การนำสมาร์ตการ์ดมาใช้ในระบบ E-cash

E-cash Using Smart Card

นายพีรพล เนื่อยทอง รหัส 46362026

นายพีรศักดิ์ ศักดิ์สกุลเดช รหัส 46362034

นายเทอดพงษ์ หลีเกษม รหัส 46362240

I50785 20 ค.ศ

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 15 ต.ค. 2550
เลขทะเบียน 5000089
เลขเรียกหนังสือ ฝธ.
มหาวิทยาลัยนเรศวร ๓๗๑๑๐

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2549

หัวข้อโครงงาน	การนำสมาร์ตการ์ดมาใช้ในระบบ E-cash		
ผู้ดำเนินโครงงาน	นายพีรพล	เนือยทอง	รหัส 46362026
	นายพีรศักดิ์	ศักดิ์สกุลเดช	รหัส 46362034
	นายเทอดพงษ์	หลีเกษม	รหัส 46362240
อาจารย์ที่ปรึกษา	อาจารย์ภาณุพงศ์ สอนคม		
สาขาวิชา	วิศวกรรมคอมพิวเตอร์		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2549		

บทคัดย่อ

โครงงานนี้ได้ทำการออกแบบระบบ E-cash โดยใช้สมาร์ตการ์ด ซึ่งจะเน้นไปที่ต้นทุนการผลิตที่ต่ำ เนื่องจากใช้ ไมโครคอนโทรลเลอร์ MCS-51 แทนการใช้คอมพิวเตอร์ และใช้สมาร์ตการ์ดชนิด Memory ซึ่งระบบที่มีอยู่ในปัจจุบันจะใช้สมาร์ตการ์ดที่เป็น Com-Bi card ซึ่งมีต้นทุนที่สูงกว่า จากการทดลอง เครื่องอ่านและเขียนสมาร์ตการ์ดสามารถทำงานร่วมกับโปรแกรมวิชวลเบสิก และสมาร์ตการ์ดสามารถสื่อสารข้อมูลต่าง ๆ ที่ใช้ในระบบชำระเงินได้ เช่น จำนวนเงินในบัตร ค่าใช้จ่ายในแต่ละครั้ง จำนวนเงินคงเหลือ เป็นต้น

Project Title	E-cash Using Smart Card		
Name	Mr. Peerapon	Nuaytong	ID. 46362026
	Mr. Phirasak	Saksakundet	ID. 46362034
	Mr. Terdpong	Leekasem	ID. 46362240
Project Advisor	Mr. Panupong	Sornkom	
Major	Computer Engineering.		
Department	Electrical and Computer Engineering.		
Academic Year	2006		

ABSTRACT

This project have design an E-cash system by use a smartcard, we are significant to reduce a capital, because of we use the microcontroller MCS-51 instead of using the PC and using smartcard (type Memory) that well-know in nowadays instead of using the Com-bi card (smartcard) that have a higher capital. From the experiment, the writing and reading smartcard machine can work combination with the visual-basic program and the smartcard can communication in many way with the system that was use in the payment system such as, the value-remained in the card, the expense, the card value etc.,

ใบรับรองโครงงานวิศวกรรม

หัวข้อโครงงาน การนำสมาร์ตการ์ดมาใช้ในระบบ E-cash

ผู้ดำเนินโครงงาน

นายพีรพล	เนียมทอง	รหัส	46362026
นายพีรศักดิ์	ศักดิ์สกุลเดช	รหัส	46362034
นายเทอดพงษ์	หลีเกษม	รหัส	46362240

อาจารย์ที่ปรึกษา อาจารย์ภาณุพงศ์ สอนคม

สาขาวิชา วิศวกรรมคอมพิวเตอร์

ภาควิชา วิศวกรรมไฟฟ้าและคอมพิวเตอร์

ปีการศึกษา 2549

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ
(อาจารย์ภาณุพงศ์ สอนคม)

.....กรรมการ
(ดร.สุรเดช จิตประไพกุลศาล)

.....กรรมการ
(อาจารย์จิราพร พุกสุข)

กิตติกรรมประกาศ

ผู้จัดทำโครงงาน ขอขอบพระคุณอาจารย์ภาณุพงศ์ สอนคม ดร.สุรเดช จิตประไพกุลศาล และอาจารย์จิราพร พุกสุข เป็นอย่างสูงที่ได้สละเวลา ตลอดจนให้คำแนะนำทั้งทางทฤษฎีและปฏิบัติ ทั้งเอาใจใส่ให้ความช่วยเหลือเป็นอย่างดียิ่ง ตลอดระยะเวลาการทำโครงงาน และขอขอบคุณท่านอาจารย์ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ทุกท่าน ที่คอยดูแลและอำนวยความสะดวกในการศึกษาเล่าเรียน และการทำโครงงานครั้งนี้

นายพีรพล เนื่อยทอง
นายพีรศักดิ์ ศักดิ์สกุลเดช
นายเทอดพงษ์ หลีเกษม

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ก

บทคัดย่อภาษาอังกฤษ.....ข

กิตติกรรมประกาศ.....ค

สารบัญ.....ง

สารบัญตาราง.....ฉ

สารบัญรูป.....ช

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน.....1

1.2 วัตถุประสงค์ของโครงงาน.....2

1.3 ขอบข่ายของโครงงาน.....2

1.4 ขั้นตอนการดำเนินงาน.....2

1.5 แผนการดำเนินงาน.....3

1.6 ผลที่คาดว่าจะได้รับ.....4

1.7 งบประมาณ.....4

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ระบบ E-cash (เงินสดอิเล็กทรอนิกส์).....5

2.2 ข้อมูลที่เกี่ยวข้องกับสมาร์ตการ์ด.....6

2.3 เกี่ยวกับไมโครคอนโทรลเลอร์ MCS-51.....10

บทที่ 3 ขั้นตอนการดำเนินโครงงาน

3.1 ศึกษาเกี่ยวกับ E-cash, สมาร์ตการ์ด, และไมโครคอนโทรลเลอร์ MCS-51.....18

3.2 ออกแบบระบบ E-cash โดยใช้สมาร์ตการ์ด, ไมโครคอนโทรลเลอร์MCS-51 และ คอมพิวเตอร์.....19

3.3 เขียนโปรแกรมติดต่อกับสมาร์ตการ์ด, บอร์ด MCS-51 และ คอมพิวเตอร์.....28

3.4 เขียนโปรแกรมในส่วนที่เกี่ยวข้องกับการใช้งานระบบ E-cash.....29

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทำโครงงานปริญญานิพนธ์

4.1 โปรแกรมวิชวลเบสิกสำหรับเติมยอดเงินให้สมาร์ตการ์ด.....33

4.2 เครื่องอ่านและเขียนสมาร์ตการ์ด ณ ร้านขายสินค้าต่าง ๆ.....36

บทที่ 5 สรุปผลการทำปริญญานิพนธ์

5.1 สรุปผลการทำโครงงาน.....39

5.2 ปัญหาและข้อเสนอแนะ.....40

5.3 แนวทางการพัฒนาต่อ.....41

เอกสารอ้างอิง.....42

ภาคผนวก

ประวัติผู้เขียนโครงงาน

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน	3
2.1 จำนวนหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51	12
5.1 ข้อดีและข้อเสียของโครงงานเมื่อเปรียบเทียบกับระบบอื่น	39

สารบัญรูป

รูปที่	หน้า
2.1 รูปโครงสร้างทางกายภาพของสมาร์ตการ์ด	6
2.2 การแบ่งสมาร์ตการ์ดตามชนิดของหน่วยความจำและประเภทของหน้าสัมผัส	8
2.3 โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051	11
2.4 การจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51	13
2.5 ลักษณะของพอร์ตเมื่อไมโครคอนโทรลเลอร์ทำงานกับหน่วยความจำภายนอก	15
2.6 โครงสร้างภายในหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51	16
3.1 แสดงรูปแบบของระบบ E-cash โดยใช้สมาร์ตการ์ด	19
3.2 บล๊อคไดอะแกรมแสดงการทำงานของเครื่องอ่านและเขียนสมาร์ตการ์ดที่ร้านขายสินค้า	20
3.3 บล๊อคไดอะแกรมแสดงการทำงานของเครื่องอ่านและเขียนสมาร์ตการ์ดที่จะจำหน่ายบัตร	21
3.4 การจัดการหน่วยความจำของสมาร์ตการ์ดในการเก็บข้อมูล	27
4.1 แสดงการดูยอดเงินที่มีอยู่ในสมาร์ตการ์ด	33
4.2 แสดงการเปลี่ยนยอดเงินลงในสมาร์ตการ์ด	34
4.3 แสดงการเพิ่มยอดเงินเกินลงในสมาร์ตการ์ด	34
4.4 แสดงการลดยอดเงินเกินลงในสมาร์ตการ์ด	35
4.5 แสดงการตรวจสอบค่า PSC ของสมาร์ตการ์ด	36
4.6 เครื่องคิดเงิน ณ ร้านขายสินค้าต่างๆ	36
4.7 KEY PAD บนบอร์ดไมโครคอนโทรลเลอร์	37
4.8 Seven Segment แสดงคำว่า error เนื่องจากไม่ได้เสียบสมาร์ตการ์ด	38

บทที่ 1
บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

เงินสดอิเล็กทรอนิกส์ (E-cash) เป็นเทคโนโลยีที่ได้รับการพัฒนาขึ้นมาให้มีความปลอดภัยและเหมาะสมกับการชำระเงินที่มีมูลค่าต่ำ ๆ หรือที่เรียกกันว่า "การชำระเงินขนาดย่อม" ซึ่งจะสามารถรองรับการพาณิชย์อิเล็กทรอนิกส์ที่เกี่ยวข้องกับสินค้าหรือบริการที่มีมูลค่าต่ำ ซึ่งเงินอิเล็กทรอนิกส์ในที่นี้จะจำกัดอยู่เฉพาะเงินสดอิเล็กทรอนิกส์ ซึ่งก็หมายถึงการใช้เงินสดที่เป็นมูลค่านำมาเก็บข้อมูลในรูปแบบของดิจิตอลโดยผู้ถือข้อมูลนั้นได้ชำระเงินไว้ก่อนล่วงหน้าแล้ว (prepaid) และสามารถนำข้อมูลนั้นไปชำระเงินด้วยวิธีการต่าง ๆ ไม่ว่าจะเป็นการชำระเงิน ณ จุดขาย หรือเป็นการเปลี่ยนมือจากผู้หนึ่งไปยังอีกผู้หนึ่ง

เนื่องจากปัจจุบันมีการใช้เงินอิเล็กทรอนิกส์กันอย่างกว้างขวาง วงเงินที่หมุนเวียนมีจำนวนมากประชาชนผู้ใช้บริการมีมากขึ้นและมีผู้ประกอบการที่ออกบริการเกี่ยวกับเงินอิเล็กทรอนิกส์กันมากขึ้นด้วย โดยที่ผู้ใช้อาจจะไม่รู้เลยว่าเงินที่ใช้อยู่นั้นคือเงินอิเล็กทรอนิกส์ ตัวอย่างเช่น บัตรโทรศัพท์ บัตรพลาสติกที่ชำระอาหารตาม Food court ของศูนย์การค้าต่าง ๆ ตั๋วรถไฟฟ้า หรือแม้กระทั่งบัตรเครดิต เป็นต้น ซึ่งรายละเอียดของโครงงานนี้จะมีลักษณะคล้ายกับบัตรเครดิต แต่สำหรับผู้ที่ใช้ E-Cash ไม่จำเป็นต้องมีรายได้ที่สูงเหมือนกับบัตรเครดิต ผู้ที่ยังไม่มีรายได้ หรือผู้คนทั่วไปไม่ว่าจะมีรายได้มากน้อยเพียงใดก็สามารถใช้เงินอิเล็กทรอนิกส์ได้

เงินสดอิเล็กทรอนิกส์มีหลายรูปแบบซึ่งแต่ที่เราเห็นกันจนชินตาซึ่งส่วนหนึ่งก็คงจะเป็นอยู่ในรูปแบบของสมาร์ตการ์ด เนื่องจากบัตรสมาร์ตการ์ดเป็นบัตรใบเล็กๆ สามารถพกติดตัวได้อย่างสะดวก และเทคโนโลยีสมาร์ตการ์ดนี้มีการนำแผ่นไมโครชิป (Microprocessor) และแผงวงจรอิเล็กทรอนิกส์แบบวงจรรวม (Integrated Circuit) มาฝังลงบนตัวบัตรซึ่งจะมีกลไกการอ่านและการเขียนข้อมูลที่ซับซ้อนมากขึ้นสามารถเก็บรหัสลับและข้อมูลบางอย่างทางชีวภาพได้ ทำให้ยากแก่การปลอมแปลงบัตรและให้ความปลอดภัยของข้อมูลที่สูง และมีแนวโน้มที่จะนำมาใช้กันอย่างแพร่หลายในอนาคต เนื่องจากสมารถช่วยอำนวยความสะดวกในชีวิตประจำวันเราได้ โดยตอนนี้ต่างประเทศได้นำมาใช้กันอย่างแพร่หลายและประเทศไทยในอนาคตนั้นก็คงเป็นเช่นเดียวกันกับต่างประเทศ

1.2 วัตถุประสงค์ของโครงงาน

1. ศึกษาเกี่ยวกับสมาร์ตการ์ดเพื่อการนำไปประยุกต์ใช้งานในด้านต่าง ๆ
2. เพื่อพัฒนาระบบ E-cash โดยใช้สมาร์ตการ์ด
3. เพื่อศึกษาการหลักการเขียนโปรแกรมเพื่อติดต่อระหว่างคอมพิวเตอร์กับเครื่องอ่านสมาร์ตการ์ดรวมทั้ง Micro-controller ด้วย

1.3 ขอบข่ายของโครงงาน

1. เขียนโปรแกรมติดต่อระหว่างคอมพิวเตอร์กับเครื่องอ่านสมาร์ตการ์ดโดยให้สมาร์ตการ์ดสามารถใช้งานในระบบ E-cash ได้ โดยหลักการใช้งานจะคล้าย ๆ กับบัตรเครดิต แต่ระบบนี้ยอดเงินจะถูกเก็บอยู่ในหน่วยความจำของบัตรเลย ไม่ต้องไปยุ่งเกี่ยวกับธนาคารเหมือนบัตรเครดิต
2. สร้างบัตรสมาร์ตการ์ดที่สามารถใช้แทนเงินสดได้ โดยที่ยอดเงินมีมูลค่าไม่สูงมาก ซึ่งก่อนที่จะใช้บัตรสมาร์ตการ์ดจะต้องนำบัตรสมาร์ตการ์ดไปเติมเงินเสียก่อน

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษาและรวบรวมข้อมูลเกี่ยวกับหัวข้อโปรเจค
1.4.2 ศึกษาและค้นคว้าข้อมูลทั้งหมด
1.4.3 ทำการทดสอบการทำงานของระบบ
1.4.4 ตรวจสอบการนำไปใช้
1.4.5 จัดทำรูปเล่มรายงาน
1.4.6 ตรวจสอบและแก้ไขรูปเล่มรายงาน
1.4.7 จัดทำคู่มือประกอบการนำเสนอ
1.4.8 ออกแบบและพัฒนาระบบ

1.5 แผนการดำเนินงาน

กิจกรรม	ปี พ.ศ.2548		ปี พ.ศ.2549								
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.
ศึกษาหาข้อมูลเกี่ยวกับหัวข้อโปรเจค											
ศึกษาและค้นคว้าข้อมูลทั้งหมด											
ออกแบบและพัฒนาระบบ											
ทดสอบการทำงานของระบบ											
ตรวจสอบการนำไปใช้											
จัดทำรูปเล่มรายงาน											
ตรวจสอบและแก้ไขรูปเล่มรายงาน											
จัดทำคู่มือประกอบการนำเสนอ											

1.6 ผลที่คาดว่าจะได้รับ

1. บัตรสมาร์ตการ์ดที่สามารถใช้งานได้จริงในระบบ E-cash
2. ได้รับความรู้เกี่ยวกับการเขียนโปรแกรมไม่ว่าจะเป็น Visual Basic และภาษา C ที่ต้องใช้ในการติดต่อระหว่างเครื่องอ่านสมาร์ตการ์ดกับบอร์ดไมโครคอนโทรลเลอร์ ตระกูล MCS-51
3. สามารถนำระบบ E-cash ที่ใช้สมาร์ตการ์ดจากโครงงานนี้มาเป็นต้นแบบเพื่อที่จะนำไปประยุกต์และพัฒนาให้ดีขึ้นและสามารถนำไปใช้ได้จริงในชีวิตประจำวันต่อไป

1.7 งบประมาณ

1.7.1 ค่าถ่ายเอกสารและค่าเข้าเล่มโครงงาน	500	บาท
1.7.2 ค่าอุปกรณ์ Hardware	2,000	บาท
1.7.3 ค่าหนังสือที่ใช้ในการศึกษาค้นคว้า	500	บาท
	3,000	

(สามพันบาทถ้วน)

หมายเหตุ ถัวเฉลี่ยทุกรายการ

บทที่ 2
หลักการและทฤษฎีที่เกี่ยวข้อง

2.1 ระบบ E-CASH (เงินสดอิเล็กทรอนิกส์)

2.1.1 ความหมายของเงินสดอิเล็กทรอนิกส์

เงินสดอิเล็กทรอนิกส์ ย่อมาจาก Electronic Cash หรือเรียกสั้น ๆ ว่า E-cash เป็นรูปแบบการค้า การสื่อสาร และการทำธุรกรรมต่างๆ ในโลกปัจจุบัน โดยจะมีการนำเอาเทคโนโลยีสารสนเทศเข้ามาเป็นส่วนประกอบในการทำธุรกรรมต่างๆ และมีอิทธิพลมากยิ่งขึ้นทุกวัน เมื่อกล่าวถึงการขายสินค้าแน่นอนว่าจะต้องมีเรื่องเงินทองเข้ามาเกี่ยวข้องการจ่ายเงินในการซื้อสินค้าในรูปแบบทั่วๆ ไปที่มีกันอยู่ ไม่ว่าจะเป็นการจ่ายเงินสด การจ่ายด้วยบัตรเครดิตหรือบัตร ATM การจ่ายเงินด้วยเช็ค การโอนเงินเข้าบัญชี และอีกหลายวิธี บางวิธีก็สามารถนำมาใช้ได้ทางอินเทอร์เน็ต ในขณะที่บางวิธีก็ใช้ไม่ได้กับเครือข่ายอินเทอร์เน็ต

ในปัจจุบันได้มีการพัฒนาและทดลองใช้ระบบเงินใหม่ ๆ หลายระบบ โดยอาจจัดกลุ่มของระบบการชำระเงินอิเล็กทรอนิกส์ใหญ่ ๆ ที่ได้รับการพัฒนาขึ้นมาได้เป็น 2 กลุ่มใหญ่ ๆ คือ กลุ่มเงินสดอิเล็กทรอนิกส์และกลุ่มระบบการชำระเงินอิเล็กทรอนิกส์

กลุ่มเงินสดอิเล็กทรอนิกส์ ยังสามารถแบ่งออกเป็นเงินสดอิเล็กทรอนิกส์ ที่จัดเก็บอยู่ในบัตรที่บรรจุวงจรรวม ซึ่งเรียกว่า "สมาร์ตการ์ด" (Smart Card) ซึ่งสามารถเติมเงินได้ โดยใช้ได้ในแบบออฟไลน์ (off-line) คือสามารถโอนเงินระหว่างสมาร์ตการ์ดกับอุปกรณ์พิเศษ

เงินสดอิเล็กทรอนิกส์ เป็นเทคโนโลยีในการเก็บข้อมูลในรูปแบบดิจิตอล สามารถเป็นตัวแทนของมูลค่า (stored value) ซึ่งผู้ถือข้อมูลนั้น ได้ชำระไว้แล้วล่วงหน้า (prepaid) โดยที่ข้อมูลนั้นสามารถนำไปใช้ชำระเงินด้วยวิธีต่างๆ ไม่ว่าจะเป็นการชำระค่าสินค้าหรือบริการ ณ จุดขาย (point of sale) หรือเป็นการเปลี่ยนมือจากผู้หนึ่งไปยังอีกผู้หนึ่งโดยผ่านอุปกรณ์บางอย่าง (direct transfer) หรือผ่านเครือข่ายอินเทอร์เน็ต เป็นต้น แน่นอนว่าจะต้องมีผู้รับรองค่าของเงินสดดิจิตอลเหล่านี้ ตัวอย่างบริษัทที่สร้างเทคโนโลยีเงินสดดิจิตอล เช่น บริษัท DigiCash และบริษัท CyberCash ได้มีการร่วมมือกับธนาคารและผู้ขายสินค้า เพื่อให้ยอมรับการใช้เงินสดลักษณะนี้

2.1.2 ปัญหาของเงินสดอิเล็กทรอนิกส์ในปัจจุบัน

ปัญหาสำคัญก็คือ การไม่มีมาตรฐานที่เข้ากันได้ระหว่างเทคโนโลยีเงินสดดิจิตอลต่างๆ ซึ่งผู้ใช้เงินสดดิจิตอลของค่ายหนึ่ง ยังไม่สามารถที่จะแลกเปลี่ยนกับเงินสดค่ายอื่นได้อย่างอิสระ นอกจากนี้ก็ยังมีปัญหาที่ผู้บริโภคจะไม่สามารถที่จะแลกเปลี่ยนกับเงินสดค่ายอื่นได้อย่างอิสระ นอกจากนี้ก็ยังมีปัญหาที่ผู้บริโภคจะต้องใช้ในการทำความเข้าใจ และให้ความไว้วางใจกับ

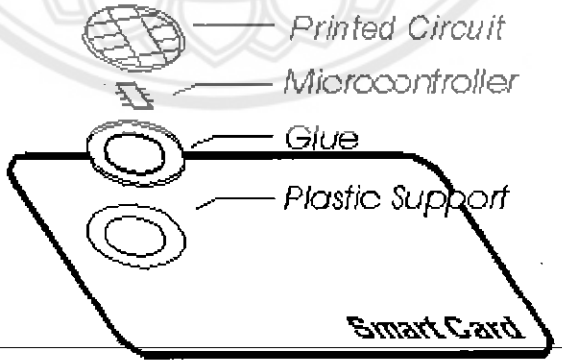
เทคโนโลยีใหม่ได้อย่างเงินสดดิจิตอลอีกด้วย ในอนาคตปัญหาเรื่องมาตรฐานอาจได้รับการแก้ไขในที่สุด

ในอนาคตอาจจะไม่มีการพกกระเป๋าสตางค์กันอีกต่อไป ไม่มีการใช้เงินสด ไม่มีบัตรเครดิต ไม่มีบัตรประจำตัวประชาชนและใบขับขี่อีกต่อไป ข้อมูลทางกายภาพอาจจะถูกเก็บไว้ที่ส่วนกลาง (อาจจะเป็นภาครัฐ) และข้อมูลบางส่วนก็ถูกเก็บไว้ที่บริษัทที่ให้บริการต่างๆ เช่น ข้อมูลทางการเงินก็จะถูกเก็บไว้ที่ธนาคาร ข้อมูลทางสุขภาพก็ถูกจัดเก็บไว้ที่โรงพยาบาล เป็นต้น เวลาซื้อสินค้าหรือบริการ ก็จะใช้ลักษณะทางกายภาพ เช่น ม่านตา หรือลายนิ้วมือ ในการชำระค่าสินค้าและบริการ

2.2 ข้อมูลที่เกี่ยวข้องกับสมาร์ตการ์ด

2.2.1 ความหมายของสมาร์ตการ์ด

สมาร์ตการ์ด คือ บัตรพลาสติกที่มี ชิป IC (Intergrated circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูล และประมวลผลภายในตัวเองโดยวิธีการเข้ารหัสตามมารตาฐาน DES Algorithm (Data Encryption Standard) เพื่อให้ระบบมีความปลอดภัยสูงขึ้น ด้วยคุณสมบัติประการหนึ่งที่ทำให้สมาร์ตการ์ดมีความแตกต่างจากบัตรพลาสติกทั่วไปก็คือ ขณะทำรายการสมาร์ตการ์ดสามารถทำรายการได้ด้วยตัวเองโดยไม่ต้องอาศัยติดต่อสื่อสารกับระบบหลัก นั่นก็คือสมาร์ตการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก และทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก



รูปที่ 2.1 รูปโครงสร้างทางกายภาพของสมาร์ตการ์ด

ในการประมวลผล และจัดเก็บข้อมูล สมาร์ตการ์ดสามารถทำได้อย่างรวดเร็วกว่าสื่อสำหรับจัดเก็บชนิดอื่น ๆ ด้วยขนาดที่เท่ากับบัตรแถบแม่เหล็ก ทำให้สะดวกในการจัดเก็บ และพกพา นอกจากนี้สมาร์ตการ์ดยังมีคุณสมบัติด้านความทนทาน ไม่ว่าจะเป็นรังสีชนิดต่าง ๆ

(ในปริมาณที่ไม่เป็นอันตรายต่อมนุษย์) สนามแม่เหล็ก ไฟฟ้าสถิต ความชื้น ความร้อน การบิดงอ ฯลฯ ก็ไม่สามารถทำให้สมาร์ตการ์ดเสียหายได้โดยง่าย จึงทำให้สมาร์ตการ์ดเทียบเท่าบัตรในอุดมคติเลยทีเดียว ในต่างประเทศก็ได้มีการใช้งานสมาร์ตการ์ดกันอย่างแพร่หลายจนเป็นไปได้ว่าสมาร์ตการ์ดกำลังเป็นบัตรชนิดใหม่ที่จะเข้ามาแทนที่บัตรแถบแม่เหล็กที่ใช้กันอยู่ในปัจจุบัน

2.2.2 องค์ประกอบในการใช้งานสมาร์ตการ์ด

ก. ตัวบัตรและตัวชิป

บัตรและชิปสมาร์ตการ์ดเป็นส่วนแรกที่มักกล่าวถึงกัน เพราะสมาร์ตการ์ดมีหลายรูปแบบ หลากหลายการใช้งาน โดยหลักการแล้วสมาร์ตการ์ดเป็นเพียงบัตรพลาสติกฝังชิป IC ที่สามารถเก็บข้อมูล และประมวลผลข้อมูลได้เท่านั้น ผู้ออกแบบระบบมีหน้าที่นำสมาร์ตการ์ดมาใช้งานอย่างชาญฉลาดและเหมาะสมกับประเภทของงาน และบริหารข้อมูลภายในสมาร์ตการ์ดให้เกิดความปลอดภัยสูงสุด โดยในปัจจุบันเราสามารถเห็นการใช้งานสมาร์ตการ์ดในหลาย ๆ รูปแบบ เช่น บัตรโทรศัพท์ ซิมการ์ดในโทรศัพท์มือถือ บัตรเข้าออกที่อยู่อาศัย บัตรนักศึกษา บัตรพนักงาน บัตรแทนเงินสด ฯลฯ สมาร์ตการ์ดที่นำมาใช้งานจะมีความแตกต่างกันในเรื่องของประเภทการใช้งาน ที่เห็นได้ชัดที่สุดก็คงเป็นซิมการ์ดในโทรศัพท์มือถือ ซึ่งมีการกำหนดเป็นมาตรฐาน GSM โดยผู้ผลิตสมาร์ตการ์ดสำหรับใช้เป็นซิมการ์ด ต้องผลิตสมาร์ตการ์ดที่มีโครงสร้างข้อมูลภายในตามมาตรฐาน GSM กำหนดไว้

ข. สมาร์ตการ์ดรีดเดอร์

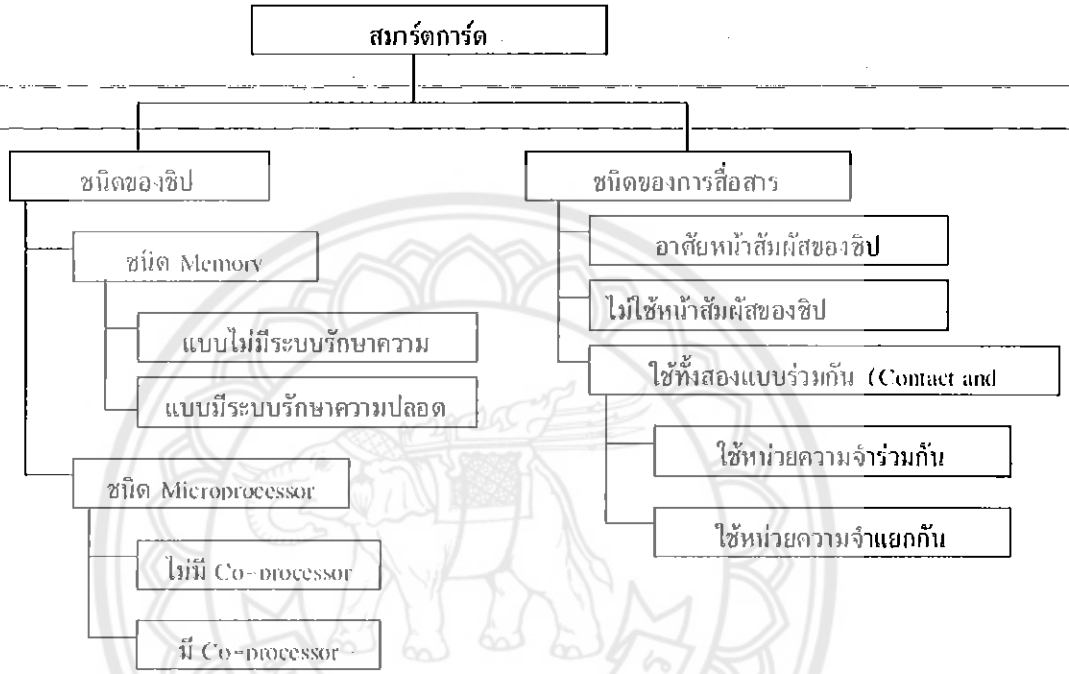
สำหรับสมาร์ตการ์ดแล้วจำเป็นต้องมีอุปกรณ์สำหรับเชื่อมต่อกับซิมสมาร์ตการ์ดโดยเฉพาะซึ่งเรียกว่า สมาร์ตการ์ดรีดเดอร์ โดยภายในสมาร์ตการ์ดรีดเดอร์จะประกอบด้วยขาสำหรับเชื่อมสัญญาณกับหน้าสัมผัสบนชิปสมาร์ตการ์ด (Card Contact) หรือเป็นเสาอากาศสำหรับส่งคลื่นวิทยุ สำหรับสมาร์ตการ์ดแบบไม่มีหน้าสัมผัส (Contactless) และหน่วยประมวลผลพร้อมหน่วยความจำสำหรับติดต่อสื่อสารกับชิปสมาร์ตการ์ดโดยตรง การสร้างสมาร์ตการ์ดรีดเดอร์ขึ้นใช้เอง ก็สามารถทำได้โดยนำไมโครโปรเซสเซอร์ ไมโครคอนโทรลเลอร์ เบอร์ใดก็ได้มาประยุกต์ใช้ในการเชื่อมต่อกับสมาร์ตการ์ด ปัจจุบันได้มีผู้ผลิตสมาร์ตการ์ดรีดเดอร์ออกมาขายในท้องตลาดหลายรูปแบบซึ่งมีราคาย่อมเยา และใช้งานได้ง่ายกว่าการสร้างสมาร์ตการ์ดรีดเดอร์เองเสียอีก

ค. ซอฟต์แวร์

ในการใช้งานสมาร์ตการ์ดนอกจากบัตรสมาร์ตการ์ด และสมาร์ตการ์ดรีดเดอร์แล้วยังต้องมีส่วนประกอบอีกส่วนหนึ่งที่สำคัญไม่ยิ่งหย่อนไปกว่ากัน นั่นก็คือ ซอฟต์แวร์ สำหรับจัดการข้อมูลในสมาร์ตการ์ด

2.2.3 ชนิดของสมาร์ตการ์ด

การแบ่งชนิดของสมาร์ตการ์ดในปัจจุบันอาจทำได้ยากสักหน่อย เนื่องจากมีการใส่เทคโนโลยีใหม่ ๆ ลงสมาร์ตการ์ดตลอดเวลา ถ้าจะแบ่งตามชนิดของหน่วยความจำภายในอาจจะไม่ชัดเจนนัก ยิ่งแบ่งตามลักษณะการเชื่อมต่อ ก็คงจะไม่ครอบคลุมสมาร์ตการ์ดทั้งหมด ดังนั้น จึงแสดงการแบ่งชนิดของสมาร์ตการ์ดให้เข้าใจง่าย ๆ ดังรูปด้านล่าง



รูปที่ 2.2 การแบ่งสมาร์ตการ์ดตามชนิดของหน่วยความจำและประเภทของหน้าสัมผัส

ในการแบ่งสมาร์ตการ์ดออกเป็นสองชนิดตามชนิดของวงจรภายในดังที่กล่าวมา อาจแบ่งได้อีกลักษณะ คือ แบ่งตามความถี่ในการรับ-ส่งข้อมูลผ่านหน้าสัมผัส I/O ของสมาร์ตการ์ดซึ่งสามารถแบ่งได้ดังนี้

Memory card (Synchronous) เนื่องจากสมาร์ตการ์ดชนิดนี้มีการรับ-ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิป (ข้อมูลแต่ละบิตที่ส่งให้แก่ชิปต้องสัมพันธ์กับสัญญาณนาฬิกา) สมาร์ตการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วย ส่วนของการวงจรสำหรับติดต่อสื่อสารภายนอก หน่วยความจำข้อมูล และหน่วยความจำสำหรับเก็บชุดคำสั่งของสมาร์ตการ์ด

Processor card (Asynchronous card) สมาร์ตการ์ดชนิดนี้เป็นสมาร์ตการ์ดที่ได้รับการปรับปรุงจากสมาร์ตการ์ดชนิด Memory ด้วยการใส่เทคโนโลยีเข้าไปในชิป เพื่อให้ชิปสามารถประมวลผลข้อมูลและเพิ่มความปลอดภัยให้แก่ข้อมูลให้สูงขึ้น การที่ใส่ไมโครโปรเซสเซอร์ลงในชิปทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำสำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผลข้อมูล นอกจากนี้ยังมีการ

ใส่ชิปประมวลผลทางคณิตศาสตร์ลงในชิปสมาร์ตการ์ดเพื่อช่วยในการประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้ารหัส ทำให้สมาร์ตการ์ดชนิด Processor มีความเร็วในการทำงานสูงกว่าสมาร์ตการ์ดชนิด Memory หลายเท่า

Contactless card สมาร์ตการ์ดชนิดนี้เป็นสมาร์ตการ์ดที่ล้ำสมัยที่สุดในปัจจุบัน ด้วยเทคโนโลยีการสื่อสารผ่านคลื่นวิทยุ โดยการส่งคลื่นวิทยุความถี่ 13.56 MHz ที่ได้รับการมอดูเลตข้อมูล และส่งให้กับชิปสมาร์ตการ์ด ทางด้านชิปสมาร์ตการ์ดจะใช้ขดลวดเป็นเสารับ-ส่งสัญญาณ โดยเสา-รับสัญญาณนี้จะเป็นขดลวดขนาดเล็กที่ถูกฝังลงในเนื้อบัตร ภายนอกบัตรชนิดนี้แทบดูไม่ออกว่าเป็นบัตรสมาร์ตการ์ด ด้วยเหตุนี้เราจึงพบสมาร์ตการ์ดชนิดนี้มักอยู่ในรูปร่างแปลก ๆ เสมอ

นอกจากสมาร์ตการ์ดแบบ Contactless ที่ใช้การสื่อสารด้วยคลื่นวิทยุเพียงอย่างเดียวแล้ว ยังมีการรวมเอาสมาร์ตการ์ดชนิด Contact และ Contactless บนบัตรใบเดียวกัน ซึ่งเป็นสมาร์ตการ์ดที่รวมเอาสมาร์ตการ์ดแบบที่มีหน้าสัมผัส กับสมาร์ตการ์ดแบบไม่มีหน้าสัมผัสเข้าด้วยกัน เพื่อความสะดวก และเพิ่มความเร็วในการใช้งานอีกด้วย สมาร์ตการ์ดชนิดนี้สามารถแบ่งออกได้อีก 2 ประเภท คือ

Com–Bi card สมาร์ตการ์ดชนิดนี้เป็นการรวมเอาสมาร์ตการ์ดแบบ Contact และ Contactless เข้าด้วยกันโดยใช้หน่วยความจำข้อมูลร่วมกันเพื่อให้การทำรายการที่จำเป็นต้องอยู่ภายใต้ระบบรักษาความปลอดภัย สามารถทำงานได้โดยผ่านทางหน้าสัมผัสที่มีไมโครโปรเซสเซอร์ควบคุมอยู่ และสามารถใช้งานทั่วไปได้อย่างสะดวกสบาย (speed pass) ผ่านทางคลื่นวิทยุ

Hybrid card สมาร์ตการ์ดชนิดนี้มีลักษณะภายในคล้ายกับประเภท Com-bi card แต่จะต่างกันในเรื่องของหน่วยความจำข้อมูล โดยหน่วยความจำข้อมูลจะถูกแยกจากกันอย่างสิ้นเชิงระหว่าง Contact และ Contactless เพื่อความสะดวกในการแยกใช้งาน ซึ่งปัจจุบัน Hybrid card จะมีความหมายรวมถึงบัตรที่มีคุณภาพในการใช้งานตั้งแต่สองอย่างขึ้นไป เช่น สมาร์ตการ์ดที่มีทั้งชิปสมาร์ตการ์ดและแถบแม่เหล็ก, บัตรสมาร์ตการ์ดที่เป็นทั้ง Contact และ Contactless

2.2.4 การบริหารหน่วยความจำข้อมูลในสมาร์ตการ์ด

หน่วยความจำข้อมูลภายในชิปสมาร์ตการ์ดเป็นหน่วยความจำชนิด EEPROM ซึ่งสามารถเขียนข้อมูลใหม่ด้วยกระแสไฟฟ้า หน่วยความจำชนิดนี้มีลักษณะเป็นผืนเดียวยาวต่อเนื่องกันตั้งแต่ข้อมูลที่ 0 จนถึงตำแหน่งสุดท้าย จะมีเพียงสมาร์ตการ์ดชนิด Memory เท่านั้นที่มีการจัดการหน่วยความจำแบบต่อเนื่อง แต่สมาร์ตการ์ดชนิด Processor จะมีการจัดการหน่วยความจำที่ต่างออกไป

การจัดการหน่วยความจำข้อมูลในสมาร์ตการ์ดชนิด Memory มีการจัดการเหมือนกับหน่วยความจำทั่วไป หน่วยความจำในสมาร์ตการ์ดชนิดนี้ อาจมีความแตกต่างกันในเรื่องของพื้นที่ที่สงวนในหน่วยความจำตามแต่ผู้ผลิตจะกำหนด ซึ่งพื้นที่หน่วยความจำที่สงวนไว้นี้มีด้วยกันหลาย

จุดประสงค์ด้วยกัน เช่น พื้นที่สำหรับจัดเก็บรายละเอียดของบัตร (ชนิดของสมาร์ตการ์ด, รหัสประจำตัวของบัตร, เลยรหัสโซนที่ผลิตและใช้งาน, รหัสสำหรับเข้าถึงข้อมูลในบัตร, ฯลฯ) โดยพื้นที่สงวนบางส่วนสามารถเข้าไปแก้ไขข้อมูลได้ (อาจทำให้สมาร์ตการ์ดใบนั้นไม่สามารถทำงานข้ามระบบได้) บางส่วนไม่อนุญาตให้สามารถแก้ไขได้ บางส่วนต้องใช้คำสั่งเฉพาะของชิปนั้น ๆ ในการอ่าน-เขียนข้อมูล ถึงอย่างไรก็ตามหน่วยความจำของสมาร์ตการ์ด ชนิด Memory จะมีลักษณะเป็นผืนเดียวยาวต่อเนื่องกันเหมือนกันทั้งหมด

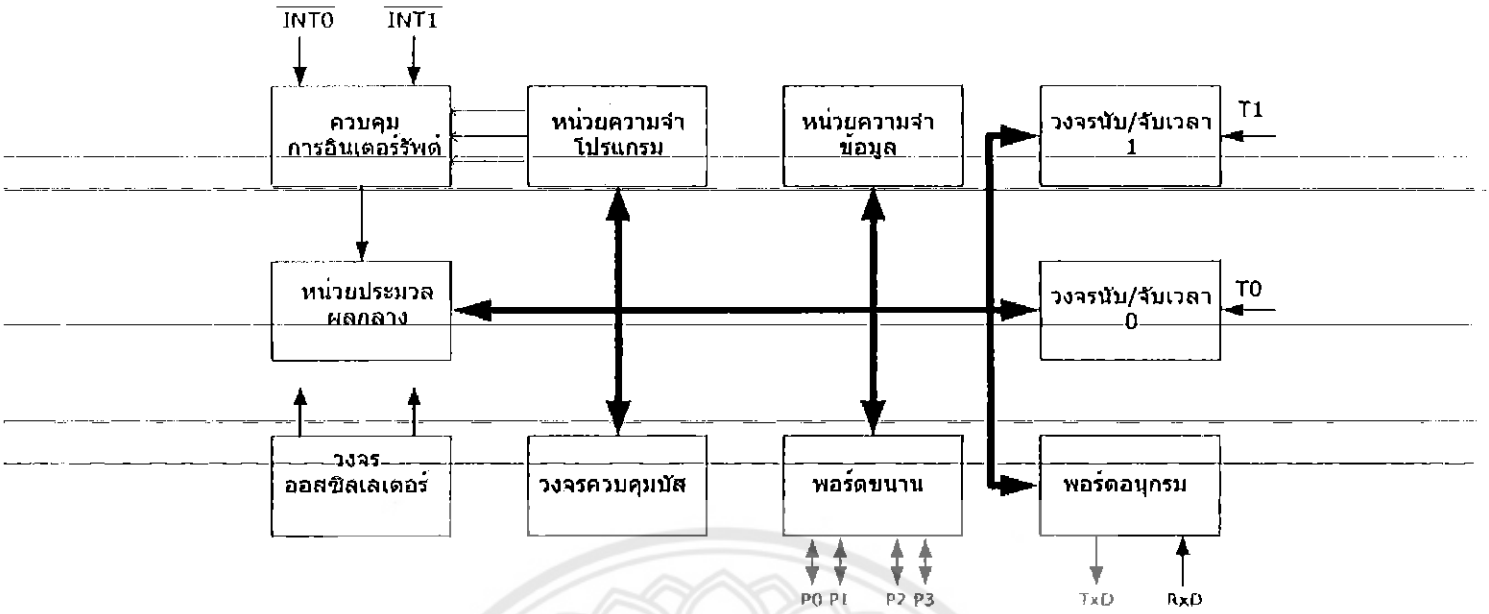
สำหรับสมาร์ตการ์ดชนิด Processor จะมีลักษณะการจัดการหน่วยความจำข้อมูลแตกต่างไปจากสมาร์ตการ์ดชนิด Memory อย่างสิ้นเชิง ด้วยเหตุผลที่สมาร์ตการ์ดชนิดนี้ไม่อนุญาตให้เข้าถึงข้อมูลโดยตรง เพื่อให้สามารถกำหนดข้อมูลเป็นหมวดหมู่ และสามารถกำหนดสิทธิ์ในการเข้าถึงข้อมูลแต่ละส่วนได้ง่ายกว่าต้องการกำหนดสิทธิ์ที่ละไบต์ จึงต้องให้ไมโครโปรเซสเซอร์ในชิปเป็นผู้จองพื้นที่ในหน่วยความจำข้อมูลตามคำสั่งที่เราส่งให้ เช่นเดียวกับในระบบปฏิบัติการของเครื่องคอมพิวเตอร์มีการจัดเก็บข้อมูลด้วยไฟล์ การจัดเก็บข้อมูลในสมาร์ตการ์ดชนิด Processor ก็มีการจัดเก็บข้อมูลเป็นไฟล์เช่นเดียวกัน

2.3 เกี่ยวกับไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ที่มีอุปกรณ์สนับสนุนประกอบอยู่ภายใน ได้แก่ หน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บตัวโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับ-ส่งข้อมูลแบบอนุกรม ซึ่งทำให้การใช้งานง่ายและมีประสิทธิภาพมากขึ้น โดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกมาเพิ่มเติมมากเหมือนกับไมโครโปรเซสเซอร์ทั่วไป นอกจากนี้หากต้องการใช้งานไมโครคอนโทรลเลอร์ร่วมกับอุปกรณ์อื่นเพิ่มเติม เช่น ไมโครโปรเซสเซอร์ 8255 หรือหน่วยความจำภายนอกเป็นต้น

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051 แสดงดังรูปที่ 2.3 ประกอบด้วยส่วนต่างๆ ดังต่อไปนี้

- หน่วยความจำสำหรับเก็บข้อมูลจำนวน 128 ไบต์
- หน่วยความจำภายในสำหรับเก็บโปรแกรมขนาด 4 กิโลไบต์
- อุปกรณ์ควบคุมการขัดจังหวะ (Interrupt)
- ตัวตั้งเวลาและตัวนับขนาด 16 บิต 2 ชุด
- พอร์ตควบคุมการสื่อสารอนุกรมดูเพลกซ์เต็มซึ่งสามารถรับส่งข้อมูลได้พร้อมกัน
- พอร์ตขนานสำหรับติดต่ออุปกรณ์ภายนอกได้ 4 พอร์ตๆ ละ 8 บิต
- วงจรผลิตสัญญาณนาฬิกาภายใน



รูปที่ 2.3 โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8051

ในปัจจุบันมีไมโครคอนโทรลเลอร์ MCS-51 ซึ่งเป็นไมโครคอมพิวเตอร์แบบชิพเดี่ยวไม่ต้องต่ออุปกรณ์ภายนอกก็สามารถทำงานได้ มีความสะดวกในการใช้งาน และเขียนโปรแกรมควบคุมด้วยภาษาเบสิกได้ โดยไม่ต้องศึกษาการทำงานของวงจรเหมือนกับภาษาแอสเซมบลี นอกจากนั้นยังเป็นอุปกรณ์ที่ออกแบบมาสนองความต้องการของผู้ใช้คือ มีอินพุตและเอาต์พุตภายในตัวเอง พอร์ตของอินพุตและการเชื่อมต่อเอาต์พุตจากบัฟเฟอร์และสายควบคุมอื่นๆ ที่ใช้สำหรับแยกข้อมูลกับตำแหน่งที่อยู่และชุดคำสั่งเพิ่มขึ้นเป็นพิเศษและเพื่อจัดการข้อมูลด้วยวงจรตั้งเวลากับวงจรนับด้วย ปกติวงจรนับจะสามารถทำงานเป็นวงจรตั้งเวลาได้ด้วย จึงเรียกควบคู่กันไปคือวงจรตั้งเวลา/วงจรนับ

จากตารางที่ 2.7 เป็นรายการของไมโครคอนโทรลเลอร์ MCS-51 ซึ่งแสดงถึงจำนวนหน่วยความจำของวงจรตั้งเวลา/วงจรนับและระดับของการขอขัดจังหวะ

ไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยคุณสมบัติดังนี้

- หน่วยประมวลผลกลาง 8 บิตที่ควบคุมได้ง่าย
- เพิ่มการทำงานลอจิกครั้งละ 1 บิตได้
- สายอินพุตและเอาต์พุตมีจำนวน 32 เส้นใช้เลือกตำแหน่งแยกต่างหากจากกันได้
- มีแรมบรรจุไว้ภายในขนาด 128 ไบต์หรือ 256 ไบต์
- วงจรตั้งเวลา/วงจรนับมีขนาด 2, 3 หรือ 16 บิต
- กำหนดรับส่งข้อมูลอนุกรมได้สองทิศทาง (Universal Synchronous Asynchronous Receiver Transmitter; UART)
- การขัดจังหวะแบ่งออกเป็น 2 ระดับจาก 5 หรือ 6 แหล่ง

- มีสัญญาณนาฬิกาอยู่ภายในตัว
- มีหน่วยความจำข้อมูลภายใน 4 หรือ 8 กิโลไบต์
- มีตำแหน่งของหน่วยความจำสำหรับเก็บโปรแกรมจำนวนทั้งหมด 64 กิโลไบต์
- คำสั่งมีทั้งหมด 111 คำสั่ง (64 รอบ)
- ทำงานด้วยเลขฐานสิบและเลขฐานสิบหก
- ตัวแปลภาษาเบสิกมีขนาด 8 กิโลไบต์ (8052-AH-BASIC)
- ควบคุมหน่วยความจำอีอีพรอมภายในตัวด้วยภาษาเบสิก (8052-AH-BASIC)
- มีคำสั่งเฉพาะของภาษาเบสิกใช้สำหรับอินพุต/เอาต์พุตของวงจรนับและเชื่อมข้อมูลแบบอนุกรม

2.3.1 ภายในหน่วยความจำ

ถ้าต้องการป้องกันการอ่านจากภายนอก ก็สามารถเลือกไมโครคอนโทรลเลอร์ 8751 และไมโครคอนโทรลเลอร์ 8752 มีหน่วยความจำแบบอีอีพรอมภายในที่จำกัดค่าเวลาการอ่านข้อมูลเพื่อกันการลอกเลียนแบบโปรแกรม ซึ่งจากรูปที่ 2.3 ได้แสดงโครงสร้างภายในหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 หน่วยความจำนี้สามารถแบ่งได้ 2 กลุ่ม คือหน่วยความจำสำหรับเก็บโปรแกรมและหน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำกลุ่มแรกมีความจุต่ำกว่า 4 หรือ 6 กิโลไบต์บรรจุอยู่ในรอมซึ่งส่วนไมโครคอนโทรลเลอร์ MCS-51 ที่ไม่มีรอมภายในและอาจใช้หน่วยความจำภายนอกซึ่งอาจเป็นรอม, แรม หรืออีพรอม ไมโครคอนโทรลเลอร์ MCS-51 จะอ่านหน่วยความจำสำหรับเก็บข้อมูล จะใช้เป็นที่เก็บตัวแปร การคำนวณหาผลลัพธ์ได้ทันที การจัดการกับข้อมูลที่มีขนาด 16 บิต หรือ คำ ตารางที่ใช้ค้นหาค่าต่างๆ และหน้าที่อื่นที่คล้ายกัน หน่วยความจำสำหรับเก็บข้อมูลโดยใช้ร่วมกับหน่วยความจำภายนอกได้ถึง 640 กิโลไบต์ ซึ่งเลือกใช้รอมหรือแรมก็ได้ และยังมีรีจีสเตอร์พิเศษที่ใช้หน่วยความจำภายนอกของแรมได้ 128 หรือ 256 กิโลไบต์

ตารางที่ 2.1 จำนวนหน่วยความจำภายในไมโครคอนโทรลเลอร์ MCS-51

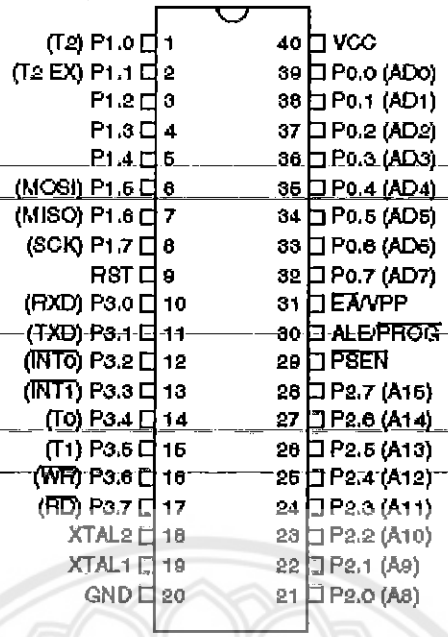
เบอร์	หน่วยความจำโปรแกรมภายใน	หน่วยความจำโปรแกรมภายใน	ตัวจับเวลา/ตัวนับ	การขัดจังหวะการทำงาน
8052AH	ROM 8 กิโลไบต์	RAM 256 ไบต์	3/16 บิต	6
8051AH	ROM 4 กิโลไบต์	RAM 128 ไบต์	2/16 บิต	5
8051	ROM 4 กิโลไบต์	RAM 128 ไบต์	2/16 บิต	5
8032AH	-	RAM 256 ไบต์	3/16 บิต	6
8031AH	-	RAM 128 ไบต์	2/16 บิต	5
8031	-	RAM 128 ไบต์	2/16 บิต	5
8751	ROM 4 กิโลไบต์	RAM 128 ไบต์	2/16 บิต	5
8752	ROM 4 กิโลไบต์	RAM 256 ไบต์	3/16 บิต	6

โครงสร้างภายในหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 มีดังนี้

Internal = หน่วยความจำภายใน
External = หน่วยความจำภายนอก
Program Counter = วงจรนับลำดับโปรแกรม
Program Memory = หน่วยความจำสำหรับเก็บโปรแกรม
Internal Data Ram = แรมภายในที่เก็บข้อมูล
Special Function Register = รีจีสเตอร์หน้าที่พิเศษ
Internal Data Memory = หน่วยความจำภายในสำหรับเก็บข้อมูล
External Data Memory = หน่วยความจำภายนอกสำหรับเก็บข้อมูล

2.3.2 รีจีสเตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 จะมีรีจีสเตอร์ซึ่งอำนวยความสะดวกในการใช้งานตามคำสั่งต่างๆ และมีแอคคิวมูเลเตอร์ รีจีสเตอร์ B ซึ่งใช้ในการคูณและหาร รีจีสเตอร์สถานะเป็นตัวชี้ตำแหน่งข้อมูล 2*8 บิตและ 1*16 บิต พอร์ตหมายเลข 0 ถึง พอร์ตหมายเลข 3 และมีรีจีสเตอร์เป็นตัวชี้ข้อมูล ซึ่งใช้ส่งและรับข้อมูลแบบอนุกรม มีรีจีสเตอร์ 16 บิต ที่เป็นวงจรตั้งเวลาและวงจรนับรีจีสเตอร์ซึ่งจะจองไว้สำหรับใช้สำหรับตัวที่ 3 เป็นรีจีสเตอร์คำสั่งสำหรับหน้าที่พิเศษ เช่น การขัดจังหวะ Read Time Clock กิโลไบต์



รูปที่ 2.4 การจัดวางขาต่างๆ ของไมโครคอนโทรลเลอร์ MCS-51

2.3.3 วงจรนับ/วงจรตั้งเวลา

จากตารางที่ 2.1 ไมโครคอนโทรลเลอร์เบอร์ 8052 มีวงจรนับและเป็นวงจรตั้งเวลาชนิด 16 บิต มากกว่าไมโครคอนโทรลเลอร์เบอร์ 8051 อยู่หนึ่งตัว ในการทำงานของวงจรนับและวงจรตั้งเวลาเป็นดังนี้

- เมื่อทำงานเป็นวงจรตั้งเวลา

รีจีสเตอร์ทำหน้าที่ตั้งเวลาจะเพิ่มค่าขึ้นหนึ่งของทุกๆ รอบคำสั่งของเครื่องและจะนับด้วยอัตราสูงสุดที่ 1/12 ของความเร็วสัญญาณนาฬิกาไมโครคอนโทรลเลอร์

- เมื่อทำงานเป็นวงจรนับ

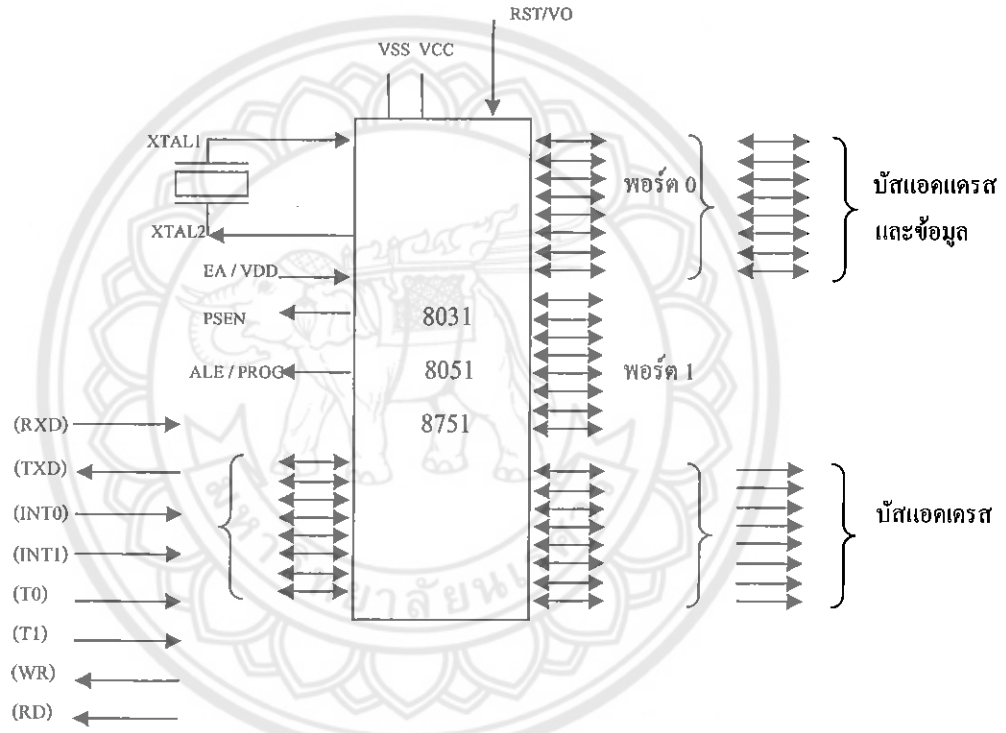
รีจีสเตอร์วงจรนับจะเพิ่มขึ้นหนึ่ง เมื่อมีสัญญาณป้อนให้ทางอินพุต T0, T1 หรือ T2 มีเฉพาะไมโครคอนโทรลเลอร์เบอร์ 8052 เป็นขอบสัญญาณขาลง อัตราการนับสัญญาณสูงสุดคือ 1/24 ของความเร็วสัญญาณนาฬิกาของไมโครคอนโทรลเลอร์ วงจรนับและวงจรตั้งเวลา 0 และ 1 มีวิธีโปรแกรมให้ทำงานได้ต่างกันถึง 4 แบบ ซึ่งรวมทั้งการทำงานเป็น 8 บิต หรือ 16 บิต และการบรรจุค่ารีเซตหนึ่งค่าได้เองอย่างอัตโนมัติ วงจรนับและวงจรตั้งเวลาที่ 1 ซึ่งเลือกโปรแกรมให้ทำหน้าที่เป็นตัวกำหนดสัญญาณของอัตราการส่งบิตออกไปยังพอร์ตอนุกรม สำหรับใช้เชื่อมต่อวงจรนับและวงจรตั้งเวลาที่ 2 เฉพาะไมโครคอนโทรลเลอร์เบอร์ 8052 เท่านั้น และมีการทำงานย่อยๆ อีก 3 ชนิด ดังนี้

1. วงจรนับ 16 บิตที่สามารถโหลดค่ากลับคืนเองอย่างอัตโนมัติ
2. วงจรนับที่จองไว้ชนิด 16 บิต

3. วงจรกำเนิดสัญญาณของการส่งบิต เพื่อใช้ในการเชื่อมโยงข้อมูล

2.3.4 สายต่างๆ ของบัสและพอร์ต

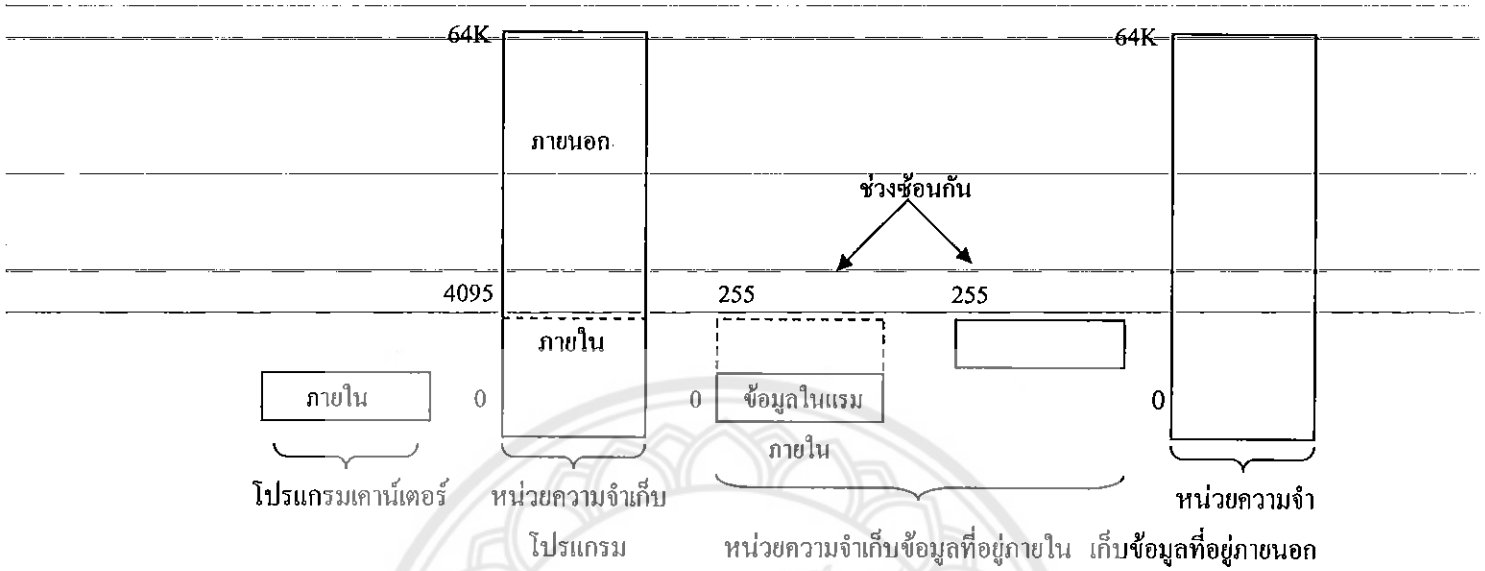
โครงสร้างภายในของไมโครคอนโทรลเลอร์เบอร์ MCS-51 โดยให้มีบัสสองทิศทาง 4 เส้น และพอร์ตขนาด 8 บิต ตามทฤษฎี แต่ที่จริงแล้วนั้นจะใช้ได้ต่อเมื่อมีการใช้หน่วยความจำภายในตัวรอมหรือแรม เมื่อไม่ใช้หน่วยความจำภายใน พอร์ต 0 และ 2 ใช้เป็นบัสของข้อมูลและตำแหน่ง ดังนั้นพอร์ต 2 พอร์ต ใช้งานเป็นอินพุตและเอาต์พุต ซึ่งพอร์ต 2 เป็นสัญญาณตำแหน่ง A15...A8 ส่วนพอร์ต 0 ทำหน้าที่เป็นสายสัญญาณตำแหน่ง A7...A0 ออกจากบิตข้อมูล D7...D0



รูปที่ 2.5 ลักษณะของพอร์ตเมื่อไมโครคอนโทรลเลอร์ทำงานกับหน่วยความจำภายนอก

เอาต์พุตของขา RD และ WR มาจากสายสัญญาณเอาต์พุตของพอร์ต P3 โดยโปรแกรมภายในใช้สัญญาณ RD และ WR เพื่อการเขียนข้อมูลและอ่านข้อมูลกับหน่วยความจำภายนอก และจะไม่ทำงานเมื่อมีภาษาเครื่องเก็บอยู่ในหน่วยความจำภายในและถ้าหากขา PSEN เป็นขารับสัญญาณสำหรับเปิดให้มีการอ่านหน่วยความจำภายนอก ถ้าสังเกตทุกๆ รอบคำสั่งระหว่างการทำงานด้วยโปรแกรมในรอมหรือหน่วยความจำแบบอีอีพรอม สัญญาณ PSEN ทำงานถึงสองครั้งเหมือนสัญญาณ ALE เพราะว่ามีการอ่านข้อมูลจำนวน 2 ไบต์ และในแต่ละรอบคำสั่งขา PSEN นี้จะไม่ทำงานเมื่อมีภาษาเครื่องเก็บอยู่ในหน่วยความจำภายใน ถ้าหน่วยความจำภายนอกไม่มีข้อมูล

บรรจุอยู่ ขา PSEN ก็จะไม่ทำงานเช่นกัน และส่วนไมโครคอนโทรลเลอร์เบอร์ 8052AH จะไม่ใช้สัญญาณขา PSEN เลย เนื่องจากใช้รอมภายในตัวซึ่งทำหน้าที่เป็นตัวแปลภาษาเบสิก



รูปที่ 2.6 โครงสร้างภายในหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51

ขา EA เป็นขาอินพุตที่จะใช้ร่วมกับตำแหน่งสัญญาณภายนอก โดยจะมีค่าเป็นลอจิก "0" และเมื่อไมโครโปรเซสเซอร์ทำการอ่านคำสั่งจากหน่วยความจำภายนอก ซึ่งโดยปกติแล้วนั้นไมโครคอนโทรลเลอร์จะอ่านหน่วยความจำภายในน้อยกว่าอ่านจากหน่วยความจำภายนอก ขา EA ยังเป็นขาอินพุตสำหรับป้อนแรงดันไฟฟ้า 21 โวลต์ เพื่อจะเขียนโปรแกรมให้กับหน่วยความจำแบบอีอีพรอม และสำหรับกรณีที่ใช้ไมโครคอนโทรลเลอร์ 8751 หรือไมโครคอนโทรลเลอร์ 8752

2.3.5 พอร์ตชนิดอนุกรมอยู่ภายใน

ไมโครคอนโทรลเลอร์ทั้งหมดในตระกูล MCS-51 เป็นไมโครคอนโทรลเลอร์ที่มีการเชื่อมโยงข้อมูลชนิดสองทาง ทำให้รับและส่งข้อมูลได้พร้อมกัน ตัวรับส่งข้อมูลชนิดอะซิงโครนัส (Asynchronous Receiver) มีบัฟเฟอร์สำหรับข้อมูลเป็นพิเศษ เพื่อเพิ่มความเร็วในการสื่อสารพอร์ตชนิดอนุกรมนั้นสามารถเลือกโปรแกรมเพื่อเลือกใช้การทำงานแบบใดแบบหนึ่งใน 4 แบบ ด้วยการใช้โปรแกรมควบคุมอัตราการส่งข้อมูลและรูปแบบของข้อมูล อัตราการส่งข้อมูลที่เลือกใช้ได้สูงถึง 19,200 บิต/วินาที ด้วยความเร็วของสัญญาณนาฬิกา 1 เมกะเฮิรตซ์ สำหรับใช้ในระบบเครือข่าย (Network) และระบบการสื่อสารของไมโครคอนโทรลเลอร์หลายตัวร่วมกันจะเลือกความเร็วของสัญญาณนาฬิกาด้วยวงจรนับและวงจรตั้งเวลา

2.3.6 การขัดจังหวะและชุดคำสั่ง

ไมโครคอนโทรลเลอร์ 8051 รับการขัดจังหวะได้ 5 แหล่ง ส่วนไมโครคอนโทรลเลอร์ 8052 รับการขัดจังหวะจากอุปกรณ์อื่นๆ ได้ถึง 6 แหล่ง คือ ขา INT0 และขา INT1 ซึ่งกำหนดให้ใช้ระดับพัลส์หรือขอบขาพัลส์ก็ได้ วงจรนับและวงจรตั้งเวลาที่ 0 และ 1 ซึ่งสำหรับการใช้งานไมโครคอนโทรลเลอร์ 8051 จะเพิ่มวงจรตั้งเวลาหรือ วงจรนับตัวที่ 2 และตัวสุดท้ายจากพอร์ตอนุกรมสามารถกำหนดลำดับความสำคัญของการขัดจังหวะได้ 2 ระดับ ไม่ต้องอาศัยวงจรภายนอกเข้ามาช่วยแต่ละแหล่ง การขัดจังหวะ 5 หรือ 6 แหล่งนั้น สามารถกำหนดให้เป็นเวคเตอร์เฉพาะตัวชี้ตำแหน่ง ดังนั้นเมื่อมีการขัดจังหวะเข้ามาที่ตัวไมโครโปรเซสเซอร์จะกระโดดไปที่ส่วนของโปรแกรมที่ทำงานตามวัตถุประสงค์ของการขัดจังหวะนั้น และหลังจากเก็บข้อมูลต่าง ๆ ของโปรแกรมการนับลงในสแตก

บทที่ 3
ขั้นตอนการดำเนินโครงงาน

3.1 ศึกษาเกี่ยวกับ E-cash, สมาร์ตการ์ด และไมโครคอนโทรลเลอร์ MCS-51

จากที่ได้ศึกษาระบบ E-cash มานั้นทำให้ทราบว่า ในอนาคตระบบ E-cash จะเข้ามามีบทบาทมากยิ่งขึ้น เนื่องจากในโลกยุคปัจจุบันได้มีการนำเอาเทคโนโลยีสารสนเทศเข้ามาเป็นส่วนประกอบในการทำธุรกรรมต่างๆ และมีแนวโน้มที่จะสูงขึ้นทุกวัน ดังนั้นปริญญานิพนธ์นี้ได้มีการนำเสนอระบบ E-cash ในรูปแบบของสมาร์ตการ์ด เพื่อที่จะเป็นแนวทางที่จะพัฒนาต่อไปในอนาคต และจากที่ได้ศึกษาเกี่ยวกับสมาร์ตการ์ด ทำให้เข้าใจถึงหลักการทำงานของสมาร์ตการ์ดแต่ละชนิด และได้เลือกสมาร์ตการ์ดชนิด Memory มาทำการทดลองเนื่องจากง่ายต่อการศึกษาในเบื้องต้นและมีต้นทุนที่ต่ำกว่าสมาร์ตการ์ดชนิดอื่น ๆ บัตรสมาร์ตการ์ดชนิดที่เลือกมานี้ยังมี security ในตัวเองอีกด้วยซึ่งถือได้ว่ามีความปลอดภัยของข้อมูลอยู่ในระดับหนึ่ง และได้มีการเลือกใช้ไมโครคอนโทรลเลอร์ MCS-51 แทนการใช้คอมพิวเตอร์ในส่วนของร้านค้าต่าง ๆ นั้น เพื่อประหยัดต้นทุนทำให้ร้านค้าทั่ว ๆ ไปก็สามารถเข้าร่วมโครงการได้ เพราะว่าไม่ต้อง เสียค่าใช้จ่ายสูงในการที่จะซื้อเครื่องชำระเงิน และการใช้งานก็ง่ายไม่ยุ่งยากเหมือนกับการใช้คอมพิวเตอร์

อุปกรณ์ที่ใช้ในการทำโครงงาน

อุปกรณ์ส่วนจุดขาย

- MCS 51 Microcontroler board
- Smart card read/write board
- หม้อแปลง 9 vdc

อุปกรณ์ส่วนจุดแลกบัตร

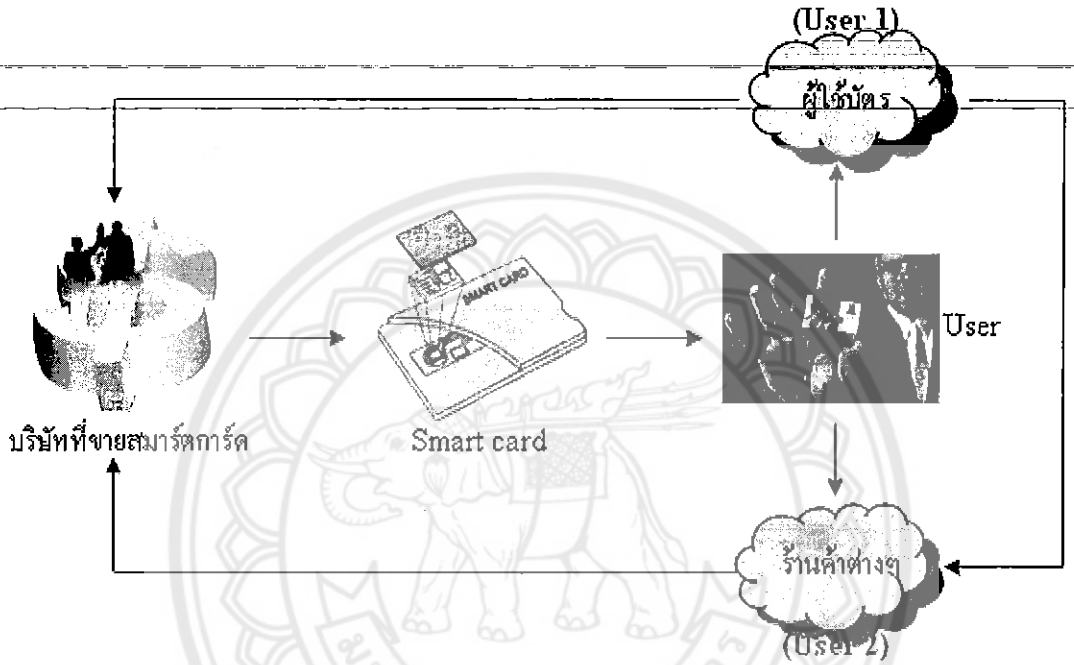
- Computer (มี Serial port)
- Smart card SLE4442 (memory card)
- สาย RS232

ผู้ซื้อ / ผู้ขาย

- Smart card SLE4442 (memory card)

3.2 ออกแบบระบบ E-cash โดยใช้สมาร์ตการ์ด, ไมโครคอนโทรลเลอร์MCS-51 และคอมพิวเตอร์

ระบบ E-cash ที่ออกแบบขึ้นมานี้จะประกอบไปด้วยบริษัทที่สร้างระบบ E-cash นั่นก็คือ บริษัทที่ขายสมาร์ตการ์ด และผู้ใช้ ซึ่งผู้ใช้สมาร์ตการ์ดนี้ยังแบ่ง ออกเป็น ผู้ที่ใช้สมาร์ตการ์ดเพื่อไปซื้อสินค้าหรือบริการและผู้ใช้ที่เป็นผู้ขายสินค้านั่นเอง ดังรูปที่ 3.1



รูปที่ 3.1 แสดงรูปแบบของระบบ E-cash โดยใช้สมาร์ตการ์ด

จากรูปที่ 3.1 ผู้ใช้สามารถมาซื้อบัตรสมาร์ตการ์ดได้ที่บริษัทที่ขายสมาร์ตการ์ด โดยผู้ใช้ยังแบ่งออกเป็นผู้ใช้ที่นำสมาร์ตการ์ดไปซื้อสินค้าและบริการต่างๆ (User1) และผู้ใช้ที่เป็นผู้ขายสินค้าและบริการ (User2) เมื่อผู้ใช้ไปซื้อสมาร์ตการ์ด บริษัทที่ผลิตสมาร์ตการ์ดก็จะเติมยอดเงินลงในสมาร์ตการ์ดให้เลยโดยที่วงเงินขั้นต่ำจะอยู่ที่ 100 บาท และไม่เกิน 5,000 บาท ต่อจากนั้นก็สามารถนำสมาร์ตการ์ดไปใช้ได้ (สำหรับ User1) นั่นก็คือ สมาร์ตการ์ดของ User1 จะใช้ในการนำไปซื้อสินค้าและบริการต่าง ๆ แต่สำหรับสมาร์ตการ์ดของ User2 จะใช้ในการบันทึกยอดเงินการขายสินค้าและบริการต่าง ๆ จากนั้นก็จะนำไปขึ้นเงินกับทางบริษัทที่ขายสมาร์ตการ์ดนั้น ๆ

3.2.1 ออกแบบเครื่องอ่านและเขียนบัตรสมาร์ตการ์ด

ในสภาวะปกติหากไม่มีการสอดบัตรในเครื่องอ่านสมาร์ตการ์ดไฟสีแดงจะติดอันเดียว แต่เมื่อสอดบัตรสมาร์ตการ์ดเข้าไปในเครื่องอ่าน จะมีไฟสีเขียวติดเพิ่มขึ้นมาอีก 1 ดวง หากมีการ

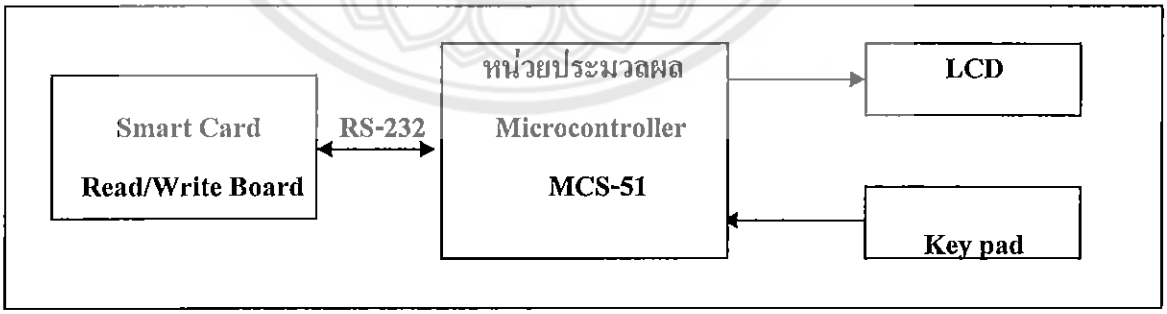
อ่านหรือเขียนข้อมูลในสมาร์ตการ์ด บัตรสมาร์ตการ์ดและเครื่องอ่าน-เขียนสมาร์ตการ์ดจะทำการตรวจสอบค่า PSC (Programmable Security Code) หากถูกต้องก็จะสามารถทำการอ่านหรือเขียนข้อมูลลงในบัตรได้ แต่ถ้าไม่ถูกต้องจะไม่สามารถทำการอ่านหรือเขียนข้อมูลลงในบัตรได้และถ้าตรวจสอบเกิน 3 ครั้งและยังไม่ถูกต้อง จะทำให้บัตรสมาร์ตการ์ดนั้น ๆ ไม่สามารถใช้งานได้อีกเลย

เมื่อผู้ใช้บริการมาใช้บริการจะต้องทำการซื้อบัตรที่จุดจำหน่ายบัตร พนักงานก็จะทำการสอดบัตรสมาร์ตการ์ดเข้าไปในเครื่องอ่าน-เขียนบัตรสมาร์ตการ์ดและหลังจากนั้นก็จะทำการใส่ยอดเงินตามที่ผู้ใช้บริการต้องการลงในบัตรสมาร์ตการ์ด และพนักงานก็จะทำการเก็บเงินสดจากผู้ใช้บริการ จากนั้นผู้ใช้บริการก็สามารถนำบัตรสมาร์ตการ์ดไปใช้ซื้อของตามร้านค้าต่าง ๆ ได้

เมื่อผู้ใช้บริการไปซื้อสินค้าจากร้านค้า ร้านค้าก็จะทำการสอดบัตรสมาร์ตการ์ดลงในเครื่องอ่านและเขียนสมาร์ตการ์ด เครื่องจะทำการอ่านข้อมูล โดยจะแสดงจำนวนเงินที่มีอยู่ในบัตรสมาร์ตการ์ด จากนั้นร้านค้าจะทำการคีย์ข้อมูลตามราคาของสินค้านั้น ๆ ลงในบัตรสมาร์ตการ์ดและจะแสดงผลจำนวนเงินคงเหลือที่จอ LCD เมื่อจำนวนเงินในบัตรสมาร์ตการ์ดหมดจะไม่สามารถใช้ซื้อสินค้าได้ จะต้องทำการติดต่อเจ้าหน้าที่ เพื่อที่จะเติมยอดเงินลงในบัตรอีกครั้ง

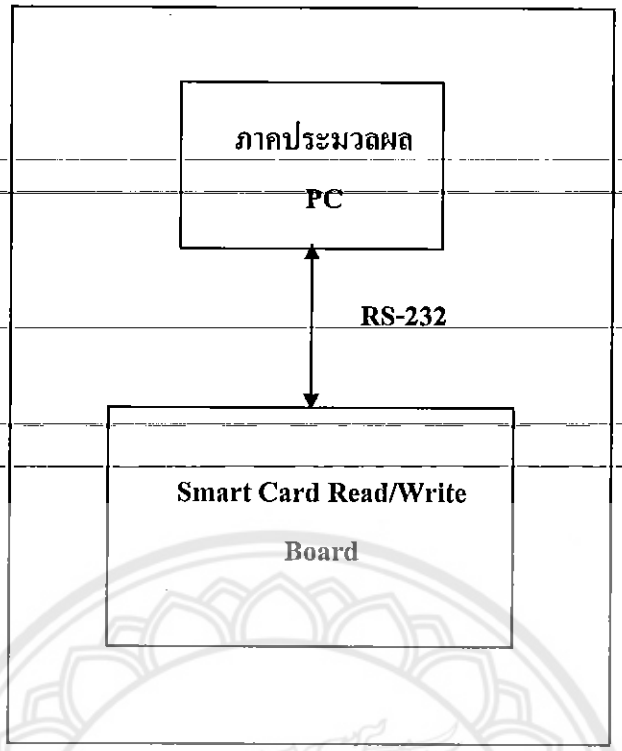
ทางด้านผู้ขายสินค้าทุกครั้งที่มีการซื้อขายสินค้า ราคาสินค้าที่ขายออกไปจะถูกเก็บไว้ในบอร์ดไมโครคอนโทรลเลอร์ MCS-51 และจะบวกยอดเงินเพิ่มไปเรื่อย ๆ ในทุก ๆ ครั้งที่มีการซื้อสินค้า เมื่อผู้ขายสินค้าได้ยอดตามที่ต้องการก็จะนำบัตรสมาร์ตการ์ดของตนเองมาเสียบเข้าที่เครื่องอ่าน-เขียนสมาร์ตการ์ดและก็ทำการเขียนยอดเงินจำนวนนั้นลงในบัตร เพื่อให้ผู้ขายสินค้านำบัตรไปขึ้นเงินจากบริษัทผู้ผลิตอีกที

3.2.2 ส่วนประกอบของเครื่องอ่าน-เขียนบัตรสมาร์ตการ์ด



รูปที่ 3.2 บล็อคไดอะแกรมแสดงการทำงานของเครื่องอ่านและเขียนสมาร์ตการ์ดที่ร้านขายสินค้า

จากรูปที่ 3.2 บล็อกไดอะแกรมแสดงการทำงานของเครื่องอ่านและเขียนสมาร์ตการ์ดที่ร้านขายสินค้า สามารถอธิบายการทำงานได้โดย เครื่องอ่าน-เขียนสมาร์ตการ์ดจะทำงานได้โดยการประมวลผลของไมโครคอนโทรลเลอร์ MCS-51 ซึ่ง Key pad จะเป็นตัวทำหน้าที่รับค่าการส่งข้อมูลต่าง ๆ ที่เกิดจากการขายสินค้า ส่วน LCD จะเป็นตัวแสดงผลออกทางหน้าจอ



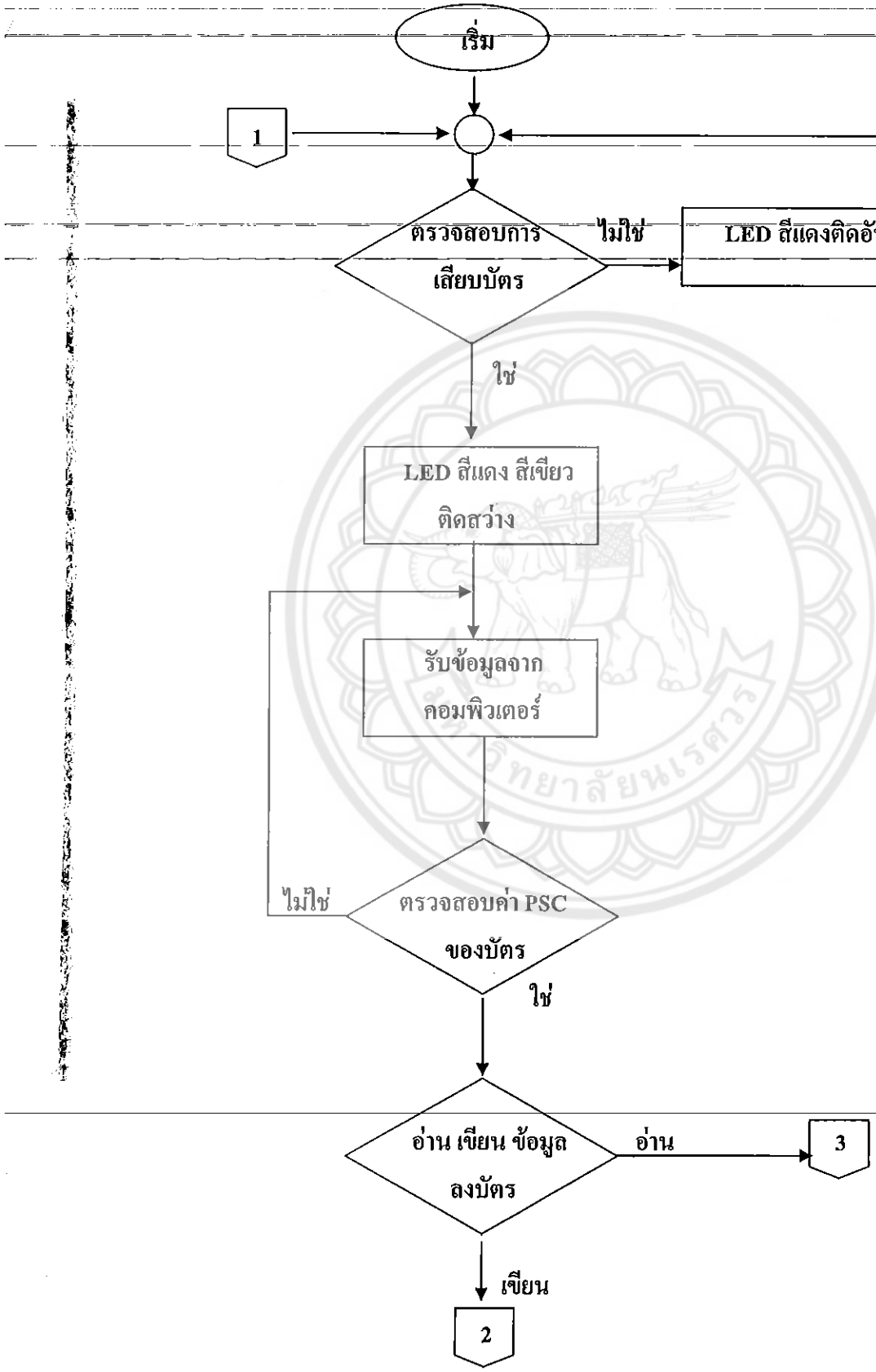
รูปที่ 3.3 บล๊อคไดอะแกรมแสดงการทำงานของเครื่องอ่านและเขียนสมาร์ตการ์ดที่จะจำหน่ายบัตร

จากรูปที่ 3.3 เป็นบล๊อคไดอะแกรมแสดงการทำงานของเครื่องอ่านและเขียนสมาร์ตการ์ดที่จุดจำหน่ายบัตร เครื่องอ่านและเขียนบัตรสมาร์ตการ์ดสามารถทำงานได้โดยการประมวลผลของเครื่องคอมพิวเตอร์ (PC) โดยจะติดต่อผ่านทางพอร์ต RS-232

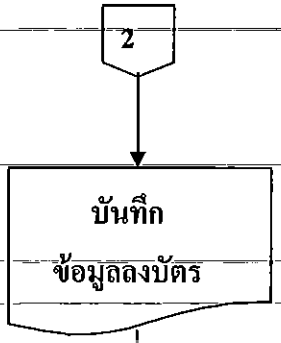
3.2.3 โฟลว์ชาร์ตแสดงขั้นตอนการทำงาน

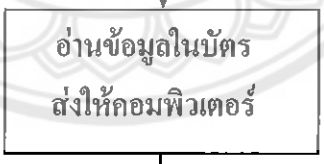
จากโฟลว์ชาร์ตขั้นตอนการทำงานของจุดจำหน่ายบัตร เมื่อเริ่มการทำงาน โปรแกรมจะคอยตรวจสอบการเสียบบัตร ถ้ายังไม่มีการเสียบบัตร LED สีแดงจะติดหลอดเดียว แต่เมื่อมีการเสียบบัตร LED ทั้งสีแดงและสีเขียวจะติดพร้อมกัน จากนั้นก็จะรอรับข้อมูลหรือคำสั่งที่ส่งมาจากคอมพิวเตอร์ โดยจะทำการตรวจสอบค่า PSC ของบัตรก่อนถ้าถูกต้องจะสามารถทำการอ่านหรือเขียนข้อมูลลงบัตรได้

โฟลว์ชาร์ตขั้นตอนการทำงานของจุดจำหน่ายบัตร



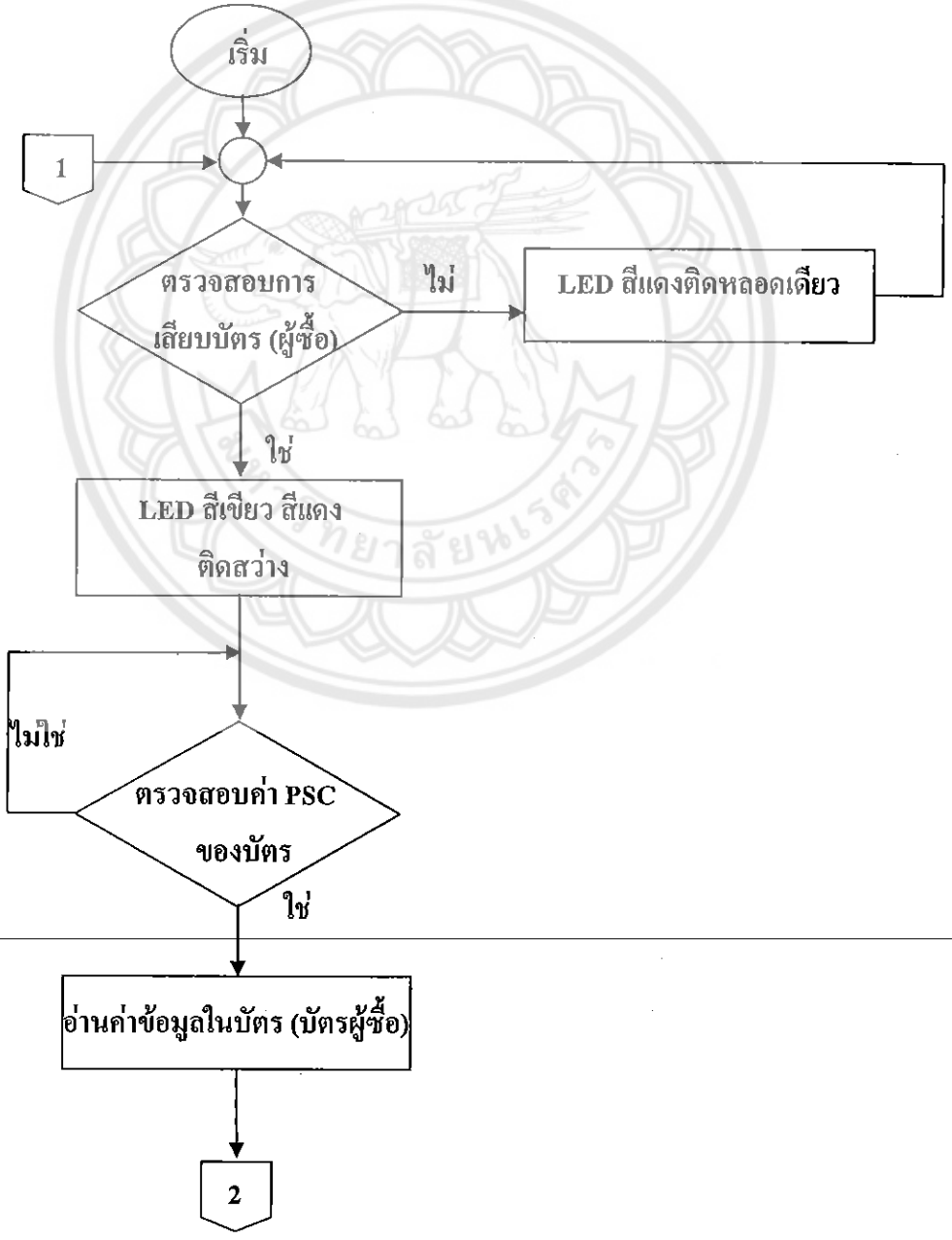
โฟลว์ชาร์ตขั้นตอนการทำงานของจุดจำหน่ายบัตร (ต่อ)





จากโฟลว์ชาร์ตขั้นตอนการทำงานของร้านค้าต่าง ๆ เมื่อเริ่มการทำงาน โปรแกรมจะคอยตรวจสอบการเสียบบัตร ถ้ายังไม่มีการเสียบบัตร LED สีแดงจะติดเพียงหลอดเดียว แต่เมื่อมีการเสียบบัตรสมาร์ตการ์ด LED ทั้งสีแดงและสีเขียวจะติดพร้อมกัน จากนั้นจะรอรับข้อมูลที่ส่งจากไมโครคอนโทรลเลอร์ MCS-51 โดยจะทำการตรวจสอบค่า PSC ถ้าถูกต้องก็จะสามารถอ่านและเขียนข้อมูลลงบัตรได้ โดยจุดนี้ จะมีผู้ใช้สมาร์ตการ์ดอยู่ 2 คน คือ ผู้ซื้อสินค้าและผู้ค้าสินค้า เมื่อผู้ซื้อสินค้ามาซื้อสินค้าแล้วบอร์ดไมโครคอนโทรลเลอร์ก็จะเก็บของมูลไว้ เพื่อ มาเพิ่มยอดเงินให้ผู้ขายนั่นเอง

โฟลว์ชาร์ตขั้นตอนการทำงานที่ร้านค้าต่าง ๆ



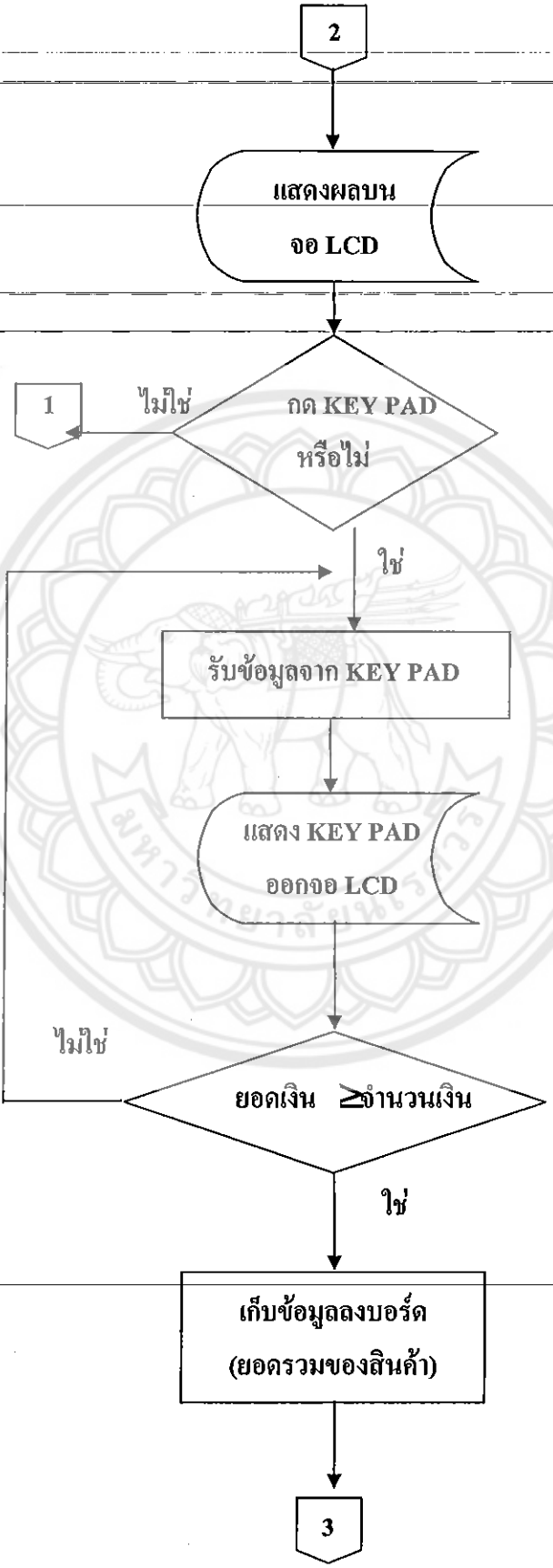
ห้องสมุดคณะวิศวกรรมศาสตร์

โ607852e

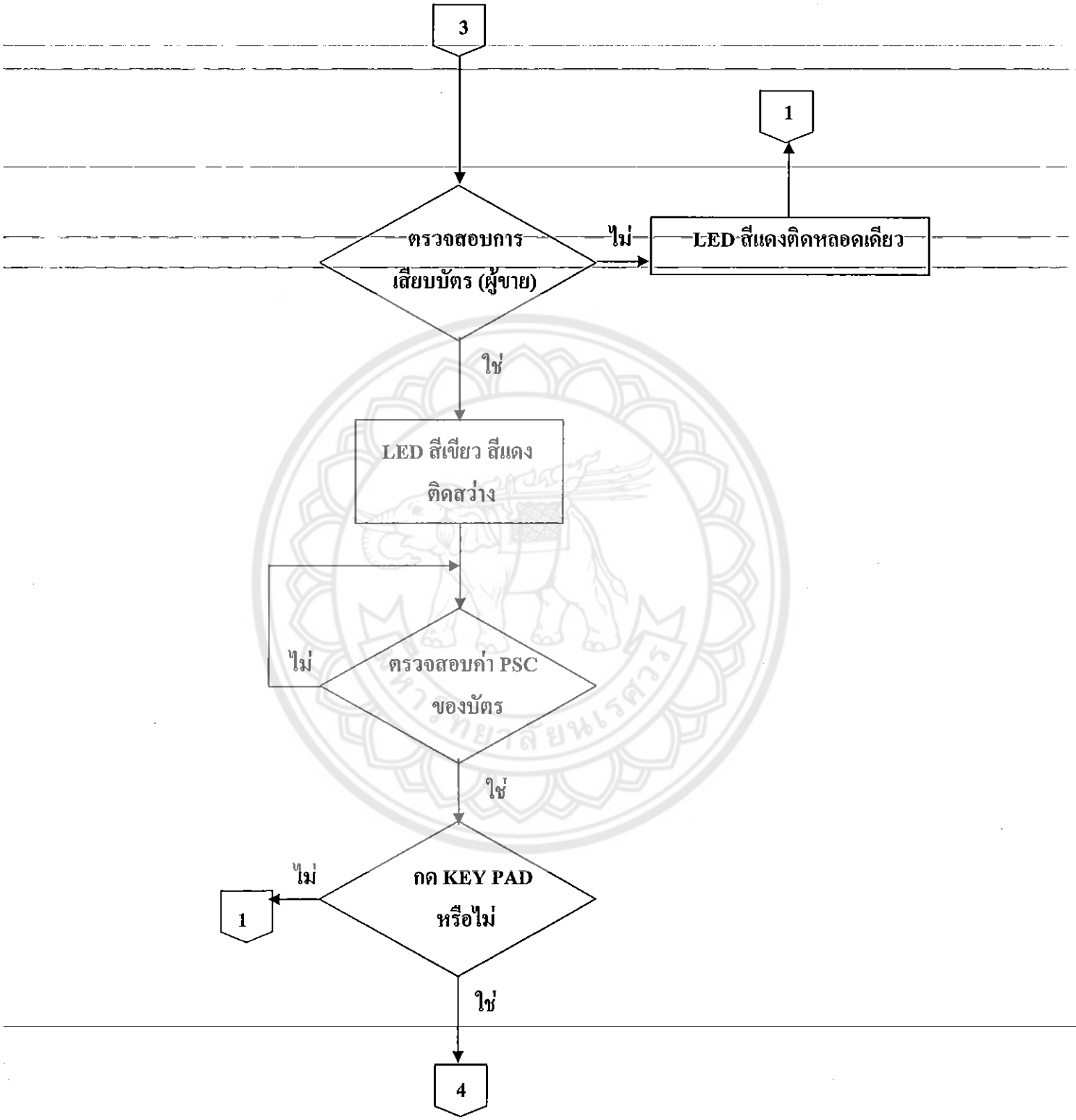
5000089

ม.
พ7911ก
2547
ค.2

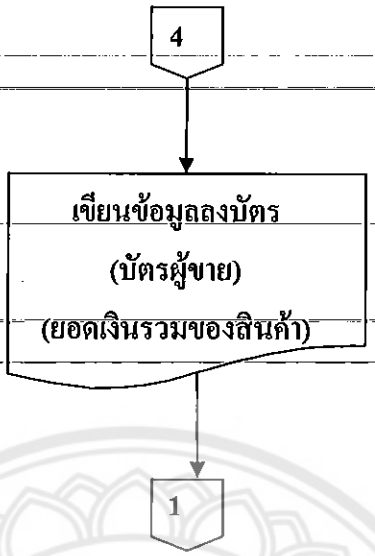
โฟลว์ชาร์ตขั้นตอนการทำงานที่ร้านค้าต่าง ๆ (ต่อ)



โฟลว์ชาร์ตขั้นตอนการทำงานที่ร้านค้าต่าง ๆ (ต่อ)

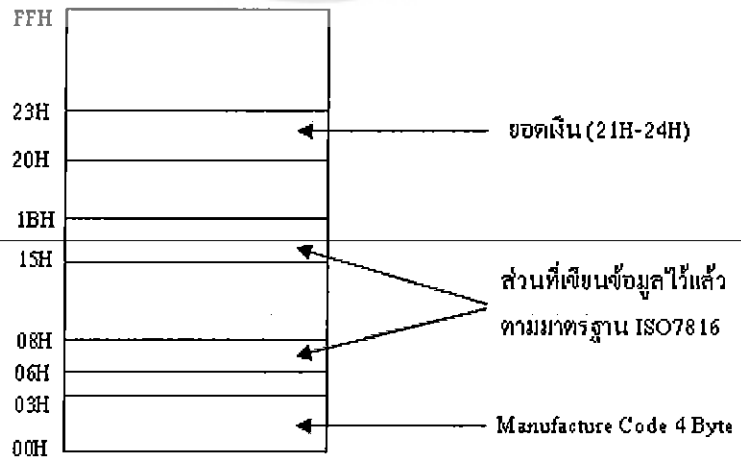


โฟลว์ชาร์ตขั้นตอนการทำงานที่ร้านค้าต่าง ๆ (ต่อ)



3.2.4 การจัดการหน่วยความจำของบัตรสมาร์ตการ์ดในการเก็บข้อมูล

จากรูปที่ 3.3 เป็นการจัดการหน่วยความจำของบัตรสมาร์ตการ์ด เบอร์ SLE4442 มีหน่วยความจำแบบ EEPROM ขนาด 256 ไบต์โดยแบ่งเป็น Protectable Data Memory 32 ไบต์ และ Unprotectable Data Memory 224 ไบต์ สามารถอ่านเขียนข้อมูลได้ 100,000 ครั้ง เก็บข้อมูลได้นานถึง 10 ปี ส่วนที่เป็น Protectable Data Memory นั้นสามารถเขียนข้อมูลถาวรไว้โดย จะลบหรือแก้ไขเปลี่ยนแปลงข้อมูลไม่ได้อีกเลย และในส่วนนี้ถูกเขียนข้อมูลไว้แล้ว 12 ไบต์ ตามมาตรฐาน ISO 7816 คือที่ตำแหน่ง 06H, 07H,15H-1AH ดังนั้นจึงต้องเริ่มทำการเขียนข้อมูลลงบัตรในไบต์ที่ 32 เป็นต้นไป โดยในโครงงานนี้จะใช้ แอดเดรส ที่ 21H -23H ในการเก็บข้อมูลที่เป็นจำนวนเงินรวมทั้งหมด 4 ไบต์



รูปที่ 3.4 การจัดการหน่วยความจำของสมาร์ตการ์ดในการเก็บข้อมูล

3.3 เขียนโปรแกรมติดต่อกับสมาร์ตการ์ด, บอร์ด MCS-51 และ คอมพิวเตอร์

โปรแกรมที่ใช้ในการติดต่อจะแบ่งออกเป็นสองส่วนคือ ส่วนที่ติดต่อระหว่างเครื่องอ่านและเขียนสมาร์ตการ์ดกับบอร์ดไมโครคอนโทรลเลอร์ MCS-51 51 จะใช้ภาษาซีในการเขียนโปรแกรม ติดต่อโดยใช้โปรแกรม Keil uVision3 ในการเขียนโปรแกรมติดต่อ และอีกส่วนจะเป็นเครื่องอ่านและเขียนสมาร์ตการ์ดกับคอมพิวเตอร์ ใช้โปรแกรม Visual basic 6 ในการเขียนโปรแกรม

3.3.1 การเซ็ตพอร์ตเชื่อมต่อในภาษาซี

```
void set_serial_port()
{
 PCON=PCON|0x80; // setup scon in bit 7 (smod)
 SCON=0x52; // mode 1
 TMOD=0x20; // timer mode 2
  TH1=244;   //set br 9600 for 11.0592*2
 TR1=1;
}
```

3.3.2 การเซ็ตพอร์ตเชื่อมต่อในภาษาวิชวลเบสิก

```
Private Sub Form_Load()
MSComm1.Settings = "9600,n,8,1" ' Tell the control to read entire buffer when Input is used.
MSComm1.InputLen = 1 ' Close the port.
MSComm1.InputMode = comInputModeText ' Data is retrieved through the Input property as
text.
MSComm1.OutBufferSize = 1024 ' Setting Receive and Transmit Buffer Properties at Design
Time
MSComm1.InBufferSize = 1024
MSComm1.RThreshold = 1 ' Set the return minimum number of character to receive
MSComm1.SThreshold = 1 ' Set the minimum number of character allowable in transmit
MSComm1.PortOpen = True ' Send the attention command to the modem.
End Sub
```

3.4 เขียนโปรแกรมในส่วนที่เกี่ยวข้องกับการใช้งานระบบ E-cash

โปรแกรมการทำงานหลักในระบบก็จะเป็นการเพิ่มและลดยอดเงินในตัวบัตร โดยจะมีการอ่านเขียนข้อมูลจากบัตร ซึ่งข้อมูลที่บันทึกจะเป็นเลขฐานสิบหกขนาด 4 ไบต์ โดยข้อมูลที่บันทึกจะเป็นจำนวนสตางค์ ซึ่งจะมีการอ่านเขียนข้อมูลดังนี้

3.4.1 ส่วนของโปรแกรมภาษาซี

- การอ่านข้อมูล (แปลงเลขฐานสิบหกเป็นจำนวนนับ)

```
int1=make_hex(str_2[0]);
int2=make_hex(str_2[1]);
int3=make_hex(str_2[2]);
int4=make_hex(str_2[3]);
int5=make_hex(str_2[4]);
int6=make_hex(str_2[5]);
int7=make_hex(str_2[6]);
int8=make_hex(str_2[7]);
value_read=(int8*1)+(int7*16)+(int6*256)+(int5*4096)+(int4*65536)+(int3*1048576)+(int2*16
777216)+(int1*268435456);        // make int value
```

- การเขียนข้อมูล

คำสั่ง verify psc คือ :6 ซึ่งตามด้วยรหัส psc คือ FFFFFF

```
out_serial(sprintf(str_buf,":6FFFFFF%c",0x0d)); // verify psc
```

Sprintf() ใช้ในการส่งข้อมูลไปที่บัพเพอร์ของ mcs-51 ส่วนคำสั่งที่ใช้ในการเขียนข้อมูลคือ :3 โดยเขียนไว้ที่ address 21 ตามด้วยข้อมูล 08lX คือ ข้อมูลชนิด long ความยาวเท่ากับ 8 โดยจะแปลงเป็นเลขฐานสิบหก และเติมศูนย์ในตัวหน้าเพื่อให้ข้อมูลครบ 8 อักษร ส่วน 0x0d ที่ตามท้ายคือคำสั่งที่บอกว่าข้อมูลที่ส่งไปได้หมดแล้ว

```
out_serial(sprintf(str_buf,":321%08lX%c",(buffer_int2-buffer_int1),0x0d)); //send data in buffer
                                                                    to function out_serial
```

ฟังก์ชัน out_serial() ใช้ในการวนรอบเพื่อส่งข้อมูลที่ละอักษรให้กับฟังก์ชัน put_serial()

```
void out_serial(unsigned char dat)
{ unsigned char i;
 for(i=1;i<=dat;i++) // put all data in buffer to serial
    { put_serial(str_buf[i-1]); // call put serial to send data to card reader
    }
}
```

ฟังก์ชัน put_serial () ส่งข้อมูลออกไปยังเครื่องอ่านเขียนสมาร์ดการ์ด

```
// send data to card reader
void put_serial(char c)
{ while (tx_counter == tx_buffer_size); // chk and wait for tx_buffer_full
 ES=0;
 if ((tx_counter > 0) || (send_status==1))
    {tx_buffer[tx_wr_index]=c;
    if (++tx_wr_index == tx_buffer_size)
       {tx_wr_index=0;
      }
     ++tx_counter;
    }
   else
    { SBUF=c;
     send_status=1;
   }
 ES=1;
}
```

3.4.2 ส่วนของโปรแกรมวิชวลเบสิก

- การอ่านข้อมูล

อ่านข้อมูลในบัตรและแปลงเป็นจำนวนเงิน โดยแสดงเป็นจำนวนบาท

```
str_buffer2 = Left(str_buffer1, 8)
str_buffer2 = "&H" & str_buffer2
buffer_int1 = Val(str_buffer2)
buffer_int1 = buffer_int1 / 100
Label2.Caption = (buffer_int1 & " บาท")
```

- การเขียนข้อมูล

เติมศูนย์ให้ตัวอักษรที่ส่งไปครบ 8 (4 byte)

```
If buffer_int3 > &HFFFFFFFF Then
        str2 = Hex(buffer_int3)
        End If
If buffer_int3 <= &HFFFFFFFF Then
        str2 = Hex(buffer_int3)
        str2 = "0" & str2
        End If
If buffer_int3 <= &HFFFFFF Then
        str2 = Hex(buffer_int3)
        str2 = "00" & str2
        End If
If buffer_int3 <= &HFFFFF Then
        str2 = Hex(buffer_int3)
        str2 = "000" & str2
        End If
If buffer_int3 <= 65535 Then
        str2 = Hex(buffer_int3)
        str2 = "0000" & str2
        End If
If buffer_int3 <= &HFFF Then
        str2 = Hex(buffer_int3)
```

```
        str2 = "00000" & str2
        End If
    If buffer_int3 <= &HFF Then
        str2 = Hex(buffer_int3)
        str2 = "000000" & str2
        End If
    If buffer_int3 <= &HF Then
        str2 = Hex(buffer_int3)
        str2 = "0000000" & str2
        End If
```

คำสั่งที่ใช้ในการ verify psc

```
MSComm1.Output = ":6FFFFFF" // verify
MSComm1.Output = Chr(13)  //end of data
```

คำสั่งที่ใช้ในการเขียนข้อมูล ซึ่งก็คือ str2

```
MSComm1.Output = ":321" // write data in address 21
MSComm1.Output = str2  // data (money)
MSComm1.Output = Chr(13) //end of data
```

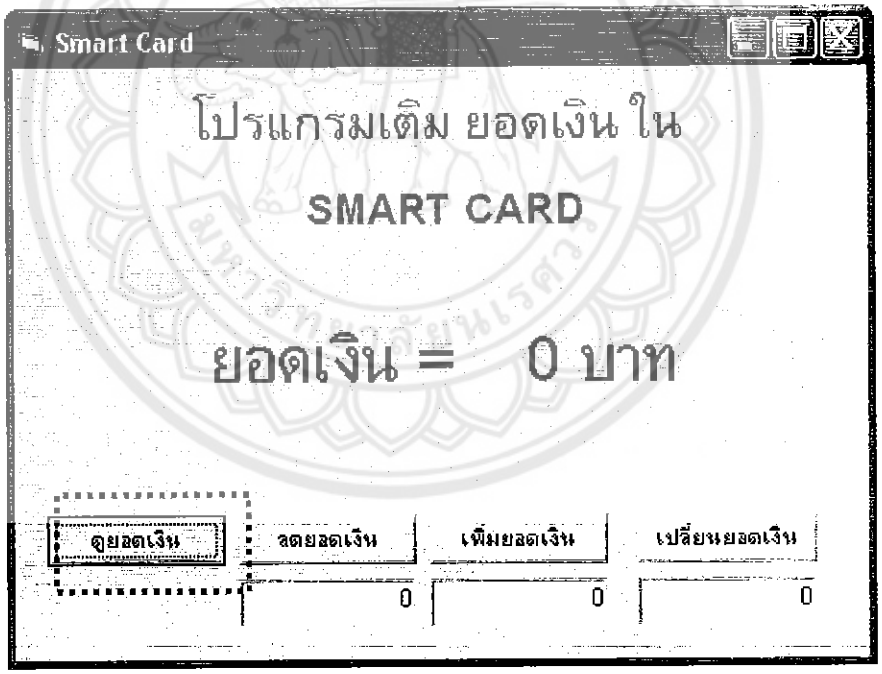
บทที่ 4
ผลของการทำโครงงานปริญญานิพนธ์

4.1 โปรแกรมวิชวลเบสิกสำหรับเติมยอดเงินให้สมาร์ตการ์ด

โดยการออกแบบรูปแบบหน้าตาของโปรแกรมวิชวลเบสิกกับเครื่องอ่านและเขียนสมาร์ตการ์ดนั้น จะออกแบบให้มีความสะดวกต่อการใช้งาน หน้าตาโปรแกรมไม่ยุ่งยากซับซ้อน โดยมีการเชื่อมต่อกันระหว่างเครื่องอ่านและเขียนสมาร์ตการ์ดกับคอมพิวเตอร์ผ่านทางพอร์ต COM1 ของเครื่องคอมพิวเตอร์

4.1.1 การดูยอดเงินสมาร์ตการ์ด

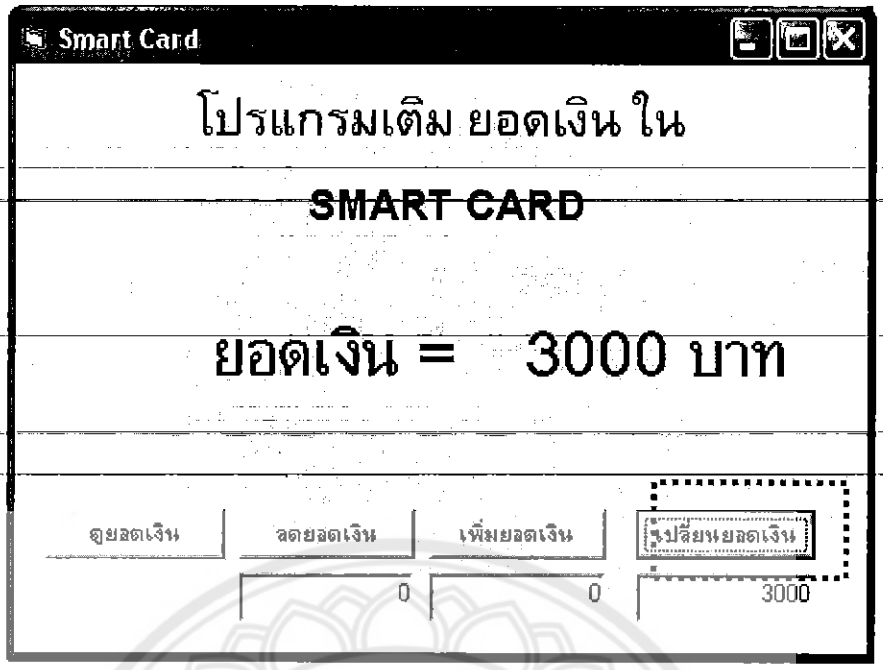
เมื่อคลิกที่ปุ่มดูยอดเงิน เครื่องอ่านและเขียนสมาร์ตการ์ดจะทำการอ่านข้อมูลในบัตร ซึ่งได้เป็นจำนวนเงินออกมา ดังรูปที่ 4.1



รูปที่ 4.1 แสดงการดูยอดเงินที่มีอยู่ในสมาร์ตการ์ด

4.1.2 การเปลี่ยนยอดเงินของสมาร์ตการ์ด

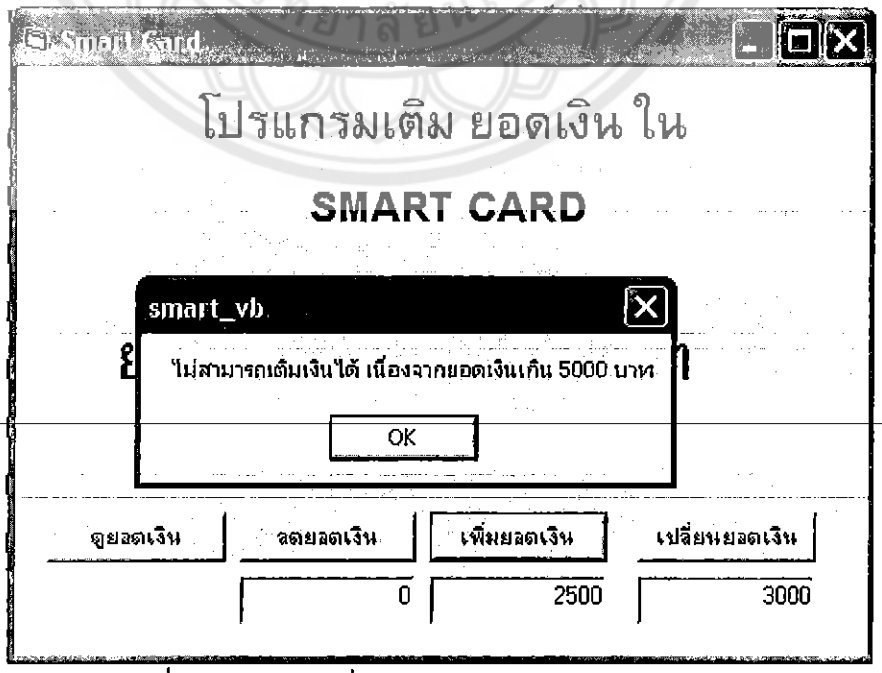
สามารถเปลี่ยนแปลงได้ทันที โดยใส่ยอดเงินที่ต้องการลงในช่อง เปลี่ยนยอดเงิน แล้วคลิกปุ่มเปลี่ยนยอดเงิน ดังรูปที่ 4.2



รูปที่ 4.2 แสดงการเปลี่ยนยอดเงินลงในสมาร์ตการ์ด

4.1.3 การเพิ่มยอดเงินลงในสมาร์ตการ์ด

ให้ใส่ยอดเงิน ลงในช่องเพิ่มยอดเงิน แล้วคลิกที่ปุ่มเพิ่มยอดเงิน แต่การเพิ่มยอดเงินในแต่ละครั้งนั้นจะไม่สามารถเพิ่มได้เกิน 5,000 บาท เมื่อรวมกับยอดเงินที่เหลืออยู่ ถ้ามีการเพิ่มยอดเงินเกินก็จะมี Message Box ขึ้นมาเตือน ดังรูปที่ 4.3



รูปที่ 4.3 แสดงการเพิ่มยอดเงินเกินลงในสมาร์ตการ์ด

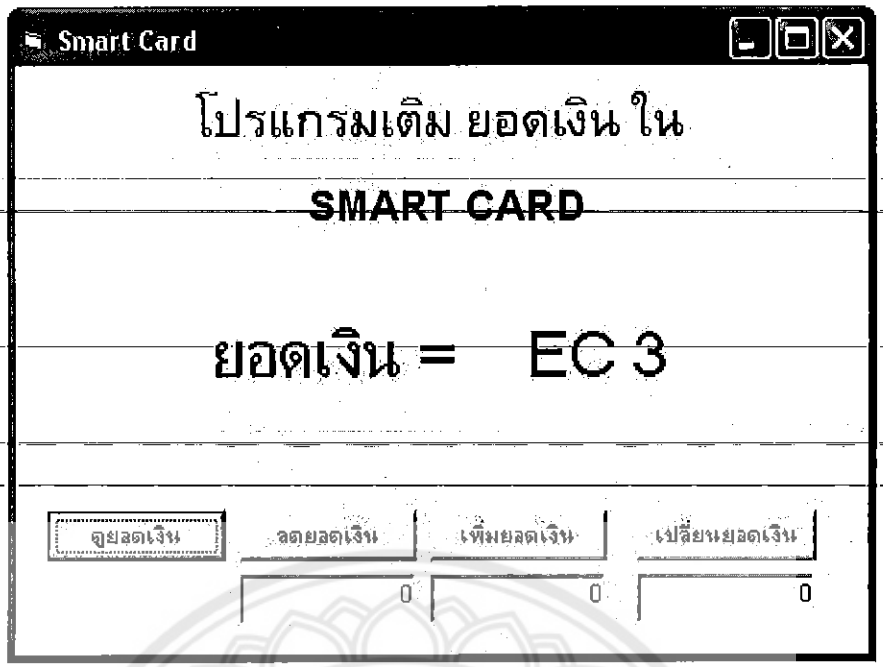
4.1.4 การลดยอดเงินลงในสมาร์ตการ์ด

ให้ใส่ยอดเงิน ลงในช่องลดยอดเงิน แล้วคลิกที่ปุ่มลดยอดเงิน แต่ถ้ามีการลดยอดเงินเกินยอดเงินที่มีอยู่ ก็จะไม่สามารถลดได้ และจะมี Message Box ขึ้นมาเตือน ดังรูปที่ 4.4

รูปที่ 4.4 แสดงการลดยอดเงินเกินลงในสมาร์ตการ์ด

4.1.5 การตรวจสอบค่า PSC ของสมาร์ตการ์ด

ก่อนที่จะมีการอ่านเขียนข้อมูลลงในบัตร โปรแกรมจะมีการตรวจสอบค่า PSC (Programmable Security Code) เพื่อใช้ในการตรวจสอบค่าให้ตรงกับค่า PSC ที่มีในบัตรก่อนจึงจะเขียนข้อมูลลงในบัตรได้ และ EC (Error Counter) เพื่อใช้ในการนับจำนวนครั้งที่ทำการตรวจสอบ Verify ค่า PSC โดยถ้าทำการตรวจสอบ Verify ค่า PSC ไม่ถูกต้องถึง 3 ครั้ง บัตรนี้จะเขียนข้อมูลไม่ได้อีกเลยทันที ในการตรวจสอบแต่ละครั้งจะแสดงข้อความเป็น EC 1, EC 2 และ EC 3 ตามลำดับ ดังรูปที่ 4.5



รูปที่ 4.5 แสดงการตรวจสอบค่า PSC ของสมาร์ตการ์ด

4.2 เครื่องอ่านและเขียนสมาร์ตการ์ด ณ ร้านขายสินค้าต่าง ๆ

ที่ร้านขายสินค้าต่างๆ จะมีอุปกรณ์ ดังนี้ เครื่องอ่านและเขียนสมาร์ตการ์ด 1 เครื่อง บอร์ดไมโครคอนโทรลเลอร์ MCS-51 อีก 1 เครื่อง ใช้การเชื่อมต่อแบบ RS-232 และสมาร์ตการ์ดอย่างน้อย 1 ใบ ดังรูปที่ 4.6 ณ จุดนี้ผู้ใช้บริการสามารถนำสมาร์ตการ์ดของตนเองมาชำระสินค้าและบริการต่าง ๆ ได้

รูปที่ 4.6 เครื่องคิดเงิน ณ ร้านขายสินค้าต่างๆ

จากรูปที่ 4.2 แสดงให้เห็นว่า เครื่องคิดเงินกำลังทำงานอยู่ โดยมีการเสียบสมาร์ตการ์ดที่เครื่องอ่านและเขียนสมาร์ตการ์ด และบอร์ดไมโครคอนโทรลเลอร์ได้ทำการอ่านยอดเงิน ที่มีอยู่ในบัตรเป็นจำนวนเงิน 5,000.00 บาท หลักการการทำงานของเครื่อง ณ จุดนี้ คือ จะสามารถหักลบยอดเงินได้อย่างเดียว โดยจะไม่สามารถเพิ่มยอดเงินได้ นั่นคือ ผู้ใช้สามารถนำสมาร์ตการ์ดมาชำระสินค้าหรือบริการให้อย่างเดียวจะไม่สามารถมาเติมเงินลงในบัตรได้ หลังจากที่มีผู้มาชำระสินค้าและบริการแล้วเครื่องก็จะเก็บยอดเงินที่ขายออกไปไว้ ใน RAM ของบอร์ดไมโครคอนโทรลเลอร์ MCS-51 หลังจากนั้นเจ้าของร้านก็จะนำบัตรของตนเองมาเสียบ เพื่อบันทึกยอดข้อมูลการขายลงไป (ไม่ใช่เป็นการเพิ่มยอดเงินลงไปโดยตรง แต่เป็นการเพิ่มจากยอดเงินที่เราขายไปจริงๆ) เพื่อที่จะนำไปขึ้นเงินกับทางบริษัท

4.2.1 KEY PAD บนบอร์ดไมโครคอนโทรลเลอร์

KEY PAD บนบอร์ดไมโครคอนโทรลเลอร์ มีทั้งหมด 20 ปุ่มดังรูปที่ 4.7 โดยแต่ละปุ่มจะมีหน้าที่ ที่แตกต่างกันออกไป เช่น ปุ่มที่ใช้ในการคีย์ยอดเงิน ปุ่มที่ใช้ในการลบยอดเงิน ปุ่มยกเลิกการคีย์ยอดเงิน ปุ่มที่ใช้ในการอ่านสมาร์ตการ์ด เป็นต้น

รูปที่ 4.7 KEY PAD บนบอร์ดไมโครคอนโทรลเลอร์

หน้าที่ของปุ่มต่างๆ

MON ใช้อ่านยอดเงินจากสมาร์ตการ์ด เมื่อเราเสียบสมาร์ตการ์ดของลูกค้าเข้าไปในเครื่องแล้วเมื่อกดจะแสดงยอดเงินที่มีอยู่ในสมาร์ตการ์ดออกทาง SEVEN-SEGMENT

ENT ใช้สำหรับผู้ขายสินค้าและบริการสามารถตรวจสอบยอดเงินที่ได้ขายไปแล้ว

INC ใช้สำหรับผู้ขายสินค้า และบริการบันทึกยอดเงินที่ได้ขายไปแล้วลงในสมาร์ตการ์ดของตนเอง ในที่นี้ไม่ใช่การเพิ่มยอดเงินลงในบัตรของตนเอง แต่เป็นการบันทึกยอดเงินการขาย

DEC ใช้ลบยอดเงินจากสมาร์ตการ์ด หลังจากที่กดปุ่มนี้ เครื่องจะให้เรากดยอดเงินที่ต้องการจะลบ

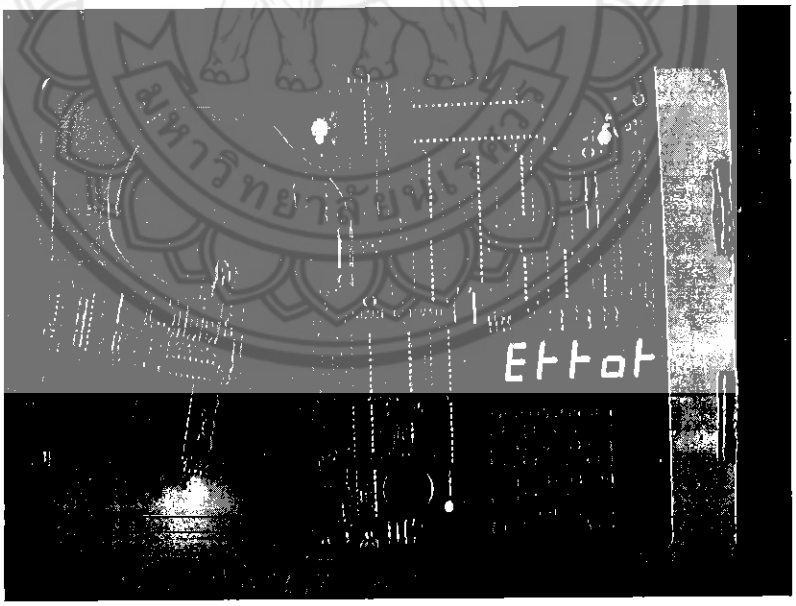
ปุ่ม C ปุ่มนี้ยกเลิกเมื่อกดยอดเงินที่ต้องการจะลบผิด เมื่อกดแล้วเราก็กดยอดเงินใหม่อีกครั้ง

ปุ่ม-A ทศนิยมเมื่อราคาสินค้าคิดเป็นสตางค์

0-9 ใช้ในการกดยอดเงิน

4.2.2 SEVEN SEGMENT แสดงข้อความว่า error

จากรูปที่ 4.8 เป็นการ error ที่เกิดจากการไม่ได้เสียบสมาร์ตการ์ดเข้าไปที่เครื่องอ่านและเขียนสมาร์ตการ์ด แล้วผู้ขายไปกดปุ่ม MON เพื่อที่จะดูยอดเงินในสมาร์ตการ์ด ซึ่งการ error นี้ อาจเกิดได้หลายสาเหตุ เช่น ไม่มีการเชื่อมต่อผ่านพอร์ต RS-232 อาจจะเป็นเพราะว่าสายขาด ไม่ได้เสียบพอร์ต RS-232 หรือพอร์ต RS-232 หลุด หรืออาจจะเกิดจากการนำสมาร์ตการ์ดที่ชำรุดมาใช้



รูปที่ 4.8 Seven Segment แสดงคำว่า error เนื่องจากไม่ได้เสียบสมาร์ตการ์ด

บทที่ 5
สรุปผลการทำปริญญานิพนธ์

5.1 สรุปผลการทำโครงงาน

ระบบ E-cash ที่พัฒนาขึ้นสามารถทำงานได้จริงในต้นทุนของระบบที่ต่ำกว่าระบบที่ใช้ในปัจจุบัน โดยพัฒนาขึ้นเพื่อให้เหมาะสมกับตลาดตลาดระดับล่างและการใช้จ่ายขนาดย่อม เช่น ร้านค้าต่าง ๆ ซึ่งได้นำสมาร์ตการ์ดชนิด Memory และใช้บอร์ดไมโครคอนโทรลเลอร์ MCS-51 มาใช้ในการพัฒนาระบบ

ตารางที่ 5.1 ข้อดีและข้อเสียของโครงงานเมื่อเปรียบเทียบกับระบบสมาร์ตเพิสต์

ข้อดี	ข้อเสีย
ใช้บัตรสมาร์ตการ์ดชนิด memory ทำให้ราคาบัตรถูกกว่า	ความปลอดภัยของบัตรยังน้อยกว่าสมาร์ตเพิสต์เนื่องจากใช้ (Com- Bi Card)
ระบบ ณ จุดขายใช้เครื่องไมโครคอนโทลเลอร์ทำให้ต้นทุนต่ำ	การทำรายการยังทำได้ช้ากว่าเนื่องจากระบบสมาร์ตเพิสต์สามารถเชื่อมต่อแบบ contactless ได้
มีการทำรายการแบบออฟไลน์ทำให้สะดวกในการย้ายที่และประหยัดกว่าระบบออนไลน์	ไม่สามารถตรวจสอบการทำรายการของบัตรได้สมาร์ตเพิสต์สารถตรวจสอบผ่านระบบออนไลน์ได้
ใช้งานง่ายไม่ต้องกรอกใบสมัครทำให้รวดเร็วและสะดวกกับผู้ใช้	ไม่มีการบันทึกข้อมูลผู้ใช้ แต่สมาร์ตเพิสต์เมื่อบัตรเสียหรือทำหายสามารถทำใหม่ได้โดยมียอดเงินเท่ากับยอดล่าสุด

จากการทดลองสามารถสรุปผลการทำโครงงานได้ดังนี้

1. ระบบในส่วนของจุดแลกบัตรพัฒนาขึ้นโดยใช้ภาษาวิชวลเบสิกจึงทำให้สามารถพัฒนาแก้ไขและปรับปรุงได้ง่าย และมีหน้าตาสะดวกต่อการใช้งาน ซึ่งสามารถทำงานร่วมกันกับเครื่องอ่านเขียนบัตรสมาร์ตการ์ดได้ดี

2. ระบบในส่วนของจุดร้านค้ามีต้นทุนต่ำเหมาะสมกับการใช้งานเนื่องจากใช้ไมโครคอนโทลเลอร์บอร์ด โดยใช้ภาษาซีในเขียนโปรแกรม ซึ่งเป็นภาษาเชิงวัตถุจึงทำให้ง่ายต่อการนำไปพัฒนาต่อ และสามารถทำงานร่วมกับเครื่องอ่านเขียนบัตรสมาร์ตการ์ดได้ดี

3. ระบบเหมาะกับการใช้จ่ายขนาดย่อม ดังนั้นจึงไม่เหมาะกับการใช้จ่ายที่มีราคาสูง เนื่องจากปัญหาเรื่องความปลอดภัยของบัตรที่น้อย และระบบไม่สามารถเช็คการทำรายการได้ เนื่องจากเป็นแบบออฟไลน์

4. ความปลอดภัยของตัวบัตรสมาร์ตการ์ด(memory type)จะแบ่งออกเป็นสองส่วนคือ ความปลอดภัยของค่า PSC บนบัตร และข้อมูลในบัตร ซึ่งเป็นข้อจำกัดส่วนหนึ่งในการออกแบบและพัฒนาระบบ

5. ระบบ E-cash ที่พัฒนาขึ้นในส่วนของผู้ซื้อและผู้ขายจะมีเงินอยู่ในการ์ด โดยการหมุนเวียนเงินในระบบจะมีเพียงการ์ดและจุดขายบัตรเท่านั้น

5.2 ปัญหาและข้อเสนอแนะ

5.2.1 ปัญหา

ณ. ร้านค้า

1) ปัญหาในเรื่องความปลอดภัยของ PSC ที่จุดร้านค้า

- วิธีการนำบัตรมาใส่ค่า PSC เพื่อหาค่า PSC ที่เครื่องเก็บไว้ซึ่งความยาวของ PSC คือ 3 ไบต์ ดังนั้นก็จะมีความเป็นไปได้ทั้งหมด 16777216 ค่า หากความน่าจะเป็นในการเจอเท่ากับครึ่งหนึ่งของการสุ่มก็จะเท่ากับ 8388608 ครั้ง บัตรหนึ่งจะใช้ในการสุ่มตัวเลขได้ 3 ครั้ง โดยครั้งที่ 3 จะทำให้บัตรเสียซึ่งอาจจะสุ่ม 2 ครั้งและ Verify PSC ใหม่โดยโดยการลบยอดเงินในบัตรก็จะไม่ทำให้บัตรเสียและรีเซตค่าจำนวนครั้งการนับ Error ใหม่ให้เท่ากับศูนย์ แต่วิธีนี้ก็ไม่เหมาะเนื่องจากจำนวนครั้งในการสุ่มที่เยอะ

- การหาโดยการอ่านข้อมูลที่ใช้ติดต่อกันผ่านทาง Serial port โดยส่วนนี้จะอยู่ระหว่างตัวไมโครคอนโทลเลอร์กับกับตัวอ่านเขียนบัตรสมาร์ตการ์ด โดยวิธีอ่านข้อมูลคือต้องเซ็ตการเชื่อมต่อกับตัวไมโครคอนโทลเลอร์ซึ่งจะส่งคำสั่งในการหาค่า PSC ก่อนการอ่านเขียนข้อมูล และผู้ทำจะต้องรู้คำสั่งของเครื่องอ่านเขียนบัตรสมาร์ตการ์ดด้วย

2) ปัญหาเครื่องไมโครคอนโทลเลอร์เมื่อไฟดับจะทำให้ยอดขายที่ขายได้เป็นศูนย์

ณ. จุดขายบัตร

1) พนักงานขายบัตรอาจจะทำบัตรขึ้นมาเองเพื่อขายในราคาถูกหรืออาจจะขโมยเงินที่ขายบัตรได้ซึ่งส่วนนี้จะทำให้บริษัทจำหน่ายบัตรขาดทุน

5.2.2 ข้อเสนอแนะ

ณ. ร้านค้า

1) เพิ่มความปลอดภัยของข้อมูลในบัตรให้สูงขึ้นและสามารถตรวจสอบการเขียนข้อมูลที่ไม่ถูกต้องได้

2) กรณียอดขายบนบัตรเป็นศูนย์เมื่อไฟดับก็อาจจะเก็บข้อมูลลงบนหน่วยความจำบอร์ดหรือใช้ตัวสำรองไฟ

ณ. จุดขายบัตร

1) เพิ่มฐานข้อมูลเพื่อบันทึกการทำรายการของพนักงานซึ่งไม่สามารถลบได้โดยกรณีแก้ไขข้อมูลก็จะบันทึกข้อมูลเดิมและข้อมูลที่แก้ไขเพื่อตรวจสอบการทำรายการของพนักงาน

5.3 แนวทางการพัฒนาต่อ

1. เนื่องจากสมาร์ตการ์ด SLE4442 ที่ใช้นี้มีความปลอดภัยของข้อมูลที่สูงและมีความยืดหยุ่นต่อการใช้งานดังนั้นจึงสามารถนำบัตรสมาร์ตการ์ดจากปริญญานิพนธ์นี้ไปพัฒนาเพิ่มเติมเป็นบัตรประจำตัวที่มีข้อมูลของผู้ถือบัตร เช่น บัตรประจำตัวนักเรียน นิสิต นักศึกษา เป็นต้น

2. สามารถทำการเพิ่มความปลอดภัยของระบบให้มากขึ้นได้

3. บอร์ดไมโครคอนโทรเลอร์ MCS-51 ที่ใช้ในปริญญานิพนธ์นี้เป็นบอร์ดสำเร็จดังนั้นจึงมีส่วนประกอบที่ไม่ได้ใช้ เช่น แรม หน่วยความจำบอร์ด ซึ่งในส่วนนี้อาจจะพัฒนาขึ้นมาใหม่โดยรวมให้เป็นอุปกรณ์ชิ้นเดียวกับเครื่องอ่านและเขียนสมาร์ตการ์ดได้ อุปกรณ์ชิ้นใดที่ไม่ใช้ก็ไม่ต้องซื้อมา ซึ่งจะทำให้ประหยัดต้นทุนได้อีก

4. สามารถเพิ่มในส่วนของการบันทึกการทำรายการ ณ จุดขายลงในฐานข้อมูล

5. อาจจะทำการพัฒนาโปรแกรมที่อยู่บนบอร์ดไมโครคอนโทรเลอร์ให้เขียนยอดขายลงบนบอร์ดได้

เอกสารอ้างอิง

[1] เลิศ แซ่ตั้ง . **เทคโนโลยีสมาร์ตการ์ด** กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2546

[2] อุดม รานอก . **ภาษาซีสำหรับควบคุมไมโครคอนโทรลเลอร์ MCS-51** พิมพ์ครั้งที่ 1 พระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537. นนทบุรี : ไอดีซีฯ . 2548

[3] ประภาพร ช่างไม้, สัจจะ จรัสรุ่งรวีวร . **คู่มือการเขียนภาษาซี ฉบับผู้เริ่มต้น** . พระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 . นนทบุรี : อินโฟเพรส, 2545

[4] ฉัททวุฒิ พืชผล . **คู่มือเรียน Visual Basic 6 (ฉบับปรับปรุง)** . พิมพ์ ครั้งที่ 11 พระราชบัญญัติลิขสิทธิ์ พ.ศ. 2537 . กรุงเทพฯ : โปรวิชั่น, 2547

[5] กฤษดา ใจเย็น, อรรถพล บุญยะโภคา และชัยวัฒน์ สิ้มพรจิตวิไล . **เรียนรู้ปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม** . กรุงเทพมหานคร : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2543

[6] วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตวิไล . **เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์** . กรุงเทพมหานคร : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด, 2543

ภาคผนวก

ภาคผนวก ก

โค้ดโปรแกรม

โค้ดโปรแกรมภาษาซี

1. main_pro.c

```
//=============================================
//======================== main_pro.c
//=============================================
#include <reg52.h>
#include <stdio.h>
#define enter_code 0x12

static unsigned char xdata segment_data _at_ 0xD000;
static unsigned char xdata digit_sw _at_ 0xF000;
static unsigned char xdata sw_row _at_ 0xE000;
static unsigned char key_code=0xFF,buffer_key_code=0xFF;
static unsigned char str_buf[20];
static unsigned char seg_buf[10];
static unsigned char scan_counter=0;
static unsigned char digit_sw_buffer;
static unsigned char str_1[100],str_2[100];
static unsigned char error_status=1,messege_status=0,showdot=0;
static unsigned char ok_status=0;
static unsigned long int1,int2,int3,int4,int5,int6,int7,int8,value_read;

//=============================================
// code for serial interrupts
#define rx_buffer_size 200 // define size for rx
   static unsigned char xdata rx_buffer[rx_buffer_size];
#if rx_buffer_size <256
   static unsigned char rx_wr_index=0,rx_rd_index=0,rx_counter=0;
#else
   static unsigned int rx_wr_index=0,rx_rd_index=0,rx_counter=0;
#endif
#define tx_buffer_size 100 // define size for tx buffer
   static unsigned char xdata tx_buffer[tx_buffer_size];
#if tx_buffer_size<256
    static unsigned char  tx_wr_index=0,tx_rd_index=0,tx_counter=0;
#else
     static unsigned int tx_wr_index=0,tx_rd_index=0,tx_counter=0;
#endif

//=============================================
static unsigned char rx_buffer_over_flow=0;
static unsigned char tx_buffer_over_flow=0;
```

```
static unsigned char send_status=0;
#include <func1.c>
#include <commu.c>
#include <output_lib.c>
#include <card_lib.c>
//=========================================================
void main()
{   unsigned char buffer_char1=0;
    unsigned long buffer_int1=0;

    delay_ms(1000);
    set_serial_port();
    setup_timer_0();
    ET0=1;
    ES=1;
    EA=1;
    TR0=1;
    out_segment(sprintf(str_buf,"s-card"));
    delay_ms(2000);
    out_segment(sprintf(str_buf,"000000"));

    while(1)
       {  buffer_char1=key_code;

          if(buffer_char1==0x13)
              {  beep();
                 read_data_from_card();
                 delay_ms(200);
              }
          if(buffer_char1==0x10)
              {  decr_money();
              }
          if(buffer_char1==0x11)
              {  incr_money();
              }
          if(buffer_char1==0x12)
              {  show_current();
              }
       } //end of while 1

} // end main
```

2. func1.c

```
//=========================================================
//======================== func1.c
//=========================================================
void delay_ms(unsigned int ms)
{   unsigned int i;
    unsigned int j;
    for(j=0;j<=ms;j++)
        {for(i=0;i<=320;i++);
        }
}

//=========================================================
void delay_us(unsigned int us)
{   unsigned int j;
    for(j=0;j<=us;j++)
        {
        }
}

//=========================================================
void set_serial_port()
    {   PCON=PCON|0x80; // setup scon in bit 7 (smod)
        SCON=0x52; // mode 1
        TMOD=0x20; // timer mode 2
        TH1=244;   //set br9600 for 11.0592*2
        TR1=1;
    }

//=========================================================
void setup_timer_0()
    {   char timer_mode;
        // timer mode
        timer_mode=0x01;   //16 bit internal gate
        TMOD=TMOD|timer_mode;
    }
```

3. commu.c

```
//=============================================
//=========================  commu.c
//=============================================
void FIFO_SERVICE() interrupt 4  // serial interrupt service
   {
      if(RI) // if rx occer
         {  RI=0;
            rx_buffer[rx_wr_index]=SBUF;
      if (++rx_wr_index == rx_buffer_size)
         {  rx_wr_index=0;// chk for ring buffer return0 to index
         }
      if (++rx_counter >= rx_buffer_size)// chk for receive over flow
         {  rx_counter=0;
            rx_buffer_over_flow=1;
         }
    }// end of rx interrupts

 if(TI) // if tx occer
    {  TI=0;
       if (tx_counter>0) // chk for tx buffer if not empty
          {  --tx_counter;
             SBUF=tx_buffer[tx_rd_index];
             if(++tx_rd_index == tx_buffer_size) // chk index ring buffer return0
                {  tx_rd_index=0;
                }
          }
       else
          {  send_status=0;
          }
    } // end of tx_service

} // end of serial interrupts

//=============================================
char get_serial() // function for get serial from FIFO
{  char buffer;
   while (rx_counter==0); // wait for data no empty
   buffer=rx_buffer[rx_rd_index]; // hook data from rx buffer
   if (++rx_rd_index == rx_buffer_size) // increse index and chk for return 0 ring buffer
      {  rx_rd_index=0;
      }
   ES=0;
   --rx_counter;
   ES=1;
   return buffer;
}

//=============================================
void put_serial(char c)
{  while (tx_counter == tx_buffer_size); // chk and wait for tx_buffer_full
   ES=0;
```

```
    if ((tx_counter > 0) || (send_status==1))
        {   tx_buffer[tx_wr_index]=c;
            if (++tx_wr_index == tx_buffer_size)
                {   tx_wr_index=0;
                }
            ++tx_counter;
        }
    else
        {   SBUF=c;
            send_status=1;
        }
    ES=1;
}

//=================================================
void out_serial(unsigned char dat)
    {   unsigned char i;
        for(i=1;i<=dat;i++)
            {   put_serial(str_buf[i-1]);
            }
    }

//=================================================
unsigned char make_hex(unsigned char dat)
{   unsigned char buffer1=0;
    if(dat=='1'){buffer1=1;}
    if(dat=='2'){buffer1=2;}
    if(dat=='3'){buffer1=3;}
    if(dat=='4'){buffer1=4;}
    if(dat=='5'){buffer1=5;}
    if(dat=='6'){buffer1=6;}
    if(dat=='7'){buffer1=7;}
    if(dat=='8'){buffer1=8;}
    if(dat=='9'){buffer1=9;}
    if(dat=='A'){buffer1=10;}
    if(dat=='B'){buffer1=11;}
    if(dat=='C'){buffer1=12;}
    if(dat=='D'){buffer1=13;}
    if(dat=='E'){buffer1=14;}
    if(dat=='F'){buffer1=15;}
    if(dat=='0'){buffer1=0;}
    if(dat=='a'){buffer1=10;}
    if(dat=='b'){buffer1=11;}
    if(dat=='c'){buffer1=12;}
    if(dat=='d'){buffer1=13;}
    if(dat=='e'){buffer1=14;}
    if(dat=='f'){buffer1=15;}
    return buffer1;
}
```

```
//=============================================
void chk_serial()
{  static unsigned char index_1,index_2;
    unsigned char buffer_char1;
    unsigned char i;
    if(index_1>99){index_1=0;}
    if(index_2>99){index_2=0;}

    buffer_char1=get_serial();

    if(buffer_char1!=13) // no return
        {  str_1[index_1]=buffer_char1;
           index_1++;
        }
    if(buffer_char1==13) // have return
        {  for(i=0;i<=99;i++){str_2[i]=0;} // clear string 2
           for(i=0;i<=99;i++){str_2[i]=str_1[i];} //move string 2 to string 1
           for(i=0;i<=99;i++){str_1[i]=0;} // clear str 1
           index_1=0; // stert new receive at 0
           messege_status=0;

           if(str_2[0] == 'O' && str_2[1]=='K')
              {  error_status=0;
                 messege_status=1;
                 ok_status=1;
              }
           if(str_2[0] == 'E' && str_2[1]=='R')
              {  error_status=1;
                 messege_status=1;
              }
           if(str_2[0] == 'E' && str_2[1]=='1')
              {  error_status=1;
                 messege_status=1;
              }
           if(str_2[0] == 'E' && str_2[1]=='2')
              {  error_status=1;
                 messege_status=1;
              }
           if(str_2[0] == 'E' && str_2[1]=='3')
              {  error_status=1;
                 messege_status=1;
              }
           if(error_status==1)
              {  //out_segment(sprintf(str_buf," ERROR"));
              }
           if(error_status==0 && messege_status==0) // no error no messege
              {  int1=make_hex(str_2[0]);
                 int2=make_hex(str_2[1]);
                 int3=make_hex(str_2[2]);
                 int4=make_hex(str_2[3]);
                 int5=make_hex(str_2[4]);
```

```
                int6=make_hex(str_2[5]);
                int7=make_hex(str_2[6]);
                int8=make_hex(str_2[7]);
        value_read=(int8*1)+(int7*16)+(int6*256)+(int5*4096)+(int4*65536)+(int3*10485
        76)+(int2*16777216)+(int1*268435456); //make int value
                }
            }
        }
```

4. card_lib.c

```
//=============================================
//========================  card_lib.c
//=============================================
unsigned long current_money=0;
void chk_vsc()
{  delay_ms(100);
   ok_status=0;
   out_serial(sprintf(str_buf,":6FFFFFF%c",0x0d));
   beep();     delay_ms(2000);
   beep();
}

//=====================================================
void read_data_from_card()
{  unsigned int loop=0;
   showdot = 0;
   delay_ms(5);
   ok_status=0;
   out_segment(sprintf(str_buf,"  REAd"));beep();
   out_serial(sprintf(str_buf,":6FFFFFF%c",0x0d));
   delay_ms(100);
   out_serial(sprintf(str_buf,":6FFFFFF%c",0x0d));
   delay_ms(300);

   if(ok_status==0 || ok_status==2) // error
      {  out_segment(sprintf(str_buf," ERROR"));
         beep();
         delay_ms(2000);
         showdot = 1;
         out_segment(sprintf(str_buf,"   000"));
         beep();
      }
   if(ok_status==1) // chk_ ok
      {  out_segment(sprintf(str_buf,"YES   "));
         out_serial(sprintf(str_buf,":221%c",0x0d));
         beep();
         delay_ms(500);
         showdot = 1;

         //show condition
         if (value_read<100)
            {  out_segment(sprintf(str_buf,"   0%2lu",value_read));
            }
         if (value_read<10)
            {  out_segment(sprintf(str_buf,"   00%1lu",value_read));
                  }
         if (value_read==0)
            {  out_segment(sprintf(str_buf,"   000"));
            }
```

```
                if (value_read>=100)
                    {   out_segment(sprintf(str_buf,"%6lu",value_read));
                    }
                beep();
            }
    }

    //=================================================
    void decr_money()
        {   unsigned long buffer_int1=0,buffer_int2;
            unsigned char out_loop_flag=0;
            buffer_int1=get_number();          //Get Amount of money to delete
            showdot =1;
        if(buffer_int1==4294967295)
            {   out_loop_flag=1;
                out_segment(sprintf(str_buf,"   000")); // return 0
                read_data_from_card();
            }
         if(out_loop_flag==0) // continue
            {   read_data_from_card();

                if( ok_status!=1) // out ;oop
                    {   out_loop_flag=1;
                        out_segment(sprintf(str_buf,"   000")); // return 0
                    }
                if(ok_status==1)
                    {   buffer_int2=value_read;
                        if(buffer_int2<buffer_int1)
                            {   //show condition
                                if (value_read<100)
                                    {   out_segment(sprintf(str_buf,"   0%2lu",value_read));
                                    }
                                if (value_read<10)
                                    {   out_segment(sprintf(str_buf,"   00%1lu",value_read));
                                    }
                                if (value_read==0)
                                    {   out_segment(sprintf(str_buf,"   000"));
                                    }
                                if (value_read>=100)
                                    {   out_segment(sprintf(str_buf,"%6lu",value_read));
                                    }
                            }
                        if(buffer_int2>=buffer_int1)
                            {   current_money+=buffer_int1;     //Calculate Total Money
                                out_serial(sprintf(str_buf,":321%08lX%c",(buffer_int2-
buffer_int1),0x0d));
                                beep();
                                delay_ms(500);
                                read_data_from_card();
                            }
                    }
            }
    }
```

```
//=============================================
void incr_money()
   {  unsigned long buffer_int1=0,buffer_int2=0;
      unsigned char out_loop_flag=0;
      if(out_loop_flag==0) // continue
         {  read_data_from_card();
            if( ok_status!=1) // out ;oop
               {  out_loop_flag=1;
                  out_segment(sprintf(str_buf,"   000")); // return 0
               }
            if(ok_status==1) //card in ok
               {  buffer_int2=value_read;
                  buffer_int1=500000-value_read;
                  showdot = 1;
                  if(value_read>=500000 ) // full card
                     {  buffer_int1=0;  // does not increat
                        beep();
                        delay_ms(500);
                        showdot=0;
                        out_segment(sprintf(str_buf,"FullCd"));
                     }

                  if(buffer_int1<=current_money) // if money more than store
                     {  current_money=current_money-buffer_int1;
                     }
                  else   // can be store all
                     {  buffer_int1=current_money;
                        current_money=0;
                     }
                  // increment money
                  showdot=1;

out_serial(sprintf(str_buf,":321%08lX%c",(buffer_int2+buffer_int1),0x0d));
                  beep();
                  delay_ms(500);
                  read_data_from_card();
               }
         }
   }

void show_current()
   {  showdot = 1;
      //show condition
      if (current_money<100)
         {  out_segment(sprintf(str_buf,"   0%2lu",current_money));
         }
      if (current_money<10)
         {  out_segment(sprintf(str_buf,"   00%1lu",current_money));
         }
      if (current_money==0)
         {  out_segment(sprintf(str_buf,"   000"));
         }
      if (current_money>=100)
```

```
            {   out_segment(sprintf(str_buf,"%6lu",current_money));
            }
        beep();
        delay_ms(500);
    }
```

5. output_lib.c

```
//=============================================================
//========================== output_lib.c
//=============================================================
void out_segment_ram(unsigned char dat)
    {   segment_data=dat;
    }
//=============================================================
unsigned char bit_test(unsigned char buffer1,unsigned char buffer2)
    {   return ((buffer1>>buffer2)&0x01);
    }

//=============================================================
unsigned char decode_segment(unsigned char dat)
    {   unsigned char buffer;
        switch(dat)
            {   case '0':buffer=0x3f;break;
                case '1':buffer=0x06;break;
                case '2':buffer=0x5b;break;
                case '3':buffer=0x4f;break;
                case '4':buffer=0x66;break;
                case '5':buffer=0x6d;break;
                case '6':buffer=0x7d;break;
                case '7':buffer=0x07;break;
                case '8':buffer=0x7f;break;
                case '9':buffer=0x6f;break;
                case 'a':buffer=0x77;break;
                case 'A':buffer=0x77;break;
                case 'b':buffer=0x7c;break;
                case 'B':buffer=0x7c;break;
                case 'C':buffer=0x39;break;
                case 'c':buffer=0x58;break;
                case 'd':buffer=0x5e;break;
                case 'D':buffer=0x3f;break;
                case 'E':buffer=0x79;break;
                case 'e':buffer=0x79;break;
                case 'F':buffer=0x71;break;
                case 'f':buffer=0x71;break;
                case 'h':buffer=0x76;break;
                case 'H':buffer=0x76;break;
                case 'i':buffer=0x06;break;
                case 'I':buffer=0x06;break;
                case 'o':buffer=0x5c;break;
                case 'O':buffer=0x5c;break;
                case '-':buffer=0x40;break;
                case 't':buffer=0x78;break;
                case 'T':buffer=0x78;break;
                case 's':buffer=0x6d;break;
                case 'S':buffer=0x6d;break;
```

```
                case 'L':buffer=0x38;break;
                case 'l':buffer=0x38;break;
                case 'n':buffer=0x54;break;
                case 'N':buffer=0x54;break;
                case 'p':buffer=0x73;break;
                case 'P':buffer=0x73;break;
                case 'q':buffer=0x67;break;
                case 'Q':buffer=0x67;break;
                case 'r':buffer=0x70;break;
                case 'R':buffer=0x70;break;
                case 'u':buffer=0x1c;break;
                case 'U':buffer=0x3e;break;
                case 'G':buffer=0x3d;break;
                case 'g':buffer=0x3d;break;
                case 'y':buffer=0x6e;break;
                case 'Y':buffer=0x6e;break;
                case '.':buffer=0x80;break;
                default:buffer=0x00;
            }
            return buffer;
    }
//=============================================
void on_digit(unsigned char dat)
    {   digit_sw_buffer=digit_sw_buffer&0xF0;
        digit_sw_buffer=digit_sw_buffer|dat;
        digit_sw=digit_sw_buffer;
     }
//=============================================
void out_digit(unsigned char dat)
    {   out_segment_ram(dat);
    }
//=============================================
void scan_segment()
    {   unsigned char buffer_char1,buffer_char2;
        scan_counter++;
        if(scan_counter>5)
            {   scan_counter=0;
            }
        buffer_char1=seg_buf[scan_counter];
        out_segment_ram(0);
        on_digit(scan_counter);
        //out_digit(buffer_char1);
        out_segment_ram(buffer_char1);

        if(scan_counter==0)
            {   buffer_key_code=0xFF;
                buffer_char2=sw_row;
                buffer_char2=~buffer_char2;
                buffer_char2=buffer_char2&0x0F;

               if(buffer_char2==0x01){buffer_key_code=0x0c;}
               if(buffer_char2==0x02){buffer_key_code=0x08;}
```

```
            if(buffer_char2==0x04){buffer_key_code=0x04;}
            if(buffer_char2==0x08){buffer_key_code=0x00;}
          }
         if(scan_counter==1)
          {  //key_code=0xFF;
            buffer_char2=sw_row;
            buffer_char2=~buffer_char2;
            buffer_char2=buffer_char2&0x0F;

            if(buffer_char2==0x01){buffer_key_code=0x0d;}
            if(buffer_char2==0x02){buffer_key_code=0x09;}
            if(buffer_char2==0x04){buffer_key_code=0x05;}
            if(buffer_char2==0x08){buffer_key_code=0x01;}
          }
        if(scan_counter==2)
          {  //key_code=0xFF;
            buffer_char2=sw_row;
            buffer_char2=~buffer_char2;
            buffer_char2=buffer_char2&0x0F;

            if(buffer_char2==0x01){buffer_key_code=0x0e;}
            if(buffer_char2==0x02){buffer_key_code=0x0a;}
            if(buffer_char2==0x04){buffer_key_code=0x06;}
            if(buffer_char2==0x08){buffer_key_code=0x02;}
          }
        if(scan_counter==3)
          {  //key_code=0xFF;
            buffer_char2=sw_row;
            buffer_char2=~buffer_char2;
            buffer_char2=buffer_char2&0x0F;

            if(buffer_char2==0x01){buffer_key_code=0x0F;}
            if(buffer_char2==0x02){buffer_key_code=0x0B;}
            if(buffer_char2==0x04){buffer_key_code=0x07;}
            if(buffer_char2==0x08){buffer_key_code=0x03;}
          }
         if(scan_counter==4)
          {  //key_code=0xFF;
            buffer_char2=sw_row;
            buffer_char2=~buffer_char2;
            buffer_char2=buffer_char2&0x0F;
            if(buffer_char2==0x01){buffer_key_code=0x13;}
            if(buffer_char2==0x02){buffer_key_code=0x12;}
            if(buffer_char2==0x04){buffer_key_code=0x11;}
            if(buffer_char2==0x08){buffer_key_code=0x10;}
            key_code=buffer_key_code;
          }

      }
```

```
//=============================================
unsigned char dotposition=3;
void out_segment(unsigned long dat)
   {  unsigned char i,j,k;
      dat++;
      k=0;
      for(i=0;i<=5;i++)
         {  j=str_buf[i];
            seg_buf[i]=decode_segment(j);
            if(showdot == 1)  // showdot
               {  seg_buf[dotposition]=seg_buf[dotposition]+ 0x80;
               }
         }
   }
//=============================================
void beep()
   {  unsigned char i,j=0;
      for(i=0;i<=50;i++)
         {  digit_sw_buffer=digit_sw_buffer|0x40;
            digit_sw=digit_sw_buffer;
            delay_us(100);
            digit_sw_buffer=digit_sw_buffer&0xBF;
            digit_sw=digit_sw_buffer;
            delay_us(100);
         }
   }

//=============================================
void t0_service() interrupt 1
   {  TR0=0;
      TF0=0;
      scan_segment();
      if(rx_counter!=0)
         {  chk_serial();
         }
      TH0=0xEA;
      TL0=0x00;
      TR0=1;
   }

//=============================================
 unsigned long get_number()
    {  unsigned long buffer_int1=0;
       unsigned char buffer_sw;
       unsigned char out_loop_flag=0;
       unsigned char pressdot=0,press_count=0;
       buffer_sw=key_code;
       showdot = 0;
       out_segment(sprintf(str_buf,"     0"));
       delay_ms(100);
       beep();
```

```
while(out_loop_flag==0)
    {   buffer_sw=key_code;delay_ms(20);
        if(buffer_sw>=0 && buffer_sw<=9 ) // number occer
            {
                if (pressdot==0)
                    {   buffer_int1=(buffer_int1*10)+buffer_sw;
                        // check overload number
                        if(buffer_int1>9999 || press_count>2 )
                            {   buffer_int1=(buffer_int1-buffer_sw)/10;
                            }
                    }
                if(pressdot==1 && press_count<2) //after pressdot
                    {   press_count=press_count+1;
                        buffer_int1=(buffer_int1*10)+buffer_sw;
                        if(buffer_int1>999999) //check overload number
                            {   buffer_int1=(buffer_int1-buffer_sw)/10;
                            }
                        if(buffer_int1>99999&&press_count<2)
                            {   press_count=3;
                                dotposition=dotposition-1;
                            }
                        if(press_count<3 ) // move dot position
                            { dotposition=dotposition-1;
                            }
                        showdot=1;
                    }

                    // show number
                    if (buffer_int1<10 && press_count==2)
                        {   out_segment(sprintf(str_buf,"    0%1lu",buffer_int1));
                            //show zero after dot
                        }
                    else
                        { out_segment(sprintf(str_buf,"%6lu",buffer_int1));
                        }
                    beep();
                    delay_ms(200);
            }
            if(buffer_sw==0x0a && pressdot==0) // press dot
                {   showdot=1;
                    pressdot=1;
                    dotposition=5;
                    out_segment(sprintf(str_buf,"%6lu",buffer_int1));
                    beep();
                    delay_ms(200);
                } // end pressdot
            if(buffer_sw==enter_code || buffer_sw==0x10) //decrease money
                {
                    if (pressdot==0)
                        {   buffer_int1=buffer_int1*100;
                        }
```

```
                    if (press_count==1)
                        {   buffer_int1=buffer_int1*10;
                        }
                        dotposition=3;
                        pressdot=0;
                        out_loop_flag=1;
                        beep();
                }

            if(buffer_sw==0x0d)//delete number
                {   if(buffer_int1>9)
                        {   buffer_int1=(buffer_int1-buffer_sw)/10;
                            buffer_int1=buffer_int1+1;
                        }
                    else {buffer_int1=0;}

                    if(press_count>0) //decrese press count
                        {   press_count=press_count-1;
                            dotposition=dotposition+1;
                        }
                    else
                        {   pressdot=0;
                            showdot=0;
                        }
                        out_segment(sprintf(str_buf,"%6lu",buffer_int1));
                        beep();
                        delay_ms(200);
                }
            if(buffer_sw==0x0c) // clear number
                {   buffer_int1=0;
                    showdot=0;
                    pressdot=0;
                    press_count=0;
                    out_segment(sprintf(str_buf,"     0"));
                     beep();
                     delay_ms(200);
                }
             if(buffer_sw==0x0f) // cancle sw
                {   out_loop_flag=1;
                    dotposition=3;
                    beep();
                    buffer_int1=4294967295;
                }
        }
        return buffer_int1;
}
```

โค้ดโปรแกรมภาษาวิชวลเบสิก

```
Option Explicit
Public timeout As Byte
Public str_buffer1 As String
Public str_buffer2 As String
Public command_status1 As Byte
Public command_status2 As Byte
Public error_status As Byte
Public messege_status As Byte
Public minus_status As Byte
Private Sub Command1_Click()
command_status1 = 1
messege_status = 0
error_status = 0
Text3.Text = ""
MSComm1.Output = ":6FFFFFF"
MSComm1.Output = Chr(13)
Sleep (1000)
MSComm1.Output = ":221"
MSComm1.Output = Chr(13)
command_status1 = 1  ' read money value
End Sub

Private Sub Command2_Click()  'Increase Money
Dim buffer_int1 As Currency
Dim buffer_int2 As Currency
Dim buffer_int3 As Currency  ' New Money Amount

Dim str2 As String

buffer_int1 = Val(Label2.Caption)  ' Total Money
buffer_int2 = Val(Text1.Text) ' Increase Money
buffer_int1 = buffer_int1 * 100
buffer_int2 = buffer_int2 * 100
If buffer_int2 <= buffer_int1 Then
buffer_int3 = buffer_int1 + buffer_int2
'Check Money Over 5000
If buffer_int3 > 500000 Then
    MsgBox ("ไม่สามารถเติมเงินได้ เนื่องจากยอดเงินเกิน 5000 บาท")
    Exit Sub
End If
If buffer_int3 > &HFFFFFFF Then
           str2 = Hex(buffer_int3)
           End If
If buffer_int3 <= &HFFFFFFF Then
           str2 = Hex(buffer_int3)
           str2 = "0" & str2
           End If
If buffer_int3 <= &HFFFFFF Then
           str2 = Hex(buffer_int3)
           str2 = "00" & str2
           End If
If buffer_int3 <= &HFFFFF Then
           str2 = Hex(buffer_int3)
           str2 = "000" & str2
           End If
If buffer_int3 <= 65535 Then
           str2 = Hex(buffer_int3)
           str2 = "0000" & str2
           End If
If buffer_int3 <= &HFFF Then
           str2 = Hex(buffer_int3)
           str2 = "00000" & str2
           End If
```

```
If buffer_int3 <= &HFF Then
            str2 = Hex(buffer_int3)
            str2 = "000000" & str2
            End If
If buffer_int3 <= &HF Then
            str2 = Hex(buffer_int3)
            str2 = "0000000" & str2
            End If

Text4.Text = str2
' verify
MSComm1.Output = ":6FFFFFF"
MSComm1.Output = Chr(13)
Sleep (500)

MSComm1.Output = ":321"
MSComm1.Output = str2
MSComm1.Output = Chr(13)
Sleep (500)

MSComm1.Output = ":221"
MSComm1.Output = Chr(13)
command_status1 = 1
End If    ' End of Check Money

End Sub

Private Sub Command3_Click()    ' chang money
Dim buffer_int3 As Currency
Dim str2 As String
' Check Value in TextBox2
If Val(Text2.Text) <= 5000 Then
    buffer_int3 = Val(Text2.Text)
    buffer_int3 = buffer_int3 * 100
Else
        MsgBox ("Error , Over Value.")
    Exit Sub
End If

If buffer_int3 > &HFFFFFFF Then
            str2 = Hex(buffer_int3)
            End If
If buffer_int3 <= &HFFFFFFF Then
            str2 = Hex(buffer_int3)
            str2 = "0" & str2
            End If
If buffer_int3 <= &HFFFFFF Then
            str2 = Hex(buffer_int3)
            str2 = "00" & str2
            End If
If buffer_int3 <= &HFFFFF Then
            str2 = Hex(buffer_int3)
            str2 = "000" & str2
            End If
If buffer_int3 <= 65535 Then
            str2 = Hex(buffer_int3)
            str2 = "0000" & str2
            End If
If buffer_int3 <= &HFFF Then
            str2 = Hex(buffer_int3)
            str2 = "00000" & str2
            End If
If buffer_int3 <= &HFF Then
            str2 = Hex(buffer_int3)
            str2 = "000000" & str2
            End If
If buffer_int3 <= &HF Then
            str2 = Hex(buffer_int3)
            str2 = "0000000" & str2
```

```
                End If

            Text4.Text = str2
            ' verify
            If Val(Text2.Text) < 0 Then
                    MsgBox ("Error , negative Value.")
            Else
            MSComm1.Output = ":6FFFFFF"
            MSComm1.Output = Chr(13)
            Sleep (500)

            MSComm1.Output = ":321"
            MSComm1.Output = str2
            MSComm1.Output = Chr(13)
            Sleep (500)

            MSComm1.Output = ":221"
            MSComm1.Output = Chr(13)
            command_status1 = 1
            Exit Sub
            End If
            End Sub


            Private Sub Command4_Click() 'Decrease Money
            Dim buffer_int1 As Currency
            Dim buffer_int2 As Currency
            Dim buffer_int3 As Currency  ' New Money Amount

            Dim str2 As String

            buffer_int1 = Val(Label2.Caption)  ' Total Money
            buffer_int2 = Val(Text6.Text) ' Decrease Money
            buffer_int1 = buffer_int1 * 100
            buffer_int2 = buffer_int2 * 100

            If buffer_int2 <= buffer_int1 Then 'Check Correct Decrease Money
            buffer_int3 = buffer_int1 - buffer_int2

            If buffer_int3 > &HFFFFFFF Then
                        str2 = Hex(buffer_int3)
                        End If
            If buffer_int3 <= &HFFFFFFF Then
                        str2 = Hex(buffer_int3)
                        str2 = "0" & str2
                        End If
            If buffer_int3 <= &HFFFFFF Then
                        str2 = Hex(buffer_int3)
                        str2 = "00" & str2
                        End If
            If buffer_int3 <= &HFFFFF Then
                        str2 = Hex(buffer_int3)
                        str2 = "000" & str2
                        End If
            If buffer_int3 <= 65535 Then
                        str2 = Hex(buffer_int3)
                        str2 = "0000" & str2
                        End If
            If buffer_int3 <= &HFFF Then
                        str2 = Hex(buffer_int3)
                        str2 = "00000" & str2
                        End If
            If buffer_int3 <= &HFF Then
                        str2 = Hex(buffer_int3)
                        str2 = "000000" & str2
                        End If
            If buffer_int3 <= &HF Then
                        str2 = Hex(buffer_int3)
                        str2 = "0000000" & str2
```

```
                End If

            Text4.Text = str2
            ' verify
            MSComm1.Output = ":6FFFFFF"
            MSComm1.Output = Chr(13)
            Sleep (500)

            MSComm1.Output = ":321"
            MSComm1.Output = str2
            MSComm1.Output = Chr(13)
            Sleep (500)

            MSComm1.Output = ":221"
            MSComm1.Output = Chr(13)
            command_status1 = 1
            Else
            MsgBox ("ไม่สามารถลบยอดเงินได้ เนื่องจากจำนวนเงินไม่พอ")
            End If ' End of Check Money
            End Sub

            Private Sub delay_timer_Timer()
            timeout = 1
            End Sub

            Private Sub Form_Load()
            MSComm1.Settings = "9600,n,8,1"
            MSComm1.InputLen = 1
            MSComm1.InputMode = comInputModeText
            MSComm1.OutBufferSize = 1024
            MSComm1.InBufferSize = 1024
            MSComm1.RThreshold = 1
            MSComm1.SThreshold = 1
            MSComm1.PortOpen = True
            Text4.Text = "0"
            Sleep (1)
            error_status = 0
            End Sub


            Private Sub Timer1_Timer()
            Dim a As String * 1
            Dim buffer_int1 As Currency
            If MSComm1.InBufferCount Then
                a = MSComm1.Input
                If a = Chr(13) Then
                      str_buffer1 = Text3.Text
                      Text3.Text = ""
                      Text5.Text = str_buffer1
                      '================================
                      '================================
                      '== chk for read money
                      '================================
                      messege_status = 0
                      If command_status1 = 1 Then
                        If str_buffer1 = "ER" Then
                            Label2.Caption = "CARD ERROR"
                            error_status = 1
                            messege_status = 1
                            End If
                        If str_buffer1 = "OK" Then
                            Label2.Caption = "CARD OK"
                            error_status = 0
                            messege_status = 1
                            End If
                        If str_buffer1 = "E1" Then
                            Label2.Caption = "EC 1"
                            error_status = 1
```

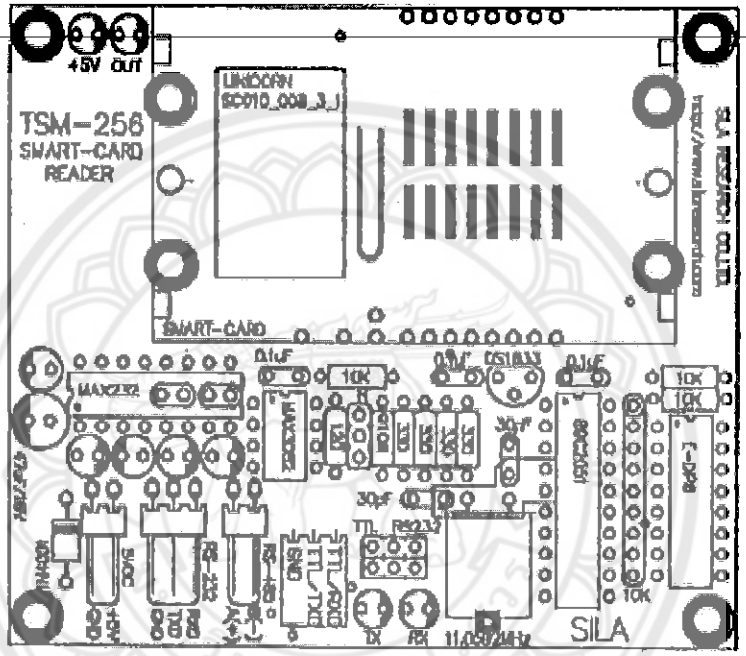
```
                messege_status = 1
                End If
            If str_buffer1 = "E2" Then
                Label2.Caption = "EC 2"
                error_status = 1
                messege_status = 1
                End If
            If str_buffer1 = "E3" Then
                Label2.Caption = "EC 3"
                error_status = 1
                messege_status = 1
                End If
            If messege_status = 0 And error_status = 0 Then
                 str_buffer2 = Left(str_buffer1, 8)
                 str_buffer2 = "&H" & str_buffer2
                 buffer_int1 = Val(str_buffer2)
                 buffer_int1 = buffer_int1 / 100
                 Label2.Caption = (buffer_int1 & " °Ò·")
                 Text2.Text = buffer_int1

                End If
           End If
           '================================
           '================================
           End If
      If a <> Chr(13) Then
           Text3.Text = Text3.Text & a
           End If

  End If
End Sub
```

ภาคผนวก ข

ข้อมูลอุปกรณ์

TSM-256

Smart Card Read/Write Board



TSM-256 เป็นบอร์ดที่ใช้เพื่อการอ่านและเขียนข้อมูลกับบัตร Smart Card เบอร์ SLE4442 รับคำสั่งติดต่อสื่อสารผ่าน ทางRS232 หรือ RS485 โดย RS485 สามารถต่อพ่วงกันเป็น Network ได้สูงสุดถึง 8 บอร์ด เลือกใช้ค่าความเร็วการสื่อสารได้ตั้งแต่ 2400-19200 ชุดคำสั่งที่ใช้งานเป็นแบบ ASCII สะดวกใช้งานง่าย สามารถเขียนข้อมูลถาวรลงไปในบัตรได้ มีระบบตรวจสอบค่า PSC พร้อม Error Counter ให้ความปลอดภัยกับข้อมูลสูง ประยุกต์ใช้ต่อเข้ากับ MCU ได้โดยตรงด้วยชุดต่อแบบ TTL Level Socket ที่ใช้เสียบบัตรคุณภาพดีเสียบค้างไว้ได้แน่นแต่ดึงออกง่าย TSM-256 บอร์ดที่เหมาะกับการประยุกต์ใช้งานที่ต้องการให้ความปลอดภัยของข้อมูลสูงของบัตร Smart Card

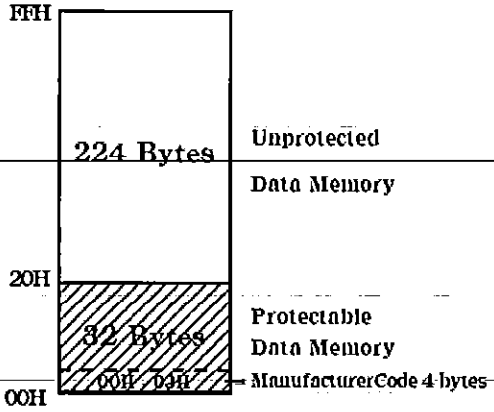
คุณสมบัติทั่วไป

1. ทำงานด้วยไมโครคอนโทรลเลอร์เบอร์ 89C4051
2. เขียนและอ่านบัตร Smart Card เบอร์ SLE 4442 ของบริษัท SIEMENS ตามมาตรฐาน ISO 7816
3. รับคำสั่งจากเครื่องคอมพิวเตอร์ PC เพื่อเขียนและอ่านข้อมูลบัตร Smart Card ได้ โดยผ่านพอร์ทสื่อสารอนุกรม RS232 (MAX232) มีสายเชื่อมต่อให้พร้อมหรือจะประยุกต์ต่อกับ MCU ได้โดยตรงด้วยจุดต่อ RS232แบบTTL Level คุณสมบัติการสื่อสารคือ Parity=None, Data=8, StopBit=1 กำหนด Baudrate ได้ตั้งแต่ 2400 - 19200
4. มีระบบการสื่อสารแบบ Network สามารถเชื่อมต่อกันได้สูงสุด 8 บอร์ดด้วยพอร์ทสื่อสารอนุกรมRS485 (75176 เป็นOPTION) ส่งสัญญาณได้ไกลถึง 1.2 k.m. รับคำสั่งติดต่อสื่อสารกับคอมพิวเตอร์ได้ด้วยการแปลง สัญญาณ RS232 จาก PC เป็น RS485 ก่อนด้วย CON-485 ของศิลา
5. DIP-SWITCH 8 หลักใช้สำหรับตั้งค่าความเร็วการสื่อสารและตั้งค่า Address กรณีที่ใช้ระบบการสื่อสารแบบ Network
6. จัมพ์เปอร์ใช้ปรับเลือกว่าจะใช้ RS232 แบบ TTL Level หรือใช้กับ RS232 ที่ต่อเข้ากับ PC
7. ขั้วต่อสีขาว 3 Pin สำหรับ RS232 และขั้วต่อสีขาว 2 Pin สำหรับ RS485
8. มี LED แสดง POWER (สีแดง) , TX (สีแดง) และ RX (สีเขียว) รวมทั้ง LED (สีเขียว) แสดงสถานะการทำงาน
9. ใช้ไฟเลี้ยง 5 VDC โดยต่อเข้ากับขั้วต่อสีขาว 2 Pin

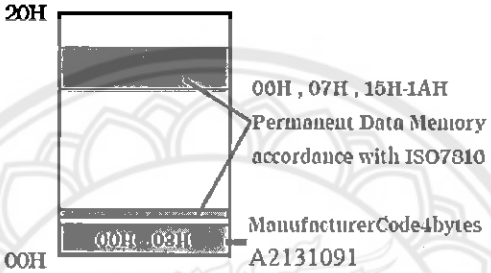
Smart Card SLE4442

Smart Card เบอร์ SLE4442 มีหน่วยความจำแบบ EEprom ขนาด 256 Bytes โดยแบ่งเป็น Protectable Data Memory 32 Bytes และ Unprotected Data Memory 224 Bytes สามารถอ่านและเขียนได้ 100,000 ครั้ง เก็บข้อมูลได้นานถึง 10 ปี ส่วนที่เป็น Protectable Data Memory นั้นสามารถเขียนข้อมูลถาวรไว้โดยจะลบหรือแก้ไขเปลี่ยนแปลงไม่ได้อีกเลยและในส่วนนี้ได้ถูกเขียนข้อมูลถาวรไว้แล้ว 12 Bytes ตามมาตรฐาน ISO7816 นอกจากนี้ SLE4442 ยังมี PSC (Programmable Security Code) 3 Bytes เพื่อใช้ในการตรวจสอบค่าให้ตรงกับค่า PSC ที่มีในบัตรก่อนจึงจะเขียนข้อมูลลงในบัตรได้ และ EC(Error Counter) เพื่อใช้ในการนับจำนวนครั้งที่ทำการตรวจสอบ Verify ค่า PSC โดยถ้าทำการตรวจสอบ Verify ค่า PSC ไม่ถูกต้องถึง 3 ครั้ง บัตรนี้จะเขียนข้อมูลไม่ได้อีกเลยทันที การนับ Error Counter นี้จะถูก Reset เมื่อได้ทำการ Verify ค่า PSC ได้ถูกต้อง ค่า PSC มาตรฐานของบัตรใหม่ที่ผลิตจากโรงงานคือ **FFFFFF**

EEPROM



Protectable Data Memory



การตั้ง DIP-SWITCH

*** SW.3 - SW.4 มีผลต่อการทำงานเมื่อทำการ POWER UP
ส่วน SW.5 - SW.8 จะมีผล ต่อการทำงานทันที***

Default ON (1) OFF (0)

SW.1 SW.2	Reserve ไว้ไม่ได้ใช้งาน
SW.3	SW.4	อัตราเร็วการ Baudrate (Default ON, ON)
OFF	OFF	BR 2400
OFF	ON	BR 4800
ON	OFF	BR 9600
ON	ON	BR 19200

SW.5	การกำหนดค่าสถานี (Default : OFF)
OFF	ไม่มี Address
ON	มี Address (กรณีที่ใช้แบบ Network)

SW.6	SW.7	SW.8	ADDRESS (Default : OFF, OFF, OFF)
OFF	OFF	OFF	0
OFF	OFF	ON	1
OFF	ON	OFF	2
OFF	ON	ON	3
ON	OFF	OFF	4
ON	OFF	ON	5
ON	ON	OFF	6
ON	ON	ON	7

ชุดคำสั่งควบคุม TSM-256

TSM-256 มีคำสั่งในการติดต่อสั่งงานควบคุมทั้งหมด 8 คำสั่ง รูปแบบเป็นรหัส ASCII ทั้งหมด โดยมีลักษณะดังนี้

:ACXX….X<C R>	กรณีตั้ง DIP-SW.5 เป็น ON (NETWORK)
:CXX….X<C R>	กรณีตั้ง DIP-SW.5 เป็น OFF

:	คือ รหัสนำของคำสั่ง (3AH)
A	คือ Address ของบอร์ดตั้งแต่ 0 - 7
C	คือ รหัสคำสั่งตั้งแต่ 0 - 7
XX….X	คือ ข้อมูลติดตามของแต่ละคำสั่งซึ่งอาจจะมีหรือไม่ก็ได้รวมมีความยาวตามกำหนดแต่ละคำสั่ง
<C R>	คือ รหัสลงท้ายของแต่ละคำสั่ง (0DH)

โดยเมื่อ TSM-256 รับคำสั่งแล้วก็จะทำงานตามคำสั่งทันที รายละเอียดของแต่ละรหัสคำสั่งมีดังต่อไปนี้

คำสั่ง CHECK :0<CR> / :A0<CR> (กรณีที่ต่อแบบ NETWORK)

เมื่อ TSM-256 ได้รับคำสั่งนี้จะส่งข้อความแสดงชื่อสินค้าและ Version ของสินค้ากลับมาดังนี้

TSM-256 V1.0<CR>

คำสั่ง STATUS :1<CR> / :A1<CR> (กรณีที่ต่อแบบ NETWORK)

คำสั่งนี้ TSM-256 จะทำการตรวจสอบว่ามีบัตรเสียบอยู่บนบอร์ดหรือไม่ หากไม่มีบัตรอยู่บนบอร์ดจะส่งข้อมูลกลับมา ดังนี้คือ **ER<CR>** หากมีบัตรเสียบอยู่บนบอร์ดจะส่งข้อมูลแสดง Manufacturer Code (4 Bytes) กลับมาดังนี้

XXXXXXXX<CR>

เช่น เมื่อเสียบบัตร Smart Card เบอร์ SLE4442 อยู่ TSM-256 จะส่ง Manufacturer Code กลับมาคือ **A2131091<CR>**

<u>คำสั่ง Read Data</u> :2BB<CR> / :A2BB<CR> (กรณีที่ต่อแบบ NETWORK)

เป็นคำสั่งที่ใช้ในการอ่านข้อมูลของบัตรออกมาแสดงเป็นจำนวน 16 Bytes โดยผู้ใช้ต้องกำหนดตำแหน่ง Address เริ่มต้น (เป็นเลขฐานสิบหก) **BBH** ด้วยทุกครั้ง ข้อมูลที่ส่งกลับมามีลักษณะดังนี้

XXXXXXXX_ _ _ _XX<CR>

เช่น เมื่อส่งคำสั่ง **:221<CR>** TSM-256 เมื่อได้รับคำสั่งจะเริ่มอ่านข้อมูลในตำแหน่ง Address 21H ไปจนถึงข้อมูลในตำแหน่ง Address 30H หากไม่ได้เสียบบัตรบนบอร์ดหรือไม่ใช่บัตร Smart Card เบอร์ SLE4442 จะไม่สามารถอ่านข้อมูลออกมาได้และบอร์ด TSM-256 จะส่งคำว่า **ER<CR>** กลับมา

<u>คำสั่ง Write Data</u> :3BBXXXX_ _XX<CR>/:A3BBXXXX_ _XX<CR>(กรณีที่ต่อแบบ NETWORK)

คำสั่งนี้ใช้ในการเขียนข้อมูลลงในบัตร Smart Card สามารถเขียนข้อมูลความยาวสูงสุดได้ครั้งละ 16 Bytes เริ่มเขียนข้อมูลที่ตำแหน่ง Address เริ่มต้น **BBH** ที่ได้กำหนดไว้จนถึงตำแหน่ง Address ของข้อมูล Byte สุดท้าย เมื่อเขียนข้อมูลลงในบัตรเรียบร้อยแล้ว TSM-256 จะส่งคำว่า **OK<CR>** กลับมา หากไม่สามารถเขียนข้อมูลลงในบัตรได้หรือบนบอร์ดไม่มีบัตรหรือไม่ใช่บัตร Smart Card เบอร์ SLE4442 จะส่งคำว่า ER<CR> กลับมา เช่น เมื่อส่งคำสั่ง **:32115151515<CR>** ข้อมูลในตำแหน่ง Address 21H ไปจนถึงข้อมูลในตำแหน่ง Address 24H จะมีค่าเป็น 15H ทั้งหมด หรือเมื่อส่งคำสั่ง

:30831313131<CR> ข้อมูลในตำแหน่ง Address 08H ไปจนถึงข้อมูลในตำแหน่ง Address 0BH จะมีค่าเป็น 31H ทั้งหมดแต่เป็นการเขียนแบบไม่ถาวรสามารถลบหรือแก้ไขเปลี่ยนแปลงได้

<u>คำสั่ง Read Protection Memory</u> :4<CR> / :A4<CR> (กรณีที่ต่อแบบ NETWORK)

คำสั่งนี้เป็นการอ่านค่า Bit Organization ของส่วน Protection Memory ทั้ง32Bits เรียงจาก 0-31 (ซ้ายไปขวา) โดยแต่ละ Bit จะแสดงค่าว่าตำแหน่ง Address ของส่วน Protection Memory (00H-20H) ตำแหน่งใดที่ยังสามารถเขียนข้อมูล ลงไปได้ และตำแหน่งใดได้มีการเขียนข้อมูลถาวรไว้แล้ว โดยตำแหน่งที่มีการเขียนข้อมูลถาวรไว้นั้นจะมีค่าเป็น 0 ส่วนตำแหน่งที่ยังสามารถเขียนข้อมูลลงไปได้จะมีค่าเป็น 1 เช่น **00001100111111111111100000011111<CR>** จะเห็นว่า Bit ในตำแหน่งที่ 0-3 เป็น 0 แสดงว่าตำแหน่ง Address ที่ 00H-03H มีข้อมูลถาวรอยู่ไม่สามารถเขียนข้อมูลทับลงไปได้ แต่ถ้าไม่มีบัตรนี้อยู่หรือไม่ใช่บัตรเบอร์ SLE4442 จะได้รับคำว่า **ER<CR>** กลับมา

<u>คำสั่ง Write Protection Memory</u> :5BBXXXX_XX<CR>/:A5BBXXXX_XX<CR>(กรณีที่ต่อแบบ NETWORK)

คำสั่งนี้จะคล้ายกับคำสั่ง Write Data ต่างกันตรงที่เมื่อใช้คำสั่งนี้ตามหลังคำสั่งที่ 3 จะเป็นการเขียนข้อมูลถาวรลงในส่วน Protection Memory โดย BBH จะเป็นตำแหน่ง Address เริ่มต้นที่จะเขียนข้อมูลลงไป เมื่อเขียนข้อมูลเสร็จเรียบร้อยจะส่งคำว่า **OK<CR>** กลับมาแต่ถ้าไม่สามารถเขียนข้อมูลลงในบัตรนี้ได้หรือบัตรไม่ใช่เบอร์ SLE4442 หรือไม่มีบัตรอยู่ก็จะส่งคำว่า **ER<CR>** กลับมา เช่น ถ้าต้องการเขียนข้อมูลถาวรลงในตำแหน่ง Address 08H ถึงข้อมูลในตำแหน่ง Address 0BH ให้มีค่าเป็น 31H ให้ส่งคำสั่ง **:30831313131<CR>** และตามหลังด้วยคำสั่ง **:50831313131<CR>** ให้กับ TSM-256

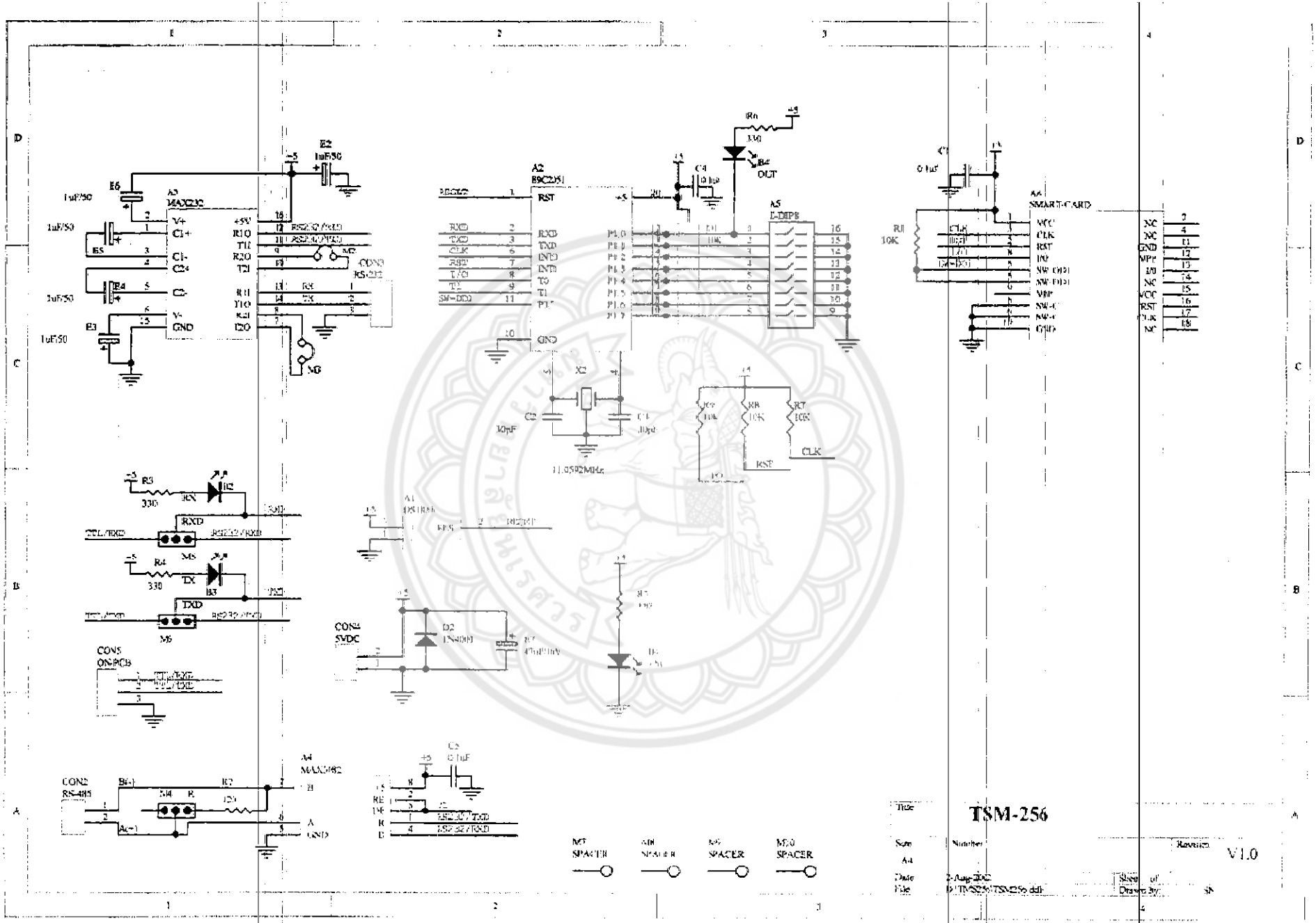
จะพบว่าในตำแหน่ง Address 08H ถึงข้อมูลในตำแหน่ง Address 0BH จะมีค่าเป็น 31H และไม่สามารถเปลี่ยนแปลงแก้ไขใดๆ ได้อีกเลย เมื่อส่งคำสั่ง **:4<CR>** ให้กับ TSM-256 เพื่ออ่านค่า Bit Organization ของส่วนProtection Memory Bit ในตำแหน่งที่ 8-11 จะมีค่าเป็น 0 ดังนี้ **000011000**0001111111110000011111<CR> แต่ถ้าใส่เพียงคำสั่งที่ 5 อย่างเดียวโดยไม่ได้ส่งคำสั่งที่ 3 ไปก่อนก็จะไม่มีผลกับการเปลี่ยนแปลงของข้อมูลใดๆ เลย

<u>คำสั่ง Verify PSC</u> :6PPPPPP<CR> / :A6PPPPPP<CR> (กรณีที่ต่อแบบ NETWORK)

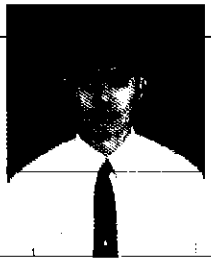
คำสั่งนี้ใช้ในการตรวจสอบ Verify ค่า PSC (Programmable Security Code) ซึ่งเป็นรหัสขนาด 3 Bytes (PPPPPPH) โดยจะต้องทำการตรวจสอบ Verify ค่า PSC ก่อนเสมอหลังจากที่จ่ายไฟเข้าเพื่อที่จะสามารถเขียนข้อมูลลงไปในบัตรได้ด้วยการใส่ค่า PSC (PPPPPPH) ตามหลังรหัสคำสั่ง หาก Verify ค่า PSC ได้ตรงกันกับค่าในบัตรก็จะส่งคำว่า **OK<CR>** กลับมาแต่ถ้าไม่ถูกต้องก็จะส่งคำว่า **EN**<CR> กลับมา (N คือจำนวนครั้งที่ทำการ Verify ค่า PSC เช่น ถ้า Verify ค่า PSC ไม่ถูกต้องครั้งที่ 1 ก็ส่งคำว่า **E1<CR>** กลับมา) และจะนับเก็บค่า Error Counter (EC) ไว้จนกว่าจะถูก Reset เมื่อได้ทำ การ Verify ค่า PSC ได้ถูกต้อง การตรวจสอบ Verify ค่า PSC สามารถทำได้ 3 ครั้ง ถ้าตรวจสอบ Verify ค่า PSC ไม่ถูกต้องจนถึงครั้งที่ 3 (TSM-256 จะส่งคำว่า **E3<CR>** กลับมา) บัตรนี้จะไม่ สามารถเขียนข้อมูลใดๆ ลงไปได้อีก บัตรใหม่ที่ผลิตจากโรงงานนั้นค่า PSC Code คือ FFFFFF การ Verify หลังจากจ่ายไฟเข้าบัตรนี้ถ้าถูกต้องจะมีผลตลอดไป จนกว่าจะดึงบัตรออก ถึงแม้ว่าจะส่งคำสั่งไป Verify ค่า PSC ที่ไม่ถูกต้องอีกครั้งก็จะไม่มีผลใดๆทั้งสิ้นเพราะถือว่าได้ทำการ Verify ค่า PSC ถูกต้องตรงกันกับค่าในบัตรไปแล้ว

<u>คำสั่ง CHANGE PSC</u> :7PPPPPP<CR> / :A7PPPPPP<CR> (กรณีที่ต่อแบบ NETWORK)

คำสั่งนี้ใช้ในการเปลี่ยนค่า PSC โดยข้อมูล PPPPPPH ที่ตามหลังรหัสคำสั่งจะเป็นค่า PSC Code ที่ต้องการกำหนด ขึ้นมาใหม่ เมื่อ TSM-256 ได้ทำการเปลี่ยนค่าเสร็จเรียบร้อยจะส่งคำว่า **OK<CR>** กลับมาแต่ถ้าไม่สามารถเปลี่ยน ค่าได้หรือไม่ใช่บัตรเบอร์ SLE4442 หรือไม่มีบัตรก็จะส่งคำว่า **ER<CR>** มาแทน


TSM-256
V1.0
MAX232
STC89C51
SMART-CARD
L-DIP8
MAX485
CON2
RS-485
CON3
RS-232
CON4
5VDC
CON5
ON-PCB
D2
1N4001
X2
11.0592MHz
RST
CLK
RXD
TXD
MOSI
MISO
SCK
VCC
GND
NOT SPACER
AIR SPACER
SMT SPACER
M3.0 SPACER

ประวัติผู้เขียนโครงงาน



ชื่อ นายพีรพล เนือยทอง

ภูมิลำเนา 461 หมู่ 9 ต.โอโล อ. ภูเขียว จ.ชัยภูมิ 36110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนภูเขียว
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : peerapon_nuaytong@hotmail.com

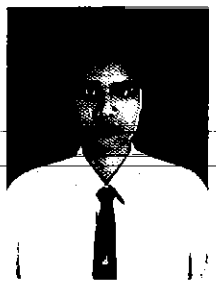
ชื่อ นายพีรศักดิ์ ศักดิ์สกุลเดช

ภูมิลำเนา 279/7-8 หมู่ 1 ถ.สุขาภิบาล1 อ.ลานสัก จ.อุทัยธานี 61160

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนลาซาลโชติรวีนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : super_jeaw@hotmail.com



ชื่อ นายเทอดพงษ์ หลีเกษม

ภูมิลำเนา 227 ถ.แดงทองดี อ.ตะพานหิน จ.พิจิตร 66110

ประวัติการศึกษา

- จบระดับมัธยมศึกษาจากโรงเรียนนครสวรรค์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail : tleekasem@hotmail.com