สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบเข้าคิวใช้บริการโรงพยาบาลโดยใช้บัตรสมาร์ตการ์ด

Queuing System Service Of Hospital by smartcard

โดย

นาย วรศิลป์ พันทวี

นาย สำรอง สีลาดเลา

เลขหมู่..........
เลขทะเบียน 72754
วัน,เดือน,ปี 22 มิ.ย. 2550

b. 11292268
i..........

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

ผ่านการตรวจชิ้นงานแล้ว
(ลงชื่อ)..........ผู้ตรวจ

ภาควิชา
วิศวกรรมโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเข้าคิวใช้บริการโรงพยาบาลโดยใช้บัตรสมาร์ตการ์ด

Queuing System Service Of Hospital by smartcard

โดย

นาย วรศิลป์ พันทวี 47015749

นาย สำรอง สีลาดเลา 47015753

อาจารย์ที่ปรึกษา

ผศ. สุรพล บุญจันทร์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **ระบบเข้าคิวใช้บริการโรงพยาบาลโดยใช้บัตรสมาร์ตการ์ด**

Queuing System Service Of Hospital by smartcard

ผู้จัดทำ

1. **นายวรศิลป์ พันทวี 47015749**
2. **นายสำรอง สีฉาดเฉา 47015753**

.... อาจารย์ที่ปรึกษา

(ผศ.สุรพล บุญจันทร์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเข้าคิวใช้บริการโรงพยาบาลโดยใช้บัตรสมาร์ตการ์ด

Queuing System Service Of Hospital by smartcard

โดย นายวรศิลป์ พันทวี 47015749

นายสำรอง สีลาดเลา 47015753

อาจารย์ที่ปรึกษา ผศ.สุรพล บุญจันทร์

บทคัดย่อ

โครงงานนี้มีวัตถุประสงค์ในการจัดระบบคิวของผู้ใช้บริการในโรงพยาบาลโดยใช้บัตรสมาร์ตการ์ด เนื่องจากอนาคตอันใกล้นี้ผู้ประดิษฐ์คาดว่าจะมีการนำบัตรสมาร์ตการ์ดมาใช้กันมาก เพื่อทำให้เกิดความสะดวกในการใช้บริการ โดยเครื่องอ่านข้อมูลจากบัตรสมาร์ตการ์ดแล้วนำข้อมูลมาเก็บไว้ในคอมพิวเตอร์ และบอกผู้ใช้บริการทราบว่าได้หมายเลขคิวที่เท่าไหร่ด้วยบอร์ดแสดงผลเป็นตัวเลข โดยที่เคาน์เตอร์บริการโรงพยาบาลจะแสดงหมายเลขคิวและช่องบริการพร้อมมีเสียงเรียกให้ผู้ใช้บริการทราบ

ABSTRACT

The objective of this project is to create a queuing system service of hospital by smartcards. We think smartcard will be usefull device for in the future. This system will effective in hospital service. The smartcard reader can receive information and store in computer. This system device will queued number and show to user by 7 segment board. Each counters in hospital will show queued number and sound calling for user to know.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จขึ้นได้นั้น ต้องขอขอบพระคุณ ผศ.สุรพล บุญจันทร์ อาจารย์ที่ปรึกษาปริญญานิพนธ์ ผู้ซึ่งแนะนำและเสนอแนวความคิดมุมมองใหม่ๆ ให้คำปรึกษารวมทั้งดูแลและตรวจทานจนกระทั่งทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จได้

ขอขอบพระคุณ คณะอาจารย์ในภาควิชาวิศวกรรมโทรคมนาคมทุกๆท่าน รวมถึงพี่ธีรศักดิ์ จันทร์วิเมลือง และนาย พยอม ศรีพิมพ์ ตลอดจนรุ่นพี่ภาคต่อเนื่อง ที่และให้คำปรึกษาและให้ความช่วยเหลือด้านต่างๆโดยตลอดมา

ท้ายที่สุดนี้ คณะผู้จัดทำขอกล่าวถึงบุคคลที่มีความสำคัญที่สุด คือ บิดามารดา บุคคลที่ซึ่งให้กำลังใจ รวมถึงให้ความสนับสนุนในทุกด้านมาโดยตลอดมา และเป็นบุคคลที่ทำให้คณะผู้จัดทำมีวันนี้ขึ้นมาได้ จึงขอกราบขอบพระคุณมา ณ ที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เนื้อหา	หน้าที่
บทที่ 1 บทนำ	1
1.1 ที่มาของหัวข้อปริญญานิพนธ์	1
1.2 วัตถุประสงค์ของปริญญานิพนธ์	1
1.3 ขอบเขตของปริญญานิพนธ์	1
1.4 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์	1
1.5 เนื้อหาของปริญญานิพนธ์	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 บัตรสมาร์ตการ์ด	3
2.1.1 สมาร์ตการ์ดคืออะไร	3
2.1.2 ส่วนประกอบและโครงสร้างของสมาร์ตการ์ด	3
2.1.2.1 ตัวบัตรพลาสติก	4
2.1.2.2 หน้าสัมผัสและชิปสมาร์ตการ์ด (Smart card Module)	4
2.1.3 การประกอบสมาร์ตการ์ดโมดูลลงในบัตรพลาสติก	5
2.1.3.1 การสร้างสมาร์ตการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก (TAB – Tape automated bonding)	5
2.1.3.2 การสร้างสมาร์ตการ์ดด้วยวิธีการวางสมาร์ตการ์ดโมดูลลงในเนื้อบัตร (Chip-On-Flex)	6
2.1.3.3 การสร้างสมาร์ตการ์ดด้วยวิธีการสร้างหน้าสัมผัสผิวของบัตร (Chip-On-Surface)	6
2.1.4 มาตรฐานที่เกี่ยวข้อง	7
2.1.4.1 มาตรฐาน ISO7810 – ISO7811 และ ISO7813	7
2.1.4.2 มาตรฐาน IS07816	7
2.1.5 ชนิดของสมาร์ตการ์ด	8
2.1.5.1 Memory Card (Synchronous card)	9
2.1.5.2 Processor Card (Asynchronous Card)	10
2.1.5.3 Contactless Card	10
2.1.5.4 Com – Bi Card	10
2.1.5.5 Hybrid Card	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เนื้อหา **หน้าที่**

2.1.6 การเชื่อมต่อชิปสมาร์ตการ์ด 12

2.1.7 การสื่อสารกับชิปสมาร์ตการ์ดในระดับสัญญาณไฟฟ้า 12

2.1.8 การรีเซตชิปสมาร์ตการ์ด (Card Present) 13

2.1.9 ส่วนประกอบของข้อมูล ATR(Answer To Reset) 15

2.1.9.1 Initial character (TS) 16

2.1.9.2 Format character (TO) 17

2.1.9.3 Interface characters (TA,TB,TC,TD) 17

2.1.9.4 Historical character (T1, T2 , T3, ... Tn) 22

2.1.9.5 Check character (TCK) 22

2.1.10 การเลือกชนิดโปรโตคอลสำหรับติดต่อกับสมาร์ตการ์ด 22

2.1.11 โปรโตคอลที่ใช้ในการสื่อสารกับสมาร์ตการ์ด 26

2.1.11.1 โปรโตคอล I^2C (Inter- Integrated circuit) 27

2.1.11.2 โปรโตคอล T = 0 29

2.1.11.3 โปรโตคอล T = 1 30

2.1.11.4 โปรโตคอล APDU 34

2.1.12 การเชื่อมต่อชิปสมาร์ตการ์ด (Smart card Interface) 36

2.1.12.1 การเชื่อมต่อชิปสมาร์ตการ์ดชนิด Memory (Synchronous card Interface) 38

2.1.12.2 การเชื่อมต่อชิปสมาร์ตการ์ดชนิด Processor (Asynchronous card Interface) 38

2.1.13 สมาร์ตการ์ดรีดเดอร์ 39

2.1.13.1 สมาร์ตการ์ดรีดเดอร์ชนิดหน้าสัมผัส 39

2.1.13.2 สมาร์ตการ์ดรีดเดอร์ชนิด Contactless 41

2.2 การรับส่งข้อมูล 41

2.2.1 การส่งข้อมูลแบบขนาน (Parallel transmission) 41

2.2.2 การส่งข้อมูลแบบอนุกรม (Serial transmission) 42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

เนื้อหา **หน้าที่**

2.2.3 การรับส่งข้อมูลแบบอนุกรม (RS-232) 42

2.2.4 แบบอะซิงโครนัส (Asychonous transmission) 44

2.2.5 แบบซิงโครนัส (Synchronous Transmission) 44

2.3 ไมโครคอนโทรลเลอร์ MCS – 51 44

2.3.1 ความหมายของไมโครคอนโทรลเลอร์ 44

2.3.2 ไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชจาก ATMEL 44

2.3.3 การจัดการขาของไมโครคอนโทรลเลอร์ MCS-51 45

2.4 จอแสดงผลแบบผลึกของเหลว (Liquid Crystal Display : LCD) 47

2.4.1 พื้นฐานในการใช้งานจอแสดงผลแบบผลึกของเหลว 48

บทที่ 3 การสร้างและการออกแบบ 53

3.1 ออกแบบและสร้างวงจรเครื่องอ่าน – เขียนสมาร์ตการ์ด 53

3.2 เขียนโปรแกรมการการอ่านข้อมูลจากสมาร์ตการ์ด 55

3.2.1 การเขียนโฟล์วชาร์ทควบคุมการอ่านข้อมูลจากบัตร 56

3.2.2 การเขียนโฟล์วชาร์ทควบคุมกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว 57

3.2.3 เขียนโปรแกรมในส่วนของเครื่องคอมพิวเตอร์จะได้โฟล์วชาร์ทดังนี้ 58

บทที่ 4 การทดลองและผลการทดลอง 59

4.1 การทดลองเครื่องอ่านเขียนสมาร์ตการ์ด 59

4.1.1 แสดงผลที่จอ LCD เมื่อเครื่องอ่านเขียนยังไม่มีบัตรสมาร์ตการ์ดเสียบเข้ามา 59

4.1.2 แสดงการเขียนบันทึกข้อมูลหมายเลขลงบัตรสมาร์ตการ์ด 59

4.2 ขั้นตอนระบบคิวใช้บริการ 60

4.3 การทำงานระบบของเสียงพูด 62

4.4 การแสดงผลที่ seven segment 65

4.5 การทดลองการรับข้อมูลผ่านพอร์ตอนุกรม RS-232 66

บทที่ 5 สรุปและวิจารณ์ 69

ภาคผนวก ก

ภาคผนวก ข

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง		หน้าที่
ตารางที่ 2.1	ชื่อหน้าสัมผัสและหน้า	9
ตารางที่ 2.2	ตารางแสดงความหมายของ Format character	17
ตารางที่ 2.3	อัตราหารความถี่สัญญาณนาฬิกา	17
ตารางที่ 2.4	Adjustment factor	18
ตารางที่ 2.5	รูปแบบข้อมูล TB	18
ตารางที่ 2.6	รูปแบบข้อมูล TC1	19
ตารางที่ 2.7	รูปแบบของข้อมูล TDi	19
ตารางที่ 2.8	รูปแบบของข้อมูล TA2	20
ตารางที่ 2.9	รูปแบบของข้อมูล TB2	20
ตารางที่ 2.10	รูปแบบของข้อมูล TD	20
ตารางที่ 2.11	รูปแบบของข้อมูล TAi เมื่อ i > 2	21
ตารางที่ 2.12	รูปแบบของข้อมูล TBi เมื่อ i > 2	21
ตารางที่ 2.13	รูปแบบของข้อมูล Tci เมื่อ i > 2	22
ตารางที่ 2.14	รูปแบบของข้อมูล PTS0	25
ตารางที่ 2.15	รูปแบบของข้อมูล PTS1	26
ตารางที่ 2.16	รูปแบบของข้อมูล PTS2	26
ตารางที่ 2.17	รูปแบบของข้อมูล NAD	31
ตารางที่ 2.18	รูปแบบข้อมูล PCB ชนิด I block	31
ตารางที่ 2.19	รูปแบบข้อมูล PCB ชนิด R block	32
ตารางที่ 2.20	รูปแบบข้อมูล PCB ชนิด S block	32
ตารางที่ 2.21	ส่วนประกอบของโปรโตคอล APDU	34
ตารางที่ 2.22	ชนิดของ Status Word ตามชนิดของสมาร์ตการ์ดและการประมวลผล	36
ตารางที่ 2.23	การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB และ DB-25	43
ตารางที่ 2.24	การกำหนดคุณสมบัติต่างๆ ในการใช้งานของจอแสดงผลแบบผลึกของเหลว	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพ **หน้าที่**

รูปที่ 1.1 บล็อกแสดงการทำงานของระบบ 2

รูปที่ 2.1 ส่วนประกอบของสมาร์ตการ์ด 3

รูปที่ 2.2 การแบ่งชนิดของบัตรตามรูปร่างที่นำไปใช้งาน และขนาด 4

รูปที่ 2.3 การประกอบสมาร์ตการ์ดโมดูลด้วยวิธีการทับซ้อนของชั้นพลาสติก 5

รูปที่ 2.4 การสร้างสมาร์ตการ์ดด้วยวิธีการวางสมาร์ตการ์ดโมดูลลงในเนื้อบัตร 6

รูปที่ 2.5 การสร้างสมาร์ตการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวบัตร 6

รูปที่ 2.6 การแบ่งสมาร์ตการ์ดตามชนิดของหน่วยความจำ และประเภทของหน้าสัมผัส 8

รูปที่ 2.7 ตำแหน่งของหน้าสัมผัสของสมาร์ตการ์ด 9

รูปที่ 2.8 โครงสร้างภายในของสมาร์ตการ์ดชนิด Com - Bi Card 11

รูปที่ 2.9 โครงสร้างภายในของสมาร์ตการ์ดชนิด Hybrid Card 11

รูปที่ 2.10 ตัวอย่างรูปแบบสัญญาณของการสื่อสารแบบซิงเกิลบัสแบบ Direct convention และ Converse convention 12

รูปที่ 2.11 ตัวอย่างข้อมูลที่ใช้ในการรับ-ส่งข้อมูลกับสมาร์ตการ์ด 13

รูปที่ 2.12 ไทม์มิ่งไดอะแกรมการรีเซตชิปสมาร์ตการ์ด 14

รูปที่ 2.13 โครงสร้างและส่วนประกอบของข้อมูล ATR 15

รูปที่ 2.14 ข้อมูล Initial character ในรูปแบบของ Direct convention และ Converse convention 16

รูปที่ 2.15 ขั้นตอนตรวจสอบการเปลี่ยนโปรโตคอลของสมาร์ตการ์ด 23

รูปที่ 2.16 ขั้นตอนการเปลี่ยนโปรโตคอลของสมาร์ตการ์ด 24

รูปที่ 2.17 โครงสร้างข้อมูล PTS 25

รูปที่ 2.18 รายเอียดของการสื่อสารแบบ Synchronous และ Asynchronous ในสมาร์ตการ์ด 26

รูปที่ 2.19 สัญญาณเริ่มต้น และสิ้นสุดการส่งข้อมูล 27

รูปที่ 2.20 การอ่านข้อมูลจากหน่วยความจำข้อมูลของสมาร์ตการ์ด ชนิด Memory ด้วยโปรโตคอล I^2C 28

รูปที่ 2.21 การเขียนข้อมูลลงในหน่วยความจำข้อมูลของสมาร์ตการ์ดชนิด Memory ด้วยโปรโตคอล I^2C 28

รูปที่ 2.22 รูปแบบคำสั่งของโปรโตคอล T = 0 29

รูปที่ 2.23 ไทม์มิ่งไดอะแกรมของการรับส่งข้อมูลด้วยโปรโตคอล T = 0 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพ	หน้าที่
รูปที่ 2.24 ข้อมูลตอบกับจากชิปสมาร์ตการ์ดในโปรโตคอล T = 0	30
รูปที่ 2.25 รูปแบบของโปรโตคอล T=0	30
รูปที่ 2.26 รูปแบบการรับส่งข้อมูลโปรโตคอล T=0	33
รูปที่ 2.27 รูปแบบการส่งข้อมูล S block จากสมาร์ตการ์ด เมื่อสมาร์ตการ์ด ต้องประมวลผลเป็นเวลานาน	33
รูปที่ 2.28 รูปแบบชุดคำสั่งของโปรโตคอล APDU	34
รูปที่ 2.29 การส่งชุดคำสั่งให้แก่ชุดสมาร์ตการ์ดทั้ง 4 แบบ	35
รูปที่ 2.30 ข้อมูลตอบกลับจากสมาร์ตการ์ดทั้ง 2 แบบ	36
รูปที่ 2.31 ตัวอย่างวงจรสำหรับติดต่อกับชิปสมาร์ตการ์ด	37
รูปที่ 2.32 ตัวอย่างวงจรเชื่อมต่อกับชิปสมาร์ตการ์ดด้วยวงจรลอจิก (C-MOS)	37
รูปที่ 2.33 ตัวอย่างวงจรเชื่อมต่อชิปสมาร์ตการ์ดชนิด Memory	38
รูปที่ 2.34 ตัวอย่างวงจรเชื่อมต่อชิปสมาร์ตการ์ดชนิด Processor	38
รูปที่ 2.35 ตัวอย่างวงจรเชื่อมต่อกับชิปสมาร์ตการ์ดด้วย IC เบอร์ TDA8006AH	39
รูปที่ 2.36 ตัวอย่างสมาร์ตการ์ดรีดเดอร์ที่ใช้บัตรชนิด ID-0I	40
รูปที่ 2.37 ตัวอย่างสมาร์ตการ์ดรีดเดอร์ที่ใช้บัตรชนิด ID-000	40
รูปที่ 2.38 การส่งข้อมูลแบบขนาน (Parallel transmission)	41
รูปที่ 2.39 การส่งข้อมูลแบบอนุกรม (Serial transmission)	42
รูปที่ 2.40 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB และ DB-25	42
รูปที่ 2.41 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ	43
รูปที่ 2.42 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS – 51 ตามมาตรฐาน Intel	46
รูปที่ 2.43 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชของ Atmel	47
รูปที่ 3.1 แสดงวงจรการทำงานของเครื่องอ่าน – เขียนสมาร์ตการ์ด	53
รูปที่ 3.2 แสดงวงจรการทำงานไอซี MAX 232 กับ พอร์ตอนุกรม (DB9)	54
รูปที่ 3.3 โฟล์วชาร์ทการทำงานของโปรแกรมการอ่านข้อมูลจากบัตรสมาร์ตการ์ด	55
รูปที่ 3.4 โฟล์วชาร์ทการทำงานของโปรแกรมควบคุมการอ่านข้อมูลจากบัตร	56
รูปที่ 3.5 โฟล์วชาร์ทการทำงานของโปรแกรมการกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว	57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ(ต่อ)

ภาพ		หน้าที่
รูปที่ 3.6	ไฟล์วชาร์ทแสดงการทำงานของ โปรแกรมรับข้อมูลจากบัตรคิวพร้อมส่งเรียกบอกหมายบัตรมาที่เคาน์เตอร์บริการ	58
รูปที่ 4.1	แสดงผลเมื่อยังไม่มีการเสียบบัตรสมาร์ตการ์ด	59
รูปที่ 4.2	แสดงการบันทึกหมายเลขลงบัตรสมาร์ตการ์ด	59
รูปที่ 4.3	แสดงหน้าจอโปรแกรมเมื่อมีการเสียบบัตรคิวเข้ามา	60
รูปที่ 4.4	แสดงหน้าจอโปรแกรมเมื่อยังไม่มีหมายเลขบัตรคิวเข้ามา	61
รูปที่ 4.5	แสดงหน้าจอโปรแกรมเมื่อมีหมายเลขบัตรคิวเข้ามา	61
รูปที่ 4.6	แสดงหน้าจอโปรแกรมเมื่อกดปุ่มหมายเลขบัตรคิวออกไป	62
รูปที่ 4.7	แสดงกราฟเสียงพูด "หนึ่ง"	62
รูปที่ 4.8	แสดงกราฟเสียงพูด "สอง"	63
รูปที่ 4.9	แสดงกราฟเสียงพูด "สาม"	63
รูปที่ 4.10	แสดงกราฟเสียงพูด "เชิญหมายเลข"	63
รูปที่ 4.11	แสดงกราฟเสียงพูด " ที่ช่องบริการ"	64
รูปที่ 4.12	แสดงกราฟเสียง "ค่ะ"	64
รูปที่ 4.13	แสดงผล seven segment	65
รูปที่ 4.14	การอ่านข้อมูลจากบัตรสมาร์ดการ์ดผ่านพอร์ตอนุกรม RS-232	66
รูปที่ 4.15	การวัดสัญญาณเทียบกับ GROUND	67
รูปที่ 4.16	การวัดสัญญาณเทียบที่ขา TX ในขณะที่ยังไม่อ่านการ์ด	67
รูปที่ 4.17	การวัดสัญญาณเทียบที่ขา TX ขณะที่มีการอ่านข้อมูลในการ์ด	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มาของหัวข้อปริญญานิพนธ์

ในปัจจุบันนี้ เมื่อเราเดินทางไปหาหมอที่คลินิกหรือโรงพยาบาลพบว่าในแต่ละวันมีผู้มาใช้บริการเป็นจำนวนมากซึ่ง โรงพยาบาลบางที่ยังใช้ระบบคิวแบบเรียกผู้มาใช้บริการทีละคนตามบัตรประชาชนที่วางไว้ในตะกร้าก่อนซึ่งในแต่ละวันนั้นมีผู้มาใช้บริการเป็นจำนวนมากการที่จะเรียกรายชื่อทุกคนตามคิวมาใช้บริการนั้นจะเหนื่อยมาก ดังนั้นต้องมีระบบคิวที่มีบัตรคิวพร้อมกับมีเสียงพูดเรียกรายชื่อแต่ละคนตามบัตรคิวมาใช้บริการที่เคาน์เตอร์ของโรงพยาบาล

โครงงานนี้มีวัตถุประสงค์ในการจัดระบบคิวของผู้ใช้บริการในโรงพยาบาลโดยใช้บัตรสมาร์ตการ์ด เพื่อทำให้เกิดความสะดวกในการใช้บริการซึ่งเราจะบันทึกหมายเลขคิวบริการลงในบัตรสมาร์ตการ์ด จากนั้นนำบัตรไปเสียบที่เครื่องอ่านสมาร์ตการ์ดโดยเครื่องอ่านข้อมูลจากบัตรสมาร์ตการ์ดแล้วนำข้อมูลมาเก็บไว้ในคอมพิวเตอร์ และบอกผู้ใช้บริการทราบว่าได้หมายเลขคิวที่เท่าไหร่ด้วยบอร์ดแสดงผลเป็นตัวเลข โดยที่เคาน์เตอร์บริการโรงพยาบาลจะแสดงหมายเลขคิวและช่องบริการพร้อมมีเสียงเรียกให้ผู้ใช้บริการทราบตามหมายเลขคิว

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1.2.1 เพื่อศึกษาเกี่ยวกับการเขียนโปรแกรมไมโครคอนโทรลเลอร์ และการประยุกต์ใช้งาน

1.2.2 เพื่อศึกษาการทำงานของสมาร์ตการ์ด และการประยุกต์ใช้งาน

1.2.3 เพื่อศึกษาการเชื่อมต่อไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก

1.2.4 เพื่อศึกษาการเขียนโปรแกรมด้านแอพพลิเคชั่นบนวินโดว์

1.3 ขอบเขตของปริญญานิพนธ์

ทำการสร้างเครื่องอ่าน – เขียนสมาร์ตการ์ดเพื่อใช้กับบัตรสมาร์ตการ์ด โดยจะใช้บัตรสมาร์ตการ์ดแทนลำดับคิวพร้อมสามารถบันทึกข้อมูลผู้ใช้บริการได้ และมีเสียงเรียกลำดับหมายเลขคิวมาที่ช่องเคาน์เตอร์บริการโรงพยาบาล

1.4 ประโยชน์ที่คาดว่าจะได้รับจากการทำปริญญานิพนธ์

1.4.1 เข้าใจหลักการทำงานของไมโครคอนโทรลเลอร์

1.4.2 สามารถเขียนโปรแกรมไมโครคอนโทรลเลอร์ติดต่อกับอุปกรณ์ภายนอกได้

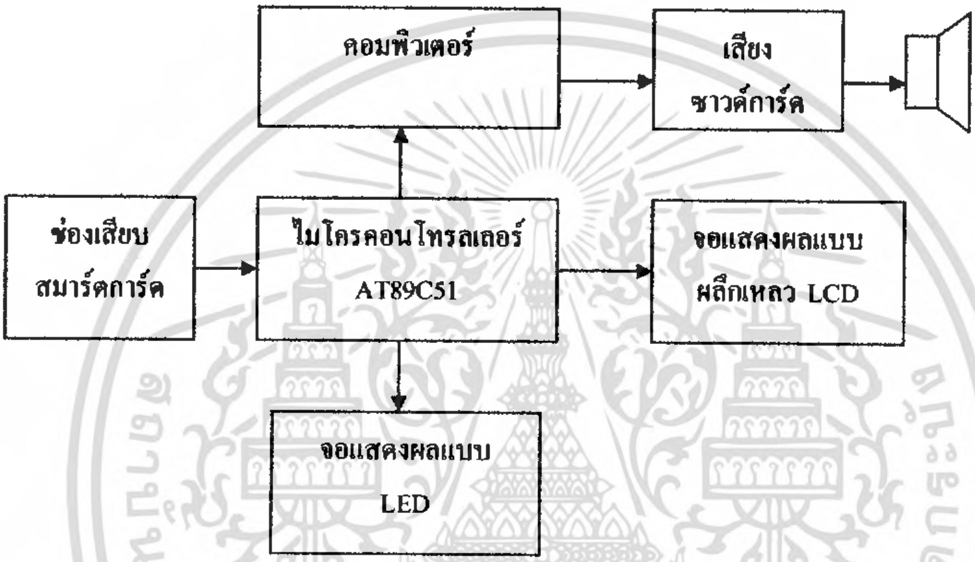
1.4.3 เพื่อความสะดวกและเป็นระเบียบเรียบร้อยในการใช้บริการโรงพยาบาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.4 สามารถเขียนโปรแกรมด้านแอพพลิเคชั่นได้

1.4.5 สามารถนำสมาร์ตการ์ดไปประยุกต์ใช้งานกับงานด้านอื่นๆได้

1.5 เนื้อหาของปริญญานิพนธ์



รูปที่ 1.1 บล็อกแสดงการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 สมาร์ตการ์ด

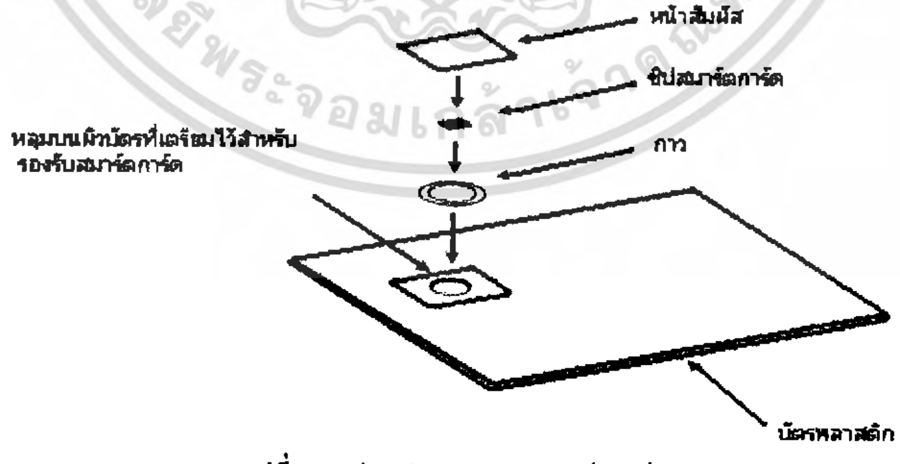
2.1.1 สมาร์ตการ์ดคืออะไร

สมาร์ตการ์ดคือบัตรพลาสติกที่มีชิป IC (Integrated circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเอง โดยวิธีการเข้ารหัสตามมาตรฐาน DES Algorithm (Data Encryption standard) เพื่อให้ระบบมีระดับความปลอดภัยสูงขึ้น ด้วยคุณสมบัติสำคัญประการหนึ่งที่ทำให้สมาร์ตการ์ดมีความแตกต่างจากบัตรพลาสติกทั่วไปก็คือ ขณะทำรายการ(Transaction) สมาร์ตการ์ดสามารถทำงานด้วยตัวมันเองโดยไม่ต้องอาศัยการติดต่อสื่อสารกับระบบหลัก(Front end) นั้นก็คือสมาร์ตการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสาร กับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก(off-line)ทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก

ในการประมวลผล และจัดเก็บข้อมูล สมาร์ตการ์ดสามารถทำได้อย่างรวดเร็วกว่าสื่อสำหรับจัดเก็บข้อมูลชนิดอื่นๆด้วยขนาดที่เท่ากับบัตรแถบแม่เหล็กทำให้สะดวกในการจัดเก็บและพกพา ซึ่งสมาร์ตการ์ดยังมีคุณสมบัติด้านความทนทานที่น่าทึ่งไม่ว่าจะเป็น พวกรังสีชนิดต่างๆ(ในปริมาณที่ไม่เป็นอันตรายต่อมนุษย์) รวมทั้งสนามแม่เหล็ก ไฟฟ้าสถิต ความชื้น ความร้อน การบิดงอ ฯลฯ ก็ไม่สามารถทำให้สมาร์ตการ์ดเสียหายได้โดยง่าย จึงทำให้สมาร์ตการ์ดเทียบเท่าบัตรในทางอุดมคติเลยที่เดียว ในต่างประเทศก็ได้มีการใช้งานบัตรสมาร์ตการ์ดแพร่หลายรวมถึงในประเทศไทยด้วย จนเป็นไปว่าสมาร์ตการ์ดกำลังเป็นบัตรชนิดใหม่ที่จะเข้ามาแทนที่บัตรแถบแม่เหล็กที่ใช้งานกันอยู่ในปัจจุบัน

2.1.2 ส่วนประกอบและโครงสร้างของสมาร์ตการ์ด

สมาร์ตการ์ดประกอบด้วยบัตรพลาสติก กาวหรือวัสดุที่ใช้เชื่อมต่อ และหน้าสัมผัสที่บรรจุชิปสมาร์ตการ์ดเรียบร้อยแล้ว ซึ่งส่วนประกอบต่างๆแสดงให้เห็นดังรูป



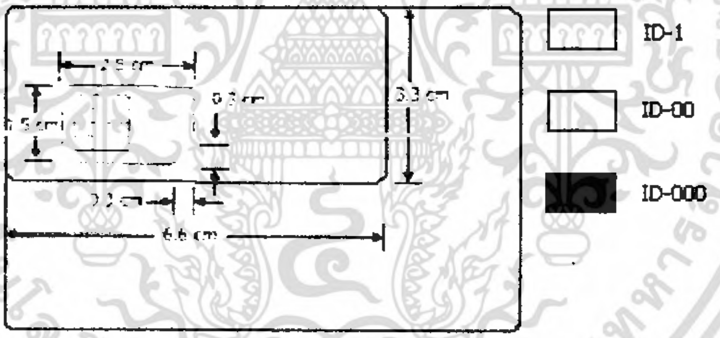
รูปที่ 2.1 ส่วนประกอบของสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.1 ตัวบัตรพลาสติก

สมาร์ตการ์ดเป็นชิป IC ขนาดเล็กที่ถูกสร้างขึ้นเช่นเดียวกับเช่นเดียวกับชิ้นส่วนอิเล็กทรอนิกส์อื่นๆ ที่สร้างจากสารกึ่งตัวนำ นำมาติดลงบนหน้าสัมผัสและทำการฝังลงในเนื้อบัตรพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นตัวบัตรจะใช้พลาสติก 4 ชนิดได้แก่ PVC (Polyvinyl Choloride), ABS (acrylonitrile Butadiene styrene), PC (polycarbonate), และ PET (polyethylene Terephthalate) ในประเทศไทยจะใช้บัตรพลาสติกชนิด PVC มากเป็นอันดับหนึ่ง ส่วนอันดับสองเป็นบัตรพลาสติกชนิด ABS ซึ่งบัตรพลาสติกชนิด PVC มักนำมาใช้เป็นบัตร ATM, บัตรเครดิต-เดบิต,บัตรประจำตัวประชาชน ฯลฯ ส่วนบัตรพลาสติกชนิด ABS ไม่ค่อยพบว่าใช้งานกันมามากนักเนื่องจากราคาสูงกว่า และลายที่พิมพ์ลงบนบัตรไม่สวยงามไม่คงทนเท่าบัตรพลาสติกชนิด PVC จะพบก็เพียงบัตรพลาสติกเนื้อผสมโดยใช้พลาสติกชนิด ABS เป็นแกนและฉาบผิวด้วยพลาสติกชนิด PVC ซึ่งจะทำให้ลายที่พิมพ์ลงบนบัตรสวยงามคงทนเท่าบัตรพลาสติกชนิด PVC แต่ความทนทานของตัวบัตรจะสู้บัตรพลาสติกเนื้อ PVC ล้วนไม่ได้

สำหรับบัตรพลาสติกอีกสองชนิดที่เหลือ ยังไม่ได้พบว่ามีการใช้งานในประเทศไทย อาจเนื่องมาจากราคาที่สูงเกินไปของวัสดุที่นำมาใช้ทำเป็นตัวบัตร และคุณสมบัติของวัสดุที่ด้อยกว่าพลาสติกชนิด PVC แต่ข้อเสียที่สำคัญของพลาสติกชนิด PVC ก็ไม่ด้อยไปกว่าข้อดีของมัน นั้นก็คือมันไม่สามารถย่อยสลายได้ตามธรรมชาติ ซึ่งเท่ากับเป็นขยะสำหรับสิ่งแวดล้อมเลยทีเดียว



รูปที่ 2.2 การแบ่งชนิดของบัตรตามรูปร่างที่นำไปใช้งาน และขนาด

2.1.2.2 หน้าสัมผัสและชิปสมาร์ตการ์ด (Smart card Module)

สมาร์ตการ์ดโมดูลหรือหน้าสัมผัสและชิปสมาร์ตการ์ด คือส่วนที่แสดงความเป็นตัวตนของสมาร์ตการ์ดที่ชัดเจนที่สุด สมาร์ตการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่ามันคือ สมาร์ตการ์ดที่มีการฝังชิปไว้ในเนื้อบัตร ดังนั้นการที่จะระบุว่าบัตรใบใดเป็นบัตรสมาร์ตการ์ดนั้น ต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก ซึ่งต้องใช้ประสบการณ์เกี่ยวกับสมาร์ตการ์ดพอสมควร แต่ในที่นี้จะขอแนะนำให้เห็นภาพลักษณ์ที่ชัดเจนของสมาร์ตการ์ดเป็นหลัก ซึ่งก็คือส่วนของการผลิตสมาร์ตการ์ดโมดูล ส่วนที่เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

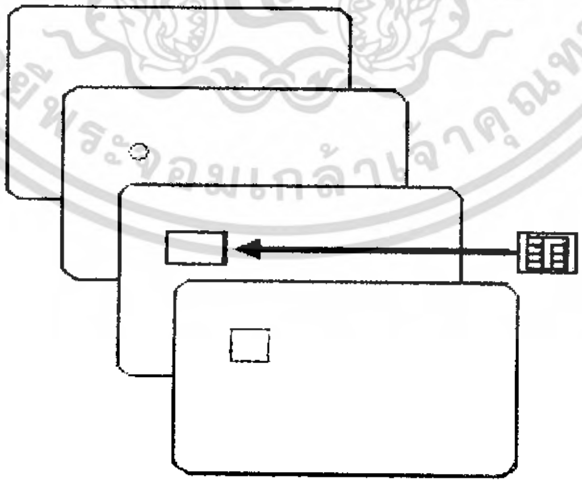
หน้าสัมผัสของสมาร์ตการ์ดประกอบด้วยโลหะหลายชิ้นประกอบกัน แต่ส่วนจะถูกยึดด้วยแถบฟิล์มบางๆ ทางด้านหลังของหน้าสัมผัสเพื่อให้คงรูปอยู่ได้แถบฟิล์มจะมีการเจาะช่องเล็กๆ สำหรับการเชื่อมต่อสายนำสัญญาณกับชิปสมาร์ตการ์ดกับหน้าสัมผัส หลังจากที่วาง ชิปสมาร์ตการ์ดลงในตำแหน่งที่ต้องการ และเชื่อมต่อนำสัญญาณจากชิปสมาร์ตการ์ดเข้าหน้าสัมผัสเรียบร้อยแล้ว ขั้นตอนสุดท้ายจะเป็นการผนึกชิปสมาร์ตการ์ดเพื่อป้องกันตัวชิป และสายนำสัญญาณต่างๆ จากสิ่งแวดล้อมภายนอก(เป็นการทดสอบขั้นต้น) ส่วนขั้นตอนที่เหลือ จะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติก และทดสอบการทำงานของชิปขั้นสุดท้ายมาร์ตการ์ดโมดูลนั้นเองใน

2.1.3 การประกอบสมาร์ตการ์ดโมดูลลงในบัตรพลาสติก

การประกอบสมาร์ตการ์ดโมดูลลงในบัตรนั้นมีหลายวิธีด้วยกันตามแต่ชนิดของสมาร์ตการ์ดโมดูลและการเตรียมบัตรพลาสติก ซึ่งการเตรียมบัตรพลาสติกจะนำมาใส่สมาร์ตการ์ดโมดูลมีด้วยกันสองแบบคือ บัตรพลาสติกที่ถูกขุดหลุมบนบัตร และบัตรพลาสติกที่เกิดจากการทับซ้อนของชั้นพลาสติกที่เจาะช่องมาแล้ว โดยสมาร์ตการ์ดโมดูลจะใช้บัตรพลาสติกที่มีการเตรียมการมาแล้วดังนี้

2.1.3.1 การสร้างสมาร์ตการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก (TAB – Tape automated bonding)

สมาร์ตการ์ดชนิดนี้เกิดจากการทับซ้อนของชั้นพลาสติกตั้งแต่ 3 ชั้นขึ้นไป โดยแต่ละชั้นจะมีการเจาะช่องตามขนาดของหน้าสัมผัสและชิปสมาร์ตการ์ดไว้อยู่แล้ว ส่วนที่เป็นหน้าสัมผัสและชิปจะถูกแทรกอยู่ชั้นในของพลาสติก เมื่อวางซ้อนกันเรียบร้อยแล้วก็จะทำการอัดแต่ละชั้นด้วยความร้อน เมื่อความร้อนถึงจุดที่ทำให้พลาสติกแต่ละชั้นประสานตัวเอง ก็จะทำการตัดแต่งขอบบัตรและทำการทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย

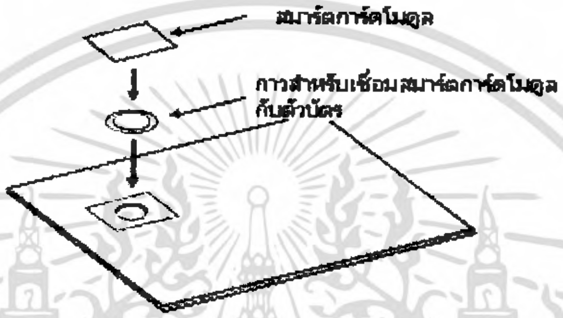


รูปที่ 2.3 การประกอบสมาร์ตการ์ดโมดูลด้วยวิธีการทับซ้อนของชั้นพลาสติก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 การสร้างสมาร์ตการ์ดด้วยวิธีการวางสมาร์ตการ์ดโมดูลลงในเนื้อบัตร (Chip-On-Flex)

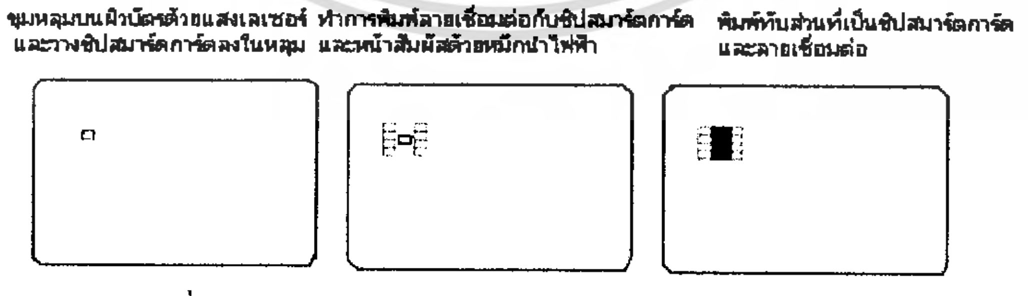
สมาร์ตการ์ดโมดูลที่จะนำมาใส่ลงในบัตรพลาสติก ผู้ผลิตจะทำการตัดตามขนาดของหลุมบนบัตรพลาสติกที่ขุดรอไว้แล้วด้วยเครื่องจักร ทำการเชื่อมด้วยกาว และอบด้วยความร้อนเพื่อให้สมาร์ตการ์ดโมดูลติดสนิทกับเนื้อพลาสติก จากนั้นจึงทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย การใส่หน้าสัมผัสและชิปสมาร์ตการ์ดด้วยกาวนี้ เป็นวิธีที่นิยมกันมากที่สุด เพราะผู้ผลิตสามารถประหยัดต้นทุนในการผลิตได้มาก เนื่องจากวิธีการนี้เสียค่าใช้จ่ายในการผลิตน้อยที่สุด ไม่ว่าจะเป็นเรื่องของแรงงาน ความรวดเร็วในการผลิต และเปอร์เซ็นต์สินค้าชำรุดต่ำ ในคุณภาพของสมาร์ตการ์ดที่ยังพอยอมรับได้



รูปที่ 2.4 การสร้างสมาร์ตการ์ดด้วยวิธีการวางสมาร์ตการ์ดโมดูลลงในเนื้อบัตร

2.1.3.3 การสร้างสมาร์ตการ์ดด้วยวิธีการสร้างหน้าสัมผัสผิวของบัตร (Chip-On-Surface)

สมาร์ตการ์ดโมดูลชนิดติดบนผิวบัตร ผลิตโดยด้วยการใช้แสงเลเซอร์ทำการขุดหลุมบนบัตรพลาสติกขนาดเท่ากับตัวชิปของสมาร์ตการ์ด วางชิปสมาร์ตการ์ดลงในตำแหน่งที่กำหนด สร้างหน้าสัมผัสและเชื่อมสายสัญญาณกับสมาร์ตการ์ดด้วยหมึกนำไฟฟ้า สุดท้ายพิมพ์ทับส่วนที่เป็นชิปและหมึกนำไฟฟ้าส่วนที่เป็นสายสัญญาณ ด้วยหมึกที่มีคุณสมบัติเป็นฉนวนไฟฟ้าเพื่อป้องกันวงจรภายใน โดยปล่อยส่วนที่เป็นหน้าสัมผัสที่สร้างจากหมึกที่นำไฟฟ้าไว้เท่านั้น สมาร์ตการ์ดชนิดนี้ไม่ค่อยมีให้เห็นมากนัก เนื่องจากต้องใช้เทคโนโลยีในการผลิตที่สูงกว่าหน้าสัมผัสแบบอื่น ทำให้ต้นทุนการผลิตสูงตามไปด้วย อีกทั้งความทนทานก็ยังน้อยกว่าโมดูลชนิดอื่นๆ



รูปที่ 2.5 การสร้างสมาร์ตการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4 มาตรฐานที่เกี่ยวข้อง

มาตรฐานที่เกี่ยวข้องกับสมาร์ตการ์ดมีด้วยกันหลายมาตรฐาน มาตรฐานหลายๆตัวมีเนื้อหาที่ซ้ำซ้อนกันเป็นผลให้สมาร์ตการ์ดเป็นเรื่องที่ค่อนข้างยุ่งยาก ถึงกระนั้น มาตรฐานเหล่านั้น มาตรฐานที่นำมาใช้กับสมาร์ตการ์ดนี้จะมีทั้งมาตรฐานที่มีอยู่แต่เดิมก่อนที่จะมีอยู่แต่เดิมก่อนที่จะมีสมาร์ตการ์ด และมาตรฐานที่กำหนดขึ้นสำหรับสมาร์ตการ์ดโดยเฉพาะซึ่งอ้างอิงจากมาตรฐานบัตรพลาสติกและบัตรพลาสติกและบัตรแถบแม่เหล็กที่มีอยู่แล้ว

2.1.4.1 มาตรฐาน ISO7810 – ISO7811 และ ISO7813

ก่อนที่จะเข้าถึงเรื่องของสมาร์ตการ์ดคงต้องกล่าวถึงมาตรฐานอีกตัวหนึ่งซึ่งมีความสำคัญพอๆ กับมาตรฐานของสมาร์ตการ์ดนั้นก็คือมาตรฐานของบัตรพลาสติก และบัตรแถบแม่เหล็ก(บัตรเครดิต) นั้นก็คือมาตรฐาน ISO7810 และ ISO7811 ซึ่งรายละเอียดจะขอกล่าวในบทถัดไป

ISO7810 :มาตรฐานที่กำหนดด้วยเรื่องของคุณสมบัติทางกายภาพเบื้องต้นของบัตรพลาสติก

ISO7811-1 :มาตรฐานที่กำหนดด้วยเรื่องของตัวอักษรนูนบนบัตร

ISO7811-2 :มาตรฐานที่กำหนดด้วยเรื่องของแถบแม่เหล็กบนบัตร (Low Coercivity)

ISO7811-3 :มาตรฐานที่กำหนดตำแหน่งของการพิมพ์ตัวอักษรนูนบนบัตร

ISO7811-4 :มาตรฐานที่กำหนดตำแหน่งที่อยู่ของข้อมูลแทร็ก 1 และ 2

ISO7811-5 :มาตรฐานที่กำหนดตำแหน่งที่อยู่ของข้อมูลแทร็ก 3

ISO7811-๖:มาตรฐานที่กำหนดด้วยเรื่องของแถบแม่เหล็กแบบความหนาแน่นสูงบนบัตร(High Coercivity)

ISO7813 :มาตรฐานที่กำหนดด้านการเงินการธนาคาร

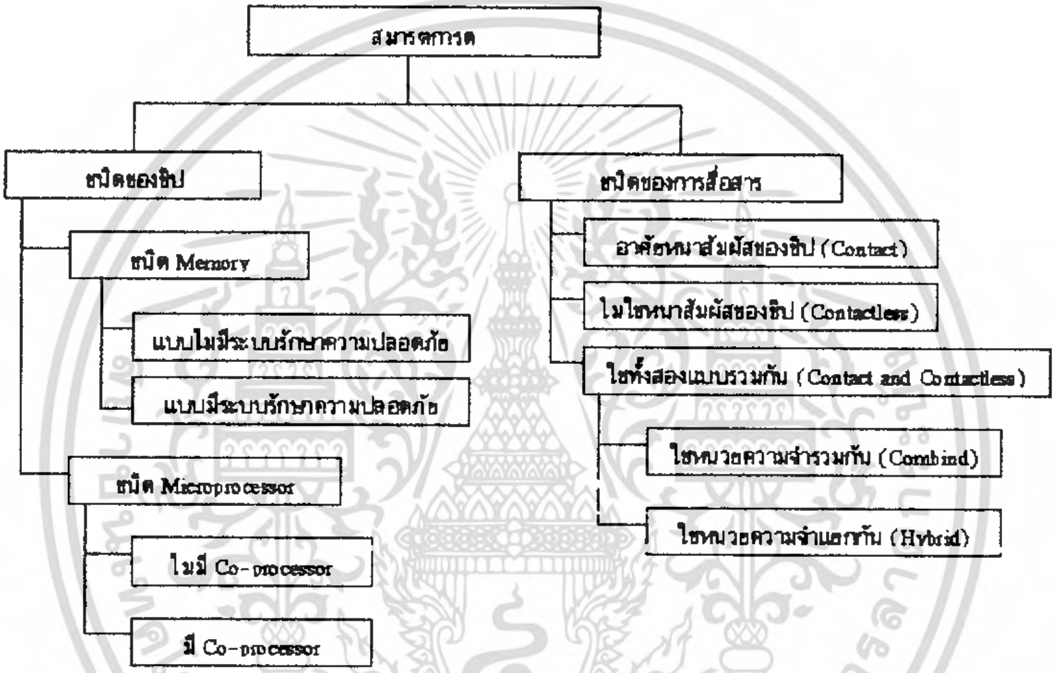
2.1.4.2 มาตรฐาน ISO7816

เพื่อให้เกิดความเข้ากันได้ของสมาร์ตการ์ด จึงมีการกำหนดมาตรฐานของสมาร์ตการ์ดในชื่อของ ISO7816 เป็นการกำหนดในเรื่องของคุณลักษณะของบัตรพลาสติกที่จะนำมาทำเป็นสมาร์ตการ์ด โดยมาตรฐาน ISO7816 มีหัวข้อย่อยโดยแบ่งเป็น ISO7816-1, ISO7816-2, ISO7816-3, และปัจจุบันมีถึง ISO7816-6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 ชนิดของสมาร์ตการ์ด

การแบ่งชนิดของสมาร์ตการ์ดในปัจจุบันอาจทำได้ยากสักหน่อย เนื่องจากมีการใส่เทคโนโลยีใหม่ๆ ลงสมาร์ตการ์ดตลอดเวลา ถ้าจะแบ่งตามชนิดของหน่วยความจำภายในอาจไม่ชัดเจนนักยิ่งแบ่งตามลักษณะการเชื่อมต่อ ก็คงไม่ครอบคลุมสมาร์ตการ์ดทั้งหมดดังนั้นจึงขอแสดงการแบ่งชนิดของสมาร์ตการ์ดให้เข้าใจได้ง่ายดังรูปที่ 1.7



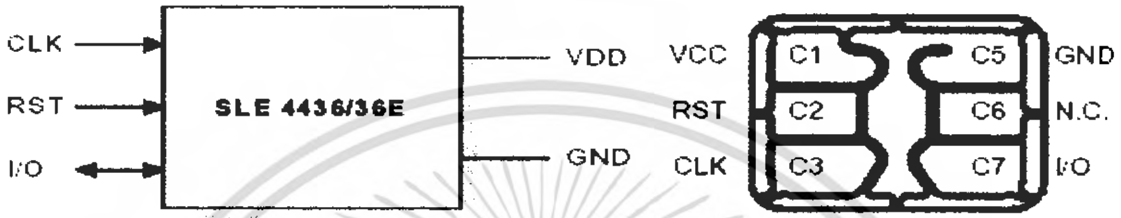
รูปที่ 2.6 การแบ่งสมาร์ตการ์ดตามชนิดของหน่วยความจำ และประเภทของหน้าสัมผัส

ถ้าเราจะแบ่งโดยใช้ชนิดของชิปสมาร์ตการ์ด สามารถแยกได้ 2 แบบคือ แบบ Memory ได้แก่พวกบัตรโทรศัพท์ อีกแบบหนึ่งคือ แบบไมโครโปรเซสเซอร์ ได้แก่ซิมการ์ดในโทรศัพท์มือถือ หรือถ้าจะยึดหน้าสัมผัสก็อาจจะแบ่งได้เป็น 3 แบบ คือ 1.แบบสื่อสารโดยใช้หน้าสัมผัส 2. แบบสื่อสารโดยไม่ใช้หน้าสัมผัส 3. ใช้ทั้งแบบมีหน้าสัมผัสและไม่มีรวมกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.1 Memory Card (Synchronous card)

สมาร์ตการ์ดชนิด Memory ใช้การสื่อสารแบบอนุกรมตามสัญญาณนาฬิกาที่ให้กับชิป และการรับส่งข้อมูลต้องสอดคล้องกับสัญญาณนาฬิกา สมาร์ตการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วย ขาอินพุทและเอาท์พุท หน่วยความจำข้อมูลหรือ RAM , หน่วยความจำสำหรับเก็บชุดคำสั่งหรือ ROM ขาต่างๆ ของสมาร์ตการ์ดยี่ห้อหนึ่งมีตำแหน่งของหน้าสัมผัสและหน้าที่การทำงานดังรูปและตารางต่อไปนี้



รูปที่ 2.7 ตำแหน่งของหน้าสัมผัสของสมาร์ตการ์ด

ตารางที่ 2.1 ชื่อหน้าสัมผัสและหน้าที่

Card contact	Symbol	Fungtion
C1	VCC	Supply voltage
C2	RST	Control input (Reset Signle)
C3	CLK	Clock input
C4	GND	Ground
C5	NC	Not connected
C7	IO	Bi-directional data line (open drain)

สมาร์ตการ์ดในรุ่นแรก ๆ เป็นสมาร์ตการ์ดประเภท Free Access Memory สมาร์ตการ์ดแบบนี้สามารถทำการอ่านและเขียนข้อมูลใน Address ต่าง ๆ ได้โดยไม่ต้องมีรหัสผ่าน จุดนี้เป็นจุดอ่อนของสมาร์ตการ์ดรุ่นก่อน ๆ ในปัจจุบันนี้สมาร์ตการ์ดมีการเข้ารหัสที่ยากแก่การเข้าถึงข้อมูลโดยการ Hack

สมาร์ตการ์ดแบบที่ใช้ในบัตรโทรศัพท์ TOT ในประเทศไทยนั้นเป็น Memory Card ชนิด Token ภายในสมาร์ตการ์ดนี้จะมีการเก็บข้อมูลในลักษณะของ Counter ซึ่งเมื่อทำการใช้ไปหน่วยความจำก็จะถูกลดลงเรื่อย ๆ เมื่อใดที่หน่วยความจำถูกใช้หมดแล้วย่อมหมายถึงบัตรนั้นไม่สามารถใช้โทรออกได้แล้ว แต่ถึงบัตรจะถูกใช้หมดแล้วก็ตาม แต่ก็ยังสามารถอ่านข้อมูลจากบัตรได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5.2 Processor Card (Asynchronous Card)

สมาร์ตการ์ดแบบ Processor Card มีชื่อเรียกอีกอย่างหนึ่งว่า Asynchronous เพราะใช้การสื่อสารแบบ Asynchronous ในการติดต่อ สมาร์ตการ์ดชนิดนี้ได้เพิ่มหน่วยประมวลผล (Processor) เข้าไปด้วย ทำให้สมาร์ตการ์ดชนิดนี้มีทั้ง Memory และ Processor การที่ใส่ Processor ทำให้ต้องเพิ่มหน่วยความจำสำรองเข้าไป ทั้งให้ราคาของการ์ดชนิดนี้มีราคาสูงขึ้นด้วยซึ่งแปรผันตรงกันความสามารถในการทำงานของมัน หน้าสัมผัสของการ์ดชนิดนี้เหมือนกับแบบ Free Access Memory เพราะใช้มาตรฐาน ISO-7816 เหมือนกัน การเข้าถึงข้อมูลของการ์ดชนิดนี้ไม่สามารถทำได้เหมือน Free Access Memory การเข้าถึงข้อมูลต้องกระทำผ่านทาง Processor เท่านั้น เพราะหน่วยความจำจะอยู่ในความควบคุมของ Processor ถึงแม้จะมีการเข้าถึงข้อมูลที่ยากกว่า Free Acess Memory แต่ก็มีความปลอดภัยสูงจากการ Hack

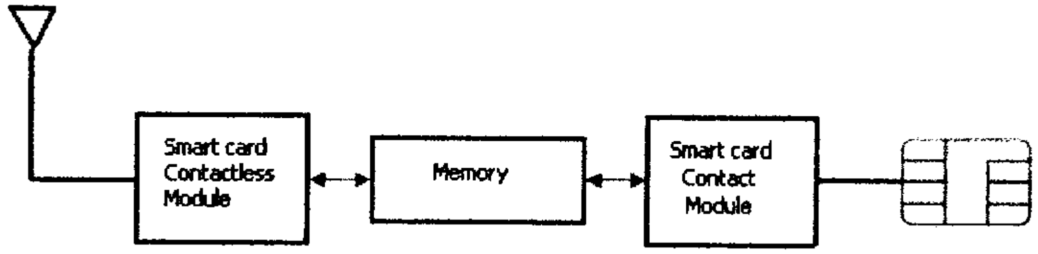
2.1.5.3 Contactless Card

สมาร์ตการ์ดแบบ Contactless ไม่ใช้หน้าสัมผัสในการเข้าถึงข้อมูล ระบบของสมาร์ตการ์ดแบบนี้เป็นระบบที่ทันสมัยที่สุดเลยก็ว่าได้ การสื่อสารกับสมาร์ตการ์ดชนิดนี้ใช้การสื่อสารโดยใช้คลื่นวิทยุ โดยการส่งความถี่ 13.56 MHz ไปยังสมาร์ตการ์ด ที่ตัวสมาร์ตการ์ดจะมีเสาอากาศที่เป็นขดลวดที่ได้รับการแมชชิ่งมาอย่างดีคอยรับสัญญาณ จะเห็นได้ว่าสมาร์ตการ์ดประเภทนี้จะแปลกกว่าชนิดอื่นตรงที่ว่าใช้กระแสไฟฟ้าที่มาจากคลื่นวิทยุทำงานเท่านั้น ดังนั้นการออกแบบสมาร์ตการ์ดแบบนี้จึงต้องออกแบบให้ใช้ไฟต่ำที่สุดเท่าที่จะน้อยได้ ไม่งั้นจะไม่เพียงพอในการทำงานของการ์ด ถ้ามองดูที่สมาร์ตการ์ดประเภทนี้แล้วเราไม่อาจบอกได้ว่าเป็นสมาร์ตการ์ดแบบ Contactless เพราะรูปร่างภายนอกเหมือนบัตรพลาสติกใบหนึ่ง สมาร์ตการ์ดแบบนี้ พบบ่อยในอาคารที่จอดรถ เพราะว่าสามารถอ่านข้อมูลได้อย่างรวดเร็ว หรือจะพบว่านิยมใช้เป็น Security Card

2.1.5.4 Com – Bl Card

สมาร์ตการ์ดชนิดนี้เป็นการรวมเอาสมาร์ตการ์ดแบบมีหน้าสัมผัสและไม่มีหน้าสัมผัสเข้าด้วยกัน โดยการใช้หน่วยความจำเข้าด้วยกันในการใช้งานที่ต้องการความปลอดภัยสูงก็จะใช้การเข้าถึงแบบมีหน้าสัมผัส โดยผ่านตัวโปรเซสเซอร์ ส่วนงานที่ต้องการความรวดเร็วสะดวกสบายก็จะใช้การสื่อสารทางคลื่นวิทยุโดยไม่มีการใช้หน้าสัมผัส

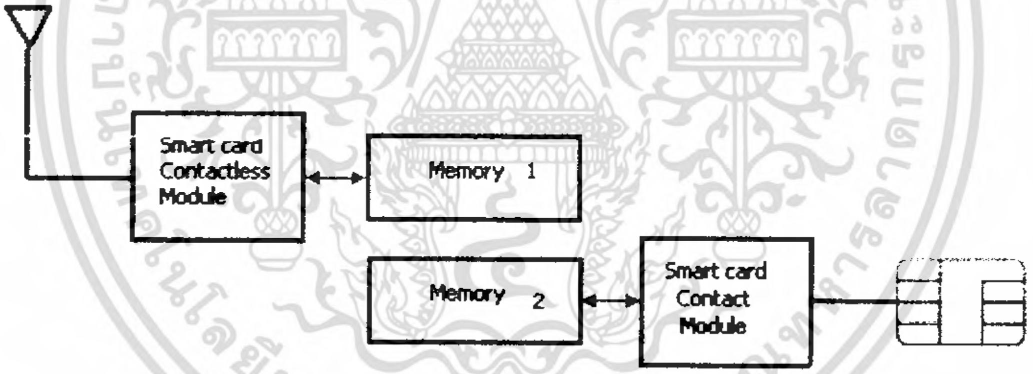
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 โครงสร้างภายในของสมาร์ตการ์ดชนิด Com - Bi Card

2.1.5.5 Hybrid Card

สมาร์ตการ์ดชนิดนี้มีลักษณะโครงสร้างเหมือนการ์ดประเภท Com - Bi Card แต่จะแตกต่างกันที่หน่วยความจำข้อมูล โดยหน่วยความจำระหว่างมีหน้าสัมผัสและไม่มีหน้าสัมผัสจะถูกแยกออกจากกันอย่างสิ้นเชิง เพื่อความสะดวกในการใช้งาน ในปัจจุบัน Hybrid Card จะมีความหมายรวมถึงบัตรที่มีคุณสมบัติในการใช้งานตั้งแต่สองอย่างขึ้นไปเช่น การ์ดที่มีทั้งแถบแม่เหล็กและชิปสมาร์ตการ์ด บัตรสมาร์ตการ์ดที่มีหน้าสัมผัสและไม่มีหน้าสัมผัส



รูปที่ 2.9 โครงสร้างภายในของสมาร์ตการ์ดชนิด Hybrid Card

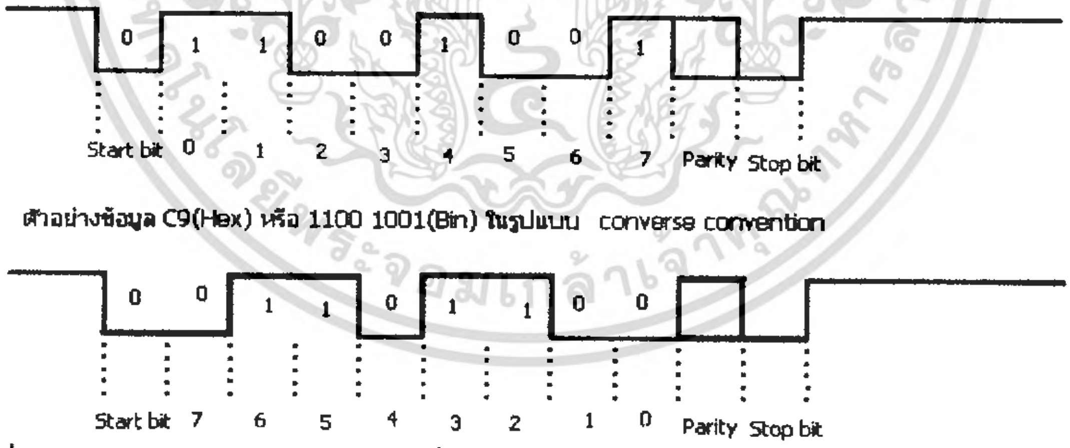
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การเชื่อมต่อชิปสมาร์ตการ์ด

(โปรแกรมติดต่อสื่อสารกับสมาร์ตการ์ดมีชื่อเรียกว่า Card Driver) โดย Card Driver สำหรับสมาร์ตการ์ดแต่ละชนิดจะมีลักษณะการทำงานใกล้เคียงกันจะต่างกันก็เพียงวิธีการสื่อสารระหว่างสมาร์ตการ์ดชนิด Memory กับ Processor เท่านั้นที่ไม่เหมือนกัน สำหรับการเลือกภาษาแอสแซมบลีมาใช้เขียนโปรแกรมก็ต้องเลือกใช้ให้ตรงกับตระกูลของโปรเซสเซอร์ที่ใช้ ส่วนโปรแกรมที่ใช้สำหรับบริหารข้อมูล และติดต่อกับผู้ใช้งานในเทอร์มินอล สามารถเขียนด้วยภาษาระดับสูงอย่างภาษาซี เหตุที่เลือกภาษาซีเพราะภาษาซีมีความสามารถในการประมวลผลที่รวดเร็ว สามารถจัดการกับข้อมูลในระดับได้ดีกว่าภาษาระดับสูงภาษาอื่นและทำการพัฒนาโปรแกรมได้ง่ายกว่า โดยทั่วไปภาษาซีที่ใช้ในการเขียนเป็นโปรแกรมสำหรับเทอร์มินอลจะต้องชุดพัฒนาเฉพาะตระกูลโปรเซสเซอร์ที่ใช้งาน เช่นเดียวกับภาษาแอสแซมบลี

การสร้าง Card Driver ด้วยภาษาระดับต่ำอย่างภาษาแอสแซมบลีอาจเป็นเรื่องยุ่งยากเกินความจำเป็นนักพัฒนาระบบอาจนำ IC สำหรับเชื่อมต่อสมาร์ตการ์ดมาสร้างเป็นวงจรสำหรับเชื่อมต่อสมาร์ตการ์ดได้เช่นกัน IC ที่ใช้มีด้วยกันหลายเบอร์ หลายผู้ผลิตที่น่าสนใจเช่น IC เบอร์ TDA8002 , TDA8003 , TDA8004 , TDA8006 , TDA8007 , และ TDA8008 ของบริษัท Phillips , MC33560 ของบริษัท ON Semiconductor และ ML69240 ของบริษัท OKI Semiconductor ฯลฯ แต่การสั่งซื้ออาจทำได้ยากเนื่องจากต้องสั่งซื้อจากต่างประเทศคราวละจำนวนมากๆ (ต้องสั่งซื้อ 2000 ตัวต่อครั้งเป็นอย่างน้อย) เนื่องจาก IC สำหรับเชื่อมต่อสมาร์ตการ์ดเหล่านี้ถูกผลิตขึ้นเพื่อส่งให้ผู้ผลิตสมาร์ตการ์ดรีดเดอร์เท่านั้น แต่ปัจจุบันเริ่มมีผู้ประกอบการจำหน่ายชิ้นส่วนอิเล็กทรอนิกส์บางรายได้สั่ง IC สำหรับเชื่อมต่อสมาร์ตการ์ดเข้ามาขายในประเทศบ้าง

2.1.7 การสื่อสารกับชิปสมาร์ตการ์ดในระดับสัญญาณไฟฟ้า

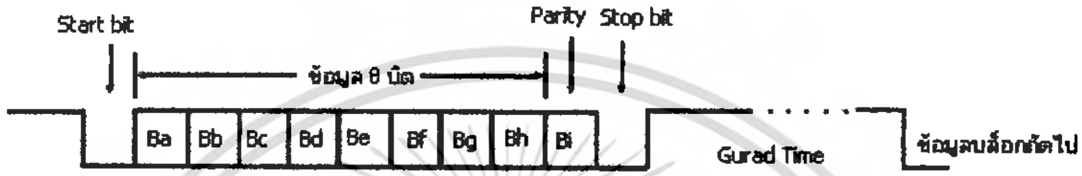


รูปที่ 2.10 ตัวอย่างรูปแบบสัญญาณของการสื่อสารแบบซิงเกิลบัสแบบ Direct convention และ Converse convention

การสื่อสารแบบซิงเกิลบัสแบบ Direct convention และ Converse convention มีความแตกต่างกันในเรื่องของลอจิกที่ใช้แทนความหมายของบิตข้อมูล โดยแบบ Direct convention จะมีการแทนลอจิก 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนค่าบิตข้อมูลที่เป็น 0 และแทนลอจิก 1 แทนค่าบิตข้อมูลที่เป็น 1 และบิตข้อมูลที่จะถูกส่งมาเป็นบิตแรกคือบิตที่มีค่านัยสำคัญต่ำ (LSB – Least Significant Bit) ส่วนแบบ Converse convention จะมีการแทนลอจิก 0 แทนค่าบิตข้อมูลที่เป็น 1 และแทนลอจิก 1 แทนค่าบิตข้อมูลที่เป็น 0 และบิตข้อมูลที่จะถูกส่งมาเป็นบิตแรกคือบิตที่มีค่านัยสำคัญสูง (MSB – Most Significant Bit) ซึ่งจะกลับค่ากับแบบแรก ส่วนมากแล้วจะนิยมใช้แบบ Direct convention มากกว่าแบบ Converse convention แต่ทางที่ดีควรสร้าง Card driver ให้สามารถตอบสนองการสื่อสารได้ทั้งสองแบบ



รูปที่ 2.11 ตัวอย่างข้อมูลที่ใช้ในการรับ-ส่งข้อมูลกับสมาร์ตการ์ด

2.1.8 การรีเซตชิปสมาร์ตการ์ด (Card Present)

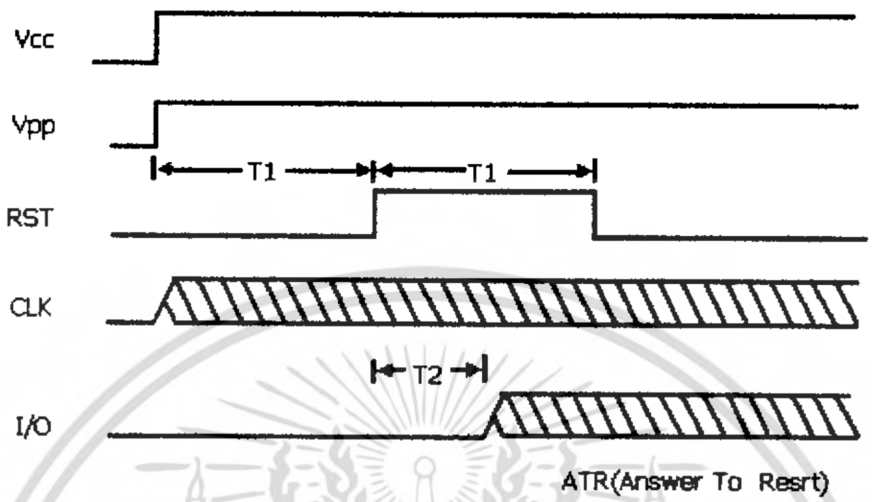
ขั้นตอนการรีเซตชิปสมาร์ตการ์ดเป็นขั้นตอนที่ต้องทำทุกครั้งเมื่อตรวจสอบพบว่ามีการเสียบบัตรเข้ามาในคอนแทกต์รีดเดอร์ เพื่อให้สมาร์ตการ์ดพร้อมทำงานหลังจากได้รับกระแสไฟฟ้าเลี้ยงวงจรภายใน (Vcc) และเป็นการเริ่มต้นการสื่อสารกับอุปกรณ์ที่มาเชื่อมต่อ โดยในที่นี้จะเรียกอุปกรณ์เชื่อมต่อว่าเทอร์มินอล ในสภาวะว่าง (Idle) ของขาเชื่อมต่อหน้าสัมผัสชิปสมาร์ตการ์ด (ยกเว้นกราวนด์) ขณะที่ยังไม่มีสมาร์ตการ์ดเสียบเข้ามาจะค่าเท่ากับแรงดันไฟฟ้าของเทอร์มินอลเสมอ

เมื่อตรวจสอบพบว่ามีการเสียบบัตรเข้ามาในคอนแทกต์รีดเดอร์ โดยตรวจสอบได้จากสวิทช์ตรวจสอบการเสียบบัตรภายในคอนแทกต์รีดเดอร์อันดับแรกต้องทำการตรวจสอบว่าแรงดันไฟฟ้าที่หน้าสัมผัส Vcc ลดลงหรือไม่ หากแรงดันไฟฟ้าลดต่ำลงให้ทำการตัดวงจร (Open circuit) หรือลอยวงจร (Float) โดยการเพิ่มอิมพีแดนซ์ของวงจร ทำให้วงจรจะถูกแยกออกจากหน้าสัมผัสทันทีเพราะสิ่งที่เสียบเข้ามาอาจเป็นโลหะหรือสมาร์ตการ์ดที่การลัดวงจรภายในชิป หากไม่ตัดวงจรออกอาจทำให้วงจรสำหรับเชื่อมต่อสมาร์ตการ์ดของเทอร์มินอลเสียหายได้

หลังจากที่จ่ายกระแสไฟฟ้าและตรวจสอบวงจรไฟฟ้าของ Vcc ว่าคงที่แล้ว ให้ทำการเปลี่ยนระดับสัญญาณ RST เป็นลอจิก 0 ในเป็นระยะเวลาไม่น้อยกว่า 400 ไซเกิลสัญญาณนาฬิกาหรือ 112 ไมโครวินาที แต่ไม่เกิน 40,000 ไซเกิลสัญญาณนาฬิกาหรือ 11.2 มิลลิวินาทีแล้วจึงค่อยเปลี่ยนสัญญาณ RST เป็นลอจิก 1 ซึ่งชิปสมาร์ตการ์ดจะทำการรีเซตตัวเองที่ขอบขาขึ้นของสัญญาณ RST การรีเซตแบบนี้เรียกว่า Cold Reset การรีเซตด้วยวิธีนี้จะทำให้ชิปสมาร์ตการ์ดทำการเคลียร์ข้อมูลภายในหน่วยความจำชั่วคราวต่างๆให้อยู่ในสภาวะเริ่มต้นทั้งหมด ซึ่งแตกต่างจากการรีเซตด้วยคำสั่งของสมาร์ตการ์ด (คำสั่ง Card Present) หรือที่เรียกว่า Warm Reset การรีเซตวิธีนี้จะทำโดยการเปลี่ยนแปลง RST แบบเดียวกับ Cold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Reset แต่ไม่มีการตัดกระแสไฟฟ้าที่จ่ายให้แก่ชิป การรีเซตแบบนี้จะทำให้ชิปสมาร์ตการ์ดกลับไปสู่สภาวะเริ่มต้นแต่ข้อมูลภายในหน่วยความจำชั่วคราวต่างๆจะยังคงอยู่เช่นเดิมในสถานะก่อนที่จะรีเซต



รูปที่ 2.12 ไทม์มิ่งไดอะแกรมการรีเซตชิปสมาร์ตการ์ด

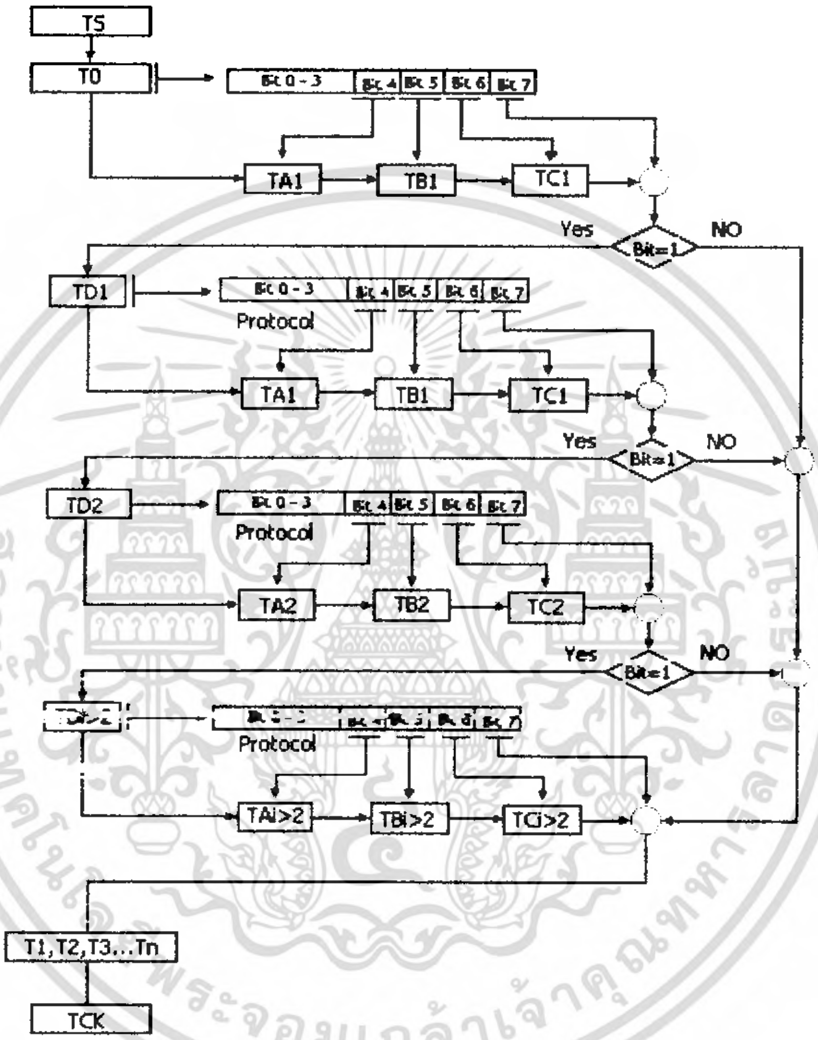
จากไทม์มิ่งไดอะแกรมในรูปที่ 2.12 จะเห็นได้ว่า หลังจากที่ทำการจ่ายกระแสไฟให้กับชิปสมาร์ตการ์ด สัญญาณรีเซตจะต้องเปลี่ยนเป็นลอจิก 0 เป็นระยะเวลาไม่ต่ำกว่า 400 ไซเกิลสัญญาณนาฬิกา และไม่เกิน 40,000 ไซเกิล สัญญาณนาฬิกา (T1) จากนั้นทำการเปลี่ยนเป็นลอจิก 1 ซึ่งชิปสมาร์ตการ์ดจะทำการรีเซตที่ขอบขาขึ้นของสัญญาณรีเซต

หลังจากที่ชิปสมาร์ตการ์ดทำการรีเซตตัวเองเรียบร้อยแล้ว ชิปสมาร์ตการ์ดจะทำการส่งข้อมูล ART (Answer To Reset) หรือสัญญาณตอบรับการรีเซต เพื่อบอกชนิดของโปรโตคอลสำหรับการสื่อสาร เนื่องจากเราไม่มีทางทราบได้ว่าสมาร์ตการ์ดที่เสียบเข้ามานั้นเป็นสมาร์ตการ์ดชนิดใด สำหรับอัตราการส่งข้อมูล ART จะใช้อัตราส่วนเท่ากับ CLK/372 (Asynchronous) หรือ 9600 บิตต่อวินาที (Synchronous) ซึ่งความถี่ของสัญญาณนาฬิกาของสมาร์ตการ์ดชนิด Memory เกิดจากการเปลี่ยนลอจิก 0 และ 1 ไปมาต่อเนื่องกันโดยไม่ต้องคำนึงถึงเรื่องของระยะเวลาเพราะการรับ-ส่งข้อมูลในแต่ละบิตจะสัมพันธ์กับสัญญาณนาฬิกาโดยตรง แต่ถ้าเป็นสมาร์ตการ์ดชนิด Processor ข้อมูลที่รับ-ส่งไม่ต้องสัมพันธ์กับสัญญาณนาฬิกา เพียงกำหนดอัตราการรับ-ส่งข้อมูลเป็น 9600 บิตต่อวินาที ก็จะสามารถติดต่อสื่อสารได้แล้ว ส่วนสัญญาณนาฬิกาก็ป้อนให้ด้วยความถี่ 3.5712 เมกะเฮิรตซ์ต่อเนื่องกัน (Continuos Clock) ดังนั้นในตอนแรกที่รอรับข้อมูลจากสมาร์ตการ์ดให้เตรียมการสื่อสารแบบ Asynchronous ไว้เป็นแบบแรกก่อน หากรับข้อมูลไม่ได้หรือรับข้อมูลได้แต่นำข้อมูลมาตีความหมายได้ไม่ถูกต้องให้ทำการรีเซตสมาร์ตการ์ด แล้วเปลี่ยนไปใช้การสื่อสารแบบ Asynchronous แล้วรอรับข้อมูลมาตรวจสอบดูอีกครั้ง ถ้ายังไม่สามารถรับข้อมูลได้หรือตีความหมายของข้อมูลไม่ได้อีก ให้ถือว่าไม่สามารถใช้งานสมาร์ตการ์ดใบนั้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9 ส่วนประกอบของข้อมูล ATR (Answer To Reset)

หลังจากที่ทำการรีเซตด้วยขอบขาขึ้นของสัญญาณ RST ในชั่วระยะเวลาหนึ่ง (ตามสเปกของสมาร์ตการ์ดแต่ละรุ่น) ชิปสมาร์ตการ์ดจะทำการส่งข้อมูล ATR กลับออกมา ซึ่งข้อมูล ATR เปรียบเสมือนข้อมูลตัวอย่างสำหรับช่วยในการค้นหาโปรโตคอล ที่ใช้ในการสื่อสารกับสมาร์ตการ์ด



รูปที่ 2.13 โครงสร้างและส่วนประกอบของข้อมูล ATR

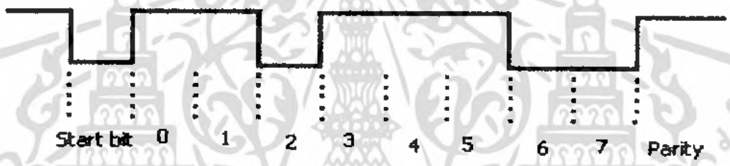
ใบนั้นๆเนื่องจากในตอนแรกเราจะไม่มีทางทราบได้เลยว่าสมาร์ตการ์ดที่นำมาใช้งานเป็นชนิดใดและใช้โปรโตคอลแบบใดในการติดต่อสื่อสาร ดังนั้นจึงต้องการวิเคราะห์ข้อมูล ATR เพื่อให้ทราบรายละเอียดเกี่ยวกับการติดต่อสื่อสารของสมาร์ตการ์ดใบนั้นๆเป็นขั้นตอนแรกๆของการสื่อสาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่ ATR ที่ถูกส่งออกมาจะมีลักษณะที่เป็นโซ่ลูก (Link chain) นั้นก็คือ ข้อมูลจะถูกแบ่งออกเป็นกลุ่มๆ โดยแต่ละกลุ่มจะมีข้อมูลควบคุม (Control byte) เป็นข้อมูลไบต์แรกของกลุ่ม (To,TDx) ข้อมูลควบคุมนี้จะเป็นตัวบอกให้ทราบถึงจำนวนข้อมูลสมาชิกในกลุ่ม(Tax,TBx,TCx) ที่จะถูกส่งตามมาโดยข้อมูลสมาชิกไม่จำเป็นต้องถูกส่งออกมาครบทุกตัว และบิตที่ 7 ของข้อมูลควบคุมจะเป็นตัวบอกให้ทราบว่า จะมีการส่งข้อมูลกลุ่มถัดไปอีกหรือไม่ หากไม่มีก็จะถูกปิดท้ายด้วยกลุ่มข้อมูลปิดท้าย (Hitorical Character-T1-T2-T3,….Tn) และ TCK (Check Character) ดังรูปที่ 2.13

2.1.9.1 Initial character (TS)

Initial character เป็นข้อมูลขนาด 1 ไบต์ที่จะบอกให้ทราบว่ารูปแบบการส่งข้อมูลที่จะเกิดขึ้นในไบต์ถัดๆไปเป็นแบบ Direct convention หรือ Inverse convention โดยที่ Initial character นี้จะมีรูปแบบที่ตรงกันข้าม ข้อมูลที่ส่งมานั้นจะมีค่าเป็น 3B สำหรับแทนการสื่อสารแบบ Direct convention และ 3F สำหรับแทนการสื่อสารแบบ Converse convention ดังรูปที่ 2.14



ข้อมูล Initial character (TS) ค่า 3B(hex) ในรูปแบบ Direct convention

ข้อมูล Initial character (TS) ค่า 3F(hex) ในรูปแบบ Converse convention

รูปที่ 2.14 ข้อมูล Initial character ในรูปแบบของ Direct convention และ Converse convention

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.1.9.2 Format character (TO)

Format character เป็นข้อมูลที่ใช้สำหรับบอกจำนวนข้อมูลที่จะตามมาและจำนวนไบต์ของ Historical character(T1,T2,t3,....Tn) ซึ่งข้อมูล Format character จะสามารถพบได้จากสมาร์ตการ์ดทุกใบที่มีการส่งข้อมูล ATR สำหรับความหมายของ Format character สามารถอธิบายได้ด้วยตารางที่ 2.2

ตารางที่ 2.2 ตารางแสดงความหมายของ Format character

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
-	#####	-	-	x	x	x	x	จำนวนข้อมูล Historical charachter (T1, T2, T3 ... Tn)
-	-	-	1	-	-	-	-	บิตแสดงการส่งข้อมูล TA1
-	-	1	-	-	-	-	-	บิตแสดงการส่งข้อมูล TB1
-	1	-	-	-	-	-	-	บิตแสดงการส่งข้อมูล TC1
1	-	-	-	-	-	-	-	บิตแสดงการส่งข้อมูล TD1

2.1.9.3 Interface characters (TA,TB,TC,TD)

Interface characters เป็นข้อมูลที่ส่งต่อจากข้อมูล TO ซึ่งประกอบด้วยข้อมูลหลายไบต์ แต่ละไบต์มีความหมายดังนี้

TA1 – เป็นข้อมูลที่ใช้แสดงแทนอัตราหารความถี่สัญญาณนาฬิกา (ค่า FI) ซึ่งอยู่ในข้อมูล 4 บิตบนของข้อมูล TA1 และ Adjustment factor (DI) ซึ่งอยู่ในข้อมูล 4 บิตล่างของข้อมูล TA1

ตารางที่ 2.3 อัตราหารความถี่สัญญาณนาฬิกา

F	Internal CLK	372	558	744	1116	1488	1860	RFU
FI	0000	0001	0010	0011	0100	0101	0110	0111
fs	0	5	6	8	12	16	20	-
F	RFU	512	768	1024	1536	2048	RFU	RFU
FI	1000	1001	1010	1011	1100	1101	1110	1111
fs	-	5	7.5	10	15	20	-	-

72754

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 Adjustment factor

D	RFU	1	2	4	8	16	RFU	RFU
DI	0000	0001	0010	0011	0100	0101	0110	0111
DI	RFU	1/2	1/4	1/8	1/16	1/32	1/64	-
DI	1000	1001	1010	1011	1100	1101	1110	-

ข้อมูลอัตราหารความถี่สัญญาณนาฬิกา และ Adjustment factor เป็นข้อมูลที่ช่วยในการคำนวณค่าคุณสมบัติของสัญญาณต่างๆที่ต้องใช้ในชิปสมาร์ตการ์ดจากสูตรคำนวณดังนี้

สำหรับสัญญาณนาฬิกาภายใน

Initial etu = 1/9600 Sec Work etu = (1/D) * (1/9600)

สำหรับสัญญาณนาฬิกาภายนอก

Initial etu = 372/fi Sec Work etu = (1/D) * (F/fs)

etu (Elementary Time Unit) – ระยะความกว้างของสัญญาณข้อมูลในแต่ละบิต

Initial etu – ระยะความกว้างระหว่างบิตของข้อมูล ATR และ PTS

Work etu – ระยะความกว้างระหว่างบิตของโปรโตคอลที่ตามหลังข้อมูล ATR และ PTS ซึ่งไม่ขึ้นตรงต่อ ATR

fi - ความถี่ที่ป้อนให้แก่ชิปสมาร์ตการ์ด

fs – ความถี่สูงสุดที่ชิปสมาร์ตการ์ดสามารถทำงานได้ มีหน่วยเป็นเมกะเฮิรตซ์

TB1 – เป็นข้อมูลที่ใช้แสดงรูปแบบของแรงดันไฟฟ้าสำหรับเขียนข้อมูล (Vpp) ข้อมูลตัวนี้มีการใช้งานในสมาร์ตการ์ดที่มีหน่วยความจำชนิด EEPROM ซึ่งสามารถพบได้ในสมาร์ตการ์ดรุ่นใหม่ๆ ภายในมีการจัดการข้อมูลออกเป็นสองส่วนคือ PI1 และ PI2 โดยค่าทั้งสองจะเป็น 0 สำหรับสมาร์ตการ์ดชนิด Processor

ตารางที่ 2.5 รูปแบบข้อมูล TB1

บิตข้อมูล								IFSC
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Information Field size of the Card)
0	X	-	-	-	-	-	-	II
-	-	X	X	X	X	X	X	PI1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TC1 – เป็นข้อมูลที่ใช้แสดงถึงส่วนเพิ่มเติมของ Guard time (ค่า N) ใช้สำหรับกำหนดจำนวนระยะความกว้างของสัญญาณที่เพิ่มขึ้นมาซึ่งมีค่าตั้งแต่ 0 ถึง 254 ส่วนค่า 255 จะใช้สำหรับกำหนดค่า N และจะสัมพันธ์กับโปรโตคอลที่ใช้ด้วย

ตารางที่ 2.6 รูปแบบข้อมูล TC1

บิตข้อมูล								IFSC
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	(Information Field size of the Card)
X=255, T=0								N=12 etu
X=255, T=1								N=11 etu

TDi (I ลำดับที่ของ TD) – เป็นข้อมูลที่ใช้สำหรับบอกจำนวนข้อมูลที่จะตามมาเป็นจำนวนไบต์ และชนิดของโปรโตคอล (ในกรณีที่เป็นสมาร์ดการ์ดชนิด Processor) ในกรณีที่ไม่มีการส่งข้อมูล TD ออกมาให้สันนิษฐานว่ามีโปรโตคอล T=0 เอาไว้ก่อน (เมื่อใช้วิธีสื่อสารแบบ Asynchronous)

ตารางที่ 2.7 รูปแบบของข้อมูล TDi

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
-	-	-	-	x	x	x	x	โปรโตคอลที่ใช้ (0-15)
-	-	-	1	-	-	-	-	บิตแสดงการส่งข้อมูล TAi+1
-	-	1	-	-	-	-	-	บิตแสดงการส่งข้อมูล TBi+1
-	1	-	-	-	-	-	-	บิตแสดงการส่งข้อมูล TCi+1
1	-	-	-	-	-	-	-	บิตแสดงการส่งข้อมูล TDi+1

TA2 – เป็นข้อมูลที่ใช้สำหรับบอกให้ทราบถึง โหมดที่จะใช้ในการเลือกโปรโตคอล (PTS – Protocol Type Selection) ซึ่งจะกล่าวถึงในหัวข้อถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.8 รูปแบบของข้อมูล TA2

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit3	Bit2	Bit1	Bit0	
0	-	-	-	-				สามารถสลับโหมด Negoriable mode กับ Specific mode ได้
1	-	-	-	-				ไม่สามารถสลับโหมด Negoriable mode กับ Specific mode ได้
-	0	0	-	-				สงวนไว้ใช้ในภายหลัง
-	-	-	0	-				โปรโตคอลสำหรับรับ-ส่งข้อมูลมีการกำหนดอย่างชัดเจนใน Interface character
-	-	-	1	-				โปรโตคอลสำหรับรับ-ส่งข้อมูลมีการกำหนดอย่างคร่าวๆ ใน Interface character
-	-	-	-	X				หมายเลขโปรโตคอลที่ใช้

TB2 – เป็นข้อมูลที่ใช้สำหรับบอกให้ทราบถึงปริมาณแรงดันไฟฟ้าของ Vpp ซึ่งข้อมูลภายในจะมีความหมายเป็นเลขฐาน 10

ตารางที่ 2.9 รูปแบบของข้อมูล TB2

บิตข้อมูล								IFSC (Information Field size of the Card)
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
X								PI2

TC2 – เป็นข้อมูลที่ใช้สำหรับบอกให้ทราบถึงค่าของ work waiting time ซึ่งเป็นระยะห่างสูงสุดของขอบขาขึ้นของสัญญาณสองสัญญาณ หากไม่มีการส่ง TC2 จะถือว่า work waiting time มีค่าเป็น 10 โดยปริยาย

work waiting time = (960 * D * WI) work etu

ตารางที่ 2.10 รูปแบบของข้อมูล TD

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
X								WI

***work etu - มีค่าเท่ากับ 1/9600 สำหรับสัญญาณนาฬิกาที่ 3.5712 เมกะเฮิรตซ์ หรือ $(fi/F)^{-1}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TAi (i>2) – เป็นข้อมูลที่ใช้สำหรับบอกให้ทราบถึงขนาดของบัฟเฟอร์ที่ชิปสมาร์ตการ์ดจะสามารถรับข้อมูลได้ โดยค่าที่เป็นไปได้จะอยู่ระหว่าง 1 ถึง 254 และค่าเริ่มต้นจะอยู่ที่ 32 ไบต์

ตารางที่ 2.11 รูปแบบของข้อมูล TAi เมื่อ i > 2

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
X								IFSC

TBi (I > 2) – เป็นข้อมูลที่ใช้สำหรับบอกให้ทราบถึงค่า CWT (character wait time) และ BWT (Block Wait Time) ซึ่งสามารถหาได้จากสูตร

$CWT = (2^{CWI} + 11)$ work etu

$BWT = 2 * 960 * \frac{372}{f} s + 11$ work etu

f – ความถี่ของสัญญาณนาฬิกา

ตารางที่ 2.12 รูปแบบของข้อมูล TBi เมื่อ i > 2

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
-				-	-	-	-	CWI
x				X	X	X	X	BWI

*** CWT – ระยะห่างของการส่งข้อมูลในแต่ละบล็อกข้อมูล(บล็อกข้อมูลประกอบด้วยสตาร์ตบิต, บิตข้อมูล, พาริตี้, และสต็อปบิต) ที่เกิดจากการส่งข้อมูลที่ต่อเนื่องกันโดยนับจากจุดเริ่มต้นของสตาร์ตบิตในข้อมูลบล็อกแรกถึงจุดเริ่มต้นขอสตาร์ตบิตในข้อมูลบล็อกถัดไป

*** BWT – ระยะห่างของข้อมูลที่ตอบกลับจากชิปสมาร์ตการ์ด กับข้อมูลหรือคำสั่งที่ส่งให้แก่ชิปสมาร์ตการ์ด ซึ่งช่วงเวลาดังกล่าวเป็นเวลาที่ชิปสมาร์ตการ์ดกำลังประมวลผลและเตรียมส่งข้อมูลกลับออกมา โดยนับจากจุดเริ่มต้นของสตาร์ตบิตในข้อมูลบล็อกแรกถึงจุดเริ่มต้นของสตาร์ตบิตในข้อมูลที่ตอบกลับจากชิปสมาร์ตการ์ด

TCi (i > 2) – เป็นข้อมูลที่ใช้สำหรับบอกให้ทราบถึงวิธีที่ใช้ในการคำนวณค่าตรวจสอบความผิดพลาดของการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.13 รูปแบบของข้อมูล Tci เมื่อ i > 2

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
-	-	-	-	-	-	-	0	LCR
-	-	-	-	-	-	-	1	CRC
0	0	0	0	0	0	0	0	สงวนไว้ใช้ในภายหลัง

2.1.9.4 Historical character (T1, T2 , T3, ... Tn)

ข้อมูล Historical character ไม่ได้ถูกกำหนดให้เป็นข้อมูลมาตรฐานสำหรับสมาร์ตการ์ดทุกชนิด ซึ่งสมาร์ตการ์ดบางรุ่นส่งเฉพาะข้อมูลชื่อรหัสรุ่น และจำนวนหน่วยความจำของชิปสมาร์ตการ์ดออกมา บางรุ่นส่งเฉพาะข้อมูลหมายเลขประจำชิปออกมา (Card ID) ซึ่งไม่ซ้ำกับสมาร์ตการ์ดรุ่นเดียวกันในสายการผลิตเดียวกัน แต่บางรุ่นก็ไม่ส่งอะไรออกมาเลย ฯลฯ ดังนั้นข้อมูลในส่วนนี้จึงไม่ค่อยมีประโยชน์ต่อการติดต่อสื่อสารกับชิปสมาร์ตการ์ดมากนัก

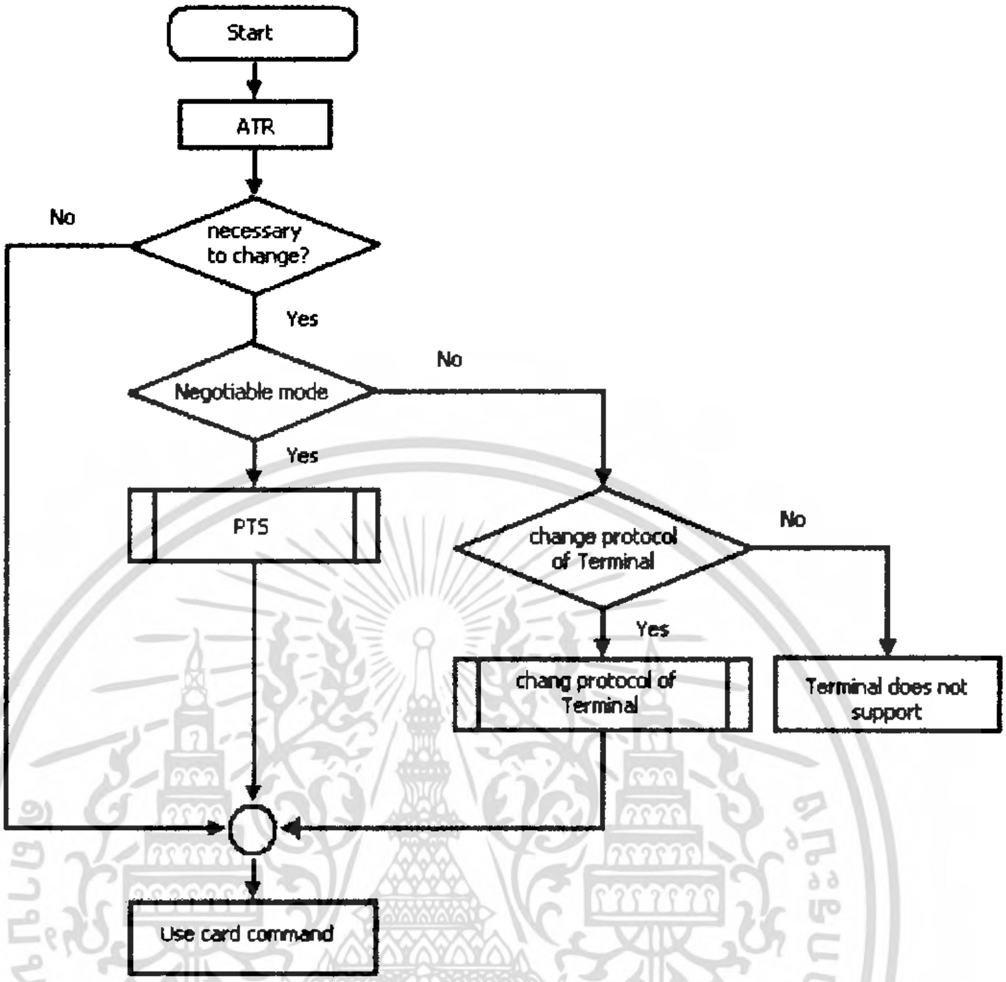
2.1.9.5 Check character (TCK)

Check character เป็นข้อมูลที่ได้จากการ X-OR ข้อมูล ATR ตั้งแต่ Format character(TO) จนถึงข้อมูลไบต์สุดท้าย ซึ่งสามารถตรวจสอบถูกต้องของข้อมูล ATR ที่ได้รับจากชิปสมาร์ตการ์ดได้อย่างง่ายๆ แต่ในบางครั้ง Check character อาจไม่มีการส่งกลับออกมาในบางโปรโตคอล

2.1.10 การเลือกชนิดโปรโตคอลสำหรับติดต่อกับสมาร์ตการ์ด (PTS – Protocol Type Selection)

สมาร์ตการ์ดรุ่นใหม่ๆในปัจจุบัน มีความสามารถในการติดต่อสื่อสารหลายรูปแบบในตัวเองการที่เทอร์มินอลต้องทำการติดต่อสื่อสารกับสมาร์ตการ์ดโดยตรงนั้น จำเป็นต้องเลือกโปรโตคอลที่ใช้สื่อสารให้ตรงกับ เนื่องจากสมาร์ตการ์ดแต่ละรุ่นจะมีการกำหนดโปรโตคอลเริ่มต้นไว้อยู่แล้วหากเทอร์มินอลติดต่อกับชิปสมาร์ตการ์ดโดยไม่ตรวจสอบ และเปลี่ยนโปรโตคอลของสมาร์ตการ์ดจะทำให้ไม่สามารถติดต่อสื่อสารกันได้ หรือถ้าสามารถทำได้ก็จะทำให้ข้อมูลที่รับ-ส่งมีเปอร์เซ็นต์ผิดพลาดค่อนข้างมาก แต่ถ้าตรวจสอบแล้วพบว่ามีโปรโตคอลที่ตรงกันก็สามารถข้ามขั้นตอนการเลือกโปรโตคอลและใช้งานสมาร์ตการ์ดได้ทันที

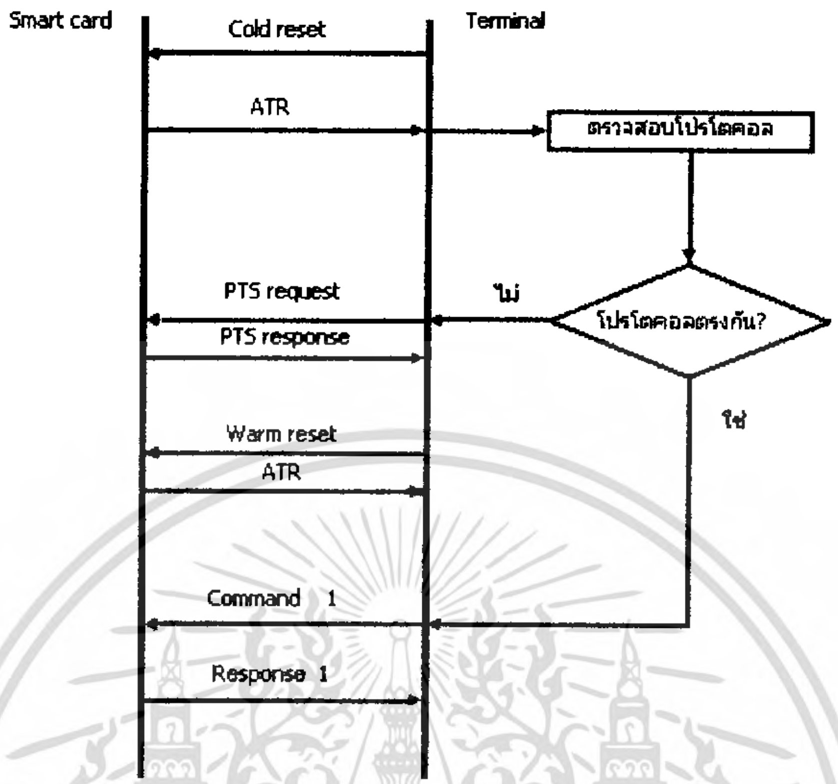
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 ขั้นตอนตรวจสอบการเปลี่ยนโปรโตคอลของสมาร์ตการ์ด

การเลือกชนิดโปรโตคอลสำหรับติดต่อกับสมาร์ตการ์ด เป็นขั้นตอนที่สลับซับซ้อนที่ต้องอาศัยความเข้าในเรื่องของข้อมูล ATR (Answer To Reset) ที่สมาร์ตการ์ดส่งกลับมา และการรีเซตชิปสมาร์ตการ์ดเป็นพิเศษ ซึ่งการเลือกโปรโตคอลนั้นจะทำโดยการเปลี่ยนแปลงค่าตัวแปลงบางตัวเช่นFI, DI, Guard time(ค่า N), และหมายเลขโปรโตคอลของสมาร์ตการ์ดมีการทำงานในสองโหมดคือ Negotiable mode mode และ Specification mode เราจะทราบถึงโหมดการติดต่อกับสมาร์ตการ์ดได้จากข้อมูล TA2 ของ ATR (ตารางที่ 2.8)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

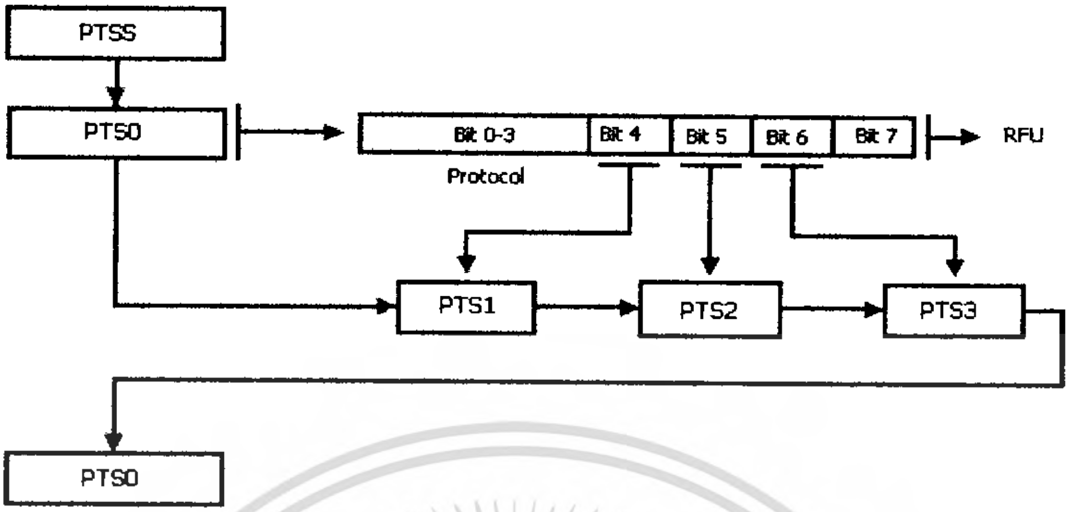


รูปที่ 2.16 ขั้นตอนการเปลี่ยนโปรโตคอลของสมาร์ตการ์ด

ในการส่งข้อมูล PTS เพื่อขอเปลี่ยนโปรโตคอลจะต้องทำทันทีหลังจากที่ได้รับข้อมูล ATR หลังจากทำการส่งข้อมูล PTS ไปแล้ว หากเป็นสมาร์ตการ์ดชนิดที่สามารถเปลี่ยนแปลงโปรโตคอลได้ สมาร์ตการ์ดจะทำการส่งข้อมูล PTS คืนมาให้ แต่ถ้าไม่มีข้อมูลใดๆส่งกลับมาก็เป็นไปได้ว่าสมาร์ตการ์ดชนิดนั้นไม่สามารถเปลี่ยนแปลงโปรโตคอลได้ หากต้องการเปลี่ยนโปรโตคอลอีกก็สามารถทำซ้ำได้โดยทำการรีเซตสมาร์ตการ์ดและรอรับข้อมูล ATR แล้วจึงทำการส่งข้อมูล PTS ใหม่

ข้อมูล PTS ที่จะใช้ส่งให้กับสมาร์ตการ์ดเพื่อเปลี่ยนโปรโตคอลจะมีลักษณะเป็นลูกโซ่คล้ายกับข้อมูล ATR แต่จะมีความยาวที่น้อยกว่า ซึ่งมีรายละเอียดดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 โครงสร้างข้อมูล PTS

จากรูปที่ 2.17จะเห็นได้ว่าข้อมูล PTS มีความคล้ายคลึงกับข้อมูล ATR ซึ่งข้อมูล PTS จะมีข้อมูล PTSS เป็นข้อมูลนำ ข้อมูล PTS0 เป็นข้อมูลของหมายเลขโปรโตคอล และข้อมูล PTS1, PTS2, PTS3 ซึ่งเป็นพารามิเตอร์ของโปรโตคอล โดยข้อมูล PTSS เป็นข้อมูลที่ไม่ค่อยมีการเปิดเผยมากนัก แต่มักจะใช้เป็นค่า FF (Hex) ส่วนข้อมูล PTS0, PTS1, PTS2 เป็นข้อมูลของโปรโตคอลที่ต้องการ สำหรับข้อมูล PTS3 ยังไม่พบว่ามีการใช้งาน

ตารางที่ 2.14 รูปแบบของข้อมูล PTS0

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
-	-	-	-		X			หมายเลขโปรโตคอลที่ใช้งาน
-	-	-	1					บิตแสดงการส่งข้อมูล PTS1
-	-	1	-					บิตแสดงการส่งข้อมูล PTS2
-	1	-	-					บิตแสดงการส่งข้อมูล PTS3
0	-	-	-					สงวนไว้ใช้ในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.15 รูปแบบของข้อมูล PTS1

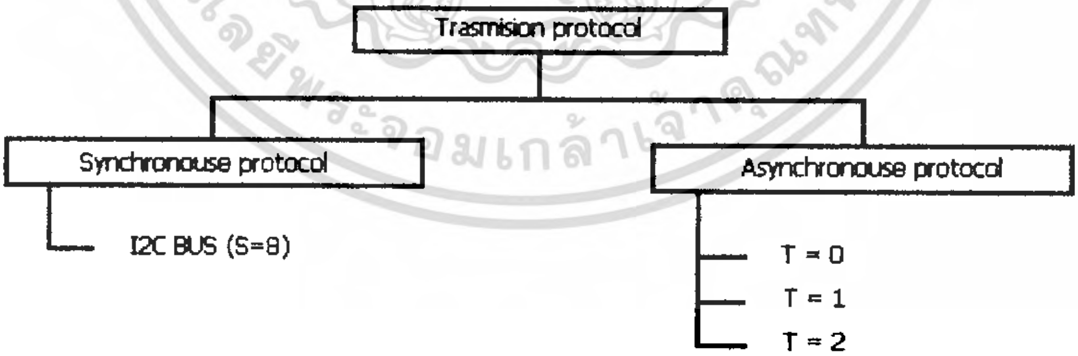
บิตข้อมูล								IFSC (Information Field size of the Card)
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
X	-							FI
-	X							DI

ตารางที่ 2.16 รูปแบบของข้อมูล PTS2

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
-	-	-	-	-	-	0	0	ไม่มีการแก้ไข Guard time
						0	1	N = 255
-	-	-	-	-	-	1	0	เพิ่ม Guard time 12 etu
X	X	X	X	X	X	-	-	สงวนไว้ใช้ภายหลัง

2.1.11 โปรโตคอลที่ใช้ในการสื่อสารกับสมาร์ตการ์ด

การสื่อสารกับชิปสมาร์ตการ์ด จำเป็นต้องใช้โปรโตคอลในการสื่อสารที่แตกต่างกันตามชนิดของสมาร์ตการ์ด ดังที่ทราบไปแล้วสมาร์ตการ์ดมีอยู่ด้วยกัน 2 ชนิดคือ Memory และ Processor โปรโตคอลที่ใช้ในการสื่อสารกับสมาร์ตการ์ดจึงถูกแบ่งออกเป็น 2 ชนิดหลักเช่นกัน คือ การสื่อสารแบบ Synchronous และการสื่อสารแบบ Asynchronous ซึ่งทั้งสองแบบมีรายละเอียดย่อยดังรูปที่ 2.18



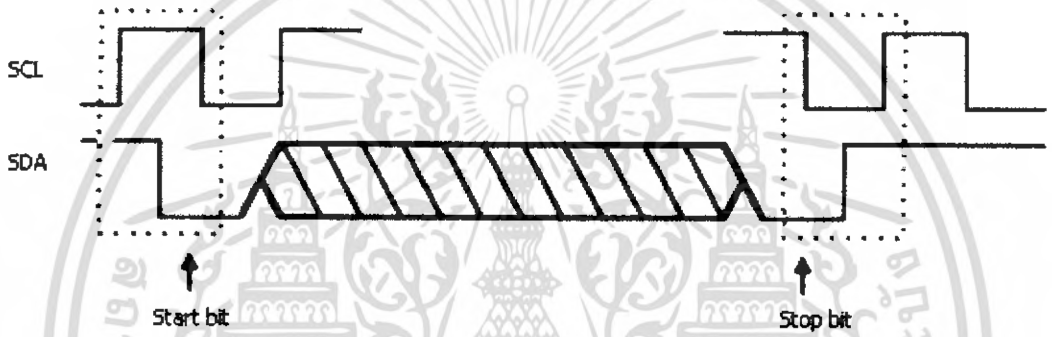
รูปที่ 2.18 รายเอียดของการสื่อสารแบบ Synchronous และ Asynchronous ในสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.11.1 โปรโตคอล I^2C (Inter-Integrated circuit)

โปรโตคอล I^2C เป็นโปรโตคอลที่มักพบได้บ่อยๆในอุปกรณ์อิเล็กทรอนิกส์ IC ที่อาศัยการสื่อสารแบบซิงเกิลบัส ซึ่งสมาร์ตการ์ดชนิด Memory ส่วนใหญ่จะใช้โปรโตคอลชนิดนี้ โปรโตคอลชนิดนี้มีการใช้สัญญาณเพียง 2 สัญญาณในการเชื่อมต่อกับอุปกรณ์ได้แก่ สัญญาณข้อมูล (SDA) และสัญญาณนาฬิกา (SCK) การรับส่งข้อมูลในโปรโตคอลชนิดนี้จะใช้สัญญาณนาฬิกาเป็นตัวกำหนดจังหวะการรับ-ส่งข้อมูลแต่ละบิต โดยมีข้อกำหนดเพียงความกว้างของสัญญาณนาฬิกาและบิตข้อมูลจะต้องไม่ต่ำกว่าที่กำหนดเท่านั้น

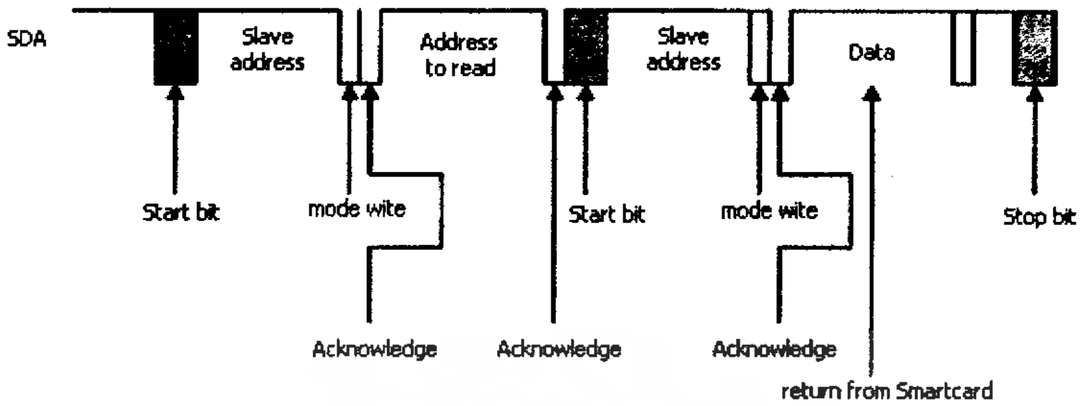
การรับส่งข้อมูลในโปรโตคอล I^2C อาศัยการเปลี่ยนระดับสัญญาณนาฬิกาเป็นตัวกำหนดการรับส่งบิตข้อมูล ซึ่งบิตเริ่มต้นส่งข้อมูล (Start bit) จะกระทำที่ขอบขาลงของสัญญาณนาฬิกา ส่วนบิตสิ้นสุด (Stop bit) จะกระทำที่ขอบขึ้นลงของสัญญาณนาฬิกาดังรูปที่ 2.19



รูปที่ 2.19 สัญญาณเริ่มต้น และสิ้นสุดการส่งข้อมูล

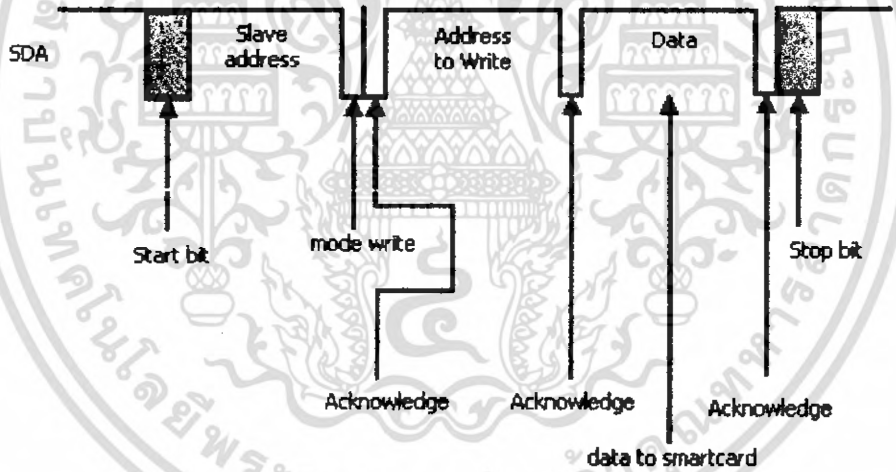
การอ่านข้อมูลจากสมาร์ตการ์ดด้วยโปรโตคอล I^2C จะทำได้ครั้ง 1 ไบต์เท่านั้น ซึ่งการส่งข้อมูลเพื่อขออ่านข้อมูลนั้นต้องมีการส่ง slave address (7 บิต + 1 บิตกำหนดโหมด) ไปเป็นไบต์แรกและตามด้วยบิตกำหนดโหมดกำหนด (ให้เป็นการเขียนข้อมูลที่ลอจิก 0) จากนั้นทำการเปลี่ยนลอจิกของสัญญาณ SDA ให้เป็น 1 เพื่อรอบบิต Acknowledge (บิตตอบรับ)จากทางชิปสมาร์ตการ์ดซึ่งจะมีลอจิกเป็น 0 เมื่อได้รับบิต Acknowledge แล้วจึงค่อยส่งข้อมูลของแอดเดรสที่ต้องการอ่านข้อมูลตามเข้าไปและรอบิต Acknowledge เช่นเดิม (การที่บิตกำหนดโหมดถูกกำหนดให้เป็นการเขียนข้อมูล ก็เพื่อให้สมาร์ตการ์ดทำการจัดเก็บแอดเดรสที่ส่งไปลงบัฟเฟอร์และเตรียมเข้าสู่ขั้นตอนการอ่านหรือการเขียนข้อมูลต่อไป) หลังจากนั้นการอ่านข้อมูลก็จะเริ่มขึ้นโดยการส่ง slave address ไปเป็นไบต์แรกและตามบิตกำหนดโหมดให้เป็นโหมดการอ่านข้อมูล(ลอจิก 1) หลังจากที่ได้รับบิต Acknowledge สมาร์ตการ์ดจะส่งข้อมูลจากแอดเดรสที่ต้องการออกมาดังรูปที่ 2.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การอ่านข้อมูลจากหน่วยความจำข้อมูลของสมาร์ตการ์ด ชนิด Memory ด้วยโปรโตคอล I^2C

สำหรับการเขียนข้อมูลจะมีส่วนของข้อมูลสองไบต์แรกที่เหมือนกับการอ่านข้อมูล แต่หลังจากที่ส่งแอดเดรสและได้รับบิต Acknowledge จากชิปสมาร์ตการ์ด ให้ทำการส่งข้อมูลที่ต้องการเขียนไปทันที ถ้าการเขียนข้อมูลไม่เกิดความผิดพลาดชิปสมาร์ตการ์ดจะไม่ส่งบิต Acknowledge กลับออกมา ดังรูปที่ 2.21

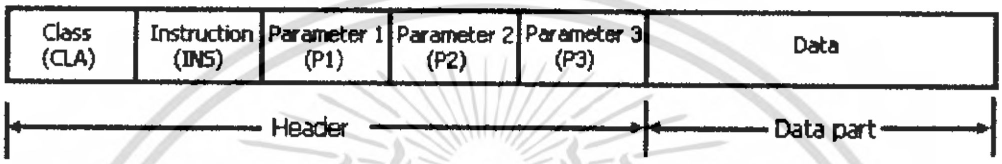


รูปที่ 2.21 การเขียนข้อมูลลงในหน่วยความจำข้อมูลของสมาร์ตการ์ดชนิด Memory ด้วยโปรโตคอล I^2C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

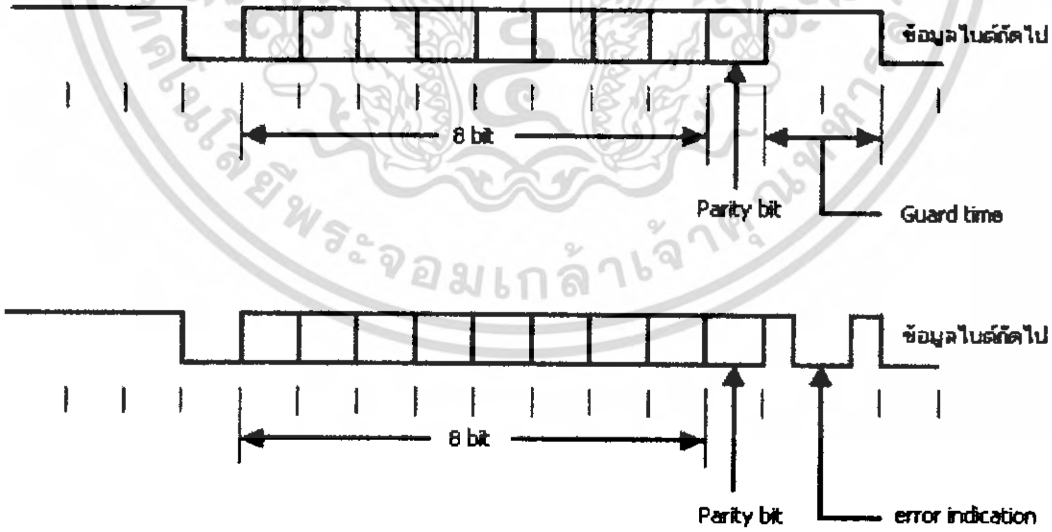
2.1.11.2 โปรโตคอล T = 0

โปรโตคอล T = 0 เป็นโปรโตคอลที่ถูกออกแบบให้มีจำนวนข้อมูลไม่มาก สามารถทำงานได้รวดเร็ว ถูกพัฒนาและใช้งานครั้งแรกในฝรั่งเศสจนกระทั่งกลายเป็นมาตรฐานตัวแรกของสมาร์ตการ์ด(ISO7816-3) และมีใช้กันอย่างแพร่หลายจนถึงปัจจุบัน สมาร์ตการ์ดชนิด processor ส่วนใหญ่ใช้งานโปรโตคอล T = 0 เป็นโปรโตคอลเริ่มต้นหลังจาก Cold reset โปรโตคอล T = 0 เป็นโปรโตคอลที่ทำงานในระดับไบต์ หรือเรียกว่า Byte-Oriented ซึ่งหมายถึงการประมวลผลส่วนที่เป็นข้อมูล(Data part)สามารถทำได้ในหน่วยที่เล็กที่สุดในระดับไบต์



รูปที่ 2.22 รูปแบบคำสั่งของโปรโตคอล T = 0

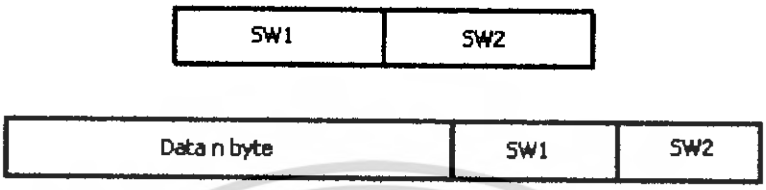
จากรูปจะเห็นได้ว่าโปรโตคอล T = 0 ประกอบด้วยข้อมูล Class (ชนิดหรือโปรโตคอลที่ใช้ติดต่อกับชิปสมาร์ตการ์ด), Instruction (คำสั่งที่ต้องการให้ชิปสมาร์ตการ์ดปฏิบัติ),Parameter 1-2-3 (ข้อมูลเพิ่มเติมของคำสั่ง), และData (ข้อมูลที่ต้องการรับหรือส่งกับชิปสมาร์ตการ์ด)โดยข้อมูลแต่ละตัวจะมีรูปแบบการรับส่งข้อมูลดังรูปที่ 2.23



รูปที่ 2.23 ไทม์มิ่งไดอะแกรมของการรับส่งข้อมูลด้วยโปรโตคอล T = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

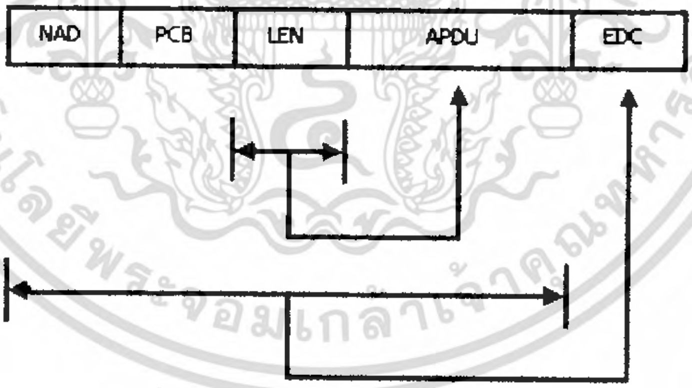
ในกรณีที่สมาร์ตการ์ดรับข้อมูลผิดพลาด ชิปสมาร์ตการ์ดจะทำการเปลี่ยนลอจิกเป็น 0 หลังการปิดบล็อกข้อมูลด้วย Stop bit ซึ่งเป็นส่วนของ Guard time ก่อนที่จะมีการรับข้อมูลไบต์ถัดไปโดยบิตข้อมูลข้อมูลนี้เรียกว่า Error indication หากไม่เกิดความผิดพลาดชิปสมาร์ตการ์ดจะทำการส่งข้อมูลตอบกลับซึ่งมีสองแบบ คือ ข้อมูลตอบกลับที่มีข้อมูล Status word (SW1,SW2)เพียงอย่างเดียว และข้อมูลตอบกลับที่มีทั้งข้อมูลที่ชิปสมาร์ตการ์ดต้องการส่งคืนให้ กับข้อมูล Status word ดังรูปที่ 2.24



รูปที่ 2.24 ข้อมูลตอบกลับจากชิปสมาร์ตการ์ดในโปรโตคอล T = 0

2.1.11.3 โปรโตคอล T = 1

โปรโตคอล T = 1 เป็นโปรโตคอลชนิด Asynchronous สำหรับสมาร์ตการ์ดชนิด processor โดยเฉพาะ ซึ่งโปรโตคอลชนิดนี้เรียกอีกอย่าง Block-Oriented เนื่องจากการประมวลส่วนที่เป็นข้อมูล(Data part) จะทำในระดับกลุ่มข้อมูลเท่านั้น โปรโตคอลนี้จะมีส่วนที่ครอบคลุมการทำงานของโปรโตคอล T=0 เพื่อให้สมาร์ตการ์ดที่ไม่สามารถประมวลผลหรือรับส่งข้อมูลไม่เกิน 256 ไบต์สามารถทำงานได้ด้วยโปรโตคอลชนิดนี้ โปรโตคอล T=0 จะมีรูปแบบดังรูปที่ 2.25



รูปที่ 2.25 รูปแบบของโปรโตคอล T=0

NAD (Node Address) ส่วนหัวของโปรโตคอล T=0 จะถูกเรียกว่า NOD ใช้สำหรับระบุชนิดของสมาร์ตการ์ดที่ต้องการสื่อสาร แต่ยังไม่มีเอกสารยืนยันวิธีใช้งานที่ชัดเจน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.17 รูปแบบของข้อมูล NAD

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
X	-	-	-	-	-	-	-	ควบคุม Vpp
-	X	X	X	-	-	-	-	DAD (destination address)
-	-	-	-	-	X	X	X	SAD (source addrass)

PCB (Protocol control byte) – ข้อมูลกำหนดรูปแบบการรับส่งข้อมูล และสั่งงาน สามารถแบ่งได้ 3 แบบคือ I block (Information block), R block (Reception acknow), และ S block (System block) ในการรับส่งข้อมูลที่มากกว่า 256 ไบต์ จำเป็นต้องทำการแบ่งข้อมูลออกเป็นบล็อก หนึ่งบล็อกสามารถรับส่งข้อมูลได้สูงสุด 255 ไบต์ และทยอยส่งที่ละบล็อก ซึ่งข้อมูลที่ส่งจากเทอร์มินอลจะเป็นข้อมูล I block เมื่อข้อมูลบิต M ถูกเซ็ตเป็น 1 ในบล็อกจะหมายถึงเทอร์มินอลจะทำการส่งข้อมูลเป็นจำนวนหลายบล็อกต่อเนื่องกัน ซึ่งสมาร์ตการ์ดจะทราบว่าเทอร์มินอลต้องการทำการส่งข้อมูลต่อเนื่องก็จะทำการส่งข้อมูล R block กลับมา หากไม่เกิดความผิดพลาด R block จะถูกใช้สำหรับนับจำนวนบล็อกที่ทำการรับส่ง(บิต 7 มีค่าเป็น 1, บิต 6 มีค่าเป็น 0) หลังจากนั้นเทอร์มินอลจะเซ็ตบิต N(S) ให้มีค่าเป็น 1 ในการส่งข้อมูลบล็อกถัดไป(บิต M ยังคงมีค่าเป็น 1) จนถึงข้อมูลบล็อกสุดท้าย ซึ่งข้อมูลบล็อกสุดท้าย บิต M จะถูกเปลี่ยนค่าเป็น 0 (บิต N(S) ยังคงมีค่าเป็น 1) ทำให้สมาร์ตการ์ดทราบว่าเป็นข้อมูลบล็อกสุดท้าย

ตารางที่ 2.18 รูปแบบข้อมูล PCB ชนิด I block

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
0	-	-	-	-	-	-	-	บิตแสดงให้ทราบว่าเป็น I block
-	N(S)	-	-	-	-	-	-	บิตแสดงให้ทราบว่าเป็นข้อมูลบล็อกต่อเนื่อง
-	-	M	-	-	-	-	-	บิตแสดงให้ทราบว่าเป็นรับส่งข้อมูลต่อเนื่อง
-	-	-	x	x	x	x	x	สงวนไว้ใช้ในภายหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.19 รูปแบบข้อมูล PCB ชนิด R block

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	0	-	-	-	-	-	-	บิตแสดงให้ทราบว่าเป็น R block
-	-	0	N(S)	0	0	0	0	ไม่มีความผิดพลาดในการข้อมูล
-	-	0	N(S)	0	0	0	1	เกิดความผิดพลาดของข้อมูล(EDC)
-	-	0	N(S)	0	0	1	0	ความผิดพลาดแบบอื่น

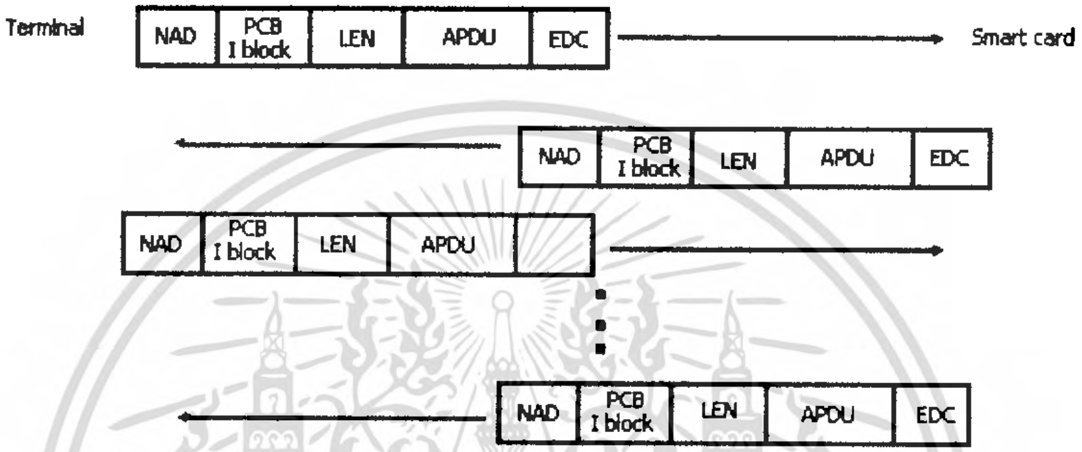
ตาราง 2.20 รูปแบบข้อมูล PCB ชนิด S block

บิตข้อมูล								ความหมาย
Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
1	1	-	-	-	-	-	-	บิตแสดงให้ทราบว่าเป็น S block
-	-	0	0	0	0	0	0	Resynch request (จากเทอร์มินอล)
-	-	1	0	0	0	0	0	Resynch response (จากสมาร์ตการ์ด)
-	-	0	0	0	0	0	1	การร้องขอเพื่อเปลี่ยนขนาดของข้อมูล
-	-	1	0	0	0	0	1	ข้อมูลตอบกลับจากการเปลี่ยนขนาดของข้อมูล
-	-	0	0	0	0	1	0	การร้องขอเพื่อยกเลิกคำสั่ง
-	-	1	0	0	0	1	0	ข้อมูลตอบกลับจากการร้องขอเพื่อยกเลิกคำสั่ง
-	-	0	0	0	0	1	1	การร้องขอเพื่อขอแก้ไขระยะเวลารอข้อมูล
								ตอบกลับ (จากสมาร์ตการ์ด)
-	-	1	0	0	0	1	1	ข้อมูลตอบกลับจากการร้องขอเพื่อขอแก้ไข
								ระยะเวลารอข้อมูลตอบกลับ (จากเทอร์มินอล)
-	-	1	0	0	1	0	0	Vpp มีความผิดพลาด (จากสมาร์ตการ์ด)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับส่งข้อมูลด้วยโปรโตคอล T=0 ในกรณีที่เกิดความผิดพลาดขณะรับส่งข้อมูลจะสามารถแก้ไขได้ 3 วิธีตามระดับความยุ่งยากที่เกิดขึ้น ซึ่งการแก้ไขจะทำเป็นลำดับๆจนถึงขั้นสุดท้าย

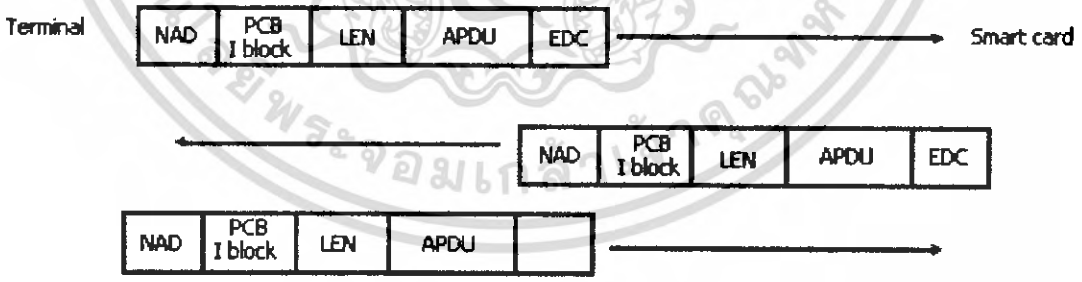
1. ทำการส่งข้อมูลบล็อกนั้นๆซ้ำอีกครั้ง
2. ทำการส่งข้อมูล S block เพื่อขอ Resynchronize และทำการส่งข้อมูลบล็อกนั้นซ้ำอีกครั้ง
3. ทำการรีเซ็ตสมาร์ตการ์ดและเริ่มขบวนการใหม่ทั้งหมด



รูปที่ 2.26 รูปแบบการรับส่งข้อมูลโปรโตคอล T=0

LEN (Length) = ความยาวของข้อมูลที่ใช้รับส่ง (APDU)

EDC – ข้อมูลตรวจสอบความถูกต้องของการรับส่งข้อมูลทั้งหมด ซึ่งมีการใช้คำนวณในสองลักษณะคือ LRC และ CRC

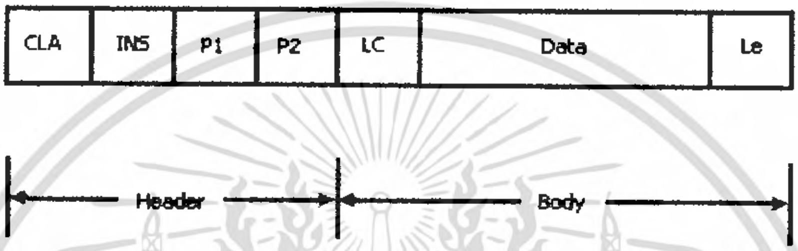


รูปที่ 2.27 รูปแบบการส่งข้อมูล S block จากสมาร์ตการ์ด เมื่อสมาร์ตการ์ดต้องประมวลผลเป็นเวลานาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.11.4 โปรโตคอล APDU

โปรโตคอล APDU (Application Protocol Data Unit)เป็นโปรโตคอลสำหรับการสื่อสารกับสมาร์ตการ์ดชนิด Processor ที่อยู่ในระดับชั้นที่ 7 ของ OSI model โปรโตคอล APDU เป็นโปรโตคอลตามมาตรฐาน ISO7816-4 ซึ่งถูกควบคุมโดยโปรโตคอล T=0 และ T=1 โดยที่โปรโตคอล APDU จะเป็นการเข้าถึงรายละเอียดของรูปแบบคำสั่งสำหรับสั่งงานชิปสมาร์ตการ์ด และกำหนดให้สมาร์ตการ์ดมีการใช้งานชุดคำสั่งที่เหมือนกันตามมาตรฐาน ISO7816-4 โดยเอาข้อดีในโปรโตคอล T=0 และ T=1 มาปรับเปลี่ยนให้เหมาะสมยิ่งขึ้น ซึ่งมีรายละเอียดดังนี้



รูปที่ 2.28 รูปแบบชุดคำสั่งของโปรโตคอล APDU

ข้อมูลแต่ละไบต์ที่ส่งให้แก่ชิปสมาร์ตการ์ดมีความหมายดังนี้

CLA (Class) – เป็นข้อมูลที่บอกให้ชิปทราบว่าจะใช้ชุดคำสั่งของสมาร์ตการ์ดชนิดใดซึ่งมีรายละเอียดดังตารางที่ 3.21

ตารางที่ 2.21 ส่วนประกอบของโปรโตคอล APDU

Class	ชนิดของสมาร์ตการ์ด
0x	สมาร์ตการ์ดทั่วไปที่ถูกสร้างขึ้นตามมาตราฐาน ISO7816-4
80	สมาร์ตการ์ดที่ใช้เป็นบัตรเงินสดอิเล็กทรอนิกส์ตามมาตรฐาน EN1546-3
8x	สมาร์ตการ์ดที่ใช้เป็นบัตรเครดิตตามมาตรฐาน EMV-2
A0	สมาร์ตการ์ดที่ใช้เป็น SIM card ของโทรศัพท์มือถือ (GSM card) ตามมาตรฐาน
	prETS300,600,GSM,11.11,และEN726-3

Class

ชนิดของสมาร์ตการ์ด

Ox

สมาร์ตการ์ดทั่วไปที่ถูกสร้างขึ้นตามมาตรฐาน ISO7816-4

80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ตการ์ดที่ใช้เป็นบัตรเงินสดอิเล็กทรอนิกส์ตามมาตรฐาน EN1546-3

8X

สมาร์ดการ์ดที่ใช้เป็นบัตรเครดิตตามมาตรฐาน EMV-2

A0

สมาร์ตการ์ดที่ใช้เป็น SIM card ของโทรศัพท์มือถือ (GSM card) ตามมาตรฐาน prETS300, 600 GSM 11.11, และ EN726-3

INS (Instruction) – คำสั่งในการสั่งงานสมาร์ตการ์ด ซึ่งต้องใส่ตามชุดคำสั่งของสมาร์ตการ์ดรุ่นที่ใช้งาน

P1 (Parameter 1) - ข้อมูลเพิ่มเติมสำหรับชุดคำสั่งส่วนที่หนึ่ง

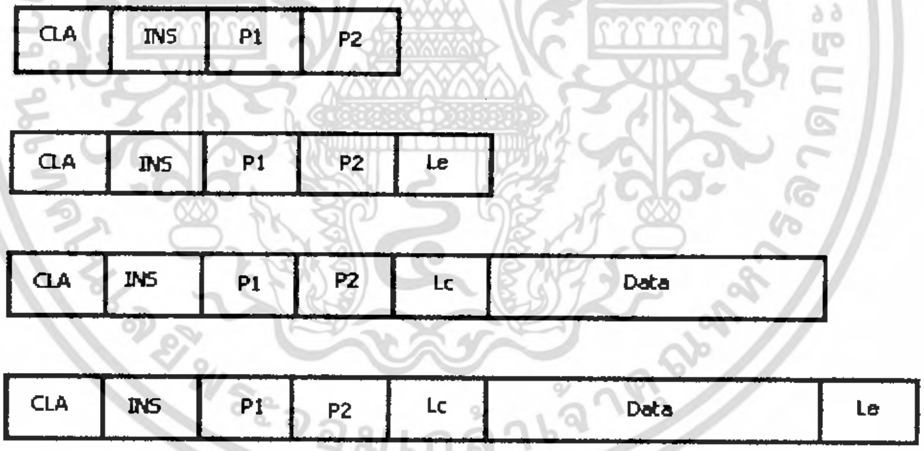
P2 (Parameter 2) – ข้อมูลเพิ่มเติมสำหรับชุดคำสั่งส่วนที่สอง

Lc (length command) – ความยาวข้อมูลคำสั่งที่ต้องการส่งให้แก่ชิปสมาร์ตการ์ด

Data – ข้อมูลคำสั่งที่ต้องการส่งให้แก่ชิปสมาร์ตการ์ด

Le (Length executed) – ความยาวข้อมูลที่ได้รับจากชิปสมาร์ตการ์ด

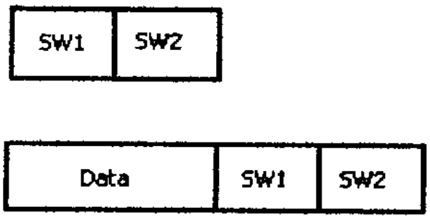
การส่งชุดคำสั่งให้แก่ชิปสมาร์ตการ์ดสามารถแบ่งออกได้ 4 แบบ โดยแต่ละแบบจะได้รับข้อมูลตอบกลับ(Response)ที่แตกต่างกัน 2 แบบดังนี้



รูปที่ 2.29 การส่งชุดคำสั่งให้แก่ชุดสมาร์ตการ์ดทั้ง 4 แบบ

การส่งชุดคำสั่งให้แก่มี 4 แบบได้แก่ การส่งเฉพาะส่วนที่เป็นชุดคำสั่ง การส่งชุดคำสั่งลักษณะนี้ส่วนมากเป็นคำสั่งที่ใช้สั่งงานให้สมาร์ตการ์ดบางอย่างที่ไม่ต้องส่งข้อมูลกลับมาเช่น คำสั่งเลือกไฟล์, คำสั่ง Card Present, ฯ การส่งชุดคำสั่งแบบที่สองเป็นการส่งชุดคำสั่งเพื่อร้องขอข้อมูลจากชิปสมาร์ตการ์ดเช่น คำสั่งอ่านไฟล์, คำสั่งสำหรับอ่านผลลัพธ์ ส่วนการส่งชุดคำสั่งแบบสุดท้ายเป็นการส่งข้อมูลคำสั่งพิเศษ หรือข้อมูลสำหรับเขียนลงในไฟล์ซึ่งจะต้องมีการส่งข้อมูลต่อท้ายชุดคำสั่งไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 ข้อมูลตอบกลับจากสมาร์ดการ์ดทั้ง 2 แบบ

ในส่วนของข้อมูลตอบกลับจากสมาร์ตการ์ดจากรูปที่ 2.30 จะเห็นได้ว่าแบบแรกมีเพียง SW1 และ SW2 (SW – Status Word) Status Word เป็นข้อมูลที่ชิปสมาร์ตการ์ดส่งกลับมาให้ทราบถึงผลการปฏิบัติตามคำสั่ง ซึ่งจะมีความหมายตามชนิดของสมาร์ตการ์ดดังตารางที่ 3.21 ส่วนแบบที่สองเป็นแบบที่มีการส่งข้อมูลจากชิปสมาร์ตการ์ดจากการส่งชุดคำสั่ง เพื่อร้องขอข้อมูลให้แก่ชิปสมาร์ตการ์ด

ตารางที่ 2.22 ชนิดของ Status Word ตามชนิดของสมาร์ตการ์ดและการประมวลผล

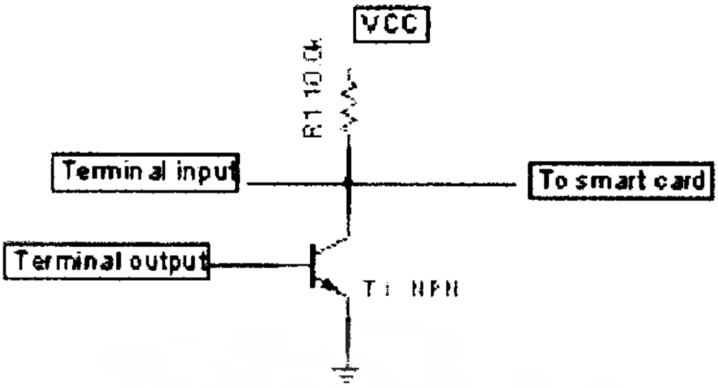
status Word	สามารถทำการประมวลผลได้		ไม่สามารถทำการการประมวลผลได้	
	ประมวลผลเสร็จสมบูรณ์	มีบางสิ่งผิดพลาด	ประมวลผลผิดพลาด	คำผิดพลาด/ใช้คำสั่งผิด
61XX,9000	X			
62XX		X		
63XX		X		
64XX			X	
65XX			X	
67XX ถึง 6FXX				X

สำหรับข้อมูลตอบกลับจากสมาร์ตการ์ดแบบที่สอง (จากรูปที่ 2.22) จะมีข้อมูลที่ส่งกลับมาจากชิปสมาร์ตการ์ด ซึ่งข้อมูลส่วนนี้จะเกิดขึ้นในกรณีใช้คำสั่งจำพวกร้องขอข้อมูลจากชิปสมาร์ตการ์ด โดยความยาวของข้อมูลที่ส่งกลับจะเท่ากับความยาวที่ร้องขอไป หากชิปสมาร์ตการ์ดไม่สามารถปฏิบัติตามได้ ข้อมูลตอบกลับจะมีเพียง Status Word ที่เป็นโค้ดแสดงความผิดพลาดเท่านั้น ส่วนรายละเอียดทั้งหมดของ Status Word สามารถดูได้จาก Data Sheet ของสมาร์ตการ์ดแต่ละรุ่น

2.1.12 การเชื่อมต่อชิปสมาร์ตการ์ด (Smart card Interface)

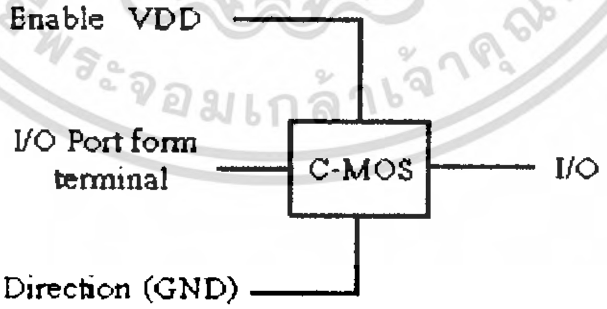
ในการสร้างวงจรสำหรับเชื่อมต่อกับชิปสมาร์ตการ์ดเป็นเรื่องที่ไม่ยากนักหากมีพื้นฐานความรู้ด้านไฟฟ้าอิเล็กทรอนิกส์ การที่สมาร์ตการ์ดถูกแบ่งออกเป็นสองชนิดทำให้วงจรสำหรับเชื่อมต่อต้องมีการออกแบบที่แตกต่างกัน ลำพังเพียงใช้ซอฟต์แวร์มาควบคุมการเปลี่ยนระดับสัญญาณเพียงอย่างเดียวไม่สามารถทำได้อย่างแน่นอน ดังนั้นการใช้ฮาร์ดแวร์เข้าช่วยจึงเป็นทางออกที่ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.31 ตัวอย่างวงจรสำหรับติดต่อกับชิปสมาร์ดการ์ด

วงจรที่ใช้ในการเชื่อมต่อกับชิปสมาร์ดการ์ด จะเป็นวงจรที่ทำงานในลักษณะแบบเดียวกับวงจรดิจิตอลชนิด C-MOS ซึ่งสามารถทำงานได้ที่แรงดันไฟฟ้า 3 – 5 โวลต์ แต่ในสมาร์ดการ์ดมีการใช้งานเพียง 3 โวลต์เท่านั้น จากรูปที่ 3.22 จะเห็นได้ว่าเพียงทรานซิสเตอร์เพียงตัวเดียวก็สามารถเชื่อมต่อกับชิปสมาร์ดการ์ดได้แล้ว วิธีการนี้ในสภาวะนิ่ง (Idle) ระดับลอจิก (แรงดันไฟฟ้า)ที่ขาเบสจะเป็น 0 ทำให้ทรานซิสเตอร์ไม่ทำงาน เมื่อมีข้อมูลถูกส่งมาที่ขาเบสกระแสที่เกิดขึ้นจากบิตข้อมูลที่เปลี่ยนลอจิกจาก 0 เป็น 1 จะทำให้ทรานซิสเตอร์ทำงาน สัญญาณข้อมูลจะไปปรากฏที่ขาคอลเล็กเตอร์ ซึ่งสัญญาณที่ขาคอลเล็กเตอร์จะถูกส่งต่อให้ชิปสมาร์ดการ์ดโดยตรงในทางกลับกันถ้าชิปสมาร์ดการ์ดทำการส่งข้อมูลกลับออกมาที่ขาคอลเล็กเตอร์ ระดับแรงดันไฟฟ้าที่ขาคอลเล็กเตอร์จะมีการเปลี่ยนแปลงตามลอจิกของบิตข้อมูลที่ส่งมาจากชิปสมาร์ดการ์ด โดยขณะที่บิตที่มีค่าเป็น 1 ค่าแรงดันไฟฟ้าที่ขาคอลเล็กเตอร์จะคงที่ แต่ถ้าบิตที่มีค่าเป็น 0 ค่าแรงดันไฟฟ้าที่ขาคอลเล็กเตอร์จะลดลงเช่นกัน หรืออีกวิธีหนึ่งที่ง่ายกว่าวิธีการใช้ทรานซิสเตอร์ เราสามารถใช้วงจรลอจิก (C-MOS) มาต่อเป็นวงจรง่ายดังรูปที่ 2.32



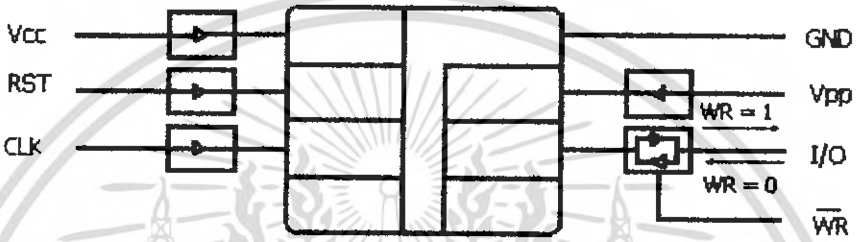
รูปที่ 2.32 ตัวอย่างวงจรเชื่อมต่อกับชิปสมาร์ดการ์ดด้วยวงจรลอจิก (C-MOS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเชื่อมต่อกับสมาร์ตการ์ดทั้งชนิด Memory และ Processor จะมีความแตกต่างกันในเรื่องของสัญญาณนาฬิกา (CLK) และวิธีการรับ-ส่งข้อมูล (I/O) ซึ่งในที่นี้จะทำการแยกวงจรออกเป็นสองแบบดังนี้

2.1.12.1 การเชื่อมต่อชิปสมาร์ตการ์ดชนิด Memory (Synchronous card Interface)

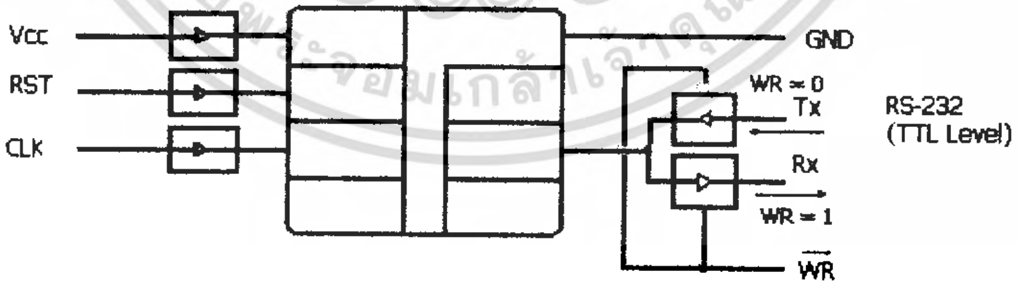
การเชื่อมต่อชิปสมาร์ตการ์ดชนิด Memory เราจะทำการสวิงสัญญาณนาฬิกาเองเพื่อกำหนดจังหวะการส่งข้อมูลแต่ละบิต สำหรับข้อมูล I/O เราสามารถใช้บิตข้อมูลแบบสองทิศทางเพียงบิตเดียว โดยกำหนดจังหวะการส่งข้อมูลด้วยสัญญาณ $\overline{WR}$ ซึ่งจะมีลอจิกเป็น 0 เมื่อต้องการส่งข้อมูลให้แก่สมาร์ตการ์ด ดังรูปที่ 2.33



รูปที่ 2.33 ตัวอย่างวงจรเชื่อมต่อชิปสมาร์ตการ์ดชนิด Memory

2.1.12.2 การเชื่อมต่อชิปสมาร์ตการ์ดชนิด Processor (Asynchronous card Interface)

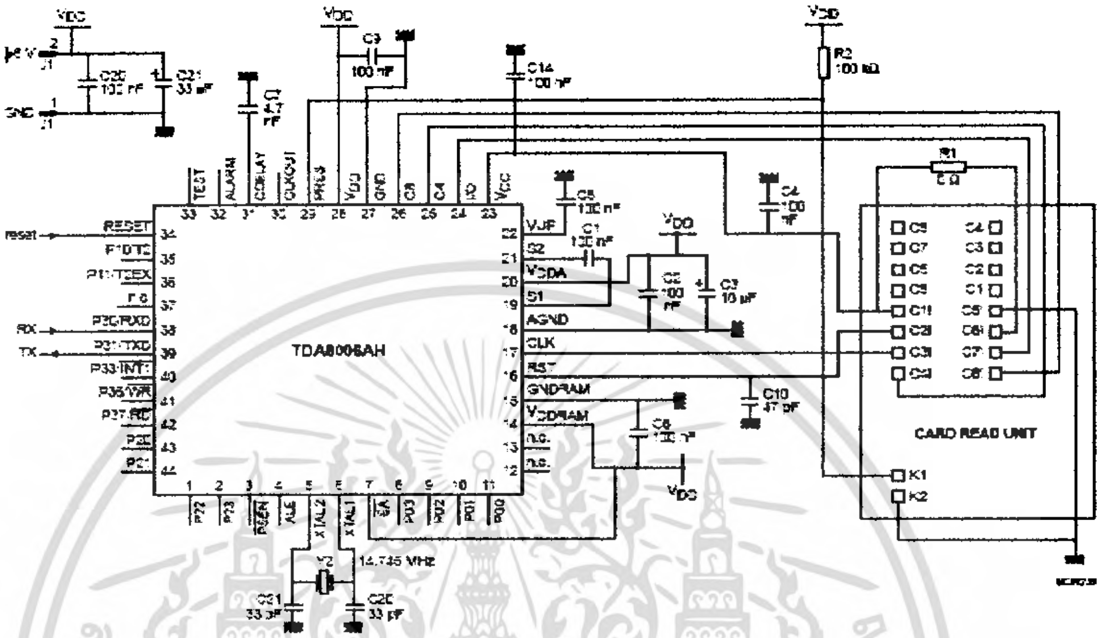
ในกรณีที่เป็นสมาร์ตการ์ดชนิด Processor วงจรเชื่อมต่อจะมีความยุ่งยากมากขึ้น สัญญาณนาฬิกาที่ป้อนให้แก่ชิปสมาร์ตการ์ด จะเป็นสัญญาณนาฬิกาแบบต่อเนื่อง (Continuous clock) ที่ความถี่ 3.5712 เมกะเฮิรตซ์ ตลอดเวลาที่ใช้งาน ส่วน I/O จะใช้การรับส่งข้อมูลแบบ Asynchronous โดยใช้สัญญาณ $\overline{WR}$ เป็นตัวกำหนดช่วงการรับหรือการส่งข้อมูล ซึ่งในสภาวะปกติจะรอรับข้อมูลตลอดเวลา ($\overline{WR}$=1) ดังรูปที่ 2.34



รูปที่ 2.34 ตัวอย่างวงจรเชื่อมต่อชิปสมาร์ตการ์ดชนิด Processor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ใช้ IC สำหรับเชื่อมต่อชิปสมาร์ตการ์ดก็สามารถทำได้ง่ายๆ ตามเอกสารแสดงคุณสมบัติ Data Sheet ของ IC เบอร์นั้นๆ ดังรูปที่ 2.35



รูปที่ 2.35 ตัวอย่างวงจรเชื่อมต่อกับชิปสมาร์ตการ์ดด้วย IC เบอร์ TDA8006AH

2.1.13 สมาร์ตการ์ดรีดเดอร์

ส่วนสำคัญที่ใช้ในการเชื่อมต่อกับสมาร์ตการ์ดก็คือ สมาร์ตการ์ดรีดเดอร์ เนื่องจากเป็นส่วนที่ใช้เชื่อมต่อกับหน้าสัมผัสกับชิปสมาร์ตการ์ดโดยตรง ทำให้สมาร์ตการ์ดรีดเดอร์ต้องถูกออกแบบมาเป็นพิเศษ ซึ่งสมาร์ตการ์ดรีดเดอร์เป็นชิ้นส่วนที่มีราคาสูงที่สุดในบรรดาอุปกรณ์ที่เกี่ยวข้องกับสมาร์ตการ์ดทั้งหมด เพราะส่วนที่เป็นขาโลหะที่ใช้ในการกดลงกับหน้าสัมผัสชิปต้องผลิตด้วยโลหะที่มีคุณสมบัติที่ทนทาน แต่ต้องไม่ทำอันตรายต่อหน้าสัมผัสชิปสมาร์ตการ์ด และต้องมีขนาดตามมาตรฐาน ISO7816-2 ทำให้ต้องใช้วัสดุและกระบวนการผลิตที่ยุ่งยากมากกว่าปกติ สมาร์ตการ์ดรีดเดอร์ที่ใช้กันอยู่ มีสองชนิดการเชื่อมต่อของสมาร์ตการ์ด ดังนี้

2.1.13.1 สมาร์ตการ์ดรีดเดอร์ชนิดหน้าสัมผัส

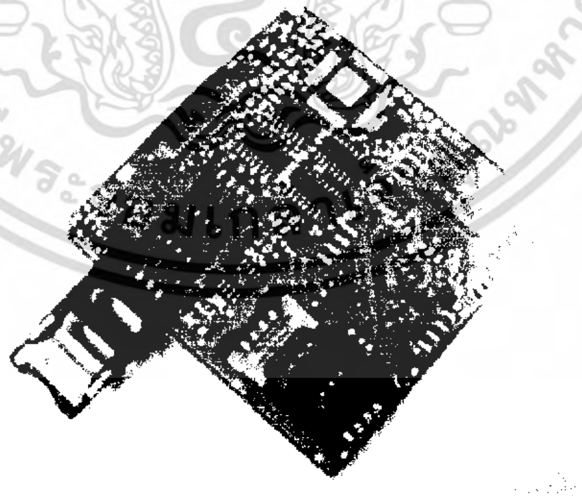
สมาร์ตการ์ดรีดเดอร์ชนิดหน้าสัมผัส เป็นสมาร์ตการ์ดรีดเดอร์ชนิดแรกที่ถูกสร้างพร้อมๆกับสมาร์ตการ์ดถูกสร้างขึ้นมาครั้งแรก โดยมีส่วนประกอบของโครงหลักของสมาร์ตการ์ดรีดเดอร์,ขาสัมผัส, และสวิตช์ตรวจสอบบัตร ซึ่งโครงสร้างหลักของสมาร์ตการ์ดรีดเดอร์เป็นแผ่นพลาสติกที่ขึ้นรูปสำหรับเป็นช่องทางให้สมาร์ตการ์ดสอดเข้ามา โดยส่วนในสุดของช่องจะติดตั้งสวิตช์ตรวจสอบบัตร เมื่อบัตรถูกเสียบเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาจนสุด ตัวบัตรจะไปกดสวิตช์ตรวจสอบบัตรทำให้สวิตช์ ON ทำให้เทอร์มินอลทราบว่ามีบัตรสอดเข้ามา นอกจากนี้โครงหลักของสมาร์ตการ์ดรีดเดอร์จะมีการเจาะช่องสำหรับติดตั้งขาสัมผัสในตำแหน่งที่ตรงกับหน้าสัมผัสของสมาร์ตการ์ดดังรูปที่ 2.36

รูปที่ 2.36 ตัวอย่างสมาร์ตการ์ดรีดเดอร์ที่ใช้บัตรชนิด ID-01

สำหรับสมาร์ตการ์ดรีดเดอร์ที่ใช้บัตรชนิด ID-000 ส่วนมากใช้ในโทรศัพท์เคลื่อนที่ระบบGSM และ EDC รุ่นใหม่ๆที่สามารถใช้ในการทำรายการเกี่ยวกับสมาร์ตการ์ด ซึ่งมีหลักการทำงานเหมือนกับสมาร์ตการ์ดรีดเดอร์ที่ใช้กับบัตรชนิด ID-01 แต่บัตรชนิด ID-000 จะอยู่ในสมาร์ตการ์ดรีดเดอร์ตลอดเวลาเท่านั้นเอง



รูปที่ 2.37 ตัวอย่างสมาร์ตการ์ดรีดเดอร์ที่ใช้บัตรชนิด ID-000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.13.2 สมาร์ตการ์ดรีดเดอร์ชนิด Contactless

สมาร์ตการ์ดรีดเดอร์ชนิด Contactless จะแตกต่างจากชนิดหน้าสัมผัส เนื่องจากสมาร์ตการ์ดชนิด Contactless ใช้คลื่นวิทยุความถี่ 13.56 เมกะเฮิรตซ์ ทำให้ส่วนที่เป็นขาสัมผัสต้องเปลี่ยนเป็นวงจรรับส่งและสายส่งคลื่นวิทยุ ซึ่งมีหลักการคล้ายกับเครื่องรับส่งคลื่นวิทยุ ซึ่งคลื่นที่รับส่งนั้นจะมีความแรงคลื่นไม่สูงนัก ทำให้ระยะการรับส่งข้อมูลลดลง โดยปกติแล้วจะอยู่ในระยะเพียง 3 ถึง 10 เซนติเมตรเท่านั้น (ความผิดพลาดของข้อมูลน้อยกว่า 1 เปอร์เซ็นต์) แต่ก็มีสมาร์ตการ์ดรีดเดอร์ชนิด Contactless บางรุ่นสามารถทำระยะได้สูงถึง 1-10 เมตร ทำให้เกิดความผิดพลาดของข้อมูลมากกว่า 10 เปอร์เซ็นต์ ซึ่งเกินกว่าจะรับได้ในงานบางอย่าง

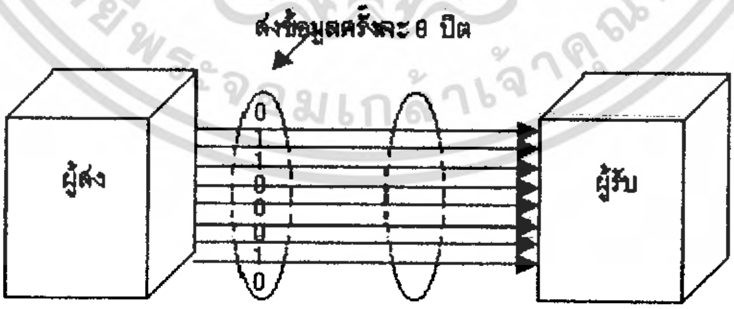
สมาร์ตการ์ดรีดเดอร์ชนิด Contactless ส่วนใหญ่จะเป็นชุดสำเร็จรูปจากผู้ผลิตโดยตรงซึ่งจะรวมเอาเสารับส่งสัญญาณ, วงจรภาครับส่ง, และชุดคอนโทรลเลอร์เข้าด้วยกัน โดยผู้ที่ต้องการนำไปใช้งานจะสามารถควบคุม และรับส่งข้อมูลโดยเชื่อมต่อผ่านพอร์ตสื่อสารของสมาร์ตการ์ดรีดเดอร์โดยตรง อีกทั้งเทคโนโลยีเกี่ยวกับสมาร์ตการ์ดชนิด Contactless ยังถูกผู้ผลิตสมาร์ตการ์ดจำกัดไม่ให้นำไปเผยแพร่จึงทำให้ไม่ค่อยพบเห็นอุปกรณ์ของสมาร์ตการ์ดรีดเดอร์ชนิด Contactless

2.2 การรับส่งข้อมูล

การรับส่งข้อมูลจะแบ่งออกเป็น 2 ประเภท คือการส่งข้อมูลแบบอนุกรม (Serial transmission) และการส่งแบบขนาน (parallel transmission)

2.2.1 การส่งข้อมูลแบบขนาน (Parallel transmission)

เป็นการส่งข้อมูลออกไปพร้อมๆกันได้คราวละหลายบิตการส่งแบบนี้สามารถส่งข้อมูลได้เร็วแต่จะมีข้อเสียคือ ค่าใช้จ่ายจะสูง เพราะจะต้องใช้สายจำนวนมาก นอกจากนั้นแล้วการส่งข้อมูลแบบขนานนี้จะเหมาะกับการส่งข้อมูลในระยะสั้นๆเท่านั้น เนื่องจากถ้าส่งไปในระยะทางที่ไกลแล้วข้อมูลภายในสายจะไปถึงปลายทางไม่พร้อมกัน ทำให้เกิดความผิดพลาดของการส่งข้อมูลได้

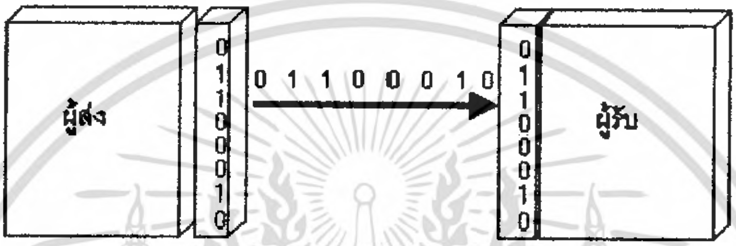


รูปที่ 2.38 การส่งข้อมูลแบบขนาน (Parallel transmission)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การส่งข้อมูลแบบอนุกรม (Serial transmission)

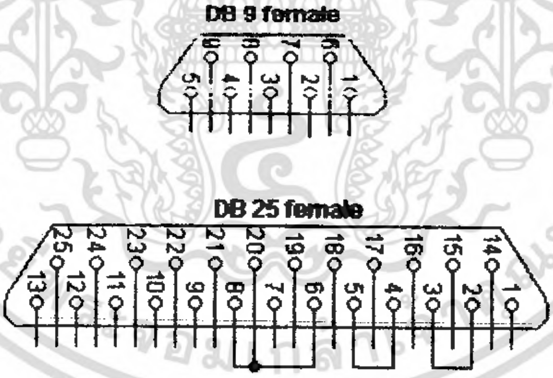
เป็นการส่งข้อมูลที่ใช้สายเพียงเส้นเดียว ดังนั้นบิตของข้อมูลที่ถูกส่งออกจะเรียงตามลำดับกันไป โดยปกติแล้วการส่งข้อมูลกันภายในเครื่องคอมพิวเตอร์จะใช้วิธีการส่งข้อมูลแบบขนาน แต่การส่งข้อมูลกับอุปกรณ์ภายนอกแล้วอาจจะเป็นการส่งแบบอนุกรมก็ได้ ดังนั้นถ้าการติดต่อกับอุปกรณ์ภายนอกเครื่องคอมพิวเตอร์ต้องใช้การส่งข้อมูลแบบอนุกรมแล้ว จะต้องทำการแปลงข้อมูลจากแบบขนานไปเป็นแบบอนุกรม (parallel-to-serial) เสียก่อน ส่วนเครื่องคอมพิวเตอร์ฝั่งรับก็จะต้องแปลงข้อมูลแบบอนุกรมจากสายส่งไปเป็นแบบขนานด้วย (serial-to-parallel)



รูปที่ 2.39 การส่งข้อมูลแบบอนุกรม (Serial transmission)

2.2.3 การรับส่งข้อมูลแบบอนุกรม (RS-232)

ในการส่งข้อมูลแบบอนุกรมนั้น สามารถแบ่งออกได้เป็น แบบซิงโครนัส และแบบอะซิงโครนัส

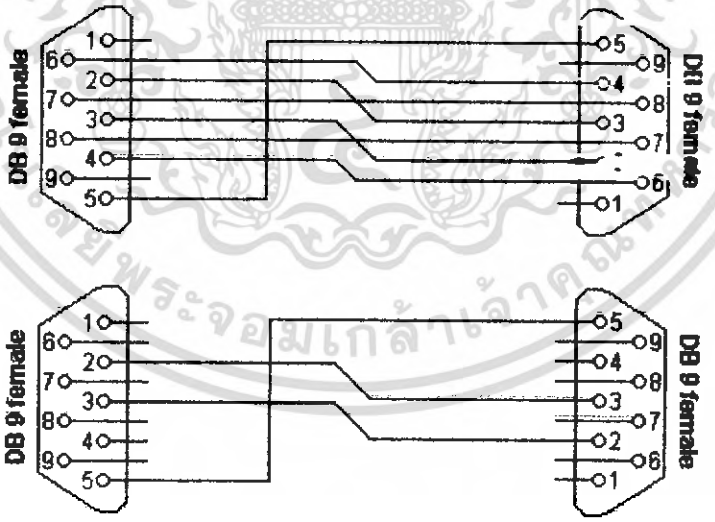


รูปที่ 2.40 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB และ DB-25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.23 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB และ DB-25

คอนเน็กเตอร์ DB-9	คอนเน็กเตอร์ DB-25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect : DCD	อินพุต
2	3	Received Data : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Ready : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Sent : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต



รูปที่ 2.41 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 แบบอะซิงโครนัส (Asychonous transmission)

เป็นการส่งข้อมูลครั้งละไบต์(1 Byte = 8 bit) การส่งแบบอะซิงโครนัสนี้ไม่ต้องอาศัยสัญญาณนาฬิกา (clock) ในการควบคุมจังหวะของการส่งข้อมูล ดังนั้นฝ่ายรับข้อมูลจึงไม่สามารถทราบเวลาที่แน่นอนของข้อมูลที่จะเข้ามาได้ ดังนั้นในแต่ละไบต์ที่ถูกส่งออกไปจึงต้องมีบิตเริ่มต้น (start bit) ซึ่งจะกำหนดค่าเป็น "0" และบิตสิ้นสุด (stop bit) ซึ่งจะกำหนดค่าให้เป็น "1"

2.2.5 แบบซิงโครนัส (Synchronous Transmission)

วิธีการส่งข้อมูลแบบนี้ไม่ต้องอาศัยบิตเริ่มต้น และบิตสิ้นสุด อีกทั้งยังสามารถส่งข้อมูลได้หลายๆ ไบต์ต่อเนื่องกันเป็น " เฟรม " ได้โดยไม่จำเป็นที่จะต้องมีช่องว่าง (gap) ระหว่างไบต์ของข้อมูล ฝ่ายรับข้อมูล (receiver) จะต้องทำหน้าที่ในการแยกเฟรมข้อมูลที่ได้มาออกเป็นไบต์เอง

เนื่องจากไม่จำเป็นต้องมีช่องว่าง (gap) และบิตเริ่มต้น-บิตสิ้นสุด ดังนั้นในการส่งข้อมูลแบบซิงโครนัสจะต้องมีสัญญาณนาฬิกาในการควบคุมจังหวะของการรับส่งข้อมูล ทำให้เกิดความสอดคล้องกันในการส่งและรับ ซึ่งจะส่งผลให้ผู้รับได้รับข้อมูลที่ถูกต้อง

2.3 ไมโครคอนโทรลเลอร์ MCS – 51

2.3.1 ความหมายของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งซึ่งรวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาต์พุต หน่วยคำนวณ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์และขนาดของระบบ ในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันก็คือ "ไมโคร" (Micro) ซึ่งหมายถึง ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmatic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า "คอนโทรลเลอร์" (Controller) หมายถึง อุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุมโดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.3.2 ไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชจาก ATMEL

ไมโครคอนโทรลเลอร์ที่ใช้อ้างอิงเพื่อการเรียนรู้และใช้งานในที่นี้เป็นไมโครคอนโทรลเลอร์ตระกูล MCS – 51 ซึ่งมีหน่วยความจำภายในเป็น แบบแฟลช (flash memory) ของ Atmel microcontroller โดยจะเน้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปที่เบอร์ AT89C51 มีหน่วยความจำโปรแกรมแบบแฟลชขนาด 4 กิโลไบต์ หน่วยความจำข้อมูลแรม 128 ไบต์

คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียู ขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช ลบ และเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความ จำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทางสามารถใช้งานได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทม์เมอร์/เคาน์เตอร์ ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ในภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวอตช์ด็อกไทม์เมอร์ในตัว สำหรับในอนุกรม AT89Sxx

2.3.3 การจัดการขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปภาพที่ 3 และรูปภาพ 4 โดยมีรายละเอียดขั้นต้น ดังนี้

ขา VCC ใช้สำหรับต่อไฟเลี้ยง + 5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0 – P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (Float) จึงมีอินพุตอิมพีแดนซ์สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาของแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7X) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

ขาพอร์ต 1 (P1.0 – P1.7) มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย นอกจากนั้นในอนุกรม AT89Sxx จะใช้ขา P1.0 เป็นขาอินพุตสำหรับนับค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของไทม์เมอร์ 2 และ P1.1 เป็นขาอินพุตทริกเกอร์2 ในขณะที่ขา P1.4 – P1.7 เป็นขาสำหรับเชื่อมต่อแบบ SPI เพื่อทำการโปรแกรมข้อมูลในหน่วยความจำแฟลชภายใน

ขาพอร์ต 2 (P2.0 – P2.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8 – A15)

ขาพอร์ต 3 (P3.0 – P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล “1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุตอิมพีแดนซ์สูง จึงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ต 3 ยังมีขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นต้นต่อไปนี้

P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา INT0

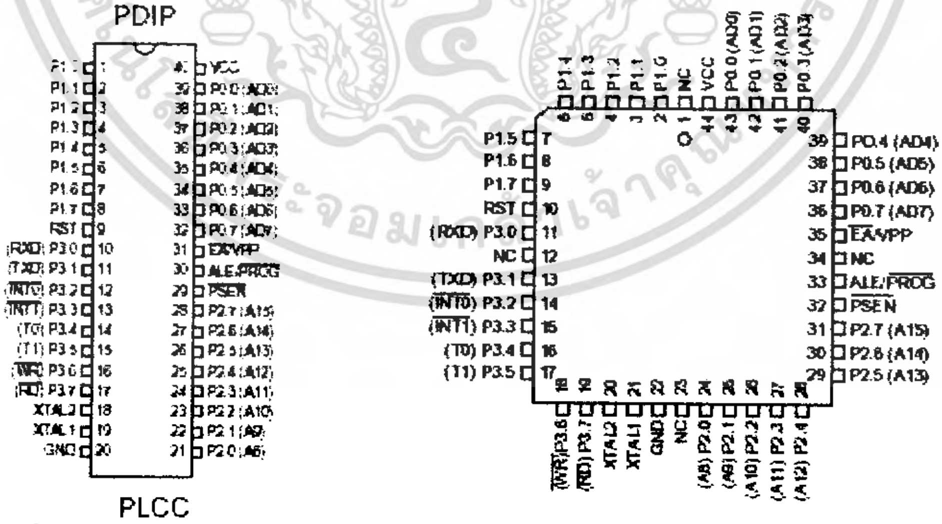
P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา INT1

P3.4 ใช้เป็นขาอินพุตสำหรับสัญญาณไทม์เมอร์จากภายนอกช่อง 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา T1

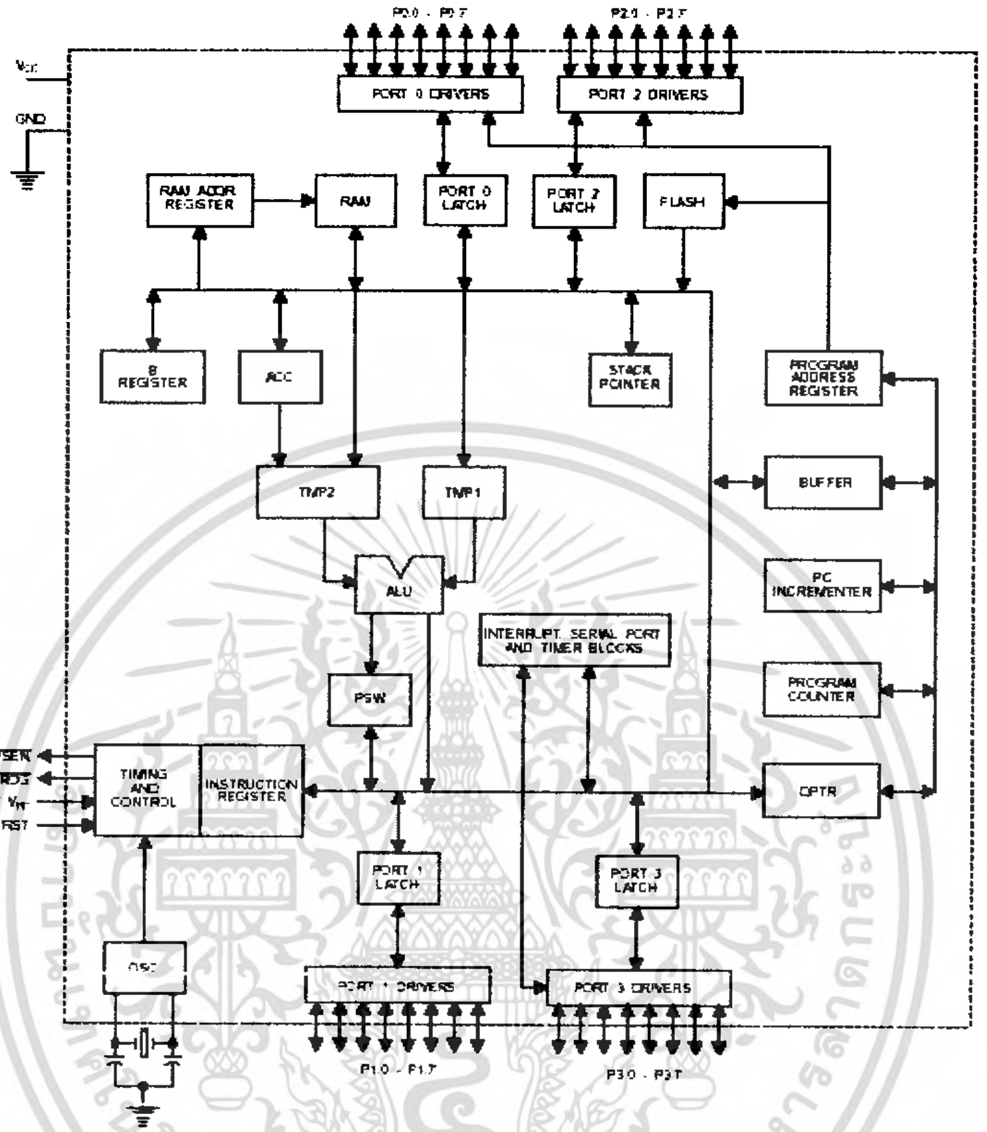
P3.6 ใช้เป็นขาสัญญาณ WR ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ RD ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก



รูปที่ 2.42 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS – 51 ตามมาตรฐาน Intel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.43 รายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS – 51 แบบแฟลชของ Atmel

2.4 จอแสดงผลแบบผลึกของเหลว (Liquid Crystal Display : LCD)

จอแสดงผลแบบผลึกของเหลว (Liquid Crystal Display : LCD) เป็นอุปกรณ์แสดงผล ที่นิยมใช้ในปัจจุบันเนื่องจากมีความเหมาะสมหลายๆด้าน เช่น การใช้กระแสไฟฟ้าต่ำ และสามารถแสดงผลเป็นตัวอักษรและตัวเลข หรือแสดงเป็นกราฟฟิก (เฉพาะรุ่น) ได้ จอแสดงผลแบบผลึกของเหลวนั้นมีขาสัญญาณ 14 ขา โดยแต่ละขามีหน้าที่การทำงานดังนี้

1. ขา 1 (Vss) เป็นขากราวด์ของจอแสดงผลแบบผลึกของเหลว
2. ขา 2 (Vcc) เป็นขาแรงดันไฟฟ้าขนาด +5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขา 3 (Vee) เป็นขารับแรงดันไฟฟ้าเพื่อปรับความเข้มของตัวอักษร โดยต่อกับกราวด์และมีความเข้มสูงสุด และถ้าต่อกับ Vcc จะมีความเข้มต่ำสุด
4. ขา 4 (RS) เป็นขาเลือกที่มีรีจิสเตอร์ภายในซึ่งมีอยู่ 2 ตัวคือ รีจิสเตอร์คำสั่ง (Instruction Registor : IR) และรีจิสเตอร์ข้อมูล (Data Registor : DR) ขณะเดียวกันถ้าเป็น " 1 " จะเป็นการเลือกรีจิสเตอร์ข้อมูล DR และถ้าเป็น " 0 " จะเป็นการเลือกคำสั่งที่รีจิสเตอร์ IR
5. ขา 5 (R/W) เป็นตัวเลือกว่าจะเขียนหรือจะอ่านข้อมูลโดยถ้าขานี้เป็น " 1 " จะเป็นการอ่านข้อมูลจากจอแสดงผลแบบผลึกของเหลว แต่ถ้าเป็น " 0 " จะเขียนข้อมูลเข้าจอแสดงผลแบบผลึกของเหลว
6. ขา 6 (E) เป็นขากำหนดสภาพการรับเขียนอ่านข้อมูล (Enable)
7. ขา 7 – ขา 14 (DB0 – DB7) เป็นขาที่ใช้ในการรับส่งข้อมูล

ในการใช้งานจอแสดงผลแบบผลึกของเหลวนั้นจะใช้ไมโครโปรเซสเซอร์ส่วนไมโครคอนโทรลเลอร์เป็นตัวควบคุมการทำงานโดยจอแสดงผลแบบผลึกของเหลว รับคำสั่งเฉพาะตามตารางที่ 2.25 มีคำสั่งควบคุมการทำงาน คำสั่งอ่าน หรือคำสั่งเขียนข้อมูล การใช้งานไมโครคอนโทรลเลอร์กับจอแสดงผลแบบผลึกของเหลว จะมีวิธีที่นิยมอยู่ 2 วิธีคือการต่อแบบหน่วยความจำในการเชื่อมต่อไมโครคอนโทรลเลอร์ จะมองเห็นจอแสดงผลแบบผลึกของเหลวเป็นหน่วยความจำตัวหนึ่ง และการต่อแบบอินพุต/เอาต์พุต โดยต่อร่วมกับไมโครโปรเซสเซอร์เบอร์ 8255

2.4.1 พื้นฐานในการใช้งานจอแสดงผลแบบผลึกของเหลว

การเขียนข้อมูลให้กับจอแสดงผลแบบผลึกของเหลวจะแบ่งเป็น 2 ลักษณะคือการเขียนคำสั่งและการเขียนข้อมูลโดยจะกำหนดด้วยขา RS คือถ้าขา RS เป็น " 0 " หมายถึงการเขียนคำสั่งควบคุมหรือการอ่านค่าแฟลกสภาพการทำงานของจอแสดงผลแบบผลึกของเหลว

หลักการในการเขียนข้อมูลให้กับจอแสดงผลแบบผลึกของเหลว นี่คือเมื่อมีการเขียนข้อมูลส่งไปยังจอแสดงผลแบบผลึกของเหลว จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่งตามค่าเวลาปฏิบัติการ (Execute time) ในตารางที่ 2.25 โดยระบบของไมโครคอนโทรลเลอร์จะสามารถตรวจสอบได้จากแฟลก BF และถ้าเรียบร้อยแล้วจึงจะสามารถเขียนข้อมูลชุดต่อไปได้ กรณีการต่อเป็นแบบพอร์ตอินพุตเอาต์พุตคือไม่สามารถอ่านข้อมูลกลับได้ โดยระบบไมโครคอนโทรลเลอร์ก็จะใช้วิธีการหน่วงเวลาแทน ในการเขียนข้อมูลให้กับจอแสดงผลแบบผลึกของเหลวนั้น สามารถที่จะทำได้ทั้ง 8 บิต แต่ให้เขียนสองครั้งคือ ขา DB4 – DB7 ใช้ระบบไมโครคอนโทรลเลอร์แบบ 4 บิต หรือเพื่อการประหยัดสาย การเขียนจะเหมือนกับ 8 บิต แต่ให้เขียนสองครั้งคือขา DB4 – DB7 ก่อนแล้วตามด้วยขา DB0 – DB3 ต้องกำหนดตามค่า DL ในคำสั่งฟังก์ชั่นเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำภายในจอแสดงผลแบบผลึกของเหลว (Display Data Ram) คือหน่วยความจุภายในจอแสดงผลแบบผลึกของเหลว ที่เป็นบัฟเฟอร์ของข้อมูลโดยถ้าเขียนรหัสแอสกี (ASCII) ใดๆลงไปในหน่วยความจำนี้ โดยจะปรากฏที่หน้าจอตัวอักษรทันที

CGRAM (Character Generator Ram) คือหน่วยความจำแรมภายในของจอแสดงผลแบบผลึกของเหลว สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8ตัว) โดยสามารถจะอ้างตำแหน่งได้ทั้งหมด 64 ไบต์ คือ 8 ตัวอักษรคูณกับ 8 แถว

รายละเอียดของแต่ละคำสั่ง

CLEAR DISPLAY

RS	R/w	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการลบหน้าจอนั้น สามารถเขียนตัวอักษรว่างลงไปในหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวทั้งหมด และกำหนดค่าที่อยู่ของหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวเป็น 0 พร้อมทั้งเคอร์เซอร์จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

CURSOR AT HOME

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

สำหรับการกำหนดค่าที่อยู่ (Address) ของหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวเป็น 0 พร้อมทั้งตัวเคอร์เซอร์ จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลในหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวไม่มีการเปลี่ยนแปลง

ENTRY MODE SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	I/D	S

I/D = 0 กำหนดทิศทางของเคอร์เซอร์ ของหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวให้เป็นแบบลดตำแหน่งลง

I/D = 1 กำหนดทิศทางของเคอร์เซอร์ ของหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวให้เป็นแบบลดตำแหน่งขึ้น

S = 0 เมื่อเขียนข้อมูลแล้วตัวเคอร์เซอร์ จะถูกเลื่อนไปตามทิศทางของค่า I/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S = 1 เมื่อเขียนข้อมูลแล้วตัวเคอร์เซอร์ จะอยู่กับที่และตัวอักษรจะถูกดันตามทิศทางของเคอร์เซอร์ในการกำหนดลักษณะของการแสดงผลนี้ให้กำหนดก่อนการเขียนข้อมูลลงในหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลว และเมื่อมีการกำหนดแล้วจะต้องไม่ใช้คำสั่งลบหน้าจออีก

DISPLAY ON/OFF

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	D	C	B

D = 0 กำหนดให้ปิดหน้าจอ

D = 1 กำหนดให้เปิดหน้าจอ

C = 0 กำหนดให้ปิดเคอร์เซอร์

C = 1 กำหนดให้เปิดเคอร์เซอร์ โดยเคอร์เซอร์จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่งเคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่งเคอร์เซอร์ กระพริบเป็นรูปสี่เหลี่ยม

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

S/C = 0 กำหนดให้เลื่อนเคอร์เซอร์ตามทิศทางซ้ายหรือขวาไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความบนแผงซึ่งจะแสดงตามทิศทางซ้ายหรือขวาไป 1 หลักทุกบรรทัด

FUNCTION SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับจอแสดงผลแบบผลึกของเหลวเป็นแบบ 4 บิต ซึ่งการกำหนดค่าทิศทางของเคอร์เซอร์นี้สามารถกระทำได้ที่ขา DB4 – DB7

DL = 1 กำหนดให้การติดต่อกับจอแสดงผลแบบผลึกของเหลว เป็นแบบ 8 บิต โดยการกำหนดค่าทิศทางของเคอร์เซอร์ สามารถกระทำได้ที่ขา DB4 – DB7 และมีการกำหนดให้เป็นแบบ 4 บิต ตั้งแต่ครั้งแรกหลังจากการจ่ายกระแสไฟฟ้าจะทำให้จอแสดงผล LCD มีการรับข้อมูลแบบ 4 บิตทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 ส่วน และ 1/11 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 ส่วน

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5*7 จุด

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5*10

SET CGRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนดของแรมเก็บตัวอักษร (CGRAM) เมื่อได้ทำการกำหนดไว้อ่านและเขียนข้อมูลที่ต่อจากนี้ จะเป็นไปตามตำแหน่งที่กำหนดทันที

SET DDRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

สำหรับการกำหนดตำแหน่งของหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลว ในการอ่านและเขียนข้อมูลที่ต่อจากนี้จะเป็นไปตามตำแหน่งที่กำหนดทันที ซึ่งตำแหน่งของหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลว ในแต่รุ่นจะมีความแตกต่างกันเนื่องจากจำนวนตัวอักษรต่อบรรทัดไม่เท่ากัน

BUSY FLAG AND ADDRESS READ

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	ADDRESS						

สำหรับการอ่านค่าแฟลก BF (BUSY FLAG) ซึ่งจะบอกถึงความพร้อมของจอแสดงผลแบบผลึกของเหลว ในการรับข้อมูลถ้าแฟลก BF เป็นลอจิก " 0 " หมายความว่า พร้อมที่จะรับข้อมูลชุดถัดไปได้แต่ถ้าแฟลก BF เป็นลอจิก " 1 " หมายความว่ายังไม่พร้อม นอกจากยังเป็นการอ่านค่าตำแหน่งหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวด้วย

WRITE DATA TO DDRAM OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	DATA							

สำหรับการเขียนข้อมูลลงหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลว หรือ CGRAM โดยเมื่อทำการเขียนแล้ว และตำแหน่งจะถูกเพิ่มหรือลดลงโดยอัตโนมัติตามที่กำหนดจากทิศทางของเคอร์เซอร์ในคำสั่ง ENTRY MODE SET และการเขียนข้อมูลลงหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลว หรือหน่วยความจำแรม ขึ้นกับว่าก่อนหน้าคำสั่งนี้กำหนดตำแหน่งที่ใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

READ DATA FROM DDRAM LCD OR CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	ADDRESS							

สำหรับการอ่านข้อมูลจากหน่วยความจำภายในของจอแสดงผลแบบผลึกของเหลว หรือ CGRAM เมื่อทำการเขียนแล้วตำแหน่งจะถูกเพิ่มหรือลดลงโดยอัตโนมัติ โดยตามที่กำหนดจากค่าทิศทางของเคอร์เซอร์ ในคำสั่ง ENTRY MODE SET และการอ่านหน่วยความจำภายในจอแสดงผลแบบผลึกของเหลวหรือ CGRAM ก็ขึ้นกับว่าก่อนหน้าคำสั่งนี้มีการกำหนดตำแหน่งที่ใด

ตารางที่ 2.24 การกำหนดคุณสมบัติต่างๆ ในการใช้งานของจอแสดงผลแบบผลึกของเหลว

INSTRUCTION	RS	R/W	DATA BIT								EXE TIME(us)
			7	6	5	4	3	2	1	0	
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*	40
SET CGRAM ADD	0	0	0	1	CGRAM ADDRESS						40
SET DDRAM ADD	0	0	1	DDRAM ADDRESS							40
BUSY FLAG, ADD.READ	0	1	BF	ADDRESS							0
CGRAM DDRAM WR	1	0	WRITE DATA								40
CGRAM DDRAM RD	1	1	READ DATA								40

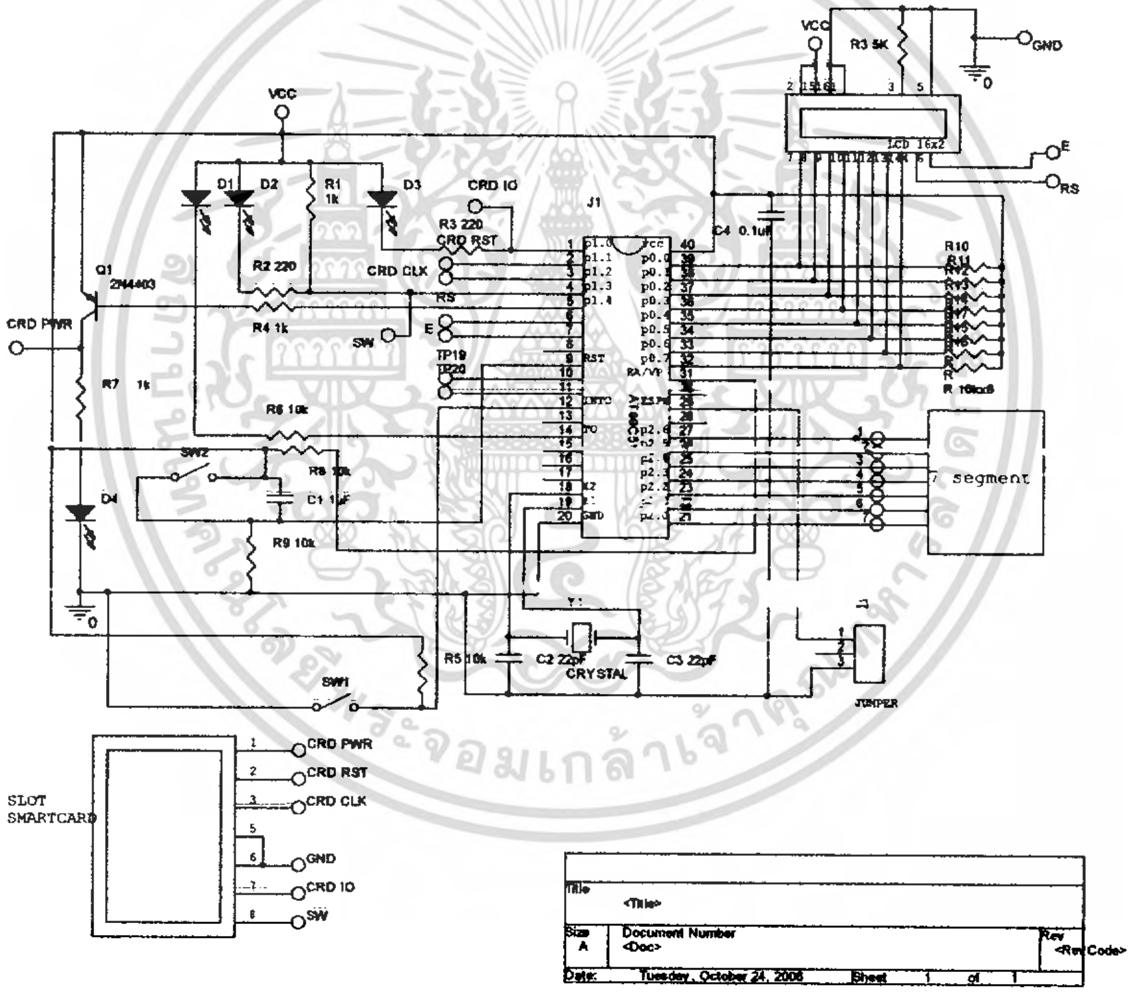
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การสร้างและการออกแบบ

ลำดับขั้นตอนในการออกแบบมีดังนี้

1. ออกแบบและสร้างวงจรเครื่องอ่าน – เขียนสมาร์ตการ์ด
2. เขียนโฟล์วชาร์ทการทำงานของเครื่องอ่าน – เขียนสมาร์ตการ์ด
3. เขียนโปรแกรมติดต่อกับบัตรสมาร์ตการ์ดตามโฟล์วชาร์ท
4. เขียนโปรแกรมรับหมายเลขลำดับคิวและบันทึกเสียงโดยใช้ซาวด์การ์ด

3.1 ออกแบบและสร้างวงจรเครื่องอ่าน – เขียนสมาร์ตการ์ด

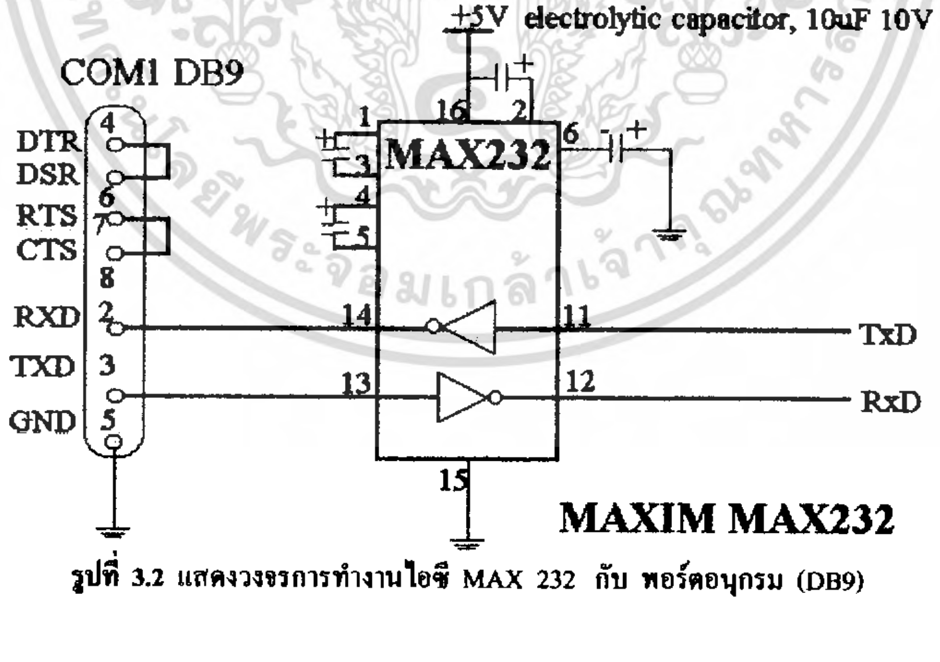


รูปที่ 3.1 แสดงวงจรการทำงานของเครื่องอ่าน – เขียนสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 3.1 แสดงวงจรการทำงานของเครื่องอ่าน – เขียนสมาร์ตการ์ด ซึ่งจะเห็นว่ามีช่องเสียบบัตรสมาร์ตการ์ด (Slot Smartcard) ซึ่งถูกต่อเข้ากับขาไมโครคอนโทรลเลอร์ โดยขาที่ 1 ต่อเข้ากับตำแหน่ง CRD IO ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาที่ใช้เขียนหรืออ่านข้อมูลจากบัตรที่ขานี้มีแอลอีดี 3 เป็นตัวแสดงผลเมื่อไมโครคอนโทรลเลอร์เขียนหรืออ่านข้อมูลจากบัตร โดยแอลอีดี 3 จะติดหรือดับตามสถานะลอจิกของขาที่ 1 ของไมโครคอนโทรลเลอร์ ขาที่ 2 ต่อเข้ากับตำแหน่ง CRD RST ของช่องเสียบบัตรสมาร์ตการ์ดซึ่งเป็นขาที่รีเซตการทำงาน ขาที่ 3 ต่อเข้ากับตำแหน่ง CRD CLK ของช่องเสียบบัตรสมาร์ตการ์ดเป็นขาที่ป้อนสัญญาณนาฬิกาอ้างอิงให้กับบัตรสมาร์ตการ์ด และขาที่ 4 ต่อเข้ากับตำแหน่ง SW ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาที่ใช้ตรวจสอบและแสดงผลว่ามีการเสียบบัตรเข้ามายังช่องเสียบบัตรแล้วหรือยัง ที่ขานี้จะมีแอลอีดี 2 เป็นตัวแสดงผลและติดสว่างเมื่อบัตรถูกเสียบเข้ามายังช่องเสียบบัตร และขาที่ 5 ต่อเข้ากับตำแหน่ง CRD PWR ของช่องเสียบบัตรสมาร์ตการ์ด ซึ่งเป็นขาควบคุมการจ่ายไฟเลี้ยงให้กับบัตร ด้วยการส่งให้ Q1 ปิดหรือเปิดวงจรที่ขานี้จะมีแอลอีดี 4 เป็นตัวแสดงผล เมื่อไฟเลี้ยงถูกจ่ายไปยังบัตร แอลอีดี 4 จะติดสว่างตาม และขาที่ 14 ของไมโครคอนโทรลเลอร์จะติดต่ออยู่กันแอลอีดี 1 ซึ่งจะกระพริบทุกๆ 1 วินาทีเพื่อแสดงว่าบอร์ดกำลังทำงานอยู่

จอแสดงผลแบบผลึกเหลวขนาด 16 ตัวอักษร 2 บรรทัด มีขาข้อมูล 8 เส้นต่อเข้ากับพอร์ต P0 และพอร์ต P1.5 กับ P1.6 จะต่อเข้ากับขา RS และ E ตามลำดับที่ 3 ของจอแสดงผลแบบผลึกเหลวจะเป็นขาปรับความเข้มของตัวอักษร ถ้าต่อลงกราวด์จะมีความเข้มสูงสุดและถ้าต่อกับแหล่งจ่ายไฟจะมีความเข้มต่ำสุดและที่ขา 15 กับ 16 เป็นตัวปรับความเข้มของไฟที่หน้าจอ และขาที่ 10 กับ 11 จะใช้ติดต่อสื่อสารข้อมูลกับคอมพิวเตอร์แบบพอร์ตอนุกรม

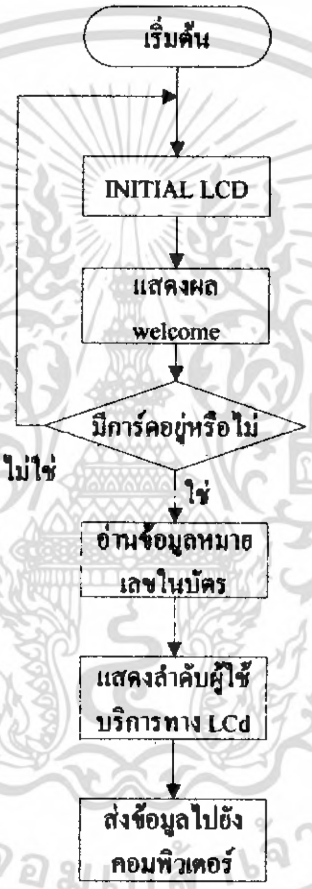


รูปที่ 3.2 แสดงวงจรการทำงานไอซี MAX 232 กับ พอร์ตอนุกรม (DB9)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 เป็นการแสดงวงจรไอซีMAX232 กับ พอร์ตอนุกรม (DB9) ซึ่งใช้สำหรับติดต่อสื่อสารกับคอมพิวเตอร์ โดยขา 11 ของไอซี MAX 232 ต่อกับ ขาที่ 11 ของไมโครคอนโทรลเลอร์เป็นขา ส่งข้อมูล (TXD) ส่วนขา 12 ของไอซีMAX232 ต่อกับขา 10 ของไมโครคอนโทรลเลอร์เป็นขารับข้อมูล(RXD) และที่ขา 14 ของไอซีMAX232 ต่อกับ ขา 2 ของ พอร์ตอนุกรม(DB9) ซึ่งเป็นขารับข้อมูล(RXD) ส่วนขา 13 ของไอซี MAX232 9 ต่อกับขา 3 ของพอร์ตอนุกรม(DB9) เป็นขาส่งข้อมูล(TXD)

3.2 เขียนโปรแกรมการการอ่านข้อมูลจากสมาร์ตการ์ด

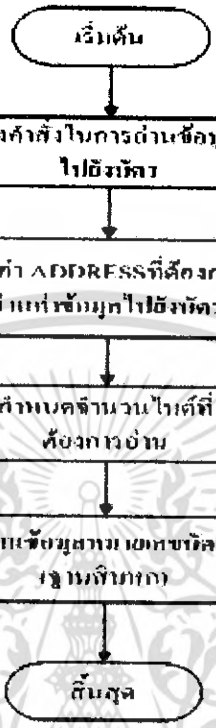


รูปที่ 3.3 โฟล์วชาร์ทการทำงานของโปรแกรมการอ่านข้อมูลจากบัตรสมาร์ตการ์ด

เริ่มต้นเครื่องอ่านสมาร์ตการ์ดจะทำการตรวจสอบว่ามีบัตรสมาร์ตการ์ดเข้ามาหรือไม่พร้อมแสดงคำว่า " WELCOME "เมื่อมีการเสียบบัตรสมาร์ตการ์ดเข้ามาซึ่งบัตรสมาร์ตการ์ด จากนั้นเครื่องอ่านสมาร์ตการ์ดก็จะทำการอ่านข้อมูลและแสดงลำดับผู้ใช้บริการทางจอ LCD จากนั้นก็ส่งข้อมูลหมายเลขบัตรไปเก็บไว้ในคอมพิวเตอร์ที่เคาน์เตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 การเขียนโฟล์วชาร์ทควบคุมการอ่านข้อมูลจากบัตร

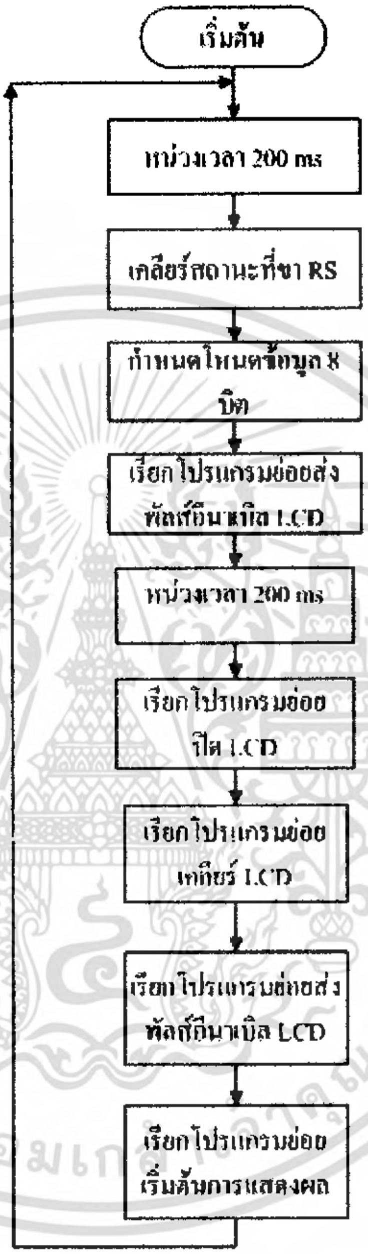


รูปที่ 3.4 โฟล์วชาร์ทการทำงานของโปรแกรมควบคุมการอ่านข้อมูลจากบัตร

เป็นการอ่านข้อมูลจากหน่วยความจำหลักของสมาร์ตการ์ดแล้วนำมาแสดงผล ในสภาวะแรกต้องทำการส่งข้อมูลซึ่งเป็นรหัสคำสั่งเพื่อบอกให้สมาร์ตการ์ดทราบว่าต้องการอ่านข้อมูลจากหน่วยความจำหลักแล้ว จากนั้นก็ต้องส่งค่าแอดเดรสที่เราต้องการอ่านข้อมูลไปด้วย พร้อมกับจำไบต์ที่ต้องการจะอ่านจากนั้นจะทำการอ่านข้อมูลจากบัตรซึ่งอยู่ในรูปของเลขฐานสิบหก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การเขียนโฟล์วชาร์ทควบคุมกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว



รูปที่ 3.5 โฟล์วชาร์ทการทำงานของโปรแกรมการกำหนดค่าเริ่มต้นของจอแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การที่ให้จอแสดงผลแบบผลึกเหลวทำงาน ต้องมีการตั้งค่าก่อนเพื่อให้จอแสดงผลแบบผลึกเหลวทำงานตามที่เราต้องการ เริ่มต้นต้องทำการหน่วงเวลาเพื่อให้จอแสดงผลแบบผลึกเหลวเตรียมพร้อมทำงาน เนื่องจากว่าจอแสดงผลแบบผลึกเหลวทำงานได้ช้ากว่าไมโครคอนโทรลเลอร์ จากนั้นเคลียร์สถานะที่ขา RS ให้เป็นลอจิก " 0 " เพื่อบอกให้จอแสดงผลแบบผลึกเหลวทราบว่าข้อมูลที่เข้ามาเป็นข้อมูลควบคุม จากนั้นก็ตั้งค่าให้เป็นการรับ – ส่งข้อมูลแบบ 8 บิต และทำการอีนาเบิลส่งพัลส์ไปยังขา E ของจอแสดงผลแบบผลึกเหลว พร้อมทำการหน่วงเวลาเพื่อให้จอแสดงผลแบบผลึกเหลวทำงานเสร็จเรียบร้อย จากนั้นทำการเคลียร์ค่าในทุกๆ ตำแหน่งแอดเดรสของจอแสดงผลแบบผลึกเหลว

3.2.3 เขียนโปรแกรมในส่วนของเครื่องคอมพิวเตอร์จะได้โฟล์วชาร์ทดังนี้

รูปที่ 3.6 โฟล์วชาร์ทแสดงการทำงานของ โปรแกรมรับข้อมูลจากบัตรคิวพร้อมส่งเรียกบอกหมายเลขบัตรมาที่เคาน์เตอร์บริการ

จากรูปที่ 3.6 เป็นการเขียนโฟล์วชาร์ทการทำงานของโปรแกรมรับข้อมูลหมายเลขจากบัตรสมาร์ดการ์ด ขั้นตอนแรกก็รับข้อมูลหมายเลขบัตรที่เสียบบัตรเข้ามา จากนั้นนำข้อมูลหมายเลขบัตรมาเก็บไว้ที่คอมพิวเตอร์ซึ่งจะใช้ภาษาวิชวลเบสิกในการเขียน แล้วนำข้อมูลที่รับเข้ามาจัดเรียงลำดับ จากนั้นก็ทำการเรียกลำดับหมายเลขบัตรที่เข้ามาก่อนโดยบอกเป็นหมายเลขที่บันทึกลงบัตรมาที่เคาน์เตอร์บริการของโรงพยาบาลเป็นเสียงเรียกจากซาวด์การ์ดโดยที่ผู้ใช้บริการต้องนำบัตรสมาร์ตการ์ดมาด้วย

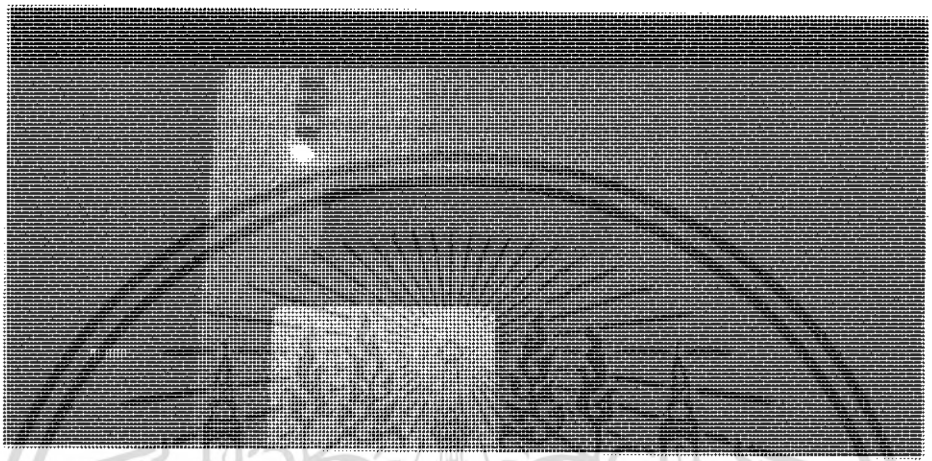
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองเครื่องอ่านเขียนสมาร์ตการ์ด

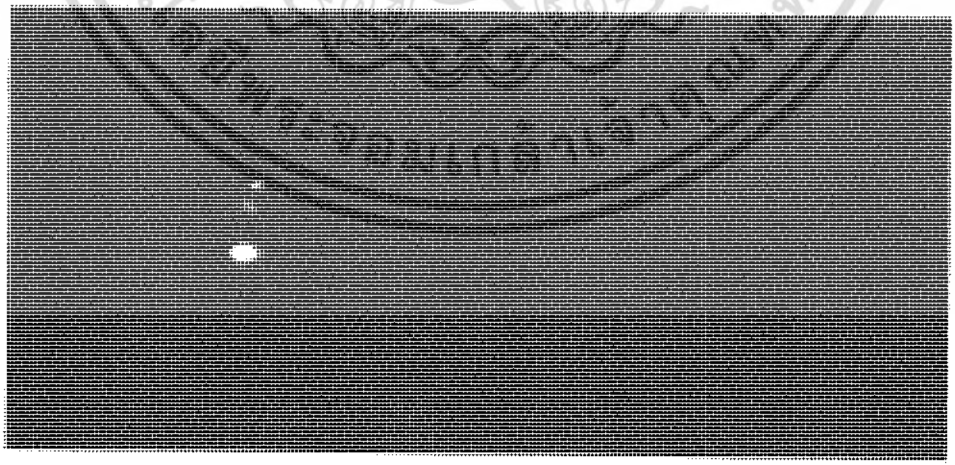
4.1.1 แสดงผลที่จอ LCD เมื่อเครื่องอ่านเขียนยังไม่มีบัตรสมาร์ตการ์ดเสียบเข้ามา



รูปที่ 4.1 แสดงผลเมื่อยังไม่มีการเสียบบัตรสมาร์ตการ์ด

4.1.2 แสดงการเขียนบันทึกข้อมูลหมายเลขของบัตรสมาร์ตการ์ด

เมื่อมีการเสียบบัตรสมาร์ตการ์ดที่เครื่องสมาร์ตการ์ด จากนั้นก็เปิดโปรแกรมที่ใช้เขียนข้อมูลหมายเลขของบัตรสมาร์ตการ์ดซึ่งโปรแกรมที่ช่วยบันทึกจะใช้โปรแกรมวิชวลเบสิกแล้วก็กดเลือกหมายเลขที่ต้องการบันทึกลงบัตร จากนั้นก็กดปุ่มบันทึกหมายเลขบัตรลงบัตรสมาร์ตการ์ด



รูปที่ 4.2 แสดงการบันทึกหมายเลขของบัตรสมาร์ตการ์ด

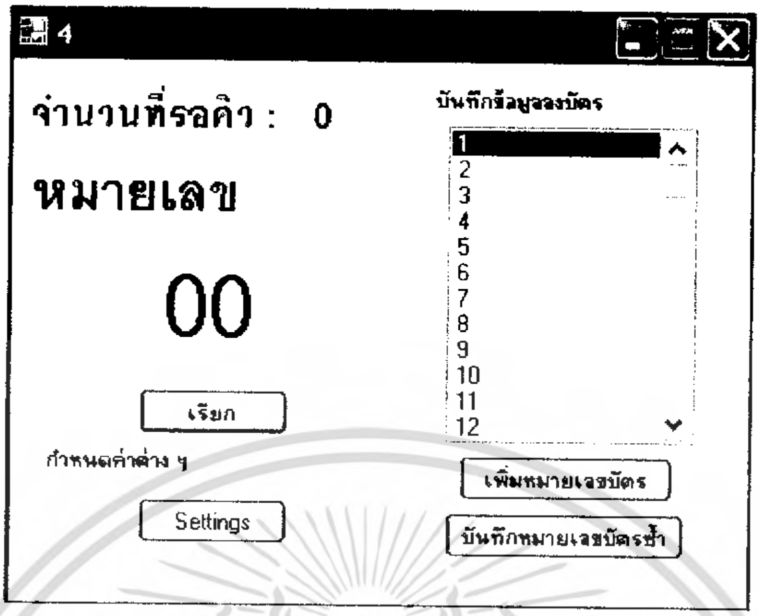
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ขั้นตอนระบบคิวใช้บริการ

หลังจากที่มีการบันทึกหมายเลขลงบัตรสมาร์ตการ์ดแล้วก็ทำการพิมพ์หมายเลขกำกับไว้บนบัตรสมาร์ตการ์ดด้วย เมื่อผู้มาใช้บริการหยิบบัตรคิวแล้วก็นำไปเสียบที่เครื่องอ่านสมาร์ตการ์ดที่จัดเตรียมไว้ ไมโครคอนโทรลเลอร์ที่ควบคุมก็จะนำหมายเลขบัตรคิวมาเก็บค่าพร้อมกับส่งค่าหมายเลขบัตรผ่านพอร์ตอนุกรมไปเก็บไว้ในโปรแกรมวิชวลเบสิกบนคอมพิวเตอร์ ในส่วนของจอแอลซีดีที่เครื่องอ่านสมาร์ตการ์ดก็แสดงเวลาที่รอคิวให้ผู้ใช้บริการทราบด้วย จากนั้นพนักงานที่เคาน์เตอร์บริการก็จะดูที่โปรแกรมบนคอมพิวเตอร์ว่ามีคิวเข้ามาหรือไม่ หากมีก็กดปุ่มเรียกมาใช้บริการที่เคาน์เตอร์บริการ แล้วโปรแกรมก็จะส่งข้อมูลหมายเลขบัตรพร้อมช่องบริการ ไปที่ไมโครคอนโทรลเลอร์โดยผ่านพอร์ตอนุกรมเพื่อให้ไมโครคอนโทรเลอร์ประมวลผลส่งไปแสดงบน seven segment พร้อมส่งเสียงเรียกหมายเลขคิวและช่องบริการ

รูปที่ 4.3 แสดงหน้าจอโปรแกรมเมื่อมีการเสียบบัตรคิวเข้ามา

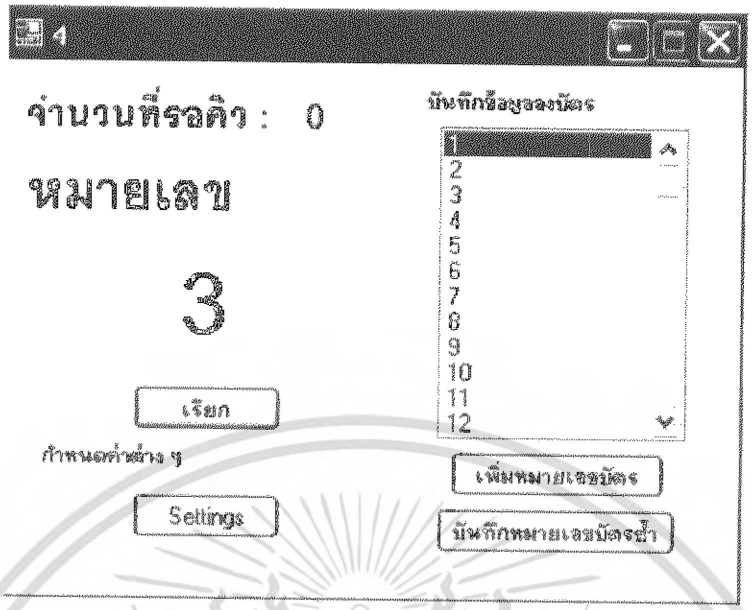
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงหน้าจอโปรแกรมเมื่อยังไม่มีหมายเลขบัตรคิวเข้ามา

รูปที่ 4.5 แสดงหน้าจอโปรแกรมเมื่อมีหมายเลขบัตรคิวเข้ามา

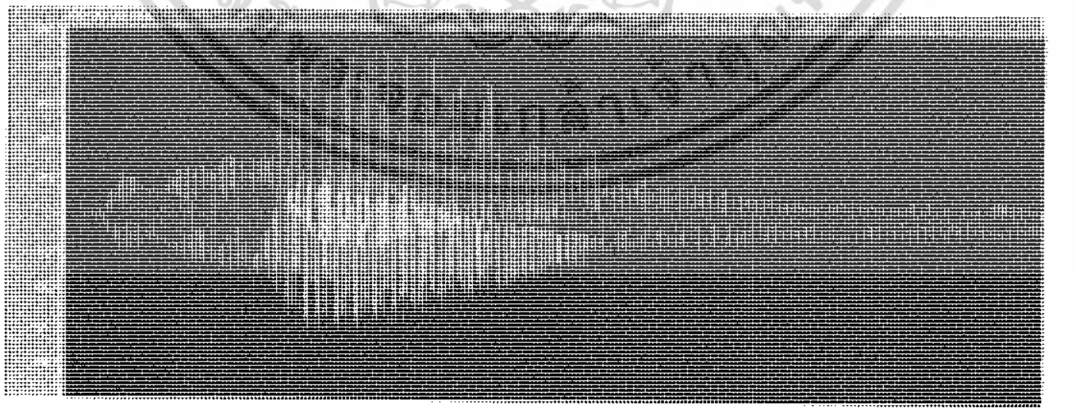
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงหน้าจอโปรแกรมเมื่อกดปุ่มหมายเลขบัตรคิวออกไป

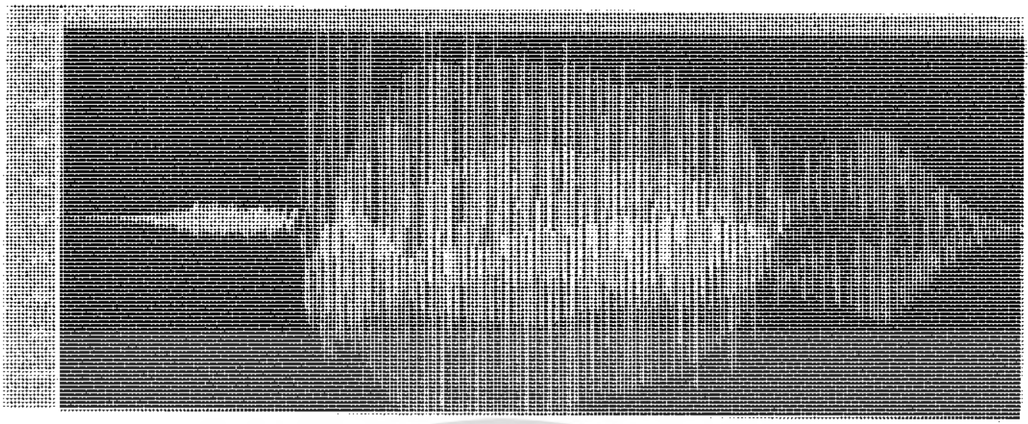
4.3 การทำงานระบบของเสียงพูด

การทำงานของระบบเสียงพูดจะบันทึกเสียงลงในคอมพิวเตอร์และเล่นเสียงผ่านการ์ดเสียง (sound card) ของคอมพิวเตอร์ การบันทึกเสียงนั้นจะบันทึกทีละคำคือ หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก้า สิบ ยี่สิบ สามสิบ สี่สิบ ห้าสิบ หกสิบ เจ็ดสิบ แปดสิบ เก้าสิบ เชิญหมายเลข ที่ช่องบริการค่ะ เมื่อโปรแกรมรับข้อมูลหมายเลขคิวมาประมวลค่าว่าเป็นหมายเลขและช่องบริการใด จากนั้นโปรแกรมก็จะทำการผสมคำพร้อมกับเล่นเสียงออกมา

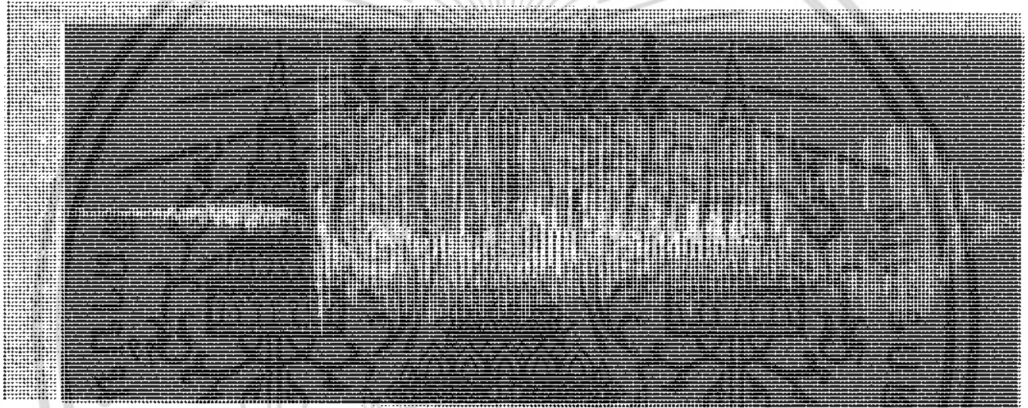


รูปที่ 4.7 แสดงกราฟเสียงพูด "หนึ่ง"

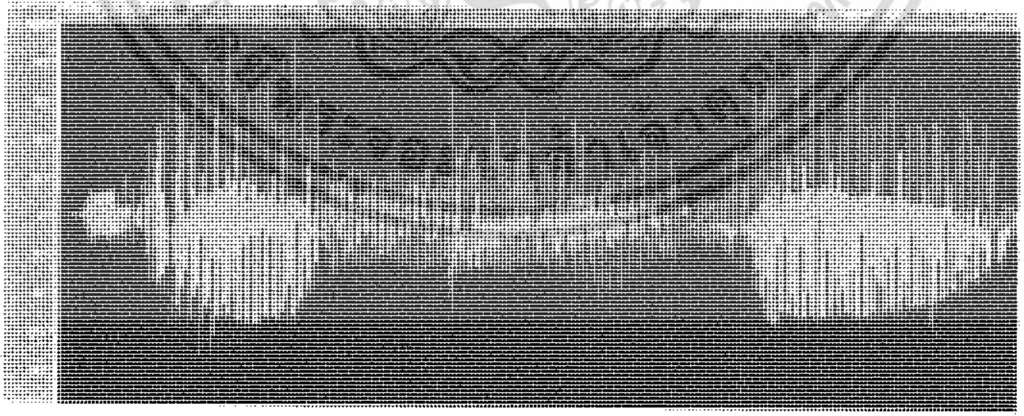
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงกราฟเสียงพูด "สอง"

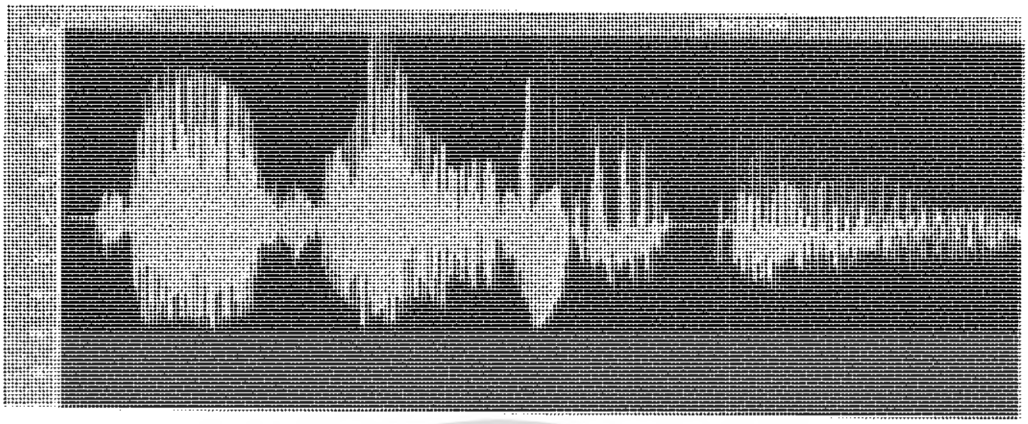


รูปที่ 4.9 แสดงกราฟเสียงพูด "สาม"

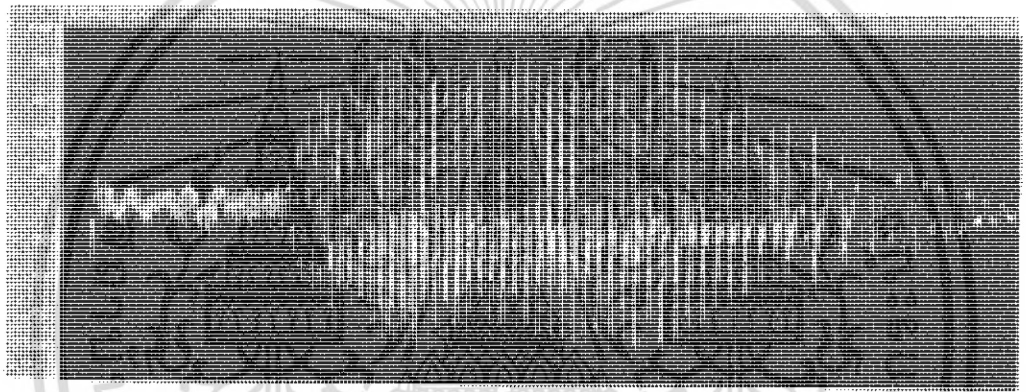


รูปที่ 4.10 แสดงกราฟเสียงพูด "เชิญหมายเลข"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 แสดงกราฟเสียงพูด "ที่ช่องบริการ"



รูปที่ 4.12 แสดงกราฟเสียง "ค่ะ"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

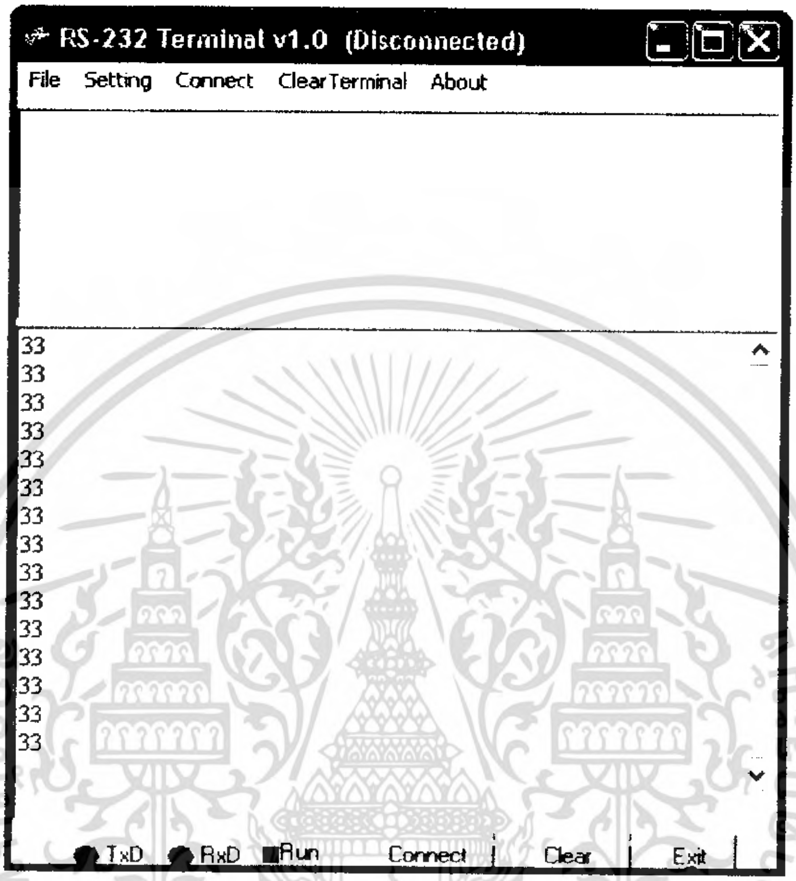
4.4 การแสดงผลที่ seven segment

เมื่อไมโครคอนโทรลเลอร์รับหมายเลขและช่องบริการจากโปรแกรมแล้วก็ทำการประมวลผลจากนั้นก็ส่งออกแสดงผลที่ seven segment ซึ่งประกอบไปด้วยหมายเลขคิวและช่องบริการ

รูปที่ 4.13 แสดงผล seven segment

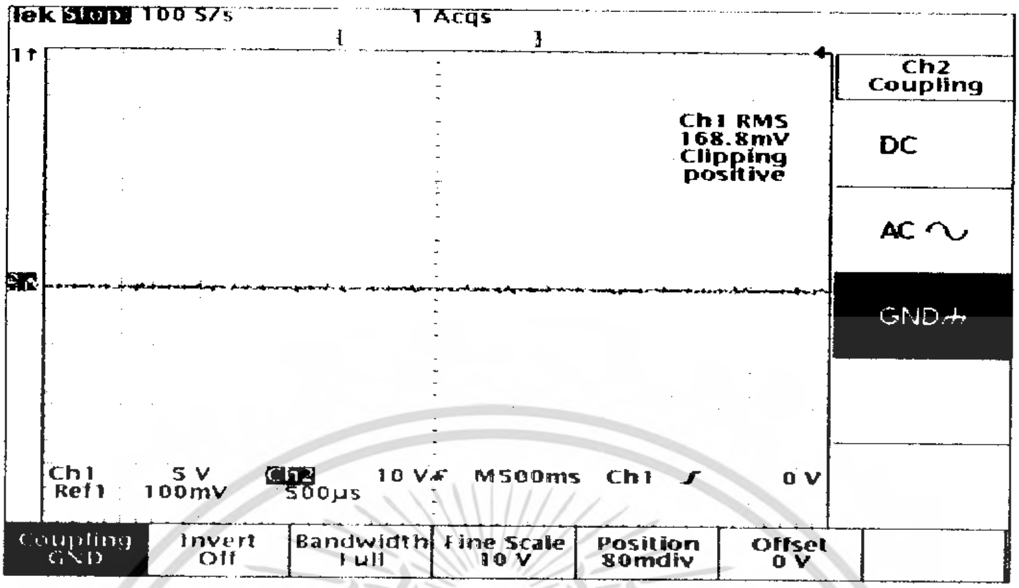
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดลองการรับข้อมูลผ่านพอร์ตอนุกรม RS-232

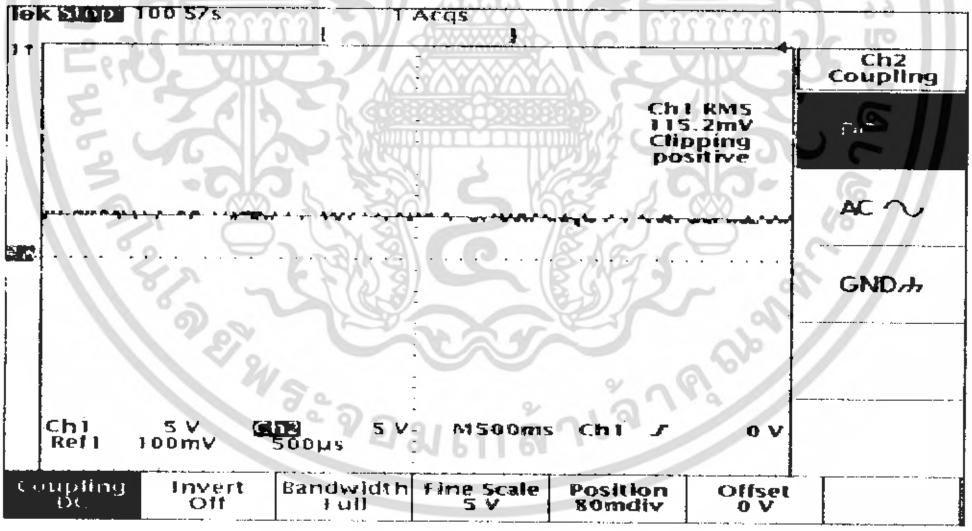


รูปที่ 4.14 การอ่านข้อมูลจากบัตรสมาร์ดการ์ดผ่านพอร์ตอนุกรม RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

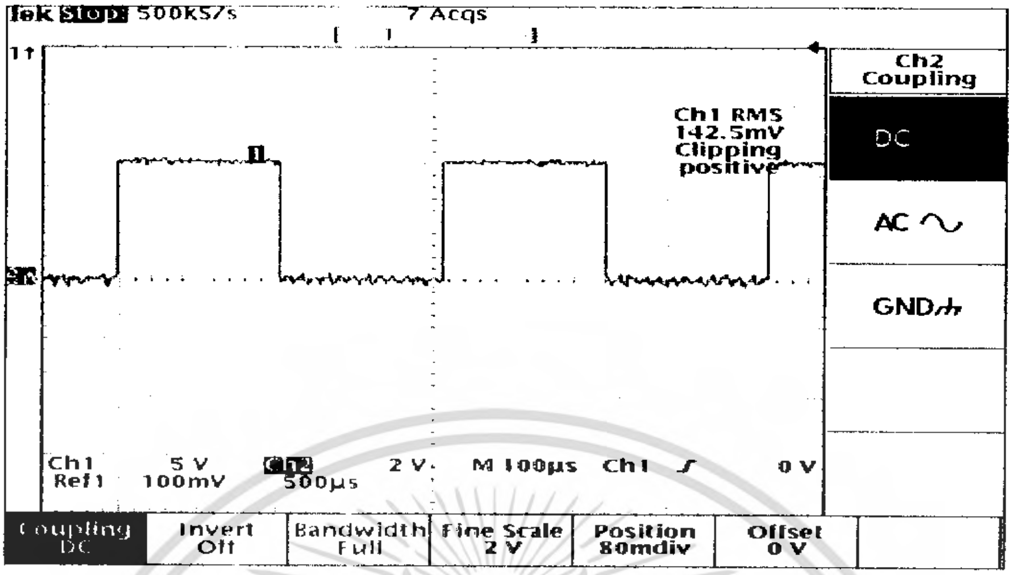


รูปที่ 4.15 การวัดสัญญาณเทียบกับ GROUND



รูปที่ 4.16 การวัดสัญญาณเทียบที่ขา TX ในขณะที่ยังไม่อ่านการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 การวัดสัญญาณเทียบที่ขา TX ขณะที่มีการอ่านข้อมูลในการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่5
สรุปและวิจารณ์

จากผลการทดลองพบว่าสามารถทำงานตามที่จุดประสงค์ที่ตั้งไว้ส่วนเสียงพูดซาวด์การ์ดก็พอสามารถฟังได้แต่ความต่อเนื่องของเสียงอาจไม่ดีเหมือนเสียงธรรมชาติเพราะเกิดจากขั้นตอนบันทึกเสียงเป็นคำๆ แล้วนำเสียงมาผสมกันทำให้เสียงไม่เรียบ และรวมถึงคุณภาพของไมโครโฟนที่ใช้ในการบันทึกเสียงด้วย

ในส่วนของเครื่องอ่านเขียนสมาร์ดการ์ดนั้นมีความซับซ้อนมาก ต้องมีการเปรียบเทียบข้อมูลต่างๆ ก่อนจึงสามารถเข้าถึงหน่วยความจำของบัตรสมาร์ดการ์ดได้ และจึงสามารถบันทึกข้อมูลลงบัตรสมาร์ดการ์ดได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. เลิศ แซ่ตั้ง, เทคโนโลยีสมาร์ตการ์ด, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), 2546.
2. สมยศ จุณณะปิย, การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51, คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง, 2546.
3. อุทัย สุขสิงห์, เอกสารประกอบการสอนวิชาไมโครคอนโทรลเลอร์, คณะวิศวกรรมศาสตร์ มหาวิทยาลัยอุบลราชธานี, 2539-2543.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

			b7	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
			b6	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
			b5	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
			b4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b3	b2	b1	b0																
0	0	0	0				0	@	P	`	p				ฐ	ภ	ะ	เ	๐
0	0	0	1			!	1	A	Q	a	q			ก	ฑ	ม	ั	แ	๑
0	0	1	0			"	2	B	R	b	r			ข	ฒ	ย	า	โ	๒
0	0	1	1			#	3	C	S	c	s			ฃ	ณ	ร	ำ	ใ	๓
0	1	0	0			$	4	D	T	d	t			ค	ด	ฤ	ิ	ไ	๔
0	1	0	1			%	5	E	U	e	u			ฅ	ต	ล	ี	ๅ	๕
0	1	1	0			&	6	F	V	f	v			ฆ	ถ	ฦ	ึ	ๆ	๖
0	1	1	1			'	7	G	W	g	w			ง	ท	ว	ื	็	๗
1	0	0	0			(	8	H	X	h	x			จ	ธ	ศ	ุ	่	๘
1	0	0	1			)	9	I	Y	i	y			ฉ	น	ษ	ู	้	๙
1	0	1	0			*	:	J	Z	j	z			ช	บ	ส		๊	๚
1	0	1	1			+	;	K	[	k	{			ซ	ป	ห		๋	๛
1	1	0	0			,	<	L	\	l	|			ฌ	ผ	ฬ		์	
1	1	0	1			-	=	M	]	m	}			ญ	ฝ	อ		ํ	
1	1	1	0			.	>	N	^	n	~			ฎ	พ	ฮ		๎	
1	1	1	1			/	?	O		o				ฏ	ฟ	ฯ	฿	๏	

รูปที่ 2.4 ตารางแสดงรหัส ASCII แทนตัวอักษรภาษาอังกฤษและภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

Code microcontroller

```
#include <REGX52.H>
#include <STDIO.h>

/********** Port **********/
/********** LED **********/
sbit channel = P3^5;
sbit minor = P3^6;
sbit major = P3^7;
sfr LED = 0xA0; //P2

/********** LCD **********/
sbit E = P1^5;
sbit RS = P1^6;
sfr LCD = 0x80;

/********** Smart Card **********/
sbit IO = P1^0;
sbit RST = P1^1;
sbit CLK = P1^2;
sbit SW = P1^3;
sbit PWR = P1^4;

/********** Variable **********/
unsigned char cnt;

/********** LED **********/
unsigned char ch,ma,mi;
unsigned char seven_seg[10];

/********** LCD **********/
unsigned char min;
bit ena_wcard,ena_wait,ena_inst;

/********** Smart Card **********/
unsigned char SMCBUF[8];
unsigned char wdat;
bit ena_read,ena_write;


/********** Function Prototype **********/
void delay(unsigned int count);
void init();

/********** LED **********/
void init_7segment();
void clrLED();
void print_LED();

/********** LCD **********/
void Enable_LCD();
void ClrScr();
void Screen(unsigned char dat,bit rs);
void init_LCD();
void print_LCD_insrt();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void print_LCD_Write_dat(unsigned char wdat);
void print_LCD(char dat,char num);

/********** Smart Card **********/
void init_SmartCard();
void smcdelay();
void smcstart();
void smcstop();
void smcwbyte(unsigned char x);
unsigned char smcrbyte();
void smcread(unsigned char cmd,add,len);
void smcwrite(unsigned char cmd,add,dat);
unsigned char smcver(unsigned char p1,p2,p3);
void clearbuf();
void ena_clock();

void main()
{
      delay(300);                               // power delay
    init();                                    // start RS232 & init I/O
    delay(200);

      while(1)
      {
            if(!SW)
            {
                  if(ena_read) //Events read Smart Card
                  {
                        clearbuf();
                        ES = 0;
                        TI = 1;
                        smcread(0x30,0x20,1);
                        if(SMCBUF[0] > 0x99)
                              ena_read = 1;
                        else
                        {
                              printf("%c",SMCBUF[0]);
                              ES = 1;
                              ena_read = 0;
                              print_LCD(min,SMCBUF[0]);
                              ena_inst = 1; //Enable Display LCD
                        }
                  }
            }
            else
            {
                  ena_read = 1;
                  ena_wait = 1;
                  print_LCD_insrt();
            }
            print_LED();
      }
}

void comport()interrupt SIO_VECTOR
{
      unsigned char rcv,a;
      if(RI)
      {
            TI = 1;
            rcv = _getkey();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RI = 0;

            if((rcv / 10) == 21) //receive channel
            {
                  rcv -= 210;
                  ch = rcv;
            }
            else if((rcv / 100) == 0) //receive number
            {
                  ma = rcv / 10;
                  mi = rcv % 10;
            }
            else if((rcv / 100) == 1) //wait data to smart card
            {
                  rcv -= 100;
                  min = rcv;
            }
            else if((rcv / 100) == 2) //write data to smart card
            {
                  rcv -= 200;
                  clrLED();
                  delay(100);
                  a = smcver(0xff,0xff,0xff);
                  if(a == 0)
                  {
                        smcwrite(0x38,0x20,rcv);
                        clearbuf();
                        smcread(0x30,0x20,1);
                        print_LCD_Write_dat(SMCBUF[0]);
                  }
                  delay(100);
            }
      }
      if(TI)
            TI = 0;
}

void status_board()interrupt TF0_VECTOR
{
      if(cnt == 100)
      {
            P3_4 = ~P3_4;
            cnt = 0;
      }
      else
            cnt++;
}

/********** Start BASIC FUNCTION **********/
void init()
{
      TCON = 0; //stop timer
      SCON = 0x52; // set speacial function register
      TMOD = 0x20;
      TH1 = 0xFD; // Baud Rate = 9600
      TCON = 0x50;  // start timer
      IE = 0x92; // Enable Interrupt

      init_LCD(); //initial LCD
      init_7segment(); //initial LED
      init_SmartCard(); //initial Smart Card
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

void delay(unsigned int count) // mSec Delay
{
    unsigned char i;
    while(count)
      {
        for(i = 1; i < 114; i++);
        count--;
    }
}

/********** End BASIC FUNCTION **********/

/********** Start LED **********/
void init_7segment()
{
      seven_seg[0] = 0xC0;
      seven_seg[1] = 0xF9;
      seven_seg[2] = 0xA4;
      seven_seg[3] = 0xB0;
      seven_seg[4] = 0x99;
      seven_seg[5] = 0x12;
      seven_seg[6] = 0x82;
      seven_seg[7] = 0xF8;
      seven_seg[8] = 0x80;
      seven_seg[9] = 0x90;
}

void clrLED()
{
      channel = 1;
      major = 1;
      minor = 1;
}

void print_LED()
{
      clrLED();
      major = 0;
      LED = seven_seg[ma];
      delay(2);

      clrLED();
      minor = 0;
      LED = seven_seg[mi];
      delay(2);

      clrLED();
      channel = 0;
      LED = seven_seg[ch];
      delay(2);
}

/********** Stop LED **********/

/********** Start LCD **********/
void Enable_LCD()
{
      E = 1;
      delay(2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
	E = 0;
}

void ClrScr()
{
	Screen(0x01,0);
	delay(5);
}

void Screen(unsigned char dat,bit rs)
{
	RS = rs;
	LCD = dat;
	Enable_LCD();
	delay(5);
}

void init_LCD()
{
	Screen(0x38,0);
	delay(5);
	Screen(0x0C,0);
	delay(5);
	Screen(0x01,0);
	delay(5);
	Screen(0x0F,0);
	delay(5);
	Screen(0x80,0);
	delay(5);

	min = 0;
	ena_inst = 1;
}

void print_LCD_insrt()
{
	code unsigned char str[] = "Insert Card";
	char i;

	if(ena_inst)
	{
		ClrScr();
		for(i = 0;i < 11; i++)
		{
			Screen(str[i],1);
			delay(5);
		}
	}
	ena_inst = 0;
}

void print_LCD_Write_dat(unsigned char wdat)
{
	code unsigned char str[] = "Write Data ";
	code unsigned char str2[] = " Complete";
	char i;
	unsigned char wma,wmi;

	ClrScr();
	for(i = 0;i < 11; i++)
	{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Screen(str[i],1);
            delay(5);
        }
        Screen(0xC0,0);
        delay(5);

        wma = wdat / 10;
        wmi = wdat % 10;

        wma += 0x30;
        wmi += 0x30;

        Screen(wma,1);
        delay(5);
        Screen(wmi,1);
        delay(5);

        for(i = 0;i < 9; i++)
        {
            Screen(str2[i],1);
            delay(5);
        }
}

void print_LCD(char dat,char num)
{
        code unsigned char str2[] = "No. ";
        code unsigned char str[] = " wait";
        code unsigned char str1[] = " minute";
        unsigned char i,ma,mi;

        if(ena_wait)
        {
            ClrScr();
            for(i = 0;i < 4; i++)
            {
              Screen(str2[i],1);
              delay(5);
            }

            ma = num / 10;
            mi = num % 10;

            ma += 0x30;
            mi += 0x30;

            Screen(ma,1);
            delay(5);
            Screen(mi,1);
            delay(5);

            for(i = 0;i < 5; i++)
            {
              Screen(str[i],1);
              delay(5);
            }

            Screen(0xC0,0);
            delay(5);

            ma = dat / 10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mi = dat % 10;

            ma += 0x30;
            mi += 0x30;

            Screen(ma,1);
            delay(5);
            Screen(mi,1);
            delay(5);

            for(i = 0;i < 7; i++)
          {
              Screen(str1[i],1);
              delay(5);
          }
      }
      ena_wait = 0;
}
/********** Stop LCD **********/

/********** Start Smart Card **********/

void init_SmartCard()  // init I/O for smart card
{
    RST = 0;
    CLK = 0;
    IO  = 1;
      ena_read = 1;
}

void ena_clock()
{
      CLK = 1;
      smcdelay();
    CLK = 0;
    smcdelay();
}

void smcdelay() // pluse delay
{
    unsigned char i;
    i = 20;
    while(i > 0)
            i--;
}

void smcstart()// start condition IFD
{
    CLK = 0;
    IO  = 1;
    CLK = 1;
    smcdelay();
    IO  = 0;
    smcdelay();
    CLK = 0;
    smcdelay();
}

void smcstop()// stop condition IFD
{
    IO  = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    CLK = 1;
    smcdelay();
    IO  = 1;
    smcdelay();
    CLK = 0;
    smcdelay();
}

void smcwbyte(unsigned char x)// write 1 byte
{
    unsigned char i;
    for (i = 0; i <= 7; i++)
      {
        IO = (x & 0x1) ? 1 : 0;
        ena_clock();
        x >>= 1;
    }
}

unsigned char smcrbyte()// read 1 byte
{
    unsigned char i,x;
    x = 0;
    for (i = 0; i <= 7; i++)
      {
        IO = 1;
        x >>= 1;
        if(IO)
                  x |= 0x80;
            else
                  x &= 0x7f;
        CLK = 1;
        smcdelay();
        CLK = 0;
        smcdelay();
    }
    return (x);
}

void smcread(unsigned char cmd,add,len) // read data
{
    unsigned char i;                          // output = SMCBUF[]
    smcstart();                               // start
    smcwbyte(cmd);                            // command
    smcwbyte(add);                            // address
    smcwbyte(0);
    smcstop();                                // stop
    for (i = 0; i < len; i++)
        SMCBUF[i] = smcrbyte();

    RST = 1;                                  // reset
      smcdelay();
      ena_clock();
    RST = 0;
}

void smcwrite(unsigned char cmd,add,dat)// write data
{
    smcstart();                               // start
    smcwbyte(cmd);                            // command
    smcwbyte(add);                            // address
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    smcwbyte(dat);                              // data
    smcstop();                                  // stop
    do
      {                                         // wait
        ena_clock();
        IO = 1;
    }
      while(!IO);
}

unsigned char smcver(unsigned char p1,p2,p3)  // verify PSC code
{
    unsigned char mask,i;                                  // out 0=OK
1,2,3=Error
    mask = 1;
    smcread(0x31,0,4);                          // check error counter
(EC)
    if(SMCBUF[0] > 0)
      {
        for(i = 0; i <= 2; i++)
            {
            if (SMCBUF[0] & mask) break;
            mask <<= 1;
        }
        SMCBUF[0] ^= mask;
        smcwrite(0x39,0,SMCBUF[0]);             // write free bit in EC
        smcwrite(0x33,1,p1);                    // compare verification
data
        smcwrite(0x33,2,p2);
        smcwrite(0x33,3,p3);
        smcwrite(0x39,0,0xff);                  // erase EC
        smcread(0x31,0,4);                      // check EC
    }
    mask = 0;                                    // change EC to 1,2,3
    for(i = 0; i <= 2; i++)
      {
        if((SMCBUF[0] & 0x01) == 0)
                  mask++;
          SMCBUF[0] >>= 1;
    }
    return (mask);
}

void clearbuf()
{                        // clear ram buffer
    unsigned char i;
    for (i = 0; i <= 7; i++)
            SMCBUF[i] = 0;
}

/********** End Smart Card **********/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้