

การควบคุมการเข้า-ออกด้วยสมาร์ทการ์ด
ACCESS CONTROL SYSTEM WITH SMART CARD



โดย
นาย กิตติพงษ์ แก้วเจริญ
นาย นิวัฒน์ ฉันทะสมบุญเสรี

46557
11/2546
2546

เลขหมู่.....
เลขทะเบียน...46557
วัน, เดือน, ปี... 4 เม.ย. 2546

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบเสนอปริญญาบัตร

หัวข้อวิทยานิพนธ์

การควบคุมการเข้า-ออกด้วยสมาร์ตการ์ด

นักศึกษา

นายกิตติพงศ์ แก้วเจริญ รหัสประจำตัว 43015800

นายนิวัฒน์ ฉันทะสมบุญเสรี รหัสประจำตัว 43015821

อาจารย์ผู้ควบคุมวิทยานิพนธ์

อ.พิชญ สุพรรณกุล

ระดับการศึกษา

ปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชา

สาขาวิชาเทคโนโลยีโทรคมนาคม

ปีการศึกษา

เทคนิคอุตสาหกรรม

2544

ปริญญาบัตรฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

(อ.พิชญ สุพรรณกุล)

อาจารย์ผู้ควบคุมปริญญาบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์

การควบคุมการเข้า-ออกด้วยสมาร์ทการ์ด

นักศึกษา

นายกิตติพงษ์ แก้วเจริญ รหัสประจำตัว 43015800
นายนิวัฒน์ ฉันทะสมบุญเสรี รหัสประจำตัว 43015821

อาจารย์ผู้ควบคุมวิทยานิพนธ์ อ.พิชญ สุพรรณกุล

ระดับการศึกษา

ปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีโทรคมนาคม

ภาควิชา

เทคนิคอุตสาหกรรม

ปีการศึกษา

2544

บทคัดย่อ

วิทยานิพนธ์นี้แสดงคุณสมบัติของสมาร์ทการ์ดและออกแบบสร้างชุดจำลองระบบรักษาความปลอดภัยโดยใช้สมาร์ทการ์ดเก็บข้อมูลรหัสนักศึกษา ชุดจำลองประกอบด้วยเครื่องอ่านเขียนข้อมูลลงสมาร์ทการ์ด และอุปกรณ์เชื่อมต่อระหว่างสมาร์ทการ์ดกับคอมพิวเตอร์ ส่วนที่ควบคุมการอ่านเขียนข้อมูลได้ใช้โปรแกรมวิซวลเบสิก (VB) บันทึกข้อมูลที่เป็นรหัสนักศึกษาลงสมาร์ทการ์ดพร้อมกับบันทึกข้อมูลต่าง ๆ เช่น ชื่อ เวลาเข้า-ออก ลงในฐานข้อมูล เมื่อมีการนำบัตรสมาร์ทการ์ดมาใช้จะทำการอ่านข้อมูลจากบัตรมาเปรียบเทียบกับข้อมูลที่มีอยู่ในฐานข้อมูลและบันทึกเวลาเข้าออก

THESIS TITLE **ACCESS CONTROL SYSTEM WITH SMART CARD**

STUDENT **Mr. Kittipong Keawchareon** **No. 43015800**

Mr. Niwat **Chantasomboonsaeree** **No. 43015821**

ADVISOR **Mr. Pichaya Supanakoon**

COURSE **Bachelor of Industrial Technology in Telecommunication**

DEPARTMENT **Industrial Technology**

YEAR **2001**

ABSTRACT

This thesis presents the characteristics of smartcard and design of the simulate security system with storing records of student's ID . This system consists of the smartcard read and write and the interface between smartcard and computer. Visual Basic Language is employed for R/W controller to retain the student's ID in smartcard and other records such as the enter time and exit time are kept in the database. When smartcard is used, data in smartcard are read and compared with those in the database and the enter time and exit time are also recorded in the database.

กิตติกรรมประกาศ

ปริญญาบัตรฉบับนี้อาจจะไม่ประสบความสำเร็จด้วยดีหากไม่ได้รับความกรุณาจากบุคคล
หลายๆฝ่ายโดยเฉพาะคำแนะนำและสั่งสอนที่ดีและความช่วยเหลือดูแลเอาใจใส่เสมอมา ขอขอบคุณ
อ.พิชญ สุพรรณกุล ซึ่งเป็นอาจารย์ที่ปรึกษาให้คำชี้แนะจนปริญญาบัตรสำเร็จไปด้วยดีและอาจารย์ใน
ภาควิชาเทคนิคอุตสาหกรรมที่เป็นผู้ให้ความรู้ทุกท่าน

บุคคลอันเป็นบุคคลที่สำคัญคือ บิดา มารดา อันเป็นที่เคารพยิ่ง ซึ่งได้เลี้ยงดูพร้อมกับให้โอกาส
ในการศึกษาเป็นอย่างดีและยังให้กำลังใจ เอาใจใส่เสมอมาในทุกๆด้านอันหาที่เปรียบมิได้ สถาบัน
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อันเป็นสถานที่ให้ความรู้ที่ดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
1.4 ขั้นตอนการดำเนินงาน	2
บทที่ 2 หลักการทำงานและทฤษฎีเบื้องต้น	3
2.1 บัตรสมาร์ตการ์ด	3
2.1.1 ประเภทของหน่วยความจำภายในการ์ด	3
2.1.2 การเลือกบัตรที่จะใช้	4
2.1.3 ข้อดีของบัตรสมาร์ตการ์ด	5
2.1.3 ลักษณะโดยรวมของมาตรฐาน ISO7816	15
2.2 ความรู้เบื้องต้นเกี่ยวกับระบบบัส I2C	15
2.2.1 คุณสมบัติโดยทั่วไปของบัส I2C	16
2.2.2 หลักการของบัส I2C	17
2.2.3 สภาวะที่เกิดขึ้นบนบัส I2C	18
2.2.4 การทำงานบนบัส I2C	19
2.2.5 การอ้างถึงแบบ 7 บิต (7- bit addressing)	20
2.2.6 การอ้างถึงแบบ 10 บิต (10- bit addressing)	21
2.2.7 การเขียนโปรแกรมติดต่อบัส I2C	22
2.3 พอร์ตอนุกรม	24
2.3.1 การสื่อสารแบบอนุกรม	24
2.3.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232	28
2.3.3 UART	32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4	วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232	33
2.3.5	ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232	34
2.3.6	แอดเดรสของพอร์ตอนุกรม	35
2.4	การเขียน โปรแกรมเพื่อใช้งานพอร์ต	36
บทที่ 3	หลักการและวิธีการออกแบบ	42
3.1	ออกแบบโปรแกรมส่วนของไมโครคอนโทรลเลอร์ 8051	43
3.1.1	โปรแกรมย่อยในการเขียนข้อมูล	44
3.1.2	โปรแกรมย่อยในการอ่านข้อมูล	45
3.2	ออกแบบโปรแกรมส่วน Visual Basic	46
3.2.1	การใช้ VB ส่งข้อมูล	46
3.2.2	การใช้ VB รับข้อมูล	47
3.3	ออกแบบวงจรไมโครคอนโทรลเลอร์เพื่อติดต่อสมาร์ตการ์ด	48
บทที่ 4	การทดลองวงจรและผลลัพธ์ที่ได้	49
4.1	ลักษณะโปรแกรมที่รันครั้งแรก	49
4.2	การเพิ่มข้อมูลในฐานข้อมูลและเขียนข้อมูลรหัสลงสมาร์ตการ์ด	50
4.3	การโปรแกรมเพื่อรองรับการอ่านค่าจากสมาร์ตการ์ด	52
4.4	การแสดงรายละเอียดฐานข้อมูลวันเวลา	53
บทที่ 5	สรุปและวิจารณ์การทดลอง	54
5.1	สรุป	54
5.2	วิจารณ์	54
ภาคผนวก		
บรรณานุกรม		

สารบัญรูปร่าง

หน้า

บทที่ 2 หลักการทำงานและทฤษฎีเบื้องต้น	6
2.1 แสดงขนาดของบัตรตามมาตรฐาน ISO 7816	7
2.2 แสดงลักษณะการงอบัตร	7
2.3 แสดงลักษณะชีพที่ติดบนการ์ด	7
2.4 แสดงขนาดของหน้าสัมผัสตามมาตรฐาน ISO 7816	8
2.5 แสดงตำแหน่งมาตรฐาน ISO 7816	8
2.6 ไคอะแกรมเวลาของสัญญาณการรีเซตการ์ด	13
2.7 แสดงการเชื่อมต่อของอุปกรณ์ต่าง ๆ บนระบบบัส	16
2.8 แสดงการต่อความต้านทานพูลอัพบนสายสัญญาณ	17
2.9 แสดงไคอะแกรมเวลาของสถานะต่าง ๆ	19
2.10 แสดงรูปแบบการกำหนดแอดเดรสที่ใช้ในการอ้างแบบ 7 บิต	20
2.11 รูปแบบของข้อมูลที่ใช้ติดกับอุปกรณ์บัส I2C เมื่อใช้การอ้างแบบ 7 บิต	21
2.12 รูปแบบของข้อมูลที่ใช้ติดกับอุปกรณ์บัส I2C เมื่อใช้การอ้างแบบ 10 บิต	21
2.13 Flowchart ของโปรแกรมย่อยของสถานะเริ่มต้นและสถานะหยุด	23
2.14 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม	25
2.15 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส	26
2.16 การจัดขาของคอนเน็คเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25	29
2.17 การต่ออุปกรณ์ภายนอกกับ พอร์ตอนุกรมของคอมพิวเตอร์ในแบบต่าง ๆ	31

บทที่ 3 หลักการและวิธีการออกแบบ

3.1 รูปแบบโดยรวมของโครงการนี้	42
3.2 Flowchart ของโปรแกรมที่เขียนลงไมโครคอนโทรลเลอร์	43
3.3 แสดง Flowchart ของโปรแกรมย่อยในการเขียนโปรแกรม	44
3.4 แสดงโปรแกรมย่อยในการอ่านข้อมูล	45
3.5 Flowchart การส่งข้อมูลให้ไมโครคอนโทรลเลอร์ และบันทึกรหัส ชื่อ ลงฐานข้อมูล	46
3.6 Flowchart ของโปรแกรมย่อยในการรับค่าที่ส่งมาจากไมโครคอน โทรลเลอร์	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 แสดงวงจรไมโครคอนโทรลเลอร์ที่ใช้ติดกับสมาร์ทการ์ด	48
บทที่ 4 กาททดลองและผลลัพธ์ที่ได้	
4.1 แสดงรูปตอนรันโปรแกรมครั้งแรก	49
4.2 แสดงการเพิ่มข้อมูลให้ฐานข้อมูลและเขียนข้อมูลให้กับบัตร	50
4.3 แสดงตารางฐานข้อมูลชื่อ ,รหัสและเวลาเข้า – ออกที่ยังไม่มีข้อมูลใด ๆ	51
4.4 แสดงข้อมูลที่จะเพิ่มในฐานข้อมูล	51
4.5 แสดงข้อมูลที่เก็บในตารางฐานข้อมูล	51
4.6 ภาพแสดงว่าไม่พบข้อมูล	52
4.7 แสดงตารางฐานข้อมูลที่บันทึกเวลาเข้าและออก	52
4.8 แสดงรายละเอียดของเวลาเข้าออก	53



สารบัญตาราง

หน้า

บทที่ 2 หลักการทำงานและทฤษฎีเบื้องต้น

2.1 คุณลักษณะสัญญาณของ I/O ภายใต้การปฏิบัติงานปกติ	10
2.2 คุณลักษณะสัญญาณของ V_{pp} ภายใต้การปฏิบัติงานปกติ	11
2.3 คุณลักษณะสัญญาณของ CLK ภายใต้การปฏิบัติงานปกติ	12
2.4 คุณลักษณะสัญญาณของ RST ภายใต้การปฏิบัติงานปกติ	12
2.5 คุณลักษณะสัญญาณของ VCC ภายใต้การปฏิบัติงานปกติ	13
2.6 แสดงบิตพาริตีของข้อมูล	27
2.7 แสดงชื่อและทิศทางสัญญาณของ DB9 และ DB 25	30
2.8 แสดงข้อมูลในแอดเดรส 000: 0411 H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม	35



บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

ระบบรักษาความปลอดภัยของสถานที่ต่างๆ เราสามารถกำหนดและควบคุมการเข้า-ออกของแต่ละบุคคล ได้ตามต้องการโดยใช้กุญแจบัตร (key card) ที่ทำหน้าที่เปรียบเสมือนเป็นกุญแจและปัจจุบันมีบัตรหลายชนิดที่นิยมนำมาใช้งานลักษณะดังกล่าว เช่น บัตรแถบแม่เหล็ก บัตรบาร์โค้ด แต่ในโครงการนี้จะใช้บัตรสมาร์ทการ์ดเนื่องจากปัจจุบันเทคโนโลยีของวงจรรวม (Integrated Circuit) ได้พัฒนาไปไกลมากถึงขนาดย่อขนาดลงมา อยู่บนเคดิตการ์ด เรียกว่าสมาร์ทการ์ด (Smart card) จึงได้รับความนิยมที่จะใช้การ์ดที่บรรจุชิปไอซีแทนบัตรแถบแม่เหล็กในงานด้านต่างๆ มากขึ้นเรื่อยๆ เนื่องจากความสามารถที่เหนือกว่าบัตรแถบแม่เหล็ก(Magneticstripcard)ในหลายๆด้านเช่นความสามารถในการประมวลผลข้อมูลได้ในตัวเองไม่มีผลกระทบของสนามแม่เหล็กไฟฟ้าขนาดของหน่วยความจำสมาร์ทการ์ดจึงได้ถูกนำมาใช้อย่างแพร่หลายไม่ว่าจะใช้ในเชิงธุรกิจการเก็บข้อมูลการรักษาความปลอดภัย และการเก็บคุณสมบัติของสิ่งต่างๆ เป็นต้น

โครงการนี้จึงทำการศึกษาเกี่ยวกับการนำคุณสมบัติด้านหนึ่งของสมาร์ทการ์ดมาใช้งานทางด้านระบบรักษาความปลอดภัยแบบใช้สมาร์ทการ์ด โดยจะนำการสร้างเครื่องอ่านและบันทึกข้อมูลบนบัตรสมาร์ทการ์ดเพื่อกำหนดการเข้า-ออกของแต่ละบุคคล

1.2 วัตถุประสงค์

- ศึกษาคุณสมบัติต่างๆของสมาร์ทการ์ด
- นำคุณสมบัติของสมาร์ทการ์ดมาประยุกต์ใช้
- ศึกษารูปแบบของสัญญาณและการสื่อสารข้อมูลของบัตรสมาร์ทการ์ดตามมาตรฐานISO
- ศึกษาการทำงานของโปรแกรม Visual Basic เพื่อใช้อ่าน/เขียนข้อมูลบนบัตรสมาร์ทการ์ด

1.3 ขอบเขตของโครงการ

ในโครงการนี้จะนำเสนอการนำสมาร์ทการ์ดมาใช้ในรูปแบบ key card โดยการบันทึกรหัสนักเรียนศึกษาลงบนบัตรและเก็บข้อมูลที่เป็น ชื่อ นามสกุล และจะบันทึกเวลาเข้าและออกไว้ด้วย โดยโครงการ

นี้จะประกอบด้วยMicrocontroller8051ซึ่งเป็นส่วนที่ใช้อ่านและเขียนข้อมูลลงบนสมาร์ทการ์ดและส่วนที่เป็นการเก็บข้อมูลต่างๆ เช่น ชื่อ นามสกุลและบันทึกเวลาเข้าออกโดยใช้โปรแกรม Visual Basic

1.4 ขั้นตอนการดำเนินงาน

การดำเนินงานนั้นจะเริ่มจากการศึกษาลักษณะคุณสมบัติของสมาร์ทการ์ดแบบต่างๆจากนั้นต้องเลือกชนิดของบัตรที่จะนำมาใช้งานและในโครงการนี้เลือกใช้บัตรแบบ EEPROM เมื่อเลือกชนิดของบัตรที่ต้องการได้แล้วก็เขียนโปรแกรมใน 8051 เพื่อติดต่อกับสมาร์ทการ์ดโดยการเขียนโปรแกรมติดต่อเขียนตามโปรโตคอลของบัตรแต่ละชนิดจากนั้นเราจะต้องกำหนดรูปแบบของData base ที่เราจะใช้เก็บข้อมูลและเขียนโปรแกรม Visual Basic เพื่อทำการประมวลผลที่ได้จาก 8051 และเก็บข้อมูล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการงานและทฤษฎีเบื้องต้น

2.1 บัตรสมาร์ตการ์ด

บัตรสมาร์ตการ์ด คือ บัตรพลาสติกขนาดมาตรฐาน ISO ที่มีวงจรรีเลย์ทรอนิกส์ หรือที่เรียกว่า Microchip ติดอยู่เพื่อเก็บข้อมูล หรือประมวลผลข้อมูลได้ภายในตัว โดยวิธีการเข้ารหัสตามมาตรฐานของ DES Algorithm (Data Encryption Standard) จึงทำให้ระบบความปลอดภัยของระบบสูง และทำให้การปลอมแปลงตัวบัตรหรือข้อมูลเป็นไปได้ยาก

สมาร์ตการ์ด แบ่งเป็น 3 ประเภทตามคุณสมบัติและระดับของความสามารถ ได้แก่

1. memorycard คือ บัตรที่มีวงจรมหน่วยความจำบรรจุอยู่ภายในเพื่อใช้เก็บข้อมูลซึ่งมีเปลี่ยนแปลงได้อย่างจำกัด เช่น สำหรับบันทึกข้อมูลค่าเงินซึ่งจะถูกหักออกหรือลบมูลค่าลงเมื่อนำไปใช้

2. Microprocessor Card คือ บัตรที่มีวงจรรับซ็อนกว่าชนิดแรกเพื่อเก็บข้อมูลและระบบประมวลผลในตัวเอง เรียกว่า COS (Card Operating System) สำหรับข้อมูลที่มีปริมาณมากหรือต้องการความปลอดภัยสูงหรือต้องการแก้ไขข้อมูลในระดับที่สูงขึ้นเช่น ข้อมูลส่วนบุคคล ข้อมูลด้านการเงิน บัญชี PIN ข้อมูลสุขภาพ ฯลฯ

3. Super Smart Card คือ บัตรที่มีแผงวงจรมหน่วยความจำ และหน่วยประมวลผลที่บัตรเครดิตจนถึงรุ่นที่สามารถทำงานได้เสมือนเครื่องคอมพิวเตอร์ขนาดเล็กย่อๆ ได้ปัจจุบันอยู่ในขั้นการทดลองพัฒนา เพราะราคายังสูงมากเกินกว่าที่จะนำมาใช้งานในเชิงพาณิชย์

2.1.1 ประเภทของหน่วยความจำภายในการ์ด

ภายในบัตรนั้นประกอบไปด้วยหน่วยความจำชนิดต่าง ๆ เพื่อเก็บค่าข้อมูล , โปรแกรมและข้อมูลชนิดพิเศษเพื่อใช้ในการทำงานต่าง ๆ ปกติจะมีหน่วยความจำถาวร และหน่วยความจำชั่วคราว ทั้งชนิดที่ต้องการไฟเลี้ยง (Volatile Memory) และชนิดที่ไม่ต้องการไฟเลี้ยง (Nonvolatile Memory)

1. หน่วยความจำชนิดที่ต้องการไฟเลี้ยง (Volatile Memory)

หน่วยความจำประเภทนี้ประกอบด้วย ไดนามิกและสแตติกแรม ซึ่งถูกนำมาใช้งานในบัตรหลายชนิด ข้อแตกต่างระหว่างไดนามิกและสแตติกแรม คือ สแตติกแรมไม่ต้องการการรีเฟรช และเก็บข้อมูลได้โดยใช้พลังงานต่ำ แต่ข้อมูลจะสูญหายถ้าไม่มีไฟเลี้ยงซึ่งสแตติกแรมมักนิยมใช้ในบัตรที่มีแบตเตอรี่

ภายในส่วนไดนามิกแรมใช้งานกับบัตรที่ต้องการไฟเลี้ยงภายนอก (จากหน้าสัมผัสของบัตร) เพื่อใช้งานเป็นที่เก็บข้อมูลชั่วคราวในการคำนวณ

2. หน่วยความจำชนิดที่ไม่ต้องการไฟเลี้ยง (Nonvolatile Memory)

บัตรส่วนใหญ่นิยมใช้หน่วยความจำชนิดนี้เพราะข้อมูลต่าง ๆ ภายในสามารถจัดเก็บไว้ได้แม้จะไม่มีไฟเลี้ยง ซึ่งไม่มีการนำเทคโนโลยีเหล่านี้มาใช้งานในการเก็บข้อมูลประเภทต่าง ๆ ดังนี้

ROM (Read only Memory) ใช้เก็บข้อมูลที่ถาวร ซึ่งบันทึกในขณะที่ผลิตชิพ ซึ่งประกอบไปด้วยข้อมูลประจำบัตร, รหัสคำสั่ง, โปรแกรมและอื่น ๆ ข้อดีของ ROM คือ ราคาไม่แพงมีความน่าเชื่อถือสูงและเก็บข้อมูลได้จำนวนมาก ซึ่งถ้าเราต้องการเปลี่ยนแปลงข้อมูลใด ๆ ก็ต้องกระทำร่วมกับผู้ผลิตชิพ

PROM (Programmable ROMs) หน่วยความจำนี้สามารถเขียนได้ 1 ครั้ง ผู้ผลิตบัตรสามารถป้อนข้อมูลได้เอง จึงทำให้สะดวกกว่าแบบแรก เพราะสามารถป้องกันไม่ให้ผู้อื่นทราบ ข้อมูลที่ต้องการปกปิดได้

EEPROM (Electrically Erasable Programmable ROMs) หน่วยความจำชนิดนี้สามารถบันทึกค่าและแก้ไขใหม่ได้โดยใช้สัญญาณไฟฟ้าซึ่งหน่วยความจำชนิดนี้มีการนำมาประยุกต์ใช้งานมาก เพราะมีการทำงานคล้ายแรม แต่ข้อมูลภายในไม่สูญหายไปขณะไม่มีไฟเลี้ยง แต่มีราคาสูงกว่ามากและปัญหาหลักของการนำ EEPROM มาใช้งานก็คือในการลบและเขียนใหม่จะต้องใช้ขนาดแรงดันที่สูงกว่าปกติ คือ ค่าแรงดันมาตรฐานในการอ่านและเขียนจะอยู่ที่ 5 โวลต์ แต่ถ้าการลบและเขียนใหม่จะต้องใช้ขนาดแรงดัน 21 โวลต์ ซึ่งทำให้ต้องหาแหล่งจ่ายแรงดันมาใช้ในงานนี้เพิ่มขึ้นอีกชุดหนึ่ง

Flash Memory เป็นเทคโนโลยีที่เกิดจากการพัฒนา EEPROM แต่มีขนาดความจุสูงจึ้นราคาถูกลง และมีความน่าเชื่อถือสูงซึ่ง Flash Memory นี้สามารถใช้งานขนาดแรงปกติในการลบและเขียนใหม่ได้ทันทีและสามารถลบข้อมูลเฉพาะแอดเดรสได้อีกด้วยจึงเหมาะที่จะนำมาใช้ในส่วนข้อมูลที่มีการเปลี่ยนแปลงค่าบ่อยๆ

2.1.2 การเลือกบัตรที่จะใช้

การเลือกบัตรที่จะใช้พิจารณาถึงความเหมาะสมกับการใช้งาน (Application) ซึ่งการใช้งานแต่ละชนิดจะมีความต้องการต่าง ๆ กันในด้านของปริมาณข้อมูลที่ใช้ วิธีการนำข้อมูลมาใช้อาจเป็นการให้อ่านได้อย่างเดียว (Read Only) หรือสามารถเขียน (Read and Write) และแก้ไขข้อมูลได้ (Update) ข้อพิจารณาอีกประการหนึ่งคือจำนวนการใช้งานในบัตรจะมีเพียงหนึ่งหรือหลาย Application ก็ได้ ซึ่ง

เป็นส่วนหนึ่งที่ต้องตัดสินใจในการเลือกใช้บัตรให้มีความสามารถพอเพียงกับการใช้งาน ซึ่งในปัจจุบันนี้บัตรประจำตัวนักศึกษาที่ให้อยู่สามารถนำมาใช้งานได้ทันที

2.1.3 ข้อดีของบัตรสมาร์ทการ์ด

เนื่องจากการออกแบบให้มีความสามารถที่แถบแม่เหล็ก และบัตรแสงเลเซอร์ (Optical Card) ไม่สามารถทำได้ บัตรสมาร์ทการ์ดจึงมีความได้เปรียบดังนี้

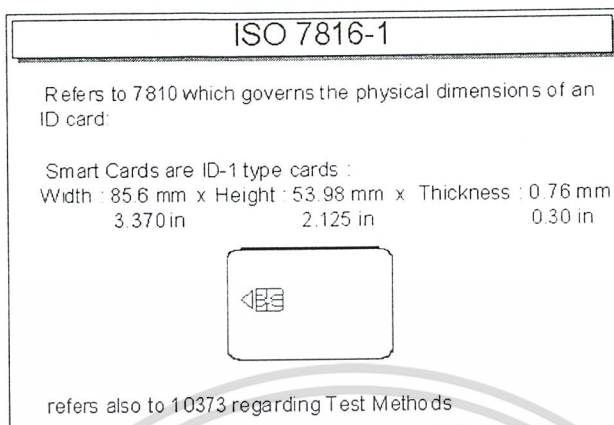
1. เก็บข้อมูลได้มากกว่าทั้งในรายละเอียด และปริมาณ
2. มีหน่วยความจำ และหน่วยประมวลผลในตัวเอง
3. สามารถเปลี่ยนแปลงแก้ไขข้อมูลได้ตามความต้องการใช้งาน
4. ป้องกันการปลอมแปลงได้ดีกว่า
5. ข้อมูลในไมโครชิปไม่ถูกทำลายลงได้ง่าย
6. สร้างจากเทคโนโลยีสมัยใหม่ และมีการพัฒนาอยู่เสมอ
7. อายุการใช้งานนานกว่า
8. ลดค่าใช้จ่ายในการดำเนินงานในระยะยาว

2.1.4 ลักษณะโดยรวมของมาตรฐาน ISO 7816

เราสามารถแบ่งแยกมาตรฐาน ISO 7816 ได้เป็น 3 ส่วน

- ISO 7816 - 1 เป็นการกำหนดลักษณะทางกายภาพของบัตร
- ISO 7816 - 2 เป็นการกำหนดขนาดมิติและตำแหน่งหน้าสัมผัสของบัตร
- ISO 7816 - 3 เป็นการกำหนดลักษณะสัญญาณและการสื่อสารทาง protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงขนาดของบัตรตามมาตรฐาน ISO 7816-1

มาตรฐาน ISO 7816-1

มาตรฐาน ISO 7816 นั้นได้กำหนดคุณสมบัติทางกายภาพด้านต่างๆ ไว้มาก แต่ขณะนี้เราจะสนใจลักษณะคุณสมบัติที่ทดสอบได้

1. แสง UV

การป้องกันต่างๆที่นอกเหนือจากระดับแสง UV เป็นหน้าที่ของผู้ผลิตการ์ด

2. พื้นผิวของหน้าสัมผัส

ส่วนต่างๆของหน้าสัมผัสทั้งหมดที่ติดอยู่กับการ์ดจะห่างจากกันน้อยกว่า 0.1 มม

3. ความคงทน (ของการ์ดและหน้าสัมผัส)

พื้นผิวการ์ด และ ส่วนประกอบต่างๆบนการ์ดจะต้องมีความทนทาน ไม่เสียหายในการใช้งานปกติพื้นผิวต้องไม่เกิดการเสียหายที่เกิดจากแรงกดขนาด 1.5 N ของบอลเหล็กขนาด 1.5 มม

4. ค่าความต้านทานทางไฟฟ้า

ค่าความต้านทานทั้งหมดจะถูกวัดระหว่างจุดต่างๆ 2 จุด ต้องไม่มากกว่า 0.5 โอห์ม ซึ่งค่ากระแสต่างๆจะอยู่ในช่วง 5 μ A-300 mA

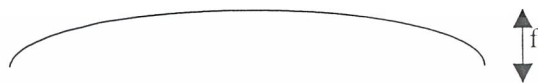
5. สนามแม่เหล็ก

ชิพบนการ์ดจะต้องไม่เสียหายที่เกิดขึ้นจากสนามแม่เหล็กขนาด 79500 A.tr/m

6. สนามไฟฟ้าสถิต

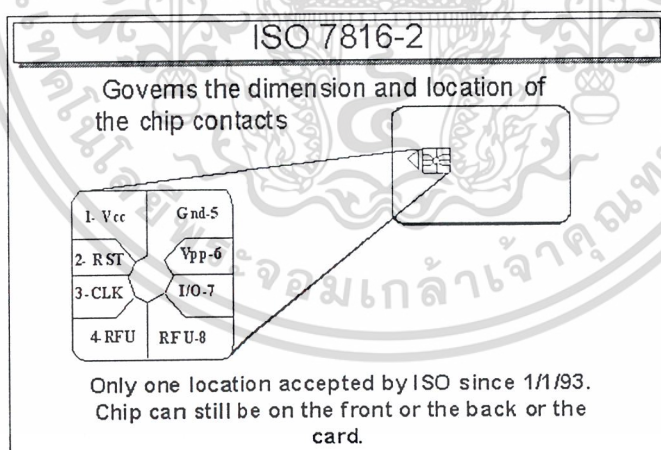
การ์ดจะต้องไม่เกิดความเสียหายจากการ discharge แรงดันขนาด 1500 V ของ C ขนาด 100 pF ผ่าน R 1500 โอห์ม

7. การรบกวนได้สูงสุดของการ์ด



รูปที่ 2.2 แสดงลักษณะการรบกวนของบัตร

- ด้านยาวของการ์ด
ทำการรบกวน $f = 2$ cm จำนวน 30 ครั้ง/นาทีก
 - ด้านสั้นของการ์ด
ทำการรบกวน $f = 1$ cm จำนวนรบกวน 30 ครั้ง/นาทีก
- การ์ดจะต้องใช้งานได้อย่างดีและต้องไม่มีรอยร้าว หลังจากการรบกวน 1000

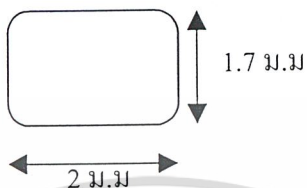


รูปที่ 2.3 แสดงลักษณะชิพที่ติดบนการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

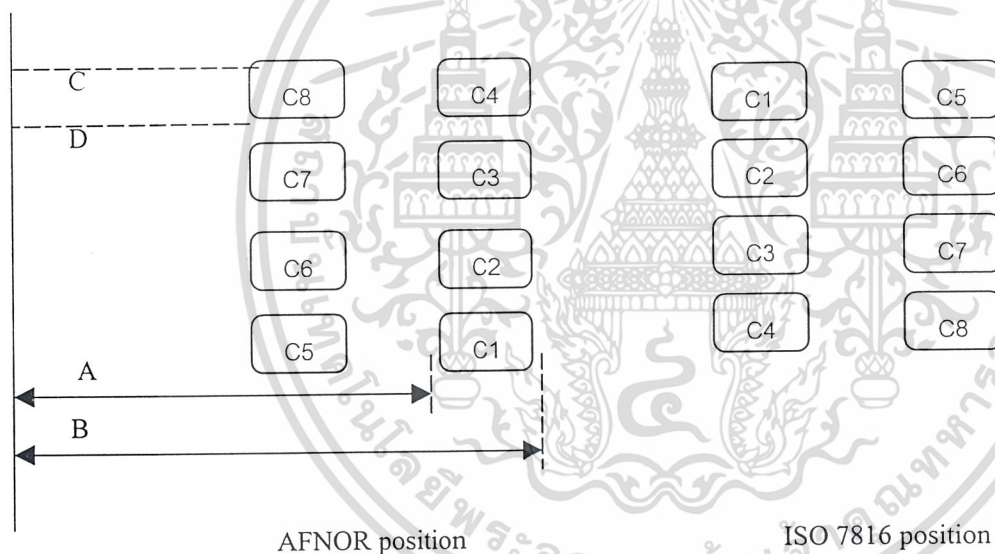
มาตรฐาน ISO 7816-2

1. ขนาดของหน้าสัมผัสของชิพ



รูปที่ 2.4 แสดงขนาดของหน้าสัมผัสตามมาตรฐาน ISO 7816

2. ตำแหน่งของขา



รูปที่ 2.5 แสดงตำแหน่งของขาคตามมาตรฐาน ISO 7816

3. ชื่อของขาที่อยู่บนสมิทการ์ด

C1 : VCC = 5 V	C5 : Gnd
C2 : Reset	C6 : Vpp
C3 : Clock	C7 : I/O
C4 : RFU	C8 : RFU

ตำแหน่ง AFNOR จะมีความเกี่ยวข้อง และสามารถใช้กับการ์ดแถบแม่เหล็กที่มีอยู่ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรฐาน ISO 7816 – 3

1. ลักษณะสัญญาณไฟฟ้า

I/O : input หรือ output สำหรับข้อมูลแบบอนุกรม ที่จะเข้าสู่ IC ภายในการ์ด

VPP : programming voltage input (เลือกใช้โดยการ์ด)

GND : Ground

CLK : สัญญาณนาฬิกา หรือ สัญญาณเวลา (กำหนดการใช้โดยการ์ด)

RST : สามารถใช้สัญญาณรีเซตจากอุปกรณ์ interface หรือใช้วงจรควบคุมการรีเซตที่มีอยู่ภายใน

VCC : Power supply input

ความหมายต่างของคำย่อ

Vih : แรงดันไฟ input ระดับสูง (High level input voltage)

Vil : แรงดันไฟ input ระดับต่ำ (Low level input voltage)

Vcc : แรงดันไฟจากแหล่งจ่าย (Power supply voltage at VCC)

Vpp : แรงดันไฟที่ใช้โปรแกรม (Programming voltage at VPP)

Voh : แรงดันไฟ output ระดับสูง (High level output voltage)

Vol : แรงดันไฟ output ระดับต่ำ (Low level output voltage)

Tr : เวลาช่วงขาขึ้น (Rise time between 10% and 90% of signal amplitude)

Tf : เวลาช่วงขาลง (Fall time between 90% and 10% of signal amplitude)

Iih : กระแส input ระดับสูง (High level input current)

Iil : กระแส input ระดับต่ำ (Low level input current)

Icc : กระแสที่เกิดจากแหล่งจ่าย (Supply current at VCC)

Ipp : กระแสที่ใช้ในการโปรแกรม (Programming current at VPP)

Ioh : กระแส output ระดับสูง (High level output current)

Iol : กระแส input ระดับต่ำ (low level output current)

I/O

หน้าสัมผัสนี้สามารถใช้เป็น input หรือ output สำหรับแลกเปลี่ยนข้อมูล และสองสถานะที่เป็นไปได้สำหรับ I/O

- mark หรือ high state (state Z) ถ้าการ์ดและอุปกรณ์ interface อยู่ใน reception mode หรือถ้า (state Z)ถูกกำหนดโดยตัวที่ส่งข้อมูล

- space หรือ low state (state A) ถ้าเป็นสถานะนี้จะถูกกำหนดโดยตัวส่งข้อมูล

เมื่อสัญญาณของการ์ดและ I/O อยู่ในโหมด reception mode สัญญาณจะเป็น state Z และเมื่อสัญญาณของทั้งสองไม่เหมือนกันก็จะเป็น transmit mode โดยไม่สนใจสถานะของ logic ว่าเป็น 0 หรือ 1 และการ์ดและ interface จะเป็น transmit mode พร้อมกันไม่ได้

ตารางที่ 2.1 คุณสมบัติสัญญาณของ I/Oภายใต้การปฏิบัติงานปกติ

Symbol	Condition	Minimum	Maximum	Unit
(1) Vih	Iih max = +/- 500 μ A	2	VCC	V
	Iih max = +/- 500 μ A	0.7 VCC	VCC (3)	V
Vil	Iil max = 1 mA	0	0.8	V
(2) Voh	Iol max = +/- 100 μ A	2.4	VCC	V
	Iol max = +/- 20 μ A	3.8	VCC	V
Vol	Iol max = 1mA	0	0.4	V
tr,tf	Cin = 30pF Cout = 30 pF		1	us

VPP

ในการเขียนหรือลบข้อมูลต้องใช้แรงดันป้อนให้กับหน้าสัมผัสด้วย VPP มีสองสถานะคือ idle state และ active state ทั้งสองสถานะถูกกำหนดตามตาราง อุปกรณ์ interface จะเป็นตัวทำให้เกิด idle state เว้นแต่ว่า จะต้องการ active state

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 คุณสมบัติขณะสัญญาณของ VPP ภายใต้การปฏิบัติงานปกติ

Symbol	Conditions	Minimum	Maximum	Unit
Vpp	Idle State	$0.95 \cdot V_{cc}$	$1.05 \cdot V_{cc}$	V
Ipp	(programming non active)		20	mA
Vpp	Active state	0.975	$1.025 \cdot P$	V
Ipp	(programming the active)		I	mA

P และ I เป็นค่าที่กำหนดไว้ที่ card แต่ละใบ Rise time หรือ fall time : สูงสุด 200 μ s อัตราการเปลี่ยนแปลงของ VPP จะไม่เกิน 2V/ μ s ค่า Power สูงสุด $V_{pp} \cdot I_{pp}$ ต้องไม่เกิน 1.5 W เมื่อเฉลี่ยในคาบ 1s

CLK

ความถี่ในที่นี่เป็นความถี่ที่ถูกส่งมาจาก อุปกรณ์ interface แบบของ CLK จะเป็นแบบ fi คือความถี่เริ่มต้นที่จะให้เกิดสัญญาณ ตอรับเพื่อทำการ reset หรืออาจเป็น fs คือความถี่ที่ใช้ในการส่งข้อมูล Duty cycle ของการทำงานแบบ asynchronous จะต้องอยู่ระหว่าง 45% และ 55% ของคาบการทำงาน

ตารางที่ 2.3 คุณลักษณะสัญญาณของ CLK ภายใต้การปฏิบัติงานปกติ

Symbol	Condition	Minimum	Maximum	Unit
Vih	Iih max = +/- 200 μ A	2.4	VCC (2)	V
	Iih max = +/- 20 μ A	0.7*VCC	VCC (2)	V
	Iih max = +/- 10 μ A	VCC-0.7	VCC(2)	V
Vil	Iil max = +/- 200 μ A	0 (2)	0.5	V
tr,tf	Cin = 30 pF		9% ของคาบโดยที่สูงสุด 0.5 μ s	

- (1). สำหรับ อุปกรณ์ interface ใช้ได้ทั้ง 3 สถานะ
- (2). แรงดันของ CLK จะอยู่ระหว่าง 0.3V และ VCC+0.3V

RST

ตารางที่ 2.4 คุณลักษณะสัญญาณของ RST ภายใต้การปฏิบัติงานปกติ

Symbol	Condition	Minimum	Maximum	Unit
(1) Vih	Iih max = +/- 200 μ A	4	VCC (2)	V
	Iih max = +/- 10 μ A	VCC - 0.7	VCC (2)	V
Vil	Iil max = +/- 200 μ A	0 (2)	0.6	V

- (1). สำหรับอุปกรณ์ interface ใช้ได้ทั้ง 2 สถานะ
- (2). แรงดันของ RST จะอยู่ระหว่าง 0.3V และ VCC+0.3VCC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5 คุณลักษณะสัญญาณของ VCC ภายใต้การปฏิบัติงานปกติ

Symbol	Minimum	Maximum	Unit
VCC	4.75	5.25	V
ICC		200	mA

ลักษณะการทำงานสำหรับ IC บนการ์ดนี้สามารถใช้ได้กับ IC การ์ดโดยการติดต่อ การติดต่อระหว่างอุปกรณ์ interface และการ์ด จะสามารถดำเนินการได้จากปฏิบัติการดังนี้

- การเชื่อมต่อและการกระตุ้น ของหน้าสัมผัสโดยอุปกรณ์ interface
- การรีเซตการ์ด
- ตอบรับการรีเซตโดยการ์ด
- การแลกเปลี่ยนข้อมูลระหว่างการ์ดและอุปกรณ์ interface
- การไม่กระตุ้นของการติดต่อโดยอุปกรณ์ interface

การปฏิบัติงานเหล่านี้จะระบุรายละเอียดต่อไป

หมายเหตุ สภาวะ active VPP จะไม่อยู่ที่ระดับเดิมเมื่อถูกรื้อขออกจากการ์ด การเชื่อมต่อและการกระตุ้น

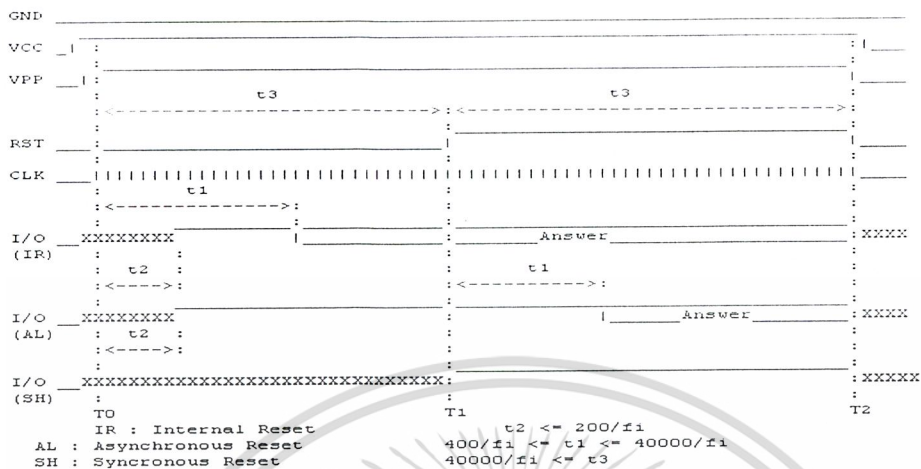
วงจรไฟฟ้าจะไม่ถูกกระตุ้นจนกว่าหน้าสัมผัสจะถูกเชื่อมต่อกับอุปกรณ์ interface เป็นการป้องกันความเสียหายที่จะเกิดขึ้นกับการ์ดและการ์ดต่างๆที่มีมาตรฐานนี้ การกระตุ้นหน้าสัมผัสของอุปกรณ์ interface จะประกอบด้วย

- RST อยู่ในสภาวะ L
- VCC เป็นแหล่งจ่าย
- I/O จะถูกอุปกรณ์ interface กำหนดให้อยู่ใน reception mode
- VPP จะเพิ่มขึ้นเป็น idle state
- CLK เป็นค่าใดๆที่เหมาะสมและเป็น CLK ที่คงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรีเซตการ์ด

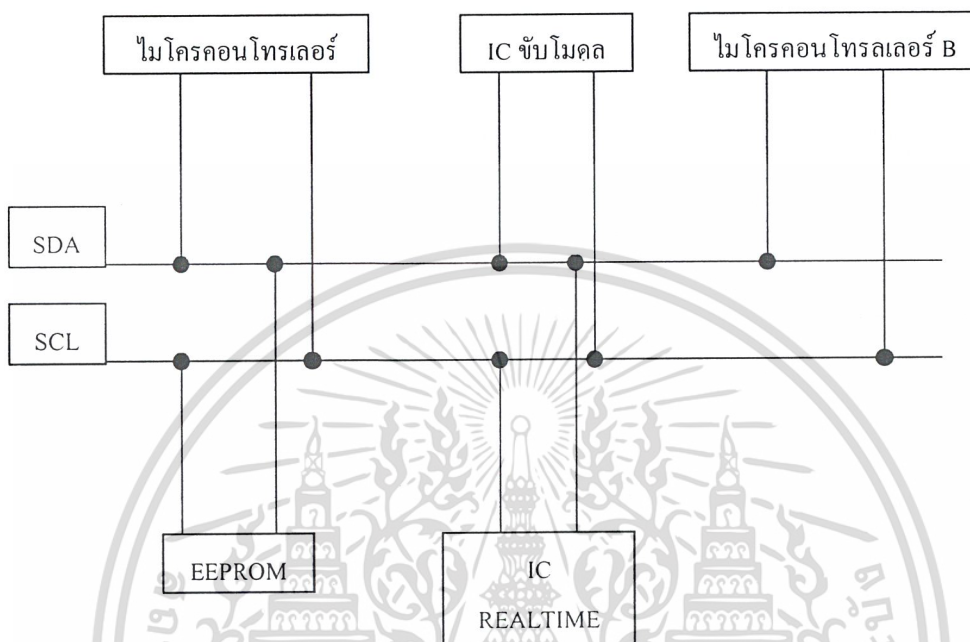
การรีเซตการ์ดจะเริ่มโดยอุปกรณ์ interface และการ์ดจะตอบสนองโดยการส่งสัญญาณตอบรับในการรีเซต โดยสุดท้ายของการกระตุ้นการ์ดของหน้าสัมผัส (RST เป็น Low ,VCCเป็นแหล่งจ่ายและคางที่ I/O อยู่ใน reception mode ,VPP คางที่ idle, CLK เป็นสัญญาณที่เหมาะสมและคางที่)การ์ดจะส่งสัญญาณตอบรับว่าพร้อมจะรีเซต เริ่มให้สัญญาณนาฬิกา CLK และขาสัญญาณ I/O จะถูกเซตให้เป็นสถานะ Z (high) ด้วยสัญญาณ CLK จำนวน 200 clock (t2) ในช่วงที่เริ่มมีสัญญาณ CLK ลูกที่สอง การ์ดจะรีเซตสถานะหลังจากมีสัญญาณ CLK สัญญาณในการรีเซตที่ I/O เริ่มระหว่างสัญญาณนาฬิกาที่ 40 และ 40000 (t1) หลังจากที่มีสัญญาณ CLK เป็น เวลาที่ t1 หลังจาก T0 การ์ดจะทำการ low reset คือการ reset โดยยังคงให้สัญญาณ RST อยู่ในสถานะ L ที่สัญญาณนาฬิกา น้อยกว่า 40000 clk (t3) หลังจากสัญญาณนาฬิกาเป็น t3 หลังจาก T0 ดังนั้นถ้าไม่มีสัญญาณตอบรับ ในการรีเซตจะเริ่มด้วย CLK จำนวน 40000 CLK (t3) ให้สัญญาณ RST เป็นสถานะ H (ที่เวลา T1)ทำให้มีสัญญาณตอบรับที่ I/O หลังจากขอขาขึ้นของสัญญาณที่ RST ถ้าสัญญาณตอบรับการรีเซตไม่เริ่มต้นที่ 40000 clock (t3) ขณะที่ RST อยู่ในสถานะ H (t3 หลังจาก T1) สัญญาณ RST จะเป็นสถานะ L อีกครั้งหนึ่ง (T2)



รูปที่ 2.6 ไคอะแกรมเวลาของสัญญาณการรีเซตการ์ด

2.2 ความรู้เบื้องต้นเกี่ยวกับระบบบัส I2C

I2C ย่อมาจาก Inter IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I2C ได้รับการพัฒนาขึ้นโดยฟิลลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ ทำงานและควบคุมภายใต้สายสัญญาณ 2 เส้น เส้นหนึ่งคือสายข้อมูล อีกเส้นเป็นสายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์ระบบบัส I2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานกัน ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัวจะใช้รหัสข้อมูลและการกำหนดสถานะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

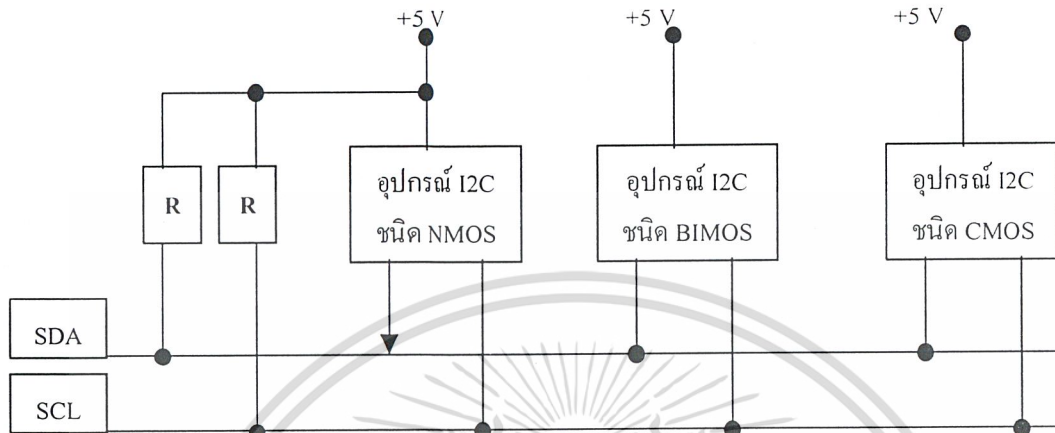


รูปที่ 2.7 แสดงการเชื่อมต่อของอุปกรณ์ต่างๆ บนระบบบัส I2C

สายข้อมูลบนบัส I2C มีชื่อเรียกว่า สายข้อมูลอนุกรม SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกา เรียกว่า สายสัญญาณนาฬิกาอนุกรม SCL (Serial Clock Line)

2.2.1 คุณสมบัติโดยทั่วไปของบัส I2C

สาย SDA และ SCL เป็นสายสัญญาณสองทิศทาง (bi-directional line) ต้องมีการต่อความต้านทานพูลอัพกับแรงดัน +5 V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูง ในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาท์พุทของอุปกรณ์ที่ต่ออยู่บนบัสจะต้องเป็นลักษณะแคทรินเปิด (open drain) หรือคอลเล็กเตอร์เปิด (open collector)



รูปที่ 2.8 แสดงการต่อความต้านทานพูลอัพบนสายสัญญาณ

อัตราการส่งผ่านข้อมูลบนบัส สูงถึง 100 Kbit/s ในโหมดปกติ(Standard mode) และสูงถึง 400 Kbit/s ในโหมดความเร็วสูง(Fast mode) อุปกรณ์ที่ต่อร่วมบนบัสจะต้องมีค่าความจุไฟฟ้าที่เกิดขึ้นระหว่างสาย SDA กับ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I2C ใช้ข้อมูลการเข้าถึง 2 รูปแบบ คือ 7 บิต (7 bit addressing) หรือ 10 บิต (10 bit addressing)

ข้อเด่นของบัส I2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ การต่อร่วมกันบนบัส I2C สามารถทำได้เช่นเดียวกับอุปกรณ์ที่มีไฟเลี้ยงเท่ากัน และต้องต่อความต้านทานพูลอัพเข้ากับแรงดัน +5 V เสมอ

2.2.2 หลักการของบัส I2C

บัส I2C ประกอบด้วยสายสัญญาณ 2 เส้น และอุปกรณ์ต่อพ่วงสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือโปรโตคอล(Protocol)เพื่อให้ผู้ใช้รู้ว่าขณะนี้ อุปกรณ์ใดติดต่ออยู่ และอุปกรณ์ใดเป็นตัวรับหรือส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์

- อุปกรณ์ที่เป็นตัวสร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง
- อุปกรณ์ที่ทำหน้าที่รับข้อมูล เรียกว่า ตัวรับ (receiver)
- อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I2C เรียกว่า มาสเตอร์ (master)
- อุปกรณ์ที่ถูกควบคุมหรือเป็นอุปกรณ์ที่ต่อพ่วงเข้ากับบัส I2C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการที่สำคัญในการติดต่อบนบัส I2C คือ

1. การส่งผ่านข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

2. ในระหว่างการส่งผ่านข้อมูลเมื่อใดก็ตามที่สาย SCL ต้องมีลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้ข้อมูลเกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

2.2.3 สภาวะที่เกิดขึ้นบนบัส I2C

1. บัสว่าง (Bus not busy) สภาวะนี้จะเกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL มีลอจิกเป็น 1 ทั้งคู่ ซึ่งหมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

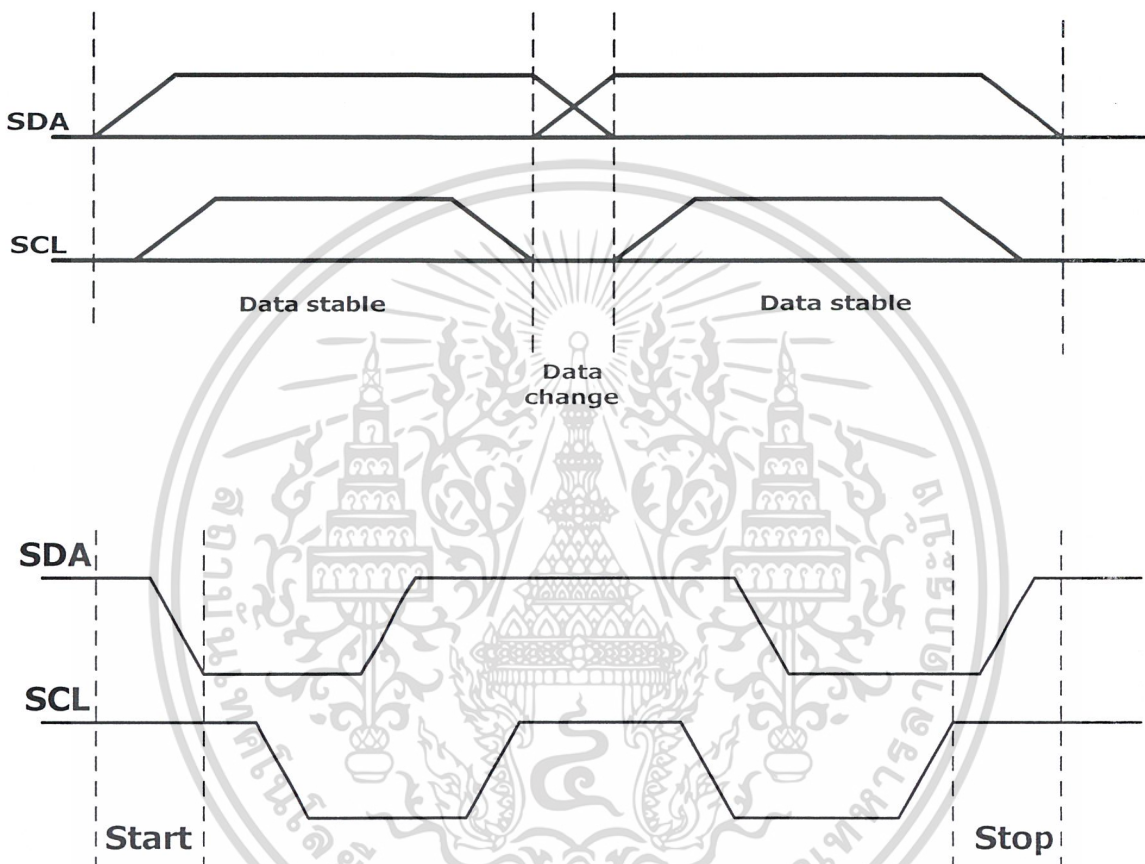
2. เริ่มต้นการส่งข้อมูล (Start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL ยังมีสถานะเป็นลอจิกสูง เรียกสภาวะนี้ว่า สภาวะเริ่มต้น

3. หยุดการส่งข้อมูล (Stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL ยังคงสถานะลอจิกสูงอยู่ เรียกว่า สภาวะหยุด (STOP)

4. ข้อมูลดำรงอยู่บนบัส (Data valid) สภาวะนี้เกิดถัดจากสภาวะเริ่มต้น โดยสถานะที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่ทำการส่ง เมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการส่งข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่ SCL มีลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่ อุปกรณ์มาสเตอร์ที่ทำการควบคุมการส่งข้อมูลจะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการส่งและรับเกิดความผิดพลาดขึ้น

5. รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากการส่งข้อมูลจากตัวส่งมายังตัวรับ เกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า บิตรับรู้ (acknowledge bit) มีสถานะเป็นลอจิกสูง หลังจากส่งข้อมูลครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะส่งสัญญาณรับรู้พิเศษซึ่งจะสัมพันธ์กันกับสัญญาณนาฬิกาเพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านรับจะส่งบิตรับรู้ที่มีสถานะลอ

จิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อหรือกำลังจะติดต่ออยู่ในขณะนั้นก็จะเกิดบิตรับรู้เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

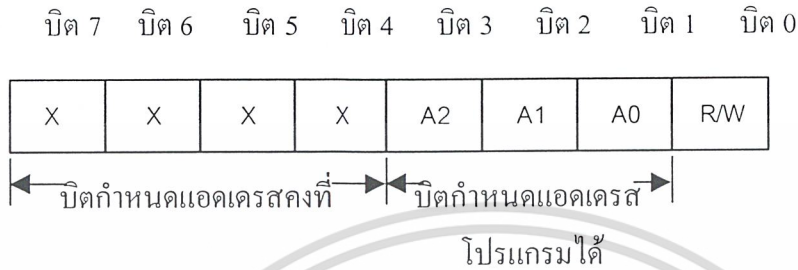


รูปที่ 2.9 แสดงไคอะแกรมเวลาของสถานะต่างๆ

2.2.4 การทำงานบนบัส I2C

ก่อนจะเริ่มดำเนินการอ่านเขียนข้อมูลระหว่างอุปกรณ์ต่างๆ ที่ต่ออยู่บนบัส จะต้องมีการอ้างถึงเสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I2C จะใช้การอ้างถึงแบบ 7 บิตและ 10 บิตในการต่ออุปกรณ์บนบัส ไม่มากใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้าอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ได้ติดต่อกับอุปกรณ์แต่ละตัวได้แล้ว ก็จะเริ่มต้นอ่านเขียนข้อมูล (R/W)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

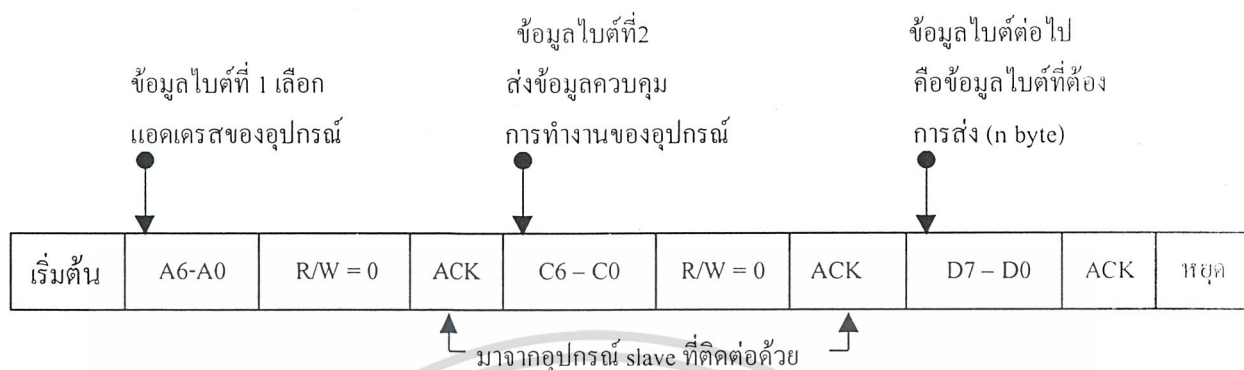


รูปที่ 2.10 แสดงรูปแบบการกำหนดแอดเดรสที่ใช้ในการอ้างแบบ 7 บิต

2.2.5 การอ้างถึงแบบ 7 บิต (7 – bit addressing)

ข้อมูลไบต์แรกที่เกิดขึ้นจากสถานะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ หรือ ข้อมูลกำหนดแอดเดรส โดยมีรูปแบบแสดงในรูปที่ 4 ใน 7 บิต

บรรทัดทั้ง MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ อุปกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0 – A2 ของอุปกรณ์ที่มีการเชื่อมบนบัสแบบ I2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟ ดังนั้นหากบิต LSB เป็น “0” หมายถึงการเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น “1” จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟข้อมูลในไบต์ต่อมา คือ ข้อมูลควบคุม (control byte) ในอุปกรณ์แต่ละตัว จะมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ข้อมูลไบต์ต่อมา คือ ข้อมูลที่ทำการส่งจริง (data) หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ออกกลับมาด้วยทุกครั้ง เพื่อให้กระบวนการส่งข้อมูลสามารถดำเนินต่อไปได้ ดังรูปที่ 2.8 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I2C ของการอ้างถึงแบบ 7 บิต



รูปที่ 2.11 รูปแบบของข้อมูลที่ใช้ติดต่อกับอุปกรณ์บัส I2C เมื่อใช้การอ้างแบบ 7 บิต

2.2.6 การอ้างถึงแบบ 10 บิต (10 bit addressing)

ในการอ้างถึงแบบ 10 บิต ยังคงใช้รูปแบบเหมือนกับการอ้างถึงแบบ 7 บิต เพียงแต่ต้องมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในข้อมูลไบต์แรกหลังจากเกิดสถานะเริ่มต้น ต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิต ถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อกับ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่าต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์ที่เลขที่ใดที่ต้องการติดต่อกับ ข้อมูลต่อมาเป็นแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการสื่อสารติดต่อกับ



รูปที่ 2.12 รูปแบบของข้อมูลที่ใช้ติดต่อกับอุปกรณ์บัส I2C เมื่อใช้การอ้างถึงแบบ 10 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 การเขียนโปรแกรมติดต่อบัส I2C

เริ่มต้นจากการสร้างสถานะมาตรฐานของบัส I2C ประกอบด้วย สถานะเริ่มต้น สถานะสิ้นสุดการส่งข้อมูล สถานะหยุด สัญญาณนาฬิกาบนขา SCL การเขียนข้อมูลบนระบบบัส I2C ดังแสดงในไฟล์ชาร์ต

2.2.7.1 การสร้างสถานะเริ่มต้น

1. เมื่อต้องการติดต่อบัส I2C สิ่งแรกที่ต้องทำสำหรับไมโครคอนโทรลเลอร์ที่ถือว่าเป็นอุปกรณ์มาสเตอร์คือ การทำให้เป็นบัสว่างด้วยการกำหนดให้ขา SCL และขา SDA มีลอจิกเป็น “1” ทั้งคู่

2. จากนั้นกำหนดให้ขา SDA มีลอจิกเป็น “0” โดยที่ขา SCL ยังคงเป็นลอจิก “1” อยู่

3. กำหนดให้ขา SCL มีลอจิกเป็น “0” ถึงตอนนี้ SCL และ SDA จะเป็น “0” ทั้งคู่

พร้อมที่จะติดต่อบัสได้แล้ว

2.2.7.2 การสร้างสถานะหยุด

1. เมื่อต้องการหยุดส่งข้อมูลจะต้องหยุดส่งสถานะหยุดออกไป โดยในตอนแรกต้องกำหนดขา SCL และ SDA เป็นลอจิก “0” ทั้งคู่ก่อน

2. กำหนดให้ขา SCL มีลอจิกเป็น “1” โดย SDA ยังคงเป็น “0”

3. จากนั้นทำให้ขา SDA มีลอจิกเป็น “1” ซึ่งจะทำให้ระบบบัสเข้าสู่บัสว่างอีกครั้ง พร้อมทั้งจะรับหรือส่งข้อมูลต่อไป

2.2.7.3 การส่งข้อมูลลอจิก “0” และลอจิก “1”

เป็นการส่งข้อมูลควบคุมหลังจากที่ส่งบิตเริ่มต้นไปแล้ว ซึ่งเป็นการส่งข้อมูลลอจิก “0” และลอจิก “1”

2.2.7.4 สำหรับการส่งข้อมูลลอจิก “0” มีขั้นตอนดังนี้

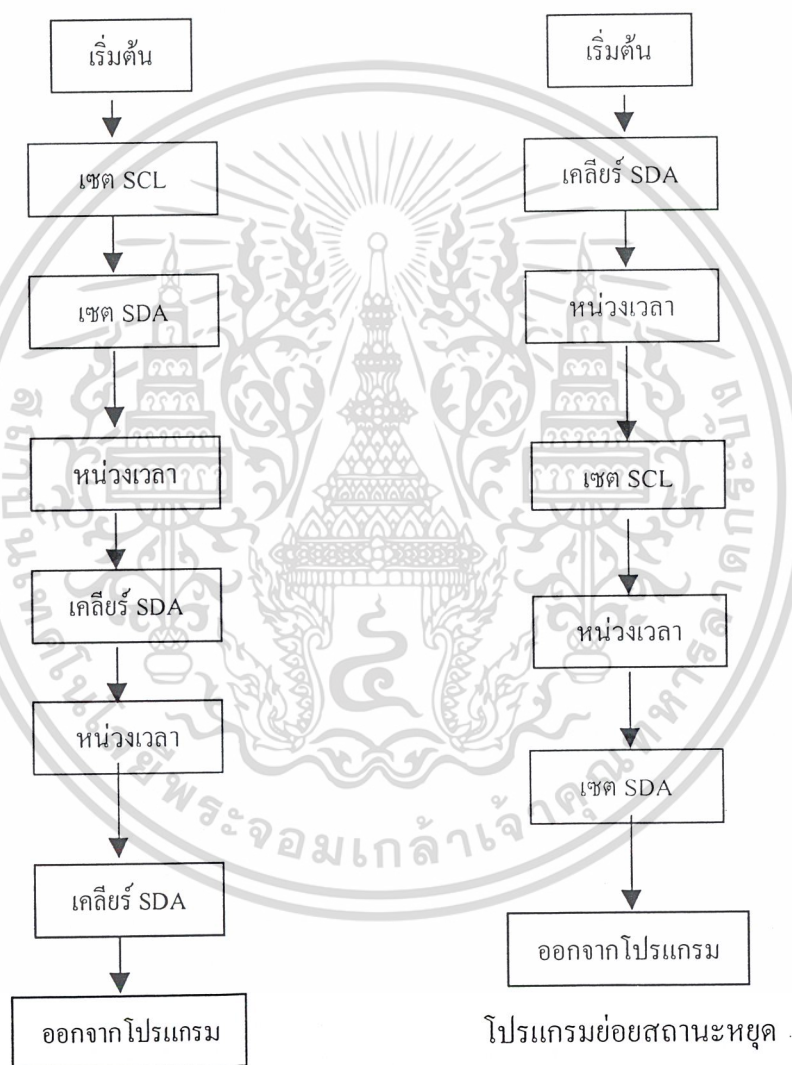
1. ทำให้ขา SDA เป็น “0” สำหรับการส่งข้อมูลลอจิก “0”

2. ทำให้ขา SCL เป็น “1” สำหรับการป้อนสัญญาณนาฬิกา ในขณะที่ขา SDA ยังเป็น “0” อยู่

4. จากนั้นทำให้ขา SCL กลับมาเป็น “0” เหมือนเดิม

2.2.7.5 สำหรับการส่งข้อมูลลอจิก “1” มีขั้นตอนดังนี้

1. ทำให้ขา SDA มีลอจิกเป็น “1” สำหรับการส่งข้อมูลลอจิก “1”
2. ทำให้ขา SCL เป็น “1” สำหรับการส่งสัญญาณนาฬิกา ในขณะที่ SDA ยังเป็น “1” อยู่ จากนั้นทำขา SCL กลับมามีสถานะเป็นลอจิก “0” เหมือนเดิม



โปรแกรมย่อยสถานะเริ่มต้น

รูปที่ 2.13 Flowchart ของ โปรแกรมย่อยของสถานะเริ่มต้นและสถานะหยุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

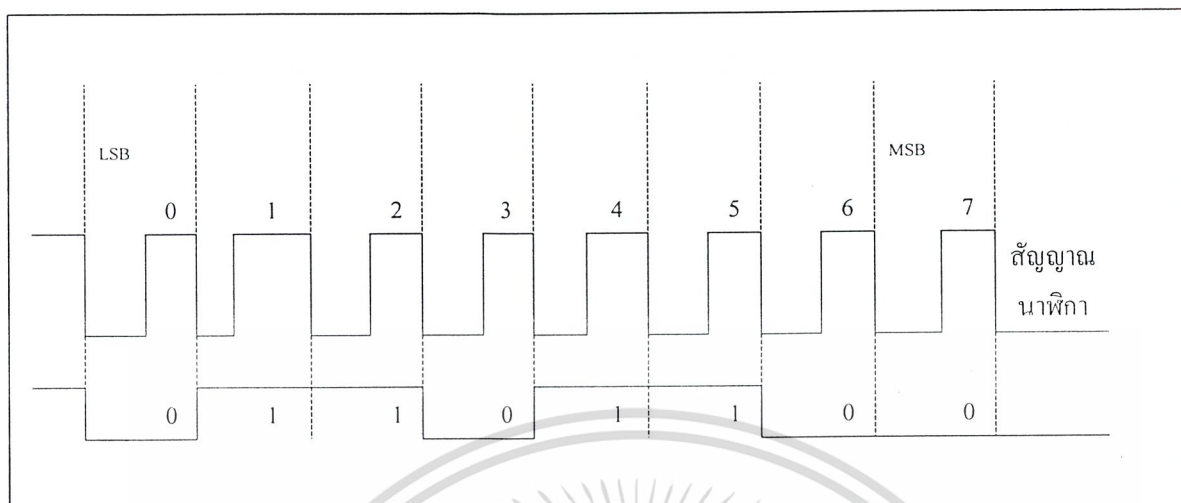
2.3 พอร์ตอนุกรม

มีทางเลือกอยู่ 2 แนวทางในการที่จะเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่นๆหรือคอมพิวเตอร์ด้วยกัน นั่นคือการรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรมการรับส่งข้อมูลแบบขนาน จะเป็นการรับหรือส่งข้อมูลคราวละ 4 หรือ 8 บิต ในเวลาเดียวกัน ซึ่งจะทำการรับและส่งข้อมูลทำได้ที่ความเร็วสูง ซึ่งก็หมายความว่าจำนวนของสายที่ใช้ในการส่งจะต้องมีมากเท่ากับจำนวนบิตของข้อมูลที่จะส่งด้วย นอกจากนี้ยังจะต้องรวมถึงสายที่ใช้สำหรับการควบคุมและการตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูล ที่จะส่งก็ได้ ซึ่งก็เป็นปัญหาในเรื่องราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักจะมีราคาแพง

ในขณะที่การรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลายๆบิต ได้ หากแต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่า จะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายที่น้อยกว่ามาก อย่างน้อยที่สุดใช้เพียง 2 – 3 เส้นเท่านั้น แต่อัตราเร็วในการรับส่งข้อมูลอาจต่ำกว่าแบบขนาน อย่างไรก็ตามการรับส่งข้อมูลแบบอนุกรมสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

2.3.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบ คือการสื่อสารอนุกรมแบบซิงโครนัส และการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา ,ข้อมูลและกราวด์ รูปที่ 2.11 แสดงให้เห็นถึงไทมิงไคอะแกรมของการส่งข้อมูลแบบซิงโครนัส



รูปที่ 2.14 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

2.3.1.1 การสื่อสารข้อมูลแบบอะซิงโครนัส

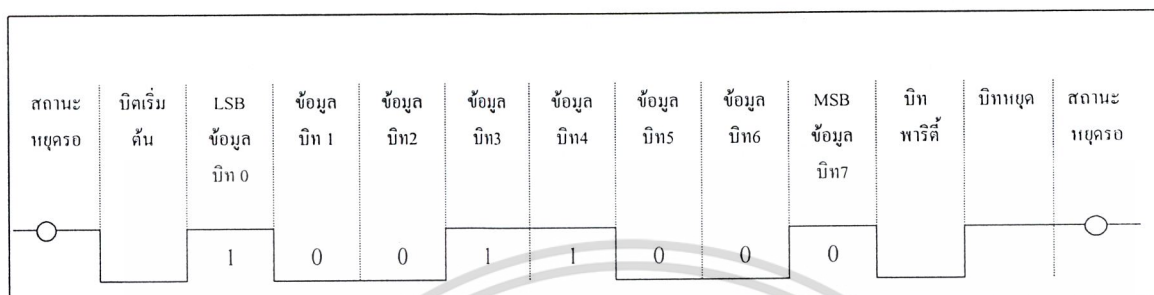
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกาพร้อมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ 9 บิต
3. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิต หรือ ไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

รูปที่ 2.15 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่งหา DATA จะมีสถานะ ลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้หา DATA มีลอจิก “0” ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่า บิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นจากตามด้วยบิตพาริตี ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิด

ท้าย ซึ่งจะให้ขนาดตัวมีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต ,1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว



รูปที่ 2.15 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า **Universal Asynchronous Receiver / Transmitter** หรือ UART อัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือ ค่าบอดเรต ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูลบอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232

ได้แก่ 110,150,300,600,1200,2400,4800,9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบิตเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบิตเรตคือจำนวนบิตข้อมูลที่สามารถถ่ายถอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บิตเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตีความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตีสามารถกำหนดในเป็นแบบคี่ (odd),แบบคู่ (even)หรือไม่มีการตรวจสอบพาริตีก็ได้ การตรวจสอบพาริตีเป็นการตรวจสอบจำนวนของบิตที่เป็นลอจิก "1" ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตีเข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก "1" จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตีเป็นคู่ค่าในบิตพาริตี จะต้องมีลอจิกเป็น "0" แต่ถ้าพาริตีเป็นคี่ ค่าบิตพาริตีจะต้องเป็น "1" เพื่อให้ข้อมูล 1 ไบต์ รวมทั้งบิตพาริตีมีจำนวน

บิตที่เป็นลอจิก “1” มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 4.1 แสดงตัวอย่างของบิตพาริตีในการรับส่งข้อมูลอนุกรม

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องการกำหนดคุณสมบัติการตรวจสอบพาริตีให้ตรงกันว่าจะตรวจสอบพาริตีคี่หรือพาริตีคู่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ทั้งภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้งานทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียว เท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับและภาคส่งจะไม่มีตรวจสอบพาริตี

ตารางที่ 2.6 แสดงบิตพาริตีของข้อมูล

ข้อมูล	บิตพาริตีคู่	บิตพาริตีคี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UARTเบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ UARTเบอร์ 8250 UART ชิพเหล่านี้มีระดับแรงดันแบบทีทีแอล (1 และ -5V) แต่เพื่อให้มีแรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ที่ระยะทางไกลมากขึ้น ระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิก “0” มีระดับแรงดัน +3V ถึง +12V ในขณะที่ลอจิก “1” มีระดับแรงดัน -3 V จนถึง -12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12V แสดงว่าเป็นช่องว่าง (Space)

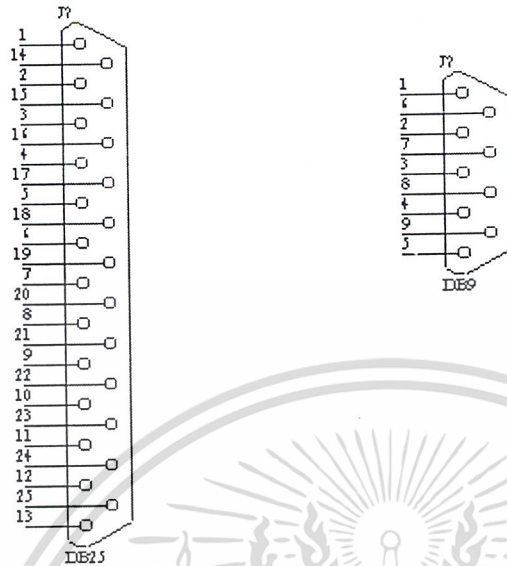
มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่ได้เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักจะถูกใช้เชื่อมต่อกับโมเด็มหรือเมาท์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

2.3.2.1 คอนเน็กเตอร์สำหรับพอร์ต Rs-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 2.9



- (1) คอนเน็กเตอร์อนุกรม 25 ขา หรือแบบ DB-25 (2) คอนเน็กเตอร์อนุกรม 9 ขา หรือแบบ DB-9
 (มองจากด้านหลังคอมพิวเตอร์) (มองจากด้านหลังคอมพิวเตอร์)

รูปที่ 2.16 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9 และ DB-25

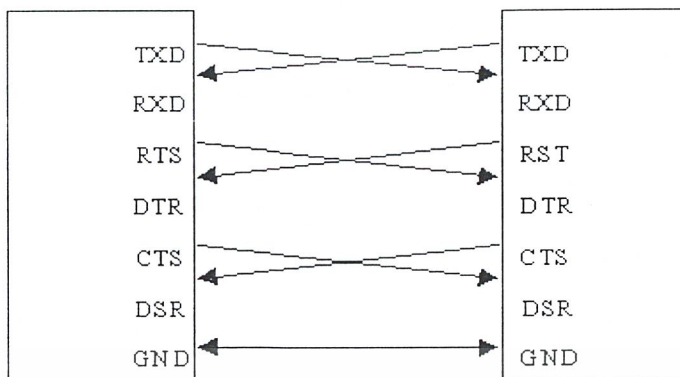
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 แสดงชื่อและทิศทางสัญญาณของ DB9 และ DB25

คอนเน็กเตอร์ DB9	คอนเน็กเตอร์ DB25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect:DCD	อินพุต
2	3	ReceicedData : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Teady : 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 Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังรูป 2.17 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 2.17(1) เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 2.17(2) เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

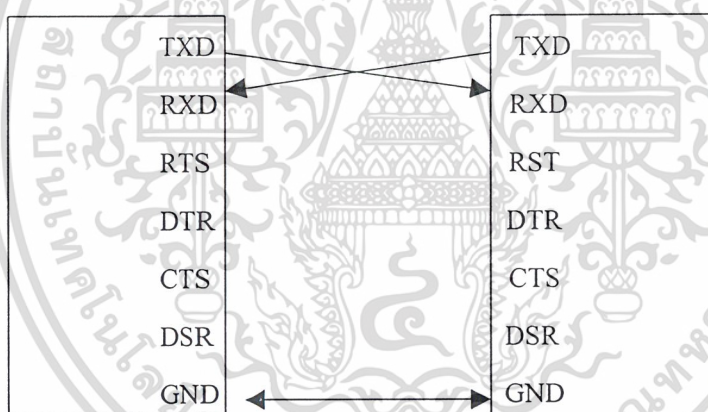
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



คอมพิวเตอร์

อุปกรณ์ภายนอก

(1) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null Modem



คอมพิวเตอร์

อุปกรณ์ภายนอก

(2) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS232 โดยใช้สัญญาณ 3 เส้น

รูปที่ 2.17 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในแบบต่างๆ

-**Data Carrier Detect : DCD** หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

-**Receive Data : RD** หรือ RxD ขานี้ใช้เพื่อรับส่งสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-**Transmitted Data : TD** หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

-**Data Terminal Ready : DTR** เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อขา DCDด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

-**Signal Ground : GND** ขากราวด์ของระบบ

-**Data Set Ready : DSR** ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSRนี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

-**Request To Send : RTS** เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

-**Clear To Send :CTS** ขานี้จะคอยรับสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสาร โดยทั่วไปสายนี้จะไม่ถูกใช้งานจะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.3.3 UART

UART มาจากคำว่า **Universal Asynchronous Receiver Transmitter** ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเองสำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขานานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขานานก่อนที่จะส่งข้อมูลเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต),รูปแบบการส่งข้อมูล,ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี,เฟรมข้อมูล,โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (programmable buadrate generator) โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1-65,535 UARTสามารถรับส่งข้อมูลได้ทั้งแบบ ฮาล์ฟดูเพล็กซ์ (half duplex) และฟูลดูเพล็กซ์ (full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

2.3.3.1 ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมาช้านาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ถือว่าเป็นต้นแบบ ของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกรุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนี้ยังเพิ่มส่วนของจ็พรีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UART เบอร์นี้หรือใหม่กว่า เช่น เบอร์ TL15C750 ซึ่งมีรีจิสเตอร์แบบ FIFO ขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5V และ +3V มีโหมดการประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MHz

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆ ก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาแบบในการแปลงข้อมูลเพียง 1.8432 MHz เท่านั้น

2.3.4 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม Rs-232

เครื่องคอมพิวเตอร์โดยทั่วไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COM 1, COM2, COM3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

การทำงานภายในพอร์ตอนุกรม ซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดเดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของ

พอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM 1 มีแอดแตรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆจะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมดังนี้

00H เป็นรีจิสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนابلการอินเตอร์รัปต์ ใช้ในการเซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดของอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจิสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจิสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจิสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD,RI,DSR และ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

รีจิสเตอร์ตำแหน่ง 00H : รีจิสเตอร์บัฟเฟอร์

เป็นรีจิสเตอร์สำหรับเก็บข้อมูลที่รับเข้ามาและข้อมูลที่จะส่งออกไป โดยการติดต่อกับรีจิสเตอร์นี้เพื่อเก็บข้อมูลที่ต้องการจะส่งจะต้องกำหนดให้บิต DLAB ในรีจิสเตอร์กำหนดรูปแบบข้อมูล (03H) จะต้องมีสถานะเป็น 0 ซึ่งการเขียนข้อมูลมายังแอดแตรสนี้ เป็นการส่งข้อมูลไปยังรีจิสเตอร์ส่งข้อมูล และข้อมูลจะถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อข้อมูลที่รับเข้ามาเรียบร้อยและแปลงเป็นแบบขนานแล้ว ข้อมูลจะถูกส่งมายังรีจิสเตอร์เก็บข้อมูล หลังจากมีการอ่านรีจิสเตอร์นี้ออกไปรีจิสเตอร์นี้จะถูกเคลียร์ และเตรียมพร้อมสำหรับการรับข้อมูลในไบต์ต่อไป

2.3.5 ลักษณะสัญญาณอินพุตและเอาต์พุตของพอร์ต RS-232

สัญญาณเอาต์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS,DSR และ DCD) ของพอร์ตอนุกรม RS-232จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232ก่อนส่งออกไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรม มี 4 ตำแหน่งดังนี้คือ

COM 1 : 3F8H

COM 2 : 2F8H

COM 3 : 3E8H

COM 4 : 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1จะเก็บไว้ที่แอดเดรส 0000 : 0400 H และ 0000 : 0401H ส่วนตำแหน่งอื่นๆมีรายละเอียดดังนี้

COM 2 = 0000 : 0402 H – 0000 : 0403 H

COM 3 = 0000 : 0404 H – 0000 : 0405 H

COM 4 = 0000 : 0406 H – 0000 : 0407 H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000 : 0411 H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใ้ช้อยู่ในคอมพิวเตอร์อีกด้วย โดยมีรายละเอียดดังแสดงในตารางที่ 2.8

ตารางที่ 2.8 แสดงข้อมูลในแอดเดรส 0000 : 0411 H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

บิต3	บิต2	บิต1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม

คอนโทรล MSComm

สำหรับการใช้งาน Visual BASIC ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual BASIC จะมี-ก๊สต่อมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเตอร์มาให้โดยใน Visual BASIC เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า MSCOMM.VBX ส่วนเวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ MACOMM32.OCX สำหรับการทำงานกับระบบปฏิบัติการ 32 บิตสำหรับใน Visual BASIC เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือการสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier (DCD) หรือขา Request To Send (RTS) เหตุการณ์ Oncomm ของ MSComm จะสามารถจับตรวจสอบสัญญาณนั้นได้ทันทีซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEven ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ ConnEven หลังจากให้โปรแกรมทำงานในฟังก์ชันต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต จะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอแดปเตอร์ของพอร์ตอนุกรมและแอแดปเตอร์ของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่อยู่ (COM1, COM2, COM3, COM4) รูปแบบการใช้งาน `object.CommPort [= value]` โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1) เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณ

แสดงข้อผิดพลาด error 68 ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด ,พาริตี,จำนวนบิตข้อมูล,จำนวนบิตปิดท้ายรูปแบบการใช้งาน `object.Setting [= value]` โดยค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น “BBBB,P,D,S” โดย BBBBเป็นค่าอัตราบอด , P เป็นค่าพาริตี ,D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น “9600,N,8,1”

ค่าบิตเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110	บิตต่อวินาที
300	บิตต่อวินาที
600	บิตต่อวินาที
1,200	บิตต่อวินาที
2,400	บิตต่อวินาที
9,600	บิตต่อวินาที (ค่าปกติ)
14,400	บิตต่อวินาที
19,200	บิตต่อวินาที
28,800	บิตต่อวินาที
28,400	บิตต่อวินาที
38,400	บิตต่อวินาที (สงวน)
56,000	บิตต่อวินาที (สงวน)
128,000	บิตต่อวินาที (สงวน)
256,000	บิตต่อวินาที (สงวน)

สำหรับค่ามาตรฐานในการกำหนด ค่าพาริตีมีดังนี้

สัญลักษณ์	รายละเอียด
E	พาริตีคู่ (Even)
M	ลอจิก “1” (Mark)
N	ไม่ใช้ (ค่าปกติ)
O	พาริตีคี่ (Odd)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S ลอจิก “0” (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4,5,6,7 และ 8 (เป็นค่าปกติ)ค่าที่ระบุจำนวนบิตปิดท้าย มี 3 ค่าคือ 1 (เป็นค่าปกติ) ,1,5และ 2 ตัวอย่างการใช้งานคำสั่ง Setting โดยจะเป็นการกำหนดค่าบิตเรตเท่ากับ 9600 ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

```
MSCONN1.Setting = “9600 , N , 8 , 1 “
```

หมายเหตุ สามาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด “” เนื่องจากค่าที่กำหนดนั้นอยู่ในรูปตัวแปร String

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม รูปแบบการใช้งาน `object.PortOpen [= value]` ค่า Value มีชนิดข้อมูลเป็นแบบบูลีน คือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ Falseหมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ตแต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้งานคำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บิตต่อวินาที ไม่มีพาริตี จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1.Setting = “9600,n,8,1 “
```

```
MSComm1.CommPort = 1
```

```
MSComm1.PortOpen = True
```

Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ รูปแบบการใช้งาน **object . Input**

คุณสมบัติ **InputLen** เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่าน โดยคุณสมบัติ **Input** การกำหนดค่าให้ **InputLen** เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ **Input** ทำให้การอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ **InputMode** เป็นตัวกำหนดชนิดของข้อมูลที่มีคุณสมบัติ **Input** รับเข้ามา ถ้า **InputMode** ถูกกำหนดเป็น **comInputModeText** คุณสมบัติ **Input** จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ **Variant** ถ้า **InputMode** กำหนดเป็น **comInputModeBinary** คุณสมบัติ **Input** จะส่งข้อมูลกลับมาในรูปแบบของไบนารีและชนิดข้อมูลเป็นแบบ **Variant**

ตัวอย่าง โปรแกรมแสดงให้เห็นถึงวิธีการรับข้อมูลจากบัฟเฟอร์ข้อมูลทั้งหมด

```
Private Sub Command1_Click ()
```

```
Dim InString as String
```

```
MSComm1.InputLen = 0           Retrieve all available data.
```

```
If MSComm1.InBufferCount Then Check for data
```

```
InString = MSComm1.Input       Read data
```

```
End If
```

```
End Sub
```

InbufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับรูปแบบการใช้งานคำสั่ง **object.InBufferCount [= value]** คำสั่ง **InBufferCount** จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไปสำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ **InBufferCount** มีค่าเป็น 0

หมายเหตุ อย่างสับสนระหว่าง คำสั่ง **InBufferSize** และ **InBufferCount** คำสั่ง **InBufferSize** นั้นใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์ รูปแบบการใช้งานคำสั่ง `object . InBufferSize [= value]` คำสั่ง `InBufferSize` ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆจะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือข้อมูลสั้นบัฟเฟอร์ เว้นแต่จะมีการใช้ `SendCheck` ดังนั้นค่าปานกลางที่เหมาะสมก็คือค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั่นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่าเพิ่มมากขึ้น

Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูลรูปแบบการใช้งาน `object.Output(value)` ค่า `value` เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ `Output` สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารีก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ `Variant` และมีข้อมูลภายในเป็นแบบ `byte`

OutBufferCount

คืนค่าจำนวนข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้อำนาจนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง `object.OutBufferCount [value]` ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาคส่งได้โดยกำหนดค่า `OutBufferCount` เท่ากับ "0"

หมายเหตุ ระวังการสับสนระหว่างคุณสมบัติ `OutBufferCount` กับ `OutBufferSuze` ซึ่ง `OutBufferSize` ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาคส่ง

OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง `object.OutBufferSize [= object]` คุณสมบัติ `OutBufferSize` ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไป จะทำให้มีหน่วยความจำเหลือให้ใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลล้นบัฟเฟอร์ขึ้นได้ ยกเว้นจะมีการใช้แฮนด์เช็ก วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512ไบต์ดูก่อน ถ้าโปรแกรมทำงานแล้วเกิดการล้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น.

เหตุการณ์ OnComm

เหตุการณ์ OnComm จะถูกสร้างขึ้นเมื่อค่าของคุณสมบัติ CommEvent มีการเปลี่ยนแปลงเพื่อแสดงผลการเปลี่ยนแปลงเหล่านั้นแบบทันทีทันใดหรือแสดงข้อผิดพลาดที่เกิดขึ้น.

การใช้ MSComm เพื่อการติดต่อฮาร์ดแวร์

จากรายละเอียดของ MSComm ที่กล่าวไปในตอนต้นนั้น จะเห็นได้ว่าวิธีการที่จะอ่านค่าหรือเขียนค่าไปยังขาสถานะและขาคควบคุมของพอร์ตอนุกรมสามารถทำได้ง่ายคามาก โดยใช้คำสั่งเหล่านี้

DTREnable	สำหรับคำสั่งให้ขา DTR มีลอจิก “0” หรือ “1”
RTSEnable	สำหรับสั่งให้ขา RTS มีลอจิก “0” หรือ “1”
CTSHolding	สำหรับอ่านค่าสถานะจากขา CTS ว่ามีลอจิก “0” หรือ “1”
CDHolding	สำหรับอ่านค่าสถานะจากขา DCDว่ามีลอจิก “0” หรือ “1”
DSRHolding	สำหรับอ่านค่าสถานะจากขา DSR ว่ามีลอจิก “0” หรือ “1”
Break	สำหรับการสั่งให้ขา Txd มีลอจิก “0” หรือ “1”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

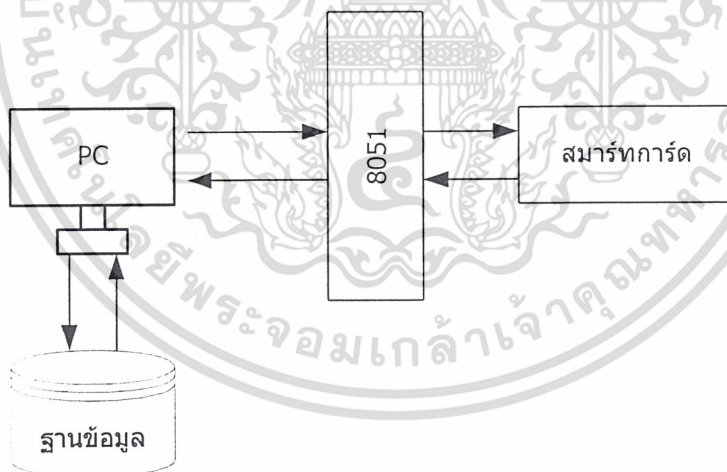
บทที่ 3

หลักการและวิธีการออกแบบ

จุดมุ่งหมายคือ ต้องการนำคุณสมบัติของสมาร์ทการ์ดมาประยุกต์ใช้โดยการให้สมาร์ทการ์ดเก็บข้อมูลที่เป็นรหัสนักศึกษาจะเหมือนเป็น ID CARD ส่วนโปรแกรม VB จะประกอบด้วยส่วนที่เป็นโปรแกรมรับข้อมูลที่อ่านมาจากสมาร์ทการ์ดโดยผ่านไมโครคอนโทรลเลอร์ 8051 และส่วนที่เป็นระบบฐานข้อมูลซึ่งเก็บรายชื่อและรหัส รวมทั้งการบันทึกเวลาด้วย

ในขั้นตอนของการออกแบบแบ่งเป็นส่วนต่างๆดังนี้

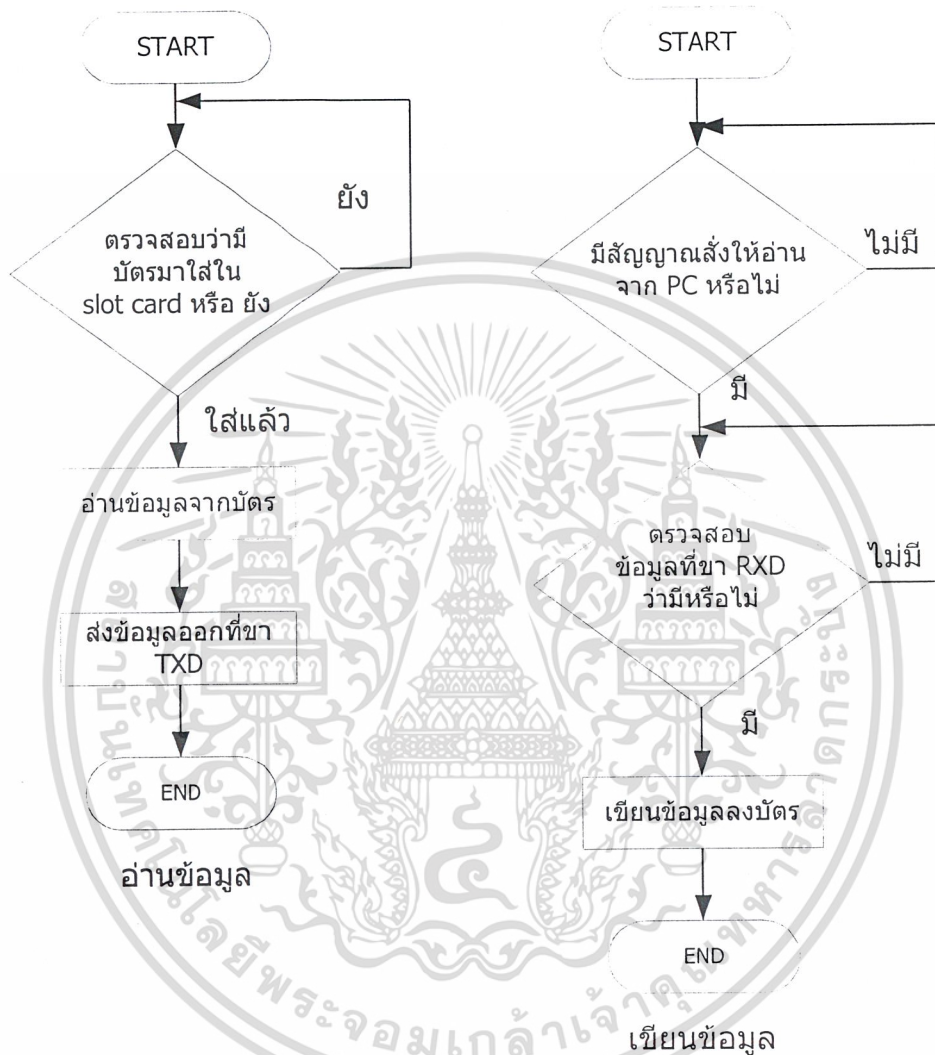
- ออกแบบโปรแกรมในส่วนไมโครคอนโทรลเลอร์ 8051 ซึ่งเป็นส่วนที่ใช้ในการอ่านเขียนสมาร์ทการ์ด
- ออกแบบโปรแกรมในส่วน Visual Basic ในส่วนนี้จะเป็นส่วนที่แสดงค่าที่อ่านออกมาได้ซึ่งเป็นรหัสนักศึกษาที่เขียนไว้หรือเป็นข้อมูลอื่นๆและนำข้อมูลที่ได้มาเปรียบเทียบกับฐานข้อมูลทำไว้



รูปที่ 3.1 รูปโดยรวมของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ออกแบบโปรแกรมส่วนของไมโครคอนโทรลเลอร์ 8051



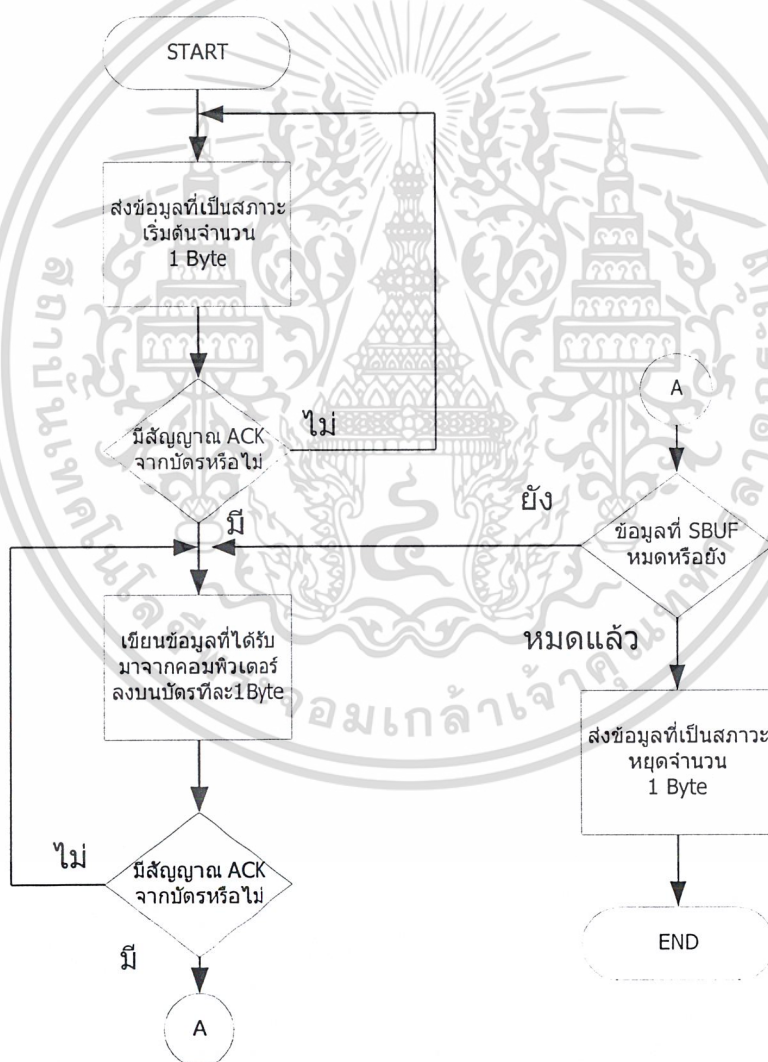
รูปที่ 3.2 flowchart ของโปรแกรมที่เขียนลงไมโครคอนโทรลเลอร์ 8051

จาก flowchart โปรแกรมจะทำการรับคำสั่งอ่าน-เขียนโดยใช้การอินเทอร์รัพของสัญญาณที่มาจากคอมพิวเตอร์ และสัญญาณที่มาจากสวิทช์ของ Slot card โดยในการอ่านหรือเขียนนั้นต้องทำตามโปรโตคอลของบัตรแต่ละชนิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 โปรแกรมย่อยในการเขียนข้อมูล

จากโปรโตคอลของบัตรที่นำมาใช้การเขียน โปรแกรมย่อยจะเริ่มการส่งข้อมูลจำนวน 1 Byte ซึ่งเป็นคำสั่งสถานะเริ่มต้นของบัตร และบิตที่เป็นคำสั่งใช้งานในโหมดการเขียน เมื่อส่งข้อมูลแล้วบัตรจะทำการส่งสัญญาณรับรู้ (ACK) กลับมาจากนั้นเราจะส่งข้อมูลที่เป็นข้อมูลที่ใช้งาน (ข้อมูลจริง) จะถูกเก็บไว้ที่ SBUF ของ 8051 และส่งออกมาทีละ 1 Byte ในการส่งข้อมูลลงบัตรจะเกิดสัญญาณรับรู้ (ACK) ทุกๆ 1 Byte จากบัตรเพื่อให้รู้ว่าได้รับข้อมูลที่ส่งไป เมื่อส่งข้อมูลจนครบแล้วจะต้องทำการส่งข้อมูล 1 Byte ซึ่งเป็นคำสั่งหยุดการติดต่อ

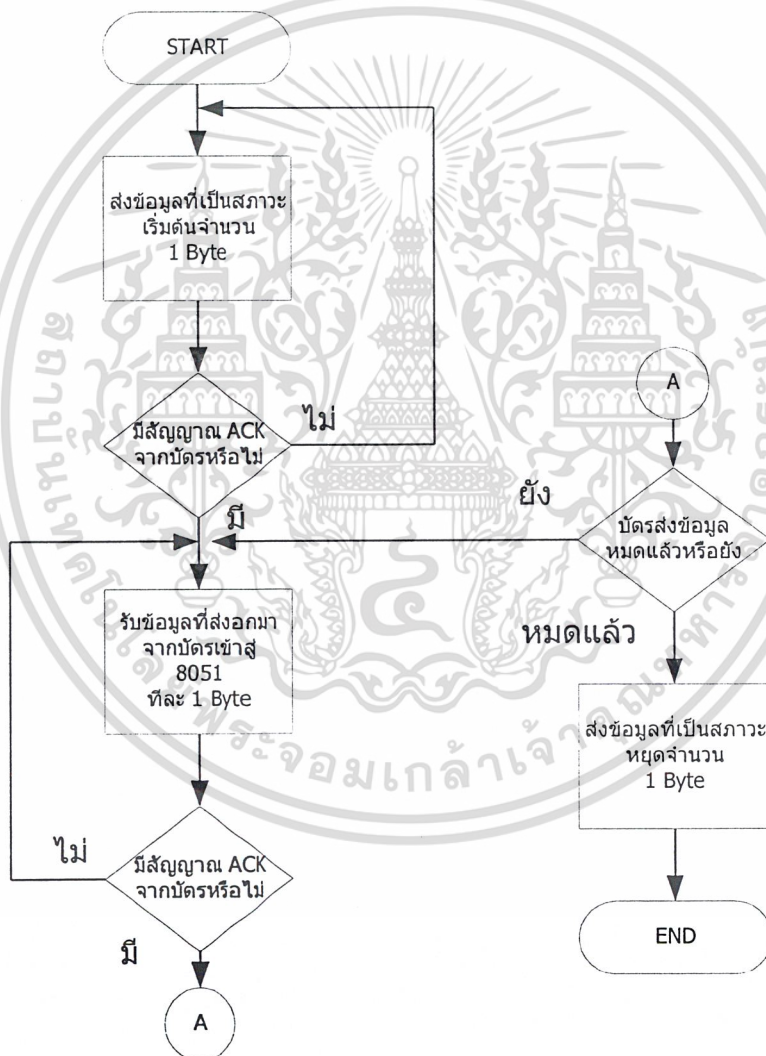


รูปที่ 3.3 แสดง flowchart ของ โปรแกรมย่อยในการเขียน โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 โปรแกรมย่อยในการอ่านข้อมูล

สำหรับในการอ่านข้อมูลไปทีละบิตจะคล้ายกับการเขียนคือ เริ่มที่ส่งข้อมูลที่เริ่มสถานะเริ่มต้น จำนวน 1 Byte ที่รวมคำสั่งสำหรับสั่งให้บิตทำงานในโหมดอ่านจากนั้นบิตจะส่งสัญญาณรับรู้เพื่อบอกให้รู้ว่าเป็นการตอบสนองข้อมูลและตามด้วยการส่งข้อมูลที่มีอยู่ในบิตออกมาทีละ 1 Byte และทุกๆ 1 Byte จะส่งสัญญาณรับรู้ออกมาด้วยเมื่อได้รับข้อมูลจากบิตแล้วต้องทำการส่งข้อมูล 1 Byte เพื่อเป็นคำสั่งสถานะหยุด



รูปที่ 3.4 แสดงโปรแกรมย่อยในการอ่านข้อมูล

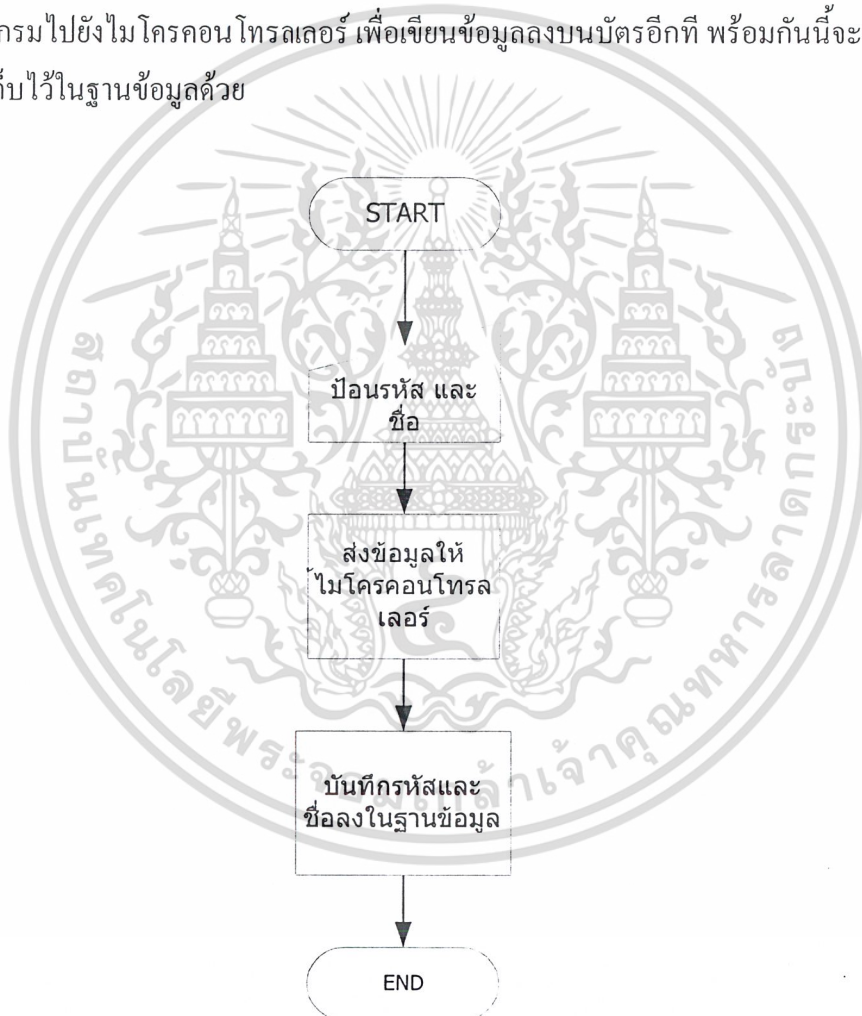
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ออกแบบโปรแกรมส่วน Visual Basic

โปรแกรมในส่วนนี้จะเป็นการนำข้อมูลที่รับเข้ามาประมวลผลกับฐานข้อมูล โดยการกระทำกับฐานข้อมูลนั้นมีทั้งการเปรียบเทียบข้อมูลและการเก็บข้อมูลโดยจะแบ่งเป็น 2 ส่วน

3.2.1 การใช้ VB ส่งข้อมูล

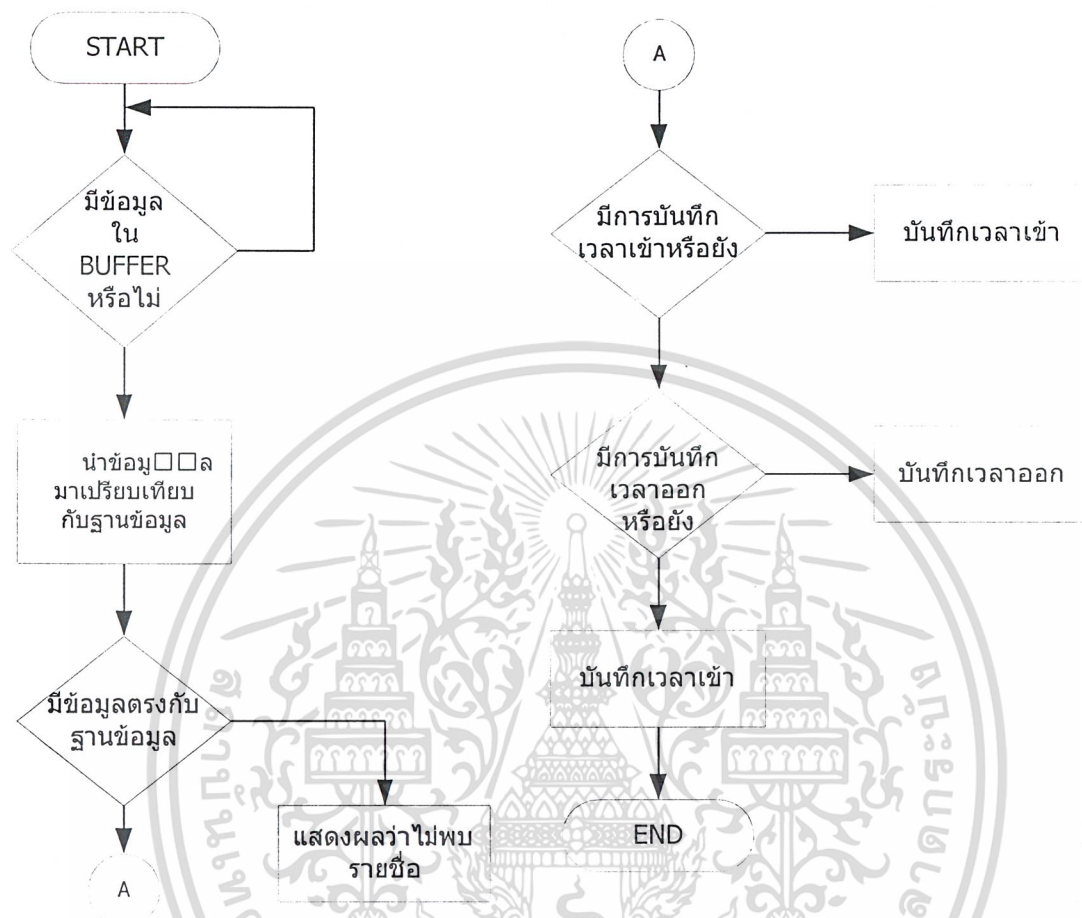
เริ่มต้นด้วยใส่รหัสนักศึกษาที่เป็นตัวเลขและใส่ชื่อใน Textbox หลังจากนั้นทำการส่งข้อมูลผ่านพอร์ตอนุกรมไปยังไมโครคอนโทรลเลอร์ เพื่อเขียนข้อมูลลงบนบัตรอีกที พร้อมกันนี้จะบันทึกรหัสและชื่อ เก็บไว้ในฐานข้อมูลด้วย



รูปที่ 3.5 flowchart การส่งข้อมูลให้ไมโครคอนโทรลเลอร์ และบันทึกรหัส,ชื่อ ลงฐานข้อมูล

3.2.2 การใช้ VB รับข้อมูล

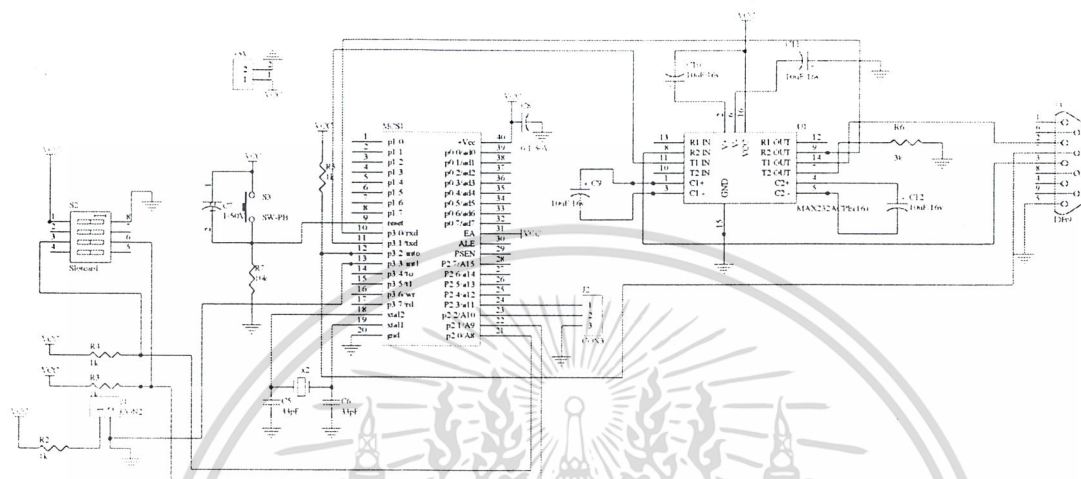
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 flowchart โปรแกรมย่อยในการรับค่าที่ส่งมาจากไมโครคอนโทรลเลอร์

เริ่มแรกนั้นต้องให้โปรแกรมวนลูปรอรับข้อมูลโดยเช็คข้อมูลที่เข้ามายังบัฟเฟอร์ เมื่อมีข้อมูลเข้ามาแล้วนำข้อมูลที่ได้ออกไปกระทำกับฐานข้อมูลที่เรามีอยู่ตาม flowchart จะเห็นว่าถ้าข้อมูลที่เข้ามาซึ่งเป็นรหัสนักศึกษาไม่ตรงกับข้อมูลที่เก็บไว้ในฐานข้อมูลโปรแกรมก็ทำการฟ้อง Error ว่าไม่มีข้อมูลและถ้าข้อมูลที่เข้ามาตรงกับฐานข้อมูลลำดับต่อไปจะเป็นการเช็คฐานข้อมูลเก็บเวลาเข้าแล้วหรือยัง ถ้ายังก็จะบันทึกเวลาเข้าไว้ในฐานข้อมูลและถ้ามีการบันทึกเวลาเข้าแล้วลำดับต่อไปก็คือเช็คว่ามีบันทึกเวลาออกหรือยัง

3.3 ออกแบบวงจรไมโครคอนโทรลเลอร์เพื่อติดต่อสมาร์ตการ์ด



รูปที่ 3.7 แสดงวงจรไมโครคอนโทรลเลอร์ที่ใช้ติดต่อกับสมาร์ตการ์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลลัพธ์ที่ได้

ในบทนี้จะเป็นผลที่ได้จากการรัน โปรแกรม VB โดยจะมีผลลัพธ์จากการเขียน,อ่านและรูปแบบของฐานข้อมูลที่ใช้ในโครงการนี้ดังจะแสดงเป็นขั้นตอนต่อไปนี้

1. ในครั้งแรกที่รัน โปรแกรม
2. เมื่อต้องการเขียนรหัสลงสู่บัตร
3. การรัน โปรแกรมเพื่อรอรับการอ่านค่าจากสมาร์ตการ์ด
4. แสดงรายละเอียดของวัน เวลา ที่มีอยู่ในฐานข้อมูล

4.1 ลักษณะโปรแกรมที่รันครั้งแรก

The screenshot shows a window titled 'Form1' with a data entry interface. It contains several text boxes for 'ID', 'Name', 'IN TIME', and 'OUT TIME'. To the right of these fields are buttons for 'Add new', 'Update', 'Delete', 'Start', 'Clear', and 'DEL TIME'. Below the text boxes are navigation buttons: '<', '>', '<|', and '>|'. At the bottom of the form, there are two buttons labeled '<Detail' and 'Detail'. The date and time '14/3/2545 19:14:33' are displayed at the bottom center of the window.

รูปที่ 4.1 แสดงรูปตอนรัน โปรแกรมครั้งแรก

จากรูปที่ 4.1 จะมี textbox และ ปุ่มต่างประกอบด้วย

- ปุ่ม Add new เป็นปุ่มที่ใช้เมื่อต้องการเขียนรหัสลงบัตรและเพิ่มข้อมูลในฐานข้อมูล ใช้คู่กับปุ่ม Update
- ปุ่ม Delete เป็นปุ่มที่ใช้ลบฐานข้อมูลที่ประกอบด้วย รหัสและชื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปุ่ม Start เป็นปุ่มที่ใช้เมื่อตอนรอรับการอ่านข้อมูลจากบัตรที่เข้ามาใส่ในSlot card
- ปุ่ม Clear เป็นปุ่มที่ใช้เคลียร์ค่าใน textbox
- ปุ่ม Del Time ใช้ลบข้อมูลเวลาในข้อมูลเวลา
- ปุ่มลูกศรต่างๆเป็นการเลื่อนฐานข้อมูลเพื่อดูรายละเอียด

4.2 การเพิ่มข้อมูลในฐานข้อมูลและเขียนรหัสลงสมาร์ทการ์ด

การเขียนรหัสและชื่อเก็บไว้ในฐานข้อมูลและเขียนลงบัตรจะสามารถทำได้โดยการกดปุ่ม Add new จากนั้นใส่รหัสจำนวน 8 ตัว และชื่อลงใน textbox แล้วกดปุ่ม update เพื่อเป็นการบันทึกข้อมูลลงในฐานข้อมูลและบัตร

The screenshot shows a window titled "Form1" with a data entry interface. It features two input fields labeled "ID" and "Name". To the right of these fields is a text prompt: "New record! Please Enter your Name.ID". Below the input fields are several navigation buttons: "<", "<|", ">|", and ">". To the right of these are buttons for "Add new", "Update", "Delete", "Start", "Clear", and "DEL TIME". At the bottom of the form, there are two buttons labeled "<Detail" and "Detail>". The date and time "14/3/2545 19:52:03" are displayed at the bottom center of the window.

รูปที่ 4.2 แสดงการเพิ่มข้อมูลให้ฐานข้อมูล และเขียนข้อมูลให้กับบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	NAME	ID
▶		

	NAMES	ENTRY_TIME	OUT_TIME
▶			No

รูปที่ 4.3 แสดงตารางฐานข้อมูลชื่อ,รหัสและเวลาเข้า-ออก
ที่ยังไม่มีข้อมูลใดๆ

รูปที่ 4.4 แสดงข้อมูลที่จะเพิ่มในฐานข้อมูล
และเมื่อเราเพิ่มข้อมูลเข้าไปในฐานข้อมูล ข้อมูลนั้นจะถูกเก็บในตารางฐานข้อมูลที่ทำไว้ใน
ตอนต้น

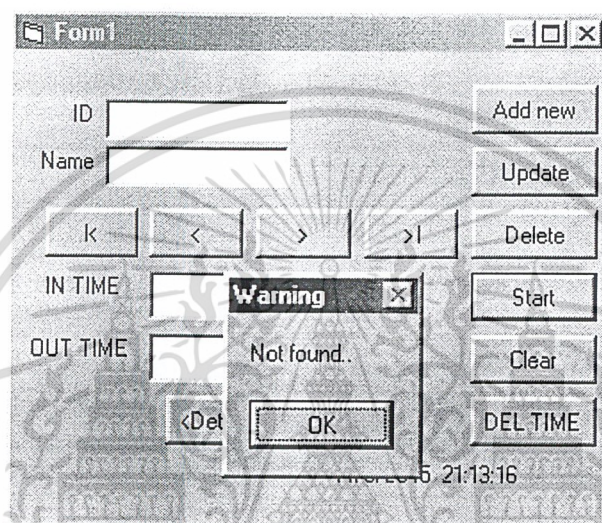
	NAME	ID
▶	niwat	43015821
*		

รูปที่ 4.5 แสดงข้อมูลที่เก็บในตารางฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การรันโปรแกรมเพื่อรับการอ่านค่าจากสมาร์ทการ์ด

ในการรันโปรแกรมเพื่อรับการอ่านค่านั้นจะทำได้จากการกดปุ่ม Start จากนั้น โปรแกรมจะรอรับค่าจนกว่าจะมีการใส่บัตรเข้ามาใน Slot card แล้วจะทำการประมวลผลโดยถ้าในการรับค่าแล้วไม่มีข้อมูลที่ตรงกับบัตรอยู่ในฐานข้อมูลจะแสดงผลดังนี้



รูปที่ 4.6 ภาพแสดงว่าไม่พบข้อมูล

และถ้ามีข้อมูลอยู่ก็จะบันทึกข้อมูลเวลาเข้าหรือเวลาออกไว้ในตารางข้อมูลเวลาดังจะแสดงในรูปต่อไปนี้

	NAMES	ENTRY_TIME	OUT_TIME
	niwat	14/3/2545 20:33:41	no
▶			No

	NAMES	ENTRY_TIME	OUT_TIME
	niwat	14/3/2545 20:33:41	14/3/2545 20:59:31
▶			No

รูปที่ 4.7 แสดงตารางฐานข้อมูลที่บันทึกเวลาเข้าและออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การแสดงรายละเอียดฐานข้อมูลวันเวลา

การแสดงรายละเอียดฐานข้อมูลของวันเวลาสามารถทำได้โดยกดปุ่มลูกศร <Detail , Detail > โดยข้อมูลจะแสดงที่ textbox

The screenshot shows a window titled "Form1" with the following elements:

- ID:** An empty text box.
- Name:** A text box containing "niwat".
- Navigation Buttons:** A row of buttons labeled "k", "<", ">", and ">|".
- IN TIME:** A text box containing "14/3/2545 20:33:41".
- OUT TIME:** A text box containing "14/3/2545 20:59:31".
- Action Buttons:** A column of buttons labeled "Add new", "Update", "Delete", "Start", "Clear", and "DEL TIME".
- Detail Button:** A button labeled "<Detail> Detail" is highlighted with a red box.
- Time Display:** The text "14/3/2545 21:28:16" is displayed at the bottom of the form.

รูปที่ 4.8 แสดงรายละเอียดของเวลาเข้าออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุป

สิ่งที่ทำในโครงการนี้คือวงจรที่ใช้เป็นอินเตอร์เฟซของการอ่านเขียนข้อมูลระหว่างบัตรสมาร์ทการ์ดกับคอมพิวเตอร์ และมีฐานข้อมูลเป็นการบันทึกค่าต่างๆที่แสดงให้เห็นว่าสามารถนำสมาร์ทการ์ดมาใช้ในการรักษาความปลอดภัยได้อย่างไรในระหว่างการทำโครงการนี้จะมีอุปสรรคในระหว่างการทำดำเนินงานบ้าง เช่น การเลือกบัตรสมาร์ทการ์ดที่จะนำมาใช้งานต้องดูด้วยว่ามีขายในประเทศหรือไม่ เพราะถึงแม้ว่าจะเป็นที่นิยมกันมากแล้วแต่ส่วนใหญ่ก็เป็นในต่างประเทศฉะนั้นการเลือกการ์ดจึงเป็นส่วนสำคัญในการพิจารณา

5.2 วิจารณ์ผลการทดลอง

โครงการนี้เป็นการนำสมาร์ทการ์ดมาใช้งานซึ่งจะเป็นตัวอย่างให้เห็นถึงการใช้งานสมาร์ทการ์ดด้านรักษาความปลอดภัย แต่ในโครงการนี้บัตรที่นำมาใช้งานเป็นแบบ EEPROM ซึ่งเป็นบัตรที่มีระบบป้องกันไม่สูงนัก ถ้าต้องการระบบป้องกันที่มีประสิทธิภาพอย่างเช่น SIM CARD ในโทรศัพท์ติดตามตัวอาจจะใช้สมาร์ทการ์ดแบบMicroprocessorCard ซึ่งจะเป็นแบบที่มีการประมวลผลด้วยตัวมันเอง

บรรณานุกรม

รองศาสตราจารย์ สมยศ จุณณะปิยะ, *การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51*,
แผนกตำราคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

กฤษดา ใจเย็น , อรรถพลบุญยะโกศา , ชัยวัฒน์ ลิ้มพรจิตรวิไล, *เรียนรู้และปฏิบัติการเชื่อมต่อ
คอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม* , บริษัทอินเวทีฟ เอ็กเพอริเมนต์
ธาริน สิทธิธรรมชารี, ธนันชัย จำนงค์ภักดี, *Microsoft Visual Basic Version 5.0 พิมพ์ครั้งที่ 2*,
บริษัท ซัคเซส มีเดีย จำกัด

กิตติ ภักดีวัฒนะกุล, จำลองครูอุตสาหะ, *Visual Basic6 ฉบับฐานข้อมูล*, KTPCOMP &
CONSULT, 2543

ISO 7816 PART 1-3 www.scai.org/about Smart Cards/iso7816_wimages.thm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Low-voltage and Standard-voltage Operation
 - 5.0 ($V_{CC} = 4.5V$ to 5.5V)
 - 2.7 ($V_{CC} = 2.7V$ to 5.5V)
- Internally Organized 128 x 8 (1K), 256 x 8 (2K), 512 x 8 (4K), 1024 x 8 (8K) or 2048 x 8 (16K)
- 2-wire Serial Interface
- Schmitt Trigger, Filtered Inputs for Noise Suppression
- Bi-directional Data Transfer Protocol
- 100 kHz (2.7V) and 400 kHz (5V) Compatibility
- 8-byte Page (1K, 2K), 16-byte Page (4K, 8K, 16K) Write Modes
- Partial Page Writes Are Allowed
- Self-timed Write Cycle (10 ms max)
- High-reliability
 - Endurance: 100,000 Write Cycles
 - Data Retention: 100 Years
 - ESD Protection: >3000V

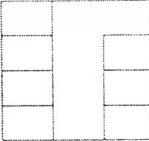
Description

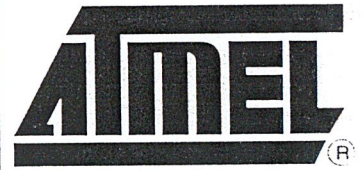
The AT24C01A/02SC/04SC/08SC/16SC provides 1024/2048/4096/8192/16384 bits of serial, electrically-erasable and programmable read-only memory (EEPROM) organized as 128/256/512/1024/2048 words of 8 bits each. The devices are optimized for use in smart card applications where low-power and low-voltage operation may be essential. The devices are available in several standard ISO 7816 smart card modules (see Ordering Information). The entire family is available in both high-voltage (4.5V to 5.5V) and low-voltage (2.7V to 5.5V) versions. All devices are functionally equivalent to Atmel serial EEPROM products offered in standard IC packages (PDIP, SOIC, EIAJ, LAP), with the exception of the slave address and Write Protect functions, which are not required for smart card applications.

Pin Configurations

Pad Name	Description	ISO Module Contact
VCC	Power Supply Voltage	C1
GND	Ground	C5
SCL	Serial Clock Input	C3
SDA	Serial Data Input/Output	C7
NC	No Connect	C2, C4, C6, C8

Card Module Contact

VCC = C1		C5 = GND
NC = C2		C6 = NC
SCL = C3		C7 = SDA
NC = C4		C8 = NC



2-wire Serial EEPROM Smart Card Modules

1K (128 x 8)

2K (256 x 8)

4K (512 x 8)

8K (1024 x 8)

16K (2048 x 8)

AT24C01ASC

AT24C02SC

AT24C04SC

AT24C08SC

AT24C16SC

Rev. 1610A-05/00



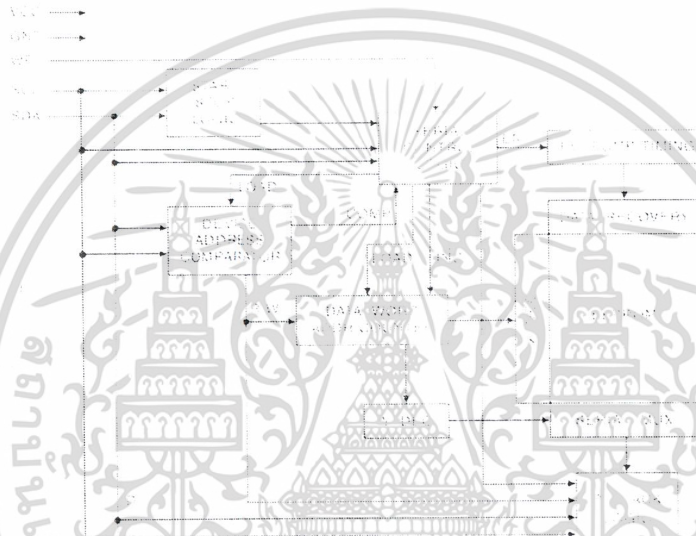
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Operating Temperature	-55 °C to +125 °C
Storage Temperature	-65 °C to +150 °C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.25V
DC Output Current	5.0 mA

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

Block Diagram



Pin Description

SERIAL CLOCK (SCL): The SCL input is used to positive edge clock data into each EEPROM device and negative edge clock data out of each device.

SERIAL DATA (SDA): The SDA pin is bi-directional for serial data transfer. This pin is open-drain driven and may be wire-ORed with any number of other open-drain or open-collector devices.

Memory Organization

AT24C01ASC, 1K SERIAL EEPROM: Internally organized with 16 pages of 8 bytes each, the 1K requires a 7-bit data word address for random word addressing.

AT24C02SC, 2K SERIAL EEPROM: Internally organized with 32 pages of 8 bytes each, the 2K requires an 8-bit data word address for random word addressing.

AT24C04SC, 4K SERIAL EEPROM: The 4K is internally organized with 32 pages of 16 bytes each. Random word addressing Chip Number requires a 9-bit data word address.

AT24C08SC, 8K SERIAL EEPROM: The 8K is internally organized with 64 pages of 16 bytes each. Random word addressing requires a 10-bit data word address.

AT24C16SC, 16K SERIAL EEPROM: The 16K is internally organized with 128 pages of 16 bytes each. Random word addressing requires an 11-bit data word address.

Pin Capacitance⁽¹⁾

Applicable over recommended operating range from $T_A = 25^\circ\text{C}$. $f = 1.0\text{ MHz}$. $V_{CC} = +2.7\text{V}$.

Symbol	Test Condition	Max	Units	Conditions
C_{in}	Input/Output Capacitance (SDA)	8	pF	$V_{CC} = 0\text{V}$
C_{in}	Input Capacitance (SCL)	6	pF	$V_{CC} = 0\text{V}$

Note: 1. This parameter is characterized and is not 100% tested.

DC Characteristics

Applicable over recommended operating range from: $T_{AC} = 0^\circ\text{C}$ to $+70^\circ\text{C}$. $V_{CC} = +2.7\text{V}$ to $+5.5\text{V}$ (unless otherwise noted).

Symbol	Parameter	Test Condition	Min	Typ	Max	Units
V_{DD}	Supply Voltage		2.7		5.5	V
V_{OL}	Supply Voltage		4.5		5.5	V
I_{CC}	Supply Current $V_{CC} = 5.0\text{V}$	READ at 100 kHz		0.4	1.0	mA
I_{CC}	Supply Current $V_{CC} = 5.0\text{V}$	WRITE at 100 kHz		2.0	3.0	mA
I_{SDA}	Standby Current $V_{CC} = 2.7\text{V}$	$V_{IN} = V_{CC}$ or GND		1.6	4.0	μA
I_{SCL}	Standby Current $V_{CC} = 5.0\text{V}$	$V_{IN} = V_{CC}$ or GND		8.0	18.0	μA
I_{IL}	Input Leakage Current	$V_{IN} = V_{CC}$ or GND		0.10	3.0	μA
I_{OL}	Output Leakage Current	$V_{OUT} = V_{CC}$ or GND		0.05	3.0	μA
V_{IL}	Input Low Level ⁽¹⁾		-0.6		$V_{CC} \times 0.3$	V
V_{IH}	Input High Level ⁽¹⁾		$V_{CC} \times 0.7$		$V_{CC} + 0.5$	V
V_{OL}	Output Low Level $V_{CC} = 3.0\text{V}$	$I_{OL} = 2.1\text{ mA}$			0.4	V

Note: 1. V_{IL} min and V_{IH} max are reference only and are not tested.

AC Characteristics

Applicable over recommended operating range from $T_A = 0^\circ\text{C}$ to $+70^\circ\text{C}$. $V_{CC} = +2.7\text{V}$ to $+5.5\text{V}$. $CL = 1\text{ TTL Gate}$ and 100 pF (unless otherwise noted).

Symbol	Parameter	2.7-volt		5.0-volt		Units
		Min	Max	Min	Max	
f_{SCL}	Clock Frequency, SCL		100		400	kHz
t_{PLW}	Clock Pulse Width Low	4.7		1.2		μs
t_{PHW}	Clock Pulse Width High	4.0		0.6		μs
t_{NS}	Noise Suppression Time ⁽¹⁾		100		50	ns
t_{L2D}	Clock Low to Data Out Valid	0.1	4.5	0.1	0.9	μs
t_{D2F}	Time the bus must be free before a new transmission can start ⁽¹⁾	4.7		1.2		μs
t_{HSTA}	Start Hold Time	4.0		0.6		μs
t_{SSTA}	Start Setup Time	4.7		0.6		μs
t_{HDAT}	Data In Hold Time	0		0		μs
t_{SDAT}	Data In Setup Time	200		100		ns
t_{IR}	Inputs Rise Time ⁽¹⁾		1.0		0.3	μs





AC Characteristics (Continued)

Applicable over recommended operating range from $T_A = 0\text{ }^{\circ}\text{C}$ to $+70\text{ }^{\circ}\text{C}$. $V_{CC} = +2.7\text{V}$ to $+5.5\text{V}$. $CL = 1\text{ TTL Gate and } 100\text{ pF}$ (unless otherwise noted).

Symbol	Parameter	2.7-volt		5.0-volt		Units
		Min	Max	Min	Max	
t_{F}	Inputs Fall Time ⁽¹⁾		300		300	ns
$t_{SU,STO}$	Stop Setup Time	4.7		0.6		μs
t_{OH}	Data Out Hold Time	100		50		ns
t_{WC}	Write Cycle Time		10		10	ms
Endurance ⁽¹⁾	5.0V, 25 $^{\circ}\text{C}$, Page Mode		100K		100K	Write Cycles

Note: 1. This parameter is characterized and is not 100% tested.

Device Operation

CLOCK and DATA TRANSITIONS: The SDA pin is normally pulled high with an external device. Data on the SDA pin may change only during SCL low time periods (refer to Data Validity timing diagram). Data changes during SCL high periods will indicate a start or stop condition as defined below.

START CONDITION: A high-to-low transition of SDA with SCL high is a start condition that must precede any other command (refer to Start and Stop Definition timing diagram).

STOP CONDITION: A low-to-high transition of SDA with SCL high is a stop condition. After a read sequence, the STOP command will place the EEPROM in a standby power mode (refer to Start and Stop Definition timing diagram).

ACKNOWLEDGE: All addresses and data words are serially transmitted to and from the EEPROM in 8-bit words. Each word requires the receiver to acknowledge that it has received a valid command or data byte. During the transmission of commands from the host to the EEPROM, the EEPROM will send a zero to the host to

acknowledge that it has received a valid command byte. This occurs on the ninth clock cycle of the command byte. During read operations, the host will send a zero to the EEPROM to acknowledge that it has received a valid data byte and that it requests the next sequential data byte to be transmitted during the subsequent eight clock cycles. This occurs on the ninth clock cycle of the data byte. If the host does not transmit this acknowledge bit, the EEPROM will disable the Read operation and return to standby mode.

STANDBY MODE: The AT24C01ASC/02SC/04SC/08SC/16SC features a low-power standby mode that is enabled: (a) upon power-up and (b) after the receipt of the STOP bit and the completion of any internal operations.

MEMORY RESET: After an interruption in protocol, power loss or system reset, any 2-wire part can be reset by following these steps:

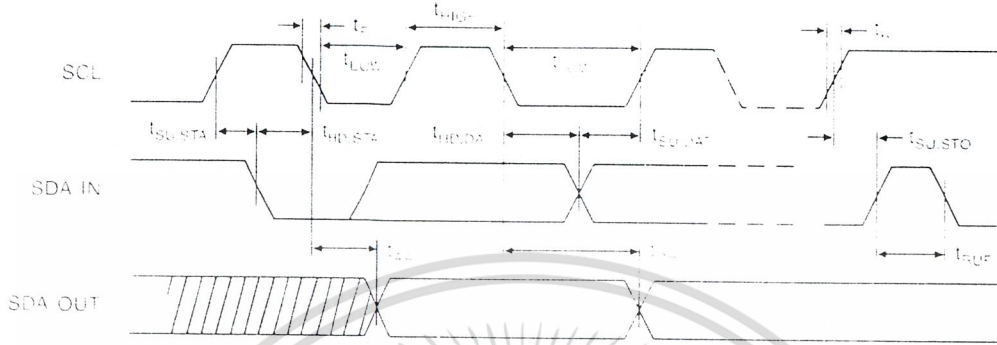
1. Clock up to 9 cycles.
2. Look for SDA high in each cycle while SCL is high.
3. Create a start condition as SDA is high.

AT24C01ASC/02SC/04SC/08SC/16SC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

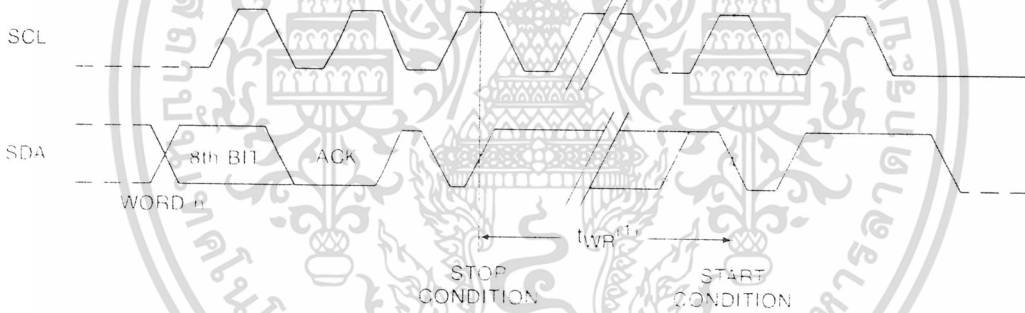
Bus Timing

SCL: Serial Clock, SDA: Serial Data I/O



Write Cycle Timing

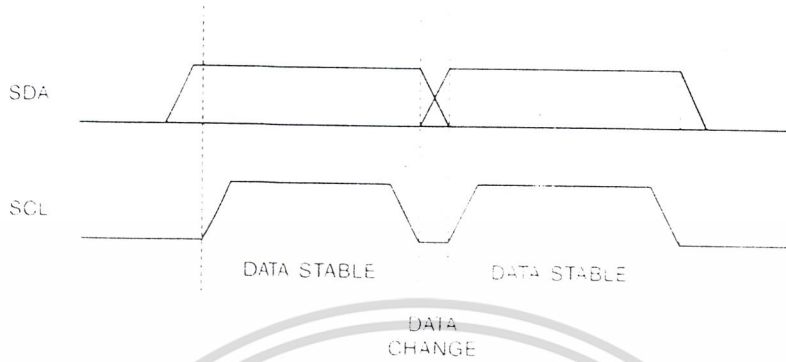
SCL: Serial Clock, SDA: Serial Data I/O



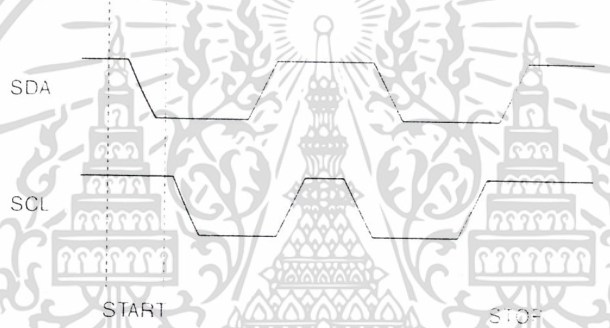
Note 1. The write cycle time t_{WR} is the time from a valid stop condition of a write sequence to the end of the internal clear/write cycle.



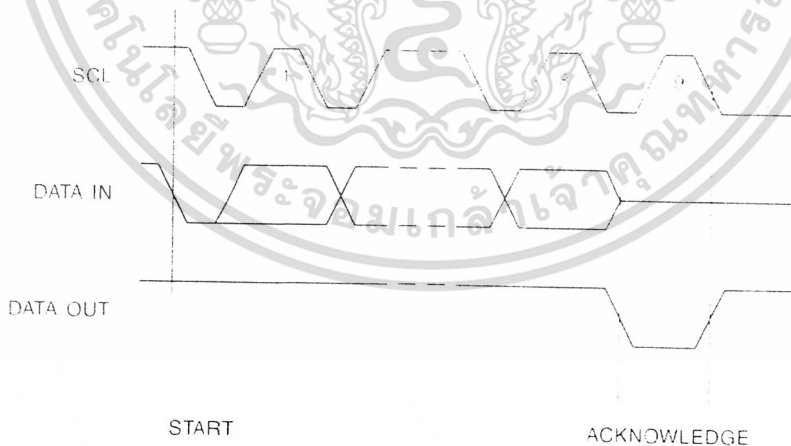
Data Validity



Start and Stop Definition



Output Acknowledge



Device Addressing

The 1K, 2K, 4K, 8K and 16K EEPROM devices all require an 8-bit device address word following a start condition to enable the chip for a read or write operation (refer to Figure 1).

The device address word consists of a mandatory one, zero, one, zero sequence for the first four most significant bits as shown. This is common to all the serial EEPROM devices.

The next 3 bits of the device address word are the most significant data word address bits for the AT24C16SC (16K), which requires a total of 11 address bits. The AT24C08SC (8K) requires only 10 total word address bits. The most significant 2 bits are included in the device address word. The unused bit of the device address word should be set to "0". The AT24C04SC (4K) requires only 9 total data word address bits. The most significant bit is included in the device address word. The 2 unused bits of the device address word should be set to "0". The AT24C02SC (2K) and AT24C01ASC (1K) do not require any address bits in the device address word. The 3 unused bits of the device address word should be set to "0".

The eighth bit of the device address is the read/write operation select bit. A read operation is initiated if this bit is high and a write operation is initiated if this bit is low.

Upon a compare of the device address, the EEPROM will output a zero (ACK). If a successful compare is not made, the chip will return to a standby state (NO ACK).

Write Operations

BYTE WRITE: A write operation requires an 8-bit data word address following the device address word and acknowledgment. Upon receipt of this address, the EEPROM will again respond with a zero (ACK) and then clock in the first 8-bit data word. Following receipt of the 8-bit data word, the EEPROM will output a zero (ACK) and the addressing device, such as a microcontroller, must terminate the write sequence with a stop condition. At this time the EEPROM enters an internally-timed write cycle, programming the nonvolatile memory. All inputs are disabled during this write cycle and the EEPROM will not respond until the write is complete (refer to Figure 2).

PAGE WRITE: The 1K/2K EEPROM is capable of an 8-byte page write, and the 4K, 8K and 16K devices are capable of 16-byte page writes.

A page write is initiated the same as a byte write, but the microcontroller does not send a stop condition after the first data word is clocked in. Instead, after the EEPROM acknowledges receipt of the first data word, the microcontroller can transmit up to 7 (1K/2K) or 15 (4K, 8K, 16K) more data words. The EEPROM will respond with a zero (ACK) after each data word received. The microcontroller

must terminate the page write sequence with a stop condition (refer to Figure 3).

The data word address lower 3 (1K/2K) or 4 (4K, 8K, 16K) bits are internally incremented following the receipt of each data word. The higher data word address bits are not incremented, retaining the memory page row location. When the word address, internally generated, reaches the page boundary, the following byte is placed at the beginning of the same page. If more than 8 (1K/2K) or 16 (4K, 8K, 16K) data words are transmitted to the EEPROM, the data word address will "roll over" and previous data will be overwritten.

ACKNOWLEDGE POLLING: Once the internally timed write cycle has started and the EEPROM inputs are disabled, acknowledge polling can be initiated. This involves sending a start condition followed by the device address word. The read/write bit is representative of the operation desired. Only if the internal write cycle has completed will the EEPROM respond with a zero (ACK), allowing the read or write sequence to continue.

Read Operations

Read operations are initiated the same way as write operations, with the exception that the read/write select bit in the device address word is set to one. There are three read operations: current address read, random address read and sequential read.

CURRENT ADDRESS READ: The internal data word address counter maintains the last address accessed during the last read or write operation, incremented by one. This address stays valid between operations as long as the chip power is maintained. The address "rollover" during read is from the last byte of the last memory page to the first byte of the first page. The address "rollover" during write is from the last byte of the current page to the first byte of the same page.

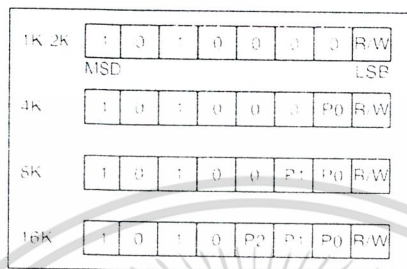
Once the device address with the read/write select bit set to "1" is clocked in and acknowledged by the EEPROM, the current address data word is serially clocked out. The microcontroller does not respond with an input zero, but does generate a following stop condition (refer to Figure 4).

RANDOM READ: A random read requires a "dummy" byte write sequence to load in the data word address. Once the device address word and data word address are clocked in and acknowledged by the EEPROM, the microcontroller must generate another start condition. The microcontroller now initiates a current address read by sending a device address with the read/write select bit high. The EEPROM acknowledges the device address and serially clocks out the data word. The microcontroller does not respond with a zero (NO ACK) but does generate a following stop condition (refer to Figure 5).

SEQUENTIAL READ: Sequential reads are initiated by either a current address read or a random address read. After the microcontroller receives a data word, it responds with an acknowledge. As long as the EEPROM receives an acknowledge, it will continue to increment the data word address and serially clock out sequential data words. When

the memory address limit is reached, the data word address will "rollover" and the sequential read will continue. The sequential read operation is terminated when the microcontroller does not respond with a zero (NO ACK) but does generate a following stop condition (refer to Figure 6).

Figure 1. Device Address



Note: P0, P1, P2 = Data word address bits

Figure 2. Byte Write



Figure 3. Page Write

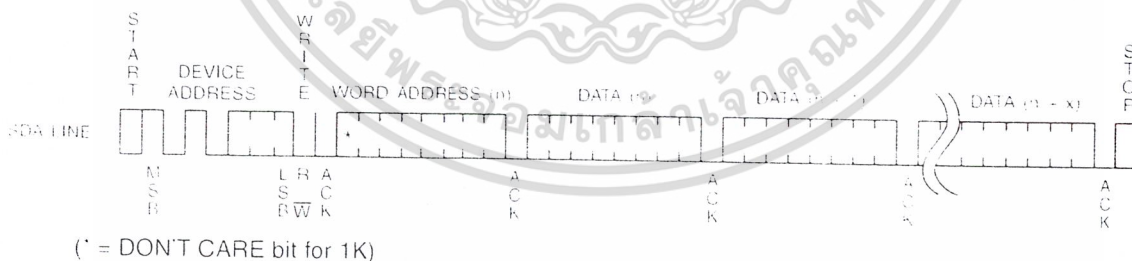


Figure 4. Current Address Read

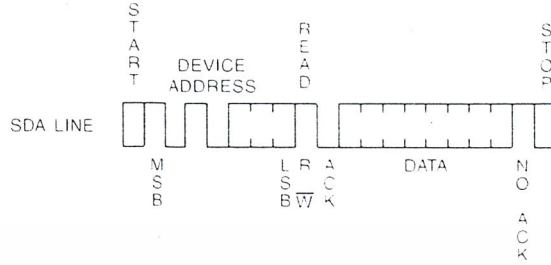


Figure 5. Random Read

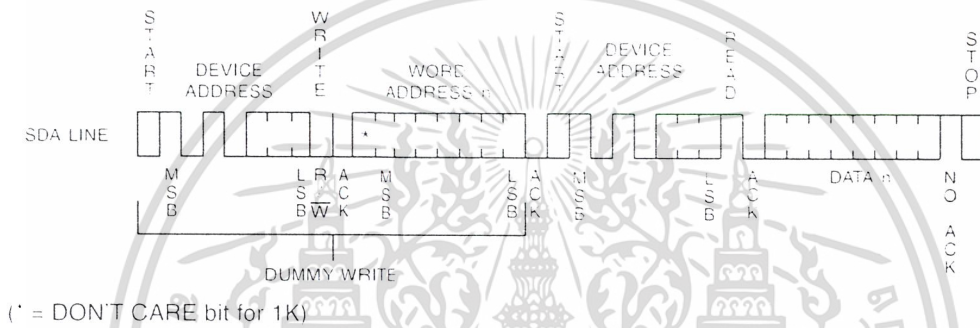
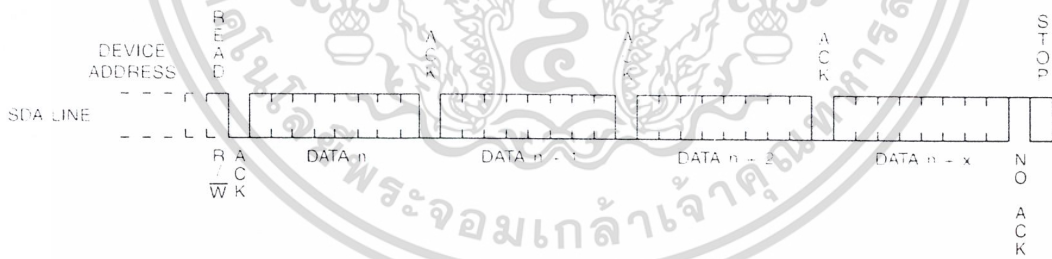


Figure 6. Sequential Read





AT24C01ASC Ordering Information

Ordering Code	Package	Voltage Range	Operation Range
AT24C01ASC - 09AT - 2.7	M2 - A Module	2.7V - 5.5V	Commercial (0 C to 70 C)
AT24C01ASC - 09BT - 2.7	M2 - B Module		
AT24C01ASC - 09CT - 2.7	M4 - C Module		
AT24C01ASC - 09DT - 2.7	M4 - D Module		
AT24C01ASC - 09ET - 2.7	M2 - E Module		
AT24C01ASC - 09FT - 2.7	M2 - F Module		
AT24C01ASC - 09GT - 2.7	M3 - G Module		
AT24C01ASC - 09HT - 2.7	M3 - H Module		
AT24C01ASC - 09AT	M2 - A Module	4.5V - 5.5V	Commercial (0 C to 70 C)
AT24C01ASC - 09BT	M2 - B Module		
AT24C01ASC - 09CT	M4 - C Module		
AT24C01ASC - 09DT	M4 - D Module		
AT24C01ASC - 09ET	M2 - E Module		
AT24C01ASC - 09FT	M2 - F Module		
AT24C01ASC - 09GT	M3 - G Module		
AT24C01ASC - 09HT	M3 - H Module		

Package Type ⁽¹⁾	
M2 - A Module	M2 ISO 7816 Smart Card Module
M2 - B Module	M2 ISO 7816 Smart Card Module with Atmel Logo
M4 - C Module	M4 ISO 7816 Smart Card Module
M4 - D Module	M4 ISO 7816 Smart Card Module with Atmel Logo
M2 - E Module	M2 ISO 7816 Smart Card Module
M2 - F Module	M2 ISO 7816 Smart Card Module
M3 - G Module	M3 ISO 7816 Smart Card Module
M3 - H Module	M3 ISO 7816 Smart Card Module with Atmel Logo

Note: 1. Formal drawings may be obtained from an Atmel sales office.

AT24C01ASC/02SC/04SC/08SC/16SC

AT24C02SC Ordering Information

Ordering Code	Package	Voltage Range	Operation Range
AT24C02SC - 09AT - 2.7	M2 - A Module	2.7V - 5.5V	Commercial (0 C to 70 C)
AT24C02SC - 09BT - 2.7	M2 - B Module		
AT24C02SC - 09CT - 2.7	M4 - C Module		
AT24C02SC - 09DT - 2.7	M4 - D Module		
AT24C02SC - 09ET - 2.7	M2 - E Module		
AT24C02SC - 09FT - 2.7	M2 - F Module		
AT24C02SC - 09AT	M2 - A Module	4.5V - 5.5V	Commercial (0 C to 70 C)
AT24C02SC - 09BT	M2 - B Module		
AT24C02SC - 09CT	M4 - C Module		
AT24C02SC - 09DT	M4 - D Module		
AT24C02SC - 09ET	M2 - E Module		
AT24C02SC - 09FT	M2 - F Module		

AT24C04SC Ordering Information

Ordering Code	Package	Voltage Range	Operation Range
AT24C04SC - 09AT - 2.7	M2 - A Module	2.7V - 5.5V	Commercial (0 C to 70 C)
AT24C04SC - 09BT - 2.7	M2 - B Module		
AT24C04SC - 09CT - 2.7	M4 - C Module		
AT24C04SC - 09DT - 2.7	M4 - D Module		
AT24C04SC - 09ET - 2.7	M2 - E Module		
AT24C04SC - 09FT - 2.7	M2 - F Module		
AT24C04SC - 09AT	M2 - A Module	4.5V - 5.5V	Commercial (0 C to 70 C)
AT24C04SC - 09BT	M2 - B Module		
AT24C04SC - 09CT	M4 - C Module		
AT24C04SC - 09DT	M4 - D Module		
AT24C04SC - 09ET	M2 - E Module		
AT24C04SC - 09FT	M2 - F Module		

Package Type⁽¹⁾

M2 - A Module	M2 ISO 7816 Smart Card Module
M2 - B Module	M2 ISO 7816 Smart Card Module with Atmel Logo
M4 - C Module	M4 ISO 7816 Smart Card Module
M4 - D Module	M4 ISO 7816 Smart Card Module with Atmel Logo
M2 - E Module	M2 ISO 7816 Smart Card Module
M2 - F Module	M2 ISO 7816 Smart Card Module
M3 - G Module	M3 ISO 7816 Smart Card Module
M3 - H Module	M3 ISO 7816 Smart Card Module with Atmel Logo

Note: 1. Formal drawings may be obtained from an Atmel Sales Office.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



AT24C08SC Ordering Information

Ordering Code	Package	Voltage Range	Operation Range
AT24C08SC - 09AT - 2.7	M2 - A Module	2.7V - 5.5V	Commercial (0 C to 70 C)
AT24C08SC - 09BT - 2.7	M2 - B Module		
AT24C08SC - 09CT - 2.7	M4 - C Module		
AT24C08SC - 09DT - 2.7	M4 - D Module		
AT24C08SC - 09ET - 2.7	M2 - E Module		
AT24C08SC - 09FT - 2.7	M2 - F Module		
AT24C08SC - 09AT	M2 - A Module	4.5V - 5.5V	Commercial (0 C to 70 C)
AT24C08SC - 09BT	M2 - B Module		
AT24C08SC - 09CT	M4 - C Module		
AT24C08SC - 09DT	M4 - D Module		
AT24C08SC - 09ET	M2 - E Module		
AT24C08SC - 09FT	M2 - F Module		

AT24C16SC Ordering Information

Ordering Code	Package	Voltage Range	Operation Range
AT24C16SC - 09AT - 2.7	M2 - A Module	2.7V - 5.5V	Commercial (0 C to 70 C)
AT24C16SC - 09BT - 2.7	M2 - B Module		
AT24C16SC - 09CT - 2.7	M4 - C Module		
AT24C16SC - 09DT - 2.7	M4 - D Module		
AT24C16SC - 09ET - 2.7	M2 - E Module		
AT24C16SC - 09FT - 2.7	M2 - F Module		
AT24C16SC - 09AT	M2 - A Module	4.5V - 5.5V	Commercial (0 C to 70 C)
AT24C16SC - 09BT	M2 - B Module		
AT24C16SC - 09CT	M4 - C Module		
AT24C16SC - 09DT	M4 - D Module		
AT24C16SC - 09ET	M2 - E Module		
AT24C16SC - 09FT	M2 - F Module		

Package Type⁽¹⁾

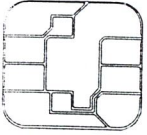
M2 - A Module	M2 ISO 7816 Smart Card Module
M2 - B Module	M2 ISO 7816 Smart Card Module with Atmel Logo
M4 - C Module	M4 ISO 7816 Smart Card Module
M4 - D Module	M4 ISO 7816 Smart Card Module with Atmel Logo
M2 - E Module	M2 ISO 7816 Smart Card Module
M2 - F Module	M2 ISO 7816 Smart Card Module
M3 - G Module	M3 ISO 7816 Smart Card Module
M3 - H Module	M3 ISO 7816 Smart Card Module with Atmel Logo

Note: 1. Formal drawings may be obtained from an Atmel Sales Office.

AT24C01ASC/02SC/04SC/08SC/16SC

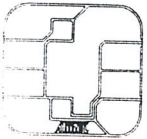
Smart Card Modules

M2 - A Module - Ordering Code: 09AT



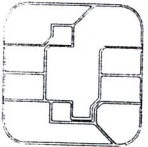
Module Size: M2
Dimension⁽¹⁾: 12.6 x 11.4 mm
Glob Top: Black. Square: 8.6 x 8.6 mm
Thickness: 0.58 mm max.
Pitch: 14.25 mm

M2 - B Module - Ordering Code: 09BT



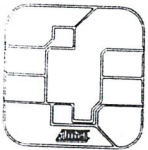
Module Size: M2
Dimension⁽¹⁾: 12.6 x 11.4 mm
Glob Top: Black. Square: 8.6 x 8.6 mm
Thickness: 0.58 mm max.
Pitch: 14.25 mm

M4 - C Module - Ordering Code: 09CT



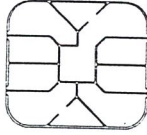
Module Size: M4
Dimension⁽¹⁾: 12.6 x 12.6 mm
Glob Top: Black. Square: 8.6 x 8.6 mm
Thickness: 0.58 mm
Pitch: 14.25 mm

M4 - D Module - Ordering Code: 09DT



Module Size: M4
Dimension⁽¹⁾: 12.6 x 12.6 mm
Glob Top: Black. Square: 8.6 x 8.6 mm
Thickness: 0.58 mm max.
Pitch: 14.25 mm

M2 - E Module - Ordering Code: 09ET



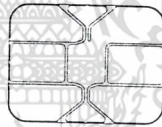
Module Size: M2
Dimension⁽¹⁾: 12.6 x 11.4 mm
Glob Top: Clear. Round: O 7.5 mm max.
Thickness: 0.58 mm max.
Pitch: 14.25 mm

M2 - F Module - Ordering Code: 09FT



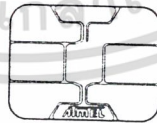
Module Size: M2
Dimension⁽¹⁾: 12.6 x 11.4 mm
Glob Top: Clear. Round: O 8.0 mm max.
Thickness: 0.58 mm max.
Pitch: 14.25 mm

M3 - G Module - Ordering Code: 09GT



Module Size: M3
Dimension⁽¹⁾: 10.6 x 8.0 mm
Glob Top: Clear. Round: O 6.5 mm max.
Thickness: 0.58 mm max.
Pitch: 9.5 mm

M3 - H Module - Ordering Code: 09HT



Module Size: M3
Dimension⁽¹⁾: 10.6 x 8.0 mm
Glob Top: Clear. Round: O 6.5 mm max.
Thickness: 0.58 mm max.
Pitch: 9.5 mm

Note: 1. The module dimensions listed refer to the dimensions of the exposed metal contact area. The actual dimensions of the module after excise or punching from the carrier tape are generally 0.4 mm greater in both directions (i.e. a punched M2 module will yield 13.0 x 11.8 mm).





Atmel Headquarters

Corporate Headquarters

2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia

Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan

Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs

1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset

Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand

North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail

literature@atmel.com

Web Site

<http://www.atmel.com>

BBS

1-(408) 436-4309

© Atmel Corporation 2000.

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing "®" and/or "™" are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

1610A-05/00/xM

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้