สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การประยุกต์ใช้งานบัตรสมาร์ทการ์ดสำหรับอพาร์ตเมนท์

AN APPICATION OF SMART CARD FOR APARTMENT

โดย

นายกิตติศักดิ์ ดลราษี

นายจตุรงค์ ไส้เพี้ย

วพ.
ก6757
2549

เลขหมู่.....
เลขทะเบียน 71966
วัน,เดือน,ปี - 7 มิ.ย. 2550

.b 11761131
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

ผ่านการตรวจรูปเล่มแล้ว

(ลงชื่อ).....ผู้ตรวจ

ผ่านการตรวจชิ้นงานแล้ว

(ลงชื่อ).....ผู้ตรวจ

การประยุกต์ใช้งานบัตรสมาร์ทการ์ดสำหรับอพาร์ตเมนท์

AN APPICATION OF SMART CARD FOR APARTMENT

โดย

นายกิตติศักดิ์ ดลราษี 47015733

นายจตุรงค์ ไส้เพี้ย 47015738

อาจารย์ที่ปรึกษา

รศ. สมยศ จุณณะปิยะ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **การประยุกต์ใช้งานบัตรสมาร์ทการ์ดสำหรับอพาร์ตเมนท์**

AN APPICATION OF SMART CARD FOR APARTMENT

ผู้จัดทำ

1.นายกิตติศักดิ์ ตลราษี 47015733

2. นายจตุรงค์ ไส้เพี้ย 47015738

อาจารย์ที่ปรึกษา

(รศ. สมยศ จุณณะปิยะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประยุกต์ใช้งานบัตรสมาร์ทการ์ดสำหรับอพาร์ตเมนท์

AN APPICATION OF SMART CARD FOR APARTMENT

โดย นายกิตติศักดิ์ ดลรายี 47015733

นายจตุรงค์ ไส้เพี้ย 47015738

อาจารย์ที่ปรึกษา รศ. สมยศ จุณณะปิยะ

บทคัดย่อ

โครงงานนี้เป็นการนำเสนอวิธีการป้องกันความปลอดภัยภายในตัวอาคาร ในเรื่องของการตรวจสอบบุคคลเพื่อป้องกันบุคคลภายนอกที่ไม่ได้รับอนุญาตให้ เข้า หรือ ออก อพาร์ตเมนท์ ที่เรียกว่า "Smart Card" แทนที่ระบบเก่าที่ใช้กุญแจอีกทั้งเพื่อความสะดวกในการพกพา และยากต่อการปลอมแปลง

ABSTRACT

This project presents a security system for apartments to preventing strangers by using "Smart Card" instead of keys because it is more comfortable and difficult to counterfeit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จได้อย่างดี ด้วยคำแนะนำและคำปรึกษาจากรศ. สมยศ จุฉเณะปิยะ ซึ่งเป็นอาจารย์ที่ปรึกษา ข้าพเจ้ารู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์และขอขอบพระคุณเป็นอย่างสูง

ขอขอบคุณเพื่อนๆ พี่ๆ น้องๆ ในภาควิชาวิศวกรรมโทรคมนาคมและสาขาอื่นๆของ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง และ ขอบคุณทุกคนที่ให้คำแนะนำต่างๆและคอยให้กำลังใจเสมอมา

สุดท้ายนี้ที่สำคัญที่สุดข้าพเจ้าขอกราบขอบพระคุณ บิดา มารดา และครอบครัวของข้าพเจ้าที่เป็นกำลังใจและให้การสนับสนุนในทุกเรื่อง ทำให้เข้าพเจ้าสามารถทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงด้วยดี

คุณค่าและประโยชน์อันพึงมาจากปริญญานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบแด่ผู้มีพระคุณทุกท่าน

นาย กิตติศักดิ์ ดลราษี

นาย จตุรงค์ ไส้เพี้ย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทที่ 1 บทนำ....1

1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์....1

1.2 ขอบเขตของปริญญานิพนธ์....1

1.3 ประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์ฉบับนี้....1

1.4 เนื้อหาของปริญญานิพนธ์....1

บทที่ 2 ทฤษฎี....3

2.1 สมาร์ทการ์ดคืออะไร....3

2.2 ประวัติความเป็นมาของสมาร์ทการ์ด....3

2.3 ส่วนประกอบและโครงสร้างของสมาร์ทการ์ด....4

2.3.1 ตัวบัตรพลาสติก....4

2.3.2 หน้าสัมผัสและชิปสมาร์ทการ์ด (Smart card Module)....4

2.3.3 ไมโครคอนโทรลเลอร์....5

2.4 การประกอบสมาร์ทการ์ดโมดูลลงในบัตรพลาสติก....7

2.4.1 การสร้างสมาร์ทการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก (TAB-Tape automated bonding)....7

2.4.2 การสร้างสมาร์ทการ์ดด้วยวิธีการวางสมาร์ทการ์ดโมดูลลงในเนื้อบัตร (Chip-On-Flex)....7

2.4.3 การสร้างสมาร์ทการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวของบัตร (Chip-On-Surface)....8

2.5 รายละเอียดพื้นฐานของสมาร์ทการ์ด....8

2.6 ชนิดของสมาร์ทการ์ด....10

2.6.1 การ์ดหน่วยความจำ (Memory card)....11

2.6.2 การ์ดชนิดโปรเซสเซอร์ (Processor card)....13

2.7 มาตรฐานของสมาร์ทการ์ด....15

2.8 เครื่องอ่านสมาร์ทการ์ด (smart card reader)....15

2.9 บอร์ดไมโครคอนโทรลเลอร์ รุ่น CP-JR51AC2 V2.0....16

2.9.1 บอร์ด CP-JR51AC2 V2.0....17

2.9.2 แหล่งจ่ายไฟ (POWER SUPPLY)....19

2.9.3 สัญญาณนาฬิกา CLOCK....19

2.9.3.1 การทำงานของรีจิสเตอร์ CKCON และผลต่อการทำงานของ Clock ของ CPU....20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

2.9.4 โหมดการทำงานของบอร์ด....22
2.9.5 การสื่อสารอนุกรมแบบ RS232/RS422/RS485 25
2.9.5.1 การสื่อสารอนุกรมแบบ RS232 25
2.9.5.2 การสื่อสารอนุกรมแบบ RS422 26
2.9.5.3 การสื่อสารอนุกรมแบบ RS485 28
บทที่ 3 การออกแบบและการสร้าง 30
3.1หลักการสร้างและการออกแบบบอร์ด JR51AC2V2....30
3.2 หลักการสร้าง Reader....30
3.3 การเชื่อมต่อแบบ RS232....32
3.4 การเขียนโปรแกรม....33
3.5 บล็อคไดอะแกรมของระบบ....33
บทที่ 4 การทดลองและผลการทดลอง....34
4.1 ผลการทดลองการอ่านข้อมูลของบัตร....34
บทที่ 5 บทวิจารณ์และสรุป....35

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 แสดงหน้าที่การทำงานของขาต่างๆ ของบัตรสมาร์ทการ์ด.....15

2.2 แสดงโหมดการทำงานของ CPU หลังการรีเซ็ตแบบปรกติ.....24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	การฝังไมโครโมดูลลงในบัตรพลาสติก	5
2.2	(TAB-Tape automated bonding)	7
2.3	Chip-On-Flex	8
2.4	แสดงรูปบัตรสมาร์ทการ์ด	9
2. 5	แสดงการแบ่งสมาร์ทการ์ดตามชนิดของหน่วยความจำ	11
2.6	แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดหน่วยความจำ	12
2.7	แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดโปรเซสเซอร์	14
2.8	แสดงขาต่างๆ บนหน้าสัมผัสของบัตรสมาร์ทการ์ด	14
2.9	แสดง ลักษณะของบอร์ด CP-JR51AC2 V2.0	18
2.10	แสดงบอร์ด CP-JR51AC2 V 2.0.	18
2.11	ลักษณะโครงสร้างของรีจิสเตอร์ CKCON (ตำแหน่ง 8FH)	21
2.12	แสดงขั้วต่อสัญญาณ RS232 ของบอร์ด CP-JR51AC2 V1.0 & V2.0	26
2.13	แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-JR51AC2 V2.0 เมื่อเลือกเป็น RS422	27
2.14	แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-JR51AC2 V2.0 เมื่อเลือกเป็น RS485	29
3.1	วงจรบอร์ด JR51AC2V2	30
3.2	การทำงานของขา reader ProxID GP8	31
3.3	Access control	32
3.4	บล็อคไดอะแกรมของระบบ	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

1.1 ความเป็นมาของหัวข้อปริญญานิพนธ์

ในโลกทุกวันนี้ มีการพัฒนาเทคโนโลยีด้านต่างๆ โดยที่ล้วนแต่มีจุดมุ่งหมายเพื่อให้มนุษย์มีชีวิตมีความเป็นอยู่ที่ดีขึ้น มีความสะดวกสะบายและปลอดภัยมากยิ่งขึ้น สมาร์ทการ์ด (Smart card) ก็เป็นหนึ่งในการพัฒนาเพื่อเหตุผลนี้เช่นกัน สมาร์ทการ์ดเป็นการพัฒนาความสามารถของบัตร (Card) ที่ใช้ในการเก็บข้อมูลให้มีขนาดของหน่วยความจำที่ใหญ่ขึ้น และมีความปลอดภัยของข้อมูลภายในบัตรสูงขึ้นกว่าบัตรแถบแม่เหล็กแบบเก่า (Magnetic card) ที่ใช้กันอยู่เดิม ทำให้มีแนวโน้มในการที่จะทำบัตรสมาร์ทการ์ดไปใช้ในการเก็บข้อมูลสำคัญต่างๆ มากขึ้นเรื่อยๆ

สำหรับในประเทศไทยนั้นได้เริ่มมีการนำสมาร์ทการ์ดไปใช้ในงานด้านต่างๆ บ้างแล้ว เช่น บัตรประจำตัวประชาชน บัตรโทรศัพท์ บัตรพนักงาน เป็นต้น โดยผู้จัดจึงมีแนวความคิดที่จะประยุกต์การใช้งานบัตรสมาร์ทการ์ดกับประตูทางเข้าอพาร์ทเม้นท์หรือบ้านเรือนทั่วไป เพื่อที่จะให้มีความปลอดภัยมากกว่าระบบเก่าที่ใช้กุญแจ เพื่อความสะดวกในการพกพาและยากต่อการปลอมแปลง

1.2 ขอบเขตของปริญญานิพนธ์

ศึกษาการใช้งานบอร์ด CP-JR51AC2 ตัวบัตรสมาร์ทการ์ด และเขียนโปรแกรมควบคุมสมาร์ทการ์ดรีดเตอร์ ProxID GP8-5 ด้วยไมโครคอนโทรลเลอร์ จึงจะได้เครื่องอ่านสมาร์ทการ์ดที่สมบูรณ์ไปใช้เพื่อความปลอดภัยในอพาร์ทเม้นท์

1.3 ประโยชน์ที่คาดว่าจะได้รับจากปริญญานิพนธ์ฉบับนี้

1.3.1 เขียนโปรแกรม ไมโครคอนโทรลเลอร์ เพื่อควบคุมอุปกรณ์ต่างๆได้

1.3.2 เข้าใจถึงการทำงาน และโครงสร้างของสมาร์ทการ์ด

1.3.3 ได้เครื่องอ่านสมาร์ทการ์ดที่พร้อมจะนำไปประยุกต์ใช้งานกับประตูของอพาร์ทเม้นท์ได้

1.3.4 เข้าใจถึงการทำงานบอร์ดไมโครคอนโทรลเลอร์ รุ่น CP-JR51AC2 V2

1.4 เนื้อหาของปริญญานิพนธ์

สามารถแบ่งได้เป็น 2 ส่วน คือ

1.4.1 ทางด้านฮาร์ดแวร์ (Hard Ware) ประกอบด้วย

1.4.1.1 เครื่องอ่านบัตรสมาร์ทการ์ต

1.4.1.2 บอร์ดไมโครคอนโทรลเลอร์ รุ่น CP-JR51AC2 V2

1.4.2 ทางด้านซอฟต์แวร์ (Soft Ware) ประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4.1.2 โปรแกรมควบคุมการทำงานของเครื่องอ่านสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่2
ทฤษฎี

2.1 สมาร์ทการ์ดคืออะไร

สมาร์ทการ์ดคือบัตรพลาสติกที่มีชิปไอซี (Integrated Circuit) ติดหรือฝังอยู่ในตัวพลาสติกตามมาตรฐาน ISO (International Standrad Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเอง โดยวิธีการเข้ารหัสตามมาตรฐาน DES Algorithm (Data Encryption Standard) เพื่อให้ระบบมีระดับความปลอดภัยสูงขึ้น ด้วยคุณสมบัติสำคัญประการหนึ่งที่ทำให้สมาร์ทการ์ดแตกต่างจากบัตรพลาสติกทั่วไปก็คือขณะทำรายการ(transaction) สมาร์ทการ์ดสามารถทำงานได้ด้วยตัวของมันเอง นั่นคือสมาร์ทการ์ดไม่ต้องมีการติดต่อสื่อสารกับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก ทำให้ประหยัดในเรื่องระบบสื่อสารไปได้มาก

2.2 ประวัติความเป็นมาของสมาร์ทการ์ด

สมาร์ทการ์ดปรากฏขึ้นครั้งแรกในประเทศเยอรมัน ในปี 1968 โดยชาวเยอรมัน (Jurgen Dethloff และ Helmut Grotupp) เป็นผู้คิดค้น แต่ผู้ที่ได้สิทธิบัตรกลับเป็นชาวญี่ปุ่น(Kunnitaka Arimura) ในปี 1970 และมีการจดสิทธิบัตรในชื่อของสมาร์การ์ดโดยชาวฝรั่งเศส (Roland Moreno) ในปี 1974 ในระยะแรกนั้นสมาร์ทการ์ดยังทำงานได้ไม่สมบูรณ์นัก เพราะสมาร์ทการ์ดรุ่นแรกๆ ยังมีปัญหาเล็กน้อยในด้านเทคนิค แม้ว่าสมาร์ทการ์ดจะถือกำเนิดในยุโรป แต่ในระยะแรกสมาร์ทการ์ดกลับไม่ค่อยได้รับความสนใจเท่าที่ควร จนกระทั่งในปี 1984 บริษัท French PTT (postal and Telecommunication Services) ได้นำสมาร์ทการ์ดมาใช้งานเป็นบัตรโทรศัพท์เพื่อป้องกันการโกงค่าโทรศัพท์ ในครั้งนั้นโครงการเป็นโครงการนำร่องโดยมีการนำบัตรแถบแม่เหล็ก บัตรแถบแสงและสมาร์ทการ์ดมาทำการทดลองใช้เปรียบเทียบกัน ซึ่งสมาร์ทการ์ดได้พิสูจน์ให้เห็นว่า คุณลักษณะที่เหนือกว่าบัตรชนิดอื่นๆ ทั้งในเรื่องของความทนทาน ความปลอดภัย ความสวยงาม เป็นผลให้สมาร์ทการ์ดในรูปของบัตรโทรศัพท์มีการนำเอาไปใช้ถึง 60 ล้านใบ (เฉพาะประเทศฝรั่งเศส) และดอกย้ำความสำเร็จกว่าอีกล้านใบ

และในปี 1960 เทคโนโลยีการประมวลผลเพื่อเข้ารหัสข้อมูลของฮาร์ดแวร์และซอฟต์แวร์มีความพร้อมมากขึ้น จึงมีการนำมาใช้ในการเข้ารหัสข้อมูลในสมาร์ทการ์ด ซึ่งแต่เดิมนั้นการเข้ารหัสจะมีการใช้งานเฉพาะในหน่วยงานของทหาร หรือหน่วยงานราชการลับเท่านั้น ด้วยเหตุนี้สามารถทำให้สมาร์ทการ์ดการเข้า-ถอดรหัสข้อมูลได้ด้วยตัวเอง ทำให้การใช้สมาร์ทการ์ดมีความปลอดภัยสูงขึ้นจนสามารถนำมาใช้เป็นบัตรเครดิต หรือ เงินสด ได้อย่างสมบูรณ์แบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ส่วนประกอบและโครงสร้างของสมาร์ทการ์ด

2.3.1 ตัวบัตรพลาสติก

ตามมาตรฐานของ ISO สมาร์ทการ์ดต้องมีขนาด 85.6 x 54 มิลลิเมตร และมีความหนา 0.76 มิลลิเมตร ซึ่งมีไมโครคอนโทรลเลอร์ฝังอยู่ภายใน ขนาดของสมาร์ทการ์ดนี้อิงตาม ID-1 นอกจากนี้ยังมีการแบ่งขนาดของสมาร์ทการ์ดออกเป็นสองขนาด ขนาดเล็กที่สุดจะสะดวกสำหรับโทรศัพท์เคลื่อนที่ จะเรียกสมาร์ทการ์ดนี้ว่า ID-000 ส่วนสมาร์ทการ์ดชนิดที่มีขนาดปานกลางจะเรียกว่า ID-00 ส่วนใหญ่ผู้ผลิตมักนิยมผลิตเพียงขนาด ID-1 จากนั้นจึงวาดโครงร่างให้มีขนาดเท่ากับ ID-000 หรือ ID-00 แล้วจึงตัดออกตามขนาดที่ต้องการจะใช้งาน

สมาร์ทการ์ดเป็นชิปไอซีขนาดเล็กที่ถูกสร้างขึ้นเช่นเดียวกับชิ้นส่วนอิเล็กทรอนิกส์อื่นๆ ที่สร้างจากสารกึ่งตัวนำ นำมาติดลงบนหน้าสัมผัส และทำการฝังลงในเนื้อบัตรพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นตัวบัตรจะใช้พลาสติก 4 ชนิด ได้แก่ PVC (Polyvinyl Cholide), ABS (Acrylonitlie Butadiene Styrene), PC (Polycarbonate) และ PET (Polyethylene Terephathalate) ในประเทศไทยจะใช้บัตรพลาสติก PVC มากเป็นอันดับหนึ่ง ส่วนอันดับสองเป็นบัตรพลาสติกชนิด ABS บัตรพลาสติกชนิด PVC มักนำมาใช้เป็นบัตรเอทีเอ็ม บัตรเครดิต-เดบิต บัตรประจำตัวประชาชน ฯลฯ ส่วนบัตรพลาสติกชนิด ABS ไม่ค่อยพบว่าใช้งานกันมากนักเนื่องจากราคาสูงกว่า และลายที่พิมพ์ลงบนบัตรไม่สวยงามคงทนเท่าบัตรพลาสติกชนิด PVC จะพบก็เพียงบัตรพลาสติกเนื้อผสมโดยใช้พลาสติกชนิด ABS เป็นแกนและฉาบผิวด้วยพลาสติกชนิด PVC แต่ความทนทานของตัวบัตรจะสู้บัตรพลาสติกเนื้อ PVC ล้วนไม่ได้

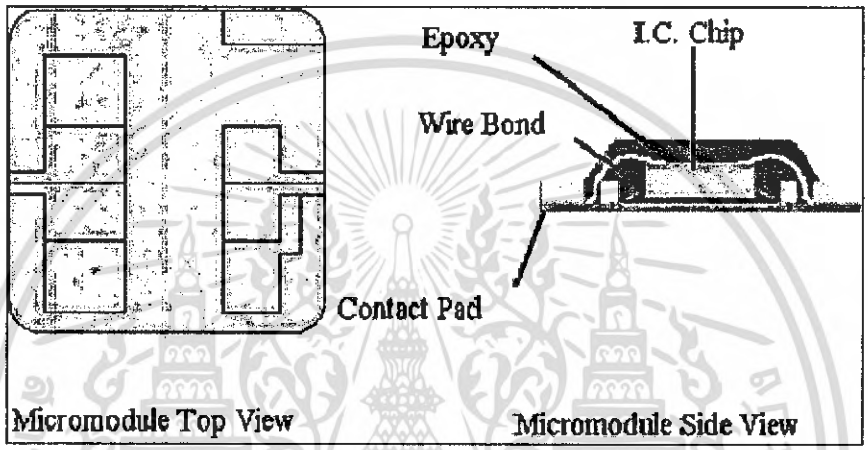
สำหรับบัตรพลาสติกอีก 2 ชนิดที่เหลือ ยังไม่พบว่ามีการใช้งานในประเทศไทย อาจเนื่องจากราคาที่สูงเกินไปของวัสดุที่นำมาใช้ทำเป็นตัวบัตร และคุณสมบัติของวัสดุที่ด้อยกว่าพลาสติกชนิด PVC แต่ข้อเสียที่สำคัญของพลาสติกชนิด PVC ก็ไม่ด้อยไปกว่าข้อดี นั่นก็คือไม่สามารถย่อยสลายได้ตามธรรมชาติ และไวนิลคลอไรด์ซึ่งเป็นส่วนประกอบของ PVC เป็นสารก่อมะเร็ง เมื่อถูกเผาไหม้จะมีการปลดปล่อยกรดไฮโดรคลอริกออกมา จึงเป็นอันตรายต่อสิ่งแวดล้อม

2.3.2 หน้าสัมผัสและชิปสมาร์ทการ์ด (Smart card Module)

สมาร์ทการ์ดโมดูล หรือ หน้าสัมผัสและชิปสมาร์ทการ์ด คือ ส่วนที่แสดงความเป็นตัวตนของสมาร์ทการ์ดที่ชัดเจนที่สุด สมาร์ทการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่าคือสมาร์ทการ์ดที่มีการฝังชิปเอาไว้ในเนื้อบัตร ดังนั้นการที่จะระบุว่าบัตรใบใดเป็นบัตรสมาร์ทการ์ดนั้น ต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก ซึ่งต้องใช้ประสบการณ์ที่เกี่ยวกับสมาร์ทการ์ดพอสมควร แต่ในที่นี้จะแนะนำให้เห็นภาพลักษณ์ที่ชัดเจนของสมาร์ทการ์ดเป็นหลัก ซึ่งก็คือส่วนของสมาร์ทการ์ดโมดูลนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการผลิตสมาร์ทการ์ดโมดูล ส่วนที่เป็นหน้าสัมผัสของสมาร์ทการ์ดประกอบด้วยโลหะหลายชิ้นประกอบกัน แต่ละส่วนจะถูกยึดด้วยแถบฟิล์มบางๆ ทางด้านหลังของสัมผัสเพื่อให้คงรูปอยู่ได้ แถบฟิล์มนี้จะมีการเจาะช่องเล็กๆ สำหรับการเชื่อมต่อสายนำสัญญาณกับชิปสมาร์ทการ์ดกับหน้าสัมผัส หลังจากที่วางชิปสมาร์ทการ์ดลงในตำแหน่งที่ต้องการและเชื่อมต่อสายนำสัญญาณจากชิปสมาร์ทการ์ดเข้ากับหน้าสัมผัสเรียบร้อยแล้ว ขั้นตอนสุดท้ายจะเป็นการผนึกชิปสมาร์ทการ์ดเพื่อป้องกันตัวชิป และสายนำสัญญาณต่างๆ จากสิ่งแวดล้อมภายนอก (เป็นการทดสอบขั้นต้น) ส่วนขั้นตอนที่เหลือจะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติก และทดสอบการทำงานของชิปขั้นสุดท้าย



รูปที่ 2.1 การฝังไมโครโมดูลลงในบัตรพลาสติก

ที่มา : http://www.smartcardalliance.org/industry_info/smart_cards_primer.cfm

2.3.3 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์คือคอมพิวเตอร์รูปแบบหนึ่ง ซึ่งมีขนาดเล็กและเน้นการใช้งานในแบบ Embedded (ฝังตัว) คืออยู่ในรูปของบอร์ดที่อยู่ภายในสินค้าต่าง ๆ อีกที มักจะเรียกกันว่า "แผงวงจร" ความจริงบอร์ดเหล่านี้อาจจะเป็นเพียงวงจรอิเล็กทรอนิกส์ธรรมดา หรืออาจเป็นวงจรในรูปแบบดิจิตอล ก็ได้ แต่ในปัจจุบันจะพบบอร์ดในรูปแบบไมโครคอนโทรลเลอร์มากขึ้น เนื่องด้วยการทำงานที่มีความยืดหยุ่นสูงง่ายต่อการปรับปรุงแก้ไข ใช้ได้ดีกับงานที่มีความซับซ้อนมาก และที่สำคัญก็คือ บอร์ดไมโครคอนโทรลเลอร์จะเปลี่ยนแนวทางการพัฒนาจากที่เน้นด้านฮาร์ดแวร์ไปเป็นเน้นด้านซอฟแวร์แทน ตัวอย่างงานเหล่านี้ได้แก่ บอร์ดเก็บข้อมูลเครื่องตอกบัตร บอร์ดควบคุมแผงแสดงตัวอักษร บอร์ดเครื่องนับจำนวน บอร์ดตั้งเวลา

ไมโครคอนโทรลเลอร์ของสมาร์ทการ์ดแตกต่างจากไมโครคอนโทรลเลอร์ใน PC ทั่วไป ซึ่งส่วนประกอบเบื้องต้นได้แก่ CPU ตำแหน่งของข้อมูลและหน่วยความจำ รวมทั้ง RAM ROM และ EEPROM ชิปเหล่านี้ต่างจากชิปทั่วไปเนื่องจาก ส่วนประกอบต่างๆ เหล่านี้ต้องมีในคอมพิวเตอร์ที่แท้จริง แต่ชิปบางชนิดไม่จำเป็นสำหรับสมาร์ทการ์ด นอกจากนี้ยังต้องมีขนาดเล็กเท่าที่จะเล็กได้เนื่องจากมีความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไปได้ที่จะแตกหักเมื่อมีการโค้งงอ โปรเซสเซอร์คือ CISC และมีหน่วยความจำระหว่าง 6 – 30 กิโลไบต์ อย่างไรก็ตาม ต้องมีการจำกัด RAM และ EEPROM เท่าที่จะทำได้เนื่องจากเกี่ยวข้องกับขนาดต่อบิต อาจมีการนำ flash EEPROM และ FRAM (ferroelectric random-acccss memory) มาใช้ในอนาคต

ไมโครคอนโทรลเลอร์ถูกสร้างจากผู้ผลิตชิพมากมายหลากหลายรูปแบบ ตระกูลของชิพที่ถือได้ว่าเป็นมาตรฐานก็คือ MCS-51 ซึ่งเริ่มต้นมาจาก Intel (ผู้ผลิตไมโครโปรเซสเซอร์รายใหญ่ให้กับวงการคอมพิวเตอร์ PC ในปัจจุบัน) โดยในปัจจุบันนี้ มีผู้ผลิตชิพในตระกูลนี้มากกว่าสิบราย และยังเพิ่มคุณสมบัติต่าง ๆ เข้าไปอีกมากมายนับไม่ถ้วน ในขณะที่เนื้อหาหลักทางด้านโปรแกรมจะเหมือนกันทุกประการ นั่นหมายถึงว่า การเรียนรู้โครงสร้างพื้นฐานและชุดคำสั่งของ MCS-51 จะสามารถนำไปใช้ได้ต่อเนื่องอีกยาวนาน เรื่องทางด้านไมโครคอนโทรลเลอร์ยังคงต้องสนใจในระดับพื้นฐาน เพราะการออกแบบจะยังเกี่ยวของกับตัวบอร์ด โดยมีเนื้อหาทั้งทางด้านฮาร์ดแวร์และ ซอฟแวร์อยู่ (แม้จะเน้นการพัฒนาในเชิงซอฟแวร์ก็ตาม)

ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ที่ผลิตโดยบริษัท Intel ได้มีการนำไปใช้งานอย่างแพร่หลาย โดยได้ทำการเพิ่มประสิทธิภาพและหน่วยการทำงานต่างๆ มากขึ้น ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ ขึ้นอยู่กับโครงสร้างภายใน เช่น บางเบอร์มีหน่วยความจำแบบ ROM บางเบอร์เป็นแบบ EPROM หรือบางเบอร์ไม่มีหน่วยความจำภายใน เป็นต้น อย่างไรก็ตามลักษณะขาต่างๆ จะเหมือนกัน

คุณลักษณะพื้นฐานของ MCS-51

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

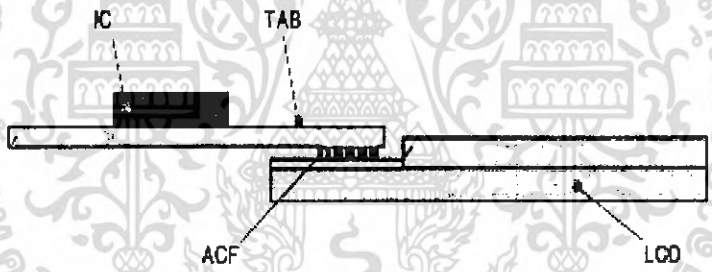
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การประกอบสมาร์ทการ์ดโมดูลลงในบัตรพลาสติก

การประกอบสมาร์ทการ์ดโมดูลลงในบัตรนั้นมีหลายวิธีด้วยกันตามแต่ชนิดของสมาร์ทการ์ดโมดูล และการเตรียมบัตรพลาสติก ซึ่งการเตรียมบัตรพลาสติกจะนำมาใส่สมาร์ทการ์ดโมดูลมีด้วยกัน 2 แบบ คือ บัตรพลาสติกที่ถูกขุดหลุมบนบัตร และบัตรพลาสติกที่เกิดจากการทับซ้อนของชั้นพลาสติกที่เจาะช่องมาแล้ว โดยสมาร์ทการ์ดโมดูลจะใช้บัตรพลาสติกที่มีการเตรียมการมาแล้วดังนี้

2.4.1 การสร้างสมาร์ทการ์ดด้วยวิธีทับซ้อนของแผ่นพลาสติก (TAB-Tape automated bonding)

สมาร์ทการ์ดชนิดนี้เกิดจากการทับซ้อนของพลาสติกตั้งแต่ 3 ชั้นขึ้นไป โดยแต่ละชั้นจะมีการเจาะช่องตามขนาดหน้าสัมผัสและชิปสมาร์ทการ์ดไว้อยู่แล้ว ส่วนที่เป็นหน้าสัมผัสและชิปจะถูกแทรกอยู่ในชั้นในของพลาสติก เมื่อวางซ้อนกันเรียบร้อยแล้วก็จะทำการอัดแต่ละชั้นด้วยความร้อน เมื่อความร้อนถึงจุดที่ทำให้พลาสติกแต่ละชั้นประสานตัวเอง ก็จะนำมาตัดแต่งขอบบัตรและทำการทดสอบการทำงานของชิปเป็นขั้นตอนสุดท้าย



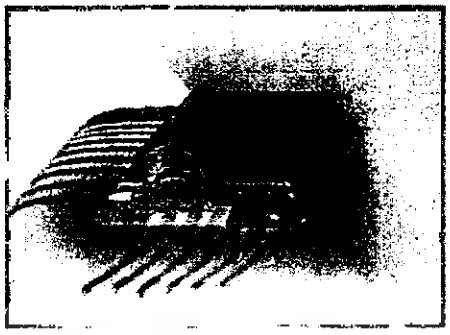
รูปที่ 2.2 (TAB-Tape automated bonding)

ที่มา : www.e-mediana.com/eng/eng/tech/lcd_tech.htm

2.4.2 การสร้างสมาร์ทการ์ดด้วยวิธีการวางสมาร์ทการ์ดโมดูลลงในเนื้อบัตร (Chip-On-Flex)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 Chip-On-Flex

ที่มา : http://www.semicorp.com/Articles/Flip%20Chip%20on%20Flex.htm

2.4.3 การสร้างสมาร์ทการ์ดด้วยวิธีการสร้างหน้าสัมผัสบนผิวของบัตร (Chip-On-Surface)

สมาร์ทการ์ดโมดูลชนิดติดบนผิวของบัตร ผลิตโดยการใช้แสงเลเซอร์ทำการขุดหลุมบนบัตรพลาสติกขนาดเท่ากับตัวชิปสมาร์ทการ์ด วางชิปสมาร์ทการ์ดลงในตำแหน่งที่กำหนด สร้างหน้าสัมผัสและเชื่อมสายสัญญาณกับชิปสมาร์ทการ์ดด้วยหมึกนำไฟฟ้า สุดท้ายพิมพ์ทับส่วนที่เป็นชิปและหมึกนำไฟฟ้าส่วนที่เป็นสายสัญญาณด้วยหมึกที่มีคุณสมบัติเป็นฉนวนไฟฟ้าเพื่อป้องกันวงจรภายในโดยปล่อยส่วนที่เป็นหน้าสัมผัสที่สร้างจากหมึกนำไฟฟ้าไว้เท่านั้น สมาร์ทการ์ดชนิดนี้ไม่ค่อยมีให้เห็นมากนักเนื่องจากต้องใช้เทคโนโลยีในการผลิตที่สูงกว่าหน้าสัมผัสแบบอื่น ทำให้ต้นทุนการผลิตสูงตามไปด้วย อีกทั้งความทนทานก็ยังน้อยกว่าสมาร์ทการ์ดโมดูลชนิดอื่นๆ

2.5 รายละเอียดพื้นฐานของสมาร์ทการ์ต

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.4 แสดงรูปบัตรสมาร์ทการ์ด

ที่มา : http://www.silaresearch.com/acc/proxid1.jpg

สมาร์ทการ์ดมีพื้นฐานมาจากระบบไมโครโปรเซสเซอร์ ซึ่งมีแนวคิดเริ่มแรกจากการนำชิปหน่วยความจำ (EEPROM) มาฝังลงในบัตรพลาสติก โดยมีหน้าสัมผัสเป็นขาเชื่อมต่อกับระบบภายนอก ในการเชื่อมต่อต้องมีการป้อนกระแสไฟฟ้าให้ชิปหน่วยความจำสามารถทำงานได้ การสั่งงานเพื่ออ่านหรือเขียนข้อมูลจากชิปหน่วยความจำสมาร์ทการ์ด ก็ทำได้โดยการเชื่อมต่อสัญญาณผ่านหน้าสัมผัสที่กำหนดไว้แล้ว ในการเชื่อมต่อขาสัญญาณของชิปหน่วยความจำแบบธรรมดา อาจไม่เหมาะสมสำหรับบัตรพลาสติกขนาดเล็ก เนื่องจากจำนวนขาสัญญาณของของหน่วยความจำ (Bus) มีจำนวนไม่น้อยทีเดียว ยิ่งหน่วยความจำที่มีความจุสูงๆ ยิ่งต้องใช้สัญญาณอ้างอิงตำแหน่งของข้อมูล (Address Bus) มากขึ้น จึงมีการนำเอาระบบสื่อสารแบบซิงเกิลบัสมาใช้ในการรับส่งข้อมูล โดยในการนำเอาระบบสื่อสารแบบอนุกรมมาใช้จำเป็นต้องมีการป้อนสัญญาณนาฬิกาเพื่อกำกับจังหวะการรับ-ส่งข้อมูลแต่ละบิต ทำให้ต้องมีหน้าสัมผัสสำหรับสัญญาณนาฬิกาบนชิปสมาร์ทการ์ดเพื่อขึ้นมา แต่ก็นับว่าทำให้ขาเชื่อมต่อลดลงไปไม่น้อยทีเดียว ด้วยเหตุนี้จึงสมาร์ทการ์ดชนิดหน่วยความจำจึงเป็นสมาร์ทการ์ดชนิดแรกที่ถูกสร้างขึ้น

การนำเอาชิปหน่วยความจำมาใส่ในบัตรพลาสติก ทำให้เกิดข้อดีเหนือบัตรแถบแม่เหล็กด้วยความจุข้อมูลที่มากกว่า ไม่มีผลต่อสนามแม่เหล็กไฟฟ้า และรอยขีดข่วน ทำให้สมาร์ทการ์ดโดดเด่นกว่าบัตรแถบแม่เหล็กอย่างเทียบกันไม่ได้ แต่ข้อเสียประการหนึ่งของการใช้หน่วยความจำเพียงอย่างเดียวคือสามารถทำการอ่านและเขียนข้อมูลได้อย่างอิสระเช่นเดียวกับบัตรแถบแม่เหล็ก จึงถือได้ว่าความปลอดภัยของข้อมูลเกือบเป็นศูนย์ นั่นก็คือ ข้อมูลภายในสมาร์ทการ์ดชนิดนี้ไม่เป็นความลับ ด้วยเหตุนี้จึงมีการเพิ่มวงจรสำหรับป้องกันลงไปอีก เพื่อให้ผู้ออกบัตร (Card Issue) สามารถกำหนดสิทธิในการเข้าถึงข้อมูลแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบด์ด้วยวงจรฟิวส์แมทริกธรรมดาๆ ที่เมื่อกำหนดเงื่อนไขไปแล้วจะไม่สามารถแก้ไขได้อีก ต่อมาเมื่อเทคโนโลยีทางด้านเซมิคอนดักเตอร์สูงขึ้น จึงมีการออกแบบวงจรที่สามารถกำหนดเป็นกุญแจรหัส (PIN) สำหรับเข้าถึงข้อมูลในบัตร ซึ่งต้องทำการแสดงกุญแจรหัสทุกครั้งที่บัตรเริ่มทำงาน เพื่อป้องกันการเจาะระบบอีกชั้นหนึ่ง อีกทั้งกุญแจรหัสก็ยังสามารถเปลี่ยนแปลงได้อีกด้วย

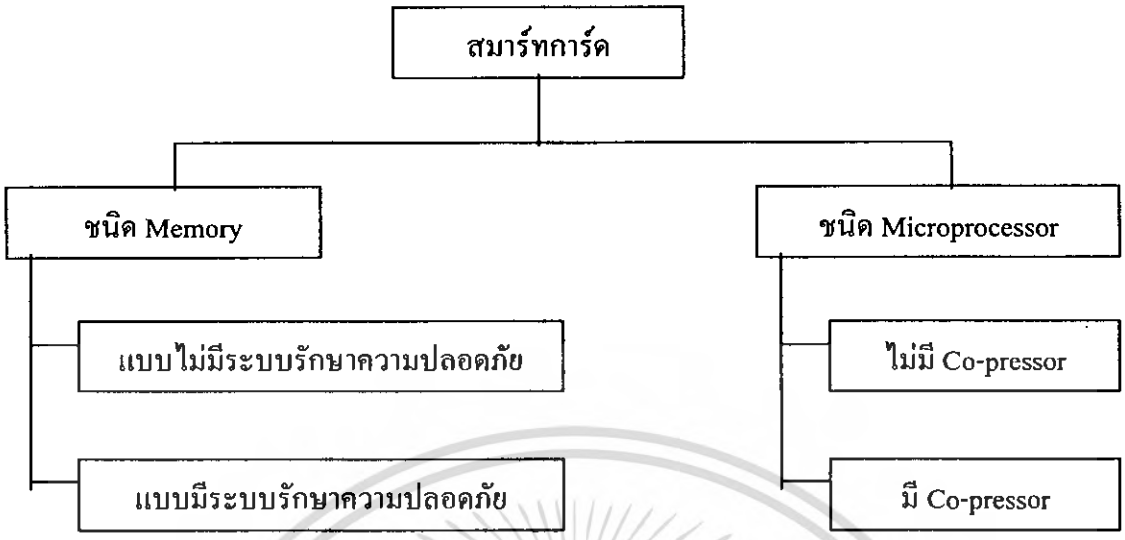
ต่อมามีการนำเอาไมโครโปรเซสเซอร์ (ที่จริงแล้วเป็นไมโครคอนโทรลเลอร์) มาใส่ลงในสมาร์ทการ์ด ทำให้เกิดเป็นสมาร์ทการ์ดชนิดใหม่ที่มีความซับซ้อนยิ่งขึ้น การเข้าถึงข้อมูลไม่สามารถทำได้โดยตรงเหมือนอย่างสมาร์ทการ์ดชนิดหน่วยความจำ การใช้งานสมาร์ทการ์ดชนิดนี้ต้องเขียนขึ้นเป็นชุดคำสั่ง และส่งให้กับชิปไมโครโปรเซสเซอร์ทำงานแทน การที่ใส่ชิปไมโครโปรเซสเซอร์ลงไปในสมาร์ทการ์ด ทำให้ต้องมีการเพิ่มส่วนของหน่วยความจำโปรแกรม (OS-Operating System) สำหรับไมโครโปรเซสเซอร์เพื่อให้ไมโครโปรเซสเซอร์สามารถทำการประมวลผลคำสั่งต่างๆ และสามารถโปรแกรมการเข้าถึงข้อมูลทำให้ช่องโหว่ที่สำคัญของสมาร์ทการ์ดได้รับการแก้ไขจนเกือบสมบูรณ์แบบ

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

2.6 ชนิดของสมาร์ทการ์ด

การแบ่งชนิดของสมาร์ทการ์ดในปัจจุบันอาจทำได้ยาก เนื่องจากมีการใส่เทคโนโลยีใหม่ๆ ลงสมาร์ทการ์ดตลอดเวลา ถ้าจะแบ่งตามชนิดของหน่วยความจำภายในอาจไม่ชัดเจนนัก ยิ่งแบ่งตามลักษณะการเชื่อมต่อก็คงไม่ครอบคลุมสมาร์ทการ์ดทั้งหมด ดังนั้นจึงแบ่งชนิดของสมาร์ทการ์ดให้เข้าใจได้ง่าย ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2. 5 แสดงการแบ่งสมาร์ทการ์ดตามชนิดของหน่วยความจำ

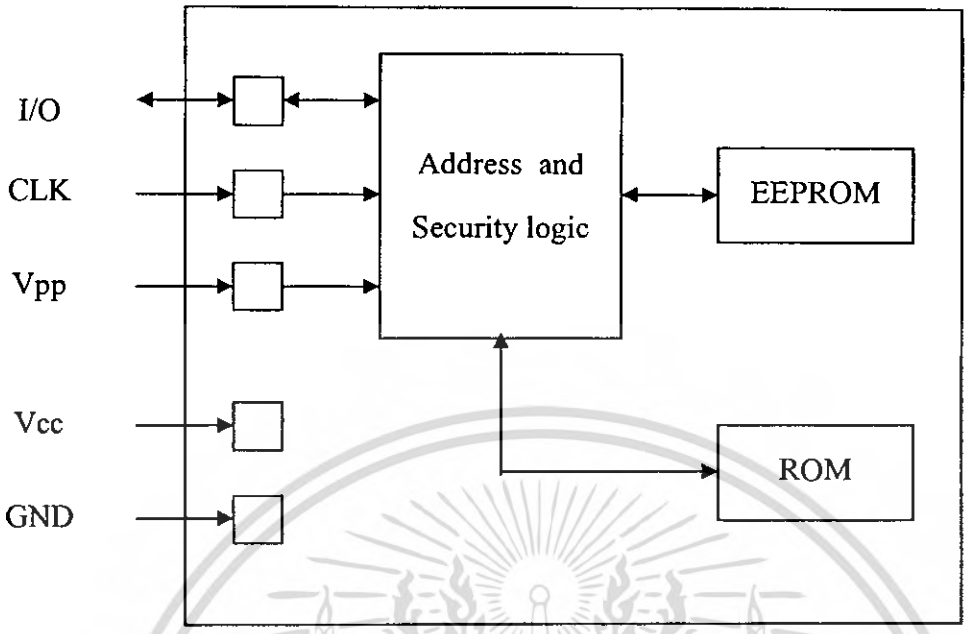
จากรูปที่ 2.5 จะเห็นได้ว่าเราสามารถแบ่งสมาร์ทการ์ดจากโครงสร้างภายในได้ 2 ชนิดก็คือ สมาร์ทการ์ดชนิดหน่วยความจำ (Memory Card) และสมาร์ทการ์ดชนิดไมโครโปรเซสเซอร์ (Processor Card) ซึ่งชิปทั้งสองแบบจะมีหน้าสัมผัสเหมือนกัน แต่สัญญาณที่ต้องป้อนให้แก่หน้าสัมผัสบางหน้าสัมผัสจะไม่มีการใช้งานในสมาร์ทการ์ดต่างชนิดกัน เช่น แรงดันไฟฟ้าสำหรับการเขียนข้อมูลลงในชิป (Vpp) จะมีใช้ในสมาร์ทการ์ดชนิดหน่วยความจำเท่านั้น สัญญาณนาฬิกาสำหรับป้อนให้ชิปทำงาน (CLK) ต้องป้อนให้กับชิปทำงานได้ เพราะภายในชิปสมาร์ทการ์ดไม่มีวงจรสำหรับสร้างสัญญาณนาฬิกา แต่หน้าสัมผัส I/O จะมีการรับ-ส่งข้อมูลที่แตกต่างกันในเรื่องของความถี่ และวิธีการควบคุมจังหวะการรับ-ส่งของข้อมูลแต่ละชนิด

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

2.6.1 การ์ดหน่วยความจำ (Memory card)

สมาร์ทการ์ดชนิดหน่วยความจำ (Memory) หรืออีกชื่อหนึ่งคือ Synchronous card เนื่องจากสมาร์ทการ์ดชนิดนี้มีการรับ-ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิป (ข้อมูลแต่ละบิตที่ส่งให้แก่ชิปต้องสัมพันธ์กับสัญญาณนาฬิกา) สมาร์ทการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วยส่วนวงจรสำหรับการติดต่อสื่อสารกับภายนอก หน่วยความจำข้อมูล และหน่วยความจำสำหรับเก็บชุดคำสั่งของสมาร์ทการ์ดดังรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดหน่วยความจำ

สมาร์ทการ์ดที่เป็นพื้นฐานของสมาร์ทการ์ดในปัจจุบัน ก็คือสมาร์ทการ์ดชนิด Free Access Memory สมาร์ทการ์ดชนิดนี้เปิดโอกาสให้อ่านหรือเขียนข้อมูลในแอดเดรสใดๆ ก็ได้ตามชื่อของสมาร์ทการ์ดชนิดนี้ ไม่มีการป้องกันข้อมูลใดๆ ภายในสมาร์ทการ์ดชนิดนี้ ซึ่งแน่นอนว่าเป็นสมาร์ทการ์ดที่มีความปลอดภัยต่ำที่สุด ถึงกระนั้นการอ่านข้อมูลก็ไม่ใช่เรื่องง่ายนักเมื่อมีการออกแบบหน่วยความจำข้อมูลให้มีการสลับตำแหน่งของบิตข้อมูล โดยมีวงจรควบคุมการสลับตำแหน่งของบิตเป็นส่วนป้องกันข้อมูลอีกต่อหนึ่ง ดังนั้นการอ่านข้อมูลออกแบบธรรมดาจะไม่ได้ข้อมูลที่ถูกต้องหากไม่ติดต่อกับวงจรควบคุมการสลับตำแหน่งของบิตโดยตรง

นอกจากนี้สมาร์ทการ์ดชนิดหน่วยความจำแบบธรรมดา ยังมีการใส่วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลลงไปด้วย ทำให้สามารถกำหนดเงื่อนไขการอ่าน-เขียนข้อมูลได้ทุกไบต์ โดยสมาร์ทการ์ดที่มีวงจรป้องกันการอ่าน-เขียนชนิดนี้ถูกเรียกว่า PIN Protect Memory เนื่องจากการเข้าถึงข้อมูลจะต้องแสดงรหัสผ่านให้บัตรทราบก่อนจึงจะสามารถเข้าถึงข้อมูลได้ วงจรกำหนดเงื่อนไขการอ่าน-เขียนจะมีบิตพิเศษที่มีชื่อว่า Bit Protect ซึ่งเป็นบิตข้อมูลที่ฝากไว้กับข้อมูลให้เป็นบิตที่ 9 แต่ไม่สามารถแก้ไขด้วยคำสั่งเขียนข้อมูลธรรมดา เพราะ Bit Protect ไม่ได้เป็นส่วนหนึ่งของข้อมูลจริงๆ ในการแก้ไข Bit Protect นี้จะสามารถทำการเปลี่ยนแปลงได้เพียงครั้งเดียวด้วยคำสั่งเฉพาะเท่านั้น เช่น หากต้องการบังคับไม่ให้ข้อมูลไบต์ใดไม่สามารถแก้ไขได้ก็ให้ทำการเคลียร์บิตที่ 9 ของข้อมูลไบต์นั้นๆ แต่สำหรับรหัสผ่านในการเข้าถึงข้อมูลสามารถเปลี่ยนแปลงได้ แต่ต้องแสดงรหัสผ่านชุดเก่าให้บัตรได้ทราบเสียก่อนจึงจะสามารถเปลี่ยนแปลงรหัสผ่านได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

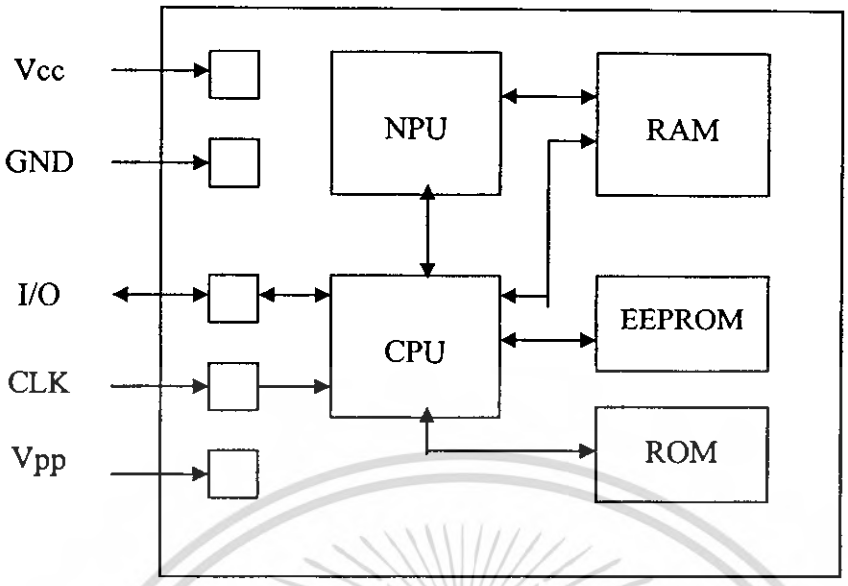
สมาร์ทการ์ดอีกชนิดหนึ่งที่มีใช้เป็นบัตรโทรศัพท์ในประเทศไทยนั้นก็คือ การ์ดหน่วยความจำชนิด Token ภายในสมาร์ทการ์ดชนิดนี้จะมีการเก็บข้อมูลในลักษณะจำนวนนับ (Counter) ซึ่งจำนวนนับนี้จะเป็นตัวเลขแทนมูลค่าของเงินที่ระบุบนบัตร การนับเลขเป็นการนับถอยหลังเพื่อเป็นการนับมูลค่าที่คงเหลือในบัตร หมายความว่าหากใช้บัตรในการโทรศัพท์ไปเรื่อยๆ มูลค่าในบัตรก็จะถูกลดลงตามไปด้วยเช่นกัน ในการเข้าถึงข้อมูลของสมาร์ทการ์ดชนิดนี้ต้องมีการแสดงรหัสผ่านให้บัตรทราบเหมือนกับการ์ดหน่วยความจำ ชนิด PIN Protect แต่ไม่มี Bit Protect เท่านั้นเอง

สมาร์ทการ์ดชนิดหน่วยความจำเป็นสมาร์ทการ์ดพื้นฐานของสมาร์ทการ์ดรุ่นใหม่ๆ ในปัจจุบัน ด้วยโครงสร้างและการทำงานที่ง่ายต่อการทำความเข้าใจ ราคาถูก สามารถเก็บข้อมูลได้จำนวนมาก และความเร็วในการทำงานของชิปไม่สูงมากนัก จึงทำให้สมาร์ทการ์ดชนิดนี้เหมาะที่จะนำไปประยุกต์ใช้กับงานที่ข้อมูลไม่ค่อยสำคัญมากนัก เช่น บัตรลงเวลาทำงาน บัตรผ่านประตู บัตรโทรศัพท์ ฯลฯ ปัจจุบันสมาร์ทการ์ดชนิดหน่วยความจำ มีขนาดหน่วยความจำสูงสุดถึง 64 กิโลไบต์ และอีกไม่นานจะมีสมาร์ทการ์ดที่มีขนาดหน่วยความจำข้อมูลถึง 128 กิโลไบต์

2.6.2 การ์ดชนิดโปรเซสเซอร์ (Processor card)

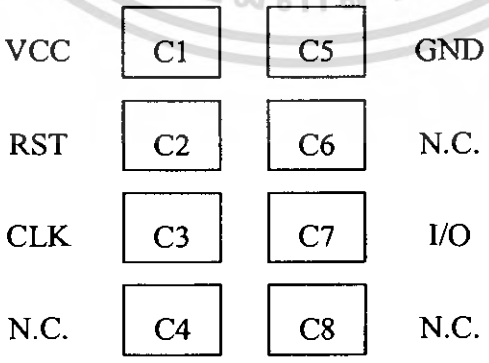
สมาร์ทการ์ดชนิดโปรเซสเซอร์ หรือเรียกอีกชื่อหนึ่งว่า Asynchronous card เป็นสมาร์ทการ์ดที่ได้รับการปรับปรุงจากสมาร์ทการ์ดชนิดหน่วยความจำด้วยการใส่เทคโนโลยีไมโครโปรเซสเซอร์เข้าไปในชิป เพื่อให้ชิปสามารถประมวลผลข้อมูล และเพิ่มความปลอดภัยให้แก่ข้อมูลได้สูงขึ้น การที่ไมโครโปรเซสเซอร์ลงในชิปทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำสำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผลข้อมูล นอกจากนี้ยังมีการใส่ชิปประมวลผลทางคณิตศาสตร์ลงในชิปสมาร์ทการ์ดเพื่อช่วยให้การประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้า-ถอดรหัส ทำให้สมาร์ทการ์ดชนิดโปรเซสเซอร์มีความเร็วในการทำงานสูงกว่าสมาร์ทการ์ดชนิดหน่วยความจำหลายเท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 แสดงบล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ทการ์ดชนิดโปรเซสเซอร์

ในการรับส่งข้อมูลให้กับสมาร์ทภาร์ดชนิดนี้ จะใช้หน้าสัมผัสเดียวกับสมาร์ทการ์ดชนิดหน่วยความจำ โดยสัญญาณนาฬิกาที่ป้อน จะถูกใช้เป็นสัญญาณนาฬิกาให้แก่โปรเซสเซอร์ภายในสมาร์ทการ์ด ข้อมูลที่รับส่งจึงไม่จำเป็นต้องสัมพันธ์กับสัญญาณนาฬิกาที่ป้อนให้แก่ชิป เพียงกำหนดอัตราการรับ-ส่งเป็น 9600 บิด/วินาที ก็จะสามารถติดต่อกับโปรเซสเซอร์ของชิปได้แล้ว แต่การเข้าถึงข้อมูลจะไม่สามารถทำได้เหมือนอย่างในสมาร์ทการ์ดชนิดหน่วยความจำ การเข้าถึงข้อมูลต้องกระทำผ่านโปรเซสเซอร์ของสมาร์ทการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตาม เพราะหน่วยความจำจะอยู่ภายในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว ข้อดีอย่างหนึ่งที่ไม่สามารถติดต่อกับหน่วยความจำในชิปได้โดยตรงก็คือ การลอบเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตแทบเป็นไปไม่ได้ ยกเว้นมีความบกพร่องในการกำหนดเงื่อนไขในการเข้าถึงข้อมูลที่เป็นความลับ



รูปที่ 2.8 แสดงขาต่างๆ บนหน้าสัมผัสของบัตรสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 แสดงหน้าที่การทำงานของขาต่างๆ ของบัตรสมาร์ทการ์ด

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

2.7 มาตรฐานของสมาร์ทการ์ด

มาตรฐานที่เกี่ยวข้องกับสมาร์ทการ์ดมีอยู่ 2 มาตรฐาน คือ ISO TC68/SC6 และ ISO/IEC JTC1/SC17 มาตรฐานของสมาร์ทการ์ดพัฒนามาจากมาตรฐานของไอดีการ์ด (identification card) ซึ่งมีทั้ง มาตรฐาน ISO 7810, 7811 7812 และ 7813 ซึ่งระบุลักษณะ ที่เป็นที่ทราบกันดีในรูปแบบ ID-1 เมื่อสมาร์ทการ์ดสามารถพิมพ์อักษรแบบนูนลงไปได้ (ตัวอักษรที่นูนขึ้นมาทำให้กระดาษคาร์บอนพิมพ์ข้อมูลของเครดิตการ์ดออกมาได้) ทำให้ส่วนที่บรรจุซิลิกอนต้องมีความแข็งแรงและทนทานมากยิ่งขึ้น

2.8 เครื่องอ่านสมาร์ทการ์ด (smart card reader)

สมาร์ทการ์ดรีดเดอร์ที่ใช้กันอยู่ มี 2 ชนิด

1) สมาร์ทการ์ดรีดเดอร์ชนิดหน้าสัมผัส

สมาร์ทการ์ดรีดเดอร์ชนิดหน้าสัมผัส เป็นสมาร์ทการ์ดรีดเดอร์ชนิดแรกที่ถูกสร้างพร้อมๆ กับสมาร์ทการ์ดถูกสร้างขึ้นมาครั้งแรก โดยมีส่วนประกอบของโครงหลักของสมาร์ทการ์ดรีดเดอร์ ขาสัมผัส และสวิตช์ตรวจสอบบัตร ซึ่งโครงหลักของสมาร์ทการ์ดรีดเดอร์เป็นแผ่นพลาสติกที่ขึ้นรูปสำหรับเป็นช่องทางให้สมาร์ตการ์ดสอดเข้ามา โดยส่วนในสุดของช่องจะติดตั้งสวิตช์ตรวจสอบบัตร เมื่อบัตรถูกสอดเข้ามาจนสุดตัวบัตรจะไปกดสวิตช์ตรวจสอบบัตร ทำให้สวิตช์ ON ทำให้เทอร์มินอลทราบได้ว่ามีบัตรสอดเข้ามา นอกจากนี้โครงหลักของสมาร์ทการ์ดรีดเดอร์จะมีการเจอช่องสำหรับติดตั้งขาสัมผัสในตำแหน่งที่ตรงกับหน้าสัมผัสของสมาร์ทการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) สมาร์ทการ์ดรีดเดอร์ชนิด Contactless

สมาร์ทการ์ดรีดเดอร์ชนิด Contactless จะแตกต่างจากชนิดหน้าสัมผัส เนื่องจากสมาร์ทการ์ดชนิด Contactless ใช้คลื่นวิทยุความถี่ 13.56 เมกะเฮิรตซ์ ทำให้ส่วนที่เป็นขาสัมผัสต้องเปลี่ยนเป็นวงจรรับส่งและสายส่งคลื่นวิทยุ ซึ่งมีหลักการคล้ายกับเครื่องรับส่งคลื่นวิทยุ ซึ่งคลื่นที่รับส่งนั้นจะมีความแรงคลื่นไม่สูงนัก ทำให้ระยะการรับส่งข้อมูลลดลง โดยปกติแล้วจะอยู่ในระยะเพียง 3 ถึง 10 เซนติเมตรเท่านั้น (ความผิดพลาดของข้อมูลน้อยกว่า 1 เปอร์เซ็นต์) แต่ก็มีสมาร์ทการ์ดรีดเดอร์ชนิด Contactless บางรุ่นสามารถทำระยะได้สูงถึง 1-10 เมตร ทำให้เกิดความผิดพลาดของข้อมูลมากกว่า 10 เปอร์เซ็นต์ ซึ่งเกินกว่าจะยอมรับได้ในงานบางอย่าง สมาร์ทการ์ดรีดเดอร์ชนิด Contactless ส่วนใหญ่จะเป็นชุดสำเร็จรูปจากผู้ผลิตโดยตรง ซึ่งจะรวมเอาเสารับส่งสัญญาณ วงจรภาครับส่ง และชุดคอนโทรลเลอร์เข้าด้วยกัน โดยผู้ที่ต้องการนำไปใช้งานจะสามารถควบคุม และรับส่งข้อมูลโดยเชื่อมต่อผ่านพอร์ตสื่อสารของสมาร์ทการ์ดรีดเดอร์โดยตรง อีกทั้งเทคโนโลยีเกี่ยวกับสมาร์ทการ์ดชนิด Contactless ยังถูกผู้ผลิตสมาร์ทการ์ดจำกัดไม่ให้นำไปเผยแพร่ จึงทำให้ไม่ค่อยพบเห็นอุปกรณ์ของสมาร์ทการ์ดรีดเดอร์ชนิด Contactless ขายแยกเป็นส่วนๆ

2.9 บอร์ดไมโครคอนโทรลเลอร์ รุ่น CP-JR51AC2 V2.0

บอร์ด CP-JR51AC2 นั้น ถูกพัฒนาขึ้น โดยได้แบ่งแยกรุ่นของบอร์ด ออกเป็น 3 รุ่น สำหรับตอบสนองต่อความต้องการของกลุ่มผู้ใช้ที่มีลักษณะของความต้องการที่แตกต่างกัน โดยทุกบอร์ดต่างก็อาศัยพื้นฐานทางโครงสร้างของวงจรที่เหมือนกัน แต่จะมีความแตกต่างกันในรายละเอียดและส่วนประกอบย่อยๆ เพื่อให้คุณสมบัติของบอร์ดสามารถตอบสนองต่อความต้องการและตรงกับจุดประสงค์ของการนำไปใช้งานของผู้ใช้แต่ละกลุ่มมากที่สุด

บอร์ดไมโครคอนโทรลเลอร์ในกลุ่ม CP-JR51AC2 เป็นบอร์ดไมโครคอนโทรลเลอร์ขนาดกลาง โดยเลือกใช้ CPU ไมโครคอนโทรลเลอร์ขนาด 8 บิต ของ ATMEL ในตระกูล MCS51 เบอร์ T89C51AC2 เป็น CPU ประจำบอร์ด ซึ่ง CPU ตัวนี้บรรจุอยู่ภายในตัวถังแบบ PLCC ขนาด 44 ขา และมีทรัพยากรต่างๆบรรจุไว้ภายในตัว CPU อย่างครบถ้วน ไม่ว่าจะเป็นADC/TIMER/COUNTER/ PWM หรือ PORT I/O ต่างๆ ซึ่งมีความเหมาะสมในการนำไปประยุกต์ใช้งานในลักษณะต่างๆ ได้เป็นอย่างดี เนื่องจากสถาปัตยกรรมทางด้านฮาร์ดแวร์ของ CPU เบอร์นี้จะมีความอ่อนตัวในการใช้งานได้ค่อนข้างดี กล่าวคือ ฟังก์ชั่นการทำงานต่างๆของฮาร์ดแวร์สามารถปรับเปลี่ยนการทำงานได้ด้วยโปรแกรม ดังนั้นผู้ใช้งานจึงสามารถนำระบบฮาร์ดแวร์แบบเดียวกันไปประยุกต์ใช้งานในลักษณะต่างๆกันได้โดยไม่ยากนักโดยการปรับเปลี่ยนโปรแกรม สำหรับควบคุมการทำงานของบอร์ดเพียงเล็กน้อยเท่านั้น

สำหรับอุปกรณ์ I/O ต่างๆ ที่ไม่ได้มีบรรจุไว้ในตัว แต่ได้มีการจัดหาและทำการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ต่างๆ ที่มีความจำเป็นไว้แล้ว ไม่ว่าจะเป็นจอแสดงผลแบบ LCD ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฐานเวลา RTC วงจร Line Driver สำหรับการสื่อสารข้อมูลอนุกรมแบบ RS232 และ RS422/485 และยังสามารถเพิ่มเติมอุปกรณ์ I/O อื่นๆเข้าไปได้อีกตามความจำเป็นในการใช้งาน

ลักษณะของบอร์ดจะออกแบบโครงสร้างของบอร์ดเพื่อ 3 จุดประสงค์หลัก คือ

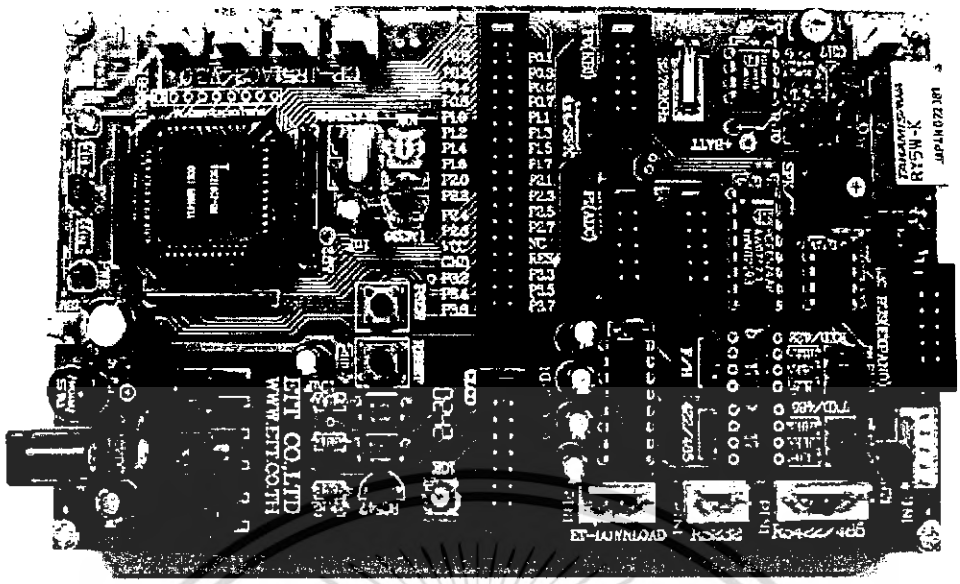
- รุ่นแรก CP-JR51AC2 V1.0 EXPANSION สำหรับกลุ่มลูกค้าที่ต้องการนำบอร์ดไปใช้ศึกษาทดลองและเรียนรู้ทำความเข้าใจกับ CPU อย่างง่ายๆโดยมุ่งเน้นที่จะต่อทดลองวงจรส่วนของ I/O ต่างๆขึ้นมาทดลองเอง โดยใช้วิธีการต่อวงจรด้วยแผงทดลอง Photo Board เป็นหลัก
- รุ่นที่สอง คือ CP-JR51AC2 V1.0 นั้น จุดประสงค์ของการออกแบบบอร์ดเพื่อใช้ตอบสนองกลุ่มลูกค้าที่ต้องการนำบอร์ดไปใช้พัฒนางานต้นแบบ โดยมุ่งเน้นการออกแบบวงจรในส่วน I/O เอง ซึ่งภายในบอร์ดจะมีพื้นที่สำหรับต่อวงจร I/O ไว้สำหรับบัดกรีต่อวงจรเพิ่มเติม
- รุ่นที่สาม คือ CP-JR51AC2 V2.0 นั้นมุ่งเน้นสำหรับตอบสนองความต้องการของกลุ่มผู้ใช้ที่ต้องการสร้าง Application ของงาน โดยไม่สะดวกที่จะสร้างบอร์ดไมโครคอนโทรลเลอร์ขึ้นมาใช้งานเอง โดยบอร์ดจะเพิ่มเติมส่วนของวงจร I/O พื้นฐานต่างๆที่มีความจำเป็นสำหรับการนำไปประยุกต์ใช้งานในรูปแบบต่างๆไว้สำหรับตอบสนองความต้องการของผู้ใช้

2.9.1 บอร์ด CP-JR51AC2 V2.0

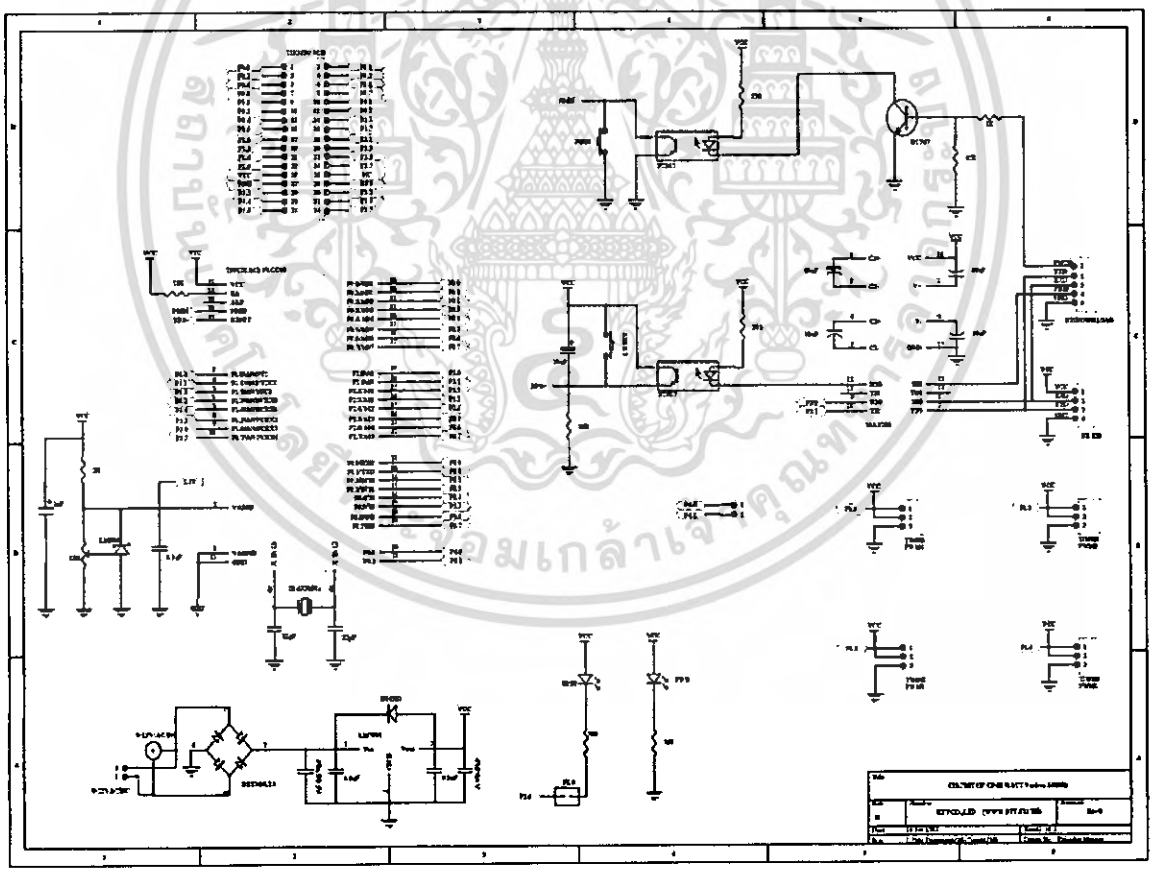
เป็นบอร์ดไมโครคอนโทรลเลอร์ที่มีการออกแบบวงจรสำหรับเชื่อมต่อกับอุปกรณ์ I/O ภายนอกอื่นๆ ที่มีความจำเป็นไว้รองรับการใช้งานในลักษณะต่างๆ เพื่อให้ผู้ใช้งานสามารถนำบอร์ดไปใช้งานในลักษณะงานที่แตกต่างกันได้ โดยไม่ต้องดัดแปลงวงจร หรือ อาจดัดแปลงวงจรเพียงเล็กน้อยสำหรับงานบางอย่าง ซึ่งบอร์ดรุ่นนี้เหมาะสำหรับกลุ่มผู้ที่ต้องการนำบอร์ดไมโครคอนโทรลเลอร์ไปใช้งานจริงๆ แต่ไม่สะดวกที่จะสร้างบอร์ดเอง

71966

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดง ลักษณะของบอร์ด CP-JR51AC2 V2.0



รูปที่ 2.10 แสดงบอร์ด CP-JR51AC2 V 2.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.2 แหล่งจ่ายไฟ (POWER SUPPLY)

สำหรับแหล่งจ่ายไฟของบอร์ดในกลุ่ม CP-JR51AC2 สามารถต่อใช้งานได้ทั้งกับไฟกระแสตรงและกระแสสลับ เนื่องจากในบอร์ดมีวงจร RECTIFIER แบบ BRIDGE พร้อมวงจร FILTER และ REGULATOR ขนาด +5 V สามารถป้อนแรงดันไฟตรงหรือไฟสลับที่มีระดับแรงดันประมาณ 9-12 V ให้กับบอร์ด โดยสามารถเลือกต่อกับขั้ว CONNECTOR แบบ CPA ขนาด 2 ขา หรือต่อผ่านขั้ว CONNECTOR สำหรับ ADAPTER จ่ายไฟก็ได้ โดยการทำงานของแหล่งจ่ายไฟจะมีหลอดแสดงผล LED "PWR" สำหรับแสดงผลการทำงานให้ทราบ

2.9.3 สัญญาณนาฬิกา CLOCK

ความถี่ของสัญญาณนาฬิกาที่จะป้อนให้กับ CPU เบอร์ T89C51AC2 นั้น ตามปกติทั่วๆ ไปแล้ว จะสามารถป้อนค่าความถี่ของ Crystal ได้มากถึง 40 MHz ในกรณีที่จะโปรแกรมโหมดการทำงานของ CPU ให้ทำงานใน Standard Mode (12 Clock / 1 Machine Cycle) แต่ในกรณีที่โปรแกรมโหมดการทำงานของ CPUใน X2 Mode (6 Clock / 1 Machine Cycle) จะสามารถใช้ค่าความถี่สูงสุดได้ 20 MHz ซึ่งเทียบเท่ากับความเร็ว 40 MHz ใน Standard Mode แต่สำหรับบอร์ด CP-JR51AC2 นั้นจะกำหนดให้ใช้ค่าความถี่ของ Crystal ที่ป้อนให้กับ CPU ด้วยค่าความถี่ Crystal เป็น 18.432 MHz เพื่อให้การใช้งานพอร์ตสื่อสารอนุกรมสามารถหาค่า Baudrate ได้ลงตัวตามมาตรฐานของการสื่อสารอนุกรมทั่วๆไป ซึ่งค่าความเร็วการทำงานของ CPU ในบอร์ดจะอ้างอิงการทำงานจากความถี่ 18.432 MHz นี้เป็นหลัก แต่อย่างไรก็ตามค่าความเร็วในการปฏิบัติงานของ CPUสามารถปรับเปลี่ยนได้จากโปรแกรมเพื่อให้การทำงานเร็วขึ้นเป็น 2 เท่า โดยกำหนดให้การทำงานของ CPUทำงานใน X2 Mode ซึ่งจะเปรียบเทียบได้กับการทำงานด้วยความเร็วเท่ากับความถี่ 36.864 MHz ใน StandardMode โดยคุณสมบัติการทำงานของ สัญญาณนาฬิกามีดังนี้

- กำหนดให้ CPU ทำงานใน Standard Mode หรือ 12 Clock / 1 Machine Cycle ซึ่งคุณสมบัตินี้จะเหมือนกับ CPU ในตระกูล MCS51 มาตรฐานทั่วๆไป
- กำหนดให้ CPU ทำงานใน X2 Mode หรือ 6 Clock / 1 Machine Cycle ซึ่งจะทำให้การทำงานของ CPU เร็วกว่า CPU ในตระกูล MCS51 มาตรฐานทั่วๆไปถึง 2 เท่า เมื่อเปรียบเทียบโดยใช้ค่าความถี่ของ Crystal ด้วยค่าความถี่เดียวกัน

สำหรับการกำหนดโหมดการทำงานของสัญญาณนาฬิกาของ CPU นั้นสามารถกระทำได้ 2 วิธี ด้วยกันคือ การกำหนดจากบิต X2 (บิต0) ของรีจิสเตอร์ CKCON ในคำสั่งของโปรแกรมที่ผู้ใช้เขียนขึ้น หรืออีกวิธีหนึ่งคือการกำหนดจาก Fuse Bit X2 จากขั้นตอนของการ Downloadโปรแกรมให้กับ CPU ใน Monitor Mode โดยสามารถสั่งเลือกเครื่องหมายถูก (√) หน้าบิต X2 ของโปรแกรม FLIP ซึ่งใช้สำหรับ Download โปรแกรมให้กับCPU ใน Monitor Mode แล้วสั่ง Set DeviceSpecial Byte ซึ่งเมื่อสั่งเลือกบิต X2 ในโปรแกรม FLIP ไปแล้ว หลังจากรีเซ็ตทุกครั้งจะทำให้ CPU เริ่มต้นทำงานแบบ X2 Mode (6 Clock) แต่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าไม่เลือกกำหนดบิต X2 ในโปรแกรม FLIP ไว้ หลังการรีเซ็ตทุกครั้งจะทำให้ CPU เริ่มต้นทำงานใน Standard Mode (12 Clock) แทนอย่างไรก็ตามถึงแม้ว่าจะมีการกำหนดบิต X2 ในส่วนของโปรแกรม FLIP ไว้อย่างไรก็ตาม ผู้ใช้สามารถสั่งเปลี่ยนแปลงการทำงานของ CPU ได้อีกหลังการรีเซ็ตแล้ว โดยเขียนโปรแกรมสั่งให้เซ็ตหรือเคลียร์บิตX2 (บิต0) ของรีจิสเตอร์ CKCON ได้อีกตามต้องการ แต่ถ้าในส่วนของโปรแกรมของผู้ใช้ไม่ได้ไปสั่งเปลี่ยนแปลงค่าของบิต X2 (บิต0) ในรีจิสเตอร์ CKCON เลย การทำ งานของ CPU ก็จะขึ้นอยู่กับการเลือกบิต X2 จากโปรแกรม FLIP เพียงอย่างเดียวเท่านั้น

ดังนั้นเพื่อความแน่นอนและป้องกันความผิดพลาดในการกำหนดความเร็วการทำงานของ CPU นั้น ในส่วนเริ่มต้นโปรแกรมควรสั่งจัดการกับบิต X2 ในรีจิสเตอร์ CKCON ด้วยเสมอ ทั้งนี้ก็เพื่อป้องกันความผิดพลาดจากขั้นตอนของการ Download โปรแกรมใน Monitor Mode โดยเพิ่มคำสั่งในการจัดการบิต X2 ของรีจิสเตอร์ CKCON ไว้ในส่วนเริ่มต้นของโปรแกรมด้วยซึ่งมีวิธีการดังนี้

```
        ORG     0000H                   ; ตำแหน่งแอดเดรสเริ่มต้นของโปรแกรม
MAIN:   ORL     CKCON,#00000001B        ; สั่งเซ็ตค่า X2 บิตของ CKCON (8FH) ให้เป็น "1"
        คำสั่งอื่นๆของโปรแกรม
```

ตัวอย่างการกำหนดให้ CPU ทำงานแบบ X2 Mode (6 Clock / Machine Cycle)

```
        ORG     0000H                   ; ตำแหน่งแอดเดรสเริ่มต้นของโปรแกรม
MAIN:   ANL     CKCON,#11111110B        ; สั่งเคลียร์ค่า X2 บิตของ CKCON (8FH) ให้เป็น "0"
        คำสั่งอื่นๆของโปรแกรม
```

ตัวอย่างการกำหนดให้ CPU ทำงานแบบ Standard Mode (12 Clock / Machine Cycle)

2.9.3.1 การทำงานของรีจิสเตอร์ CKCON และผลต่อการทำงานของ Clock ของ CPU

CKCON เป็นรีจิสเตอร์สำหรับกำหนดความเร็วในการทำ งานของ CPU และอุปกรณ์ต่างๆที่บรรจุไว้ภายในตัว CPU โดยหลังการรีเซ็ตทุกๆครั้งค่าของรีจิสเตอร์ตัวนี้จะมีค่าเป็น "0" ทุกบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
-	WDX2	PCAX2	SIX2	T2X2	T1X2	T0X2	X2

รูปที่ 2.11 ลักษณะโครงสร้างของรีจิสเตอร์ CKCON (ตำแหน่ง 8FH)

- **WDX2 เป็นบิต Watch Dog Clock** ใช้สำหรับกำหนดคุณสมบัติโหมดการทำงานของสัญญาณนาฬิกาที่จะป้อนให้กับวงจร Watch Dog ซึ่งตามปรกติหลังการรีเซ็ตทุกๆ ครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยบิตนี้จะมีผลเมื่อบิต X2 ถูกเซ็ตเป็น "1" ไว้แล้วเท่านั้น โดยถ้ากำหนดให้บิตนี้มีค่าเป็น "0" จะหมายถึง เลือกโหมดการทำงานของความถี่ของวงจร Watch Dog เป็นแบบ 6 Clock / 1 Watch Dog Clock แต่ถ้ากำหนดให้บิตนี้มีค่าเป็น "1" จะหมายถึง การเลือกโหมดการทำงานของความถี่ของจร Watch Dog เป็น 12 Clock / 1 Watch Dog Clock

- **PCAX2 เป็นบิต Programmable Counter Array Clock** ใช้สำหรับกำหนดคุณสมบัติโหมดการทำงานของสัญญาณนาฬิกาที่จะป้อนให้กับวงจร PCA ซึ่งตามปรกติหลังการรีเซ็ตทุกๆ ครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยบิตนี้จะมีผลเมื่อบิต X2 ถูกเซ็ตเป็น "1" ไว้แล้วเท่านั้น โดยถ้ากำหนดให้บิตนี้มีค่าเป็น "0" จะหมายถึง เลือกโหมดการทำงานของความถี่ของวงจร PCA เป็นแบบ 6Clock / 1 PCA Clock แต่ถ้ากำหนดให้บิตนี้มีค่าเป็น "1" จะหมายถึง การเลือกโหมดการทำงานของความถี่ของจร PCA เป็น 12 Clock / 1 PCA Clock

- **SIX2 เป็นบิต Enhanced UART Clock (Mode 0 and 2)** ใช้สำหรับกำหนดคุณสมบัติโหมดการทำงานของสัญญาณนาฬิกาที่จะป้อนให้กับวงจร UART หรือพอร์ตสื่อสารอนุกรม ซึ่งตามปรกติหลังการรีเซ็ตทุกๆ ครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยบิตนี้จะมีผลเมื่อบิต X2 ถูกเซ็ตเป็น "1"ไว้แล้วเท่านั้น โดยถ้ากำหนดให้บิตนี้มีค่าเป็น "0" จะหมายถึง เลือกโหมดการทำงานของความถี่ของวงจร UART เป็นแบบ 6 Clock / 1 UART Clock แต่ถ้ากำหนดให้บิตนี้มีค่าเป็น "1" จะหมายถึง การเลือกโหมดการทำงานของความถี่ของจร UART เป็น 12 Clock / 1 UART Clock

- **T2X2 เป็นบิต Timer2 Clock** ใช้สำหรับกำหนดคุณสมบัติโหมดการทำงานของสัญญาณนาฬิกาที่จะป้อนให้กับวงจร Timer2 ซึ่งตามปรกติหลังการรีเซ็ตทุกๆครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยบิตนี้จะมีผลเมื่อบิต X2 ถูกเซ็ตเป็น "1" ไว้แล้วเท่านั้น โดยถ้ากำหนดให้บิต T2X2 นี้ มีค่าเป็น "0" จะหมายถึง เลือกโหมดการทำงานของความถี่ของวงจร Timer2 เป็นแบบ 6 Clock แต่ถ้ากำหนดให้บิต T2X2 นี้มีค่าเป็น "1" จะหมายถึง การเลือกโหมดการทำงานของความถี่ของจร Timer2 เป็นแบบ 12 Clock

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **T2X1 เป็นบิต Timer1 Clock** ใช้สำหรับกำหนดคุณสมบัติโหมดการทำงานของสัญญาณนาฬิกาที่จะป้อนให้กับวงจร Timer1 ซึ่งตามปรกติหลังการรีเซ็ตทุกๆ ครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยบิตนี้จะมีผลเมื่อบิต X2 ถูกเซ็ตเป็น "1" ไว้แล้วเท่านั้น โดยถ้ากำหนดให้บิต T1X2 นี้ มีค่าเป็น "0" จะหมายถึง เลือกโหมดการทำงานของความถี่ของวงจร Timer1 เป็นแบบ 6 Clock แต่ถ้ากำหนดให้บิต T1X2 นี้มีค่าเป็น "1" จะหมายถึง การเลือกโหมดการทำงานของความถี่ของจร Timer1 เป็นแบบ 12 Clock

- **T0X2 เป็นบิต Timer0 Clock** ใช้สำหรับกำหนดคุณสมบัติโหมดการทำงานของสัญญาณนาฬิกาที่จะป้อนให้กับวงจร Timer0 ซึ่งตามปรกติหลังการรีเซ็ตทุกๆครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยบิตนี้จะมีผลเมื่อบิต X2 ถูกเซ็ตเป็น "1" ไว้แล้วเท่านั้น โดยถ้ากำหนดให้บิต T0X2 นี้ มีค่าเป็น "0"จะหมายถึง เลือกโหมดการทำงานของความถี่ของวงจร Timer0 เป็นแบบ 6 Clock แต่ถ้ากำหนดให้บิต T0X2 นี้มีค่าเป็น "1" จะหมายถึง การเลือกโหมดการทำงานของความถี่ของจร Timer0 เป็นแบบ 12 Clock

- **X2 เป็นบิต CPU Clock** ใช้สำหรับกำหนดโหมดการทำงานของสัญญาณนาฬิกาที่ป้อนให้กับ CPU ซึ่งตามปรกติหลังการรีเซ็ตทุกๆครั้งนั้นบิตนี้จะมีค่าเป็น "0" เสมอ โดยเมื่อกำหนดให้บิตนี้มีค่าเป็น "0" จะเป็นการกำหนดโหมดการทำงานของสัญญาณความถี่นาฬิกาของ CPU ให้ทำงานใน Standard Mode (12 Clock / 1Machine Cycle) ซึ่งการทำงานของ CPU และอุปกรณ์ภายในตัว CPU ทั้งหมดจะอ้างอิงการทำงานจาก Machine Cycle แบบ 12 Clock ทั้งหมด แต่ถ้ากำหนดให้บิดนี้มีค่าเป็น "1" จะเป็นการกำหนดโหมดการทำงานของสัญญาณความถี่นาฬิกาของ CPUให้ทำงานใน X2 Mode (6 Clock / 1Machine Cycle) ซึ่งการทำงานของ CPU จะอ้างอิงการทำงานจาก Machine Cycle แบบ 6 Clock ส่วนอุปกรณ์ภายในอื่นๆ นั้นจะขึ้นอยู่กับการกำหนดบิตเลือกโหมดสัญญาณนาฬิกาของแต่ละอุปกรณ์อีกครั้งหนึ่ง

2.9.4 โหมดการทำงานของบอร์ด

การทำงานของบอร์ด CP-JR51AC2 นั้น สามารถกำหนดโหมดการทำงานของบอร์ดได้ 2 โหมดการทำงานด้วยกัน คือ Monitor Mode และโหมดการทำงานปรกติ User Mode (Run โปรแกรมในตำแหน่ง 0000H) โดยในการเลือกโหมดการทำงานของบอร์ดนั้นจะกระทำในขณะรีเซ็ต CPU โดยถ้าใช้วิธีการรีเซ็ต CPU ตามปรกติด้วยวิธีการจ่ายไฟเลี้ยงให้บอร์ดในครั้งแรก (Power-on Reset) หรือใช้วิธีการรีเซ็ตด้วยการกดสวิตช์รีเซ็ตเพียงอย่างเดียวนั้นบอร์ด CP-JR51AC2 จะเข้าทำงานใน User Mode หรือโหมดการทำงานปรกติ ตามโปรแกรมในตำแหน่ง 0000H แต่ถ้าใช้วิธีการรีเซ็ตโดยให้ขาสัญญาณ PSEN ของ CPU มีสภาวะทางลอจิกเป็น"0" ด้วย (มีการกดสวิตช์ PSEN ไว้ด้วย) บอร์ด CP-JR51AC2 จะเข้าทำงานใน Monitor Mode

โดยวิธีการตรวจสอบโหมดการทำงานของ CPU เบอร์ T89C51AC2 ของ ATMEL นั้น หลังจากขอบขาลง (Falling Edge) ของสัญญาณ RESET บิต ENBOOT ในรีจิสเตอร์ AUXR1 จะถูกกำหนดให้มีค่าเหมือนกับค่าของบิต BLJB (Boot Loader Jump Bit) หลังจากนั้น CPU จะตรวจสอบเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสถานะทางฮาร์ดแวร์ของขาสัญญาณต่างๆในตัว CPU เพื่อตรวจสอบเงื่อนไขการทำงานของ CPU ว่าต้องทำงานใน User Mode(ตำแหน่ง 0000H) หรือ Monitor Mode (ตำแหน่ง F800H) ซึ่งเงื่อนไขการทำงานทางฮาร์ดแวร์มีดังนี้

- ขาสัญญาณ PSEN เป็น "0" หรือไม่
- ขาสัญญาณ EA เป็น "1" หรือไม่
- ขาสัญญาณ ALE เป็น "1" หรือปล่อยลอยหรือไม่

ซึ่งถ้าหากการตรวจสอบเงื่อนไขทางฮาร์ดแวร์ดังกล่าวข้างต้นเป็นจริง รีจิสเตอร์ FCON จะถูกกำหนดให้มีค่าเป็น 00H ส่วนค่าของรีจิสเตอร์ PC จะถูกกำหนดให้มีค่าเป็น F800H เพื่อชี้ไปยังตำแหน่งเริ่มต้นในการทำงานของโปรแกรมใน Monitor Mode ที่อยู่ในตัวของ CPU และเริ่มต้นทำงานตามคำสั่งของโปรแกรมที่อยู่ในตำแหน่ง F800H ทันที ซึ่งโปรแกรมในส่วนนี้จะถูกโปรแกรมมาพร้อมกับตัว CPU ทุกตัวจากโรงงานอยู่แล้ว โดยเมื่อ CPU ตรวจสอบเงื่อนไขการทำงานของฮาร์ดแวร์ข้างต้นแล้วพบว่าเงื่อนไขดังกล่าวเป็นจริง ก็จะเข้าทำงานในMonitor Mode ในทันที โดยไม่สนใจเงื่อนไขของบิต BLJB ว่าเป็นอย่างไร

โดยวงจรของบอร์ด CP-JR51AC2 นั้นจะออกแบบให้ขาสัญญาณ EA ของ CPU มีค่าเป็น "1" ไว้ตลอดส่วนขาสัญญาณ ALE ก็จะปล่อยลอยไว้ ส่วนขาสัญญาณ PSEN จะทำการ Pull-Up ให้มีค่าเป็น "1" ไว้โดยมีการต่อสวิตช์ PSEN เพื่อให้สามารถบังคับให้ขา PSEN เป็น "0" ได้ด้วยถ้ามีการกดสวิตช์ PSEN ไว้ ซึ่งจะเห็นได้ว่าสภาวะของขาสัญญาณต่างๆที่ใช้เลือกโหมดการทำงานของ CPU นั้นเตรียมพร้อมที่จะเข้าทำงานใน Monitor Mode อยู่แล้วขาดแต่เพียงขาสัญญาณ PSEN เพียงอย่างเดียว ดังนั้นการเลือกโหมดการทำงานของบอร์ดจึงขึ้นอยู่กับขาสัญญาณ PSEN โดยควบคุมจากสวิตช์ PSEN เท่านั้น ซึ่งถ้ามีการกดสวิตช์ PSEN รอไว้ก่อนแล้วจึงสั่งรีเซ็ตบอร์ดก็จะทำให้ CPU ตรวจสอบพบว่าขาสัญญาณ PSEN มีค่าเป็น "0" ซึ่งก็จะทำให้ CPU เข้าทำงานในMonitor Mode ทันทีเนื่องจากขาสัญญาณ EA และ ALE นั้นตรงตามเงื่อนไขอยู่ก่อนแล้ว แต่ถ้าสั่งรีเซ็ต CPUโดยไม่มีการกดสวิตช์ PSEN ด้วย CPU ก็จะตรวจพบว่าขาสัญญาณ PSEN มีค่าเป็น "1" จึงทำให้เงื่อนไขทางฮาร์ดแวร์จะไม่ถูกต้องเนื่องจากขา PSEN จะมีค่าเป็น "1" ดังนั้น CPU จึงเข้าทำงานใน User Mode หรือโหมดการทำงานปรกติตามโปรแกรมในตำแหน่งแอดเดรส 0000H

แต่สำหรับในกรณีที่การตรวจสอบเงื่อนไขทางฮาร์ดแวร์แล้วเป็นเท็จ ค่าของ FCON จะถูกกำหนดให้มีค่าเป็น F0H หลังจากนั้นจึงจะมีการตรวจสอบค่าของบิต BLJB ว่าเป็นอย่างไร

- ถ้าบิต BLJB มีค่าเป็น "1" แล้ว CPU ก็จะกระโดดไปทำงานในตำแหน่ง 0000H ซึ่งเป็นตำแหน่งการทำงานตามปรกติของ CPU เหมือนกับ CPU ในตระกูล MCS51 ปรกติทั่วๆไป
- ถ้าบิต BLJB มีค่าเป็น "0" แล้ว CPU จึงจะตรวจสอบค่าของ BSB และ SBV เพื่อตัดสินใจกระโดดไปเริ่มต้นทำงานยังตำแหน่งแอดเดรสต่างๆ ดังตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 แสดงโหมดการทำงานของ CPU หลังการรีเซ็ตแบบปรกติ

BLJB	BSB	SBV	โหมดการทำงานหลังการทำงานแบบปรกติ
( ) ไม่เลือก	XX	XX	User Mode (เริ่ม Run ตำแหน่ง 0000H)
(√) เลือก	00H	XX	User Mode (เริ่ม Run ตำแหน่ง 0000H)
(√) เลือก	ไม่ใช่ 00H	ไม่ใช่ FCH	User Mode (เริ่ม Run จากตำแหน่ง [SBV:00H])
(√) เลือก	ไม่ใช่ 00H	FCH	Monitor Mode (เริ่ม Run ตำแหน่ง F800H)

หมายเหตุ

- BLJB หมายถึง Boot Loader Jump Bit
- BSB หมายถึง Boot Status Byte
- SBV หมายถึง Software Boot Vector
- XX หมายถึง ค่าใดๆ

ซึ่งจะเห็นได้ว่าถ้ามีการเลือกบิต BLJB ไว้ แล้วค่าของ BSB ไม่ได้ถูกกำหนดให้มีค่าเป็น "00H" ไว้ด้วยแล้ว จะทำให้ CPU กระโดดไปทำงานที่ตำแหน่งอื่นๆ ที่ไม่ใช่ 0000H ซึ่งจะขึ้นอยู่กับค่าของ SBV โดยถ้าค่าของSBV มีค่าเป็น FCH จะทำให้ CPU กลับเข้าไปทำงานใน Monitor Mode ที่ตำแหน่ง F800H แต่ถ้าค่าของ SBVเป็นค่าอื่นๆ ที่ไม่ใช่ FCH จะทำให้ CPU กระโดดไปทำงานยังตำแหน่งที่ชี้โดย SBV โดยค่าที่กำหนดให้ SBV จะเป็นค่าแอดเดรสไบต์สูงส่วนค่าแอดเดรสไบท์ต่ำจะมีค่าเป็น 00H เสมอ

1) การทำงานใน MONITOR MODE

ในโหมดนี้ใช้สำหรับในกรณีที่ต้องการพัฒนาโปรแกรมของบอร์ดหรือการ Download โปรแกรมแบบ HEX File จากเครื่องคอมพิวเตอร์ PC ให้กับหน่วยความจำโปรแกรมของ CPU ซึ่งตามปรกติแล้วจะต้องใช้ร่วมกับโปรแกรม FLIP (Flexible In-system Programmer) ของ ATMEL โดยใช้การติดต่อสื่อสารระหว่างเครื่องคอมพิวเตอร์ PC กับ CPU เบอร์ T89C51AC2 ในบอร์ดด้วยพอร์ตสื่อสารอนุกรม RS232 ซึ่งในโหมดนี้ผู้ใช้สามารถสั่งจัดการหน่วยความจำ FLASH ภายในตัวของ CPU ได้โดยตรงไม่ว่าจะเป็นการสั่งลบข้อมูล หรือเขียนข้อมูลใหม่ให้กับหน่วยความจำ FLASH ของ CPU

ข้อจำกัดของการใช้งานบอร์ดใน MONITOR MODE

1. สัญญาณ ALE ของ CPU ต้องปล่อยลอยไว้ หรือกำหนดให้มีสภาวะเป็น "1"
2. สัญญาณ EA ของ CPU ต้องกำหนดให้มีสภาวะเป็น "1"
3. สัญญาณ PSEN ของ CPU ต้องมีสภาวะเป็นลอจิก "0" ก่อนปล่อยสัญญาณรีเซ็ต จาก "1" เป็น "0"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การทำงานใน USER MODE หรือ RUN MODE

การทำงานในโหมดนี้เป็นโหมดการทำงานปรกติของบอร์ด โดยจะใช้สำหรับในกรณีที่ผู้ใช้ทำการโปรแกรมข้อมูลให้กับหน่วยความจำ FLASH ของ CPU เรียบร้อยแล้ว ซึ่งในโหมดการทำงานนี้สามารถจะใช้งานทรัพยากรต่างๆของ CPU ได้อย่างครบถ้วนโดยไม่มีข้อจำกัดใดๆ โดยวิธีการกำหนดโหมดการทำงานของบอร์ด CP-JR51AC2 เป็น RUN MODE หลังการรีเซ็ตบอร์ดทุกๆครั้ง ถ้าสภาวะของขาสัญญาณ PSEN มีค่าเป็น"1" อยู่ CPU ก็จะกระโดดเข้ามาทำงานในโหมดนี้โดยอัตโนมัติอยู่แล้วซึ่งหลังจาก CPU ฟื้นสภาวะจากการรีเซ็ตแล้วตรวจพบว่าขาสัญญาณ PSEN มีค่าเป็น "1" อยู่ CPU ก็จะเริ่มทำงานตามโปรแกรมที่บรรจุไว้ในหน่วยความจำของ CPU ในทันที

2.9.5 การสื่อสารอนุกรมแบบ RS232/RS422/RS485

2.9.5.1 การสื่อสารอนุกรมแบบ RS232

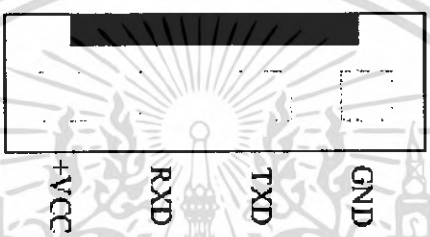
กรณีนี้จะต้องทำ าการติดตั้งไอซี Line Driver เพื่อเปลี่ยนระดับสัญญาณทางไฟฟ้าของขาสัญญาณสำหรับ รับ-ส่ง ข้อมูลแบบ TTL ของ CPU (RX และ TX) ให้เป็นระดับสัญญาณทางไฟฟ้าแบบ RS232 (±12V)โดยการติดตั้งไอซีเบอร์ MAX232 เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณ TTL จากขาสัญญาณส่งข้อมูล (TX) ของCPU ให้เป็นระดับสัญญาณ ±12V สำหรับส่งไปยังขารับสัญญาณ (RX) ของอุปกรณ์ภายนอก และในทางกลับกัน ก็จะทำหน้าที่เปลี่ยนระดับสัญญาณส่ง (TX) แบบ RS232 (±12V) จากอุปกรณ์กายนอก ให้กลับมาเป็นระดับ TTL เพื่อส่งให้กับขารับข้อมูล (RX) ของ CPU ด้วย โดยเมื่อเปลี่ยนระดับสัญญาณในการรับส่งข้อมูลจากTTL มาเป็นแบบ RS232 นี้แล้วจะทำ าให้สามารถทำการ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอกที่ใช้ระดับสัญญาณทางไฟฟ้าในการ รับ-ส่ง แบบเดียวกัน (RS232) ได้ไกลขึ้น ประมาณ 50 ฟุต หรือ ประมาณ 15 เมตร โดยสามารถทำการ รับ-ส่ง ข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะของตัวต่อตัว (Point-to-Point) เท่านั้น

สำ าหรับสายสัญญาณที่จะนำมาใช้สำ าหรับทำการสื่อสารแบบ RS232 นั้น จะใช้สัญญาณเพียง 2-3 เส้นเท่านั้น ทั้งนี้ขึ้นอยู่กับความต้องการในการสื่อสารว่าต้องการสื่อสารแบบทิศทางเดียวหรือสองทิศทาง

- **การสื่อสาร RS232 แบบสองทิศทาง** มีทั้งการรับข้อมูลและส่งข้อมูลไปมาระหว่างด้านรับและด้านส่ง โดยในกรณีนี้จะต้องใช้สายสัญญาณจำ านวน 3 เส้น สัญญาณรับข้อมูล (RXD) สัญญาณส่งข้อมูล(TXD) และสัญญาณอ้างอิง (GND) โดยในการเชื่อมต่อสายนั้นจะต้องทำการสลับสัญญาณกับอุปกรณ์ปลายทางด้วย คือ สัญญาณส่ง (TXD) จากบอร์ด CP-JR51AC2 จะต้องต่อเข้ากับสัญญาณรับ (RXD) ของอุปกรณ์ และสัญญาณส่ง (TXD) จากอุปกรณ์ก็ต้องต่อกับสัญญาณรับ (RXD) ของบอร์ด ส่วนสัญญาณอ้างอิง (GND) จะต้องต่อตรงถึงกัน จึงจะสามารถทำการรับ-ส่ง ข้อมูลกันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การสื่อสาร RS232 แบบทิศทางเดียว ซึ่งอาจเป็นการรอรับข้อมูลจากด้านส่งเพียงอย่างเดียวหรืออาจเป็นการส่งข้อมูลออกไปยังปลายทางเพียงอย่างเดียว โดยไม่มีการโต้ตอบข้อมูลซึ่งกันและกัน ซึ่งวิธีนี้จะใช้สายสัญญาณเพียง 2 เส้น เท่านั้น โดยถ้าเป็นทางด้านส่งก็จะต่อเพียงสัญญาณส่ง (TXD) และสัญญาณอ้างอิง (GND) แต่ถ้าเป็นทางด้านรับ ก็จะต่อเพียงสัญญาณรับ (RXD)และ สัญญาณอ้างอิง (GND) เท่านั้น โดยขั้วต่อของสัญญาณ RS232 ของบอร์ด CP-JR51AC2 ทั้ง 2 รุ่น นั้น จะเป็นจุดเชื่อมต่อของสัญญาณ รับ-ส่ง ข้อมูล ที่เปลี่ยนระดับสัญญาณเป็นแบบ RS232 แล้ว ซึ่งจะมีลักษณะเป็นแบบขั้ว CPA ขนาด 4 PIN สำหรับใช้เป็นจุดเชื่อมต่อสัญญาณ รับ-ส่ง ข้อมูลกับอุปกรณ์ภายนอก โดยมีลักษณะการจัดเรียงสัญญาณดังนี้



รูปที่ 2.12 แสดงขั้วต่อสัญญาณ RS232 ของบอร์ด CP-JR51AC2 V1.0 & V2.0

ซึ่งจะเห็นได้ว่าขั้วต่อสัญญาณ RS232 ของบอร์ดนั้น จะมีทั้งหมด 4 เส้น แต่ในการรับ-ส่ง ข้อมูลแบบปรกตินั้น จะใช้สัญญาณเพียงแค่ 3 เส้น คือ RXD, TXD และ GND เท่านั้น ส่วน +VCC ซึ่งเป็นไฟเลี้ยงวงจร +5Vนั้น จะไม่จำเป็นต้องนำมาใช้ในการสื่อสารกันแต่อย่างใด โดย +VCC หรือ +5V นี้ จะออกแบบเผื่อไว้ในกรณีที่อุปกรณ์ปลายทางเป็นวงจรขนาดเล็กและไม่สะดวกที่จะหาแหล่งจ่ายไฟให้กับอุปกรณ์ปลายทางด้วย ก็อาจต่อไฟเลี้ยงวงจร +VCC นี้ออกไปให้กับอุปกรณ์ปลายทางด้วยก็ได้เช่นกัน

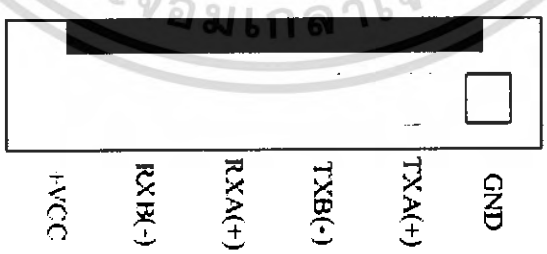
2.9.5.2 การสื่อสารอนุกรมแบบ RS422

ในกรณีนี้จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176 หรือ MAX3088 จำนวน 1-2 ตัว เพื่อทำหน้าที่เปลี่ยนระดับสัญญาณการไฟฟ้าในการ รับ-ส่ง แบบ TTL จาก CPU ให้เป็นระดับสัญญาณแบบ BalanceLine เพื่อ รับ-ส่งสัญญาณกับอุปกรณ์ที่มีระดับสัญญาณทางไฟฟ้าเป็นแบบ Balance Line เหมือนกัน โดยถ้าต้องการใช้การสื่อสารแบบ 2 ทิศทาง ต้องติดตั้งไอซี Line Driver จำนวน 2 ตัว โดยแบ่งเป็นตัวแปลงสัญญาณทางด้านรับ 1 ตัว และตัวแปลงสัญญาณด้านส่งอีก 1 ตัว แต่ถ้าต้องการสื่อสารแบบทิศทางเดียวก็อาจทำการติดตั้งไอซี Line Driver เพียงตัวเดียว โดยถ้าต้องการให้เป็นฝ่ายรับข้อมูลเพียงอย่างเดียวก็ให้ติดตั้งไอซี Line Driver เฉพาะในตำแหน่งของ “RXD/422” เพียงตัวเดียว แต่ถ้าต้องการให้เป็นฝ่ายส่งข้อมูลเพียงอย่างเดียวก็ให้ทำการติดตั้งไอซี Line Driver เฉพาะในตำแหน่ง “TXD/485” เพียงตัวเดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการสื่อสารแบบ RS422 นี้ สามารถนำไปทดแทนการสื่อสารแบบ RS232 ได้ทันที โดยไม่ต้องดัดแปลงหรือแก้ไขโปรแกรมเลย ซึ่งการสื่อสารโดยใช้ระดับสัญญาณในการรับ-ส่ง แบบ RS422 นี้จะมีข้อดี คือสามารถทำการสื่อสารกันได้ในระยะทางที่ไกลขึ้นกว่าแบบ RS232 มาก กล่าวคือ สามารถจะทำ าการรับ-ส่ง ข้อมูลกันได้ในระยะทางประมาณ 4000 ฟุต หรือ 1200 เมตร หรือ 1.2 กิโลเมตรเลย ทีเดียว เพียงแต่ต้องใช้สายสัญญาณที่ออกแบบมาสำหรับรองรับการใช้งานในด้านการสื่อสารแบบนี้ โดยเฉพาะ ซึ่งได้แก่ สายสัญญาณแบบUTP (Un-Shiled Twist Pair) หรือ STP (Shiled Twist Pair) โดยการสื่อสารด้วยระดับสัญญาณทางไฟฟ้าแบบRS422 นี้ ถ้าเป็นการสื่อสารแบบ 2 ทิศทาง คือ ทั้งรับข้อมูลและส่งข้อมูลจะสามารถทำการรับส่งข้อมูลกับอุปกรณ์ต่างๆได้ในลักษณะของตัวต่อตัว (Point-to-Point) เหมือนกับ RS232 ทุกประการ แต่ในกรณีที่เป็นการสื่อสารแบบทิศทางเดียวนั้น สามารถจะทำการต่อขนานสัญญาณทางด้านรับจำนวนหลายๆ จุดเข้ากับสัญญาณส่งเพียงจุดเดียวได้ โดยถ้าเลือกใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานจำนวนอุปกรณ์สำ าหรับด้านรับข้อมูลได้ประมาณ 32จุด แต่ถ้าเลือกใช้ไอซี Line Driver เบอร์ MAX3088 นั้น จะสามารถต่อขนานจำ านวนอุปกรณ์ทางด้านรับข้อมูลได้มากถึง 256 จุด เลย ทีเดียว แต่ถ้าเป็นอุปกรณ์ทางด้านส่งนั้น จะไม่สามารถนำ ามาต่อขนานสัญญาณส่งข้อมูลเข้าด้วยกันมากกว่า 1 จุด เหมือนทางด้านฝ่ายรับได้ ซึ่งวงจร Line Driver แบบ RS422 นี้จะมีอยู่เฉพาะในบอร์ดรุ่น CP-JR51AC2 V2.0 เท่านั้น

สำหรับลักษณะของขั้วต่อของสัญญาณ RS422 นั้น จะเป็นแบบ CPA ขนาด 6 PIN ดังรูป โดยในการสื่อสารกันนั้น จะใช้สายสัญญาณในการ รับ-ส่ง ข้อมูลกัน จำ านวน 4 เส้นสัญญาณ คือ สัญญาณในการรับข้อมูลจำ านวน 2 เส้น คือ RXA (RX+) และ RXB (RX-) และสัญญาณในการส่งข้อมูลอีก 2 เส้น คือ TXA (TX+) และTXB (TX-) ซึ่งในการต่อสัญญาณนั้น จะต้องทำการต่อสัญญาณในลักษณะของการสลับกัน คือ สัญญาณส่งจะต้องต่อเข้ากับสัญญาณรับ นั่นก็คือ สัญญาณ RXA (RX+) จะต้องต่อกับ TXA (TX+) ส่วน RXB (RX-) ก็จะต้องต่อกับ TXB (TX-) ด้วยเช่นกัน โดยลักษณะของขั้วต่อสัญญาณ RS422 เป็นดังรูป



รูปที่ 2.13 แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-JR51AC2 V2.0 เมื่อเลือกเป็น RS422

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.5.3 การสื่อสารอนุกรมแบบ RS485

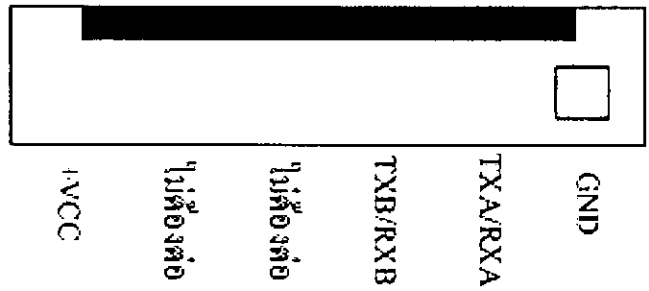
ในการสื่อสารแบบ RS485 นี้จะมีคุณสมบัติของสัญญาณทางไฟฟ้าเหมือนกับ RS422 ทุกประการเพียงแต่ว่าในการสื่อสารแบบ RS485 นี้จะใช้สายสัญญาณในการรับส่งข้อมูลกันเพียง 2 เส้น เท่านั้น แต่จะมีความพิเศษกว่าแบบ RS422 ตรงที่ ทิศทางของสัญญาณจะสามารถปรับเปลี่ยนได้จากโปรแกรม กล่าวคือสัญญาณทั้ง 2 เส้น นี้สามารถจะสลับหน้าที่เป็นด้านส่ง และ เป็นด้านรับได้ ตามต้องการ โดยการควบคุมจากCPU โดยจากบอร์ด CP-JR51AC2 V2.0 นั้น จะกำหนดให้สัญญาณ PTC3 ทำหน้าที่สำหรับควบคุมทิศทางของข้อมูลว่าจะให้เป็นรับหรือส่ง โดยถ้าควบคุมให้ PTC3 มีสภาวะเป็น "1" จะเป็นการกำหนดทิศทางให้เป็นฝ่ายส่งข้อมูล แต่ถ้าสภาวะของ PTC3 เป็น "0" จะเป็นการกำหนดทิศทางให้เป็นฝ่ายรับข้อมูล ซึ่งจากคุณสมบัติข้อนี้จะทำให้การสื่อสารแบบ RS485 สามารถทำ าการต่อขนานอุปกรณ์ร่วมกันในสายส่งเดียวกันได้จำ านวนหลายๆจุดโดยถ้าใช้ไอซี Line Driver เบอร์ 75176 จะสามารถต่อขนานอุปกรณ์กันได้ จำนวน 32 จุด แต่ถ้าเลือกใช้ไอซี Line Driver เบอร์ MAX3088 แล้วจะสามารถต่อขนานอุปกรณ์ในสายคู่เดียวกันได้มากถึง 256 จุด เลยทีเดียวแต่มีข้อแม้ว่า เมื่อมีการต่ออุปกรณ์ขนานกันในสายสัญญาณคู่เดียวกันมากกว่า 2 จุดแล้ว จะต้องเขียนโปรแกรมควบคุมให้มีการส่งข้อมูลออกมาในสายครั้งละ 1 จุดเท่านั้น เพราะถ้ามีการกำ าหนดทิศทางของข้อมูลให้เป็นส่งในเวลาเดียวกันมากกว่า 1 จุดแล้วจะทำให้เกิดการชนกันของข้อมูลและไม่สามารถสื่อสารกันได้อย่างถูกต้อง

โดยเมื่อต้องการใช้วิธีการสื่อสารแบบ RS485 นี้ จะต้องทำการติดตั้งไอซี Line Driver เบอร์ 75176หรือ MAX3088 ในตำแหน่งของ "TXD/485" เพียงตัวเดียว พร้อมกับเลือกกำหนดเป็นแบบ RS485 ดังนี้

- ทำ าการเลือก Jumper สำ าหรับเลือก "422/485" ไว้ทางด้าน 485 (RS485)
- ทำ าการเลือก Jumper "F/H" ไว้ทางด้าน H (Half Duplex)
- ทำการ Short Jumper สำ าหรับต่อตัวต้านทาน Fail Safe Resister คือ "TL"
- ทำการ Short Jumper สำ าหรับต่อตัวต้านทาน Fail Safe Resister คือ "TH"
- สายสัญญาณที่ใช้จะต่อจาก TXB(TX-) และ TXA(TX+) เพียง 2 เส้น ออกไปใช้งาน

ซึ่งในการสื่อสารข้อมูลแบบ RS485 นี้ จะต้องเขียนโปรแกรมขึ้นมารองรับการสื่อสารโดยเฉพาะ เนื่องจากทิศทางของข้อมูลสามารถจะกำหนดจากโปรแกรมได้โดยตรง ซึ่งการสื่อสารวิธีนี้จะมีข้อดีคือ ใช้สายสัญญาณในการรับส่งน้อยเส้น แต่จะเสียเวลาในการสื่อสารมากกว่าวิธีอื่นๆ เนื่องจากการสื่อสารแบบนี้จะไม่สามารถทำ าการรับและส่งข้อมูลในเวลาเดียวกันได้ แต่จะต้องใช้วิธีการผลัดกันรับผลัดกันส่งแทน ซึ่งในความเป็นจริงแล้วในปัจจุบันนี้ ราคาของสายสัญญาณแบบ 2 เส้น และ 4 เส้น แทบจะไม่มีความแตกต่างกันเลย ดังนั้นเพื่อลดความยุ่งยากในการเขียนโปรแกรมสำ าหรับควบคุมการรับส่งข้อมูลของ CPU ควรเลือกใช้วิธีการสื่อสารแบบ RS422 ซึ่งง่ายและสะดวกรวดเร็วกว่ากันมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

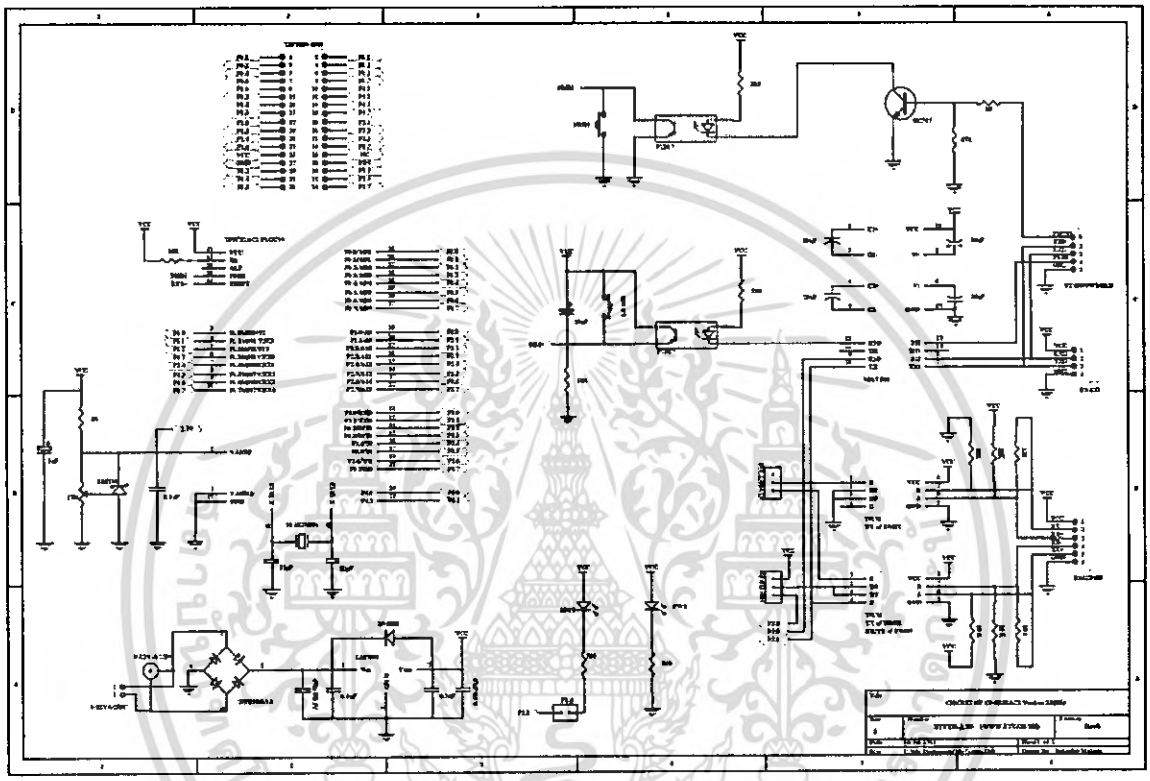


รูปที่ 2.14 แสดงขั้วต่อสัญญาณ RS422/485 ของบอร์ด CP-JR51AC2 V2.0 เมื่อเลือกเป็น RS485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง

3.1 หลักการสร้างและการออกแบบบอร์ด JR51AC2V2

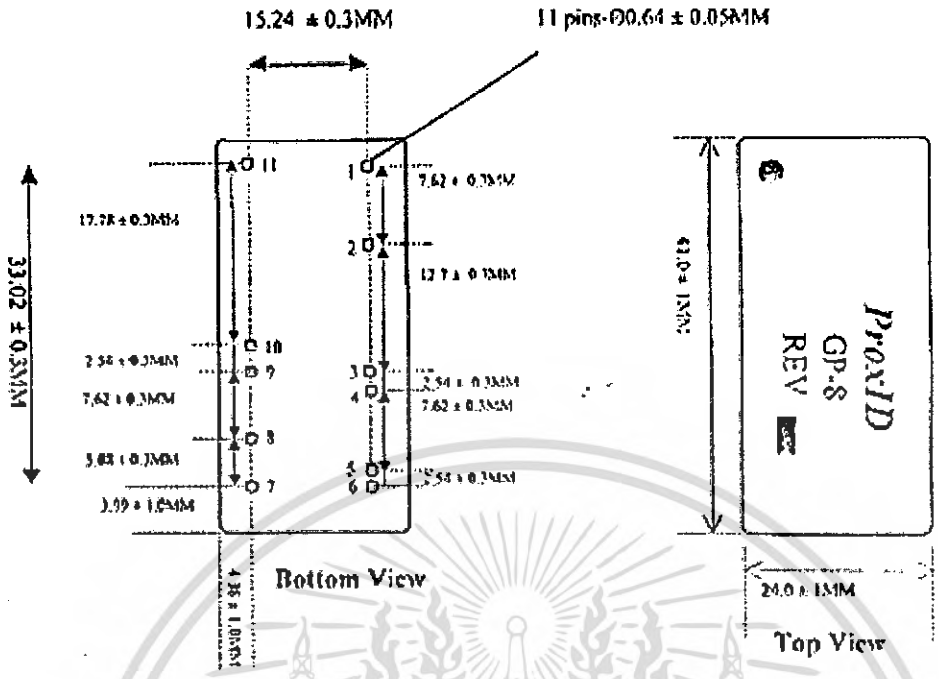


รูปที่ 3.1 วงจรบอร์ด JR51AC2V2

3.2 หลักการสร้าง Reader

หัวอ่านบัตร (Head Reader) เป็นส่วนอ่านและรับข้อมูลจากบัตร และส่งข้อมูลไปยังส่วนควบคุม ส่วนควบคุม (Controller) มีลักษณะเป็นแผ่นปริ้นหรือ Main Board มีไอซีเป็นตัวทำงานตามคำสั่ง โดยจะตรวจเช็คข้อมูลบัตรที่รับมาจากส่วนหัวอ่านบัตร หากข้อมูลถูกต้องตรงกับที่โปรแกรมไว้ส่วนควบคุมนี้ก็จะสั่งให้อุปกรณ์ต่างๆ ที่ต่อเชื่อมให้ทำงานตามฟังก์ชั่นที่กำหนด Housing หมายถึง พื้นผิววัสดุภายนอก ทำจากเหล็กหนา ชุบโครเมี่ยมกันสนิม (สำหรับ Magnetic) และทำจาก PVC (สำหรับ Proximity) คีย์แพ็ด (Keypad) เป็นอุปกรณ์รับข้อมูล หรือรับการโปรแกรมการทำงานต่างๆ มีลักษณะเป็นแผ่นจากยางมีความเหนียวและคงทน (หมายเหตุ ส่วนประกอบดังกล่าว หากหมดอายุหรือเสื่อมอายุการใช้งาน สามารถเปลี่ยนใหม่ได้โดยไม่ต้องเปลี่ยนใหม่ทั้งเครื่อง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 การทำงานของขา reader ProxID GP8

จากรูปที่ 3.2 แต่ละขามีการทำงานดังนี้

ขาที่ 1 ไฟ 0 โวลต์

ขาที่ 2 ไฟ 5.0 – 13.5 โวลต์

ขาที่ 3 Input โปรแกรม

ขาที่ 4 Output ของการ์ด

ขาที่ 5 Output RS232 ของข้อมูล, ข้อมูลแถบแม่เหล็ก & Wiegand0

ขาที่ 6 สัญญาณนาฬิกาของแถบแม่เหล็ก & Wiegand1

ขาที่ 7 input ของ beep control ภายนอก

ขาที่ 8 ไดรฟ์ LED (ใช้ตัวต้านทานซีรีส์ 470-1K)

ขาที่ 9 Buzzer

ขาที่ 10 กราวด์แอทเทนนาภายนอก

ขาที่ 11 ไดรฟ์แอทเทนนาภายนอก

ลักษณะการทำงานของเครื่องควบคุมในแบบต่างๆ

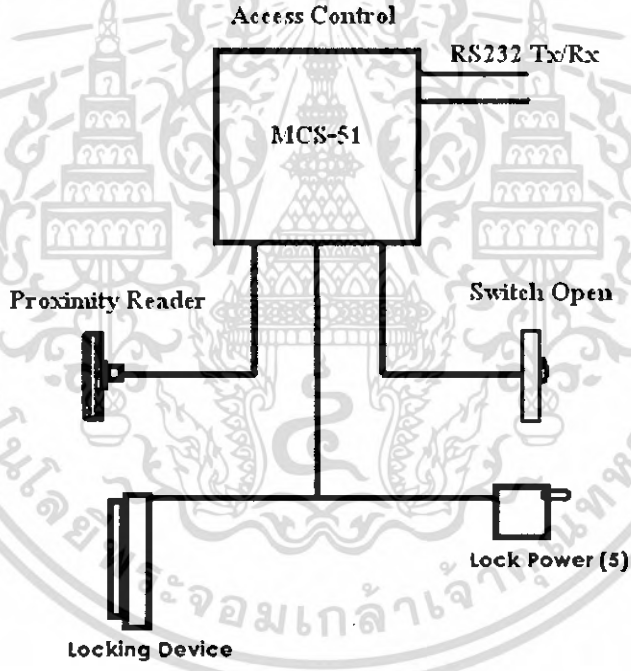
1. เครื่องควบคุม (Controller) สำหรับบัตร แถบแม่เหล็ก (Access Control with Magnetic Striper Reader)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เครื่องควบคุม (Controller) สำหรับบัตรความถี่ (Access Control with Proximity Card)

3. เครื่องควบคุม (Controller) สำหรับบัตรความถี่ใช้สำหรับลงเวลาพนักงาน (Access Control with Proximity Card)

ระบบ Access Control เป็นระบบที่ควบคุมการเข้าหรือออกอัตโนมัติ โดยใช้บัตรเป็นอุปกรณ์สำหรับเข้าผ่าน โดยที่เครื่องควบคุมจะประกอบด้วย 2 ส่วนหลักคือ ส่วนควบคุม (Controller) การทำงานและส่วนของหัวอ่านบัตร (Reader) โดยเครื่องควบคุมจะอ่านข้อมูลในบัตร หากข้อมูลถูกต้องก็จะส่งคำสั่งให้อุปกรณ์อื่นๆ เช่น ตัวล็อค ให้คลายหรือปลดล็อคและเซ็นเซอร์อื่นๆ เช่น ตัวดักจับกันงัด ตัวดักจับควันไฟ ดักจับความร้อน ดักจับการเคลื่อนไหว หรือแม้แต่ เปิด-ปิด ไฟฟ้าแสงสว่าง ให้ทำงานต่อไปและหากข้อมูลบัตร ไม่ถูกต้อง อุปกรณ์ต่างๆ ก็ไม่ทำงานจึงไม่สามารถเข้าผ่านได้ บัตรซึ่งเป็นอุปกรณ์สำหรับเข้า-ผ่าน มี 2 ชนิด คือ บัตรแถบแม่เหล็ก (Magnetic Card) และบัตรความถี่ (Proximity Card)



รูปที่ 3.3 Access control

3.3 การเชื่อมต่อแบบ RS232

การเชื่อมต่อแบบ RS232 นั้นทำได้โดยการนำสาย RS232 ต่อเข้ากับตัวเครื่องคอมพิวเตอร์และตัวอุปกรณ์ที่ใช้ในการบันทึกเวลา จากนั้นทำการเช็คค่าของ Comport ในตัวเครื่องให้เรียบร้อยให้ค่าของ RS232 เป็น Y ดูค่า Baud Rate ที่ตัวเครื่องบันทึกเวลา ตั้งค่าไว้ที่เท่าไร หมายเลขเครื่อง (Dev Num) ว่ามีค่าเท่าไร

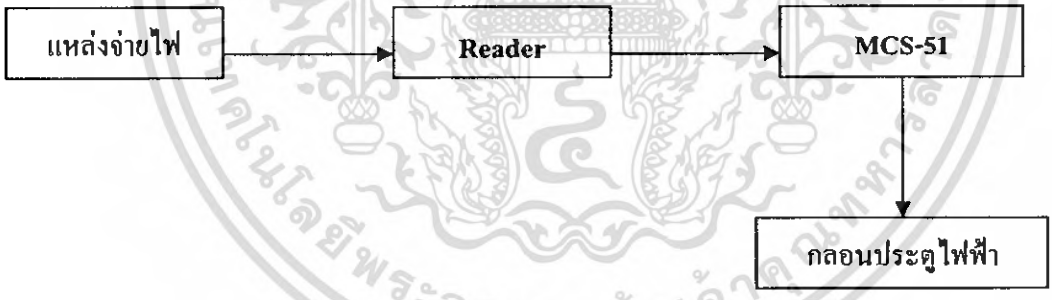
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำการเปิดซอร์ฟแวร์ (Program-Time Attendance) เข้าสู่เมนูการเชื่อมต่อตัวเครื่อง ให้ทำการเลือกประเภทการเชื่อมต่อเป็นแบบ Com Port/RS485ใส่ค่าของหมายเลขเครื่อง (Dev Num) ในส่วนของหมายเลขพอร์ตนั้นโดยปกติแล้วถ้าเป็นคอมพิวเตอร์แบบตั้งโต๊ะ ก็จะมีค่า Default อยู่ที่ Com1 เมื่อมาถึงค่า Baud Rate ก็ให้ใส่ค่าที่ตั้งไว้ที่ตัวเครื่องบันทึกเวลาให้มีค่าเท่ากัน มาถึงในส่วนของ Com Key ในค่านี้จะมีค่าเท่ากับ Password หากที่ตัวเครื่องบันทึกเวลาไม่ได้มีการใส่หมายเลข Com Key ไว้ ก็ไม่ต้องใส่จากนั้นทำการ กดปุ่มเชื่อมต่อ (**หมายเหตุ หากปรากฏการเชื่อมต่อผิดพลาด ให้ลองเช็คที่สาย (RS232) ดูก่อนว่าหลุดหรือไม่ ค่าที่ตัวเครื่องเซ็ตไว้ครงกับที่ S/W จะทำการเชื่อมต่อหรือไม่)

3.4 การเขียนโปรแกรม

ภายในตัว CPU เบอร์ T89C51AC2 ที่ใช้กับบอร์ด CP-JR51AC2 นั้น จะมีวงจรสื่อสารแบบอนุกรม (UART) บรรจุรวมไว้ด้วยแล้ว ซึ่งวงจรส่วนนี้สามารถทำการเขียนโปรแกรมควบคุมการสื่อสารข้อมูลของ CPU กับอุปกรณ์อื่นๆได้ตามต้องการ โดยในส่วนของโปรแกรมนั้นสามารถกำหนดรูปแบบของการสื่อสารข้อมูลได้เองจากโปรแกรมที่เขียนขึ้น ไม่ว่าจะเป็นความเร็วในการสื่อสาร (Baudrate) จำนวนบิตข้อมูลในการรับส่ง(Data Bit) การกำหนดบิตรวจสอบความถูกต้องข้อมูล (Parity) และคุณสมบัติอื่นๆ

3.5 บล็อคไดอะแกรมของระบบ



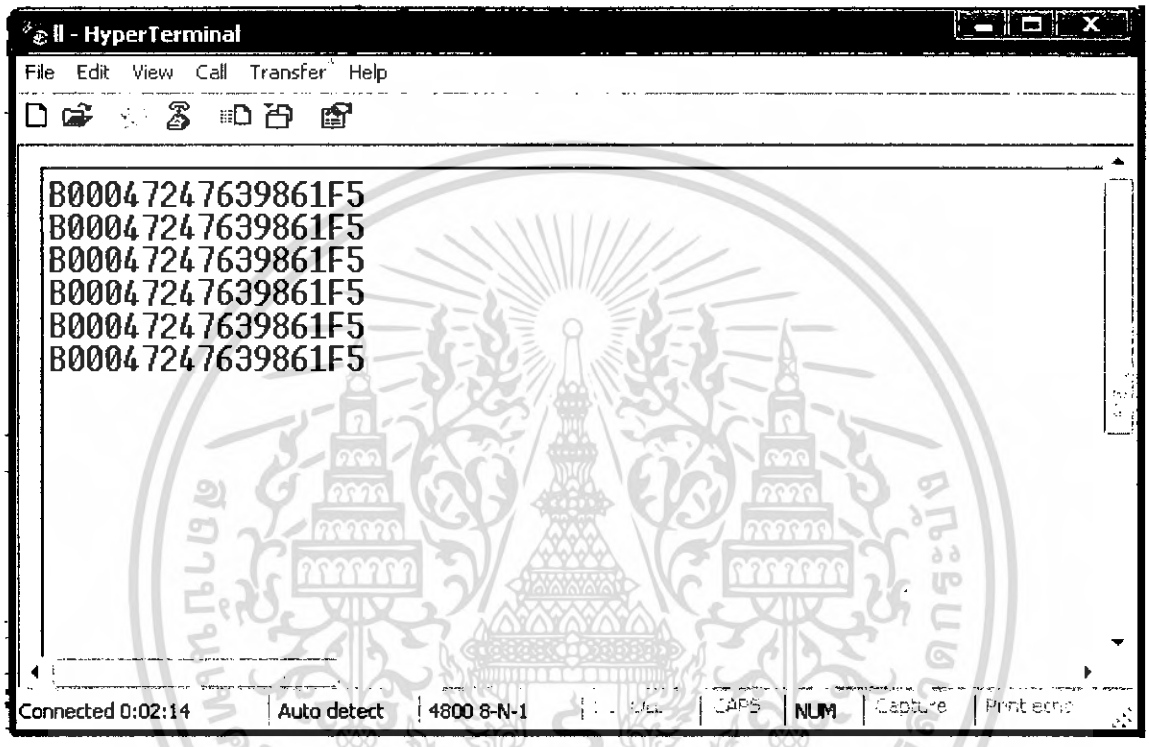
รูปที่ 3.4 บล็อคไดอะแกรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 ผลการทดลองการอ่านข้อมูลของบัตร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

อุปกรณ์ที่ใช้มี 2 ส่วน

1. หัวอ่านการ์ดรีดเดอร์
   - ระบบหัวอ่านแบบไร้สัมผัส (proximity)
   - สามารถกำหนดใช้บัตรได้ 2 วิธี รับบัตรไร้สัมผัสอย่างเดียวและรับบัตรกดรหัส 4 หลัก
   - สามารถทำการประมวลผลรายงานบุคคลเข้าออก
2. อุปกรณ์เปิด-ปิดประตูไฟฟ้า
   - ระบบไฟฟ้ากระแสตรง 12 VDC หรือตามมาตรฐานผู้ผลิต
   - กลอนประตูเป็นชนิดที่ทำงานด้วยไฟฟ้า ปกติจะล็อกตลอดเวลา จะปิดล็อกก็ต่อเมื่อได้รับคำสั่งจากหัวอ่านในกรณีเปิดเข้าหรือกดปุ่มเปิดประตูจากด้านใน กรณีเมื่อกระแสไฟฟ้าขัดข้องผู้ที่อยู่ภายในสามารถเปิดประตูออกมาข้างนอกได้

ผลการทดสอบ

- ผลการทดสอบโปรแกรมโดยให้มีการทดสอบใช้งานจริง ผลอยู่ในระดับเป็นที่พึงพอใจ ใช้งานได้อย่างมีประสิทธิภาพ คือสามารถติดต่อกับสมาร์ทการ์ดแต่ทำการเปิดปิดประตูได้อย่างถูกต้อง
- การทดสอบ Proximity card สามารถทำงานได้อย่างมีประสิทธิภาพ ทำให้ได้ข้อมูลที่ถูกต้อง
- การบันทึกข้อมูลลงฐานข้อมูลมีประสิทธิภาพทำให้ได้ข้อมูลที่ถูกต้องและแม่นยำ

สรุปผลการดำเนินโครงงาน

โครงงานเรื่องการประยุกต์ใช้งานบัตรสมาร์ทการ์ดสำหรับอพาร์ตเมนท์ ได้จัดทำขึ้นเพื่ออำนวยความสะดวกในการควบคุมบริหารและจัดความปลอดภัยให้มีประสิทธิภาพ เนื่องด้วยในปัจจุบันค่าใช้จ่ายด้านแรงงานมีแนวโน้มสูงขึ้นเรื่อยๆ จึงได้มีการจัดทำโครงงานนี้ขึ้นเพื่อปลูกจิตสำนึกในการใช้พลังให้เกิดประโยชน์สูงสุด รวมทั้งช่วยในการครอบคลุมและตรวจสอบบุคคลเข้า-ออกได้โดยสะดวก

โครงงานนี้ได้มีการใช้เทคนิคซึ่งมีการพัฒนาโปรแกรมหลักการเขียนโปรแกรมติดต่อกับอุปกรณ์ภายนอกคือสมาร์ทการ์ดและอุปกรณ์อิเล็กทรอนิกส์อีกหลายอย่าง ส่วนการเขียนโปรแกรมบนวินโดวส์นั้นได้ใช้ในการเก็บข้อมูลของผู้ที่เป็นสมาชิกของระบบ การทำงานของระบบผู้ที่เป็นสมาชิกต้องลงทะเบียนการใช้งานกับผู้ดูแลระบบก่อน เพื่อเป็นการขอโควต้าโดยผู้ที่เป็นสมาชิกจะได้สมาร์ทการ์ดเพื่อใช้การระบุข้อมูลสมาชิกและโควต้า โดยการใช้งานก่อนเข้าอาคารจะมีการใช้สมาร์ทการ์ดเข้ากับอุปกรณ์อ่านเพื่อร้องขอโควต้าที่มีอยู่ว่าสามารถเปิดประตูได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการพัฒนาโครงงานอยู่ในเกณฑ์ดี สามารถทำตามเป้าหมายที่กำหนดไว้ อีกทั้งสามารถพัฒนาต่อเพื่อเพิ่มขีดความสามารถต่อไปได้อีก

ปัญหาอุปสรรคและแนวทางแก้ไข

ปัญหาและอุปสรรคของระบบนี้คือ การทำงานของระบบนี้มีความเสี่ยงสูงจากการติดตั้งและใช้งานจำเป็นต้องมีการออกแบบบรรจุภัณฑ์ที่มีความปลอดภัยสูง อีกทั้งในการใช้งานจริงยังต้องมีการปรับปรุงให้มีความปลอดภัยสูงสุดเพื่อป้องกันไม่ให้เกิดปัญหาการทำงานที่ผิดพลาดภายหลัง แนวทางแก้ไขการทำงานของระบบยังต้องมีการศึกษาพัฒนาจากการใช้งานจริงเพื่อเป็นแนวทางในการแก้ไขข้อผิดพลาดที่อาจเกิดขึ้นในกรณีที่ไม่ได้เตรียมการป้องกันได้

ข้อเสนอแนะในการพัฒนาต่อไป

- ศึกษาปัญหาและอุปสรรคที่เกิดขึ้นกับการพัฒนาของระบบและปรับใช้เทคโนโลยีการสื่อสารการควบคุมให้เหมาะสมกับปัจจุบัน
- ทำการพัฒนาระบบต่อให้สามารถมีฟังก์ชัน โดยให้สามารถมีฟังก์ชันการทำงานที่ซับซ้อนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

[1] ร.ศ.สมยศ จุณณะปิยะ "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์"ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2546

[2] บริษัท อีทีที จำกัด . คู่มือการใช้งานบอร์ดไมโครคอนโทรลเลอร์ รุ่น "CP-JR51AC2 V1.0 & V2.0". [ซีดีรอม].

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. โปรแกรมอ่านข้อมูลของบัตร

```
#include <c51ac2.h>
#include <absacc.h>
#include <assert.h>
#include <ctype.h>
#include <intrins.h>
#include <math.h>
#include <setjmp.h>
#include <stdarg.h>
#include <stddef.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

/********** I/O PORT **********/

#define  SEGM   XBYTE [0xd000]          // segment
#define  KEYI   XBYTE [0xe000]          // Key-input,Dip-sw
#define  DIGT   XBYTE [0xf000]          // Digit,BL,485con,sound,user-
led

sbit     CARDDA = P3^3;
sbit     CARDCK = P3^7;
sbit     CARDCP = P3^6;
sbit     CARDC = P2^6;

/********** INT-RAM WORKING AREA **********/

unsigned char    DIGMEM;                // digit port memory
unsigned char    DISBUF[6];             // display buffer
unsigned char    HEXBUF[3];             // hex buffer
bit              KEYFAG;
unsigned char    STRBUF[3];             // string buffer
unsigned char    n=0;                    //B000472476398 61F5

                                        // DATA MEMORY (89C51RD2)
unsigned char xdata MAGBUF[40];         // magnetic buffer
unsigned char xdata MAGOUT[40];         // magnetic output

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

void dmsec (unsigned int count) {       // mSec Delay
    unsigned int i;                     // Keil CA51 (x2)
    while (count) {
        i = 230; while (i>0) i--;
        count--;
    }
}

void sound (unsigned char freq,int time) {        // Sound Generate
    unsigned char i;
    while (time>0) {
        DIGMEM = DIGMEM & 0xbf;                   // out signal low
        DIGT = DIGMEM;
        for (i=1;i<=freq;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            time--;
        DIGMEM = DIGMEM | 0x40;                    // out signal high
        DIGT = DIGMEM;
        for (i=1;i<=freq;i++)
            time--;
    }
}

void beepk1 (void) {sound (80,1000);}              // beep key 0-f
void beepk2 (void) {sound (60,1000);}              // beep key 10-13
void beeper (void) {sound (150,30000);}            // beep error
void beepok (void) {sound (15,20000);}             // beep ok

unsigned char code SEGTAB[16] =
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,

0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71}; // standard segment code

unsigned char htosx (unsigned char hex) {          // change hex to
segment
    return (SEGTAB[hex]);
}

void htos (void) {                                 // change hex to
segment
    DISBUF[0] = htosx ((HEXBUF[0] & 0xf0) >> 4);   // hex[3] to
disbuf[6]
    DISBUF[1] = htosx (HEXBUF[0] & 0x0f);
    DISBUF[2] = htosx ((HEXBUF[1] & 0xf0) >> 4);
    DISBUF[3] = htosx (HEXBUF[1] & 0x0f);
    DISBUF[4] = htosx ((HEXBUF[2] & 0xf0) >> 4);
    DISBUF[5] = htosx (HEXBUF[2] & 0x0f);
}

unsigned char atohr (unsigned char a) {            // change ascii to hex
(R)

    if ((a>='a' && a<='f') || (a>='A' && a<='F'))
        return ((a + 9) & 0xf);
    else return (a & 0xf);
}

unsigned char atohl (unsigned char a) {            // change ascii to hex
(L)
    a = atohr (a);
    a = a << 4;
    return (a);
}

void clear (void) {                // clear DISBUF
    unsigned char i;
    for (i=0;i<=7;i++)
        DISBUF[i] = 0;
}

unsigned char code KEYTAB[] = {0x70,0x71,0x72,0x73,
                               0xb0,0xb1,0xb2,0xb3,
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                0xd0,0xd1,0xd2,0xd3,
                                0xe0,0xe1,0xe2,0xe3,
                                0x74,0xb4,0xd4,0xe4};
                                // 20 key table

unsigned char scan (void) {             // scan display & get key
    unsigned char i,j,x;
    bit pressok;
    pressok = 0;
    for (i=0;i<=5;i++) {
        DIGMEM = (DIGMEM & 0xf0) | i;   // out digit
        DIGT = DIGMEM;
        SEGM = DISBUF[i];               // out segment
        dmsec(1);
        SEGM = 0;
        x = KEYI & 0x0f;                // check key
        if (x!=0x0f) {
            pressok = 1;
            if (!KEYFAG) {              // new press
                x = (x << 4) | i;
                for (j=0;j<=19;j++) {
                    if (x==KEYTAB[j]) {
                        if (j>0xf) beepk2 (); else beepk1 ();
                        KEYFAG = 1;
                        return (j);
                    }
                }
            }
        }
    }
    if (!pressok) KEYFAG = 0;           // No any key press , clear
KEYFAG (release)
    return (0xff);
}

void waitdot (void) {                   // display wait dot
    DIGMEM = DIGMEM & 0xf0;
    DIGT = DIGMEM;
    SEGM = 0x80;
}

/********** SPECIFIC FUNCTION **********/

void magclear (void) {                  // magnetic clear buffer
    unsigned char i;
    for (i=0;i<=39;i++) {
        MAGBUF[i] = 0;
        MAGOUT[i] = 0;
    }
}

void magread (void) {                  // magnetic read raw data
    unsigned char a,i,j;                 // to MAGBUF
    bit dat,first;
    while (CARDCP);                      // wait for CP=0
    a = 0;
    i = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    j = 0;
    first = 1;
    while (!CARDCP) {
        while (CARDCK) {                    // wait for CK=0
            if (CARDCP) return;
        }
        _nop_ ();
        _nop_ ();
        dat = CARDDA;
        if (first) {
            if (!dat) first = 0;            // don't care first bit = 1
        }
        if (~first) {
            a = a >> 1;
            if (dat) a = a | 0x80;
            i++;
            if (i==5) {
                a = a >> 3;
                a = ~a & 0x1f;
                MAGBUF[j] = a;
                j++;
                i = 0;
            }
        }
        while (!CARDCK);                    // wait for CK=1
    }
}

bit magchk (void) {                         // magnetic check parity
    unsigned char a,b,i,j;                  // and change to ascii (to
MAGOUT)
    bit flag;
    for (i=0;i<=39;i++) {
        a = MAGBUF[i];
        if (a==0) break;
        b = 0;
        for (j=0;j<=4;j++) {
            flag = a & 1;
            a = a >> 1;
            if (flag) b++;
        }
        if ((b % 2)==0) return (1);
        a = MAGBUF[i] & 0xf;
        if (a>=0xa) a = (a - 9) | 0x40; else a = a | 0x30;
        MAGOUT[i] = a;
    }
    if (MAGOUT[0]!='B') return (1);
    else return (0);
}

void magreverse (void) {                                    // magnetic
reverse data
    unsigned char a,b,i,j,p,x,y;
    bit dat,skip;
    for (i=0;i<=39;i++) MAGOUT[i] = 0;                      // clear
MAGOUT
    i = 40; p = 0; x = 0; y = 0; skip = 1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while (i>0) {
        i--;
        a = MAGBUF[i];
        if (skip && a==0);                                    // nothing
        else {
            a = a << 3;
            for (j=1;j<=5;j++) {
                b = a & 0x80; a = a << 1;
                if (b==0x80) dat = 1; else dat = 0;
                if (skip && dat==1) skip = 0;                 // find
first bit = 1
                if (!skip) {                                  // reverse
data
                    if (dat) p = p | 0x80; else p = p & 0x7f;
                    p = p >> 1; y++;
                    if (y==5) {
                        MAGOUT[x++] = p >> 2;
                        p = 0; y = 0;
                    }
                }
            }
        }
    }
    if (y>0) {                                                // check end data

        while (y!=5) { p = p >> 1; y++; }
        MAGOUT[x] = p >> 2;
    }
    for (i=0;i<=39;i++) MAGBUF[i] = MAGOUT[i];                // move to
MAGBUF
}

void start (void) {                          // start program
    SEGM = 0;                                // clear segment
    DIGMEM = 0xff;                           // Digit buffer port (don't care
rs485)
    DIGT = DIGMEM;

    SCON = 0x52;                             // set RS232 parameter (19200)
    TMOD = 0x20;
    TH1  = 0xf6;
    TR1  = 1;
}

/********** MAIN **********/

void main (void) {
    unsigned char i;
    bit error;
      CARDC=1;

    start ();
    while (1) {
        waitdot ();
        magclear ();                         // clear magnetic buffer
        magread ();                          // read raw data
        error = magchk ();                   // check data
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (error) {                        // if error type reverse
            magreverse ();
            error = magchk ();
        }
        if (error) {                        // read error
            beeper ();                      // beep error
            clear ();
            DISBUF[0] = 0x79;               // Err
            DISBUF[1] = 0x50;
            DISBUF[2] = 0x50;
        }
       else {
            beepok ();                          // beep ok

                                                // display first 6 digit on
LED display
            HEXBUF[0] = atohl (MAGOUT[1]) | atohr (MAGOUT[2]);
            HEXBUF[1] = atohl (MAGOUT[3]) | atohr (MAGOUT[4]);
            HEXBUF[2] = atohl (MAGOUT[5]) | atohr (MAGOUT[6]);
            Pos ();
          for (i=0;i<=39;i++) {             // send all data to RS232
               if (MAGOUT[i]==0); //break;
               printf ("%c",MAGOUT[i]);
               SBUF=MAGOUT[i];

                  CARDC=0
           }
           CARDC=1;
            printf ("\n");
        }
       while (_testkey ()==0xff);
       if (i==0x1b) return;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้