การใช้ประยุกต์ใช้บัตรสมาร์ตการ์ด

SMART CARD APPLICATION

โดย

นายนันทวัฒน์ จิระวัฒนพงศ์

นายพิรุณ บุญคง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมระบบควบคุม สาขาวิศวกรรมแมคคาทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมระบบควบคุม สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

คณะวิศวกรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ใช้บัตรสมาร์ตการ์ด
SMART CARD APLICATION

ผู้จัดทำ นายนันทวัฒน์ จิระวัฒนพงศ์ 46010354
นายพิรุณ บุญคง 46010529

.....ธวัชชัย คำศรี..... อาจารย์ที่ปรึกษา
(อาจารย์ ธวัชชัย คำศรี)

การประยุกต์ใช้บัตรสมาร์ตการ์ด

โดย
นายนันทวัฒน์ จิระวัฒนพงศ์ 46010354
นายพิรุณ บุญคง 46010529

อาจารย์ที่ปรึกษา
อาจารย์ ธวัชชัย คำศรี

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ จะนำเสนอการออกแบบระบบการจอดรถยนต์ โดยการประยุกต์ใช้เทคโนโลยีสมาร์ตการ์ด โดยใช้โปรแกรมที่เขียนขึ้นจากโปรแกรมวิชวลเบสิกในการควบคุมระบบบัตรสมาร์ตการ์ดที่ใช้นั้นเป็นแบบใช้หน้าสัมผัสรุ่น SLE 4442 โดยใช้ควบคู่กับเครื่องอ่าน – เขียนบัตรสมาร์ตการ์ดรุ่น TSM 256 ในส่วนของการสร้างฐานข้อมูลนั้นได้มีการนำโปรแกรมไมโครซอฟท์แอกเซสมาใช้ในการสร้างฐานข้อมูลเพื่อเก็บข้อมูลของผู้ใช้งาน

สำหรับภาพโดยรวมของระบบนั้น ในบัตรสมาร์ตการ์ดจะมีรหัสของผู้ใช้เก็บไว้ เมื่อเครื่องอ่าน – เขียนบัตรสมาร์ตการ์ดรุ่น TSM 256 อ่านรหัสจากบัตรแล้วจะส่งไปยังคอมพิวเตอร์ผ่านสายสัญญาณ RS – 232 โปรแกรมบนคอมพิวเตอร์จะประมวลผลรหัสและทำการสืบค้นฐานข้อมูลเพื่อค้นหาข้อมูลของผู้ใช้งาน จากนั้นการประมวลผลจะแบ่งออกเป็น 2 กรณี คือ 1. ในกรณีการลงทะเบียนเข้าใช้ โปรแกรมจะบันทึกเวลาเข้าใช้งานในฐานข้อมูล 2. ในกรณีที่มีการลงทะเบียนเลิกใช้งานโปรแกรมจะทำการคำนวณค่าจอดรถโดยคำนวณเวลาเข้าใช้ และเลิกใช้นำมาลบกันคูณด้วยอัตราค่าจอด จากนั้นจึงแสดงผลให้ผู้ใช้งานทราบ

SMART CARD APPLICATION

By

Mr. Nantawat Jirawattanapong

Mr. Phiroon Boonkong

Advisor

Mr. Thawatchai Kamsri

Academic Year 2006

ABSTRACT

In this thesis designing car parking system by using smart card technology are proposed. A program that was written by Visual basic will be used for control a system. A smart card we use is SLE 4442 contact smart card. This smart card will be access by using TSM 256 smart card reader - writer. As for a database, we use Microsoft access to create a database to store user data.

For the system overview every smart card will store user code. This code will be read by TSM 256 and send to PC via RS – 232 protocal. Once a program in a PC detect a user code. It will search the database for user data. After this a process will separate in to 2 cases, 1st case is log in case, in this thesis program will store login time into database. For 2nd case, log out case. In this case program will calculate parking cast by comparing log in and log out time and display in to user.

กิตติกรรมประกาศ

ปริญญานิพนธ์ ฉบับนี้สำเร็จได้ด้วย ความกรุณาของท่านอาจารย์ ธวัชชัย คำศรี อาจารย์ที่ปรึกษาปริญญานิพนธ์ซึ่งได้ให้คำปรึกษาข้อชี้แนะ และความช่วยเหลือในหลายสิ่งหลายอย่างจนกระทั่งลุล่วงไปได้ด้วยดี ผู้ทำโครงงานขอขอบพระคุณเป็นอย่างสูงมา ณ ที่นี้ ข้อบกพร่องต่าง ๆ ของโครงงานรวมทั้งผู้ทรงคุณวุฒิที่ ตรวจสอบและให้คำแนะนำ ขอกราบขอบพระคุณคณาจารย์ ภาควิชาแมคคาทรอนิกส์ ที่ได้ให้ความรู้ ให้คำแนะนำ ให้กำลังใจตลอดการศึกษาที่ผ่านมา ขอบคุณขอบใจพี่เพื่อนและน้องภาควิชาแมคคาทรอนิกส์ทุกคนที่คอยถามไถ่ด้วยความห่วงใยว่าเมื่อไหร่จะสำเร็จการศึกษา และขอขอบคุณเพื่อนร่วมรุ่น ที่ลุ้นเอาใจช่วยรวมถึงผู้มีพระคุณทุกท่าน ณ ที่นี้

ท้ายที่สุดนี้ คุณความดีและกุศลที่พึงบังเกิดมีจากปริญญานิพนธ์เล่มนี้เป็นผลมาจากความเมตตา กรุณา ของ บิดา มารดาผู้คอยให้กำลังใจ จึงขอยกความดีเหล่านั้นเป็นเครื่องบูชาพระคุณด้วยความเคารพและสักการะยิ่ง

ผู้จัดทำ

นาย นันทวัฒน์ จิระวัฒนพงศ์

นาย พิรุณ บุญคง

สารบัญ

สารบัญ (ต่อ)

สารบัญ (ต่อ)

สารบัญภาพ

สารบัญภาพ (ต่อ)

สารบัญตาราง

บทที่ 1
บทนำ

1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์

การพัฒนาเทคโนโลยีด้านต่างๆ ในทุกวันนี้ล้วนมีจุดมุ่งหมายเพื่อให้มนุษย์มีชีวิตความเป็นอยู่ที่ดีขึ้น มีความสะดวกสบายและปลอดภัยมากขึ้น สมาร์ตการ์ด (Smart Card) ก็เป็นหนึ่งในการพัฒนาเพื่อเหตุผลนี้เช่นเดียวกัน โดยสมาร์ตการ์ดเป็นการพัฒนาความสามารถของบัตร (Card) ที่ใช้ในการเก็บข้อมูลให้มีขนาดของหน่วยความจำที่ใหญ่ขึ้น และมีความปลอดภัยของข้อมูลภายในบัตรสูงขึ้นกว่าบัตรแถบแม่เหล็ก (Magnetic Card) ที่ใช้กันอยู่เดิม และด้วยคุณสมบัติเช่นนี้ทำให้มีแนวในการที่จะทำสมาร์ตการ์ดไปใช้ในการเก็บข้อมูลสำคัญต่าง ๆ มากขึ้นเรื่อย ๆ

สำหรับในประเทศไทยนั้น ได้เริ่มมีการนำสมาร์ตการ์ดไปใช้ในงานด้านต่างๆ บ้างแล้ว เช่น บัตรเอทีเอ็ม (ATM) บัตรประจำตัวพนักงาน บัตรโทรศัพท์ เป็นต้น และในตอนนี้รัฐบาลก็เริ่มที่จะนำสมาร์ตการ์ดมาใช้ทำบัตรประจำตัวประชาชนแล้ว โดยให้สมาร์ตการ์ดทำหน้าที่เป็นตัวแทนในการเก็บข้อมูลพื้นฐานต่างๆ เช่น ชื่อ-นามสกุล ที่อยู่ตามทะเบียนราษฎร์ วันเดือนปีเกิด กรุ๊ปเลือด และหมายเลขประจำตัว เป็นต้น และแนวความคิดในการที่นำสมาร์ตการ์ดไปทำเป็นบัตรประจำตัวประชาชนจึงทำให้ทางผู้จัดเกิดความคิดในการที่จะนำสมาร์ตการ์ดมาทำเป็นบัตรประจำตัวผู้จอดรถ เนื่องจากมีการใช้รถในประเทศมากขึ้นทุกวันตามจำนวนประชากรที่เพิ่มขึ้น และเพื่อป้องกันอาชญากรรมจากการโดนขโมยรถ โดยให้สมาร์ตการ์ดทำหน้าที่เป็นตัวแทนในการเก็บข้อมูลต่างๆ เกี่ยวกับเจ้าของรถ เช่น ชื่อ-นามสกุล ยี่ห้อรถ ทะเบียนรถ เป็นต้น ทั้งนี้ก็เพื่อให้การดำเนินการเกี่ยวกับระบบการจอดรถมีความสะดวกรวดเร็วและปลอดภัยมากขึ้น

1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. เข้าใจถึงการทำงานและโครงสร้างของสมาร์ตการ์ด
2. สามารถนำเครื่องอ่าน–เขียนสมาร์ตการ์ดไปประยุกต์ใช้ในงานอื่นๆ ได้
3. เขียนโปรแกรมคอมพิวเตอร์เพื่อเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกได้
4. เขียนโปรแกรมรองรับการใช้งานสมาร์ตการ์ดในคอมพิวเตอร์ได้

1.3 ขอบเขตของปริญญานิพนธ์

เขียนโปรแกรมคอมพิวเตอร์ที่รองรับการใช้งานสมาร์ตการ์ด เพื่อติดต่อใช้กับระบบการจอดรถ ซึ่งประกอบด้วย ระบบการเข้าหรือออกลานจอดรถ และระบบเติมเงินของบัตร โดยโปรแกรมนั้นเขียนขึ้นด้วยภาษา Visual Basic ที่มีการติดต่อกับระบบฐานข้อมูลที่จัดทำโดยโปรแกรม Microsoft Access

ตารางที่ 1.1 ขอบเขตของปริญญานิพนธ์

เทอมที่หนึ่ง	ศึกษาข้อมูลการใช้บัตร โปรแกรม Visual Basic และทดลองการเขียน การอ่านข้อมูลจากบัตรเบื้องต้น
เทอมที่สอง	ศึกษาข้อมูลเพิ่มเติม เพื่อนำมาใช้ในระบบการจอดรถ รูปแบบการเป็นสมาชิก การคิดค่าจอดในระยะเวลาและอัตราต่าง ๆ

1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึงวัตถุประสงค์ หลักการใหม่ ขั้นตอนการศึกษา และการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์แต่ละบท

บทที่ 2 ประวัติความเป็นมา ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องในส่วนของสมาร์ตการ์ดชนิดต่างๆ ส่วนประกอบมาตรฐานต่างๆ และชุดคำสั่งของโปรแกรมที่ใช้ในการควบคุม

บทที่ 3 หลักการออกแบบและพัฒนาระบบจอดรถโดยใช้บัตรสมาร์ตการ์ด นำเสนอโครงสร้างระบบ แนวคิดในการออกแบบและการทำงานของระบบ

บทที่ 4 การทดลอง เป็นส่วนทดสอบใช้งานโปรแกรมในระบบและฐานข้อมูล ของระบบจอดรถ

บทที่ 5 บทสรุปและวิจารณ์ จะสรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้นและแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

บทที่ 2
ทฤษฎีพื้นฐานที่ใช้ในโครงงาน

ทฤษฎีพื้นฐานต่างๆ ที่เกี่ยวข้องในการวิจัย และพื้นฐานของสมาร์ตการ์ด ซึ่งเนื้อหาในบทนี้จะกล่าวถึงการเปรียบเทียบตัวสมาร์ตการ์ดแบบต่างๆ คุณลักษณะของบัตรต่างๆ และพื้นฐานของโครงสร้างของเครื่องอ่าน-เขียนบัตรสมาร์ตการ์ด ซึ่งเนื้อหาทั้งหมดนี้จำเป็นสำหรับการศึกษาและเปรียบเทียบหาข้อดี–ข้อเสีย เพื่อประเมินประสิทธิภาพของสมาร์ตการ์ด ดังต่อไปนี้

2.1 สมาร์ตการ์ดคืออะไร

สมาร์ตการ์ดคือบัตรพลาสติกที่มีชิปไอซี (Integrated Circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเองโดยวิธีการเข้ารหัสตามมาตรฐาน DES Algorithm (Data Encryption Standard) เพื่อให้ระบบมีระดับความปลอดภัยสูงขึ้น ด้วยคุณสมบัติสำคัญประการหนึ่งที่ทำให้สมาร์ตการ์ดมีความแตกต่างจากบัตรพลาสติกทั่วไปก็คือ ขณะทำรายการ (Transaction) สมาร์ตการ์ดสามารถทำงานได้ด้วยตัวเองโดยไม่ต้องอาศัยติดต่อสื่อสารกับระบบหลัก (Font End) นั่นก็คือสมาร์ตการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก (Off-Line) ทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก

2.2 ประวัติความเป็นมาของสมาร์ตการ์ด

สมาร์ตการ์ดปรากฏขึ้นครั้งแรกในประเทศเยอรมัน ในปี 1986 โดยชาวเยอรมัน (Jurgen Dethloff และ Helmut Grotupp) เป็นผู้คิดค้น แต่ผู้ที่ได้มาซึ่งสิทธิบัตรกลับเป็นชาวญี่ปุ่น (Kunitaka Arimura) ในปี 1970 และมีการจดสิทธิบัตรในชื่อของสมาร์ตการ์ดโดยชาวฝรั่งเศส (Roland Moreno) ในปี 1974 ในระยะแรกนั้นสมาร์ตการ์ดยังทำงานได้ไม่สมบูรณ์นัก เพราะสมาร์ตการ์ดรุ่นแรกๆ ยังมีปัญหาทางเทคนิคเล็กๆ น้อยๆ แม้ว่าสมาร์ตการ์ดจะถือกำเนิดในยุโรป แต่ในระยะแรกสมาร์ตการ์ดกลับไม่ค่อยได้รับความสนใจเท่าที่ควร จนกระทั่งปี 1984 บริษัท French PTT (Postal and Telecommunications Service) ได้นำสมาร์ตการ์ดมาใช้งานเป็นบัตรโทรศัพท์ เพื่อป้องกันการโกงค่าโทรศัพท์ ในครั้งนั้นโครงการเป็นโครงการนำร่องโดยมีการนำบัตรแถบแม่เหล็ก บัตรแถบแสง (Optical Storage) และสมาร์ตการ์ดมาทำการทดลองใช้งานเปรียบเทียบกัน ซึ่งแน่นอนว่าสมาร์ตการ์ดได้พิสูจน์ให้เห็นคุณลักษณะที่เหนือกว่าบัตรชนิดอื่น ทั้งในเรื่องความทนทาน ความปลอดภัย ความสวยงาม เป็นผลให้สมาร์ตการ์ดในรูปของบัตรโทรศัพท์มีการนำไปใช้ถึง 60 ล้านใบ (เฉพาะประเทศฝรั่งเศส) และตอกย้ำความสำเร็จอีกกว่า 100 ล้านใบจาก 50 ประเทศทั่วโลกในปี

1997 กระนั้นสมาร์ตการ์ดก็ยังเป็นเพียงบัตรโทรศัพท์ การนำสมาร์ตการ์ดมาใช้ทางด้านการเงินการธนาคารกลับเป็นไปอย่างเชื่องช้า เนื่องจากบัตรที่เกี่ยวข้องกับระบบการเงินการธนาคารมีความยุ่งยากมากกว่าบัตรโทรศัพท์

และในปี 1960 เทคโนโลยีการประมวลผลเพื่อเข้ารหัสข้อมูลของฮาร์ดแวร์และซอฟแวร์มีความพร้อมมากขึ้น จึงมีการนำมาใช้ในการเข้ารหัสข้อมูลในสมาร์ตการ์ด ซึ่งแต่เดิมนั้นการเข้ารหัสจะมีการใช้งานเฉพาะในหน่วยงานทหารหรือหน่วยงานราชการลับเท่านั้น ด้วยเหตุนี้ทำให้สมาร์ตการ์ดสามารถทำการเข้า-ถอดรหัสข้อมูลได้ด้วยตัวเอง ทำให้การใช้สมาร์ตการ์ดมีความปลอดภัยสูงขึ้นจนสามาถนำมาใช้เป็นบัตรเครดิตหรือบัตรเงินสดได้อย่างสมบูรณ์แบบ

ในปี 1984 ธนาคารในฝรั่งเศสได้นำสมาร์ตการ์ดมาใช้เป็นบัตรเครดิตเป็นครั้งแรก ในระยะแรกนั้นต้องประสบกับปัญหามากมายเกี่ยวกับการเข้ากันได้ของบัตรต่างธนาคาร ซึ่งต้องใช้เวลาถึง 10 ปีที่จะทำให้เข้ากันได้ทั้งหมด เป็นเหตุให้มีการรวมกันของ Europay, VISA และMASTER เพื่อกำหนดมาตรฐานแก่เครดิตการ์ด ในรูปของสมาร์ตการ์ดให้มีมาตรฐานเดียวกันทุกธนาคารในชื่อของมาตรฐาน EMV (Europay , MASTER , VISA) โดยอ้างอิงกับมาตรฐาน ISO 7816 เป็นหลัก ทำให้มีผู้ที่ต้องการพัฒนาแอพพลิเคชั่นเครดิตหรือเดบิตบนสมาร์ตการ์ด ต้องทำตามข้อกำหนดของมาตรฐาน EMV เท่านั้น

2.3 ส่วนประกอบและโครงสร้างของสมาร์ตการ์ด

ในปัจจุบันมีบัตรสมาร์ตการ์ดมากมายหลายแบบให้เลือกใช้งาน แต่ก็มีส่วนประกอบที่ไม่ต่างกันมากนัก ซึ่งบัตรสมาร์ตการ์ดจะประกอบไปด้วยบัตรพลาสติก กาวหรือวัสดุที่ใช้เชื่อมต่อและหน้าสัมผัสที่บรรจุชิปสมาร์ตการ์ดเรียบร้อยแล้ว ซึ่งรายละเอียดของส่วนประกอบต่างๆ มีดังนี้

2.3.1 ตัวบัตรพลาสติก

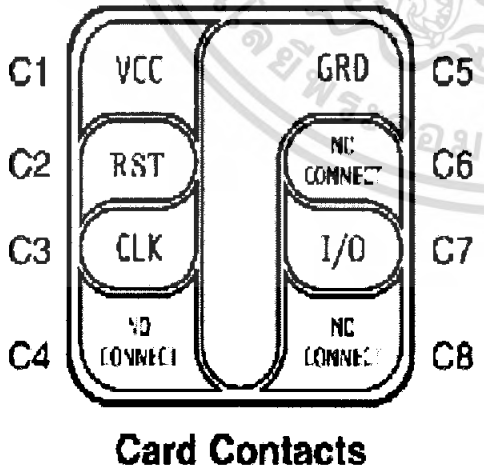
สมาร์ตการ์ดเป็นชิปไอซีขนาดเล็กที่ถูกสร้างขึ้นเช่นเดียวกับชิ้นส่วนอิเลกทรอนิกส์อื่นๆ ที่สร้างจากสารกึ่งตัวนำ โดยนำมาติดลงบนหน้าสัมผัส และทำการฝังลงในเนื้อบัตรพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นตัวบัตรจะใช้พลาสติก 4 ชนิด ได้แก่ PVC (Polyvinyl Chloride), ABS (Acrylonitrile Butadiene Styrene), PC (Polycarbonate) และ PET (Polyethylene Terephthalate) ในประเทศไทยจะใช้บัตรพลาสติก PV มากเป็นอันดับหนึ่ง ส่วนอันดับสองเป็นบัตรพลาสติกชนิด ABS ซึ่งบัตรพลาสติกชนิด PVC มักนำมาใช้เป็นบัตรเอทีเอ็ม บัตรเครดิต-เดบิต บัตรประจำตัวประชาชน ฯลฯ ส่วนบัตรพลาสติกชนิด ABS ไม่ค่อยพบว่าใช้งานกันมากนักเนื่องจากราคาสูงกว่าและลายที่พิมพ์ลงบนธนบัตรไม่สวยงามคงทนเท่าบัตรพลาสติกชนิด PVC จะพบก็เพียงบัตรพลาสติกเนื้อผสมโดยใช้พลาสติกชนิด ABS เป็นแกนและฉาบผิวด้วยพลาสติกชนิด PVC แต่ความทนทานของตัวบัตรจะสู้บัตรพลาสติกเนื้อ PVC ล้วนไม่ได้

สำหรับบัตรพลาสติกอีก 2 ชนิดที่เหลือ ยังไม่พบว่ามีการใช้งานในประเทศไทย อาจเนื่องมาจากราคาที่สูงเกินไปของวัสดุที่นำมาใช้ทำเป็นตัวบัตร และคุณสมบัติของวัสดุที่ด้อยกว่าพลาสติกชนิด PVC แต่ข้อเสียที่สำคัญของพลาสติกชนิด PVC ก็ไม่ด้อยไปกว่าข้อดีของมันนั้นก็คือ มันไม่สามารถย่อยสลายได้ตามธรรมชาติ ซึ่งเท่ากับเป็นขยะสำหรับสิ่งแวดล้อมเลยทีเดียว

2.3.2 หน้าสัมผัสและชิปสมาร์ตการ์ด

สมาร์ตการ์ดโมดูล (Smart Card Module) หรือหน้าสัมผัสและชิปสมาร์ตการ์ด คือ ส่วนที่แสดงความเป็นตัวตนของสมาร์ตการ์ดที่ชัดเจนที่สุด สมาร์ตการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่าคือ สมาร์ตการ์ดที่มีการฝังชิปไว้ในเนื้อบัตร ดังนั้นการที่จะระบุว่าบัตรใบใดเป็นบัตรสมาร์ตการ์ดนั้น ต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก ซึ่งต้องใช้ประสบการณ์ที่เกี่ยวกับสมาร์ตการ์ดพอสมควร แต่ในที่นี้จะขอแนะนำให้เห็นภาพลักษณ์ที่ชัดเจนของสมาร์ตการ์ดเป็นหลัก ซึ่งก็คือส่วนของสมาร์ตการ์ดโมดูลนั่นเอง

ในการผลิตสมาร์ตการ์ดโมดูล ส่วนที่เป็นหน้าสัมผัสของสมาร์ตการ์ดประกอบด้วยโลหะหลายชิ้นประกอบกัน แต่ละส่วนจะถูกยึดด้วยแถบฟิล์มบางๆ ทางด้านหลังของหน้าสัมผัสเพื่อให้คงรูปอยู่ได้แถบฟิล์มตัวนี้จะมีการเจาะช่องเล็กๆ สำหรับการเชื่อมต่อสายนำสัญญาณกับชิปสมาร์ตการ์ดกับหน้าสัมผัส หลังจากที่วางชิปสมาร์ตการ์ดเข้ากับหน้าสัมผัสเรียบร้อยแล้ว ขั้นตอนสุดท้ายจะเป็นการผนึกชิปสมาร์ตการ์ดเพื่อป้องกันตัวชิป และสายนำสัญญาณต่างๆ จากสิ่งแวดล้อมภายนอก (เป็นการทดสอบขั้นต้น) ดังรูปที่ 2.1 ส่วนขั้นตอนที่เหลือจะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติก และทดสอบการทำงานของชิปขั้นสุดท้าย



หน้าสัมผัส	ชื่อขา	การใช้งาน
C1	Vcc	แหล่งจ่ายไฟ
C2	RST	รีเซต
C3	CLK	สัญญาณนาฬิกา
C5	GND	กราวนด์
C7	I/O	รับ-ส่งข้อมูล
C4,6,8	N.C.	ไม่ใช้งาน

รูปที่ 2.1 หน้าสัมผัสและการทำงานของขาต่างๆ ของบัตรสมาร์ตการ์ด

2.4 การประกอบสมาร์ตการ์ดโมดูลลงในบัตรพลาสติก

การประกอบโมดูลลงในบัตรนั้นมีอยู่หลายวิธีด้วยกัน ตามแต่ชนิดของสมาร์ตการ์ดโมดูลและการเตรียมบัตรพลาสติก ซึ่งการเตรียมบัตรพลาสติกจะนำมาใส่สมาร์ตการ์ดโมดูลมีด้วยกัน 2 แบบ คือ บัตรพลาสติกที่ถูกขุดหลุมบนบัตร และบัตรพลาสติกที่เกิดจากการทับซ้อนของชั้นพลาสติกที่เจาะช่องมาแล้ว โดยสมาร์ตการ์ดโมดูลจะใช้บัตรพลาสติกที่มีการเตรียมการมาแล้วดังนี้

2.4.1 การสร้างสมาร์ตการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก

การสร้างสมาร์ตการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก (TAB: Tape Automated Bonding) สมาร์ตการ์ดชนิดนี้เกิดจากการทับซ้อนของพลาสติกตั้งแต่ 3 ชั้นขึ้นไป โดยแต่ละชั้นจะมีการเจาะช่องตามขนาดหน้าสัมผัสและชิปสมาร์ตการ์ดไว้อยู่แล้ว ส่วนที่เป็นหน้าสัมผัสและชิปจะถูกแทรกอยู่ในชั้นในพลาสติก เมื่อวางซ้อนทับกันเรียบร้อยแล้วก็จะอัดแต่ละชั้นด้วยความร้อนเมื่อความร้อนถึงจุดที่ทำให้พลาสติกแต่ละชั้นประสานตัวเอง ก็จะนำมาตัดแต่งขอบบัตรและทำการทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย

2.4.2 การสร้างสมาร์ตการ์ดด้วยวิธีการวางสมาร์ตการ์ดโมดูลลงในเนื้อบัตร

การสร้างสมาร์ตการ์ดด้วยวิธีการวางสมาร์ตการ์ดโมดูลลงในเนื้อบัตร (Chip-on-Flex) สมาร์ตการ์ดโมดูลที่จะนำมาใส่ลงในบัตรพลาสติก ผู้ผลิตจะทำการตัดตามขนาดของหลุมบนบัตรพลาสติกที่ขุดรอไว้แล้วด้วยเครื่องจักร ทำการเชื่อมด้วยกาว และอบด้วยความร้อนเพื่อให้สมาร์ตการ์ดโมดูลติดสนิทกับเนื้อพลาสติก จากนั้นจึงทำการทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย การใส่หน้าสัมผัสและชิปสมาร์ตการ์ดด้วยกาวนี้ เป็นวิธีที่นิยมทำกันมากที่สุด เพราะผู้ผลิตสามารถประหยัดต้นทุนในการผลิตได้มาก เนื่องจากวิธีการนี้เสียค่าใช้จ่ายในการผลิตน้อยที่สุด ไม่ว่าจะเป็นเรื่องของแรงงานความรวดเร็วในการผลิต และเปอร์เซ็นต์สินค้าชำรุดต่ำในคุณภาพของสมาร์ตการ์ดที่ยังพอยอมรับได้

2.4.3 การสร้างสมาร์ตการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวของบัตร

การสร้างสมาร์ตการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวของบัตร (Chip-On-Surface) สมาร์ตการ์ดโมดูลชนิดติดบนผิวของบัตร ผลิตโดยการใช้แสงเลเซอร์ทำการขุดหลุมบนบัตรพลาสติกขนาดเท่ากับตัวชิปสมาร์ตการ์ด วางชิปสมาร์ตการ์ดลงในตำแหน่งที่กำหนด สร้างหน้าสัมผัสและเชื่อมสายสัญญาณกับชิปสมาร์ตการ์ดด้วยหมึกนำไฟฟ้า สุดท้ายพิมพ์ทับส่วนที่เป็นชิปและหมึกนำไฟฟ้าส่วนที่เป็นสายสัญญาณด้วยหมึกที่มีคุณสมบัติเป็นฉนวนไฟฟ้า เพื่อป้องกันวงจรภายในโดยปล่อยส่วนที่เป็นหน้าสัมผัสที่สร้างจากหมึกนำไฟฟ้าเท่านั้น สมาร์ตการ์ดชนิดนี้ไม่ค่อยมีให้เห็นมากนักเนื่องจากต้องใช้เทคโนโลยีในการผลิตที่สูงกว่าหน้าสัมผัสแบบอื่น ทำให้ต้นทุนการผลิตสูงตามไปด้วยอีกทั้งความทนทานก็ยังน้อยกว่าสมาร์ตการ์ดโมดูลชนิดอื่น ๆ

2.5 ข้อกำหนดอีเอ็มวี

ข้อกำหนดอีเอ็มวี (EMV: Europay, MASTER, VISA) เป็นข้อ กำหนดที่ใช้ระบุข้อกำหนดขั้นต่ำของสมาร์ตการ์ด และเครื่องรับบัตรสมาร์ตการ์ด ซึ่งเป็นการร่วมกันระหว่าง Europay, MASTER และ VISA ในการกำหนดคุณสมบัติของสมาร์ตการ์ดที่จะนำมาประยุกต์ใช้ในการทำธุรกรรมทางด้านการเงินและการธนาคาร ซึ่งกำลังได้รับการผลักดันให้กลายเป็นมาตรฐานในการธุรกรรมบนสมาร์ตการ์ด ประกอบด้วยข้อกำหนด 3 ตัว ได้แก่

2.5.1 ข้อกำหนดสำหรับบัตรสมาร์ตการ์ดที่ใช้ในระบบการชำระเงิน

เป็นข้อกำหนดส่วนใหญ่ในข้อกำหนดอีเอ็มวี สำหรับบัตรสมาร์ตการ์ดนี้อ้างอิงกับมาตรฐาน ISO7816 โดยแบ่งออกเป็น 4 ส่วนย่อย ดังนี้

1. **คุณสมบัติทางกายภาพของบัตร** คุณสมบัติทางกายภาพของบัตรและข้อกำหนดในการแลกเปลี่ยนข้อมูลประกอบด้วย
   - คุณสมบัติเชิงกล เช่น ขนาด และตำแหน่งของชิปสมาร์ตการ์ด
   - คุณสมบัติทางไฟฟ้า เช่น แรงดัน กระแสไฟฟ้า ความต้านทาน ความถี่ที่ใช้งาน
   - โปรโตคอล (Protocol) ที่ใช้ในการสื่อสาร

2. **รายละเอียดข้อมูลและชุดคำสั่งที่ใช้ในการสื่อสาร** ประกอบด้วย
   - โครงงานข้อมูลที่ใช้สำหรับการสื่อสารเพื่อธุรกรรมทางการเงิน
   - โครงสร้างไฟล์ข้อมูลในสมาร์ตการ์ด
   - ชุดคำสั่งสำหรับการสื่อสาร

3. **ขั้นตอนการประมวลผล** เป็นการกำหนดขั้นตอนสำหรับเครื่องรับบัตรสมาร์ตการ์ดว่าในการทำธุรกรรมต้องประมวลผลอะไรบ้าง รวมถึงการบังคับให้บัตรหลายๆ บัตรสามารถใช้งานร่วมกันได้ และกำหนดให้มีฟังก์ชั่นการทำงานภายในบัตรมากพอเพื่อใช้ในการทำธุรกรรม ซึ่งข้อกำหนดนี้ประกอบด้วย
   - โครงสร้างไดเร็กทอรี่ของข้อมูลในสมาร์ตการ์ด
   - กระบวนการประมวลผล

4. **มาตรฐานการรักษาความปลอดภัย** ประกอบด้วย
   - ป้องกันการแก้ไขข้อมูลที่หวงห้าม
   - กุญแจรหัสสำหรับการแจกจ่าย
   - การตรวจสอบความถูกต้องของกุญแจรหัส (Key Verification)
   - รูปแบบการรักษาความปลอดภัยของสายข้อมูล (Secure Messaging)

2.5.2 ข้อกำหนดสำหรับรับบัตรสมาร์ตการ์ดที่ใช้ในระบบการชำระเงิน

เป็นข้อกำหนดสำหรับเครื่องรับบัตรสมาร์ตการ์ดที่ระบุถึงส่วนบังคับ ส่วนที่แนะนำ และส่วนที่เป็นทางเลือกให้แก่ผู้ผลิตเครื่องรับบัตรสมาร์ตการ์ด โดยรวมถึงเครื่องเอทีเอ็ม เครื่อง POS อุปกรณ์ประกอบเครื่องคอมพิวเตอร์ส่วนบุคคล เครื่องเก็บเงินอิเล็กทรอนิกส์ เครื่องอ่านบัตรสำหรับผู้ถือบัตร และเครื่องเติมเงิน ซึ่งประกอบด้วย

- ชนิดของเครื่องรับบัตรสมาร์ตการ์ด และความสามารถในการทำงาน
- คุณสมบัติทางกายภาพทั่วไป
- สถาปัตยกรรมทางด้านซอฟต์แวร์
- มาตรการรักษาความปลอดภัยของเครื่องรับบัตรสมาร์ตการ์ด
- ส่วนแสดงผลสำหรับผู้ถือบัตร
- ส่วนแสดงผลสำหรับรับบัตร หรือร้านค้า

2.5.3 ข้อกำหนดมาตรฐานสำหรับการใช้งานบัตรสมาร์ตการ์ดในระบบการชำระเงิน

เป็นข้อกำหนดที่เจาะจงสำหรับกระบวนการในการใช้บัตรเพื่อชำระเงินประกอบด้วย

- โครงสร้างไฟล์ข้อมูลสำหรับจัด การเรื่องของธุรกรรม
- ขั้นตอนการทำธุรกรรมด้วยบัตรสมาร์ตการ์ด
- การประมวลผลสำหรับธุรกรรมที่ผิดปกติ

2.6 มาตรฐานที่เกี่ยวข้อง

มาตรฐานที่เกี่ยวข้องกับสมาร์ตการ์ดมีด้วยกันหลายมาตรฐาน มาตรฐานหลายๆ ตัวมีเนื้อหาที่ซ้ำซ้อนกันเป็นผลให้สมาร์ตการ์ดเป็นเรื่องที่ค่อนข้างยุ่งยาก ถึงกระนั้นการพัฒนาระบบเพื่อใช้งานร่วมกับสมาร์ตการ์ด ก็ยังต้องยึดถือตามมาตรฐานเหล่านั้น มาตรฐานที่นำมาใช้กับสมาร์ตการ์ดนี้จะมีทั้งมาตรฐานที่มีอยู่แต่เดิมก่อนที่จะมีสมาร์ตการ์ด และมาตรฐานที่กำหนดขึ้นสำหรับสมาร์ตการ์ดโดยเฉพาะซึ่งอ้างอิงจากมาตรฐานบัตรพลาสติกและบัตรแถบแม่เหล็กที่มีอยู่แล้ว มาตรฐาน ISO7810 – ISO7811 และ ISO7813

ก่อนที่จะเข้าถึงเรื่องของสมาร์ตการ์ด คงต้องขอกล่าวถึงมาตรฐานอีกตัวหนึ่ง ซึ่งมีความสำคัญพอๆ กับมาตรฐานของสมาร์ตการ์ด นั่นก็คือมาตรฐานของบัตรพลาสติก และบัตรแถบแม่เหล็ก นั่นก็คือมาตรฐาน ISO7810 และมาตรฐาน ISO7811

ISO7810 : มาตรฐานที่กำหนดด้วยเรื่องคุณสมบัติทางกายภาพเบื้องต้นของบัตรพลาสติก

ISO7811-1 : มาตรฐานที่กำหนดด้วยเรื่องของตัวอักษรนูนบนบัตร

ISO7811-2 : มาตรฐานที่กำหนดด้วยเรื่องของแถบแม่เหล็กบนบัตร

ISO7811-3 : มาตรฐานที่กำหนดตำแหน่งของการพิมพ์ตัวอักษรนูนบนบัตร

ISO7811-4 : มาตรฐานที่กำหนดตำแหน่งที่อยู่ของข้อมูลแทรก 1 และ 2

ISO7811-5 : มาตรฐานที่กำหนดตำแหน่งของข้อมูลแทรก 3

ISO7811-6 : มาตรฐานที่กำหนดด้วยเรื่องของแถบแม่เหล็กแบบความหนาแน่นสูงบนบัตร

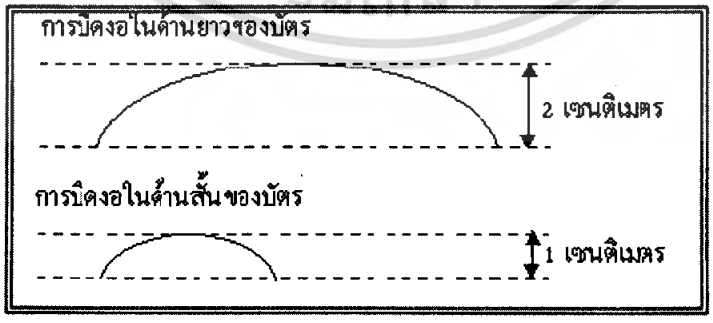
ISO7813 : มาตรฐานที่กำหนดด้านการเงินการธนาคาร

2.6.1 มาตรฐาน ISO7816

เพื่อให้เกิดความเข้ากันได้ของสมาร์ตการ์ด จึงมีการกำหนดมาตรฐานของสมาร์ตการ์ดในชื่อของ ISO7816 เป็นการกำหนดในเรื่องของคุณลักษณะของบัตรพลาสติกที่จะนำมาทำเป็นบัตรสมาร์ตการ์ด โดยมาตรฐาน ISO7816 มีหัวข้อย่อยโดยแบ่งเป็น ISO7816-1, ISO7816-3 และปัจจุบันมีถึง ISO7816-6 ซึ่งมีข้อกำหนดเกี่ยวกับเรื่องของสมาร์ตการ์ดมากมายเกินกว่าจะจดจำได้ทั้งหมด ดังนั้นขอแสดงให้เห็นเพียวคร่าวๆ เบื้องต้นดังนี้

ISO7816-1: มาตรฐานที่กำหนดด้วยเรื่องคุณสมบัติทางกายภาพเบื้องต้นของสมาร์ตการ์ดมีดังนี้ ความทนทานต่อแสง และรังสีชนิดต่าง ๆ

- ขนาดความหนาของชิปสมาร์ตการ์ด
- ความทนทานต่อแรงกดดันของหน้าสัมผัส (ทนทานต่อแรงกดดัน 1.5 นิวตันได้โดยไม่เสียหาย)
- ค่าความต้านทานของหน้าสัมผัส (ไม่เกิน 0.5 โอห์ม ที่กระแส 0.5 ไมโครแอมป์ – 300 มิลลิแอมป์)
- ความทนทานต่อสนามแม่เหล็ก
- ความทนทานต่อไฟฟ้าสถิต (1500 โวลต์ ประจุ 100 พิโกฟารัด ที่ 1500 โอห์ม)
- ความทนทานต่อการบิดงอ เป็นจำนวน 30 ครั้งต่อนาทีโดยบัตรและชิปต้องไม่เกิดความเสียหาย ดังรูปที่ 2.2



รูปที่ 2.2 ความทนทานต่อการบิดงอของบัตรสมาร์ตการ์ด

2.6.2 ISO7816-2: มาตรฐานที่กำหนดขนาดของหน้าสัมผัส และตำแหน่งของหน้าสัมผัสชิปสมาร์ตการ์ดบนบัตร ประกอบด้วย

- ขนาดของหน้าสัมผัสของชิปสมาร์ตการ์ด
- ตำแหน่งของหน้าสัมผัสของบัตร

2.7 รายละเอียดพื้นฐานของสมาร์ตการ์ด

สมาร์ตการ์ดเป็นบัตรพลาสติกขนาดเท่าบัตรเครดิต หรือบัตรเอทีเอ็ม (ATM: Automatic Teller Machine) ที่มีหน่วยเก็บข้อมูล และหน่วยประมวลผลที่เรียกว่า "ไมโครชิป" ติดอยู่บนบัตร ซึ่งข้อมูลนี้อาจจะอยู่ในรูปของตัวเลขหรือตัวอักษรก็ได้ โดยมีกลไกในการเขียนและการอ่านข้อมูลที่ซับซ้อน ทำให้ยากต่อการปลอมแปลง จึงสามารถนำมาใช้ประโยชน์ในด้านต่างๆ มากมาย เช่น ด้านการเงินและการธนาคาร ด้านโทรคมนาคม ด้านงานทะเบียน ด้านการศึกษา ด้านการรักษาความปลอดภัย เป็นต้น

รูปที่ 2.3 รูปบัตรสมาร์ตการ์ด

สมาร์ตการ์ดมีพื้นฐานมาจากระบบไมโครโปรเซสเซอร์ ซึ่งมีแนวคิดเริ่มแรกจากการนำชิปหน่วยความจำ(EEPROM) มาฝังลงในบัตรพลาสติก โดยมีหน้าสัมผัสเป็นขาเชื่อมต่อกับระบบภายนอก ในการเชื่อมต่อต้องมีการ ป้อนกระแสไฟฟ้าให้ชิปหน่วยความจำสามารถทำงานได้ การสั่งงานเพื่ออ่านหรือเขียนข้อมูลจากชิปหน่วยความจำสมาร์ตการ์ด ก็ทำได้โดยการเชื่อมต่อสัญญาณผ่านหน้าสัมผัสที่กำหนดไว้แล้ว ในการเชื่อมต่อขาสัญญาณของชิปหน่วยความจำแบบธรรมดา อาจไม่เหมาะสมนักสำหรับบัตรพลาสติกขนาดเล็ก เนื่องจากจำนวนขาสัญญาณของหน่วยความจำ (Bus) มีจำนวนไม่น้อยทีเดียว ยิ่งหน่วยความจำที่มีความจุสูง ๆ ยิ่งต้องใช้สัญญาณอ้างอิงตำแหน่ง

ของข้อมูล (Address Bus) มากขึ้น จึงมีการนำเอาระบบสื่อสารแบบซิงเกิลบัสมาใช้ในการรับส่งข้อมูล โดยในการนำเอาระบบสื่อสารแบบอนุกรมมาใช้ จำเป็นต้องมีการป้อนสัญญาณนาฬิกาเพื่อกำกับจังหวะการรับ-ส่งข้อมูลแต่ละบิต ทำให้ต้องมีหน้าสัมผัสสำหรับสัญญาณนาฬิกาบนชิปสมาร์ตการ์ดเพิ่มขึ้นมา แต่ก็นับว่าทำให้ขาเชื่อมต่อลดลงไปไม่น้อยทีเดียว ด้วยเหตุนี้สมาร์ตการ์ดชนิดหน่วยความจำจึงเป็นสมาร์ตการ์ดชนิดแรกที่ถูกสร้างขึ้น

การนำเอาชิปหน่วยความจำมาใส่ในบัตรพลาสติก ทำให้เกิดข้อดีเหนือบัตรแถบแม่เหล็กด้วยความจุข้อมูลที่มากกว่า ไม่มีผลต่อสนามแม่เหล็กไฟฟ้า และรอยขีดข่วน ทำให้สมาร์ตการ์ดโดดเด่นกว่าบัตรแถบแม่เหล็กอย่างเทียบกันไม่ได้ แต่ข้อเสียประการหนึ่งของการใช้หน่วยความจำเพียงอย่างเดียวคือสามารถทำการอ่านและเขียนข้อมูลได้อย่างอิสระเช่นเดียวกับบัตรแถบแม่เหล็ก จึงถือได้ว่าความปลอดภัยของข้อมูลเกือบเป็นศูนย์ นั่นก็คือ ข้อมูลภายในสมาร์ตการ์ดชนิดนี้ไม่เป็นความลับ ด้วยเหตุนี้จึงมีการเพิ่มวงจรสำหรับป้องกันลงไปอีก เพื่อให้ผู้ออกบัตร (Card Issue) สามารถกำหนดสิทธิในการเข้าถึงข้อมูลแต่ละไบต์ด้วยวงจรฟิวส์แมทริกธรรมดา ๆ ที่เมื่อกำหนดเงื่อนไขไปแล้วจะไม่สามารถแก้ไขได้อีก ต่อมาเมื่อเทคโนโลยีทางด้านเซมิคอนดักเตอร์สูงขึ้น จึงมีการออกแบบวงจรที่สามารถกำหนดเป็นกุญแจรหัส (PIN) สำหรับเข้าถึงข้อมูลในบัตร ซึ่งต้องทำการแสดงกุญแจรหัสทุกครั้งที่บัตรเริ่มทำงาน เพื่อป้องกันการเจาะระบบอีกชั้นหนึ่ง อีกทั้งกุญแจรหัสก็ยังสามารถเปลี่ยนแปลงได้อีกด้วย

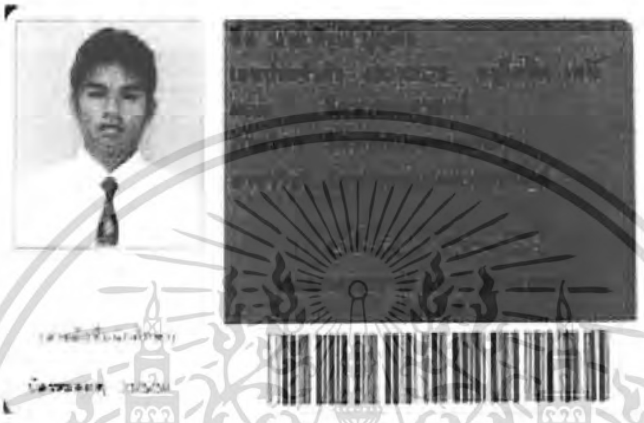
ต่อมามีการนำเอาไมโครโปรเซสเซอร์ (ที่จริงแล้วเป็นไมโครคอนโทรลเลอร์ แต่ในที่นี้จะขอเรียกว่าไมโครโปรเซสเซอร์เป็นหลัก) มาใส่ลงในสมาร์ตการ์ด ทำให้เกิดเป็นสมาร์ตการ์ดชนิดใหม่ที่มีความซับซ้อนยิ่งขึ้น การเข้าถึงข้อมูลไม่สามารถทำได้โดยตรงเหมือนอย่างสมาร์ตการ์ดชนิดหน่วยความจำ การใช้งาน สมาร์ตการ์ดชนิดนี้ ต้องเขียน ขึ้นเป็นชุดคำสั่ง และส่งให้กับชิปไมโครโปรเซสเซอร์ทำงานแทนการที่ใส่ชิปไมโครโปรเซสเซอร์ลงไปในสมาร์ตการ์ด ทำให้ต้องมีการเพิ่มส่วนของหน่วยความจำโปรแกรม (OS-Operating System) สำหรับไมโครโปรเซสเซอร์เพื่อให้ไมโครโปรเซสเซอร์สามารถทำการประมวลผลคำสั่งต่าง ๆ และสามารถโปรแกรมการเข้าถึงข้อมูล ทำให้ช่องโหว่ที่สำคัญของสมาร์ตการ์ดได้รับการแก้ไขจนเกือบสมบูรณ์แบบ

นอกจากสมาร์ตการ์ดทั้งสองชนิดที่ได้กล่าวมายังมีสมาร์ตการ์ด อีกชนิดหนึ่งที่ไม่ใช้หน้าสัมผัส (Contactless) ในการรับส่งสัญญาณโดยอาศัยเทคโนโลยีคลื่นวิทยุในการติดต่อสื่อสาร สมาร์ตการ์ดชนิดนี้อาศัยการแปลงคลื่นวิยุส่วนหนึ่งมาใช้เป็นกระแสไฟฟ้าสำหรับป้อนให้ชิป อีกส่วนหนึ่งมาดีเทกเอาข้อมูลคำสั่งให้ชิป สมาร์ตการ์ด ชนิดนี้ได้รับความนิยมค่อนข้างมาก เพราะความน่าตื่นตาตื่นใจ และล้ำสมัยของมัน แต่กระนั้นราคาของมันก็ย่อมสูงตามไปด้วย

2.8 การศึกษาและเปรียบเทียบข้อดี – ข้อเสียของบัตรชนิดอื่น ๆ

บัตรแต่ละชนิดย่อมมีข้อดี–ข้อเสียต่างกันไป ก่อนจะศึกษาถึงการใช้บัตรสมาร์ตการ์ดต้องศึกษาข้อดี –ข้อเสียของบัตรชนิดอื่น ๆ ก่อน ซึ่งมีรายละเอียด ดังนี้

2.8.1 บัตรชนิดรหัสแท่ง



รูปที่ 2.4 รูปด้านหน้าของบัตรชนิดรหัสแท่ง (Barcode)

รูปที่ 2.5 รูปด้านหลังของบัตรชนิดรหัสแท่ง (Barcode)

- **ข้อดี** คือ เป็นเทคโนโลยีที่ง่าย และเป็นมาตรฐานที่ได้รับการยอมรับ และใช้งานอย่างกว้างขวางมีความสะดวกสบาย และง่ายต่อการนำมาประยุกต์ใช้งาน และยังมีราคาถูกที่สุดในบรรดาบัตรทั้ง 3 แบบนี้

- **ข้อเสีย** คือ สามารถปลอมแปลงได้ง่ายมาก เพียงแค่นำบัตรต้นฉบับไปถ่ายเอกสารก็สามารถใช้งาน ข้อมูลแทนที่ได้เหมือน ต้นฉบับ ทุกประการ ดังนั้นจึง ไม่เหมาะ ที่จะนำมาใช้งานที่ต้องการความปลอดภัยของข้อมูล โดยทั่วไปจะใช้เป็นบัตรประจำตัวหรือบัตรสมาชิกต่าง ๆ โดยใช้งานควบคู่ไปกับข้อมูลแสดงความเป็นตัวตนบุคคลตามข้อมูลที่ปรากฏบนบัตรซึ่งจะใช้รหัสแท่งเพียงเพื่อเป็นตัวนำข้อมูลเข้าสู่ระบบการประมวลผล

2.8.2 บัตรชนิดแถบแม่เหล็ก (Magnetic Stripe Card)

รูปที่ 2.6 รูปด้านหน้าของบัตรชนิดแถบแม่เหล็ก (Magnetic Stripe Card)

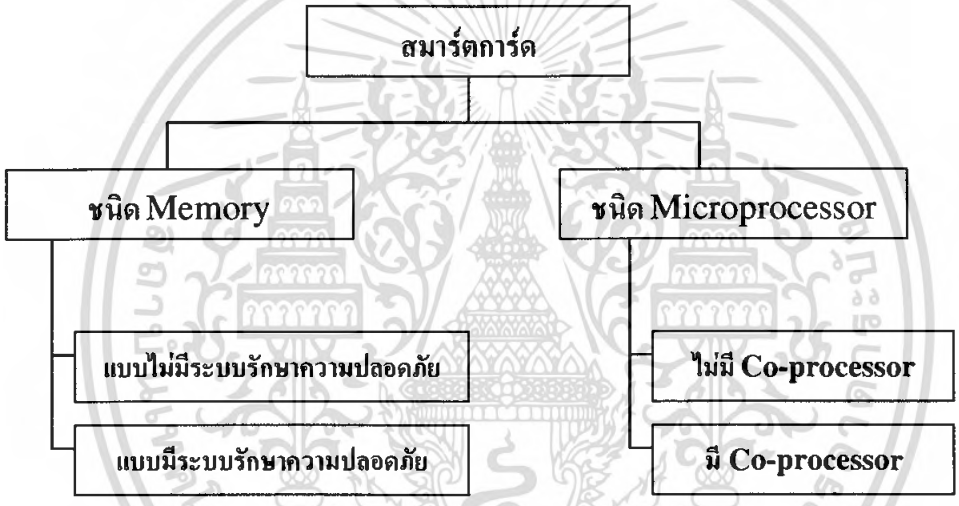
รูปที่ 2.7 รูปด้านหลังของบัตรชนิดแถบแม่เหล็ก (Magnetic Stripe Card)

- **ข้อดี** คือ ผู้ใช้สามารถอ่านข้อมูลเบื้องต้นได้ แล้วใช้อุปกรณ์ อ่าน / บันทึก ข้อมูลลงแถบแม่เหล็กเพื่อนำไปประมวลผลทางระบบคอมพิวเตอร์ใด ๆ โดยมีราคาที่ค่อนข้างถูกทั้งยังสามารถอ่านและบันทึกข้อมูลลงในบัตรได้

- **ข้อเสีย** คือ ไม่มีความปลอดภัยของข้อมูล เนื่องจากถ้าหากเราทราบถึงโครงสร้างรูปแบบของการจัดเก็บข้อมูลภายในแถบแม่เหล็กก็จะสามารถทำการอ่านข้อมูลนั้นขึ้นมา และทำการอ่านข้อมูลนั้นขึ้นมาและทำการปลอมแปลงได้ และข้อมูลที่เก็บไว้ภายในแถบแม่เหล็ก ก็มีความเสี่ยงที่จะสูญหาย / เสียหายได้ เมื่อถูกความร้อนหรือสนามแม่เหล็กไฟฟ้า

2.9 ชนิดของสมาร์ตการ์ด

การแบ่งชนิดของสมาร์ตการ์ดในปัจจุบันอาจทำได้ยากสักหน่อย เนื่องจากมีการใส่เทคโนโลยีใหม่ ๆ ลงสมาร์ตการ์ดตลอดเวลา ถ้าจะแบ่งตามชนิดของหน่วยความจำภายในอาจไม่ชัดเจนนัก ยิ่งแบ่งตามลักษณะการเชื่อมต่อก็คงไม่ครอบคลุมสมาร์ตการ์ดให้เข้าใจได้ง่าย ดังรูปที่ 2.8



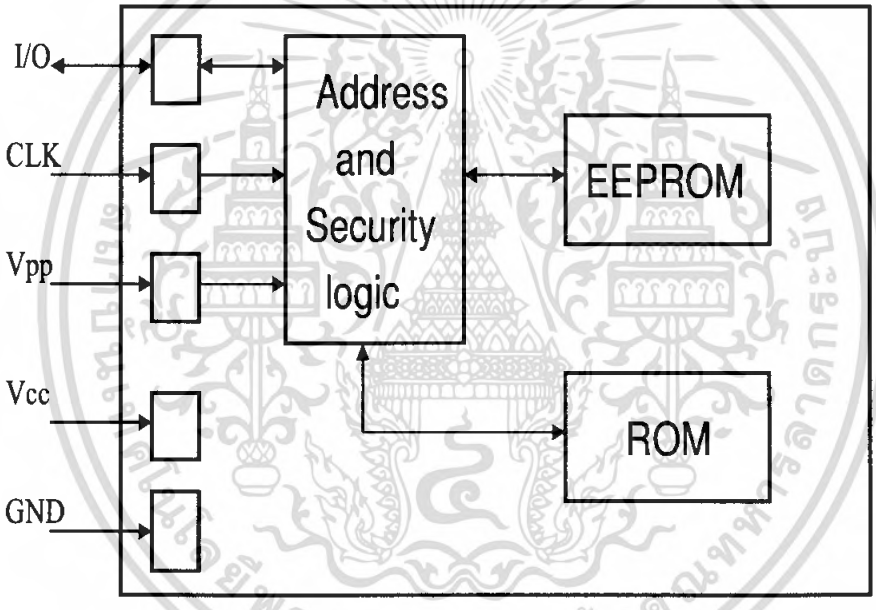
รูปที่ 2.8 การแบ่งสมาร์ตการ์ดตามชนิดของหน่วยความจำ

จะเห็นได้ว่าเราสามารถแบ่งสมาร์ตการ์ดจากโครงสร้างภายในได้ 2 ชนิดก็คือสมาร์ตการ์ดชนิดหน่วยความจำ (Memory Card) และ สมาร์ตการ์ดชนิดไมโครโปรเซสเซอร์ (Processor Card) ซึ่งชิปทั้งสองแบบจะมีหน้าสัมผัสเหมือนกัน แต่สัญญาณที่ต้องป้อนให้แก่หน้าสัมผัสบางหน้าสัมผัสจะไม่มีการใช้งานในสมาร์ตการ์ดต่างชนิดกัน เช่น แรงดันไฟฟ้าสำหรับการเขียนข้อมูลลงในชิป (Vpp) จะมีใช้ในสมาร์ตการ์ดชนิดหน่วยความจำเท่านั้น สัญญาณนาฬิกาสำหรับป้อนให้ชิปทำงาน (CLK) ต้องป้อนให้กับชิปเหมือนกัน สำหรับสัญญาณนาฬิกา (CLK) ที่ป้อนให้ชิปสมาร์ตการ์ดเป็นสัญญาณนาฬิกาภายนอกที่ป้อนให้ชิปทำงานได้ เพราะภายในชิปสมาร์ตการ์ดไม่มีวงจรสำหรับสร้างสัญญาณนาฬิกา แต่หน้าสัมผัส I/O จะมีการรับ-ส่งข้อมูลที่แตกต่างกันในเรื่องของความถี่ และวิธีการควบคุมจังหวะการรับ-ส่งของข้อมูลแต่ละบิต

ในการแบ่งสมาร์ตการ์ดออกเป็น 2 ชนิด ตามชนิดของวงจรภายในดังที่กล่าวมา อาจแบ่งได้อีกลักษณะคือ แบ่งตามความถี่ในการรับ-ส่งข้อมูลผ่านหน้าสัมผัส I/O ของสมาร์ตการ์ด ดังที่กล่าวไปแล้ว ซึ่งสามารถแบ่งได้ดังนี้

2.9.1 การ์ดชนิดหน่วยความจำ

สมาร์ตการ์ดชนิดหน่วยความจำ (Memory) หรืออีกชื่อหนึ่งคือ Synchronous card เนื่องจากสมาร์ตการ์ดชนิดนี้มีการรับ-ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิป (ข้อมูลแต่ละบิตที่ส่งให้แก่ชิปต้องสัมพันธ์กับสัญญาณนาฬิกา) สมาร์ตการ์ดชนิดนี้มี โครงสร้างที่ประกอบไปด้วย ส่วนวงจรสำหรับ การติดต่อสื่อสารกับภายนอก หน่วยความจำข้อมูล และหน่วยความจำสำหรับเก็บชุดคำสั่งของสมาร์ตการ์ด ดังรูปที่ 2.9



รูปที่ 2.9 บล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ตการ์ดชนิดหน่วยความจำ

สมาร์ตการ์ดที่เป็นพื้นฐานของสมาร์ตการ์ดในปัจจุบัน ก็คือสมาร์ตการ์ดชนิด Free Access Memory สมาร์ตการ์ดชนิดนี้เปิดโอกาสให้อ่านหรือเขียนข้อมูลในแอดเดรสใดๆ ก็ได้ตาม ชื่อของสมาร์ตการ์ดชนิดนี้ ไม่มีการป้องกันข้อมูลใดๆ ภายในสมาร์ตการ์ดชนิดนี้ ซึ่งแน่นอนว่าเป็นสมาร์ตการ์ดที่มีความปลอดภัยต่ำสุด ถึงกระนั้นการอ่านข้อมูล โดยมีวงจรควบคุมการสลับตำแหน่งของบิตเป็นส่วนป้องกันข้อมูลอีกต่อหนึ่ง ดังนั้นการอ่านข้อมูลออกแบบธรรมดาจะไม่ได้ข้อมูลที่ถูกต้องหากไม่ติดต่อกับวงจรควบคุมการสลับตำแหน่งของบิตโดยตรง

นอกจากนี้สมาร์ตการ์ดชนิดหน่วยความจำแบบธรรมดา ยังมีการใส่วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลลงไปด้วย ทำให้สามารถกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลได้ทุกไบต์ โดยสมาร์ตการ์ดที่มีวงจรป้องกันการอ่าน-เขียนชนิดนี้ถูกเรียกว่า PIN Protect Memory เนื่องจากการเข้าถึงข้อมูลจะต้องแสดงรหัสผ่าน ให้บัตรทราบก่อนถึงจะสามารถเข้าถึงข้อมูลได้ วงจรกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลจะมีบิตพิเศษที่มีชื่อว่า Bit Protect ซึ่งเป็นบิตข้อมูลที่ฝากไว้กับ ข้อมูลให้เป็นบิตที่ 9 แต่ไม่สามารถแก้ไขได้ด้วยคำสั่งเขียนข้อมูลธรรมดา เพราะ Bit Protect ไม่ได้เป็นส่วนหนึ่งของข้อมูลจริงๆ ในการแก้ไข Bit Protect นี้จะสามารถทำการเปลี่ยนแปลงได้เพียงครั้งเดียวด้วยคำสั่งเฉพาะเท่านั้น เช่น หากต้องการบังคับไม่ให้ข้อมูลไบต์ใดไม่สามารถแก้ไขได้ก็ให้ทำการเคลียร์บิตที่ 9 ของข้อมูลนั้นๆ แต่สำหรับรหัสผ่านในการเข้าถึงข้อมูลสามารถเปลี่ยนแปลงได้ แต่ต้องแสดงรหัสผ่านชุดเก่าให้บัตรได้ทราบเสียก่อนจึงจะสามารถเปลี่ยนแปลงรหัสผ่านได้

สมาร์ตการ์ดอีกชนิดหนึ่งที่มีใช้เป็นบัตรโทรศัพท์ในประเทศไทยนั้นคือ การ์ดหน่วยความจำชนิด Token ภายในสมาร์ตการ์ดชนิดนี้ จะมีการเก็บข้อมูลในลักษณะจำนวนนับ (Counter) ซึ่งจำนวนนับนี้จะเป็นตัวเลขแทนมูลค่าของเงินที่ระบุบนบัตร การนับเลขเป็นการนับถอยหลังเพื่อเป็นการนับมูลค่าที่คงเหลือในบัตร หมายความว่าหากใช้บัตรในการโทรศัพท์ไปเรื่อยๆ มูลค่าในบัตรก็จะถูกลดลงตามไปด้วยเช่นกัน ในการเข้าถึงข้อมูลของ สมาร์ตการ์ดชนิดนี้ต้องมีการแสดงหัสผ่านให้บัตรทราบเหมือนกับการ์ดหน่วยความจำ ชนิด PIN Protect แต่ไม่มี Bit Protect เท่านั้นเอง

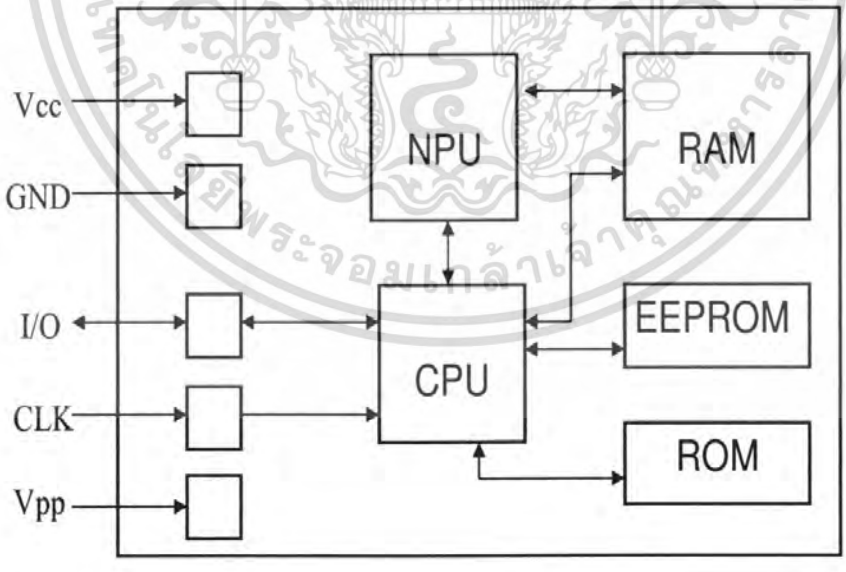
สมาร์ตการ์ดชนิดหน่วยความจำ เป็นสมาร์ตการ์ดพื้นฐานของสมาร์ตการ์ดรุ่นใหม่ๆ ในปัจจุบันด้วยโครงสร้าง และการทำงาน ที่ง่ายต่อการทำความเข้าใจ ราคาถูก สามารถเก็บข้อมูลได้จำนวนมาก และความเร็วในการทำงานของชิปไม่สูงมากนัก จึงทำให้สมาร์ตการ์ดชนิดนี้เหมาะที่จะนำไปประยุกต์ใช้กับงานที่ข้อมูลไม่ค่อยสำคัญมากนัก เช่น บัตรลงเวลาทำงาน บัตรผ่านประตู บัตรโทรศัพท์ ฯลฯ

2.9.2 การ์ดชนิดโปรเซสเซอร์

สมาร์ตการ์ดชนิดโปรเซสเซอร์ (Processor card) หรืออีกชื่อหนึ่งว่า Asynchronous card เป็นสมาร์ตการ์ดที่ได้รับการปรับปรุงจากสมาร์ตการ์ดชนิดหน่วยความจำ ด้วยการใส่เทคโนโลยีไมโครโปรเซสเซอร์เข้าไปในชิป เพื่อให้ชิปสามารถประมวลผลข้อมูล และเพิ่มความปลอดภัยให้แก่ข้อมูลได้สูงขึ้น การที่ไมโครโปรเซสเซอร์ลงในชิปทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำสำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผล ข้อมูล นอกจากนี้ยังมี การใส่ชิป ประมวลผล ทาง คณิตศาสตร์ ลงใ นชิปสมาร์ตการ์ดเพื่อช่วยให้การประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้า-ถอดรหัส ทำให้สมาร์ตการ์ดชนิดโปรเซสเซอร์มีความเร็วในการทำงานสูงกว่าชนิดหน่วยความจำหลายเท่า ดังรูปที่ 2.12

รูปที่ 2.10 รูปด้านหน้าของบัตรชนิดโปรเซสเซอร์ (Processor Card)

รูปที่ 2.11 รูปด้านหลังของบัตรชนิดโปรเซสเซอร์ (Processor Card)



รูปที่ 2.12 บล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ตการ์ดชนิดโปรเซสเซอร์

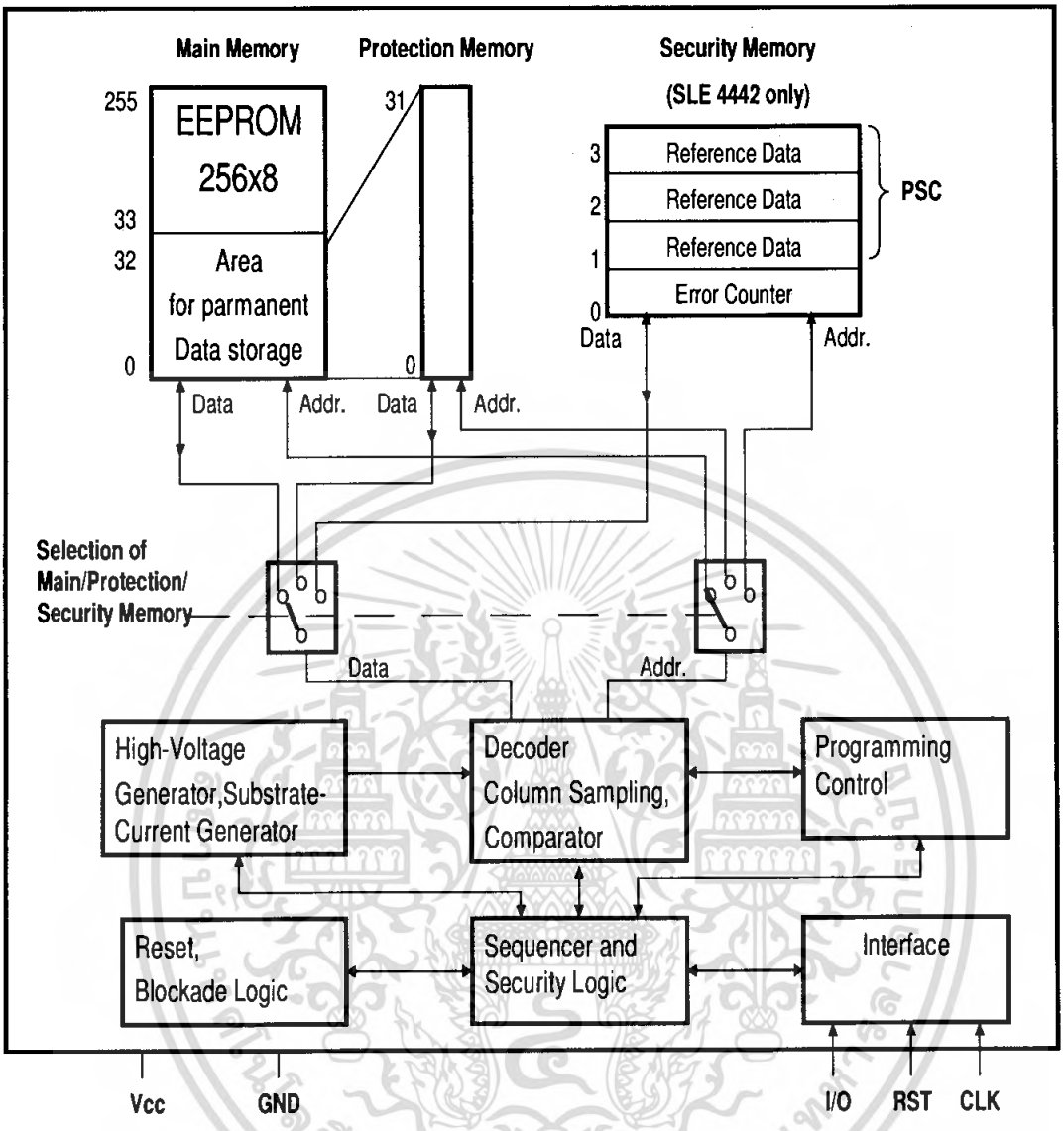
ในการรับส่งข้อมูลให้กับสมาร์ตการ์ดชนิดนี้ จะใช้หน้าสัมผัสเดียวกับสมาร์ตการ์ดชนิดหน่วยความจำ โดยสัญญาณนาฬิกาที่ป้อนจะถูกใช้เป็นสัญญาณนาฬิกาให้แก่โปรเซสเซอร์ภายในสมาร์ตการ์ด ข้อมูลที่รับส่งจึงไม่จำเป็นต้องสัมพันธ์กับสัญญาณนาฬิกาที่ป้อนให้แก่ชิปเพียงกำหนดอัตราการรับ-ส่งข้อมูลเป็น 9600 บิต/วินาที ก็จะสามารถติดต่อกับโปรเซสเซอร์ของชิปได้แล้ว แต่การเข้าถึงข้อมูลจะไม่สามารถทำได้เหมือนอย่างในสมาร์ตการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตาม เพราะหน่วยความจำจะอยู่ภายในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว ข้อดีอย่างหนึ่งที่ไม่สามารถติดต่อกับหน่วยความจำในชิปได้โดยตรงก็คือ การลอบเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตแทบเป็นไปไม่ได้ ยกเว้นมีความบกพร่องในการกำหนดเงื่อนไขในการเข้าถึงข้อมูลที่เป็นความลับ

2.10 การ์ดที่มีระบบป้องกันข้อมูล

การ์ดที่มีระบบป้องกันความปลอดภัยข้อมูลคือ สมาร์ตการ์ดที่การอ่านข้อมูลสามารถทำได้อย่างอิสระ แต่การเขียนข้อมูลจะไม่สามารถทำได้หากไม่มีรหัสผ่านที่ถูกต้อง วิธีการในลักษณะนี้ช่วยให้ข้อมูลภายในสมาร์ตการ์ดได้รับการปกป้องและมีความน่าเชื่อถือ รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ดชนิดนี้เป็นการสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) ตามมาตรฐาน ISO 7816 ซึ่งรูปแบบคำสั่งจะแตกต่างกันไปในผู้ผลิตแต่ละราย โดยในโครงงานนี้ได้เลือกใช้สมาร์ตการ์ดเบอร์ SLE4442 เนื่องจากเป็นการ์ดที่มีคุณสมบัติในการรักษาความปลอดภัยได้อย่างครบถ้วนและสามารถหามาใช้งานได้ง่ายในบ้านเรา

2.10.1 คุณสมบัติโดยทั่วไปของสมาร์ตการ์ดเบอร์ SLE4442

- ใช้หน่วยความจำอีอีพรอม (EEPROM) 8 บิต ความจุข้อมูล 256 ไบต์
- ใช้รูปแบบของ ATR (Answer To Reset) ตามมาตรฐาน ISO 7816
- อินเตอร์เฟสแบบซิงโครนัส (Synchronous) ตามมาตรฐาน ISO 7816
- ป้องกันการเขียนข้อมูลด้วยรหัสผ่าน PSC (Programmable Security Code)
- การลบและเขียนข้อมูลในแต่ละไบต์ใช้เวลาเพียง 2.5 วินาที
- มีฟังก์ชันป้องกันข้อมูลในพื้นที่หน่วยความจำ 32 ไบต์แรก โดยสามารถที่จะกำหนดให้ข้อมูลที่เขียนลงไปยังพื้นที่ช่วงดังกล่าวถูกเขียนลงไปอย่างถาวรได้

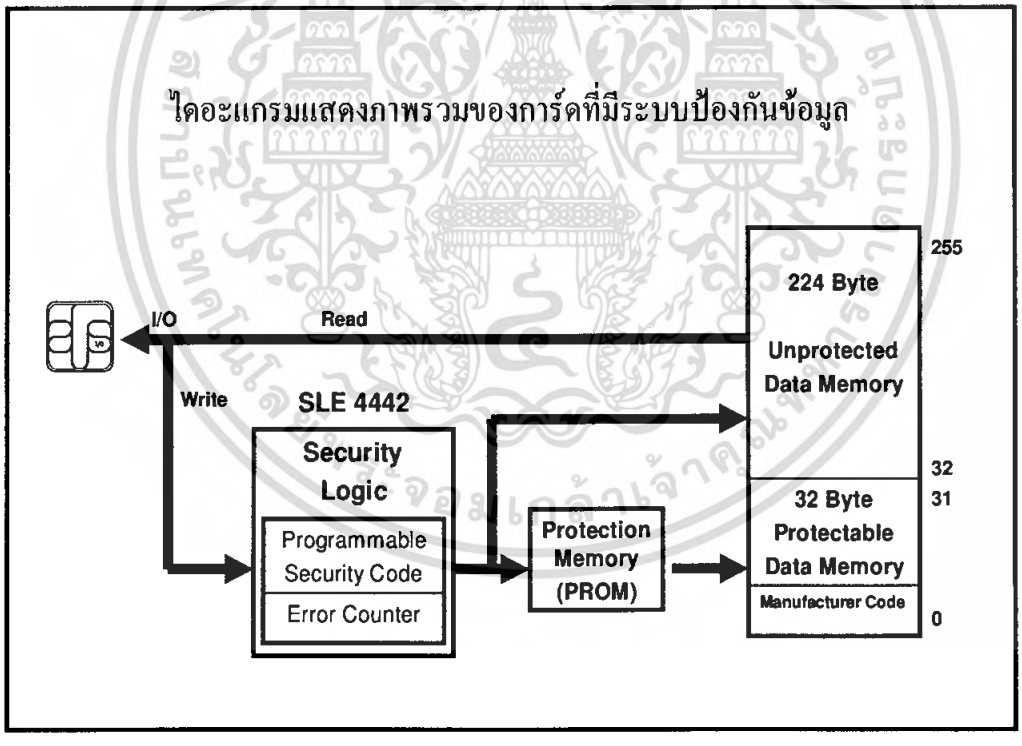


รูปที่ 2.13 บล็อกไดอะแกรมโครงสร้างภายในของสมาร์ตการ์ดเบอร์ SLE4442

จากรูปที่ 2.13 จะเห็นได้ว่าหน่วยความจำขนาด 256 ไบต์ ที่อยู่ภายในสมาร์ตการ์ดเบอร์ SLE4442 จะถูกแบ่งออกเป็น 2 ส่วนด้วยกันได้แก่ ข้อมูลในช่วง 32 ไบต์แรกซึ่งเป็นพื้นที่ที่มีระบบป้องกันการเขียนข้อมูลทับ และหน่วยความจำส่วนถัดมาซึ่งเป็นอีอีพรอม (EEPROM) ที่สามารถทั้งเขียนและอ่านได้ กลไกในการปกป้องข้อมูลของสมาร์ตการ์ดเบอร์ SLE4442 มาจากส่วนที่เป็นหน่วยความจำปลอดภัย (Security Memory) ที่ได้รับการปกป้องโดยข้อมูลสำคัญ 2 ส่วน คือ

- **ข้อมูลอ้างอิง (Reference data หรือ PSC)** เป็นข้อมูลขนาด 3 ไบต์ ที่เก็บค่าของรหัสผ่านสำหรับการเข้าไปแก้ไขข้อมูลในหน่วยความจำเอาไว้ (รหัส PSC ไม่สามารถถูกอ่านออกมาได้) รหัส PSC จะถูกกำหนดเป็นค่าที่หนึ่งมา โดยผู้ผลิตก่อนซึ่งสามารถจะมาปรับเปลี่ยนเองได้ในภายหลังเมื่อใช้งาน

- **ไบต์แสดงความผิดพลาด (Error Counter Byte)** เป็นข้อมูลที่บอกถึงจำนวนครั้งที่ป้อนรหัส PSC ผิดซึ่งถูกกำหนดเอาไว้ตายตัวว่าจะผิดได้ไม่เกิน 3 ครั้ง หากเกินกว่านั้นการ์ดนะล็อคตัวเองอย่างถาวรทันทีและไม่มีทางปลดล็อคได้ แม้ว่าจะป้อนรหัส PSC ที่ถูกต้องไปแล้วก็ตามการเขียนข้อมูลยังหน่วยความจำก็จะไม่สามารถทำได้อีกต่อไป แต่ยังคงอ่านข้อมูลออกมาได้ตามปกติ การป้อนรหัส PSC ผิดแต่ละครั้ง Error Counter จะถูกลดลง 1 ค่าทันที ถ้าหากค่า Error Counter ถูกลดลงจนมีค่าเป็น 0 เมื่อไรก็แสดงว่าการ์ดได้ถูกล็อคไปเรียบร้อยแล้ว (ในกรณีที่ป้อนรหัสถูกในครั้งที่ 3 ค่าของ Error Counter จะถูกรีเซตกลับไปเป็น 3 ครั้งเหมือนอย่างตอนแรกเริ่ม)



รูปที่ 2.14 บล็อกไดอะแกรมแสดงภาพรวมของการ์ดที่มีระบบป้องกันข้อมูล

จากรูปที่ 2.14 จะเห็นได้ว่าการอ่านข้อมูลจากหน่วยความจำนั้น เราสามารถจะอ่านข้อมูลออกมาได้โดยไม่ต้องผ่านขั้นตอนของการป้อนรหัส PSC แต่สำหรับการเขียนข้อมูลแล้วเราจะต้อง

ป้อนรหัส PSC ที่ถูกต้องเสียก่อน เพื่อเปิดลอจิกในการเขียนข้อมูลลงหน่วยความจำ นอกจากนั้นก็จะเห็นได้ว่าข้อมูล 4 ไบต์แรก เป็นข้อมูลของผู้ผลิตหรือ Manufacturer Code โดยพื้นที่ส่วนนี้ใช้เก็บข้อมูลของ ATR โดยความหมายของข้อมูลที่อยู่ในพื้นที่ส่วนนี้แต่ละไบต์จะถูกกำหนดโดยผู้ผลิตการ์ดแต่ละราย

2.10.2 รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ดเบอร์ SLE4442

รูปแบบการสื่อสารข้อมูลของสมาร์ตการ์ดเบอร์ SLE4442 เป็นการรับส่งข้อมูลระหว่างเครื่องอ่าน และสมาร์ตการ์ดแบบ 2 ทิศทาง (ข้อมูลบนสาย I/O จะถูกอ่านค่าที่ขอบขาลงของสัญญาณนาฬิกา) โดยรูปแบบการสื่อสารนี้ประกอบด้วย 4 โหมดการทำงาน ได้แก่

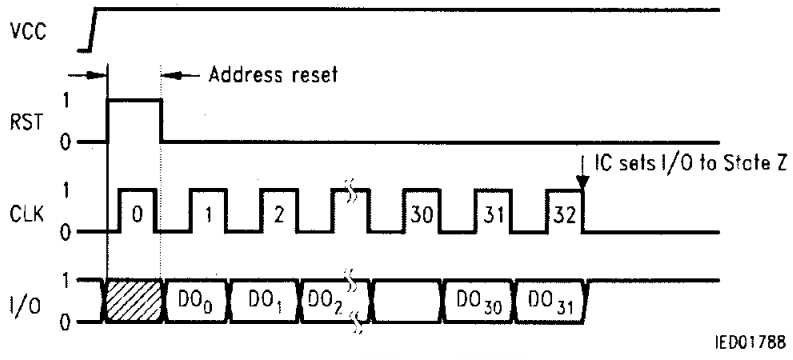
- การรีเซตและการตอบรับการรีเซต ATR (Answer To Reset)
- โหมดการส่งคำสั่ง (Command Mode)
- โหมดการอ่านข้อมูล (Outgoing Data Mode)
- โหมดการดำเนินการ (Processing Mode)

1. การรีเซตและการตอบรับการรีเซตด้วย ATR

เมื่อรีเซตการทำงานของการ์ดจะทำให้การ์ดมีการตอบรับการรีเซตด้วยข้อมูล ATR (Answer To Reset) สำหรับข้อมูล ATR ที่ตอบรับมาจากสมาร์ตการ์ดเบอร์ SLE4442 จะประกอบด้วยข้อมูล 4 ไบต์ การอ่านข้อมูลที่ว่านี้สามารถทำได้โดยอ้างอิงจากสัญญาณในรูปที่ 2.1 โดยหลังจากที่ขา RST เป็นลอจิกต่ำ เมื่อมีสัญญาณนาฬิกาลูกต่อไปเข้ามา จะทำให้เกิดสัญญาณเอาท์พุตของสมาร์ตการ์ดขึ้นที่ขา I/O จะเปลี่ยนเป็นลอจิกสูง เพื่อเป็นการบอกถึงการสิ้นสุดการรีเซต

ตารางที่ 2.1 ลักษณะของข้อมูลที่ได้จากการตอบรับการรีเซต

Byte 1	Byte 2	Byte 3	Byte 4
DO7…DO0	DO15…DO8	DO23…DO16	DO31…DO24



รูปที่ 2.15 รูปสัญญาณของการรีเซตและการตอบรับการรีเซตด้วย ATR

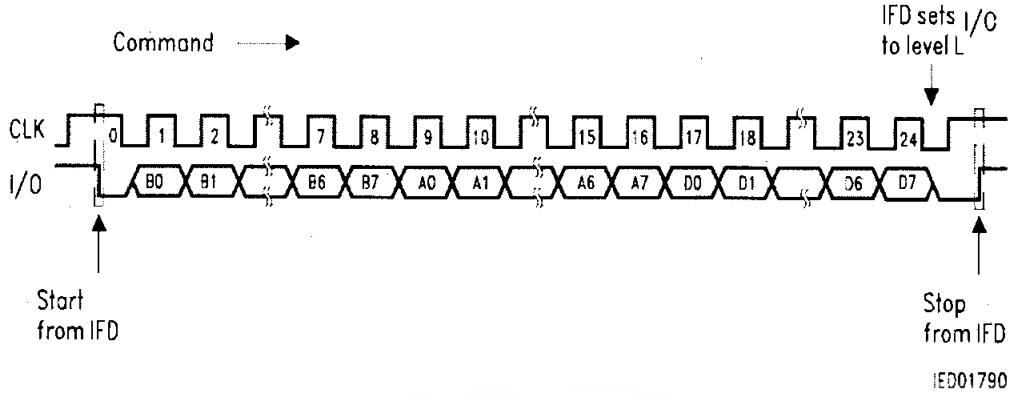
2 . โหมดการส่งคำสั่ง

การส่งคำสั่งไปยังสมาร์ตการ์ดหรือการทำงานในโหมดการส่งคำสั่ง (Command Mode) ก็คือกระบวนการต่อเนื่องหลังจากการ์ดถูกรีเซตไปเรียบร้อยแล้ว โดยการ์ดจะรอรับคำสั่งที่ส่งมาจากเครื่องอ่านซึ่งมีรูปแบบเป็นข้อมูลมีความยาว 3 ไบต์ โครงสร้างของข้อมูลดังกล่าวประกอบด้วยคำสั่ง (Command) แอดเดรส (Address) และข้อมูล (Data) โดยคำสั่งทั้งหมดที่สมาร์ตการ์ดเบอร์ SLE4442 รองรับถูกแสดง ดังตารางที่ 2.2 ส่วนรูปสัญญาณที่เกิดขึ้นระหว่างการทำงานของโหมดการส่งคำสั่งก็เป็นดังรูปที่ 2.16 จะเห็นได้ว่าการส่งข้อมูลแต่ละครั้งจะต้องมีการส่งสภาวะเริ่มต้นและสภาวะสิ้นสุดกำกับไปกับตัวข้อมูลด้วย โดยสภาวะเริ่มต้นก็คือการเปลี่ยนระดับจากลอจิกค่าสูงเป็นค่าต่ำที่ขา I/O ในขณะที่ระดับลอจิกที่ขา CLK เป็นค่าสูง ส่วนสภาวะสิ้นสุดก็คือการเปลี่ยนระดับจากลอจิกค่าต่ำเป็นสูงที่ขา I/O ในขณะที่ขา CLK เป็นค่าสูง

ตารางที่ 2.2 โครงสร้างและความหมายของชุดคำสั่งที่สมาร์ตการ์ดเบอร์ SLE4442 รองรับ

Byte								**Byte2**	**Byte3**	**Operation**	**Mode**
Control								**Address**	**Data**		
B7	**B6**	**B5**	**B4**	**B3**	**B2**	**B1**	**B0**	**A7-A0**	**D7-D0**		
0	**0**	**1**	**1**	**0**	**0**	**0**	**0**	address	no effect	**READ MAIN MEMORY**	outgoing data
0	**0**	**1**	**1**	**1**	**0**	**0**	**0**	address	Input data	**UPDATE MAIN MEMORY**	processing

0	**0**	**1**	**1**	**0**	**1**	**0**	**0**	no effect	no effect	**READ PROTECTION MEMORY**	outgoing data
0	**0**	**1**	**1**	**1**	**1**	**0**	**0**	address	Input data	**WRITE PROTACTION MEMORY**	processing
0	**0**	**1**	**1**	**0**	**0**	**0**	**1**	no effect	no effect	**READ SECURITY MEMORY**	outgoing data
0	**0**	**1**	**1**	**1**	**0**	**0**	**1**	address	Input data	**UPDATE SECURITY MEMORY**	processing
0	**0**	**1**	**1**	**0**	**0**	**1**	**1**	address	Input data	**COMPARE VERIFICATION DATA**	processing



รูปที่ 2.16 รูปสัญญาณของการส่งคำสั่งไปยังการ์ด

ตารางที่ 2.3 รูปแบบและส่วนประกอบของคำสั่ง

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

- การอ่านข้อมูลจากหน่วยความจำหลัก

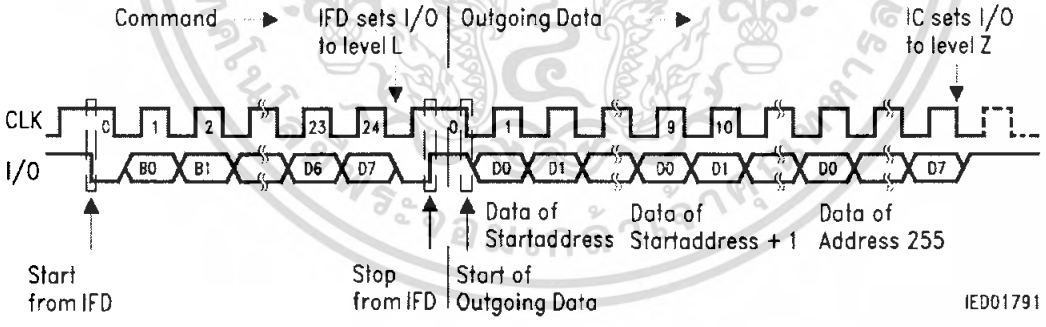
การอ่านข้อมูลจากหน่วยความจำหลัก (Read Main Memory) คือ คำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมด ออกมาจากหน่วยความจำของการ์ด ทั้งจากพื้นที่ส่วนที่ได้รับการป้องกัน (หน่วยความจำ 32 ไบต์แรก) และส่วนที่ไม่ได้รับการป้องกัน (หน่วยความจำ 224 ไบต์หลัง) โดยจะเป็นการอ่านค่า โดย เริ่มต้นจาก แอดเดรส ที่ส่ง ไปจนถึงแอดเดรสสุดท้าย (0FFH) ของพื้นที่หน่วยความจำ

ตารางที่ 2.4 ลักษณะหน่วยความจำจากหน่วยความจำหลัก

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

ตารางที่ 2.5 ลักษณะหน่วยความจำและรูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำหลัก

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



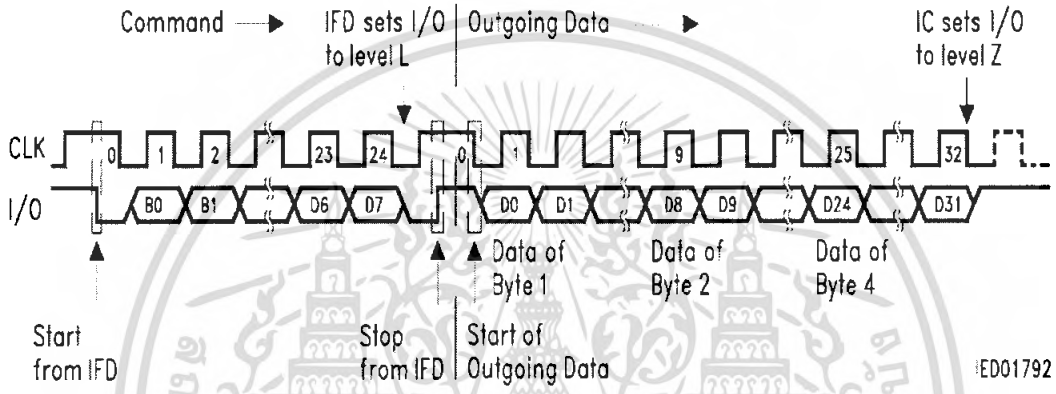
รูปที่ 2.17 รูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำหลัก

- การอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน

การอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน (Read Protection Memory) คือคำสั่งที่ใช้ในการอ่านข้อมูลทั้งหมดออกมาจากหน่วยความจำ 32 ไบต์แรก

ตารางที่ 2.6 รูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน

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



รูปที่ 2.18 รูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำที่มีการป้องกัน

- การเขียนข้อมูลลงในหน่วยความจำหลัก

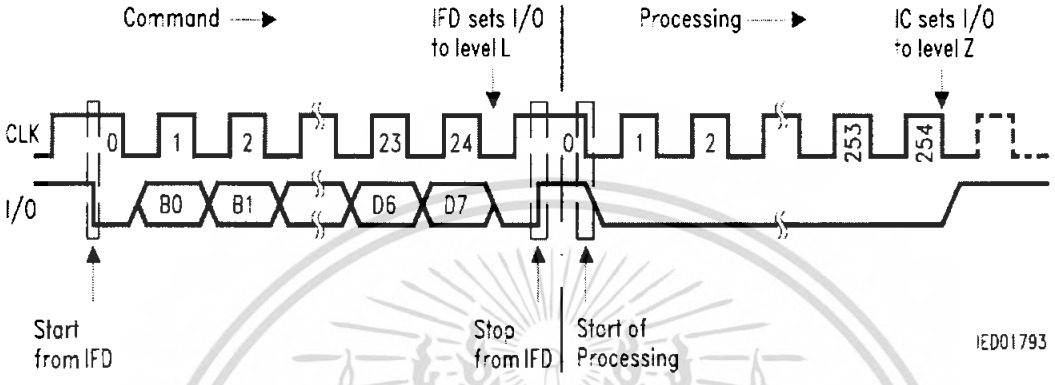
การเขียนข้อมูลลงในหน่วยความจำหลัก (Update Main Memory) คือคำสั่งที่ใช้ในการเขียนข้อมูลยังแอดเดรสใดๆ ของหน่วยความจำทั้ง 256 ไบต์ ในกรณีที่ใช้คำสั่งนี้ในการเขียนข้อมูลลงยังหน่วยความจำ 32 ไบต์แรก ข้อมูลจะยังคงแก้ไขเปลี่ยนแปลงได้ภายหลัก

ตารางที่ 2.7 รูปแบบการเขียนคำสั่งในการเขียนข้อมูลลงในหน่วยความจำหลัก

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

สำหรับการเขียนข้อมูลจะประกอบด้วย 3 เงื่อนไข คือ

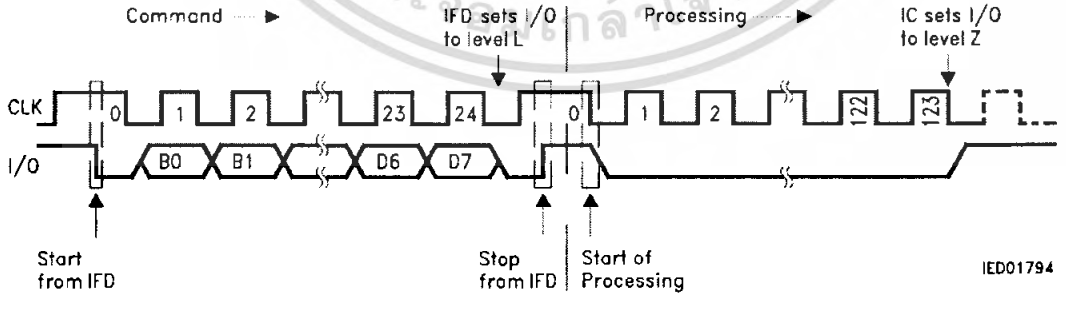
1 การลบข้อมูลที่แอดเดรสของ หน่วยความจำที่กำหนดให้เป็น 0FFH แล้วทำการเขียนข้อมูลซ้ำลงยังแอดเดรสเดิม กระบวนการนี้ต้องใช้เวลา 5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 255 ลูก



รูปที่ 2.19 รูปสัญญาณการเขียนข้อมูลในหน่วยความจำหลักแบบลบข้อมูลแล้วเขียนข้อมูลซ้ำ

2 การเขียนข้อมูลที่แอดเดรสของหน่วยความจำที่กำหนดโดยไม่ต้องลบข้อมูลออก สำหรับในกรณีนี้แอดเดรส ดังกล่าว จะต้องเป็นที่ว่าง (มีค่าข้อมูลเป็น 0FFH) อยู่ก่อนหน้านี้ แล้ว เท่านั้น กระบวนการนี้จะใช้เวลา 2.5 มิลลิวินาที หรือเท่ากับสัญญาณนาฬิกา 124 ลูก

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



รูปที่ 2.20 รูปสัญญาณการเขียนข้อมูลในหน่วยความจำหลักแบบการลบหรือเขียน

- การเขียนข้อมูลลงหน่วยความจำที่มีการป้องกัน

การเขียนข้อมูลลงหน่วยความจำที่มีการป้องกัน (Write Protection Memory) คือ การเขียนข้อมูลลงยังแอดเดรสของ หน่วยความจำใดๆ ใน 32 ไบต์แรก คำสั่งนี้มีเงื่อนไขว่าข้อมูลที่เขียนลงไปจะถูก เขียนลงยังแอดเดรสของหน่วยความจำที่กำหนดอย่างถาวร ไม่สามารถแก้ไขเปลี่ยนแปลงอะไรได้อีก สำหรับรูปแบบสัญญาณของกระบวนการนี้อ้างอิงได้จากรูปสัญญาณของการเขียนข้อมูลลงในหน่วยความจำหลัก (Update Main Memory)

ตารางที่ 2.8 รูปแบบคำสั่งในการเขียนข้อมูลลงในหน่วยความจำที่มีการป้องกัน

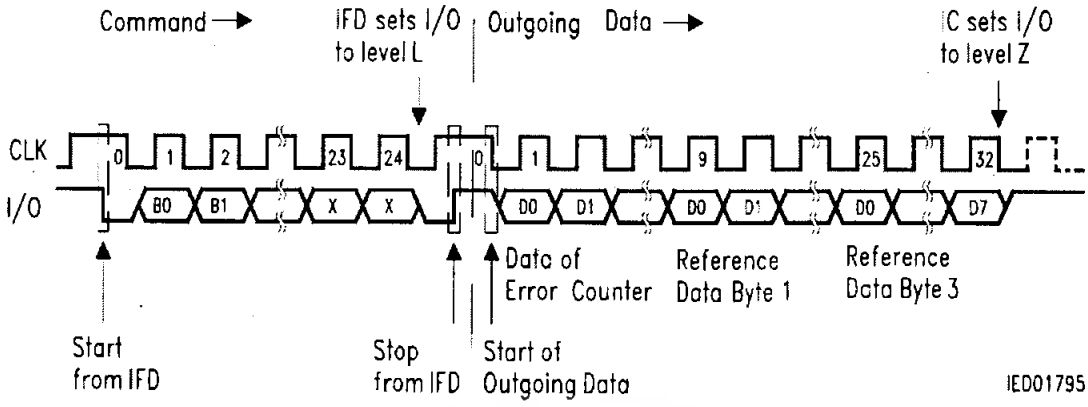
	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	1	0	0	Address	Input data
Hexadecimal	$3C_H$								$00_H...1F_H$	Input data

- การอ่านข้อมูลจากหน่วยความจำปลอดภัย

การอ่านข้อมูลจากหน่วยความจำปลอดภัย (Read Security Memory) คือ การอ่านค่าของ Error Counter เพื่อตรวจดูว่าการ์ดใบนั้น ๆ ได้ถูกล็อคไปแล้วหรือยัง โดยค่าภายในบิต D2 , D1 และ D0 ของ Error Counter จะเป็นส่วนที่บอกถึงสถานะของการ์ดใบนั้น ๆ หากค่าของบิต D2 , D1 และD0 เป็น 0 ทั้งหมด ก็แสดงว่าการ์ดได้ถูกล็อคไปแล้ว ซึ่งจะไม่สามารถแก้ไขอะไรได้ และจะไม่สามารถเขียนข้อมูลลงยังการ์ดนั้นได้อีกต่อไป (แต่ว่าการอ่านข้อมูลในการ์ดจะยังคงทำได้ตามปกติ)

ตารางที่ 2.9 รูปแบบคำสั่งในการอ่านข้อมูลจากหน่วยความจำปลอดภัย

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



รูปที่ 2.21 รูปสัญญาณของการอ่านข้อมูลจากหน่วยความจำปลอดภัย

- การเขียนข้อมูลลงในหน่วยความจำปลอดภัย

การเขียนข้อมูลลงในหน่วยความจำปลอดภัย (Update Security Memory) คือ การเข้าไปแก้ไขข้อมูลของรหัส PSC ภายในการ์ด หรืออาจกล่าวได้ว่าเป็นการเข้าไปเปลี่ยนรหัสป้องกันของการ์ดนั้นเอง คำสั่งจะถูกกระทำต่อเมื่อมีการส่งรหัส PSC ที่ถูกต้องไปยังการ์ดเสียก่อนโดยในกรณีที่ป้อนรหัสผิด ค่าของบิต D2 , D1 และD0 ใน Error Counter จะค่อย ๆ ถูกเปลี่ยนจากค่า "1" เป็น "0" ไล่ไปทีละบิตตามจำนวนครั้งที่ป้อนผิด หากทั้งหมดกลายเป็นศูนย์เมื่อไรการ์ดก็จะถูกล็อคทันทีซึ่งนั่นหมายความว่าโอกาสที่ป้อนรหัสผิดมีเพียง 3 ครั้งเท่านั้น สำหรับรูปสัญญาณของกระบวนการนี้จะเหมือนกับ รูปสัญญาณของการเขียน ข้อมูลลง ในหน่วย ความจำหลัก

ตารางที่ 2.10 รูปคำสั่งในการเขียนข้อมูลลงในหน่วยความจำปลอดภัย

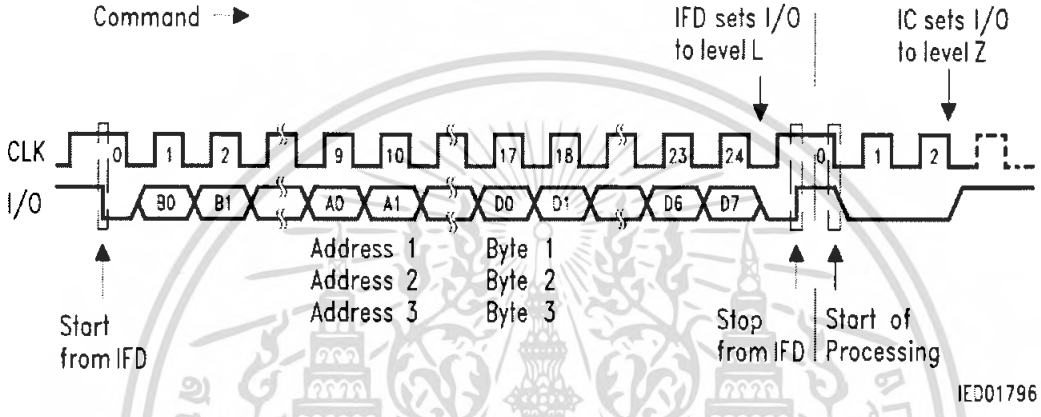
	Control								Address	Data
	B7	B6	B5	B4	B3	B2	B1	B0	A7...A0	D7...D0
Binary	0	0	1	1	1	0	0	1	Address	Input data
Hexadecimal	39_H								$00_H...03_H$	Input data

- การเปรียบเทียบและพิสูจน์ข้อมูล

การเปรียบเทียบและพิสูจน์ข้อมูล (Compare Verification Data) คือ การสั่งให้การ์ดทำการเปรียบเทียบรหัส PSC กับรหัสผ่านที่เราส่งไปยังการ์ด ในการเปรียบเทียบที่ว่านี้ ข้อมูลที่การ์ดจะส่งกลับคือค่าของ Error Counter ที่จะบอกเราว่ารหัสที่เราป้อนนั้นถูกต้องหรือไม่ และยังเหลือโอกาสพลาดอีกกี่ครั้งเท่านั้น (โดยเราจะไม่สามารถเข้าไปอ่านรหัส PSC ของการ์ดออกมาได้)

ตารางที่ 2.11 รูปแบบคำสั่งในการเปรียบเทียบและพิสูจน์ข้อมูล

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



รูปที่ 2.22 รูปสัญญาณของการเปรียบเทียบและพิสูจน์ข้อมูล

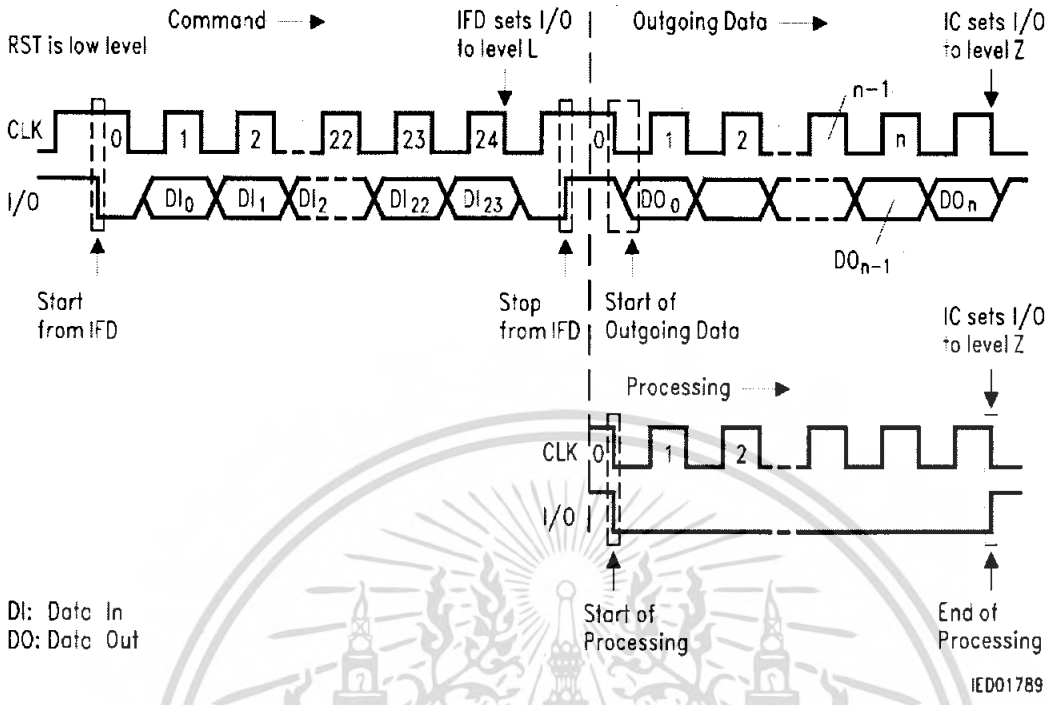
3. โหมดการอ่านข้อมูล

โหมดการอ่านข้อมูล (Outgoing Data Mode) นี้จะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขออ่านข้อมูลไปยังสมาร์ตการ์ด เพื่อขออ่านข้อมูลจากพื้นที่ใดๆ ในหน่วยความจำ หลังจากที่ได้รับคำสั่งดังกล่าว สมาร์ตการ์ดจะส่งข้อมูลที่ถูกร้องขอกลับมายังเครื่องอ่าน ซึ่งก็เท่ากับว่าเครื่องอ่านจะสามารถอ่านข้อมูลที่ต้องการออกมาได้สำเร็จจากโหมดการทำงานนี้

4. โหมดดำเนินการ

โหมดดำเนินการ (Processing Mode) จะเกิดขึ้นหลังจากที่มีการส่งคำสั่งในกลุ่มของการขอเขียนหรือลบข้อมูลออกจากพื้นที่ใดๆ ในหน่วยความจำ โดยหลังจากที่ได้รับคำสั่งดังกล่าว สมาร์ตการ์ดจะเริ่มดำเนินกระบวนการตามที่ได้รับการสั่งมา ในโหมดการทำงานนี้ข้อมูลจากขา I/O จะไม่ถูกนำมาใช้ร่วมในการทำงานเลย (โดยจะมีสถานะเป็นลอจิกต่ำตลอดทั้งช่วง)

โหมดการส่งคำสั่ง โหมดการอ่านข้อมูล และโหมดการดำเนินการ ซึ่งเรียกรวมกันว่าเป็นโหมดการประมวลผล (Operational Modes) ซึ่งรูปสัญญาณของโหมดการประมวลผลแสดงดังรูปที่ 2.23



รูปที่ 2.23 รูปสัญญาณของโหมดการประมวลผล

2.11 โปรแกรมวิชวลเบสิก

วิชวลเบสิก (Visual Basic) เป็นภาษาคอมพิวเตอร์ที่นิยมนำมาใช้พัฒนาโปรแกรมบน Windows เนื่องจากเป็นภาษาคอมพิวเตอร์ที่ใช้เทคโนโลยีในลักษณะ Visualize ซึ่งเพียงแต่เลือก Control ที่เหมาะสมแล้ววางลงบน Form ก็สามารถที่จะสร้างจอภาพที่ใช้ สำหรับ ติดต่อกับผู้ใช้ รวมทั้งการใช้เทคนิคการเขียนโปรแกรมรวมแบบ Event-driven ซึ่งเป็นการเขียนโปรแกรมเพื่อกำหนดขั้นตอนการทำงานให้กับ Control ต่างๆ ที่สร้างขึ้นตามเหตุการณ์ต่างๆ ที่เกิดขึ้น เช่น การเลื่อนเม้าส์ หรือการรับข้อมูลจากคีย์บอร์ด ฯลฯ เป็นต้น

2.11.1 โปรแกรมติดต่อและควบคุมผ่านพอร์ตอนุกรม

Control สำคัญที่ทำให้วิชวลเบสิกสามารถติดต่อผ่านพอร์ตอนุกรมได้นั่นคือ Control MSComm โดยพร็อพเพอร์ตี้ที่สำคัญในการใช้งาน MSComm มีดังนี้

- **CommPort** ใช้ในการกำหนดหมายเลขของพอร์ตอนุกรมที่เราต้องการติดต่อ โดยรูปแบบการใช้งาน ดังนี้

  Object.Commport = value

  ยกตัวอย่างเช่น ถ้าเขียนโปรแกรมติดต่อกับพอร์ต Com1 จะเขียน

เป็น

MSComm1.CommPort = 1

- **Setting** ใช้กำหนดอัตราบอด (Baud Rate) หรือความเร็วในการส่งข้อมูล มีหน่วยเป็นบิตต่อวินาที, พาริตี้, จำนวนของบิตข้อมูล, จำนวนของบิตปิดท้ายโดยมีรูปแบบการใช้งานดังนี้

  Object.Setting = value

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

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

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

  Object.PortOpen = value

- **InBuffersize** เป็นการกำหนดขนาดของ Buffer ในการรับข้อมูลเข้ามา โดยมีรูปแบบการทำงานดังนี้

  Object.InBuffersize = value

- **OutBuffersize** เป็นการกำหนดขนาดของ Buffer ในการส่งข้อมูลออกไป โดยมีรูปแบบการทำงานดังนี้

  Object.OutBuffersize = value

- **Inputlen** เป็นการกำหนดค่าของข้อมูลที่อ่านจาก Buffer ภาครับโดยมีรูปแบบการทำงานดังนี้

  Object.Inputlen = value

- **InputMode** เป็นการกำหนดค่าชนิดของข้อมูลที่รับเข้ามา ถ้าข้อมูลที่เข้ามาเป็นข้อความปรกติจะต้องกำหนด value = เป็น "0" และถ้าข้อมูลที่เข้ามาเป็นข้อมูลไบนารี่จะต้องกำหนด value = "1" โดยมีรูปแบบการกำหนดค่าดังนี้

  Object.InputMode = value

- **Input** ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบการส่งดังนี้

  Value = Object.Input

- **Output** ใช้ในการส่งข้อมูลออกไปจากพอร์ตอนุกรม โดยมีรูปแบบการส่งดังนี้

  Object.Output = value

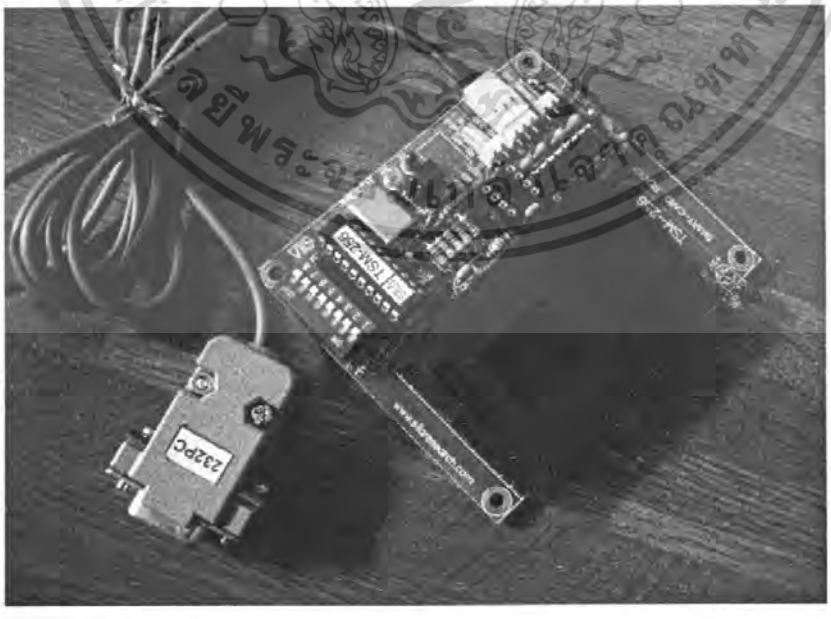
- **EOFEnable** เป็นการบอกการสิ้นสุดของไฟล์ End of File (EOF) โดยมีรูปแบบการใช้งานดังนี้

  Object.EOFEnable = value

2.11.2 โปรแกรมเพื่อสร้างระบบฐานข้อมูล

ในการติดต่อกับฐานข้อมูลจะใช้ ADO Data Control ซึ่งเป็น Control ใหม่ที่ปรากฏใน Visual Basic 6.0 ที่ไมโครซอฟท์พัฒนาขึ้นเพื่อรองรับแนวคิด Universal Data Access (UDA) ที่จุดมุ่งหมายที่ ต้องการให้เกิดศูนย์กลางการติดต่อระหว่าง Application ที่มีการใช้ข้อมูลที่ต่างรูปแบบกันสามารถข้อมูลร่วมกันได้ โดยไม่ต้องเปลี่ยนรูปแบบกันสามารถข้อมูลร่วมกันได้ โดยไม่ต้องเปลี่ยนรูปแบบของข้อมูลเดิมของแต่ละ Application แต่อย่างใดสำหรับ Visual Basic 6.0 นั้น สามารถติดต่อกับฐานข้อมูลได้ทุกชนิด โดยอาศัยเทคโนโลยีหลายๆ อย่าง แต่ที่นิยมใช้คือ Microsoft Access ซึ่งมีโครงสร้างแบบ Relational Database ที่มีข้อดีคือ สามารถนำข้อมูลจากหลายๆ ตาราง ที่มีความสัมพันธ์กันมาใช้งานร่วมกันได้

2.12 เครื่องอ่าน-เขียนบัตร สมาร์ตการ์ด TSM-256



รูปที่ 2.24 เครื่องอ่าน – เขียนบัตรสมาร์ตการ์ดรุ่น TSM- 256

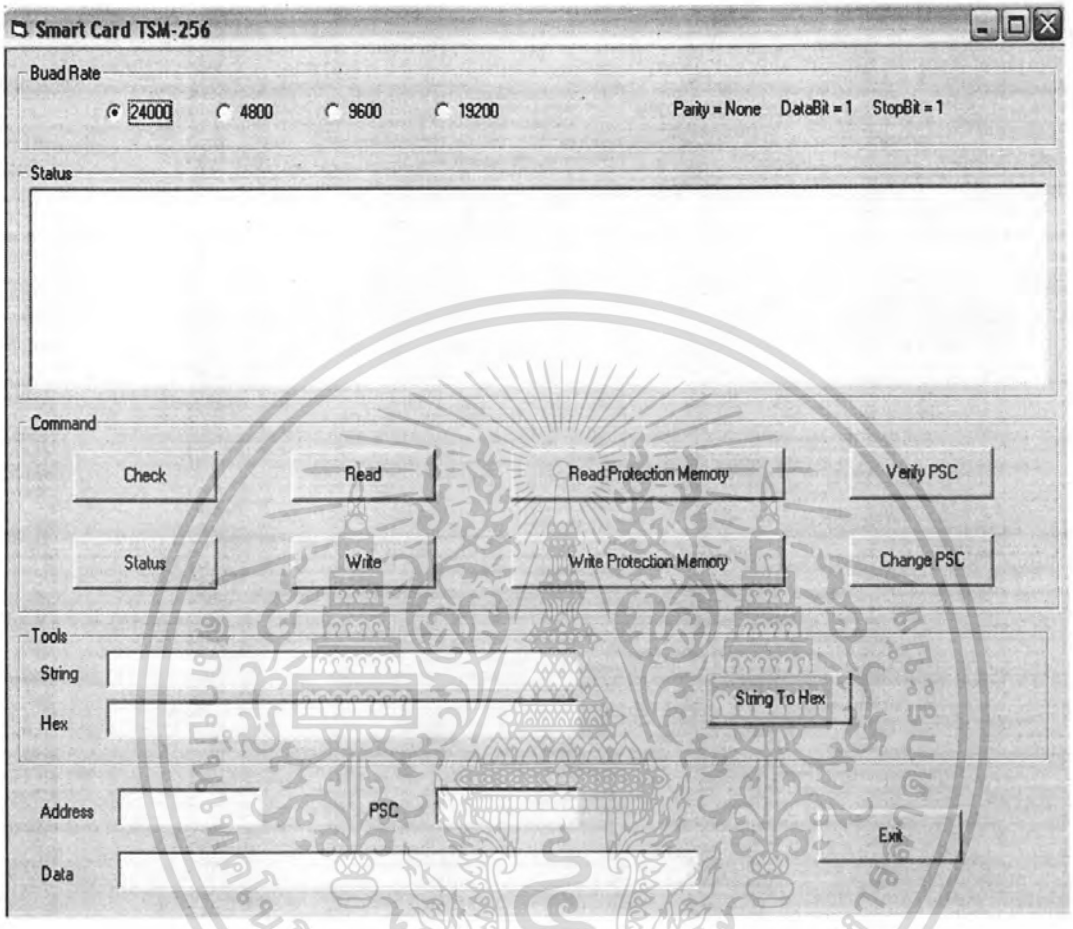
2.12.1 คุณสมบัติทั่วไป

1. ทำงานด้วยไมโครคอนโทลเลอร์เบอร์ 89C4051
2. เขียนและอ่านบัตร Smart Card เบอร์ SLE 4442 ตามมาตรฐาน ISO 7816
3. รับคำสั่งจากเครื่องคอมพิวเตอร์เพื่อเขียนและอ่านบัตรได้ โดยผ่านพอร์ทอนุกรม RS232 คุณสมบัตรการสื่อสารคือ Parity=None,Data=8,StopBit=1 กำหนดBoudrate ได้ตั้งแต่ 2400-19200
4. DIP-SWITCH 8 หลักใช้สำหรับตั้งค่าความเร็วในการสื่อสารและตั้งค่า Address กรณีใช้ระบบสื่อสารแบบ Network โดยตั้งค่าต่างๆได้ ดังตารางที่ 2.12

ตารางที่ 2.12 ค่าที่ได้จากการติดตั้งค่า Dip-Switch

SW.1 SW.2	Reserveไว้ไม่ได้ใช้งาน	**SW.6**	**SW.7**	**SW.8**	**Address**
SW.3 SW.4	กำหนด BUADRATE	OFF	OFF	OFF	0
OFF OFF	BR 2400	OFF	OFF	ON	1
OFF ON	BR 4800	OFF	ON	OFF	2
ON OFF	BR 9600	OFF	ON	ON	3
ON ON	BR 19200	ON	OFF	OFF	4
SW.5	**ชุดรับคำสั่ง**	ON	OFF	ON	5
OFF	ไม่มี Address	ON	ON	OFF	6
ON	มี Address(Network)	ON	ON	ON	7

2.12.2 โปรแกรมและชุดคำสั่งควบคุม



รูปที่ 2.25 หน้าจอควบคุมในรูปแบบของโปรแกรมวิชวลเบสิก

ชุดคำสั่งควบคุม TSM-256

TSM-256 มีคำสั่งในการติดต่อสั่งงานควบคุมทั้งหมด 8 คำสั่ง รูปแบบเป็น ASCII ทั้งหมด โดยมีลักษณะดังนี้

: คือ รหัสนำของคำสั่ง (3AH)

A คือ Address ของบอร์ดตั้งแต่ 0-7

C คือ รหัสคำสั่งตั้งแต่ 0-7

XX...X คือ ข้อมูลติดตามของแต่ละคำสั่งซึ่งอาจจะมีหรือไม่ก็ได้รวมมีความยาวตามกำหนดแต่ละคำสั่ง

<CR> คือ รหัสลงท้ายของแต่ละคำสั่ง (0DH)

คำสั่ง CHECK :0<CR> เมื่อ TSM-256 ได้รับคำสั่งนี้จะส่งข้อความแสดงชื่อสินค้าและ version ของสินค้ากลับมาดังนี้

TSM-256 V1.0<CR>

คำสั่ง STATUS :1<CR> คำสั่งนี้ TSM-256 จะทำการตรวจสอบ ว่ามีบัตรเสียบอยู่บนบอร์ดหรือไม่ หากไม่มีบัตรอยู่บนบอร์ดจะส่งข้อมูลกลับมา ดังนี้คือ ER<CR> หากมีบัตรเสียบอยู่บนบอร์ดจะส่งข้อมูลแสดง Manufacturer Code (4 Byte) กลับมาดังนี้

XXXXXXXX<CR>

เช่น เมื่อเสียบบัตร Smart card เบอร์ SLE 4442 อยู่ TSM-256 จะส่ง Manufacturer Code กลับมาคือ A2131091<CR>

คำสั่ง Read Data :2BB<CR> เป็นคำสั่งที่ใช้ในการอ่านข้อมูลของบัตรออกมา แสดงเป็นจำนวน 16 Byte โดยผู้ใช้กำหนดตำแหน่ง 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

คำสั่ง Write Data :3BBXXXX_ _XX<CR> คำสั่งนี้ใช้ในการเขียนข้อมูล ลงในบัตร Smart Card สามารถเขียนข้อมูล ความยาวสูงสุดได้ครั้งละ 16 Byte เริ่มเขียนข้อมูลที่ตำแหน่ง Address เริ่มต้น BBH ที่ได้กำหนดไว้จนถึงตำแหน่ง Address ของข้อมูล Byte สุดท้ายเมื่อเขียนข้อมูลลงในบัตรเรียบร้อยแล้ว TSM-256 จะส่งคำว่า OK<CR> กลับมา หากไม่สามารถเขียนข้อมูลลงในบัตรได้ หรือบนบอร์ด ไม่มีบัตรหรือ ไม่ใช่บัตร Smart Card เบอร์ SLE4442 จะส่งคำว่า ER<CR> กลับมา เช่นเมื่อส่งคำสั่ง :32115151515<CR> ข้อมูลลงในตำแหน่ง Address 21H ไปจนถึงข้อมูลใน ตำแหน่ง Address 24H จะมีค่าเป็น 15H ทั้งหมด หรือ เมื่อส่งคำสั่ง :30831313131<CR> ข้อมูลในตำแหน่ง Address 08H ไปจนถึงข้อมูลในตำแหน่ง Address 0BH จะมีค่าเป็น 31H ทั้งหมดแต่เป็นการเขียนแบบไม่ถาวรสามารถลบหรือแก้ไขเปลี่ยนแปลงได้

คำสั่ง Read Protection Memory :4<CR> คำสั่งนี้เป็นการอ่านค่า Bit Organization ของส่วน Protection Memory ทั้ง 32 Bits เรียงจาก 0-31 (ซ้ายไปขวา) โดยแต่ละ Bit จะแสดงค่าว่าตำแหน่ง Address ของส่วน Protection Memory (00H-20H) ตำแหน่งใดที่ยังสามารถเขียนข้อมูล ลงไปได้และตำแหน่งใดได้มีการเขียนข้อมูลถาวรไว้แล้ว โดยตำแหน่งที่มีการเขียนข้อมูลถาวรไว้นั้นจะมี ค่าเป็น 0 ส่วน ตำแหน่ง ที่ยัง สามารถ เขียน ข้อมูล ลงไปได้ จะมี ค่าเป็น 1 เช่น 00001100111111111111100000011111<CR> จะเห็นว่า Bit ในตำแหน่งที่ 0 - 3 เป็น 0 แสดงว่าตำแหน่ง Address ที่ตำแหน่ง 00H-03H มีข้อมูลถาวรอยู่ไม่สามารถเขียนข้อมูลทับลงไปได้ แต่ถ้าไม่มีบัตรนี้อยู่หรือไม่ใช่บัตรเบอร์ SLE4442 จะได้รับคำว่า ER<CR> กลับมา

คำสั่ง Write Protection Memory :5BBXXXX_XX<CR> คำสั่งนี้จะคล้ายกับคำสั่ง Write Data ต่างกันตรงที่เมื่อใช้คำสั่งนี้ตามหลังคำสั่งที่ 3 จะเป็นการเขียน ข้อมูลถาวรลงในส่วน Protection Memory โดย BBH จะเป็นตำแหน่ง Address เริ่มต้นที่จะเขียนข้อมูลลงไป เมื่อเขียนเสร็จเรียบร้อยจะส่งคำว่า OK<CR> กลับมาแต่ถ้าไม่สามารถเขียนข้อมูลลงในบัตรนี้ได้หรือบัตรไม่ใช่เบอร์ SLE 4442 หรือไม่มีบัตรอยู่ก็จะส่งคำว่า ER<CR> กลับมา เช่น ถ้าต้องการเขียนข้อมูลถาวรลงในตำแหน่ง Address 08H ถึง ข้อมูลใน ตำแหน่ง Address 0BH ให้มีค่าเป็น 31H ให้ส่งคำสั่ง :30831313131<CR> และตามด้วยหลังคำสั่ง :50831313131<CR> ให้กับ TSM-256 จะพบว่าในตำแหน่ง Address 08H ถึงข้อมูลในตำแหน่ง Address 0BH จะมีค่าเป็น 31H และไม่สามารถแก้ไขใดๆได้อีกเลย เมื่อส่งคำสั่ง :4<CR> ให้กับ TSM-256 เพื่ออ่านค่า Bit Organization ของส่วน Protection Memory Bit ใน ตำ แหน่ง ที่ 8 - 11 จะ มี ค่า เป็น 0 ดังนี้ 00001100000011111111100000011111<CR> แต่ถ้าเพียงใส่คำสั่งที่ 5 อย่างเดียวโดยไม่ได้ส่งคำสั่งที่ 3 ไปก่อนก็จะไม่มีผลกับการเปลี่ยนแปลงของข้อมูลใดๆเลย

คำสั่ง Verify PSC :6PPPPPP<CR> คำสั่งนี้ใช้ในการตรวจสอบ Verify ค่า PSC (Programmable Security Code) ซึ่งเป็นรหัสขนาด 3 Byte (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 เมื่อได้ทำการ Verify ค่า PSC ได้ถูกต้อง การตรวจสอบ Verify ค่า PSC ได้ถูกต้อง การตรวจสอบ Verify ค่า PSC สามารถทำได้ 3 ครั้ง ถ้าตรวจสอบ Verify ค่า PSC ไม่ถูกต้องจนถึงครั้งที่ 3 (TSM-256 จะส่งคำว่า E3<CR> กลับมา) บัตรนี้จะไม่สามารถเขียนข้อมูลใดๆลงไปได้อีก บัตรใหม่ที่ผลิตจากโรงงานนั้นค่า PSC Code คือ FFFFFF

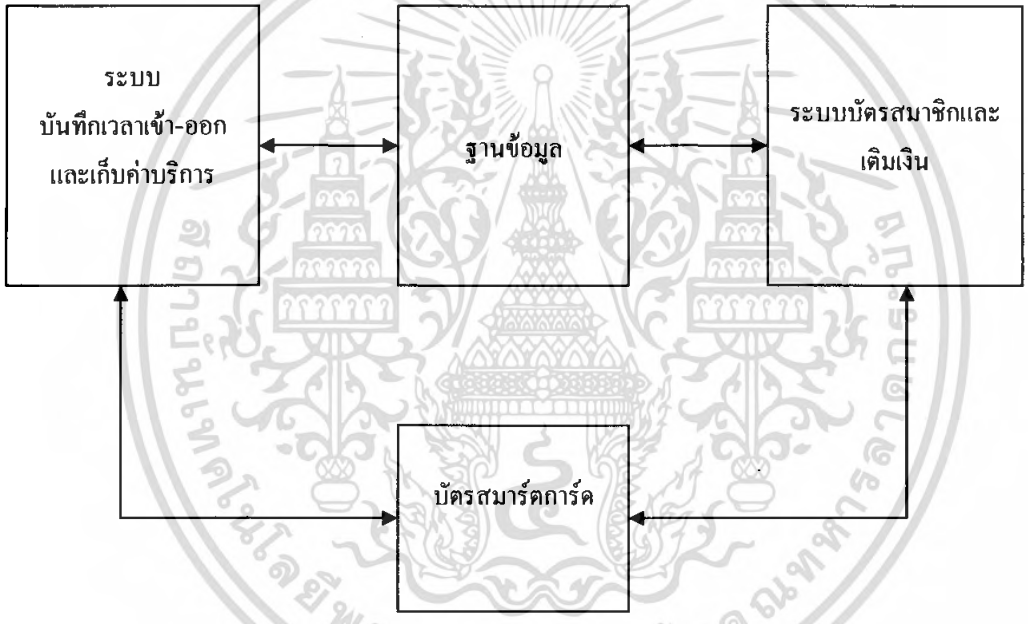
การ Verify หลังจากจ่ายไฟเข้าบัตรนี้ถ้าถูกต้องจะมีผลตลอดไปจนกว่าจะดึงบัตรออก ถึงแม้ว่าจะส่งคำสั่งไป Verify ค่า PSC ที่ไม่ถูกต้องอีกครั้งก็จะไม่มีผลใดๆทั้งสิ้นเพราะถือว่าได้ทำการ Verify ค่า PSC ถูกต้องตรงกันกับค่าในบัตรไปแล้ว

คำสั่ง CHANGE PSC :7PPPPPP<CR> คำสั่งนี้ใช้ในการเปลี่ยนค่า PSC โดยข้อมูล PPPPPPH ที่ตามหลังรหัสคำสั่งจะเป็นค่า PSC Code ที่ต้องการกำหนด ขึ้นมาใหม่เมื่อ TSM-256 ได้ทำการเปลี่ยนค่าเสร็จเรียบร้อยจะส่งคำว่า OK<CR> กลับมาแต่ถ้าไม่สามารถเปลี่ยนค่าได้หรือไม่ใช่บัตรเบอร์ SLE4442 หรือไม่มีบัตรก็จะส่งคำว่า ER<CR> มาแทน

หมายเหตุ คำสั่งที่ 3 ,5 ,7 ทั้ง 3 คำสั่งนี้จะต้องกระทำคำสั่งที่ 6 ก่อนเสมอ คือทำการตรวจสอบ Verify ค่า PSC ให้ตรงกันกับค่า PSC ในบัตรก่อนจึงจะสามารถทำงานในคำสั่งเหล่านี้ได้

บทที่ 3
การออกแบบและพัฒนา

ในโครงงานการประยุกต์ใช้บัตรสมาร์ตการ์ดนั้น ได้นำมาประยุกต์ใช้กับระบบจอดรถ โดยแบ่งการทำงานของระบบจอดรถเป็น 2ส่วน คือ ส่วนแรกเป็นส่วนของการบันทึกเวลาเข้า-ออกคำนวณค่าจอดรถและเก็บค่าบริการ ส่วนที่สองเป็นส่วนของการทำบัตรสมาชิก และเติมเงิน ทั้งสองระบบนี้ใช้ฐานข้อมูลร่วมกันพัฒนาขึ้นโดยใช้ Microsoft Access และตัวโปรแกรมพัฒนาขึ้นโดยใช้ภาษา Visual Basic 6.0

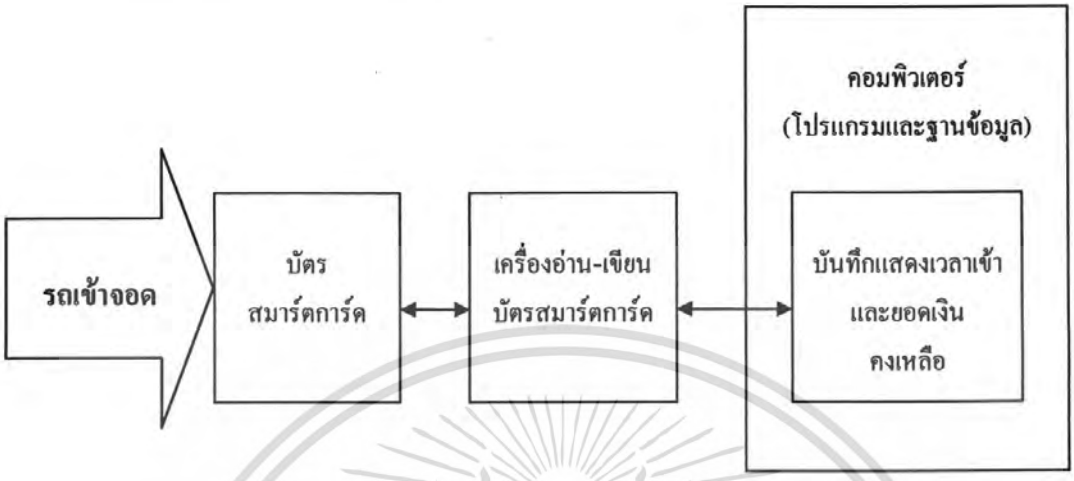


รูปที่ 3.1 บล็อกไดอะแกรมของระบบจอดรถทั้งระบบ

3.1 การออกแบบและพัฒนาระบบจอดรถบันทึกเวลาเข้า -ออก และเก็บค่าบริการ

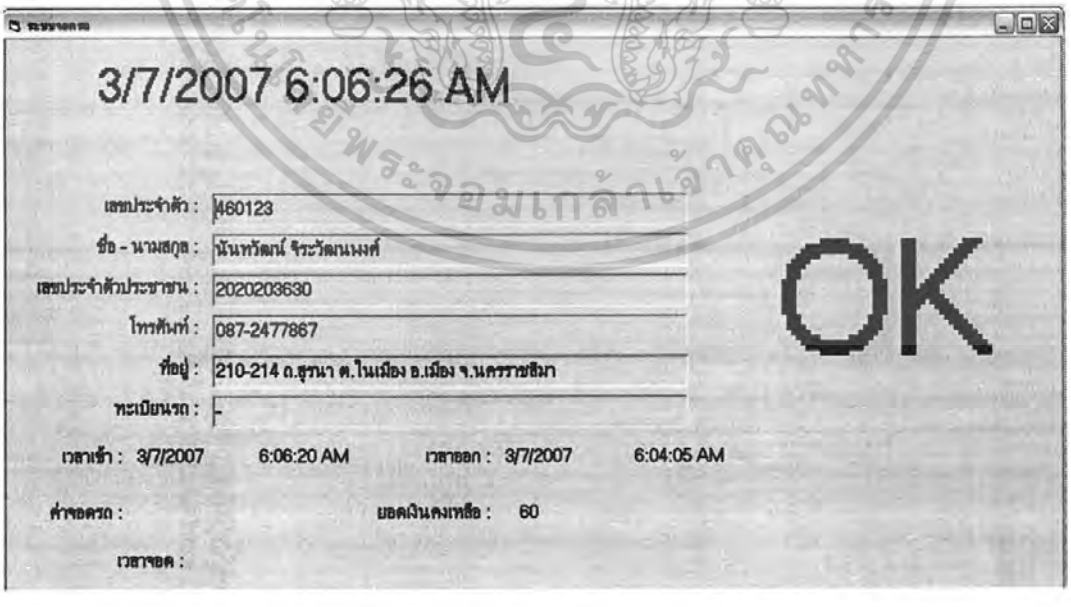
ในระบบจอดรถนี้ถูกออกแบบให้ผู้ใช้บริการที่นำรถเข้ามาจอดหรือนำรถออกสามารถติดต่อกับระบบจอดรถ โดยมีบัตรสมาร์ตการ์ดเป็นสื่อกลางผ่านเครื่องอ่านบัตรสมาร์ตการ์ดสู่คอมพิวเตอร์ (โปรแกรมและฐานข้อมูล) โปรแกรมจะแสดงข้อมูลของผู้ใช้ บันทึกเวลาเข้า-ออกลงในฐนข้อมูล พร้อมทั้งคำนวณค่าบริการและหักค่าบริการจากบัตร

3.1.1 ส่วนบันทึกและแสดงเวลาเข้าจอด



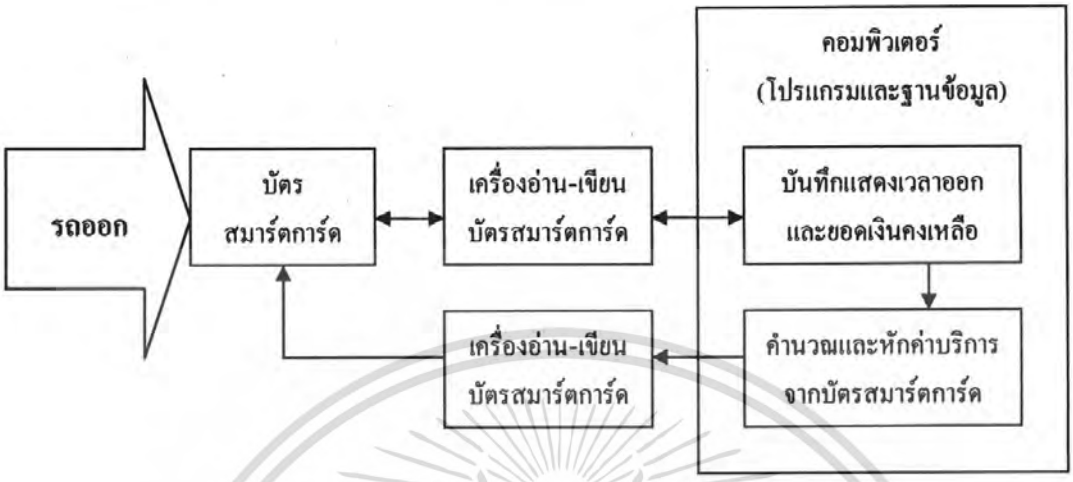
รูปที่ 3.2 บล็อกไดอะแกรมแสดงส่วนบันทึกเวลาเข้าจอด

เมื่อนำรถเข้าจอด ผู้ใช้บริการจะนำบัตรเสียบเข้ากับเครื่องอ่านบัตร เครื่องอ่านบัตรจะทำการอ่านรหัสจากบัตรสมาร์ตการ์ด ส่งไปที่โปรแกรมเพื่อบันทึกเวลาเข้าในฐานข้อมูล พร้อมแสดงข้อมูลผู้ใช้ เวลาเข้า และยอดเงินคงเหลือในบัตร



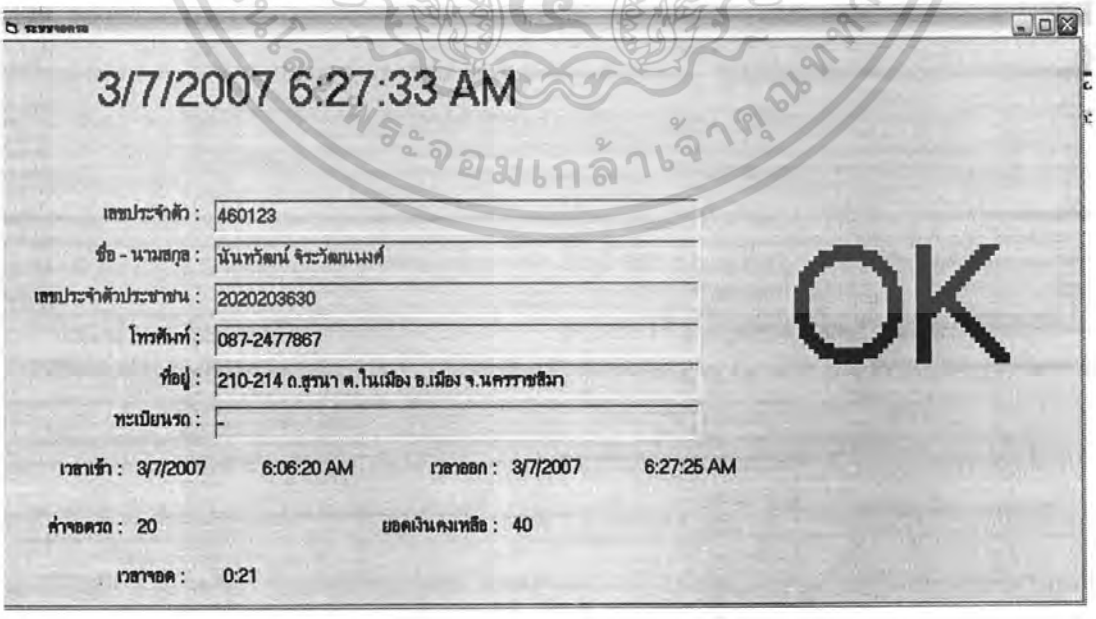
รูปที่ 3.3 หน้าจอของโปรแกรมบันทึกเวลาเข้าจอด

3.1.2 ส่วนบันทึก แสดงเวลาออกและเก็บค่าบริการ



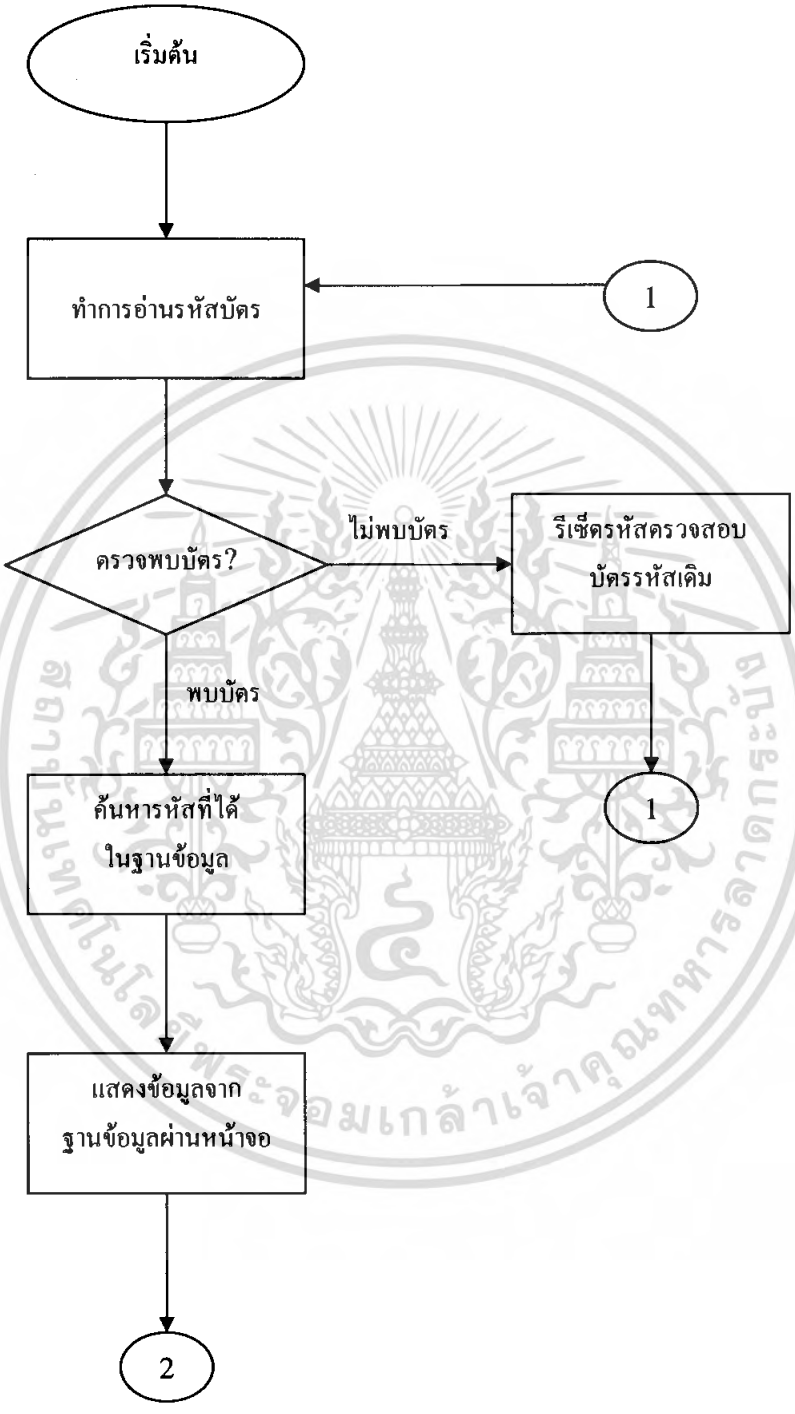
รูปที่ 3.4 บล็อกไดอะแกรมแสดงส่วนบันทึกเวลาออก คำนวณและหักค่าบริการ

เมื่อนำรถออกจากลานจอด ผู้ใช้บริการจะนำบัตรเสียบเข้ากับเครื่องอ่านบัตร เครื่องอ่านบัตรจะทำการอ่านรหัสจากบัตรสมาร์ตการ์ด ส่งไปที่โปรแกรมเพื่อบันทึกเวลาออกในฐานข้อมูล แสดงข้อมูลผู้ใช้ เวลาออก ทำการคำนวณระยะเวลาที่จอดพร้อมทั้งค่าบริการ แล้วทำการหักค่าบริการจากยอดเงินในบัตรผ่านเครื่องอ่าน-เขียนบัตรสมาร์ตการ์ด และแสดงยอดเงินคงเหลือ

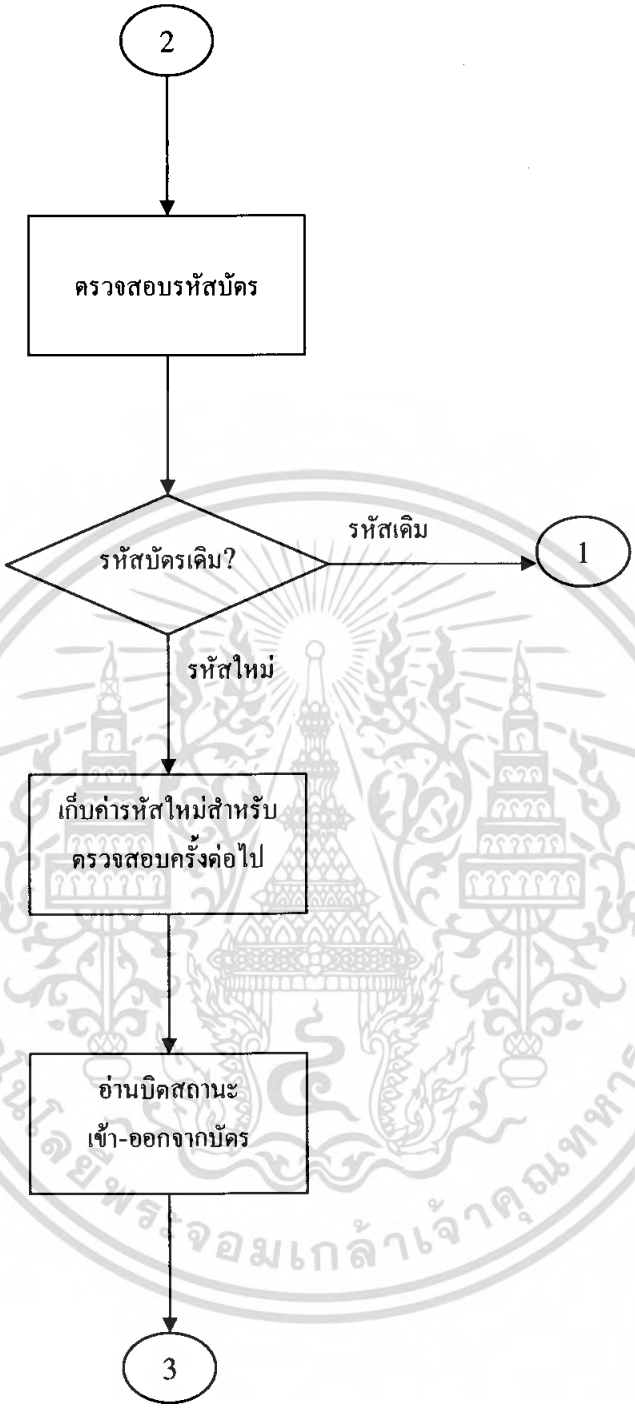


รูปที่ 3.5 หน้าจอของโปรแกรมบันทึกเวลาออก คำนวณและหักค่าบริการ

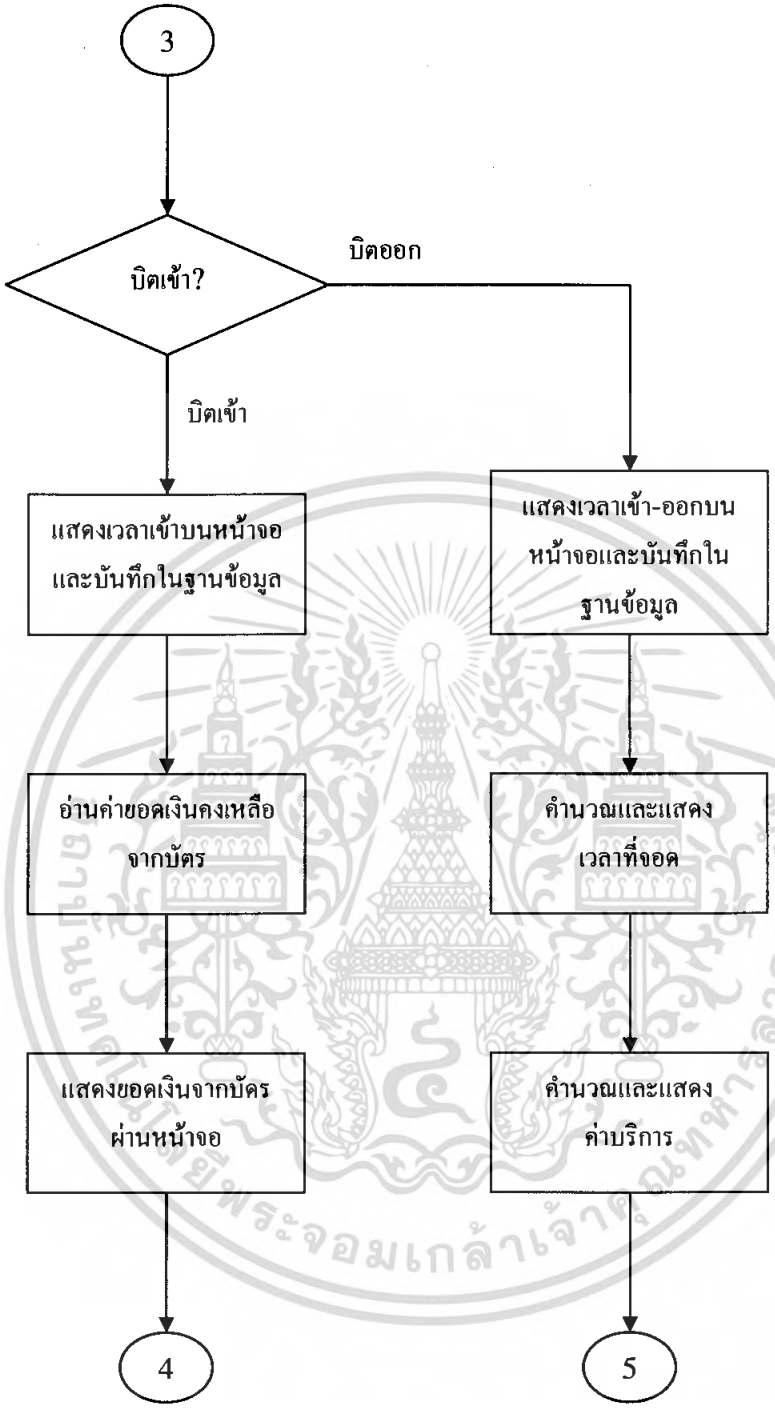
3.1.3 โปรแกรมระบบบันทึกเวลาเข้า-ออกและเก็บค่าบริการ



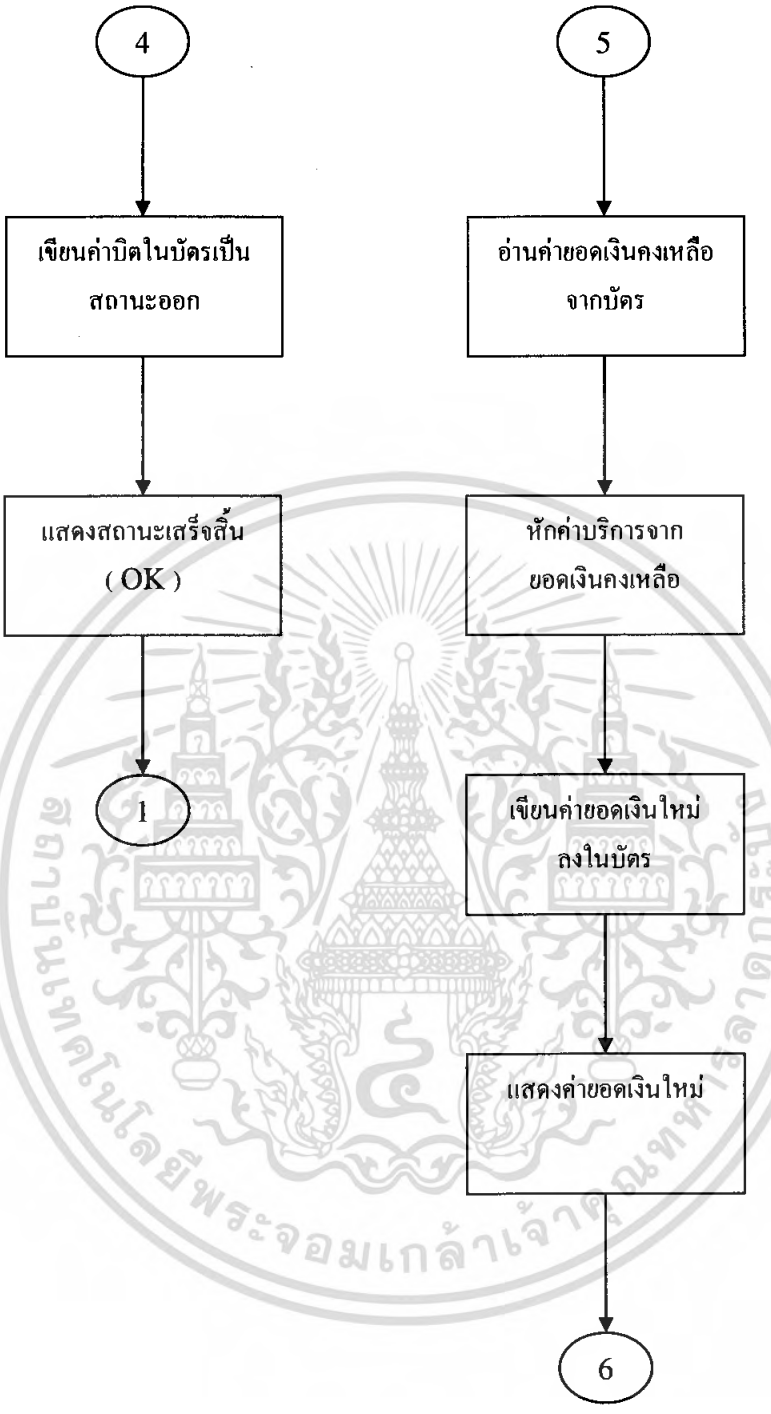
รูปที่ 3.6 โฟลวชาร์ทการทำงานของระบบบันทึกเวลาเข้า - ออกและเก็บค่าบริการ



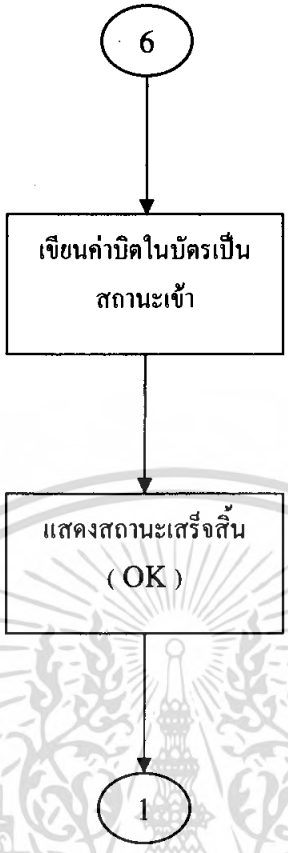
รูปที่ 3.7 โฟลวชาร์ทการทำงานของระบบบันทึกเวลาเข้า - ออกและเก็บค่าบริการ (ต่อ)



รูปที่ 3.8 โฟลวชาร์ทการทำงานของระบบบันทึกเวลาเข้า - ออกและเก็บค่าบริการ (ต่อ)



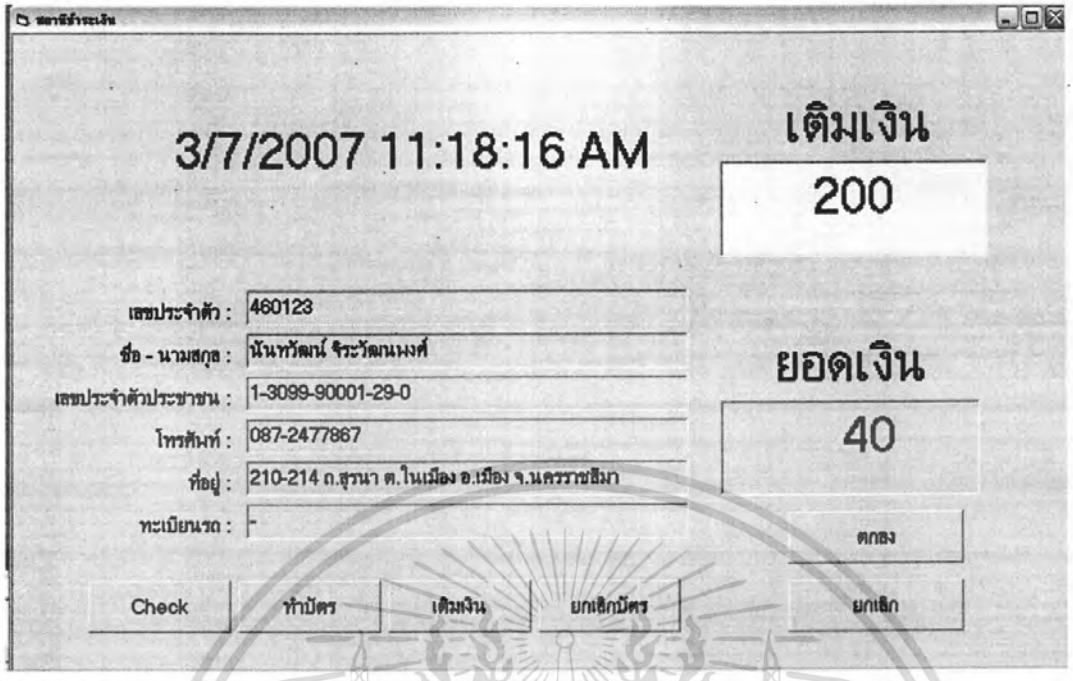
รูปที่ 3.9 โฟลวชาร์ทการทำงานของระบบบันทึกเวลาเข้า- ออกและเก็บค่าบริการ (ต่อ)



รูปที่ 3.10 โฟลวชาร์ทการทำงานของระบบบันทึกเวลาเข้า - ออกและเก็บค่าบริการ (ต่อ)

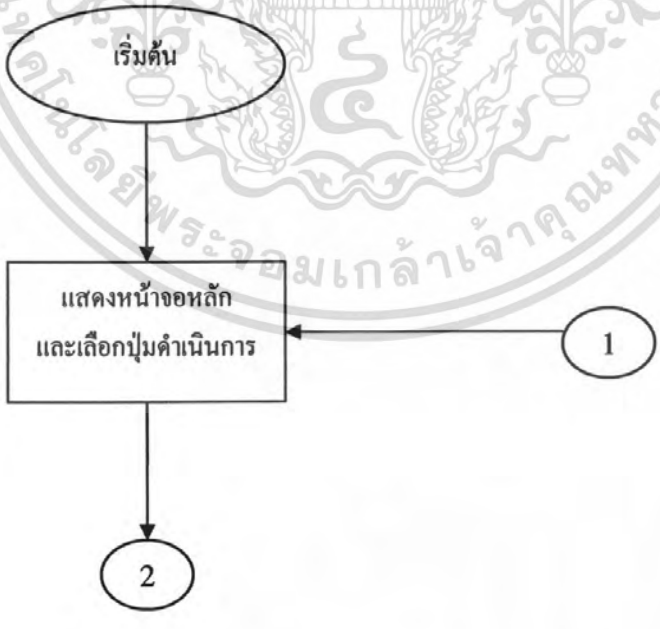
3.2 การออกแบบและพัฒนาระบบบัตรสมาชิกและเติมเงิน

ในส่วนการทำบัตรสมาชิกและการเติมเงินจะอยู่แยกจากส่วนของทางเข้า-ออก ในการทำบัตรใหม่ การเติมเงิน และการยกเลิกบัตร จะดำเนินการผ่านเจ้าหน้าที่ประจำจุด โดยเจ้าหน้าที่จะทำการทำการลงข้อมูลให้และชำระเงินที่เจ้าหน้าที่ ข้อมูลจะถูกบันทึกลงในฐานข้อมูล และยอดเงินจะถูกบันทึกลงในบัตร เพื่อที่จะนำไปใช้กับระบบจอดรถบันทึกเวลาเข้า-ออก และเก็บค่าบริการได้ต่อไป

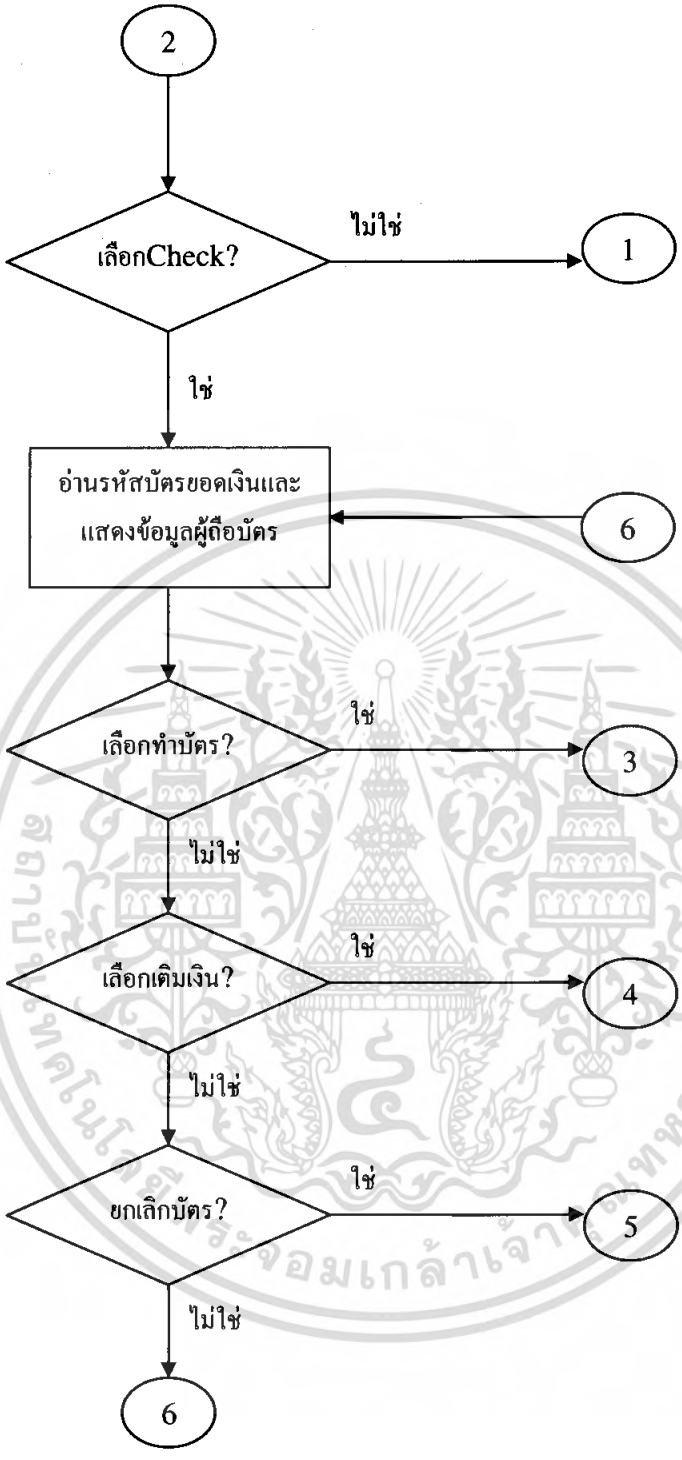


รูปที่ 3.11 หน้าจอของโปรแกรมระบบบัตรสมาชิกและเติมเงิน

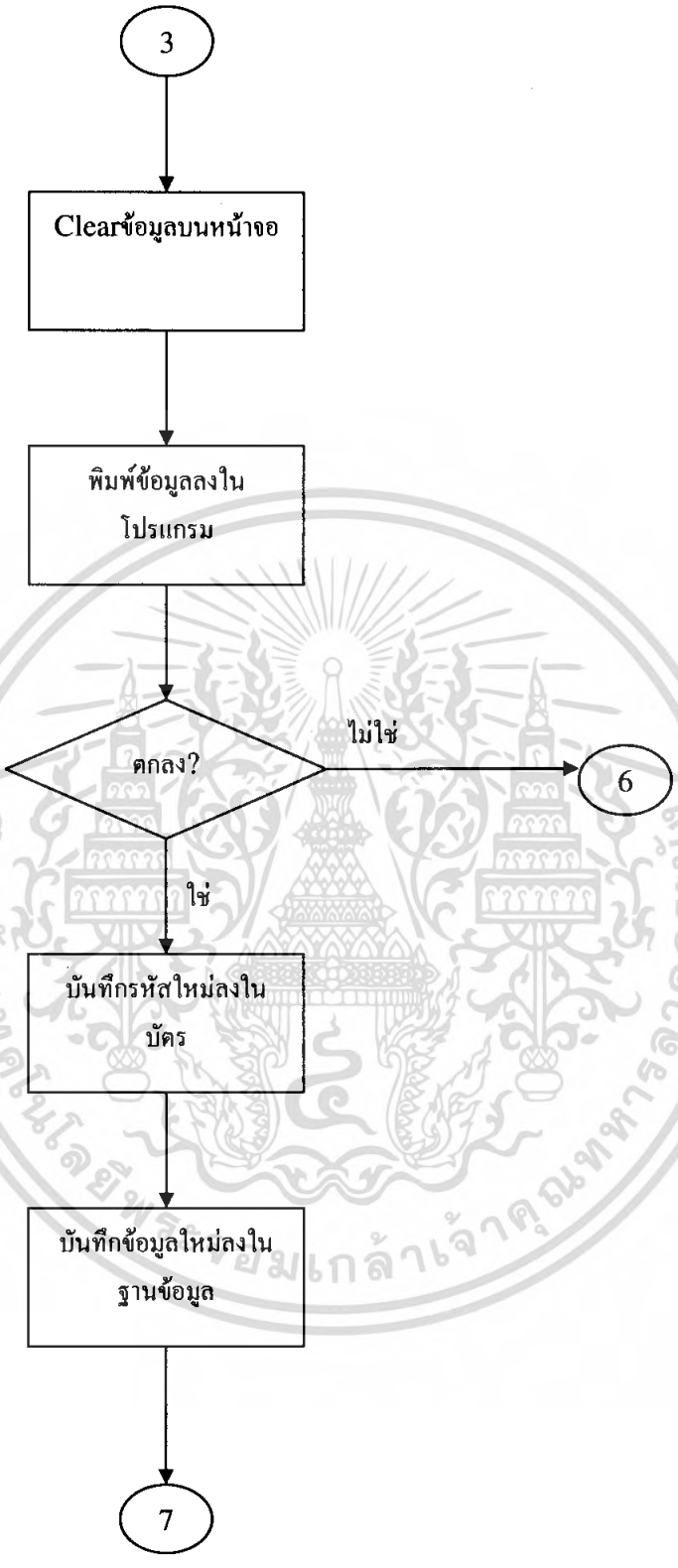
3.2.1 โปรแกรมระบบบัตรสมาชิกและเติมเงิน



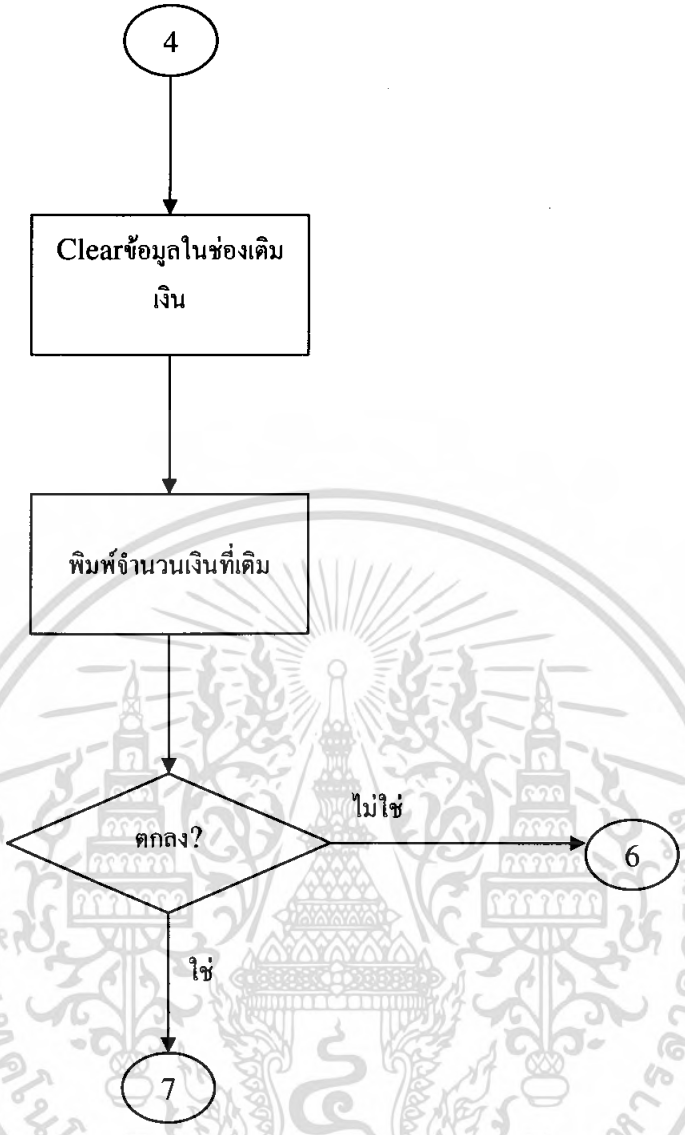
รูปที่ 3.12 โฟลวชาร์ทการทำงานของระบบบัตรสมาชิกและเติมเงิน



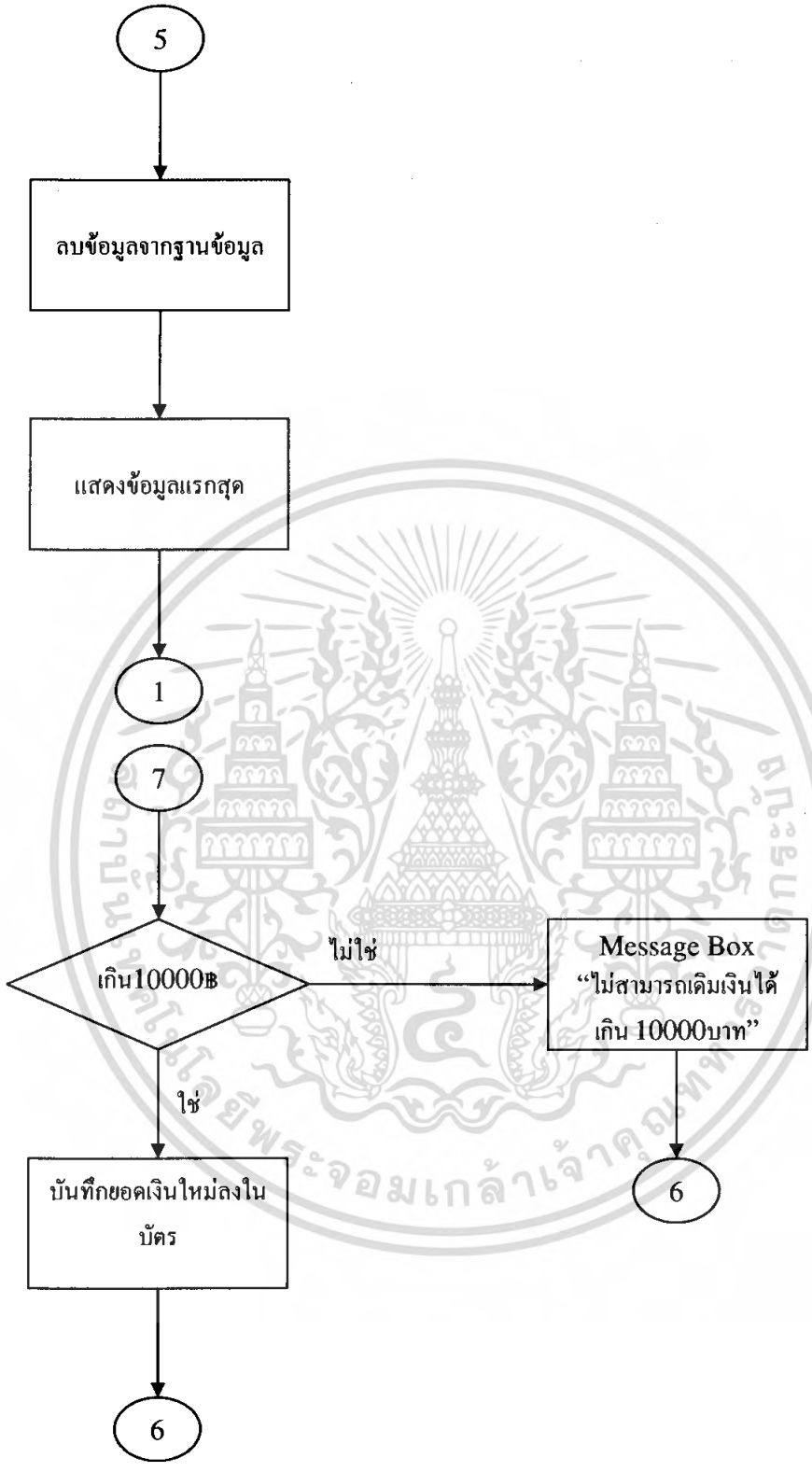
รูปที่ 3.13 โฟลวชาร์ทการทำงานของระบบบัตรสมาชิกและเติมเงิน (ต่อ)



รูปที่ 3.14 โฟลวชาร์ทการทำงานของระบบบัตรสมาชิกและเติมเงิน (ต่อ)



รูปที่ 3.15 โฟลวชาร์ทการทำงานของระบบบัตรสมาชิกและเติมเงิน (ต่อ)



รูปที่ 3.16 โฟลวชาร์ทการทำงานของระบบบัตรสมาชิกและเติมเงิน (ต่อ)

บทที่ 4

การทดลองและผลการทดลอง

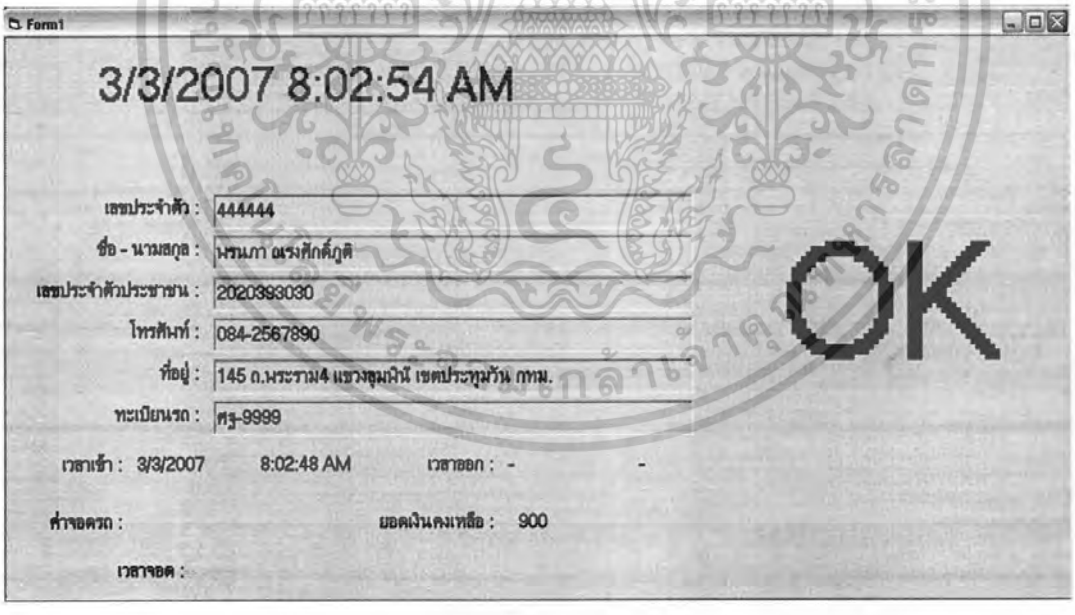
ในบทนี้จะกล่าวถึงระบบจอดรถที่ใช้บัตรสมาร์ตการ์ดโดยรวม และผลที่ได้จากการทดสอบระบบ โดยผลที่ได้จะแสดงให้เห็นวิธีการใช้งานและผลที่ได้จากการทดสอบ โดยแบ่งออกเป็น 2 ส่วน ดังนี้

4.1 ระบบจอดรถบันทึกเวลาเข้า-ออก และเก็บค่าบริการ

ระบบนี้จะต้องเสียบบัตรเข้ากับเครื่องอ่าน-เขียนบัตรทั้งหมด 2 ครั้ง แบ่งตามส่วนได้ดังนี้

4.1.1 ส่วนบันทึกและแสดงเวลาเข้าจอด

เมื่อผู้ใช้งานจะนำรถเข้าจะต้องนำบัตรสมาร์ตการ์ดเสียบเข้าอ่านเครื่องอ่านบัตร เครื่องอ่านบัตรจะอ่านรหัสบัตรและยอดเงิน แล้วโปรแกรมจะแสดงข้อมูลของผู้ใช้บริการ ยอดเงินในบัตร และแสดงเวลาเข้าพร้อมบันทึกลงในฐานข้อมูล ดังรูปที่ 4.1 และตารางที่ 4.1



รูปที่ 4.1 หน้าจอของโปรแกรมเมื่อนำรถเข้าจอด

ตารางที่ 4.1 การบันทึกเวลาที่เข้าจอดลงในฐานข้อมูล

	name	customerid	personal id	address	phone	password	date in	time in	date out	time out
▶		000000	0-0000-00000-00-0	-	-	-	-	-	-	-
	สมคิด จิตตรง	460001	1-2345-67890-12-3	1313 หอพักกมล ซ.เกริงงาม2 เขตลาดกระบัง กทม.	02-6333333	ffffff	-	-	-	-
	นันทวัฒน์ จิระวัฒนพงศ์	460123	1-3099-90001-29-0	210-214 ถ.สุรนา ต.ในเมือง อ.เมือง จ.นครราชสีมา	087-2477867	ffffff	-	-	-	-
	กำพล อินทรกำแหง	444445	1-3013-01301-30-0	11/3 ม.7 ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ	0872434572	ffffff	-	-	-	-
	อาคม กันเอง	460001	1-1111-11111-11-1	321 ถ.กิ่งแก้ว เขตลาดกระบัง กทม.	-	ffffff	-	-	-	-
	พรนภา ณรงศักดิ์ภูติ	444444	1-3004-42517-10-3	145 ถ.พระราม4 แขวงสุมพินี เขตประทุมวัน กทม.	084-2567890	ffffff	3/3/2007	8:02:48 AM	-	-
	นายก หาดใหญ่	456654	1-2222-33333-44-5	455 ถ.พัฒนาการ เขตประเวศ กทม.	044-222222	ffffff	-	-	-	-
*										

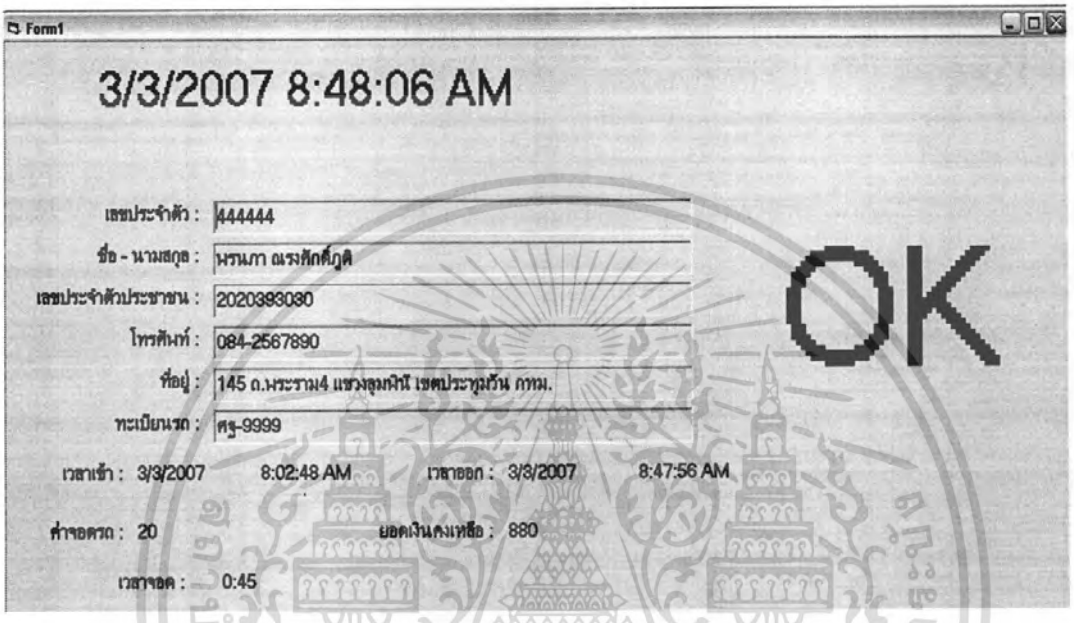
โดยแต่ละหลักในตารางจะเก็บข้อมูลดังนี้

- name ชื่อ-นามสกุลของสมาชิก
- customerid เลขประจำตัวของสมาชิก
- personal id เลขประจำตัวประชาชนของสมาชิก
- address ที่อยู่ของสมาชิก
- phone เบอร์โทรศัพย์ของสมาชิก
- password รหัสผ่านของบัตร ใช้สำหรับเขียนบัตรหรือเปลี่ยนแปลงรหัส
- date in วันที่สมาชิกนำรถเข้าจอด
- time in เวลาที่สมาชิกนำรถเข้าจอด
- date out วันที่สมาชิกนำรถออก
- time out เวลาที่สมาชิกนำรถเข้าออก

จากตารางที่ 4.1 จะพบว่า ในหลัก date in และหลัก time in ของแถว พรนภา ณรงศักดิ์ภูติ จะมีการเก็บข้อมูล โดยที่หลัก date in จะทำการเก็บข้อมูลรถเข้า คือ 3/3/2007 และหลัก time in จะทำการเก็บข้อมูลเวลาเข้าคือ 8:02:48 AM โดยจะสอดคล้องกับรูปที่ 4.1

4.1.2 ส่วนบันทึก แสดงเวลาออกและเก็บค่าบริการ

เมื่อผู้ใช้งานจะนำรถออก จะต้องนำบัตรสมาร์ตการ์ดเสียบเข้าอ่านเครื่องอ่านบัตร เครื่องอ่านบัตรจะอ่านรหัสบัตรและยอดเงิน แล้วโปรแกรมจะแสดงข้อมูลของผู้ใช้บริการ แสดงเวลาเข้า-ออก พร้อมบันทึกลงในฐานข้อมูล และคำนวณค่าบริการจอดรถแล้วหักค่าบริการออกจากบัตร



รูปที่ 4.2 หน้าจอของโปรแกรมเมื่อนำรถออก

ตารางที่ 4.2 การบันทึกเวลาที่ออกลงในฐานข้อมูล

	name	customerid	personal id	address	phone	password	date in	time in	date out	time out
▶		000000	0-0000-00000-00-0	-	-	-	-	-	-	-
	สมคิด จิตตรง	460001	1-2345-67890-12-3	1313 หอพักกมล ซ.เกกิงหม2 เขตลาดกระบัง กทม.	02-6333333	ffffff	-	-	-	-
	นันทวัฒน์ จิระวัฒนพงศ์	460123	1-3099-90001-29-0	210-214 ถ.สุรนา ต.ในเมือง อ.เมือง จ.นครราชสีมา	087-2477867	ffffff	-	-	-	-
	กำพล อินทรกำแหง	444445	1-3013-01301-30-0	11/3 ม.7 ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ	0872434572	ffffff	-	-	-	-
	อาคม กันเอง	460001	1-1111-11111-11-1	321 ถ.กิ่งแก้ว เขตลาดกระบัง กทม.	-	ffffff	-	-	-	-
	พรนภา ณรงศักดิ์ภูติ	444444	1-3004-42517-10-3	145 ถ.พระราม4 แขวงสุขุมวิท เขตประทุมวัน กทม.	084-2567890	ffffff	3/3/2007	8:02:48 AM	3/3/2007	8:47:56 AM
	นายก หาดใหญ่	456654	1-2222-33333-44-5	455 ถ.พัฒนาการ เขตประเวศ กทม.	044-222222	ffffff	-	-	-	-
*										

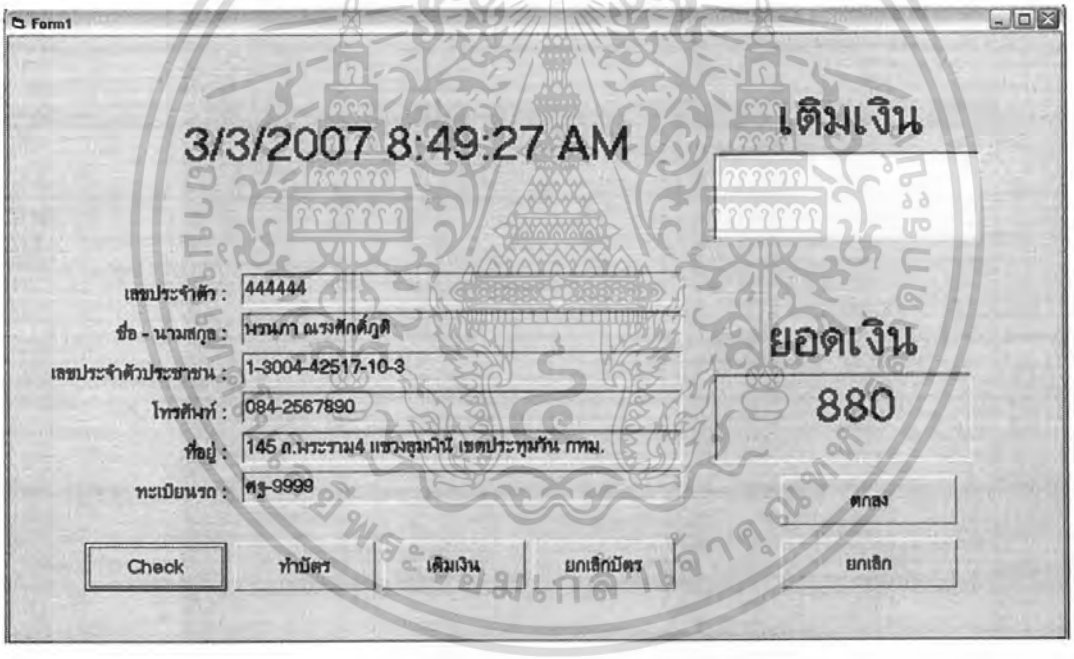
จากตารางที่ 4.2 จะพบว่า ในหลัก date out และหลัก time out ของแถว พรนภา ณรงศักดิ์ภูติ จะมีการเก็บข้อมูล โดยที่หลัก date out จะทำการเก็บข้อมูลรถออก คือ 3/3/2007 และหลัก time out จะทำการเก็บข้อมูลเวลาออกคือ 8:47:56 AM โดยจะสอดคล้องกับรูปที่ 4.2

4.2 ระบบบัตรสมาชิกและเติมเงิน

เป็นส่วนที่ใช้ทำบัตรสมาชิกและเติมเงินโดยแยกจากจุดเข้า-ออกของระบบจอดรถ แบ่งเป็น4 ส่วน คือ

4.2.1 ส่วน Check บัตร

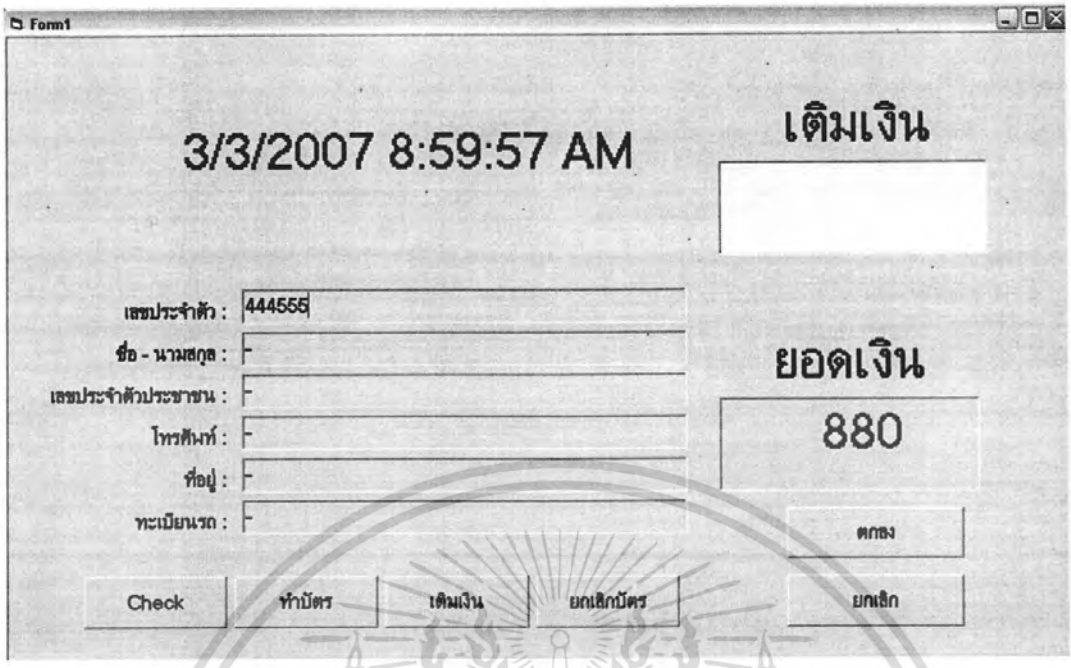
เมื่อเสียบบัตรแล้วเลือกปุ่ม "Check" โปรแกรมจะค้นหาข้อมูลที่ได้จากบัตรในฐานข้อมูล และแสดงข้อมูลพร้อมยอดเงินคงเหลือ



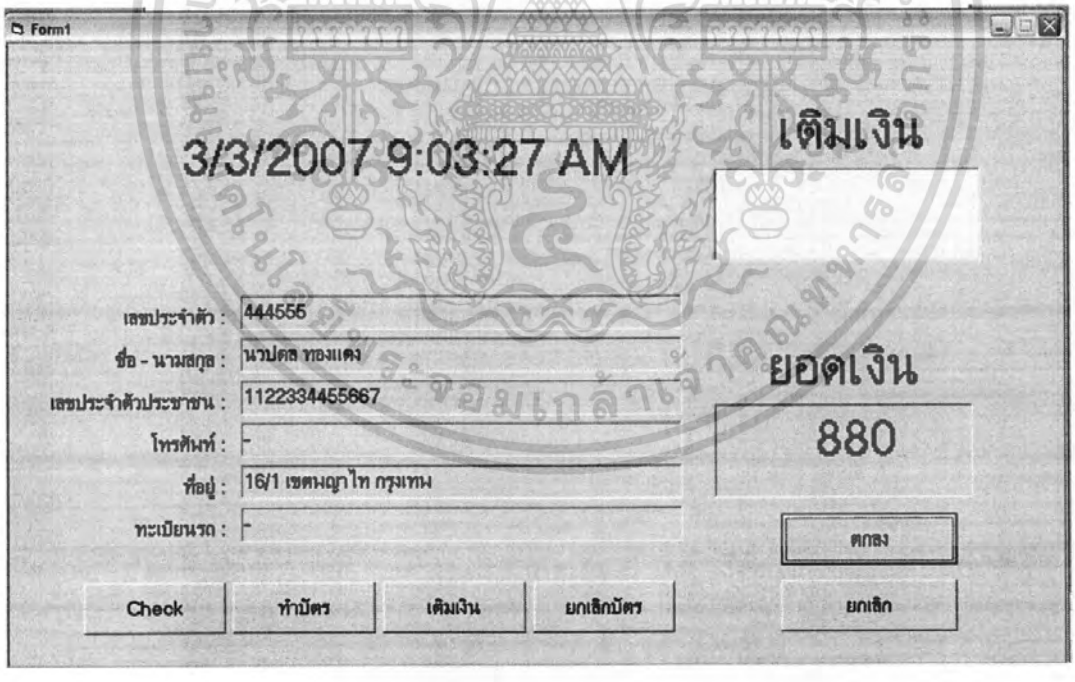
รูปที่ 4.3 หน้าจอของโปรแกรมเมื่อทำการCheck

4.2.2 ส่วนทำบัตร

เมื่อเลือกปุ่ม "ทำบัตร" ข้อมูลทั้งหมดบนหน้าจอจะถูกลบ สำหรับเพิ่มข้อมูลใหม่ลงไปแทน ได้แก่ รหัสประจำตัว ข้อมูลส่วนบุคคล และยอดเงินที่จะเติม จากนั้นเลือกปุ่ม "ตกลง" เพื่อเพิ่มข้อมูลใหม่ลงในฐานข้อมูล และเขียนรหัสกับยอดเงินลงในบัตร หรือเลือกปุ่ม "ยกเลิก" เพื่อยกเลิกการทำบัตร



รูปที่ 4.4 หน้าจอของโปรแกรมก่อนจะทำบัตรใหม่



รูปที่ 4.5 หน้าจอของโปรแกรมเมื่อตกลงทำบัตรใหม่

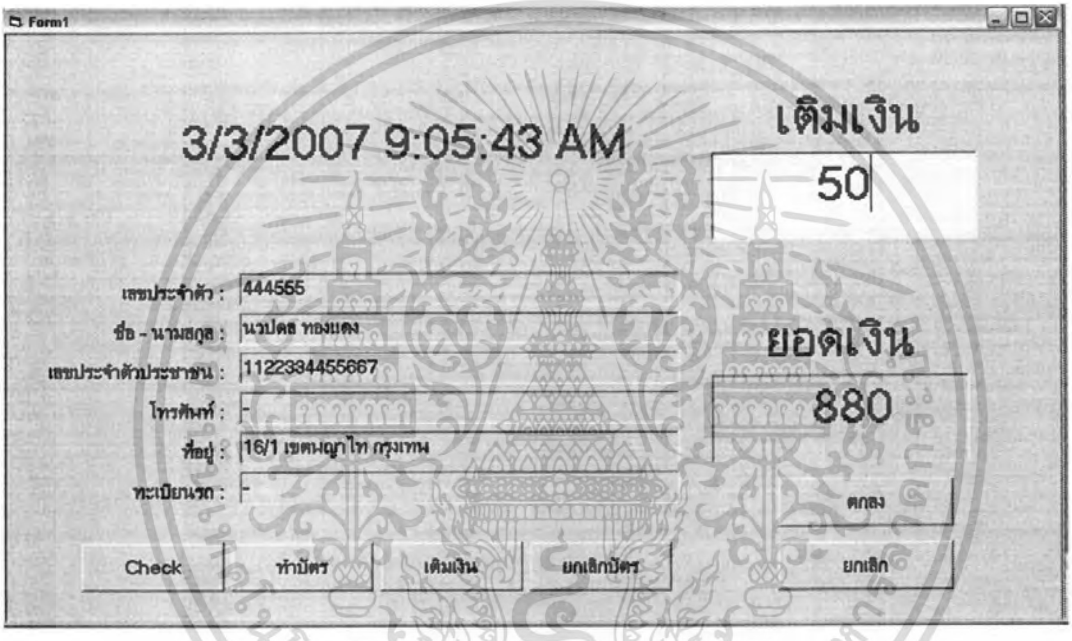
ตารางที่ 4.3 การบันทึกข้อมูลสมาชิกใหม่ลงในฐานข้อมูล

	name	customerid	personal id	address	phone	password	date in	time in	date out	time out
▶		000000	0-0000-00000-00-0	-	-	-	-	-	-	-
	นวปัดล ทองแดง	444555	1-1223-34455-66-7	16/1 เขตพญาไท กรุงเทพ	-					
	สงคิต จิตตรง	460001	1-2345-67890-12-3	1313 หอพักกมล ซ.เกกิงงาม2 เขตลาดกระบัง กทม.	02-6333333	ffffff	-	-	-	-
	นันทวัฒน์ จิระวัฒนพงศ์	460123	1-3099-90001-29-0	210-214 ถ.สุรนา ต.ในเมือง อ.เมือง จ.นครราชสีมา	087-2477867	ffffff	-	-	-	-
	กำพล อินทรกำแหง	444445	1-3013-01301-30-0	11/3 ม.7 ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ	0872434572	ffffff	-	-	-	-
	อาคม กันเอง	460001	1-1111-11111-11-1	321 ถ.กิ่งแก้ว เขตลาดกระบัง กทม.	-	ffffff	-	-	-	-
	พรนภา ณรงศักดิ์ภูติ	444444	1-3004-42517-10-3	145 ถ.พระราม4 แขวงลุมพินี เขตประทุมวัน กทม.	084-2567890	ffffff	3/3/2007	8:02:48 AM	3/3/2007	8:47:56 AM
	นายก หาดใหญ่	456654	1-2222-33333-44-5	455 ถ.พัฒนาการ เขตประเวศ กทม.	044-222222	ffffff	-	-	-	-
*										

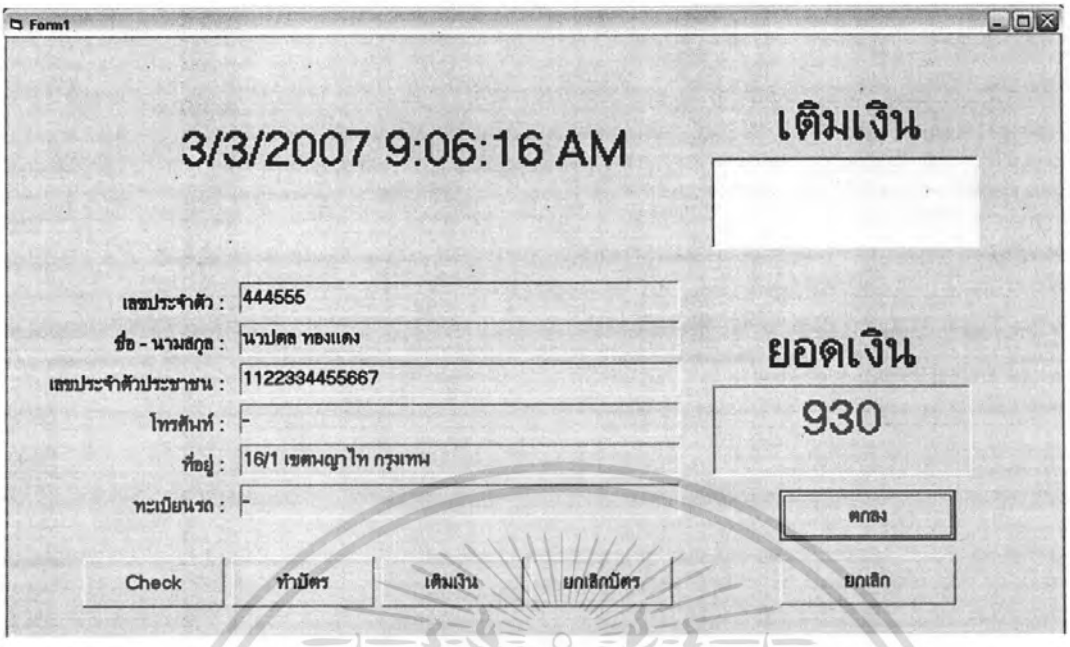
จากตารางที่ 4.3 จะพบว่า ในตารางมีแถวเพิ่มขึ้น คือแถว นวปดล ทองแดง โดยแต่ละหลักในแถวนี้จะมีการเก็บข้อมูลเหมือนข้อมูลที่ใส่ลงดังรูปที่ 4.5 ยกเว้นหลัก password , date in , time in , date out และ time out ที่ไม่มีการใส่ข้อมูลไว้

4.2.3 ส่วนเติมเงิน

เมื่อเลือกปุ่ม "เติมเงิน" จะต้องทำการใส่จำนวนเงินที่จะเติมลงในช่องเติมเงิน แล้วเลือกปุ่ม "ตกลง" เงินที่เติมจะนำไปบวกกับยอดเงินเก่าแล้วบันทึกยอดเงินใหม่ลงในบัตรดังรูปที่ 4.6 และรูปที่4.7

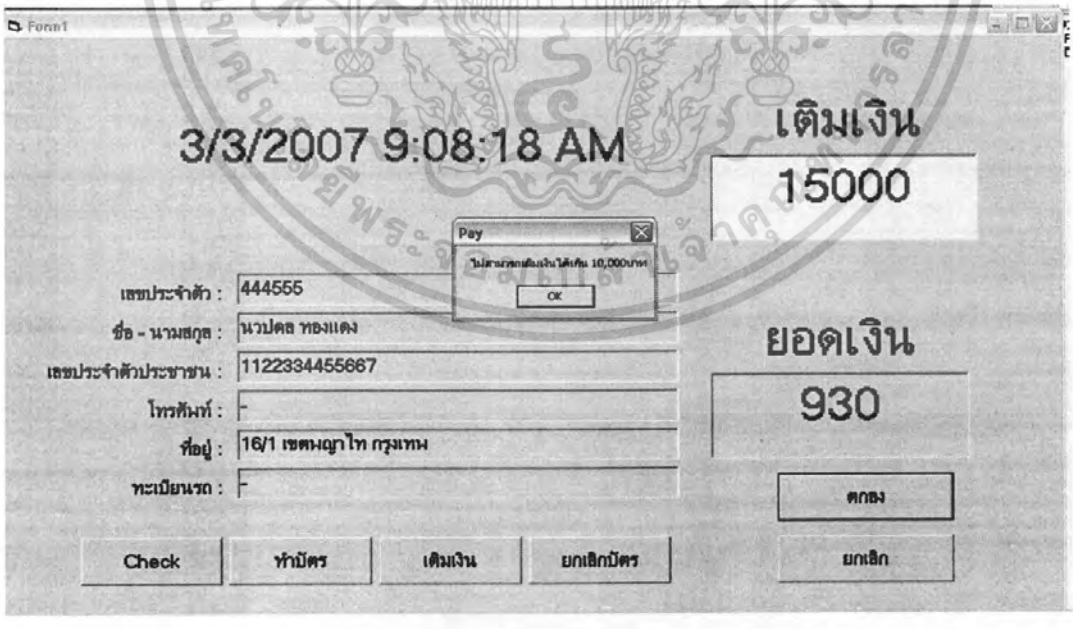


รูปที่ 4.6 หน้าจอของโปรแกรมเมื่อทำการเติมเงิน



รูปที่ 4.7 หน้าจอของโปรแกรมเมื่อทำการเติมเงินแล้ว

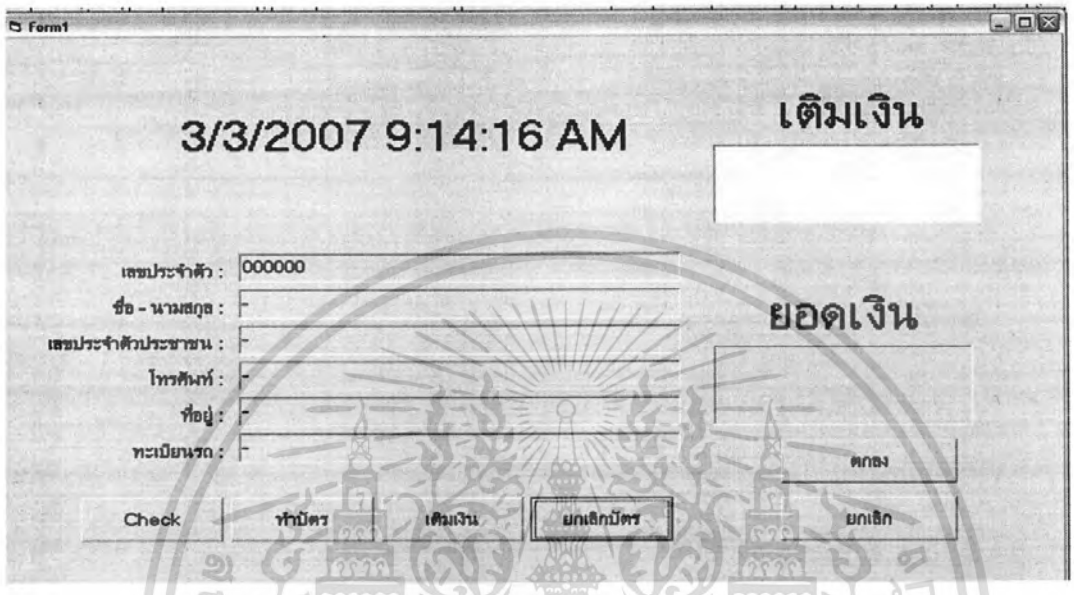
หากมีการเติมเงินแล้วบัตรมียอดเงินเกิน 10000บาท จะไม่ได้รับอนุญาต และการเติมเงินครั้งนั้นจะถูกยกเลิก ดังรูปที่ 4.8



รูปที่ 4.8 หน้าจอของโปรแกรมเมื่อทำการเติมเงินเกิน

4.2.4 ส่วนยกเลิกบัตร

เมื่อเลือกปุ่ม "ยกเลิกบัตร" ข้อมูลของรหัสบัตรในฐานข้อมูลนั้นจะถูกลบทั้งหมด ดังรูปที่ 4.9 , ตารางที่ 4.4 และตารางที่ 4.5



รูปที่ 4.9 หน้าจอของโปรแกรมเมื่อทำการยกเลิกบัตร

ตารางที่ 4.4 การลบข้อมูลที่ยกเลิกออกจากฐานข้อมูล

	name	customerid	personal id	address	phone	password	date in	time in	date out	time out
▶		000000	0-0000-00000-00-0	-	-	-	-	-	-	-
	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted	#Deleted
	สมคิด จิตตรง	460001	1-2345-67890-12-3	1313 หอพักกมล ซ.เกกิ่งงาม2 เขตลาดกระบัง กทม.	02-6333333	ffffff	-	-	-	-
	นันทวัฒน์ จิระวัฒนพงศ์	460123	1-3099-90001-29-0	210-214 ถ.สุรนา ต.ในเมือง อ.เมือง จ.นครราชสีมา	087-2477867	ffffff	-	-	-	-
	กำพล อินทรกำแหง	444445	1-3013-01301-30-0	11/3 ม.7 ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ	0872434572	ffffff	-	-	-	-
	อาคม กันเอง	460001	1-1111-11111-11-1	321 ถ.กิ่งแก้ว เขตลาดกระบัง กทม.	-	ffffff	-	-	-	-
	พรนภา ณรงศักดิ์ภูติ	444444	1-3004-42517-10-3	145 ถ.พระราม4 แขวงสุขุมวิท เขตประเวศ กทม.	084-2567890	ffffff	3/3/2007	8:02:48 AM	3/3/2007	8:47:56 AM
	นายก หาดใหญ่	456654	1-2222-33333-44-5	455 ถ.พัฒนาการ เขตประเวศ กทม.	044-222222	ffffff	-	-	-	-
*										

ตารางที่ 4.5 ฐานข้อมูลที่ลบข้อมูลที่ยกเลิกออกแล้ว

	name	customerid	personal id	address	phone	password	date in	time in	date out	time out
▶		000000	0-0000-00000-00-0	-	-	-	-	-	-	-
	สมคิด จิตตรง	460001	1-2345-67890-12-3	1313 หอพักกมล ซ.เกกิงาม2 เขตลาดกระบัง กทม.	02-6333333	ffffff	-	-	-	-
	นันทวัฒน์ จิระวัฒนพงศ์	460123	1-3099-90001-29-0	210-214 ถ.สุรนา ต.ในเมือง อ.เมือง จ.นครราชสีมา	087-2477867	ffffff	-	-	-	-
	กำพล อินทรกำแหง	444445	1-3013-01301-30-0	11/3 ม.7 ต.เทพารักษ์ อ.เมือง จ.สมุทรปราการ	0872434572	ffffff	-	-	-	-
	อาคม กันเอง	460001	1-1111-11111-11-1	321 ถ.กิ่งแก้ว เขตลาดกระบัง กทม.	-	ffffff	-	-	-	-
	พรนภา ณรงศักดิ์ภูติ	444444	1-3004-42517-10-3	145 ถ.พระราม4 แขวงสมพันี เขตประทุมวัน กทม.	084-2567890	ffffff	3/3/2007	8:02:48 AM	3/3/2007	8:47:56 AM
	นายก หาดใหญ่	456654	1-2222-33333-44-5	455 ถ.พัฒนาการ เขตประเวศ กทม.	044-222222	ffffff	-	-	-	-
*										

จากตารางที่ 4.4 จะพบว่า เมื่อทำการยกเลิกข้อมูล ในแถวที่ 2 หรือแถว นวปดล ทองแดง จะถูกแทนด้วยข้อมูล #Deleted ทั้งแถว และจากตรางที่ 4.5 จะพบว่า เมื่อยกเลิกข้อมูลเรียบร้อยแล้ว แถวของ นวปดล ทองแดง ที่ถูกแทนด้วยข้อมูล #Deleted ทั้งแถวจะหายไป จำนวนแถวในตารางจะลดลง 1 แถว

บทที่ 5
บทสรุปและวิจารณ์

5.1 สรุป

เนื่องจากรูปแบบของงานที่นำมาประยุกต์ใช้นั้น ต้องการความปลอดภัยของข้อมูลเป็นสำคัญ เนื่องจากต้องการเก็บจำนวนเงินไว้ในตัวบัตรหรือรหัส เป็นต้น ดังนั้นจึงมีความสำคัญเป็นอย่างยิ่งที่จะต้องเลือกฟช้บัตรที่มีความปลอดภัยในการเก็บรักษาข้อมูลสูง และสามารถทำการปลอมแปลงได้ยาก และเนื่องจากเครื่องอ่าน-เขียนบัตรกับบัตรสมาร์ตการ์ดมีต้นทุนต่ำ จึงเหมาะที่จะนำสมาร์ตการ์ดมาประยุกต์ใช้มากที่สุด โดยสมาร์ตการ์ดที่เลือกใช้ในโครงงานนี้เป็นแบบไมโครโปรเซดเซอร์การ์ดชนิดมีหน้าสัมผัส (Contact Microprocessor Cards) ซึ่งจะมีหน่วยประมลผลในตัวสมาร์ตการ์ด ซึ่งจะสามารถทำให้ใช้หน่วยประมวลผลดังกล่าวนี้ในการประมวลผลการเข้าและถอดรหัสข้อมูลต่างๆ ที่ติดต่อระหว่างตัวบัตรสมาร์ตการ์ด และคอมพิวเตอร์ได้ซึ่งจะทำให้มีความปลอดภัยของข้อมูลสูงมาก

จากโครงงานที่ได้จัดทำขึ้นเป็นเรื่องของการนำสมาร์ตการ์ดมาประยุกต์ใช้งานเข้ากับระบบจอดรถและระบบสมาชิก-เติมเงิน โดยในด้านระบบจอดรถสามารถทำงานได้แต่ถ้าจะนำไปพัฒนาใช้ในเชิงพานิชย์ควรนำบัตรสมาร์ตการ์ดแบบไม่สัมผัส (Contactless) ซึ่งสามารถใช้งานได้สะดวกกว่าแม้จะมีราคาสูงกว่า ส่วนในด้านระบบสมาชิก-เติมเงินสามารถทำงานได้ตามความคาดหมาย

5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่พบในส่วนโปรแกรมระบบจอดรถ หรือโปรแกรมระบบสมาชิกและเติมเงิน ที่พัฒนาขึ้นด้วยภาษา Visual Basic 6.0 มาใช้งานร่วมกันเครื่องอ่าน-เขียนบัตรสมาร์ตการ์ด เกิดเนื่องจากช่วงเวลาที่ใช้ในการทำงานที่ไม่สัมพันธ์กันระหว่างโปรแกรม และเครื่อง-อ่านเขียนบัตรสมาร์ตการ์ดจนเกิดการผิดพลาดในการรับส่งข้อมูลระหว่างกันขึ้นทั้งนี้เนื่องจากว่าเครื่องอ่าน-เขียนบัตรต้องใช้เวลาช่วงหนึ่งในการติดต่อรับ-ส่งข้อมูลกับบัตรสมาร์ตการ์ด ดังนั้นเพื่อแก้ไขปัญหานี้จึงทำการเพิ่มโปรแกรมหน่วงเวลาเพื่อให้การติดต่อระหว่างเครื่องอ่าน -เขียนกับโปรแกรมเรียบร้อยก่อนนำค่าที่ได้ไปใช้ต่อไป

5.3 ข้อเสนอแนะแนวทางในการพัฒนาต่อไป

สำหรับแนวทางที่คาดว่าจะสามารถนำไปพัฒนาต่อนั้น น่าจะเปลี่ยนไปใช้บัตรสมาร์ตการ์ดแบบไม่สัมผัส (Contactless) ซึ่งสามารถใช้งานได้สะดวกรวดเร็วกว่าแบบมีหน้าสัมผัส (Contact) เพื่อให้การนำรถเข้า-ออกเป็นไปได้เร็วไม่ทำให้การจราจรติดขัดที่ทางเข้าออก

ในส่วนของโปรแกรมระบบจอดรถ อาจเพิ่มจะช่วยหาที่จอดอัตโนมัติหรือใช้ประกอบกับระบบจอดรถอัตโนมัติที่ตัวอาคารจอดรถนำรถไปเก็บในช่องจอดได้เอง และอาจใช้ในการขยายกับธุรกิจหรือระบบอื่นๆ เช่น ระบบบัตรพนักงานบริษัทที่สามารถใช้บันทึกเวลาเข้า-ออกงาน รวมทั้งสวัสดิการต่างๆ ที่ทางบริษัททำสัญญากับแหล่งสวัสดิการนั้นด้วย

เอกสารอ้างอิง

[1] สัจจะ จรัสรุ่งรวีวร, **คู่มือการเขียนโปรแกรมและใช้งาน Visual Basic 6.0,** กรุงเทพมหานคร : อินโพรเฟส, 2544

[2] ธนพล ฉันจรัสวิชัย, **การออกแบบและการสร้างฐานข้อมูลด้วย Visual Basic 6.0,** กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2543

[3] เลิศ แซ่ตั้ง, **เทคโนโลยีสมาร์ตการ์ด,** กรุงเทพมหานคร : ซีเอ็ดยูเคชั่น, 2546

[4] ทิพย์วรรณ อมรชัยทรัพย์, **จับประเด็นร้อนๆ Microsoft Access 2002 ฉบับสมบูรณ์,** กรุงเทพมหานคร : บริษัท เฟิร์ส แปซิฟิก (ดอกหญ้า) จำกัด, 2545

[5] ศุภชัย สมพานิช, **Database Programming ด้วย Visual Basic ฉบับมืออาชีพ,** นนทบุรี : อินโฟเพรส, 2546

[6] อมรรัตน์ จิรัฐติกาลโชติ, โอพาร พิสิฐสวัสดิ์, **"สมาร์ทการ์ดของระบบสารสนเทศนักศึกษา",** วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546

[7] กฤษฎา บุญมีวิเศษ, สมเจตน์ สวนทอง, สัมฤทธิ์ ประทุมจิต, **"การประยุกต์สมาร์ตการ์ด",** วิทยานิพนธ์วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2546

ภาคผนวก

SIEMENS

ICs for Chip Cards

Intelligent 256-Byte EEPROM
SLE 4432/SLE 4442

Data Sheet 07.95

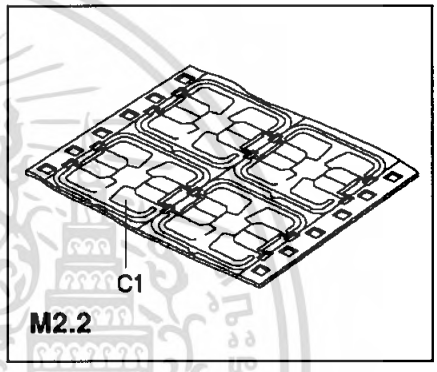
SIEMENS

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

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

Features

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



M2.2

Additional Feature of SLE 4442

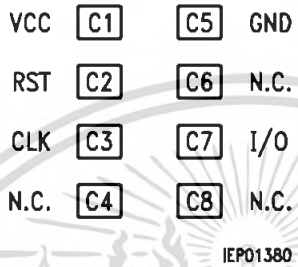
- Data can only be changed after entry of the correct 3-byte programmable security code (security memory)

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

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

1 Pin Configuration

(top view)



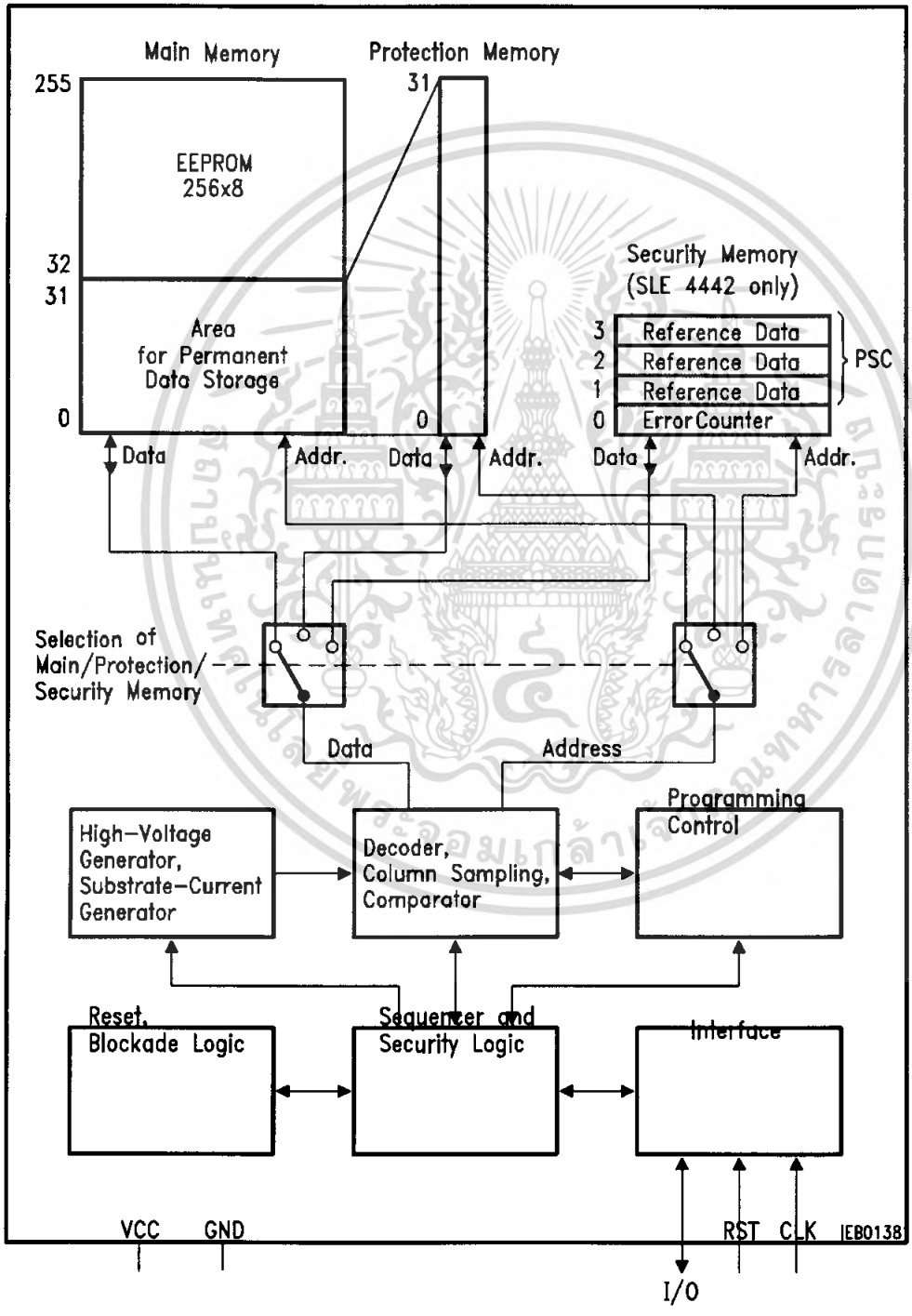
M2.2 (Card Contacts)

Pin Definitions and Functions

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

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

2 Functional Description



Block Diagram

2.1 Memory Overview

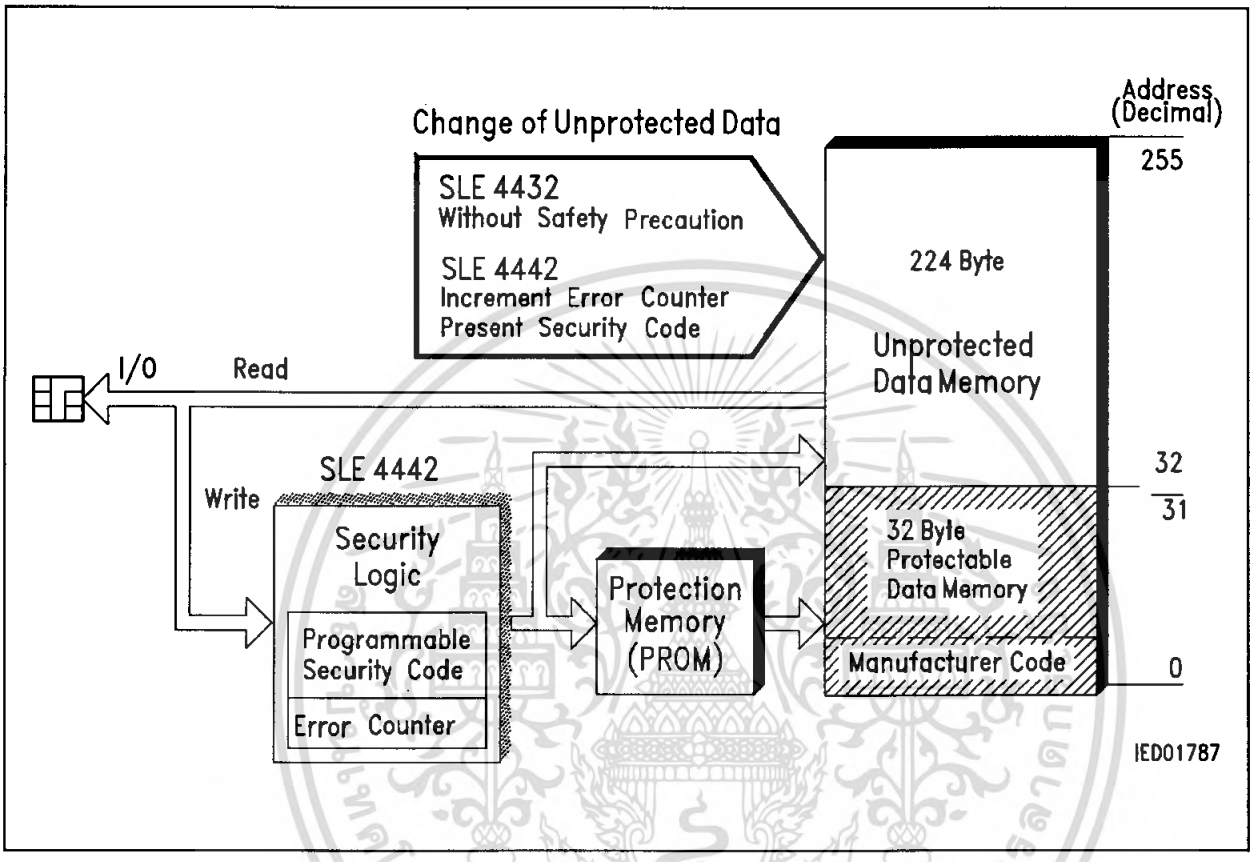


Figure 1
Memory Overwiew

SLE 4432

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

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

SLE 4442

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

2.2 Transmission Protocol

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

The transmission protocol consists of the 4 modes:

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

} Operational modes

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

2.2.1 Reset and Answer-to-Reset

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

Answer-to-Reset (Hex)

Byte 1	Byte 2	Byte 3	Byte 4
$DO_7 \ldots DO_0$	$DO_{15} \ldots DO_8$	$DO_{23} \ldots DO_{16}$	$DO_{31} \ldots DO_{24}$

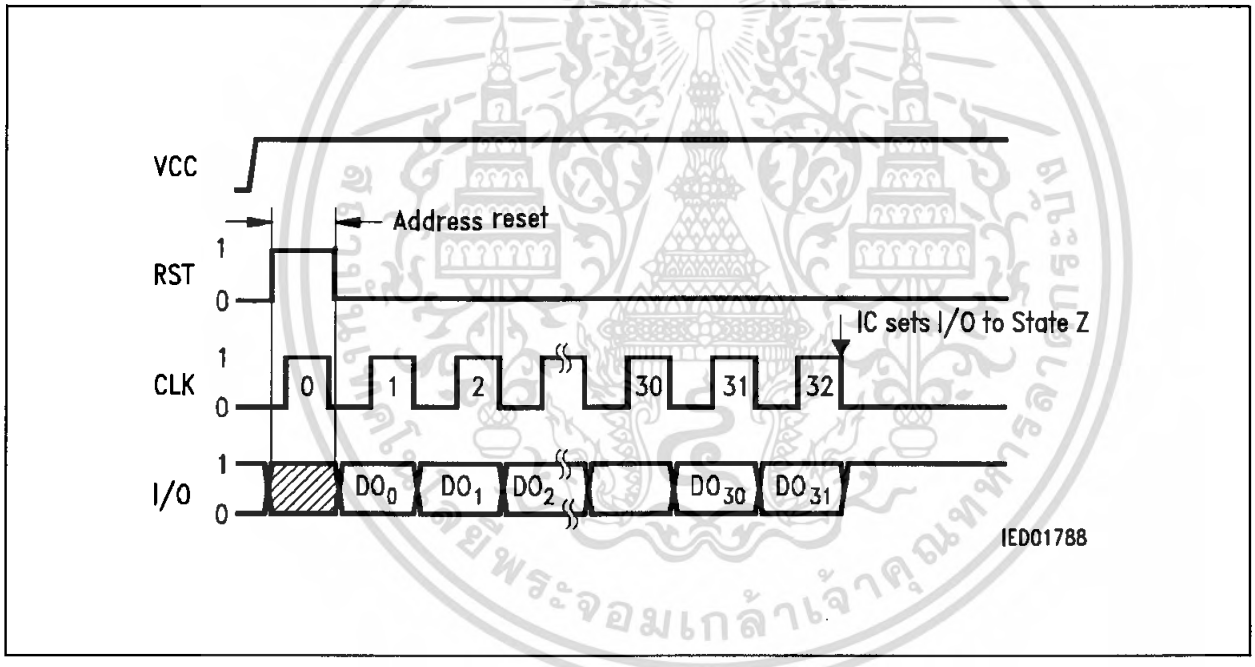


Figure 2
Reset and Answer-to-Reset

2.2.2 Operational Modes

Command Mode

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

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

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

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

Outgoing Data Mode

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

Processing Mode

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

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

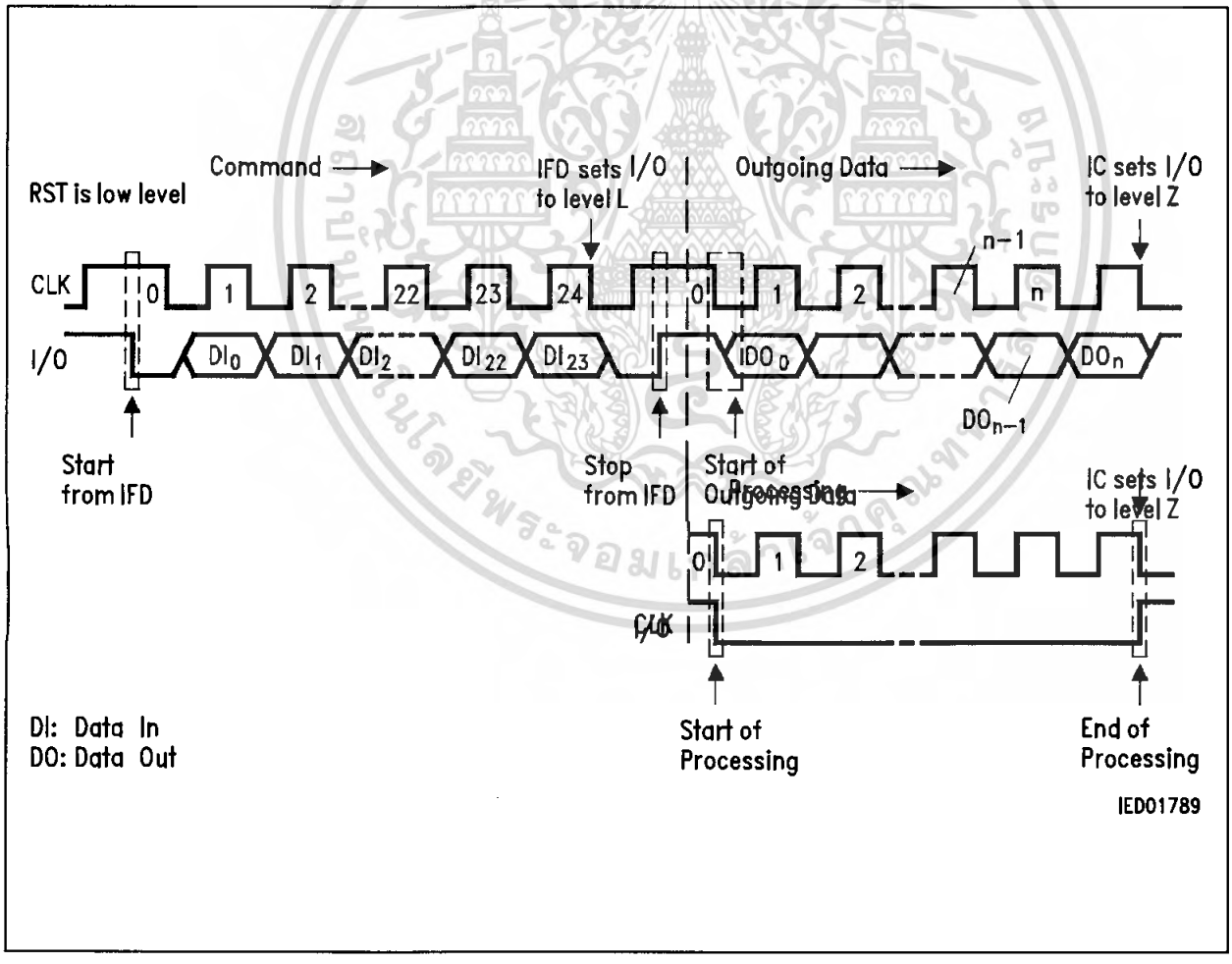


Figure 3
Operational Modes

2.3 Commands

Command Format

Each command consists of three bytes:

MSB			Control				LSB	MSB			Address				LSB	MSB			Data				LSB
&	&	&	&	&	&	&	&	%	%	%	%	%	%	%	%	(	(	(	(	(	(	(	(

Beginning with the control byte LSB is transmitted first.

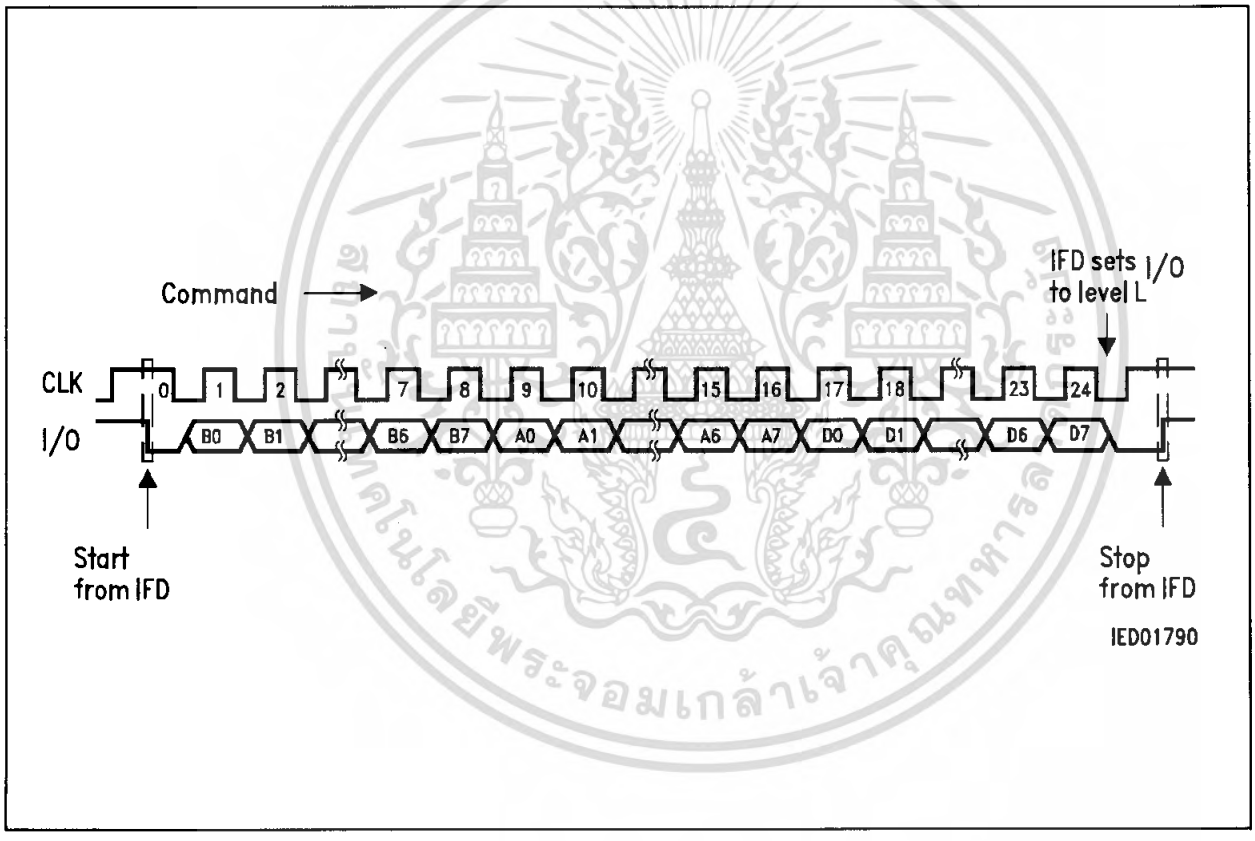


Figure 4
Command Mode

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

Table 1

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

Table 2
SLE 4442 only

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

2.3.1 Read Main Memory (SLE 4432 and SLE 4442)

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

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

Command: READ MAIN MEMORY

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

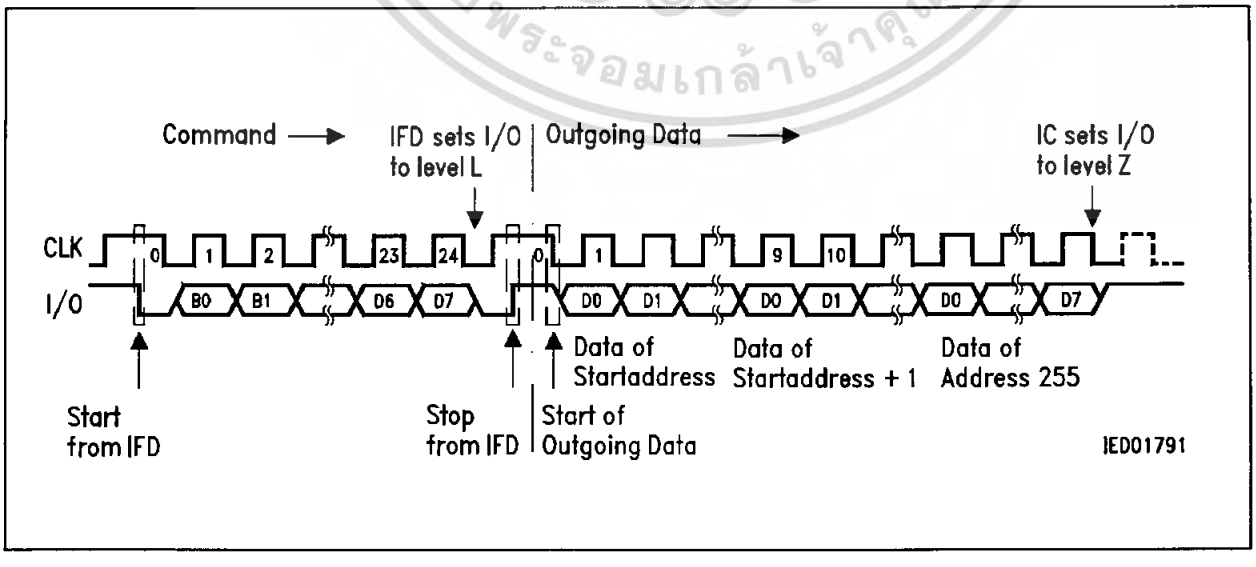


Figure 5
Read Main Memory

2.3.2 Read Protection Memory (SLE 4432 and SLE 4442)

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

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

Command: READ PROTECTION MEMORY

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

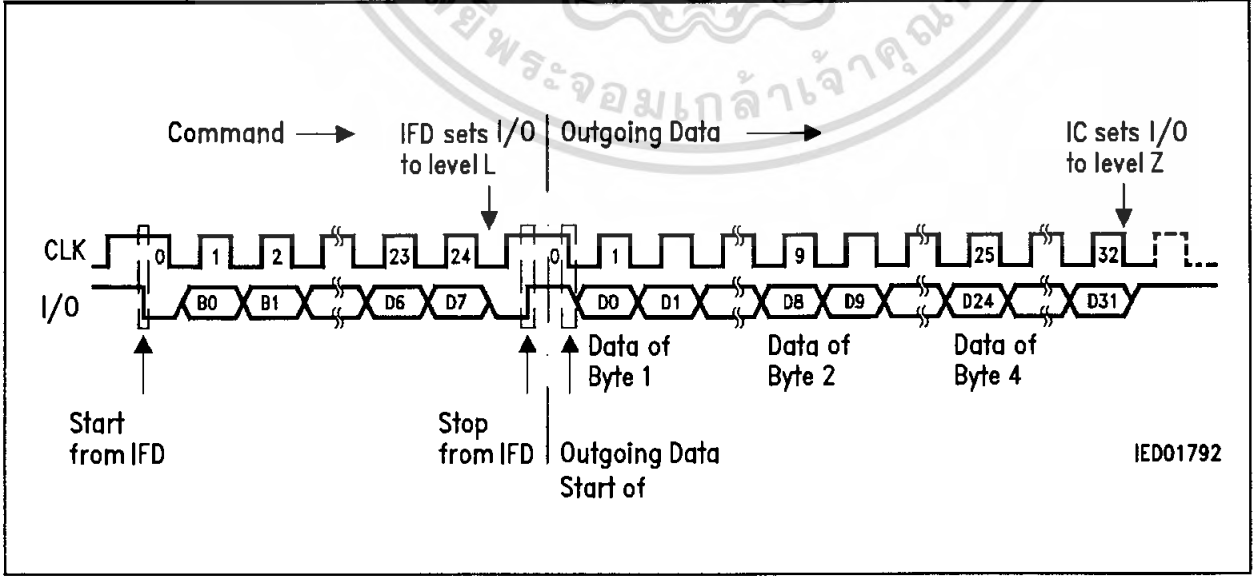


Figure 6
Read Protection Memory

2.3.3 Update Main Memory (SLE 4432 and SLE 4442)

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

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

(All values at 50 kHz clock rate.)

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

Command: UPDATE MAIN MEMORY

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

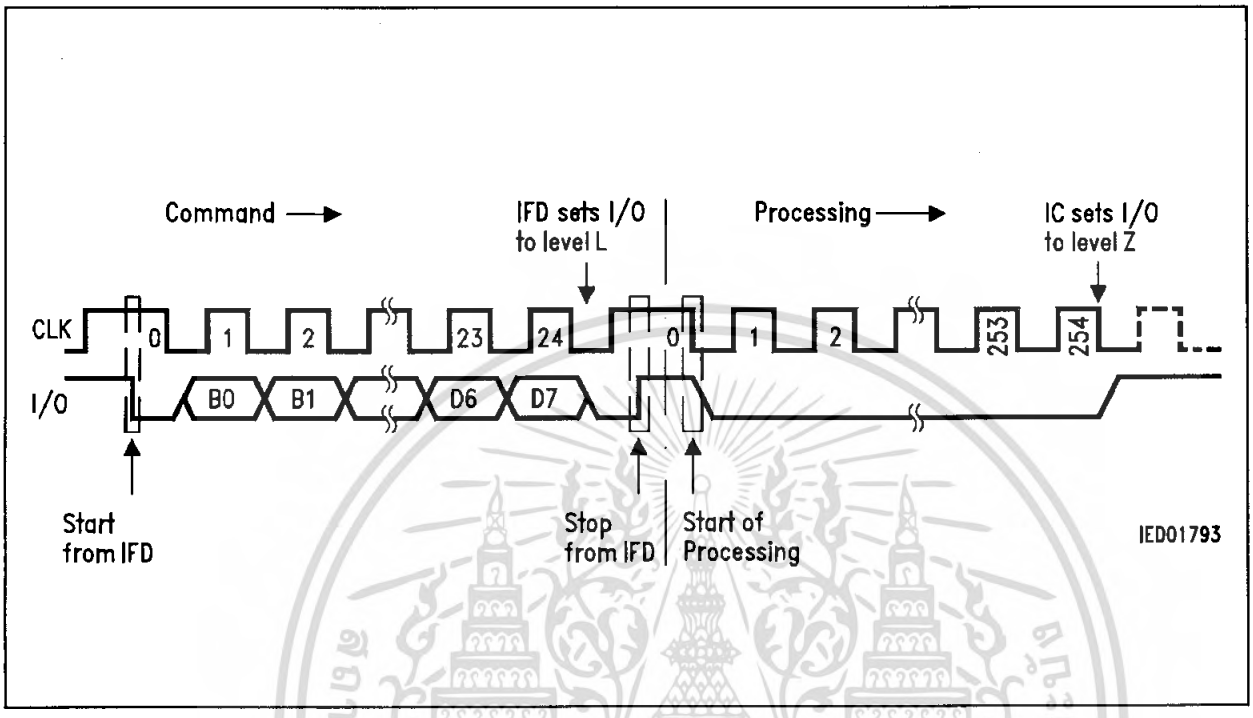


Figure 7
Erase and Write Main Memory

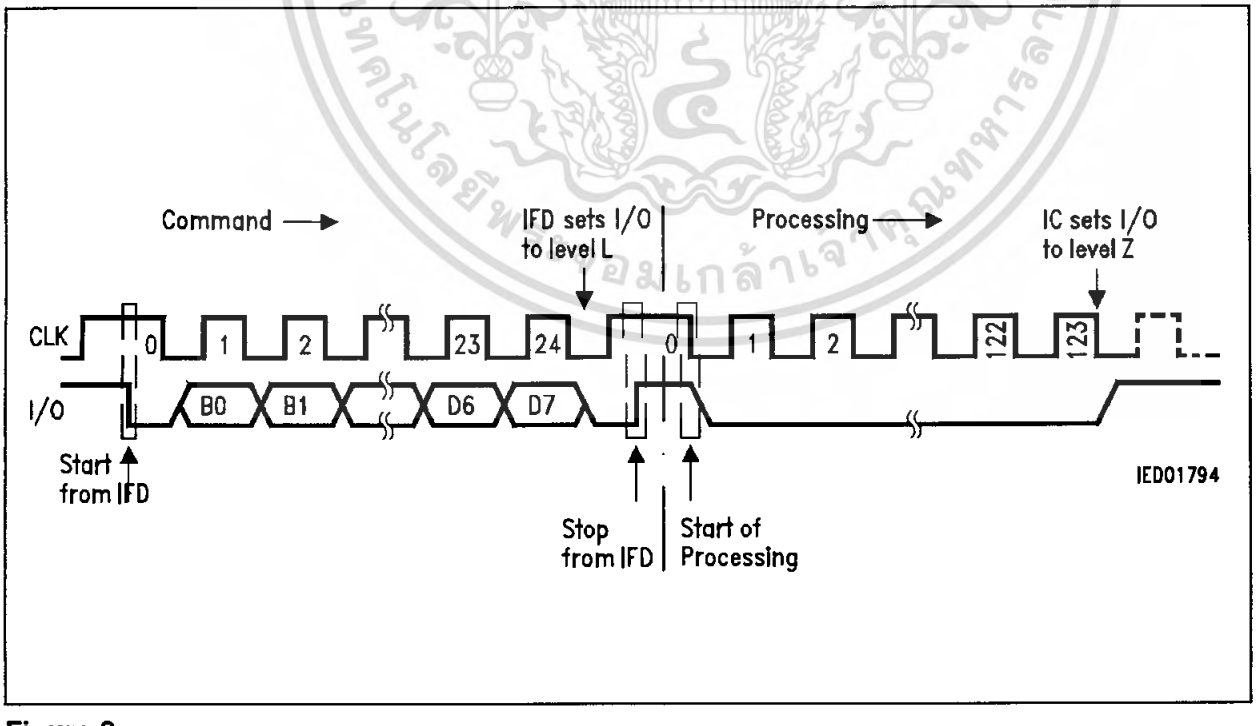


Figure 8
Erase or Write Main Memory

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

2.3.4 Write Protection Memory (SLE 4432 and SLE 4442)

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

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

Command: WRITE PROTECTION MEMORY

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

2.3.5 Read Security Memory (SLE 4442 only)

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

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

Command: READ SECURITY MEMORY

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

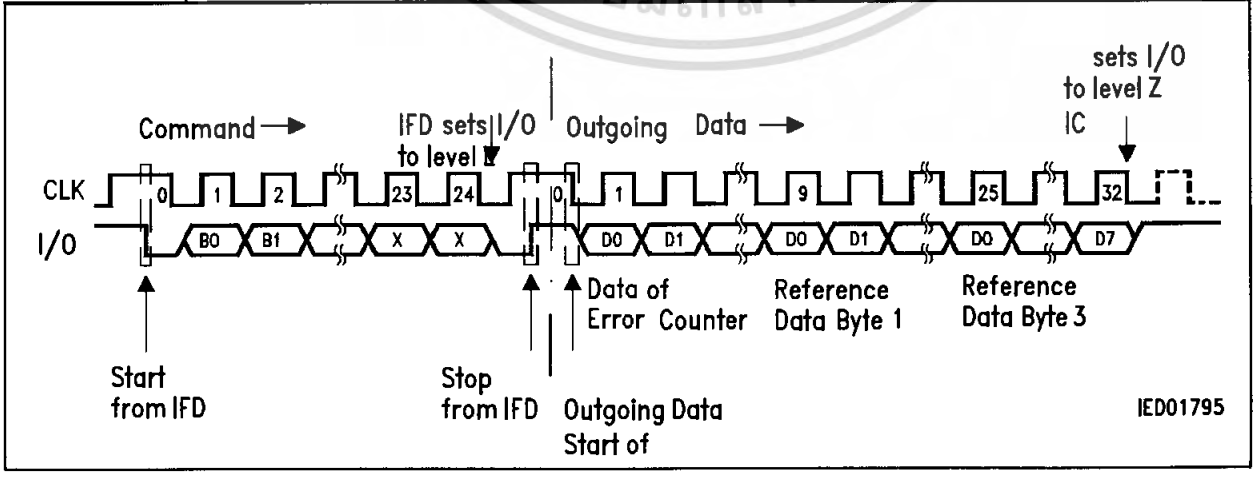


Figure 9
Read Security Memory

2.3.6 Update Security Memory (SLE 4442 only)

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

Command: UPDATE SECURITY MEMORY

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

2.3.7 Compare Verification Data (SLE 4442 only)

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

Command: COMPARE VERIFICATION DATA

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

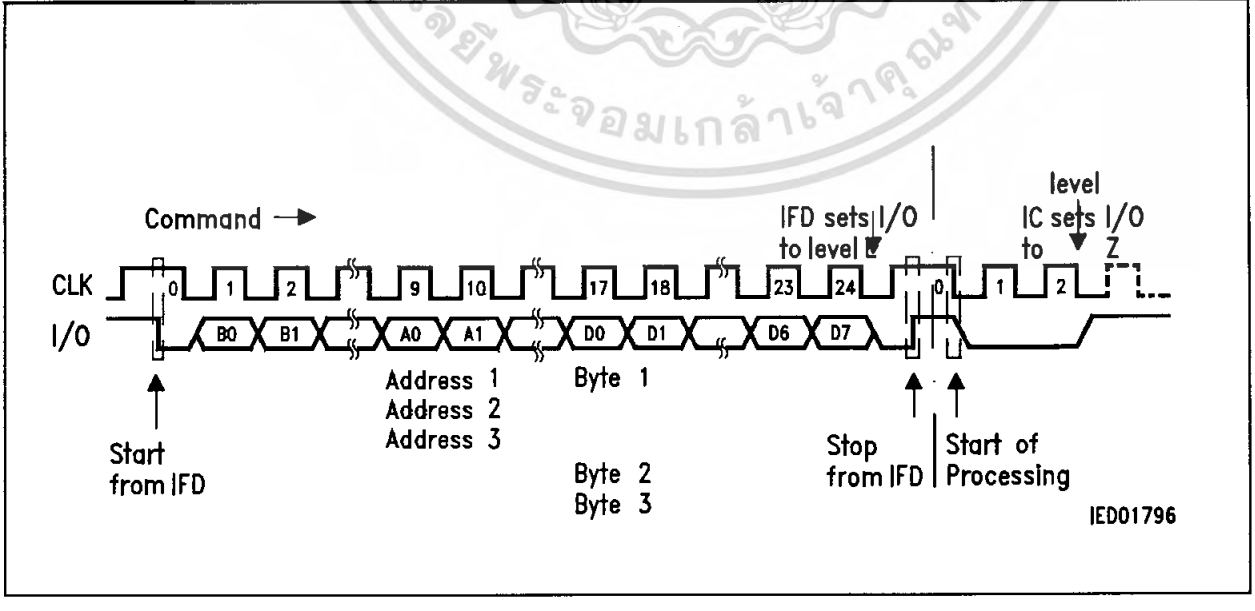


Figure 10
Compare Verification Data

2.4 PSC Verification (SLE 4442 only)

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

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

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

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

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

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