การพัฒนาโปรแกรมเชิงวัตถุ

OBJECT - ORIENTED PROGRAMMING

โดย

นาย ชาญสิทธิ์ ศุภรานนท์รัตน์

นาย สมศักดิ์ ร่วมกุลกิจ

อาจารย์ที่ปรึกษา

ดร. เอื้อน ปิ่นเงิน

วัน เดือน ปี 15 ธค. 2541

เลขทะเบียน 038962

เลขเรียกหนังสือ T 40103 ศ 481 ก

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

038962

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาโปรแกรมเชิงวัตถุ

ผู้จัดทำ

1. นายชาญสิทธิ์ ศุภรานนท์รัตน์
2. นายสมศักดิ์ ร่วมกุลกิจ

..... อาจารย์ที่ปรึกษา

(_____)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพัฒนาโปรแกรมเชิงวัตถุ

นาย ชาญสิทธิ์ ศุภรานนท์รัตน์
นาย สมศักดิ์ ร่วมกุลกิจ

ดร. เอื้อน ปิ่นเงิน อาจารย์ที่ปรึกษา
ปีการศึกษา 2540

บทคัดย่อ

ปัจจุบันสำนักหอสมุดกลางเมื่อถึงกำหนด จะต้องมีการตรวจสอบครุภัณฑ์ของสำนักหอสมุดกลาง ซึ่งครุภัณฑ์เหล่านี้มีอยู่เป็นจำนวนมาก ทำให้ยากลำบากต่อการตรวจสอบ ด้วยเหตุผลนี้ทางภาควิชาวิศวกรรมคอมพิวเตอร์ จึงได้นำคอมพิวเตอร์เข้ามาช่วยในการจัดเก็บข้อมูล และตรวจสอบครุภัณฑ์เหล่านั้น โดยข้อมูลของครุภัณฑ์เหล่านี้จะต้องสามารถระบุถึงรายละเอียด สถานที่ตั้ง และข้อมูลทางภาพ ของครุภัณฑ์แต่ละชิ้น โดยที่ครุภัณฑ์แต่ละชิ้นจะติดบาร์โค๊ด (Barcode) ซึ่งจะใช้แทนรหัสของครุภัณฑ์ เพื่อความสะดวกต่อการทำงาน

โครงงานนี้ได้แบ่งการทำงานออกเป็น 2 ส่วน คือ ในส่วนแรกซึ่งออกแบบและพัฒนาโดยใช้โปรแกรมเชิงวัตถุ เป็นส่วนปฏิบัติการเกี่ยวกับฐานข้อมูล ซึ่งในฐานข้อมูลนั้นประกอบด้วย รหัสครุภัณฑ์ ประเภทของครุภัณฑ์ แบบครุภัณฑ์ ชนิดครุภัณฑ์ และภาพของครุภัณฑ์ รวมถึงสถานะของครุภัณฑ์ชิ้นนั้น ๆ ด้วย ในโปรแกรมนี้จะมีการค้นหารายละเอียด เพิ่มครุภัณฑ์ เพิ่มและแก้ไขรายละเอียดต่าง ๆ จนกระทั่งถึงการติดต่อสื่อสารกับกล่องเก็บข้อมูลรหัสแถบ ในส่วนที่ 2 เป็นส่วนของกล่องเก็บข้อมูลรหัสแถบ (Hardware) ใช้ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวเอ็มซีเอส 51 เป็นอุปกรณ์ควบคุมการทำงานของส่วนนี้และในส่วนนี้ยังมีวงจรชาร์จแบตเตอร์รี่รวมอยู่ด้วย ทั้งนี้เพื่อสะดวกในการนำไปใช้งาน เมื่อถึงกำหนดที่ทางสำนักหอสมุดกลาง จะทำการตรวจสอบครุภัณฑ์ เจ้าหน้าที่จะนำเครื่องอ่านรหัสแถบและกล่องเก็บข้อมูลรหัสแถบ ไปทำการอ่านรหัสแถบที่ติดไว้กับครุภัณฑ์แต่ละชิ้น ในสำนักหอสมุดกลาง หลังจากนั้นนำข้อมูลที่อ่านได้เข้าคอมพิวเตอร์ เพื่อนำข้อมูลนั้นไปประมวลผล โดยการประมวลผลนั้นจะให้ผลลัพธ์ว่าครุภัณฑ์ชิ้นใดที่ผ่านการตรวจสอบแล้ว และครุภัณฑ์ชิ้นใดที่ยังไม่ได้ผ่านการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OBJECT - ORIENTED PROGRAMMING

Chansit Supranonrut

Somsak Ruamkulkij

Dr. Ouen Pinngern Advisor

1997

Abstract

Nowaday , the Central Library is required for checking a lot of assets in its period and it is difficult . Therefore , we try to apply the computer system to collect all data and check those assets . It consists of each location and photograph detials of assets . Barcode system is used for substituting the assets code to facilitate .

This project was divided into 2 parts. The first part is a software that operates on database which consists of assets code , assets type , form of assets , picture of assets and assets status. The program also facilitates the search for assets details , modify the assets database via Barcode Box. The second part is the part of Barcode Box , which works with microcontroller single MCS 51 to control the part and there is the Battery-Changer Circuit to support processing. In principle , when an officer checked the assets , the decoding machine would identify each barcode of assets . Then the decoded data were transfered to computer system for processing those data . Finally , they have been checked .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่โครงงานนี้สำเร็จลุล่วงไปได้ด้วยดีนั้น อันเป็นผลสืบเนื่องมาจากความรู้พื้นฐานหลาย ๆ แขนงที่ได้เรียนมา คำแนะนำต่าง ๆ ที่ได้รับจากอาจารย์ในภาควิชาวิศวกรรมคอมพิวเตอร์ทุกท่าน โดยเฉพาะอาจารย์ที่ปรึกษา คือ อาจารย์ ดร. เอื้อน ปิ่นเงิน , คุณ รัชนี วิมลภักตร์ และคณะเจ้าที่ของสำนักหอสมุดกลางทุกท่าน ที่คอยให้ข้อมูลและคำปรึกษาที่เกี่ยวข้องกับการทำโครงงานด้วยดีมาตลอด คอยชี้แนะสิ่งที่เป็นประโยชน์ต่อโครงงานนี้เป็นอย่างมาก และขอบคุณเพื่อน ๆ ทุกท่านที่ได้ช่วยเหลือให้คำแนะนำสิ่งต่าง ๆ ในสิ่งที่ติดขัด และคอยให้กำลังใจจนกระทั่งโครงงานนี้สำเร็จลุล่วงไปได้

และขอขอบพระคุณอาจารย์ผู้คุมสอบทุกท่านที่ได้สละเวลาอันเป็นประโยชน์ของท่านมาในการสอบโครงงานนี้ให้ผ่านลุล่วงไปด้วยดีครั้งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หัวเรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VIII
สารบัญรูปภาพ	IX
บทที่ 1 บทนำ	1
1.1 แนวคิดและเหตุผล	1
1.2 วัตถุประสงค์	1
1.3 ประโยชน์ที่คาดว่าจะได้รับ	1
1.4 ขอบเขตของโครงงาน	1
1.5 ลำดับขั้นการทำงาน	2
บทที่ 2 การโปรแกรมเชิงวัตถุ	3
2.1 ความสำคัญของการโปรแกรมเชิงวัตถุ	3
2.2 ภาษาเชิงกระบวนการ	3
2.3 การแบ่งออกไปเป็นฟังก์ชัน	3
2.4 ปัญหาที่เกิดขึ้นกับการเขียนโปรแกรมเชิงโครงสร้าง	4
2.5 แบบแผนของการโปรแกรมเชิงวัตถุ	4
2.6 เทคโนโลยีของการโปรแกรมเชิงวัตถุได้นำประโยชน์ของการพัฒนาซอฟต์แวร์	5
2.7 ประสิทธิภาพในการทำงาน	6
2.8 วิวัฒนาการของการโปรแกรมเชิงวัตถุ	6
2.9 วัตถุ (Object)	7
2.9.1 แอททริบิว (Attribute)	7
2.9.2 กิจกรรม (Method)	7
2.10 คลาส (Class)	8
2.11 การสืบทอดของคลาส (Inheritance)	8
2.11.1 การแก้ไข	9
2.11.2 การต่อเติม	10
2.11.3 เอนแคปซูลเลชั่น (Encapsulation)	10
2.11.4 โพลิมอร์ฟิซึม (Polymorphism)	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวเรื่อง **หน้า**

2.12 การออกแบบเชิงวัตถุ 11

2.13 เอกลักษณ์ของคลาส 12

2.14 การออกแบบจากบนลงล่าง และการออกแบบจากล่างขึ้นบน 13

บทที่ 3 ไมโครคอนโทรลเลอร์แบบชิพเดียวเอ็มซีเอส 51 15

3.1 โครงสร้างของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวเอ็มซีเอส 51 15

3.2 หน่วยความจำ 17

3.3 รีจิสเตอร์ภายในเอ็มซีเอส 17

3.4 ความเร็วของเอ็มซีเอสเทียบกับ 6502 และ Z80 18

3.5 บัสและพอร์ต 18

3.5.1 ขา $\overline{PSEN}$ 18

3.5.2 ขา $\overline{EA}$ 18

3.6 วงจรนับและวงจรตั้งเวลา 19

3.7 พอร์ตชนิดอนุกรมอยู่ภายใน 19

3.8 อินเตอร์รัพต์และชุดคำสั่ง 19

3.9 รีจิสเตอร์ที่เกี่ยวข้อง 21

บทที่ 4 รหัสแถบ (Barcode) 25

4.1 ความเป็นมาของรหัสแถบ และมาตรฐาน 25

4.2 หลักการของรหัสแถบ 25

4.3 การอ่านรหัสแถบ 26

4.4 ชนิดของรหัสแถบ 28

4.4.1 ชนิดรหัส 2 ใน 5 (2 of 2 Code) 28

4.4.2 ชนิดรหัส 2 ใน 5 แบบสอดแทรก (Interleaved 2 of 5) 28

4.4.3 ชนิดรหัส 3 ใน 9 (3 of 9 or 39 Code) 29

4.4.4 ชนิดรหัสโคดาบาร์ (Codabar) 33

4.4.5 ชนิดรหัส UPC หรือ EAN Code (Universal Product Code or European Aritcle Numbering)..... 34

4.5 การพิมพ์รหัสแถบ 42

4.6 ลักษณะมาตรฐานของรหัสแถบ 42

4.7 การกำหนดชุดตัวเลข 43

4.8 เครื่องหมายเสริม (Auxiliary Characters) 44

4.9 ขนาดความยาวตามกำหนดของอักขระ 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวเรื่อง **หน้า**

4.10 แฟกเตอร์ของแสงสะท้อนกลับ และความหนาแน่นของแสงสะท้อน 45
4.11 เงื่อนไขทางเรขาคณิตสำหรับการวัดค่าแสงสะท้อน 45
4.12 เงื่อนไขการแยกเาสีสำหรับการวัดค่าแสงสะท้อน 45
บทที่ 5 การออกแบบ 46
5.1 การออกแบบวงจร 46
5.1.1 ส่วนไมโครโปรเซสเซอร์หลัก 46
5.1.2 ส่วนของหน่วยความจำ 47
5.1.3 ส่วนของการเชื่อมต่อกับเครื่องอ่านหรัสแถบ 48
5.1.4 ส่วนของการเชื่อมต่อกับคอมพิวเตอร์ (RS- 232) 49
5.1.5 ส่วนของวงจรภาคจ่ายไฟ 49
5.1.6 ส่วนของวงจรตรวจจับแรงดัน 50
5.1.7 ส่วนของวงจรประจุแบตเตอรี่ 50
5.1.8 วงจรสร้างเสียง 53
5.2 การออกแบบโปรแกรมเครื่องเก็บรหัสแถบ 53
5.2.1 ส่วนที่ติดต่อสื่อสาร (RS-232-C) 53
5.2.2 ส่วนที่รับข้อมูลจากเครื่องอ่านรหัสแถบ เขียนโปรแกรมแบบ
อินเตอร์รัพต์ 55
5.2.3 ส่วนที่ตรวจสอบแบตเตอรี่แรงดันต่ำ เขียนโปรแกรมแบบ
อินเตอร์รัพต์ 56
5.2.4 ส่วนที่สร้างเสียง 56
5.2.5 โปรแกรมหลัก 56
5.3 การออกแบบโปรแกรม 57
5.3.1 การโปรแกรมในฟอร์มค้นหารายการ 58
5.3.2 การโปรแกรมในฟอร์มเพิ่มรายการครุภัณฑ์ 58
5.3.3 การโปรแกรมในฟอร์มเปลี่ยนแปลงสถานที่ 59
5.3.4 การโปรแกรมในฟอร์มการสำรวจรายการครุภัณฑ์ 60
5.3.5 การโปรแกรมในฟอร์มเพิ่มเติมและแก้ไขรายละเอียดครุภัณฑ์ 60
5.4 การออกแบบฐานข้อมูล 61
บทที่ 6 การโปรแกรม การทดสอบโปรแกรมและการติดตั้ง 63
6.1 การโปรแกรม 63
6.1.1 ก่ารโปรแกรมในฟอร์มค้นหารายการ 63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หัวเรื่อง **หน้า**

6.1.2 การโปรแกรมในฟอร์มเพิ่มรายการครุภัณฑ์ 63

6.1.3 การโปรแกรมในฟอร์มเปลี่ยนแปลงสถานที่ 63

6.1.4 การโปรแกรมในฟอร์มการสำรวจรายการครุภัณฑ์ 63

6.1.5 การโปรแกรมในฟอร์มเพิ่มเติมและแก้ไขรายละเอียดครุภัณฑ์ 63

6.1.6 การโปรแกรมในฟอร์มทดสอบการสื่อสาร 64

6.2 การทดสอบโปรแกรม 64

6.2.1 โปรแกรมที่เครื่องเก็บข้อมูลรหัสแถบ 64

6.2.2 โปรแกรมการเพิ่มครุภัณฑ์ 64

6.2.3 โปรแกรมการค้นหารายการ 64

6.2.4 โปรแกรมการแก้ไขรายละเอียด 64

6.2.5 โปรแกรมการติดต่อสื่อสาร 64

6.3 การติดตั้งและการใช้งานเครื่องเก็บรหัสแถบ 65

6.3.1 การติดตั้งแพื่อประจุแบตเตอรี่ 65

6.3.2 การติดตั้งกับเครื่องอ่านรหัสแถบ 66

6.3.3 การเชื่อมต่อกับคอมพิวเตอร์ 66

บทที่ 7 สรุปผลการพัฒนาและข้อเสนอแนะ 68

7.1 สรุปผลการพัฒนา 68

7.2 ข้อเสนอแนะ 68

บรรณานุกรม 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 3.1 การเลือกโหมดการทำงานของไทเมอร์ / เคาน์เตอร์ โดยใช้ M0 , M1 23
ตารางที่ 3.2 การเลือกโหมดการทำงานและบ็อตเรตของ SCON โดยใช้ M0 และ M1 24
ตารางที่ 4.1 รหัสเลขมาตรฐาน 2 ของรหัส 2 ใน 5 ทั้งสองแบบ 29
ตารางที่ 4.2 เลขฐาน 2 ของรหัส 3 ใน 9 30
ตารางที่ 4.3 รหัสเลขฐานสอง (Binary Code) 31
ตารางที่ 4.4 เลขฐาน 2 ของรหัส Codabar 34
ตารางที่ 4.5 รหัสเลขฐาน 2 ของ UPC โซนทางซ้ายและโซนทางขวา 35
ตารางที่ 4.6 อักขระของรหัส EAN 36
ตารางที่ 4.7 แฟล็กและคาต้า 37
ตารางที่ 4.8 พาริตี้แพตเทิร์นของ 2 หลักที่เพิ่มมาของ EAN 39
ตารางที่ 4.9 พาริตี้แพตเทิร์นของ 5 หลัก ที่เพิ่มมาของ EAN 13 41
ตารางที่ 4.10 รหัสแถบ ANSI 42
ตารางที่ 4.11 การกำหนดชุดตัวเลข 43
ตารางที่ 4.12 เครื่องหมายเสริม 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

หน้า

รูปที่ 2.1 วิวัฒนาการของแบบแผนการโปรแกรม 5
รูปที่ 2.2 การออกแบบจากบนลงล่างและจากล่างขึ้นบน 14
รูปที่ 3.1 การจัดวางขาต่าง ๆ ของเอ็มซีเอส 51 16
รูปที่ 3.2 โครงสร้างภายในหน่วยความจำภายในเอ็มซีเอส 51 17
รูปที่ 3.3 บล็อคไดอะแกรมการทำงานของ MCS-51 20
รูปที่ 4.1 การแทนค่าเลขฐานสองของแถบต่าง ๆ 26
รูปที่ 4.2 ลักษณะและขนาดของรหัสแถบมาตรฐาน 26
รูปที่ 4.3 ลักษณะของเครื่องอ่านแบบที่ต่อกับ Com Port ของไมโครฯ 27
รูปที่ 4.4 รหัสแถบ ชนิด 2 ใน 5 28
รูปที่ 4.5 รหัสแถบ ชนิด 2 ใน 5 แบบสอดแทรก 29
รูปที่ 4.6 รหัสแถบ ชนิดรหัส 3 ใน 9 30
รูปที่ 4.7 Code 39 เป็นบาร์โค้ดที่ใช้กันอย่างกว้างขวาง โดยเฉพาะสินค้าขายปลีก
ทั่วไป 33
รูปที่ 4.8 รหัสแถบ ชนิด Codabar 33
รูปที่ 4.9 รหัสแถบ ชนิด UPC/EAN 35
รูปที่ 4.10 EAN 13 ใช้กับข้อความที่มีความยาว 10 หลัก 38
รูปที่ 4.11 ตัวอย่างบาร์โค้ด 5 หลัก ที่เพิ่มมาของ EAN 40
รูปที่ 5.1 แผนภาพแสดงส่วนประกอบของเครื่องเก็บหรัสแถบ 46
รูปที่ 5.2 วงจรป้องกันความผิดพลาดของ ไมโครโปรเซสเซอร์ 47
รูปที่ 5.3 การเชื่อมต่อหน่วยความจำ 47
รูปที่ 5.4 การทำงานของ Keyboard 48
รูปที่ 5.5 วงจรส่วนของการเชื่อมต่อกับเครื่องอ่านรหัสแถบ 48
รูปที่ 5.6 การเชื่อมต่อระบบสื่อสาร 49
รูปที่ 5.7 วงจรภาคจ่ายไฟ 49
รูปที่ 5.8 วงจรเปรียบเทียบแรงดัน 50
รูปที่ 5.9 โครงสร้างภายในนิกเกิล-เมตัลไฮ ไดรด์ 51
รูปที่ 5.10 วงจรประจุแบตเตอรี่นิกเกิล-เมตัลไฮไดรด์ 52
รูปที่ 5.11 การกำเนิดเสียง 53
รูปที่ 5.12 วงจรสร้างเสียง 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

หน้า

รูปที่ 5.13 การทำงานของโปรแกรม 54
รูปที่ 5.14 การออกแบบโปรแกรมส่วนรับข้อมูลจากเครื่องอ่านรหัส 55
รูปที่ 5.15 แผนภาพการตรวจสอบแบตเตอรี่หมด 56
รูปที่ 5.16 การออกแบบโปรแกรมส่วนสร้างเสียง 56
รูปที่ 5.17 แผนภาพแสดงการออกแบบโปรแกรมหลัก 57
รูปที่ 5.18 การเชื่อมต่อของโปรแกรม 57
รูปที่ 5.19 ขั้นตอนของการค้นหา 58
รูปที่ 5.20 ขั้นตอนของการเพิ่มข้อมูล 59
รูปที่ 5.21 ขั้นตอนการเปลี่ยนแปลงสถานที่ 60
รูปที่ 5.22 ผลลัพธ์การออกแบบฐานข้อมูลโดยวิธี ER 61
รูปที่ 5.23 หัวข้อของตารางทั้ง 5 ตาราง 62
รูปที่ 6.1 เครื่องเก็บข้อมูลรหัสแถบ 65
รูปที่ 6.2 การต่อแหล่งจ่ายไฟให้กับเครื่องเก็บข้อมูลรหัสแถบ 66
รูปที่ 6.3 เครื่องเก็บข้อมูลรหัสแถบพร้อมใช้งาน 66
รูปที่ 6.4 การต่อเครื่องเก็บข้อมูลรหัสแถบเข้ากับคอมพิวเตอร์ 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 แนวคิดและเหตุผล

เนื่องจากทางสำนักหอสมุดกลางมีครุภัณฑ์เป็นจำนวนมาก ดังนั้นจึงจำเป็นอย่างยิ่งที่จะต้องมีคอมพิวเตอร์มาใช้ในการจัดเก็บข้อมูล และตรวจสอบครุภัณฑ์แต่ละชิ้น โดยอาศัยการโปรแกรมเชิงวัตถุ (Object - Oriented Programming) มาเป็นโปรแกรมช่วยในการจัดการฐานข้อมูล ก่อให้เกิดความสะดวกในการข้อดูข้อมูลของครุภัณฑ์ รายละเอียดต่างๆ ของครุภัณฑ์ จึงได้มีการพัฒนาระบบการจัดเก็บครุภัณฑ์ โดยใช้รหัสแถบ (Barcode) มาใช้ควบคู่ไปกับรหัสครุภัณฑ์ซึ่งมีอยู่เดิม ทำให้เจ้าหน้าที่ของสำนักหอสมุดกลาง สามารถตรวจสอบครุภัณฑ์เหล่านั้นได้สะดวกมากขึ้น และยังทำให้เกิดความผิดพลาดน้อยลง

1.2 วัตถุประสงค์

1.2.1 ลดความผิดพลาดในการตรวจสอบครุภัณฑ์

1.2.2 เกิดความสะดวกมากยิ่งขึ้น

1.2.3 สามารถทราบรายละเอียดและข้อมูลทางภาพ แล้วยังตรวจสอบสถานที่ตั้งของครุภัณฑ์ได้ด้วย

1.2.4 ศึกษาการวิเคราะห์และออกแบบระบบเชิงวัตถุ

1.2.5 ศึกษาการเขียนโปรแกรมเชิงวัตถุ

1.2.6 ศึกษาการทำงาน การอ่านข้อมูลโดยใช้รหัสแถบ

1.3 ประโยชน์ที่คาดว่าจะได้รับ

1.3.1 สำนักหอสมุดกลาง สามารถนำไปใช้ในงานตรวจสอบครุภัณฑ์ได้จริง

1.3.2 สามารถนำความรู้ที่ศึกษามาประยุกต์ในการเขียนโปรแกรมเชิงวัตถุด้านอื่น ๆ ได้

1.3.3 สามารถนำข้อมูลที่ได้ ไปกำหนดนโยบายและวางแผนเกี่ยวกับการจัดเก็บครุภัณฑ์ในอนาคตต่อไปได้

1.3.4 มีความรู้ ความสามารถ ในการเขียนโปรแกรมด้านเชิงวัตถุมากขึ้น

1.4 ขอบเขตของโครงงาน

1.4.1 ศึกษาการวิเคราะห์และออกแบบระบบเชิงวัตถุ

1.4.2 ศึกษาการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming)

1.4.3 ศึกษาความต้องการระบบครุภัณฑ์ที่สำนักหอสมุดกลาง

1.4.4 ศึกษาการ Interface ระหว่างฐานข้อมูลภาพกับฐานข้อมูลรายละเอียด และ รหัสแถบ

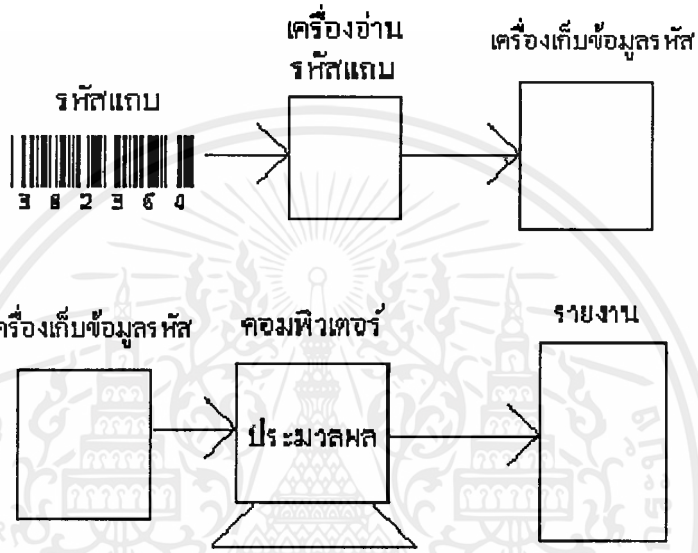
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.5 ลงมือเขียนโปรแกรม

1.4.6 ทดสอบและติดตั้ง

1.4.7 ใช้งานจริง

1.5 ลำดับขั้นการทำงาน



ขั้นตอนการทำงานเราจะอ่านข้อมูลรหัสแถบมาเก็บไว้ที่เครื่องเก็บข้อมูลรหัส หลังจากนั้นจะนำข้อมูลรหัสแถบที่ได้มาทำการประมวลผลยังเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2

การโปรแกรมเชิงวัตถุ

ปัญหาหลักของการพัฒนาโปรแกรมในทุกวันนี้คือ ต้องการเครื่องมือ และเทคนิคที่ดีขึ้นเพื่อใช้ในการสร้างงานที่มีประสิทธิภาพและเร็วในการพัฒนา การพัฒนาซอฟต์แวร์จะทำให้มันมีประสิทธิภาพมากขึ้นและซอฟต์แวร์ที่เราพัฒนานี้จะมีความเชื่อถือได้สูง และในความสำเร็จแต่ละครั้งจะนำไปสู่ความคาดหวังที่ดีขึ้น ๆ ไปอีก

เทคโนโลยีการพัฒนาของการโปรแกรมเชิงวัตถุนั้นเป็นความหวังที่ดีที่สุดที่เราพบตอนนี้ ด้วยเทคโนโลยีแบบออบเจ็กต์โอเรียนเท็ดนี้ เราจะสามารถสร้างออบเจ็กต์ของโปรแกรมไว้เพื่อตอบสนองกับออบเจ็กต์ที่มีใช้งานกันอยู่ได้ เราสามารถสร้างออบเจ็กต์โปรแกรมของมอเตอร์ซึ่งใช้ในการควบคุมตัวมอเตอร์จริงๆ ได้ เราสามารถสร้างออบเจ็กต์ของเวกเตอร์ในการตอบสนองเวกเตอร์ทิศทางที่เคลื่อนที่จริงได้ จากทั้งหมดที่กล่าวมาออบเจ็กต์ซอฟต์แวร์สามารถให้ความเชื่อถือได้สูง และสามารถเรียกใช้ส่วนประกอบทางซอฟต์แวร์ตัวอื่น ๆ ได้อีก ในตอนที่เราจะสร้างโปรแกรมแอปพลิเคชั่นขึ้นมานั้น เราสามารถทำได้โดยการรวบรวมเพิ่ม และปรับปรุงออบเจ็กต์ซอฟต์แวร์ที่มีอยู่เท่านั้นเอง

2.1 ความสำคัญของโปรแกรมเชิงวัตถุ

สาเหตุที่ต้องพัฒนาการโปรแกรมแบบกำหนดวัตถุเป้าหมายหรือโอโอพีขึ้นมา ก็เนื่องมาจากการค้นพบขีดจำกัดในวิธีการโปรแกรมที่มีอยู่ก่อนหน้านี้ การที่จะทำให้เกิดแนวความคิดที่ว่าโอโอพีทำอะไรได้นั้น เราจำเป็นต้องเข้าใจเสียก่อนว่าขีดจำกัดเหล่านี้คืออะไร และเกิดขึ้นมาจากภาษาการโปรแกรมแบบดั้งเดิมอย่างไร

2.2 ภาษาเชิงกระบวนการ

ภาษาปาสคาล ภาษา C ภาษาเบสิก ภาษาฟอร์แทรน และภาษาอื่นๆ เป็นภาษาประเภทเชิงกระบวนคำสั่งหรือภาษากระบวนงาน (procedural languages) นั่นคือ ข้อความสั่ง (statement) แต่ละข้อความสั่งที่ปรากฏอยู่ในภาษา จะบอกคอมพิวเตอร์ให้ทำบางสิ่งบางอย่าง เช่น การรับเข้าบางอย่าง บวกตัวเลข หรือการคูณด้วย 6 แล้วแสดงผลลัพธ์นั้นออกมา โปรแกรมที่อยู่ในภากระบวนการงานจะเป็น รายการคำสั่ง (Instructions)

2.3 การแบ่งออกไปเป็นฟังก์ชัน

เมื่อโปรแกรมเริ่มที่จะมีขนาดใหญ่ขึ้น รายการของคำสั่งแต่เพียงอย่างเดียวก็จะเริ่มมีขนาดเทอะทะ นักโปรแกรมที่สามารถเข้าในโปรแกรมที่มีข้อความสั่งนับร้อยๆนั้นมีจำนวนไม่มากนัก ยกเว้นแต่จะได้แบ่งย่อยโปรแกรมออกไปเป็นหน่วยเล็กๆด้วยเหตุนี้จึงได้ทีการยอมรับหลักการของฟังก์ชัน(Function) มาเป็นแนวทางในการทำให้โปรแกรมเป็นสิ่งที่เข้าใจได้ง่ายขึ้นแก่ผู้เขียนโปรแกรม (คำว่าฟังก์ชันเป็นคำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ใช้อยู่ใน C++ และ C ส่วนในภาษาอื่นๆนั้นอาจเรียกว่าเป็นรูทีนย่อย (Subrutine) หรือโปรแกรมย่อย (Subprogram) หรือกระบวนงาน (Procedure)) เราจะแบ่งย่อยออกไปเป็นฟังก์ชัน ซึ่งแต่ละฟังก์ชันมีจะมีจุดประสงค์ที่ถูกนิยามไว้อย่างชัดเจน (อย่างน้อยที่สุดในอุดมคติ) และมีอินเตอร์เฟซ (Interface) กับฟังก์ชันอื่นในโปรแกรมที่ถูกนิยามไว้อย่างชัดเจนแล้วเช่นกัน

ความคิดเกี่ยวกับการเขียนโปรแกรมออกเป็นฟังก์ชัน สามารถที่จะขยายออกไปให้กว้างขึ้นอีก โดยการจัดกลุ่มฟังก์ชันจำนวนหนึ่งเข้าไว้ด้วยกันไปเป็นเอนทิตี (Entity) ขนาดใหญ่ เรียกว่า โมดูล (Module) แต่หลักการนั้นยังคงเหมือนเดิม คือ เป็นการจัดกลุ่มส่วนประกอบที่ปฏิบัติภารกิจเฉพาะอย่าง

การแบ่งโปรแกรมย่อยออกเป็นฟังก์ชันและโมดูล เป็นพื้นฐานอีกประการหนึ่งของการโปรแกรมเชิงโครงสร้าง (Structured Programming) ซึ่งเป็นสาขาของการเขียนโปรแกรมที่ถูกนิยามไว้อย่างกว้างๆ และการมีอิทธิพลต่อการจัดระบบการโปรแกรมมามากกว่าหนึ่งทศวรรษแล้ว

2.4 ปัญหาที่เกิดขึ้นกับการเขียนโปรแกรมเชิงโครงสร้าง

เมื่อโปรแกรมมีขนาดใหญ่ขึ้นและซับซ้อนเพิ่มมากขึ้น แม้แต่วิธีการโปรแกรมเชิงโครงสร้างเองก็เริ่มที่จะส่อเค้าความยุ่งยากคุณคงเคยได้ยินหรือเคยเข้าไปสัมผัสกับเรื่องเล่าที่น่ากลัวของการพัฒนาโปรแกรมมาบ้างแล้ว เช่นโครงสร้างซับซ้อนจนเกินไป ต้องเลื่อนตารางเวลาออกไป มีการเพิ่มเติมนักโปรแกรมมากขึ้น ค่าใช้จ่ายเพิ่มสูงขึ้น และตามมาด้วยความล้มเหลว เป็นต้น

เมื่อวิเคราะห์หาสาเหตุที่ทำให้เกิดความล้มเหลวนี้ ได้พบว่ามีจุดบกพร่องปรากฏอยู่ในตัวของแบบแผนของกระบวนงานเอง ดังนั้นไม่ว่าจะมีการใช้วิธีการโปรแกรมเชิงโครงสร้างที่ดีเพียงใดก็ตาม แต่โปรแกรมขนาดใหญ่ก็เริ่มที่จะมีความซับซ้อนมากจนเกินไป

ถ้าเช่นนั้นสาเหตุที่ทำให้ภาษากระบวนงานนี้มีความล้มเหลวคืออะไร? เหตุผลที่สำคัญมากที่สุดประการหนึ่งก็คือ บทบาทที่ข้อมูลแสดงออกมา

2.5 แบบแผนของออบเจ็กต์โอเรียนเท็ด

แบบแผนการทำงานเป็นสิ่งที่จำเป็นในการเขียนโปรแกรมซอฟต์แวร์คอมพิวเตอร์ เช่น เราอาจใช้วิธีการจัดการแบบบนลงล่าง หรือแบบโครงสร้างในการแก้ปัญหาโปรแกรมโดยงานหลัก ๆ จะถูกแบ่งออกเป็นงานย่อย และงานย่อยก็จะถูกแบ่งย่อยลงไปอีกเรื่อย ๆ จนกระทั่งงานที่ได้อยู่ในรูปแบบซึ่งง่ายเพียงพอที่จะเขียนโปรแกรมภาาที่ได้เลือกไว้ได้ง่าย ๆ

การพัฒนาโปรแกรมภาษาจากปี ค.ศ. 1940 ถึงปี ค.ศ. 1980 ทำให้สามารถเขียนโปรแกรมภาษาแอสเซมบลีแทนภาษาเครื่องได้ และกลายเป็นโปรแกรมภาษาที่สูงขึ้นเรื่อย ๆ เช่น COBOL , FORTRAN , C และ ADA ภาษาระดับสูงเหล่านี้ต่างทำให้การโปรแกรมง่ายขึ้นแต่มันไม่เปลี่ยนแปลงวิธีพื้นฐานของแบบแผนการโปรแกรม ซึ่งรากฐานของมันจะไม่แตกต่างไปจากภาษาเดิม ภาษาระดับสูงนี้ทำให้การเขียนโปแกรมง่าย และไม่ผูกมัดกับรายละเอียดของภาษาเครื่อง แต่แบบแผนหรือรูปแบบการโปรแกรมจะเหมือนกันตรงที่มีการทำงานแบบเรียงลำดับการทำงานไปเรื่อย ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคโนโลยีแบบออบเจ็กต์โอเรียนเท็ดได้ประยุกต์ เอารูปแบบการโปรแกรมของ ปี ค.ศ. 1990 โดยในครึ่งศตวรรษแรกของการพัฒนาซอฟต์แวร์จะเปลี่ยนพื้นฐาน ของแบบแผนการโปรแกรมเป็นแบบออบเจ็กต์โอเรียนเท็ด ซึ่งเราสามารถที่จะสร้างซอฟต์แวร์ได้โดยการรวบรวมเอาออบเจ็กต์ซอฟต์แวร์ต่าง ๆ เข้าในโปแกรมเดียวกัน ในหลักการของมันคือ มีส่วนปะกอบหลักหลาย ๆ ตัวซึ่งสามารถเรียกไปใช้งานได้หลาย ๆ ครั้ง การโปรแกรมแบบออบเจ็กต์โอเรียนเท็ดมีรูปแบบพื้นฐานของการแยกโปรแกรมออกเป็นส่วนย่อย ๆ โดยการสร้างเป็นออบเจ็กต์แต่ละตัวแยกกันไป ออบเจ็กต์แต่ละตัวจะมองได้เหมือนกับชนิดข้อมูลตัวหนึ่ง เช่น โครงสร้างของภาษา C สามารถที่จะเพิ่มฟังก์ชั่นใหม่ที่สร้างขึ้นมาเองได้ โดยออบเจ็กต์จะเป็นเหมือนกับฟังก์ชั่นเพียงแต่เรามองเฉพาะพฤติกรรมการทำงานของมันเท่านั้น

เทคโนโลยีของออบเจ็กต์โอเรียนเท็ดเป็นส่วนสำคัญ ในขั้นตอนวิวัฒนาการไปตามโครงสร้างแบบบนลงล่าง และภาษาระดับสูงจะเปลี่ยนรูปแบบ และวิธีการพัฒนาซอฟต์แวร์มันจะเปลี่ยนแนวความคิดในการเขียนโปรแกรมใหม่ มันจะเพิ่มจำนวนและชนิดของโปรแกรมแอปพลิเคชั่นที่สามารถติดต่อได้ให้มากขึ้นและจำทำให้การปรับปรุงโปรแกรมสามารถทำได้ง่ายขึ้น

รูปที่ 2.1 วิวัฒนาการของแบบแผนการโปรแกรม

2.6 เทคโนโลยีของการโปรแกรมเชิงวัตถุได้นำประโยชน์ของการพัฒนาซอฟต์แวร์

ความสามารถในการเรียกใช้ได้หลายครั้งออบเจ็กต์ได้ถูกออกแบบตามหลักการที่ว่ามันสามารถเรียกใช้งานได้หลาย ๆ ครั้ง ในหลักการนี้ทำให้โปรแกรมแอปพลิเคชั่นของออบเจ็กต์โอเรียนเท็ดตัวแรกอาจจะทำได้ยากมาก แต่ว่าโปรแกรมแอปพลิเคชั่นที่เขียนภายหลังจะสร้างง่าย เพราะพวกมันสามารถเรียกใช้ออบเจ็กต์ที่ถูกสร้างไว้ตั้งแต่โครงงานแรกได้

ความเชื่อถือได้ โปรแกรมแอปพลิเคชั่นของออบเจ็กต์โอเรียนเท็ด จะมีความเชื่อถือได้สูงเพราะพวกมันจะรวมเอาส่วนย่อยที่ทดสอบจนได้มาตฐานแล้วมารวมเข้าไว้ด้วยกัน รหัสที่เขียนขึ้นมาใหม่ในแต่ละแอปพลิเคชั่นจะมีไม่มากนักเนื่องจากรหัสส่วนใหญ่จะถูกดึงมาจากไลบรารีที่มีความเชื่อถือได้สูงอยู่แล้ว และในการโปรแกมภาษา C++ ยังมีคุณประโยชน์อื่นอีก

ความต่อเนื่องกัน การพัฒนาซอฟต์แวร์แบบออบเจ็กต์โอเรียนเท็ดใน C++ จะเปลี่ยนไปตามฝีมือ และจำนวนนักเขียนโปรแกรมภาษา C และโปรแกรมแอปพลิเคชั่นภาษา C ภาษา C++ ยังไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นพื้นฐานที่เป็นกิจลักษณะที่แน่นอน แต่วัตถุประสงค์ส่วนใหญ่ของ C++ คือเหมาะสำหรับผู้ผลิตหลาย ๆ ราย หรือมาตรฐานที่มีหลายรูปแบบ ความสามารถในการเรียกใช้ได้หลายครั้งและความเชื่อได้ที่เพิ่มเข้าไป ทำให้การโปรแกรมเชิงวัตถุสามารถมุ่งไปสู่ทางสายหลักเดียวกันได้ และจะเกี่ยวข้องไปถึงความต่อเนื่องกันทำให้ภาษา C++ ถูกใช้กันอย่างกว้างขวางในภาษาโปแกรมแบบออบเจ็กต์โอเรียนเท็ด

2.7 ประสิทธิภาพในการทำงาน

จากประวัติที่ผ่านมา นักพัฒนาโปแกรมทางออบเจ็กต์โอเรียนเท็ดได้รับผลกระทบทางด้านลบด้วยคือ ในภาษาออบเจ็กต์โอเรียนเท็ดสมัยเริ่มแรกบางตัว เช่น SIMULA และ LISP เป็นเลื่องลือในความไม่มีประสิทธิภาพของพวกมัน เช่น ภาษาC++ จะเน้นอยู่กับประสิทธิภาพในการทำงานตามคุณลักษะของภาษา C เท่านั้น ภาษา C++ สงวนไว้ซึ่งคุณลักษณะทั้งหมดของภาษา C ซึ่งให้เราเขียนแบบพาดพิงตัวอื่นได้ แต่มันก็ยอมให้เราเสนอคุณลักษณะที่เหมาะสมเองได้ และความสามารถอื่น ๆ ซึ่งมีเพิ่มขึ้นมาใหม่ในตัว C++ เอง ตัวอย่างเช่น INLINE FUNCTION , REFERENCES การทำต้นแบบของโปรแกรมย่อยแบบเต็มรูปแบบ นั่นหมายความว่าตัวแปลภาษา C++ สามารถที่จะกำเนิดรหัสที่มีประสิทธิภาพขึ้นมาได้มากกว่าตัวแปลภาษา C

2.8 วิวัฒนาการของการโปรแกรมเชิงวัตถุ

แนวความคิดของการโปรแกมแบบออบเจ็กต์โอเรียนเท็ดมีตั้งแต่กลางปี ค.ศ. 1960 แต่ไม่เป็นที่ยอมรับจนกระทั่งปลายปี ค.ศ.1980 นั้นนักพัฒนาโปรแกรมเริ่มมีการสังเกต และประเมินแนวคิดเหล่านี้ แต่ก็ยังไม่เป็นที่ยอมรับอีกจนกระทั่งต้นปี ค.ศ.1990 เทคโนโลยีของออบเจ็กต์โอเรียนเท็ดได้เริ่มมีการใช้กันอย่างแพร่หลาย เหตุผลว่าทำไมจึงยอมรับกันช้านักเนื่องจากความจริงที่ว่าส่วนใหญ่ แนวความคิดของการทำงานด้วยออบเจ็กต์โดยพื้นฐานแล้วมันไม่ได้มีการพูดจากันเลย เช่น ถ้าเราพูดกับวิศวกรไฟฟ้าเราจะพบว่าเขาหรือหล่อนทำงานกับอุปกรณ์ต่าง ๆ เช่น ไอซี ตัวต้านทาน ตัวเก็บประจุ และชอบที่จะออกแบบวงจรดิจิตอลต่าง ๆ ในกาพูดคุยวิศวกรไฟฟ้าอีกคนหนึ่ง เราก็จะได้ยินเกี่ยวกับตัวทรานซิสเตอร์ หม้อแปลงไฟ และอุปกรณ์ทางไฟฟ้ากำลังต่างๆ มันเป็นที่ยอมรับกันว่านักออกแบบวงจรดิจิตอบมักจะไม่ไปกังวลเกี่ยวกับแหล่งจ่ายไฟฟ้ามากนัก

จนกระทั่งเร็ว ๆ นี้ นักพัฒนาซอฟต์แวร์หลายคนเริ่มรู้สึกสงสัยว่าเรากำลังคุยกันเกี่ยวกับเรื่องอะไรอยู่ ถ้าเราถามพวกเขาว่าออบเจ็กต์คืออะไรแล้ว เราจะได้คำตอบหลาย ๆ อย่าง โดยนักพัฒนาซอฟต์แวร์ที่คุ้นเคยกับการสร้างทุกสิ่งทุกอย่างขึ้นมาใหม่ ทุกครั้งนั้นจะมีหนังสือจำนวนมากในการทำงาน แต่พวกเขาจะต้องใช้เวลาในการทำงานมากพอสมควร

เหตุการณ์ครั้งล่าสุดของนักพัฒนาซอฟต์แวร์ ก่อนที่จะมีการพบวิธีการพัฒนาโปรแกรมเชิงวัตถุก็คือ มีการพัฒนาและปรับปรุงโครงสร้างของการโปรแกรมใหม่ในช่วงปี ค.ศ.1970 ถึงปี ค.ศ.1980 มีแนวความคิดที่จะแบ่งส่วนโครงสร้างของการโปรแกรมออกเป็นส่วนย่อย เพื่อให้วิธีการทำงานดูง่ายขึ้นและการแปลภาษาให้มีความน่าเชื่อถือสูงขึ้น (อย่างเช่น การตรวจสอบชนิดของข้อมูล การตรวจความต่อเนื่องกันของอาร์เรย์) การโปรแกรมแบบออบเจ็กต์โอเรียนเท็ดไม่ได้ทำตาม หรือยกเลิกการโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างแบบเดิมทิ้ง นักพัฒนาโปรแกรมยังคงต้องการโปรแกรมโครงสร้างเพื่อการพัฒนาซอฟต์แวร์ตามความหมายของออบเจ็กต์โอเรียนเท็ด ความหมายของออบเจ็กต์โอเรียนเท็ดในการโปรแกรมขนาดใหญ่ คือทำอย่างไรให้โปรแกรมมีระบบระเบียบในการทำงานที่ดีบนรหัสกซึ่งสามารถเรียกใช้ซ้ำ ขยายเพิ่มเติมได้ และสามารถจัดการได้ไม่ยาก ปัจจัยต่าง ๆ ที่ได้ในช่วง 2 ทศวรรษที่ผ่านมา มันได้ถูกตีความหมายเป็นการโปรแกรมโครงสร้างแบบใหม่ คอมพิวเตอร์ได้กลายเป็นสิ่งหนึ่งที่จำเป็นในการจัดการระบบต่าง ๆ ซึ่งทำได้ยากและใช้เวลานานหากใช้คนทำอย่างเดียว มีโปรแกรมหลายตัวที่ทำให้โครงงานล้มเหลว หรือล่าช้ากว่ากำหนดได้ หรือมีราคาแพงมาก มีความยากในการใช้งาน ในความหมายนี้การที่จะลดความเสี่ยงเป็นสิ่งที่ต้องอาศัยความตั้งใจในการปฏิบัติงานสูงมาก โดยมากมักจะได้รับอิทธิพลครอบงำกับคำพูดที่ว่า "มีภาษาอะไรที่นำมาใช้กับโครงการทางซอฟต์แวร์นี้ได้บ้าง"

การจะปรับเปลี่ยนแนวคิดของออบเจ็กต์โอเรียนเท็ดได้ภายในวันสองวันนี้นมันทำไม่ได้ และภาษา C++ ก็เป็นภาษาที่สลับซับซ้อนมากกว่าภาษา C มาก แต่ C++ ในตอนนี้ได้มีความอยู่ตัวในตัวมันเองแล้ว และการใช้งาน C++ ในปัจจุบันได้เข้มแข็งมากกว่า C ในเวอร์ชั่นก่อน ๆ ประโยชน์ของการโปรแกรมแบบออบเจ็กต์โอเรียนเท็ดได้มีการแสดงให้เห็นอย่างชัดเจนยิ่งขึ้น โดยเราสามารถที่จะนำเอาวิธีใหม่นี้ไปใช้กับการออกแบบขบวนการทำงานได้ เราสามารถสร้างงานหรือผลิตภัณฑ์ใหม่ให้มีประสิทธิภาพในการทำงานสูง ในระยะเวลาอันสั้น และมีความน่าเชื่อสูงได้ด้วย

2.9 วัตถุ (Object)

วัตถุ คือสิ่งต่าง ๆ รอบตัวเราที่สามารถนำมาทำแบบจำลอง (MODEL) ได้ด้วยโปรแกรมคอมพิวเตอร์ เช่น วงกลม หนังสือ บุคคล ซึ่งอาจมีความหมายในตัวหรือต้องมีวัตถุประสงค์ของการใช้งานประกอบ เพราะออบเจ็กต์จะต้องประกอบด้วยรายละเอียดและกิจกรรม

2.9.1แอททริบิว (Attribute) คือรายละเอียดของออบเจ็กต์ซึ่งได้มาจากวัตถุประสงค์ของการใช้งาน เช่น

วงกลม โดยทั่วไปจะมีวัตถุประสงค์ของการใช้งานเพื่อการแสดงภาพวงกลมบนจอภาพ จึงมีรายละเอียดประกอบด้วย โคออร์ดิเนตของจุดศูนย์กลาง (บนจอภาพ) กับรัศมี

บุคคล ขึ้นอยู่กับวัตถุประสงค์ของการใช้งาน เช่น เพื่อการติดต่อ ก็จะมีรายละเอียดประกอบด้วย ชื่อ นามสกุล ที่อยู่ และหมายเลขโทรศัพท์ แต่ถ้าเป็นบุคคลที่เป็นคนไข้ ก็จะเพิ่มายละเอียดทางด้านการแพทย์ หรือเราจะมองบุคคลตามประเภทก็ได้ เช่น เป็นพนักงาน คนไข้ นักเรียน ฯลฯ และกำหนดเป็นออบเจ็กต์โดยตรง

หนังสือ ในงานห้องสมุดจะมีรายละเอียดเป็น ชื่อหนังสือ ชื่อผู้แต่ง สำนักพิมพ์ ฯลฯ ทางด้านภาษา รายละเอียดคือข้อมูลของออบเจ็กต์เก็บอยู่ในตัวแปรเทียบได้กับฟิลด์หนึ่ง ๆ ของเรคอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 กิจกรรม (Method) ทางด้านโอโอพี คือ กริยาที่ออบเจ็กต์นั้นดำเนินการได้ หรือให้ดำเนินการได้ หรือที่ออบเจ็กต์นั้นรู้จัก แล้วแต่การมอง จึงนิยมตั้งชื่อกิจกรรมให้เป็นกริยา ทางด้านภาษา กิจกรรมคือ โปรแกรมย่อยที่จัดไว้สำหรับดำเนินการกับออบเจ็กต์ เช่น

วงกลม จะมีกิจกรรมเป็นแสดงตัว (วาดภาพวงกลม) หายตัว (ลบภาพวงกลม) และย้ายไปที่ (เปลี่ยนตำแหน่งโคออร์ดิเนตของจุดศูนย์กลาง)

บุคคล ในการติดต่อจะมีกิจกรรมเป็น ให้แจ้งชื่อ ที่อยู่ หมายเลขโทรศัพท์ ให้แก้ชื่อ ที่อยู่ หมายเลขโทรศัพท์

2.10 คลาส (Class)

ในโอโอพีนั้นเรากล่าวว่า วัตถุเป้าหมายเป็นสมาชิกของคลาส (member of classes)ประโยคนี้หมายความว่าอย่างไร ? เพื่อให้เข้าใจมากขึ้นลองมาดูขอ้เปรียบเทียบดังนี้ ภาษาคอมพิวเตอร์เกือบทั้งหมดจะมีชนิดข้อมูลแบบในตัว เช่นในภาษา C++ จะนิยามข้อมูลชนิด int ซึ่งหมายถึงจำนวนเต็ม (integer) ไว้เป็นการล่วงหน้า คุณจะสามารถประกาศตัวแปรให้เป็นชนิด int ได้มากตามจำนวนที่คุณต้องการให้มีอยู่ในโปรแกรม เช่น

```
int day:
int count;
int divisor;
int answer;
```

ในทำนองเดียวกันคุณสามารถนิยามวัตถุเป้าหมาย จำนวนมากให้อยู่ในคลาสเดียวกันได้ แต่ละคลาสจะทำหน้าที่เปรียบเสมือนกับผังเป็นเทมเพลต โดยจะระบุว่าจะรวมเอาข้อมูลอะไรและฟังก์ชันอะไรไว้ในวัตถุเป้าหมายของคลาสนั้น การนิยามคลาสไม่ได้ทำให้เกิดการสร้างวัตถุเป้าหมายใดๆในทำนองเดียวกับการที่มีแต่เพียงชนิด int ก็ไม่ได้ทำให้เกิดการสร้างตัวแปรใด ๆ ขึ้น

ดังนั้นคลาสจึงเป็นกลุ่มของวัตถุเป้าหมายที่คล้ายคลึงกัน ความหมายเช่นนี้ตรงกันกับความเข้าใจเกี่ยวกับคำว่า "คลาส" ในทางที่ไม่ใช่วิชาการ ตัวอย่างเช่น นักร้องชื่อดังเช่น ปรินซ์สติง และมาดอนนา เป็นสมาชิกของคลาสนักร้องเพลงร็อค เราไม่เรียกคนทั่วไปว่าเป็น "นักร้องเพลงร็อค" แต่คนที่มีลักษณะเฉพาะและมีชื่อเฉพาะเท่านั้นที่จะเป็นสมาชิกของคลาสนี้ถ้าเขาเหล่านั้นสามารถครอบครองลักษณะเฉพาะอย่างไว้

2.11 การสืบทอดของคลาส (Inheritance)

ความคิดเกี่ยวกับคลาสได้ก่อให้เกิดความคิดเกี่ยวกับการสืบทอด (Inheritance) ขึ้นในชีวิตประจำวัน เราใช้แนวคิดเกี่ยวกับการแบ่งคลาสย่อย (Subclass) เช่น เราทราบดีว่าจะแบ่งคลาสของสัตว์ออกเป็นสัตว์เลี้ยงลูกด้วยนม สัตว์เลื้อยคลาน แมลง นก ฯลฯ ส่วนคลาสของพาหนะจะแบ่งออกไปเป็นรถยนต์ รถบรรทุก รถประจำทาง และรถจักรยานยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการที่นำมาใช้ในการแบ่งเช่นนี้อยู่ตรงที่ คลาสย่อยแต่ละคลาสย่อยจะมีลักษณะร่วมกันบางประการซึ่งจะเหมือนกันกับคลาสที่เป็นคลาสต้นกำเนิด เช่น รถยนต์ รถบรรทุก รถประจำทาง และรถจักรยานยนต์ ทั้งหมดมีล้อและเครื่องยนต์ ซึ่งลักษณะนี้เป็นลักษณะของพาหนะที่นิยามไว้ นอกเหนือจากลักษณะที่มีร่วมกันของสมาชิกอื่นๆของคลาสแล้ว แต่ละคลาสย่อยยังมีลักษณะเฉพาะบางอย่างที่เป็นของตัวเองอีกด้วย เช่น รถประจำทางจะมีที่นั่งสำหรับผู้คนจำนวนมาก ในขณะที่รถบรรทุกจะมีเนื้อที่สำหรับบรรทุกสัมภาระที่หนักๆ

ความคิดเช่นนี้ได้แสดงไว้ในรูปที่ 1 จากรูปจะพบว่าคุณลักษณะ A และ B ซึ่งเป็นส่วนหนึ่งของคลาสมาตราฐาน จะเป็นลักษณะร่วมของคลาสอนุพัทธ์ทั้งหมด แต่ถึงกระนั้นแต่ละคลาสอนุพัทธ์ก็ยังมีคุณลักษณะที่เป็นของตัวเองอีกด้วย ในทำนองเดียวกัน เราสามารถแบ่งคลาสของโอโอพีออกไปเป็นคลาสย่อย ใน C++ เรียกคลาสต้นกำเนิดว่า คลาสมาตราฐาน (Vase class) จากคลาสมาตราฐานเราสามารถนิยามคลาสอื่นๆที่มีลักษณะเฉพาะของคลาสมาตรฐานร่วมอยู่ หร้อมกับมีการเพิ่มเติมลักษณะของตัวเองร่วมไปด้วยเรียกคลาสที่นิยามขึ้นมาใหม่นี้ว่า คลาสอนุพัทธ์ (Devired classes)

อย่าสับสนระหว่างความสัมพันธ์ของวัตถุเป้าหมายกับคลาสในด้านหนึ่ง กับความสัมพันธ์ของคลาสฐานกับคลาสอนุพัทธ์ในอีกด้านหนึ่ง วัตถุเป้าหมายซึ่งปรากฏอยู่ในหน่วยความจำของคอมพิวเตอร์จะรวบรวมเอาคลาสลักษณะเฉพาะที่ชัดเจนเอาไว้ ซึ่งคลาสจะเปรียบเสมือนได้กับเป็นเทมเพลต ส่วนคลาสอนุพัทธ์จะสืบทอดลักษณะเฉพาะบางอย่างจากคลาสมาตราฐาน แต่มีการเพิ่มเติมลักษณะใหม่ของตัวมันเองเข้าไปด้วย

การสืบทอดที่ค่อนข้างจะคล้ายคลึงกับการใช้ฟังก์ชันเพื่อทำงานโปรแกรมกระบวนงานแบบดั้งเดิมให้ง่ายขึ้น ถ้าเราพบว่าส่วน (Section)ของโปรแกรมกระบวนงานที่แตกต่างกันสามส่วน ทำในสิ่งเดียวกันเกือบจะชัดแจ้ง เราจะรู้ถึงโอกาสที่จะสกัดส่วนประกอบร่วมทั้งสามส่วนนี้ และนำเอาส่วนประกอบนี้เข้าไปไว้ในฟังก์ชันเดี่ยวๆทั้งสามส่วนของโปรแกรมสามารถเรียกฟังก์ชันนั้นเพื่อทำการดำเนินงานที่เป็นลักษณะการดำเนินงานร่วม และส่วนเหล่านี้ยังสามารถทำการประมวลที่เป็นของตัวเองได้อีกด้วย ในทำนองเดียวกัน คลาสฐานจะประกอบไปด้วยส่วนประกอบที่พบร่วมกันอยู่ในกลุ่มของคลาสอนุพัทธ์ การสืบทอดจะช่วยทำให้โปรแกรมแบบกำหนดวัตถุเป้าหมายมีขนาดสั้นลงและทำให้มีสัมพันธภาพในหมู่ส่วนประกอบของโปรแกรมกระจ่างชัดขึ้น ในลักษณะเดียวกันกับฟังก์ชันกระทำในโปรแกรมกระบวนงาน

ในกรณีที่โปรแกรมมีขนาดเล็กมากๆ ก็ไม่จำเป็นต้องอาศัยหลักการจัดระบบอื่นๆ(Paradigm) นักโปรแกรมหรือโปรแกรมเมอร์ (Programmer) จะสร้างรายการของคำสั่งขึ้นและคอมพิวเตอร์ก็จะทำตามรายการคำสั่งนั้น ความมุ่งหมายของการอินเฮอริตที่ให้สืบสกุลได้คือ เพื่อให้ปรับปรุง (แก้และต่อเติม) โปรแกรมได้โดยไม่ต้องกระทำในซอร์สโค้ดเดิม กล่าวคือ

2.11.1 การแก้ไข สามารถทำได้ดังนี้

- จากสกุลบุคคลคือ PERSON ได้กำหนดตัวแปรสำหรับเก็บหมายเลขโทรศัพท์ไว้เพียง 7 ตัวอักษร และเขียนกิจกรรม GET PHONE ไว้เพื่ออ่านหมายเลขโทรศัพท์ กับ SET PHONE เพื่อให้และแก้หมายเลขโทรศัพท์ ต่อมาต้องการให้เก็บหมายเลขโทรศัพท์ได้ 9 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การอินเฮอริดทำให้ไม่ต้องแก้ที่กซอร์สโค้ดเดิม ทำได้โดยการสร้างสกุลใหม่สมมติว่าใช้ชื่อว่า NEW PERSON ให้เป็นผู้สืบสกุลของสกุล PERSON กำหนดตัวแปรสำหรับหมายเลขโทรศัพท์ใหม่ กับกำหนดและเขียนโปรแกรมย่อย GET PHONG กับ SET PHONE ใหม่ ซึ่งจะเป็นการกำหนดชื่อทับของเดิม

- เมื่อดำเนินการเช่นนี้ จะทำให้สกุลใหม่นี้สามารถใช้ตัวแปรและกิจกรรมของสกุล PERSON ได้ เช่น สามารถอ่านและแก้ชื่อกับที่อยู่ได้ (โดยไม่ต้องเขียนโปรแกรมใหม่) แต่ในการอ่านและแก้หมายเลขโทรศัพท์จะใช้ของตนเอง

2.11.2 การต่อเติม สามารถทำได้ดังนี้

- จากสกุล PERSON (หรืออาจจากสกุล NEW PERSON ได้เช่นกัน) สมมติว่าต้องการรายละเอียดเป็นวันเกิดเพิ่ม ก็อาจสร้างเป็นสกุล VIP PERSON ให้เป็นผู้สืบสกุลของ PERSON (หรือ NEW PERSON) กำหนดตัวแปรสำหรับเก็บวันเกิดและเขียนกิจกรรม GET BIRTHDAY กับ SET BIRTHDAY เพื่ออ่านและให้ค่าวันเกิดตามลำดับ ก็จะเป็นการต่อเติม เพราะสามารถใช้ตัวแปรและกิจกรรมในสกุลบรรพบุรุษได้ทั้งหมด ซึ่งถือว่าเป็นไปตามหลักการของอินเฮอริต

การดำเนินการดังกล่าวมานี้ ไม่ว่าจะเป็นการแก้หรือต่อเติม จะต้องแก้ในโปรแกรมใช้งาน คือเปลี่ยนการกำหนดอินสแตนซ์จากของสกุลเดิมเป็นของสกุลใหม่ ไม่เช่นนั้นก็เท่ากับไม่ได้แก้หรือต่อเติม ซึ่งเป็นข้อดีอีกประการหนึ่งของการอินเฮอริตเพราะในงานบางงานอาจต้องการให้ของเดิมคงเป็นไปตามเดิม

สำหรับสกุลบรรพบุรุษที่มีผู้สืบสกุลไม่ว่ากี่ชั้นและตนเองไม่ได้เป็นผู้สืบสกุลของสกุลใด เช่น PERSON จะเรียกเป็นต้นสกุลหรือต้นตระกูลตามความเหมาะสม

ทางด้านภาษาดงให้มีการดำเนินการดังกล่าวมา แตกต่างกันบ้างในข้อจำกัดที่จะปกป้องตัวแปรและกิจกรรมคือ มีทั้งที่เข้มงวดและไม่เข้มงวด หากเป็นไปตามหลักการทางด้านโอโอพีตามที่ปรากฏในภาษา SMALTALK จะเข้มงวดมาก ผู้สืบสกุลจะเข้าถึงตัวแปรในบรรพบุรุษไม่ได้เลยทั้งการอ่านและเขียน หากผู้เขียนโปรแกรมบรรพบุรุษต้องการให้เข้าถึงตัวแปรใดได้จะต้องเตรียมกิจกรรมไว้ให้สำหรับกิจกรรมตามแต่ผู้เขียนโปรแกรมบรรพบุรุษจะเป็นผู้กำหนด ส่วน C++ ให้ผู้เขียนโปรแกรมบรรพบุรุษเป็นผู้กำหนดว่าจะปกป้องตัวแปรและกิจกรรมใดบ้าง สำหรับในโปรแกรมเทอร์โบปาสคาลมีทั้งที่ปกป้องได้กับปกป้องไม่ได้

2.11.3 เอนแคปซูลเลชั่น (Encapsulation) คือความเป็นเอกเทศ หมายถึง ให้สามารถรวมรายละเอียดกับกิจกรรมไว้ในที่แห่งเดียวกัน และให้อยู่ในที่ปกป้องอันหนึ่ง ซึ่งทางด้านภาษาให้ทำสกุลเป็นแบบข้อมูลแบบหนึ่งคล้ายกับ RECORD ในภาษาปาสคาล ที่มีฟิลด์ตัวแปรสำหรับเก็บรายละเอียดจำนวนหนึ่งกับฟิลด์กิจกรรมอีกจำนวนหนึ่ง สำหรับเทอร์โบปาสคาลกำหนดให้เป็นข้อมูลแบบเชิงวัตถุ

ที่กล่าวว่าให้อยู่ในที่ปกป้องอันหนึ่งคือ ไม่ให้โปรแกรมส่วนอื่นเข้าถึงรายละเอียดหรือกิจกรรมได้โดยตรง อย่างน้อยที่สุดให้ได้รับการปกป้องในระดับเรคอรด์ คือต้องดำเนินการผ่านอินสแตนซ์ของสกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเอนแคปซูลมีจุดมุ่งหมายหลายประการคือ ทางด้านโอโอพี ถือว่าเป็นการสร้างแบบจำลองของออบเจ็กต์ ทางด้านภาษาก็คล้ายกันคือ มองในแง่ความสมบูรณ์ของโปรแกรมที่ไม่ต้องกำหนดตัวแปรหรือเขียนโปรแกรมเพิ่มอีกเลยสำหรับออบเจ็กต์นั้น เป็นการทำงานเพียงครั้งเดียวที่ใช้ได้ตลอดไป อีกประเด็นหนึ่งคือ ตัวแปรที่ใช้เก็บรายละเอียดหากไม่ทำเป็นสกุล คือไม่เอนแคปซูลก็จะต้องกำหนดเป็นโกลบอล (Global) ซึ่งผิดหลักการเขียนโปรแกรมโดยทั่วไป เมื่อเอนแคปซูลแล้วก็จะเป็นโกลบอลอยู่ภายในสกุลนั้น ทำให้ได้ใช้ข้อดีของโกลบอลที่ไม่ต้องผ่านค่าตัวแปรซึ่งการผ่านค่าตัวแปรทำให้เสียเวลาในการรัน

2.11.4 โพลิมอร์ฟิซึม (Polymorphism) คำว่า POLYMORPHISM มาจากภาษากรีก หมายถึง MANY SHAPES ในที่นี้หมายถึงการเลือกปฏิบัติ เช่น ในสกุลจุดคือ POINT มีกิจกรรม MOVE TO เพื่อย้ายตำแหน่งที่โคออร์ดเนตของจุดจากจุดหนึ่งไปยังอีกจุดหนึ่ง โดยเรียกใช้ HIDE เพื่อลบภาพจุดเดิม และ SHOW เพื่อแสดงภาพจุด ณ ตำแหน่งใหม่

ต่อมาต้องการกำหนดสกุลสี่เหลี่ยมคือ RECTANGLE ให้ทำงานเช่นเดียวกับ POINT และผู้เขียนโปรแกรมต้องการใช้ MOVE TO ของ POINT หากในโอโอพีมีแต่อินเฮอริตจะเรียกใช้ไม่ได้ เพราะ MOVE TO จะเรียกใช้ HIDE และ SHOW ใน POINT

ทางโอโอพีจึงกำหนดให้มีโพลิมอร์ฟิซึม คือ ให้กำหนดกิจกรรมในสกุลใหม่ มีชื่อซ้ำกับสกุลบรรพบุรุษ (ขอให้สังเกตว่าใช้คำว่าซ้ำแทนทับ เพื่อให้ความหมายแตกต่างกัน) ซึ่งทางด้านภาษาให้กำหนดเป็นกิจกรรมที่เป็นเวอร์ชวล (VIRTUAL ยังไม่ต้องทำความเข้าใจในขณะนี้)

ดังเช่น ตัวอย่างที่ยกมา จำกำหนดสกุล RECTANGLE ให้เป็นผู้สืบสกุลของ POINT โดยใช้จุดโคออร์ดิเนตในสกุลนี้เป็นจุดมุมบนซ้ายของสี่เหลี่ยม และกำหนดตัวแปรสำหรับเก็บความกว้างและความสูงของสี่เหลี่ยม กับกำหนดและเขียนกิจกรรม SHOW และ HIDE เป็นเวอร์ชวลเพื่อวาดและลบภาพสี่เหลี่ยมตามลำดับ

ในการใช้งานเมื่อเราอ้างถึง MOVE TO ในสกุล RECTANGLE คือส่งสื่อความ MOVE TO ให้แก่อินสแตนซ์ของสกุล RECTANGLE จะเป็นการเรียกใช้ MOVE TO ใน POINT แต่จะเป็น SHOW และ HIDE ใน RECTANGLE เพื่อการลบและวาดภาพสี่เหลี่ยม แต่ถ้าเราส่งสื่อความ MOVE TO ให้แก่อินสแตนซ์ของสกุล POINT จะเป็นการเรียกใช้ HIDE และ SHOW ในPOINT เพื่อลบและวาดภาพวงกลมเป็นการเลือกปฏิบัติ

2.12 การออกแบบเชิงวัตถุ

การออกแบบด้วยออบเจ็กต์โอเรียนเท็ด เป็นเรื่องของการกำหนดชุดของคลาสและการนำไปใช้แก้โจทย์ของปัญหา ในบทนี้เราจะดูหลักการพื้นฐานของการออกแบบโปรแกรมออบเจ็กต์โอเรียนเท็ด เราจะนำแนวความคิดจากบทที่ผ่านมา เรื่อง ออบเจ็กต์ คลาส Encapsulation, Abstraction, Inheritance และ Polymorphism มาใช้ และดูขบวนการออกแบบด้วยวิธีออบเจ็กต์โอเรียนเท็ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบด้วยออบเจ็กต์โอเรียนเท็ด เป็นหัวข้อหลักสำหรับหนังสือเล่มนี้ ในบทนี้เราจะทำการสร้างพื้นฐานให้กับตัวเรา เพื่อให้เราสามารถใช้งาน C++ และสามารถใช้งานวิธีของออบเจ็กต์โอเรียนเท็ดได้

เราจะเรียนวิธีการออกแบบด้วยออบเจ็กต์โอเรียนเท็ดที่ถูกวิธีจากแบบฝึกหัด เราไม่ควรจะข้ามขั้นตอนบางส่วนของหนังสือ เราควรที่จะยอมรับความเป็นจริงที่ว่า ห้องครัวที่มีอุปกรณ์ครบครันไม่ได้ทำให้เราเป็นพ่อ (แม่) ครัวที่ดีได้ และการที่เรารู้หลักการของอ็อบเจ็กต์โอเรียนเท็ดก็ไม่จำเป็นที่เราจะเป็นนักออกแบบชั้นยอดได้ มันเพียงแต่ทำให้เราออกแบบได้ง่ายขึ้นเท่านั้น

2.13 เอกลักษณ์ของคลาส

อย่าคาดหวังความสะดวกสบายว่าการออกแบบด้วยออบเจ็กต์โอเรียนเท็ด จะสามารถเรียนรู้ได้ในระยะเวลาอันสั้น ถ้าเรามีประสบการณ์มานานกับวิธีการออกแบบโปรแกรมชนิดแบ่งงานออกเป็นโปรแกรมย่อย แล้วมันจะเป็นพื้นฐานที่ดีสำหรับการสร้างคลาส

ลักษณะของการออกแบบคลาสนั้นจะเน้นที่มันสามารถเชื่อมโยงและทำงานสัมพันธ์กันระหว่างคลาสได้ ในการทำงานกับปัญหาที่มีความซับซ้อนมาก ๆ นั้น เราควรที่จะแบ่งงานออกเป็นกลุ่มของคลาสย่อยที่ทำงานสัมพันธ์กัน

- ทำคลาสให้สามารถแสดงแนวความคิดของการทำงานได้เด่นชัด โดยที่เราสามารถบรรยายให้กับคนที่ไม่ใช่นักโปรแกรมฟังรู้เรื่องได้วัตถุ (ออบเจ็กต์) ในโลกแห่งความเป็นจริงแล้วจะมีหน้าที่การทำงานซึ่งไม่เด่นชัด หรือคลุมเครือ ทำให้เราไม่สามารถสรุปหน้าที่ของพวกมันได้อย่างถูกต้องแน่นอน แต่ออบเจ็กต์ทางซอฟต์แวร์จะมีคุณสมบัติที่ทำให้ได้เด่นชัดถ้าเหตุผลในการสร้างคลาสของเรามีมาก และยุ่งยากแล้วให้เราพยายามแยกหัวข้อเหตุผลตามแนวความคิดของเราให้ออกเป็นส่วนย่อย เพื่อง่ายแก่การทำความเข้าใจ และออกแบบ
- ทำคลาสให้มีขนาดเล็กและง่ายต่อการทำความเข้าใจ คลาสแต่ละตัวควรจะอธิบายการทำงานให้เข้าใจได้ง่าย แต่มันยังคงมีความซับซ้อนในการทำงานอยู่มาก โดยมากแล้วคลาสจะมีขนาดใหญ่
- ขึ้นตามนวนครั้งในการปรับปรุง เช่น วันนี้คลาสอาจมีขนาด 10 หน้ากระดาษ แต่ในปีหน้าอาจมีขนาด 50 หน้า กระดาษก็ได้ทำให้การอ่าน และทำความเข้าใจยากขึ้นเรื่อย ๆ เพื่อลดปัญหาตรงนี้ เราควรจะให้คลาสแต่ละตัวมีขนาดไม่มากนัก และทำความเข้าใจได้ง่าย (ถ้ามีขบวนการทำงานเพิ่มอาจจะแบ่งงานออกเป็นคลาสย่อยออกไปอีกก็ได้)
- ทำคลาสให้ปรับเปลี่ยนได้ตามแนวความคิด ชนิดของข้อมูลหรือโครงสร้าง (Structure) โดยทั่วไปจะถูกเลือกว่าให้มัน "เป็น" อะไร แต่คลาสจะเป็นชนิดของข้อมูลหรือโครงสร้างที่เสริมการทำงานให้กับเรา จึงมักเรียกมันว่าควรจะ "ทำ" อะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้คุณสมบัติ Encapsulation เพื่อให้คลาสมีความน่าเชื่อถือสูงขึ้น โดยรหัสจะเรียกใช้งานซ้ำได้อีก ก็ต่อมาเมื่อคลาสนั้นมีคุณสมบัติที่สามารถนำข้อมูลจากสภาวะแวดล้อมหนึ่ง ไปใช้ยังอีกสภาวะแวดล้อมหนึ่งได้

ใช้คุณสมบัติการถ่ายทอด (Inheritance) สร้างตระกูลของคลาสต่าง ๆ ให้สามารถใช้ข้อดีของคุณสมบัติ polymorphism โครงสร้างข้อมูลที่สืบทอดมาจะอนุญาตให้ใช้เฉพาะสมาชิกที่เกี่ยวข้องกันเท่านั้น โดยมันไม่สามารถใช้ในการสร้างตระกูลของโครงสร้างข้อมูลทั่วไปได้ การสร้างคลาสจะทำได้เฉพาะที่มันมีความเกี่ยวข้องกันเท่านั้น โดยคลาสที่สร้างนั้นจะนำมาทำเป็นตระกูลของคลาสอีกทีหนึ่ง

พิจารณาวิธีการออกแบบครั้งแรกนี้และจำไว้เป็นต้นแบบ เพื่อนำไปใช้งาน เราต้องฝึกฝนการออกแบบในขั้นต้นก่อนที่เราจะไปออกแบบงานจริง

การฝึกฝนทำแบบฝึกหัดมาก ๆ เป็นวิธีที่ดีที่สุดในการเรียนรู้เรื่องคลาสให้มีประสิทธิภาพ ถ้าเราทำงานกับปัญหาที่ต้องใช้เวลามากแล้ว เรามักจะมองไปที่ต้นเหตุหรือปัจจัยหลักของปัญหาและหาวิธีแก้ปัญหาให้ตรงจุดทางออก ซึ่งเราต้องการคือโปรแกรมซึ่งสามารถจัดการงานได้ ไม่ว่าจะเป็นงานชนิดใดก็ตาม รหัสซึ่งเราคัดลอกจากงานหนึ่งไปใช้ในอีกงานหนึ่งนั้นเรียกได้ว่าเป็นผู้สมัครที่ดี (รหัสที่เข้ากันได้กับทุก ๆ งาน) สำหรับคลาสโครงสร้างข้อมูลซึ่งเรานำไปใช้ได้ในหลายงานนั้นก็เรียกได้ว่ามันเป็นผู้สมัครที่ดีสำหรับการขยายงานเป็นคลาสต่าง ๆ ถ้าเรามีคลาสที่มีศักยภาพในการใช้งานสูงแล้วเราจะรู้สึกเหมือนกับว่างานที่จะทำนั้นไม่ยาก

โดยปกติแล้ว เรามักจะพบปัญหาที่มองได้หลายแง่มุม เช่น มีผู้สมัครของคลาสพื้นฐานหลายคนได้มาจากคลาสอื่นที่สร้างขึ้นมาทีหลัง ตะปู ควง เป็นวัตถุที่สร้างจากโลหะ หรือว่าเป็นวัตถุที่ใช้ในการสร้างเครื่องจักรในอุตสาหกรรม ตะปูควงจะผลิตได้มาจากโลหะดิบ แต่ในอุตสาหกรรมอีกชนิดหนึ่งตะปูควบจะนำไปใช้ในการประกอบชิ้นส่วนของเครื่องใช้สำนักงานก็ได้

2.14 การออกแบบจากบนลงล่าง และการออกแบบจากล่างขึ้นบน

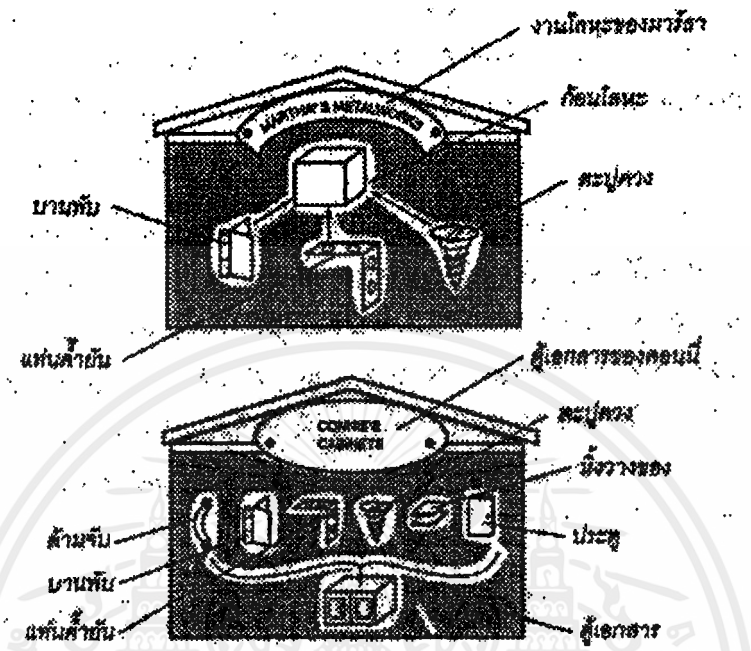
ในการออกแบบจากบนลงล่างนั้น ให้เรามองและทำความเข้าใจปัญหาในวงกว้างเพื่อหาโครงสร้างของการแก้ปัญหาโดยรวม แล้วค่อยแบ่งปัญหาออกเป็นส่วนย่อย ๆ และวิเคราะห์หาข้อปลีกย่อยลงไปอีกเรื่อย ๆ จะเห็นได้ว่า วิธีนี้ปลายทางของปัญหาจะถูกขยายออก

วัตถุแต่ละตัวเช่น ตะปูควง สามารถมองได้หลายแง่มุม ถ้าเป็นอุตสาหกรรมโลหะแล้ว ตะปูควงจะเป็นผลิตภัณฑ์ชนิดหนึ่งที่ได้จากการหลอมโลหะดิบ แต่ถ้าเป็นร้านเฟอร์นิเจอร์แล้ว ตะปูควงจะเป็นส่วนประกอบอย่างหนึ่งที่ใช้ในการทำตู้เอกสาร

ในการออกแบบจากบนลงล่างนี้ มีวิธีการทำง่าย ๆ เพียงแค่แยกแยะปัญหาในแต่ละระดับของโครงสร้างแบบหว่านแหดูภาพโดยรวมทั้งหมด เราจะยังไม่ไปสนใจในรายละเอียดของแต่ละตัวในระดับล่าง แต่วิธีนี้ไม่ได้ทำให้การแก้ปัญหาง่ายลงเสมอไปนัก นักออกแบบที่ดีควรจะมองปัญหาโดยรวมและวิเคราะห์เอาเฉพาะปัญหาที่เราต้องการแก้เท่านั้น ตัวที่ไม่เกี่ยวข้องก็ไม่ต้องไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนใจตรงกันข้ามกับการออกแบบจากล่างขึ้นบน วิธีนี้เราจะมองและแก้ไขตั้งแต่ปัญหาพื้นฐานต่าง ๆ ทั้งหมดแล้ว รวมปัญหาต่าง ๆ ที่เกี่ยวข้องกันให้อยู่ในพวกเดียวกัน ซึ่งมองเป็นระดับบนขึ้นไปเรื่อย ๆ จนกระทั่งครบทุกปัญหา



รูปที่ 2.2 การออกแบบจากบนลงล่างและจากล่างขึ้นบน

ในการออกแบบโปรแกรมจากบนลงล่างที่สมบูรณ์นั้นจะต้องทำให้ปัญหาที่ยาก และท้าทายสติปัญญาถูกแบ่งออกเป็นอนุกรมของปัญหาทางเทคนิคย่อย ๆ โดยที่แต่ละปัญหาทางเทคนิคพวกนี้สามารถแก้ไขได้ด้วยนักเขียนโปรแกรมดี ๆ ตั้งแต่หนึ่งคนขึ้นไปได้ แต่การออกแบบที่สมบูรณ์จะไม่

มีกฎเกณฑ์แน่นอนตายตัว เพราะว่าปัญหาต่าง ๆ จะมีความยากง่ายแตกต่างกันไป ในการออกแบบจากบนลงล่างนี้ให้เราพยายามแยกปัญหาที่ไม่เกี่ยวข้องกันออกจากกันไปเรื่อย ๆ ทำให้ปัญหาที่ยากมากถูกแบ่งออกเป็นปัญหาย่อยที่ยากน้อยลงไปเรื่อย ๆ

การออกแบบชนิดออบเจ็กต์โอเรียนเท็ดจะไม่ใช่ทั้งสองวิธีนี้ โดยมันจะมีคุณสมบัติของทั้งสองวิธีอยู่ในตัว เช่น ในการออกแบบชนิดออบเจ็กต์โอเรียนเท็ดจะแก้ปัญหาหลักด้วยการแบ่งออกเป็นส่วนพื้นฐานต่าง ๆ โดยแบ่งส่วนพื้นฐานต่าง ๆ ออกเป็นออบเจ็กต์ย่อยหลาย ๆ ตัว ซึ่งวิธีนี้เป็นรูปแบบของการออกแบบจากบนลงล่าง และถ้ามองเป็นการออกแบบจากล่างขึ้นบนก็จะได้ว่าการเรียกใช้ออบเจ็กต์ซอฟต์แวร์แต่ละตัวเป็นการแก้ปัญหาในส่วนย่อย เพื่อให้ได้งานโดยรวมที่มีประสิทธิภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่3
ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวเอ็มซีเอส 51

3.1 โครงสร้างของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวเอ็มซีเอส 51

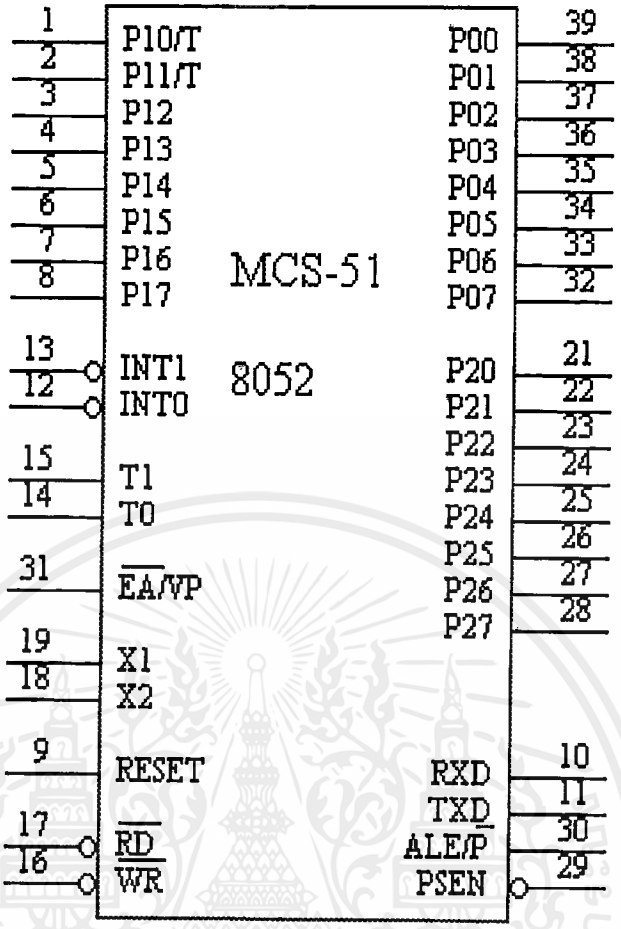
เอ็มซีเอส 51 นี้เป็นอุปกรณ์ที่ออกแบบมาสนอง ตามความต้องการของผู้ใช้ คือ มีสายอินพุต และเอาต์พุตภายในตัวเอง พอร์ตของอินพุตและเอาต์พุตบัฟเฟอร์อินเตอร์เฟส และสายควบคุมอื่น ๆ ที่ใช้สำหรับแยกข้อมูลกับแอดเดรสและยังมีชุดคำสั่งเพิ่มขึ้นเป็นพิเศษเพื่อจัดการข้อมูล แถมท้ายด้วยวงจรตั้งเวลากับวงจรนับด้วย (ปกติวงจรนับจะสามารถทำงานเป็นวงจรตั้งเวลาได้ด้วย จึงเรียกควบคู่กันไปคือ วงจรตั้งเวลา / วงจรนับ)

ตารางที่ 3.1 เป็นรายการของตระกูลเอ็มซีเอส 51 ซึ่งแสดงถึงจำนวนของหน่วยความจำ วงจรตั้งเวลา / วงจรนับและระดับของการขออินเตอร์รัพต์

เอ็มซีเอส 51 ไมโครคอนโทรลเลอร์ประกอบด้วยคุณสมบัติ

- ซีพียู 8 บิต ที่ควบคุมได้ง่าย
- เพิ่มการทำงานลอจิกครั้งละ 1 บิตได้
- สายอินพุตและเอาต์พุตมีจำนวน 32 เส้น ใช้เลือกแอดเดรสแยกต่างหากจาก กันได้
- มีแรมบรรจุไว้ภายในขนาด 128 ไบต์หรือ 256 ไบต์
- วงจรตั้งเวลา / วงจรนับมีขนาด 16 บิต 2 หรือ 3 ชุด
- กำหนดเป็น UART (Universal Synchronous Asynchronous Receiver Transmitter) ที่รับส่งข้อมูลอนุกรมได้สองทิศทาง
- อินเตอร์รัพต์ แบ่งออกเป็น 2 ระดับ จาก 5 หรือ 6 แหล่ง
- มีสัญญาณนาฬิกาอยู่ภายในตัว
- มีหน่วยความจำสำหรับเก็บข้อมูลภายในขนาด 4 หรือ 8 กิโลไบต์ (อีพรอพ 8751 และ 8752)
- มีแอดเดรสของหน่วยความจำสำหรับเก็บโปรแกรม จำนวนทั้งหมด 64 กิโลไบต์
- มีแอดเดรสของหน่วยความจำสำหรับเก็บข้อมูล จำนวนทั้งหมด 64 กิโลไบต์
- คำสั่งทั้งหมดมี 111 คำสั่ง (64 รอบ)
- ทำงานด้วยเลขฐานสิบ และฐานหก
- ตัวแปลภาษาเบสิกมีขนาด 8 กิโลไบต์ (8025 AH-BASIC)
- มีคำสั่งเฉพาะของภาษาเบสิกที่ใช้สำหรับอินพุตและเอาต์พุต วงจรนับ และอินเตอร์เฟสแบบอนุกรม (8025 AH-BASIC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 การจัดวางขาต่าง ๆ ของเอ็มซีเอส 51

จากรูปที่ 3.1 นั้นแสดงจำนวนขา และหน้าที่ของ 8051 ที่หมายถึงตระกูลเอ็มซีเอส 51 ทั้งหมด

8051 , 8051AH และ 8052AH เป็นอุปกรณ์ที่ทำงานโดยการควบคุมจากโปรแกรมในหน่วยความจำ (รอม) โดยการโปรแกรมด้วยต้นแบบ เพื่อให้เกิดข้อมูลบนรอมครั้งหนึ่งเป็นจำนวนมากจึงเหมาะกับการผลิตเพื่อใช้งานควบคุมจำนวนมาก ๆ (8051 นั้นมีหน่วยความจำรอบจำนวน 4 กิโลไบต์ ส่วน 8052AH จะมีหน่วยความจำเพิ่มขึ้นโดยสร้างจาก HMOS Iix

มีอีกสองเบอร์ที่ทำงานคล้ายกันคือ 8031AH และ 8032AH ที่ใช้แทน 8051 และ8052 AH ได้ตามลำดับ โดยไม่ต้องส่งให้โรงงานโปรแกรมให้ เพราะเราจะเขียนและทดสอบโปรแกรมด้วยหน่วยความจำภายนอกแทน (ไม่มีรอมภายใน)

สำหรับท่านที่ต้องการป้องกันการอ่านจากภายนอก ก็สามารถเลือก 8751 และ 8752 ที่มีอีพรอมภายในที่จัดค่าเวลาในการอ่านข้อมูล เพื่อกันการลอกเลียนแบบโปรแกรม

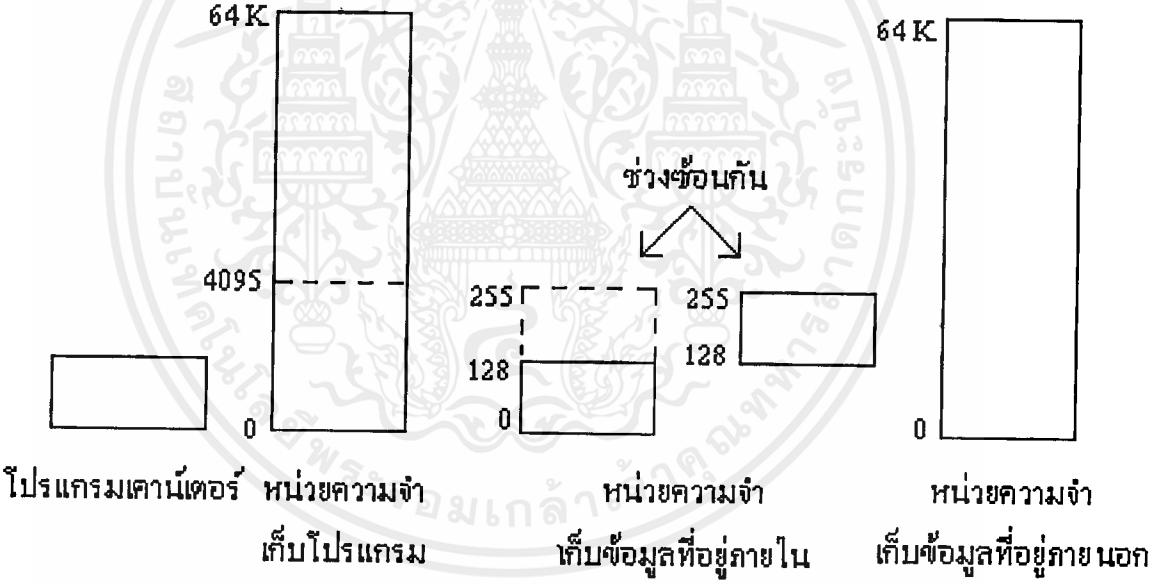
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 หน่วยความจำ

จากรูปที่ 3.2 เป็นหน่วยความจำภายในตัว เอ็มซีเอส 51 หน่วยความจำนี้แบ่งได้ 2 กลุ่ม คือ หน่วยความจำสำหรับเก็บโปรแกรม และหน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำแรกมีแอดเดรสที่ต่ำกว่า 4 หรือ 6 กิโลไบต์ บรรจุอยู่ในรอบ ส่วนเอ็มซีเอส 51 ที่ไม่มีรอบภายในจะใช้หน่วยความจำภายนอกซึ่งอาจเป็นรอบ , แรม , หรืออีพรอมแทนก็ได้

เอ็มซีเอสจะอ่านหน่วยความจำสำหรับเก็บโปรแกรม เข้ามาเป็นภาษาเครื่องตามลำดับ ส่วนหน่วยความจำสำหรับเก็บข้อมูล จะใช้เป็นที่เก็บตัวแปร , การคำนวณหาผลลัพธ์ทันที , การจัดการกับข้อมูลที่มีขนาด 16 บิต (Word) , ตารางที่ใช้ค้นหาค่าต่าง ๆ และหน้าที่อื่นที่คล้ายกัน

หน่วยความจำสำหรับเก็บข้อมูลใช้ร่วมกับหน่วยความจำภายนอกได้ 640 กิโลไบต์ ซึ่งเลือกใช้รอม หรือแรมก็ได้ และยังมีรีจิสเตอร์พิเศษที่ใช้หน่วยความจำภายนอกของแรมได้ 128 หรือ 256 กิโลไบต์



รูปที่ 3.2 โครงสร้างภายในหน่วยความจำภายในเอ็มซีเอส 51

3.3 รีจิสเตอร์ภายในเอ็มซีเอส

เอ็มซีเอสมีรีจิสเตอร์ที่อำนวยความสะดวกในการใช้งานตามคำสั่งต่าง ๆ ประกอบด้วยแอคคิวมูเลเตอร์ รีจิสเตอร์ B ที่ใช้ในการคูณและหาร รีจิสเตอร์สถานะ สแต็กพอยน์เตอร์ ดาต้าพอยน์เตอร์ (2 x 8 บิด หรือ 1 x 16 บิด) พอร์ตหมายเลขศูนย์ถึงพอร์ตหมายเลขสามรีจิสเตอร์แบบคู่ ซึ่งใช้ส่งและรับข้อมูลชนิดอนุกรม รีจิสเตอร์ 16 บิด ที่เป็นวงจรตั้งเวลาและวงจรนับ รีจิสเตอร์ซึ่งจองไว้สำหรับใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับนับตัวที่ 3 รีจิสเตอร์คำสั่ง สำหรับหน้าที่พิเศษ (เช่น การอินเตอร์รัพต์ RTC : Read Time Clock) และอินพุต เอาต์พุตแบบอนุกรม

3.4 ความเร็วของเอ็มซีเอสเทียบกับ 6502 และ Z80

คำสั่งทั้งหมดของเอ็มซีเอส ทำงานด้วยรอบคำสั่งเดียวของภาษาเครื่องคือ ภายในระยะเวลาของสัญญาณนาฬิกา 12 ลูก เมื่อใช้ความถี่ 12 เมกะเฮิรตซ์ ดังนั้นในหนึ่งรอบคำสั่งภาษาเครื่องเท่ากับ 1 ไมโครวินาที ดังนั้นความเร็วของเอ็มซีเอส 51 เท่ากับ 6502 ซีพียูที่ใช้ความถี่ 2 เมกะเฮิรตซ์ หรือเท่ากับ Z80 ซีพียู ที่ความถี่ 8 เมกะเฮริตซ์ นอกจากนี้ เอ็มซีเอสยังมีความสามารถจัดการกับข้อมูลได้น้อยกว่า 8 บิต ต่อหนึ่งครั้งเพื่อช่วยให้ความสะดวกในการเขียนโปรแกรมโดยไม่ต้องลำบากในการปิดกั้น (mask) บิดที่ไม่ต้องการ

3.5 บัสและพอร์ต

จะเห็นโครงสร้างภายในของเอ็มซีเอส 51 ในรูปที่ 3.3 เราจะสมมติให้มีบัสสองทิศทาง 4 เส้น และพอร์ตขนาด 8 บิด ตามทฤษฎีแต่ที่จริงนั้นจะใช้ได้ก็ต่อเมื่อมีการใช้หน่วยความจำภายในตัว (รอบหรือแรม)

เมื่อไม่ใช้หน่วยความจำภายในพอร์ต 0 และ 2 จะถูกใช้เป็นบัสของข้อมูลและแอดเดรส (ขอให้ย้อนกลับไปดูรูปที่ 2) ดังนั้นพอร์ต 2 พอร์ต ยังคงใช้งานเป็นอินพุตและเอาต์พุต พอร์ 2 ทำหน้าที่เป็นสายสัญญาณแอดเดรส $A_{15} \ldots A_8$ ส่วนพอร์ต 0 ทำหน้าที่เป็นสายสัญญาณแอดเดรส $A_7 \ldots A_0$ ออกจากบิตข้อมูล $D_7 \ldots D_0$

เอาต์พุตของขา $\overline{RD}$ และ $\overline{WR}$ มาจากสายเอาต์พุตของพอร์ต 3 โดยโปรแกรมภายในใช้สัญญาณ $\overline{RD}$ และ $\overline{WR}$ เพื่อการเขียนและอ่านข้อมูล กับข้อมูลของหน่วยความจำภายนอก

3.5.1 ขา $\overline{PSEN}$

ขา $\overline{PSEN}$ เป็นขารับสัญญาณสำหรับเปิดให้มีการอ่านหน่วยความจำภายนอก ถ้าสังเหตทุก ๆ รอบคำสั่งระหว่างการทำงานด้วยโปรแกรมในรอม หรือ อีพรอม จะเห็นว่าสัญญาณ $\overline{PSEN}$ ทำงานถึงสองครั้งเหมือนสัญญา $\overline{ALE}$ เพราะว่ามีการอ่านข้อมูลจำนวน 2 ไบต์ ในแต่ละรอบคำสั่ง

ขา $\overline{PSEN}$ นี้ไม่ทำงานเมื่อมีภาษาเครื่องเก็บอยู่ในหน่วยความจำภายในและถ้าหากหน่วยความจำภายนอกไม่มีข้อมูลบรรจุอยู่ $\overline{PSEN}$ ก็จะไม่ทำงานด้วยเช่นกัน

ส่วน 8052 AH BASIC ไม่ใช้สัญญาณ $\overline{PSEN}$ เลย เพราะใช้รอมภายในตัว ซึ่งมีหน้าที่เป็นตัวแปลภาษาเบสิก

3.5.2 ขา $\overline{EA}$

เป็นขาอินพุตที่ใช้ร่วมกับแอดเดรสภายนอก โดยจะมีค่าลอจิก "0" เมื่อโปรเซสเซอร์อ่านคำสั่งจากหน่วยความจำภายนอก (ปกติโปรเซสเซอร์จะอ่านหน่วยความจำภายในน้อยกว่าอ่านจากหน่วยความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำภายนอก) ขา$\overline{EA}$ ยังมีหน้าที่เป็น T/R สำหรับโปรแกรมแรงดัน 21 โวลต์ เพื่อโปรแกรมให้กับอีพรอม (สำหรับกรณีที่ใช้ 8751 หรือ 8752)

3.6 วงจรนับและ วงจรตั้งเวลา

จากตางที่ 3.1 จะ เห็นว่า 8052 มีวงจรนับและวงจรตั้งเวลาชนิด 16 บิต มากกว่า 8052 อยู่หนึ่งตัวต่อไปเราจะศึกษาการทำงานอย่างย่อ ๆ ของวงจรนับและวงจรตั้งเวลา

เมื่อทำงานเป็นวงจรตั้งเวลา รีจิสเตอร์ที่ทำหน้าที่ตั้งเวลาจะเพิ่มค่าขึ้นหนึ่งของทุก ๆ รอบคำสั่งของเครื่อง และจะนับด้วยอัตราสูงสุดที่ 0.5 ของความเร็วสัญญาณนาฬิกาของโปรเซสเซอร์

เมื่อทำงานเป็นวงจรนับ รีจิสเตอร์วงจรนับจะเพิ่มขึ้นหนึ่งเมื่อมีสัญญาณป้อนให้อินพุต T0 , T1 หรือสัญญาณภายใน โดยจะใช้สัญญาณขาลง อัตราการนับสูงสุดคือ 0.5 ของความเร็วสัญญาณนาฬิกาของโปรเซสเซอร์

วงจรนับและวงจรตั้งเวลา 0 และ 1 มีวิธีโปรแกรมให้ทำงานได้ต่างกันถึง 4 แบบ รวมทั้งการทำงานเป็น 8 บิต หรือ 16 บิต และการบรรจุค่าพรีเซตหนึ่งค่าได้เองอย่างอัตโนมัติ

วงจรนับและวงจรตั้งเวลาที่ 1 เลือกโปรแกรมให้ทำหน้าที่เป็นตัวกำเนิดสัญญาณของอัตราการส่งบิตออกไปอย่างอนุกรม สำหรับใช้ในการอินเตอร์เฟสได้

วงจรนับและวงจรตั้งเวลาที่ 2 (เฉพาะ 8052 เท่านั้น) มีการทำงานย่อย ๆ อีก 3 ชนิด

1. วงจรนับ 16 บิตที่สามารถโหลดค่ากลับคืนเองอย่างอัตโนมัติ
2. วงจรนับที่จองไว้ ชนิด 16 บิต
3. วงจรกำเนิดสัญญาณของการส่งบิต เพื่อใช้อินเตอร์เฟส

3.7 พอร์ตชนิดอนุกรมอยู่ภายใน

ไมโครคอนโทรลเลอร์ทั้งหมดในตระกูล เอ็มซีเอส 51 เป็นชิพที่มีอินเตอร์เฟสอนุกรมชนิดสองทิศทาง ทำให้รับและส่งข้อมูลพร้อมกัน ตัวรับข้อมูลชนิดอะซิงโครนัส (Asynchronous Receiver) มีบัฟเฟอร์สำหรับข้อมูลเป็นพิเศษเพื่อเพิ่มความเร็วในการสื่อสาร

พอร์ตชนิดอนุกรมนั้น เราสามารถเลือกโปรแกรม เพื่อเลือกใช้การทำงานแบบใดแบบหนึ่งใน 4 แบบ ด้วยการใช้โปรแกรมควบคุมอัตราการส่งข้อมูลและรูปแบบของข้อมูล

อัตราการส่งข้อมูลที่เลือกใช้ได้จะสูงถึง 19,200 บิต / วินาที ด้วยความเร็วของสัญญาณนาฬิกา 1 เมกะเฮิรตซ์ สำหรับใช้ในระบบการสื่อสารของโปรเซสเซอร์หลายตัวร่วมกัน เราจะเลือกความเร็วของสัญญาณนาฬิกาด้วยวงจรนับและวงจรตั้งเวลา

3.8 อินเตอร์รัพต์และชุดคำสั่ง

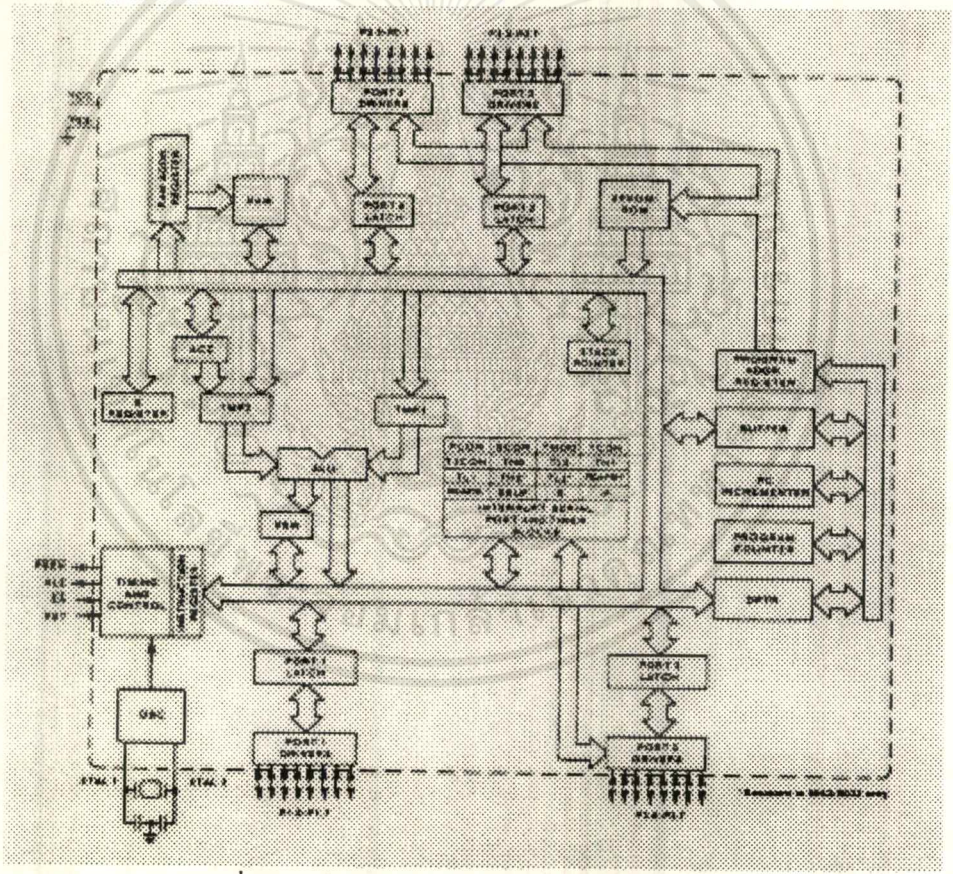
8051 รับอินเตอร์รัพต์ได้ 5 แหล่ง ส่วน 8052 รับอินเตอร์รัพต์จากอุปกรณ์อื่น ๆ ได้ถึง 6 แหล่ง คือ ขา$\overline{INT0}$ และ ขา $\overline{INT1}$ (กำหนดให้ใช้ระดับพัลส์หรือขอบขาพัลส์ก็ได้) วงจรนับและวงจรตั้งเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038962

ที่ 0 และ 1 (สำหรับไมโครโปรเซสเซอร์ 8052 เพิ่มวงจรตั้งเวลา / วงจรนับตัวที่ 2) และสุดท้ายจากพอร์ตอนุกรม เราสามารถกำหนดลำดับความสำคัญของอินเตอร์รัพต์ได้ 2 ระดับ โดยไม่ต้องอาศัยวงจรภายนอกช่วย แต่ละแหล่งของอินเตอร์รัพต์ 5 หรือ 6 แหล่งนั้นจะกำหนดเป็นค่าเวกเตอร์เฉพาะ (ตัวชี้แอดเดรส)

ดังนั้นเมื่อมีอินเตอร์รัพต์เข้ามาแล้ว ตัวโปรเซสเซอร์จะกระโดดไปที่ส่วนของโปรแกรมที่นำมาประสงค์ของอินเตอร์รัพต์นั้น หลังจากเก็บข้อมูลต่าง ๆ ของโปรแกรมเคาน์เตอร์ลงในสแต็ก ชุดคำสั่งทั้งหมดของตัวคอนโทลเลอร์ในตระกูลเอ็มซีเอส แสดงไว้ในตารางที่ 3.2

เนื่องจาก MCS-51 เป็นชิงเกิลชิพที่ค่อนข้างจะสมบูรณ์ โดยภายในตัวมันประกอบด้วย ไทเมอร์/ตัวนับ,ออสซิลเลเตอร์, พอร์ต I/O , พอร์ตอนุกรม ซึ่งรายละเอียดและข้อดีของไมโครคอนโทรลเลอร์ตระกูลนี้มีคนเขียนหลายท่านแล้ว จะขอไม่กล่าวถึงในที่นี้อีก แต่จะเขียนถึงการใช้งานเกี่ยวกับพอร์ตสื่อสารอนุกรมเลย โดยจะเริ่มด้วยการทำความเข้าใจถึงรีจิสเตอร์ต่าง ๆ ที่เกี่ยวข้องและจบด้วยโปรแกรมตัวอย่าง ซึ่งได้ทำการทดลองแล้วและได้ผลเป็นอย่างดี



รูปที่ 3.3 บล็อคไดอะแกรมการทำงานของ MCS-51

รูปที่ 3.3 เป็นบล็อคไดอะแกรมการทำงานของ MCS-51 แสดงส่วนที่เกี่ยวข้อง และจัดการเกี่ยวกับการสื่อสารข้อมูลทางพอร์ตอนุกรม โดยใช้พอร์ต 3 บิด P.3.O และ P.3.1 เป็นส่วนติดต่อข้อมูลที่ P.3.O หรือขา 10 จะทำหน้าที่เป็นอินพุตรับข้อมูลที่ส่งเข้ามาเรียกว่า ขา RXD ส่วนในการส่งข้อมูล MCS-51 จะส่งออกที่ขา P.3.1 หรือ TXD ขา 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.9 รีจิสเตอร์ที่เกี่ยวข้อง

การใช้งาน MCS-51 ในด้านสื่อสารข้อมูลทางพอร์ตอนุกรม จะมีรีจิสเตอร์ที่เกี่ยวข้องหลายตัว คือ

PCON : เพาเวอร์คอนโทรลรีจิสเตอร์

การจัดเรียงบิดต่าง ๆ ใน PCON จากบิด 7 - บิด 0 เป็นไปดังนี้

SMOD	-	-	-	GFI	GFO	PD	IDL
7	6	5	4	3	2	1	0

สำหรับรีจิสเตอร์ตัวนี้จะใช้งานเพียงบิดเดียว คือ SMOD สำหรับรีจิสเตอร์ตัวนี้มีสิ่งที่ต้องคำนึงถึง คือ ถ้า SMOD เป็น "0" ค่า K เป็น 1 และ SMOD เป็น "1" ค่า K เป็น 2 รีจิสเตอร์ จะไม่สามารถใช้คำสั่ง SETB ได้ เพราะฉนั้นเราจะต้องใช้วิธีการ OR เพื่อเซตค่า และ AND เพื่อเคลียร์ค่าในรีจิสเตอร์ เช่น ORL PCON, #10000000B (SETB PCON.7) เมื่อ SMOD เป็น "1" K เป็น 2 และ ANLPCON, #01111111B (CLR PCON.7) เมื่อ SMOD เป็น "0" K เป็น 1

เหตุที่ต้องใช้วิธีการเช่นนี้เนื่องจาก PCON เป็นรีจิสเตอร์เฉพาะงาน ทำให้การเซ็ตหรือรีเซตบิตไม่สามารถใช้คำสั่ง SETB ได้ ซึ่งต่างจากรีจิสเตอร์ตัวอื่น ๆ เช่น IE, IP, TCON เป็นต้น

IE : อินเตอร์รัพต์ อีนาเบิล รีจิสเตอร์

ภายใน IE ประกอบด้วยบิดต่าง ๆ ดังต่อไปนี้ (เรียงจากบิด 7-บิด 0)

EA	-	ET2	ES	ET1	EX1	ET0	EX0
7	6	5	4	3	2	1	0

ในกรณีที่ต้องการให้พอร์ตอนุกรมมีการอินเตอร์รัพต์ขณะส่งและรับข้อมูล ให้ใช้คำสั่ง SET EA และ SET ES แล้วนำเอาโปรแกรมในการรับและส่งไปไว้ที่ตำแหน่ง RI & TI (0023H) แต่ต้องไม่เกิน 8 ไบต์ ซึ่งในความเป็นจริงแล้วโปรแกรมมักจะเกิน 8 ไบต์ ดังนั้นที่ตำแหน่ง 0023H ส่วนมากจึงเป็นคำสั่ง JUMP เสียมากกว่า หรือใครจะสามารถเขียนโปรแกรมโดยมีความยาว 8 ไบต์ก็ไม่ว่ากันซึ่งอาจเป็นไปได้คือ เป็นโปรแกรมรับและเก็บข้อมูลเฉย ๆ แล้วค่อยใช้โปรแกรมหลักเป็นตัวกำหนดการทำงานอีกทีแต่ในโปรแกรมตัวอย่างไม่ได้ใช้อินเตอร์รัพต์ ดังนั้นจึงใช้คำสั่ง CLR EA และใช้วิธีการอ่านสถานะจากรีจิสเตอร์ SCON อีกที

IP : อินเตอร์รัพต์ ไพออริตี้ รีจิสเตอร์

บิดทั้ง 8 ภายใน IP มีการกำหนดลักษณะการทำงานต่างกัน ดังนี้

-	-	PT2	PS	PT1	PX1	PT0	PX0
7	6	5	4	3	2	1	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ตัวนี้เป็นรีจิสเตอร์ใช้กำหนดความสำคัญในการอินเตอร์รัพต์ ถ้าให้บิดใดเป็น "1" ลักษณะการอินเตอร์รัพต์ในบิดนั้นจะมีความสำคัญสูงสุด เช่น ถ้าเราต้องการให้การรับ/ส่ง ข้อมูลมีการอินเตอร์รัพต์ และมีความสำคัญสูงสุดก็เซตบิด PS ให้เป็น 1 โดยใช้คำสั่ง SETB PS พอพูดถึงลำดับความสำคัญของการอินเตอร์รัพต์ทำให้นึกถึงเพื่อนผมคนหนึ่ง เขาเล่นเซตทุกบิดเลย ด้วยเหตุผลว่าทุกตัวมีความสำคัญหมด คงพอจะนึกผลลัพธ์ออกนะครับ ผลก็คือมีค่าเหมือนตอนที่ยังไม่ได้เซตค่าอะไรเลยนั่นแหละ ดังนั้นในการใช้งานสื่อสารข้อมูลทางพอร์ตอนุกรมจึงต้องเซ็ตว่า PS ให้เป็น "1" บิดเดียวเท่านั้น หากต้องการให้มีการอินเตอร์รัพต์

TCON : ไทเมอร์ / เคาน์เตอร์

คอนโทรล รีจิสเตอร์

การจัดเรียงความหมายในแต่ละบิดของ TCON แสดงดังต่อไปนี้

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
7	6	5	4	3	2	1	0

ในรีจิสเตอร์มีบิดที่เราจะต้องใช้คือ TR0 และ TR 1 เป็นตัวควบคุมไทเมอร์และเคาน์เตอร์ โดยถ้าให้เป็น 0 จะต้องหมายถึง "Off" และ "1" จะหมายถึง "On" เพราะฉนั้นหลังจากที่เซตค่าต่าง ๆ พร้อมหมดแล้ว เราก็จะ On เคาน์เตอร์โดยใช้คำสั่ง SETB TR0 หรือ SETB TR1 ถ้าต้องการ Off ก็ใช้คำสั่ง CLR TR0 หรือ CLR TR1

TMOD : ไทเมอร์ / เคาน์เตอร์

โหมด คอนโทรล รีจิสเตอร์

มีลักษณะการจัดเรียงบิตดังนี้

GATE	C/T	M1	M0	GATE	C/T	M1	M0
7	6	5	4	3	2	1	0

ในการใช้งานพอร์ตสื่อสารอนุกรมจะใช้งานรีจิสเตอร์ TMOD นี้บางบิต คือ

GATE - ถ้า GATE เป็น "1" TRx เป็น "1" (จะเป็น TR0 หรือ TR1 ก็ได้) ไทเมอร์ / เคาน์เตอร์ จะทำงานเมื่อ INTx เป็น "1" แต่ถ้า GATE เป็น "0" ไทเมอร์ / เคาน์เตอร์จะทำงานก็ต่อเมื่อ TRx เป็น "1" เท่านั้น

นั่นคือถ้าเราให้ GATE ของไทเมอร์ 1 เป็น "1" และ TR1 ใน TCON เป็น "1" แล้วไทเมอร์ / เคาน์เตอร์ 1 จะทำงานเมื่อ INT1 เป็น "1" เท่านั้น แต่ถ้าเราให้ GATE เป็น "0" แล้ว ไทเมอร์ / เคาน์เตอร์ 1 จะทำงานเมื่อ TR1 เป็น "1" เท่านั้น ไทเมอร์ 0 ก็เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

C/T - ถ้า C/T เป็น "1" จะเป็นโหมดเคาน์เตอร์ โดยจะนับสัญญาณนาฬิกาจากขา Tx (ซึ่งก็คือ T0 หรือ T1 ขึ้นอยู่กับไทเมอร์ / เคาน์เตอร์ตัวไหน) ถ้า C/T เป็น "0" จะเป็นโหมดไทเมอร์ โดยจะเอาสัญญาณนาฬิกาจากภายในซีพียูมาตั้งเวลา

M1,M0 - เป็นตัวเลือกโหมดการทำงานของไทเมอร์ / เคาน์เตอร์ โดยมีรายละเอียดดังตารางที่ 3.1

เราจะใช้ไทเมอร์ 1 ในการสร้างบ็อดเรต โดยใช้ไทเมอร์โหมด 2 โดยต้องกำหนดค่าดังนี้

GATE = "0" (สั่งให้ทำงานโดยให้ TR1 เป็น "1" อย่างเดียว)

C/T = "0" (โหมดไทเมอร์)

M0 = "0"

M1 ="1" (8 บิด ออโตรีโหลด)

เราจึงต้องเขียนโปรแกรม MOVE ค่าดังนี้ MOV TMOD, #0010xxxxB ; x "0" หรือ "1"

ตารางที่ 3.1 การเลือกโหมดการทำงานของไทเมอร์ / เคาน์เตอร์ โดยใช้ M0, M1

M0	M1	โหมด	คำอธิบาย
0	0	0	13 บิดไทเมอร์
0	1	1	16 บิดไทเมอร์
1	0	2	8 บิดออโต้รีโหลด
1	1	3	ไทเมอร์ / เคาน์เตอร์ 8 บิด 2 ชุด

SCON : ซีเรียล พอร์ต

คอนโทรลรีจิสเตอร์

ภายในรีจิสเตอร์ SCON มีการแบ่งหน้าที่ของบิดทั้ง 8 เรียงจากบิดที่ 7 - บิดที่ 0 ดังนี้

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
7	6	5	4	3	2	1	0

การเลือกโหมดการทำงานของ SCON จะต้องใช้การควบคุมจากขา M0 และขา M1. ดังตารางที่ 3.2

บิดต่าง ๆ ที่เกี่ยวข้องในการติดต่อสื่อสารทางพอร์ตอนุกรมในรีจิสเตอร์ SCON ได้แก่

SM2 - ถ้า SM2 เป็น "1" ในโหมดที่ 2 หรือ 3 RI จะไม่เป็น "1" ถ้าข้อมูลใน bit ที่ 9 เป็น "0" ส่วนในโหมดที่ 1 ถ้า SM2 เป็น "1" RI จะไม่เป็น "1" ถ้ายังไม่ได้รับบิทที่เป็นสตอปบิท

REN - ถ้าบิดนี้เป็น "0" จะส่งได้ทางเดียว ถ้าเป็น "1" จะทำงานได้ทั้งรับและส่ง

TBB / RB8 - ถ้าใช้งานในโหมด 2 หรือ 3 RB8 จะเป็นข้อมูลที่รับบิดที่ 8 และ TR8 จะเป็นข้อมูลที่รับบิดที่ 8 และ TR8 จะเป็นข้อมูลที่ต้องการส่งออกบิดที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TI - จะเป็น "1" เมื่อข้อมูลใน SBUF (serial buffer) ได้ส่งไปแล้ว

RI - จะเป็น "1" เมื่อได้รับข้อมูลครบทุกบิต หรือพบสตอปบิตแล้ว

ทั้งหมดที่กล่าวมานั้นเป็นการเซตอัพใน 80341 , 8051 และ 8751 แต่ถ้าเป็น 8032 และ 8052 จะมีรีจิสเตอร์เพิ่มเข้ามาอีกหนึ่งตัวคือ T2CON

T2CON : ไทเมอร์ / เคาน์เตอร์ 2

คอนโทรล รีจิสเตอร์

ในแต่ละบิตของรีจิสเตอร์มีความหมายเรียงลำดับจากบิตที่ 7- บิตที่ 0 ดังนี้

TF2	EXF2	RLCK	TCLK	EXEN2	TR2	C/T2	CP/RL2
7	6	5	4	3	2	1	0

บิตที่ต้องสนใจได้แก่

RLCK - ถ้าเป็น "1" บ๊อตเรตในการรับจะได้มาจากไทเมอร์ 2 ถ้า RLCK เป็น "0" บ๊อตเรตในการรับจะได้มาจากไทเมอร์ 1

ตารางที่ 3.2 การเลือกโหมดการทำงานและบ๊อตเรตของ SCON โดยใช้ M0 และ M1

M0	M1	โหมด	คำอธิบาย	บ๊อตเรต
0	0	0	ซิฟต์รีจิสเตอร์	ความถี่สัญญาณนาฬิกา / 12
0	1	1	UART 8 บิต	เลือกค่าได้
1	0	2	UART 9 บิต*	ความถี่สัญญาณนาฬิกา /32 หรือ 64
1	1	3	UART 9 บิต*	เลือกค่าได้

*UART : Universal Asynchronous Receiver Transceiver

TLCK - ถ้าเป็น " 1 " บ๊อตเรตในการส่งจะได้มาจากไทเมอร์ 2 และถ้า RLCK เป็น "0" บ๊อตเรตในการส่งจะได้มาจากไทเมอร์ 1 ประโยชน์ก็คือ ในกรณีที่การรับและส่งใช้บ๊อตเรตไม่เท่ากัน เราก็ยังสามารถทำงานได้

TR2 - ถ้า TR2 เป็น "0" ไทเมอร์ 2 จะหยุดทำงานและถ้าเป็น "1" ไทเมอร์ 2 จะทำงาน

C/T2 - เป็นตัวเลือกโหมด (Mode) ถ้าเป็น "1" จะเป็นโหนด เคาน์เตอร์ ถ้าเป็น "0" จะเป็นโหมดไทเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

รหัสแถบ (Barcode)

รหัสแถบมีประโยชน์มาก ในการใช้เก็บข้อมูลหรือบันทึกข้อมูลเกี่ยวกับปริมาณการขายสินค้านั้น ๆ เพื่อเป็นประโยชน์ในด้านการบริหารจัดการความเคลื่อนไหวของสินค้าประเภทต่าง ๆ ด้วยความรวดเร็วและให้ความถูกต้องและความปลอดภัยของข้อมูลสูง

ในเรื่องของความเร็วนั้น เราสามารถที่จะอ่านรายละเอียดของรหัสของอักขระจำนวน 12 ตัว ได้ภายในเวลาเพียง 2 วินาที ขณะเดียวกันในเรื่องความถูกต้องปลอดภัยนั้น รหัสแถบจะถูกพิมพ์ด้วยการใช้หมึกพิมพ์ที่มีผลตอบสนองต่อแสงอินฟราเรด ซึ่งเป็นการป้องกันการก๊อปปี้ จากการถ่ายแบบธรรมดา (Photocopiers)

การอ่านรหัสของบาร์โค้ดสามารถอ่านผ่านข้ามช่องอากาศได้ โดยมีความยาวโฟกัส 2-4 มิลลิเมตร และเมื่อบาร์โค้ดถูกอ่านและถอดรหัสมาเป็นตัวอักขระแล้วจะไม่เกิดการสับสนหรือ

4.1 ความเป็นมาของรหัสแถบ และมาตรฐาน

เริ่มต้นเมื่อปี ค.ศ. 1990 สหรัฐอเมริกาออกสิทธิบัตรรับรองรหัสแถบ แบบที่เรียกว่า Circular Barcode ในปี ค.ศ. 1960 ก็มีการรับรองรหัสแถบแบบที่เรียกว่า Rail Identification symbol หลังจากนั้นเป็นต้นมา วิวัฒนาการของรหัสแถบ รูปแบบต่าง ๆ ก็มีมากขึ้นและเริ่มใช้งานจริงจังเมื่อปี ค.ศ. 1970 เมื่อคณะกรรมการบริหารด้านห้างสรรพสินค้าของสหรัฐอเมริกา ได้นำรหัสที่เรียกว่า UPC (Universal Product Code) ซึ่งเป็นรหัสที่ใช้กันมากในสินค้า ออกเผยแพร่และใช้กันแพร่หลายในสหรัฐอเมริกา และยุโรปตั้งแต่ ค.ศ. 1973 และ ค.ศ. 1977 ตามลำดับ

การใช้งานในด้านอื่น ๆ เริ่มตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา เมื่อหน่วยป้องกันประเทศของสหรัฐอเมริกา ใช้เพื่อเช็คอุปกรณ์เครื่องมือและยุทโธปกรณ์คงคลังขณะที่โรงงานอุตสาหกรรมนำไปใช้เพื่องานต่าง ๆ มากขึ้น แต่คนทั่วไปเริ่มคุ้นเคยกับรหัสแถบเป็นอย่างดี จากรหัสสินค้าและการชำระเงินที่คอมพิวเตอร์รวมออกมาจากการอ่านรหัสแถบบนสินค้าเหล่านั้น จากความสะดวกสบายเหล่านี้ทำให้สามารถลดพนักงาน ณ จุดนี้ลงได้ ในปี ค.ศ. 1981 มีห้างสรรพสินค้ามากกว่า 4000 แห่ง ในสหรัฐอเมริกา และแคนาดาใช้รหัสแถบในธุรกิจนี้ นอกจากนี้ยังใช้กับกิจการอื่น ๆ เช่น ห้องสมุด บริการสุขภาพ งานเอกสาร งานการทหาร การวิจัย การผลิตสินค้าเป็นต้น

4.2 หลักการของรหัสแถบ

รหัสแถบเป็นการแทนข้อมูลที่เป็นรหัสเลขฐานสอง (Binary Code) ในรูปแบบของแถบขาว-ดำ ที่มีความกว้างของแถบที่ต่างกันในแถบดำหรือแถบขาวที่มีความกว้าง แทนค่าเป็น 1 และแถบที่มีความแคบ (หรือมองด้วยตาเหมือนเป็นเส้นตรงเล็ก ๆ) ทั้งขาวและดำจะมีค่าเป็น 0 แถบขาว-ดำ ที่ว่านี้มีลักษณะและชื่อที่ใช้คือ

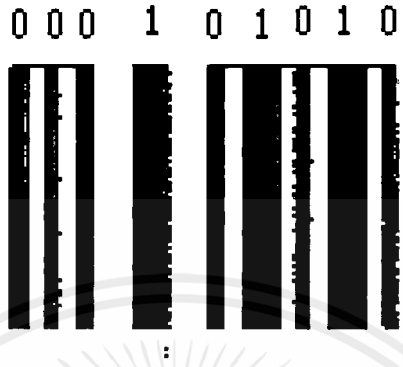
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถบสีดำที่มีความกว้างมาก เรียกว่า Wide Bar

เส้นหรือแถบสีดำที่มีความกว้างน้อย เรียกว่า Narrow Bar

ช่องว่างหรือแถบสีขาวที่มีความกว้างมาก เรียกว่า Wide Space

ช่องว่างหรือเส้นสีขาว ที่มีความกว้างน้อย เรียกว่า Narrow Space



รูปที่ 4.1 การแทนค่าเลขฐานสองของแถบต่าง ๆ

4.3 การอ่านรหัสแถบ

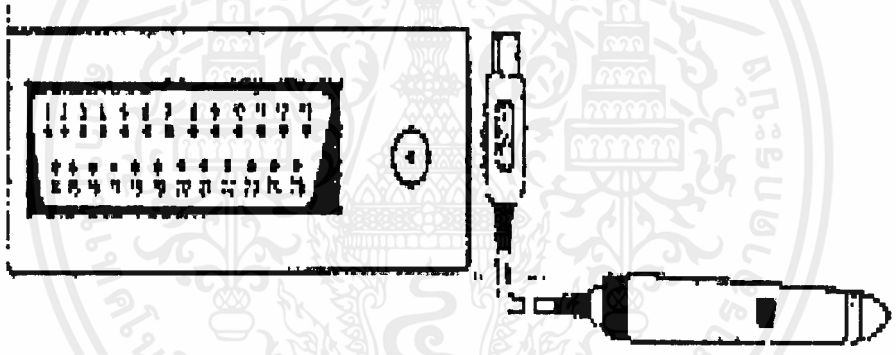
ในการอ่านรหัสแถบใช้หลักการเปลี่ยนรหัสแอสกี้ (ASCI) ซึ่งดูได้จากภาคผนวก ข. โดยอาศัยความแตกต่างกันระหว่าง แถบเข้มและพื้นที่ว่าง โดยที่พื้นที่ว่าง จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบเข้ม (ซึ่งใช้สีดำหรือสีอื่นที่มีความเข้มมาก) ตัวอ่าน (Barcode Reader) จะประกอบด้วยตัวกำเนิดแสงที่ส่งผ่านเลนซ์ออกมา โดยถูกบังคับทิศทางให้มีจุดรวมแสงเล็กที่สุด กับตัวรับแสงที่มีความไวสูง ทั้ง 2 อย่างนี้จะบรรจุไว้ในตัวอ่านเดียวกันที่มีหลายรูปแบบ แต่แบบที่เป็นพื้นฐานที่สุดอยู่ในรูปคล้ายปากกาขนาดใหญ่ (Wand Type)

รูปที่ 4.2 ลักษณะและขนาดของรหัสแถบมาตรฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอ่านจะถูกสแกนผ่านรหัสแถบในขณะที่ตัวกำเนิดแสงจะทำให้เกิดแสงส่งผ่านเลนซ์ไปกระทบบนรหัสแถบและสะท้อนกลับจากรหัสแถบ (แถบและช่องว่าง) กลับไปยังตัวรับแสง (Photo Sensor) ที่เกิดค่าความแตกต่างขึ้นตามหลักการสะท้อนกลับในแต่ละแถบ ทำให้เกิดสภาวะลอจิก "0" และลอจิก "1" ขึ้นตามที่กล่าวมาแล้วข้างต้น ซึ่งเมื่อรวมสภาวะลอจิก "0" และ "1" ทั้งหมดตลอดความกว้างของทุกแถบแล้วจะตรงกับแพตเทิร์นที่ได้กำหนดไว้แล้วในตัวอ่านรหัสแถบจะใช้สีแดงหรือสีขาว แต่ส่วนใหญ่จะใช้สีแดง เนื่องจากแถบสีขาวต้องการพลังงานและความเข้มของแสงสูงมากกว่าสีแดง แสงสีแดงสามารถอ่านรหัสที่พิมพ์ (Contrast) ด้วยสีต่าง ๆ ได้ ยกเว้นรหัสที่พิมพ์ด้วยสีแดง

องค์ประกอบที่สำคัญ 2 ประการที่จำเป็นอย่างมากในการอ่านรหัสแถบได้ถูกต้อง ประการแรก คือ พื้นที่ภายในแถบและช่องว่าง จะต้องทำให้เกิดความแตกต่างของการสะท้อนกลับ (Contrast) อย่างมาก เช่น แถบสีดำและช่องว่างสีขาว เป็นต้น ซึ่งปกติความแตกต่างจะต้องอยู่ในช่วงระหว่างอัตรา 80-90 เปอร์เซ็นต์ขึ้นไป ประการสุดท้ายคือ ความกว้างระหว่างแถบกว้าง หรือช่องว่างกว้างต่อแถบแคบหรือช่องว่างแคบจะเป็นอัตราส่วน 2:0.5} 2:1 และ 3:1



รูปที่ 4.3 ลักษณะของเครื่องอ่านแบบที่ต่อกับ Com Port ของไมโครฯ

ตัวอ่านรหัสแถบแบบปากกาที่มีจำหน่ายในเมืองไทยชนิดที่นำมาต่อกับไมโครคอมพิวเตอร์มี 2 แบบ ที่นิยมกันมากคือ แบบที่ต่อเข้ากับ RS422 หรือ RS232 (COM1, COM2) ซึ่งมีชุดควบคุม (COM1, COM2) ซึ่งจะมีชดต่าง ๆ ได้ตามต้องการ เช่น ความเร็ว ชนิดของรหัส ฯลฯ และอีกแบบหนึ่งจะใช้ต่อแทนคีย์บอร์ด โดยใช้ Keyboard Mulator เป็นตัวควบคุมการทำงานทั้งสองแบบให้คุณสมบัติในการอ่านและเปลี่ยนรหัสได้ทั้ง 2 ทิศทาง โดยไม่ผิดพลาดนอกเหนือจากคุณสมบัติเฉพาะของเครื่อง แล้วยังขึ้นอยู่กับโปรแกรมที่เขียนขึ้นมาควบคุม โดยผู้ใช้อีกด้วย

นอกจากนี้ยังมีตัวอ่านที่มีประสิทธิภาพสูงประเภทเลเซอร์ (Laser Barcode Scanner) ซึ่งมีทั้งชนิดที่ยังต้องใช้คน และแบบสแกนอัตโนมัติ ทั้ง 2 แบบนี้ให้ความรวดเร็ว ความเชื่อถือได้ และลดขนาดการพิมพ์รหัสแถบให้เล็กลงได้อย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ชนิดของรหัสแถบ

ปัจจุบันชนิดของรหัสแถบที่นิยมใช้แพร่หลายแบ่งได้เป็น

4.4.1. ชนิดรหัส 2 ใน 5 (2 of 2 Code)

เป็นรหัสที่มีใช้ตั้งแต่ ค.ศ. 1980 เป็นแบบที่ง่ายที่สุดในการใช้งาน การที่ชื่อเรียกว่า 2 ใน 5 เพราะใน 1 รหัสจะประกอบไปด้วยแถบ 5 แถบ (5 บิต) แต่จะมีแถบกว้างที่มีค่าเป็น 1 (แถบกว้าง) เพียง 2 แถบ (2 บิต) เท่านั้น ส่วนบิตที่เหลือเป็น 0 ทั้งหมดคือการแทนด้วยแถบแคบ (Narrow Bar) 3 แถบ โดยไม่นำส่วนที่เป็นช่องว่าง (Space Bar) มาใช้เลย

รหัส 2 ใน 5 นี้เป็นรหัสที่ใช้แทนข้อมูลได้เฉพาะตัวเลข 0-9 เพียงแค่ 10 รหัสเท่านั้น เริ่มต้นจาก Start Code 3 บิต คือ 110 (แถบกว้าง 2 และ แถบแคบ 1) กับปิดท้ายด้วย Stop Code 3 บิต คือ 101 ส่วนรหัสทั้ง 5 บิต ที่แทนเลข 0-9 ดูได้จากตารางในรหัส Interleaved 2 of 5

รูปที่ 4.4 รหัสแถบชนิด 2 ใน 5

4.4.2. ชนิดรหั 2 ใน 5 แบบสอดแทรก (Interleaved 2 of 5)

รหัสแบบนี้คล้ายคลึงกับแบบแรกมาก เพราะพัฒนามาจากรหัสแบบแรก เนื่องจากรหัส 2 ใน 5 ไม่ได้นำส่วนที่เป็นช่องว่างกว้าง และช่องว่างแคบมาใช้ คงใช้แต่เพียงแถบกว้างและแถบแคบ จึงทำให้ความหนาแน่นของข้อมูลน้อยลง นั่นคือ เมื่อต้องการบรรจุข้อมูลต่อเนื่องหลายตัวเลขจะต้องใช้แถบที่มีความกว้างมากขึ้น

รหัส 2 ใน 5 แบบสอดแทรก ได้ดัดแปลงนำส่วนที่เป็นช่องว่างทั้ง 2 ชนิดมาใช้งานด้วย โดยการสอดแทรกรหัสลงไปอีก 1 รหัส ทุก ๆ ช่วง 5 แถบของรหัสปกติที่เป็นแถบสีดำ แต่ก็ยังแทนรหัสตัวเลข 0-9 ได้เพียง 10 รหัสเท่านั้น

จากรูปที่ 5 ใช้รหัสแทนตัวเลข 1234. ตามลำดับ การใช้งานของรหัส 2 ใน 5 แบบสอดแทรกจะเริ่มต้นส่วนที่เป็นตัวเริ่มต้น (Start Code) ทางด้านซ้ายประกอบด้วยแถบแคบ 5 แถบ และช่องว่างแคบ 2 แถบสลับกัน ส่วนทางด้านขวาเป็นตัวหยุด (Stop Code) ประกอบด้วยแถบกว้าง 1 แถบ ช่องว่างแคบ 1 แถบ และแถบแคบ 1 แถบ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.5 รหัสแถบชนิด 2 ใน 5 แบบสอดแทรก

ภายในระหว่างตัวเริ่มต้น (Start Code) และตัวหยุด (Stop Code) แบ่งเป็น 2 ส่วน ส่วนแรกคือส่วนที่เป็นแถบดำกว้างและแคบ จะใช้แทนรหัสเหมือนตัวเริ่มต้น 2 ใน 5 ขณะเดียวกันในส่วนของแถบเหล่านี้จะมีแถบช่องสีขาวกว้างและแคบแทนได้เช่นเดียวกับรหัส 2 ใน 5 ปกติ จากตัวอย่าง 5 แถบแรกที่เป็น สีดำแทนค่าได้เท่ากับ 1 แถบขาวในช่วงเดียวกันเท่ากับ 2 สีดำ ช่วงต่อมา 5 แถบแทนได้เท่ากับ 3 และ 5 แถบขาวต่อมาเท่ากับ 4 เช่นนี้ ตลอดไปจนหมดรวมถึงเป็ นค่าที่อ่านได้เท่ากับ 1234....

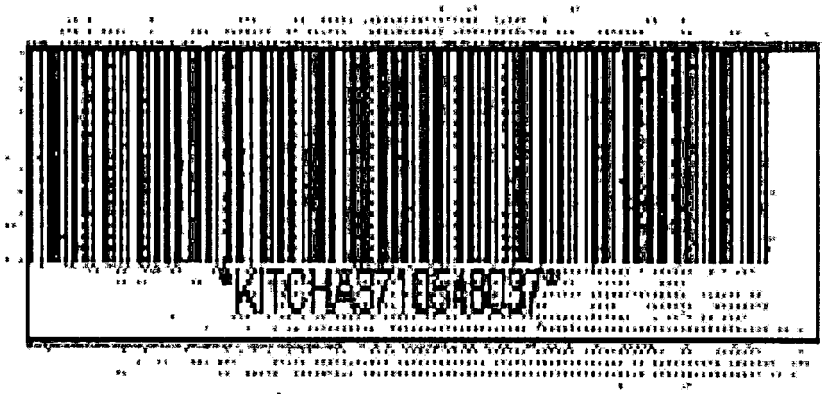
ตารางที่ 4.1 รหัสเลขมาตรฐาน 2 ของรหัส 2 ใน 5 ทั้งสองแบบ

ตัวเลข	เลขฐาน 2 ทั้ง 5 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

4.4.3. ชนิดรหัส 3 ใน 9 (3 of 9 or 39 Code)

รหัส 3 ใน 9 เป็นรหัสที่ใช้แทนตัวอักขระทั้งหมด 44 อักขระ เป็นอักษรตัวใหญ่ 26 รหัส เลข 0-9 10 รหัส และอักขระพิเศษอีก 8 รหัส เป็นการประยุกต์ใช้รหัส 2 ใน 5 โดยการนำเอาส่วนที่เป็นแถบดำ 5 แถบ และแถบว่าง 4 แถบ รวมเป็น 9 แถบ แทน 1 รหัส ในแถบสีดำ 5 แถบนั้น ประกอบด้วยแถบกว้างที่เป็นบิต 1 อยู่ 2 แถบ และแถบแคบที่เป็นบิต 0 อยู่ 3 แถบ ส่วนแถบว่าง (Space) 4 แถบ ประกอบด้วยแถบกว้างที่เป็นบิต 1 อยู่ 1 แถบ และแถบแคบที่เป็นบิต 0 อีก 3 แถบ ดังนั้นเมื่อรวมทั้งหมด 9 แถบ จะเป็นบิต 1 อยู่ 3 แถบ และบิต 0 อยู่ 6 แถบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 รหัสแถบชนิดรหัส 3 ใน 9

รหัส 3 ใน 9 มีส่วนเริ่มต้น (Start code) และสิ้นสุด (Stopcode) ด้วยรหัสเดียวกันคือ (Asterisk) ซึ่งมีรหัสฐาน 2 เป็นแถบ 00110 และช่องว่าง 1000 ข้อดีของรหัสชนิดนี้คือใช้งานได้กว้างขวางมากขึ้น เพราะสามารถใช้ตัวเลขปนกับตัวอักษรและเครื่องหมายต่าง ๆ ได้ ซึ่งรหัสทั้งหมดแทนได้ตามตารางที่ 4.2

ตารางที่ 4.2 เลขฐาน 2 ของรหัส 3 ใน 9

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

Code 39 บางครั้งเราเรียกกันว่า 3 ใน 9 (3 of 9) ซึ่งพบเห็นได้ทุก ๆ แห่ง โดยเฉพาะสินค้าขายปลีก สามารถแทนอักขระ A ถึง Z , ตัวเลข 0 ถึง 9 และอักขระพิเศษอีก 8 ตัวด้วยรหัสส่วนใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะพบเห็นรหัสบาร์โค้ดชนิดนี้โดยทั่วไปเช่น สินค้าที่มีอยู่ในร้านค้า หรือหมายเลขที่มีอยู่บนข้อความของเครื่องใช้ไฟฟ้า และส่วนประกอบของเครื่องยนต์ต่าง ๆ

รหัส Code 39 นี้มีอยู่ 2 ระดับ (กว้างและแคบ) นั่นคือแถบเส้นและช่องว่างจะเป็นได้ 2 ระดับไม่กว้างก็แคบ โดยมีอัตราส่วนของช่วงแคบต่อช่วงกว้างเท่ากับ 1 : 2.5

อักขระแต่ละตัวจะถูกแทนด้วยส่วนประกอบ 9 ส่วน โดยจะเป็นแถบเส้น 5 ส่วน และช่องว่างอีก 4 ส่วนสำหรับ 3 ใน 9 ส่วนนั้น เป็นส่วนกว้างซึ่งก็คือแถบเส้นหรือช่องว่างกว้าง (Wide) แทนด้วยไบนารี "1" และส่วนที่แคบ (Narrow) ก็แทนด้วยไบนารี "0" และจะมีช่องว่างแคบ ๆ กั้นระหว่างอักขระแต่ละตัว

ถึงแม้ว่าจะมีทั้งหมด 512 คอมไบเนชั่นของไบนารี 9 บิดก็ตามแต่ที่เราจะใช้นั่นมีเพียง 44 คอมไบเนชั่น ดังนั้นรหัสนี้จะมีการตรวจสอบรหัสประจำตัวของมันเองอยู่ตลอดเวลา เพื่อความถูกต้องปลอดภัยเป็นพิเศษ เราสามารเพิ่มอักขระตรวจสอบลงไปในข้อความแต่ละข้อความ ด้วยการคำนวณหาค่า Check-Sum โดยการบวกค่าตรวจสอบ (Check-Sum) ประจำตัวของอักขระนั้น ๆ ในหนึ่งข้อความและนำผลรวมที่ได้มาหารด้วย 43 ซึ่งจะให้ค่าเศษที่เหลือหนึ่งค่า แล้วนำค่าเศษที่เหลือนี้มาเทียบกับค่า Check-Sum ในตารางที่ 4.3 ก็จะได้อักขระตรวจสอบ (Check Character) ที่จะนำมาเพิ่มต่อท้ายข้อความนั้น

เราสามารถประยุกต์ใช้งานรหัส Code 39 ให้เป็นรหัสที่ซับซ้อนยิ่งขึ้นได้ โดยการใช้อักขระที่แน่นอนหนึ่งอักขระเป็นตัวเริ่มต้นของอักขระโดยใช้ 2 อักขระหรือกลุ่มอักขระ เช่น การแทนด้วยรหัสแอสกี้ (ASCII code) รวมถึงรหัสควบคุม (Control Code) ยกตัวอย่างเช่น Carriage Return จะถูกแทนด้วยรหัส $M

จากตารางที่ 4.3 จะแสดงถึงรหัสเลขฐานสอง (Binary Code) ของอักขระแต่ละตัว และค่าอักขระตรวจสอบของมันเอง

หมายเหตุ อักขระ * นั้นไม่มีค่าตรวจสอบ (Check Value) ประจำตัวของมันเอง เพราะว่ามันถูกใช้เป็นตัวแสดงถึงการเริ่มต้นและสิ้นสุดของรหัสข้อความใน Code 39 ทุก ๆ ข้อความ

ตารางที่ 4.3 รหัสเลขฐานสอง (Binary Code)

ตัวอักษร	เลขฐานสอง	Check-Sum	ตัวอักษร	เลขฐานสอง	Check-Sum
0	00110100	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอักษร	เลขฐานสอง	Check-Sum	ตัวอักษร	เลขฐานสอง	Check-Sum
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

ตัวอย่างการเข้ารหัสบาร์โค้ด Code 39

กำหนดให้รหัสของข้อความเป็น 98PQ

- ขั้นแรกหาผลรวมของค่า check-sum ของอักขระทุก ๆ ตัวในข้อความนั้น คือ 98PQ (9+8+25+26 = 68)
- หารผลรวมที่ได้ด้วย 43 (68 / 43) ก็จะได้เท่ากับ 1 กับเศษที่เหลืออีก 25
- ต่อมาให้ไปดูตารางที่ 1 ว่าอักขระที่ได้คือ ตัว P
- ดังนั้นข้อความก็จะถูกแปลงเป็นรหัส Code 39 รวมทั้ง Check-Sum และอักขระเริ่มต้นและสิ้นสุดด้วย ดังนี้ *98PQP*
- จากนั้นนำข้อความที่ได้มาแปลงเป็นรหัสไบนารีจะได้ดังนี้

  98PQP $\Rightarrow$ 010010100/0/001100100/0/100100100/0/001010010/0/000000111/0/0100 10100
- แล้วนำเลขไบนารีที่ได้มาแทนด้วยแถบเส้นหรือช่องว่าง โดยให้ไบนารี "0" แทนด้วยแถบเส้นหรือช่องว่างที่แคบ (Narrow) และไบนารี "1" แทนด้วยส่วนที่กว้าง (Wide) ก็จะได้บาร์โค้ดตามความต้องการ

หมายเหตุ อักขระแต่ละตัวในข้อความหนึ่ง ๆ จะถูกแยกจากกันด้วยช่องว่างแคบ ๆ (Narrow Space) ซึ่งมีค่าไบนารีเป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 Code 39 เป็นบาร์โค้ดที่ใช้กันอย่างกว้างขวาง โดยเฉพาะสินค้าขายปลีกทั่วไป

4.4.4. ชนิดรหัสโคดาบาร์ (Codabar)

รหัส Codabar ประกอบด้วย 7 บิต โดย 7 บิต เป็นแถบดำ และ 3 บิต เป็นช่องว่างใช้แทนตัวเลข 0-9 เครื่องหมาย -$: . + A B C และ D รหัส Codabar ที่สมบูรณ์จะต้องมีรหัสที่ใช้แทนตัวอักษร A B C และ D (เช่น A = 0011010) เป็นส่วนเริ่มต้นหรือสิ้นสุด ภายในประกอบด้วยรหัสของ Codabar ที่เป็นตัวเลขและเครื่องหมายซึ่งทำให้มีความยาว ไม่แน่นอน เพราะ 12 รหัสแรกมีบิตที่เป็น 1 อยู่ 2 บิต 4 รหัสสุดท้ายเป็นรหัสของ A,B,C,D กำหนดขึ้นมาเพื่อใช้เป็นรหัสเริ่มต้นและสิ้นสุด (Start/Stop Code)

รูปที่ 4.8 รหัสแถบชนิด Codabar

จากรูปจะเห็นได้ว่ารหัสแถบชนิดนี้แบ่งเป็น 2 ส่วน ซึ่งถูกแบ่งด้วยแถบสีดำเล็ก ๆ แต่ยาวกว่าแถบอื่น 2 แถบ คั่นอยู่ตรงกลาง (เลขรหัสฐาน 2 ของแถบคั่นกลางเป็น 0100010) และยังมีแถบลักษณะเดียวกัน 2 ชุด อยู่ทางซ้ายขวาสุด (เลขรหัสฐาน 2 ของแถบนี้คือ 101) แถบทั้ง 3 ชุดนี้ เรียกว่า Guide Bar ซึ่งปกติจะมีความยาวกว่าแถบอื่น ๆ เป็นข้อสังเกตทำให้แบ่งรหัสแถบเป็น 2 ส่วน คือ โซนทางซ้ายและโซนทางขวา หลักสุดท้ายทางซ้ายขวาสุดเป็นตัวตรวจสอบความถูกต้อง (Check Digit) ซึ่งคำนวณมาจากหลักที่เหลือ โดยตัวตรวจสอบทางซ้ายสุดมาจากเลข 5 หลักทางซ้าย และหลักทางขวามาจากเลข 5 หลักที่อยู่ด้านขวา ซึ่งแถบสำหรับตรวจสอบนี้บางครั้งก็พิมพ์ยาวเท่ากับส่วนที่เป็น Guide Bar

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.4 เลขฐาน 2 ของรหัส Codabar

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

4.4.5. ชนิดรหัส UPC หรือ EAN Code (Universal Product Code or European Aritcle Numbering)

เป็นรหัสที่นิยมใช้กับสินค้าที่มาจากต่างประเทศหลายประเภท หรือสินค้าที่ส่งออกไปขายต่างประเทศโครงสร้างของรหัสชนิดนี้ต่างจากรหัสแถบประเภทอื่น ๆ โดยสิ้นเชิง

รหัสแถบชนิดนี้ใช้เป็นต้นแบบในการพัฒนาบาร์รหัสแถบ EAN (European Article Numbering ปัจจุบันเรียก International Article Numbering แทน) ซึ่งใช้กับสินค้าอุปโภค/บริโภคในหลายประเทศ ทั่วโลก (รวมประเทศไทยด้วย) ในระบบรหัสแถบแบบ UPC เองยังมีรูปแบบต่าง ๆ อีกหลายรูปแบบ แต่รูปแบบที่ใช้เป็นมาตรฐานมีอยู่ 2 แบบ คือ แบบ UPC-A และ UPC-E

ส่วนประกอบของรหัสแถบแบบ UPC รหัสแถบชนิดนี้ใช้แทนตัวเลขที่มีจำนวนหลักแน่นอน ไม่สามารถใช้แทนตัวอักษรได้เป็นรหัสแถบที่ใช้แทนรหัสสินค้าแบบ 10 หลัก โดย 5 หลักทางด้านซ้ายจะหมายถึงรหัสผู้ผลิตและกลุ่ม 5 หลักทางด้านขวาจะใช้แทนลำดับที่ของชนิดสินค้าที่ผู้ผลิตออกจำหน่าย นอกจากนี้ยังมีตัวเลขเสริมอีก 2 หลัก คือ ตัวเลขนำ (Number System) จะอยู่ในแถบด้านซ้ายก่อนรหัสผู้ผลิต สำหรับตัวเลขสำนี้จะมีค่าตั้งแต่ 0 ถึง 9 ส่วนตัวเลขอีกชุดหนึ่งเป็นตัวเลขสำหรับตรวจสอบความถูกต้องของรหัส (Check Digit) ซึ่งจะอยู่ทางด้านซ้ายถัดจากเลขชุดลำดับของชนิดสินค้าของผู้ผลิต ซึ่งตัวเลขทั้ง 2 หลัก ทางผู้ผลิตจะแสดงค่าตัวเลขประจำหลักบนรหัสแถบหรือไม่ก็ได้ และไม่ว่าจะแสดงตัวเลขดังกล่าวให้คนเห็นหรือไม่ แต่เมื่ออ่านด้วยเครื่องอ่านก็สามารถตรวจสอบค่าทั้งสองได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรหัสของ UPC รหัสทางซ้ายจะใช้กับรหัสแถบแบบ UPC ในโซนทางซ้าย ส่วนรหัสทางขวาจะใช้ได้กับโซนทางขวาของรหัสแถบชนิด UPC เท่านั้น

รูปที่ 4.9 รหัสแถบชนิด UPC/EAN

ตารางที่ 4.5 รหัสเลขฐาน 2 ของ UPC โซนทางซ้ายและโซนทางขวา

ตัวเลข	รหัสโซนซ้าย	รหัสโซนขวา
0	0001101	1110010
1	0011001	1100110
2	0010011	1101100
3	0111101	1000010
4	0100011	1011100
5	0110001	1001110
6	0101111	1010000
7	0111011	1000100
8	0110111	1001000
9	0001011	1110100

จะใช้สลับกันไม่ได้ ในส่วนของรหัสทางซ้ายจะขึ้นต้นด้วยบิต 0 และลงท้ายด้วยบิต 1 เสมอ จะมีการตรวจสอบเป็นแบบบิตคี่ (Odd parity) ส่วนรหัสทางขวาจะกลับกับรหัสทางซ้าย คือ มีบิต 1 เป็นบิตเริ่มต้น และ 0 เป็นบิตสิ้นสุด การตรวจสอบบิตเป็นแบบคู่ (Even Parity) นอกจากนี้จากตารางเลขรหัสทางซ้ายและทางขวายังเป็นเลขแบบ 1's Comprement ซึ่งกันและกัน

รหัส UPC/EAN มีหลายประเภท คือ UPC-A, UPC-B, UPC-C, UPC-D, UPC-E, EAN-8 และ EAN-13 ซึ่งมีความแตกต่างกันไปบ้าง การอ่านรหัสชนิดนี้จะยากกว่าแบบอื่น ถึงแม้ว่าแถบดำมีค่าเป็นบิต 1 และแถบขาวมีค่าเป็นบิต 0 เหมือนแบบอื่นๆ ก็ตามเนื่องจากในแถบดำ-ขาว ที่ใช้ไม่ได้มีค่าแถบกว้าง, ช่องว่างกว้าง, แถบแคบ หรือช่องว่างแคบเท่านั้น ในแถบดำ (Bar) และแถบขาว (Space) ยังแบ่งอย่างละ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขนาด คือ แถบดำแคบสุด มีค่า 1 ขนาดที่ 2 กว้างกว่าขนาดแคบสุดเล็กน้อยมีค่า 11 และขนาดที่ 3 มีค่า 111 ส่วน แถบกว้างสุดมีค่า 1111 ทำนองเดียวกันแถบขาว 4 ขนาดมีค่า 0,00,000 และ 0000 ตามลำดับ แต่ละรหัสตัวเลขจะประกอบด้วยแถบขาว-ดำ อย่างละ 2 แถบ

รหัสแถบแบบ EAN (European Artcle Number) นี้มาจากรหัสแถบหลายแบบหลายประเภท ซึ่งรหัสแถบชนิดนี้จะใช้จำนวนตัวเลข 5 หรือ 10 หลักเป็นรหัสหลัก และเพิ่มได้อีก 2 หรือ 5 หลัก โดยมีเพียงอักขระ 0-9 เท่านั้นที่ใช้แถบแทนด้วยรหัสไบนารี รหัส EAN นี้มี 4 ระดับ (ความกว้าง) คือ ในแต่ละแถบเส้น หรือช่องว่างจะมีระดับความกว้าง 1,2,3 หรือ 4 โดยให้แถบเส้นแทนด้วยไบนารี "1" ในขณะที่ช่องว่างแทนด้วยไบนารี "0" ตัวอย่างเช่น รหัส 00011 ก็จะถูกแทนด้วยช่องว่างที่มีความกว้าง 3 ส่วน และตามด้วยแถบเส้นที่มีความกว้าง 2 ส่วน

อักขระแต่ละตัวถูกสร้างขึ้นด้วยเลขไบนารี 7 บิด โดยรหัสแถบหนึ่ง ๆ จะประกอบด้วยรหัสกั้นหน้ารหัสกั้นกลาง และหัสกั้นหลัง (start , center and end guard bars) รหัสที่อยู่ทางด้านซ้ายของรหัสกั้นกลาง ถูกเข้ารหัสโดยการใช้คอลัมน์ด้านซ้ายมือ ดังแสดงในตารางที่ 2 ดังนั้นอักขระที่อยู่ทางด้านขวามือของรหัสกั้นกลาง ก็จะเข้ารหัสโดยใช้คอลัมน์ด้านขวามือ ความแตกต่างระหว่าง 2 คอลัมน์ทางด้านซ้ายมือนั่นคือ A จะใช้เข้ารหัสข้อมูลกับข้อความที่มีจำนวนเป็นที่คี่ (odd parity) และจำนวนเป็นคู่ (even parity) ก็ใช้คอลัมน์ B

ตารางที่ 4.6 อักขระของรหัส EAN

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

EAN 8 ใช้กับรหัสที่มีความยาว 5 หลัก ส่วนแบบ EAN 13 นั้นใช้กับรหัสที่มีความยาว 10 หลัก รหัส ทั้งสองชนิดนี้ก็ประกอบไปด้วยรหัสกั้นหน้า รหัสกั้นกลาง และรหัสกั้นหลัง , อักขระแฟลก (flag character) 2 ตัว และอาจจะมีรหัส 2-5 หลักเพิ่มขึ้นมาอีก ซึ่งมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัส EAN 8

รหัสแถบแบบ EAN 8 นั้นประกอบไปด้วยส่วนต่าง ๆ ตามลำดับดังนี้

- แถบรหัสกั้นหน้า ซึ่งเข้ารหัสด้วย 101
- อักขระแฟล็ก 2 ตัว เข้ารหัสด้วยคอลัมน์ซ้ายมือ A (ดูตารางที่ 2)
- ข้อมูลอักขระ 2 ตัวแรกเข้ารหัสด้วยคอลัมน์ซ้ายมือ A เช่นกัน
- แถบรหัสกั้นกลาง ซึ่งเข้ารหัสด้วย 01010
- ข้อมูลอักขระ 3 ตัวหลัง เข้ารหัสด้วยคอลัมน์ขวามือ
- อักขระตรวจสอบ ซึ่งเข้ารหัสด้วยคอลัมภ์ขวามือเช่นกัน
- แถบรหัสกั้นหลัง เข้ารหัสด้วย 101

รหัส EAN 13

ส่วนรหัสแถบแบบ EAN 13 นั้นก็มีส่วนประกอบคล้ายคลึงกันดังนี้

- แถบรหัสกั้นหน้า ซึ่งเข้ารหัสด้วย 101
- อักขระแฟล็กที่สอง (second flag character) 1 ตัว เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B
- ข้อมูลอักขระ 5 ตัวแรก เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B
- แถบรหัสกั้นกลาง เข้ารหัสด้วย 01010
- ข้อมูลอักขระ 5 ตัวหลัง เข้ารหัสด้วยคอลัมน์ขวามือ
- อักขระตรวจสอบเข้ารหัสด้วยคอลัมน์ขวามือเช่นกัน
- และแถบรหัสภั้นหลัง เข้ารหัสตารางที่ 3 รหัสสำหรับอักขระแฟล็กที่หนึ่งของ EAN 13 ด้วย 101

ตารางที่ 4.7 แฟล็กและคาต้า

NO	แฟล็ก 2	คาต้า 1	ดาต้า 2	ดาต้า 3	ดาต้า 4	ดาต้า 5
0	A	A	B	B	A	B
3	A	A	B	B	B	A
4	A	B	A	A	B	B
5	A	B	B	A	A	B
6	A	B	B	B	A	A
7	A	B	A	B	A	B
8	A	B	A	B	B	A
9	A	B	B	A	B	A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักขระแฟล็กที่หนึ่งของรหัส EAN 13 ถูกเข้ารหัสด้วยการใช้พาริตี้แพ็ทเทอน (parity pattern) ของอักขระแฟลกที่สอง ข้อมูลอักขระ 5 ตัวแรก (first five data character) ดังตารางที่ 3

รหัสบาร์โค้ต EAN 8 ซึ่งมีข้อความเป็น 80123453 สามารถแทนด้วยเลขไบนารีได้ดังนี้ 101/0110111/0001101/0011001/0010011/01010/1000010/1011100/1001110/100010/101 (เพื่อให้ดูง่าย ๆ เราจะใช้เครื่องหมาย "/" คั่นระหว่างรหัสหรืออักขระแต่ละตัว)

อักขระตรวจสอบสามารถหาได้ โดยการสมมติว่าตัวอักขระขวาสุดเป็นตำแหน่งคี่ (odd) ➔ EOEOEO แล้วบวกอักขระทั้งหมดในตำแหน่งคี่ (odd) และคูณด้วย 3 ได้เป็นผลลัพธ์แรก ส่วนผลลัพธ์ที่สองหาจากผลรวมของอักขระทั้งหมดในตำแหน่งคู่ (even) ดังนั้นผลลัพธ์ที่ได้คือ ผลรวมของทั้งสองกรณีข้างต้น

อักขระตรวจสอบคือ จำนวนเลขที่น้อยที่สุดที่บวกเข้ากับผลลัพธ์ (ดังกล่าวแล้วในข้างต้น) แล้วสามารถหารจำนวนนั้นด้วย 10 ได้ลงตัว (หรือบวกให้หลักหน่วยเป็นศูนย์)

ตัวอย่างรหัส EAN 13

ตัวอย่างรหัส EAN 13 กำหนดให้อักขระแฟล็กคือ 97 และมีข้อความเป็น 7095983300 ดังนั้นเมื่อรวมกันแล้วจะได้ข้อความเป็น 9770959983300 ตำแหน่งคู่/คี่ (even/odd) เป็นดังนี้ EOEOEOEOEOEO ดังนั้งผลรวมของเลขในตำแหน่งคี่คูณด้วย 3 ได้เท่ากับ 69 [(7+0+5+8+3+0)*3] และผลรวมของเลขในตำแหน่งคู่เท่ากับ 37(9+7+9+9+3+0) จากนั้นนำผลมารวมทั้งสองข้างต้นมาบวกกันได้เป็นผลลัพธ์เท่ากับ 106 (69+37) ดังนั้นเมื่อนำ 4 บวกเข้ากับ 106 จะได้เท่ากับ 110 ซึ่งหารด้วย 10 ได้ลงตัวพอดี ฉะนั้นอักขระตรวจสอบคือ 4 และได้ข้อความเต็ม ๆ ดังนี้ 9770959833004

รหัสเพิ่ม 2 หลัก

รหัสแถบ 2 หลักที่เพิ่มขึ้นมา (Two Digit Supplement) ซึ่งอยู่ด้านหน้าของรหัสแถบหลักนั้นจะแสดงหมายเลขเดือน โดยเริ่มจาก 01 สำหรับเดือนมกราคม (January)

รหัส 2 หลักที่เพิ่มมานี้ จะประกอบด้วยส่วนต่าง ๆ ตามลำดับ ดังนี้

- แถบรหัสกั้นข้าง (Guard Bars) เข้ารหัสด้วย 1011
- ข้อมูลอักขระตัวแรก เข้ารหัส

รูปที่ 4.10 EAN 13 ใช้กับข้อความที่มีความยาว 10 หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 พาริตี้แพตเทิร์นของ 2 หลักที่เพิ่มมาของ EAN

A-A	A-B	B-A	B-B
00	01	02	03
04	05	06	07
08	09	10	11
12	13	14	15
16	17	18	19
20	21	22	23
24	25	26	27
26	29	30	31
32	33	34	35
36	37	38	39
40	41	42	43
44	45	46	47
48	49	50	51
52	53	54	55
56	57	58	59
60	61	62	63
64	65	66	67
68	69	70	71
70	73	74	75
76	77	78	79
80	81	82	83
84	85	86	87
88	89	90	91
92	93	94	95
96	97	98	99

ด้วยคอลัมน์ซ้ายมือ A หรือ B

- อักขระแยก (Character Delineator) เข้ารหัสด้วย 01
- ข้อมูลอักขระตัวที่สอง เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอักขระที่ถูกเข้ารหัสโดยใช้คอลัมน์ซ้ายมือ A หรือ B นั้น ขึ้นอยู่กับหลักของอักขระนั้น ตัวอย่าง เช่น ถ้าส่วนที่เพิ่มขึ้นมานั้นคือ 13 เราจะใช้ตารางที่ 4 ในการอ้างอิง ซึ่งจะได้คอลัมน์ซ้ายมือ A ใช้สำหรับเลข 1 และ 3 ก็จะเข้ารหัสจากคอลัมน์ซ้ายมือ B

(เลข 13 อยู่ในคอลัมน์ A-B)

รูปที่ 4.11 ตัวอย่างรหัสแถบ 5 หลัก ที่เพิ่มมาของ EAN

รหัสเพิ่ม 5 หลัก

ส่วนรหัสแถบที่เพิ่มขึ้นมา 5 หลัก (five digit supplement) นั้น ส่วนมากมักจะพบเห็นกันบนหนังสือหรือนิตยสารที่ป้ายบอกราคาหรือปกหนังสือ ใน 5 หลักที่เพิ่มมานั้นประกอบไปด้วยส่วนต่าง ๆ ตามลำดับดังนี้

- แถบเส้นด้านข้าง ซึ่งเข้ารหัสด้วย 1011
- ข้อมูลอักขระตัวแรก เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B
- อักขระแยก เข้ารหัสด้วย 1
- ข้อมูลอักขระตัวที่สอง เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B
- อักขระแยก เข้ารหัสด้วย 1
- ข้อมูลอักขระตัวที่สาม เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B
- อักขระแยก เข้ารหัสด้วย 01
- ข้อมูลอักขระตัวที่ห้า เข้ารหัสด้วยคอลัมน์ซ้ายมือ A หรือ B

เช่นเดียวกันตัวอักขระจะถูกเข้ารหัสโดยการใช้คอลัมน์ซ้ายมือ ซึ่งคอลัมน์ที่ใช้นั้นคิดจากค่า Check Sum คิดเหมือนกับอักขระตรวจสอบของข้อความหลัก โดยสมมติให้อักขระด้านขวาสุดเป็นตำแหน่งคี่ จากนั้นนำอักขระที่อยู่ในตำแหน่งคี่มาบวกกันแล้วคูณด้วย 3 และนำอักขระที่อยู่ตำแหน่งคู่มาบวกกันแล้วคูณด้วย 9 เสร็จแล้วนำผลลัพธ์ทั้งสองข้างต้นมาบวกกัน ก็จะได้ค่า ๆ หนึ่ง โดยเราจะสนใจเฉพาะเลขในหลักหน่วยของค่าผลลัพธ์ที่ได้ ซึ่งก็คือหมายเลขของพาริตี้แพตเทิร์นในตารางที่ 4.9

ตัวอย่างเช่น ใน 5 หลักที่เพิ่มมาคือ 12345 มีตำแหน่งคู่-คี่ คือ OEOEO จะมีผลรวมของตำแหน่งคี่ทั้งหมดคูณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.9 พาริตี้แพตเทิร์นของ 5 หลัก ที่เพิ่มม่าของ EAN 13

NO	ดาต้า 1	ดาต้า 2	ดาต้า 3	ดาต้า 4	ดาต้า 5
0	B	B	A	A	A
1	B	A	B	A	A
2	B	A	A	B	A
3	B	A	A	A	B
4	A	B	B	A	A
5	A	A	B	B	A
6	A	A	A	B	B
7	A	B	A	B	A
8	A	B	A	A	B
9	A	A	B	A	B

ด้วย 3 เท่ากับ 27 และผลรวมของตำแหน่งคู่คูณด้วย 9 เท่ากับ 54 เสร็จแล้วนำผลลัพธ์ทั้งสองมารวมกันได้เท่ากับ 81 (27+54) ดังนั้นหมายเลข (No.) ของพาริตี้แพตเทิร์นคือ หลักหน่วย ซึ่งก็คือ 1 นั่นเอง ฉะนั้นเราก็สามารถรู้ได้ว่าจะใช้คอลัมน์ซ้ายมือ A หรือ B ในการเข้ารหัสโดยดูตารางที่ 4.9 หมายเลข 1 (ตามตัวอย่าง) ซึ่งคอลัมน์ที่ใช้เข้ารหัสข้อมูลที่ 1-5 ได้เรียงตามลำดับดังนี้ BABAA

นอกจากนี้ยังมีรหัสแถบชนิดอื่นที่พัฒนาขึ้นมาอีกเพื่องานเฉพาะด้านในแต่ละกิจการ เช่น รหัส 11, รหัส Plessley, รหัส AMES, รหัส Nixdoef, รหัส Jan, รหัส 128 Subset C เป็นต้น แต่อย่างไรก็ตาม ลักษณะของรหัสแถบที่ดี ควรมีคุณสมบัติดังนี้

- สามารถตรวจสอบความถูกต้องภายในรหัสได้
- ความกว้างและจำนวนของแถบต่อรหัสควรจะคงที่
- สามารถใช้แทนตัวเลขหรือตัวเลขปนตัวอักษรได้ครบ
- มีโครงสร้างแบบง่าย ๆ
- การอ่านด้วยความเร็วที่ต่างกันควรได้ค่าที่ถูกต้องเสมอ
- มีความหนาแน่นของข้อมูลต่อความกว้างของแถบสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 รหัสแถบ ANSI

Barcode	Elements	Element Width Tolerance T	Wide To Arrow Element Ratio N
Enterleaved 2 of 5	Narrow Bar W Narrow Space W Wide Bar Wide Space	$=\frac{(18N-20)}{80}W$	2:1 To 3:1 (Must Exceed 2.2:1 When Ever Narrow Element<0.02-1N Wide)
3 of 9	Same As Above	$=\frac{4}{27}(N-\frac{2}{3})W$	Same As Above
Codabar	9 Bar Widths 10 Space Widths	$=\frac{0.0015}{0.0065}\times\frac{Element}{Width}$	Does Not Apply

4.5 การพิมพ์รหัสแถบ

หลังจากทำความรู้จักกับรหัสแถบมาพอสมควรแล้วเราจะเห็นได้ว่า เราสามารถเลือกชนิดของรหัสแถบ มาใช้งานตามความต้องการได้ทุกแบบ แต่จะมีแบบ UPC เท่านั้น มีค่อนข้างยุ่งยาก เพราะนอกจากคุณสมบัติเฉพาะแล้วยังต้องคำนึงถึงการใช้รหัสตัวเลขแทนค่าด้วยเช่น รหัสประเทศผู้ผลิต รหัสผู้ผลิต และรหัสประเภทสินค้า ที่ได้กำหนดไว้คงที่เป็นมาตรฐานทั่วโลก ซึ่งผู้ใช้จะต้องศึกษาให้เข้าใจก่อนจะนำมาใช้งาน ส่วน Codabar มีความกว้างของแถบรหัส เป็น 2 แบบ และรหัสทั้งหมดก็สามารถใช้แบบ 3 of 9 แทนได้

4.6 ลักษณะมาตรฐานของรหัสแถบ

ลักษณะมาตรฐานของรหัสแถบนี้สามารถนำมาพิจารณากับรหัสแถบต่าง ๆ ได้ทุกแบบซึ่งจะมีลักษณะร่วมที่เหมือนกันดังนี้

1. สัญลักษณ์รหัสแถบจะเป็นรูปสี่เหลี่ยมผืนผ้าทั้งหมด สร้างขึ้นจากชุดของแท่งขนานสีอ่อนและเข้มวางสลับกัน ซึ่งตั้งฉากอยู่บนแนวฐานด้านล่าง โดยมีบริเวณ Light Margin รอบทุกด้าน

2. แท่งสีอ่อนและแท่งสีเข้ม ของสัญลักษณ์รหัสแถบประกอบขึ้นจากหลาย ๆ แท่ง (แท่งสีเข้มและแท่งสีอ่อน) เรียกว่า 'Module' (ขนาดของ Module มีความกว้าง = 0.33 มิลลิเมตร) ซึ่งคำว่าแท่งหรือ Module จะมีความหมายเดียวกัน และจะแทนค่าของส่วนที่เป็น Module สีเข้มด้วยหมายเลข 1 สำหรับค่าของส่วนที่เป็น Module สีอ่อนด้วยหมายเลข 0

3. ตัวเลขแต่ละหลักจะถูกแทนด้วยรหัสแถบ โดยแต่ละอักขระ (Character) เกิดขึ้นจาก 7 Module รวมกันทั้งสีอ่อนหรือสีเข้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ใน อักขระ เหล่านี้ Module ต่าง ๆ จะถูกรวมเข้ากันเป็นแถบ ตัวเลขในแต่ละหลักจะแทนด้วยแท่งสีเข้ม 2 แท่งและแท่งสีอ่อน 2 แท่ง

5. ในแท่งสีเข้ม 1 แท่ง หรือ แท่งสีอ่อน 1 แท่ง อาจประกอบขึ้นจากแท่งที่มีความกว้างตั้งแต่ 1 ถึง 4 Module

6. นอกเหนือจากเครื่องหมายแทนรหัสตัวเลขแล้ว ยังมีเครื่องหมายเสริม (Auxiliary Character) ซึ่งประกอบขึ้นด้วยแท่งจำนวนน้อยกว่า 7 แท่ง ใช้เป็นหน่วยกั้นด้านข้าง (Guard Bar) และหน่วยกั้นกลาง (Center Bar) สำหรับเป็นตัวกำหนดการเริ่มต้น การสิ้นสุด และการแบ่งรหัส

7. สัญลักษณ์นี้ถูกออกแบบมา เพื่อให้อ่านได้ทุกทิศทางด้วยเครื่องอ่านแบบติดตั้งอยู่กับที่ ทั้งยังสามารถอ่านแบบทิศทางเดียวจากเครื่องอ่านแบบมือถือหรือแบบปากกาแสง

8. ขนาดของสัญลักษณ์ยืดหยุ่นได้ภายในขอบเขตของกำลังขยาย เพื่ออำนวยความสะดวกในด้านการผลิตให้ได้ตามคุณภาพจากการพิมพ์ระบบต่าง ๆ

9. ขนาดกว้างยาวของสัญลักษณ์จะถูกกำหนดขึ้นเฉพาะสำหรับแต่ละขนาดของสัญลักษณ์เรียกว่า ขนาดตามกำหนด (Nominal Size) ขอบเขตกำลังขยาย (Magnification) ทำได้ตั้งแต่ขนาด 0.8 ถึง 2.0 เท่าของขนาดตามกำหนด

4.7 การกำหนดชุดตัวเลข

ค่าตัวเลขจะถูกแปลงเป็นรูปสัญลักษณ์รหัสแถบประกอบด้วยอักขระแบบ 7 Module (แทนค่าแท่งสีเข้มด้วย 1 แท่งสีอ่อนด้วย 0) ซึ่งประกอบด้วย 7 Module ที่จัดเรียงเป็นตัวเลขชุดต่าง ๆ ในชื่อของชุด A ชุด B และชุด C ดังนี้

ตารางที่ 4.11 การกำหนดชุดตัวเลข

ค่าของหมายเลข ในรหัสแถบ	แทนค่าหมายเลข ในชุด A	แทนค่าหมายเลข ในชุด B	แทนค่าหมายเลข ในชุด C
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

อักขระ ในชุดตัวเลข A ประกอบด้วย Module สีเข้ม ที่มีจำนวนเป็นเลขคี่ ซึ่งเรียกอักขระ เหล่านี้ว่า Odd Parity

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อักขระ ในชุดตัวเลข B และ C ประกอบด้วย Module สีเข้ม ที่มีจำนวนเป็นเลขคู่ ซึ่งเรียก อักขระเหล่านี้ว่า Even Parity

อักขระ ในชุดตัวเลขข้างต้น ได้ขยายให้เห็นในแผนผัง

อักขระ ในชุดตัวเลข A และ B จะเริ่มต้นจากด้านซ้ายด้วย Module สีอ่อน และสิ้นสุดลงทางด้านขวาด้วย Module สีเข้ม

เครื่องหมายในชุดตัวเลข C จะเริ่มต้นจากด้านซ้านด้วย Module สีเข้ม และจบลงทางด้านขวาด้วย Module สีอ่อน เมื่อประกอบเข้ากับหน่วยกั้นด้านข้าง (Guard Pattern) และหน่วยกั้นกลาง (Centre Pattern) ก็จะเห็นได้ว่าทุก ๆ เครื่องหมายในหนึ่งสัญลักษณ์ เริ่มต้นและจบลงด้วย Module สีที่ต่างกันสีอ่อนหรือสีเข้มจากตัวเครื่องหมายที่อยู่ต่อไปไม่ว่าทางด้านซ้ายหรือด้านขวาลักษณะนี้หมายความว่าจะสามารถแยกอักขระแต่ละตัวออกจากกันอย่างเห็นได้ชัดด้วยตาเปล่า ซึ่งจำเป็นสำหรับการถอดรหัส

4.8 เครื่องหมายเสริม (Auxiliary Characters)

เครื่องหมายเสริมเป็นรหัสแถบที่ไม่ใช่เลขรหัสในสัญลักษณ์รหัสแถบ ทำหน้าที่เป็นหน่วยกั้นด้านข้าง (Guard Pattern) และหน่วยกั้นกลาง (Centre Pattern) ดังนี้

ตารางที่ 4.12 เครื่องหมายเสริม

เครื่องหมายเสริม	จำนวน Module	ชุดของ Module
หน่วยกั้นด้านข้างแบบปกติ	3	101
หน่วยกั้นกลาง	5	01010
หน่วยกั้นด้านขวาแบบ 'E'	6	010101

4.9 ขนาดความยาวตามกำหนดของอักขระ

ในขนาดตามกำหนด (เมื่อกำลังขยาย = 1.0) ค่าความกว้างตามทฤษฎีของหนึ่ง Module คือ 0.33 มม. ความกว้างของอักขระแต่ละชุด คือ

- ทั้งชุดของอักขระ : 7 Module = 2.31 มม.
- Auxiliary Character

หน่วยกั้นด้านข้าง : 3 หน่วย = 0.99 มม.

หน่วยกั้นกลาง : 3 หน่วย = 1.65 มม.

หน่วยกั้นด้านขวาแบบ 'E' : 3 หน่วย = 1.98 มม.

ขนาดกว้างยาวตามทฤษฎีในอุดมคติสำหรับอักขระทุกตัวมีขนาดตามกำหนดต่าง ๆ

หมายเหตุ 1 : ขนาดกว้างยาวภายในสำหรับอักขระที่เป็นเลข 1, 2, 7, และ 8 มีขนาดยืดหยุ่นเล็กน้อยไม่ตรงตามผลคูณของความกว้าง 0.33 มม. ซึ่งไม่ใช่ข้อผิดพลาด ขนาดของอักขระเหล่านี้จะถูกย่อหรือขยายได้ไม่เกิน 1/13 เท่าของหนึ่ง Module เพื่อให้รักษาความแม่นยำในการอ่านของเครื่องสแกนเนอร์ได้

หมายเหตุ 2 : ความกว้างของอักขระตัวเลขสามารถวัดจากขอบที่มองเห็นด้านหนึ่ง (ประกอบด้วย Module เข้ม) ไปจนถึงของของอักขระตัวถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ 3 : ความกว้างของ Auxiliary Character ใช้วัดจากมุมของ Light Margin ซึ่งมีเครื่องหมายชี้ให้เห็น

หมายเหตุ 4 : ขนาดที่กำหนดขึ้นทุกขนาด ต่างเป็นขนาดกว้างยาวในอุดมคติตามทฤษฎีที่สอดคล้องกับขนาดตามกำหนดของสัญลักษณ์ ขนาดกว้างยาวเหล่านี้ ไม่ได้มุ่งหมายเพื่อที่จะใช้โดยตรงในการเตรียมจัดทำสัญลักษณ์

4.10 แฟกเตอร์ของแสงสะท้อนกลับ และความหนาแน่นของแสงสะท้อน

แฟคเตอร์ของแสงสะท้อนกลับ R คือ อัตราส่วนของการเปลี่ยนแปลงจากแสงสะท้อนกลับ Tr ต่อการสะท้อนกลับของแสงที่เกี่ยวข้องกับ Trs การเปลี่ยนแปลงจากแสงสะท้อนกลับคือ กำลังของการส่งรังสีสะท้อนกลับมาจากวัสดุ และประเมินผลโดยอุปกรณ์พิเศษที่รับแสงสะท้อนนั้น การเปลี่ยนแปลงการสะท้อนกลับของแสงที่เกี่ยวข้องกัน คือ กำลังของการส่งรังสีสะท้อนกลับ โดยการเทียบแสงมาตรฐานจากแมกนีเซี่ยมออกไซด์ หรือ แบเรียมซัลเฟต (R = 100%)

ความหนาแน่นของแสงสะท้อน (D) เท่ากับ : $D = -\log_{10} R$

ความหนาแน่นของแสงสะท้อนที่ต้องการสำหรับรหัสแถบส่วนที่เป็นสีทึบ ขึ้นอยู่กับความหนาแน่นของแสงที่สะท้อนกลับมาจากส่วนที่เป็นพื้นที่สีอ่อน หรือส่วนที่เป็น Module สีอ่อนในสัญลักษณ์

4.11 เงื่อนไขทางเรขาคณิตสำหรับการวัดค่าแสงสะท้อน

การตกของแสงบนสัญลักษณ์จะต้องทำมุมที่ 45^0 ลงกลางสัญลักษณ์และเครื่องรับที่อยู่ตรงข้ามกับวัสดุที่จะรับแสงที่สะท้อนกลับจากวัตถุนั้น ช่องรับแสงควรมีขนาดเส้นผ่านศูนย์กลางกว้าง 0.2 มม.

4.12 เงื่อนไขการแยกเาสีสำหรับการวัดค่าแสงสะท้อน

ตัวอย่างของแสงที่ใช้ยิงบนสัญลักษณ์จะต้องใช้แสงที่มีกำลังการกระจายการแยกเงาสี ซึ่งมีคุณสมบัติดังนี้ :

- แสงจาก CIE ชนิด A ที่ได้จากการใช้หลอดไฟที่มีขดลวดทังสเตนฉายแสงในหลอดก๊าซทำงานท่อุณหภูมิ 2856 K

- เครื่องรับสำหรับเทียบแสงของการเปลี่ยนแปลงสะท้อนกลับ จะต้องมีความไวแสงในการแยกเงาสีตามคุณสมบัติดังนี้ :

เครื่องขยายแสงที่รับได้ขนาด S-4 ตามมาตรฐานที่กำหนดโดย American Joint Electron Devices Engineering Council โดยใช้ฟิลเตอร์มาตรฐานของ Wratten 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การออกแบบ

5.1 การออกแบบวงจร

เครื่องเก็บแถบรหัส (Bar-code) นั้น ประกอบด้วยส่วนต่าง ๆ ดังต่อไปนี้

1) ส่วนไมโครโปรเซสเซอร์หลัก

2) ส่วนของหน่วยความจำ

3) ส่วนของการเชื่อมต่อกับเครื่องอ่านหรัสแถบ

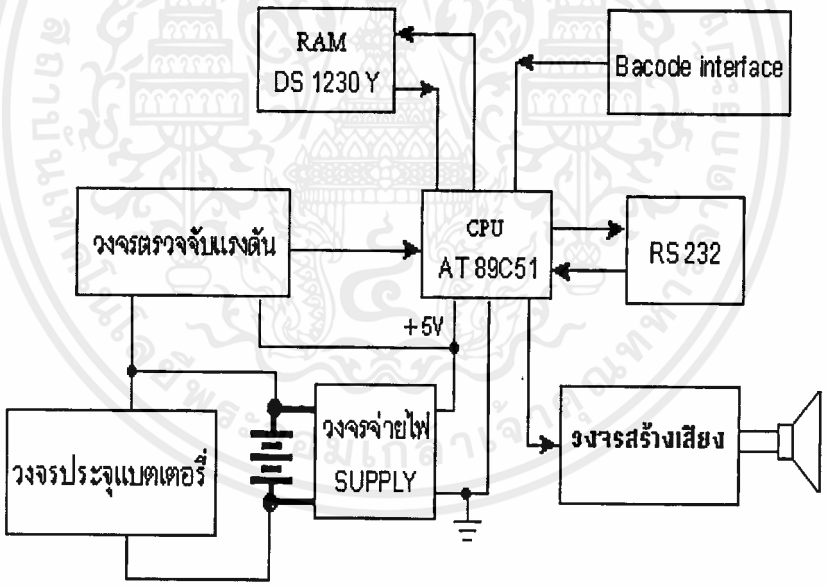
4) ส่วนของการเชื่อมต่อกับคอมพิวเตอร์ (RS- 232)

5) ส่วนของวงจรภาคจ่ายไฟ

6) ส่วนของวงจรตรวจจับแรงดัน

7) ส่วนของวงจรประจุแบตเตอรี่

8) ส่วนของวงจรสร้างเสียง ดังแผนภาพต่อไปนี้

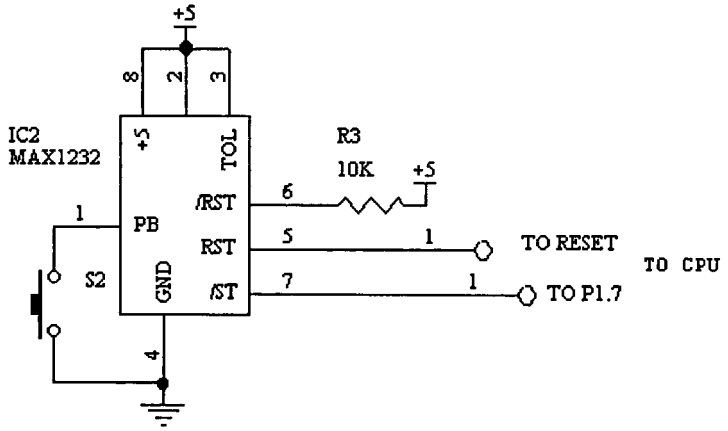


รูปที่ 5.1 แผนภาพแสดงส่วนประกอบของเครื่องเก็บหรัสแถบ

5.1.1 ส่วนไมโครโปรเซสเซอร์หลัก

ไมโครโปรเซสเซอร์หลักจะใช้ควบคุมส่วนต่าง ๆ เข้าด้วยกัน ไมโครโปรเซสเซอร์หลักนี้ใช้เบอร์ AT89C51 ซึ่งเป็นไมโครโปรเซสเซอร์ในตระกูล MCS -51 ใช้สัญญาณนาฬิกาที่ความถี่ 11.059 MHz และยังมีวงจรป้องกันกรณีที่ ไมโครโปรเซสเซอร์ทำงานผิดพลาด ดังรูปที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

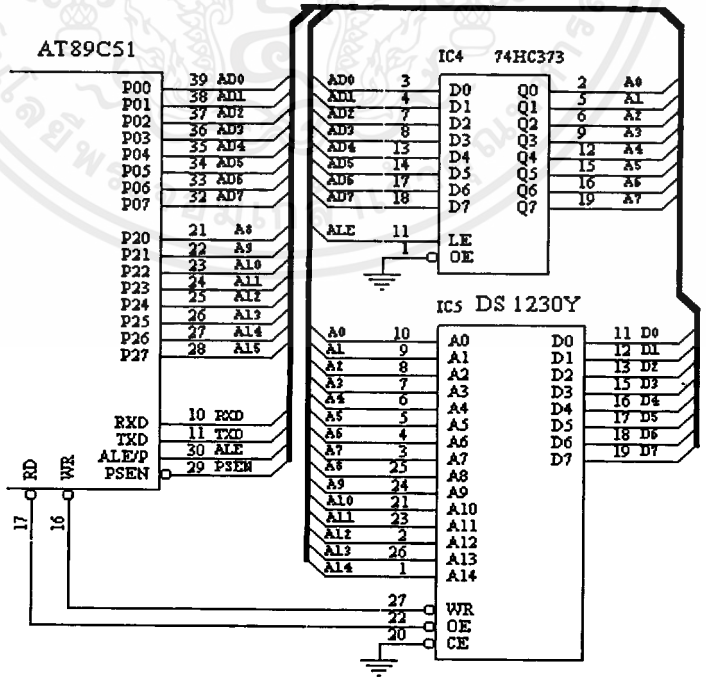


รูปที่ 5.2 วงจรป้องกันความผิดพลาดของ ไมโครโปรเซสเซอร์

5.1.2 ส่วนของหน่วยความจำ

นอนโวลาไทล์แรม เป็นหน่วยความจำที่สามารถอ่านหรือเขียนข้อมูลได้ตลอดเวลา และข้อมูลจะยังคงอยู่ตราบเท่าที่ยังไม่มีการเขียนข้อมูลใหม่มาทับ นอนโวลาไทล์แรมจะถือว่าเป็นสแตติกแรมชนิดหนึ่ง แม้ว่าไม่มีไปเลี้ยงแก่ไอซี ข้อมูลจะไม่สูญหาย ทั้งนี้เพราะโครงสร้างภายในตัวนอนไวลาไทล์แรมนี้จะบรรจุแบตเตอรี่ลิเธียมขนาด +3 โวลต์ และวงจรจ่ายไฟสำรองไว้ แบตเตอรี่ภายในจะมีอายุการใช้งานอย่างน้อยประมาณ 10 ปี

ในวงจรส่วนนี้จะใช้หน่วยความจำขนาด 32K x 8 bits หน่วยความจำตัวนี้ถูกนำมาใช้เป็นที่เก็บรหัสแถบ การต่อหน่วยความจำจะเห็นว่าจะต้องใช้ไอซี่เบอร์ 74LS373 เนื่องจากไมโครโปรเซสเซอร์หลักนี้ใช้เบอร์ AT89C51 นั้นสัญญาณของดาต้าบัสกับแอดแดรสบัสใช้ร่วมกัน ดังนั้นไอซี 74LS373 จึงมีไว้เพื่อแยกข้อมูลระหว่างดาต้ากับแอดแดรสบัสออกจากกันดังรูปที่ 5.3

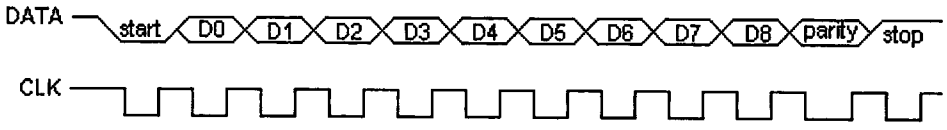


รูปที่ 5.3 การเชื่อมต่อหน่วยความจำ

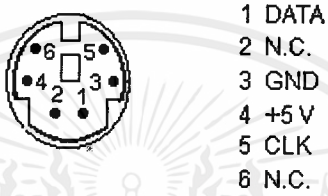
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 ส่วนของการเชื่อมต่อกับเครื่องอ่านหรัสแถบ

เครื่องอ่านรหัสแถบนั้น มีอัตราการส่งข้อมูลคล้ายกับ Keyboard ซึ่งมี 6 ขา การทำงานจะทำงานดังรูปที่ 5.4

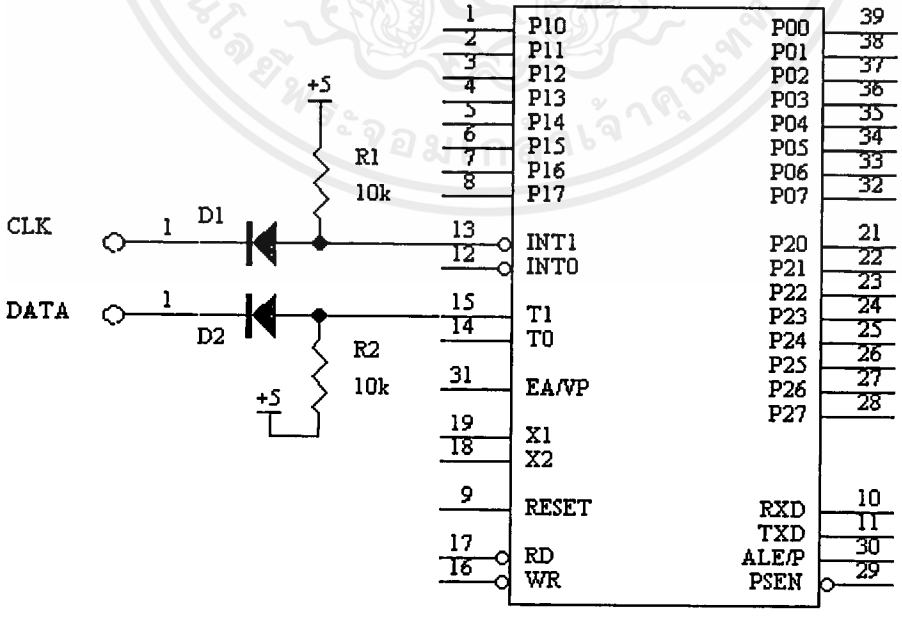


รูป สัญญาณแสดงการส่งข้อมูลของ เครื่องอ่านรหัส บาร์โค๊ต



รูปที่ 5.4 การทำงานของ Keyboard

เราจะเชื่อมต่อวงจรส่วนนี้ผ่าน ไดโอด เพื่อป้องกันการไหลย้อนกลับของกระแสไฟฟ้า ตัวต้านทาน 10 KΩ มีไว้เพื่อยกระดับแรงดัน สัญญาณ DATA ของเครื่องอ่านรหัสแถบจะต่อเข้ากับขา T1 ของ ไมโครโปรเซสเซอร์ สัญญาณ CLK ของเครื่องอ่านรหัสแถบจะต่อเข้ากับขา INT เพื่อให้เกิดการอินเทอรัพต์ ดังวงจรรูปที่ 5.5



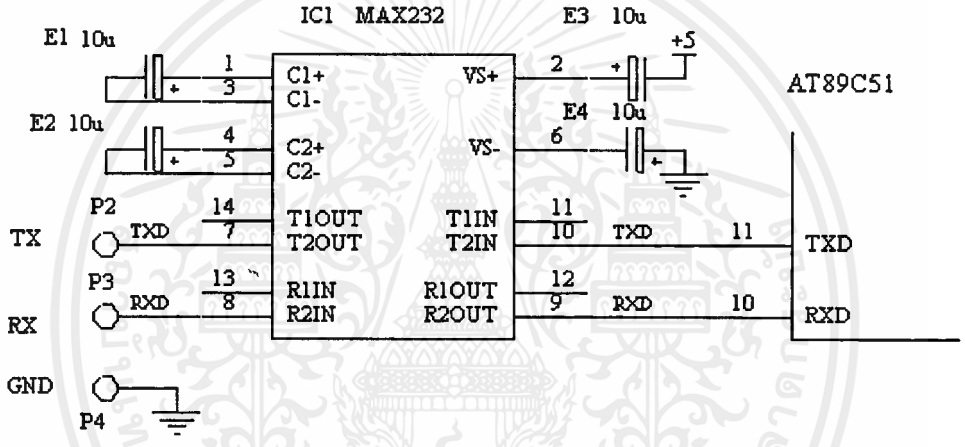
รูปที่ 5.5 วงจรส่วนของการเชื่อมต่อกับเครื่องอ่านรหัสแถบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.4 ส่วนของการเชื่อมต่อกับคอมพิวเตอร์ (RS- 232)

มาตรฐการสื่อสาร RS-232-C กำหนดสัญญาณทางไฟฟ้าไว้ สองสถานะ คือ สถานะที่เป็น '1' จะส่งแรงดันลบออกมา และสถานะที่เป็น '0' จะส่งสัญญาณแรงดันบวกออกมา โดยแรงดันลบกำหนดให้อยู่ระหว่าง -3 ถึง - 15 โวลต์สำหรับอินพุท ส่วนเอาต์พุทนั้น จะอยู่ระหว่าง -5 ถึง - 15 โวลต์ และแรงดันลบกำหนดให้อยู่ระหว่าง +3 ถึง +15 โวลต์ สำหรับอินพุท ส่วนเอาต์พุทนั้นจะอยู่ระหว่าง -5 ถึง - 15 โวลต์ ความแตกต่างนี้เผื่อไว้กรณีที่แรงดันสูญหายเนื่องจากความยาวของสาย RS-232-C ถูกกำหนดไว้ให้มีความยาวสายไม่เกิน 50 เมตร

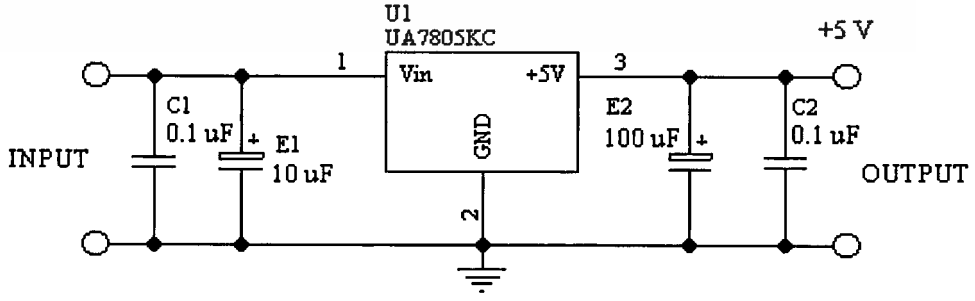
เนื่องด้วยไมโครโปรเซสเซอร์หลักติดต่อสื่อสารแบบอนุกรมจะมี แรงดันใช้งานในช่วง 0 โวลต์ กับ 5 โวลต์เท่านั้น ดังนั้นการสื่อสารอนุกรมของไมโครโปรเซสเซอร์หลักจะต้องมีไอซี MAX 232 ไว้เพื่อปรับเปลี่ยนแรงดันให้ถูกต้องตามมาตรฐการสื่อสาร RS-232-C ดังรูปที่ 5.6



รูป 5.6 การเชื่อมต่อระบบสื่อสาร

5.1.5 ส่วนของวงจรภาคจ่ายไฟ

เนื่องจากวงจรต้องการแรงดันที่คงที่ที่มีขนาด 5 โวลต์ ดังนั้นวงจรภาคจ่ายไฟจึงจำเป็นเพื่อลดแรงดันลงมาให้ได้ขนาด 5 โวลต์คงที่ วงจรภาคจ่ายไฟจึงออกแบบได้ดังรูปที่ 5.7

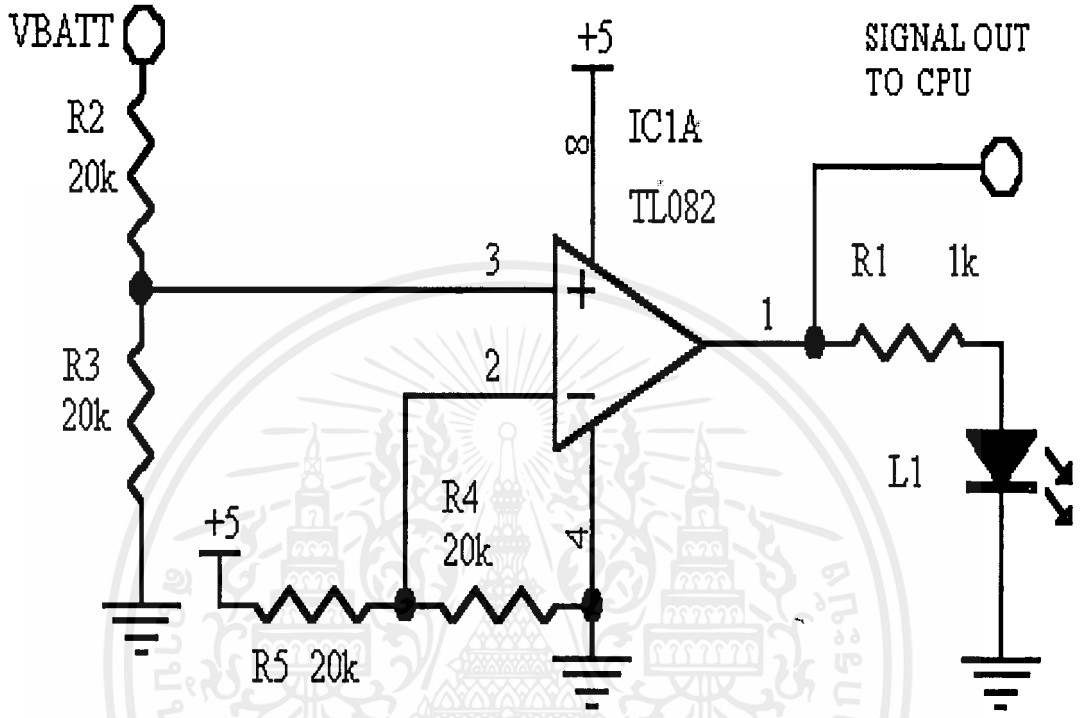


รูปที่ 5.7 วงจรภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.6 ส่วนของวงจรตรวจจับแรงดัน

เนื่องจากภาคจ่ายรับกระแสไฟฟ้าจากแบบเตอรี่ ดังนั้นจึงจำเป็นต้องมีวงจรตรวจจับกรณีแรงดันต่ำกว่า 7.5 โวลต์ วงจรส่วนนี้จะใช้ออปแอมส่วนตัวตรวจจับ โดยจัด Bias ให้ออปแอมทำงานในลักษณะวงจรเปรียบเทียบ และ เอาต์พุทจะส่งไปยัง ไมโครโปรเซสเซอร์หลัก ดังรูปที่ 5.8

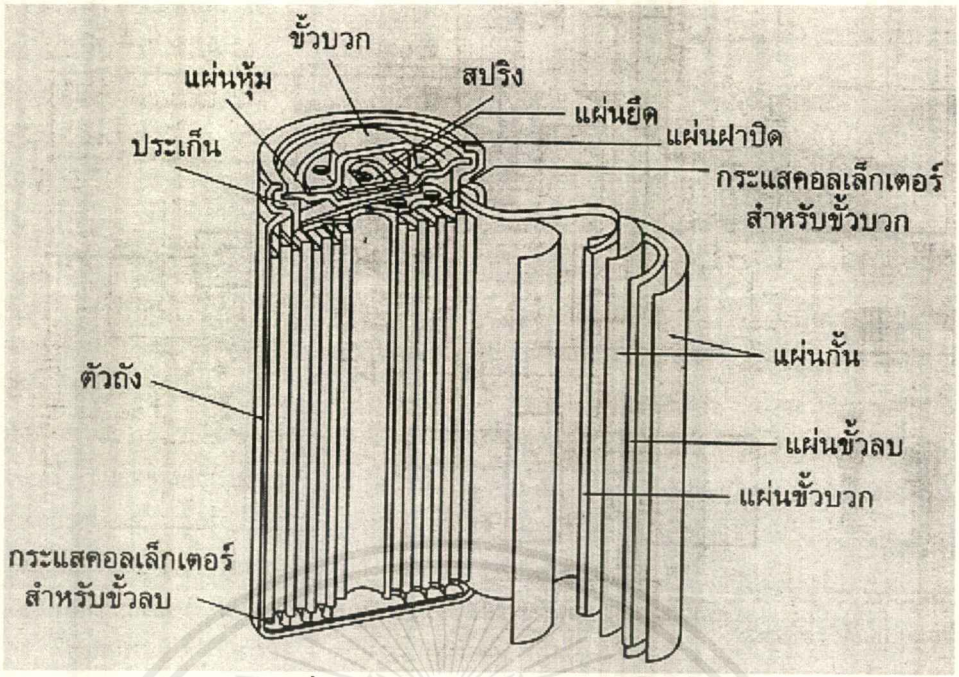


รูปที่ 5.8 วงจรเปรียบเทียบแรงดัน

5.1.7 ส่วนของวงจรประจุแบตเตอรี่

แบตเตอรี่แบบ นิกเกิล-เมตัลไฮไดรด์ (Nickle -Metal Hydriede : Ni -MH) โครงสร้างโดยหลักแล้วก็มีความคล้ายคลึงกับแบตเตอร์รี่นิกเกิล-แคดเมียมแต่จะมีข้อแตกต่างกันบางประการที่เกี่ยวกับสารที่ถูกนำมาใช้เคลือบที่อิเล็กโตรดหรือขั้วของแบตเตอรี่ที่ต้องแตกต่างกันออกไป โครงสร้างของแบตเตอรี่นิกเกิล-เมตัลไฮไดรด์ที่ขั้วบวกจะเป็นสารประกอบของนิเกิลออกไซด์ ขั้วลบเป็นโลหะผสมไฮโดรเจนที่สามารถดูดกลืนไฮโดรเจนได้ และสารอิเล็กทรอไลต์เป็นสารละลายที่เป็นด่างที่มีส่วนประกอบหลักเป็นโปแตสเซียมไฮดรอกไซด์ ดังแสดงโครงสร้างส่วนประกอบหลักเป็นสารละลายที่เป็นด่างที่มีส่วนประกอบหลักเป็นโปแตสเซียม ไฮดรอกไซด์ ดังแสดงโครงสร้างส่วนประกอบในแบตเตอรี่นิกเกิล-เมตัลไฮไดรด์ไว้ใน รูปที่ 5.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 โครงสร้างภายในนิกเกิล-เมตัลไฮ ไดรด์

คุณสมบัติหลักของแบตเตอร์รี่นิกเกิล-เมตัลไฮไดรด์ จะอยู่ที่ปฏิกิริยาในการเก็บและคายประจุของ แบตเตอร์รี่เองกล่าวคือ ขณะทำการประจุเป็นเวลานานเกินไป จะทำให้เกิดปฏิกิริยาทางเคมีขึ้นมาก่อให้เกิดเป็นก๊าซออกซิเจนและก๊าซไฮโดรเจน ซึ่งก๊าซออกซิเจนจะทำปฏิกิริยากับขั้วลบ เป็นผลที่ไม่ดีสำหรับการชาร์จประจุให้กับแบตเตอรี่ แต่คุณสมบัติของโลหะผสมที่ขั้วลบจะทำการดูดกลืนก๊าซออกซิเจนที่เกิดขึ้นมาได้ นอกจากนั้นสารประกอบไฮไดรด์ที่ขั้วลบยังสามารถดูดกลืนก๊าซไฮโดรเจนได้ในปริมาณมากด้วย สำหรับปฏิกิริยาในการคายประจุ ก๊าซไฮโดรเจนจะถูกดูดกลืนอีกครั้งด้วยปฏิกิริยาทางเคมีระหว่างโลหะผสมไฮโดรเจนกับไฮดรอกไซด์อิออน ทำให้ก๊าซไม่สามารถแพร่กระจายออกมาภายนอกได้ ขณะทำการชาร์จประจุและคายประจุ ทำให้มลภาวะไม่เสีย

ลักษณะเด่นของแบตเตอรี่นิกเกิล-เมตัลไฮไดรด์ก็คือไม่มีผลทางหน่วยความจำ (memory effect) ซึ่งเป็นผลของการตกค้างของแรงดันจากการใช้งานที่ยังไม่หมดจริงแล้วนำมาทำการชาร์จประจุใหม่ (ไม่ใช่หน่วยความจำที่ใช้กันในระบบคอมพิวเตอร์) ลักษณะเช่นนี้แล้วเมื่อนำเอาแบตเตอรี่นั้นกลับมาชาร์จประจุใหม่ก็จะทำให้เกิดการชาร์จกลับขั้วแรงดัน ทำให้การชาร์จประจุครั้งนั้นไม่สามารถประจุพลังงานเข้าไปไว้ในแบตเตอรี่ได้อย่างเต็มที่ และยังทำให้เกิดปฏิกิริยาการระเหยเป็นไอของก๊าซขึ้นทำให้อายุการใช้งานของแบตเตอรี่ชนิดนิกเกิล-แคดเมียม แต่สำหรับนิเกิลเมตัลไฮไดรด์ที่ไม่มีผลทางความจำนี้จะไม่เกิดเหตุการ์เช่นนี้ขึ้น

การออกแบบวงจรประจุแบตเตอรี่ ได้แบ่งเป็นส่วนย่อย ๆ ดังนี้ วงจรจ่ายกระแสคงที่ วงจรตั้งเวลา วงตรวจจับแรงดันขณะประจุเกิน วงจรจ่ายกระแสคงที่ใช้ไอซี LM 317T เป็นตัวกำหนดหาได้จากสูตรดังนี้

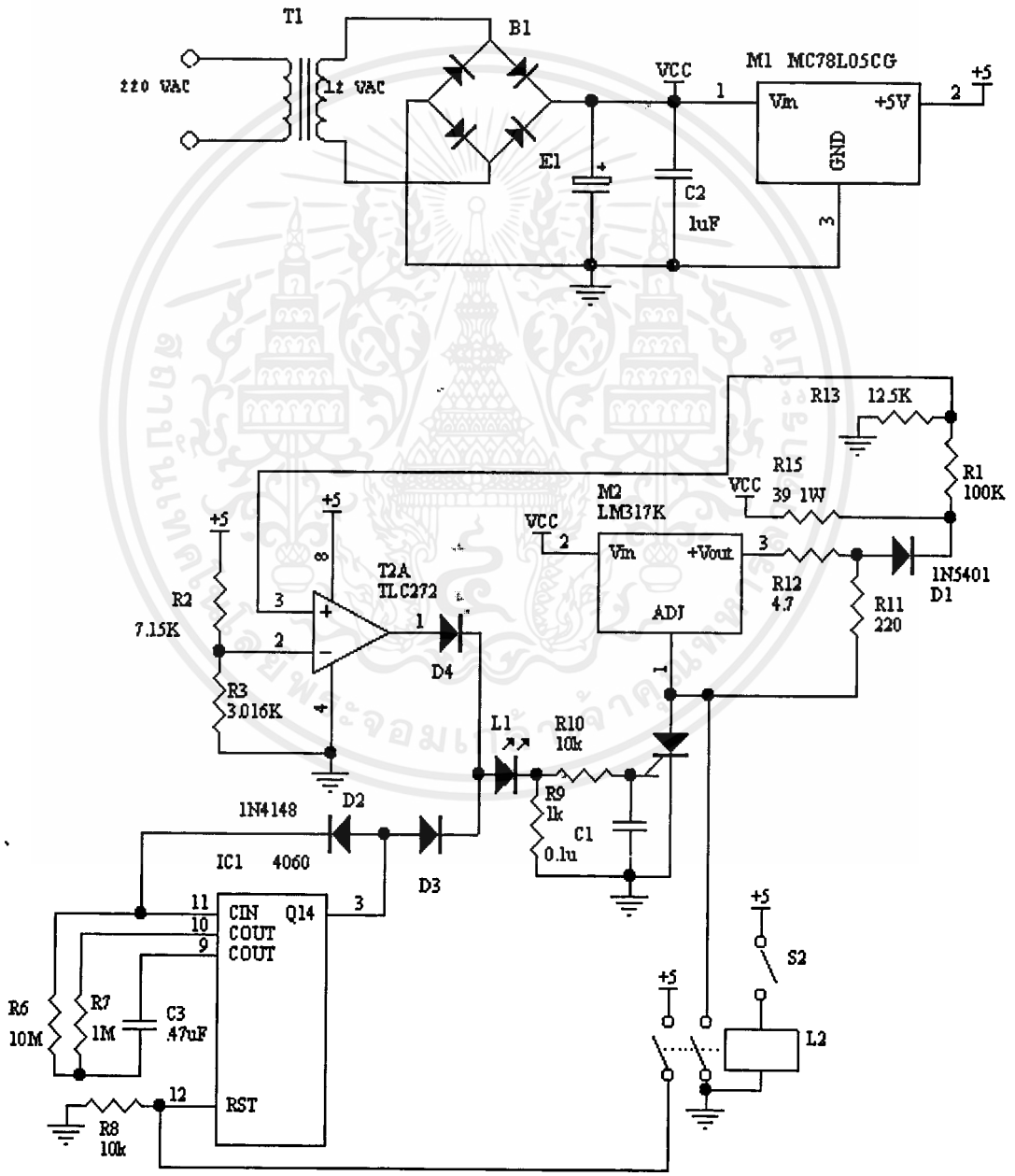
$$\text{กระแส} = 1.25 / R\ 12$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรตั้งเวลาจะใช้ไอซีเบอร์ 4060 ประกอบกับ ตัวต้านทานและคาปาซิเตอร์ เพื่อสร้างสัญญาณนาฬิกาโดยกำหนดการตั้งเวลาไว้ที่ 3.50 ชั่วโมง เอาต์พุทจะออกมายังขา 3 เพื่อกระตุ้น SCR ให้ทำงาน เมื่อ SCR ทำงาน LM 317T ก็จะลดแรงดันลงมาเหลือ 1.25 โวลต์ ทำให้ไม่มีกระแสไปประจุแบตเตอรี่ เพราะแรงดันตรงนี้จะต้องผ่านไดโอด ซึ่งเอาต์พุทของไดโอดมีแรงดันสูงกว่านั่นเอง

วงตรวจจับแรงดันขณะประจุเกิน เมื่อ ขณะประจุแบตเตอรี่ ถ้าแรงดันเกิน 11.92 โวลต์แสดงว่าแบตเตอรี่อยู่พนสภาพเต็ม วงจรนี้ก็จะไปกระตุ้นให้ SCR ทำงานเพื่อหยุดประจุแบตเตอรี่ที่กระแสสูง

วงจรสำเร็จของวงจรประจุแบตเตอรี่นิกเกิล-เมตัลไฮไดรด์ ดังรูปที่ 5.10

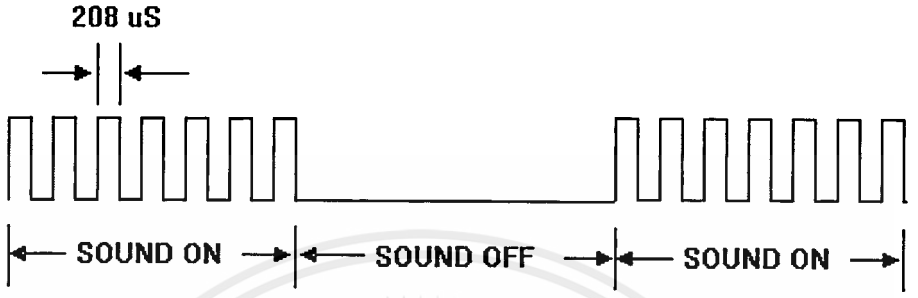


รูปที่ 5.10 วงจรประจุแบตเตอรี่นิกเกิล-เมตัลไฮไดรด์

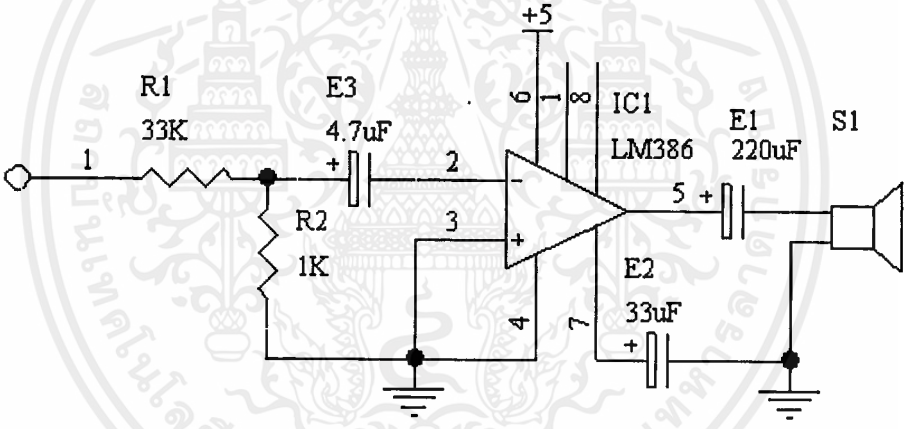
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.8 วงจรสร้างเสียง

การสร้างเสียงจากไมโครโปรเซสเซอร์ใช้หลักการดังนี้ ใช้การสร้างพัลซ์ที่ความถี่ประมาณ 2.4 Khz จากการทดลองจะให้เสียงดังที่สุด จากนั้นนำสัญญาณที่ได้ผ่าน ออปแอมป์ขยายอีกครั้งแล้วจึงส่งออกลำโพง ภาคเสียงจะใช้ในกรณีที่ หน่วยความจำเต็ม และกรณีที่ แบบเตอรี่อ่อนกำลังลง



รูปที่ 5.11 การกำเนิดเสียง



รูปที่ 5.12 วงจรสร้างเสียง

5.2 การออกแบบโปรแกรมเครื่องเก็บรหัสแถบ

โปรแกรมนั้นได้แบ่งออกเป็นส่วนต่าง ๆ ดังต่อไปนี้

1) ส่วนที่ติดต่อสื่อสาร (RS-232-C) เขียนโปรแกรมแบบอินเตอร์รัพต์

2) ส่วนที่รับข้อมูลจากเครื่องอ่านรหัสแถบ เขียนโปรแกรมแบบอินเตอร์รัพต์

3) ส่วนที่ตรวจสอบแบตเตอรี่แรงดันต่ำ เขียนโปรแกรมแบบอินเตอร์รัพต์

4) ส่วนที่สร้างเสียง

5) โปรแกรมหลัก

5.2.1 ส่วนที่ติดต่อสื่อสาร (RS-232-C)

การสื่อสารเนื่องจากต้องการความถูกต้อง และต้องการมีหลายคำสั่ง โดยคำสั่งเหล่านี้จะถูกกำหนดจากคอมพิวเตอร์ ดังนั้นจึงมีการตั้งรหัสสำหรับคำสั่งเกิดขึ้นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

'EOT' 00 'ACK' หมายถึง การทดสอบการติดต่อหรือ ทดสอบสถานะ ไมโครโปรเซสเซอร์จะตอบ 01 'ACK' เมื่อคอมพิวเตอร์ได้รับ 01 'ACK' ก็จะหมายถึงการสื่อสารถูกต้อง

'EOT' 01 'ACK' หมายถึง ทดสอบหน่วยความจำ ภายในไมโครโปรเซสเซอร์ เมื่อคอมพิวเตอร์ได้รับ 00 'ACK' ก็จะหมายถึงหน่วยความจำ ภายในไมโครโปรเซอร์ผิดพลาด เมื่อคอมพิวเตอร์ได้รับ 01 'ACK' ก็จะหมายถึงหน่วยความจำ ภายในไมโครโปรเซสเซอร์ถูกต้อง

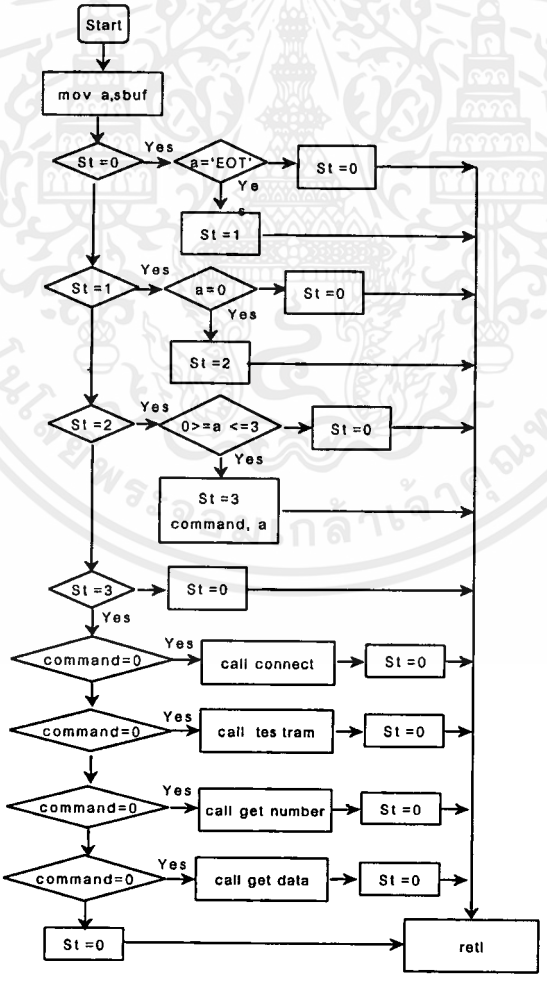
'EOT' 02 'ACK' หมายถึง การขออ่านจำนวนรายการขนาด 4 หลัก และรหัสของรายการขนาด 16 ตัวอักษรนั้น ไมโครโปรเซสเซอร์ก็จะตอบส่งข้อมูลเหล่านั้นออกไปให้ โดยมีรูปแบบดังนี้

XXXX YYYYYYYYYYYYYYYY 'ACK' X แทน 1 ตัวอักษร Y แทน 1 ตัวอักษร XXXX แทนจำนวนรายการรหัสแถบที่เก็บ YYYY YYYY YYYY YYYY แทนข้อมูลหรัสแบ บนั้น

'EOT' 03 'ACK' หมายถึง การขออ่านจำนวนรายการขนาด 4 หลัก ไมโครโปรเซสเซอร์ก็จะตอบส่งข้อมูลเหล่านั้นออกไปให้ โดยมีรูปแบบดังนี้

XXXX 'ACK' X แทน 1 ตัวอักษร XXXX แทนจำนวนรายการรหัสแถบที่เก็บ

โปรแกรมส่วนของการสื่อสารสามารถเขียนเป็นแผนภาพได้ดังนี้



รูปที่ 5.13 การทำงานของโปรแกรม

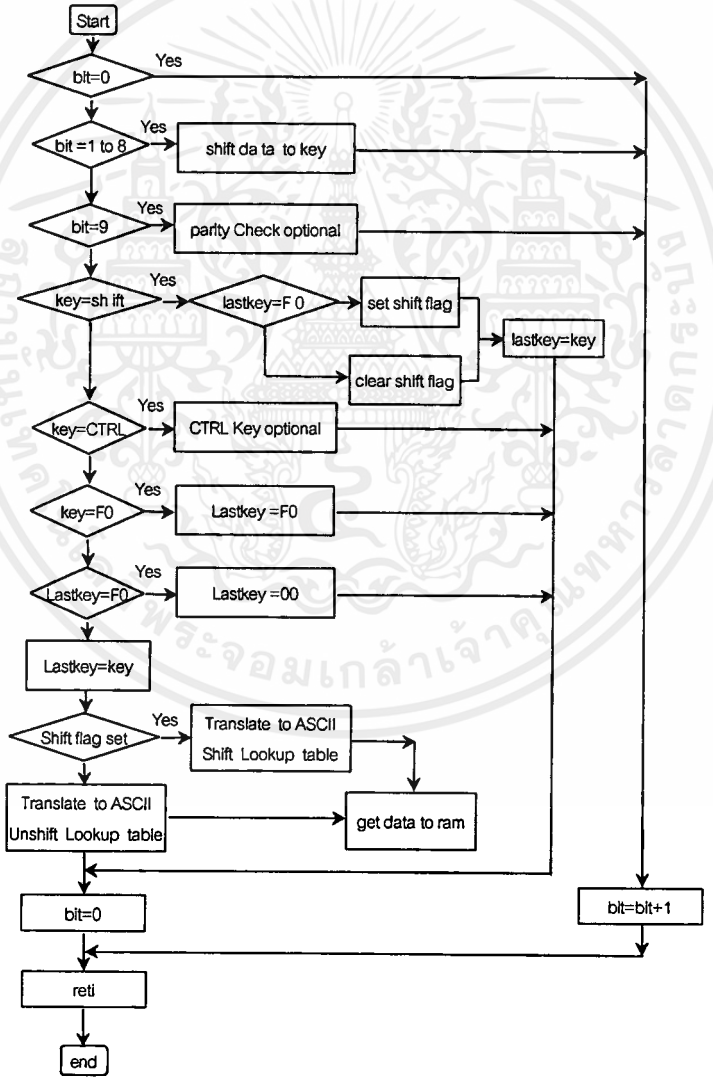
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ

การตรวจสอบหน่วยความจำ ใช้วิธีการเขียนค่า 00 ลงในหน่วยความจำทุกตำแหน่ง แล้วอ่านกลับมาตรวจสอบถ้าไม่เท่ากับค่า 00 แสดงว่าหน่วยความจำมีปัญหา การอ่านข้อมูลทุกครั้งจำนวนข้อมูลจะลดลง 1 รายการเสมอ

5.2.2 ส่วนที่รับข้อมูลจากเครื่องอ่านรหัสแถบ เขียนโปรแกรมแบบอินเตอร์รัพต์

การส่งข้อมูลของเครื่องอ่านรหัสแถบนั้นส่งแบบอนุกรม ดังนั้นการรับข้อมูลของไมโครโปรเซสเซอร์จะต้องรับข้อมูลในระดับบิทและหลังจากได้ข้อมูล 1 ไบต์ แล้วจะต้องมาตรวจสอบข้อมูลนั้นว่าเป็นข้อมูลหรือเป็นการกดหรือปล่อยรหัส SHIFT ,CONTROL และการ BREAK การออกแบบโปรแกรมแสดงเป็นแผนภาพได้ดังต่อไปนี้

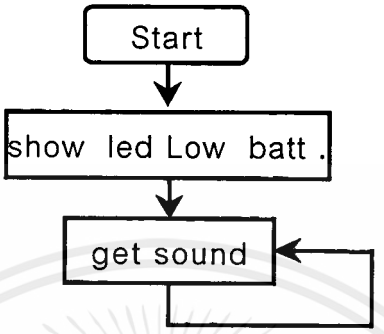


รูปที่ 5.14 การออกแบบโปรแกรมส่วนรับข้อมูลจากเครื่องอ่านรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.3 ส่วนที่ตรวจสอบแบตเตอรี่แรงดันต่ำ เขียนโปรแกรมแบบอินเตอร์รัพต์

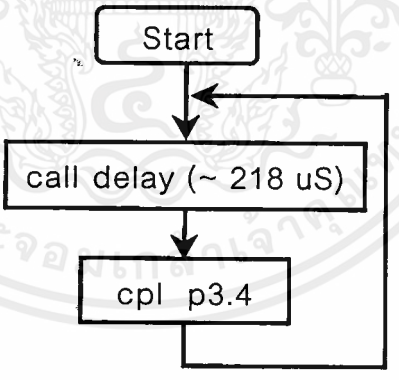
ภาคนี้อาศัยการอินเทอร์รัพต์ทางฮาร์ดแวร์ คือเมื่อมีสัญญาณแย้งมากระตุ้นไมโครโปรเซสเซอร์ก็จะไปทำงาน โดยหยุดการทำงานทั้งหมด แสดงไฟแบตหมด และส่งเสียงร้องตลอดเวลาการทำงานดังแผนภาพต่อไปนี้



รูปที่ 5.15 แผนภาพการตรวจสอบแบตเตอรี่หมด

5.2.4 ส่วนที่สร้างเสียง

อาศัยหลักการทำสวิทช์ของขาไมโครโปรเซสเซอร์ และนำสัญญาณที่ได้ไปขยายผ่านออปแอมป์เพื่อให้เสียงดังขึ้น ดังแสดงแผนภาพต่อไปนี้

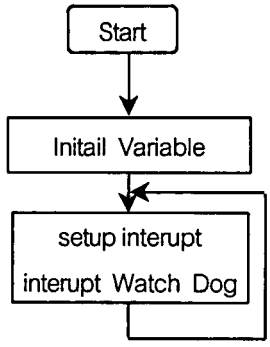


รูปที่ 5.16 การออกแบบโปรแกรมส่วนสร้างเสียง

5.2.5 โปรแกรมหลัก

โปรแกรมหลักมีหน้าที่ตั้งค่าตัวแปรขณะที่ไมโครโปรเซสเซอร์เริ่มทำงานและ ส่งสัญญาณไปกระตุ้นวงจรรีเซต เพื่อป้องกันการรีเซตไมโครโปรเซสเซอร์ เนื่องจากภายใน 1.2 วินาทีถ้าไม่มีการกระตุ้นวงจรรีเซต วงจรนี้ก็จะทำงาน แผนภาพแสดงการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



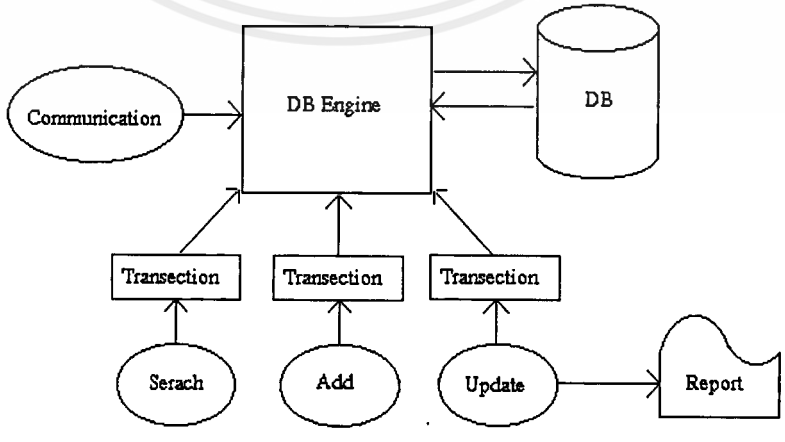
รูปที่ 5.17 แผนภาพแสดงการออกแบบโปรแกรมหลัก

ข้อมูลจำเพาะของเครื่องเก็บรหัส

- เมื่อไม่ต่อกับเครื่องอ่านรหัสจะกินกระแสเพียง 50 mA เท่านั้น
- มีวงจรรีเซตกรณีที่ CPU ทำงานผิดพลาด
- มีวงจรประจุแบตเตอรี่ภายใน
- ใช้ไฟ AC 12 V หรือไฟตรง 17 V
- มีพอร์ตเชื่อมต่อแบบ RS-232-C (ติดต่อกับคอมพิวเตอร์) ความเร็ว 9600 บิตต่อวินาที
- มีพอร์ตเชื่อมต่อแบบ Keyboard (ต่อเข้ากับเครื่องอ่านแถบรหัส)
- ใช้หน่วยความจำแบบนอนโวลาไทล์ขนาด 32 K
- สามารถเก็บรหัสได้ 1,280 รายการ
- ใช้แบตเตอรี่แบบนิกเกิล-เมตัลไฮไดรด์ไม่เป็นอันตรายกับสิ่งแวดล้อม

5.3 การออกแบบโปรแกรม

ในการโปรแกรมนี้จะถูกแบ่งเป็นหลายส่วน ตามลักษณะของหน้าที่ในแต่ละฟอร์มดังรูปที่ 5.18 เป็นรูปที่แสดงการเชื่อมต่อกันของโปรกแรมและจะได้ทำการกล่าวหน้าที่ของแต่ละฟอร์มไว้ดังนี้

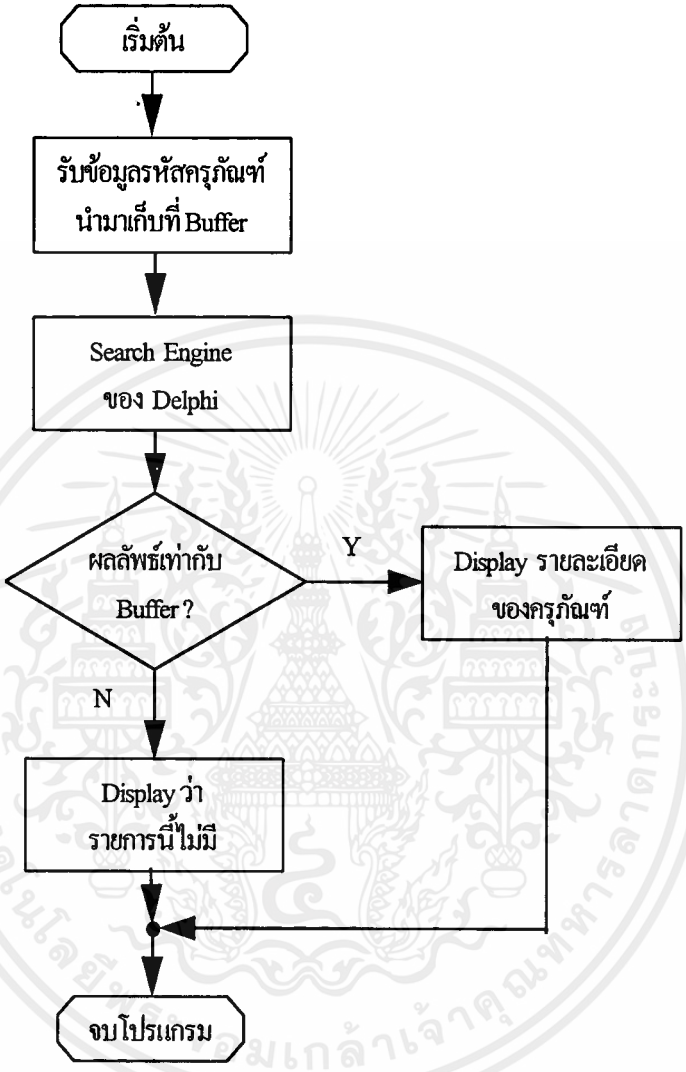


รูปที่ 5.18 การเชื่อมต่อของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3.1 การโปรแกรมในฟอร์มค้นหารายการ

ในฟอร์มนี้ มีหน้าที่ค้นหารายการครุภัณฑ์ จากข้อมูลรหัสครุภัณฑ์ที่ผู้ใช้โปรแกรมป้อนรหัสครุภัณฑ์เข้ามา จะมีขั้นตอนการทำงานตามผังล่างดังต่อไปนี้

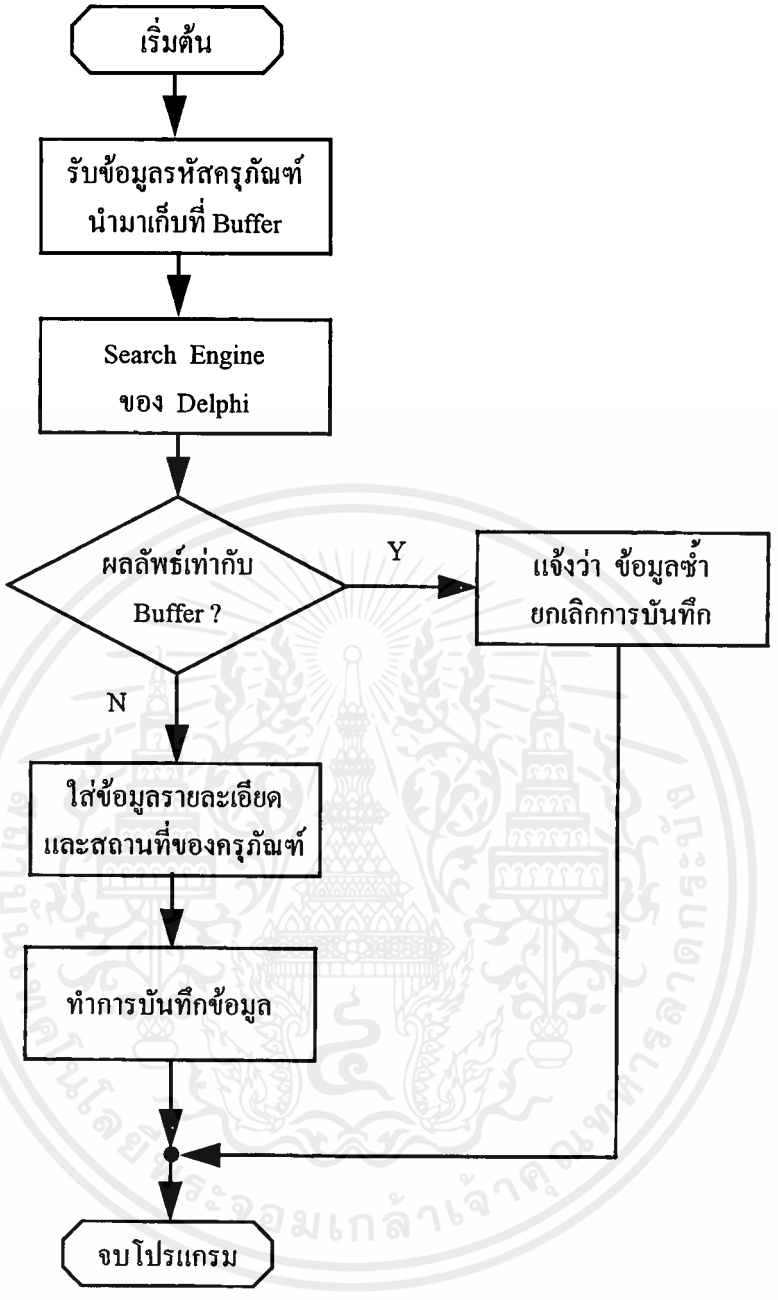


รูปที่ 5.19 ขั้นตอนของการค้นหา

5.3.2 การโปรแกรมในฟอร์มเพิ่มรายการครุภัณฑ์

การโปรแกรมในฟอร์มนี้มีหน้าที่เพิ่มรายการครุภัณฑ์ โดยการป้อนรหัสครุภัณฑ์ 16 หลัก และทำการเลือกว่ามีรายละเอียดเป็นแบบใด ตั้งอยู่สถานที่ใด หลังจากนั้นจะทำการบันทึกจะมีขั้นตอนตามผังดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

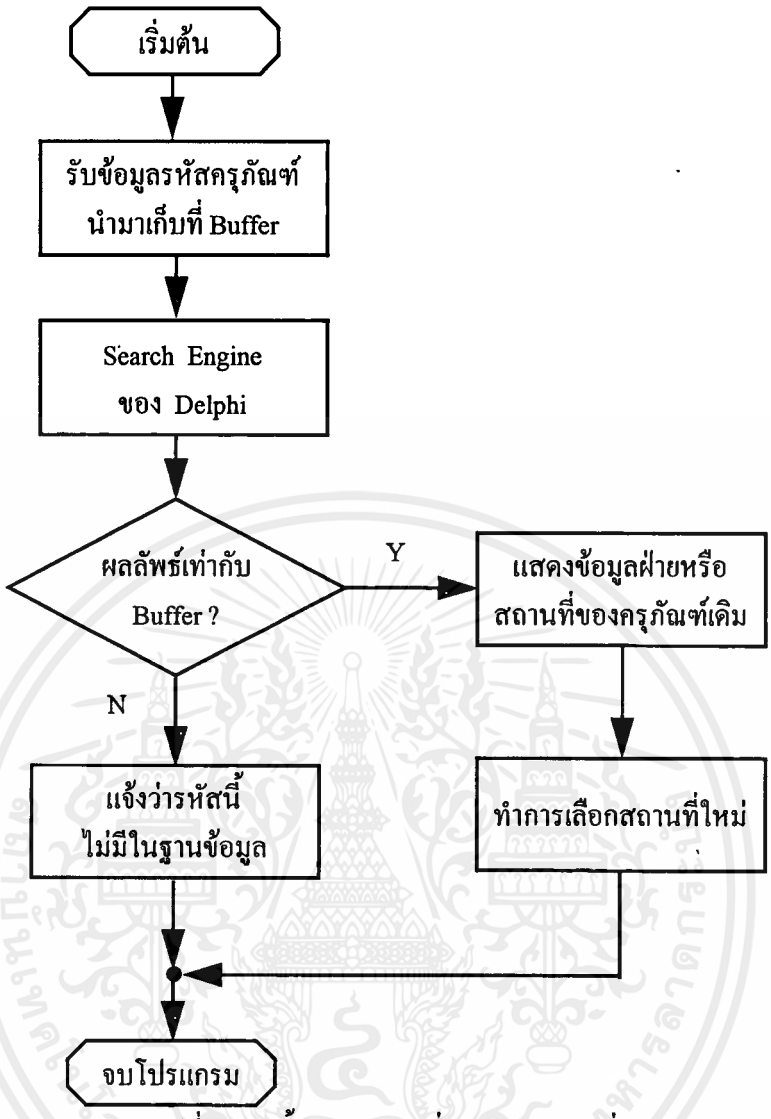


รูปที่ 5.20 ขั้นตอนของการเพิ่มข้อมูล

5.3.3 การโปรแกรมในฟอร์มเปลี่ยนแปลงสถานที่

ส่วนโปรแกรมในฟอร์มนี้จะมีหน้าที่เปลี่ยนแปลงสถานที่ของครุภัณฑ์ โดยทำการป้อนรหัสของครุภัณฑ์ โปรแกรมนี้ก็จะทำการค้นหารายการและแสดงถึงสถานที่ตั้งของครุภัณฑ์ ถ้าไม่มีรายการอยู่ในฐานข้อมูล โปรแกรมจะแสดงว่ารายการนี้ไม่มีอยู่ในฐานข้อมูล หลังจากนั้นให้ทำการเลือกสถานที่ใหม่แล้วทำการบันทึกข้อมูลที่แก้ไขแล้ว ดังจะแสดงตามผังดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.21 ขั้นตอนการเปลี่ยนแปลงสถานที่

5.3.4 การโปรแกรมในฟอร์มการสำรวจรายการครุภัณฑ์

การโปรแกรมในฟอร์มนี้จะทำการสำรวจรายการครุภัณฑ์ใน 1 รอบกำหนด จะมีขั้นตอนการทำงานคือจะทำการแก้ไขรายการตรวจสอบครุภัณฑ์ในตารางหลัก เมื่อเลือกที่จะทำการอ่านข้อมูลจากพอร์ตอนุกรม ก็จะมีการตรวจสอบพอร์ตอนุกรม ถ้ามีการเชื่อมต่อกันเป็นปกติ เครื่องเก็บข้อมูลรหัสแถบจะทำการส่งรายการครุภัณฑ์มาครั้งละ 1 รายการ เพื่อทำการปรับปรุงครุภัณฑ์นั้น ๆ รายการที่ได้รับมานั้นจะแสดงผลที่หน้าจอ

5.3.5 การโปรแกรมในฟอร์มเพิ่มเติมและแก้ไขรายละเอียดครุภัณฑ์

การโปรแกรมในฟอร์มนี้จะมีให้เลือกทำรายการใหญ่ ๆ อยู่ 3 รายการคือ เพิ่มเติมรายละเอียดครุภัณฑ์ แก้ไขรายละเอียดครุภัณฑ์ และเพิ่มเติมและแก้ไขฝ่ายหรือสถานที่ ซึ่งในรายการที่กล่าวมานั้นก็จะมีให้เลือกทำรายการย่อย ๆ อีก เช่น ในรายการเพิ่มเติมรายละเอียดครุภัณฑ์ก็จะมีให้เลือกว่าจะทำรายการเพิ่มประเภทครุภัณฑ์ เพิ่มชนิดครุภัณฑ์ และเพิ่มแบบครุภัณฑ์

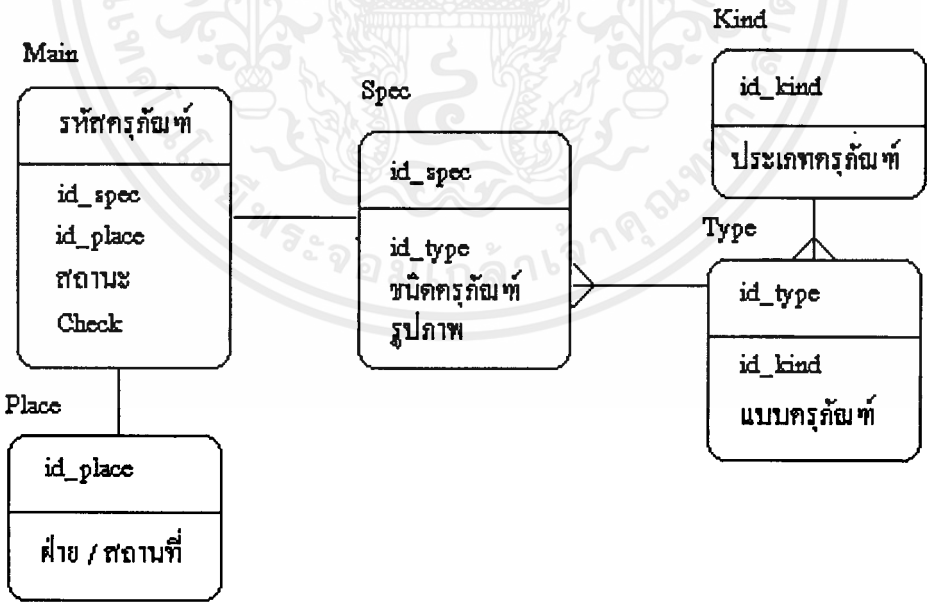
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 การออกแบบฐานข้อมูล

เมื่อได้ศึกษาการจัดเก็บฐานข้อมูลและการตรวจของครุภัณฑ์ของสำนักหอสมุดกลางแล้ว ทางคณะผู้จัดทำโครงงานจึงได้ปรึกษาเพื่อกำหนดหัวข้อ (Attribute) ต่างๆ ที่จะใช้ในการทำฐานข้อมูล ในขั้นต้นดังนี้

รหัสครุภัณฑ์	ชื่อครุภัณฑ์	ข้อมูลของครุภัณฑ์	ภาพของครุภัณฑ์	อาคาร	ห้อง	สถานะเดิม	สถานะปัจจุบัน

จากตาราง ข้างต้นเมื่อได้พิจารณาดูแล้วว่า จะทำให้เกิดการซ้ำซ้อนของการจัดเก็บฐานข้อมูล ในตารางทำให้เปลืองเนื้อที่โดยเปล่าประโยชน์ จากข้อมูลจริง ๆ จะเห็นว่า ครุภัณฑ์ชนิดเดียวกันจะมีจำนวนในการใช้งานมาก เช่น โต๊ะ เก้าอี้ เป็นต้น จะต่างกันก็เพียงรหัสของครุภัณฑ์ ดังนั้น ถ้าเป็นเช่นนี้ ก็จะส่วนที่ซ้ำกันในหัวข้อ (Attribute) ชื่อครุภัณฑ์ ข้อมูลของครุภัณฑ์ ภาพของครุภัณฑ์ และยังมีข้อมูลในหัวข้อ (Attribute) อาคาร และห้องที่ครุภัณฑ์ตั้งอยู่ เมื่อเราพิจารณาจะเห็นได้ว่า ในอาคาร 1 อาคาร หรือ ห้อง 1 ห้อง ก็จะมีครุภัณฑ์อยู่หลายชิ้นเช่นเดียวกัน ก็จะทำให้เกิดความซ้ำซ้อนของข้อมูลอีก ดังนั้น จึงได้ใช้วิธีการออกแบบฐานข้อมูลโดยใช้วิธีของ ER จะได้ผลดังรูปด้านล่าง เมื่อได้ศึกษาและได้ออกแบบฐานข้อมูลอีกครั้ง จึงออกแบบได้เป็น 5 ตารางดังนี้



รูปที่ 5.22 ผลลัพธ์การออกแบบฐานข้อมูลโดยวิธี ER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAIN

รหัสครุภัณฑ์	รหัสข้อมูลครุภัณฑ์	รหัสสถานที่ตั้ง	สถานะเดิม	สถานะปัจจุบัน

SPEC

รหัสข้อมูลครุภัณฑ์	รหัสแบบ	ชนิดของครุภัณฑ์	ภาพของครุภัณฑ์

PLACE

รหัสสถานที่ตั้ง	ฝ่าย / ห้อง

TYPE

รหัสแบบ	รหัสประเภท	ข้อมูลแบบ

KIND

รหัสประเภท	ข้อมูลประเภท

รูปที่ 5.23 หัวข้อของตารางทั้ง 5 ตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การโปรแกรม การทดสอบโปรแกรมและการติดตั้ง

6.1 การโปรแกรม

การโปรแกรมแบ่งออกเป็นหลายส่วนต่าง ๆ กัน ดังนี้

6.1.1 การโปรแกรมในฟอร์มค้นหารายการ

การโปรแกรมในส่วนนี้เป็นการโปรแกรมของการค้นหารายละเอียดของครุภัณฑ์ การโปรแกรมในส่วนนี้จะมีการใช้คำสั่งในการค้นหาของภาษา Delphi เป็นคำสั่งในการค้นหาข้อมูลในฐานข้อมูลดังจะเป็นไปตามขั้นตอนของผังโปรแกรมในบทที่ 5 รูปที่ 5.19

6.1.2 การโปรแกรมในฟอร์มเพิ่มรายการครุภัณฑ์

โปรแกรมส่วนนี้เป็นการทำการเพิ่มข้อมูลของครุภัณฑ์ลงในฐานข้อมูลของโปรแกรม ทั้งนี้ก็จะใช้คำสั่งในการเพิ่มข้อมูลไปครั้งละ 1 เรคคอร์ด โดยมีการเชื่อมโยงระหว่างฐานข้อมูลหลายตาราง เช่น ตารางของรายละเอียดครุภัณฑ์ ตารางฝ่ายหรือสถานที่ครุภัณฑ์ โดยการใช้รหัสในการเชื่อมโยง ดังจะแสดงตามขั้นตอนในผังโปรแกรมในบทที่ 5 รูปที่ 5.20

6.1.3 การโปรแกรมในฟอร์มเปลี่ยนแปลงสถานที่

ในฟอร์มนี้จะใช้การโปรแกรมแบบแก้ไขข้อมูล โดยจะมีวิธีการแก้ไขข้อมูลโดยใช้รหัสครุภัณฑ์เป็นข้อมูลที่เรียกข้อมูลในหัวข้อฝ่ายหรือสถานที่ของครุภัณฑ์นั้นขึ้นมาแก้ไขหรือเปลี่ยนแปลงสถานที่นั่นเอง ทั้งนี้จะไม่ได้แก้ที่ตารางฝ่ายหรือสถานที่ของครุภัณฑ์ แต่เป็นการเปลี่ยนรหัสของสถานที่ในตารางหลัก

6.1.4 การโปรแกรมในฟอร์มการสำรวจรายการครุภัณฑ์

ส่วนการโปรแกรมในฟอร์มนี้จะมีการติดต่อสื่อสารกับพอร์ตอนุกรมเป็นการเรียกข้อมูลจากเครื่องเก็บข้อมูลรหัสแถบ รวมถึงมีการแก้ไขสภาวะการตรวจสอบในตารางหลัก ยังมีการค้นหาครุภัณฑ์จากข้อมูลที่ได้จากเครื่องเก็บรหัสแถบ โปรแกรมส่วนนี้จะทำหน้าที่รับข้อมูลจากเครื่องเก็บข้อมูลรหัสแถบนำไปค้นหาครุภัณฑ์ชิ้นนั้น ๆ และทำการแก้ไขสภาวะการตรวจสอบในตารางหลัก เพื่อดูว่าครุภัณฑ์ชิ้นใดอยู่หรือครุภัณฑ์ชิ้นใดสูญหาย

6.1.5 การโปรแกรมในฟอร์มเพิ่มเติมและแก้ไขรายละเอียดครุภัณฑ์

การโปรแกรมในฟอร์มนี้ จะทำหน้าที่เพิ่มเติมและแก้ไขรายละเอียดครุภัณฑ์ รวมถึงเพิ่มเติมและแก้ไขฝ่ายหรือสถานที่ มีรายละเอียดประเภท ชนิด แบบ ข้อมูลภาพของครุภัณฑ์ จะมีขั้นตอนของโปรแกรม โดยมีการให้เลือกว่าจะทำรายการเพิ่มเติมหรือแก้ไข ถ้าเป็นรายการเพิ่มเติมในรายการต่าง ๆ จะมีการทำรหัสประเภท รหัสชนิด รหัสแบบและรหัสสถานที่ เมื่อทำการเพิ่มเติมหรือแก้ไขเสร็จจึงทำการบันทึกข้อมูลนั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.1.6 การโปรแกรมในฟอร์มทดสอบการสื่อสาร

ฟอร์มทดสอบการสื่อสาร ใช้โปรแกรมในส่วนของการสื่อสาร จะมรายละเอียดการติดตั้งพอร์ตอนุกรม ทดสอบสถานะการณ์เชื่อมต่อ ทดสอบหน่วยความจำของเครื่องเก็บรหัสแถบและทดสอบสถานะของเครื่องเก็บข้อมูลรหัสแถบ

6.2 การทดสอบโปรแกรม

การทดสอบโปรแกรมแบ่งออกเป็นหลายส่วนต่าง ๆ ดังนี้

6.2.1 โปรแกรมที่เครื่องเก็บข้อมูลรหัสแถบ

ทดสอบโดยการนำเครื่องอ่านรหัสแถบ (Laser Gun) ต่อเข้ากับเครื่องเก็บข้อมูลรหัสแถบแล้วทำการทดสอบอ่านข้อมูลรหัสแถบที่เตรียมไว้ หลังจากนั้นจึงทำการทดสอบว่าข้อมูลที่เครื่องเก็บข้อมูลรหัสแถบเป็นข้อมูลเดียวกัน กับข้อมูลที่รหัสแถบหรือไม่ โดยนำเครื่องเก็บข้อมูลรหัสแถบมาต่อเข้ากับเครื่องคอมพิวเตอร์ที่เตรียมไว้ ทดลองอ่านข้อมูลกลับมาจากเครื่องเก็บข้อมูลรหัสแถบ

6.2.2 โปรแกรมการเพิ่มครุภัณฑ์

โปรแกรมนี้เป็นส่วนที่ใช้ในการเพิ่มข้อมูลต่างๆ ของครุภัณฑ์ เช่น รหัสครุภัณฑ์ รายละเอียดของครุภัณฑ์ ฝ่ายหรือสถานที่ที่ครุภัณฑ์ตั้งอยู่ ฉนั้นจึงมีการทดสอบโดยดารใส่ข้อมูลต่างๆ ไปยังฐานข้อมูลของโปรแกรม หลังจากนั้นจึงไปขอดูข้อมูลจาก Database Desktop เป็นโปรแกรมตัวหนึ่งที่จัดการกับฐานข้อมูล ต่อมาให้ออกจากโปรแกรมแล้วทำการปิดเครื่องคอมพิวเตอร์ไว้สักระยะ หลังจากนั้นให้ทำการขอดูข้อมูลจาก Database Desktop อีกครั้งหนึ่ง

6.2.3 โปรแกรมการค้นหารายการ

ในส่วนนี้เป็นส่วนที่ใช้ในการค้นหารายการครุภัณฑ์ โดยดารป้อนรหัสครุภัณฑ์และนำข้อมูลส่วนนั้นไปทำการค้นหา มีวิธีการทดสอบคือ ทดลองใส่รหัสครุภัณฑ์ที่มีอยู่ในฐานข้อมูลแล้วทำการค้นหารายการ สิ่งที่จะแสดงให้เราดูจะมีรายละเอียดต่างๆ ของครุภัณฑ์ และทดลองใส่รหัสครุภัณฑ์ที่ไม่มีอยู่ในฐานข้อมูลสิ่งที่จะแสดงให้เราดู คือ รายการนี้ไม่ในฐานข้อมูล

6.2.4 โปรแกรมการแก้ไขรายละเอียด

การแก้ไขรายละเอียดของครุภัณฑ์ ทดสอบโดยใช้วิธีการเลือกนำข้อมูลของครุภัณฑ์ ขึ้นมาแสดงหนึ่งชุดหลังจากนั้นจึงไปขอดูข้อมูลจาก Database Desktop เป็นโปรแกรมตัวหนึ่งที่จัดการกับฐานข้อมูล ต่อมาให้ออกจากโปรแกรมแล้วทำการปิดเครื่องคอมพิวเตอร์ไว้สักระยะ หลังจากนั้นให้ทำการขอดูข้อมูลจาก Database Desktop อีกครั้งหนึ่ง

6.2.5 โปรแกรมการติดต่อสื่อสาร

ในส่วนนี้ เป็นโปรแกรมที่ใช้ในการติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์กับเครื่องเก็บข้อมูลรหัสแถบ ทดสอบโดยดารนำเครื่องเก็บข้อมูลรหัสแถบมาต่อกับเครื่องคอมพิวเตอร์และทดลองอ่านข้อมูลจากเครื่องเก็บข้อมูลรหัสแถบ นำข้อมูลนั้นมาแสดงที่โปรแกรม

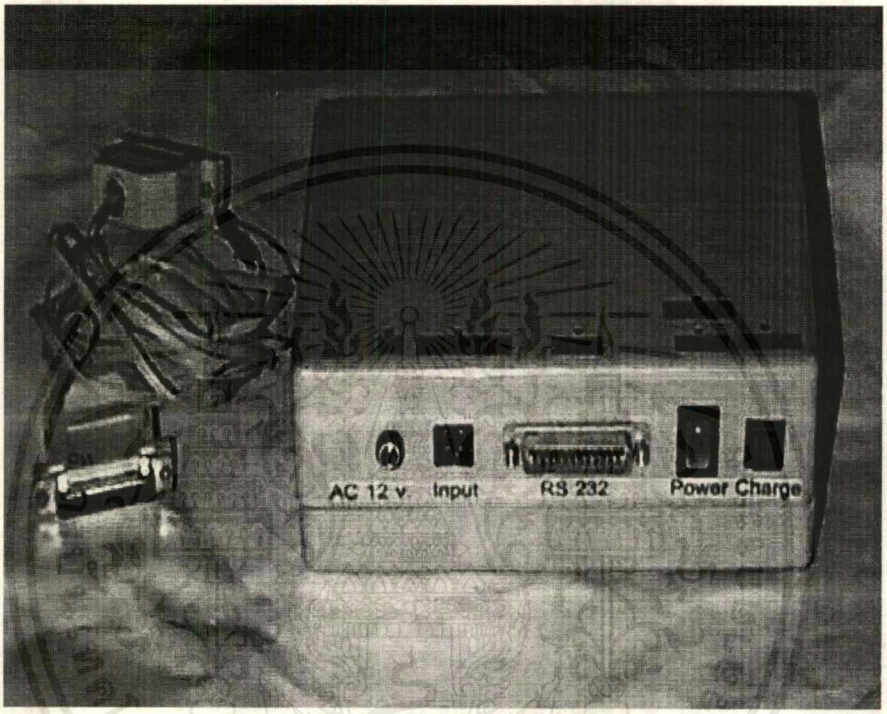
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.3 การติดตั้งและการใช้งานเครื่องเก็บรหัสแถบ

เครื่องเก็บรหัสแถบประกอบด้วยส่วนต่าง ๆ ส่วนดังนี้

1. แหล่งจ่ายไฟกระแสสลับ
2. สายสื่อสาร
3. เครื่องเก็บรหัสแถบ
4. อุปกรณ์ต่อร่วมอื่น ๆ คือ เครื่องอ่านรหัสแถบ และ คอมพิวเตอร์

ดังรูปต่อไปนี้



รูปที่ 6.1 เครื่องเก็บข้อมูลรหัสแถบ

การติดตั้งสามารถติดตั้งได้ 3 แบบ ตามการใช้งานดังนี้

6.3.1 การติดตั้งเพื่อประจุแบตเตอรี่

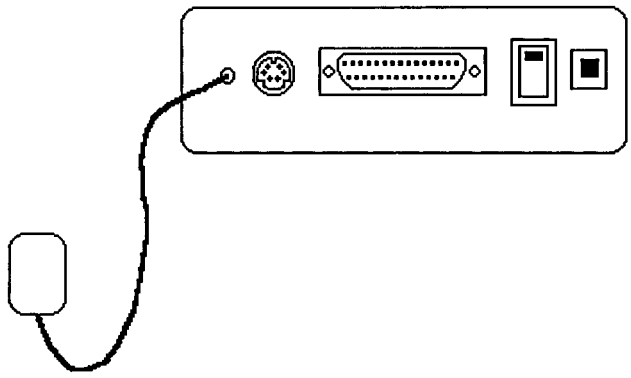
กระทำได้โดยการนำเอาแหล่งจ่ยไฟ AC 12 V เข้ามาเชื่อมต่อช่อง AC 12 V การประจุจะประจุได้ 2 ลักษณะคือ

1) การประจุแบบช้า การประจุแบบนี้จะกระทำก็ต่อเมื่อต่อแหล่งจ่ายไฟช่อง AC 12 V ทันที การประจุ แบบนี้สามาร๔ประจุได้นาน ๆ นับ 10-30 ชั่วโมง โดยแบตเตอรี่จะไม่เสียหาย

2) การประจุแบบเร็ว การประจุแบบนี้จะต้องกดสวิทช์ Charge 1 ครั้ง เครื่องก็จะทำการประจุทันที ใช้เวลาประมาณ 3.5 ชั่วโมง และเมื่อแบตเต็มจะมีไฟ LED สว่าง เป็นการบอกว่าแบตเต็ม การประจุแบบนี้เมื่อแบตเต็ม สามารถทิ้งไว้ได้อีกนานประมาณ 10 ชั่วโมง โดยไม่เสียหาย

เมื่อประจุแบตเต็มก็จะพร้อมใช้งานได้ ทันที ดังรูปการติดตั้งต่อไปนี้

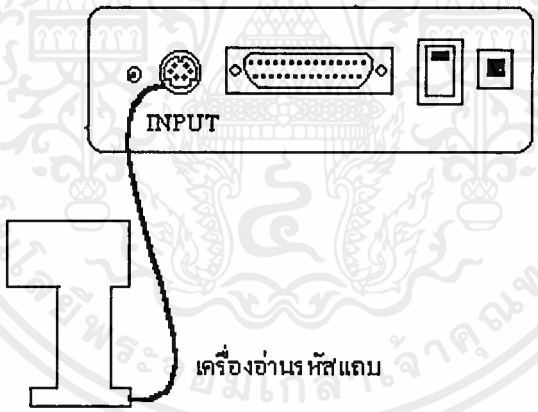
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.2 การต่อแหล่งจ่ายไฟกับเครื่องเก็บข้อมูลรหัสแถบ

6.3.2 การติดตั้งกับเครื่องอ่านรหัสแถบ

การติดตั้งแบบนี้ เป็นการนำไปใช้งานเพื่อเก็บรหัสแถบจากเครื่องเครื่องอ่านรหัสโดยการเสียบปลายสายของเครื่องอ่านรหัสเข้ากับช่อง INPUT ของเครื่องเก็บรหัส ดังรูป



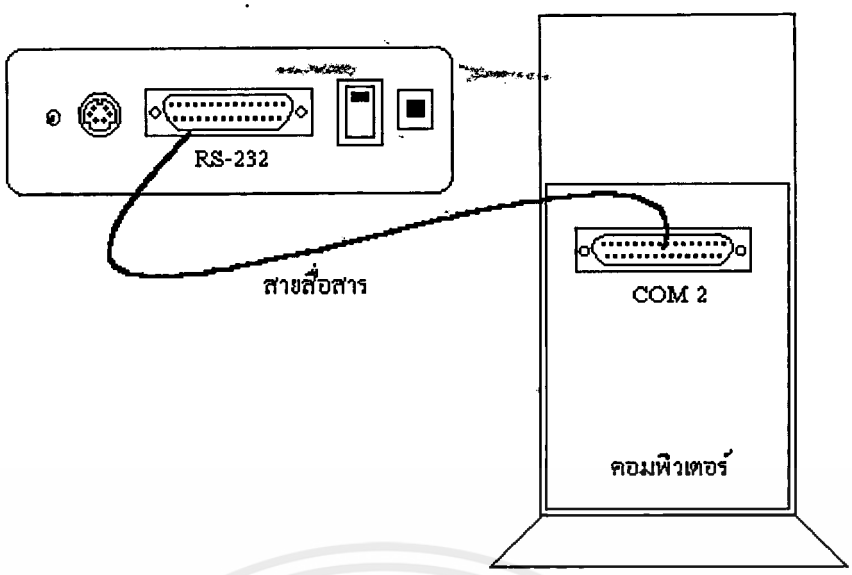
รูปที่ 6.3 เครื่องเก็บข้อมูลรหัสแถบพร้อมใช้งาน

การติดตั้งเรียบร้อยแล้วสามารถนำไปใช้งานได้ทันที ขณะใช้งานเมื่อแบตอ่อนจะมีเสียงเตือนและไฟ LOW BATT สว่าง ให้ปิดเครื่องทันทีและนำไปประจุ แบตเพื่อใช้งานต่อ เมื่อหน่วยความจำเต็มจะมีเสียงเตือนและไฟทุกดวงจะสว่างค้างให้ปิดเครื่องและนำไปถ่ายข้อมูลแก่ คอมพิวเตอร์ทันที

6.3.3 การเชื่อมต่อกับคอมพิวเตอร์

เป็นการใช้งานทางด้านการตรวจสอบ เครื่องเก็บรหัส และการถ่ายข้อมูล การเชื่อมต่อให้นำสายสื่อสารต่อเข้าที่ ช่อง RS-232 อีกด้านต่อเข้ากับ คอมพิวเตอร์ ช่อง COM 2 จากนั้นก็สามารถที่จะใช้งานกับคอมพิวเตอร์ได้ทันที ดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.4 การต่อเครื่องเก็บข้อมูลรหัสแถบเข้ากับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

สรุปผลการพัฒนาและข้อเสนอแนะ

7.1 สรุปผลการพัฒนา

การพัฒนาโปรแกรมโดยใช้วิธีการโปรแกรมเชิงวัตถุ (Object - Oriented Programming) เป็นการพัฒนาโปรแกรมแบบใหม่ ซึ่งถ้าเป็นในอดีตจะเป็นการพัฒนาโปรแกรมในลักษณะของโครงสร้าง (Structure Programming) เป็นการพัฒนาที่ผู้พัฒนานั้น จะต้องกระทำการพัฒนาด้วยตนเองตั้งแต่ต้น ในการพัฒนาโปรแกรมในลักษณะนี้ จะเห็นได้ว่าเป็นการพัฒนาโปรแกรมที่ทำให้การพัฒนาเป็นไปได้ช้า การพัฒนาในลักษณะโครงสร้างก็จะมีอยู่หลายภาษาที่เรายังรู้จักกัน เช่น ภาษา C , ภาษา Pascal , ภาษา Basic เป็นต้น หลังจากที่ได้มีการพัฒนาโปรแกรมเชิงวัตถุทำให้การพัฒนาโปรแกรมเป็นไปได้เร็วยิ่งขึ้น เนื่องมาจากเหตุผลที่ว่า เป็นการนำสิ่งที่เคยมีผู้อื่นพัฒนาไว้ในแต่ละด้านนำมารวมกัน จนกระทั่งเป็นโปรแกรมที่เราต้องการ การพัฒนาโปรแกรมนี้ก็จะมีอยู่หลายภาษาที่เราพอทราบกันแล้ว เช่น ภาษา Delphi , ภาษา Visual Basic , ภาษา visual C++ เป็นต้น

ในโครงงานนี้ เป็นโครงงานที่นำการพัฒนาโปรแกรมเชิงวัตถุ มาประยุกต์เข้ากับงานฐานข้อมูลของสำนักหอสมุดกลาง เพื่อทำให้เกิดความสะดวกแก่เจ้าหน้าที่ของสำนักหอสมุดกลาง และลดความผิดพลาดในการทำงานของเจ้าหน้าที่ เป็นโปรแกรมที่ใช้งานเกี่ยวกับการตรวจสอบครุภัณฑ์ ฉนั้น ในโปรแกรมนี้จะมีการกระทำกับฐานข้อมูล เช่น การเพิ่มข้อมูลครุภัณฑ์ การแก้ไขรายละเอียดต่างๆ ของครุภัณฑ์ และการค้นหาข้อมูลต่างๆ ของครุภัณฑ์แต่ละชิ้น ในฐานข้อมูลของครุภัณฑ์แต่ละชิ้นจะมีข้อมูล จำพวกรหัสของครุภัณฑ์ ชนิดของครุภัณฑ์ แบบของครุภัณฑ์ ข้อมูลภาพของครุภัณฑ์ สถานที่ที่ครุภัณฑ์แต่ละชิ้น และสถานะของครุภัณฑ์แต่ละชิ้น

เนื่องจากการพัฒนาโปรแกรมเชิงวัตถุ เป็นเรื่องที่ค่อนข้างใหม่ และภาษาที่ใช้ก็เป็นภาษาที่แตกต่างจากภาษาที่เคยพบมา จึงทำให้ต้องมีการศึกษาการโปรแกรมเชิงวัตถุจากที่ต่างๆ รวมถึงภาษาที่จะใช้ในการพัฒนาโปรแกรมด้วย

7.2 ข้อเสนอแนะ

เนื่องจากโปรแกรมเป็นโปรแกรมเฉพาะด้านควรจะปฏิบัติการตามคู่มือการใช้งานและเพื่อความปลอดภัยของข้อมูลในโปรแกรมนี้จะอนุญาตให้มีผู้ที่รับผิดชองเพียงผู้เดียวฉะนั้นไม่ควรให้ผู้ใดทราบรหัสผ่าน (Password)ได้ การใช้งานเครื่องเก็บข้อมูลรหัสแถบควรจะทำเมื่อไม่ได้ใช้งานไม่ควรเปิดเครื่องทิ้งไว้เพราะจะทำให้แบตเตอรี่หมดเร็วยิ่งขึ้น ไม่ควรทำการชาร์จแบตเตอรี่เมื่อแบตเตอรี่ยังไม่หมดหรือไฟที่กล่องเตือนว่า แบตเตอรี่อ่อน (Low Batt.) เพราะจะทำให้อายุการใช้งานของแบตเตอรี่สั้นลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ไทยโปรดักชั่นนัมเบอริ่งแอสโซเอชั่น . คู่มือการใช้บาร์โค้ด , กรุงเทพมหานคร , 2536

สมยศ ภูงามแปลง , เซมิคอนดักเตอร์อิเล็คทรอนิกส์ ฉบับที่ 93 เรื่อง ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว MCS 51 , สำนักพิมพ์ ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2532 , หน้า 264 – 269

เศกสิทธิ์ คำชมภู , เซมิคอนดักเตอร์อิเล็คทรอนิกส์ ฉบับที่ 148 เรื่อง โมดูลชาร์จแบตเตอรี่นิกเกิล – เมตัลไฮไดรด์ , สำนักพิมพ์ ซีเอ็ดยูเคชั่น จำกัด (มหาชน) , 2538 , หน้า 30 – 35 .

Philips Semiconductors , Application Note And Development Tools For 80C51 Philips Microcontrollers 1995 , 2538 ,หน้า 146 – 163 .

จิรศักดิ์ เหลืองอุไร , คัมภีร์การใช้งาน การสื่อสารอนุกรม PC , สำนักพิมพ์ ซีเอ็ดยูเคชั่น จำกัด (มหาชน) .

Mohammad Ali Mazidi and Janice Gillispie Mazidi , The 80X86 IBM PC & Compatible Computers , Published by Prentice – Hall ,Inc , หน้า 244 – 256 .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้