สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องอ่านและเขียนสมาร์ทการ์ดสำหรับการประยุกต์ใช้งานในระบบห้องพัก

SMARTCARD READER AND WRITER FOR APPLICATION IN APARTMENT

โดย

นายธีรัช พิทักษ์รัตน์

นายอรรถพล วสุธากีรตินันท์

นายเอกพงษ์ เบญจพรกุลนิจ

เลขหมู่.....
เลขทะเบียน 62557
วัน,เดือน,ปี 19 ส.ค. 2549

b.11626195x
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องอ่านและเขียนสมาร์ทการ์ดสำหรับการประยุกต์ใช้งานในระบบห้องพัก

SMARTCARD READER AND WRITER FOR APPLICATION IN APARTMENT

โดย

นายธีรัช พิทักษ์รัตน์ 45010359

นายอรรถพล วสุธากีรตินันท์ 45010942

นายเอกพงษ์ เบญจพรกุลนิจ 45010984

อาจารย์ที่ปรึกษา

ผศ. เกรียงไกร วงศ์โรจนภรณ์

อาจารย์ที่ปรึกษาร่วม

รศ.ดร. สุวิพล สิทธิชีวภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องอ่านและเขียนสมาร์ทการ์ดสำหรับการประยุกต์ใช้งานในระบบห้องพัก

SMARTCARD READER AND WRITER FOR APPLICATION IN APARTMENT

ผู้จัดทำ

1. นายธีรัช พิทักษ์รัตน์ 45010359

2. นายอรรถพล วสุธากีรตินันท์ 45010942

3. นายเอกพงษ์ เบญจพรกุลนิจ 45010984

..... อาจารย์ที่ปรึกษา

(ผศ. เกรียงไกร วงศ์โรจนภรณ์)

..... อาจารย์ที่ปรึกษาร่วม

(รศ.ดร. สุวิพล สิทธิชีวภาค)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องอ่านและเขียนสมาร์ทการ์ดสำหรับการประยุกต์ใช้งานในระบบห้องพัก

SMARTCARD READER AND WRITER FOR APPLICATION IN APARTMENT

โดย นายธีรัช พิทักษ์รัตน์ 45010359
นายอรรถพล วสุธากีรตินันท์ 45010942
นายเอกพงษ์ เบญจพรกุลนิจ 45010984

อาจารย์ที่ปรึกษา ผศ. เกรียงไกร วงศ์โรจนภรณ์
อาจารย์ที่ปรึกษาร่วม รศ.ดร. สุวิพล สิทธิชีวภาค

บทคัดย่อ

โครงงานนี้เป็นการสร้างเครื่องอ่านและเขียนสมาร์ทการ์ด โดยใช้ไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของเครื่องอ่านและเขียน ซึ่งจะนำมาประยุกต์ใช้ในระบบห้องพัก โดยที่ห้องพักต่าง ๆ จะมีเครื่องอ่านบัตรติดอยู่เพื่อใช้ในการเปิดปิดประตู และเชื่อมต่ออยู่กับเครื่องคอมพิวเตอร์หลักผ่านพอร์ทอนุกรม โดยเครื่องคอมพิวเตอร์จะทำการบันทึกรหัสผ่านลงในสมาร์ทการ์ดและส่งรหัสผ่านนี้ไปยังเครื่องอ่านสมาร์ทการ์ดที่ห้องพัก

Abstract

This project presents smartcard reader and writer by using microcontroller to control its process. It will be applied in apartment. Each of them will have a smartcard reader connected to the door and the main computer via serial port in order to open and close the door. The main computer will record code in smartcard and send it to smartcard reader at the room.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 พื้นฐานของสมาร์ทการ์ด	2
2.2 ส่วนประกอบและโครงสร้างของสมาร์ทการ์ด	3
2.2.1 ตัวบัตรพลาสติก	3
2.2.2 หน้าสัมผัสและชิพสมาร์ทการ์ด (Smart card Module)	4
2.2.3 ตำแหน่งขาสำหรับการอินเตอร์เฟสกับสมาร์ทการ์ด	5
2.3 มาตรฐานที่เกี่ยวข้อง	9
2.3.1 มาตรฐาน ISO7816	9
2.3.1.1 ISO7816-1	9
2.3.1.2 ISO7816-2	10
2.3.1.3 ISO7816-3	10
2.3.1.4 ISO7816-4	10
2.3.1.5 ISO7816-5	10
2.3.1.6 ISO7816-6	10
2.4 องค์ประกอบต่างๆ ในการใช้งานสมาร์ทการ์ด	11
2.4.1 ตัวบัตรและตัวชิพ	11
2.4.2 สมาร์ทการ์ดรีดเดอร์ (Smart Card reader)	11
2.4.3 ซอฟแวร์	11
2.5 ชนิดของสมาร์ทการ์ด	11
2.5.1 การแบ่งสมาร์ทการ์ดโดยใช้ชนิดของชิพเป็นเกณฑ์	12
2.5.1.1 เมมโมรีการ์ด (Memory Card) หรือ ซิงโครนัสการ์ด (Synchronous Card)	12
2.5.1.2 โปรเซสเซอร์การ์ด (Processor Card) หรือ อะซิงโครนัสการ์ด (Asynchronous Card)	14
2.5.2 การแบ่งสมาร์ทการ์ดโดยใช้ชนิดของการสื่อสารเป็นเกณฑ์	15
2.5.2.1 ใช้หน้าสัมผัสของชิพ (Contact Card)	15
2.5.2.2 ไม่ใช้หน้าสัมผัสของชิพ (Contactless Card)	15
2.5.2.3 ใช้ทั้งสองแบบร่วมกัน (Contact and Contactless)	16
2.5.2.3.1 ใช้หน่วยความจำร่วมกัน (Combind Card)	16
2.5.2.3.2 ใช้หน่วยความจำแยกกัน (Hybrid Card)	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
2.6 การเชื่อมต่อชิพสมาร์ทการ์ด	17
2.6.1 การเชื่อมต่อสมาร์ทการ์ดชนิดเมมโมรี (Synchronous card Interface)	18
2.6.1.1 การเชื่อมต่อกับการ์ดที่ไม่มีระบบป้องกันข้อมูล (Free Access Memory Card)	19
2.6.1.2 การเชื่อมต่อกับการ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล (Security Memory Card)	19
2.6.2 การเชื่อมต่อสมาร์ทการ์ดชนิดโปรเซสเซอร์ (Asynchronous card Interface)	21
2.6.3 โปรโตคอลที่ใช้ในการสื่อสารกับสมาร์ทการ์ด	21
2.6.3.1 Reset and Answer-to-Reset	22
2.6.3.2 Command Mode	23
2.6.4 คำสั่งที่ใช้ในการติดต่อกับสมาร์ทการ์ด	24
2.6.4.1 การอ่านข้อมูลในหน่วยความจำหลัก (Main memory)	25
2.6.4.2 การอ่านข้อมูลในหน่วยความจำป้องกัน (Protection memory)	26
2.6.4.3 การเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก	27
2.6.4.4 การเขียนข้อมูลลงในหน่วยความจำป้องกัน	28
2.6.4.5 การอ่านข้อมูลในซีเคียวริตี้เมมโมรี่ (Security memory)	28
2.6.4.6 การเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่	29
2.6.4.7 การเปรียบเทียบข้อมูลที่ใช้ในการตรวจสอบ	29
2.6.5 การตรวจสอบรหัสป้องกัน (Programmable Security Code: PSC)	30
2.6.6 Reset Mode	31
2.6.7 การยกเลิกการทำงาน (Break)	32
2.7 การใช้งานไมโครคอนโทรเลอร์ (Microcontroller: MCS-51)	35
2.7.1 โครงสร้างของ MCS-51	35
2.7.2 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51	36
2.7.3 คุณสมบัติทั่วไปของ AT89C51	36
2.7.4 การเชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์	38
2.8 ความรู้เบื้องต้นเกี่ยวกับพอร์ตอนุกรม	40
2.8.1 การสื่อสารข้อมูลแบบซิงโครนัส	41
2.8.2 การสื่อสารข้อมูลแบบอะซิงโครนัส	42
2.8.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232	43
2.8.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง **หน้า**

2.8.5 มาตรฐานการรับส่งข้อมูลแบบอะซิงโครนัส (UART) 46

2.8.6 ULN 2803 47

2.9 การโพล (Polling) 48

2.10 การเลือก (Selection) 51

2.11 สวิตช์แม่เหล็กไฟฟ้าหรือรีเลย์ (Relay) 52

2.12 การใช้งานคีย์สวิตช์หรือคีย์แพด (Keypad) 54

2.13 ระบบฐานข้อมูล 56

2.13.1 ความหมายของระบบฐานข้อมูล 56

2.13.1.1 ระบบจัดการฐานข้อมูล (Database Management System (DBMS)) 56

2.13.1.2 แอปพลิเคชันฐานข้อมูล (Database Application) 56

2.13.2 ระบบการจัดการฐานข้อมูล 56

2.13.3 Relational Database 57

2.13.4 โครงสร้างของฐานข้อมูล 57

2.13.5 การออกแบบฐานข้อมูล 58

2.13.6 ภาษาเอสคิวแอล (SQL: structured Query Language) 58

2.14 สมาร์ทการ์ดกับการประยุกต์ใช้งานด้านต่าง ๆ 61

2.14.1 การประยุกต์ใช้งานด้านการสื่อสาร 61

(SmartCard Application in telecommunication)

2.14.1.1 ใช้ในการบริการโทรศัพท์สาธารณะ (PUBLIC PAYPHONES) 61

2.14.1.2 ใช้กับโทรศัพท์เคลื่อนที่ (MOBILE PHONES) 61

2.14.2 การประยุกต์ใช้งานด้านการเงิน การธนาคาร (Finance & Banking) 61

2.14.2.1 ใช้เป็นบัตรที่ใช้แทนเงินสด (Cash Card) 61

2.14.2.2 ใช้เป็นบัตรเครดิต (Credit Card) 61

2.14.3 การประยุกต์ใช้งานโดยใช้เป็นบัตรประจำตัวต่างๆ (Identification Card) 62

2.14.3.1 บัตรพนักงาน (Company Card) 62

2.14.3.2 บัตรนักศึกษา 62

2.14.4 การประยุกต์ใช้งานด้านการแพทย์ 62

2.14.5 การประยุกต์ใช้งานด้านระบบรักษาความปลอดภัย 62

2.14.6 การประยุกต์ใช้งานด้านการขนส่งมวลชน 62

2.14.6.1 ในระบบการจอดรถ (Parking System) 62

2.14.6.2 ใช้ในรับการชำระค่าผ่านทางและค่าโดยสาร 62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
บทที่ 3 การคำนวณและการสร้าง	63
3.1 วิธีการเลือกใช้บัตรสมาร์ทการ์ดและอุปกรณ์เชื่อมต่อกับหน้าสัมผัสของบัตรสมาร์ทการ์ด	63
3.1.1 การเลือกใช้บัตรสมาร์ทการ์ด	63
3.1.2 วิธีการเลือกใช้อุปกรณ์เชื่อมต่อหน้าสัมผัสของบัตรสมาร์ทการ์ด	63
3.2 วิธีการออกแบบในส่วนของฮาร์ดแวร์	64
3.2.1 วิธีการออกแบบวงจรที่ใช้ในการติดต่อกับสมาร์ทการ์ด	64
3.2.2 วิธีการออกแบบการสื่อสารผ่านพอร์ทอนุกรม RS232 (Serial Port)	65
3.2.3 วงจรขับรีเลย์ (Relay)	66
3.3 วิธีการออกแบบกระบวนการอ่านและเขียนข้อมูลของเครื่องอ่านเขียนบัตรสมาร์ทการ์ด	67
3.3.1 วิธีการออกแบบกระบวนการอ่านข้อมูลภายในบัตรสมาร์ทการ์ด	67
3.3.2 วิธีการออกแบบกระบวนการเขียนข้อมูลลงบนบัตรสมาร์ทการ์ด	67
3.4 วิธีการออกแบบกระบวนการทำงานโดยรวมของโครงงานนี้	68
3.4.1 วิธีการออกแบบกระบวนการทำงานของเครื่องเขียนบัตร (Card Writer)	68
3.4.2 วิธีการออกแบบกระบวนการทำงานของเครื่องแม่ (Master)	69
3.4.2.1 การทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1	70
3.4.2.2 การทำงานของไมโครคอนโทรลเลอร์ตัวที่ 2	71
3.4.2.3 การทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3	72
3.4.3 วิธีการออกแบบกระบวนการทำงานของเครื่องลูก (Slave)	73
3.4.4 โปรโตคอลที่ใช้ในการสื่อสารระหว่างฮาร์ดแวร์	74
3.5 วิธีการออกแบบโปรแกรมคอมพิวเตอร์	75
3.5.1 การออกแบบส่วนติดต่อกับผู้ใช้ (Interface)	75
3.5.2 กระบวนการทำงานของโปรแกรม	76
3.5.2.1 ส่วนการเขียนข้อมูลลงบัตรใหม่	76
3.5.2.2 ส่วนการโพลลิ่ง (Polling)	77
3.6 การออกแบบชิ้นงานทั้งหมดในโครงงานนี้	78
บทที่ 4 การทดลองและผลการทดลอง	81
4.1 การทดลองตรวจสอบช่วงเวลาของสัญญาณต่าง ๆ ที่ใช้ในการติดต่อกับชิพสมาร์ตการ์ด	81
4.1.1 สัญญาณ Vcc และ Reset	81
4.1.2 สัญญาณ Reset และสัญญาณนาฬิกา	82
4.1.3 สัญญาณ RS-485	84
4.2 การทดลองการสื่อสารผ่านพอร์ทอนุกรม RS232 (Serial Port)	85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เรื่อง	หน้า
4.3 การทดลองกระบวนการอ่านและเขียนข้อมูลลงในสมาร์ทการ์ดโดยใช้โปรแกรม Hyper Terminal	87
4.3.1 การทดลองในส่วนของกระบวนการ เขียนข้อมูลลงบนบัตรสมาร์ทการ์ด	87
4.3.2 การทดลองในส่วนของกระบวนการ อ่านข้อมูลภายในบัตรสมาร์ทการ์ด	92
4.3.3 การทดลองในส่วนของการลบข้อมูลทั้งหมดในสมาร์ทการ์ด	93
4.3.4 การทดลองในส่วนของกระบวนการสิ้นสุดการทำงานของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ด	93
4.4 การทดลองการทำงานของโปรแกรมแอพพลิเคชั่นที่เขียนขึ้น	94
4.4.1 ผลการทดลองเมื่อกดปุ่ม New Card	95
4.4.2 ผลการทดลองเมื่อกดปุ่ม Add Card	97
4.4.3 ผลการทดลองเมื่อกดปุ่ม Update Data	98
4.4.4 ผลการทดลองเมื่อนำบัตรมาใช้ในการเปิดประตูห้อง	98
บทที่ 5 บทวิจารณ์และบทสรุป	102
5.1 บทวิจารณ์และบทสรุป	102
5.1.1 ในส่วนกระบวนการอ่านข้อมูลภายในบัตรสมาร์ทการ์ด	102
5.1.2 ในส่วนกระบวนการเขียนข้อมูลลงบนบัตรสมาร์ทการ์ด	102
5.1.3 ในส่วนของการนำไปประยุกต์ใช้งาน	102
5.2 บทสรุป	103
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 2.1	ส่วนประกอบของสมาร์ทการ์ด	3
รูปที่ 2.2	ขนาดความกว้าง ×ยาว ×หนา ของบัตรสมาร์ทการ์ด	4
รูปที่ 2.3	ส่วนประกอบของสมาร์ทการ์ดโมดูลในสายการผลิตสมาร์ทการ์ด	4
รูปที่ 2.4	ขนาดของหน้าสัมผัสและทิศทางการเสียบบัตรสมาร์ทการ์ด ตามมาตรฐาน ISO7816	5
รูปที่ 2.5	ตำแหน่งหน้าสัมผัสที่ผิวหน้าของสมาร์ทการ์ด ตามมาตรฐาน ISO7816	5
รูปที่ 2.6	ตำแหน่งขาสำหรับการอินเตอร์เฟสกับสมาร์ทการ์ด	5
รูปที่ 2.7	วิธีการทดสอบการบิดงอสมาร์ทการ์ด	9
รูปที่ 2.8	หน้าที่การทำงานของแต่ละหน้าสัมผัส	10
รูปที่ 2.9	การแบ่งสมาร์ทการ์ดตามชนิดของหน่วยความจำ และประเภทของหน้าสัมผัส	12
รูปที่ 2.10	บล็อกไดอะแกรมโครงสร้างภายในชิพสมาร์ทการ์ดชนิด เมมโมรีการ์ด	12
รูปที่ 2.11	ตัวอย่างหน่วยความจำของ เมมโมรีการ์ด ชนิด PIN Protect	13
รูปที่ 2.12	บล็อกไดอะแกรมโครงสร้างภายในชิพสมาร์ทการ์ดชนิดโปรเซสเซอร์การ์ด	14
รูปที่ 2.13	บล็อกไดอะแกรมโครงสร้างภายในชิพสมาร์ทการ์ดชนิดคอนแทกต์เลส	15
รูปที่ 2.14	โครงสร้างภายในของสมาร์ทการ์ดชนิด คอนแทกต์เลส แบบ คอมไบน์การ์ด	16
รูปที่ 2.15	โครงสร้างภายในของสมาร์ทการ์ดชนิด คอนแทกต์เลส แบบ ไฮบริดจ์การ์ด	17
รูปที่ 2.16	ตัวอย่างวงจรเชื่อมต่อกับชิพสมาร์ทการ์ดด้วยวงจรลอจิก (C-MOS)	18
รูปที่ 2.17	ตัวอย่างวงจรเชื่อมต่อชิพสมาร์ทการ์ดชนิดเมมโมรี	18
รูปที่ 2.18	บล็อกไดอะแกรมแสดงส่วนต่างๆของการ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล	20
รูปที่ 2.19	ตัวอย่างวงจรเชื่อมต่อสมาร์ทการ์ดชนิดโปรเซสเซอร์	21
รูปที่ 2.20	รายละเอียดของการสื่อสารแบบซิงโครนัส และ การสื่อสารแบบ อะซิงโครนัส ในสมาร์ทการ์ด	22
รูปที่ 2.21	Reset และ Answer-to-Reset	22
รูปที่ 2.22	Command mode, Outgoing data mode และ Processing mode	24
รูปที่ 2.23	Command mode	24
รูปที่ 2.24	รูปแบบคำสั่งที่ใช้ในกระบวนการอ่านข้อมูลในหน่วยความจำหลัก	26
รูปที่ 2.25	รูปแบบคำสั่งที่ใช้ในกระบวนการอ่านข้อมูลในหน่วยความจำป้องกัน	27
รูปที่ 2.26	รูปแบบคำสั่งที่ใช้ในกระบวนการเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก	27
รูปที่ 2.27	รูปแบบคำสั่งที่ใช้ในกระบวนการเขียนข้อมูลลงในหน่วยความจำป้องกัน	28
รูปที่ 2.28	รูปแบบคำสั่งที่ใช้ในกระบวนการอ่านข้อมูลในซีเคียวริตี้เมมโมรี่	29
รูปที่ 2.29	รูปแบบคำสั่งที่ใช้ในกระบวนการเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่	29
รูปที่ 2.30	รูปแบบคำสั่งที่ใช้ในกระบวนการเปรียบเทียบข้อมูลที่ใช้ในการตรวจสอบ	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 2.31 กระบวนการตรวจสอบรหัสป้องกัน (PSC verification) 31
รูปที่ 2.32 ไทม์มิ่งไดอะแกรมของสัญญาณ Reset และ Answer-to-Reset 33
รูปที่ 2.33 ไทม์มิ่งไดอะแกรมของ Command Mode 33
รูปที่ 2.34 ไทม์มิ่งไดอะแกรมของ Outgoing Data Mode 34
รูปที่ 2.35 ไทม์มิ่งไดอะแกรมของ Processing Mode 34
รูปที่ 2.36 ไทม์มิ่งไดอะแกรมของ Break 35
รูปที่ 2.37 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 37
รูปที่ 2.38 การจัดขาของ MAX232 หรือ ICL232 38
รูปที่ 2.39 โครงสร้างภายในของ MAX232 หรือ ICL232 39
รูปที่ 2.40 แสดงวงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ และไมโครคอนโทรลเลอร์ MCS-51 39
รูปที่ 2.41 การส่งแบบขนาน 41
รูปที่ 2.42 การส่งแบบอนุกรม 41
รูปที่ 2.43 แผนผังการทำงานเวลาของการสื่อสารข้อมูลแบบซิงโครนัส 41
รูปที่ 2.44 รูปแบบของข้อมูลแบบอะซิงโครนัส 42
รูปที่ 2.45 คอนเน็กเตอร์อนุกรม 44
รูปที่ 2.46 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในรูปแบบต่างๆ 46
รูปที่ 2.47 ผังการทำงานภายในของขาสัญญาณต่างๆ ของพอร์ตอนุกรม 47
รูปที่ 2.48 การจัดขาของ ULN 2803 47
รูปที่ 2.49 การโพลบนสายแบบหลายจุด 48
รูปที่ 2.50 การโพลในสายแบบหลายจุดชนิด 2 ระดับ 49
รูปที่ 2.51 ลำดับการโพลของรูปที่ 2.50 49
รูปที่ 2.52 ตัวควบคุมจุดต่อ 50
รูปที่ 2.53 รีเลย์ (Relays) 53
รูปที่ 2.54 แมกเนติกคอนแทกเตอร์ (Magnetic Contactors) 53
รูปที่ 2.55 หรีดรีเลย์ (Reed Relay) 54
รูปที่2.56 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด 55
รูปที่2.57 รูปคีย์สวิตช์หรือคีย์แพดแบบ 3 × 4 คีย์ 55
รูปที่ 3.1 ลักษณะบัตรสมาร์ทการ์ดที่มีชิพไอซีเบอร์ SLE4442 บรรจุอยู่ 63
รูปที่ 3.2 ลักษณะซ็อกเกต ที่ใช้ในการเชื่อมต่อกับหน้าสัมผัสของสมาร์ทการ์ด 64
รูปที่ 3.3 วงจรการใช้งานของซ็อกเกตที่ใช้ในการเชื่อมต่อกับหน้าสัมผัสของบัตร 65
รูปที่ 3.4 วงจรทดลองการสื่อสารข้อมูลผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์ 65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 3.5 แผนผังการทำงานของการสื่อสารข้อมูลผ่านพอร์ทอนุกรม 66

รูปที่ 3.6 วงจรขับรีเลย์ 66

รูปที่ 3.7 แผนผังการทำงานของโปรแกรมกระบวนการอ่านข้อมูลภายในบัตรสมาร์ทการ์ด 67

รูปที่ 3.8 แผนผังการทำงานของโปรแกรมกระบวนการเขียนข้อมูลลงบนบัตรสมาร์ทการ์ด 67

รูปที่ 3.9 โครงสร้างการทำงานโดยรวมของโครงงานนี้ 68

รูปที่ 3.10 แผนผังการทำงานของเครื่องเขียนบัตร 69

รูปที่ 3.11 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1 70

รูปที่ 3.12 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 2 71

รูปที่ 3.13 แผนผังการทำงานในส่วนอินเทอร์รัพท์ของไมโครคอนโทรลเลอร์ตัวที่ 2 72

รูปที่ 3.14 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3 72

รูปที่ 3.15 แผนผังการทำงานของเครื่องลูก(slave) 73

รูปที่ 3.16 แผนผังการทำงานในส่วนอินเทอร์รัพท์ของเครื่องลูก 74

รูปที่ 3.17 โปรโตคอลที่ใช้ในการสื่อสารระหว่างฮาร์ดแวร์ 74

รูปที่ 3.18 หน้าต่างโปรแกรมที่ใช้ควบคุมการทำงานของเครื่องอ่านบัตร 75

รูปที่ 3.19 หน้าต่างแสดงผลที่อ่านมาจากฐานข้อมูล 76

รูปที่ 3.20 แผนผังการทำงานในส่วนของการเขียนข้อมูลลงบัตรใหม่ 77

รูปที่ 3.21 แผนผังการทำงานในส่วนของการโพลลิ่ง 78

รูปที่ 3.22 เครื่องแม่ (Master) 79

รูปที่ 3.23 เครื่องเขียนบัตร (Card Writer) 79

รูปที่ 3.24 เครื่องลูก (Slave) 79

รูปที่ 3.25 แบบจำลองประตูห้องพัก 80

รูปที่ 4.1 Channel 1: สัญญาณ Vcc, Channel 2 : สัญญาณ Reset 81

รูปที่ 4.2 ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ Vcc และ Reset 81
Channel 1 : สัญญาณ Vcc, Channel 2 : สัญญาณ Reset

รูปที่ 4.3 ช่วงเวลาขณะที่สัญญาณ Reset อยู่ในสถานะ High 82
Channel 1 : สัญญาณ Vcc, Channel 2 : สัญญาณ Reset

รูปที่ 4.4 Channel 1 : สัญญาณ Reset, Channel 2 : สัญญาณนาฬิกา 82

รูปที่ 4.5 ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ Reset และ สัญญาณนาฬิกา 83
Channel 1 : สัญญาณ Reset, Channel 2 : สัญญาณนาฬิกา

รูปที่ 4.6 ความถี่ของสัญญาณนาฬิกา 83
Channel 1 : สัญญาณ Reset, Channel 2 : สัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

		หน้า
รูปที่ 4.7	ข้อมูลที่อ่านออกมาในกระบวนการ Answer-to-Reset Channel 1 : สัญญาณข้อมูล, Channel 2 : สัญญาณนาฬิกา	84
รูปที่ 4.8	สัญญาณอินพุทจากไมโครคอนโทรลเลอร์	84
รูปที่ 4.9	สัญญาณเอาต์พุทจาก RS-485	85
รูปที่ 4.10	หน้าต่างโปรแกรมไฮเปอร์เทอร์มินอลเมื่อเปิดขึ้นมาใช้งานครั้งแรก	85
รูปที่ 4.11	หน้าต่างการกำหนดโหมดการทำงานให้ไฮเปอร์เทอร์มินอล	86
รูปที่ 4.12	หน้าต่างการกำหนดค่าต่าง ๆ ที่ใช้ในการติดต่อสื่อสาร	86
รูปที่ 4.13	ผลการทดลองการสื่อสารข้อมูลผ่านทางพอร์ทสื่อสารอนุกรม RS-232	87
รูปที่ 4.14	หน้าต่างโปรแกรม Hyper Terminal	88
รูปที่ 4.15	ผลการทดลองของหน้าจอโปรแกรม เพื่อเตรียมรอรับบัตรสมาร์ทการ์ด	88
รูปที่ 4.16	ผลการทดลองที่เกิดความผิดพลาดขึ้นเมื่อบัตรสมาร์ทการ์ดที่เสียบไม่ใช่เบอร์ SLE4442 หรือเสียบบัตรไม่ถูกต้อง	89
รูปที่ 4.17	ผลการทดลองเมื่อมีการเสียบบัตรสมาร์ทการ์ดเบอร์ SLE4442 เข้ามาจะมีเมนูให้เลือกทำรายการ	90
รูปที่ 4.18	เลือกทำรายการที่ 1 เพื่อทำการเขียนข้อมูลลงในบัตร	90
รูปที่ 4.19	ใส่ตำแหน่ง address ที่ต้องการจะเก็บข้อมูล	91
รูปที่ 4.20	ผลการทดลองเมื่อเขียนข้อมูลลงในบัตร	91
รูปที่ 4.21	เลือกทำรายการที่ 2 อ่านข้อมูลจากบัตรโดยเลือกตำแหน่ง address ที่ต้องการอ่าน	92
รูปที่ 4.22	แสดงการอ่านข้อมูลจากบัตร	92
รูปที่ 4.23	เลือกทำรายการที่ 3 ลบข้อมูลทั้งหมดที่อยู่ในบัตร	93
รูปที่ 4.24	แสดงผลการทดลองการลบข้อมูลทั้งหมด	93
รูปที่ 4.25	ผลการทดลองในกระบวนการสิ้นสุดการทำงานของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ด	94
รูปที่ 4.26	หน้าต่างโปรแกรมที่ใช้ควบคุมการทำงานของเครื่องอ่านบัตร	95
รูปที่ 4.27	หน้าต่างใหม่เมื่อกดปุ่ม New Card	95
รูปที่ 4.28	หน้าต่างแสดงข้อความและรูปภาพให้เสียบบัตร	96
รูปที่ 4.29	หน้าต่างขณะเขียนข้อมูลลงในบัตร	96
รูปที่ 4.30	หน้าต่างแสดงข้อความและรูปภาพให้ดึงบัตรออก	97
รูปที่ 4.31	ข้อความบนจอแอลซีดีที่เครื่องลูก (slave)	97
รูปที่ 4.32	หน้าต่างเมื่อกดปุ่ม Update Data	98
รูปที่ 4.33	ข้อความต้อนรับบนจอแอลซีดี	98
รูปที่ 4.34	ข้อความรอรับการเสียบบัตร	99
รูปที่ 4.35	ข้อความเสียบบัตรผิดประเภท	99

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

หน้า

รูปที่ 4.36 ข้อความให้ดึงบัตรออก 99

รูปที่ 4.37 ข้อความอนุญาตให้ผ่าน 99

รูปที่ 4.38 ลักษณะกลอนประตูที่ติดอยู่ที่ห้อง 100

รูปที่ 4.39 ข้อความหลังจากดึงบัตรออก 100

รูปที่ 4.40 ข้อความบัตรถูกประเภท แต่รหัสผ่านไม่ถูกต้อง 100

รูปที่ 4.41 หน้าต่างแสดงผลที่อ่านมาจากฐานข้อมูล 101

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงคุณลักษณะทางไฟฟ้าของสัญญาณที่ใช้ในการสื่อสารกับสมาร์ทการ์ด 6

ตารางที่ 2.2 แสดงคุณลักษณะทางไฟฟ้าของ V_{pp} ภายใต้การทำงานปกติ 7

ตารางที่ 2.3 แสดงคุณลักษณะทางไฟฟ้าของสัญญาณที่ขา *CLK* ภายใต้การทำงานปกติ 8

ตารางที่ 2.4 แสดงคุณลักษณะทางไฟฟ้าของสัญญาณที่ขา *RST* ภายใต้การทำงานปกติ 8

ตารางที่ 2.5 แสดงระดับแรงดันไฟเลี้ยงและกระแสที่ป้อนให้กับสมาร์ทการ์ด ภายใต้การทำงานปกติ 8

ตารางที่ 2.6 ชุดคำสั่งของสมาร์ดการ์ดเบอร์ SLE4442 25

ตารางที่ 2.7 หน่วยความจำหลักในสมาร์ทการ์ด 25

ตารางที่ 2.8 หน่วยความจำป้องกันในสมาร์ทการ์ด 26

ตารางที่ 2.9 ซีเคียวริตี้เมมโมรี่ในสมาร์ทการ์ด 28

ตารางที่ 2.10 ชุดคำสั่งที่ใช้ในกระบวนการตรวจสอบรหัสป้องกัน 30

ตารางที่ 2.11 ช่วงเวลาของสัญญาณต่าง ๆ ที่ใช้ในการติดต่อกับสมาร์ทการ์ด 32

ตารางที่ 2.12 ตารางของไมโครคอนโทลเลอร์แบบชิพเดี่ยวในตระกูล 51 35

ตารางที่ 2.13 การจัดขาสัญญาณของพอร์ตอนุกรมในแบบต่างๆ และหน้าที่การทำงาน 44

ตารางที่ 2.14 สรุปคำสั่งของ ภาษาเอสคิวแอล 60

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันเทคโนโลยีการใช้บัตรเพื่อใช้ในการเก็บรักษาข้อมูลและการเข้าถึงข้อมูลนั้นมีการพัฒนาไปเป็นอันมากโดยเฉพาะระบบรักษาความปลอดภัยของข้อมูลที่ถูกเก็บ โดยมีความสามารถที่สูงขึ้นกว่าเมื่อก่อนมาก ซึ่งเทคโนโลยีที่มีความก้าวหน้าในการเก็บข้อมูลนั้นจะอยู่ในรูปวงจรรวม (Integrated *Circuit:IC)* *อันที่จะนำไปใช้งานได้อย่างสะดวก โดยสามารถบรรจุลงบนบัตรพลาสติกที่มีขนาดเท่ากับ* บัตรเครดิตหรือบัตรประจำตัวทั่วไป ซึ่งเราเรียกกันว่า บัตรสมาร์ทการ์ด (Smart Card)

สมาร์ทการ์ด คือ บัตรพลาสติกที่มีชิพ IC (Integrated circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเองโดยวิธีการเข้ารหัสตามมาตรฐาน DES Algorithm (Data Encryption Standard) ทำให้ระบบมีความปลอดภัยสูงขึ้น ด้วยคุณสมบัติประการหนึ่งที่ทำให้สมาร์ทการ์ดแตกต่างจากบัตรพลาสติกทั่วไปก็คือ ขณะทำรายการ (Transaction) สมาร์ทการ์ดสามารถทำงานได้ด้วยตัวของมันเองโดยไม่ต้องอาศัยติดต่อสื่อสารกับระบบหลัก (Font End) นั้นก็คือ สมาร์ทการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับศูนย์กลางข้อมูลเหมือน กับบัตรแถบแม่เหล็ก (Off-line) ทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก

ในการประมวลผล และจัดเก็บข้อมูล สมาร์ทการ์ดสามารถทำได้อย่างรวดเร็วกว่าสื่อสำหรับ*จัดเก็บข้อมูลชนิดอื่นๆ* *ด้วยขนาดที่เท่ากับบัตรแถบแม่เหล็กทำให้สะดวกในการจัดเก็บและพกพา* นอกจากนี้สมาร์ทการ์ดยังมีคุณสมบัติด้านความทนทานไม่ว่าจะเป็นรังสีชนิดต่างๆ สนามแม่เหล็ก ไฟฟ้าสถิต ความชื้น ความร้อน การบิดงอ ฯลฯ ก็ไม่สามารถทำให้สมาร์ทการ์ดเสียหายได้โดยง่าย

จากคุณสมบัติของบัตรสมาร์ทการ์ดและความก้าวหน้าทางเทคโนโลยีของกระบวนการทำงานต่างๆ ที่เกี่ยวข้องกับข้อมูลที่จะถูกเก็บหรือใช้งานผ่านตัวบัตรสมาร์ทการ์ดนั้น มีราคาที่ยังค่อนข้างสูง ทางผู้จัดทำได้ จึงได้ทำการประดิษฐ์เครื่องอ่าน-เขียนบัตรสมาร์ทการ์ดนี้ขึ้น โดยสามารถนำไปใช้งานในลักษณะต่างๆตามที่ต้องการ ภายใต้งบประมาณการทำที่ค่อนต่ำ เป็นการลดต้นทุนของการนำไปประยุกต์ใช้งานอีกด้วย

โดยในโครงงานนี้ได้รวบรวมและอธิบายถึงโครงสร้างและการทำงานต่างๆที่เกี่ยวข้องกับเทคโนโลยีของบัตรสมาร์ทการ์ด รวมทั้งการออกแบบกระบวนการที่เกี่ยวข้องกับตัวบัตรและข้อมูลต่างๆ *และจะนำไปประยุกต์ใช้งานเป็นบัตรสำหรับใช้ในการเปิดประตูห้องพักในโรงแรมหรือในอาพาร์ทเมนท์* ซึ่งจะมีการบันทึกเวลาลงในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

จากบทที่แล้วที่ได้กล่าวถึงเทคโนโลยีและคุณสมบัติต่าง ๆ ของสมาร์ทการ์ดมาบ้างแล้ว ในบทนี้จะขอกล่าวถึงในรายละเอียดต่าง ๆ ไม่ว่าจะเป็นโครงสร้าง คุณสมบัติ มาตรฐาน วิธีการใช้งานของสมาร์ทการ์ด รวมถึงวิธีการติดต่อกับชิพสมาร์ทการ์ด

2.1 พื้นฐานของสมาร์ทการ์ด

สมาร์ทการ์ด (Smart Card) มีพื้นฐานมาจากระบบไมโครโปรเซสเซอร์ ซึ่งมีแนวคิดเริ่มแรกจากการนำชิพหน่วยความจำ (EEPROM) มาฝังลงในบัตรพลาสติกที่มีขนาดเท่ากับบัตรเครดิตหรือบัตรเอทีเอ็ม ที่มีหน่วยเก็บข้อมูลและหน่วยประมวลผลที่เรียกว่า ไมโครชิพ (microchip) หรือชิพ (chip) ติดอยู่บนบัตร ทำงานโดยมีหน้าสัมผัสเป็นขาเชื่อมต่อกับระบบภายนอก ในการเชื่อมต่อต้องมีการป้อนกระแสไฟฟ้าให้ชิพหน่วยความจำจึงจะสามารถทำงานได้ การสั่งงานเพื่ออ่านหรือเขียนข้อมูลจากชิพหน่วยความจำนั้นทำได้โดยการเชื่อมต่อสัญญาณผ่านหน้าสัมผัสที่กำหนดไว้แล้ว

ในการเชื่อมต่อขาสัญญาณของชิพหน่วยความจำแบบธรรมดา อาจจะไม่เหมาะสมนักสำหรับบัตรพลาสติกขนาดเล็ก เนื่องจากจำนวนขาสัญญาณของหน่วยความจำ (Bus) มีจำนวนไม่น้อย ยิ่งหน่วยความจำที่มีความจุสูงๆ ยิ่งต้องใช้สัญญาณอ้างอิงตำแหน่งของข้อมูล (Address Bus) มากขึ้น จึงมีการนำเอาระบบสื่อสารแบบซิงเกิลบัส (Single Bus) มาใช้ในการรับส่งข้อมูล ซึ่งมีด้วยกันหลายรูปแบบ ในการนำเอาระบบสื่อสารแบบอนุกรมมาใช้ จำเป็นต้องมีการป้อนสัญญาณนาฬิกาเพื่อกำกับจังหวะการรับส่งข้อมูลแต่ละบิต ทำให้ต้องมีหน้าสัมผัสสำหรับสัญญาณนาฬิกาบนชิพสมาร์ทการ์ดเพิ่มขึ้นมา แต่ก็ทำให้ขาเชื่อมต่อลดลง ด้วยเหตุนี้สมาร์ทการ์ดชนิดหน่วยความจำจึงเป็นสมาร์ทการ์ดชนิดแรกที่ถูกสร้างขึ้น

การนำเอาหน่วยความจำมาใส่ในบัตรพลาสติก ทำให้เกิดข้อดีเหนือบัตรแถบแม่เหล็กด้วยความจุข้อมูลที่มากกว่า ไม่มีผลต่อสนามแม่เหล็กไฟฟ้าและรอยขีดข่วน แต่ข้อเสียประการหนึ่งของการใช้หน่วยความจำเพียงอย่างเดียวคือ สามารถทำการอ่านและเขียนข้อมูลได้อย่างอิสระเช่นเดียวกับบัตรแถบแม่เหล็ก จึงถือได้ว่าความปลอดภัยของข้อมูลเกือบเป็นศูนย์ นั่นคือข้อมูลภายในสมาร์ทการ์ดชนิดนี้ไม่เป็นความลับ ด้วยเหตุนี้จึงมีการเพิ่มวงจรสำหรับป้องกันลงไปอีก เพื่อให้ผู้ออกบัตรสามารถกำหนดสิทธิ์ในการเข้าถึงข้อมูลแต่ละไบต์ด้วยวงจรฟิวส์แมทริกธรรมดาๆ ที่เมื่อกำหนดเงื่อนไขไปแล้วไม่สามารถแก้ไขได้อีก ต่อมาเมื่อเทคโนโลยีด้านเซมิคอนดักเตอร์สูงขึ้น จึงมีการออกแบบวงจรที่สามารถกำหนดเป็นกุญแจรหัสทุกครั้งที่บัตรเริ่มทำงาน เพื่อป้องกันการเจาะระบบอีกชั้นหนึ่ง อีกทั้งกุญแจรหัสก็ยังสามารถเปลี่ยนแปลงได้อีกด้วย

ต่อมามีการนำเอาไมโครโปรเซสเซอร์มาใส่ลงในสมาร์ทการ์ด ทำให้เกิดเป็นสมาร์ทการ์ดชนิดใหม่ที่มีความซับซ้อนยิ่งขึ้น การเข้าถึงข้อมูลไม่สามารถทำได้โดยตรงเหมือนอย่างสมาร์ทการ์ดชนิดหน่วยความจำ การใช้งานสมาร์ทการ์ดชนิดนี้ต้องเขียนขึ้นเป็นชุดคำสั่ง และส่งให้กับชิพไมโครโปรเซสเซอร์ทำงานแทน การที่ใส่ชิพไมโครโปรเซสเซอร์ลงไปในสมาร์ทการ์ด ทำให้ต้องมีการเพิ่มส่วนของหน่วยความจำโปรแกรม (OS – Operating System) สำหรับไมโครโปรเซสเซอร์ เพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครโปรเซสเซอร์สามารถทำการประมวลผลคำสั่งต่างๆ และสามารถโปรแกรมการเข้าถึงข้อมูลทำให้ช่องโหว่ที่สำคัญของสมาร์ทการ์ดได้รับการแก้ไขจนเกือบสมบูรณ์แบบ

2.2 ส่วนประกอบและโครงสร้างของสมาร์ทการ์ด

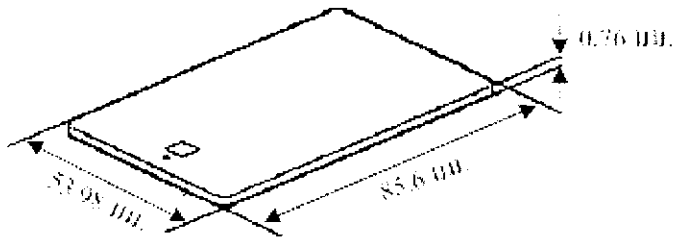
สมาร์ทการ์ดประกอบด้วยบัตรพลาสติก กาวหรือวัสดุที่ใช้เชื่อมต่อ และหน้าสัมผัสที่บรรจุชิพสมาร์ทการ์ดเรียบร้อยแล้ว ซึ่งส่วนประกอบต่างๆกำหนดขึ้นโดยองค์การมาตรฐานสากล ISO (International Standard Organization) โดยมีมาตรฐานตาม ISO7816 ซึ่งเนื้อหาหลักของมาตรฐานนี้ คือ ISO7816-1, ISO7816-2, ISO7816-3 มีข้อกำหนดต่างๆ ที่เกี่ยวกับบัตรสมาร์ทการ์ดได้แก่ ลักษณะทางกายภาพของสมาร์ทการ์ด ขนาดสัดส่วน (กว้าง× ยาว×หนา) และตำแหน่งหน้าสัมผัสบนสมาร์ทการ์ด รวมทั้งคุณสมบัติทางไฟฟ้าของสัญญาณต่างๆ และรูปแบบของโปรโตคอลที่ใช้ในการสื่อสาร ซึ่งบัตรจะมีส่วนประกอบต่างๆ แสดงดังรูปที่ 2.1

รูปที่ 2.1 ส่วนประกอบของสมาร์ทการ์ด

2.2.1 ตัวบัตรพลาสติก

สมาร์ทการ์ดเป็นชิพไอซีที่มีขนาดเล็กที่ถูกสร้างขึ้นเช่นเดียวกับชิ้นส่วนอิเล็กทรอนิกส์อื่นๆ ที่สร้างจากสารกึ่งตัวนำ นำมาติดลงบนหน้าสัมผัส และทำการฝังลงในเนื้อบัตรพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นตัวบัตรจะใช้พลาสติก 4 ชนิด ได้แก่ พีวีซี (PVC :Polyvinyl Chloride), เอบีเอส (ABS: Acrylonitrile Butadiene styrene) , พีซี (PC : Polycarbonate) , และ พีอีที (PET :Polyethylene Terephthalate) ในประเทศไทยจะใช้บัตรพลาสติกชนิดพีวีซีมากเป็นอันดับหนึ่ง ส่วนอันดับสองเป็นบัตรพลาสติกชนิด เอบีเอส ซึ่งบัตรพลาสติกชนิด พีวีซี มักนำมาใช้เป็นบัตรเอทีเอ็ม (ATM) , บัตรเครดิต – เดบิต , บัตรประจำตัวประชาชน ฯลฯ ส่วนบัตรพลาสติกชนิด เอบีเอส ไม่ค่อยพบว่าใช้งานกันมากนัก เนื่องจากราคาสูงกว่า และลายที่พิมพ์ลงบนบัตรไม่สวยงามคงทนเท่าบัตรพลาสติกชนิด พีวีซี

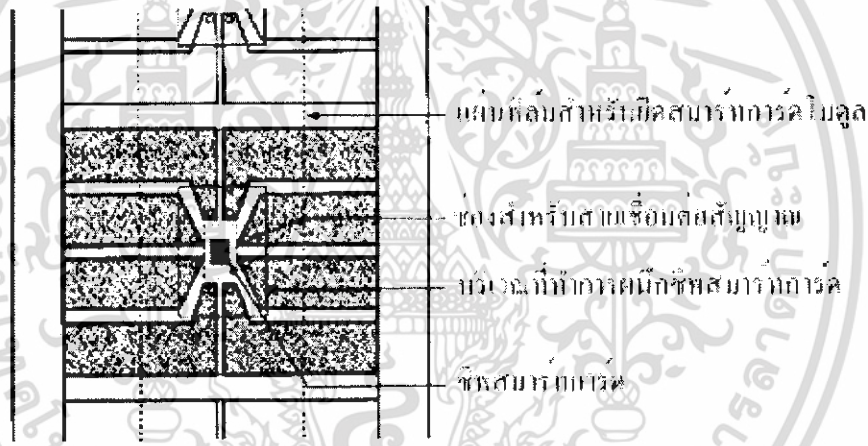
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 ขนาดความกว้าง ×ยาว ×หนา ของบัตรสมาร์ทการ์ด

2.2.2 หน้าสัมผัสและชิพสมาร์ทการ์ด (Smart card Module)

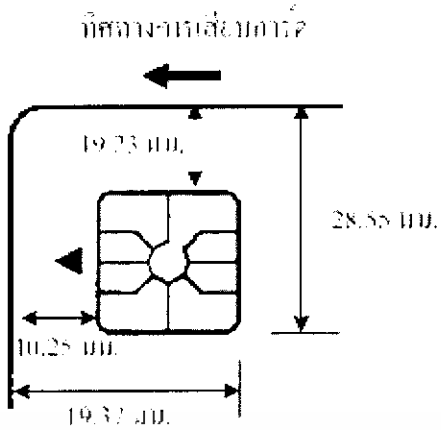
สมาร์ทการ์ดโมดูลหรือหน้าสัมผัสและชิพสมาร์ทการ์ด คือ ส่วนที่แสดงความเป็นตัวตนของสมาร์ทการ์ดที่ชัดเจนที่สุด สมาร์ทการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่ามันคือสมาร์ทการ์ดที่มีการฝังชิพไว้ในเนื้อบัตร ดังนั้นการที่จะระบุว่าบัตรใบใดเป็นบัตรสมาร์ทการ์ดนั้น ต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก



รูปที่ 2.3 ส่วนประกอบของสมาร์ทการ์ดโมดูลในสายการผลิตสมาร์ทการ์ด

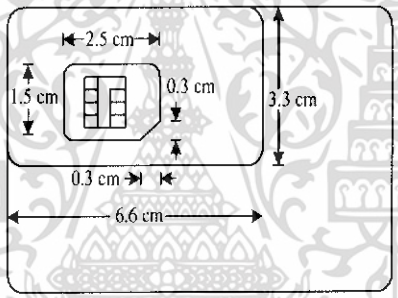
สมาร์ทการ์ดแต่ละใบจะมีการกำหนดขนาดของหน้าสัมผัสและทิศทางการเสียบบัตรไว้โดยกำหนดตามมาตรฐาน ISO7816 ซึ่งมีขนาดและทิศทางดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ขนาดของหน้าสัมผัสและทิศทางการเสียบบัตรสมาร์ทการ์ดตามมาตรฐาน ISO7816

ตำแหน่งหน้าสัมผัสที่วางอยู่บนบัตรสมาร์ทการ์ด จะมีข้อกำหนดตามมาตรฐาน ISO7816 โดยผู้ผลิตทุกรายนั้นจะมีการใช้มาตรฐานนี้ในการผลิตบัตรที่มีขนาดและตำแหน่งเดียวกัน ทุกบริษัท โดยขนาดและตำแหน่งแสดงดังรูปที่ 2.5



รูปที่ 2.5 ตำแหน่งหน้าสัมผัสที่ผิวหน้าของสมาร์ทการ์ด ตามมาตรฐาน ISO7816

2.2.3 ตำแหน่งขาสำหรับการอินเตอร์เฟสกับสมาร์ทการ์ด

รูปที่ 2.6 ตำแหน่งขาสำหรับการอินเตอร์เฟสกับสมาร์ทการ์ด

จากรูปที่ 2.6 เป็นตำแหน่งของขาสัมผัสทั้ง 8 ขา ซึ่งเป็นจุดที่เครื่องอ่านใช้อินเตอร์เฟสกับวงจรภายในการ์ด เนื่องจากขาที่ใช้งานเป็นหลักจะมีอยู่เพียง 6 ขาโดยที่ขา อาร์เอฟยู (RFU : Reserved for future use)มักไม่ถูกใช้งาน (อาจมีการใช้งานบ้างกับการ์ดของบางผู้ผลิตเท่านั้น) ขาสัมผัสที่จะถูกกล่าวถึงนี้จึงจะขอกล่าวเพียง 6 ขาที่สำคัญเท่านั้น ในส่วนของคุณลักษณะทางไฟฟ้าของสัญญาณที่ขาต่างๆ (ซึ่งก็คือช่วงค่าทางไฟฟ้าของแต่ละสัญญาณที่มาตรฐาน ISO7816 กำหนดไว้นั่นเอง) สามารถเขียนได้ตาม ตารางที่ 2.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์	สภาวะเงื่อนไข	ค่าต่ำสุด	ค่าสูงสุด	หน่วย
V_{ih} (1)	I_{ih} (max) = ± 500 μA	2.0	V_{cc}	V
	I_{ih} (max) = ± 50 μA	0.7 × V_{cc}	V_{cc} (3)	V
V_{il}	I_{il} (max) = 1 mA	0	0.8	V
V_{oh} (2)	I_{ol} (max) = ± 100 μA	2.4	V_{cc}	V
	I_{ol} (max) = ± 20 μA	3.8	V_{cc}	V
V_{ol}	I_{ol} (max) = 1 mA	0	0.4	V
t_r,t_f	C_{in} = 30 pF ; C_{out} = 30 pF	-	1	μS

ตารางที่ 2.1 แสดงคุณลักษณะทางไฟฟ้าของสัญญาณที่ใช้ในการสื่อสารกับสมาร์ทการ์ด

หมายเหตุ : (1) สำหรับเครื่องอ่านการ์ดบนสภาวะเงื่อนไขที่ต่างกัน 3 กรณี

(2) ในที่นี้กำหนดให้มีการต่อตัวต้านทานพูลอัพ 20 กิโลโอห์ม อยู่ด้วย

(3) ระดับแรงดันของสัญญาณที่ขา ไอ/โอ(I/O) ควรมีค่าอยู่ระหว่าง 0.3 โวลต์, V_{cc} +0.3 โวลต์

โดยที่

V_{ih} : แรงดันอินพุทค่าสูง

V_{il} : แรงดันอินพุตค่าต่ำ

V_{cc} : ระดับแรงดันไฟเลี้ยงที่ใช้งาน

V_{pp} : ระดับแรงดันที่ใช้โปรแกรมสมาร์ทการ์ดที่ภายในเป็น PROM

V_{oh} : แรงดันเอาต์พุตค่าสูง

V_{ol} : แรงดันเอาต์พุตค่าต่ำ

t_r : ช่วงระยะเวลาขาขึ้น (คิดช่วง 10% ถึง 90% ของแอมปลิจูด)

t_f : ช่วงระยะเวลาขาลง (คิดช่วง 10% ถึง 90% ของแอมปลิจูด)

I_{ih} : กระแสไฟฟ้าอินพุตค่าสูง

I_{il} : กระแสไฟฟ้าอินพุตค่าต่ำ

I_{cc} : ระดับกระแสไฟฟ้าจากแหล่งจ่ายไฟ

I_{pp} : กระแสที่ใช้โปรแกรมสมาร์ทการ์ดที่ภายในเป็น PROM

I_{oh} : กระแสไฟฟ้าที่เอาต์พุตค่าสูง

I_{ol} : กระแสไฟฟ้าที่เอาต์พุตค่าต่ำ

C_{in} : ค่าความจุไฟฟ้าที่อินพุต

C_{out} : ค่าความจุไฟฟ้าที่เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.6 แต่ละขาในรูปจะมีหน้าที่ต่างๆดังนี้ คือ

1. *ขา ไอ/โอ*

ทำงานเป็นอินพุตหรือเอาท์พุตเพื่อรับหรือส่งข้อมูลอนุกรม (Serial data) ด้วยรูปแบบแบบฮาล์ฟดูเพลกซ์ (Half duplex) โดยการผลัดกันส่งข้อมูลบนสายนำสัญญาณเดียว โดยที่ ขา ไอ/โอ นี้จะมีสถานะเป็นได้อยู่ 2 สถานะด้วยกัน คือ สถานะลอจิกสูง (High) เมื่อการ์ดรับหรือส่งบิตข้อมูลที่เป็นลอจิก "1" สถานะลอจิกต่ำ (Low) เมื่อการ์ดรับหรือส่งข้อมูลที่เป็นบิตลอจิก "0"

2. *ขา* V_{pp}

ใช้ป้อนแรงดันค่าสูงเพื่อเขียนหรือลบข้อมูลในหน่วยความจำ Non-volatile ขานี้จะมีสถานะเป็นได้ 2 สถานะด้วยกัน คือ สถานะว่าง (Idle state) และสถานการณ์ทำงาน (Active state) โดยคุณลักษณะทางไฟฟ้าของสัญญาณที่ขา V_{pp} ควรจะมีค่าเป็นดังตารางที่ 2.2 จะเห็นได้ว่า เมื่อต้องการจะเขียนหรือลบข้อมูลในหน่วยความจำ จะต้องใช้ระดับแรงดันและกระแสไฟฟ้าที่สูงขึ้นกว่าปกติ ซึ่งในทางปฏิบัติแล้วค่าที่ใช้จะมากหรือน้อยเพียงใดก็ขึ้นกับว่าผู้ผลิตการ์ดชนิดนั้นกำหนดค่าไว้เป็นเท่าไหร่ด้วย

สัญลักษณ์	สภาวะเงื่อนไข	ค่าต่ำสุด	ค่าสูงสุด	หน่วย
V_{pp}	สถานะว่าง (ยังไม่ทำการโปรแกรมการ์ด)	$0.95 \times V_{cc}$	$1.05 \times V_{cc}$	*V*
I_{pp}		-	*20*	*mA*
V_{pp}	สถานะทำงาน (เริ่มการโปรแกรมการ์ด)	$0.975 \times V_{cc}$	$1.025 \times P$	*V*
I_{pp}		-	*1*	*mA*

ตารางที่ 2.2 แสดงคุณลักษณะทางไฟฟ้าของ V_{PP} ภายใต้การทำงานปกติ

หมายเหตุ : ในที่นี้กำหนดให้ค่า P และ I ในการอินเตอร์เฟสการ์ด มีค่าเท่ากับ 5 และ 50 ตามลำดับ

3. ขาคล็อค (CLK)

ใช้รับสัญญาณที่ส่งมาจากเครื่องอ่าน เพื่อใช้เป็นสัญญาณนาฬิกาให้กับระบบ CPU/MPU Card หรือ เพื่อเป็นสัญญาณนาฬิกาอ้างอิงในการเข้าถึงข้อมูลสำหรับการอินเตอร์เฟสกับ เมมโมรีการ์ด (หมายถึงการที่ขา *CLK* จะถูกใช้เป็นสัญญาณอ้างอิงในการอ่านหรือเขียนข้อมูลแต่ละบิตให้ขาไอ/โอ โดยเครื่องอ่านจะต้องสร้างสัญญาณนาฬิกาขึ้น 1 ลูก เมื่อต้องการรับหรือส่งข้อมูล 1 บิต) คุณลักษณะทางไฟฟ้าของขา *CLK* ควรจะเป็นตามตารางที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์	สภาวะเงื่อนไข	ค่าต่ำสุด	ค่าสูงสุด	หน่วย
V_{ih} (1)	$I_{ih}(max) = \pm\ 200\ \mu A$	2.4	V_{cc} (2)	V
	$I_{ih}(max) = \pm\ 20\ \mu A$	$0.7 \times V_{cc}$	V_{cc} (2)	V
	$I_{ih}(max) = \pm\ 10\ \mu A$	$V_{cc} - 0.7$	V_{cc} (2)	V
V_{il}	$I_{il}(max) = \pm\ 200\ \mu A$	0 (2)	0.5	V
t_r , t_f	$C_{in} = 30\ pF$	-	9% ของคาบเวลา (แต่ไม่เกิน 5 μ S)	V

ตารางที่ 2.3 แสดงคุณลักษณะทางไฟฟ้าของสัญญาณที่ขา *CLK* ภายใต้การทำงานปกติ

หมายเหตุ : (1) สำหรับเครื่องอ่านการ์ดบนสภาวะเงื่อนไขที่ต่างกัน 3 กรณี

(2) ระดับแรงดันของสัญญาณที่ขา *CLK* ควรมีค่าอยู่ระหว่าง 0.3 โวลต์ ถึง V_{cc}+0.3โวลต์

4. *ขา RST*

ใช้ในการรีเซตสมาร์ทการ์ดให้เริ่มต้นทำงานใหม่ คุณลักษณะทางไฟฟ้าของสัญญาณที่ขา *RST* ควรจะเป็นไปตาม ตารางที่ 2.4

สัญลักษณ์	สภาวะเงื่อนไข	ค่าต่ำสุด	ค่าสูงสุด	หน่วย
V_{ih} (1)	$I_{ih}(max) = \pm\ 200\ \mu A$	4	V_{cc} (2)	V
	$I_{ih}(max) = \pm\ 10\ \mu A$	$V_{cc} - 0.7$	V_{cc} (2)	V
V_{il}	$I_{il}(max) = \pm\ 200\ \mu A$	0 (2)	0.6	V

ตารางที่ 2.4 แสดงคุณลักษณะทางไฟฟ้าของสัญญาณที่ขา *RST* ภายใต้การทำงานปกติ

หมายเหตุ : (1) สำหรับเครื่องอ่านการ์ดบนสภาวะเงื่อนไขที่ต่างกัน 2 กรณี

(2) ระดับแรงดันของสัญญาณที่ขา *CLK* ควรมีค่าอยู่ระหว่าง 0.3 โวลต์ถึง V_{cc}+0.3โวลต์

5. *ขา V_{cc} และ ขา GND*

ใช้รับไฟเลี้ยงและกราวด์ที่ป้อนมาจากเครื่องอ่าน โดยระดับแรงดันไฟเลี้ยงและกระแสก็ควรจะมีค่าเป็นตามตาราง ที่ 2.5 ด้วย

สัญลักษณ์	ค่าต่ำสุด	ค่าสูงสุด	หน่วย
V_{cc}	4.75	5.25	V
I_{cc}	-	200	mA

ตารางที่ 2.5 แสดงระดับแรงดันไฟเลี้ยงและกระแสที่ป้อนให้กับสมาร์ทการ์ดภายใต้การทำงานปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 มาตรฐานที่เกี่ยวข้อง

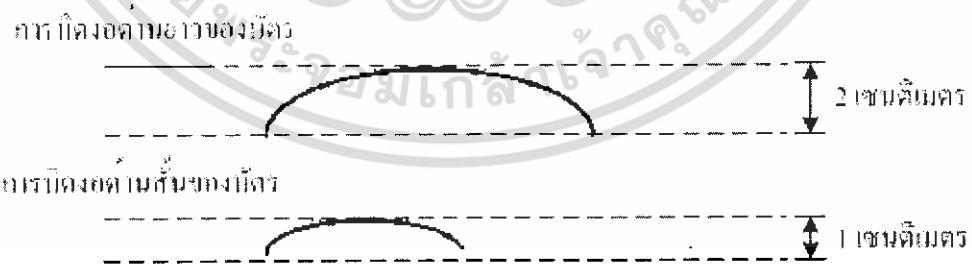
มาตรฐานที่เกี่ยวข้องกับสมาร์ทการ์ดมีด้วยกันหลายมาตรฐาน มาตรฐานหลายๆตัวมีเนื้อหาที่ซ้ำซ้อนกันเป็นผลให้สมาร์ทการ์ดเป็นเรื่องที่ค่อนข้างยุ่งยาก ถึงกระนั้นการพัฒนาระบบเพื่อใช้งานร่วมกับสมาร์ทการ์ดก็ยังยึดถือตามมาตรฐานเหล่านั้น มาตรฐานที่นำมาใช้กับสมาร์ทการ์ดนี้จะมีทั้งมาตรฐานที่มีอยู่แต่เดิมก่อนที่จะมีสมาร์ทการ์ด และมาตรฐานที่กำหนดขึ้นสำหรับสมาร์ทการ์ดโดยเฉพาะซึ่งอ้างอิงจากมาตรฐานบัตรพลาสติกและบัตรแถบแม่เหล็กที่มีอยู่แล้ว

2.3.1 มาตรฐาน ISO7816

เพื่อให้เกิดความเข้ากันได้ของสมาร์ทการ์ด จึงมีการกำหนดมาตรฐานของสมาร์ทการ์ดใน ชื่อของ ISO7816 เป็นการกำหนดในเรื่องของคุณลักษณะของบัตรพลาสติกที่จะนำมาทำเป็นสมาร์ทการ์ด โดยมาตรฐาน ISO7816 มีหัวข้อย่อยโดยแบ่งเป็น ISO7816-1 , ISO7816-2 , ISO7816-3 เป็นต้น

2.3.1.1 ISO7816-1 : มาตรฐานที่กำหนดด้วยเรื่องของคุณสมบัติทางกายภาพเบื้องต้นของสมาร์ทการ์ด ประกอบด้วย

1. ความทนทานต่อแสงและรังสีชนิดต่างๆ
2. ขนาดความหนาของชิพสมาร์ทการ์ด
3. ความทนทานต่อแรงกดดันของหน้าสัมผัส (ทนทานต่อแรงกด 1.5 นิวตันได้โดยไม่เสียหาย)
4. ค่าความต้านทานของหน้าสัมผัส (ไม่เกิน 0.5 โอห์ม ที่กระแส 0.5 ไมโครแอมป์ ถึง 300 มิลลิแอมป์)
5. ความทนทานต่อสนามแม่เหล็ก
6. ความทนทานต่อไฟฟ้าสถิต (1500 โวลต์ ประจุ 100 พิโกฟารัด ที่ 1500 โอห์ม)
7. ความทนทานต่อการบิดงอ เป็นจำนวน 30 ครั้งต่อนาทีโดยบัตรและชิพต้องไม่เกิดความเสียหาย ดังรูปที่ 2.7



รูปที่ 2.7 วิธีการทดสอบการบิดงอสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.2 ISO7816-2 : มาตรฐานที่กำหนดขนาดของหน้าสัมผัส และตำแหน่งของหน้าสัมผัสชิพสมาร์ทการ์ดบนบัตร ประกอบด้วย

1. ขนาดของหน้าสัมผัสของชิพสมาร์ทการ์ด
2. ตำแหน่งของหน้าสัมผัสบนบัตร

2.3.1.3 ISO7816-3 : มาตรฐานที่กำหนดคุณสมบัติทางไฟฟ้า และ โปรโตคอล ที่ใช้ในการสื่อสารกับชิพสมาร์ทการ์ด ประกอบด้วย

1. การอธิบายเกี่ยวกับหน้าที่การทำงานของแต่ละหน้าสัมผัส มีดังนี้

V_{cc} : แรงดันไฟบวกของแหล่งจ่ายกระแสไฟฟ้าที่ป้อนให้แก่ชิพ

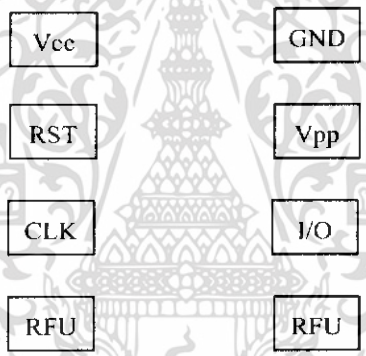
V_{pp} : แรงดันไฟฟ้าสำหรับการเขียนข้อมูลลงในชิพสมาร์ทการ์ด

GND : กราวน์ของแหล่งจ่ายกระแสไฟฟ้าที่ป้อนให้แก่ชิพ

RST : แรงดันไฟฟ้าสำหรับรีเซตชิพสมาร์ทการ์ด

I/O : อินพุต – เอาท์พุต สำหรับการส่งข้อมูลแบบอนุกรม

CLK : สัญญาณนาฬิกาสำหรับกำหนดจังหวะการรับ-ส่งข้อมูล



รูปที่ 2.8 หน้าที่การทำงานของแต่ละหน้าสัมผัส

2. คุณสมบัติทางไฟฟ้าของสัญญาณ V_{cc} , V_{pp} , RST , I/O และ CLK
3. ขั้นตอน และไทมิ่งไดอะแกรมสำหรับการเชื่อมต่อกับสมาร์ทการ์ด
4. โปรโตคอลที่ใช้ในการติดต่อสื่อสารกับสมาร์ทการ์ด

2.3.1.4 ISO7816-4 : มาตรฐานที่กำหนดรายละเอียดในเรื่องของชุดคำสั่งที่ใช้ในสมาร์ทการ์ด

2.3.1.5 ISO7816-5 : มาตรฐานที่กำหนดรายละเอียดในเรื่องของโครงสร้างไฟล์ในสมาร์ทการ์ด

2.3.1.6 ISO7816-6 : มาตรฐานที่กำหนดรายละเอียดในเรื่องของชื่อและโครงสร้างของไฟล์ในสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 องค์ประกอบต่างๆ ในการใช้งานสมาร์ทการ์ด

2.4.1 ตัวบัตรและตัวชิพ

บัตรและชิพสมาร์ทการ์ดเป็นส่วนแรกที่มักกล่าวถึงกัน เพราะสมาร์ทการ์ดมีหลากหลายรูปแบบหลากหลายการใช้งาน โดยหลักการแล้วสมาร์ทการ์ดเป็นเพียงบัตรพลาสติกฝังชิพ ไอซี ที่สามารถเก็บข้อมูล และประมวลผลข้อมูลได้เท่านั้น ผู้ออกแบบระบบมีหน้าที่นำสมาร์ทการ์ดมาใช้งานอย่างชาญฉลาดเหมาะสมตามประเภทของงาน และบริหารข้อมูลภายในสมาร์ทการ์ดให้เกิดความปลอดภัยสูงสุด

2.4.2 สมาร์ทการ์ดรีดเดอร์ (Smart Card reader)

สำหรับสมาร์ทการ์ดแล้วจำเป็นต้องมีอุปกรณ์สำหรับเชื่อมต่อกับชิพสมาร์ทการ์ดโดยเฉพาะซึ่งเรียกว่า สมาร์ทการ์ดรีดเดอร์ โดยภายในสมาร์ทการ์ดรีดเดอร์จะประกอบด้วยขาสำหรับเชื่อมสัญญาณกับหน้าสัมผัสบนชิพสมาร์ทการ์ด (Card Contact) หรือเป็นเสาอากาศรับส่งคลื่นวิทยุสำหรับสมาร์ทการ์ดแบบที่ไม่มีหน้าสัมผัส (Contactless) และหน่วยประมวลผลพร้อมหน่วยความจำสำหรับติดต่อสื่อสารกับชิพสมาร์ทการ์ดโดยตรง การสร้างสมาร์ทการ์ดรีดเดอร์ขึ้นใช้เองก็สามารถทำได้โดยนำเอาไมโครโปรเซสเซอร์ไมโครคอนโทรลเลอร์เบอร์ใดก็ได้มาประยุกต์ใช้ในการเชื่อมต่อกับสมาร์ทการ์ด แต่ปัจจุบันได้มีผู้ผลิตสมาร์ทการ์ดรีดเดอร์ออกมาขายในท้องตลาดหลากหลายรูปแบบซึ่งมีราคาย่อมเยาและใช้งานได้ง่ายกว่าการสร้างสมาร์ทการ์ดรีดเดอร์เองเสียอีก

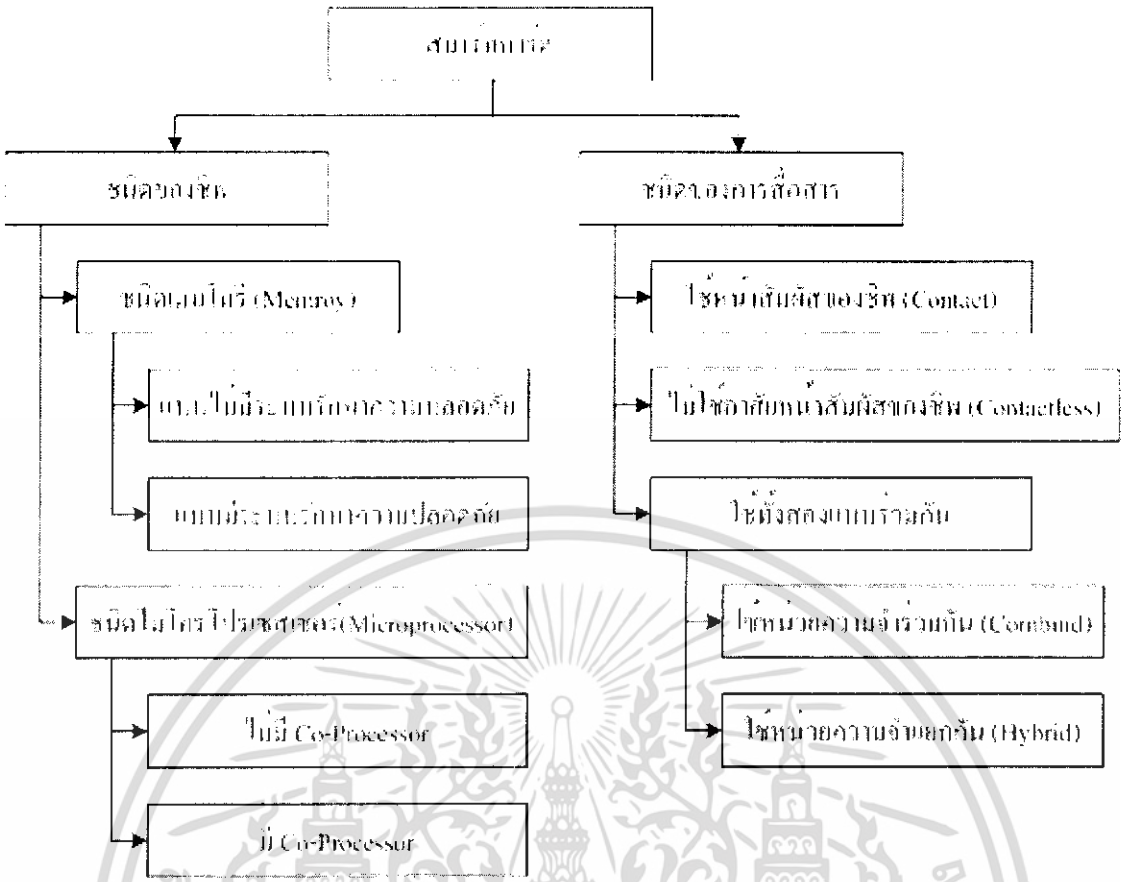
2.4.3 ซอฟแวร์

ในการใช้งานสมาร์ทการ์ดนอกจากตัวบัตรสมาร์ทการ์ด และสมาร์ทการ์ดรีดเดอร์แล้วยังต้องมีส่วนประกอบที่สำคัญอีกส่วนหนึ่งที่สำคัญ คือ ซอฟแวร์สำหรับจัดการข้อมูลในสมาร์ทการ์ด และซอฟแวร์สำหรับบริหารงานด้านบัตร หรืออาจเรียกได้อีกอย่างว่า ระบบ ฟรอนด์เอน (Font-End) ซึ่งระบบฟรอนด์เอน ในสมาร์ทการ์ดจะแตกต่างจากระบบในบัตรพลาสติก เนื่องจากสมาร์ทการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับ ฟรอนด์เอน ทุกครั้งที่ทำรายการเหมือนในระบบบัตรเครดิต ทำให้ระบบ ฟรอนด์เอนของสมาร์ทการ์ดมีเวลามากพอในการบริหารงานในด้านอื่นๆ หากต้องการติดต่อสื่อสารกับระบบ ฟรอนด์เอน จำเป็นต้องใช้สมาร์ทการ์ดรีดเดอร์ที่มีส่วนสำหรับการติดต่อสื่อสารไม่ว่าจะเป็น โมเดม(Modem) ระบบสื่อสารด้วยคลื่นวิทยุ ระบบสื่อสารอนุกรม RS-485/422 สำหรับการสื่อสารในบริเวณพื้นที่ให้บริการที่ไม่กว้างใหญ่นัก ฯลฯ เพื่อใช้สำหรับ รับ-ส่งข้อมูลระหว่าง ฟรอนด์เอน เมื่อจำเป็น

2.5 ชนิดของสมาร์ทการ์ด

เราสามารถแบ่งชนิดของสมาร์ทการ์ด โดยใช้ชนิดของชิพ และชนิดของการสื่อสาร เป็นเกณฑ์ในการแบ่ง ดังรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

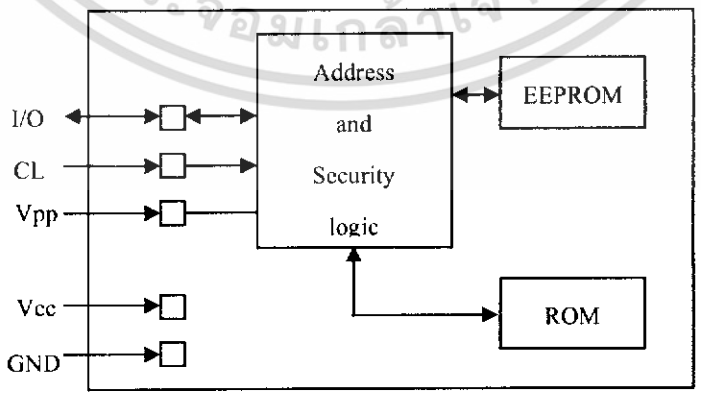


รูปที่ 2.9 การแบ่งสมาร์ทการ์ดตามชนิดของหน่วยความจำ และประเภทของหน้าสัมผัส

2.5.1 การแบ่งสมาร์ทการ์ดโดยใช้ชนิดของชิพเป็นเกณฑ์

2.5.1.1 เมมโมรีการ์ด (Memory Card) หรือ ซิงโครนัสการ์ด (Synchronous Card)

สมาร์ทการ์ดชนิด เมมโมรี หรืออีกชื่อหนึ่งคือ ซิงโครนัสการ์ด เนื่องจากสมาร์ทการ์ดชนิดนี้มีการรับ – ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิพ สมาร์ทการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วย ส่วนของวงจรสำหรับติดต่อสื่อสารภายนอก, หน่วยความจำข้อมูล, และหน่วยความจำสำหรับเก็บชุดคำสั่งของสมาร์ทการ์ด ดังรูปที่ 2.10

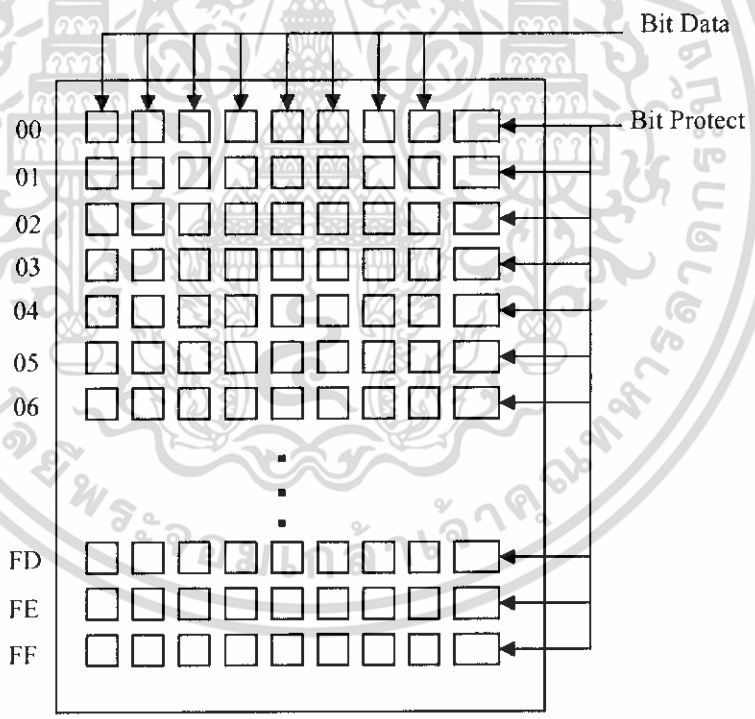


รูปที่ 2.10 บล็อกไดอะแกรมโครงสร้างภายในชิพสมาร์ทการ์ดชนิด เมมโมรีการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ทการ์ดที่เป็นพื้นฐานของสมาร์ทการ์ดในปัจจุบัน ก็คือ สมาร์ทการ์ดชนิด ที่ไม่มีระบบป้องกันความปลอดภัยของข้อมูล (Free Access Memory) สมาร์ทการ์ดชนิดนี้เปิดโอกาสให้อ่านหรือเขียนข้อมูลในแอดเดรสใดๆ ก็ได้ ตามชื่อของสมาร์ทการ์ดชนิดนี้ ไม่มีการป้องกันข้อมูลใดๆ ภายในสมาร์ทการ์ดชนิดนี้ จึงทำให้สมาร์ทการ์ดชนิดนี้มีความปลอดภัยต่ำที่สุด

นอกจากนี้สมาร์ทการ์ดชนิด เมมโมรี แบบธรรมดา ยังมีการใส่วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลลงไปด้วย ทำให้สามารถกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลได้ทุกไบต์ โดยสมาร์ทการ์ดที่มีวงจรป้องกันการอ่าน-เขียนชนิดนี้ถูกเรียกว่า PIN Protect Memory เนื่องจากการเข้าถึงข้อมูลจะต้องแสดงรหัสผ่านให้บัตรรับทราบก่อนจึงจะสามารถเข้าถึงข้อมูลได้ วงจรกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลจะมีบิตพิเศษที่มีชื่อว่า Bit Protect ซึ่งเป็นบิตข้อมูลที่ฝากไว้กับข้อมูลให้เป็นบิตที่ 9 แต่ไม่สามารถแก้ไขด้วยคำสั่งเขียนข้อมูลธรรมดาเพราะ Bit Protect ไม่ได้เป็นส่วนหนึ่งของข้อมูลจริงๆ ในการแก้ไข Bit Protect นี้จะสามารถทำการเปลี่ยนแปลงได้เพียงครั้งเดียวด้วยคำสั่งเฉพาะเท่านั้นเช่น หากต้องการบังคับไม่ให้ข้อมูลไบต์ใดไม่สามารถแก้ไขได้ก็ให้ทำการเคลียร์บิตที่ 9 ของข้อมูลไบต์นั้นๆ แต่สำหรับรหัสผ่านในการเข้าถึงข้อมูลสามารถเปลี่ยนแปลงได้แต่ต้องแสดงรหัสผ่านชุดเก่าให้บัตรได้รับทราบเสียก่อนจึงสามารถเปลี่ยนแปลงรหัสผ่านได้

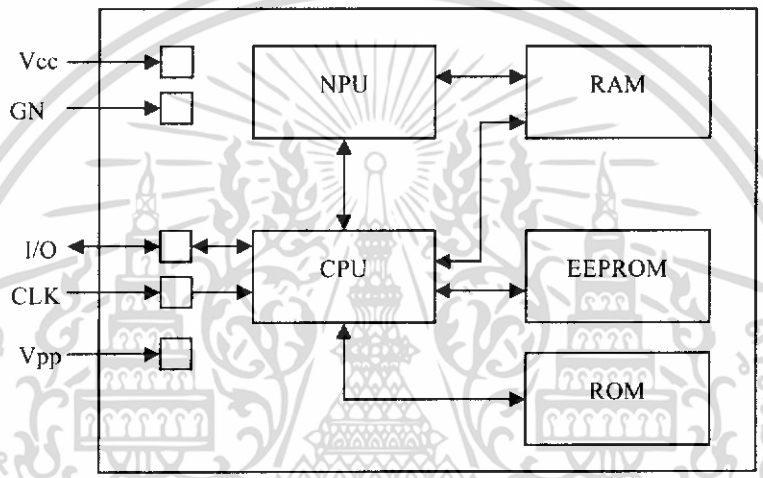


รูปที่ 2.11 ตัวอย่างหน่วยความจำของ เมมโมรีการ์ด ชนิด PIN Protect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.2 โปรเซสเซอร์การ์ด (Processor Card) หรือ อะซิงโครนัสการ์ด (Asynchronous Card)

สมาร์ทการ์ดชนิดนี้เป็นสมาร์ทการ์ดที่ได้รับการปรับปรุงจากสมาร์ทการ์ดชนิด เมมโมรี ด้วยการใส่เทคโนโลยีไมโครโปรเซสเซอร์เข้าไปในชิพ เพื่อให้ชิพสามารถประมวลผลข้อมูล และเพิ่มความปลอดภัยให้แก่ข้อมูลสูงขึ้น การที่ใส่ไมโครโปรเซสเซอร์ลงในชิพทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำสำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผลข้อมูล นอกจากนี้ยังมีการใส่ชิพประมวลผลทางคณิตศาสตร์ลงในชิพสมาร์ทการ์ดเพื่อช่วยในการประมวลผลข้อมูลด้วยอัลกอลิทึมสำหรับเข้า - ถอด รหัส ทำให้สมาร์ทการ์ดชนิด Processor มีความเร็วในการทำงานสูงกว่าสมาร์ทการ์ดชนิด เมมโมรีหลายเท่า



รูปที่ 2.12 บล็อกไดอะแกรมโครงสร้างภายในชิพสมาร์ทการ์ดชนิดโปรเซสเซอร์การ์ด

ในการรับ-ส่งข้อมูลให้กับสมาร์ทการ์ดชนิดนี้ จะใช้หน้าสัมผัสเดียวกับสมาร์ทการ์ดชนิดเมมโมรี โดยสัญญาณนาฬิกาที่ป้อน จะถูกใช้เป็นสัญญาณนาฬิกาแก่โปรเซสเซอร์ภายในสมาร์ทการ์ด ข้อมูลที่ รับ-ส่ง จึงไม่จำเป็นต้องสัมพันธ์กับสัญญาณนาฬิกาที่ป้อนให้แก่ชิพ เพียงกำหนดอัตราการรับ-ส่งข้อมูลเป็น 9600 บิต/วินาที ก็จะสามารถติดต่อกับโปรเซสเซอร์ของชิพได้แล้ว แต่การเข้าถึงข้อมูลไม่สามารถทำได้เหมือนอย่างในสมาร์ทการ์ดชนิด เมมโมรี การเข้าถึงข้อมูลต้องกระทำผ่านโปรเซสเซอร์ของสมาร์ทการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตามเพราะหน่วยความจำจะอยู่ในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว ข้อดีอย่างหนึ่งที่ไม่สามารถติดต่อกับหน่วยความจำในชิพโดยตรงก็คือ การลอบเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตแทบเป็นไปไม่ได้ ยกเว้นมีความบกพร่องในการกำหนดเงื่อนไขในการเข้าถึงข้อมูลที่เป็นความลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

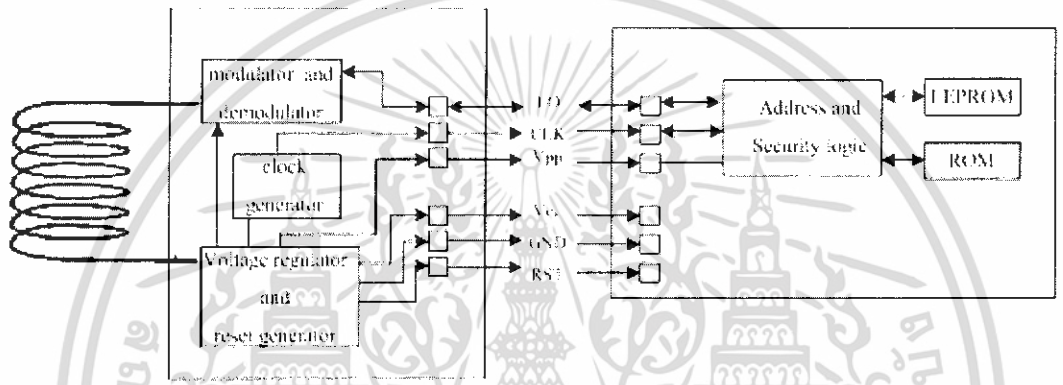
2.5.2 การแบ่งสมาร์ทการ์ดโดยใช้ชนิดของการสื่อสารเป็นเกณฑ์

2.5.2.1 ใช้หน้าสัมผัสของชิพ (Contact Card)

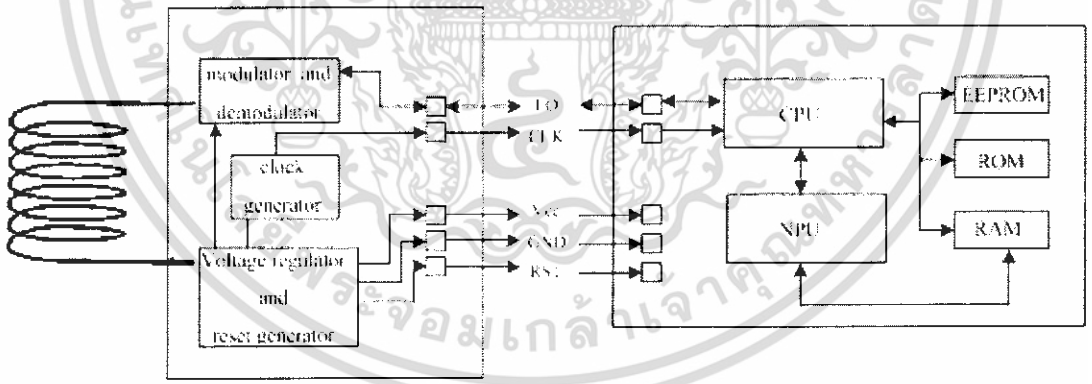
ประเภทนี้นั้นแบ่งได้เป็น เมมโมรีการ์ด และ โปรเซสเซอร์การ์ด ดังที่ได้กล่าวมาแล้วในข้างต้น

2.5.2.2 ไม่ใช้หน้าสัมผัสของชิพ (Contactless Card)

สมาร์ทการ์ดชนิดคอนแทกต์เลส เป็นสมาร์ทการ์ดที่ล้ำสมัยที่สุดในปัจจุบัน ด้วยเทคโนโลยีการสื่อสารผ่านคลื่นวิทยุ โดยการส่งคลื่นวิทยุคลื่นความถี่ 13.56 เมกะเฮิรตซ์ ที่ได้รับการมอดูเลตข้อมูลและส่งให้กับชิพสมาร์ทการ์ด ทางด้านชิพสมาร์ทการ์ดจะใช้ขดลวดเป็นเสารับ-ส่งสัญญาณ โดยเสา รับ-ส่งสัญญาณนี้จะเป็นขดลวดขนาดเล็กที่ถูกฝังลงในเนื้อบัตร



สมาร์ทการ์ดชนิด เมมโมรี แบบ คอนแทกต์เลส



สมาร์ทการ์ดชนิด โปรเซสเซอร์ แบบ คอนแทกต์เลส

รูปที่ 2.13 บล็อกไดอะแกรมโครงสร้างภายในชิพสมาร์ทการ์ดชนิดคอนแทกต์เลส

จากรูปที่ 2.13 จะเห็นว่าส่วนที่เพิ่มเข้ามาเป็นส่วนที่ใช้รับสัญญาณคลื่นวิทยุมาแบ่งเป็นสองส่วน โดนส่วนแรกจะถูกแปลงเป็นกระแสไฟฟ้าสำหรับป้อนชิพ และวงจรสร้างสัญญาณนาฬิกาให้สามารถทำงานได้ อีกส่วนหนึ่งจะถูกดีมอดูเลต เอาข้อมูลออกจากคลื่นวิทยุ และส่งให้แก่ชิพสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีกต่อหนึ่ง ส่วนการส่งข้อมูลกลับก็จะใช้กระแสไฟฟ้าที่ได้จากคลื่นวิทยุมาทำการมอดูเลต ข้อมูลและส่งกลับไปยังเสารับ-ส่งสัญญาณภายในเนื้อบัตร

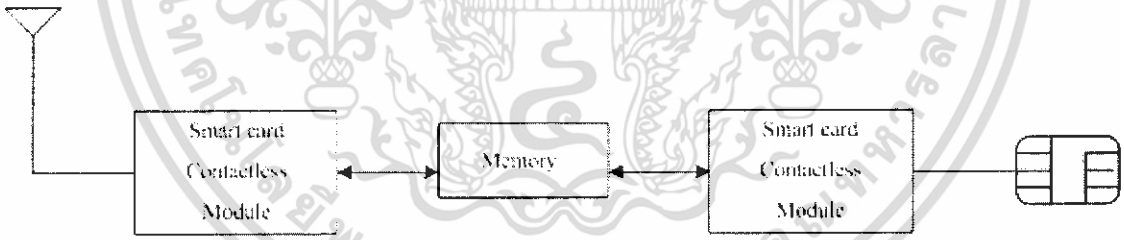
ด้วยเหตุนี้จึงทำให้ชิพสมาร์ทการ์ดชนิดคอนแทกต์เลส ถูกออกแบบให้ใช้กระแสไฟฟ้าที่ต่ำ เพราะกระแสไฟฟ้าที่ได้จากคลื่นวิทยุนั้นมีปริมาณเพียงเล็กน้อยไม่เพียงพอที่จะทำให้สมาร์ทการ์ดแบบธรรมดาสามารถทำงานได้ ฉะนั้นสมาร์ทการ์ดชนิดคอนแทกต์เลส รุ่นเก่าๆ จะไม่สามารถทำคำสั่งที่ซับซ้อนมากๆ เช่นคำสั่งในการเข้ารหัสข้อมูล หรือคำสั่งที่ต้องใช้เวลาในการประมวลผลมากๆ และระยะในการรับ-ส่งสัญญาณก็ไม่มากนัก

2.5.2.3 ใช้ทั้งสองแบบร่วมกัน (Contact and Contactless)

เป็นการรวมเอาสมาร์ทการ์ดชนิด คอนแทกต์ และ คอนแทกต์เลส บนบัตรใบเดียวกัน ซึ่งเป็นสมาร์ทการ์ดที่รวมเอาสมาร์ทการ์ดแบบที่หน้าสัมผัส กับสมาร์ทการ์ดแบบไม่มีหน้าสัมผัสเข้าด้วยกัน เพื่อความสะดวก และเพิ่มความรวดเร็วในการใช้งานอีกด้วย สมาร์ทการ์ดชนิดนี้สามารถแบ่งออกได้อีก 2 ประเภท คือ

2.5.2.3.1 ใช้หน่วยความจำร่วมกัน (Combind Card)

สมาร์ทการ์ดชนิดนี้เป็นการรวมเอาสมาร์ทการ์ดแบบ คอนแทกต์ และสมาร์ทการ์ดชนิด คอนแทกต์เลส เข้าด้วยกัน โดยใช้หน่วยความจำข้อมูลร่วมกันเพื่อให้การทำรายการที่จำเป็นต้องอยู่ภายใต้ระบบรักษาความปลอดภัย สามารถทำได้โดยผ่านทางหน้าสัมผัสที่มีไมโครโปรเซสเซอร์ควบคุมอยู่ และสามารถใช้งานทั่วไปได้อย่างสะดวกสบาย (Speed Pass) ผ่านทางคลื่นวิทยุ โดยมีโครงสร้างภายในตามรูปที่ 2.14



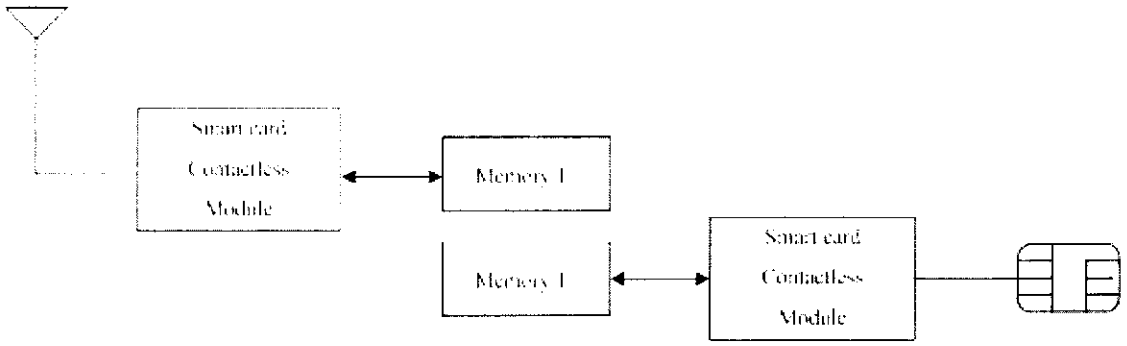
รูปที่ 2.14 โครงสร้างภายในของสมาร์ทการ์ดชนิด คอนแทกต์เลส แบบ คอมไบน์การ์ด

2.5.2.3.2 ใช้หน่วยความจำแยกกัน (Hybrid Card)

สมาร์ทการ์ดชนิดนี้มีลักษณะโครงสร้างภายในคล้ายกับคอมไบน์ แต่จะแตกต่างในเรื่องของหน่วยความจำข้อมูล โดยหน่วยความจำข้อมูลจะถูกแยกจากกันอย่างสิ้นเชิงระหว่าง คอนแทกต์ และ คอนแทกต์เลส เพื่อความสะดวกในการแยกใช้งาน โดยมีโครงสร้างภายในตามรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.15 โครงสร้างภายในของสมาร์ทการ์ดชนิด คอนแทกต์เลส แบบ ไฮบริดจ์การ์ด

2.6 การเชื่อมต่อชิพสมาร์ทการ์ด

การใช้งานสมาร์ทการ์ดการเชื่อมต่อ (Interface) มีความแตกต่างจากการเชื่อมต่อและใช้งานชิพไอซีธรรมดาทั่วไป ด้วยระบบสื่อสารแบบชิงเกิลบัส ซึ่งแตกต่างจากการสื่อสารแบบอนุกรมทั่วๆไป ทำให้การเชื่อมต่อกับสมาร์ทการ์ดเป็นเรื่องที่ไม่ง่ายนัก

สำหรับโปรแกรมใช้งานสมาร์ทการ์ด จะมีการแบ่งออกเป็น 2 ส่วน คือ

1.โปรแกรมส่วนที่ใช้ในการติดต่อสื่อสารกับชิพสมาร์ทการ์ด

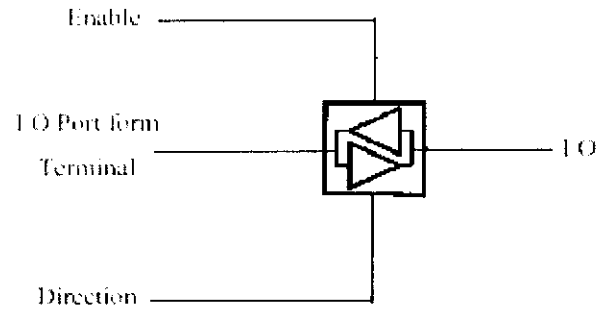
การเขียนโปรแกรมติดต่อสื่อสารกับสมาร์ทการ์ดจะใช้ภาษาระดับต่ำอย่างภาษาแอสแซมบลีเป็นส่วนใหญ่ เพราะภาษาระดับสูงหรือภาษาที่มีความใกล้เคียงภาษาแอสแซมบลีอย่างภาษาซีไม่สามารถทำงานได้ทันเวลา เนื่องจาก การ์ดไดร์ฟเวอร์ (Card Driver) ต้องทำการสื่อสารกับชิพสมาร์ทการ์ดโดยตรง (โปรแกรมติดต่อสื่อสารกับสมาร์ทการ์ดมีชื่อเรียกว่า Card Driver) โดย การ์ดไดร์ฟเวอร์ สำหรับสมาร์ทการ์ดแต่ละชนิดจะมีลักษณะการทำงานใกล้เคียงกันจะต่างกันก็เพียงวิธีการสื่อสารระหว่างสมาร์ทการ์ดชนิดเมมโมรี กับ โปรเซสเซอร์ เท่านั้นที่ไม่เหมือนกัน สำหรับการเลือกภาษาแอสแซมบลีมาใช้เขียนโปรแกรม ก็ต้องเลือกใช้ให้ตรงกับตระกูลของโปรเซสเซอร์ที่ใช้

2.โปรแกรมที่ใช้สำหรับบริหารข้อมูลและติดต่อกับผู้ใช้งานในเทอร์มินอล

ในส่วนนี้จะเขียนด้วยภาษาระดับสูงอย่างภาษาซี เหตุที่เลือกภาษาซีเพราะภาษาซีมีความสามารถในการประมวลผลที่รวดเร็ว สามารถจัดการกับข้อมูลในระดับบิตได้ดีกว่าภาษาระดับสูงภาษาอื่น และพัฒนาเป็นโปรแกรมได้ง่ายกว่า

ส่วนวงจรที่ใช้ในการเชื่อมต่อกับชิพสมาร์ทการ์ด จะเป็นวงจรที่ทำงานในลักษณะแบบเดียวกับวงจรดิจิตอลชนิด C-MOS ซึ่งสามารถทำงานได้ที่แรงดันไฟฟ้า 3-15 โวลต์ แต่ในสมาร์ทการ์ดมีการใช้งานเพียง 3 โวลต์เท่านั้น จากรูปที่ 2.16 จะเห็นได้ว่าเพียงทรานซิสเตอร์ตัวเดียวก็สามารถเชื่อมต่อกับชิพสมาร์ทการ์ดได้แล้ว วิธีการนี้ในสภาวะนิ่ง (Idle) ระดับลอจิก (แรงดันไฟฟ้า) ที่ขาเบสจะเป็น 0 ทำให้ทรานซิสเตอร์ไม่ทำงาน เมื่อมีข้อมูลถูกส่งมาที่ขาเบสกระแสที่เกิดขึ้นจากบิตข้อมูลจะเปลี่ยนจาก ลอจิก 0 เป็น ลอจิก 1 จะทำให้ทรานซิสเตอร์ทำงาน สัญญาณข้อมูลจะไปปรากฏที่ขาอีมิตเตอร์ ซึ่งสัญญาณที่ขานี้จะถูกส่งต่อให้กับชิพสมาร์ทการ์ดโดยตรง หรืออีกวิธีหนึ่งที่ง่ายกว่าการใช้ทรานซิสเตอร์ เราสามารถใช้วงจรลอจิก (C-MOS) มาต่อเป็นวงจรง่ายๆ ดังรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62557

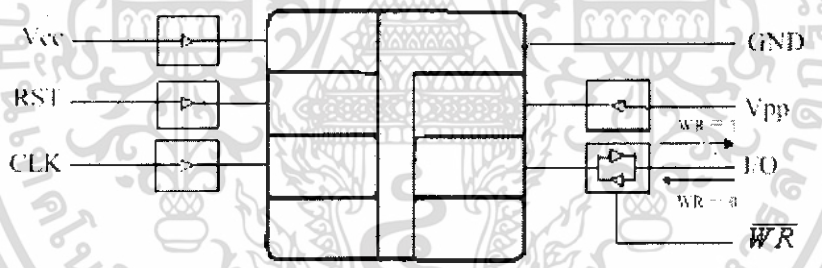


รูปที่ 2.16 ตัวอย่างวงจรเชื่อมต่อกับชิพสมาร์ทการ์ดด้วยวงจรลอจิก (C-MOS)

ในการเชื่อมต่อกับสมาร์ทการ์ดทั้งชนิดเมมโมรี และ โปรเซสเซอร์ จะมีความแตกต่างกันในเรื่องของสัญญาณนาฬิกา (CLK) และวิธีการรับ-ส่งข้อมูล (I/O) ซึ่งในที่นี้สามารถแยกวงจรออกเป็น 2 แบบ คือ

2.6.1 การเชื่อมต่อสมาร์ทการ์ดชนิดเมมโมรี (Synchronous card Interface)

การเชื่อมต่อหรือที่เราเรียกว่าการอินเตอร์เฟส(Interface) นั้น ในกรณีของสมาร์ทการ์ดชนิดเมมโมรี เราจะทำการสวิงสัญญาณนาฬิกาเอง เพื่อกำหนดจังหวะการส่งข้อมูลแต่ละบิต สำหรับข้อมูล ไอ/โอ เราสามารถใช้บิตข้อมูลแบบ 2 ทิศทางเพียงบิตเดียว โดยกำหนดจังหวะการส่งข้อมูลด้วยสัญญาณ $\overline{WR}$ ซึ่งมีลอจิกเป็น 0 เมื่อต้องการส่งข้อมูลให้แก่สมาร์ทการ์ด ดังรูปที่ 2.17



รูปที่ 2.17 ตัวอย่างวงจรเชื่อมต่อชิพสมาร์ทการ์ดชนิดเมมโมรี

การเชื่อมต่อกับสมาร์ทการ์ดแบบเมมโมรีนั้น จะแบ่งออกเป็น 2 ชนิด คือ การ์ดที่ไม่มีระบบป้องกันข้อมูล(Free Access Memory Card) และการ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล (Security Memory Card)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1.1 การเชื่อมต่อกับการ์ดที่ไม่มีระบบป้องกันข้อมูล (Free Access Memory Card)

การ์ดที่ไม่มีระบบป้องกันข้อมูล คือการ์ดที่สามารถจะถูกอ่านหรือเขียนข้อมูลได้อย่างอิสระ โดยทั่วไปแล้วการ์ดชนิดนี้จะมีลักษณะคล้ายกับหน่วยความจำที่สามารถเข้าถึงได้โดยการเชื่อมต่อแบบซิงโครนัส (Synchronous) ในรูปแบบของระบบบัส ไอสแควร์ซี (I^2C) โดยรูปแบบของคำสั่งที่ใช้ในการติดต่อและควบคุมการ์ดอาจมีความแตกต่างกันบ้างในแต่ละผู้ผลิต แต่ว่าพื้นฐานและหลักการที่ใช้โดยทั่วไปแล้วจะเหมือนกัน โดยการทำความเข้าใจในการติดต่อกับการ์ดที่ไม่มีระบบป้องกันข้อมูลนั้นจะมีคำศัพท์ที่เกี่ยวข้อง คือ

- *โปรโตคอลสื่อสารแบบ ไอสแควร์ซี (I^2C)* คือ รูปแบบการสื่อสารข้อมูลบนระบบบัสที่มีสายสัญญาณ 2 เส้น ได้แก่ สายเอสซีแอล (SCL) ซึ่งเป็นสายสัญญาณนาฬิกาอ้างอิงสำหรับกำกับจังหวะในการเขียนหรืออ่านข้อมูลแต่ละบิตของอุปกรณ์ใดๆ บนระบบบัส และสายเอสดีเอ (SDA) ซึ่งเป็นสายข้อมูล

- *การรับรู้ (Acknowledge)* ตามความหมายในระบบบัสแบบ ไอสแควร์ซี จะหมายถึงสัญญาณที่ตอบรับหลังจากการรับหรือส่งข้อมูลครบจำนวน 1 ไบต์ ก็จะมีการส่งบิตข้อมูลที่เรียกว่า บิตแอ็คโนเลจด์ (Acknowledge) ต่อท้ายไปด้วยเพื่อแสดงขอบเขตของข้อมูลและเพื่อลดความผิดพลาดที่อาจเกิดในการรับหรือส่งข้อมูลลง

- *สเลฟแอดเดรส (Slave Address)* คือข้อมูลที่จะระบุถึงชนิดและแอดเดรสของอุปกรณ์ที่เชื่อมอยู่ในระบบบัสรวมทั้งยังมีการระบุถึงการอ่านหรือรับข้อมูลผนวกอยู่ด้วย

-*อุปกรณ์แม่ข่าย (Master Device)* คือ อุปกรณ์ที่ทำหน้าที่ควบคุมกระบวนการทั้งหมดที่เกิดขึ้นบนบัส ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลกับอุปกรณ์ลูกข่าย (Slave Device) หรือสมาร์ทการ์ด โดยในที่นี้อุปกรณ์มาสเตอร์ก็คือเครื่องอ่านการ์ดนั่นเอง

- *เพจ (Page)* คือพื้นที่หน่วยความจำของสมาร์ทการ์ดแบบไม่มีระบบป้องกันข้อมูล โดยในแต่ละเพจประกอบด้วยข้อมูล 8 ไบต์

- *สภาวะเริ่มต้น (Start Condition)* คือ การจัดระดับแรงดันที่ขา ไอ/โอ และขาคล็อก (CLK) ของเครื่องอ่านการ์ด (อุปกรณ์มาสเตอร์) เพื่อเริ่มการสื่อสารข้อมูล

- *สภาวะสิ้นสุด (Stop Condition)* คือ การจัดระดับแรงดันที่ขาไอ/โอ และขาคล็อก ของเครื่องอ่านการ์ด (อุปกรณ์มาสเตอร์) เพื่อการหยุดการสื่อสารข้อมูล

-*กระบวนการส่งข้อมูล(Transmission)* คือ ข้อมูลสมบูรณ์ที่ถูกส่งจากอุปกรณ์แม่ข่ายไปยังอุปกรณ์ลูกข่าย โดยกระบวนการดังกล่าวจะเริ่มต้นด้วยการจัดสภาวะบนบัสทั้ง 2 เส้นให้สอดคล้องกับสภาวะเริ่มต้น ตามด้วยการส่งกลุ่มของบิตข้อมูล และจบด้วยสภาวะสิ้นสุด

2.6.1.2 การเชื่อมต่อกับการ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล (Security Memory Card)

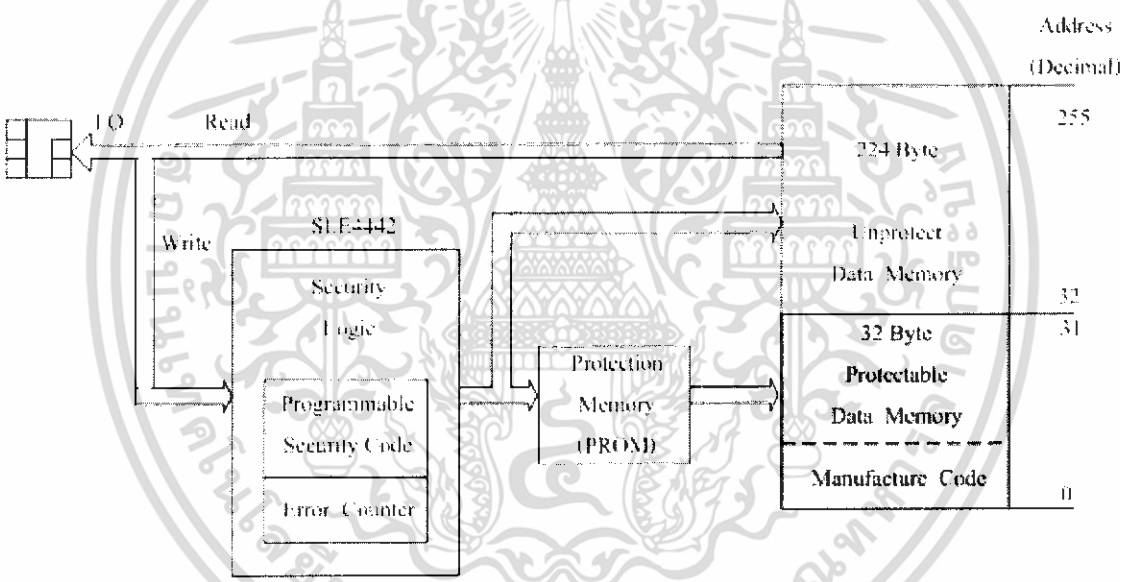
การ์ดที่มีระบบป้องกันความปลอกภัยข้อมูล คือ การ์ดที่การอ่านข้อมูลสามารถทำได้อย่างอิสระ แต่การเขียนข้อมูลจะไม่สามารถทำได้หากไม่มีรหัสผ่านหรือรหัสพีเอสซี(PSC:Programmable Security Code) ที่ถูกต้องวิธีการในลักษณะนี้ ช่วยให้ข้อมูลภายในสมาร์ทการ์ด ได้รับการปกป้องและมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความน่าเชื่อถือ รูปแบบการสื่อสารข้อมูลของการ์ดชนิดนี้ เป็นการสื่อสารข้อมูลแบบซิงโครนัส ตามมาตรฐาน ISO7816 ซึ่งรูปแบบคำสั่งที่ใช้ในการติดต่อและควบคุมการ์ด จะมีความแตกต่างกันไปในแต่ละผู้ผลิตแต่ละราย โดยส่วนใหญ่แล้วนิยมใช้กับบัตรสมาร์ทการ์ดเบอร์ SLE4442 ซึ่งผลิตโดยบริษัทซีเมนส์ เนื่องจากเป็นการ์ดที่มีคุณสมบัติในการรักษาความปลอดภัยข้อมูลอย่างครบถ้วนและสามารถหามาใช้งานได้ง่าย

คุณสมบัติโดยทั่วไปของ SLE4442

- ใช้หน่วยความจำ EEPROM 8 บิต ความจุข้อมูล 256 ไบต์
- ใช้รูปแบบของ ATR (Answer To Reset) ตามมาตรฐาน ISO7816-3
- อินเตอร์เฟสแบบซิงโครนัส ตามมาตรฐาน ISO7816
- ป้องกันการเขียนข้อมูลด้วยรหัสผ่านพีเอสซี (PSC)
- การลบและเขียนข้อมูลในแต่ละไบต์ใช้เวลาเพียง 2.5 มิลลิวินาที
- มีฟังก์ชันป้องกันข้อมูลในพื้นที่หน่วยความจำ 32 ไบต์แรก โดยสามารถจะกำหนดให้ข้อมูลที่เขียนลงไปยังพื้นที่ช่วงดังกล่าวถูกเขียนลงไปอย่างถาวรได้



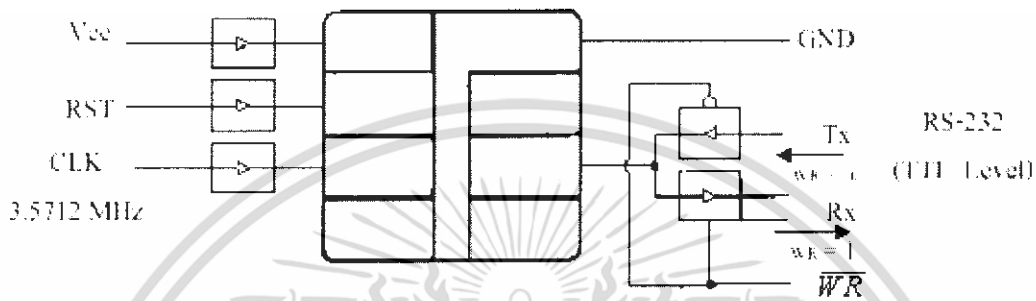
รูปที่ 2.18 บล็อกไดอะแกรมแสดงส่วนต่างๆของการ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล

จากรูปที่ 2.18 เป็นไดอะแกรมที่แสดงภาพรวมของ SLE4442 จะเห็นได้ว่าการอ่านข้อมูลจากหน่วยความจำนั้น เราสามารถจะอ่านข้อมูลออกมาได้โดยไม่ผ่านขั้นตอนของการป้อนรหัสพีเอสซี แต่สำหรับการเขียนข้อมูลแล้วเราจะต้องป้อนรหัส พีเอสซี ที่ถูกต้องเสียก่อน เพื่อเปิดลอจิกในการเขียนข้อมูลลงยังหน่วยความจำ นอกจากนั้นก็จะเห็นได้ว่าข้อมูล 4 ไบต์แรก เป็นข้อมูลของผู้ผลิต มีขนาด 4 ไบต์ พื้นที่ส่วนนี้ใช้เก็บข้อมูลของ เอทีอาร์ (ATR) โดยความหมายของข้อมูลที่อยู่ในพื้นที่ส่วนนี้แต่ละไบต์จะถูกกำหนดโดยบริษัทผู้ผลิตการ์ดแต่ละราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2 การเชื่อมต่อสมาร์ทการ์ดชนิดโปรเซสเซอร์ (Asynchronous card Interface)

ในกรณีที่เป็นสมาร์ทการ์ดโปรเซสเซอร์ วงจรเชื่อมต่อจะมีความยุ่งยากมากขึ้น สัญญาณนาฬิกาที่ป้อนให้แก่ชิพสมาร์ทการ์ด จะเป็นสัญญาณนาฬิกาแบบต่อเนื่อง (Continuous clock) ที่ความถี่ 3.5712 เมกะเฮิรตซ์ ตลอดเวลาที่ใช้งาน ส่วน ไอ/โอ จะใช้การรับ-ส่งข้อมูลแบบอะซิงโครนัส โดยใช้สัญญาณ $\overline{WR}$ เป็นตัวกำหนดช่วงการรับหรือส่งข้อมูล ซึ่งในสภาวะปกติจะรอรับข้อมูลตลอดเวลา ($\overline{WR}=1$) ดังรูปที่ 2.19



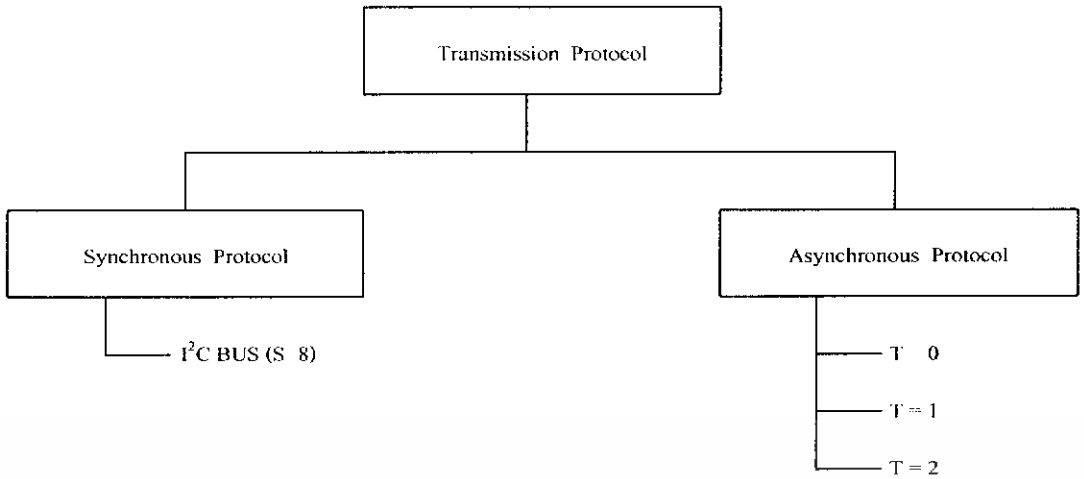
รูปที่ 2.19 ตัวอย่างวงจรเชื่อมต่อสมาร์ทการ์ดชนิดโปรเซสเซอร์

2.6.3 โปรโตคอลที่ใช้ในการสื่อสารกับสมาร์ทการ์ด

การสื่อสารกับชิพสมาร์ทการ์ด จำเป็นต้องใช้โปรโตคอลในการสื่อสารที่แตกต่างกันตามชนิดของสมาร์ทการ์ด ดังที่ทราบไปแล้วว่าสมาร์ทการ์ดมีอยู่ 2 ชนิดคือ เมมโมรี และ โปรเซสเซอร์ โปรโตคอลที่ใช้ในการสื่อสารกับสมาร์ทการ์ดจึงถูกแบ่งออกเป็น 2 ชนิดหลักเช่นกัน คือ การสื่อสารแบบอะซิงโครนัส (Asynchronous) และการสื่อสารแบบซิงโครนัส (Synchronous) ซึ่งแสดงได้ดังรูป 2.20 และในโครงงานนี้เราจะใช้การสื่อสารแบบซิงโครนัส ซึ่งโปรโตคอล ในการสื่อสารจะมีอยู่ 4 แบบ ได้แก่

- Reset and Answer-to-Reset
- Command Mode
- Outgoing Data Mode
- Processing Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

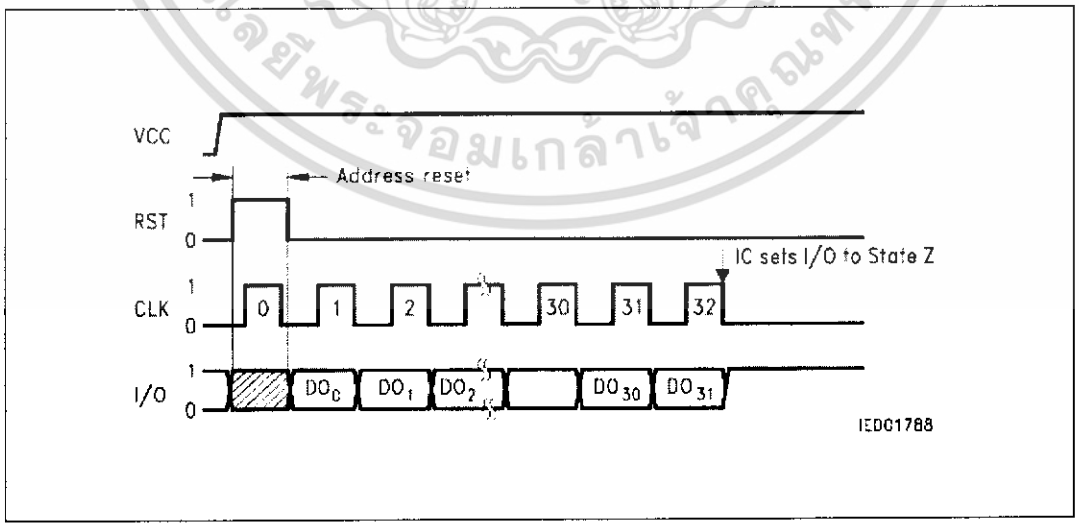


รูปที่ 2.20 รายละเอียดของการสื่อสารแบบซิงโครนัส และ การสื่อสารแบบ อะซิงโครนัส ในสมาร์ทการ์ด

2.6.3.1 Reset and Answer-to-Reset

Answer-to-Reset เกิดขึ้นตามมาตรฐาน ISO 7816-3 (ATR) ในการรีเซ็ตค่านั้นสามารถทำได้ตลอดเวลาในการทำงาน ในตอนแรก ค่าแอดเดสเคาท์เตอร์จะถูกกำหนดให้เป็นศูนย์จากสัญญาณนาฬิกา พร้อมกับข้อมูลบิตแรก (LSB) ที่ถูกส่งไปยังสมาร์ทการ์ด และในช่วงของสัญญาณนาฬิกา 31 ลูกต่อมา ข้อมูล 4 ไบท์ ที่อยู่ใน EEPROM จะถูกอ่านออกมา สัญญาณนาฬิกาลูกที่ 33 จะเปลี่ยนขา I/O ให้อยู่ในสถานะ high impedance และสิ้นสุดขั้นตอนการทำงานส่วน ATR

Answer-to-Reset (Hex)	Byte 1	Byte 2	Byte 3	Byte 4
	$DO_7 \dots DO_0$	$DO_{15} \dots DO_8$	$DO_{23} \dots DO_{16}$	$DO_{31} \dots DO_{24}$



รูปที่ 2.21 Reset และ Answer-to-Reset

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.3.2 Command Mode

หลังจากการทำงานในส่วน Answer-to-Reset แล้วก็จะหยุดรอคำสั่งต่อไป โดยในทุกคำสั่งจะต้องเริ่มต้นด้วย start condition ตามด้วยคำสั่งขนาดความยาว 3 ไบท์ และสัญญาณนาฬิกา และจบด้วย stop condition

- Start condition : ขอบขาลงของ I/O ในขณะที่สัญญาณนาฬิกามีสถานะ High
- Stop condition : ขอบขาขึ้นของ I/O ในขณะที่สัญญาณนาฬิกามีสถานะ High

หลังจากทำการรับคำสั่งแล้ว จะมีการทำงานที่เป็นไปได้ 2 แบบ คือ

- Outgoing data mode สำหรับการอ่านข้อมูล
- Processing mode สำหรับการเขียนและการลบข้อมูล

Outgoing Data Mode

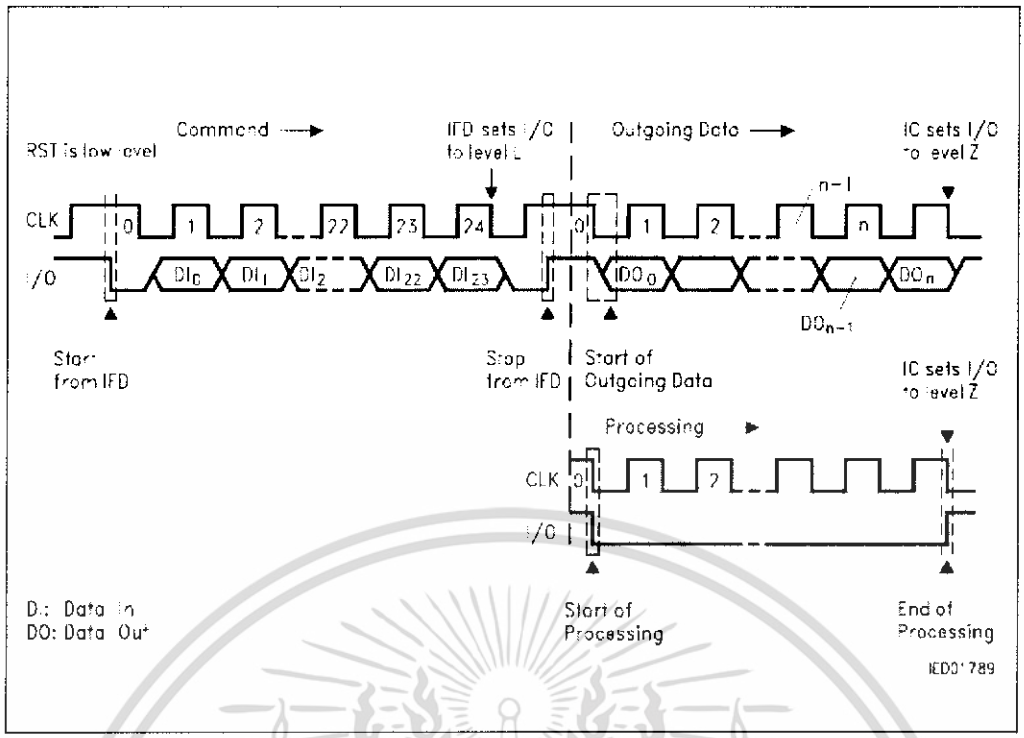
การทำงานในโหมดนี้ สมาร์ทการ์ดจะทำการส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ โดยข้อมูลบิทแรกจะถูกส่งออกมาหลังจากขอบขาลงของสัญญาณนาฬิกาลูกแรก หลังจากข้อมูลบิทสุดท้ายถูกส่งออกมาแล้ว ไมโครคอนโทรลเลอร์จะต้องส่งสัญญาณนาฬิกาอีก 1 ลูกไปยังสมาร์ทการ์ด เพื่อทำให้ ขา I/O อยู่ในสถานะ high impedance และทำให้สมาร์ทการ์ดเตรียมตัวรับคำสั่งต่อไป ระหว่างการทำงานในโหมดนี้ จะไม่มีการใช้งาน start condition และ stop condition

Processing Mode

การทำงานในโหมดนี้จะเป็นการทำงานภายในตัวสมาร์ทการ์ด ขอบขาลงของสัญญาณนาฬิกาลูกแรกจะทำให้ขา I/O อยู่ในสถานะ Low และสมาร์ทการ์ดจะต้องได้รับการป้อนสัญญาณนาฬิกาอย่างต่อเนื่องจนกระทั่งขา I/O จะกลับมาอยู่ในสถานะ High ระหว่างการทำงานในโหมดนี้ จะไม่มีการใช้งาน start condition และ stop condition

ระหว่างการทำงานในโหมดนี้ ขา RST จะต้องอยู่ในสถานะ Low การเซตให้ขา RST ให้อยู่ในสถานะ High ในขณะที่สัญญาณนาฬิกาอยู่ในสถานะ Low จะเป็นการยกเลิกการทำงาน และขา I/O จะกลับไปอยู่ในสถานะ High impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

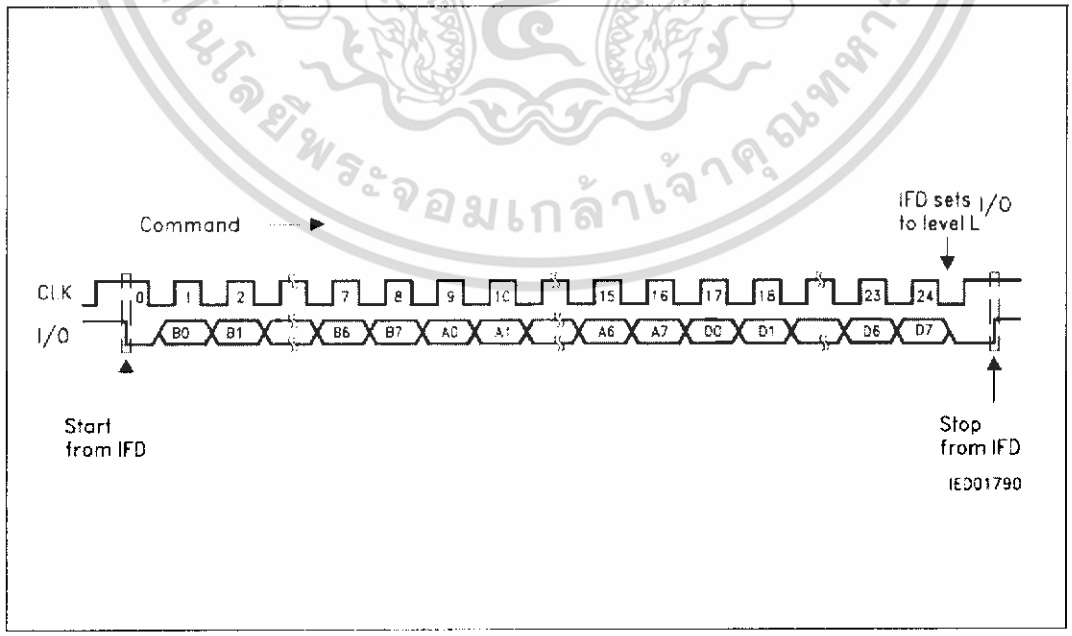


รูปที่ 2.22 Command mode, Outgoing data mode และ Processing mode

2.6.4 คำสั่งที่ใช้ในการติดต่อกับสมาร์ทการ์ด

รูปแบบของคำสั่งแต่ละคำสั่งจะมีขนาด 3 ไบท์ โดยในการส่งข้อมูล บิท LSB จะถูกส่งออกไปเป็นลำดับแรก

MSB			Control				LSB	MSB			Address				LSB	MSB			Data				LSB
B7	B6	B5	B4	B3	B2	B1	B0	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0



รูปที่ 2.23 Command mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสมาร์ทการ์ดเบอร์ SLE4442 จะมีชุดคำสั่งดังตารางที่ 2.6

Byte 1 Control								Byte 2 Address	Byte 3 Data	Operation	Mode
B7	B6	B5	B4	B3	B2	B1	B0	A7-A0	D7-D0		
0	0	1	1	0	0	0	0	address	no effect	READ MAIN MEMORY	outgoing data
0	0	1	1	1	0	0	0	address	input data	UPDATE MAIN MEMORY	processing
0	0	1	1	0	1	0	0	no effect	no effect	READ PROTECTION MEMORY	outgoing data
0	0	1	1	1	1	0	0	address	input data	WRITE PROTECTION MEMORY	processing

ตารางที่ 2.6 ชุดคำสั่งของสมาร์ดการ์ดเบอร์ SLE4442

2.6.4.1 การอ่านข้อมูลในหน่วยความจำหลัก (Main memory)

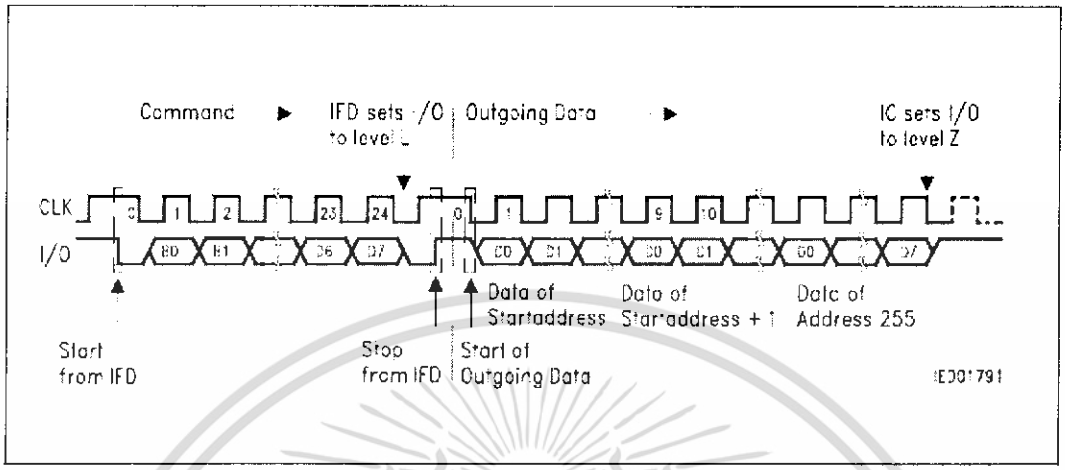
กระบวนการนี้จะเป็นการอ่านข้อมูลที่ถูกบรรจุอยู่ในหน่วยความจำหลักออกมา โดยข้อมูลบิท LSB จะถูกอ่านออกมาเป็นลำดับแรก และไมโครคอนโทรลเลอร์จะต้องป้อนสัญญาณนาฬิกาให้กับสมาร์ทการ์ดอย่างต่อเนื่องจนกว่ากระบวนการอ่านข้อมูลจะเสร็จสิ้น

Address (decimal)	Main Memory	Protection Memory	Security Memory (only SLE 4442)
255	Data Byte 255 (D7 ... D0)	–	–
:	:	–	–
32	Data Byte 32 (D7 ... D0)	–	–
31	Data Byte 31 (D7 ... D0)	Protection Bit 31 (D31)	–
:	:	:	–
3	Data Byte 3 (D7 ... D0)	Protection Bit 3 (D3)	Reference Data Byte 3 (D7 ... D0)
2	Data Byte 2 (D7 ... D0)	Protection Bit 2 (D2)	Reference Data Byte 2 (D7 ... D0)
1	Data Byte 1 (D7 ... D0)	Protection Bit 1 (D1)	Reference Data Byte 1 (D7 ... D0)
0	Data Byte 0 (D7 ... D0)	Protection Bit 0 (D0)	Error Counter

ตารางที่ 2.7 หน่วยความจำหลักในสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	0	0	0	0	Address	No effect
Hexadecimal	30_H								00_H....FF_H	No effect



รูปที่ 2.24 รูปแบบคำสั่งที่ใช้ในกระบวนการอ่านข้อมูลในหน่วยความจำหลัก

2.6.4.2 การอ่านข้อมูลในหน่วยความจำป้องกัน (Protection memory)

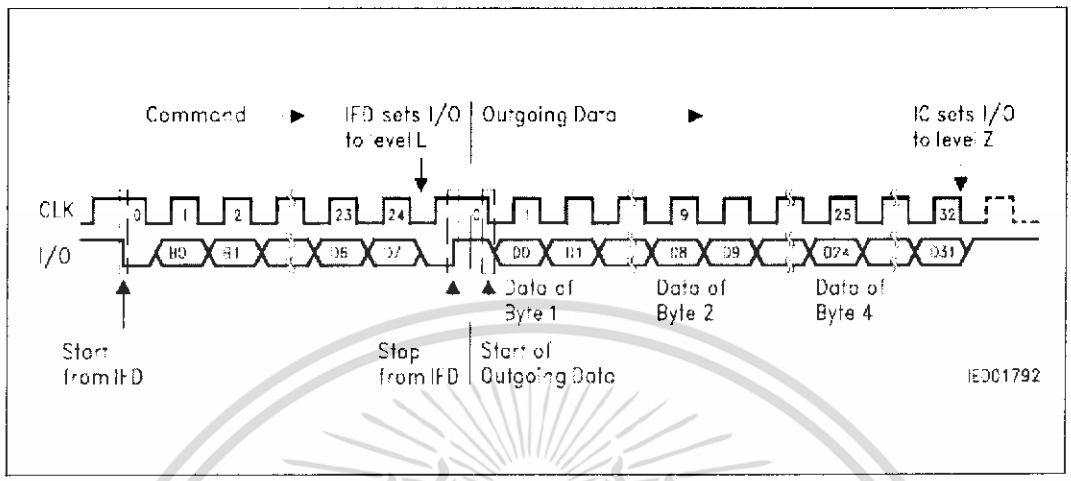
กระบวนการนี้จะเป็นการอ่านบิทป้องกัน (Protection bit) จำนวน 32 บิทออกมา โดยจะต้องมีการป้อนสัญญาณนาฬิกาอย่างต่อเนื่องจำนวน 32 ลูก และหลังจากสัญญาณนาฬิกาลูกสุดท้าย ขา I/O จะถูกเซตให้อยู่ในสถานะ high impedance โดยบิทป้องกันนี้จะแสดงว่าข้อมูลในหน่วยความจำหลักสามารถแก้ไขได้หรือไม่

Address (decimal)	Main Memory	Protection Memory	Security Memory (only SLE 4442)
255	Data Byte 255 (D7 D0)	–	–
:	:	–	–
32	Data Byte 32 (D7 ... D0)	–	–
31	Data Byte 31 (D7 ... D0)	Protection Bit 31 (D31)	–
:	:	:	–
3	Data Byte 3 (D7 ... D0)	Protection Bit 3 (D3)	Reference Data Byte 3 (D7 ... D0)
2	Data Byte 2 (D7 ... D0)	Protection Bit 2 (D2)	Reference Data Byte 2 (D7 ... D0)
1	Data Byte 1 (D7 ... D0)	Protection Bit 1 (D1)	Reference Data Byte 1 (D7 ... D0)
0	Data Byte 0 (D7 ... D0)	Protection Bit 0 (D0)	Error Counter

ตารางที่ 2.8 หน่วยความจำป้องกันในสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	0	1	0	0	No effect	No effect
Hexadecimal	34_H								No effect	No effect

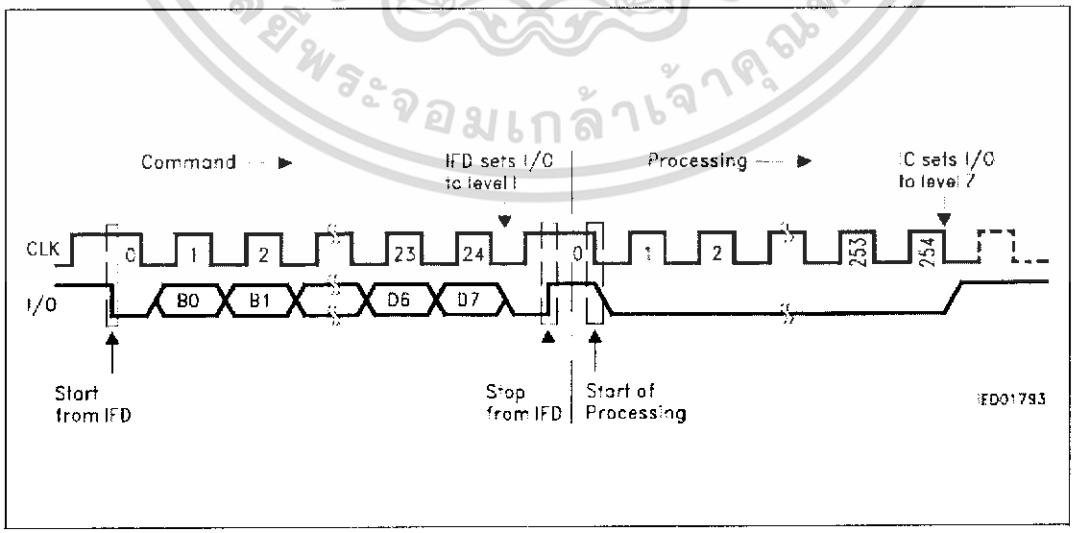


รูปที่ 2.25 รูปแบบคำสั่งที่ใช้ในกระบวนการอ่านข้อมูลในหน่วยความจำป้องกัน

2.6.4.3 การเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก

กระบวนการนี้จะเป็นการเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก โดยไมโครคอนโทรลเลอร์จะต้องส่งข้อมูลใหม่ให้แก่สมาร์ทการ์ด และสมาร์ทการ์ดจะใช้เวลาในการเปลี่ยนแปลงข้อมูลประมาณ 5 ms ที่ความถี่ของสัญญาณนาฬิกา 50 kHz

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	0	0	0	Address	Input data
Hexadecimal	38_H								00_H....FF_H	Input data



รูปที่ 2.26 รูปแบบคำสั่งที่ใช้ในกระบวนการเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าข้อมูลในหน่วยความจำหลักตำแหน่งที่ต้องการเปลี่ยนแปลงนั้นมีการป้องกันอยู่ ขา I/O จะถูกเซตให้อยู่ในสถานะ high หลังจากสัญญาณนาฬิกาลูกที่สอง

2.6.4.4 การเขียนข้อมูลลงในหน่วยความจำป้องกัน

กระบวนการนี้จะมีการเปรียบเทียบระหว่างข้อมูลที่ส่งไปยังสมาร์ทการ์ดและข้อมูลที่อยู่ในหน่วยความจำหลัก ถ้าข้อมูลทั้งสองตรงกัน บิทป้องกันจะถูกเซต ซึ่งจะทำให้ข้อมูลที่อยู่ในหน่วยความจำหลักไม่สามารถแก้ไขได้ แต่ถ้าหากข้อมูลทั้งสองไม่ตรงกัน บิทป้องกันก็จะไม่ถูกเซต โดยเวลาในกระบวนการนี้จะเหมือนกับการเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	1	0	0	Address	Input data
Hexadecimal	$3C_H$								$00_H...1F_H$	Input data

รูปที่ 2.27 รูปแบบคำสั่งที่ใช้ในกระบวนการเขียนข้อมูลลงในหน่วยความจำป้องกัน

2.6.4.5 การอ่านข้อมูลในซีเคียวริตี้เมมโมรี่ (Security memory)

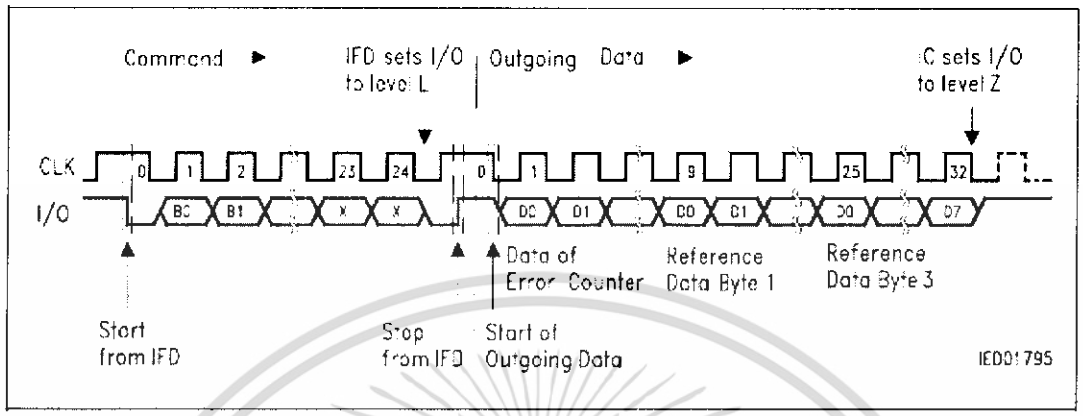
กระบวนการนี้จะมีลักษณะคล้ายกับการอ่านข้อมูลในหน่วยความจำป้องกัน โดยจะทำการอ่านข้อมูลจำนวน 4 ไบท์ในซีเคียวริตี้เมมโมรี่ออกมา และต้องมีการป้อนสัญญาณนาฬิกาให้กับสมาร์ทการ์ดอย่างต่อเนื่องจำนวน 32 ลูก และหลังจากสัญญาณนาฬิกาลูกสุดท้าย ขา I/O จะถูกเซตให้อยู่ในสถานะ high impedance โดยก่อนกระบวนการอ่านข้อมูลในซีเคียวริตี้เมมโมรี่นี้จะต้องมีการตรวจสอบ PSC (PSC verification) ว่าถูกต้องหรือไม่ ถ้าไม่ถูกต้อง ข้อมูลในซีเคียวริตี้เมมโมรี่จะไม่สามารถอ่านออกมาได้ โดยในช่วงของข้อมูล ขา I/O จะอยู่ในสถานะ Low

Address (decimal)	Main Memory	Protection Memory	Security Memory (only SLE 4442)
255	Data Byte 255 (D7 ... D0)	–	–
:	:	–	–
32	Data Byte 32 (D7 ... D0)	–	–
31	Data Byte 31 (D7 ... D0)	Protection Bit 31 (D31)	–
:	:	:	–
3	Data Byte 3 (D7 ... D0)	Protection Bit 3 (D3)	Reference Data Byte 3(D7 ... D0)
2	Data Byte 2 (D7 ... D0)	Protection Bit 2 (D2)	Reference Data Byte 2(D7 ... D0)
1	Data Byte 1 (D7 ... D0)	Protection Bit 1 (D1)	Reference Data Byte 1(D7 ... D0)
0	Data Byte 0 (D7 ... D0)	Protection Bit 0 (D0)	Error Counter (0,0,0,0,0,D2,D1,D0)

ตารางที่ 2.9 ซีเคียวริตี้เมมโมรี่ในสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	0	0	0	1	No effect	No effect
Hexadecimal	31_H								No effect	No effect



รูปที่ 2.28 รูปแบบคำสั่งที่ใช้ในกระบวนการอ่านข้อมูลในซีเคียวริตี้เมมโมรี่

2.6.4.6 การเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่

ก่อนกระบวนการเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่นี้จะต้องมีการตรวจสอบ PSC (PSC verification) ว่าถูกต้องหรือไม่ โดยการแก้ไขจะสามารถทำได้ถ้าการตรวจสอบ PSC ถูกต้อง แต่ถ้าหากการตรวจสอบไม่ถูกต้อง ตัวนับความผิดพลาด (Error counter) จะทำงาน และเปลี่ยนค่าบิทจาก 1 เป็น 0 โดยเวลาในกระบวนการนี้จะเหมือนกับการเปลี่ยนแปลงข้อมูลในหน่วยความจำหลัก

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	0	0	1	Address	Input data
Hexadecimal	39_H								$00_H...03_H$	Input data

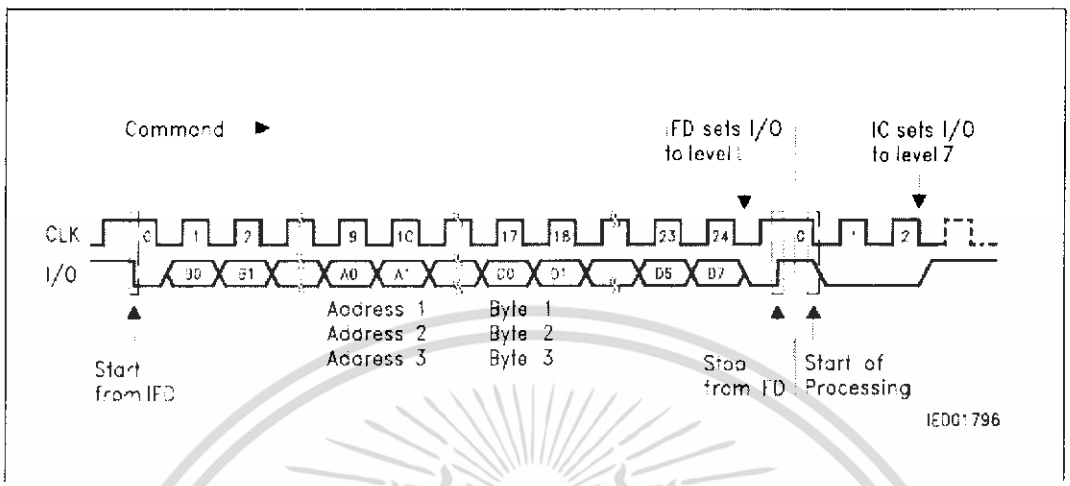
รูปที่ 2.29 รูปแบบคำสั่งที่ใช้ในกระบวนการเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่

2.6.4.7 การเปรียบเทียบข้อมูลที่ใช้ในการตรวจสอบ

กระบวนการนี้จะใช้ร่วมกับการเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่ ในกระบวนการตรวจสอบ PSC (PSC verification) โดยจะทำการเปรียบเทียบข้อมูลที่เข้ามากับข้อมูลอ้างอิงที่อยู่ในซีเคียวริตี้เมมโมรี่ และระหว่างกระบวนการนี้จะต้องมีการป้อนสัญญาณนาฬิกาให้สมาร์ทการ์ดอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	0	0	1	1	Address	Input data
Hexadecimal	33_H								$00_H...03_H$	Input data



รูปที่ 2.30 รูปแบบคำสั่งที่ใช้ในกระบวนการเปรียบเทียบข้อมูลที่ใช้ในการตรวจสอบ

2.6.5 การตรวจสอบรหัสป้องกัน (Programmable Security Code: PSC)

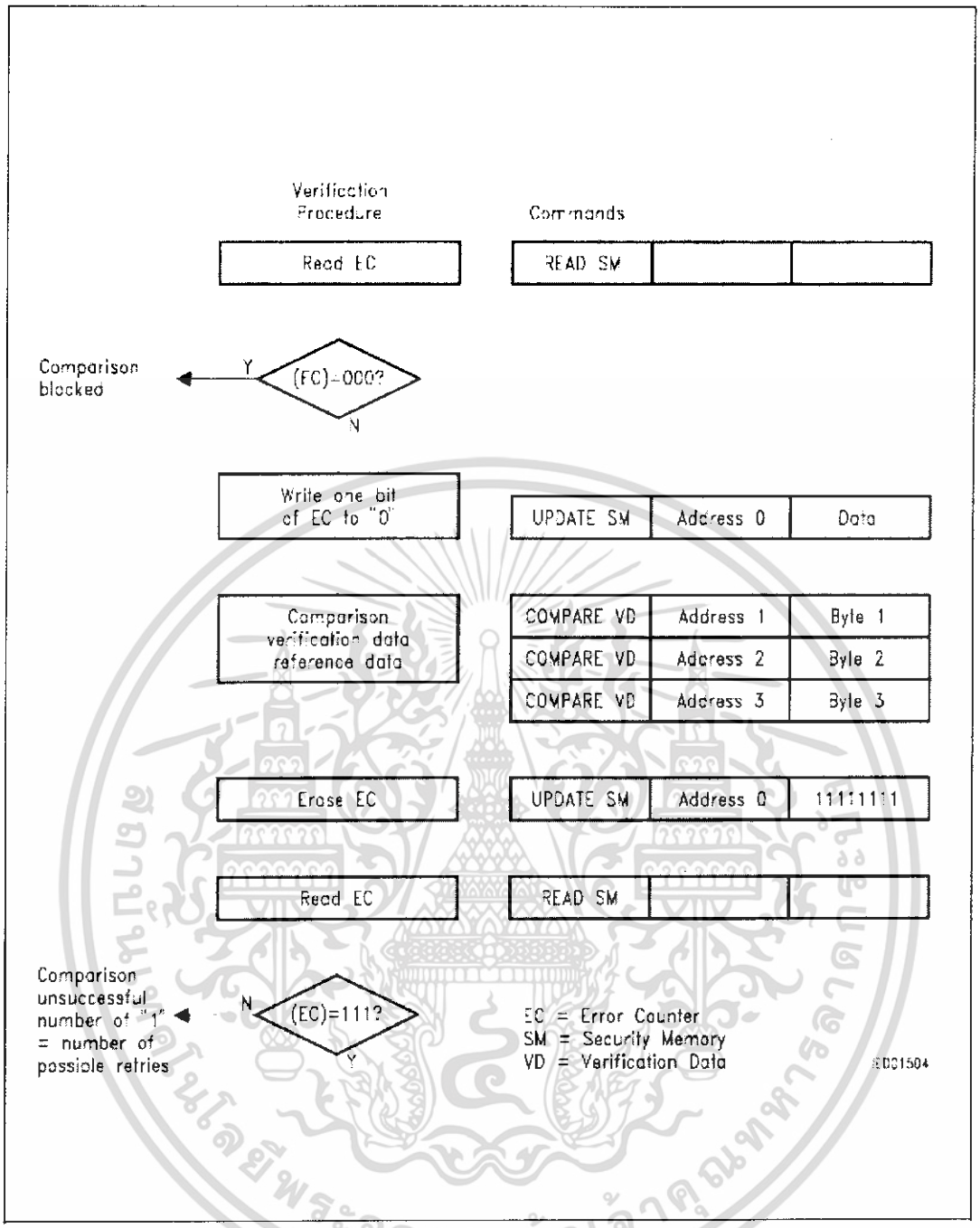
ในการเปลี่ยนแปลงข้อมูลต่าง ๆ ในสมาร์ทการ์ดนั้นจะต้องมีการตรวจสอบรหัสป้องกันก่อน โดยถ้าหากไม่มีการตรวจสอบรหัสป้องกันก่อน ก็จะไม่สามารถเขียนหรือลบข้อมูลในสมาร์ทการ์ดได้ ถ้าการตรวจสอบนี้ไม่ถูกต้อง บิทนับค่าความผิดพลาด (Error counter bit) จะเปลี่ยนจาก 1 เป็น 0

ในขั้นตอนแรกบิทนับค่าความผิดพลาดจะต้องถูกเปลี่ยนจาก 1 เป็น 0 โดยกระบวนการเปลี่ยนแปลงข้อมูลในซีเคียวริตี้เมมโมรี่ ดังรูปที่ 2.29 ตามด้วยการเปรียบเทียบข้อมูลที่ใช้ในการตรวจสอบ ถ้ากระบวนการตรวจสอบรหัสป้องกันนี้ถูกต้อง ก็จะสามารถลบค่าของตัวนับค่าความผิดพลาดได้ ซึ่งโดยปกติจะไม่ถูกลบโดยอัตโนมัติ และสามารถเขียนหรือลบข้อมูลในหน่วยความจำทั้งหมดในสมาร์ทการ์ดได้ ตราบเท่าที่ยังป้อนไฟเลี้ยงให้แก่สมาร์ทการ์ด

Command	Control	Address	Data	Remark
	B7...B0	A7...A0	D7...D0	
Read security Memory	31_H	No effect	No effect	Check Error Counter
Update Security Memory	39_H	00_H	Input data	Write free bit in Error Counter input data: 0000 0ddd binary
Compare Verification Data	33_H	01_H	Input data	Reference Data Byte 1
Compare Verification Data	33_H	02_H	Input data	Reference Data Byte 2
Compare Verification Data	33_H	03_H	Input data	Reference Data Byte 3
Update Security Memory	39_H	00_H	FF_H	Erase Error Counter
Read Security Memory	31_H	No effect	No effect	Check Error Counter

ตารางที่ 2.10 ชุดคำสั่งที่ใช้ในกระบวนการตรวจสอบรหัสป้องกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 กระบวนการตรวจสอบรหัสป้องกัน (PSC verification)

2.6.6 Reset Mode

พาวเวอร์ออนรีเซท (Power on Reset)

ในตอนเริ่มต้น หลังจากเสียบบัตร และป้อนไฟเลี้ยงให้กับบัตรแล้ว จะต้องทำการรีเซทบัตรก่อนการอ่านหรือเปลี่ยนแปลงข้อมูลใด ๆ ในบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

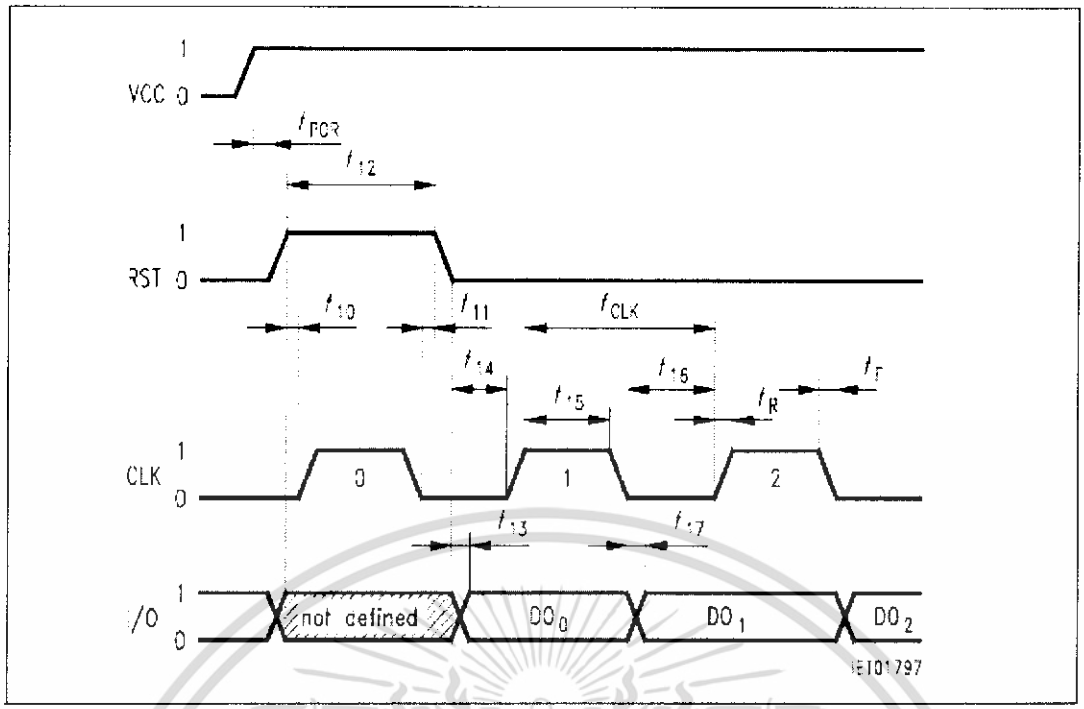
2.6.7 การยกเลิกการทำงาน (Break)

ถ้าขา RST ถูกเซตให้อยู่ในสถานะ high ในขณะที่ CLK อยู่ในสถานะ Low การทำงานทุกอย่างจะถูกยกเลิก และขา I/O จะกลับไปอยู่ในสถานะ high impedance และสมาร์ทการ์ดจะอยู่ในสภาวะที่พร้อมจะรับคำสั่งต่อไป

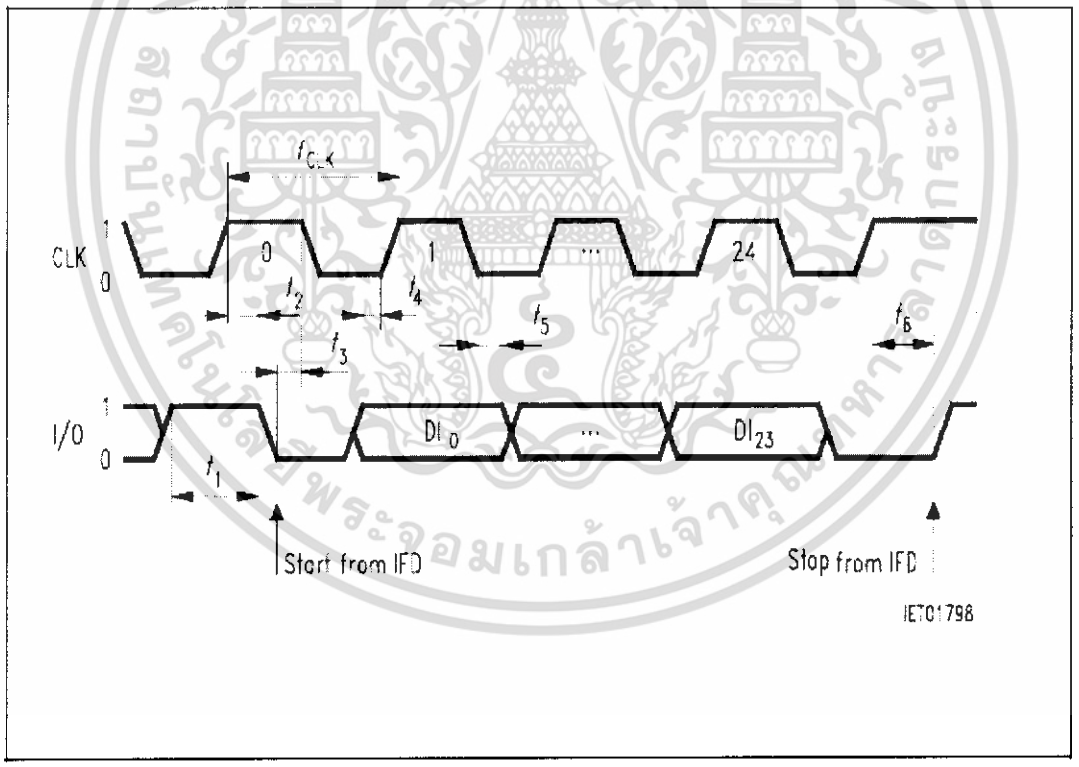
Parameter	Symbol	Limit Values			Unit	Test Condition
		min.	typ.	max.		
RST High to CLK Setup time	t_{10}	4			µs	
CLK Low to RST Hold time	t_{11}	4			µs	
RST High time (address reset)	t_{12}	20	50		µs	
RST Low to I/O Valid time	t_{13}			2.5	µs	
RST Low to CLK Setup time	t_{14}	4			µs	
CLK Frequency	f_{CLK}	7		50	kHz	
CLK Rise time	t_R			1	µs	
CLK Fall time	t_F			1	µs	
CLK High time	t_{15}	9			µs	
CLK Low time	t_{16}	9			µs	
CLK Low to I/O Valid time	t_{17}			2.5	µs	
Reset time for Break	t_{18}	5			µs	
RST High to I/O Clear time (Break)	t_{19}	2.5			µs	
I/O High time (Start Condition)	t_1	10			µs	
CLK High to I/O Hold time	t_2	4			µs	
I/O Low to CLK Hold time (Start Condition)	t_3	4			µs	
I/O Setup to CLK High time	t_4	1			µs	
CLK Low to I/O Hold time	t_5	1			µs	
CLK High to I/O Clear time (Stop Condition)	t_6	4			µs	
CLK Low to I/O Valid time	t_7			2.5	µs	
CLK Low to I/O Valid time	t_8			2.5	µs	
CLK Low to I/O Clear time	t_9			2.5	µs	
Erase time	t_{ER}	2.5			ms	f_{CLK} = 50 kHz
Write time	t_{WR}	2.5			ms	f_{CLK} = 50 kHz
Power on reset time	t_{POR}			100	µs	

ตารางที่ 2.11 ช่วงเวลาของสัญญาณต่าง ๆ ที่ใช้ในการติดต่อกับสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

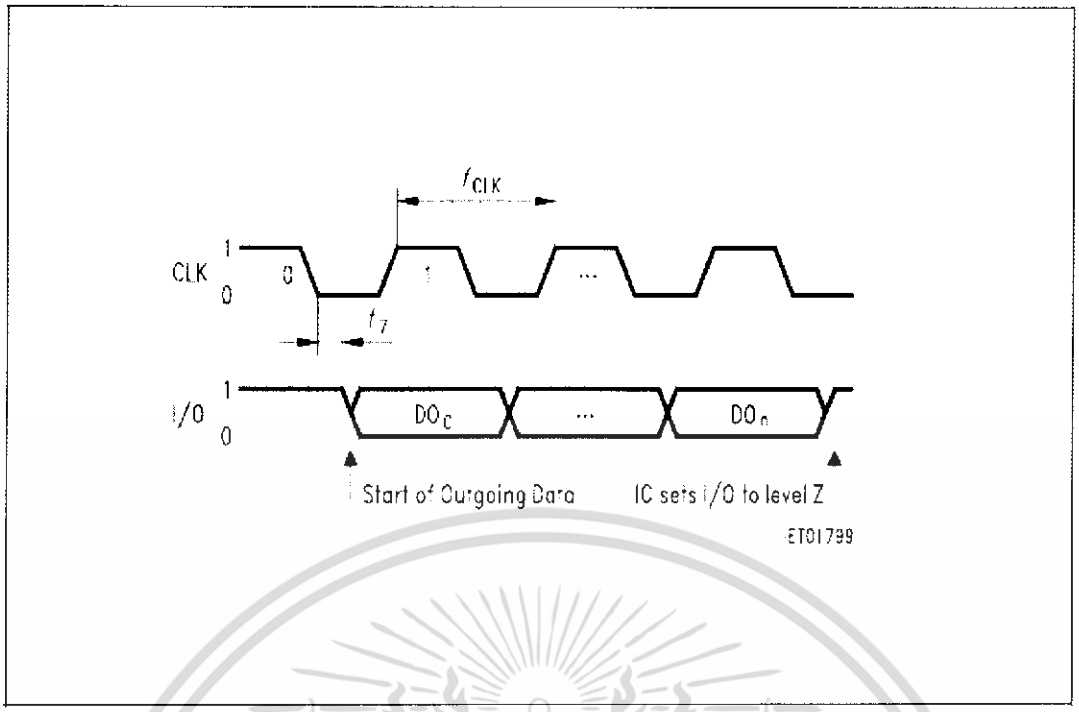


รูปที่ 2.32ไทม์มิ่งไดอะแกรมของสัญญาณ Reset และ Answer-to-Reset

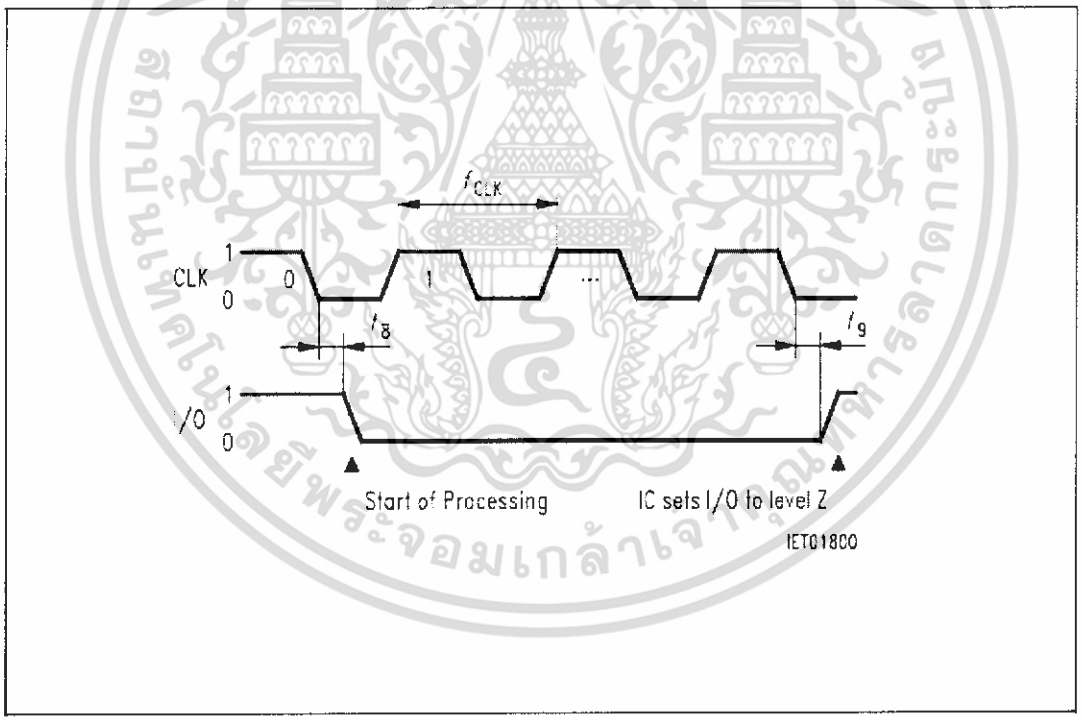


รูปที่ 2.33 ไทม์มิ่งไดอะแกรมของ Command Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

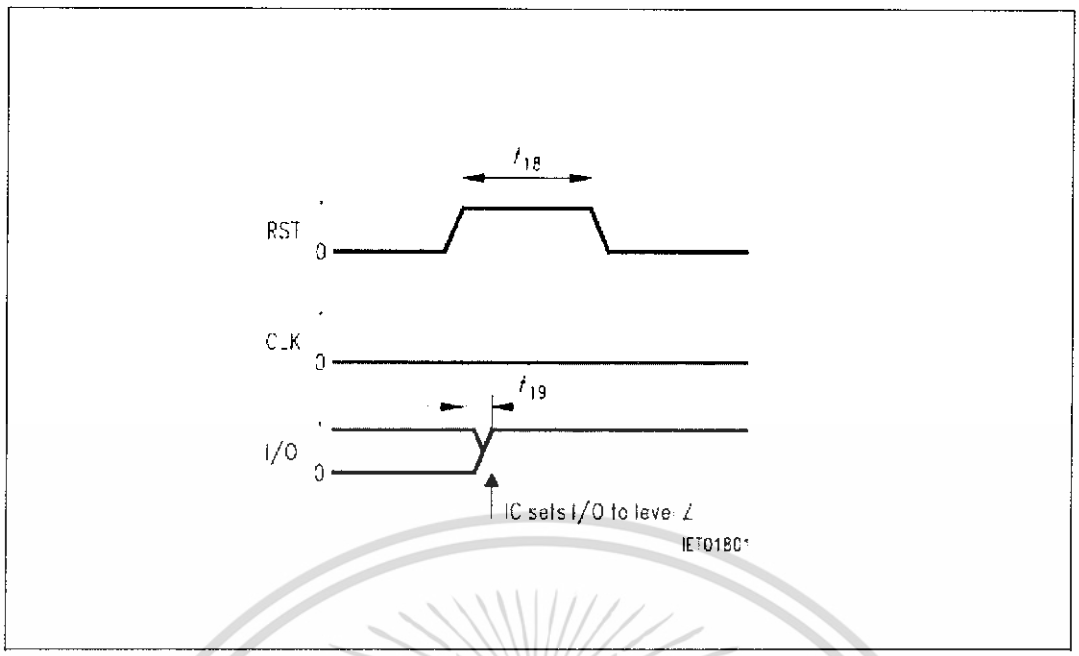


รูปที่ 2.34 ไทม์มิ่งไดอะแกรมของ Outgoing Data Mode



รูปที่ 2.35 ไทม์มิ่งไดอะแกรมของ Processing Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 ไทม์มิ่งไดอะแกรมของ Break

2.7 การใช้งานไมโครคอนโทรลเลอร์ (Microcontroller: MCS-51)

2.7.1 โครงสร้างของ MCS-51

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51 นี้ผลิตโดยบริษัทอินเทลมีอยู่ด้วยกันหลายเบอร์ ซึ่งมีรายละเอียดดังตารางที่ 2.12

Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEP)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/ Vectors	Power Down and Idle Modes
8051	8031	-	4K	128	4	2		√					6/5	
8051AH	8031AH	87C51H 87C51FA	4K	128	4	2		√					6/5	
8052AH	8032AH	87C52H	8K	256	4	3		√					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2		√					6/5	√
80C52	80C32	-	8K	256	4	3		√					8/6	√
83C51FA	80C51FA	87C51FA	8K	256	4	3	√	√					14/7	√
83C51FB	80C51FB	87C51FB	16K	256	4	3	√	√					14/7	√
83C152JA	80C152JA	-	8K	256	5	2		√		√	2		19/11	√
-	80C152JB	-	-	256	7	2		√		√	2		19/11	√
83C152JC	80C152JC	-	8K	256	5	2		√		√	2		19/11	√
-	80C152JD	-	-	256	7	2		√		√	2		19/11	√
83C452	80C452	87C452P	8K	256	5	2		√					9/8	√

ตารางที่2.12 ตารางของไมโครคอนโทลเลอร์แบบชิพเดี่ยวในตระกูล 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.2 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

คุณสมบัติของไมโครคอนโทรลเลอร์ MCS-51 ที่สำคัญๆมีดังนี้

- ต้องการแหล่งจ่ายไฟ 5โวลต์ เพียงชุดเดียว

- มีหน่วยความจำ สำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ภานในชิปจำนวน 4 กิโลไบต์ (เบอร์ 8031, 8032 ไม่มีหน่วยความจำส่วนนี้ ส่วนเบอร์ 8052 มีหน่วยความจำส่วนนี้ 8 กิโลไบต์ และเบอร์ 83C51FB จะมีหน่วยความจำส่วนนี้รวมทั้งสิ้น 16 กิโลไบต์)

- มีหน่วยความจำสำหรับโปรแกรมและข้อมูล (RAM) อยู่ภายในชิปจำนวน 128 ไบต์ (ในเบอร์ 8031, 8051) หรือ 256 ไบต์ (ในเบอร์ 8031, 8052)

- สามารถใช้หน่วยความจำสำหรับโปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้ อย่างละ 64 กิโลไบต์ แยกจากกัน

- คำสั่งส่วนใหญ่ใช้เวลาเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกกะเฮิรตซ์

- มีพอร์ตที่สามารถรับหรือส่งข้อมูลได้ทั้ง 2 ทิศทาง จำนวน 4 พอร์ต พอร์ตละ 8 บิต หรือสามารถใช้งานเป็นพอร์ตขนาด 1บิต แยกจากกัน ทำให้เหมือนมีพอร์ตขนาด 1บิต ใช้งานรวมทั้งสิ้น 32 พอร์ต

- รับและส่งข้อมูลแบบอนุกรมได้ในตัวโดยสามารถกำหนดความเร็วในการรับและส่งข้อมูล (Baud rate) ได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที

- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 2 ระดับ

- มีรีจิสเตอร์สำหรับใช้งานเป็นไทเมอร์ หรือ เคาน์เตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิป หรือนับเปลี่ยนสถานะของสัญญาณภายนอก 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวน pulse วัดความกว้างของ pulse หรือใช้วัดช่วงเวลา (ในเบอร์ 8052 จะมี 3ตัว)

- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วน สามารถเข้าถึงข้อมูลได้ทั้งระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น

- มีคำสั่งคูณและหารเลขในตัวเอง

- สามารถประมวลผลแบบบูลีน เพื่อใช้งานควบคุมโดยเฉพาะ

2.7.3 คุณสมบัติทั่วไปของ AT89C51

ไมโครคอนโทรลเลอร์ AT89C51 จะมีขาใช้งานพื้นฐานดังแสดงในรูปที่ 2.37 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc (ขา 20) ใช้สำหรับต่อไฟเลี้ยง +5

ขา GND (ขา 40) เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (ขา 32-39 หรือ P0.7-P0.0) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล 1 ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย เพื่อให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) ซึ่งมีสถานะ high impedance จึงจะสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (ขา 1-8 หรือ P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล 1 ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย เพื่อให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float)

รูปที่ 2.37 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51

ขาพอร์ต 2 (ขา 21-28 หรือ P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล 1 ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) ซึ่งมีสถานะ high impedance สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (ขา 10-17 หรือ P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล 1 ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) ซึ่งมีสถานะ high impedance สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

ขา P3.0 ใช้สำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

ขา P3.1 ใช้สำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

ขา P3.2 ใช้รับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{\text{INT0}}$

ขา P3.3 ใช้รับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{\text{INT1}}$

ขา P3.4 ใช้รับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา P3.5 ใช้รับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

ขา P3.6 ใช้เป็นขาสัญญาณ $\overline{WR}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

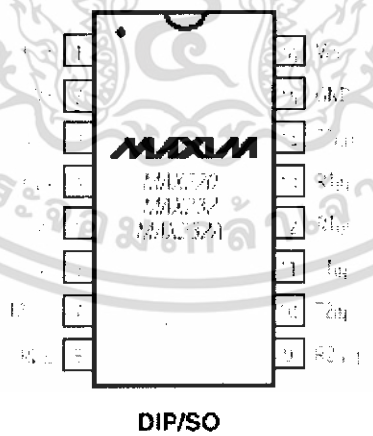
ขา P3.7 ใช้เป็นขาสัญญาณ $\overline{RD}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ซึ่งการใช้งานขาพอร์ต 3 ในหน้าที่พิเศษดังกล่าวนี้ จะต้องโหลดค่า 1 ไปยังแต่ละบิตที่ต้องการใช้ก่อนทุกครั้ง

ขารีเซต(ขา 9 หรือ RST) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์เพื่อเริ่มต้นการทำงานใหม่ ซึ่งจะใช้เมื่อเริ่มจ่ายพลังงานหรือเมื่อโปรแกรมทำงานผิดพลาด โดยในการป้อนสัญญาณเพื่อรีเซต สถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ

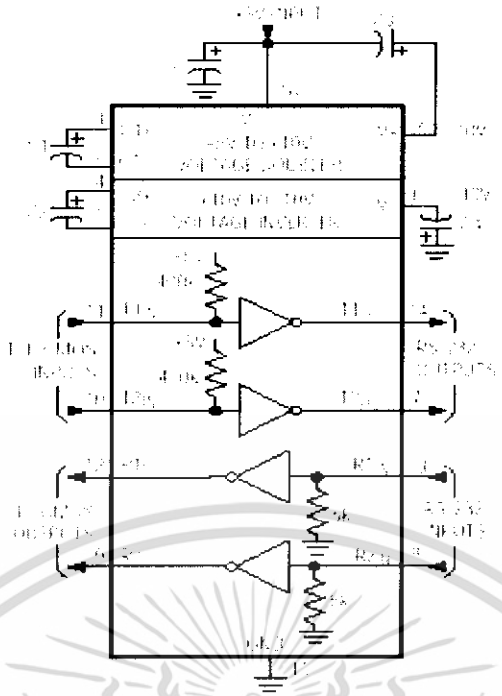
2.7.4 การ เชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์

การใช้งานวงจรพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในมาตรฐาน RS-232 เป็นส่วนใหญ่ แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ ±3 ถึง ±12 V ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับทีทีแอล ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่าน ไอซีที่ทำหน้าที่ในการแปลงระดับแรงดันของสัญญาณจากระดับทีทีแอลไปเป็นระดับแรงดันตามมาตรฐาน RS-232 ไอซีดังกล่าวมีด้วยกันหลายเบอร์จากหลายผู้ผลิต อาทิ MAX232 จาก MAXIM หรือ ICL232 จาก HARRIS เป็นต้น ในรูปที่ 2.38 แสดงการจัดขาของไอซี ICL232 ซึ่งใช้ในการแปลงสัญญาณ RS-232 และในรูปที่ 2.39 แสดงโครงสร้างภายในของไอซี ส่วนวงจรของการต่อกับไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 2.40

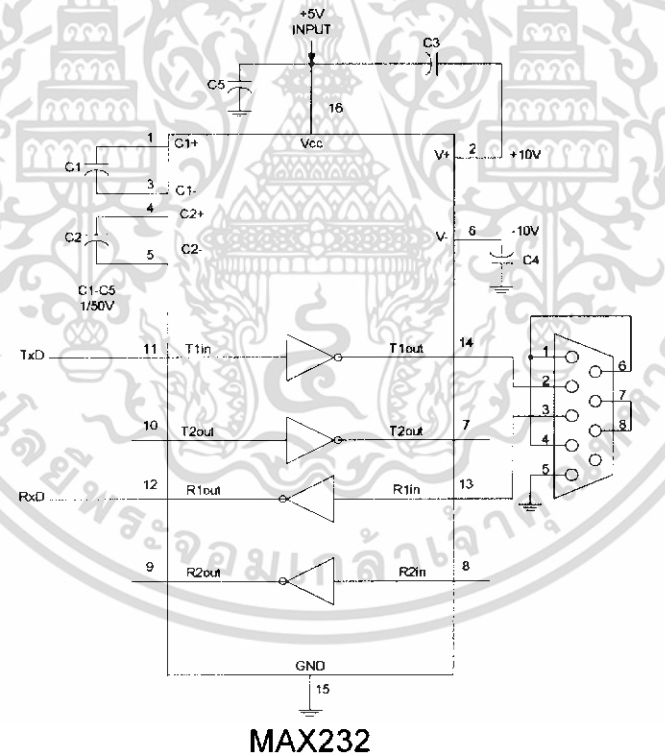


รูปที่ 2.38 การจัดขาของ MAX232 หรือ ICL232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.39 โครงสร้างภายในของ MAX232 หรือ ICL232



รูปที่ 2.40 แสดงวงจรเชื่อมต่อ MAX232 หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ และไมโครคอนโทรลเลอร์ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ความรู้เบื้องต้นเกี่ยวกับพอร์ตอนุกรม

การเคลื่อนที่ย้ายข้อมูลจากเครื่องคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงภายนอกหรือคอมพิวเตอร์ด้วยกันมี 2 รูปแบบคือ รับส่งข้อมูลแบบขนานและรับส่งข้อมูลแบบอนุกรม

การรับส่งข้อมูลแบบขนานเป็นการรับและส่งข้อมูลคราวละ 4 ถึง 8 บิตในเวลาเดียวกัน ทำให้การรับและส่งข้อมูลมีความเร็วสูง แต่จำนวนสายที่ใช้ในการถ่ายทอดข้อมูลมีมากเท่ากับจำนวนบิตของข้อมูลที่ทำการถ่ายทอด นอกจากนั้นยังมีสายที่ใช้สำหรับควบคุมและตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูลก็ได้

ในขณะที่การรับส่งข้อมูลแบบอนุกรมจะเป็นการรับส่งข้อมูลครั้งละ 1 บิต โดยมีรูปแบบการรับส่งข้อมูลที่เป็นมาตรฐาน ต้องมีการตรวจสอบความพร้อมในการรับและส่งข้อมูลของตัวส่งและตัวรับ การรับส่งข้อมูลแบบอนุกรมมีข้อดีในเรื่องจำนวนสายสัญญาณที่น้อยมาก และไม่แปรผันตามจำนวนบิตของข้อมูล ระยะทางในการรับส่งข้อมูลสูงกว่าแบบขนานมาก

การจำแนกวิธีการส่งตามลักษณะการจัดข้อมูลแบ่งออกได้เป็น 2 วิธี ได้แก่

1. การส่งแบบขนาน (Parallel Transmission)

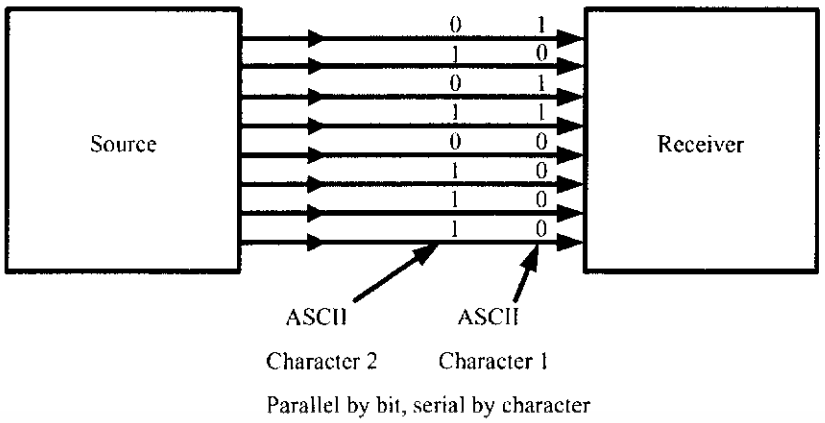
วิธีการส่งในลักษณะนี้นั้น เราจะนำทุกๆ บิทของรหัสของอักษรหนึ่งตัวมาส่งไปพร้อมๆ กันนั่นหมายความว่า หากเรามีรหัสขนาด 8 บิทก็ต้องมีแซ็นแน็ลสำหรับการส่งจำนวนเท่ากัน ตัวอย่างเช่น ในรหัสอัสกีย์ที่กล่าวไว้ ซึ่งมีจำนวนบิทสำหรับตัวอักษรแต่ละตัว 8 บิทจึงต้องมีแซ็นแน็ลติดต่อระหว่างด้านรับและด้านส่งจำนวน 8 แซ็นแน็ลดังแสดงในรูป 2.41

คำว่าการส่งแบบขนานในที่นี้หมายความว่าทุกๆ บิทของข่าวสารหนึ่งตัวอักษรจะถูกส่งไปในลักษณะขนานกัน แต่การส่งระหว่างตัวอักษรนั้นเป็นแบบอนุกรม คือส่งตัวอักษรที่ 1, ที่ 2, ที่ 3 ไปเรื่อยๆ การส่งแบบขนานนั้นมักใช้ในระบบสื่อสารที่มีระยะทางไม่ไกลนักและโดยเฉพาะอย่างยิ่งในการส่งข้อมูลระหว่างคอมพิวเตอร์กับอุปกรณ์ประกอบของคอมพิวเตอร์ต่างๆ เช่น เครื่องอ่านบัตร เครื่องเทปแม่เหล็กต่างๆ เป็นต้น โดยวิธีการนี้ทำให้เราได้ระบบการสื่อสารที่มีอัตราการส่งข้อมูลสูงมาก แต่ในกรณีที่ระยะทางไกลมากๆ นั้นระบบการส่งแบบขนานจะไม่ได้รับความนิยม เพราะราคาของการวางแซ็นแน็ลแบบขนานค่อนข้างแพง

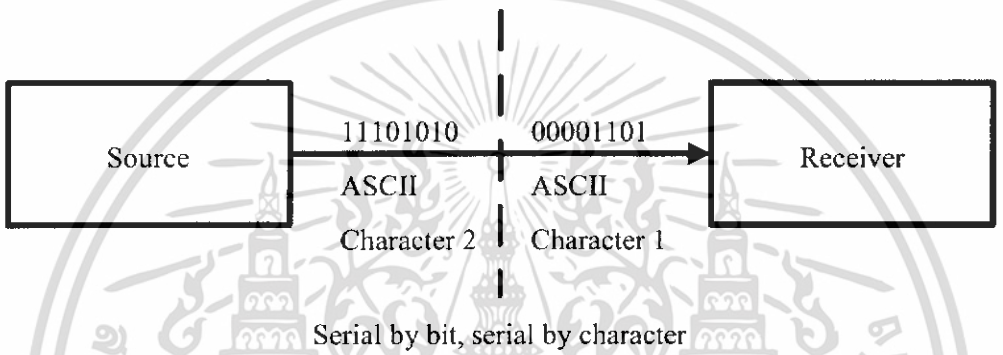
2. การส่งแบบอนุกรม(Scrial Transmission)

วิธีการที่ได้รับความนิยมแพร่หลายที่สุดสำหรับการส่งข่าวสารก็คือ การส่งแบบอนุกรม ในการส่งแบบอนุกรมนั้น บิททั้งหมดของตัวอักษรหนึ่งตัวจะถูกนำมาส่งไปทีละบิทติดต่อกันไปเรื่อยๆ ตามแซ็นแน็ลซึ่งมีเพียงแซ็นแน็ลเดียวดังแสดงในรูป 2.42 ทางด้านรับเมื่อรับข้อมูลมาแล้วก็จะนำมาจัดเป็นตัวอักษรขึ้นใหม่ให้ตรงกับชุดของตัวอักษรที่ทางด้านส่ง ส่งมา ซึ่งวิธีการดังกล่าวนี้จะต้องประกอบด้วยความสัมพันธ์ในการทำงานระหว่างด้านรับและด้านส่งสองชนิดคือ (1) ความสัมพันธ์ของบิท (Bit Synchronization) และ (2) ความสัมพันธ์ของตัวอักษร (Character Synchronization) ซึ่งจะกล่าวถึงรายละเอียดต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.41 การส่งแบบขนาน

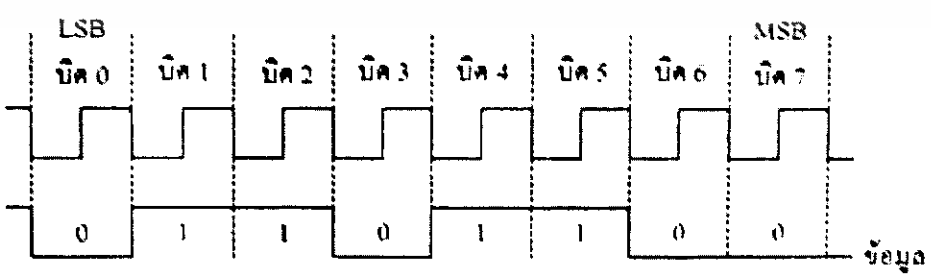


รูปที่ 2.42 การส่งแบบอนุกรม

การสื่อสารแบบอนุกรมแบ่งออกเป็น 2 แบบคือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส

2.8.1 การสื่อสารข้อมูลแบบซิงโครนัส

การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมกันอยู่กับการรับและการส่งสัญญาณ ด้วยตัวอย่างการส่งแบบซิงโครนัส ก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของนาฬิกา ส่วนสายอีกเส้นจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์ รูปที่ 2.43 แสดงให้เห็นถึงไดอะแกรมเวลาของการสื่อสารข้อมูลแบบซิงโครนัส



รูปที่ 2.43 แผนผังการทำงานเวลาของการสื่อสารข้อมูลแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

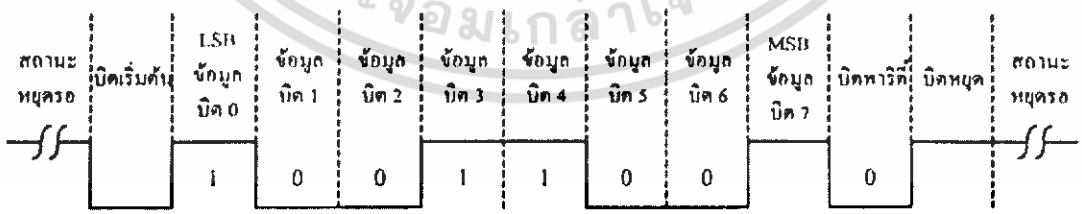
การสื่อสารข้อมูลแบบอะซิงโครนัส คือการรับและส่งข้อมูลโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วย แต่จะใช้การกำหนดอัตราความเร็วในการรับและส่งข้อมูลให้มีค่าเท่ากัน ซึ่งเรียกอัตราเร็วนี้ว่า อัตราบอดเรต (Baud Rate) มีหน่วยเป็น บิตต่อวินาที

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนด้วยกันคือ

1) บิตเริ่มต้น
2) บิตข้อมูลแบบอนุกรม มีขนาด 5, 6, 7 หรือ 8 บิต
3) บิตตรวจสอบพาริตี้ (Parity Bit) มีขนาด 1 บิต หรือไม่มีบิต
4) บิตปิดท้ายหรือบิตหยุด (Stop Bit) มีขนาด 1, 1.5 หรือ 2 บิต

รูปที่ 2.44 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส เมื่อไม่มีการส่งข้อมูลขา DATA จะมีสถานะลอจิก "1" เรียกสถานะนี้ว่า สถานะหยุดรอ (Waiting Stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากให้ขา DATA มีลอจิก "0" ด้วยช่วงระยะเวลา 1 บิต เรียกบิตนี้ว่าบิตเริ่มต้น (Start Bit) จากนั้นบิตข้อมูลจะถูกส่งออกไปโดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุดหรือบิต LSB ก่อน ซึ่งข้อมูลที่ต้องการส่งอาจมีจำนวน 5, 6, 7 หรือ 8 บิตก็ได้ จากนั้นตามด้วยพาริตี้บิต (Parity Bit) ซึ่งใช้ในการตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งก็คือ บิตปิดท้ายหรือบิตหยุด (Stop Bit) โดยจะเป็นการทำให้ขา DATA มีสถานะลอจิก "1" อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสหรือ อัตราบอด หรือ บอดเรตที่ใช้สำหรับพอร์ตอนุกรม RS-232 มีด้วยกันหลายค่า ได้แก่ 110, 150, 300, 600, 1200, 2,400, 4,800, 9,600 และ 19,200 บิตต่อวินาที โดยมีค่ามากขึ้นตามเทคโนโลยีของคอมพิวเตอร์เนื่องจากบอดเรตคือค่าของจำนวนบิตที่สามารถส่งได้ใน 1 วินาที สมมติว่า ข้อมูลอนุกรมมีขนาด 8 บิตไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูล 1 ไบต์ จะมีความยาวเท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9,600 ต่อวินาที ก็จะสามารถส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที



รูปที่ 2.44 รูปแบบของข้อมูลแบบอะซิงโครนัส

การตรวจสอบพาริตี้สามารถกำหนดเป็นแบบคี่ (Odd), แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตี้ก็ได้ พาริตี้คี่ หรือพาริตี้คู่แสดงถึงจำนวนลอจิก "1" ทั้งหมดภายในข้อมูลที่ส่งไป 1 ไบต์รวมพาริตี้ว่ามีจำนวนเป็นเลขคู่หรือเลขคี่ ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต มีค่าเท่ากับ 99H หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10011 001B จะเห็นว่าข้อมูลในไบต์มีจำนวนลอจิก “1” จำนวน 4 ตัว ซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ ค่าของพาริตี้บิตจะต้องมีลอจิกเป็น “0” แต่ถ้ากำหนดพาริตี้เป็นคี่ ค่าของพาริตี้บิตจะต้องมีลอจิกเป็น “1” เพื่อให้ข้อมูลไบต์รวมทั้งบิตพาริตี้เป็นคี่

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART (Universal Asynchronous Receiver Transmitter) ซึ่งทางภาครับจะต้องกำหนดคุณสมบัติการตรวจสอบพาริตี้ที่ตรงกันเอาไว้ว่า จะตรวจสอบพาริตี้คี่ หรือพาริตี้คู่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่ แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้รับทราบ กระบวนการดังกล่าวเป็นวิธีการตรวจสอบความผิดพลาดที่เกิดขึ้นในการรับส่งข้อมูลที่ง่ายที่สุด แต่มันสามารถตรวจสอบได้เมื่อมีบิตข้อมูลที่ทำการรับส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี้

2.8.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้ส่งผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกล โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association: EIA) ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3V จนถึง -12V แสดงว่ามีข้อมูล (mark) และ +3V จนถึง +12V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ถูกใช้ในการกำหนดรูปแบบการสื่อสารข้อมูลกันระหว่างอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating: DCE) อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE ทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ในโมเด็มจะเป็นแบบ DCE

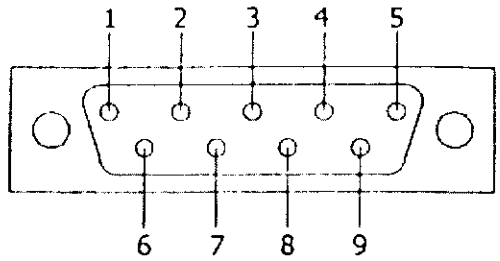
สำหรับการใช้งานคอมพิวเตอร์ พอร์ตอนุกรม RS-232 ถูกใช้เพื่อเชื่อมต่อกับโมเด็ม, เมาส์ และเครื่องพิมพ์ที่สามารถติดต่อทางพอร์ตอนุกรมได้

2.8.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

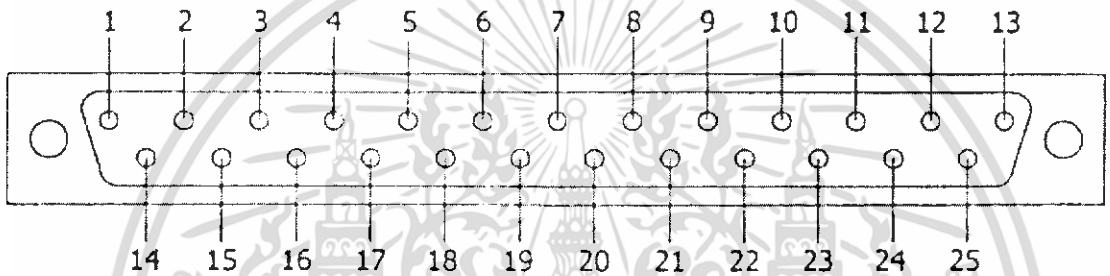
มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อื่นๆ ที่เคยมีใช้งานมาในอดีต ไม่ค่อยมีความสำคัญมากนักจึงถูกยกเลิกไป โดยแสดงรูปร่างตำแหน่งขาใน รูปที่ 2.45



(ก) คอนเน็กเตอร์อนุกรม 9 ขา หรือแบบ DB-9 (มองจากด้านหลังคอมพิวเตอร์)



(ข) คอนเน็กเตอร์อนุกรม 25 ขา หรือแบบ DB-25 (มองจากด้านหลังคอมพิวเตอร์)

รูปที่ 2.45 คอนเน็กเตอร์อนุกรม

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	Single 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 2.13 การจัดขาสัญญาณของพอร์ตอนุกรมในแบบต่างๆ และหน้าที่การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect: CD ขานี้จะแอคทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับใช้งานปกติ ขานี้จะไม่ถูกนำมาใช้งานมากนัก

ขา Received Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยจะนำข้อมูลที่อ่านได้ไปเก็บไว้ในรีจิสเตอร์บัฟเฟอร์

ขา Transmitted Data: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์โดยการนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

ขา Data Terminal Ready: DTR เป็นขาเอาต์พุตที่ใช้สำหรับส่งสัญญาณออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อกับอุปกรณ์ปลายทาง โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์และถ้าใช้การเชื่อมต่อแบบ 3 สาย ต้องเชื่อมต่อขา DTR และ DSR ของพอร์ตอนุกรมเข้าด้วยกัน และจะต้องต่อเชื่อมเข้ากับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

ขา Single Ground: GND เป็นขากราวด์ของสัญญาณ

ขา Data Set Ready: DSR ขานี้จะใช้ควบคู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอก

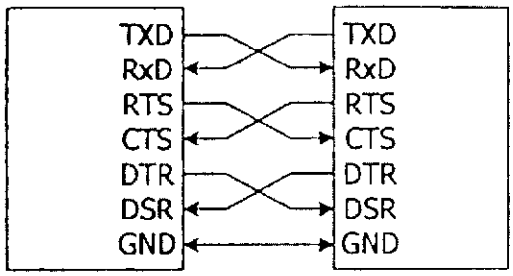
ขา Request To Send: RTS เป็นขาเอาต์พุตสำหรับส่งสัญญาณร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลมาให้คอมพิวเตอร์โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ซึ่งในกรณีที่มีการเชื่อมต่อแบบ 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS เข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

ขา Clear To Send: CTS เป็นขาอินพุตทำหน้าที่รอรับสัญญาณที่ส่งเข้ามา เมื่อมีการส่งสัญญาณเข้ามาที่ขานี้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ขานี้จะใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลแล้วหรือยัง

ขา Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มแล้วยังมีความต้องการตรวจสอบสัญญาณเรียกสายโทรศัพท์

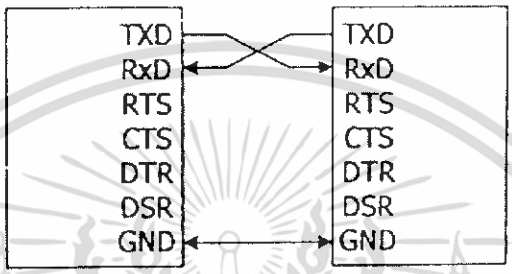
สำหรับการเชื่อมต่อสายระหว่างคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังในรูปที่ 2.44 ลูกศรในรูป แสดงถึงทิศทางของข้อมูล การเชื่อมต่อ การเชื่อมต่อในรูปที่ 2.46 (ก) เป็นการเชื่อมต่อแบบ NULL MODEM หรือการเชื่อมต่อโดยตรง โดยไม่ต้องผ่านโมเด็ม ส่วนการเชื่อมต่อในรูปที่ 2.46 (ข) เป็นการเชื่อม ต่อโดยใช้สัญญาณน้อยที่สุดเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คอมพิวเตอร์ อุปกรณ์ภายนอก

(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null Modem



คอมพิวเตอร์ อุปกรณ์ภายนอก

(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS-232
ในลักษณะที่ใช้สายสัญญาณน้อยที่สุดเพียง 3 เส้น

รูปที่ 2.46 การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรมในรูปแบบต่างๆ

2.8.5 มาตรฐานการรับส่งข้อมูลแบบอะซิงโครนัส (UART)

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

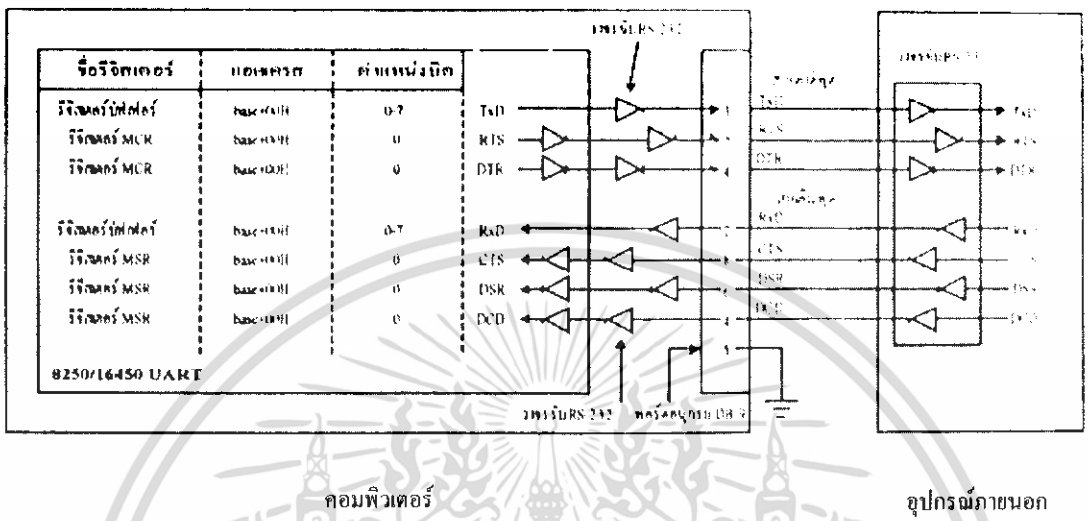
หน้าที่หลักของ UART คือแปลงข้อมูลที่อยู่ในรูปแบบขนานจากซีพียูให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วทำการส่งออกไปและแปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งเข้าสู่ซีพียู ซึ่งนอกจาก UART จะส่งข้อมูลไปยังซีพียูแล้วยังแจ้งรายละเอียดอื่นๆ ของข้อมูล ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราความเร็วในการรับส่งข้อมูลหรือบอดเรต, รูปแบบการส่งข้อมูล, ความผิดพลาดที่เกิดขึ้นระหว่างการส่งข้อมูล เช่น ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน เป็นต้น

ภายใน UART จะมีวงจรสร้างบอดเรตโปรแกรมได้ (Programmable Baud rate Generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้จะมีขนาด 16 บิต ดังนั้นจะสามารถกำหนดตัวหารอยู่ในช่วง 1-65, 535

มาตรฐานการรับส่งข้อมูลแบบอะซิงโครนัสที่ใช้คอมพิวเตอร์ทั่วไปจะมี UART ที่ใช้งานกันอยู่ 2 เบอร์ คือ เบอร์ 8250 และ 16550 สำหรับ UART เบอร์ 8250 เป็น UART มาตรฐานที่มีใช้กันมายาวนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

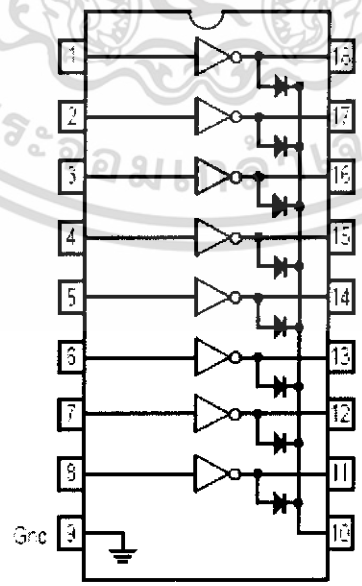
UART เบอร์นี้มีบัฟเฟอรสำหรับรับและส่งข้อมูลเป็นตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาที สำหรับ UART เบอร์ 16550 จะเพิ่มส่วนของชิฟรีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์ เข้าไปด้วย ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ระดับ 256 กิโลบิตต่อวินาทีได้



รูปที่ 2.47 ผังการทำงานภายในของขาสัญญาณต่างๆ ของพอร์ตอนุกรม

2.8.6 ULN 2803

ULN 2803 เหมาะสำหรับการใช้งานที่ใช้ในการเชื่อมต่อระหว่างวงจรดิจิตอลที่มีระดับแรงดันลอจิกต่ำ (เช่น TTL, CMOS หรือ PMOS/NMOS) กับอุปกรณ์อื่นๆ ที่ต้องการค่ากระแส หรือแรงดันที่มีระดับสูงกว่า



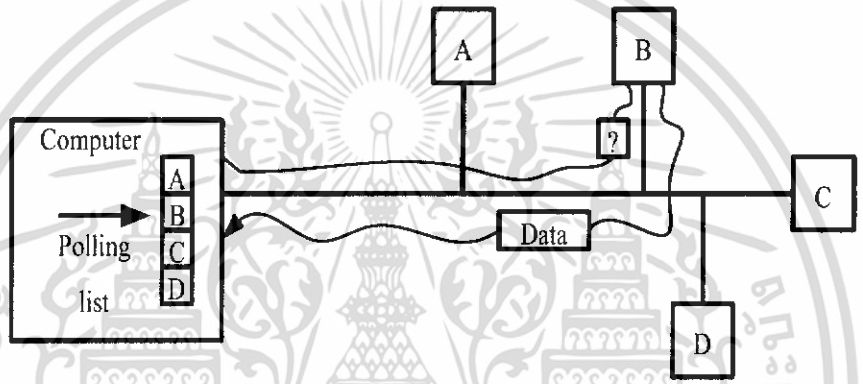
รูปที่ 2.48 การจัดขาของ ULN 2803

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 การโพล (Polling)

เทคนิคที่ตัวควบคุมสายใช้ควบคุมการไหลของข้อมูลบนสายเรียกว่า การโพล (Polling) ซึ่งเราอาจจะให้คำจำกัดความของการโพลว่าเป็นกระบวนการในการเรียกให้อุปกรณ์ปลายทางต่างๆ ทำการส่งข้อมูล (ตามลำดับที่กำหนด)

วิธีการที่ง่ายที่สุดของการโพล ก็คือการให้คอมพิวเตอร์เป็นตัวควบคุมอุปกรณ์ปลายต่างๆ เป็นลำดับขั้น โดยการให้คอมพิวเตอร์คอยถาม (ตรวจสอบว่าอุปกรณ์ปลายตัวใดมีข้อมูลที่จะส่งบ้าง) เรียงตามลำดับไป เมื่อถามครบทุกตัวแล้วจะกลับมาถามใหม่อีกครั้งโดยเรียงตามลำดับเดิมอีกเป็นเช่นนี้ไปเรื่อยๆ ซึ่งในกรณีเช่นนี้อุปกรณีปลายที่ถูกถามจะตอบกลับมาว่ามีข่าวสารหรือไม่มี นั่นหมายถึงว่าผลตอบสนองของการไหลคือข้อมูลหรือไม่มีข้อมูลที่เรียกว่า NTR (No Traffic Response) ซึ่งจากรูปที่ 2.49 จะเป็นการแสดงถึงการโพลของคอมพิวเตอร์ในสายแบบหลายจุดที่มีอุปกรณ์ปลาย 4 ตัว

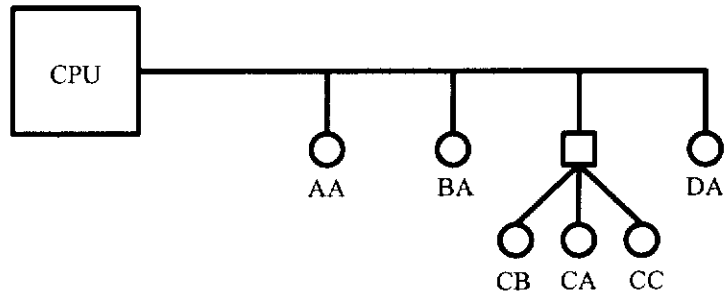


รูปที่ 2.49 การโพลบนสายแบบหลายจุด

ภายในคอมพิวเตอร์จะมีรายละเอียดลำดับของอุปกรณ์ที่จะทำการโพล ในรูปแสดงถึงการโพลของอุปกรณ์ปลาย B โดยอุปกรณ์ปลาย B มีข้อมูลที่จะส่ง ฉะนั้นเมื่อคอมพิวเตอร์ส่งสัญญาณถามไปยังอุปกรณ์ปลาย B (การโพลอุปกรณ์ปลาย B) อุปกรณ์ B ตอบสนองกลับโดยการส่งข่าวสารกับมา ในการจัดลำดับการโพลอุปกรณ์ปลายนั้นเราอาจจะจัดลำดับสิทธิการโพลเหนือกว่าอุปกรณ์ปลายอื่นๆ โดยการใส่ชื่อ A ลงไปในรายชื่อทุกๆ ตัวที่ 2 ทำให้ได้ลำดับการโพลดังนี้ A-B-A-C-A-D-A-B ในทำนองเดียวกันหากเราไม่ต้องการอุปกรณ์ปลายใดๆ เราก็สามารถตัดออกจากการโพลได้

ในการสร้างระบบการโพลจะต้องมีอุปกรณ์ปลายที่เรียกได้โดยคอมพิวเตอร์โดยอาจจะกำหนดเป็นตำแหน่งของแต่ละตัวที่แตกต่างกันไป รูปที่ 2.50 แสดงให้เห็นสายแบบหลายจุดที่มีการต่ออยู่แบบ 4 จุด แต่ละจุดจะต่ออยู่กับอุปกรณ์ปลายหนึ่งตัวนอกจากจุดที่สามที่มีอุปรณ์ปลายต่ออยู่ 3 ตัว สำหรับกรณีนี้เราต้องใช้การเรียกตำแหน่งถึง 2 ระดับ ระดับแรกกำหนดจุดต่อและระดับที่สองกำหนดถึงอุปกรณ์ปลาย จากรูป ที่อุปกรณ์ปลาย AA นั้นหมายถึงจุดต่อ A และอุปกรณ์ A ที่ตำแหน่ง CB หมายถึงจุดต่อ C และอุปกรณ์ B และเป็นเช่นเดียวกันสำหรับจุดต่อและอุปรณ์อื่นๆ

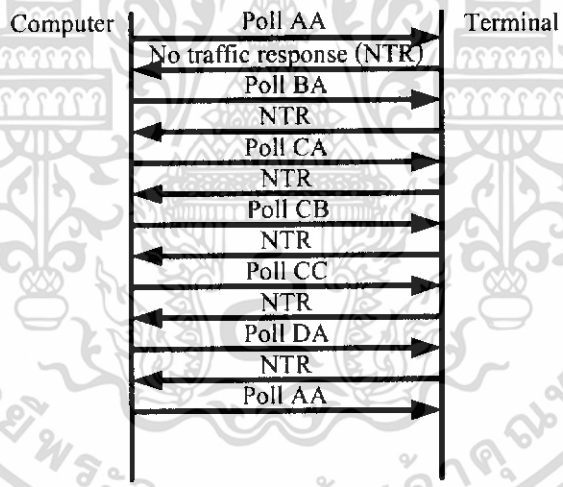
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.50 การโพลในสายแบบหลายจุดชนิด 2 ระดับ

สัญญาณโพลจากคอมพิวเตอร์ที่ส่งออกไปนั้นอุปกรณ์ปลายทุกๆ ตัวจะรับโหมด แต่จะมีเฉพาะอุปกรณ์ปลายที่มีตำแหน่งตรงกับที่ส่งมาในสัญญาณปลายเท่านั้นที่รับสัญญาณเข้ามา กล่าวคือ อุปกรณ์ปลายใดๆ ก็ตามถ้าหากสัญญาณที่รับมาไม่ใช่สัญญาณของตำแหน่งของตนเองก็จะไม่รับสัญญาณนั้น อุปกรณ์ปลายที่ถูกโพลจะตอบรับการโพลด้วยข่าวสารที่มีอยู่หรือการไม่มีข่าวสาร NTR ปกติแล้วอัตราความเร็วของการโพลจะสูงกว่าอัตราการสร้างข่าวสาร ฉนั้นการตอบรับของการโพลมักจะเป็นกรณี NTR มากกว่ากรณีมีข่าวสาร

รูป 2.51 แสดงถึงลำดับการทำงานที่เกิดขึ้นของรูปที่ 2.50 ในกรณีที่ไม่มีข้อมูลที่จะส่งจากอุปกรณ์ปลายใดๆ ในโครงข่าย โดยเรากำหนดคอมพิวเตอร์ไว้ทางซ้ายมือและอุปกรณ์ปลายทางอยู่ทางขวามือ โดยการทำงานเริ่มจากบนลงล่าง

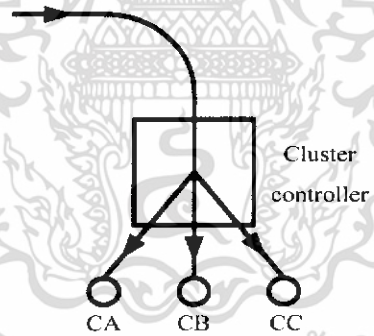


รูปที่ 2.51 ลำดับการโพลของรูปที่ 2.50

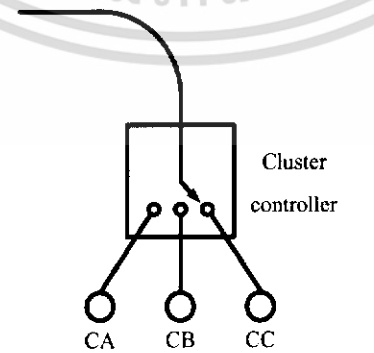
แรกสุด คอมพิวเตอร์จะส่งสัญญาณโพลของ AA ออกไปก่อน อุปกรณ์ปลาย AA จะตอบกลับมาด้วย NTR จากนั้นการโพลจะเปลี่ยนไปเป็นของ BA ผลตอบกลับมาก็คือ NTR เช่นกันแล้วการโพลก็เปลี่ยนไปยัง CA ซึ่งก็ยังตอบด้วย NTR ไปเรื่อยๆ จนกระทั่งการโพลเกิดขึ้นครบทุกตัวแล้วคอมพิวเตอร์จะกลับมาโพล AA ใหม่ ซึ่งลำดับการโพลดังกล่าวนี้จะใช้เวลาก่อนที่อุปกรณ์เดิมจะถูกโพลใหม่ เวลาดังกล่าวนี้เรียกว่า เวลาหน่วงในการโพล (Polling Delay)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการของการกำหนดเกี่ยวกับการโพลนั้นมีหลายวิธี บางระบบอาจจะมีการโพลด้วยความเร็วสูงสุดเท่าที่จะทำได้ในกรณีเช่นนี้คอมพิวเตอร์จะต้องส่งสัญญาณไปยังอุปกรณ์ปลายในทันทีที่ได้รับสัญญาณ NTR จากอุปกรณ์ปลายตัวก่อน ในระบบบางระบบการโพลจะเกิดขึ้นเป็นระยะ ๆ ตัวอย่างเช่น คอมพิวเตอร์ส่งสัญญาณโพลทุก ๆ 20 มิลลิวินาที ในกรณีดังกล่าวนี้ถ้าเป็นโครงข่ายในรูป 2.51 คอมพิวเตอร์จะต้องเสียเวลา = 6x200 = 1200 มิลลิวินาที สำหรับการโพลอุปกรณ์ปลายจนครบ 1 รอบ หากโอเปอร์เรเตอร์ของอุปกรณ์ปลายใดต้องการส่งข่าวสารก็จะกดสวิทซ์สำหรับส่ง (Transmit key) บนอุปกรณ์ปลาย จากการกดสวิทซ์นี้จะไม่มีผลเกิดขึ้นจนกว่าอุปกรณ์ปลายนั้นจะถูกโพลจากคอมพิวเตอร์ เนื่องจากความเร็วในการโพลจะสูงมากทำให้เวลาของการโพลหนึ่งรอบน้อยกว่าเวลาที่กดสวิทซ์ทำให้คอมพิวเตอร์รับข่าวสารที่เกิดจากการกดสวิทซ์ได้ แต่สำหรับในกรณีที่มีอุปกรณ์ปลายเป็นจำนวนมากในสายเช่นในระบบการสำรองที่นั่งของเครื่องบินจะทำให้เวลาในการโพลหนึ่งรอบของอุปรณ์ปลายมีค่ามากขึ้น ในกรณีเช่นนี้เราอาจเพิ่มความเร็วของการโพลโดยการรวมอุปกรณ์ปลายเข้าเป็นชุดแล้วทำการโพลทั้งชุด เราสามารถกระทำได้โดยการจำแนกตำแหน่งเป็นกลุ่ม (Group Address) ซึ่งตำแหน่งนี้อุปกรณ์ปลายทุก ๆ ตัวในชุดต่อนั้นจะตอบสนองได้โดยการใช้ตัวควบคุมกลุ่มของอุปกรณ์ปลานนี้ในการติดต่อกับคอมพิวเตอร์ดังแสดงในรูปที่ 2.52 แต่การกำหนดในลักษณะนี้มีข้อเสียคือจะทำให้เกิดคอนทนชั่นในตัวควบคุมฉนั้นจึงต้งใช้วิธีการเลือกให้มีอุปกรณ์ปลายตัวเดียวเท่านั้นที่ทำการส่งข้อมูล ตัวควบคุมนี้บางครั้งเรียก ตัวมัลติเพลกซ์อุปกรณ์ปลาย (Terminal Multiplexer) หรือตัวแปลงสายร่วม (line-sharing Adapter) เป็นต้น



(a) All terminal see the group poll



(b) Terminal CA signals that it has a message to send

รูปที่ 2.52 ตัวควบคุมจุดต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.52 ตัวควบคุมกลุ่ม (Cluster Controller) จะส่งข่าวสารที่มาจากคอมพิวเตอร์ไปยังอุปกรณ์ปลายทุก ๆ ตัวที่อยู่ในกลุ่มนั้น อุปกรณ์ปลายทุก ๆ ตัวที่อยู่ในกลุ่มดียวกันจะได้รับสัญญาณนี้ การทำงานของตัวควบคุมนี้หากพิจารณาจากทางด้านอุปกรณ์ปลาย ตัวควบคุมกลุ่มจะมีสวิทซ์ทางตรรก (Logical Switch) ที่ทำหน้าที่ ต่อสายของอุปกรณ์ปลายเพียงตัวเดียวเพื่อการส่วข้อมูลไปยังคอมพิวเตอร์ ในกรณีที่มีอุปกรณ์ปลายหลายตัวมีข่าวสารที่จะส่งไปยังคอมพิวเตอร์นั้น อุปกรณ์ปลายเหล่านั้นก็จะส่งสัญญาณไปยังตัวควบคุมกลุ่ม ซึ่งตัวควบคุมกลุ่มก็จะทำการต่ออุปกรณ์ปลายตัวหนึ่งเข้ากับสายต่อ เพื่อส่งข่าวสารไปยังคอมพิวเตอร์ วิธีการในลักษณะนี้จะช่วยประหยัดเวลาได้มาก

วิธีการโพลเป็นกลุ่ม (Group polling) แบบนี้นั้นสามารถใช้ได้กับการโพลอุปกรณ์ปลายที่ต่ออยู่กับตัวควบคุมกลุ่มตัวเดียวกัน แต่หากเราใช้วิธีการนี้กับการโพลอุปกรณ์การควบคุมต่างกัน ก้จะให้เกิดการคอนเทนชั่นขึ้นเพราะ อาจจะมีอุปกรณ์ปลายหนึ่งตัวของแต่ละส่วนอาจจะมีข่าวสารที่จะส่งในเวลาเดียวกัน ซึ่งเป็นผลเสียทำให้ไม่สามารถรับ/ส่งได้อย่างถูกต้อง ตัวควบคุมกลุ่มนั้นจะแตกต่างกันไปสำหรับแต่ละระบบ ในบางระบบตัวควบคุมกลุ่มอาจจะเป็นอุปกรณ์แบบง่าย ๆ ดังรูปที่ 2.52 แต่ในบางระบบอาจใช้อุปกรณ์ทางตรรกเข้าช่วย

2.10 การเลือก (Selection)

ในทิศทางตรงกันข้ามเมื่อคอมพิวเตอร์ต้องการส่งข่าวสารไปยังอุปกรณ์ปลายตัวหนึ่ง คอมพิวเตอร์จะทำการเลือกอุปกรณ์ปลายตัวที่ต้องการให้รับข้อมูล ในกรณีนี้คอมพิวเตอร์จำเป็นต้องส่งชื่อของอุปกรณ์ปลายที่เรียกว่าตำแหน่ง (Address) ไปก่อนเพื่อให้มีเฉพาะอุปกรณ์ปลายที่กำหนดนั้นรับข้อมูล มีวิธีการเลือกอยู่ 2 วิธี คือ การเลือกแบบรวดเร็ว (Fast Selection) และการเลือกแบบสุภาพ (Polite Selection) ในระบบการเลือกแบบรวดเร็วนั้น ค่าตำแหน่งของอุปกรณ์ปลายจะถูกส่งไปก่อนแล้ว ข่าวสารจึงจะถูกส่งไป ทางด้านอุปกรณ์ปลายก็จะต้องรับข่าวสารที่ส่งมา ในกรณีที่เป็นอุปกรณ์ปลายแบบจอภาพ ข่าวสารที่รับเข้ามาจะถูกนำไปแสดงที่จอภาพในทันทีทันใด วิธีการนี้จะเหมาะสำหรับการจราจรของข่าวสารแบบเรียกร้อง (Solicited Traffic) เพราะข่าวสารนี้เป็นข่าวสารที่ทางโอเปอร์เรเตอร์ของอุปกรณ์ปลายสอบถามมาและต้องการคำตอบ วิธีการนี้เป็นวิธีการของระบบ ถาม-ตอบ (Enquiry and response System) โดยที่โอเปอร์เรเตอร์ทางอุปกรณ์ปลายทางแล้วคำตอบก็จะถูกส่งออกไปยังอุปกรณ์ปลายนั้น ๆ ทันทีโดยจะไปปรากฏบนจอ

สมมติว่ามีอุปกรณ์ปลายตัวหนึ่งที่โอเปอร์เรเตอร์มีข่าวสารจำนวนมาก เช่น 2,000 ตัวอักษร โดยโอเปอร์เรเตอร์ใส่ข่าวสารเข้าไปด้วยความเร็ว 2 ตัวอักษร/วินาที และมีอุปกรณ์ปลายอีกตัวหนึ่งที่โอเปอร์เรเตอร์ตอการส่งข่าวสารไปยังอุปกรณ์ปลายตัวแรก เมื่อใช้วิธีการของการเลือกแบบรวดเร็วนี้ ข่าวสารจากอุปกรณ์ปลายตัวที่สองจะไปปรากฏบนจอภาพของอุปกรณ์ปลายตัวแรกในทันทีทันใด และอาจจะไปรบกวนทำให้ข้อมูลเก่าบนจอภาพที่โอเปอร์เรเตอร์กำลังป้อนเข้าถูกทำลาย ข่าวสารที่เข้ามาในลักษณะนี้เรียกว่า ข่าวสารที่ไม่ได้เรียกร้อง (Unsolicited Message) ซึ่งเป็นข่าวสารที่โอเปอร์เรเตอร์ไม่ได้ถามหาหรือไม่ทราบว่าจะได้รับและเป็นข่าสารที่กล่าวได้ว่าไม่ต้องการในขณะนั้น จึงจำเป็นต้องมีวิธีการพิเศษในการจัดการกับข่าสารนี้ ซึ่งมีวิธีการหลายวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

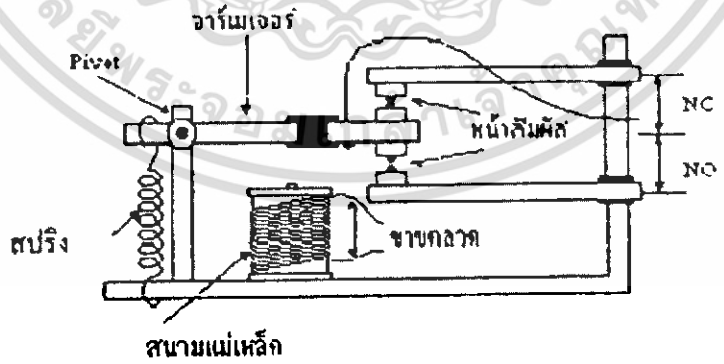
วิธีการหนึ่งก็คือ การส่งซีเควนซ์การเลือก (Selection sequence) ออกไปยังอุปกรณ์ปลาย ซึ่งมีลักษณะคล้ายๆ กับซีเควนซ์การโพล (Polling sequence) โดยจะถามอุปกรณ์ปลายว่าต้องการรับข้อมูลหรือไม่ ถ้าหากอุปกรณ์ปลายไม่อยู่ในฐานะที่จะรับข่าวสารก็จะตอบกลับไปว่าไม่ต้องการ ซึ่งในกรณีนี้ข่าวสารที่จะส่งมานั้นก็จะถูกเก็บที่คอมพิวเตอร์ แต่ถ้าอุปกรณ์ปลายอยู่ในส-นะที่สามารถรับข่าวสารได้ก็จะตอบกลับไปว่าได้ทำให้คอมพิวเตอร์ส่งข่าวสารมายังอุปกรณ์ปลายนั้นโดยวิธีการของการเลือกแบบรวดเร็ว

ระบบอีกชนิดหนึ่งนั้นเรียกว่า ระบบรอข่าวสาร (Message-waiting System) ในอุปกรณ์ปลายบางชนิดนั้นจะมีไฟเตือน (Indicator Lamp) หรือเสียงเตือนติดตั้งอยู่ เพื่อแสดงว่าคอมพิวเตอร์มีข่าวสารรออยู่เพื่อการส่งมายังอุปกรณ์ปลายนั้นเสียงเตือนหรือไฟเตือนนี้เกิดขึ้นจากการที่อุปกรณ์ปลายนั้นได้รับข่าวสารที่เป็นตำแหน่งของมัน ซึ่งข่าวสารนี้แสดงว่าคอมพิวเตอร์มีข่าวสารรออยู่ ทำให้โอเปอร์เรเตอร์ทราบ และเมื่อโอเปอร์เรเตอร์เสร็จสิ้นจากการเตรียมข่าวสารต่างๆ ที่กำลังกระทำอยู่และพร้อมที่จะรับข่าวสารที่มาจากคอมพิวเตอร์ โอเปอร์เรเตอร์ก็จะกดปุ่มเกี่ยวกับการรอข่าวสาร จากนั้นก็เข้าสู่สถานะการโพลของอุปกรณ์ปลายนั้น อุปกรณ์ปลายจะส่งสัญญาณไปยังคอมพิวเตอร์ เพื่อบอกว่าพร้อมที่จะรับข่าวสาร คอมพิวเตอร์จึงจะเริ่มส่งข่าวสารมายังอุปกรณ์ปลายนั้นโดยใช้วิธีการเลือกแบบรวดเร็ว (Fast Select)

2.11 สวิตช์แม่เหล็กไฟฟ้าหรือรีเลย์ (Relay)

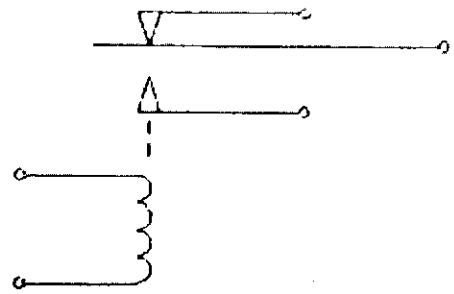
รีเลย์เป็นสวิตช์ที่ทำงานด้วยไฟฟ้า และมีการใช้งานอย่างขว้างในอุตสาหกรรม รีเลย์อาจแบ่งออกเป็น 2 ประเภทตามการใช้งาน คือ รีเลย์ควบคุม (Control Relays) และรีเลย์กำลัง (Power Relays)

รีเลย์ควบคุมเป็นรีเลย์ที่ใช้การควบคุมวงจรไฟฟ้ากำลังต่ำ หรือรีเลย์ตัวอื่นในวงจรการทำงานแบบอัตโนมัติ ส่วนรีเลย์กำลัง บางครั้งเรียกว่า “คอนแทกเตอร์” (Contactors) เป็นรีเลย์ที่ใช้กับระบบไฟฟ้ากำลังสูง เช่น การควบคุมมอเตอร์ เป็นต้น



(ก) โครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

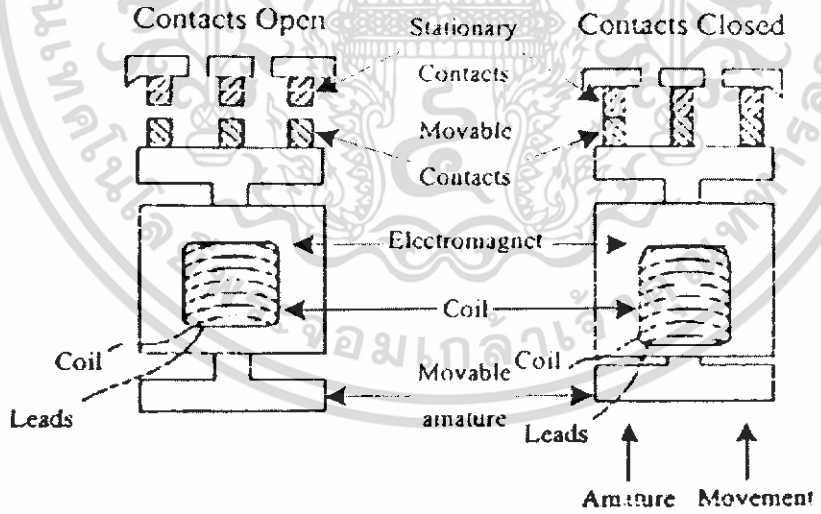


(ข) สัญลักษณ์

รูปที่ 2.53 รีเลย์ (Relays)

จากรูปที่ 2.53 รีเลย์ประกอบด้วยขดลวดที่พันอยู่รอบแกนแท่งเหล็ก หน้าสัมผัสที่อยู่กับที่ (Stationary Contact) ซึ่งยึดติดกับอาร์เมเจอร์ (Armature)

การทำงานของรีเลย์ เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดที่พันรอบแท่งเหล็กอันทำให้เกิดมีอำนาจแม่เหล็กดึงอาร์เมเจอร์ ซึ่งเป็นแท่งเหล็กอ่อนเข้ามาหาตัวมัน และทำให้หน้าสัมผัสทั้งสองแตะกันวงจรทางด้านโหลดจะปิดวงจร ขณะเมื่อไม่มีกระแสไฟไหลผ่านขดลวด แท่งเหล็กจะหมดอำนาจแม่เหล็กสปริงก็จะดันอาร์เมเจอร์กลับมาสู่ที่ตำแหน่งปกติตามเดิม อันทำให้หน้าสัมผัสทั้งสองแยกออกจากกันและวงจรทางด้านโหลดจะเปิดวงจร



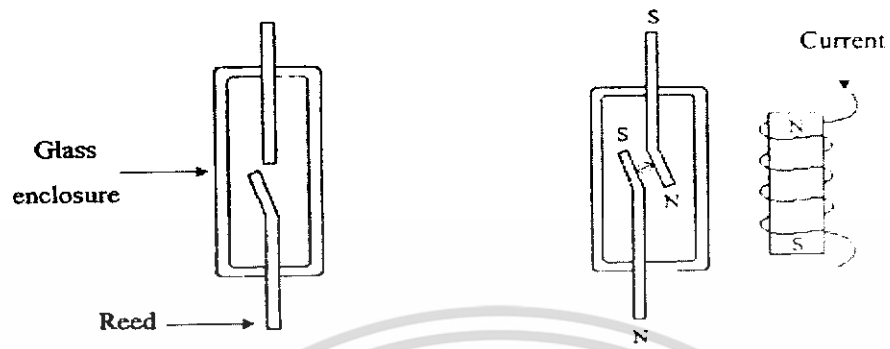
(ก) ขั้วสัมผัสเปิดวงจร (ข) ขั้วสัมผัสปิดวงจร

รูปที่ 2.54 แมกเนติกคอนแทกเตอร์ (Magnetic Contactors)

จากรูปที่ 2.54 เป็นรีเลย์กำลังที่ใช้ในการควบคุมมอเตอร์ในงานอุตสาหกรรม ที่เรียกว่า "แมกเนติก คอนแทกเตอร์" นอกจากนี้ยังมีรีเลย์อีกชนิดหนึ่งที่นิยมใช้งานกันอย่างมากคือหรีดรีเลย์ (Reed Relay)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรีดรีเลย์ เป็นรีเลย์ที่ประกอบด้วยขั้วสัมผัสของโลหะแผ่นบางๆ 2 ชิ้น ที่มีความไว ที่มีความไวต่อสนามแม่เหล็ก ซึ่งบรรจุอยู่ในหลอดแก้ว หน้าสัมผัสนี้เรียกว่า "หวีดสวิตซ์" (Reed Switch) และภายนอกหลอดแก้วจะมีขดลวดพันรอบตัวมัน ดังแสดงในรูปที่ 2.55



(ก) หรีดรีเลย์ (ข) การทำงานของหรีดสวิตซ์

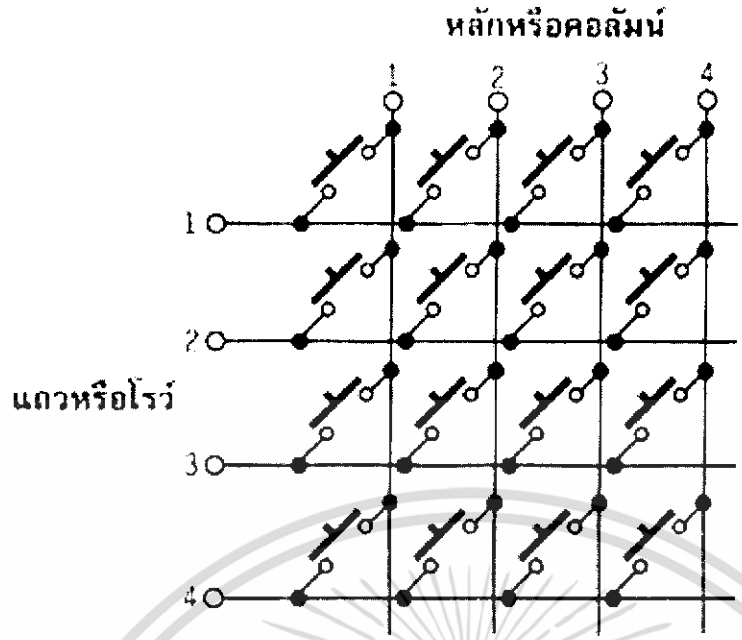
รูปที่ 2.55 หรีดรีเลย์ (Reed Relay)

การทำงานของหรีดรีเลย์จะเหมือนกับรีเลย์ทั่วไป คือ เมื่อมีกระแสไฟฟ้าไหลผ่านขดลวดรีเลย์จะทำให้เกิดสนามแม่เหล็กขึ้นและมีผลทำให้หรีดสวิตซ์ภายในหลอดแก้วแตะกันได้ดังแสดงในรูปที่2.55 (ข) ขณะที่หรีดสวิตซ์แตะกันย่อมทำให้เกิดการอาร์กที่ขั้วสัมผัสทั้งสอง และมีผลทำให้ขั้วสัมผัสเกิดความเสียหาย การป้องกันการอาร์กของขั้วสัมผัสทำได้โดยการบรรจุโลหะเหลวที่เป็นปรอทไว้ภายในหลอดแก้วเพื่อขจัดปัญหาที่เกิดขึ้น

2.12 การใช้งานคีย์สวิตช์หรือคีย์แพด (Keypad)

เพื่อให้ง่ายต่อการนำไมโครคอนโทรลเลอร์ไปใช้งาน ดังนั้นการทำงานของคีย์แพดจึงมีความจำเป็น โดยที่คีย์สวิตช์แบบเมทริกต์เพื่อใช้ในรับค่าอินพุตต่างๆ และนำค่าที่รับมานั้นไปใช้กับไมโครคอนโทรลเลอร์ คีย์แพดมีอยู่ด้วยกันหลายแบบแต่แบบที่ง่ายต่อการนำไปใช้งานก็ คือคีย์แพดขนาด 3 ×4 คีย์ ดังรูปที่ 2.57 ซึ่งจะมีการตรวจสอบว่าสวิตช์ใดถูกกดนั้นจะทำโดยการส่งค่าออกไปที่พอร์ทเอาท์พุตให้เป็นลอจิก "0" แล้วอ่านค่าเข้ามาใหม่ การส่งค่าที่สแกนได้ไปที่พอร์ทเอาท์พุตนั้นจะต้องทำอย่างรวดเร็วเพื่อที่จะตรวจสอบการกดสวิตช์ให้ทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.56 วงจรของสวิตช์แบบเมตริกซ์หรือคีย์แพด

รูปที่2.57 รูปคีย์สวิตช์หรือคีย์แพดแบบ 3 × 4 คีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13 ระบบฐานข้อมูล

ระบบฐานข้อมูล คือ ระบบที่มีการนำเอาข้อมูลที่เกี่ยวข้องมาจัดเก็บไว้ด้วยกันอย่างเป็นระเบียบ เพื่อประยุกต์ใช้งานหลายๆ โดยเฉพาะงานที่จำเป็นต้องใช้ข้อมูลร่วมกัน เป็นการลดความซ้ำซ้อนในการก็บข้อมูล แต่ทั้งนี้ไม่ได้หมายความว่าในองค์กรหนึ่งๆ จะต้องมีระบบฐานข้อมูลเพียงระบบเดียว แต่หมายความว่าข้อมูลทั้งหมดที่อยู่ในฐานข้อมูลจะต้องมีความสัมพันธ์กัน

โดยสรุปแล้วเป้าหมายของฐานข้อมูลมีดังนี้

- เพื่อลดความซ้ำซ้อนในการจัดเก็บข้อมูล
- เพื่อให้ผู้ใช้สามารถใช้ข้อมูลร่วมกัน
- ทำการใช้และเปลี่ยนแปลงแก้ไขข้อมูลเป็นไปอย่างสะดวกและถูกต้อง
- ลดค่าใช้จ่ายในการจัดเก็บและเรียกใช้ข้อมูล
- เพื่อความถูกต้องและความคงเส้นคงวา
- ให้ความปลอดภัยในการใช้ระบบ คือ อนุญาตให้เฉพาะผู้มีสิทธิใช้ได้เท่านั้น

2.13.1 ความหมายของระบบฐานข้อมูล

ระบบฐานข้อมูลเป็น โครงสร้างสารสนเทศ (Information) ที่ประกอบด้วยข้อมูลต่างๆ เช่น ข้อมูลของลูกค้าของบริษัท ข้อมูลรายการสินค้า เป็นต้น ซึ่งข้อมูลเหล่านี้สามารถรวมเป็นหัวข้อที่สัมพันธ์กันได้ ระบบฐานข้อมูลได้สร้างวิธีการสำหรับการรวบรวมรายการ ข้อมูลที่เกี่ยวข้องกับงานนั้นๆ เข้าด้วยกัน สร้างวิธีสำหรับการเก็บและบำรุงรักษาข้อมูลเหล่านั้นระบบฐานข้อมูลประกอบด้วยส่วนประกอบ 2 ส่วน คือ

2.13.1.1 ระบบจัดการฐานข้อมูล (Database Management System (DBMS)) เป็นโปรแกรมที่ใช้ เพื่อจัดระเบียบและบำรุงรักษารายการของข้อมูลเหล่านี้

2.13.1.2 แอปพลิเคชันฐานข้อมูล (Database Application) เป็นโปรแกรมที่ช่วยให้เราดูและแก้ไข ข้อมูลที่ถูกเก็บอยู่ใน DBMS โดยปกติทั้ง DBMS และ Database Application จะทำงาน อยู่บนเครื่องเดียวกัน ส่วนมากทั้งสอง ส่วนจะถูก รวมอยู่ภายในโปรแกรมเดียวกัน

2.13.2 ระบบการจัดการฐานข้อมูล

การใช้ระบบฐานข้อมูลนั้น ผู้ใช้ (หรือ โปรแกรม) จะไม่เข้าไปใช้ข้อมูลในระบบโดยตรง แต่จะทำงานผ่านทางซอฟแวร์ที่เป็นตัวจัดการระบบที่เรียกกันว่า ระบบจัดการฐานข้อมูลหรือ ดีบีเอ็มแอส (DBMS: Database Management System) โดยในการเรียกใช้ข้อมูลในระบบสามารถใช้ภาษาถามซึ่งส่วนจะเป็นภาษาที่มีไวยากรณ์คล้ายๆ กับภาษาอังกฤษทั่วไป หรืออาจจะเขียนเป็นโปรแกรมด้วยภาษาที่เราชำนาญ เช่น โคบอล หรือ C ก็ได้

ดีบีเอ็มแอสมีวิธีในการดูและจัดการข้อมูลโดยใช้สิ่งที่เรียกว่า พจนานุกรมข้อมูล (data dictionary) ซึ่งเป็นองค์ประกอบส่วนหนึ่งของระบบฐานข้อมูล พจนานุกรมข้อมูลเป็นแหล่งเก็บโครงสร้างของข้อมูลในระบบ ได้แก่ ชนิดและขนาดของข้อมูล ลักษณะ การใช้งานของข้อมูลแต่ละส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดเก็บเชิงกายภาพ รายนามผู้มีสิทธิในการใช้ข้อมูล เป็นต้น นั่นก็คือ ในการเรียกใช้ข้อมูลแต่ละครั้ง DBMS จะอาศัยสิ่งที่อยู่ในพจนานุกรมนี้เป็นแนวทางการจัดการ เช่น

- จะทำให้การเช็คพจนานุกรมก่อนว่าฟิลด์ข้อมูลที่กำลังจะเติมเข้าไปนั้น มีตัวตนอยู่ในระบบหรือยัง
- ในการแก้ไขขนาดของข้อมูล เช่น แก้ฟิลด์ชื่อพนักงานจากตัวอักษร 20 หลัก เป็น 25 หลัก ก็จะกระทำที่ส่วนพจนานุกรมข้อมูล โดยไม่ต้องเข้าไปวุ่นวายกับส่วนที่เป็นข้อมูลจริง
- ใช้ในการตรวจดูความสัมพันธ์ระหว่างข้อมูลส่วนต่างๆ เช่น ถ้าได้มีกำหนดความสัมพันธ์ไว้ว่า พนักงานขายมีความสัมพันธ์กับลูกค้า ดังนั้นค่าข้อมูลของพนักงานขายในเรคอร์ดของลูกค้าจะต้องเป็นค่าที่มีตัวตนอยู่ในเรคอร์ดของพนักงานขาย เป็นต้น

2.13.3 Relational Database

เป็น DBMS ประเภทหนึ่งที่นิยมใช้ในปัจจุบัน ในระบบ RDM ข้อมูล จะถูกจัดระเบียบเป็นเซตในทางคณิตศาสตร์ในโครงสร้างของตาราง ฟิลด์ข้อมูลแต่ละตัวจะเป็นคอลัมน์ในตาราง และแต่ละเรคอร์ดจะกลายเป็นแถวในตาราง Relation Database มีข้อดีที่สำคัญคือ ความยืดหยุ่นที่สมบูรณ์ในการกล่าวถึง ความสัมพันธ์ระหว่างรายการข้อมูลที่แตกต่างกัน การตัดสินใจหลักๆ ของผู้ออกแบบฐานข้อมูลคือการกำหนดตาราง Relation Database

ข้อดีในการใช้โปรแกรมระบบจัดการฐานข้อมูล

(1) ลดความซ้ำซ้อนของฐานข้อมูล

(2) ลดความผิดพลาดของข้อมูล ในกรณีที่เรามีการจัดเก็บข้อมูลไว้หลายๆ แห่ง การเปลี่ยนแปลงแก้ไขข้อมูลจะเกิดปัญหาว่าข้อมูลไม่เหมือนกัน ทำให้การทำงานผิดพลาดและเกิดความเสียหายได้ ซึ่งระบบฐานข้อมูลสามารถลดปัญหาเหล่านี้ได้อย่างดี

(3) สามารถใช้ข้อมูลร่วมกันได้ ทำให้ประหยัดค่าใช้จ่ายและลดความผิดพลาด

(4) ควบคุมความเป็นมาตรฐานของข้อมูลได้การจัดเก็บข้อมูลที่มีศูนย์กลางที่เดียวทำให้สามารถควบคุมรูปแบบของข้อมูลได้ทำให้ข้อมูลเป็นอันหนึ่งอันเดียวกัน ง่ายต่อการดูแลรักษา

(5) ความปลอดภัยในการเก็บรักษาข้อมูล เนื่องจากการเก็บข้อมูลไว้ที่ศูนย์กลางข้อมูลที่เดียวจึงทำให้สามารถควบคุมและจัดสรรระดับของผู้ใช้ที่เข้ามาขอใช้ข้อมูลได้ เช่นการกำหนดรหัสผ่านให้การเข้ามาใช้ข้อมูล แบ่งระดับความปลอดภัยของผู้เข้ามาใช้ข้อมูล

2.13.4 โครงสร้างของฐานข้อมูล

โครงสร้างของข้อมูลในฐานข้อมูลสามารถมองได้ 2 แง่มุม คือ เป็นโครงสร้างเชิงกายภาพ (Physical data structure) ได้แก่ วิธีการจัดเก็บข้อมูลจริงๆ ในสื่อ เช่น ในดิสก์ เนื้อหาเกี่ยวกับการจัดเก็บแบบกายภาพนี้เป็นเรื่องจำเป็นสำหรับผู้ที่จะออกแบบและพัฒนาดีบีเอ็มแอส (DBMS) แต่สำหรับผู้ใช้ทั่วไป รวมไปถึงผู้วิเคราะห์ระบบแล้ว โครงสร้างเชิงตรรกะ (logical data structure) ดูเป็นเรื่องที่จำเป็นต้องให้ความสนใจมากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างเชิงตรรกะ ได้แก่ โครงสร้างที่ผู้ใช้ระบบมองดูว่าการจัดเก็บและความสัมพันธ์ระหว่างส่วนต่างๆ ของข้อมูลในระบบดูเหมือนจะเป็นอย่างนั้น ในการนิยามโครงสร้าง หรือ โมเดล (Model) นั้น กระทำในสิ่งที่เรียกว่า เค้าโครง (schema) กล่าวคือ หลังจากที่เราได้ออกแบบแล้วว่าระบบฐานข้อมูลควรจะประกอบด้วยข้อมูลหลักๆ ซึ่งควรจะประกอบไปด้วยฟิลด์อะไรบ้าง และแต่ละฟิลด์มีชนิดและขนาดเป็นอย่างไรและได้กำหนดความสัมพันธ์ระหว่างข้อมูลได้แล้ว ก็จะทำการกำหนดโครงสร้างเชิงตรรกะที่ได้ออกแบบไว้ด้วยการสร้างเค้าร่างขึ้นมา

2.13.5 การออกแบบฐานข้อมูล

ปัจจุบันในการออกแบบฐานข้อมูลที่รู้จักกันอย่างแพร่หลาย ได้แก่ นอมัลไรเซชั่น (Normalizatin) ซึ่งเป็นวิธีที่มีแนวความคิดในการปรับปรุง คุณสมบัติของ รีเลชันขึ้นเป็นขั้นตอนอย่างมีระบบเพื่อลดโอกาสที่จะเกิดความผิดพลาดเนื่องจากการเปลี่ยนแปลงข้อมูล การออกแบบฐานข้อมูลด้วยวิธีนี้ เริ่มต้นด้วยการกำหนด universal relation พร้อมทั้งกำหนดความสัมพันธ์ระหว่างแอทริบิวท์ (attribute) ต่างๆ ของยูนิเวอร์เซล รีเลชัน (universal relation) ในรูปของฟังชั่น ดีเพนแดนซี (function dependency), มัลติเวลู ดีเพนแดนซี (multivalued dependency) และจอยน์ ดีเพนแดนซี (joindependancy) แล้วทำให้ รีเลชัน ในแต่ละขั้นตอนมีคุณสมบัติตรงตามที่กำหนดไว้ ตั้งแต่ขั้นตอนที่ 1 ถึง ขั้นตอนที่ 5 ผลที่ได้ใน ขั้นตอนสุดท้ายจะได้ fifth normal (5nf)

ถึงแม้วิธีการนอมัลไรเซชัน (Normalization) นี้จะเป็นวิธีที่มีขั้นตอนที่เป็นระบบ แต่ก็เป็นวิธีที่ค่อยข้างยุ่งยากและซับซ้อน โดยเฉพาะอย่างยิ่งเมื่อระบบงานมีขนาดใหญ่ แอทริบิวท์มีจำนวนมาก การออกแบบด้วยวิธีนอมัลไรเซชันจึงเป็นเรื่องลำบาก

นอกจากวิธีนอมัลไลเซชัน แล้วยังมีวิธีการออกแบบฐานข้อมูลแบบรีเลชัน อีกวิธีหนึ่งโดยใช้ในแอม (NIAM: missus's information system analysis method) เป็นเครื่องมือในการออกแบบ โดยแบบจำลองข้อมูลในแอมเป็นแบบจำลองข้อมูลระดับแนวคิดซึ่งมีพื้นฐานมาจากภาษาธรรมชาติ ให้รูปประโยคที่มีประธาน กริยา กรรม วิธีแสดงรูปแบบความสัมพันธ์ของข้อมูลและข้อจำกัดของข้อมูลได้อย่างชัดเจน นอกจากนั้นยังสามารถแปลง Conceptual schema เป็น relational database schema ซึ่งอยู่ในรูปของ fifth normal form และเนื่องจากวิธีการนี้ใช้รูปสัญลักษณ์ที่แสดงความสัมพันธ์ของข้อมูลและง่ายต่อความเข้าใจ ดังนั้นจึงจะสะดวกในการออกแบบฐานข้อมูลของระบบงานใหญ่ๆ

2.13.6 ภาษาเอสคิวแอล (SQL: structured Query Language)

ภาษาเอสคิวแอลเป็นภาษายุคที่ 4 ภาษาหนึ่งซึ่งพัฒนาโดยบริษัทไอบีเอ็ม (IBM) ในปัจจุบันเป็นที่แพร่หลายกันมาก เป็นภาษาที่คล้ายกับภาษาอังกฤษ ในการประมวลข้อมูลก็สามารถที่จะเรียนรู้ โครงสร้างพื้นฐานของภาษาเอสคิวแอลนั้น จะให้คำสั่งซึ่งมีความสามารถ และมีความสมบูรณ์ในตัวในการดำเนินงานได้อย่างดี

ดังนั้นภาษาเอสคิวแอลจึงกลายเป็นภาษาร่วมกันระหว่างผู้ใช้งานธรรมดา กับผู้ที่มีความรู้ประสบการณ์ในการประมวลผล แต่สำหรับผู้ใช้ทั่วไปแล้ว คงเป็นการลำบากที่จะใช้ภาษาเอสคิวแอลในการสร้างคำถามที่ซับซ้อนได้ ดังนั้น ตามความเป็นจริงแล้ว ในปัจจุบัน ภาษาเอสคิวแอล ใช้โดยบุคคล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในวงการระดับที่มีความรู้และประสบการณ์ในการณ์ในการประมวลผล, นักพัฒนาระบบงาน, นักบริหารข้อมูล (DBA: database administrator)

ภาษาเอสคิวแอล เป็นภาษาฐานข้อมูลแบบเชิงสัมพันธ์ (Relational database) คือประกอบด้วยตาราง (table) หลายตาราง และในตารางหนึ่งมี 2 มิติ ได้แก่ หลัก (columns) ในแนวตั้งและแถว (rows) ในแนวนอน

การใช้ภาษาเอสคิวแอล กระทำได้ 3 วิธี ได้แก่

1. ออกคำสั่งแบบออนไลน์ กล่าวคือ ผู้ใช้สามารถพิมพ์ประโยคคำสั่งผ่านทางเทอร์มินัล (Terminal) โดยที่คำสั่งเหล่านี้จะถูกปฏิบัติงานโดยทันที
2. ส่งคำสั่งในลักษณะงานออฟไลน์ (off line) หรืองาน batch ลักษณะการใช้งานประเภทนี้เหมาะกับการสร้างรายงาน หรือประเภทของงานที่ไม่จำเป็นต้องการทราบผลทันที
3. สอดแทรกประโยคคำสั่งไว้ในโปรแกรมประยุกต์ที่เขียนขึ้นมาสำหรับการใช้ระบบฐานข้อมูล ซึ่งโปรแกรมประยุกต์เหล่านี้อาจจะเขียนด้วยภาษา โคบอล ฟอร์แทรน หรือภาษาซี ก็ได้

ภาษาเอสคิวแอล มีประเภทคำสั่งโดยสรุป ดังนี้

- การคิวรี (query) เป็นการสอบถามข้อมูลจากฐานข้อมูล
- การดำเนินงานกับข้อมูล (data manipulation) ได้แก่ การเพิ่มเติม (insert), การลบ (delete) และการแก้ไข (update) ข้อมูลในฐานข้อมูล
- การกำหนดลักษณะของข้อมูล (data definition) ได้แก่ การกำหนดตาราง (table), วิว (view) และดัชนีในการค้นหา (indexes) ในฐานข้อมูล
- การควบคุมข้อมูล (data control) ได้แก่ การป้องกัน ควบคุมข้อมูลให้ปลอดภัย จากผู้ใช้แต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งสำหรับเรียกดูข้อมูล	
SELECT	เรียกดูข้อมูลจากตารางได้จาก 1 หรือมากกว่า 1 ตาราง
คำสั่งสำหรับการใช้ข้อมูล	
INSERT	เติมข้อมูล 1 แถวหรือมากกว่า 1 แถว ลงตาราง
UPDATE	แก้ไขข้อมูลใน 1 แถว หรือมากกว่า 1 แถว
DELETE	ลบข้อมูล 1 แถว หรือมากกว่า 1 แถว จากตาราง
คำสั่งสำหรับนิยามข้อมูล	
CREATE TABLE	นิยามโครงสร้างของข้อมูลในตารางที่จะสร้างขึ้นใหม่
DROP TABLE	ลบโครงสร้างของตารางข้อมูลออกจากระบบ
ALTER TABLE	เพิ่มคอลัมใหม่ลงในตาราง
REATE INDEX	สร้างดัชนีของตารางซึ่งประกอบไปด้วย 1 คอลัมน์หรือมากกว่า
DROP INDEX	ยกเลิกดัชนีออกจากระบบ
CREATE VIEW	นิยามวิวของผู้ใช้
DROP VIEW	ลบวิวออกจากระบบ

ตารางที่ 2.14 สรุปคำสั่งของ ภาษาเอสคิวแอล

การเรียกดูข้อมูล ประโยคคำสั่งสำหรับการเรียกดูข้อมูลได้แก่ คำสั่ง Select ซึ่งมีรูปแบบคำสั่งดังนี้

SELECT ...
FROM (ชื่อตาราง)
WHERE (เงื่อนไข)
ORDER BY ...

ซึ่งผู้ใช้สามารถระสิ่งต่างๆ ในคำสั่งได้ดังนี้

- ส่วนหลัง SELECT ใช้ระบุ คอลัมน์หรือกลุมของคอลัมน์ที่เราต้องการดูข้อมูล
- ส่วนหลัง FROM ใช้ระบุ ชื่อของตารางที่เราต้องการดูข้อมูล
- ส่วนหลัง WHERE ใช้ระบุ เงื่อนไขของข้อมูลที่เราสนใจซึ่งอาจจะมีกรือไม่มีก็ได้
- ส่วนหลัง ORDER BY ใช้ระบุ วิธีการเรียงลำดับการแสดงผลข้อมูล ซึ่งอาจจะมีหรือไม่มีก็ได้

ตัวอย่างง่ายๆ เช่น

SELECT * FROM ชื่อตาราง WHERE เงื่อนไข

ซึ่งจะทำการแสดงข้อมูลทั้งหมดจากตารางที่ระบุออกมา ขอให้สังเกตว่าเราใส่เครื่องหมาย "*" ไว้ข้างหลัง SELECT หมายความว่าให้เรียกดูทุกคอลัมน์นั่นเอง โดยระบุเงื่อนไขของข้อมูลหลัง WHERE เฉพาะแถวที่ต้องการเรียกดู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14 สมาร์ทการ์ดกับการประยุกต์ใช้งานด้านต่าง ๆ

2.14.1 การประยุกต์ใช้งานด้านการสื่อสาร (SmartCard Application in telecommunication)

2.14.1.1 ใช้ในการบริการโทรศัพท์สาธารณะ (PUBLIC PAYPHONES) เป็นการนำสมาร์ทการ์ดไปใช้ในรูปแบบบัตรโทรศัพท์ (Phone Card) เนื่องจากมีข้อได้เปรียบบัตรโทรศัพท์ประเภทอื่น คือ

- มีความปลอดภัยสูง ถูกป้องกันโดยระบบความปลอดภัยในบัตรชนิดหน่วยความจำและชนิดไมโครโปรเซสเซอร์
- มีราคาถูกเมื่อเทียบกับประโยชน์ที่ได้รับ (มีความน่าเชื่อถือได้สูง, ระบบและเครื่องอ่านมีราคาถูก)

2.14.1.2 ใช้กับโทรศัพท์เคลื่อนที่ (MOBILE PHONES) เป็นการใช้สมาร์ทการ์ดในลักษณะการเก็บเบอร์โทรศัพท์ของผู้ใช้ และรหัสผ่านลงในบัตร โดยเป็นการเอื้อประโยชน์แก่ผู้ใช้บัตรดังนี้

- มีความปลอดภัยสูง และมีระบบการล็อคเครื่องเมื่อกดรหัสผิด
- ใช้เป็นบัตรอ้างอิงของผู้ใช้เมื่อใช้เมื่อใช้งานกับเครื่องอื่นๆได้โดยไม่ต้องเปลี่ยน เบอร์ เช่น เมื่อเกิดความเสียหายกับเครื่องโทรศัพท์ของตน สามารถนำบัตรไปใช้งานกับเครื่องอื่นๆได้โดยไม่ต้องเปลี่ยนเบอร์ หรือ ในกรณีการใช้งานต่างประเทศ สามารถนำบัตรไปใช้งานกับเครื่องที่ต่างประเทศได้ โดยไม่ต้องนำเครื่องไปและไม่ต้องเปลี่ยนเลขหมายในการใช้งาน

2.14.2 การประยุกต์ใช้งานด้านการเงิน การธนาคาร (Finance & Banking)

2.14.2.1 ใช้เป็นบัตรที่ใช้แทนเงินสด (Cash Card) เพราะให้ความสะดวกและความมั่นใจในการป้องกันการโกงและการปลอมแปลงบัตร ดังนั้นในต่างประเทศจึงใช้แทนเงินสดในการชำระค่าสินค้าหรือบริการที่มีมูลค่าไม่สูงนัก แต่จะเน้นทางด้านความสะดวกเป็นหลัก ซึ่งโดยทั่วไปแล้ว บัตรที่ใช้แทนเงินสดได้นี้สามารถแบ่งออกได้เป็น 2 ประเภท ได้แก่

- *บัตรประเภทเติมเงินได้ (Electronic Purse Card)* สามารถเติมจำนวนเงินลงไปใหม่ได้เรื่อยๆ (Refillable) คือ เป็นกระเป๋าอิเล็กทรอนิกส์ ซึ่งโดยมากจะออกโดยธนาคารเพื่อทำการโอนเงินไปมาได้ระหว่างบัญชีและบัตร
- *บัตรประเภทเติมเงินไม่ได้ (Token Card)* ถ้าใช้หมดแล้วไม่สามารถเติมเงินลงไปใหม่ได้อีก คือ ใช้ได้ครั้งเดียว (Disposable)

2.14.2.2 ใช้เป็นบัตรเครดิต (Credit Card) เป็นการนำเอาสมาร์ทการ์ดไปใช้ในลักษณะบัตรเครดิต โดยใช้ร่วมกับแถบแม่เหล็ก เพื่อทำให้เกิดความปลอดภัยยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.14.3 การประยุกต์ใช้งานโดยใช้เป็นบัตรประจำตัวต่างๆ (Identification Card)

2.14.3.1 บัตรพนักงาน (Company Card) ใช้เป็น

- บัตรอ้างอิงของพนักงาน เก็บประวัติส่วนตัว,ประวัติสุขภาพและประวัติการทำงาน
- เวลาในการเข้า-ออกงาน
- บัตรเข้า-ออก ในบางบริเวณ

2.14.3.2 บัตรนักศึกษา ใช้เป็นบัตรอ้างอิงของนักศึกษา เก็บประวัติการศึกษาและการใช้บริการอื่นๆ ในสถานศึกษา เช่น ชำระค่าลงทะเบียน การใช้บริการห้องสมุด การใช้บริการห้องคอมพิวเตอร์ เป็นบัตรผ่านเข้า-ออกหอพัก ซึ่งการใช้บัตรสมาร์ทการ์ดในงานด้านต่างๆนั้นนอกจากจะช่วยอำนวยความสะดวกแล้ว ยังเป็นการลดการใช้เจ้าหน้าที่ในการปฏิบัติงานและเป็นการลดการใช้กระดาษในงานต่างๆ ซึ่งเป็นการช่วยประหยัดทรัพยากรธรรมชาติอีกทางหนึ่งด้วย

2.14.4 การประยุกต์ใช้งานด้านการแพทย์

ใช้เป็นบัตรสุขภาพ (Health Card) ซึ่งจะมีประวัติการรักษาพยาบาล การแพ้ยา เป็นต้น นอกจากนี้ยังมีประโยชน์ คือ ใช้เป็นบัตรอ้างอิงตัวผู้ถือบัตรพร้อมประวัติทางสุขภาพในกรณีเกิดอุบัติเหตุฉุกเฉินได้

2.14.5 การประยุกต์ใช้งานด้านระบบรักษาความปลอดภัย

เป็นการนำเอาบัตรสมาร์ทการ์ดมาใช้ในการอ้างอิงผู้ถือบัตรและผ่านเข้า-ออก ในพื้นที่ (Access Control) เช่น การนำไปใช้เป็นบัตรประจำห้องพักของโรงแรม บัตรผ่านของพนักงาน สำหรับระบบที่ต้องการความปลอดภัยสูงมีการนำเอาสมาร์ทการ์ดไปใช้ร่วมกับการตรวจสอบลายนิ้วมือ

2.14.6 การประยุกต์ใช้งานด้านการขนส่งมวลชน

2.14.6.1 ในระบบการจอดรถ (Parking System) เป็นการใช้สมาร์ทการ์ดในการชำระเงินค่าจอดรถซึ่งเป็นตัวช่วยอำนวยความสะดวกในการชำระเงินโดยไม่ต้องมีขั้นตอนการทอนเงิน

2.14.6.2 ใช้ในรับการชำระค่าผ่านทางและค่าโดยสาร เช่น บัตรผ่านทางด่วน บัตรรถโดยสาร เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

หลังจากที่ได้ทำการศึกษาถึงทฤษฎีและรายละเอียดต่าง ๆ ของโครงสร้าง คุณสมบัติและวิธีการใช้งานของสมาร์ทการ์ด ในบทที่ 2 มาแล้วนั้น ในบทนี้จึงจะขอกล่าวถึงการออกแบบ ทั้งในส่วนของฮาร์ดแวร์ ซอฟท์แวร์ การเลือกใช้สมาร์ทการ์ด และการออกแบบวิธีการติดต่อกับชิพสมาร์ทการ์ด

3.1 วิธีการเลือกใช้สมาร์ทการ์ดและอุปกรณ์เชื่อมต่อกับหน้าสัมผัสของสมาร์ทการ์ด

3.1.1 การเลือกใช้สมาร์ทการ์ด

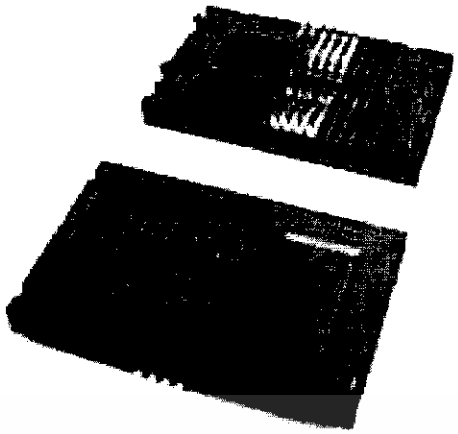
ในส่วนของการนำสมาร์ทการ์ดมาใช้งานนั้น ในโครงงานนี้ได้เลือกใช้สมาร์ทการ์ดชนิดที่มีการป้องกันหน่วยความจำ(Security Access Memory) โดยมีชิพไอซีเบอร์ SLE4442 ของบริษัทซีเมนส์ ฝังอยู่บนตัวบัตร เพื่อใช้บัตรสำหรับเก็บข้อมูลต่างๆ ตามที่ต้องการ การเลือกใช้บัตรเบอร์ SLE4442 ซึ่งจะมีหน่วยความจำสำหรับใช้ในการเก็บข้อมูลต่างๆ อย่างเหมาะสมเพื่อใช้ในการทำโครงงานครั้งนี้นั้น เนื่องมาจาก สามารถหาซื้อได้ง่ายตามท้องตลาดรวมทั้งมีราคาที่เหมาะสมที่จะนำมาประยุกต์ใช้งานและทำการพัฒนาต่อไปในอนาคตได้ โดยบัตรสมาร์ทการ์ดที่มีชิพไอซีเบอร์ SLE4442 บรรจุอยู่นั้น จะลักษณะดังรูปที่ 3.1

รูปที่ 3.1 ลักษณะสมาร์ทการ์ดที่มีชิพไอซีเบอร์ SLE4442 บรรจุอยู่

3.1.2 วิธีการเลือกใช้อุปกรณ์เชื่อมต่อหน้าสัมผัสของสมาร์ทการ์ด

การเชื่อมต่อเข้ากับหน้าสัมผัสของบัตรนั้นจะใช้ซ็อกเกตในการเชื่อมต่อ เพื่ออ่านข้อมูลหรือทำการถ่ายโอนข้อมูลลงบนบัตร โดยซ็อกเกตที่ใช้มีลักษณะดังรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ลักษณะซ็อกเกต ที่ใช้ในการเชื่อมต่อกับหน้าสัมผัสของสมาร์ทการ์ด

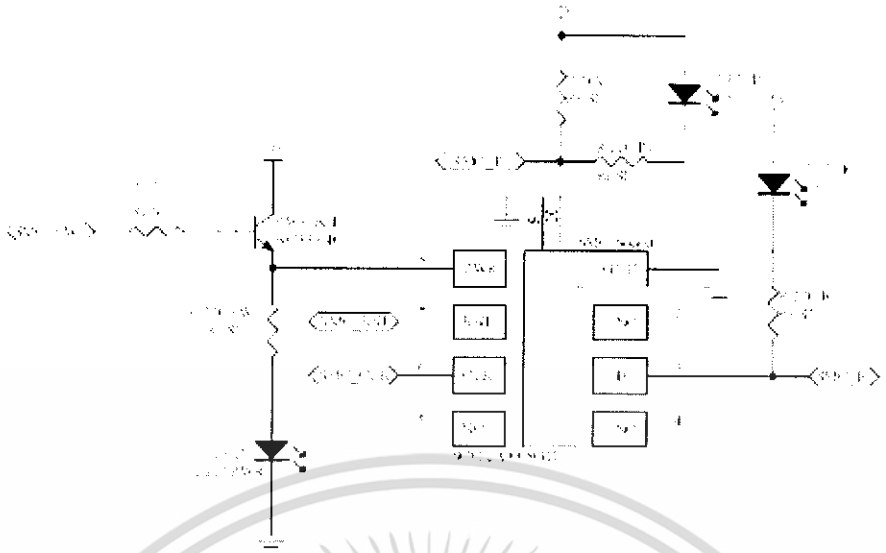
3.2 วิธีการออกแบบในส่วนของฮาร์ดแวร์

สำหรับการทำงานของเครื่องอ่านและเขียนสมาร์ทการ์ดนั้น จะต้องทำการออกแบบการทำงานของไมโครคอนโทรลเลอร์ เพื่อให้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ด โดยที่ส่วนของไมโครคอนโทรลเลอร์นั้น ในโครงงานนี้ได้ออกแบบให้ไมโครคอนโทรลเลอร์มีหน้าที่ ดังนี้ คือ เป็นตัวควบคุมกระบวนการในการอ่านและเขียนข้อมูลบนตัวบัตรสมาร์ทการ์ด ซึ่งกระบวนการนี้จะมีการทำงานโดยการควบคุมจากคอมพิวเตอร์ สรุปได้ว่า หน้าที่ที่ถือว่าเป็นหน้าที่ที่สำคัญที่สุดของไมโครคอนโทรลเลอร์ในโครงงานนี้ก็คือ เป็นอุปกรณ์หลักที่ใช้ในการควบคุมการทำงานของระบบทั้งหมดของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ด นั่นเอง

3.2.1 วิธีการออกแบบวงจรที่ใช้ในการติดต่อกับสมาร์ทการ์ด

วงจรที่ออกแบบสำหรับการใช้งานของซ็อกเกตเพื่อเชื่อมต่อเข้ากับหน้าสัมผัสของบัตร จะมีการออกแบบวงจรดังรูปที่ 3.3

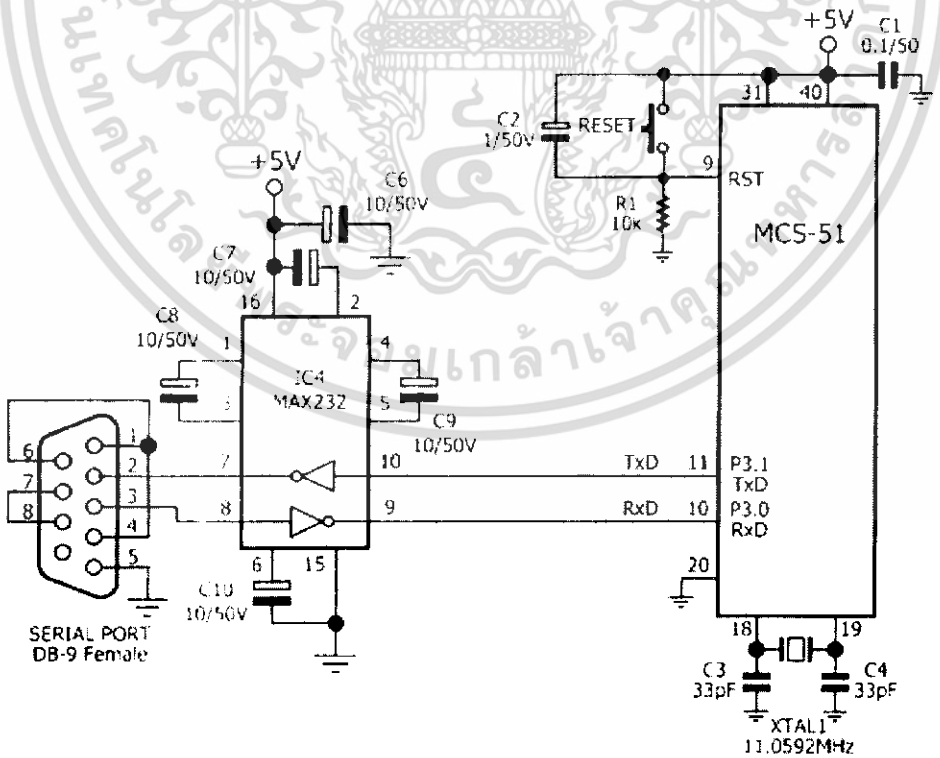
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรการใช้งานของซ็อกเกตที่ใช้ในการเชื่อมต่อกับหน้าสัมผัสของบัตร

3.2.2 วิธีการออกแบบการสื่อสารผ่านพอร์ทอนุกรม RS232 (Serial Port)

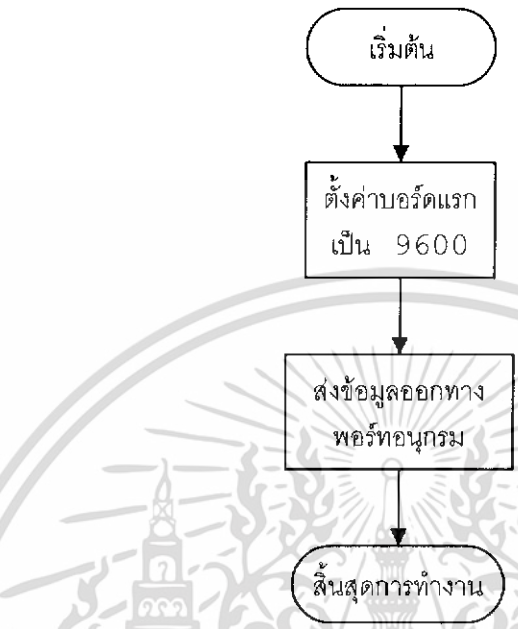
การออกแบบเพื่อให้ไมโครคอนโทรลเลอร์สามารถสื่อสารข้อมูลผ่านพอร์ทอนุกรมนั้นจะทำการออกแบบวงจรการทำงาน โดยก่อนที่จะส่งข้อมูลเข้าพอร์ทสื่อสารอนุกรมนั้น จะต้องทำการแปลงระดับแรงดันไฟจากทีทีแอล(TTL) ให้เป็นแรงดันไฟตามตามมาตรฐานของ RS232 ก่อนโดยมีการใช้ไอซีสำหรับการแปลงแรงดันเบอร์ IC MAX232 ดังวงจรในรูปที่ 3.4



รูปที่ 3.4 วงจรทดลองการสื่อสารข้อมูลผ่านพอร์ทอนุกรมของไมโครคอนโทรลเลอร์

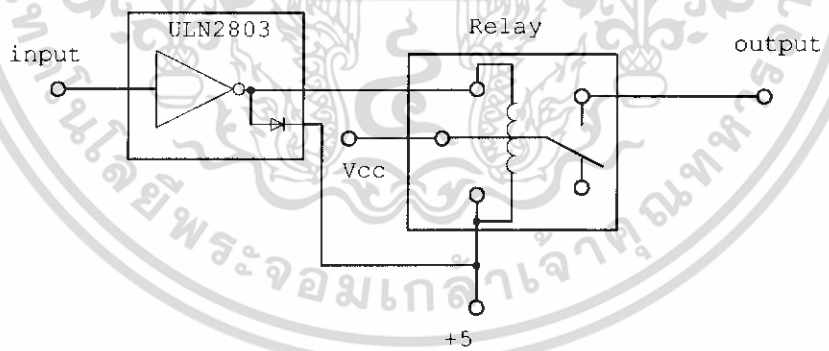
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการออกแบบเพื่อทดลองการสื่อสารข้อมูลผ่านพอร์ทอนุกรมแล้วนั้น ก็จะทำการเขียนโปรแกรมเพื่อทดสอบการทำงานของวงจรในรูปที่ 3.4 โดยมีการออกแบบกระบวนการเขียนโปรแกรมดังแผนผังการทำงานดังรูปที่ 3.5



รูปที่ 3.5 แผนผังการทำงานของการสื่อสารข้อมูลผ่านพอร์ทอนุกรม

3.2.3 วงจรขับรีเลย์ (Relay)



รูปที่ 3.6 วงจรขับรีเลย์

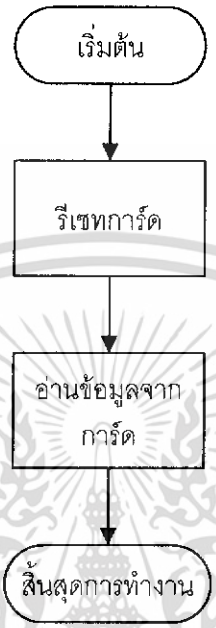
สัญญาณที่ออกมาจากเครื่องลูก(slave) จะเข้าที่ขาอินพุท ส่วนเอาต์พุทจะต่ออยู่กับกลอนประตู เมื่อมีสัญญาณเข้ามารีเลย์จะทำงาน Vcc จะไหลออกทางเอาต์พุททำให้กลอนประตูทำงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วิธีการออกแบบกระบวนการอ่านและเขียนข้อมูลของเครื่องอ่านเขียนบัตรสมาร์ทการ์ด

3.3.1 วิธีการออกแบบกระบวนการอ่านข้อมูลภายในบัตรสมาร์ทการ์ด

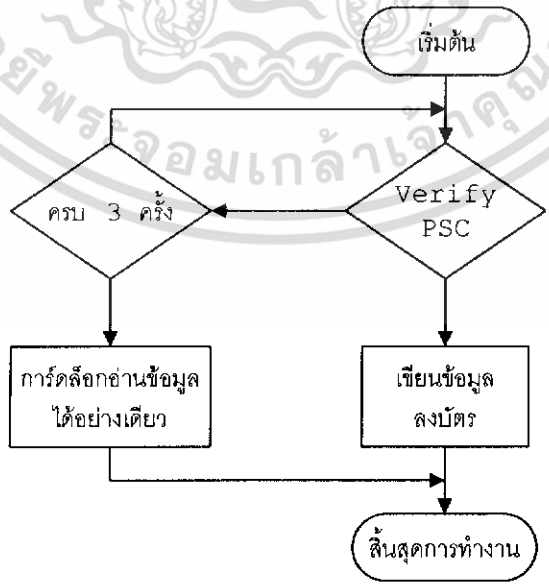
กระบวนการออกแบบเพื่อให้เครื่องอ่านและเขียนสมาร์ทการ์ด สามารถที่จะอ่านข้อมูลภายในสมาร์ทการ์ดได้นั้น จะมีการออกแบบแผนผังการทำงาน (Flow Chart) ดังรูปที่ 3.7



รูปที่ 3.7 แผนผังการทำงานของโปรแกรมกระบวนการอ่านข้อมูลภายในสมาร์ทการ์ด

3.3.2 วิธีการออกแบบกระบวนการเขียนข้อมูลลงบนสมาร์ทการ์ด

กระบวนการออกแบบเพื่อให้เครื่องอ่านและเขียนสมาร์ทการ์ด สามารถที่จะเขียนข้อมูลลงบนสมาร์ทการ์ดได้นั้น จะมีการออกแบบตามแผนผังการทำงาน (Flow Chart) ดังรูปที่ 3.8

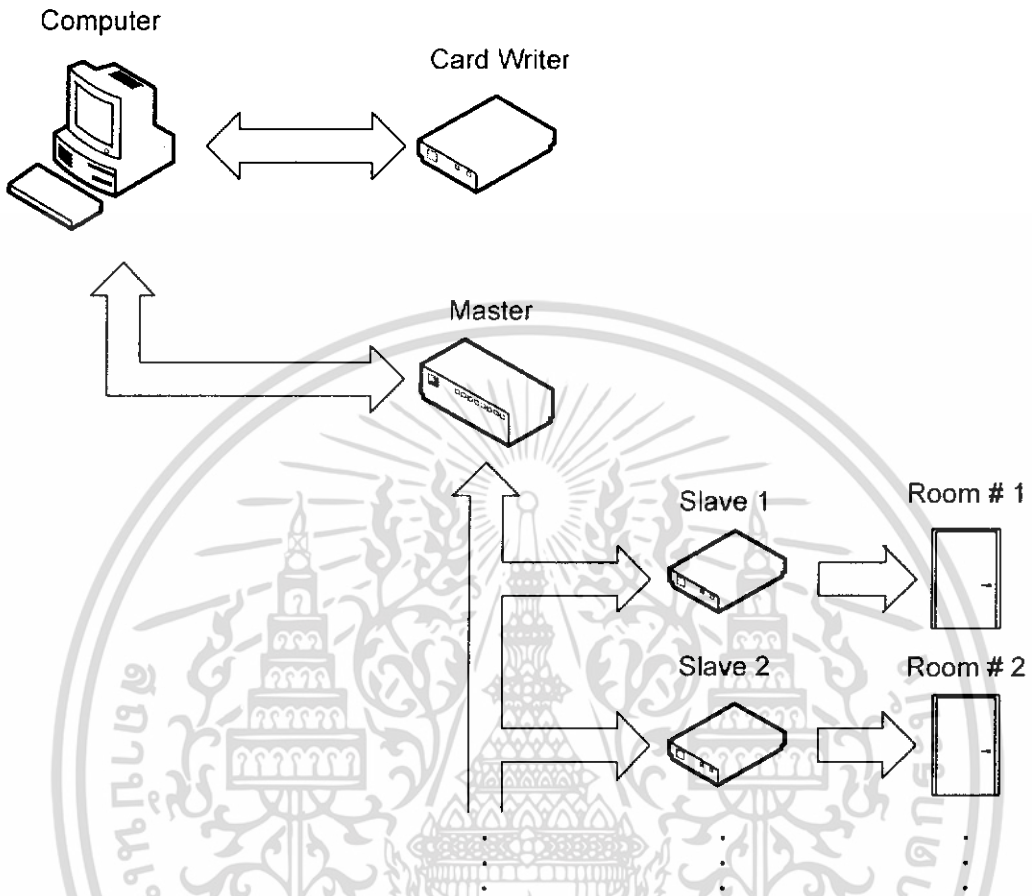


รูปที่ 3.8 แผนผังการทำงานของโปรแกรมกระบวนการเขียนข้อมูลลงบนสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 วิธีการออกแบบกระบวนการทำงานโดยรวมของโครงงานนี้

โครงสร้างและการทำงานโดยรวมของเครื่องอ่านและเขียนสมาร์ทการ์ดที่นำมาประยุกต์ใช้งานในระบบห้องพัก สามารถแสดงได้ ดังรูปที่ 3.9



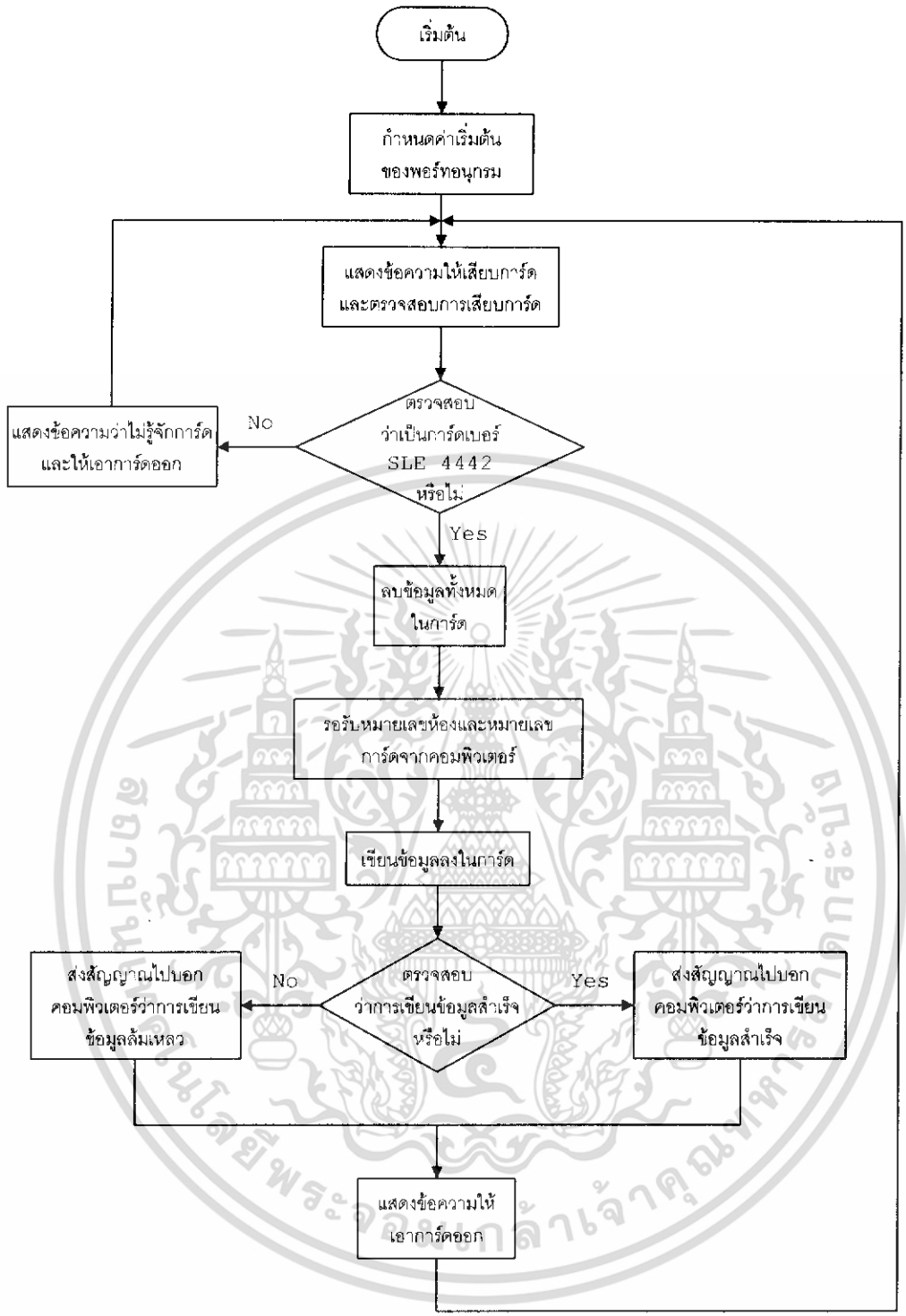
รูปที่ 3.9 โครงสร้างการทำงานโดยรวมของโครงงานนี้

จากรูปที่ 3.9 จะมีการออกแบบการทำงานเป็นส่วนใหญ่ ๆ ได้แก่ เครื่องเขียนบัตร (Card Writer) เครื่องแม่ (Master) และเครื่องลูก (Slave)

3.4.1 วิธีการออกแบบกระบวนการทำงานของเครื่องเขียนบัตร (Card Writer)

แผนผังการทำงานของเครื่องเขียนบัตรสามารถแสดงได้ดังรูปที่ 3.10 โดยเครื่องเขียนบัตรนี้จะตรวจสอบการเสียบบัตรและทำหน้าที่เขียนข้อมูลที่ได้รับมาจากคอมพิวเตอร์ลงหน่วยความจำที่อยู่ในบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แผนผังการทำงานของเครื่องเขียนบัตร

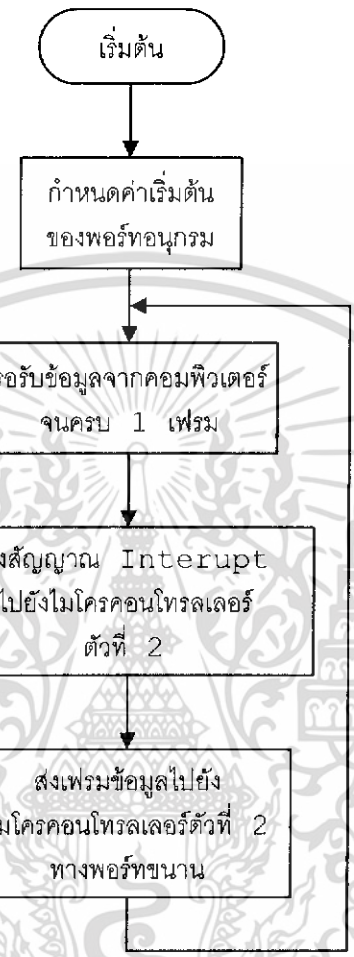
3.4.2 วิธีการออกแบบกระบวนการทำงานของเครื่องแม่ (Master)

กระบวนการทำงานของเครื่องแม่จะใช้ไมโครคอนโทรลเลอร์จำนวน 3 ตัว ในการสื่อสารกับเครื่องคอมพิวเตอร์ โดยไมโครคอนโทรลเลอร์ตัวที่ 1 จะใช้ในการรับข้อมูลจากคอมพิวเตอร์และส่งข้อมูลนั้นให้ไมโครคอนโทรลเลอร์ตัวที่ 2 เพื่อกระจายไปยังเครื่องลูก ส่วนไมโครคอนโทรลเลอร์ตัวที่ 3 จะรับข้อมูลจากไมโครคอนโทรลเลอร์ตัวที่ 2 ที่มาจากเครื่องลูก แล้วส่งไปยังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.1 การทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1

แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1 สามารถแสดงได้ดังรูปที่ 3.11 โดยไมโครคอนโทรลเลอร์จะรอรับข้อมูลจากคอมพิวเตอร์แล้วส่งข้อมูลนั้นต่อไปยังไมโครคอนโทรลเลอร์ตัวที่ 2

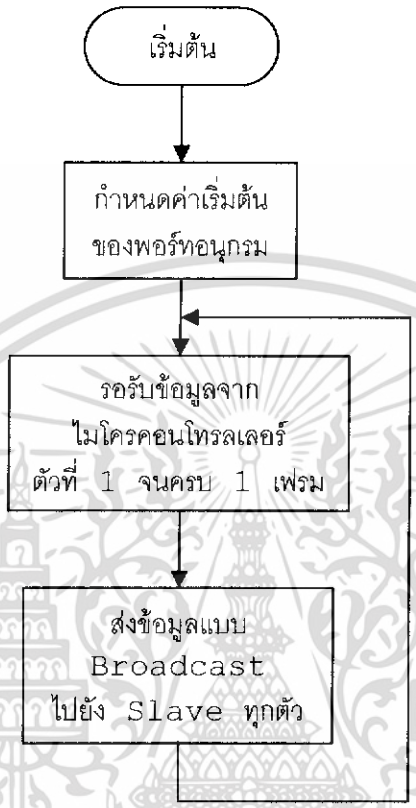


รูปที่ 3.11 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2.2 การทำงานของไมโครคอนโทรลเลอร์ตัวที่ 2

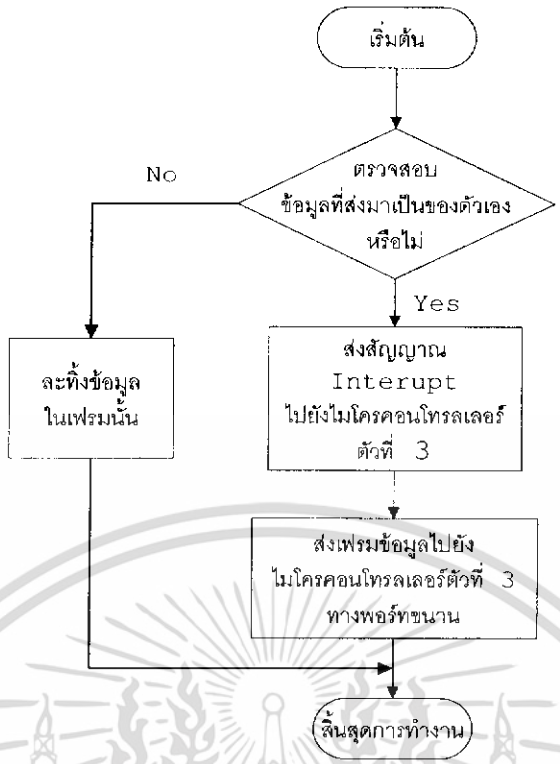
การทำงานในส่วนของโปรแกรมหลักสามารถแสดงได้ ดังรูปที่ 3.12 โดยในสภาวะปกติไมโครคอนโทรลเลอร์จะรอรับข้อมูลจากไมโครคอนโทรลเลอร์ตัวที่ 1 ที่มาจากคอมพิวเตอร์ เพื่อส่งต่อไปยังเครื่องลูก (Slave) แต่ละตัว



รูปที่ 3.12 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 2

การทำงานในส่วนอินเทอร์รัพท์สามารถแสดงได้ ดังรูปที่ 3.13 โดยกระบวนการอินเทอร์รัพท์จะเกิดขึ้น เมื่อเครื่องลูกมีการส่งข้อมูลมายังไมโครคอนโทรลเลอร์ตัวที่ 2 จากนั้นไมโครคอนโทรลเลอร์ตัวที่ 2 ก็จะส่งข้อมูลนี้ต่อไปยังไมโครคอนโทรลเลอร์ตัวที่ 3

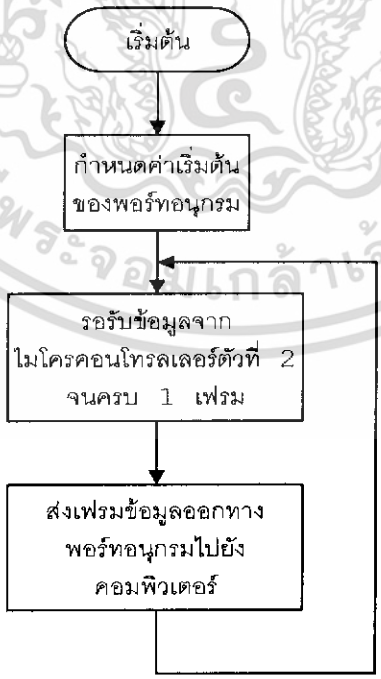
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แผนผังการทำงานในส่วนอินเทอร์รัพท์ของไมโครคอนโทรลเลอร์ตัวที่ 2

3.4.2.3 การทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3

แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3 สามารถแสดงได้ดังรูปที่ 3.14โดยไมโครคอนโทรลเลอร์ตัวที่ 3 จะรอรับข้อมูลจากไมโครคอนโทรลเลอร์ตัวที่ 2 แล้วส่งข้อมูลนั้นต่อไปยังคอมพิวเตอร์

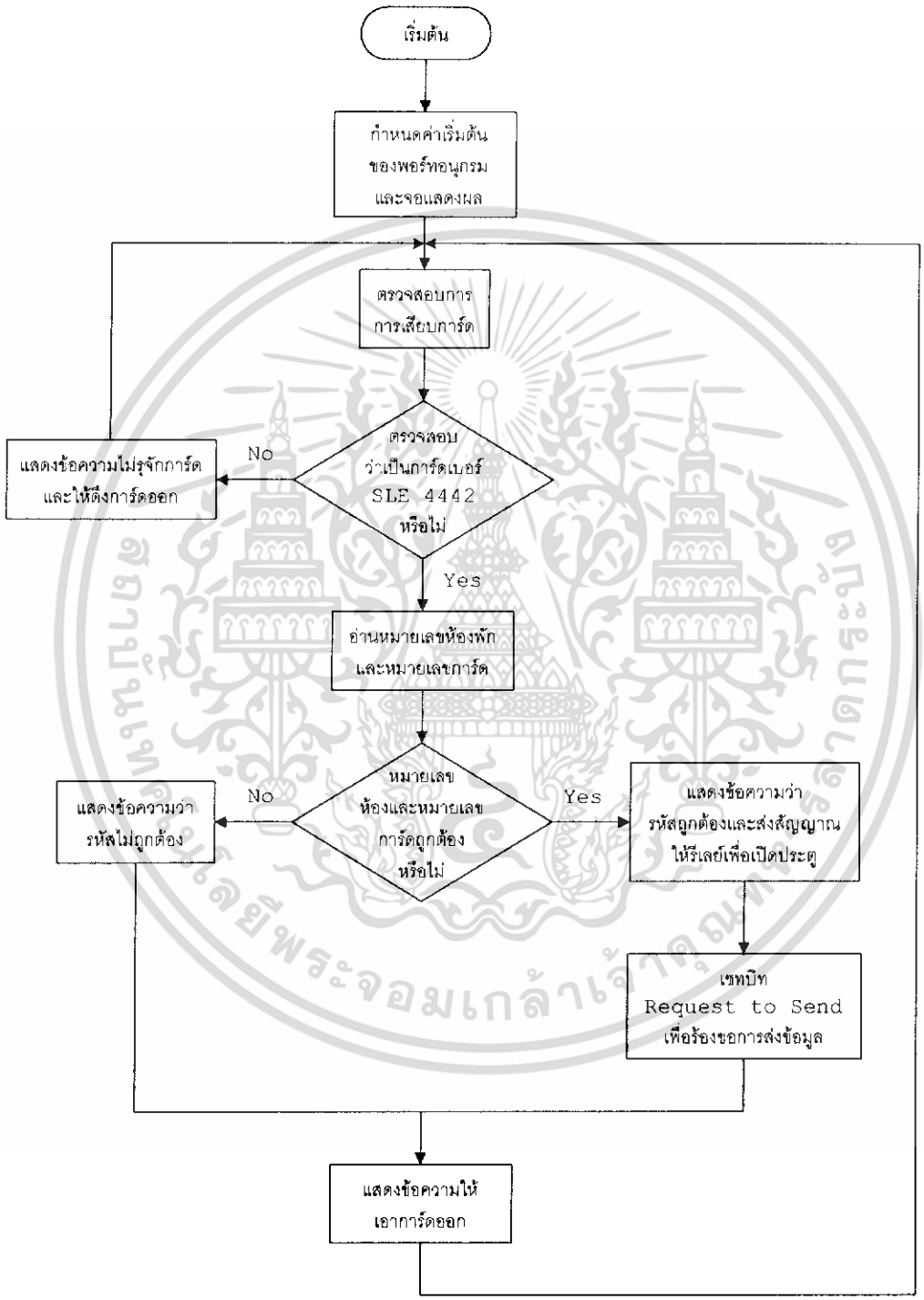


รูปที่ 3.14 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตัวที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 วิธีการออกแบบกระบวนการทำงานของเครื่องลูก (Slave)

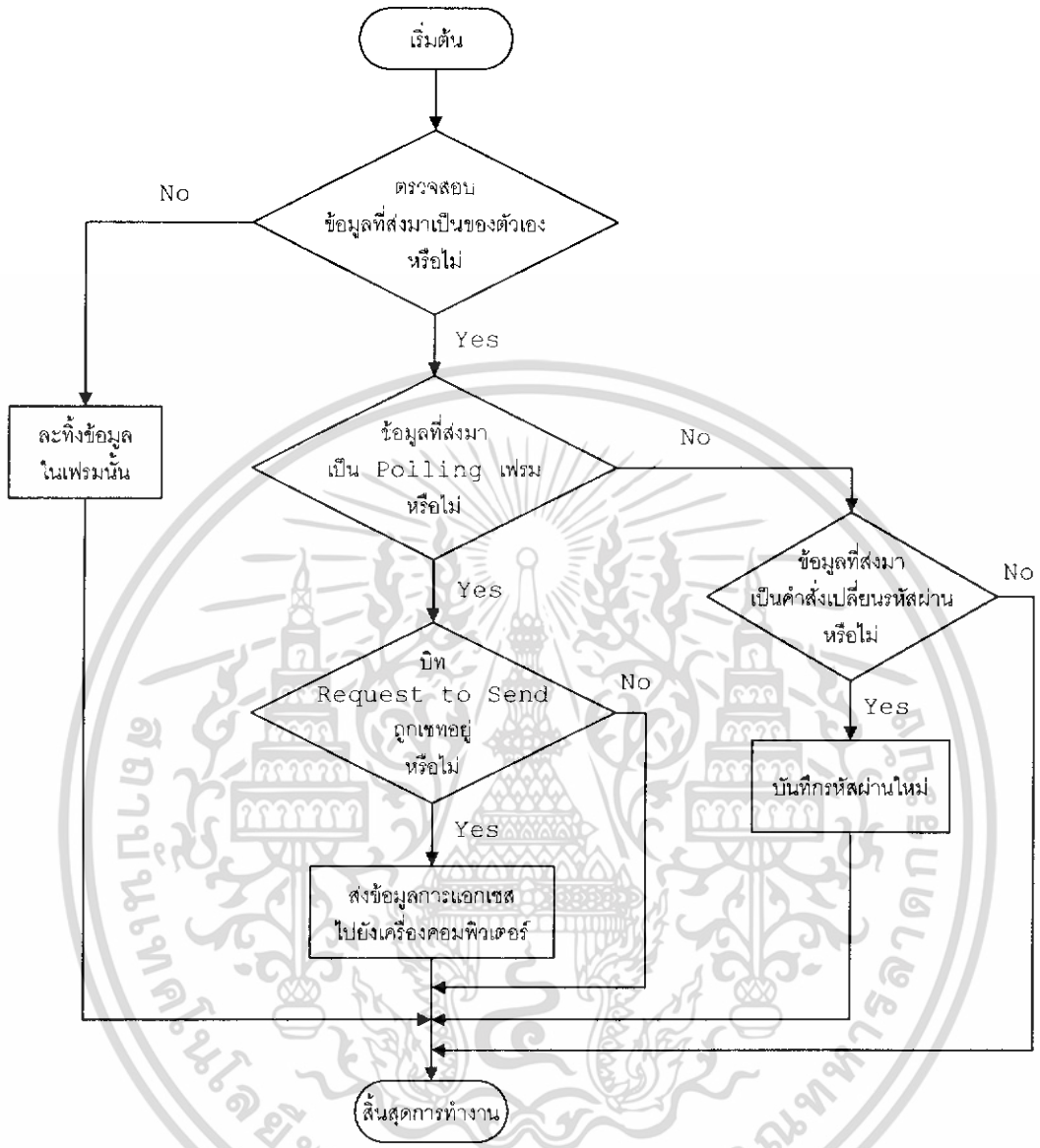
การทำงานในส่วนของโปรแกรมหลักของเครื่องลูกสามารถแสดงได้ ดังรูปที่ 3.15 โดยไมโครคอนโทรลเลอร์จะรอรับการเสียบการ์ด แล้วทำการตรวจสอบข้อมูลในการ์ดนั้น ถ้าหากข้อมูลนั้นถูกต้องก็จะอนุญาตให้ผ่านได้ แต่ถ้าไม่ถูกต้องก็จะแสดงข้อความให้ผู้ใช้ทราบ



รูปที่ 3.15 แผนผังการทำงานของเครื่องลูก(slave)

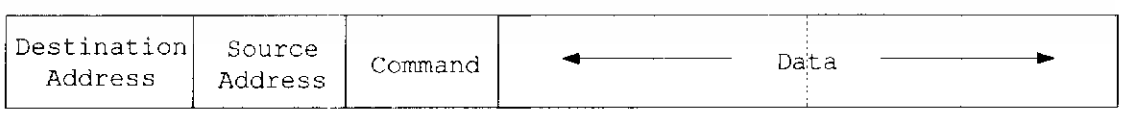
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานในส่วนอินเทอร์รัพท์สามารถแสดงได้ ดังรูปที่ 3.16 ซึ่งกระบวนการนี้จะเกิดขึ้นเมื่อเครื่องลูกได้รับเฟรมข้อมูลจากเครื่องแม่ ซึ่งเฟรมข้อมูลนี้อาจเป็นได้ทั้งคำสั่งหรือการโพล



รูปที่ 3.16 แผนผังการทำงานในส่วนอินเทอร์รัพท์ของเครื่องลูก

3.4.4 โปรโตคอลที่ใช้ในการสื่อสารระหว่างฮาร์ดแวร์



รูปที่ 3.17 โปรโตคอลที่ใช้ในการสื่อสารระหว่างฮาร์ดแวร์

Destination Address : เป็นข้อมูลส่วนที่ระบุถึงแอดเดรสของเครื่องรับ

Source Address : เป็นข้อมูลส่วนที่ระบุถึงแอดเดรสของเครื่องส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

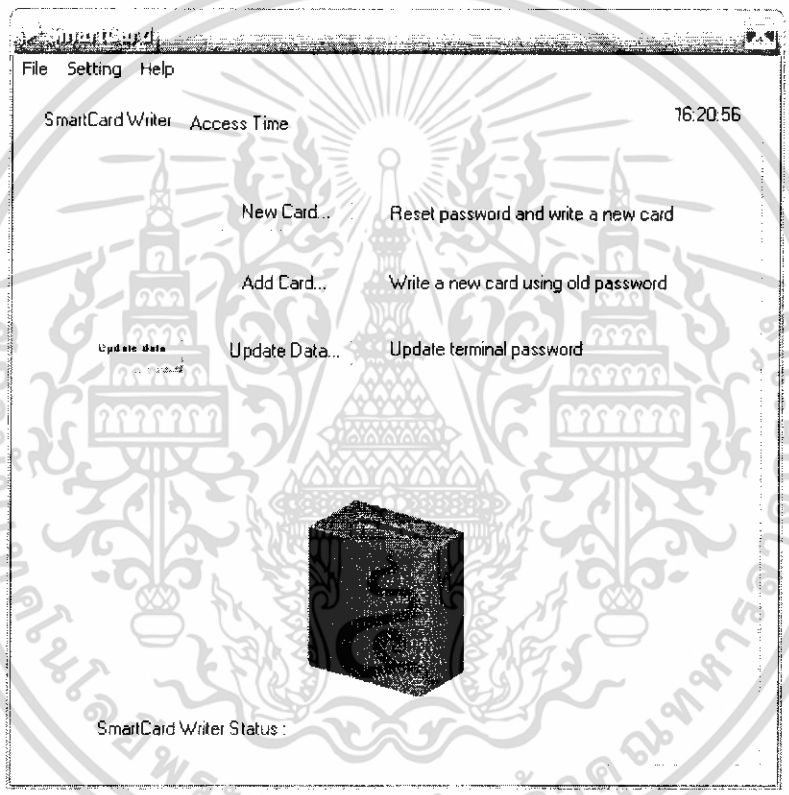
Command : เป็นข้อมูลที่ระบุว่าข้อมูลในเฟรมนี้เป็นคำสั่งหรือเป็นการโพล โดยถ้าเป็นการโพลจะมีค่าเป็น 1 ถ้าเป็นคำสั่งจะมีค่าเป็น 2

Data : ข้อมูลที่ส่ง ถ้าเฟรมนั้นเป็นการโพล ข้อมูลในส่วนนี้จะไม่มีการใช้งาน

3.5 วิธีการออกแบบโปรแกรมคอมพิวเตอร์

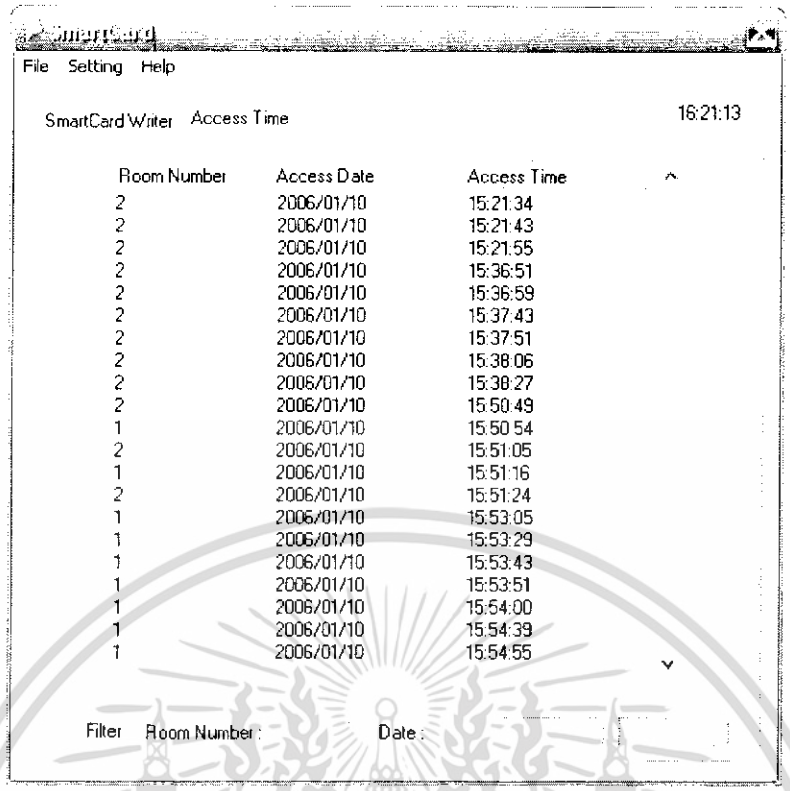
3.5.1 การออกแบบส่วนติดต่อกับผู้ใช้ (Interface)

ในส่วนของโปรแกรมที่ใช้ควบคุมการทำงานของคอมพิวเตอร์ จะประกอบด้วยสองส่วนที่สำคัญ ได้แก่ ส่วนของการควบคุมการทำงานของเครื่องเขียนบัตร และส่วนแสดงผลที่อ่านมาจากฐานข้อมูล โดยลักษณะของหน้าต่างที่ใช้ควบคุมการทำงานของเครื่องอ่านบัตร และส่วนแสดงผลที่อ่านมาจากฐานข้อมูลสามารถแสดงได้ดังรูปที่ 3.18 และรูปที่ 3.19 ตามลำดับ



รูปที่ 3.18 หน้าต่างโปรแกรมที่ใช้ควบคุมการทำงานของเครื่องอ่านบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 หน้าต่างแสดงผลที่อ่านมาจากฐานข้อมูล

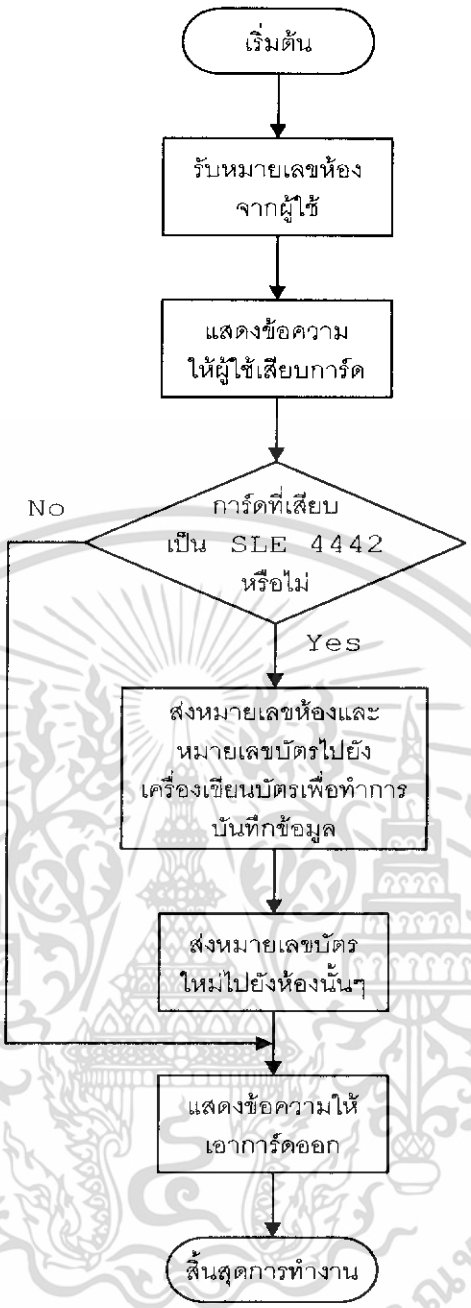
3.5.2 กระบวนการทำงานของโปรแกรม

การทำงานของโปรแกรมคอมพิวเตอร์สามารถแบ่งออกได้เป็นสองส่วนใหญ่ ๆ ได้แก่ ส่วนของการเขียนข้อมูลลงบัตรใบใหม่ผ่านทางเครื่องเขียนบัตร และส่วนของการโพลลิ่ง (Polling)

3.5.2.1 ส่วนการเขียนข้อมูลลงบัตรใหม่

กระบวนการนี้จะใช้ในการเขียนข้อมูล ได้แก่ หมายเลขห้องและหมายเลขบัตรลงในหน่วยความจำที่อยู่บนสมาร์ทการ์ด ซึ่งแผนผังการทำงานสามารถแสดงได้ดังรูปที่ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

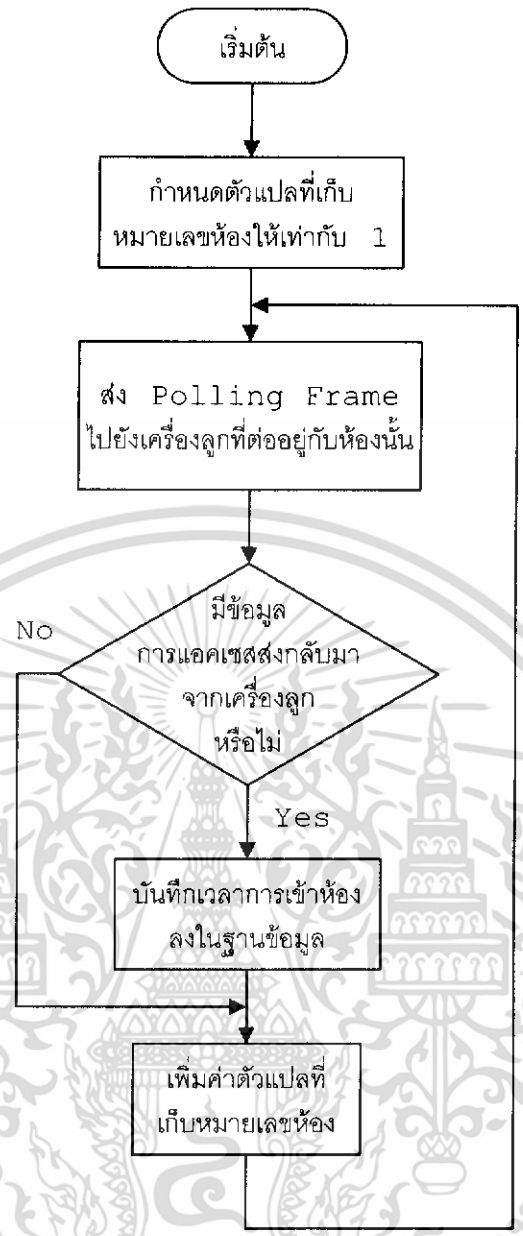


รูปที่ 3.20 แผนผังการทำงานในส่วนของการเขียนข้อมูลลงบัตรใหม่

3.5.2.2 ส่วนการโพลลิ่ง (Polling)

กระบวนการนี้จะใช้ในการโพลไปยังเครื่องลูกว่าเครื่องนั้นมีข้อมูลที่ต้องการส่งหรือไม่ ถ้ามีก็จะนำข้อมูลนั้นมาประมวลผลต่อไป ซึ่งแผนผังการทำงานสามารถแสดงได้ดังรูปที่ 3.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



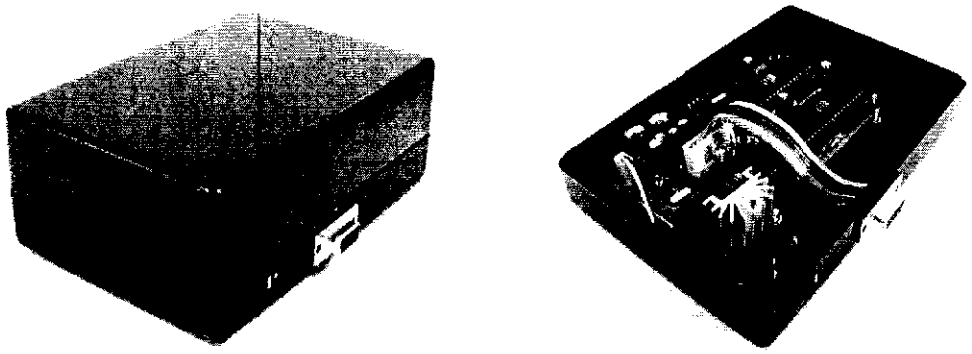
รูปที่ 3.21 แผนผังการทำงานในส่วนของการโพลลิ่ง

3.6 การออกแบบชิ้นงานทั้งหมดในโครงงานนี้

จะมีส่วนประกอบต่างๆดังนี้

- เครื่องแม่ (Master)
- เครื่องเขียนบัตร (Card Writer)
- เครื่องลูก (Slave)
- แบบจำลองประตูห้องพัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

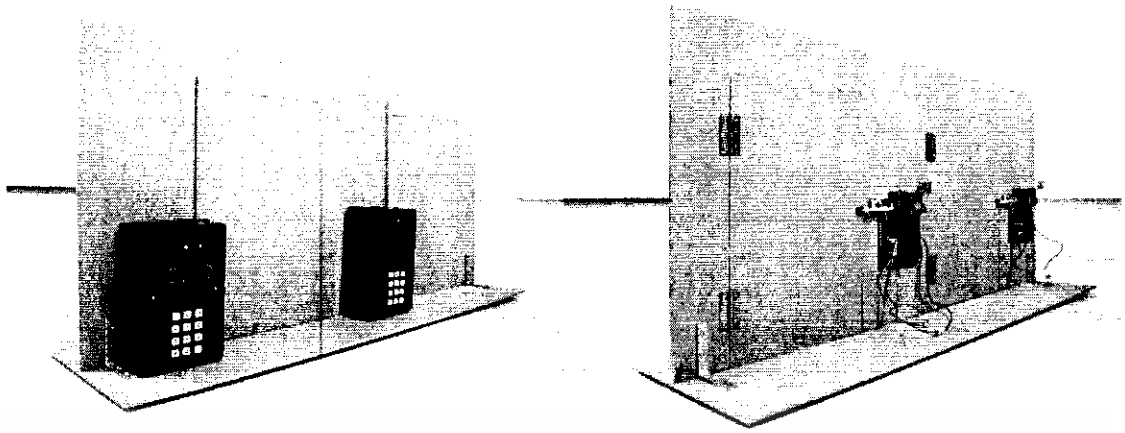


รูปที่ 3.22 เครื่องแม่ (Master)

รูปที่ 3.23 เครื่องเขียนบัตร (Card Writer)

รูปที่ 3.24 เครื่องลูก (Slave)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) หน้าห้อง (ข) ในห้อง

รูปที่ 3.25 แบบจำลองประตูห้องพัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

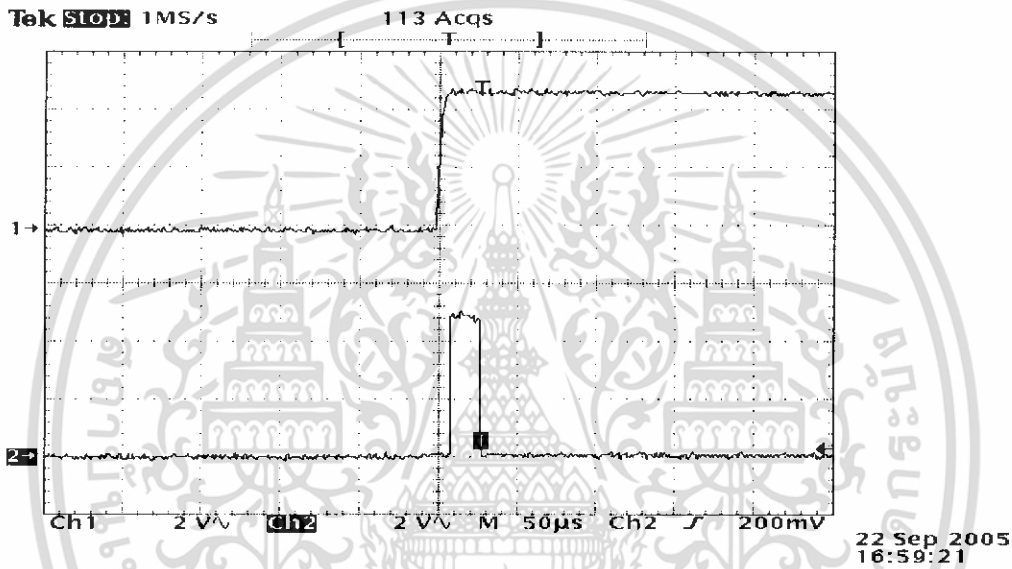
บทที่ 4

การทดลองและผลการทดลอง

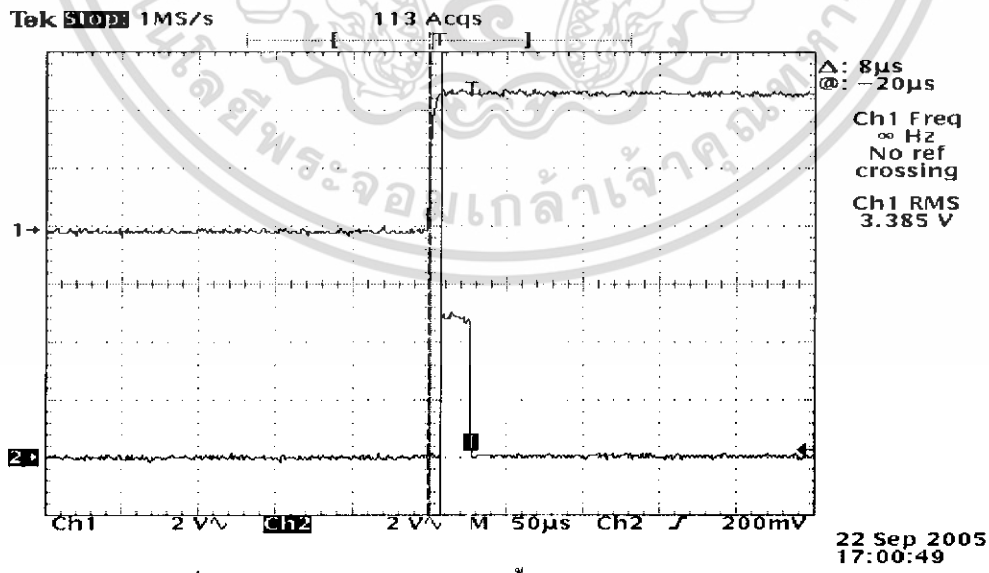
หลังจากได้ทำการออกแบบทั้งในส่วนของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ดและในส่วนของโปรแกรมในบทที่ 3 มาแล้วนั้น จึงได้ทำการทดลองและเก็บผลการทดลองโดยแบ่งออกเป็นส่วน ต่าง ๆ ได้แก่ การทดลองตรวจสอบช่วงเวลาของสัญญาณที่ใช้ติดต่อกับชิพสมาร์ทการ์ด การทดลองการสื่อสารผ่านพอร์ทอนุกรม และการทดลองกระบวนการทำงานต่าง ๆ ทั้งหมดของโครงงานนี้

4.1 การทดลองตรวจสอบช่วงเวลาของสัญญาณต่าง ๆ ที่ใช้ในการติดต่อกับชิพสมาร์ทการ์ด

4.1.1 สัญญาณ Vcc และ Reset



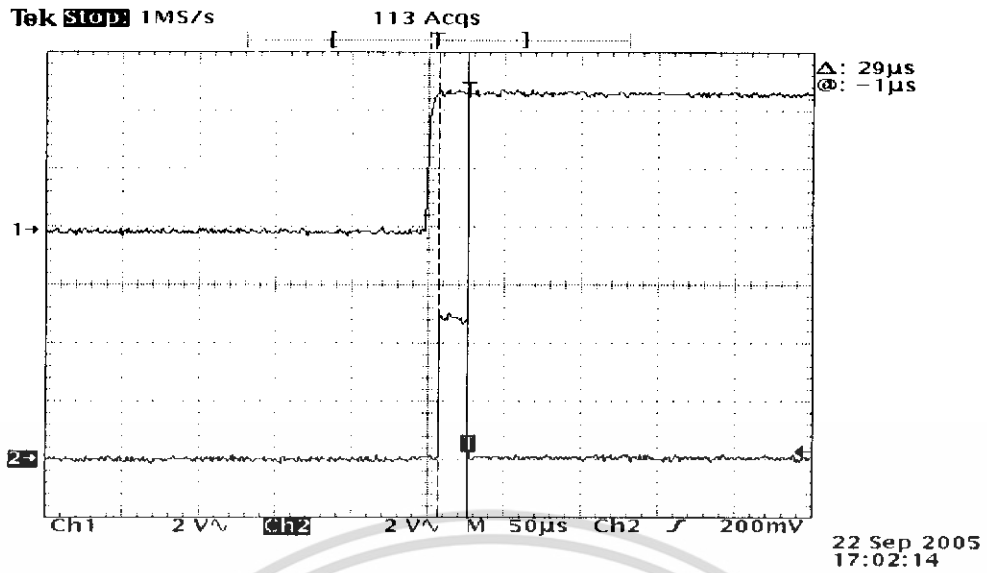
รูปที่ 4.1 Channel 1: สัญญาณ Vcc, Channel 2 : สัญญาณ Reset



รูปที่ 4.2 ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ Vcc และ Reset

Channel 1 : สัญญาณ Vcc, Channel 2 : สัญญาณ Reset

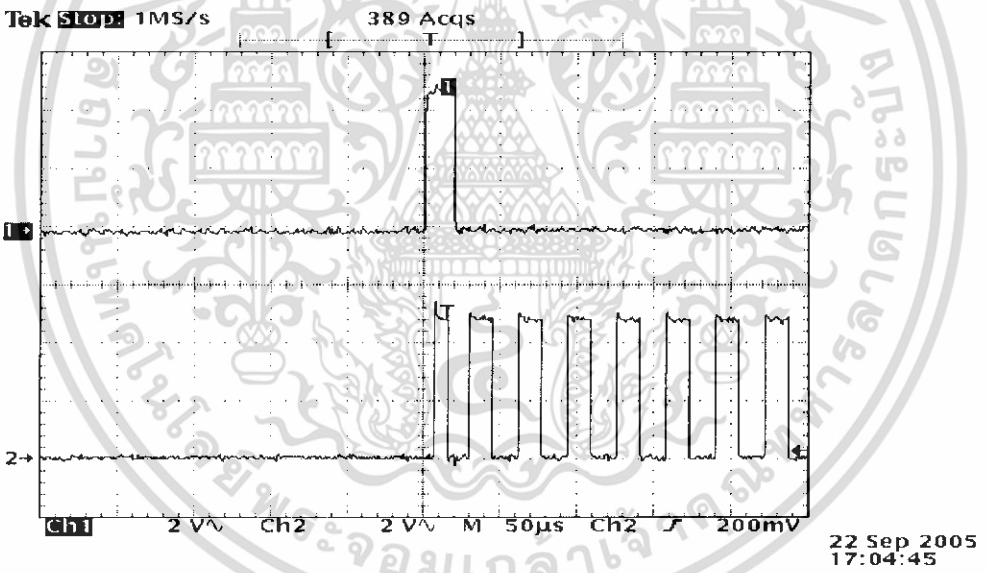
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 ช่วงเวลาขณะที่สัญญาณ Reset อยู่ในสถานะ High

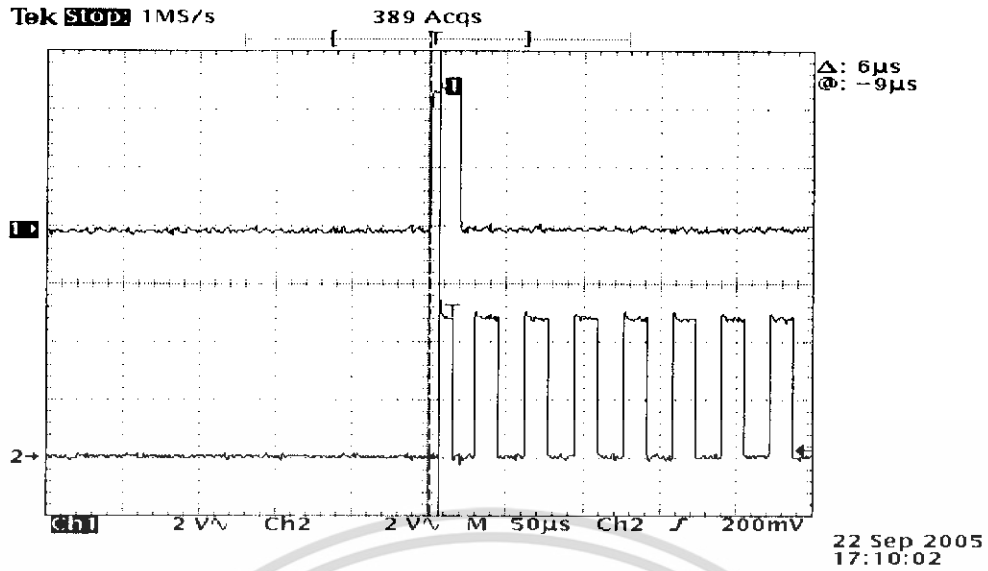
Channel 1 : สัญญาณ Vcc, Channel 2 : สัญญาณ Reset

4.1.2 สัญญาณ Reset และสัญญาณนาฬิกา



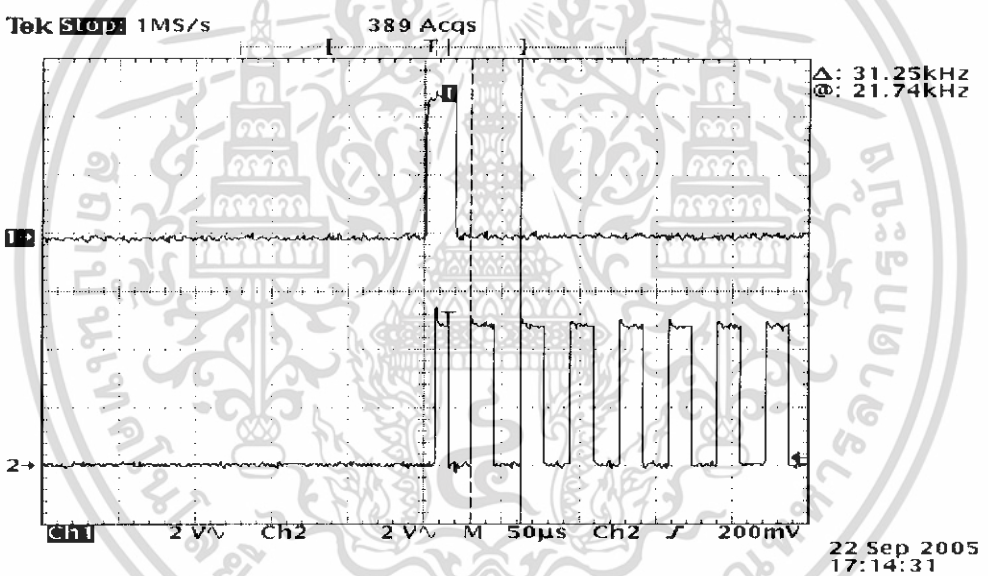
รูปที่ 4.4 Channel 1 : สัญญาณ Reset, Channel 2 : สัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ Reset และ สัญญาณนาฬิกา

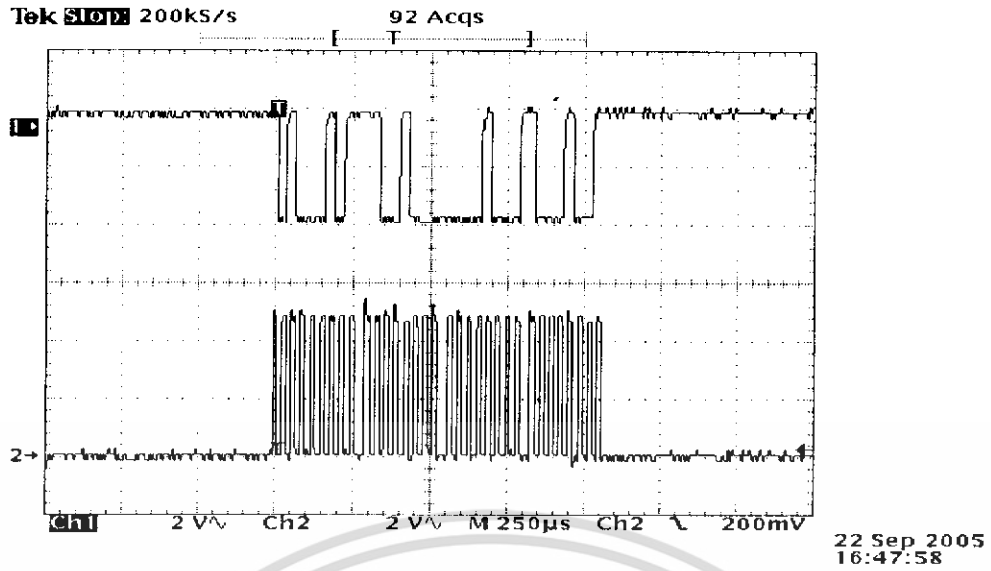
Channel 1 : สัญญาณ Reset, Channel 2 : สัญญาณนาฬิกา



รูปที่ 4.6 ความถี่ของสัญญาณนาฬิกา

Channel 1 : สัญญาณ Reset, Channel 2 : สัญญาณนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

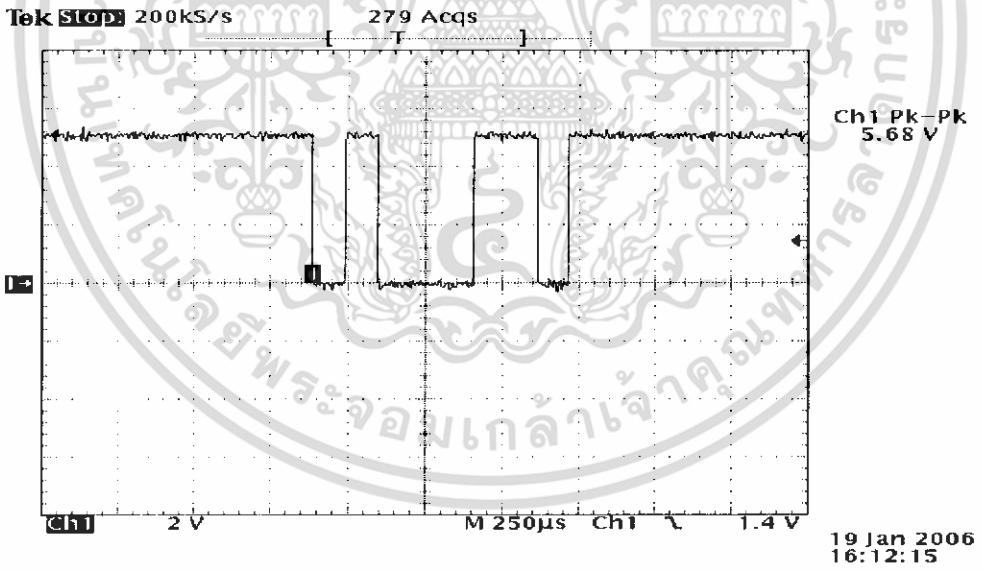


รูปที่ 4.7 ข้อมูลที่อ่านออกมาในกระบวนการ Answer-to-Reset

Channel 1 : สัญญาณข้อมูล, Channel 2 : สัญญาณนาฬิกา

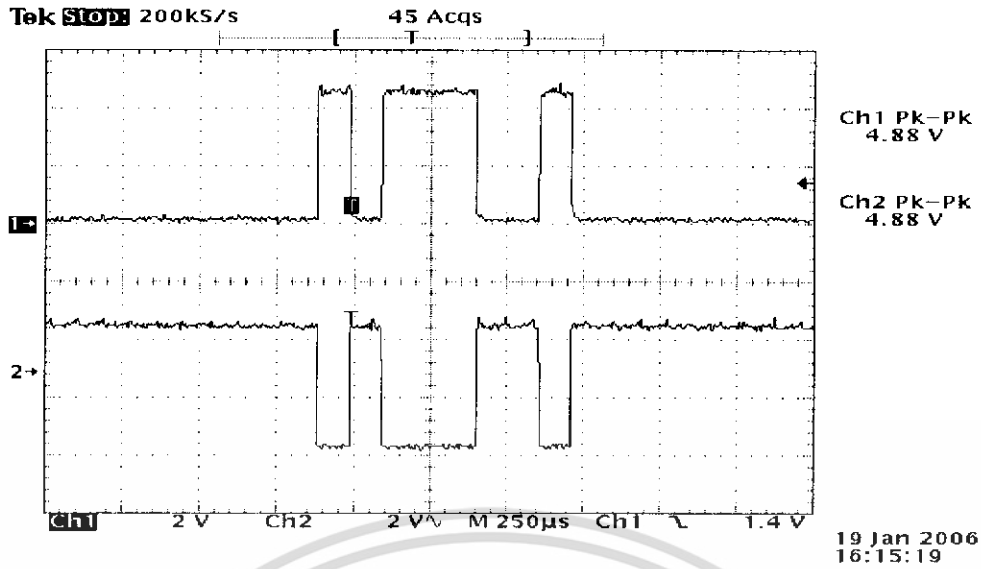
จากผลการทดลองที่ได้พบว่า ช่วงเวลาของสัญญาณต่าง ๆ ที่ใช้ในการติดต่อสื่อสารกับสมาร์ทการ์ด เป็นไปตามข้อกำหนดของสมาร์ทการ์ดเบอร์ SLE4442 ดังตารางที่ 2.11 จากบทที่ 2

4.1.3 สัญญาณ RS-485



รูปที่ 4.8 สัญญาณอินพุทจากไมโครคอนโทรลเลอร์

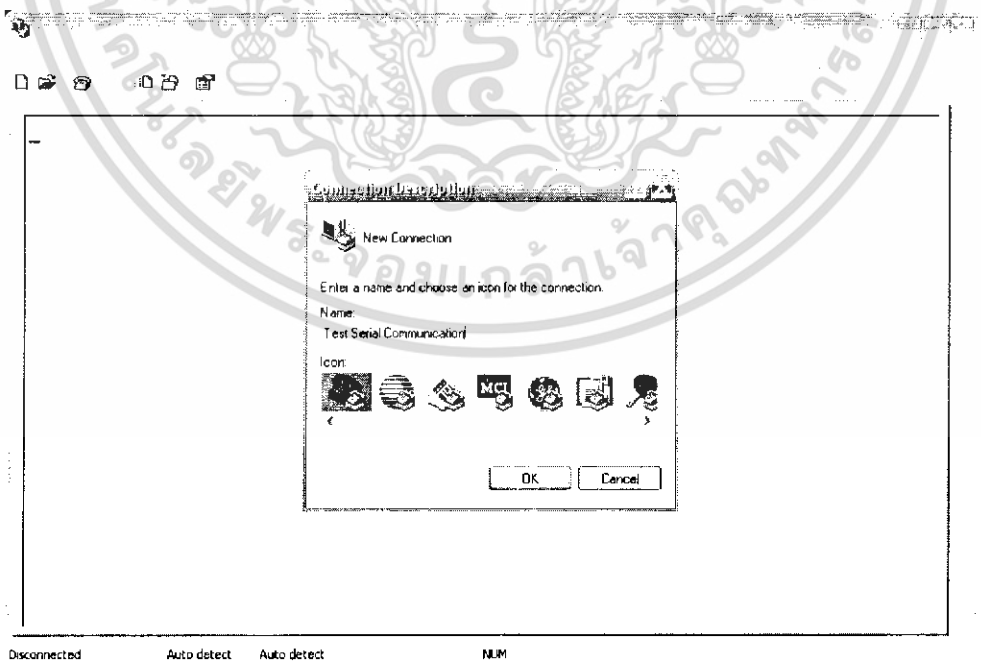
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 สัญญาณเอาต์พุทจาก RS-485

4.2 การทดลองการสื่อสารผ่านพอร์ทอนุกรม RS-232 (Serial Port)

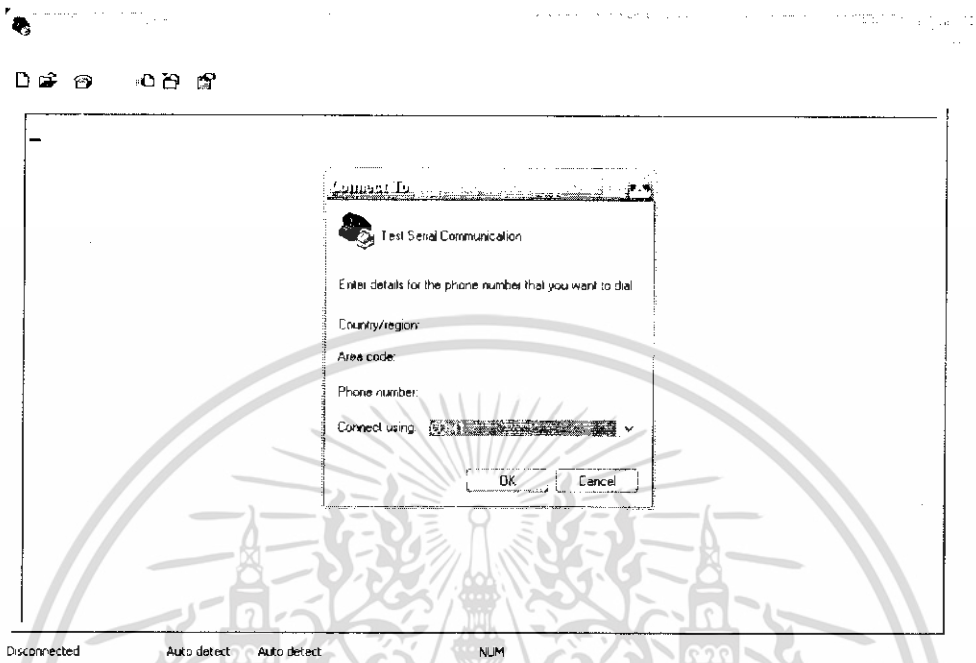
การทดลองการทำงานในส่วนของการสื่อสารผ่านพอร์ทอนุกรมนั้นจะทำการต่อวงจรตามรูปที่ 3.5 และทำการเขียนโปรแกรมเพื่อรับค่าข้อมูลทางคีย์บอร์ดตามแผนผังการทำงานในรูปที่ 3.6 แล้วทำการป้อนไฟเลี้ยงขนาด +5 โวลต์ให้กับวงจร จากนั้นทำการทดลองการใช้งานโดยใช้โปรแกรมไฮเปอร์เทอร์มินอล (Hyper Terminal) โดยเมื่อเปิดโปรแกรมไฮเปอร์เทอร์มินอลขึ้นมาจะทำให้ปรากฏหน้าต่างดังรูปที่ 4.10 โดยกำหนดให้มีการเชื่อมต่อกับโปรแกรมที่จะใช้ทดลองโปรแกรม ในการทดลองนี้ได้ทำการตั้งชื่อโปรแกรมว่า " Test Serial Communication "



รูปที่ 4.10 หน้าต่างโปรแกรมไฮเปอร์เทอร์มินอลเมื่อเปิดขึ้นมาใช้งานครั้งแรก

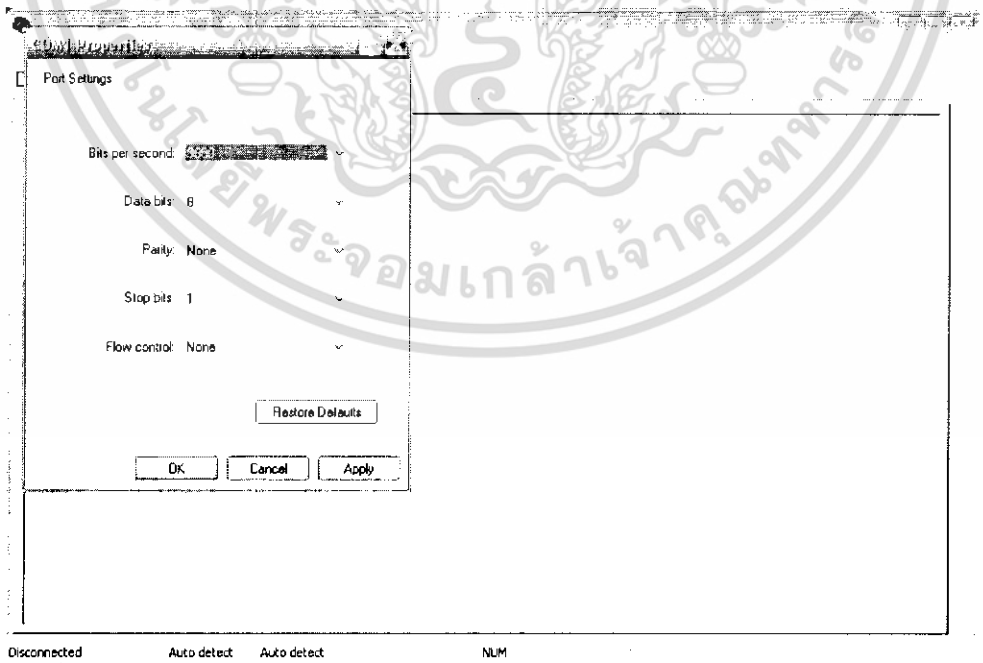
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการกำหนดโหมดการทำงานให้กับโปรแกรมไฮเปอร์เทอร์มินอล โดยในการทดลองนี้จะเลือก "Com1 " ในการเลือกโหมดการทำงานนั้นโปรแกรมไฮเปอร์เทอร์มินอลจะปรากฏหน้าต่างดังรูปที่ 4.11



รูปที่ 4.11 หน้าต่างการกำหนดโหมดการทำงานให้ไฮเปอร์เทอร์มินอล

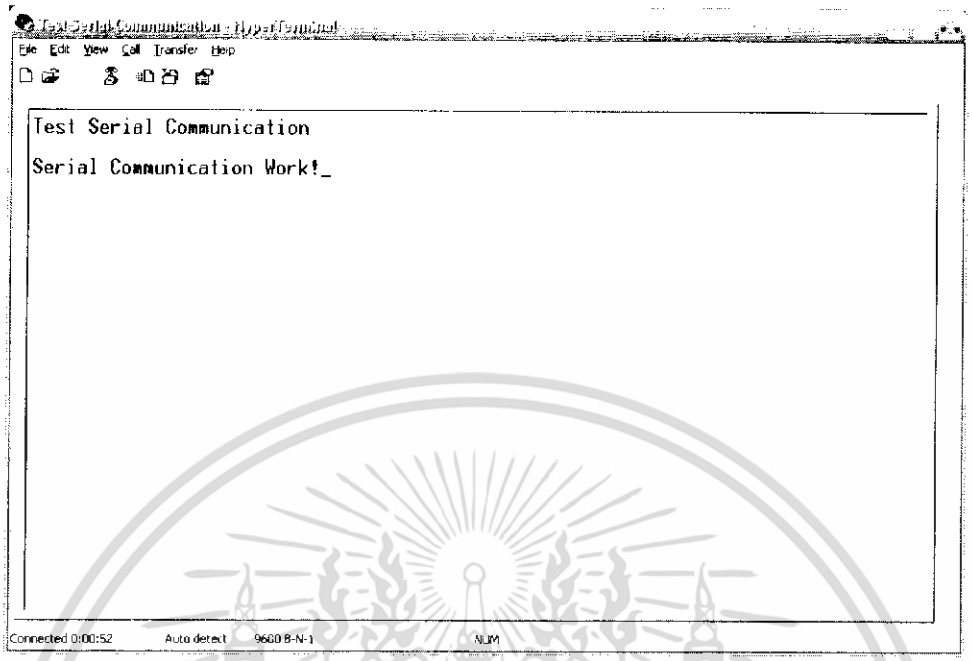
จากนั้นทำการกำหนดค่าต่าง ๆ ที่ใช้ในการติดต่อสื่อสาร ซึ่งในการกำหนดค่านั้นโปรแกรมไฮเปอร์เทอร์มินอลจะปรากฏหน้าต่างดังรูปที่ 4.12



รูปที่ 4.12 หน้าต่างการกำหนดค่าต่าง ๆ ที่ใช้ในการติดต่อสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสั่งให้โปรแกรมทำงาน ไมโครคอนโทรลเลอร์ก็จะส่งข้อความมายังคอมพิวเตอร์ผ่านทางพอร์ทอนุกรม เพื่อแสดงว่าการสื่อสารผ่านพอร์ทอนุกรมสามารถใช้งานได้ ดังรูปที่ 4.13



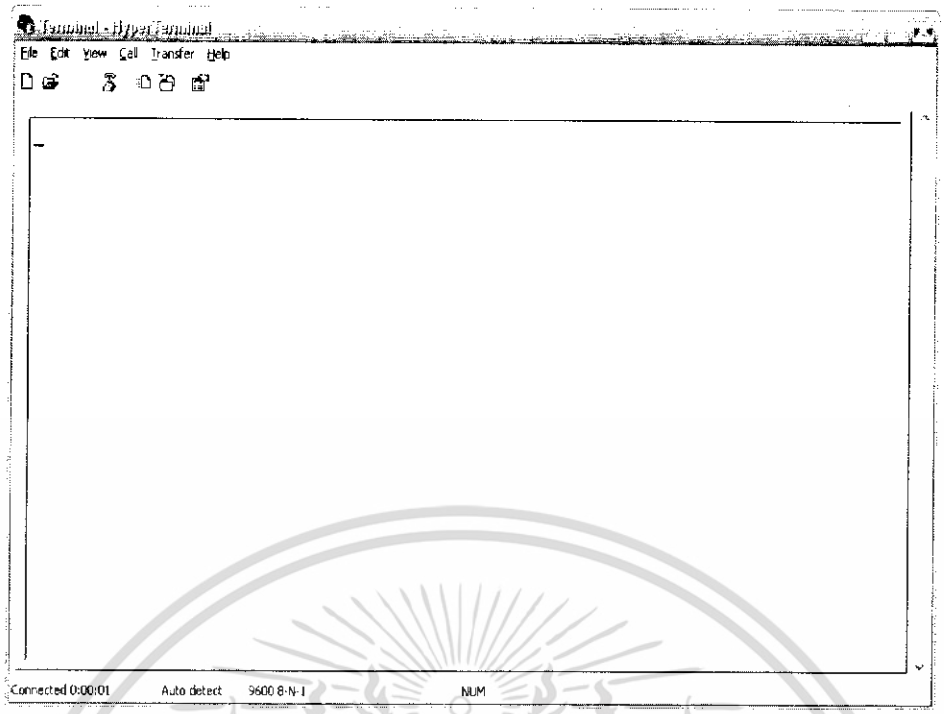
รูปที่ 4.13 ผลการทดลองการสื่อสารข้อมูลผ่านทางพอร์ทสื่อสารอนุกรม RS-232

4.3 การทดลองกระบวนการอ่านและเขียนข้อมูลลงในสมาร์ทการ์ดโดยใช้โปรแกรม Hyper Terminal

4.3.1 การทดลองในส่วนของกระบวนการ เขียนข้อมูลลงในสมาร์ทการ์ด

การทดลองกระบวนการเขียนข้อมูลลงบนบัตรนั้น จะทำการทดลองโดยการเขียนโปรแกรมควบคุมให้เครื่องสามารถเขียนข้อมูลลงบนบัตรสมาร์ทการ์ดเบอร์ SLE4442 โดยจะมีขั้นตอนการออกแบบการทำงานของไมโครคอลโทรลเลอร์ดังที่ได้กล่าวมาแล้วในบทที่ 3 เมื่อเขียนโปรแกรมแล้วจะใช้โปรแกรมไฮเปอร์เทอร์มินอลในการทดสอบโดยใช้ความเร็วในการติดต่อ 9600 bps โดยเมื่อเริ่มต้นให้เครื่องทำงาน บนหน้าจอคอมพิวเตอร์จะปรากฏหน้าต่างดังรูปที่ 4.14 และจากการออกแบบจะให้โปรแกรมเริ่มทำงานจากขั้นตอนการรอรับบัตร โดยจะได้ผลการทดลองเป็นข้อความที่ปรากฏขึ้นบนหน้าจอดังรูปที่ 4.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

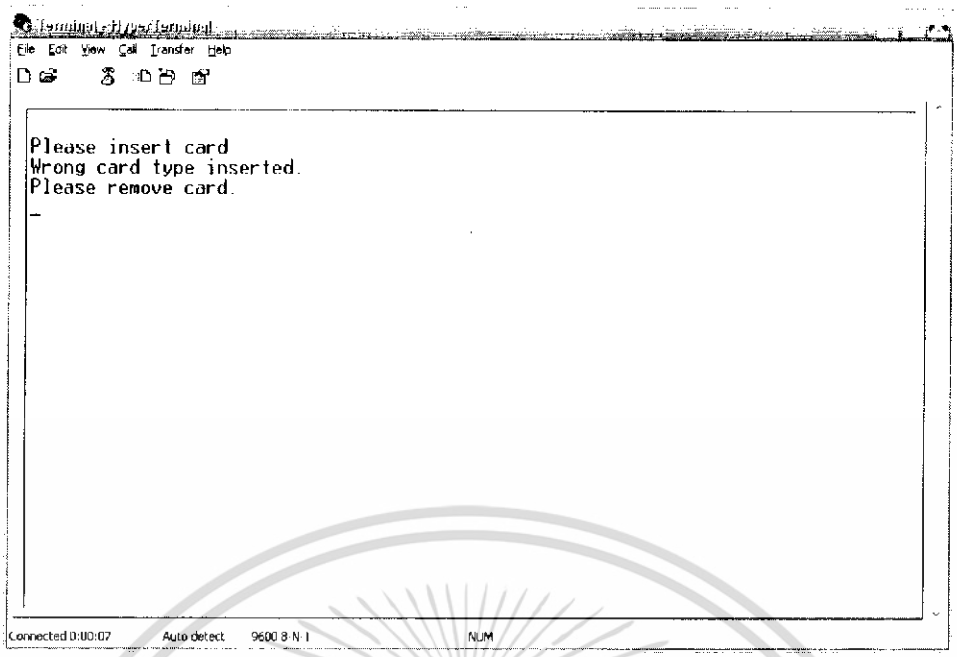


รูปที่ 4.14 หน้าต่างโปรแกรม Hyper Terminal

รูปที่ 4.15 ผลการทดลองของหน้าจอโปรแกรม เพื่อเตรียมรอรับบัตรสมาร์ทการ์ด

เมื่อมีการใส่บัตรเข้าไปในซ็อกเกต (Socket) แล้วโปรแกรมที่ออกแบบไว้นั้นจะทำการตรวจสอบชนิดหรือเบอร์ของบัตรที่ใส่เข้ามาว่าเป็นเบอร์ SLE4442 หรือไม่เพราะจากการออกแบบนั้นทางผู้จัดทำได้ทำการออกแบบโปรแกรมให้สามารถทำงานได้กับบัตรสมาร์ทการ์ดเบอร์ SLE4442 ซึ่งเป็นบัตรแบบมีระบบป้องกันความปลอดภัยของข้อมูล ถ้าเป็นบัตรที่ใส่เข้ามาไม่ใช่เบอร์ SLE4442 โปรแกรมจะบอกให้เอาบัตรนั้นออก ซึ่งจะได้ผลการทดลองปรากฏเป็นข้อความขึ้นมาดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

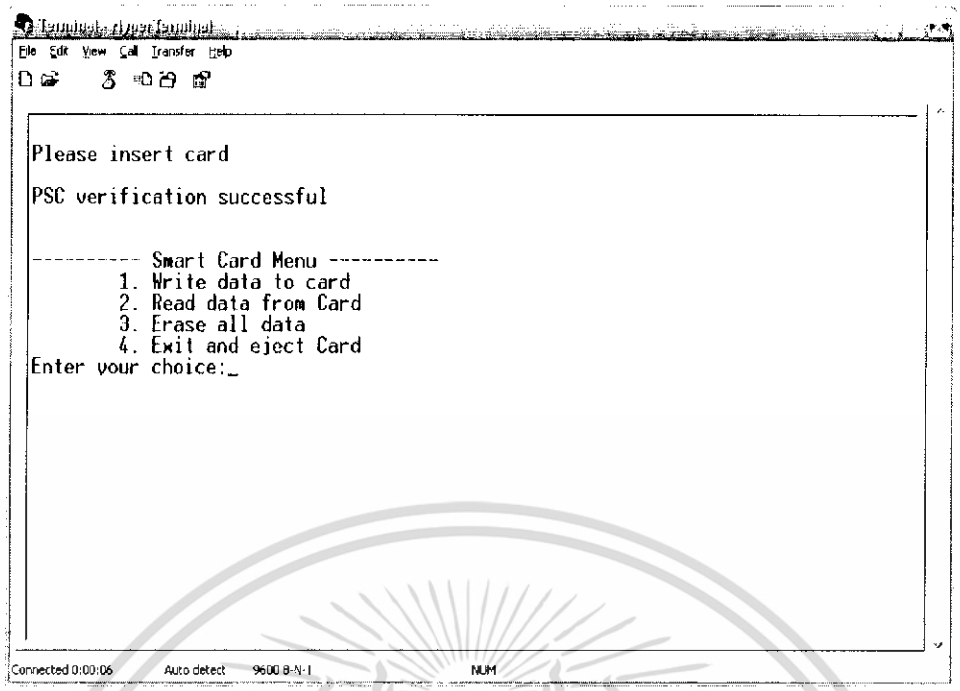


รูปที่ 4.16 ผลการทดลองที่เกิดความผิดพลาดขึ้นเมื่อบัตรสมาร์ทการ์ดที่เสียบ
ไม่ใช่เบอร์ SLE4442 หรือเสียบบัตรไม่ถูกต้อง

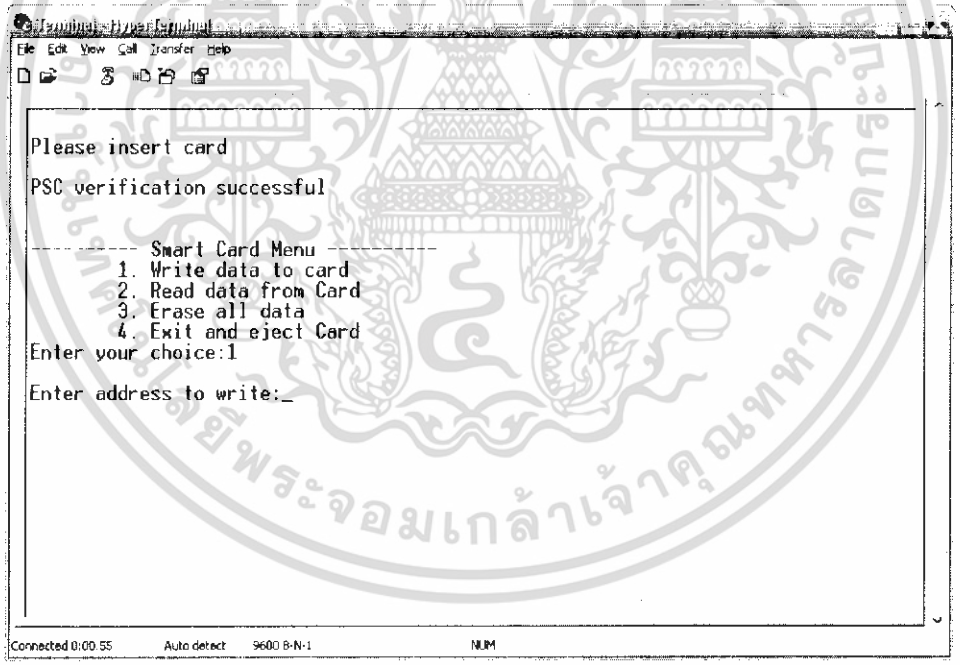
จากนั้นทำการทดลองโดยเปลี่ยนเป็นบัตรสมาร์ทการ์ดเบอร์ SLE4442 ซึ่งเป็นเบอร์ที่สามารถทำงานกับโปรแกรมที่ออกแบบไว้ได้ จะทำให้ปรากฏข้อความขึ้นมาเป็นรายการให้เลือกทำงาน ในที่นี้ทางผู้จัดทำได้ทำการออกแบบให้ สามารถเลือกกระบวนการทำงานได้ 4 กระบวนการ คือ กระบวนการอ่านข้อมูลลงบนบัตร, กระบวนการเขียนข้อมูลลงบนบัตร, กระบวนการลบข้อมูลทั้งหมดในบัตร และกระบวนการยกเลิกการทำงานของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ดดังรูปที่ 4.17

เพื่อเป็นการทดลองใช้งานของกระบวนการต่างๆ ตามที่ทางผู้จัดทำได้กำหนดไว้นั้นในการเลือกกระบวนการทำงาน โดยครั้งแรกจะทดลองโดยการเลือกกระบวนการทำงานในข้อที่ 1 คือ การเขียนข้อมูลลงบนบัตรสมาร์ทการ์ด แล้วหน้าจอจะปรากฏข้อความขึ้นมา ดังผลการทดลองในรูปที่ 4.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

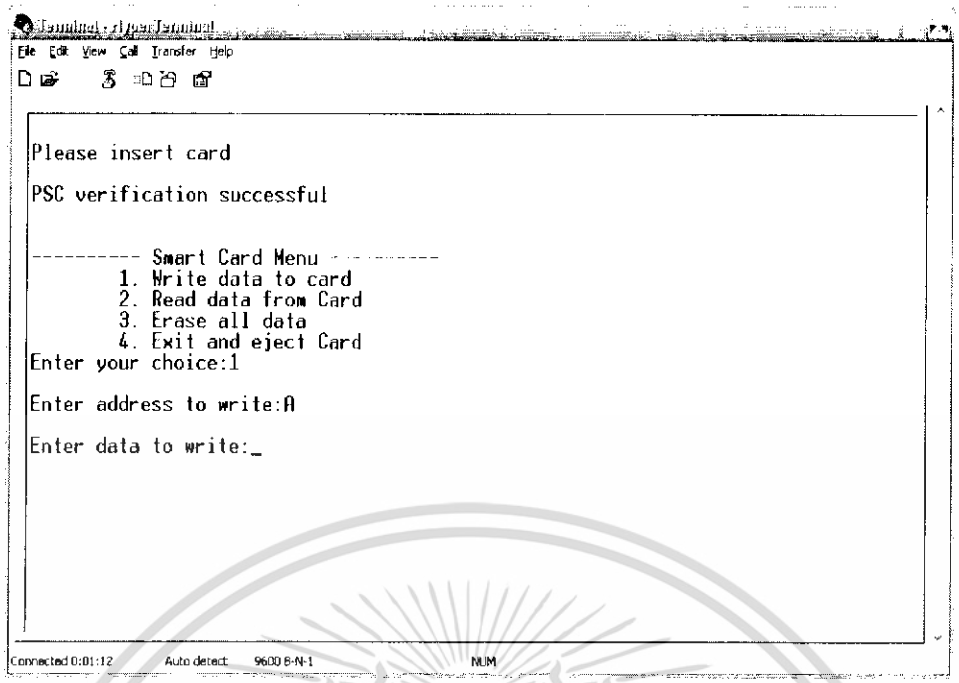


รูปที่ 4.17 ผลการทดลองเมื่อมีการเสียบบัตรสมาร์ทการ์ดเบอร์ SLE4442 เข้ามาจะมีเมนูให้เลือกทำรายการ



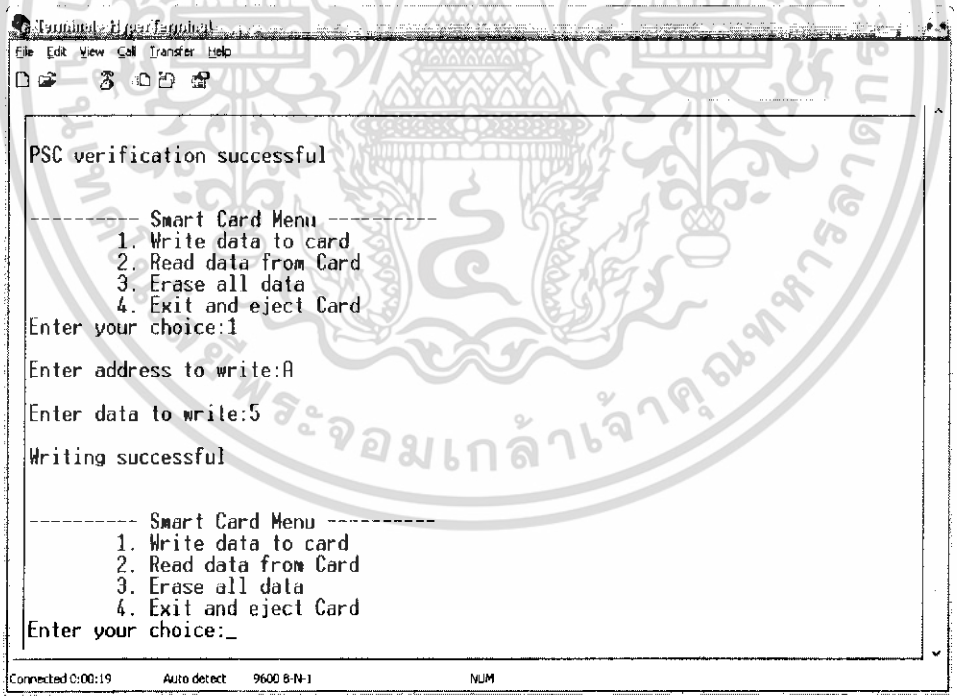
รูปที่ 4.18 เลือกทำรายการที่ 1 เพื่อทำการเขียนข้อมูลลงในบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 ใส่ตำแหน่ง address ที่ต้องการจะเก็บข้อมูล

จากรูปที่ 4.19 เมื่อทำการใส่ตำแหน่งที่จะเก็บข้อมูลแล้ว จึงเขียนข้อมูลลงในบัตรจะได้ผลดังรูปที่ 4.20

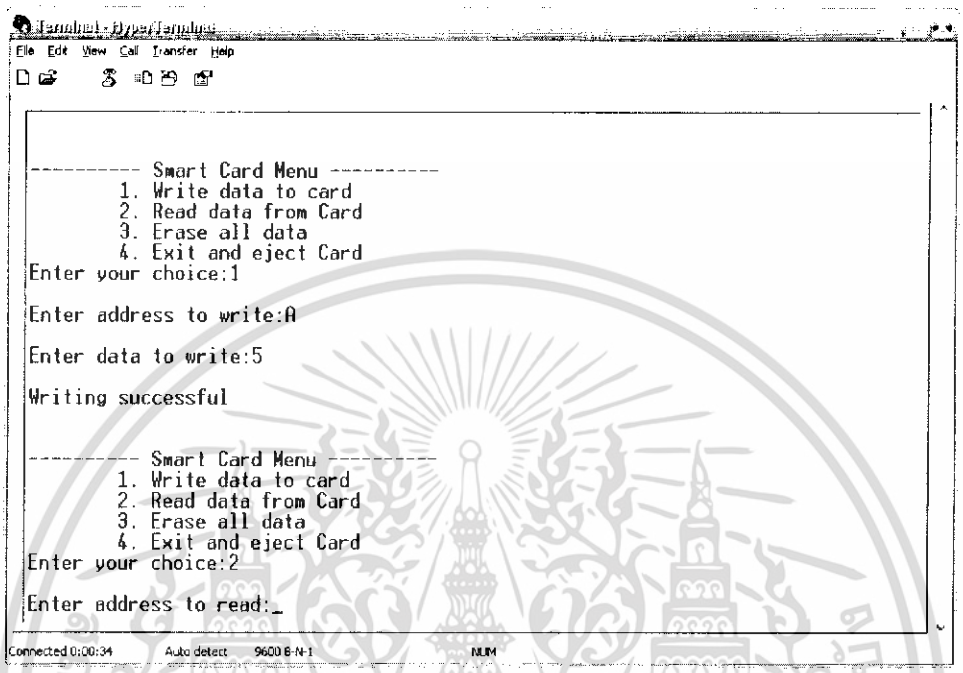


รูปที่ 4.20 ผลการทดลองเมื่อเขียนข้อมูลลงในบัตร

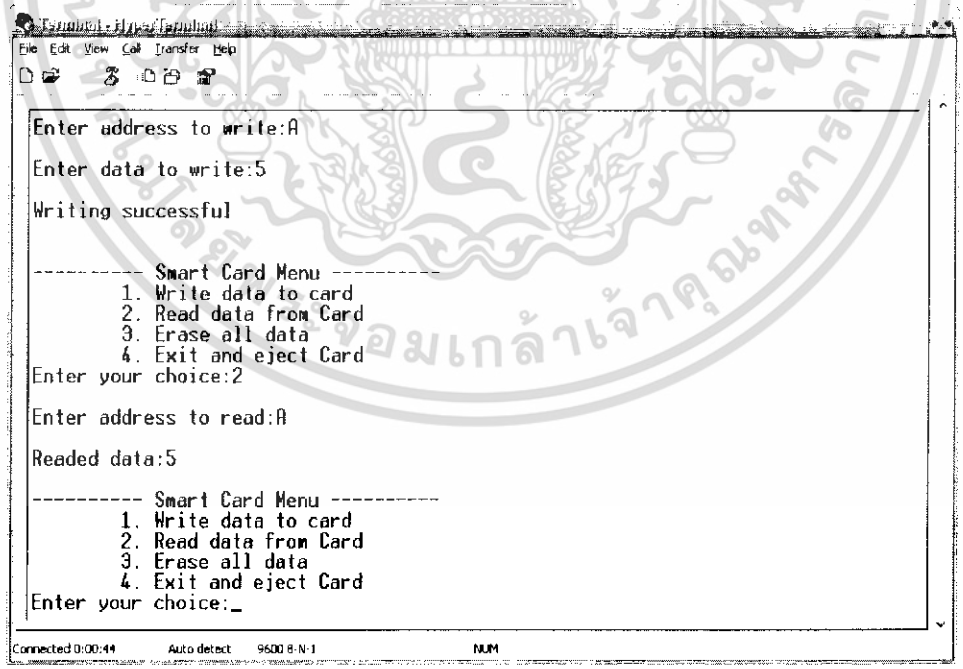
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 การทดลองในส่วนของกระบวนการอ่านข้อมูลภายในสมาร์ทการ์ด

จากนั้นทำการเลือกกระบวนการทำงานในการทดลองกระบวนการอ่านข้อมูลภายในบัตรนี้ จะเลือกรายการที่ 2 ดังรูปที่ 4.21 หลังจากการเลือกให้เครื่องมีกระบวนการในการอ่านข้อมูลภายในบัตรแล้วจะได้ ข้อมูลแสดงเป็นข้อความที่ทางผู้จัดทำได้ทำการเขียนไว้ออกมา ดังผลการทดลองในรูปที่ 4.22



รูปที่ 4.21 เลือกทำรายการที่ 2 อ่านข้อมูลจากบัตรโดยเลือกตำแหน่ง address ที่ต้องการอ่าน

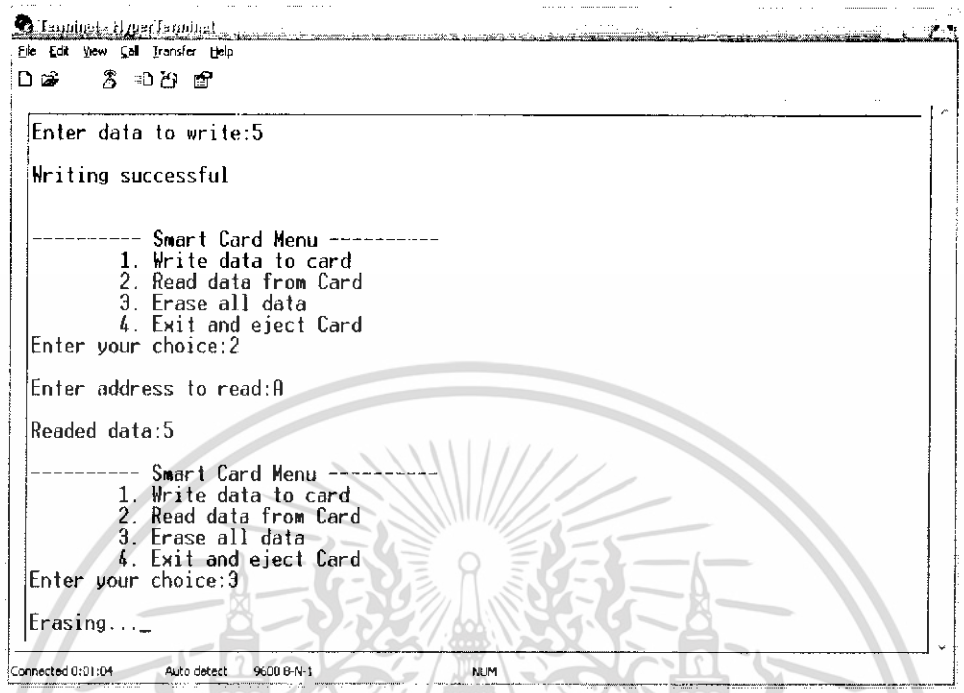


รูปที่ 4.22 แสดงการอ่านข้อมูลจากบัตร

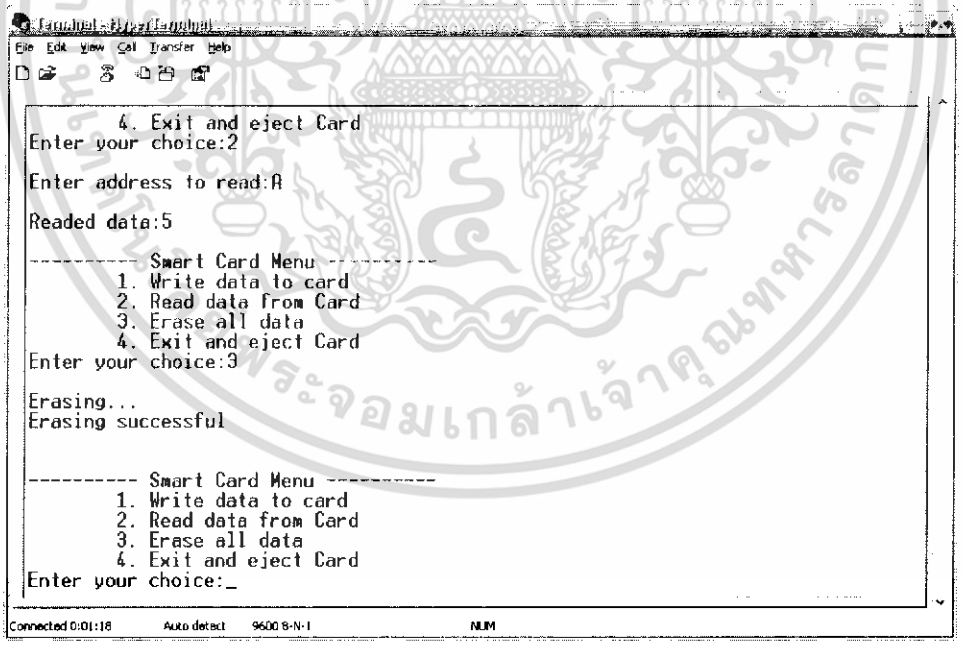
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 การทดลองในส่วนของการลบข้อมูลทั้งหมดในสมาร์ทการ์ด

เมื่อเลือกรายการที่ 3 จะเป็นการลบข้อมูลที่อยู่ภายในบัตรออกจนหมดจะได้ผลดังรูปที่ 4.23



รูปที่ 4.23 เลือกทำรายการที่ 3 ลบข้อมูลทั้งหมดที่อยู่ในบัตร



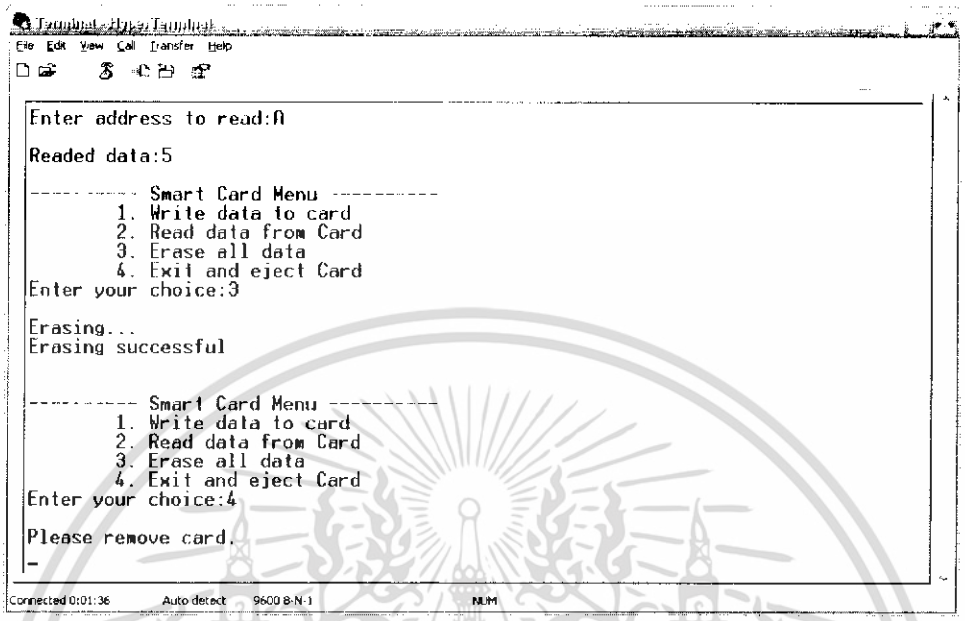
รูปที่ 4.24 แสดงผลการทดลองการลบข้อมูลทั้งหมด

4.3.4 การทดลองในส่วนของกระบวนการสิ้นสุดการทำงานของเครื่องอ่านและเขียนสมาร์ทการ์ด

หลังจากที่เราเลือกให้เครื่องอ่านและเขียนบัตรสมาร์ทการ์ดทำงานในกระบวนการอ่านข้อมูลภายในบัตรหรือกระบวนการเขียนข้อมูลลงบนบัตรเสร็จเรียบร้อยแล้ว หน้าจอจะแสดงรายการเพื่อให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกกระบวนการขึ้นมา และการทดลองในส่วนของกระบวนการสิ้นสุดการทำงานของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ดนั้นเมื่อทำการเลือกหมายเลข 4 แล้วหน้าจอจะปรากฏข้อความเพื่อให้เอาการ์ดออก ซึ่งเป็นการบอกว่าได้สิ้นสุดการทำงานแล้ว ดังผลการทดลองในรูปที่ 4.25



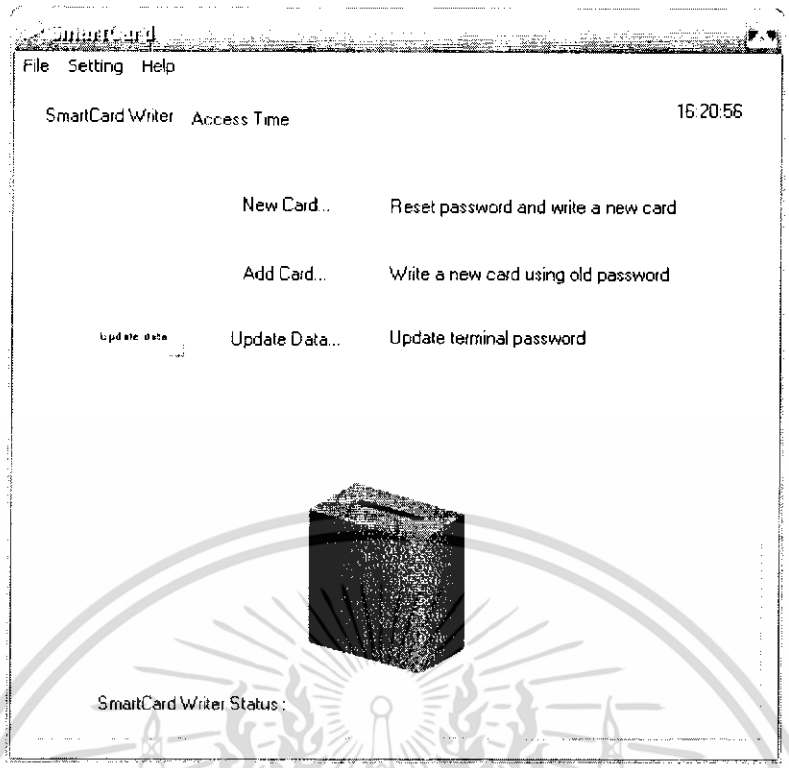
รูปที่ 4.25 ผลการทดลองในกระบวนการสิ้นสุดการทำงานของเครื่องอ่านและเขียนบัตรสมาร์ทการ์ด

จากการทดลองพบว่า ไมโครคอนโทรลเลอร์สามารถติดต่อกับสมาร์ทการ์ดได้ และแสดงผลการติดต่อออกมาทางพอร์ทอนุกรม โดยใช้โปรแกรม Hyper Terminal ในการรับข้อมูลมาแสดงที่หน้าจอคอมพิวเตอร์ได้

4.4 การทดลองการทำงานของโปรแกรมแอพพลิเคชั่นที่เขียนขึ้น

ในส่วนของการทดลองการทำงานนั้น เพื่อให้ง่ายต่อความเข้าใจ จึงขอนำรูปที่ 3.19 มาแสดงไว้อีกครั้ง ดังรูปที่ 4.26

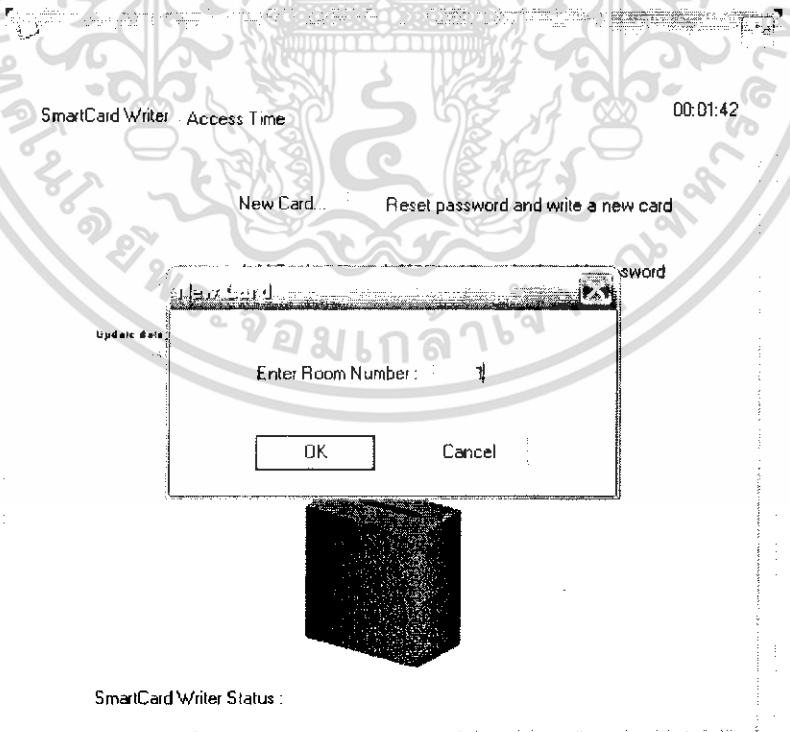
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 หน้าต่างโปรแกรมที่ใช้ควบคุมการทำงานของเครื่องอ่านบัตร

4.4.1 ผลการทดลองเมื่อกดปุ่ม New Card

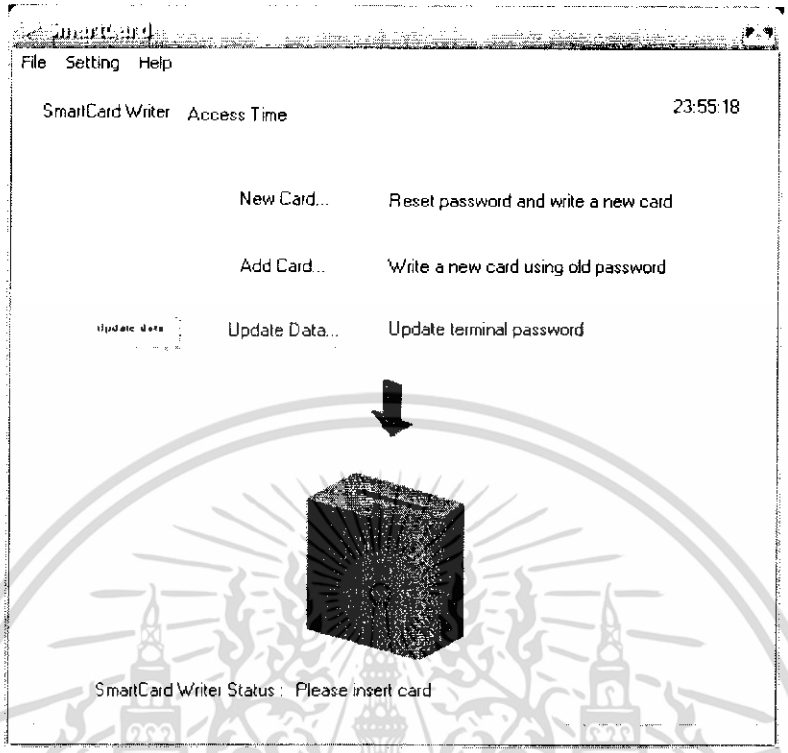
เมื่อกดปุ่ม New Card โปรแกรมจะมีหน้าต่างใหม่ขึ้นมาให้ใส่หมายเลขห้องที่ต้องการ ซึ่งจากตัวอย่างจะใส่ห้องหมายเลข 1 ดังรูปที่ 4.27



รูปที่ 4.27 หน้าต่างใหม่เมื่อกดปุ่ม New Card

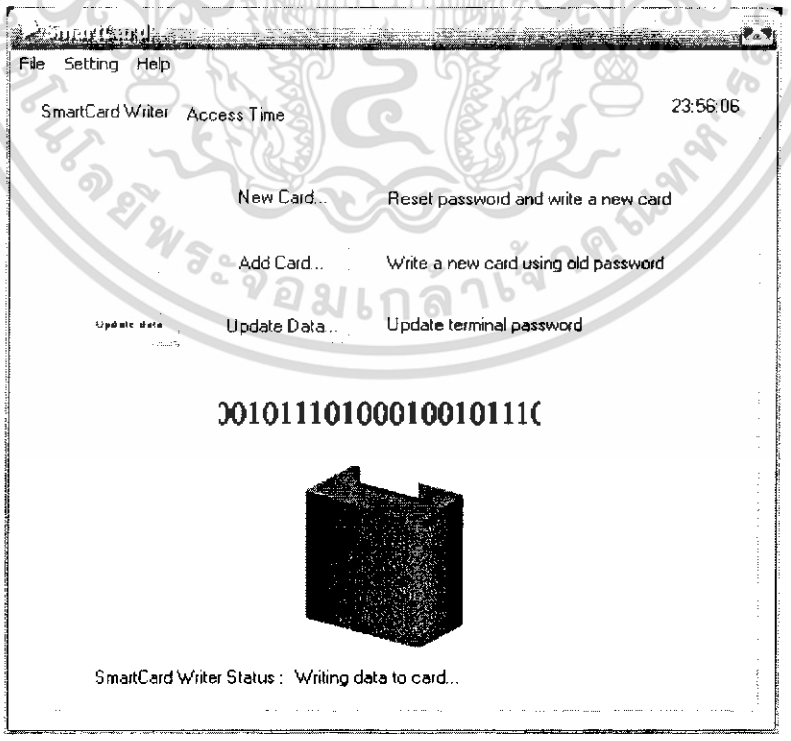
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นโปรแกรมจะแสดงข้อความและรูปภาพให้เสียบบัตรเข้าไปในเครื่องเขียนบัตร ดังรูปที่ 4.28



รูปที่ 4.28 หน้าต่างแสดงข้อความและรูปภาพให้เสียบบัตร

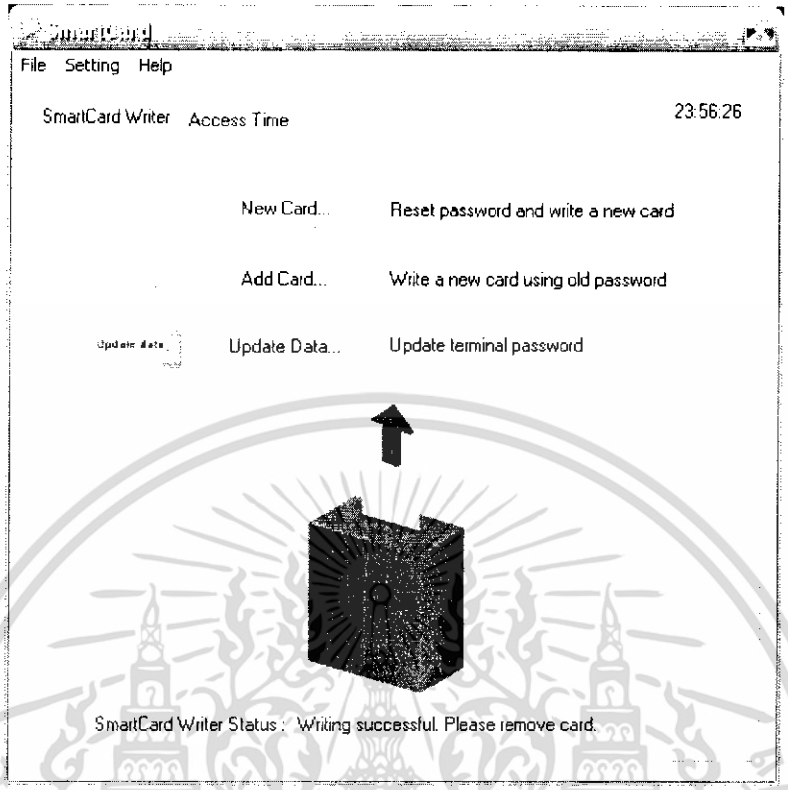
เมื่อเสียบบัตรเข้าไปแล้ว โปรแกรมก็จะส่งหมายเลขห้องและรหัสผ่านใหม่ไปให้เครื่องเขียนบัตรเพื่อทำการบันทึกลงในหน่วยความจำที่อยู่ในบัตร ดังรูปที่ 4.29



รูปที่ 4.29 หน้าต่างขณะเขียนข้อมูลลงในบัตร

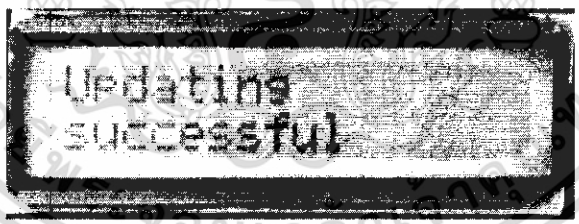
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากบันทึกหมายเลขห้องและรหัสผ่านลงไปแล้ว โปรแกรมก็จะแสดงข้อความและรูปภาพให้ดึงบัตรออก ดังรูปที่ 4.30



รูปที่ 4.30 หน้าต่างแสดงข้อความและรูปภาพให้ดึงบัตรออก

หลังจากบันทึกข้อมูลใหม่ลงในบัตรแล้ว โปรแกรมก็จะบันทึกรหัสผ่านใหม่ลงในฐานข้อมูล และส่งรหัสผ่านใหม่นี้ไปยังเครื่องลูก (slave) ที่ติดอยู่กับห้องนั้น เพื่อทำการอัพเดท ดังรูปที่ 4.31



รูปที่ 4.31 ข้อความบนจอแอลซีดีที่เครื่องลูก (slave)

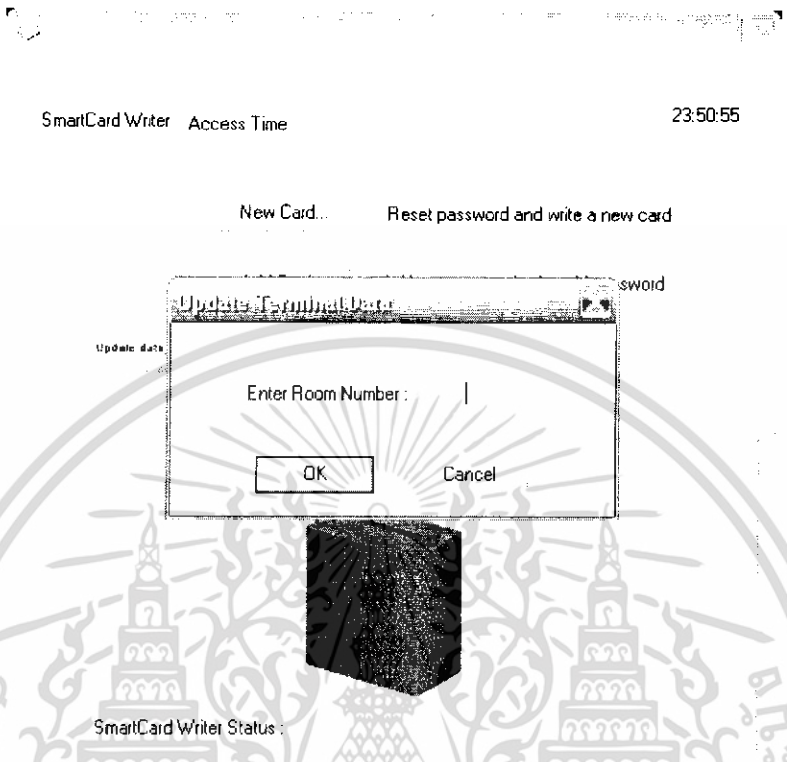
4.4.2 ผลการทดลองเมื่อกดปุ่ม Add Card

ผลของการกดปุ่ม Add Card นั้นจะมีลักษณะเหมือนกับการกดปุ่ม New Card แต่จะแตกต่างกันตรงที่การทำงาน โดยการทำงานของปุ่ม Add Card นั้นจะใช้รหัสผ่านเดิมที่อยู่ในฐานข้อมูลมาบันทึกลงหน่วยความจำที่อยู่ในบัตร และจะไม่มีการส่งรหัสผ่านใหม่ไปยังเครื่องลูกที่ติดอยู่กับห้องนั้น ปุ่มนี้จะใช้ในกรณีที่บัตรมีการสูญหายหรือต้องการที่จะเพิ่มบัตรให้กับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.3 ผลการทดลองเมื่อกดปุ่ม Update Data

เมื่อกดปุ่ม Update Data โปรแกรมจะมีหน้าต่างใหม่ขึ้นมาให้ใส่หมายเลขห้องที่ต้องการ ดังรูปที่ 4.32

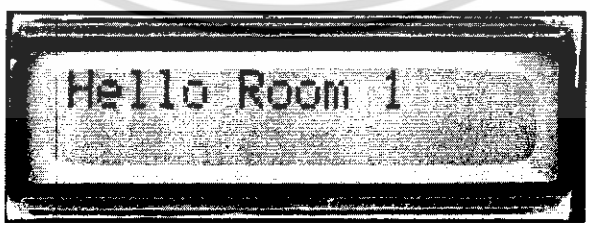


รูปที่ 4.32 หน้าต่างเมื่อกดปุ่ม Update Data

เมื่อใส่หมายเลขห้องที่ต้องการอัพเดทแล้ว โปรแกรมก็จะทำการส่งรหัสผ่านปัจจุบันที่มีอยู่ในฐานข้อมูลไปให้กับเครื่องลูกที่อยู่ติดกับห้องนั้น ซึ่งก็จะได้ผลเช่นเดียวกันกับรูปที่ 4.31

4.4.4 ผลการทดลองเมื่อนำบัตรมาใช้ในการเปิดประตูห้อง

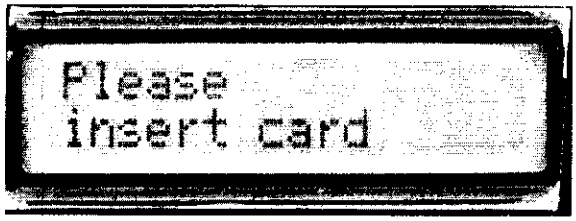
เมื่อมีการเปิดสวิตช์เพื่อจ่ายไฟให้กับเครื่องลูก (slave) หน้าจอแอลซีดีจะแสดงข้อความต้อนรับ ดังรูปที่ 4.33



รูปที่ 4.33 ข้อความต้อนรับบนจอแอลซีดี

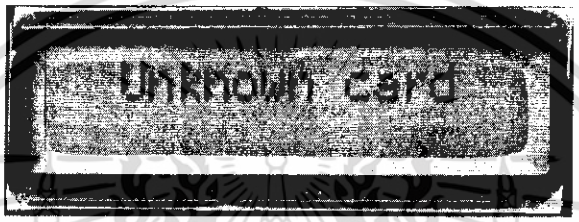
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นจะแสดงข้อความรอรับการเสียบบัตร ดังรูปที่ 4.34



รูปที่ 4.34 ข้อความรอรับการเสียบบัตร

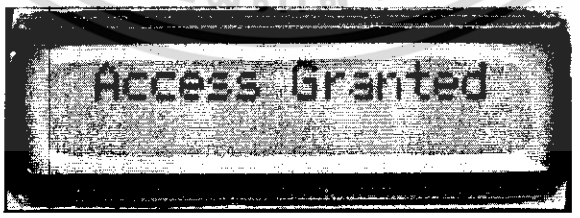
ถ้าหากว่าบัตรที่เสียบไปนั้นเป็นบัตรผิดประเภท หน้าจอก็จะแสดงข้อความเตือนและให้ดึงบัตรออก ดังรูปที่ 4.35 และรูปที่ 4.36 ตามลำดับ



รูปที่ 4.35 ข้อความเสียบบัตรผิดประเภท

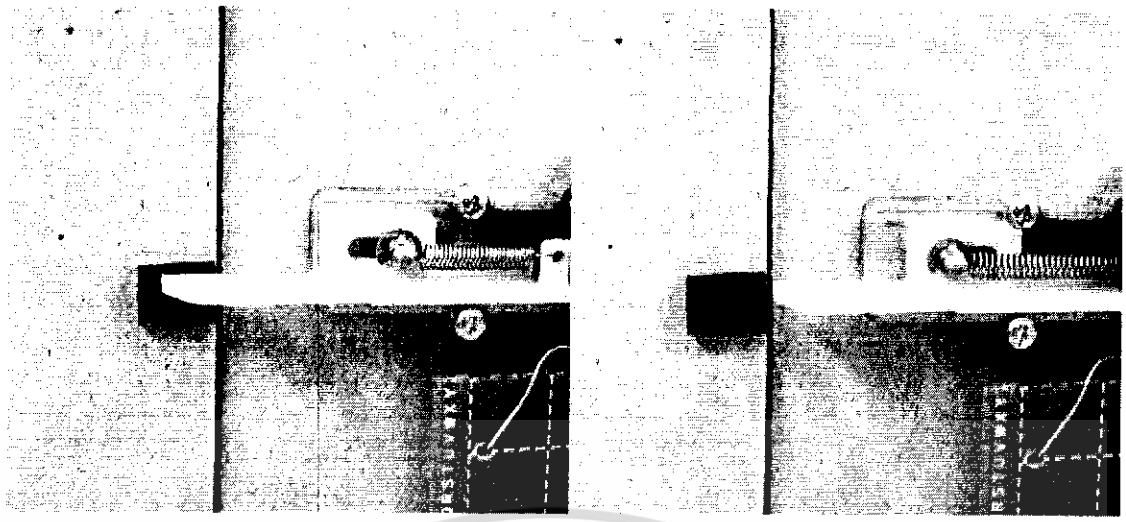
รูปที่ 4.36 ข้อความให้ดึงบัตรออก

ถ้าบัตรที่เสียบนั้นเป็นบัตรที่ถูกต้อง หน้าจอก็จะแสดงข้อความให้ผ่านแล้วส่งสัญญาณให้รีเลย์เพื่อเปิดประตู และหลังจากดึงบัตรออกแล้วประตูจะทำการล็อกภายในเวลา 3 วินาที ดังรูปที่ 4.37 , 4.38 และ 4.39 ตามลำดับ



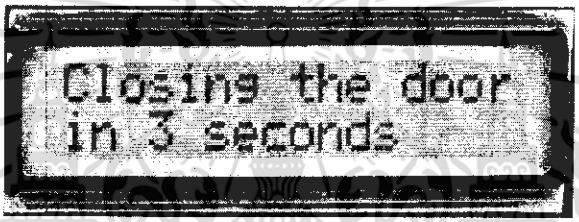
รูปที่ 4.37 ข้อความอนุญาตให้ผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) กลอนประตูเมื่อวงจรรีเลย์ไม่ทำงาน (ข) กลอนประตูเมื่อวงจรรีเลย์ทำงาน

รูปที่ 4.38 ลักษณะกลอนประตูที่ติดอยู่ที่ห้อง



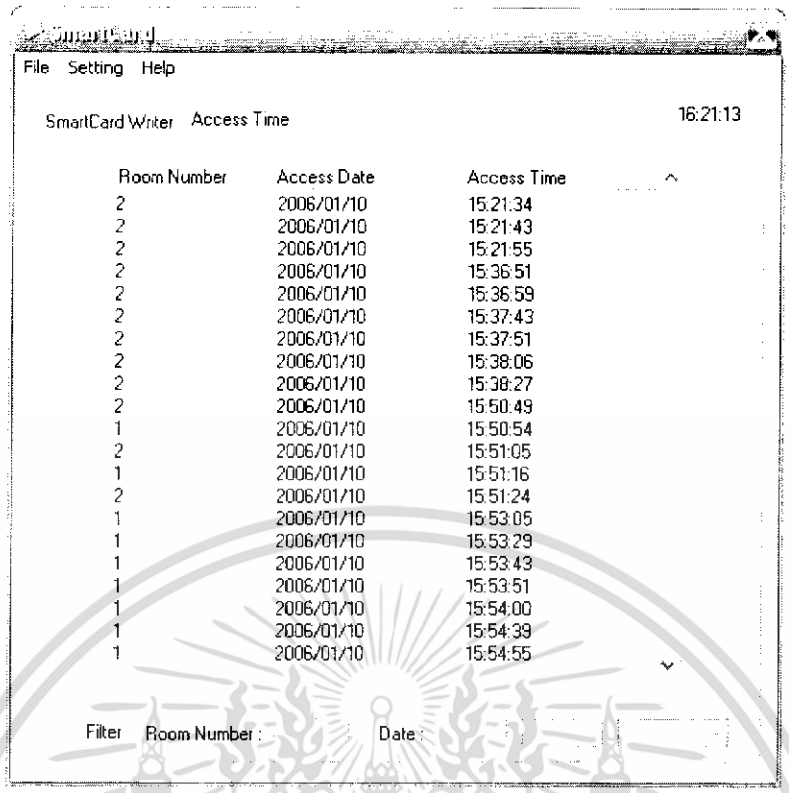
รูปที่ 4.39 ข้อความหลังจากดึงบัตรออก

ในกรณีที่บัตรนั้นเป็นบัตรถูกประเภท แต่รหัสผ่านไม่ถูกต้อง หน้าจอก็จะแสดงข้อความปฏิเสธ ดังรูปที่ 4.40

รูปที่ 4.40 ข้อความบัตรถูกประเภท แต่รหัสผ่านไม่ถูกต้อง

เมื่อผู้ใช้มีการเสียบบัตรเพื่อเข้าสู่ห้องพัก เครื่องลูกที่ติดอยู่กับห้องนั้นก็จะส่งข้อมูลกลับมายังคอมพิวเตอร์ว่ามีการเสียบบัตร คอมพิวเตอร์ก็จะทำการบันทึกเวลานั้นลงไปในฐานข้อมูล และนำมาแสดงบนหน้าจอ ดังรูปที่ 4.41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.41 หน้าต่างแสดงผลที่อ่านมาจากฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และบทสรุป

5.1 บทวิจารณ์

จากการทดลองการทำงานในส่วนต่างๆ ของเครื่องอ่านและเขียนสมาร์ทการ์ดในโครงงานนั้น ได้พบปัญหาและมีข้อสังเกต ดังนี้ คือ

5.1.1 ในส่วนกระบวนการอ่านข้อมูลภายในบัตรสมาร์ทการ์ด

กระบวนการในการอ่านข้อมูลภายในบัตรสมาร์ทการ์ด ของชิพที่บรรจุอยู่บนบัตรของแต่ละเบอร์นั้นจะมีกระบวนการอ่านข้อมูลที่แตกต่างกันในการที่จะเขียนโปรแกรมเพื่อให้บัตรสมาร์ทการ์ดแต่ละเบอร์สามารถอ่านข้อมูลได้นั้น จะต้องเป็นไปตามคุณสมบัติของการ์ดแต่ละเบอร์นั้น

ในโครงงานนี้ได้เลือกใช้สมาร์ทการ์ดเบอร์ SLE4442 เนื่องจากสามารถหาซื้อได้ง่าย ตามท้องตลาดประกอบกับราคาที่ไม่สูงจนเกินไปเมื่อเปรียบเทียบกับคุณสมบัติที่จะนำบัตรนั้นมาใช้งาน พร้อมทั้งสมาร์ทการ์ดเบอร์นี้ยังมีหน่วยความจำที่มีขนาดเพียงพอกับการนำไปประยุกต์ใช้งานในอนาคต ผนวกกับบัตรสมาร์ทการ์ดเบอร์นี้ยังเป็นบัตรแบบที่มีระบบการป้องกันความปลอดภัยให้กับข้อมูลที่บรรจุอยู่บนบัตร ทำให้สามารถนำบัตรไปใช้ในงานต่างๆ ที่ต้องการความปลอดภัยของข้อมูลได้อีกด้วย

ในการออกแบบเพื่อให้เครื่องอ่านและเขียนสมาร์ทการ์ดในโครงงานนี้สามารถที่จะอ่านข้อมูลภายในบัตรได้นั้นสามารถทำได้อย่างมีประสิทธิภาพแล้ว

5.1.2 ในส่วนกระบวนการเขียนข้อมูลลงบนบัตรสมาร์ทการ์ด

กระบวนการในการเขียนข้อมูลลงบนบัตรสมาร์ทการ์ด ของชิพที่บรรจุอยู่บนบัตรของแต่ละเบอร์นั้นจะมีกระบวนการเขียนข้อมูลที่แตกต่างกัน ในการที่จะเขียนโปรแกรมเพื่อให้บัตรสมาร์ทการ์ดแต่ละเบอร์สามารถเขียนข้อมูลได้นั้น จะต้องเป็นไปตามคุณสมบัติของการ์ดแต่ละเบอร์นั้น เช่นเดียวกับกระบวนการในการอ่านข้อมูล

กระบวนการที่สำคัญสำหรับการเขียนข้อมูลลงบนบัตรสมาร์ทการ์ดเบอร์ SLE4442 นั้นคือ ต้องมีการใส่รหัสผ่านก่อน (รหัสพีเอสซี: PSC=Programmable Security Code) การเข้าไปดำเนินกระบวนการเขียนข้อมูล ซึ่งรหัสผ่านนี้จะสามารถป้อนผิดได้เพียง 3 ครั้งติดต่อกันเท่านั้น ถ้าป้อนผิดเกิน 3 ครั้งติดต่อกัน ก็จะทำให้ไม่สามารถเข้าไปทำกระบวนการเขียนหรือเปลี่ยนแปลงข้อมูลที่อยู่ภายในบัตรได้อีกเลย แต่จะยังสามารถเข้าไปทำการอ่านข้อมูลภายในได้อยู่

5.1.3 ในส่วนของการนำไปประยุกต์ใช้งาน

ในโครงงานนี้ได้นำเครื่องอ่านและเขียนสมาร์ทการ์ดไปประยุกต์ใช้งานเป็นบัตรเปิดประตูห้องพัก โดยจะมีการบันทึกเวลาเมื่อผู้ใช้บริการเสียบบัตรเพื่อเปิดประตูลงในฐานข้อมูลของคอมพิวเตอร์เพื่อใช้ในการตรวจสอบในกรณีต่างๆ เช่น บัตรสูญหาย หรือ มีขโมยเข้าห้อง เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 บทสรุป

เครื่องอ่านและเขียนสมาร์ทการ์ดสำหรับการประยุกต์ใช้งานในระบบห้องพักนี้สามารถนำไปใช้งานได้จริงไม่ว่าจะเป็นอพาร์ทเมนท์ โรงแรม ห้องแลป หรือห้องโปรเจค ซึ่งสามารถที่จะเปลี่ยนรหัสผ่านได้ตลอดเวลา แต่ในโครงงานนี้ได้จำลองประตูห้องพักขึ้นโดยใช้กลอนประตูแบบโซลินอย เนื่องจากมีราคาถูก แต่มีข้อเสียคือไม่แข็งแรง จึงไม่สามารถที่จะนำไปใช้งานจริงได้ ถ้าจะนำไปใช้งานจริงจะต้องใช้กลอนประตูแบบแมกเนติกล็อก ซึ่งจะแข็งแรงกว่า แต่มีราคาแพง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ประทีป บัญญัตินพรัตน์, "การสื่อสารข้อมูล",สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2525

[2] รศ.สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ

[3] ธีรบูลย์ หล่อวิเชียรรุ่ง, นคร ภักดีชาติ และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, "ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C", Innovative Experiment Co, Ltd.

[4] ยุทธนา ลีลาศวัฒนกุล, "เริ่มต้นการเขียนโปรแกรมด้วยภาษา C++", สำนักพิมพ์อินโฟเพรส, นนทบุรี

[5] ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรม Microsoft Advanced Visual C++ version 6.0", สำนักพิมพ์ซัคเซสมีเดีย, กรุงเทพฯ

[6] ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรมด้วย Visual C++ 6.0 ฉบับโปรแกรมเมอร์", สำนักพิมพ์อินโฟเพรส, นนทบุรี

[7] ยุทธนา ลีลาศวัฒนกุล, "คู่มือการเขียนโปรแกรม Microsoft Advanced Visual C++ version 6.0 ฉบับ database programming", สำนักพิมพ์อินโฟเพรส, นนทบุรี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของไมโครคอนโทรลเลอร์

โปรแกรมที่ใช้ในการทดสอบการสื่อสารผ่านพอร์ทอนุกรม

ส่วนของโปรแกรมหลัก main.c

```
#include <AT89X52.H>
#include "serialport_h.h"

void main(void)
{
      serial_init();
      printf("Test Serial Communication\n\nCommunication Work!");
      while(1);
}
```

ส่วนของโปรแกรมย่อย serialport.c

```
#include <AT89X52.H>
#include "serialport_h.h"

void serial_init(void)
{
      PCON = 0x00;
      SCON = 0x50;
      TMOD = 0x20;
      TH1 = 0xFF;
      TR1 = 1;
}

void sndSerial(unsigned char dat)
{
      SBUF = dat;
      while(~TI);
      TI = 0;
}

unsigned char recSerial(void)
{
      unsigned char dat;
      while(~RI);
      RI = 0;
      dat = SBUF;
      return(dat);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ใช้ในกระบวนการอ่านและเขียนข้อมูลลงในสมาร์ทการ์ด

ส่วนของโปรแกรมย่อย smartcard.c

```
#include <AT89X51.H>
#include <intrins.h>
#include "smartcard_h.h"

sbit SDA = P3^5;                        /*Card Serial Data Line*/
sbit SCL = P3^4;                        /*Card Serial Clock Line*/
sbit SMC_RST = P3^3;                    /*Card Reset*/
sbit SMC_IN = P0^7;                     /*Check Card In*/
sbit SMC_PWR = P0^6;                    /*Card Power*/

void I2C_delay(void)                    /*delay about 13us*/
{
      _nop_();
      _nop_();
      _nop_();
      _nop_();
      _nop_();
      _nop_();
      _nop_();
      _nop_();
      _nop_();
      _nop_();
}

void delay_s(void)                      /*delay about 4us*/
{
      _nop_();
      _nop_();
      _nop_();
      _nop_();
}

void I2C_high(void)                     /*build clock high*/
{
      SCL = 1;
      I2C_delay();                       /*SCL = 1*/
}

void I2C_low(void)                      /*build clock low*/
{
      SCL = 0;
      I2C_delay();                       /*SCL = 0*/
}

void I2C_start(void)                    /*build start event*/
{
      SDA = 1;
      I2C_high();
      SDA = 0;
      I2C_delay();
      I2C_low();
      SDA = 1;                          /*SDA = 1,SCL = 0*/
}

void I2C_stop(void)                     /*build stop event*/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
      SDA = 0;
      I2C_high();
      SDA = 1;                                  /*SDA = 1,SCL = 1*/
}

void I2C_break(void)
{
      SMC_RST = 0;
      SCL = 0;
      SDA = 0;
      SMC_RST = 1;
      I2C_delay();
      SDA = 1;
      I2C_delay();
      SMC_RST = 0;
}

void ATR(unsigned char* datATR)      /*function Answer-to-Reset*/
{
      unsigned char a,j,dat;
      bit inbit;
      SMC_PWR = 1;
      for(j = 0;j<=3;j++)
      {
            dat = 0x00;
            for(a = 0;a<=7;a++)
            {
                  SCL = 0;                          /*SCL = 0*/
                  if(a+j == 0)
                  {
                        SMC_RST = 1;                /*card reset = hi*/
                        delay_s();                  /*delay 4us*/
                        I2C_high();                 /*SCl = 1*/
                        SCL = 0;                    /*SCL = 0*/
                        delay_s();                  /*delay 4us*/
                        SMC_RST = 0;                /*card reset low*/
                        I2C_delay();
                  }                                 /*SCL = 0*/

                  I2C_high();                       /*SCL = 1*/
                  inbit = SDA;                      /*receive data*/
                  dat = dat >> 1;                   /*shift right*/
                  if(inbit == 1)
                        dat = dat | 0x80;

                  I2C_low();                        /*SCL = LOW*/
            }
            *(datATR+j) = dat;
      }
}

void I2C_wrbyte(unsigned char dat)
{
      unsigned char i;
      bit outbit;
      for(i=1;i<=8;i++)
      {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            outbit = dat & 0x01;
            SDA = outbit;
            dat = dat >> 1;
            I2C_high();
            I2C_low();                          /*SCL = 0*/

      }
}

unsigned char I2C_rdbyte(void)
{
      unsigned char i,dat;
      bit inbit;
      dat = 0x00;
      for(i = 1;i<=8;i++)
      {
            I2C_high();
            inbit = SDA;                        /*receive data from Card*/
            dat = dat >> 1;
            if(inbit == 1)
                  dat = dat | 0x80;

            I2C_low();                          /*SCL = LOW*/
      }
      return(dat);                            /* C_PW = 0,SCL = 0*/
}

void com_proc(unsigned char mode,unsigned char addr,unsigned char
dat)
{
      unsigned char com,num_CLK,z;
      switch(mode)
            {
                  case 1:
                        com = 0x38;             /*Update Main Memory*/
                        num_CLK = 0x0FD;        /*FD = 253*/
                  break;
                  case 2:
                        com = 0x3C;             /*Protection Memory*/
                        num_CLK = 0x7A;         /*7D = 122*/
                  break;
                  case 3:
                        com = 0x39;             /*Security Memory*/
                        num_CLK = 0x0FD;        /*FD = 253*/
                  break;
                  case 4:
                        com = 0x33;             /* Verification Data*/
                        num_CLK = 0x01;         /*one pulse*/
                  break;

            }

      I2C_start();                            /*SDA = 1,SCL = 0*/
      I2C_wrbyte(com);
      I2C_wrbyte(addr);
      I2C_wrbyte(dat);
      I2C_stop();                             /*SDA = 1,SCL = 1*/
      SCL = 0;                               /*SDA = 1,SCl = 0*/
      _nop_();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SDA = 0;                            /*SDA = 0,SCl = 0*/
        I2C_delay();                          /*Start of Processing*/
        for(z = 1;z<=num_CLK;z++)             /*build pulse = num_CLK*/
        {
              I2C_high();
              I2C_low();
        }                                     /*SCL = 0,SDA = 0*/
        I2C_high();                           /*SCL = 1,SDA = 0*/
        SCL = 0;                            /*SCL = 0,SDA = 0*/
        _nop_();
        SDA = 1;                             /*SCL = 0,SDA = 1*/
        I2C_delay();                          /* End of Procssing*/
        I2C_high();                           /*build addition pulse*/
        I2C_low();                            /*                    */
}

void com_outgo(unsigned char mode,unsigned char addr,unsigned char*
buff)
{
        unsigned char a,com,dat = 0x00;
        switch(mode)
        {
              case 1:
                    com = 0x34;                 /*Protection Memory*/
              break;
              case 2:
                    com = 0x31;                 /*Security Memory*/
              break;
        }

        for(a = 0;a<=3;a++)
        {
              if(a == 0)
              {
                    I2C_start();                          /*SDA = 1,SCL = 0*/
                    I2C_wrbyte(com);
                    I2C_wrbyte(addr);
                    I2C_wrbyte(dat);
                    I2C_stop();                         /*SDA = 1,SCL = 1*/
                    I2C_low();                          /*SDA = 1,SCL = 0*/
              }

              *(buff+a) = I2C_rdbyte();             /*store to memory*/
        }
        I2C_high();                           /*build addition pulse*/
        I2C_low();                            /*                    */
}

unsigned char com_rd(unsigned char addr)
{
        unsigned char com,dat;
        com = 0x30;
        dat = 0x0FF;

        I2C_start();                          //SDA = 1,SCL = 0
        I2C_wrbyte(com);
        I2C_wrbyte(addr);
        I2C_wrbyte(dat);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        I2C_stop();                             //SDA = 1,SCL = 1
        I2C_low();                              //SDA = 1,SCL = 0

        dat = I2C_rdbyte();                 //store content to memory
        I2C_high();                             //build addition pulse
        I2C_low();
        I2C_break();
        return(dat);
}

unsigned char veri_PSC(void)
{
        unsigned char securMem[4];
        com_outgo(2,0x00,&securMem);                    //READ SM
        if(securMem[0] == 0)                    //test card lock
                return 0x00;
        else
        {
                com_proc(3,0x00,0x03);                  //Write free bit
        com_proc(4,0x01,0x0FF);                 //Reference Data Byte 1
                com_proc(4,0x02,0x0FF);                 //Reference Data Byte 2
                com_proc(4,0x03,0x0FF);                 //Reference Data Byte 3
                com_proc(3,0x00,0x0FF);                 //Erase Error Counter
                com_outgo(2,0x00,&securMem);                    //READ SM
        if(securMem[0] == 0x07)
                        return 0xFF;
                else
                {
                 if(securMem[0] == 0x03)
                        return 0x02;
                  else
                        return 0x01;
                }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้