สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งานเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ดในระบบรักษาความปลอดภัย และอำนวยความสะดวก

Fingerprint and Smart cards application in Security and Facilitate system

โดย

นาย ชาญวิทย์ กิตติมากุล

นาย ฐานันดร อินทฤทธิ์

นาย วิรุณ มณีโชติ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

เลขทะเบียน 61936

วัน,เดือน,ปี 25 ม.ค. 2549

ล.....

ส.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ดในระบบรักษาความปลอดภัย และอำนวยความสะดวก

Fingerprint and Smart cards application in Security and Facilitate system

โดย

นาย ชาญวิทย์ กิตติมากุล รหัส 45015190

นาย ฐานันดร อินทฤทธิ์ รหัส 45015191

นาย วิรุณ มณีโชติ รหัส 45015209

อาจารย์ที่ปรึกษา รศ. สุชาติ คุณทวีเทพ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ดในระบบรักษาความปลอดภัยและอำนวยความสะดวก

Fingerprint and Smart cards application in Security and Facilitate system

ผู้จัดทำ

นาย ชาญวิทย์ กิตติมากุล รหัส 45015190

นาย ฐานันดร อินทฤทธิ์ รหัส 45015191

นาย วิรุณ มณีโชติ รหัส 45015207

โครงงานนี้ได้รับการตรวจสอบพร้อมที่ทำการสอบได้

อาจารย์ที่ปรึกษา

(รศ. สุชาติ คุณาวีเทพ)

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ดในระบบรักษาความปลอดภัยและอำนวยความสะดวก

นาย ชาญวิทย์ กิตติมากุล รหัส 45015190
นาย ฐานันดร อินทฤทธิ์ รหัส 45015191
นาย วิรุณ มณีโชติ รหัส 450151209
รศ.สุชาติ คูณทวีเทพ อาจารย์ที่ปรึกษา
ภาคการศึกษาที่ 1ปีการศึกษา 2547

บทคัดย่อ

ในปัจจุบันนี้เทคโนโลยีทางอิเล็กทรอนิกส์ได้ก้าวไปไกลมาก ซึ่งระบบรักษาความปลอดภัยและอำนวยความสะดวกสามารถนำเอาเครื่องตรวจสอบลายนิ้วมือ ร่วมกับสมาร์ตการ์ดมาประยุกต์ใช้ได้ ในโครงงานนี้ได้ใช้เครื่องตรวจสอบลายนิ้วมือมาทำเป็นระบบรักษาความปลอดภัยร่วมกับสมาร์ตการ์ด โดยใช้ MCS-51 เป็นบอร์ดอินเทอร์เฟสคอยติดต่อกับบอร์ดควบคุม BKS-1700 ข้อมูลที่ได้จะนำมาวิเคราะห์ในฐานข้อมูลที่มีอยู่โดยใช้โปรแกรม Microsoft Visual Basic Version 6.0 Enterprise ในการออกแบบและสร้างโปรแกรมที่ทำการวิเคราะห์ฐานข้อมูลดังกล่าว รวมทั้งการแสดงรายละเอียดผลการวิเคราะห์ผ่านจอคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Fingerprint and Smart cards application in Security and Facilitate system

Mr.Charnwit Kittimakul 45015190
Mr.Thanundorn Intarit 45015191
Mr.Wirun Maneechot 45015209
Assoc.Prof.Suchart Khuntaweetep
Academic year 2004

Abstract

Electronics technologies have been made progresses every day now we can apply fingerprint and smart cards for security and facilitate system. This project utilizes disgarded fingerprint module BKS-1700 to make security and facilitate system card. Board interface was used micro controller MCS-51 to get fingerprint data size and control data transfer from fingerprint to computer by using RS-232 serial port standard .The data analysis is using Microsoft Visual Basic Version 6.0 Enterprise and show the analysis result detail through computer monitor .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี ก็ด้วยความเมตตาของ รศ .สุชาติ คูณทวีเทพ ที่ท่านได้ให้คำปรึกษาอย่างเป็นกันเอง เอื้ออำนวยเรื่องอุปกรณ์การทดลอง และห้องทำงานอันสะดวกสบาย

และโดยส่วนตัวของพวกกระผม ก็ขอขอบคุณครอบครัวของพวกผมที่ได้ให้ การสนับสนุนกำลังทรัพย์โดยตลอดมาในการทำโครงงานที่ไม่จำกัด อีกทั้งยังคอยเป็นกำลังใจตลอดเวลาในการทำงานจนปริญญานิพนธ์ ฉบับนี้ สำเร็จลุล่วงไปได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	V
สารบัญตาราง	VI
บทที่ 1 บทนำ	1
บทที่ 2 เครื่องตรวจสอบลายนิ้วมือ	3
2.1 ประวัติโดยย่อของเครื่องตรวจสอบลายนิ้วมือ	3
2.2 ลักษณะลายนิ้วมือ	3
2.3 ความรู้เบื้องต้นของลายนิ้วมือ	4
2.3.1 ตำหนิและลักษณะต่างๆ	4
2.3.2 ลักษณะพิเศษบางอย่าง	4
2.4 คำจำกัดความที่สำคัญบนลายนิ้วมือ	5
2.4.1 เส้นขอบ	5
2.4.2 สันดอน	6
2.4.3 จุดใจกลาง	6
2.4.4 บริเวณลายนิ้วมือที่อยู่ภายใน	6
2.5 แบบของลายนิ้วมือ แบ่งเป็น 4 กลุ่มดังนี้	6
2.5.1 กลุ่มที่ 1 เส้นโค้ง (Arch)	6
2.5.2 กลุ่มที่ 2 ลูปหรือมัดหวาย	7
2.5.3 กลุ่มที่ 3 ก้นหอย	8
2.5.4 กลุ่มที่ 4 ซับซ้อน	9
2.6 ชนิดของเครื่องตรวจสอบลายนิ้วมือ	10
2.6.1 เครื่องตรวจสอบลายนิ้วมือที่ใช้แสง	10
2.6.2 เครื่องตรวจสอบลายนิ้วมือที่ใช้ความจุไฟฟ้า	11
2.7 การวิเคราะห์	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ข้อดีและข้อเสียของเครื่องตรวจสอบลายนิ้วมือ 13
2.9 รูปแบบพื้นฐานที่ควรรู้เกี่ยวกับ Biometric 14
2.10 คุณภาพของลายนิ้วมือ 14
2.11 รายละเอียดเครื่องตรวจสอบลายนิ้วมือที่ใช้ในโครงงาน 17
บทที่ 3 สมาร์ตการ์ด 19
3.1 ความหมายของสมาร์ตการ์ด 19
3.2 ข้อดีของสมาร์ตการ์ด 19
3.3 ส่วนประกอบและโครงสร้างของสมาร์ตการ์ด 20
3.3.1 ตัวบัตรพลาสติก (Plastic Card) 20
3.3.2 หน้าสัมผัสและชิปสมาร์ตการ์ด (Smart Card Module) 21
3.4 องค์ประกอบในการใช้งานสมาร์ตการ์ด 22
3.4.1 ตัวบัตรและตัวชิป 22
3.4.2 สมาร์ตรีดเดอร์ 22
3.5 รูปแบบของสมาร์ตการ์ดที่นำมาใช้ในโครงงาน 23
3.3.1 บัตรโทรศัพท์สาธารณะ TOT CARD 23
3.5.2 การ์ดที่มีระบบป้องกันข้อมูล SLE4442 30
บทที่ 4 การสื่อสารแบบอนุกรม 35
4.1 การสื่อสารแบบอนุกรม 35
4.2 การสื่อสารข้อมูลแบบอะซิงโครนัส 35
4.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232 38
4.3.1 คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ 38
4.3.2 UART (Universal Asynchronous Receiver Transmitter) 41
4.3.3 ชนิดของ UART 42
4.4 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232 42
4.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ตRS-232 50
4.6 แอดเดรสของพอร์ตอนุกรม 51
บทที่ 5 ไมโครคอนโทรลเลอร์ MCS-51 53
5.1 ไมโครคอนโทรลเลอร์ MCS-51 53
5.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 โครงสร้างหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51 54
5.4 หน่วยความจำโปรแกรม 38
5.5 หน่วยความจำข้อมูล 55
5.6 รีจิสเตอร์ที่เกี่ยวกับไมโครคอนโทรลเลอร์MCS-51 56
5.6.1 รีจิสเตอร์หน้าที่พิเศษ 56
5.6.2 แอกคิวมูเลเตอร์ 56
5.7 โครงสร้างการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51 58
5.8 ไทเมอร์ /เคาน์เตอร์ของไมโครคอนโทรลเลอร์ MCS-51 59
5.9 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 60
บทที่ 6 การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ด้วยภาษาซี 61
6.1 โครงสร้างของภาษาซี 61
6.2 ตัวดำเนินการในภาษาซี 62
6.2.1 ตัวดำเนินการที่กระทำกับตัวถูกกระทำตัวเดียว 63
6.2.2 ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว 64
6.3 ประโยคควบคุมในภาษาซี 65
6.3.1 ประโยค if / else 65
6.3.2 ประโยค switch 66
6.4 การทำซ้ำ 66
6.4.1 ประโยค for 66
6.4.2 ประโยค while 66
6.4.3 ประโยค do – while 67
6.5 อาร์เรย์ พอยน์เตอร์ และสตรักเจอร์ 67
บทที่ 7 การออกแบบและหลักการทำงานของวงจร 74
7.1 องค์ประกอบของระบบโดยรวม 74
7.2 หลักการทำงานโครงงาน 76
บทที่ 8 การทดลองและผลการทดลอง 81
บทที่ 9 วิเคราะห์และสรุปผลการทดลอง 86
บรรณานุกรม 87
ภาคผนวก 88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่2.1 ลักษณะสำคัญต่างๆ 4
รูปที่2.2 ความแตกต่างสำคัญๆบนลายนิ้วมือ 5
รูปที่2.3 Pattern Area 6
รูปที่2.4 โค้งราบ 6
รูปที่2.5 โค้งกระโจม 7
รูปที่2.6 มัดหวายปัดขวา 7
รูปที่2.7 มัดหวายปัดซ้าย 7
รูปที่2.8 มัดหวายแฝด 8
รูปที่2.9 ก้นหอยธรรมดา 8
รูปที่2.10 ก้นหอยกระเป๋ากลางปัดขวา 8
รูปที่2.11 ก้นหอยกระเป๋ากลางปัดซ้าย 9
รูปที่2.12 ก้นหอยกระเป๋าข้างปัดขวา 9
รูปที่2.13 ก้นหอยกระเป๋าข้างปัดซ้าย 9
รูปที่2.14 กลุ่มซับซ้อน 10
รูปที่ 2.15 สภาพลายนิ้วมือที่แสดงการเปรียบเทียบคุณภาพของภาพที่ได้ 16
รูปที่2.16 ขบวนการทำงานของการใช้งานเครื่องตรวจสอบลายนิ้วมือ 16
รูปที่2.17 แผนภาพโครงร่างบอร์ด 18
รูปที่ 3.1 ส่วนประกอบของสมาร์ตการ์ด 20
รูปที่ 3.2 การแบ่งชนิดของบัตรตามรูปร่างที่นำไปใช้งาน และขนาด 21
รูปที่ 3.3 ส่วนประกอบของสมาร์ตการ์ดโมดูลในสายการผลิตสมาร์ตการ์ด 21
รูปที่ 3.4 ตัวอย่างสมาร์ตการ์ดโมดูล 22
รูปที่ 3.5 ตัวอย่างบัตรโทรศัพท์ TOT 23
รูปที่ 3.6 ตัวอย่างหน่วยความจำของสมาร์ตการ์ดชนิด Memory แบบ Token 24
รูปที่ 3.7 หน่วยความจำข้อมูลของสมาร์ตการ์ดชนิด Memory 25
รูปที่ 3.8 วิธีทดสอบการบิดงอสมาร์ตการ์ด 26
รูปที่ 3.9 ตำแหน่งหน้าสัมผัสของชิปสมาร์ตการ์ด 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 หน้าที่การทำงานของแต่ละหน้าสัมผัส 27
รูปที่ 3.11 สมาร์ตชิปในบัตร TOT 27
รูปที่ 3.12 ไดอะแกรมแสดงส่วนการทำงานภายในสมาร์ตการ์ดตระกูล SLE4436 28
รูปที่ 3.13 แผนผังทางเวลาของสัญญาณที่เกี่ยวข้องสำหรับการอ่านข้อมูลจากบัตร SLE4436 30
รูปที่ 4.1 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม 35
รูปที่ 4.2 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 36
รูปที่ 4.3 คอนเนคเตอร์อนุกรม 9ขาหรือแบบ DB-9 39
รูปที่ 4.4 คอนเนคเตอร์อนุกรม 25 ขาหรือแบบ DB-25 39
รูปที่ 4.5 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ 41
รูปที่ 4.6ไดอะแกรมการทำงานภายในของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ 43
รูปที่ 4.7 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม 51
รูปที่ 5.1 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51 54
รูปที่ 5.2 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51 55
รูปที่ 5.3 แสดงการจัดหน่วยความจำข้อมูล 56
รูปที่ 7.1 โครงสร้างของระบบทั้งหมด 74
รูปที่ 7.2 ลักษณะการต่อทางฮาร์ดแวร์ 75
รูปที่ 7.3 โฟลชาร์ทการทำงานของเครื่องโหมดเริ่มต้น 77
รูปที่ 7.4 โฟลชาร์ทการทำงานของเครื่องโหมดที่ 1 78
รูปที่ 7.5 โฟลชาร์ทการทำงานของเครื่องโหมดที่ 2 79
รูปที่ 7.6 โฟลชาร์ทการทำงานของเครื่องโหมดที่ 3 80
รูปที่ 8.1 ตัวอย่างหน้าจอแสดงสถานะเริ่มต้นเพื่อที่จะเลือกโหมดการทำงาน 81
รูปที่ 8.2 ตัวอย่างหน้าจอแสดงผลและเป็นพื้นฐานการแสดงผลในทุกๆ โหมดการทำงาน 82
รูปที่ 8.3 ตัวอย่างหน้าจอแสดงการเพิ่ม ID ในฐานข้อมูลเครื่องแสกนลายนิ้วมือและสมาร์ตการ์ด 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

1.1 ความเป็นมาของโครงงาน

เนื่องจากในปัจจุบัน การพัฒนาอย่างรวดเร็วของเทคโนโลยีคอมพิวเตอร์ทำให้ระบบรักษาความปลอดภัยแบบใช้รหัสลับ,บัตรแถบแม่เหล็ก, การสแกนม่านตา,หรือจะเป็นการสแกนลายนิ้วมือซึ่งมีรหัสเป็นรหัสประจำตัว ที่มีเฉพาะบุคคลหนึ่งบุคคลใดเท่านั้นไม่สามารถลอกเลียนแบบกันได้ และในปัจจุบันราคาของเครื่องสแกนลายนิ้วมือ ก็มีราคาถูกลงสามารถนำมาประยุกต์ใช้ กับการตรวจสอบตัวบุคคลก่อนจะเข้าไปในสถานที่เข้มงวด ในการรักษาความปลอดภัยหรือใช้บันทึกเวลาทำงานของพนักงาน แทนเครื่องตอกบัตร การตรวจลายนิ้วมือจะดีกว่าการใช้วิธีตรวจบัตรประจำตัวเพียงอย่างเดียวซึ่งอาจปลอมแปลงกันได้ง่าย ปัจจุบันเครื่องเหล่านี้ถูกผลิตออกมามากมายหลายแบบ และการเก็บข้อมูลการเชื่อมต่อข้อมูลก็มีความก้าวหน้าไปอย่างมาก และน่าสนใจในการจะนำมาประยุกต์ใช้ในระบบรักษาความปลอดภัยมากขึ้น โดยเฉพาะเทคโนโลยีการเก็บข้อมูลแบบหนึ่งที่กำลังเป็นที่น่าสนใจเป็นอย่างมากในปัจจุบันคือสมาร์ตการ์ด ซึ่งเริ่มจะมีการใช้มากขึ้นในปัจจุบัน จากเหตุผลนี้จึงเป็นที่น่าสนใจในการที่จะ นำมาประยุกต์ใช้เครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ด ให้มีบทบาทในระบบรักษาความปลอดภัยที่มีประสิทธิภาพมากยิ่งขึ้น โดยในโครงงานนี้ได้มุ่งเน้นการประยุกต์ไปในงานด้านระบบความปลอดภัยให้แก่ผู้ใช้ ซึ่งในปัจจุบันนี้นับว่าเป็นสิ่งสำคัญที่สุดในชีวิตประจำวันของเราอย่างมาก สำหรับเครื่องตรวจสอบลายนิ้วมือในโครงงานนี้ได้ใช้โมดุลของ BKS-1700 ซึ่งใช้ตัวสแกนเนอร์เป็นแบบแสง ในส่วนสมาร์ตการ์ดที่ใช้ในโครงงานนี้ได้ประยุกต์มาจากบัตรโทรศัพท์ TOT Card ที่ไม่สามารถใช้ได้แล้ว ซึ่งบัตร TOT Card นี้ก็เป็นบัตรสมาร์ตการ์ดชนิดหนึ่งที่เราหาได้ง่าย และถือว่าเป็นการนำของที่หมดค่าแล้วมาประยุกต์ให้เกิดประโยชน์อีกทางหนึ่ง

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาโครงสร้างและการอินเตอร์เฟสเครื่องตรวจสอบลายนิ้วมือร่วมกับบัตรสมาร์ตการ์ด
2. ศึกษาการสื่อสารแบบอนุกรมมาตรฐาน RS-232 เพื่อส่งข้อมูลที่ได้จากเครื่องตรวจสอบลายนิ้วและบัตรสมาร์ตการ์ดเพื่อวิเคราะห์ผล
3. ศึกษาการเขียนโปรแกรมประยุกต์สำหรับงานด้านการจัดการฐานข้อมูลและการแสดงผล
4. ศึกษาการใช้งานเกี่ยวกับ mcs-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของโครงงาน

ทำการสร้างเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ด และนำข้อมูลที่ได้จากการสแกนของเครื่องตรวจสอบลายนิ้วมือ เพื่อนำมาเปรียบเทียบกับข้อมูลที่เก็บไว้ในฐานข้อมูลและแสดงผลของเจ้าของลายนิ้วมือและเจ้าของบัตรแต่ละใบที่ได้ทำการลงทะเบียนไว้กับระบบ และทำการฟ้องเมื่อลายนิ้วมือไม่ถูกต้องหรือที่ไม่ได้ลงทะเบียนกับระบบ

1.4 วิธีการดำเนินงาน

1. ศึกษาการอินเตอร์เฟสของเครื่องตรวจสอบลายนิ้วมือ.สมาร์ตการ์ด กับไมโครคอนโทรลเลอร์ และ คอมพิวเตอร์
2. เขียนโปรแกรมประยุกต์ในการรับข้อมูลจากเครื่องตรวจสอบลายนิ้วมือพร้อมทั้งนำข้อมูลที่ได้มาใช้ในการเปรียบเทียบข้อมูลที่สอดคล้องกันในฐานข้อมูลที่มีอยู่
3. ทดลองแล้วสรุปปัญหาที่เกิดขึ้น

1.5 ประโยชน์ที่ได้รับ

1. เพิ่มประสิทธิภาพในการรักษาความปลอดภัย
2. สร้างความสะดวกสบายแก่ผู้ใช้ในการทำกิจกรรมต่างๆในชีวิตประจำวันมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 เครื่องตรวจสอบลายนิ้วมือ

2.1 ประวัติโดยย่อของเครื่องตรวจสอบลายนิ้วมือ

เครื่องตรวจสอบลายนิ้วมือ สามารถนำมาใช้ประโยชน์ในด้านรักษาความปลอดภัยมากขึ้นในปัจจุบัน โดยเครื่องตรวจสอบลายนิ้วมือนี้แบ่งออกเป็น 2 ชนิด คือ แบบที่ใช้ในระบบ Fingerprint Identification และระบบ Fingerprint Verification ทั้งสองระบบนี้มีความแตกต่างกันคือ ระบบ Identification เป็นระบบที่ทำการเปรียบเทียบลายนิ้วมือที่ป้อนเข้ามากับลายนิ้วมือที่มีอยู่ในฐานข้อมูลของระบบทั้งหมด แล้วหาลายนิ้วมือที่ใกล้เคียงกันที่สุด ระบบนี้จะเป็นการเปรียบเทียบ matching แบบ one-to-many ส่วนแบบ verification นั้น จะเปรียบเทียบลายนิ้วมือที่ป้อนเข้ามากับลายนิ้วมือที่อยู่ในฐานข้อมูลไปทีละอัน เมื่อได้รับอันที่ตรงกันก็จะดึงข้อมูลที่เกี่ยวข้องกันออกมา ระบบนี้จะเป็นการเปรียบเทียบ matching แบบ one-to-one โดยเครื่องอ่านลายนิ้วมือที่ได้ออกแบบในการอ่านแบบ Total Reflection Method ได้โดยตรง แล้วทำการประมวลผลเบื้องต้น และ matching แบบ realtime ใช้เวลาเพียงชั่วครู่ก็สามารถตอบกลับมาได้ว่า เป็นลายนิ้วมือที่ต้องการหรือไม่ ระบบนี้มักใช้ในกิจการด้านรักษาความปลอดภัย เช่น เชื่อมกับระบบเปิดปิดประตูใหญ่ของสถานที่สำคัญ หรือใช้ในกิจการด้านรักษาความปลอดภัยของธนาคาร

2.2 ลักษณะลายนิ้วมือ

ลายนิ้วมือ เป็นความคดเคี้ยวอันน่าอัศจรรย์ ที่ธรรมชาติสร้างขึ้นบนร่างกายมนุษย์ ซึ่งแต่ละบุคคลจะถูกออกแบบมาไม่ซ้ำใครบนโลก จึงแสดงถึงความเป็นตัวคุณเท่านั้นโดยแท้จริง

มนุษย์จะมีร่องเล็กๆ ของผิวหนังบนนิ้วมือของแต่ละคนและลักษณะพิเศษเฉพาะนี้ บ่งบอกถึงเผ่าพันธุ์มนุษย์ได้อย่างดีที่สุด ซึ่งลักษณะของร่องทั้งหมด ที่เกิดขึ้นเป็นการรวมกันขององค์ประกอบทางสภาพแวดล้อม และพันธุ์กรรม รหัสในดีเอ็นเอจะทำให้เกิดลำดับชั้นขึ้นบนแนวผิวหนัง ที่มีการพัฒนารูปแบบตามพัฒนาการของทารกในครรภ์ ลักษณะพิเศษของแนวเส้นที่ก่อเป็นรูปเป็นร่างขึ้นคือผลลัพธ์ที่เกิดจากสถานการณ์ในการสุ่ม ตามตำแหน่งที่อยู่ของทารกในครรภ์ขณะนั้น และสัดส่วนที่แท้จริง รวมทั้งความหนาแน่นของน้ำคร่ำที่อยู่โดยรอบ จะเป็นตัวตัดสินใจเลือกเอาว่า จะแบ่งแต่ละส่วนอย่างไร ในการก่อตัวเป็นรูปร่างขึ้น

ดังนั้นจึงได้เกิดสิ่งที่ตามมาอีกมากมาย ที่นำไปความเป็นตัวตนของคุณ โดยในขั้นแรกนั้นมีองค์ประกอบทางสภาพแวดล้อมมากมายนับไม่ถ้วน ที่มีอิทธิพลต่อการสร้างลายนิ้วมือเช่นเดียวกับเงื่อนไขของสภาวะอากาศที่อาจจะเป็นลักษณะของเมฆหมอก หรือสภาพอากาศชายฝั่งทะเล ซึ่งกระบวนการพัฒนาทั้งหมดนั้นไม่มีระเบียบแบบแผนแน่นอน แม้แต่เส้นทางวิวัฒนาการของมนุษยชาติ นั้นคือไม่มีโอกาสที่รูปแบบนั้นจะตรงกันเป็นครั้งที่สอง ดังนั้นลายนิ้วมือจึงเป็นสิ่งถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างขึ้นมาแบบเดียวในโลกสำหรับแต่ละบุคคล แม้ว่าจะมีฝาแฝดก็ตาม ซึ่งลายพิมพ์นิ้วมือ ของทั้งสองคนดูเผินๆแล้วจะคล้ายกันมาก แต่เมื่อทำการตรวจสอบแล้ว จะพบว่าจะมีความแตกต่างกัน

นี้คือลักษณะพื้นฐานของลายนิ้วมือ ที่จะนำไปสู่การวิเคราะห์เพื่อระบบการรักษาความปลอดภัย เครื่องตรวจสอบลายนิ้วมือจะทำหน้าที่ในการวิเคราะห์บุคคล โดยการรวมกลุ่มตัวอย่างลายนิ้วมือ และทำการเปรียบเทียบกับตัวอย่างอื่นๆที่อยู่ในฐานข้อมูล

2.3 ความรู้เบื้องต้นของลายนิ้วมือ

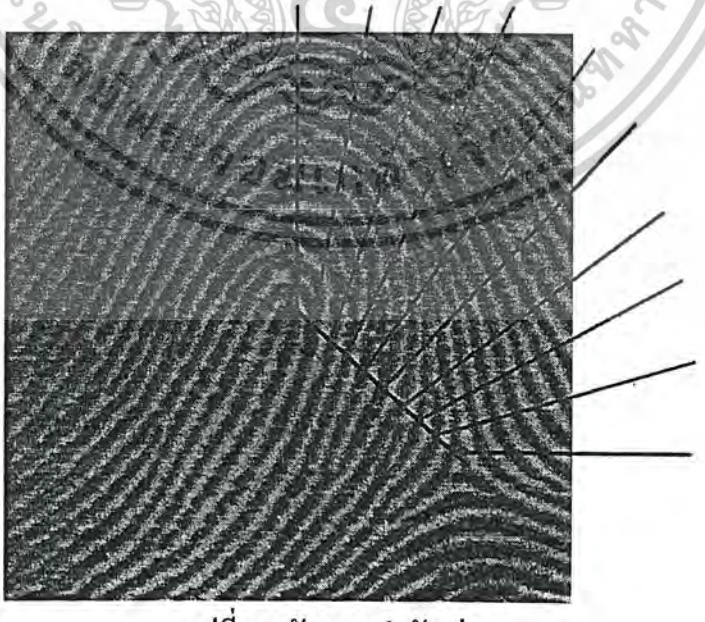
บริเวณปลายนิ้วมือของมนุษย์โดยทั่วไปจะเห็นลายนิ้วมือที่มีลักษณะประกอบไปด้วยเส้น 2 ลักษณะ คือ เส้นนูน (Ridges) และ เส้นร่อง (Furrows) ซึ่งเส้นทั้ง 2 ลักษณะจะอยู่สลับกันไปตลอด จุดลักษณะสำคัญบนลายนิ้วมือ (Characteristics) คือ ตำหนิต่างๆบนลายนิ้วมือ สามารถแบ่งได้เป็น 2 ลักษณะดังนี้

2.3.1 ตำหนิและลักษณะต่างๆ

ของลายเส้นทั่วๆ ไปเช่น เส้นตรง, เส้นโค้ง, จุด, เส้นแตก, เส้นวกกลับ,เส้นเวียน, เส้นขาด, เส้นทะเลสาบ, เส้น 2 เส้นมาพบกัน (เส้นหักมุม)

2.3.2 ลักษณะพิเศษบางอย่าง เช่น

- ไบฟูร์เคชัน คือ เส้นขอบหนึ่งที่ได้ถูกแยกออกเป็น 2 เส้นหรือมากกว่า 2 เส้น
- ไดเวอร์เจนซ์ คือ เส้นขอบที่ได้วิ่งขนานกันมาหรือเกือบจะขนานและได้แยกถ่างออกไป
- จุดมินูเทีย คือ จุดบนปลายเส้นหยุดหรือเส้นแยก



รูปที่ 2.1 ลักษณะสำคัญต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

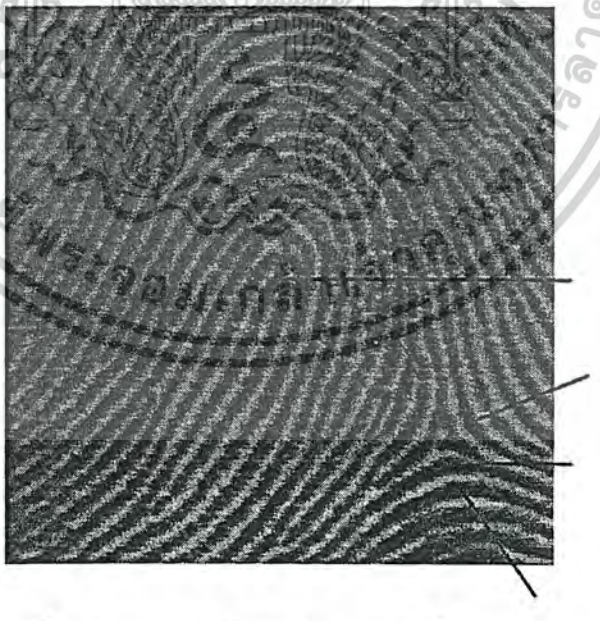
จากรูปที่ 2.1 อธิบายความสัมพันธ์ของเส้นต่างๆได้ดังนี้ (เรียงจากด้านซ้ายของรูปไปด้านขวาของรูป)

- 1 และ 2 เป็นไบฟูร์เคชัน
- 2 และ 3 เป็นไบฟูร์เคชัน
- 4 เป็น เส้นตรง
- 5 และ 6 เป็นไบฟูร์เคชัน
- 7 และ 8 เป็นไบฟูร์เคชัน
- 9 เป็น จุดมินูเทีย
- 10 เป็น เส้นวกกลับ

2.4 คำจำกัดความที่สำคัญบนลายนิ้วมือ

เป็นการอธิบายคุณลักษณะหลักสำคัญใหญ่ๆ ที่ต้องศึกษาและทำความเข้าใจเพราะมีคุณประโยชน์ที่แสดงให้เห็นถึงความแตกต่างของแต่ละลายนิ้วมือซึ่งมีอยู่ 4 ข้อ ได้แก่

- เส้นขอบ (Type Line)
- สันดอน (Delta)
- จุดใจกลาง (Core)
- บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area)



รูปที่ 2.2 ความแตกต่างสำคัญๆบนลายนิ้วมือ

2.4.1 เส้นขอบ (Type Line)

คือ เส้นคู่ขนานคู่ในสุด ซึ่งได้คู่กันมาพอสมควรแล้วแยกตัวออกเพื่อจะโอบล้อมหรือพยายามโอบล้อมบริเวณลายนิ้วมือที่อยู่ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 สันดอน (Delta)

คือ ลายเส้นในลายนิ้วมือ ซึ่งอยู่ตรงหน้าและใกล้ที่สุด กับกึ่งกลางของปากทางแยกของเส้นขอบ

2.4.3 จุดใจกลาง (Core)

คือ จุดใดบนปลายเส้นหรือบนบ่าหรือไหล่ของเส้นวกกลับรูปในสุดและต้องอยู่ภายในของลายนิ้วมือ

2.4.4 บริเวณลายนิ้วมือที่อยู่ภายใน (Pattern Area)

คือ พื้นที่บริเวณภายในของลายนิ้วมือที่ถูกเส้นขอบโอบล้อม

รูปที่ 2.3 Pattern Area

2.5 แบบของลายนิ้วมือ แบ่งเป็น 4 กลุ่มดังนี้

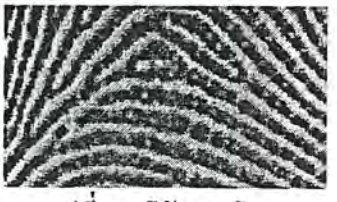
2.5.1 กลุ่มที่ 1 เส้นโค้ง (Arch) ประกอบด้วย

- โค้งราบ (Plain Arch) คือ ลายเส้นวิ่งหรือไหลออกไปข้างหนึ่ง ไม่เกิดมุมแหลมหรือพุ่งขึ้นตรงกลาง

รูปที่ 2.4 โค้งราบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โค้งกระโจม (Tented Arch) คือ ลายเส้นตรงกลางเกิดเป็นลายเส้นพุ่งขึ้นจากแนวนอนเป็นมุมแหลมหรือมุมฉาก



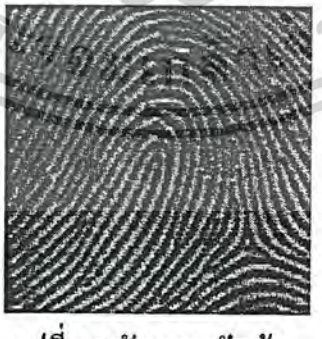
รูปที่ 2.5 โค้งกระโจม

2.5.2 กลุ่มที่ 2 ลูปหรือมัดหวาย (Loop)

- มัดหวายปัดขวา (Right Slant Loop) มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อย 1 เส้นมีทิศทางไปด้านขวา

รูปที่ 2.6 มัดหวายปัดขวา

- มัดหวายปัดซ้าย (Left Slant Loop) มีสันดอนเพียงจุดเดียว มีเส้นวกหลักที่สมบูรณ์อย่างน้อย 1 เส้นมีทิศทางไปด้านซ้าย



รูปที่ 2.7 มัดหวายปัดซ้าย

- มัดหวายคู่หรือมัดหวายแฝด (Double Loop) มีลักษณะคล้ายกับลายนิ้วมือแบบมัดหวายข้างบน แต่มากอดหรือกล้ากันจนเกิดมีสันดอน 2 จุด โดยไม่จำเป็นต้องมีขนาดเท่ากัน ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.8 มัดหวายแฝด

2.5.3 กลุ่มที่ 3 ก้นหอย (Whorl)

ลายนิ้วมือที่มีเส้นเวียนรอบเป็นวงจร ลักษณะเหมือนลานนาฬิกา, รูปไข่, วงกลม, หรือลักษณะอื่นๆ ประกอบด้วย

- ก้นหอยธรรมดา (Plain Whorl)

รูปที่ 2.9 ก้นหอยธรรมดา

- ก้นหอยกระเป๋ากลางปัดขวา (Right Central Pocket)

รูปที่ 2.10 ก้นหอยกระเป๋ากลางปัดขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ก้นหอยกระเป๋ากลางปัดซ้าย (Left Central Pocket)

รูปที่ 2.11 ก้นหอยกระเป๋ากลางปัดซ้าย

- ก้นหอยกระเป๋าข้างปัดขวา (Right Lateral Pocket)

รูปที่ 2.12 ก้นหอยกระเป๋าข้างปัดขวา

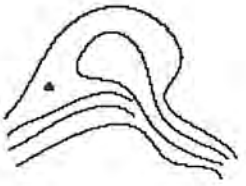
- ก้นหอยกระเป๋าข้างปัดซ้าย (Left Lateral Pocket)

รูปที่ 2.13 ก้นหอยกระเป๋าข้างปัดซ้าย

2.5.4 กลุ่มที่ 4 ซับซ้อน (Accidental Whorl)

ลายนิ้วมือที่มีลักษณะพิเศษที่ไม่จัดเข้าเป็นลายนิ้วมือชนิดหนึ่งชนิดใดโดยเฉพาะ ประกอบด้วย ลายนิ้วมือ 2 แบบมาผสมกันและมีสันดอน 2 สันดอน หรือมากกว่าเช่น กรณีที่ไม่สามารถเข้ากับลายนิ้วมือกลุ่มที่กล่าวมาแล้วข้างต้นไม่ได้เลย โดยมีความยุ่งเหยิงและเป็นรูปแบบที่ไม่แน่นอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.14 กลุ่มซับซ้อน

2.6 ชนิดของเครื่องตรวจสอบลายนิ้วมือ มีหลายชนิดดังนี้คือ

- เครื่องตรวจสอบลายนิ้วมือที่ใช้แสง
- เครื่องตรวจสอบลายนิ้วมือที่ใช้คลื่นเสียง
- เครื่องตรวจสอบลายนิ้วมือที่ใช้สนามไฟฟ้า
- เครื่องตรวจสอบลายนิ้วมือที่ใช้ความจุไฟฟ้า
- เครื่องตรวจสอบลายนิ้วมือที่ใช้ความร้อน

เครื่องตรวจสอบลายนิ้วมือ แม้จะมีอยู่หลายชนิดแต่ที่นิยมใช้ทั่วไปมากกว่าชนิดอื่นคือแบบที่ใช้แสง กับที่ใช้ความจุไฟฟ้า

2.6.1 เครื่องตรวจสอบลายนิ้วมือที่ใช้แสง

ระบบเครื่องตรวจสอบลายนิ้วมือ มีงานพื้นฐาน 2 อย่างที่จะต้องทำคือ ระบบต้องการที่จะใช้ภาพของลายนิ้วมือที่ปรากฏขึ้น และต้องการที่จะกำหนดทั้งรูปแบบของรอยสันนูนและรอยร่องลึกในภาพนี้ เพื่อให้ตรงกันกับรูปแบบของสันนูนและร่องลึกในภาพที่สแกนไว้ก่อนหน้านี้

หัวใจของเครื่องตรวจสอบลายนิ้วมือแบบใช้แสง ก็คืออุปกรณ์ถ่ายภาพ CCD (Charge Coubled Device) ซึ่งก็เหมือนกันกับระบบเซ็นเซอร์แสงที่ใช้ในกล้องดิจิตอลและกล้องถ่ายวีดีโอทั่วไป CCD อย่างง่ายจะใช้ไดโอดไวแสงที่เรียกว่าโฟโต้ไซต์วางเรียงติดต่อกันเป็นแถว ซึ่งจะกำเนิดสัญญาณไฟฟ้าตามปริมาณของแสงที่ตกกระทบ แต่ละโฟโต้ไซต์จะบันทึกเป็นพิกเซล จุดเล็กๆ จะเป็นตัวแสดงถึงแสงที่ตกกระทบไปยังจุดนั้น แล้วนำพิกเซลที่สว่างและมืดมารวมกันก่อเป็นรูปของสิ่งที่สแกน (ในที่นี้คือนิ้วมือ) โดยทั่วไปตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอล ในระบบเครื่องตรวจสอบลายนิ้วมือ จะทำการประมวลผลสัญญาณไฟฟ้าอนาล็อกให้เป็นดิจิตอลเพื่อแสดงความหมายของภาพนั้นๆ

กระบวนการสแกนจะเริ่มต้นเมื่อคุณวางนิ้วมือลงไปบนแผ่นแก้ว ซึ่งจะมีกล้อง CCD ทำการถ่ายภาพ เครื่องตรวจสอบลายนิ้วมือจะใช้แหล่งกำเนิดแสงในตัวมันเอง โดยทั่วไปจะเป็น LED เรียงกันเป็นแถวเพื่อส่องแสงไปยังลายสันนูนของนิ้วมือ ซึ่งอันที่จริงแล้วระบบ CCD จะทำให้เกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพลายนิ้วมือแบบกลับกัน นั้นคือพื้นที่ที่มืดมากกว่าจะแสดงถึงแสงที่สะท้อนน้อย(ลายร่องลึกนิ้วมือ)

ก่อนการเปรียบเทียบลายนิ้วมือกับฐานข้อมูลที่บันทึกไว้ โปรเซสเซอร์ในเครื่องตรวจสอบลายนิ้วมือจะต้องแน่ใจว่า CCD มีการจับภาพที่คมชัดมันจะทำการตรวจสอบค่าเฉลี่ยพิกเซลมืด หรือค่าทั้งหมดในตัวอย่าง และมันจะไม่ยอมรับการสแกนถ้าภาพมืดมากหรือสว่างมากกว่าตัวอย่าง และถ้าภาพนั้นถูกปฏิเสธเครื่องตรวจสอบลายนิ้วมือจะทำการปรับการปล่อยแสง เพื่อให้ได้แสงมากหรือน้อย และจะพยายามสแกนอีกครั้ง

ถ้าระดับความมืดนั้นเพียงพอที่จะใช้ได้ ระบบเครื่องตรวจสอบลายนิ้วมือจะทำงานต่อไปเพื่อตรวจสอบความชัดเจนของภาพ ตัวโปรเซสเซอร์จะดูที่เส้นแนวต่างๆที่ตรงกันโดยการเลื่อนไปตามแนวนอนและแนวตั้งบนภาพ ถ้าภาพลายนิ้วมือนั้นมีชัดเจนดี ก็จะตรงกันกับแนวรอยสันนูน ทำให้เกดส่วนที่สลับกันของพิกเซลที่มืดและพิกเซลที่สว่าง ถ้าตัวโปรเซสเซอร์พบว่าภาพนั้นสว่างชัดและมีการเปิดแสงที่ถูกต้อง มันจะทำงานต่อไปเพื่อเปรียบเทียบภาพลายนิ้วมือที่สแกนกับลายนิ้วมือในฐานข้อมูล ซึ่งกระบวนการนี้จะมีความพิถีพิถันมาก

2.6.2 เครื่องตรวจสอบลายนิ้วมือที่ใช้ความจุไฟฟ้า

หลักการก็คล้ายกันแบบที่ใช้แสงคือ เครื่องตรวจสอบลายนิ้วมือที่ใช้ความจุไฟฟ้า จะทำให้เกิดเป็นภาพรอยนูนและร่องลึกเพื่อสร้างเป็นรูปลายนิ้วมือขึ้น แต่แทนที่จะใช้การตรวจวัดด้วยแสงก็จะเป็นการใช้กระแสไฟฟ้าแทน

โดยเซนเซอร์จะทำมาจากชิปสารกึ่งตัวนำหนึ่งตัวหรือมากกว่า ที่บรรจุด้วยเซลล์เล็กๆที่วางเรียงติดกันอยู่ภายในแต่ล่ะเซลล์จะประกอบไปด้วยแผ่นตัวนำ 2 แผ่น แล้วหุ้มด้วยชั้นที่เป็นฉนวน โดยที่แต่ล่ะเซลล์จะมีขนาดที่เล็กมาก ส่วนขั้วอินเวอร์ติ้งจะต่ออยู่กับแรงดันอ้างอิงและเป็นลูปป้อนกลับ ซึ่งลูปป้อนกลับนี้ก็ยังต่อเข้ากับเอาต์พุต รวมทั้งแผ่นตัวนำทั้งสองด้วย

จะเห็นว่าแผ่นตัวนำทั้งสอง คือรูปแบบพื้นฐานของตัวเก็บประจุนั้นเอง ซึ่งมีส่วนประกอบทางไฟฟ้าที่สามารถเก็บประจุไว้ได้ พื้นผิวของนิ้วมือจะเปรียบเสมือนแผ่นที่ 3 ของตัวเก็บประจุ ที่ถูกแบ่งแยกโดยชั้นฉนวนในโครงสร้างเซลล์ และเมื่อเกิดช่องอากาศในบริเวณที่เป็นช่องลึก (โดยการเลื่อนนิ้วมือเข้าไปใกล้หรือห่างจากแผ่นตัวนำ) จะเป็นการเปลี่ยนแปลงค่าความจุรวมของตัวเก็บประจุ ซึ่งปริมาณความจุของตัวเก็บประจุในเซลล์ ที่อยู่ใต้รอยสันนูนจะมีค่าความจุมากกว่าตัวเก็บประจุในเซลล์ที่อยู่ใต้ลายร่องลึก

ในการสแกนนิ้วมือ ครั้งแรกตัวประมวลผลจะปิดสวิตช์รีเซตในแต่ละเซลล์ ซึ่งจะเป็นการชอร์ตตัวขยายแต่ละตัว ให้อินพุตและเอาต์พุตของไอซีมีความสมดุล เมื่อเปิดสวิตช์ครั้งต่อไป ตัวประมวลผลจะทำการจ่ายประจุที่คงที่ค่าหนึ่งไปยังไอซี และตัวเก็บประจุก็จะเริ่มทำการเก็บประจุ ค่าความจุของลูปป้อนกลับ จะมีผลกระทบต่อแรงดันที่ตัวเก็บประจุอินพุต ซึ่งจะส่งผลต่อเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อระยะทางจากตัวเก็บประจุที่ไปยังนิ้วมือ จะมีผลให้ค่าความจุเปลี่ยนแปลง ซึ่งรอยสันนูนจะให้แรงดันเอาต์พุตที่มีความแตกต่างกันมากกว่ารอยร่องลึกของนิ้วมือ

ตัวประมวลผลของเครื่องสแกน จะอ่านค่าแรงดันเอาต์พุตนี้ และทำการกำหนดคุณลักษณะทั้งรอยสันนูนและรอยร่องลึก โดยการอ่านค่าทุกๆ เซลล์ในแถวของตัวเซนเซอร์ตัวประมวลผลสามารถที่จะรวมภาพทั้งหมดเข้าด้วยกัน เหมือนกับภาพที่เกิดขึ้นจากเครื่อง สแกนแบบที่ใช้แสง

ประโยชน์สำคัญของเครื่องสแกนแบบที่ใช้ความจุไฟฟ้าก็คือ มันต้องการรูปร่างนิ้วมือจริงมากกว่ารูปแบบของความสว่าง และมือที่ประกอบกันขึ้น เพื่อให้มองเห็นเป็นลายพิมพ์ของนิ้วมือ ทำให้ระบบนี้ยากที่จะหลอกได้ แถมยังใช้ชิปสารกึ่งตัวนำที่มีขนาดเล็กกว่าอุปกรณ์ CCD เครื่องสแกนแบบที่ใช้ความจุไฟฟ้าจึงมีแนวโน้มที่จะมีขนาดกะทัดรัดกว่าเครื่องสแกนแบบที่ใช้แสง

2.7 การวิเคราะห์เปรียบเทียบลายนิ้วมือ

การตรวจสอบลายนิ้วมือส่วนมาก จะทำการเปรียบเทียบคุณลักษณะพิเศษของลายนิ้วมือ ที่เรียกว่า มินูชิอี (**Minutiae)** ซึ่งก็คือรายละเอียดปลีกย่อย มนุษย์และนักสืบคอมพิวเตอร์จะทำการรวมจุดที่ปลายสุดของรอยเส้นนูน หรือรอยแตกที่แยกออกเป็นสองทาง (Bifurcation) ซึ่งกลุ่มของจุดเหล่านี้และคุณลักษณะพิเศษอื่นๆ นั้นบางที่อาจเรียกว่า Typica

ซอฟต์แวร์ของระบบเครื่องสแกน จะใช้อัลกอริทึมที่มีความซับซ้อนสูงมาก เพื่อการจดจำและวิเคราะห์รายละเอียดปลีกย่อยเหล่านี้ แนวคิดพื้นฐานก็คือ การวัดตำแหน่งที่สัมพันธ์กันของมินูชิอี ในทำนองเดียวกันกับวิธีที่ใช้ในการจดจำส่วนของท้องฟ้า โดยมีตำแหน่งที่สัมพันธ์กับดวงดาว วิธีง่ายๆ ที่จะทำให้มองเห็นภาพการก่อรูปขึ้นเป็นมินูชิอีต่างๆ นั้น ก็โดยการลากเส้นตรงแนวระหว่างรอยสันนูนเหล่านั้น ถ้าทั้งสองลายพิมพ์มีปลายเส้นนูน 3 เส้นที่แยกออกเป็น 2 ทาง สร้างเป็นรูปที่เหมือนกันด้วยมิติที่เหมือนกัน ก็มีความน่าจะเป็นสูงมากที่จะมาจากแม่พิมพ์เดียวกัน

เพื่อดูว่าลายพิมพ์นั้นตรงกันหรือไม่ ระบบเครื่องสแกนจะไม่ค้นหารูปแบบทั้งหมดของทุกมินูชิอีทั้งในลายพิมพ์ตัวอย่างและที่บันทึกไว้ มันจะค้นหาอย่างง่ายๆ ด้วยปริมาณของรูปแบบมินูชิอีที่เพียงพอจะเห็นว่าลายพิมพ์ทั้งสองมีส่วนที่เหมือนกัน ซึ่งจำนวนที่ถูกต้องแน่นอน จะแปรผันไปตามการโปรแกรมของเครื่องสแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ข้อดีและข้อเสียของเครื่องตรวจสอบลายนิ้วมือ

มีวิธีการต่างๆ มากมาย ที่ระบบรักษาความปลอดภัยสามารถที่จะพิสูจน์ตัวบุคคล เพื่ออนุญาตให้เป็นผู้ใช้ ซึ่งระบบส่วนใหญ่จะมองหาสิ่งต่อไปนี้คือ

- คุณมีอะไร
- คุณรู้อะไร
- คุณเป็นใคร

ในส่วนของการมองหาว่า "คุณมีอะไร" นั้นก็หมายถึงว่าคุณจะต้องมีบางสิ่งบางอย่างที่เป็นเครื่องแสดงถึงการเป็นสมาชิกอย่างเช่น สมาร์ตการ์ด "คุณรู้อะไร" นั้นคือระบบจะร้องข้อรหัสผ่านหรือรหัส PIN ที่มีข้อมูลต้องกันกับบัตร "คุณเป็นใคร" ระบบจะมองหาสิ่งที่แสดงลักษณะทางกายภาพ ที่จะระบุได้ว่าเป็นตัวคุณ อย่างเช่น ลายนิ้วมือ เสียง หรือม่านตา เป็นต้น

สำหรับเครื่องสแกนลายนิ้วมือ จัดอยู่ในส่วนของระบบที่ต้องการรู้ว่า "คุณเป็นใคร" ซึ่งมีข้อดีดังนี้

- คุณลักษณะทางกายภาพ หลอกลวงได้ยากกว่าการใช้สมาร์ตการ์ด
- ไม่สามารถเดารูปแบบของลายนิ้วมือได้ เหมือนกับการเดารหัสผ่าน
- ไม่สามารถที่จะวางลายนิ้วมือ, ม่านตา, เสียง ไว้อย่างผิดที่ผิดทางได้เหมือนกับการวางการ์ด
- ลายนิ้วมือไม่สามารถลบลืมได้เหมือนกับการลืมรหัสผ่าน

ถึงแม้ว่าการระบุตัวบุคคล ด้วยเทคโนโลยีไบโอเมตริกจะมีคุณประโยชน์ และให้ความแน่นอนอย่างที่ไม่มีทางผิดพลาดได้ แต่ก็ยังมีข้อเสียบางอย่างคือ เครื่องสแกนแบบที่ใช้แสงไม่สามารถแยกแยะระหว่างภาพถ่ายนิ้วมือกับนิ้วมือได้ด้วยตัวมันเอง ส่วนเครื่องสแกนแบบที่ใช้ความจุไฟฟ้า บางครั้งอาจถูกหลอกได้โดยการใช้แบบพิมพ์นิ้วมือปลอม เพื่อให้ใครบ้างคนสามารถเข้าไปเป็นผู้ใช้ระบบได้ โดยที่ผู้นั้นไม่ใช่เจ้าของลายนิ้วมือที่แท้จริง อย่างเช่นในภาพยนตร์ ที่อาชญากรทำการตัดนิ้วใครบ้างคนไป เพื่อนำนิ้วไปเข้าเครื่องสแกน แล้วผ่านระบบรักษาความปลอดภัยเข้าไปได้ แต่เครื่องสแกนบางชนิดจะมีการเพิ่มตัวเซนเซอร์ อุณหภูมิเพื่อตรวจจับว่าเป็นนิ้วมือของผู้ที่ยังมีชีวิตอยู่ โดยการการตรวจจับอุณหภูมิความร้อนจากร่างกาย แต่ระบบเหล่านี้ก็ยังสามารถถูกหลอกได้โดยใช้แบบหล่อนิ้วมือยาง แล้วสวมทับกับนิ้วมือของคนที่ต้องการลักลอบเข้าระบบได้

เพื่อให้ระบบรักษาความปลอดภัยมีความน่าเชื่อถือสูง จึงเป็นแนวความคิดที่ดีที่จะรวมเอาการวิเคราะห์ทางชีวภาพเข้ากับสิ่งที่แสดงความหมายในการระบุตัวบุคคลแบบที่ใช้ทั่วไป อย่างเช่น รหัสผ่าน (อย่างที่เครื่อง ATM ต้องมีการร้องขอบัตร พร้อมกับการใช้รหัสผ่าน)

ปัญหาที่แท้จริงของระบบรักษาความปลอดภัยด้วยวิธีการทางชีวภาพก็คือ ของมีเขตของความเสียหายท่วมอาจบระเมน เดเม้อมเครบางคนทาการข เมยข้อมูลท่ใช้ระบุตวบุคคล เบ ถ้าบตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครดิตหายหรือเผลอบอกรหัสผ่านกับใครไป ก็สามารถที่จะสั่งอายัดบัตร และทำบัตรใหม่พร้อมกับขอรหัสผ่านใหม่ได้ แต่ถ้ามีใครขโมยลายนิ้วมือคุณไป นับว่าชีวิตคุณโชคร้ายเป็นอย่างมาก คุณอาจจะไม่สามารถใช้ลายพิมพ์นิ้วมือของคุณเพื่อระบุความเป็นตัวคุณได้อีกต่อไป เพราะไม่มีทางที่คุณจะสร้างลายนิ้วมือแบบใหม่เพื่อใช้แทนอันเก่าได้

แม้ว่าจะมีข้อเสียเกิดขึ้น เครื่องสแกนลายนิ้วมือและระบบไบโอเมตริก ก็นับว่าเป็นสิ่งที่ใช้แสดงความหมายได้อย่างดีเยี่ยมในการระบุตัวบุคคล ในอนาคตสิ่งเหล่านี้อาจจะกลายมาเป็นส่วนหนึ่ง ที่มิอาจแยกออกไปจากการดำเนินชีวิตได้ อย่างกุญแจ, บัตร ATM และรหัสผ่าน ที่ทุกคนต้องพกติดตัวอย่างในปัจจุบัน

2.9 รูปแบบพื้นฐานที่ควรรู้เกี่ยวกับ Biometric

การลงทะเบียน (Enrollment) คือ ลักษณะการทำงานที่ทำการสแกนลายนิ้วมือ เพื่อบันทึกเป็นฐานข้อมูลลงไปในหน่วยความจำภายในบอร์ด

การพิสูจน์ยืนยัน (Verification) คือลักษณะการทำงานโดยการป้อน PIN Code หรือในที่นี้ก็คือลายนิ้วมือของเจ้าของ เพื่อนำไปเปรียบเทียบกับลายนิ้วมือที่เก็บบันทึกไว้ก่อน แล้วในฐานข้อมูลหน่วยความจำที่แสดงได้ว่าถูกต้องหรือไม่ กับลายนิ้วมือที่สแกนเข้าไปใหม่ซึ่งจะเป็น ลักษณะการใช้งานแบบ 1 ต่อ 1 หรือเฉพาะบุคคลเพียงคนเดียว

การสืบค้น (Indentification) คือ ลักษณะการทำงานของผู้ใช้งานเมื่อสแกนลายนิ้วมือ เพื่อเปรียบเทียบกับอีกหลาย ๆ ลายนิ้วมือ ที่เก็บบันทึกข้อมูลไว้ ในลักษณะของการสืบค้นจากลายนิ้วมือ กลาย ๆ อัน เพื่อยืนยันว่าเป็นกลุ่มที่สามารถเข้าใช้งานได้

รูปลายนิ้วมือ (Fingerprint template) คือ ส่วนของการอธิบายข้อมูลที่ถูกบันทึกเก็บ ไว้เป็นฐานข้อมูลอ้างอิงในลักษณะข้อมูลทางคณิตศาสตร์ที่เป็นการใช้เทคนิค และวิธีการประมวลผลจากรูปภาพลายนิ้วมือตามลักษณะ อัลกอริธึมแบบพิเศษ ที่เป็นขบวนการในการจัดเก็บข้อมูลที่จะสามารถนำเอามาเปรียบเทียบได้ในภายหลัง

เครื่องตรวจสอบลายนิ้วมือคอร์ (Fingerprint Core) เป็นส่วนที่ใช้อธิบายถึงลักษณะ ที่ช่วยให้ สังเกตุเห็นคุณสมบัติโดยทั่วไปของภาพที่ทำการบันทึก ที่อยู่ในลักษณะของลายนิ้วมือที่มีลักษณะ เป็นเส้นโค้ง ซึ่งเป็นส่วนที่สำคัญที่จะเป็นตัวยืนยันหรือข้อมูลสำคัญในการตรวจสอบว่าเป็นของคน ๆ นั้น หรือเฉพาะบุคคลไป

2.10 คุณภาพของลายนิ้วมือ

ในการทำความเข้าใจถึงผลกระทบที่มีต่อคุณภาพของลายนิ้วมือ ที่เก็บในรูปแบบของภาพแบบหนึ่งนั้น สำหรับตัวสแกนของเครื่องตรวจสอบลายนิ้วมือเอง ก็มี พารามิเตอร์ตัวที่สามารถบ่งบอกถึงคุณภาพและ การใช้งานของตัวสแกนนี้ได้ สำหรับค่า FRR และ FAR ที่ตัวเครื่องตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลายนิ้วมือแต่ละตัวจะแสดงค่า ออกมาเพื่อบ่งบอกถึงคุณภาพในเรื่องของความปลอดภัยหรือการยอมรับผู้ใช้งานที่มีการลงทะเบียนไว้แล้ว

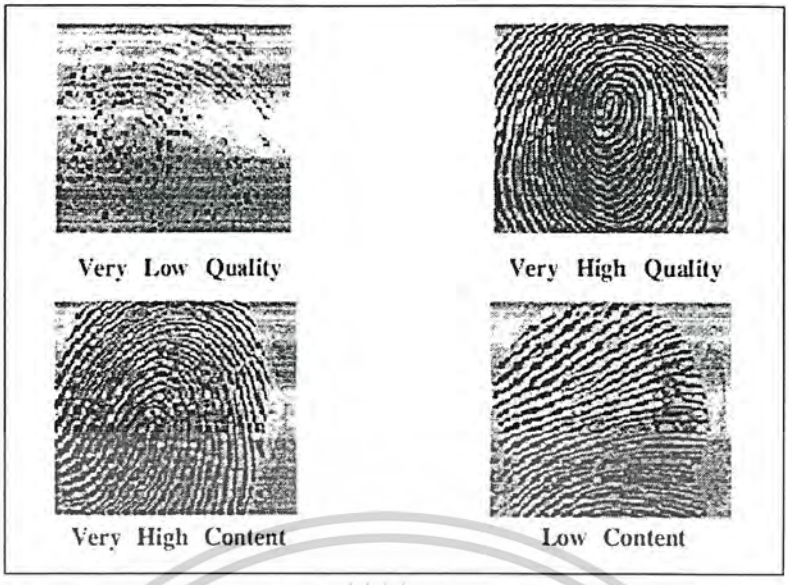
ค่า FRR (False Rejection Rate) คือ ค่าของการปฏิเสธหรือการไม่ยอมรับผู้ใช้งานโดยค่า FRRนี้จะมีอัตราส่วนแสดงอยู่เช่น 1/100 นั้นหมายถึงการไม่ยอมรับใน 100 ภาพจะมีการคัดออก 1 ภาพซึ่งเป็นระดับความปลอดภัยระดับ 1 สูงที่สุด

ค่า FAR (False Acceptance Rate) คือค่าของการยอมรับผู้ใช้งานที่จะเป็นตัวแปรบ่งบอกถึงคุณภาพ ค่าสำหรับพารามิเตอร์ตัวนี้ เมื่อมีค่ามากจะถือว่ามีคุณภาพดีที่สุดอย่างเช่น 1/20,000 หมายถึงค่า 20,000 ภาพจะยอมรับได้เพียง 1 ภาพเท่านั้น ก็คือต้องมีภาพที่มีความเหมือน ,หรือความแตกต่างน้อยที่สุดจึงจะยอมรับได้

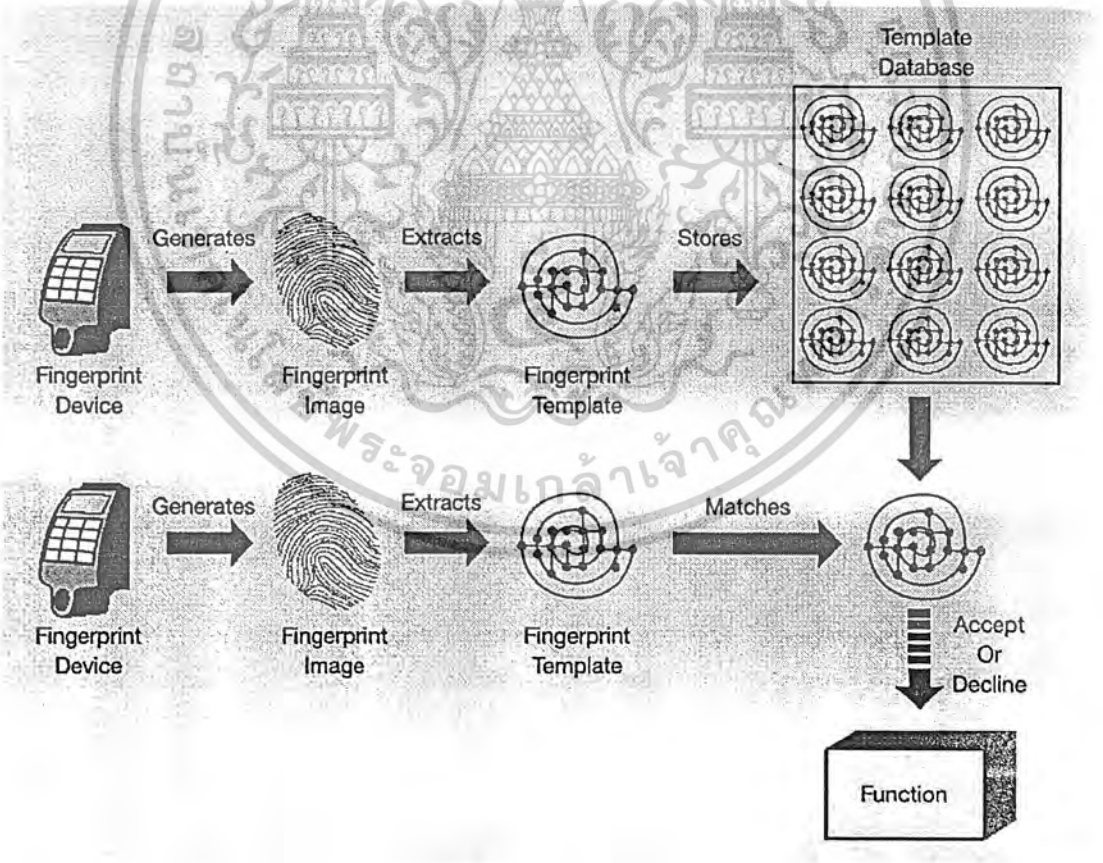
ค่า EER (Equal Error Rate) คือค่าความผิดพลาดจากความคลายคลึงของภาพที่เก็บและที่นำมาตรวจสอบ

ดังนั้นคงจะพอเห็นได้ว่าในการนำเครื่องตรวจสอบลายนิ้วมือไปใช้งานควร ที่จะกำหนดพารามิเตอร์ 2 ตัว ข้างต้น (FRR และ FAR) ตัวนี้ให้ เหมาะสมกับประเภทของการใช้งานด้วยในส่วนของคุณภาพลายนิ้วมือ ซึ่งก็จะมีตัวบ่งชี้ให้ผู้ใช้งาน สามารถนำไปเป็นตัวพิจารณา หรือเลือกใช้ประเภทของเครื่องตรวจสอบลายนิ้วมือได้ค่า คุณภาพของภาพนั้นจะมีค่าอยู่ในช่วง 0-100 ช่วงที่สามารถ ยอมรับได้ก็จะอยู่ประมาณ 40 ขึ้นไปซึ่งจะแสดงดังตารางคุณภาพ และความละเอียดของภาพ ทั้งนี้ยังมีค่าความชัดเจน หรือความสมบูรณ์ของภาพลายนิ้วมือเข้ามาเกี่ยวข้องด้วยที่มีช่วง ค่าของการยอมรับได้อยู่ที่ 0-100 และค่าที่ยอมรับได้จะอยู่ ประมาณ 20 ขึ้นไป ถ้าคุณภาพของรูปลายนิ้วมือไม่สมบูรณ์ หรือความคมชัดไม่ดีก็จะต้องทำการลงทะเบียนใหม่อีกครั้ง ซึ่งสาเหตุของการตรวจเช็ค และได้ภาพที่ไม่สมบูรณ์นี้ ก็จะมี สาเหตุอยู่หลายประการ เช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 สภาพลายนิ้วมือที่แสดงการเปรียบเทียบคุณภาพของภาพที่ได้



รูปที่ 2.16 ขบวนการทำงานของการใช้งานเครื่องตรวจสอบลายนิ้วมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.11 รายละเอียดเครื่องตรวจสอบลายนิ้วมือที่ใช้ในโครงงาน

1. คุณสมบัติของเครื่องตรวจสอบลายนิ้วมือ

- ราคาถูกกว่าเครื่องในท้องตลาดทั่วไป
- ใช้หน่วยประมวลผลชนิด ARM9
- ลักษณะตัวเครื่องใช้เซ็นเซอร์แบบแสง
- พื้นผิวเซ็นเซอร์ด้านหน้าเป็นกระจกแข็ง
- ภาพลายนิ้วมือที่สแกนได้คุณภาพสูง
- มีหน่วยประมวลผล และ หน่วยความจำติดตั้งบนตัวเครื่อง 8MB

2. รายละเอียดขอบเขตการทำงานเครื่อง

Parameter	Min	Typical	Max	Unit
Supply Voltage	4.75	5	5.25	V
Current at 5V	90	100	130	mA
Temperature Range	-20	-	60	^{0}C
Humidity	0	-	90	%

ตารางที่ 2.1 แสดงรายละเอียดขอบเขตการทำงานของเครื่อง

3. การเชื่อมต่อ

- 1 พอร์ท RS-232C
- ความเร็วที่ใช้เชื่อมต่อ 9600,19200,38400,115200 bps
- ความเร็วเริ่มต้น 19200 bps

	In/Out	Type	Description	Voltage
RXD	IN	-	232input signal	Max ±15V
TXD	OUT	-	232 Output Signal	Min ± 5V
SG	-	-	Signal ground	-

ตารางที่ 2.2 แสดงรูปแบบการเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61936

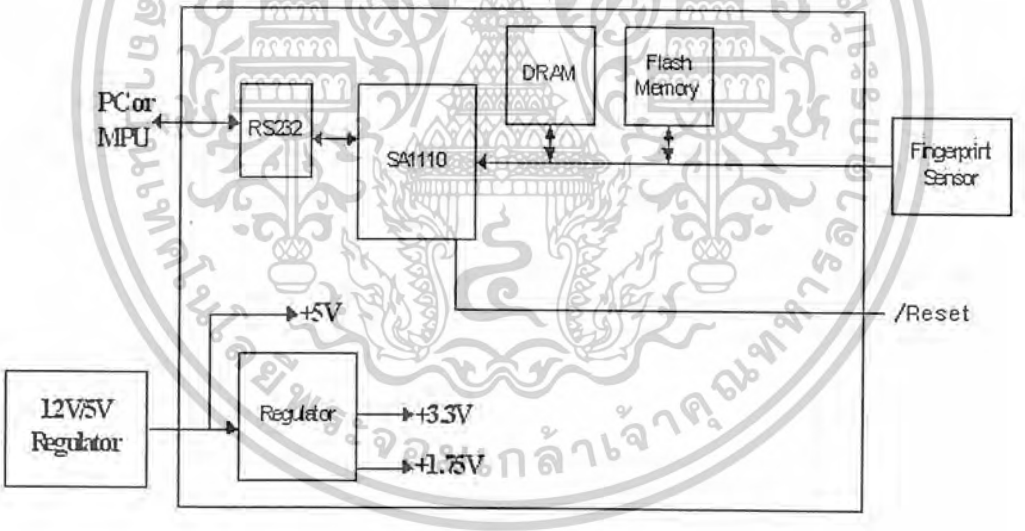
4. ความสามารถในการทำงานของเครื่อง

- ใช้เซ็นเซอร์ชนิดแสง
- ความละเอียด 500dpi
- พื้นที่สแกน 18มม.(สูง) x 16มม.(กว้าง)
- ขนาดข้อมูลลายนิ้วมือ 256 bytes
- ใช้เวลาในการสแกนโดยเฉลี่ย 1.0 วินาที
- ความผิดพลาดเนื่องจากการไม่ทำงานน้อยกว่า 0.01 %
- ความผิดพลาดเนื่องจากการยอมรับได้น้อยกว่า 0.001 %
- ความจุสูงสุดของผู้ใช้10,000 บุคคล (2 ลายนิ้วต่อบุคคล)

5. ขนาดของเครื่องตรวจสอบลายนิ้วมือ

- ตัวเซ็นเซอร์ลายนิ้วมือแสง : 21มม.(กว้าง) x 23มม.(ยาว) x 52มม.(สูง)
- ขนาดของช่องสแกน : 18มม.(กว้าง) x 16มม.(สูง)
- ขนาดของบอร์ด : 60มม.(กว้าง) x 43มม.(ยาว) x 8.6มม.(สูง) น้ำหนัก : 100 กรัม

6. แผนภาพโครงร่างบอร์ด



รูปที่ 2.17 แผนภาพโครงร่างบอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 สมาร์ตการ์ด (Smart card)

3.1 ความหมายของสมาร์ตการ์ด

Smart Card หมายถึง การ์ดที่มีหน่วยความจำ หรือ มี Microprocessor ฝังอยู่ในการ์ดอาจจะเป็น Chip หน่วยความจำชนิดที่ถูกโปรแกรมเรียบร้อยแล้วมาจากโรงงาน แต่ถ้าเป็นแบบ Microprocessor นั้นก็จะสามารถเพิ่มข้อมูล หรือลบข้อมูล หรือไม่เช่นนั้นก็จะสามารถปรับปรุงเปลี่ยนแปลง ข้อมูลบนตัวการ์ดนี้ อีกประเภทคือ การ์ดที่มีหน่วยความจำคงที หรือที่เรียกว่า memory-chip card เช่น การ์ดโทรศัพท์ เป็นต้น ลักษณะตัวการ์ด Smart card นั้นจะเป็นแผ่นพลาสติก ขนาดเท่ากับ บัตรเครดิต หรือ ขนาดใกล้เคียงกับนามบัตร ซึ่งเป็นขนาดมาตรฐานทั่วโลก ภายในการ์ดนี้ก็จะมีเนื้อที่ส่วนที่เป็นหน่วยความจำอัดอยู่บนการ์ด ซึ่งในส่วนนี้นี่เองที่จะเป็นส่วนบรรจุข้อมูลอยู่ภายใน หากจะเพิ่มเติมข้อมูล หรือ อ่านข้อมูลจากบัตรก็จะต้องมีเครื่องอ่านบัตร ที่เรียกว่าSmartCardReader

Smart Card นั้นจะไม่เหมือนกับการ์ดชนิดใช้แถบแม่เหล็ก เพราะ Smart Card นั้นจะมีข้อมูลที่จำเป็น และข้อมูลข่าวสารอื่นๆอยู่บนการ์ด ดังนั้นการอ่านข้อมูลจึงไม่ต้องย้อนกลับไปค้นข้อมูลจากศูนย์ข้อมูลอันเป็นการเสียเวลาเช่นเดียวกับการ์ดแถบแม่เหล็ก(เช่น บัตรเอทีเอ็ม) นี่ก็จะเป็นข้อดีของ Smart Card

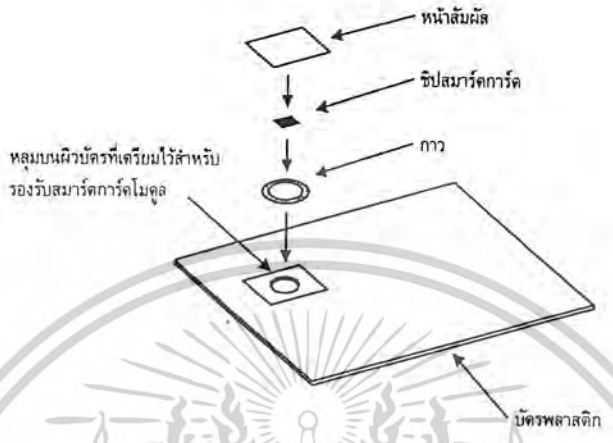
3.2 ข้อดีของสมาร์ตการ์ด

1. มีความไว้วางใจได้ดีกว่าบัตรที่ใช้แถบแม่เหล็ก
2. สามารถเก็บสะสมข้อมูลได้มากกว่าบัตรที่ใช้แถบแม่เหล็กเป็นร้อย ๆ เท่า
3. ลดโอกาสที่จะเข้าไปยุ่งเกี่ยวและป้องกันการปลอมแปลงด้วยระบบป้องกันที่ซับซ้อน
4. สามารถเปลี่ยนมือและนำกลับมาใช้ใหม่ได้
5. ทำหน้าที่ต่าง ๆ ได้มากมาย
6. สามารถนำไปใช้ในงานต่าง ๆ ได้อย่างกว้างขวาง เช่น การขนส่ง ธนาคาร และการรักษาสุขภาพ เป็นต้น
7. สามารถประยุกต์ใช้กับอุปกรณ์อิเล็กทรอนิกส์แบบพกพาต่าง ๆ ได้ เช่น เครื่องโทรศัพท์และเครื่องคอมพิวเตอร์กระเป๋าหิ้ว
8. ทำงานด้วยเทคโนโลยีเซมิคอนดัคเตอร์ที่มีการพัฒนาอย่างรวดเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนประกอบและโครงสร้างของสมาร์ตการ์ด

สมาร์ตการ์ดประกอบด้วยบัตรพลาสติก และหน้าสัมผัสที่บรรจุชิปสมาร์ตการ์ดเรียบร้อยแล้ว ซึ่งส่วนประกอบต่างๆแสดงดังรูป

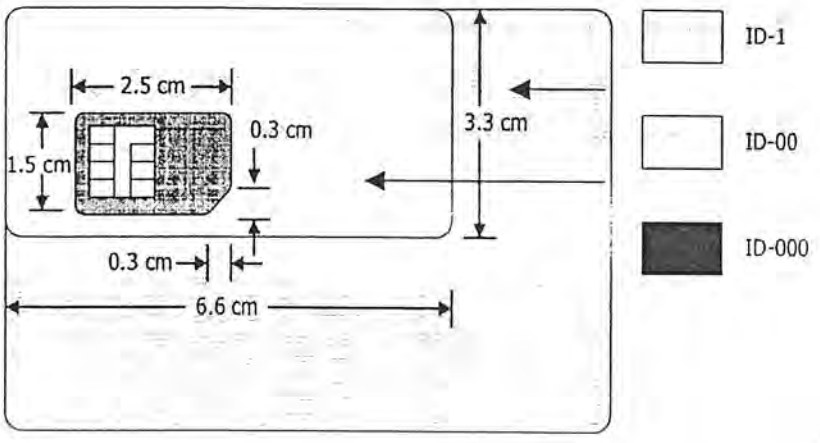


รูปที่ 3.1 ส่วนประกอบของสมาร์ตการ์ด

3.3.1 ตัวบัตรพลาสติก (Plastic Card)

ขนาดของบัตรพลาสติคที่นำมาทำสมาร์ตการ์ดกำหนดโดยมาตรฐานระหว่างประเทศ คือ ISO 7810 โดยมาตรฐานนี้ยังได้กำหนดถึงคุณลักษณะทางกายภาพของพลาสติคที่นำมาใช้ทำบัตรด้วย เช่น ความคลาดเคลื่อนของอุณหภูมิ และความยืดหยุ่นตัวในการใช้งาน ตำแหน่งของหน้าสัมผัสทางไฟฟ้าและการทำงานของมัน ตลอดจนกำหนดว่าการติดต่อระหว่างวงจรรวม หรือ IC กับโลกภายนอกเป็นอย่างไรอีกด้วย มีพลาสติกอยู่4ชนิดที่นำมาใช้ผลิตสมาร์ตการ์ดได้แก่PVC (Polyethylene Terephthalate), ABS (Acrylonitrile Butadiene Styrene), PC (Polycarbonate), และ PET (Polyethylene Terephthalate) แต่ที่นิยมใช้กันมากในประเทศไทยคือ พีวีซี (PVC - Polyvinyl Chloride) และเอบีเอส (ABS - Acrylonitrile Butadiene Styrene) อย่างไรก็ดี การใช้พีวีซีมีข้อดีคือสามารถพิมพ์ลายนูนได้ แต่ไม่สามารถนำกลับมาใช้ใหม่ไม่ย่อยสลายในธรรมชาติได้ ส่วนเอบีเอสไม่สามารถพิมพ์นูนได้แต่นำกลับมาใช้งานใหม่ได้

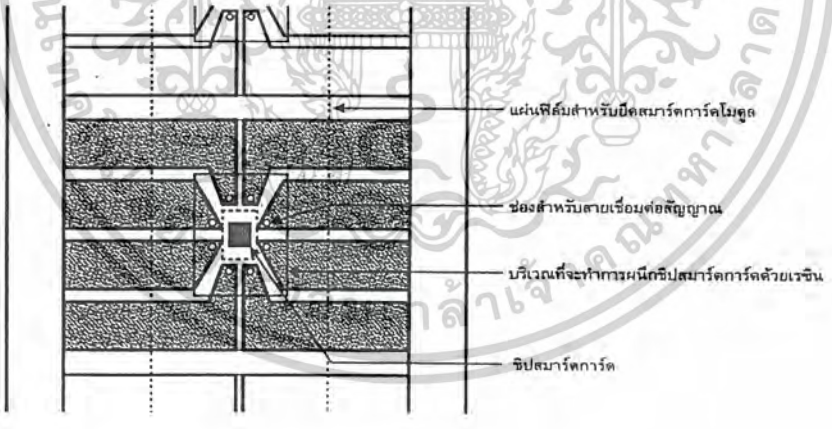
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การแบ่งชนิดของบัตรตามรูปร่างที่นำไปใช้งาน และขนาด

3.3.2 หน้าสัมผัสและชิปสมาร์ตการ์ด (Smart Card Module)

สมาร์ตการ์ดโมดูลหรือหน้าสัมผัสและชิปสมาร์ตการ์ดคือ ส่วนที่แสดงความเป็นตัวตนของสมาร์ตการ์ดที่สุด สมาร์ตการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่อาจทราบได้เลยว่ามันคือสมาร์ตการ์ดที่มีการฝังชิปไว้ในบัตร โดยส่วนที่จะแสดงภาพลักษณ์ที่ชัดเจนของสมาร์ตการ์ดคือสมาร์ตการ์ดโมดูล

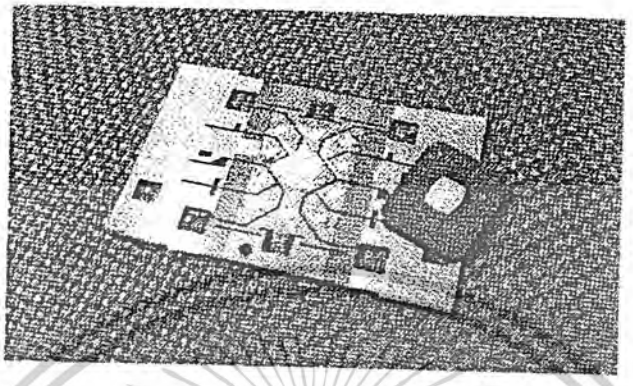


รูปที่ 3.3 ส่วนประกอบของสมาร์ตการ์ดโมดูลในสายการผลิตสมาร์ตการ์ด

ในการผลิตสมาร์ตการ์ดโมดูล ส่วนที่เป็นหน้าสัมผัสของสมาร์ตการ์ดประกอบด้วยโลหะหลายชนิดประกอบกัน แต่ละส่วนจะถูกยึดด้วยฟิล์มบางๆทางด้านหลังของหน้าสัมผัสเพื่อให้คงรูปอยู่ได้ แถบฟิล์มตัวนี้จะมีการเจาะช่องเล็กๆสำหรับการเชื่อมต่อสายนำสัญญาณกับสมาร์ตชิปกับหน้าสัมผัสหลังจากที่วางชิปสมาร์ตการ์ดลงในตำแหน่งที่ต้องการ และทำการเชื่อมต่อสายนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณจากชิปสมาร์ตการ์ดเข้ากับหน้าสัมผัสเรียบร้อยแล้ว ขั้นสุดท้ายจะเป็นการผนึกชิปเพื่อป้องกันตัวชิป และสายนำสัญญาณต่างๆจากสิ่งแวดล้อมภายนอกขั้นสุดท้ายจะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติกและทดสอบการทำงานของชิปขั้นสุดท้าย



รูปที่ 3.4 ตัวอย่างสมาร์ตการ์ดโมดูล

3.4 องค์ประกอบในการใช้งานสมาร์ตการ์ด

3.4.1 ตัวบัตรและตัวชิป

บัตรและชิปสมาร์ตการ์ดเป็นส่วนแรกที่จะกล่าวถึงเพราะสมาร์ตการ์ดมีหลากหลายรูปแบบหลากหลายการใช้งาน โดยหลักการแล้วสมาร์ตการ์ดเป็นเพียงบัตรฝังชิป IC ที่สามารถเก็บข้อมูลได้เท่านั้นผู้ออกแบบระบบมีหน้าที่นำสมาร์ตการ์ดมาใช้งานอย่างชาญฉลาดเหมาะสมตามประเภทงานและบริหารข้อมูลภายในสมาร์ตการ์ดให้เกิดความปลอดภัยสูงสุด

สมาร์ตการ์ดที่นำมาใช้งานมีตั้งแต่ราคาใบละไม่กี่ร้อยบาท ถึงใบละหลายพันบาน โดยในปัจจุบันเราสามารถเห็นการใช้งานสมาร์ตการ์ดในหลายรูปแบบเช่น บัตรโทรศัพท์ ซิมการ์ดในโทรศัพท์มือถือ , บัตรเข้าออกที่อยู่อาศัย (คอนโดมิเนียมบางแห่ง) , บัตรนักศึกษา , บัตรพนักงาน, บัตรเติมน้ำมันแบบเครดิต (Fleet Card), บัตรแทนเงินสด, ซิมการ์ดในโทรศัพท์มือถือซึ่งมีการกำหนดเป็นมาตรฐาน GSM โดยผู้ผลิตสมาร์ตการ์ดต้องผลิตสมาร์ตการ์ดที่มีโครงสร้างที่มีโครงสร้างข้อมูลภายในตามที่มาตรฐาน GSM กำหนด

3.4.2 สมาร์ตการ์ดรีดเดอร์

สมาร์ตการ์ดรีดเดอร์ จะประกอบด้วยขาสำหรับเชื่อมสัญญาณกับหน้าสัมผัสบนชิปสมาร์ตการ์ด (Card Contact) หรือเป็นเสาอากาศรับส่งคลื่นวิทยุสำหรับสมาร์ตการ์ดแบบไม่มีหน้าสัมผัส (Contact less) และหน่วยประมวลผลพร้อมหน่วยความจำสำหรับติดต่อสื่อสารกับชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ตการ์ดโดยตรง การสร้างสมาร์ตการ์ดรีดเดอร์ขึ้นใช้เองสามารถทำได้ โดยการนำไมโครโปรเซสเซอร์หรือไมโครคอนโทลเลอร์ มาประยุกต์ใช้ในการเชื่อมต่อกับสมาร์ตการ์ด

3.5 รูปแบบของสมาร์ตการ์ดที่นำมาใช้ในโครงงาน

ในโครงงานได้นำบัตรโทรศัพท์สาธารณะ TOT Card ที่ใช้งานหมดแล้วมาใช้ เป็นบัตรโทรศัพท์รูปแบบใหม่ที่ใช้เทคโนโลยีใหม่ล่าสุดของระบบโทรศัพท์สาธารณะ ที่ใช้อยู่ในปัจจุบัน โดยบัตรโทรศัพท์สาธารณะ TOT Card โดย TOT Card เป็นรูปแบบหนึ่งของ Memory Card เครื่องโทรศัพท์จะมีอุปกรณ์รักษาความปลอดภัย หรือ SAM (Security Access Module) ติดตั้งในเครื่องเพื่อทำการตรวจสอบบัตร และมูลค่าบัตร จึงเป็นการป้องกันการนำบัตรปลอมมาใช้งาน

3.5.1 บัตรโทรศัพท์สาธารณะ TOT CARD

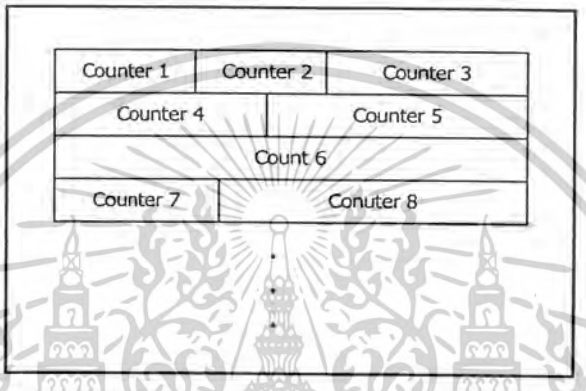
รูปที่ 3.5 ตัวอย่างบัตรโทรศัพท์ TOT

เป็นบัตรพลาสติกที่มีขนาดเท่ากับขนาดมาตรฐาน ของบัตรเครดิต ที่ใช้ในปัจจุบันซึ่งบรรจุ CHIP บันทึกข้อมูล ติดไว้บนหน้าบัตร สำหรับเป็นสื่อสัญญาณ ระหว่างบัตรกับ เครื่องโทรศัพท์ที่มีหัวอ่าน พร้อมระบบรักษาความปลอดภัย ซึ่งใช้ในการตรวจสอบความถูกต้องของข้อมูลในบัตร องค์การโทรศัพท์ฯ ใช้ มาตรฐาน EURO CHIP นำมาให้บริการ และเพื่อป้องกัน การลอกเลียนหรือปลอมแปลงบัตร องค์การโทรศัพท์จึงนำระบบ การสร้างรหัสหลัก Master Key อันเป็น ลิขสิทธิ์ขององค์การโทรศัพท์ เป็นองค์ประกอบหลักในการเข้ารหัสข้อมูล เพื่อผลิตบัตร ซึ่ง สามารถป้องกันการปลอมแปลงบัตรโทรศัพท์ ได้เป็นอย่างดี คุณสมบัติ ทนทานในทุกสภาพ อายุการใช้งาน 3 ปีโดยระบุที่ด้านหลังบัตรทุกใบ ไอซี (CHIP) เป็นแบบ MEMORY CHIP ซึ่งเมื่อใช้จนมูลค่าในบัตรหมดไปแล้ว ไม่สามารถเติม มูลค่าเงินลงในบัตรได้อีก ทั้งนี้ในอนาคตอันใกล้ องค์การโทรศัพท์ จะพัฒนาบริการ ไปสู่บริการกระเป๋าเงินสดอิเลคทรอนิคส์ คือบัตรใบเดียวสามารถจับจ่ายใช้สอย หลายๆ บริการทั่วประเทศ

TOT Card เป็น Token Memory Card ภายในสมาร์ตการ์ดชนิดนี้จะมีการเก็บข้อมูลในลักษณะของการนับจำนวน (Counter) ซึ่งจำนวนนับนี้จะเป็นตัวเลขแทนมูลค่าของเงินที่ระบบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัตร การนับเลขเป็นการนัยถอยหลังเพื่อเป็นการนับมูลค่าคงเหลือในบัตร หมายความว่าหากใช้บัตรในการโทรศัพท์ไปเรื่อยๆ มูลค่าในบัตรก็จะถูกลดลงตามไปด้วยเช่นกัน ในการเข้าถึงข้อมูลของสมาร์ตการ์ดชนิดนี้ต้องมีการแสดงรหัสผ่านให้บัตรรับทราบเหมือนกับ Memory Card แต่ไม่มีBit Protect เท่านั้นการเข้าถึงข้อมูลในหน่วยความจำของสมาร์ตการ์ดชนิดนี้ต้องอ้างอิงกับแอดเดสเสมอ ทั้งนี้ต้องขึ้นอยู่กับข้อกำหนดของสมาร์ตการ์ดแต่ละรุ่น แต่ใน Token Memory Card นี้จะสามารถเข้าถึงและสามารถเปลี่ยนแปลงข้อมูลได้เพียงบางส่วนเท่านั้น โดยส่วนที่สามารถทำการเข้าถึงได้จะเป็นส่วนหมายเลขประจำของแต่ละบัตร (Serial Number) ซึ่งบัตร TOT แต่ละใบจะมีเลขประจำบัตรที่ไม่ซ้ำกัน



รูปที่ 3.6 ตัวอย่างหน่วยความจำของสมาร์ตการ์ดชนิด Memory แบบ Token

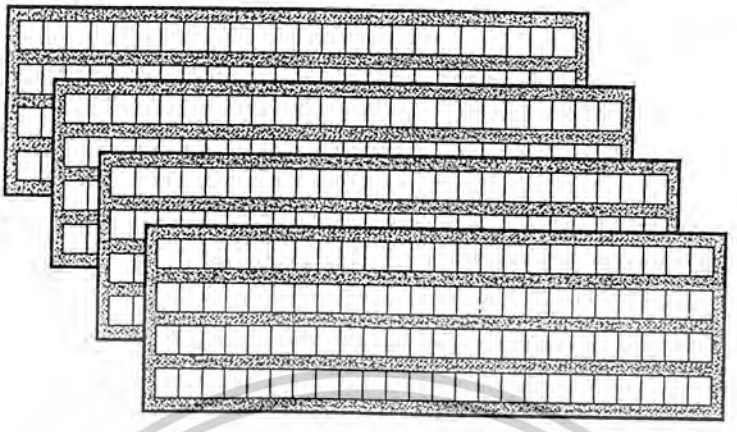
3.5.1.1 การจัดการหน่วยความจำภายใน

ในสมาร์ตการ์ดชนิด Memory มีการแบ่งวิธีการจัดการหน่วยความจำเป็นสองแบบ คือ bitwise และ bytewise การจัดการหน่วยความจำแบบ bitwise เป็น การจัดการหน่วยความจำที่ใช้ในสมาร์ตการ์ดรุ่นแรกๆ การจักการหน่วยความจำแบบนี้มักใช้บอกขนาดของหน่วยความจำของสมาร์ตการ์ดเป็นหน่วยบิตเช่น 1 กิโลบิต (128 ไบต์), 8 กิโลบิต(1 กิโลไบต์) สาเหตุที่ bitwise จัดการหน่วยความจำข้อมูลเป็นบิต เนื่องมาจากข้อมูลที่ใช้รับ-ส่งในสมาร์ตการ์ดชนิดนี้ทำกันในระดับบิตเท่านั้น หมายความว่าการรับส่งข้อมูลไม่จำเป็นต้องทำให้ครบทั้ง 8 บิตหรือ 1 ไบต์ การจัดการหน่วยความจำแบบนี้ สามารถอ่านข้อมูลที่บิตใดก็ได้ ซึ่งมีใช้ในสมาร์ตการ์ดที่มีหน่วยความจำข้อมูลไม่มากนัก

สำหรับการจัดการหน่วยความจำแบบ bytewise เป็นการจัดการหน่วยความจำที่อ้างถึงข้อมูลขนาด 8 บิตหรือ 1 ไบต์เต็ม การรับ-ส่งข้อมูลกับชิปต้องทำการรับ-ส่งข้อมูลทั้ง 8 บิตจนครบจึงจะทำให้การรับ-ส่งข้อมูลเสร็จสมบูรณ์ (หากทำไม่เสร็จสมบูรณ์ ชิปสมาร์ตการ์ดจะยกเลิกการรับส่งข้อมูลครั้งนั้นๆ) นอกจากนี้การอ้างอิงหน่วยความจำยังมีความแตกต่างกันเช่น บางผู้ผลิตกำหนดให้หน่วยความจำเป็นแอดเดสที่ต่อเนื่องกันตั้งแต่แอดเดสที่ 0 ถึงแอดเดสสุดท้ายบางผู้ผลิตแบ่งหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำออกเป็นเพจ (Page) แต่ละเพจมีขนาดแตกต่างกันตามแต่รุ่นที่ผลิต ทำให้การอ้างถึงข้อมูลใดๆในหน่วยความจำของสมาร์ตการ์ดแต่ละแบบไม่เหมือนกันตามแต่ผู้ผลิตจะออกแบบ



รูปที่ 3.7 หน่วยความจำข้อมูลของสมาร์ตการ์ดชนิด Memory

3.5.1.2 มาตรฐานของสมาร์ตการ์ดที่เกี่ยวข้อง

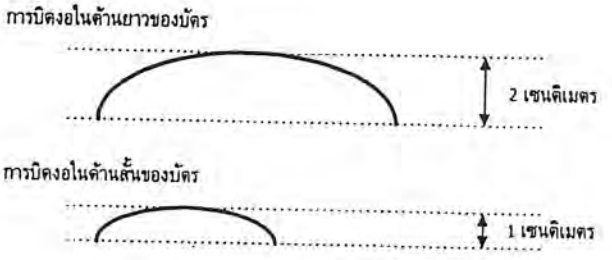
เพื่อให้เกิดความเข้าใจกันได้ของสมาร์ตการ์ด จึงมีการกำหนดมาตรฐานของสมาร์ตการ์ด คือ ISO7816 เป็นข้อกำหนดในเรื่องของคุณสมบัติของบัตรพลาสติกที่จะนำมาใช้ทำเป็นสมาร์ตการ์ด โดยมีหัวข้อย่อยแบ่งเป็น ISO7816-1, ISO7816-2, ISO7816-3, ISO7816-4, ISO7816-5, ISO7816-6 ในที่นี้จะกล่าวในรายละเอียดของ3มาตรฐานแรกเท่านั้นเนื่องจากมีความสำคัญต่อการใช้งานในโครงงาน

1) มาตรฐานISO7816-1

-เป็นมาตรฐานที่กำหนดด้วยเรื่องของคุณสมบัติทางกายภาพเบื้องต้นของสมาร์ตการ์ด ประกอบด้วย

- ความคงทนต่อแสงและรังสีต่างๆ
- ขนาดความหนาของชิปสมาร์ตการ์ด
- ความทนต่อแรงกดของหน้าสัมผัส (ทนต่อแรงกด 1.5 นิวตันได้โดยไม่เสียหาย)
- ค่าความต้านทานของหน้าสัมผัส (ไม่เกิน 0.5 โอห์ม ที่กระแส 0.5 ไมโครแอมป์ – 300 มิลลิแอมป์)
- ความทนต่อสนามแม่เหล็ก
- ความทนต่อไฟฟ้าสถิต (1500 โวลต์ ประจุ 100 พิโกฟารัด ที่ 1500 โอห์ม)
- ความทนทานต่อการบิดงอ เป็นจำนวน 30 ครั้งต่อนาที โดยที่บัตรและชิปต้องไม่เกิดความเสียหาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

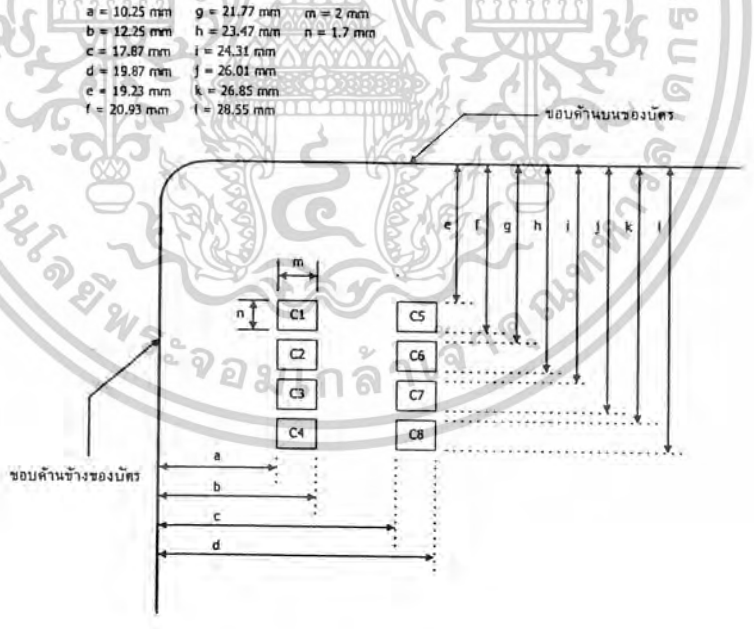


รูปที่ 3.8 วิธีทดสอบการบิดงอสมาร์ตการ์ด

2) มาตรฐาน ISO7816-2

เป็นมาตรฐานที่กำหนดขนาดของหน้าสัมผัส และตำแหน่งของหน้าสัมผัสชิปสมาร์ตการ์ดบนบัตร ประกอบด้วย

- ขนาดของหน้าสัมผัสชิปสมาร์ตการ์ด
- ตำแหน่งของหน้าสัมผัสบนบัตร ดังรูป



รูปที่ 3.9 ตำแหน่งหน้าสัมผัสของชิปสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) มาตรฐาน ISO7816-3

มาตรฐานที่กำหนดคุณสมบัติทางไฟฟ้าและ Protocol ที่ใช้ในการสื่อสารกับชิปสมาร์ตการ์ด จะเป็นการบรรยายเกี่ยวกับหน้าสัมผัสมีดังนี้

Vcc แรงดันไฟบวกของแหล่งจ่ายไฟฟ้าที่ป้อนให้แก่ชิป

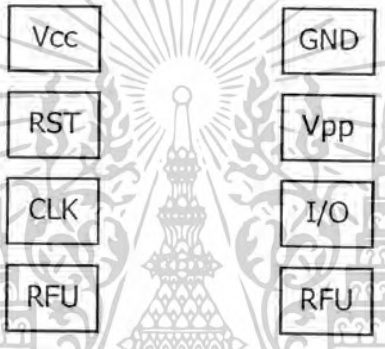
Vpp แรงดันไฟฟ้าสำหรับการเขียนข้อมูลลงในชิปสมาร์ตการ์ด

GND กราวด์ของแหล่งจ่ายกระแสไฟฟ้าที่ป้อนให้แก่ชิป

RST แรงดันไฟฟ้าสำหรับรีเซตชิปสมาร์ตการ์ด

I/O Input – Output สำหรับการรับส่งข้อมูลแบบอนุกรม

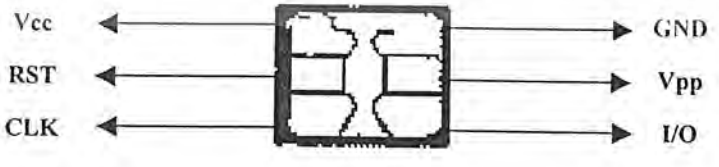
CLK สัญญาณนาฬิกาสำหรับกำหนดจังหวะการรับ-ส่งข้อมูล



รูปที่ 3.10 หน้าที่การทำงานของแต่ละหน้าสัมผัส

3.5.1.3 รูปแบบของสมาร์ตชิปในบัตร TOT

ในบัตร TOT การ์ด จะมีหน้าสัมผัสอยู่ 6 หน้าสัมผัสโดยจะไม่มี ขา RFU ทั้งสองขา



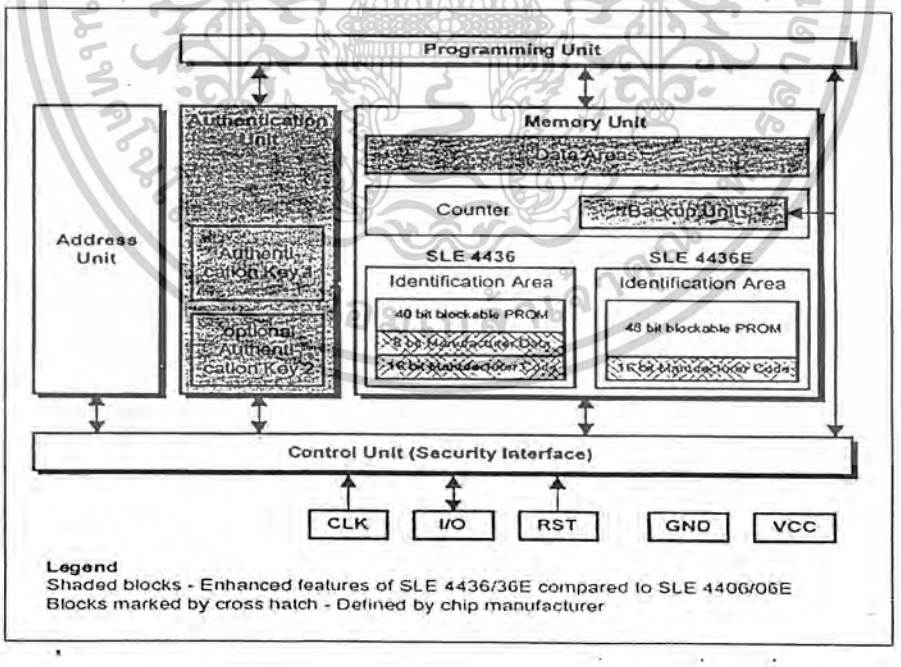
รูปที่ 3.11 สมาร์ตชิปในบัตร TOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัตรโทรศัพท์สาธารณะ TOT Card เป็นสมาร์ตการ์ดแบบ Synchronous Card เบอร์ SLE4436 โดยจะเป็นรูปแบบของเดบิตการ์ดหรือบัตรชนิดพื้นฐานที่นำไปใช้เพื่อเป็นมูลค่าแทนเงินสดในการใช้งานโทรศัพท์สาธารณะภายในบัตรจะมีการบันทึกข้อมูลในรูปของ Unit Counter อยู่ภายในส่วนของหน่วยบันทำข้อมูลในบัตร (Memory Unit) โดยหลังจากการบันทึกข้อมูลลงยังบัตรโดยผู้ผลิตแล้วข้อมูลส่วนหนึ่งเช่น Customer Code, หมายเลขบัตร, วันหมดอายุของบัตร, จะถูกเขียนลงไปอย่างถาวรไม่สามารถแก้ไขได้ ในขณะที่ข้อมูลอีกส่วนหนึ่งคือ Unit Counter จะสามารถลดค่าลงได้เพียงอย่างเดียว ไม่สามารถเพิ่มค่าได้ โครงสร้างของส่วนที่เก็บข้อมูล Unit Counter เป็นฟิวส์ขนาด 40 บิต (Logical Fuse) จากการใช้งานข้อมูลของ Unit Counter จะถูกลดค่าลงจนเป็นศูนย์ แต่ข้อมูลอื่นๆในบัตรยังคงอยู่ในโครงนี้จะใช้ข้อมูลที่ยังคงอยู่นี้ประยุกต์ใช้ในงานระบบรักษาความปลอดภัย

3.5.1.4 โครงสร้างภายในบัตร SLE4436

บัตร SLE4436 บัตรชนิดนี้ถูกออกแบบมาเพื่อการใช้บริการพรีเพด (Pre-Paid) หรือบริการจ่ายเงินก่อนค่อยใช้บริการโครงสร้างภายในชิปไอซีของบัตร SLE4436 โดยทั่วไปจะประกอบด้วยหน่วยความจำแบบ EEPROM ขนาด 221 บิต, หน่วยความจำ ROM ขนาด 16 บิต, ส่วนควบคุมและรักษาความปลอดภัยให้ข้อมูล (Control Security Unit), ส่วนประมวลผลเฉพาะสำหรับการรับรอง (Authentication)



รูปที่ 3.12 ไดอะแกรมแสดงส่วนการทำงานภายในสมาร์ตการ์ดตระกูล SLE4436

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับบัตร SLE4436 ที่ถูกนำมาผ่านกระบวนการ เพื่อใช้เป็นบัตร TOT Card ภายในหน่วยบันทึกข้อมูล (Memory Unit) จะถูกบรรจุไว้เป็นข้อมูลขนาด 48 ไบต์ ข้อมูลจะถูกบันทึกตั้งแต่ในกระบวนการผลิตของโรงงาน ไม่สามารถเข้าไปเปลี่ยนแปลงหรือแก้ไขค่าข้อมูลได้ ที่สามารถแบ่งออกได้เป็น 5 ส่วน อันได้แก่

- ข้อมูลชุดที่ 1 มีขนาด 3 ไบต์ เป็นข้อมูลที่ระบุถึง Factory Code มีไว้สำหรับระบุข้อมูลของบัตรไปสร้างแอพพิเคชันเพื่อให้บริการ เช่น ถ้าต้องการใช้เกี่ยวกับการทำบัตรนี้ทำบริการเครื่องขายสินค้าอัตโนมัติก็จะมีการระบุรหัสประจำตัวของผู้ให้บริการ เพื่อให้สามารถแยกแยะตัวผู้ให้บริการได้

- ข้อมูลชุดที่ 2 มีขนาด 5 ไบต์ เป็นข้อมูลหมายเลขบัตรซึ่งถูกเก็บอยู่ในรูปของรหัส BCD โดยเป็นตัวเลขขนาด 10 หลัก

- ข้อมูลชุดที่ 3 มีขนาด 5 ไบต์ เป็น Balance Unit หรือมูลค่าตัวเงินของบัตร การนับมูลค่าเงินหรือ Balance Counter จะมีวิธีการคำนวณตามการใช้งาน

- ข้อมูลชุดที่ 4 มีขนาด 32 ไบต์ เป็นข้อมูลลับของทางผู้ผลิตสมาร์ตการ์ด

- ข้อมูลชุดที่ 5 มีขนาด 1 ไบต์ เป็นข้อมูลของปีและเดือนที่บัตรจะหมดอายุ

การจัดเก็บข้อมูลภายในหน่วยบันทึกข้อมูลของบัตร SLE4436				
ข้อมูลไบต์ที่ 0-2	ข้อมูลไบต์ที่ 3-7	ข้อมูลไบต์ที่ 8-12	ข้อมูลไบต์ที่ 13-46	ข้อมูลไบต์ที่ 47
Customer Code	Card Number	Balance Counter	ข้อมูลที่เป็นความลับของผู้ผลิต	วันหมดอายุของบัตร

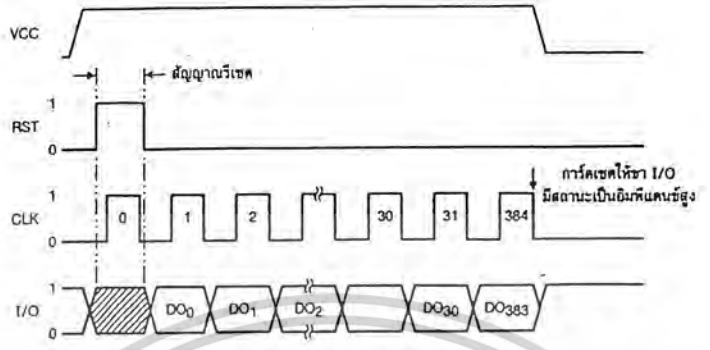
ตารางที่ 3.1 โครงสร้าง และรายละเอียดที่เกี่ยวกับข้อมูลทั้ง 48 ไบต์ ในบัตร SLE4436

3.5.1.5 การติดต่อกับการ์ด SLE4436

จ่ายไฟตรงให้แก่บัตร กำหนดให้สัญญาณที่ขา Clock เป็นลอจิกสูง กำหนดให้สัญญาณที่ขา RST เป็นลอจิกสูงประมาณ 10 uS แล้วให้ลอจิกต่ำที่ขาเป็นเวลาประมาณ 10 uS จากนั้น กำหนดให้สัญญาณที่ขา Clock เป็นลอจิกสูงค้างไว้นานประมาณ 5 uS ให้อ่านข้อมูลจากขา I/O โดยเลื่อนบิตข้อมูลที่อ่านได้ไปทีละ 1 บิต ซึ่งหมายความว่าเมื่อป้อนสัญญาณ Clock ครบ 8 ลูก อ่านข้อมูลจากขา I/O เลื่อนข้อมูลและเก็บค่าจนครบ 8 บิต ข้อมูลที่ได้มาสุดท้ายก็คือข้อมูลขนาด 1 ไบต์ (ข้อมูลบิตแรกที่ได้ เป็น MSB) ใช้รูปแบบของสัญญาณตามที่ได้กล่าวมาจนได้สัญญาณที่ขา Clock ครบ 384

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลูก หรือเทียบได้เป็นข้อมูล 48 ไบต์ (เนื่องจากเป็นการอ่านข้อมูลแบบอนุกรมจึงต้องอ่านข้อมูลออกมาทั้ง 48 ไบต์) หลังจากนั้นจึงนำข้อมูลที่ได้มาแยกแยะว่าส่วนใดคือข้อมูลอะไร



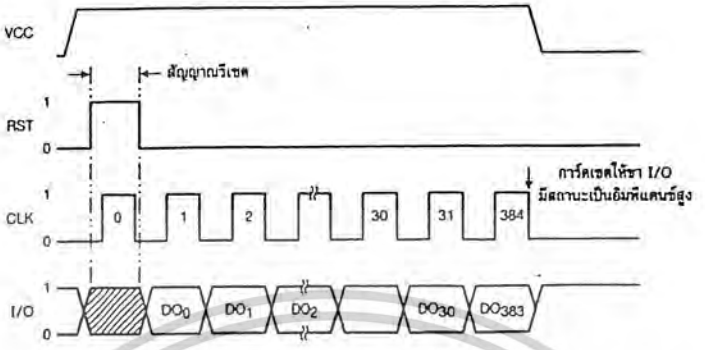
รูปที่ 3.13 แผนผังทางเวลาของสัญญาณที่เกี่ยวข้องสำหรับการอ่านข้อมูลจากบัตร SLE4436

3.5.2 การ์ดที่มีระบบป้องกันข้อมูล SLE4442

การ์ดที่มีระบบป้องกันความปลอดภัยข้อมูลหรือ Security Memory Card คือสมาร์ตการ์ดที่การอ่านข้อมูลสามารถทำได้อย่างอิสระ แต่การเขียนข้อมูลจะไม่สามารถทำได้หากไม่มีรหัสผ่านหรือรหัส PSC ที่ถูกต้อง วิธีการในลักษณะนี้ช่วยให้ข้อมูลภายในสมาร์ตการ์ดได้รับการปกป้องและมีความน่าเชื่อถือ จุดนี้เองเป็นส่วนที่ทำให้ Security Memory Card แตกต่างไปจาก Free Access Memory Card อย่างชัดเจน รูปแบบการสื่อสารข้อมูลของการ์ดชนิดนี้เป็นการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) ตามมาตรฐาน ISO7816 ซึ่งรูปแบบคำสั่งที่ใช้ในการติดต่อและควบคุมการ์ดจะแตกต่างกันไปในผู้ผลิตการ์ดแต่ละราย (ข้อมูลส่วนนี้สามารถดูได้จากเอกสารของผู้ผลิตการ์ด) ซึ่งภายในหัวข้อนี้จะอ้างอิงจากสมาร์ตการ์ด SLE4442 ของบริษัทSiemens เนื่องจากเป็นการ์ดที่มีคุณสมบัติในการรักษาความปลอดภัยข้อมูลอย่างครบถ้วนและสามารถหามาใช้งานได้ง่ายในบ้านเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลูก หรือเทียบได้เป็นข้อมูล 48 ไบต์ (เนื่องจากเป็นการอ่านข้อมูลแบบอนุกรมจึงต้องอ่านข้อมูลออกมาทั้ง 48 ไบต์) หลังจากนั้นจึงนำข้อมูลที่ได้มาแยกแยะว่าส่วนใดคือข้อมูลอะไร



รูปที่ 3.13 แผนผังทางเวลาของสัญญาณที่เกี่ยวข้องสำหรับการอ่านข้อมูลจากบัตร SLE4436

3.5.2 การ์ดที่มีระบบป้องกันข้อมูล SLE4442

การ์ดที่มีระบบป้องกันความปลอดภัยข้อมูลหรือ Security Memory Card คือสมาร์ตการ์ดที่การอ่านข้อมูลสามารถทำได้อย่างอิสระ แต่การเขียนข้อมูลจะไม่สามารถทำได้หากไม่มีรหัสผ่านหรือรหัส PSC ที่ถูกต้อง วิธีการในลักษณะนี้ช่วยให้ข้อมูลภายในสมาร์ตการ์ดได้รับการปกป้องและมีความน่าเชื่อถือ จุดนี้เองเป็นส่วนที่ทำให้ Security Memory Card แตกต่างไปจาก Free Access Memory Card อย่างชัดเจน รูปแบบการสื่อสารข้อมูลของการ์ดชนิดนี้เป็นการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) ตามมาตรฐาน ISO7816 ซึ่งรูปแบบคำสั่งที่ใช้ในการติดต่อและควบคุมการ์ดจะแตกต่างกันไปในผู้ผลิตการ์ดแต่ละราย (ข้อมูลส่วนนี้สามารถดูได้จากเอกสารของผู้ผลิตการ์ด) ซึ่งภายในหัวข้อนี้จะอ้างอิงจากสมาร์ตการ์ด SLE4442 ของบริษัทSiemens เนื่องจากเป็นการ์ดที่มีคุณสมบัติในการรักษาความปลอดภัยข้อมูลอย่างครบถ้วนและสามารถหามาใช้งานได้ง่ายในบ้านเรา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติโดยทั่วไปของ SLE4442

- ใช้หน่วยความจำ EEPROM 8บิต ความจุข้อมูล 256 ไบต์
- ใช้รูปแบบของ ATR (Answer To Reset) ตามมาตรฐาน ISO7816-3
- อินเตอร์เฟสแบบซิงโครนัส(Synchronous) ตามมาตรฐาน ISO7816
- ป้องกันการเขียนข้อมูลด้วยรหัสผ่าน PSC (Programmable Security Code)
- การลบและเขียนข้อมูลในแต่ละไบต์ใช้เวลาเพียง 2.5 มิลลิวินาที
- มีฟังค์ชันป้องกันข้อมูลในพื้นที่หน่วยความจำ 32 ไบต์แรก โดยสามารถจะกำหนดให้ข้อมูลที่เขียนลงไปยังพื้นที่ช่วงดังกล่าวถูกเขียนลงไปอย่างถาวรได้

จะเห็นได้ว่าหน่วยความจำขนาด 256 ไบต์ ที่อยู่ภายใน SLE4442 จะถูกแบ่งออกเป็น 2 ส่วนด้วยกัน ได้แก่ข้อมูลในช่วง 32 ไบต์แรกซึ่งเป็นพื้นที่ที่มีระบบป้องกันการเขียนข้อมูลทับ และหน่วยความจำส่วนถัดมาซึ่งเป็นอีอีพรอม (EEPROM) ที่สามารถทั้งเขียนและอ่านได้ กลไกในการป้องกันข้อมูลของ SLE4442 มาจากส่วนที่เป็น Security Memory (ในรูปที่ 2.22) ที่ได้รับการปกป้องโดยข้อมูลสำคัญ 2 ส่วน คือ

- Reference Data(PSC) เป็นข้อมูลขนาด 3 ไบต์ที่เก็บค่าของรหัสผ่านสำหรับการเข้าไปแก้ไขข้อมูลในหน่วยความจำเอาไว้ (รหัส PSC ไม่สามารถถูกอ่านออกมาได้) รหัส PSC จะถูกกำหนดเป็นค่าหนึ่งมาโดยผู้ผลิตก่อนซึ่งสามารถจะมาปรับเปลี่ยนเองได้ภายหลังเมื่อใช้งาน
- Error Counter Byte เป็นข้อมูลที่บอกถึงจำนวนครั้งที่ป้อนรหัส PSC ผิดซึ่งจะถูกกำหนดเอาไว้ตายตัวว่าจะผิดได้ไม่เกิน 3 ครั้ง หากเกินกว่านั้นการ์ดจะล๊อกตัวเองอย่างถาวรทันทีและไม่มีทางปลดล็อกได้ แม้จะป้อนรหัส PSC ที่ถูกต้องไปแล้วก็ตาม การเขียนข้อมูลยังหน่วยความจำก็จะไม่สามารถทำได้อีกต่อไป แต่ยังคงอ่านข้อมูลออกมาได้ตามปกติ การป้อนรหัส PSC ผิดแต่ละครั้ง Error Counter จะถูกลดลงไป 1 ค่าทันที ถ้าหากค่า Error Counter ถูกลดจนมีค่าเป็น 0 เมื่อไรก็แสดงว่าการ์ดได้ถูกล๊อกไปเรียบร้อยแล้ว (ในกรณีที่ป้อนรหัส PSC ผิดมาแล้ว 2 ครั้ง แต่ป้อนรหัสถูกในครั้งที่ 3 ค่าของError Counter จะถูกรีเซตกลับไปเป็น 3 ครั้งเหมือนอย่างตอนแรกเริ่ม)

จะเห็นได้ว่าการอ่านข้อมูลจากหน่วยความจำนั้น เราสามารถจะอ่านข้อมูลออกมาได้โดยไม่ต้องผ่านขั้นตอนการป้อนรหัส PSC แต่สำหรับการเขียนข้อมูลแล้ว เราจะต้องป้อนรหัส PSC ที่ถูกต้องเสียก่อน เพื่อเปิดลอจิกในการเขียนข้อมูลลงยังหน่วยความจำ นอกจากนั้นก็จะเห็นได้ว่าข้อมูล 4 ไบต์แรก เป็นข้อมูลของผู้ผลิต หรือManufacturer Code มีขนาด 4 ไบต์ พื้นที่ส่วนนี้ใช้เก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของATR โดยความหมายของข้อมูลที่อยู่ในพื้นที่ส่วนนี้ แต่ละไบต์จะถูกกำหนดโดยบริษัทผู้ผลิตการ์ดแต่ละราย

รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ด SLE4442

รูปแบบการสื่อสารข้อมูลของสมาร์ดเบอร์ SLE4442 เป็นการรับส่งข้อมูลระหว่างเครื่องอ่านและสมาร์ตการ์ดแบบ 2 ทิศทาง (ข้อมูลบนสาย I/O จะถูกอ่านค่าที่ขอบขาลงของสัญญาณนาฬิกา) โดยรูปแบบการสื่อสารที่ว่านี้ประกอบด้วย 4 โหมดการทำงาน ได้แก่

- การรีเซตและการตอบกลับด้วย ATR (Answer To Reset)
- โหมดการส่งคำสั่ง (Command Mode)
- โหมดการอ่านข้อมูล (Out-going Data Mode)
- โหมดการดำเนินการ (Processing Mode)

การรีเซตและการตอบกลับ

การอินเตอร์เฟสเข้ากับ Security Memory Card ทั่วๆไป รวมทั้ง SLE4442 จะสอดคล้องกับมาตรฐานในการอินเตอร์เฟสแบบซิงโครนัสภายในมาตรฐาน ISO7816 โดยเมื่อรีเซตการทำงานของการ์ดจะทำให้การ์ดมีการตอบกลับด้วยข้อมูล ATR สำหรับข้อมูลที่ตอบกลับมาจาก SLE4442 จะประกอบด้วยข้อมูล 4 ไบต์

การทำงานของโหมดการส่งคำสั่ง

การส่งคำสั่งไปยังสมาร์ตการ์ดหรือการทำงานในโหมดการส่งคำสั่ง ก็คือกระบวนการต่อเนื่องหลังจากการ์ดฟฤกรีเซตไปเรียบร้อยแล้ว โดยการ์ดจะรอรับคำสั่งที่ส่งมาจากเครื่องอ่านซึ่งมีรูปแบบเป็นข้อมูลความยาว 3 ไบต์ โครงสร้างของข้อมูลดังกล่าวประกอบด้วยคำสั่ง(Command), แอดเดรส(Address) และข้อมูล(Data) โดยคำสั่งทั้งหมดที่การ์ดSLE4442 รองรับถูกแสดงอยู่ในรูปที่ 2.26 ส่วนรูปสัญญาณที่จะเกิดขึ้นระหว่างการทำงานของโหมดการส่งคำสั่งก็เป็นดังรูป2.27 จะเห็นได้ว่าการส่งข้อมูลแต่ละครั้ง จะต้องมีการส่งสภาวะเริ่มต้นและสภาวะสิ้นสุดกำกับไปกับตัวข้อมูลด้วย ในที่นี้สภาวะเริ่มต้นก็คือการเปลี่ยนระดับจากลอจิกค่าสูงเป็นต่ำที่ขาI/O ในขณะที่ระดับลอจิกที่ขา CLK เป็นค่าสูง , ส่วนภาวะสิ้นสุดก็คือการเปลี่ยนระดับจากลอจิกค่าต่ำเป็นสูงที่ขา I/O ในขณะที่ระดับลอจิกที่ขา I/O ในขณะที่ระดับลอจิกที่ขา CLK เป็นค่าสูง ต่อไปคือความหมายและวิธีการทำงานของแต่ละคำสั่ง

- Read Main Memory คือคำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมาจากหน่วยความจำของการ์ด ทั้งจากพื้นที่ส่วนที่ได้รับการป้องกัน (หน่วยความจำ 32 ไบต์แรก) และส่วนที่ไม่ได้รับการป้องกัน (หน่วยความจำ 224 ไบต์หลัง) โดยจะเป็นการอ่านค่าโดยเริ่มต้นจากแอดเดรสที่ส่งไปจนถึงแอดเดรสสุดท้าย (0ffh) ของพื้นที่หน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Read Protection Memory คือคำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมา จากหน่วยความจำ 32 ไบต์แรก
- Update Main Memory คือคำสั่งที่ใช้ในการเขียนข้อมูลยังแอดเดรสใดๆ ของหน่วยความจำทั้ง 256 ไบต์ ในกรณีที่ใช้คำสั่งนี้ในการเขียนข้อมูลลงยังหน่วยความจำ 32 ไบต์แรก ข้อมูลจะยังคงแก้ไขเปลี่ยนแปลงได้ภายหลัง สำหรับการเขียนข้อมูลจะประกอบด้วย 3 เงื่อนไข คือ
    - การลบข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนดให้เป็น 0FFH แล้วทำการเขียนข้อมูลซ้ำลงยังแอดเดรสเดิม กระบวนการนี้จะต้องใช้สัญญาณนาฬิกา 255 ลูก
    - การเขียนข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนดโดยไม่ต้องลบข้อมูลออก สำหรับแอดเดรสดังกล่าวจะต้องเป็นที่ว่าง (มีข้อมูลเป็น 0FFH) อยู่ก่อนหน้านี้แล้วเท่านั้น กระบวนการนี้จะใช้สัญญาณนาฬิกา 124 ลูก
    - การลบข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนด (มีค่าข้อมูลเป็น0FFH) โดยไม่มีการเขียนข้อมูลต่อ สำหรับกระบวนการนี้จะใช้สัญญาณนาฬิกา 124 ลูกเช่นกัน
- Write Protection Memory คือการเขียนข้อมูลลงยังแอดเดรสของหน่วยความจำใดๆใน 32 ไบต์แรก คำสั่งนี้มีเงื่อนไขว่าข้อมูลที่เขียนลงไปจะถูกเขียนลงยังแอดเดรสของหน่วยความจำที่กำหนดอย่างถาวร ไม่สามารถแก้ไขเปลี่ยนแปลงอะไรได้อีก สำหรับรูปสัญญาณของกระบวนการนี้อ้างอิงได้จากรูปสัญญาณของคำสั่ง Update Main Memory
- Read Security Memory คือการอ่านค่าของ Error Counter เพื่อตรวจดูว่าการ์ดใบนั้นๆ ได้ถูกล็อคไปแล้วหรือยัง โดยค่าภายในบิต D2,D1 และ D0 ของError Counter จะเป็นส่วนที่บอกถึงสถานะของการ์ดในขณะนั้น หากค่าของบิต D2,D1 และ D0 เป็น 0 ทั้งหมดก็แสดงว่าการ์ดได้ถูกล็อคไปแล้ว ซึ่งจะไม่สามารถแก้ไขอะไรได้และจะไม่สามารถเขียนข้อมูลลงยังการ์ดนั้นได้อีกต่อไป (แต่ว่าการอ่านข้อมูลการ์ดจะยังคงทำได้อย่างตามปกติ)
- Update Security Memory คือการเข้าไปแก้ไขข้อมูลของรหัส PSC ภายในการ์ด หรือพูดง่ายๆ ก็คือการเข้าไปเปลี่ยนรหัสป้องกันของการ์ดนั่นเอง คำสั่งจะถูกกระทำต่อเมื่อมีการส่งรหัส PSC ที่ถูกต้องไปยังการ์ดเสียก่อน โดยในกรณีที่ป้อนรหัสผิด ค่าของบิต D2,D1 และ D0 ใน Error Counter จะค่อยๆถูกเปลี่ยนจากค่า "1" เป็น "0" ไล่ไปทีละบิตตามจำนวนครั้งที่ป้อนรหัสผิด หากทั้งหมดกลายเป็นศูนย์เมื่อไรการ์ดก็จะถูกล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทันทีซึ่งนั่นหมายความว่าโอกาสป้อนรหัสผิดจะมีเพียง 3 ครั้งเท่านั้น สำหรับรูปสัญญาณของกระบวนการนี้จะเหมือนกับรูปสัญญาณของคำสั่ง Update Main Memory

- Compare Verification Data คือการสั่งให้การ์ดทำการเปรียบเทียบรหัส PSC กับรหัสผ่านที่เราได้ส่งไปยังการ์ด ในการเปรียบเทียบที่ว่านี้ ข้อมูลที่การ์ดจะส่งกลับมาคือค่าของ Error Counter ที่จะบอกว่ารหัสที่เราป้อนไปนั้นถูกต้องหรือไม่ และยังเหลือโอกาสพลาดอีกกี่ครั้งเท่านั้น (เราไม่สามารถจะเข้าไปอ่านค่ารหัส PSC ของการ์ดออกมาได้)

โหมดการอ่านข้อมูล(Outgoing Data Mode)

โหมดการทำงานนี้จะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขออ่านข้อมูล (เช่น Read Main Memory, Read Protection Memory และ Read Security Memory) ไปยังสมาร์ตการ์ดเพื่อขออ่านข้อมูลจากพื้นที่ใดๆในหน่วยความจำหลังจากได้รับคำสั่งดังกล่าว สมาร์ตการ์ดจะส่งข้อมูลที่ถูกร้องขอกลับมายังเครื่องอ่านซึ่งก็เท่ากับว่าเครื่องอ่านจะสามารถอ่านข้อมูลที่ต้องการออกมาได้สำเร็จจากโหมดการทำงานนี้

โหมดดำเนินการ(Processing Mode)

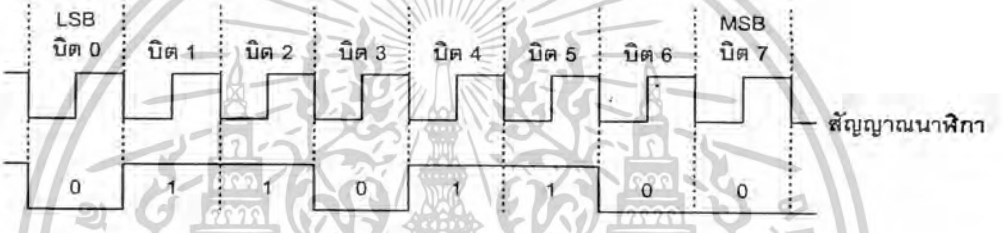
โหมดการดำเนินการจะเกิดขึ้นหลังจากที่มีการส่งคำสั่ง ในกลุ่มของการขอเขียนหรือลบข้อมูลออกจากพื้นที่ใดๆ ในหน่วยความจำ (เช่น Update Main Memory, Write Protection Memory, Update Security Memory และ Compare Verification Data) โดยหลังจากได้รับคำสั่งดังกล่าว สมาร์ตการ์ดจะเริ่มดำเนินกระบวนการตามที่ได้รับการสั่งมาในระหว่างโหมดการทำงานนี้จะสังเกตว่าข้อมูลจากขา I/O จะไม่ถูกนำมาใช้ร่วมในการทำงานเลย (เนื่องจากมีสถานะเป็นลอจิกต่ำตลอดทั้งช่วง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การสื่อสารแบบอนุกรม

4.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้ต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา , ข้อมูล และ กราวด์ รูปที่ 3.1 แสดงให้เห็นถึงไทม์มิ่งไดอะแกรมของการส่งข้อมูล แบบ ซิงโครนัส



รูปที่ 4.1 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

4.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัส คือ การรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนแบบการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (Baud rate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วนด้วยกัน คือ

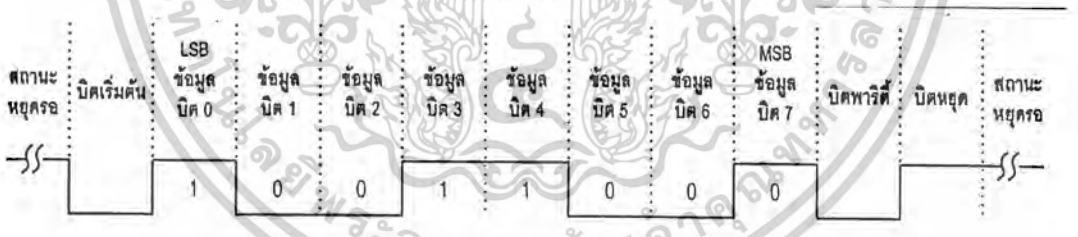
1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5 , 6 , 7 หรือ 8 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit) จะมีขนาด 1 บิต หรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1 , 1.5 หรือ 2 บิต

รูปที่ 4.2 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีข้อมูลที่จะส่ง ขา Data จะมีสถานะลอจิก "1" ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มูลจะเริ่มจากการให้ขา Data มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่า บิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจจะมีจำนวนบิต 5 , 6 , 7 หรือ 8 บิตก็ได้ จากนั้นตามด้วย บิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่ส่งคือ บิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต , 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า Universal Asynchronous Receiver/Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัส คือ บอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110 , 150 , 300 , 600 , 1200 , 2400 , 4800 , 9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรตคือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตเปิดปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็สามารถรับส่งข้อมูลได้ด้วยอัตราเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งจะเหลือ 872 ไบต์แต่วินาที



รูป 4.2 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

การตรวจสอบพาริตี้สามารถกำหนดให้เป็นแบบคี่(odd) , แบบคู่ (even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิกสูง ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัว ซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าในบิตพาริตี้ จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตี้เป็นคี่ ค่าที่บิตพาริตี้จะต้องเป็น "1" เพื่อให้ข้อมูล 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบต์รวมทั้งบิตพาริตี้มีจำนวนบิตที่ลอจิก "1" รวมกันเป็นเลขคี่ ในตาราง 3.1 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART โดยภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่า จะตรวจสอบพาริตี้คี่หรือคู่ จากนั้นภาครับของ UART จะตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือ เป็นคี่ โดยการนับจำนวนลอจิก "1" ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่ แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแจ้งข้อผิดพลาดให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดของการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อ มีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	1	0
11111111	0	1

ตารางที่ 4.1 แสดงบิตพาริตี้ของข้อมูล

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UART เบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่นXT ใช้ UART เบอร์ 8250 UART ชิปเหล่านี้มีระดับแรงดันเป็นแบบทีทีแอล (0 และ +5 V) แต่เพื่อให้แรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ที่ระยะทางไกลมากขึ้น ระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้นโดยลอจิก "0" มีระดับแรงดัน +3V ถึง +12 V ในขณะที่ลอจิก "1" มีระดับแรงดัน -3 V ถึง -12 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่าสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12 V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12 V แสดงว่าเป็นช่องว่าง(Space)

มาตรฐานRS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง(Data Circuit Terminating: DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

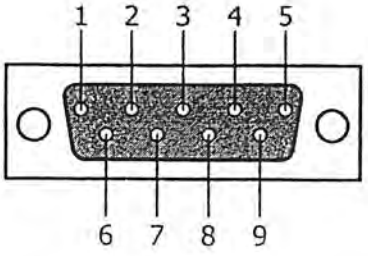
ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็คเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็คเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็คเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุด ถึง 20 เมตร

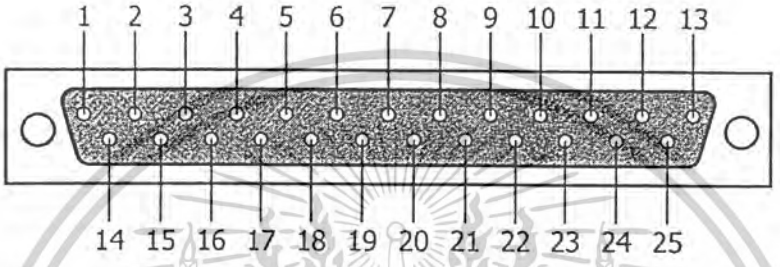
4.3.1 คอนเน็คเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็คเตอร์แบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็คเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 3.3 และตาราง 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 คอนเนคเตอร์อนุกรม 9 ขาหรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



รูปที่ 4.4 คอนเนคเตอร์อนุกรม 25 ขาหรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

ตารางที่ 4.2 หน้าที่การทำงานของขาอนุกรม

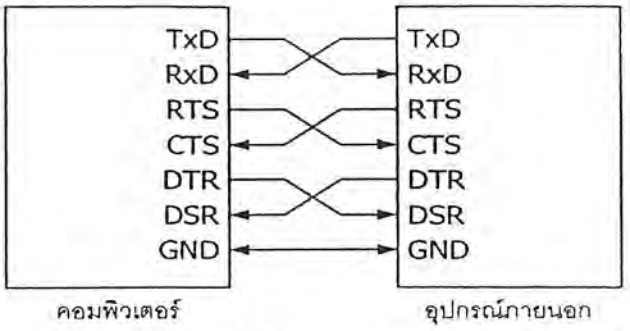
สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในรูปที่ 3.4 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 3.4 ก เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็คเต็มรูปแบบ ส่วนในรูปที่ 3.4 ข เป็นการเชื่อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

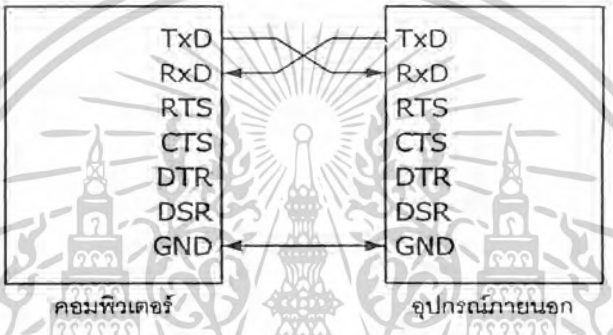
- Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect :CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก
- Receive Data :RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์
- Transmitted Data :TD หรือ TxD ใช้ส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป
- Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกมาจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์
- Signal Ground : GND กราวด์ระบบ
- Data Set Ready :DSR ขานี้จะใช้คู่กับขาDTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขาDSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขาDTR
- Request to Send :RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณRTS ก็คือขาCTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา
- Clear To Send :CTS ขานี้จะคอยรับสัญญาณจากขาRTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่
- Ring Indicator :RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์
แบบ Null modem

รูปที่ 4.5 ก



การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์
แบบ RS-232 โดยใช้สายสัญญาณเพียง 3 เส้น

รูปที่ 4.5 ข

รูปที่ 4.5 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

4.3.2 UART (Universal Asynchronous Receiver Transmitter)

เป็นอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของUART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต) , รูปแบบการส่งข้อมูล , ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้ , เฟรมข้อมูล , โอเวอร์รัน) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (Programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกา UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1 - 65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟลูดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

4.3.3 ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมีUART ที่ใช้งานกันอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็นUART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตัวแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่UART เบอร์นี้ก็ถือว่าเป็นต้นแบบของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆรุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของUART เบอร์นี้

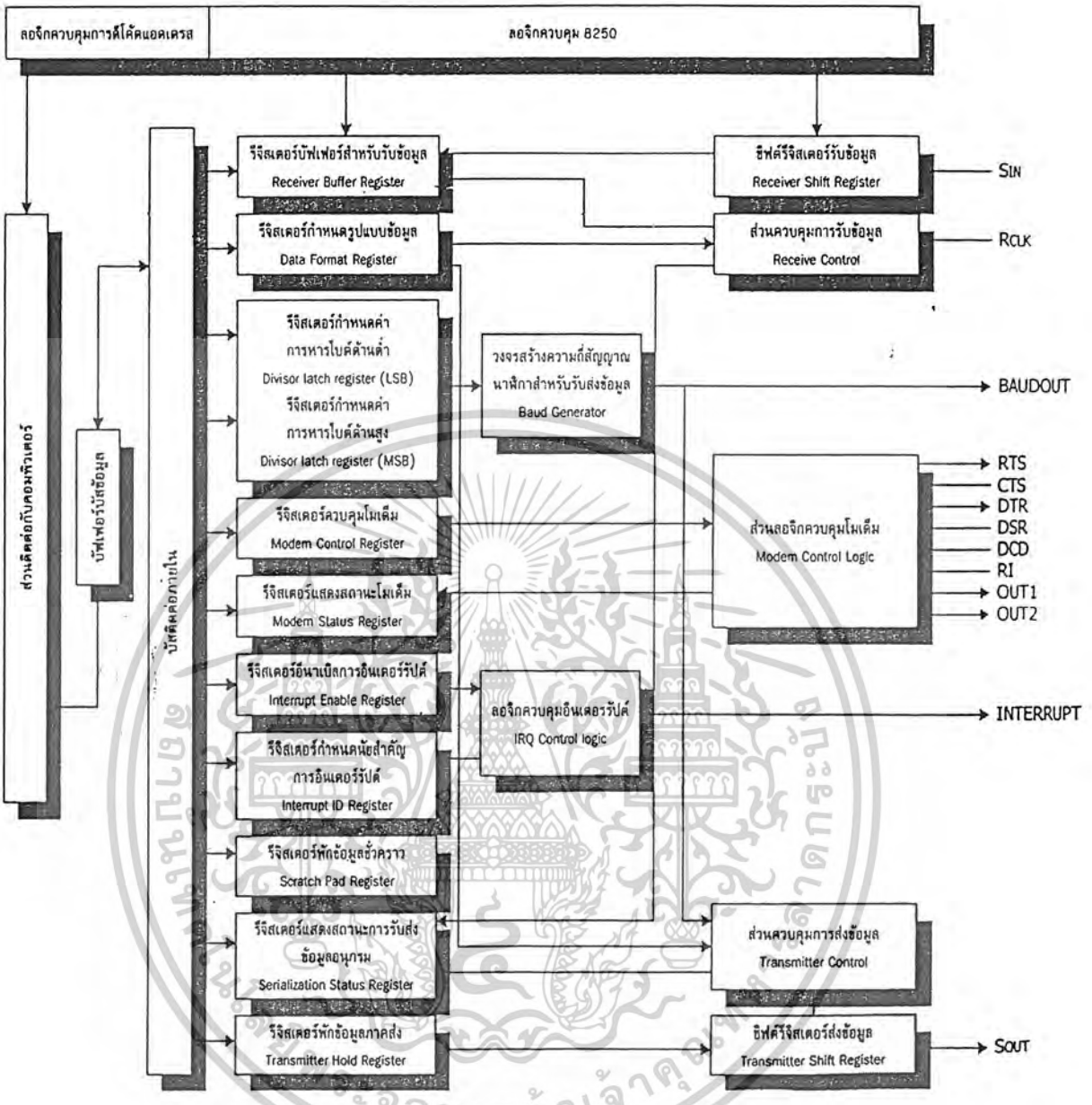
UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First IN First Out) ขนาด 16 ไบต์เข้าไปทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่า เช่น เบอร์ TL16C750 ซึ่งมีรีจิสเตอร์แบบFIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาในการแปลงข้อมุลเพียง 1.8432 MHZ เท่านั้น

4.4 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรมสูงสุดได้ 4 พอร์ต มีชื่อเรียกเป็น COM1,COM2,COM3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน ในรูปที่ 3.5 แสดงผังการทำงานภายในของพอร์ตอนุกรมประกอบด้วยรีจิสเตอร์ 8 บิต 8 ตัว ที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายใน พอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของ พอร์ตอนุกรมยกตัวอย่างพอร์ตอนุกรม COM1 มีแอดเดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆ จะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมมีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 ไดอะแกรมการทำงานภายในของพอร์ตอนุกรมของเครื่องคอมพิวเตอร์

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้าหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์เอ็นเนเบิลการอินเตอร์รัปต์ ใช้เซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดของการอินเตอร์รัปต์03Hรีจิสเตอร์กำหนดรูปแบบขอข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

04H รีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจิสเตอร์แสดงสถานการณ์รับและการส่งข้อมูลแบบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD ,RI,DSR และ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

4.4.1 รีจิสเตอร์ตำแหน่ง 00H (รีจิสเตอร์บัฟเฟอร์)

เป็นรีจิสเตอร์เก็บข้อมูลที่รับเข้ามาและส่งออก โดยการติดต่อกับรีจิสเตอร์นี้เพื่อเก็บข้อมูล จะต้องกำหนดให้บิตDLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล(03H) มีสถานะเป็น "0" ซึ่งการเขียนข้อมูลมายังแอดเดรสนี้เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูล และข้อมูลจะถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อรับเข้ามาแล้ว จะส่งต่อไปยังรีจิสเตอร์เก็บข้อมูล หลังจากอ่านค่าจากรีจิสเตอร์นี้ออกไป รีจิสเตอร์นี้จะถูกเคลียร์ และเตรียมพร้อมสำหรับรับข้อมูลในไบต์ต่อไป

4.4.2 รีจิสเตอร์ตำแหน่ง 01H (รีจิสเตอร์เอ็นเนเบิลการอินเตอร์รัปต์)

เป็นรีจิสเตอร์สำหรับการเอ็นเนเบิลการอินเตอร์รัปต์ ซึ่งเป็นการกำหนดให้UART สร้างสัญญาณอินเตอร์รัปต์ขึ้นมา ฟังก์ชันการทำงานในแต่ละบิตของรีจิสเตอร์นี้มีดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	SINP	ERBK	TBE	RxRD

ตารางที่ 4.3 ฟังก์ชั่นการทำงานของรีจิสเตอร์ตำแหน่ง 01H

บิต 4-7 บิตเหล่านี้ไม่ถูกใช้งาน กำหนดให้เท่ากับ"0"

SINP เอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากเกิดจากเปลี่ยนสถานะที่ขาอินพุต CTS,DSR,DCD หรือขา RI

"1" เอ็นเอเบิลการอินเตอร์รัปต์

"0" ดิสเอเบิล

ERBK เอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากเกิดความผิดพลาดขึ้นด้วยสาเหตุจาก พาริตี้ , โอเวอร์รัน , เฟรมข้อมูล หรือการเบรกข้อมูล

"1" เอ็นเอเบิลการอินเตอร์รัปต์

"0" ดิสเอเบิล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TBE เอ็นเอเบิลการอินเตอร์รัปต์เมื่อรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง
"1" เอ็นเอเบิลการอินเตอร์รัปต์
"0" ดิสเอเบิล

RxRD เอ็นเอเบิลการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ได้รับข้อมูลแล้ว
"1" เอ็นเอเบิลการอินเตอร์รัปต์
"0" ดิสเอเบิล

4.4.3 รีจิสเตอร์ตำแหน่ง 02H (รีจิสเตอร์แสดงโหมดและสถานะการอินเตอร์รัปต์) มีรายละเอียดของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	ID1	ID0	PND

ตารางที่ 4.4 ฟังก์ชันการทำงานของรีจีสเตอร์ 02H

บิต 3-7 ไม่ได้ใช้งาน อ่านค่าได้เท่ากับ "0"

ID1,ID0 ใช้งานร่วมกันเพื่อแจ้งสาเหตุของการเกิดอินเตอร์รัปต์
"00" เกิดการอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตขึ้น การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 4
"01" เกิดการอินเตอร์รัปต์เนื่องจากรีจิสเตอร์บัฟเฟอร์ส่งข้อมูลว่างขึ้น การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 3
"10" เกิดการอินเตอร์รัปต์เนื่องจากข้อมูลถูกเก็บลงในรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเรียบร้อยแล้ว การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 2
"11" เกิดการอินเตอร์รัปต์เนื่องจากความผิดพลาดในการถ่ายทอดข้อมูลหรือเกิดการเบรก (Break :เกิดการหยุดถ่ายทอดข้อมูลกระทันหัน) การอินเตอร์รัปต์แบบนี้มีนัยสำคัญเป็นอันดับ 1 หรือมีนัยสำคัญสูงสุด

PND ใช้แสดงสถานะของการเกิดอินเตอร์รัปต์
"1" แสดงว่าไม่มีการอินเตอร์รัปต์
"0" แสดงว่ามีการอินเตอร์รัปต์เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการสร้างสัญญาณอินเตอร์รัปต์ขึ้น จะต้องมีการเคลียร์ค่าก่อนที่จะให้เกิดอินเตอร์รัปต์ครั้งต่อไป โดยสามารถทำได้ดังนี้ คือ

- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเปลี่ยนแปลงของขาอินพุตจะต้องอ่านค่าจากรีจิสเตอร์แสดงสถานะของโมเด็ม (รีจิสเตอร์ตำแหน่ง 06H) เพื่อเคลียร์ค่าการอินเตอร์รัปต์
- ถ้าเกิดการอินเตอร์รัปต์ เนื่องจากบัฟเฟอร์ส่งข้อมูลว่าง จะต้องเขียนข้อมูลไปยังรีจิสเตอร์บัฟเฟอร์ข้อมูล (รีจิสเตอร์ตำแหน่ง 00H) หรืออ่านค่ารีจิสเตอร์แสดงสถานะอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H)เพื่อเคลียร์ค่าการอินเตอร์รัปต์
- ถ้าเกิดอินเตอร์รัปต์เนื่องจากการเก็บข้อมูลลงในรีจิสเตอร์บัฟเฟอร์สำหรับข้อมูลเรียบร้อย จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านข้อมูลจากรีจิสเตอร์บัฟเฟอร์
- ถ้าเกิดอินเตอร์รัปต์เนื่องจากความผิดพลาดในการรับส่งข้อมูลหรือเกิดการเบรก จะต้องเคลียร์ค่าอินเตอร์รัปต์โดยการอ่านค่ารีจิสเตอร์แสดงสถานการณ์รับและส่งข้อมูลแบบอนุกรม

4.4.4 รีจิสเตอร์ตำแหน่ง 03H (รีจิสเตอร์กำหนดรูปแบบของข้อมูล)

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DLAB	BRK	PAR2	PAR1	PAR0	STOP	DAB1	DAB0

ตารางที่ 4.5 ฟังก์ชั่นการทำงานของรีจีสเตอร์ 03H

DLAB ใช้ในการกำหนดหน้าที่การทำงานของรีจิสเตอร์บัฟเฟอร์ (00H)

"1" เป็นการเข้าสู่โหมดการหาร่าบอดเรต

"0" เป็นการเข้าถึงรีจิสเตอร์บัฟเฟอร์(รีจิสเตอร์ตำแหน่ง 00H) และรีจิสเตอร์สำหรับเอ็นเอเบิลการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง01H) เมื่อบิตDLAB เป็น "1" รีจิสเตอร์บัฟเฟอร์ (00H) และรีจิสเตอร์เอ็นเอเบิลการอินเตอร์รัปต์ (01H) จะใช้สำหรับโหลดค่าการหารความถี่สำหรับกำหนดค่าบอดเรต โดยรีจิสเตอร์ 00H เก็บค่าตัวหารไบต์ต่ำ ส่วนรีจิสเตอร์ 01H ใช้เก็บค่าตัวหารไบต์สูง การหาค่าบอดเรตสามารถเขียนเป็นสมการได้ดังนี้

บอดเรต = 115200/ ค่าตัวหาร16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าตัวเลข 115200 มาจากความถี่คริสตอลในวงจร UART ภายในเครื่องคอมพิวเตอร์ โดยคริสตอลที่ใช้มีความถี่ 1.8432 MHz วงจรภายใน UART จะหารค่าความถี่นี้ด้วย 16 ทำให้ได้ค่าความถี่ 115200 Hz ออกมา ค่าตัวหาร16บิต = ข้อมูลในรีจิสเตอร์00H + (56*ข้อมูลในรีจิสเตอร์01H) ถ้าต้องการบอดเรตเท่ากับ 9600 ค่าตัวหารที่ใช้จะต้องมีค่าเท่ากับ 12 ซึ่งค่านี้จะถูกเขียนลงในรีจิสเตอร์ 00H และเขียนค่า 0 ลงไปในรีจิสเตอร์ 01H ค่าตัวหารที่ทำให้เกิดค่าบอดเรตสูงสุดที่ 115200 บิตต่อวินาที คือ ค่า 0001 นั่นคือ รีจิสเตอร์ 00H มีค่าเท่ากับ 1 และรีจิสเตอร์ 01H มีค่าเท่ากับ 0

BRK ใช้ควบคุมการหยุดถ่ายทอดข้อมูล

"1" สามารถหยุดหรือเบรกได้

"0" ไม่มีการหยุดหรือเบรกได้

PAR2,PAR1,PAR0 ใช้เพื่อกำหนดบิตพาริตี้ได้

"000" ไม่ใช้บิตพาริตี้

"001" กำหนดพาริตี้คี่

"011" กำหนดพาริตี้คู่

"101" มาร์ก (Mark)

"111" ช่องว่าง(Space)

STOP ใช้กำหนดจำนวนบิตปิดท้าย

"1" มีบิตปิดท้าย 2 บิต

"0" มีบิตปิดท้าย 1 บิต

DAB1,DAB0 ใช้ร่วมกันในการกำหนดจำนวนบิตของข้อมูลที่ต้องการถ่ายทอด

"00" จำนวนบิตข้อมูลเท่ากับ 5 บิต

"01" จำนวนบิตข้อมูลเท่ากับ 6 บิต

"10" จำนวนบิตข้อมูลเท่ากับ 7 บิต

"11" จำนวนบิตข้อมูลเท่ากับ 8 บิต

4.4.5 รีจิสเตอร์ตำแหน่ง 04H (รีจิสเตอร์ควบคุมโมเด็ม)

มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	LOOP	OUT2	OUT1	RTS	DTR

ตารางที่ 4.6 ฟังก์ชั่นการทำงานของรีจีสเตอร์ 04H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต 5-7 ไม่มีการใช้งาน อ่านค่าได้เท่ากับ 0

LOOP "1" เอ็นเอเบิลการส่งค่ากลับ

"0" ดิสเอเบิล

OUT1,OUT2 "1" เอ็นเอเบิลการใช้งานภายใน

"0" ดิสเอเบิล

RTS ใช้ควบคุมการทำงานของขาRTS(Ready TO Send)

"1" เอ็นเอเบิล

"0" ดิสเอเบิล

DTR ใช้ควบคุมการทำงานของขาDTR (Data Terminal Ready)

"1" เอ็นเอเบิล

"0" ดิสเอเบิล

4.4.6 รีจิสเตอร์ตำแหน่ง 05H (รีจิสเตอร์แสดงสถานะรอรับและส่งข้อมูลอนุกรมของ UART)

ใช้งานร่วมกับรีจิสเตอร์แสดงโหมดและสถานะของการอินเตอร์รัปต์ (รีจิสเตอร์ตำแหน่ง 02H) เพื่อแสดงสาเหตุของการเกิดอินเตอร์รัปต์ มีรายละเอียดหน้าที่ของแต่ละบิตดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	TXE	TBE	BREK	FRME	PARE	OVFE	RxRD

ตารางที่ 4.7 ฟังก์ชั่นการทำงานของรีจีสเตอร์ 05H

TXE(Transmitter Empty)

"1" แสดงว่ารีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" แสดงว่ายังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

TBE(Transmitter Buffer Empty)

"1" รีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูลว่าง

"0" ยังคงมีข้อมูล 1 ไบต์เก็บอยู่ในรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล

BREK(Break)

"1" UART ตรวจพบการเบรก

"0" ไม่มีการเบรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FRME (Frame Error)

"1" UART ตรวจพบความผิดพลาดด้านเฟรมข้อมูล

"0" ไม่พบความผิดพลาดทางพาริตี้

OVRE(Overrun Error)

"1" UART ตรวจพบความผิดพลาดแบบโอเวอร์รัน

"0" ไม่พบความผิดพลาดแบบโอเวอร์รัน

RxRD(Received Data Ready)

"1" มีการรับข้อมูลเข้ามาเก็บไว้ในบัฟเฟอร์

"0" ไม่มีข้อมูล

4.4.7 รีจิสเตอร์ตำแหน่ง 06H (รีจิสเตอร์แสดงสถานะของโมเด็ม)

ใช้เพื่อกำหนดสถานะสัญญาณอินพุต ของพอร์ตอนุกรม RS-232 ซึ่งได้แก่ สัญญาณ DCD,DSR,CTS และ RI สำหรับการเชื่อมต่อใช้งานแบบอเนกประสงค์ ดังมีรายละเอียดหน้าที่ของแต่ละบิตต่อไปนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
DCD	RI	DSR	CTS	DDCD	DRI	DDSR	DCTS

ตารางที่ 4.8 ฟังก์ชั่นการทำงานของรีจีสเตอร์ 06H

DCD ใช้แสดงสภาวะของขา DCD

"1" แสดงว่าที่ขา DCD เป็นลอจิก "1"

"0" แสดงว่าที่ขา DCD เป็นลอจิก "0"

RI ใช้แสดงสภาวะของขา RI

"1" แสดงว่าที่ขา RI เป็นลอจิก "1"

"0" แสดงว่าที่ขา RI เป็นลอจิก "0"

DSR ใช้แสดงสภาวะของขา DSR

"1" แสดงว่าที่ขา DSR เป็นลอจิก "1"

"0" แสดงว่าที่ขา DSR เป็นลอจิก "0"

DCTS(Delta Clear To Send) ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต CTS

"1" แสดงว่าบิตCTS เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0" แสดงว่าบิตCTS ไม่เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

DRI (Delta Ring Indicator)ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

"1" แสดงว่าบิตRI เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าบิตRIไม่ เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

DDCD(Delta Data Carrier Detect)ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นในบิตDDCD

"1" แสดงว่าบิต DDCD เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่ เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

DCTS(Delta Clear to send) ใช้แสดงสภาวะของ CTS

"1" แสดงว่าที่ขา CTS มีลอจิกเป็น "1"

"0" แสดงว่าที่ขา CTS มีลอจิกเป็น "0"

4.4.8 รีจิสเตอร์ตำแหน่ง07H (รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว)

ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบต์ การอ่านและเขียนข้อมูลที่รีจิสเตอร์ตัวนี้ไม่ส่งผลใดๆ ต่อการใช้งานUART

4.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ตRS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTS) และสัญญาณแสดงสถานะอินพุต(CTS,CSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจากUART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับแรงดันให้ได้ระดับแรงดันเป็นไปตามมาตรฐานRS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะดังแสดงเป็นบล็อกไดอะแกรมในรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

"0" แสดงว่าบิตCTS ไม่เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

DRI (Delta Ring Indicator)ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นของบิต RI

"1" แสดงว่าบิตRI เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าบิตRIไม่ เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

DDCD(Delta Data Carrier Detect)ใช้แจ้งการเปลี่ยนแปลงที่เกิดขึ้นในบิตDDCD

"1" แสดงว่าบิต DDCD เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

"0" แสดงว่าไม่ เกิดการเปลี่ยนแปลงเมื่อเทียบจากการอ่านค่าครั้งที่แล้ว

DCTS(Delta Clear to send) ใช้แสดงสภาวะของ CTS

"1" แสดงว่าที่ขา CTS มีลอจิกเป็น "1"

"0" แสดงว่าที่ขา CTS มีลอจิกเป็น "0"

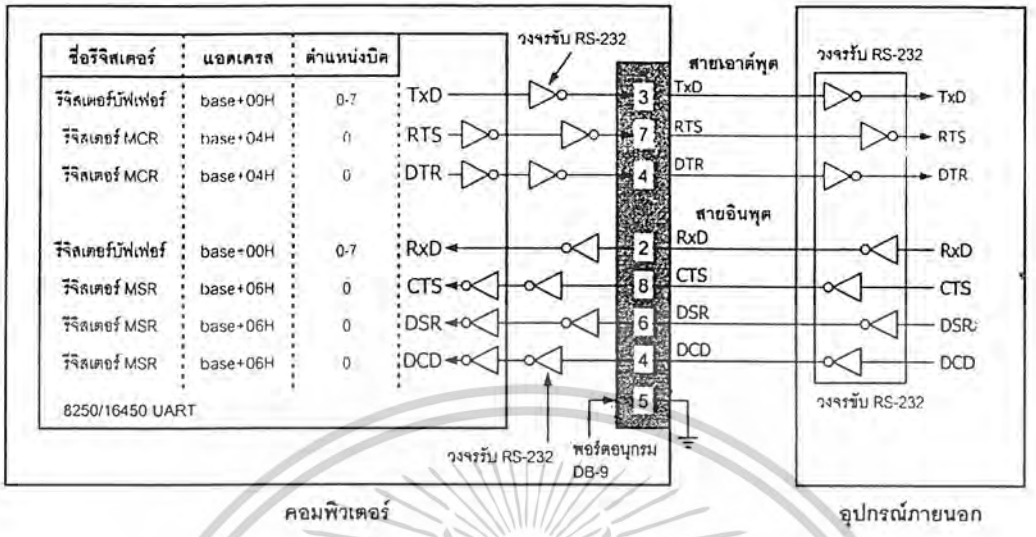
4.4.8 รีจิสเตอร์ตำแหน่ง07H (รีจิสเตอร์สำหรับเก็บข้อมูลชั่วคราว)

ทำหน้าที่เป็นหน่วยความจำแรมขนาด 1 ไบต์ การอ่านและเขียนข้อมูลที่รีจิสเตอร์ตัวนี้ไม่ส่งผลใดๆ ต่อการใช้งานUART

4.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ตRS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTS) และสัญญาณแสดงสถานะอินพุต(CTS,CSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจากUART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับแรงดันให้ได้ระดับแรงดันเป็นไปตามมาตรฐานRS-232 ก่อนส่งออกไปจากคอมพิวเตอร์ สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะดังแสดงเป็นบล็อกไดอะแกรมในรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 ไดอะแกรมแสดงโครงสร้างทางฮาร์ดแวร์ของพอร์ตอนุกรม

4.6 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

COM1 : 3F8H

COM2 : 2F8H

COM3 : 3E8H

COM4 : 2E8H

บิต 3	บิต 2	บิต1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 1 พอร์ต
0	1	1	มีพอร์ตอนุกรม 2 พอร์ต
1	0	0	มีพอร์ตอนุกรม

ตารางที่ 4.9 ข้อมูลในแอดเดรส 0000:0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรมไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2ไบต์ สำหรับพอร์ตอนุกรม COM1จะเก็บไว้ที่แอดเดรส 0000:0400H และ 0000:0401H ส่วนตำแหน่งอื่นๆมีรายละเอียดดังนี้

COM2 = 0000:0402H – 0000:0403H

COM3 = 0000:0404H – 0000:0405H

COM4 = 0000:0406H – 0000:0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000:0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในเครื่องคอมพิวเตอร์อีกด้วย โดยรายละเอียดดังแสดงในตารางที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 ไมโครคอนโทรลเลอร์ MCS-51

5.1 ไมโครคอนโทรลเลอร์ MCS-51

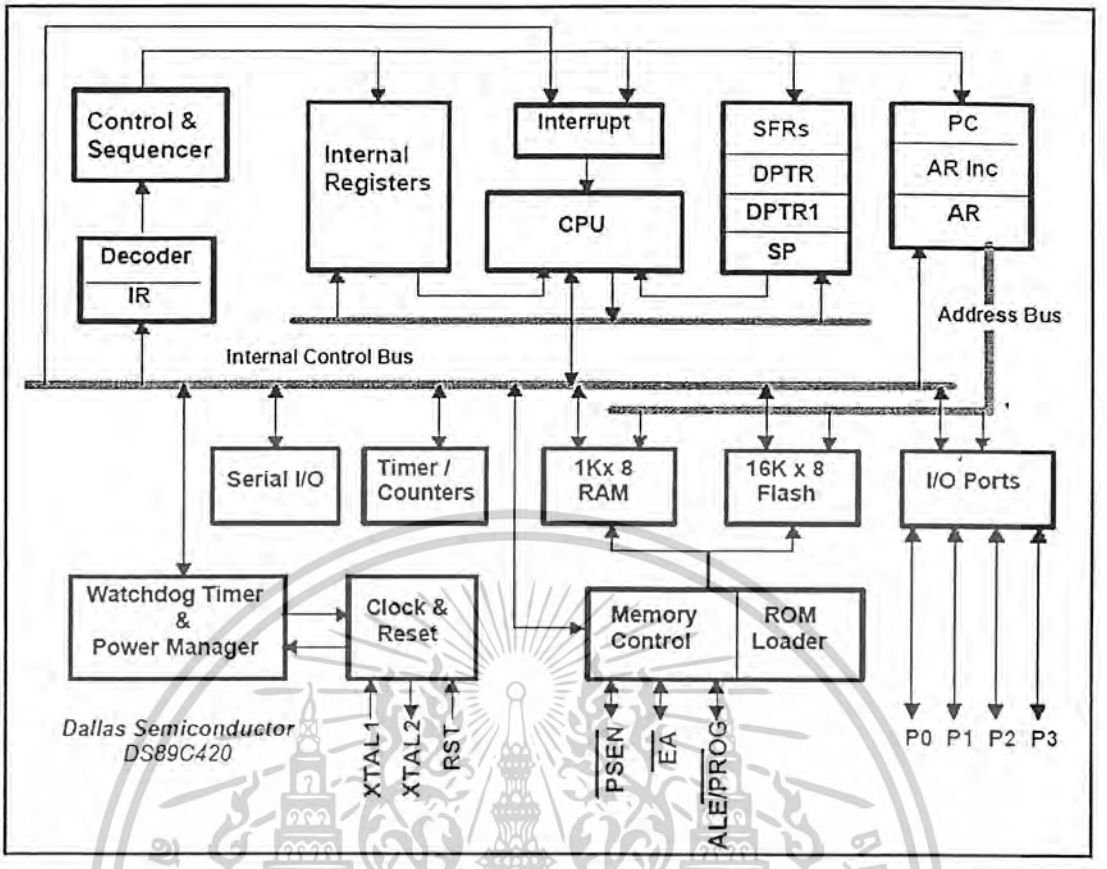
ไมโครคอนโทรลเลอร์ MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตที่มีอุปกรณ์สนับสนุนประกอบอยู่ภายในหลายอย่างได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายในนี้เอง ทำให้การใช้งานง่ายขึ้นและมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับ ตัวไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้หากเราต้องการใช้งานไมโครคอนโทรลเลอร์ร่วมกับ อุปกรณ์อื่นเพิ่มเติมก็สามารถกระทำได้ง่ายอีกด้วย

5.2 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ DS89C420 ที่แสดงในรูปที่ 5.1 ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (BOOLEAN PROCESSOR)
- หน่วยความจำโปรแกรมภายในขนาด 16 กิโลไบต์
- หน่วยความจำแบบ แรม ภายในจำนวน 256 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวนสามวงจร
- 2 วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็ก (FULL DUPLEX)
- วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 13 ประเภท 6 ภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ DS89C420

โดยไมโครคอนโทรลเลอร์ตระกูลนี้จะมีรูปร่างของไอซีเป็นแบบขนาด 40 ขา ดังแสดงในรูปที่ 5.2 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อ ที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ภายในไมโครคอนโทรลเลอร์ MCS-51 จะประกอบด้วยหน่วยการทำงานต่างๆ ภายในไอซี MCS-51 จำนวนมาก โดยแต่ละบล็อกซึ่งเป็นวงจร

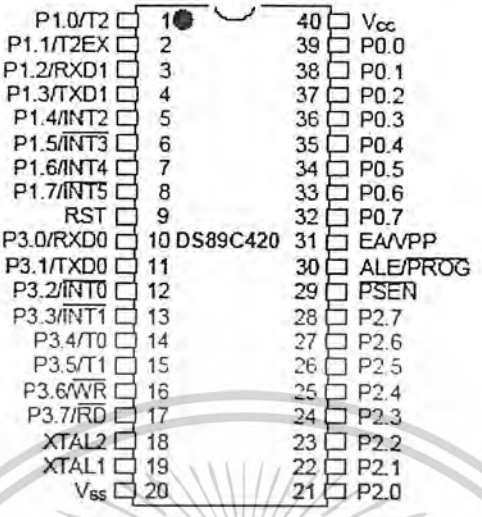
การควบคุมรีจิสเตอร์ (REGISTER) หรือหน่วยความจำภายในของไอซี MCS-51 จะถูกเชื่อมต่อเข้าด้วยกันผ่านทางเส้นสัญญาณที่เรียกว่าบัสข้อมูลภายใน รีจิสเตอร์และหน่วยความจำเหล่านี้จะถูกนำไปใช้ระหว่างการประมวลผลคำสั่ง หน้าที่ของโปรแกรมที่ผู้ใช้สร้างขึ้นมาก็เป็นการควบคุมการรับหรือส่งข้อมูลระหว่างรีจิสเตอร์เหล่านี้ ซึ่งอาจจะมีการดำเนินการร่วมกับหน่วยการดำเนินงานประมวลผลทางคณิตศาสตร์และลอจิก(Arithmatic and Logic Unit (ALU)

5.3 โครงสร้างหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 แยกการจัดการหน่วยความจำออกเป็นสองส่วน คือ หน่วยความจำโปรแกรม (PROGRAM MEMORY) และหน่วยความจำข้อมูล (DATA MEMORY) หน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำทั้งสองนี้ มีหน้าที่แตกต่างไปจากกัน และใช้วิธีการอ้างแอดเดรสสัญญาณการติดต่อแยกออกจากกัน



รูปที่ 5.2 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51

5.4 หน่วยความจำโปรแกรม

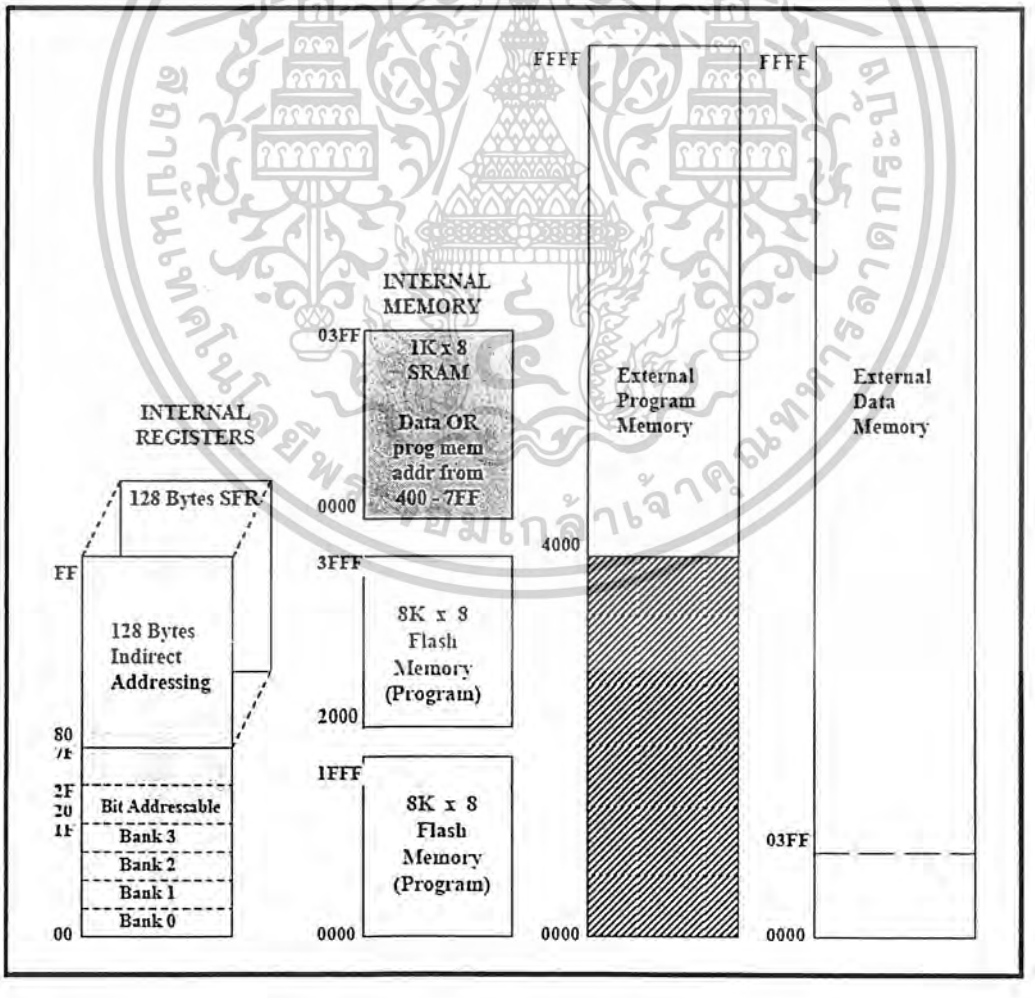
หน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหายโครงสร้างของหน่วยความจำโปรแกรม มีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไมโครคอนโทรลเลอร์ MCS-51 ของหน่วยความจำ ประเภทต่างๆ เช่น หน่วยความจำแบบรอม (READ ONLY MEMORY) หรือ อีพรอม (ERASABLE PROGRAMABLE READ ONLY MEMORY) ในไมโครคอนโทรลเลอร์ MCS-51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (INTERNAL PROGRAM MEMORY) ซึ่งเป็นหน่วยความจำรอม หรือ อีพรอม ที่อยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง ในเบอร์ DS89C420 ที่ใช้สามารถอ้างตำแหน่งได้ตั้งแต่ 0000h - 3FFFh และหน่วยความจำโปรแกรมภายนอก (EXTERNAL PROGRAM MEMORY) ตั้งแต่ 4000h – FFFFh ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

ไมโครคอนโทรลเลอร์เบอร์ต่างๆนี้สามารถขยายให้ใช้งาน ในหน่วยความจำภายนอกได้ทั้งสิ้น โดยกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้ว การอ้างตำแหน่งแอดเดรสที่มี ทั้งในหน่วยความจำโปรแกรมภายใน และภายนอกนั้นจะต้องทำการควบคุมระดับลอจิกของสัญญาณ ในขณะนั้นด้วย ขนาดหน่วยความจำโปรแกรมภายในของไมโครคอนโทรลเลอร์เบอร์ต่างๆจะแตกต่างกันออกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 หน่วยความจำข้อมูล

หน่วยความจำข้อมูล (DATA MEMORY) ซึ่งโดยพื้นฐานแล้วเป็นหน่วยความจำแรมสามารถเขียนหรืออ่านข้อมูลได้ (READ OR WRITE MEMORY) ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว ซึ่งโดยพื้นฐานแล้วหน่วยความจำข้อมูลจัดเป็นหน่วยความจำแรมแบบสแตติก ดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ ภายในหน่วยความจำนี้สูญไป พื้นที่ของหน่วยความจำข้อมูล ของไมโครคอนโทรลเลอร์ MCS-51 ทั่วไปมีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น ตามลักษณะของหน่วยความจำโปรแกรมภายในซึ่งก็เป็นแรมที่อยู่ภายในตัวไอซีในตระกูลของไมโครคอนโทรลเลอร์ และหน่วยความจำข้อมูลภายนอกซึ่งเป็นการใช้ไอซีหน่วยความจำแรมมาเพิ่มเติมเข้าไปในวงจรลักษณะเดียวกับ การนำไอซีอีพรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง



รูปที่ 5.3 แสดงการจัดการหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่หน่วยความจำสำหรับเก็บข้อมูลของไมโครคอนโทรลเลอร์นี้ สามารถแบ่งออกเป็น 2 ส่วนคือ ในส่วนที่เป็นหน่วยความจำสำหรับเก็บข้อมูลภายในไอซี และหน่วยความจำสำหรับเก็บข้อมูลภายนอก ไมโครคอนโทรลเลอร์แต่ล่ะเบอร์จะมีหน่วยความจำเก็บข้อมูลทั่วๆ ไปภายในไอซีไม่เท่ากันขึ้นกับเบอร์ของไอซี แต่ในเบอร์ DS89C420 มี 256 ไบต์

5.6รีจิสเตอร์ที่เกี่ยวกับไมโครคอนโทรลเลอร์MCS-51

รีจิสเตอร์ในกลุ่มนี้จะเป็นรีจิสเตอร์ขนาด 16บิตที่ใช้งานเพื่อเก็บข้อมูลของตัวแอดเดรสเป็นสำคัญโดยค่าที่อยู่ภายในแอดเดรสนี้จะนำไปเป็นค่าของข้อมูลที่ส่งออกไปทางบัสแอดเดรส ในส่วนของไมโครคอนโทรลเลอร์ เพื่อบอกตำแหน่งที่ต้องการติดต่อ รีจิสเตอร์ที่จัดในกลุ่มนี้ประกอบด้วย รีจิสเตอร์ใช้งานทั่วไป (GENERAL-PURPOSE REGISTERS) รีจิสเตอร์ในกลุ่มนี้จัดเป็นพื้นที่หน่วยความจำที่ใช้ในการสนับสนุนในการประมวลผล การทำงานจากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) เพื่อให้สามารถจัดการข้อมูลให้เร็วที่สุด นอกจากนี้โปรแกรมที่ไม่ได้ใช้คำสั่งเหล่านี้ก็ยังใช้เป็นการเก็บข้อมูลตัวแปรภายในโปรแกรม จะเห็นได้ว่าชื่อของรีจิสเตอร์ไม่ว่าจะอยู่ในรีจิสเตอร์แบงก์ใด ก็จะมีชื่อว่า R0 ถึง R7 เหมือนกันทั้งสิ้น ดังนั้นในการใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่า ต้องการรีจิสเตอร์นั้นๆ จากแบงก์ใดๆ ซึ่งการกำหนดเลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในแฟลก (PSW) เท่านั้นอย่างไรก็ตามโดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 ถึง R7 เฉพาะในแบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำแรม

5.6.1 รีจิสเตอร์หน้าที่พิเศษ

เป็นรีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับ การควบคุมหน้าที่ และการทำงานของอุปกรณ์หรือพอร์ตของไมโครคอนโทรลเลอร์ทั้งหมด ตำแหน่งของรีจิสเตอร์เหล่านี้จะจัดอยู่ในบริเวณแอดเดรส 80H - FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรส ที่เป็นของรีจิสเตอร์นั้นก็ได้ การจัดพื้นที่หน่วยความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่ารีจิสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่มีจำนวนเป็นทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H 88H 90H A0H A8H B0H B8H D0H E0H และ F0H)

5.6.2แอกคิวมูเลเตอร์ (ACCUMULATOR)

หรือ ACC เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในหน่วยประมวลผลกลาง และเก็บผลลัพธ์ที่ได้จากการทำงานเท่านั้น การทำงานของรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นมีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานในโปรแกรมซึ่งใช้เรียกเป็น รีจิสเตอร์ A

5.7 โครงสร้างการอินเตอร์รัปต์ของไมโครคอนโทรลเลอร์ MCS-51

จากรูปที่ 5.5 โครงสร้างระบบอินเตอร์รัปต์ของ ไมโครคอนโทรลเลอร์ MCS-51สัญญาณที่เข้ามาทำการอินเตอร์รัปต์ MCS-51 นั้นเกิดขึ้นได้ 13 ลักษณะ (6 ภายนอก) ตามตารางในรูปที่ 5.5 โดยจะเห็นได้ว่าสามารถที่จะกำหนดเลือกเพื่อยินยอม (หรืออีนาเบิล : ENABLE) และห้าม (หรือดิสเอเบิล : DISABLE) ไม่ให้มีการอินเตอร์รัปต์แต่ละประเภทได้ โดยการกำหนดบิตของข้อมูลที่เกี่ยวข้องซึ่งมักจะอยู่ภายในรีจิสเตอร์ TCON และ SCON นอกจากนี้ยังมีตำแหน่งบิตภายในรีจิสเตอร์ IE (INTERRUPT ENABLE REGISTER) ซึ่งทำหน้าที่เสมือนกับเป็นสวิตซ์หลักที่เกี่ยวข้องกับสัญญาณอินเตอร์รัปต์ทั้งหมด หากว่ากำหนดไม่ให้เกิดการอินเตอร์รัปต์แล้วการกำหนดบิตเพื่อห้ามหรือยินยอมของแต่ละอินเตอร์รัปต์ก็จะไม่มีผลใดๆเกิดขึ้น ยังแสดงให้เห็นว่าสัญญาณอินเตอร์รัปต์แต่ละประเภทยังสามารถกำหนดระดับความสำคัญ (PRIORITY) ของการอินเตอร์รัปต์ได้สองลักษณะ คือ ระดับความสำคัญสูงหรือต่ำ (HIGH OR LOW PRIORITY) กล่าวคือขณะที่กำลังประมวลผลอยู่ภายในส่วนของโปรแกรมย่อย บริการอินเตอร์รัปต์ของสัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็อาจจะถูกขัดจังหวะให้ไปประมวลผลของสัญญาณอินเตอร์รัปต์ที่มีระดับความสำคัญสูงกว่า แต่หากว่าเป็นสัญญาณอินเตอร์รัปต์ที่มีระดับความสำคัญต่ำเช่นเดียวกันแล้ว ก็ต้องรอให้เสร็จสิ้นการประมวลผลที่ ดำเนินการอยู่ก่อน

ในไมโครคอนโทรลเลอร์เบอร์ DS89C420 นี้มีการจัดลำดับความสำคัญอยู่ 5 กลุ่ม ตั้งแต่ระดับ 4 ถึง 0 โดยที่ระดับที่มีความสำคัญสูงสุดคือระดับ 4 ซึ่งถูกจองไว้สำหรับอินเตอร์รัปต์ power – fail และ อินเตอร์รัปต์อื่นๆ ก็จะถูกกำหนดลำดับความสำคัญตั้งแต่ 3 ถึง 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERRUPT	VECTOR	NATURAL ORDER	FLAG	ENABLE	PRIORITY CONTROL
Power-Fail	33h	0 (Highest)	PFI (WDCON.4)	EPFI(WDCON.5)	N/A
External Interrupt 0	03h	1	IE0 (TCON.1)**	EX0 (IE.0)	LPX0 (IP0.0) MPX0 (IP1.0)
Timer 0 Overflow	0Bh	2	TF0 (TCON.5)*	ET0 (IE.1)	LPT0 (IP0.1) MPT0 (IP1.1)
External Interrupt 1	13h	3	IE1 (TCON.3)**	EX1 (IE.2)	LPX1 (IP0.2) MPX1 (IP1.2)
Timer 1 Overflow	1Bh	4	TF1 (TCON.7)*	ET1 (IE.3)	LPT1 (IP0.3) MPT1 (IP1.3)
Serial Port 0	23h	5	RI_0 (SCON0.0) TI_0 (SCON0.1)	ES0 (IE.4)	LPS0 (IP0.4) MPS0 (IP1.4)
Timer 2 Overflow	2Bh	6	TF2 (T2CON.7) EXF2 (T2CON.6)	ET2 (IE.5)	LPT2 (IP0.5) MPT2 (IP1.5)
Serial Port 1	3Bh	7	RI_1 (SCON1.0) TI_1 (SCON1.1)	ES1 (IE.6)	LPS1 (IP0.6) MPS1 (IP1.6)
External Interrupt 2	43h	8	IE2 (EXIF.4)	EX2 (EIE.0)	LPX2 (EIP0.0) MPX2 (EIP1.0)
External Interrupt 3	4Bh	9	IE3 (EXIF.5)	EX3 (EIE.1)	LPX3 (EIP0.1) MPX3 (EIP1.1)
External Interrupt 4	53h	10	IE4 (EXIF.6)	EX4 (EIE.2)	LPX4 (EIP0.2) MPX4 (EIP1.2)
External Interrupt 5	5Bh	11	IE5 (EXIF.7)	EX5 (EIE.3)	LPX5 (EIP0.3) MPX5 (EIP1.3)
Watchdog	63h	12 (Lowest)	WDIF (WDCON.3)	EWDI (EIE.4)	LPWDI (EIP0.4) MPWDI (EIP1.4)

ตารางที่ 5.1 การอินเตอร์รัปต์ในไมโครโทรลเลอร์ MCS-51 เบอร์ DS89C420

5.8 ไทเมอร์/ เคาน์เตอร์ของไมโครคอนโทรลเลอร์ MCS-51

ไทเมอร์ / เคาน์เตอร์ เป็นอีกส่วนหนึ่งที่สำคัญของไมโครคอนโทรลเลอร์ เนื่องจากในการทำงานของไมโครคอนโทรลเลอร์จะต้องมีการเก็บ และตรวจสอบค่าของเวลาและจำนวนสัญญาณนาฬิกาอยู่ตลอดเวลา ทั้งนี้เพื่อประโยชน์ในการสร้างฐานเวลา สร้างสัญญาณพัลส์ เปรียบเทียบค่าเวลา หรือ ค่าการนับ รวมไปถึงการกำหนดอัตราเร็วในการสื่อสารข้อมูลของพอร์ตอนุกรมด้วย

ในการใช้งานไทเมอร์ / เคาน์เตอร์นั้น จะต้องมีการกำหนดหรือควบคุมการทำงานของไทเมอร์ / เคาน์เตอร์ โดยใช้รีจิสเตอร์ที่สำคัญ 2 ตัว คือ

TCON (Timer / Counter Control Register) : มีแอดเดรสอยู่ที่ 88H เป็นรีจิสเตอร์ที่ใช้ควบคุมการเปิด / ปิดไทเมอร์แต่ละตัว , และแสดงถึงการเกิดโอเวอร์โฟรวที่เกิดจากไทเมอร์ / เคาน์เตอร์ อีกทั้งเป็นตัวกำหนดลักษณะการเกิดอินเตอร์รัปต์ของสัญญาณภายนอกด้วย

TMOD (Timer / Counter Mode Control Register) : มีแอดเดรสอยู่ที่ 89H เป็นรีจิสเตอร์ที่ใช้กำหนดการทำงานว่าเป็นไทเมอร์ หรือ เคาน์เตอร์ อีกทั้งเป็นตัวกำหนดโหมดการทำงานของไทเมอร์ / เคาน์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.9 พอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์DS89C420 มีวงจรสื่อสารอนุกรมแบบ ฟลูดูเพล็กซ์ 2 ชุด โดยใช้ขาสัญญาณของพอร์ต 3 คือ ขาP3.0 เป็นขารับข้อมูลหรือ RxD0 และขา P3.1 เป็นขาส่งข้อมูลหรือ TxD0 และขาสัญญาณของพอร์ต 1 คือ ขาP1.2 เป็นขารับข้อมูลหรือ RxD1 และขา P1.3 เป็นขาส่งข้อมูลหรือ TxD1โดยวงจรสื่อสารข้อมูลแบบอนุกรมของ MCS-51 เป็นแบบอะซิงโครนัส ปกติแล้วพอร์ตอนุกรมจะใช้ติดต่อสื่อสารกับพอร์ตอนุกรมของคอมพิวเตอร์ โดยใช้มาตรฐาน RS-232 แต่ในปัจจุบัน สามารถติดต่อกับ RS-422 หรือ RS-485 ได้แล้ว โดยใช้ IC พิเศษ ทำหน้าที่ในการแปลงสัญญาณการสื่อสาร

รีจิสเตอร์ที่เกี่ยวข้องกับการทำงานของพอร์ตอนุกรมในMCS -51 ได้แก่ SBUF (Serial data buffer register) ซึ่งเป็นบัฟเฟอร์สำหรับ ส่งข้อมูล และรับข้อมูล และรีจิสเตอร์อีกตัวที่สำคัญคือ SCON (Serial Port Control) ซึ่งใช้ในการกำหนดโหมดการทำงานของพอร์ตอนุกรม ซึ่งมี 4 โหมดด้วยกัน คือ

1. โหมด 0 เป็นการกำหนดให้พอร์ตอนุกรมทำงานในลักษณะชิฟรีจิสเตอร์
2. โหมด 1 เป็นการกำหนดให้UART ขนาด 8 บิต สามารถเลือกอัตรา บอร์ดได้
3. โหมด 2 UART ขนาด 8 บิต สามารถเลือกอัตรา บอร์ดได้
4. โหมด 3 เป็นตัวกำหนดให้เป็น UART ขนาด9 บิต สามารถเลือกอัตราบอดได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6 การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ด้วยภาษาซี

การเขียนโปรแกรมสำหรับไมโครคอนโทรลเลอร์ด้วยภาษาแอสเซมบลีนั้น โครงสร้างจะดูยุ่งยาก การควบคุมโปรแกรมก็เข้าใจยาก ในอดีตนั้นการเขียนโปรแกรมด้วยภาษาซีสำหรับการควบคุมไมโครคอนโทรลเลอร์นั้นเมื่อแปลภาษาออกมาเป็นภาษาเครื่องแล้วนั้น จะมีขนาดของไฟล์ที่ใหญ่กว่าการเขียนด้วยภาษาแอสเซมบลีมาก แต่ในปัจจุบันคอมไพเลอร์ส่วนใหญ่จะแปลภาษาออกมาได้มีขนาดเล็กลงเกือบใกล้เคียงภาษาแอสเซมบลี

การเขียนโปรแกรมด้วยภาษาซีจะมีขั้นตอนต่างๆ ดังนี้

1. เขียนชุดคำสั่งบนโปรแกรม Editor ให้มีนามสกุลเป็น .C
2. กำหนดพรีโพรเซสเซอร์ (preprocessor) สำหรับกำหนดการคอมไพล์โปรแกรม ซึ่งอาจกำหนดไว้ใน source code หรือกำหนดตอมคอมไพล์ก็ได้
3. คอมไพล์โปรแกรม ในที่นี้จะใช้โปรแกรม C51 และจะได้ไฟล์ออปเจกต์ ที่มีนามสกุลเป็น OBJ ออกมา
4. เชื่อมต่อชุดคำสั่งเสริม (link) โดยใช้โปรแกรม L51 ซึ่งจะทำการรวมชุดคำสั่งต่างๆ ที่อ้างอิงถึง และโปรแกรมจะสร้างออปเจกต์ไพล์ที่สมบูรณ์ออกมา
5. แปลงให้เป็นไพล์ hexadecimal file ด้วยโปรแกรม OHS51 ซึ่งจะได้ไฟล์นามสกุล .HEX ออกมา

ถ้าหากใช้ซีคอมไพเลอร์ตัวอื่นๆ ขั้นตอนเหล่านี้อาจแตกต่างกันไป

6.1 โครงสร้างของภาษาซี

ด้วยภาษาซีเป็นภาษาที่สามารถเขียนโปรแกรมเป็นแบบโครงสร้างได้ โดยโปรแกรม แบ่งการทำงานต่าง ๆ ออกเป็นกลุ่ม ๆ หรือฟังก์ชัน โดยฟังก์ชันเหล่านั้นสามารถเรียกขึ้นมาใช้ใหม่ได้ ในการเขียนโปรแกรมจะต้องระบุไว้ว่าในโปรแกรมนั้นมีฟังก์ชันใดให้ใช้บ้าง แต่ทุกโปรแกรมจะต้องมีฟังก์ชันหลักที่ชื่อว่า main () เสมอ เช่น ตัวอย่างโครงสร้างของโปรแกรมต่อไปนี้

```
#include < reg52.h>          / * Preprocessor * /
void  func1 (void);           / * Pro to type * /
int  func2 (int x);
void  main ()                 / * ฟังก์ชันหลัก * /
{
      int a;                  / * ประกาศตัวแปร * /
      P1 = 0x0FF;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Func1 ()                    / * เรียกใช้ฟังชัน * /
        a = func2 ( 4 );            / * เรียกใช้ฟังก์ชันที่มีการส่งค่า * /
        P = a
}
void func1 (void)                   / * ฟังก์ชันที่ไม่มีการคืนค่า * /
{
        .....
        .....
}
int func2( int x)                   / * ฟังก์ชันที่มีการคืนค่า * /
{
        return (x*2);
}
```

จากโปรแกรมจะพบว่า ส่วนประกาศโพรโตไทป์จะบอกว่าโปรแกรมนี้มีฟังก์ชันชื่อ func1 ให้ใช้งานและฟังก์ชันนี้จะทำงานเป็นโปรแกรมย่อยเพราะมี void นำหน้า และมีฟังก์ชันชื่อ func2 (int x) ซึ่งจะรับค่าเข้าไปผ่านทางตัวแปร x และคืนค่าออกมาเป็นจำนวนเต็ม

ตัวแปรและค่าคงที่

การใช้งานตัวแปรและค่าคงที่ต่าง ๆ จะต้องมีการประกาศชื่อตัวแปรขึ้นมาเสียก่อนเมื่อมีการคอมไพล์โปรแกรมตัวคอมไพเลอร์จะเตรียมพื้นที่ในหน่วยความจำแรมเอาไว้สำหรับเก็บตัวแปรและค่าคงที่เหล่านั้นในการประกาศตัวแปรสามารถทำได้ดังนี้

ประเภทข้องข้อมูล	ขนาด (บิต)	ค่าที่เก็บได้
bit	1	0 ถึง 1
char	8	-128 ถึง 127
unsigned char	8	0 ถึง 255
int	16	-32768 ถึง 32767
unsigned int	16	0 ถึง 65535
long	32	-2147483648 ถึง + 2147483647
unsigned long	32	0 ถึง 4294967295
float	32	-1.17549e -38 ถึง + 3.402823e + 38

ตารางที่ 6.1 การประกาศตัวแปรและค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับคำสงวนเป็นคำที่คอมไพเลอร์รู้จักและจะถูกใช้งานเฉพาะ เราไม่สามารถนำตั้งเป็นชื่อตัวแปรและฟังก์ชันได้ คำสงวนของโปรแกรม C51 เป็นดังนี้

_at_	idata	sfr
alien	interrupt	sfr16
bdata	large	small
bit	pdata	_task_
code	_priority_	using
compact	reentrant	xdata
data	sbit	

6.2 ตัวดำเนินการในภาษาซี

ตัวนำเนินการจะเป็นตัวที่ใช้กระทำกับตัวแปร ค่าคงที่ต่าง ๆ ให้รวมเป็นค่าเดียวกันโดยอาจกระทำทางคณิตศาสตร์หรือกระทำทางลอจิกก็ได้ ในการเขียนโปรแกรมด้วยภาษาซีนั้น ตัวดำเนินการจะแบ่งออกเป็นสองกลุ่มใหญ่ๆ คือ ตัวดำเนินการที่กระทำกับ ตัวถูกกระทำตัวเดียว (single operand operators) ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว (two operands operators)

6.2.1 ตัวดำเนินการที่กระทำกับตัวถูกกระทำตัวเดียว

ตัวดำเนินการประเภทนี้จะกระทำกับตัวถูกกระทำเพียงตัวเดียว ประกอบด้วยตัวดำเนินการต่าง ๆ ดังนี้

- \- ลบ (negate)
- ~ กลับค่าลอจิกของบิตข้อมูล (bit wise complement)
- ! กลับค่าทางลอจิก (logical complement)
- ++ เพิ่มค่าขึ้นหนึ่งค่า (increment)
- -- ลดค่าลงหนึ่งค่า (decrement)
- * ตัวดำเนินการทางพอยน์เตอร์
- & ตัวตำแหน่งหน่วยความจำของตัวแปร

6.2.2 ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวดำเนินการประเภทนี้จะกระทำกับตัวถูกกระทำสองตัว ถ้าหากมีตัวถูกกระทำหลาย ๆ ตัว สามารถนำมาเขียนรวมกันเป็นประโยคได้ ซึ่งประกอบไปด้วยตัวดำเนินการที่ใช้กำหนดค่า ตัวดำเนินการทดสอบค่า ซึ่งจะทำให้ผลลัพธ์เป็นค่าทางลอจิก (จริง , เท็จ) ตัวดำเนินการทางคณิตศาสตร์ และตัวดำเนินการทางลอจิก ดังต่อไปนี้

=	กำหนดค่าในประโยค (assignment)
+	บวก
-	ลบ
*	คูณ
/	หาร (division)
%	หารแบบบวก (modulo)
&&	การแอนด์ (logical AND)
||	การออร์ (logical OR)
&	การแอนด์แบบบิตต่อบิต (bit wise AND)
|	การออร์แบบบิตต่อบิต (bit wise OR)
^	การเอ็กคลูซีฟออร์แบบบิตต่อบิต (bit wise exclusive OR)
<<	เลื่อนบิตไปทางซ้าย
>>	เลื่อนบิตไปทางขวา
==	ทดสอบว่าเท่ากันหรือไม่
!=	ทดสอบว่าไม่เท่ากันหรือไม่
>	ทดสอบว่ามากกว่าหรือไม่
<	ทดสอบว่าน้อยกว่าหรือไม่
>=	ทดสอบว่ามากกว่าหรือเท่ากับหรือไม่
<=	ทดสอบว่าน้อยกว่าหรือเท่ากับหรือไม่

นอกจากนี้ตัวดำเนินการบางประเภทสามารถนำมารวมกันเป็น compound operators ได้ ตัวอย่างเช่น ถ้ามีประโยค y = y * 2; อาจเขียนตัวดำเนินการรวมได้เป็น y *= 2; รูปแบบของตัวดำเนินการที่รวมกันได้ เป็นดังนี้

+=	บวกและให้เท่ากับ
-=	ลบและให้เท่ากับ
*=	คูณและให้เท่ากับ
/=	หารและให้เท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%= หารแบบบวกและให้เท่ากับ

&= ทำการแอนด์และให้เท่ากับ

|= ทำการออร์และให้เท่ากับ

^= ทำการเอ็กคลูซีฟออร์และให้เท่ากับ

<<= เลื่อนบิตไปทางซ้ายและให้เท่ากับ

>>= เลื่อนบิตไปทางขวาและให้เท่ากับ

6.3 ประโยคควบคุมในภาษาซี

การทำงานของโปรแกรมนั้นจะทำคำสั่งแต่ละคำสั่งเรียงลำดับกันไป และเราสามารถให้ใช้โปรแกรมตัดสินใจในการเลือกทำได้ หรือให้ทำงานใดงานหนึ่งซ้ำ ๆ ตามเงื่อนไขที่กำหนดได้โดยใช้คำสั่งควบคุมในภาษาซีนั้นจะมีประโยคคำสั่งควบคุมที่ใช้ในการเลือกทำและทำงานซ้ำ ๆ ดังนี้

6.3.1 ประโยค IF / ELSE

ประโยคคำสั่งนี้จะใช้ควบคุมทิศทางการทำงานของโปรแกรม โดยจะถูกแปลออกมาเป็นคำสั่งในภาษาแอสเซมบลีดังนี้ jz, jnz, jb, jnb, jc, และ jnc โดยมีรูปแบบประโยคดังนี้

ประโยคนี้ใช้ในการทดสอบว่าจะทำสเตตเมนต์ที่ตามมาหรือไม่ ถ้าค่าใน expression เป็นจริงหรือมีค่าไม่เป็นศูนย์จะทำสเตตเมนต์ที่ตามมา ถ้าเป็นเท็จหรือมีค่าเป็นศูนย์จะไม่ทำ และสเตตเมนต์ที่จะทำงานนั้นอาจเป็นประโยคคำสั่งประโยคเดียว หรือเป็นสเตตเมนต์ซ้อนก็ได้ (ต้องมีปีกกาคุม)

6.3.2 ประโยค switch

การเลือกทำที่มีทางเลือกหลาย ๆ ทางเลือกนั้นเราสามารถนำประโยค If-else มาซ้อนกันได้แต่จะทำให้มองดูเข้าใจได้ยาก ในภาษาซีจะมีประโยค switch ที่ใช้ในการเลือกทำอย่างใดอย่างหนึ่งจากหลาย ๆ ทางเลือกโดยมีรูปแบบของประโยคดังนี้

```
Switch (k)
{
        case 1 :        statement 1;
                        break;
        case 2 :        statement 2;
                        break;
        case 3 :        statement 3;
                        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        :                   :
        :                   :
    default :               statement n;
}
```

การทำงานของโปรแกรมจะนำค่าในตัวแปร k ที่อยู่ในวงเล็บหลัง switch มาเปรียบเทียบว่าเท่ากับค่าคงที่หลังคำสั่ง case ตัวใด และจะทำงานสเตตเมนต์ที่ตามหลัง case นั้น และจะออกนอกปีกกาของ switch เมื่อพบคำสั่ง break โดยสเตตเมนต์ที่ทำงานนั้นจะเป็นสเตตเมนต์ซ้อนก็ได้ แต่ถ้าค่าในตัวแปร k ไม่เท่ากับค่าคงที่ค่าใดเลยโปรแกรมจะทำสเตตเมนต์ที่ตามหลัง default

6.4 การทำซ้ำ

คำสั่งให้โปรแกรมทำงานซ้ำถือว่าเป็นประโยคคำสั่งควบคุมอย่างหนึ่ง การทำซ้ำหรือที่เรียกว่าการทำลูปนั้นจะมีประโยคคำสั่งอยู่สามประเภทคือ for, white และ do – white ซึ่งแต่ละแบบจะต่างกันตรงเงื่อนไขของการทำซ้ำ

6.4.1 ประโยค for

ประโยคคำสั่งนี้จะใช้ในกรณีที่มีจำนวนรอบของการกระทำซ้ำที่แน่นอน โดยมีรูปแบบดังนี้

```
For (initialization ; condition ; increment)
        Statement ;
```

โดยที่ initialization เป็นค่ากำหนดเริ่มต้นให้กับตัวแปรของการทำลูป condition เป็นเงื่อนไขที่ใช้ทดสอบการทำซ้ำครั้งต่อไป ซึ่งจะเป็นการกระทำลอจิก increment เป็นการเพิ่มค่าให้ตัวแปรในการทำซ้ำแต่ละครั้ง สำหรับ statement จะเป็นสเตตเมนต์ของคำสั่งที่จะทำซ้ำ ซึ่งอาจเป็น สเตตเมนต์รวม

6.4.2 ประโยค while

การทำซ้ำแบบนี้จะตรวจสอบเงื่อนไขก่อนการทำซ้ำ ถ้าเงื่อนไขเป็นจริงจะทำสเตตเมนต์ที่กำหนด และทดสอบเงื่อนไขใหม่ ถ้าเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำทันที โดยมีรูปแบบดังนี้

```
While (expression)
{
statement ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ expression นั้นสามารถตรวจสอบค่าคงที่ได้ด้วย ถ้าค่าไม่เท่ากับศูนย์จะทำซ้ำ ถ้าค่าเท่ากับศูนย์จะไม่เท่าซ้ำ

6.4.3 ประโยค do – while

การทำซ้ำประเภทนี้ จะตรวจสอบเงื่อนไขภายหลังการทำสเตตเมนต์แต่ละครั้ง ถ้าหากเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำทันที มีรูปแบบดังนี้

```
do{
statement ;
}
While (condition)
```

6.5 อาร์เรย์ พอยน์เตอร์ และสตรักเจอร์

ในการเขียนโปรแกรมด้วยภาษาซีถ้าหากต้องการใช้งานตัวแปรหลายตัว เราสามารถประกาศชื่อตัวแปรออกมาหลายตัวได้ เช่น x1, x2, x3, …, xn แต่ถ้าหากตัวแปรทุกตัวใช้เก็บข้อมูลประเภทเดียวกัน เราสามารถประกาศเป็นตัวแปรอาร์เรย์ (array) ได้ การประกาศตัวแปรแบบอาร์เรย์สามารถทำได้ดังรูปแบบต่อไปนี้

ประเภทของข้อมูล ชื่ออาร์เรย์[ขนาดอาร์เรย์]

ถ้าเป็นอาร์เรย์แบบสองมิติสามารถประกาศได้ดังนี้

ประเภทของข้อมูล ชื่ออาร์เรย์ [ขนาดอาร์เรย์] [ขนาดอาร์เรย์]

ตัวอย่าง เช่น char x [8];

เป็นการประกาศตัวแปรชื่อ x จำนวน 8 เซล แต่ละเซลจะเก็บข้อมูลประเภทตัวอักขระ การอ้างถึงอาร์เรย์ x แต่ละเซลจะใช้อินเด็กซ์เป็นตัวอ้าง เช่น x [0] เป็นการอ้างถึงเซลแรก นอกจากนี้การประกาศตัวแปรอาร์เรย์สามารถกำหนดค่าข้อมูลเข้าไปในตัวแปรอาร์เรย์เลยได้

ตัวอย่างเช่น unsigned char ab [] = {0xa, 0x9,0x5,0x6};

จะเห็นว่าในการประกาศตัวแปรอาร์เรย์ ab จะไม่ระบุขนาด ของอาร์เรย์ระบบจะจองหน่วยความจำเท่ากับค่าที่กำหนด การประกาศแบบนี้เซลแรก ab[0] จะเก็บค่า 0A ฐานสิบหกขนาดหนึ่งไบต์

ในระบบไมโครคอนโทรลเลอร์ เมื่อมีการประกาศตัวแปรอาร์เรย์ ระบบจะจองหน่วยความจำสำหรับเก็บตัวแปรอาร์เรย์นั้น ตัวแปรประเภทนี้สามารถนำไปประยุกต์ใช้ ในการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบเปิดตารางได้ ในการอ้างถึงข้อมูลแต่ละเซลในอาร์เรย์ สามารถใช้ตัวแปรพอยน์เตอร์ชี้ไปที่ตำแหน่งของอาร์เรย์ได้โดยตรง โดยพอยน์เตอร์จะเป็นตัวแปรที่ใช้เก็บแอดเดรสหรือตำแหน่งหน่วยความจำ ตัวดำเนินการที่ใช้กับพอยน์เตอร์คือ & และ *

จากที่ผ่านมาจะพบว่าตัวแปรประเภทอาร์เรย์นั้นเราอาจมองว่าเป็นกลุ่มของข้อมูลได้ โดยข้อมูลในกลุ่มนั้นจะเป็นข้อมูลประเภทเดียวกัน ถ้าหากต้องการประกาศตัวแปรเป็นกลุ่มของข้อมูลที่ข้อมูลในกลุ่มนั้นเป็นชนิดต่างกัน จะต้องประกาศตัวแปรเป็นแบบโครงสร้าง หรือสตรักเจอร์ (structure) ซึ่งมีรูปแบบดังนี้

```
struct {
        ประเภทของข้อมูล ชื่อตัวแปร;
        ……..
        ……..
} ชื่อโครงสร้าง;
```

โดยกลุ่มของข้อมูลที่ประกาศขึ้นนั้นจะอยู่ในเครื่องหมายปีกกา และเราสามารถอ้างไปที่ข้อมูลใด ๆ ได้ โดยใช้เครื่องหมายจุด (.)

ตัวอย่างเช่น

```
struct {
        unsigned long s;
        unsigned int t;
        unsigned char done;
}state;
```

จะเป็นการประกาศตัวแปรโครงสร้างชื่อ state ซึ่งจะใช้หน่วยความจำทั้งหมด 7 ไบต์ (long 4 + int 2 + char 1) ถ้าหากต้องการใส่ข้อมูลค่า 321 ให้กับตัวแปร t ในโครงสร้างจะทำได้ดังนี้

```
state.t = 321;
```

ถ้าหากมีการประกาศตัวแปรดังต่อไปนี้จะทำให้ผลลัพธ์มีค่าเหมือนกับตัวแปรแบบโครงสร้างในตัวอย่างที่ผ่านมา

```
#define uchar unsigned char
#define uint unsigned                    ---------- ประกาศตัวแปรใหม่
struct stateform{
        unsigned long s;
        uint  t;                         ---------- เรียกใช้ในโครงสร้าง
        uchar done;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
};
```

จะเห็นว่าจะใช้ #define ประกาศตัวแปรประเภทของข้อมูลใหม่ขึ้นมา และถูกเรียกใช้ในตัวแปรโครงสร้าง ถ้าหากต้องการให้ตัวแปรโครงสร้างนี้มีชื่อว่า state จะทำได้ดังนี้

```
struct stateform          state;
```

ข้อมูลแบบโครงสร้าง ชื่อตัวแปร

ถ้าหากต้องการประกาศตัวแปรแบบโครงสร้างหลายตัว สามารถทำได้ในรูปของอาร์เรย์ของโครงสร้าง (array of structures) ดัง

ตัวอย่างเช่น

```
#define uchar unsigned char
#define uint unsigned
    struct stateform {
        unsigned long s;
        uint  t;
        uchar done;
    };
    struct stateform state [20];
```

จะเป็นการประกาศตัวแปรอาร์เรย์ชื่อ state ที่มีทั้งหมด 20 เซล แต่ละเซลจะเป็นตัวแปรแบบโครงสร้างขนาด 7 ไบต์ ทำให้ตัวแปรนี้ใช้หน่วยความจำทั้งหมด 140 ไบต์ (20 x 7)

6.5 การเขียนโปรแกรมจัดการหน่วยความจำ

การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 จะใช้การจัดหน่วยความจำแบบ Harvard โดยแยกหน่วยความจำออกเป็นส่วนของหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล สำหรับหน่วยความจำข้อมูลนั้นยังถูกแบ่งออกเป็นหน่วยความจำภายในตัว MCS-51 เอง และหน่วยความจำภายนอก ในการขยายพอร์ตเพิ่มเพื่อใช้กับตัว MCS-51 ตำแหน่งของพอร์ตจะเป็นตำแหน่งของหน่วยความจำภายนอกนี้ด้วย

สำหรับหน่วยความจำภายในของ MCS-51 ยังถูกแบ่งออกเป็น หน่วยความจำ 128 ไบต์แรก (ตำแหน่ง 00H – 7FH) ซึ่งสามารถเข้าถึงข้อมูลได้แบบ direct และ indirect ส่วนตำแหน่ง 80H ถึง FFH จะเป็นรีจิสเตอร์ฟังก์ชันพิเศษ หรือ SFR (Special Function Register) ซึ่งจะใช้การเข้าถึงข้อมูลแบบ direct

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการใช้ภาษาซี MCS-51 จะต้องมีการจัดวางหน่วยความจำให้ตัวแปรต่าง ๆ ที่ถูกประกาศขึ้นนั้นสามารถอยู่ในหน่วยความจำที่เหมาะสมได้ โดยการจัดหน่วยความจำของตัวคอมไพเลอร์มีรูปแบบดังนี้

TINY เป็นการจัดหน่วยความจำให้ตัวคอมไพเลอร์มองว่าไม่มีหน่วยความจำแรมภายนอก ซึ่งจะใช้กับ MCS-51 แบบที่เป็นชิงเกิลชิป เช่น เบอร์ 89C8051

SMILL หน่วยความจำแบบนี้คอมไพเลอร์จะมองว่าแรมและรอมอยู่ซ้อนกันภายในพื้นที่ 64 กิโลไบต์ ตัวแปรต่าง ๆ ที่ถูกประกาศใช้งาน หรือประกาศเป็นแบบโลเคิลจะถูกกำหนดอยู่ในหน่วยความจำข้อมูลภายใน (internal data memory) ขนาดของหน่วยความจำสแตติกจะถูกกำหนดตามที่ใช้งานจริง รูปแบบหน่วยความจำโหมดนี้จะเป็นค่าที่โปรแกรมคอมไพเลอร์ได้กำหนดไว้แล้ว

COMPACT หน่วยความจำแบบนี้จะมีลักษณะเดียวกับหน่วยความจำแบบ SMALL แต่ตัวแปรที่ถูกประกาศใช้งาน หรือประกาศแบบโลเคิลจะถูกกำหนดอยู่ในความจำภายนอก ทำให้ตัวแปรสามารถมีได้มากกว่าแบบ SMALL แต่การทำงานของโปรแกรมที่ต้องประมวลผลกับตัวแปรจะทำงานได้ช้ากว่าแบบ SMALL เพราะแบบ SMALL ตัวแปรจะอยู่ในแรมภายในชิป

LARGE หน่วยความจำแบบนี้จะมีขนาดใหญ่ตัวแปรทั้งหมดจะถูกกำหนดให้ อยู่ในหน่วยความจำข้อมูลภายนอก

การเขียนโปรแกรมด้วยภาษาซีให้กับไมโครคอนโทรลเลอร์ด้วย C51 นี้สามารถประกาศตัวแปรต่าง ๆ แบบเจาะจงได้ โดยใช้แมโครที่ถูกกำหนดไว้ในไฟล์ absacc.h ดังต่อไปนี้

CODE	แทนพื้นที่หน่วยความจำโปรแกรมภายนอก
DATA	แทนพื้นที่หน่วยความจำข้อมูลภายใน 128 ไบต์แรก (ตำแหน่ง 00H ถึง 7FH)
IDATA	แทนเนื้อที่หน่วยความจำข้อมูลภายใน 256 ไบต์
BDATA	แทนตำแหน่งของหน่วยความจำระดับบิต 128 บิต (20H ถึง 2HF) ของหน่วยความจำภายใน
XDATA	แทนตำแหน่งหน่วยความจำภายนอก
PDATA	แทนตำแหน่งหน่วยความจำภายนอก 256 ไบต์แรก

ในการกำหนดชื่อตัวแปรต่าง ๆ เพื่อแทนตำแหน่งหน่วยความจำมีรูปแบบดังนี้

ชนิดของตัวแปร ชื่อแมโคร ตัวแปร

ตัวอย่างเช่น

```
char CODE text [ ] = "COMPUTER"
unsigned char XDATA AB [100]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned int IDATA x, y, z
```

การกำหนดบรรทัดแรกจะทำให้คำว่า COMPUTER ถูกเก็บอยู่ในหน่วยความจำโปรแกรม ภายนอกบรรทัดที่สองจะทำให้อาร์เรย์ AB ถูกประกาศเอาไว้ในหน่วยความจำข้อมูลภายนอก

ในการเขียนโปรแกรมให้ไมโครคอนโทรลเลอร์ ติดต่อกับพอร์ตจะต้องทำการโอนย้ายข้อมูลกับพอร์ตตำแหน่งแอดเดรสนั้น ๆ ซึ่งการอ้างแอดเดรสของพอร์ตสามารถใช้แมโครได้เช่นกัน ทำให้สามารถแทนพอร์ตด้วยตัวแปรต่าง ๆ ได้ โดยแมโครที่ใช้จะเก็บอยู่ในในไฟล์ absacc.h โดยมีชื่อดังต่อไปนี้

CBYTE , CWORD แทนไบต์หรือเวิร์ดของหน่วยความจำโปรแกรม
DBYTE , DWORD แทนไบต์หรือเวิร์ดของหน่วยความจำข้อมูลภายใน
PBYTE , PWORD แทนไบต์หรือเวิร์ดของหน่วยความจำภายนอก 256 ตำแหน่งแรก
XBYTE , XWORD แทนไบต์หรือเวิร์ดของหน่วยความจำภายนอกทั้งหมด

ในการกำหนดตัวแปรให้แทนตำแหน่งต่าง ๆ จะต้องใช้ร่วมกับไดเร็กทีฟ #define ดังตัวอย่างเช่น

```
#define port8 XBYTE [0x8000]
```

จะเป็นการแทนตัวแปร port8 ด้วยความจำภายนอกตำแหน่ง 8000H ถ้าหากต้องการส่งค่า FF ไปยังหน่วยความจำตำแหน่ง 8000H สามารถทำได้ดังนี้

```
Port8 = 0x0FF;
```

ถ้าหากต้องการอ่านข้อมูลจากตำแหน่ง 8000H มาเก็บในตัวแปร X สามารถทำได้ดังนี้

```
Char x;
X = port8;
```

การเข้าถึงข้อมูลระดับบิตและรีจิสเตอร์พิเศษ

ในโปรแกรม C51 ถ้ามีการประกาศตัวแปรและกำหนดให้เป็นหน่วยความจำแบบ BDATA จะเป็นการใช้งานในแอดเดรสหน่วยความจำที่สามารถเข้าถึงข้อมูลในระดับบิตได้ นอกจากนี้เราสามารถใช้คำสั่ง sbit ของ C51 กำหนดตัวแปรแบบบิตขึ้นมาได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น

```
Int BDATA ibase;                    / * Bit – addressable int * /
char DBATA bary [4];                / * Bit – addressable array * /
sbit mybit0 = ibasa^0;              / * bit 0 of ibase * /
sbit mybit15 = ibase^15;            / * bit 15 of ibase * /
sbit Ary07 = bary [0]^7;            / * bit 7 of bary [0] * /
```

บรรทัดแรกจะเป็นการประกาศตัวแปรชื่อ ibase สำหรับเก็บตัวเลขจำนวนเต็มลงในหน่วยความจำพื้นที่ที่สามารถเข้าถึงข้อมูลในระดับบิตได้ บรรทัดต่อมาประกาศตัวแปรอาร์เรย์จำนวน 4 ไบต์ลงในหน่วยความจำระดับบิต ในบรรทัดที่สามจะใช้คำสั่ง sbit กำหนดตัวแปรขึ้นมาใหม่ แทนบิตต่ำสุด โดยใช้สัญลักษณ์ carat (' ^') ซึ่งจะทำให้ตัวแปรชื่อ mybit0 แทนบิตที่ 0 ของหน่วยความจำชื่อ ibase บรรทัดที่สี่จะทำให้ตัวแปรชื่อ mybit15 แทนบิตสูงสุดของตัวแปร ibase และบรรทัดสุดท้ายจะให้ตัวแปรชื่อ Ary07 แทนบิตที่ 7 ของ bary[0] สำหรับขนาดของบิตของตัวแปรจะขึ้นกับประเภทของข้อมูลที่กำหนด เช่น ถ้าเป็นข้อมูลแบบ char และ unsigned char ข้อมูลบิตจะอยู่ในช่วง 0 ถึง 7 บิต ถ้าเป็นแบบ int, unsigned int, short และ unsigned short จะอยู่ในช่วง 0 ถึง 15 บิต

ถ้าต้องการตัวแปรชื่อ OUT แทนบิตสูงสุดของพอร์ต P1 ของ MCS-51 สามารถทำได้ดังนี้

```
sbit OUT = P1^7;
```

พื้นที่หน่วยความจำภายในของ MCS-51 ส่วนหนึ่งจะเป็นพื้นที่ของเรจิสเตอร์ฟังก์ชันพิเศษ (SFR) เช่น รีจิสเตอร์ควบคุมไทเมอร์ รีจิสเตอร์ตัวนับ พอร์ตอนุกรม พอร์ตอินพุตเอาต์พุตต่าง ๆ ซึ่งอยู่ในหน่วยความจำภายในตั้งแต่ตำแหน่ง 0x80 ถึง 0xFF ถ้าหากเราต้องการกำหนดชื่อใหม่แทนสัญลักษณ์เหล่านั้นสามารถใช้คำสั่ง sfr ในการประกาศได้ โดยใช้เครื่องหมายเท่ากับ (=) ระบุค่าแอดเดรสให้กับตัวแปรนั้น เช่น

```
Sfr PO  =  0x80;                    / * Port_0 address 80H * /
Sfr P1  =  0x90;                    / * Port_1 address 90H * /
Sfr P2  =  0xA0;                    / * Port_2 address OAH * /
sfr P3  =  0xB0;                    / * Port_3 address BOH * /
```

ถ้าหากต้องการกำหนดชื่อแทนรีจิสเตอร์ฟังชันพิเศษที่มีขนาด 16 บิต จะต้องใช้คำสั่ง sfr16 และระบุแอดเดรสไบต์ต่ำให้กับตัวแปรนั้น ตัวอย่างเช่น

```
sfr16   T2  =  0xcc;                /* Timer 2 : T2L 0CCH , T2H 0CDH  */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นการให้ T2 แทนไทเมอร์ 2 ซึ่งรีจิสเตอร์นี้จะมีขนาด 2 ไบต์ ไบต์ต่ำหรือ T2L จะอยู่ในแอดเดรส CCH และไบต์สูง T2H จะอยู่ในแอดเดรส 0CDH

สำหรับในการเขียนโปรแกรมด้วย C51 นั้นจะเห็นว่ารีจิสเตอร์บางตัว ตำแหน่งบิตบางตำแหน่ง เราสามารถอ้างชื่อขึ้นมาได้เลย เนื่องจากในไฟล์ res51.h ที่ได้ include ขึ้นมาเป็นไฟล์ที่กำหนดรีจิสเตอร์ต่าง ๆ เอาไว้แล้ว เช่น P0, P1, P2 และ P3 สำหรับตำแหน่งบิตของรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลระดับบิตได้ เช่น TF1, TR1, TF0, TRO, IE1, IT1, IE0, และ ITO ซึ่งบิตที่อยู่ในรีจิสเตอร์ TCON ก็กำหนดไว้แล้วเช่นกันโดยในไฟล์ reg51.h ส่วนหนึ่งได้กำหนดไว้ดังนี้

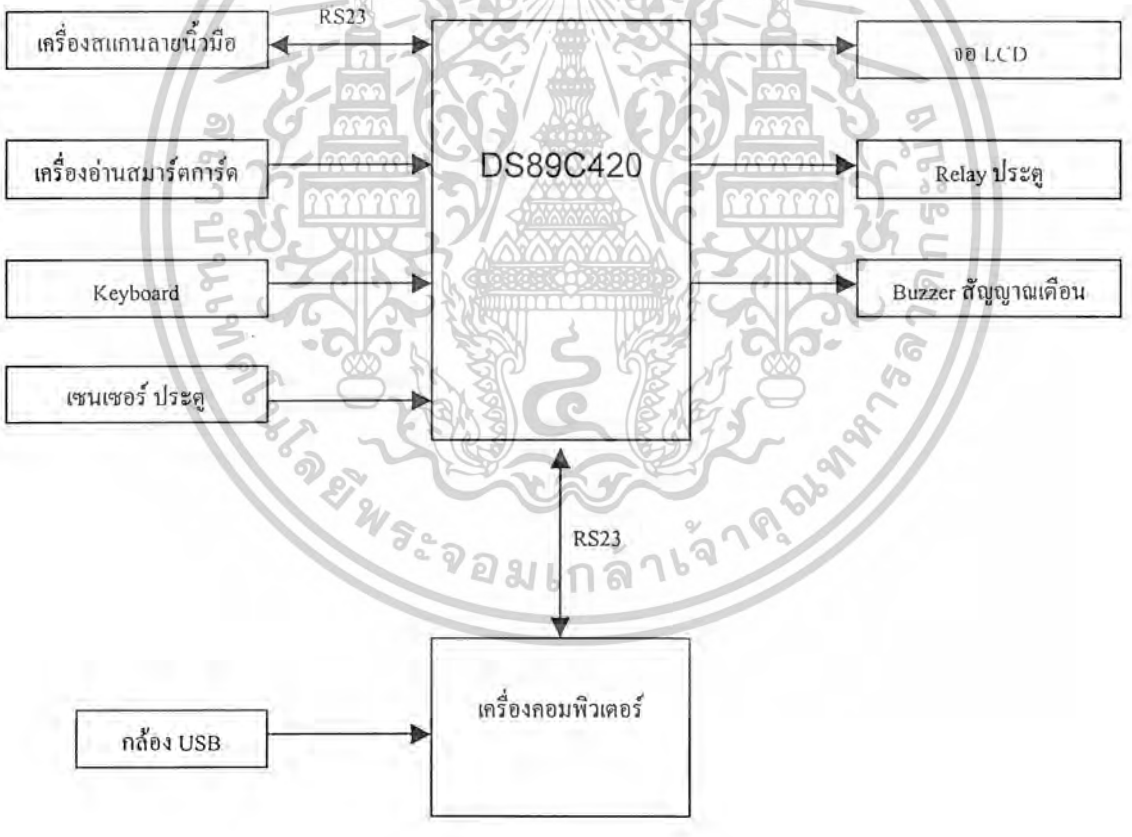
```
/* BYTE register */
sfr P0  =  0x80;
sfr P1  =  0x90;
sfr P2  =  0xA0;
sfr P3  =  0xB0;
/*BIT register * /
/*TCON */
sbit TF1  =  0x8F;
sbit TR1  =  0x8E;
sbit TF0  =  0x8D;
sbit TR0  =  0x8C;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7 การออกแบบและหลักการทำงานของวงจร

7.1 องค์ประกอบของระบบโดยรวม

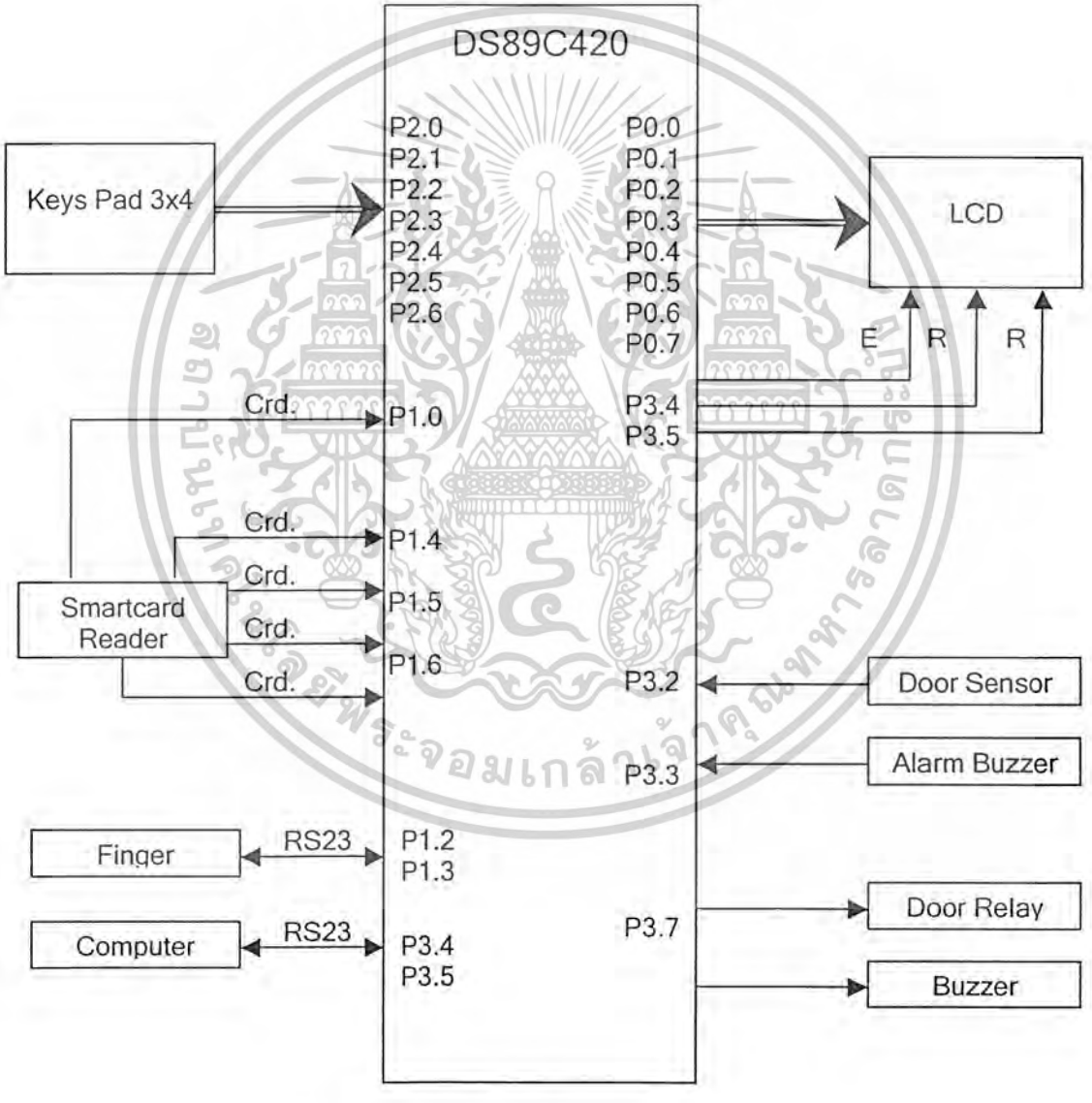
การประยุกต์ใช้งานเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ด ในโครงงานนี้ได้ถูกออกแบบมาเพื่อใช้ในงานด้านการรักษาความปลอดภัยและการอำนวยความสะดวก นั่นคือ ในส่วนของการรักษาความปลอดภัยนั้น จะเป็นการนำเครื่องตรวจสอบลายนิ้วมือร่วมกับสมาร์ตการ์ดใช้ในการควบคุมการเข้าออกสถานที่ต่างๆ เช่นห้องพักส่วนตัว หรือ สถานที่ต้องการไม่ให้บุคคลแปลกปลอมหลบหลีกเข้าไปได้ เป็นต้น โดยจะมีการตรวจสอบลายนิ้วมือทุกครั้งขณะที่มีการเข้าออกผ่านประตูและสามารถบันทึกเวลาในขณะนั้นไว้ด้วย จากที่กล่าวมาข้างต้นทั้งหมดนี้คือระบบโดยรวมที่ได้ออกแบบไว้ สามารถเขียนเป็นภาพโครงสร้างโดยรวมอย่างง่ายๆได้ดังรูปที่ 7.1



รูปที่ 7.1 โครงสร้างของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากข้างต้น สิ่งที่สำคัญที่สุดสำหรับการสร้างระบบนี้มี 3 ส่วนคือ ส่วนของเครื่องตรวจสอบลายนิ้วมือใช้โมดุล BKS-1700 ที่ทำหน้าที่ในการตรวจสอบถึงความเป็นสมาชิกในฐานข้อมูลหรือไม่ ส่วนที่สองคือส่วนของสมาร์ตการ์ด และส่วนที่สามคือ บอร์ดควบคุมที่คอยทำหน้าที่ติดต่อระหว่างเครื่องตรวจสอบลายนิ้วมือ กับคอมพิวเตอร์ ผ่านทางพอร์ท RS232 เพื่อแสดงผลผ่านหน้าจอคอมพิวเตอร์ ตลอดจนรับคำสั่งที่ผู้ใช้ป้อนเข้ามาทางคีย์แพด โดยหัวใจสำคัญที่ควบคุมการทำงานคือ IC DS89C420 ซึ่งเป็นไอซีแบบไฮสปีด และมีพอร์ทอนุกรม 2พอร์ท



รูปที่ 7.2 ลักษณะการต่อทางฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7.2 หลักการทำงานโครงงาน

การใช้งานจะประกอบไปด้วย 3 ขั้นตอนสำคัญคือ การลงทะเบียนผู้ใช้ การเข้าใช้งานระบบ และการลบข้อมูลผู้ใช้งานออกจากระบบ และสามารถเลือกระดับรักษาความปลอดภัยได้ 3 ระดับ

คุณสมบัติของเครื่อง

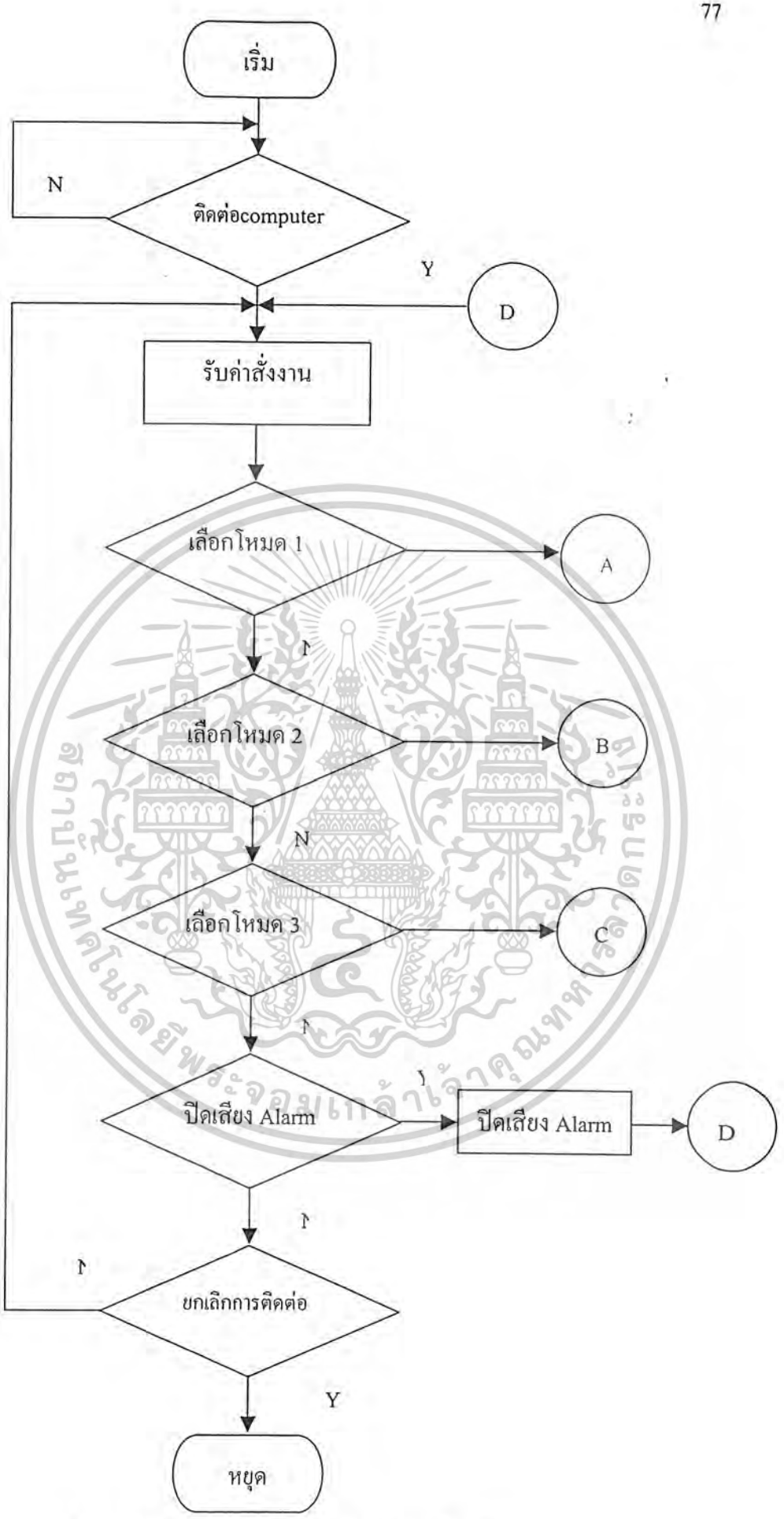
1. รองรับ 500 คน คนละ 2 ลายนิ้ว
2. เลือกระดับความปลอดภัย 3 ระดับ
   - โหมด1 ใช้ลายนิ้วหรือสมาร์ตการ์ดคู่กับรหัสผ่าน อย่างใดอย่างหนึ่ง
   - โหมด2 ใช้ลายนิ้วและสมาร์ตการ์ดคู่กัน
   - โหมด3 ใช้ลายนิ้ว, สมาร์ตการ์ด และรหัสผ่าน
3. เก็บฐานข้อมูลและรหัสผ่านในคอมพิวเตอร์
4. บันทึกข้อมูลผู้ใช้ ลงในคอมพิวเตอร์และเรียกดูข้อมูลย้อนหลังได้
   - ID วัน เวลา
5. จะทำการถ่ายภาพ คนแสกนและแสดงเพื่อให้ผู้ดูแลเปรียบเทียบ
6. การเพิ่มหรือลบ ฐานข้อมูลต้องเป็นผู้ดูแลระบบเท่านั้น
   - มี User และ Password เพื่อแก้ไข
7. ส่งเสียงสัญญาณเตือนภัยและถ่ายภาพพร้อมบันทึก ภาพ และ วัน เวลา เมื่อมีการบุกรุก

โฟลชาร์ตแสดงการทำงานของเครื่อง

FG. = Finger Print

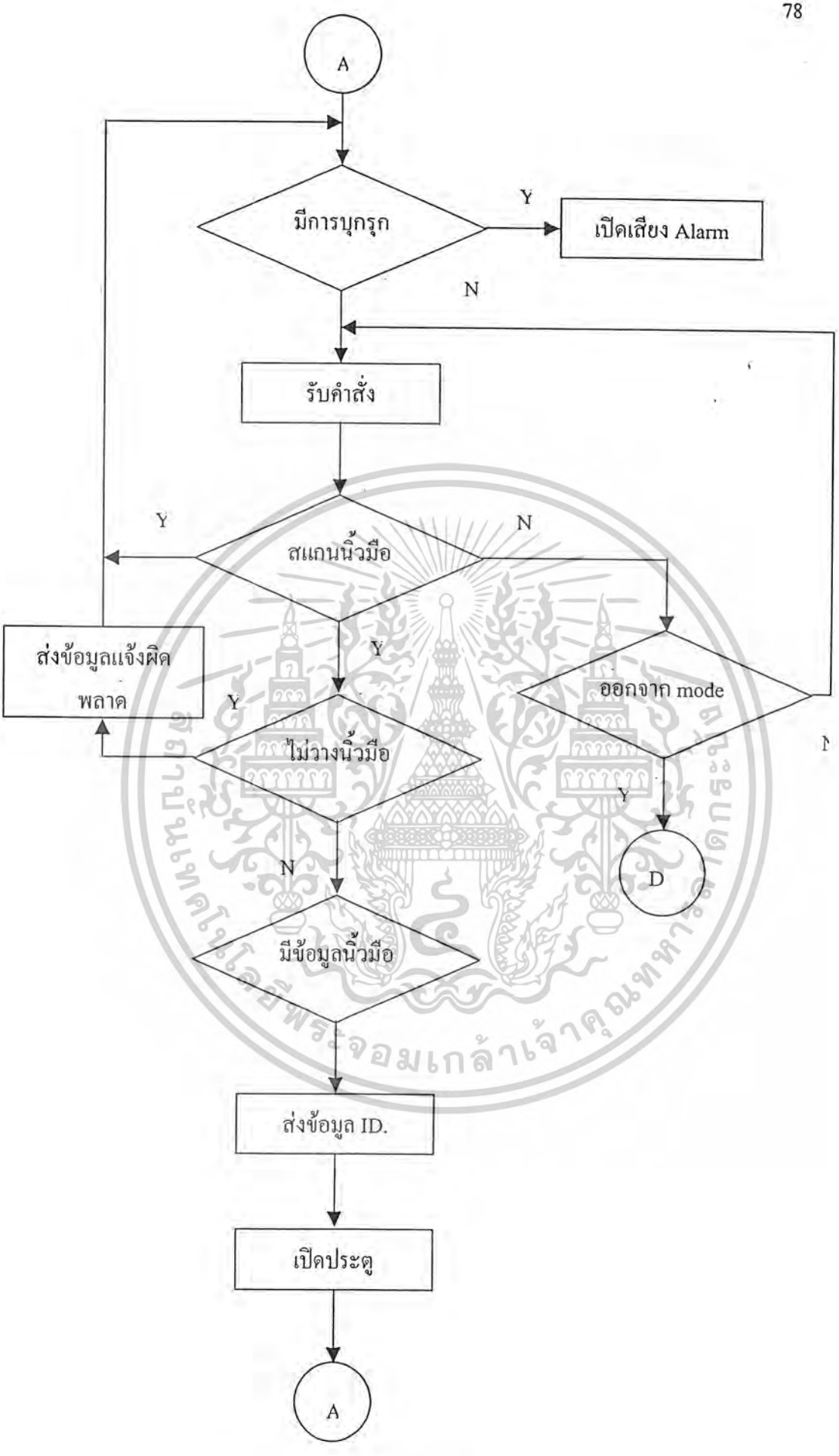
SM. = Smart Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



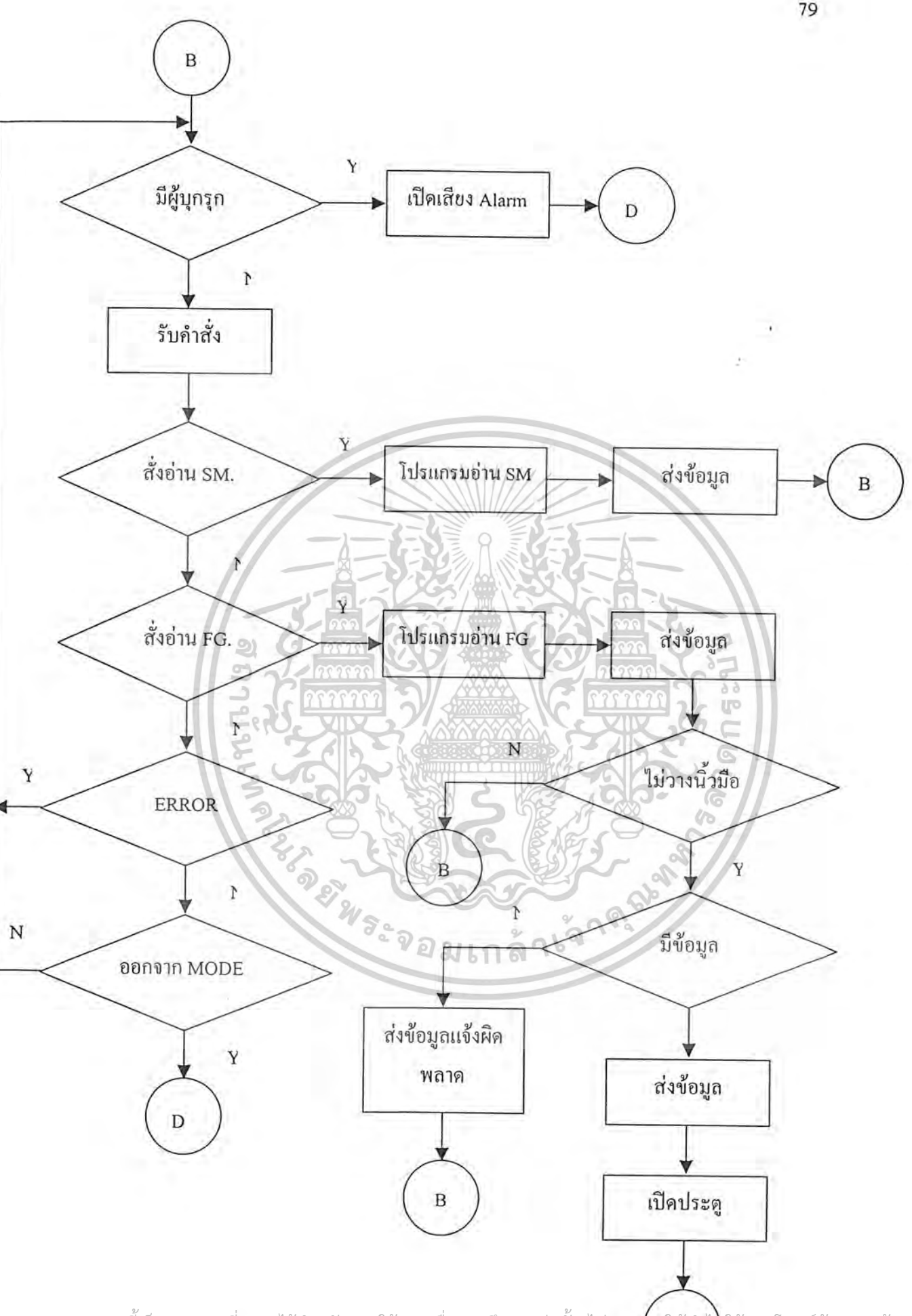
รูปที่ 7.3 โฟลชาร์ทการทำงานของเครื่องโหมดเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



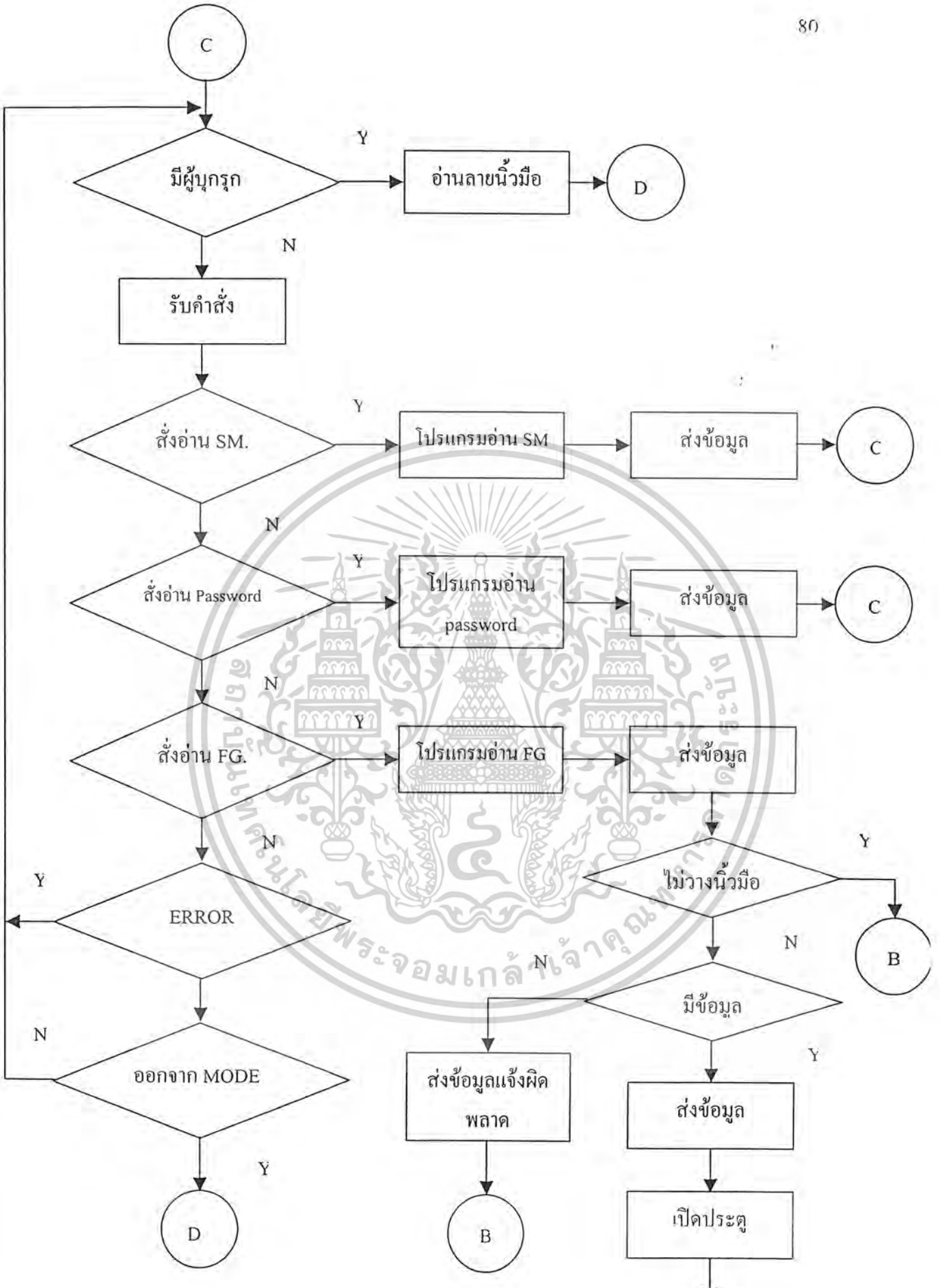
รูปที่ 7.4 โฟลชาร์ทการทำงานของเครื่องโหมดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น การทำซ้ำ ดัดแปลง หรือเผยแพร่ต่อสาธารณชน ไม่ว่าในรูปแบบใดๆ ของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.5 โฟลชาร์ทการทำงานของเครื่องโหมดที่ 2



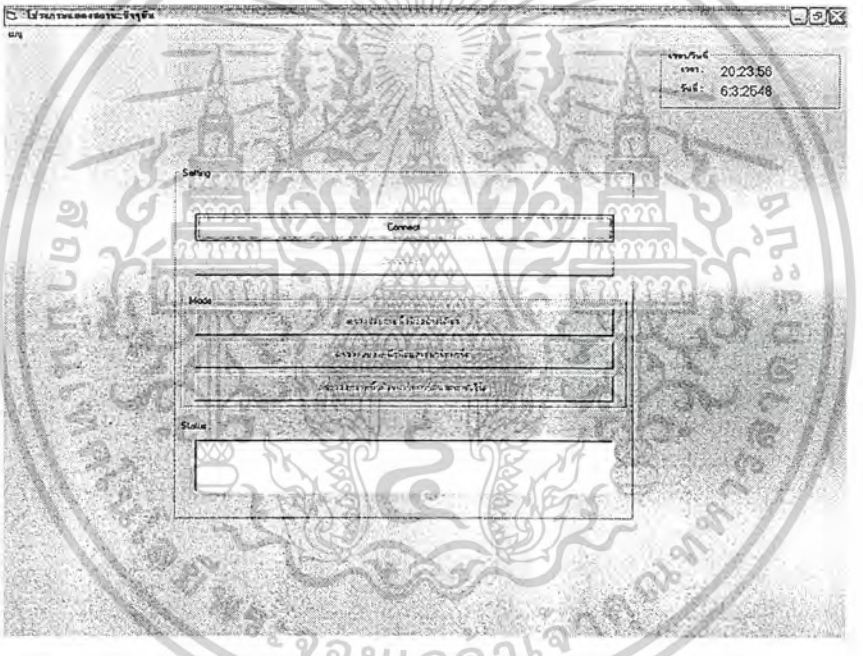
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 7.6 โฟลชาร์ทการทำงานของเครื่องโหมดที่ 3

บทที่ 8 การทดลองและผลการทดลอง

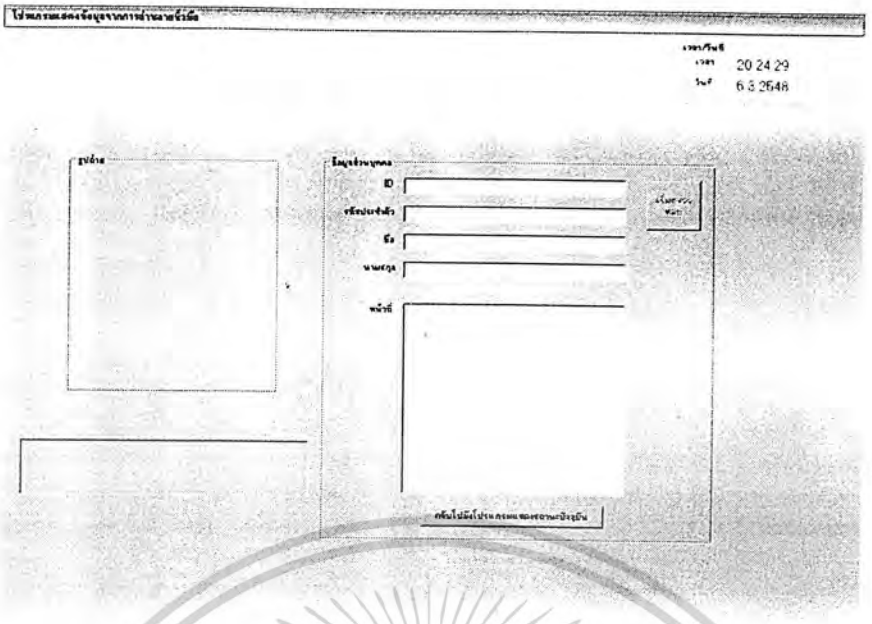
สำหรับวิธีการทดสอบและใช้งาน สามารถทำได้โดยหลังจากลงอุปกรณ์บนแผงวงจรครบถ้วน และประกอบเข้ากับเครื่องตรวจสอบลายนิ้วมือ และสมาร์ตการ์ด ตรวจดูความเรียบร้อยต่างๆของสายวงจร และการบัดกรีอีกครั้งว่าถูกต้องหรือไม่ หลังจากนั้นจึงทำการจ่ายไฟเลี้ยงให้แก่วงจร จะเห็นว่า LED1 ติด และจอแอลซีดีแสดงข้อความ Press Finger แสดงว่าวงจรทั้งหมดพร้อมรับคำสั่งเพื่อทำงานต่อได้

รูปแสดงโปรแกรมที่ โหมดต่างๆ

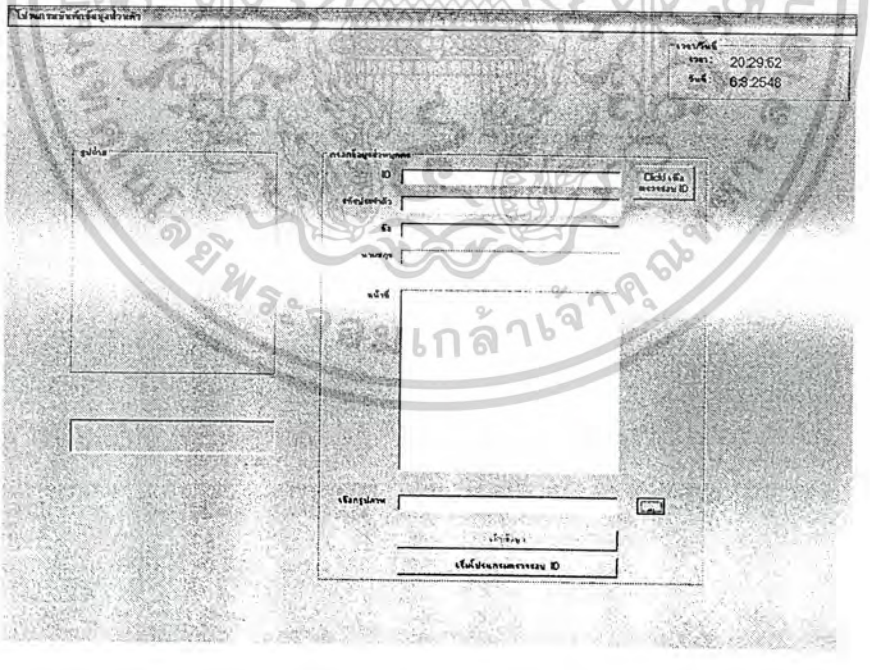


รูปที่ 8.1 ตัวอย่างหน้าจอแสดงสถานะเริ่มต้นเพื่อที่จะเลือกโหมดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8.2 ตัวอย่างหน้าจอแสดงผลและเป็นพื้นฐานการแสดงผลในทุกๆโหมดการทำงาน



รูปที่ 8.3 ตัวอย่างหน้าจอแสดงการเพิ่ม ID ในฐานข้อมูลเครื่องแสกนลายนิ้วมือและสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานที่ 1

เป็นโหมดการทดลองโดยการแสกนลายนิ้วมือเพียงอย่างเดียวเท่านั้นโดยจะทำการทดลองทั้งหมด 100 ครั้ง แล้วหาความผิดพลาดว่ามีความผิดพลาดเท่าไร

1. การทดลองวัดความผิดพลาดการรับส่งข้อมูลจากเครื่องอ่านลายนิ้วมือมายังโปรแกรมฐานข้อมูล
2. ความเร็วในการรับข้อมูล 115200 บิต/วินาที ผ่านพอร์ตคอม 1
3. ทำการทดลองโดยการใช้ลายนิ้วมือเพียงอย่างเดียวเท่านั้น

จำนวนผู้ทดลอง	หมายเลขประจำตัว	จำนวนครั้งที่ถูกต้อง	จำนวนครั้งที่ผิดพลาด	เปอร์เซ็นต์ความผิดพลาด
1	45015209	100	0	0%
2	45015190	100	0	0%
3*	45015191	99	1	1%
4	45015207	100	0	0%
5	45015206	100	0	0%
	รวม	499	1	0.01%

หมายเหตุ 3* ความผิดพลาดเกิดจากการวางนิ้วที่ไม่ถูกต้องเท่าที่ควร

ตารางที่ 8.1 ผลการทดลองรับค่าจากเครื่องอ่านลายนิ้วมือมายังโปรแกรมฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานที่ 2

เป็นโหมดการทำงานระหว่างสมาร์ตการ์ดและเครื่องสแกนลายนิ้วมือตามลำดับโดยจะเป็นการป้องกันที่สูงอีกขั้นหนึ่ง

1. การทดลองวัดความผิดพลาดการรับส่งข้อมูลจากเครื่องอ่านลายนิ้วมือและสมาร์ตการ์ดมายังโปรแกรมฐานข้อมูล
2. ความเร็วในการรับข้อมูล 115200 บิต/วินาที ผ่านพอร์ตคอม 1
3. มีการเปรียบเทียบกันระหว่างรหัสของลายนิ้วมือและรหัสของสมาร์ตการ์ด

จำนวนผู้ทดลอง	หมายเลขประจำตัว	หมายเลขบัตรสมาร์ตการ์ด	จำนวนครั้งที่ถูกต้อง	จำนวนครั้งที่ผิดพลาด	เปอร์เซ็นต์ความผิดพลาด
1	45015209	1281124001	100	0	0%
2	45015190	1282173972	100	0	0%
3	45015191	1279070253	100	0	0%
4	45015207	1282173989	100	0	0%
5	45015206	1289043389	100	0	0%
	รวม		100	0	0%

หมายเหตุ การทดลองครั้งนี้ไม่พบข้อผิดพลาดเพราะมีการวางนิ้วมือที่ถูกต้อง

ตารางที่ 8.3 ผลการทดลองรับค่าจากเครื่องอ่านลายนิ้วมือและสมาร์ตการ์ดมายังโปรแกรมฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมดการทำงานที่ 3

เป็นโหมดการทำงานระหว่างสมาร์ตการ์ดและเครื่องสแกนลายนิ้วมือพร้อมด้วยรหัสป้องกันอีกขั้นตอนหนึ่งเพื่อการป้องที่มีประสิทธิภาพมากยิ่งขึ้น

1. ทำการทดสอบหาค่าความผิดพลาด
2. มีการเปรียบเทียบกับถึง 3 ครั้งเป็นการป้องกันที่ปลอดภัยมากที่สุดในระบบนี้

จำนวนผู้ทดลอง	หมายเลขประจำตัว	หมายเลขบัตรสมาร์ตการ์ด	รหัสผ่าน	จำนวนครั้งที่ถูกต้อง	จำนวนครั้งที่ผิดพลาด	เปอร์เซ็นต์ความผิดพลาด
1	45015209	1281124001	261224	100	0	0%
2	45015190	1282173972	120525	100	0	0%
3*	45015191	1279070253	021024	99	1	0%
4	45015207	1282173989	200824	100	0	0%
5*	45015206	1289043389	170125	99	1	0%
	รวม			498	2	0.02%

หมายเหตุ 3* และ 5* เป็นความผิดพลาดแบบเดิมที่การว่างนิ้วมือไม่ถูกลักษณะทำให้เกิดข้อผิดพลาดได้อีก

ตารางที่ 8.3 ผลการทดลองรับค่าจากเครื่องอ่านลายนิ้วมือและสมาร์ตการ์ดมายังโปรแกรมฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 9 วิเคราะห์และสรุปผลการทดลอง

จากผลการทดลอง จะเห็นได้ว่า เปอร์เซ็นต์การผิดพลาดของข้อมูลที่เครื่องอ่านบัตรสมาร์ตการ์ดและเครื่องอ่านลายนิ้วมือได้และส่งข้อมูลให้คอมพิวเตอร์นั้น มีความผิดพลาดน้อยมาก และสำหรับส่วนที่ผิดพลาดนี้อาจเนื่องมาจากการวางนิ้วที่ไม่เหมาะสม เช่น วางเบาเกินไป วางในลักษณะเอียงหรือแบบรีบร้อนจนเกินไป ซึ่งความผิดพลาดส่วนนี้สามารถแก้ไขได้โดยผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1) คู่มือเรียน Visual Basic 6 ฉันทวุฒิ พืชผล พิชัย สันติกุลานนท์ พิมพ์ครั้งที่ 5 บริษัทโปรวิชั่น จำกัด

2) Visual Basic 6 ฉบับโปรแกรมเมอร์ กิตติ ภักดีวัฒนะกุล จำลอง ครูอุตสาหะ พิมพ์ครั้งที่ 11 สำนักพิมพ์ เคพีที คอมพ์ แอนด์ คอนซัลท์

3) เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม โดย อรรถพล บุญยะโภคา วรพจน์ กรแก้ววัฒนกุล ชัยวัฒน์ ลิ้มพรจิตร วิไล Innovation Experiment Co.,Ltd.

4) เซมิคอนดัคเตอร์ฟิสิกส์ โครงงานเครื่องตรวจสอบลายนิ้วมือ สำหรับระบบรักษาความปลอดภัย ฉบับที่ 244 หน้าที่ 149 โดย อารัมภีย์ จันทร์ไย โสรัตน์ อุณหะวรากร บริษัท ซีเอ็ดยูเคชั่น จำกัด

5) การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี ผศ. ธีรวัฒน์ ประกอบผล พิมพ์ครั้งที่ 5 สำนักพิมพ์ ส.ท.ท.

6) เทคโนโลยีสมาร์ตการ์ด เลศ แซ่ตั้ง บริษัท ซีเอ็ดยูเคชั่น จำกัด(หมาชน)

7) http://mastertech.co.th/news-fingerprint.html

8) http://www.hosdoc.com/staff/Contents/07-Fingerprint/FP-content-main.htm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

โปรแกรม MAIN.C

```
#include "LCD.h"
unsigned char code idn_code[] ={0x06,0x02,0x00,0x00,0x00,0x0a,0x01,0x19,0x2c,0x03};
unsigned char code add_code[] ={0x06,0x02,0x00,0x00,0x00,0x0e,0x01,0x16};
unsigned char code del_code[] ={0x06,0x02,0x00,0x00,0x00,0x0e,0x01,0x33};
unsigned char fg_code[14];
unsigned char ID_from_PC[4];
unsigned char ID_code[4];
unsigned char key_data,crd_buff;
sbit crd_in  = P1^0;
sbit crd_vdd = P1^4;
sbit crd_rst = P1^5;
sbit crd_clk = P1^6;
sbit crd_io      = P1^7;
sbit door        = P3^7;
sbit busser      = P1^1;
sbit sensor  = P3^2;
sbit Alarm   = P3^3;
sfr  keyport = 0xA0;

//---------- RS 232 ----------
void sbytep (unsigned char x)
{
  while (!TI);
  TI = 0;
  SBUF0 = x;
}

void sbytep1 (unsigned char x)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while (!TI1);
    TI1 = 0;
    SBUF1 = x;
}

unsigned char rbytep (void)
{
    while (!RI);
    RI = 0;
    return (SBUF0);
}

unsigned char rbytep1 (void)
{
    while (!RI1);
    RI1 = 0;
    return (SBUF1);
}

// -------- Identify Finger Print ---------
void idn_fg (void)
{
        unsigned char i;
        for (i=0;i<10;i++)
         sbytep1(idn_code[i]);
}

void idn_read_fg (void)
{
        unsigned char i;
        LCD_WCMD(0x01);
        LCD_STRING("Put Your Finger!");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        for (i=1;i<=8;i++)
    fg_code[i]=rbytep1();
        if (fg_code[8]==0x00)
        {
                for (i=9;i<=14;i++)
                fg_code[i]=rbytep1();
        }
        else
        {
                for (i=9;i<=10;i++)
                fg_code[i]=rbytep1();
        }
}

void beep (unsigned int time)
{
        busser=0;
        dmsec(time);
        busser=1;
}

void open_door(void)
{
        door=1;
        busser=0;
        sensor=1;
        dmsec(20000);
        busser=1;
        dmsec(1000);
        while(!sensor);
        {
                dmsec(5000);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                door=1;
        }
        door=0;
}

void idn_send_PC (void)
{
        unsigned char i;
        if (fg_code[8]==0x00)
        {
         sbytep('F');
         for (i=9;i<=12;i++)
         {
                sbytep(fg_code[i]);
         }
         LCD_WCMD(0x01);
         LCD_STRING(" WellCome +^_^+ ");
         open_door();
         LCD_WCMD(0x01);
         sbytep('Y');
        }
        else
        {
         if (fg_code[8]==0x01)
         {
          sbytep('F');
          sbytep('F');
                LCD_WCMD(0x01);
                LCD_STRING("No ID. In System");
                beep(5000);
                dmsec(5000);
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
        {
         sbytep('N');
         sbytep('F');
        }
       }
}
//---------------------------------

//------- Add and Dellete ID --------
void read_PC (void)
{
       unsigned char i,j;
       for (i=0;i<=4;i++)
        ID_from_PC[i]=rbytep();
       if (ID_from_PC[4]=='Y')
       {
            for (j=0;j<=3;j++)
            {
                 ID_code[j]=ID_from_PC[j];
                 switch (ID_code[j])
                 {
                        case '0':ID_code[j]=0x00;
                                 break;
                       case '1':ID_code[j]=0x01;
                                 break;
                       case '2':ID_code[j]=0x02;
                                 break;
                       case '3':ID_code[j]=0x03;
                                 break;
                       case '4':ID_code[j]=0x04;
                                 break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                case '5':ID_code[j]=0x05;
                        break;
                case '6':ID_code[j]=0x06;
                        break;
                case '7':ID_code[j]=0x07;
                        break:
                case '8':ID_code[j]=0x08;
                        break;
                case '9':ID_code[j]=0x09;
                        break;
                }
            }
        }
}

//-----------------------------

//----- Send Code To Add ID ------
void add_fg (void)
{
        unsigned char i,j;
        unsigned int sum;
        sum=0;
        for (i=0;i<=7;i++)
         sbytep1(add_code[i]);
        for (j=0;j<=3;j++)
         sbytep1(ID_code[j]);
        for (i=0;i<=7;i++)
         sum += add_code[i];
        for (j=0;j<=3;j++)
         sum += ID_code[j];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sbytep1(sum);
        sbytep1(0x03);
}
//---------------------------------

//----- Send Code To Del ID -------
void del_fg (void)
{
        unsigned char i,j;
        unsigned int sum;
        sum=0;
        for (i=0;i<=7;i++)
         sbytep1(del_code[i]);
        for (j=0;j<=3;j++)
         sbytep1(ID_code[j]);
        for (i=0;i<=7;i++)
         sum += del_code[i];
        for (j=0;j<=3;j++)
         sum += ID_code[j];
        sbytep1(sum);
        sbytep1(0x03);
}
//---------------------------------

//----- Read From Finger Scanner ---
void read_fg (void)
{
        unsigned char i;
        for (i=1;i<=10;i++)
         fg_code[i]=rbytep1();
}
//---------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//------- Sent Status To PC --------
void add_send_PC (void)
{
 read_fg();
 if (fg_code[8]==0x00)
        {
         sbytep('A');
         sbytep('Y');
         LCD_WCMD(0x01);
         LCD_STRING("Add Fingel OK !!");
         beep(500);
         dmsec(9500);
        }
        else
        {
         sbytep('N');
         sbytep('F');
         LCD_WCMD(0x01);
         LCD_STRING("-- No  Fingel --");
         beep(5000);
         dmsec(5000);
        }
}

void del_send_PC (void)
{
 read_fg();
 if (fg_code[8]==0x00)
        {
         sbytep('D');
         sbytep('Y');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCD_WCMD(0x01);
        LCD_STRING("Delete Fingel ID");
        beep(500);
        dmsec(9500);
       }
       else
       {
        sbytep('F');
        sbytep('F');
        LCD_WCMD(0x01);
        LCD_STRING("No Fingel ID !!");
        beep(5000);
        dmsec(5000);
       }
}
//-----------------------------
//---------- Scankey ----------
unsigned char getkey()
{
       bit keypress;
       unsigned char KEYCODE[12]={ 0xb7,0xde,0xbe,
                                   0x7e,0xdd,0xbd,
                                   0x7d,0xdb,0xbb,
                                   0x7b,0xd7,0x77 };
       unsigned char i,j,k,a,x;
       k=0xef;
       for(i=0;i<=3;i++)
        {
              a=k|0xf;
              keyport=a;
              x=keyport;
              x=x&0xf;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                dmsec(100);
                if (x!=0xf)
                {
                        if(~keypress)
                        {
                                keypress=1;
                                beep(500);
                                a=k&0xf0;
                                x=x|a;
                                for(j=0;j<=11;j++)
                                if(x==KEYCODE[j])
                                        return(j);
                                return(0xff);
                        }
                return(0xff);
                }
                k=k<<1;
        }
        keypress=0;
        return(0xff);
}

void conv_key(void)
{
        unsigned char x;
        while(1)
        {
                x=getkey();
                if(x!=0xff)
                        {
                                switch(x) {
                                case 0x00:key_data='0';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
                case 0x01:key_data='1';
                    break;
                case 0x02:key_data='2';
                    break;
                case 0x03:key_data='3';
                    break:
                case 0x04:key_data='4';
                    break;

                case 0x05:key_data='5';
                    break;
                case 0x06:key_data='6';
                    break;

                case 0x07:key_data='7';
                    break;
                case 0x08:key_data='8';
                    break;
                case 0x09:key_data='9';
                    break;
                case 0x0A:key_data='c';
                    break;
                case 0x0B:key_data='e';
                    break;}

                break;
            }
        }
}

void read_key(void)
{       unsigned char i;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCD_WCMD(0x01);
        LCD_STRING("PassWord = ");
        sbytep('P');
        LCD_WCMD(0xC3);
        for (i=0;i<4;i++)
                {
                conv_key();
                LCD_WDAT('*');
                sbytep(key_data);
                }
        sbytep('Y');
}
//-----------------------------------
//---------- Smart Card ----------
void reset_crd()
{
 crd_rst=1;
 crd_clk=1;
 dmsec(100);
 crd_rst=0;
 crd_clk=0;
 crd_buff=0;
}

void read_idcrd()
{
        int i,j;
        crd_vdd=0;
        reset_crd();
        for (j=0;j<=3;j++)
        {
        for (i=0;i<8;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
                crd_clk=1;
            crd_io=1;
                dmsec(100);
                crd_buff<<=1;
                if (crd_io==1)
                        crd_buff++;
                crd_clk=0;
        }
    }
}

void read_crd(void)
{
    LCD_WCMD(0x01);
    LCD_STRING("Insert Your Card");
    dmsec(20000);
    if(crd_in==1)
            {
                read_iderd();
                if (crd_buff!=0xFF)
                {
                        sbytep('S');
                        sbytep(crd_buff);
                        sbytep('Y');
                }
                else
                {
                        sbytep('S');
                        sbytep('F');
                }
            }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        else
                {
                        sbytep('M');
                        sbytep('F');
                }
}
//-------------------------------------
//---------- Mode ----------
void mode1 (void)
{
        unsigned char idata st;
        sbytep('M');
        sbytep('1');
        while(!RI)
        {
                if (sensor==0)
                {
                        sbytep('E');
                        sbytep('F');
                        LCD_WCMD(0x01);
                        LCD_STRING("!!!! Danger !!!!");
                        Alarm=0;
                        beep(5000);
                        dmsec(5000);
                        break;
                }
                else
                        Alarm=1;
                sbytep('M');
                sbytep('1');
                st=rbytep();
                if (st=='F')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                        idn_fg();
                        idn_read_fg();
                        idn_send_PC();
                }
        if (st=='X')
                {
                        LCD_WCMD(0x01);
                        LCD_STRING(" Connecting  PC ");
                        break;
                }
    }

}

void mode2 (void)
{
    unsigned char idata st;
    sbytep('M');
    sbytep('2');
        while(!RI)
        {
            if (sensor==0)
            {
                sbytep('E');
                sbytep('F');
                Alarm=0;
                LCD_WCMD(0x01);
                LCD_STRING("!!!! Danger !!!!");
                beep(5000);
                dmsec(5000);
                break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        else
                Alarm=1;
        sbytep('M');
        sbytep('2');
        st=rbytep();
        if (st=='S')
        {
                read_crd();
        }
        if (st=='F')
        {
                idn_fg();
                idn_read_fg();
                idn_send_PC();
        }
        if (st=='X')
        {
                LCD_WCMD(0x01);
                LCD_STRING(" Connecting PC ");
                break;
        }
        if (st=='M')
        {
                LCD_WCMD(0x01);
                LCD_STRING("Please Try Again");
                beep(5000);
                dmsec(5000);
                sbytep('P');
                sbytep('F');
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void mode3 (void)
{
        unsigned char idata st;
        sbytep('M');
        sbytep('3');
                while(!RI)
                {
                        if (sensor==0)
                        {
                                sbytep('E');
                                sbytep('F');
                                LCD_WCMD(0x01);
                                LCD_STRING("!!!! Danger !!!!");
                                Alarm=0;
                                beep(5000);
                                dmsec(5000);
                                break;
                        }
                        else
                                Alarm=1;
                        sbytep('M');
                        sbytep('3');
                        st=rbytep();

                        if (st=='X')
                        {
                                LCD_WCMD(0x01);
                                LCD_STRING(" Connecting  PC ");
                                break;
                        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (st=='S')
{
        read_crd();
}
if (st=='P')
{
        read_key();
}
if (st=='F')
{
        idn_fg();
        idn_read_fg();
        idn_send_PC();
}
if (st=='K')
{
        LCD_WCMD(0x01);
        LCD_STRING("Smart Card Added");
        beep(500);
        dmsec(9500);
        break;
}
if (st=='M')
{
        LCD_WCMD(0x01);
        LCD_STRING("Please Try Again");
        beep(5000);
        dmsec(5000);
        sbytep('P');
        sbytep('F');
}
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void Add_crd(void)
{
        mode3();
}

//---------------------------------------
void main(void)
{
        unsigned char idata st;
        SCON0 = 0x52;         // set com1
        SCON1 = 0x52;         // set com2
  TMOD = 0x20;          // set T1 for com1,2
        CKMOD = 0x38;                    // set Ultra High Speed
  TH1  = 253;                    //250 115200 (TH1=256-
(2^(SMOD_x))*InputClock/(32*BaudRate))
        TR1  = 1;
        door = 0;
        LCD_INIT();
        sensor=1;
        Alarm=1;
        while(1)
        {
         LCD_WCMD(0x01);
         LCD_STRING("PleaseConnect PC");
         while(!RI)
         {
                st=rbytep();
                if (st=='C')
                        {
                        sbytep('C');
                        sbytep('Y');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCD_WCMD(0x01);
                LCD_STRING(" Connecting  PC ");
                break;
                }
        sbytep('C');
        sbytep('F');
}
while(1)
{
        if (RI)
        {
         st=rbytep();
         if(st=='Q')
                {
                sbytep('C');
                sbytep('F');
                break;
                }
         switch(st)
         {
                case '1':LCD_WCMD(0x01);
                                LCD_STRING("Start Mode 1 !!!");
                                dmsec(4000);
                                mode1();
                                break;
                case '2':LCD_WCMD(0x01);
                                LCD_STRING("Start Mode 2 !!!");
                                dmsec(4000);
                                mode2();
                                break;
                case '3':LCD_WCMD(0x01);
                                LCD_STRING("Start Mode 3 !!!");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    dmsec(4000);
                    mode3();
                    break;
            case 'K':LCD_WCMD(0x01);
                    LCD_STRING(" Add Smart Card ");
                    dmsec(4000);
                    Add_crd();
                    break;
            case 'A':LCD_WCMD(0x01);
                    LCD_STRING("  -- Add ID --  ");
                    read_PC();
                    add_fg();
                    add_send_PC();
                    break;
            case 'D':LCD_WCMD(0x01);
                    LCD_STRING("-- Dellete ID --");
                    read_PC();
                    del_fg();
                    del_send_PC();
                    break;
            case 'O':LCD_WCMD(0x01);
                    LCD_STRING(" Connecting PC ");
                    Alarm=1;
                    break;
          }
          sbytep('C');
          sbytep('Y');
          }
      }
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม LCD.C

```
#include "LCD.H"

void dmsec (unsigned int count)
{
   unsigned char i;
   while (count)
         {
     for (i=1;i<=228;i++);
     count--;
   }
}

unsigned char ReadLCD(void)
{
   unsigned char ret;
         LCD_DATA=0xFF;
         RS=0;
         RW=1;
         EN=1;
         dmsec(50);
         ret=LCD_DATA;
         return(ret);
}

void LCD_WCMD(unsigned char cmd)
{
         RS=0;
         RW=0;
         EN=1;
         LCD_DATA=cmd;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        EN=0;
        while (ReadLCD()&0x80);
}

void LCD_WDAT(unsigned char DATA)
{
        RS=1;
        RW=0;
        EN=1;
        LCD_DATA=DATA;
        EN=0;
        while (ReadLCD()&0x80);
}

void LCD_INIT(void)
{
        LCD_WCMD(0x02);
        LCD_WCMD(0x01);
        LCD_WCMD(0x0C);
        LCD_WCMD(0x06);
        LCD_WCMD(0x38);

}

void LCD_STRING(char *mess)
{
        unsigned char len,i;
        len=strlen(mess);
        for (i=0;i<len;i++)
        {
                if(i<8)
                        LCD_WCMD(0x80+i);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            else
                    LCD_WCMD(0xC0+(i-8));
        LCD_WDAT(mess[i]);
    }
}
```

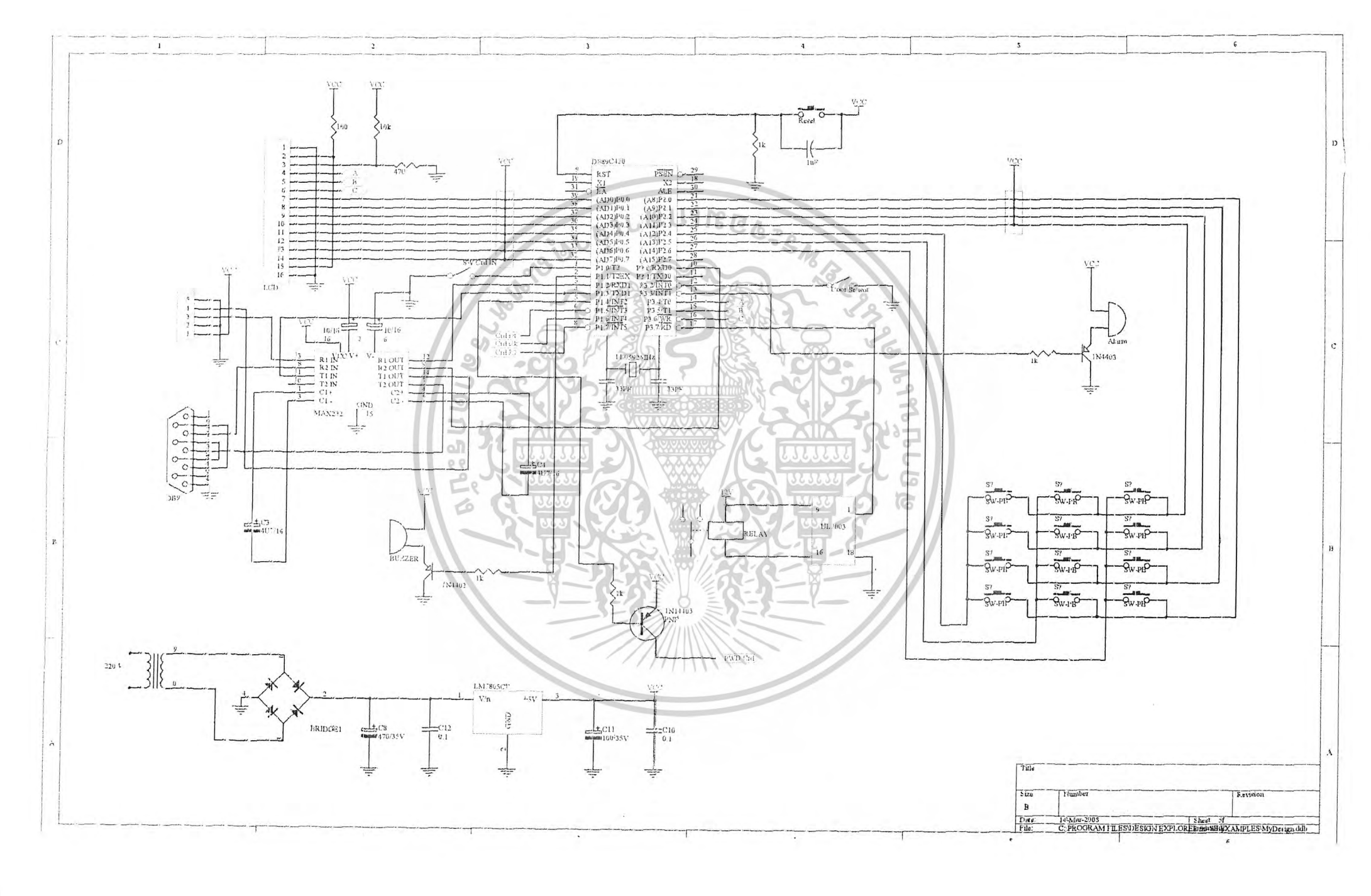

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม LCD.H

```
#include <REG420.H>
#include <string.h>
sbit RS=P3^6;
sbit RW=P3^5;
sbit EN=P3^4;
sfr LCD_DATA=0x80;

void dmsec (unsigned int count);
unsigned char ReadLCD(void);
void LCD_WCMD(unsigned char cmd);
void LCD_WDAT(unsigned char DATA);
void LCD_INIT(void);
void LCD_STRING(char *mess);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


VCC
140
10k
470
LCD
SCONTRIN
D889C120
RST
XTAL1
EA
P1.0/T2
P1.1/T2EX
P1.2/RXD1
P1.3/TXD1
P1.4/INT2
P1.5/INT3
P1.6/INT4
P1.7/INT5
PSEN
XTAL2
ALE
P3.0/RXD0
P3.1/TXD0
P3.2/INT0
P3.3/INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
Reset
1k
1uF
Boot Reload
11.0592MHz
33PF
33PF
MAX232
R1 IN
R2 IN
T1 IN
T2 IN
C1+
C1-
R1 OUT
R2 OUT
T1 OUT
T2 OUT
C2+
C2-
GND
DB9
C1
C2
C3
BUZZER
1N4403
RELAY
UL1003
1N4148
PNP
Alarm
SW-PB
320 V
BRIDGE1
C8
470/35V
C12
0.1
LM7805CT
Vin
+5V
GND
C11
100/35V
C16
0.1
Title
Size
B
Number
Revision
Date:
14-May-2005
Sheet
of
File:
C:\PROGRAM FILES\DESIGN EXPLORER 99 SE\EXAMPLES\MyDesign.ddb