สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ปริญญานิพนธ์

เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

CLASS ATTENDANCE CHECKING AND RECORDING MACHINE USING SMART CARD

นายมงคล หมวดเพชร
นายมานะชัย ทึกว้าง
นายสุรเดช สว่างวัน
นางสาวสุปราณี ดอนเตาเหล็ก

เลขหมู่.....
เลขทะเบียน 51859
วัน,เดือน,ปี 3 ส.ค. 2547

ใบ.....
เล่ม.....

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาครุศาสตร์วิศวกรรม

คณะครุศาสตร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อ เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

Class Attendance Checking and Recording Machine using Smart Card

ชื่อนักศึกษา

1. นายมงคล หมวดเพชร รหัสประจำตัว 45035394
2. นายมานะชัย ทีกว้าง รหัสประจำตัว 45035395
3. นางสาวสุปราณี ดอนเตาเหล็ก รหัสประจำตัว 45035407
4. นายสุรเดช สว่างวัน รหัสประจำตัว 45035408

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ที่ปรึกษา ผศ.กิติพงค์ มะโน

อาจารย์ที่ปรึกษาร่วม อาจารย์สุรพงษ์ สิริพงศ์ดี

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1. ผศ.พีระวุฒิ สุวรรณจันทร์	
2. อาจารย์พิชญ์สินี มงคลขจิต	
3. ผศ.กิติพงค์ มะโน	
4. อาจารย์โกศล ตราชู	
5. อาจารย์สมชาย หมื่นสายญาติ	

วัน/เดือน/ปีที่สอบ วันพฤหัสบดีที่ 1 เมษายน พ.ศ. 2547 เวลา 13.00 น.

สถานที่สอบ ห้อง ค.311 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม.....

(ผศ.สุรสิทธิ์ ราตรี)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่...1...เดือน...มี.ค...พ.ศ...๔๒...

<BT4620272>

เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

Class Attendance Checking and Recording Machine using Smart Card

วัตถุประสงค์

1. เพื่อศึกษาการทำงานของบาร์โค้ดกับการทำงานของสมาร์ตการ์ดและการเขียนโปรแกรมควบคุมการเขียน – อ่านสมาร์ตการ์ด
2. เพื่อทำการออกแบบเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด
3. เพื่อสร้างเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดและเขียนโปรแกรมควบคุมการเขียน – อ่านสมาร์ตการ์ด
4. เพื่อทดลองการทำงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดและทดลองการทำงานของโปรแกรมควบคุมการเขียน – อ่านสมาร์ตการ์ด
5. เพื่อนำไปใช้สำหรับการตรวจสอบและบันทึกการเข้าชั้นเรียนได้จริง

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับความรู้และเข้าใจการทำงานของบาร์โค้ดกับการทำงานของสมาร์ตการ์ดและการเขียนโปรแกรมควบคุมการเขียน – อ่านสมาร์ตการ์ด
2. ได้ต้นแบบเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด
3. ได้เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดและได้โปรแกรมควบคุมการเขียน – อ่านสมาร์ตการ์ด
4. ได้ผลการทดลองของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดและผลการทดลองของโปรแกรมควบคุมการเขียน – อ่านสมาร์ตการ์ด
5. ได้นำเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดนำไปใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อหัวข้อ	เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด
นักศึกษา	นายมงคล หมวดเพชร์
	นายมานะชัย ทีกว้าง
	นายสุรเดช สว่างวัน
	นางสาวสุปราณี ดอนเตาเหล็ก
อาจารย์ที่ปรึกษา	ผศ.กิติพงค์ มะโน
อาจารย์ที่ปรึกษาร่วม	อาจารย์สุรพงษ์ สิริพงศ์ดี
หลักสูตร	ครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชา	อิเล็กทรอนิกส์และคอมพิวเตอร์
ปีการศึกษา	2546

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการออกแบบและสร้างเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด โดยแบ่งการทำงานออกเป็น 2 ส่วน คือ ส่วนที่หนึ่งเป็นส่วนที่ใช้ตรวจสอบและบันทึกการเข้าชั้นเรียน ส่วนที่สองใช้ติดต่อสมาร์ตการ์ดกับฐานข้อมูลในคอมพิวเตอร์ ข้อมูลที่เก็บอยู่ในสมาร์ตการ์ดสามารถนำมาตรวจสอบว่ามีนักศึกษาคนใดมาเข้าเรียนขาดเรียนหรือมาสายโดยตรวจสอบจากการรูดบัตรนักศึกษา ทำให้ช่วยลดปัญหาในการตรวจสอบรายชื่อนักศึกษาในการเข้าชั้นเรียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Class Attendance Checking and Recording Machine using Smart Card	
Students	Mr.Mongkol	Maodpet
	Mr.Manachai	Teekwang
	Mr.Suradech	Sawangwan
	Miss.Supranee	Dontaolek
Advisor	Asst.Prof.Kitipong	Mano
Co-Advisor	Mr.Surapong	Siripongdee
Education Level	Bachelor of Science in Industrial Education	
Program in	Electronics and Computer	
Academic Year	2003	

ABSTRACT

This thesis presents design and implementation the Class Attendance Checking and Recording Machine using Smart Card. Which divide into two parts. The first part is used for checking and recording attendance. The second part is the communication between smart card and database in computer store in the smart card. The database is used for checking that students are presenting, lately coming and absenting by checking from student identity card. This system is to reduce problem in checking student name in the class.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกท่าน ขอขอบพระคุณผู้ช่วยศาสตราจารย์กิติพงค์ มะโน อาจารย์ที่ปรึกษา อาจารย์สุรพงษ์ สิริพงศ์ดี อาจารย์ที่ปรึกษาร่วมและอาจารย์อำพล ทองระอา รวมทั้งคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ได้ให้คำแนะนำ แนวความคิด และความรู้ต่างๆ รวมทั้งแนวทางในการแก้ไขปัญหาที่ได้สละเวลามาให้คำปรึกษาและเพื่อนๆ ร่วมห้องทุกท่านที่ได้ให้ความช่วยเหลือในเรื่องวัสดุอุปกรณ์ และเป็นกำลังใจในการทำโครงงานชิ้นนี้จนประสบผลสำเร็จด้วยดี

สุดท้ายขอพระคุณ บิดา มารดา ที่ได้ทำงานด้วยความเหน็ดเหนื่อยเพื่อให้บุตรได้รับการศึกษาที่สูงที่สุดและได้ให้ทุนในการทำโครงงานชิ้นนี้ คณาจารย์ที่ได้อบรมสั่งสอนมาในอดีตจนถึงปัจจุบันรวมถึงญาติพี่น้องที่เป็นผู้ให้การสนับสนุนด้านการศึกษาและเป็นผู้ให้กำลังใจด้วยดีตลอดมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญรูป	VII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 ขีดความสามารถของโครงงาน	1
1.3 เนื้อหาโดยสังเขป	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 กล่าวนำ	3
2.2 สมาร์ตการ์ด	3
2.2.1 การแบ่งชนิดของสมาร์ตการ์ด	3
2.2.2 ข้อดีของสมาร์ตการ์ด	8
2.2.3 การประยุกต์ใช้งานสมาร์ตการ์ด	8
2.2.4 มาตรฐาน ISO-7816	8
2.2.5 รูปแบบการเชื่อมต่อของสมาร์ตการ์ดแบบมีหน้าสัมผัส	12
2.3 การสื่อสารข้อมูลแบบอนุกรม	13
2.3.1 การสื่อสารข้อมูลอนุกรมมาตรฐาน RS-232	15
2.4 การติดต่อสื่อสารโดยระบบ I^2C บัส	20
2.5 บาร์โค้ด	20
2.5.1 ความเป็นมาของบาร์โค้ด	20
2.5.2 หลักการของบาร์โค้ด	21
2.6 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51	25
2.7 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ MCS-51	26
2.8 คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ AT89C52	29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
2.9 โครงสร้างและการทำงานของพอร์ต	30
2.9.1 การใช้งานพอร์ตอินพุต	30
2.9.2 การใช้งานพอร์ตเอาต์พุต	31
2.11 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล	31
2.11.1 คำศัพท์พื้นฐาน	31
2.11.2 เอนทิตี, แอททริบิวต์, และความสัมพันธ์	32
2.11.3 รีเลชั่น	34
2.12 ฐานข้อมูล	34
2.12.1 ระบบการจัดเก็บฐานข้อมูล	35
2.12.2 โมเดล	37
2.12.3 ประโยชน์จากการประมวลผลฐานข้อมูล	38
2.13 สถาปัตยกรรมของระบบฐานข้อมูล	40
บทที่ 3 การออกแบบ การสร้าง และการทำงาน	42
3.1 กล่าวนำ	42
3.2 แผนผังการทำงาน	43
3.2.1 วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด	44
3.2.2 วงจรรับข้อมูลดีย์เมตริกซ์สวิตช์	45
3.2.3 วงจรแสดงผลจอแอลซีดี	45
3.2.4 วงจรภาคจ่ายไฟเลี้ยง	47
3.2.5 โมดูลเครื่องอ่าน – เขียนสมาร์ตการ์ด	47
3.3 การออกแบบและการทำงานของโปรแกรม	49
3.3.1 การออกแบบโปรแกรมของฮาร์ดแวร์	49
3.3.2 การออกแบบโปรแกรมวิชวลเบสิก	49
บทที่ 4 การทดลองและผลการทดลอง	51
4.1 กล่าวนำ	51
4.2 การทดลองเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	51
4.2.1 การทดลองการตรวจสอบและบันทึกข้อมูลลงสมาร์ตการ์ด	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เรื่อง	หน้า
4.2.2 การทดลองการรับข้อมูลจากเครื่องอ่านบาร์โค้ด	55
4.3 การทดลองการทำงานของโปรแกรมอ่าน-เขียนสมาร์ตการ์ด	55
บทที่ 5 บทสรุป	63
5.1 สรุป	63
5.2 ปัญหาและแนวทางการแก้ไข	63
5.2.1 ปัญหาทางด้านแหล่งข้อมูลของสมาร์ตการ์ด	63
5.2.2 ปัญหาทางด้านฮาร์ดแวร์	64
5.2.3 ปัญหาทางด้านซอฟต์แวร์	64
5.3 แนวทางการพัฒนา	65
บรรณานุกรม	66
ภาคผนวก ก เครื่องต้นแบบ	67
ภาคผนวก ข วงจรและแผ่นวงจรพิมพ์	71
ภาคผนวก ค รายการอุปกรณ์	76
ภาคผนวก ง แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม	79
ภาคผนวก จ คู่มือการใช้งาน	143
ภาคผนวก ฉ รายละเอียดและคุณสมบัติของอุปกรณ์	161
ประวัติผู้แต่ง	199

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ขนาดของตำแหน่งหน้าสัมผัสบนสมาร์ตการ์ด C1-C8	12
2.2 การจัดขาคอนเนคเตอร์พอร์ตอนุกรมแบบ DB-9 และ DB-25	18
2.3 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลทาง EIA	19
2.4 หน้าที่พิเศษของแต่ละขาของพอร์ต P3	28
ค.1 รายการอุปกรณ์วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียน โดยสมาร์ตการ์ด	77
ค.1 (ต่อ) รายการอุปกรณ์วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียน โดยสมาร์ตการ์ด	78
ง.1 ฐานข้อมูลคณะ	137
ง.2 ฐานข้อมูลภาควิชา	137
ง.3 ฐานข้อมูลสาขาวิชา	137
ง.4 ฐานข้อมูลวิชาเรียน	137
ง.5 ฐานข้อมูลจำนวนครั้งที่อาจารย์สอน	137
ง.6 ฐานข้อมูลนักศึกษา	138
ง.6 (ต่อ) ข้อมูลนักศึกษา	139
ง.7 ฐานข้อมูลการลงทะเบียนของนักศึกษา	139
ง.7 (ต่อ) ฐานข้อมูลการลงทะเบียนของนักศึกษา	140
ง.8 ฐานข้อมูลการเข้าเรียนของนักศึกษา	141
ง.8 (ต่อ) ฐานข้อมูลการเข้าชั้นเรียนของนักศึกษา	142

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่	หน้า
2.1 สมาร์ตการ์ดแบบมีหน้าสัมผัส	4
2.2 สมาร์ตการ์ดแบบที่มีหน่วยความจำเพียงอย่างเดียว	5
2.3 สมาร์ตการ์ดแบบที่มีหน่วยประมวลผลกลางอยู่ในตัว	5
2.4 โครงสร้างภายในชิปแบบมีหน่วยประมวลอยู่ในตัว	6
2.5 สมาร์ตการ์ดแบบไม่มีหน้าสัมผัส	7
2.6 ขนาดความกว้าง ความยาว และความหนาของสมาร์ตการ์ด	9
2.7 ตำแหน่งหน้าสัมผัสที่ผิวหน้าของสมาร์ตการ์ดตามมาตรฐาน ISO7816-1	10
2.8 ตำแหน่งหน้าสัมผัสบนสมาร์ตการ์ด	11
2.9 รูปสัญญาณการเชื่อมต่อแบบอะซิงโครนัส	13
2.10 รูปสัญญาณการเชื่อมต่อแบบซิงโครนัส	13
2.11 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรม	14
2.12 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม	15
2.13 คอนเนคเตอร์อนุกรม 9 ขา หรือแบบ DB-9	17
2.14 คอนเนคเตอร์อนุกรม 25 ขา หรือแบบ DB-25	17
2.15 การแทนค่าเลขฐานสองของแถบต่าง ๆ	22
2.16 สัญญาณที่อ่านได้จากบาร์โค้ด	23
2.17 โครงสร้างของรหัสบาร์โค้ด 39	25
2.18 การจัดตำแหน่งขาต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51	28
2.19 หน้าที่ของพอร์ตเมื่อไมโครคอนโทรลเลอร์ทำงานกับหน่วยความจำภายนอก	29
2.20 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ AT89C52	29
2.21 รูปตัวอย่าง Entity และ Attribute	33
2.22 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม โดยใช้หัวลูกศรคู่แสดงความเป็นกลุ่ม	33
2.23 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม โดยใช้หัวลูกศรเดียวแสดงความเป็นกลุ่ม	34
2.24 การจัดระบบไฟล์	35
2.25 การใช้ระบบฐานข้อมูล	36
2.26 การใช้ระบบฐานข้อมูลด้วย DBMS	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.27 โครงสร้างฐานข้อมูลแบบเชิงสัมพันธ์	37
2.28 โครงสร้างฐานข้อมูลแบบเน็ตเวิร์ค	38
2.29 โครงสร้างของโมเดลแบบแตกสาขา	39
2.30 ระดับของข้อมูล 3 ระดับ	41
3.1 แผนผังการทำงานเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	43
3.2 การต่อวงจรเครื่องอ่านบาร์โค้ด	44
3.3 วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด	45
3.4 วงจรคีย์เมตริกซ์สวิตช์	46
3.5 วงจรแสดงผลจอแอลซีดี	46
3.6 วงจรภาคจ่ายไฟเลี้ยง	47
3.7 โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด WACR20S	48
3.8 วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	50
4.1 ตัวเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	52
4.2 หน้าจอแอลซีดีรอการรูดบัตรประจำตัวนักศึกษา	53
4.3 หน้าจอแอลซีดีเมื่อต้องการวันเดือนปี	54
4.4 หน้าจอแอลซีดีเมื่อต้องการตั้งเวลา	54
4.5 หน้าจอแอลซีดีเมื่อมีการรูดบัตรประจำตัวนักศึกษา	55
4.6 หน้าจอแอลซีดีให้เปลี่ยนสมาร์ตการ์ดที่มีข้อมูลเต็ม	56
4.7 แสดงการเสียบบัตรสมาร์ตการ์ดเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด	56
4.8 หน้าต่างโปรแกรมการป้อนชื่อผู้ใช้และรหัสผ่าน	57
4.9 หน้าต่างโปรแกรมให้เลือกพอร์ตการติดต่อและรายวิชาที่จะโหลดข้อมูล	58
4.10 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลนักศึกษา	58
4.11 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลทั่วไป	59
4.12 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลการลงวิชาเรียน	59
4.13 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลนักศึกษา	60
4.14 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลทั่วไป	61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.15 หน้าต่างโปรแกรมเพื่อค้นหาข้อมูลการเข้าชั้นเรียน	61
4.16 ข้อมูลที่ได้จากเครื่องพิมพ์แสดงการเข้าเรียน เข้าสาย และขาดเรียน	62
ก.1 ภาพด้านหน้าของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	68
ก.2 ภาพด้านหลังของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	68
ก.3 ภาพด้านหน้าของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด WACR20S	69
ก.4 ภาพด้านข้างของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด WACR20S	69
ก.5 การเชื่อมต่อโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดกับเครื่องคอมพิวเตอร์	70
ก.6 บัตรสมาร์ตการ์ด SLE 4428	70
ข.1 วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	72
ข.2 แผ่นวงจรพิมพ์ของวงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	73
ข.3 วงจรวงจรคีย์เมตริกซ์สวิตช์	73
ข.4 แผ่นวงจรพิมพ์ของวงจรคีย์เมตริกซ์สวิตช์	74
ข.5 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	74
ข.6 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรคีย์เมตริกซ์สวิตช์	75
ง.1 แผนผังเริ่มต้นการทำงานของโปรแกรมและการทำงานของโปรแกรมย่อย	80
ง.2 แผนผังการทำงานการกำหนดค่าเริ่มต้นของโปรแกรมหลัก	81
ง.3 แผนผังการทำงานของโปรแกรมตรวจสอบและบันทึกข้อมูลลงสมาร์ตการ์ด	82
ง.4 แผนผังการทำงานของโปรแกรมตรวจสอบการเสียบบัตรและถอดบัตรสมาร์ตการ์ด	83
ง.5 แผนผังการทำงานของโปรแกรมเลือกการกำหนดค่าของวันเดือนปีและเวลา	84
ง.6 แผนผังการทำงานของโปรแกรมกำหนดค่าของวันเดือนปีและเวลา	85
ง.7 แผนผังการทำงานโปรแกรมแสดงผลวันเดือนปีและเวลาบนจอแอลซีดี	86
ง.8 แผนผังการทำงานของโปรแกรมเขียนค่าของวันเดือนปีและเวลาลงสมาร์ตการ์ด	87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
ง.9 แผนผังการทำงานของโปรแกรมตรวจสอบข้อมูลในสมาร์ตการ์ดและเปลี่ยนหน้าของการเขียนข้อมูลในบัตรสมาร์ตการ์ด	88
ง.10 แผนผังการทำงานของโปรแกรมแสดงรหัสนักศึกษาบนจอแอลซีดีและบันทึกรหัสนักศึกษาลงบัตรสมาร์ตการ์ด	89
ง.11 แผนผังการทำงานของโปรแกรมการอินเตอร์รัพต์สมาร์ตการ์ดและเครื่องอ่านบาร์โค้ด	90
ง.12 แผนผังการทำงานของโปรแกรมอ่านค่าของวันเดือนปีและเวลาจาก Real Time Clock	91
ง.13 แผนผังการทำงานของโปรแกรมเขียนค่าของวันเดือนปีและเวลาลง Real Time Clock	92
ง.14 แผนผังการทำงานของโปรแกรมเขียนค่าของเวลาและรหัสนักศึกษาลงสมาร์ตการ์ด	93
ง.15 หน้าต่างโปรแกรมให้ป้อนชื่อผู้ใช้และรหัสผ่าน	119
ง.16 โปรแกรมอ่าน-เขียนสมาร์ตการ์ด	119
จ.1 ส่วนประกอบของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด	145
จ.2 แอลซีดีสีเขียวติดขณะเสียบบัตรสมาร์ตการ์ด	147
จ.3 หน้าจอแอลซีดีเมื่อต้องการตั้งค่าวันเดือนปี	147
จ.4 หน้าจอแอลซีดีเมื่อต้องการตั้งค่าเวลา	148
จ.5 หน้าจอแอลซีดีเมื่อเครื่องพร้อมรับข้อมูลจากเครื่องอ่านบาร์โค้ด	148
จ.6 หน้าจอแอลซีดีแสดงผลการรับข้อมูลจากเครื่องอ่านบาร์โค้ด	149
จ.7 หน้าจอแอลซีดีแสดงผลว่าสมาร์ตการ์ดเก็บข้อมูลเต็มและให้เปลี่ยนบัตร	150
จ.8 แสดงการเสียบบัตรสมาร์ตการ์ดเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด	151
จ.9 หน้าต่างโปรแกรมการอ่านข้อมูลจากสมาร์ตการ์ด	151
จ.10 หน้าต่างโปรแกรมให้ป้อนชื่อผู้ใช้งานและรหัสผ่าน	152
จ.11 หน้าต่างโปรแกรมการใช้งานหลัก	152
จ.12 หน้าต่างโปรแกรมการป้อนชื่อผู้ใช้และรหัสผ่าน	153
จ.13 หน้าต่างโปรแกรมให้เลือกพอร์ตการติดต่อและรายวิชาที่จะโหลดข้อมูลเก็บ	154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
จ.14 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลนักศึกษา	155
จ.15 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลทั่วไป	156
จ.16 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลการลงวิชาเรียน	156
จ.17 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลนักศึกษา	157
จ.18 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลทั่วไป	158
จ.19 หน้าต่างโปรแกรมเพื่อค้นหาข้อมูลการเข้าชั้นเรียน	158

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในการวัดและประเมินผลการศึกษาของนักเรียนนักศึกษา การวัดพฤติกรรมด้านจิตพิสัยถือเป็นพฤติกรรมด้านหนึ่งที่อาจารย์ผู้สอนได้นำมาคิดรวมกับการประเมินผลพฤติกรรมด้านอื่นๆ เพื่อนำมาใช้ในการวัดและประเมินผลการศึกษาของนักศึกษา การเข้าชั้นเรียนแต่ละครั้งเป็นตัวแสดงถึงพฤติกรรมด้านจิตพิสัยของนักศึกษาได้ ดังนั้นในการเข้าชั้นเรียนจึงมีความจำเป็นต้องมีการตรวจสอบการเข้าชั้นเรียนของนักศึกษา แต่ในการเข้าชั้นเรียนทุกครั้งทำให้เกิดปัญหากับอาจารย์ผู้สอนคือ อาจารย์ต้องมาตรวจสอบเรียกชื่อนักศึกษาทำให้เสียเวลาในการสอนบ้าง โดยเฉพาะในกรณีที่มีจำนวนนักศึกษามากๆ

แนวการแก้ไขปัญหาใหม่คือ ใช้วิธีการตรวจสอบการเข้าชั้นเรียนซึ่งใช้เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด เป็นเครื่องบันทึกวันและเวลาการเข้าชั้นเรียนของนักศึกษาแต่ละคน โดยให้นักศึกษารูดบัตรประจำตัวนักศึกษาที่เครื่องอ่านบาร์โค้ดทุกครั้งในการเข้าชั้นเรียน เมื่อมีการอ่านรหัสจากบัตรนักศึกษาแล้วเครื่องจะบันทึกรหัสประจำตัวนักศึกษาวันที่และเวลาเข้าชั้นเรียนของนักศึกษาแต่ละคนเก็บลงในสมาร์ตการ์ดของอาจารย์ เนื่องจากสมาร์ตการ์ดนั้นมีหน่วยความจำมากและมีความปลอดภัยสูงในการจัดเก็บข้อมูล เมื่ออาจารย์ต้องการตรวจสอบข้อมูลในสมาร์ตการ์ดที่เก็บไว้สามารถทำได้โดยการเสียบบัตรเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดเพื่อติดต่อกับฐานข้อมูลในเครื่องคอมพิวเตอร์

1.2 ขีดความสามารถของโครงงาน

โครงงานนี้มีขีดความสามารถดังนี้

1. ใช้รองรับจำนวนนักศึกษาได้ไม่เกิน 50 คน
2. สามารถบันทึกวันเดือนปีและเวลาของนักศึกษาแต่ละคนลงในสมาร์ตการ์ดได้
3. สามารถแก้ไขข้อมูลที่บันทึกไว้ในสมาร์ตการ์ดผ่านทางคอมพิวเตอร์ได้
4. สามารถพกพาได้สะดวกและมีขนาดเล็กกระทัดรัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้แบ่งออกเป็นบทต่างๆ เพื่อสะดวกต่อการศึกษาและทำความเข้าใจ ในแต่ละบทจะประกอบด้วยเนื้อหาดังต่อไปนี้

บทที่ 1 กล่าวถึงความเป็นมาและความสำคัญของปริญญานิพนธ์ ขีดความสามารถของโครงงานและเนื้อหาในบทต่างๆ โดยสังเขป

บทที่ 2 ทฤษฎีและหลักการ กล่าวถึงทฤษฎีต่างๆ เกี่ยวกับความรู้ทั่วไปของสมาร์ตการ์ด ชนิดของสมาร์ตการ์ด มาตรฐาน ISO-7816 การสื่อสารข้อมูลแบบอนุกรม หลักการของบาร์โค้ด การติดต่อสื่อสารระบบ I^2C บัส ทฤษฎีและหลักการของสถาปัตยกรรมโครงสร้างของไมโครคอนโทรลเลอร์ MCS-51 และระบบฐานข้อมูล

บทที่ 3 กล่าวถึงเนื้อหาที่เกี่ยวกับ แผนผังการทำงานของโครงงาน ผังวงจรต่างๆ ที่ใช้ในโครงงาน ตลอดจนการออกแบบและการสร้างส่วนประกอบต่างๆ เช่น วงจรที่ใช้ติดต่อกับเครื่องอ่านบาร์โค้ด วงจรที่ใช้เขียนรหัสประจำตัวนักศึกษาวันที่และเวลาลงในสมาร์ตการ์ด ตัวเครื่องโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดที่ใช้ติดต่อกับฐานข้อมูล โครงสร้างของชิ้นงานต่างๆ พร้อมทั้งการทำงานของส่วนประกอบต่างๆ โดยละเอียด

บทที่ 4 ประกอบด้วย การทดลองและผลการทดลองของวงจรที่ใช้ติดต่อกับเครื่องอ่านบาร์โค้ด วงจรที่ใช้เขียนรหัสประจำตัวนักศึกษาวันที่และเวลาลงในสมาร์ตการ์ด ตัวเครื่องโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดที่ใช้ติดต่อกับฐานข้อมูล

บทที่ 5 เป็นการสรุปผลการจัดทำโครงงาน ปัญหาที่เกิดขึ้นและแนวทางในการแก้ไข รวมทั้งแนวทางการพัฒนา

ภาคผนวก ก เครื่องต้นแบบ

ภาคผนวก ข รายละเอียดวงจร และแผ่นวงจรพิมพ์

ภาคผนวก ค รายการอุปกรณ์

ภาคผนวก ง แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม

ภาคผนวก จ คู่มือการใช้งาน

ภาคผนวก ฉ รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 กล่าวนำ

เนื้อหาของปริญญานิพนธ์ในบทนี้เป็นทฤษฎีและหลักการที่นำมาใช้ประกอบการสร้างโครงงาน โดยประกอบไปด้วยเรื่องสมาร์ตการ์ด ชนิดของสมาร์ตการ์ด มาตรฐาน ISO-7816 การสื่อสารข้อมูลแบบอนุกรม หลักการของบาร์โค้ด การติดต่อสื่อสารระบบ I^2C บัส คุณสมบัติไมโครคอนโทรลเลอร์ และระบบฐานข้อมูล ซึ่งจะกล่าวรายละเอียดดังต่อไปนี้

2.2 สมาร์ตการ์ด

สมาร์ตการ์ด คือ บัตรพลาสติกขนาดเล็กที่มีขนาดเทียบเท่ากับบัตรเครดิตภายในบัตรมีการบรรจุชิปหรือหน่วยประมวลผลทางคอมพิวเตอร์ไว้ ชิปที่มีการบรรจุไว้อาจเป็นได้ทั้งไมโครโปรเซสเซอร์ที่มีหน่วยความจำติดตั้งอยู่รวมกันหรืออาจเป็นชิปหน่วยความจำที่ยังไม่มีการบรรจุโปรแกรมการทำงานใดๆ ไว้ การติดต่อสื่อสารระหว่างชิปและอุปกรณ์ภายนอกสามารถกระทำได้ทั้งแบบหน้าสัมผัสกันโดยตรงหรือโดยการเหนี่ยวนำผ่านตัวกลางที่เป็นสนามแม่เหล็ก

2.2.1 การแบ่งชนิดของสมาร์ตการ์ด

สมาร์ตการ์ดแบ่งได้ตามชนิดของชิปคอมพิวเตอร์และความจุที่ผนึกอยู่ภายในซึ่งแบ่งได้เป็น 3 ชนิด คือ

1) สมาร์ตการ์ดแบบมีหน้าสัมผัส

สมาร์ตการ์ดแบบมีหน้าสัมผัสมีลักษณะการใช้งานคือ ต้องสอดบัตรเข้าไปในเครื่องอ่าน – เขียนสมาร์ตการ์ดเพื่อให้หน้าสัมผัสสัมผัสกับเครื่องอ่าน-เขียนสมาร์ตการ์ด สมาร์ตการ์ดประเภทนี้จะมีการผนึกชิปทองขนาดเส้นผ่านศูนย์กลางประมาณครึ่งนิ้วเอาไว้ที่ด้านหน้าบัตรเมื่อผู้ใช้สอดบัตรเข้าไปในเครื่องอ่าน-เขียนสมาร์ตการ์ดจะทำให้หน้าสัมผัสสัมผัสกับหัวต่อหรือคอนเนคเตอร์ทางไฟฟ้าซึ่งจะทำการถ่ายทอดข้อมูลเข้าและออกจากชิปสมาร์ตการ์ดแบบมีหน้าสัมผัสแสดงดังรูปที่ 2.1 โดยสมาร์ตการ์ดแบบมีหน้าสัมผัสนี้สามารถแบ่งออกได้อีก 2 ชนิด คือ

1.1) สมาร์ตการ์ดประเภทที่มีหน่วยความจำเพียงอย่างเดียว จะไม่มีหน่วยประมวลผลกลางควบคุมอยู่ภายในไม่สามารถประมวลผลข้อมูลและไม่สามารถจัดการไฟล์ข้อมูลได้มีไว้สำหรับเก็บข้อมูลต่างๆ โดยทั่วไปมักจะนำไปประยุกต์ใช้งานที่ไม่ต้องการการประมวลผลจากบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรืองานที่ต้องการเก็บข้อมูลเพียงอย่างเดียว สมาร์ตการ์ดประเภทที่มีหน่วยความจำเพียงอย่างเดียวแบบนี้แสดงดังรูปที่ 2.2 สมาร์ตการ์ดแบบนี้ยังแบ่งออกได้อีก 3 ชนิด คือ

1.1.1) สมาร์ตการ์ดที่ใช้เก็บข้อมูลได้อย่างเดียว (Straight Memory Cards) ไม่สามารถประมวลผลข้อมูลได้และมีราคาต่อบิตของหน่วยความจำที่ใช้งานต่ำที่สุดซึ่งคล้ายกับแผ่นดิสก์ที่มีให้เลือกใช้อยู่ด้วยกันหลายๆ ขนาดโดยที่ไม่มีกลไกในการประมวลผล ซึ่งเครื่องอ่าน-เขียนสมาร์ตการ์ดจะไม่สามารถตรวจสอบได้ว่าเป็นการ์ดชนิดที่เก็บข้อมูลได้เพียงอย่างเดียว

1.1.2) สมาร์ตการ์ดที่มีความสามารถป้องกันการอ่านและการเขียนข้อมูลได้ (Protected/Segmented Memory Cards) จะมีระบบควบคุมภายในใช้ในการติดต่อกับหน่วยความจำภายในการ์ด บางครั้งอาจเรียกการ์ดชนิดนี้ว่า Intelligent Memory Cards สมาร์ตการ์ดชนิดนี้สามารถป้องกันพื้นที่ในหน่วยความจำทั้งหมดหรือบางส่วนก็ได้ สามารถจำกัดการเข้าถึงข้อมูลได้ทั้งการอ่านและเขียนโดยการใช้รหัสผ่าน และสามารถแบ่งพื้นที่ต่างๆ ของหน่วยความจำเพื่อออกแบบให้ทำหน้าที่หลายๆ อย่างได้อีกด้วย

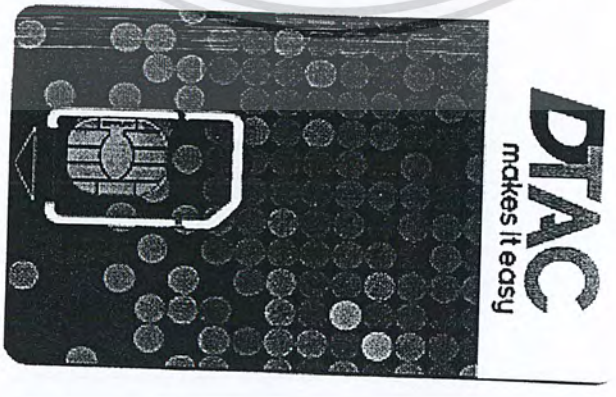
1.1.3) สมาร์ตการ์ดชนิดที่ถูกออกแบบมา เพื่อใช้ในการเก็บข้อมูลและนำเอาข้อมูลออกมาใช้งานโดยเฉพาะ (Stored Value Memory Cards) ระดับความปลอดภัยของสมาร์ตการ์ดจะขึ้นอยู่กับรหัสผ่านและลอจิกภายในชิปที่สร้างขึ้น โดยบริษัทผู้ผลิตการ์ดแต่ละบริษัท เมื่อเกิดการใช้งานพื้นที่ของหน่วยความจำจะลดลงหรือเกิดการนับขึ้น โดยมีหน่วยความจำของการ์ดเพียงเล็กน้อยสำหรับใช้งาน ตัวอย่างเช่น ชิปในบัตรโทรศัพท์จะมีความจำอยู่ 60 หรือ 12 เซลล์ ซึ่งแต่ละหน่วยความจำจะถูกลบออกไป 1 เซลล์ เมื่อหน่วยความจำทุกเซลล์ถูกลบไปหมดการ์ดนั้นก็จะไม่สามารถใช้งานได้อีกด้วย

รูปที่ 2.1 สมาร์ตการ์ดแบบมีหน้าสัมผัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.2 สมาร์ตการ์ดแบบที่มีหน่วยความจำเพียงอย่างเดียว

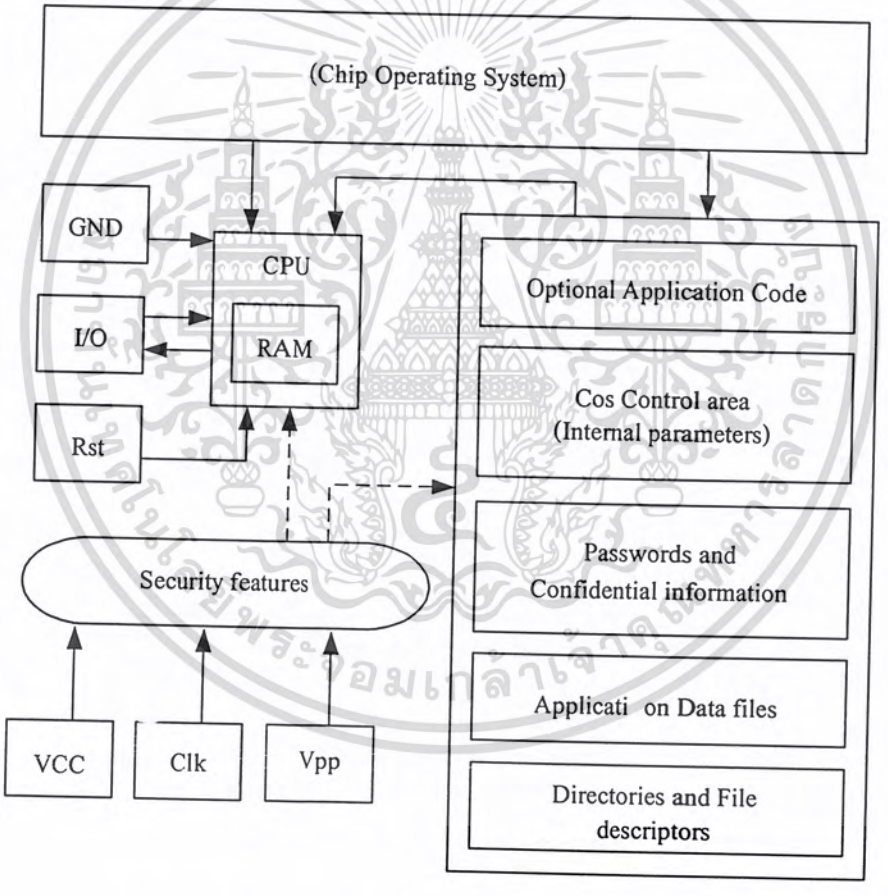
1.2) สมาร์ตการ์ดที่มีหน่วยประมวลผลกลางอยู่ในตัว จะมีความสามารถในการจัดการข้อมูลและสามารถจองหน่วยความจำบางส่วนเพื่อประยุกต์ใช้งานได้ สมาร์ตการ์ดชนิดนี้จะประกอบไปด้วยไมโครโปรเซสเซอร์และหน่วยความจำแบบแฟลช ผนึกอยู่ภายในซึ่งจะสามารถกำหนดหน่วยความจำและการเข้าถึงไฟล์ข้อมูลได้ สมาร์ตการ์ดชนิดนี้จะเหมือนกับคอมพิวเตอร์ขนาดย่อมที่จัดการกับข้อมูลและโครงสร้างภายใน โดยผ่านระบบปฏิบัติการของการ์ด ซึ่งสมาร์ตการ์ดแบบนี้จะมีความปลอดภัยของข้อมูลค่อนข้างสูง เนื่องจากส่วนของหน่วยความจำในการ์ดนั้นจะไม่มีการติดต่อกับขาภายนอกของสมาร์ตการ์ดโดยตรง ซึ่งการติดต่อกับหน่วยความจำนั้นจะต้องติดต่อทางหน่วยประมวลผลกลางหรือซีพียูของบัตรเท่านั้น รูปที่ 2.3 แสดงสมาร์ตการ์ดแบบที่มีหน่วยประมวลผลกลางอยู่ในตัว



รูปที่ 2.3 สมาร์ตการ์ดแบบที่มีหน่วยประมวลผลกลางอยู่ในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 เป็นตัวอย่างโครงสร้างภายในชิปแบบมีหน่วยประมวลผลอยู่ในตัว ซึ่งได้รับการออกแบบให้มีความสมบูรณ์ของการทำงานอยู่ภายในตัวชิป จากรูปจะเห็นว่ามีการเขียนระบบปฏิบัติการลงในหน่วยความจำรอม เพื่อใช้ควบคุมการทำงานของหน่วยประมวลผลกลางที่มีการติดตั้งหน่วยความจำแบบแรมไว้ภายใน สำหรับข้อมูลการประยุกต์ใช้งานต่างๆ ที่เกี่ยวกับสมาร์ตการ์ด สมาร์ตการ์ดแต่ละใบนั้นจะได้รับการเขียนลงบนหน่วยความจำอีพรอมหรืออีอีพรอม โดยการแก้ไขปรับเปลี่ยนผ่านทางจุดเชื่อมต่อทางหน้าสัมผัสบนสมาร์ตการ์ดหรือโดยการเหนี่ยวนำทางคลื่นแม่เหล็กไฟฟ้า ภายในตัวชิปจะมีวงจรควบคุมแรงดันไฟเลี้ยงซึ่งจะได้รับเมื่อมีการเชื่อมต่อสมาร์ตการ์ดกับเครื่องอ่าน-เขียนสมาร์ตการ์ดไม่ว่าจะโดยการสัมผัสหรือการเหนี่ยวนำ

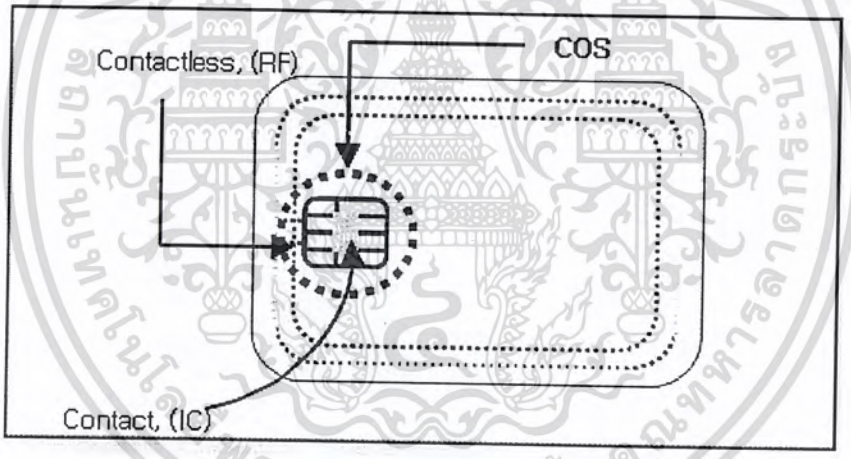


รูปที่ 2.4 โครงสร้างภายในชิปแบบมีหน่วยประมวลอยู่ในตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สมาร์ตการ์ดแบบไม่มีหน้าสัมผัส

เป็นสมาร์ตการ์ดที่สามารถอ่าน-เขียนข้อมูลได้ โดยไม่จำเป็นต้องมีการสัมผัสโดยตรงกับเครื่องอ่าน-เขียนสมาร์ตการ์ด เหมาะสำหรับการใช้ที่ไม่ต้องการให้สมาร์ตการ์ดเกิดรอยขูดขีดเนื่องจากการสอดเข้าเครื่องอ่าน-เขียนสมาร์ตการ์ดบ่อยๆ การใช้งานสมาร์ตการ์ดชนิดนี้เพียงแค่นำไปวางให้อยู่ใกล้ๆ กับบริเวณสายอากาศ ของเครื่องอ่าน-เขียนสมาร์ตการ์ด โดยไม่ต้องมีการสัมผัสใดๆ กับเครื่องอ่าน-เขียนสมาร์ตการ์ด ถ้ามองจากภายนอกแล้วตัวสมาร์ตการ์ดชนิดนี้จะมีลักษณะคล้ายกับบัตรพลาสติกแบบหนึ่ง แต่ด้วยโครงสร้างภายในที่มีการผนึกชิปและขดลวดสายอากาศไว้ ทำให้การสื่อสารระหว่างการ์ดกับเครื่องอ่าน-เขียนสมาร์ตการ์ดทำได้จากระยะไกล ทำให้สมาร์ตการ์ดชนิดนี้นิยมใช้กับงานที่ต้องการความรวดเร็วเป็นสำคัญ เช่น ใช้กับระบบเก็บเงินค่าผ่านทางด่วน เครื่องเก็บค่าโดยสารบนรถประจำทาง เป็นต้น ลักษณะของบัตรสมาร์ตการ์ดแบบไม่มีหน้าสัมผัสแสดงดังรูปที่ 2.5



รูปที่ 2.5 สมาร์ตการ์ดแบบไม่มีหน้าสัมผัส

3) สมาร์ตการ์ดแบบผสม

สมาร์ตการ์ดแบบนี้จะเป็นสมาร์ตการ์ดใบเดียวแต่ทำหน้าที่เป็นทั้งสมาร์ตการ์ดแบบมีหน้าสัมผัสและสมาร์ตการ์ดแบบไม่มีหน้าสัมผัส โดยมีการรวมชิป 2 ชุดไว้เป็นชิปชุดเดียวกันทำให้เพิ่มความสะดวกสำหรับผู้ใช้งาน เนื่องจากสามารถติดต่ออ่านและเขียนข้อมูลได้ เพราะมีตัวเลือกทั้งแบบการสื่อสารแบบสัมผัสและแบบไม่มีการสัมผัส โดยโครงสร้างภายนอกของสมาร์ตการ์ดแบบผสมจะมีการติดตั้งทั้งเสาอากาศและหน้าสัมผัสทองไว้สำหรับผู้ใช้งานเลือกรูปแบบในการติดต่อสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 ข้อดีของสมาร์ตการ์ด

1) มีความปลอดภัยของข้อมูลสูง

2) ตัวการ์ดมีหน่วยประมวลผลอยู่ภายใน (เฉพาะการ์ดแบบไมโครโปรเซสเซอร์) ทำให้สามารถประมวลผลข้อมูลได้ที่ตัวการ์ดเอง

3) มีความจุของข้อมูลที่สามารถเก็บในการ์ดสูงกว่าการ์ดชนิดอื่น

4) ทนทานต่อสนามแม่เหล็ก

5) มีชนิดของการ์ดและการเชื่อมต่อให้เลือกหลายแบบทำให้สามารถนำไปประยุกต์ใช้งานในด้านต่างๆ ได้

2.2.3 การประยุกต์ใช้งานสมาร์ตการ์ด

ปัจจุบันความต้องการใช้งานสมาร์ตการ์ดในกลุ่มธุรกิจประเภทต่างๆ มีเพิ่มมากขึ้นส่วนหนึ่งมาจากต้นทุนต่อหน่วยของสมาร์ตการ์ดที่ลดลงตามกลไกทางการตลาดและแรงผลักดันของการใช้ชีวิตอยู่ร่วมกับเทคโนโลยีของมนุษย์ ทำให้สามารถแบ่งกลุ่มประเภทธุรกิจที่นำสมาร์ตการ์ดเข้ามาใช้งานได้เป็นดังนี้

1) ธุรกิจสื่อสารไร้สาย

2) ธุรกิจรักษาความปลอดภัย

3) สถาบันการศึกษา

4) ธุรกิจให้บริการที่จอดรถ

5) ธุรกิจขนส่งมวลชน

6) ส่วนราชการและสาธารณะสุข

7) ธุรกิจโทรศัพท์สาธารณะ

8) ธุรกิจการเงินและการธนาคาร

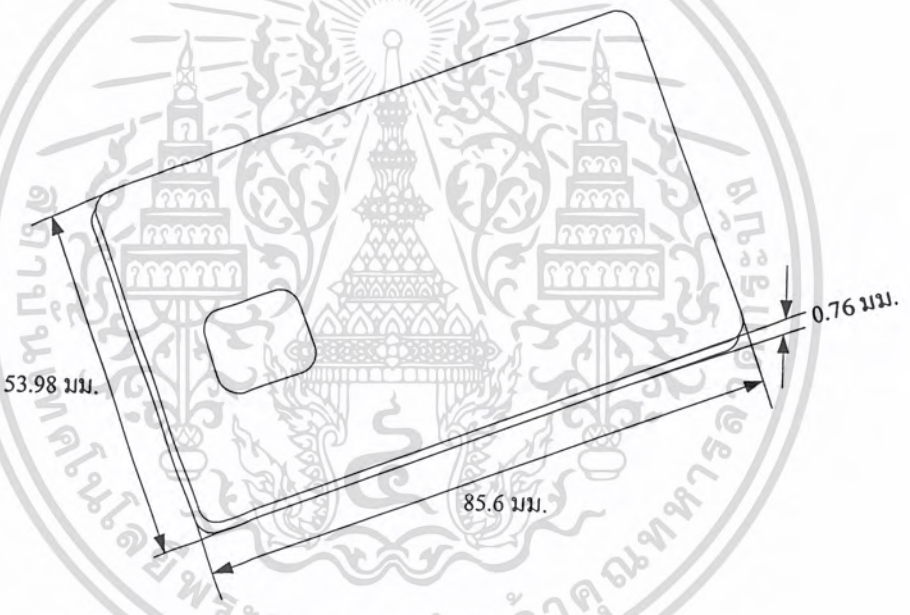
2.2.4 มาตรฐาน ISO7816

มาตรฐาน ISO7816 เป็นมาตรฐานของสมาร์ตการ์ดแบบมีหน้าสัมผัส ที่กำหนดขึ้นโดยองค์กรมาตรฐานสากล ISO (International Standards Organization) ที่อธิบายลักษณะข้อมูลจำเพาะของสมาร์ตการ์ดซึ่งมาตรฐานนี้ถูกแบ่งออกเป็น 6 ส่วนคือ ISO7816-1 ถึง ISO7816-6 ซึ่งแต่ละส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะอธิบายถึงข้อมูลต่างๆ ของสมาร์ตการ์ดโดยส่วนสำคัญพื้นฐานคือ ISO7816-1 ถึง ISO7816-3 มาตรฐาน ISO7816 ทั้งหมดมีรายละเอียดดังต่อไปนี้

1) ISO7816-1 อธิบายถึงลักษณะของสมาร์ตการ์ด ขนาด รูปร่างของบัตร และข้อจำกัดทางโครงสร้างของตัวการ์ด โดยลักษณะทางกายภาพของสมาร์ตการ์ดกำหนดให้มีความกว้างของบัตรเท่ากับ 85.6 มิลลิเมตร (3.370 นิ้ว) มีความยาวเท่ากับ 53.98 มิลลิเมตร (2.125 นิ้ว) และมีความหนาเท่ากับ 0.76 มิลลิเมตร (0.30 นิ้ว) ดังรูปที่ 2.6 เป็นขนาดความกว้าง ความยาว และความหนาของสมาร์ตการ์ด ตามมาตรฐาน ISO7816-1 และรูปที่ 2.7 เป็นตำแหน่งหน้าสัมผัสที่ผิวหน้าของสมาร์ตการ์ดตามมาตรฐาน ISO7816-1

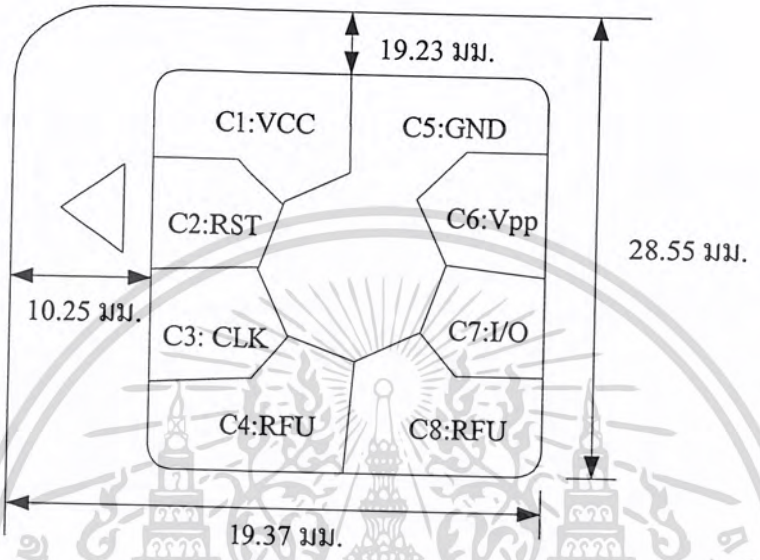


รูปที่ 2.6 ขนาดความกว้าง ความยาวและความหนาของสมาร์ตการ์ด

2) ISO7816-2 อธิบายถึงลักษณะและตำแหน่งของหน้าสัมผัสของสมาร์ตการ์ด จากรูปที่ 2.7 กำหนดให้ตำแหน่งหน้าสัมผัส C1 : VCC = 5V, C2 : Reset, C3 : Clock, C4 : RFU, C5 : GND, C6 : Vpp, C7 : I/O, C8 : RFU โดยตำแหน่งขาของสมาร์ตการ์ดเป็นตำแหน่งของขาสัมผัสทั้ง 8 ขา ซึ่งเป็นจุดที่เครื่องอ่าน-เขียนสมาร์ตการ์ดใช้เชื่อมต่อกับวงจรภายในการ์ด ขาที่ใช้งานหลักๆ จะมีอยู่เพียง 6 ขา ขา RFU (Reserved For Future User) จะไม่ได้ใช้งาน ดังรูปที่ 2.7 แสดงถึงตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าสัมผัสบนสมาร์ตการ์ดและตารางที่ 2.1 แสดงขนาดของหน้าสัมผัสบนสมาร์ตการ์ด C1-C8 ตามมาตรฐาน ISO7816-2



รูปที่ 2.7 ตำแหน่งหน้าสัมผัสที่ผิวหน้าของสมาร์ตการ์ดตามมาตรฐาน ISO7816-1

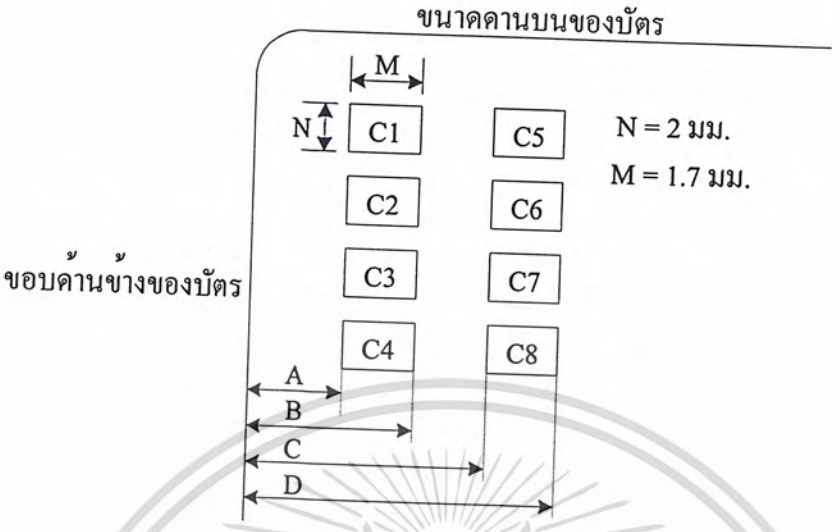
การเชื่อมต่อกับสมาร์ตการ์ด ทำได้โดยการเชื่อมต่อเข้ากับขาสัมผัส ซึ่งมีอยู่ทั้งหมด 8 ขา โดยแต่ละขามีหน้าที่และประโยชน์ใช้งานดังนี้

1) ขา I/O ทำงานเป็นอินพุตหรือเอาต์พุตเพื่อรับ-ส่งข้อมูลอนุกรมด้วยรูปแบบ ผลัดกันส่งข้อมูลบนสายนำสัญญาณเดียวโดยที่ขา I/O นี้จะมีสถานะที่เป็นได้อยู่ 2 สถานะด้วยกัน คือ สถานะลอจิกสูงเมื่อการ์ดรับหรือส่งบิตข้อมูลที่เป็นลอจิก "1" สถานะลอจิกต่ำเมื่อการ์ดรับหรือส่งบิตข้อมูลที่เป็นลอจิก "0"

2) ขา VPP ใช้ป้อนแรงดันค่าสูงเพื่อเขียนหรือลบข้อมูลในหน่วยความจำ Non-Volatile จะมีสถานะเป็นได้ 2 สถานะด้วยกัน คือ สถานะว่างและสถานะทำงาน

3) ขา CLK ใช้สำหรับรับสัญญาณนาฬิกาที่ส่งมาจากเครื่องอ่าน-เขียนสมาร์ตการ์ด เพื่อใช้เป็นสัญญาณนาฬิกาให้กับระบบไมโครโปรเซสเซอร์ของการ์ด (โดยทั่วไปจะใช้ความถี่ค่า 3.57 เมกะเฮิรตซ์) หรือเพื่อเป็นสัญญาณนาฬิกาอ้างอิง ในการเข้าถึงข้อมูลสำหรับการเชื่อมต่อกับสมาร์ตการ์ดที่มีหน่วยความจำเพียงอย่างเดียว (หมายถึงการที่ขา I/O โดยเครื่องอ่าน-เขียนสมาร์ตการ์ดจะต้องสร้างสัญญาณนาฬิกาขึ้นมา 1 ลูก เมื่อต้องการรับหรือส่งข้อมูล 1 บิต)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ตำแหน่งหน้าสัมผัสบนสมาร์ตการ์ด

4) ขา RST ใช้ในการรีเซตสมาร์ตการ์ดให้เริ่มต้นทำงานใหม่

5) ขา VCC และขาGND รับไฟเลี้ยงและกราวด์ที่ป้อนจากเครื่องอ่าน-เขียนสมาร์ตการ์ด

3) ISO7816-3 อธิบายลักษณะคุณสมบัติทางไฟฟ้าของสัญญาณ Vcc, Vpp, RST, I/O และ CLK ขั้นตอนและไทม์มิ่งไดอะแกรมสำหรับการเชื่อมต่อกับสมาร์ตการ์ดและโปรโตคอลที่ใช้ในการสื่อสารกับชิปสมาร์ตการ์ด

4) ISO7816-4 อธิบายโครงสร้างข้อมูลที่บรรจุอยู่ในสมาร์ตการ์ด ชุดคำสั่งที่ใช้ในสมาร์ตการ์ด (Application Protocol Data Unit: APDU) และรูปแบบการเข้ารหัสข้อมูล

5) ISO7816-5 อธิบายระบบลงทะเบียนหมายเลขสำหรับการค้นหา การประยุกต์ใช้งานซึ่งประกอบด้วยข้อมูลแบบไบต์ โดยที่ 5 ไบต์แรกระบุถึงผู้สร้างการประยุกต์ใช้งานและ 11 ไบต์หลังระบุชนิดของการประยุกต์ใช้งาน กำหนดรายละเอียดโครงสร้างไฟล์ของสมาร์ตการ์ด

6) ISO7816-6 อธิบายถึงรูปแบบการสื่อสารข้อมูลของอุปกรณ์ รูปแบบค่า Answer To Reset (ATR) และรูปแบบโปรโตคอลการส่งข้อมูล (T=0, T=1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ขนาดของตำแหน่งหน้าสัมผัสบนสมาร์ตการ์ด C1-C8

ตำแหน่งขา	ชื่อเรียก	A(mm)	B(mm)	C(mm)	D(mm)
C1	VCC	10.25	12.25	19.23	20.93
C2	RST	10.25	12.25	21.77	23.47
C3	CLK	10.25	12.25	24.31	26.01
C4	RFU	10.25	12.25	26.85	28.55
C5	GND	17.87	19.87	19.23	20.93
C6	Vpp	17.87	19.87	21.77	23.47
C7	I/O	17.87	19.87	24.31	26.01
C8	RFU	17.87	19.87	28.85	28.85

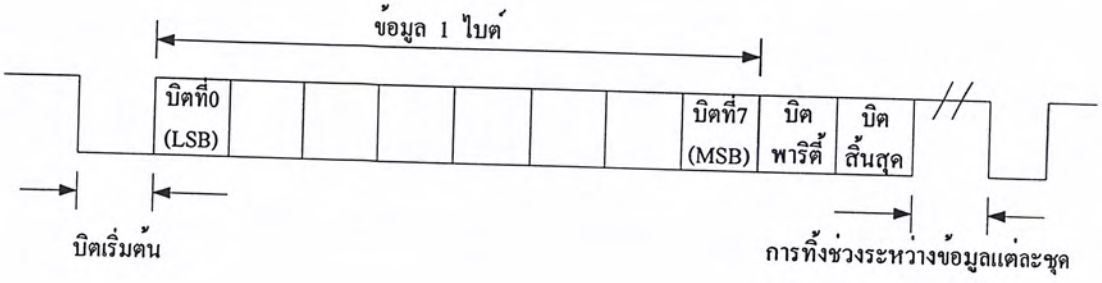
2.2.5 รูปแบบการเชื่อมต่อของสมาร์ตการ์ดแบบมีหน้าสัมผัส

สมาร์ตการ์ดแบบมีหน้าสัมผัสโดยหลักแล้วมีอยู่ 2 ชนิด คือ แบบมีหน่วยความจำเพียงอย่างเดียวและแบบมีหน่วยประมวลผลกลางภายในตัว โดยสมาร์ตการ์ดทั้ง 2 ชนิดจะมีการเชื่อมต่อแตกต่างกัน คือ แบบมีหน่วยประมวลผลกลางภายในตัวจะใช้การเชื่อมต่อแบบอะซิงโครนัส ส่วนแบบมีหน่วยความจำเพียงอย่างเดียวจะใช้การเชื่อมต่อแบบซิงโครนัส

1) การเชื่อมต่อแบบอะซิงโครนัส

เป็นการสื่อสารข้อมูลในรูปของตัวอักขระแอสกี้โดยผ่านสายข้อมูล ไปที่ตำแหน่งขา I/O ของสมาร์ตการ์ดด้วยรูปแบบ Asynchronous Half Duplex Mode ตัวอักขระแต่ละตัวมีขนาด 8 บิต การเชื่อมต่อแบบนี้ทำงานโดยอาศัยสัญญาณนาฬิกาที่มาจากภายนอกเครื่องอ่าน-เขียนสมาร์ตการ์ด ซึ่งโดยส่วนใหญ่จะใช้ความถี่ค่า 3.57 เมกะเฮิรตซ์ เป็นสัญญาณนาฬิกาของระบบและใช้ป้อนให้วงจรหาร (หาร372) เพื่อเป็นค่า Baud Rate ที่ 9600 บิตต่อวินาที สำหรับรูปสัญญาณการเชื่อมต่อแบบอะซิงโครนัสแสดงดังรูปที่ 2.9 ซึ่งจะเห็นว่าเฟรมข้อมูลแต่ละไบต์นั้นจะประกอบด้วยบิตเริ่มต้นข้อมูลขนาด 8 บิต บิตพาริตี้ บิตสิ้นสุดและการทิ้งช่วงเวลาระหว่างข้อมูลแต่ละชุด

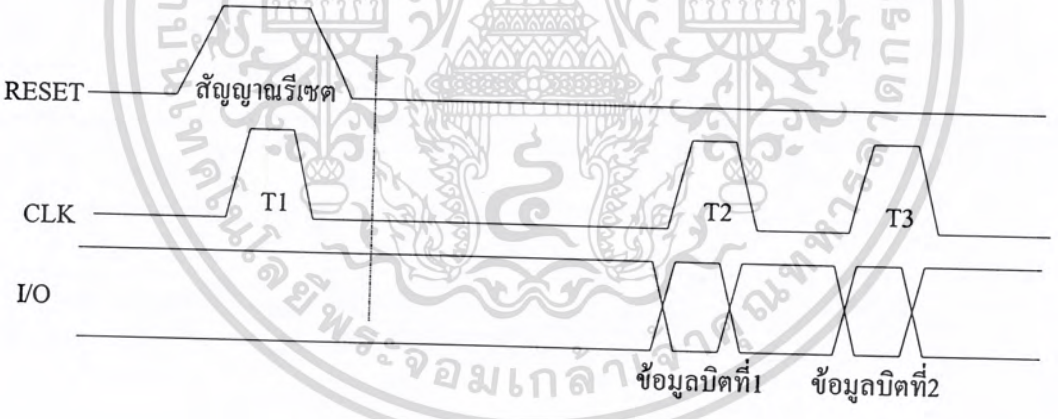
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 รูปสัญญาณการเชื่อมต่อแบบอะซิงโครนัส

2) การเชื่อมต่อแบบซิงโครนัส

เป็นการสื่อสารข้อมูลแบบอนุกรมโดยการรับ-ส่งข้อมูลจะทำกันทีละบิต ผ่านทางสายข้อมูลไปที่ตำแหน่งขา I/O ของสมาร์ตการ์ดด้วยรูปแบบ (Synchronous Half Duplex Mode) ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลจะถูกทำไปทีละบิต โดยอ้างอิงกับสัญญาณนาฬิกาที่การ์ดได้รับที่ขาสัญญาณนาฬิกาเป็นหลัก รูปสัญญาณการเชื่อมต่อแบบซิงโครนัสแสดงดังรูปที่ 2.10



รูปที่ 2.10 รูปสัญญาณการเชื่อมต่อแบบซิงโครนัส

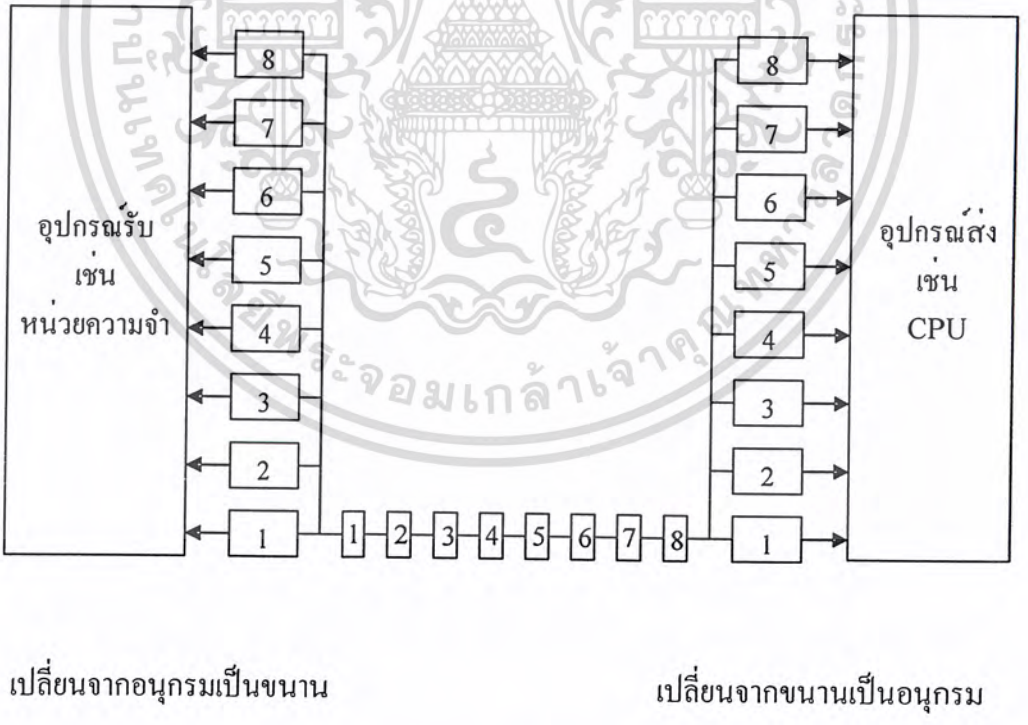
2.3 การสื่อสารข้อมูลแบบอนุกรม

ลักษณะของการสื่อสารข้อมูลแบบอนุกรม การสื่อสารข้อมูลแบบอนุกรมเป็นการสื่อสารข้อมูล โดยที่ข้อมูลที่จะรับหรือส่งผ่านสายนำสัญญาณเพียง 1 คู่ เท่านั้น คือ สายนำสัญญาณที่จะใช้เป็นสายข้อมูล และสายกราวด์ ลักษณะของการรับหรือส่งข้อมูลครั้งละบิต ซึ่งถ้าหากเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันกับการสื่อสารข้อมูลแบบขนาน ที่จำนวนข้อมูลและอัตราเร็วในการสื่อสารข้อมูลเท่ากันแล้ว การสื่อสารข้อมูลแบบอนุกรมจะต้องใช้เวลาในการรับส่งข้อมูลมากกว่า ดังแสดงในรูปที่ 2.11 แต่ข้อดีของการสื่อสารข้อมูลแบบอนุกรมคือ การใช้สายนำสัญญาณน้อยกว่าและสามารถส่งสัญญาณได้ในระยะทางที่ไกลกว่า แม้อัตราการลดทอน หรือการผิดเพี้ยนของสัญญาณที่มีผลจากความยาวของสายสัญญาณจะมีค่าเท่ากับการสื่อสารข้อมูลแบบขนาน การสื่อสารข้อมูลแบบอนุกรมจะมีวิธีการที่จะลดผลการลดทอนของสัญญาณ โดยอาศัยหลักการรับหรือส่งสัญญาณแบบดิฟเฟอเรนเชียล

ดังนั้นการสื่อสารข้อมูลแบบอนุกรมจึงเหมาะสำหรับใช้กับการสื่อสารข้อมูลในระยะไกล หรือการสื่อสารที่ต้องใช้สายหรือช่องสัญญาณในการรับหรือส่งข้อมูลจำนวนน้อย เช่น การสื่อสารข้อมูลโครงข่ายแบบท้องถิ่น (Local Area Network : LAN) การสื่อสารข้อมูลแบบอนุกรมตามที่ได้กล่าวมาแล้วนั้น การสื่อสารข้อมูลแบบอนุกรมยังสามารถที่จะแบ่งตามลักษณะของทิศทางในการสื่อสารข้อมูลตามโครงสร้างและความต้องการของระบบ



รูปที่ 2.11 โครงสร้างของการสื่อสารข้อมูลแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) การสื่อสารข้อมูลในทิศทางเดียวตลอดเวลา หรือแบบซิมเพล็กซ์ (Simplex)

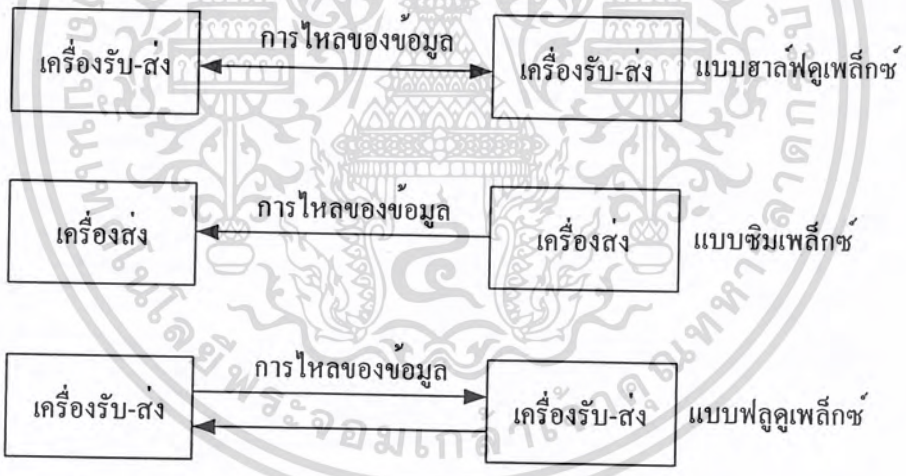
เป็นการสื่อสารข้อมูลที่ข้อมูลสามารถส่งได้ในทิศทางเดียวเท่านั้น เมื่อทำการสื่อสารในทิศทางใดก็จะใช้ทิศทางนั้นตลอดเวลา ไม่มีการเปลี่ยนแปลงทิศทาง เช่นการส่งสัญญาณภาพจากสถานีโทรทัศน์ไปยังเครื่องรับโทรทัศน์หรือการส่งข้อมูลจากศูนย์บริการไปยังวิทยุติดตามตัว

2) การสื่อสารข้อมูลแบบ 2 ทิศทางคนละเวลา หรือแบบฮาล์ฟดูเพล็กซ์ (Half duplex)

เป็นการสื่อสารข้อมูลที่สามารถส่งได้ 2 ทิศทาง โดยจะทำการส่งในลักษณะการผลัดกันรับและส่ง โดยในขณะเวลาหนึ่งนั้น สัญญาณจะได้ไปได้ในทิศทางเดียว ดังนั้นอุปกรณ์แต่ละตัวที่จะเชื่อมต่อหรือสื่อสารข้อมูลในลักษณะนี้จะต้องเป็นไปได้ทั้งตัวรับและตัวส่ง (Transceiver and Receiver) และจะต้องมีวงจรที่จะเลือกว่า ณ เวลานั้นจะทำงานเป็นตัวรับหรือตัวส่ง

3) การสื่อสารข้อมูลแบบ 2 ทิศทางตลอดเวลา หรือแบบฟูลดูเพล็กซ์ (Full duplex)

เป็นการสื่อสารข้อมูลที่คล้ายกับฮาล์ฟดูเพล็กซ์แต่เป็นการสื่อสารข้อมูลในสองทิศทางตลอดเวลา



รูปที่ 2.12 รูปแบบของการติดต่อสื่อสารข้อมูลแบบอนุกรม

2.3.1 การสื่อสารข้อมูลแบบอนุกรมมาตรฐาน RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง มาตรฐานนี้ในช่วงแรกจะใช้คอนเนคเตอร์ (Connector) เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12 โวลต์ แสดงว่ามีข้อมูล (Mark) และ +3 โวลต์ ถึง +12 โวลต์ แสดงว่าเป็นช่องว่าง (Space)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Communication Equipment : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ (Microcomputer) ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วน อุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูล ระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเนคเตอร์ ของ DTE จะเป็นตัวผู้ ส่วนคอนเนคเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเนคเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือ เมาส์ (Mouse) โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

พอร์ตติดต่อสื่อสารกับอุปกรณ์ภายนอกของคอมพิวเตอร์แบบอนุกรมมาตรฐาน RS-232 นี้ เป็นระบบการส่งข้อมูลในรูปแบบอนุกรม คือ ข้อมูลจะส่งไปได้ทีละ 1 บิต โดยจะมีอัตราการส่ง เป็นบิตต่อวินาที หรืออัตราการเปลี่ยนแปลงของสัญญาณใน 1 วินาที เรียกว่า Baud Rate โดยจะมี การส่งบิตเริ่มต้น (Start Bit) มีระดับสัญญาณเป็น “0” และบิตข้อมูล (Data Bit) ซึ่งอาจจะมีข้อมูล 7 หรือ 8 บิต และอาจตามด้วยบิตพาริตี้ (Parity Bit) ซึ่งอาจจะเป็นแบบคู่ (Even) หรือแบบคี่ (Odd) เพื่อตรวจความถูกต้องของข้อมูล บิตพาริตี้นี้อาจจะมีหรือไม่มีก็ได้ และสุดท้ายจะตามด้วยบิตจบ การทำงาน (Stop Bit) ซึ่งอาจจะมีความกว้างของสัญญาณเป็น 1 หรือ 2 บิตก็ได้ ซึ่งการส่งข้อมูล แบบนี้ จำเป็นต้องมีข้อตกลงกันระหว่างการรับการส่งคือ

1) ความเร็วในการส่ง Baud Rate

2) จำนวนข้อมูล

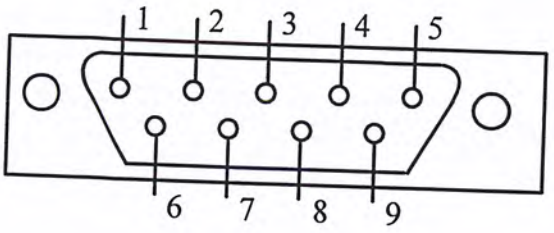
3) มีบิตพาริตี้หรือไม่ ถ้ามีจะเป็นแบบคู่หรือคี่

4) จำนวนบิตเริ่มต้นเป็น 1 หรือ 2 บิต

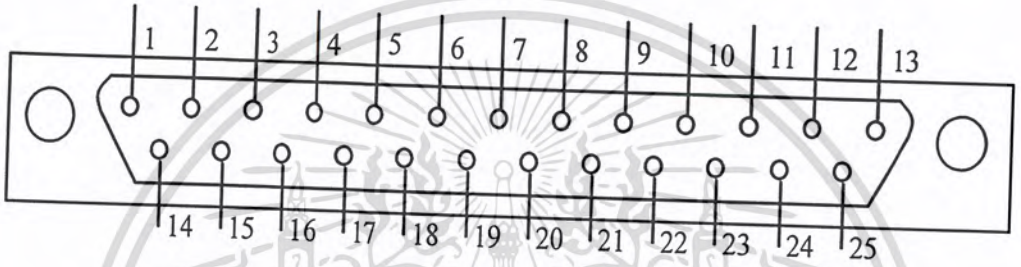
มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเนคเตอร์แบบ DB-9 ตัวผู้ ซึ่งคอนเนคเตอร์ แบบ DB-9 จะมีขาต่อใช้งานเพียง 9 เส้น เนื่องจากขาอื่นๆ ที่เคยใช้งานในอดีต แต่ปัจจุบันมีการใช้ งานไม่มากนักจึงถูกยกเลิกไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.13 คอนเนคเตอร์อนุกรม 9 ขา หรือแบบ DB-9



รูปที่ 2.14 คอนเนคเตอร์อนุกรม 25 ขา หรือแบบ DB-25

สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

ขา 1 CD : Carrier Detect หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกทีฟ (Active) เมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการใช้งานปกติขานี้จะไม่ได้ถูกใช้งานมากนัก

ขา 2 RXD : Receiver Data ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามาคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์บัฟเฟอร์

ขา 3 TXD : Transmitters Data ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยจะนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

ขา 4 DTR : Data Terminal Ready เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็นแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา CD ด้วย

ขา 5 GND : Signal Ground ขากราวด์ของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

51859

ขา 6 DSR : Data Set Ready ใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

ขา 7 RTS : Request To Send เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อขา RTS เข้ากับขา CTS ของตัวมันเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

ขา 8 CTS : Clear To Send ขารับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TXD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

ขา 9 RI : Ring Indicator ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

ตารางที่ 2.2 การจัดขาคอนเนคเตอร์พอร์ตอนุกรม แบบ DB-9 และ DB-25

DB-9	DB-25	ชื่อของสายสัญญาณ	ชนิดสายสัญญาณ
1	8	CD : Carrier Detect	อินพุต
2	3	RXD : Receiver Data	อินพุต
3	2	TXD : Transmitters Data	เอาต์พุต
4	20	DTR : Data Terminal Ready	เอาต์พุต
5	7	GND : Signal Ground	-
6	6	DSR : Data Set Ready	อินพุต
7	4	RTS : Request To Send	เอาต์พุต
8	5	CTS : Clear To Send	อินพุต
9	22	RI : Ring Indicator	อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การเปรียบเทียบมาตรฐานการสื่อสารข้อมูลทาง EIA

พารามิเตอร์	RS-232	RS-423A	RS-422A	RS-485
โหมดการทำงาน	Single-ended	Single-ended	Differential	Differential
จำนวนของตัวรับและตัวส่งข้อมูล	1 ตัวรับ 1 ตัวส่ง	1 ตัวรับ 10 ตัวส่ง	1 ตัวรับ 10 ตัวส่ง	32 ตัวรับ 32 ตัวส่ง
ความยาวของคู่สายสัญญาณรับส่งข้อมูล	50 ฟุต	4,000 ฟุต	4,000 ฟุต	4,000 ฟุต
อัตราการส่งข้อมูลสูงสุด (บิตต่อวินาที)	20 k	100 k	10 M	10 M
แรงดันไฟฟ้าโหมดร่วมสูงสุด	± 2.5 V	± 6 V	+ 6 V - 2.5 V	+ 12 V - 7 V
Driver Output	± 5 V ต่ำสุด ± 15 V สูงสุด	± 3.6 V ต่ำสุด ± 6.0 V สูงสุด	± 2 V ต่ำสุด	± 1.5 V ต่ำสุด
Driver Load (W)	3k ถึง 7k	450 ต่ำสุด	100 ต่ำสุด	60 ต่ำสุด
Driver Slow Rate	30V/A สูงสุด	-	NA	NA
ค่าความต้านทานเอาต์พุตของตัวส่ง (W)	NA-Power On 300-Power off	NA-Power On 600k-Power off	NA-Power On 60k-Power off	120k-Power On, off
ค่าความต้านทานอินพุตของตัวส่ง (W)	3k ถึง 7k	4k	4k	12k
ความไวตัวรับ	± 3 V	± 200 Ma	± 200 ma	± 200 ma

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การติดต่อสื่อสารโดยระบบ I^2C บัส

I^2C (Inter-IC Communication) หมายถึง การติดต่อสื่อสารระหว่างไอซีโดย I^2Cบัส การติดต่อสื่อสารแบบนี้ได้รับการพัฒนาขึ้นโดยบริษัทฟิลิปส์ ด้วยจุดมุ่งหมายหลัก คือ ต้องการติดต่อให้ไอซีหรือโมดูลสามารถติดต่อสั่งงานและควบคุมภายใต้สายสัญญาณเพียง 2 เส้น คือ สายข้อมูล และ สายสัญญาณนาฬิกาที่ใช้กำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์แบบ I^2Cบัส ทำได้โดยการต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว สายข้อมูลบน I^2C บัสมีชื่อเรียกอีกอย่างว่าสายข้อมูลอนุกรมหรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่าสายสัญญาณนาฬิกาอนุกรมหรือ SCL (Serial Clock Line) อุปกรณ์ที่ทำการเชื่อมต่อบน I^2Cบัสมีหลากหลายไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาต์พุต ไอซีแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล ไอซีแปลงสัญญาณดิจิตอลเป็นสัญญาณแอนะลอก ไอซีเรียลไทม์คล็อก ไอซีขับจอแสดงผลแบบผลึกเหลว หน่วยความจำอีพรอมและไมโครคอนโทรลเลอร์

คุณสมบัติโดยทั่วไปของ I^2Cบัส สาย SDA และสาย SCL เป็นสัญญาณแบบ 2 ทิศทาง ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน + 5 โวลต์ เพื่อให้สถานะลอจิกสูงขณะที่ไม่มีการติดต่อใช้งานและมีลักษณะเป็นวงจรเดรนเปิดหรือคอนเลคเตอร์เปิด

อัตราการถ่ายทอดข้อมูลบน I^2Cบัส สูงถึง 100 กิโลไบต์ต่อวินาทีในโหมดปกติและสูงถึง 400 กิโลไบต์ต่อวินาทีในโหมดความเร็วสูง อุปกรณ์ที่ต่อร่วมอยู่แบบ I^2C บัสนี้ต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 พิโคฟารัด การเข้าถึงอุปกรณ์บน I^2C บัสใช้ข้อมูลสำหรับการเข้าถึง 2 ค่า คือ 7 บิตหรือ 10 บิต ข้อเด่นประการหนึ่งของ I^2Cบัส คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้ใช้ไฟเลี้ยง +12 โวลต์ การต่อร่วมกันบน I^2Cบัส สามารถกระทำให้ได้ในลักษณะเดียวกันกับกรณีที่อุปกรณ์ทั้งสองที่ใช้ไฟเลี้ยงร่วมกัน

2.5 บาร์โค้ด (BARCODE)

2.5.1 ความเป็นมาของบาร์โค้ด

สหรัฐอเมริกาได้ออกสิทธิบัตรรับรองบาร์โค้ดขึ้นเมื่อปี ค.ศ. 1949 ในแบบที่เรียกว่า Circular Bar Code ต่อมาในปี ค.ศ. 1960 ก็มีการรับรองบาร์โค้ดแบบที่เรียกว่า Rall Identification Symbol หลังจากนั้นเป็นต้นมาเทคนิคของบาร์โค้ดรูปแบบต่างๆ ก็มีมากขึ้น และเริ่มใช้งานจริงจังเมื่อปี ค.ศ. 1970 เมื่อคณะกรรมการการบริหารด้านห้างสรรพสินค้าของสหรัฐอเมริกา ได้นำรหัสที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรียกว่า UPC (Universal Product Code) ซึ่งเป็นที่ใช้กันมากในสินค้า ออกเผยแพร่และใช้กันอย่างแพร่หลายในสหรัฐอเมริกาและยุโรป ตั้งแต่ ค.ศ. 1973 และ ค.ศ. 1977 ตามลำดับ

การใช้งานด้านอื่นๆ เริ่มตั้งแต่ปี ค.ศ. 1980 เป็นต้นมา แต่คนส่วนใหญ่เริ่มคุ้นเคยกับบาร์โค้ดเป็นอย่างดีจากรหัสสินค้าและการชำระเงินที่คอมพิวเตอร์รวมออกมาจากบาร์โค้ดสินค้าเหล่านั้น จากความสะดวกเหล่านี้สามารถลดพนักงาน ณ จุดนี้ลงได้

ในปี ค.ศ. 1981 ห้างสรรพสินค้ามากกว่า 4,000 แห่ง ในสหรัฐอเมริกาและแคนนาดาใช้บาร์โค้ดในธุรกิจนี้ นอกจากนี้ยังใช้กับกิจการอื่นๆ เช่น ห้องสมุด บริการสุขภาพ งานบริการ การผลิตสินค้าเป็นต้น

2.5.2 หลักการของบาร์โค้ด

บาร์โค้ดเป็นการแทนข้อมูลที่เป็นรหัสฐานสอง (Binary Code) ในรูปแบบของแถบขาว-ดำที่มีความกว้างแคบต่างกัน วางเรียงขนานสลับกัน ด้วยจำนวนของแถบรหัสสลับกันไป ขึ้นอยู่กับข้อมูลที่ต่างกันและชนิดของบาร์โค้ดที่เลือกใช้ต่างกันด้วย

1) การเข้ารหัสของบาร์โค้ด(Bar Code) แบ่งออกเป็น 2 วิธีการคือ

1.1) จะใช้สีของบาร์โค้ดนำมาเข้ารหัส โดยใช้แถบสีดำแทน “1” และแถบสีขาวหรือแถบว่างเป็น “0” ลักษณะเช่นนี้เรียกว่า “เดลต้าโค้ด” (Delta Code)

1.2) จะใช้ความกว้างของแถบรหัสนำมาเข้ารหัส โดยถ้าเป็นแถบกว้างจะเป็น “1” และแถบแคบจะเป็น “0” ลักษณะเช่นนี้เรียกว่า “วิดท์โค้ด” (Width Code) การเข้ารหัสชนิดนี้จะไม่สนใจสีของแถบเลย

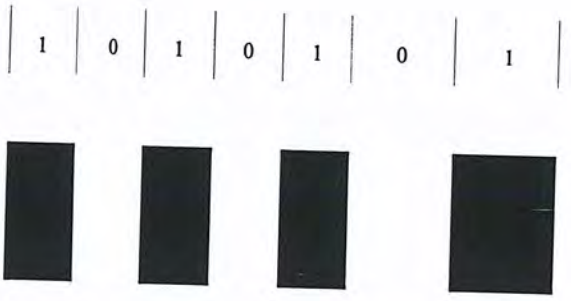
แถบขาว-ดำ มีลักษณะและชื่อที่ใช้คือ

-แถบสีดำมีความกว้างมาก เรียกว่า Width Bar

-เส้นหรือแถบสีดำที่มีความกว้างน้อยเรียกว่า Narrow Bar

-ช่องว่าง (แถบสีขาว) ที่มีความกว้างมาก เรียกว่า Wide Space – ช่องว่าง (เส้นสีดำ) ที่มีความกว้างน้อย เรียกว่า Narrow Space

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 การแทนค่าเลขฐานสองของแถบต่าง ๆ

2) การอ่านบาร์โค้ด

ในการอ่านบาร์โค้ดใช้หลักการเปลี่ยนรหัสบาร์โค้ดให้เป็นรหัสแอสกี้ โดยอาศัยความแตกต่างกันระหว่างแถบเข้มกับพื้นที่ว่าง โดยพื้นที่ว่าง (ปกติเป็นสีขาวหรือสีอ่อน) จะมีการสะท้อนกลับของแสงได้มากกว่าบริเวณที่เป็นแถบเข้ม (ซึ่งจะใช้สีดำหรือสีที่มีความเข้มมาก) หัวอ่าน (Barcode Reader) จะประกอบด้วยตัวกำเนิดแสงที่ผ่านเลนส์ออกมา โดยถูกบังคับทิศทางให้มีจุดรวมแสงเล็กที่สุดกับตัวรับแสงที่มีความไวสูง ทั้งสองอย่างนี้จะบรรจุไว้ในหัวอ่านเดียวกัน ที่มีหลายรูปแบบ แต่แบบที่เป็นพื้นฐานที่สุดอยู่ในรูปคล้ายปากกาขนาดใหญ่ (Wand Type)

หัวอ่านจะสแกนผ่านรหัสบาร์โค้ดในขณะที่ตัวกำเนิดแสง จะทำให้เกิดแสงส่งผ่านเลนส์ไปกระทบบนรหัสบาร์โค้ด และสะท้อนกลับจากแถบกลับไปยังตัวรับแสง (Photo sensor) ที่เกิดค่าความแตกต่างขึ้น ตามหลักการสะท้อนกลับในแต่ละแถบ ทำให้เกิดสภาวะลอจิก “1” และสภาวะ “0” ขึ้นตามที่กล่าวมาข้างต้นซึ่งเมื่อรวมสภาวะลอจิก”1” และ “0” ทั้งหมด ตลอดความกว้างของทุกแถบแล้วตรงกับแพตเทอร์นที่ได้กำหนดไว้แล้ว ในหัวอ่านรหัสบาร์โค้ดจะใช้ตัวกำเนิดแสงสีแดงหรือสีขาว แต่ส่วนใหญ่จะใช้สีแดงเพราะสีขาวต้องการพลังงานและความเข้มของแสงสูงกว่าแสงสีแดง แสงสีแดงสามารถอ่านรหัสอ่านรหัสที่พิมพ์ด้วยสีต่างๆ ได้ทุกสี ยกเว้นรหัสที่พิมพ์ด้วยสีแดง

องค์ประกอบสำคัญสองประการที่จำเป็นอย่างมากในการอ่านบาร์โค้ดได้ถูกต้อง

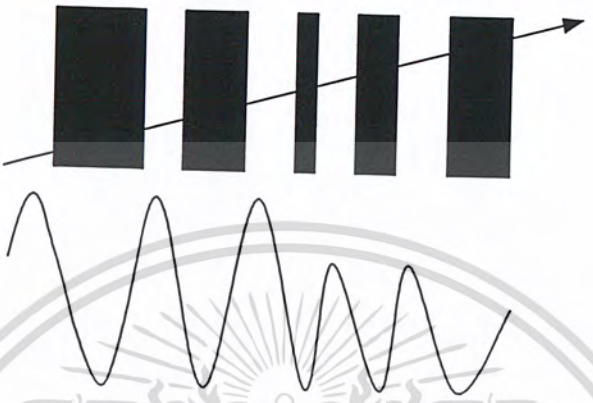
ประการแรกคือ พื้นที่ภายในในแถบและช่องว่างจะต้องทำให้เกิดการสะท้อนกลับอย่างมาก (Contrast) เช่น แถบสีดำและช่องว่างสีขาว เป็นต้น ซึ่งปกติความแตกต่างนี้จะต้องอยู่ในช่วงระหว่างอัตรา 80 – 90 % ขึ้นไป

ประการที่สองคือ ความกว้างระหว่างแถบกว้างหรือช่องว่างต่อแถบแคบหรือช่องว่างแคบจำนวนอัตราส่วน 2 : 0.5, 2 : 1 และ 3 : 1

ตามหลักการของบาร์โค้ดแล้ว สัญญาณที่อ่านได้จากหัวอ่านจะไม่ขึ้นอยู่กับชนิดของบาร์โค้ด แต่จะขึ้นอยู่กับแถบขาว-ดำที่รูดผ่านคือถ้าเป็นแถบดำสัญญาณที่อ่านได้จากหัวอ่านจะเป็น “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าเป็นแถบสีขาวสัญญาณที่อ่านได้จะเป็น “0” ความกว้างของสัญญาณที่อ่านได้จะเท่ากับความกว้างของแถบขาว-ดำ



รูปที่2.16 สัญญาณที่อ่านได้จากบาร์โค้ด

3) ชนิดของบาร์โค้ด

ปัจจุบันบาร์โค้ดที่นิยมใช้กันอย่างแพร่หลาย แบ่งได้เป็น

3.1) ชนิดรหัส 2 ใน 5 (2 of 5 codes)

3.2) ชนิดรหัส 2 ใน 5 แบบสอดแทรก (interleaved 2 of 5)

3.3) ชนิดรหัส 3 ใน 9 (3 of 9)

3.4) ชนิดรหัส Code bar

3.5) ชนิดรหัสสากล UPC (Universal Product Code)

3.6) ชนิดรหัสตัวเลขของยุโรป EAN (European Article Numbering)

3.7) ชนิดรหัส Code 39

4) บาร์โค้ดชนิด 3 ใน 9 (3 of 9)

ชื่อของบาร์โค้ดชนิดนี้บอกถึงโครงสร้างของรหัสว่า 3 ใน 9 ส่วนตัวต่ออักษรจะเป็นแถบกว้างและที่เหลือเป็นแถบแคบอีก 5 แถบ แต่ละตัวอักษรใน Code39 จะแสดงเป็นกลุ่มของแถบดำ (Bar) 5 แถบ และแถบว่าง (Space) 4 แถบ แต่ละตัวอักษรที่สมบูรณ์จะต้องรวมตัวอักษร Star และ Stop เข้าไปด้วย โดยใช้เป็นตัว Asterisk (*) ข้อมูลทั้ง 43 ตัวอักษรประกอบด้วย ตัวเลข 10 ตัวคือ 0 – 9 ตัวอักษร 26 ตัว คือ A- Z Space และเครื่องหมายอีก 6 ตัว (-, ., /, +, %, &) คุณสมบัติที่สำคัญของ Code 39 คือมี Self-Checking ซึ่งทำให้ข้อมูลปลอดภัยสูงและถ้ามีเครื่องอ่านที่ดีและบาร์โค้ดมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณภาพสูง จะพบข้อผิดพลาด (Error) 1 ครั้ง ใน 70 ล้านครั้งของการอ่าน และถ้าบาร์โค้ดมีการพิมพ์มาอย่างดี (โดยใช้เครื่องพิมพ์ที่มีคุณภาพดีกว่า dot matrix) จะพบความผิดพลาดเพียง 1 ครั้ง ใน 3 ล้านครั้ง

Code39 เป็นรหัสที่มีความยาวเปลี่ยนแปลงได้ ขึ้นอยู่กับเครื่องอ่านที่และ Code 39 มี Self-Checking จึงไม่ต้องมี Check Character Code 39 เป็นตัวอักษรชนิดไม่ต่อเนื่อง โดยจะมีช่องว่างระหว่างตัวอักขระ และเป็นรหัสที่อ่านได้ 2 ทาง คือสามารถอ่านได้ทั้งจากขวาไปซ้ายและจากซ้ายไปขวาก็ได้ ขนาดของรหัส 39 จะเปลี่ยนแปลงไปตามความกว้าง สำหรับรหัส 39 ที่มีความหนาแน่นสูงจะมี 9.4 ตัวอักษรต่อนิ้วและรหัส 39 ที่มีความหนาแน่นต่ำจะมี 1.4 ตัวอักษรต่อนิ้ว

5) สรุปคุณลักษณะของรหัส 39

รูปแบบตัวอักษรทั้งหมด	อักษรอังกฤษตัวใหญ่ 26 ตัว (A-Z) ตัวเลข 10 อักษร พิเศษ 7 ตัว สามารถขยายได้เป็น 128 อักษร
ความยาวของแถบรหัส	เปลี่ยนแปลงได้
อักษรที่ใช้ตรวจสอบความผิดพลาด	เป็นส่วนที่มีหรือไม่มีก็ได้
อักษร Overload	2 ตัวต่อ 1 แถบรหัส
ความหนาแน่น	มากที่สุดได้ 9.8 หลักต่อ 1 นิ้ว

6) คำอธิบายของบาร์โค้ดชนิด 39

ทุก ๆ แถบรหัสของบาร์โค้ดชนิด 39 ประกอบด้วย

6.1) นำด้วย Quite Zone

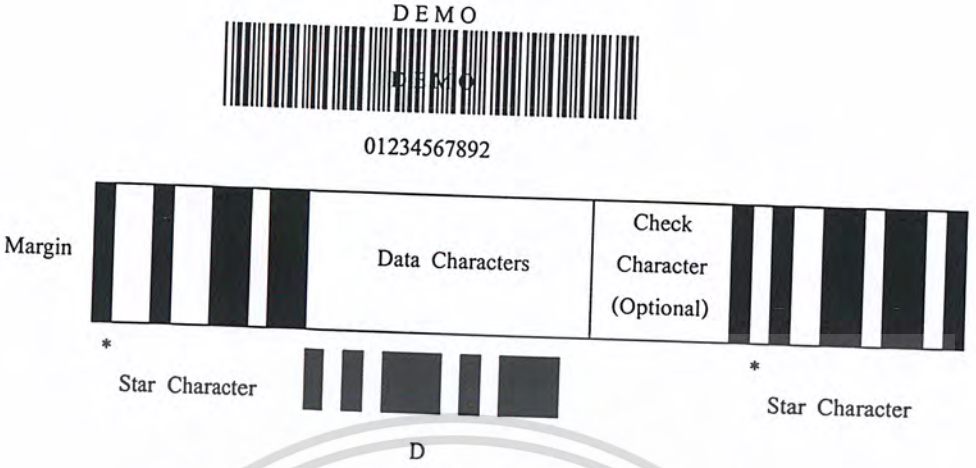
6.2) รูปแบบตัวอักษรเริ่มขึ้น

6.3) อักษรข้อมูล

6.4) รูปแบบอักษรสิ้นสุด

6.5) ปิดท้ายด้วย Quite Zone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 โครงสร้างของรหัสบาร์โค้ด 39

2.6 คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ตระกูล MCS-51

คุณสมบัติทั่วไปของไมโครคอนโทรลเลอร์ MCS-51 มีดังนี้

1) เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต

2) มีวงจรออสซิลเลเตอร์ และวงจรผลิตสัญญาณนาฬิกาภายในไอซี

3) มีขาสัญญาณอินพุตเอาต์พุตขนาด 32 บิต

4) สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก (External Data) โดยอ้างตำแหน่งหน่วยความจำได้ถึง 64 กิโลไบต์

5) สามารถเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก (External Program Memory) โดยสามารถอ้างตำแหน่งหน่วยความจำได้ถึง 64 กิโลไบต์

6) มีหน่วยความจำโปรแกรมภายในตัว (On-Chip Program Memory) ขนาด 4 กิโลไบต์ โดยเฉพาะเบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 กิโลไบต์ สำหรับเบอร์ 8031 และM8032 จะไม่มีหน่วยความจำในส่วนนี้

7) มีหน่วยความจำภายในตัว (On-Chip Data Memory) ขนาด 128 ไบต์ โดยเฉพาะเบอร์ 8032 และ 8052 จะมีในส่วนนี้ถึง 256 ไบต์

8) มีหน่วยความจำบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุม หรือ การตรวจสอบสถานะบิตทำได้ง่าย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายขึ้น

9) มีไทม์เมอร์/เคาน์เตอร์ (Timers/Counters) ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะเบอร์ 8032 และ 8052 จะมีไทม์เมอร์/เคาน์เตอร์จำนวน 3 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10) การอินเตอร์รัพต์ทำได้จาก 6 แหล่งกำเนิด โดยเฉพาะเบอร์ 8032 และ8052 จะทำการอินเตอร์รัพต์ได้จาก 6 แหล่งกำเนิด โดยการอินเตอร์รัพต์ยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ

11) มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบฟูลดูเพล็กซ์ (Full Duplex)

12) มีคำสั่งในการคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์คำสั่งโดยส่วนใหญ่ใช้เวลาการทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์

13) ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว

2.7 โครงสร้างภายนอกของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาที่เหมือนกัน ดังแสดงในรูปที่ 2.18 สำหรับหน้าที่การใช้งานของแต่ละขามีดังนี้

1) ขา Vcc เป็นขาป้อนแรงดันไฟเลี้ยง +5 โวลต์

2) ขา Vss เป็นกราวด์

3) ขาพอร์ต 0 (PORT 0) มี 8 ขาได้แก่ P0.0 -P0.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับการใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตต้องทำการเขียนค่าเป็น 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสถานะปล่อยลอย ซึ่งในสถานะนี้เองที่สามารถนำมาใช้เป็นพอร์ตอินพุตอิมพิแดนซ์สูงได้ นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งหน่วยความจำไบต์ต่ำ (A0-A7) ซึ่งจะใช้งานเป็นแบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 บิต (D0-D7)

4) ขาพอร์ต 1 (PORT 1) มี 8 ขา ได้แก่ขา P1.0-P1.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับการใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตต้องทำการเขียนค่าเป็น 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต

5) ขาพอร์ต 2 (PORT 2) มี 8 ขา ได้แก่ขา P2.0-P2.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับการใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตต้องทำการเขียนค่าเป็น 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุตนอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วยโดยทำหน้าที่ในการกำหนดตำแหน่งหน่วยความจำภายนอก (A8-A15) และใช้เป็นพอร์ตอินพุตอิมพิแดนซ์สูงได้

6) ขาพอร์ต 3 (PORT 3) มี 8 ขา ได้แก่ขา P3.0-P3.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับการใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตต้องทำการเขียนค่าเป็น 1 ไปยังแต่ละบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกำหนดให้เป็นพอร์ตอินพุตนี้นอกจากจะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้ว ยังถูกใช้งานในหน้าที่พิเศษต่างๆ ดังแสดงในตารางที่ 2.4

7) ขารีเซตใช้สำหรับรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซตต้องคงสถานะเป็น 1 อย่างน้อยนาน 2 รอบการทำงานของคำสั่ง ในขณะที่ออสซิลเลเตอร์ยังทำงานอยู่

8) ขา ALE / $\overline{\text{PROG}}$ เป็นขาสัญญาณที่ทำหน้าที่ควบคุมการค้างสภาวะ (Latch) ค่าตำแหน่งไบต์ต่ำ (Address Latch Enable) เมื่อติดต่อกับหน่วยความจำภายนอกและทำหน้าที่เป็นอินพุตรับพัลส์ในการโปรแกรม ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ตระกูล MCS-51 ที่มีหน่วยความจำตำแหน่งภายใน EPROM

9) ขา $\overline{\text{PSEN}}$ ทำหน้าที่ส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอกเมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกที่ขานี้ 2 ครั้งในแต่ละรอบการทำงานของคำสั่ง แต่ถ้ามีการติดต่อกับหน่วยความจำข้อมูลภายนอก ขานี้จะไม่มีการส่งสัญญาณใดๆ ออกมา

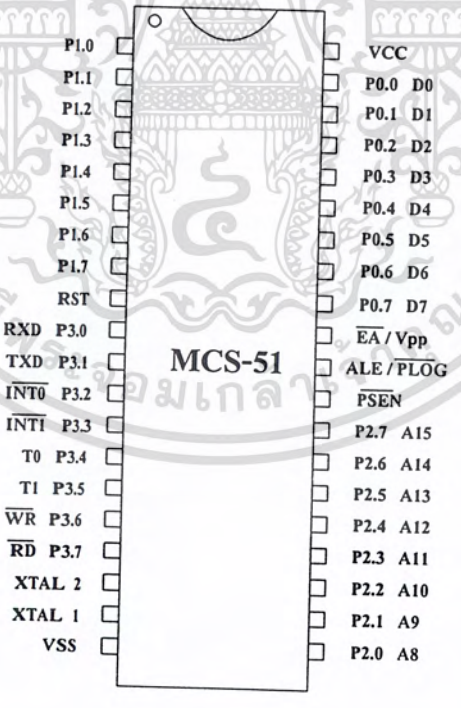
10) ขา $\overline{\text{EA}}$ /Vpp เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายใน หรือจากภายนอก โดยถ้ามีสถานะเป็น “0” จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-0FFFH อย่างไรก็ตามถ้าบิตป้องกัน (Security Bit) ในหน่วยความจำ EPROM ถูกโปรแกรม ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำจากนี้ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (Vpp) ขนาด 12 โวลต์ เพื่อใช้ในระหว่างการโปรแกรม EPROM

11) ขา XTAL1, XTAL2 เป็นขาอินพุตและเอาต์พุตของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟเออร์ (Inverting Oscillator Amplifier) สำหรับใช้ต่อร่วมกับคริสตอลภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

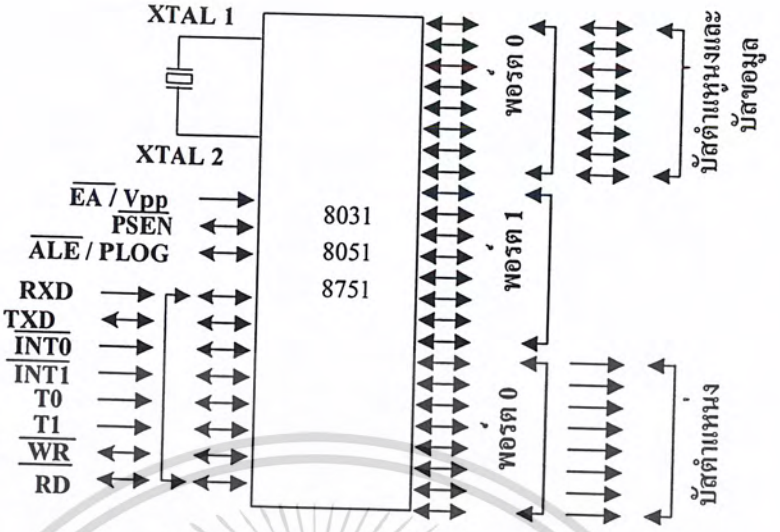
ตารางที่ 2.4 หน้าที่พิเศษของแต่ละขาของพอร์ต P3

ขาพอร์ต	หน้าที่พิเศษ
P3.0	RXD (Serial Input Port)
P3.1	TXD (Serial Output Port)
P3.2	$\overline{\text{INT0}}$ (External Interrupt 0)
P3.3	$\overline{\text{INT1}}$ (External Interrupt 1)
P3.4	T0 (Timer 0 External Input)
P3.5	T1 (Timer 1 External Input)
P3.6	$\overline{\text{WR}}$ (External Data Memory Write Strobe)
P3.7	$\overline{\text{RD}}$ (External Data Memory Read Strobe)



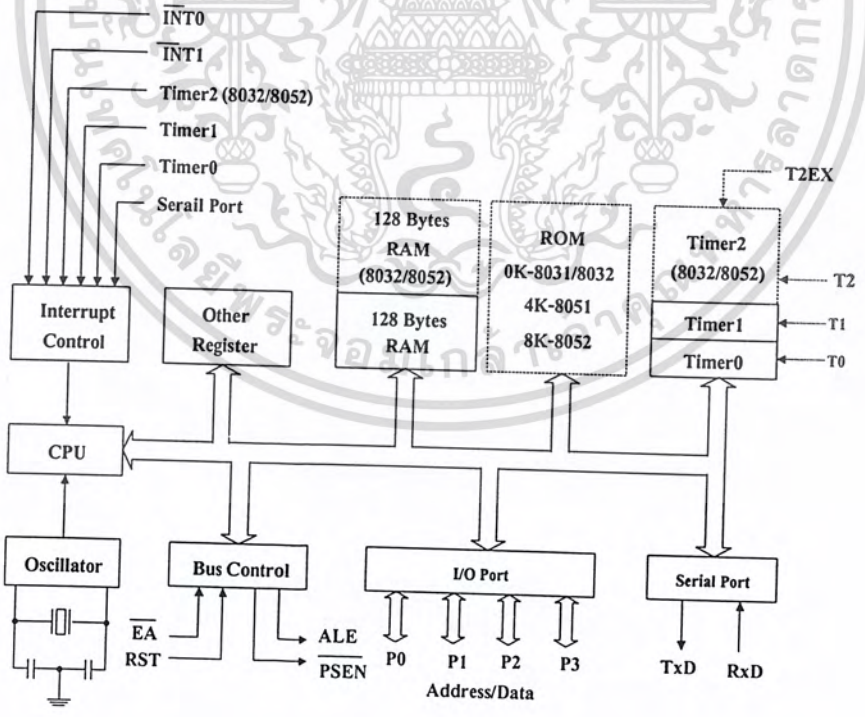
รูปที่ 2.18 การจัดตำแหน่งขาต่างๆ ของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 หน้าที่ของพอร์ตเมื่อไมโครคอนโทรลเลอร์ทำงานกับหน่วยความจำภายนอก

2.8 คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล MCS-51



รูปที่ 2.20 โครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต

2) ภายในมีหน่วยความจำโปรแกรมแบบแฟลชสามารถลบและเขียนใหม่ได้ 1 หมื่นครั้ง

3) หน่วยความจำข้อมูลพื้นฐานเป็นความจำแบบแรม 256 ไบต์ แบบอีพรอม 2 กิโลไบต์

4) ขาพอร์ตเป็นแบบ 2 ทิศทาง สามารถใช้งานได้เป็นทั้งอินพุต และเอาต์พุต

5) มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์

6) ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต 3 ตัว

7) สามารถรองรับแหล่งกำเนิดอินเตอร์รัพต์ได้ 6 แหล่ง

8) มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

9) มีวอตช์ด็อกไทเมอร์ในตัว

2.9 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีพอร์ตใช้งานทั้งสิ้น 4 พอร์ต คือ พอร์ต 0 ถึง พอร์ต 3 มีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง สามารถเป็นได้ทั้งพอร์ตอินพุต สำหรับรับสัญญาณข้อมูลเข้าและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออกทุกพอร์ต ของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีวงจรแลตช์ วงจรขับบัฟเฟอร์อินพุตส่วนที่พอร์ต 0 กับพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตสำหรับงานทั่วไปและใช้ในการติดต่อกับหน่วยความจำภายนอก สำหรับพอร์ต 3 และพอร์ต 1 บางขานอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก

2.9.1 การใช้งานพอร์ตอินพุต

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาต์พุตดังนั้นจึงมีความจำเป็นที่ต้องกำหนดลักษณะการทำงานให้พอร์ตไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาทีละบิตของพอร์ตที่ต้องการใช้งานเป็นพอร์ตอินพุต เพื่อหยุดการทำงานของมอสเฟตที่ใช้ในการขับสัญญาณเอาต์พุตของพอร์ตนั้นๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัปภายในโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณ "0" จากอุปกรณ์ได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ในวงจรบัฟเฟอร์ภายในพอร์ต แล้วให้ซีพียูมาอ่านค่าเข้าไป เมื่ออุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 การใช้งานเป็นพอร์ตเอาต์พุต

โดยปกติแล้วขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาต์พุตอยู่แล้ว ดังนั้นจึงสามารถส่งข้อมูลออกไปได้อย่างง่ายดายและตรงไปตรงมา เมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาต์พุต ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ ซึ่งก็จะส่งต่อไปขับมอสเฟต ทำให้มอสเฟตทำงานที่ขาพอร์ตที่กำหนดให้ ทำงานก็จะเกิดลอจิก "0" ขึ้น ในทางตรงกันข้ามหากต้องการส่งข้อมูล "1" ออกไปก็ให้เขียนข้อมูล "1" ที่ขาพอร์ตนั้น ซึ่งจะคล้ายกับการกำหนดให้เป็นขาอินพุตมากเพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาต์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่กรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุต

เมื่อใช้งานพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเป็นพอร์ตเอาต์พุต แต่ละขาของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสซอร์ส ได้สูงสุด 10 มิลลิแอมป์ และทุกขารวมกันในพอร์ต 0 (ทั้ง 8 บิต) สูงสุด 26 มิลลิแอมป์และ 15 มิลลิแอมป์ สำหรับพอร์ต 1-3 ในกรณีที่ใช้งานเป็นพอร์ตเอาต์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแส จึงควรต่อวงจรบัฟเฟอร์ทางเอาต์พุตเพื่อช่วยในการขับกระแสให้กับโหลด

2.10 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล

ในหัวข้อนี้ จะกล่าวถึงความรู้พื้นฐานที่ควรทราบและแนะนำให้รู้จักคำศัพท์ที่เกี่ยวข้องกับระบบฐานข้อมูล

2.10.1 คำศัพท์พื้นฐาน

1) ข้อมูล (Data) หมายถึง ข้อเท็จจริง ซึ่งอาจจะเรียกว่าเป็นวัตถุดิบของสารสนเทศเมื่อข้อมูลถูกนำมาประมวลผล (เรียงลำดับ, แยกประเภท, เชื่อมโยง, คำนวณหรือสรุปผล) และจัดให้อยู่ในรูปแบบที่นำมาใช้ประโยชน์ได้ เราจึงเรียกว่าเป็น สารสนเทศ

2) บิต (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด

3) ไบต์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำบิตหลายๆ บิตมารวมกันเป็นตัวอักษร

4) ฟิลด์ (Field) หมายถึง หน่วยของข้อมูลที่ประกอบด้วยตัวอักษรหลายๆ ตัวอักษร เพื่อแทนความหมายของสิ่งๆ หนึ่ง

5) เรคคอร์ด (Record) หมายถึง หน่วยของข้อมูลที่เกิดจาก การนำเอาฟิลด์หลายๆ ฟิลด์ มารวมกัน เพื่อแสดงรายละเอียดของข้อมูลเรื่องใดเรื่องหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) แฟ้มข้อมูล (File) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำเอาเรคคอร์ดหลายๆ เรคคอร์ดมารวมกันสำหรับในระบบฐานข้อมูล

7) ฐานข้อมูล (Database) คือ โครงสร้างของสารสนเทศ (Information) ที่ประกอบด้วยเอนทิตีหลายๆตัว ซึ่งเอนทิตีเหล่านี้มีความสัมพันธ์กัน

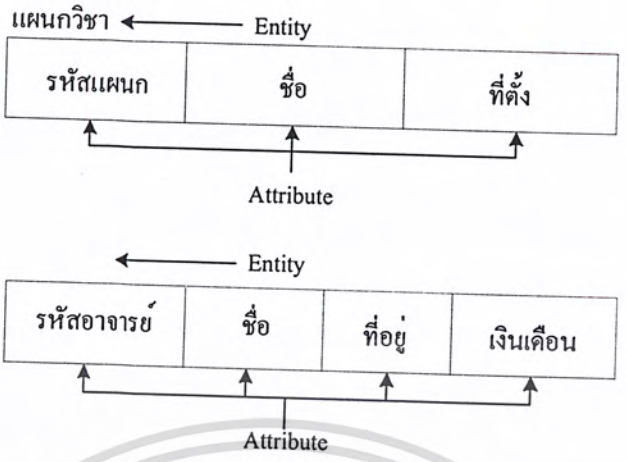
2.10.2 เอนทิตี (Entity), แอททริบิวต์ (Attribute) และความสัมพันธ์

คำว่าเอนทิตีเปรียบเสมือนกับเป็นคำนามอันได้แก่ บุคคล สถานที่ และสิ่งของ เช่น เอนทิตีในระบบฐานข้อมูลก็ได้แก่ อาจารย์ แผนกวิชา แผนการประกันสุขภาพ ประวัติการทำงาน หรือถ้าเรากำลังสนใจในการสร้างระบบฐานข้อมูล เกี่ยวกับระบบการขายของบริษัทแห่งหนึ่ง เอนทิตี ของระบบนี้ก็จะได้แก่ พนักงานขาย ลูกค้า การสั่งซื้อ และสินค้า เป็นต้น

แอททริบิวต์ ก็คือ ข้อมูลที่แสดงลักษณะและคุณสมบัติของเอนทิตี เช่น แอททริบิวต์ของบุคคลก็อาจจะได้แก่ สีผม สีตา อายุ เพศ ชื่อ ฯลฯ และถ้าจะออกแบบแอททริบิวต์ ของอาจารย์ที่เราสนใจได้แก่ รหัสอาจารย์ ชื่อ ที่อยู่ เงินเดือนและอื่นๆ ดังที่แสดงในรูปที่ 2.21 เป็นเอนทิตี 2 ตัว ได้แก่อาจารย์และแผนกวิชา โดยที่แผนกวิชาเป็นเอนทิตีที่ประกอบด้วยแอททริบิวต์ 3 ตัว คือ รหัสแผนก ชื่อแผนกและสถานที่ตั้ง ส่วนอาจารย์ก็เป็นเอนทิตีที่ประกอบด้วยแอททริบิวต์ 4 ตัว คือรหัสอาจารย์ ชื่อ ที่อยู่ และเงินเดือน

ส่วนความสัมพันธ์นั้นหมายถึง ความสัมพันธ์ระหว่างเอนทิตีตัวอย่างเช่น ความสัมพันธ์ระหว่างแผนกวิชาและอาจารย์ ก็เป็นในลักษณะที่ว่าแผนกวิชาที่อาจารย์นั้นสังกัดอยู่ ส่วนความสัมพันธ์ระหว่างอาจารย์และแผนกวิชา เป็นในลักษณะที่ว่าอาจารย์ที่ทำงานอยู่กับแผนกวิชานั้นๆ ในรูปที่ 2.22 เป็นการแสดงความสัมพันธ์ระหว่างเอนทิตีทั้งสองนี้ โดยใช้หัวลูกศรแสดงความสัมพันธ์ ซึ่งในรูปนี้ความสัมพันธ์จากแผนกวิชาไปสู่อาจารย์เป็นความสัมพันธ์แบบหนึ่งต่อกลุ่ม (One-To-Many) กล่าวคือ ในแผนกวิชาหนึ่งแผนกจะมีอาจารย์สังกัดอยู่ได้มากกว่า 1 คน ในทางกลับกันความสัมพันธ์จากอาจารย์ไปแผนกวิชาเป็นลักษณะแบบกลุ่มต่อหนึ่ง (Many-To-One) ทั้งนี้ก็เพราะมีอาจารย์มากกว่า 1 คน ทำงานอยู่ในแผนกวิชาหนึ่งแผนกขอให้สังเกตหัวลูกศรที่แสดงความสัมพันธ์แบบนี้ ในรูปที่ 2.23 จะเห็นว่า หัวลูกศรเดียวแสดงความเป็น “หนึ่ง” และหัวลูกศรสองหัวแสดงความเป็น “กลุ่ม” คือ จากหนึ่งแผนกวิชาไปยังอาจารย์หลายคนหรือในบางครั้งเราอาจจะใช้หัวลูกศรเดียวแสดงความเป็นกลุ่มและไม่มีหัวลูกศรแสดงความเป็นหนึ่งก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 ตัวอย่างเอนทิตีและแอททริบิวต์

รูปที่ 2.22 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม โดยใช้หัวลูกศรคู่แสดงความเป็นกลุ่ม

จากรูปที่ 2.22 จะเห็นได้ว่า กรอบสี่เหลี่ยมแสดงถึงชนิดของเรคคอร์ดในฐานข้อมูลซึ่งจะมี 1 อันต่อ 1 เอนทิตีเครื่องหมายลูกศรแสดงถึงความสัมพันธ์ ซึ่งในรูปนี้แสดงความสัมพันธ์แบบหนึ่งต่อกลุ่มโดยหัวลูกศรจะออกจากส่วนของ “หนึ่ง” ไปยังส่วนของ “กลุ่ม” จะเห็นว่าแผนกวิชามีความสัมพันธ์กับอาจารย์ โดยที่ว่า ในหนึ่งแผนกวิชาอาจมีจำนวนอาจารย์ที่สังกัดได้มากกว่าหนึ่งคน วิธีการค้นหาข้อมูลสามารถทำได้โดยการใช้หัวลูกศรในการเชื่อมโยงความสัมพันธ์ ซึ่งบางครั้งอาจจะวิ่งทวนหัวลูกศรก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10.3 รีเลชั่น (Relation)

รีเลชั่น คือ ตาราง 2 มิติ ที่มีลักษณะดังนี้

1) แต่ละช่องของตารางจะบรรจุข้อมูลเพียงค่าเดียว

2) ชื่อหัวข้อในแต่ละคอลัมน์มีความแตกต่างกัน ได้แก่ ชื่อของแอททริบิวต์

3) ค่าของข้อมูลที่อยู่ในแต่ละคอลัมน์ ได้แก่ ค่าของแอททริบิวต์ที่ระบุไว้ในหัวข้อคอลัมน์นั้นๆ

4) การเรียงลำดับคอลัมน์ไม่ถือว่ามีความสำคัญ

5) ข้อมูลในแต่ละแถวจะต้องแตกต่างกัน

6) การเรียงลำดับแถวไม่ถือว่ามีความสำคัญ

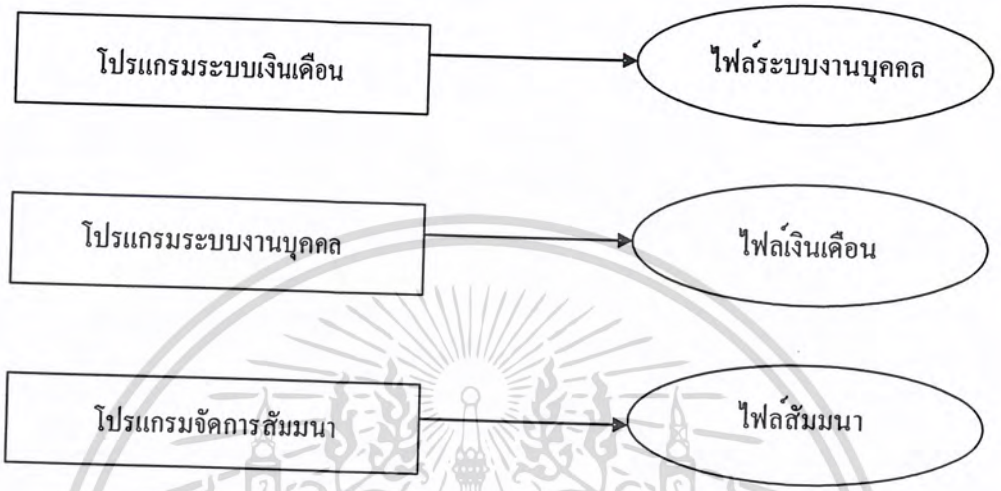
รูปที่ 2.23 ความสัมพันธ์แบบหนึ่งต่อกลุ่ม โดยใช้หัวลูกศรเดียวแสดงความเป็นกลุ่ม

2.11 ฐานข้อมูล (Database)

ในการประมวลผลไฟล์ทั่วๆ ไปนั้น ผู้ใช้แต่ละคนจะมีไฟล์เฉพาะงาน และโปรแกรมที่เขียนขึ้นมา เพื่อดึงเอาข้อมูลจากไฟล์ส่วนตัวมาใช้งานตามต้องการดังรูปที่ 2.24 การที่ผู้ใช้แต่ละกลุ่มต่างก็เก็บข้อมูลไว้ในไฟล์เช่นนี้ ก่อให้เกิดปัญหาที่เด่นชัด นั้นก็คือเกิดการซ้ำซ้อนของข้อมูลขึ้น เช่นที่อยู่อาจารย์แต่ละคนจะปรากฏซ้ำๆ กันอยู่หลายๆ ไฟล์ ความซับซ้อนเหล่านี้ นอกจากจะทำให้เกิดความสิ้นเปลืองเนื้อที่ในการจัดเก็บข้อมูลแล้ว ยังก่อให้เกิดปัญหาขึ้นเมื่อมีการเปลี่ยนแปลงค่าข้อมูล เพราะอาจจะหลงลืมเปลี่ยนแปลงข้อมูลในบางไฟล์ ทำให้ค่าข้อมูลเดียวกันที่เก็บในแต่ละไฟล์มีค่าไม่ตรงกัน ปัญหาดังกล่าว ทำให้เกิดความคิดที่น่าจะนำข้อมูลเหล่านี้มาเก็บรวบรวมไว้ในที่เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือจะเก็บไว้ในฐานข้อมูลแทนที่จะเก็บไว้ในไฟล์ ดังรูปที่ 2.25 แสดงรูปแบบการใช้ระบบฐานข้อมูล

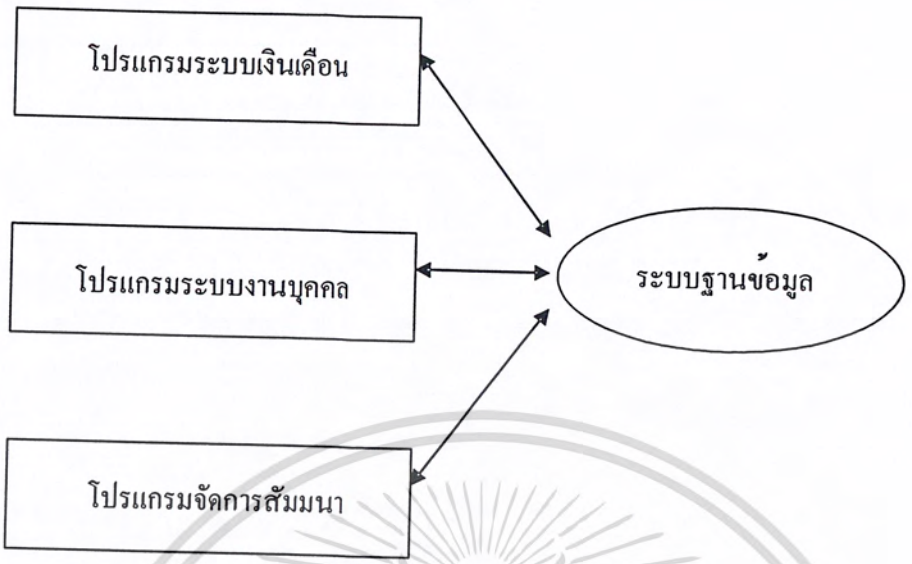


รูปที่ 2.24 การจัดระบบไฟล์

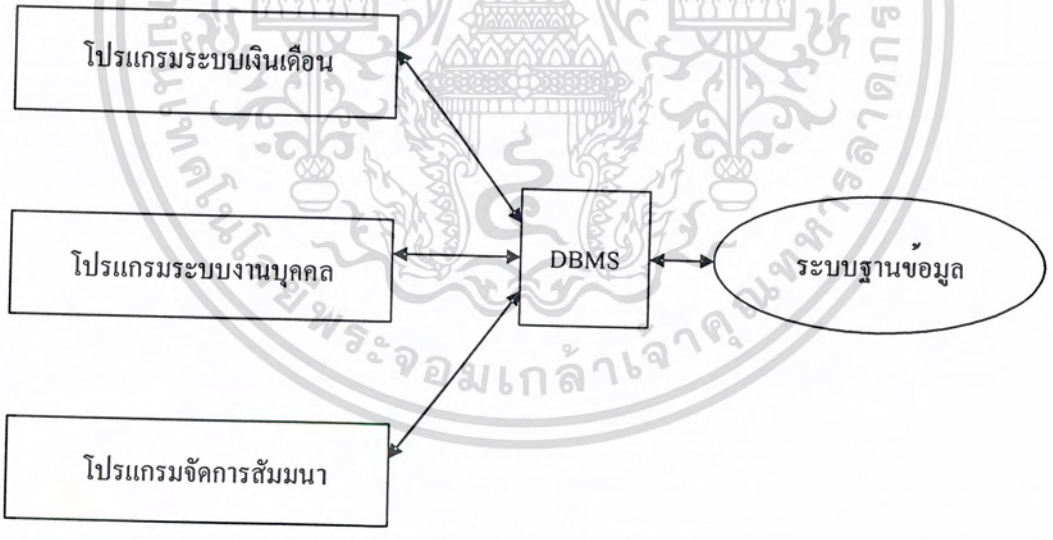
2.11.1 ระบบการจัดเก็บฐานข้อมูล

การควบคุมดูแลการใช้ฐานข้อมูลนั้นเป็นเรื่องที่ยุ่งยากกว่าการใช้ไฟล์มาก เพราะว่าเราจะต้องตัดสินใจว่าโครงสร้างในการเก็บข้อมูลควรจะเป็นเช่นไร และการเขียนโปรแกรมเพื่อสร้างและเรียกใช้ข้อมูลจากโครงสร้างไฟล์เหล่านี้ ก็เป็นเรื่องที่ยุ่งยากพอดู และถ้าเกิดโปรแกรมเหล่านี้ทำงานผิดพลาดขึ้นมา ก็ทำให้เกิดความเสียหายต่อโครงสร้างของข้อมูลทั้งหมดได้ เพื่อเป็นการลดภาระการทำงานของผู้ใช้ จึงมีซอฟต์แวร์ขึ้นมาตัวหนึ่งมีชื่อเรียกว่า ระบบการจัดการฐานข้อมูล หรือ DBMS ซึ่งจะทำหน้าที่ในการควบคุมดูแลการสร้างและเรียกใช้ฐานข้อมูล โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล เปรียบเสมือนว่า DBMS นี้จะเป็นสื่อกลางระหว่างผู้ใช้และโปรแกรมต่างๆ ที่เกี่ยวข้องกับการใช้ฐานข้อมูลดังรูปที่ 2.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.25 การใช้ระบบฐานข้อมูล



รูปที่ 2.26 การใช้ระบบฐานข้อมูลด้วย DBMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

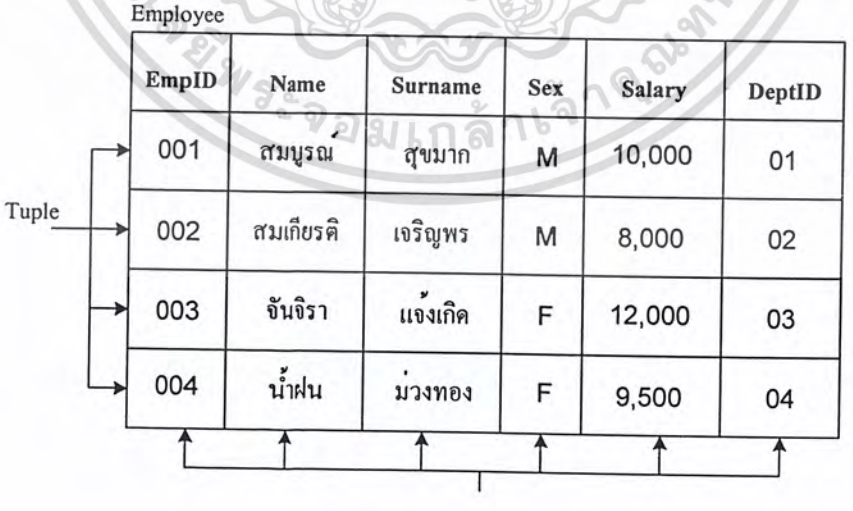
2.11.2 โมเดล (Model)

ประเภทของระบบการจัดการฐานข้อมูล แบ่งตามชนิดของโมเดลซึ่งโมเดลทุกชนิดจะประกอบไปด้วยองค์ประกอบ 2 อย่างคือ โครงสร้าง (Structure) และการใช้งาน (Operation)

โครงสร้างหมายถึง โครงสร้างของระบบข้อมูล ซึ่งอาจจะไม่ใช่โครงสร้างที่จัดเก็บจริงๆ ก็ได้แต่อย่างน้อยก็คือ โครงสร้างในแง่การมองของผู้ใช้ DBMS การใช้งาน คือ วิธีการที่จะให้ผู้ใช้สามารถเรียกดู และแก้ไขข้อมูลในระบบได้ โดยไม่ต้องรู้ว่าจริงๆ แล้วข้อมูลมีการจัดเก็บอย่างไร ในปัจจุบันมีโมเดลอยู่ 3 ชนิด

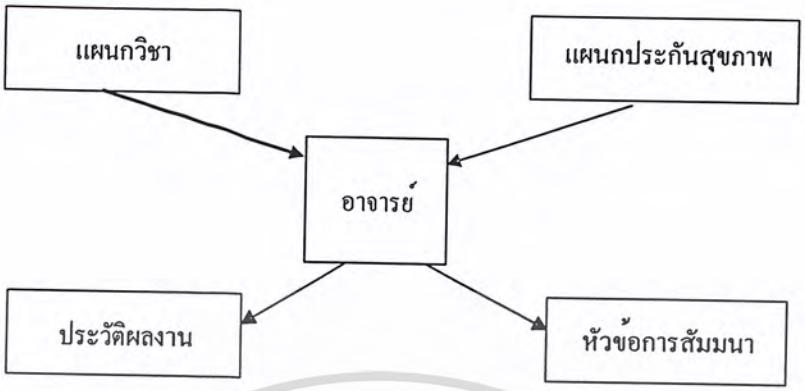
1) โมเดลเชิงสัมพันธ์ (Relational model) มีการเก็บข้อมูลเป็นแบบตาราง (Table) ซึ่งตารางนี้ก็คือ รีเลชั่น (Relation) มีลักษณะเป็น 2 มิติ คือ ด้านแถว (Row) และด้านคอลัมน์ (Column) ดังตัวอย่างที่แสดงในรูปที่ 2.27 เป็นฐานข้อมูลเชิงสัมพันธ์ที่ประกอบด้วยเอนทิตีต่างๆ ซึ่งข้อมูลของแต่ละเอนทิตีจะถูกจัดเก็บในลักษณะของตาราง Employee

2) โมเดลเน็ตเวิร์ค (Network model) โมเดลแบบนี้เป็นลักษณะของการรวบรวมเรคคอร์ดต่างๆและความสัมพันธ์ระหว่างเรคคอร์ด ความแตกต่างที่เห็นได้ชัดระหว่างโมเดลเชิงสัมพันธ์และแบบเน็ตเวิร์คคือ ในโมเดลเชิงสัมพันธ์จะแฝง (Implicit) การแสดงความสัมพันธ์เอาไว้ (หมายความว่าเรคคอร์ดที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในฟิลด์ใดฟิลด์หนึ่งเหมือนกัน) ส่วนการแสดงความสัมพันธ์ในโมเดลแบบเน็ตเวิร์คจะเป็นไปอย่างชัดเจน (Explicit) คือ การแสดงไว้ในโครงสร้างอย่างชัดเจนดังรูปที่ 2.28



รูปที่ 2.27 โครงสร้างฐานข้อมูลแบบเชิงสัมพันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 โครงสร้างฐานข้อมูลแบบเน็ตเวิร์ค

3) โมเดลแบบแตกสาขา (Hierarchical Model) ใช้อธิบายถึงฐานข้อมูลที่มีโครงสร้างของข้อมูลในแบบลำดับชั้น (Hierarchy) คิดค้นโดยบริษัท North American Rockwell เพื่อลดการซ้ำซ้อนของข้อมูล (Data Redundancy) มีการนำข้อมูลแต่ละส่วน (Part) มาจัดเก็บเป็นกลุ่ม ที่เรียกว่า Component แล้วรวมแต่ละกลุ่มเป็นกลุ่มใหญ่ เรียกว่า Final Component มีโครงสร้างในรูปแบบ Tree ที่เรียกว่า Upside-down Tree ต่อมาโครงสร้างในลักษณะนี้ได้ถูกเรียกว่าโครงสร้างแบบ Hierarchy โครงสร้างของโมเดลแบบแตกสาขา (Hierarchical Model) แสดงดังรูปที่ 2.29

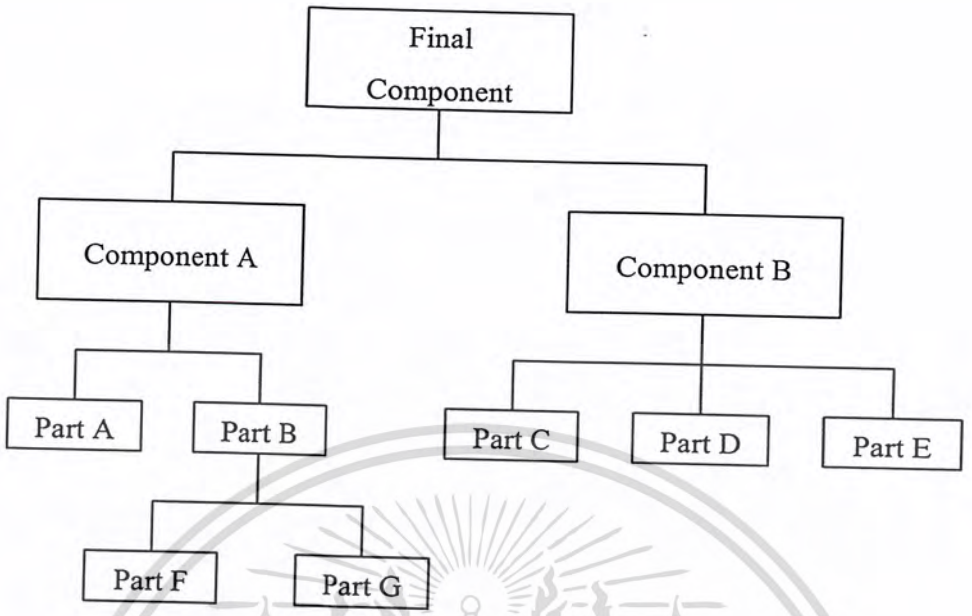
2.11.3 ประโยชน์จากการประมวลผลฐานข้อมูล

ประโยชน์จากการใช้ฐานข้อมูลในการประมวลผล สามารถสรุปได้ดังนี้

1) ลดความซ้ำซ้อนของข้อมูล การประมวลผลโดยใช้ระบบไฟล์นั้นจำเป็นที่ผู้ใช้แต่ละคนจะต้องมีไฟล์ส่วนตัวเอาไว้ ดังนั้นจึงเกิดเหตุการณ์ที่ข้อมูลชนิดเดียวกันถูกเก็บไว้หลายๆ แห่งหรือที่เรียกว่าความซ้ำซ้อน การนำข้อมูลทั้งหมดมาเก็บไว้ในที่เดียวกันในฐานข้อมูลนี้ เป็นการลดความซ้ำซ้อนของข้อมูลได้

2) สามารถหลีกเลี่ยงความขัดแย้งของข้อมูลได้ในระดับหนึ่งประโยชน์ในข้อนี้สืบเนื่องมาจากข้อที่แล้ว เพราะการเก็บข้อมูลไว้หลายๆแห่งอาจจะก่อให้เกิดปัญหาที่ว่า การแก้ไขข้อมูลเดียวกันทำให้ไม่เหมือนกันในทุกๆ แห่ง ทำให้ข้อมูลชุดเดียวกันอาจมีค่าไม่ตรงกัน ดังนั้นการใช้ระบบฐานข้อมูลทำให้เราลดความซ้ำซ้อนลงโดยมี DBMS เมื่อเกิดการแก้ไขข้อมูลขึ้น ก็จะแก้ไขให้เหมือนกันครบทุกแห่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.29 โครงสร้างของโมเดลแบบแตกสาขา

3) สามารถใช้ข้อมูลร่วมกันได้ ซึ่งการที่ใช้ข้อมูลร่วมกันได้นี้ ไม่ได้จำกัดอยู่เฉพาะโปรแกรมที่ใช้ข้อมูลอยู่ในปัจจุบันเท่านั้น แต่หมายความถึงโปรแกรมประยุกต์ที่พัฒนาขึ้นมาใหม่ด้วยที่สามารถใช้ข้อมูลที่มีอยู่ได้เลย โดยไม่จำเป็นต้องเพิ่มเติมข้อมูลเข้าไปในระบบอีก

4) สามารถควบคุมความเป็นมาตรฐานได้ จากการนำข้อมูลมารวมกันไว้ในฐานข้อมูล ทำให้ผู้ที่มีหน้าที่ควบคุมดูแลการใช้ระบบฐานข้อมูลสามารถกำหนดมาตรฐานของข้อมูลขึ้นมาได้ เช่น การใช้หน่วยวัดมาตรฐานที่เหมือนกัน รูปแบบการเขียนวันที่ให้เหมือนกันเป็นต้น การใช้มาตรฐานเดียวกันนี้ ทำให้การแลกเปลี่ยนข้อมูลระหว่างระบบเป็นไปอย่างสะดวกและถูกต้องเราเรียกผู้ควบคุมระบบว่า ผู้บริหารฐานข้อมูล (Data Base Administrator) หรือ DBA

5) สามารถจัดหาระบบความปลอดภัยที่ดีได้ คำว่า ระบบความปลอดภัยนี้ หมายถึง การป้องกันไม่ให้ผู้ที่ไม่มีสิทธิ์มาใช้ข้อมูลในระบบได้ เนื่องจาก DBA เป็นผู้ควบคุมดูแลการใช้ข้อมูลจึงสามารถกำหนดสิทธิ์การใช้ให้แก่ผู้ใช้คนใดๆ ก็ได้ตามความเหมาะสมและผู้ใช้แต่ละคนก็อาจจะใช้ข้อมูลได้ระดับที่ต่างกัน

6) สามารถควบคุมความคงสภาพของข้อมูลได้ โดยความไม่คงสภาพของข้อมูล คือการที่เกิดความขัดแย้งของข้อมูลดังที่ได้กล่าวไปแล้ว ซึ่งในกรณีของความขัดแย้งนี้จะเกิดขึ้นได้ก็เมื่อข้อมูลมีความซับซ้อนเท่านั้น แต่ในอีกแง่หนึ่งของความคงสภาพอาจเกิดขึ้นได้แม้ว่าจะไม่มีความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซ้ำซ้อน เช่น ข้อมูลเกี่ยวกับอายุของพนักงานของบริษัทอาจจะมีค่า 300 แทนที่จะเป็น 30 ซึ่งความผิดพลาดนี้เกิดขึ้นได้ง่ายๆ จากความผิดพลาดในการกรอกข้อมูลก็ได้ ในลักษณะของความไม่ถูกต้องนี้ ผู้ที่ออกแบบระบบฐานข้อมูลสามารถใส่กฎเกณฑ์ลงไปเพื่อควบคุมความคงสภาพไว้ เช่น ตามตัวอย่างนี้อาจจะใส่กฎว่า ค่าของอายุจะต้องเป็นตัวเลขระหว่าง 16 ถึง 60 เป็นต้น

7) สามารถสร้างความสมดุล ในความขัดแย้งของความต้องการได้ การที่ผู้ใช้ทั้งหมดขององค์กรใช้ข้อมูลจากฐานข้อมูลร่วมกันเช่นนี้ ทำให้ DBA ทราบถึงความต้องการและความสำคัญของผู้ใช้งานทั้งหมด จึงสามารถกำหนดโครงสร้างของฐานข้อมูลเพื่อการให้บริการที่ดีที่สุดได้ เช่น เลือกเก็บข้อมูลจะต้องใช้บ่อยๆ ไว้ในสื่อข้อมูลที่มีความเร็วเป็นพิเศษ เป็นต้น

8) เกิดความเป็นอิสระของข้อมูล วิธีการที่จะทำความเข้าใจ ความเป็นอิสระของข้อมูลนั้นคืออะไร ก็โดยดูในด้านตรงข้ามก่อนว่า ข้อมูลที่ไม่เป็นอิสระนั้นเป็นอย่างไรข้อมูลที่ไม่เป็นอิสระก็คือข้อมูลที่ถูกนำมาประยุกต์ใช้งานยังมีความผูกพันอยู่กับวิธีการจัดเก็บและเรียกใช้ข้อมูลซึ่งในลักษณะการเขียนโปรแกรมประยุกต์บางประเภท เราอาจจำเป็นต้องใส่เทคนิคการจัดเก็บ และเรียกใช้ข้อมูลไว้ในตัวโปรแกรมด้วย นั่นหมายความว่า ถ้าเกิดต้องมีการเปลี่ยนแปลงวิธีการจัดเก็บหรือเรียกใช้แล้วผู้ใช้จำเป็นที่จะต้องสร้างวิธีการประยุกต์ใช้ขึ้นมาใหม่ ซึ่งเป็นความไม่สะดวกอย่างยิ่งและทำให้เราหมดโอกาสที่จะปรับปรุงโครงสร้างของข้อมูล เพื่อให้ใช้งานได้อย่างมีประสิทธิภาพมากขึ้น

2.12 สถาปัตยกรรมของระบบฐานข้อมูล

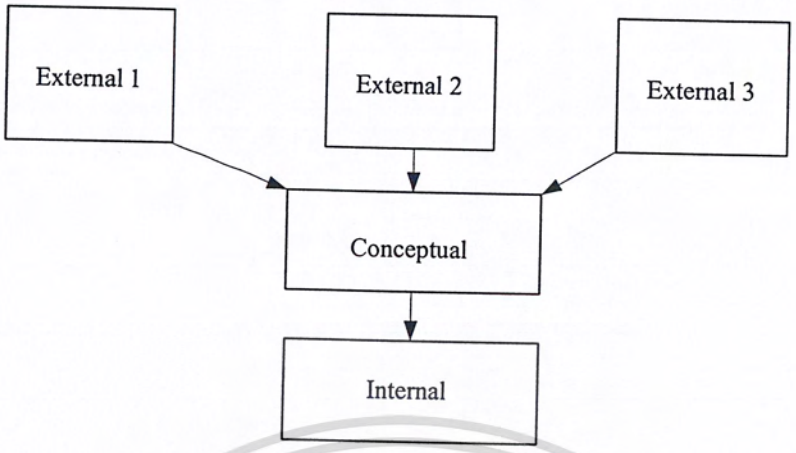
สถาปัตยกรรมของระบบฐานข้อมูล หมายถึง องค์ประกอบต่างๆ ของระบบฐานข้อมูลซึ่งสามารถแบ่งเป็นระดับ ได้ 3 ระดับดังรูปที่ 2.30 ซึ่งได้แก่

1) ระดับภายใน (Internal หรือ Physical level) เป็นระดับที่ต่ำที่สุด ได้แก่ ระดับของการจัดเก็บข้อมูลจริงๆ

2) ระดับหลักการ (Conceptual level) เป็นระดับที่อยู่ถัดลงมา ได้แก่ ระดับของการมอง Entity และความสัมพันธ์ระหว่าง Entity ทั้งหมดรวมทั้งกฎเกณฑ์ต่างๆ เกี่ยวกับข้อมูลและผู้มีสิทธิ์จะใช้ เป็นต้น

3) ระดับภายนอก (External หรือ View level) เป็นระดับที่อยู่สูงที่สุด เป็นข้อมูลที่จะมองเห็นจากการใช้งานของผู้ใช้แต่ละคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 ระดับของข้อมูล 3 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบ การสร้าง และการทำงาน

3.1 กล่าวนำ

การออกแบบและการสร้างเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยบัตรสมาร์ตการ์ด ได้แบ่งการทำงานออกเป็น 3 ส่วนใหญ่ๆ ด้วยกันคือ ส่วนที่ 1 เป็นส่วนของการออกแบบวงจรตรวจสอบและบันทึกการเข้าชั้นเรียนโดยใช้ไมโครคอนโทรลเลอร์ประมวลผล ส่วนที่ 2 เป็นส่วนที่ใช้โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดติดต่อกับฐานข้อมูลนักศึกษาในเครื่องคอมพิวเตอร์ และส่วนที่ 3 เป็นส่วนของการออกแบบโปรแกรมควบคุมในส่วนต่างๆ

ส่วนที่ 1 เป็นส่วนของการออกแบบวงจรตรวจสอบและบันทึกการเข้าชั้นเรียนโดยใช้ไมโครคอนโทรลเลอร์ประมวลผล ในการตรวจสอบจะใช้เครื่องอ่านบาร์โค้ดอ่านข้อมูลจากบัตรประจำตัวนักศึกษาและใช้สมาร์ตการ์ดในการเก็บบันทึกข้อมูลการเข้าชั้นเรียน โดยแสดงผลของรหัสนักศึกษา วันเดือนปีและเวลาการเข้าชั้นเรียนทางจอแอลซีดี

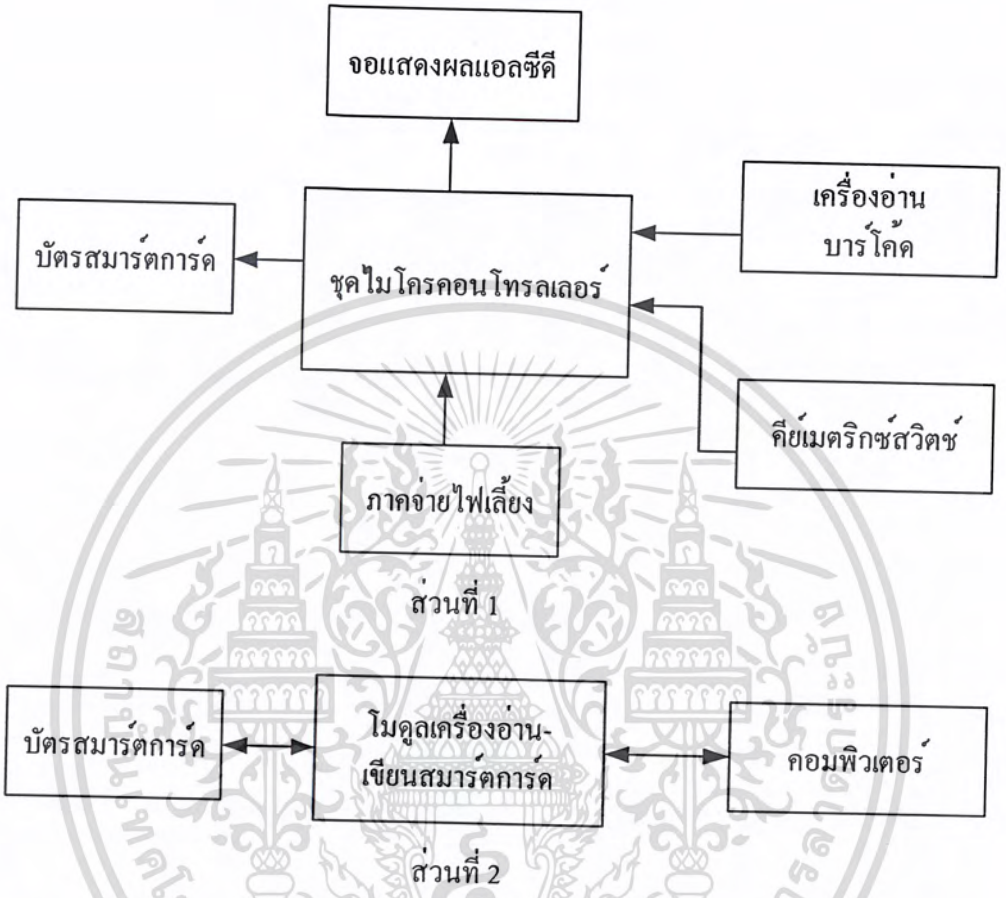
เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดประกอบด้วย วงจรประมวลผลการเขียนสมาร์ตการ์ด วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด วงจรรับข้อมูลจากคีย์เมตริกซ์สวิตช์ วงจรแสดงผลทางจอแอลซีดีและวงจรภาคจ่ายไฟเลี้ยง

ส่วนที่ 2 เป็นส่วนที่ใช้โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดติดต่อกับฐานข้อมูลนักศึกษาในเครื่องคอมพิวเตอร์ โดยนำสมาร์ตการ์ดที่เก็บบันทึกข้อมูลนักศึกษาแต่ละคนไว้มาประมวลผลร่วมกับฐานข้อมูลนักศึกษาในเครื่องคอมพิวเตอร์เพื่อให้อาจารย์ผู้สอนใช้ในการตรวจสอบและแก้ไขข้อมูลนักศึกษา โครงสร้างการทำงานทั้งหมดของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดแบ่งออกเป็น 2 ส่วน ซึ่งมีโครงสร้างการทำงานดังรูปที่ 3.1

ส่วนที่ 3 เป็นส่วนของการออกแบบโปรแกรมควบคุมส่วนต่างๆ ซึ่งประกอบไปด้วยโปรแกรมรับข้อมูลจากเครื่องอ่านบาร์โค้ด โปรแกรมรับข้อมูลจากคีย์เมตริกซ์สวิตช์ โปรแกรมแสดงผลแอลซีดีเพื่อใช้ในการเก็บบันทึกข้อมูลลงสมาร์ตการ์ด ซึ่งเขียนโดยใช้ภาษาแอสเซมบลี และในส่วนของการเขียนโปรแกรมติดต่อระหว่างโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดกับฐานข้อมูลนักศึกษาในเครื่องคอมพิวเตอร์ เพื่อใช้ในการตรวจสอบและแก้ไขข้อมูลนักศึกษาโดยใช้โปรแกรมไมโครซอฟท์แอคเซสและโปรแกรมวิชวลเบสิกในการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 แผนผังการทำงาน



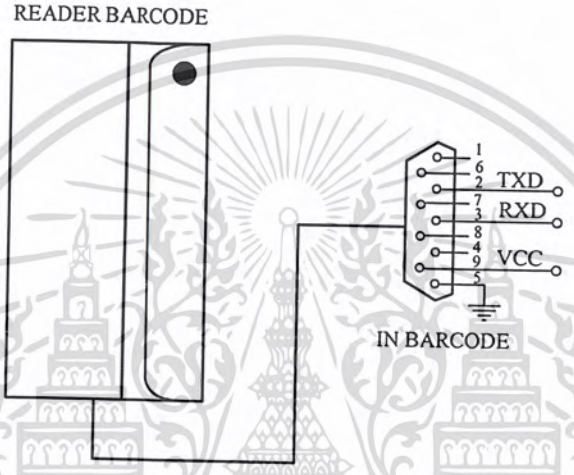
รูปที่ 3.1 แผนผังการทำงานเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

จากรูปที่ 3.1 เป็นแผนผังการทำงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด ในการออกแบบวงจรของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดจะแบ่งออกเป็นส่วนของวงจรต่างๆ ได้ดังนี้ คือ วงจรประมวลผลการเขียนสมาร์ตการ์ด วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด วงจรรับข้อมูลจากคีย์เมตริกซ์สวิตช์ วงจรแสดงผลทางจอแอลซีดี วงจรภาคจ่ายไฟเลี้ยงและโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด ซึ่งมีขั้นตอนการออกแบบวงจรและการทำงานของวงจรดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด

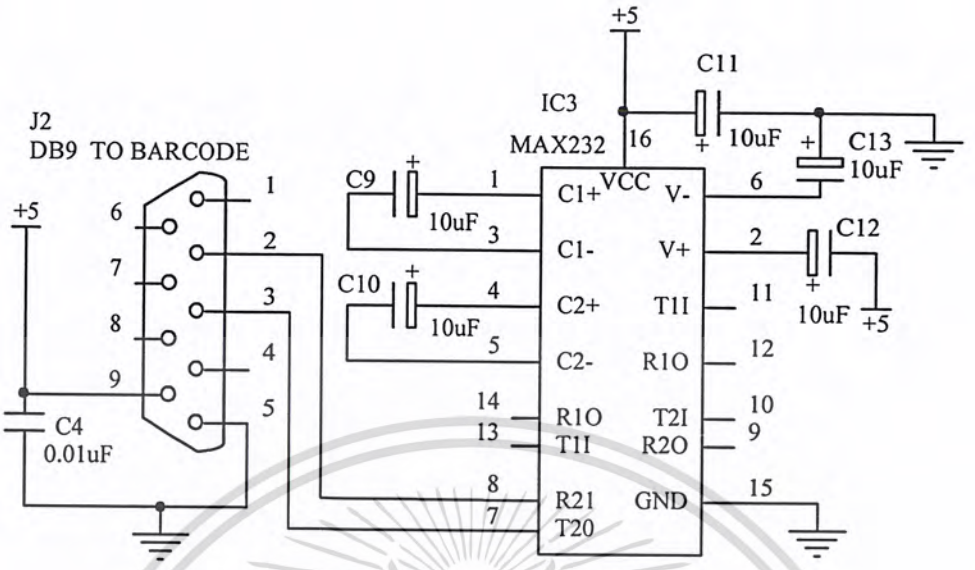
ในส่วนของวงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด ทำหน้าที่ในการรับข้อมูลจากการรูดบัตรประจำตัวนักศึกษาแต่ละคน จากนั้นทำการส่งข้อมูลให้กับขา TX และ RX ของ IC MAX232 ซึ่งเป็นตัวกลางในการติดต่อระหว่างเครื่องอ่านบาร์โค้ดกับไมโครคอนโทรลเลอร์สัญญาณที่ได้จะถูกส่งต่อให้กับพอร์ตอินพุตของไอซี AT89S8252 เพื่อนำไปประมวลผลและแสดงผลทางจอแอลซีดี แสดงการต่อวงจรเครื่องอ่านบาร์โค้ดดังรูปที่ 3.2



รูปที่ 3.2 การต่อวงจรเครื่องอ่านบาร์โค้ด

ในการทำงานของวงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ดจะนำไปต่อร่วมกับไอซี MAX232 และวงจรไมโครคอนโทรลเลอร์ ซึ่งการจัดเก็บรหัสประจำตัวนักศึกษาจะถูกเก็บไว้ในสมาร์ตการ์ดทุกครั้งที่มีการรูดบัตร นอกจากรหัสประจำตัวนักศึกษาแล้วในสมาร์ตการ์ดยังเก็บวันเดือนปีและเวลาที่เข้าเรียนของนักศึกษาแต่ละคนด้วย โดยที่ข้อมูลจะส่งไปยังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ AT89S8252 ที่ขา RX0 ข้อมูลที่ได้ทั้งหมดจะอยู่ในรีจีสเตอร์ SBUF รหัสของเครื่องอ่านบาร์โค้ดจะแปลงเป็นรหัสแอสกี้ และนำไปประมวลผลต่อไป วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ดแสดงดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด

3.2.2 วงจรรับข้อมูลคีย์เมตริกซ์สวิตช์

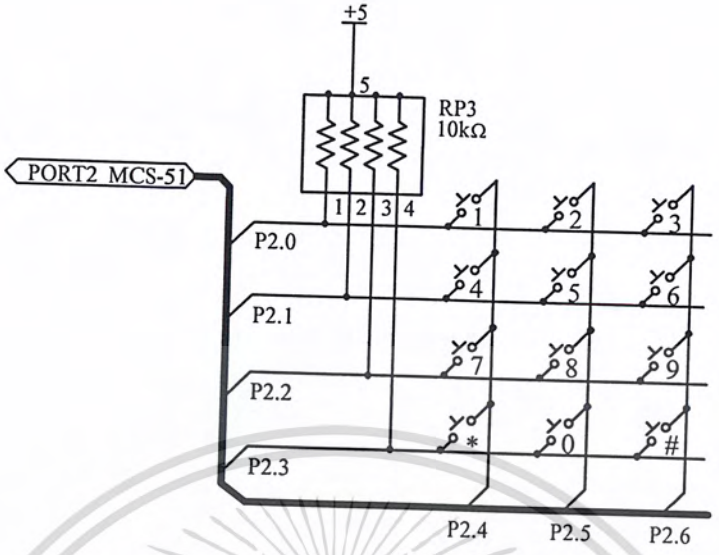
ทำหน้าที่ในการส่งข้อมูลจากการกดตัวเลขเพื่อใช้สำหรับการตั้งวันเดือนปีและเวลา ให้ถูกต้อง ถ้ากรณีที่ข้อมูลของวันเดือนปีหรือเวลาไม่เป็นปัจจุบันผู้ใช้งานสามารถเลือกได้ว่าจะตั้งวันเดือนปีหรือเวลาใหม่หรือไม่ การทำงานของวงจรคีย์เมตริกซ์สวิตช์จะเริ่มเมื่อมีการกดคีย์ สัญญาณจากการกดจะต่อเข้าที่พอร์ต 2 ของไมโครคอนโทรลเลอร์ใช้การสแกนคีย์ตรวจสอบสัญญาณแล้วนำไปแสดงผลที่จอแอลซีดี วงจรคีย์เมตริกซ์สวิตช์แสดงดังรูปที่ 3.4

3.2.3 วงจรแสดงผลจอแอลซีดี

ในส่วนนี้จะทำหน้าที่ในการแสดงผลการทำงานโดยใช้จอแอลซีดี แบบ (Dot Matrix) ขนาด 16 ตัวอักษร 2 บรรทัดทำหน้าที่ในการแสดงผลข้อมูลที่ได้จากการประมวลผลของวงจรไมโครคอนโทรลเลอร์ที่รับข้อมูลจากเครื่องอ่านบาร์โค้ดและจากวงจรคีย์เมตริกซ์สวิตช์ โดยที่จอแสดงผลแอลซีดีจะแสดงรหัสประจำตัวนักศึกษาของนักศึกษาแต่ละคนวันเดือนปีและเวลา

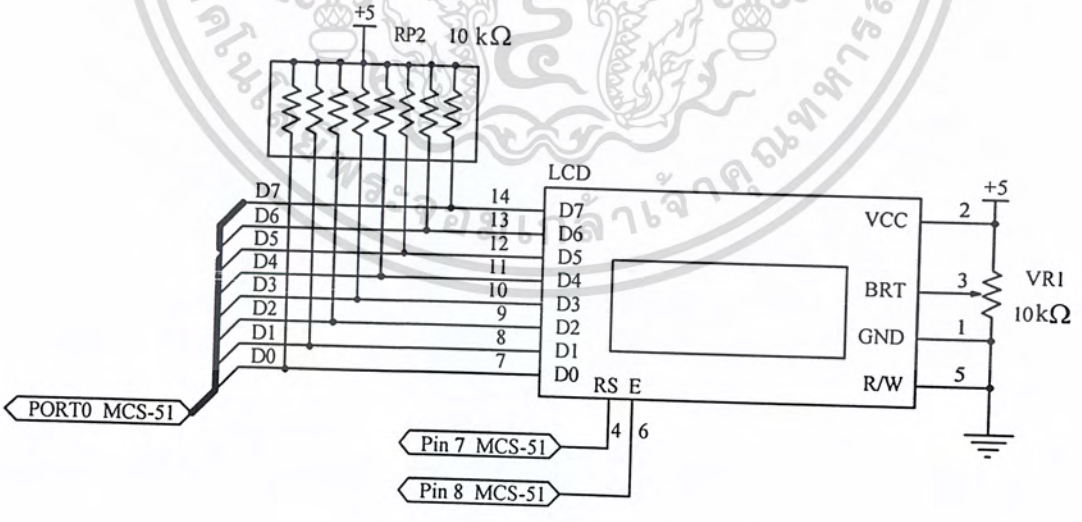
การทำงานของวงจรแสดงผลทางจอแอลซีดี เมื่อทำการรูดบัตรนักศึกษาด้วยเครื่องอ่านบาร์โค้ดข้อมูลจะถูกส่งค่าไว้ในรีจีสเตอร์ SBUF ของไมโครคอนโทรลเลอร์ AT89S8252 ทำให้รีจีสเตอร์ RI มีค่าเป็น “1” เมื่อข้อมูลถูกส่งมาเรียบร้อยแล้วจะนำค่าของข้อมูลใน SBUF เก็บไว้ในตำแหน่งหน่วยความจำเนื่องจากรหัสนักศึกษามีทั้งหมด 8 ตัวต้องทำการข้างต้นทั้งหมด 8 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรคีย์เมตริกซ์สวิตช์

การแสดงผลทางจอแอลซีดี เมื่อทำการกดคีย์เมตริกซ์ใช้การสแกนวนลูปในการตรวจสอบว่าข้อมูลที่ส่งมาเป็นเลขอะไร เมื่อมีการกดคีย์เมตริกซ์สวิตช์ไมโครคอนโทรลเลอร์ AT89S8252 จะทำการสแกนข้อมูลที่พอร์ต 2 จากวงจรคีย์เมตริกซ์สวิตช์ ข้อมูลที่ออกมาจะเป็นเลขฐาน 16 แปลงเป็นรหัสแอสกี้จากนั้นจึงนำไปแสดงผลทางจอแอลซีดี วงจรแสดงผลจอแอลซีดีแสดงดังรูปที่ 3.5

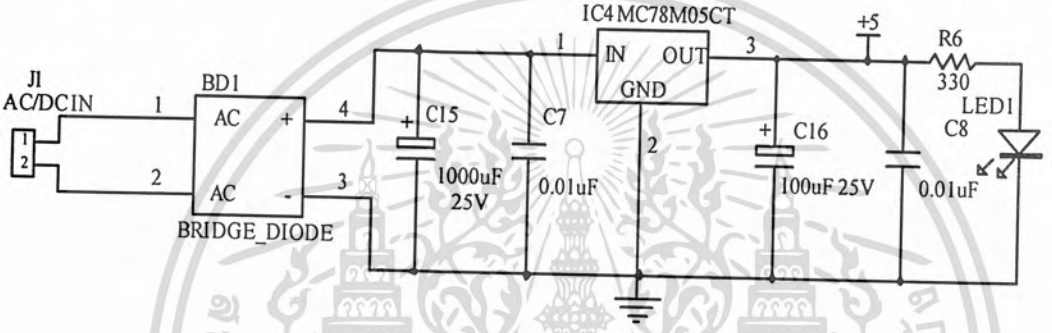


รูปที่ 3.5 วงจรแสดงผลจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 วงจรภาคจ่ายไฟเลี้ยง

เนื่องจากอุปกรณ์ที่ใช้ในการประกอบวงจรโครงงานนี้ ส่วนใหญ่เป็นที่ใช้ไฟเลี้ยงไม่เกิน 5 โวลต์ และกินกระแสในการทำงานต่ำ ดังนั้นจึงออกแบบวงจรการจ่ายไฟเลี้ยงโดยเลือกใช้ไอซีเร็กกูเลเตอร์ขนาด 5 โวลต์ ชนิดไฟบวกเบอร์ 7805 ที่สามารถจ่ายแรงดันคงที่ 5 โวลต์และกระแสไม่เกิน 1 แอมแปร์ ไดโอดที่ใช้เป็นแบบโมดูลบริดจ์ไดโอดต่อร่วมกับตัวเก็บประจุเพื่อใช้ในการกรองกระแสไฟให้เรียบ โดยมีไดโอดเปล่งแสงในการแสดงผลการทำงาน วงจรภาคจ่ายไฟเลี้ยงแสดงดังรูปที่ 3.6



รูปที่ 3.6 วงจรภาคจ่ายไฟเลี้ยง

3.2.5 โมดูลเครื่องอ่าน–เขียนสมาร์ตการ์ด

WACR20S คือชื่อรุ่นของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด ที่นำมาใช้ติดต่อการทำงานระหว่างสมาร์ตการ์ดกับเครื่องคอมพิวเตอร์มีขนาดเล็กกะทัดรัดสามารถใช้งานกับสมาร์ตการ์ดที่มีหน่วยความจำได้หลายแบบและสมาร์ตการ์ดแบบมีหน่วยประมวลผลกลาง ชนิดโปรโตคอล T = 0 และ T = 1 WACR20S สามารถนำไปติดตั้งที่คอมพิวเตอร์ในช่องของฟลอปปี้ดิสก์ไดร์ฟได้อย่างสะดวก

1) มุมมองมาตรฐานของ WACR20S

1.1) ติดต่อกับพอร์ตอนุกรม RS-232

1.2) สามารถติดตั้งในช่องฟลอปปี้ดิสก์ไดร์ฟของเครื่องคอมพิวเตอร์ได้

1.3) ใช้มาตรฐาน ISO 7816 -1/2/3 และ PC/SC Compliant

1.4) สามารถอ่านและเขียนสมาร์ตการ์ดแบบมีหน่วยประมวลผลกลาง ชนิดโปรโตคอล T = 0 และ T = 1

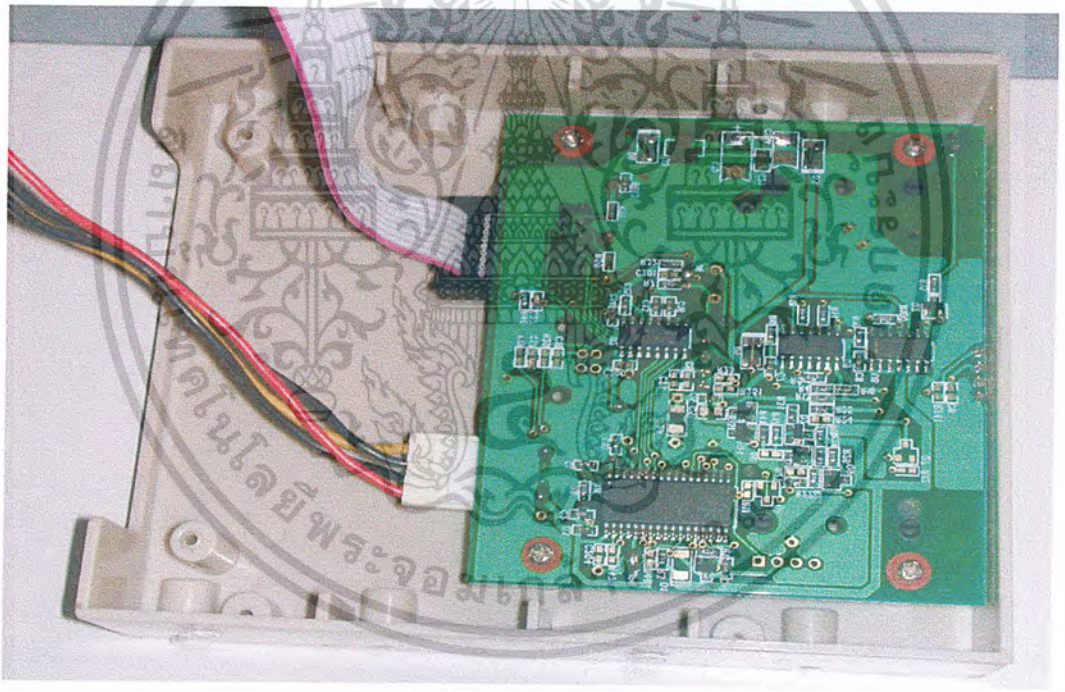
1.5) สามารถอ่านและเขียนสมาร์ตการ์ดแบบหน่วยความจำ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Siemens : SLE4406, SLE4404, SLE4418, SLE4428, SLE4442, SLE5536, SLE4436
- Atmel : AT88SC06, AT24C01-16, AT88SC101, AT88SC102, AT88SC1604
- STMicroelectronics : ST1305, ST14C02C, ST14C04C, ST1333, ST135
- Xicor : X76F041, X24645, X76F128, X76F640, X76F100
- Microchip : XIIC, 24C65, 24C01SC, 24C02SC

1.6) สามารถใช้ได้กับระบบปฏิบัติการ Windows 95, 98, ME, NT, 2000 และ Windows XP

1.7) การอินเตอร์เฟส 32 Bit DLL files สำหรับ Windows 95, 98, ME, NT, 2000 และ Windows XP



รูปที่ 3.7 โมดูลเครื่องอ่าน- เขียนสมาร์ตการ์ด WACR20S

2) คุณสมบัติเฉพาะของ WACR20S

2.1) อินเตอร์เฟสแบบ RS-232

2.2) ใช้แรงดันไฟเลี้ยง 5V/DC กระแสน้อยกว่า 100 mA (ไม่มีสมาร์ตการ์ดเสียบอยู่)

2.3) อุณหภูมิในการทำงาน 0-50 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4) ความถี่ที่ป้อนให้สมาร์ตการ์ด 3.068 MHz

2.5) ใช้มาตรฐานสมาร์ตการ์ดแบบ ISO 7816-1/2/3, PC/SC

3.3 การออกแบบและการทำงานของโปรแกรม

ในการออกแบบโปรแกรมเพื่อควบคุมการทำงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยบัตรสมาร์ตการ์ด ได้เลือกใช้ตัวประมวลผลตระกูลไมโครคอนโทรลเลอร์ MCS-51 เบอร์ AT89S8252 ในการทำงานและการออกแบบอีกส่วนก็คือ โปรแกรมฐานข้อมูลของนักศึกษาเพื่อใช้ในการเปลี่ยนแปลงแก้ไขเพิ่มเติมข้อมูลของนักศึกษาซึ่งออกแบบการสร้างมาจากโปรแกรมไมโครซอฟท์แอคเซสและใช้โปรแกรมวิชวลเบสิก ในการสร้างรูปแบบฟอร์มหน้าต่างๆ ของโปรแกรมเพื่อใช้ในการทำงานกับโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดและฐานข้อมูลของนักศึกษา

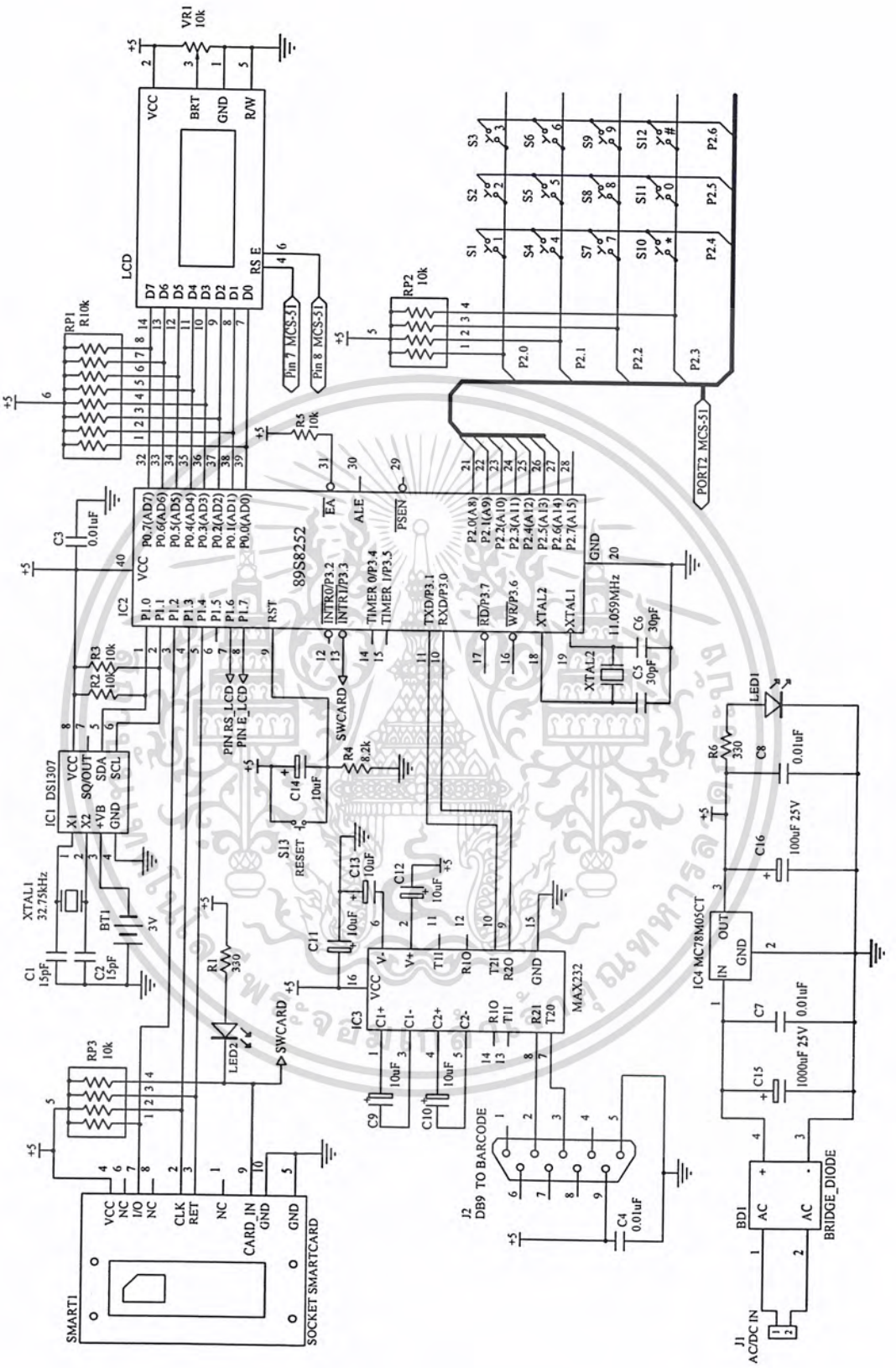
3.3.1 การออกแบบโปรแกรมของฮาร์ดแวร์

ในการออกแบบโปรแกรมของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด นี้ได้แบ่งออกเป็นส่วนๆ ได้ดังนี้คือ ส่วนของการรับข้อมูลจากเครื่องอ่านบาร์โค้ด การรับข้อมูลจากคีย์เมตริกซ์สวิตช์ การแสดงผลทางจอแอลซีดีและการเขียนข้อมูลลงสมาร์ตการ์ด ในส่วนของโปรแกรม แสดงดังในภาคผนวก ง.

3.3.2 การออกแบบโปรแกรมวิชวลเบสิก

ในการออกแบบโปรแกรมวิชวลเบสิก โปรแกรมที่ออกแบบมีความสามารถ คือ การเลือกพอร์ตอนุกรมในการทำงานของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด การจัดเก็บข้อมูล การค้นหารายชื่อนักศึกษา การลบข้อมูลนักศึกษาและสามารถพิมพ์ข้อมูลของนักศึกษาที่มาเข้าเรียน มาสายและขาดเรียนได้ โดยข้อมูลนักศึกษาประกอบด้วย รหัสนักศึกษา ชื่อ นามสกุล สาขาวิชา ภาควิชาและคณะที่ศึกษา ข้อมูลทั้งหมดนี้มีไว้เพื่อตรวจสอบรายชื่อนักศึกษาที่มาเข้าเรียน มาสายหรือขาดเรียน ส่วนของโปรแกรม แสดงดังในภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 กล่าวนำ

ปริญญานิพนธ์บทนี้จะกล่าวถึงวิธีการทดลองและผลการทดลองที่ได้ ซึ่งประกอบด้วยการทดลองการทำงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดและการทดลองการทำงานของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดที่ใช้ติดต่อกับฐานข้อมูลในเครื่องคอมพิวเตอร์

สำหรับการทดลองการทำงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด หลักการทำงานของเครื่องจะใช้ไมโครคอนโทรลเลอร์ เบอร์ AT89S8252 ในการควบคุมการทำงานในส่วนต่างๆ คือ การรับข้อมูลจากคีย์เมตริกซ์สวิตช์ การแสดงผลทางจอแอลซีดี การตั้งวันเดือนปีและเวลาที่เข้าเรียนเพื่อนำข้อมูลที่ได้บันทึกลงในสมาร์ตการ์ด และส่วนที่ 2 คือ การทดลองการทำงานของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดที่ใช้ติดต่อกับฐานข้อมูลในเครื่องคอมพิวเตอร์ หลักการทำงานของโปรแกรมเขียนจากโปรแกรมวิชวลเบสิกในการควบคุมการทำงานของโปรแกรมในส่วนต่างๆ คือ การติดต่อกับโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดเพื่อโหลดข้อมูลจากสมาร์ตการ์ด การเพิ่มและแก้ไขข้อมูลนักศึกษาในฐานข้อมูล การพิมพ์ข้อมูลการเข้าชั้นเรียน การเก็บข้อมูลการเข้าชั้นเรียน

4.2 การทดลองเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

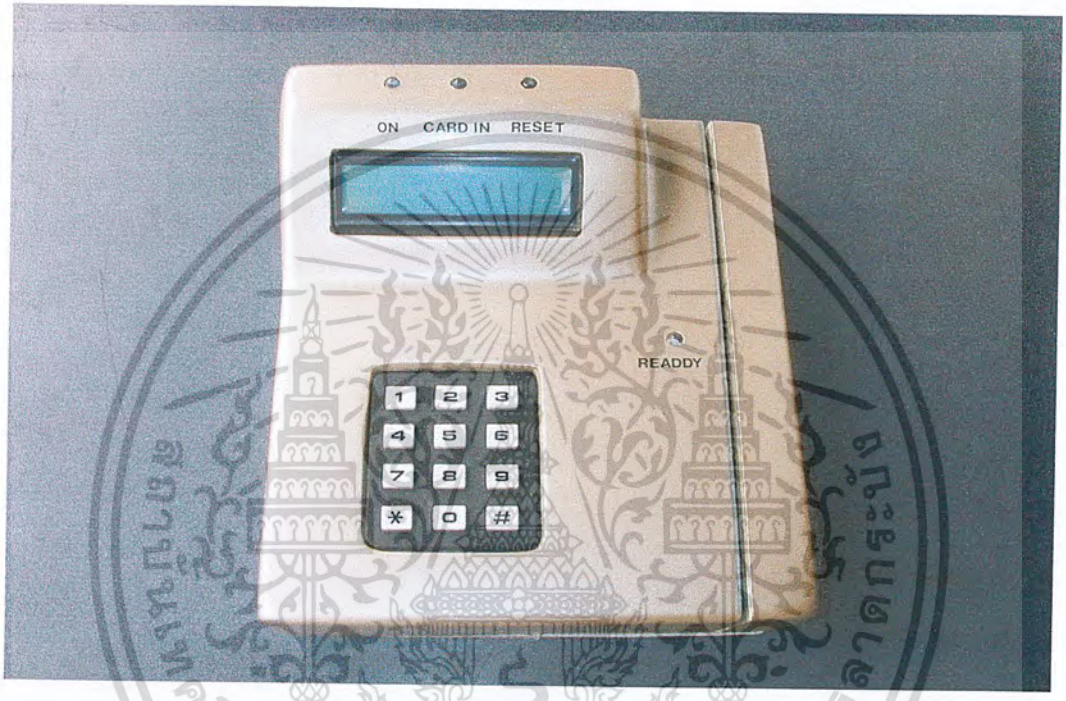
การทำงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด เกิดจากการทำงานร่วมกันของวงจรต่างๆ คือ วงจรประมวลผลการเขียนสมาร์ตการ์ด วงจรรับข้อมูลจากเครื่องอ่านบาร์โค้ด วงจรรับข้อมูลจากคีย์เมตริกซ์สวิตช์ วงจรแสดงผลทางจอแอลซีดีและวงจรภาคจ่ายไฟเลี้ยง ประกอบเข้าด้วยกันโดยมีลำดับขั้นตอนการทดลองในส่วนต่างๆ ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1 การทดลองการตรวจสอบและบันทึกข้อมูลลงสมาร์ตการ์ด

1) ทำการป้อนไฟเลี้ยงจากอะแด็ปเตอร์เข้าให้กับเครื่อง หลอดแอลอีดีสีแดงจะติด

2) เครื่องจะอยู่ในสภาวะการรอเสียบสมาร์ตการ์ด เมื่อมีการเสียบสมาร์ตการ์ดแอลอีดีสีเขียวจะติด



รูปที่ 4.1 ตัวเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

3) ทำการทดลองการตรวจสอบและบันทึกข้อมูลลงสมาร์ตการ์ด

ในการทดลองให้เสียบสมาร์ตการ์ดเข้าที่ช่องใส่สมาร์ตการ์ด ทำการทดลองการรับข้อมูลจากคีย์เมตริกซ์สวิตช์ โดยจะมีการป้อนข้อมูล 2 ชุด คือ การป้อนเลขวันเดือนปีและการป้อนเลขค่าเวลา ซึ่งการป้อนข้อมูลทั้งสองจะใช้ตอนที่ต้องการทดลองป้อนค่าข้อมูลใหม่ในกรณีที่วันเดือนปีหรือเวลาไม่เป็นปัจจุบัน โดยจะมีการให้เลือกคำตอบว่าต้องการตั้งค่าเวลาและวันเดือนปีใหม่หรือไม่

3.1) หลังจากเสียบสมาร์ตการ์ดแล้ว ทดลองการตั้งค่าวันเดือนปีหรือไม่ ถ้าต้องการให้กด "*" เพื่อเป็นการยืนยัน ถ้าไม่ต้องการให้กด "#" เพื่อเป็นการยกเลิก ถ้าเลือกตกลงจะต้องทดลองไปตามข้อ 3.2 และข้อ 3.3 แต่ถ้ายกเลิกโปรแกรมก็จะพร้อมรับการรูดบัตรประจำนักศึกษาหน้าจอแอลซีดีแสดงข้อความ ID_No. : ดังรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

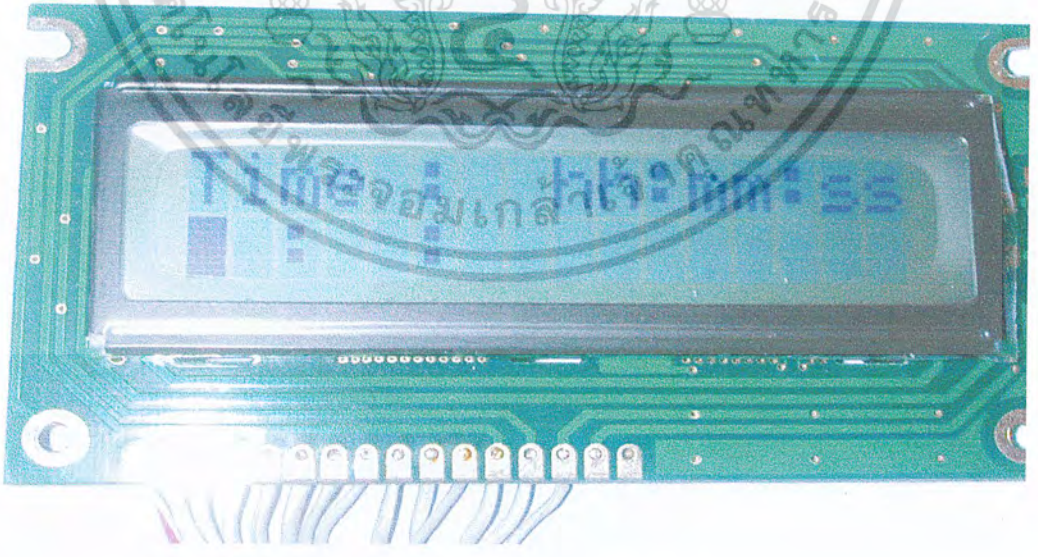
3.2) โปรแกรมจะถามว่าต้องการตั้งค่าวันเดือนปีหรือไม่ ถ้าต้องการให้กดปุ่ม "*" เมื่อต้องการตั้งค่า หรือถ้าไม่ต้องการให้กดปุ่ม "#" เมื่อไม่ต้องการตั้งค่า จากนั้นโปรแกรมจะถามว่าค่าที่ตั้งไปนั้นถูกต้องหรือไม่ ถ้าถูกต้องให้กดปุ่ม "*" เพื่อเป็นการยืนยัน หรือกดปุ่ม "#" เพื่อเป็นการยกเลิก ดังรูปที่ 4.3

3.3) เสร็จจากการตั้งวันเดือนปีแล้ว โปรแกรมจะถามว่าต้องการตั้งเวลาต่อ ถ้าต้องการกด "*" คือ ใช่ หรือกด "#" คือ ไม่ใช่ จากนั้นโปรแกรมจะถามว่าค่าที่ตั้งไปนั้นถูกต้องหรือไม่ ถ้าถูกต้องให้กด "*" เพื่อเป็นการยืนยัน และกด "#" เพื่อเป็นการยกเลิก ดังรูปที่ 4.4

รูปที่ 4.2 หน้าจอแอลซีดีรอการรูดบัตรประจำตัวนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.3 หน้าจอแอลซีดีเมื่อต้องการตั้งค่าวันเดือนปี



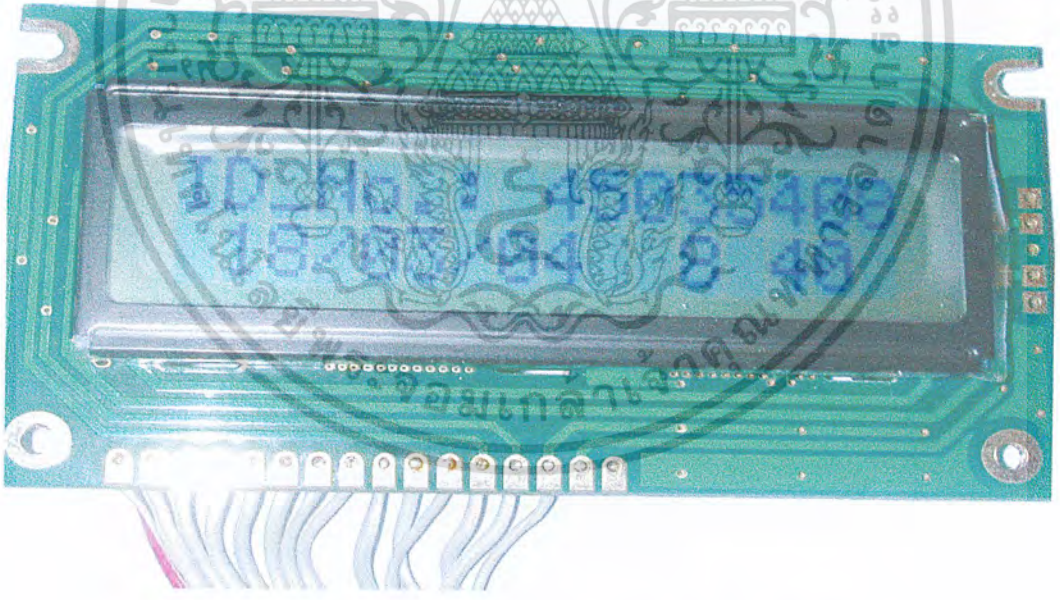
รูปที่ 4.4 หน้าจอแอลซีดีเมื่อต้องการตั้งค่าเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 การทดลองการรับข้อมูลจากเครื่องอ่านบาร์โค้ด

หลังจากเสียบสมาร์ตการ์ดและตั้งค่าวัน เดือนปีและเวลาแล้ว หน้าจอแอลซีดีจะอยู่ในสภาวะพร้อมที่จะรับข้อมูลจากเครื่องอ่านบาร์โค้ด ซึ่งสภาวะการรอรับข้อมูลนี้ดูได้จากรูปที่ 4.2 ซึ่งตอนนี้เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดพร้อมที่จะรับข้อมูลเพื่อนำไปเขียนเก็บในสมาร์ตการ์ด เมื่อมีการรูดบัตรประจำตัวนักศึกษาคนแรก ข้อมูลจะถูกเขียนลงไปคือ วัน เดือน ปี เวลาและรหัสประจำตัวนักศึกษา เนื่องจากต้องการเฉพาะวันเดือนปีของนักศึกษาคนแรกที่รูดเท่านั้นที่ใช้อ้างอิงวันเดือนปีในการเข้าชั้นเรียนเพื่อเป็นการประหยัดหน่วยความจำของสมาร์ตการ์ด ดังรูปที่ 4.5

เมื่อนักศึกษาคนต่อไปมารูดบัตร โปรแกรมจะเขียนเฉพาะค่าของเวลาและรหัสประจำตัวนักศึกษาเท่านั้นข้อมูลที่ได้จากการรูดบัตรประจำตัวนักศึกษาจะถูกเขียนในสมาร์ตการ์ดไปเรื่อยๆ จนถึงคนที่ 53 โปรแกรมจะบอกว่าตอนนี้หน่วยความจำของบัตรสมาร์ตการ์ดเต็ม ดังรูปที่ 4.6

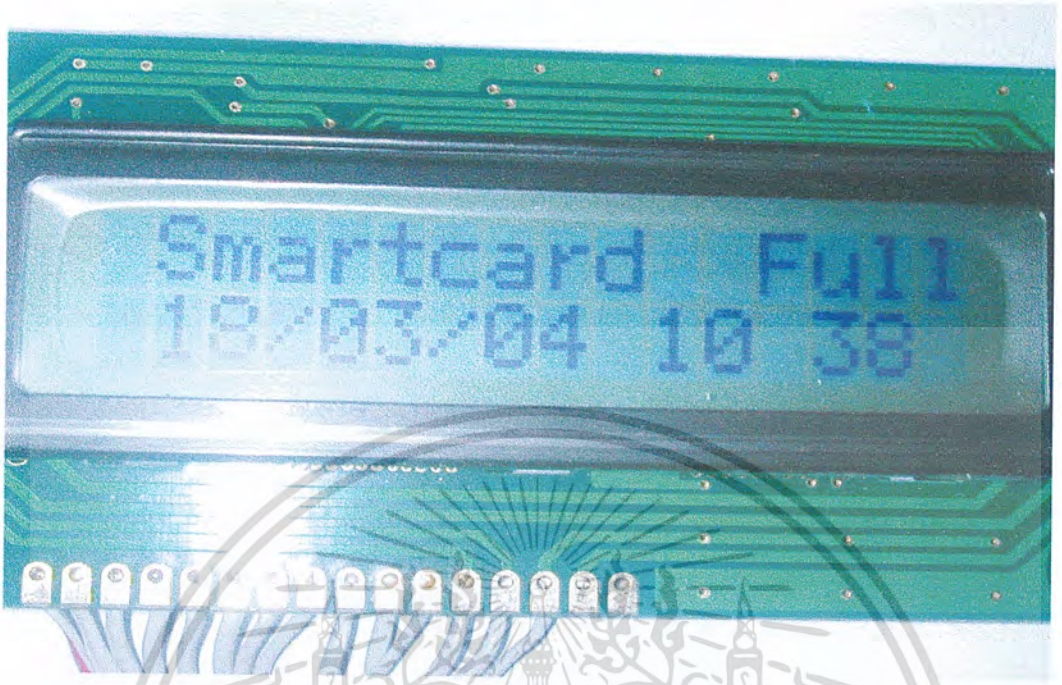


รูปที่ 4.5 หน้าจอแอลซีดีเมื่อมีการรูดบัตรประจำตัวนักศึกษา

4.3 การทดลองการทำงานของโปรแกรมอ่าน-เขียนสมาร์ตการ์ด

1) นำบัตรสมาร์ตการ์ดที่ได้ผ่านการเขียนข้อมูลลงในบัตรแล้วมาเสียบเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด ดังรูปที่ 4.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 หน้าจอแอลซีดีให้เปลี่ยนสมาร์ตการ์ดที่มีข้อมูลเต็ม

รูปที่ 4.7 แสดงการเสียบบัตรสมาร์ตการ์ดเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

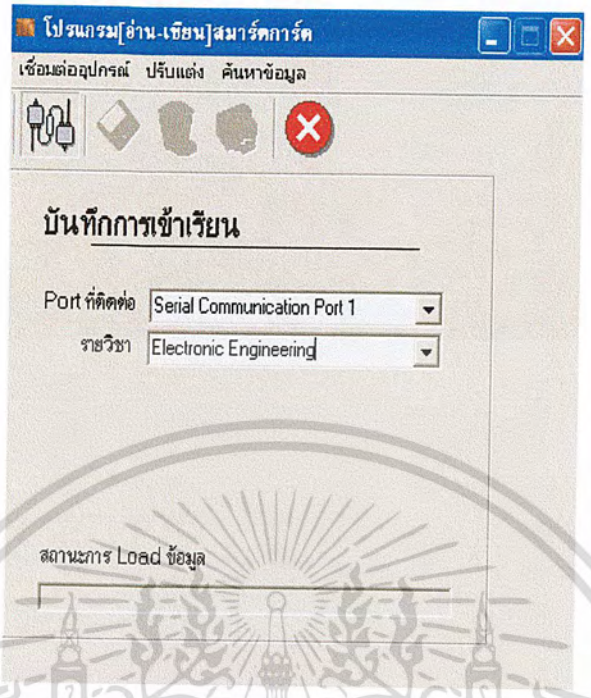
2) เปิดโปรแกรมเครื่องอ่าน-เขียนสมาร์ตการ์ดจากไอคอน ป้อนชื่อผู้ใช้งานและรหัสผ่าน จากนั้นจะเข้าสู่หน้าต่างโปรแกรมหลัก ดังรูปที่ 4.8

3) กดปุ่ม “เชื่อมต่ออุปกรณ์” เลือกรายวิชาและพอร์ตอนุกรมที่ใช้ในการติดต่อดังรูปที่ 4.9

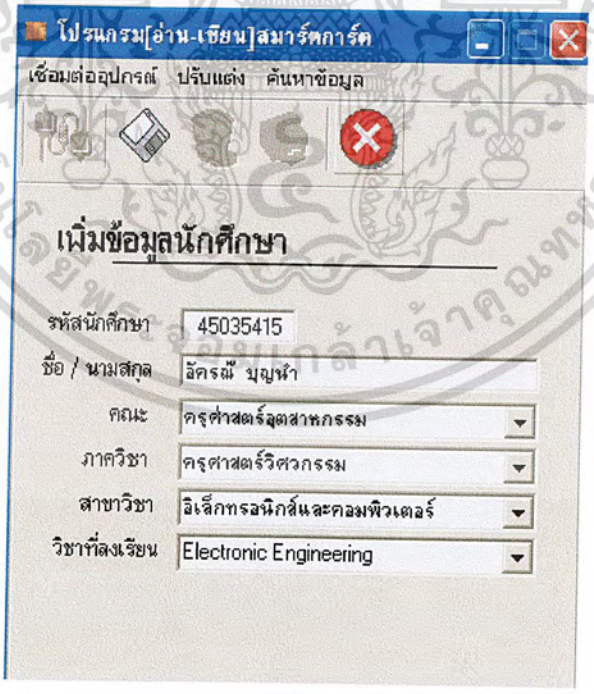
4) การเพิ่มข้อมูลนักศึกษา คือ การเพิ่มชื่อคณะ ภาควิชา สาขาวิชา และรายวิชา ให้กดปุ่ม “ปรับแต่ง” จากนั้นเลื่อนเมาส์มาคลิกที่ การเพิ่มข้อมูลทั่วไปจะได้หน้าต่างโปรแกรม ดังรูปที่ 4.10 รูปที่ 4.11 และรูปที่ 4.12 จากนั้นผู้ใช้งานสามารถเลือกได้ว่าจะต้องการเพิ่มข้อมูลอะไรเมื่อเพิ่มข้อมูลเสร็จแล้ว จากนั้นกดปุ่มเก็บข้อมูล สำหรับการเพิ่มข้อมูลอื่นๆ สามารถทำได้เหมือนตามขั้นตอนที่กล่าวมา

รูปที่ 4.8 หน้าต่างโปรแกรมการป้อนชื่อผู้ใช้และรหัสผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

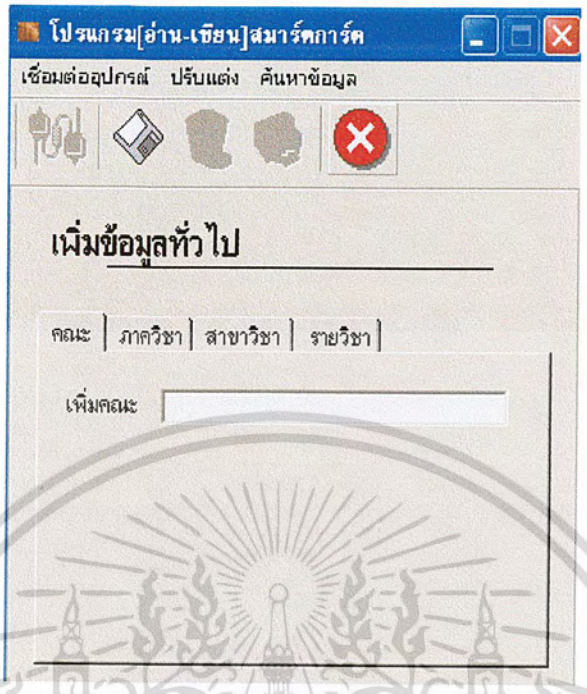


รูปที่ 4.9 หน้าต่างโปรแกรมให้เลือกพอร์ตการติดต่อและรายวิชาที่จะโหลดข้อมูล

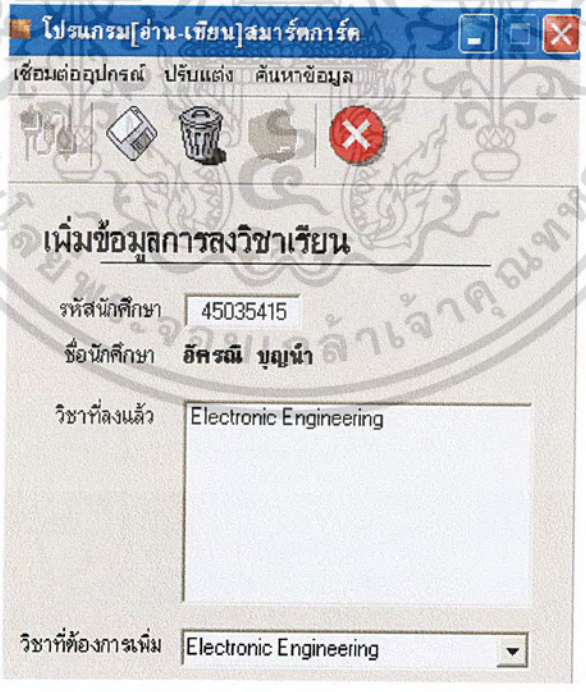


รูปที่ 4.10 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลทั่วไป

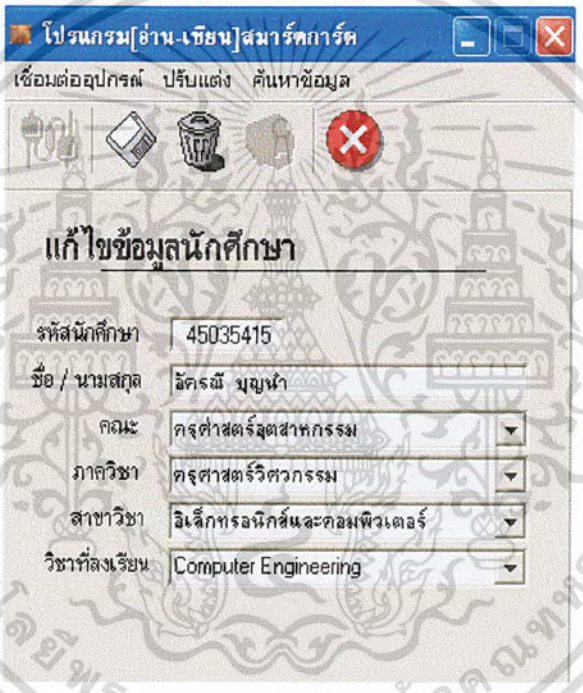


รูปที่ 4.12 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลการลงวิชาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

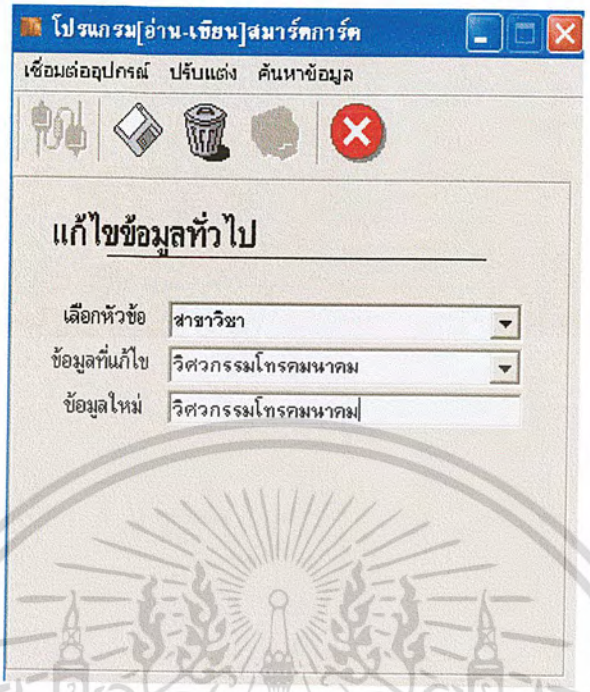
5) การแก้ไขข้อมูลกดปุ่ม “ปรับแต่ง” เลือกที่การแก้ไขข้อมูลทั่วไปหรือเลือกที่การแก้ไขข้อมูลนักศึกษา ถ้าต้องการจะบันทึกข้อมูลก็ทำการกดที่ปุ่มบันทึกจะเห็นเป็นรูปแผ่นดิสก์ หรือถ้าต้องการลบข้อมูลก็ให้กดที่ปุ่มลบข้อมูลจะเห็นเป็นรูปถังขยะ เพื่อให้ผู้ใช้เลือกว่าจะเก็บข้อมูลหรือลบข้อมูลที่ได้แก้ไข ดังรูปที่ 4.13 และรูปที่ 4.14

6) กดปุ่ม “ค้นหาข้อมูล” เพื่อเลือกข้อมูลการเข้าชั้นเรียนของนักศึกษาจากรายวิชาต่างๆ ที่นักศึกษาได้ลงทะเบียนเรียน ดังรูปที่ 4.15 จากนั้นข้อมูลที่ได้จะสามารถสั่งพิมพ์ข้อมูลผ่านทางเครื่องพิมพ์ได้ โดยกดปุ่มเครื่องพิมพ์ จะได้ข้อมูลนักศึกษา ดังรูปที่ 4.16

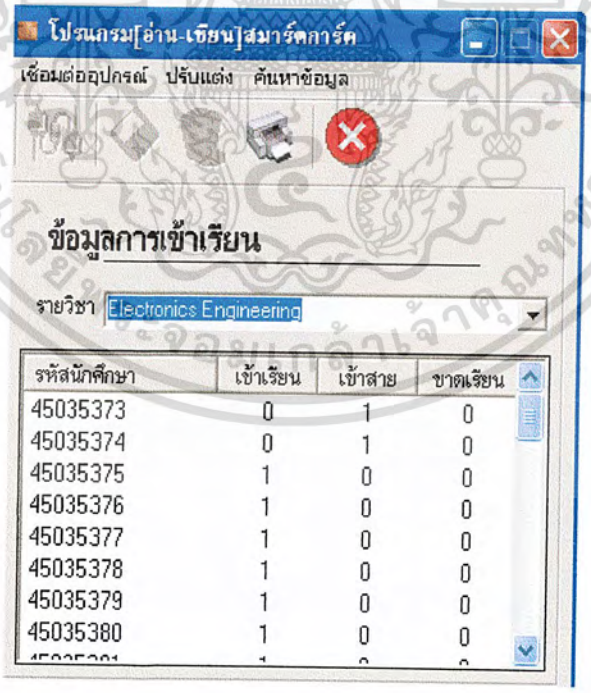


รูปที่ 4.13 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลนักศึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลทั่วไป



รูปที่ 4.15 หน้าต่างโปรแกรมเพื่อค้นหาข้อมูลการเข้าชั้นเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
คณะครุศาสตร์อุตสาหกรรม
รหัสวิชา 03321200 รายวิชา Electronics Engineering

รหัสนักศึกษา	ชื่อ - นามสกุล	เข้าเรียน	เข้าสาย	ขาดเรียน
45035373	กิติชน สชรจิ	0	1	0
45035374	เกริกชัย ชะวะนะเวช	0	1	0
45035375	โกสินทร์ บัวขาว	1	0	0
45035376	โกสินทร์ อินทราพงษ์	1	0	0
45035377	เควดิด จำปา	1	0	0
45035378	จิรวัฒน์ สาระพงศ์	1	0	0
45035379	จิราย สุรำไพ	1	0	0
45035380	ชัชชัย ไชยโย	1	0	0
45035381	ชัยณรงค์ จายเขียว	1	0	0
45035383	ชำนัญ สุขศรีนาค	1	0	0
45035385	ณานุตย์ อินทสุวรรณ	0	1	0
45035386	ธาภร สุทธิศิริกุล	0	1	0
45035387	ดนพงษ์ วัดนะ	0	1	0
45035388	ทิพย์สดา สุดประเสริฐ	0	1	0
45035389	บันทิต ศิลปะระเสริฐ	0	1	0
45035390	ประกายวรรณ ธรรมสังวาล	0	1	0
45035391	ปัญญา ภมิเจริญ	1	0	0
45035392	พรชัย ณรงค์ราช	1	0	0
45035393	ภานวัตร อนันต์แดง	1	0	0
45035394	มงคล หมวดเพชร์	1	0	0
45035395	มานะชัย ทึกว้าง	1	0	0
45035396	วชิระ เจริญรัน	1	0	0
45035397	วัชระ กลิ่นสุคนธ์	1	0	0
45035398	วิกรม เพชรสุดใส	1	0	0
45035399	วุฒินันท์ หะรารักษ์	1	0	0
45035400	ศรายุทธ สมบัติ	1	0	0
45035401	ศุภชัย สวรรณสุข	1	0	0
45035402	เศกสิทธิ์ โพธิ์ศรี	1	0	0
45035403	สดายุ ธรรมสะเร	1	0	0
45035404	สมโภช อรรคิรี	1	0	0
45035405	สลิลพรรณ คำฤทธิ์	1	0	0
45035406	สิทธิชัย ขนทองปาน	1	0	0
45035407	สุปราณี ดอนเดาเหล็ก	0	0	1
45035408	สรเดช สว่างวัน	1	0	0
45035409	สรศักดิ์ สุภาสชากุล	1	0	0
45035410	อชนิษฐ เครืออนันต์	1	0	0
45035411	อนันต์ชัย วัดดา	1	0	0
45035413	อภินันท์ รอดจินดา	1	0	0
45035414	อรรถพร อักษรปา	1	0	0

ลงชื่อ.....(อาจารย์ผู้สอน)

( )

รูปที่ 4.16 ข้อมูลที่ได้จากเครื่องพิมพ์แสดงการเข้าเรียน เข้าสาย และขาดเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุป

5.1 สรุป

ปริญญานิพนธ์ฉบับนี้ได้เสนอผลงานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด สำหรับในการออกแบบและการสร้างเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยบัตรสมาร์ตการ์ด ได้แบ่งการทำงานออกเป็น 3 ส่วนใหญ่ๆ ด้วยกันคือ ส่วนที่ 1 เป็นส่วนของการออกแบบวงจรตรวจสอบและบันทึกการเข้าชั้นเรียนโดยใช้ไมโครคอนโทรลเลอร์ประมวลผล ส่วนที่ 2 เป็นส่วนที่ใช้โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดติดต่อกับฐานข้อมูลนักศึกษาในเครื่องคอมพิวเตอร์ และส่วนที่ 3 เป็นส่วนของการออกแบบโปรแกรมควบคุมในส่วนต่างๆ

ส่วนการทำงานทั้งหมดของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดจะประกอบด้วยการทำงาน 2 ส่วน คือ ส่วนที่ใช้ตรวจสอบและบันทึกการเข้าชั้นเรียนในสมาร์ตการ์ด และส่วนที่สองใช้ติดต่อสมาร์ตการ์ดกับฐานข้อมูลในคอมพิวเตอร์

การทำงานทั้งหมดของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดมีประโยชน์อย่างมากในกรณีที่มีจำนวนนักศึกษามาก คือ ช่วยให้อาจารย์ผู้สอนมีความสะดวกในการตรวจสอบและบันทึกการเข้าชั้นเรียนของนักศึกษาในภาควิชาครุศาสตร์วิศวกรรม นอกจากนี้แล้วยังช่วยลดปัญหาเกี่ยวกับเวลาในการตรวจสอบรายชื่อและอำนวยความสะดวกในการทำงานของอาจารย์แต่ละคนได้เป็นอย่างดี

5.2 ปัญหาและแนวทางแก้ไข

จากการดำเนินการสร้างและทดสอบโครงงานพบว่ามีปัญหาเกิดขึ้นหลายประการ ซึ่งสรุปได้ดังนี้

5.2.1 ปัญหาทางด้านแหล่งข้อมูลของสมาร์ตการ์ด

1. ปัญหา การหาข้อมูลที่เกี่ยวกับทฤษฎีของสมาร์ตการ์ดเพื่อนำมาประกอบจัดทำเป็นเนื้อหาบทที่ 2 เนื่องจากสมาร์ตการ์ดเป็นเรื่องที่ต้องศึกษาใหม่ทั้งหมด สำหรับคณะผู้จัดทำโครงงานชิ้นนี้

แนวทางแก้ไข ค้นหาข้อมูลจากห้องสมุดโดยศึกษาจาก ปริญญานิพนธ์ที่มีเนื้อหาเกี่ยวกับสมาร์ตการ์ด, วารสารเซมิคอนดักเตอร์และค้นหาจากเว็บไซต์ในอินเตอร์เน็ตที่เกี่ยวกับสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ปัญหาทางด้านฮาร์ดแวร์

1. ปัญหา การหาตำแหน่งชื่อขาที่ถูกต้องของซอกเกตสมาร์ตการ์ด ที่จะนำมาวางบนแผ่นวงจรพิมพ์

แนวทางแก้ไข ค้นหาข้อมูลที่เกี่ยวกับตำแหน่งชื่อขาของซอกเกตสมาร์ตการ์ด จากโครงงานที่เกี่ยวกับสมาร์ตการ์ดในวารสารเซมิคอนดักเตอร์และค้นหาจากค้นหาจากเว็บไซต์ของต่างประเทศที่เกี่ยวกับสมาร์ตการ์ดมาเปรียบเทียบข้อมูลกัน

2. ปัญหา การสร้างตำแหน่งขาของตัวซอกเกตสมาร์ตการ์ด ในโปรแกรมออกแบบลายวงจรพิมพ์ซึ่งจะไม่ได้มีไว้ในโปรแกรมออกแบบลายวงจรพิมพ์ ทำให้สร้างปัญหาในการวางอุปกรณ์บนแผ่นวงจรพิมพ์

แนวทางแก้ไข ใช้การวัดตำแหน่งขาของซอกเกตสมาร์ตการ์ดเปรียบเทียบกับตำแหน่งขาของไอซีที่เป็นมาตรฐานทั่วไปในโปรแกรมออกแบบลายวงจรพิมพ์

5.2.3 ปัญหาทางด้านซอฟต์แวร์

1. ปัญหา การเขียนโปรแกรมด้วยภาษาแอสเซมบลีนั้นการแก้ไขการควบคุมการทำงานแต่ละส่วนทำได้ยากกว่าการเขียนโปรแกรมด้วยภาษาซีเพราะเป็นภาษาระดับต่ำ คำสั่งที่ใช้งานมีความซับซ้อนทำให้เกิดการผิดพลาดได้ง่าย

แนวทางแก้ไข เขียนโปรแกรมแยกออกเป็นส่วนย่อย ของการทำงานแต่ละส่วนแล้วจึงนำมารวมเข้าด้วยกันและทำความเข้าใจคำสั่งที่ใช้ให้มากขึ้น

2. ปัญหา การเขียนโปรแกรมวิชวลเบสิกติดต่อกับฐานข้อมูลในโปรแกรมไมโครซอฟท์แอคเซส เนื่องจากคณะผู้จัดทำมีความรู้พื้นฐานน้อยในการเขียนโปรแกรมเพื่อติดต่อกับฐานข้อมูลในคอมพิวเตอร์

แนวทางแก้ไข ศึกษาจากตำราการเขียนโปรแกรมวิชวลเบสิกที่ติดต่อกับฐานข้อมูลและศึกษาจากโค้ดโปรแกรมตัวอย่างเบื้องต้น

3. ปัญหา การอ่านข้อมูลจากสมาร์ตการ์ดผ่านทางเครื่องอ่าน-เขียนสมาร์ตการ์ดเพื่อโหลดข้อมูลมาเก็บที่ฐานข้อมูล

แนวทางแก้ไข ศึกษาจากโค้ดตัวอย่างที่อยู่ในแผ่นซีดีรอมตัวอย่างที่มาพร้อมกับเครื่องอ่าน-เขียนสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 แนวทางการพัฒนา

1. สำหรับกรณีที่หน่วยความจำของสมาร์ตการ์ดเต็มแล้วจะแสดงข้อความแจ้งเตือนผ่านทางจอแอลซีดี วิธีการแสดงผลแบบนี้ทำให้ต้องใช้การสังเกตจากสายตามาก ดังนั้นจึงควรแสดงผลในรูปแบบของไฟแสดงผลหรือใช้สัญญาณเสียงต่างๆ เพื่อบอกว่าหน่วยความจำสมาร์ตการ์ดเต็มแล้ว

2. ควรพัฒนาความสามารถของโปรแกรมวิชวลเบสิกให้สามารถคิดคะแนนเวลาเข้าชั้นเรียนที่เก็บในฐานข้อมูลได้ด้วย

3. เนื่องจากสมาร์ตการ์ดที่ใช้มีหน่วยความจำน้อยสามารถใช้ได้ต่อการเรียน 1 คาบเท่านั้น จึงไม่สะดวกในการใช้งานครั้งต่อไปที่จะต้องไปโหลดเก็บข้อมูลทุกครั้ง ดังนั้นควรใช้สมาร์ตการ์ดที่มีหน่วยความจำมากกว่านี้

4. ในส่วนของตัวกล่องบรรจุวงจรและวงจรพิมพ์นั้นควรพัฒนาให้มีขนาดเล็กและน้ำหนักเบาเพื่อทำให้สะดวกในการใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

กฤษดา ใจเย็น. **เรียนรู้การเชื่อมต่อ PC กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม.** กรุงเทพฯ : บริษัทอินโนเวตีฟ เอ็กเพอริเมนต์จำกัด. 2542

สัจจะ จรัสรุ่งรวีวร. **คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.** พิมพ์ครั้งที่ 2 . กรุงเทพฯ : อินโฟเพรส. 2544

ธราธร รัตนเสถียร. "เครื่องอ่าน-เขียนสมาร์ตการ์ด." ปริญญานิพนธ์ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2545

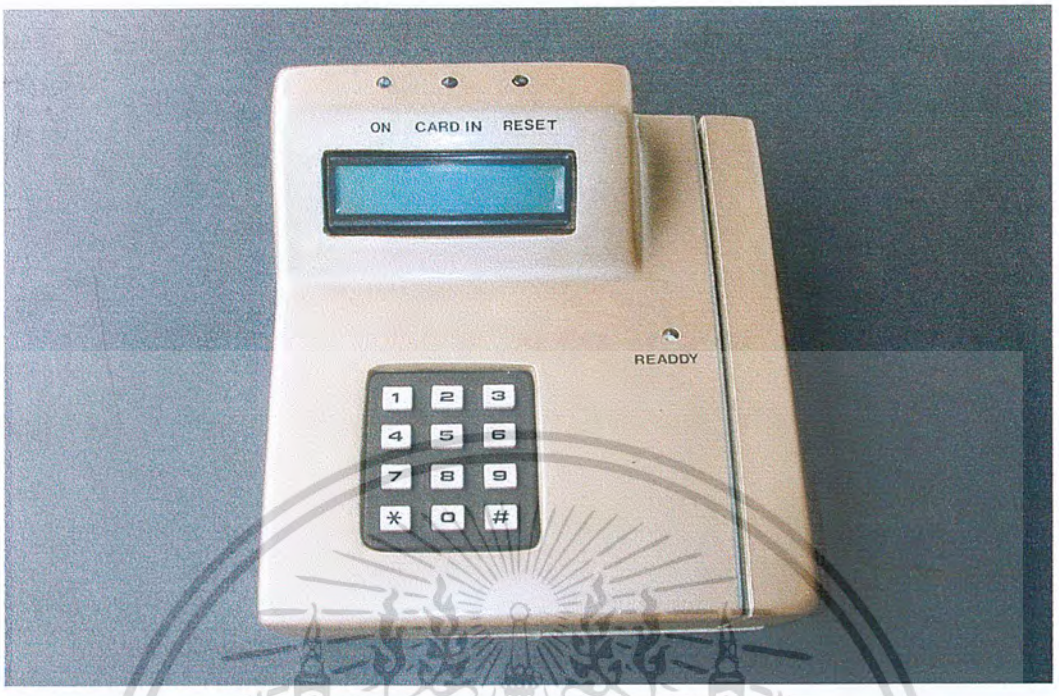
ธีรวัฒน์ ประกอบผล. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์.** กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). 2545

อารัมภีย์ จันทร์ใย. "การสร้าง/เขียนบัตรสมาร์ตการ์ด." **เซมิคอนดัคเตอร์อิเล็กทรอนิกส์.** ฉบับที่ 240 : หน้า 155-162. 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
เครื่องต้นแบบ

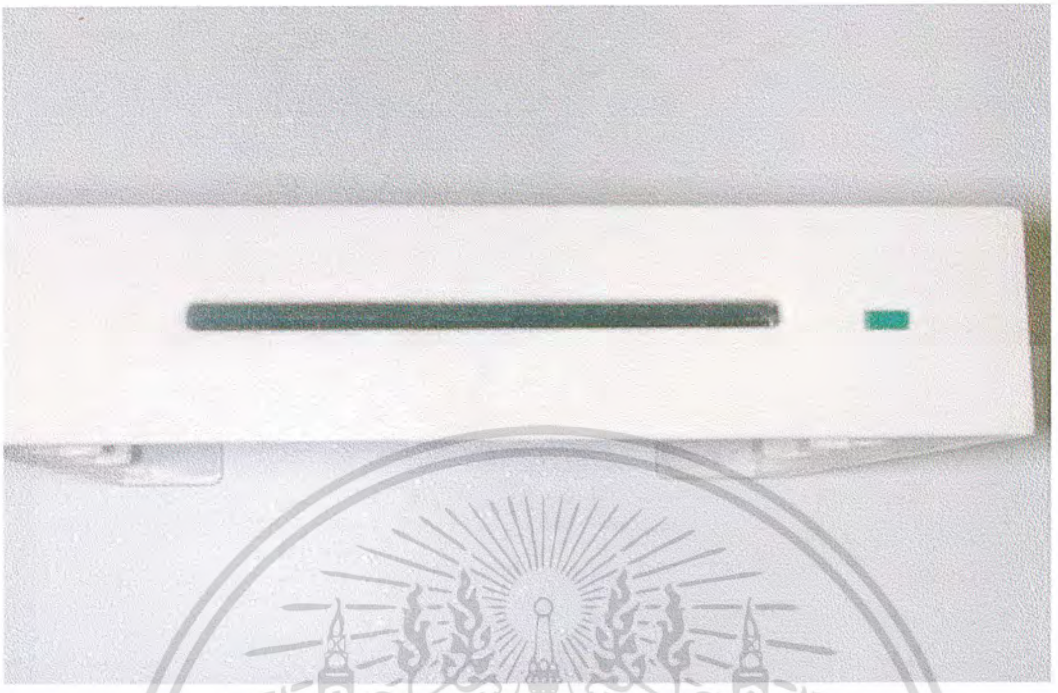
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



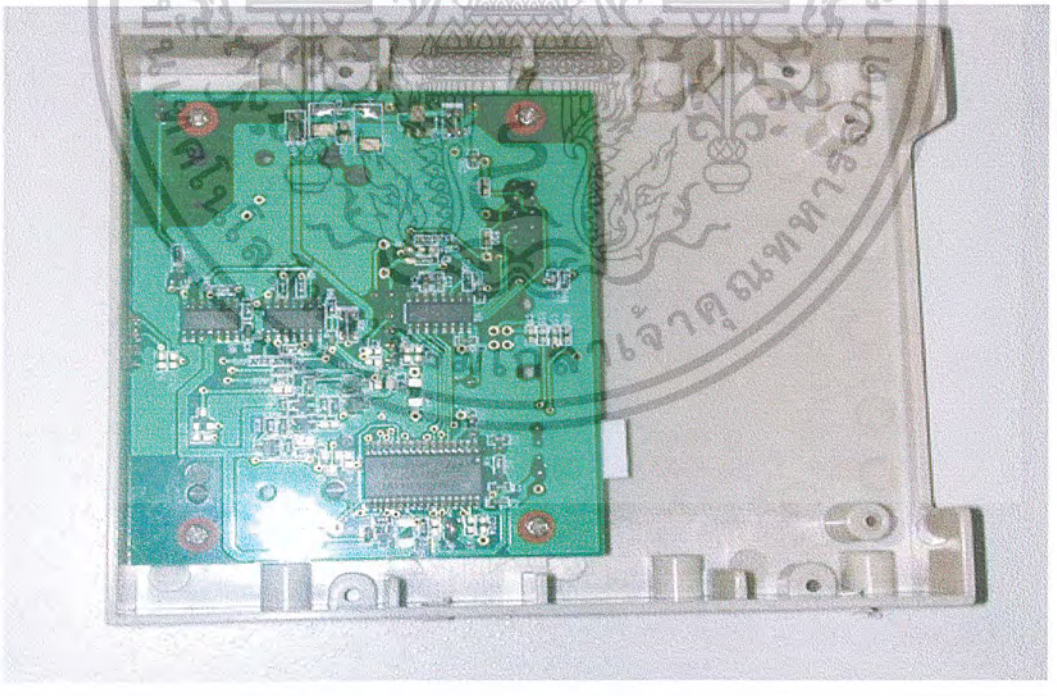
รูปที่ ก.1 ภาพด้านหน้าของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

รูปที่ ก.2 ภาพด้านข้างของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

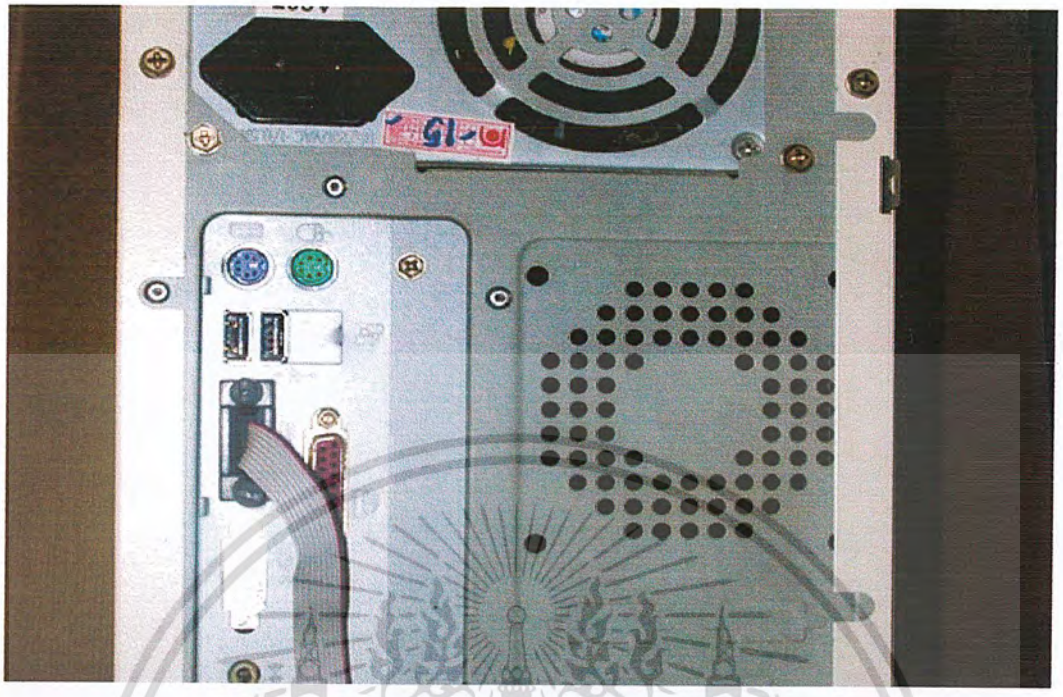


รูปที่ ก.3 ภาพด้านหน้าของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด WACR20S



รูปที่ ก.4 ภาพด้านในของโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด WACR20S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.5 การเชื่อมต่อโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดกับเครื่องคอมพิวเตอร์

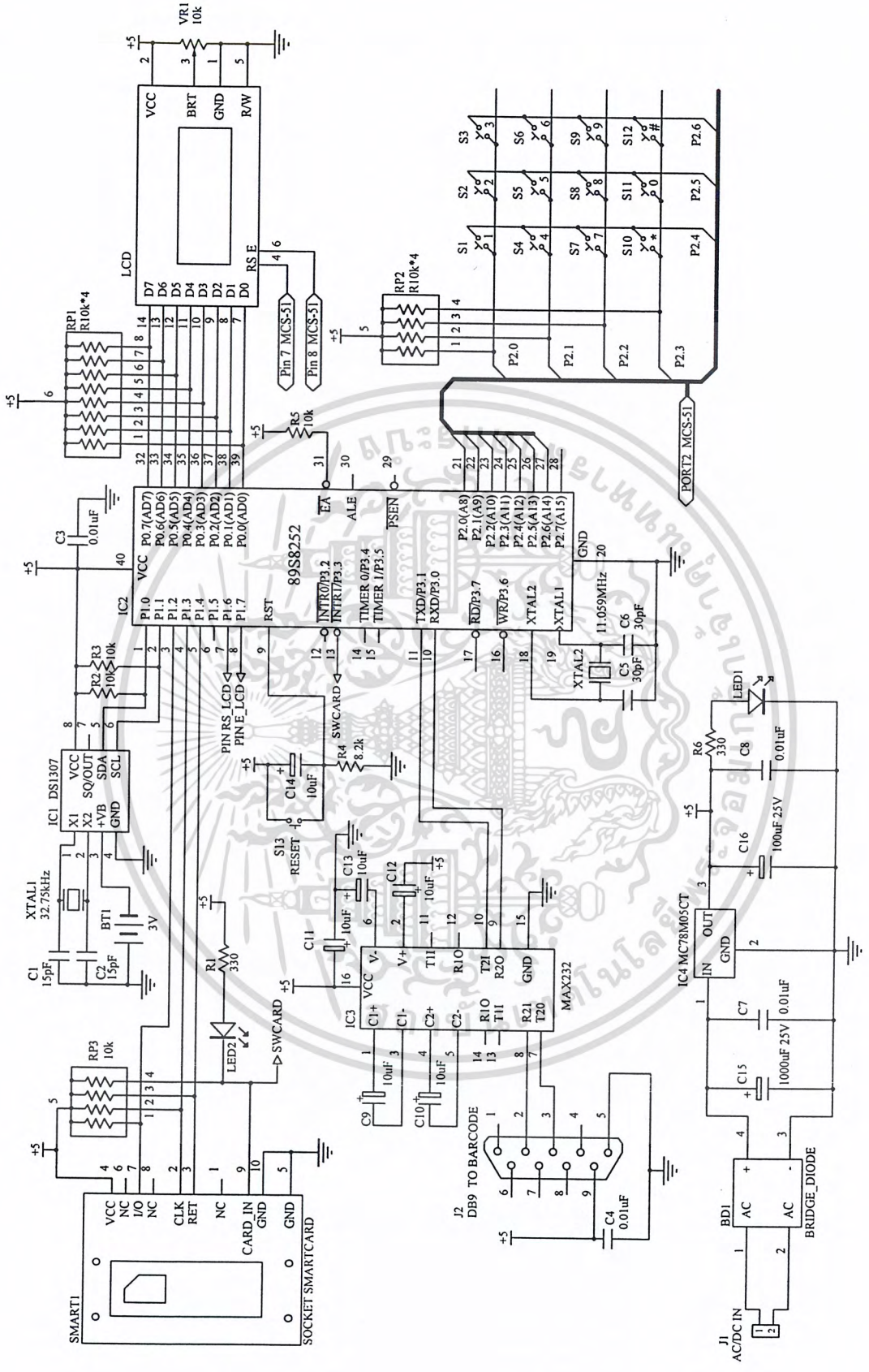
รูปที่ ก.6 บัตรสมาร์ตการ์ด SLE 4428

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

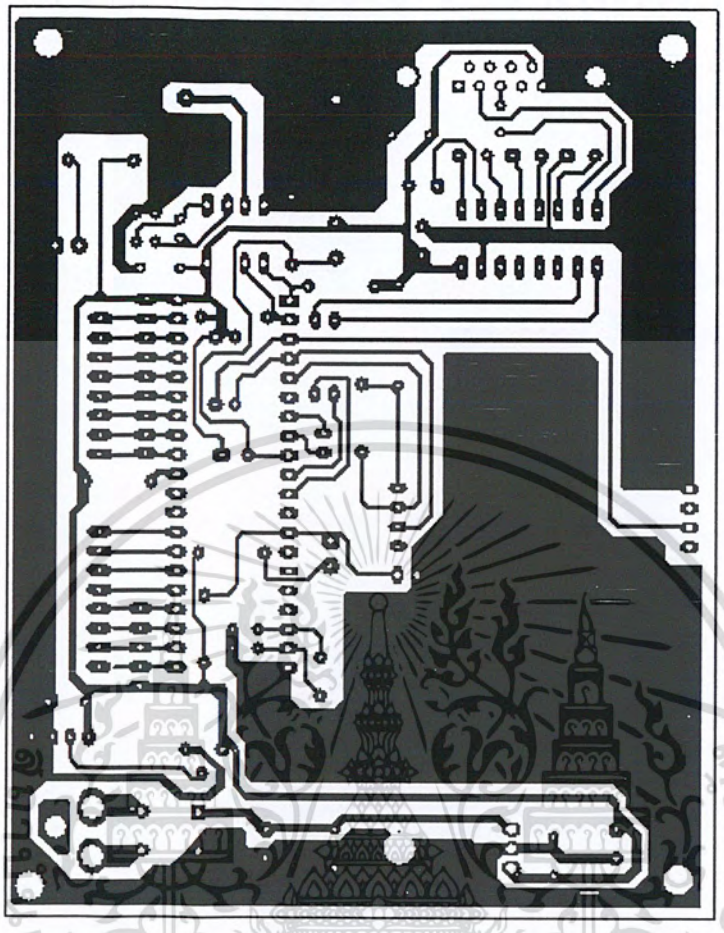
ภาคผนวก ข

วงจรและแผ่นวงจรพิมพ์

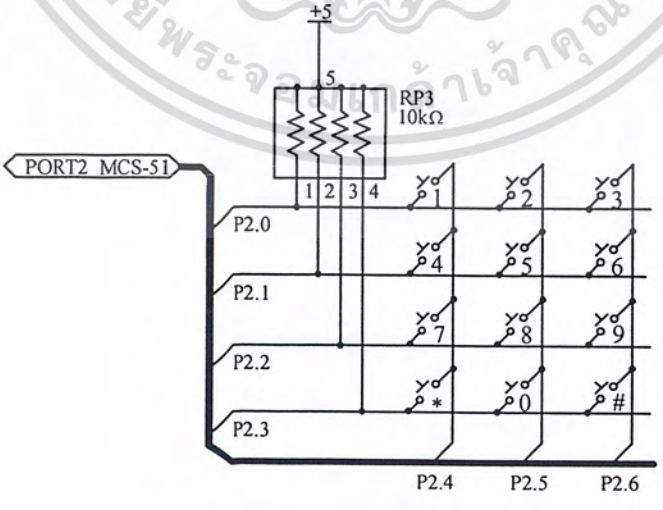
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.1 วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

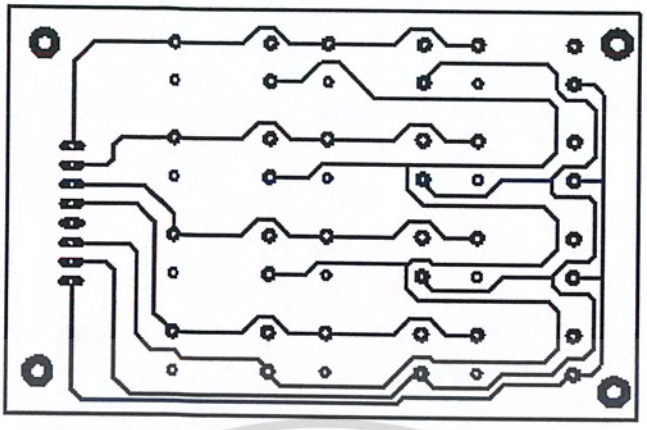


รูปที่ ข.2 แผ่นวงจรพิมพ์ของวงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

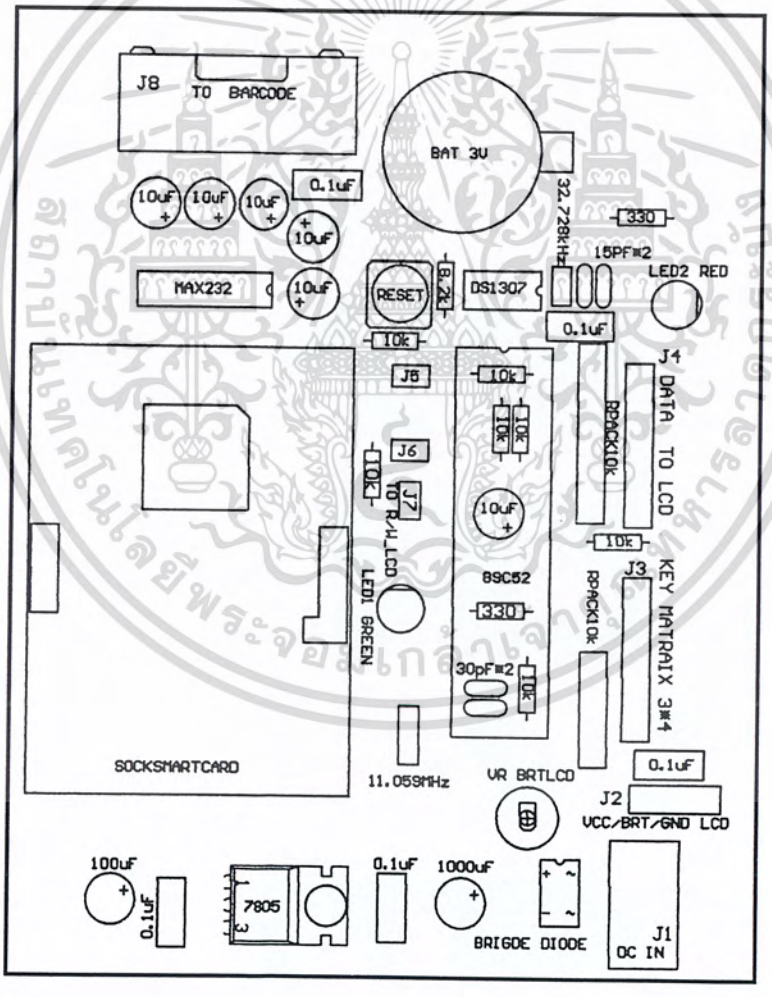


รูปที่ ข.3 วงจรวงจรคีย์เมตริกซ์สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

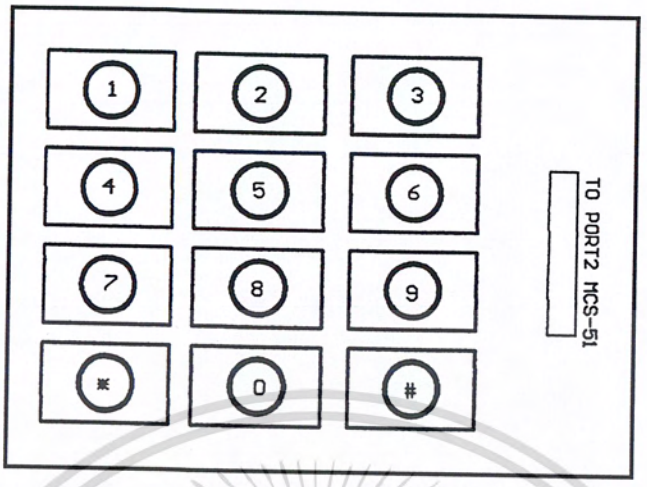


รูปที่ ข.4 แผ่นวงจรพิมพ์ของวงจรคีย์เมตริกซ์สวิตช์



รูปที่ ข.5 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียน โดยสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 ตำแหน่งการวางอุปกรณ์แผ่นวงจรพิมพ์วงจรคีย์เมตริกซ์สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

รายการอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ค.1 รายการอุปกรณ์วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

ชื่ออุปกรณ์	รายละเอียด	จำนวน
วงจรรวม		
IC1	DS1307	1 ตัว
IC2	AT89S8252	1 ตัว
IC3	MAX232	1 ตัว
IC4	MC7805	1 ตัว
BD1	ไดโอดบริดจ์ ขนาด 2 A	1 ตัว
ตัวเก็บประจุ		
C1, C2	15 PF 50 V เซรามิค	2 ตัว
C3, C4, C7, C8	0.01 μF 50 V เซรามิค	4 ตัว
C5, C6	33 PF 50 V เซรามิค	2 ตัว
C9 – C14	10 μF 16 V อิเล็กทรอไลต์	6 ตัว
C15	1000 μF 25 V อิเล็กทรอไลต์	1 ตัว
C16	100 μF 25 V อิเล็กทรอไลต์	1 ตัว
ตัวความต้านทาน		
R1, R6	330 Ω 1/4 W	2 ตัว
R2, R3, R5	10 kΩ 1/4 W	3 ตัว
RP1	10 kΩ 9 ขา	1 ตัว
RP2, RP3	10 kΩ 5 ขา	2 ตัว
VR1	10 kΩ	1 ตัว
อุปกรณ์อื่นๆ		
XTAL1	คริสตอล 32.75 KHz	1 ตัว
XTAL2	คริสตอล 11.059 MHz	1 ตัว
S1 – S12	สวิตซ์กดติดปล่อยดับ	12 ตัว
	หรือโมดูลคีย์เมตริกซ์ 4×3	1 ตัว
S13	สวิตซ์กดติดปล่อยดับ	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

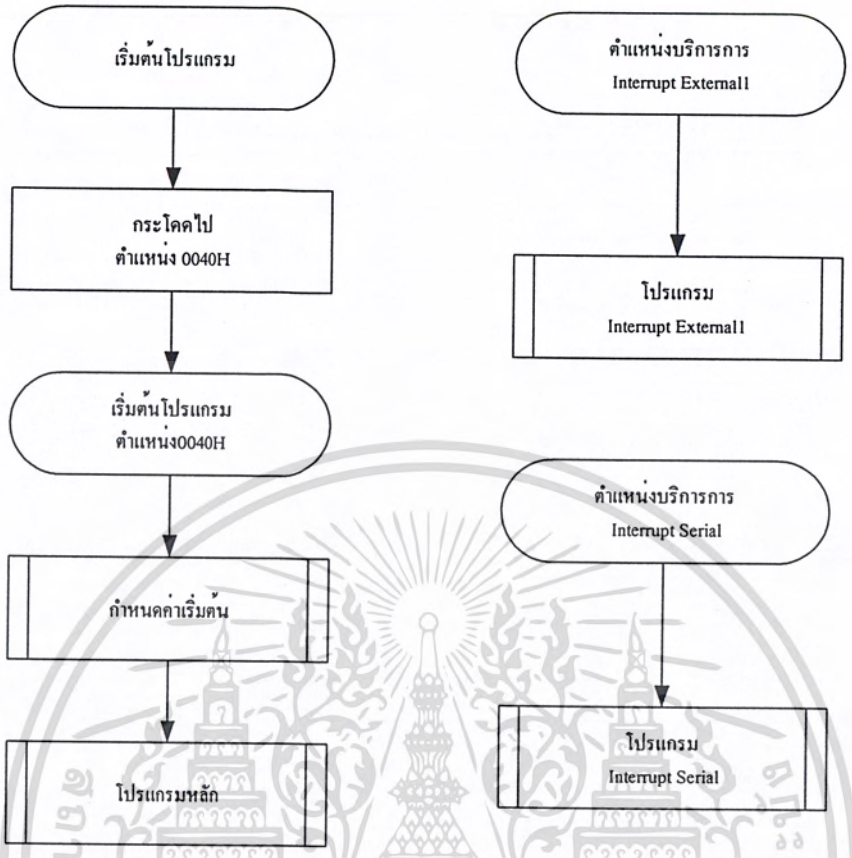
ตารางที่ ค.1 (ต่อ) รายการอุปกรณ์วงจรเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

ชื่ออุปกรณ์	รายละเอียด	จำนวน
อุปกรณ์อื่นๆ		
J1	แจ็คแบบอะแด็ปเตอร์ลงปริ้นซ์	1 ตัว
J2	Connector 4 pin	1 ตัว
J3, J4	Connector 8 pin	2 ตัว
J5	Connector 2 pin	3 ตัว
J6, J7	ลวดจัมเปอร์	2 ตัว
J8	DB 9 ตัวผู้ ลงปริ้นซ์	1 ตัว
LED1	ไดโอดเปล่งแสงสีแดง	1 ตัว
LED2	ไดโอดเปล่งแสงสีเขียว	1 ตัว
LCD1	โมดูลแบบ 16 ตัวอักษร 2 บรรทัด	1 ตัว
ซอกเกต วงจรรวม	แบบ 40 ขา สำหรับ IC AT89S8252	1 ตัว
ซอกเกต วงจรรวม	แบบ 16 ขา สำหรับ IC MAX 232	1 ตัว
ซอกเกต วงจรรวม	แบบ 8 ขา สำหรับ IC DS1307	1 ตัว
SMART CARD SLOT	ซอกเกตสำหรับเสียบสมาร์ตการ์ด	1 ตัว
SMART CARD	เบอร์ SLE 4428 ของซีเมนส์	2 ใบ
เครื่องอ่านบาร์โค้ด	รุ่น 103 DR	1 เครื่อง
เครื่องอ่าน-เขียนสมาร์ตการ์ด	รุ่น WACR20S	1 เครื่อง
อะแด็ปเตอร์	แบบเอาต์พุต DC 12 - 18 โวลต์	1 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

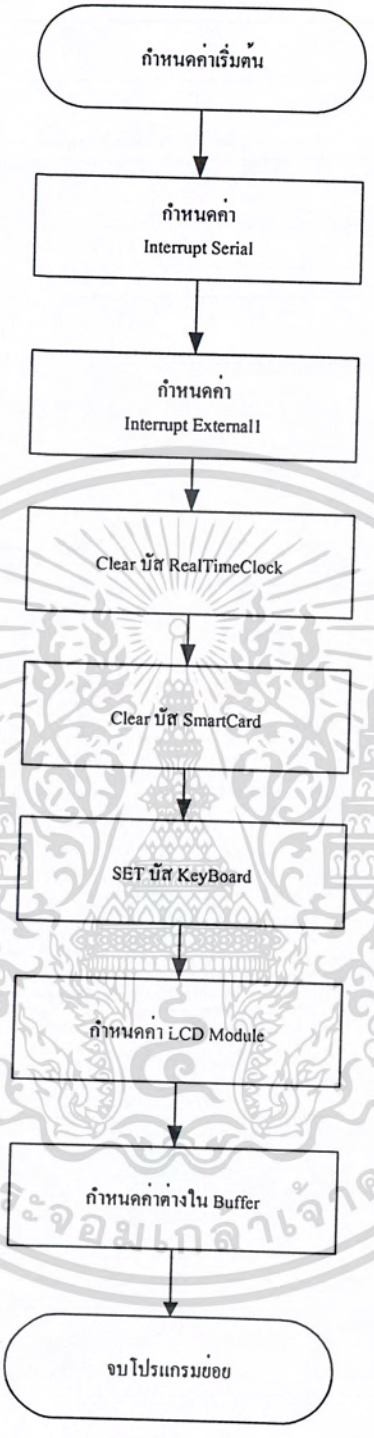
ภาคผนวก ง
แผนผังการทำงานและรหัสต้นฉบับของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



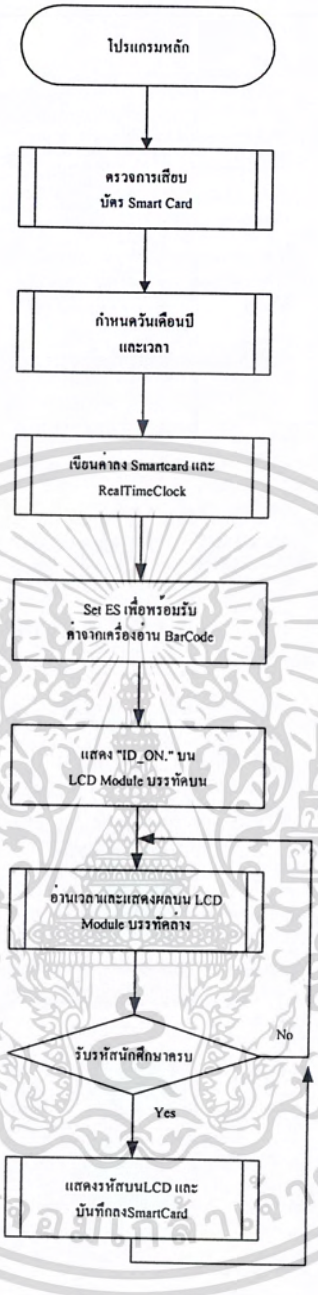
รูปที่ ง.1 แผนผังเริ่มต้นการทำงานของโปรแกรมและการทำงานของโปรแกรมย่อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



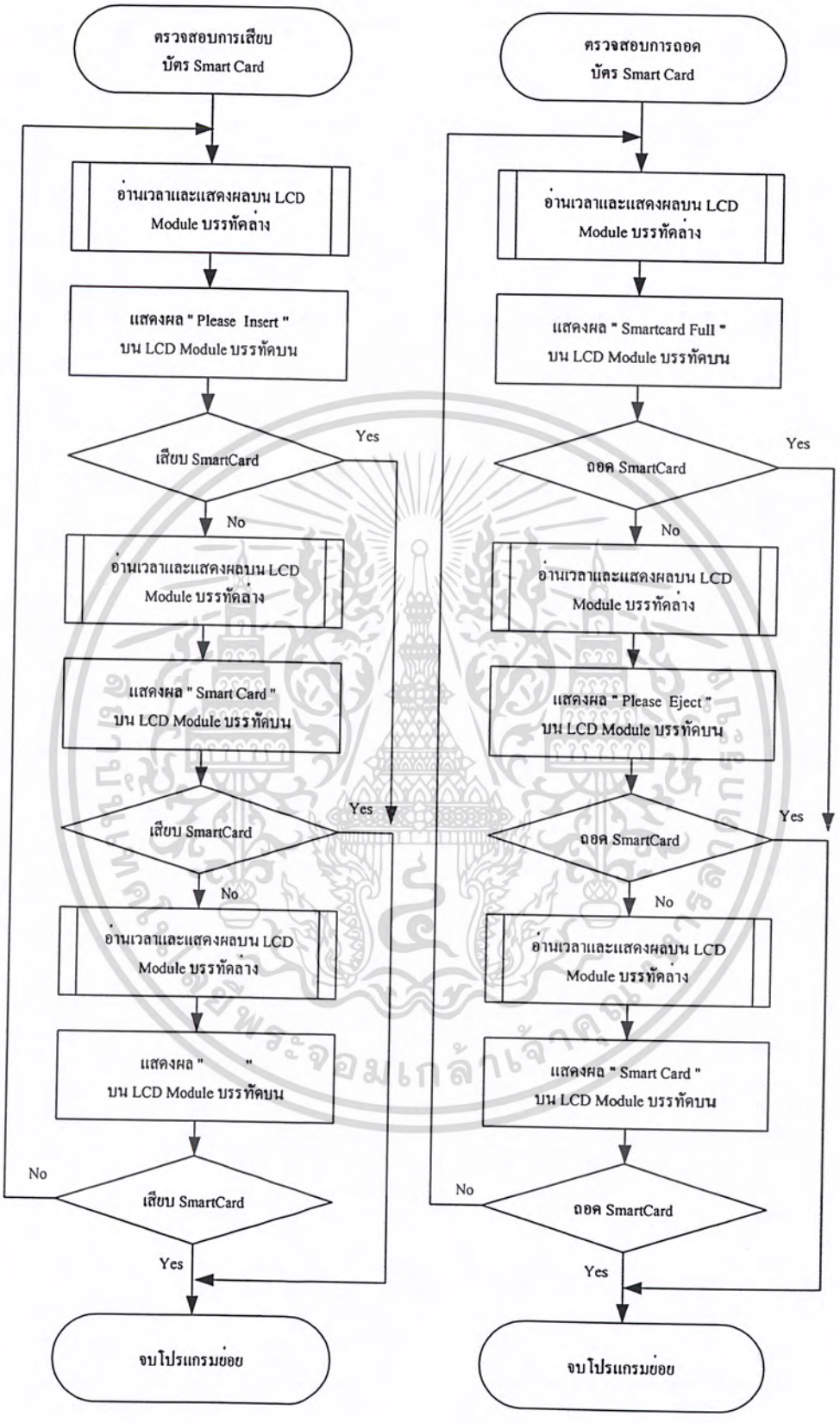
รูปที่ ง.2 แผนผังการทำงานการกำหนดค่าเริ่มต้นของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



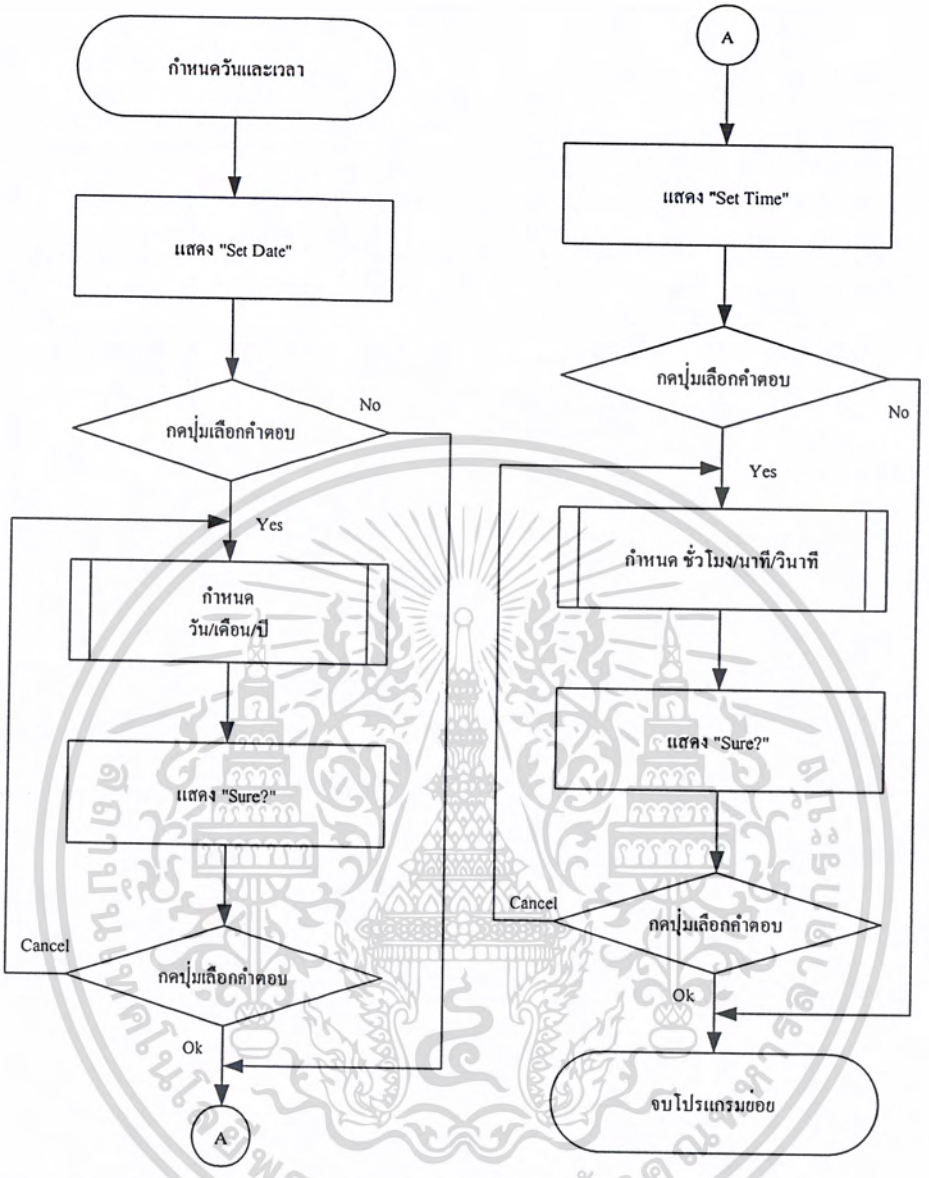
รูปที่ ง.3 แผนผังการทำงานของโปรแกรมตรวจสอบและบันทึกข้อมูลลงสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



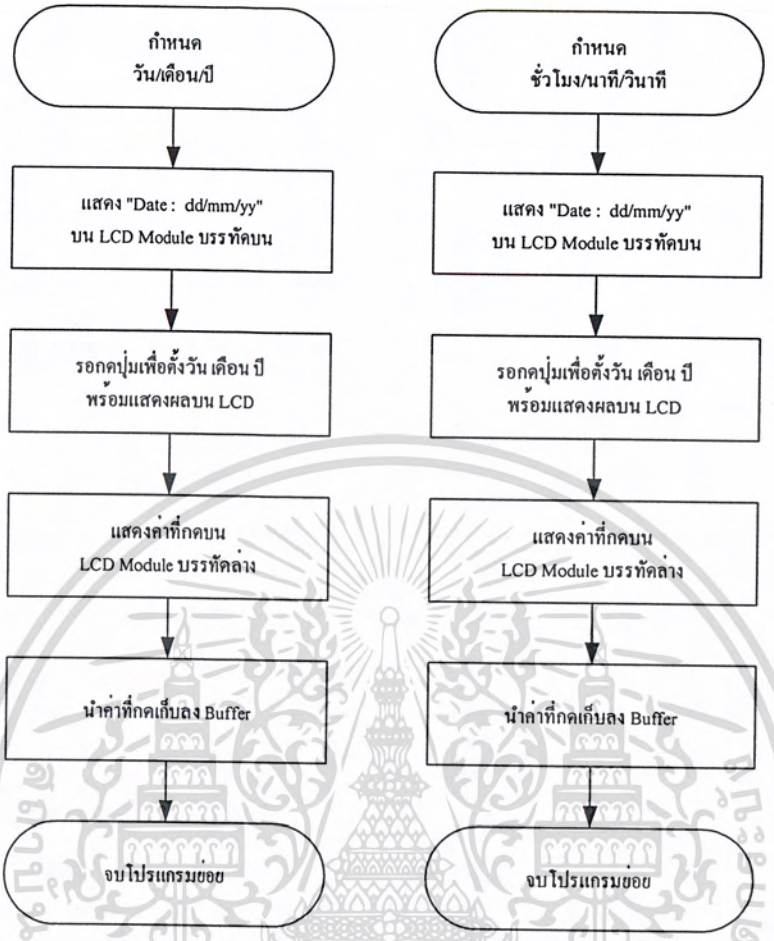
รูปที่ ง.4 แผนผังการทำงานของโปรแกรมตรวจสอบการเสียบบัตรและถอดบัตรสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



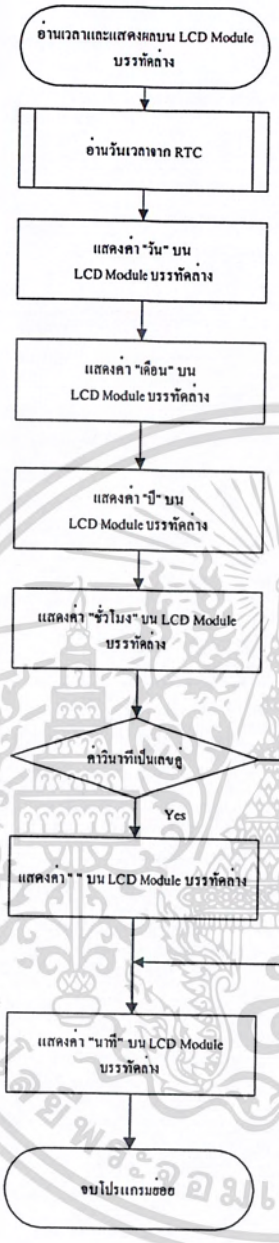
รูปที่ ง.5 แผนผังการทำงานของโปรแกรมเลือกการกำหนดค่าของวันเดือนปีและเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



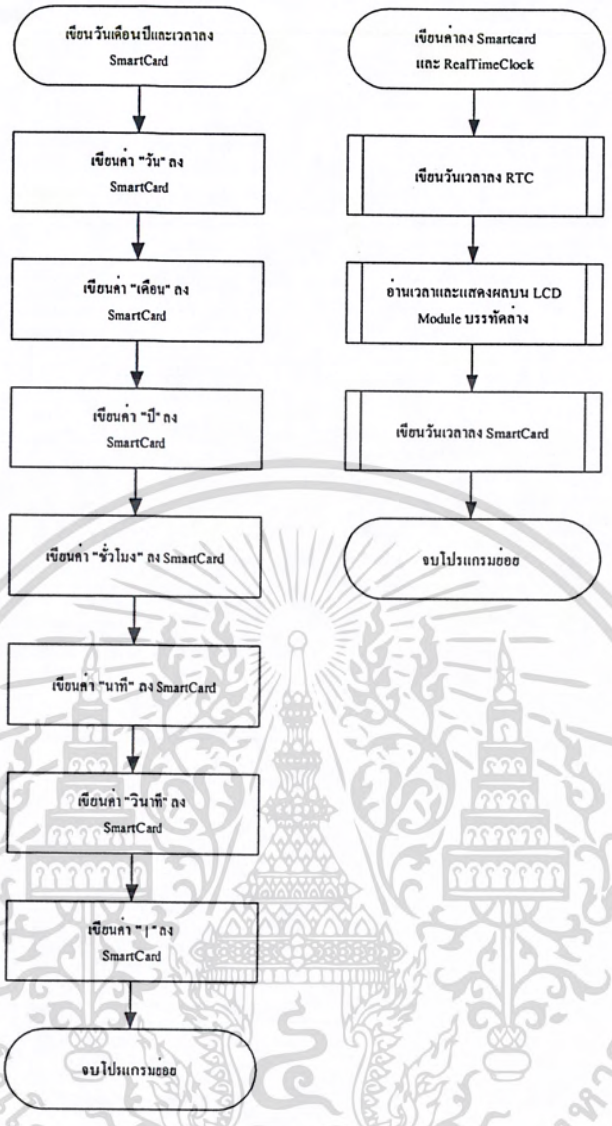
รูปที่ ง.6 แผนผังการทำงานของโปรแกรมกำหนดค่าของวันเดือนปีและเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



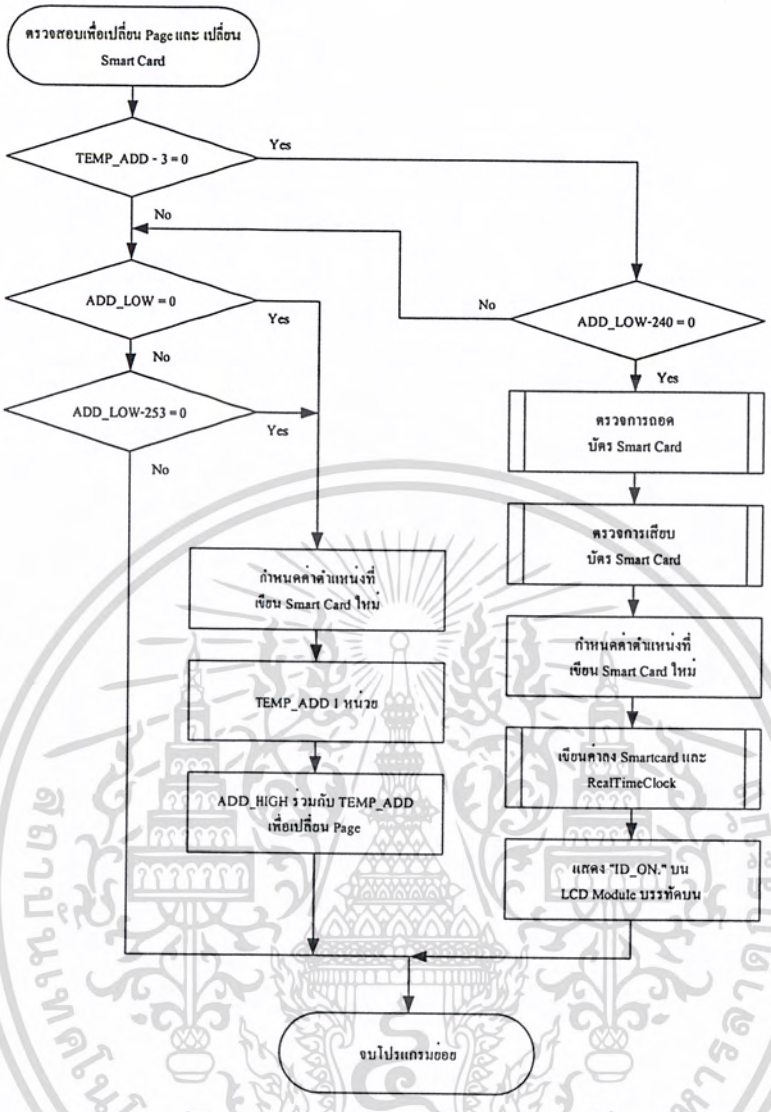
รูปที่ ง.7 แผนผังการทำงานโปรแกรมแสดงผลวันเดือนปีและเวลาบนจอแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



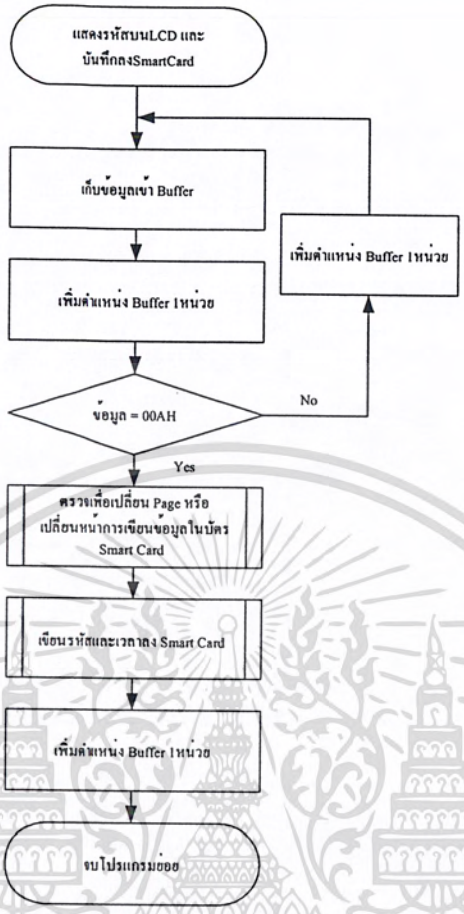
รูปที่ ง.8 แผนผังการทำงานของโปรแกรมเขียนค่าของวันเดือนปีและเวลาลงสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



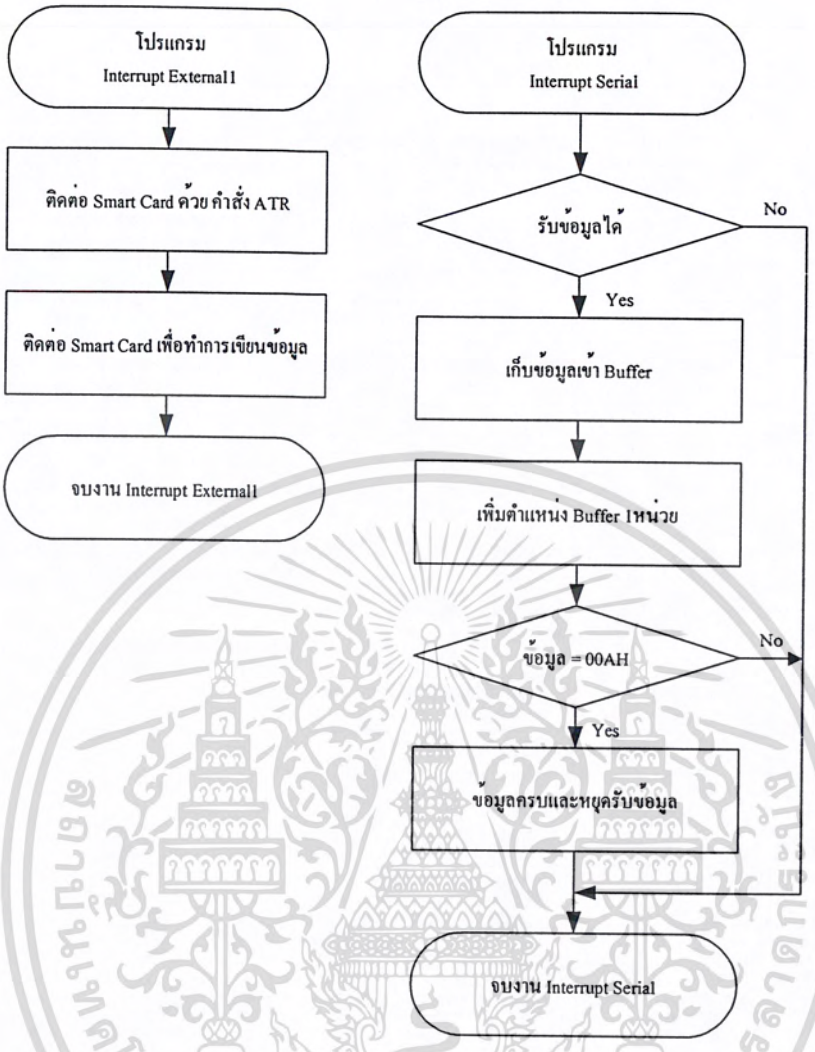
รูปที่ ง.9 แผนผังการทำงานของโปรแกรมตรวจสอบข้อมูลในสมาร์ตการ์ดและเปลี่ยนหน้าของการเขียนข้อมูลในบัตรสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



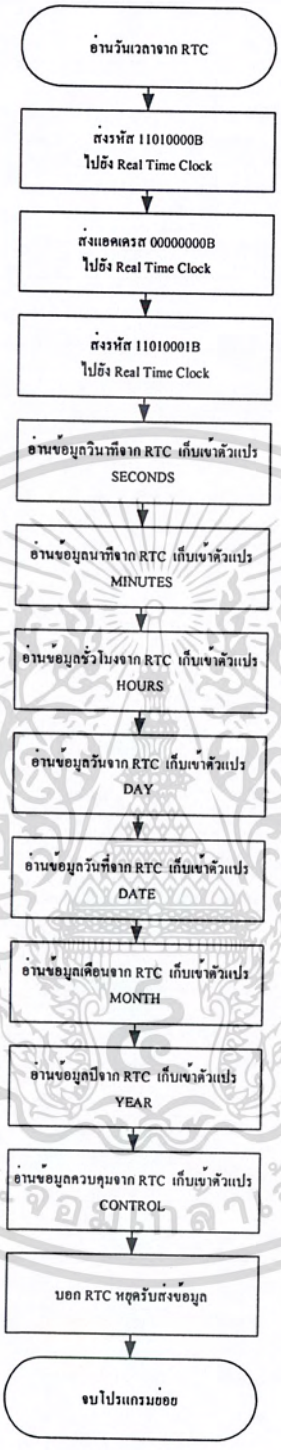
รูปที่ ง.10 แผนผังการทำงานของโปรแกรมแสดงรหัสนักศึกษาบนจอแอลซีดีและบันทึกรหัสนักศึกษาลงบัตรสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



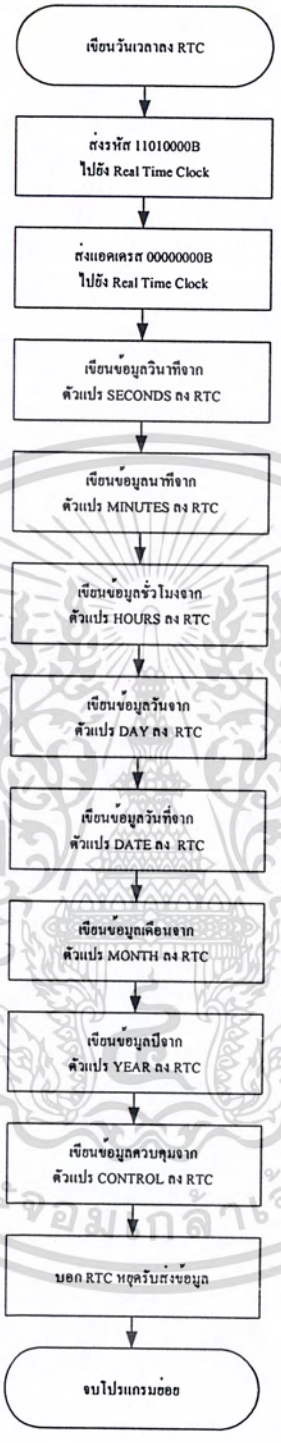
รูปที่ ง.11 แผนผังการทำงานของโปรแกรมการอินเตอร์รัพต์สมาร์ตการ์ดและเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



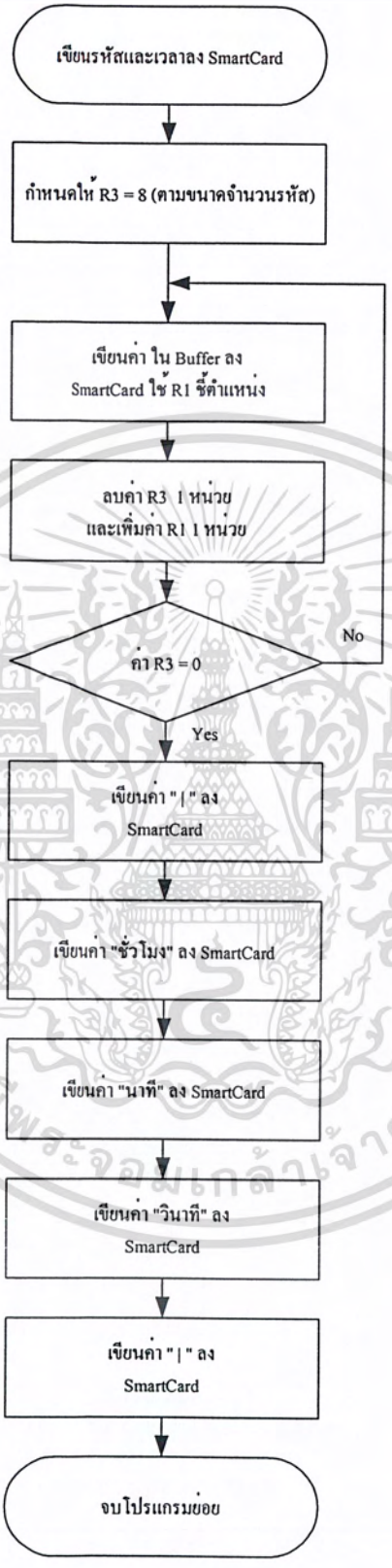
รูปที่ ง.12 แผนผังการทำงานของโปรแกรมอ่านค่าของวันเดือนปีและเวลาจาก Real Time Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.13 แผนผังการทำงานของโปรแกรมเขียนค่าของวันเดือนปีและเวลาลง Real Time Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ง.14 แผนผังการทำงานของโปรแกรมเขียนค่าของเวลาและรหัสนักศึกษาลงสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

```
;-------------------------------------------------------------------
; Define Port&Pin Name
;-------------------------------------------------------------------
SDA1        BIT   P1.0  ; SDA I2C Bus
SCL1        BIT   P1.1  ; SCL I2C Bus
SDA         BIT   P1.2  ; SDA Bus SMART
SCL         BIT   P1.3  ; SCL Bus SMART
RST         BIT   P1.4  ; RST Bus SMART
SW          BIT   P3.3  ; Switch Bus SMART

LCD_EN      BIT   P1.6  ; LCD Module Enable (Active High: Level)
LCD_RS      BIT   P1.7  ; LCD Module Register Select

KPAD_ROW0   BIT   P2.0  ; Keypad Input Row 0
KPAD_ROW1   BIT   P2.1  ; Keypad Input Row 1
KPAD_ROW2   BIT   P2.2  ; Keypad Input Row 2
KPAD_ROW3   BIT   P2.3  ; Keypad Input Row 3
KPAD_COL3   BIT   P2.4  ; Keypad Output Column 3
KPAD_COL2   BIT   P2.5  ; Keypad Output Column 2
KPAD_COL1   BIT   P2.6  ; Keypad Output Column 1
KPAD_COL0   BIT   P2.7  ; Keypad Output Column 0
RX_OK       BIT   F0    ; Define bit to keep received already


;-------------------------------------------------------------------
; Define User Register
;-------------------------------------------------------------------
FLAG        EQU   02FH        ; User FLAG
I2C_ACK     BIT   FLAG.0      ; Define I2C Acknowledge as bit
TX_DATA     EQU   02FH        ; For keep TX Data
LCD_ADDR    EQU   030H        ; For keep LCD Address
LCD_DATA    EQU   031H        ; For keep LCD Data
LCD_PTR     EQU   032H        ; For keep LCD 3 Char Pointer
I2C_ADDR    EQU   033H        ; For keep I2C Address
I2C_DATA    EQU   034H        ; For keep I2C Data
KPAD_DATA   EQU   035H        ; For keep Keypad Data
SECONDS     EQU   040H        ; For keep Seconds
MINUTES     EQU   041H        ; For keep Minutes
HOURS       EQU   042H        ; For keep Hours
DAY         EQU   043H        ; For keep Day
DATE        EQU   044H        ; For keep Date
MONTH       EQU   045H        ; For keep Month
YEAR        EQU   046H        ; For keep Year
CONTROL     EQU   047H        ; For keep Control Byte
ADD_LOW     EQU   048H        ; For keep Address Low SMART
ADD_HIGH    EQU   049H        ; For keep Address High SMART
TEMP_ADD    EQU   04AH
SMART_DATA  EQU   04BH
BUFFER      EQU   050H        ; Buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------
; Define I2C Slave Address
;-------------------------------------------------------------------
RTC_ID          EQU   11010000B         ; RTC Slave Address
SMART_RD        EQU   00001110B         ; Read
SMART_RD1021    EQU   11001110B         ; Read 1021
SMART_RDP       EQU   00001100B         ; Read Protect
SMART_WR        EQU   00110011B         ; Write
SMART_WR1021    EQU   11110011B         ; Write 1021
SMART_WRP       EQU   00110001B         ; Write Protect
BMK             EQU   11110010B         ; Bit Mask
PSC             EQU   11001101B         ; PSC Byte

;-------------------------------------------------------------------
; Main Program.
;-------------------------------------------------------------------
                ORG   0000H             ; Reset Vector
                AJMP  INIT              ; Jump to Main
                ORG   0013H             ; E Vector
                AJMP  EX1_INT
                ORG   0023H             ; TI+RI Vector
                AJMP  SER_INT
INIT:           ORG   0040H             ; Program
                ACALL INITIAL           ; Initial program
BEGIN:          ACALL MAIN              ; Main Program

;-------------------------------------------------------------------
; Initial Program
;-------------------------------------------------------------------
INITIAL:    MOV   TMOD,#021H        ; T1 8Bit Auto, T0 16Bit
            MOV   TH1,#0FDH         ; 9600 bps Timer1 Default
            MOV   TL1,#0FDH
            MOV   TCON,#00000101B   ;SET IE0 & IE1 TRIGGERED
            MOV   SCON,#050H        ; Mode1 RX Enable
            SETB  TR1               ; Start Timer1
            CLR   RX_OK             ; Clear RX_OK flag
            MOV   R1,#BUFFER        ; Clear BUFFER to start pointer
            MOV   P0,#00000000B     ; Clear Data bus
            SETB  SDA1              ; Clear I2C bus
            SETB  SCL1
            LCALL SMART_CLR
            MOV   P2,#11111111B     ; Clear status keypad
            MOV   ADD_LOW,#020h
            MOV   ADD_HIGH,#SMART_WR
            MOV   TEMP_ADD,#00h
            MOV   IE,#10000100B     ; En. EA, ES,EX1
            ACALL INIT_LCD          ; Call LCD Initial subroutine
            RET

;-------------------------------------------------------------------
; Main Program
;-------------------------------------------------------------------
MAIN:           ACALL INSERT_CARD
SET_DATE_TIME:  ACALL SET_DATE_TIME_?
                MOV   R1,#BUFFER
                SETB  ES            ; Set Enable Interrupt Serial
                MOV   LCD_ADDR,#000H    ; Set Address 00H
                ACALL SET_ADDR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   DPTR,#SCR_SHOW_ID ; Index Pointer ROM to Show LCD
            ACALL WRLINE_LCD  ; 00H-07H,40H-47H(Increase automatic)
LOOP:       ACALL READ_TIME_DATE
            JNB   RX_OK,END_LOOP    ; Wait message received
            MOV   LCD_ADDR,#008H    ; Set Address 00H
            ACALL SET_ADDR_LCD
            ACALL BUFFER_LCD        ; Buffer to LCD
END_LOOP:   AJMP  LOOP              ; Jump to loop
            RET

;---------------------------------------------------------------------
; check Insert card
;---------------------------------------------------------------------
INSERT_CARD:      ACALL READ_TIME_DATE
                  MOV   LCD_ADDR,#000H          ; Set Address 00H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_TITLE_1
                  ACALL WRLINE_LCD
                  JNB   P3.3,INSERT_CARD_END
                  LCALL DELAY_1s                ; Delay
                  ACALL READ_TIME_DATE
                  MOV   LCD_ADDR,#000H          ; Set Address 40H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_TITLE_2
                  ACALL WRLINE_LCD
                  JNB   P3.3,INSERT_CARD_END
                  LCALL DELAY_1s                ; Delay
                  ACALL READ_TIME_DATE
                  MOV   LCD_ADDR,#000H          ; Set Address 00H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_TITLE_3
                  ACALL WRLINE_LCD
                  JNB   P3.3,INSERT_CARD_END
                  LCALL DELAY_1s                ; Delay
                  JB    P3.3,INSERT_CARD
INSERT_CARD_END:  RET

;---------------------------------------------------------------------
; check Eject card
;---------------------------------------------------------------------
EJECT_CARD: ACALL READ_TIME_DATE
            MOV   LCD_ADDR,#000H    ; Set Address 00H
            ACALL SET_ADDR_LCD
            MOV   DPTR,#SCR_TITLE_4 ; Index Pointer ROM to Show LCD
            ACALL WRLINE_LCD  ; 00H-07H,40H-47H (Increase automatic)
            JB    P3.3,EJECT_CARD_END
            LCALL DELAY_1s          ; Delay
            ACALL READ_TIME_DATE
            MOV   LCD_ADDR,#000H    ; Set Address 40H
            ACALL SET_ADDR_LCD
            MOV   DPTR,#SCR_TITLE_5 ; Index Pointer ROM to Show LCD
            ACALL WRLINE_LCD        ; 40H-47H (Increase automatic)
            JB    P3.3,EJECT_CARD_END
            LCALL DELAY_1s          ; Delay
            ACALL READ_TIME_DATE
            MOV   LCD_ADDR,#000H    ; Set Address 00H
            ACALL SET_ADDR_LCD
            MOV   DPTR,#SCR_TITLE_2 ; Index Pointer ROM to Show LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ACALL WRLINE_LCD  ; 00H-07H,40H-47H(Increase automatic)
            JB    P3.3,EJECT_CARD_END
            LCALL DELAY_1s          ; Delay
            JNB   P3.3,EJECT_CARD
EJECT_CARD_END:   RET

;-----------------------------------------------------------------------
; SET date time with RTC?
;-----------------------------------------------------------------------
SET_DATE_TIME_?:  MOV   LCD_ADDR,#000H    ; Set Address 00H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_DATE
                  ACALL WRLINE_LCD
                  MOV   LCD_ADDR,#040H    ; Set Address 40H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_YES_NO
                  ACALL WRLINE_LCD

PASS_AGIAN_DATE:  ACALL WAIT_KEYPRESSED   ; Wait until Keypad pressed
                  MOV   A,KPAD_DATA       ; Copy in BUFFER
                  CJNE  A,#10,PASS_NO_DATE; Check keypad pressed?
                  AJMP  SET_DATE
PASS_NO_DATE:     CJNE  A,#12,PASS_AGIAN_DATE
                  AJMP  SET_TIME_YES_NO
SET_DATE:         ACALL SET_DATE_RTC
SET_DATE_SURE:    MOV   LCD_ADDR,#000H    ; Set Address 00H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_SURE
                  ACALL WRLINE_LCD
                  MOV   LCD_ADDR,#040H    ; Set Address 40H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_OK_CANCLE
                  ACALL WRLINE_LCD

PASS_OK_DATE:     ACALL WAIT_KEYPRESSED   ; Wait until Keypad pressed
                  MOV   A,KPAD_DATA       ; Copy in BUFFER
                  CJNE  A,#10,PASS_CANCLE_DATE
                  AJMP  SET_TIME_YES_NO
PASS_CANCLE_DATE: CJNE  A,#12,PASS_OK_DATE; Check keypad pressed?
                  AJMP  SET_DATE

SET_TIME_YES_NO:  MOV   LCD_ADDR,#000H    ; Set Address 00H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_TIME
                  ACALL WRLINE_LCD
                  MOV   LCD_ADDR,#040H    ; Set Address 40H
                  ACALL SET_ADDR_LCD
                  MOV   DPTR,#SCR_YES_NO
                  ACALL WRLINE_LCD

PASS_AGIAN_TIME:  ACALL WAIT_KEYPRESSED   ; Wait until Keypad pressed
                  MOV   A,KPAD_DATA       ; Copy in BUFFER
                  CJNE  A,#10,PASS_NO_TIME
                  AJMP  SET_TIME

PASS_NO_TIME:     CJNE  A,#12,PASS_AGIAN_TIME
                  AJMP  SET_DATETIME_END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SET_TIME:          ACALL SET_TIME_RTC

SET_TIME_SURE:     MOV   LCD_ADDR,#000H    ; Set Address 00H
                   ACALL SET_ADDR_LCD
                   MOV   DPTR,#SCR_SURE
                   ACALL WRLINE_LCD

                   MOV   LCD_ADDR,#040H    ; Set Address 40H
                   ACALL SET_ADDR_LCD
                   MOV   DPTR,#SCR_OK_CANCLE
                   ACALL WRLINE_LCD

PASS_OK_TIME:      ACALL WAIT_KEYPRESSED
                   MOV   A,KPAD_DATA       ; Copy in Buffer
                   CJNE  A,#10,PASS_CANCLE_TIME
                   AJMP  SET_DATETIME_END
PASS_CANCLE_TIME:  CJNE  A,#12,PASS_OK_TIME
                   AJMP  SET_TIME

SET_DATETIME_END:  ACALL SET_DATETIME_RTC
                   RET

;-------------------------------------------------------------------
; SET date with RTC
;-------------------------------------------------------------------
SET_DATE_RTC:      MOV   LCD_ADDR,#000H        ; Set Address 00H
                   ACALL SET_ADDR_LCD
                   MOV   DPTR,#SCR_SET_DATE
                   ACALL WRLINE_LCD

                   MOV   LCD_ADDR,#040H        ; Set Address 40H
                   ACALL SET_ADDR_LCD
                   MOV   DPTR,#SCR_SHOW_DATE
                   ACALL WRLINE_LCD

                   MOV   LCD_ADDR,#040H        ; Set Address 40H
                   ACALL SET_ADDR_LCD
                   ACALL LCD_BLINK             ; Blink LCD

                   ACALL WAIT_KEYPRESSED
                   MOV   BUFFER,KPAD_DATA      ; Copy in Buffer

                   ACALL KPAD2LCD              ; Show Keypad on LCD
                   ACALL LCD_BLINK             ; Blink LCD
                   ACALL WAIT_KEY
                   ACALL WAIT_KEYPRESSED
                   MOV   BUFFER+1,KPAD_DATA    ; Copy in BUFFER+1
                   ACALL KPAD2LCD              ; Show Keypad on LCD
                   ACALL WAIT_KEY
                   ACALL BUFFER2ACC            ; Get 2 buffer to ACC
                   MOV   DATE,A                ; Keep Date Data

                   MOV   LCD_ADDR,#043H        ; Set Address 43H
                   ACALL SET_ADDR_LCD
                   ACALL LCD_BLINK             ; Blink LCD

                   ACALL WAIT_KEYPRESSED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   BUFFER,KPAD_DATA          ; Copy in Buffer

                ACALL KPAD2LCD                  ; Show Keypad on LCD
                ACALL LCD_BLINK                 ; Blink LCD
                ACALL WAIT_KEY
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER+1,KPAD_DATA        ; Copy in Buffer+1

                ACALL KPAD2LCD                  ; Show Keypad on LCD
                ACALL WAIT_KEY
                ACALL BUFFER2ACC                ; Get 2 buffer to ACC
                MOV   MONTH,A                   ; Keep Month Data

                MOV   LCD_ADDR,#046H            ; Set Address 46H
                ACALL SET_ADDR_LCD
                ACALL LCD_BLINK                 ; Blink LCD

                ACALL WAIT_KEYPRESSED
                MOV   BUFFER,KPAD_DATA          ; Copy in Buffer

                ACALL KPAD2LCD                  ; Show Keypad on LCD
                ACALL LCD_BLINK                 ; Blink LCD
                ACALL WAIT_KEY
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER+1,KPAD_DATA        ; Copy in Buffer+1

                ACALL KPAD2LCD                  ; Show Keypad on LCD
                ACALL WAIT_KEY
                ACALL BUFFER2ACC                ; Get 2 buffer to ACC
                MOV   YEAR,A                    ; Keep Year Data
                LCALL DELAY_1s                  ; Delay
                RET
;-----------------------------------------------------------------------
; SET time with RTC
;-----------------------------------------------------------------------
SET_TIME_RTC:   MOV   LCD_ADDR,#000H            ; Set Address 00H
                ACALL SET_ADDR_LCD
                MOV   DPTR,#SCR_SET_TIME
                ACALL WRLINE_LCD

                MOV   LCD_ADDR,#040H            ; Set Address 40H
                ACALL SET_ADDR_LCD
                MOV   DPTR,#SCR_SHOW_TIME
                ACALL WRLINE_LCD
                MOV   LCD_ADDR,#040H            ; Set Address 40H

                ACALL SET_ADDR_LCD
                ACALL LCD_BLINK                 ; Blink LCD
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER,KPAD_DATA          ; Copy in BUFFER
                ACALL KPAD2LCD                  ; Show Keypad on LCD
                ACALL LCD_BLINK                 ; Blink LCD
                ACALL WAIT_KEY
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER+1,KPAD_DATA        ; Copy in BUFFER+1
                ACALL KPAD2LCD                  ; Show Keypad on LCD
                ACALL WAIT_KEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL BUFFER2ACC            ; Get 2 buffer to ACC
                MOV   HOURS,A               ; Keep Hours Data
                MOV   LCD_ADDR,#043H        ; Set Address 43H
                ACALL SET_ADDR_LCD
                ACALL LCD_BLINK             ; Blink LCD
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER,KPAD_DATA      ; Copy in BUFFER
                ACALL KPAD2LCD              ; Show Keypad on LCD
                ACALL LCD_BLINK             ; Blink LCD
                ACALL WAIT_KEY
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER+1,KPAD_DATA    ; Copy in Buffer+1
                ACALL KPAD2LCD              ; Show Keypad on LCD
                ACALL WAIT_KEY
                ACALL BUFFER2ACC            ; Get 2 buffer to ACC
                MOV   MINUTES,A             ; Keep Minutes Data
                MOV   LCD_ADDR,#046H        ; Set Address 46H
                ACALL SET_ADDR_LCD
                ACALL LCD_BLINK             ; Blink LCD
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER,KPAD_DATA      ; Copy in BUFFER
                ACALL KPAD2LCD              ; Show Keypad on LCD
                ACALL LCD_BLINK             ; Blink LCD
                ACALL WAIT_KEY
                ACALL WAIT_KEYPRESSED
                MOV   BUFFER+1,KPAD_DATA    ;Copy in Buffer+1
                ACALL KPAD2LCD              ; Show Keypad on LCD
                ACALL WAIT_KEY
                ACALL BUFFER2ACC            ; Get 2 buffer to ACC
                MOV   SECONDS,A             ; Keep Seconds Data
                LCALL DELAY_1s              ; Delay
                RET

;-----------------------------------------------------------------
; set date & time into RTC & smartcard
;-----------------------------------------------------------------
SET_DATETIME_RTC: ACALL RTC_WR              ; Write RTC
                ACALL READ_TIME_DATE
                ACALL WRITE_DATE_TIME
                RET

;-----------------------------------------------------------------
; Write Date and time to LCD Module
;-----------------------------------------------------------------
READ_TIME_DATE:   ACALL RTC_RD              ; Read RTC

                MOV   LCD_ADDR,#040H        ; Set Address 00H
                ACALL SET_ADDR_LCD

                MOV   LCD_DATA,#' '         ; Write space to LCD
                ACALL WRCHAR_LCD

tDate:          MOV   LCD_DATA,DATE         ; Write Date to LCD
                ACALL BCD2LCD
                MOV   LCD_DATA,#'/'         ; Write space to LCD
                ACALL WRCHAR_LCD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
tMonth:             MOV   LCD_DATA,MONTH    ; Write Month to LCD
                    ACALL BCD2LCD
                    MOV   LCD_DATA,#'/'     ; Write space to LCD
                    ACALL WRCHAR_LCD

tYear:              MOV   LCD_DATA,YEAR     ; Write Year to LCD
                    ACALL BCD2LCD
                    MOV   LCD_DATA,#' '     ; Write space to LCD
                    ACALL WRCHAR_LCD

                    MOV   A,HOURS           ; Get Hour
                    ANL   A,#00110000B      ; Get x10 Digit
                    JZ    WRITE_TIME_HN
                    SWAP  A
                    ADD   A,#030H           ; Convert to ASCII
                    AJMP  WRITE_TIME_HH
WRITE_TIME_HN:      MOV   A,#' '            ; x10 = 0 then Write space
WRITE_TIME_HH:      MOV   LCD_DATA,A        ; Write to LCD
                    ACALL WRCHAR_LCD
                    MOV   A,HOURS
                    ANL   A,#00001111B
                    ADD   A,#030H           ; Convert to ASCII
                    MOV   LCD_DATA,A
                    ACALL WRCHAR_LCD
                    MOV   A,SECONDS
                    ANL   A,#001H
                    JNZ   WRITE_SPACE       ; Even => Write space
                    MOV   LCD_DATA,#':'     ; Odd => Write ':'
                    ACALL WRCHAR_LCD
                    AJMP  WRITE_MINUTES
WRITE_SPACE:        MOV   LCD_DATA,#' '
                    ACALL WRCHAR_LCD

WRITE_MINUTES:      MOV   LCD_DATA,MINUTES  ; Write Minutes to LCD
                    ACALL BCD2LCD           ; Write from RTC BCD Data
                    RET
;-------------------------------------------------------------------
; Write RTC into Smartcard
;-------------------------------------------------------------------
WRITE_DATE_TIME:    MOV   SMART_DATA,DATE
                    ACALL BCD2SMART

                    MOV   SMART_DATA,MONTH
                    ACALL BCD2SMART

                    MOV   SMART_DATA,YEAR
                    ACALL BCD2SMART

                    MOV   SMART_DATA,HOURS
                    ACALL BCD2SMART

                    MOV   SMART_DATA,MINUTES
                    ACALL BCD2SMART

                    MOV   SMART_DATA,SECONDS
                    ACALL BCD2SMART

                    MOV   SMART_DATA,#'|'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL SMART_WRITE
                LCALL SMART_DELAY

                LCALL SMART_CLR
                RET
;-------------------------------------------------------------------
; Write Barcode_data and RTC into Smartcard
;-------------------------------------------------------------------
WRITE_BAR_TIME:   PUSH  02H
                  MOV   02H,#08
                  MOV   R1,#BUFFER

WRITE_BAR_AGIAN:  MOV   A,@R1
                  INC   R1
                  MOV   SMART_DATA,A      ; Write space to LCD
                  ACALL SMART_WRITE
                  LCALL SMART_DELAY
                  LCALL SMART_DELAY
                  DJNZ  02H,WRITE_BAR_AGIAN
                  POP   02H

                  MOV   SMART_DATA,#'|'   ; Write space to LCD
                  ACALL SMART_WRITE
                  LCALL SMART_DELAY
                  LCALL SMART_DELAY

                  MOV   SMART_DATA,HOURS
                  ACALL BCD2SMART

                  MOV   SMART_DATA,MINUTES
                  ACALL BCD2SMART

                  MOV   SMART_DATA,SECONDS
                  ACALL BCD2SMART

                  MOV   SMART_DATA,#'|'
                  ACALL SMART_WRITE
                  LCALL SMART_DELAY

                  LCALL SMART_CLR
                  RET

;-------------------------------------------------------------------
; BCD Code to show Smart
;-------------------------------------------------------------------
BCD2SMART:        PUSH  ACC               ; Push ACC. To Stack
                  PUSH  B                 ; Push B to Stack
                  MOV   A,SMART_DATA      ; Get input data value
                  MOV   B,A               ; Copy to B
                  ANL   A,#11110000B      ; Get higher 4 bit
                  SWAP  A                 ; Swap nibble
                  ADD   A,#030H           ; Convert to ASCII
                  MOV   SMART_DATA,A
                  LCALL SMART_WRITE
                  LCALL SMART_DELAY
                  LCALL SMART_DELAY
                  MOV   A,B               ; Restore value
                  ANL   A,#00001111B      ; Get lower 4 bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ADD   A,#030H           ; Convert to ASCII
                MOV   SMART_DATA,A
                LCALL SMART_WRITE
                LCALL SMART_DELAY
                LCALL SMART_DELAY
                POP   B                 ; Pop B from Stack
                POP   ACC               ; Pop ACC. from Stack
                RET                     ; Return

;-------------------------------------------------------------------
; Check for Change Page
;-------------------------------------------------------------------
CHECK_CHANGE:   PUSH  ACC
                CLR   C

                MOV   A,#03
                SUBB  A,TEMP_ADD
                JZ    CHANGE_CARD
                SJMP  CHANGE_PAGE

CHANGE_CARD:    MOV   A,#0E0H
                SUBB  A,ADD_LOW
                JZ    CHANGE_CARD1

CHANGE_PAGE:    MOV   A,ADD_LOW
                JZ    CHANGE_PAGE1
                MOV   A,#0FDH
                SUBB  A,ADD_LOW
                JZ    CHANGE_PAGE1
                AJMP  CHECK_END

CHANGE_PAGE1:   MOV   ADD_LOW,#020h
                MOV   ADD_HIGH,#SMART_WR
                INC   TEMP_ADD
                MOV   A,TEMP_ADD
                RR    A
                RR    A
                ORL   A,ADD_HIGH
                MOV   ADD_HIGH,A
                AJMP  CHECK_END

CHANGE_CARD1:   ACALL EJECT_CARD
NEW_CARD:       ACALL INSERT_CARD
NEW_INIT:       MOV   ADD_LOW,#020h
                MOV   ADD_HIGH,#SMART_WR
                MOV   TEMP_ADD,#00h
                ACALL SET_DATETIME_RTC
                MOV   LCD_ADDR,#000H    ; Set Address 00H
                ACALL SET_ADDR_LCD
                MOV   DPTR,#SCR_SHOW_ID
                ACALL WRLINE_LCD
CHECK_END:      POP         ACC
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------
; Serial interrupt subroutine
;-------------------------------------------------------------------
EX1_INT:          LCALL SMART_DELAY
                  LCALL SMART_ATR
                  LCALL SMART_WRITE_INIT
                  LCALL SMART_CLR
                  RETI

;-------------------------------------------------------------------
; BARCODE CARD SLOTER
;-------------------------------------------------------------------
;-------------------------------------------------------------------
; Serial interrupt subroutine
;-------------------------------------------------------------------
SER_INT:    PUSH  ACC                 ; Push ACC. to stack

            JBC   RI,SER_RX
            SJMP  SER_EXIT

SER_RX:     CLR   RX_OK               ; Clear RI
            MOV   A,SBUF              ; Get data to ACC.
            MOV   @R1,A               ; Move to Buffer @R1
            INC   R1                  ; Increase R1
            CJNE  A,#00AH,SER_EXIT    ; Check 00DH End of Buffer text
            SETB  RX_OK               ; Set RX_OK received message ok.

SER_EXIT:   POP   ACC                 ; Pop Acc. to Main
            RETI                      ; Return interrupt

;-------------------------------------------------------------------
; Buffer Text from ROM Pointer to LCD
;-------------------------------------------------------------------
BUFFER_LCD:       MOV   R1,#BUFFER
BUFFER_LCD_1:     MOV   A,@R1
                  INC   R1                ; Increase buffer pointer
                  CJNE  A,#00AH,TX_BUFFER_CHAR
                  ACALL CHECK_CHANGE
                  ACALL WRITE_BAR_TIME
                  MOV   R1,#BUFFER
                  CLR   RX_OK
                  RET                     ; End => Return

TX_BUFFER_CHAR:   MOV   LCD_DATA,A        ; Write space to LCD
                  ACALL WRCHAR_LCD
                  AJMP  BUFFER_LCD_1      ; Wait for TX

;-------------------------------------------------------------------
; KEYPAD Module
;-------------------------------------------------------------------
;-------------------------------------------------------------------
; BUFFER to ACC
; I/P:            BUFFER,BUFFER+1
; O/P:            ACC
;-------------------------------------------------------------------
BUFFER2ACC:       MOV   A,BUFFER          ; Restore BUFFER
                  ANL   A,#00FH           ; Get lower 4 bit
                  SWAP  A                 ; Swap nibble
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   B,A                    ; Move to B
                MOV   A,BUFFER+1             ; Restore BUFFER+1
                ANL   A,#00FH                ; Get lower 4 bit
                ADD   A,B
                RET

;-------------------------------------------------------------------
; KEYPAD to show LCD
; I/P:          KPAD_DATA
;-------------------------------------------------------------------
KPAD2LCD:   MOV   A,KPAD_DATA                ; Get Keypad Data
            ADD   A,#030H                    ; Convert to ASCII
            MOV   LCD_DATA,A                 ; Write to lCD
            ACALL WRCHAR_LCD                 ;
            RET

;-------------------------------------------------------------------
; Wait keypad depressed
;-------------------------------------------------------------------
WAIT_KEY:   MOV   A,P2                       ; Move P2 to ACC.
            ANL   A,#00FH                    ; Get only Lower 4 bit
            CJNE  A,#00FH,WAIT_KEY           ; All bit are high ?
            RET

;-------------------------------------------------------------------
; Wait keypad pressed 0-9 Only
;-------------------------------------------------------------------
WAIT_KEYPRESSED:  ACALL GET_KPAD             ; Check keypad Data
                  MOV   A,KPAD_DATA
                  LCALL DELAY_100ms
                  LCALL DELAY_100ms
                  CJNE  A,#0,CHK_KEY_STAR ; Check keypad pressed?
                  AJMP  WAIT_KEYPRESSED   ; Loop until keypad pressed
CHK_KEY_STAR:     CJNE  A,#10,CHK_KEY_0   ; Check key *
                  RET
CHK_KEY_0:        CJNE  A,#11,CHK_KEY_SHAP; Check key 0
                  MOV   KPAD_DATA,#0      ; Replace KPAD_DATA = 0
                  RET                     ; Return
CHK_KEY_SHAP:     CJNE  A,#12,CHK_VALID_KEY     ; Check key *
                  RET
CHK_VALID_KEY:    JNC   WAIT_KEYPRESSED
                  RET                              ; Return

;-------------------------------------------------------------------
; Keypad Scan key Subroutine
;-------------------------------------------------------------------
GET_KPAD:   MOV   P2,#0FFH                   ; Pull P2 to High
            MOV   KPAD_DATA,#0               ; Clear Keypad Data

CHK_COL0:   CLR   KPAD_COL0                  ; Begin Scan Column 0
            MOV   A,P2                       ; Get Port2 Value
            ANL   A,#00FH                    ; Get only lower 4 bit
            CJNE  A,#00FH,COL0_DETECT        ; Check All rows '1'?
            AJMP  CHK_COL1
COL0_DETECT:MOV   KPAD_DATA,#01              ; Initial KPAD_DATA = 1
            AJMP  GET_ROW                    ; Jump to get row value

CHK_COL1:   SETB  KPAD_COL0                  ; Stop Scan Column 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CLR   KPAD_COL1                  ; Begin Scan Column 1
            MOV   A,P2                       ; Get Port2 Value
            ANL   A,#00FH                    ; Get only lower 4 bit
            CJNE  A,#00FH,COL1_DETECT        ; Check All rows '1'?
            AJMP  CHK_COL2

COL1_DETECT:      MOV   KPAD_DATA,#02        ; Initial KPAD_DATA = 2
                  AJMP  GET_ROW              ; Jump to get row value

CHK_COL2:         SETB  KPAD_COL1            ; Stop Scan Column 1
                  CLR   KPAD_COL2            ; Begin Scan Column 2
                  MOV   A,P2                 ; Get Port2 Value
                  ANL   A,#00FH              ; Get only lower 4 bit
                  CJNE  A,#00FH,COL2_DETECT      ; Check All rows '1'?
                  RET                        ; All rows '1' => return

COL2_DETECT:      MOV   KPAD_DATA,#03        ; Initial KPAD_DATA = 2

GET_ROW:          CLR   KPAD_COL0
                  CLR   KPAD_COL1
                  CLR   PAD_COL2

                  JB    KPAD_ROW0,CHK_ROW1       ; Check Row 0 Detect?
                  RET                        ; Row 0 Detect => return

CHK_ROW1:         JB    KPAD_ROW1,CHK_ROW2       ; Check Row 2 Detect?
                  MOV   A,KPAD_DATA          ; Add 3 with KPAD_DATA
                  ADD   A,#3
                  MOV   KPAD_DATA,A
                  RET                        ; Return

CHK_ROW2:         JB    KPAD_ROW2,CHK_ROW3       ; Check Row 2 Detect?
                  MOV   A,KPAD_DATA            ; Add 6 with
KPAD_DATA
                  ADD   A,#6
                  MOV   KPAD_DATA,A
                  RET                          ; Return

CHK_ROW3:         MOV   A,KPAD_DATA            ; Add 9 with
KPAD_DATA
                  ADD   A,#9
                  MOV   KPAD_DATA,A
                  RET                        ; Return

;-------------------------------------------------------------------
; LCD Module Display
;-------------------------------------------------------------------
;-------------------------------------------------------------------
; BCD Code to show LCD
; I/P:            LCD_DATA
;-------------------------------------------------------------------
BCD2LCD:          PUSH  ACC                ; Push ACC. To Stack
                  PUSH  B                  ; Push B to Stack
                  MOV   A,LCD_DATA         ; Get input data value
                  MOV   B,A                ; Copy to B
                  ANL   A,#11110000B       ; Get higher 4 bit
                  SWAP  A                  ; Swap nibble
                  ADD   A,#030H            ; Convert to ASCII
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV   LCD_DATA,A        ; Write LCD
                ACALL WRCHAR_LCD
                MOV   A,B               ; Restore value
                ANL   A,#00001111B      ; Get lower 4 bit
                ADD   A,#030H           ; Convert to ASCII
                MOV   LCD_DATA,A        ; Write LCD
                ACALL WRCHAR_LCD
                POP   B                 ; Pop B from Stack
                POP   ACC               ; Pop ACC. from Stack
                RET                     ; Return

;-----------------------------------------------------------------
; LCD Initialize
;-----------------------------------------------------------------
INIT_LCD:       LCALL DELAY_100ms       ; Delay
                CLR   LCD_RS            ; Clear LCD_RS Pin

                MOV   P0,#00111000B     ; 8bit Mode
                ACALL LCD_CLK           ; Pulse LCD Clock
                LCALL DELAY_10ms        ; Delay

                MOV   P0,#00111000B     ; 8bit Mode
                ACALL LCD_CLK           ; Pulse LCD Clock

                ACALL LCD_OFF           ; Display Off

                ACALL LCD_CLR           ; Clear Display

                MOV   P0,#00000110B     ; Entry Mode
                ACALL LCD_CLK           ; Pulse LCD Clock
                ACALL LCD_HOME          ; Return Home Display

;-----------------------------------------------------------------
; LCD Module Display
;-----------------------------------------------------------------
;-----------------------------------------------------------------
; LCD Clear Display
;-----------------------------------------------------------------
LCD_CLR:        CLR   LCD_RS            ; Clear LCD_RS Pin
                MOV   P0,#00000001B     ; Display Clear
                ACALL LCD_CLK           ; Pulse LCD Clock
                RET

;-----------------------------------------------------------------
; LCD Return Home
;-----------------------------------------------------------------
LCD_HOME:       CLR   LCD_RS            ; Clear LCD_RS Pin
                MOV   P0,#00000010B     ; Return Home
                ACALL LCD_CLK           ; Pulse LCD Clock
                RET

;-----------------------------------------------------------------
; LCD Display Off
;-----------------------------------------------------------------
LCD_OFF:        CLR   LCD_RS            ; Clear LCD_RS Pin
                MOV   P0,#00001000B     ; Display Off
                ACALL LCD_CLK           ; Pulse LCD Clock
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------------------------------------------------------
; LCD CLK
;-----------------------------------------------------------------------
LCD_CLK:    SETB  LCD_EN                ; Pulse Clock to LCD_EN
            LCALL LCD_DELAY
            CLR   LCD_EN
            LCALL LCD_DELAY
            RET

;-----------------------------------------------------------------------
; LCD Display On
;-----------------------------------------------------------------------
LCD_ON:     CLR   LCD_RS                ; Clear LCD_RS Pin
            MOV   P0,#00001100B         ; Display On
            ACALL LCD_CLK               ; Pulse LCD Clock
            RET

;-----------------------------------------------------------------------
; LCD Cursor On
;-----------------------------------------------------------------------
LCD_BLINK:  CLR   LCD_RS                ; Clear LCD_RS Pin
            MOV   P0,#00001111B         ; Display Cursor and Blink
            ACALL LCD_CLK               ; Pulse LCD Clock
            RET

;-----------------------------------------------------------------------
; LCD Left Shift Display
;-----------------------------------------------------------------------
LCD_LSHF:   CLR   LCD_RS                ; Clear LCD_RS Pin
            MOV   P0,#00011000B         ; Left Shift Display
            ACALL LCD_CLK               ; Pulse LCD Clock
            RET

;-----------------------------------------------------------------------
; LCD Right Shift Display
;-----------------------------------------------------------------------
LCD_RSHF:   CLR   LCD_RS                ; Clear LCD_RS Pin
            MOV   P0,#00011100B         ; Right Shift Display
            ACALL LCD_CLK               ; Pulse LCD Clock
            RET

;-----------------------------------------------------------------------
; Set LCD Address
; I/P:      LCD_ADDR
;-----------------------------------------------------------------------
SET_ADDR_LCD:   CLR   LCD_RS            ; Clear LCD_RS Pin
                MOV   A,LCD_ADDR        ; Move LCD_ADDR to ACC.
                SETB  ACC.7             ; Set bit ACC.7
                MOV   P0,A              ; Move to DATABUS
                ACALL LCD_CLK           ; Pulse LCD Clock
                RET

;-----------------------------------------------------------------------
; Write Character to show LCD
; I/P:          LCD_DATA
;-----------------------------------------------------------------------
WRCHAR_LCD:     SETB  LCD_RS            ; Set LCD_RS Pin
                MOV   P0,LCD_DATA       ; Move LCD_DATA to DATABUS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ACALL LCD_CLK           ; Pulse LCD Clock
                ACALL LCD_ON            ; Display On
                RET

;-----------------------------------------------------------------
; Write Line of 16 Characters from ROM
; I/P:          DPTR: Locate ROM Address
;-----------------------------------------------------------------
WRLINE_LCD:     MOV   R0,#0             ; Clear loop counter
WRLINE_LCD_1:   SETB  LCD_RS            ; Set LCD_RS Pin
                CLR   A                 ; Clear ACC.
                MOVC  A,@A+DPTR
                MOV   P0,A              ; Move ACC. to DATABUS
                ACALL LCD_CLK           ; Pulse LCD Clock
                INC   DPTR              ; Increase Pointer
                INC   R0                ; Increase loop counter
                CJNE  R0,#16,WRLINE_LCD_1     ; Do until 8 times
                ACALL LCD_ON                  ; Display On
                RET

;-----------------------------------------------------------------
; I2C Real Time Clock DS1307
;-----------------------------------------------------------------
;-----------------------------------------------------------------
; I2C RTC Read
;-----------------------------------------------------------------
RTC_RD:         MOV   I2C_ADDR,#RTC_ID
                LCALL I2C_SLAVE         ; Connect Slave

                MOV   I2C_DATA,#000H    ; Set Slave Address 00H
                LCALL I2C_DATA_WR       ; Write Data to Slave

                MOV   I2C_ADDR,#RTC_ID+1; Set RTC as I2C Read Slave
                LCALL I2C_SLAVE         ; Connect Slave

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   SECONDS,I2C_DATA  ; Read Data to SECONDS
                LCALL I2C_ACK_BIT       ; Send Acknowledge

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   MINUTES,I2C_DATA  ; Read Data to MINUTES
                LCALL I2C_ACK_BIT       ; Send Acknowledge

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   HOURS,I2C_DATA    ; Read Data to HOURS
                LCALL I2C_ACK_BIT       ; Send Acknowledge

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   DAY,I2C_DATA      ; Read Data to DAY
                LCALL I2C_ACK_BIT       ; Send Acknowledge

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   DATE,I2C_DATA     ; Read Data to DATE
                LCALL I2C_ACK_BIT       ; Send Acknowledge

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   MONTH,I2C_DATA    ; Read Data to MONTH
                LCALL I2C_ACK_BIT       ; Send Acknowledge
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   YEAR,I2C_DATA     ; Read Data to YEAR
                LCALL I2C_ACK_BIT       ; Send Acknowledge

                LCALL I2C_DATA_RD       ; Read Data from Slave
                MOV   CONTROL,I2C_DATA  ; Read Data to CONTROL
                LCALL I2C_NACK_BIT      ; Send Not Acknowledge

                LCALL I2C_STOP          ; Send Stop Condition
                RET                     ; Return

;-------------------------------------------------------------------
; I2C RTC Write
;-------------------------------------------------------------------
RTC_WR:     MOV   I2C_ADDR,#RTC_ID
            LCALL I2C_SLAVE             ; Connect Slave

            MOV   I2C_DATA,#000H        ; Set Slave Address 00H
            LCALL I2C_DATA_WR           ; Write Data to Slave

            MOV   I2C_DATA,SECONDS      ; Write SECONDS to RTC
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,MINUTES      ; Write MINUTES to RTC
            LCALL I2C_DATA_WR
            MOV   I2C_DATA,HOURS        ; Write HOURS to RTC
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,DAY          ; Write DAY to RTC
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,DATE         ; Write DATE to RTC
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,MONTH        ; Write MONTH to RTC
            LCALL I2C_DATA_WR

            MOV   I2C_DATA,YEAR         ; Write YEAR to RTC
            LCALL I2C_DATA_WR           ;

            MOV   I2C_DATA,CONTROL      ; Write CONTROL to RTC
            LCALL I2C_DATA_WR           ;

            LCALL I2C_STOP              ; Send Stop Condition
            RET                         ; Return

;-------------------------------------------------------------------
; I2C Data Write
; I/P:          I2C_DATA
; Reserve:  R5
;-------------------------------------------------------------------
I2C_DATA_WR:    PUSH  ACC           ; Push ACC.
                SETB  I2C_ACK       ; Set ACK. bit
                MOV   A,I2C_DATA    ; Get Data
                MOV   R5,#008       ; Set loop 8 times
I2C_DATA_WR_1:  RLC   A             ; Rotate ACC. to Left with Carry
                MOV   SDA1,C        ; Move Carry Flag to SDA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    ACALL I2C_CLK           ; Pulse I2C Clock
                    DJNZ  R5,I2C_DATA_WR_1  ; Do until 8 times
                    SETB  SDA1              ; Set SDA
                    LCALL I2C_DELAY         ; Delay
                    SETB  SCL1              ; Set SCL
                    LCALL I2C_DELAY         ; Delay
                    JB    SDA1,I2C_DATA_WR_2
                    CLR   I2C_ACK           ; Clear ACK. bit

I2C_DATA_WR_2:      CLR   SCL1              ; Clear SCL
                    POP   ACC               ; Pop ACC.
                    RET                     ; Return

;-------------------------------------------------------------------
; I2C Data Read
; O/P:              I2C_DATA
; Reserve:  R5
;-------------------------------------------------------------------
I2C_DATA_RD:        PUSH  ACC           ; Push ACC.
                    CLR   A             ; Clear ACC.
                    MOV   R5,#008       ; Set loop 8 times
I2C_DATA_RD_1:      LCALL I2C_DELAY     ; Delay
                    SETB  SCL1          ; Set SCL
                    LCALL I2C_DELAY     ; Delay
                    MOV   C,SDA1        ; Get SDA to Carry Flag
                    RLC   A             ; Rotate ACC. to Left with Carry
                    CLR   SCL1          ; Clear SCL
                    DJNZ  R5,I2C_DATA_RD_1  ; Do until 8 times
                    MOV   I2C_DATA,A        ; Move Data to I2C_DATA
                    POP   ACC               ; Pop ACC
                    RET                     ; Return

;-------------------------------------------------------------------
; I2C Slave Connect
; I/P:              I2C_ADDR
; O/P Flag: I2C_ACK
; Reserve:  R5
;-------------------------------------------------------------------
I2C_SLAVE:          PUSH  ACC               ; Push ACC.
                    SETB  I2C_ACK           ; Set ACK. bit
                    MOV   A,I2C_ADDR        ; Get Slave Address
                    ACALL I2C_START         ; Send Start Condition
                    MOV   R5,#008           ; Set loop 8 times
I2C_SLAVE_1:        RLC   A             ; Rotate ACC. to Left with Carry
                    MOV   SDA1,C        ; Move Carry Flag to SDA
                    ACALL I2C_CLK       ; Pulse I2C Clock
                    DJNZ  R5,I2C_SLAVE_1    ; Do until 8 times

                    SETB  SDA1              ; Set SDA
                    LCALL I2C_DELAY         ; Delay
                    SETB  SCL1              ; Set SCL
                    LCALL I2C_DELAY         ; Delay
                    JB    SDA1,I2C_SLAVE_2
                    CLR   I2C_ACK           ; Clear ACK.
I2C_SLAVE_2:        CLR   SCL1              ; Clear SCL
                    POP   ACC               ; Pop ACC.
                    RET                     ; Return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------------------------------------------------
; I2C Start Condition
;-----------------------------------------------------------------
I2C_START:          JNB   SCL1,I2C_START_1  ; Check current SCL set?
                    CLR   SCL1              ; Clear SCL

I2C_START_1:        SETB  SDA1              ; Set SDA
                    SETB  SCL1              ; Set SCL

                    LCALL I2C_DELAY         ; Delay
                    CLR   SDA1              ; Clear SDA during SCL set
                    LCALL I2C_DELAY         ; Delay
                    CLR   SCL1              ; Clear SCL
                    RET                     ; Return

;-----------------------------------------------------------------
; I2C Stop Condition
;-----------------------------------------------------------------
I2C_STOP:           JNB   SCL1,I2C_STOP_1   ; Check current SCL set?
                    CLR   SCL1              ; Clear SCL

I2C_STOP_1:         CLR   SDA1              ; Clear SDA
                    LCALL I2C_DELAY         ; Delay
                    SETB  SCL1              ; Set SCL
                    LCALL I2C_DELAY         ; Delay
                    SETB  SDA1              ; Set SDA during SCL set
                    RET                     ; Return

;-----------------------------------------------------------------
; I2C Clock
;-----------------------------------------------------------------
I2C_CLK:            LCALL I2C_DELAY         ; Pulse SCL
                    SETB  SCL1
                    LCALL I2C_DELAY
                    CLR   SCL1
                    RET                     ; Return

;-----------------------------------------------------------------
; I2C Acknowledge
;-----------------------------------------------------------------
I2C_ACK_BIT:        CLR   SDA1              ; Clear SDA
                    LCALL I2C_DELAY         ; Delay
                    ACALL I2C_CLK           ; Pulse I2C Clock
                    SETB  SDA1
                    RET                     ; Return

;-----------------------------------------------------------------
; I2C Not Acknowledge
;-----------------------------------------------------------------
I2C_NACK_BIT:       SETB  SDA1              ; Set SDA
                    LCALL I2C_DELAY         ; Delay
                    ACALL I2C_CLK           ; Pulse I2C Clock
                    SETB  SCL1
                    RET                     ; Return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************************************************************
; SMART CARD
;*********************************************************************
;---------------------------------------------------------------------
; SMART CARD ATR
;---------------------------------------------------------------------
SMART_ATR:          CLR   SCL
                    SETB  SDA
                    SETB  RST
                    LCALL SMART_CLK
                    LCALL SMART_DELAY
                    LCALL SMART_DELAY
                    LCALL SMART_DELAY
                    CLR   RST
                    LCALL SMART_DELAY
                    MOV   R5,#32
LOOP_CLK_ATR:       SETB  SDA
                    LCALL SMART_CLK
                    DJNZ  R5,LOOP_CLK_ATR
                    LCALL DELAY_100ms
                    RET

;---------------------------------------------------------------------
; SMART CARD Read
;---------------------------------------------------------------------
SMART_READ:         MOV   I2C_ADDR,#SMART_RD
                    LCALL CON_SMART          ; Connect Smart Card
                    MOV   I2C_DATA,ADD_LOW
                    INC   ADD_LOW
                    LCALL SMART_DATA_WR      ; Write Data to Slave
                    MOV   I2C_DATA,#000H     ; Set Slave Address 00H
                    LCALL SMART_DATA_WR
                    LCALL SMART_DELAY
                    CLR   RST
                    LCALL SMART_CLK
SMART_READ_1:       LCALL SMART_DATA_RD      ; Read Data from Slave
                    MOV   SMART_DATA,I2C_DATA
                    RET                      ; Return

;---------------------------------------------------------------------
; SMART CARD Write
;---------------------------------------------------------------------
SMART_WRITE_INIT:   MOV   I2C_ADDR,#11001100B
                    LCALL CON_SMART                ; Connect Smart Card
                    MOV   I2C_DATA,#11111101B
                    LCALL SMART_DATA_WR            ; Write Data to Slave
                    MOV   I2C_DATA,#00000000B
                    LCALL SMART_DATA_WR
                    SETB  SDA
                    LCALL SMART_DELAY
                    CLR   RST
                    LCALL SMART_CLK
                    LCALL SMART_DATA_RD
                    LCALL SMART_CLK
                    LCALL SMART_DATA_RD      ; Read Data from Slave
                    LCALL SMART_CLK
                    LCALL SMART_DATA_RD      ; Read Data from Slave
                    LCALL SMART_CLK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                LCALL SMART_CLR
;-------------------------------------------------------------------
                MOV   I2C_ADDR,#11110010B
                LCALL CON_SMART                 ; Connect Smart Card
                MOV   I2C_DATA,#11111101B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                MOV   I2C_DATA,#11111110B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                LCALL SMART_DELAY
                CLR   RST
                MOV   R5,#103

LOOP_CLK:       SETB  SDA
                LCALL SMART_CLK
                DJNZ  R5,LOOP_CLK
                LCALL SMART_CLR
;-------------------------------------------------------------------
                MOV   I2C_ADDR,#11001101B
                LCALL CON_SMART                 ; Connect Smart Card
                MOV   I2C_DATA,#11111110B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                MOV   I2C_DATA,#11111111B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                LCALL SMART_DELAY
                CLR   RST
                SETB  SDA
                LCALL SMART_CLK
                LCALL SMART_CLK
                LCALL SMART_CLR
;-------------------------------------------------------------------
                MOV   I2C_ADDR,#11001101B
                LCALL CON_SMART                 ; Connect Smart Card
                MOV   I2C_DATA,#11111111B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                MOV   I2C_DATA,#11111111B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                LCALL SMART_DELAY
                CLR   RST
                SETB  SDA
                LCALL SMART_CLK
                LCALL SMART_CLK
                LCALL SMART_CLR
;-------------------------------------------------------------------
                MOV   I2C_ADDR,#11110011B
                LCALL CON_SMART                 ; Connect Smart Card
                MOV   I2C_DATA,#11111101B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                MOV   I2C_DATA,#11111111B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                LCALL SMART_DELAY
                CLR   RST
                MOV   R5,#103

LOOP_CLK_1:     SETB  SDA
                LCALL SMART_CLK
                DJNZ  R5,LOOP_CLK_1
                LCALL SMART_CLR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------
                MOV   I2C_ADDR,#11001100B
                LCALL CON_SMART                 ; Connect Smart Card
                MOV   I2C_DATA,#11111101B
                LCALL SMART_DATA_WR             ; Write Data to Slave
                MOV   I2C_DATA,#00000000B
                LCALL SMART_DATA_WR
                SETB  SDA
                LCALL SMART_DELAY
                CLR   RST
                LCALL SMART_CLK
                LCALL SMART_DATA_RD     ; Read Data from Slave
                LCALL SMART_CLK
                LCALL SMART_DATA_RD     ; Read Data from Slave
                LCALL SMART_CLK
                LCALL SMART_DATA_RD     ; Read Data from Slave
                LCALL SMART_CLK
                LCALL SMART_CLR
                LCALL DELAY_100ms
                LCALL DELAY_100ms
                RET
;-------------------------------------------------------------------
SMART_WRITE:    MOV   I2C_ADDR,ADD_HIGH
                LCALL CON_SMART         ; Connect Smart Card
                MOV   I2C_DATA,ADD_LOW
                INC   ADD_LOW
                LCALL SMART_DATA_WR     ; Write Data to Slave
                MOV   I2C_DATA,SMART_DATA
                LCALL SMART_DATA_WR     ; Write Data to Slave
                SETB  SDA
                LCALL SMART_DELAY
                CLR   RST
                MOV   R5,#3
LOOP_CLK_2:     SETB  SDA
                LCALL SMART_CLK
                DJNZ  R5,LOOP_CLK_2
                RET                     ; Return
;-------------------------------------------------------------------
; SMART CARD Data Write
; I/P: SMART CARD_DATA
; Reserve:  R5
;-------------------------------------------------------------------
SMART_DATA_WR:  PUSH  ACC               ; Push ACC.
                MOV   A,I2C_DATA        ; Get Data
                MOV   R5,#008           ; Set loop 8 times
SMART_DATA_WR_1: RRC  A           ; Rotate ACC. to Left with Carry
                MOV   SDA,C       ; Move Carry Flag to SDA
                LCALL SMART_CLK   ; Pulse I2C Clock
                DJNZ  R5,SMART_DATA_WR_1; Do until 8 times
                POP   ACC               ; Pop ACC.
                RET                     ; Return
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-----------------------------------------------------------------
; SMART CARD Data Read
; O/P:              SMART CARD_DATA
; Reserve:  R5
;-----------------------------------------------------------------
SMART_DATA_RD:     PUSH  ACC          ; Push ACC.
                   SETB  SDA
                   CLR   A            ; Clear ACC.
                   MOV   R5,#008      ; Set loop 8 times

SMART_DATA_RD_1:   LCALL SMART_DELAY ; Delay
                   SETB  SCL          ; Set SCL
                   LCALL SMART_DELAY ; Delay
                   MOV   C,SDA        ; Get SDA to Carry Flag
                   RRC   A            ; Rotate ACC. to Left with Carry
                   CLR   SCL          ; Clear SCL
                   DJNZ  R5,SMART_DATA_RD_1      ; Do until 8 times
                   MOV   I2C_DATA,A          ; Move Data to I2C_DATA
                   POP   ACC                 ; Pop ACC.
                   RET                       ; Return

;-----------------------------------------------------------------
; SMART CARD Connect
; I/P: SMART CARD_ADDR
; O/P Flag: SMART CARD_ACK
; Reserve:  R5
;-----------------------------------------------------------------
CON_SMART:         PUSH  ACC          ; Push ACC.
                   MOV   A,I2C_ADDR   ; Get Slave Address
                   LCALL SMART_START ; Send Start Condition
                   MOV   R5,#008      ; Set loop 8 times

CON_SMART_1:       RRC   A            ; Rotate ACC. to Left with Carry
                   MOV   SDA,C        ; Move Carry Flag to SDA
                   ACALL SMART_CLK    ; Pulse I2C Clock
                   DJNZ  R5,CON_SMART_1     ; Do until 8 times
                   POP   ACC                ; Pop ACC.
                   RET                      ; Return

;-----------------------------------------------------------------
; SMART CARD Start Condition
;-----------------------------------------------------------------
SMART_START:       CLR   SCL                ; Clear SCL
                   CLR   RST                ; Clear RST
                   ACALL SMART_DELAY        ; Delay
                   ACALL SMART_DELAY        ; Delay
                   SETB  RST                ; Set RST
                   SETB  SDA
                   ACALL SMART_DELAY
                   RET                      ; Return

;-----------------------------------------------------------------
; SMART CARD Clock
;-----------------------------------------------------------------
SMART_CLK:         ACALL SMART_DELAY        ; Pulse SCL
                   SETB  SCL
                   NOP
                   ACALL SMART_DELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR   SCL
                NOP
                RET                         ; Return
;-------------------------------------------------------------------
; SMART CARD Clear
;-------------------------------------------------------------------
SMART_CLR:      CLR   SCL
                CLR   SDA
                CLR   RST
                LCALL SMART_DELAY
                RET
;-------------------------------------------------------------------
; Dummy Delay time I2C_DELAY, LCD_DELAY, 10m, 100m, 1s
;-------------------------------------------------------------------
I2C_DELAY:      MOV   R6,#00CH              ; Each loop = 50 us
I2C_DELAY_1:    NOP
                NOP
                DJNZ  R6,I2C_DELAY_1
                RET

SMART_DELAY:    MOV   R6,#07H               ; Each loop = 50 us
SMART_DELAY_1:  NOP
                NOP
                DJNZ  R6,SMART_DELAY_1
                RET

LCD_DELAY:      MOV   R7,#002               ; Do 2 times
LCD_DELAY_1:    MOV   R6,#0E6H              ; Each loop = 1 ms
LCD_DELAY_2:    NOP
                NOP
                DJNZ  R6,LCD_DELAY_2
                DJNZ  R7,LCD_DELAY_1
                RET

DELAY_10ms:     MOV   R7,#010               ; Do 10 times
DELAY_10ms_1:   MOV   R6,#0E6H              ; Each loop = 1 ms
DELAY_10ms_2:   NOP
                NOP
                DJNZ  R6,DELAY_10ms_2
                DJNZ  R7,DELAY_10ms_1
                RET

DELAY_100ms:    MOV   R7,#100               ; Do 100 times
DELAY_100ms_1:  MOV   R6,#0E6H              ; Each loop = 1 ms
DELAY_100ms_2:  NOP
                NOP
                DJNZ  R6,DELAY_100ms_2
                DJNZ  R7,DELAY_100ms_1
                RET

DELAY_1s:       MOV   R5,#100               ; Do 100 times
DELAY_1s_1:     LCALL DELAY_10ms
                DJNZ  R5,DELAY_1s_1
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------------
; Define Constant < Store in Flash EEPROM Program Memory >
;-------------------------------------------------------------------
; 01234567
SCR_TITLE_1:       DB          ' Please  Insert '
SCR_TITLE_2:       DB          '   Smart Card   '
SCR_TITLE_3:       DB          '                '
SCR_TITLE_4:       DB          ' Smartcard  Full'
SCR_TITLE_5:       DB          ' Please  Change '

SCR_DATE:          DB          ' Set  Date      '
SCR_TIME:          DB          ' Set  Time      '
SCR_SURE:          DB          ' Sure?          '
SCR_YES_NO:        DB          ' *:Yes     #:No '
SCR_OK_CANCLE:     DB          ' *:Ok  #:Cancle '

SCR_SET_DATE:      DB          'Date :  dd/mm/yy'
SCR_SHOW_DATE:     DB          '   /  /         '
SCR_SET_TIME:      DB          'Time :  hh:mm:ss'
SCR_SHOW_TIME:     DB          '   :  :         '
SCR_SHOW_ID:       DB          'ID No.:         '

end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ง.15 หน้าต่างโปรแกรมให้ป้อนชื่อผู้ใช้และรหัสผ่าน

รูปที่ ง.16 หน้าต่างโปรแกรมอ่าน-เขียนสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'------------------------------------------------------------------
' FROM 2
'------------------------------------------------------------------
Option Explicit

Private Sub Command1_Click()
Unload Me
End Sub

Private Sub Form_Load()

End Sub

Private Sub Text2_KeyPress(Index As Integer, KeyAscii As Integer)
    If KeyAscii = 13 Then
        If Text2(0).Text = "123456" And Text2(1).Text = "123456" Then
            Form1.Show
            Unload Me
        Else
            MsgBox "รหัสผ่านผิดพลาด กรุณาตรวจสอบรหัสผ่านใหม่", vbCritical, "ผิดพลาด"
        End If
    End If
End Sub

'------------------------------------------------------------------
' FROM 1
'------------------------------------------------------------------
Option Explicit
Dim Item As ListItem
Dim Db As Database
Dim Rs, TmpRs As Recordset
Dim SQL, TMP As String
Dim CheckFrame, i, Page As Integer
Dim Pass, Late As Integer

Private Sub Combo1_Change()
    Combo1.Text = ""
End Sub

Private Sub Combo10_Change()
    Combo10.Text = ""
End Sub

Private Sub Combo10_Click()
    Select Case Combo10.ListIndex
    Case 0
        TMP = "faculty"
    Case 1
        TMP = "department"
    Case 2
        TMP = "academic"
    Case 3
        TMP = "subject"
    End Select
    Set Rs = Db.OpenRecordset("tbl" & TMP, dbOpenDynaset)
    Combo9.Clear
    While Rs.EOF <> True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Combo9.AddItem Rs.Fields(1)
        Rs.MoveNext
    Wend
    Text9.Text = ""
End Sub

Private Sub Combo11_Change()
    Combo11.Text = ""
End Sub

Private Sub Combo12_Change()
    Combo12.Text = ""
End Sub

Private Sub Combo2_Change()
    Combo2.Text = ""
End Sub

Private Sub Combo3_Change()
    Combo3.Text = ""
End Sub

Private Sub Combo6_Change()
    Combo6.Text = ""
End Sub

Private Sub Combo9_Change()
    Combo9.Text = ""
End Sub

Private Sub Combo9_Click()
    SQL = "select * from tbl" & TMP & " where " & TMP & " = '" &
Combo9.Text & "'"
    Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
    Text9.Text = Rs.Fields(1)
    SQL = "select * from tbl" & TMP & " where " & TMP & " = '" &
Text9.Text & "'"
End Sub

Private Sub Form_Load()
    CheckFrame = 0: Page = 0
    ClearFrame
    ConnectDataBase
    SetCombo
    ClearText
    SetToolBar
End Sub

Public Sub ClearFrame()
    Frame1.Visible = False: Frame2.Visible = False
    Frame3.Visible = False: Frame4.Visible = False
    Frame5.Visible = False: Frame6.Visible = False
    Frame7.Visible = False
End Sub

Public Sub ConnectDataBase()
    Set Db = DBEngine.Workspaces(0).OpenDatabase(App.Path & "\data")
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Unload(Cancel As Integer)
    Dim r As Long
    r = AC_Close(hReader)
End Sub

Private Sub mnuaddgen_Click()
    CheckFrame = 3
    ClearFrame
    Frame2.Visible = True
    Text3.SetFocus
    SetToolBar
End Sub

Private Sub mnuaddstd_Click()
    CheckFrame = 2
    ClearFrame
    Frame1.Visible = True
    Text1.SetFocus
    SetToolBar
End Sub

Private Sub mnucome_Click()
    CheckFrame = 6
    ClearFrame
    Frame5.Visible = True
    Combo11.SetFocus
    SetToolBar
End Sub

Private Sub mnueditgen_Click()
    CheckFrame = 5
    ClearFrame
    Frame4.Visible = True
    Combo10.SetFocus
    SetToolBar
End Sub

Private Sub mnueditstd_Click()
    CheckFrame = 4
    ClearFrame
    Frame3.Visible = True
    Text6.SetFocus
    SetToolBar
End Sub

Private Sub mnuexit_Click()
    If MsgBox("ต้องการจะออกจากโปรแกรม ?", vbOKCancel, "ออกจากโปรแกรม") = vbOK
Then
        Unload Me
    End If
End Sub

Private Sub mnuload_Click()
    CheckFrame = 1
    ClearFrame
    Frame6.Visible = True
    Combo12.SetFocus
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    SetToolBar
End Sub

Private Sub mnureg_Click()
    CheckFrame = 7
    ClearFrame
    Frame7.Visible = True
    Text11.SetFocus
    SetToolBar
End Sub

Private Sub SSTab1_Click(PreviousTab As Integer)
    Select Case SSTab1.Tab
        Case 0
            Text3.SetFocus
        Case 1
            Text4.SetFocus
        Case 2
            Text5.SetFocus
        Case 3
            Text10.SetFocus
    End Select
End Sub

Private Sub Text11_Change()
    On Error GoTo hell
    SQL = "select name from tbldata where id = '" & Text11.Text & "'"
    Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
    Label35.Caption = Rs.Fields(0)
    SetList
    Exit Sub
hell:
End Sub

Private Sub Text6_Change()
    On Error GoTo hell
    SQL = "select * from tbldata where id = '" & Text6.Text & "'"
    Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
    Text8.Text = Rs.Fields(1)
    Combo4.Text = Rs.Fields(2)
    Combo5.Text = Rs.Fields(3)
    Combo7.Text = Rs.Fields(4)
    SQL = "select subject from tblreg where idstd = '" & Text6.Text &
"'"
    Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
    If Rs.RecordCount > 0 Then
        Combo8.Text = Rs.Fields(0)
    End If
    Exit Sub
hell:
    Text8.Text  = "": Combo4.Text = ""
    Combo5.Text = "": Combo7.Text = ""
    Combo4.Text = "": Combo8.Text = ""
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComctlLib.Button)
    On Error GoTo hell
    Select Case Button.Index
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Case 2  'Load Data
        If Combo12.Text <> "" And Combo14.Text <> "" Then
            ConnectCard
            ReadCard
            ClearCard
        Else
            MsgBox "กรุณาเลือกข้อมูลและราชวิชา Port และราชวิชา", vbCritical, "Error"
        End If
    Case 4  ' Save
        SaveData
    Case 5  ' Search
        DeleteData
    Case 6 'Print
        PrintData
    Case 8 'Exit
        mnuexit_Click
    End Select
    Exit Sub
hell:
End Sub

Public Sub SaveData()
    'On Error GoTo hell
    Select Case CheckFrame
        Case 2 'Add Student Data
            Set Rs = Db.OpenRecordset("tbldata", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(0) = Text1.Text
                Rs.Fields(1) = Text2.Text
                Rs.Fields(2) = Combo1.Text
                Rs.Fields(3) = Combo2.Text
                Rs.Fields(4) = Combo3.Text
                Rs.Update
            Set Rs = Db.OpenRecordset("tblreg", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(1) = Text1.Text
                Rs.Fields(2) = Combo6.Text
                Rs.Update
            ClearText
            Text1.SetFocus
        Case 3 'Add General Data
            Select Case SSTab1.Tab
            Case 0 ' Faculty
             Set Rs = Db.OpenRecordset("tblfaculty", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(1) = Text3.Text
                Rs.Update
                Text3.Text = ""
                SetCombo
                Text3.SetFocus
            Case 1 ' Academic
            Set Rs = Db.OpenRecordset("tblacademic", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(1) = Text4.Text
                Rs.Update
                Text4.Text = ""
                SetCombo
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Text4.SetFocus
            Case 2 'Department
            Set Rs = Db.OpenRecordset("tbldepartment", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(1) = Text5.Text
                Rs.Update
                Text5.Text = ""
                SetCombo
                Text5.SetFocus
            Case 3 'Subject
            Set Rs = Db.OpenRecordset("tblsubject", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(0) = Text10.Text
                Rs.Fields(1) = Text7.Text
                Rs.Update
                Set Rs = Db.OpenRecordset("tblteach", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(0) = Text7.Text
                Rs.Update
                Text7.Text = "": Text10.Text = ""
                SetCombo
                Text10.SetFocus
            End Select
        Case 4 'Edit Student Data
            SQL = "select * from tbldata where id = '" & Text6.Text &
"'"
            Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
            Rs.Edit
                Rs.Fields(1) = Text8.Text
                Rs.Fields(2) = Combo4.Text
                Rs.Fields(3) = Combo5.Text
                Rs.Fields(4) = Combo7.Text
            Rs.Update
            SQL = "select subject from tblreg where idstd = '" &
Text6.Text & "'"
            Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
            Rs.Edit
                Rs.Fields(0) = Combo8.Text
            Rs.Update
            ClearText
            Text6.SetFocus
        Case 5 'Edit General Data
            TMP = Rs.Fields(1)
            Rs.Edit
            Rs.Fields(1) = Text9.Text
            Rs.Update
            Select Case Combo10.ListIndex
             Case 0
                SQL = "update tbldata set faculty ='" & Text9.Text &
"' where faculty ='" & Combo9.Text & "'"
                Db.Execute (SQL)

            Case 1
                SQL = "update tbldata set academic ='" & Text9.Text &
"' where academic ='" & Combo9.Text & "'"
                Db.Execute (SQL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Case 2
                SQL = "update tbldata set department ='" & Text9.Text
& "' where department ='" & Combo9.Text & "'"
                Db.Execute (SQL)
            Case 3
                SQL = "update tblstudy set subject ='" & Text9.Text &
"' where subject='" & Combo9.Text & "'"
                Db.Execute (SQL)
                SQL = "update tblteach set subject ='" & Text9.Text &
"' where subject='" & Combo9.Text & "'"
                Db.Execute (SQL)
                SQL = "update tblreg set subject ='" & Text9.Text &
"' where subject='" & Combo9.Text & "'"
                Db.Execute (SQL)
            End Select
            Text9.Text = ""
            Combo10_Click
        Case 7 'Add Subject
            Set Rs = Db.OpenRecordset("tblreg", dbOpenDynaset)
            Rs.AddNew
            Rs.Fields(1) = Text11.Text
            Rs.Fields(2) = Combo13.Text
            Rs.Update
            SetList
    End Select
    MsgBox "บันทึกข้อมูลเรียบร้อย", vbOKOnly, "บันทึก"
    Exit Sub
hell:
End Sub

Public Sub SetCombo()
    Set Rs = Db.OpenRecordset("tblfaculty", dbOpenDynaset)
    Combo1.Clear: Combo4.Clear
    While Rs.EOF <> True
        Combo1.AddItem Rs.Fields(1)
        Combo4.AddItem Rs.Fields(1)
        Rs.MoveNext
    Wend
    Set Rs = Db.OpenRecordset("tblacademic", dbOpenDynaset)
    Combo2.Clear: Combo5.Clear
    While Rs.EOF <> True
        Combo2.AddItem Rs.Fields(1)
        Combo5.AddItem Rs.Fields(1)
        Rs.MoveNext
    Wend
    Set Rs = Db.OpenRecordset("tbldepartment", dbOpenDynaset)
    Combo3.Clear: Combo7.Clear
    While Rs.EOF <> True
        Combo3.AddItem Rs.Fields(1)
        Combo7.AddItem Rs.Fields(1)
        Rs.MoveNext
    Wend
    Set Rs = Db.OpenRecordset("tblsubject", dbOpenDynaset)
    Combo6.Clear: Combo8.Clear
    Combo11.Clear: Combo13.Clear
    Combo14.Clear
    While Rs.EOF <> True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Combo6.AddItem Rs.Fields(1)
        Combo8.AddItem Rs.Fields(1)
        Combo11.AddItem Rs.Fields(1)
        Combo13.AddItem Rs.Fields(1)
        Combo14.AddItem Rs.Fields(1)
        Rs.MoveNext
    Wend
End Sub

Public Sub ClearText()
    Text1.Text = "": Text2.Text = ""
    Combo1.Text = "": Combo2.Text = ""
    Combo3.Text = "": Combo6.Text = ""
    Text6.Text = "": Text8.Text = ""
    Combo4.Text = "": Combo5.Text = ""
    Combo7.Text = "": Combo8.Text = ""
    Combo10.Text = "": Combo9.Text = ""
    Combo13.Text = "": Text11.Text = ""
    List1.Clear
End Sub

Public Sub DeleteData()
    On Error GoTo hell
    Select Case CheckFrame
        Case 4 'Delete Student Data
            If Len(Text6.Text) = 8 Then
                If MsgBox("ต้องการที่จะลบข้อมูล ?", vbOKCancel, "ยืนยันการลบข้อมูล")
= vbOK Then
                    SQL = "delete from tbldata where id ='" &
Text6.Text & "'"
                    Db.Execute (SQL)
                    SQL = "delete from tblreg where idstd ='" &
Text6.Text & "'"
                    Db.Execute (SQL)
                    SQL = "delete from tblstudy where idstudent ='" &
Text6.Text & "'"
                    Db.Execute (SQL)
                    ClearText
                End If
            End If
        Case 5 'Delete General Data
            If Text9.Text <> "" Then
                If MsgBox("ต้องการที่จะลบข้อมูล ?", vbOKCancel, "ยืนยันการลบข้อมูล")
= vbOK Then
                    Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
                    Rs.Delete
                    If Combo10.ListIndex = 3 Then
                        SQL = "delete from tblstudy where subject='"
& Combo9.Text & "'"
                        Db.Execute (SQL)
                        SQL = "delete from tblteach where subject='"
& Combo9.Text & "'"
                        Db.Execute (SQL)
                        SQL = "delete from tblreg where subject='" &
Combo9.Text & "'"
                        Db.Execute (SQL)
                    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    Combo10_Click
                    SetCombo
                End If
            End If
        Case 7 'Delete Subject
            If Len(Text11.Text) = 8 Then
                If MsgBox("ต้องการที่จะลบข้อมูล ?", vbOKCancel, "ยืนยันการลบข้อมูล")
= vbOK Then
                    SQL = "delete from tblreg where idstd = '" &
Text11.Text & "' and subject = '" & List1.Text & "'"
                    Db.Execute (SQL)
                    SetList
                End If
            End If
    End Select
    MsgBox "ลบข้อมูลเรียบร้อย", vbOKOnly, "ลบข้อมูล"
    Exit Sub
hell:
    MsgBox "กรุณาตรวจสอบความถูกต้องของข้อมูลที่ต้องการลบ", vbCritical, "ผิดพลาด"
End Sub

Public Sub SetToolBar()
    On Error GoTo hell
    Toolbar1.Buttons.Item(2).Enabled = False
    Toolbar1.Buttons.Item(4).Enabled = False
    Toolbar1.Buttons.Item(5).Enabled = False
    Toolbar1.Buttons.Item(6).Enabled = False
    Select Case CheckFrame
        Case 1 'Connect
            Toolbar1.Buttons.Item(2).Enabled = True
        Case 2 'Add Data Student
            Toolbar1.Buttons.Item(4).Enabled = True
        Case 3 'Add General
            Toolbar1.Buttons.Item(4).Enabled = True
        Case 4 'Edit Data Student
            Toolbar1.Buttons.Item(4).Enabled = True
            Toolbar1.Buttons.Item(5).Enabled = True
        Case 5 'Edit General
            Toolbar1.Buttons.Item(4).Enabled = True
            Toolbar1.Buttons.Item(5).Enabled = True
        Case 6 'Search Data
            Toolbar1.Buttons.Item(6).Enabled = True
        Case 7  'Add Subject
            Toolbar1.Buttons.Item(4).Enabled = True
            Toolbar1.Buttons.Item(5).Enabled = True
    End Select
    Exit Sub
hell:
End Sub

Private Sub Combo11_Click()
    On Error GoTo hell
    ListView1.ListItems.Clear
    SQL = "select teach from tblteach where subject ='" &
Combo11.Text & "'"
    Set TmpRs = Db.OpenRecordset(SQL, dbOpenDynaset)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    i = TmpRs.Fields(0)
    SQL = "select idstd from tblreg where subject = '" & Combo11.Text
& "' order by idstd asc"
    Set TmpRs = Db.OpenRecordset(SQL, dbOpenDynaset)
    While TmpRs.EOF <> True
        SQL = "select count(status) from tblstudy where idstudent =
'" & TmpRs.Fields(0) & "' and subject = '" & Combo11.Text & "' and
status = 0"
        Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
        Pass = Rs.Fields(0)
        SQL = "select count(status) from tblstudy where idstudent =
'" & TmpRs.Fields(0) & "' and subject = '" & Combo11.Text & "' and
status = 1"
        Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
        Late = Rs.Fields(0)
        Set Item = ListView1.ListItems.Add(, , TmpRs.Fields(0))
        Item.SubItems(1) = Pass
        Item.SubItems(2) = Late
        Item.SubItems(3) = i - (Pass + Late)
        TmpRs.MoveNext
    Wend
    Exit Sub
hell:
End Sub

Public Sub PrintData()
    Dim X, Y As Integer
    SQL = "select teach from tblteach where subject ='" &
Combo11.Text & "'"
    Set TmpRs = Db.OpenRecordset(SQL, dbOpenDynaset)
    i = TmpRs.Fields(0)
    If Combo11.Text <> "" Then
        X = 900: Y = 2300
        Printer.Orientation = 1
        Printer.PaperSize = 9
        Printer.PaintPicture Picture1.Picture, 500, 500, 1500, 1500
        Printer.CurrentX = 2300: Printer.CurrentY = 700:
Printer.FontSize = 18
        Printer.Print "สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง"
        Printer.CurrentX = 2300: Printer.CurrentY = 1200:
Printer.FontSize = 16
        Printer.Print "คณะครุศาสตร์อุตสาหกรรม"
        SQL = "select * from tblsubject where subject = '" &
Combo11.Text & "'"
        Set TmpRs = Db.OpenRecordset(SQL, dbOpenDynaset)
        Printer.CurrentX = 2300: Printer.CurrentY = 1700:
Printer.FontSize = 14
        Printer.Print "รหัสวิชา"
        Printer.CurrentX = 3300: Printer.CurrentY = 1700:
Printer.FontSize = 14
        Printer.Print TmpRs.Fields(0)
        Printer.CurrentX = 5300: Printer.CurrentY = 1700:
Printer.FontSize = 14
        Printer.Print "ราชวิชา"
        Printer.CurrentX = 6300: Printer.CurrentY = 1700:
Printer.FontSize = 14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Printer.Print TmpRs.Fields(1)
        'Header of Page
        Printer.CurrentX = X: Printer.CurrentY = Y: Printer.FontSize
= 12
        Printer.Print "รหัสนักศึกษา"
        Printer.CurrentX = X + 3000: Printer.CurrentY = Y
        Printer.Print "ชื่อ - นามสกุล"
        Printer.CurrentX = X + 7000: Printer.CurrentY = Y
        Printer.Print "เข้าเรียน"
        Printer.CurrentX = X + 8200: Printer.CurrentY = Y
        Printer.Print "เข้าสาย"
        Printer.CurrentX = X + 9300: Printer.CurrentY = Y
        Printer.Print "ขาดเรียน"
        Y = Y + 400
        'End ofHeader
        SQL = "select idstd from tblreg where subject = '" &
Combo11.Text & "' order by idstd asc"
        Set TmpRs = Db.OpenRecordset(SQL, dbOpenDynaset)
        While TmpRs.EOF <> True
            SQL = "select name from tbldata where id ='" &
TmpRs.Fields(0) & "'"
            Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
            Printer.CurrentX = X + 100: Printer.CurrentY = Y:
Printer.FontSize = 9
            Printer.Print TmpRs.Fields(0)
            Printer.CurrentX = X + 2800: Printer.CurrentY = Y
            Printer.Print Rs.Fields(0)
            SQL = "select count(status) from tblstudy where idstudent
= '" & TmpRs.Fields(0) & "' and subject = '" & Combo11.Text & "' and
status = 0"
            Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
            Pass = Rs.Fields(0)
            SQL = "select count(status) from tblstudy where idstudent
= '" & TmpRs.Fields(0) & "' and subject = '" & Combo11.Text & "' and
status = 1"
            Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
            Late = Rs.Fields(0)
            Printer.CurrentX = X + 7250: Printer.CurrentY = Y
            Printer.Print Pass
            Printer.CurrentX = X + 8450: Printer.CurrentY = Y
            Printer.Print Late
            Printer.CurrentX = X + 9550: Printer.CurrentY = Y
            Printer.Print i - (Pass + Late)
            Y = Y + 200
            TmpRs.MoveNext
        Wend
        'End of Page
        Printer.CurrentX = 7000: Printer.CurrentY = 15700:
Printer.FontSize = 10
        Printer.Print "ลงชื่อ.....(อาจารย์ผู้สอน)"
        Printer.CurrentX = 7400: Printer.CurrentY = 16200:
Printer.FontSize = 10
        Printer.Print "(              )"
        Printer.EndDoc
    Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MsgBox "กรุณาเลือกรายวิชา", vbCritical, "ผิดพลาด"
    End If
End Sub

Public Sub ConnectCard()
    hReader = AC_Open(ACR20_115200, AC_COM1)
    If (hReader < 0) Then
        nResponse = MsgBox("Cannot Open Port", vbCritical, "Error")
    End
    End If
    Session.CardType = AC_AM8KS
End Sub

Public Sub SetList()
    List1.Clear
    SQL = "select * from tblreg where idstd = '" & Text11.Text & "'"
    Set Rs = Db.OpenRecordset(SQL, dbOpenDynaset)
    While Rs.EOF <> True
        List1.AddItem Rs.Fields(2)
        Rs.MoveNext
    Wend
End Sub

Private Sub ReadCard()
    Dim GioX, Day(3), ID As String
    Dim RTime As Long
    Dim Status As Long
    SQL = "select teach from tblteach where subject ='" &
Combo14.Text & "'"
    Set TmpRs = Db.OpenRecordset(SQL, dbOpenDynaset)
    i = TmpRs.Fields(0)
    TmpRs.Edit
    TmpRs.Fields(0) = i + 1
    TmpRs.Update
Loop1:
    RtnCode = AC_StartSession(hReader, Session)
    If (RtnCode < 0) Then
        nResponse = MsgBox("Cannot Start Session", vbCritical,
"Error")
    Exit Sub
    End If
    '    Read Card
        APDU.CLA = &H0                          '0
        APDU.INS = &H1                          '1
        APDU.P1 = Page                          ' page select
        APDU.P2 = &H20                          '32
        APDU.Lc = &H0                           '0
        APDU.Le = 225                           '225
        RtnCode = AC_ExchangeAPDU(hReader, APDU)
        For i = 0 To 225
        '================================================
        Select Case Chr(APDU.DataOut(i))
        Case "|"
            Select Case Len(GioX)
            Case 6
                Status = CalculateTime(RTime, ((Left(GioX, 2) * 3600)
+ (Mid(GioX, 3, 2) * 60) + (Right(GioX, 2))))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Set Rs = Db.OpenRecordset("tblstudy", dbOpenDynaset)
                Rs.AddNew
                Rs.Fields(1) = ID
                Rs.Fields(2) = Combo14.Text
                Rs.Fields(3) = Day(0)
                Rs.Fields(4) = Day(1)
                Rs.Fields(5) = Day(2)
                Rs.Fields(6) = Status
                Rs.Update
            Case 8
                ID = GioX
            Case 12
                Day(0) = Left(GioX, 2)
                Day(1) = Mid(GioX, 3, 2)
                Day(2) = Mid(GioX, 5, 2)
                RTime = (Mid(GioX, 7, 2) * 3600) + (Mid(GioX, 9, 2) *
60) + (Mid(GioX, 11, 2))
            End Select
            GioX = ""
        Case "ÿ"
            If Page < 3 Then
                Page = Page + 1
                RtnCode = AC_EndSession(hReader)
                GoTo Loop1
            Else
                RtnCode = AC_EndSession(hReader)
                Exit Sub
            End If
        Case Else
            GioX = GioX & Chr(APDU.DataOut(i))
        End Select
        ProgressBar1.value = ProgressBar1.value + 1
    '=================================================
    Next i
End Sub

Public Function CalculateTime(TR As Long, CT As Long) As Long
    Dim Ans As Long
    Ans = CT - TR
    If Ans <= 900 Then
        CalculateTime = 0    'มาทัน
    Else
        CalculateTime = 1    'มาสาย
    End If
End Function

Public Sub ClearCard()
    Dim X As Integer
    For X = 0 To 3
    RtnCode = AC_StartSession(hReader, Session)
    If (RtnCode < 0) Then
        nResponse = MsgBox("Cannot Start Session", vbCritical,
"Error")
        Exit Sub
    End If
'        Verify Card
    APDU.DataIn(0) = &HFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    APDU.DataIn(1) = &HFF
    APDU.CLA = &H0
    APDU.INS = &H4
    APDU.P1 = &H0
    APDU.P2 = &H0
    APDU.Lc = &H2
    APDU.Le = &H3
    RtnCode = AC_ExchangeAPDU(hReader, APDU)
        For i = 1 To 224
            APDU.DataIn(i - 1) = Asc("ÿ")
            ProgressBar1.value = ProgressBar1.value + 1
        Next
        APDU.CLA = &H0
        APDU.INS = &H2
        APDU.P1 = X
        APDU.P2 = &H20
        APDU.Lc = 224
        APDU.Le = &H0
        RtnCode = AC_ExchangeAPDU(hReader, APDU)
'End If
    RtnCode = AC_EndSession(hReader)
    Next X
    ProgressBar1.value = 0
End Sub

'-------------------------------------------------------------------
' MODULE 1
'-------------------------------------------------------------------
Option Explicit
'-------------------------------------------------------------------
'Test Sim Array
'-------------------------------------------------------------------
Public SimArray(1024) As String
'-------------------------------------------------------------------
'Definitions of ACSR20 hardware port connection
'-------------------------------------------------------------------
Global Const AC_COM1 As Integer = 0
Global Const AC_COM2 As Integer = 1
Global Const AC_COM3 As Integer = 2
Global Const AC_COM4 As Integer = 3
Global Const AC_USB As Integer = 15
Global Const AC_KEYB As Integer = 16
Global Const AC_PCSC As Integer = 20
'-------------------------------------------------------------------
' Definition of reader type (use in AC_Open)
'-------------------------------------------------------------------
'It will detect the reader automatically
Global Const ACR_AUTODETECT As Long = &HFF
'The attached reader is an ACR10 reader
Global Const ACR10 As Long = &H0
'ACR20 reader (default = 57600 baud)
Global Const ACR20 As Long = &H1
'ACR20 reader (57600 baud)
Global Const ACR20_57600 As Long = &H1
'ACR20 reader (19200 baud)
Global Const ACR20_19200 As Long = &H2
'ACR20 reader (115200 baud)
Global Const ACR20_115200 As Long = &H3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
'ACR20 reader (9600 baud)
Global Const ACR20_9600 As Long = &H4

'-----------------------------------------------------------------
'Card section:
'-----------------------------------------------------------------
Global Const AC_AM104 As Byte = 1
Global Const AC_AM221 As Byte = 2
Global Const AC_SLE4404 As Byte = 3
Global Const AC_GPM896 As Byte = 4
Global Const AC_AT101 As Byte = 5
Global Const AC_AT102 As Byte = 6
Global Const AC_AM8KP As Byte = 7
Global Const AC_AM8KS As Byte = 8
Global Const AC_AM2KP As Byte = 9
Global Const AC_AM2KS As Byte = 10
Global Const AC_IIC As Byte = 11
Global Const AC_XIIC As Byte = 12
Global Const AC_AT1604 As Byte = 13
Global Const AC_T0 As Byte = 14
Global Const AC_T1 As Byte = 15
Global Const AC_SCModule As Byte = 16
Global Const AC_AM256 As Byte = 17
Global Const AC_AM4KP As Byte = 18
Global Const AC_X76F041 As Byte = 19
Global Const AC_X24645 As Byte = 20

'-----------------------------------------------------------------
'ACI Command Set
'-----------------------------------------------------------------
Global Const ACI_Read As Integer = 1
Global Const ACI_Write As Integer = 2
Global Const ACI_SetFuse As Integer = 3
Global Const ACI_CardOptions As Integer = 3
Global Const ACI_Verify As Integer = 4
Global Const ACI_WritePr As Integer = 5
Global Const ACI_ChangePIN As Integer = 6
Global Const ACI_Erase As Integer = 7
Global Const ACI_WriteCarry As Integer = 8
Global Const ACI_Authenticate As Integer = 9
Global Const ACI_SetProtect As Integer = 10
Global Const ACI_ReadProtect As Integer = 11
Global Const ACI_SetHE As Integer = 12
Global Const ACI_BlowFuse As Integer = 10
Global Const ACI_LockProtect As Integer = 13
Global Const ACI_ClearProtect As Integer = 14
Global Const ACI_WriteAll As Integer = 15
Global Const ACI_EraseAll As Integer = 16

'-----------------------------------------------------------------
'ACI Error Constant
'-----------------------------------------------------------------
Global Const ERR_HANDLE_INVALID As Long = -603
Global Const ERR_SESSION_NULL As Long = -600
Global Const ERR_HANDLE_NOFREE As Long = -108
Global Const ERR_PORT_INVALID As Long = -100
Global Const ERR_READER_INVALID As Long = -101
Global Const ERR_PORT_OCCUPLIED As Long = -102
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Global Const ERR_NO_ As Long = -1001
Global Const ERR_NO_CARD As Long = -1002
Global Const ERR_WRONG_ As Long = -1003
Global Const ERR_NO_POWERUP As Long = -1004
Global Const ERR_INVALID_INS As Long = -1005
Global Const ERR_CARD_FAILURE As Long = -1006
Global Const ERR_CARD_PROTOCOL As Long = -1007
Global Const ERR_UNSUPPORT_ As Long = -1008
Global Const ERR_INCOMPATIBLE_COMMAND As Long = -1009
Global Const ERR_ADDRESS As Long = -1010
Global Const ERR_DATA_LEN As Long = -1011
Global Const ERR_RESPONSE_LEN As Long = -1012
Global Const ERR_SECRET_CODE_LOCK As Long = -1013
Global Const ERR_INVALID_MOD_NUMBER As Long = -1014
Global Const ERR_INCORRECT_PASSWORD As Long = -1015
Global Const ERR_INVALID_CLA As Long = -1050
Global Const ERR_APDU_PARAM As Long = -1051
Global Const ERR_COM_BUFFER_FULL As Long = -1052
Global Const ERR_WORD_BOUNDARY As Long = -1053
Global Const ERR_PROTOCOL_FRAME As Long = -1080
Global Const ERR_NO_RESPONSE As Long = -1081
Global Const ERR_INCORRECT_PARAM As Long = -1082
Global Const ERR_UNKNOWN As Long = -1099
Global hReader, nResponse, RtnCode, dummy As Long
Declare Function GetReaderErrorCode Lib "ACSR2032.DLL" () As Integer

'-------------------------------------------------------------------
'ACRInterface
'-------------------------------------------------------------------
Type AC_SESSION
        CardType                    As Byte
        SCModule                    As Byte
        ATRLen                      As Byte
        ATR                         As String * 256
        HistLen                     As Byte
        HistOffset                  As Byte
        ApduLenMax                  As Long
End Type

Type AC_APDU
        CLA                         As Byte
        INS                         As Byte
        P1                          As Byte
        P2                          As Byte
        Lc                          As Long
        Le                          As Long
        DataIn(0 To 255)            As Byte 'String * 256
        DataOut(0 To 255)           As Byte 'String * 256
        Status                      As Long
End Type

Type AC_INFO
        nMaxC                       As Long
        nMaxR                       As Long
        CType                       As Long
        CStat                       As Byte
        CSel                        As Byte
        szRev                       As String * 256
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        nLibVer                         As Long
        lBaudRate                       As Long
End Type
'-------------------------------------------------------------------
'Prototype section
'-------------------------------------------------------------------
Public Declare Function AC_Open Lib "ACSR2032.DLL" (ByVal ReaderType
As Long, ByVal ReaderPort As Integer) As Long
Public Declare Function AC_Close Lib "ACSR2032.DLL" (ByVal hReader As
Long) As Long
Public Declare Function AC_StartSession Lib "ACSR2032.DLL" (ByVal
hReader As Long, Session As AC_SESSION) As Long
Public Declare Function AC_EndSession Lib "ACSR2032.DLL" (ByVal
hReader As Long) As Long
Public Declare Function AC_GetInfo Lib "ACSR2032.DLL" (ByVal hReader
As Long, Info As AC_INFO) As Long
Public Declare Function AC_SetOptions Lib "ACSR2032.DLL" (ByVal
hReader As Long, ByVal OType As Integer, ByVal value As Integer) As
Long
Public Declare Function AC_ExchangeAPDU Lib "ACSR2032.DLL" (ByVal
hReader As Long, APDU As AC_APDU) As Long

Global Session  As AC_SESSION
Global APDU As AC_APDU
Global Info As AC_INFO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.1 ฐานข้อมูลคณะ

id	academic
1	ครุศาสตร์อุตสาหกรรม

ตารางที่ ง.2 ฐานข้อมูลภาควิชา

id	academic
1	ครุศาสตร์วิศวกรรม

ตารางที่ ง.3 ฐานข้อมูลสาขาวิชา

id	department
1	อิเล็กทรอนิกส์และคอมพิวเตอร์
2	วิศวกรรมโทรคมนาคม
3	เทคโนโลยีวัดคุมทางอุตสาหกรรม

ตารางที่ ง.4 ฐานข้อมูลวิชาเรียน

id	subject
03321200	Electronics Engineering
03321206	Computer Engineering
03321202	Electromagnetics Engineering

ตารางที่ ง.5 ฐานข้อมูลจำนวนครั้งที่อาจารย์สอน

subject	teach
Electronics Engineering	0
Computer Engineering	0
Electromagnetics Engineering	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.6 ฐานข้อมูลนักศึกษา

id	name	faculty	academic	department
45035373	กิติชน สุขรุจิ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035374	เกริกชัย ชะวะนะเวช	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035375	โกสินทร์ บัวขาว	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035376	โกสินทร์ อินทราพงษ์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035377	เครดิต จำปา	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035378	จิรวัฒน์ สาระพงศ์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035379	จิรายุ สุรำไพ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035380	ชัชชัย ไชยโย	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035381	ชัยณรงค์ จายเขียว	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035383	ชำนัญ สุขศรีนาค	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035385	ฌานุตย์ อินทสุวรรณ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035386	ฐากูร สุทธิศิริกุล	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035387	ดนุพงษ์ รัตนะ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035388	ทิพย์สุดา สุดประเสริฐ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035389	บันฑิต ศิลปะระเสริฐ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035390	ประกายวรรณ ธรรมสังวาล	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035391	ปัญญา ภูมิเจริญ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035392	พรชัย ณรงค์ราช	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035393	ภานุวัตร อนันต์แดง	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035394	มงคล หมวดเพชร์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035395	มานะชัย ทีกว้าง	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035396	วชิระ เจริญรื่น	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035397	วัชระ กลิ่นสุคนธ์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035398	วิกรม เพชรสดใส	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035399	วุฒินันท์ หะรารักษ์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035400	ศรายุทธ สมบัติ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------
45035401	ศุภชัย สุวรรณสุข	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์---------

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.6 (ต่อ) ฐานข้อมูลนักศึกษา

45035402	เศกสิทธิ์ โพธิ์ศรี	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035403	สดายุ ธรรมสะเร	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035404	สมโภช อรุรศิริ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035405	สลิลพรรณ ดำฤทธิ์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035406	สิทธิชัย ขุนทองปาน	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035407	สุปราณี ดอนเตาเหล็ก	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035408	สรุเดช สว่างวัน	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035409	สุรศักดิ์ สุภาสธากุล	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035410	อขนิษฐ เครืออนันต์	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035411	อนันต์ชัย วัตตา	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035413	อภินันท์ รอดจินดา	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------
45035414	อรรถพร อักษรนำ	ครุศาสตร์----	ครุศาสตร์----	อิเล็กทรอนิกส์--------

ตารางที่ ง.7 ฐานข้อมูลการลงทะเบียนของนักศึกษา

id	idstudent	subject
1	45035373	Electronics Engineering
2	45035374	Electronics Engineering
3	45035375	Electronics Engineering
4	45035376	Electronics Engineering
5	45035377	Electronics Engineering
6	45035378	Electronics Engineering
7	45035379	Electronics Engineering
8	45035380	Electronics Engineering
9	45035381	Electronics Engineering
10	45035383	Electronics Engineering
11	45035385	Electronics Engineering
12	45035386	Electronics Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.7 (ต่อ) ฐานข้อมูลการลงทะเบียนของนักศึกษา

13	45035387	Electronics Engineering
14	45035388	Electronics Engineering
15	45035389	Electronics Engineering
16	45035390	Electronics Engineering
17	45035391	Electronics Engineering
18	45035392	Electronics Engineering
19	45035393	Electronics Engineering
20	45035394	Electronics Engineering
21	45035395	Electronics Engineering
22	45035396	Electronics Engineering
23	45035397	Electronics Engineering
24	45035398	Electronics Engineering
25	45035399	Electronics Engineering
26	45035400	Electronics Engineering
27	45035401	Electronics Engineering
28	45035402	Electronics Engineering
29	45035403	Electronics Engineering
30	45035404	Electronics Engineering
31	45035405	Electronics Engineering
32	45035406	Electronics Engineering
33	45035407	Electronics Engineering
34	45035408	Electronics Engineering
35	45035409	Electronics Engineering
36	45035410	Electronics Engineering
37	45035411	Electronics Engineering
38	45035413	Electronics Engineering
39	45035414	Electronics Engineering

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.8 ฐานข้อมูลการเข้าเรียนของนักศึกษา

id	idstudent	subject	day	month	year	status
1	45035373	Electronics Engineering	19	02	04	1
2	45035374	Electronics Engineering	19	02	04	1
3	45035375	Electronics Engineering	19	02	04	1
4	45035376	Electronics Engineering	19	02	04	1
5	45035377	Electronics Engineering	19	02	04	1
6	45035378	Electronics Engineering	19	02	04	1
7	45035379	Electronics Engineering	19	02	04	1
8	45035380	Electronics Engineering	19	02	04	1
9	45035381	Electronics Engineering	19	02	04	1
10	45035383	Electronics Engineering	19	02	04	1
11	45035385	Electronics Engineering	19	02	04	1
12	45035386	Electronics Engineering	19	02	04	1
13	45035387	Electronics Engineering	19	02	04	1
14	45035388	Electronics Engineering	19	02	04	1
15	45035389	Electronics Engineering	19	02	04	1
16	45035390	Electronics Engineering	19	02	04	1
17	45035391	Electronics Engineering	19	02	04	1
18	45035392	Electronics Engineering	19	02	04	1
19	45035393	Electronics Engineering	19	02	04	1
20	45035394	Electronics Engineering	19	02	04	1
21	45035395	Electronics Engineering	19	02	04	1
22	45035396	Electronics Engineering	19	02	04	1
23	45035397	Electronics Engineering	19	02	04	1
24	45035398	Electronics Engineering	19	02	04	1
25	45035399	Electronics Engineering	19	02	04	1
26	45035400	Electronics Engineering	19	02	04	1
27	45035401	Electronics Engineering	19	02	04	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ ง.8 (ต่อ) ฐานข้อมูลการเข้าเรียนของนักศึกษา

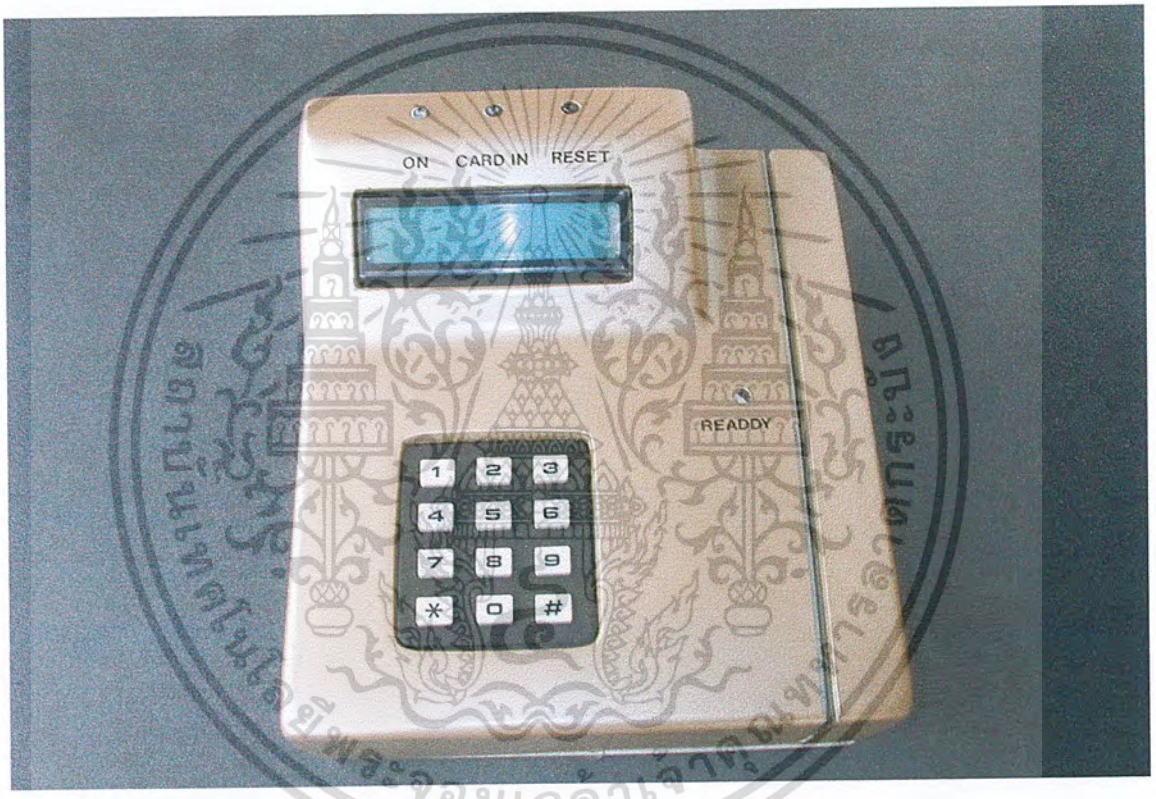
28	45035402	Electronics Engineering	19	02	04	1
29	45035403	Electronics Engineering	19	02	04	1
30	45035404	Electronics Engineering	19	02	04	1
31	45035405	Electronics Engineering	19	02	04	1
32	45035406	Electronics Engineering	19	02	04	1
33	45035407	Electronics Engineering	19	02	04	1
34	45035408	Electronics Engineering	19	02	04	1
35	45035409	Electronics Engineering	19	02	04	1
36	45035410	Electronics Engineering	19	02	04	1
37	45035411	Electronics Engineering	19	02	04	1
38	45035413	Electronics Engineering	19	02	04	1
39	45035414	Electronics Engineering	19	02	04	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ
คู่มือการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน
เครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด



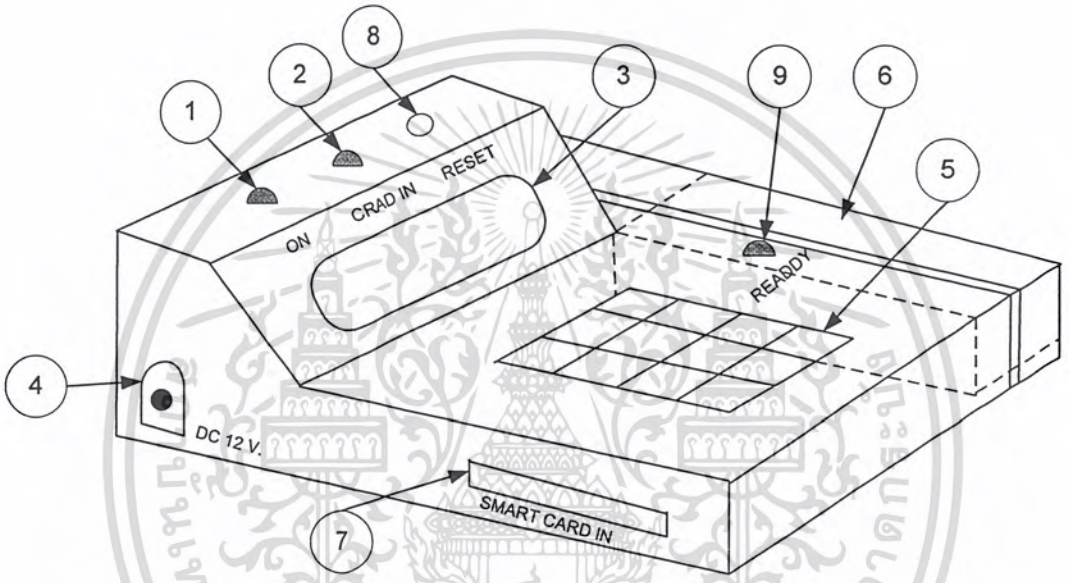
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คำแนะนำเบื้องต้น

ก่อนลงมือใช้งานเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดควรศึกษาการใช้งานจากคู่มือให้เข้าใจก่อนเพื่อการใช้งานที่ถูกต้อง และเป็นการป้องกันการเสียหายที่อาจเกิดขึ้นกับเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดได้

2. ส่วนประกอบและปุ่มควบคุม



รูปที่ จ.1 ส่วนประกอบของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

จากรูปที่ จ.1 มีรายละเอียดต่างๆ ดังนี้

1. หลอดแอลอีดีแสดงไฟเลี้ยงวงจร
2. หลอดแอลอีดีแสดงว่ามีการเสียบบัตรสมาร์ตการ์ด
3. จอแสดงผลแอลซีดีขนาด 16 ตัวอักษร 2 บรรทัด
4. จุดต่อกับตัวอะแด็ปเตอร์ไฟเลี้ยงภายนอก
5. คีย์สวิทช์เมตริกซ์ขนาด 12 ปุ่ม
6. เครื่องอ่านบาร์โค้ด
7. ช่องเสียบบัตรสมาร์ตการ์ด
8. สวิตซ์รีเซต
9. หลอดแอลอีดีแสดงการทำงานของเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. การติดตั้งและการใช้งาน

การใช้งานของเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด แบ่งการใช้งานเป็น 3 ส่วน คือ ส่วนที่หนึ่งคือ การใช้ตรวจสอบและบันทึกข้อมูลโดยสมาร์ตการ์ด ส่วนที่สองคือโปรแกรมที่ใช้ติดต่อระหว่างโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ดกับฐานข้อมูลในเครื่องคอมพิวเตอร์ ซึ่งเขียนขึ้นจากโปรแกรมวิชวลเบสิก เพื่อโหลดข้อมูลจากสมาร์ตการ์ดที่ผ่านการเขียนข้อมูลจากเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด และส่วนที่สาม คือโมดูลเครื่องอ่านเขียนสมาร์ตการ์ด ดังนั้นเพื่อให้การทำงานของโปรแกรมวิชวลเบสิกและการใช้งานเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดได้ถูกต้อง จึงต้องศึกษาการใช้งานซึ่งมีขั้นตอนดังต่อไปนี้

3.1 การใช้งานเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด

3.1.1 ต่ออะแด็ปเตอร์เข้ากับเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ดถ้าไฟเข้าแอลอีดีสีแดงจะติด

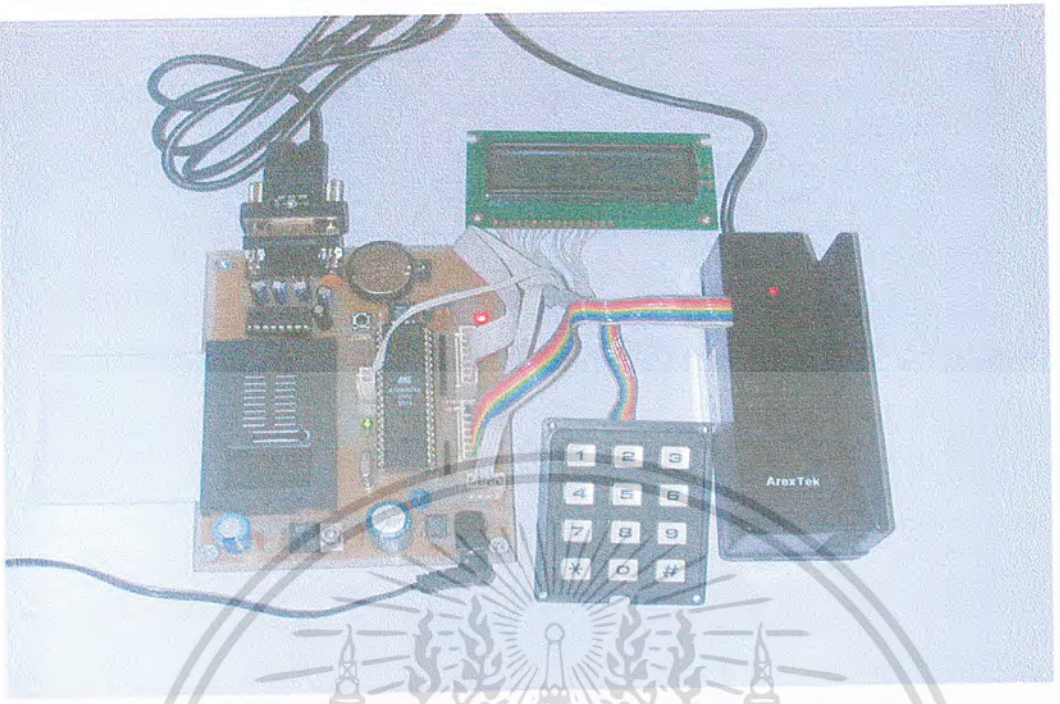
3.1.2 เสียบสมาร์ตการ์ดเข้าที่ช่องเสียบสมาร์ตการ์ดของเครื่อง ถ้ามีการเชื่อมต่อแอลอีดีสีเขียวจะติด ดังรูปที่ จ.2

3.1.3 หน้าจอแอลซีดีจะถามว่า ต้องการจะตั้งวันเดือนปีหรือไม่ ถ้าต้องการให้กดปุ่ม “*” หรือถ้าไม่ต้องการให้กด “#” ดังรูปที่ จ.3

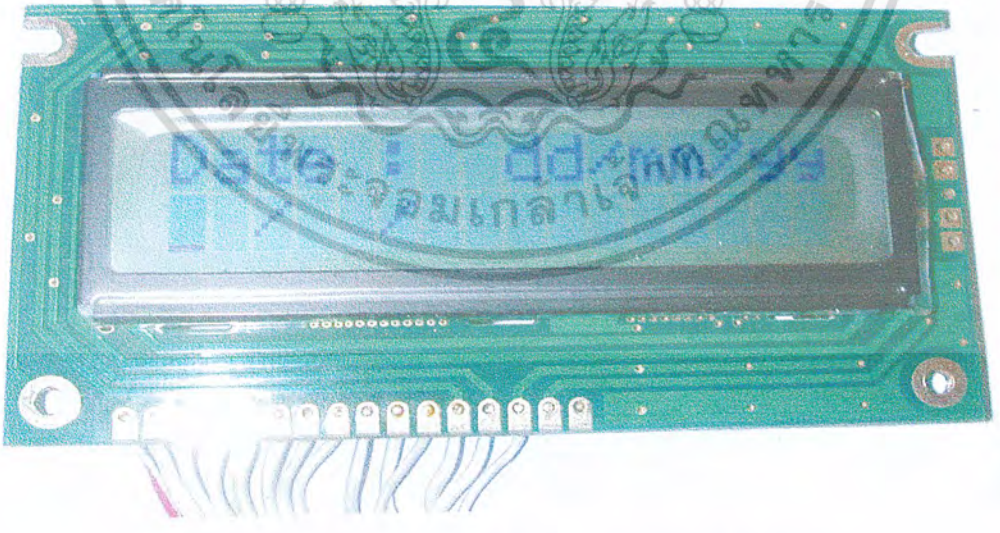
3.1.4 หน้าจอแอลซีดีจะถามว่า ต้องการจะตั้งเวลาหรือไม่ ถ้าต้องการให้กด “*” หรือถ้าไม่ต้องการให้กด “#” ดังรูปที่ จ.4

3.1.5 เสร็จจากกระบวนการตั้งวันเดือนปีและเวลาแล้วหน้าจอแอลซีดีบรรทัดบนก็จะแสดงคำว่า ID_No. : ซึ่งเป็นการรอรับรหัสประจำตัวนักศึกษาจากเครื่องอ่านบาร์โค้ด ส่วนบรรทัดล่างก็จะแสดงวันเดือนปีและเวลา ดังรูปที่ จ.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

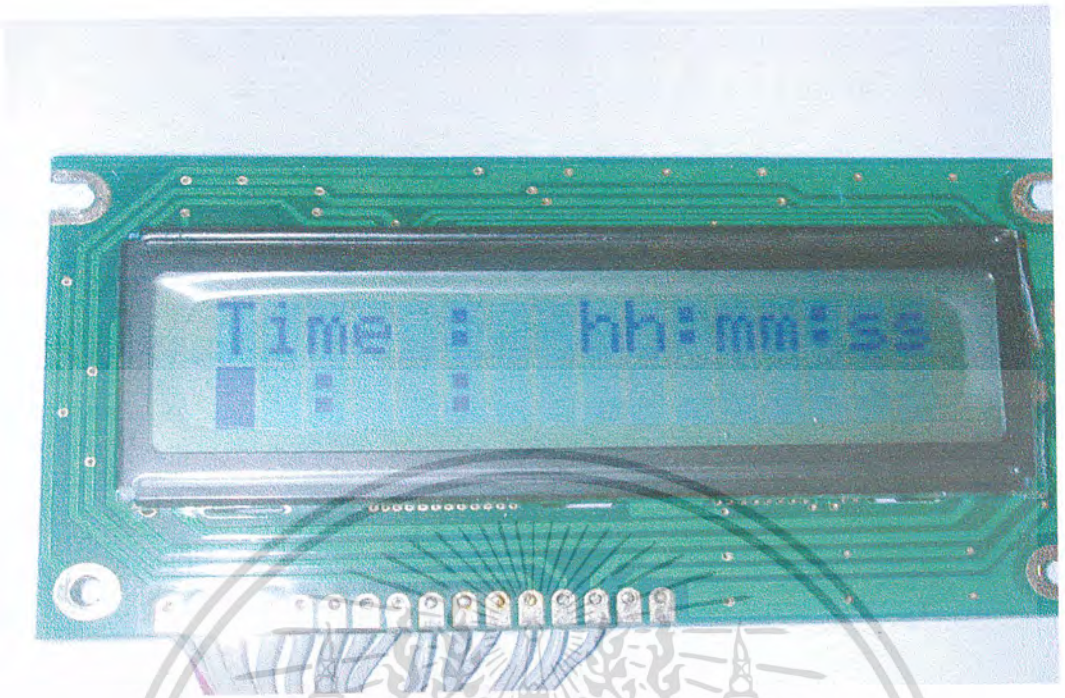


รูปที่ จ.2 แอลอีดีสีเขียวติดขณะเสียบบัตรสมาร์ตการ์ด

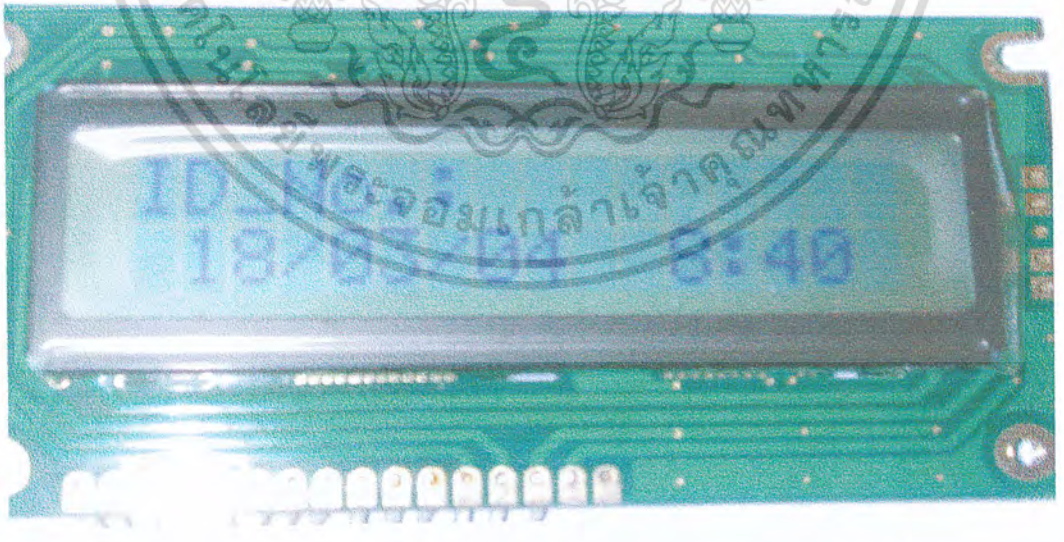


รูปที่ จ.3 หน้าจอแอลซีดีเมื่อต้องการตั้งค่าวันเดือนปี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.4 หน้าจอแอลซีดีเมื่อต้องการตั้งค่าเวลา

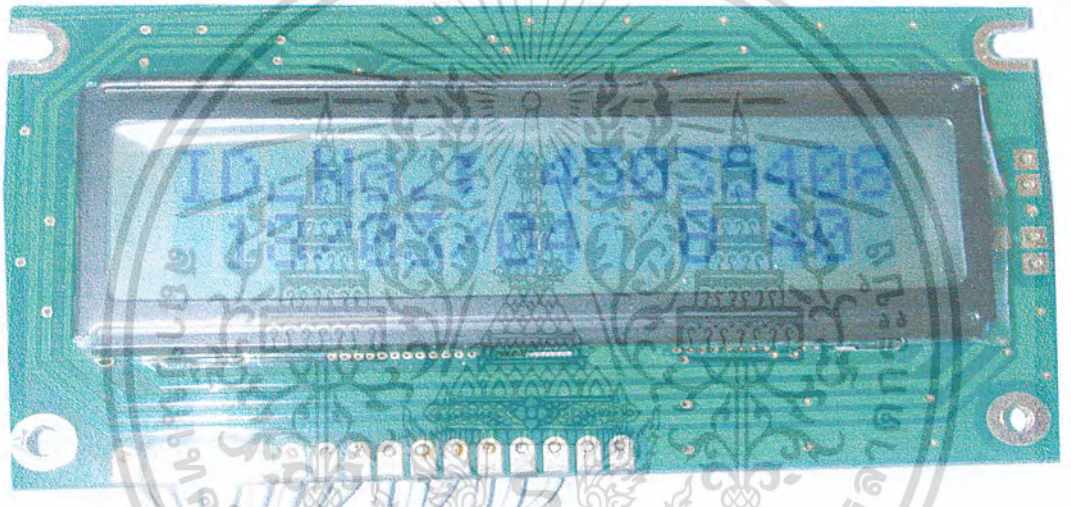


รูปที่ จ.5 หน้าจอแอลซีดีเมื่อเครื่องพร้อมรับข้อมูลจากเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

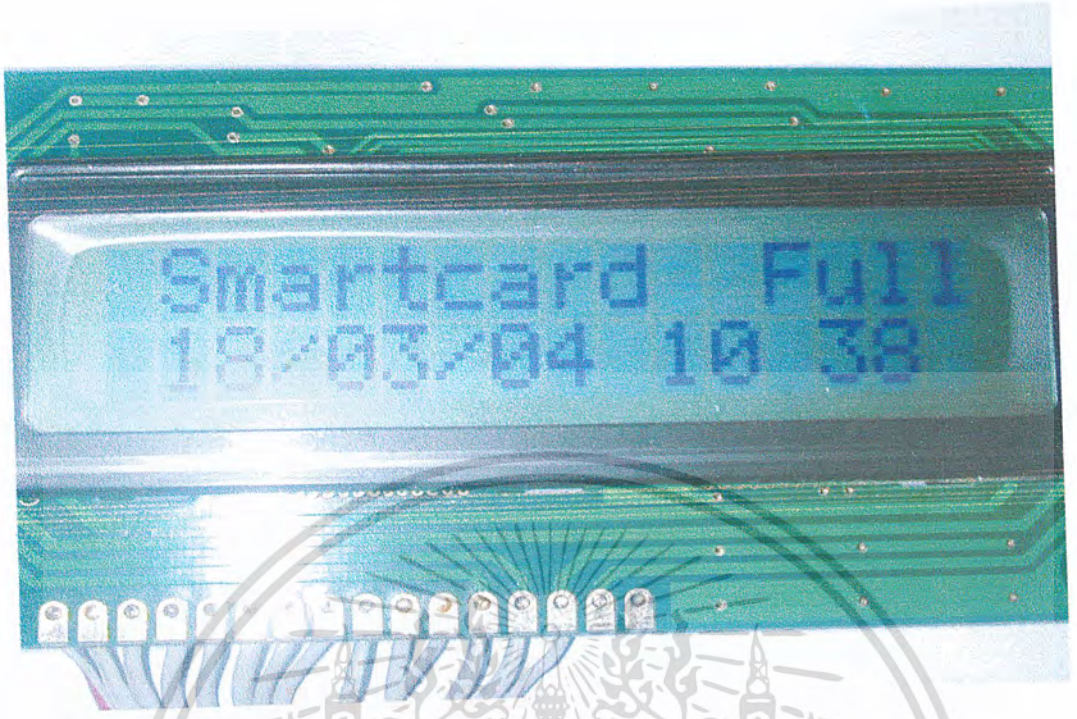
3.1.6 นำบัตรประจำตัวนักศึกษารูดผ่านที่เครื่องอ่านบาร์โค้ด รหัสนักศึกษาที่อ่านได้จะแสดงที่หน้าจอแอลซีดีและข้อมูลถูกเขียนลงสมาร์ตการ์ด ดังรูปที่ จ.6

3.1.7 นำบัตรนักศึกษาแต่ละคนมารูดผ่านเครื่องอ่านบาร์โค้ดไปเรื่อยๆ จนกว่าหน้าจอแอลซีดีจะแสดงคำว่า "Smartcard Full Please Chang" ซึ่งแสดงว่าข้อมูลในบัตรสมาร์ตการ์ดได้เต็มแล้วให้ทำการเปลี่ยนบัตรสมาร์ตการ์ดใหม่ บัตรสมาร์ตการ์ดเบอร์ SLE 4428 หนึ่งใบจะรับข้อมูลรหัสประจำตัวนักศึกษาได้ไม่เกิน 53 คน ดังรูปที่ จ.7



รูปที่ จ.6 หน้าจอแอลซีดีแสดงผลการรับข้อมูลจากเครื่องอ่านบาร์โค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.7 หน้าจอแอลซีดีแสดงผลว่าสมาร์ตการ์ดเก็บข้อมูลเต็มและให้เปลี่ยนบัตร

3.2 การใช้งานโมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด

3.2.1 นำบัตรสมาร์ตการ์ดที่เขียนข้อมูลลงในบัตรแล้ว มาเสียบเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด ดังรูปที่ จ.8

3.2.2 ทำการเปิดฟอร์มการอ่านข้อมูลจากโปรแกรมอ่านเขียนสมาร์ตการ์ดเพื่อทำการบันทึกข้อมูลในบัตรสมาร์ตการ์ดลงในตารางในโปรแกรมไมโครซอฟท์แอคเซส ดังรูป จ.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

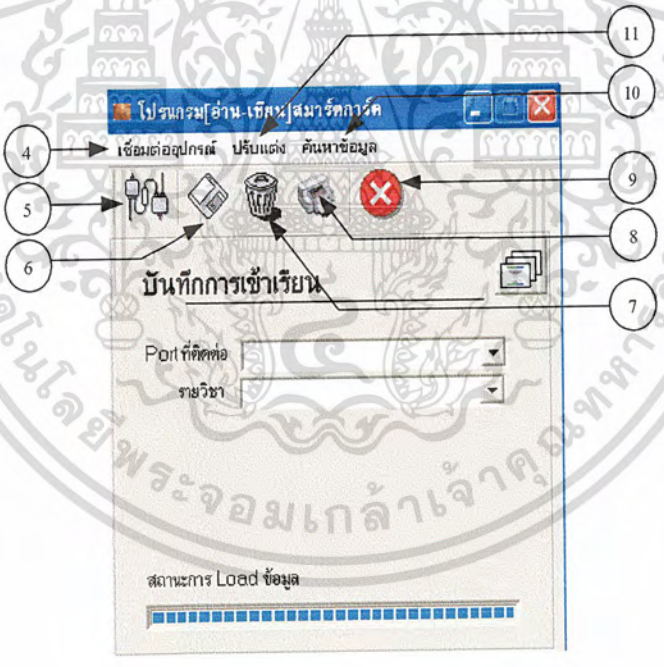
รูปที่ จ.8 แสดงการเสียบบัตรสมาร์ตการ์ดเข้าที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด

3.3 การใช้งานโปรแกรมอ่าน-เขียนสมาร์ตการ์ด

รูปที่ จ.9 หน้าต่างโปรแกรมการอ่านข้อมูลจากสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ จ.10 หน้าต่างโปรแกรมให้ป้อนชื่อผู้ใช้งานและรหัสผ่าน



รูปที่ จ.11 หน้าต่างโปรแกรมการใช้งานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ จ.10 และรูปที่ จ.11 มีรายละเอียดดังนี้

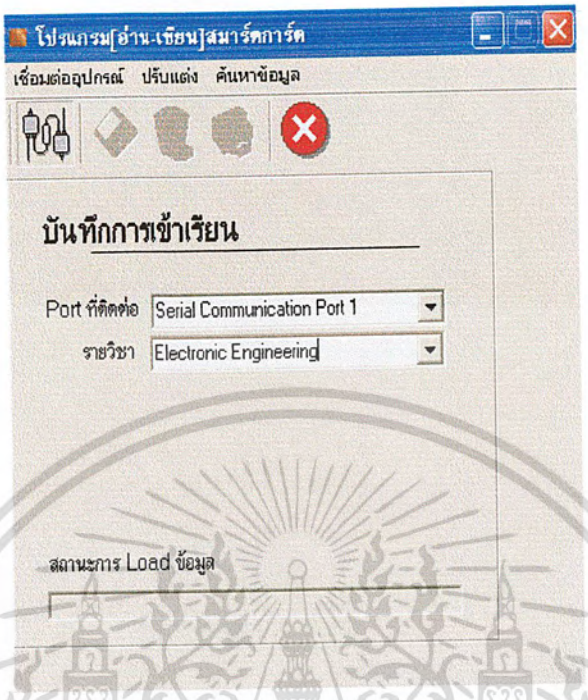
1. ช่องป้อนชื่อผู้ใช้งาน
2. ช่องป้อนรหัสผ่าน
3. ปุ่มออกจากโปรแกรมการป้อนรหัสผ่าน
4. ปุ่มเลือกการเชื่อมต่อของพอร์ตอนุกรม
5. ปุ่มโหลดข้อมูลจากสมาร์ตการ์ด
6. ปุ่มบันทึกข้อมูลที่แก้ไข เช่น การเพิ่มข้อมูลนักศึกษา การแก้ไขข้อมูล เป็นต้น
7. ปุ่มลบข้อมูลต่างๆ เช่น การลบรายชื่อนักศึกษา การลบรายวิชา เป็นต้น
8. ปุ่มพิมพ์ข้อมูลการเข้าชั้นเรียนของนักศึกษา
9. ปุ่มออกจากโปรแกรมอ่าน-เขียนสมาร์ตการ์ด
10. ปุ่มค้นหาข้อมูลการเข้าชั้นเรียน
11. ปุ่มเพิ่มเติมหรือแก้ไขข้อมูลนักศึกษาและข้อมูลทั่วไป

3.3.1 เปิดโปรแกรมเครื่องอ่าน-เขียนสมาร์ตการ์ดจากไอคอน ป้อนชื่อผู้ใช้งานและรหัสผ่าน จากนั้นจะเข้าสู่หน้าต่างโปรแกรมหลัก ดังรูปที่ จ.12

รูปที่ จ.12 หน้าต่างโปรแกรมการป้อนชื่อผู้ใช้และรหัสผ่าน

3.3.2 กดปุ่ม "เชื่อมต่ออุปกรณ์" เลือกรายวิชาและพอร์ตอนุกรมที่ใช้ในการติดต่อ เสร็จแล้วก็ทำการเลือกรายวิชาที่ทำการสอน จากนั้นทำการบันทึกการเข้าชั้นเรียน ข้อมูลก็จะไปเก็บที่ตารางในโปรแกรมไมโครซอฟท์แอคเซส ดังรูปที่ จ.13

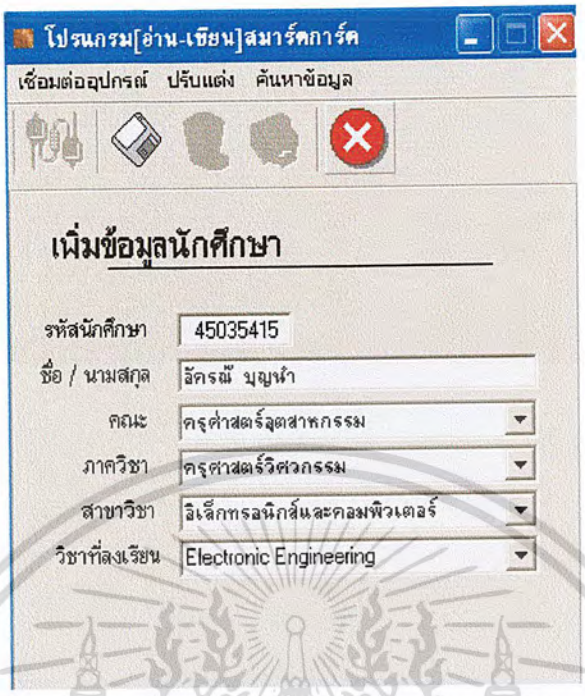
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.13 หน้าต่างโปรแกรมให้เลือกพอร์ตการติดต่อและรายวิชาที่จะโหลดข้อมูลเก็บ

3.3.3 การเพิ่มข้อมูลนักศึกษา คือ การเพิ่มรหัสนักศึกษา ชื่อนามสกุล คณะ ภาควิชา สาขาวิชา และวิชาที่ต้องการลงทะเบียน แล้วทำการกดปุ่มบันทึกข้อมูล ข้อมูลก็จะถูกเก็บที่ตารางในโปรแกรมไมโครซอฟท์แอคเซส ดังรูปที่ จ.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



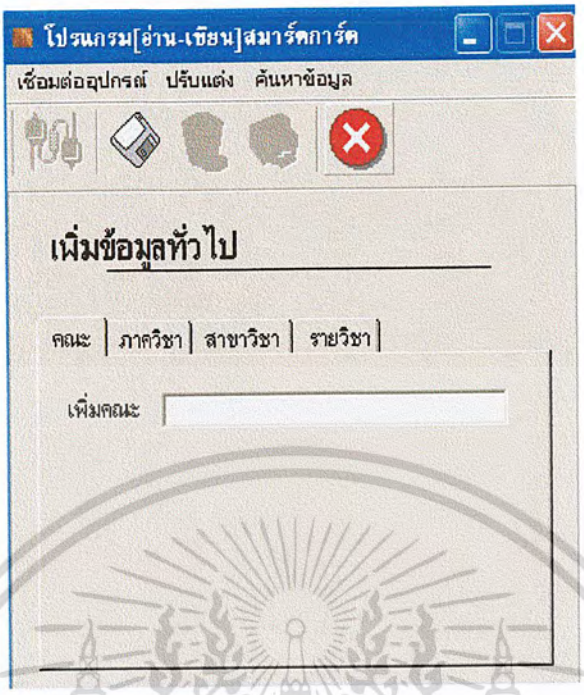
รูปที่ จ.14 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลนักศึกษา

3.3.4 การเพิ่มข้อมูลข้อมูลทั่วไป คือ การเพิ่มคณะ ภาควิชา สาขาวิชา และรายวิชา ทำการเลือกเพิ่มข้อมูลเสร็จและให้ทำการกดปุ่มบันทึกเพื่อนำข้อมูลที่เพิ่มไปเก็บที่ตารางในโปรแกรมไมโครซอฟท์แอคเซส ดังรูปที่ จ.15

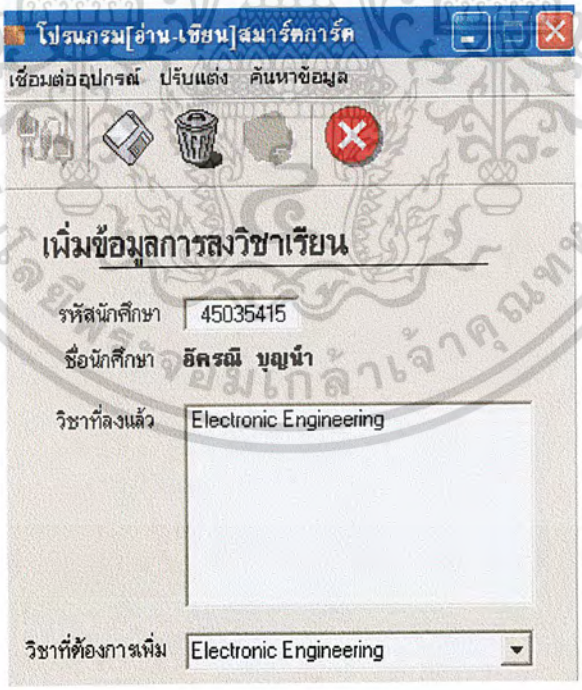
3.3.5 การเพิ่มข้อมูลการลงวิชาเรียน คือ รหัสนักศึกษา และวิชาที่ต้องการเพิ่มข้อมูลในการลงวิชาเรียน เมื่อทำการเพิ่มข้อมูลเสร็จเรียบร้อยแล้วให้ทำการกดปุ่มบันทึก เพื่อนำข้อมูลที่เพิ่มไปทำการเก็บไว้ในตารางที่โปรแกรมไมโครซอฟท์แอคเซส ดังรูปที่ จ.16

3.3.6 การแก้ไขข้อมูลนักศึกษา คือ การแก้ไขรหัสนักศึกษา ชื่อนามสกุล คณะ ภาควิชา สาขาวิชา และวิชาที่ลงทะเบียน เมื่อทำการแก้ไขข้อมูลใหม่เรียบร้อยแล้วให้ทำการกดปุ่มบันทึก หรือถ้าต้องการลบก็ให้ทำการกดปุ่มลบ จากนั้นข้อมูลที่ทำการแก้ไขก็จะไปเก็บยังตารางข้อมูลในโปรแกรมไมโครซอฟท์แอคเซส ดังรูปที่ จ.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

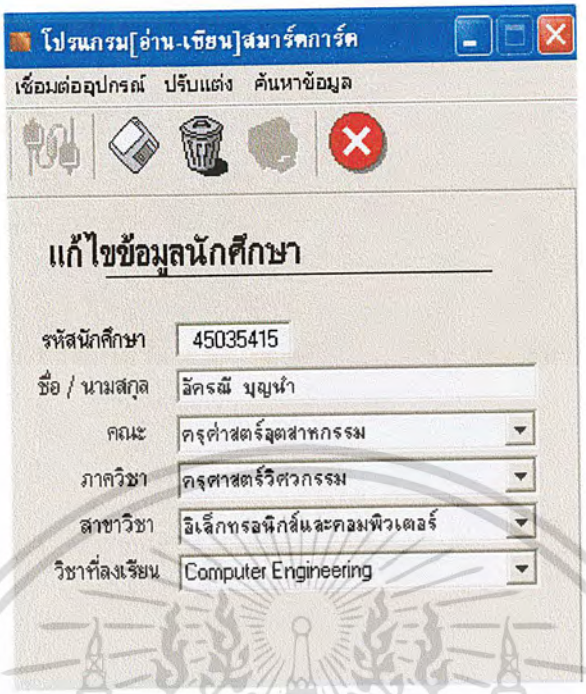


รูปที่ จ.15 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลทั่วไป



รูปที่ จ.16 หน้าต่างโปรแกรมให้เลือกการเพิ่มข้อมูลการลงวิชาเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

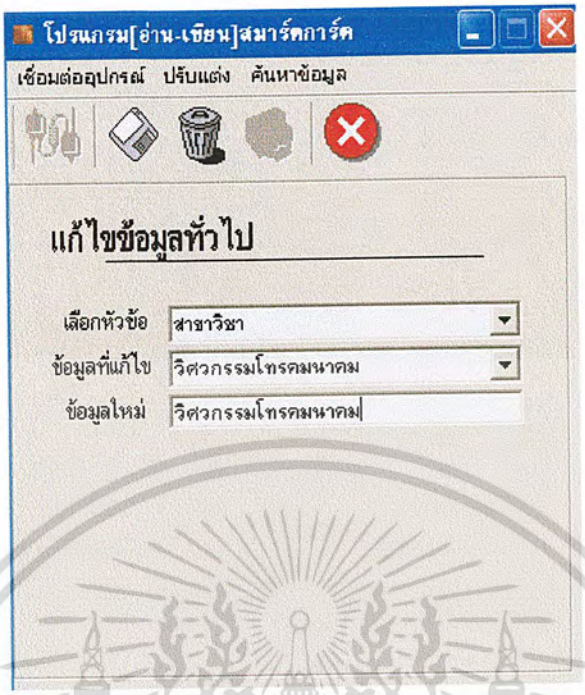


รูปที่ จ.17 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลนักศึกษา

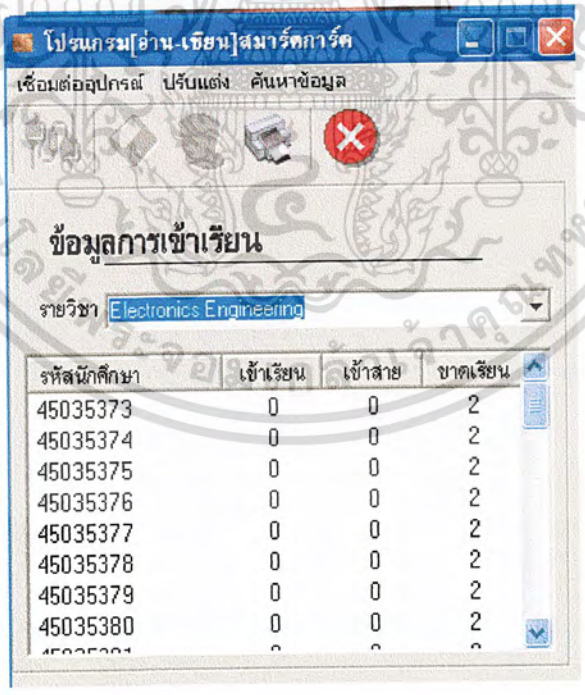
3.3.7 การแก้ไขข้อมูลทั่วไป คือ การแก้ไขคณะ ภาควิชา สาขาวิชา และรายวิชาที่ลงทะเบียน เมื่อทำการแก้ไขข้อมูลใหม่เรียบร้อยแล้วให้ทำการกดปุ่มบันทึกหรือถ้าต้องการลบก็ให้ทำการกดปุ่มลบ จากนั้นข้อมูลที่ทำการแก้ไขก็จะไปเก็บยังตารางข้อมูลในโปรแกรมไมโครซอฟท์แอคเซส ดังรูปที่ จ.18

3.3.8 ขั้นตอนการประมวลผลการเข้าเรียนของนักศึกษาให้ทำการเลือกที่ปุ่มค้นหาข้อมูลแล้ว เลือกข้อมูลการเข้าเรียนโปรแกรมก็จะทำการประมวลผลการเข้าเรียนการขาดเรียน และการเข้าสาย จากนั้นถ้าต้องการดูข้อมูลที่เป็นเอกสารให้ทำการกดที่ปุ่มเครื่องพิมพ์ จากนั้นโปรแกรมก็จะทำการพิมพ์หน้านี้ออกมาให้ ดังรูปที่ จ.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ จ.18 หน้าต่างโปรแกรมให้เลือกการแก้ไขข้อมูลทั่วไป



รูปที่ จ.19 หน้าต่างโปรแกรมเพื่อค้นหาข้อมูลการเข้าชั้นเรียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. การแก้ปัญหาเบื้องต้น

เมื่อท่านประสบปัญหาในการใช้งานเครื่องตรวจสอบและบันทึกการเข้าชั้นเรียนโดยสมาร์ตการ์ด สามารถตรวจสอบแนวทางแก้ไขปัญหาเบื้องต้นได้จากตารางข้างล่างนี้

อาการ	สาเหตุและ/หรือวิธีการแก้ไข
หลอดแอลอีดีสีแดงไม่ติด	ตรวจสอบปลั๊กไฟเสียบที่อแด็ปเตอร์, ตรวจสอบอะแด็ปเตอร์, ขั้วต่อสายไฟ
หลอดแอลอีดีสีเขียวไม่ติด	ตรวจสอบการเสียบสมาร์ตการ์ดว่าสนิทหรือไม่ และตรวจสอบหลอดแอลอีดีสีเขียว
รูดบัตรประตัวนักศึกษาไม่ได้	ตรวจสอบสภาพของบัตรประจำตัวนักศึกษา, รูดบัตรสูงกว่าหัวอ่านบาร์โค้ด วางบัตรให้สนิทก่อนที่จะรูด
หน้าจอคอมพิวเตอร์แสดง “Cannot Open Port”	ตรวจสอบการต่อสายพอร์ตอนุกรม, ตรวจสอบว่าสายภายในขั้วต่อว่าขาด, ไม่ได้เสียบสมาร์ตการ์ดที่โมดูลเครื่องอ่าน-เขียนสมาร์ตการ์ด
ตั้งค่าของวันเดือนปี และเวลาแล้วไม่เป็นปัจจุบัน	ตรวจสอบแบตเตอรี่ของไอซีเรียลไทม์คล็อก, เปลี่ยนแบตเตอรี่ใหม่
จอแสดงผลแอลซีดีไม่สว่าง	ปรับตัวต้านทานที่ใช้ควบคุมความสว่างของจอแอลซีดี

5. การดูแลรักษาและข้อควรระวัง

5.1 การดูแลรักษา

- ถอดสมาร์ตการ์ดออกจากเครื่องทุกครั้งหลังใช้งานเสร็จ
- ถอดอะแด็ปเตอร์ทุกครั้งหลังใช้งานเสร็จ
- ควรเก็บสมาร์ตการ์ดในซองสำหรับเก็บบัตรให้เรียบร้อย

5.2 ข้อควรระวัง

- อย่าให้หน้าสัมผัสของสมาร์ตการ์ดเกิดรอยขีดข่วนอย่างรุนแรง
- ระวังอย่าป้อนไฟเลี้ยงผ่านทางหัวต่อของเครื่องอ่านบาร์โค้ด
- ระวังการเสียบสมาร์ตการ์ดผิดด้าน
- ควรเสียบสมาร์ตการ์ดให้สนิท สังเกตได้จากหลอดไฟแสดงสถานะการเสียบบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ข้อมูลจำเพาะ

คุณสมบัติ	รายละเอียด
แรงดันไฟฟ้าที่ใช้งาน	DC 12 -18 โวลต์
ไมโครคอนโทรลเลอร์	ตระกูล MCS-51 เบอร์ AT89S8252
ส่วนแสดงผล	จอแอลซีดีและหลอดไฟแอลอีดี
การส่งผ่านข้อมูล	ใช้การส่งข้อมูลผ่านทางพอร์ตอนุกรม RS-232
บัตรสมาร์ตการ์ด	เบอร์ SLE4428 ของบริษัทซีเมนส์
เครื่องอ่านบาร์โค้ด	Arex Tek 103 DR
เครื่องอ่าน-เขียนสมาร์ตการ์ด	WACR20S
น้ำหนัก	1500 กรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ฉ

รายละเอียดและคุณสมบัติของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 4V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low-power Idle and Power-down Modes
- Interrupt Recovery From Power-down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S8252 provides the following standard features: 8K bytes of downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.

8-bit Microcontroller with 8K Bytes Flash

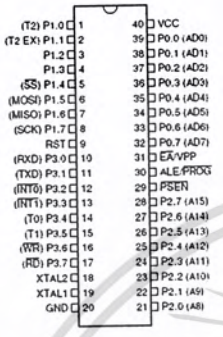
AT89S8252

Rev. 0401E–02/00

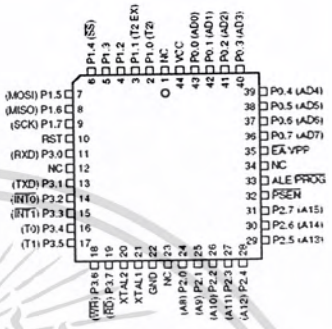
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

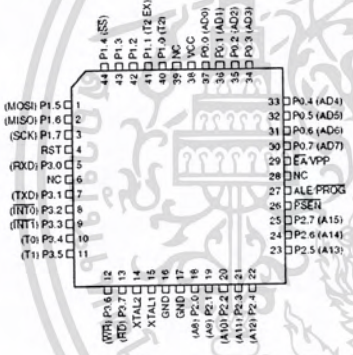
PDIP



PLCC



PQFP/TQFP



Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bbi-didirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

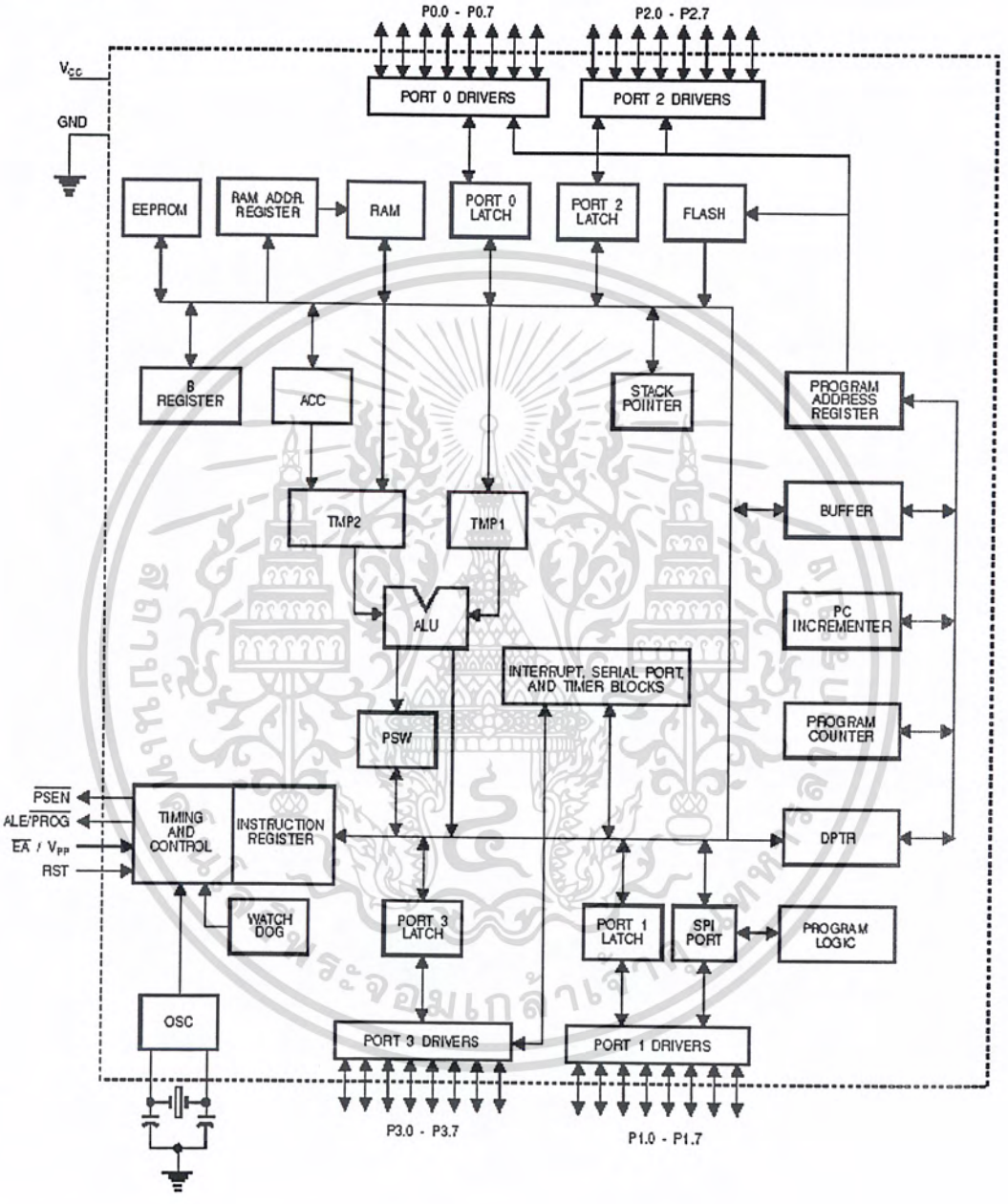
Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively.

Pin Description

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)
P1.4	$\overline{SS}$ (Slave port select input)
P1.5	MOSI (Master data output, slave data input pin for SPI channel)
P1.6	MISO (Master data input, slave data output pin for SPI channel)
P1.7	SCK (Master clock output, slave clock input pin for SPI channel)

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8 bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89S8252 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/VPP

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external pro-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

gram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions. This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Table 1. AT89S8252 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000					SPCR 000001XX			0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000		SPSR 00XXXXXX						0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111						WMCON 00000010		97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DP0L 00000000	DP0H 00000000	DP1L 00000000	DP1H 00000000	SPDR XXXXXXXX	PCON 0XXX0000	87H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Watchdog and Memory Control Register The WMCON register contains control bits for the Watchdog Timer (shown in Table 3). The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

Table 3. WMCON—Watchdog and Memory Control Register

WMCON Address = 96H — Reset Value = 0000 0010B

Bit	PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN
	7	6	5	4	3	2	1	0

Symbol	Function
PS2 PS1 PS0	Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms.
EEMWE	EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed.
EEMEN	Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory.
DPS	Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1
WDTRST RDY/$\overline{BSY}$	Watchdog Timer Reset and EEPROM Ready/$\overline{Busy}$ Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/$\overline{BSY}$ flag in a Read-Only mode during EEPROM write. RDY/$\overline{BSY}$ = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/$\overline{BSY}$ bit equals "0" and is automatically reset to "1" when programming is completed.
WDTEN	Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer.

SPI Registers Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

Interrupt Registers The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

Dual Data Pointer Registers To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

Power Off Flag The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Memory – EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

```
MOV 0A0H, #data
```

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

```
MOV @R0, #data
```

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

The on-chip EEPROM data memory is selected by setting the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0".

The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/$\overline{BSY}$ bit (read-only) in SFR WMCON. RDY/$\overline{BSY}$ = 0 means programming is still in progress and RDY/$\overline{BSY}$ = 1 means EEPROM write cycle is completed and another write cycle can be initiated.

In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at V_{CC} = 5V) are within ±30% of the nominal.

The WDT is disabled by Power-on Reset and during Power-down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

Table 7. Watchdog Timer Period Selection

WDT Prescaler Bits			
PS2	PS1	PS0	Period (nominal)
0	0	0	16 ms
0	0	1	32 ms
0	1	0	64 ms
0	1	1	128 ms
1	0	0	256 ms
1	0	1	512 ms
1	1	0	1024 ms
1	1	1	2048 ms

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Microcontroller Data Book, page 2-45, section titled, "Timer/Counters."

Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

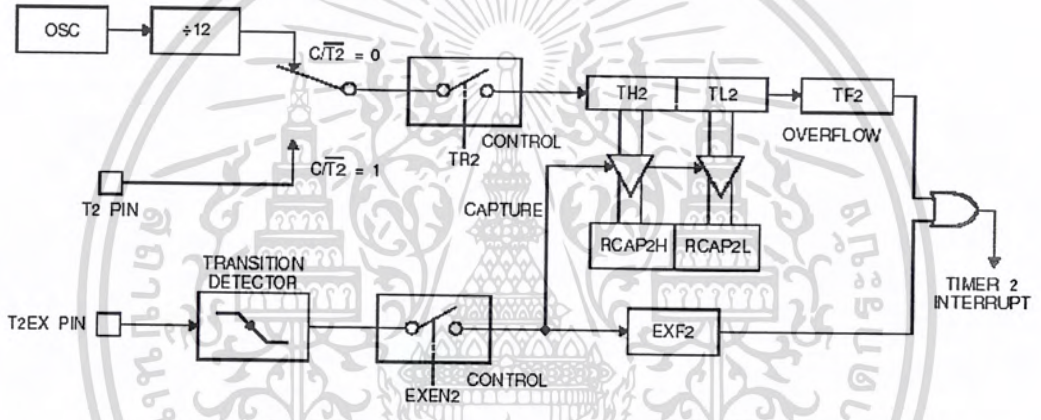
Table 8. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-bit Auto-reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Figure 1. Timer 2 in Capture Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash and EEPROM Parallel Programming Modes

Mode	RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7	Data I/O P0.7:0	Address P2.5:0 P1.7:0
Serial Prog. Modes	H	h(1)	h(1)	x						
Chip Erase	H	L	(2)	12V	H	L	L	L	X	X
Write (10K bytes) Memory	H	L		12V	L	H	H	H	DIN	ADDR
Read (10K bytes) Memory	H	L	H	12V	L	L	H	H	DOUT	ADDR
Write Lock Bits:	H	L		12V	H	L	H	L	DIN	X
Bit - 1									P0.7 = 0	X
Bit - 2									P0.6 = 0	X
Bit - 3									P0.5 = 0	X
Read Lock Bits:	H	L	H	12V	H	H	L	L	DOUT	X
Bit - 1									@P0.2	X
Bit - 2									@P0.1	X
Bit - 3									@P0.0	X
Read Atmel Code	H	L	H	12V	L	L	L	L	DOUT	30H
Read Device Code	H	L	H	12V	L	L	L	L	DOUT	31H
Serial Prog. Enable	H	L	(2)	12V	L	H	L	H	P0.0 = 0	X
Serial Prog. Disable	H	L	(2)	12V	L	H	L	H	P0.0 = 1	X
Read Serial Prog. Fuse	H	L	H	12V	H	H	L	H	@P0.0	X

Notes:
1. "h" = weakly pulled "High" internally.
2. Chip Erase and Serial Programming Fuse require a 10 ms PROG pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
3. P3.4 is pulled Low during programming to indicate RDY/BSY.
4. "X" = don't care

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0V ± 20%, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage [1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.5	V
V_{OL1}	Output Low-voltage [1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.5	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, V_{CC} = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	uA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power-down Mode [2]	V_{CC} = 6V		100	uA
		V_{CC} = 3V		40	uA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin: 10 mA
Maximum I_{OL} per 8-bit port:
Port 0: 26 mA
Ports 1, 2, 3: 15 mA
Maximum total I_{OL} for all output pins: 71 mA
If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

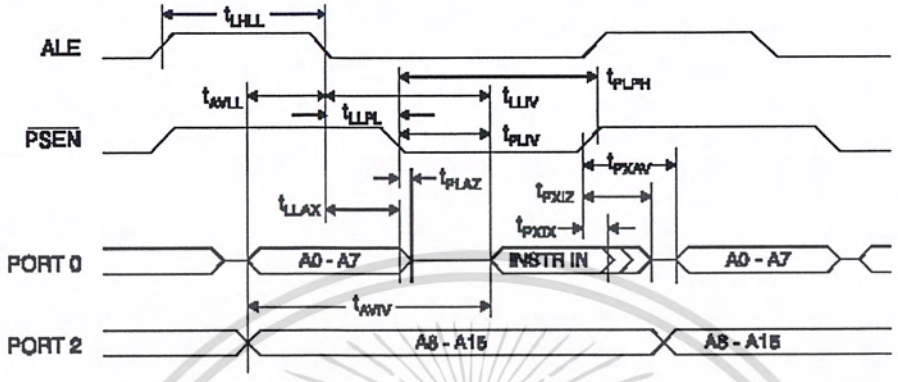
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

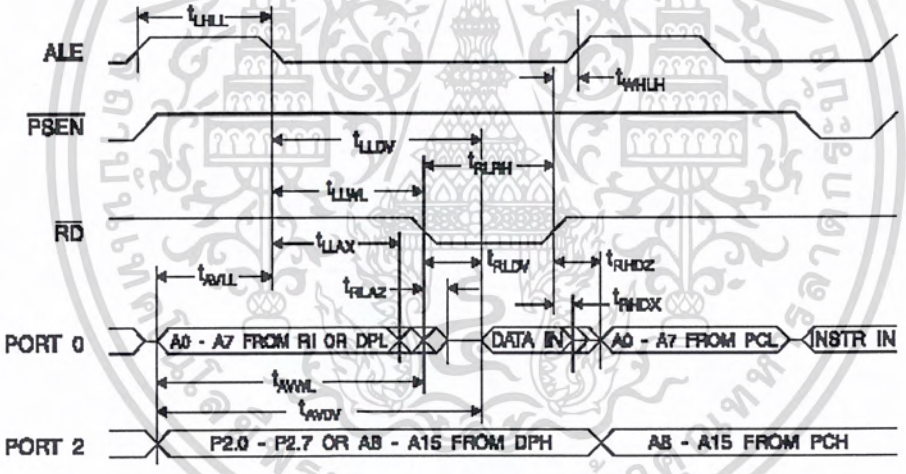
Symbol	Parameter	Variable Oscillator		Units
		Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{LHLL}	ALE Pulse Width	$2t_{CLCL}$ - 40		ns
t_{AVLL}	Address Valid to ALE Low	t_{CLCL} - 13		ns
t_{LLAX}	Address Hold after ALE Low	t_{CLCL} - 20		ns
t_{LLIV}	ALE Low to Valid Instruction In		$4t_{CLCL}$ - 65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	t_{CLCL} - 13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	$3t_{CLCL}$ - 20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		$3t_{CLCL}$ - 45	ns
t_{PXIX}	Input Instruction Hold after $\overline{PSEN}$	0		ns
t_{PXIZ}	Input Instruction Float after $\overline{PSEN}$		t_{CLCL} - 10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	t_{CLCL} - 8		ns
t_{AVIV}	Address to Valid Instruction In		$5t_{CLCL}$ - 55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	$6t_{CLCL}$ - 100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	$6t_{CLCL}$ - 100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		$5t_{CLCL}$ - 90	ns
t_{RHDX}	Data Hold after $\overline{RD}$	0		ns
t_{RHDZ}	Data Float after $\overline{RD}$		$2t_{CLCL}$ - 28	ns
t_{LLDV}	ALE Low to Valid Data In		$8t_{CLCL}$ - 150	ns
t_{AVDV}	Address to Valid Data In		$9t_{CLCL}$ - 165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	$3t_{CLCL}$ - 50	$3t_{CLCL}$ + 50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	$4t_{CLCL}$ - 75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	t_{CLCL} - 20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	$7t_{CLCL}$ - 120		ns
t_{WHQX}	Data Hold after $\overline{WR}$	t_{CLCL} - 20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	t_{CLCL} - 20	t_{CLCL} + 25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

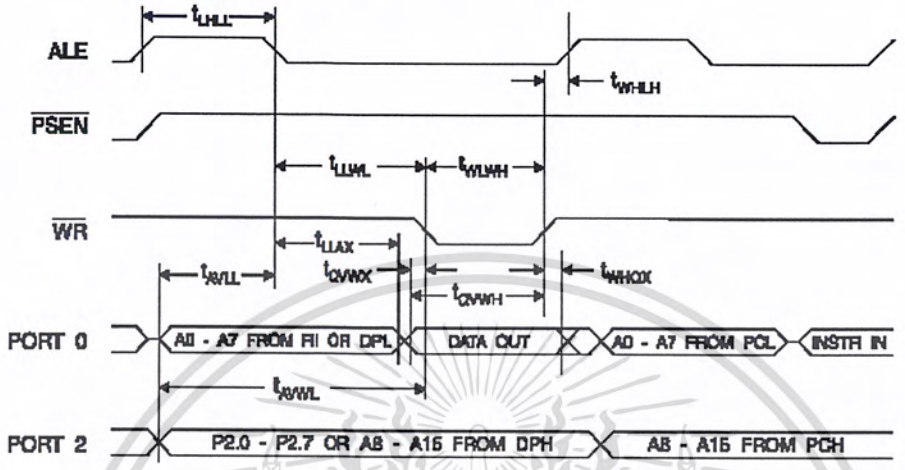


External Data Memory Read Cycle

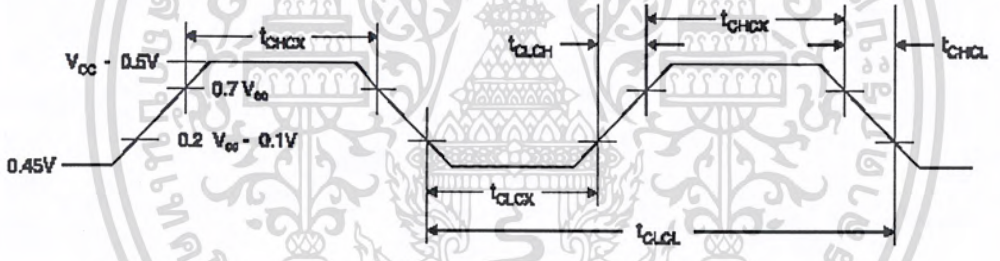


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



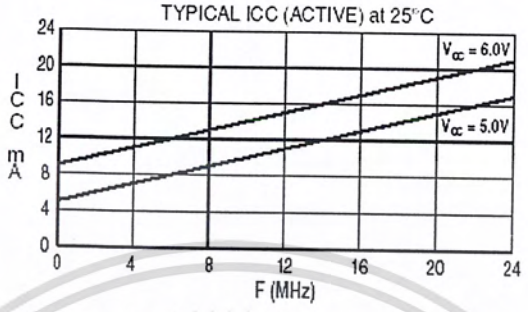
External Clock Drive

Symbol	Parameter	V_{CC} = 4.0V to 6.0V		Units
		Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

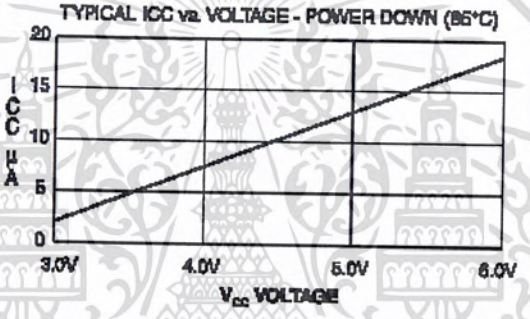
AT89S8252

TYPICAL ICC (ACTIVE) at 25°C



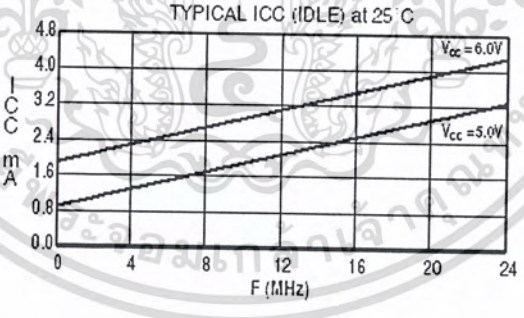
AT89S8252

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



AT89S8252

TYPICAL ICC (IDLE) at 25°C



Notes: 1. XTAL1 tied to GND for Icc (power-down)
2. Lock bits programmed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

19-0175; Rev 3; 5/06

MAXIM

±15kV ESD-Protected, +5V RS-232 Transceivers

General Description

The MAX202E–MAX213E, MAX232E/MAX241E line drivers/receivers are designed for RS-232 and V.28 communications in harsh environments. Each transmitter output and receiver input is protected against ±15kV electrostatic discharge (ESD) shocks, without latchup. The various combinations of features are outlined in the *Selection Guide*. The drivers and receivers for all ten devices meet all EIA/TIA-232E and CCITT V.28 specifications at data rates up to 120kbps, when loaded in accordance with the EIA/TIA-232E specification.

The MAX211E/MAX213E/MAX241E are available in 28-pin SO packages, as well as a 28-pin SSOP that uses 60% less board space. The MAX202E/MAX232E come in 16-pin narrow SO, wide SO, and DIP packages. The MAX203E comes in a 20-pin DIP/SO package, and needs no external charge-pump capacitors. The MAX205E comes in a 24-pin wide DIP package, and also eliminates external charge-pump capacitors. The MAX206E/MAX207E/MAX208E come in 24-pin SO, SSOP, and narrow DIP packages. The MAX232E/MAX241E operate with four 1µF capacitors, while the MAX202E/MAX206E/MAX207E/MAX208E/MAX211E/MAX213E operate with four 0.1µF capacitors, further reducing cost and board space.

Applications

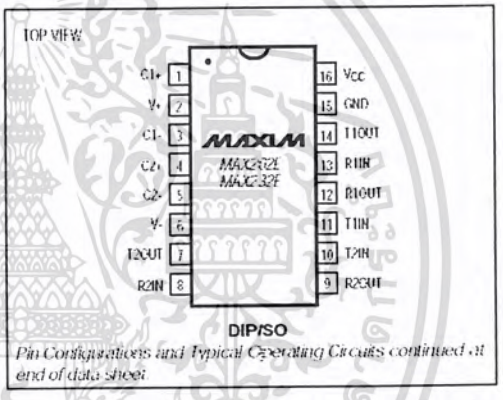
Notebook, Subnotebook, and Palmtop Computers
Battery-Powered Equipment
Hand-Held Equipment

Ordering Information appears at end of data sheet.

Features

- ♦ **ESD Protection for RS-232 I/O Pins:**
  ±15kV—Human Body Model
  ±8kV—IEC1000-4-2, Contact Discharge
  ±15kV—IEC1000-4-2, Air-Gap Discharge
- ♦ **Latchup Free (unlike bipolar equivalents)**
- ♦ **Guaranteed 120kbps Data Rate—LapLink™ Compatible**
- ♦ **Guaranteed 3V/µs Min Slew Rate**
- ♦ **Operate from a Single +5V Power Supply**

Pin Configurations



DIP/SO

Pin Configurations and Typical Operating Circuits continued at end of data sheet.

Selection Guide

PART	No. of RS-232 DRIVERS	No. of RS-232 RECEIVERS	RECEIVERS ACTIVE IN SHUTDOWN	No. of EXTERNAL CAPACITORS	LOW-POWER SHUTDOWN	TTL THREE-STATE
MAX202E	2	2	0	4 (0.1µF)	No	No
MAX203E	2	2	0	None	No	No
MAX205E	5	5	0	None	Yes	Yes
MAX206E	4	3	0	4 (0.1µF)	Yes	Yes
MAX207E	5	3	0	4 (0.1µF)	No	No
MAX208E	4	4	0	4 (0.1µF)	No	No
MAX211E	4	5	0	4 (0.1µF)	Yes	Yes
MAX213E	4	5	2	4 (0.1µF)	Yes	Yes
MAX232E	2	2	0	4 (1µF)	No	No
MAX241E	4	5	0	4 (1µF)	Yes	Yes

LapLink is a registered trademark of Traveling Software, Inc.

For free samples & the latest literature: http://www.maxim-ic.com, or phone 1-800-998-8800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

ABSOLUTE MAXIMUM RATINGS

V_{CC} ..... -0.3V to +6V
$V+$ ..... (V_{CC} - 0.3V) to +14V
$V-$ ..... +14V to -0.3V
Input Voltages
T_IN ..... -0.3V to ($V+$ + 0.3V)
R_IN ..... ±30V
Output Voltages
T_OUT ..... ($V-$ - 0.3V) to ($V+$ + 0.3V)
R_OUT ..... -0.3V to (V_{CC} + 0.3V)
Short-Circuit Duration, T_OUT ..... Continuous
Continuous Power Dissipation (T_A = +70°C)
16-Pin Plastic DIP (derate 10.53mW/°C above +70°C) 842mW
16-Pin Narrow SO (derate 8.70mW/°C above +70°C) 696mW
16-Pin Wide SO (derate 9.52mW/°C above +70°C) 762mW
20-Pin Plastic DIP (derate 11.11mW/°C above +70°C)....889mW
20-Pin SO (derate 10.00mW/°C above +70°C).....800mW
24-Pin Narrow Plastic DIP
(derate 13.33mW/°C above +70°C) 1.07W
24-Pin Wide Plastic DIP
(derate 14.29mW/°C above +70°C) 1.14W
24-Pin SO (derate 11.76mW/°C above +70°C).....941mW
24-Pin SSOP (derate 8.00mW/°C above +70°C) 640mW
28-Pin SO (derate 12.50mW/°C above +70°C).....1W
28-Pin SSOP (derate 9.52mW/°C above +70°C) 762mW
Operating Temperature Ranges
MAX2_ _EC_ _ ..... 0°C to +70°C
MAX2_ _EE_ _ ..... -40°C to +85°C
Storage Temperature Range ..... -65°C to +165°C
Lead Temperature (soldering, 10sec) ..... +300°C

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

ELECTRICAL CHARACTERISTICS

(V_{CC} = +5V ±10% for MAX202E/206E/208E/211E/213E/232E/241E; V_{CC} = +5V ±5% for MAX203E/205E/207E; C1–C4 = 0.1µF for MAX202E/206E/207E/208E/211E/213E; C1–C4 = 1µF for MAX232E/241E; T_A = T_{MIN} to T_{MAX}, unless otherwise noted. Typical values are at T_A = +25°C.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
DC CHARACTERISTICS							
V_{CC} Supply Current	I_{CC}	No load, T_A = 25°C	MAX202E/203E		8	15	mA
			MAX205E/208E		11	20	
			MAX211E/213E		14	20	
			MAX232E		5	10	
			MAX241E		7	15	
Shutdown Supply Current		T_A = +25°C, Figure 1	MAX205E/206E		1	10	µA
			MAX211E/241E		1	10	
			MAX213E		15	50	
LOGIC							
Input Pull-Up Current		T_IN = 0V (MAX205E–208E/211E/213E/241E)			15	200	µA
Input Leakage Current		T_IN = 0V to V_{CC} (MAX202E/203E/232E)				±10	µA
Input Threshold Low	V_{IL}	T_IN; EN, $\overline{SHDN}$ (MAX213E) or $\overline{EN}$, SHDN (MAX205E–208E/211E/241E)				0.8	V
Input Threshold High	V_{IH}	T_IN		2.0			V
		EN, $\overline{SHDN}$ (MAX213E) or $\overline{EN}$, SHDN (MAX205E–208E/211E/241E)		2.4			
Output Voltage Low	V_{OL}	R_OUT; I_{OUT} = 3.2mA (MAX202E/203E/232E) or I_{OUT} = 1.6mA (MAX205E/208E/211E/213E/241E)				0.4	V
Output Voltage High	V_{OH}	R_OUT; I_{OUT} = -1.0mA		3.5	V_{CC} - 0.4		V
Output Leakage Current		$\overline{EN}$ = V_{CC}, EN = 0V, 0V ≤ R_{OUT} ≤ V_{CC} MAX205E–208E/211E/213E/241E outputs disabled			-0.05	+10	µA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

ELECTRICAL CHARACTERISTICS (continued)

(VCC = +5V ±10% for MAX202E/206E/208E/211E/213E/232E/241E; VCC = +5V ±5% for MAX203E/205E/207E; C1–C4 = 0.1µF for MAX202E/206E/207E/208E/211E/213E; C1–C4 = 1µF for MAX232E/241E; TA = TMIN to TMAX; unless otherwise noted. Typical values are at TA = +25°C.)

PARAMETER	SYMBOL	CONDITIONS		MIN	TYP	MAX	UNITS
EIA/TIA-232E RECEIVER INPUTS							
Input Voltage Range				-30		30	V
Input Threshold Low		TA = +25°C, VCC = 5V	All parts, normal operation	0.8	1.2		V
			MAX213E, $\overline{SHDN}$ = 0V, EN = VCC	0.6	1.5		
Input Threshold High		TA = +25°C, VCC = 5V	All parts, normal operation		1.7	2.4	V
			MAX213E (R4, R5), $\overline{SHDN}$ = 0V, EN = VCC		1.5	2.4	
Input Hysteresis		VCC = 5V, no hysteresis in shutdown		0.2	0.5	1.0	V
Input Resistance		TA = +25°C, VCC = 5V		3	5	7	kΩ
EIA/TIA-232E TRANSMITTER OUTPUTS							
Output Voltage Swing		All drivers loaded with 3kΩ to ground (Note 1)		±5	±9		V
Output Resistance		VCC = V+ = V- = 0V, VOUT = ±2V		300			Ω
Output Short-Circuit Current					±10	±60	mA
TIMING CHARACTERISTICS							
Maximum Data Rate		RL = 3kΩ to 7kΩ, CL = 50pF to 1000pF, one transmitter switching		120			kbps
Receiver Propagation Delay	tPLHR, tPHLR	CL = 150pF	All parts, normal operation		0.5	10	µs
			MAX213E (R4, R5), $\overline{SHDN}$ = 0V, EN = VCC		4	40	
Receiver Output Enable Time		MAX205E/206E/211E/213E/241E normal operation, Figure 2			600		ns
Receiver Output Disable Time		MAX205E/206E/211E/213E/241E normal operation, Figure 2			200		ns
Transmitter Propagation Delay	tPLHT, tPHLT	RL = 3kΩ, CL = 2500pF, all transmitters loaded			2		µs
Transition-Region Slew Rate		TA = +25°C, VCC = 5V, RL = 3kΩ to 7kΩ, CL = 50pF to 1000pF, measured from +3V to -3V or -3V to +3V, Figure 3		3	6	30	V/µs
ESD PERFORMANCE: TRANSMITTER OUTPUTS, RECEIVER INPUTS							
ESD Protection Voltage		Human Body Model			±15		kV
		IEC1000-4-2, Contact Discharge			±8		
		IEC1000-4-2, Air-Gap Discharge			±15		

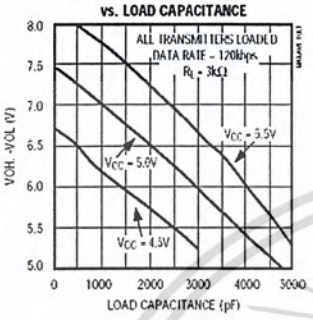
Note 1: MAX211EE_ _ tested with VCC = +5V ±5%.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

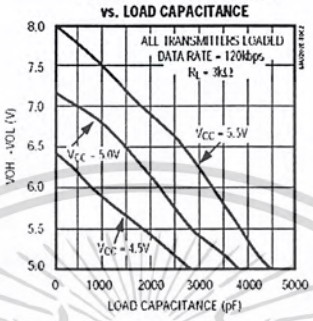
Typical Operating Characteristics

(Typical Operating Circuits, V_{CC} = +5V, T_A = +25°C, unless otherwise noted.)

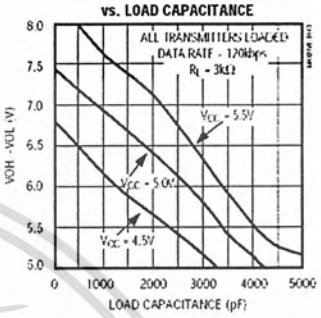
MAX232E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



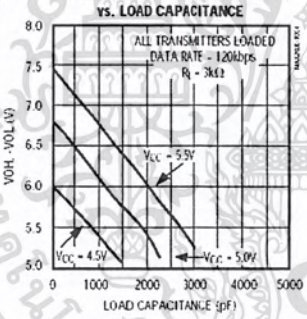
MAX202E/MAX203E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



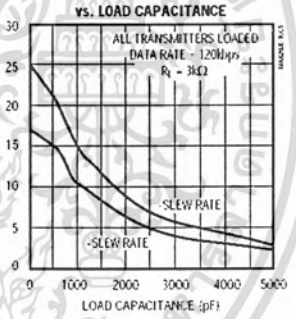
MAX241E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



MAX211E/MAX213E
TRANSMITTER OUTPUT VOLTAGE
vs. LOAD CAPACITANCE



MAX211E/MAX213E/MAX241E
TRANSMITTER SLEW RATE
vs. LOAD CAPACITANCE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

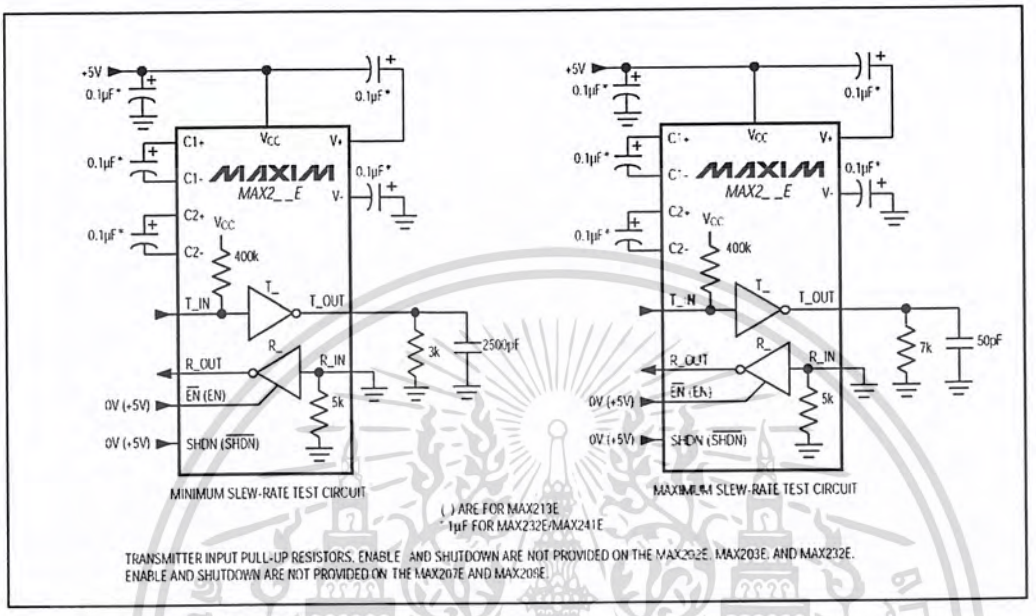


Figure 3. Transition Slew-Rate Circuit

Detailed Description

The MAX202E–MAX213E, MAX232E/MAX241E consist of three sections: charge-pump voltage converters, drivers (transmitters), and receivers. These E versions provide extra protection against ESD. They survive ±15kV discharges to the RS-232 inputs and outputs, tested using the Human Body Model. When tested according to IEC1000-4-2, they survive ±8kV contact-discharges and ±15kV air-gap discharges. The rugged E versions are intended for use in harsh environments or applications where the RS-232 connection is frequently changed (such as notebook computers). The standard (non-"E") MAX202, MAX203, MAX205–MAX208, MAX211, MAX213, MAX232, and MAX241 are recommended for applications where cost is critical.

+5V to ±10V Dual Charge-Pump Voltage Converter

The +5V to ±10V conversion is performed by dual charge-pump voltage converters (Figure 4). The first charge-pump converter uses capacitor C1 to double the +5V into +10V, storing the +10V on the output filter capacitor, C3. The second uses C2 to invert the +10V into -10V, storing the -10V on the V- output filter capacitor, C4.

In shutdown mode, V+ is internally connected to VCC by a 1kΩ pull-down resistor, and V- is internally connected to ground by a 1kΩ pull-up resistor.

RS-232 Drivers

With VCC = 5V, the typical driver output voltage swing is ±8V when loaded with a nominal 5kΩ RS-232 receiver. The output swing is guaranteed to meet EIA/TIA-232E and V.28 specifications that call for ±5V minimum output levels under worst-case conditions. These include a 3kΩ load, minimum VCC, and maximum operating temperature. The open-circuit output voltage swings from (V+ - 0.6V) to V-.

Input thresholds are CMOS/TTL compatible. The unused drivers' inputs on the MAX205E–MAX208E, MAX211E, MAX213E, and MAX241E can be left unconnected because 400kΩ pull-up resistors to VCC are included on-chip. Since all drivers invert, the pull-up resistors force the unused drivers' outputs low. The MAX202E, MAX203E, and MAX232E do not have pull-up resistors on the transmitter inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When in low-power shutdown mode, the MAX205E/MAX206E/MAX211E/MAX213E/MAX241E driver outputs are turned off and draw only leakage currents—even if they are back-driven with voltages between 0V and 12V. Below -0.5V in shutdown, the transmitter output is diode-clamped to ground with a 1kΩ series impedance.

RS-232 Receivers

The receivers convert the RS-232 signals to CMOS-logic output levels. The guaranteed 0.8V and 2.4V receiver input thresholds are significantly tighter than the ±3V thresholds required by the EIA/TIA-232E specification. This allows the receiver inputs to respond to TTL/CMOS-logic levels, as well as RS-232 levels.

The guaranteed 0.8V input low threshold ensures that receivers shorted to ground have a logic 1 output. The 5kΩ input resistance to ground ensures that a receiver with its input left open will also have a logic 1 output.

Receiver inputs have approximately 0.5V hysteresis. This provides clean output transitions, even with slow rise/fall-time signals with moderate amounts of noise and ringing.

In shutdown, the MAX213E's R4 and R5 receivers have no hysteresis.

Shutdown and Enable Control (MAX205E/MAX206E/MAX211E/MAX213E/MAX241E)

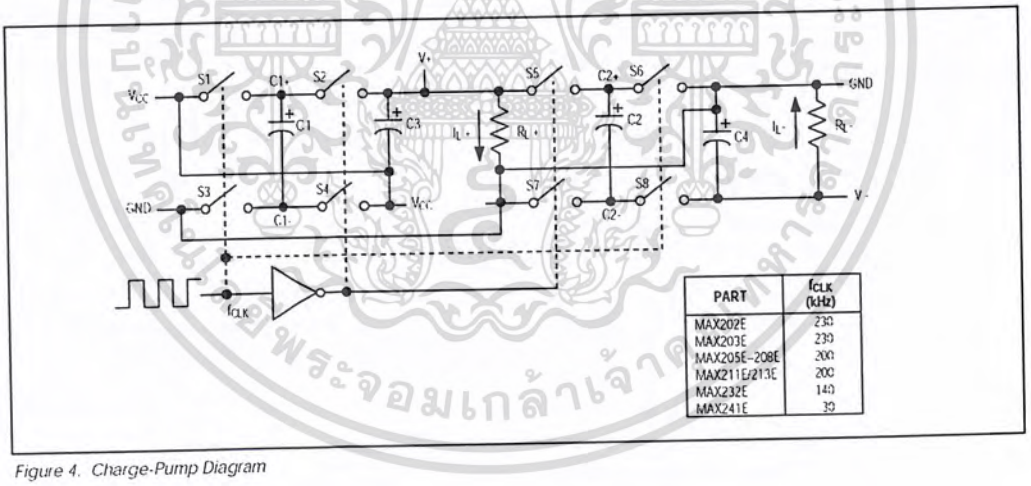
In shutdown mode, the charge pumps are turned off, V+ is pulled down to VCC, V- is pulled to ground, and the transmitter outputs are disabled. This reduces supply current typically to 1µA (15µA for the MAX213E). The time required to exit shutdown is under 1ms, as shown in Figure 5.

Receivers

All MAX213E receivers, except R4 and R5, are put into a high-impedance state in shutdown mode (see Tables 1a and 1b). The MAX213E's R4 and R5 receivers still function in shutdown mode. These two awake-in-shutdown receivers can monitor external activity while maintaining minimal power consumption.

The enable control is used to put the receiver outputs into a high-impedance state, to allow wire-OR connection of two EIA/TIA-232E ports (or ports of different types) at the UART. It has no effect on the RS-232 drivers or the charge pumps.

Note: The enable control pin is active low for the MAX211E/MAX241E (EN), but is active high for the MAX213E (EN). The shutdown control pin is active high for the MAX205E/MAX206E/MAX211E/MAX241E (SHDN), but is active low for the MAX213E (SHDN).



PART	fCLK (kHz)
MAX202E	230
MAX203E	230
MAX205E–208E	200
MAX211E/213E	200
MAX232E	140
MAX241E	39

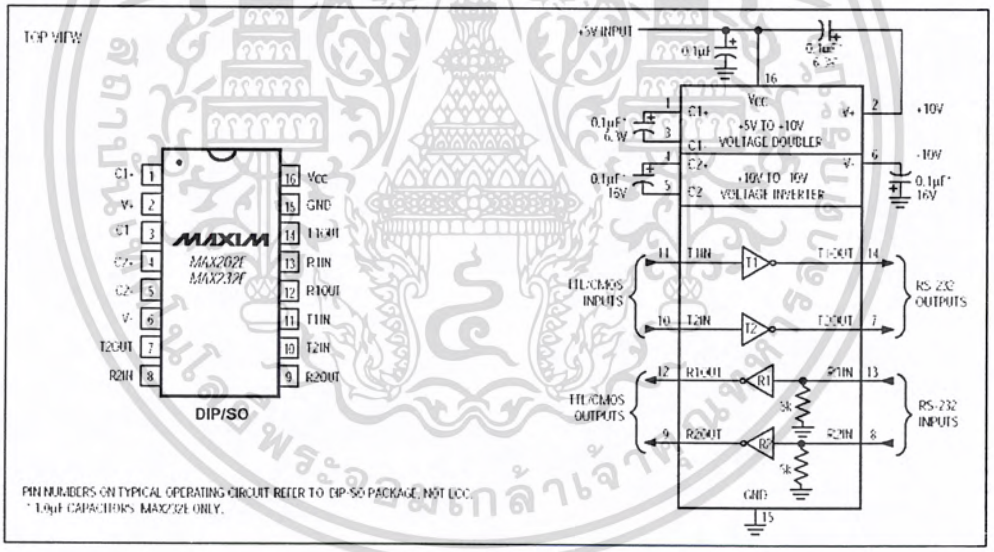
Figure 4. Charge-Pump Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 3. DB9 Cable Connections Commonly Used for EIA/TIAE-232E and V.24 Asynchronous Interfaces

PIN	CONNECTION	
1	Received Line Signal Detector (sometimes called Carrier Detect, DCD)	Handshake from DCE
2	Receive Data (RD)	Data from DCE
3	Transmit Data (TD)	Data from DTE
4	Data Terminal Ready	Handshake from DTE
5	Signal Ground	Reference point for signals
6	Data Set Ready (DSR)	Handshake from DCE
7	Request to Send (RTS)	Handshake from DTE
8	Clear to Send (CTS)	Handshake from DCE
9	Ring Indicator	Handshake from DCE

Pin Configurations and Typical Operating Circuits (continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

±15kV ESD-Protected, +5V RS-232 Transceivers

Ordering Information

PART	TEMP. RANGE	PIN-PACKAGE
MAX202ECPE	0°C to +70°C	16 Plastic DIP
MAX202ECSE	0°C to +70°C	16 Narrow SO
MAX202ECWE	0°C to +70°C	16 Wide SO
MAX202EC/D	0°C to +70°C	Dice*
MAX202EEPE	-40°C to +85°C	16 Plastic DIP
MAX202EESE	-40°C to +85°C	16 Narrow SO
MAX202EEWE	-40°C to +85°C	16 Wide SO
MAX203ECPP	0°C to +70°C	20 Plastic DIP
MAX203ECWP	0°C to +70°C	20 SO
MAX203EEPP	-40°C to +85°C	20 Plastic DIP
MAX203EEWP	-40°C to +85°C	20 SO
MAX205ECPG	0°C to +70°C	24 Wide Plastic DIP
MAX205EEPG	-40°C to +85°C	24 Wide Plastic DIP
MAX206ECNG	0°C to +70°C	24 Narrow Plastic DIP
MAX206ECWG	0°C to +70°C	24 SO
MAX206ECAG	0°C to +70°C	24 SSOP
MAX206EENG	40°C to +85°C	24 Narrow Plastic DIP
MAX206EEWG	-40°C to +85°C	24 SO
MAX206EEAG	-40°C to +85°C	24 SSOP
MAX207ECNG	0°C to +70°C	24 Narrow Plastic DIP
MAX207ECWG	0°C to +70°C	24 SO
MAX207ECAG	0°C to +70°C	24 SSOP
MAX207EENG	-40°C to +85°C	24 Narrow Plastic DIP
MAX207EEWG	-40°C to +85°C	24 SO
MAX207EEAG	40°C to +85°C	24 SSOP

PART	TEMP. RANGE	PIN-PACKAGE
MAX208ECNG	0°C to +70°C	24 Narrow Plastic DIP
MAX208ECWG	0°C to +70°C	24 SO
MAX208ECAG	0°C to +70°C	24 SSOP
MAX208EENG	40°C to +85°C	24 Narrow Plastic DIP
MAX208EEWG	-40°C to +85°C	24 SO
MAX208EEAG	-40°C to +85°C	24 SSOP
MAX211ECWI	0°C to +70°C	28 SO
MAX211ECAI	0°C to +70°C	28 SSOP
MAX211EEWI	-40°C to +85°C	28 SO
MAX211EEAI	40°C to +85°C	28 SSOP
MAX213ECWI	0°C to +70°C	28 SO
MAX213ECAI	0°C to +70°C	28 SSOP
MAX213EEWI	-40°C to +85°C	28 SO
MAX213EEAI	40°C to +85°C	28 SSOP
MAX232ECPE	0°C to +70°C	16 Plastic DIP
MAX232ECSE	0°C to +70°C	16 Narrow SO
MAX232ECWI	0°C to +70°C	16 Wide SO
MAX232EC/D	0°C to +70°C	Dice*
MAX232EEPE	40°C to +85°C	16 Plastic DIP
MAX232EESE	-40°C to +85°C	16 Narrow SO
MAX232EEWE	40°C to +85°C	16 Wide SO
MAX241ECWI	0°C to +70°C	28 SO
MAX241ECAI	0°C to +70°C	28 SSOP
MAX241EEWI	-40°C to +85°C	28 SO
MAX241EEAI	40°C to +85°C	28 SSOP

*Dice are specified at T_A = +25°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIEMENS

ICs for Chip Cards

SLE 4418 / SLE 4428
Intelligent 8-Kbi t EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLE 4418/SLE 4428 **Revision History:**	**04.94**
Previous Releases:	09.92
Page	Subjects (changes since last revision)
	Update

Data Classification

Maximum Ratings

Maximum ratings are absolute ratings; exceeding only one of these values may cause irreversible damage to the integrated circuit.

Characteristics

The listed characteristics are ensured over the operating range of the integrated circuit. Typical characteristics specify mean values expected over the production spread. If not otherwise specified, typical characteristics apply at T_A = 25 °C and the given supply voltage.

Operating Range

In the operating range the functions given in the circuit description are fulfilled. Important:

Important:	Further information is confidential and on request. Please contact: Siemens Semiconductor Group in Munich, Germany, Marketing Chip Card ICs. Tel.: 089-4144-2058 or 4144-4362 The supply of this component does not include a licence for its use in smart card applications. This licence is due to: INNOVATRON Patents 137 Boulevard de Sébastopol. 75002 Paris. France. Fax 33 (1) 40 13 39 09

This edition was realized using the software system FrameMaker®

Published by Siemens AG, Bereich Halbleiter, Marketing-Kommunikation, Balanstraße 73, D-81541 München.

© Siemens AG 1994. All Rights Reserved.

As far as patents or other rights of third parties are concerned, liability is only assumed for components, not for applications, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices please contact the Semiconductor Group Offices in Germany or the Siemens Companies and Representatives worldwide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office, Semiconductor Group.

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you - get in touch with your nearest sales office. By agreement we will take packing material back, if it is sorted. You must bear the costs of transport.

For packing material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

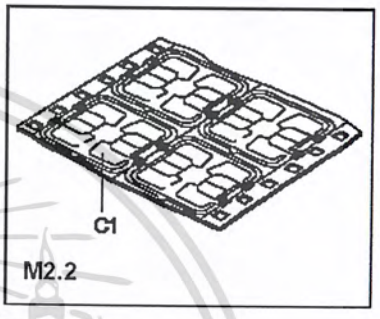
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SIEMENS

Intelligent 1-KByte EEPROM with Write Protect Function — SLE 4418
Intelligent 1-KByte EEPROM with Write Protect Function and Programmable Security Code (PSC) — SLE 4428

Features

- 1024 x 8 bit EEPROM organization
- Byte-wise addressing
- Irreversible byte-wise write protection
- 1024 x 1 bit protection memory organization
- Serial three-wire bus
- End of programming indicated on data output
- Minimum of 10^4 write/erase cycles
- Data retention for minimum of ten years
- Contact configuration and serial interface in accordance to ISO standard 7816 (synchronous transmission)

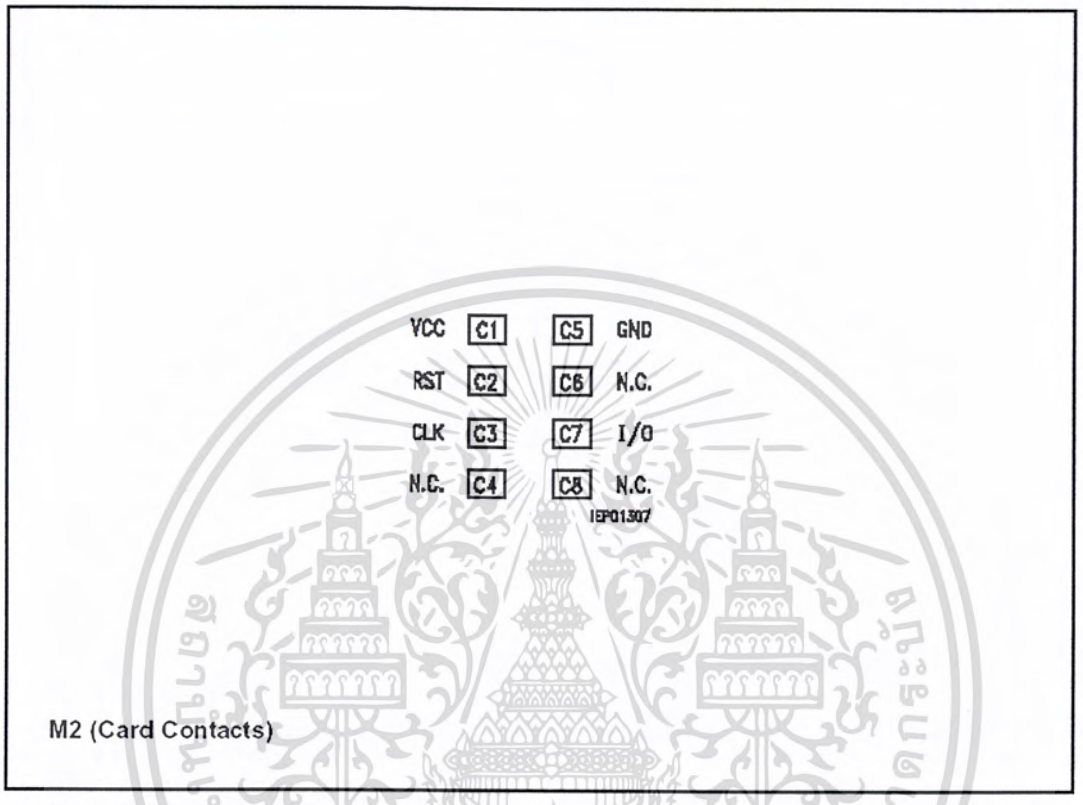


Additional Feature of SLE 4428

- Data can only be changed after entry of the correct 2-byte programmable security code (PSC)

Type	Ordering Code	Package
SLE 4418 C	on request	Chip
SLE 4418 M2.2	on request	Wire-Bonded Module M2.2
SLE 4428 C	on request	Chip
SLE 4428 M2.2	on request	Wire-Bonded Module M2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



M2 (Card Contacts)

Pin Configurations (top view)

Pin Definitions and Functions

Pin	Card Contact	Symbol	Function
1	C1	VCC	Operating voltage 5 V
2	C2	RST	Chip control
3	C3	CLK	Clock
4	C4	N.C.	Not connected
5	C5	GND	Ground
6	C6	N.C.	Not connected
7	C7	I / O	Data line (open drain)
8	C8	N.C.	Not connected

SLE 4418/4428 comes out 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

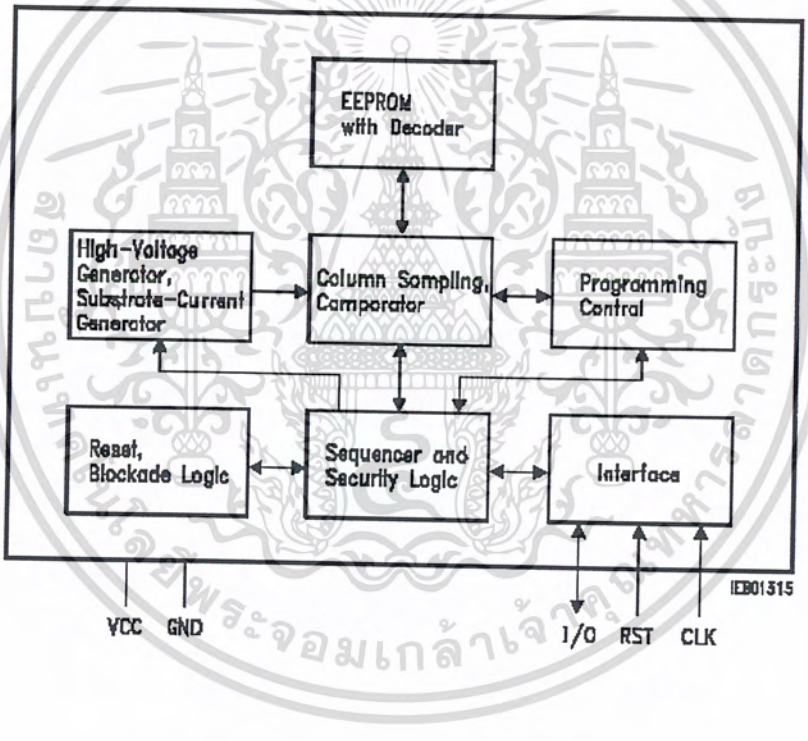


Figure 1
Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 Functional Description

1.1 SLE 4418

The chip contains an EEPROM organized 1024 x 8 bit offering the possibility of programmable write protection for each byte. Reading of the whole memory is always possible. The memory can be written and erased byte by byte. Input data and the contents of the adressed byte are compared so that only bits are written which were not written before. Erasing is only possible byte-wise, even if only one bit is to be erased, but bits may be written individually. Each byte can be write/erase-protected individually by setting a protect bit (EEPROM → ROM). The protect bit is only one time programmable and cannot be erased.

1.2 SLE 4428

Additionally to the above functions this version has a PSC verification logic. All the memory, except for the PSC, can always be read. The memory can be written or erased only after PSC verification. The error counter can always be written. After eight successive incorrect entries the error counter will block any subsequent attempt at PSC verification and hence any possibility to write and erase.

2 Reset and Answer-to-Reset

2.1 Reset

When the operating voltage is applied, the chip goes into the power-on reset (POR) state. POR is terminated by reset. Reset is started by RST changing from "0" to "1" and finished by CLK going from "0" to "1". This reset operation aborts any currently active command. After POR a read operation must be performed before any change of data is possible.

2.2 Answer to Reset (ISO 7816)

Answer to reset sets the address counter to "0" and the first data bit appears on the output. The contents of the following addresses can be read out with the following clock pulses. Answer to reset is executed by the following steps (**Fig. 2**) :

- RST goes from "0" to "1",
- clock pulse is applied,
- RST changes back from "1" to "0".

3 Commands

The state of RST defines the data direction on I/O.

RST	I/O
1	Command entry
0	Data output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

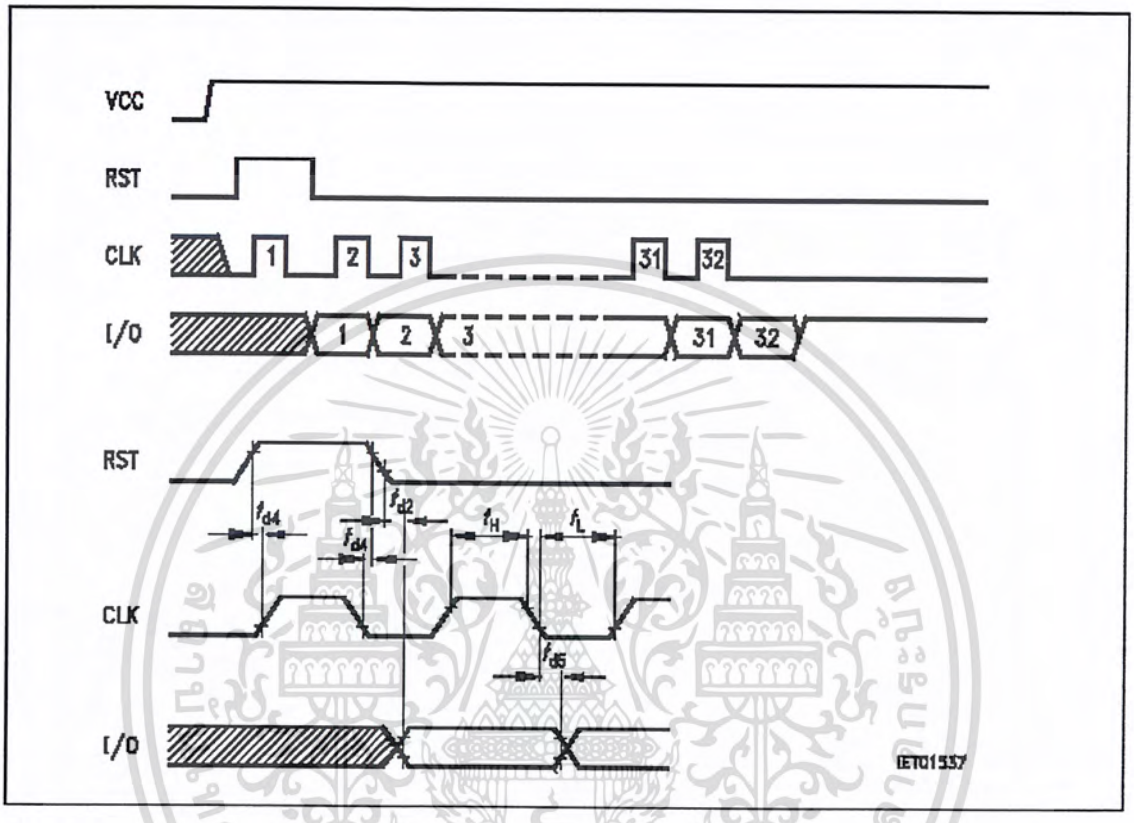


Figure 2
Reset and Answer-to-Reset

The possible commands are listed in **table 1.**

Table 1
Control Words for Command Entry

Byte 1								Byte 2	Byte 3	Operation
S0	S1	S2	S3	S4	S5	A8	A9	A0-A7	D0-D7	
1	0	0	0	1	1	Address bit		Address bit	Input data	Write and erase with protect bit
1	1	0	0	1	1				Input data	Write and erase without protect bit
0	0	0	0	1	1				Comparison data	Write protect bit with data comparison (verification)
0	0	1	1	0	0	8 and 9		0 – 7	No effect	Read 9 bits, data with protect bit
0	1	1	1	0	0				No effect	Read 8 bits, data without protect bit

Figure 3 shows the command entry, whereas the general timing is given in **figure 4.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

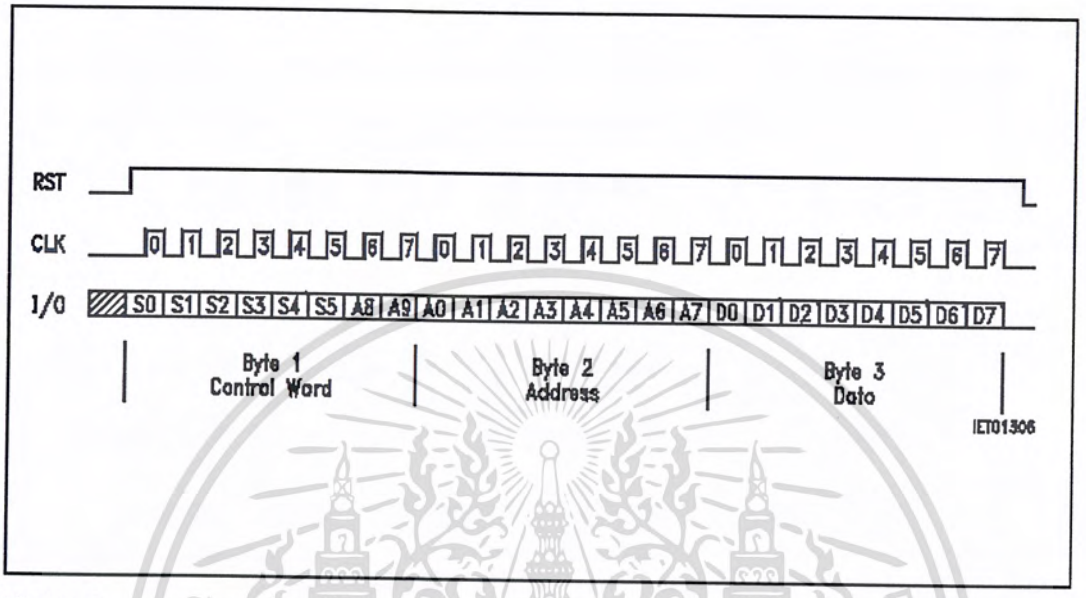


Figure 3
Command Entry

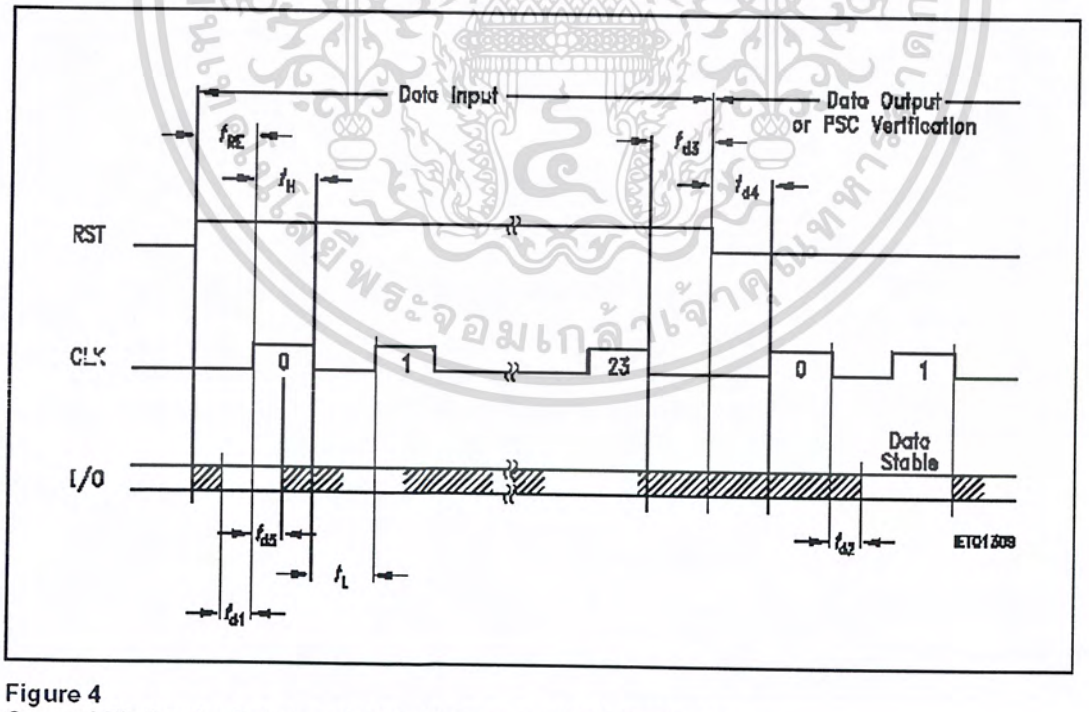


Figure 4
General Timing for Data Input, Data Output and PSC Verification

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 Write/Erase Operations

Write/Erase Data Byte without Protect Bit

Remark: Erase means "0" →"1", write means "1" → "0".

There are three kinds of write/erase operations which are automatically executed by the chip:

- Erase and subsequent write (duration 203 clock pulses, $f \leq 20$ kHz) **(Fig.5)**
- Write only: This procedure is suitable if single bits of one byte shall be changed only from "1" to "0". (duration 103 clock pulses, $f \leq 20$ kHz) **(Fig.6)**
- Erase only (= FF; duration 103 clock pulses, $f \leq 20$ kHz) **(Fig.6)**.

Write/Erase Data Byte with Protect Bit

The protect bit is erased at delivery, it can be written only once. Write procedure see above.

Write Protect Bit with Data Comparison

The data has to be entered a second time. The protect bit is only written if the old and the new data are identical.
The execution of write/erase commands are terminated internally after a given number of clock cycles. The end of the operation is indicated by the transition from "1" to "0" on I/O. Only RST transition from "0" to "1" sets I/O to state "1".

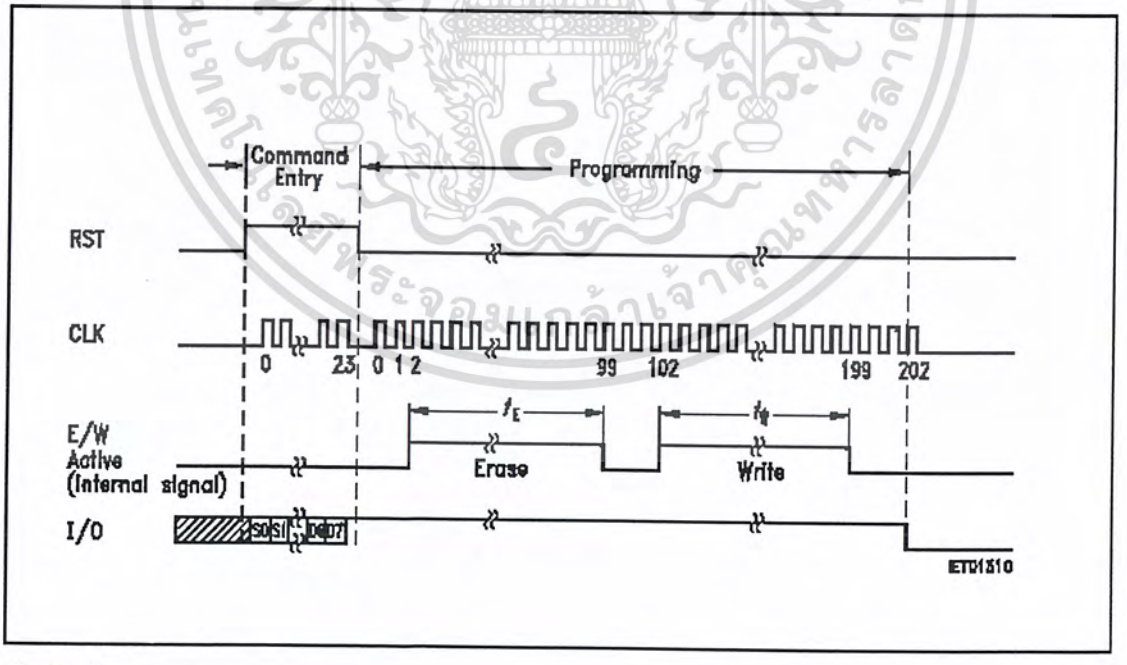


Figure 5
Programming: Erase and Write

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

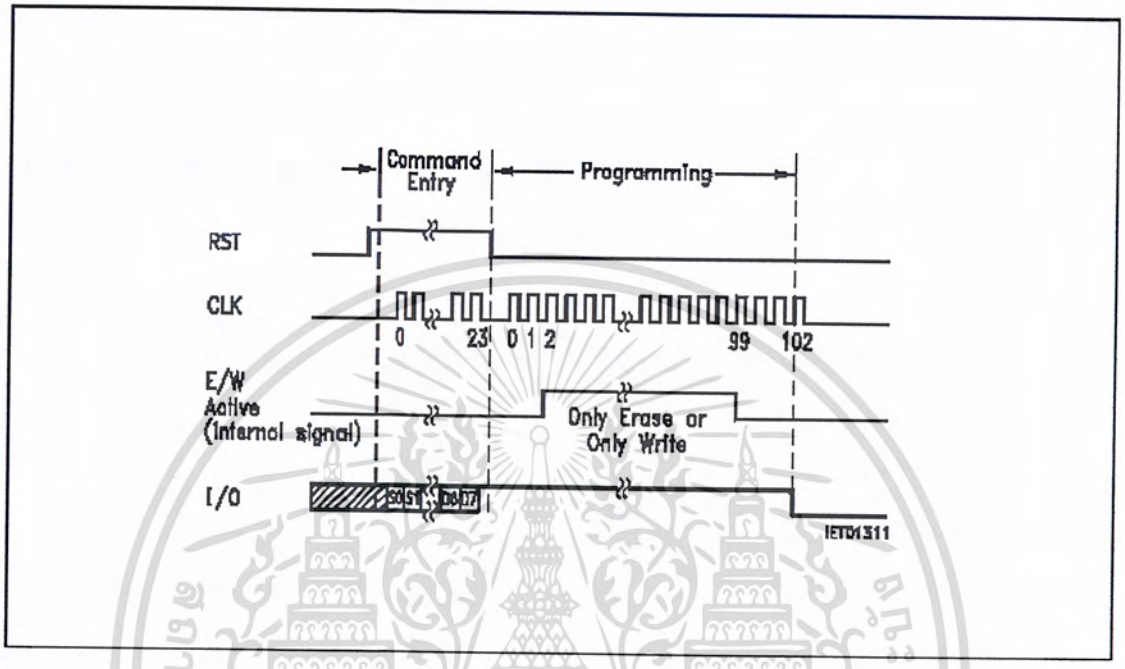


Figure 6
Programming: Only Erase or Only Write

3.2 Read Operations with Address Increment

Read Data Byte (Read 8 Bits) (Fig.7)

In this operation the output of the protect bit is suppressed. The address is incremented after the eighth clock pulse.

Read Data Byte together with Protect Bit (Read 9 Bits) (Fig. 8)

After command entry the eight data bits are read out by eight consecutive clock pulses, and the protect bit on the ninth pulse. After the ninth clock pulse the address counter is incremented.

The read operation is terminated by a RST transition "0" to "1".

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

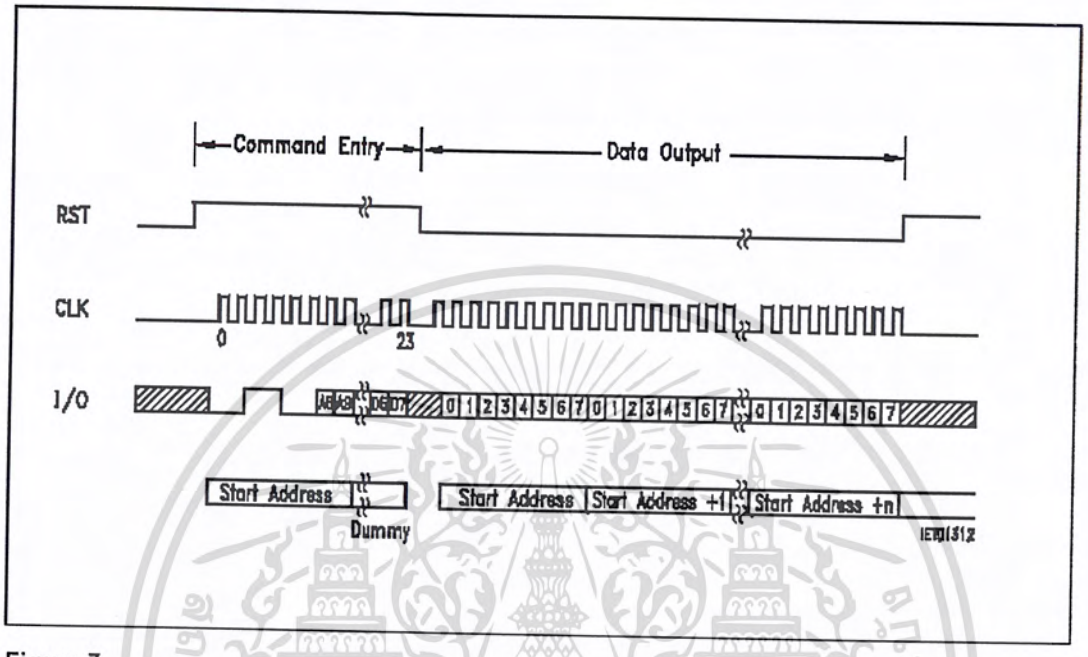


Figure 7
Read Data Byte: Read 8 Bits, Data D0-D7

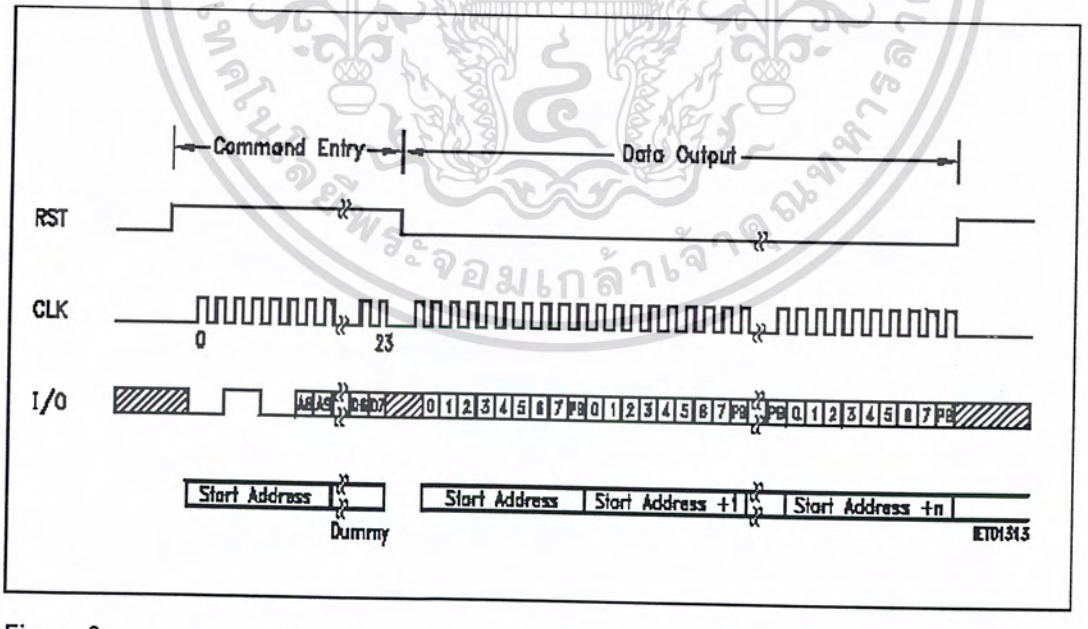


Figure 8
Read Data Byte together with Protect Bit: Read 9 Bits, Data D0-D7 and Protect Bit PB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4 Security Features SLE 4428

Extra Operation for User Identification (Fig. 9)

Without a PSC entry only reading is possible. The contents of the PSC addresses cannot be read out. If reading is attempted, "00" will appear. The verification procedure of the chip must be performed in the following sequence **(Table 1)**:

- write one not written error-counter bit, address "1021",
- enter first PSC-code byte, address "1022",
- enter second PSC-code byte, address "1023",
- after correct input the error counter can be erased.

After the PSC verification, I/O goes from "1" to "0". It is switched back to "1" by RST transition "0" to "1". The error counter is not automatically erased.

Writing Error Counter

Before PSC entry only writing of the error counter is possible. The number of erased bits of the error counter determines the number of possible attempts (max. 8). After successful access the error counter should be erased before disconnecting the supply voltage in order to reactivate the 8 attempts. Each error when entering the PSC requires the writing of a new counter bit.

Entry of PSC

The least-significant PSC byte beginning with the least significant bit must be entered first and then the most-significant one. If the internal data comparison proves correct, the EEPROM is enabled for erasing and writing as long as the operating voltage is applied. After enabling, the PSC may be altered as wished, except the corresponding protect bits are "0".

Condition when supplied

SLE 4428 is only supplied by the producer with a 2-byte PSC (transport code) which is agreed with the customer.

Table 1
Control Words for Command Entry, User Identification

Byte 1								Byte 2	Byte 3	Operation
S0	S1	S2	S3	S4	S5	A8	A9	A0-A7	D0-D7	
0	1	0	0	1	1	1	1	253	Bit mask	Write error counter
1	0	1	1	0	0	1	1	254	PSC byte 1	Verify 1st PSC byte
1	0	1	1	0	0	1	1	255	PSC byte 2	Verify 2nd PSC byte

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

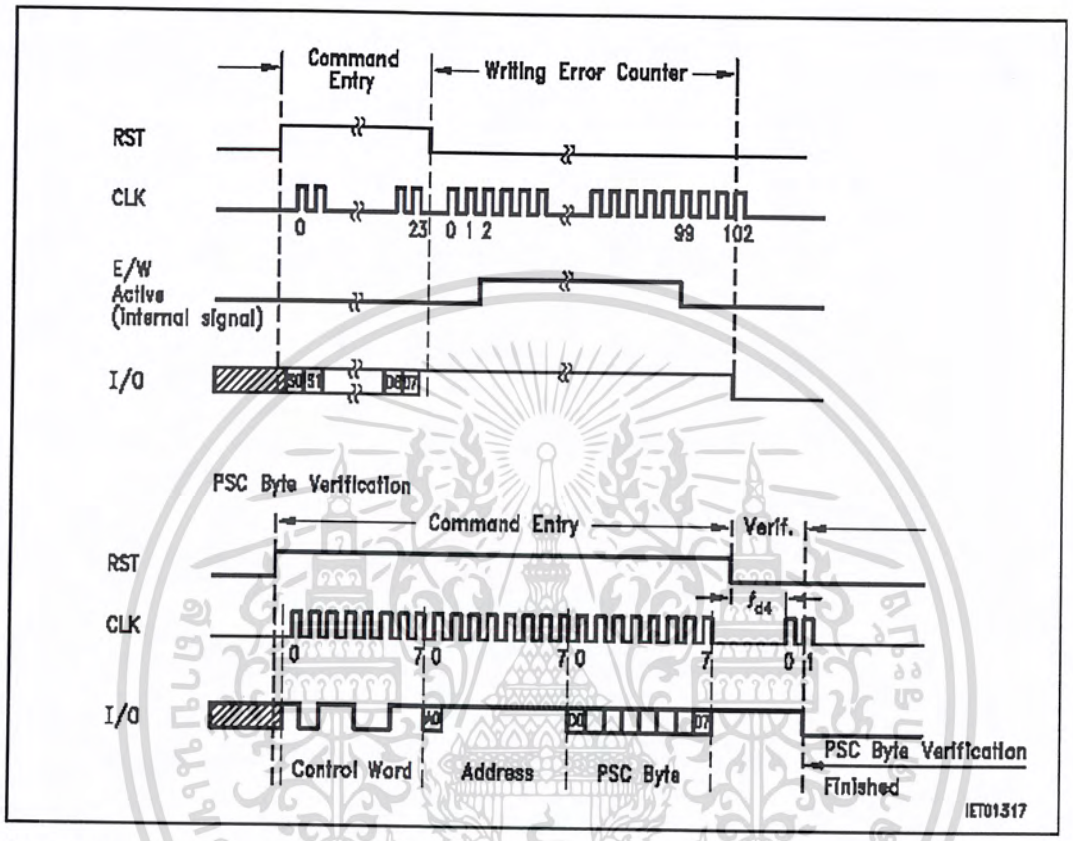


Figure 9
PSC Verification

5 Technical Data

Absolute Maximum Ratings

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
Supply voltage	V_{CC}	– 0.3		6	V
Input voltage	V_I	– 0.3		6	V
Storage temperature	T_{stg}	– 40		125	°C
Power dissipation	P_{tot}			60	mW

Operating Range

Supply voltage	V_{CC}	4.5		5.5	V
Ambient temperature	T_A	– 35		100	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC Characteristics

Parameter	Symbol	Limit Values			Unit
		min.	typ.	max.	
Supply					
Supply voltage	V_{CC}	4.5	5	5.5	V
Supply current	I_{CC}	–	3	10	mA
Data Input					
H input voltage (I/O, CLK, RST)	V_{IH}	3.5	–	V_{CC}	V
L input voltage (I/O, CLK, RST)	V_{IL}	0	–	0.8	V
H input current (I/O, CLK, RST)	I_{IH}	–	–	10	µA
Data Output					
L output current (V_L = 0.4 V, open drain)	I_{IL}	0.5	–	–	mA
H leakage current (V_H = V_{DD}, open drain)	I_{IH}	–	–	10	µA
Input capacitance	C_I	–	–	10	pF
Test pin	T	open or on V_{SS}			
Clock frequency	f_C	–	20	–	kHz
Pulse Duration					
Reset time	t_{RE}	9	–	–	µs
Answer to reset (RST)	t_{d6}	20	50	–	µs
CLK (count, H level)	t_H	10	–	–	µs
CLK (count, L level)	t_L	10	–	–	µs
Write time (f = 20 kHz)	t_W	5	–	–	ms
Erase time (f = 20 kHz)	t_E	5	–	–	ms

AC Characteristics

Setup time (D/CLK)	t_{d1}	4	–	–	µs
Setup time (CLK/RST)	t_{d3}	4	–	–	µs
Setup time (RST/CLK)	t_{d4}	4	–	–	µs
Hold time (D/CLK)	t_{d5}	4	–	–	µs
Delay time (CLK/D)	t_{d2}	6	–	–	µs
Rise time (I/O, CLK, RST)	t_R	–	–	1	µs
Fall time (I/O, CLK, RST)	t_F	–	–	1	µs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นายมงคล หมวดเพชร์
วัน เดือน ปีเกิด	20 พฤศจิกายน พ.ศ. 2507
ภูมิลำเนา	21 หมู่ที่ 1 ตำบลคลองทราย อำเภอนาทวี
	จังหวัดสงขลา 90160
	โทรศัพท์ 0-9970-7195

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนวัดนาหมอศรี
มัธยมศึกษาตอนต้น	โรงเรียนนาทวีวิทยาคม
ประกาศนียบัตรวิชาชีพ	วิทยาลัยเทคนิคหาดใหญ่
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคโนโลยีและอาชีวศึกษา วิทยาเขต เทคนิคภาคใต้ สงขลา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
คติพจน์	ทุกย่างก้าวที่พลาดไป มันทำให้เราเข้มแข็ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นายมานะชัย ทีกว้าง
วัน เดือน ปีเกิด	23 กรกฎาคม พ.ศ. 2524
ภูมิลำเนา	170 หมู่4 ตำบลแม่พูล อำเภอลับแล
	จังหวัดอุตรดิตถ์ 53130
	โทรศัพท์ 0-5545-7142

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเปี่ยมเมธีวิทยาลัย
มัธยมศึกษาตอนต้น	โรงเรียนอุตรดิตถ์
ประกาศนียบัตรวิชาชีพ	โรงเรียนอุตรดิตถ์เทคโนโลยี
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคอุตรดิตถ์
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
คติพจน์	ฝันไม่ต้องไกลแต่ต้องไปให้เกินฝัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นายสุรเดช สว่างวัน
วัน เดือน ปีเกิด	2 กันยายน พ.ศ. 2521
ภูมิลำเนา	146 หมู่5 ตำบล หัวนา อำเภอ เขมราฐ
	จังหวัด อุบลราชธานี 34170
	โทรศัพท์ 0-6615-6176

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนบ้านม่วงเฒ่า
มัธยมศึกษาตอนต้น	โรงเรียนพังเคนพิทยา
มัธยมศึกษาตอนปลาย	โรงเรียนเขมราฐพิทยาคม
ประกาศนียบัตรวิชาชีพชั้นสูง	วิทยาลัยเทคนิคร้อยเอ็ด
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์
	ภาควิชาครุศาสตร์วิศวกรรม
	คณะครุศาสตร์อุตสาหกรรม

ผลงานที่ได้รับรางวัล -

คติพจน์ ผลงานที่ดี ย่อมได้มาจากความยากลำบาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้แต่ง

ชื่อ-สกุล	นางสาวสุปราณี คอนเตาเหล็ก
วัน เดือน ปีเกิด	30 มกราคม พ.ศ. 2524
ภูมิลำเนา	1/184 หมู่ 1 ตำบล หนองไผ่ล้อม อำเภอ เมือง จังหวัด นครราชสีมา 30000

ประวัติการศึกษา

ประถมศึกษา	โรงเรียนเมืองนครราชสีมา
มัธยมศึกษาตอนต้น	โรงเรียนสุรธรรมพิทักษ์
มัธยมศึกษาตอนปลาย	โรงเรียนสุรธรรมพิทักษ์
ประกาศนียบัตรวิชาชีพชั้นสูง	สถาบันเทคโนโลยีราชมงคล วิทยาเขตภาคตะวันออกเฉียงเหนือ นครราชสีมา
ปริญญาตรี	สาขาวิชาอิเล็กทรอนิกส์และคอมพิวเตอร์ ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
ผลงานที่ได้รับรางวัล	-
คติพจน์	ผู้ที่ไม่พบปัญหาคือผู้ที่ไม่ทำอะไรเลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้