดงใน พรอพโพรตีCommEvent ดังนี้

CommEvent Property	ความหมาย
ComEvCD	CD(Carrier Detect)เปลี่ยนสถานะคือสายสัญญาณ Receive Line Signal Detect (RLSD)
ComEvCTS	CTS (Clear To Send) มีการเปลี่ยนสถานะเกิดขึ้น
ComEvDSR	DSR (Data Set Ready) มีการเปลี่ยนสถานะเกิดขึ้น
ComEvRecieve	ได้รับข้อมูลเก็บลง Input Buffer
ComEvSend	ส่งข้อมูลออกจาก Output Buffer
ComEvEof	พบอักขระ EOF

พรอพโพรตีของ MSCOMM Control ประกอบด้วยพรอพโพรตีต่างๆดังนี้

Property	ความหมาย
Break	กำหนดหรือเคลียร์สัญญาณเบรก
CDHolding	ตรวจสอบสัญญาณ carrier detect (CD) ว่ายังคงมีสถานะอยู่หรือไม่
CDTimeout	กำหนดค่าหรือให้ค่าของเวลาที่รอสัญญาณ Carrier Detect (CD)
CommEvent	ให้ผลของการเกิด Event ของ Communication
CommID	ให้ผลของเฮลเดิลของ communication ที่เปิด
CommPort	กำหนดหรือให้ผลของหมายเลขคอมพอร์ตที่เปิด
CTSHolding	ตรวจสอบสัญญาณ Clear To Send ว่ายังมีสถานะอยู่หรือไม่
CTSTimeout	กำหนดหรือให้ค่าของเวลาที่รอสัญญาณ Clear To Send
DSRHolding	ตรวจสอบสัญญาณ Data Set Ready ว่ายังมีสถานะอยู่หรือไม่
DSRTimeout	กำหนดหรือให้ค่าของเวลาที่รอสัญญาณ Data Set Ready
DTREnable	อีนาเบิล สายสัญญาณ Data Terminal Ready
Handshaking	กำหนดการแฮนด์เช็คทางฮาร์ดแวร์เพื่อคอยตรวจสอบการรับส่งข้อมูล
InBufferCount	ให้ผลของจำนวนข้อมูลที่อยู่ในบัฟเฟอร์รับข้อมูล
InBufferSize	กำหนดหรือให้ขนาดของบัฟเฟอร์รับข้อมูล
Input	ให้ผลหรือย้ายข้อมูลจากบัฟเฟอร์รับข้อมูล
InputLen	กำหนดหรือให้ผลขนาดของบัฟเฟอร์รับข้อมูล
Interval	กำหนดอัตราความเร็วสำหรับการใช้งานในโหมดโพลลิ่ง
NullDiscard	กำหนดให้มีหารรับ Null เก็บลงบัฟเฟอร์รับข้อมูล
OutBufferCount	จำนวนข้อมูลที่รออยู่ในบัฟเฟอร์ส่งข้อมูล
OutBufferSize	กำหนดหรือให้ผลขนาดของบัฟเฟอร์ส่งข้อมูล
Output	ส่งข้อมูลให้กับบัฟเฟอร์ส่งข้อมูล
ParityReplace	กำหนดให้ส่งอักขระนี้แทนหากเกิดการผิดพลาดในข้อมูล
PortOper	กำหนดหรือให้ผลของสถานะของพอร์ต
Rtheshold	กำหนดหรือให้ผลของจำนวนข้อมูลที่เก็บลงบัฟเฟอร์รับข้อมูลก่อนเกิด CommEvent ในการรับข้อมูล
RTSEnable	อีนาเบิลสัญญาณ Request to Send
Setting	กำหนดอัตราบอด พาริตี้ ข้อมูล บิตหยุด
Stheshold	กำหนดหรือให้ผลของจำนวนข้อมูลที่เก็บลงบัฟเฟอร์ส่งข้อมูลก่อนการเกิด CommEvent ในการส่งข้อมูล

ภาคผนวก ค

การเขียนคำสั่งภาษา SQL ด้วย Local SQL

SQL เป็นคำสั่งมาตรฐานที่ใช้ในการแสดง/แก้ไขข้อมูลที่มีอยู่ในฐานข้อมูล ซึ่งระบบฐานข้อมูลที่เป็นที่นิยมโดยทั่วไป จะสามารถใช้คำสั่งภาษา SQL เป็นคำสั่งหลักในการจัดการกับข้อมูล การเรียนรู้การใช้คำสั่ง SQL จะเป็นประโยชน์อย่างมากในการจัดการกับฐานข้อมูล

แนะนำ Local SQL

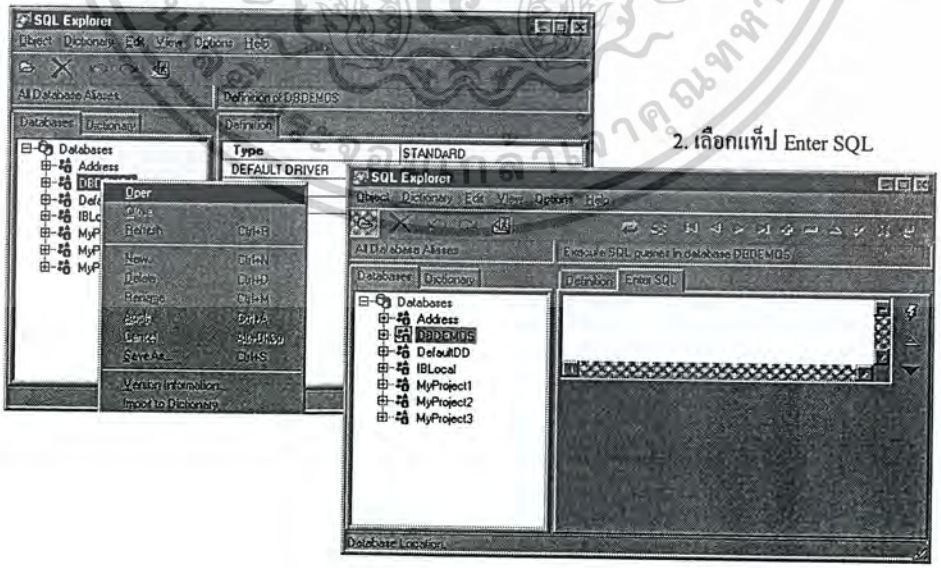
Delphi มีรูปแบบของภาษา SQL สำหรับใช้ในการแสดงและแก้ไขข้อมูลของตัวองเรียกว่า Local SQL ซึ่งเป็นภาษา SQL ที่มีรูปแบบตาม Standard SQL (ภาษา SQL มาตรฐาน)

ถึงแม้ Delphi จะใช้ Local SQL เป็นภาษามาตรฐานในการจัดการกับฐานข้อมูล แต่ในกรณีที่เราเชื่อมต่อกับระบบจัดการฐานข้อมูลที่มีภาษา SQL ของตัวเอง เราสามารถใช้คำสั่ง SQL ของระบบฐานข้อมูลนั้นๆ ก็ได้ ซึ่งข้อดีของการใช้ภาษา SQL ของระบบจัดการฐานข้อมูล ก็คือ ทำให้เราสามารถที่จะควบคุมการทำงานในฐานข้อมูลได้อย่างมีประสิทธิภาพสูงสุด แต่ข้อดีในการใช้ Local SQL ซึ่งเป็นภาษา SQL มาตรฐานของ Delphi คือจะทำให้เราสามารถโอนย้ายการทำงานของแอปพลิเคชั่นไปยังระบบฐานข้อมูลประเภทอื่นๆ ได้โดยที่เราไม่จำเป็นต้องแก้ไขตัวโปรแกรมเลย หรือแก้ไขเพียงเล็กน้อยเท่านั้น

การใช้คำสั่งภาษา SQL ในโปรแกรม SQL Explorer

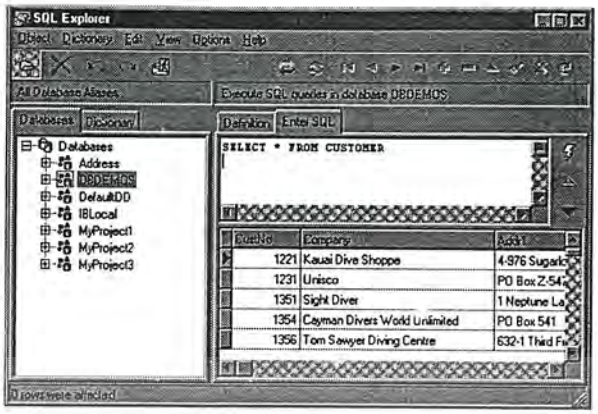
เราสามารถแสดงและแก้ไขข้อมูลด้วยคำสั่งภาษา SQL ได้ในโปรแกรม SQL Explorer สามารถทำได้ดังขั้นตอนดังนี้

1. เรียกโปรแกรม SQL Explorer เลือกเปิดฐานข้อมูล



2. เลือกแท็ป Enter SQL

3. ใส่คำสั่ง SQL



รูปที่ ก.1 แสดงขั้นตอนการใช้คำสั่งภาษา SQL ในโปรแกรม SQL Explorer

DML และ DDL

Local SQL และ SQL ทั่วไป แบ่งออกเป็น 2 ลักษณะ คือ DML (Data Manipulation Language) ภาษาสำหรับปรับปรุงข้อมูลและ DDL (Data Definition Language) ภาษาสำหรับปรับปรุงโครงสร้างฐานข้อมูล

Data Manipulation Language

Data Manipulation Language (DML) คือ คำสั่ง SQL ที่ใช้ในการแสดง และแก้ไขข้อมูล ได้แก่

- SELECT เป็นคำสั่งที่ใช้ในการแสดงข้อมูลในฐานข้อมูล
- INSERT เป็นคำสั่งที่ใช้ในการเพิ่มข้อมูลในฐานข้อมูล
- UPDATE เป็นคำสั่งที่ใช้ในการแก้ไขข้อมูล
- DELETE เป็นคำสั่งที่ใช้ในการลบข้อมูล

Data Definition Language

Data Definition Language (DDL) คือ คำสั่ง SQL ที่ใช้สำหรับการสร้าง (CREATE), เปลี่ยนแปลง (ALTER) และลบ (DROP) ตาราง และอินเด็กซ์ในฐานข้อมูล ซึ่งคำสั่งจำพวกนี้ เราสามารถใช้โปรแกรม Database Desktop ทดแทนได้อีกทั้งยังสะดวกและรวดเร็วกว่า

คำสั่ง SELECT

คำสั่ง SELECT เป็นคำสั่งที่ใช้ในการแสดงข้อมูลในฐานข้อมูล ซึ่งการแสดงข้อมูลด้วยคำสั่ง SELECT นั้น เราสามารถดึงข้อมูลจากตารางได้มากกว่า 1 ตาราง โดยเราสามารถ กำหนดความสัมพันธ์ของแต่ละตารางได้ นอกจากนี้ยังสามารถระบุเงื่อนไขในการดึงข้อมูล และสามารถระบุรูปแบบการเรียงลำดับข้อมูลได้

โครงสร้างของคำสั่ง SELECT คือ

```
SELECT [DISTINCT] รายชื่อของฟิลด์
        FROM  รายชื่อชื่อตาราง
        [WHERE  เงื่อนไข – where]
        [ORDER  BY การเรียงลำดับ]
        [GROUP BY การแบ่งกลุ่ม]
        [HAVING เงื่อนไข – having]
        [UNION คำสั่ง – select]
```

รายชื่อฟิลด์ คือ รายชื่อของฟิลด์ที่เราต้องการดึงข้อมูลมาแสดง เช่น

```
SELECT CustNo, Company FROM Customer
```

ถ้าเรามีการระบุคำสั่ง DISTINCT ข้างหน้ารายชื่อฟิลด์ จะทำให้เราได้ผลลัพธ์ที่มีข้อมูลไม่ซ้ำกัน เช่น

```
SELECT DISTINCT State FROM Customer
```

รายชื่อตาราง คือ ชื่อของตารางที่เราจะดึงข้อมูลมาใช้ ซึ่งเราอาจระบุชื่อตารางเดียว หรือ หลายตารางก็ได้ โดยในกรณีที่เราระบุชื่อของตารางมากกว่าหนึ่งตาราง เราต้องใส่เครื่องหมายคอมมา (,) คั่นระหว่างชื่อของตารางด้วย

```
SELECT CustNo, Company, OrderNo, SaleDate
FROM    Customer, Orders
WHERE  Customer.CustNo = Orders.CustNo
```

นอกจากนี้ยังสามารถเชื่อมข้อมูลแบบ Inner และ Outer Join (การเชื่อมสองตารางโดยกำหนดให้แสดงข้อมูลจากตารางใดตารางหนึ่งทั้งหมด) ได้ด้วย

```
SELECT CustNo, Company, OrderNo, SaleDate
FROM    Customer LEFT OUTER JOIN Orders ON
        Customer.CustNo = Orders.CustNo
```

เงื่อนไข – **WHERE** เป็นการระบุเงื่อนไขการแสดงข้อมูล ซึ่งข้อมูลที่ตรงตามเงื่อนไขเท่านั้น ถึงจะถูกเลือกมาแสดง

```
SELECT * FROM Items WHERE Qty > 5
```

เราสามารถใช้โอเปอร์เรเตอร์ IN เพื่อเลือกแสดงเฉพาะข้อมูลที่มีอยู่ในลิสต์ได้

```
SELECT * FROM Items WHERE Qty IN (5, 8, 10)
```

เราสามารถระบุรูปแบบการเรียงลำดับข้อมูล โดยใส่ประโยค ASC หรือ DESC หน้าชื่อฟิลด์ เพื่อระบุให้มีการเรียงลำดับข้อมูลจากน้อยไปมาก หรือจากมากไปน้อย ตามลำดับ

```
SELECT CustNo, Company FROM Customer
ORDER BY Company ASC
```

สามารถระบุรูปแบบการจัดกลุ่มของข้อมูล โดยการใช้ประโยค GROUP BY ในคำสั่ง SELECT ทุกๆ ฟิลด์ที่แสดงจะต้องอยู่ในการจัดกลุ่ม นอกจากจะมีการใช้ฟังก์ชั่นต่างๆ เช่น SUM, COUNT, MAX, MIN กับฟิลด์เหล่านั้น

```
SELECT CustNo, COUNT(OrderNo) FROM Orders
GROUP BY CustNo
```

เงื่อนไข – **HAVING** จะทำหน้าที่คล้ายๆกับเงื่อนไขที่อยู่หลังประโยค WHERE แตกต่างกันตรงที่เงื่อนไขที่ระบุหลังประโยค HAVING นี้ จะเป็นเงื่อนไขที่มักจะอยู่ในฟังก์ชั่นต่างๆ

```
SELECT CustNo, COUNT(OrderNo) FROM Orders
GROUP BY CustNo HAVING COUNT(OrderNo) > 3
```

ประโยค **UNION** เป็นการระบุให้รวมผลลัพธ์จากคำสั่ง SELECT 2 คำสั่งหลังประโยค UNION จะตามด้วยคำสั่ง SELECT อีกคำสั่งหนึ่ง ซึ่งผลลัพธ์ของคำสั่งทั้งสองจะรวมกันเป็นผลลัพธ์เดียว

```
SELECT Contact as Name FROM Customer
UNION
Select LastName as Name FROM Employee
```

คำสั่ง INSERT

คำสั่ง INSERT เป็นคำสั่งที่ใช้ในการเพิ่มข้อมูลในตารางฐานข้อมูล

รูปแบบของคำสั่ง INSERT คือ

```
INSERT INTO ชื่อตาราง
[รายชื่อฟิลด์]
VALUES ค่าของข้อมูลที่จะเพิ่ม
```

- ชื่อตาราง คือ ชื่อของตารางที่เราจะเพิ่มข้อมูลเข้าไป
- รายชื่อฟิลด์ คือ รายชื่อฟิลด์ที่จะมีการใส่ข้อมูล
- ค่าของข้อมูลที่เพิ่ม คือ ค่าของข้อมูลใหม่ที่จะเพิ่มในฐานข้อมูล เช่น

```
INSERT INTO Customer
(CustNo, Company, Contact)
VALUES (40001, "Test", "Peter")
```

เราสามารถละรายชื่อฟิลด์ ได้ในกรณีที่เราต้องการกำหนดค่าให้กับทุกๆ ฟิลด์ เช่น

```
INSERT INTO Holdings
VALUES (4094095, "BORL", 5000, 10.500, "1/2/1998")
```

เราสามารถเพิ่มข้อมูลในตาราง โดยนำข้อมูลมาจากตารางอื่นได้ โดยใช้คำสั่ง SELECT แทนในประโยค VALUES เช่น

```
INSERT INTO Customer
(CustNo, Company)
SELECT CustNo, Company
FROM OldCustomer
```

คำสั่ง UPDATE

คำสั่ง UPDATE เป็นคำสั่งที่ใช้ในการปรับปรุง เปลี่ยนแปลงข้อมูลในฐานข้อมูล
โครงสร้างของคำสั่ง UPDATE คือ

```
UPDATE ชื่อตาราง
SET    ชื่อฟิลด์ = ค่าของข้อมูล [, ชื่อฟิลด์ = ค่าของข้อมูล ...]
[WHERE เงื่อนไข – Where]
```

- ชื่อตาราง คือ ชื่อตารางที่จะทำการแก้ไข
- ชื่อฟิลด์ คือ ชื่อฟิลด์ที่ต้องการแก้ไข ซึ่งเราสามารถแก้ไขข้อมูลมากกว่าหนึ่งฟิลด์ได้ โดยใช้เครื่องหมาย "," คั่นระหว่างการแก้ไขของแต่ละฟิลด์
- ค่าของข้อมูล คือ ค่าที่จะนำไปใส่ในฟิลด์แทนที่ข้อมูลเดิม
- เงื่อนไข – where คือ เงื่อนไขการแก้ไขข้อมูล เช่น

```
UPDATE  Employee
SET     Salary = 45000
WHERE  EmpNp = 2
```

คำสั่ง DELETE

คำสั่ง DELETE เป็นคำสั่งในการลบข้อมูลในตาราง
โครงสร้างของคำสั่ง DELETE คือ

```
DELETE FROM ชื่อตาราง
[WHERE  เงื่อนไข  - where]
```

- ชื่อตาราง คือ ชื่อตารางที่ต้องการจะลบข้อมูล
- เงื่อนไข – where คือ เงื่อนไขในการลบข้อมูล เช่น

```
DELETE FROM  ORDER
```

```
DELETE FROM ANIMALS
WHERE  NAME = 'Angel Fish'
```

บรรณานุกรม

[1] กนก กุสุมาลย์นุกูล และ ไกรวุฒิ มั่นเสถียรสิน(1989) : "คู่มือการเขียนโปรแกรม Delphi 4",Success Media CO..LTD.

[2] ศุภมิตร จิตตะยโศธร,ดร. : "เอกสารประกอบการสัมนา Database Design and SQL",สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

[3] สุนทร วิทูสุรพจน์ : "การโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051", ซีเอ็ดยูเคชั่น

[4] http://www.activebarcode.com/index.htm

[5] http://www.punmanee.com

[6] http://www.delphisource.com