สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

T054933

สมาร์ทการ์ดของระบบสารสนเทศนักศึกษา

SMART CARD OF STUDENT INFORMATION SYSTEM

โดย

นางสาวอมรรัตน์ จิรัฐิติกาลโชติ

นายโอฬาร พิสิฐสวัสดิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เลขหมู่..........

เลขทะเบียน 54933

วัน,เดือน,ปี - 1 เม.ย. 2548

๖..........

๑..........

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ทการ์ดของระบบสารสนเทศนักศึกษา

SMART CARD OF STUDENT INFORMATION SYSTEM

โดย

นางสาวอมรรัตน์ จิรัฐิติกาลโชติ 43010525

นายโอฬาร พิสิฐสวัสดิ์ 43010568

อาจารย์ที่ปรึกษา

รศ.ดร. กอบชัย เดชหาญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **สมาร์ทการ์ดของระบบสารสนเทศนักศึกษา**

SMART CARD OF STUDENT INFORMATION SYSTEM

ผู้จัดทำ

1. นางสาวอมรรัตน์	**จิรัฐิติกาลโชติ**	**43010525**
2. นายโอพาร	**พิสิฐสวัสดิ์**	**43010568**

.... อาจารย์ที่ปรึกษา

(รศ.ดร. กอบชัย เดชหาญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ทการ์ดของระบบสารสนเทศนักศึกษา

SMART CARD OF STUDENT INFORMATION SYSTEM

โดย นางสาวอมรรัตน์ จิรัฐิติกาลโชติ 43010525

นายโอพาร พิสิฐสวัสดิ์ 43010568

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ

บทคัดย่อ

ในปัจจุบันนี้การเข้าสู่ระบบสารสนเทศนักศึกษาส่วนบุคคล ไม่ว่าจะเป็นประวัตินักศึกษา ระบบการลงทะเบียน การตรวจผลการศึกษา หรือตารางสอน-ตารางสอบนั้น ต้องผ่านขั้นตอนหลายขั้นตอน ทางผู้จัดทำจึงมีความคิดที่จะพัฒนาขั้นตอนการดำเนินการเกี่ยวกับระบบสารสนเทศนักศึกษาให้มีความสะดวกและรวดเร็วมากยิ่งขึ้น โดยการจัดทำระบบสารสนเทศนักศึกษาส่วนบุคคลที่สามารถเข้าถึงได้ง่าย โดยการใช้สมาร์ทการ์ดประจำตัวนักศึกษาคนนั้น ๆ ไม่เพียงเท่านี้ ทางผู้จัดทำยังได้เพิ่มเติมในส่วนของรายละเอียดการยืม-คืนหนังสือห้องสมุดและประวัติสุขภาพของนักศึกษาเข้าไป เพื่อให้ได้ระบบข้อมูลสารสนเทศที่มีความสมบูรณ์ยิ่งขึ้น โดยในส่วนของฐานข้อมูลระบบสารสนเทศนักศึกษานั้น ทางผู้จัดทำได้เลือกใช้ภาษา Visual Basic และนำสมาร์ทการ์ดมาใช้แทนบัตรนักศึกษา โดยมีการจัดทำเครื่องอ่าน-เขียนสมาร์ทการ์ดซึ่งใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงาน

ABSTRACT

Today, the access to the data of individual student information system such as personal academic record, registration record, class schedule of examination schedule must be passed through many steps and processes. Therefore, the developers aim to solve all problems by mean of creating the student information system that is easy to access by using student's personal smart card. Moreover, the developers hope to create this system to cover the records of borrowing books from library and also student's personal health record and other useful records related in order to make this system much more useful and most profitable to all students in the system. Contenting data base are written by Visual Basic, this smart card could also use as the student's ID card and the read and write smart card machine is controlled by microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	
1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์	1
1.2 ขอบเขตของปริญญานิพนธ์	1
1.3 ประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์	1
1.4 เนื้อหาของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎี	
2.1 สมาร์ทการ์ดคืออะไร	3
2.2 ประวัติความเป็นมาของสมาร์ทการ์ด	3
2.3 ส่วนประกอบและโครงสร้างของสมาร์ทการ์ด	4
2.3.1 ตัวบัตรพลาสติก	4
2.3.2 หน้าสัมผัสและชิปสมาร์ทการ์ด	4
2.4 การประกอบสมาร์ทการ์ดโมดูลลงในบัตรพลาสติก	5
2.4.1 การสร้างสมาร์ทการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก	5
2.4.2 การสร้างสมาร์ทการ์ดด้วยวิธีวางสมาร์ทการ์ดโมดูลลงในเนื้อบัตร	5
2.4.3 การสร้างสมาร์ทการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวของบัตร	5
2.5 ข้อกำหนดอีเอ็มวี	6
2.6 รายละเอียดพื้นฐานของสมาร์ทการ์ด	7
2.7 ชนิดของสมาร์ทการ์ด	9
2.7.1 การ์ดหน่วยความจำ	10
2.7.2 การ์ดชนิดโปรเซสเซอร์	11
2.8 การ์ดที่มีระบบป้องกันข้อมูล	13
2.8.1 คุณสมบัติทั่วไปของสมาร์ทการ์ดเบอร์ SLE 4442	13
2.8.2 รูปแบบการสื่อสารข้อมูลของสมาร์ทการ์ดเบอร์ SLE 4442	15
2.9 ไมโครคอนโทรลเลอร์ ตระกูล MCS-51	26
2.9.1 คุณลักษณะพื้นฐานของ MCS-51	27
2.9.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051	27
2.9.3 พอร์ตชนิดอนุกรมอยู่ภายใน	30
2.10 แนวทางการใช้ LCD Module	30
2.10.1 โครงสร้างภายในของตัวควบคุม LCD Module	31
2.10.2 LCD Module ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16x1)	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.10.3 คำสั่งควบคุม LCD Module	33
2.10.4 การเขียนคำสั่งและข้อมูลให้แก่ LCD Module	37
2.10.5 จังหวะการทำงานของ LCD Module	38
2.11 พอร์ต RS-232C	39
2.11.1 มาตรฐาน RS-232C	39
2.11.2 ลักษณะของสัญญาณ RS-232C	40
2.11.3 การกำหนดข้อต่อของพอร์ต RS-232C	41
2.11.4 การเชื่อมต่อมาตรฐาน RS-232C	43
2.12 โปรแกรมวิชวลเบสิก	45
2.12.1 โปรแกรมติดต่อและควบคุมผ่านพอร์ตอนุกรม	45
2.12.2 โปรแกรมเพื่อสร้างระบบฐานข้อมูล	46
บทที่ 3 การออกแบบและการสร้าง	
3.1 การออกแบบและสร้างเครื่องอ่าน-เขียนสมาร์ทการ์ด	48
3.2 โปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด	51
3.2.1 โปรแกรมการกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว	67
3.2.2 โปรแกรมการส่งข้อมูลออกพอร์ตอนุกรม	69
3.2.3 โปรแกรมการรับข้อมูลจากพอร์ตอนุกรม	71
3.2.4 โปรแกรมการอ่านข้อมูลจากบัตรสมาร์ทการ์ดและแสดงผลทางจอแสดงผลแบบผลึกเหลว	73
3.2.5 โปรแกรมการเปรียบเทียบรหัสผ่าน	75
3.2.6 โปรแกรมการเขียนข้อมูลลงบัตรสมาร์ทการ์ด	77
3.3 โปรแกรมควบคุมคอมพิวเตอร์	78
บทที่ 4 การทดลองและผลการทดลอง	
4.1 ผลการทดลองวัดสัญญาณต่าง ๆ จากเครื่องอ่าน-เขียนสมาร์ทการ์ด	105
4.1.1 ผลการทดลองสัญญาณการรีเซต	105
4.1.2 ผลการทดลองสัญญาณการตอบรับการรีเซต	106
4.1.3 ผลการทดลองสัญญาณการเปรียบเทียบรหัสผ่าน	107
4.1.4 ผลการทดลองสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก	108
4.1.5 ผลการทดลองสัญญาณการเขียนข้อมูลลงหน่วยความจำหลัก	109
4.2 ผลการทดลองของเครื่องอ่าน-เขียนสมาร์ทการ์ดในส่วนของจอแสดงผลแบบผลึกเหลวและผลการทดลองในหน้าจอคอมพิวเตอร์	110

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.2.1 เมื่อเริ่มการทำงาน	110
4.2.2 เมื่อทำการใส่รหัสผ่าน	111
4.2.3 เมื่อเข้าสู่ระบบการทำงานหลัก	112
4.2.4 เมื่อเข้าสู่งานทะเบียน	114
4.2.4.1 เมื่อเข้าสู่ระบบสารสนเทศนักศึกษา	115
4.2.4.1.1 เมื่อข้อสู่ข้อมูล-สถานะนักศึกษา	116
4.2.4.1.2 เมื่อเข้าสู่ระบบลงทะเบียน	117
4.2.4.1.3 เมื่อเข้าสู่ตารางสอน-ตารางสอบ	120
4.2.4.1.4 เมื่อเข้าสู่ผลการศึกษา	121
4.2.4.1.5 เมื่อเข้าสู่เปลี่ยนรหัสผ่าน	122
4.2.4.2 เมื่อเข้าสู่การป้อนข้อมูลนักศึกษา	125
4.2.4.3 เมื่อเข้าสู่การกำหนดวิชาที่เปิดสอน	129
4.2.5 เมื่อเข้าสู่งานห้องสมุด	131
4.2.5.1 เมื่อเข้าสู่การยืม-คืนหนังสือ	132
4.2.5.2 เมื่อเข้าสู่การจัดการข้อมูลหนังสือในห้องสมุด	138
4.2.6 เมื่อเข้าสู่งานอนามัย	139
บทที่ 5 บทสรุปและวิจารณ์	144
ภาคผนวก	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูป	หน้า
รูปที่ 2.1 แสดงรูปบัตรสมาร์ทการ์ด	7
รูปที่ 2.2 แสดงการแบ่งสมาร์ทการ์ดตามชนิดของหน่วยความจำ	9
รูปที่ 2.3 แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดหน่วยความจำ	10
รูปที่ 2.4 แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดโปรเซสเซอร์	12
รูปที่ 2.5 แสดงขาต่าง ๆ บนหน้าสัมผัสของบัตรสมาร์ทการ์ด	12
รูปที่ 2.6 แสดงบล็อกไดอะแกรมแสดงโครงสร้างภายในของสมาร์ทการ์ดเบอร์ SLE4442	14
รูปที่ 2.7 แสดงไดอะแกรมแสดงภาพรวมของการ์ดที่มีระบบป้องกันข้อมูล	15
รูปที่ 2.8 แสดงรูปสัญญาณของการรีเซตและการตอบรับการรีเซตด้วย ATR	16
รูปที่ 2.9 แสดงรูปสัญญาณของการส่งคำสั่งไปยังการ์ด	18
รูปที่ 2.10 แสดงรูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำหลัก	19
รูปที่ 2.11 แสดงรูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน	20
รูปที่ 2.12 แสดงรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก แบบการลบข้อมูลแล้วเขียนข้อมูลซ้ำ	21
รูปที่ 2.13 แสดงรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก แบบการลบหรือเขียนข้อมูล (อย่างใดอย่างหนึ่ง)	22
รูปที่ 2.14 แสดงรูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำปลอดภัย	23
รูปที่ 2.15 แสดงรูปสัญญาณของการเปรียบเทียบและพิสูจน์ข้อมูล	24
รูปที่ 2.16 แสดงกระบวนการเปรียบเทียบรหัสผ่านกับรหัส PSC	25
รูปที่ 2.17 แสดงรูปสัญญาณของโหมดการประมวลผล	26
รูปที่ 2.18 แสดงโครงสร้างภายในของ 8051	27
รูปที่ 2.19 การจัดขามาตรฐานของของไมโครคอนโทรลเลอร์ MCS-51	28
รูปที่ 2.20 แสดงไดอะแกรมการทำงานของ LCD Module แบบอักขระ	31
รูปที่ 2.21 แสดงรูปร่างและการจัดขา LCD Module แบบอักขระ	32
รูปที่ 2.22 การใช้งานพอร์ต RS-232C เชื่อมต่ออุปกรณ์	40
รูปที่ 2.23 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS-232C	41
รูปที่ 2.24 คอนเน็คเตอร์อนุกรม 9 ขา	41
รูปที่ 2.25 คอนเน็คเตอร์อนุกรม 25 ขา	42
รูปที่ 2.26 คุณลักษณะโดยย่อของสัญญาณ RS-232C	44
รูปที่ 2.27 การเปลี่ยนแปลงสัญญาณทีทีแอล (TTL)	45
รูปที่ 3.1 แสดงบล็อกไดอะแกรมเครื่องอ่าน-เขียนสมาร์ทการ์ด	48
รูปที่ 3.2 แสดงวงจรของเครื่องอ่าน-เขียนสมาร์ทการ์ด	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูป	หน้า
รูปที่ 3.3 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด	51
รูปที่ 3.4 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	52
รูปที่ 3.5 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	53
รูปที่ 3.6 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	54
รูปที่ 3.7 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	55
รูปที่ 3.8 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	56
รูปที่ 3.9 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	57
รูปที่ 3.10 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	58
รูปที่ 3.11 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	59
รูปที่ 3.12 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	60
รูปที่ 3.13 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	61
รูปที่ 3.14 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	62
รูปที่ 3.15 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)	63
รูปที่ 3.16 แสดงโฟลวชาร์ทของโปรแกรมกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว	67
รูปที่ 3.17 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการส่งข้อมูลออกพอร์ตอนุกรม	69
รูปที่ 3.18 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการรับข้อมูลจากพอร์ตอนุกรม	71
รูปที่ 3.19 แสดงโฟลวชาร์ทของโปรแกรมการอ่านข้อมูลจากบัตรสมาร์ทการ์ด และแสดงผลทางจอแสดงผลแบบผลึกเหลว	73
รูปที่ 3.20 แสดงโฟลวชาร์ทของโปรแกรมการเปรียบเทียบรหัสผ่าน	75
รูปที่ 3.21 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการเขียนข้อมูลลงบัตรสมาร์ทการ์ด	77
รูปที่ 3.22 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์	78
รูปที่ 3.23 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	79
รูปที่ 3.24 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	80
รูปที่ 3.25 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	81
รูปที่ 3.26 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	82
รูปที่ 3.27 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	83
รูปที่ 3.28 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	84
รูปที่ 3.29 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	85
รูปที่ 3.30 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	86
รูปที่ 3.31 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	87
รูปที่ 3.32 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	88

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูป	หน้า
รูปที่ 3.33 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	89
รูปที่ 3.34 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	90
รูปที่ 3.35 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	91
รูปที่ 3.36 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	92
รูปที่ 3.37 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	93
รูปที่ 3.38 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	94
รูปที่ 3.39 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	95
รูปที่ 3.40 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	96
รูปที่ 3.41 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	97
รูปที่ 3.42 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)	98
รูปที่ 4.1 แสดงสัญญาณการรีเซต	105
รูปที่ 4.2 แสดงสัญญาณตอบรับการรีเซต	106
รูปที่ 4.3 แสดงสัญญาณการเปรียบเทียบรหัสผ่าน	107
รูปที่ 4.4 แสดงสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก	108
รูปที่ 4.5 แสดงสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก (ภาพขยาย)	108
รูปที่ 4.6 แสดงสัญญาณการเขียนข้อมูลลงหน่วยความจำหลัก	109
รูปที่ 4.7 แสดงส่วนแสดงผลเมื่อเริ่มการทำงาน	110
รูปที่ 4.8 แสดงส่วนแสดงผลเมื่อใส่รหัสผ่าน	111
รูปที่ 4.9 แสดงส่วนแสดงผลเมื่อเข้าสู่ระบบการทำงานหลัก	113
รูปที่ 4.10 แสดงส่วนแสดงผลเมื่อเข้าสู่งานทะเบียน	114
รูปที่ 4.11 แสดงส่วนแสดงผลเมื่อเข้าสู่ระบบสารสนเทศนักศึกษา	115
รูปที่ 4.12 แสดงส่วนแสดงผลเมื่อเข้าสู่ข้อมูล-สถานะนักศึกษา	116
รูปที่ 4.13 แสดงส่วนแสดงผลเมื่อเข้าสู่ระบบลงทะเบียน	117
รูปที่ 4.14 แสดงส่วนแสดงผลเมื่อลงทะเบียนเกิน 22 หน่วย	118
รูปที่ 4.15 แสดงส่วนแสดงผลที่แสดงรายละเอียดของการลงทะเบียน	119
รูปที่ 4.16 แสดงส่วนแสดงผลเมื่อเข้าสู่ตารางสอน-ตารางสอบ	120
รูปที่ 4.17 แสดงส่วนแสดงผลเมื่อเข้าสู่ผลการศึกษา	121
รูปที่ 4.18 แสดงส่วนแสดงผลเมื่อกรอกรหัสผ่านใหม่ในช่องข้อมูลทั้ง 2 ช่องไม่ตรงกัน	122
รูปที่ 4.19 แสดงส่วนแสดงผลเมื่อกรอกรหัสผ่านใหม่ไม่ครบ 6 อักขระ	123
รูปที่ 4.20 แสดงส่วนแสดงผลเมื่อรหัสผ่านใหม่ที่กรอกนั้นถูกต้องตามที่กำหนดและระบบได้ทำการเปลี่ยนรหัสผ่านให้แล้ว	124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูป	หน้า
รูปที่ 4.21 แสดงส่วนแสดงผลเมื่อกรอกข้อมูลที่ต้องการบันทึกเรียบร้อยแล้ว	125
รูปที่ 4.22 แสดงส่วนแสดงผลเพื่อถามความแน่ใจว่าต้องการทำบัตรใหม่หรือไม่	126
รูปที่ 4.23 แสดงส่วนแสดงผลเมื่อทำการบันทึกข้อมูลเรียบร้อยแล้ว	127
รูปที่ 4.24 แสดงส่วนแสดงผลเมื่อบันทึกข้อมูลลงในบัตรสมาร์ทการ์ดเรียบร้อยแล้ว	128
รูปที่ 4.25 แสดงส่วนแสดงผลเมื่อบันทึกข้อมูลลงในบัตรสมาร์ทการ์ดเรียบร้อยแล้ว (ต่อ)	129
รูปที่ 4.26 แสดงส่วนแสดงผลเมื่อเข้าสู่การกำหนดวิชาที่เปิดสอน	129
รูปที่ 4.27 แสดงส่วนแสดงผลที่ให้กำหนดวิชาที่เปิดสอน	130
รูปที่ 4.28 แสดงส่วนแสดงผลเมื่อเข้าสู่งานห้องสมุด	131
รูปที่ 4.29 แสดงส่วนแสดงผลเมื่อเข้าสู่การยืม-คืนหนังสือ	132
รูปที่ 4.30 แสดงส่วนแสดงผลเมื่อเลือก "เพิ่มข้อมูล" ในหน้าต่างการยืม-คืนหนังสือ	133
รูปที่ 4.31 แสดงส่วนแสดงผลเมื่อทำการบันทึกข้อมูลชื่อหนังสือที่ทำการยืมแล้ว	134
รูปที่ 4.32 แสดงส่วนแสดงผลเพื่อให้เลือกหนังสือที่นำมาคืน	135
รูปที่ 4.33 แสดงส่วนแสดงผลเมื่อเลือกหนังสือที่นำมาคืน	136
รูปที่ 4.34 แสดงส่วนแสดงผลเมื่อเลือก "ค้นหาข้อมูล" ในหน้าต่างการยืม-คืนหนังสือ	137
รูปที่ 4.35 แสดงส่วนแสดงผลเมื่อเข้าสู่การจัดการข้อมูลหนังสือในห้องสมุด	138
รูปที่ 4.36 แสดงส่วนแสดงผลเมื่อเลือก "เพิ่มข้อมูล" ในหน้าต่างการจัดการข้อมูลหนังสือในห้องสมุด	139
รูปที่ 4.37 แสดงส่วนแสดงผลเมื่อเข้าสู่งานอนามัย	140
รูปที่ 4.38 แสดงส่วนแสดงผลที่แสดงข้อความเพื่อให้เลือกชื่อโรคและชื่อยาที่แพ้	141
รูปที่ 4.39 แสดงส่วนแสดงผลเมื่อที่แสดงหน้าต่างเพื่อให้เลือกชื่อโรคและชื่อยาที่แพ้	142
รูปที่ 4.40 แสดงส่วนแสดงผลที่แสดงชื่อโรคประจำตัวและชื่อยาที่แพ้ของนักศึกษาที่อ่านได้จากบัตรสมาร์ทการ์ด	143

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์

การพัฒนาเทคโนโลยีด้านต่าง ๆ ในทุกวันนี้ล้วนมีจุดมุ่งหมายเพื่อให้มนุษย์มีชีวิตความเป็นอยู่ที่ดีขึ้น มีความสะดวกสบายและปลอดภัยมากยิ่งขึ้น สมาร์ทการ์ด (Smart card) ก็เป็นหนึ่งในการพัฒนาเพื่อเหตุผลนี้เช่นเดียวกัน โดยสมาร์ทการ์ดเป็นการพัฒนาความสามารถของบัตร (card) ที่ใช้ในการเก็บข้อมูลให้มีขนาดของหน่วยความจำที่ใหญ่ขึ้นและมีความปลอดภัยของข้อมูลภายในบัตรสูงขึ้นกว่าบัตรแถบแม่เหล็ก (Magnetic card) ที่ใช้กันอยู่เดิม และด้วยคุณสมบัติเช่นนี้ ทำให้มีแนวโน้มในการที่จะทำสมาร์ทการ์ดไปใช้ในการเก็บข้อมูลสำคัญต่าง ๆ มากขึ้นเรื่อย ๆ

สำหรับในประเทศไทยนั้น ได้เริ่มมีการนำสมาร์ทการ์ดไปใช้ในงานด้านต่าง ๆ บ้างแล้ว เช่น บัตรเอทีเอ็ม (ATM), บัตรประจำตัวพนักงาน, บัตรโทรศัพท์ เป็นต้น และอนาคตรัฐบาลก็มีแผนการที่จะนำสมาร์ทการ์ดมาใช้ทำบัตรประจำตัวประชาชน โดยจะให้สมาร์ทการ์ดทำหน้าที่เป็นตัวแทนในการเก็บข้อมูลพื้นฐานต่าง ๆ เช่น ชื่อ-นามสกุล, ที่อยู่ตามทะเบียนราษฎร์, วันเดือนปีเกิด, กรุ๊ปเลือด และหมายเลขประจำตัว เป็นต้น และแนวคิดในการที่จะนำสมาร์ทการ์ดไปทำเป็นบัตรประจำตัวประชาชนนี่เอง ที่ทำให้ทางผู้จัดเกิดความคิดในการที่จะนำสมาร์ทการ์ดมาทำเป็นบัตรประจำตัวนักศึกษา โดยให้สมาร์ทการ์ดทำหน้าที่เป็นตัวแทนในการเก็บข้อมูลต่าง ๆ เกี่ยวกับนักศึกษาแต่ละบุคคล เช่น ชื่อ-นามสกุล, รหัสนักศึกษา, คณะ เป็นต้น ทั้งนี้ก็เพื่อให้การดำเนินการเกี่ยวกับระบบสารสนเทศนักศึกษามีความสะดวกและรวดเร็วมากยิ่งขึ้น

1.2 ขอบเขตของปริญญานิพนธ์

สร้างเครื่องอ่าน-เขียนสมาร์ทการ์ดที่ควบคุมการทำงานโดยไมโครคอนโทรลเลอร์จำนวน 1 เครื่อง โดยจะทำงานร่วมกับโปรแกรมคอมพิวเตอร์ที่รองรับการใช้งานสมาร์ทการ์ดเพื่อติดต่อกับระบบสารสนเทศนักศึกษา ซึ่งประกอบด้วยระบบงานทะเบียน งานห้องสมุด และงานอนามัย โดยโปรแกรมคอมพิวเตอร์นั้นเขียนขึ้นด้วยภาษา Visual Basic ที่มีการติดต่อกับระบบฐานข้อมูลที่จัดทำโดยโปรแกรม Microsoft Access

1.3 ประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์

- เข้าใจถึงการทำงานและโครงสร้างของสมาร์ทการ์ด
- เขียนโปรแกรมไมโครคอนโทรลเลอร์ เพื่อควบคุมอุปกรณ์ต่าง ๆ ได้
- ได้เครื่องอ่าน-เขียนสมาร์ทการ์ดที่พร้อมจะนำไปประยุกต์ใช้งานอื่น ๆ ได้
- เขียนโปรแกรมคอมพิวเตอร์เพื่อเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกได้
- เขียนโปรแกรมรองรับการใช้งานสมาร์ทการ์ดในคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 เนื้อหาของปริญญานิพนธ์

สามารถแบ่งได้เป็น 2 ส่วน คือ

1.4.1 ทางด้านฮาร์ดแวร์ (Hard Ware) ประกอบด้วย

- วงจรเครื่องอ่าน-เขียนสมาร์ทการ์ด
- อุปกรณ์เชื่อมต่อพอร์ตอนุกรม

1.4.2 ทางด้านซอฟต์แวร์ (Soft Ware) ประกอบด้วย

- โปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด
- โปรแกรมรับข้อมูลจากเครื่องอ่าน-เขียนสมาร์ทการ์ดเข้ามายังคอมพิวเตอร์โดยผ่านอุปกรณ์เชื่อมต่อพอร์ตอนุกรม
- โปรแกรมคอมพิวเตอร์รองรับการใช้งานสมาร์ทการ์ดเพื่อติดต่อกับระบบสารสนเทศนักศึกษา
- โปรแกรมคอมพิวเตอร์เชื่อมต่อกับระบบฐานข้อมูล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎี

2.1 สมาร์ทการ์ดคืออะไร

สมาร์ทการ์ดคือบัตรพลาสติกที่มีชิปไอซี (Integrated Circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเองโดยวิธีการเข้ารหัสตามมาตรฐาน DES Algorithm (Data Encryption Standard) เพื่อให้ระบบมีระดับความปลอดภัยสูงขึ้น ด้วยคุณสมบัติสำคัญประการหนึ่งที่ทำให้สมาร์ทการ์ดมีความแตกต่างจากบัตรพลาสติกทั่วไปก็คือ ขณะทำรายการ (Transaction) สมาร์ทการ์ดสามารถทำงานได้ด้วยตัวของมันเองโดยไม่ต้องอาศัยติดต่อสื่อสารกับระบบหลัก (Font End) นั้นก็คือสมาร์ทการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสาร กับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก (Off-line) ทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก

2.2 ประวัติความเป็นมาของสมาร์ทการ์ด

สมาร์ทการ์ดปรากฏขึ้นครั้งแรกในประเทศเยอรมัน ในปี 1968 โดยชาวเยอรมัน (Jurgen Dethloff และ Helmut Grotupp) เป็นผู้คิดค้น แต่ผู้ที่ได้มาซึ่งสิทธิบัตรกลับเป็นชาวญี่ปุ่น (Kunitaka Arimura) ในปี 1970 และมีการจดสิทธิบัตรในชื่อของสมาร์ทการ์ดโดยชาวฝรั่งเศส (Roland Moreno) ในปี 1974 ในระยะแรกนั้นสมาร์ทการ์ดยังทำงานได้ไม่สมบูรณ์นัก เพราะสมาร์ทการ์ดรุ่นแรก ๆ ยังมีปัญหาทางเทคนิคเล็ก ๆ น้อย ๆ แม้ว่าสมาร์ทการ์ดจะถือกำเนิดในยุโรป แต่ในระยะแรกสมาร์ทการ์ดกลับไม่ค่อยได้รับความสนใจเท่าที่ควร จนกระทั่งปี 1984 บริษัท French PTT (Postal and Telecommunications Services) ได้นำสมาร์ทการ์ดมาใช้งานเป็นบัตรโทรศัพท์ เพื่อป้องกันการโกงค่าโทรศัพท์ ในครั้งนั้นโครงการเป็นโครงการนำร่องโดยมีการนำบัตรแถบแม่เหล็ก บัตรแถบแสง (Optical Storage) และสมาร์ทการ์ดมาทำการทดลองใช้งานเปรียบเทียบกัน ซึ่งแน่นอนว่าสมาร์ทการ์ดได้พิสูจน์ให้เห็นคุณลักษณะที่เหนือกว่าบัตรชนิดอื่น ทั้งในเรื่องของความทนทาน ความปลอดภัย ความสวยงาม เป็นผลให้สมาร์ทการ์ดในรูปของบัตรโทรศัพท์มีการนำไปใช้ถึง 60 ล้านใบ (เฉพาะประเทศฝรั่งเศส) และตอกย้ำความสำเร็จอีกกว่า 100 ล้านใบจาก 50 ประเทศทั่วโลกในปี 1997 กระนั้นสมาร์ทการ์ดก็ยังเป็นเพียงบัตรโทรศัพท์ การนำสมาร์ทการ์ดมาใช้ทางด้านการเงินการธนาคารกลับเป็นไปอย่างเชื่องช้า เนื่องจากบัตรที่เกี่ยวข้องกับระบบการเงินการธนาคารมีความยุ่งยากมากกว่าบัตรโทรศัพท์

และในปี 1960 เทคโนโลยีการประมวลผลเพื่อเข้ารหัสข้อมูลของฮาร์ดแวร์และซอฟต์แวร์มีความพร้อมมากขึ้น จึงมีการนำมาใช้ในการเข้ารหัสข้อมูลในสมาร์ทการ์ด ซึ่งแต่เดิมนั้นการเข้ารหัสจะมีการใช้งานเฉพาะในหน่วยงานทหาร หรือหน่วยงานราชการลับเท่านั้น ด้วยเหตุนี้ทำให้สมาร์ทการ์ดสามารถทำการเข้า–ถอดรหัสข้อมูลได้ด้วยตัวมันเอง ทำให้การใช้สมาร์ทการ์ดมีความปลอดภัยสูงขึ้นจนสามารถนำมาใช้เป็นบัตรเครดิต หรือบัตรเงินสดได้อย่างสมบูรณ์แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในปี 1984 ธนาคารในฝรั่งเศสได้นำสมาร์ทการ์ดมาใช้เป็นบัตรเครดิตเป็นครั้งแรก ในระยะแรกนั้นต้องประสบกับปัญหามากมาย เกี่ยวกับการเข้ากันได้ของบัตรต่างธนาคาร ซึ่งต้องใช้เวลาถึง 10 ปีที่จะทำให้เข้ากันได้ทั้งหมด เป็นเหตุให้มีการรวมกันของ Europay, VISA และ MASTER เพื่อกำหนดมาตรฐานแก่เครดิตการ์ด ในรูปของสมาร์ทการ์ดให้มีมาตรฐานเดียวกันทุกธนาคารในชื่อของมาตรฐาน EMV (Europay, MASTER, VISA) โดยอ้างอิงกับมาตรฐาน ISO7816 เป็นหลัก ทำให้มีผู้ที่ต้องการพัฒนาแอพพลิเคชั่นเครดิตหรือเดบิตบนสมาร์ทการ์ด ต้องทำตามข้อกำหนดของมาตรฐาน EMV เท่านั้น

2.3 ส่วนประกอบและโครงสร้างของสมาร์ทการ์ด

2.3.1 ตัวบัตรพลาสติก

สมาร์ทการ์ดเป็นชิปไอซีขนาดเล็กที่ถูกสร้างขึ้นเช่นเดียวกับชิ้นส่วนอิเล็กทรอนิกส์อื่น ๆ ที่สร้างจากสารกึ่งตัวนำ นำมาติดลงบนหน้าสัมผัส และทำการฝังลงในเนื้อบัตรพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นตัวบัตรจะใช้พลาสติก 4 ชนิด ได้แก่ PVC (Polyvinyl Choloride), ABS (Acrylonitrile Butadiene Styrene), PC (Polycarbonate) และ PET (Polyethylene Terephthalate) ในประเทศไทยจะใช้บัตรพลาสติก PVC มากเป็นอันดับหนึ่ง ส่วนอันดับสองเป็นบัตรพลาสติกชนิด ABS ซึ่งบัตรพลาสติกชนิด PVC มักนำมาใช้เป็นบัตรเอทีเอ็ม บัตรเครดิต-เดบิต บัตรประจำตัวประชาชน ฯลฯ ส่วนบัตรพลาสติกชนิด ABS ไม่ค่อยพบว่าใช้งานกันมากนักเนื่องจากราคาสูงกว่า และลายที่พิมพ์ลงบนบัตรไม่สวยงามคงทนเท่าบัตรพลาสติกชนิด PVC จะพบก็เพียงบัตรพลาสติกเนื้อผสมโดยใช้พลาสติกชนิด ABS เป็นแกนและฉาบผิวด้วยพลาสติกชนิด PVC แต่ความทนทานของตัวบัตรจะสู้บัตรพลาสติกเนื้อ PVC ล้วนไม่ได้

สำหรับบัตรพลาสติกอีก 2 ชนิดที่เหลือ ยังไม่พบว่ามีการใช้งานในประเทศไทย อาจเนื่องมาจากราคาที่สูงเกินไปของวัสดุที่นำมาใช้ทำเป็นตัวบัตร และคุณสมบัติของวัสดุที่ด้อยกว่าพลาสติกชนิด PVC แต่ข้อเสียที่สำคัญของพลาสติกชนิด PVC ก็ไม่ด้อยไปกว่าข้อดีของมัน นั่นก็คือมันไม่สามารถย่อยสลายได้ตามธรรมชาติ ซึ่งเท่ากับเป็นขยะสำหรับสิ่งแวดล้อมเลยทีเดียว

2.3.2 หน้าสัมผัสและชิปสมาร์ทการ์ด (Smart card Module)

สมาร์ทการ์ดโมดูล หรือ หน้าสัมผัสและชิปสมาร์ทการ์ด คือ ส่วนที่แสดงความเป็นตัวตนของสมาร์ทการ์ดที่ชัดเจนที่สุด สมาร์ทการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่ามันคือ สมาร์ทการ์ดที่มีการฝังชิปไว้ในเนื้อบัตร ดังนั้นการที่จะระบุว่าบัตรใบใดเป็นบัตรสมาร์ทการ์ดนั้น ต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก ซึ่งต้องใช้ประสบการณ์ที่เกี่ยวกับสมาร์ทการ์ดพอสมควร แต่ในที่นี้จะขอแนะนำให้เห็นภาพลักษณ์ที่ชัดเจนของสมาร์ทการ์ดเป็นหลัก ซึ่งก็คือส่วนของสมาร์ทการ์ดโมดูลนั่นเอง

ในการผลิตสมาร์ทการ์ดโมดูล ส่วนที่เป็นหน้าสัมผัสของสมาร์ทการ์ดประกอบด้วยโลหะหลายชิ้นประกอบกัน แต่ละส่วนจะถูกยึดด้วยแถบฟิล์มบาง ๆ ทางด้านหลังของหน้าสัมผัสเพื่อให้คงรูปอยู่ได้ แถบฟิล์มตัวนี้จะมีการเจาะช่องเล็ก ๆ สำหรับการเชื่อมต่อสายนำสัญญาณกับชิปสมาร์ทการ์ดกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าสัมผัส หลังจากที่วางชิปสมาร์ทการ์ดลงในตำแหน่งที่ต้องการและเชื่อมต่อสายนำสัญญาณจากชิปสมาร์ทการ์ดเข้ากับหน้าสัมผัสเรียบร้อยแล้ว ขั้นตอนสุดท้ายจะเป็นการผนึกชิปสมาร์ทการ์ดเพื่อป้องกันตัวชิป และสายนำสัญญาณต่าง ๆ จากสิ่งแวดล้อมภายนอก (เป็นการทดสอบขั้นต้น) ส่วนขั้นตอนที่เหลือจะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติก และทดสอบการทำงานของชิปขั้นสุดท้าย

2.4 การประกอบสมาร์ทการ์ดโมดูลลงในบัตรพลาสติก

การประกอบสมาร์ทการ์ดโมดูลลงในบัตรนั้นมีหลายวิธีด้วยกันตามแต่ชนิดของสมาร์ทการ์ดโมดูลและการเตรียมบัตรพลาสติก ซึ่งการเตรียมบัตรพลาสติกจะนำมาใส่สมาร์ทการ์ดโมดูลมีด้วยกัน 2 แบบ คือ บัตรพลาสติกที่ถูกขุดหลุมบนบัตร และบัตรพลาสติกที่เกิดจากการทับซ้อนของชั้นพลาสติกที่เจาะช่องมาแล้ว โดยสมาร์ทการ์ดโมดูลจะใช้บัตรพลาสติกที่มีการเตรียมการมาแล้วดังนี้

2.4.1 การสร้างสมาร์ทการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก (TAB – Tape automated bonding)

สมาร์ทการ์ดชนิดนี้เกิดจากการทับซ้อนของพลาสติกตั้งแต่ 3 ชั้นขึ้นไป โดยแต่ละชั้นจะมีการเจาะช่องตามขนาดหน้าสัมผัสและชิปสมาร์ทการ์ดไว้อยู่แล้ว ส่วนที่เป็นหน้าสัมผัสและชิปจะถูกแทรกอยู่ในชั้นในของพลาสติก เมื่อวางซ้อนกันเรียบร้อยแล้วก็จะทำการอัดแต่ละชั้นด้วยความร้อน เมื่อความร้อนถึงจุดที่ทำให้พลาสติกแต่ละชั้นประสานตัวเอง ก็จะนำมาตัดแต่งขอบบัตรและทำการทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย

2.4.2 การสร้างสมาร์ทการ์ดด้วยวิธีการวางสมาร์ทการ์ดโมดูลลงในเนื้อบัตร (Chip-On-Flex)

สมาร์ทการ์ดโมดูลที่จะนำมาใส่ลงในบัตรพลาสติก ผู้ผลิตจะทำการตัดตามขนาดของหลุมบนบัตรพลาสติกที่ขุดรอไว้แล้วด้วยเครื่องจักร ทำการเชื่อมด้วยกาว และอบด้วยความร้อนเพื่อให้สมาร์ทการ์ดโมดูลติดสนิทกับเนื้อพลาสติก จากนั้นจึงทำการทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย การใส่หน้าสัมผัสและชิปสมาร์ทการ์ดด้วยกาวนี้ เป็นวิธีที่นิยมทำกันมากที่สุด เพราะผู้ผลิตสามารถประหยัดต้นทุนในการผลิตได้มาก เนื่องจากวิธีการนี้เสียค่าใช้จ่ายในการผลิตน้อยที่สุด ไม่จะเป็นเรื่องของแรงงาน ความรวดเร็วในการผลิต และเปอร์เซ็นต์สินค้าชำรุดต่ำ ในคุณภาพของสมาร์ทการ์ดที่ยังพอยอมรับได้

2.4.3 การสร้างสมาร์ทการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวของบัตร (Chip-On-Surface)

สมาร์ทการ์ดโมดูลชนิดติดบนผิวของบัตร ผลิตโดยการใช้แสงเลเซอร์ทำการขุดหลุมบนบัตรพลาสติกขนาดเท่ากับตัวชิปสมาร์ทการ์ด วางชิปสมาร์ทการ์ดลงในตำแหน่งที่กำหนด สร้างหน้าสัมผัสและเชื่อมสายสัญญาณกับชิปสมาร์การ์ดด้วยหมึกนำไฟฟ้า สุดท้ายพิมพ์ทับส่วนที่เป็นชิปและหมึกนำไฟฟ้าส่วนที่เป็นสายสัญญาณด้วยหมึกที่มีคุณสมบัติเป็นฉนวนไฟฟ้าเพื่อป้องกันวงจรภายใน โดยปล่อยส่วนที่เป็นหน้าสัมผัสที่สร้างจากหมึกนำไฟฟ้าไว้เท่านั้น สมาร์ทการ์ดชนิดนี้ไม่ค่อยมีให้เห็นมากนัก เนื่องจากต้องใช้เทคโนโลยีในการผลิตที่สูงกว่าหน้าสัมผัสแบบอื่น ทำให้ต้นทุนการผลิตสูงตามไปด้วย อีกทั้งความทนทานก็ยังน้อยกว่าสมาร์ทการ์ดโมดูลชนิดอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ข้อกำหนดอีเอ็มวี (EMV: Europay, MASTER, VISA)

ข้อกำหนดอีเอ็มวี เป็นข้อกำหนดที่ใช้ระบุข้อกำหนดขั้นต่ำของสมาร์ทการ์ด และเครื่องรับบัตรสมาร์ทการ์ด ซึ่งเป็นการร่วมกันระหว่าง Europay, MASTER และ VISA ในการกำหนดคุณสมบัติของสมาร์ทการ์ด ที่จะนำมาประยุกต์ใช้ในการทำธุรกรรมทางด้านการเงินและการธนาคาร ซึ่งกำลังได้รับการผลักดันให้กลายเป็นมาตรฐานในการทำธุรกรรมบนสมาร์ตการ์ด ประกอบด้วยข้อกำหนด 3 ตัว ได้แก่

1. ข้อกำหนดสำหรับบัตรสมาร์ทการ์ดที่ใช้ในระบบการชำระเงิน

เป็นข้อกำหนดส่วนใหญ่ในข้อกำหนดอีเอ็มวี สำหรับบัตรสมาร์ทการ์ดนี้อ้างอิงกับมาตรฐาน ISO7816 โดยแบ่งออกเป็น 4 ส่วนย่อย ดังนี้

1.1 คุณสมบัติทางกายภาพของบัตร และข้อกำหนดในการแลกเปลี่ยนข้อมูล ประกอบด้วย

- คุณสมบัติเชิงกล เช่น ขนาด และตำแหน่งของชิปสมาร์ทการ์ด
- คุณสมบัติทางไฟฟ้า เช่น แรงดัน กระแสไฟฟ้า ความต้านทาน ความถี่ที่ใช้งาน
- โปรโตคอล (Protocol) ที่ใช้ในการสื่อสาร

1.2 รายละเอียดข้อมูลและชุดคำสั่งที่ใช้ในการสื่อสาร ประกอบด้วย

- โครงสร้างข้อมูลที่ใช้สำหรับการสื่อสารเพื่อธุรกรรมทางการเงิน
- โครงสร้างไฟล์ข้อมูลในสมาร์ทการ์ด
- ชุดคำสั่งสำหรับการสื่อสาร

1.3 ขั้นตอนการประมวลผล เป็นการกำหนดขั้นตอนสำหรับเครื่องรับบัตรสมาร์ทการ์ด ว่าในการทำธุรกรรมต้องประมวลผลอะไรบ้าง รวมถึงการบังคับให้บัตรหลาย ๆ บัตรสามารถใช้งานร่วมกันได้ และกำหนดให้มีฟังก์ชั่นการทำงานภายในบัตรมากพอเพื่อใช้ในการทำธุรกรรม ซึ่งข้อกำหนดนี้ ประกอบด้วย

- โครงสร้างไดเร็กทอรี่ของข้อมูลในสมาร์ทการ์ด
- กระบวนการประมวลผล

1.4 มาตรการรักษาความปลอดภัย ประกอบด้วย

- ป้องกันการแก้ไขข้อมูลที่หวงห้าม
- กุญแจรหัสสำหรับการแจกจ่าย
- การตรวจสอบความถูกต้องของกุญแจรหัส (Key Verification)
- รูปแบบการรักษาความปลอดภัยของสายข้อมูล (Secure Messaging)

2. ข้อกำหนดสำหรับเครื่องรับบัตรสมาร์ทการ์ดที่ใช้ในระบบการชำระเงิน

เป็นข้อกำหนดสำหรับเครื่องรับบัตรสมาร์ทการ์ดที่ระบุถึงส่วนบังคับ ส่วนที่แนะนำ และส่วนที่เป็นทางเลือกให้แก่ผู้ผลิตเครื่องรับบัตรสมาร์ทการ์ด โดยรวมถึงเครื่องเอทีเอ็ม เครื่อง POS อุปกรณ์ประกอบเครื่องคอมพิวเตอร์ส่วนบุคคล เครื่องเก็บเงินอิเล็กทรอนิกส์ เครื่องอ่านบัตรสำหรับผู้ถือบัตร และเครื่องเติมเงิน ซึ่งประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชนิดของเครื่องรับบัตรสมาร์ทการ์ด และความสามารถในการทำงาน
- คุณสมบัติทางกายภาพทั่วไป
- สถาปัตยกรรมทางด้านซอฟต์แวร์
- มาตรการรักษาความปลอดภัยของเครื่องรับบัตรสมาร์ทการ์ด
- ส่วนแสดงผลสำหรับผู้ถือบัตร
- ส่วนแสดงผลสำหรับรับบัตร หรือร้านค้า

3. ข้อกำหนดมาตรฐานสำหรับการใช้งานบัตรสมาร์ทการ์ดในระบบการชำระเงิน เป็นข้อกำหนดที่เจาะจงสำหรับกระบวนการในการใช้บัตรเพื่อชำระเงิน ประกอบด้วย
- โครงสร้างไฟล์ข้อมูลสำหรับจัดการเรื่องของธุรกรรม
- ขั้นตอนการทำธุรกรรมด้วยบัตรสมาร์ทการ์ด
- การประมวลผลสำหรับธุรกรรมที่ผิดปกติ

2.6 รายละเอียดพื้นฐานของสมาร์ทการ์ด

สมาร์ทการ์ดเป็นบัตรพลาสติกขนาดเท่าบัตรเครดิต หรือบัตรเอทีเอ็ม (ATM: Automatic Teller Machine) ที่มีหน่วยเก็บข้อมูล และหน่วยประมวลผลที่เรียกว่า ไมโครชิป ติดอยู่บนบัตร ซึ่งข้อมูลนี้อาจจะอยู่ในรูปของตัวเลขหรือตัวอักษรก็ได้ โดยมีกลไกในการเขียนและการอ่านข้อมูลที่ซับซ้อน ทำให้ยากต่อการปลอมแปลง จึงสามารถนำมาใช้ประโยชน์ในด้านต่าง ๆ มากมาย เช่น ด้านการเงินและการธนาคาร ด้านโทรคมนาคม ด้านงานทะเบียน ด้านการศึกษา ด้านการรักษาความปลอดภัย เป็นต้น

รูปที่ 2.1 แสดงรูปบัตรสมาร์ทการ์ด

สมาร์ทการ์ดมีพื้นฐานมาจากระบบไมโครโปรเซสเซอร์ ซึ่งมีแนวคิดเริ่มแรกจากการนำชิปหน่วยความจำ (EEPROM) มาฝังลงในบัตรพลาสติก โดยมีหน้าสัมผัสเป็นขาเชื่อมต่อกับระบบภายนอก ในการเชื่อมต่อต้องมีการป้อนกระแสไฟฟ้าให้ชิปหน่วยความจำสามารถทำงานได้ การสั่งงานเพื่ออ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเขียนข้อมูลจากชิปหน่วยความจำสมาร์ทการ์ด ก็ทำได้โดยการเชื่อมต่อสัญญาณผ่านหน้าสัมผัสที่กำหนดไว้แล้ว ในการเชื่อมต่อขาสัญญาณของชิปหน่วยความจำแบบธรรมดา อาจไม่เหมาะสมนักสำหรับบัตรพลาสติกขนาดเล็ก เนื่องจากจำนวนขาสัญญาณของหน่วยความจำ (Bus) มีจำนวนไม่น้อยทีเดียว ยิ่งหน่วยความจำที่มีความจุสูง ๆ ยิ่งต้องใช้สัญญาณอ้างอิงตำแหน่งของข้อมูล (Address Bus) มากขึ้น จึงมีการนำเอาระบบสื่อสารแบบซิงเกิลบัสมาใช้ในการรับส่งข้อมูล โดยในการนำเอาระบบสื่อสารแบบอนุกรมมาใช้ จำเป็นต้องมีการป้อนสัญญาณนาฬิกาเพื่อกำกับจังหวะการรับ-ส่งข้อมูลแต่ละบิต ทำให้ต้องมีหน้าสัมผัสสำหรับสัญญาณนาฬิกาบนชิปสมาร์ทการ์ดเพิ่มขึ้นมา แต่ก็นับว่าทำให้ขาเชื่อมต่อลดลงไปไม่น้อยทีเดียว ด้วยเหตุนี้สมาร์ทการ์ดชนิดหน่วยความจำจึงเป็นสมาร์ทการ์ดชนิดแรกที่ถูกสร้างขึ้น

การนำเอาชิปหน่วยความจำมาใส่ในบัตรพลาสติก ทำให้เกิดข้อดีเหนือบัตรแถบแม่เหล็กด้วยความจุข้อมูลที่มากกว่า ไม่มีผลต่อสนามแม่เหล็กไฟฟ้า และรอยขีดข่วน ทำให้สมาร์ทการ์ดโดดเด่นกว่าบัตรแถบแม่เหล็กอย่างเทียบกันไม่ได้ แต่ข้อเสียประการหนึ่งของการใช้หน่วยความจำเพียงอย่างเดียวคือสามารถทำการอ่านและเขียนข้อมูลได้อย่างอิสระเช่นเดียวกับบัตรแถบแม่เหล็ก จึงถือได้ว่าความปลอดภัยของข้อมูลเกือบเป็นศูนย์ นั่นก็คือ ข้อมูลภายในสมาร์ทการ์ดชนิดนี้ไม่เป็นความลับ ด้วยเหตุนี้จึงมีการเพิ่มวงจรสำหรับป้องกันลงไปอีก เพื่อให้ผู้ออกบัตร (Card Issue) สามารถกำหนดสิทธิในการเข้าถึงข้อมูลแต่ละไบต์ด้วยวงจรฟิวส์แมทริกซ์ธรรมดา ๆ ที่เมื่อกำหนดเงื่อนไขไปแล้วจะไม่สามารถแก้ไขได้อีก ต่อมาเมื่อเทคโนโลยีทางด้านเซมิคอนดักเตอร์สูงขึ้น จึงมีการออกแบบวงจรที่สามารถกำหนดเป็นกุญแจรหัส (PIN) สำหรับเข้าถึงข้อมูลในบัตร ซึ่งต้องทำการแสดงกุญแจรหัสทุกครั้งที่บัตรเริ่มทำงาน เพื่อป้องกันการเจาะระบบอีกชั้นหนึ่ง อีกทั้งกุญแจรหัสก็ยังสามารถเปลี่ยนแปลงได้อีกด้วย

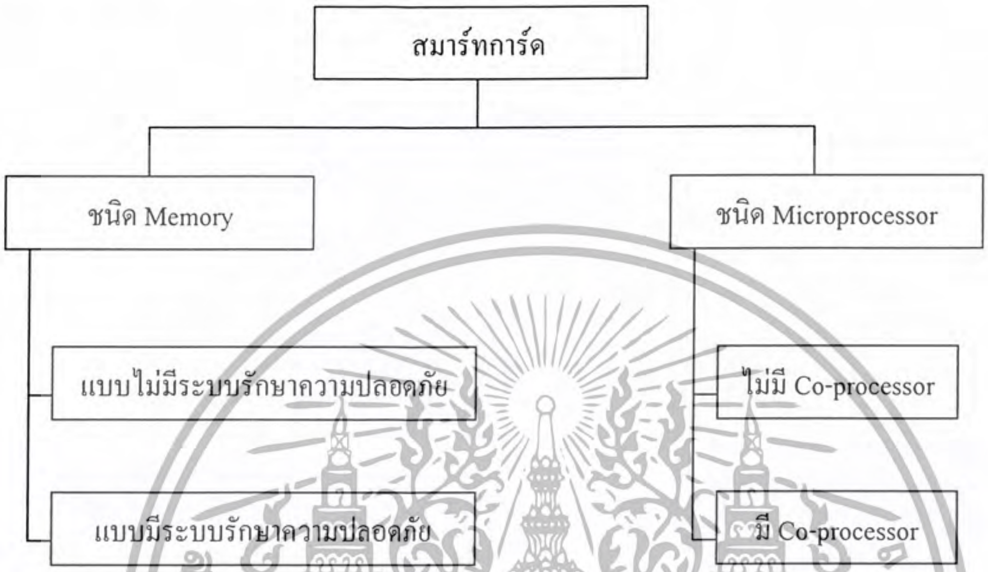
ต่อมามีการนำเอาไมโครโปรเซสเซอร์ (ที่จริงแล้วเป็นไมโครคอนโทรลเลอร์ แต่ในที่นี้จะขอเรียกว่าไมโครโปรเซสเซอร์เป็นหลัก) มาใส่ลงในสมาร์ทการ์ด ทำให้เกิดเป็นสมาร์ทการ์ดชนิดใหม่ที่มีความซับซ้อนยิ่งขึ้น การเข้าถึงข้อมูลไม่สามารถทำได้โดยตรงเหมือนอย่างสมาร์ทการ์ดชนิดหน่วยความจำ การใช้งานสมาร์ทการ์ดชนิดนี้ต้องเขียนขึ้นเป็นชุดคำสั่ง และส่งให้กับชิปไมโครโปรเซสเซอร์ทำงานแทน การที่ใส่ชิปไมโครโปรเซสเซอร์ลงไปในสมาร์ทการ์ด ทำให้ต้องมีการเพิ่มส่วนของหน่วยความจำโปรแกรม (OS-Operating System) สำหรับไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์สามารถทำการประมวลผลคำสั่งต่าง ๆ และสามารถโปรแกรมการเข้าถึงข้อมูล ทำให้ช่องโหว่ที่สำคัญของสมาร์ทการ์ดได้รับการแก้ไขจนเกือบสมบูรณ์แบบ

นอกจากสมาร์ทการ์ดทั้งสองชนิดที่ได้กล่าวมา ยังมีสมาร์ทการ์ดอีกชนิดหนึ่งที่ไม่ใช้หน้าสัมผัส (Contactless) ในการรับส่งสัญญาณ โดยอาศัยเทคโนโลยีคลื่นวิทยุในการติดต่อสื่อสาร สมาร์ทการ์ดชนิดนี้อาศัยการแปลงคลื่นวิทยุส่วนหนึ่งมาใช้เป็นกระแสไฟฟ้าสำหรับป้อนให้ชิป อีกส่วนหนึ่งมาดีเทกเอาข้อมูลคำสั่งให้ชิป สมาร์ทการ์ดชนิดนี้ได้รับความนิยมค่อนข้างมาก เพราะความน่าตื่นตาตื่นใจ และล้ำสมัยของมัน แต่กระนั้นราคาของมันก็ย่อมสูงตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 ชนิดของสมาร์ทการ์ด

การแบ่งชนิดของสมาร์ทการ์ดในปัจจุบันอาจทำได้ยากสักหน่อย เนื่องจากมีการใส่เทคโนโลยีใหม่ ๆ ลงสมาร์ทการ์ดตลอดเวลา ถ้าจะแบ่งตามชนิดของหน่วยความจำภายในอาจไม่ชัดเจนนัก ยิ่งแบ่งตามลักษณะการเชื่อมต่อก็คงไม่ครอบคลุมสมาร์ทการ์ดทั้งหมด ดังนั้นจึงขอแสดงการแบ่งชนิดของสมาร์ทการ์ดให้เข้าใจได้ง่าย ดังรูปที่ 2.2



รูปที่ 2.2 แสดงการแบ่งสมาร์ทการ์ดตามชนิดของหน่วยความจำ

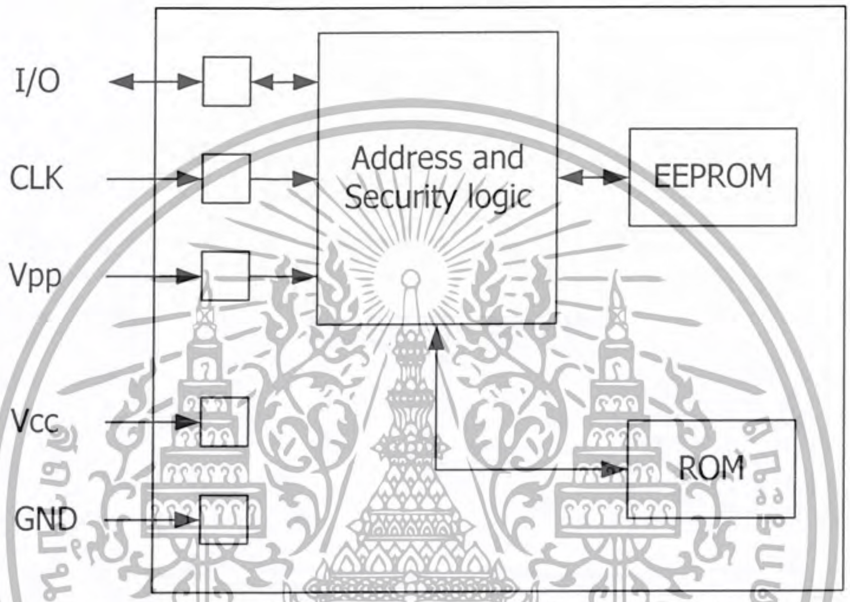
จากรูปที่ 2.2 จะเห็นได้ว่าเราสามารถแบ่งสมาร์ทการ์ดจากโครงสร้างภายในได้ 2 ชนิดก็คือ สมาร์ทการ์ดชนิดหน่วยความจำ (Memory Card) และ สมาร์ทการ์ดชนิดไมโครโปรเซสเซอร์ (Processor Card) ซึ่งชิปทั้งสองแบบจะมีหน้าสัมผัสเหมือนกัน แต่สัญญาณที่ต้องป้อนให้แก่หน้าสัมผัสบางหน้าสัมผัส จะไม่มีการใช้งานในสมาร์ตการ์ดต่างชนิดกัน เช่น แรงดันไฟฟ้าสำหรับการเขียนข้อมูลลงในชิป (Vpp) จะมีใช้ในสมาร์ทการ์ดชนิดหน่วยความจำเท่านั้น, สัญญาณนาฬิกาสำหรับป้อนให้ชิปทำงาน (CLK) ต้องป้อนให้กับชิปเหมือนกัน สำหรับสัญญาณนาฬิกา (CLK) ที่ป้อนให้ชิปสมาร์ทการ์ดเป็นสัญญาณนาฬิกาภายนอกที่ป้อนให้ชิปทำงานได้ เพราะภายในชิปสมาร์ตการ์ดไม่มีวงจรสำหรับสร้างสัญญาณนาฬิกา แต่หน้าสัมผัส I/O จะมีการรับ-ส่งข้อมูลที่แตกต่างกันในเรื่องของความถี่ และวิธีการควบคุมจังหวะการรับ-ส่งของข้อมูลแต่ละบิต

ในการแบ่งสมาร์ทการ์ดออกเป็น 2 ชนิด ตามชนิดของวงจรภายในดังที่กล่าวมา อาจแบ่งได้อีกลักษณะคือ แบ่งตามความถี่ในการรับ-ส่งข้อมูลผ่านหน้าสัมผัส I/O ของสมาร์ทการ์ด ดังที่กล่าวไปแล้ว ซึ่งสามารถแบ่งได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1 การ์ดหน่วยความจำ (Memory card)

สมาร์ทการ์ดชนิดหน่วยความจำ (Memory) หรืออีกชื่อหนึ่งคือ Synchronous card เนื่องจากสมาร์ทการ์ดชนิดนี้มีการรับ-ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิป (ข้อมูลแต่ละบิตที่ส่งให้แก่ชิปต้องสัมพันธ์กับสัญญาณนาฬิกา) สมาร์ทการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วย ส่วนวงจรสำหรับการติดต่อสื่อสารกับภายนอก หน่วยความจำข้อมูล และหน่วยความจำสำหรับเก็บชุดคำสั่งของสมาร์ทการ์ด ดังรูปที่ 2.3



รูปที่ 2.3 แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดหน่วยความจำ

สมาร์ทการ์ดที่เป็นพื้นฐานของสมาร์ทการ์ดในปัจจุบัน ก็คือสมาร์ทการ์ดชนิด Free Access Memory สมาร์ทการ์ดชนิดนี้เปิดโอกาสให้อ่านหรือเขียนข้อมูลในแอดเดรสใด ๆ ก็ได้ตามชื่อของสมาร์ทการ์ดชนิดนี้ ไม่มีการป้องกันข้อมูลใด ๆ ภายในสมาร์ทการ์ดชนิดนี้ ซึ่งแน่นอนว่าเป็นสมาร์ทการ์ดที่มีความปลอดภัยต่ำที่สุด ถึงกระนั้นการอ่านข้อมูลก็ไม่ใช่เรื่องง่ายนักเมื่อมีการออกแบบหน่วยความจำข้อมูลให้มีการสลับตำแหน่งของบิตข้อมูล โดยมีวงจรควบคุมการสลับตำแหน่งของบิตเป็นส่วนป้องกันข้อมูลอีกต่อหนึ่ง ดังนั้นการอ่านข้อมูลออกแบบธรรมดาจะไม่ได้ข้อมูลที่ถูกต้องหากไม่ติดต่อกับวงจรควบคุมการสลับตำแหน่งของบิตโดยตรง

นอกจากนี้สมาร์ทการ์ดชนิดหน่วยความจำแบบธรรมดา ยังมีการใส่วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลลงไปด้วย ทำให้สามารถกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลได้ทุกไบต์ โดยสมาร์ทการ์ดที่มีวงจรป้องกันการอ่าน-เขียนชนิดนี้ถูกเรียกว่า PIN Protect Memory เนื่องจากการเข้าถึงข้อมูลจะต้องแสดงรหัสผ่านให้บัตรทราบก่อนจึงจะสามารถเข้าถึงข้อมูลได้ วงจรกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลจะมีบิตพิเศษที่มีชื่อว่า Bit Protect ซึ่งเป็นบิตข้อมูลที่ฝากไว้กับข้อมูลให้เป็นบิตที่ 9 แต่ไม่สามารถแก้ไขได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งเขียนข้อมูลธรรมดา เพราะ Bit Protect ไม่ได้เป็นส่วนหนึ่งของข้อมูลจริง ๆ ในการแก้ไข Bit Protect นี้จะสามารถทำการเปลี่ยนแปลงได้เพียงครั้งเดียวด้วยคำสั่งเฉพาะเท่านั้น เช่น หากต้องการบังคับไม่ให้ข้อมูลไบต์ใดไม่สามารถแก้ไขได้ก็ให้ทำการเคลียร์บิตที่ 9 ของข้อมูลไบต์นั้น ๆ แต่สำหรับรหัสผ่านในการเข้าถึงข้อมูลสามารถเปลี่ยนแปลงได้ แต่ต้องแสดงรหัสผ่านชุดเก่าให้บัตรได้ทราบเสียก่อนจึงจะสามารถเปลี่ยนแปลงรหัสผ่านได้

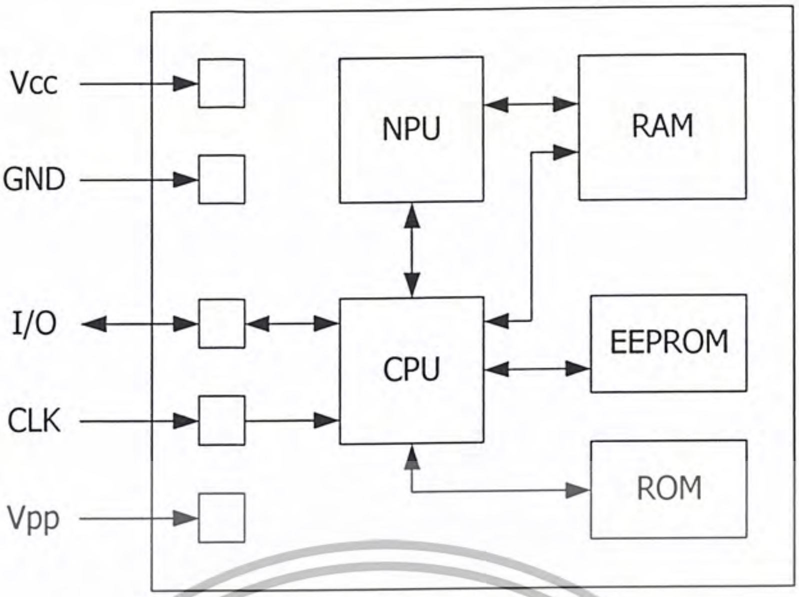
สมาร์ทการ์ดอีกชนิดหนึ่งที่มีใช้เป็นบัตรโทรศัพท์ในประเทศไทยนั้นคือ การ์ดหน่วยความจำชนิด Token ภายในสมาร์ทการ์ดชนิดนี้จะมีการเก็บข้อมูลในลักษณะจำนวนนับ (Counter) ซึ่งจำนวนนับนี้จะเป็นตัวเลขแทนมูลค่าของเงินที่ระบุบนบัตร การนับเลขเป็นการนับถอยหลังเพื่อเป็นการนับมูลค่าที่คงเหลือในบัตร หมายความว่าหากใช้บัตรในการโทรศัพท์ไปเรื่อย ๆ มูลค่าในบัตรก็จะถูกลดลงตามไปด้วยเช่นกัน ในการเข้าถึงข้อมูลของสมาร์ทการ์ดชนิดนี้ต้องมีการแสดงรหัสผ่านให้บัตรทราบเหมือนกับการ์ดหน่วยความจำ ชนิด PIN Protect แต่ไม่มี Bit Protect เท่านั้นเอง

สมาร์ทการ์ดชนิดหน่วยความจำเป็นสมาร์ทการ์ดพื้นฐานของสมาร์ทการ์ดรุ่นใหม่ ๆ ในปัจจุบันด้วยโครงสร้างและการทำงานที่ง่ายต่อการทำความเข้าใจ ราคาถูก สามารถเก็บข้อมูลได้จำนวนมาก และความเร็วในการทำงานของชิปไม่สูงนัก จึงทำให้สมาร์ทการ์ดชนิดนี้เหมาะที่จะนำไปประยุกต์ใช้กับงานที่ข้อมูลไม่ค่อยสำคัญมากนัก เช่น บัตรลงเวลาทำงาน บัตรผ่านประตู บัตรโทรศัพท์ ฯลฯ ปัจจุบันสมาร์ทการ์ดชนิดหน่วยความจำ มีขนาดหน่วยความจำสูงสุดถึง 64 กิโลไบต์ และอีกไม่นานนักเราจะได้เห็นสมาร์ทการ์ดที่มีขนาดหน่วยความจำข้อมูลถึง 128 ไบต์

2.7.2 การ์ดชนิดโปรเซสเซอร์ (Processor card)

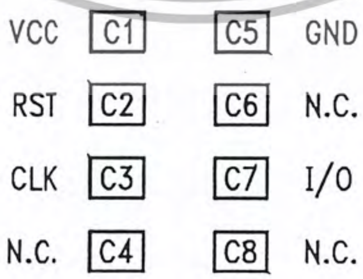
สมาร์ทการ์ดชนิดโปรเซสเซอร์ หรือเรียกอีกชื่อหนึ่งว่า Asynchronous card เป็นสมาร์ทการ์ดที่ได้รับการปรับปรุงจากสมาร์ทการ์ดชนิดหน่วยความจำด้วยการใส่เทคโนโลยีไมโครโปรเซสเซอร์เข้าไปในชิป เพื่อให้ชิปสามารถประมวลผลข้อมูล และเพิ่มความปลอดภัยให้แก่ข้อมูลได้สูงขึ้น การที่ไมโครโปรดเซสเซอร์ลงในชิปทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำสำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผลข้อมูลนอกจากนี้ยังมีการใส่ชิปประมวลผลทางคณิตศาสตร์ลงในชิปสมาร์ทการ์ดเพื่อช่วยให้การประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้า-ถอดรหัส ทำให้สมาร์ทการ์ดชนิดโปรเซสเซอร์มีความเร็วในการทำงานสูงกว่าสมาร์ทการ์ดชนิดหน่วยความจำหลายเท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดโปรเซสเซอร์

ในการรับส่งข้อมูลให้กับสมาร์ทการ์ดชนิดนี้ จะใช้หน้าสัมผัสเดียวกับสมาร์ทการ์ดชนิดหน่วยความจำ โดยสัญญาณนาฬิการที่ป้อน จะถูกใช้เป็นสัญญาณนาฬิกาให้แก่โปรเซสเซอร์ภายในสมาร์ทการ์ด ข้อมูลที่รับส่งจึงไม่จำเป็นต้องสัมพันธ์กับสัญญาณนาฬิกาที่ป้อนให้แก่ชิป เพียงกำหนดอัตราการรับ-ส่งข้อมูลเป็น 9600 บิต/วินาที ก็จะสามารถติดต่อกับโปรเซสเซอร์ของชิปได้แล้ว แต่การเข้าถึงข้อมูลจะไม่สามารถทำได้เหมือนอย่างในสมาร์ทการ์ดชนิดหน่วยความจำ การเข้าถึงข้อมูลต้องกระทำผ่านโปรเซสเซอร์ของสมาร์ทการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตาม เพราะหน่วยความจำจะอยู่ภายในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว ข้อดีอย่างหนึ่งที่ไม่สามารถติดต่อกับหน่วยความจำในชิปได้โดยตรงก็คือ การลอบเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตแทบเป็นไปไม่ได้ ยกเว้นมีความบกพร่องในการกำหนดเงื่อนไขในการเข้าถึงข้อมูลที่เป็นความลับ



รูปที่ 2.5 แสดงขาต่าง ๆ บนหน้าสัมผัสของบัตรสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าสัมผัส	ชื่อขา	การใช้งาน
C1	Vcc	แหล่งจ่ายไฟ
C2	RST	รีเซต
C3	CLK	สัญญาณนาฬิกา
C5	GND	กราวนด์
C7	I/O	รับ-ส่งข้อมูล
C4, C6, C8	N.C.	ไม่ใช้งาน

ตารางที่ 2.1 แสดงหน้าที่การทำงานของขาต่าง ๆ ของบัตรสมาร์ทการ์ด

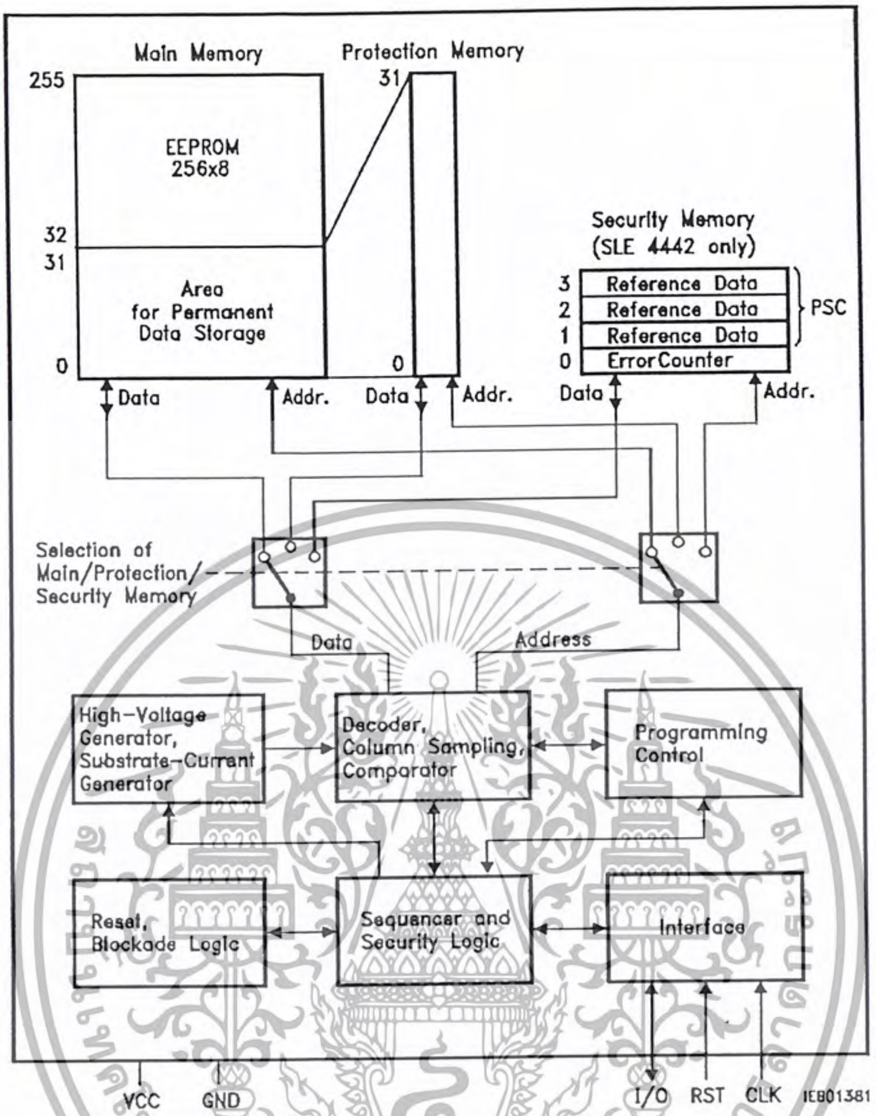
2.8 การ์ดที่มีระบบป้องกันข้อมูล

การ์ดที่มีระบบป้องกันความปลอดภัยข้อมูล คือ สมาร์ทการ์ดที่การอ่านข้อมูลสามารถทำได้อย่างอิสระ แต่การเขียนข้อมูลจะไม่สามารถทำได้หากไม่มีรหัสผ่านที่ถูกต้อง วิธีการในลักษณะนี้ช่วยให้ข้อมูลภายในสมาร์ทการ์ดได้รับการปกป้องและมีความน่าเชื่อถือ รูปแบบการสื่อสารข้อมูลของสมาร์ทการ์ดชนิดนี้เป็นการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) ตามมาตรฐาน ISO7816 ซึ่งรูปแบบคำสั่งจะแตกต่างกันไปในผู้ผลิตแต่ละราย โดยในโครงงานนี้ได้เลือกใช้สมาร์ทการ์ดเบอร์ SLE4442 เนื่องจากเป็นการ์ดที่มีคุณสมบัติในการรักษาความปลอดภัยข้อมูลอย่างครบถ้วนและสามารถหามาใช้งานได้ง่ายในบ้านเรา

2.8.1 คุณสมบัติโดยทั่วไปของสมาร์ทการ์ดเบอร์ **SLE4442**

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



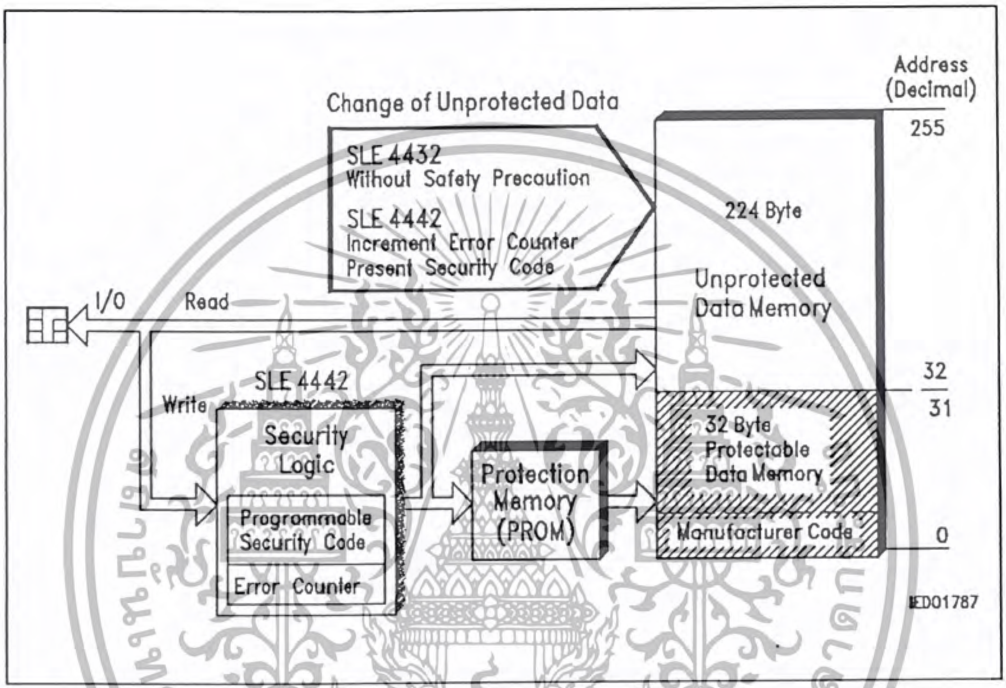
รูปที่ 2.6 แสดงบล็อกไดอะแกรมแสดงโครงสร้างภายในของสมาร์ทการ์ดเบอร์ SLE4442

จากรูปที่ 2.6 จะเห็นได้ว่าหน่วยความจำขนาด 256 ไบต์ ที่อยู่ภายในสมาร์ทการ์ดเบอร์ SLE4442 จะถูกแบ่งออกเป็น 2 ส่วนด้วยกัน ได้แก่ ข้อมูลในช่วง 32 ไบต์แรกซึ่งเป็นพื้นที่ที่มีระบบป้องกันการเขียนข้อมูลทับ และหน่วยความจำส่วนถัดมาซึ่งเป็นอีอีพรอม (EEPROM) ที่สามารถทั้งเขียนและอ่านได้ กลไกในการปกป้องข้อมูลของสมาร์ทการ์ดเบอร์ SLE4442 มาจากส่วนที่เป็นหน่วยความจำปลอดภัย (Security Memory) ที่ได้รับการปกป้องโดยข้อมูลสำคัญ 2 ส่วน คือ

- ข้อมูลอ้างอิง (Reference Data หรือ PSC) เป็นข้อมูลขนาด 3 ไบต์ ที่เก็บค่าของรหัสผ่านสำหรับการเข้าไปแก้ไขข้อมูลในหน่วยความจำเอาไว้ (รหัส PSC ไม่สามารถถูกอ่านออกมาได้) รหัส PSC จะถูกกำหนดเป็นค่าหนึ่งมาโดยผู้ผลิตก่อนซึ่งสามารถจะมาปรับเปลี่ยนเองได้ในภายหลังเมื่อใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไบต์แสดงความผิดพลาด (Error Counter Byte) เป็นข้อมูลที่บอกถึงจำนวนครั้งที่ป้อนรหัส PSC ผิด ซึ่งถูกกำหนดเอาไว้ตายตัวว่าจะผิดได้ไม่เกิน 3 ครั้ง หากเกินกว่านั้นการ์ดจะล๊อคตัวเองอย่างถาวรทันที และไม่มีทางปลดล๊อคได้ แม้ว่าจะป้อนรหัส PSC ที่ถูกต้องไปแล้วก็ตาม การเขียนข้อมูลยังหน่วยความจำก็จะไม่สามารถทำได้อีกต่อไป แต่ยังคงอ่านข้อมูลออกมาได้ตามปกติ การป้อนรหัส PSC ผิดแต่ละครั้ง Error Counter จะถูกลดลง 1 ค่าทันที ถ้าหากค่า Error Counter ถูกลดลงจนมีค่าเป็น 0 เมื่อไรก็แสดงว่าการ์ดได้ถูกล๊อคไปเรียบร้อยแล้ว (ในกรณีที่ป้อนรหัสถูกในครั้งที่ 3 ค่าของ Error Counter จะถูกรีเซตกลับไปเป็น 3 ครั้งเหมือนอย่างตอนแรกเริ่ม



รูปที่ 2.7 แสดงไดอะแกรมแสดงภาพรวมของการ์ดที่มีระบบป้องกันข้อมูล

จากรูปที่ 2.7 จะเห็นได้ว่าการอ่านข้อมูลจากหน่วยความจำนั้น เราสามารถจะอ่านข้อมูลออกมาได้โดยไม่ต้องผ่านขั้นตอนของการป้อนรหัส PSC แต่สำหรับการเขียนข้อมูลแล้วเราจะต้องป้อนรหัส PSC ที่ถูกต้องเสียก่อน เพื่อเปิดลอจิกในการเขียนข้อมูลลงยังหน่วยความจำ นอกจากนั้นก็จะเห็นได้ว่าข้อมูล 4 ไบต์แรก เป็นข้อมูลของผู้ผลิตหรือ Manufacturer Code โดยพื้นที่ส่วนนี้ใช้เก็บข้อมูลของ ATR โดยความหมายของข้อมูลที่อยู่ในพื้นที่ส่วนนี้แต่ละไบต์จะถูกกำหนดโดยผู้ผลิตการ์ดแต่ละราย

2.8.2 รูปแบบการสื่อสารข้อมูลของสมาร์ทการ์ดเบอร์ SLE4442

รูปแบบการสื่อสารข้อมูลของสมาร์ทการ์ดเบอร์ SLE4442 เป็นการรับส่งข้อมูลระหว่างเครื่องอ่านและสมาร์ทการ์ดแบบ 2 ทิศทาง (ข้อมูลบนสาย I/O จะถูกอ่านค่าที่ขอบขาลงของสัญญาณนาฬิกา) โดยรูปแบบการสื่อสารนี้ประกอบด้วย 4 โหมดการทำงาน ได้แก่

- การรีเซตและการตอบรับการรีเซตด้วย ATR (Answer To Reset)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

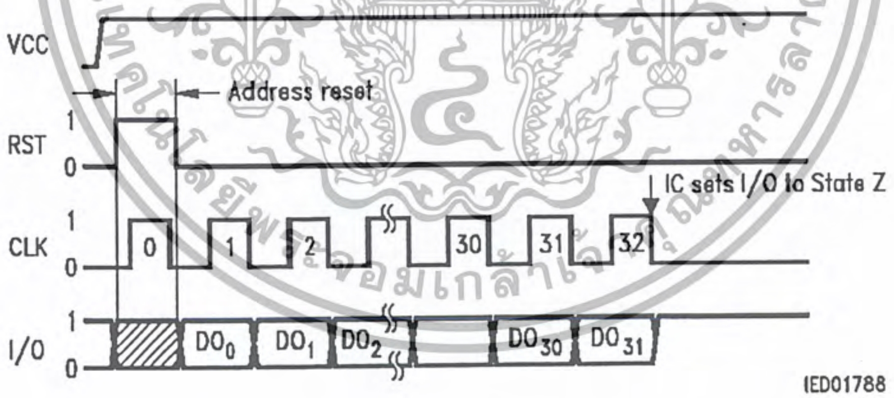
- โหมดการส่งคำสั่ง (Command Mode)
- โหมดการอ่านข้อมูล (Outgoing Data Mode)
- โหมดการดำเนินการ (Processing Mode)

2.8.2.1 การรีเซตและการตอบรับการรีเซตด้วย ATR (Answer To Reset)

เมื่อรีเซตการทำงานของการ์ดจะทำให้การ์ดมีการตอบรับการรีเซตด้วยข้อมูล ATR สำหรับข้อมูล ATR ที่ตอบกลับมาจากสมาร์ทการ์ดเบอร์ SLE4442 จะประกอบด้วยข้อมูล 4 ไบต์ การอ่านข้อมูลที่ว่านี้สามารถทำได้โดยอ้างอิงจากสัญญาณในรูปที่ 2.8 โดยหลังจากที่ขา RST เป็นลอจิกต่ำ เมื่อมีสัญญาณนาฬิกาลูกต่อไปเข้ามา จะทำให้เกิดสัญญาณเอาท์พุตของสมาร์ทการ์ดขึ้นที่ขา I/O ซึ่งก็คือ สัญญาณตอบรับการรีเซตนั้นเอง หลังจากที่ครบ 4 ไบต์แล้ว ที่ขา I/O จะเปลี่ยนเป็นลอจิกสูง เพื่อเป็นการบอกถึงการสิ้นสุดการรีเซต

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

ตารางที่ 2.2 แสดงลักษณะของข้อมูลที่ได้จากการตอบรับการรีเซต



รูปที่ 2.8 แสดงรูปสัญญาณของการรีเซตและการตอบรับการรีเซตด้วย ATR

2.8.2.2 โหมดการส่งคำสั่ง (Command Mode)

การส่งคำสั่งไปยังสมาร์ทการ์ดหรือการทำงานในโหมดการส่งคำสั่ง (Command Mode) ก็คือกระบวนการต่อเนื่องหลังจากการ์ดถูกรีเซตไปเรียบร้อยแล้ว โดยการ์ดจะรอรับคำสั่งที่ส่งมาจากเครื่องอ่านซึ่งมีรูปแบบเป็นข้อมูลมีความยาว 3 ไบต์ โครงสร้างของข้อมูลดังกล่าวประกอบด้วยคำสั่ง (Command),

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

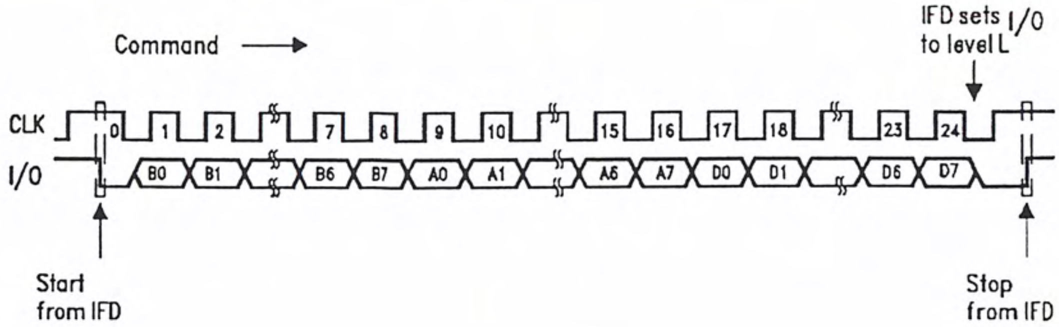
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แอดเดรส (Address) และ ข้อมูล (Data) โดยคำสั่งทั้งหมดที่สมาร์ทการ์ดเบอร์ SLE4442 รองรับถูกแสดงดังตารางที่ 2.3 ส่วนรูปสัญญาณที่เกิดขึ้นระหว่างการทำงานของโหมดการส่งคำสั่งก็เป็นดังรูปที่ 2.9 จะเห็นได้ว่าการส่งข้อมูลแต่ละครั้งจะต้องมีการส่งสภาวะเริ่มต้นและสภาวะสิ้นสุดกำกับไปกับตัวข้อมูลด้วย โดยสภาวะเริ่มต้นก็คือการเปลี่ยนระดับจากลอจิกค่าสูงเป็นค่าต่ำที่ขา I/O ในขณะที่ระดับลอจิกที่ขา CLK เป็นค่าสูง ส่วนสภาวะสิ้นสุดก็คือการเปลี่ยนระดับจากลอจิกค่าต่ำเป็นสูงที่ขา I/O ในขณะที่ขา CLK เป็นค่าสูง

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
0	0	1	1	0	0	0	1	no effect	no effect	READ SECURITY MEMORY	outgoing data
0	0	1	1	1	0	0	1	address	input data	UPDATE SECURITY MEMORY	processing
0	0	1	1	0	0	1	1	address	input data	COMPARE VERIFICATION DATA	processing

ตารางที่ 2.3 แสดงโครงสร้างและความหมายของชุดคำสั่งที่สมาร์ทการ์ดเบอร์ SLE4442 รองรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
54933



รูปที่ 2.9 แสดงรูปสัญญาณของการส่งคำสั่งไปยังการ์ด

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

ตารางที่ 2.4 แสดงรูปแบบและส่วนประกอบของคำสั่ง

2.8.2.2.1 การอ่านข้อมูลจากหน่วยความจำหลัก (Read Main Memory)

คือ คำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมาจากหน่วยความจำของการ์ด ทั้งจากพื้นที่ส่วนที่ได้รับการป้องกัน (หน่วยความจำ 32 ไบต์แรก) และส่วนที่ไม่ได้รับการป้องกัน (หน่วยความจำ 224 ไบต์หลัง) โดยจะเป็นการอ่านค่าโดยเริ่มต้นจากแอดเดรสที่ส่งไปจนถึงแอดเดรสสุดท้าย (0FFH) ของพื้นที่หน่วยความจำ

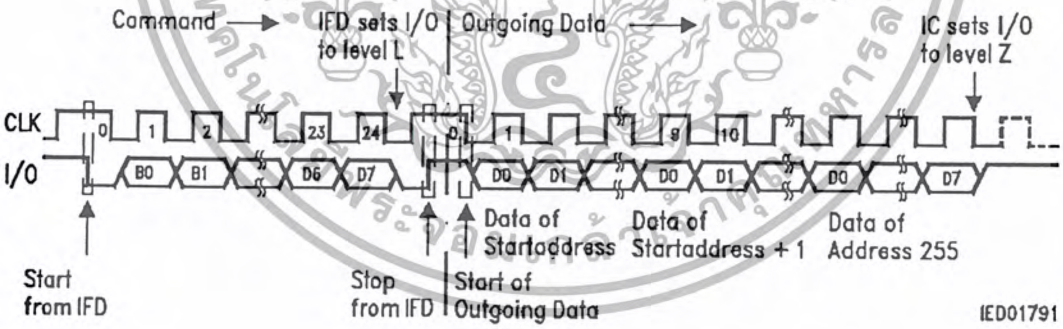
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Command: READ MAIN MEMORY

	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

ตารางที่ 2.5 แสดงลักษณะหน่วยความจำ และรูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำหลัก



รูปที่ 2.10 แสดงรูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

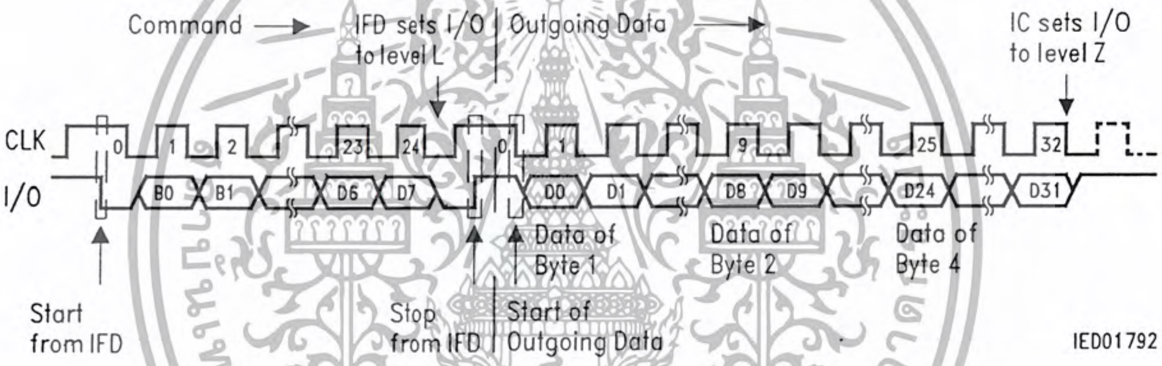
2.8.2.2.2 การอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน (Read Protection Memory)

คือ คำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมาจากหน่วยความจำ 32 ไบต์แรก

Command: READ PROTECTION MEMORY

	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

ตารางที่ 2.6 แสดงรูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน



รูปที่ 2.11 แสดงรูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน

2.8.2.2.3 การเขียนข้อมูลลงในหน่วยความจำหลัก (Update Main Memory)

คือ คำสั่งที่ใช้ในการเขียนข้อมูลยังแอดเดรสใด ๆ ของหน่วยความจำทั้ง 256 ไบต์ ในกรณีที่ใช้คำสั่งนี้ในการเขียนข้อมูลลงยังหน่วยความจำ 32 ไบต์แรก ข้อมูลจะยังคงแก้ไขเปลี่ยนแปลงได้ภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

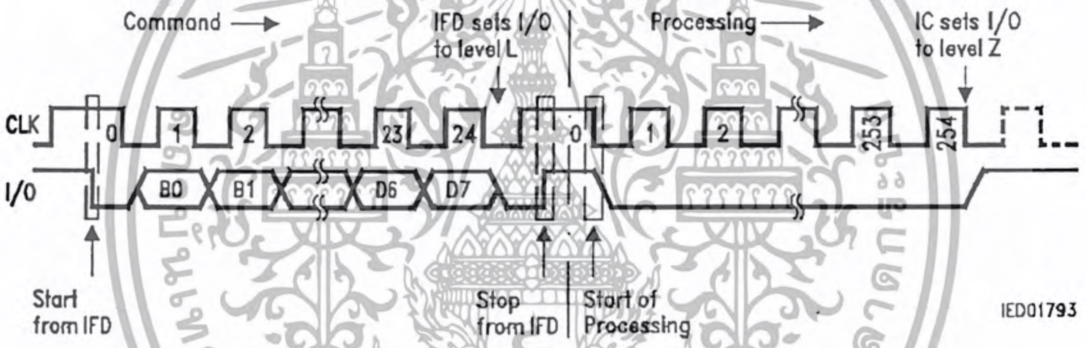
Command: UPDATE MAIN MEMORY

	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

ตารางที่ 2.7 แสดงรูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำหลัก

สำหรับการเขียนข้อมูลจะประกอบด้วย 3 เงื่อนไข คือ

- การลบข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนด ให้เป็น 0FFH แล้วทำการเขียนข้อมูลซ้ำลงยังแอดเดรสเดิม กระบวนการนี้ต้องใช้เวลา 5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 255 ลูก

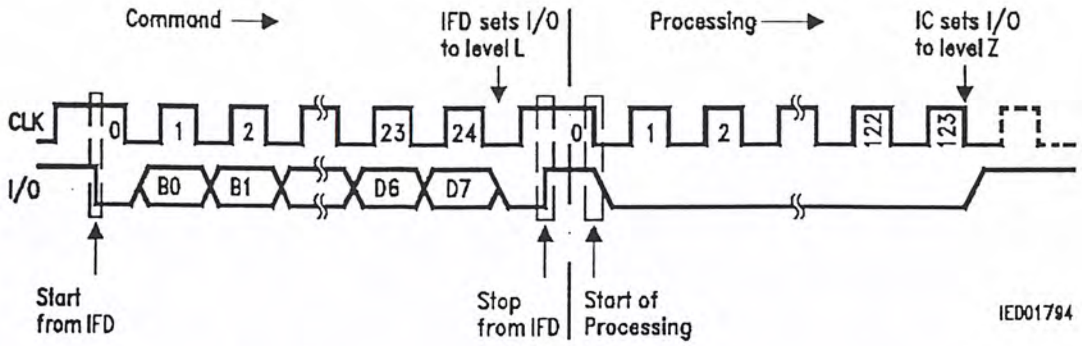


รูปที่ 2.12 แสดงรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก แบบการลบข้อมูลแล้วเขียนข้อมูลซ้ำ

- การเขียนข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนดโดยไม่ต้องลบข้อมูลออก สำหรับกรณีนี้แอดเดรสดังกล่าวจะต้องเป็นที่ว่าง (มีค่าข้อมูลเป็น 0FFH) อยู่ก่อนหน้านี้แล้วเท่านั้น กระบวนการนี้จะใช้เวลา 2.5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 124 ลูก

- การลบข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนด (ให้มีค่าข้อมูลเป็น 0FFH) โดยไม่มีการเขียนข้อมูลต่อ สำหรับกระบวนการนี้ใช้เวลา 2.5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 124 ลูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก แบบการลบหรือเขียนข้อมูล (อย่างใดอย่างหนึ่ง)

2.8.2.2.4 การเขียนข้อมูลลงในหน่วยความจำที่มีการป้องกัน (Write Protection Memory)

คือ การเขียนข้อมูลลงยังแอดเดรสของหน่วยความจำใด ๆ ใน 32 ไบต์แรก คำสั่งนี้มีเงื่อนไขว่า ข้อมูลที่เขียนลงไปจะถูกเขียนลงยังแอดเดรสของหน่วยความจำที่กำหนดอย่างถาวร ไม่สามารถแก้ไขเปลี่ยนแปลงอะไรได้อีก สำหรับรูปสัญญาณของกระบวนการนี้อ้างอิงได้จากรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก (Update Main Memory)

Command: WRITE PROTECTION MEMORY

	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

ตารางที่ 2.8 แสดงรูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำที่มีการป้องกัน

2.8.2.2.5 การอ่านข้อมูลจากหน่วยความจำปลอดภัย (Read Security Memory)

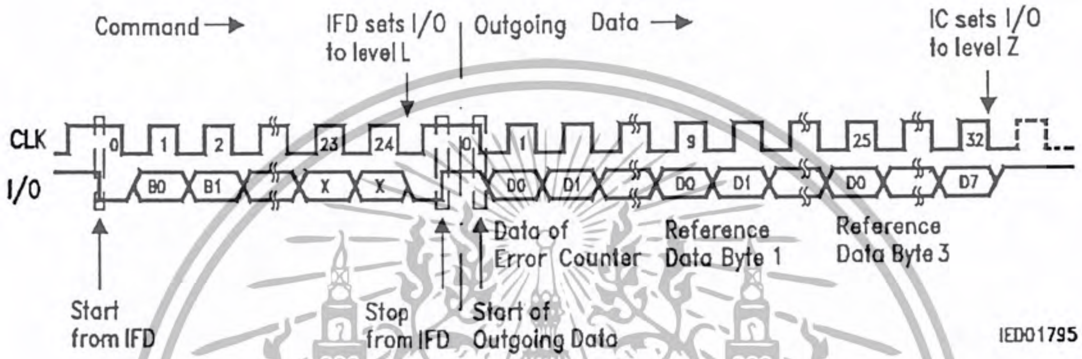
คือ การอ่านค่าของ Error Counter เพื่อตรวจดูว่าการ์ดใบนั้น ๆ ได้ถูกล็อคไปแล้วหรือยัง โดยค่าภายในบิต D2, D1 และ D0 ของ Error Counter จะเป็นส่วนที่บอกถึงสถานะของการ์ดใบนั้น ๆ หากค่าของบิต D2, D1 และ D0 เป็น 0 ทั้งหมด ก็แสดงว่าการ์ดได้ถูกล็อคไปแล้ว ซึ่งจะไม่สามารถแก้ไขอะไรได้และจะไม่สามารถเขียนข้อมูลลงยังการ์ดนั้นได้อีกต่อไป (แต่ว่าการอ่านข้อมูลในการ์ดจะยังคงทำได้ตามปกติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Command: READ SECURITY MEMORY

	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

ตารางที่ 2.9 แสดงรูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำปลอดภัย



รูปที่ 2.14 แสดงรูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำปลอดภัย

2.8.2.2.6 การเขียนข้อมูลลงในหน่วยความจำปลอดภัย (Update Security Memory)

คือ การเข้าไปแก้ไขข้อมูลของรหัส PSC ภายในการ์ด หรืออาจกล่าวได้ว่าเป็นการเข้าไปเปลี่ยนรหัสป้องกันของการ์ดนั้นเอง คำสั่งจะถูกกระทำต่อเมื่อมีการส่งรหัส PSC ที่ถูกต้องไปยังการ์ดเสียก่อน โดยในกรณีที่ป้อนรหัสผิด ค่าของบิต D2, D1 และ D0 ใน Error Counter จะค่อย ๆ ถูกเปลี่ยนจากค่า “1” เป็น “0” ไล่ไปทีละบิตตามจำนวนครั้งที่ป้อนผิด หากทั้งหมดกลายเป็นศูนย์เมื่อไรการ์ดก็จะถูกล๊อคทันที ซึ่งนั่นหมายความว่าโอกาสที่ป้อนรหัสผิดมีเพียง 3 ครั้งเท่านั้น สำหรับรูปสัญญาณของกระบวนการนี้จะเหมือนกับรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก

Command: UPDATE SECURITY MEMORY

	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

ตารางที่ 2.10 แสดงรูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำปลอดภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

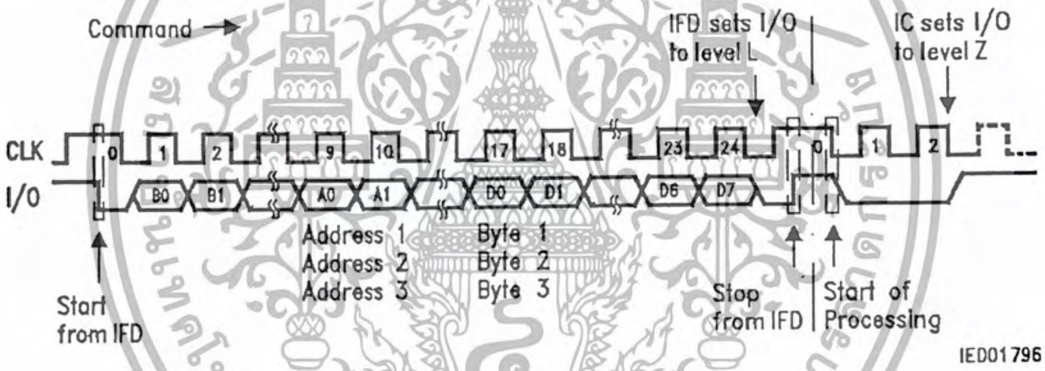
2.8.2.2.7 การเปรียบเทียบและพิสูจน์ข้อมูล (Compare Verification Data)

คือ การสั่งให้การ์ดทำการเปรียบเทียบรหัส PSC กับรหัสผ่านที่เราได้ส่งไปยังการ์ด ในการเปรียบเทียบที่ว่านี้ ข้อมูลที่การ์ดจะส่งกลับมาคือค่าของ Error Counter ที่จะบอกว่ารหัสที่เราป้อนนั้นถูกต้องหรือไม่ และยังเหลือโอกาสพลาดอีกกี่ครั้งเท่านั้น (โดยเราจะไม่สามารถเข้าไปอ่านรหัส PSC ของการ์ดออกมาได้)

Command: COMPARE VERIFICATION DATA

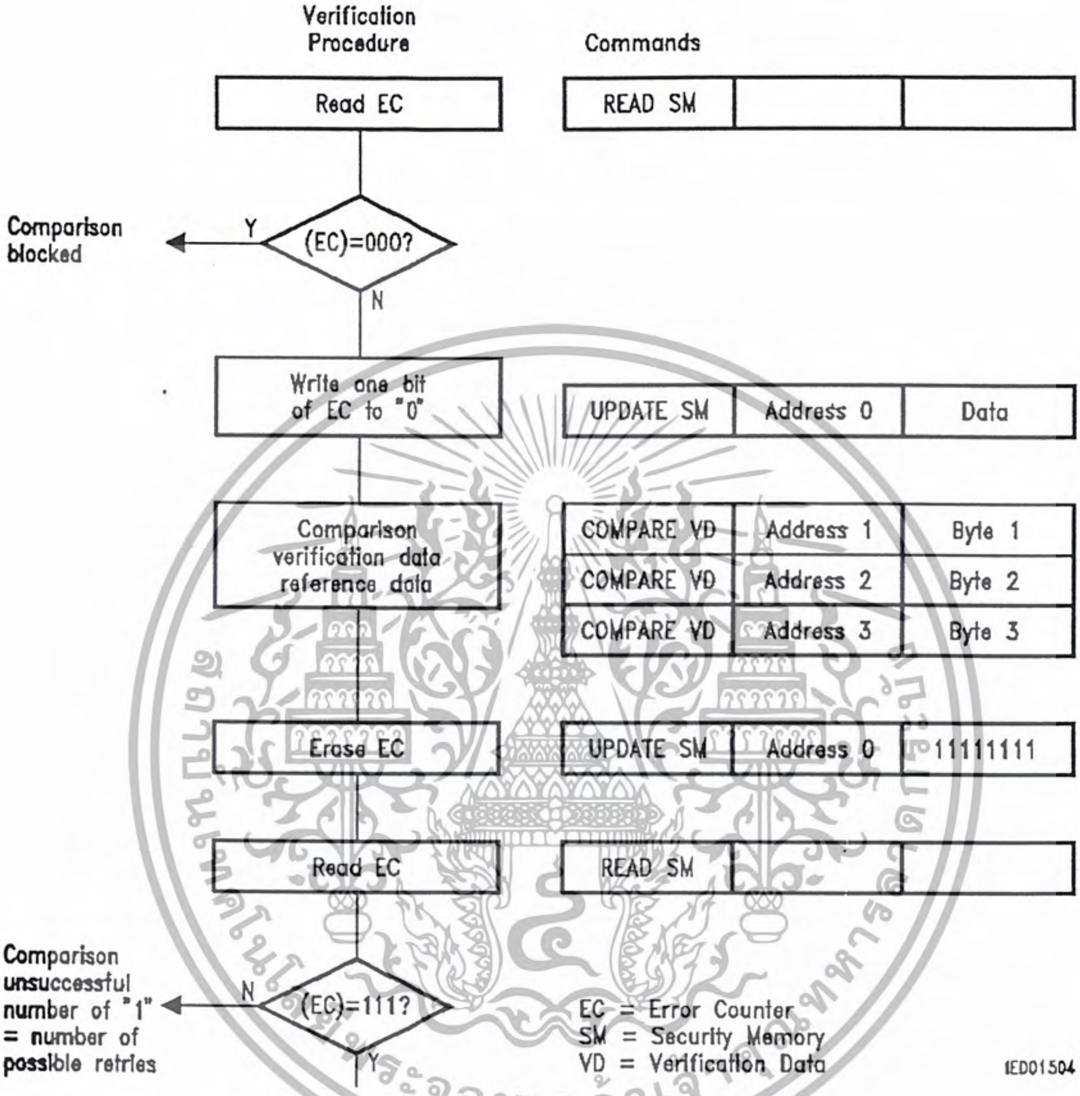
	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7…A0	D7…D0
Binary	0	0	1	1	0	0	1	1	Address	Input data
Hexadecimal	33_H								$00_H…03_H$	Input data

ตารางที่ 2.11 แสดงรูปแบบคำสั่งในการเปรียบเทียบและพิสูจน์ข้อมูล



รูปที่ 2.15 แสดงรูปสัญญาณของการเปรียบเทียบและพิสูจน์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดงกระบวนการเปรียบเทียบรหัสผ่านกับรหัส PSC

2.8.2.3 โหมดการอ่านข้อมูล (Outgoing Data Mode)

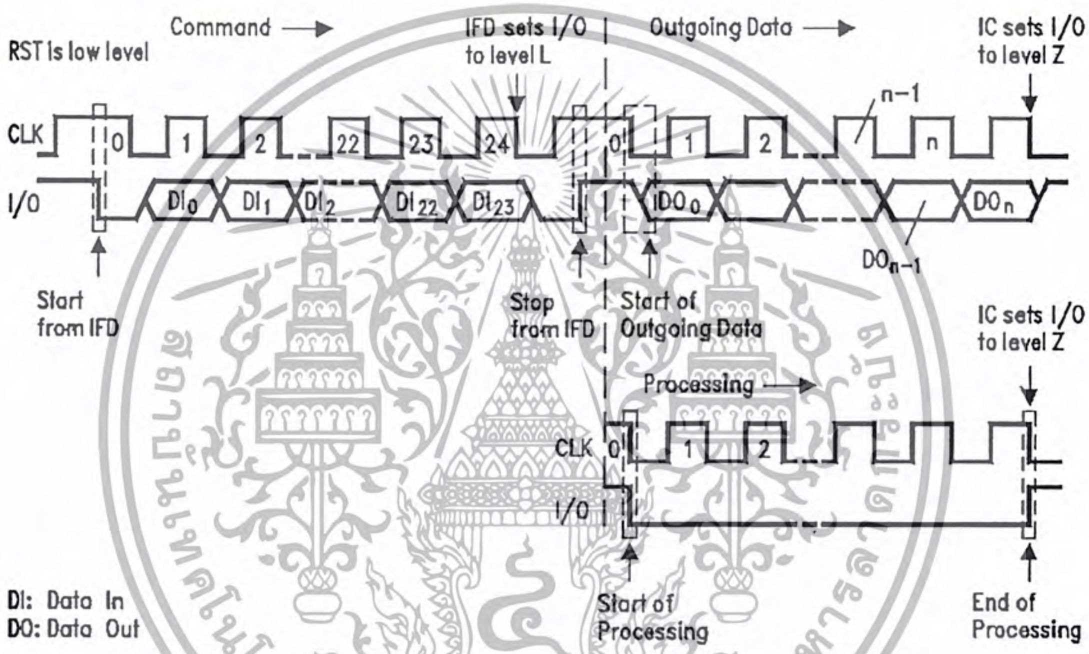
โหมดการทำงานนี้จะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขออ่านข้อมูลไปยังสมาร์ทการ์ดเพื่อขออ่านข้อมูลจากพื้นที่ใด ๆ ในหน่วยความจำ หลังจากที่ได้รับคำสั่งดังกล่าว สมาร์ทการ์ดจะส่งข้อมูลที่ถูกร้องขอกลับมายังเครื่องอ่าน ซึ่งก็เท่ากับว่าเครื่องอ่านจะสามารถอ่านข้อมูลที่ต้องการออกมาได้สำเร็จจากโหมดการทำงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2.4 โหมดดำเนินการ (Processing Mode)

โหมดดำเนินการจะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขอเขียนหรือลบข้อมูลออกจากพื้นที่ใด ๆ ในหน่วยความจำ โดยหลังจากที่ได้รับคำสั่งดังกล่าว สมาร์ทการ์ดจะเริ่มดำเนินกระบวนการตามที่ได้รับการสั่งมา ในโหมดการทำงานนี้ข้อมูลจากขา I/O จะไม่ถูกนำมาใช้ร่วมในการทำงานเลย (โดยจะมีสถานะเป็นลอจิกต่ำตลอดทั้งช่วง)

โหมดการส่งคำสั่ง, โหมดการอ่านข้อมูล และโหมดการดำเนินการ จะเรียกรวมกันว่าเป็น โหมดการประมวลผล (Operational Modes) ซึ่งรูปสัญญาณของโหมดการประมวลผลแสดงดังรูปที่ 2.17



รูปที่ 2.17 แสดงรูปสัญญาณของโหมดการประมวลผล

2.9 ไมโครคอนโทรลเลอร์ ตระกูล MCS-51

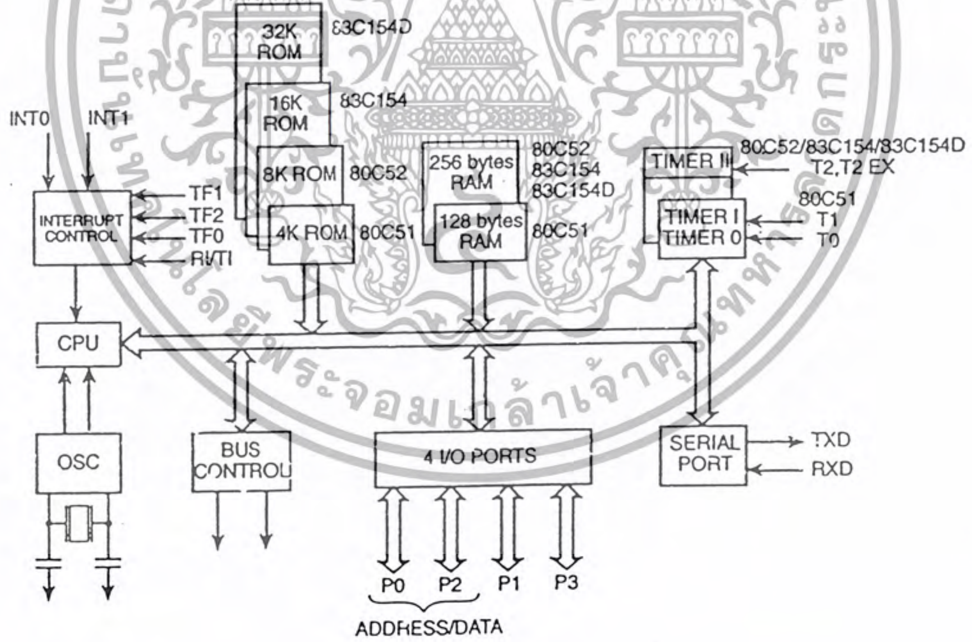
ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ที่ผลิตโดยบริษัทอินเทล ได้มีการนำไปใช้งานอย่างแพร่หลาย โดยได้ทำการเพิ่มประสิทธิภาพและหน่วยการทำงานต่าง ๆ มากขึ้น ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มีด้วยกันหลายเบอร์ ขึ้นอยู่กับโครงสร้างภายใน เช่น บางเบอร์มีหน่วยความจำแบบ ROM บางเบอร์เป็นแบบ EPROM หรือบางเบอร์ไม่มีหน่วยความจำภายใน เป็นต้น อย่างไรก็ตามลักษณะขาต่าง ๆ จะเหมือนกัน ในโครงงานนี้จะใช้ไมโครคอนโทรลเลอร์ เบอร์ MCS-51 (IC AT 89C51) ซึ่งมีหน่วยความจำโปรแกรมเป็นแบบ EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 คุณลักษณะพื้นฐานของ MCS-51

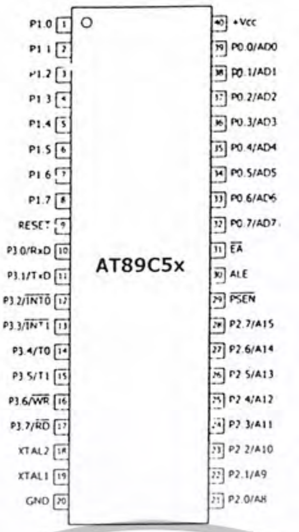
- เป็นหน่วยประมวลผลกลางขนาด 8 บิต
- มีความสามารถประมวลผลของลอจิกระดับบิต
- มีขนาดของหน่วยความจำสำหรับโปรแกรมทำงานได้ถึง 64 กิโลไบต์ (Program Memory)
- มีขนาดของหน่วยความจำสำหรับเก็บข้อมูลได้ถึง 64 กิโลไบต์ (Data Memory)
- มีหน่วยความจำสำหรับโปรแกรมภายในขนาด 4 กิโลไบต์
- มีหน่วยความจำข้อมูลภายในขนาด 128 กิโลไบต์
- มีพอร์ตสำหรับควบคุม 4 พอร์ต สามารถอ้างอิงพอร์ตได้ระดับบิตต่อบิต
- มีชุด Timer/Counter ขนาด 16 บิต 2 ชุด ทำงานได้ 4 โหมด
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ต แบบ Full Duplex เลือกรูปแบบได้ 4 โหมด
- มีวงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมทั้งกำหนดระดับความสำคัญได้ 2 ระดับ
- มีวงจรออสซิลเลเตอร์ภายใน

2.9.2 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ 8051



รูปที่ 2.18 แสดงโครงสร้างภายในของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 การจัดขามาตรฐานของของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.19 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5 โวลต์

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขานั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการ
มัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 1 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่าของไทม์เมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์ของไทม์เมอร์ 2 ในขณะที่ขา P1.4 ถึง P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในระบบ

ขาพอร์ต 2 (P2.0-P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 2 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาท์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 3 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังแสดงในตารางที่ 2.12

ขาพอร์ต	หน้าที่
P3.0	RxD – ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม
P3.1	TxD – ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม
P3.2	INT0 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 0
P3.3	INT1 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่องที่ 1
P3.4	T0 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณ ไทเมอร์จากภายนอกช่องที่ 0
P3.5	T1 - ใช้เป็นขาอินพุตสำหรับรับสัญญาณ ไทเมอร์จากภายนอกช่องที่ 1
P3.6	WR – ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอก
P3.7	RD – ใช้เป็นขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก

ตารางที่ 2.12 หน้าที่พิเศษของขาพอร์ต 3

ดังนั้นเมื่อมีการใช้สัญญาณดังกล่าวจึงไม่ควรเขียนข้อมูลไปที่พอร์ต 3 จะทำให้การทำงานผิดพลาดได้

ขารีเซต ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยในการป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา ALE/PROG (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตซ์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบ EEPROM

ขา PSEN (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้งในแต่ละแมชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใด ๆ ออกมา

ขา EA/Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหากขานี้เป็น “0” เป็นการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์ MCS-51 แบบเฟลชต้องการแรงดันสำหรับการโปรแกรม +12 โวลต์

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.9.3 พอร์ตชนิดอนุกรมอยู่ภายใน

ไมโครคอนโทรลเลอร์ทั้งหมดในตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ที่มีการเชื่อมโยงข้อมูลชนิดสองทิศทาง ทำให้รับและส่งข้อมูลพร้อมกัน ตัวรับส่งข้อมูลชนิดอะซิงโครนัส (Asynchronous Receiver) มีบัฟเฟอร์สำหรับข้อมูลเป็นพิเศษเพื่อเพิ่มความเร็วในการสื่อสาร พอร์ตชนิดอนุกรมนั้นสามารถเลือกโปรแกรมเพื่อเลือกใช้การทำงานแบบใดแบบหนึ่งใน 4 แบบ ด้วยการใช้โปรแกรมควบคุมอัตราการส่งข้อมูลและรูปแบบของข้อมูล อัตราการส่งข้อมูลที่เลือกใช้ได้สูงถึง 19,200 บิต/วินาที ด้วยความเร็วของสัญญาณนาฬิกา 1 MHz สำหรับใช้ในระบบเครือข่าย (Networks) และระบบการสื่อสารของไมโครคอนโทรลเลอร์หลายตัวร่วมกัน จะเลือกความเร็วของสัญญาณนาฬิกาด้วยวงจรนับและวงจรตั้งเวลา

2.10 แนวทางการใช้ LCD Module

รายละเอียดเกี่ยวกับ LCD Module

ใน LCD Module จะมีส่วนประกอบหลัก ๆ 3 ส่วน ดังนี้

1. ตัวแสดงผล (display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

2. ตัวควบคุม (controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของ LCD Module เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟิก

3. ตัวขับ (driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

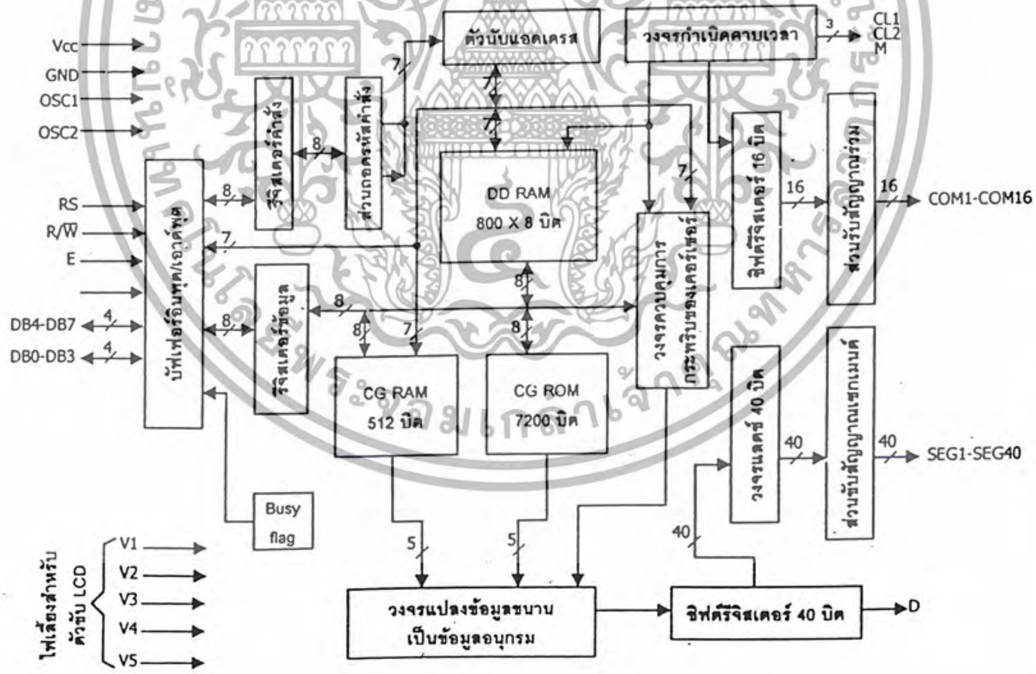
LCD Module มีอยู่หลายรุ่น และคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 แบบคือ แบบ Dot matrix และ Graphic โดยแบบ Dot matrix จะแสดงผลเป็นตัวอักษรขนาด 5x8 Dot. หรือ 5x10 Dot. มีจำนวนอักษรและบรรทัดแตกต่างกันในแต่ละรุ่น ส่วนแบบ Graphic จะแสดงผลในแบบ Bit map คือ จะสร้างเป็นภาพใด ๆ ก็ได้ตามต้องการ แนวทางการใช้งานของทั้ง 2 แบบจะมีลักษณะใกล้เคียงกัน แต่การใช้งานโดยทั่วไปมักจะใช้แบบ Dot matrix มากกว่า เนื่องจากราคาถูกกว่าและเพียงพอต่องานส่วนใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ Dot matrix LCD Module สามารถสรุปเป็นข้อ ๆ ดังนี้

1. มีให้เลือกหลายรุ่นตามแต่ต้องการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วย Dot matrix ขนาด 5x8 Dot.
3. สามารถต่อเข้ากับระบบไมโครคอนโทรลเลอร์ได้ 2 ลักษณะคือ แบบ Memory map และแบบผ่าน 8255 พอร์ต ซึ่งจะใช้ขาสัญญาณทั้งหมด 14 ขา
4. การใช้งานง่ายและสะดวก ระบบไมโครคอนโทรลเลอร์เพียงแต่ส่งข้อมูลให้กับ LCD Module เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะค้างไว้ตลอดเวลาทำให้ไม่ต้องเสียเวลาของระบบ
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกเป็นจำนวนมาก เช่น CLEAR, DISPLAY, HOME, CURSOR ON/OFF เป็นต้น
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญญาณพิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสและมีน้ำหนักน้อย รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์เท่านั้น

2.10.1 โครงสร้างภายในของตัวควบคุม LCD Module



รูปที่ 2.20 แสดงไดอะแกรมการทำงานของ LCD Module แบบอักขระ

จากรูปที่ 2.20 เป็นบล็อกไดอะแกรมภายในของชิปควบคุม LCD เบอร์ HD44780 ซึ่งเป็น LCD Module แบบอักขระ ซึ่งประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บัฟเฟอร์อินพุตเอาท์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

- รีจีสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจีสเตอร์ที่ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

- รีจีสเตอร์ข้อมูล (Data Register : DR) เป็นรีจีสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

- แรมเก็บข้อมูลแสดงผล (Display Data RAM : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจีสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up-lable) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

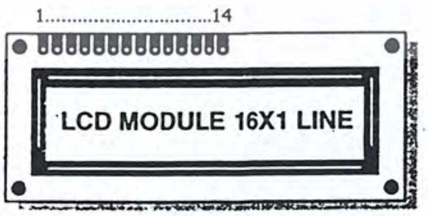
- รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

- แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

2.10.2 LCD Module ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16x1)

สำหรับ LCD Module ที่ใช้ในโครงงานนี้นั้น เป็นขนาด 16 ตัวอักษร 1 บรรทัด เนื่องจากราคาถูก หาง่าย และเป็น LCD Module ที่มีโครงสร้างที่เป็นมาตรฐาน มีผู้ผลิตหลายราย และมีการระบุเบอร์แตกต่างกันออกไปตามผู้ผลิต อาทิ LM020L ของฮิตาชิ, DMC-16117A ของออปเทร็กซ์ (Optrex) เป็นต้น แต่อย่างไรก็ตามคอนโทรลเลอร์ที่ใช้คือเบอร์เดียวกันนั่นคือ เบอร์ HD44750 ของฮิตาชิ

LCD Module ขนาด 16x1 มีขาต่อใช้งานทั้งสิ้น 14 ขา มีการจัดขาดังแสดงในรูปที่ 2.21



ขา 1 : GND
ขา 2 : +V
ขา 3 : Brightness **ปรับความสว่าง**
ขา 4 : RS
ขา 5 : R/$\overline{W}$
ขา 6 : E
ขา 7-14 : D0-D7

รูปที่ 2.21 แสดงรูปร่างและการจัดขา LCD Module แบบอักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้

V_{ss} (ขา 1) : ต่อกราวด์

V_{DD} (ขา 2) : ต่อไฟเลี้ยง +5 โวลต์

V_0 (ขา 3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจีสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจีสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6) : เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7-14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

2.10.3 คำสั่งควบคุม LCD Module

ในการเขียนคำสั่งลงในตัวควบคุม จะต้องกำหนดขา RS และ R/W เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุม LCD Module ของชิปควบคุม HD44780 ที่สำคัญมีดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (Clear Display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น “1”

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	1

ตารางที่ 2.13 แสดงคำสั่งเคลียร์ตัวแสดงผล

2. คำสั่ง Cursor at Home (หรือ Return Home)

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	1	*

ตารางที่ 2.14 แสดงคำสั่ง Cursor at Home

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังตารางที่2.5

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	1	I/D	S

ตารางที่ 2.15 แสดงคำสั่งเลือกโหมดการป้อนข้อมูล

- บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น "1" เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น "0" เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

- บิต I/D เป็นบิตที่ใช้ในการกำหนดว่าเมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น "1" แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0" แอดเดรสจะลดลง

ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล (Display ON/OFF)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	1	D	C	B

ตารางที่ 2.16 แสดงคำสั่งควบคุมการแสดงผล

- บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น "1" จะเป็นการเปิดจอแสดงผล ถ้าเป็น "0" จะเป็นการปิดจอแสดงผล

- บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0” จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

- บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกระพริบ แต่ถ้าบิตนี้เป็น “0” จะไม่มีการกระพริบที่เคอร์เซอร์

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร (Display Shift)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	1	S/C	R/L	*	*

ตารางที่ 2.17 แสดงคำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	*R/L*	*ลักษณะการเลื่อน*	*ข้อมูลคำสั่ง*
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน (Function Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	DL	N	F	*	*

ตารางที่ 2.18 แสดงคำสั่งกำหนดฟังก์ชันการทำงาน

- บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “0” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “1” จะเป็นแบบ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น "0" จะแสดงผล 1 บรรทัด ถ้าเป็น "1" จะแสดงผล 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น "1"

- บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น "0" จะเป็นการแสดงผลแบบ 5x8 จุด และถ้าเป็น "1" จะเป็นการแสดงผลแบบ 5x10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้ LCD Module ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5x8 จุด

จุดที่น่าสังเกตคือ LCD Module แบบ 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่ต้องกำหนดให้ N เป็น "1" เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่วง คือ 8 ตัวอักษรแรกจะเริ่มที่ 00H และอีก 8 ตัวอักษรถัดไปจะเริ่มที่ 40H

7. คำสั่งเลือกแอดเดรสของ CGRAM (Set CGRAM Address)

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น "0" บิต 6 เป็น "1" ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	1	CGRAM ADDRESS					

ตารางที่ 2.19 แสดงคำสั่งเลือกแอดเดรสของ CGRAM

8. คำสั่งเลือกแอดเดรสของ DDRAM (Set DDRAM Address)

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น "1" และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น "0" แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 80H-87H และ 0C0H-0C7H

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	0	1	DDRAM ADDRESS						

ตารางที่ 2.20 แสดงคำสั่งเลือกแอดเดรสของ DDRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LCD แบบ 16 ตัวอักษร 1 บรรทัด ที่ใช้ในโครงงานนี้ มีการกำหนดตำแหน่ง DDRAM Address ดังนี้

80	81	82	83	84	85	86	87	C0	C1	C2	C3	C4	C5	C6	C7

ตารางที่ 2.21 แสดงการกำหนดตำแหน่ง DDRAM Address ที่ใช้ในโครงงานนี้

9.คำสั่งอ่านแฟลก BUSY และแอดเดรส (Read BUSY flag & Address)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0
0	1	BF	CGRAM/DDRAM ADDRESS						

ตารางที่ 2.22 แสดงคำสั่งอ่านแฟลก BUSY และแอดเดรส

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “0” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “1” แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่ เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0 – บิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.10.4 การเขียนคำสั่งและข้อมูลให้แก่ LCD Module

ในการเขียนข้อมูลเพื่อควบคุมให้ LCD Module แสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดการทำงานให้แก่ LCD Module ก่อน จากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผล เนื่องจากบัสข้อมูลของ LCD Module มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “0” หมายความว่า ข้อมูลที่ป้อนให้แก่ LCD Module ขณะนั้นเป็นคำสั่ง ในทางตรงข้าม หากขา RS ได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลลงใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายใน LCD Module ทราบว่าข้อมูลที่กำหนดต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น “1” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “1” ให้ขา RS แล้ว ต้องกำหนดให้ขา R/W เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจีสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

ชุดคำสั่งและเวลาที่ LCD Module ใช้ในการทำงานแต่ละคำสั่ง แสดงดังตารางที่ 2.23

INSTRUCTION	RS	R/W	D7	D6	D5	D4	D3	D2	D1	D0	EXECUTE TIME (nS)
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADDR.	0	0	0	1	CGRAM ADDRESS						40
SET DDRAM ADDR.	0	0	1	DDRAM ADDRESS							40
BUSY, ADDR. READ	0	1	BF	CGRAM/DDRAM ADDRESS							0
CGRAM, DDRAM WR	1	0	WRITE DATA								40
CGRAM, DDRAM RD	1	1	READ DATA								40

ตารางที่ 2.23 แสดงชุดคำสั่งและเวลาที่ LCD Module ใช้ในการทำงานแต่ละคำสั่ง

2.10.5 จังหวะการทำงานของ **LCD Module**

ในการติดต่อกับ LCD Module จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD Module แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

ดังนั้น ในการใช้งาน LCD Module ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้ LCD Module พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่ LCD Module ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้ LCD Module ทำการเตรียมความพร้อมหรืออินิเชียล (initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของ LCD Module แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาทีเพื่อให้คอนโทรลเลอร์ใน LCD Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แปลความหมายของลอจิกที่ขา RS ว่าข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิล LCD Module ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของ LCD Module ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของ LCD Module จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินิเชียล LCD, โปรแกรมหน่วงเวลา และโปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิล LCD Module

2.11 พอร์ต RS-232C

โดยปกติคอมพิวเตอร์จะมีพอร์ตเป็นอนุกรม เรียกชื่อกันว่า RS-232C อยู่ในตัวเองอยู่แล้ว ซึ่งพอร์ต RS-232C นี้จะทำหน้าที่รับและส่งข้อมูลในแบบอนุกรม เหตุที่มีชื่อเรียกว่าพอร์ต RS-232C ก็เนื่องมาจากสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์ของประเทศสหรัฐอเมริกา (Electronic Industries Association: EIA) ได้กำหนดตามมาตรฐานอุปกรณ์สื่อสารแบบอนุกรม ใช้ภายใต้ชื่อว่าพอร์ต RS-232C และมาตรฐานการส่งข้อมูลแบบอนุกรมได้มีหลายมาตรฐาน แต่ที่นิยมกันมากที่สุดสำหรับคอมพิวเตอร์ก็คือ พอร์ต RS-232C

หน้าที่สำคัญของการสื่อสารแบบอะซิงโครนัส คือ

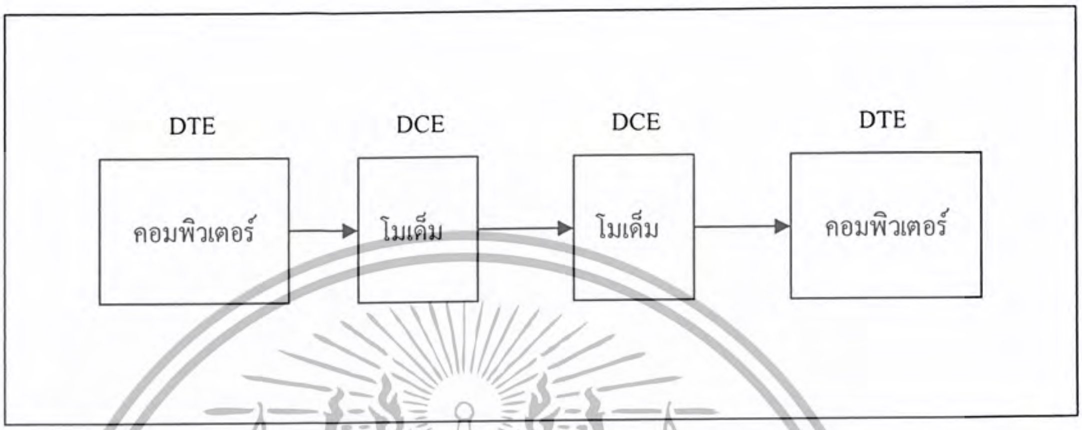
- รับสัญญาณ
  1. เปลี่ยนสัญญาณที่เข้ามาแบบอนุกรมให้เป็นแบบขนาน
  2. ตรวจสอบหาความผิดพลาดของสัญญาณที่รับ
  3. ตัดบิตหยุดและพาริตี้บิตออก
  4. ส่งสัญญาณให้โปรเซสเซอร์รู้ว่ารับสัญญาณไว้แล้ว
- ส่งสัญญาณ
  1. เปลี่ยนสัญญาณแบบขนานจากโปรเซสเซอร์ส่งออกเป็นแบบอนุกรม
  2. เพิ่มบิตหยุดและพาริตี้บิต
  3. เพิ่มสัญญาณควบคุมโมเด็มที่ต่อเชื่อม (ถ้ามี)

2.11.1 มาตรฐาน RS-232C

มาตรฐาน RS-232C ได้จัดพิมพ์ขึ้นเมื่อปี ค.ศ. 1969 โดยสมาคมผู้ผลิตอุปกรณ์อิเล็กทรอนิกส์แห่งประเทศสหรัฐอเมริกา RS ย่อมาจาก Recommended Standard และส่วน 232 จะเป็นหมายเลขบ่งบอกของมาตรฐานตัวนี้ ตัวอักษร C คือหมายเลขของฉบับหนึ่งของมาตรฐาน จุดประสงค์ของมาตรฐานตัวนี้ก็เพื่อบรรยายคุณลักษณะของการเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment: DTE) กับรูปแบบอุปกรณ์สื่อสารข้อมูล (Data Communication Equipment: DCE) สำหรับผู้ใช้คอมพิวเตอร์ DTE หมายถึง ตัวคอมพิวเตอร์ และ DCE หมายถึงโมเด็ม อุปกรณ์อื่น ๆ เช่น เครื่องพิมพ์ที่รับสัญญาณแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนุกรมอาจจะเป็นได้ทั้ง DTE และ DCE ขึ้นอยู่กับผู้ผลิต “การเชื่อมต่ออุปกรณ์รับส่งข้อมูลปลายทาง (Data Terminal Equipment: DTE)” ซึ่งต่อไปนี้จะขอเรียกว่า “DTE” “รูปแบบสื่อสารข้อมูล (Data Communication Equipment: DCE)” ซึ่งไปนี้จะขอเรียกว่า “DCE” โดยข้อแตกต่างของ DTE และ DCE จะเห็นได้จากรูปที่ 2.22 จากภาพนี้จะเห็นได้ว่า RS-232C มีส่วนสำคัญสำหรับการติดต่อสื่อสารข้อมูลระหว่างคอมพิวเตอร์



รูปที่ 2.22 การใช้งานพอร์ต RS-232C เชื่อมต่ออุปกรณ์

ความเร็วและระยะทางการเชื่อมต่อพอร์ตอนุกรม RS-232C สามารถเชื่อมต่อการถ่ายโอนข้อมูลได้จาก 0-115,000 บิตต่อวินาที ซึ่งเพียงพอสำหรับคอมพิวเตอร์ที่มีอัตราการถ่ายเทข้อมูล 110 ถึง 9,600 บิตต่อวินาที ความยาวของสายเชื่อมต่อสัญญาณตามมาตรฐานของพอร์ตอนุกรม RS-232C จำกัดแค่ 50 ฟุต ซึ่งพอสำหรับการสื่อสารคอมพิวเตอร์กับอุปกรณ์รอบนอก

2.11.2 ลักษณะของสัญญาณ RS-232C

เพื่อเป็นหลักประกันว่าข้อมูลส่งออกไปอย่างถูกต้องและอุปกรณ์ถูกควบคุมอย่างถูกต้อง จำเป็นต้องมีข้อตกลงกันในเรื่องของสัญญาณที่ใช้มาตรฐานของพอร์ตอนุกรม RS-232C และกำหนดย่านแรงดันไฟฟ้าในสัญญาณ ดังแสดงในตารางที่ 2.24 และรูปที่ 2.23 สำหรับคอมพิวเตอร์บางเครื่องใช้แต่สัญญาณลอจิกออกมาเป็นสัญญาณของพอร์ตอนุกรม RS-232C อย่างเช่น ตัวปรับต่ออะซิงโครนัสของคอมพิวเตอร์ไอบีเอ็ม กรณีเช่นนี้ระยะทางของสายที่เชื่อมต่ออาจจะสั้นกว่า 50 ฟุต ซึ่งดังที่กล่าวเอาไว้เนื่องจากระดับของกราวนด์จะเปลี่ยนแปลงไป อันเนื่องจากการสูญเสียไปในความต้านทานของสาย ผู้ที่เคยใช้คอมพิวเตอร์ไอบีเอ็มอาจจะประสบปัญหานี้เมื่อต่อสัญญาณ RS-232C เกินกว่า 10 ฟุต แล้วใช้การไม่ได้ แต่อย่างไรก็ตามพอร์ต RS-232C ของคอมพิวเตอร์ไอบีเอ็มยังมีโอกาสให้เลือกใช้กระแสไฟฟ้า 20 มิลลิแอมป์ เป็นกระแสไฟฟ้าในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

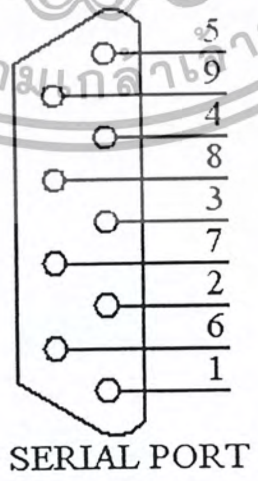
มาตรฐานของการให้แรงดันไฟฟ้า			
แรงดันไฟฟ้า	สถานภาพลอจิก	สถานภาพสัญญาณ	ฟังก์ชั่นในการควบคุม
บวก	0	SPACE	ON
ลบ	1	MARK	OFF

ตารางที่ 2.24 มาตรฐานการใช้แรงดันไฟฟ้าในสัญญาณ RS-232C

รูปที่ 2.23 ย่านของแรงดันไฟฟ้าที่ใช้ในสัญญาณ RS-232C

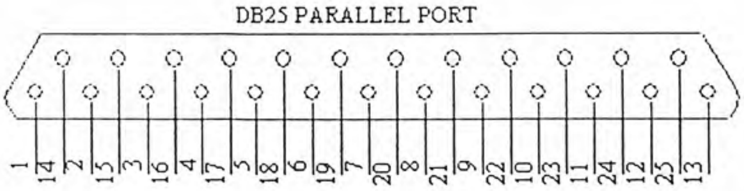
2.11.3 การกำหนดข้อต่อของพอร์ต RS-232C

มาตรฐานการเชื่อมต่อแบบ RS-232C จะใช้คอนเน็คเตอร์แบบ DB-25 หรือ DB-9 ตัวผู้ ซึ่งคอนเน็คเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป โดยสามารถแสดงรูปร่างและตำแหน่งขาได้ดังรูปที่ 2.24 และ 2.25



รูปที่ 2.24 คอนเน็คเตอร์อนุกรม 9 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 คอนเน็คเตอร์อนุกรม 25 ขา

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

ตารางที่ 2.25 การจัดขาคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232C แบบ DB-9

สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232C มีดังนี้

1. Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect: CD ขานี้จะแอคทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก

2. Received Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจีสเตอร์บัฟเฟอร์

3. Transmitted Data: TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป

4. Data Terminal Ready: DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วย ในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

5. Signal Ground: GND ขากราวนด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Data Set Ready: DSR ขานี้ใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

7. Request To Send: RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

8. Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

9. Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็ม และโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.11.4 การเชื่อมต่อมาตรฐาน RS-232C

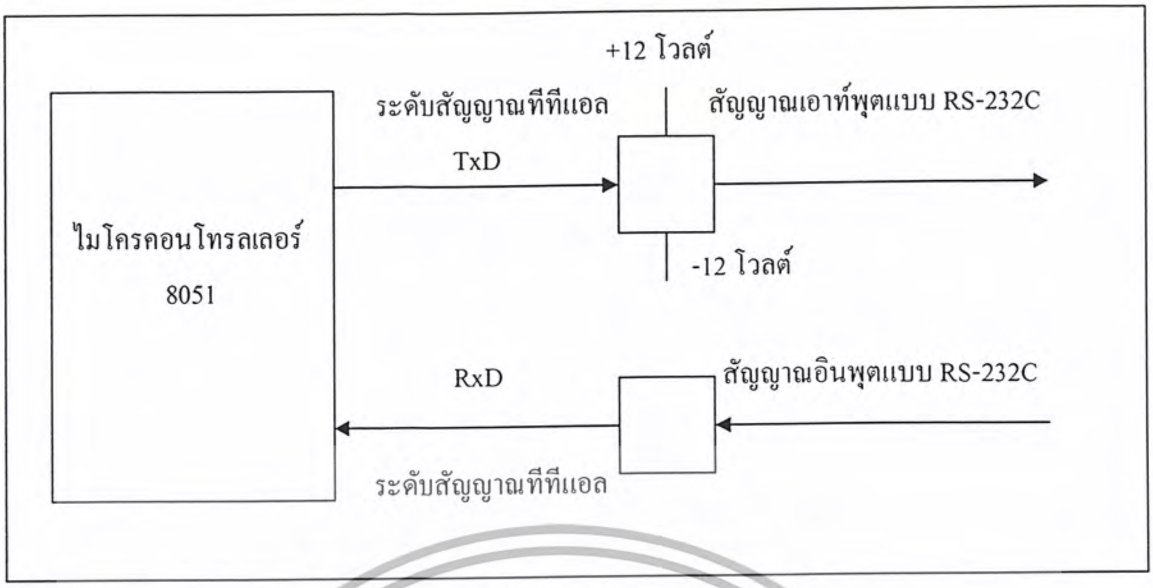
ในเชื่อมต่ออนุกรมเข้ากับอุปกรณ์คอมพิวเตอร์ต่าง ๆ เช่น โมเด็ม มักจะกำหนดในการเชื่อมต่อตามมาตรฐาน RS-232C ทั้งนี้เพื่อให้มีการใช้งานเส้นสัญญาณหรือรูปแบบของตัวเชื่อมต่อที่สอดคล้องกันจะได้ลดปัญหาการเข้ากันไม่ได้ระหว่างสัญญาณของอุปกรณ์ที่มาเชื่อมต่อทั้งสองด้านให้น้อยลง เนื่องจากระดับแรงดันไฟฟ้าที่ใช้ และการแทนความหมายของระดับลอจิกตามมาตรฐานนี้แตกต่างไปจากที่ใช้งานกันในระบบดิจิตอลทั่วไป โดยระดับสัญญาณของพอร์ต RS-232C เป็นแบบสองขั้ว (Bipolar) ที่ระดับแรงดันไฟฟ้าทางด้านลบช่วง -3 โวลต์ ถึง -15 โวลต์ ซึ่งจะแทนค่าลอจิกสูง และแรงดันไฟฟ้าบวกช่วง +3 โวลต์ ถึง +15 โวลต์ แทนค่าลอจิกต่ำ ดังนั้นจะเห็นได้ว่ามีความจำเป็นที่จะต้องเพิ่มเติมอุปกรณ์หรือวงจรพิเศษเข้าไป เพื่อเปลี่ยนระดับแรงดันไฟฟ้าจากระบบอยู่ในช่วง +3 โวลต์ ถึง +5 โวลต์ ซึ่งได้จากขาสัญญาณของตัวไมโครคอนโทรลเลอร์ 8051 ซึ่งเป็นระดับแรงดันไฟฟ้าที่สูงกว่าค่า +3 โวลต์ หรือต่ำกว่า -3 โวลต์ ดังรูปที่ 2.23 จะเห็นว่า Received Data ของไมโครคอนโทรลเลอร์ 8051 จะต้องถูกปรับเปลี่ยนไปให้เป็นระดับสัญญาณ RS-232C ก่อน หลังจากนั้นจะทำการส่งออกไปในสายสัญญาณต่อไป ลักษณะโดยย่อของสัญญาณ RS-232C ดังรูปที่ 2.26 ระดับสัญญาณแบบทีทีแอล (TTL) จากขาสัญญาณ TxD (Transmitted Data) และ RxD (Received Data)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Driver output logic levels with	15 Volt > O_h > 5 Volt
3 kbyte to 7 k load	-5 Volt > O_l > -15 Volt
Driver output Voltage when open circuit	Vo < 25 Volt
Driver output impedance with Power off Ro>300 ohms	
Output short circuit current	Io < 0.5 A
Driver slew rate	dv/dt < 30 V/s
Receiver input impedance	7 kbyte > Rin > 3 k
Receiver input voltage	+15 compatible with driver
Receiver output with open circuit input	MARK
Receiver output with +3 Volt input	SPACE
Receiver output with -3 Volt input	MARK
+15	LOGIC 0 = SPACE, CONTROL ON
+5	Noise Margin
+3	Transition Region
-3	Noise Margin
-5	LOGIC 1 = MARK byte
-15	CONTROL OFF

รูปที่ 2.26 คุณลักษณะโดยย่อของสัญญาณ RS-232C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 การเปลี่ยนแปลงสัญญาณทีทีแอล (TTL)

2.12 โปรแกรมวิชวลเบสิก (Visual Basic)

วิชวลเบสิก(Visual Basic) เป็นภาษาคอมพิวเตอร์ที่นิยมนำมาใช้พัฒนาโปรแกรมบน Windows เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแต่เลือก Control ที่เหมาะสมแล้ววางลงบน Form ก็สามารถที่จะสร้างจอภาพที่ใช้สำหรับติดต่อกันผู้ใช้ รวมทั้งการใช้เทคนิคการเขียนโปรแกรมรวมแบบ Event-driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับ Control ต่างๆ ที่สร้างขึ้นตามเหตุการณ์ต่างๆที่เกิดขึ้น เช่น การเลื่อนเม้าส์ หรือการรับข้อมูลจากคีย์บอร์ด ฯลฯ เป็นต้น

2.12.1 โปรแกรมติดต่อและควบคุมผ่านพอร์ตอนุกรม

Control สำคัญที่ทำให้วิชวลเบสิกสามารถติดต่อผ่านพอร์ตอนุกรมได้นั่นคือ Cortrol MSComm โดยพร็อพเพอร์ตี้ที่สำคัญในการใช้งาน MSComm มีดังนี้

- CommPort ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการจะติดต่อ โดยมีรูปแบบการใช้งานดังนี้

  Object.Commport = value

  ยกตัวอย่างเช่น ถ้าเขียนโปรแกรมติดต่อกับพอร์ต Com1 จะเขียนเป็น

  MSComm1.CommPort = 1

- Settings ใช้กำหนดอัตราบอด(Baud Rate) หรือความเร็วในการส่งข้อมูลมีหน่วยเป็นบิตต่อวินาที, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้าย โดยมีรูปแบบการใช้งานดังนี้

  Object.Settings = value

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่างเช่น ถ้าเขียนโปรแกรมใช้งานที่อัตราบอดเท่ากับ 9,600 บิตต่อวินาที,ไม่มีพาริตี้,จำนวนบิตข้อมูลเท่ากับ 8 บิต และมีบิตปิดท้าย 1 บิต จะเขียนได้เป็น

MSComm1.Settings = "9600,N,8,1"

- PortOpen ใช้สำหรับเปิดและปิดการใช้งานพอร์ตอนุกรม ถ้าจะเปิดใช้งานพอร์ตอนุกรม ให้กำหนดค่า value เป็น "true" ถ้าจะปิดพอร์ตอนุกรมให้กำหนดค่า value เป็น "false" โดยมีรูปแบบการใช้งานดังนี้

Object.PortOpen = value

- InBuffersize เป็นการกำหนดขนาดของ Buffer ในการรับข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

Object.InBuffersize = value

- OutBuffersize เป็นการกำหนดขนาดของ Buffer ในการส่งข้อมูลออกไป โดยมีรูปแบบการทำงานดังนี้

Object.OutBuffersize = value

- Inputlen เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ภาครับโดยมีรูปแบบการทำงานดังนี้

Object.Inputlen = value

- InputMode เป็นการกำหนดค่าชนิดของข้อมูลที่รับเข้ามา ถ้าข้อมูลที่เข้ามาเป็นข้อความปรกติจะต้องกำหนด value = เป็น "0" และถ้าข้อมูลที่เข้ามาเป็นเป็นข้อมูลไบนารี่จะต้องกำหนด value = "1" โดยมีรูปแบบการกำหนดค่าดังนี้

Object.InputMode = value

- Input ใช้ในการอ่านข้อมูลจากพอร์ตอนุกรม โดยมีรูปแบบการอ่านดังนี้

value = Object.Input

- Output ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบการส่งดังนี้

Object.Output = value

- EOFEnable เป็นการบอกการสิ้นสุดของไฟล์ End of File(EOF) โดยมีรูปแบบการใช้งานดังนี้

Object.EOFEnable = value

2.12.2 โปรแกรมเพื่อสร้างระบบฐานข้อมูล

ในการติดต่อกับฐานข้อมูลจะใช้ ADO Data Control ซึ่งเป็น Control ใหม่ที่ปรากฏใน Visual Basic 6.0 ที่ไมโครซอฟท์พัฒนาขึ้นเพื่อรองรับแนวความคิด Universal Data Access (UDA) ที่จุดมุ่งหมายที่ต้องการให้เกิดศูนย์กลางการติดต่อระหว่าง Application ที่มีการใช้ข้อมูลที่ต่างรูปแบบกันสามารถข้อมูลร่วมกันได้ โดยไม่ต้องเปลี่ยนรูปแบบของข้อมูลเดิมของแต่ละ Application แต่อย่างใด สำหรับ Visual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Basic 6.0 นั้น สามารถติดต่อกับฐานข้อมูลได้ทุกชนิด โดยอาศัยเทคโนโลยีหลายๆอย่าง แต่ที่นิยมใช้คือ Microsoft Access ซึ่งมีโครงสร้างแบบ Relational Database ที่มีข้อดีคือ สามารถนำข้อมูลจากหลายๆ ตาราง ที่มีความสัมพันธ์กันมาใช้งานร่วมกันได้

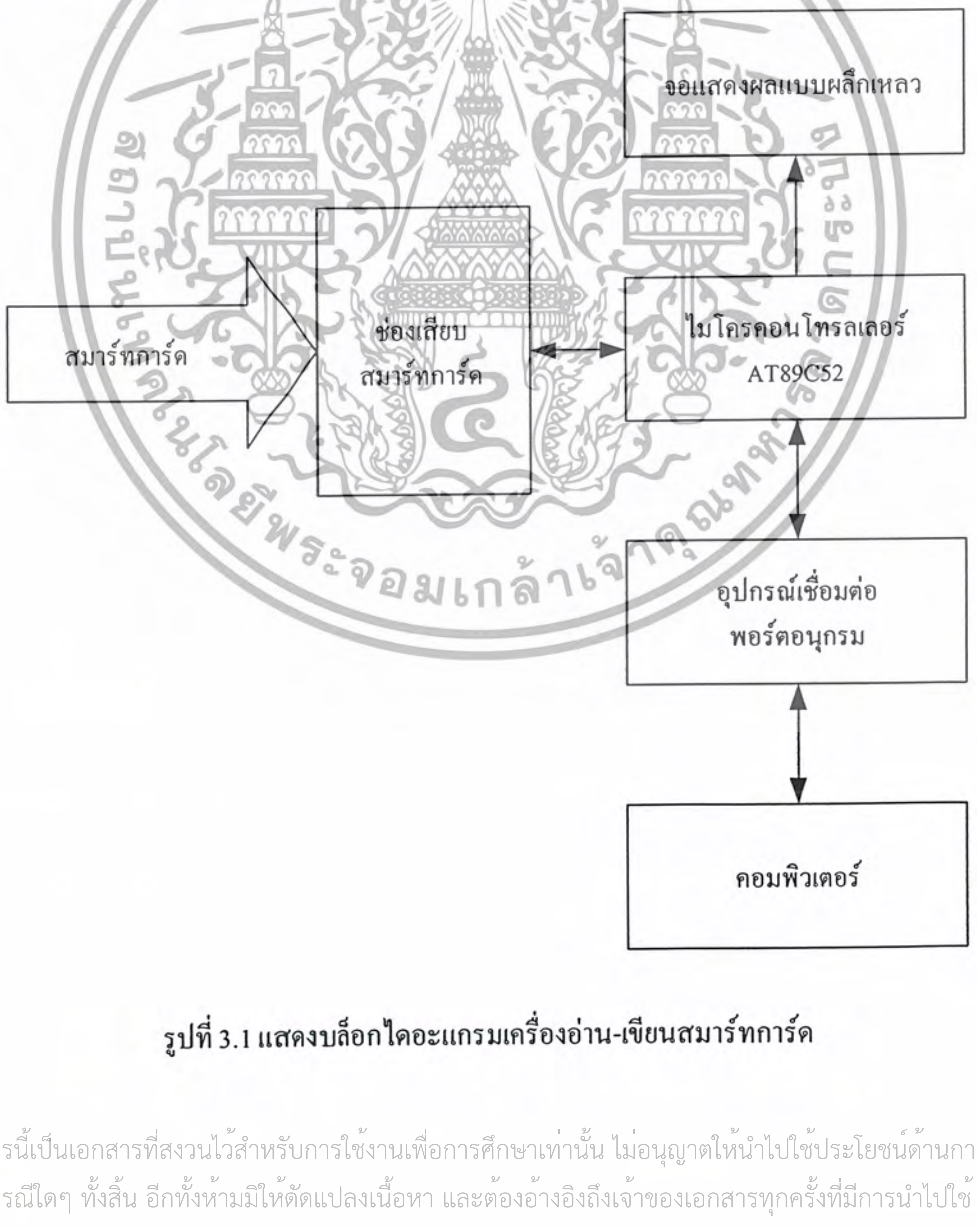
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

ในโครงงานสมาร์ทการ์ดกับระบบสารสนเทศนักศึกษานั้น จะมีส่วนประกอบที่สำคัญ 2 ส่วน คือ ส่วนแรกเป็นส่วนของเครื่องอ่าน-เขียนสมาร์ทการ์ด ที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ AT89C52 ซึ่งจะแสดงผลที่จอแสดงผลแบบผลึกเหลว (LCD) และที่เครื่องคอมพิวเตอร์ ส่วนที่สองคือ ส่วนของโปรแกรมที่ใช้ติดต่อกับผู้ใช้งาน และฐานข้อมูลระบบสารสนเทศนักศึกษาซึ่งจัดทำโดยใช้ภาษา Visual Basic

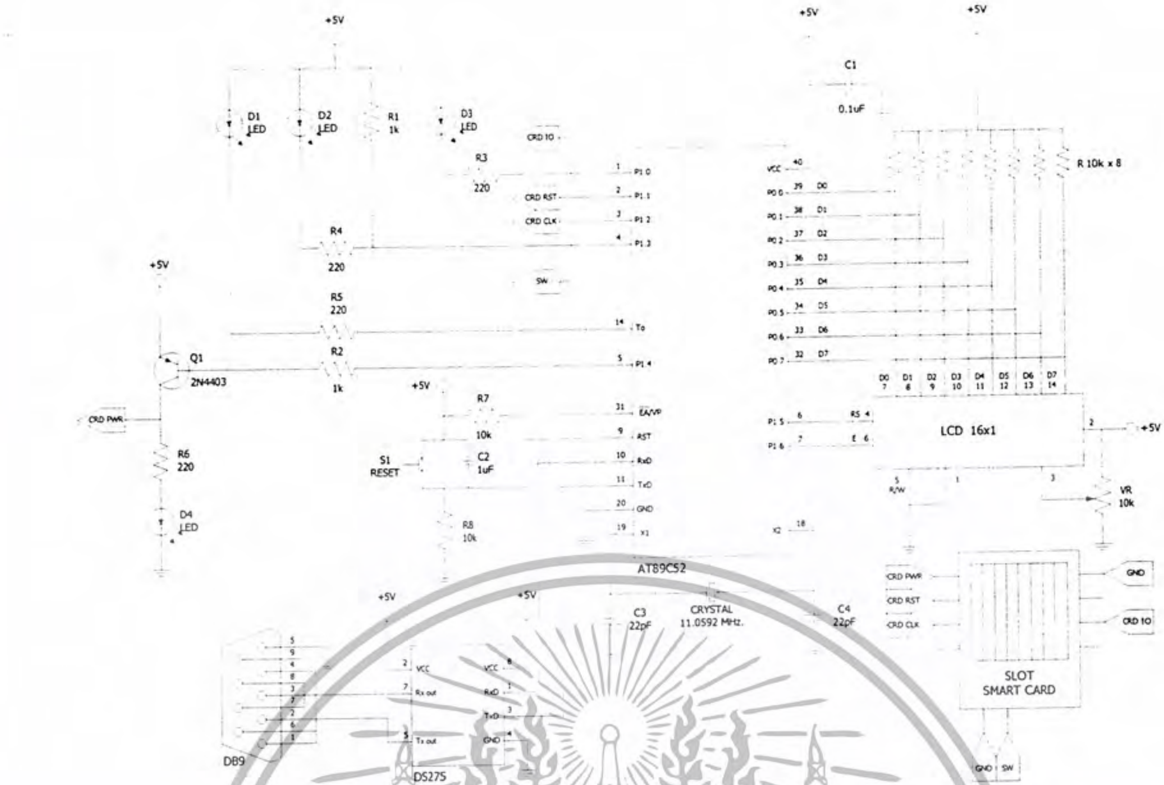
3.1 การออกแบบและสร้างเครื่องอ่าน-เขียนสมาร์ทการ์ด

ในการออกแบบเครื่องอ่าน-เขียนสมาร์ทการ์ด จะควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์ AT89C52 ซึ่งจะเชื่อมต่อกับช่องเสียบบัตรสมาร์ทการ์ดและติดต่อกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม และยังมีส่วนแสดงผลด้วยจอแสดงผลแบบผลึกเหลวอีกด้วย ดังรูปที่ 3.1



รูปที่ 3.1 แสดงบล็อกไดอะแกรมเครื่องอ่าน-เขียนสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงวงจรของเครื่องอ่าน-เขียนสมาร์ทการ์ด

จากรูปที่ 3.2 แสดงวงจรของเครื่องอ่าน-เขียนสมาร์ทการ์ด จะเห็นว่ามีช่องเสียบบัตรสมาร์ทการ์ด (Slot Smart card) ซึ่งถูกต่อเข้ากับขาของไมโครคอนโทรเลอร์ โดยที่ขา 1 ต่อเข้ากับตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ทการ์ด ซึ่งเป็นขาที่ใช้เขียนหรืออ่านข้อมูลจากบัตร ที่ขานี้จะมี LED3 เป็นตัวแสดงผลเมื่อไมโครคอนโทรลเลอร์เขียนหรืออ่านข้อมูลจากบัตร โดย LED3 จะติด/ดับตามสถานะของลอจิกที่ขา 1 ของไมโครคอนโทรลเลอร์ ขา 2 ต่อเข้ากับตำแหน่ง CRD RST ของช่องเสียบบัตรสมาร์ทการ์ด ซึ่งเป็นขาที่ใช้รีเซตการทำงานของบัตร ขา 3 ต่อเข้ากับตำแหน่ง CRD CLK ของช่องเสียบบัตรสมาร์ทการ์ด ซึ่งขานี้เป็นขาที่ใช้ป้อนสัญญาณนาฬิกาอ้างอิงให้บัตรสมาร์ทการ์ด ขา 4 ต่อเข้ากับตำแหน่ง SW ของช่องเสียบบัตรสมาร์ทการ์ด ซึ่งเป็นขาที่ใช้ตรวจสอบและแสดงผลว่ามีการเสียบบัตรเข้ามายังช่องเสียบบัตรแล้วหรือยัง ที่ขานี้จะมี LED2 เป็นตัวแสดงผล ซึ่งจะติดสว่างเมื่อบัตรถูกเสียบเข้ามายังช่องเสียบบัตร และขา 5 ต่อกับตำแหน่ง CRD PWR ของช่องเสียบบัตรสมาร์ทการ์ด ซึ่งเป็นขาควบคุมการจ่ายไฟเลี้ยงให้กับบัตร ด้วยการส่งให้ Q1 ปิดหรือเปิดวงจร ที่ขานี้จะมี LED4 เป็นตัวแสดงผล เมื่อไฟเลี้ยงถูกจ่ายไปยังบัตร LED4 ก็จะติดสว่างตาม และที่ขา 14 จะต่ออยู่กับ LED1 ซึ่งจะกระพริบทุก ๆ 1 วินาทีเพื่อแสดงว่าบอร์ดกำลังทำงานอยู่

ที่ขา 10 ของไมโครคอนโทรลเลอร์เป็นขาสำหรับรับข้อมูลเข้าหรือ RxD และที่ขา 11 เป็นขาสำหรับส่งข้อมูลออกหรือ TxD ทั้งสองขาจะทำการเชื่อมต่อเข้ากับวงจรสื่อสารกับพอร์ตอนุกรม เพื่อสื่อสารข้อมูลกับโปรแกรมควบคุมคอมพิวเตอร์ ซึ่งมีไอซีเบอร์ DS275 เป็นหัวใจการทำงานหลัก ทำหน้าที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

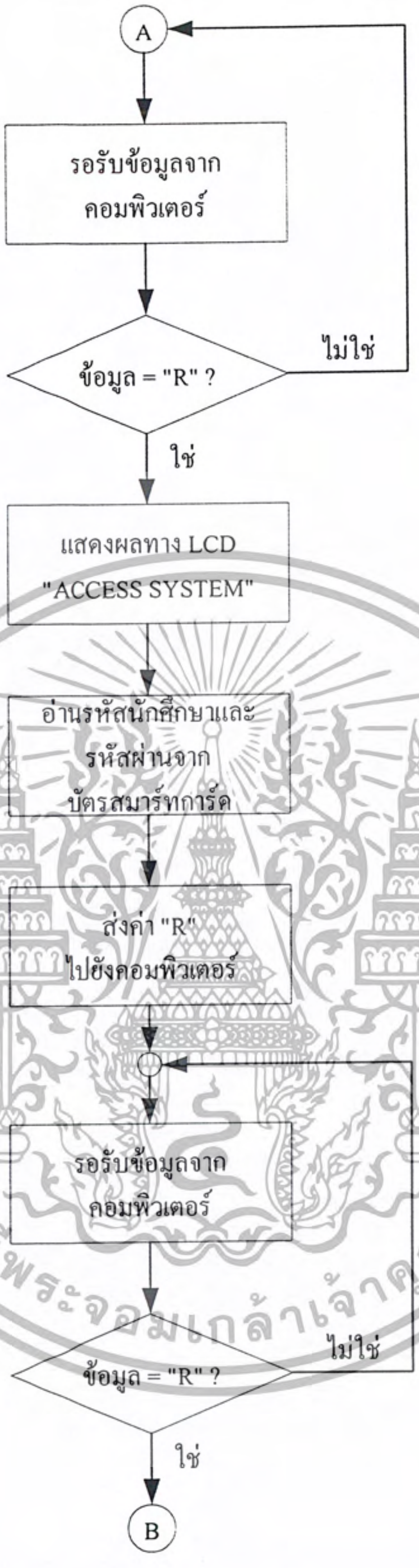
เป็นตัวกลางในการติดต่อ โดยไมโครคอนโทรลเลอร์อ่านข้อมูลที่รับเข้ามาหรือส่งออกไปที่รีจีสเตอร์บัฟเฟอร์ ที่ขา 32-39 ซึ่งเป็นพอร์ต 0 (P0) ของไมโครคอนโทรเลอร์ต่อเข้ากับขาข้อมูลทั้ง 8 เส้น ของจอแสดงผลแบบผลึกเหลวขนาด 16 ตัวอักษร 1 บรรทัด และขา 6 ซึ่งเป็น P1.5 ของไมโครคอนโทรลเลอร์จะต่อเข้ากับขา RS ของจอแสดงผลแบบผลึกเหลว ส่วนขา 7 ซึ่งเป็น P1.6 ของไมโครคอนโทรลเลอร์จะต่อเข้ากับขา E ของจอแสดงผลแบบผลึกเหลว และที่ขา 3 ของจอแสดงผลแบบผลึกเหลวจะต่อกับตัวต้านทานปรับค่าได้ (VR) ขนาด 10 kΩ ซึ่งจะเป็นตัวปรับความเข้มของจอแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด

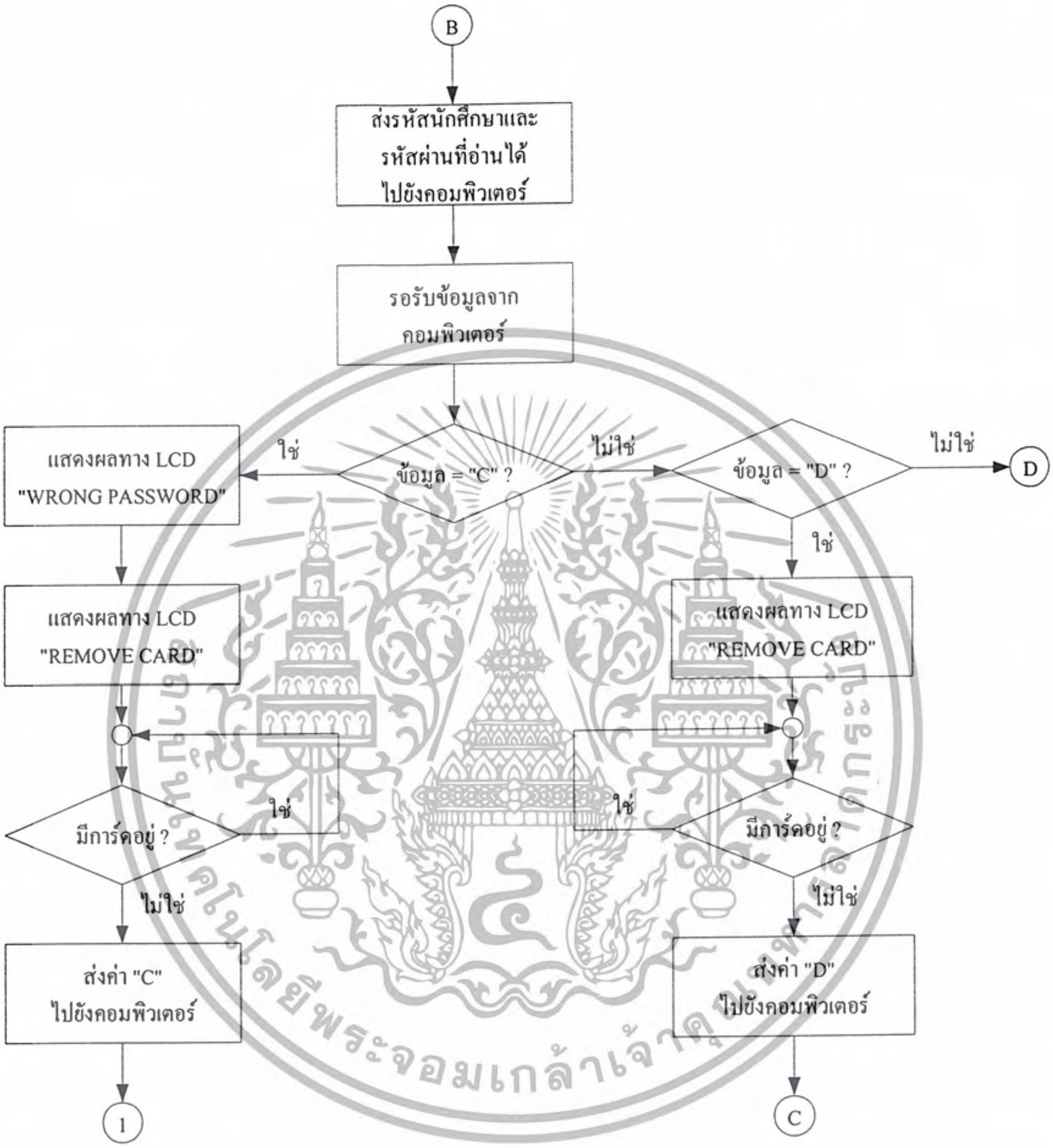
รูปที่ 3.3 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



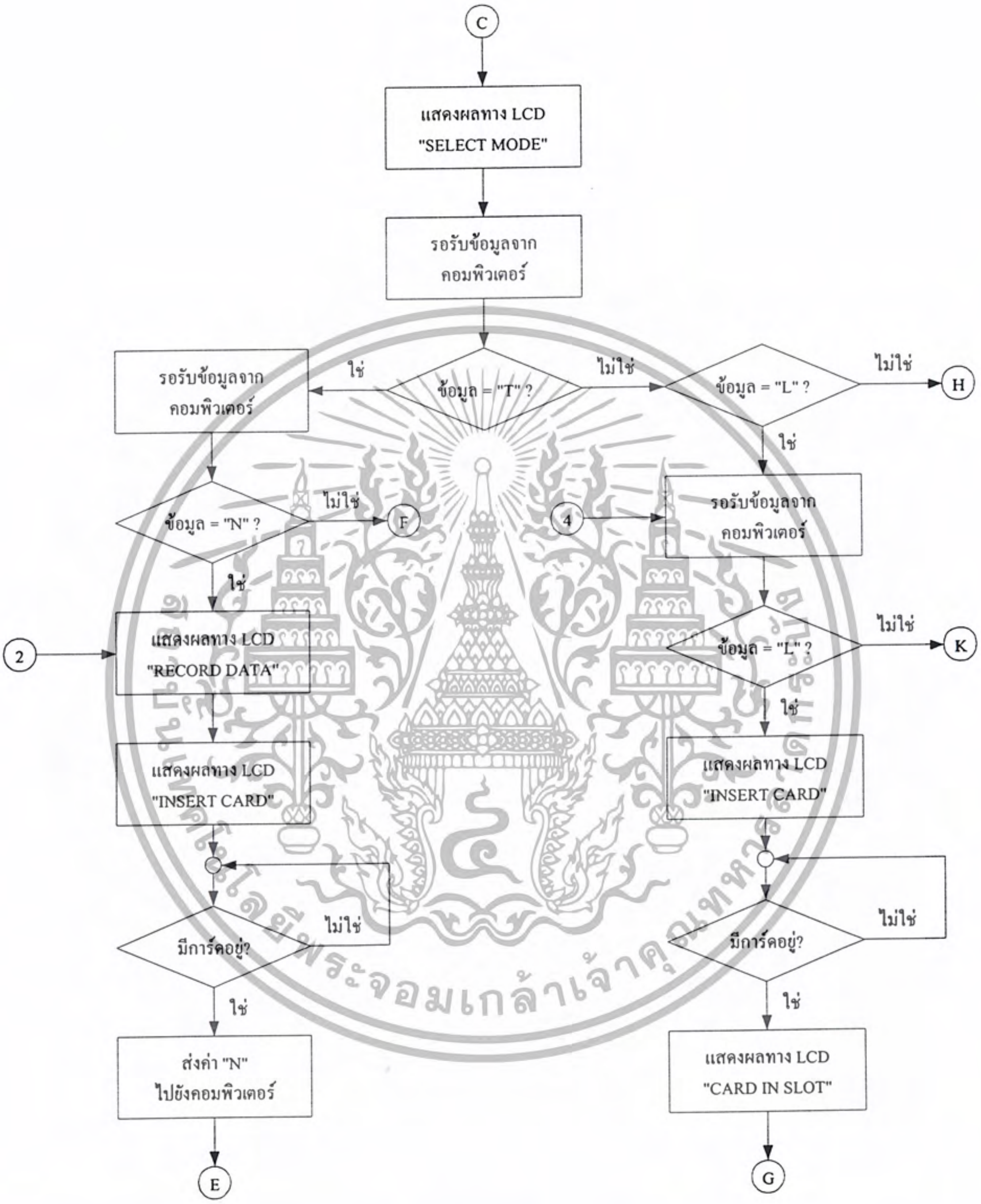
รูปที่ 3.4 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



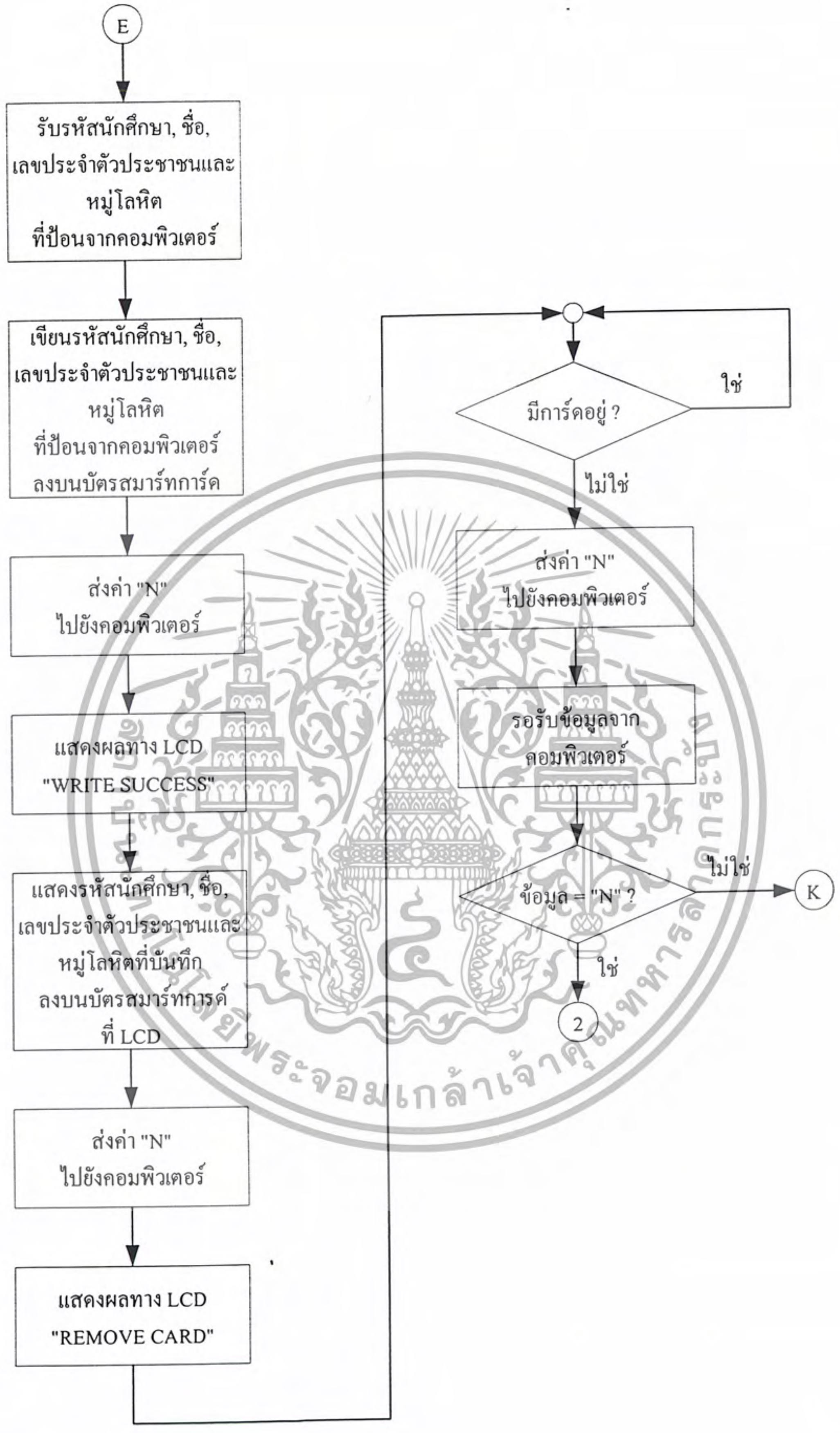
รูปที่ 3.5 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



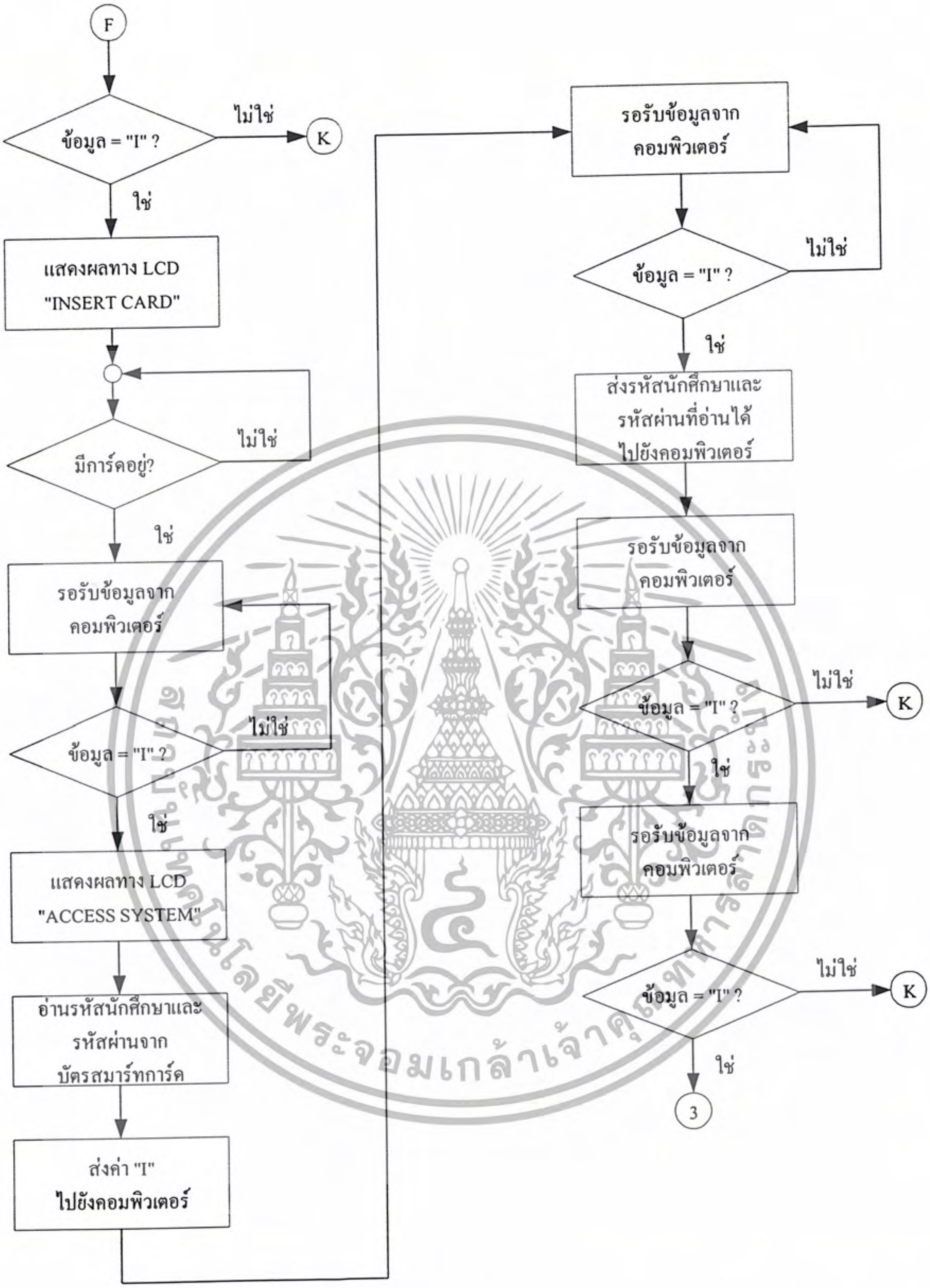
รูปที่ 3.6 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

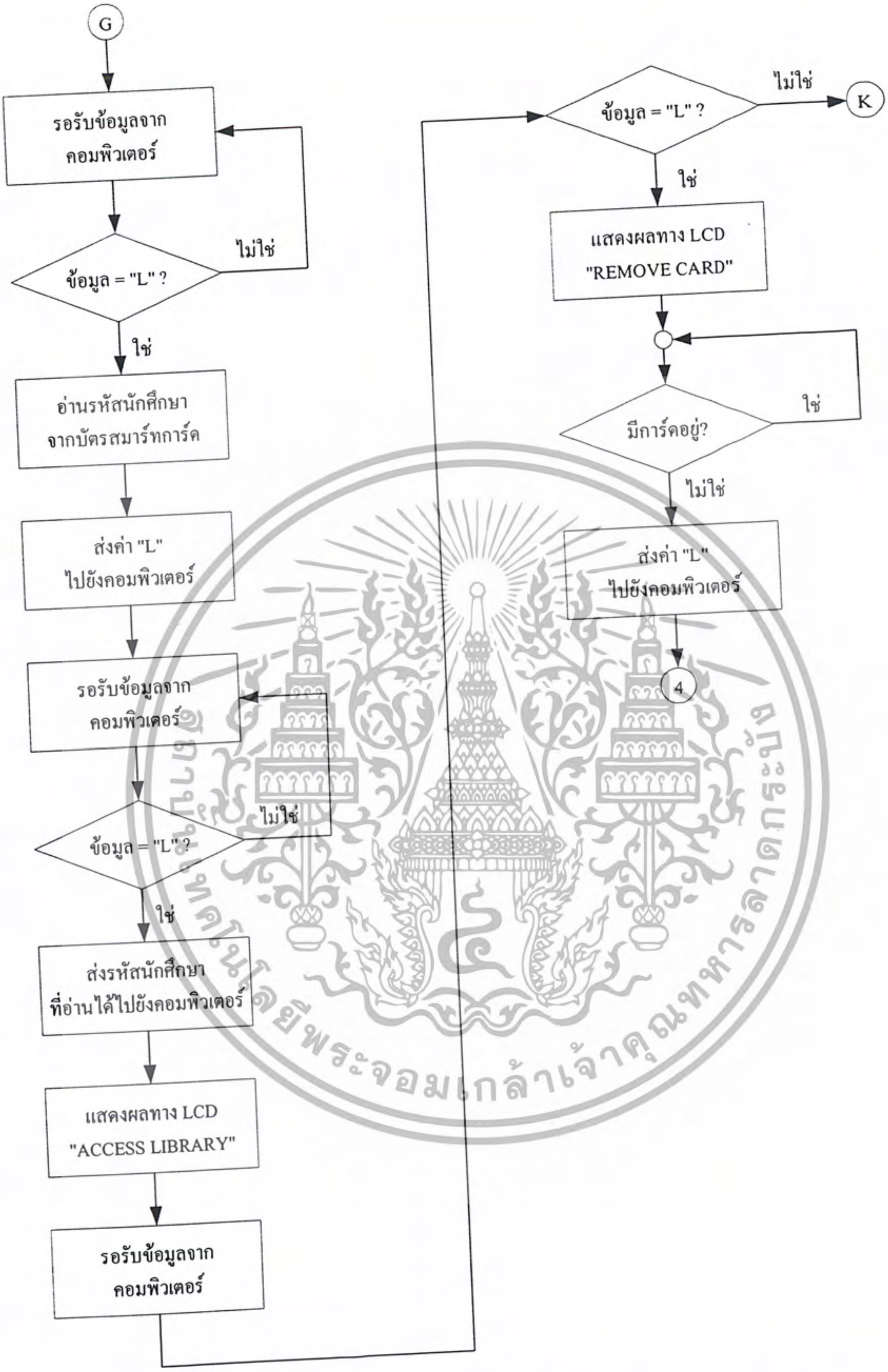


รูปที่ 3.8 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

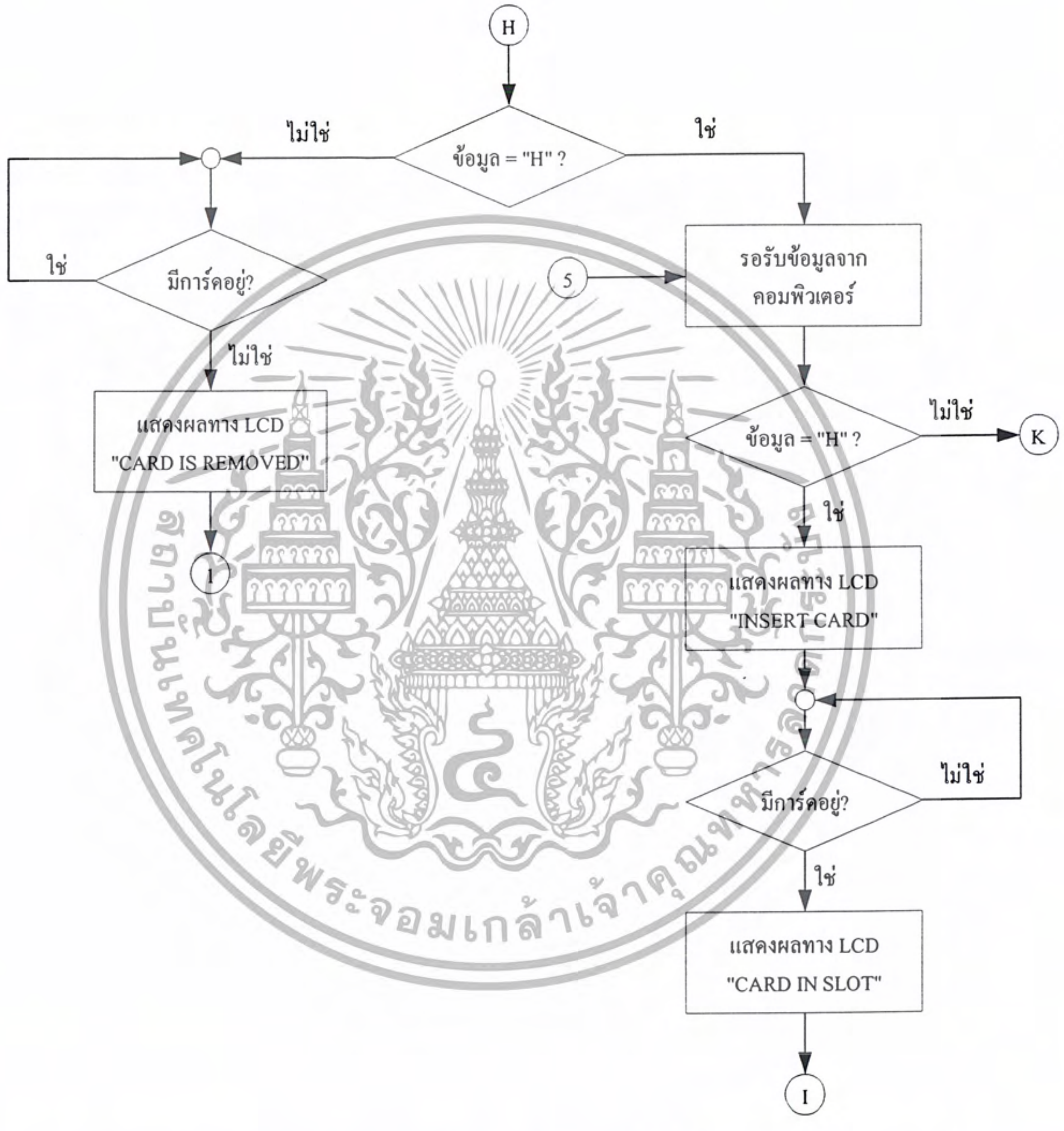
รูปที่ 3.9 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



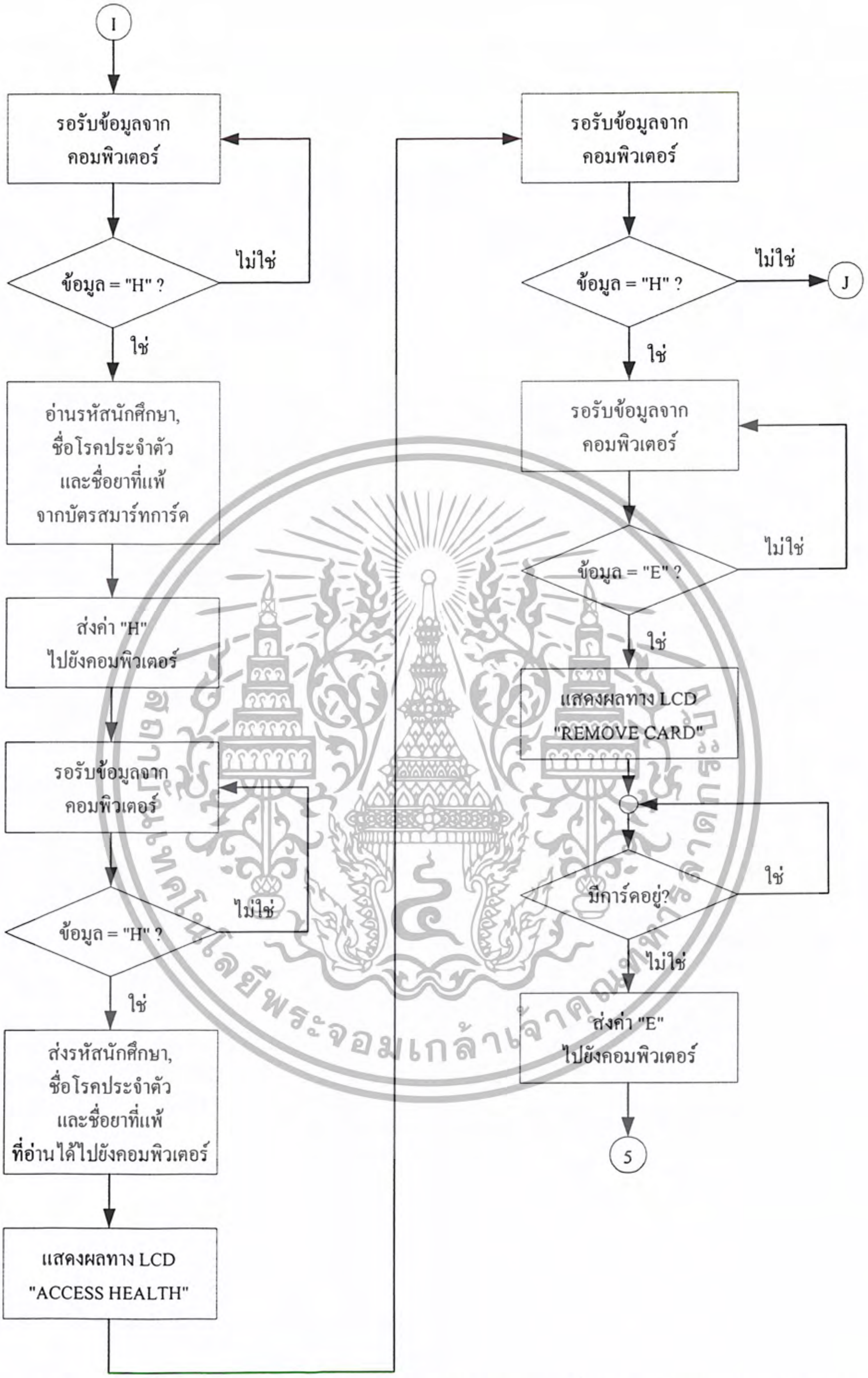
รูปที่ 3.10 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



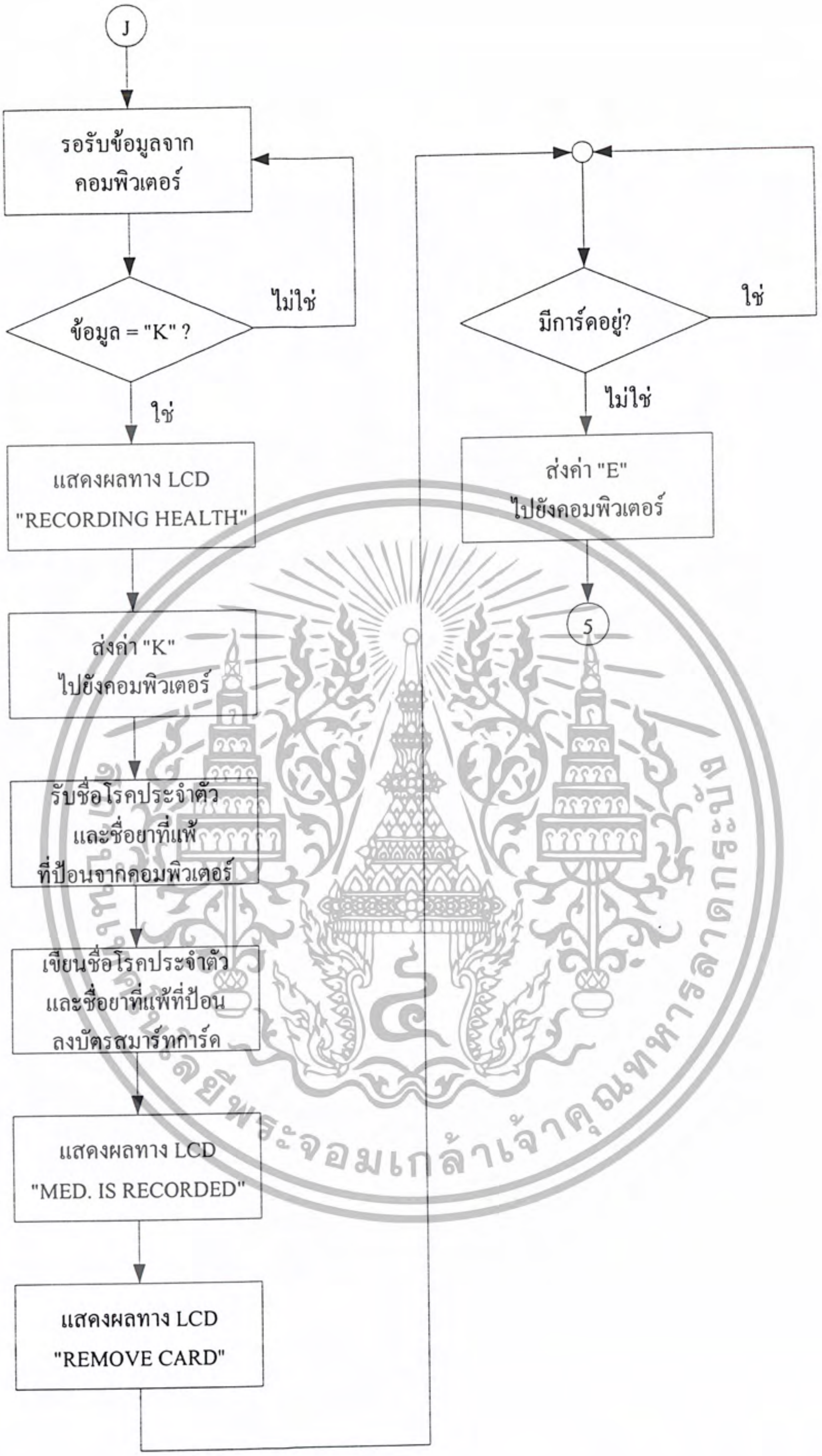
รูปที่ 3.11 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



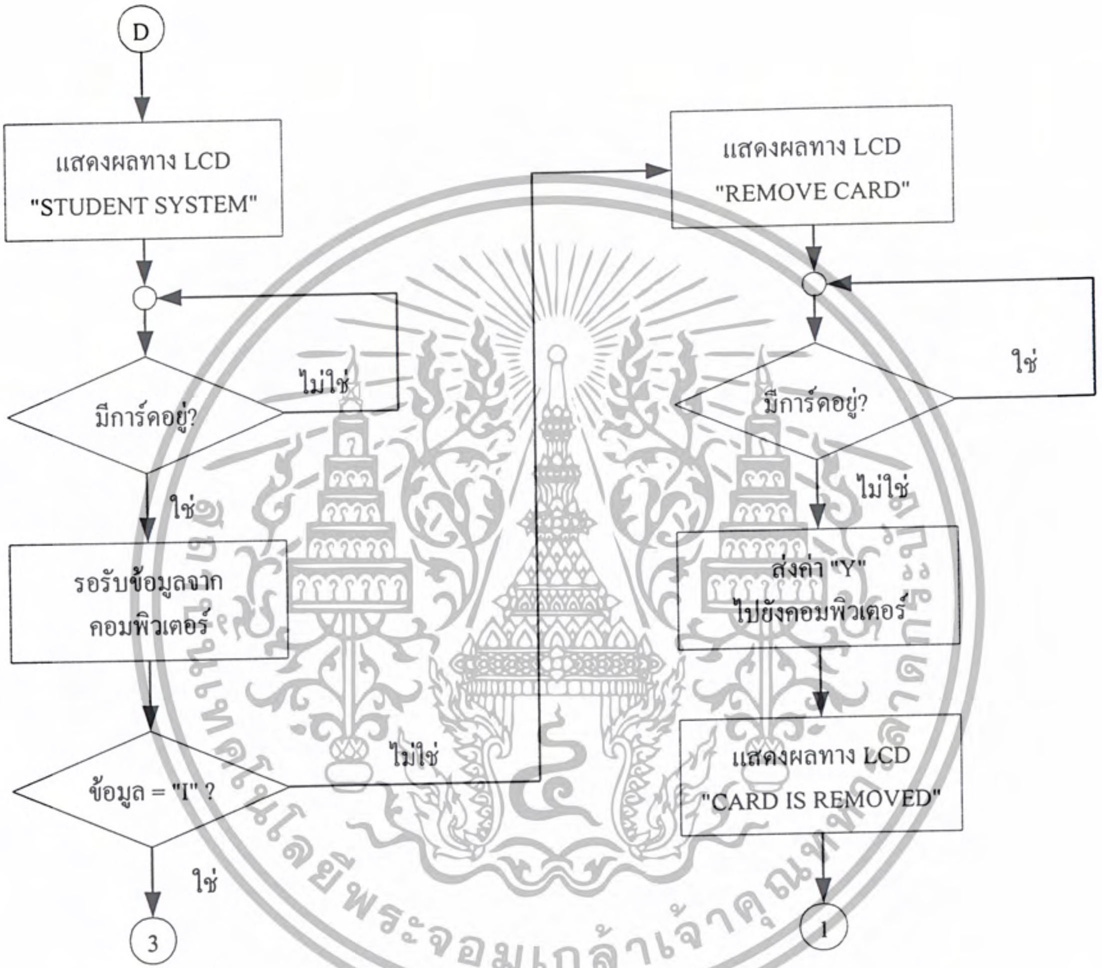
รูปที่ 3.12 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



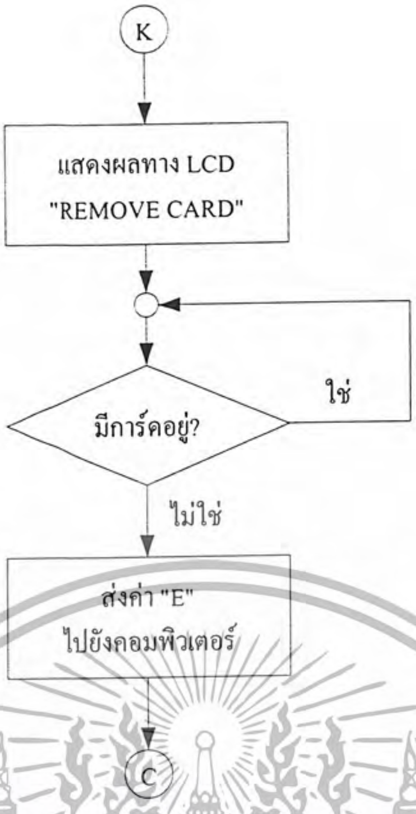
รูปที่ 3.13 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงโฟลวชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด (ต่อ)

จากรูปที่ 3.3 ถึงรูปที่ 3.15 เป็นโฟลว์ชาร์ทของโปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด ในตอนเริ่มต้นนั้น เราต้องทำการตั้งค่าการสื่อสารของพอร์ตอนุกรม ซึ่งจะต้องตั้งค่าภายในตัวไมโครคอนโทรลเลอร์ เพื่อให้สามารถทำการติดต่อสื่อสารกับคอมพิวเตอร์ได้ และเราต้องทำการตั้งค่าจอแสดงผลแบบผลึกเหลว (LCD) ให้ทำงานตามที่เราต้องการก่อน จากนั้นก็จะเข้าสู่สภาวะเตรียมพร้อม ในสภาวะเตรียมพร้อมนี้ที่จอแสดงผลแบบผลึกเหลวจะแสดงข้อความว่า "INSERT CARD" ซึ่งในขณะเดียวกันก็จะคอยตรวจสอบว่ามีบัตรสมาร์ทการ์ดเสียบเข้ามาที่ช่องเสียบบัตรหรือไม่ ถ้ายังไม่มีบัตรสมาร์ทการ์ดเสียบเข้ามา ที่จอแสดงผลแบบผลึกเหลวก็จะแสดงข้อความว่า "INSERT CARD" ค้างไว้ แต่ถ้าหากมีบัตรสมาร์ทการ์ดเสียบเข้ามา เครื่องอ่าน-เขียนสมาร์ทการ์ดจะทำการตรวจสอบว่าบัตรสมาร์ทการ์ดที่เสียบเข้ามานั้นเป็นบัตรสมาร์ทการ์ดที่ถูกต้องหรือไม่ ถ้าไม่ถูกต้องเครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงผลที่จอแสดงผลแบบผลึกเหลวว่า "WRONG CARD" และจะคอยตรวจสอบว่ายังมีบัตรเสียบอยู่หรือไม่ หากยังมีบัตรเสียบอยู่ก็จะแสดงข้อความ "WRONG CARD" ค้างไว้ แต่ถ้าตรวจพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะแสดงข้อความ "INSERT CARD" และเข้าสู่สภาวะเตรียมพร้อม แต่ถ้าบัตรที่ถูกเสียบเข้ามาถูกต้อง เครื่องอ่าน-เขียนสมาร์ทการ์ดก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "CARD IN SLOT" จากนั้นจะรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้นั้นเป็น "R" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "ACCESS SYSTEM" และจะทำการอ่านรหัสนักศึกษาและรหัสผ่านจากบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ทการ์ด และส่งข้อมูลที่อ่านได้ไปยังคอมพิวเตอร์ ต่อมาก็จะทำการรอรับข้อมูลจากคอมพิวเตอร์ ซึ่งข้อมูลที่จะได้รับในส่วนนี้ จะแบ่งออกเป็น

1. ข้อมูล "C" จะหมายถึงรหัสผ่านไม่ถูกต้อง ซึ่งหากได้รับข้อมูลนี้นั้น ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "WRONG PASSWORD" และ "REMOVE CARD" ตามลำดับ จากนั้นก็จะทำการตรวจสอบว่ายังคงมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากตรวจพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะทำการส่งข้อมูล "C" ไปยังคอมพิวเตอร์ และเข้าสู่สภาวะเตรียมพร้อม

2. ข้อมูล "D" จะหมายถึงรหัสผ่านถูกต้อง สำหรับกรณีของ Master card ซึ่งหากได้รับข้อมูลนี้นั้น ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" จากนั้นก็จะทำการตรวจสอบว่ายังคงมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากตรวจพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะทำการส่งข้อมูล "D" ไปยังคอมพิวเตอร์ และเข้าสู่สภาวะการเลือกฟังก์ชั่นการทำงาน ซึ่งในสภาวะนี้ที่หน้าจอแสดงผลแบบผลึกเหลวจะแสดงคำว่า "SELECT MODE" จากนั้นก็จะรอรับข้อมูลจากคอมพิวเตอร์ ซึ่งข้อมูลที่จะได้รับในส่วนนี้นั้น จะแบ่งออกเป็น

2.1 ข้อมูล "T" จะหมายถึงการเข้าสู่งานทะเบียน ซึ่งหากได้รับข้อมูลนี้นั้น เครื่องอ่าน-เขียนสมาร์ทการ์ดก็จะรอรับข้อมูลจากคอมพิวเตอร์ ซึ่งข้อมูลที่รับได้นั้นอาจเป็นไปได้ 2 กรณี คือ

2.1.1 ข้อมูล "N" จะหมายถึงการทำบัตรนักศึกษาใหม่ ซึ่งหากได้รับข้อมูลนี้นั้น ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "RECORD DATA" และ "INSERT CARD" ตามลำดับ จากนั้นก็จะทำการตรวจสอบว่ามีบัตรสมาร์ทการ์ดเสียบเข้ามาหรือไม่ หากมีบัตรเสียบเข้ามาแล้ว ก็จะทำการส่งข้อมูล "N" ไปยังคอมพิวเตอร์ ต่อมาก็จะทำการรับรหัสนักศึกษา ชื่อ เลขประจำตัวประชาชนและหมู่โลหิตที่ต้องการบันทึกลงบนบัตรสมาร์ทการ์ดจากคอมพิวเตอร์ แล้วทำการบันทึกข้อมูลดังกล่าวลงบนบัตร จากนั้นก็จะส่งข้อมูล "N" ไปยังคอมพิวเตอร์ แล้วทำการแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "WRITE SUCCESS" และแสดงรหัสนักศึกษา ชื่อ เลขประจำตัวประชาชน และหมู่โลหิตที่บันทึกลงบนบัตรสมาร์ทการ์ดที่หน้าจอแสดงผลแบบผลึกเหลว จากนั้นก็จะส่งข้อมูล "N" ไปยังคอมพิวเตอร์ และแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" ต่อมาก็จะทำการตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากตรวจสอบพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "N" ไปยังคอมพิวเตอร์ และรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้นั้นเป็น "N" ก็หมายความว่า ต้องการทำบัตรนักศึกษาใหม่ต่อไป เครื่องอ่าน-เขียนสมาร์ทการ์ดก็มีกระบวนการทำงานเช่นดังกล่าวข้างต้น แต่หากข้อมูลที่รับได้นั้นเป็น "E" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และทำการตรวจสอบว่ายังคงมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากไม่มีบัตรเสียบอยู่แล้ว ก็จะทำการส่งข้อมูล "E" ไปยังคอมพิวเตอร์ และกลับสู่สภาวะการเลือกฟังก์ชั่นการทำงาน

2.1.2 ข้อมูล "I" จะหมายถึงการเปลี่ยนรหัสผ่าน ซึ่งหากได้รับข้อมูลนี้ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "INSERT CARD" และทำการตรวจสอบว่ามีบัตรสมาร์ทการ์ดเสียบเข้ามาหรือยัง ซึ่งหากตรวจสอบพบว่ามีบัตรเสียบเข้ามาแล้ว ก็จะทำการแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "CARD IN SLOT" ต่อจากนั้นก็จะรอรับข้อมูลจากคอมพิวเตอร์ ถ้าข้อมูลรับได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น "I" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "ACCESS SYSTEM" และทำการอ่านรหัสนักศึกษาและรหัสผ่านจากบัตรสมาร์ทการ์ด จากนั้นก็จะส่งข้อมูลที่อ่านได้ดังกล่าวไปยังคอมพิวเตอร์ และรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "E" ซึ่งหมายความว่ารหัสผ่านไม่ถูกต้อง ที่หน้าจอแสดงผลแบบผลึกเหลวจะแสดงข้อความว่า "REMOVE CARD" และตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ และกลับไปสู่สภาวะการเลือกฟังก์ชั่นการทำงาน แต่หากข้อมูลที่รับได้เป็น "I" ซึ่งหมายความว่ารหัสผ่านถูกต้อง ก็จะทำการรอรับข้อมูลจากคอมพิวเตอร์อีกครั้ง ซึ่งหากข้อมูลที่รับได้เป็น "I" อีก ก็จะทำการส่งข้อมูล "I" กลับไปยังคอมพิวเตอร์ จากนั้นก็จะรับรหัสผ่านที่ป้อนจากคอมพิวเตอร์ และทำการบันทึกรหัสผ่านที่ป้อนนี้ลงบนบัตรสมาร์ทการ์ด แล้วแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "P.W. IS RECORDED" จากนั้นก็จะส่งข้อมูล "I" ไปยังคอมพิวเตอร์ และรอข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "I" ซึ่งหมายความว่าต้องการบันทึกรหัสผ่านอีกครั้ง เครื่องอ่าน-เขียนสมาร์ทการ์ดก็จะมีกระบวนการทำงานเช่นดังกล่าวข้างต้น แต่หากข้อมูลที่ได้รับเป็น "E" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และ ตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ และกลับไปสู่สภาวะการเลือกฟังก์ชั่นการทำงาน

2.2 ข้อมูล "L" จะหมายถึงการเข้าสู่ระบบยืม-คืนหนังสือห้องสมุด ซึ่งหากได้รับข้อมูลนี้ก็จะทำการรอรับข้อมูลจากคอมพิวเตอร์อีกครั้ง ซึ่งหากข้อมูลที่ได้รับเป็น "E" จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และ ตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ และกลับไปสู่สภาวะการเลือกฟังก์ชั่นการทำงาน แต่หากข้อมูลที่ได้รับนั้นยังคงเป็น "L" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "INSERT CARD" และทำการตรวจสอบว่ามีบัตรสมาร์ทการ์ดเสียบเข้ามาหรือไม่ หากตรวสอบพบว่ามีบัตรเสียบเข้ามาแล้ว ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "CARD IN SLOT" จากนั้นก็จะรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "L" ก็จะทำการอ่านรหัสนักศึกษาจากบัตรสมาร์ทการ์ด และส่งข้อมูล "L" ไปยังคอมพิวเตอร์ และรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "L" ก็จะทำการส่งรหัสนักศึกษาที่อ่านได้ไปยังคอมพิวเตอร์ และแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "ACCESS LIBRARY" จากนั้นก็จะรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลรับได้นั้นเป็น "E"จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และ ตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ และกลับไปสู่สภาวะการเลือกฟังก์ชั่นการทำงาน แต่หากข้อมูลที่รับได้นั้นเป็น "L" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และ ตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะทำการส่งข้อมูล "L" ไปยังคอมพิวเตอร์แล้วเข้าสู่ตอนต้นของการทำงานของระบบยืม-คืนหนังสือห้องสมุดอีกครั้ง

2.3 ข้อมูล "H" จะหมายถึงการเข้าสู่ระบบข้อมูลสุขภาพนักศึกษา ซึ่งหากได้รับข้อมูลนี้นั้น ก็จะทำการรอรับข้อมูลจากคอมพิวเตอร์อีกครั้ง และหากข้อมูลที่ได้รับเป็น "E" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และ ตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ และกลับไปสู่สภาวะการเลือกฟังก์ชั่นการทำงาน แต่หากข้อมูลที่ได้รับเป็น "H" ก็จะทำการแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "INSERT CARD" และทำการตรวจสอบว่ามีบัตรสมาร์ทการ์ดเสียบเข้ามาหรือไม่ หากตรวสอบพบว่ามีบัตรเสียบเข้ามาแล้ว ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "CARD IN SLOT" จากนั้นก็จะรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "H" ก็จะทำการอ่านรหัสนักศึกษา ชื่อโรคประจำตัวและชื่อยาที่แพ้จากบัตรสมาร์ทการ์ด จากนั้นก็ส่งข้อมูล "H" ไปยังคอมพิวเตอร์และรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "H" ก็จะทำการส่งข้อมูลที่อ่านได้จากบัตรไปยังคอมพิวเตอร์ และแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "ACCESS HEALTH" จากนั้นก็จะรอรับข้อมูลจากคอมพิวเตอร์

- หากข้อมูลที่ได้รับเป็น "H" ก็จะรอรับข้อมูลจากคอมพิวเตอร์อีกครั้ง หากข้อมูลที่ได้รับเป็น "E" จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ แล้วเข้าสู่ตอนต้นของการทำงานของระบบข้อมูลสุขภาพนักศึกษาอีกครั้ง

- หากข้อมูลที่ได้รับเป็น "K" ก็จะรอรับข้อมูลจากคอมพิวเตอร์อีกครั้ง หากข้อมูลที่ได้รับยังคงเป็น "K" อยู่ ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "RECORDING HEALTH" แล้วทำการส่งค่า "K" ไปยังคอมพิวเตอร์ จากนั้นก็จะรับชื่อโรคประจำตัวและชื่อยาที่แพ้ที่ป้อนจากคอมพิวเตอร์ แล้วทำการบันทึกข้อมูลดังกล่าวลงบนบัตรสมาร์ทการ์ด และแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "MED. IS RECORDED" และ "REMOVE CARD" ตามลำดับ จากนั้นจะทำการตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะส่งข้อมูล "E" ไปยังคอมพิวเตอร์ แล้วเข้าสู่ตอนต้นของการทำงานของระบบข้อมูลสุขภาพนักศึกษาอีกครั้ง

3. ข้อมูล "Z" จะหมายถึงรหัสผ่านถูกต้อง สำหรับกรณีของบัตรนักศึกษา ซึ่งหากได้รับข้อมูลนี้นั้น ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "STUDENT SYSTEM" และทำการตรวจสอบว่ามีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากตรวจสอบพบว่ามีบัตรเสียบอยู่ ก็จะรอรับข้อมูลจากคอมพิวเตอร์ หากข้อมูลที่รับได้เป็น "I" กระบวนการทำงานต่อไปก็จะเป็นเช่นเดียวกับกระบวนการเปลี่ยนรหัสผ่านดังที่ได้กล่าวมาแล้ว แต่หากข้อมูลที่ได้รับเป็น "E" ก็จะแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "REMOVE CARD" และ ตรวจสอบว่ายังมีบัตรสมาร์ทการ์ดเสียบอยู่หรือไม่ หากพบว่าไม่มีบัตรเสียบอยู่แล้ว ก็จะทำการส่งข้อมูล "Y" ไปยังคอมพิวเตอร์ และแสดงผลที่หน้าจอแสดงผลแบบผลึกเหลวว่า "CARD IS REMOVED" แล้วกลับเข้าสู่สภาวะเตรียมพร้อม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 โปรแกรมการกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว

รูปที่ 3.16 แสดงโฟลวชาร์ทของโปรแกรมกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว

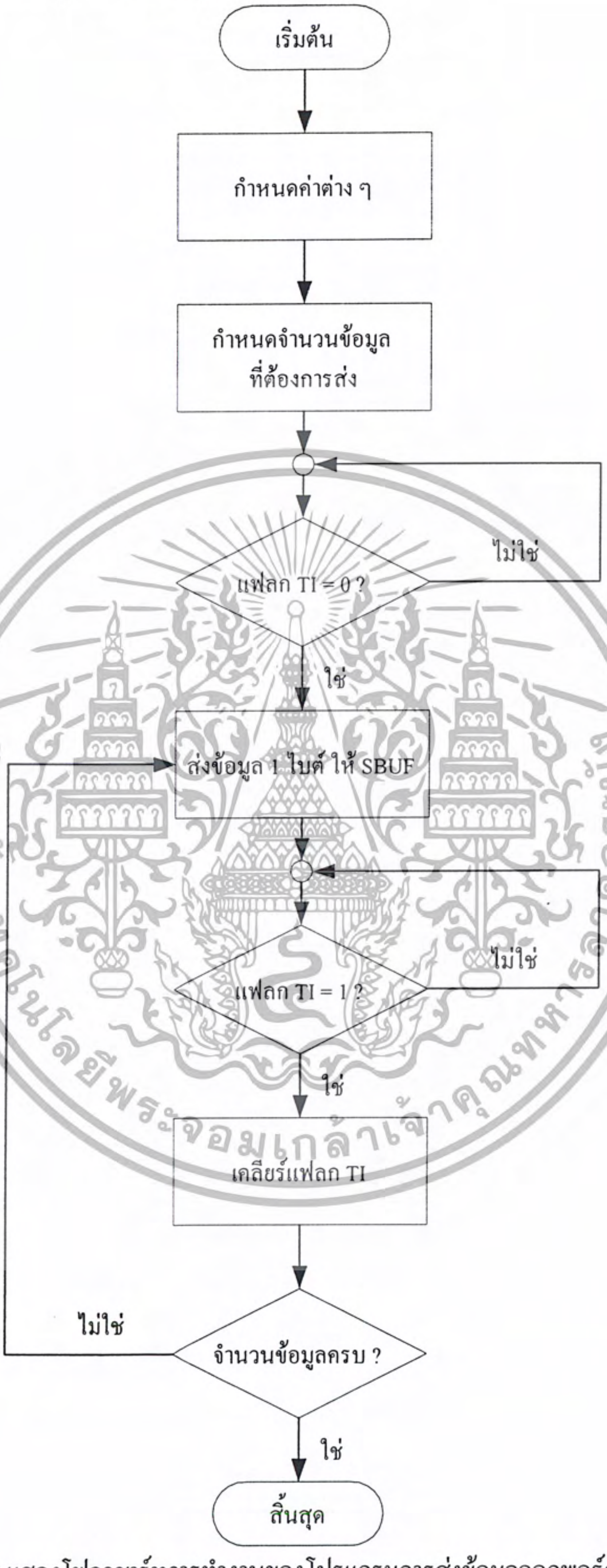
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่จะนำจอแสดงผลแบบผลึกเหลวมาใช้งานนั้น เราจำเป็นที่จะต้องตั้งค่าต่าง ๆ ของจอแสดงผลแบบผลึกเหลว เพื่อให้จอแสดงผลแบบผลึกเหลวทำงานตามที่ต้องการ จากรูปที่ 3.16 เป็นโฟลวชาร์ทของโปรแกรมกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว ในตอนเริ่มต้นหลังจากจ่ายไฟเลี้ยงให้จอแสดงผลแบบผลึกเหลว ต้องทำการหน่วงเวลาเพื่อให้จอแสดงผลแบบผลึกเหลวเตรียมพร้อม เนื่องจากว่าจอแสดงผลแบบผลึกเหลวเป็นอุปกรณ์ที่ทำงานช้าเมื่อเทียบกับไมโครคอนโทรลเลอร์ จากนั้นเคลียร์สถานะที่ขา RS ให้เป็นลอจิก "0" เพื่อบอกให้จอแสดงผลแบบผลึกเหลวทราบว่าข้อมูลที่เข้ามาเป็นข้อมูลควบคุม จากนั้นก็ตั้งค่าให้เป็นการรับ-ส่งข้อมูลแบบ 8 บิต และเป็นจอแสดงผลแบบผลึกเหลวแบบ 1 บรรทัด จากนั้นก็ทำการอีนาเบิลจอแสดงผลแบบผลึกเหลวโดยการส่งพัลส์ไปยังขา E ของจอแสดงผลแบบผลึกเหลวและทำการหน่วงเวลาเพื่อให้จอแสดงผลแบบผลึกเหลวทำงานเสร็จเรียบร้อย จากนั้นทำการเคลียร์ค่าในทุก ๆ ตำแหน่งแอดเดรสของจอแสดงผลแบบผลึกเหลวให้มีค่าเป็นศูนย์ เมื่อเสร็จแล้วให้แอดเดรสไปเริ่มต้นที่ตำแหน่งแรก จากนั้นก็สามารถเริ่มต้นการแสดงผลทางจอแสดงผลแบบผลึกเหลวได้ตามต้องการ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 โปรแกรมการส่งข้อมูลออกพอร์ตอนุกรม



รูปที่ 3.17 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการส่งข้อมูลออกพอร์ตอนุกรม

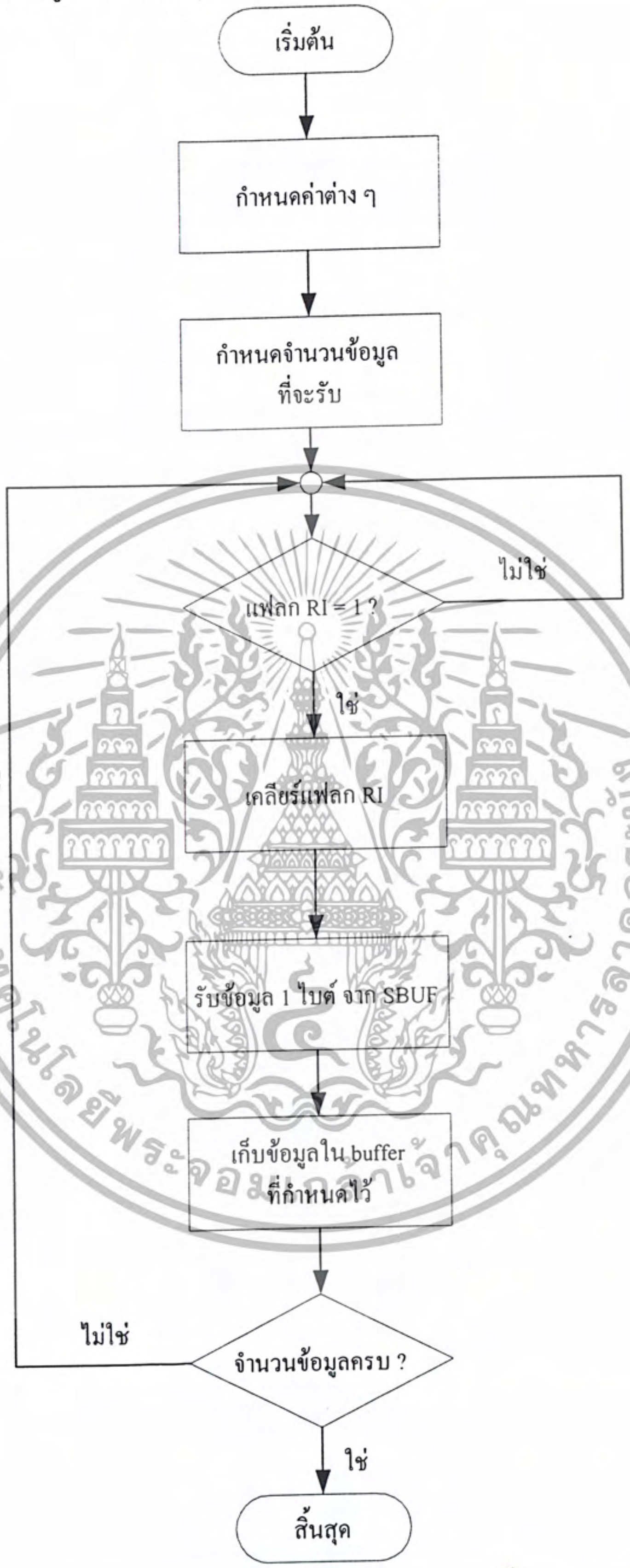
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.17 เป็นโฟลว์ชาร์ทการทำงานของโปรแกรมการส่งข้อมูลออกพอร์ตอนุกรมแบบไม่ใช้การอินเตอร์รัปต์ เราจะควบคุมไม่ให้พอร์ตอนุกรมส่งสัญญาณอินเตอร์รัปต์ (กำหนด ES = 0) แต่จะให้ไมโครคอนโทรลเลอร์คอยตรวจสอบแฟลก TI แทน ในกรณีของการส่งข้อมูล หลังจากกำหนดโหมดการทำงานของพอร์ตอนุกรมแล้ว เมื่อต้องการส่งข้อมูลเราต้องกำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจสอบแฟลก TI หากแฟลก TI มีค่าเป็น 0 จึงจะเริ่มส่งข้อมูล โดยกำหนดข้อมูลที่ต้องการส่งในรีจีสเตอร์ SBUF หลังจากส่งข้อมูลให้กับ SBUF แล้ว พอร์ตอนุกรมจะเริ่มส่งข้อมูลออกไปทางขา TxD ในขณะนี้สัญญาณที่แฟลก TI ยังคงเป็น 0 อยู่ และจะเปลี่ยนเป็น 1 เมื่อข้อมูลได้ส่งออกไปหมดทุกบิตแล้ว เราจะทำการเคลียร์แฟลก TI แล้วทำการส่งข้อมูลต่อไปอีก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 โปรแกรมการรับข้อมูลจากพอร์ตอนุกรม



รูปที่ 3.18 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการรับข้อมูลจากพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.18 เป็นโฟลว์ชาร์ทการทำงานของโปรแกรมการรับข้อมูลจากพอร์ตอนุกรมแบบไม่ใช้การอินเตอร์รัปต์ เราจะควบคุมไม่ให้พอร์ตอนุกรมส่งสัญญาณอินเตอร์รัปต์ (กำหนด ES = 0) แต่จะให้ไมโครคอนโทรลเลอร์คอยตรวจสอบแฟลก RI แทน ในกรณีของการรับข้อมูล หลังจากกำหนดโหมดการทำงานของพอร์ตอนุกรมแล้ว เมื่อต้องการรับข้อมูลเราต้องกำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจสอบแฟลก RI หากแฟลก RI มีค่าเป็น 1 ก็จะทำการเคลียร์แฟลก RI ให้มีค่าเป็น 0 จึงจะเริ่มรับข้อมูลโดยจะรับข้อมูลที่เข้ามาทางขา RxD มาเก็บไว้ในรีจีสเตอร์ SBUF ในขณะนี้สัญญาณที่แฟลก RI ยังคงเป็น 0 อยู่ และจะเปลี่ยนเป็น 1 เมื่อข้อมูลได้รับข้อมูลครบทุกบิตแล้ว

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 โปรแกรมการอ่านข้อมูลจากบัตรสมาร์ทการ์ดและแสดงผลทางจอแสดงผลแบบผลึกเหลว

รูปที่ 3.19 แสดงโฟลวชาร์ทของโปรแกรมการอ่านข้อมูลจากบัตรสมาร์ทการ์ด และแสดงผลทางจอแสดงผลแบบผลึกเหลว

จากรูปที่ 3.19 เป็นโฟลวชาร์ทของโปรแกรมการอ่านข้อมูลจากบัตรสมาร์ทการ์ดและแสดงผลทางจอแสดงผลแบบผลึกเหลว ในสภาวะแรกต้องทำการส่งข้อมูลซึ่งเป็นรหัสคำสั่งเพื่อบอกให้สมาร์ทการ์ดทราบว่าต้องการอ่านข้อมูลจากหน่วยความจำหลัก จากนั้นจะต้องส่งค่าแอดเดรสที่เราต้องการอ่าน

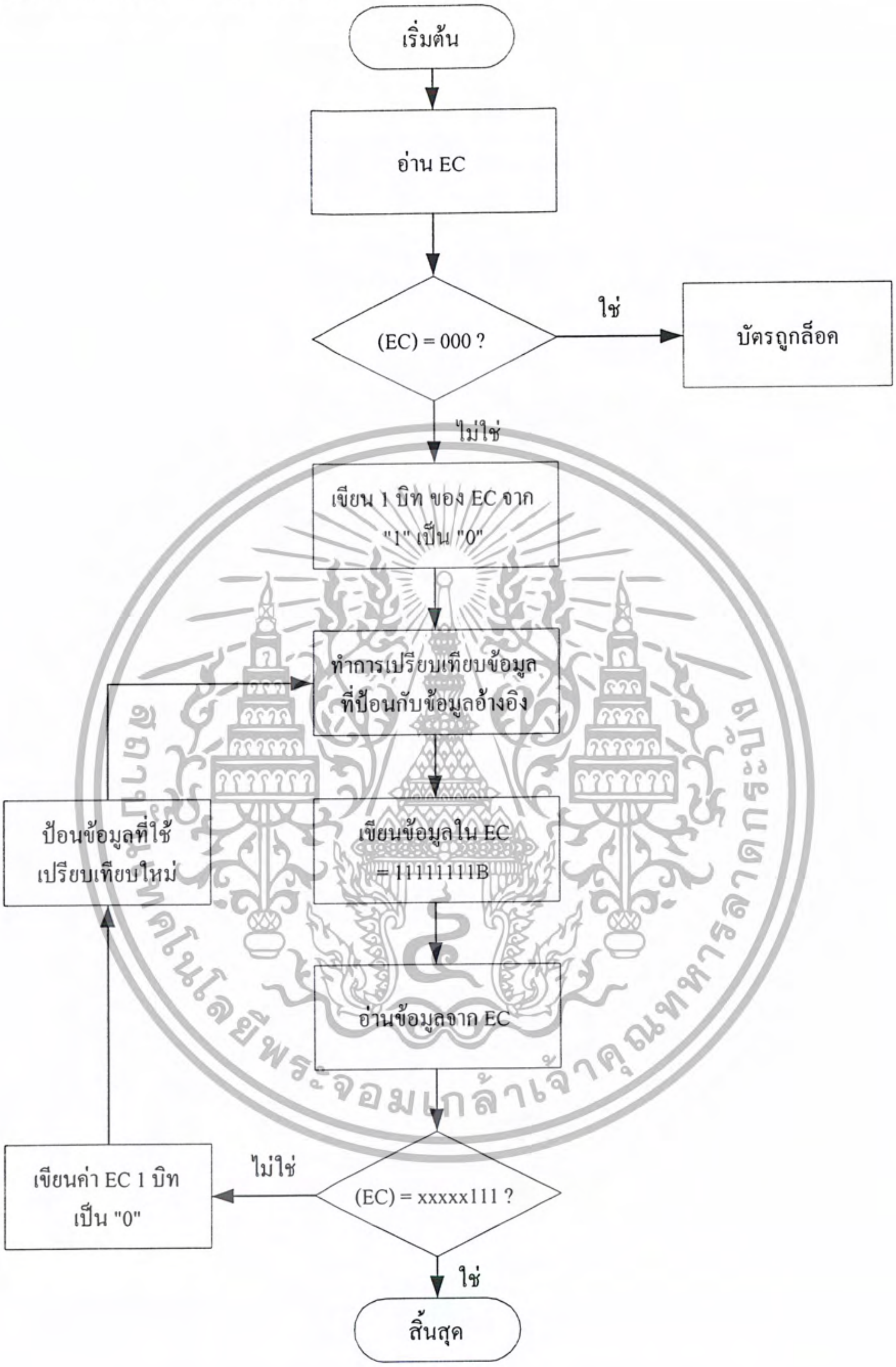
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลไปด้วย ซึ่งการอ่านข้อมูลนั้นจะสามารถอ่านได้มากน้อยเพียงไหนขึ้นอยู่กับจำนวนสัญญาณนาฬิกา โดยที่ 1 แอดเดรสจะใช้สัญญาณนาฬิกา 8 ลูก จากนั้นจะทำการอ่านข้อมูลจากบัตรแล้วนำมาเก็บไว้ใน buffer ที่กำหนดไว้ และทำการส่งข้อมูลที่อ่านได้นี้ไปยังคอมพิวเตอร์ และแสดงผลที่จอแสดงผลแบบผลึกเหลวต่อไป

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 โปรแกรมการเปรียบเทียบรหัสผ่าน



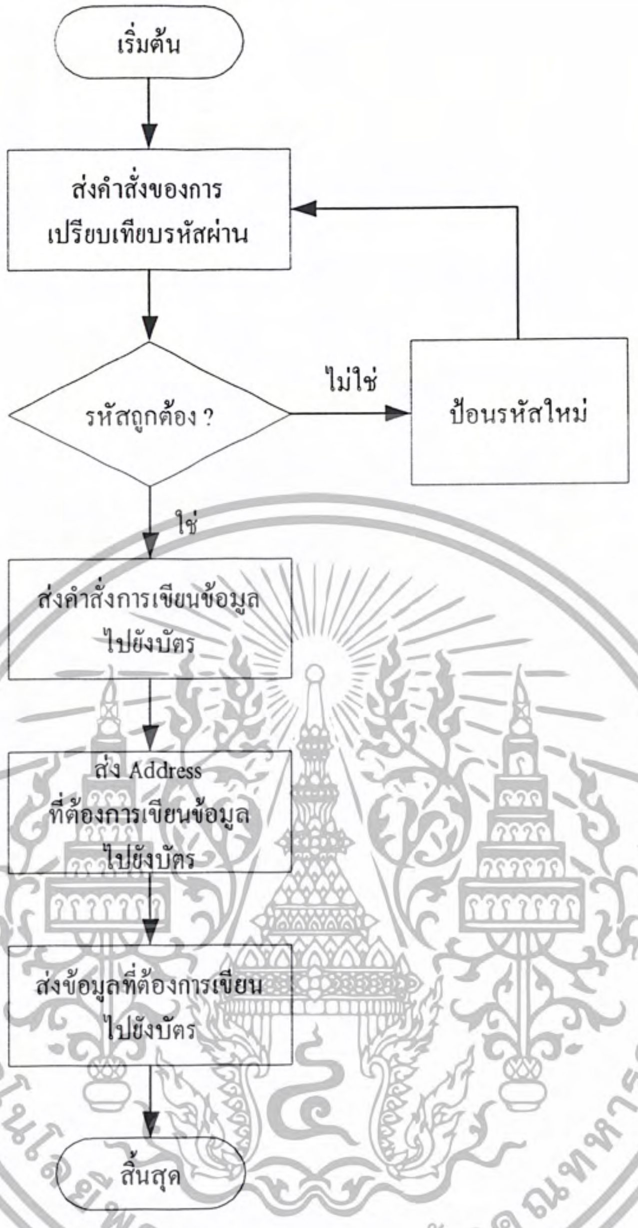
รูปที่ 3.20 แสดงโฟลวชาร์ทของโปรแกรมการเปรียบเทียบรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.20 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการเปรียบเทียบรหัสผ่านเพื่อใช้ในการเขียนข้อมูลลงบัตรสมาร์ทการ์ด ในสภาวะแรกต้องทำการอ่านข้อมูลในหน่วยความจำที่เรียกว่า Error Counter (EC) ซึ่งค่า Error Counter นี้จะเป็นตัวบอกจำนวนครั้งที่ยังสามารถป้อนรหัสผิดได้ เมื่ออ่านค่า Error Counter ออกมาแล้วก็จะทำการเปรียบเทียบค่า Error Counter ว่าเท่ากับ xxxxx000 หรือไม่ ถ้าเท่าแสดงว่าบัตรสมาร์ทการ์ดใบนั้นถูกล็อกทำให้ไม่สามารถที่จะเขียนข้อมูลไปยังบัตรสมาร์ทการ์ดนั้นได้อีก ซึ่งตามรูปที่ 3.10 ถ้าบัตรสมาร์ทการ์ดนั้นถูกล็อกก็จะสิ้นสุดการเปรียบเทียบทันที แต่ถ้าค่าใน Error Counter ไม่เท่ากับ xxxxx000 แสดงว่ายังสามารถเขียนข้อมูลไปยังบัตรสมาร์ทการ์ดใบนั้นได้ จากนั้นจะทำการเขียนข้อมูล 1 บิตของ Error Counter ให้จากลอจิก “1” เป็น “0” โดยเรียงจากบิตที่มีนัยสำคัญต่ำสุดก่อน จากนั้นจะทำการเปรียบเทียบข้อมูลอ้างอิงที่อยู่ภายในบัตรกับข้อมูลที่เราป้อน จากนั้นทำการเขียนข้อมูลลงใน Error Counter เท่ากับ 11111111B แล้วทำการอ่านข้อมูลใน Error Counter มาอีกครั้ง ถ้าการเปรียบเทียบถูกต้องค่า Error Counter ที่อ่านออกมาเที่ยวหลังนี้จะต้องมีค่าเป็น xxxxx111B แต่ถ้ารหัสที่ป้อนผิด ข้อมูลของ Error Counter จะมีค่าเป็นค่าเดิมคือ xxxxx110B ซึ่งจะเห็นว่าบิตที่ 0 ถูกทำให้เป็น “0” ถ้าการเปรียบเทียบครั้งที่ 2 ยังคงผิดอีก บิตที่จะถูกทำให้เป็น “0” คือบิตที่ 1 เพราะฉะนั้นค่า Error Counter ที่อ่านออกมาจะมีค่าเป็น xxxxx100B ถ้าป้อนผิดครบ 3 ครั้ง บัตรสมาร์ทการ์ดจะถูกล็อกทันที แต่ถ้ารหัสถูกต้องก็จะทำการเขียนค่าใน Error Counter ให้เป็น xxxxx111B เหมือนเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.6 โปรแกรมการเขียนข้อมูลลงบัตรสมาร์ทการ์ด



รูปที่ 3.21 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการเขียนข้อมูลลงบัตรสมาร์ทการ์ด

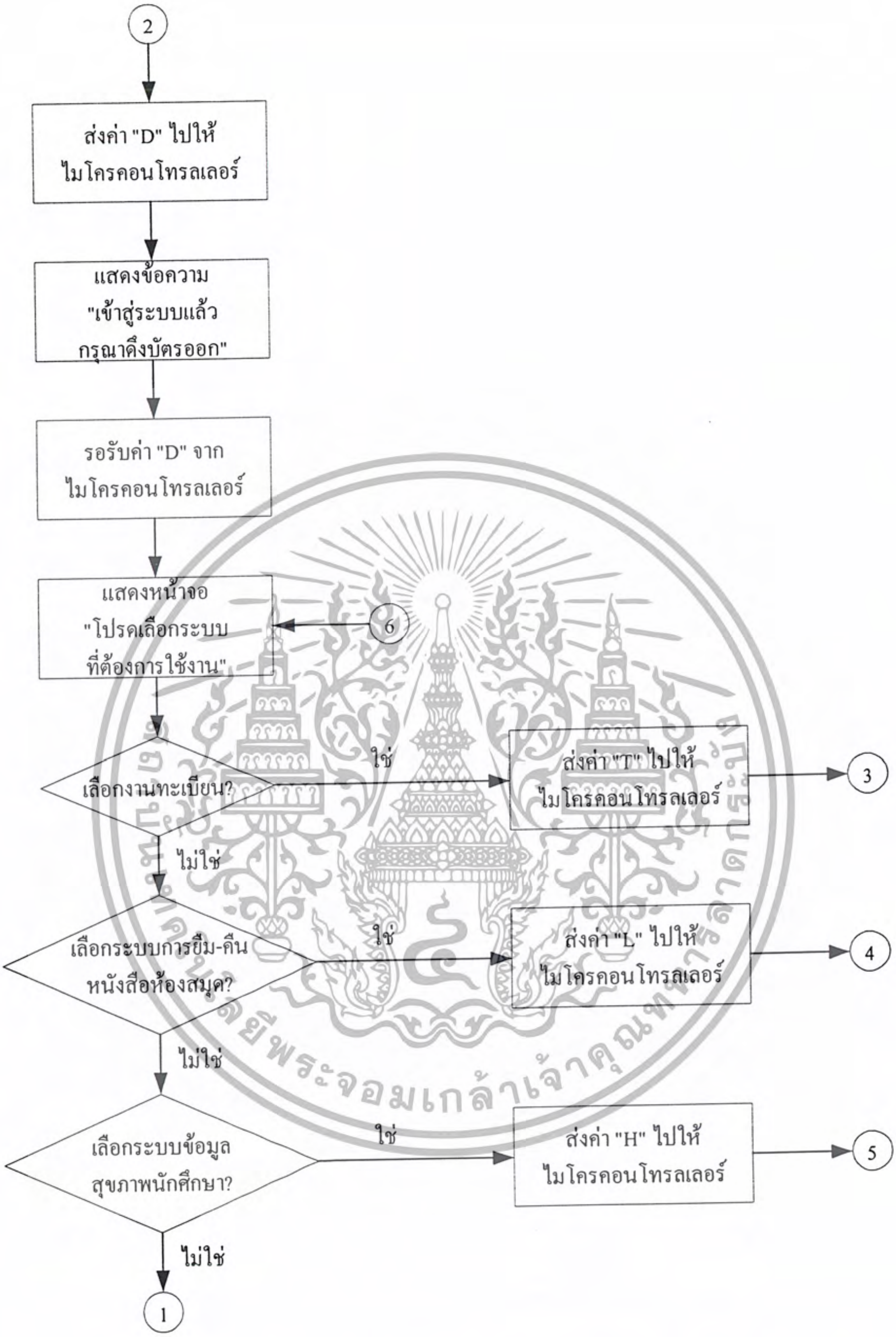
จากรูปที่ 3.21 แสดงโฟลวชาร์ทการทำงานของโปรแกรมการเขียนข้อมูลลงบัตรสมาร์ทการ์ด ในสภาวะแรกเราต้องทำการเปรียบเทียบรหัสผ่าน ถ้ารหัสผ่านถูกต้อง ไมโครคอนโทรลเลอร์ก็จะส่งคำสั่งการเขียนข้อมูลไปยังบัตรสมาร์ทการ์ด จากนั้นจะส่งค่าแอดเดรสที่ต้องการเขียนข้อมูลไปยังบัตร และทำการส่งข้อมูลที่ต้องการเขียนตามไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โปรแกรมควบคุมคอมพิวเตอร์

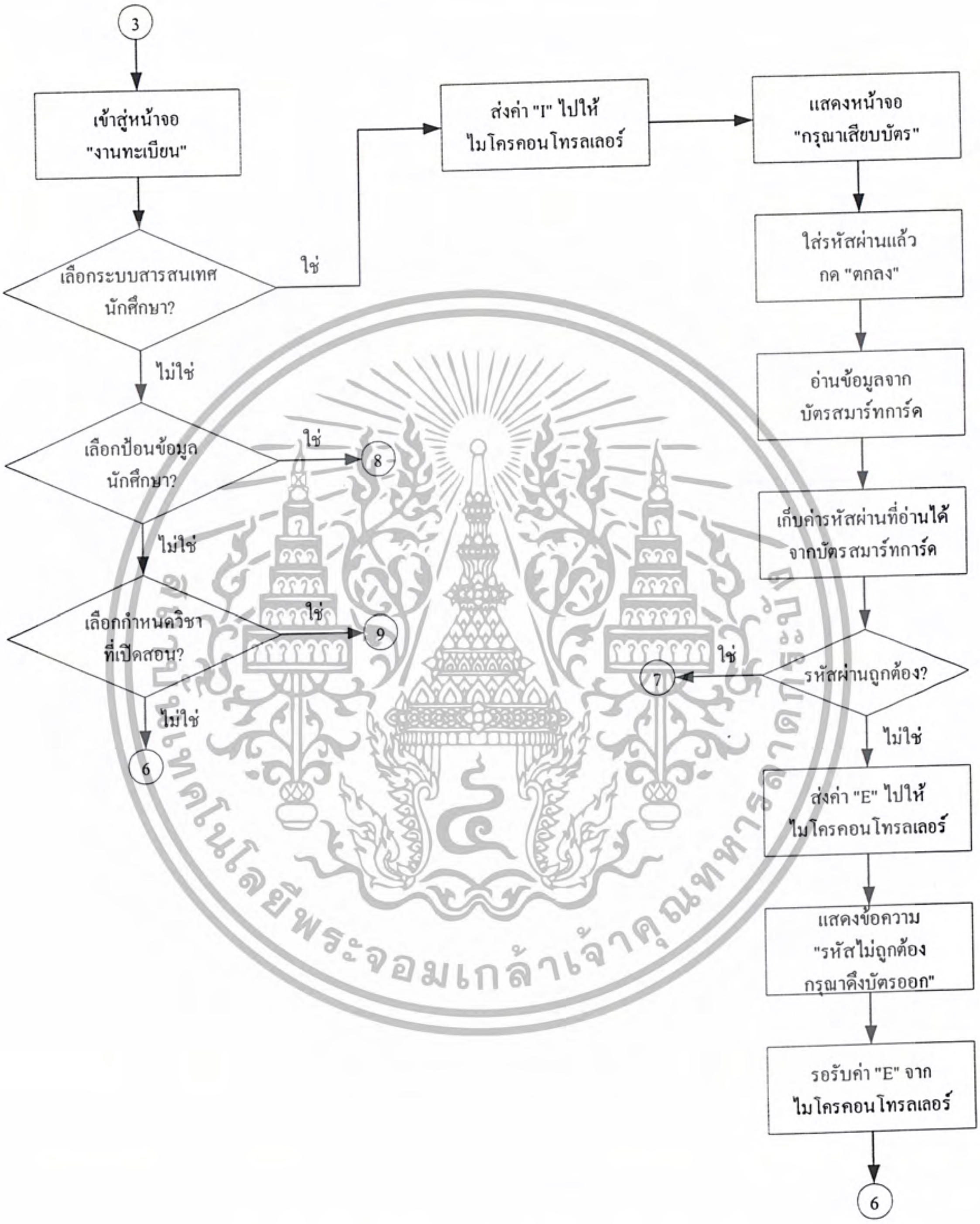
รูปที่ 3.22 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

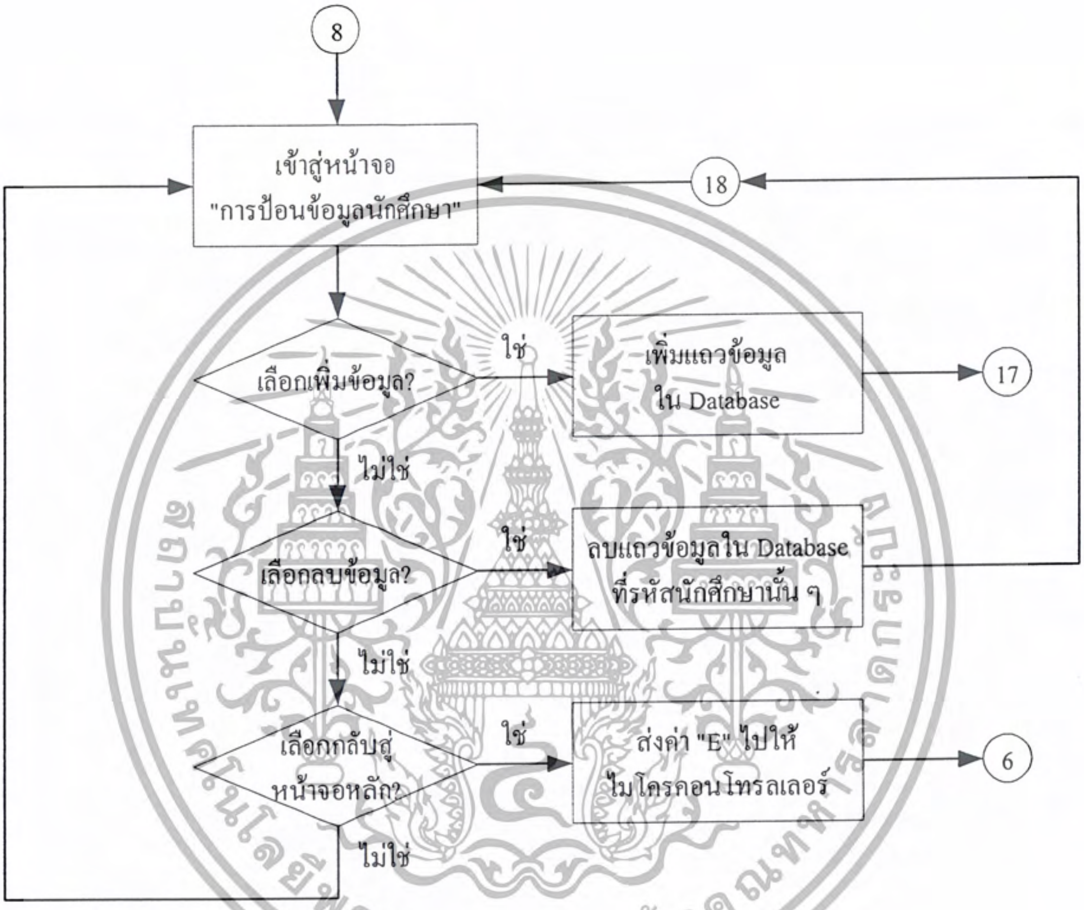


รูปที่ 3.24 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

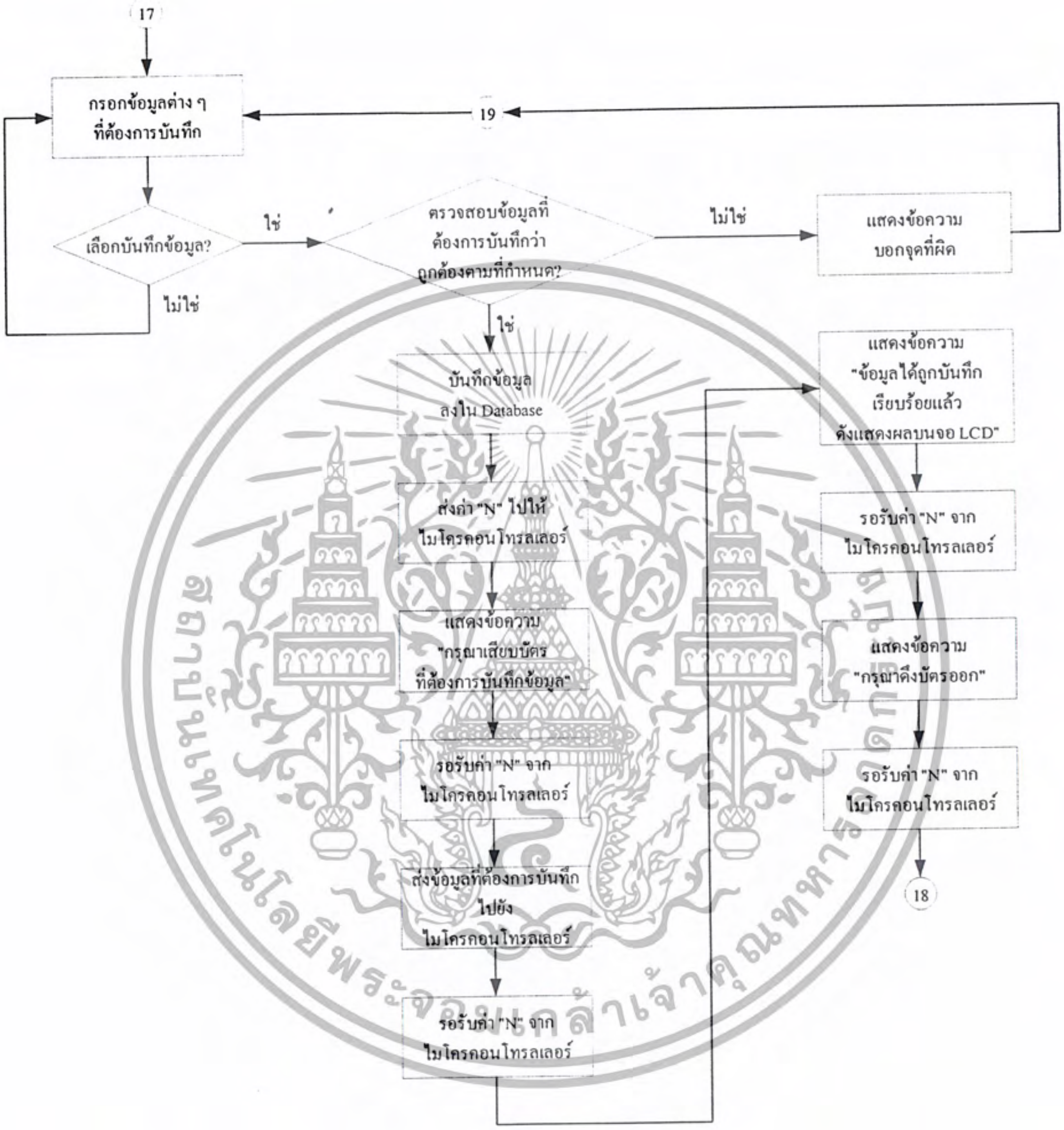
รูปที่ 3.25 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



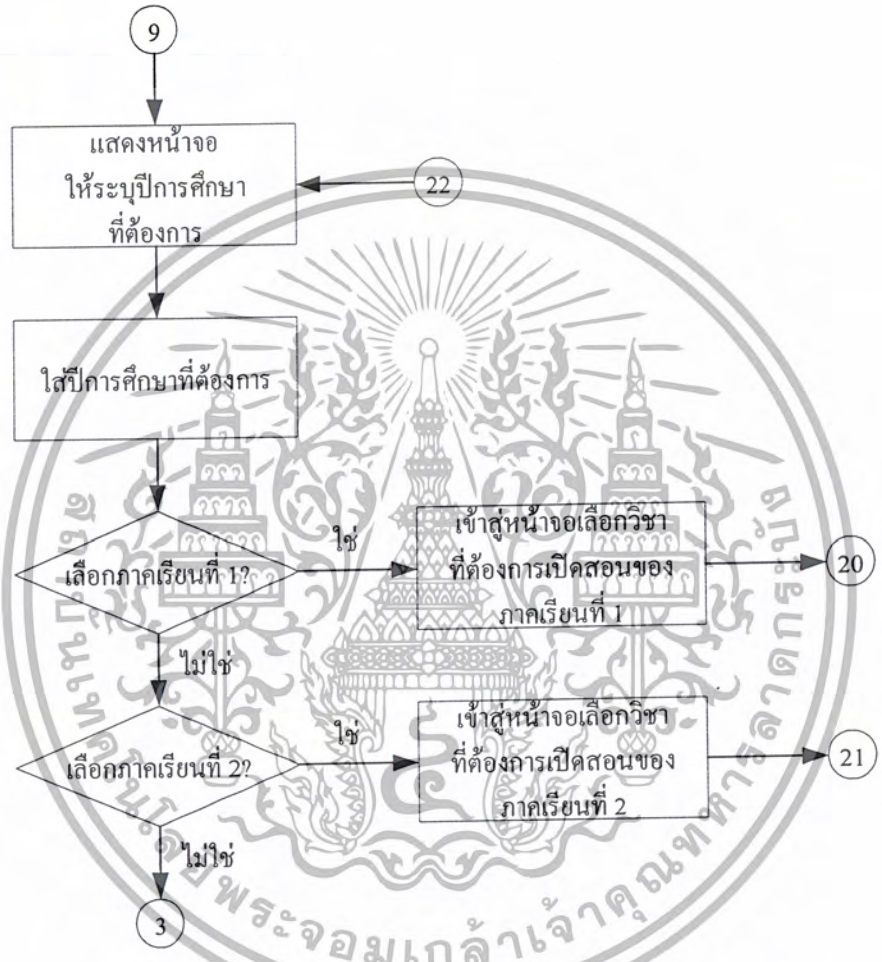
รูปที่ 3.26 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



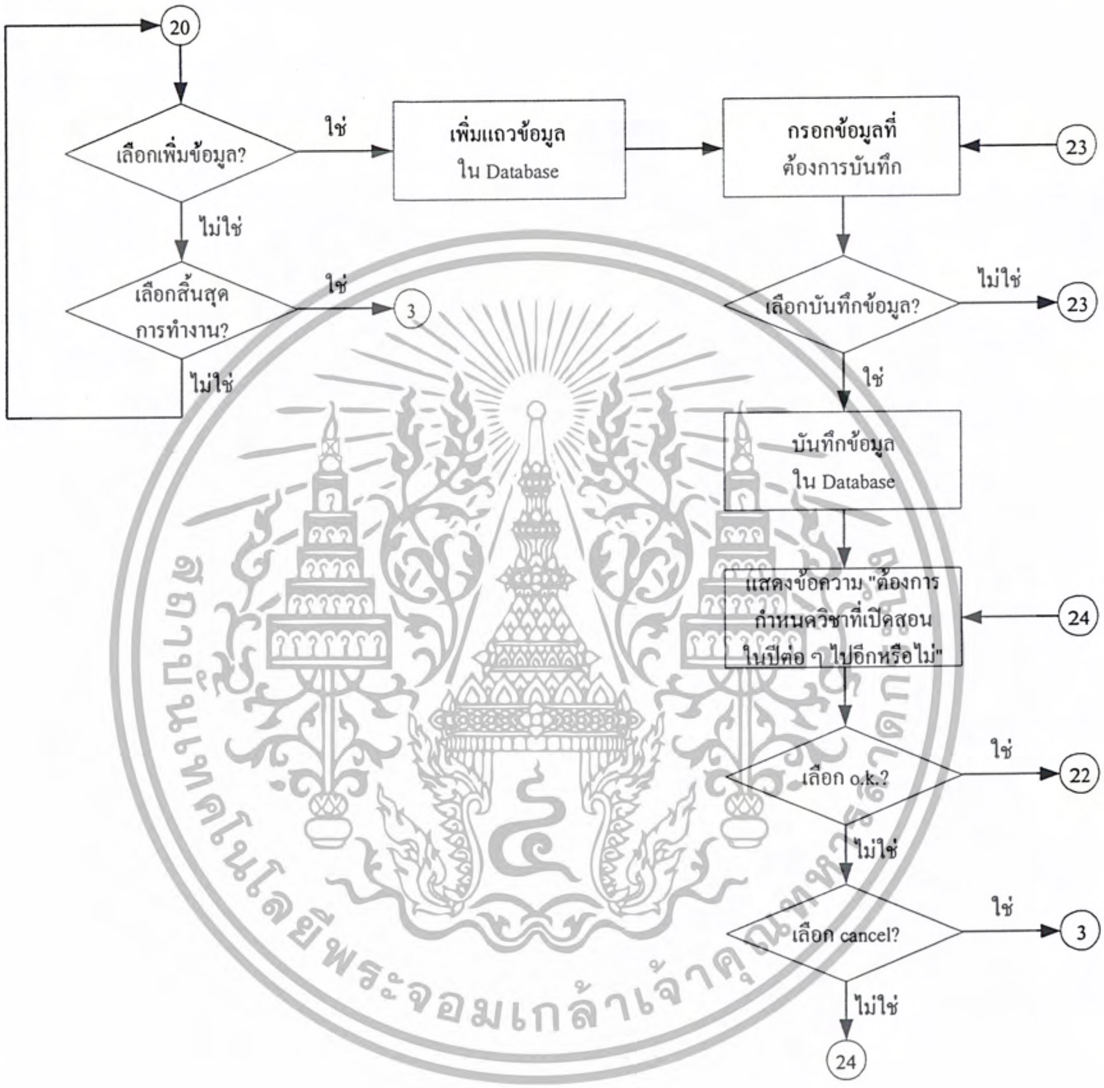
รูปที่ 3.27 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



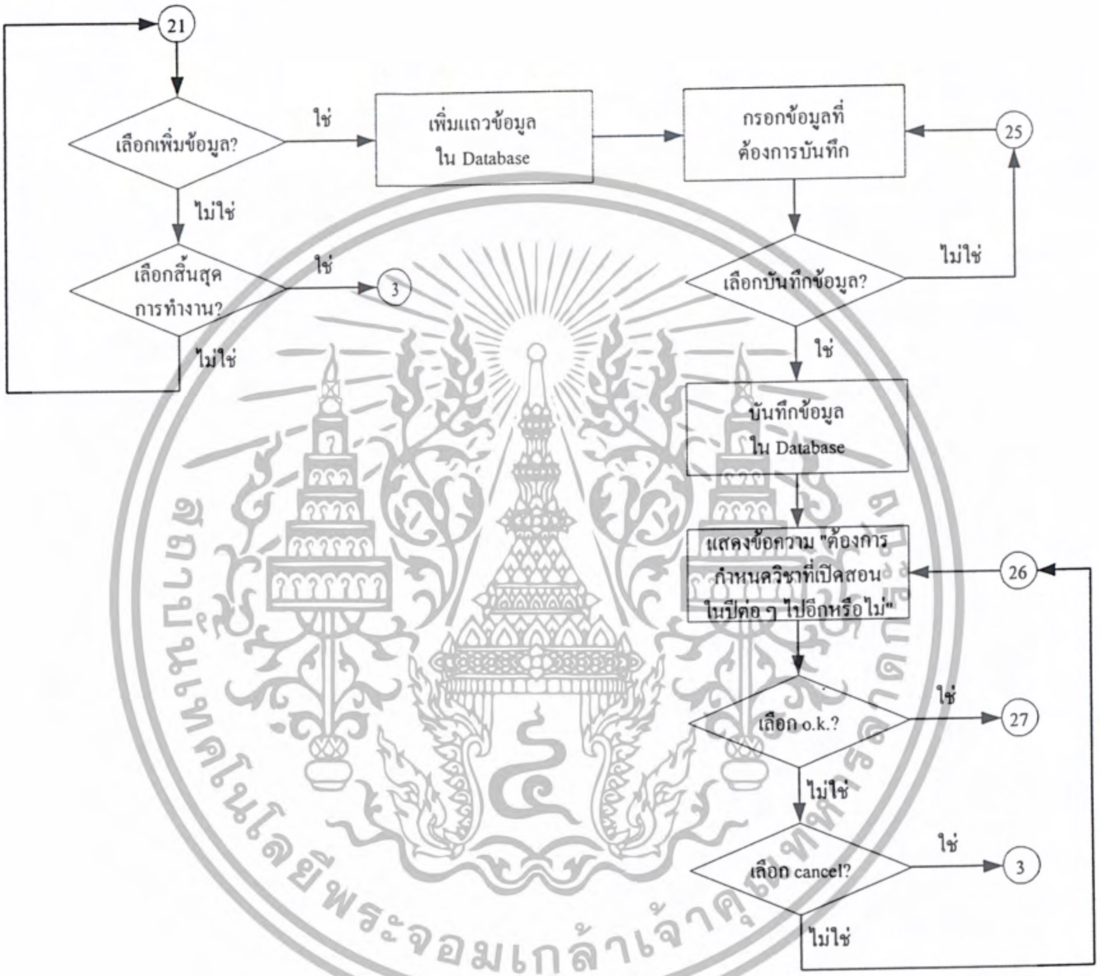
รูปที่ 3.28 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



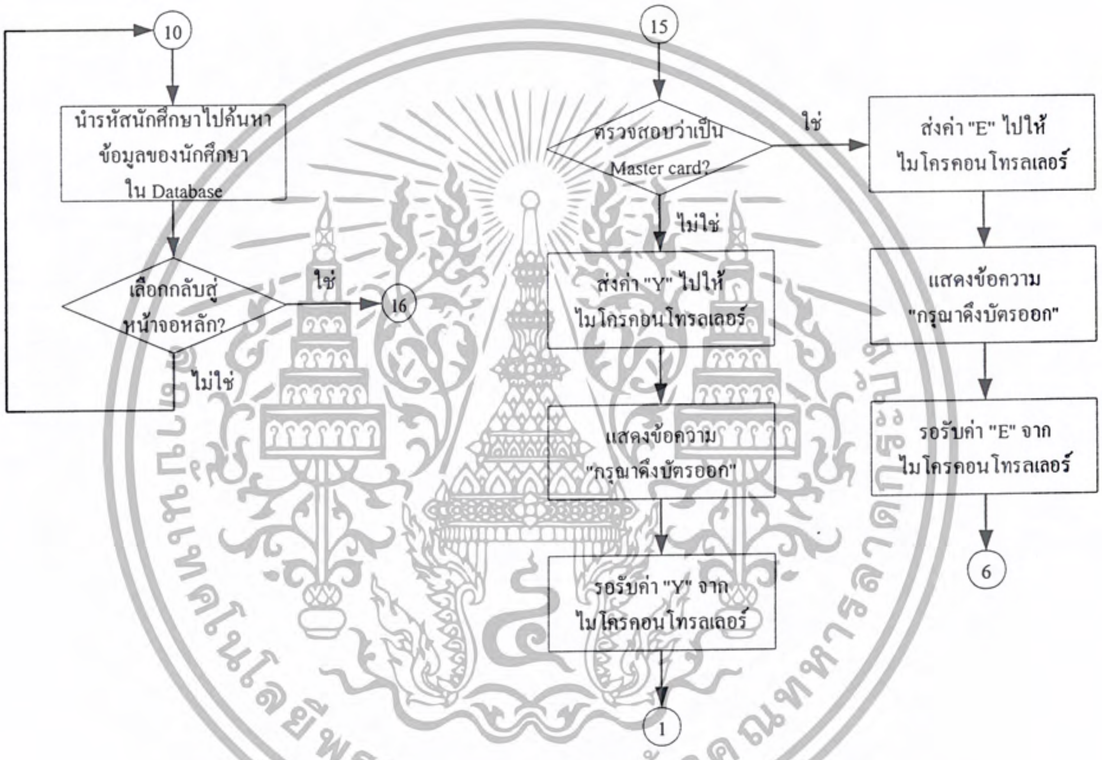
รูปที่ 3.29 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



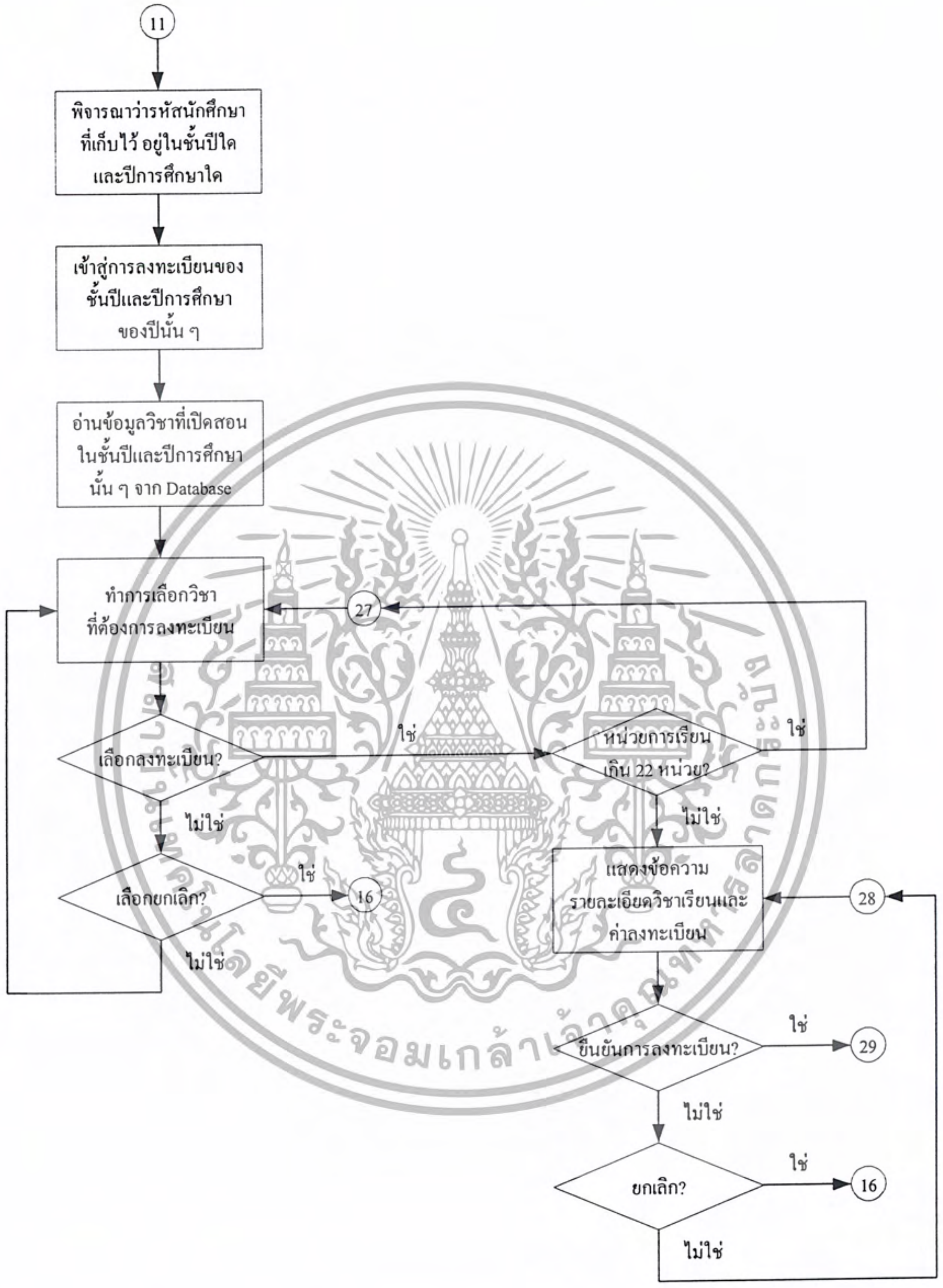
รูปที่ 3.30 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



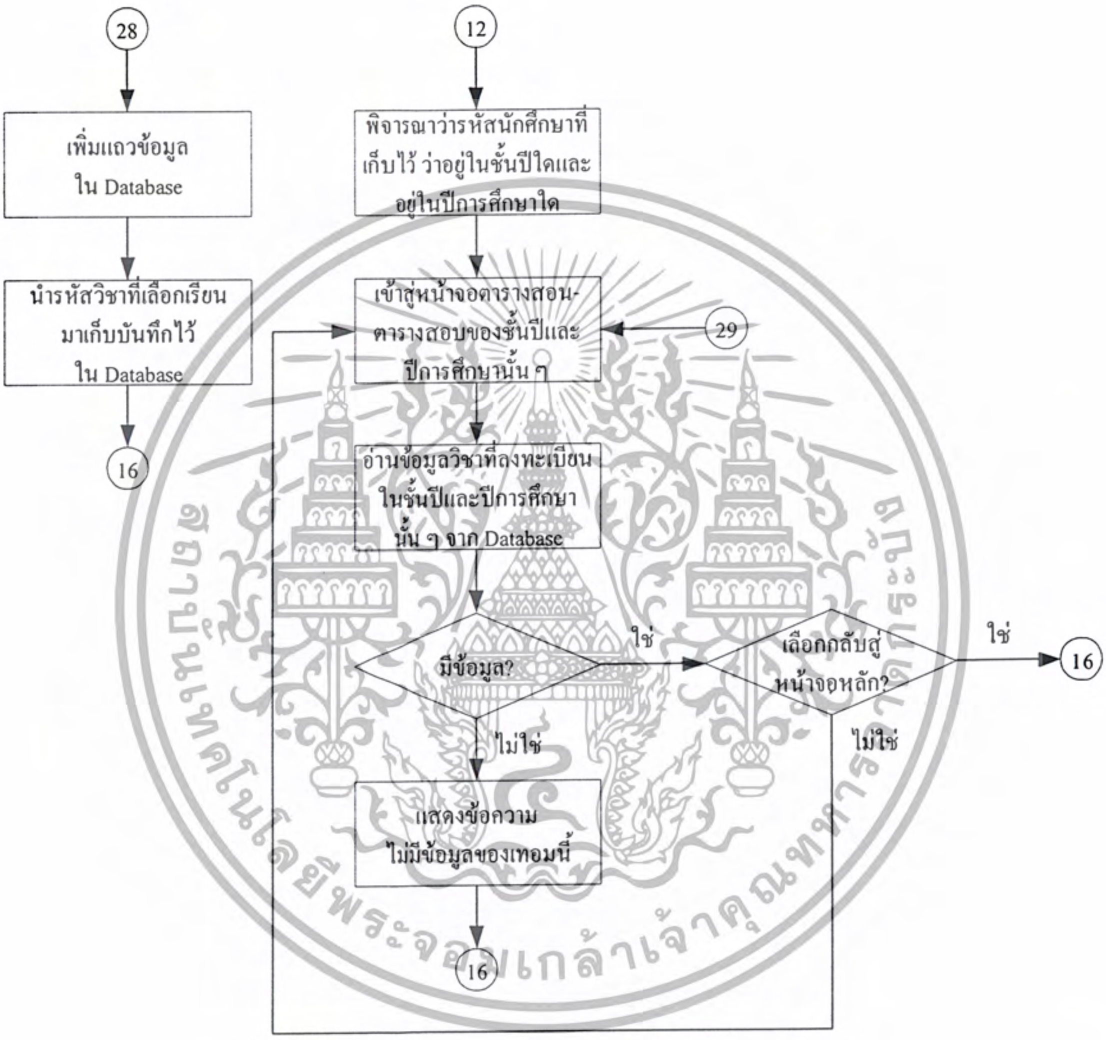
รูปที่ 3.31 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



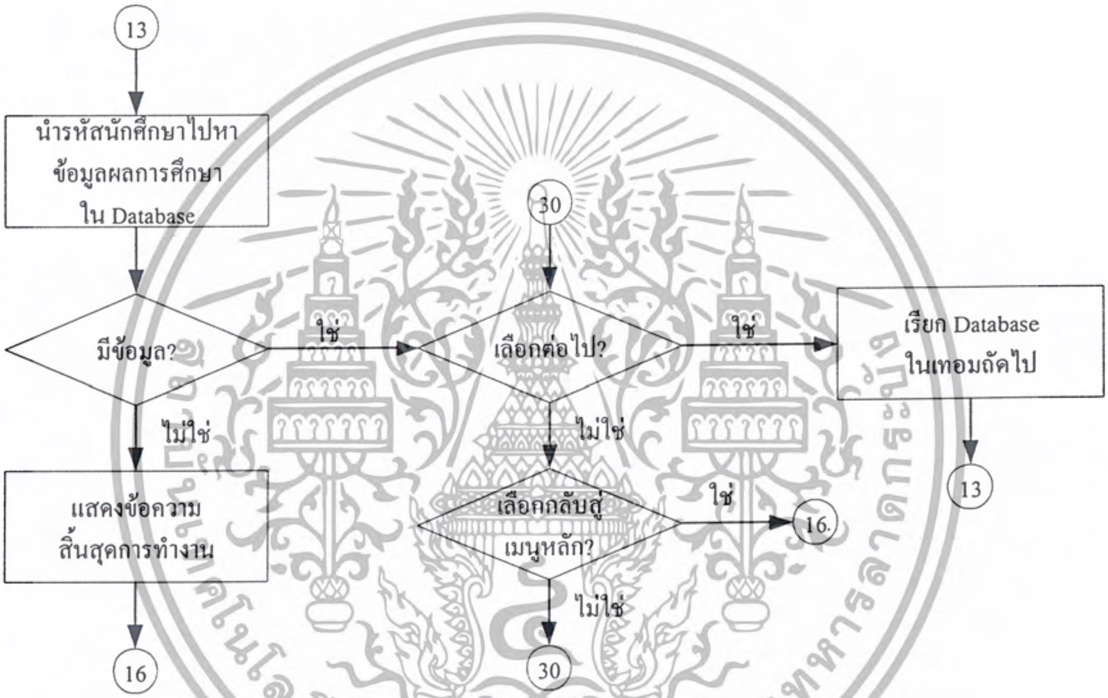
รูปที่ 3.32 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



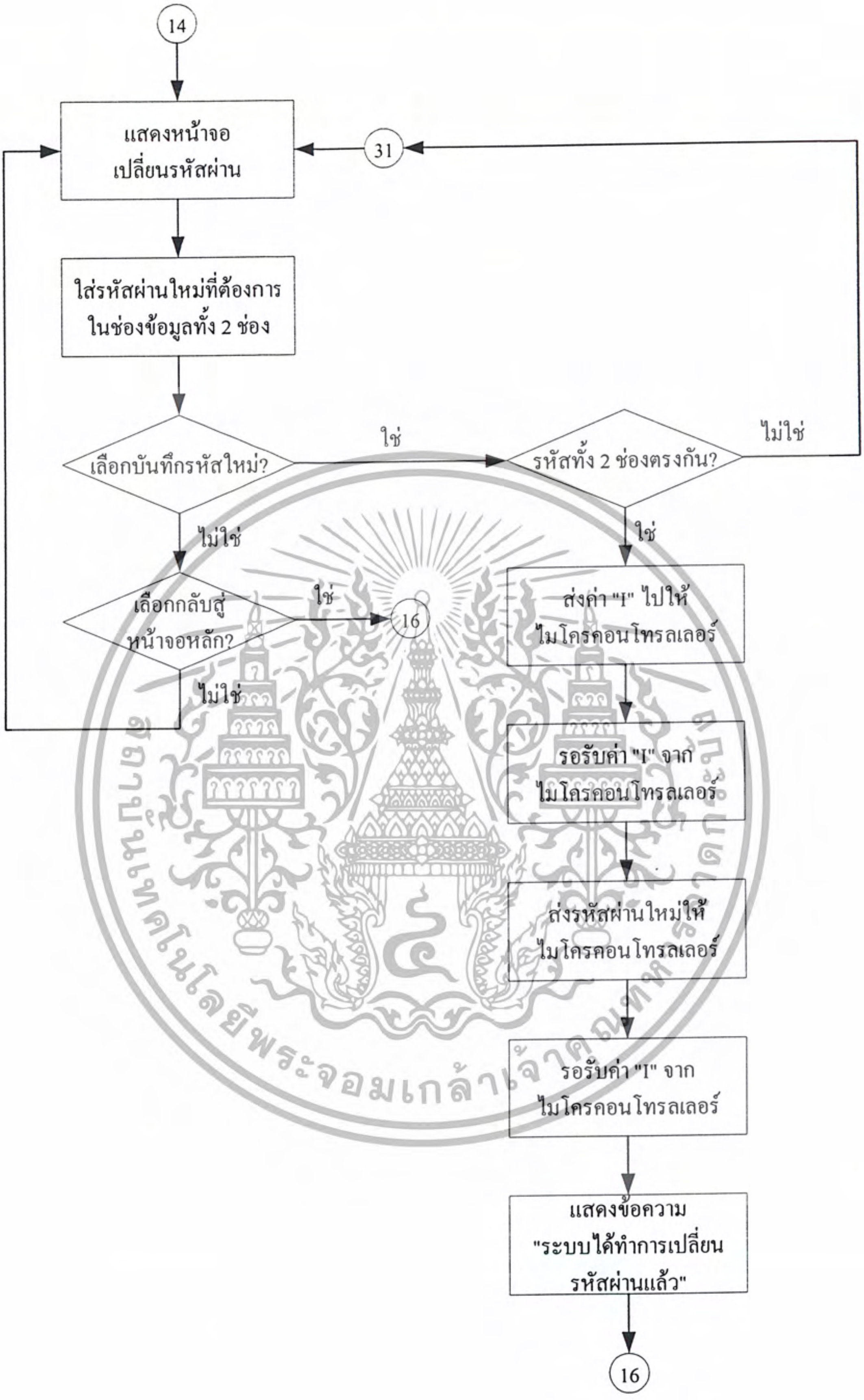
รูปที่ 3.33 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.34 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

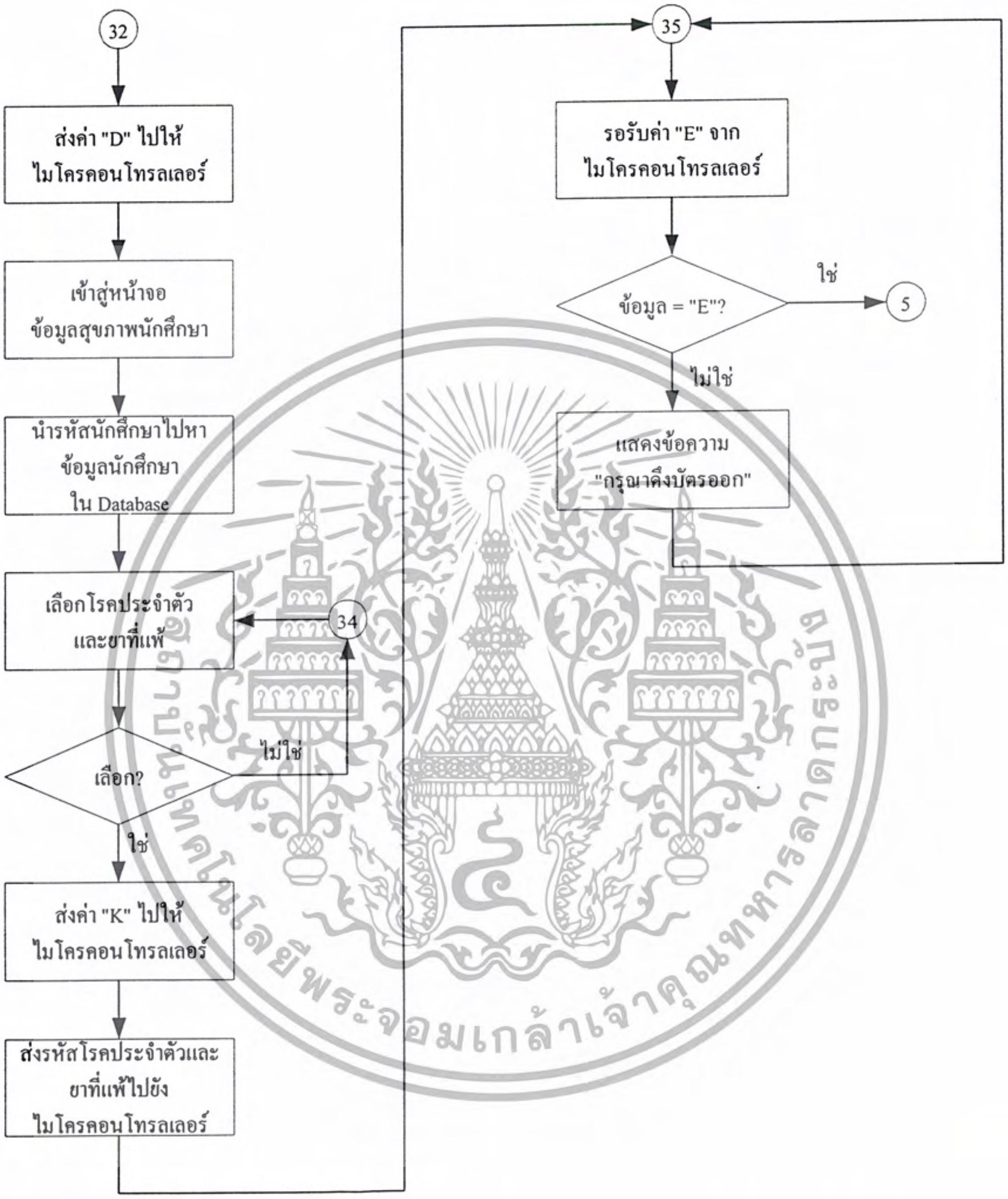


รูปที่ 3.35 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.36 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

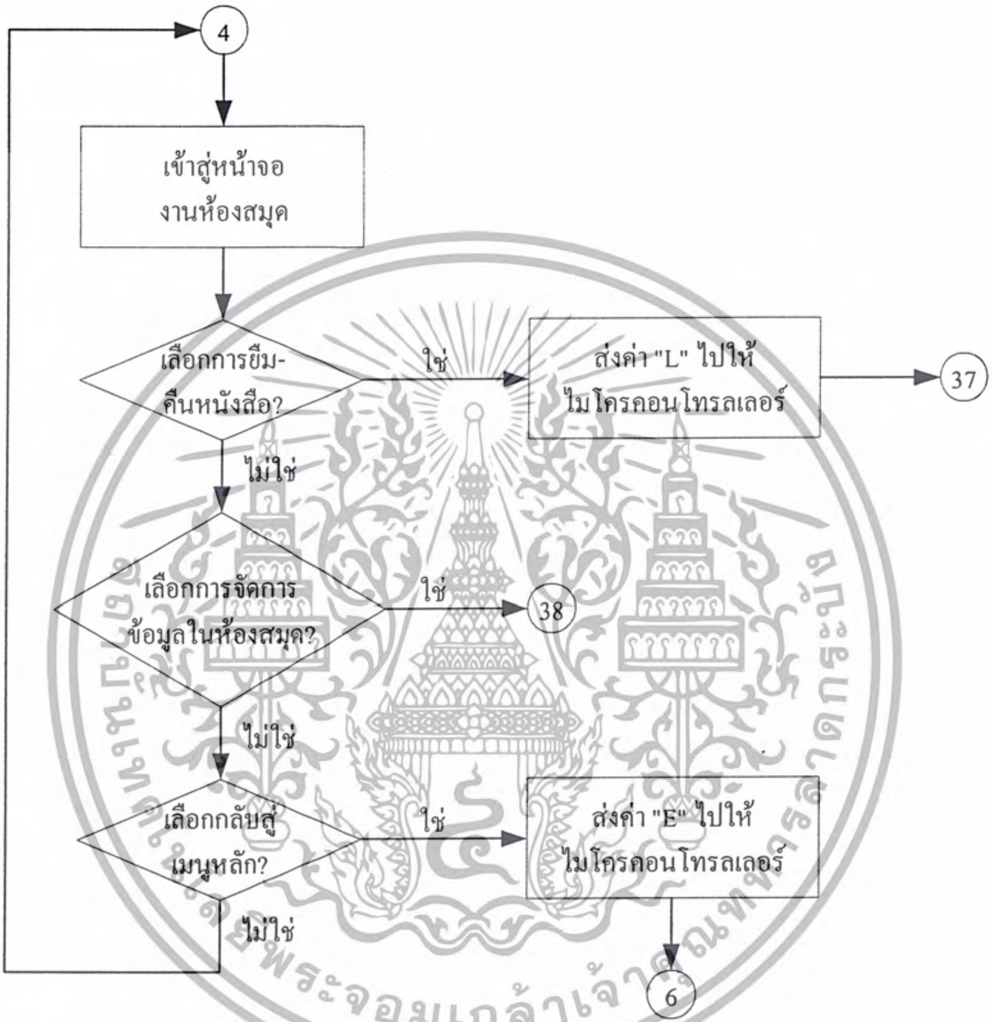


รูปที่ 3.37 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

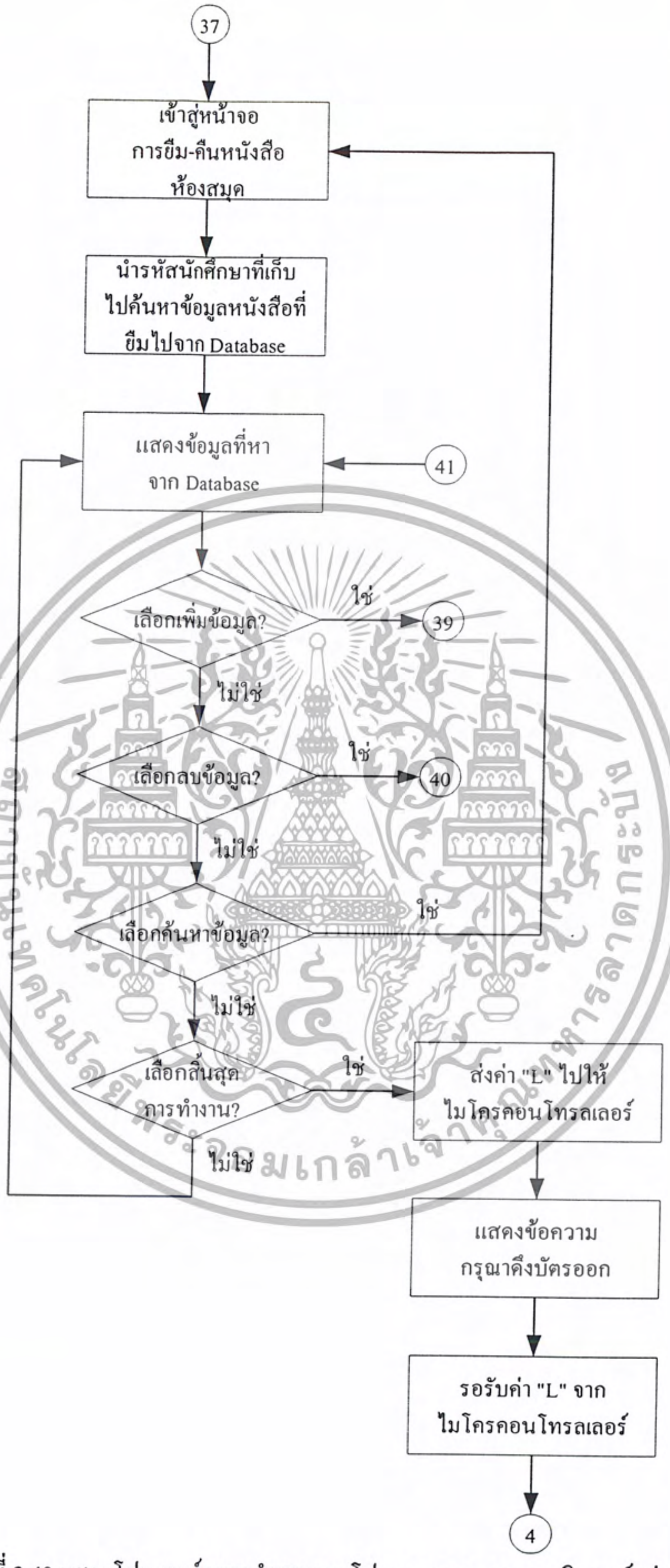
รูปที่ 3.38 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.39 แสดงโฟลวชาร์ททการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.40 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.41 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.42 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.22 ถึงรูปที่ 3.42 แสดงโฟลวชาร์ทการทำงานของโปรแกรมควบคุมคอมพิวเตอร์ ซึ่งสามารถอธิบายการทำงานของโปรแกรมได้เป็นดังนี้ เมื่อเริ่มต้นการทำงาน โปรแกรมจะแสดงหน้าจอ "กรุณาเสียบบัตร" ซึ่งเป็นสภาวะเริ่มต้นการทำงานค้างไว้ และหากทำการเสียบบัตรสมาร์ทการ์ดเข้าไปและใส่รหัสผ่านของบัตรแล้วกด "ตกลง" แล้วนั้น โปรแกรมจะทำการรับข้อมูลที่อ่านมาได้จากบัตรสมาร์ทการ์ด และทำการเก็บค่ารหัสผ่านประจำบัตรไว้ จากนั้นก็จะทำการตรวจสอบว่ารหัสผ่านถูกต้องหรือไม่ หากรหัสผ่านไม่ถูกต้อง ก็จะทำการส่งข้อมูล "C" ไปยังไมโครคอนโทรลเลอร์ และแสดงข้อความ "รหัสไม่ถูกต้อง กรุณาดึงบัตรออก" และรอรับข้อมูล "C" จากไมโครคอนโทรลเลอร์ แล้วกลับสู่หน้าจอสภาวะเริ่มต้นการทำงาน แต่ถ้าหากตรวจสอบแล้วพบว่ารหัสผ่านถูกต้อง ก็จะทำการตรวจสอบต่อไปว่าเป็น Master card หรือบัตรนักศึกษา ซึ่งในที่นี้จะขออธิบายรายละเอียดในแต่ละกรณี ดังนี้

<u>กรณีเป็น Master card</u> หากตรวจสอบพบว่าบัตรที่เข้ามานั้นเป็น Master card ก็จะทำการส่งข้อมูล "D" ไปยังไมโครคอนโทรลเลอร์และแสดงข้อความ "เข้าสู่ระบบแล้ว กรุณาดึงบัตรออก" จากนั้นก็จะรอรับข้อมูล "D" จากไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการใช้งาน" ซึ่งหากเป็น Master card แล้วนั้น ระบบที่สามารถใช้งานได้จะประกอบไปด้วย 3 ระบบหลัก ๆ คือ

1. งานทะเบียน หากเลือกเข้าสู่ระบบนี้ โปรแกรมก็จะทำการส่งข้อมูล "T" ไปยังไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจอของงานทะเบียนซึ่งสามารถเลือกการทำงานย่อยแบ่งออกได้อีก 3 ระบบ คือ

1.1 <u>ระบบสารสนเทศนักศึกษา</u> หากเลือกเข้าสู่ระบบนี้ โปรแกรมก็จะทำการส่งข้อมูล "I" ไปยังไมโครคอนโทรลเลอร์ และแสดงข้อความ "กรุณาเสียบบัตร" จากนั้นเมื่อใส่รหัสผ่านแล้วกด "ตกลง" โปรแกรมจะทำการรับข้อมูลที่อ่านมาได้จากบัตรสมาร์ทการ์ด และทำการเก็บค่ารหัสผ่านประจำบัตรไว้ จากนั้นก็จะทำการตรวจสอบว่ารหัสผ่านถูกต้องหรือไม่ หากรหัสผ่านไม่ถูกต้อง ก็จะทำการส่งข้อมูล "E" ไปยังไมโครคอนโทรลเลอร์ และแสดงข้อความ "รหัสไม่ถูกต้อง กรุณาดึงบัตรออก" และรอรับข้อมูล "E" จากไมโครคอนโทรลเลอร์ แล้วกลับสู่หน้าจอ "โปรดเลือกระบบที่ต้องการใช้งาน" แต่หากตรวจสอบแล้วพบว่ารหัสผ่านถูกต้อง ก็จะทำการส่งข้อมูล "I" ไปยังไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" ซึ่งระบบตรงส่วนนี้นั้น ยังแบ่งออกเป็นอีก 5 ระบบย่อย คือ

1.1.1 ข้อมูล-สถานะนักศึกษา หากทำการเลือกระบบนี้นั้น โปรแกรมก็จะนำรหัสนักศึกษาประจำบัตรสมาร์ทการ์ดที่เสียบเข้ามาไปหาข้อมูลของนักศึกษาใน Database จากนั้นหากเลือกกลับสู่หน้าจอหลัก การทำงานของโปรแกรมก็จะย้อนกลับเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่ได้เลือกกลับสู่หน้าจอหลัก กระบวนการทำงานของโปรแกรมก็จะย้อนกลับไปยังจุดเริ่มต้นของระบบข้อมูล-สถานะนักศึกษา

1.1.2 ระบบลงทะเบียน หากทำการเลือกระบบนี้นั้น โปรแกรมก็จะทำการพิจารณาว่ารหัสนักศึกษาของบัตรสมาร์ทการ์ดที่เสียบเข้ามานั้นอยู่ในชั้นปีใดและปีการศึกษาใด จากนั้นก็จะเข้าสู่หน้าจอการลงทะเบียนของชั้นปีและปีการศึกษานั้น ๆ ต่อมาก็จะทำการอ่านข้อมูลวิชาที่เปิดสอนของชั้นปีและปีการศึกษานั้น ๆ จาก Database และถ้าทำการเลือกวิชาที่ต้องการลงทะเบียนเรียบร้อยแล้ว หากไม่เลือกลงทะเบียน ก็จะทำการพิจารณาต่อว่าเลือกยกเลิกหรือไม่ หากเลือกยกเลือก ก็จะกลับเข้าสู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่เลือกยกเลิก ก็จะกลับสู่หน้าจอให้เลือกวิชาที่ต้องการลงทะเบียนอีกครั้ง แต่หากเลือกลงทะเบียน โปรแกรมก็จะทำการตรวจสอบหน่วยการเรียนรวมที่เลือกว่าเกิน 22 หน่วย หรือไม่ หากเกิน การทำงานก็จะย้อนกลับไปให้เลือกวิชาที่ต้องการลงทะเบียนอีกครั้ง แต่หากตรวจสอบแล้วพบว่าหน่วยการเรียนรวมไม่เกิน 22 หน่วย ก็จะแสดงข้อความรายละเอียดวิชาเรียนและค่าลงทะเบียน จากนั้นหากเลือกยืนยันการลงทะเบียน ก็จะเข้าสู่หน้าจอแสดงตารางสอน-ตารางสอบของชั้นปีและปีการศึกษานั้น ๆ ต่อมาก็จะทำการอ่านข้อมูลวิชาที่ลงทะเบียนในชั้นปีและปีการศึกษานั้น ๆ จาก Database แล้วพิจารณาว่ามีข้อมูลหรือไม่ หากพบว่าไม่มีข้อมูลก็จะแสดงข้อความไม่มีข้อมูลของภาคการศึกษานั้น ๆ แล้วกลับสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่หากพบว่ามีข้อมูล ก็จะพิจารณาต่อไปว่าเลือกกลับสู่หน้าจอหลักหรือไม่ หากเลือกกลับสู่หน้าจอหลัก โปรแกรมก็จะย้อนกลับเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่เลือกกลับสู่หน้าจอหลัก โปรแกรมก็จะแสดงหน้าจอตารางสอน-ตารางสอบของชั้นปีและปีการศึกษานั้น ๆ หากไม่เลือกยืนยันการลงทะเบียน ก็จะพิจารณาต่อมาว่าเลือกยกเลิกหรือไม่ หากเลือกก็จะกลับสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่เลือกยกเลิก ก็จะทำการเพิ่มแถวข้อมูลใน Database และนำรหัสวิชาที่เลือกเรียนมาเก็บไว้ใน Database แล้วเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" อีกครั้ง

1.1.3 ตารางสอน-ตารางสอบ หากเลือกระบบนี้นั้นโปรแกรมก็จะทำการพิจารณาว่ารหัสนักศึกษาของบัตรสมาร์ทการ์ดที่เสียบเข้ามานั้น อยู่ในชั้นปีใดและปีการศึกษาใด จากนั้นก็จะเข้าสู่หน้าจอแสดงตารางสอน-ตารางสอบของชั้นปีและปีการศึกษานั้น ๆ ต่อมาก็จะทำการอ่านข้อมูลวิชาที่ลงทะเบียนในชั้นปีและปีการศึกษานั้น ๆ จาก Database แล้วพิจารณาว่ามีข้อมูลหรือไม่ หากพบว่าไม่มีข้อมูลก็จะแสดงข้อความไม่มีข้อมูลของภาคการศึกษานั้น ๆ แล้วกลับสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่หากพบว่ามีข้อมูล ก็จะพิจารณาต่อไปว่าเลือกกลับสู่หน้าจอหลักหรือไม่ หากเลือกกลับสู่หน้าจอหลัก โปรแกรมก็จะย้อนกลับเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่เลือกกลับสู่หน้าจอหลัก โปรแกรมก็จะแสดงหน้าจอตารางสอน-ตารางสอบของชั้นปีและปีการศึกษานั้น ๆ

1.1.4 ผลการศึกษา หากเลือกระบบนี้นั้น โปรแกรมก็จะนำรหัสนักศึกษาของบัตรสมาร์ทการ์ดที่เสียบเข้ามานั้นไปหาข้อมูลผลการศึกษาใน Database และพิจารณาว่ามีข้อมูลหรือไม่ หาพบว่าไม่มีข้อมูล ก็จะแสดงข้อความสิ้นสุดการทำงาน และกลับเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่หากพบว่ามีข้อมูล ก็จะทำการพิจารณาต่อไปว่าเลือกต่อไปหรือไม่ หากเลือกต่อไป ก็จะทำการเรียก Database ในภาคการศึกษาถัดไป แล้วกลับเข้าสู่กระบวนการการทำงานหลักของผลการศึกษาอีกครั้ง แต่ถ้าหากไม่เลือกต่อไป ก็จะทำการพิจารณาต่อว่าเลือกกลับสู่เมนูหลักหรือไม่ หากไม่เลือกกลับสู่เมนูหลัก กระบวนการการทำงานก็จะย้อนกลับไปพิจารณาว่าเลือกต่อไปหรือไม่อีกครั้ง แต่หากเลือกกลับสู่เมนูหลัก กระบวนการการทำงานของโปรแกรมก็จะย้อนกลับไปสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ"

1.1.5 เปลี่ยนรหัสผ่าน หากเลือกระบบนี้นั้น โปรแกรมก็จะแสดงหน้าจอเปลี่ยนรหัสผ่านและแสดงข้อความให้ใส่หรัสผ่านที่ต้องการเปลี่ยนในช่องข้อมูล 2 ช่อง ต่อมาหากเลือกบันทึกรหัสใหม่ โปรแกรมก็จะทำการตรวจสอบว่ารหัสผ่านกี่กรอกทั้ง 2 ช่องนั้นตรงกันหรือไม่ หากพบว่าไม่ตรงกัน ก็จะกลับสู่หน้าจอเปลี่ยนรหัสผ่าน แต่ถ้าตรวจสอบแล้วพบว่ารหัสผ่านที่กรอกทั้ง 2 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงกัน ก็จะทำการส่งข้อมูล "I" ไปยังไมโครคอนโทรลเลอร์ และรอรับข้อมูล "I" จากไมโครคอนโทรลเลอร์ เมื่อรับข้อมูล "I" ได้แล้วก็จะทำการส่งรหัสผ่านใหม่ไปยังไมโครคอนโทรลเลอร์ และรอรับข้อมูล "I" จากไมโครคอนโทรลเลอร์อีกครั้ง หากได้รับข้อมูล "I" แล้ว ก็จะแสดงข้อความว่า "ระบบได้ทำการเปลี่ยนรหัสผ่านแล้ว" แล้วกลับเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่เลือกบันทึกรหัสใหม่ ก็จะพิจารณาต่อว่าเลือกกลับสู่หน้าจอหลักหรือไม่ หากเลือก กระบวนการการทำงานก็จะย้อนกลับไปสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" แต่ถ้าหากไม่เลือก กระบวนการการทำงานก็จะย้อนกลับไปสู่หน้าจอเปลี่ยนรหัสผ่าน

ถ้าหากไม่เลือกระบบทั้ง 5 ระบบดังที่กล่าวมาข้างต้น โปรแกรมก็จะทำการตรวจสอบว่าบัตรสมาร์ทการ์ดที่เสียบเข้ามานั้นเป็น Master card หรือไม่ หากพบว่าเป็น Mater card ก็จะส่งข้อมูล "E" ไปให้ไมโครคอนโทรลเลอร์ และแสดงข้อความกรุณาดึงบัตรออก แล้วรอรับข้อมูล "E" จากไมโครคอนโทรลเลอร์ ซึ่งหากได้รับข้อมูล "E" แล้วก็จะกลับเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการใช้งาน" แต่ถ้าหากตรวจแล้วพบว่าไม่ใช่ Master card ก็จะทำการส่งข้อมูล "Y" ไปยังไมโครคอนโทรลเลอร์ และแสดงข้อความให้ดึงบัตรออก แล้วรอรับข้อมูล "Y" จากไมโครคอนโทรลเลอร์ จากนั้นก็กลับสู่หน้าจอสภาวะเริ่มต้นการทำงาน

1.2 <u>ป้อนข้อมูลนักศึกษา</u> หากเลือกเข้าสู่ระบบนี้ โปรแกรมจะแสดงหน้าจอป้อนข้อมูลนักศึกษา และพิจารณาว่าทางเลือกต่าง ๆ ต่อไปนี้

1.2.1 หากเลือกเพิ่มข้อมูล ก็จะทำการเพิ่มแถวข้อมูลใน Database แล้วให้ทำการกรอกข้อมูลต่าง ๆ ที่ต้องการบันทึก และต่อมาจะพิจารณาว่าเลือกบันทึกข้อมูลหรือไม่ หากไม่เลือกบันทึกข้อมูล ก็จะกลับไปให้หน้าจอที่ให้กรอกข้อมูลอีกครั้ง แต่หากเลือกบันทึกข้อมูล ก็จะทำการตรวจสอบว่าข้อมูลที่ต้องการบันทึกนั้นถูกต้องตามที่กำหนดหรือไม่ หากตรวจสอบแล้วพบว่าข้อมูลไม่ถูกต้อง ก็จะแสดงข้อความบอกจุดที่ผิดและกลับเข้าสู่หน้าจอที่ให้กรอกข้อมูลใหม่ แต่ถ้าหากตรวจสอบแล้วว่าข้อมูลถูกต้อง ก็จะทำการบันทึกข้อมูลลงใน Database และส่งข้อมูล "N" ไปยังไมโครคอนโทรลเลอร์ แล้วแสดงข้อความ "กรุณาเสียบบัตรที่ต้องการบันทึกข้อมูล" และรอรับข้อมูล "N" จากไมโครคอนโทรลเลอร์ จากนั้นจึงส่งข้อมูลที่ต้องการบันทึกไปยังไมโครคอนโทรลเลอร์ และรอรับข้อมูล "N" จากไมโครคอนโทรลเลอร์อีก เมื่อรับข้อมูล "N" จากไมโครคอนโทรลเลอร์ได้แล้วนั้น ก็จะแสดงข้อความว่า "ข้อมูลได้ถูกบันทึกเรียบร้อยแล้ว ดังแสดงผลบนจอ LCD" จากนั้นก็จะรอรับข้อมูล "N" จากไมโครคอนโทรลเลอร์อีกครั้ง หากได้รับแล้วก็จะแสดงข้อความให้ดึงบัตรออกแล้วทำการรอรับข้อมูล "N" จากไมโครคอนโทรลเลอร์อีก ซึ่งหากได้รับแล้วก็จะกลับสู่หน้าจอป้อนข้อมูลนักศึกษา

1.2.2 หากเลือกลบข้อมูล โปรแกรมจะทำการลบข้อมูลใน Database แล้วกลับสู่หน้าจอป้อนข้อมูลนักศึกษา

1.2.3 หากเลือกกลับสู่หน้าจอหลัก โปรแกรมจะทำการส่งข้อมูล "E" ไปยังไมโครคอนโทรลเลอร์แล้วกลับสู่หน้าจอ "โปรดเลือกระบบที่ต้องการใช้งาน"

แต่หากไม่เลือกทางเลือกทั้ง 3 ทางข้างต้น โปรแกรมก็จะแสดงหน้าจอป้อนข้อมูลนักศึกษาค้างไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 <u>กำหนดวิชาที่เปิดสอน</u> หากเลือกเข้าสู่ระบบนี้ โปรแกรมจะแสดงหน้าจอให้ระบุปีการศึกษาที่ต้องการ และเมื่อใส่ปีการศึกษาที่ต้องการแล้ว โปรแกรมจะพิจารณาว่าแบ่งออกเป็น 2 ทางเลือก ดังนี้

1.3.1 หากเลือกภาคเรียนที่ 1 ก็จะเข้าสู่หน้าจอเลือกวิชาที่ต้องการเปิดสอนของภาคเรียนที่ 1 และพิจารณาต่อว่าเลือกเพิ่มข้อมูลหรือไม่ หากไม่เลือกเพิ่มข้อมูล จะพิจารณาต่อไปอีกว่าเลือกสิ้นสุดการทำงานหรือไม่ หากเลือกสิ้นสุดการทำงานก็จะย้อนกลับไปเข้าสู่หน้าจองานทะเบียน แต่ถ้าหากไม่เลือกสิ้นสุดการทำงานก็จะย้อนกลับไปที่การพิจารณาว่าเลือกเพิ่มข้อมูลหรือไม่ ถ้าหากเลือกเพิ่มข้อมูล โปรแกรมจะทำการเพิ่มแถวข้อมูลใน Database และให้กรอกข้อมูลที่ต้องการบันทึก จากนั้นจะพิจารณาว่าเลือกบันทึกข้อมูลหรือไม่ หากไม่เลือก ก็จะย้อนกลับไปหน้าจอที่ให้กรอกข้อมูล แต่ถ้าหากเลือกบันทึกข้อมูล ก็จะบันทึกข้อมูลที่กรอกลงใน Database และแสดงข้อความว่าต้องการกำหนดวิชาที่เปิดสอนในปีต่อ ๆ ไปอีกหรือไม่ หากเลือก o.k. ก็จะกลับเข้าสู่หน้าจอให้ระบุปีการศึกษาที่ต้องการ แต่ถ้าไม่เลือก o.k. ก็จะพิจารณาต่อว่าเลือก cancel หรือไม่ หากไม่เลือก ก็จะกลับเข้าสู่หน้าจอแสดงข้อความว่าต้องการกำหนดวิชาที่เปิดสอนในปีต่อ ๆ ไปอีกหรือไม่ แต่ถ้าหากเลือก cancel ก็จะกลับเข้าสู่หน้าจองานทะเบียน

1.3.2 หากเลือกภาคเรียนที่ 2 ก็จะเข้าสู่หน้าจอเลือกวิชาที่ต้องการเปิดสอนของภาคเรียนที่ 2 และพิจารณาต่อว่าเลือกเพิ่มข้อมูลหรือไม่ หากไม่เลือกเพิ่มข้อมูล จะพิจารณาต่อไปอีกว่าเลือกสิ้นสุดการทำงานหรือไม่ หากเลือกสิ้นสุดการทำงานก็จะย้อนกลับไปเข้าสู่หน้าจองานทะเบียน แต่ถ้าหากไม่เลือกสิ้นสุดการทำงานก็จะย้อนกลับไปที่การพิจารณาว่าเลือกเพิ่มข้อมูลหรือไม่ ถ้าหากเลือกเพิ่มข้อมูล โปรแกรมจะทำการเพิ่มแถวข้อมูลใน Database และให้กรอกข้อมูลที่ต้องการบันทึก จากนั้นจะพิจารณาว่าเลือกบันทึกข้อมูลหรือไม่ หากไม่เลือก ก็จะย้อนกลับไปหน้าจอที่ให้กรอกข้อมูล แต่ถ้าหากเลือกบันทึกข้อมูล ก็จะบันทึกข้อมูลที่กรอกลงใน Database และแสดงข้อความว่าต้องการกำหนดวิชาที่เปิดสอนในปีต่อ ๆ ไปอีกหรือไม่ หากเลือก o.k. ก็จะกลับเข้าสู่หน้าจอให้ระบุปีการศึกษาที่ต้องการ แต่ถ้าไม่เลือก o.k. ก็จะพิจารณาต่อว่าเลือก cancel หรือไม่ หากไม่เลือก ก็จะกลับเข้าสู่หน้าจอแสดงข้อความว่าต้องการกำหนดวิชาที่เปิดสอนในปีต่อ ๆ ไปอีกหรือไม่ แต่ถ้าหากเลือก cancel ก็จะกลับเข้าสู่หน้าจองานทะเบียน

หากไม่เลือกทางเลือกทั้ง 2 ทางดังกล่าวข้างต้น โปรแกรมก็จะกลับสู่หน้าจองานทะเบียน

หากไม่เลือกทำงานใด ๆ ในระบบของงานทะเบียนเลย โปรแกรมก็จะกลับสู่หน้าจอ “โปรดเลือกระบบที่ต้องการใช้งาน”

2. งานห้องสมุด หากเลือกเข้าสู่ระบบนี้ โปรแกรมจะทำการส่งข้อมูล “L” ไปยังไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจองานห้องสมุด ซึ่งมีทางเลือกในการทำงานทั้งหมด 3 ทางเลือก คือ

2.1 <u>การยืม-คืนหนังสือห้องสมุด</u> หากเลือกทางเลือกนี้ โปรแกรมจะทำการส่งข้อมูล “L” ไปยังไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจอการยืม-คืนหนังสือห้องสมุด จากนั้นจะนำรหัสนักศึกษาจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัตรสมาร์ทการ์ดที่เสียบเข้ามาไปทำการค้นหาข้อมูลหนังสือที่ยืมไปจาก Database แล้วแสดงข้อมูลที่หาได้จาก Database จากนั้นจะพิจารณาทางเลือกย่อยอีก 4 ทางเลือก คือ

2.1.1 เพิ่มข้อมูล หากเลือกทางเลือกนี้ โปรแกรมจะทำการเพิ่มแถวข้อมูลใน Database จากนั้นจะให้กรอกชื่อหนังสือ และวันยืม-กำหนดคืนหนังสือ จากนั้นจะพิจารณาว่าเลือกบันทึกข้อมูลหรือไม่ หากเลือกบันทึกข้อมูล โปรแกรมก็จะทำการบันทึกรหัสผู้ยืม (ซึ่งในที่นี้คือ รหัสนักศึกษา) ชื่อหนังสือ และวันยืม-กำหนดคืนหนังสือ จากนั้นก็กลับเข้าสู่กระบวนการแสดงข้อมูลที่หาจาก Database อีกครั้ง

2.1.2 ลบข้อมูล หากเลือกทางเลือกนี้ โปรแกรมก็จะนำรหัสนักศึกษาจากบัตรสมาร์ทการ์ดที่เสียบเข้ามาไปค้นหาชื่อหนังสือที่ติดค้างไว้ และตรวจสอบว่ามีข้อมูลหรือไม่ หากไม่มีข้อมูลก็จะกลับเข้าสู่กระบวนการแสดงข้อมูลที่หาจาก Database อีกครั้ง แต่ถ้าหากตรวจสอบแล้วพบว่ามีข้อมูล ก็จะให้เลือกชื่อหนังสือที่นำมาคืน แล้วลบข้อมูลหนังสือที่นำมาคืนออกจาก Database

2.1.3 ค้นหาข้อมูล หากเลือกทางเลือกนี้ โปรแกรมก็จะเข้าสู่หน้าจอการยืม-คืนหนังสือห้องสมุด จากนั้นก็จะนำรหัสนักศึกษาจากบัตรสมาร์ทการ์ดที่เสียบเข้ามาไปค้นหาข้อมูลหนังสือที่ยืมไปจาก Database แล้วเข้าสู่กระบวนการแสดงข้อมูลที่หาจาก Database อีกครั้ง

2.1.4 สิ้นสุดการทำงาน หากเลือกทางเลือกนี้ โปรแกรมจะทำการส่งข้อมูล “L” ไปยังไมโครคอนโทรเลอร์ แล้วแสดงข้อความ “กรุณาดึงบัตรออก” จากนั้นจะรอรับข้อมูล “L” จากไมโครคอนโทรลเลอร์ เมื่อได้รับแล้วก็จะกลับเข้าสู่หน้าจองานห้องสมุด

หากไม่ได้เลือกทางเลือกทั้ง 4 ทาง ดังกล่าวข้างต้น โปรแกรมก็จะแสดงข้อมูลที่หาจาก Database ค้างไว้

2.2 <u>การจัดการข้อมูลในห้องสมุด</u> หากเลือกทางเลือกนี้ โปรแกรมจะเข้าสู่หน้าจอการจัดการข้อมูลในห้องสมุด จากนั้นจะมีทางเลือกย่อยอีก 2 ทาง คือ

2.2.1 เพิ่มข้อมูล หากเลือกทางเลือกนี้ โปรแกรมจะทำการเพิ่มแถวข้อมูลใน Database และให้กรอกข้อมูลของหนังสือที่ต้องการบันทึก จากนั้นจะพิจารณาว่าเลือกบันทึกข้อมูลหรือไม่ หากเลือกบันทึกข้อมูล โปรแกรมก็จะทำการบันทึกข้อมูลของหนังสือเล่มนั้นลงใน Database แล้วกลับเข้าสู่หน้าจอการจัดการข้อมูลในห้องสมุด

2.2.2 กลับสู่งานห้องสมุด หากเลือกทางเลือกนี้ โปรแกรมจะกลับเข้าสู่หน้าจองานห้องสมุด

หากไม่ได้เลือกทางเลือกทั้ง 2 ทาง ดังกล่าวข้างต้น โปรแกรมก็จะให้เลือกดูชื่อหนังสือจาก Database แล้วพิจารณาว่าเลือกลบข้อมูลหรือไม่ หากเลือกลบข้อมูล โปรแกรมก็จะทำการลบข้อมูลของหนังสือที่เลือกออกจาก Database แต่ถ้าหากไม่เลือกลบข้อมูลโปรแกรมก็จะกลับเข้าสู่หน้าจอการจัดการข้อมูลในห้องสมุด

2.3 <u>กลับสู่เมนูหลัก</u> หากเลือกทางเลือกนี้ โปรแกรมจะทำการส่งข้อมูล “E” ไปยังไมโครคอนโทรลเลอร์ แล้วกลับเข้าสู่หน้าจอ “โปรดเลือกระบบที่ต้องการใช้งาน”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. **งานอนามัย** หากเลือกเข้าสู่ระบบนี้ โปรแกรมก็จะทำการส่งข้อมูล "H" ไปยังไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจอของงานอนามัย จากนั้นจะพิจารณว่าเลือก "ตกลง" หรือไม่ หากไม่เลือก ก็จะพิจารณาต่อว่าเลือก "ยกเลิก" หรือไม่ หากเลือกยกเลิก ก็จะทำการส่งข้อมูล "E" ไปยังไมโครคอนโทรลเลอร์ แต่ถ้าหากไม่เลือกยกเลิก ก็จะแสดงหน้าจองานอนามัยค้างไว้ แต่ถ้าเลือกตกลง ก็จะส่งข้อมูล "H" ไปยังไมโครคอนโทรลเลอร์ แล้วแสดงข้อความกรุณาเสียบบัตร และทำการอ่านข้อมูลจากบัตรสมาร์ทการ์ด และเก็บค่ารหัสนักศึกษา รหัสโรคประจำตัวและชื่อยาที่แพ้ จากนั้นจะทำการตรวจสอบว่าเคยมารับการตรวจหรือไม่ หากเคยมารับการตรวจแล้ว ก็จะส่งข้อมูล "H" ไปยังไมโครคอนโทรลเลอร์ แล้วเข้าสู่หน้าจอข้อมูลสุขภาพนักศึกษา จากนั้นจะนำรหัสนักศึกษาไปหาข้อมูลนักศึกษาใน Database แล้วนำรหัสโรคประจำตัวและชื่อยาที่แพ้มาแสดงเป็นชื่อโรคและชื่อยาที่แพ้ ต่อมาจะพิจารณาว่าเลือกสิ้นสุดการทำงานหรือไม่ หากไม่เลือกสิ้นสุดการทำงานก็จะแสดงชื่อโรคประจำตัวและชื่อยาที่แพ้ค้างไว้ แต่ถ้าหากเลือกสิ้นสุดการทำงาน ก็จะส่งข้อมูล "E" ไปยังไมโครคอนโทรลเลอร์ และแสดงข้อความกรุณาดึงบัตรออก แล้วรอรับข้อมูล "E" จากไมโครคอนโทรลเลอร์ หากได้รับแล้วก็จะกลับเข้าสู่หน้าจอของงานอนามัย ถ้าตรวจสอบพบว่ายังไม่เคยมารับการตรวจ ก็จะทำการส่งข้อมูล "O" ไปยังไมโครคอนโทรลเลอร์ และเข้าสู่หน้าจอข้อมูลสุขภาพนักศึกษา จากนั้นจะนำรหัสนักศึกษาไปหาข้อมูลใน Database ต่อมาจะให้เลือกชื่อโรคประจำตัวและชื่อยาที่แพ้ แล้วพิจารณาว่าเลือก "เลือก" หรือไม่ หาไม่ก็จะแสดงหน้าจอให้เลือกชื่อโรคประจำตัวและชื่อยาที่แพ้ค้างไว้ แต่ถ้าหากเลือก ก็จะทำการส่งข้อมูล "K" ไปยังไมโครคอนโทรลเลอร์ และส่งรหัสชื่อโรคประจำตัวและชื่อยาที่แพ้ไปยังไมโครคอนโทรลเลอร์ จากนั้นจะรอรับข้อมูล "E" จากไมโครคอนโทรลเลอร์ หากไม่ได้รับข้อมูล "E" ก็จะแสดงข้อความกรุณาดึงบัตรออก แล้วกลับไปรอรับข้อมูลจากไมโครคอนโทรลเลอร์ใหม่ แต่ถ้าหากได้รับข้อมูล "E" แล้ว ก็จะกลับเข้าสู่หน้าจองานอนามัย

<u>กรณีเป็นบัตรนักศึกษา</u> โปรแกรมก็จะทำการส่งข้อมูล "Z" ไปยังไมโครคอนโทรลเลอร์ จากนั้นเข้าสู่หน้าจอ "โปรดเลือกระบบที่ต้องการ" ซึ่งมีกระบวนการทำงานเช่นเดียวกับหากเลือกระบบสารสนเทศนักศึกษาในกรณีที่เป็น Master card แต่จะมีความแตกต่างกันตรงที่หากเป็นบัตรนักศึกษานั้น เมื่อเลือกสิ้นสุดการทำงาน โปรแกรมก็จะส่งข้อมูล "Y" ไปให้ไมโครคอนโทรลเลอร์ แล้วแสดงข้อความให้ดึงบัตรออก และรอรับข้อมูล "Y" จากไมโครคอนโทรลเลอร์ จากนั้นก็จะกลับสู่หน้าจอของสภาวะเริ่มต้นการทำงาน

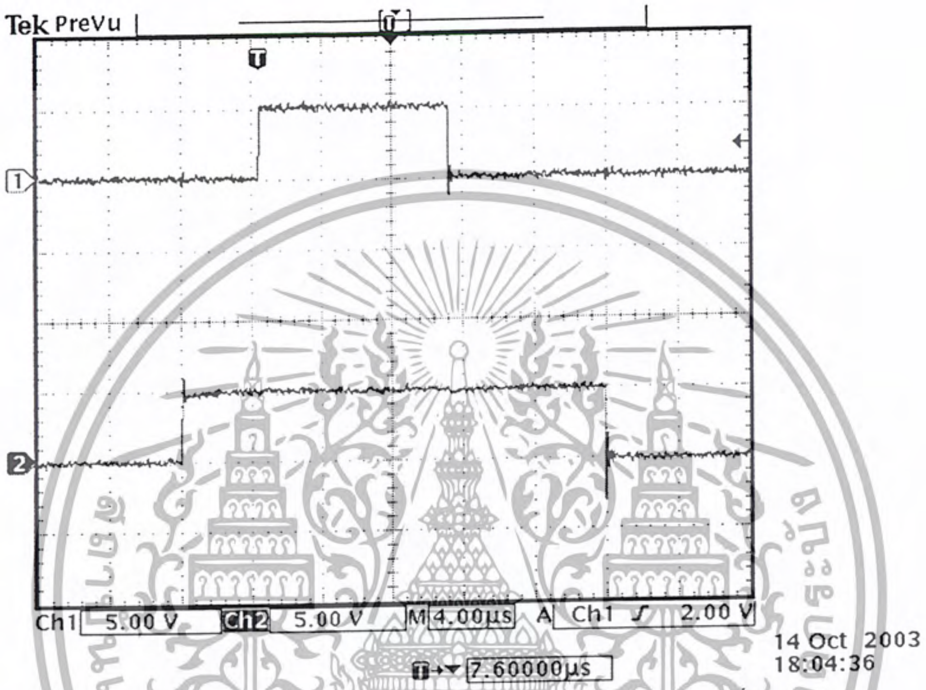
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลองวัดสัญญาณต่าง ๆ จากเครื่องอ่าน-เขียนสมาร์ทการ์ด

4.1.1 ผลการทดลองสัญญาณการรีเซต

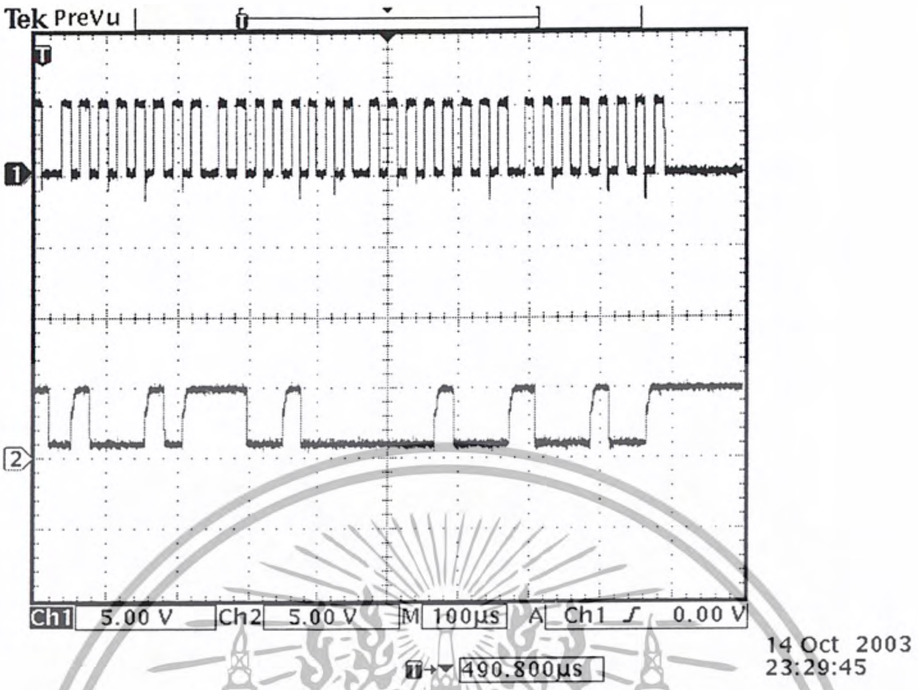


รูปที่ 4.1 แสดงสัญญาณการรีเซต

จากรูปที่ 4.1 แสดงสัญญาณการรีเซตที่วัดได้จากตำแหน่ง CRD RST ของช่องเสียบบัตรสมาร์ทการ์ด เมื่อเปรียบเทียบกับสัญญาณจากตำแหน่ง CRD CLK ของช่องเสียบบัตรสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 ผลการทดลองสัญญาณการตอบรับการรีเซต

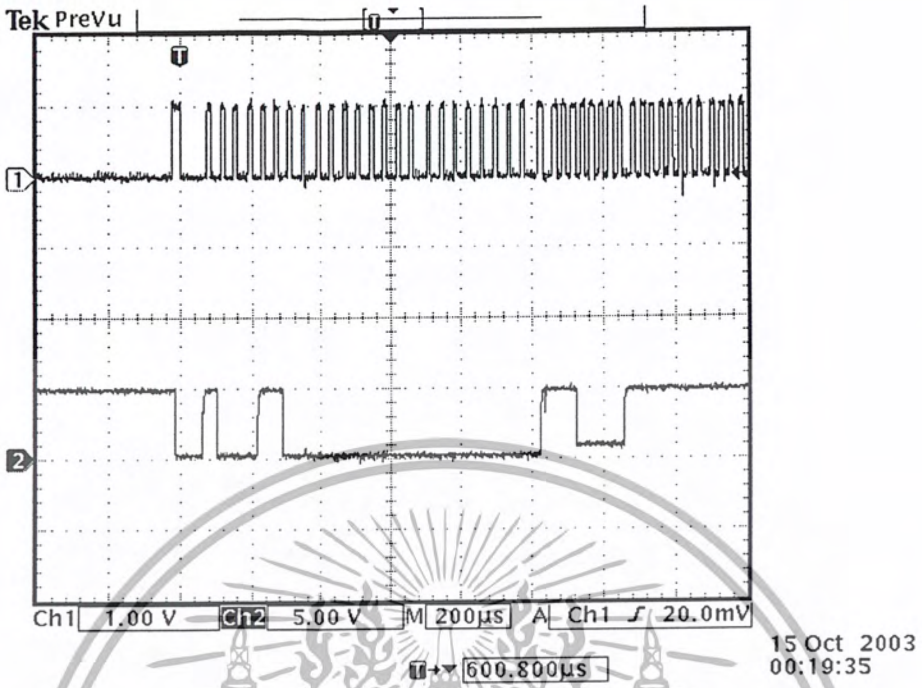


รูปที่ 4.2 แสดงสัญญาณตอบรับการรีเซต

จากรูปที่ 4.2 แสดงสัญญาณตอบรับการรีเซตที่บัตรสมาร์ทการ์ดตอบกลับมาภายหลังจากที่ได้รับสัญญาณการรีเซต โดยวัดที่ตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ทการ์ด ซึ่งเปรียบเทียบกับสัญญาณจากตำแหน่ง CRD CLK ของช่องเสียบบัตรสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ผลการทดลองสัญญาณการเปรียบเทียบรหัสผ่าน

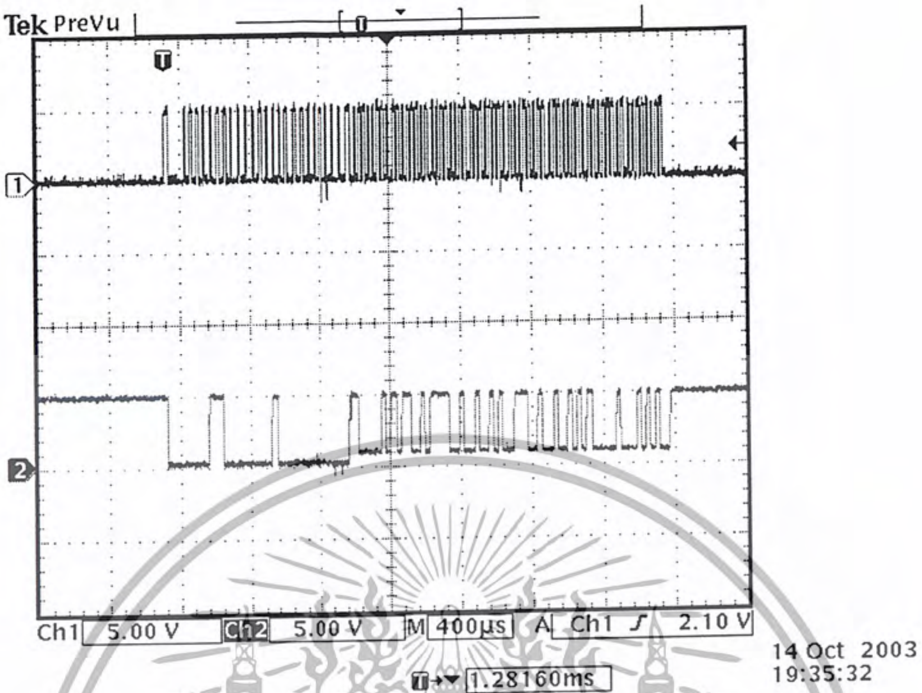


รูปที่ 4.3 แสดงสัญญาณการเปรียบเทียบรหัสผ่าน

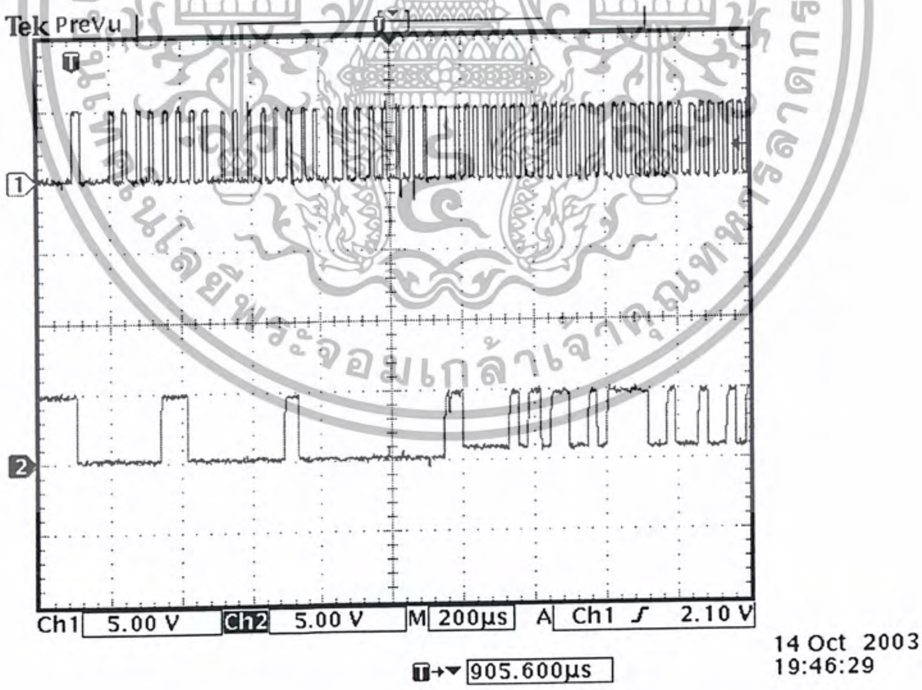
จากรูปที่ 4.3 แสดงสัญญาณการเปรียบเทียบรหัสผ่าน โดยเป็นการส่งคำสั่ง Read Security Memory ไปยังการ์ด ผ่านตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ทการ์ด และการ์ดจะส่งค่าของ Error Counter และข้อมูลอ้างอิงทั้ง 3 ไบต์กลับมาโดยผ่านตำแหน่ง CRD IO เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.4 ผลการทดลองสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก



รูปที่ 4.4 แสดงสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก

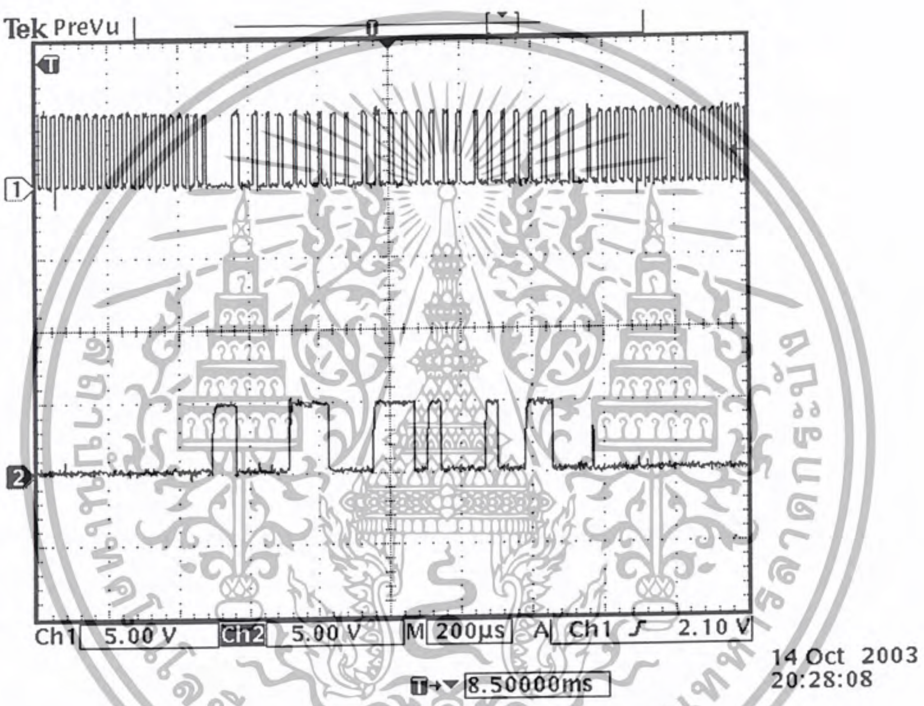


รูปที่ 4.5 แสดงสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก (ภาพขยาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.4 และ 4.5 แสดงสัญญาณการอ่านข้อมูลจากหน่วยความจำหลัก โดยเป็นการส่งคำสั่ง Read Main Memory ไปยังการ์ด ผ่านตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ทการ์ด และการ์ดจะส่งข้อมูลที่อยู่ภายในการ์ดกลับมาตั้งแต่ Address ที่เรากำหนดไปจนถึง Address สุดท้าย (0FFH) ของพื้นที่หน่วยความจำ แต่เนื่องจากโปรแกรมควบคุมการทำงานที่เขียนขึ้นนั้น ได้กำหนดให้อ่านข้อมูลจากบัตรสมาร์ทการ์ดเพียง 8 Address นับจาก Address ที่กำหนด ดังนั้นจะเห็นได้ว่าในรูปที่ 4.4 จะมีข้อมูลที่การ์ดส่งกลับมาโดยผ่านตำแหน่ง CRD IO เพียง 8 ไบต์ หรือ 8 Address เท่านั้น เนื่องจากในตัวโปรแกรมได้หยุดส่งสัญญาณนาฬิกาไปให้การ์ด ทำให้การ์ดไม่ส่งข้อมูลออกมาที่ตำแหน่ง CRD IO อีก

4.1.5 ผลการทดลองสัญญาณการเขียนข้อมูลลงหน่วยความจำหลัก



รูปที่ 4.6 แสดงสัญญาณการเขียนข้อมูลลงหน่วยความจำหลัก

จากรูปที่ 4.6 แสดงสัญญาณการเขียนข้อมูลลงหน่วยความจำหลัก โดยเป็นการส่งคำสั่ง Update Main Memory แบบการลบข้อมูลแล้วเขียนข้อมูลซ้ำ ไปยังการ์ด โดยผ่านตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ทการ์ด

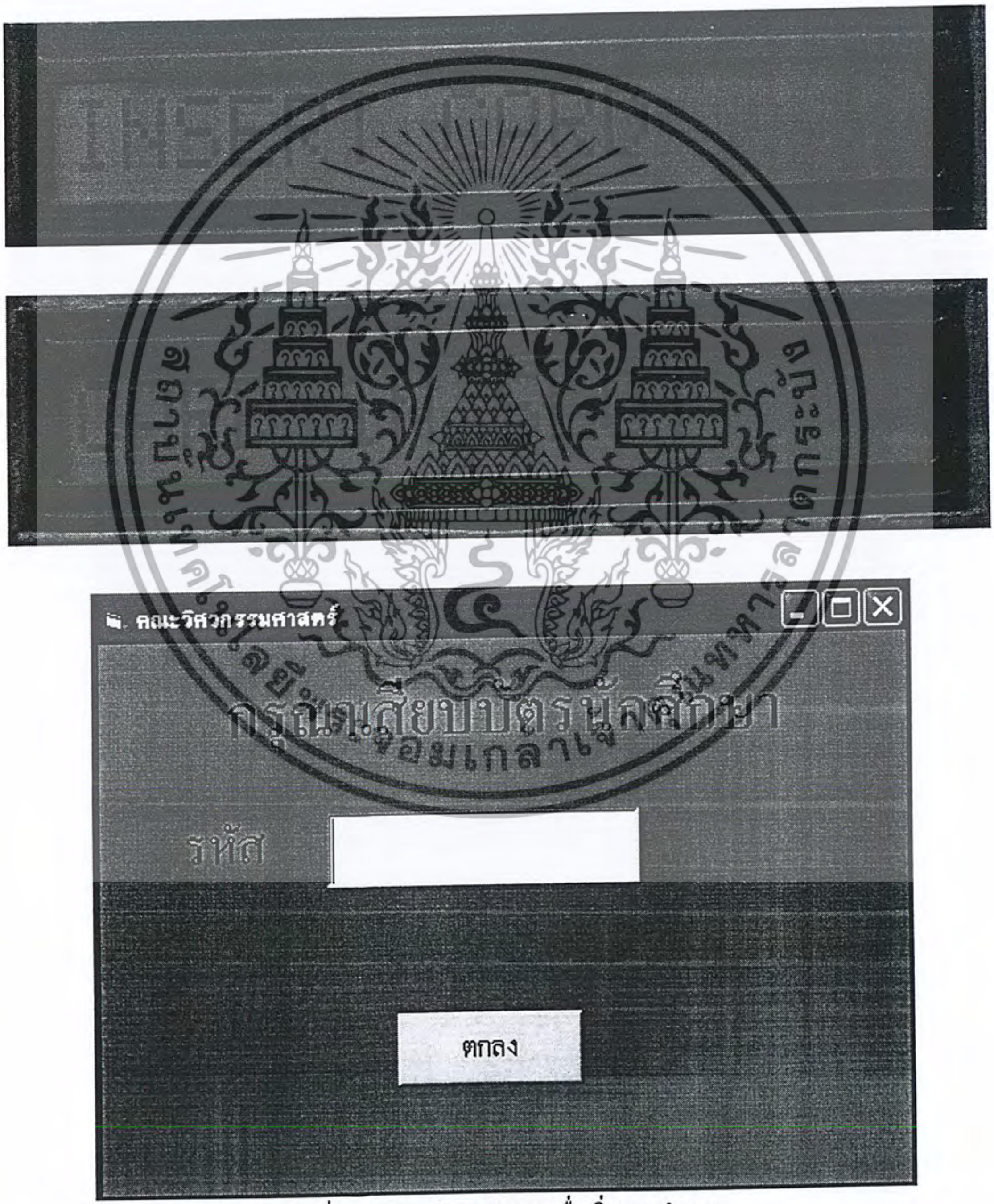
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลองของเครื่องอ่าน-เขียนสมาร์ทการ์ดในส่วนของจอแสดงผลแบบผลึกเหลว และผลการทดลองในหน้าจอคอมพิวเตอร์

4.2.1 เมื่อเริ่มการทำงาน

- ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความ “INSERT CARD” ซึ่งเป็นสภาวะเตรียมพร้อม และเมื่อมีการ์ดเสียบเข้ามายังช่องเสียบบัตรสมาร์ทการ์ด และทำการตรวจสอบแล้วว่าการ์ดถูกต้องก็จะแสดงข้อความ “CARD IN SLOT”

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง “กรุณาเสียบบัตรนักศึกษา”

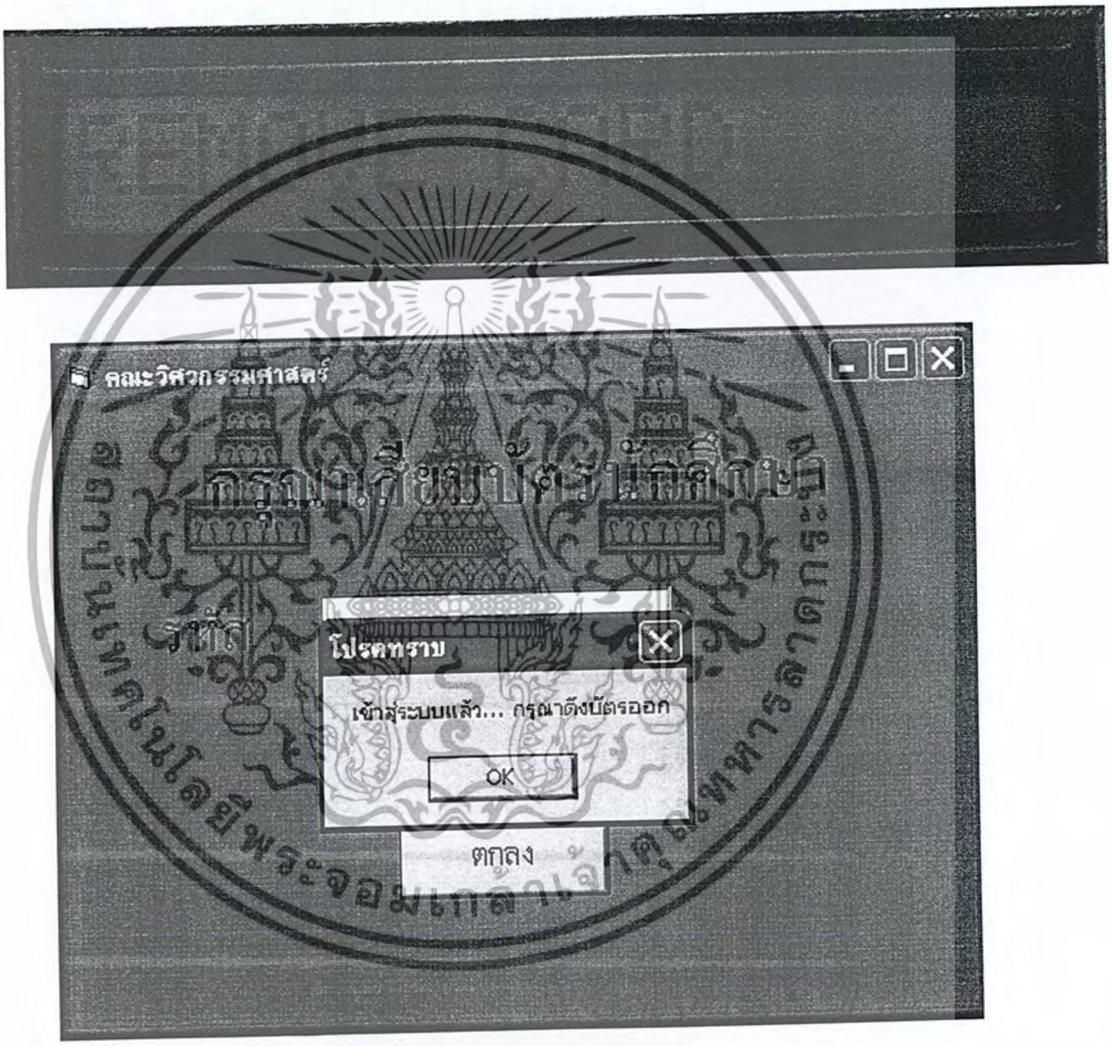


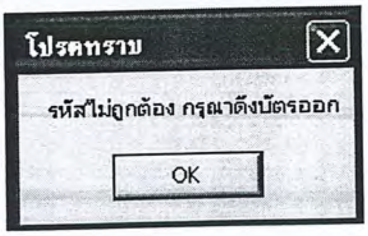
รูปที่ 4.7 แสดงส่วนแสดงผลเมื่อเริ่มการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 เมื่อทำการใส่รหัสผ่าน

- หากตรวจสอบแล้วพบว่ารหัสผ่านถูกต้อง ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความ "REMOVE CARD" เพื่อให้ผู้ใช้งานดึงบัตรออก และที่หน้าจอคอมพิวเตอร์ จะแสดงหน้าต่าง "เข้าสู่ระบบแล้ว... กรุณาดึงบัตรออก"
- หากตรวจสอบแล้วพบว่ารหัสผ่านไม่ถูกต้อง ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความ "REMOVE CARD" เพื่อให้ผู้ใช้งานดึงบัตรออก และที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "รหัสไม่ถูกต้อง กรุณาดึงบัตรออก"





รูปที่ 4.8 แสดงส่วนแสดงผลเมื่อใส่รหัสผ่าน

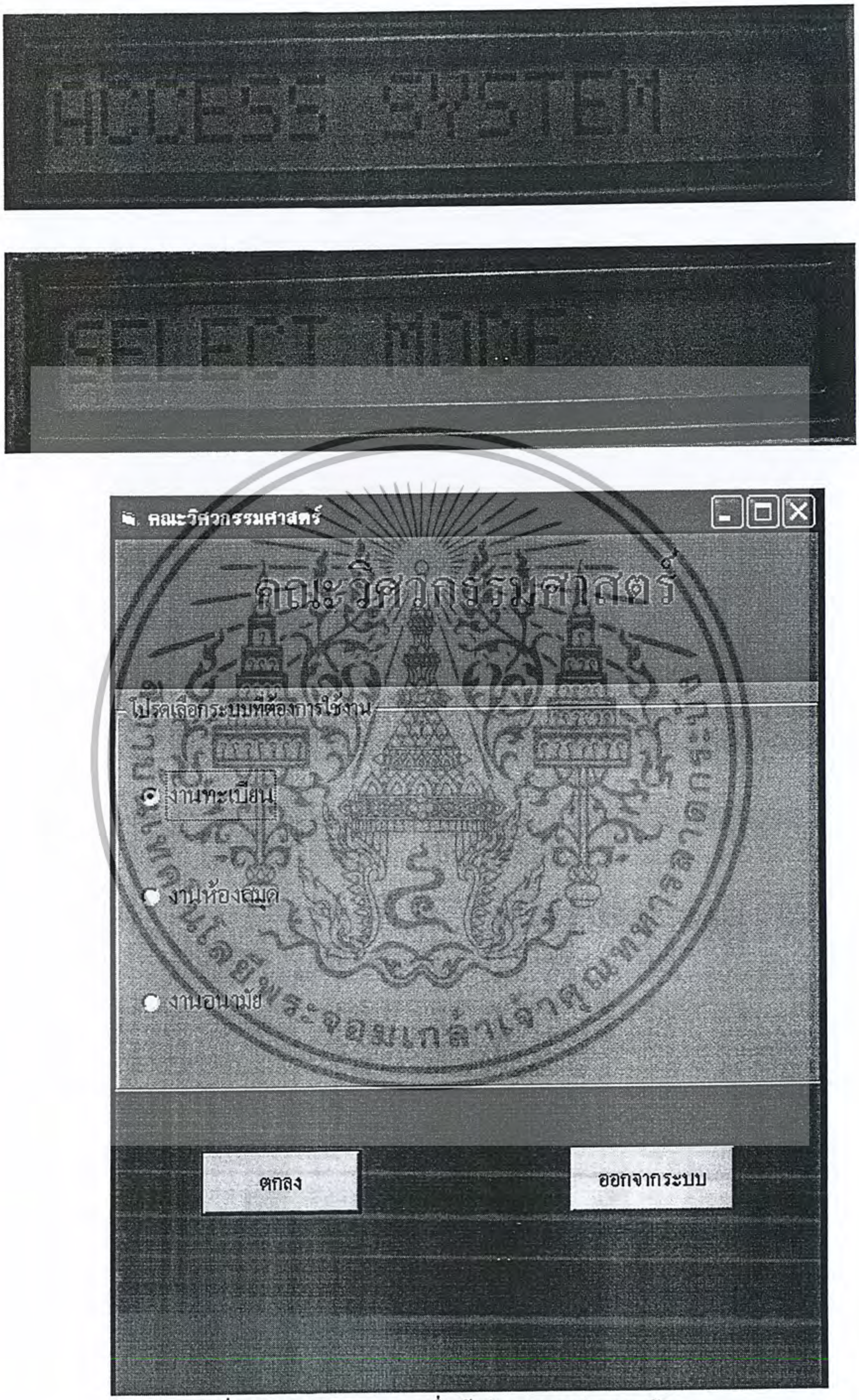
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 เมื่อเข้าสู่ระบบการทำงานหลัก

- ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความ "ACCESS SYSTEM" และข้อความ "SELECT MODE"

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "คณะวิศวกรรมศาสตร์" เพื่อให้ผู้ใช้งานเลือกระบบที่ต้องการใช้งาน ซึ่งประกอบไปด้วยงานทะเบียน งานห้องสมุด และงานอนามัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

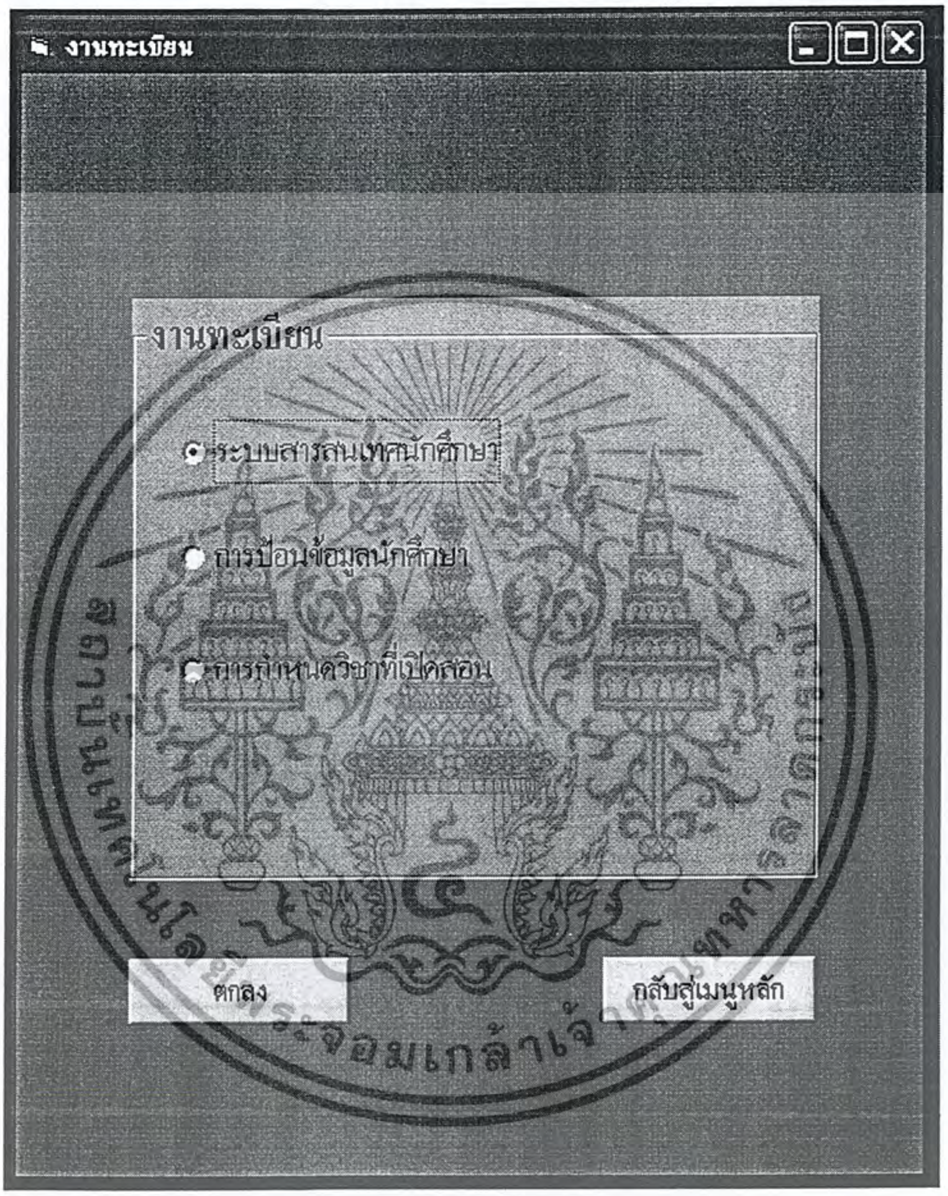


รูปที่ 4.9 แสดงส่วนแสดงผลเมื่อเข้าสู่ระบบการทำงานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 เมื่อเข้าสู่งานทะเบียน

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "งานทะเบียน" เพื่อให้ผู้ใช้งานเลือกระบบที่ต้องการใช้งาน ซึ่งประกอบไปด้วยระบบสารสนเทศนักศึกษา การป้อนข้อมูลนักศึกษา และการกำหนดวิชาที่เปิดสอน

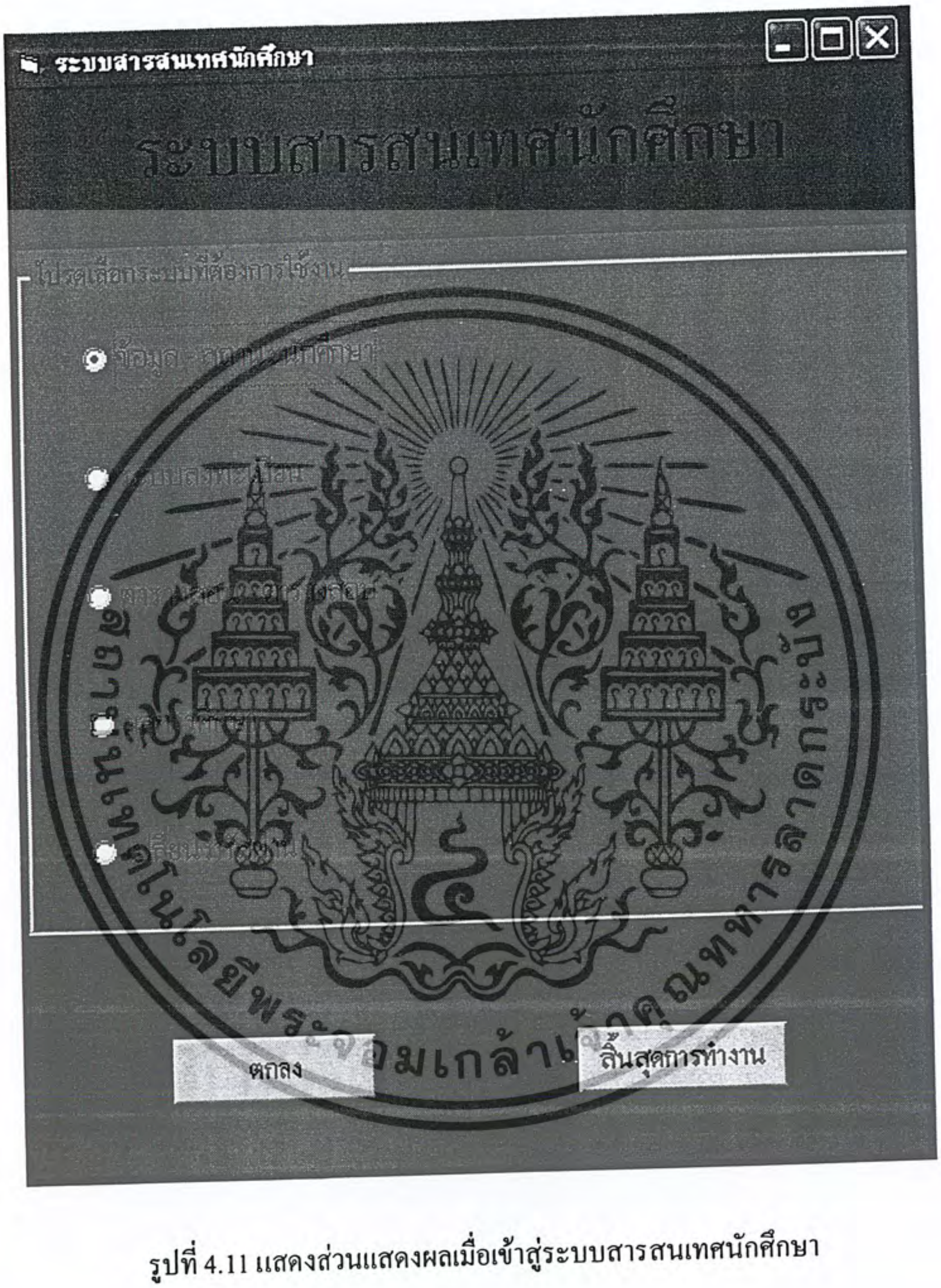


รูปที่ 4.10 แสดงส่วนแสดงผลเมื่อเข้าสู่งานทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.1 เมื่อเข้าสู่ระบบสารสนเทศนักศึกษา

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "ระบบสารสนเทศนักศึกษา" เพื่อให้ผู้ใช้งานเลือกระบบที่ต้องการใช้งาน ซึ่งประกอบไปด้วยข้อมูล-สถานะนักศึกษา ระบบลงทะเบียน ตารางสอน-ตารางสอบ ผลการศึกษา และเปลี่ยนรหัสผ่าน

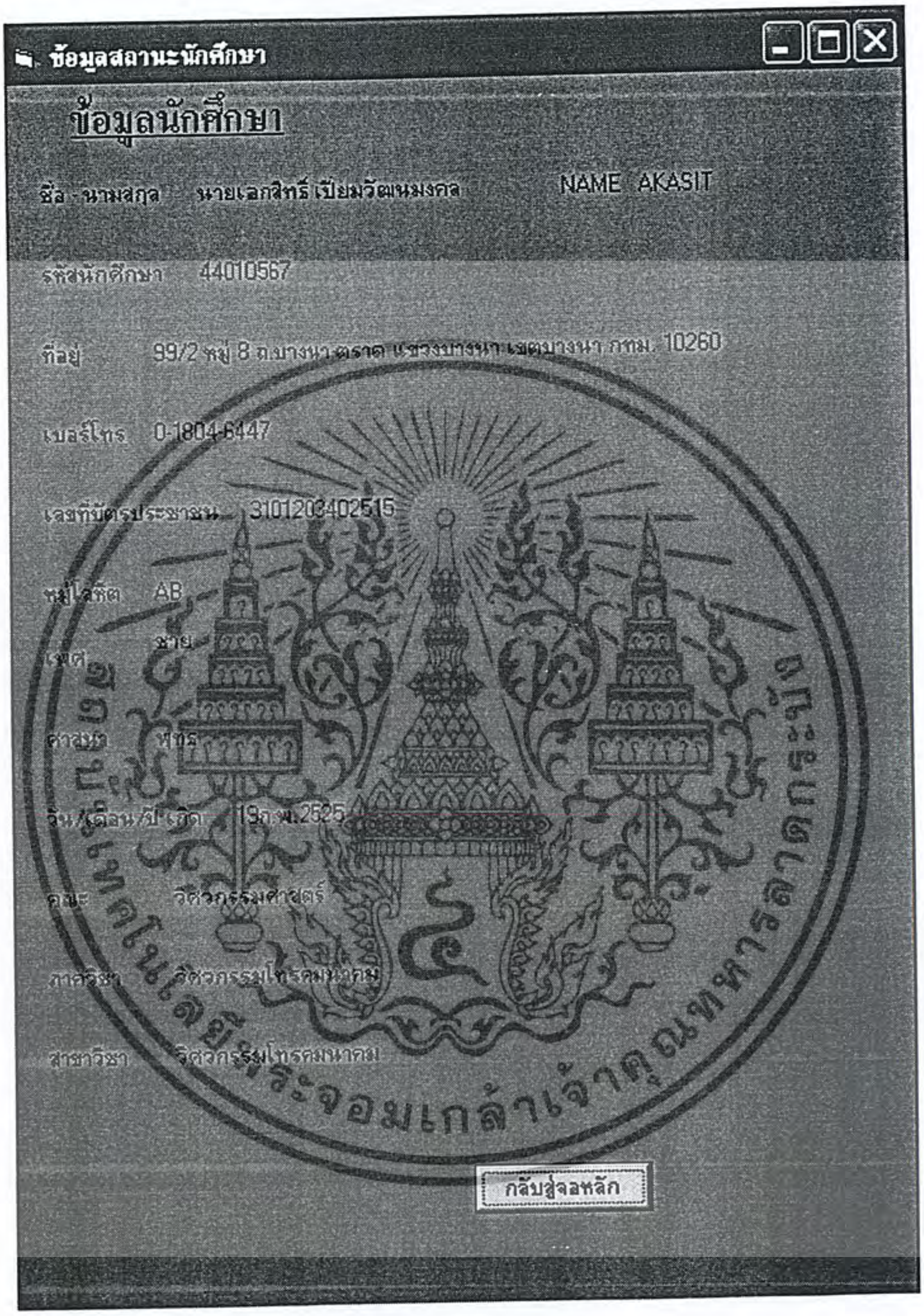


รูปที่ 4.11 แสดงส่วนแสดงผลเมื่อเข้าสู่ระบบสารสนเทศนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.1.1 เมื่อเข้าสู่ข้อมูล-สถานะนักศึกษา

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "ข้อมูลสถานะนักศึกษา" ซึ่งแสดงรายละเอียดของนักศึกษาตามรหัสนักศึกษาที่อ่านได้จากบัตรสมาร์ทการ์ดที่เสียบเข้ามา

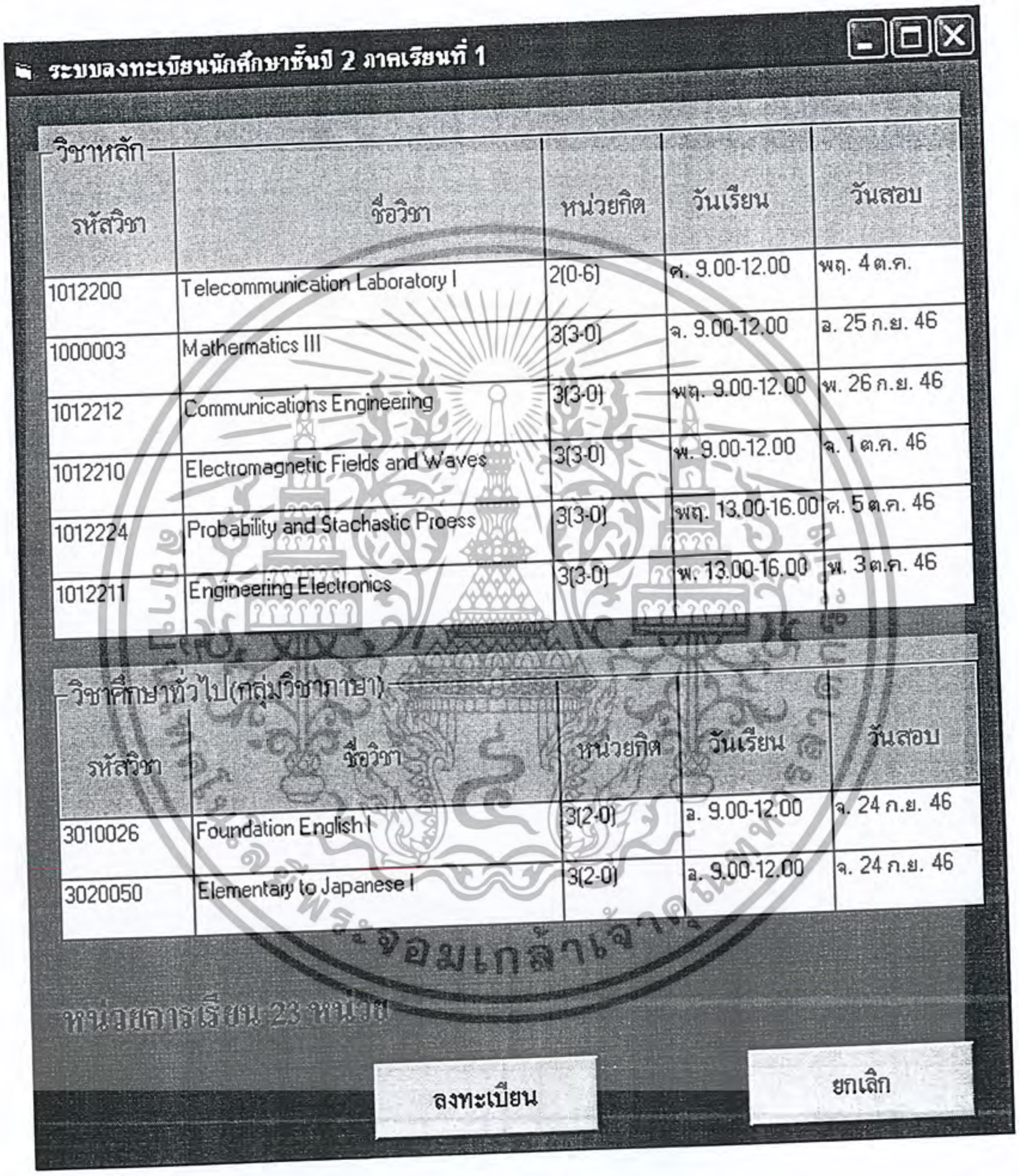


รูปที่ 4.12 แสดงส่วนแสดงผลเมื่อเข้าสู่ข้อมูล-สถานะนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.1.2 เมื่อเข้าสู่ระบบลงทะเบียน

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "ระบบลงทะเบียน" และเมื่อเลือก "ลงทะเบียน" แล้วตรวจสอบว่าพบว่าได้ลงทะเบียนเกินจำนวน 22 หน่วย ก็จะแสดงหน้าต่างเตือนขึ้นมา แต่ถ้าหากตรวจสอบแล้วพบว่าไม่เกิน 22 หน่วย ก็จะแสดงหน้าต่างที่แสดงรายละเอียดค่าลงทะเบียนพร้อมกับวิชาที่ลงทะเบียนและตารางสอน-ตารางสอบของวิชานั้น ๆ


ระบบลงทะเบียนนักศึกษาชั้นปี 2 ภาคเรียนที่ 1

วิชาหลัก

รหัสวิชา	ชื่อวิชา	หน่วยกิต	วันเรียน	วันสอบ
1012200	Telecommunication Laboratory I	2(0-6)	ศ. 9.00-12.00	พฤ. 4 ต.ค.
1000003	Mathematics III	3(3-0)	จ. 9.00-12.00	อ. 25 ก.ย. 46
1012212	Communications Engineering	3(3-0)	พฤ. 9.00-12.00	พ. 26 ก.ย. 46
1012210	Electromagnetic Fields and Waves	3(3-0)	พ. 9.00-12.00	จ. 1 ต.ค. 46
1012224	Probability and Stachastic Proess	3(3-0)	พฤ. 13.00-16.00	ศ. 5 ต.ค. 46
1012211	Engineering Electronics	3(3-0)	พ. 13.00-16.00	พ. 3 ต.ค. 46

วิชาศึกษาทั่วไป(กลุ่มวิชาภาษา)

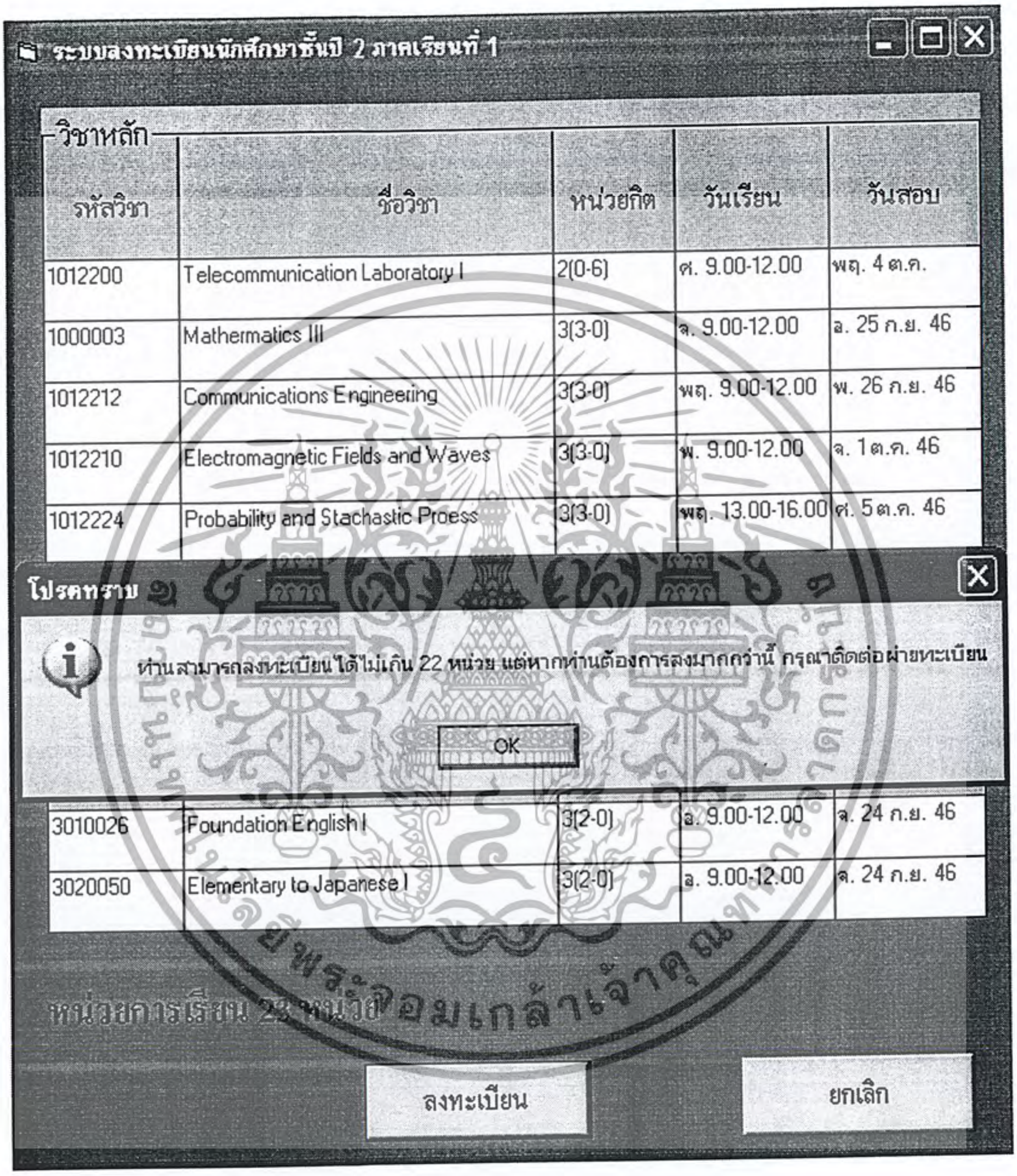
รหัสวิชา	ชื่อวิชา	หน่วยกิต	วันเรียน	วันสอบ
3010026	Foundation English I	3(2-0)	อ. 9.00-12.00	จ. 24 ก.ย. 46
3020050	Elementary to Japanese I	3(2-0)	อ. 9.00-12.00	จ. 24 ก.ย. 46

หน่วยการเรียน 23 หน่วย

ลงทะเบียน ยกเลิก

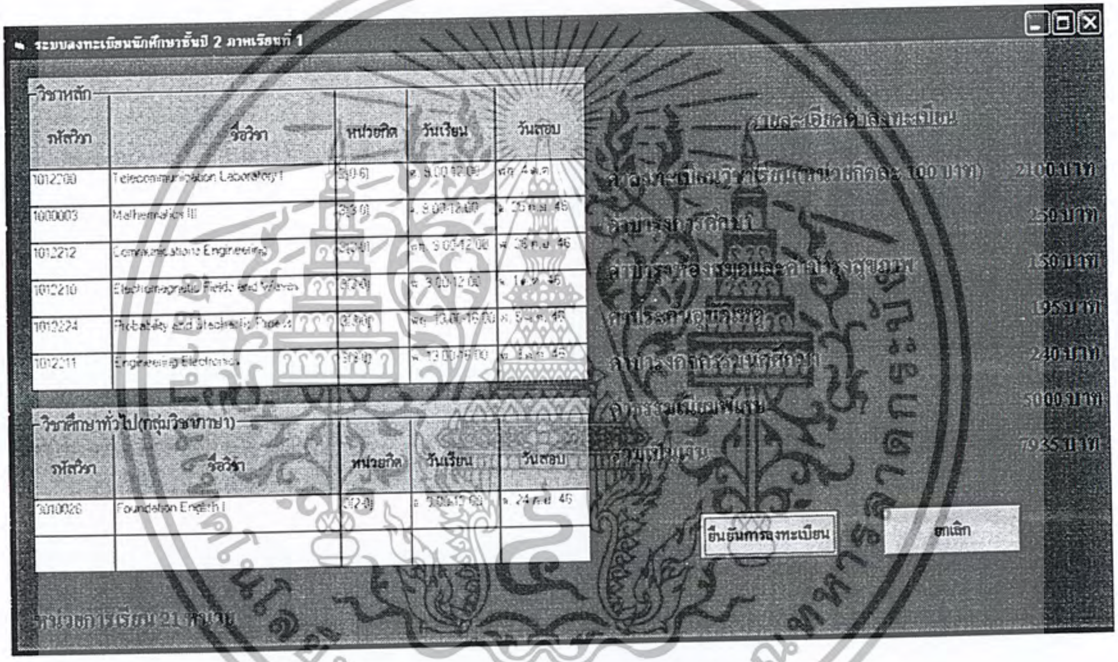
รูปที่ 4.13 แสดงส่วนแสดงผลเมื่อเข้าสู่ระบบลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 แสดงส่วนแสดงผลเมื่อลงทะเบียนเกิน 22 หน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

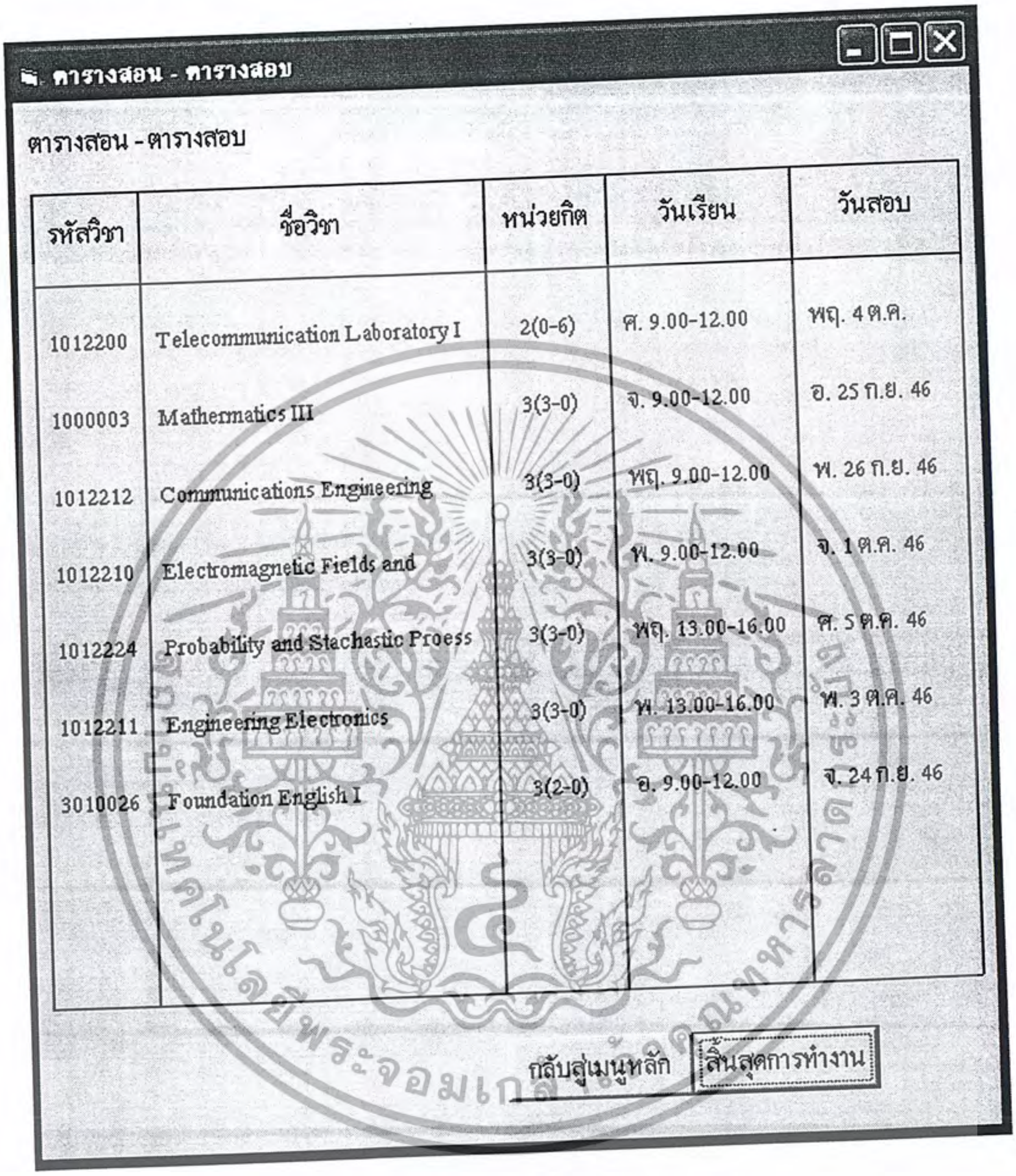


รูปที่ 4.15 แสดงส่วนแสดงผลที่แสดงรายละเอียดของการลงทะเบียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.1.3 เมื่อเข้าสู่ตารางสอน-ตารางสอบ

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "ตารางสอน-ตารางสอบ" ของวิชาที่เลือกลงทะเบียนในภาคการศึกษานั้น ๆ



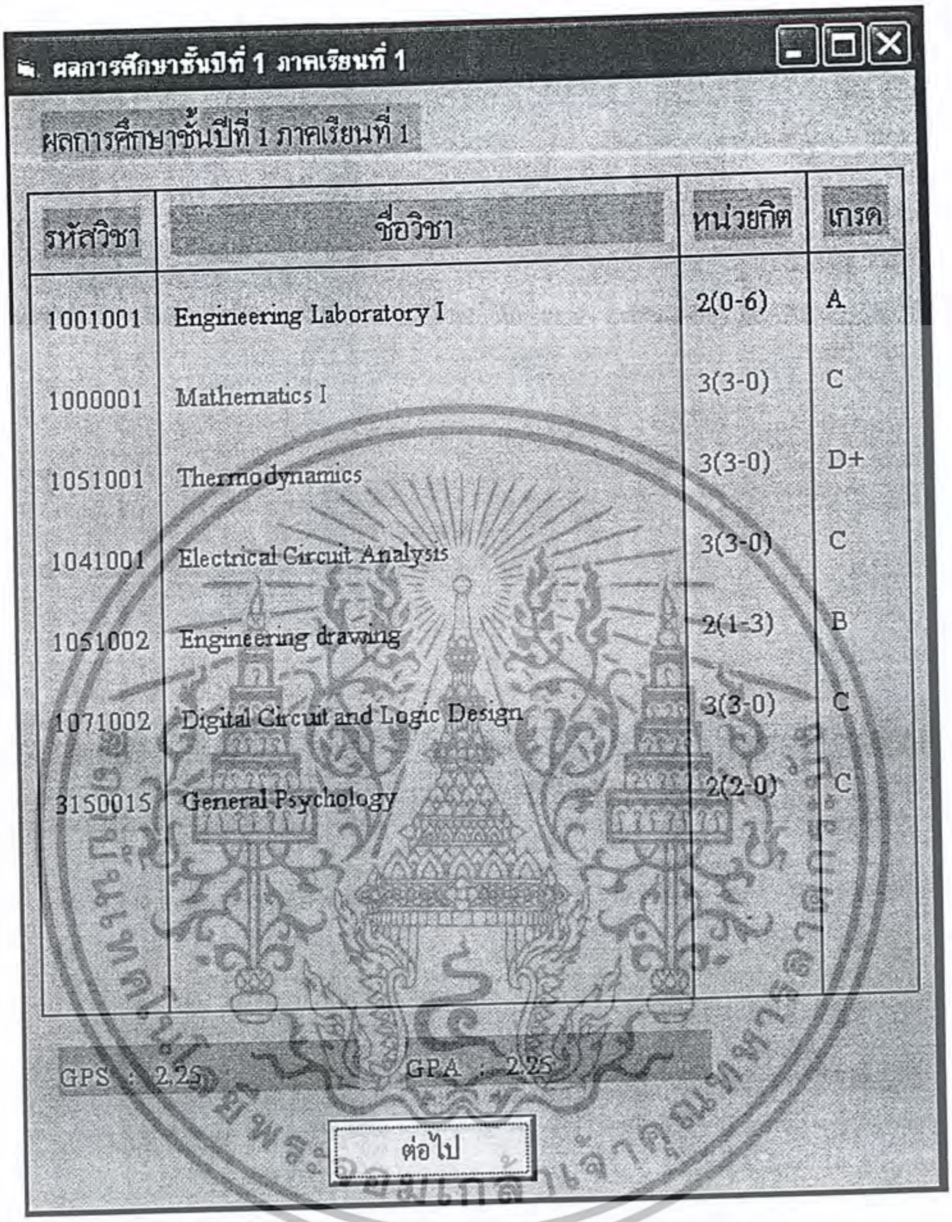
รหัสวิชา	ชื่อวิชา	หน่วยกิต	วันเรียน	วันสอบ
1012200	Telecommunication Laboratory I	2(0-6)	ศ. 9.00-12.00	พฤ. 4 ต.ค.
1000003	Mathermatics III	3(3-0)	จ. 9.00-12.00	อ. 25 ก.ย. 46
1012212	Communications Engineering	3(3-0)	พฤ. 9.00-12.00	พ. 26 ก.ย. 46
1012210	Electromagnetic Fields and	3(3-0)	พ. 9.00-12.00	จ. 1 ต.ค. 46
1012224	Probability and Stachastic Proess	3(3-0)	พฤ. 13.00-16.00	ศ. 5 ต.ค. 46
1012211	Engineering Electronics	3(3-0)	พ. 13.00-16.00	พ. 3 ต.ค. 46
3010026	Foundation English I	3(2-0)	อ. 9.00-12.00	จ. 24 ก.ย. 46

รูปที่ 4.16 แสดงส่วนแสดงผลเมื่อเข้าสู่ตารางสอน-ตารางสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.1.4 เมื่อเข้าสู่ผลการศึกษา

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง “ผลการศึกษา”



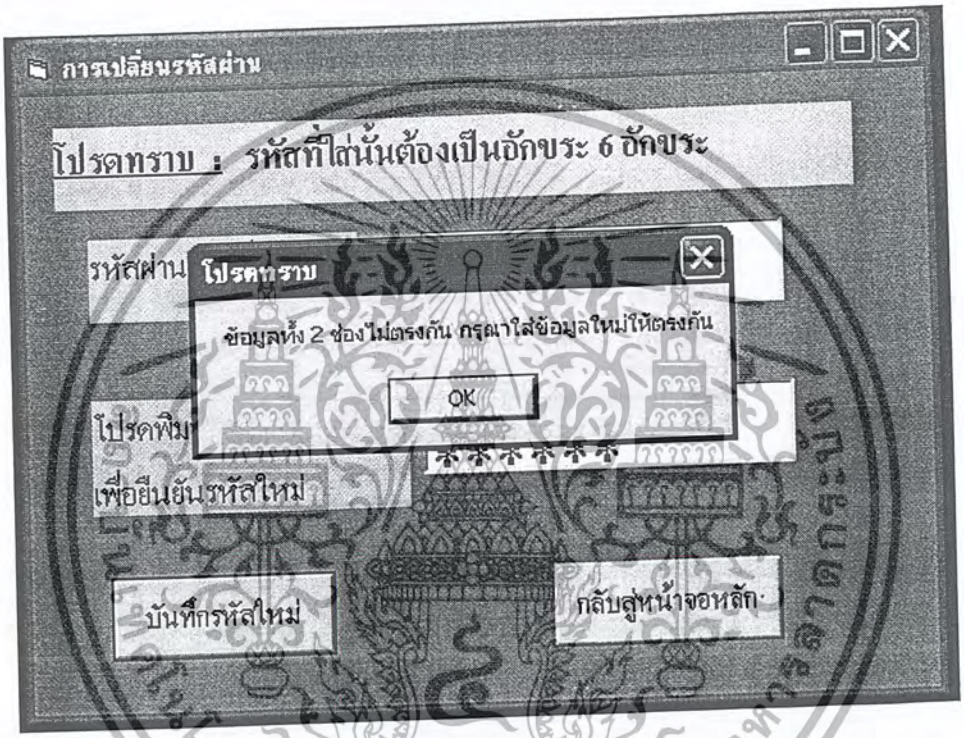
รูปที่ 4.17 แสดงส่วนแสดงผลเมื่อเข้าสู่ผลการศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4.1.5 เมื่อเข้าสู่เปลี่ยนรหัสผ่าน

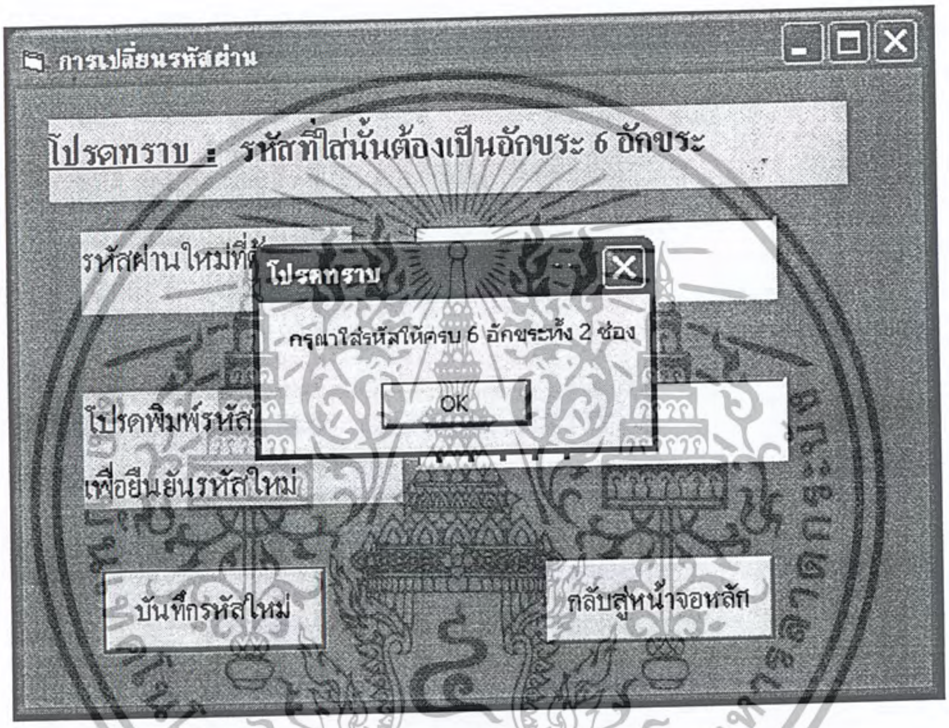
- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "การเปลี่ยนรหัสผ่าน" และหากทำการกรอกรหัสผ่านใหม่ในช่องข้อมูลทั้ง 2 ช่องเรียบร้อยแล้วตรวจพบว่ารหัสผ่านที่กรอกในทั้ง 2 ช่องนั้นไม่ตรงกัน หรือไม่ครบ 6 อักขระ ก็จะแสดงข้อความเตือนขึ้นมา แต่ถ้าหากตรวจสอบแล้วพบว่ารหัสผ่านใหม่ที่กรอกในทั้ง 2 ช่องนั้นตรงกันและครบ 6 อักขระ ก็จะแสดงข้อความ "ระบบได้ทำการเปลี่ยนรหัสผ่านให้ท่านแล้ว" ขึ้นมา

- ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความ "P.W. IS RECORDED" เมื่อได้ทำการบันทึกรหัสผ่านใหม่เรียบร้อยแล้ว



รูปที่ 4.18 แสดงส่วนแสดงผลเมื่อกรอกรหัสผ่านใหม่ในช่องข้อมูลทั้ง 2 ช่องไม่ตรงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 แสดงส่วนแสดงผลเมื่อกรอกรหัสผ่านใหม่ไม่ครบ 6 อักขระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

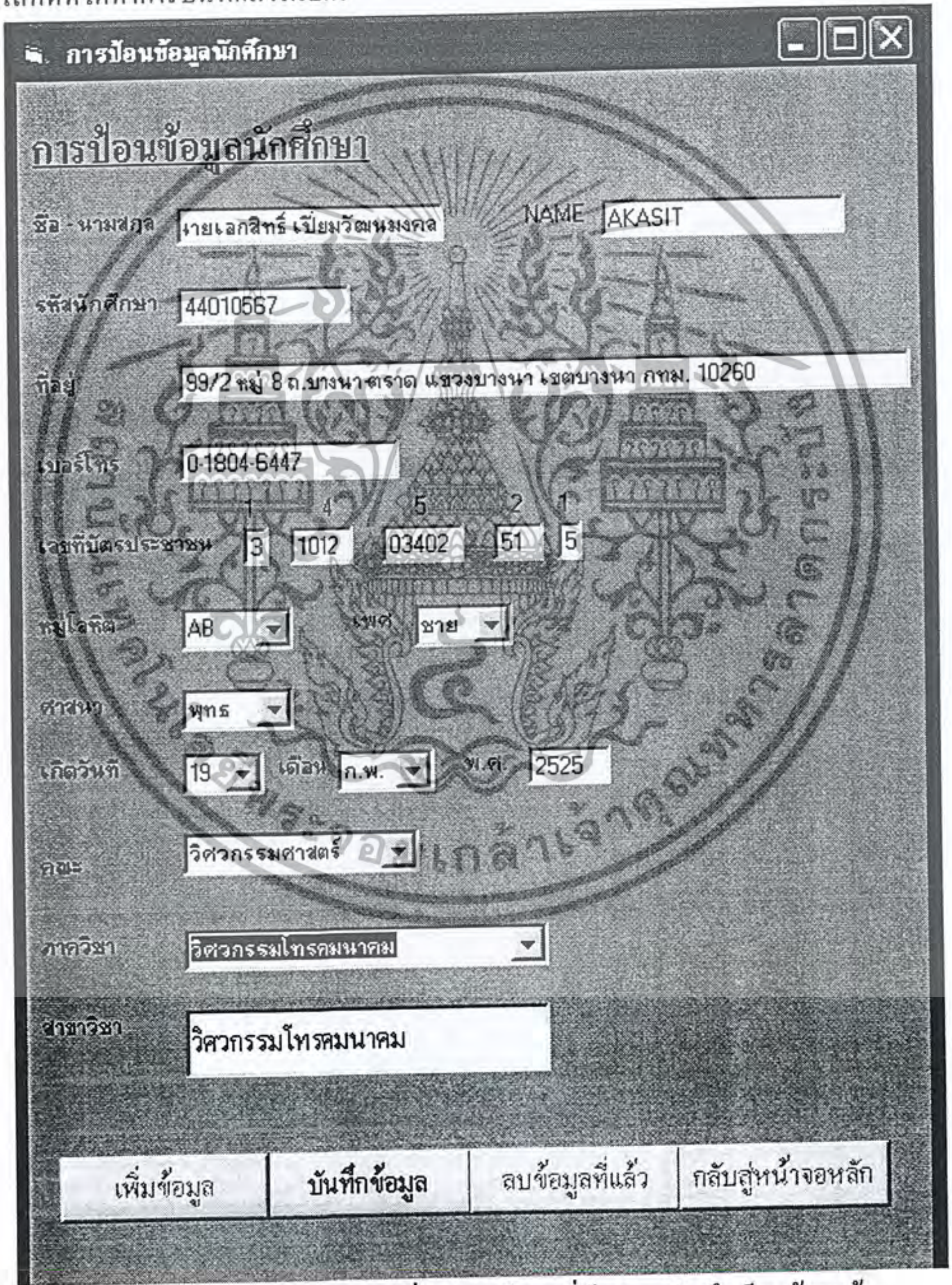
[redacted]

รูปที่ 4.20 แสดงส่วนแสดงผลเมื่อรหัสผ่านใหม่ที่กรอกนั้นถูกต้องตามที่กำหนด และระบบได้ทำการเปลี่ยนรหัสผ่านให้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

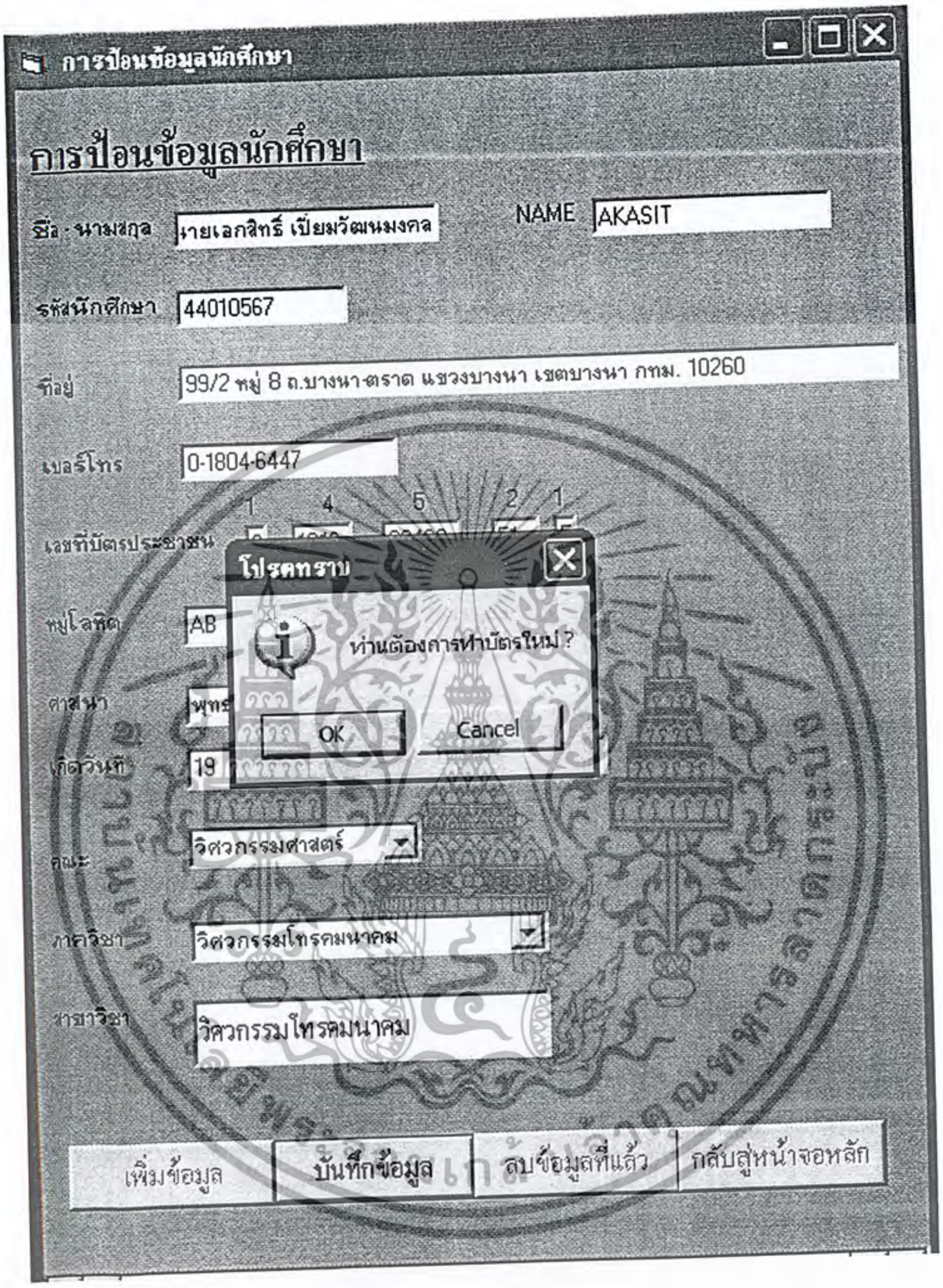
4.2.4.2 เมื่อเข้าสู่การป้อนข้อมูลนักศึกษา

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "การป้อนข้อมูลนักศึกษา" และเมื่อเลือกเพิ่มข้อมูลแล้วจะปรากฏหน้าต่างที่ให้กรอกข้อมูลที่ต้องการบันทึก และต่อมาหากเลือกบันทึกข้อมูลจะแสดงข้อความเพื่อถามความแน่ใจว่า "ท่านต้องการทำบัตรใหม่ ?" และหากเลือก "o.k." ก็จะแสดงข้อความให้เสียบบัตรที่ต้องการบันทึกข้อมูล และเมื่อทำการบันทึกข้อมูลเรียบร้อยแล้ว ก็จะแสดงข้อความ "ข้อมูลได้ถูกบันทึกเรียบร้อยแล้ว ดังแสดงผลบนจอ LCD"
- ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดนั้น เมื่อทำการบันทึกข้อมูลลงในบัตรสมาร์ทการ์ดเรียบร้อยแล้วก็จะแสดงข้อความว่า "WRITE SUCCESS" และแสดงข้อมูลรหัสนักศึกษา ชื่อ เลขประจำตัวประชาชน และหมู่โลหิตที่ได้ทำการบันทึกลงในบัตร



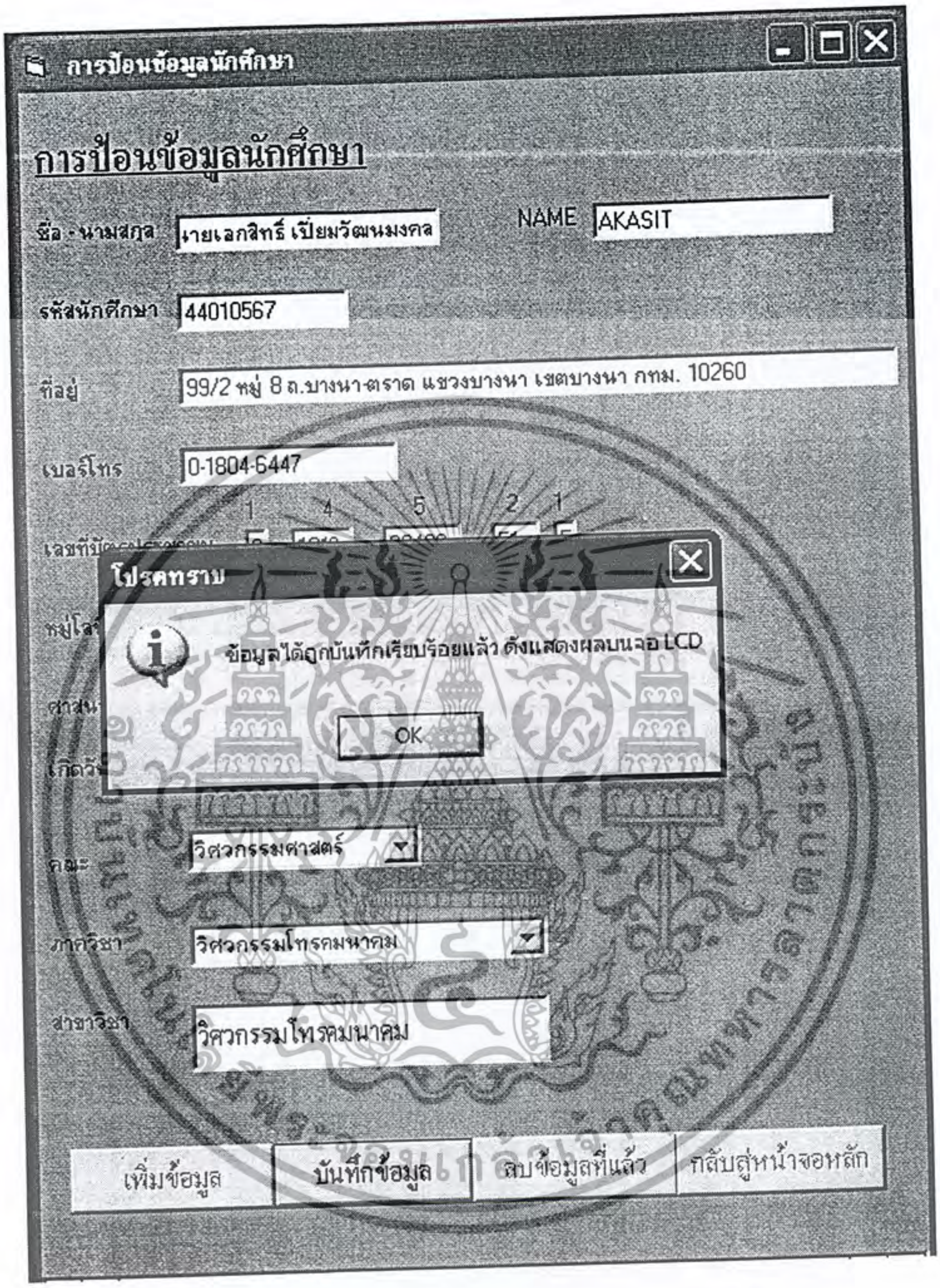
รูปที่ 4.21 แสดงส่วนแสดงผลเมื่อกรอกข้อมูลที่ต้องการบันทึกเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



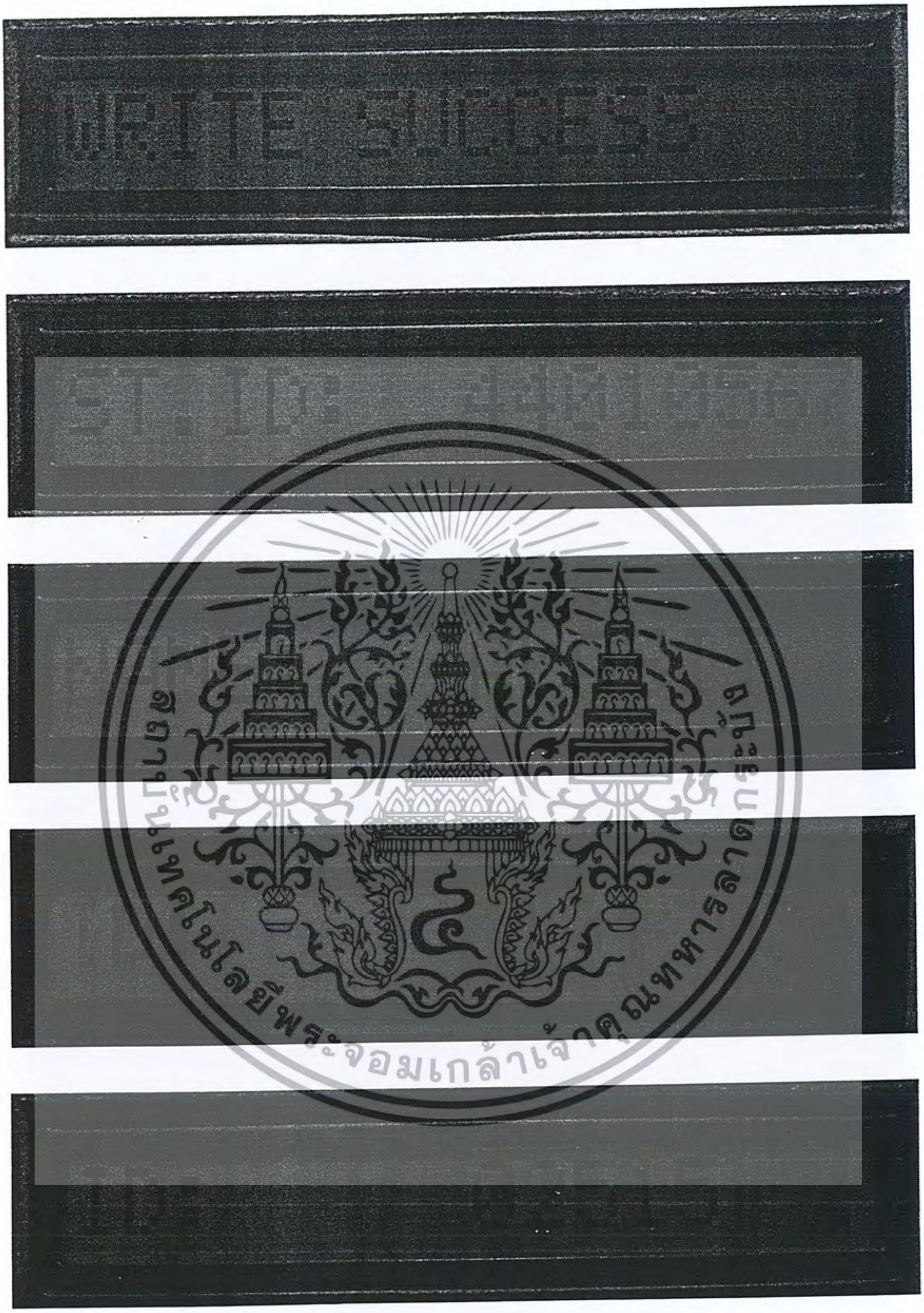
รูปที่ 4.22 แสดงส่วนแสดงผลเพื่อถามความแน่ใจว่าต้องการทำบัตรใหม่หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



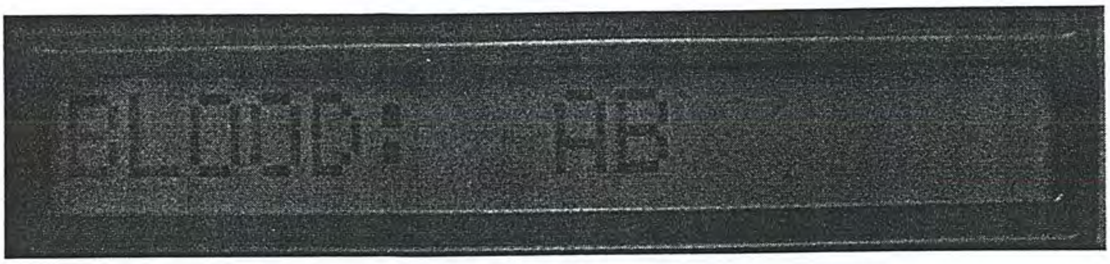
รูปที่ 4.23 แสดงส่วนแสดงผลเมื่อทำการบันทึกข้อมูลเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 แสดงส่วนแสดงผลเมื่อบันทึกข้อมูลลงในบัตรสมาร์ทการ์ดเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



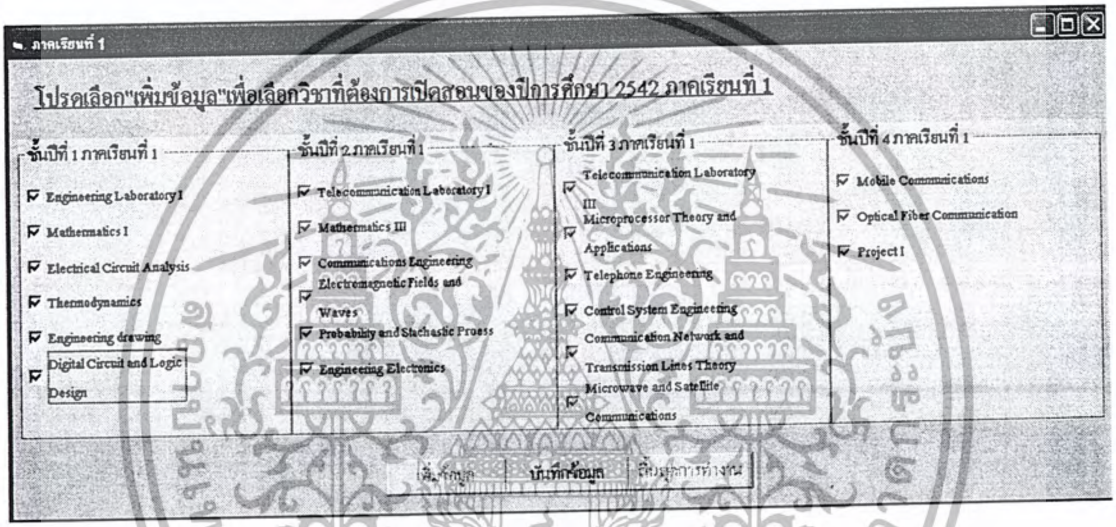
รูปที่ 4.25 แสดงส่วนแสดงผลเมื่อบันทึกข้อมูลลงในบัตรสมาร์ทการ์ดเรียบร้อยแล้ว (ต่อ)

4.2.4.3 เมื่อเข้าสู่การกำหนดวิชาที่เปิดสอน

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "การกำหนดวิชาที่เปิดสอน" โดยให้ผู้ใช้งานกรอกปีการศึกษา และเลือกภาคการศึกษาที่ต้องการ ซึ่งเมื่อทำการกรอกเรียบร้อยแล้ว ที่หน้าจอคอมพิวเตอร์ก็จะแสดงหน้าต่างรายชื่อวิชาต่าง ๆ เพื่อให้เลือกวิชาที่เปิดสอน

รูปที่ 4.26 แสดงส่วนแสดงผลเมื่อเข้าสู่การกำหนดวิชาที่เปิดสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

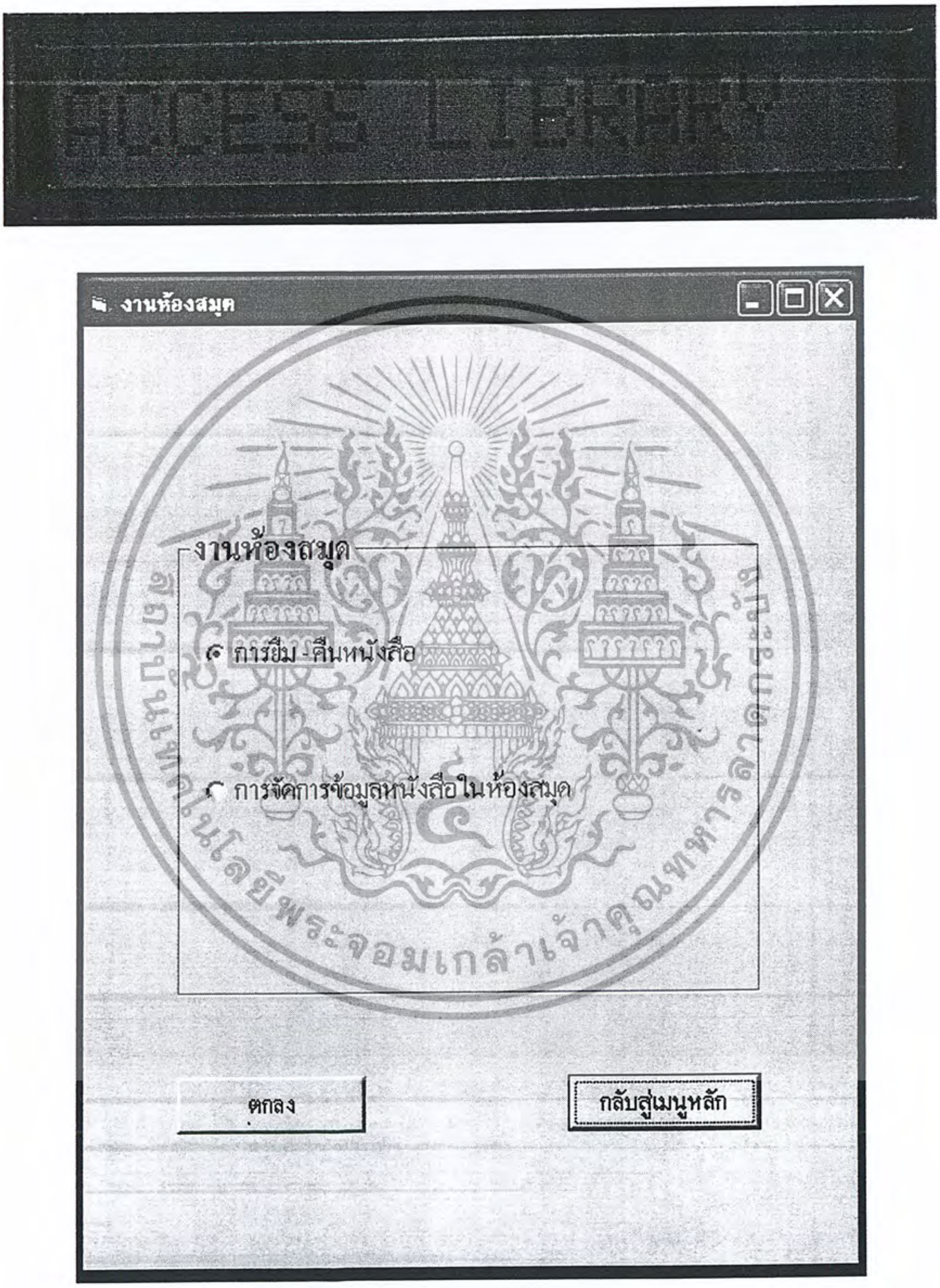


รูปที่ 4.27 แสดงส่วนแสดงผลที่ให้กำหนดวิชาที่เปิดสอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 เมื่อเข้าสู่งานห้องสมุด

- ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความว่า “ACCESS LIBRARY”
- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง “งานห้องสมุด”

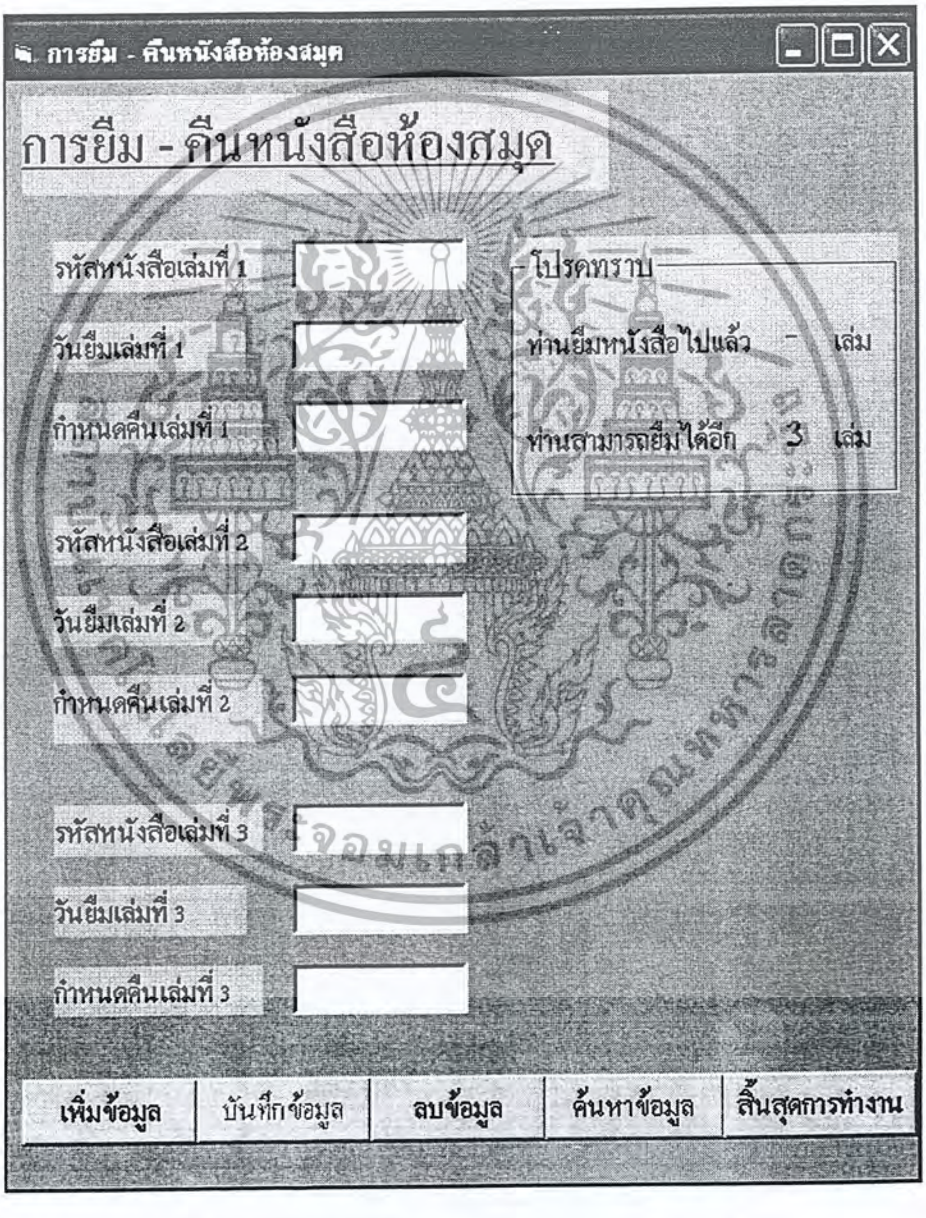


รูปที่ 3.28 แสดงส่วนแสดงผลเมื่อเข้าสู่งานห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

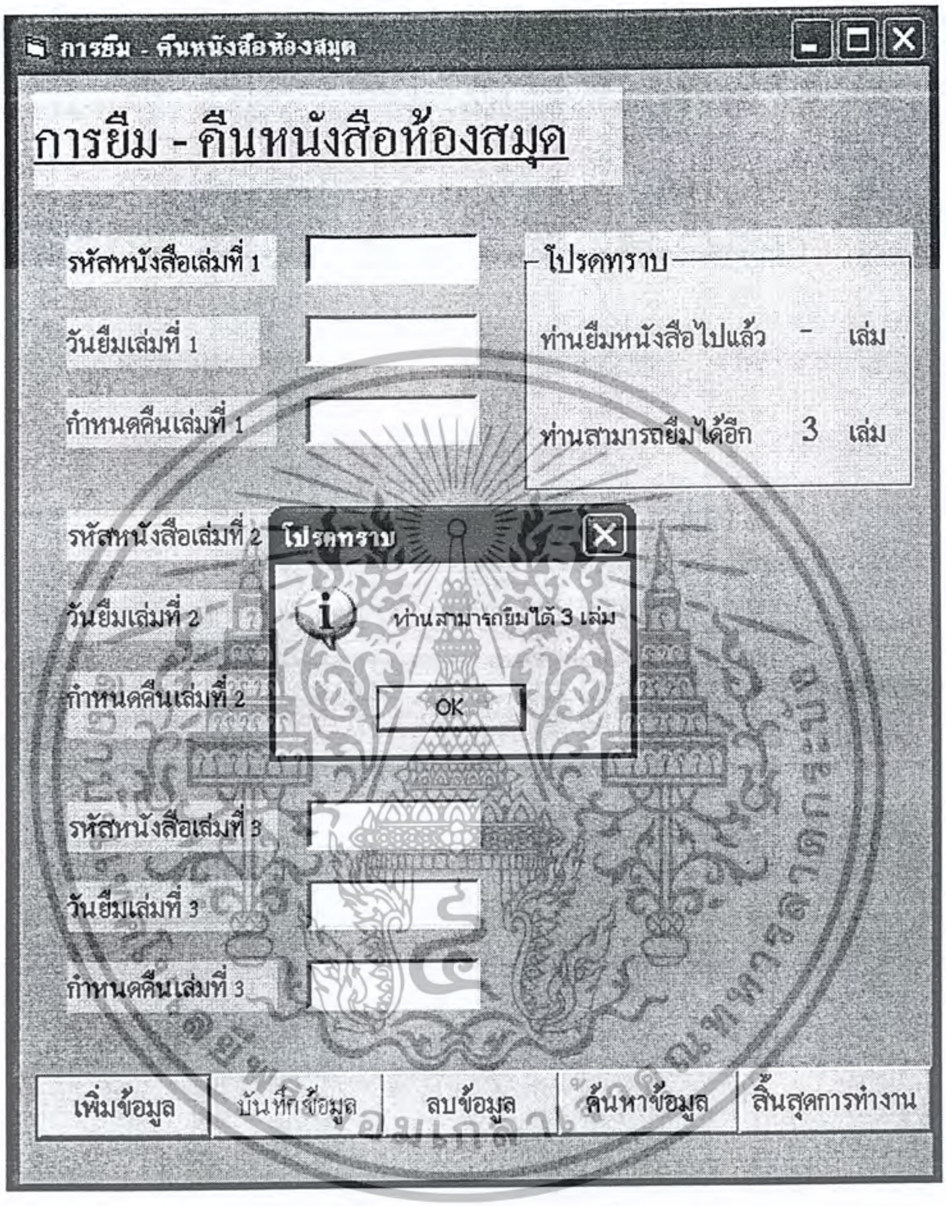
4.2.5.1 เมื่อเข้าสู่การยืม-คืนหนังสือ

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "การยืม-คืนหนังสือ" ซึ่งจะแสดงรายชื่อหนังสือของผู้ยืมตามรหัสนักศึกษาที่อ่านได้จากบัตรสมาร์ทการ์ดที่เสียบเข้ามา และเมื่อเลือก "เพิ่มข้อมูล" ก็จะแสดงข้อความบอกจำนวนหนังสือที่สามารถยืมได้อีก และเมื่อทำการระบุชื่อหนังสือที่ต้องการยืมแล้ว และเลือก "บันทึกข้อมูล" ก็จะแสดงหน้าต่างชื่อหนังสือที่ทำการบันทึกข้อมูลการยืมแล้ว หรือถ้าหากเลือก "ลบข้อมูล" ก็จะแสดงข้อความและหน้าต่างที่ให้ทำการเลือกหนังสือเล่มที่นำมาคืน หรือถ้าหากเลือก "ค้นหาข้อมูล" ก็จะแสดงรายชื่อหนังสือที่ผู้ยืมตามรหัสนักศึกษาที่อ่านได้จากบัตรสมาร์ทการ์ดที่เสียบเข้ามาทำการยืมไป



รูปที่ 4.29 แสดงส่วนแสดงผลเมื่อเข้าสู่การยืม-คืนหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 แสดงส่วนแสดงผลเมื่อเลือก “เพิ่มข้อมูล” ในหน้าต่างการยืม-คืนหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.31 แสดงส่วนแสดงผลเมื่อทำการบันทึกข้อมูลชื่อหนังสือที่ทำการยืมแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.32 แสดงส่วนแสดงผลเพื่อให้เลือกหนังสือที่นำมาคืน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.33 แสดงส่วนแสดงผลเมื่อเลือกหนังสือที่นำมาคืน

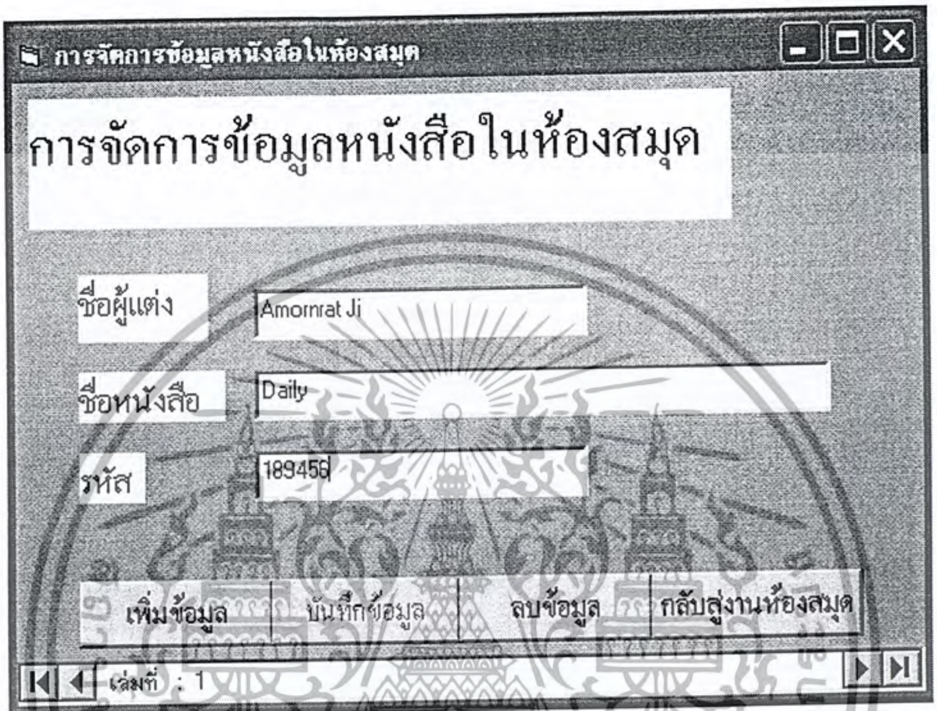
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.34 แสดงส่วนแสดงผลเมื่อเลือก “ค้นหาข้อมูล” ในหน้าต่างการยืม-คืนหนังสือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

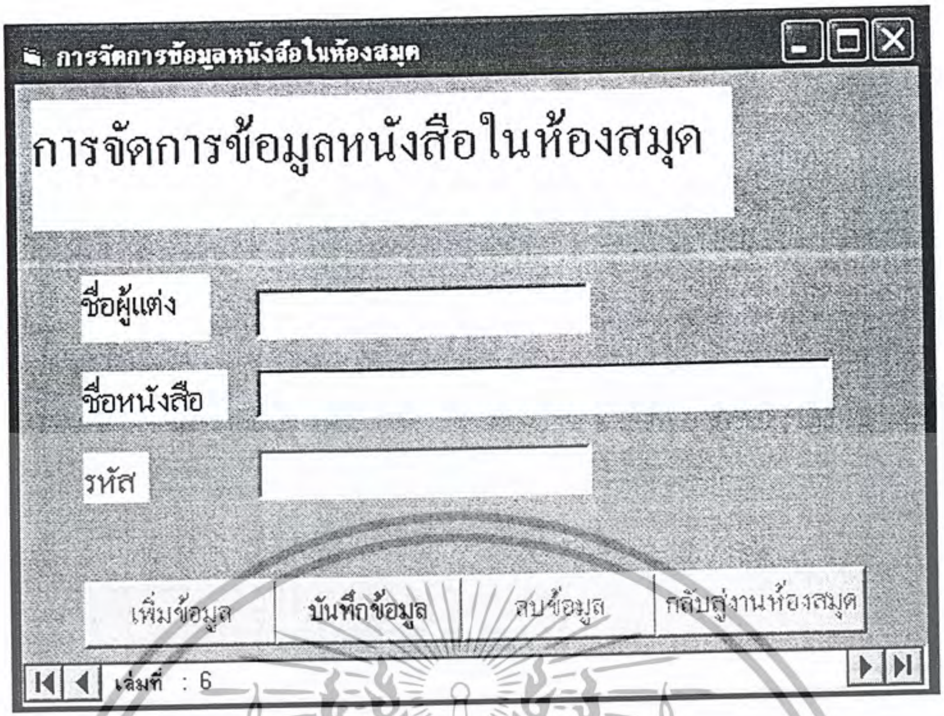
4.2.5.2 เมื่อเข้าสู่การจัดการข้อมูลหนังสือในห้องสมุด

- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "การจัดการข้อมูลหนังสือในห้องสมุด" ซึ่งจะแสดงรายละเอียดของหนังสือเล่มล่าสุดที่ทำการบันทึกข้อมูลไว้ และถ้าหากเลือก "เพิ่มข้อมูล" ก็จะแสดงหน้าต่าง "การจัดการข้อมูลหนังสือในห้องสมุด" ที่มีช่องข้อมูลที่ว่างไว้ เพื่อให้กรอกรายละเอียดของหนังสือที่ต้องการบันทึกข้อมูลเพิ่ม



รูปที่ 4.35 แสดงส่วนแสดงผลเมื่อเข้าสู่การจัดการข้อมูลหนังสือในห้องสมุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

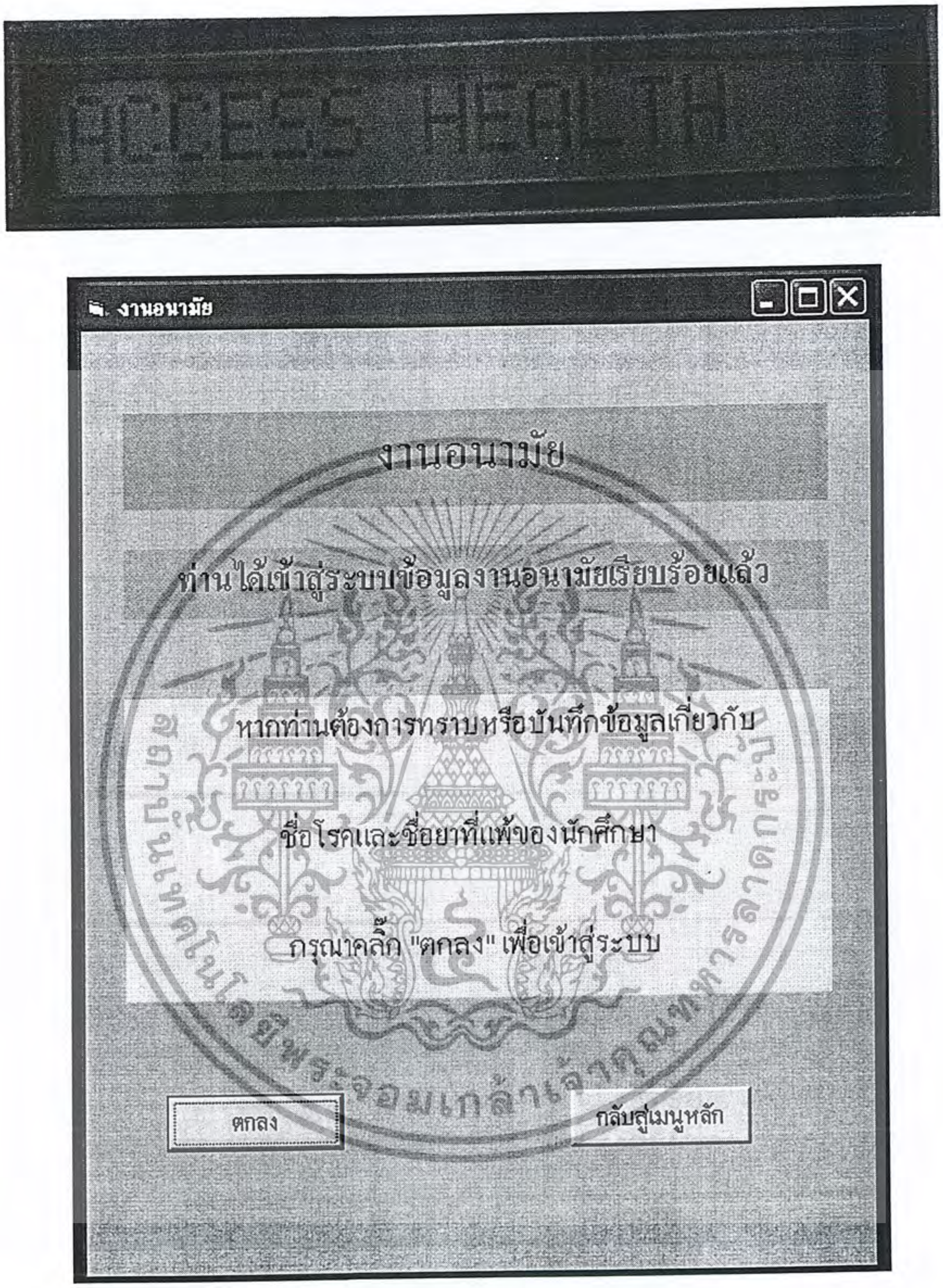


รูปที่ 4.36 แสดงส่วนแสดงผลเมื่อเลือก "เพิ่มข้อมูล" ในหน้าต่างการจัดการข้อมูลหนังสือในห้องสมุด

4.2.6 เมื่อเข้าสู่งานอนามัย

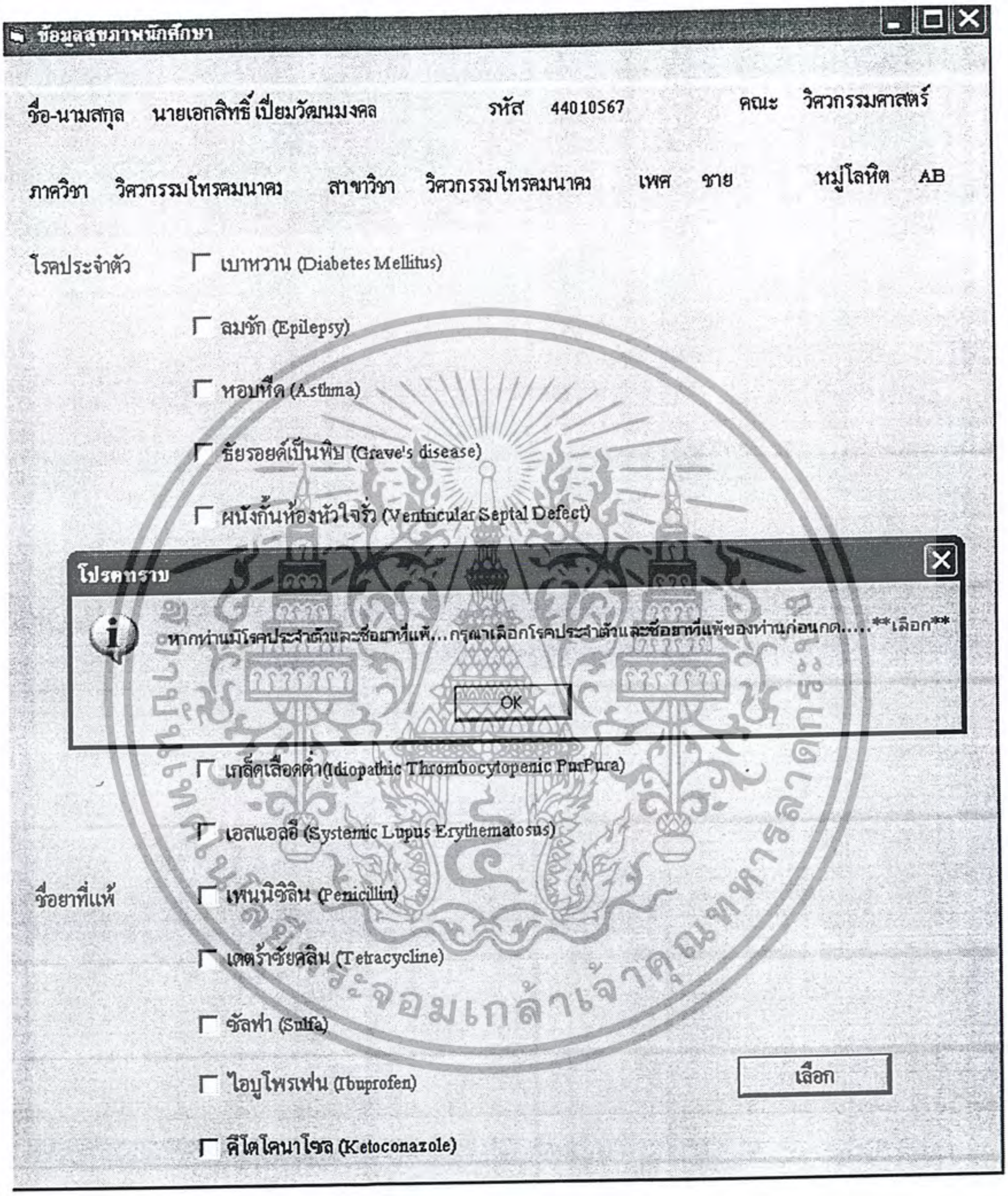
- ที่เครื่องอ่าน-เขียนสมาร์ทการ์ดจะแสดงข้อความ "ACCESS HEALTH"
- ที่หน้าจอคอมพิวเตอร์จะแสดงหน้าต่าง "งานอนามัย" จากนั้นหากเลือก "ตกลง" และทำการตรวจสอบแล้วพบว่าบัตรสมาร์ทการ์ดที่เสียบเข้ามานั้นยังไม่มีการบันทึกข้อมูลสุขภาพไว้ ก็จะแสดงข้อความและหน้าต่างเพื่อให้เลือกชื่อโรคประจำตัวและชื่อยาที่แพ้ แต่ถ้าหากตรวจสอบแล้วพบว่ามีการบันทึกข้อมูลสุขภาพไว้แล้ว ก็จะแสดงหน้าต่างที่แสดงชื่อโรคประจำตัวและชื่อยาที่แพ้ที่ได้บันทึกไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



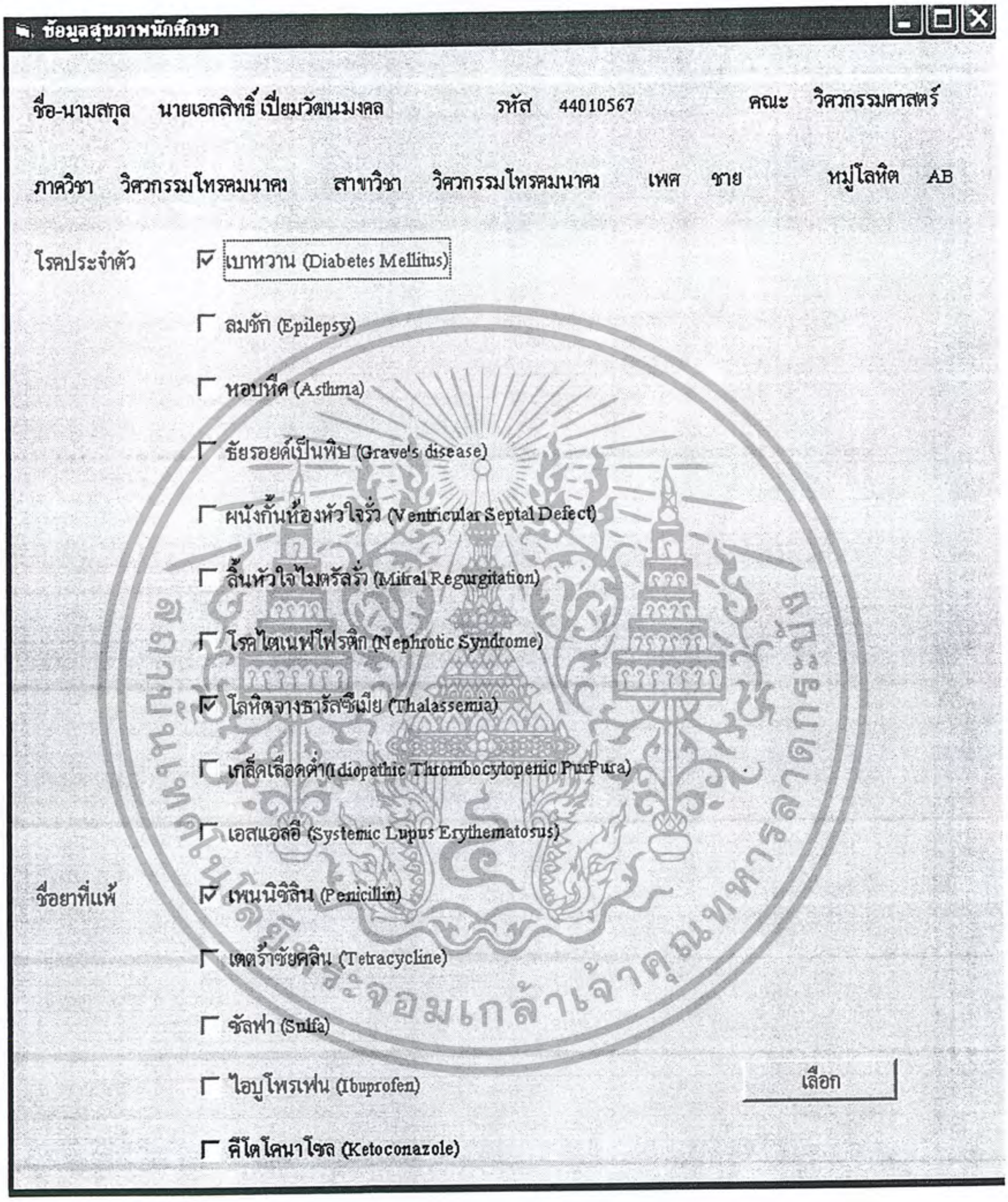
รูปที่ 4.37 แสดงส่วนแสดงผลเมื่อเข้าสู่งานอนามัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



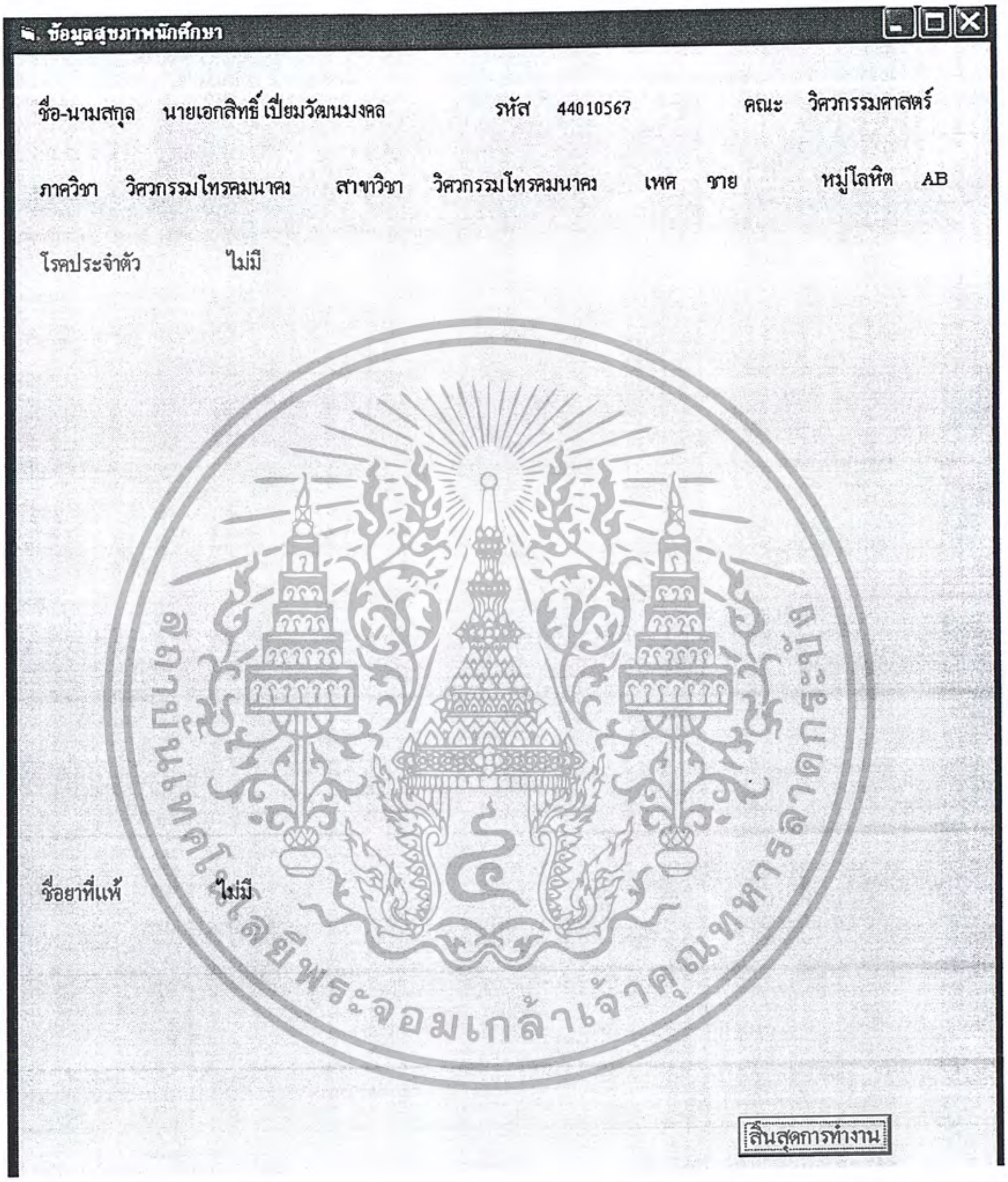
รูปที่ 4.38 แสดงส่วนแสดงผลที่แสดงข้อความเพื่อให้เลือกชื่อโรคและชื่อยาที่แพ้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.39 แสดงส่วนแสดงผลที่แสดงหน้าต่างเพื่อให้เลือกชื่อโรคและชื่อยาที่แพ้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.40 แสดงส่วนแสดงผลที่แสดงชื่อโรคประตัวและชื่อยาที่แพ้ของนักศึกษา
ที่อ่านได้จากบัตรสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

5.1 สรุป

จากการทำโครงงานนี้ทำให้ได้ศึกษาถึงโครงสร้างและการทำงานของสมาร์ทการ์ด ไมโครคอนโทรลเลอร์ และจอแสดงผลแบบผลึกเหลว (LCD) รวมทั้งยังได้ศึกษาถึงการเขียนโปรแกรมควบคุมคอมพิวเตอร์ โปรแกรมควบคุมไมโครคอนโทรลเลอร์ และการสร้างระบบฐานข้อมูล (Database) ซึ่งจากการออกแบบและการสร้างนั้นทำให้ได้เครื่องอ่าน-เขียนสมาร์ทการ์ดที่ควบคุมการทำงานโดยไมโครคอนโทรลเลอร์จำนวน 1 เครื่อง โดยจะทำงานร่วมกับโปรแกรมคอมพิวเตอร์ที่รองรับการใช้งานสมาร์ทการ์ดเพื่อติดต่อกับระบบสารสนเทศนักศึกษา ซึ่งประกอบด้วยระบบงานทะเบียน งานห้องสมุด และงานอนามัย โดยโปรแกรมคอมพิวเตอร์นั้นเขียนขึ้นด้วยภาษา Visual Basic ที่มีการติดต่อกับระบบฐานข้อมูลที่จัดทำโดยโปรแกรม Microsoft Access

5.2 ปัญหาและแนวทางแก้ไข

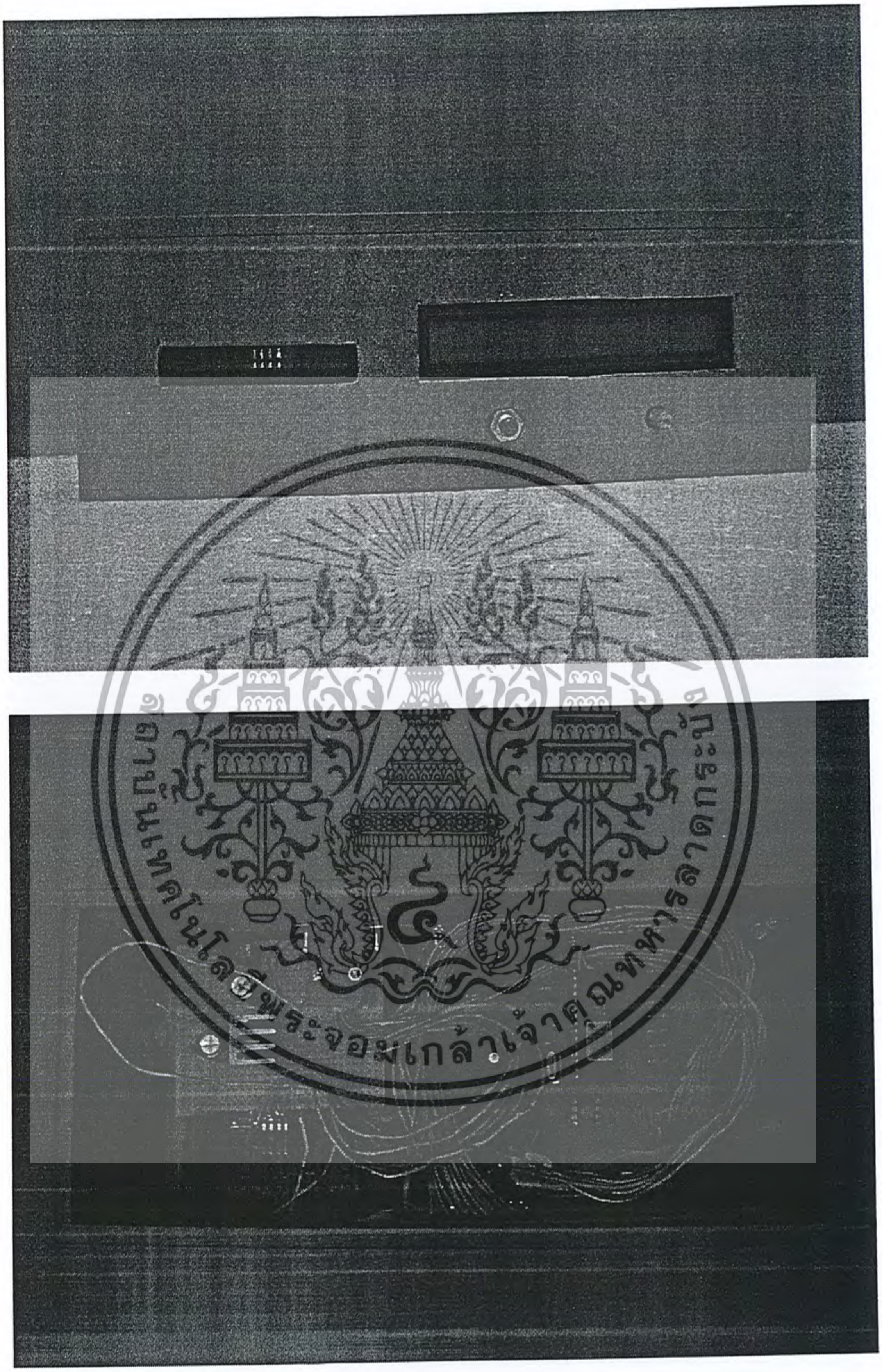
- ปัญหาที่พบในส่วนของฐานข้อมูล (Database) นั้น เกิดขึ้นเนื่องจากการที่ต้องทำการนำข้อมูลจากตารางต่าง ๆ ที่มีความสัมพันธ์กันมาใช้งานรวมกัน ดังนั้นจึงต้องเลือกใช้ฐานข้อมูลที่มีลักษณะเป็น Relational Database เพื่อนำข้อมูลจากหลาย ๆ ตารางที่มีความสัมพันธ์กันมาใช้งานร่วมกันได้ และต้องให้ความสำคัญกับลำดับขั้นตอนการดึงข้อมูลต่าง ๆ ให้สัมพันธ์กัน เพื่อให้ได้หน้าจอแสดงผลข้อมูลที่ถูกต้องครบถ้วน

- ปัญหาที่พบเมื่อนำระบบสารสนเทศนักศึกษา ที่เขียนขึ้นโดยภาษา Visual Basic มาใช้งานร่วมกับเครื่องอ่าน-เขียนสมาร์ทการ์ดที่ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์นั้น เกิดขึ้นเนื่องจากช่วงเวลาที่ใช้ในการทำงานที่ไม่สัมพันธ์กันระหว่างระบบสารสนเทศนักศึกษาและเครื่องอ่าน-เขียนสมาร์ทการ์ดจนทำให้เกิดความผิดพลาดในการรับ-ส่งข้อมูลระหว่างกันขึ้น ทั้งนี้เนื่องจากว่าไมโครคอนโทรลเลอร์จะต้องใช้เวลาช่วงหนึ่งในการติดต่อรับ-ส่งข้อมูลกับสมาร์ทการ์ด ดังนั้นเพื่อแก้ไขปัญหานี้ ระบบสารสนเทศนักศึกษาและไมโครคอนโทรลเลอร์จึงต้องมีการส่งข้อมูลติดต่อระหว่างกันเพื่อเป็นการหน่วงเวลาและแจ้งให้ไมโครคอนโทรลนั้นทำการติดต่อกับสมาร์ทการ์ดให้เรียบร้อยเสียก่อนแล้วจึงวนมารอรับข้อมูลที่ส่งมาจากคอมพิวเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงเครื่องอ่าน-เขียนสมาร์ทการ์ดของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแสดงบัตรสมาร์ทการ์ดที่ใช้ในโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมควบคุมการทำงานของเครื่องอ่าน-เขียนสมาร์ทการ์ด
โดยใช้ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CLEAR        EQU   0
DIGIT        EQU   16
LCD_ADDR     EQU   030H
LCD_DATA     EQU   031H
CRD_IN             BIT   P1.3
CRD_PWR            BIT   P1.4
CRD_IO             BIT   P1.0
CRD_CLK            BIT   P1.2
CRD_RST            BIT   P1.1
LCD_RS             BIT   P1.5
LCD_EN             BIT   P1.6
BLINK        BIT   P3.4

             ORG   21H
FLAG:        DS    1
FLAG_PWR     BIT   FLAG.0
RRFAG        BIT   FLAG.1
RX_FLAG            BIT   FLAG.2
FLAG_ERR     BIT   FLAG.3
FLAG_SE            BIT   FLAG.4
CARD_BUFF:   DS    20
MEM_COMMAND:       DS    8
FREE:        DS    10
DISBUFF:     DS    16
TIME:        DS    2
TMP:         DS    1
ERROR_C:     DS    1
CARD_BUFF_1:       DS    8
CARD_BUFF_2:       DS    13
CARD_BUFF_3:       DS    8
CARD_BUFF_4:       DS    8
N_TEST:            DS    8
ID_TEST:     DS    13
STID_TEST:   DS    8
BLOOD_TEST:  DS    2
PASS_TEST:   DS    6
READ_NAME:   DS    8
READ_STID:   DS    8
READ_ID:     DS    13
READ_BLOOD:  DS    2
READ_PASS:   DS    6
STACK:             DS    1

MAIN:
             ORG   90H
             MOV   SP,#STACK
             MOV   PSW,#18H
             MOV   SCON,#01010000B
             MOV   TMOD,#00100010B
             MOV   TCON,#01010000B
             MOV   IP,#10H
             MOV   TH1,#0FDH
             MOV   TL1,#0FDH
             MOV   TL0,#01H
             MOV   TH0,#01H
             SETB  IE.1
             SETB  TR0
             SETB  TR1
             MOV   TIME+0,#CLEAR
             MOV   TIME+1,#CLEAR
             SETB  ES
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
START:
            MOV   FLAG,#CLEAR
            LCALL INIT_LCD
            LCALL PWR_CARD
TEST:
            MOV   FLAG,#CLEAR
            LCALL CRD_CARDIN
            SETB  FLAG_PWR
            LCALL PWR_CARD
            LCALL READ_C4442
            CLR   FLAG_PWR
            LCALL PWR_CARD
            LCALL CRD_CARDOUT
            AJMP  TEST
;************************************************
;READ ATR (ANSWER TO RESET)
;************************************************
READ_C4442:
            LCALL CRD_CARDIN
            CLR   EA
            LCALL ATR_READ
            MOV   R1,#CARD_BUFF
            MOV   A,@R1
            CJNE  A,#0A2H,WRONG_CARD
            AJMP  PASS
WRONG_CARD:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#WRONGCARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#WRONGCARD_2
            LCALL WRLINE_LCD
            LJMP  EXIT_FUNC
            RET
;************************************************
;SHOW 'ACCESS SYSTEM' ON LCD
;************************************************
PASS:
            LCALL RX_BYTE
            CJNE  A,#'R',$

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#ACCESS_11
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#ACCESS_12
            LCALL WRLINE_LCD
;************************************************
;READ STID AND PASSWORD FROM CARD TO MCS-51
;************************************************
RD_PASS:
            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_1
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV    R0,#READ_STID
            MOV    R1,#CARD_BUFF
            LCALL BACKUP_CARD

            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_6
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_PASS
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD
;*************************************************
;SEND 'R' TO COM FOR SHOWING STATE
;*************************************************
            MOV   A,#'R'
            LCALL SEND_BYTE

            LCALL RX_BYTE
            CJNE  A,#'R',$
;*************************************************
;SEND STID AND PASSWORD TO COMPUTER
;*************************************************
            MOV   R1,#READ_STID
            MOV   R3,#8
MOVE_1:           MOV   A,@R1
            LCALL SEND_BYTE
            INC   R1
            DJNZ  R3,MOVE_1

            MOV   R1,#READ_PASS
            MOV   R3,#6
MOVE_2:           MOV   A,@R1
            LCALL SEND_BYTE
            INC   R1
            DJNZ  R3,MOVE_2
;*************************************************
;CHECK THAT IS THE RIGHT PASSWORD?
;*************************************************
            LCALL RX_BYTE
            CJNE  A,#'C',TEST1
WRONG_PASS:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#WRONG_PASS_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#WRONG_PASS_2
            LCALL WRLINE_LCD
            LCALL DELAY_1s

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JNB   CRD_IN,$

            MOV   A,#'C'
            LCALL SEND_BYTE

            LJMP  EXIT_FUNC

TEST1:            CJNE  A,#'D',TEST2
;*************************************************
;RIGHT PASSWORD IN CASE OF MASTER CARD
;*************************************************
RIGHT:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD
            JNB     CRD_IN,$

            MOV   A,#'D'
            LCALL SEND_BYTE
            LJMP  SELECT_MODE
;*************************************************
;RIGHT PASSWORD IN CASE OF OTHER CARDS
;*************************************************
TEST2:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#STUDENT_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#STUDENT_2
            LCALL WRLINE_LCD

            JB      CRD_IN,$

            LCALL RX_BYTE
            CJNE  A,#'I',SPEC_EXIT_Z1
            LJMP  RE_PASS
;*************************************************
;SELECT MODE FOR MASTER CARD
;*************************************************
SELECT_MODE:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#SELECT_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#SELECT_2
            LCALL WRLINE_LCD

            LCALL RX_BYTE
            CJNE  A,#'T',LIBRARY1
            AJMP  REGISTRA
;*************************************************
;SPECIAL FUNCTION_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************
SPEC_EXIT1:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD
            JNB   CRD_IN,$

            CLR   FLAG_PWR
            LCALL PWR_CARD
            LCALL CRD_CARDOUT
            SETB  EA
            RET

LIBRARY1:
            LJMP  LIBRARY
CHANGE1:
            LJMP  CHANGE
SPEC_EXIT_Z1:
            LJMP  SPEC_EXIT_Z
;*********************************************
;IF SELECT REGISTRA SYSTEM
;*********************************************
REGISTRA:
            LCALL RX_BYTE
            CJNE  A,#'N',CHANGE1
;*********************************************
;CREATE NEW CARD
;*********************************************
WRITE_NEW:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RECORD_DATA_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RECORD_DATA_2
            LCALL WRLINE_LCD

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_11
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_12
            LCALL WRLINE_LCD

            JB    CRD_IN,$

            MOV   A,#'N'
            LCALL SEND_BYTE
;*********************************************
;RECEIVE STUDENT ID FROM COMPUTER TO MCS-51
;*********************************************
            MOV   R3,#8
            MOV   R0,#STID_TEST
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RECEIVE_1:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_1
;**************************************************
;RECEIVE NAME FROM COMPUTER TO MCS-51
;**************************************************
            MOV   R3,#8
            MOV   R0,#N_TEST
RECEIVE_2:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_2
;**************************************************
;RECEIVE ID FROM COMPUTER TO MCS-51
;**************************************************
            MOV   R3,#13
            MOV   R0,#ID_TEST
RECEIVE_3:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_3
;**************************************************
;RECEIVE BLOOD TYPE FROM COMPUTER TO MCS-51
;**************************************************
            MOV   R3,#2
            MOV   R0,#BLOOD_TEST
RECEIVE_4:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_4

            LCALL READPSC
            LCALL VERIFYPIN
            JB    FLAG_ERR,NOT_SUC1
            JNB   FLAG_SE,NOT_SUC1
            AJMP  WRITE_DATA
NOT_SUC1:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_2
            LCALL WRLINE_LCD
            LCALL LCD_DELAY
            JNB   CRD_IN,$
            LJMP  EXIT_FUNC
;**************************************************
;WRITE DATA BEGINNING AT ADDRESS 32D
;**************************************************
WRITE_DATA: MOV   DPTR,#UPDATE_MAIN_1
            LCALL LOAD_COMMAND
            MOV   R1,#STID_TEST
            MOV   R0,#MEM_COMMAND+2
            MOV   A,@R1
            MOV   @R0,A
            LCALL CONTROL
            LCALL PROCESS_254
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    DPTR,#UPDATE_MAIN_2
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+1
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_3
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+2
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_4
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+3
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_5
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+4
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_6
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+5
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_7
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+6
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_8
LCALL LOAD_COMMAND
MOV    R1,#STID_TEST+7
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
```

จุฬาลงกรณ์มหาวิทยาลัย
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_9
LCALL LOAD_COMMAND
MOV   R1,#N_TEST
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_10
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+1
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_11
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+2
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_12
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+3
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_13
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+4
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_14
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+5
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_15
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+6
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_16
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+7
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_17
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_18
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+1
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_19
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+2
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_20
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+3
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_21
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+4
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_22
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+5
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_23
LCALL LOAD_COMMAND
MOV    R1,#ID_TEST+6
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_24
LCALL LOAD_COMMAND
MOV    R1,#ID_TEST+7
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_25
LCALL LOAD_COMMAND
MOV    R1,#ID_TEST+8
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_26
LCALL LOAD_COMMAND
MOV    R1,#ID_TEST+9
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_27
LCALL LOAD_COMMAND
MOV    R1,#ID_TEST+10
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_28
LCALL LOAD_COMMAND
MOV    R1,#ID_TEST+11
MOV    R0,#MEM_COMMAND+2
MOV    A,@R1
MOV    @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV    DPTR,#UPDATE_MAIN_29
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL LOAD_COMMAND
        MOV   R1,#ID_TEST+12
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   DPTR,#UPDATE_MAIN_30
        LCALL LOAD_COMMAND
        MOV   R1,#BLOOD_TEST
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   DPTR,#UPDATE_MAIN_31
        LCALL LOAD_COMMAND
        MOV   R1,#BLOOD_TEST+1
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   A,#'N'
        LCALL SEND_BYTE

        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#WR_SUCCESS_1
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#WR_SUCCESS_2
        LCALL WRLINE_LCD
        LCALL DELAY_1s
        LCALL LCD_DELAY
        MOV   R2,#4
        LCALL DTSEC
;***********************************************************
;READ STID & NAME & ID & BLOOD TYPE FROM CARD TO MCS-51
;***********************************************************
RD_DATA:
        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#RECORDED_DATA_1
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#RECORDED_DATA_2
        LCALL WRLINE_LCD

        LCALL CLEAR_CARDBUFF
        LCALL READ_MAIN_1
        SETB  CRD_PWR
        LCALL DELAY_100ms
        LCALL DELAY_100ms
        CLR   CRD_PWR
        MOV   R0,#READ_STID
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_2
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_NAME
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            LCALL CLEAR_CARDBUFF2
            LCALL READ_MAIN_3
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_ID
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD2

            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_4
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_BLOOD
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD3

            AJMP  READ
;*****************************************************************
;SHOW RECORDED DATA ON LCD
;*****************************************************************
;*****************************************************************
;SHOW STUDENT ID ON LCD
;*****************************************************************
READ:       LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_1
            SETB  CRD_PWR
            LCALL DELAY_100mS
            LCALL DELAY_100mS
            CLR   CRD_PWR
            MOV   R0,#CARD_BUFF_1
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#STID
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#SPACE
            LCALL WRLINE_LCD

            MOV   R1,#CARD_BUFF_1
            LCALL CHECK_BYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          LCALL CLEAR_DISBUFF
          MOV   B,#8
          MOV   R0,#DISBUFF
          MOV   R1,#CARD_BUFF_1
          LCALL LOAD_TOLCD
          LCALL DELAY_1s
          MOV   R2,#0CH
          LCALL DTSEC
;*********************************************************
;SHOW NAME ON LCD
;*********************************************************
          LCALL CLEAR_CARDBUFF
          LCALL CLEAR_CARDBUFF1
          LCALL READ_MAIN_2
          SETB  CRD_PWR
          LCALL DELAY_100mS
          LCALL DELAY_100mS
          CLR   CRD_PWR
          MOV   R0,#CARD_BUFF_1
          MOV   R1,#CARD_BUFF
          LCALL BACKUP_CARD

          MOV   LCD_ADDR,#000H
          LCALL SET_ADDR_LCD
          MOV   DPTR,#NAME
          LCALL WRLINE_LCD
          MOV   LCD_ADDR,#040H
          LCALL SET_ADDR_LCD
          MOV   DPTR,#SPACE
          LCALL WRLINE_LCD

          MOV   R1,#CARD_BUFF_1
          LCALL CHECK_BYTE

          LCALL CLEAR_DISBUFF
          MOV   B,#8
          MOV   R0,#DISBUFF
          MOV   R1,#CARD_BUFF_1
          LCALL LOAD_TOLCD
          LCALL DELAY_1s
          MOV   R2,#0CH
          LCALL DTSEC
;*********************************************************
;SHOW ID ON LCD
;*********************************************************
          LCALL CLEAR_CARDBUFF
          LCALL CLEAR_CARDBUFF1
          LCALL READ_MAIN_31
          SETB  CRD_PWR
          LCALL DELAY_100mS
          LCALL DELAY_100mS
          CLR   CRD_PWR
          MOV   R0,#CARD_BUFF_1
          MOV   R1,#CARD_BUFF
          LCALL BACKUP_CARD2

          MOV   LCD_ADDR,#000H
          LCALL SET_ADDR_LCD
          MOV   DPTR,#ID
          LCALL WRLINE_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#SPACE
        LCALL WRLINE_LCD

        MOV   R1,#CARD_BUFF_1
        LCALL CHECK_BYTE

        LCALL CLEAR_DISBUFF
        MOV   B,#8
        MOV   R0,#DISBUFF
        MOV   R1,#CARD_BUFF_1
        LCALL LOAD_TOLCD
        LCALL DELAY_1s

        LCALL CLEAR_CARDBUFF
        LCALL CLEAR_CARDBUFF1
        LCALL READ_MAIN_32
        SETB  CRD_PWR
        LCALL DELAY_100mS
        LCALL DELAY_100mS
        CLR   CRD_PWR
        MOV   R0,#CARD_BUFF_1
        MOV   R1,#CARD_BUFF
        LCALL BACKUP_CARD

        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#ID
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#SPACE
        LCALL WRLINE_LCD

        MOV   R1,#CARD_BUFF_1
        LCALL CHECK_BYTE

        LCALL CLEAR_DISBUFF
        MOV   B,#5
        MOV   R0,#DISBUFF
        MOV   R1,#CARD_BUFF_1
        LCALL LOAD_TOLCD
        LCALL DELAY_1s
        MOV   R2,#0CH
        LCALL DTSEC
;*****************************************************
;SHOW BLOOD TYPE ON LCD
;*****************************************************
        LCALL CLEAR_CARDBUFF
        LCALL CLEAR_CARDBUFF1
        LCALL READ_MAIN_4
        SETB  CRD_PWR
        LCALL DELAY_100mS
        LCALL DELAY_100mS
        CLR   CRD_PWR
        MOV   R0,#CARD_BUFF_1
        MOV   R1,#CARD_BUFF
        LCALL BACKUP_CARD

        MOV   LCD_ADDR,#000H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SET_ADDR_LCD
            MOV   DPTR,#BLOOD
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#SPACE
            LCALL WRLINE_LCD

            MOV   R1,#CARD_BUFF_1
            LCALL CHECK_BYTE

            LCALL CLEAR_DISBUFF
            MOV   B,#2
            MOV   R0,#DISBUFF
            MOV   R1,#CARD_BUFF_1
            LCALL LOAD_TOLCD
            LCALL DELAY_1s

            MOV   R2,#0CH
            LCALL DTSEC

            MOV   A,#'N'
            LCALL SEND_BYTE

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD
            JNB   CRD_IN,$

            MOV   A,#'N'
            LCALL SEND_BYTE
;*******************************************************
;ASK DO YOU WANT TO WRITE OTHER CARD
;*******************************************************
WANT:
            LCALL RX_BYTE
            CJNE  A,#'N',DONT_WANT
            LJMP  WRITE_NEW
DONT_WANT:
            LJMP  SELECT_MODE
NOT_SUCCESS:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_2
            LCALL WRLINE_LCD
            LCALL LCD_DELAY

EXIT_FUNC:
            CLR   FLAG_PWR
            LCALL PWR_CARD
            LCALL CRD_CARDOUT
            SETB  EA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
;************************************************
;CHANGE PASSWORD
;************************************************
CHANGE:
            CJNE  A,#'I',SELECT_MODE1
            AJMP  CHANGE2
;************************************************
;SPECIAL PROGRAM FOR JUMPING TO SELECT MODE
;************************************************
SELECT_MODE1:
            LJMP  SELECT_MODE
;************************************************
CHANGE2:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_11
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_12
            LCALL WRLINE_LCD

            JB    CRD_IN,$

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_21
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_22
            LCALL WRLINE_LCD

            LCALL RX_BYTE
            CJNE  A,#'I',$
CHANGE3:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#ACCESS_11
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#ACCESS_12
            LCALL WRLINE_LCD
;************************************************
;READ STID AND PASSWORD FROM CARD TO MCS-51
;************************************************
            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_1
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_STID
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_6
            SETB  CRD_PWR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_PASS
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            MOV   A,#'I'
            LCALL SEND_BYTE

            LCALL RX_BYTE
            CJNE  A,#'I',$
;***********************************************************
;SEND STID AND PASSWORD FROM MCS-51 TO COMPUTER
;***********************************************************
            MOV   R1,#READ_STID
            MOV   R3,#8
MOVE_6:           MOV   A,@R1
            LCALL SEND_BYTE
            INC   R1
            DJNZ  R3,MOVE_6

            MOV   R1,#READ_PASS
            MOV   R3,#6
MOVE_7:           MOV   A,@R1
            LCALL SEND_BYTE
            INC   R1
            DJNZ  R3,MOVE_7

            LCALL RX_BYTE
            CJNE  A,#'I',SELECT_MODE2
            LJMP  CHANGE4
;***********************************************************
;SPECIAL PROGRAM FOR JUMPING TO SELECT MODE
;***********************************************************
SELECT_MODE2:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD

            JNB   CRD_IN,$

            MOV   A,#'E'
            LCALL SEND_BYTE

            LJMP  SELECT_MODE
;***********************************************************
CHANGE4:
            LCALL RX_BYTE
            CJNE  A,#'I',CHANGE5
            LJMP  RE_PASS

CHANGE5:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD

            JNB   CRD_IN,$

            MOV   A,#'E'
            LCALL SEND_BYTE

            LJMP  SELECT_MODE
;****************************************************
;RECEIVE NEW PASSWORD FROM COM TO MCS-51
;****************************************************
RE_PASS:
            MOV   A,#'I'
            LCALL SEND_BYTE

            MOV   R3,#6
            MOV   R0,#PASS_TEST
RECEIVE_5:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_5

            LCALL READPSC
            LCALL VERIFYPIN
            JB    FLAG_ERR,NOT_SUC2
            JNB   FLAG_SE,NOT_SUC2
            AJMP  WRITE_PASS

NOT_SUC2:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_2
            LCALL WRLINE_LCD
            LCALL LCD_DELAY
            JNB   CRD_IN,$
            LJMP  EXIT_FUNC
;****************************************************
;RECORD NEW PASSWORD ON SMART CARD
;****************************************************
WRITE_PASS:
            MOV   DPTR,#UPDATE_MAIN_50
            LCALL LOAD_COMMAND
            MOV   R1,#PASS_TEST
            MOV   R0,#MEM_COMMAND+2
            MOV   A,@R1
            MOV   @R0,A
            LCALL CONTROL
            LCALL PROCESS_254

            MOV   DPTR,#UPDATE_MAIN_51
            LCALL LOAD_COMMAND
            MOV   R1,#PASS_TEST+1
            MOV   R0,#MEM_COMMAND+2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   DPTR,#UPDATE_MAIN_52
        LCALL LOAD_COMMAND
        MOV   R1,#PASS_TEST+2
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   DPTR,#UPDATE_MAIN_53
        LCALL LOAD_COMMAND
        MOV   R1,#PASS_TEST+3
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   DPTR,#UPDATE_MAIN_54
        LCALL LOAD_COMMAND
        MOV   R1,#PASS_TEST+4
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   DPTR,#UPDATE_MAIN_55
        LCALL LOAD_COMMAND
        MOV   R1,#PASS_TEST+5
        MOV   R0,#MEM_COMMAND+2
        MOV   A,@R1
        MOV   @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#PASS_REC_1
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#PASS_REC_2
        LCALL WRLINE_LCD
        LCALL LCD_DELAY

        MOV   A,#'I'
        LCALL SEND_BYTE

        LCALL RX_BYTE
        CJNE  A,#'I',SPEC_EXIT2
        LJMP  RE_PASS

SPEC_EXIT2:
        CJNE  A,#'E',SPEC_EXIT_Z
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_1
        LCALL WRLINE_LCD
        MOV    LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_2
        LCALL WRLINE_LCD
        LCALL LCD_DELAY

        JNB    CRD_IN,$

        MOV    A,#'E'
        LCALL SEND_BYTE
        JNB    CRD_IN,$
        LJMP   SELECT_MODE
;*******************************************************************
;SPECIAL PROGRAM FOR STUDENT CARD FOR JUMPING TO THE START POINT
;*******************************************************************
SPEC_EXIT_Z:
        CJNE   A,#'Y',$

        MOV    LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_1
        LCALL WRLINE_LCD
        MOV    LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_2
        LCALL WRLINE_LCD
        LCALL LCD_DELAY

        JNB    CRD_IN,$

        MOV    A,#'Y'
        LCALL SEND_BYTE
        JNB    CRD_IN,$
        LJMP   EXIT_FUNC
;*******************************************************
;IF SELECT LIBRARY SYSTEM
;*******************************************************
LIBRARY:
        CJNE   A,#'L',HEALTH1
        AJMP   LIBRARY2
LIBRARY2:
        LCALL RX_BYTE
        CJNE   A,#'L',SELECT_MODE3
        LJMP   LIBRARY3
;**********************************************************************
;SPECIAL PROGRAM FOR JUMPING TO SELECT MODE AND JUMPING TO HEALTH
;**********************************************************************
SELECT_MODE3:
        MOV    LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_1
        LCALL WRLINE_LCD
        MOV    LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_2
        LCALL WRLINE_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JNB   CRD_IN,$

            MOV   A,#'E'
            LCALL SEND_BYTE

            LJMP  SELECT_MODE

HEALTH1:
            LJMP  HEALTH
;*******************************************************
LIBRARY3:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_11
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_12
            LCALL WRLINE_LCD

            JB    CRD_IN,$

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_21
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_22
            LCALL WRLINE_LCD

            LCALL RX_BYTE
            CJNE  A,#'L',$
;*******************************************************
;READ STUDENT ID FROM CARD TO MCS-51
;*******************************************************
            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_1
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_STID
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            MOV   A,#'L'
            LCALL SEND_BYTE

            LCALL RX_BYTE
            CJNE  A,#'L',$
;*******************************************************
;SEND STUDENT ID FROM MCS-51 TO COMPUTER
;*******************************************************
            MOV   R1,#READ_STID
            MOV   R3,#8
MOVE_3:           MOV   A,@R1
            LCALL SEND_BYTE
            INC   R1
            DJNZ  R3,MOVE_3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#LIB_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#LIB_2
            LCALL WRLINE_LCD
            LCALL LCD_DELAY

            LCALL RX_BYTE
            CJNE  A,#'L',SELECT_MODE4

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD

            JNB   CRD_IN,$

            MOV   A,#'L'
            LCALL SEND_BYTE

            LJMP  LIBRARY2
;*******************************************************
;SPECIAL PROGRAM FOR JUMPING TO SELECT MODE
;*******************************************************
SELECT_MODE4:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD

            JNB   CRD_IN,$

            MOV   A,#'E'
            LCALL SEND_BYTE

            LJMP  SELECT_MODE
;*******************************************************
;IF SELECT HEALTH SYSTEM
;*******************************************************
HEALTH:
            CJNE  A,#'H',SPEC_EXIT3
            AJMP  HEALTH2
;*******************************************************
;SPECIAL PROGRAM FOR JUMPING AT THE START POINT
;*******************************************************
SPEC_EXIT3:
            JNB   CRD_IN,$
            LJMP  EXIT_FUNC
;*******************************************************
HEALTH2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL RX_BYTE
            CJNE  A,#'H',SELECT_MODE5
            LJMP  HEALTH3
;*******************************************************
;SPECIAL PROGRAM FOR JUMPING TO SELECT MODE
;*******************************************************
SELECT_MODE5:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#RE_CARD_2
            LCALL WRLINE_LCD

            JNB   CRD_IN,$

            MOV   A,#'E'
            LCALL SEND_BYTE

            LJMP  SELECT_MODE
;*******************************************************
HEALTH3:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_11
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_12
            LCALL WRLINE_LCD

            JB    CRD_IN,$

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_21
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#INSERT_22
            LCALL WRLINE_LCD

            LCALL RX_BYTE
            CJNE  A,#'H',$
;*******************************************************
;READ STID AND MED FROM CARD TO MCS-51
;*******************************************************
            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_1
            SETB  CRD_PWR
            LCALL DELAY_100ms
            LCALL DELAY_100ms
            CLR   CRD_PWR
            MOV   R0,#READ_STID
            MOV   R1,#CARD_BUFF
            LCALL BACKUP_CARD

            LCALL CLEAR_CARDBUFF
            LCALL READ_MAIN_5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SETB  CRD_PWR
        LCALL DELAY_100ms
        LCALL DELAY_100ms
        CLR   CRD_PWR
        MOV   R0,#READ_NAME
        MOV   R1,#CARD_BUFF
        LCALL BACKUP_CARD

        LCALL CLEAR_CARDBUFF
        LCALL READ_MAIN_51
        SETB  CRD_PWR
        LCALL DELAY_100ms
        LCALL DELAY_100ms
        CLR   CRD_PWR
        MOV   R0,#READ_ID
        MOV   R1,#CARD_BUFF
        LCALL BACKUP_CARD

        LCALL CLEAR_CARDBUFF
        LCALL READ_MAIN_52
        SETB  CRD_PWR
        LCALL DELAY_100ms
        LCALL DELAY_100ms
        CLR   CRD_PWR
        MOV   R0,#READ_BLOOD
        MOV   R1,#CARD_BUFF
        LCALL BACKUP_CARD

        MOV   A,#'H'
        LCALL SEND_BYTE

        LCALL RX_BYTE
        CJNE  A,#'H',$
;*******************************************************
;SEND STID AND MED TO COMPUTER
;*******************************************************
        MOV   R1,#READ_STID
        MOV   R3,#8
MOVE_4:       MOV   A,@R1
        LCALL SEND_BYTE
        INC   R1
        DJNZ  R3,MOVE_4

        MOV   R1,#READ_NAME
        MOV   R3,#8
MOVE_5:       MOV   A,@R1
        LCALL SEND_BYTE
        INC   R1
        DJNZ  R3,MOVE_5

        MOV   R1,#READ_ID
        MOV   R3,#8
MOVE_8:       MOV   A,@R1
        LCALL SEND_BYTE
        INC   R1
        DJNZ  R3,MOVE_8

        MOV   R1,#READ_BLOOD
        MOV   R3,#2
MOVE_9:       MOV   A,@R1
        LCALL SEND_BYTE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            INC    R1
            DJNZ   R3,MOVE_9

            MOV    LCD_ADDR,#000H
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#HEALTH_1
            LCALL  WRLINE_LCD
            MOV    LCD_ADDR,#040H
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#HEALTH_2
            LCALL  WRLINE_LCD
;*******************************************************
;CHECK THAT YOU'VE BEEN IN HEALTH SYSYTEM?
;*******************************************************
            LCALL  RX_BYTE
            CJNE   A,#'H',NEW_HEALTH1
            AJMP   HEALTH4
;*******************************************************
NEW_HEALTH1:
            LJMP   NEW_HEALTH
;*******************************************************
HEALTH4:
            LCALL  RX_BYTE
            CJNE   A,#'E',$

            MOV    LCD_ADDR,#000H
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#RE_CARD_1
            LCALL  WRLINE_LCD
            MOV    LCD_ADDR,#040H
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#RE_CARD_2
            LCALL  WRLINE_LCD

            JNB    CRD_IN,$

            MOV    A,#'E'
            LCALL  SEND_BYTE

            LJMP   HEALTH2
;*******************************************************
;RECEIVE MED FROM COMPUTER TO MCS-51
;*******************************************************
NEW_HEALTH:
            LCALL  RX_BYTE
            CJNE   A,#'K',$

            MOV    LCD_ADDR,#000H
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#REC_HEALTH_1
            LCALL  WRLINE_LCD
            MOV    LCD_ADDR,#040H
            LCALL  SET_ADDR_LCD
            MOV    DPTR,#REC_HEALTH_2
            LCALL  WRLINE_LCD

            LCALL  CLEAR_N_TEST
            LCALL  CLEAR_ID_TEST
            LCALL  CLEAR_B_TEST

            MOV    A,#'K'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL SEND_BYTE

            MOV   R3,#8
            MOV   R0,#N_TEST
RECEIVE_6:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_6

            MOV   R3,#8
            MOV   R0,#ID_TEST
RECEIVE_7:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_7

            MOV   R3,#2
            MOV   R0,#BLOOD_TEST
RECEIVE_8:  LCALL RX_BYTE
            MOV   @R0,A
            INC   R0
            DJNZ  R3,RECEIVE_8

            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#DOT_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#DOT_2
            LCALL WRLINE_LCD

            LCALL READPSC
            LCALL VERIFYPIN
            JB    FLAG_ERR,NOT_SUC3
            JNB   FLAG_SE,NOT_SUC3
            AJMP  WRITE_MED
NOT_SUC3:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#NOT_SUCCESS_2
            LCALL WRLINE_LCD
            LCALL LCD_DELAY
            JNB   CRD_IN,$
            LJMP  EXIT_FUNC
;*****************************************************
;RECORD MED FROM MCS-51 TO CARD
;*****************************************************
WRITE_MED:
            MOV   DPTR,#UPDATE_MAIN_32
            LCALL LOAD_COMMAND
            MOV   R1,#N_TEST
            MOV   R0,#MEM_COMMAND+2
            MOV   A,@R1
            MOV   @R0,A
            LCALL CONTROL
            LCALL PROCESS_254
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV   DPTR,#UPDATE_MAIN_33
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+1
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_34
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+2
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_35
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+3
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_36
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+4
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_37
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+5
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_38
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+6
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_39
LCALL LOAD_COMMAND
MOV   R1,#N_TEST+7
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_40
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_41
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+1
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_42
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+2
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_43
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+3
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_44
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+4
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_45
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+5
MOV   R0,#MEM_COMMAND+2
MOV   A,@R1
MOV   @R0,A
LCALL CONTROL
LCALL PROCESS_254

MOV   DPTR,#UPDATE_MAIN_46
LCALL LOAD_COMMAND
MOV   R1,#ID_TEST+6
MOV   R0,#MEM_COMMAND+2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV    A,@R1
        MOV    @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV    DPTR,#UPDATE_MAIN_47
        LCALL LOAD_COMMAND
        MOV    R1,#ID_TEST+7
        MOV    R0,#MEM_COMMAND+2
        MOV    A,@R1
        MOV    @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV    DPTR,#UPDATE_MAIN_48
        LCALL LOAD_COMMAND
        MOV    R1,#BLOOD_TEST
        MOV    R0,#MEM_COMMAND+2
        MOV    A,@R1
        MOV    @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV    DPTR,#UPDATE_MAIN_49
        LCALL LOAD_COMMAND
        MOV    R1,#BLOOD_TEST+1
        MOV    R0,#MEM_COMMAND+2
        MOV    A,@R1
        MOV    @R0,A
        LCALL CONTROL
        LCALL PROCESS_254

        MOV    LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#MED_1
        LCALL WRLINE_LCD
        MOV    LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#MED_2
        LCALL WRLINE_LCD
        LCALL LCD_DELAY
        LCALL DELAY_1s

        MOV    LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_1
        LCALL WRLINE_LCD
        MOV    LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV    DPTR,#RE_CARD_2
        LCALL WRLINE_LCD

        JNB    CRD_IN,$

        MOV    A,#'E'
        LCALL SEND_BYTE

        AJMP   HEALTH2
;**************************************************
;READ MAIN MEMORY
;**************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
READ_MAIN_1:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_1
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#8
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_2:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_2
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#8
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_3:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_3
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#13
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_31:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_31
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#8
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_32:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_32
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#5
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_4:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_4
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL START_OUTGOING
            MOV   R6,#2
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_5:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_5
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#8
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_51:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_51
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#8
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_52:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_52
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#2
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
READ_MAIN_6:
            LCALL START_IFD
            LCALL PHIGH
            MOV   DPTR,#READ_CMD_6
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#6
            MOV   R0,#CARD_BUFF
            LCALL GET_DATA
            RET
;*****************************************************
;READ PSC
;*****************************************************
READPSC:
            CLR   CRD_RST
            LCALL PHIGH
            LCALL START_IFD
            MOV   DPTR,#READ_PSC
            LCALL LOAD_COMMAND
            LCALL SENDCOMMAND
            LCALL START_OUTGOING
            MOV   R6,#4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   R0,#CARD_BUFF
            LCALL READOUTGOING
            MOV   R0,#CARD_BUFF
            MOV   A,@R0
            MOV   ERROR_C,A
            RET
;**********************************************************
;VERIFICATION
;**********************************************************
VERIFYPIN:
            MOV   A,ERROR_C
            ANL   A,#00000111B
            CJNE  A,#0,NONE_LOCK
LOCK:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#CARDBLOCK_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#CARDBLOCK_2
            LCALL WRLINE_LCD
            MOV   R2,#3
            LCALL LCD_DELAY
            LJMP  ERROR_EXIT
NONE_LOCK:
            MOV   DPTR,#UPDATE_SC
            LCALL LOAD_COMMAND
            MOV   A,ERROR_C
            CLR   C
            RLC   A
            ANL   A,#00000111B
            MOV   R0,#MEM_COMMAND+2
            MOV   @R0,A
            LCALL CONTROL
            LCALL PROCESS_123
            LCALL COMPARE
ENDX:
            LCALL READPSC
            MOV   A,ERROR_C
            ANL   A,#00000111B
            CJNE  A,#00000111B,WRONG
            SETB  FLAG_SE
            RET
WRONG:
            MOV   LCD_ADDR,#000H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#WRONGPIN_1
            LCALL WRLINE_LCD
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   DPTR,#WRONGPIN_2
            LCALL WRLINE_LCD
            MOV   R2,#3
            LCALL LCD_DELAY
ERROR_EXIT:
            SETB  FLAG_ERR
            RET
COMPARE:
            MOV   DPTR,#PIN1_SC
            MOV   R4,#3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VERIFY1:
            LCALL LOAD_COMMAND
            LCALL CONTROL
            LCALL PROCESS_2
            DJNZ  R4,VERIFY1
            MOV   DPTR,#UPDATE_FF
            LCALL LOAD_COMMAND
            LCALL CONTROL
            LCALL PROCESS_123
            RET
;*****************************************************
;TOOL
;*****************************************************
LOAD_COMMAND:
            MOV   R0,#MEM_COMMAND
            MOV   R6,#3
LOADCMD:
            CLR   A
            MOVC  A,@A+DPTR
            MOV   @R0,A
            INC   R0
            INC   DPTR
            DJNZ  R6,LOADCMD
            RET
CONTROL:
            CLR   CRD_RST
            LCALL PHIGH
            LCALL START_IFD
            LCALL SENDCOMMAND
            RET
;*****************************************************
;SEND COMMAND, ADDRESS, DATA  TO CARD 3 BYTE
;*****************************************************
SENDCOMMAND:
            MOV   R0,#MEM_COMMAND
            MOV   R6,#3
SELECT:
            MOV   A,@R0
            LCALL CRD_WRITBIT
            INC   R0
            DJNZ  R6,SELECT
            CLR   CRD_CLK
            CLR   CRD_IO
            LCALL PHIGH
            RET
;*****************************************************
;READ ANSWER TO RESET 4 BYTE
;*****************************************************
ATR_READ:
            SETB  FLAG_PWR
            LCALL PWR_CARD
            MOV   R0,#CARD_BUFF
            MOV   R6,#4
            LCALL READ_ATRBYTE
            LCALL CLEAR_DISBUFF
            MOV   DPTR,#SC_ATR
            MOV   R0,#DISBUFF
            MOV   B,#6
            LCALL COPY_FILE
            MOV   B,#4
            MOV   R0,#DISBUFF+5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   R1,#CARD_BUFF
            RET
LOAD_TOATR:
            MOV   A,@R1
            LCALL HTOASC
            MOV   A,R2
            MOV   @R0,A
            INC   R0
            MOV   A,R3
            MOV   @R0,A
            INC   R0
            INC   R1
            DJNZ  B,LOAD_TOATR
            LCALL WRITE_FILE
            RET
;*******************************************************
;READ ATR AND DATA IN MEMORY SLE 4442
;*******************************************************
READ_ATRBYTE:
            LCALL CRD_RESET
READOUTGOING:
GET_DATA:
            LCALL CRD_READBIT
            MOV   @R0,A
            INC   R0
            DJNZ  R6,GET_DATA
            CLR   CRD_CLK
            SETB  CRD_IO
            LCALL PHIGH
            RET
;*******************************************************
;START FROM IFD
;*******************************************************
START_IFD:
            LCALL CHIGH
            CLR   CRD_IO
            LCALL PHIGH
            LCALL CLOW
            RET
;*******************************************************
;START OUTGOING
;*******************************************************
START_OUTGOING:
            LCALL CHIGH
            SETB  CRD_IO
            LCALL PHIGH
            CLR   CRD_CLK
            LCALL PHIGH
            RET
;*******************************************************
;CYCLE CLOCK IS 2 CYCLE
;*******************************************************
PROCESS_2:
            LCALL PROCESSING
            MOV   R5,#2
            LCALL CLOCK_EN
            RET
;*******************************************************
;CYCLE CLOCK FOR WRITE ONLY
;*******************************************************
PROCESS_123:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL PROCESSING
            MOV   R5,#123
            LCALL CLOCK_EN
            RET
PROCESS_254:
            LCALL PROCESSING
            MOV   R5,#254
            LCALL CLOCK_EN
            RET
CLOCK_EN:
            LCALL CHIGH
            LCALL CLOW
            DJNZ  R5,CLOCK_EN
            SETB  CRD_IO
            NOP
            RET
PROCESSING:
            LCALL CHIGH
            SETB  CRD_IO
            NOP
            CLR   CRD_CLK
            NOP
            CLR   CRD_IO
            LCALL PHIGH
            RET
;*****************************************************
;READ DATA 1 BYTE
;*****************************************************
CRD_READBIT:
            MOV   R7,#08
            MOV   A,#0
BITCOUNT:
            LCALL CHIGH
            MOV   C,CRD_IO
ISZERO:
            RRC   A
            LCALL CLOW
            DJNZ  R7,BITCOUNT
            RET
;*****************************************************
;WRITE DATA 1 BYTE
;*****************************************************
CRD_WRITBIT:
            MOV   R7,#8
            NOP
WRITE_BIT:
            CLR   C
            RRC   A
            MOV   CRD_IO,C
            LCALL PHIGH
            LCALL CHIGH
            LCALL CLOW
            DJNZ  R7,WRITE_BIT
            RET
;*****************************************************
;RESET CARD
;*****************************************************
CRD_RESET:
            SETB  CRD_RST
            NOP
            LCALL CHIGH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LCALL CLOW
        CLR   CRD_RST
        LCALL PHIGH
        RET
;*****************************************************
;CLOCK FOR READ
;*****************************************************
READ_CK:
        LCALL CHIGH
        LCALL CLOW
        RET
;*****************************************************
;CLEAR DISBUFF
;*****************************************************
CLEAR_DISBUFF:
        MOV   R0,#DISBUFF
        MOV   B,#16
CLEAR_DIS:
        MOV   A,#' '
        MOV   @R0,A
        INC   R0
        DJNZ  B,CLEAR_DIS
        RET
;*****************************************************
;CLEAR CARDBUFF
;*****************************************************
CLEAR_CARDBUFF:
        MOV   R0,#CARD_BUFF
        MOV   B,#8
CLEAR_CARD:
        MOV   A,#' '
        MOV   @R0,A
        INC   R0
        DJNZ  B,CLEAR_CARD
        RET
;*****************************************************
;CLEAR CARDBUFF1
;*****************************************************
CLEAR_CARDBUFF1:
        MOV   R0,#CARD_BUFF_1
        MOV   B,#8
CLEAR_CARD1:
        MOV   A,#' '
        MOV   @R0,A
        INC   R0
        DJNZ  B,CLEAR_CARD1
        RET
;*****************************************************
;CLEAR CARDBUFF2
;*****************************************************
CLEAR_CARDBUFF2:
        MOV   R0,#CARD_BUFF_2
        MOV   B,#13
CLEAR_CARD21:
        MOV   A,#' '
        MOV   @R0,A
        INC   R0
        DJNZ  B,CLEAR_CARD21
        RET
;*****************************************************
;CLEAR N_TEST
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************
CLEAR_N_TEST:
            MOV    R0,#N_TEST
            MOV    B,#8
CLEAR_N_TEST1:
            MOV    A,#' '
            MOV    @R0,A
            INC    R0
            DJNZ   B,CLEAR_N_TEST1
            RET
;*****************************************************
;CLEAR ID_TEST
;*****************************************************
CLEAR_ID_TEST:
            MOV    R0,#ID_TEST
            MOV    B,#13
CLEAR_ID_TEST1:
            MOV    A,#' '
            MOV    @R0,A
            INC    R0
            DJNZ   B,CLEAR_ID_TEST1
            RET
;*****************************************************
;CLEAR BLOOD_TEST
;*****************************************************
CLEAR_B_TEST:
            MOV    R0,#BLOOD_TEST
            MOV    B,#2
CLEAR_B_TEST1:
            MOV    A,#' '
            MOV    @R0,A
            INC    R0
            DJNZ   B,CLEAR_B_TEST1
            RET
;*****************************************************
;COPY MESSAGE TO DISBUFF
;*****************************************************
COPY_FILE:
            CLR    A
            MOVC   A,@A+DPTR
            MOV    @R0,A
            INC    R0
            INC    DPTR
            DJNZ   B,COPY_FILE
            RET
;*****************************************************
;CLOCK AND DELAY FOR TOOL
;*****************************************************
CHIGH:
            SETB   CRD_CLK
PHIGH:
            NOP
            NOP
            NOP
            NOP
            NOP
            RET
CLOW:
            CLR    CRD_CLK
            NOP
            NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        NOP
        NOP
        NOP
        RET
;*****************************************************
;POWER OFF CARD
;*****************************************************
PWR_CARD_OFF:
        SETB  CRD_PWR
        MOV   R2,#2
        LCALL DTSEC
        RET
PWR_CARD:
        SETB  CRD_PWR
        JNB   FLAG_PWR,OFF
        CLR   CRD_PWR
OFF:
        MOV   R2,#2
        LCALL DTSEC
        RET
;*****************************************************
;CHECK CARD TAKE OUT FROM SLOTCARD
;*****************************************************
CRD_CARDOUT:
        SETB  EA
        JNB   CRD_IN,$
        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#REMOVE_1
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#REMOVE_2
        LCALL WRLINE_LCD
        MOV   R2,#0CH
        LCALL DTSEC
        RET
;*****************************************************
;CHECK CARD INSERT TO SLOTCARD
;*****************************************************
CRD_CARDIN:
        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#INSERT_11
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#INSERT_12
        LCALL WRLINE_LCD
        JB    CRD_IN,$
        MOV   LCD_ADDR,#000H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#INSERT_21
        LCALL WRLINE_LCD
        MOV   LCD_ADDR,#040H
        LCALL SET_ADDR_LCD
        MOV   DPTR,#INSERT_22
        LCALL WRLINE_LCD
        MOV   R2,#5
        LCALL LCD_DELAY
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*******************************************************
;INITIAL LCD
;*******************************************************
INIT_LCD:
            LCALL DELAY_100mS
            CLR   LCD_RS
            MOV   P0,#00111000B
            LCALL LCD_CLK
            LCALL DELAY_10mS
            LCALL LCD_OFF
            LCALL LCD_CLR
            MOV   P0,#00000110B
            LCALL LCD_CLK
            LCALL LCD_HOME
            RET
LCD_CLR:
            CLR   LCD_RS
            MOV   P0,#00000001B
            LCALL LCD_CLK
            RET
LCD_LSHF:
            CLR   LCD_RS
            MOV   P0,#00011000B
            LCALL LCD_CLK
            RET
LCD_HOME:
            CLR   LCD_RS
            MOV   P0,#00000010B
            LCALL LCD_CLK
            RET
LCD_OFF:
            CLR   LCD_RS
            MOV   P0,#00001000B
            LCALL LCD_CLK
            RET
LCD_CLK:
            SETB  LCD_EN
            LCALL LCD_DELAY
            CLR   LCD_EN
            LCALL LCD_DELAY
            RET
LCD_ON:
            CLR   LCD_RS
            MOV   P0,#00001100B
            LCALL LCD_CLK
            RET
LCD_BLINK:
            CLR   LCD_RS
            MOV   P0,#00001111B
            LCALL LCD_CLK
            RET
SET_ADDR_LCD:
            CLR   LCD_RS
            MOV   A,LCD_ADDR
            SETB  ACC.7
            MOV   P0,A
            LCALL LCD_CLK
            RET
WRCHAR_LCD:
            SETB  LCD_RS
            MOV   P0,LCD_DATA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL LCD_CLK
            LCALL LCD_ON
            RET
WRLINE_LCD:
            MOV   R5,#0
WRLINE_LCD_1:
            SETB  LCD_RS
            CLR   A
            MOVC  A,@A+DPTR
            MOV   P0,A
            LCALL LCD_CLK
            INC   DPTR
            INC   R5
            CJNE  R5,#8,WRLINE_LCD_1
            LCALL LCD_ON
            RET
WRITE_FILE:
            MOV   R0,#DISBUFF
            MOV   LCD_ADDR,#040H
            LCALL SET_ADDR_LCD
            MOV   A,R4
            MOV   B,A
            LCALL WR_LCD
            RET
WR_LCD:
            MOV   A,@R0
            MOV   LCD_DATA,A
            LCALL WRCHAR_LCD
            INC   R0
            DJNZ  B,WR_LCD
            RET
;*****************************************************
;EXCHANGE DATA HEXADACIMAL TO ASCII CODE
;*****************************************************
HTOASC:
            PUSH  ACC
            SWAP  A
            LCALL HTOAS
            MOV   R2,A
            POP   ACC
            LCALL HTOAS
            MOV   R3,A
            RET
HTOAS:
            ANL   A,#0FH
            CJNE  A,#0AH,$+3
            JNC   HTOAS1
            ORL   A,#30H
            RET
HTOAS1:
            SUBB  A,#9
            ORL   A,#40H
            RET
;*****************************************************
;DELAY
;*****************************************************
LCD_DELAY:
            MOV   R7,#002
LCD_DELAY_1:
            MOV   R6,#0E6H
LCD_DELAY_2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            NOP
            NOP
            DJNZ  R6,LCD_DELAY_2
            DJNZ  R7,LCD_DELAY_1
            RET
DELAY_10mS:
            MOV   R7,#010
DELAY_10mS_1:
            MOV   R6,#0E6H
DELAY_10mS_2:
            NOP
            NOP
            DJNZ  R6,DELAY_10mS_2
            DJNZ  R7,DELAY_10mS_1
            RET
DELAY_100mS:
            MOV   R7,#100
DELAY_100mS_1:
            MOV   R6,#0E6H
DELAY_100mS_2:
            NOP
            NOP
            DJNZ  R6,DELAY_100mS_2
            DJNZ  R7,DELAY_100mS_1
            RET
DELAY_1s:
            MOV   R5,#100
DELAY_1s_1:
            LCALL DELAY_10mS
            DJNZ  R5,DELAY_1s_1
            RET
DTSEC:
            MOV   R6,#179
DTSEC1:
            MOV   R7,#0FFH
            DJNZ  R7,$
            DJNZ  R6,DTSEC1
            DJNZ  R2,DTSEC
            RET
;**************************************************
;LOAD DATA TO LCD
;**************************************************
LOAD_TOLCD:
            MOV   A,@R1
            MOV   @R0,A
            INC   R0
            INC   R1
            DJNZ  B,LOAD_TOLCD
            LCALL WRITE_FILE
            RET
;**************************************************
;BACKUP CARD_BUFF
;**************************************************
BACKUP_CARD:
            MOV   B,#8
BACKUP_CARD_1:
            MOV   A,@R1
            MOV   @R0,A
            INC   R0
            INC   R1
            DJNZ  B,BACKUP_CARD_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
;*************************************************
;BACKUP CARD_BUFF2
;*************************************************
BACKUP_CARD2:
            MOV    B,#13
BACKUP_CARD_21:
            MOV    A,@R1
            MOV    @R0,A
            INC    R0
            INC    R1
            DJNZ   B,BACKUP_CARD_21
            RET
;*************************************************
;BACKUP CARD_BUFF3
;*************************************************
BACKUP_CARD3:
            MOV    B,#2
BACKUP_CARD_31:
            MOV    A,@R1
            MOV    @R0,A
            INC    R0
            INC    R1
            DJNZ   B,BACKUP_CARD_31
            RET
;*************************************************
;CHECK BYTE IS COUNT NUMBER OF BYTE
;*************************************************
CHECK_BYTE:
            MOV    R1,#CARD_BUFF_1
            MOV    A,@R1
            ANL    A,#11111111B
            CJNE   A,#00,CHECK_BYTE_1
            MOV    R1,#CARD_BUFF_1+1
            MOV    A,@R1
            ANL    A,#11110000B
            CJNE   A,#00,CHECK_BYTE_2
            MOV    R1,#CARD_BUFF_1+1
            MOV    A,@R1
            SWAP   A
            DEC    R1
            MOV    @R1,A
            MOV    R4,#01
            MOV    R1,#CARD_BUFF_1
            RET
CHECK_BYTE_2:
            MOV    R1,#CARD_BUFF_1+1
            MOV    A,@R1
            DEC    R1
            MOV    @R1,A
            MOV    R4,#02
            MOV    R1,#CARD_BUFF_1
            RET
CHECK_BYTE_1:
            MOV    R1,#CARD_BUFF_1
            MOV    A,@R1
            ANL    A,#11110000B
            CJNE   A,#00,CHECK_BYTE_3
            MOV    A,@R1
            SWAP   A
            MOV    B,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            INC   R1
            MOV   A,@R1
            ANL   A,#11110000B
            SWAP  A
            ORL   A,B
            DEC   R1
            MOV   @R1,A
            INC   R1
            MOV   A,@R1
            SWAP  A
            MOV   @R1,A
            MOV   R4,#03
            MOV   R1,#CARD_BUFF_1
            RET
CHECK_BYTE_3:
            MOV   R4,#08
            MOV   R1,#CARD_BUFF_1
            RET
;*****************************************************
;RECEIVE DATA FROM COMPUTER TO SBUF
;*****************************************************
RX_BYTE:
            JNB   RI,$
            CLR   RI
            MOV   A,SBUF
            RET
;*****************************************************
;SEND DATA TO SBUF
;*****************************************************
SEND_BYTE:
            MOV   SBUF,A
            JNB   TI,$
            CLR   TI
            RET
;*****************************************************
;           COMMAND, ADDRESS, DATA
;*****************************************************
READ_ATR:   DB    30H, 00H,     CLEAR
READ_CMD_1: DB    30H, 32D,     CLEAR
READ_CMD_2: DB    30H, 40D,     CLEAR
READ_CMD_3: DB    30H, 48D,     CLEAR
READ_CMD_31:      DB    30H, 48D,     CLEAR
READ_CMD_32:      DB    30H, 56D,     CLEAR
READ_CMD_4: DB    30H, 61D,     CLEAR
READ_CMD_5: DB    30H, 63D,     CLEAR
READ_CMD_51:      DB    30H, 71D,     CLEAR
READ_CMD_52:      DB    30H, 79D,     CLEAR
READ_CMD_6: DB    30H, 81D,     CLEAR
READ_PSC:   DB    31H, CLEAR,   CLEAR
UPDATE_SC:  DB    39H, 00H,   CLEAR
PIN1_SC:    DB    33H, 01H,   0FFH
PIN2_SC:    DB    33H, 02H,   0FFH
PIN3_SC:    DB    33H, 03H,   0FFH
UPDATE_FF:  DB    39H, 00H,   0FFH
WRITE_PROG: DB    3CH, 1FH,   0FEH
UPDATE_MAIN_1:    DB    38H, 032D,  CLEAR
UPDATE_MAIN_2:    DB    38H, 033D,  CLEAR
UPDATE_MAIN_3:    DB    38H, 034D,  CLEAR
UPDATE_MAIN_4:    DB    38H, 035D,  CLEAR
UPDATE_MAIN_5:    DB    38H, 036D,  CLEAR
UPDATE_MAIN_6:    DB    38H, 037D,  CLEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
UPDATE_MAIN_7:     DB    38H, 038D,  CLEAR
UPDATE_MAIN_8:     DB    38H, 039D,  CLEAR
UPDATE_MAIN_9:     DB    38H, 040D,  CLEAR
UPDATE_MAIN_10:    DB    38H, 041D,  CLEAR
UPDATE_MAIN_11:    DB    38H, 042D,  CLEAR
UPDATE_MAIN_12:    DB    38H, 043D,  CLEAR
UPDATE_MAIN_13:    DB    38H, 044D,  CLEAR
UPDATE_MAIN_14:    DB    38H, 045D,  CLEAR
UPDATE_MAIN_15:    DB    38H, 046D,  CLEAR
UPDATE_MAIN_16:    DB    38H, 047D,  CLEAR
UPDATE_MAIN_17:    DB    38H, 048D,  CLEAR
UPDATE_MAIN_18:    DB    38H, 049D,  CLEAR
UPDATE_MAIN_19:    DB    38H, 050D,  CLEAR
UPDATE_MAIN_20:    DB    38H, 051D,  CLEAR
UPDATE_MAIN_21:    DB    38H, 052D,  CLEAR
UPDATE_MAIN_22:    DB    38H, 053D,  CLEAR
UPDATE_MAIN_23:    DB    38H, 054D,  CLEAR
UPDATE_MAIN_24:    DB    38H, 055D,  CLEAR
UPDATE_MAIN_25:    DB    38H, 056D,  CLEAR
UPDATE_MAIN_26:    DB    38H, 057D,  CLEAR
UPDATE_MAIN_27:    DB    38H, 058D,  CLEAR
UPDATE_MAIN_28:    DB    38H, 059D,  CLEAR
UPDATE_MAIN_29:    DB    38H, 060D,  CLEAR
UPDATE_MAIN_30:    DB    38H, 061D,  CLEAR
UPDATE_MAIN_31:    DB    38H, 062D,  CLEAR
UPDATE_MAIN_32:    DB    38H, 063D,  CLEAR
UPDATE_MAIN_33:    DB    38H, 064D,  CLEAR
UPDATE_MAIN_34:    DB    38H, 065D,  CLEAR
UPDATE_MAIN_35:    DB    38H, 066D,  CLEAR
UPDATE_MAIN_36:    DB    38H, 067D,  CLEAR
UPDATE_MAIN_37:    DB    38H, 068D,  CLEAR
UPDATE_MAIN_38:    DB    38H, 069D,  CLEAR
UPDATE_MAIN_39:    DB    38H, 070D,  CLEAR
UPDATE_MAIN_40:    DB    38H, 071D,  CLEAR
UPDATE_MAIN_41:    DB    38H, 072D,  CLEAR
UPDATE_MAIN_42:    DB    38H, 073D,  CLEAR
UPDATE_MAIN_43:    DB    38H, 074D,  CLEAR
UPDATE_MAIN_44:    DB    38H, 075D,  CLEAR
UPDATE_MAIN_45:    DB    38H, 076D,  CLEAR
UPDATE_MAIN_46:    DB    38H, 077D,  CLEAR
UPDATE_MAIN_47:    DB    38H, 078D,  CLEAR
UPDATE_MAIN_48:    DB    38H, 079D,  CLEAR
UPDATE_MAIN_49:    DB    38H, 080D,  CLEAR
UPDATE_MAIN_50:    DB    38H, 081D,  CLEAR
UPDATE_MAIN_51:    DB    38H, 082D,  CLEAR
UPDATE_MAIN_52:    DB    38H, 083D,  CLEAR
UPDATE_MAIN_53:    DB    38H, 084D,  CLEAR
UPDATE_MAIN_54:    DB    38H, 085D,  CLEAR
UPDATE_MAIN_55:    DB    38H, 086D,  CLEAR
WRONGCARD_1:       DB    'WRONG CA'
WRONGCARD_2:       DB    'RD      '
WR_SUCCESS_1:      DB    'WRITE SU'
WR_SUCCESS_2:      DB    'CCESS   '
NOT_SUCCESS_1:     DB    'NOT SUCC'
NOT_SUCCESS_2:     DB    'ESS     '
CARDBLOCK_1:       DB    'CARD IS '
CARDBLOCK_2:       DB    'BLOCKED!'
WRONGPIN_1: DB    'ACCESS W'
WRONGPIN_2: DB    'RONG!   '
SC_ATR:            DB    'ATR:'
REMOVE_1:   DB    'CARD IS '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
REMOVE_2:     DB     'REMOVED '
INSERT_11:    DB     'INSERT C'
INSERT_12:    DB     'ARD     '
INSERT_21:    DB     'CARD IN '
INSERT_22:    DB     'SLOT    '
NAME:         DB     'NAME:   '
SPACE:               DB     '        '
ID:           DB     'ID:     '
STID:         DB     'ST.ID:  '
BLOOD:               DB     'BLOOD:  '
ACCESS_11:    DB     'ACCESS S'
ACCESS_12:    DB     'YSTEM   '
RECEIVE_11:   DB     'RECEIVE '
RECEIVE_12:   DB     'SUCCESS '
RECORD_DATA_1:       DB     'RECORD D'
RECORD_DATA_2:       DB     'ATA     '
RECORDED_DATA_1:     DB     'RECORDED'
RECORDED_DATA_2:     DB     ' DATA : '
READ_DATA_1:         DB     'READ DAT'
READ_DATA_2:         DB     'A       '
RE_CARD_1:    DB     'REMOVE C'
RE_CARD_2:    DB     'ARD     '
RIGHT_1:      DB     'RIGHT PA'
RIGHT_2:      DB     'SSWORD  '
WRONG_PASS_1:        DB     'WRONG PA'
WRONG_PASS_2:        DB     'SSWORD! '
SELECT_1:     DB     'SELECT M'
SELECT_2:     DB     'ODE     '
PASS_REC_1:   DB     'P.W. IS '
PASS_REC_2:   DB     'RECORDED'
LIB_1:               DB     'ACCESS L'
LIB_2:               DB     'IBRARY  '
HEALTH_1:     DB     'ACCESS H'
HEALTH_2:     DB     'EALTH   '
MED_1:               DB     'MED. IS '
MED_2:               DB     'RECORDED'
STUDENT_1:    DB     'STUDENT '
STUDENT_2:    DB     'SYSTEM  '
REC_HEALTH_1:        DB     'RECORDIN'
REC_HEALTH_2:        DB     'G HEALTH'
DOT_1:               DB     '........'
DOT_2:               DB     '...     '

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIEMENS

ICs for Chip Cards

Intelligent 256-Byte EEPROM
SLE 4432/SLE 4442

Data Sheet 07.95

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

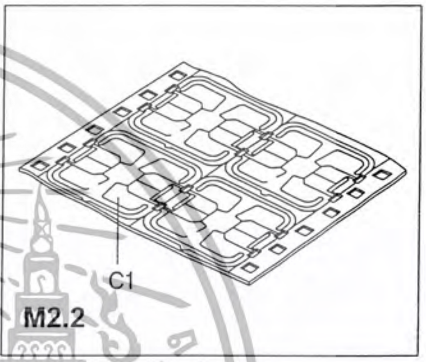
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIEMENS

Intelligent 256-Byte EEPROM with Write Protect Function — SLE 4432

Intelligent 256-Byte EEPROM with Write Protect Function and Programmable Security Code (PSC) — SLE 4442

Features



- 256 × 8-bit EEPROM organization
- Byte-wise addressing
- Irreversible byte-wise write protection of lowest 32 addresses (Byte 0 ... 31)
- 32 × 1-bit organization of protection memory
- Two-wire link protocol
- End of processing indicated at data output
- Answer-to-Reset acc. to ISO standard 7816-3
- Programming time 2.5 ms per byte for both erasing and writing
- Minimum of 10^4 write/erase cycles[1)]
- Data retention for minimum of ten years[1)]
- Contact configuration and serial interface in accordance with ISO standard 7816 (synchronous transmission)

Additional Feature of SLE 4442

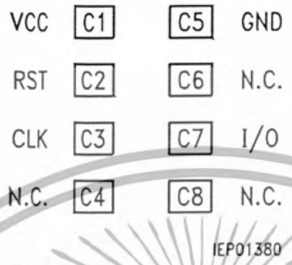
- Data can only be changed after entry of the correct 3-byte programmable security code (security memory)

Type	Ordering Code	Package
SLE 4432 M2.2	on request	Wire-Bonded Module M2.2
SLE 4432 C	on request	Chip
SLE 4442 M2.2	on request	Wire-Bonded Module M2.2
SLE 4442 C	on request	Chip

1) Values are temperature dependent, for further information please refer to your Siemens sales office.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 Pin Configuration
(top view)



M2.2 (Card Contacts)

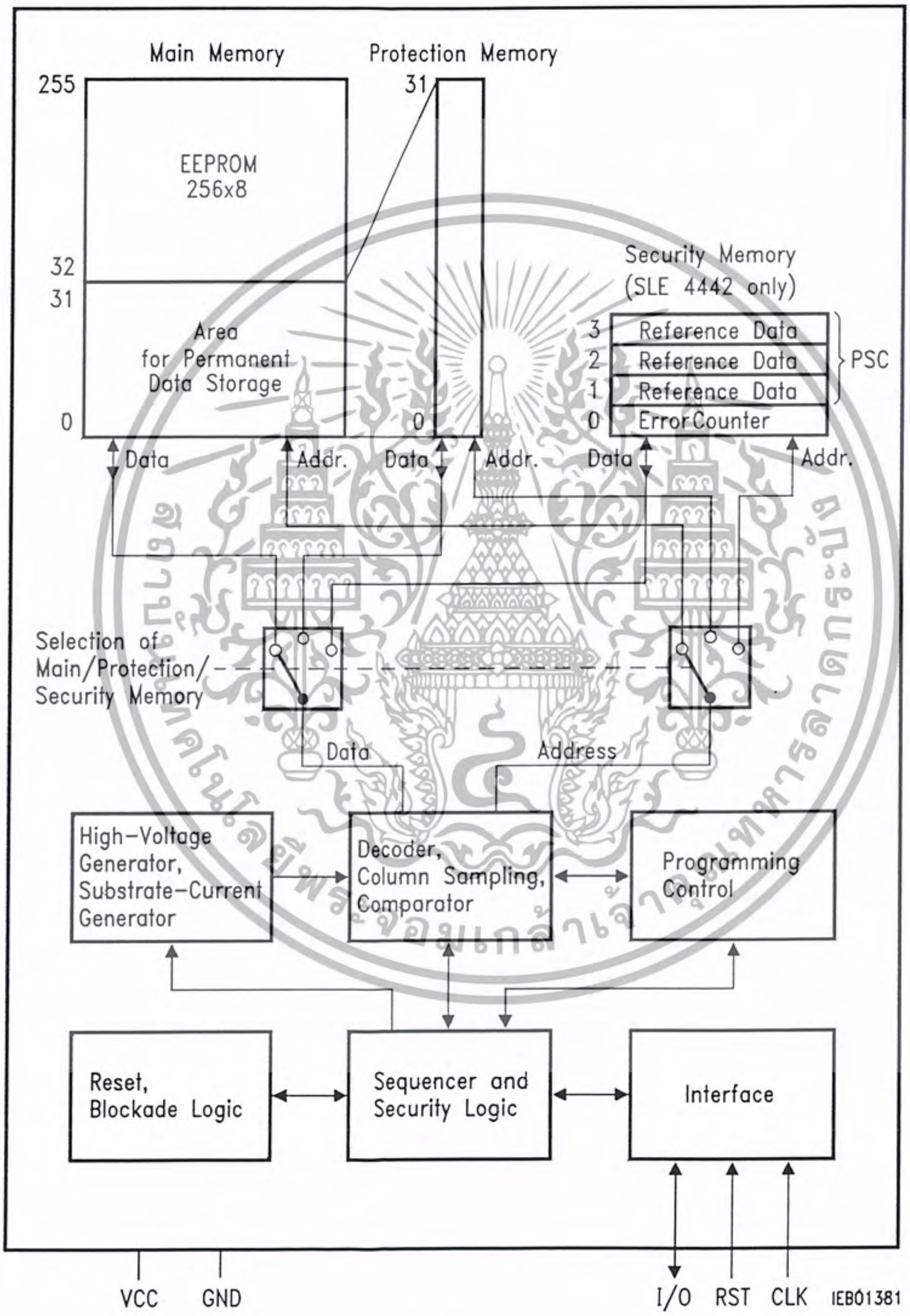
Pin Definitions and Functions

Card Contact	Symbol	Function
C1	VCC	Supply voltage
C2	RST	Reset
C3	CLK	Clock input
C4	N.C.	Not connected
C5	GND	Ground
C6	N.C.	Not connected
C7	I/O	Bidirectional data line (open drain)
C8	N.C.	Not connected

SLE 4432/SLE 4442 comes as a M2.2 wire-bonded module for embedding in plastic cards or as a die for customer packaging.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 Functional Description



Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1 Memory Overview

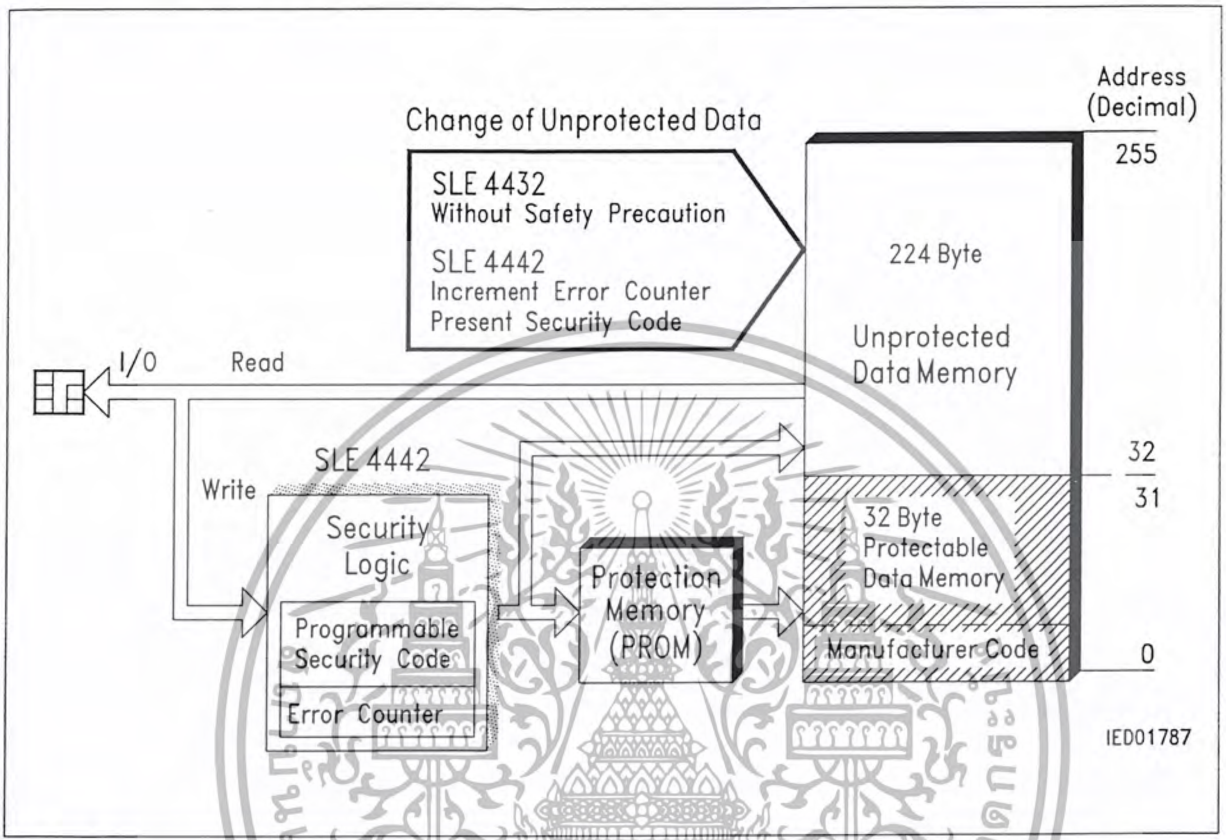


Figure 1
Memory Overwiew

SLE 4432

The SLE 4432 consists of 256 x 8 bit EEPROM main memory and a 32-bit protection memory with PROM functionality. The main memory is erased and written byte by byte. When erased, all 8 bits of a data byte are set to logical one. When written, the information in the individual EEPROM cells is, according to the input data, altered bit by bit to logical zeros (logical AND between the old and the new data in the EEPROM). Normally a data change consists of an erase and write procedure. It depends on the contents of the data byte in the main memory and the new data byte whether the EEPROM is really erased and/or written. If none of the 8 bits in the addressed byte requires a zero-to-one transition the erase access will be suppressed. Vice versa the write access will be suppressed if no one-to-zero transition is necessary. The write and the erase operation takes at least 2.5 ms each.

Each of the first 32 bytes can be irreversibly protected against data change by writing the corresponding bit in the protection memory. Each data byte in this address range is assigned to one bit of the protection memory and has the same address as the data byte in the main memory which it is assigned to. Once written the protection bit cannot be erased (PROM).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLE 4442

Additionally to the above functions the SLE 4442 provides a security code logic which controls the write/erase access to the memory. For this purpose the SLE 4442 contains a 4-byte security memory with an **E**rror **C**ounter EC (bit 0 to bit 2) and 3 bytes reference data. These 3 bytes as a whole are called **P**rogrammable **S**ecurity **C**ode (PSC). After power on the whole memory, except for the reference data, can only be read. Only after a successful comparison of verification data with the internal reference data the memory has the identical access functionality of the SLE 4432 until the power is switched off. After three successive unsuccessful comparisons the **E**rror **C**ounter blocks any subsequent attempt, and hence any possibility to write and erase.

2.2 Transmission Protocol

The transmission protocol is a two wire link protocol between the interface device IFD and the integrated circuit IC. It is identical to the protocol type "S = A". All data changes on I/O are initiated by the falling edge on CLK.

The transmission protocol consists of the 4 modes:

– Reset and Answer-to-Reset
– Command Mode ⎫
– Outgoing Data Mode ⎬ Operational modes
– Processing Mode ⎭

Note: The I/O pin is open drain and therefore requires an external pull up resistor to achieve a high level.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 Reset and Answer-to-Reset

Answer-to-Reset takes place according to ISO standard 7816-3 (ATR). The reset can be given at any time during operation. In the beginning, the address counter is set to zero together with a clock pulse and the first data bit (LSB) is output to I/O when RST is set from level H to level L. Under a continuous input of additional 31 clock pulses the contents of the first 4 EEPROM addresses is read out. The 33rd clock pulse switches I/O to high impedance Z and finishes the ATR procedure.

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

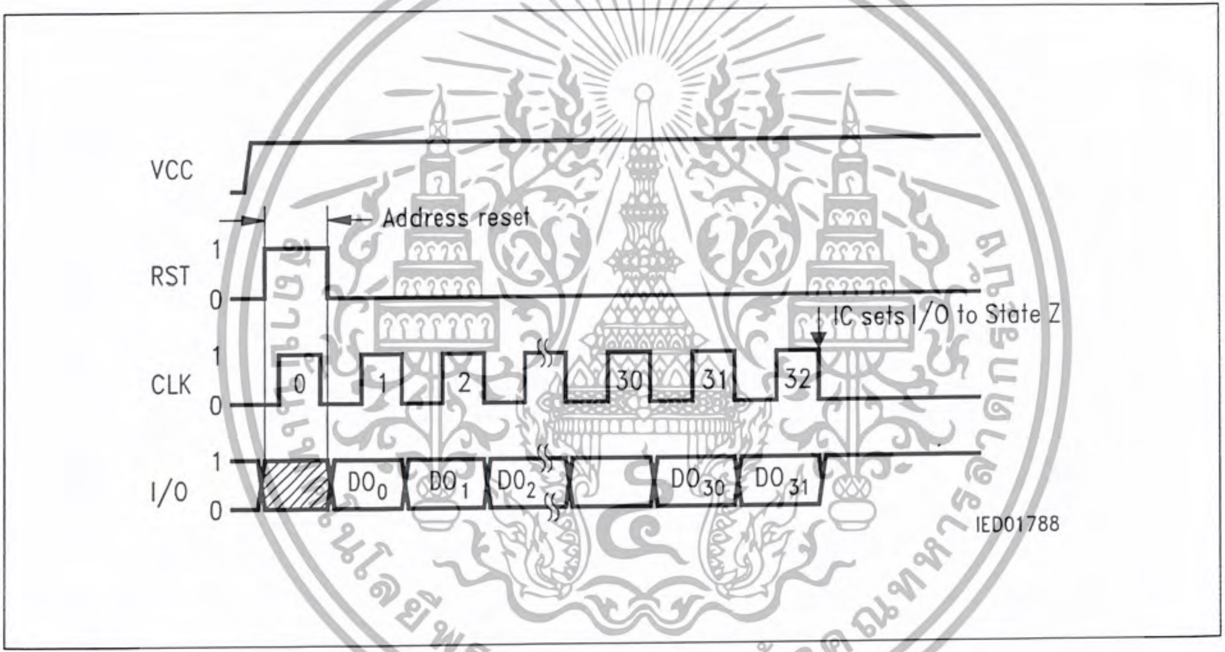


Figure 2
Reset and Answer-to-Reset

2.2.2 Operational Modes

Command Mode

After the Answer-to-Reset the chip waits for a command. Every command begins with a start condition, includes a 3 bytes long command entry followed by an additional clock pulse and ends with a stop condition.

- Start condition: Falling edge on I/O during CLK in level H
- Stop condition: Rising edge on I/O during CLK in level H

After the reception of a command there are two possible modes:

- Outgoing data mode for reading
- Processing mode for writing and erasing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Outgoing Data Mode

In this mode the IC sends data to the IFD. The first bit becomes valid on I/O after the first falling edge on CLK. After the last data bit an additional clock pulse is necessary in order to set I/O to high impedance Z and to prepare the IC for a new command entry. During this mode any start and stop condition is discarded.

Processing Mode

In this mode the IC processes internally. The IC has to be clocked continuously until I/O, which was switched to level L after the first falling edge of CLK, is set to high impedance level Z. Any start and stop condition is discarded during this mode.

Note: The RST line is low during the modes mentioned above. If RST is set to high during the CLK low level any operation is aborted and I/O is switched to high impedance Z (Break).

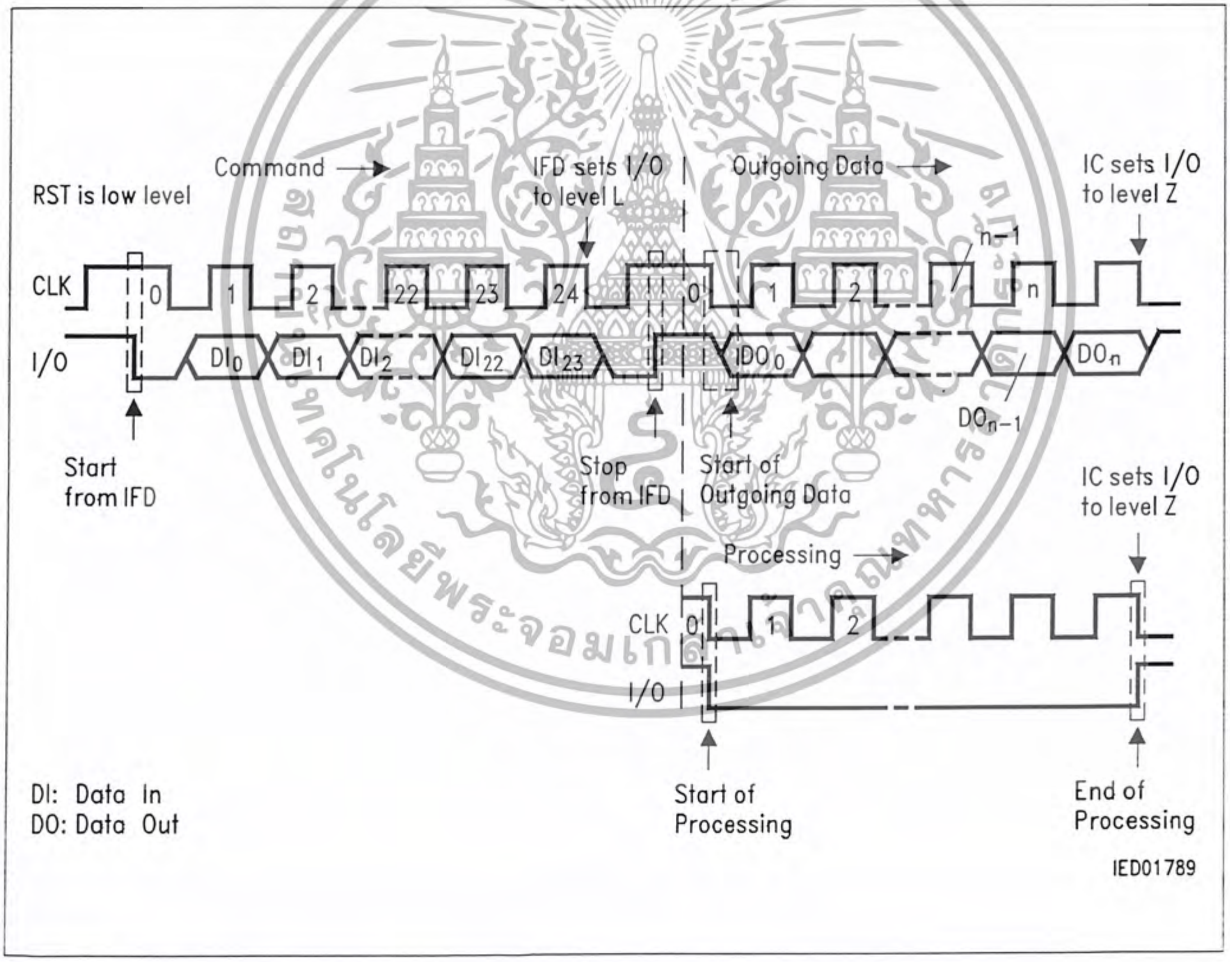


Figure 3
Operational Modes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Commands

Command Format

Each command consists of three bytes:

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

Beginning with the control byte LSB is transmitted first.

Figure 4
Command Mode

The SLE 4432 provides 4 commands which are listed in **table 1**. Additionally to these commands the SLE 4442 provides 3 commands which can be found in **table 2**.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1

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

Table 2
SLE 4442 only

0	0	1	1	0	0	0	1	no effect	no effect	READ SECURITY MEMORY	outgoing data
0	0	1	1	1	0	0	1	address	input data	UPDATE SECURITY MEMORY	processing
0	0	1	1	0	0	1	1	address	input data	COMPARE VERIFICATION DATA	processing

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 Read Main Memory (SLE 4432 and SLE 4442)

The command reads out the contents of the main memory (with LSB first) starting at the given byte address (N = 0…255) up to the end of the memory. After the command entry the IFD has to supply sufficient clock pulses. The number of clocks is $m = (256 - N) \times 8 + 1$. The read access to the main memory is always possible.

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

Command: READ MAIN MEMORY

	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

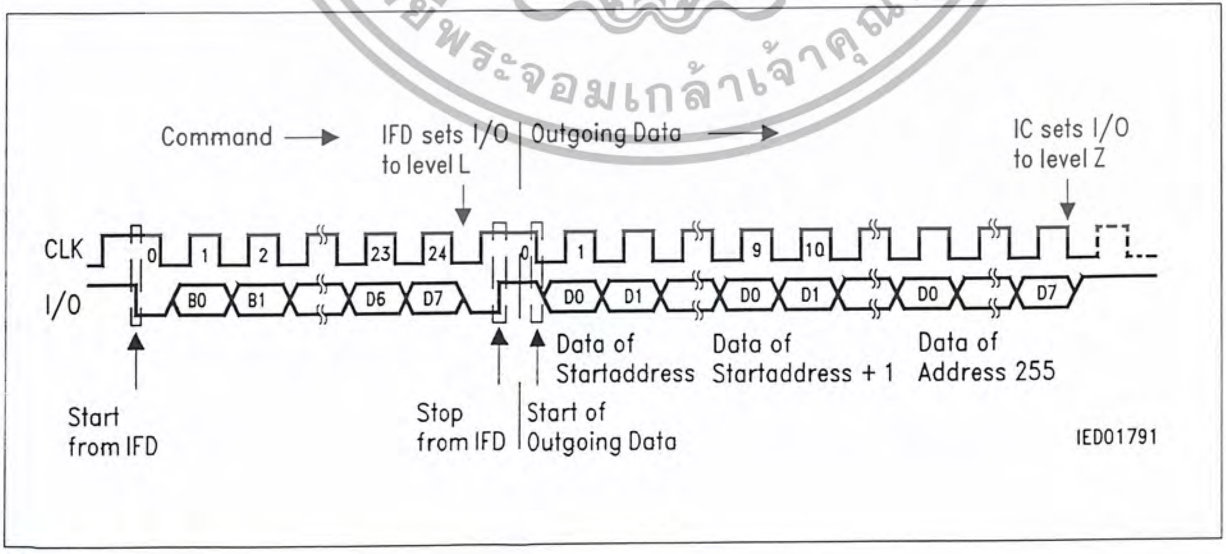


Figure 5
Read Main Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 Read Protection Memory (SLE 4432 and SLE 4442)

The command transfers the protection bits under a continuous input of 32 clock pulses to the output. I/O is switched to high impedance Z by an additional pulse. The protection memory can always be read, and indicates the data bytes of the main memory protected against changing.

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

Command: READ PROTECTION MEMORY

	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

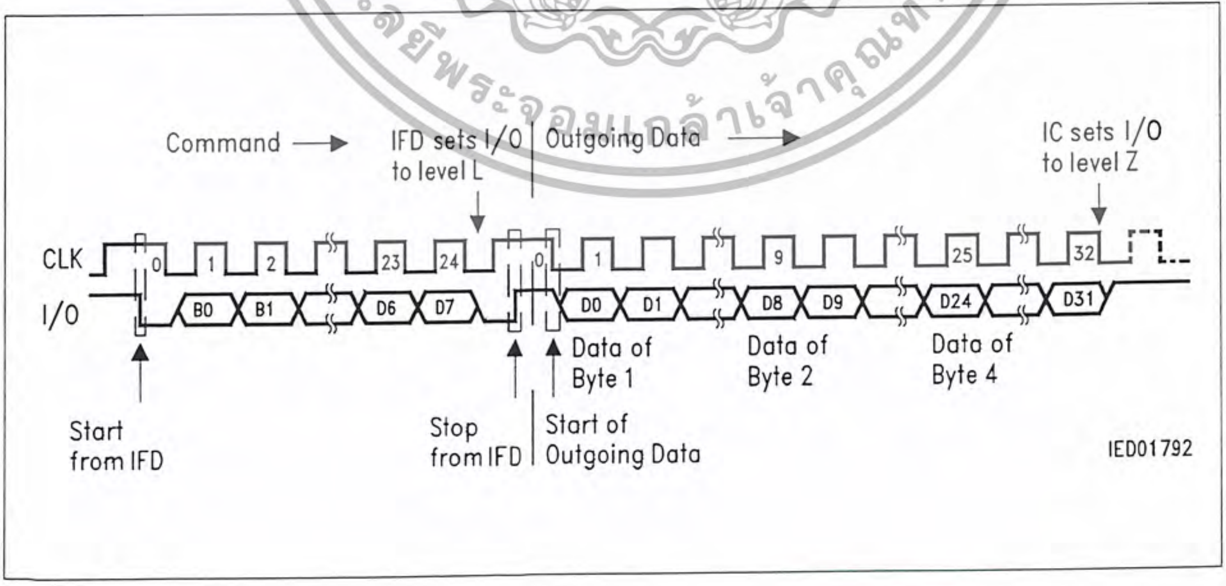


Figure 6
Read Protection Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 Update Main Memory (SLE 4432 and SLE 4442)

The command programs the addressed EEPROM byte with the data byte transmitted. Depending on the old and new data, one of the following sequences will take place during the processing mode:

- erase and write (5 ms) corresponding to m = 255 clock pulses
- write without erase (2.5 ms) corresponding to m = 124 clock pulses
- erase without write (2.5 ms) corresponding to m = 124 clock pulses

(All values at 50 kHz clock rate.)

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

Command: UPDATE MAIN MEMORY

	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

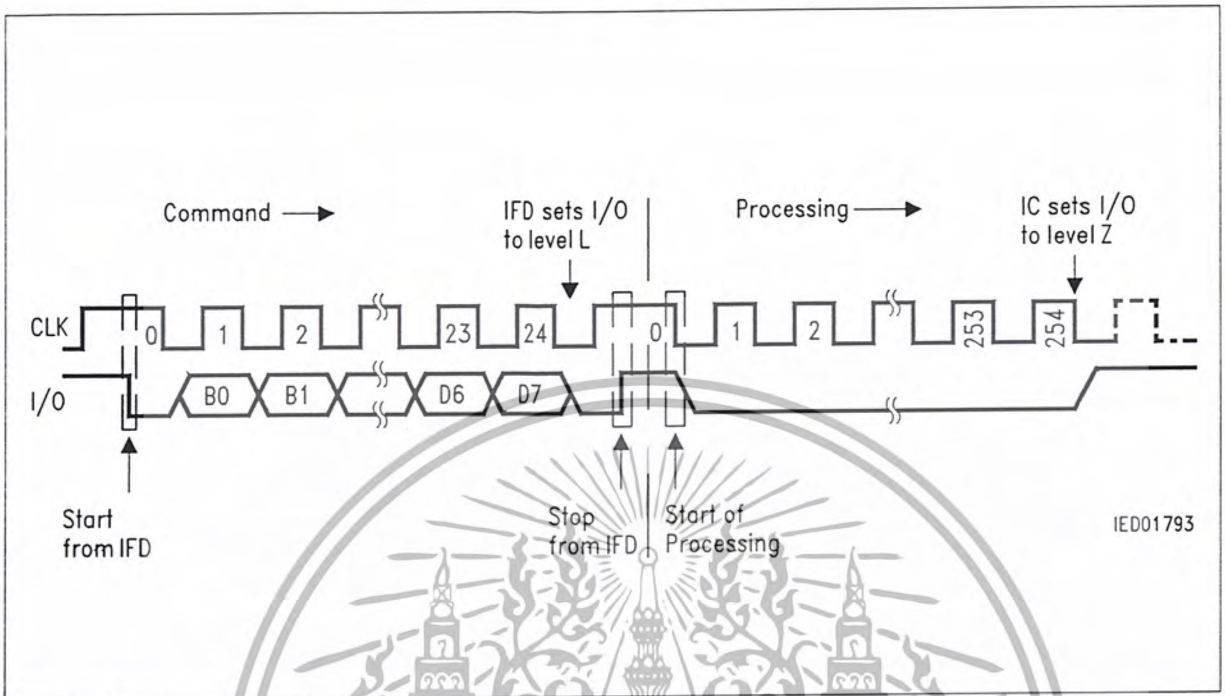


Figure 7
Erase and Write Main Memory

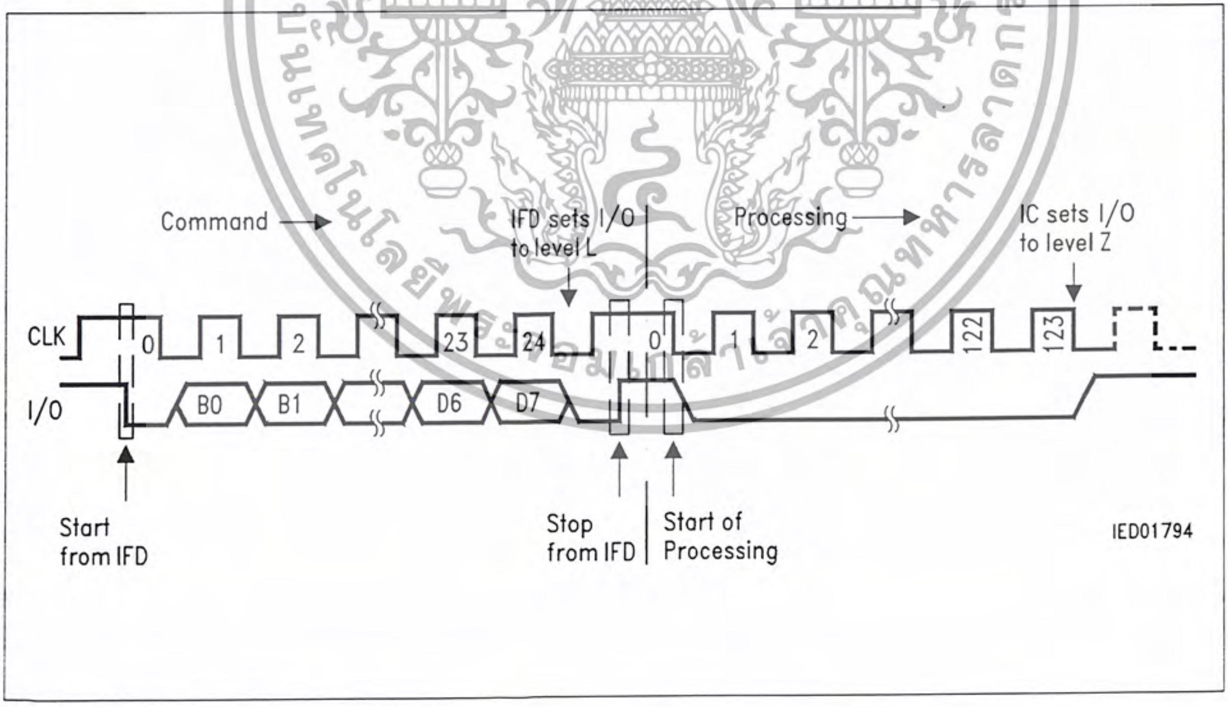


Figure 8
Erase or Write Main Memory

If the addressed byte is protected against changes (indicated by the associated written protection bit) the I/O is set to high impedance after the clock number 2 of the processing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 Write Protection Memory (SLE 4432 and SLE 4442)

The execution of this command contains a comparison of the entered data byte with the assigned byte in the EEPROM. In case of identity the protection bit is written thus making the data information unchangeable. If the data comparison results in data differences writing of the protection bit will be suppressed. Execution times and required clock pulses see UPDATE MAIN MEMORY.

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

Command: WRITE PROTECTION MEMORY

	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.5 Read Security Memory (SLE 4442 only)

Similar to the read command for the protection memory this command reads out the 4 bytes of the security memory. The number of clock pulses during the outgoing data mode is 32. I/O is switched to high impedance Z by an additional pulse. Without a preceeding successful verification of the PSC the output of the reference bytes is suppressed, that means I/O outputs state L for the reference data bytes.

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)

Command: READ SECURITY MEMORY

	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

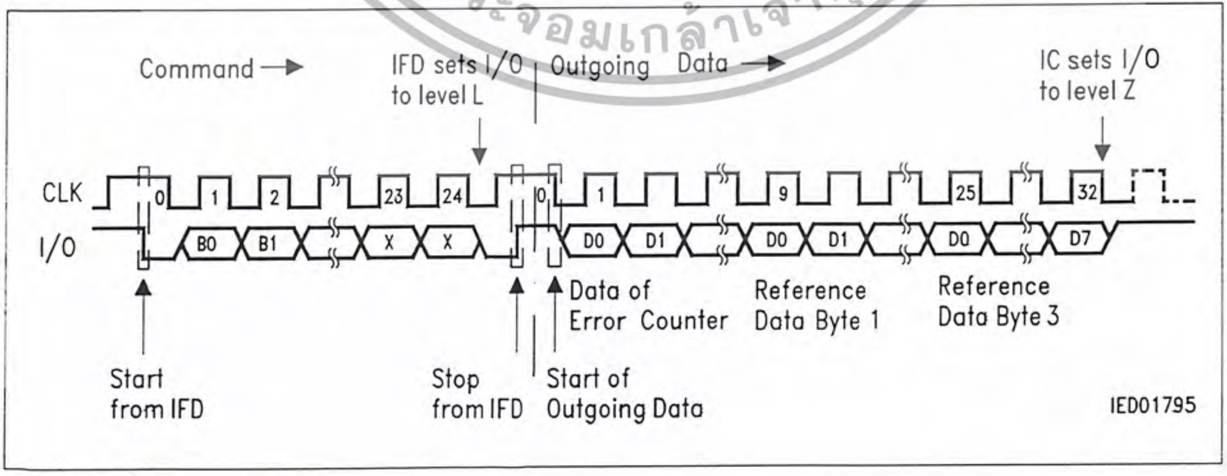


Figure 9
Read Security Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 Update Security Memory (SLE 4442 only)

Regarding the reference data bytes this command will only be executed if a PSC has been successfully verified before. Otherwise only each bit of the error counter (Address 0) can be written from “1” to “0”. The execution times and the required clock pulses are the same as described under UPDATE MAIN MEMORY.

Command: UPDATE SECURITY MEMORY

	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

2.3.7 Compare Verification Data (SLE 4442 only)

This command can only be executed in combination with an update procedure of the error counter (see PSC verification). The command compares one byte of the entered verification data byte with the corresponding reference data byte. For this procedure clock pulses are necessary during the processing mode.

Command: COMPARE VERIFICATION DATA

	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

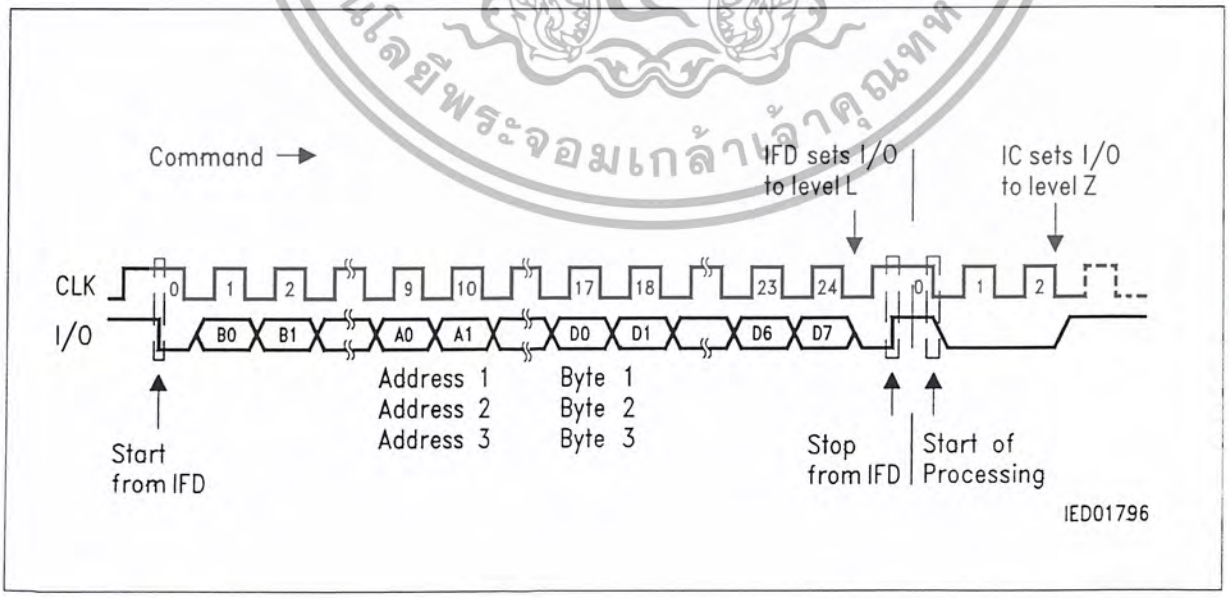


Figure 10
Compare Verification Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 PSC Verification (SLE 4442 only)

The SLE 4442 requires a correct verification of the Programmable Security Code PSC stored in the Security Memory for altering data if desired.

The following procedure has to be carried out exactly as described. Any variation leads to a failure, so that a write/erase access will not be achieved. As long as the procedure has not been successfully concluded the error counter bits can only be changed from "1" to "0" but not erased.

At first an error counter bit has to be written to "0" by an UPDATE command (**see figure 11**) followed by three COMPARE VERIFICATION DATA commands beginning with byte 1 of the reference data. A successful conclusion of the whole procedure can be recognized by being able to erase the error counter which is not automatically erased. Now write/erase access to all memory areas is possible as long as the operating voltage is applied. In case of error the whole procedure can be repeated as long as erased counter bits are available. Having been enabled, the reference data are allowed to be altered like any other information in the EEPROM.

The following table gives an overview of the necessary commands for the PSC verification. The sequence of the shaded commands is mandatory.

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

As shipped, the PSC is programmed with a code according to individual agreement with the customer. Thus, knowledge of this code is indispensable to alter data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

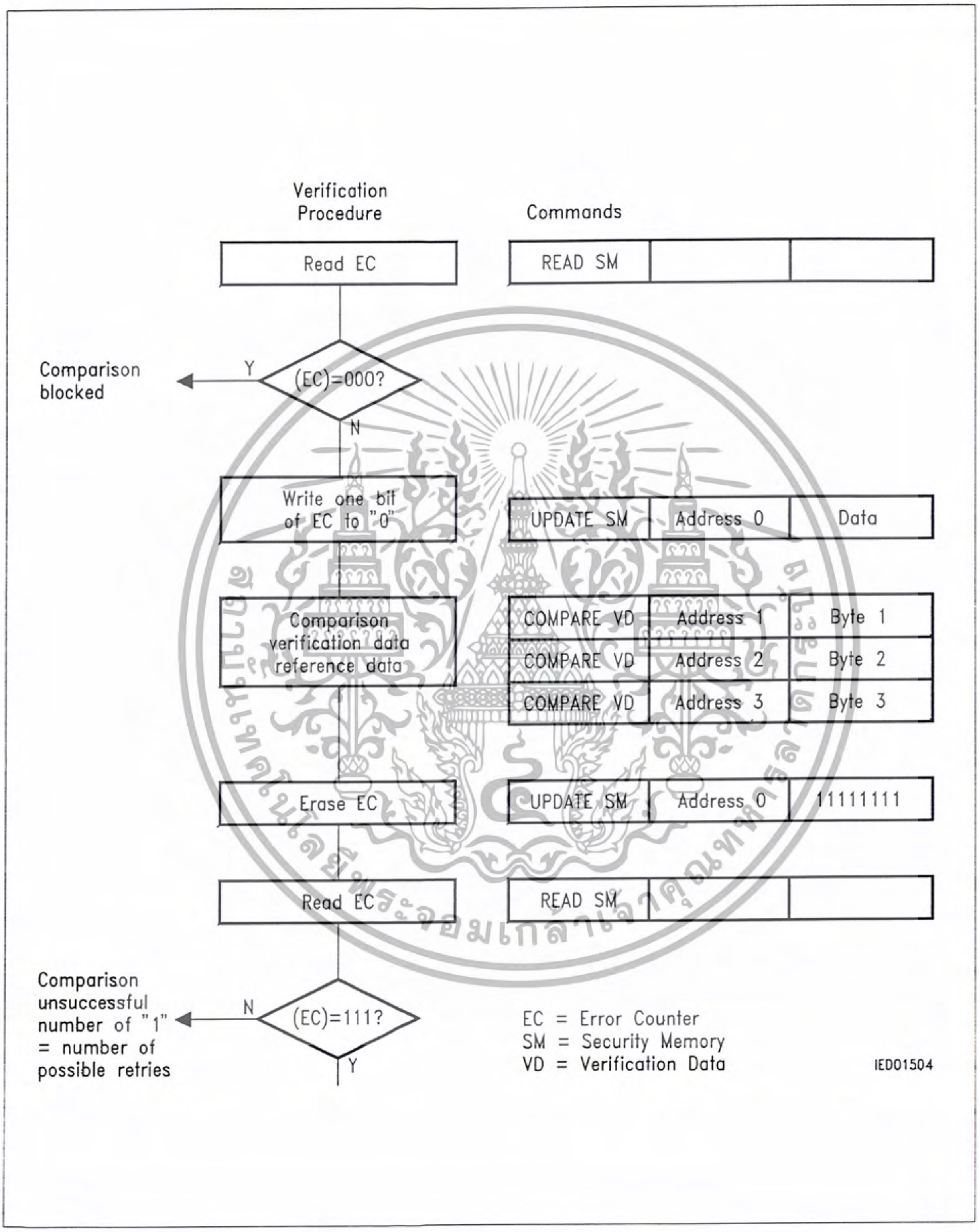


Figure 11
Verification Procedure

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณ รศ.ดร. กอบชัย เดชหาญ ที่ให้ความอนุเคราะห์ช่วยเหลือตลอดการทำโครงงานชิ้นนี้

ขอขอบคุณ อ.รพีพรรณ พิริยะกุล ที่ให้คำแนะนำปรึกษาในการจัดทำฐานข้อมูล (Database)

ขอขอบคุณ พ.ญ. อภิรดี จิรัฐิติกาลโชติ ที่ให้ข้อมูลชื่อโรคประจำตัวและชื่อยาที่แพ้ที่มักพบในผู้ป่วย

ขอขอบคุณ คุณสมชาย พิสิฐสวัสดิ์ ที่ให้ความช่วยเหลือในการจัดทำกล่องใส่ชิ้นงาน

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] สมยศ จุณณะปิยะ **การใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51** (กรุงเทพมหานคร: โครงการตำรา สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543)

[2] วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตรวิไล **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์** (กรุงเทพมหานคร: บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2543)

[3] เลิศ แซ่ตั้ง **เทคโนโลยีสมาร์ทการ์ด** (กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น, 2546)

[4] กิตติ ภักดีวัฒนะกุล และจำลอง ครูอุตสาหะ **Visual Basic 6 ฉบับโปรแกรมเมอร์** (กรุงเทพมหานคร: บริษัท เคทีพี คอมพ์ แอนด์ คอนซัลท์ จำกัด, 2545)

[5] ศุภชัย สมพานิช **Database Programming ด้วย Visual Basic ฉบับมืออาชีพ** (นนทบุรี: อินโฟเพรส, 2543)

[6] อภิชาติ ภู่พลับ **เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic** (นนทบุรี: อินโฟเพรส, 2546)

[7] ทิพย์วรรณ อมรชัยทรัพย์ **จับประเด็นร้อน ๆ Microsoft Access 2002 ฉบับสมบูรณ์** (กรุงเทพมหานคร: บริษัท เฟิสท์ แปซิฟิก (ดอกหญ้า) จำกัด, 2545)

[8] **คู่มือนักศึกษาคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้