

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องลงคะแนนเสียงอิเล็กทรอนิกส์

ELECTRONIC VOTING



เลขหมู่.....
เลขทะเบียน.....
วัน,เดือน,ปี.....

72150

11 ส.ย. 2550

b. 11761089
i.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าหลักสูตรปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

ภาควิชา
วิศวกรรมโทรคมนาคม

ผ่านการตรวจชิ้นงานแล้ว
(ลงชื่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องลงคะแนนอิเล็กทรอนิกส์
ELECTRONIC VOTING

โดย

นางสาวจิมหรรษา กรังขยวิทยาการ 46012004

อาจารย์ที่ปรึกษา

ผศ. ธเนศ พัฒนธาดาพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาคณะศึกษาศาสตร์ปริญญาวิทยาศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

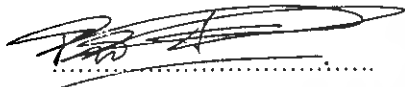
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องลงคะแนนอิเล็กทรอนิกส์

ELECTRONIC VOTING

ผู้จัดทำ

1. นางสาว จินห์วรา กรัณช์วิทย์ชากร 460120d4



อาจารย์ที่ปรึกษา

(ผศ.ธเนศ พัฒนธาดาทองษ์)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องลงคะแนนอิเล็กทรอนิกส์

ELECTRONIC VOTING

โดย ใ.น.ศ.จิณห์วรา กรณ์วิทยากร 460120004

อาจารย์ที่ปรึกษา ผศ.ชนนศ พัฒนธาคาพงษ์

บทคัดย่อ

โครงการนี้เป็นการนำเสนอเครื่องลงคะแนนอิเล็กทรอนิกส์ โดยอาศัยหลักการของไมโครคอนโทรลเลอร์และคอมพิวเตอร์ซอฟต์แวร์ในการดำเนินการ ซึ่งในส่วนของตัวชิ้นงานนั้นจะประกอบด้วย 2 ส่วน คือ ส่วนที่ใช้ในการเก็บค่าการลงคะแนนและส่วนที่ใช้ในการจัดเก็บข้อมูลการลงคะแนน และแสดงผลเมื่อสิ้นสุดการเลือกตั้ง

ABSTRACT

This project presents "Electronic Voting". This process used a theory of Microcontroller and software computer to do it. In the part of voting make up 2 parts, In the part of voting will receive point in the vote. This process used Computer for recorder and shows all points in the end of vote.

สารบัญ

| | หน้า |
|---|------|
| บทที่ 1 บทนำ | 1 |
| บทที่ 2 ทฤษฎีและหลักการ | 2 |
| 2.1 ไมโครคอนโทรลเลอร์ | 2 |
| 2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ | 2 |
| 2.1.2 การจัดขาของไมโครคอนโทรลเลอร์ | 2 |
| 2.1.3 การจัดการหน่วยความจำของไมโครคอนโทรลเลอร์ | 5 |
| 2.2 การเชื่อมต่อสวิตช์เข้ากับระบบไมโครคอนโทรลเลอร์ | 7 |
| 2.2.1 การเชื่อมต่อกับสวิตช์ | 7 |
| 2.2.2 การเชื่อมต่อกับ Keypad | 7 |
| 2.3 การเชื่อมต่อกับหน่วยแสดงผล | 8 |
| 2.3.1 การเชื่อมต่อกับไดโอดเปล่งแสง (LED) 7 ส่วน | 8 |
| 2.3.2 การเชื่อมต่อกับไดโอดเปล่งแสง (LED) 7 ส่วนหลายๆตัว | 9 |
| 2.4 ไมโครคอนโทรลเลอร์กับพอร์ตอินพุตและเอาต์พุต | 10 |
| 2.4.1 พอร์ตของ MCS-51 และการใช้งาน | 10 |
| 2.5 พอร์ตอนุกรมแบบ RS-232 | 11 |
| 2.5.1 มาตรฐานพอร์ตอนุกรมแบบ RS-232 | 11 |
| 2.5.2 คอนเนกเตอร์สำหรับพอร์ต RS-232 | 12 |
| 2.6 สมาร์ทการ์ด | 14 |
| 2.7 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล | 21 |
| 2.8 การเขียนโปรแกรมใน Visual C#. Net | 38 |
| บทที่ 3 การออกแบบและการทำงาน | 56 |
| 3.1 ขั้นตอนการทำงานของระบบ | 56 |
| 3.2 การออกแบบระบบ | 56 |
| 3.3 การออกแบบFlowchart | 67 |
| 3.4 การออกแบบฐานข้อมูล | 71 |
| 3.5 การรักษาความปลอดภัยเมื่อส่งข้อมูลผ่านเครือข่าย | 72 |
| 3.6 การออกแบบการส่งข้อมูล | 73 |
| บทที่ 4 ผลการทดลอง | 74 |
| บทที่ 5 สรุปผลการทดลอง | 93 |
| ภาคผนวก | |
| หนังสืออ้างอิง | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | | |
|-------------|--|----|
| รูปที่ 2.1 | การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 | 3 |
| รูปที่ 2.2 | แสดงการต่อขาของ MCS-51 ที่ใช้กับ XTAL | 5 |
| รูปที่ 2.3 | การจัดโครงสร้างของหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล | 6 |
| รูปที่ 2.4 | แสดงการสร้างลอจิกจากสวิตช์ | 7 |
| รูปที่ 2.5 | แสดงการสร้างสวิตช์เมตริกซ์ | 8 |
| รูปที่ 2.6 | แสดงสัญลักษณ์ของแต่ละ segment | 8 |
| รูปที่ 2.7 | แสดงการต่อมัลติเพล็กซ์ของ 7 segment | 10 |
| รูปที่ 2.8 | สัญญาณรูปแบบของข้อมูลอนุกรม | 11 |
| รูปที่ 2.9 | วงจรเชื่อมต่อ MAX หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ คอนเน็กเตอร์สำหรับพอร์ท RS-232 และการเชื่อมต่อ | 12 |
| รูปที่ 2.10 | คอนเน็กเตอร์อนุกรม 9 ขา หรือ DB-9 | 13 |
| รูปที่ 2.11 | แสดงรายละเอียดของการเชื่อมต่อพอร์ตอนุกรม RS-232 | 14 |
| รูปที่ 2.12 | ส่วนประกอบของสมาร์ตการ์ด | 16 |
| รูปที่ 2.13 | การแบ่งชนิดของบัตรตามรูปร่างที่นำไปใช้งาน และขนาด | 16 |
| รูปที่ 2.14 | ส่วนประกอบของสมาร์ตการ์ดโมดูลในการผลิตสมาร์ตการ์ด | 17 |
| รูปที่ 2.15 | Block-diagram โครงสร้างภายในชิปสมาร์ตการ์ดชนิด Memory | 19 |
| รูปที่ 2.16 | ตัวอย่างหน่วยความจำของ Memory card ชนิด PIN Protect | 19 |
| รูปที่ 2.17 | ตัวอย่างหน่วยความจำของสมาร์ตการ์ดชนิด Memory แบบ Token | 20 |
| รูปที่ 2.18 | บล็อกไดอะแกรมโครงสร้างภายในชิปสมาร์ตการ์ดชนิด Processor card (Asynchronous card) | 21 |
| รูปที่ 2.19 | ตัวอย่างข้อมูล | 25 |
| รูปที่ 2.20 | ตารางข้อมูล | 25 |
| รูปที่ 2.21 | ตัวอย่างใบบันทึกการซื้อสินค้า | 25 |
| รูปที่ 2.22 | ตัวอย่างข้อมูลที่ได้จากใบบันทึกการซื้อสินค้า | 26 |
| รูปที่ 2.23 | ตัวอย่างจากใบบันทึกการซื้อสินค้า | 26 |
| รูปที่ 2.24 | โครงสร้างฐานข้อมูลแบบลำดับขั้น | 27 |
| รูปที่ 2.25 | โครงสร้างฐานข้อมูลแบบเครือข่าย | 28 |
| รูปที่ 2.26 | โครงสร้างตารางฐานข้อมูลเชิงสัมพันธ์ | 29 |
| รูปที่ 2.27 | ตัวอย่างข้อมูลจากโครงสร้างตารางฐานข้อมูลเชิงสัมพันธ์ | 29 |
| รูปที่ 2.28 | ตัวอย่างการเพิ่มข้อมูลในตารางฐานข้อมูลเชิงสัมพันธ์ | 30 |
| รูปที่ 2.29 | แสดงระดับขั้นของข้อมูล | 34 |

| | |
|---|----|
| รูปที่ 2.30 รูปแบบการเก็บข้อมูลแบบ Multi dimensional array | 42 |
| รูปที่ 2.31 วนรอบแสดงข้อมูลในอาร์เรย์แต่ละแบบออกมา | 45 |
| รูปที่ 2.32 ข้อมูลจะถูกเรียงจากน้อยไปหามากอย่างรวดเร็ว | 46 |
| รูปที่ 2.33 ค้นหาว่า Me ที่ตำแหน่งที่ 1 | 47 |
| รูปที่ 3.1 แสดงขอบเขตการทำงานของระบบ | 56 |
| รูปที่ 3.2 แสดงรูปแผงหน้าปัดของเครื่องลงคะแนน | 57 |
| รูปที่ 3.3 แสดงบัตรที่ใส่ลงคะแนน | 58 |
| รูปที่ 3.4 โครงสร้างระบบเครือข่ายคอมพิวเตอร์สำหรับการเลือกตั้ง | 58 |
| รูปที่ 3.5 วงจรการทำงานของเครื่องลงคะแนน | 63 |
| รูปที่ 3.6 วงจรการทำงานของ 7- segment | 64 |
| รูปที่ 3.7 วงจรแสดงการทำงานของหัวอ่านบัตร smart card | 65 |
| รูปที่ 3.8 วงจรชุดการอ่านเขียนบัตร smart card | 66 |
| รูปที่ 3.9 แสดง Flowchart การทำงานของเครื่องลงคะแนน | 67 |
| รูปที่ 3.10 แสดง Flowchart การทำงานของเครื่องอ่านเขียนบัตร | 68 |
| รูปที่ 3.11 Flowchart Program E-Vote | 69 |
| รูปที่ 3.12 flowchart Program E-Vote Machine | 70 |
| รูปที่ 3.13 แสดง E-R Model ที่ได้จากการออกแบบ | 71 |
| รูปที่ 4.1 หน้าต่างการล็อกอินเข้าสู่ระบบ | 83 |
| รูปที่ 4.2 แสดงฐานข้อมูลของผู้ลงสมัคร | 84 |
| รูปที่ 4.3 แสดงฐานข้อมูลของเจ้าหน้าที่ที่ดูแลหน่วยลงคะแนนนั้น | 84 |
| รูปที่ 4.4 แสดงฐานข้อมูลของผู้มาใช้สิทธิ | 86 |
| รูปที่ 4.5 แสดงการกรอกข้อมูลประวัติของผู้สมัครโดยเจ้าหน้าที่ที่หน่วยลงคะแนน | 87 |
| รูปที่ 4.6 แสดงการกรอกข้อมูลประวัติของผู้มาลงคะแนน | 87 |
| รูป เครื่องอ่าน – เขียนบัตรสมาร์ทการ์ด | |
| รูป เครื่องลงคะแนน | |

บทที่ 1

บทนำ

ในระบบการเลือกตั้งแบบเก่าไม่มีการใช้เทคโนโลยีมาเกี่ยวข้องทำให้เกิดปัญหาล่าช้าต่อการนับคะแนนและรวบรวมคะแนนของแต่ละท้องที่ซึ่งกว่าจะทราบผลการเลือกตั้งก็ใช้เวลานาน ปัญหาที่พบอีกอย่างก็คือ บัตรเสียซึ่งเกิดจากผู้ที่มาเลือกตั้งบางคนทำผิดขั้นตอนการลงคะแนนเช่น กากบาทมากกว่า 2 เลขหมายพร้อมกัน จึงทำให้เกิดแนวความคิดในการจัดทำโครงการนี้ จะเป็นการลงคะแนนแบบปุ่มกดช่วยให้เกิดความสะดวกในการลงคะแนน และลดปัญหาบัตรเสียเนื่องจากไม่สามารถกดเครื่องลงคะแนนซ้ำได้อีกเป็นครั้งที่ 2 หลังจากการปิดหีบลงผลคะแนนแล้วการนับคะแนนและรวบรวมผลคะแนนของแต่ละท้องที่ทำให้รู้ผลคะแนนการเลือกตั้งของแต่ละท้องที่ได้รวดเร็ว และยังสามารถช่วยลดปัญหาการนับคะแนนแบบเก่าที่ใช้คนเป็นผู้นับจึงทำให้เกิดความล่าช้าซึ่งกว่าจะรู้ผลคะแนนบางครั้งต้องรอเป็นเวลาหลายวัน อีกทั้งยังสามารถตรวจสอบผลคะแนนภายหลังได้อีก ทำให้การพัฒนาาระบบเลือกตั้งดีขึ้นและเป็นที่ยอมรับแก่คนทั่วไป และในโครงการนี้เป็นการทำงานที่นำไมโครคอนโทรลเลอร์มาประยุกต์ใช้งานร่วมกับระบบฐานข้อมูลเพื่อให้เครื่องลงคะแนนมีประสิทธิภาพมากขึ้นและสามารถนำมาใช้งานได้จริงในอนาคต



บทที่ 2

ทฤษฎีและหลักการ

2.1 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชิปอุปกรณ์อิเล็กทรอนิกส์แบบหนึ่งซึ่งรวมเอาหน่วยประมวลผล หน่วยคำนวณคณิตศาสตร์ และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาต์พุต หน่วยความจำ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถใช้งานแทนวงจรอิเล็กทรอนิกส์ที่ซับซ้อนได้เป็นอย่างดี ช่วยลดจำนวนอุปกรณ์ และขนาดของระบบ ในขณะที่มีความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม AT89S51 เป็นไมโครคอนโทรลเลอร์ MCS-51 ซึ่งมีหน่วยความจำภายในแบบแฟลช (Flash Memory) ของ ATMEL Corporation เป็นไอซีแบบ DIP มีขาทั้งหมด 40 ขาโดยขาต่างๆจะใช้เป็นขาพอร์ตอินพุต, เอาต์พุต, ขาสัญญาณควบคุม, ขาค่าแห่งหน่วยความจำ และขาข้อมูล

2.1.1 คุณสมบัติของไมโครคอนโทรลเลอร์ MCS-51

1. ต้องการแหล่งจ่ายไฟ 5 โวลต์เพียงชุดเดียว
2. มีวงจรสื่อสารอนุกรมแบบ full duplex เป็นไมโครคอนโทรลเลอร์ 8 บิต
3. มีวงจรออสซิลเลเตอร์ และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
4. หน่วยความจำภายในไมโครคอนโทรลเลอร์เป็นแบบแฟลช ทำให้สามารถลบและเขียนและเขียนใหม่ได้นับพันครั้ง
5. ใช้ในรูปแบบไมโครคอนโทรลเลอร์ชิปเดี่ยว ไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุต-เอาต์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ
6. มีหน่วยความจำโปรแกรมภายในตัว (On-chip program memory) ขนาด 4 กิโลไบต์
7. มีหน่วยความจำข้อมูลภายในตัว (On-chip data memory) ขนาด 128 ไบต์
8. ขาพอร์ต 8 บิต จำนวน 4 พอร์ตเป็นแบบกึ่งสองทิศทาง (Quasibidirectional) สามารถใช้งานได้ทั้งอินพุตและเอาต์พุต
9. มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งสามารถทำการรับ และส่งข้อมูลในลักษณะ 2 ทิศทางในเวลาเดียวกัน (full duplex)
10. สามารถรองรับแหล่งอินเตอร์รัปต์ได้ 6 ประเภท
11. Timer / counter ขนาด 16 บิต 2 ตัว (Timer 0.1)
12. สามารถขยายหน่วยความจำภายนอกได้สูงสุด 64 กิโลไบต์

2.1.2 การจัดหาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ทุกเบอร์มีตำแหน่งขาพื้นฐานที่เหมือนกัน ดังแสดงในรูปที่ 2.1 สำหรับการใช้งานแต่ละขามีดังนี้

2.1.2.1 ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

2.1.2.2 ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

2.1.2.3 ขาพอร์ต 0 (P0.0 – P0.7)

มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการให้กำหนดขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้โดยการเขียนข้อมูล “ 1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอิมพีแดนซ์สูงสามารถใช้งานเป็นพอร์ตอินพุตได้



รูปที่ 2.1 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51

2.1.2.4 ขาพอร์ต 1 (P1.0-P1.7)

มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไปถ้าต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล “ 1” ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอิมพีแดนซ์สูงสามารถใช้งานเป็นพอร์ตอินพุตได้

2.1.2.5 ขาพอร์ตที่ 2 (P2.0-P2.7)

มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการกติกด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอิมพีแดนซ์สูงสามารถใช้งานเป็นพอร์ตอินพุตได้

2.1.2.6 ขาพอร์ตที่ 3 (P3.0-P3.7)

มี 8 ขาแต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการกติกด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอิมพีแดนซ์สูงสามารถใช้งานเป็นพอร์ตอินพุตได้ และนอกจากนี้พอร์ตนี้ยังถูกใช้งานในหน้าที่พิเศษต่างๆ ดังแสดงในตารางที่ 2.1

| บิต | ชื่อ | หน้าที่พิเศษ |
|------|------|------------------------------------|
| P3.0 | RXD | ใช้รับข้อมูลทางพอร์ตอนุกรม |
| P3.1 | TXD | ใช้ส่งข้อมูลทางพอร์ตอนุกรม |
| P3.2 | INT0 | อินเทอร์รัพภายนอกหมายเลข 0 |
| P3.3 | INT1 | อินเทอร์รัพภายนอกหมายเลข 1 |
| P3.4 | T0 | ตัวจับเวลา / ตัวนับตัวที่ 0 |
| P3.5 | T1 | ตัวจับเวลา / ตัวนับตัวที่ 1 |
| P3.6 | WR | สัญญาณเขียนข้อมูลหน่วยความจำภายนอก |
| P3.7 | RD | สัญญาณอ่านข้อมูลหน่วยความจำภายนอก |

ตารางที่ 2.1 แสดงบิตและหน้าที่ต่างๆ ของพอร์ต 3

2.1.2.7 ขา RST (Reset)

ขา RST ได้แก่ ขา 9 จะใช้ในการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ MCS-51 โดยจะให้ขานี้เป็นลอจิก "1" อย่างน้อย 2 แมซซีนไซเคิล (Machine Cycles) จึงจะรีเซ็ตระบบได้

2.1.2.8 ขา ALE / PROG (Address Latch Enable / Program Pulse Input)

เป็นขาที่ 30 ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์

2.1.2.9 ขา PSEN (Program Store Enable)

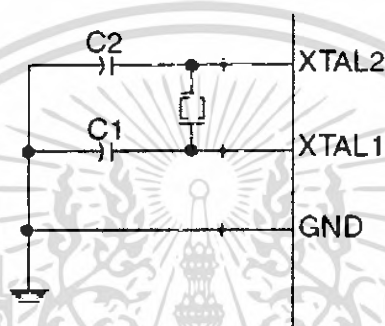
ขา PSEN เป็นขาที่ส่งสัญญาณออก คือ ขาที่ 29 ขานี้จะ Active เมื่อ MCS-51 ต้องการอ่านโค้ดหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลโปรแกรมคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณ Strobe จำนวน 2 ครั้งใน 1 แมซซีนไซเคิล แต่ในขณะที่ติดต่อกับหน่วยความจำข้อมูลภายนอกจะไม่มีส่งสัญญาณ Strobe แต่อย่างใด

2.1.2.10 ขา EA (External Access)

ขา EA ได้แก่ ขาที่ 31 ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าขานี้เป็นลอจิก “0” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็นลอจิก “1” เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อหน่วยความจำโปรแกรมภายในไมโครคอนโทรลเลอร์

2.1.2.11 XTAL1 และ XTAL2

วงจรรอสซิติเลเตอร์ (Oscillator) บนชิปได้แก่ ขา 18 และ 19 โดยต่อคริสตัล (Crystal) เข้ากับขานี้ โดยปรกติมักจะใช้คริสตัลความถี่ 11 MHz กับตัวเก็บประจุหรืออาจใช้สัญญาณนาฬิกาจาก TTL Clock Source ต่อกับ XTAL1 และ XTAL2 ดังรูปที่ 2.3



รูปที่ 2.2 แสดงการต่อขาของ MCS-51 ที่ใช้ต่อกับ XTAL

ใน AT89S51 จะใช้แหล่งจ่ายไฟ 5 V. ต่อเข้ากับขา Vcc (40) ส่วนขา Vss (20) จะต่อลงกราวด์

2.1.3 การจัดการหน่วยความจำของไมโครคอนโทรลเลอร์

ใน MCS-51 แบ่งชนิดและหน้าที่ของหน่วยความจำออกเป็น 2 ส่วน คือ หน่วยความจำโปรแกรม (Program Memory) และหน่วยความจำข้อมูล (Data Memory)

2.1.3.1 หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมสามารถแบ่งได้ออกเป็น 2 ส่วน คือ หน่วยความจำข้อมูลภายในและหน่วยความจำข้อมูลภายนอก หน่วยความจำภายในจะถูกเลือกใช้งานถ้าขาสัญญาณ EA มีค่าเป็น 1 โดยจะถูกใช้งานในช่วง address 0-FFFFH จะถูกใช้หน่วยความจำภายนอก หรือกล่าวได้ว่าถ้าขาสัญญาณ EA มีค่าเป็น 0 จะเป็นการเลือกให้หน่วยความจำโปรแกรมภายนอกทั้งหมดตลอดช่วง Address

2.1.3.2 หน่วยความจำข้อมูล

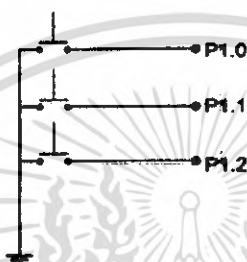
หน่วยความจำข้อมูลสามารถแบ่งออกได้เป็น 2 ส่วน คือ หน่วยความจำข้อมูลภายใน และหน่วยความจำข้อมูลภายนอก สำหรับหน่วยความจำข้อมูลภายในสามารถแบ่งได้เป็น 2 ส่วนย่อย คือ ส่วนที่ใช้เก็บข้อมูลทั่วไป และส่วนที่ใช้เป็นรีจิสเตอร์หน้าที่พิเศษ หรือ SFR (Special Function Register) โดยส่วนที่ใช้เก็บข้อมูลทั่วไปจะถูกใช้สำหรับเก็บข้อมูลหรือค่าตัวแปรต่างๆ จากการทำงานของโปรแกรม ส่วนรีจิสเตอร์หน้าที่พิเศษจะถูกใช้งานเป็นรีจิสเตอร์ควบคุมการทำงาน และบอกสถานะการทำงานของไมโครคอนโทรลเลอร์

ก็ตามค่าข้อมูลที่เก็บไว้ในรีจิสเตอร์แวงค์โคเบงค์หนึ่งก็ตามที่มีชื่อเดียวกัน แต่อยู่คนละแวงค์จะไม่มีผลซึ่งกันและกันเลย

2.2 การเชื่อมต่อสวิทช์เข้ากับระบบไมโครคอนโทรลเลอร์

2.2.1 การเชื่อมต่อกับสวิทช์

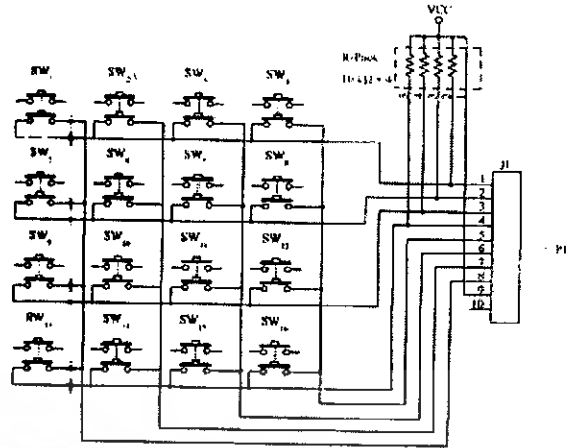
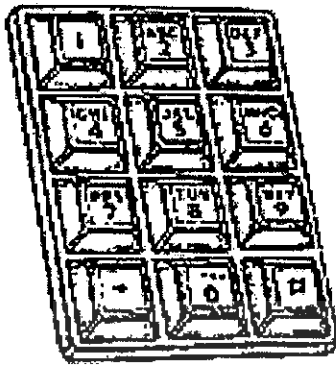
สวิทช์เป็นอุปกรณ์อินพุตพื้นฐานที่ระบบไมโครคอนโทรลเลอร์สามารถรับข้อมูลได้ การสร้างสวิทช์ให้กับระบบไมโครคอนโทรลเลอร์นั้นสามารถทำได้ดังรูปที่ 2.5 โดยการต่อเข้ากับแต่ละบิตของพอร์ตอินพุต ถ้าสวิทช์ ON จะให้ลอจิก “0” แต่ถ้าสวิทช์ OFF จะให้ลอจิก “1” ระบบนี้เป็นการต่อสวิทช์โดยตรงแบบง่ายที่สุดเหมาะสำหรับระบบที่ไม่ต้องการสวิทช์มากนัก



รูปที่ 2.4 แสดงการสร้างลอจิกจากสวิทช์

2.2.2 การเชื่อมต่อกับคีย์แพด (Key pad)

การเชื่อมตอสวิทช์มีด้วยกัน 2 ลักษณะใหญ่ๆ คือ ต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง เมื่อสวิทช์ตัวใดค่อวงจรสามารถอ่านค่าได้โดยตรงซึ่งไม่มีความซับซ้อน สามารถอ่านค่าของสวิทช์ได้ง่ายและรวดเร็ว แต่มีข้อเสียคือ ถ้าหากจำนวนสวิทช์มีมาก ๆ จำนวนของสายข้อมูลก็จะมีมากตาม ทำให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง วงจรสวิทช์อีกลักษณะหนึ่งคือ การค่อวงจรแบบเมตริกซ์ (Matrix Switch) ดังรูปที่ 2.6 สวิทช์จะถูกค่อกันในแนวแกนตั้งและแนวนอน จะเรียกแนวตั้งว่าหลัก (Column) ในแนวนอนเรียกว่า แถว (Row) ดังนั้นค่าของสวิทช์จะต้องประกอบด้วยตำแหน่งในแนวแกนหลักและแถว กระบวนการที่จะทำให้ได้มาซึ่งค่าของสวิทช์มีขั้นตอนซับซ้อนพอสมควร แต่วงจรของสวิทช์แบบนี้มีข้อดี คือ สามารถรองรับการเพิ่มของสวิทช์ได้อย่างสะดวกทำให้สวิทช์เมตริกซ์เป็นที่นิยมใช้มากในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติที่มีจำนวนสวิทช์มากกว่า 8 ตัว ในที่นี้จะยกตัวอย่างการคีย์แพดขนาด 4*4

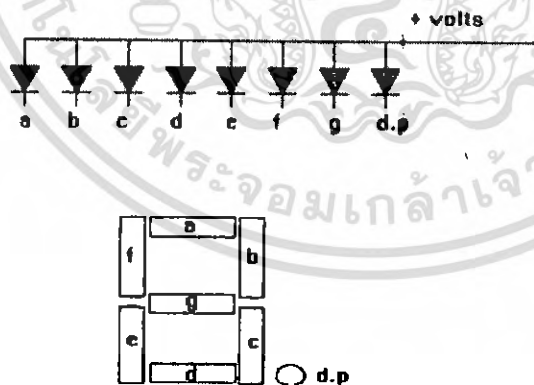


รูปที่ 2.5 แสดงการสร้างสวิทช์เมตริกซ์

2.3 การเชื่อมต่อกับหน่วยแสดงผล

2.3.1 การเชื่อมต่อกับไดโอดเปล่งแสง (LED) 7 ส่วน

การเชื่อมต่อกับไดโอดเปล่งแสงที่แสดงผลแบบตัวเลขได้เรียกว่า ไดโอดเปล่งแสง 7 ส่วน (7-segment Display) หรือ 7 segment ซึ่งมีทั้งแบบแคโทดร่วม (Common-Cathode) และแบบแอนโนดร่วม (Common-Anode) ไดโอดเปล่งแสง 7 ส่วนนี้จะเป็นการรวมไดโอดเปล่งแสง 7 หลอดประกอบกันให้สามารถแสดงเป็นตัวเลขได้ การขับให้ LED ตัวเลข 7 ส่วนแบบแคโทดร่วมสว่างจะต้องจ่ายไฟลบเข้าที่ขา ร่วมแล้วจ่ายไฟบวกเข้าที่ขาแอนโนดซึ่งก็คือขาของแต่ละ segment นั้นเอง ในขณะที่ LED ตัวเลข 7 ส่วนแบบแอนโนดร่วมสว่างจะต้องจ่ายไฟบวกเข้าที่ขาร่วมแล้วจ่ายไฟลบเข้าที่ขาแคโทดซึ่งก็คือขาของแต่ละ segment นั้นเอง



รูปที่ 2.6 แสดงสัญลักษณ์ของแต่ละ segment

ถ้าเราเชื่อมต่อกับแต่ละขากับบัสข้อมูลของไมโครคอนโทรลเลอร์โดยแต่ละบิตจะต่อเข้ากับขา a,b,c,d,e,f,g,d.p ของแต่ละหลอดเปล่งแสง 7 ส่วน ดังตารางที่ 2.2 ถ้าหากต้องการให้ไดโอดเปล่งแสง 7 ส่วนแสดงตัวเลขให้ดูตามบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

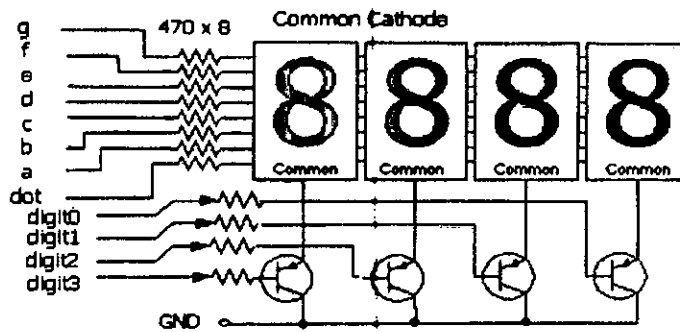
| ตำแหน่ง bit | ตำแหน่ง segment |
|-------------|-----------------|
| 7 | dot |
| 6 | g |
| 5 | f |
| 4 | e |
| 3 | d |
| 2 | c |
| 1 | b |
| 0 | a |

ตารางที่ 2.2 แสดงตำแหน่งขาที่จะต่อกับพอร์ท

ถ้าหากต้องการต่อไดโอดเปล่งแสง 7 ส่วนกับพอร์ทของ AT89S52 ไม่ควรนำมาต่อโดยตรง เนื่องจากความสามารถในการจ่ายกระแสต่อที่ทุกรมไม่สูงมาก จึงต้องอาศัยไอซีบัฟเฟอร์มาช่วยขับ LED เช่น ไอซี 74HC541 และที่เอาต์พุตของไอซีบัฟเฟอร์ที่ต่อกับ LED ตัวเลข 7 ส่วนจะต้องมีตัวต้านทานจำกัดกระแสให้แก่ LED

2.3.2 การเชื่อมต่อกับไดโอดเปล่งแสง (LED) 7 ส่วนหลายๆตัว

ระบบไมโครคอนโทรลเลอร์หากต้องการแสดงผลเป็นตัวเลขหลายๆหลักจำเป็นต้องใช้ไดโอดเปล่งแสง 7 ส่วนหลายๆตัว ไดโอดเปล่งแสง 7 ส่วนหนึ่งตัวจะต้องใช้พอร์ทขนาด 8 บิตหนึ่งพอร์ท หากต้องการแสดงตัวเลข 3 หลักจะต้องใช้พอร์ทถึง 3 พอร์ทซึ่งจะเห็นว่าถ้าต้องการแสดงผลหลายหลักจะต้องใช้พอร์ทหลายพอร์ท มีวิธีหนึ่งที่จะแสดงผลหลายหลักได้โดยจะประหยัดจำนวนพอร์ท และเป็นวิธีที่ใ้ใช้กันโดยทั่วไปเรียกว่า การมัลติเพล็กซ์ (Multiplex) ซึ่งเป็นวิธีการขับให้ LED สว่างทีละหลักด้วยอัตราเร็วที่มนุษย์ไม่สามารถตรวจจับทัน จึงดูเหมือนว่า LED ตัวเลข 7 ส่วนทุกหลักติดสว่างในเวลาเดียวกัน การขับ LED ตัวเลข 7 ส่วนแบบมัลติเพล็กซ์จะทำการต่อขาของแต่ละ segment ร่วมกัน คือ segment a ของทุกหลักจะต่อกันไล่เรียงไปจนถึง segment g ในบางงานที่ต้องการใช้จุด dp ก็ต้องต่อขาของจุด dp ร่วมกันด้วย การใช้ MCS-51 เข้ามาควบคุมการแสดงผลจะจ่ายไฟให้กับขาร่วมของ LED ตัวเลข 7 ส่วนทีละหลักไล่ไปตามลำดับด้วยความเร็วสูง โดยผ่านทรานซิสเตอร์ทั้งนี้การต่อทรานซิสเตอร์เพื่อขับขาคอมมอนนั้นจะช่วยให้ภาระในการจ่ายกระแสของไมโครคอนโทรลเลอร์ลดลงด้วย กระบวนการส่งข้อมูลไปยัง segment a-g และ dp โดยทำการส่งข้อมูล "0" ไปยังทรานซิสเตอร์ที่ต่ออยู่กับขาร่วมของ LED ตัวเลข 7 ส่วนในหลักที่ต้องการแสดงผล แสดงการต่อมัลติเพล็กซ์ในรูปที่ 2.8



รูปที่ 2.7 แสดงการต่อวงจรพลิกของ 7 segment

2.4 ไมโครคอนโทรลเลอร์กับพอร์ตอินพุตและเอาต์พุต

การที่ระบบคอมพิวเตอร์จะติดต่อกับอุปกรณ์ภายนอกได้ เช่น การรับข้อมูลจากอุปกรณ์ภายนอก (Input) หรือส่งข้อมูลให้อุปกรณ์ภายนอก (output) จะต้องติดต่อกันผ่านพอร์ต หรืออาจกล่าวได้ว่า พอร์ต คือ ช่องทางในการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก ซึ่งวิธีรับส่งข้อมูลกับอุปกรณ์ภายนอกนั้นแบ่งได้เป็น 2 ลักษณะคือ การรับส่งข้อมูลด้วยจำนวนบิตของข้อมูลทั้งหมดในเวลาเดียวกัน เรียกว่า พอร์ตแบบขนาน และการรับส่งข้อมูลที่ละบิตจนครบจำนวนของข้อมูลเรียกว่า พอร์ตอนุกรม

2.4.1 พอร์ตของ MCS-51 และการใช้งาน

MCS-51 จะประกอบไปด้วยพอร์ต 4 พอร์ตคือ P0, P1, P2, P3 ซึ่งพอร์ตทั้ง 4 พอร์ตสามารถใช้เป็นอินพุตพอร์ตและเอาต์พุตได้ แต่ส่วนใหญ่มักใช้พอร์ต P1 ติดต่อกับอุปกรณ์ภายนอก เพราะบางเวลา P0,P2,P3 ถูกใช้งานในด้านอื่นด้วย

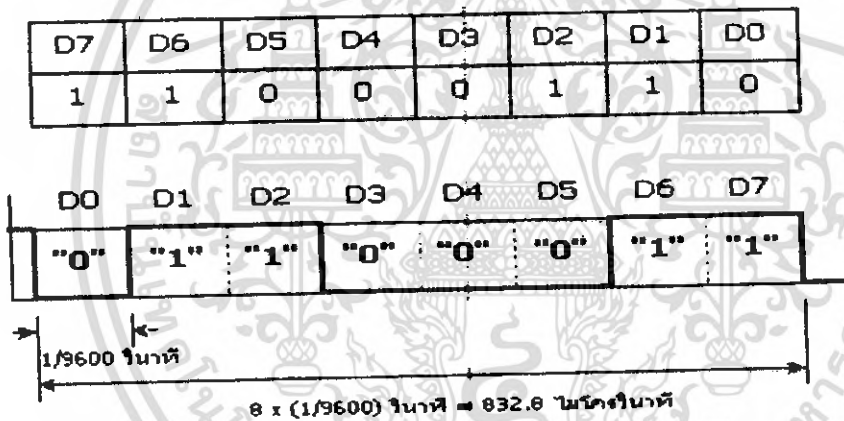
พอร์ตของ MCS-51 จะสร้างจาก FET และจะเห็นว่า P1, P2, P3 จะมีตัวต้านทานเป็นตัว pull upload อยู่ภายในชิปเรียกว่า Internal Pull-up แต่ P0 จะไม่มีดังนั้นการใช้งาน P1, P2, P3 เป็นพอร์ตเอาต์พุตเท่านั้น ถ้ามีการส่งข้อมูลเป็นลอจิก 0 ออกมาจะทำให้ FET หยุดทำงาน ดังนั้นเอาต์พุตจะเป็น 0 ด้วย ถ้ามีการส่งข้อมูลออกมาเป็นลอจิก 1 ด้วยตัวต้านทานที่ pull up อยู่ภายใน แต่ถ้ามีการใช้พอร์ต 0 ถ้าใช้เป็นพอร์ตเอาต์พุตจะอยู่ในสถานะอิมพีแดนซ์สูง ซึ่งต้องต่อตัวต้านทาน pull up ภายนอกด้วย

โดยปกติแล้วการใช้งานพอร์ตเอาต์พุตของคอมพิวเตอร์ จะต้องมีการควบคุมเวลาในการทำงานเพราะว่าคอมพิวเตอร์และอุปกรณ์ภายนอกต้องใช้เวลาในการทำงานแตกต่างกัน อุปกรณ์ภายนอกที่มีความเร็วต่ำ เช่น อุปกรณ์ที่มีส่วนประกอบของ Mechanic อย่างเช่น เครื่องพิมพ์ เราต้องใช้โปรแกรมหน่วงเวลา (Delay) ให้คอมพิวเตอร์ส่งข้อมูลช้าลง อย่างไรก็ดีตามการหน่วงเวลาการส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น จะใช้ได้ในกรณีที่การส่งผ่านข้อมูลนั้นไม่จำเป็นต้องมีความเที่ยง

ตรงตามเวลามากนัก ถ้ารูปแบบการส่งข้อมูลมีความจำเป็นที่ต้องมีความเที่ยงตรงมากจะใช้วิธีที่เรียกว่า Handshake เพื่อควบคุมการส่งข้อมูล

2.5 พอร์ตอนุกรมแบบ RS-232

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบคือ การสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิกาเกี่ยวข้องกับการรับและการส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือ คีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อย 3 เส้นคือ สัญญาณนาฬิกา, ข้อมูลและกราวด์ รูปที่ 2.9 แสดงให้เห็นถึง Timing diagram ของการส่งข้อมูลแบบซิงโครนัส หากมีการส่งข้อมูลด้วยความเร็ว 9600 บิตต่อวินาที จะใช้เวลาในการรับส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ 1/9600 หรือ 104.1 ไมโครวินาที และเวลาในการรับส่งข้อมูลทั้ง 8 บิตจะมีค่า 8×104.1 หรือ 832.8 ไมโครวินาที



รูปที่ 2.8 สัญญาณรูปแบบของข้อมูลอนุกรม

2.5.1 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการผ่านส่งข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกันโดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์(Electronic Industries Association ; EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 ถึง -12 V. แสดงว่ามีข้อมูล(Mark) และที่ +3 ถึง +12 V. แสดงว่าเป็น

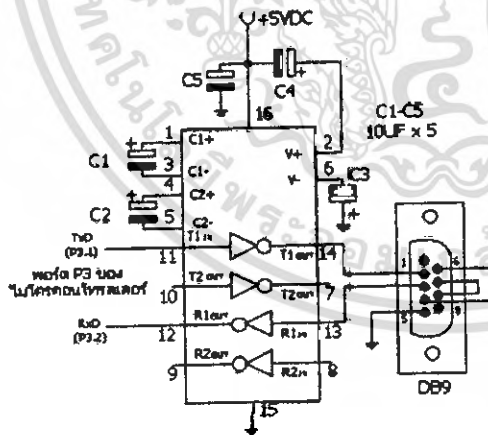
ช่องว่าง (Space) มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating : DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์หรือไมโครคอมพิวเตอร์ซึ่งมีความสามารถในการสร้างข้อมูลแบบอนุกรมได้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

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE ที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่ไม่เต็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ด้วยความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

2.5.2 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

ในการใช้งานพอร์ตอนุกรมกับไมโครคอนโทรลเลอร์ MCS-51 มักนิยมใช้ในการติดต่อเพื่อแลกเปลี่ยนข้อมูลกับคอมพิวเตอร์ผ่านพอร์ตอนุกรม RS-232 เป็นส่วนใหญ่แต่เนื่องจากระดับสัญญาณของพอร์ตอนุกรม RS-232 มีระดับตั้งแต่ +3 ถึง +12 V. ในขณะที่ระดับสัญญาณของไมโครคอนโทรลเลอร์ MCS-51 อยู่ในระดับที่ TTL ถึง 5 V. ดังนั้นจึงไม่สามารถเชื่อมต่อพอร์ตอนุกรมของ Microcontroller MCS-51 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยตรง จึงต้องอาศัยการเชื่อมต่อผ่านไอซีพิเศษที่ทำหน้าที่ในการแปลงระดับสัญญาณ และการต่อใช้งานดังแสดงในรูปที่ 2.11



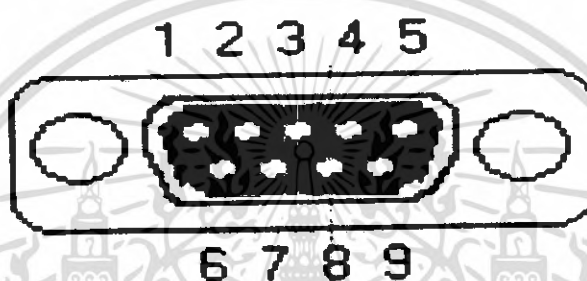
รูปที่ 2.9 วงจรเชื่อมต่อ MAX หรือ ICL232 เข้ากับพอร์ตอนุกรมของคอมพิวเตอร์
คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเนกเตอร์แบบ DB-9 ตัวผู้ ซึ่งคอนเนกเตอร์แบบ DB-9 จะมีขาใช้งาน 9 เส้น สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้

Data Carrier Detect : DCD หรือเรียกว่า Carrier Detect : CD ขานี้จะ Active เมื่อมีการส่งสัญญาณพาห้จากอุปกรณ์สื่อสารข้อมูล เช่น โมเด็ม สำหรับการทำงานปกติขานี้ไม่ได้ใช้งานมาก

Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่เกิดขึ้นในรีจิสเตอร์ บัฟเฟอร์

Transmitter Data : TD หรือ TxD ใช้ส่งข้อมูลออกมาจากคอมพิวเตอร์ โดยนำข้อมูลที่เกิดขึ้นในบัฟเฟอร์สำหรับส่งข้อมูลออกไป



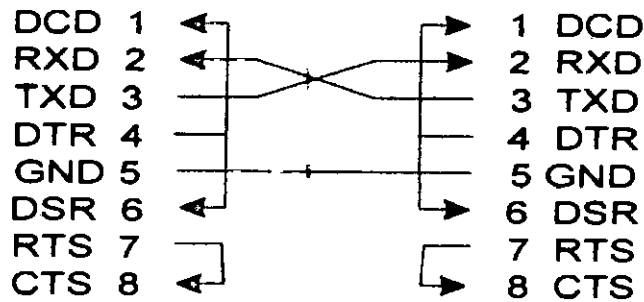
รูปที่ 2.10 คอนเนกเตอร์อนุกรม 9 ขา หรือ DB-9

| คอนเนกเตอร์ DB-9 | ชื่อของสายสัญญาณ | ชนิดของสายสัญญาณ |
|------------------|---------------------------|------------------|
| 1 | Data Carrier Detect : DCD | Input |
| 2 | Receive Data : RxD | Input |
| 3 | Transmitted Data : TxD | Output |
| 4 | Data Terminal Ready : DTR | Output |
| 5 | Signal Ground : GND | - |
| 6 | Data Set Ready : DSR | Input |
| 7 | Request To Send : RTS | Output |
| 8 | Clear To Send : CTS | Input |
| 9 | Ring Indicator : RI | Input |

ตารางที่ 2.3 การจัดขาของคอนเนกเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 แบบ DB-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่อสายสัญญาณ



รูปที่ 2.11 แสดงรายละเอียดของการเชื่อมต่อพอร์ตอนุกรม RS-232

Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วย โดยขา DTR นี้จะเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อเป็น

Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห้

Signal Ground : GND กราวด์ระบบ

Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทางซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งส่งมาจากขา DTR

Request To Send : RTS เป็นขาสำหรับการส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์โดยขาที่รับสัญญาณ RST ก็คือ ขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สายจะต้องเชื่อมต่อกับขา RST และ CTS ของตัวมันเองเข้าด้วยกัน CTS : Clear To Send ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ข้อมูลที่ขา TxD จะถูกส่งออกไปดังนั้น ขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

Ring Indicator : IR ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกนำมาไปใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อโมเด็มหรือโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.6 สมาร์ทการ์ด (SMART CARD)

สมาร์ทการ์ด คือ บัตรที่มีชิป IC (Integrated Circuit) ติดหรือฝังอยู่ในตัวบัตรพลาสติกตามมาตรฐาน ISO (International Standard Organization) เพื่อใช้ในการเก็บข้อมูลและประมวลผลภายในตัวเองโดยวิธีการเข้ารหัสตามมาตรฐาน DES Algorithm (data Encryption Standard) เพื่อให้ระบบมีระดับความปลอดภัยสูงขึ้น ด้วยคุณสมบัติสำคัญประการหนึ่งที่ทำให้สมาร์ทการ์ดมีความแตกต่างจากบัตร

พลาดตกทั่วไปก็คือ ขณะทำรายการ(Transaction) สมาร์ตการ์ดสามารถทำงานได้ด้วยตัวมันเองโดยไม่ต้องอาศัยการติดต่อสื่อสารกับระบบหลัก (Font End) นั่นก็คือ สมาร์ตการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับศูนย์กลางข้อมูลเหมือนกับบัตรแถบแม่เหล็ก (Off-line) ทำให้ประหยัดในเรื่องระบบการสื่อสารไปได้มาก

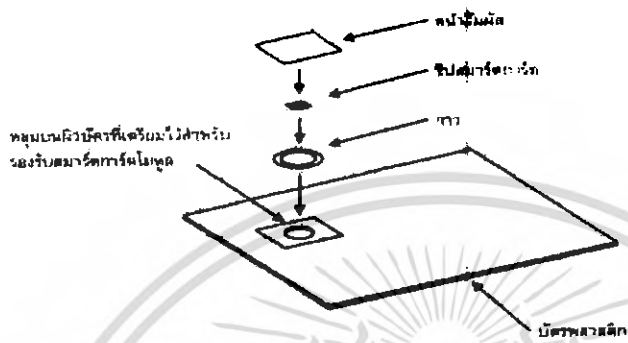
2.6.1 พื้นฐานของสมาร์ตการ์ด

สมาร์ตการ์ดมีพื้นฐานมาจากระบบไมโครโปรเซสเซอร์ โดยเริ่มจากการนำชิปหน่วยความจำ (EEPROM) มาฝังลงในบัตรพลาสติก โดยมีหน้าสัมผัสเป็นขาเชื่อมต่อกับระบบภายนอก ในการเชื่อมต่อต้องมีการป้อนกระแสไฟฟ้าให้ชิปหน่วยความจำสามารถทำงานได้ การตั้งงานเพื่ออ่านหรือเขียนข้อมูลจากชิปหน่วยความจำ ก็ทำได้โดยการเชื่อมต่อสัญญาณผ่านหน้าสัมผัสที่กำหนดไว้แล้ว ในการเชื่อมต่อขาสัญญาณของชิปหน่วยความจำแบบธรรมดา อาจไม่เหมาะสมนักสำหรับบัตรพลาสติกขนาดเล็ก เนื่องจากจำนวนขาสัญญาณของหน่วยความจำ (Bus) มีจำนวนไม่น้อยทีเดียว ยิ่งหน่วยความจำที่มีความจุสูงๆ ยิ่งต้องใช้สัญญาณอ้างอิงตำแหน่งข้อมูล (Address Bus) มากขึ้น จึงมีการนำเอาระบบซิงเกิลบัสมาใช้ในการรับส่งข้อมูล ในการนำเอาระบบสื่อสารอนุกรมมาใช้ จำเป็นต้องมีการป้อนสัญญาณนาฬิกาเพื่อกำกับจังหวะการรับ-ส่งข้อมูลแต่ละบิต ทำให้ต้องมีหน้าสัมผัสสำหรับสัญญาณนาฬิกาบนชิปสมาร์ตการ์ดเพิ่มขึ้นมาแต่ก็นับว่าทำให้ขาเชื่อมต่อลดลงไปได้ไม่น้อยทีเดียว ด้วยเหตุนี้สมาร์ตการ์ดชนิดหน่วยความจำจึงเป็นสมาร์ตการ์ดชนิดแรกที่ถูกสร้างขึ้นมา การนำชิปหน่วยความจำมาใส่ในบัตรพลาสติก ทำให้เกิดข้อดีคือ มีความจุข้อมูลที่มากกว่า ไม่มีผลต่อสนามแม่เหล็กไฟฟ้าและรอยขีดข่วนทำให้สมาร์ตการ์ดเล่นกว่าบัตรแถบแม่เหล็ก แต่ข้อเสียคือ การใช้หน่วยความจำเพียงอย่างเดียวคือ สามารถทำการอ่านและเขียนข้อมูลได้อย่างอิสระเช่นเดียวกับบัตรแถบแม่เหล็กจึงถือว่า ความปลอดภัยของข้อมูลเกือบเป็น 0 นั่นคือข้อมูลภายในสมาร์ตการ์ดชนิดนี้ไม่เป็นความลับ ด้วยเหตุนี้จึงมีการเพิ่มวงจรสำหรับป้องกันลงไปอีก เพื่อให้ผู้ออกบัตร (Card Issue) สามารถกำหนดคีย์ในการเข้าถึงข้อมูลแต่ละ บิตด้วยวงจรพีดีซีเอ็มซี (PDC) ที่เมื่อกำหนดเงื่อนไขไปแล้วไม่สามารถแก้ไขได้อีก ต่อมาเมื่อเทคโนโลยีทางด้านเซมิคอนดักเตอร์สูงขึ้น จึงมีการออกแบบวงจรที่สามารถกำหนดเป็นกุญแจรหัส (PIN) สำหรับเข้าถึงข้อมูลในบัตร ซึ่งต้องทำการแสดงกุญแจรหัสทุกครั้งที่ยังบัตรเริ่มทำงาน เพื่อป้องกันการเจาะระบบอีกชั้นหนึ่ง อีกทั้งกุญแจรหัสก็ยังสามารถเปลี่ยนแปลงได้อีกด้วย

ต่อมาได้มีการนำเอาไมโครโปรเซสเซอร์ (ที่จริงแล้วเป็นไมโครคอนโทรลเลอร์ แต่จะขอเรียกว่าไมโครโปรเซสเซอร์เป็นหลัก) มาใส่ลงในสมาร์ตการ์ด ทำให้เกิดสมาร์ตการ์ดชนิดใหม่ที่มีความซับซ้อนยิ่งขึ้น การเข้าถึงข้อมูลไม่สามารถทำได้โดยตรงเหมือนอย่างสมาร์ตการ์ดชนิดหน่วยความจำ การใช้งานสมาร์ตการ์ดชนิดนี้ต้องเขียนขึ้นเป็นชุดคำสั่ง และส่งให้ชิปไมโครโปรเซสเซอร์ทำงานแทน การที่ใส่ชิปไมโครโปรเซสเซอร์ลงในสมาร์ตการ์ดทำให้ต้องมีการเพิ่มหน่วยความจำโปรแกรม (OS - Operating system) สำหรับไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์สามารถทำการประมวลผลคำสั่งต่างๆและสามารถโปรแกรมการเข้าถึงข้อมูล

2.6.2 ส่วนประกอบและโครงสร้างของบัตรสมาร์ทการ์ด

สมาร์ทการ์ดประกอบด้วยบัตรพลาสติก กาวหรือวัสดุที่ใช้เชื่อมต่อ และหน้าสัมผัสที่บรรจุชิปสมาร์ทการ์ดเรียบเรียบร้อยแล้ว ซึ่งส่วนประกอบต่างๆแสดงให้เห็นดังรูปที่ 2.12

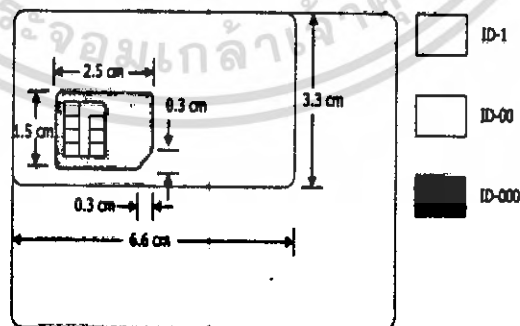


รูปที่ 1.9 ส่วนประกอบของสมาร์ทการ์ด

รูปที่ 2.12 ส่วนประกอบของสมาร์ทการ์ด

2.6.2.1 ตัวบัตรพลาสติก

สมาร์ทการ์ดเป็นชิป IC ขนาดเล็กที่ถูกสร้างขึ้นร่วมกับชิ้นส่วนอิเล็กทรอนิกส์อื่นๆที่สร้างจากสารกึ่งตัวนำ นำมาติดลงบนหน้าสัมผัส และทำการฝังลงในเนื้อพลาสติก ซึ่งพลาสติกที่นิยมนำมาทำเป็นบัตรจะใช้พลาสติก 4 ชนิด ได้แก่ PVC (Polyvinyl Chloride), ABS (acrylonitrile Butadiene Styrene), PC (Polycarbonate) และ PET (Polyethylene Terephthalate) ในประเทศไทยจะใช้บัตรพลาสติกชนิด PVC มากเป็นอันดับหนึ่ง ซึ่งมักจะนำมาใช้ทำบัตร ATM, บัตรเครดิต, บัตรประจำตัวประชาชน เป็นต้น

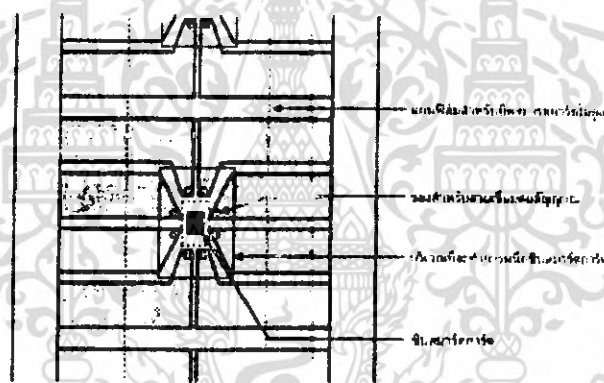


รูปที่ 2.13 การแบ่งชนิดของบัตรตามรูปร่างที่นำไปใช้งาน และขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.2.2 หน้าสัมผัสและชิปสมาร์ทการ์ด (Smart Card Module)

สมาร์ทการ์ดไมโครหรือหน้าสัมผัสและชิปสมาร์ทการ์ดคือ ส่วนที่แสดงความเป็นส่วนตัวของสมาร์ทการ์ดที่ชัดเจนที่สุด สมาร์ทการ์ดบางชนิดเมื่อหยิบขึ้นมาเราอาจไม่ทราบได้เลยว่ามันคือสมาร์ทการ์ดที่มีการฝังชิปไว้ในเนื้อบัตร ดังนั้นการที่จะระบุว่าบัตรใบใดเป็นบัตรสมาร์ทการ์ดนั้น ต้องดูที่หลักการทำงานและลูกเล่นของบัตรเป็นหลัก ซึ่งต้องใช้ประสบการณ์เกี่ยวกับสมาร์ทการ์ดพอสมควร แต่ในที่นี้จะขอแนะนำให้เห็นภาพลักษณะที่ชัดเจนของสมาร์ทการ์ดเป็นหลัก ซึ่งก็คือส่วนของสมาร์ทการ์ดนั่นเอง ในการผลิตสมาร์ทการ์ดไมโคร ส่วนที่เป็นหน้าสัมผัสของสมาร์ทการ์ดประกอบด้วยคลหะหลายชิ้นประกอบกัน แต่ละส่วนจะถูกยึดด้วยแถบฟิล์มบางๆทางด้านหลังของหน้าสัมผัสเพื่อให้คงรูปอยู่ได้ แถบฟิล์มตัวนี้จะมีการเจาะช่องเล็กๆ สำหรับการเชื่อมต่อสายนำสัญญาณกับชิปสมาร์ทการ์ดกับหน้าสัมผัส หลังจากที่วางชิปสมาร์ทการ์ดลงในตำแหน่งที่ต้องการ และเชื่อมต่อสายนำสัญญาณจากชิปสมาร์ทการ์ดเข้ากับหน้าสัมผัสเรียบร้อยแล้ว ขั้นตอนสุดท้ายจะเป็นการฉีกชิปสมาร์ทการ์ดเพื่อป้องกันตัวชิป และสายนำสัญญาณต่างๆจากสิ่งแวดล้อมภายนอก ขั้นตอนที่เหลือจะเป็นการนำหน้าสัมผัสและชิปใส่ลงในบัตรพลาสติกและทดสอบการทำงานของชิปขั้นสุดท้าย



รูปที่ 2.14 ส่วนประกอบของสมาร์ทการ์ดไมโครในการผลิตสมาร์ทการ์ด

2.6.3 องค์ประกอบต่างๆในการใช้งานสมาร์ทการ์ด

1. บัตรและตัวชิป - บัตรและชิปสมาร์ทการ์ดเป็นส่วนแรกที่มีกักตั่วถึงกัน เพราะสมาร์ทการ์ดมีหลากหลายรูปแบบ หลากหลายการใช้งาน โดยหลักการแล้วสมาร์ทการ์ดเป็นเพียงบัตรพลาสติกฝังชิป IC และประมวลผลข้อมูลได้เท่านั้น สมาร์ทการ์ดที่นำมาใช้งานมีตั้งแต่ราคาไม่กี่ร้อยบาทถึงไปทะหลายพันบาท โดยในปัจจุบันเราสามารถเห็นการใช้สมาร์ทการ์ดในหลายรูปแบบเช่น บัตรโทรศัพท์, บัตรนักศึกษา, ชิมการ์ดในโทรศัพท์มือถือ, บัตรเครดิต, บัตรพนักงาน เป็นต้น

2. สมาร์ทการ์ดรีดเดอร์ - โดยภายในสมาร์ทการ์ดรีดเดอร์จะประกอบด้วย ขาสำหรับเชื่อมต่อสัญญาณกับหน้าสัมผัสบนชิปสมาร์ทการ์ด (Card Contact) หรือเป็นเสาอากาศรับส่งคลื่นวิทยุสำหรับ

สมาร์ทการ์ดแบบไม่มีหน้าสัมผัส (contactless) และหน่วยประมวลผลพร้อมหน่วยความจำสำหรับติดต่อสื่อสารกับชิปสมาร์ทการ์ดโดยตรง การสร้างสมาร์ทการ์ดครีเคเตอร์ขึ้นใช้เองสามารถทำได้โดยนำ ไมโคร โปรเซสเซอร์ ไมโครคอนโทรลเลอร์เบอร์วอร์ดก็ได้มาประยุกต์ใช้ในการเชื่อมต่อกับสมาร์ทการ์ด ปัจจุบันได้มีผู้ผลิตสมาร์ทการ์ดครีเคเตอร์ออกมาขายในท้องตลาดหลากหลายรูปแบบซึ่งมีราคาข้อมเขี้ยวและใช้งานได้ง่ายกว่าการสร้างสมาร์ทการ์ดครีเคเตอร์เอง

3. ซอฟต์แวร์ – ซอฟต์แวร์สำหรับการจัดการข้อมูลในสมาร์ทการ์ด และซอฟต์แวร์สำหรับบริหารงานด้านบัตรหรือเรียกอีกอย่างว่า ระบบ Font-End (เหมือนกับระบบบัตรเครดิต) ซึ่งระบบ Font-End ในสมาร์ทการ์ดจะแตกต่างจากระบบบัตรพลาสติก เนื่องจากสมาร์ทการ์ดไม่จำเป็นต้องมีการติดต่อสื่อสารกับ Font-End ทุกครั้งที่ทำรายการเหมือนในบัตรเครดิต ทำให้ระบบ Font-End ของสมาร์ทการ์ดมีเวลามากพอในการบริหารงานด้านอื่นๆ หากต้องการติดต่อสื่อสารกับระบบ Font-End จำเป็นต้องใช้สมาร์ทการ์ดครีเคเตอร์ที่มีส่วนสำหรับการติดต่อสื่อสาร ไม่ว่าจะเป็น Modem , Ethernet , LAN, ระบบสื่อสารด้วยคลื่นวิทยุ, ระบบการสื่อสารอนุกรม RS-485/422 สำหรับการสื่อสารในบริเวณพื้นที่ให้บริการที่ไม่กว้างใหญ่นัก เพื่อใช้สำหรับรับ-ส่งข้อมูลระหว่าง Font-End เมื่อจำเป็น

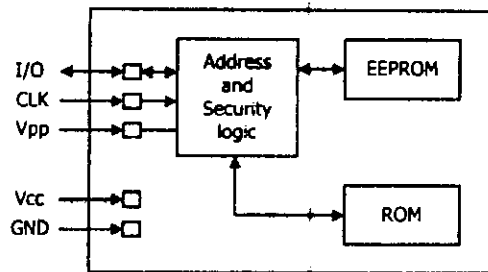
นอกจากระบบ Font-End ยังมีระบบ Back-Office ซึ่งเป็นซอฟต์แวร์สำหรับควบคุมดูแลระบบทั้งหมด ประกอบด้วยซอฟต์แวร์สำหรับป้อนข้อมูลเกี่ยวกับบัตร และผู้ถือบัตรสำหรับออกบัตรใหม่ (Card Issue) , ซอฟต์แวร์สำหรับออกรายงานต่างๆ (Report) และซอฟต์แวร์ส่วนสุดท้ายก็คือซอฟต์แวร์สำหรับให้บริการผู้ถือบัตร เช่น ซอฟต์แวร์สำหรับเติมเงินลงในชิป (สมาร์ทการ์ดที่ใช้บัตรแทนเงินสด) ปกติแล้วซอฟต์แวร์ในส่วนของ Font-End และ Back-Office ต้องทำงานร่วมกับสมาร์ทการ์ดครีเคเตอร์เสมอ เพราะเพียงสมาร์ทการ์ดไม่สามารถใช้ทำรายการใดๆเองได้

2.6.4 ชนิดของสมาร์ทการ์ด

เราสามารถแบ่งสมาร์ทการ์ดจากโครงสร้างภายในได้ 2 ชนิด คือ สมาร์ทการ์ดชนิดหน่วยความจำ (Memory Card) และสมาร์ทการ์ดชนิดไมโคร โปรเซสเซอร์ (Processor Card) ซึ่งชิปทั้งสองแบบจะมีหน้าสัมผัสเหมือนกัน แต่สัญญาณที่ต้องป้อนให้แก่หน้าสัมผัสบางหน้าสัมผัส จะไม่มีการใช้งานในสมาร์ทการ์ดต่างชนิดกัน ซึ่งมีรายละเอียดดังนี้

2.6.4.1 Memory Card (Synchronous Card)

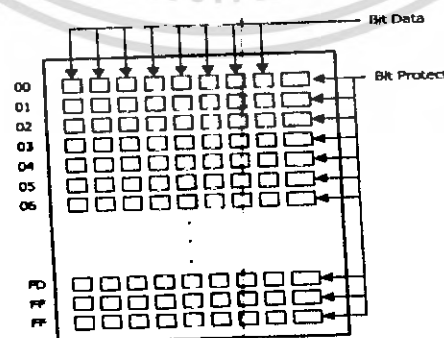
สมาร์ทการ์ดชนิด Memory หรืออีกชื่อหนึ่งคือ Synchronous Card เนื่องจากสมาร์ทการ์ดชนิดนี้มีการรับ-ส่งข้อมูลตามสัญญาณนาฬิกาที่ป้อนให้แก่ชิป (ข้อมูลแต่ละบิตที่ส่งให้แก่ชิปต้องสัมพันธ์กับสัญญาณนาฬิกา) สมาร์ทการ์ดชนิดนี้มีโครงสร้างที่ประกอบไปด้วย ส่วนของวงจรสำหรับการติดต่อสื่อสารภายนอก , หน่วยความจำข้อมูล และหน่วยความจำสำหรับการเก็บชุดคำสั่งของสมาร์ทการ์ดคั้งรูปที่ 2.16



รูปที่ 2.15 Block-diagram โครงสร้างภายในชิปสมาร์ทการ์ดชนิด Memory

สมาร์ทการ์ดที่เป็นพื้นฐานของสมาร์ทการ์ดในปัจจุบันคือ สมาร์ทการ์ดชนิด Free Access Memory สมาร์ทการ์ดชนิดนี้เปิดโอกาสให้อ่านหรือเขียนข้อมูลใน address ใดๆก็ได้ ไม่มีการป้องกันข้อมูลใดๆ ภายในสมาร์ทการ์ดชนิดนี้ซึ่งแน่นอนว่าเป็นสมาร์ทการ์ดที่มีความปลอดภัยต่ำที่สุด ถึงกระนั้นการอ่านข้อมูลก็ไม่ใช่ว่าจะง่ายนักเมื่อมีการออกแบบหน่วยความจำข้อมูลให้มีการสลับตำแหน่งของบิตข้อมูล โดยมีวงจรควบคุมการสลับตำแหน่งของบิตเป็นส่วนป้องกันข้อมูลอีกต่อหนึ่ง ดังนั้นการอ่านข้อมูลออกแบบธรรมดาจะไม่ได้ข้อมูลที่ถูกต้องหากไม่ติดต่อกับวงจรควบคุมการสลับตำแหน่งของบิตโดยตรง

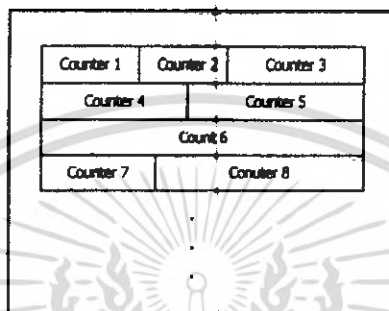
นอกจากนี้สมาร์ทการ์ดชนิด Memory แบบธรรมดา ยังมีการใส่วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลลงไปด้วย ทำให้สามารถกำหนดเงื่อนไขการอ่าน - เขียนข้อมูลได้ทุกไบต์ โดยสมาร์ทการ์ดที่มีวงจรป้องกันการอ่าน - เขียนชนิดนี้ถูกเรียกว่า PIN Protect Memory เนื่องจากการเข้าถึงข้อมูลจะต้องแสดงรหัสผ่านให้บัตรรับทราบก่อนจึงจะสามารถเข้าถึงข้อมูลได้ วงจรกำหนดเงื่อนไขการอ่านเขียนข้อมูลจะมีบิตพิเศษที่มีชื่อว่า Bit Protect ซึ่งเป็นบิตข้อมูลที่ฝากไว้กับข้อมูลให้เป็นบิตที่ 9 แต่ไม่สามารถแก้ไขด้วยคำสั่งเขียนข้อมูลธรรมดาเพราะ Bit Protect ไม่ได้เป็นส่วนหนึ่งของข้อมูลจริงๆ ในการแก้ไข Bit Protect นี้จะสามารถทำการเปลี่ยนแปลงได้เพียงครั้งเดียวด้วยคำสั่งเฉพาะเท่านั้นเช่น หากต้องการบังคับไม่ให้ข้อมูลไบต์ใดไม่สามารถแก้ไขได้ก็ให้ทำการเคลียร์บิตที่ 9 ของข้อมูลไบต์นั้น ๆ แต่สำหรับรหัสผ่านในการเข้าถึงข้อมูลสามารถเปลี่ยนแปลงได้แต่ต้องแสดงรหัสผ่านชุดเก่าให้บัตรได้รับทราบเสียก่อนจึงจะสามารถเปลี่ยนแปลงรหัสผ่านได้



รูปที่ 2.16 ตัวอย่างหน่วยความจำของ Memory card ชนิด PIN Protect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมาร์ตการ์ดอีกชนิดหนึ่งที่มีใช้เป็นบัตรโทรศัพท์ในประเทศไทยนั้นคือ Memory card ชนิด Token ภายในสมาร์ตการ์ดชนิดนี้จะมีการเก็บข้อมูลในลักษณะของจำนวนนับ (Counter) ซึ่งจำนวนนับนี้จะเป็นตัวแทนมูลค่าของเงินที่ระบุบนบัตร การนับเลขเป็นการนับถอยหลังเพื่อเป็นการนับมูลค่าที่คงเหลือในบัตร หมายความว่าหากใช้บัตรในการโทรศัพท์ไปเรื่อย ๆ มูลค่าในบัตรก็จะถูกตกลงตามไปด้วยเช่นกัน ในการเข้าถึงข้อมูลของสมาร์ตการ์ดชนิดนี้ต้องมีการแสดงรหัสผ่านให้บัตรรับทราบเหมือนกับ Memory card ชนิด PIN Protect แต่ไม่มี Bit Protect เอง

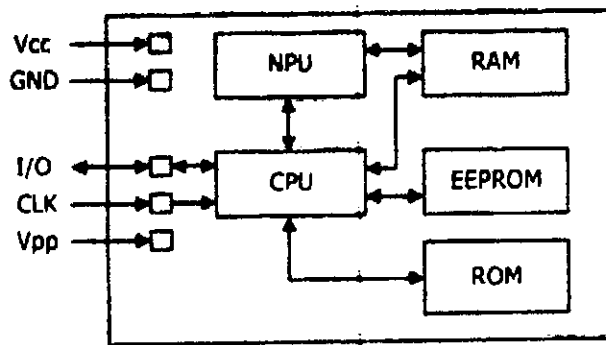


รูปที่ 2.17 ตัวอย่างหน่วยความจำของสมาร์ตการ์ดชนิด Memory แบบ Token

สมาร์ตการ์ดชนิด Memory เป็นสมาร์ตการ์ดที่เป็นพื้นฐานของสมาร์ตการ์ดรุ่นใหม่ ๆ ในปัจจุบัน ด้วยโครงสร้างและการทำงานที่ง่ายต่อการทำความเข้าใจ ราคาถูก สามารถเก็บข้อมูลได้จำนวนมาก และความเร็วในการทำงานของชิปไม่สูงนัก จึงทำให้สมาร์ตการ์ดชนิดนี้เหมาะที่จะนำไปประยุกต์ใช้กับงานที่ข้อมูลไม่ค่อยสำคัญมากนักเช่น บัตรลงเวลาทำงาน, บัตรผ่านประตู, บัตร โทรศัพท์, ฯลฯ ปัจจุบันสมาร์ตการ์ดชนิด Memory มีขนาดหน่วยความจำสูงสุดถึง 64 กิโลไบต์ และอีกไม่นานนักเราจะได้เห็นสมาร์ตการ์ดที่มีขนาดหน่วยความจำข้อมูลถึง 128 กิโลไบต์

2.6.4.2 Processor card (Asynchronous card)

สมาร์ตการ์ดชนิดนี้เป็นสมาร์ตการ์ดที่ได้รับการปรับปรุงจากสมาร์ตการ์ดชนิด Memory ด้วยการใส่เทคโนโลยีไมโครโปรเซสเซอร์เข้าไปในชิป เพื่อให้ชิปสามารถประมวลผลข้อมูล และเพิ่มความปลอดภัยให้แก่ข้อมูลที่สูงขึ้น การที่ใส่ไมโครโปรเซสเซอร์ลงในชิปทำให้จำเป็นต้องมีการเพิ่มส่วนของหน่วยความจำสำหรับจัดเก็บระบบปฏิบัติการของไมโครโปรเซสเซอร์ และหน่วยความจำชั่วคราวสำหรับการประมวลผลข้อมูล นอกจากนี้ยังมีการใส่ชิปประมวลผลทางคณิตศาสตร์ลงในชิปสมาร์ตการ์ดเพื่อช่วยในการประมวลผลข้อมูลด้วยอัลกอริทึมสำหรับเข้ารหัส – ถอดรหัส ทำให้สมาร์ตการ์ดชนิด Processor มีความเร็วในการทำงานสูงกว่าสมาร์ตการ์ดชนิด Memory หลายเท่า



รูปที่ 2.18 บล็อกไออะแกรมโครงสร้างภายในชิปสมาร์ตการ์ดชนิด

Processor card (Asynchronous card)

ในการรับส่งข้อมูลให้กับสมาร์ตการ์ดชนิดนี้ จะใช้หน้าสัมผัสเดียวกับสมาร์ตการ์ดชนิด Memory โดยสัญญาณนาฬิกาที่ป้อน จะถูกใช้เป็นสัญญาณนาฬิกาให้แก่โปรเซสเซอร์ภายในสมาร์ตการ์ด ข้อมูลที่รับ - ส่งจึงไม่จำเป็นต้องสัมพันธ์กับสัมพันธ์กับสัญญาณนาฬิกาที่ป้อนให้แก่ชิป เพียงกำหนดอัตรา การรับ - ส่งข้อมูลเป็น 9600 บิต/วินาที ก็จะสามารถติดต่อกับ โปรเซสเซอร์ของชิปได้แล้ว แต่การเข้าถึง ข้อมูลจะไม่สามารถทำได้เหมือนอย่างในสมาร์ตการ์ดชนิด Memory การเข้าถึงข้อมูลต้องกระทำผ่านโปร เซสเซอร์ของสมาร์ตการ์ดเท่านั้น ไม่ว่าจะเป็นการอ่านหรือเขียนข้อมูลก็ตาม เพราะ หน่วยความจำจะอยู่ ภายในความควบคุมของโปรเซสเซอร์เพียงอย่างเดียว ชื่อคืออย่างหนึ่งที่ไม่สามารถติดต่อกับหน่วยความจำ ในชิปโดยตรงก็คือ การทอบเข้าถึงข้อมูลโดยไม่ได้รับอนุญาตแทบเป็นไปไม่ได้ ยกเว้นมีความบกพร่องใน การกำหนดเงื่อนไขในการเข้าถึงข้อมูลที่เป็นความลับ

2.7 ความรู้พื้นฐานเกี่ยวกับระบบฐานข้อมูล

ระบบฐานข้อมูล (Database System) หมายถึง โครงสร้างสารสนเทศที่ประกอบด้วยรายละเอียดของ ข้อมูลที่เกี่ยวข้องกันที่จะนำมาใช้ในระบบต่าง ๆ ร่วมกัน ระบบฐานข้อมูล จึงนับว่าเป็นการจัดเก็บข้อมูล อย่างเป็นระบบ ซึ่งผู้ใช้สามารถจัดการกับข้อมูลได้ในลักษณะต่าง ๆ ทั้งการเพิ่ม การแก้ไข การลบ ตลอดจน การเรียกดูข้อมูล ซึ่งส่วนใหญ่จะเป็นการประยุกต์นำเอาระบบคอมพิวเตอร์เข้ามาช่วยในการจัดการฐาน ข้อมูล

2.7.1 นิยามและคำศัพท์พื้นฐานเกี่ยวกับระบบฐานข้อมูล

บิต (Bit) หมายถึง หน่วยของข้อมูลที่มีขนาดเล็กที่สุด

ไบท์ (Byte) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำบิตมารวมกันเป็นตัวอักษร (Character)

เขตข้อมูล (Field) หมายถึง หน่วยของข้อมูลที่ประกอบขึ้นจากตัวอักษรตั้งแต่หนึ่งตัวขึ้นไปมา รวมกันแล้วให้ความหมายของสิ่งใดสิ่งหนึ่ง เช่น ชื่อ ที่อยู่ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระเบียบ (Record) หมายถึง หน่วยของข้อมูลที่เกิดจากการนำเอาเขตข้อมูลหลาย ๆ เขตข้อมูลมารวมกัน เพื่อเกิดเป็นข้อมูลเรื่องใดเรื่องหนึ่ง เช่น ข้อมูลของนักศึกษา 1 ระเบียบ (1 คน) จะประกอบด้วย

รหัสประจำตัวนักศึกษา 1 เขตข้อมูล ,ชื่อนักศึกษา 1 เขตข้อมูล , ที่อยู่ 1 เขตข้อมูล

แฟ้มข้อมูล (File) หมายถึงหน่วยของข้อมูลที่เกิดจากการนำข้อมูลหลาย ๆ ระเบียบที่เป็นเรื่องเดียวกันมารวมกัน เช่น แฟ้มข้อมูลนักศึกษา แฟ้มข้อมูลลูกค้า แฟ้มข้อมูลพนักงาน

2.7.2 ความสำคัญของการประมวลผลแบบระบบฐานข้อมูล

จากการจัดเก็บข้อมูลรวมเป็นฐานข้อมูลจะก่อให้เกิดประโยชน์ดังนี้

1. สามารถลดความซ้ำซ้อนของข้อมูลได้

การเก็บข้อมูลชนิดเดียวกันไว้หลาย ๆ ที่ ทำให้เกิดความซ้ำซ้อน (Redundancy) ดังนั้นการนำข้อมูลมารวมเก็บไว้ในฐานข้อมูล จะช่วยลดปัญหาการเกิดความซ้ำซ้อนของข้อมูลได้ โดยระบบจัดการฐานข้อมูล (Database Management System : DBMS) จะช่วยควบคุมความซ้ำซ้อนได้ เนื่องจากระบบจัดการฐานข้อมูลจะทราบได้ตลอดเวลาว่ามีข้อมูลซ้ำซ้อนกันอยู่ที่ใดบ้าง

2. หลีกเลี่ยงความขัดแย้งของข้อมูลได้

หากมีการเก็บข้อมูลชนิดเดียวกันไว้หลาย ๆ ที่ และมีการปรับปรุงข้อมูลเดียวกันนี้ แต่ปรับปรุงไม่ครบทุกที่ ที่มีข้อมูลเก็บอยู่ก็จะทำให้เกิดปัญหาข้อมูลชนิดเดียวกัน อาจมีค่าไม่เหมือนกันในแต่ละที่ที่เก็บข้อมูลอยู่ จึงก่อให้เกิดความขัดแย้งของข้อมูลขึ้น (Inconsistency)

3. สามารถใช้ข้อมูลร่วมกันได้

ฐานข้อมูลจะเป็นการจัดเก็บข้อมูลรวมไว้ด้วยกัน ดังนั้นหากผู้ใช้ต้องการใช้ข้อมูลในฐานข้อมูลที่มาจากแฟ้มข้อมูลต่างๆ ก็จะทำให้ได้โดยง่าย

4. สามารถรักษาความถูกต้องเชื่อถือได้ของข้อมูล

บางครั้งพบว่าการจัดเก็บข้อมูลในฐานข้อมูลอาจมีข้อผิดพลาดเกิดขึ้น เช่น จากการที่ผู้ป้อนข้อมูลป้อนข้อมูลผิดพลาดคือป้อนจากตัวเลขหนึ่งไปเป็นอีกตัวเลขหนึ่ง โดยเฉพาะกรณีมีผู้ใช้หลายคนต้องใช้ข้อมูลจากฐานข้อมูลร่วมกัน หากผู้ใช้คนใดคนหนึ่งแก้ไขข้อมูลผิดพลาดก็ทำให้ผู้อื่นได้รับผลกระทบตามไปด้วย ในระบบจัดการฐานข้อมูล (DBMS) จะสามารถใส่กฎเกณฑ์เพื่อควบคุมความผิดพลาดที่เกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สามารถกำหนดความเป็นมาตรฐานเดียวกันของข้อมูลได้

การเก็บข้อมูลร่วมกันไว้ในฐานข้อมูลจะทำให้สามารถกำหนดมาตรฐานของข้อมูลได้รวมทั้งมาตรฐานต่าง ๆ ในการจัดเก็บข้อมูลให้เป็นไปในลักษณะเดียวกันได้ เช่นการกำหนดรูปแบบการเขียนวันที่ ในลักษณะ วัน/เดือน/ปี หรือ ปี/เดือน/วัน ทั้งนี้จะมีผู้ที่คอยบริหารฐานข้อมูลที่เราเรียกว่า ผู้บริหารฐานข้อมูล (Database Administrator : DBA) เป็นผู้กำหนดมาตรฐานต่าง ๆ

6. สามารถกำหนดระบบความปลอดภัยของข้อมูลได้

ระบบความปลอดภัยในที่นี้ เป็นการป้องกันไม่ให้ผู้ใช้ที่ไม่มีสิทธิมาใช้ หรือมาเห็นข้อมูลบางอย่างในระบบ ผู้บริหารฐานข้อมูลจะสามารถกำหนดระดับการเรียกใช้ข้อมูลของผู้ใช้แต่ละคนได้ตามความเหมาะสม

7. เกิดความเป็นอิสระของข้อมูล

ในระบบฐานข้อมูลจะมีตัวจัดการฐานข้อมูลที่ทำหน้าที่เป็นตัวเชื่อมโยงกับฐานข้อมูล โปรแกรมต่าง ๆ อาจไม่จำเป็นต้องมีโครงสร้างข้อมูลทุกครั้ง ดังนั้นการแก้ไขข้อมูลบางครั้ง จึงอาจกระทำเฉพาะกับโปรแกรมที่เรียกใช้ข้อมูลที่เปลี่ยนแปลงเท่านั้น ส่วนโปรแกรมที่ไม่ได้เรียกใช้ข้อมูลดังกล่าว ก็จะเป็นอิสระจากการ รูปแบบของ เปลี่ยนแปลง

2.7.2 รูปแบบของระบบฐานข้อมูล ฐานข้อมูล มีอยู่ด้วยกัน 3 ประเภท คือ

1. ฐานข้อมูลเชิงสัมพันธ์ (Relational Database) เป็นการเก็บข้อมูลในรูปแบบที่เป็นตาราง (Table) หรือเรียกว่า รีเลชัน (Relation) มีลักษณะเป็น 2 มิติ คือเป็นแถว (row) และเป็นคอลัมน์ (column) การเชื่อมโยงข้อมูลระหว่างตาราง จะเชื่อมโยงโดยใช้แอททริบิวต์ (attribute) หรือคอลัมน์ที่เหมือนกันทั้งสองตารางเป็นตัวเชื่อมโยงข้อมูล ฐานข้อมูลเชิงสัมพันธ์นี้จะเป็นรูปแบบของฐานข้อมูลที่นิยมใช้ในปัจจุบัน

2. ฐานข้อมูลแบบเครือข่าย (Network Database) ฐานข้อมูลแบบเครือข่ายจะเป็นการรวมระเบียบต่าง ๆ และความสัมพันธ์ระหว่างระเบียบแต่ละต่างกับฐานข้อมูลเชิงสัมพันธ์ คือ ในฐานข้อมูลเชิงสัมพันธ์จะแสดงความสัมพันธ์เอาไว้ โดยระเบียบที่มีความสัมพันธ์กันจะต้องมีค่าของข้อมูลในแอททริบิวต์ใดแอททริบิวต์หนึ่งเหมือนกัน แต่ฐานข้อมูลแบบเครือข่าย จะแสดงความสัมพันธ์อย่างชัดเจน ตัวอย่างเช่น

3. ฐานข้อมูลแบบลำดับชั้น (Hierarchical Database) ฐานข้อมูลแบบลำดับชั้น เป็นโครงสร้างที่จัดเก็บข้อมูลในลักษณะความสัมพันธ์แบบพ่อ-ลูก (Parent-Child Relationship Type : PCR Type) หรือเป็นโครงสร้างรูปแบบต้นไม้ (Tree) ข้อมูลที่จัดเก็บในที่นี้ คือ ระเบียบ (Record) ซึ่งประกอบด้วยค่าของเขตข้อมูล (Field) ของเอนทิตีหนึ่ง ๆ

ฐานข้อมูลแบบลำดับชั้นนี้คล้ายคลึงกับฐานข้อมูลแบบเครือข่าย แต่ต่างกันที่ฐานข้อมูลแบบลำดับชั้น มีกฎเพิ่มขึ้นมาอีกหนึ่งประการ คือ ในแต่ละกรอบจะมีลูกศรวิ่งเข้าหาได้ไม่เกิน 1 หัวลูกศร

2.7.3 โปรแกรมฐานข้อมูลที่นิยมใช้

โปรแกรมฐานข้อมูล เป็นโปรแกรมหรือซอฟต์แวร์ที่ช่วยจัดการข้อมูลหรือรายการต่าง ๆ ที่อยู่ในฐานข้อมูล ไม่ว่าจะเป็นการจัดเก็บ การเรียกใช้ การปรับปรุงข้อมูล โปรแกรมฐานข้อมูล จะช่วยให้ผู้ใช้สามารถค้นหาข้อมูลได้อย่างรวดเร็ว ซึ่ง โปรแกรมฐานข้อมูลที่นิยมใช้มีอยู่ด้วยกันหลายตัว เช่น Access, FoxPro, Clipper, dBase, FoxBase, Oracle, SQL เป็นต้น โดยแต่ละ โปรแกรมจะมีความสามารถต่างกัน บางโปรแกรมใช้ง่ายแต่จะจำกัดขอบเขตการใช้งาน บ้างโปรแกรมใช้งานยากกว่า แต่จะมีความสามารถในการทำงานมากกว่า

โปรแกรม Access นับเป็นโปรแกรมที่นิยมใช้กันมากในขณะนี้ โดยเฉพาะในระบบฐานข้อมูลขนาดใหญ่ สามารถสร้างแบบฟอร์มที่ต้องการจะเรียกดูข้อมูลในฐานข้อมูล หลังจากบันทึกข้อมูลในฐานข้อมูลเรียบร้อยแล้ว จะสามารถค้นหาหรือเรียกดูข้อมูลจากเขตข้อมูลใดก็ได้ นอกจากนี้ Access ยังมีระบบรักษาความปลอดภัยของข้อมูล โดยการกำหนดรหัสผ่านเพื่อป้องกันความปลอดภัยของข้อมูลในระบบได้ด้วย

โปรแกรม FoxPro เป็นโปรแกรมฐานข้อมูลที่มีผู้ใช้งานมากที่สุด เนื่องจากใช้ง่ายทั้งวิธีการเรียกจากเมนูของ FoxPro และประยุกต์โปรแกรมอื่นใช้งาน โปรแกรมที่เขียนด้วย FoxPro จะสามารถใช้กับ dBase คำสั่งและฟังก์ชันต่าง ๆ ใน dBase จะสามารถใช้งานบน FoxPro ได้ นอกจากนี้ใน FoxPro ยังมีเครื่องมือช่วยในการเขียนโปรแกรม เช่น การสร้างรายงาน

โปรแกรม dBase เป็นโปรแกรมฐานข้อมูลชนิดหนึ่ง การใช้งานจะคล้ายกับ โปรแกรม FoxPro ข้อมูลรายงานที่อยู่ในไฟล์บน dBase จะสามารถส่งไปประมวลผลใน โปรแกรม Word Processor ได้ และแม้แต่ Excel ก็สามารอ่านไฟล์ .DBF ที่สร้างขึ้นโดยโปรแกรม dBase ได้ด้วย

โปรแกรม SQL เป็นโปรแกรมฐานข้อมูลที่มีโครงสร้างของภาษาที่เข้าใจง่าย ไม่ซับซ้อน มีประสิทธิภาพการทำงานสูง สามารถทำงานที่ซับซ้อนได้โดยใช้คำสั่งเพียงไม่กี่คำสั่ง โปรแกรม SQL จึงเหมาะที่จะใช้กับระบบฐานข้อมูลเชิงสัมพันธ์ และเป็นภาษาหนึ่งที่มีผู้นิยมใช้กันมาก โดยทั่วไปโปรแกรมฐานข้อมูลของบริษัทต่าง ๆ ที่มีใ้ช้ในปัจจุบัน เช่น Oracle, DB2 ก็มักจะมีคำสั่ง SQL ที่ต่างจากมาตรฐานไปบ้างเพื่อให้เป็นจุดเด่นของแต่ละโปรแกรมไป

2.7.4 แบบจำลองข้อมูล (Data Models) การที่ฐานข้อมูลสามารถจัดระบบการเก็บข้อมูลชนิดต่าง ๆ ได้ นั้น จะใช้หลักการของแบบจำลอง (model) ข้อมูล โดยจัดเก็บโครงสร้างของข้อมูลในรูปตาราง ซึ่งประกอบด้วยแถวและหลักที่

| | | |
|--|--|---|
| Larry, Long 124 Lemon Rd. Florida, CA 19243 | Challot, Ann 8 University Ave. Oakland, CA 94709 | Grant, James 54 Yellow St. New York, NY 95104 |
| Philip, Tan 4015 City Ave. San Francisco, CA 94103 | Sidney, Phil 458 Rocklyn Ave. Illinois, CA 94501 | |

รูปที่ 2.19 ตัวอย่างข้อมูล

จากตัวอย่างข้อมูลข้างต้นสามารถนำมาจัดเก็บในรูปของตารางข้อมูล ดังนี้

| Field | | | | | | Record |
|-----------|------------|-------------------|---------------|-------|----------|--------|
| LAST NAME | FIRST NAME | STREET | CITY | STATE | ZIP CODE | |
| Larry | Long | 124 Lemon Rd. | Florida | CA | 19243 | |
| Challot | Ann | 8 University Ave. | Oakland | CA | 94709 | |
| Grant | James | 54 Yellow St. | New York | NY | 95104 | |
| Philip | Tan | 4015 City Ave. | San Francisco | CA | 94103 | |
| Sidney | Phil | 458 Rocklyn Ave. | Illinois | CA | 94501 | |

Data Item

รูปที่ 2.20 ตารางข้อมูล

จากรูปของตารางข้อมูล จะเห็นว่าข้อมูลแต่ละชิ้นที่อยู่ในกรอบสี่เหลี่ยมเรียกว่า ชิ้นข้อมูล (Data Item) เช่นชื่อต้นของคนหนึ่งคน (Philip) หรือชื่อเมืองหนึ่งเมือง (Oakland) เป็นต้น แต่ละหลักของตารางเรียกว่า ฟิวด์ (Field) ซึ่งหมายถึง ชนิดของข้อมูลหนึ่ง ๆ ซึ่งอาจแปรไปตามข้อมูลของแต่ละคน แต่หมายถึงถึงถึงเดียวกัน เช่น ชื่อถนน (street) หรือรหัสเมือง (zip code) เป็นต้น และข้อมูลในหนึ่งแถว เรียกว่า เรคคอร์ด (record) โดยที่แต่ละเรคคอร์ดจะมีจำนวนฟิวด์เท่ากันเสมอ ในกรณีนี้เรคคอร์ดหนึ่ง ๆ อาจหมายถึงฟิวด์หลายฟิวด์ที่สัมพันธ์กัน เช่น ชื่อต้น และชื่อท้ายของคนหนึ่งคนจะเป็นส่วนหนึ่งในเรคคอร์ด และหลาย ๆ เรคคอร์ดรวมกันเรียกว่า ไฟล์ถึงแม้ว่าข้อมูลมากมายหลายอย่างสามารถจัดอยู่ในรูปของตารางได้ แต่บางอย่างก็ทำได้อย่างไม่เหมาะสมนัก เช่น ใบบันทึกการซื้อสินค้าของถูกค้ารายหนึ่งซึ่งมีข้อมูลหลายอย่าง โดยทั่วไปมักมีรูปแบบดังนี้

| CHARGE RECORD | | | | |
|----------------|-------------|----------|-----------|-----------|
| ACCOUNT NUMBER | | | | |
| NAME | | | | |
| ADDRESS | | | | |
| ITEM | DESCRIPTION | QUANTITY | ITEM COST | EXT. COST |
| | | | | |
| | | | | |
| | | | | |

รูปที่ 2.21 ตัวอย่างใบบันทึกการซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากใบบันทึกการซื้อสินค้า เก็บตัวอย่างข้อมูลได้ดังนี้

| CHARGE RECORD | | | | | CHARGE RECORD | | | | |
|-----------------------|-------------|----------|-----------|----------|-----------------------|-------------|----------|-----------|----------|
| ACCOUNT NUMBER: VF326 | | | | | ACCOUNT NUMBER: CA164 | | | | |
| NAME: Philip | | | | | NAME: Tom | | | | |
| ADDRESS: Chicago | | | | | ADDRESS: Maryland | | | | |
| ITEM | DESCRIPTION | QUANTITY | ITEM COST | EXT COST | ITEM | DESCRIPTION | QUANTITY | ITEM COST | EXT COST |
| 15 | Keyboard | 1 | 1,000 | 1,000 | 47 | Diskette | 5 | 250 | 700 |
| | | | | | 43 | Mouse | 2 | 450 | 900 |
| CHARGE RECORD | | | | | CHARGE RECORD | | | | |
| ACCOUNT NUMBER: 10341 | | | | | ACCOUNT NUMBER: VF326 | | | | |
| NAME: Mark | | | | | NAME: Philip | | | | |
| ADDRESS: Florida | | | | | ADDRESS: Chicago | | | | |
| ITEM | DESCRIPTION | QUANTITY | ITEM COST | EXT COST | ITEM | DESCRIPTION | QUANTITY | ITEM COST | EXT COST |
| 47 | Diskette | 4 | 250 | 1,000 | 24 | UPS | 2 | 2,500 | 5,000 |

รูปที่ 2.22 ตัวอย่างข้อมูลที่ได้จากใบบันทึกการซื้อสินค้า

จากตัวอย่างข้อมูลใบบันทึกการซื้อสินค้า จะเห็นได้ว่าในเอกสารนี้มีข้อมูลนำเข้าเพียงอย่างเดียว ส่วนข้อมูลอื่น ๆ นั้นจะเกิดขึ้นซ้ำ ๆ กัน โดยรายละเอียดของสินค้าที่ซื้อแต่ละบรรทัดนั้นจะเกิดขึ้นซ้ำ ๆ กัน ข้อมูลตามเอกสารนี้อาจนำมาจัดในรูปของตารางได้ แต่จะต้องใส่ชื่อ ที่อยู่ หมายเลขบัญชีของลูกค้านั้น ในทุกบรรทัดของตาราง ทำให้ได้ตารางดังรูป

| ACCOUNT | NAME | ADDR | ITEM | DESCRPTIO | QUANTITY | ITEM COST | EXT.COST |
|---------|------|------|------|-----------|----------|-----------|----------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

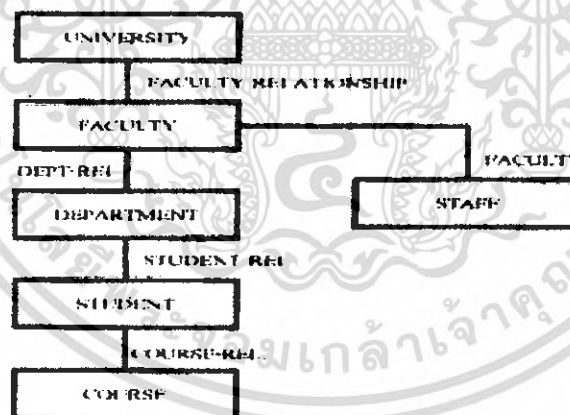
รูปที่ 2.23 ตัวอย่างจากใบบันทึกการซื้อสินค้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางจะเห็นได้ว่าส่วนของ ชื่อ ที่อยู่ และหมายเลขบัญชีของลูกค้าจะมีซ้ำกันในทุกรายการสินค้าที่ซื้อ ทำให้สิ้นเปลืองเนื้อที่ในการจัดเก็บโดยไม่จำเป็น และทำให้อ่านข้อมูลยากกว่าเดิม ดังนั้นจึงต้องมาศึกษาถึงโครงสร้างและการจัดการข้อมูล เพื่อให้เก็บข้อมูลได้อย่างมีประสิทธิภาพสามารถเรียกใช้ได้อย่างรวดเร็ว และใช้เนื้อที่ในการจัดเก็บอย่างเหมาะสม ระบบฐานข้อมูลสามารถแบ่งได้ตามวิธีการใช้งานการวางโครงสร้างและการจัดการข้อมูล รวมทั้งความสัมพันธ์ระหว่างข้อมูลที่จัดเก็บอยู่ เรียกว่า แบบจำลองข้อมูล (Data models) ซึ่งแบ่งเป็น 3 แบบด้วยกันคือ

2.7.5. แบบจำลองแบบลำดับชั้น (The Hierarchical Model)

แบบจำลองแบบลำดับชั้นได้ถูกพัฒนาโดยไอบีเอ็ม ในปีค.ศ. 1968 โดยระบบฐานข้อมูลที่ใช้แบบจำลองประเภทนี้จะเชื่อมโยงข้อมูลที่อยู่ภายใต้ด้วยความสัมพันธ์แบบลำดับชั้น และส่วนมากจะเอาข้อมูลทั้งหมดไว้ในไฟล์ขนาดใหญ่เพียงไฟล์เดียว ในระบบจัดการฐานข้อมูลแบบลำดับชั้นกลุ่มของฟิลด์จะเรียกว่า เซกเมนต์ (segment) แทนการเรียกเรคคอร์ด และชิ้นของข้อมูลซึ่งอยู่บนสุดของลำดับชั้นจะเรียกว่า พาเรนต์อีลิเมนต์ (parent element) ซึ่งจะมี ไชล์อีลิเมนต์ (child element) จำนวนหนึ่งอยู่ระดับถัดจากพาเรนต์อีลิเมนต์ลงมา สมมติว่าต้องการเก็บตัวอย่างข้อมูลของมหาวิทยาลัยแห่งหนึ่ง โดยให้มหาวิทยาลัย (UNIVERSITY) มี m คณะ (FACULTY) แต่ละคณะมี n ภาควิชา (DEPARTMENT) แต่ละภาควิชามีนักเรียน (STUDENT) สังกัดอยู่จำนวน o คน นักเรียนแต่ละคนต้องเรียน p วิชา (COURSE) และแต่ละภาควิชามีอาจารย์ (STAFF) จำนวน q คน จะเขียนโครงสร้างของฐานข้อมูลแบบจำลองแบบลำดับชั้นได้ดังนี้



รูปที่ 2.24 โครงสร้างฐานข้อมูลแบบลำดับชั้น

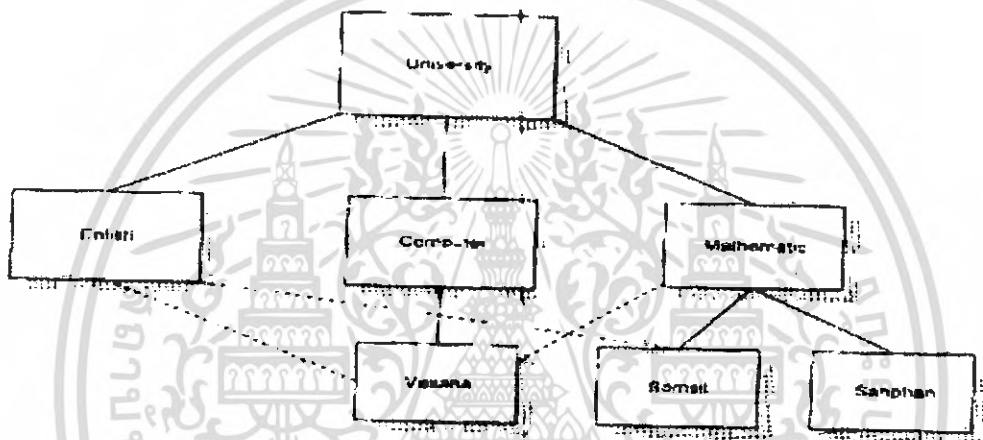
จากรูปจะเห็นได้ว่าแบบจำลองแบบลำดับชั้นเป็นการรวมความสัมพันธ์ระหว่าง parent และ child เข้าด้วยกัน ปัญหาแบบจำลองแบบลำดับชั้นคือ อีลิเมนต์ใดอีลิเมนต์หนึ่ง (child element) จะมีอีลิเมนต์ที่อยู่เหนือขึ้นไปสัมพันธ์กันโดยตรง (parent element) มากกว่าหนึ่งความสัมพันธ์ได้ และแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีลิเมนต์จะอยู่ได้เพียงที่เดียวเท่านั้น แบบจำลองแบบลำดับชั้นจะพบการใช้งานมาในเครื่องเมนเฟรม และมีคอมพิวเตอร์ เช่น ระบบ Information Management System หรือไอเอ็มเอส จากไอบีเอ็ม เป็นต้น

2.7.6 แบบจำลองแบบเครือข่าย (The Network Model)

แบบจำลองแบบเครือข่ายได้รับการพัฒนาขึ้นเมื่อปีค.ศ. 1960 มีหลักการที่คล้ายกับแบบจำลองแบบลำดับชั้น นั่นคือมีการจัดข้อมูลอยู่ในความสัมพันธ์แบบพารেন্ট-ไชลด์ (parent-child) แต่อีลิเมนต์ที่เป็นไชลด์ (child) สามารถมีความสัมพันธ์กับอีลิเมนต์ที่เป็น พารেন্ট (parent) ได้มากกว่าหนึ่งอีลิเมนต์ นั่นคือสามารถมีความสัมพันธ์ของข้อมูลในแบบ n:m ได้นั่นเอง ทำให้แบบจำลองแบบเครือข่ายสามารถแปลงเป็นแบบจำลองแบบลำดับชั้นได้ แต่แบบจำลองแบบลำดับชั้นจะแปลงเป็นแบบจำลองแบบเครือข่ายไม่ได้



รูปที่ 2.25 โครงสร้างฐานข้อมูลแบบเครือข่าย

2.7.7แบบจำลองแบบความสัมพันธ์ (The Relational Model)

แบบจำลองแบบความสัมพันธ์เป็นแบบจำลองแบบที่ได้รับความนิยมสูงสุดในปัจจุบัน โดยระบบฐานข้อมูลส่วนมากจะใช้แบบความจำลองชนิดนี้ในการจัดการข้อมูลที่เกี่ยวข้อง เรียกว่าฐานข้อมูลที่ใช้แบบจำลองแบบสัมพันธ์นี้ว่า ฐานข้อมูลเชิงสัมพันธ์ (RDBMS) จากรูปตัวอย่างใบบันทึกการซื้อสินค้า หากนำมาเก็บข้อมูลโดยใช้แบบจำลองแบบความสัมพันธ์จะเก็บไฟล์ในระบบฐานข้อมูลได้ดังนี้

| ACCOUNT | NAME | ADDR |
|---------|------|------|
| | | |
| | | |
| | | |

(ก) HEAD FILE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| ACCOUNT | ITEM | DESCRIPTION | QUANTITY | ITEM | EXT.COS |
|---------|------|-------------|----------|------|---------|
| | | | | | |
| | | | | | |
| | | | | | |

(ข) ITEM FILE

รูปที่ 2.26 โครงสร้างตารางฐานข้อมูลเชิงสัมพันธ์

จากรูป HEADER FILE เป็นไฟล์ที่เก็บข้อมูลนำเข้าในเอกสารการซื้อขายสินค้าทั้งหมด โดยข้อมูลนำเข้าของเอกสารอันหนึ่ง ก็จะเป็นข้อมูลหนึ่งชุดในไฟล์ ส่วน ITEM FILE เป็นไฟล์ที่เก็บรายละเอียดรายการสินค้าทั้งหมดในเอกสาร โดยสินค้าแต่ละรายการจะเป็นข้อมูลหนึ่งชุด เช่น เอกสารแผ่นหนึ่งมีสินค้าอยู่ 5 รายการ ข้อมูลนำเข้าก็จะเป็นข้อมูลหนึ่งชุดใน HEADER FILE ส่วนรายการสินค้าจะถูกเก็บใน ITEM FILE จำนวน 5 ชุด เป็นต้น หากสังเกตให้ดีจะเห็นว่าทั้ง 2 ไฟล์จะมีฟิลด์ ACCOUNT ที่เหมือนกัน ใช้สำหรับเชื่อมโยง (linkage) ข้อมูลที่สัมพันธ์กัน สมมติตัวอย่างข้อมูลตามโครงสร้างตารางข้างต้นได้ดังนี้

| ACCOUNT | NAME | ADDRESS |
|---------|--------|----------|
| CA154 | Tom | Maryland |
| VF326 | Philip | Chicago |
| CD581 | Mark | Florida |

(ก) HEAD FILE

| ACCOUNT | ITEM | DESCRIPTION | QUANTITY | ITEM COST | EXT COST |
|---------|------|-------------|----------|-----------|----------|
| VF326 | 15 | Keyboard | 1 | 1,000 | 1,000 |
| CA164 | 47 | Diskette | 3 | 250 | 750 |
| CA164 | 33 | Mouse | 2 | 450 | 900 |
| CD681 | 47 | Diskette | 4 | 250 | 1,000 |
| VF326 | 24 | UPS | 2 | 2,500 | 5,000 |

(ข) ITEM FILE

รูปที่ 2.27 ตัวอย่างข้อมูลจากโครงสร้างตารางฐานข้อมูลเชิงสัมพันธ์

ข้อมูลในไฟล์ ITEM FILE ทำให้ข้อมูลในไฟล์ทั้ง 2 ไม่ต้องจัดลำดับ เพราะการทำงานในระบบฐานข้อมูลเชิงสัมพันธ์จะไม่ขึ้นกับลำดับของข้อมูลที่เก็บ แต่ผู้ใช้อาจจัดลำดับของข้อมูลเพื่อให้ข้อมูลดูเป็นระเบียบมากขึ้นก็ได้ ข้อจำกัดหรือข้อกำหนดของการเก็บข้อมูลโดยใช้แบบจำลองความสัมพันธ์คือในไฟล์หนึ่ง ๆ จะมีข้อมูลหรือเรคคอร์ดที่ซ้ำกันไม่ได้ นอกจากนี้ ผู้ใช้อาจเพิ่มไฟล์เพื่อ การสร้างรายการสินค้าขึ้นมารายการหนึ่งทำได้โดยเลือกข้อมูลจาก HEADER FILE แล้วนำรายการนั้นไปเลือกเก็บข้อมูลอย่างอื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็ได้ เช่น ข้อมูลการออกใบเสร็จให้ลูกค้า โดยการเพิ่มข้อมูลใหม่สามารถทำได้โดยเพิ่มไฟล์ใหม่ขึ้นมา และไฟล์ใหม่นี้จะต้องมีส่วนที่เชื่อมโยงกับไฟล์อื่นด้วย ซึ่งในที่นี้คือฟิลด์ ACCOUNT ดังรูป

| ACCOUNT | BILLING_DATE | BILLING_AMOUNT |
|---------|--------------|----------------|
| VF326 | 01/03/97 | 1,000 |
| CA164 | 01/03/97 | 750 |
| CA164 | 01/03/97 | 900 |
| CD681 | 01/05/97 | 1,000 |
| VF326 | 01/11/97 | 5,000 |

รูปที่ 2.28 ตัวอย่างการเพิ่มข้อมูลในตารางฐานข้อมูลเชิงสัมพันธ์

ไฟล์ในระบบการจัดการฐานข้อมูลแบบความสัมพันธ์เป็นไฟล์ที่เข้าใจความหมายได้ง่ายระหว่างไฟล์ต่าง ๆ มีข้อมูลที่ซ้ำซ้อนกันน้อยมาก ทำให้ประหยัดเนื้อที่ของหน่วยเก็บข้อมูล รวมทั้งสามารถเพิ่มหรือลดข้อมูลได้ง่าย ในระบบการจัดการฐานข้อมูลประเภทนี้มักจะไม่มีการจัดโครงสร้างของไฟล์ใหม่ จะมีก็เป็นการสร้างไฟล์ใหม่ขึ้นมาเพิ่มเท่านั้น ในกรณีที่มีการเปลี่ยนแปลงความต้องการขององค์กรก็จะเป็นการเพิ่มหรือลบไฟล์ หรือเพิ่มหรือลบฟิลด์บางฟิลด์เท่านั้น หรืออาจรวมไฟล์ 2 ไฟล์เข้าด้วยกันก็ได้ แต่การรวมไฟล์ขนาดใหญ่ 2 ไฟล์จะทำให้เกิดไฟล์ที่มีขนาดใหญ่มากและต้องใช้เวลามากในการทำงานนาน

2.7.8 ฐานข้อมูลแบบรีเลชันแนล(Relational Database Theory)

ในปัจจุบันการจัดโครงสร้างข้อมูลให้เป็นแบบฐานข้อมูลกำลังเป็นที่นิยม เกือบทุกหน่วยงานที่มีการใช้ระบบสารสนเทศจะจัดทำข้อมูลให้เป็นแบบฐานข้อมูล เนื่องจากปริมาณข้อมูลมีมากถ้าจัดข้อมูลเป็นแบบเพิ่มข้อมูลจะทำให้มีเพิ่มข้อมูลเป็นจำนวนมาก ซึ่งจะทำให้เกิดข้อมูลที่ซ้ำซ้อนกันได้ ข้อมูลที่ซ้ำซ้อนนี้จะก่อให้เกิดปัญหามากมาย

2.7.8.1 ความหมายของระบบฐานข้อมูล

ฐานข้อมูล (database) หมายถึง กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้ โดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในเพิ่มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ เพิ่มข้อมูล นั่นก็คือการเก็บข้อมูลในฐานข้อมูลนั้นเราอาจจะเก็บทั้งฐานข้อมูล โดยใช้เพิ่มข้อมูลเพียงเพิ่มข้อมูลเดียวกันได้ หรือจะเก็บไว้ในหลาย ๆ เพิ่มข้อมูล ที่สำคัญคือจะต้องสร้างความสัมพันธ์ระหว่างระเบียบและเรียกใช้ความสัมพันธ์นั้นได้ มีการกำจัดความซ้ำซ้อนของข้อมูลออกและเก็บเพิ่มข้อมูลเหล่านี้ไว้ที่ศูนย์กลาง เพื่อที่จะนำข้อมูลเหล่านี้มาใช้ร่วมกัน ควบคุมดูแลรักษาเมื่อผู้ต้องการใช้งานและผู้มีสิทธิ์จะใช้ข้อมูลนั้นสามารถดึงข้อมูลที่ต้องการออกไปใช้ได้ ข้อมูลบางส่วนอาจใช้ร่วมกับผู้อื่นได้ แต่บางส่วนของผู้มีสิทธิ์เท่านั้นจึงจะสามารถใช้ได้ โดยทั่วไปองค์กรต่าง ๆ จะสร้างฐานข้อมูลไว้เพื่อเก็บข้อมูลต่าง ๆ ของตัวองค์กร โดยเฉพาะอย่างยิ่งข้อมูลในเชิงธุรกิจ เช่น ข้อมูลของลูกค้า ข้อมูลของสินค้า ข้อมูลของลูกจ้าง และการจ้างงาน เป็นต้น การควบคุมดูแลการใช้ฐานข้อมูลนั้น เป็นเรื่องที่อยู่ยากกว่าการใช้เพิ่มข้อมูลมาก เพราะเราจะต้องตัดสินใจว่าโครงสร้างในการจัดเก็บข้อมูลควรจะเป็นเช่นไร การเขียนโปรแกรมเพื่อสร้าง

และเรียกใช้ข้อมูลจากโครงสร้างเหล่านี้ ถ้าโปรแกรมเหล่านี้เกิดทำงานผิดพลาดขึ้นมา ก็จะทำให้เกิดความเสียหายต่อโครงสร้างของข้อมูลทั้งหมดได้ เพื่อเป็นการลดภาระการทำงานของผู้ใช้ จึงได้มีส่วนของฮาร์ดแวร์ และโปรแกรมต่าง ๆ ที่สามารถเข้าถึงและจัดการข้อมูลในฐานะข้อมูลนั้น เรียกว่า ระบบจัดการฐานข้อมูล หรือ DBMS (data base management system) ระบบจัดการฐานข้อมูล คือ ซอฟต์แวร์ที่เปรียบเสมือนสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้งานข้อมูล ซึ่งมีหน้าที่ช่วยให้ผู้ใช้เข้าถึงข้อมูลได้ง่ายสะดวกและมีประสิทธิภาพ การเข้าถึงข้อมูลของผู้ใช้อาจเป็นการสร้างฐานข้อมูล การแก้ไขฐานข้อมูล หรือการตั้งคำถามเพื่อให้ข้อมูลมา โดยผู้ใช้ไม่จำเป็นต้องรับรู้เกี่ยวกับรายละเอียดภายในโครงสร้างของฐานข้อมูล เปรียบเสมือนเป็นสื่อกลางระหว่างผู้ใช้และโปรแกรมต่าง ๆ ที่เกี่ยวข้องกับการใช้งานข้อมูล

การจัดข้อมูลให้เป็นระบบฐานข้อมูลทำให้ข้อมูลมีส่วนคิดว่าการเก็บข้อมูลในรูปของแฟ้มข้อมูล เพราะการจัดเก็บข้อมูลในระบบฐานข้อมูล จะมีส่วนที่สำคัญกว่าการจัดเก็บข้อมูลในรูปของแฟ้มข้อมูลคั้งนี้คือการเก็บข้อมูลที่ซ้ำซ้อน ข้อมูลบางชุดที่อยู่ในรูปของแฟ้มข้อมูลอาจมีปรากฏอยู่หลาย ๆ แห่ง เพราะมีผู้ใช้ข้อมูลชุดนี้หลายคน เมื่อใช้ระบบฐานข้อมูลแล้วจะช่วยให้ความซ้ำซ้อนของข้อมูลลดน้อยลง เช่น ข้อมูลอยู่ในแฟ้มข้อมูลของผู้ใช้หลายคน ผู้ใช้แต่ละคนจะมีแฟ้มข้อมูลเป็นของตนเอง ระบบฐานข้อมูลจะลดการซ้ำซ้อนของข้อมูลเหล่านี้ให้มากที่สุด โดยจัดเก็บในฐานข้อมูลไว้ที่เดียวกัน ผู้ใช้ทุกคนที่ต้องการใช้ข้อมูลชุดนี้จะใช้โดยผ่านระบบฐานข้อมูล ทำให้ไม่เปลืองเนื้อที่ในการเก็บข้อมูลและลดความซ้ำซ้อนลงได้

2.7.8.2 ประโยชน์ของระบบจัดการฐานข้อมูล

ในปัจจุบันองค์กรส่วนใหญ่หันมาให้ความสนใจกับระบบฐานข้อมูลกันมาก เนื่องจากระบบฐานข้อมูลมีประโยชน์ดังต่อไปนี้

1.ลดความซ้ำซ้อนของข้อมูล

เนื่องจากการใช้งานระบบฐานข้อมูลนั้นต้องมีการออกแบบฐานข้อมูลเพื่อให้มีความซ้ำซ้อนของข้อมูลน้อยที่สุด จุดประสงค์หลักของการออกแบบฐานข้อมูลเพื่อการลดความซ้ำซ้อน สาเหตุที่ต้องลดความซ้ำซ้อน เนื่องจากความยากในการปรับปรุงข้อมูล กล่าวคือถ้าเก็บข้อมูลซ้ำซ้อนกันหลายแห่ง เมื่อมีการปรับปรุงข้อมูลแล้วปรับปรุงข้อมูลไม่ครบทำให้ข้อมูลเกิดความขัดแย้งกันของข้อมูลตามมา และยังเปลืองเนื้อที่การจัดเก็บข้อมูลด้วย เนื่องจากข้อมูลชุดเดียวกันจัดเก็บซ้ำกันหลายแห่งนั่นเอง

ถึงแม้ว่าความซ้ำซ้อนช่วยให้การรายงานและตอบคำถามได้เร็วขึ้น แต่ข้อมูลจะเกิดความขัดแย้งกัน ในกรณีที่ต้องมีการปรับปรุงข้อมูลหลายแห่ง การออกรายงานจะทำได้เร็วเท่าใดนั้นจึงไม่มีความหมายแต่อย่างใด และเหตุผลที่สำคัญอีกประการหนึ่งคือปัญหาเรื่องความขัดแย้งกันของข้อมูลแก้ไขไม่ได้ด้วยฮาร์ดแวร์ ขณะที่การออกรายงานใช้นั้นใช้ความสามารถของฮาร์ดแวร์ช่วยได้

2.รักษาความถูกต้องของข้อมูล

เนื่องจากระบบจัดการฐานข้อมูลสามารถตรวจสอบกฎบังคับความถูกต้องของข้อมูลให้ได้ โดยนำกฎเหล่านี้มาไว้ที่ฐานข้อมูล ซึ่งถือเป็นหน้าที่ของระบบจัดการฐานข้อมูลที่จะจัดการเรื่องความถูกต้องของข้อมูลให้แทน แต่ถ้าเป็นระบบแฟ้มข้อมูลผู้พัฒนาโปรแกรมต้องเขียนโปรแกรมเพื่อควบคุมกฎระเบียบต่างๆ (data integrity) เองทั้งหมด ถ้าเขียนโปรแกรมครอบคลุมกฎระเบียบใดไม่ครบหรือขาดหาย

ไปบางกฎอาจทำให้ข้อมูลผิดพลาดได้ และยังช่วยลดค่าใช้จ่ายในการบำรุงรักษาและพัฒนาโปรแกรมด้วย เนื่องจากระบบจัดการฐานข้อมูลจัดการให้มันเอง เนื่องจากระบบจัดการฐานข้อมูลสามารถรองรับการใช้งานของผู้ใช้หลายคนพร้อมกันได้ ดังนั้นความคงสภาพและความถูกต้องของข้อมูลจึงมีความสำคัญมาก และต้องควบคุมให้ดีเนื่องจากผู้ใช้อาจเปลี่ยนแปลงแก้ไขข้อมูลได้ ซึ่งจะทำให้เกิดความผิดพลาดกระทบต่อการใช้ข้อมูลของผู้ใช้อื่นทั้งหมดได้ ดังนั้นประโยชน์ของระบบฐานข้อมูลในเรื่องนี้จึงมีความสำคัญมาก

3. มีความเป็นอิสระของข้อมูล

เนื่องจากมีแนวคิดที่ว่าทำอะไรให้โปรแกรมเป็นอิสระจากการเปลี่ยนแปลงโครงสร้างข้อมูล ในปัจจุบันนี้ถ้าไม่ใช้ระบบฐานข้อมูลการแก้ไขโครงสร้างข้อมูลจะกระทบถึงโปรแกรมด้วย เนื่องจากในการเรียกใช้ข้อมูลที่เก็บอยู่ในระบบแฟ้มข้อมูลนั้น ต้องใช้โปรแกรมที่เขียนขึ้นเพื่อเรียกใช้ข้อมูลในแฟ้มข้อมูลนั้น โดยเฉพาะ เช่น เมื่อต้องการรายชื่อพนักงานที่มีเงินเดือนมากกว่า 100,000 บาทต่อเดือน โปรแกรมเมอร์ต้องเขียน โปรแกรมเพื่ออ่านข้อมูลจากแฟ้มข้อมูลพนักงานและพิมพ์รายงานที่แสดงเฉพาะข้อมูลที่ตรงตามเงื่อนไขที่กำหนด กรณีที่มีการเปลี่ยนแปลงโครงสร้างของแฟ้มข้อมูลข้อมูลเช่น ให้มีดัชนี (index) ตามชื่อพนักงานแทนรหัสพนักงาน ส่งผลให้รายงานที่แสดงรายชื่อพนักงานที่มีเงินเดือนมากกว่า 100,000 บาทต่อเดือนซึ่งแต่เดิมกำหนดให้เรียงตามรหัสพนักงานนั้น ไม่สามารถพิมพ์ได้ ทำให้ต้องมีการแก้ไขโปรแกรมตามโครงสร้างดัชนี (index) ที่เปลี่ยนแปลงไป ลักษณะแบบนี้เรียกว่าข้อมูลและ โปรแกรมไม่เป็นอิสระต่อกัน

สำหรับระบบฐานข้อมูลนั้นข้อมูลภายในฐานข้อมูลจะเป็นอิสระจากโปรแกรมที่เรียกใช้ (data independence) สามารถแก้ไขโครงสร้างทางกายภาพของข้อมูลได้ โดยไม่กระทบต่อโปรแกรมที่เรียกใช้ข้อมูลจากฐานข้อมูล เนื่องจากระบบฐานข้อมูลมีระบบจัดการฐานข้อมูลทำหน้าที่แปลงรูป (mapping) ให้เป็นไปตามรูปแบบที่ผู้ใช้ต้องการ เนื่องจากในระบบแฟ้มข้อมูลนั้น ไม่มีความเป็นอิสระของข้อมูล ดังนั้นระบบฐานข้อมูลได้ถูกพัฒนาขึ้นมาเพื่อแก้ปัญหาด้านความเป็นอิสระของข้อมูล นั่นคือระบบฐานข้อมูลมีการทำงาน ไม่ขึ้นกับรูปแบบของฮาร์ดแวร์ที่นำมาใช้กับระบบฐานข้อมูลและไม่ขึ้นกับโครงสร้างทางกายภาพของข้อมูล และมีการใช้ภาษาสอบถามในการติดต่อกับข้อมูลภายในฐานข้อมูลแทนคำสั่งของภาษาคอมพิวเตอร์ในยุคที่ 3 ทำให้ผู้ใช้เรียกใช้ข้อมูลจากฐานข้อมูลโดยไม่จำเป็นต้องทราบรูปแบบการจัดเก็บข้อมูล ประเภทหรือขนาดของข้อมูลนั้นๆ

4. มีความปลอดภัยของข้อมูลสูง

ถ้าหากทุกคนสามารถเรียกดูและเปลี่ยนแปลงข้อมูลในฐานข้อมูลทั้งหมดได้ อาจก่อให้เกิดความเสียหายต่อข้อมูลได้ และข้อมูลบางส่วนอาจเป็นข้อมูลที่ไม่อาจเปิดเผยได้หรือเป็นข้อมูลเฉพาะของผู้บริหาร หากไม่มีการจัดการด้านความปลอดภัยของข้อมูล ฐานข้อมูลก็จะไม่สามารถใช้เก็บข้อมูลบางส่วนได้

ระบบฐานข้อมูลส่วนใหญ่จะมีการรักษาความปลอดภัยของข้อมูล ดังนี้

- มีรหัสผู้ใช้ (user) และรหัสผ่าน (password) ในการใช้งานฐานข้อมูลสำหรับผู้ใช้แต่ละคน ระบบฐานข้อมูลมีระบบการสอบถามชื่อหรือรหัสผ่านของผู้เข้ามาใช้ระบบงานเพื่อให้ทำงาน

ในส่วนที่เกี่ยวข้องเท่านั้น โดยป้องกันไม่ให้ผู้ที่ไม่ได้รับอนุญาตเข้ามาเห็นหรือแก้ไขข้อมูลในส่วนที่ต้องการปกป้องไว้

- ในระบบฐานข้อมูลสามารถสร้างและจัดการตารางข้อมูลทั้งหมดในฐานข้อมูล ทั้งการเพิ่มผู้ใช้ ระบุการใช้งานของผู้ใช้ อนุญาตให้ผู้ใช้สามารถเรียกดู เพิ่มเติม ลบและแก้ไขข้อมูล หรือบางส่วนของข้อมูลได้ในตารางที่ได้รับอนุญาต) ระบบฐานข้อมูลสามารถกำหนดสิทธิการมองเห็น และการใช้งานของผู้ใช้ต่างๆ ตามระดับสิทธิและอำนาจการใช้งานข้อมูลนั้นๆ
- ในระบบฐานข้อมูล (DBA) สามารถใช้วิว (view) เพื่อประโยชน์ในการรักษาความปลอดภัยของข้อมูลได้เป็นอย่างดี โดยการสร้างวิวที่เสมือนเป็นตารางของผู้ใช้จริงๆ และข้อมูลที่ปรากฏในวิวจะเป็นข้อมูลที่เกี่ยวข้องกับงานของผู้ใช้เท่านั้น ซึ่งจะ ไม่กระทบกับข้อมูลจริงในฐานข้อมูล
- ระบบฐานข้อมูลจะไม่ยอมให้โปรแกรมใดๆ เข้าถึงข้อมูลในระดับกายภาพ (physical) โดยไม่ผ่าน ระบบการจัดการฐานข้อมูล และถ้าระบบเกิดความเสียหายขึ้นระบบจัดการฐานข้อมูลรับรองได้ว่าข้อมูลที่ยืนยันการทำงานสำเร็จ (commit) แล้วจะไม่สูญหาย และถ้ากลุ่มงานที่ยังไม่สำเร็จ (rollback) นั้นระบบจัดการฐานข้อมูลรับรองได้ว่าข้อมูลเดิมก่อนการทำงานของกลุ่มงานยังไม่สูญหาย
- มีการเข้ารหัสและถอดรหัส (encryption/decryption) เพื่อปกปิดข้อมูลแก่ผู้ที่ไม่เกี่ยวข้อง เช่น มีการเข้ารหัสข้อมูลรหัสผ่าน

5. ใช้ข้อมูลร่วมกันโดยมีการควบคุมจากศูนย์กลาง

มีการควบคุมการใช้ข้อมูลในฐานข้อมูลจากศูนย์กลาง ระบบฐานข้อมูลสามารถรองรับการทำงานของผู้ใช้หลายคนได้ กล่าวคือระบบฐานข้อมูลจะต้องควบคุมลำดับการทำงานให้เป็นไปอย่างถูกต้อง เช่น ขณะที่ผู้ใช้คนหนึ่งกำลังแก้ไขข้อมูลส่วนหนึ่งยังไม่เสร็จ ก็จะไม่อนุญาตให้ผู้ใช้คนอื่นเข้ามาเปลี่ยนแปลงแก้ไขข้อมูลนั้นได้ เนื่องจากข้อมูลที่เข้ามายังระบบฐานข้อมูลจะถูกนำเข้าไปโดยระบบงานระดับปฏิบัติการตามหน่วยงานย่อยขององค์กร ซึ่งในแต่ละหน่วยงานจะมีสิทธิในการจัดการข้อมูลไม่เท่ากัน ระบบฐานข้อมูลจะทำการจัดการว่าหน่วยงานใดใช้ระบบจัดการฐานข้อมูลในระดับใดบ้าง ใครเป็นผู้นำข้อมูลเข้า ใครมีสิทธิแก้ไขข้อมูล และใครมีสิทธิเพียงเรียกใช้ข้อมูล เพื่อที่จะให้สิทธิที่ถูกต้องบนตารางที่สมควรให้ใช้

ระบบฐานข้อมูลจะบอกรายละเอียดว่าข้อมูลใดถูกจัดเก็บไว้ในตารางชื่ออะไร เมื่อมีคำถามจากผู้บริหารจะสามารถหาข้อมูลเพื่อตอบคำถามได้ทันทีโดยใช้ภาษาฐานข้อมูลที่มีประสิทธิภาพมาก คือ SQL ซึ่งสามารถตอบคำถามที่เกิดขึ้น ในขณะที่ขณะหนึ่งที่เกี่ยวข้องกับฐานข้อมูลได้ทันที โดยไม่จำเป็นต้องเขียนภาษาโปรแกรมอย่างเช่น โคบอล ซี หรือ ปาสคาล ซึ่งเสียเวลานานมากจนอาจไม่ทันต่อความต้องการใช้ข้อมูลเพื่อการตัดสินใจของผู้บริหาร

เนื่องจากระบบจัดการฐานข้อมูลนั้นสามารถจัดการให้ผู้ใช้ทำงานพร้อมๆ กันได้หลายคน ดังนั้นโปรแกรมที่พัฒนาภายใต้การดูแลของระบบจัดการฐานข้อมูลจะสามารถใช้ข้อมูลร่วมกันในฐานข้อมูลเดียวกันระบบฐานข้อมูลจะแบ่งเบาภาระในการพัฒนาระบบงานถ้าการพัฒนากระบวนการไม่ใช้ระบบฐานข้อมูล (ใช้ระบบแฟ้มข้อมูล) ผู้พัฒนาโปรแกรมจะต้องจัดการสิ่งเหล่านี้เองทั้งหมด นั่นคือระบบฐานข้อมูล

ทำให้การใช้ข้อมูลเกิดความเป็นอิสระระหว่างการจัดเก็บข้อมูลและการประยุกต์ใช้ เพราะส่วนของการจัดเก็บข้อมูลจริงถูกซ่อนจากการใช้งานจริงนั่นเอง

2.7.8.3 สถาปัตยกรรมของระบบฐานข้อมูล

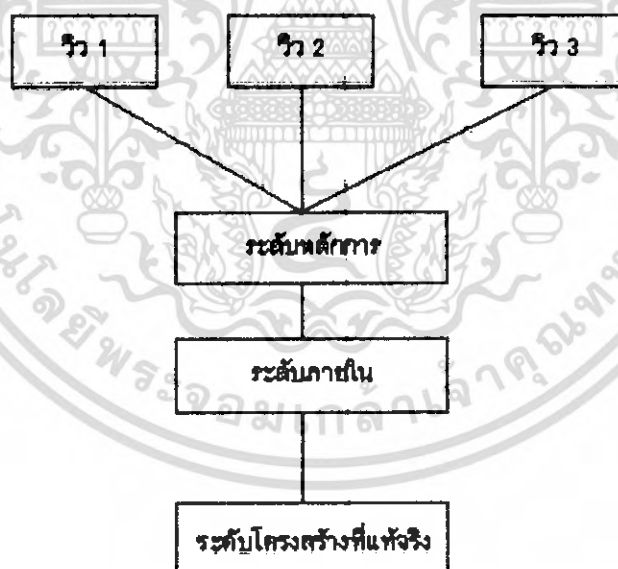
ระบบฐานข้อมูล เป็นการนำข้อมูลในองค์การที่มีความเกี่ยวข้องกันมารวมไว้อย่างเป็นระบบในที่เดียวกัน โดยที่ผู้ใช้ฐานข้อมูลจะมองข้อมูลนี้ในแง่มุมหรือวิวที่แตกต่างกันไปตามจุดประสงค์ของการประยุกต์ใช้งาน โดยผู้ใช้ไม่จำเป็นต้องสนใจว่าลักษณะการจัดเก็บข้อมูลโดยแท้จริงแล้วเป็นเช่นไร โดยระบบฐานข้อมูลจะทำการซ่อนรายละเอียดไว้ โดยจัดแบ่งระดับของข้อมูลออกเป็นระดับชั้น

ระดับชั้นของข้อมูลถูกพัฒนาขึ้น โดย The Standards Planning and Requirements Committee (SPARC) ของ American National Standards Institute (ANSI) จะถูกแบ่งออกเป็น 4 ระดับ

1.1 ระดับภายนอก (external level) เป็นระดับที่อยู่สูงสุด โดยผู้ใช้สามารถมองเห็นงานของผู้ใช้แต่ละคน และสามารถเรียกใช้ฐานข้อมูลได้ในระดับนี้

1.2 ระดับหลักการ (conceptual level) เป็นระดับที่อยู่ถัดขึ้นมาได้แก่ ระดับของการมองเห็นข้อมูลของระบบฐานข้อมูลรวมทั้งกฎเกณฑ์ต่าง ๆ เกี่ยวกับข้อมูลและผู้ที่มิสิทธิจะใช้ ข้อมูลในระดับนี้จะถูกใช้โดยโปรแกรมเมอร์หรือผู้เขียนโปรแกรม

1.3 ระดับภายใน (internal level) เป็นระดับของการจัดความสัมพันธ์ระหว่างเพิ่มข้อมูลของระบบฐานข้อมูลและการเชื่อมโยงแต่ละเพิ่มข้อมูล ข้อมูลในระดับนี้จะถูกใช้โดยผู้จัดการฐานข้อมูลและผู้เขียนโปรแกรมระบบ (system programmer)



ภาพที่ 2.29 แสดงระดับชั้นของข้อมูล

1.4 ระดับโครงสร้างแท้จริง (physical organization level) เป็นระดับที่ต่ำที่สุดอันได้แก่ กลุ่มของเพิ่มข้อมูลที่จัดเก็บไว้เป็นเพิ่มข้อมูลจริงและโครงสร้างของเพิ่มข้อมูล

ประโยชน์ของการแบ่งระดับชั้นนั้นเพื่อให้ข้อมูลเป็นอิสระต่อกัน ความเป็นอิสระของข้อมูลคือ การที่ผู้ใช้ไม่ต้องมาคอยแก้ไขโปรแกรมที่ใช้งานในทุก ๆ ครั้งที่เกิดการเปลี่ยนแปลงแก้ไขฐานข้อมูล ระบบจัดการฐานข้อมูลจะทำหน้าที่เชื่อมโยงข้อมูลระหว่างแต่ละระดับ

เนื่องจากฐานข้อมูลมีลักษณะเด่นที่เหนือกว่าระบบแฟ้มข้อมูล คือความเป็นอิสระของข้อมูล การที่ผู้ใช้ไม่ต้องแก้ไขโปรแกรมที่ใช้งานเมื่อมีการเปลี่ยนแปลงข้อมูลระดับแนวคิดหรือระดับภายใน โดยเป็นหน้าที่ของระบบจัดการฐานข้อมูลหรือคิวบีเอ็มเอสในการเชื่อมข้อมูลระดับภายนอกและระดับแนวคิด และเชื่อมข้อมูลระดับแนวคิดกับระดับภายใน ซึ่งการเชื่อมนี้เกี่ยวข้องกับความเป็นอิสระของข้อมูล ทำให้ผู้ใช้ไม่จำเป็นต้องรับทราบเกี่ยวกับข้อมูลส่วนอื่นๆ ที่คนไม่ได้ใช้ ผู้ใช้มองเห็นโครงสร้างข้อมูลระดับภายนอก เหมือนเดิมและสามารถใช้งานได้ตามปกติ กล่าวคือข้อมูลภายในฐานข้อมูลเป็นอิสระจากโปรแกรมที่เรียกใช้ เพื่อที่สามารถแก้ไขโครงสร้างทางกายภาพของข้อมูลได้ โดยไม่กระทบต่อโปรแกรมที่เรียกใช้ฐานข้อมูลนั้น ความเป็นอิสระของข้อมูลแบ่งออกเป็น 2 ลักษณะ คือ

1. ความเป็นอิสระของข้อมูลเชิงตรรกะ (logical data independence) ตรรกะ (logical) ในความหมายที่ใช้กับระบบฐานข้อมูลจะหมายถึงมุมมองของผู้ใช้ต่อข้อมูลนั้น โดยขึ้นอยู่กับผู้ใช้ว่าทำงานเกี่ยวข้องกับข้อมูลลักษณะใด ตัวอย่าง ถ้ามีคำถามว่าแฟ้มข้อมูลคืออะไร ถ้าถามบุคคลในวงการคอมพิวเตอร์ คำตอบที่ได้คือที่เก็บรวบรวมเรคคอร์ด เพราะว่ามันคือส่วนใหญ่อะไรคือโปรแกรมเมอร์ผู้เขียน โปรแกรม ซึ่งในมุมมองของผู้เขียนโปรแกรมนั้นจะมองแฟ้มข้อมูลเป็นเรคคอร์ด นั่นคือขณะที่โปรแกรมเมอร์ใช้คำสั่งอ่านข้อมูล (read) 1 คำสั่งจะได้ข้อมูล 1 เรคคอร์ด และเมื่อใช้คำสั่งเขียน (write) 1 คำสั่งจะบันทึกข้อมูล 1 เรคคอร์ด นั่นคือในมุมมองของโปรแกรมเมอร์จะเห็นแฟ้มข้อมูลเป็นเรคคอร์ด แต่ถ้าเราถามเจ้าหน้าที่สารบรรณว่าแฟ้มข้อมูลคืออะไร เจ้าหน้าที่สารบรรณจะตอบว่าคือที่เก็บรวบรวมตัวอักษรหรือข้อความ เพราะว่าเจ้าหน้าที่ดังกล่าวมองแฟ้มข้อมูลเป็นที่เก็บตัวอักษร เนื่องจากใช้โปรแกรมไมโครซอฟต์เวิร์ดจัดการกับข้อมูล และเก็บข้อมูลซึ่งเป็นตัวอักษรและข้อความต่างๆ เป็นแฟ้มข้อมูลนั่นเอง นั่นคือบุคคลเหล่านั้นทั้งโปรแกรมเมอร์และเจ้าหน้าที่สารบรรณมีมุมมองต่อแฟ้มข้อมูลต่างกันขึ้นอยู่กับลักษณะงานที่เกี่ยวข้องด้วย ซึ่งแฟ้มข้อมูลที่ผู้ใช้เห็นนี้เรียกว่า แฟ้มข้อมูลเชิงตรรกะ (logical file) ดังนั้นอาจกล่าวได้ว่า ความเป็นอิสระของข้อมูลเชิงตรรกะ (logical data independence) หมายถึง เมื่อมีการเปลี่ยนแปลงแก้ไขโครงสร้างข้อมูลในระดับแนวคิด จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในระดับภายนอกที่ผู้ใช้ใช้งานอยู่ เช่น เมื่อมีการเปลี่ยนแปลงขนาดของแอตทริบิวต์ในตารางฐานข้อมูลในระดับแนวคิด ก็ไม่จำเป็นต้องไปแก้ไขโปรแกรมประยุกต์ที่ผู้ใช้เขียนขึ้นในระดับภายนอกที่มีการเรียกใช้แอตทริบิวต์นั้นในการทำงานกับฐานข้อมูลต้องรู้ว่าสิ่งที่เราทำงานอยู่ด้วยนั้นเกี่ยวข้องกับระดับกายภาพหรือระดับตรรกะ

2. ความเป็นอิสระของข้อมูลเชิงกายภาพ (physical data independence) กายภาพ (physical) ในความหมายของระบบผู้ใช้จะหมายถึงมุมมองของระบบปฏิบัติการ (Operating System; OS) ต่อข้อมูลนั้น จากคำถามข้างต้นถามว่าแฟ้มข้อมูลคืออะไร คำตอบที่ได้ในที่นี้คือที่เก็บรวบรวมบิตโดยนำรูปแบบของบิต (bit pattern) มาเรียงต่อกันเป็นสาย ซึ่งเป็นคำตอบในมุมมองของระบบปฏิบัติการ จะเห็นว่าไม่เกี่ยวข้องกับเรคคอร์ดหรือตัวอักษร ซึ่งแฟ้มข้อมูลในมุมมองของระบบปฏิบัติการนี้เรียกว่า แฟ้มข้อมูลเชิงกายภาพ (physical file) ความเป็นอิสระของข้อมูลเชิงกายภาพ (physical data independence) หมายถึง เมื่อมีการ

เปลี่ยนแปลงแก้ไขโครงสร้างข้อมูลในระดับภายใน จะไม่มีผลกระทบต่อโครงสร้างข้อมูลในระดับแนวคิด หรือระดับภายนอก เช่น เมื่อมีการเปลี่ยนวิธีการจัดเก็บข้อมูลจากแบบเรียงลำดับ (sequential) ไปเป็นแบบดัชนี (indexed) ในระดับภายใน ในระดับแนวคิดนั้นจะไม่มีผลกระทบต่อ การเปลี่ยนแปลงดังกล่าว หรือโปรแกรมประยุกต์ที่เขียนในระดับภายนอกก็ไม่จำเป็นต้องแก้ไข โปรแกรมตามวิธีการจัดเก็บที่เปลี่ยนแปลงไปสรุปได้ว่าเพิ่มข้อมูลที่กล่าวถึงนั้นคือสิ่งเดียวกันแต่เมื่อมองจากค่ามุมมองจะมองเห็นต่างกัน ซึ่งในมุมมองของผู้ใช้นั้นเป็นมุมมองเชิงตรรกะ ขณะที่มุมมองของระบบปฏิบัติการเป็นมุมมองเชิงกายภาพ การสร้างฐานข้อมูลขึ้นใช้งานในองค์กรหรือหน่วยงานต่างๆ ก็จำเป็นต้องดำเนินการตามขั้นตอนที่เหมาะสม และต้องมีวิธีการจัดการข้อมูล โดยหลักการสร้างฐานข้อมูลจำเป็นจะต้องออกแบบฐานข้อมูลเป็นสองระยะหรือสองขั้นควบคู่กัน ขั้นแรกก็คือการออกแบบเชิงแนวคิด (conceptual design) หรือเชิงตรรกะ (logical design) และขั้นที่สองก็คือการออกแบบเชิงกายภาพ (physical design)

2.1 การออกแบบเชิงตรรกะนั้นในด้านการจัดกลุ่มข้อมูลในฐานข้อมูลให้เป็นหมวดหมู่ หรือ เป็นตารางที่เหมาะสม การออกแบบเริ่มต้นด้วยการพิจารณาว่าหน่วยงานจะต้องใช้ข้อมูลอะไรบ้าง ข้อมูลเหล่านั้นมีความสัมพันธ์กันอย่างไรบ้าง จะจัดกลุ่มข้อมูลอย่างไรจึงจะเหมาะสมและไม่เกิดความซ้ำซ้อน การพิจารณาการจัดกลุ่มนี้จะต้องคำนึงถึงลักษณะของประเภทฐานข้อมูลที่จะจัดทำขึ้นด้วย

2.2 การออกแบบเชิงกายภาพ เน้นในด้านการกำหนดว่าข้อมูลแต่ละรายการหรือตารางข้อมูลต่างๆ จะจัดเก็บลงในสื่อข้อมูลเช่นจานแม่เหล็กได้อย่างไร มีการกำหนดว่าข้อมูลแต่ละรายการเป็นข้อมูลประเภทอักขระ จำนวน หรือประเภทอื่นๆ และต้องใช้เนื้อที่ในการเก็บมากน้อยเท่าใด การออกแบบฐานข้อมูลในส่วนนี้จำเป็นจะต้องใช้ผู้เชี่ยวชาญที่ศึกษาค้นฐานข้อมูลมาโดยตรง

ฐานข้อมูลเป็นงานประยุกต์คอมพิวเตอร์ที่มีความสำคัญเป็นอย่างยิ่งในยุคปัจจุบัน อาจกล่าวได้ว่างานประยุกต์คอมพิวเตอร์ทุกงานล้วนต้องสร้างขึ้นบนฐานข้อมูลแทบทั้งสิ้น ดังนั้นการศึกษาทำความเข้าใจเรื่องของฐานข้อมูลจึงเป็นเรื่องจำเป็น ยิ่งหากได้ศึกษาจนถึงขั้นออกแบบและใช้งานได้จริงแล้วยิ่งจะเป็นประโยชน์มากขึ้นเป็นทวีคูณ

โครงสร้างของสถาปัตยกรรมของฐานข้อมูล 3 ระดับนั้น แต่ละระดับจะมี DBMS ทำหน้าที่ในการแปลงรูประดับข้อมูลจากระดับหนึ่งไปยังอีกระดับหนึ่ง ได้แก่ การแปลงรูประหว่างระดับภายนอกกับระดับแนวคิด และระหว่างระดับแนวคิดกับระดับภายใน

การถ่ายทอข้อมูลมุมมองจากสถาปัตยกรรมในระดับที่สูงกว่าไปยังระดับที่ต่ำกว่า เรียกว่า การแปลงรูป (mapping) การแปลงรูปแบ่งเป็น 2 ลักษณะ คือ หนึ่งการแปลงรูประหว่างระดับภายนอกกับระดับแนวคิด และสองการแปลงรูประหว่างระดับแนวคิดกับระดับภายใน

3..1 การแปลงรูประหว่างระดับภายนอกกับระดับแนวคิด (external/conceptual mapping) เป็นการกำหนดความสอดคล้องกันระหว่างมุมมองในระดับภายนอกและระดับแนวคิดที่เรียกว่า ความเป็นอิสระของข้อมูลเชิงตรรกะ(logical data independence) โดยถ่ายทอข้อมูลที่มีค่อข้อมูลจากสถาปัตยกรรมในระดับภายนอกไปยังสถาปัตยกรรมในระดับแนวคิด เพื่อให้ผู้ใช้ฐานข้อมูลสามารถมีมุมมองข้อมูลที่แตกต่างกันได้ในระดับแนวคิดนั้นอาจมีการเปลี่ยนแปลง ชนิดข้อมูล (data type) ของแอตทริบิวต์ เปลี่ยนแปลงชื่อแอตทริบิวต์ เป็นต้น โดยสามารถเชื่อมการเปลี่ยนแปลงนี้ไปสู่แอตทริบิวต์ระดับภายนอกได้ ทำให้

สติภายนอกก็ยังคงใช้ได้เหมือนเดิมไม่ต้องเปลี่ยนแปลงใดๆ หรือกล่าวว่าเป็นการรักษาความเป็นอิสระของข้อมูลเชิงตรรกะนั้นเอง

3.2 การแปลงรูประหว่างระดับแนวคิดกับระดับภายใน (conceptual/internal mapping) เป็นการกำหนดความสัมพันธ์กันระหว่างมุมมองในระดับแนวคิดกับระดับภายในที่เรียกว่าความเป็นอิสระของข้อมูลเชิงกายภาพ (physical data independence) โดยถ่ายทอดมุมมองที่มีต่อข้อมูลจากสถาปัตยกรรมในระดับแนวคิดไปยังสถาปัตยกรรมในระดับภายในเพื่อนำโครงสร้างของข้อมูลในระดับแนวคิดไปแปลงเป็นโครงสร้างของข้อมูลในระดับกายภาพเพื่อใช้ในการจัดเก็บข้อมูล โดยระบุโครงสร้างเรคคอร์ดและฟิลด์ที่ใช้จัดเก็บข้อมูลในระดับภายใน ถ้าโครงสร้างของข้อมูลในฐานะข้อมูลที่ยึดเก็บเปลี่ยนแปลงไป เช่น เมื่อมีการเปลี่ยนนิยามโครงสร้างการจัดเก็บทำให้การแปลงรูปจากระดับแนวคิดไปยังระดับภายในต้องเปลี่ยนแปลงตามไปด้วย แต่สติภานแนวคิดยังคงอยู่เหมือนเดิมไม่ต้องเปลี่ยนแปลงแต่อย่างใด หรือกล่าวอีกนัยหนึ่งผลของการเปลี่ยนแปลงในระดับภายในต้องเป็นอิสระจากระดับแนวคิด เพื่อที่จะรักษาความเป็นอิสระของข้อมูลเชิงกายภาพนั่นเอง

การแปลงรูปข้อมูลระหว่างระดับแนวคิดกับระดับภายใน ทำให้ผู้ใช้งานฐานข้อมูลไม่ว่าในระดับแนวคิดหรือระดับภายในไม่จำเป็นต้องทราบว่าข้อมูลที่ตนใช้งานอยู่ถูกจัดเก็บในดิสก์อย่างไร เมื่อต้องการใช้ข้อมูลใดสามารถอ้างถึงชื่อตารางและฟิลด์ได้โดยตรง ซึ่งจะเป็นหน้าที่ของระบบจัดการฐานข้อมูลที่จะดูว่าข้อมูลที่ผู้ใช้ต้องการเก็บอยู่ในตำแหน่งแตรกใด ไชลินเคอร์โคในดิสก์ แล้วทำการดึงข้อมูลนั้นมาให้แก่ผู้ใช้

1.4 ระบบจัดการฐานข้อมูล(Database Management System:DBMS)

หน้าที่ของระบบการจัดการฐานข้อมูล

ระบบจัดการฐานข้อมูลเป็นซอฟต์แวร์ที่ทำหน้าที่ดังต่อไปนี้

- ดูแลการใช้งานให้กับผู้ใช้ ในการติดต่อกับตัวจัดการระบบเพิ่มข้อมูลได้ ในระบบฐานข้อมูลนี้ข้อมูลจะมีขนาดใหญ่ ซึ่งจะถูกจัดเก็บไว้ในหน่วยความจำสำรองเมื่อผู้ใช้ต้องการจะใช้งานข้อมูล ระบบการจัดการฐานข้อมูลจะทำหน้าที่ติดต่อกับระบบเพิ่มข้อมูลซึ่งเสมือนเป็นผู้จัดการเพิ่มข้อมูล (file manager) นำข้อมูลจากหน่วยความจำสำรองเข้าสู่หน่วยความจำหลักเฉพาะส่วนที่ต้องการใช้งาน และทำหน้าที่ประสานกับตัวจัดการระบบเพิ่มข้อมูลในการจัดเก็บ เรียกใช้ และแก้ไขข้อมูล
- ควบคุมระบบความปลอดภัยของข้อมูลโดยป้องกันไม่ให้ผู้ที่มิได้รับอนุญาตเข้ามาเรียกใช้หรือแก้ไขข้อมูลในส่วนป้องกันเอาไว้ พร้อมทั้งสร้างฟังก์ชันในการจัดทำข้อมูลสำรอง โดยเมื่อเกิดความขัดข้องของระบบเพิ่มข้อมูลหรือของเครื่องคอมพิวเตอร์เกิดการเสียหายนั้น ฟังก์ชันนี้จะสามารถทำการฟื้นฟูสภาพของระบบข้อมูลกลับเข้าสู่สภาพที่ถูกต้องสมบูรณ์ได้
- ควบคุมการใช้ข้อมูลในสภาพที่มีผู้ใช้พร้อม ๆ กันหลายคน โดยจัดการเมื่อมีข้อผิดพลาดของข้อมูลเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 การเขียนโปรแกรมใน Visual C#.NET

2.8.1 บล็อกของคำสั่ง

พื้นฐานอันดับแรกของคำสั่งส่วนใหญ่ของภาษา VC# ที่ควรทราบก็คือ บล็อกของคำสั่ง ในภาษาVC# จะใช้เครื่องหมาย { ทำหน้าที่เปิดคำสั่ง และใช้เครื่องหมาย } ปิดบล็อกของคำสั่ง ในหลายๆคำสั่งของภาษา VC# ไม่จำเป็นต้องมีบล็อกคำสั่งก็ได้ ส่งผลให้รูปแบบของคำสั่งมี 2 ลักษณะ คือ ทั้งที่มีบล็อกคำสั่ง { } และไม่มีบล็อกคำสั่ง เช่น

```

01 if(เงื่อนไขที่ตรวจสอบ)
02 {
03 // กรณีเงื่อนไขเป็นจริง
04 }
05 else
06 {
07 // กรณีเงื่อนไขเป็นเท็จ
08 }

```

จากตัวอย่างข้างต้น เป็นการ ใช้คำสั่ง if...else ...ที่มีการใช้งานคำสั่ง { } โดยที่ถ้าในกรณีเงื่อนไขเป็นจริง ก็จะมาทำงานที่ได้บรรทัดที่ 2-4 แต่ถ้าในกรณีเงื่อนไขเป็นเท็จ โค้ดก็จะมาทำงานที่บรรทัด 6-8 จะเห็นได้ว่าการใช้งานบล็อกของคำสั่งทำให้ง่ายต่อการศึกษาได้มากกว่าที่จะไม่ใช่ ซึ่งมีผลเป็นอย่างยิ่งในกรณีที่โค้ดมีความยาวมาก มีการตรวจสอบเงื่อนไขหลายๆกรณี ประโยชน์ของบล็อกคำสั่งก็คือ เป็นการบ่งบอกขอบเขตของคำสั่งนั้นๆ ว่าสิ้นสุด ณ จุดใดของโค้ดของคุณ โดยจะยึดถือรูปแบบการใช้เครื่องหมาย { } เพื่อให้สามารถทำความเข้าใจขั้นตอนการทำงาน และทราบถึงขอบเขตของคำสั่งได้อย่างง่ายดาย

2.8.2 ชนิดของข้อมูล

การเลือกใช้ชนิดของตัวแปร (Variable Type) ในโค้ดถือเป็นเรื่องที่มีความสำคัญเป็นอย่างยิ่ง เพราะว่าตัวแปรแต่ละชนิดที่ภาษา VC# สนับสนุนมีมากมาย ให้เลือกใช้ชนิดของตัวแปรให้เหมาะสมกับงานของคุณ โดยพื้นฐานทั้งหมดของของภาษาในชุด .NET สิ่งที่เราควรทำความเข้าใจมีอยู่ 2 อย่าง คือ

1. ทุกสิ่งทุกอย่างที่คุณเรียกใช้งานในการเขียนโค้ด จะถือว่าเป็นออบเจกต์ (Object) หรือว่าเป็นวัตถุทั้งหมดไม่ว่าจะเป็นตัวแปร ออบเจกต์ ตัวแปรออบเจกต์ หรือแม้กระทั่งคอนโทรลที่คุณเรียกใช้งาน ฯลฯ ส่งผลให้ตัวมันเองจะมีคุณสมบัติและเมธอดประจำตัวของมันคิดตัวมาด้วยเสมอ
2. ชนิดของข้อมูลพื้นฐาน (int , string , long ฯลฯ) ถ้ามีการเปลี่ยนชนิดของข้อมูลไปจากเดิม เช่น string เป็น int มีความสำคัญเป็นอย่างยิ่ง มีข้อกำหนดหลายๆอย่างที่ควรทราบ ในบางกรณีคุณจะไม่สามารถเปลี่ยนทันทีตามที่คุณต้องการ สำหรับชนิดของตัวแปรพื้นฐานมีดังนี้

2.8.3 การประกาศตัวแปร

ก่อนที่จะใช้งานตัวแปรใดๆก็ตามจะต้องมีการประกาศตัวแปร เพื่อระบุชนิดของข้อมูลให้กับตัวแปรก่อน ซึ่งก็จะขึ้นอยู่กับว่าต้องการให้ตัวแปรนั้นๆเก็บข้อมูลประเภทใด โดยมีรูปแบบไวยากรณ์การใช้งานดังนี้

```
data_type variable_name;
```

โดยที่ `data_type` หมายถึง ชนิดของข้อมูล เช่น `int`, `long`, `string` ฯลฯ

`variable_name` หมายถึง ชื่อของตัวแปร

2.8.4 ขอบเขตของตัวแปร

หลังจากที่ประกาศตัวแปร สิ่งหนึ่งที่ควรรับทราบก็คือ ตัวแปรดังกล่าวมีขอบเขตที่สามารถเรียกใช้งานได้หรือไม่ ซึ่งมีขอบเขตอยู่ 3 ระดับ คือ

- ระดับ `public` มีขอบเขตที่กว้างที่สุด สามารถเรียกใช้ ณ ตำแหน่งใดก็ได้ที่อยู่ในโปรเจกต์ และควรเลือกใช้ตัวแปรลักษณะนี้ให้ น้อยที่สุด ควรใช้เท่าที่จำเป็น เนื่องจากว่าในการพัฒนาแอปพลิเคชันขนาดใหญ่ การประกาศตัวแปรในระดับ `public` ยากต่อการควบคุมค่า และยากต่อการตรวจสอบค่าที่ตัวมันเองจัดเก็บอยู่
- ระดับ `procedare` มีขอบเขตขนาดกลาง เป็นระดับที่เหมาะสมกับการใช้งานมากที่สุด มีขอบเขตอยู่ในแต่ละเหตุการณ์
- ระดับ `block` มีขอบเขตขนาดเล็กที่สุด มักที่จะใช้เก็บค่าชั่วคราว หรือตัวแปรที่ใช้วนลูปเสียเป็นส่วนใหญ่ เช่น ตัวแปรที่อยู่ในบล็อกของคำสั่งต่างๆ เช่น `if...else...` เป็นต้น สำหรับการเลือกใช้ตัวแปรระดับใดก็ตาม ไม่มีข้อกำหนดตายตัว หรือข้อบังคับใดๆทั้งสิ้น ขึ้นอยู่กับลักษณะของโปรเจกต์มากกว่า เลือกใช้ตามความเหมาะสม

2.8.5 การใช้งานค่าคงที่

ค่าคงที่จะมีลักษณะการใช้งานตรงกันข้ามกับตัวแปร กล่าวคือ คุณสามารถกำหนดค่าหรือปรับเปลี่ยนตัวแปรเป็นค่าอะไรก็ตามที่ตัวแปรดังกล่าวสนับสนุน ขึ้นอยู่กับชนิดของข้อมูล แต่ค่าคงที่ คุณจะกำหนดค่าได้เพียงครั้งเดียวเท่านั้นคือ ในขณะที่ประกาศ และไม่สามารถเปลี่ยนค่าในภายหลัง ซึ่งมีรูปแบบการใช้งานดังนี้

```
const data_type constant_name = value;
```

2.8.6 การแปลงชนิดของข้อมูล

การแปลงชนิดของข้อมูลเพื่อให้การเรียกใช้งานไม่เกิดข้อผิดพลาดในกรณีที่ชนิดของข้อมูลแตกต่างกัน ที่ผ่านมามักเห็นได้ชัดเจนที่สุดก็คือ ตัวเลขที่อยู่ในฐานะข้อความกับตัวเลขที่อยู่ในฐานะตัวเลขมีความแตกต่างกัน ดังนั้น การแปลงชนิดของข้อมูลจึงมีความสำคัญเป็นอย่างยิ่งซึ่งมีอยู่ 2 แบบ คือ

2.8.6.1 การแปลงข้อมูลแบบ `Implicit conversions` มีอยู่ 2 ลักษณะ คือ

1. การแปลงข้อมูลโดยอัตโนมัติ เกิดจากการที่ข้อมูลแต่ละชนิด มีขอบเขตและความสามารถในการเก็บข้อมูลแต่ละชนิดไม่เท่ากัน เมื่อมีการแปลงข้อมูลแล้ว ชนิดของข้อมูลดังกล่าวสามารถยอมรับได้

ก็จะสามารถแปลงชนิดของข้อมูลได้โดยอัตโนมัติ ผู้เขียนขอใช้ทับศัพท์ว่า การแปลงข้อมูลแบบ Implicit เช่น จะเห็นได้ว่าสูงสุดของข้อมูลชนิด short คือ 32,767 คู่กัน ไม่มีทางเกินขอบเขตของข้อมูลชนิด Long ได้โดยอัตโนมัติ จากตัวอย่างข้างต้น ตัวแปร y เป็นข้อมูลชนิด int กำหนดค่าเท่ากับ 50 สามารถถ่ายทอดให้กับตัวแปร x ซึ่งมีข้อมูลชนิด long ได้ทันที เพราะว่ามีขอบเขตที่สามารถยอมรับค่าจากตัวแปร y ได้ ส่งผลให้ตัวแปร x มีค่าเท่ากับ 50 แต่เป็นข้อมูลชนิด long ซึ่งเป็นการแปลงข้อมูลแบบ Implicit ขึ้นพื้นฐาน แต่การแปลงค่าได้แบบนี้ จะมีข้อจำกัดอยู่หลายประการ ซึ่งผู้เขียนจะกล่าวไว้ภายหลัง

2. การแปลงค่าที่เกิดจากการกำหนดค่าเริ่มต้นให้กับตัวแปรใดๆ เป็นการแปลงข้อมูลในกรณีที่สร้างตัวแปรขึ้นมา แล้วต้องการกำหนดค่าเริ่มต้นให้กับตัวแปร เช่น จากโค้ดข้างต้น เป็นการประกาศตัวแปร TmpSum เป็นข้อมูลชนิด double กำหนดค่าคัม ใช้กับตัวแปร TmpSum เท่ากับ 1.5 แต่ตัวเลข 1.5 ดังกล่าว ผู้เขียนกำหนดให้เป็นชนิด float โดยการใช้อักขระควม f (หรือ F) ส่งผลให้ตัวเลข 1.5 เป็นข้อมูลชนิด double ไม่ใช่ float เมื่อถูกเก็บอยู่ในตัวแปร TmpSum การแปลงข้อมูลแบบ Explicit Conversions เป็นการแปลงข้อมูลที่เกิดจากการที่ผู้เขียนเป็นผู้กำหนดการแปลงข้อมูล ผู้เขียนขอใช้ทับศัพท์ว่า การแปลงข้อมูลแบบ Explicit โดยอาศัยอ็อบเจกต์ที่ทำหน้าที่แปลงข้อมูล ซึ่งมีอยู่ 3 วิธี คือ

- อาศัยอ็อบเจกต์ Convert
- อาศัยอ็อบเจกต์ชนิดของข้อมูล
- อาศัยเมธอด Parse ของอ็อบเจกต์ชนิดของข้อมูล

2.8.7 รู้จักอาเรย์แบบต่างๆใน C#

ในบทความนี้เราจะได้ศึกษาถึงอาเรย์แบบต่างๆในภาษา C# ที่เป็นส่วนหนึ่งของเทคโนโลยี .NET พร้อมทั้งศึกษาการเรียงลำดับอาเรย์, การค้นหาข้อมูลของอาเรย์ สำหรับ .NET นั้นอาเรย์ก็จัดได้ว่าเป็นคลาสแบบหนึ่งเหมือนกันตัวแปรแบบอาเรย์ในภาษา C# นั้นจะเริ่มต้นตำแหน่งที่ 0 ก่อนเสมอ ซึ่งข้อมูลที่เราระบุเข้าไปนั้นก็จะไปเก็บอยู่ในตำแหน่งที่ 0 นั้นเอง ตำแหน่งของข้อมูลที่อยู่ที่สุดท้ายนั้นจะอยู่ที่ตำแหน่ง "จำนวนข้อมูลทั้งหมด - 1" ตอนที่เราระบุอาเรย์ขึ้นมาใช้งานนั้นเราสามารถระบุจำนวนสมาชิกให้กับอาเรย์นั้นได้ก็เลยหรือเราจะไม่ระบุก็ได้ซึ่งมันจะเรียกได้ว่าเป็นอาเรย์แบบไดนามิกนั่นเอง เรามาดูตัวอย่างการประกาศการเรียกใช้อาเรย์กันดีกว่า โดยในตัวอย่างเป็นการประกาศอ็อบเจกต์อาเรย์แบบ Integer ซึ่งเป็นแบบไดนามิก ดังนี้

```
int[] intArray;
```

ส่วนโค้ดตัวอย่างด้านล่างเป็นการเป็นการประกาศให้อ็อบเจกต์อาเรย์สามารถเก็บข้อมูลแบบตัวเลขได้ 10 ข้อมูล โดยเริ่มต้นที่ตำแหน่งที่ 0 จนถึง 9

```
Int [] intArray;
```

```
IntArray = new int[10];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.7.1 ประเภทของอาร์เรย์

เราสามารถแบ่งประเภทของอาร์เรย์ได้เป็น 3 ประเภทใหญ่ๆ ได้ดังนี้

1. Single dimensional arrays
2. Multi dimensional arrays หรือ Rectangular arrays
3. Jagged arrays

1.Single Dimensional arrays

สำหรับอาร์เรย์ประเภทนี้นั้นเป็นอาร์เรย์ที่มีรูปแบบการใช้งานง่าย ๆ เหมือนที่แสดงไปไว้ในโค้ดข้างต้น เมื่อเราจะใส่ข้อมูลเข้าไปนั้นสามารถทำได้ดังนี้

```
int[] intArray;
intArray = new int[3];
intArray[0]=1;
intArray[1]=2;
intArray[2]=3;
```

นอกจากการระบุข้อมูลเข้าไปในอ็อบเจกต์อาร์เรย์ในแบบข้างต้นแล้ว เรายังสามารถระบุข้อมูลเข้าไปในตอนที่ `new` อ็อบเจกต์ได้โดยข้อมูลนั้นจะอยู่ในเครื่องหมาย `{ }` เสมอๆ ดังนี้

```
int[] intArray
intArray = new int[3]{1,2,3};
```

เราสามารถย่อโค้ดให้สั้นลงได้อีกดังต่อไปนี้

```
int[] intArray = new int[3]{1,2,3};
```

และยังมีอีกสองวิธีที่เราสามารถระบุข้อมูลเข้าไปในอ็อบเจกต์อาร์เรย์ได้เลยดังนี้

```
int[] intArray = new int[]{1,2,3};
```

หรือ

```
int[] intArray = {1,2,3};
```

เราสะดวกใช้แบบไหนก็เลือกใช้กันได้ตามสะดวกกันเถย แล้วยุ่แค่ความถนัดของแต่ละคน ซึ่งจะเห็นว่าภาษา C# เป็นภาษาที่มีความยืดหยุ่นจริงๆ

2. Multi dimensional arrays

Multi dimensional arrays เป็นอาร์เรย์ที่มีมากกว่า 1 มิติ เราสามารถประกาศอาร์เรย์จaggedได้ดังนี้

```
int[,] intArray
```

ส่วนการกำหนดค่าให้กับอาร์เรย์แบบนี้สามารถกำหนดได้คล้ายๆกับ Single dimensional arrays ดังนี้

```
int[,] intArray = new int[3,2];
intArray[0,0]=0;intArray[1,0]=1;intArray[2,0]=2;
intArray[0,1]=3;intArray[1,1]=4;intArray[2,1]=5;
```

หรือ

```
int[,] intArray = new int[3,2]{{0,3},{1,4},{2,5}};
```

หรือ

```
int[,] intArray = {{0,3},{1,4},{2,5}};
```

ถ้าเราเรียกดูค่าของ `intArray[2,1]` จะมีค่าเท่ากับ 5 ขอให้ดูรูปที่ 2.30 ประกอบไปด้วยนะเพื่อจะเข้าใจการเก็บข้อมูลของอาร์เรย์ประเภทนี้ได้ดีขึ้น

| X \ Y | 0 | 1 | 2 |
|-------|---|---|---|
| 0 | 0 | 1 | 2 |
| 1 | 3 | 4 | 5 |

รูปที่ 2.30 รูปแบบการเก็บข้อมูลแบบ Multi dimensional array

3. Jagged arrays

สำหรับ Jagged arrays นั้น มีบ่อยครั้งที่เราสามารถเรียกได้อีกชื่อหนึ่งว่า "อาร์เรย์ของอาร์เรย์" ซึ่งวิธีการเรียกใช้นั้นอาจจะทำให้เราสับสนได้ โดยมีตัวอย่างวิธีการใช้ดังนี้

```
string [][] strArray = new string[2][];
strArray[0] = new String[2];
strArray[0][0]="Hello";
strArray[0][1]="you";
```

```

strArray[1] = new String[3];
strArray[1][0]="How";
strArray[1][1]="are";
strArray[1][2]="you?";

Console.WriteLine("{0} {1}
{2}",strArray[1][0].ToString(),strArray[1][1].ToString(),strArray[1][2].ToString());
Console.ReadLine();

```

ผลลัพธ์ของโปรแกรมข้างต้นนั้นจะแสดงเป็นคำว่า "How are you?" ให้เราสังเกตหน่อยว่าตอนที่เราประกาศอ็อบเจกต์ strArray ขึ้นมานั้นเราไม่ได้ใส่ขนาดของอาร์เรย์เข้าไปในอาร์เรย์ตัวที่สอง ถ้าเราใส่จำนวนสมาชิกเข้าไปตรงที่โปรแกรมจะผิดพลาดทันทีเลย สำหรับอาร์เรย์ตัวที่สองนั้นกำหนดให้เป็นอาร์เรย์แบบ Single dimensional arrays

เราอาจจะย่อโค้ดให้สั้นเหมือนกับอาร์เรย์สองแบบแรกได้ดังนี้

```

string [][] strArray = new string[2][]
{
    new string[2] {"Hello", "you"},
    new string[3] {"How", "are", "you?"},
};

```

หรือ

```

string [][] strArray = {
    new string[2] {"Hello", "you"},
    new string[3] {"How", "are", "you?"},
};

```

ตัวอย่างต่อไปเป็นแสดงถึงการใช้อาร์เรย์แบบ Multi dimensional arrays เป็นอาร์เรย์ตัวที่สองแทนดังนี้

```

string [,] strArray = new string[2][,]
{
    new string[,] {"Hello", "you"}, {"Hi", "you"}},
    new string[,] {"How", "I"}, {"are", "love"}, {"you?", "u"}
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Console.WriteLine("{0} {1}
{2}",strArray[1][0,0].ToString(),strArray[1][1,0].ToString(),strArray[1][2,0].ToString());
```

ผลที่ได้ของโปรแกรมนี้ก็จะเป็นคำว่า "How are you?" เหมือนเดิม

การวนรอบเพื่อดึงค่าในอาเรย์ออกมา

เราสามารถใ้การวนรอบที่เราเรียกว่าลูปนั้นมาแสดงค่าที่เก็บอยู่ในอาเรย์ทั้งหมดออกมาใช้งานได้ดังนี้

```
using System;
```

```
namespace ConsoleApplication1
{
class Class1
{
[STAThread]
static void Main(string[] args)
{
Console.WriteLine("Single-dimensional arrays");
int[] numArray = new int[3]{1,2,3};
foreach(int number in numArray)
Console.WriteLine(number.ToString());
Console.WriteLine("-----");

Console.WriteLine("Multi-dimensional arrays");
Console.WriteLine("Method1.");
int[,] intArray = new int[3,2]{{0,3},{1,4},{2,5}};
foreach(int number in intArray)
Console.WriteLine(number.ToString());

Console.WriteLine("");
Console.WriteLine("Method2");
for(int i=0;i<=intArray.GetUpperBound(0);i++)
for(int j=0;j<=intArray.GetUpperBound(1);j++)
Console.WriteLine(intArray[i,j].ToString());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

Single-dimensional arrays
1
2
3
-----
Multi-dimensional arrays
Method1.
0
3
1
4
2
5

Method2
0
3
1
4
2
5
-----
Jagged arrays
Hello
you
Hi
you
How
I
are
love
you?
you

```

รูปที่ 2.31 วนรอบแสดงข้อมูลในอาร์เรย์แต่ละแบบออกมา

สำหรับการวนรอบเพื่อเรียกค่าในอาร์เรย์มาใช้นั้นสามารถเขียนโค้ดโดยใช้วิธีแบบอื่นๆ ได้อีก ซึ่งอาจจะไม่เหมือนแบบตัวอย่างข้างต้นก็ได้

การเรียงลำดับอาร์เรย์

สำหรับการเรียงลำดับข้อมูลในอาร์เรย์นั้นเราก็สามารถทำได้ง่ายๆเลย โดยทำการเรียกใช้เมธอด "Sort" ของคลาส Array ดังนี้

```

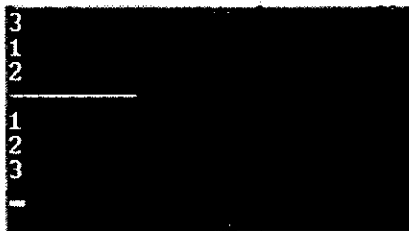
int[] numArray = new int[3]{3,1,2};
foreach(int number in numArray)
Console.WriteLine(number.ToString());

Console.WriteLine("-----");
Array.Sort(numArray);
foreach(int number in numArray)
Console.WriteLine(number.ToString());

```

ผลที่ได้เป็นดังรูปที่ 2.32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.32 ข้อมูลจะถูกเรียงจากน้อยไปมากอย่างรวดเร็ว เพียงเรียกใช้เมธอด Sort

แต่ว่ามีข้อจำกัดคือใช้ได้เฉพาะกับอาร์เรย์แบบ Single dimensional arrays

การค้นหาข้อมูลในอาร์เรย์

สำหรับการค้นหาข้อมูลในอาร์เรย์นั้นเราก็จะใช้ฟังก์ชัน "BinarySearch" ของคลาส Array ในการค้นหาค่าที่เราต้องการออกมา ดังตัวอย่างด้านล่าง

```
string[] strArray = new string[]{"Hi","Me","Hello"};
int result = Array.BinarySearch(strArray,"Me");
Console.WriteLine("Find at \"Me\" at index {0}",result);
```

ผลที่ได้จะเป็นดังรูปที่ 2.33

รูปที่ 2.33 ค้นหาว่า "Me" ที่ตำแหน่งที่ 1

แต่ว่ามีข้อจำกัดเหมือนกับเมธอด Sort ก็คือใช้ได้เฉพาะกับอาร์เรย์แบบ Single dimensional arrays

สำหรับเรื่องของอาร์เรย์นั้นถ้าเรามีความเข้าใจมันเป็นอย่างไร เราสามารถนำมันไปประยุกต์ใช้งานต่างๆได้ เป็นอย่างดีทีเดียว

2.8.8 ส่งไฟล์ข้ามเครือข่ายโดยใช้ Web Service ด้วย Microsoft Visual C#.NET

System requirement

1. Microsoft .NET Framework 1.1
2. Microsoft Visual Studio.NET 2003
3. Microsoft Internet Information Service (IIS) 5.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับเว็บแอปพลิเคชัน ปัจจุบันการส่งไฟล์จากเครื่องไคลเอนท์ไปยังเซิร์ฟเวอร์นั้นสามารถทำได้ง่ายกว่า เพราะทาง Microsoft .NET Framework นั้นได้เตรียมคลาสต่างๆที่ใช้สำหรับการรับไฟล์จากไคลเอนท์มาให้เราใช้กันอยู่แล้ว แต่สำหรับวินโดวส์แอปพลิเคชันนั้น กลับกลายเป็นเรื่องที่ยากกว่า เพราะเราจะต้องทำการเขียนโปรแกรมเกี่ยวกับ Socket ขึ้นมาเองเพื่อทำการรับส่งไฟล์ ซึ่งสำหรับหลายๆคนนั้นก็ไม่สามารถเขียนโปรแกรมเกี่ยวกับ Socket ได้ อีกทั้งเมื่อเขียนโปรแกรม Socket ที่ใช้ในการรับส่งไฟล์ขึ้นมาได้แล้ว เมื่อนำไปใช้จริงนั้น ก็อาจติดไฟร์วอลล์ อีกทั้งก็เป็นไปได้ คุณแล้วมันยุ่งยากเหมือนกัน แต่ด้วยเทคโนโลยีของเว็บเซอร์วิสนั้นมันกลับทำให้เรื่องนี้กลายเป็นเรื่องง่ายสามารถใช้ได้ทั้งเว็บและวินโดวส์แอปพลิเคชันได้ทั้งคู่ และสามารถทะลุไฟร์วอลล์ได้ด้วย เพราะว่าเว็บเซอร์วิสนั้นมันรับส่งข้อมูลผ่านทางพอร์ต 80 ซึ่งเป็นพอร์ตเดียวกับที่บราวเซอร์นั้นใช้สำหรับรับส่งข้อมูลอยู่แล้ว เรามาตัวอย่างของเค้ากันเลยดีกว่านะครับ ซึ่งมันจะมีอยู่ด้วยกันทั้งหมดสอง โปรเจกต์คือ โปรเจกต์ “WSFileUpload” และ “WSFileUpload.App” เรามาดูโปรเจกต์ WSFileUpload กันก่อนดีกว่า

● โปรเจกต์ WSFileUpload

สำหรับโปรเจกต์นี้นั้นจะมีไฟล์ที่เป็นเว็บเซอร์วิสอยู่คือไฟล์ “file_manager.aspx” ซึ่งไฟล์นี้แหละที่ทำหน้าที่เป็นเซิร์ฟวิสต์ฝั่งเซิร์ฟเวอร์ โดยเราจะทำการเรียกใช้เนมสเปซ “System.IO” และ “System.Configuration” มาด้วย เนื่องจากเราจะทำงานกับไฟล์ และอ่านค่าบางค่ามาจากไฟล์ “Web.config”

เริ่มต้นจะเห็นว่าเราจะทำการอ่านค่าของไฟล์คอนฟิกที่ระบุที่ส่งมาว่าอยู่ที่ไหนดังนี้

```
private string strBaseDirectory = ConfigurationSettings.AppSettings["FileSaveToLocation"].ToString();
```

ในเว็บเมธอดแรกคือ UploadFile นั้นเราจะต้องทำการส่งชื่อและรหัสผ่านผู้ใช้ พร้อมทั้งชื่อไฟล์และข้อมูลไฟล์ที่อ่านมาเป็น โปสต์ให้เมธอดนี้ด้วย ซึ่งจะคืนค่ากลับ ไปเป็นบูลีนว่าสามารถบันทึกรูปได้หรือไม่ ดังนี้

```
public bool UploadFile(string strUserName, string strPassword, string strFileName, byte[] objFile, ref string strReturn)
```

ตัวแปร strReturn นั้นใช้สำหรับส่งข้อความผิดพลาดกลับไป เพื่อแจ้งให้ไคลเอนท์รู้ว่าผิดพลาดเนื่องจากอะไร ภายในเมธอดนี้ก็จะทำการตรวจเช็คค่าพาทของไฟล์นั้นว่ามีเครื่องหมาย “\” ปิดท้ายหรือไม่ ถ้าไม่มีก็ให้ใส่เข้าไปด้วย

```
if (strBaseDirectory.EndsWith(@"\") == false)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
strBaseDirectory += @"\\";
```

```
}
```

ต่อมาในเงื่อนไข try...catch นั้นก็จะทำการตรวจสอบสิทธิ์ของผู้ใช้ดูก่อนว่ามีสิทธิ์บันทึกไฟล์หรือไม่

```
if (strUserName != "username" && strPassword != "password")
```

```
{
```

```
    throw new UnauthorizedAccessException();
```

```
}
```

ถ้ามีสิทธิ์ก็จะมาทำการตรวจสอบไฟล์เดอ์เป้าหมายว่ามีอยู่จริงหรือไม่ ซึ่งจะโยนให้ฟังก์ชัน “checkfolderexists” ทำหน้าที่ดังกล่าว

```
if (checkfolderexists(strBaseDirectory) == false)
```

```
{
```

```
    throw new Exception("Base Folder does not exist and could not create.");
```

```
}
```

ถ้าไม่มีอะไรผิดพลาด มันก็จะทำการสร้างไฟล์ว่างๆขึ้นมาตามชื่อเดียวกับไฟล์ที่ส่งมา และก็หาขนาดของไฟล์ที่ส่งมา แล้วค่อยเขียนข้อมูลไฟล์ที่ส่งมาเข้าไปในไฟล์ที่สร้างเตรียมไว้

```
objFileStream = File.Open(strBaseDirectory + strFileName, FileMode.Create, FileAccess.Write);
```

```
long lngLen = objFile.Length;
```

```
objFileStream.Write(objFile, 0, ((int)lngLen));
```

```
objFileStream.Flush();
```

```
objFileStream.Close();
```

```
bolRcturnStatus = true;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของเงื่อนไข Finally นั้นเราจะทำการปิดไฟล์ที่สร้างขึ้นมาด้วย ซึ่งเราเขียนค้ไว้ในกรณีที่สร้างไฟล์มาแล้วเกิดความผิดพลาดไม่สามารถเขียนข้อมูลลงไฟล์ดังกล่าวได้ ถ้าไม่ปิดมันเราจะไม่สามารถทำอะไรกับไฟล์นี้ได้เลยจนกว่าเว็บเซิร์ฟเวอร์จะรีสตาร์ท

```
finally
```

```
{
```

```
    if (objFileStream != null)
```

```
    {
```

```
        objFileStream.Close();
```

```
    }
```

```
}
```

```
return bolReturnStatus;
```

มาดูกันที่ฟังก์ชัน checkfolderexists ซึ่งเมธอดนี้จะทำหน้าที่ตรวจสอบว่ามีโฟลเดอร์ที่จะทำการบันทึกไฟล์อยู่มัย ถ้าไม่มีก็ให้ทำการสร้างขึ้นมาด้วย

```
string[] sFolderSplit = sFolder.Split(new char[] { '\\' });
```

```
string strFolder = "";
```

```
try
```

```
{
```

```
    for (int i = 0; i <= sFolderSplit.GetUpperBound(0) - 1; i++)
```

```
    {
```

```
        strFolder += sFolderSplit[i] + @"\";
```

```
        if (Directory.Exists(strFolder) == false)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Directory.CreateDirectory(strFolder);
    }
}

return true;
}

catch (Exception ex)
{
    return false;
}

```

สำหรับเซอวีวิสที่ให้บริการก็มีได้เพียงแค่นี้ครับ เรามาดูที่ไฟล์ “fileuploadexample.aspx.” กันเลย ซึ่งทำหน้าที่ในการเลือกไฟล์ต้องการส่ง มีหน้าตาดังรูปนี้

The image shows a web form with a text input field containing the filename 'fileuploadexample.aspx'. To the right of the input field is a 'Browse...' button. Below the input field is an 'Upload' button.

รูป ไฟล์ “fileuploadexample.aspx.”

เมธอด Button1_Click นั้นทำหน้าที่เรียกใช้เซอวีวิสที่เราสร้างขึ้นมา ซึ่งจะทำหน้าที่ในการส่งไฟล์ไป โดยเริ่มต้นจะทำการตรวจว่ามีไฟล์ส่งมาหรือไม่

```
if (Request.Files.Count > 0)
```

ถ้ามีไฟล์ส่งมาจริงก็ให้ทำการดึงไฟล์มา และก็หาขนาดของไฟล์แล้วประกาศตัวแปรอาเรย์ไบต์ตามขนาดข้อมูลที่ได้อามาเพื่อใช้สำหรับอ่านข้อมูลไฟล์เข้าไปเก็บไว้

```
oFileCollection = Request.Files;
```

```
oFile = oFileCollection[0];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FileLen = oFile.ContentLength;
```

```
byte[] oFileByte = new byte[FileLen];
```

ทำการหาชื่อไฟล์ที่ส่งมาว่าชื่ออะไร

```
sFileName = Path.GetFileName(oFile.FileName);
```

จากนั้นก็ทำการอ่านไฟล์เข้าไปเก็บไว้ในตัวแปร oFileByte

```
oStream = oFile.InputStream;
```

```
oStream.Read(oFileByte, 0, FileLen);
```

สุดท้ายก็ทำการเรียกใช้เซอร์วิส UploadFile

```
bolResult = oWS.UploadFile("username", "password", sFileName, oFileByte, ref strReturn);
```

สำหรับอ็อบเจกต์ oWS นั้นเป็นอินสแตนซ์ของคลาส file_manager ที่เราประกาศไว้ในตอนต้นดังนี้

```
file_manager oWS = new file_manager();
```

สำหรับไฟล์ fileuploadexample.aspx นั้นก็มีส่วนของโค้ดหลักๆเท่านั้นครับ แต่ก่อนที่จะรันโปรเจกต์นั้นให้ไปดูที่ไฟล์ Web.config กันหน่อยนะครับ ซึ่งจะเห็นว่าเรากำหนดค่าของโฟลเดอร์ไว้ดังนี้

```
<appSettings>
```

```
    <!-- Base Directory where you want the uploaded files saved -->
```

```
    <add key="FileSaveToLocation" value="C:\FileUpload" />
```

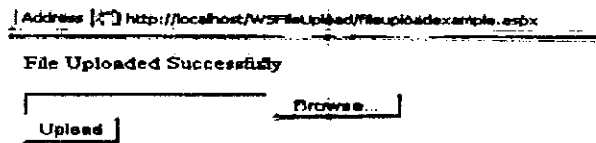
```
</appSettings>
```

อีกส่วนที่จะให้ดูคือค่าแอททริบิวต์ "maxRequestLength" ของ โหนด "httpRuntime" นั้นเราใช้กำหนดขนาดไฟล์ที่สามารถส่งได้ ซึ่งปกติจะอยู่ที่ 4 MB แต่เรากำหนดไว้ที่ 100 MB แต่การที่จะส่งไฟล์ขนาด 100 MB มาได้นั้นก็ขึ้นอยู่กับปัจจัยอื่นๆอีกด้วยนะ เช่นสมรรถภาพของเครื่องเซิร์ฟเวอร์ และที่สำคัญคือความเร็วของระบบเน็ตเวิร์ค

```
<httpRuntime maxRequestLength="102400" />
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

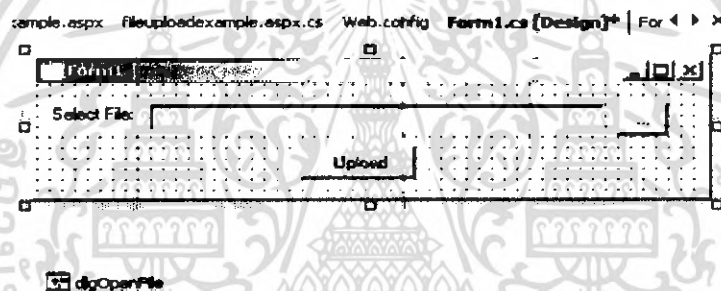
ทำการรัน โปรแกรมเช็คทดสอบดูได้เลย ถ้าไม่มีอะไรผิดพลาดผลที่ได้ก็แสดงออกมาดังรูปนี้



ลองใช้จากวินโดวส์เอพพลิเคชัน



เปิดไฟล์ Form1 ขึ้นมา จะเห็นว่าหน้าตาของโปรแกรมดังรูปนี้



เมฆอด btnFileSelect_Click นั้นจะทำการเปิดหน้าต่างขึ้นมาให้เราเลือกไฟล์ เมื่อเลือกเสร็จแล้วก็จะนำมาใส่ใช้คอนโทรล TextBox ซึ่งมีโค้ดดังนี้

```
private void btnFileSelect_Click(object sender, System.EventArgs e)
{
    if (dlgOpenFile.ShowDialog() != DialogResult.Cancel)
    {
        txtFile.Text = dlgOpenFile.FileName;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมธอด `btnUpload_Click` นั้นทำหน้าที่เรียกใช้เมธอด `doFileUploadViaWebService` เพื่อทำการส่งไฟล์ไปยังเซิร์ฟเวอร์ผ่านทางเว็บเซอร์วิส

```
private void btnUpload_Click(object sender, System.EventArgs e)
{
    if (File.Exists(txtFile.Text) == false)
    {
        return;
    }

    FileInfo oFileInfo = new System.IO.FileInfo(txtFile.Text);
    doFileUploadViaWebService(oFileInfo);
}
```

เมธอด `doFileUploadViaWebService` ทำหน้าที่อ่านไฟล์ที่ส่งเข้ามาให้ไปเก็บในตัวแปร `objWS` ก่อนเรียกใช้เซิร์ฟวิส `UploadFile` เริ่มต้นเราจะทำการสร้างอ็อบเจกต์ `Ows` จากคลาส `file_manager` ขึ้นมาก่อน

```
ws_uploadfile.file_manager objWS = new ws_uploadfile.file_manager();
```

จากนั้นจะเห็นว่ามีการอ่านค่า `Configuration` มาจากไฟล์ `"app.config"` อยู่ด้วย ซึ่งเราจะเอาค่าที่อ่านมาได้ไประบุเข้าไปในหรือบเพอร์ตี `Url` ของคลาส `file_manager` ด้วย ซึ่งที่เราระบุไปก็เพื่อที่ว่าในวันหนึ่งเราอาจทำการย้ายเซิร์ฟวิสนี้ไปอยู่ที่เซิร์ฟเวอร์ตัวอื่น เราก็มาแก้ที่ไฟล์ `app.config` โดยตรง โดยไม่ต้องทำการเข้ามาแก้ในโปรเจกต์อีกครั้งแต่ละคอมไพล์มันใหม่

2.8.9 โคลแอนต์-เซิร์ฟเวอร์

ซีพียูแต่ละตัวที่อยู่ใน ไมโครคอมพิวเตอร์ได้รับการพัฒนาให้มีประสิทธิภาพและความรวดเร็วสูงมาก การใช้งานที่ซีเป็นเพียงเทอร์มินอลที่ต่ออยู่กับเมนเฟรม จึงจะไม่ค่อยได้ใช้ประสิทธิภาพของซีพียูเลย การเพิ่มประสิทธิภาพให้ระบบในแลนจึงเน้นที่ให้ซีพียูทุกตัวช่วยกันทำงาน ซีพียูที่ให้บริการช่วยหรือที่เรียกว่า `Server` ซีพียูตัวที่ขอใช้บริการก็เรียกว่า `Client` การขอใช้บริการจะแบ่งแยกเป็นฟังก์ชันต่างๆตามหน้าที่โปรแกรมประยุกต์ในปัจจุบันต้องการแสดงผลบนหน้าจอภาพที่เป็น `window` แสดงในลักษณะ `graphic` ดังนั้น `PC` ของผู้ใช้จึงทำหน้าที่หลักในการจัดการแสดงผล แต่เมื่อต้องการข้อมูลก็จะขอใช้บริการจาก `File Server` ตัวที่ให้บริการข้อมูลหลักก็เรียกว่า `File Server` หากมองในมุมกลับถ้าเครื่องที่จัดการแสดงข้อมูลมี

ความสามารถในการดูแล file และจัดการคิวข้อมูล คิวนี้จะอยู่ในฟังก์ชัน Database Server แต่ Database Server ทำงานการแสดงผลไม่ได้แต่ข้อมูลที่ต้องการ ดังนั้นจึงต้องไปรับบริการกับตัวลูกเพื่อแสดงผลให้ผู้ใช้ ตัวลูกที่ใช้ในการแสดงผลจึงเป็นตัวให้บริการทางด้าน การแสดงผล หรือเป็นสถานีปลายทาง เราเรียกว่า Terminal Server ดังนั้นจึงเห็นได้ว่าการเป็น Client จะไปอยู่ที่ตัว Database Server ด้วยระบบการขอใช้บริการและการให้บริการจึงมีหลากหลายตามฟังก์ชันการใช้งาน

2.8.10 การทำงานในระบบ Client - Server

การทำงานภายใต้ระบบ Client – Server จึงประกอบด้วย ส่วนสำคัญ 3 ส่วน คือ ส่วนของผู้ใช้ บริการหรือที่เรียกว่า Client ส่วนเครือข่ายและส่วนของผู้ให้บริการหรือ Server ด้วยเหตุนี้การพัฒนา Software สมัยใหม่ในระบบ Client – Server จึงต้องคำนึงถึงส่วนทั้งสาม และหากจะพิจารณาให้กว้างออกไป Client ตัวหนึ่งอาจเรียกใช้บริการจาก Server หลายแห่งก็ได้ หรือจะเรียก Server ที่ทำงานแตกต่างกัน หากในระบบมี Server ที่ดูแลฐานข้อมูลหลายตัว ระบบนั้นก็จะเป็นระบบการจัดการข้อมูลแบบกระจาย ดังนั้น ระบบฐานข้อมูลแบบกระจายจึงเป็นระบบที่ต้องทำงานโดยอาศัยการทำงานตาม Model Client Server ในระบบเดียวกันมีการเรียกหลายฟังก์ชัน เช่น ตัวให้บริการการจัดพิมพ์ข้อมูลโดย Client หลายตัวมาขอใช้บริการได้ เราก็เรียกว่า Print Server ตัวให้บริการ Terminal ทำให้ต้องใช้ Terminal ได้มากก็เรียกว่า Terminal Server

บทที่ 3

การออกแบบและขั้นตอนการทำงาน

3.1 ข้อมูลเบื้องต้น

โครงการที่จัดทำนี้เก็บข้อมูลการลงทะเบียนจากเขตเลือกตั้งแล้วนำมาแสดงผลจะประกอบด้วยส่วนของผู้ดูแลระบบซึ่งสามารถจัดการฐานข้อมูลได้ดังนี้

3.1.1 ส่วนผู้ดูแลระบบ

1. จะมีหน้าจอในการจัดการระบบก่อนเข้าไปดูข้อมูลการเลือกตั้งเพื่อแสดงผลดังนี้

1.1 ชื่อในการล็อกอิน (Username)

1.2 รหัสผ่าน (password)

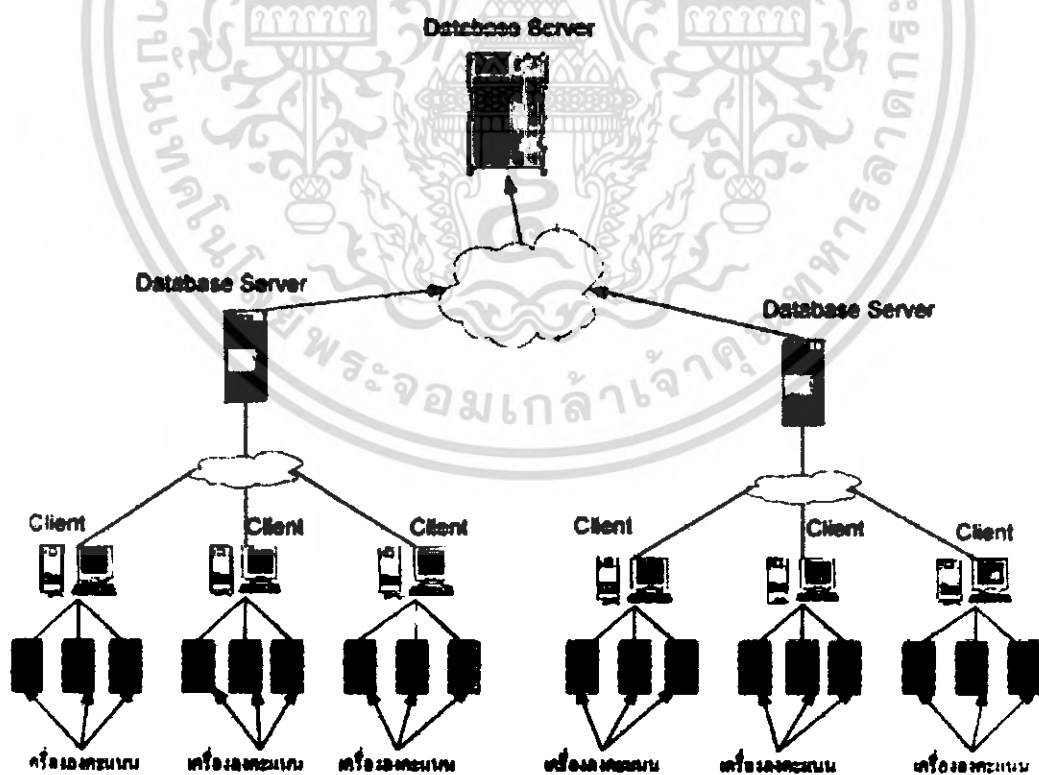
2. มีหน้าที่จัดการฐานข้อมูลต่างๆ

2.1 ป้อนข้อมูลต่างๆในระบบ

2.2 แสดงผลการลงทะเบียน

3.2 ขั้นตอนการทำงานของระบบ

โครงการนี้เป็นการนำเอาระบบอิเล็กทรอนิกส์และระบบฐานข้อมูลมาใช้ในระบบการเลือกตั้งผ่านเครือข่ายคอมพิวเตอร์ ซึ่งโครงการนี้แบ่งระบบการทำงานออกเป็น 2 ระบบ คือ ระบบของเครื่องลงคะแนน และระบบฐานข้อมูลของการเลือกตั้งผ่านเครือข่ายคอมพิวเตอร์



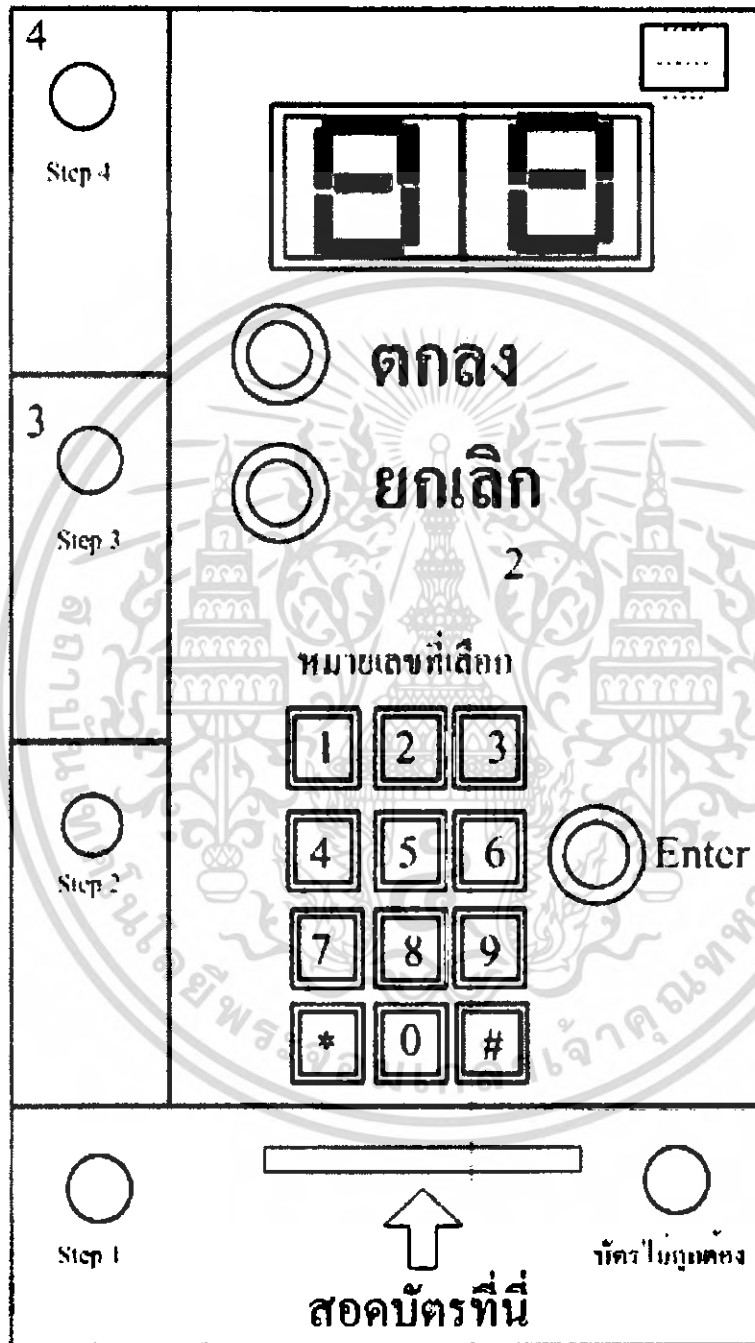
รูปที่ 3.1 แสดงขอบเขตการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1 ระบบของเครื่องลงคะแนน

ระบบของเครื่องลงคะแนนนี้ แบ่งระบบการทำงานออกเป็น 2 จุด คือ

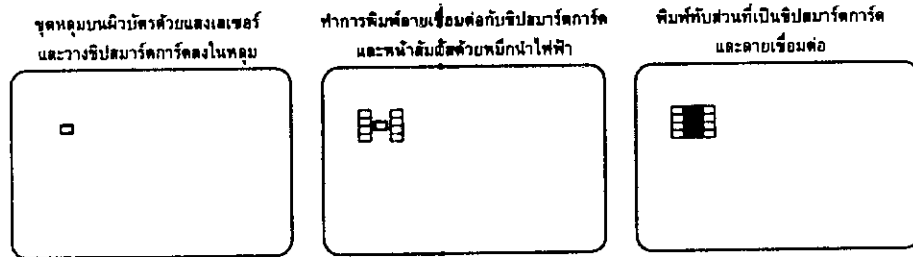
1. เครื่องลงคะแนน



รูปที่ 3.2 แสดงรูปแผงหน้าปัดของเครื่องลงคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บัตรลงคะแนน

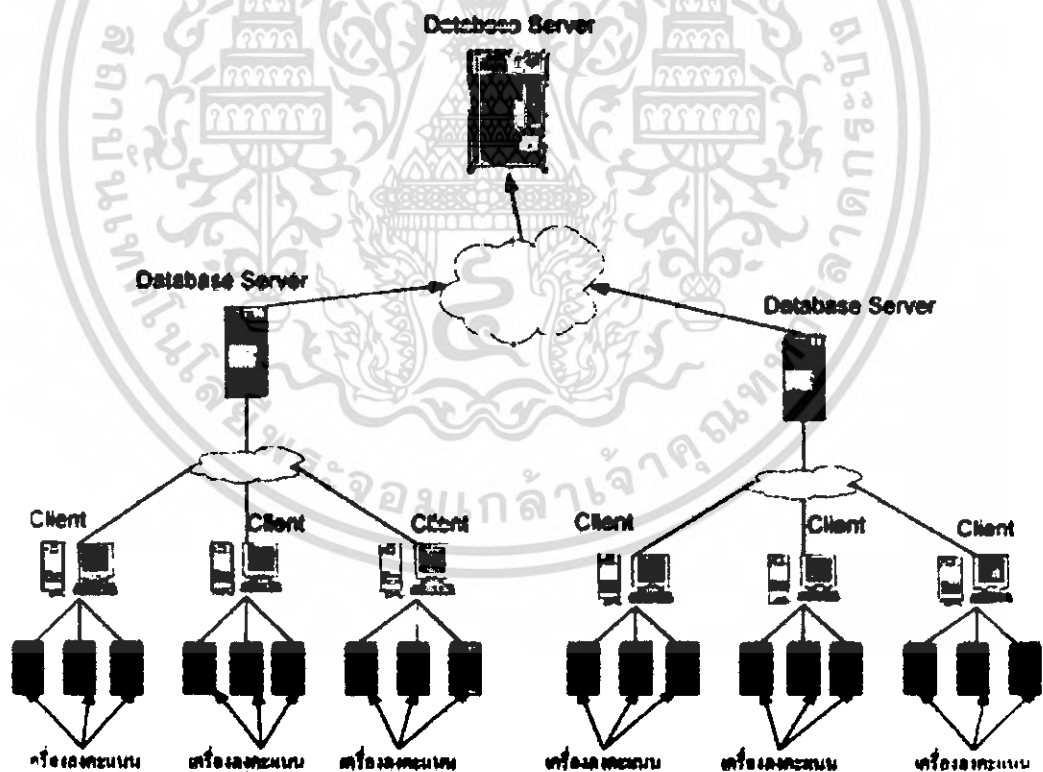


รูปที่ 3.3 แสดงบัตรที่ใช้ลงคะแนน

3.2.2 Network Diagram

อุปกรณ์ในเครือข่ายประกอบด้วยส่วนต่างๆ ดังต่อไปนี้ เครื่องคอมพิวเตอร์ Database Server ที่ทำหน้าที่เก็บฐานข้อมูลการเลือกตั้ง

- เครื่องคอมพิวเตอร์ Client
- เครื่องลงคะแนน



รูปที่ 3.4 โครงสร้างระบบเครือข่ายคอมพิวเตอร์สำหรับการเลือกตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การออกแบมระบบ

1 ชุดอ่านเขียนบัตรสมาชิกการ์ด

1. ทำการเสียบบัตรสมาชิกการ์ดจะปรากฏไฟสีเขียวขึ้น
2. ถ้าต้องการเขียนหมายเลขบัตรประจำตัวประชาชน 13 หลักลงบัตรสมาชิกการ์ด ให้พิมพ์คำสั่งดังนี้

:1xxxxxxxxxxxxx ตามด้วย Enter

หมายเหตุ **xxxxxxxxxxxxx** คือหมายเลขบัตรประจำตัวประชาชน 13 หลักเป็นตัวเลขเท่านั้น

ตัวอย่างพิมพ์ สมมุติ 1234567890123 คือหมายเลขบัตรประจำตัวประชาชน 13 หลัก

:11234567890123 ตามด้วย Enter

ถ้าสามารถเขียนหมายเลขบัตรประจำตัวประชาชน 13 หลักลงบัตรได้จะส่งค่านี้ตามรูปกลับมายังคอมพิวเตอร์

Print Hyperterminal

๐๕ ๖๖ ๐๖ ๕๕

:1

ถ้าไม่สามารถเขียนหมายเลขบัตรประจำตัวประชาชน 13 หลักลงบัตรได้จะส่งค่านี้ตามรูปกลับมายังคอมพิวเตอร์

Print Hyperterminal

๐๕ ๖๖ ๐๖ ๕๕

:9

3. ถ้าต้องการอ่านหมายเลขบัตรประจำตัวประชาชน 13 หลักจากบัตรสมาร์ตการ์ด ให้พิมพ์คำสั่งดังนี้

:0 ตามด้วย Enter

ถ้าสามารถอ่านหมายเลขบัตรประจำตัวประชาชน 13 หลักจากบัตรได้จะส่งค่านี้ตามรูปกลับมายังคอมพิวเตอร์

01234567890123

01234567890123

:01234567890123

หมายเหตุ 1234567890123 คือหมายเลขบัตรประจำตัวประชาชน 13 หลัก ที่ได้ทำการเขียนลงไป
บัตร อาจเป็นค่าอื่นก็ได้ ที่ได้ทำการเขียนลงไป

ถ้าไม่สามารถอ่านหมายเลขบัตรประจำตัวประชาชน 13 หลักจากบัตรได้จะส่งค่านี้ตามรูปกลับมา
ยังคอมพิวเตอร์

01234567890123

01234567890123

:9

2. ชุดเครื่องเลือกตั้ง

1. ขั้นตอนที่ 1

ทำการเสียบบัตรสมาชิกจะปรากฏไฟสีเขียวขึ้น และจะส่งหมายเลขบัตรประจำตัวประชาชน 13 หลักที่เสียบไปมายังคอมพิวเตอร์

Text Typing Tutorial

0 ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙

:01234567890123

:0XXXXXXXXXXXX

ข้อมูลที่ส่งกลับมาอธิบายได้ดังนี้ :0 คือบอกเป็นการทำกระบวนการขั้นตอนแรกที่เกิดต่อกับคอมพิวเตอร์

ส่วน XXXXXXXXXXXXX คือ หมายเลขบัตรประจำตัวประชาชน 13 หลัก ซึ่งก็คือ 1234567890123 นั่นเอง

ซึ่งต้องการตรวจสอบกับฐานข้อมูลที่คอมพิวเตอร์ว่าหมายเลขบัตรประจำตัวประชาชนนี้สามารถเลือกตั้งได้หรือไม่

ให้พิมพ์ :1 ตามด้วย Enter ลงไปที่คอมพิวเตอร์เพื่อเป็นการตอบกลับว่าหมายเลขบัตรประจำตัวประชาชนนี้สามารถเลือกตั้งได้ ก็จะสามารถทำขั้นตอนที่ 2 ของการเลือกตั้งได้ **LED ดวงที่ 1 จะติด**

ถ้าสมมุติว่าบัตรนี้ไม่สามารถเลือกตั้งได้ก็ ให้พิมพ์ :0 ตามด้วย Enter ลงไปที่คอมพิวเตอร์จะทำให้เครื่องกลับสู่สถานะเริ่มต้นใหม่ ต้องดึงบัตรออกจากหัวอ่าน

2. ขั้นตอนที่ 2

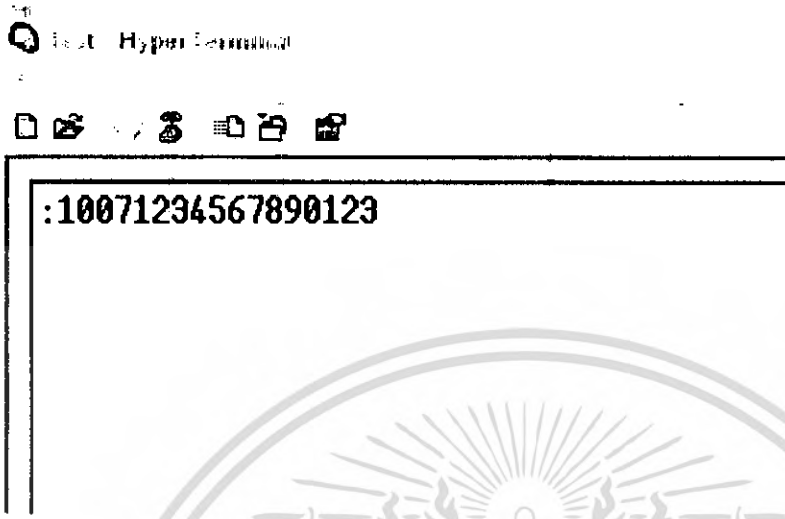
ที่ชุดเครื่องเลือกตั้ง ให้กดหมายเลขผู้สมัครที่ต้องการเลือกลงไป ตามคปุ่ม Enter เมื่อต้องการเลือกผู้สมัครคนนี้ เมื่อกดแล้วก็ไม่สามารถกดหมายเลขเพิ่มเติมได้ นอกจากกด Ok หรือ Cancel เท่านั้น

-ในขั้นตอนนี้ถ้าต้องการยกเลิกหมายเลขที่ผู้สมัครและต้องการเลือกใหม่ให้กดปุ่ม Cancel ได้เลย

-สมมุติว่าต้องการเลือกผู้สมัครเบอร์ 7 ให้กดหมายเลข 7 แล้วกด Enter ที่ชุดเครื่องเลือกตั้ง **LED ดวงที่ 2 จะติด**

3. ขั้นตอนที่ 3

กดปุ่ม Ok ที่ชุดเครื่องเลือกตั้ง เมื่อต้องการเลือกผู้สมัครคนนี้จริงๆ ที่ชุดเครื่องเลือกตั้งจะส่งข้อมูลกลับมาดังนี้ตามรูป

 :10071234567890123

 :1zzzxxxxxxxxxxxxxx

ข้อมูลที่ส่งกลับมามีรายละเอียดดังนี้ :1 คือบอกเป็นการทำกระบวนการขั้นตอนสุดท้ายที่ติดต่อกับคอมพิวเตอร์

ส่วน zzz คือหมายเลขผู้สมัคร ที่ทำการเลือกในขั้นตอนที่ 2 ได้ทำการเลือกหมายเลข 7 ก็ คือ 007

ส่วน xxxxxxxxxxxxxxx คือ หมายเลขบัตรประจำตัวประชาชน 13 หลัก ซึ่งก็คือ 1234567890123 นั่นเอง การส่งข้อมูลนี้มาเพื่อทำการเลือกผู้สมัครคนนี้โดยตรวจสอบว่าสามารถเลือกผู้สมัครคนนี้ได้หรือเปล่า

สมมุติถ้าสามารถเลือกได้ ให้พิมพ์ :1 ตามด้วย Enter ลงไปที่คอมพิวเตอร์ LED ดวงที่ 3 จะติด

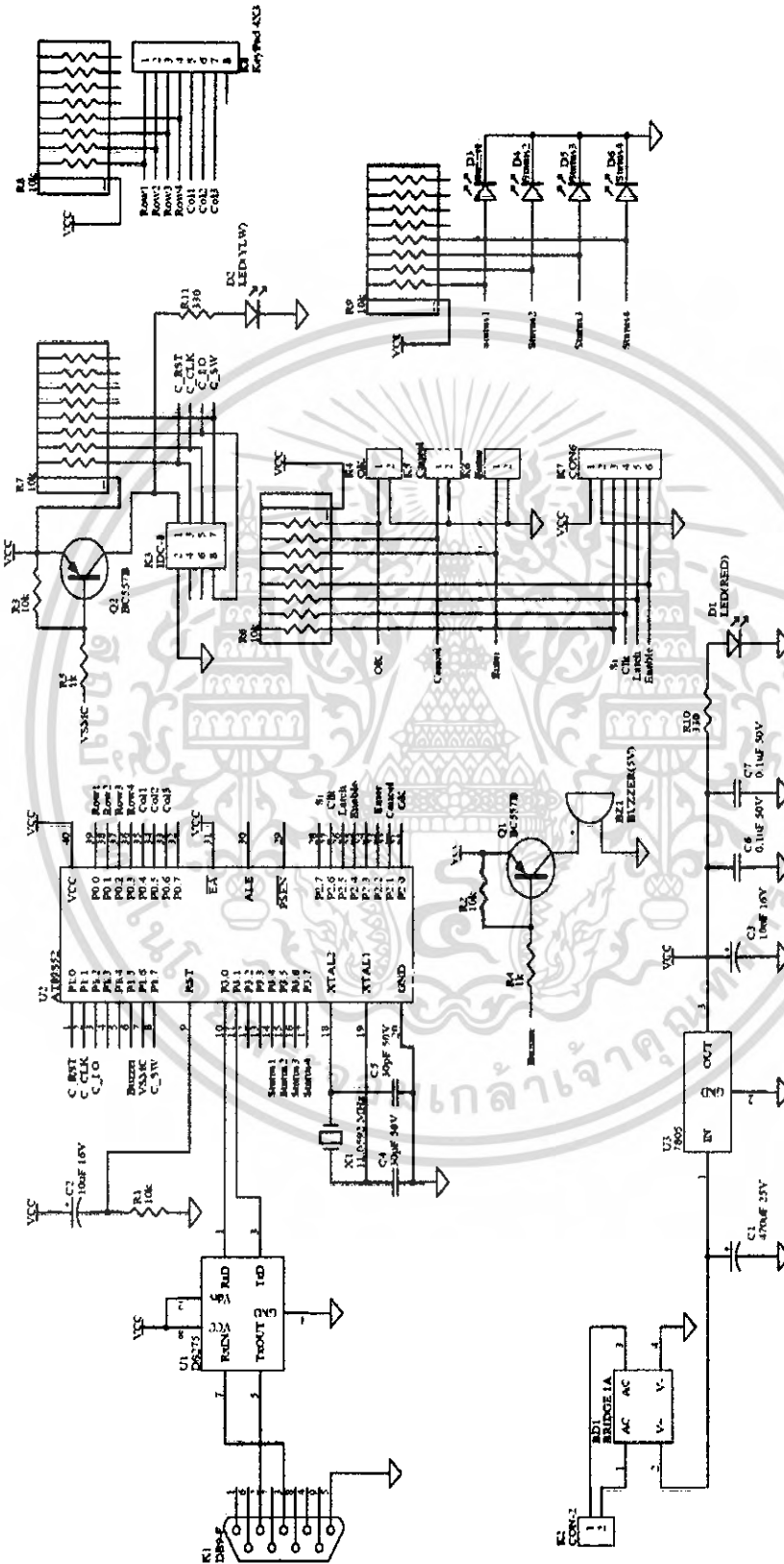
และ LED สีเหลืองจะดับ ถ้าดึงบัตรออก LED สีเขียวทั้ง 3 ดวงจะดับ

สมมุติถ้าไม่สามารถเลือกได้ ให้พิมพ์ :1 ตามด้วย Enter ลงไปที่คอมพิวเตอร์ LED สีเหลืองจะดับ

และ LED สีเขียวทั้งหมดจะดับ

3.2.4 การออกแบบวงจร

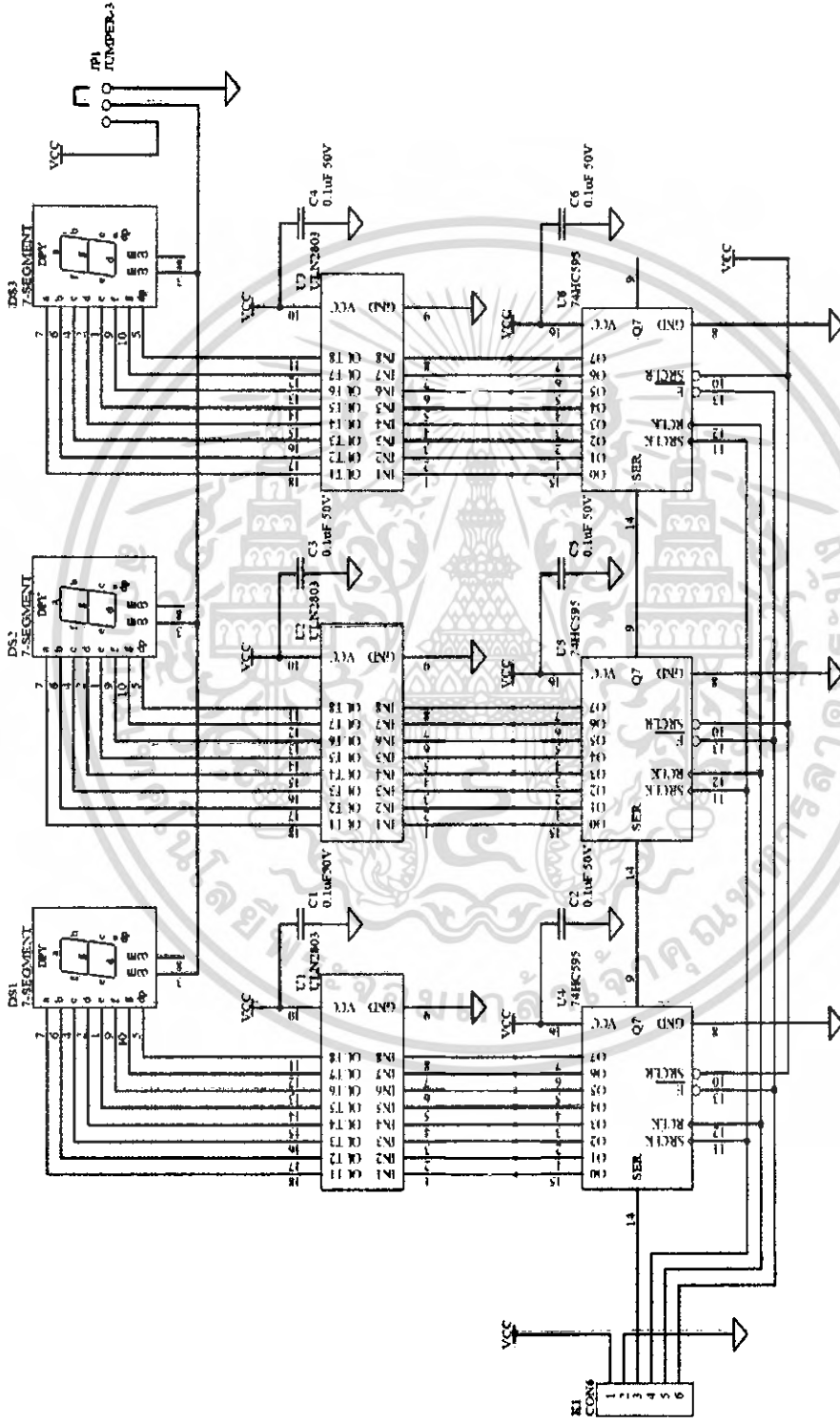
1. วงจรของเครื่องกะตะแมน



รูปที่ 3.5 วงจรการทำงานของเครื่องกะตะแมน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

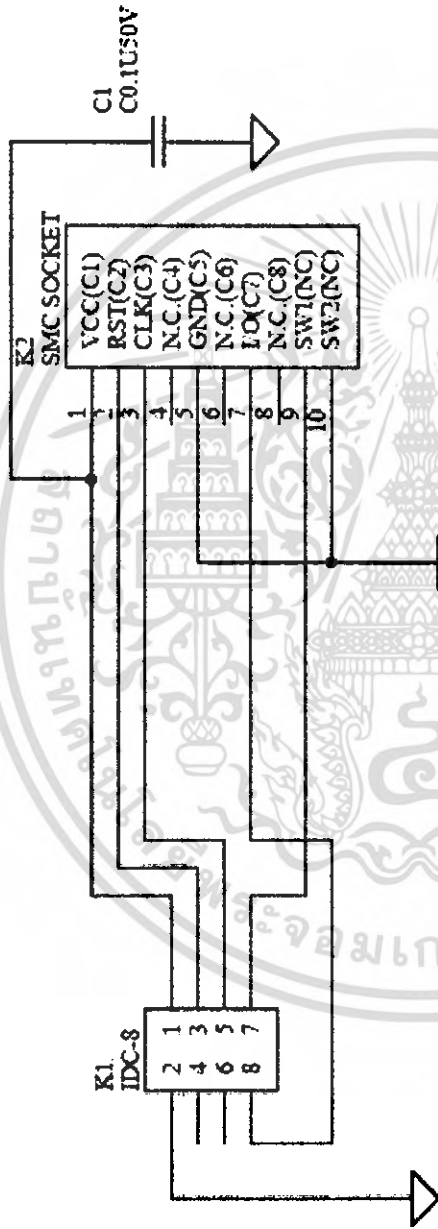
2. วงจรแสดงของ 7-segment



รูปที่ 3.6 วงจรแสดงการทำงานของ 7-segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

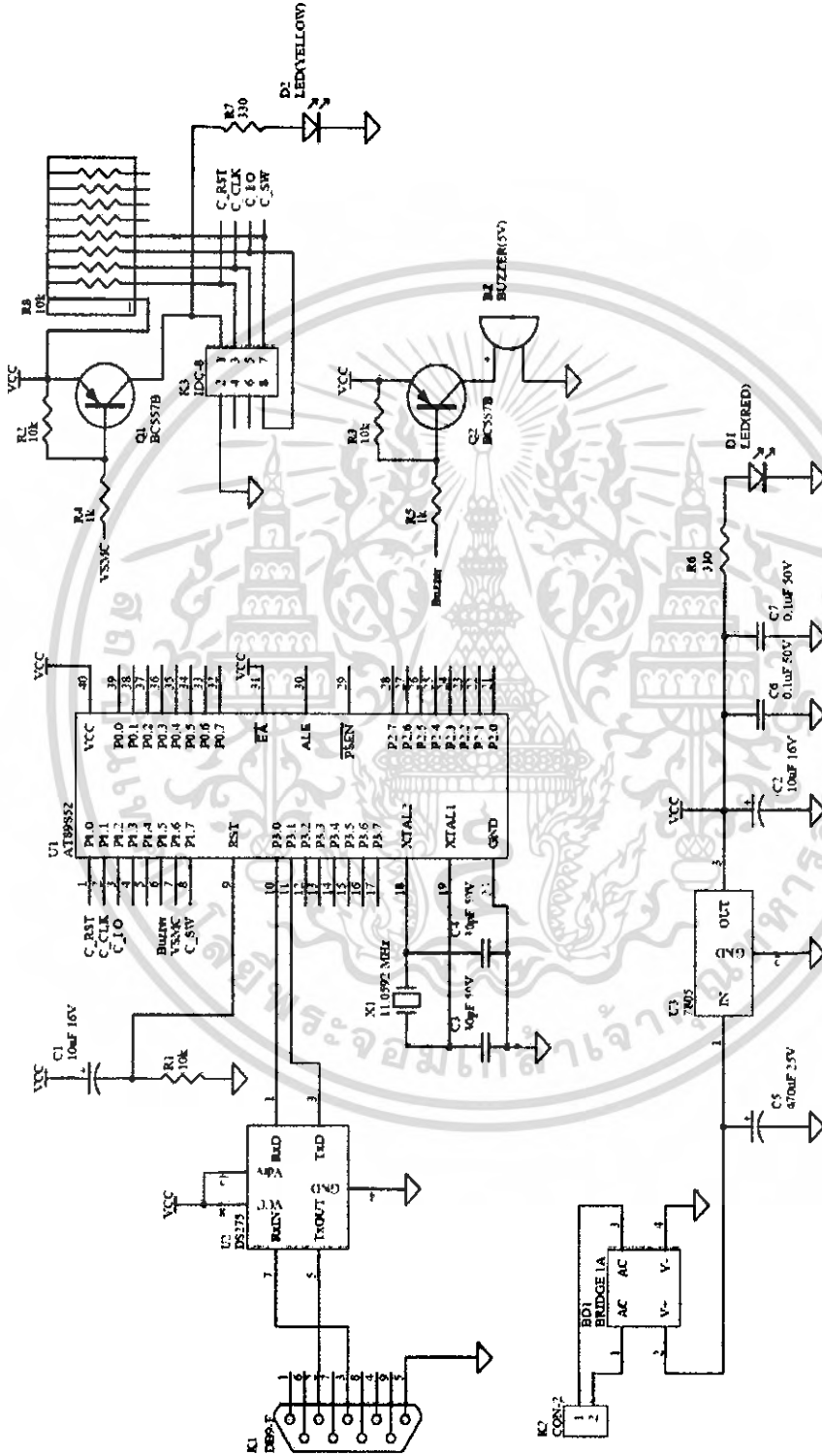
3. วงจรการทำงานของหัวอ่านบัตร Smart card



รูปที่ 3.7 วงจรแสดงการทำงานของหัวอ่านบัตร Smart Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. วงจรของชุดอ่านเขียนบัตร Smart card

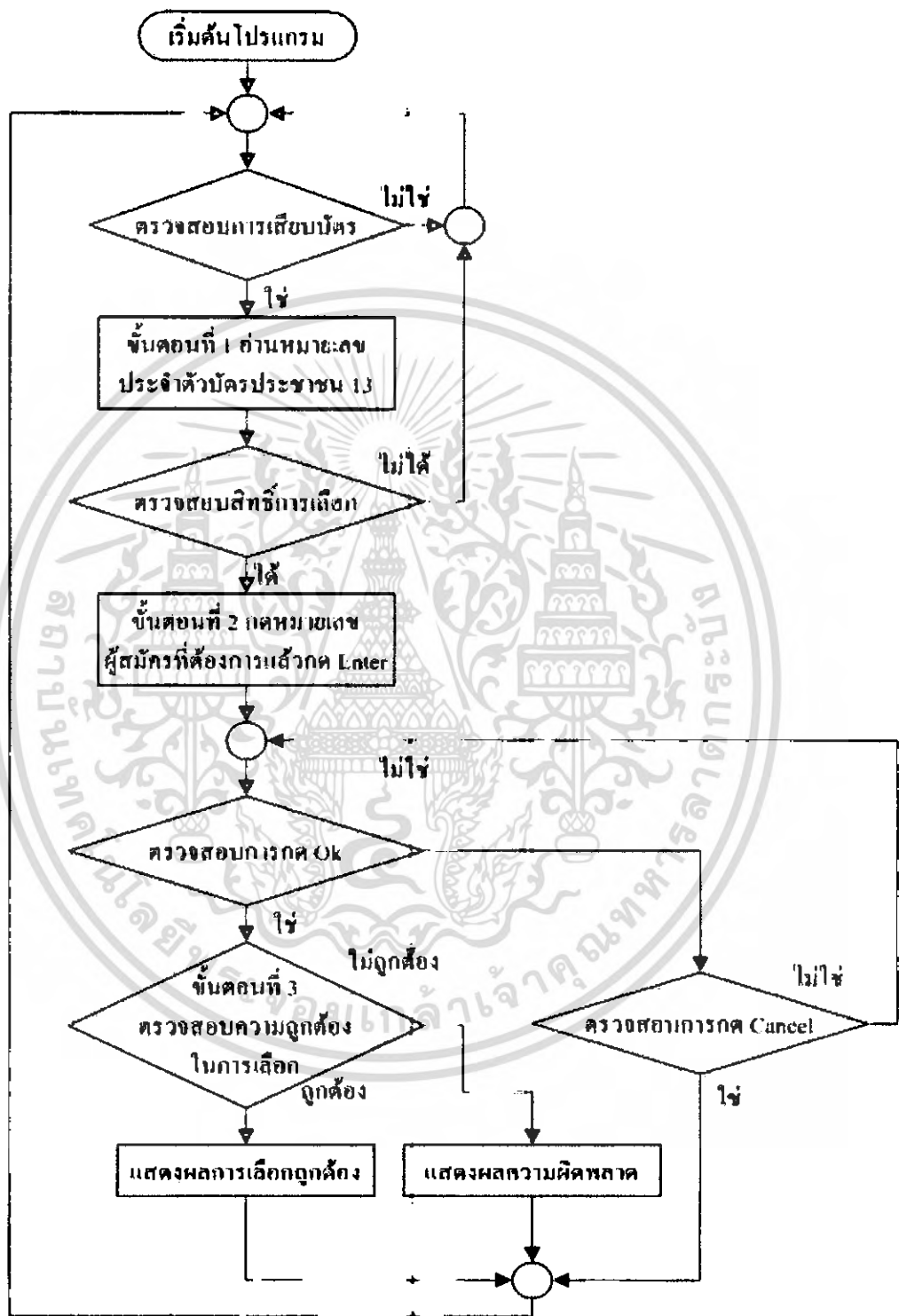


รูปที่ 3.8 แกะวงจรของชุดการอ่านเขียนบัตร Smart card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบ Flowchart แสดงการทำงาน

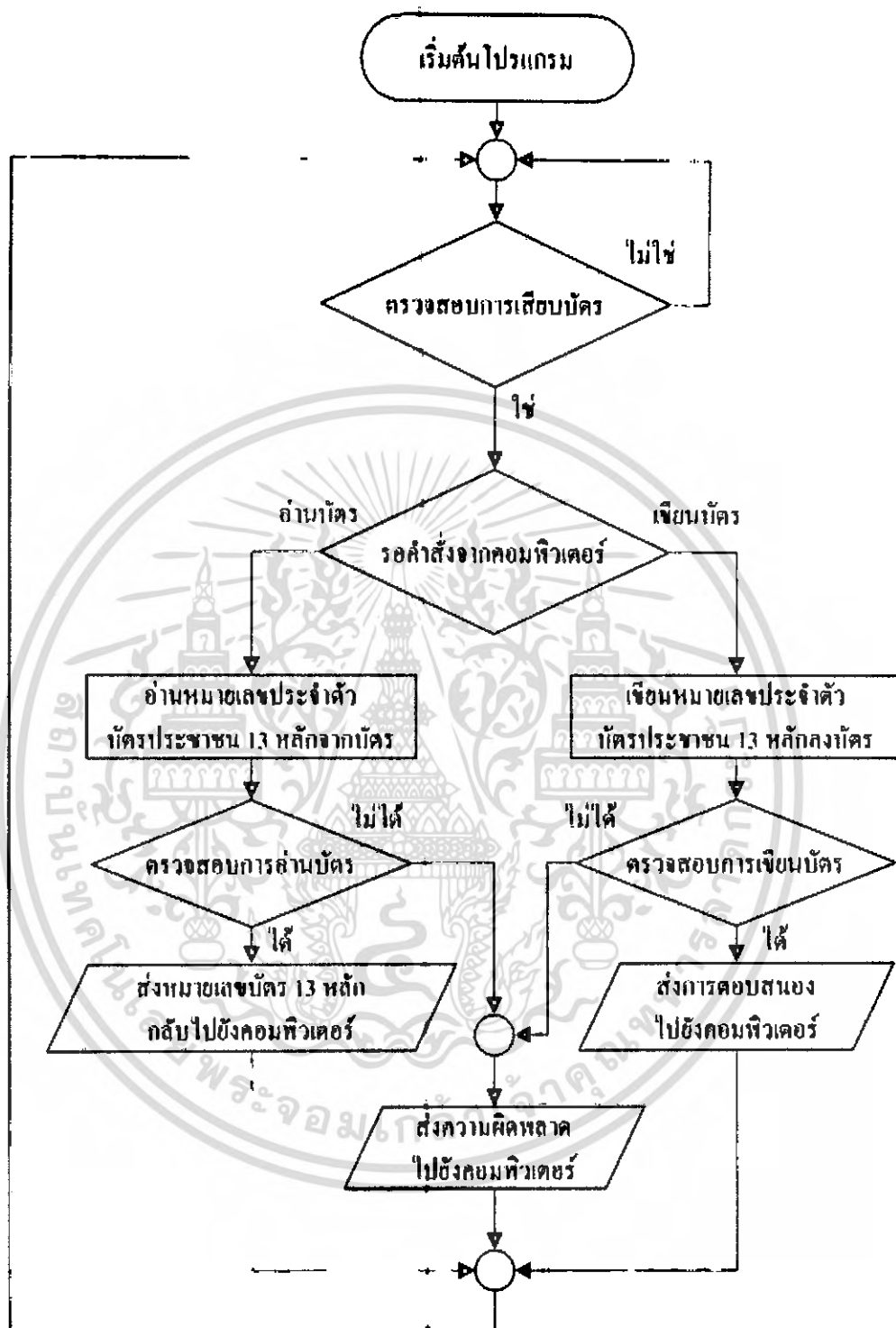
1. เครื่องลงคะแนน



รูปที่ 3.9 แสดง Flowchart การทำงานของเครื่องลงคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

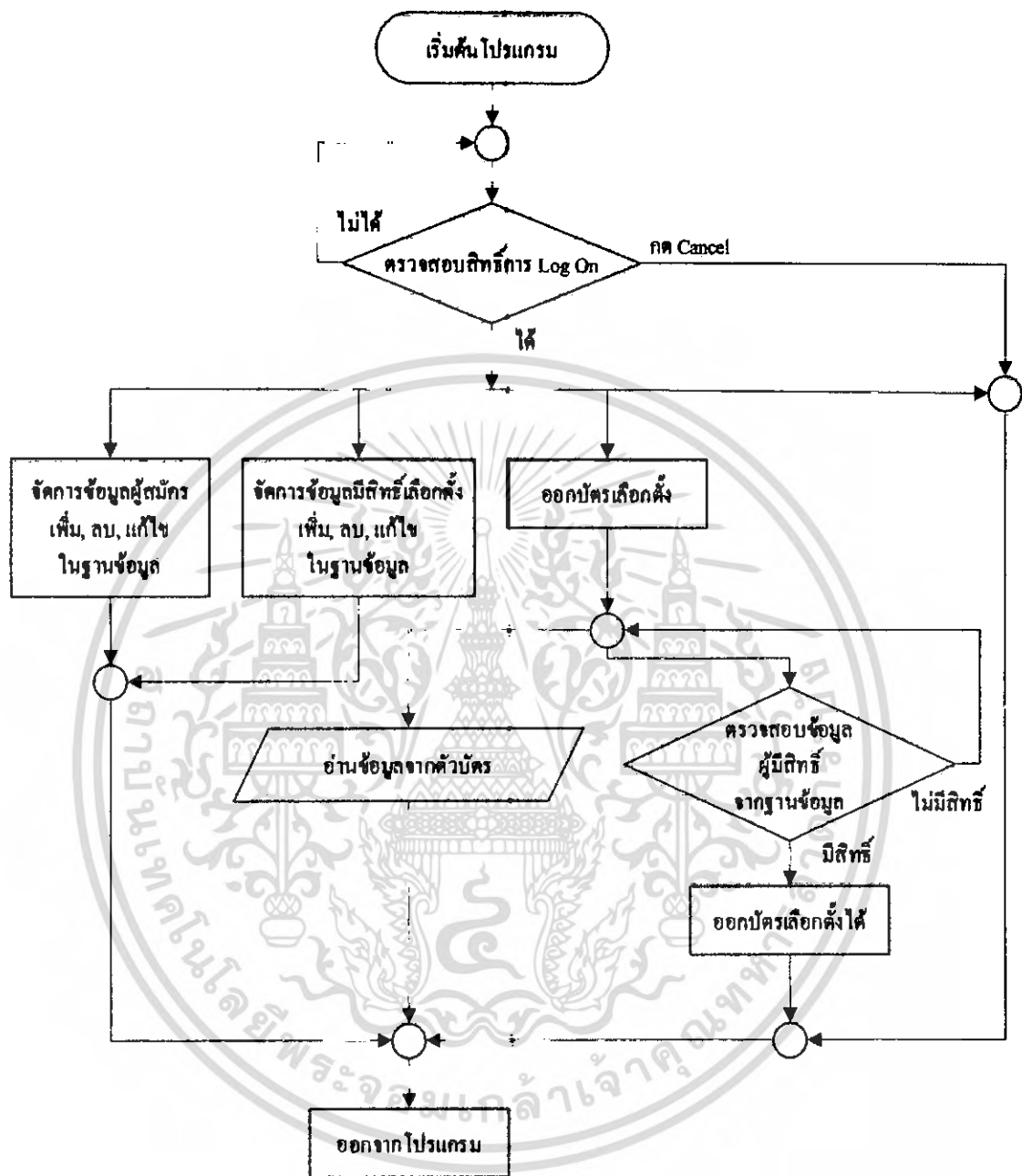
2. เครื่องอ่านเขียนบัตร



รูปที่ 3.10 แสดงFlowchart การทำงานของเครื่องอ่านเขียนบัตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

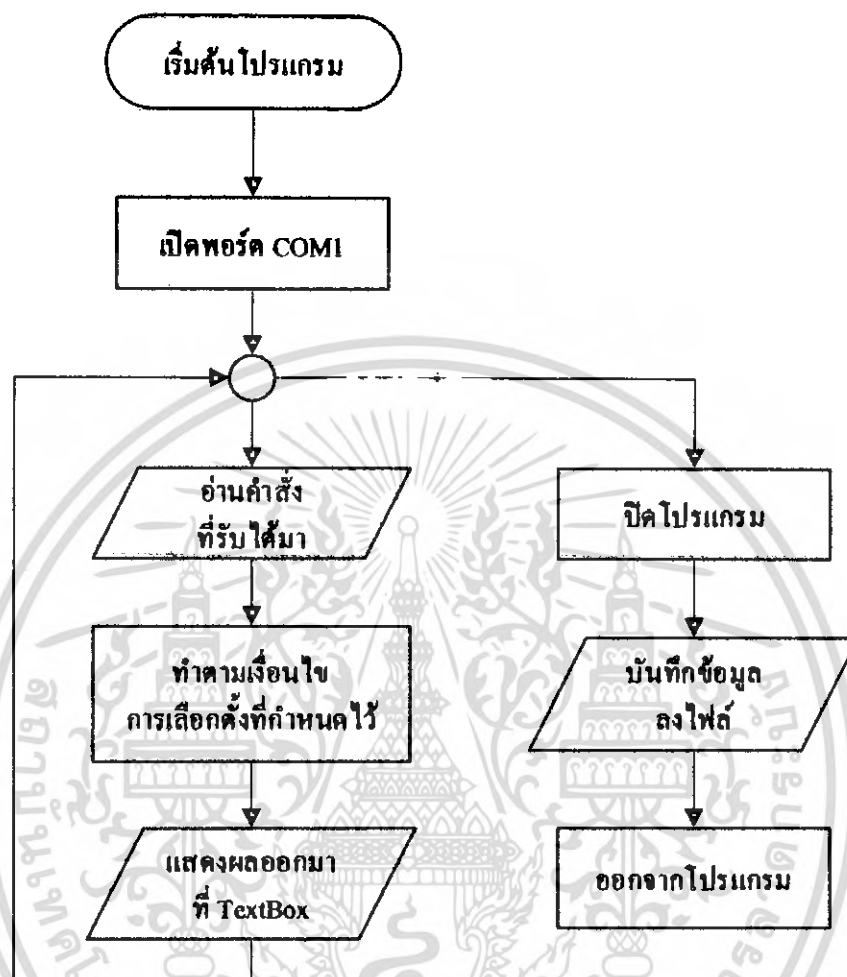
3. โปรแกรม E-Vote Program



รูปที่ 3.11 Flowchart โปรแกรม E-Vote Program

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

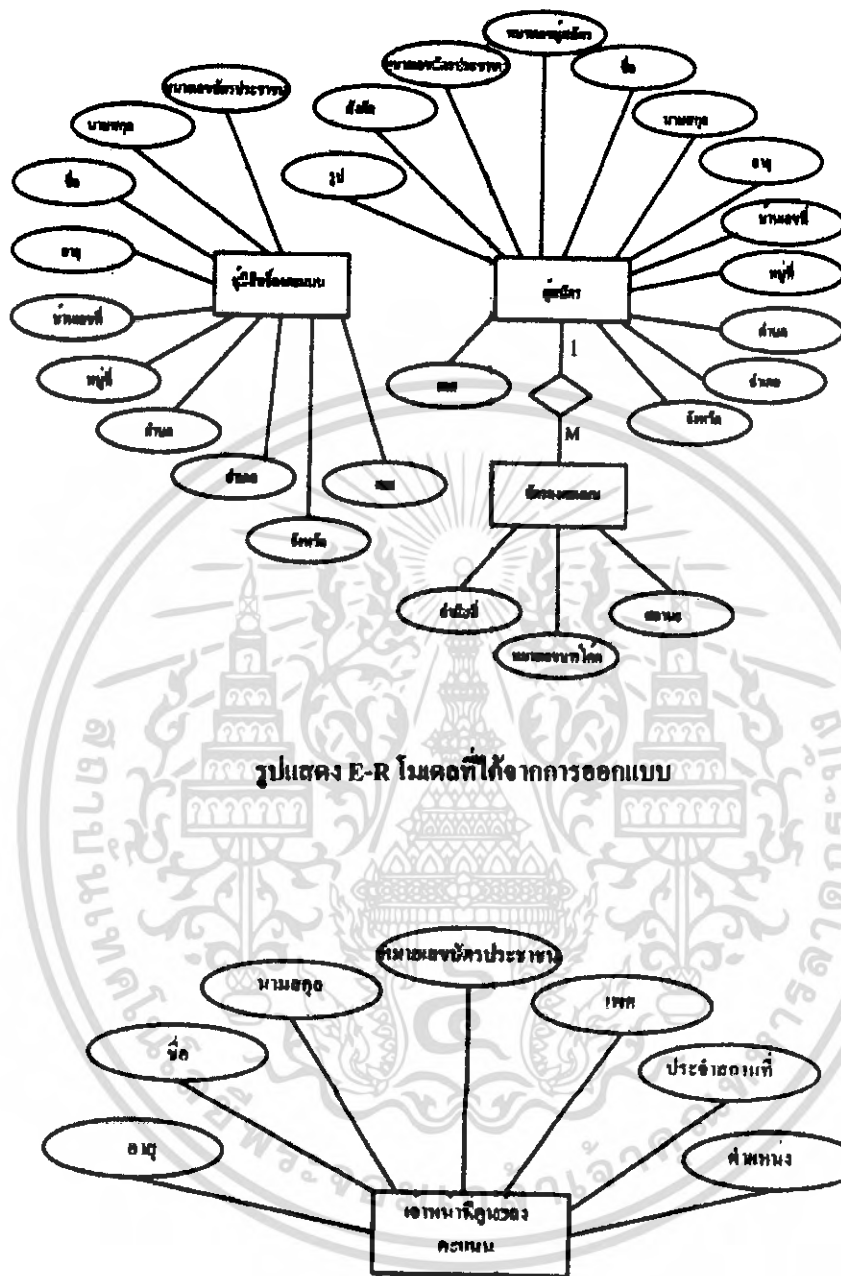
4. โปรแกรม E-Vote Machine



รูปที่ 3.12 Flowchart โปรแกรม E-Vote Machine

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบฐานข้อมูล



รูปที่ 3.13 แสดง E-R โมเดลที่ได้จากการออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ได้จากการออกแบบด้วย ER-MODE

ตารางที่ 3.1 Applicant

| ap_id | ap_idcard | ap_name | ap_surname | ap_address | ap_NOG | ap_locality | ap_county | ap_province | ap_phone | ap_party | ap_score |
|-------|-----------|---------|------------|------------|--------|-------------|-----------|-------------|----------|----------|----------|
| | | | | | | | | | | | |

ตารางที่ 3.2 OfficeGroup

| off_index | off_user | off_password | off_idcard | off_name | off_surname | off_station | off_position |
|-----------|----------|--------------|------------|----------|-------------|-------------|--------------|
| 1 | admin | admin | admin | admin | admin | admin | admin |
| 2 | evil | evil | evil | evil | evil | evil | evil |
| * | | | | | | | |

ตารางที่ 3.3 Usevote

| vol_index | vol_idcard | vol_name | vol_surname | vol_address | vol_NOG | vol_locality | vol_county | vol_province | vol_state | vol_complete |
|-----------|------------|----------|-------------|-------------|---------|--------------|------------|--------------|-----------|--------------|
| | | | | | | | | | | |

3.5 การรักษาความปลอดภัยในระบบเลือกตั้งผ่านเครือข่ายคอมพิวเตอร์

การรักษาความปลอดภัยในระบบเลือกตั้งผ่านเครือข่ายคอมพิวเตอร์ จะต้องครอบคลุมถึงเงื่อนไขดังต่อไปนี้

1. ระบบจะต้องสามารถตรวจสอบได้ว่าบัตรเลือกตั้งที่ได้รับจากหน่วยเลือกตั้งนั้น มาจากผู้มีสิทธิเลือกตั้งมาตามกฎหมาย
2. ระบบจะต้องรักษาความลับของบัตรเลือกตั้ง โดยจะต้องป้องกันไม่ให้รู้ว่าบัตรเลือกตั้งนี้ ถูกเลือกโดยผู้มีสิทธิเลือกตั้งชื่ออะไร
3. ระบบต้องสามารถยืนยันได้ว่า จะต้องไม่มีบุคคลที่ 3 หรือซอฟต์แวร์คอมพิวเตอร์ที่สามารถเปลี่ยนแปลงข้อมูลในบัตรเลือกตั้งได้
4. ระบบจะต้องสามารถตรวจสอบการปลอมแปลง ความผิดพลาดและความเสียหายของบัตรเลือกตั้ง ที่ได้รับได้
5. ระบบจะต้องป้องกันมิให้ผู้มีสิทธิเลือกตั้ง สามารถทำการเลือกตั้งได้หลายครั้ง โดยผู้มีสิทธิเลือกตั้งหนึ่งคน จะสามารถรับบัตรเลือกตั้งได้เพียงใบเดียว และจะต้องทำการเลือกตั้งได้เพียงหนึ่งครั้งเท่านั้น
5. ระบบจะต้องป้องกันมิให้ซอฟต์แวร์คอมพิวเตอร์ใด สามารถทำลายความเป็นส่วนตัวของผู้มีสิทธิเลือกตั้งได้

3.6 การออกแบบการส่งข้อมูลโดยใช้ IP Security & VPN

การออกแบบใช้งาน IPSec นั้น เราอาจใช้ IPSec ในการเชื่อมต่อระหว่างสำนักงาน สาขา (Branch Office) โดยผ่านเครือข่ายอินเทอร์เน็ต หรือ ใช้ในการทำความเชื่อมต่อระยะไกล (Remote Access) ผ่านเครือข่ายอินเทอร์เน็ต หรือ ใช้ในการเชื่อมต่อระหว่างเครือข่ายแบบ Intranet และ Extranet ระหว่างองค์กร หรือใช้ในการสื่อสารแบบ Electronic Commerce ทั้งนี้เนื่องจาก IPSec นั้น สามารถทั้งด้าน Authentication และ Encryption ในทุกๆการสื่อสารที่มีพื้นฐานบนระดับ IP ได้แสดงรูปแบบการใช้งานปกติของ IPSec โดยการสื่อสารภายในวง LAN แต่ละวงจะเป็นการสื่อสารตามปกติแต่เมื่อการสื่อสารได้ก้าวข้ามออกไปภายนอก ไม่ว่าจะเป็นเครือข่ายแบบ Private หรือ Public ก็ตาม ก็จะมีการใช้ IPSec โดยการนำ IPSec มาใช้นี้ จะเริ่มที่ Router , Firewall หรืออุปกรณ์เครือข่ายที่ทำหน้าที่เป็นจุดเอก (Gateway) ของเครือข่าย โดยจะมีการเข้ารหัสข้อมูล แล้วจึงส่งออกไป และเมื่อถึงปลายทางก็จะถอดรหัสออกมา ซึ่งการทำงานทั้งหมดนี้ จะเกิดขึ้นโดยที่เครื่องคอมพิวเตอร์ไม่มีส่วนรับทราบเลยนอกจากนั้น IPSec ยังสามารถใช้งานในกรณีที่ใช้การเชื่อมต่อแบบ Dial-Up ได้อีกด้วยไม่จำเป็นที่จะต้องเข้ามาที่หน่วยงานนั้นๆ โดยตรงหรือเป็นการเชื่อมต่อผ่านทางอินเทอร์เน็ต

3.7 ประโยชน์ของ IPSec

1. เมื่อมีการนำ IPSec มาใช้ที่ Firewall หรือ Router จะทำให้มีระบบความปลอดภัยที่แข็งแกร่งที่สามารถใช้ได้กับทุกการสื่อสาร โดยการสื่อสารภายในจะไม่มี Overhead ของ IPSec
 2. เมื่อมีการใช้ IPSec กับ Firewall ทุกการสื่อสารจะไม่สามารถข้าม IPSec ได้ เพราะการสื่อสารกับภายนอกต้องใช้ IP ซึ่งหมายความว่าต้องใช้ IPSec ด้วย และเนื่องจาก Firewall เป็นเพียงจุดเดียวที่เชื่อมต่อ กับภายนอก ดังนั้นการติดต่อกันภายในและภายนอกก็จะต้องทำโดยผ่าน IPSec เท่านั้น
 3. เนื่องจาก IPSec ทำงานอยู่ได้ TCP และ UDP ดังนั้นแอปพลิเคชันที่ทำงานบน TCP และ UDP จึงต้องทำงานผ่าน IPSec ไปด้วย และไม่ต้องรับรู้ถึงการมีอยู่ของ IPSec ดังนั้นโปรแกรมต่างๆ ก็ไม่ต้องเขียนขึ้นมาใหม่
 4. การทำงานของ IPSec ไม่กระทบกับผู้ใช้ โดยผู้ใช้จะไม่รับรู้ถึงการมีอยู่ของ IPSec เลยดังนั้นจึงไม่ต้องเสียค่าใช้จ่ายในการสอนผู้ใช้
 5. IPSec สามารถจะสร้างความปลอดภัยในระดับผู้ใช้ได้ ซึ่งเป็นผลดีที่ทำให้สามารถจะใช้ระบบถึงผู้ใช้แต่ละคนที่เข้ามาใช้งานจากระยะไกล
- นอกจากนั้น IPSec ยังช่วยให้การทำงาน Routing Protocol มีความปลอดภัยมากยิ่งขึ้น เพราะช่วยให้การทำงานต่าง ๆ ไม่ว่าจะเป็น Router Advertisement, Neighbor Advertisement หรืออื่น ๆ สามารถใช้ความสามารถของ IPSec ในการเข้ารหัสข้อมูล และพิสูจน์ถึง Router จริง ๆ ได้

บทที่ 4

ผลการทดลอง

การทดลองของโครงงานงานมีจะแบ่งออกเป็น 2 ส่วน คือ

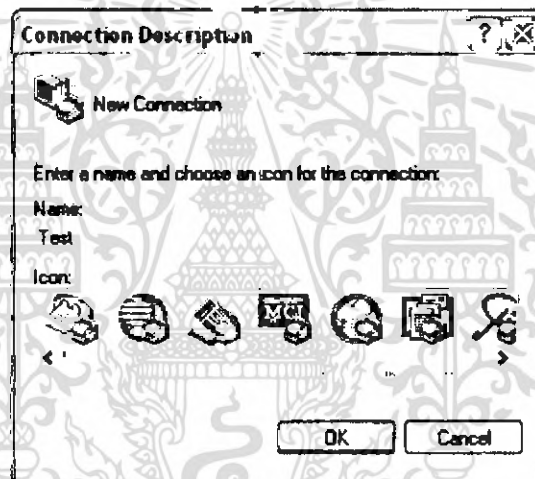
4.1 การทดลองส่วนของ Hardware

4.2 การทดลองส่วนของ Software

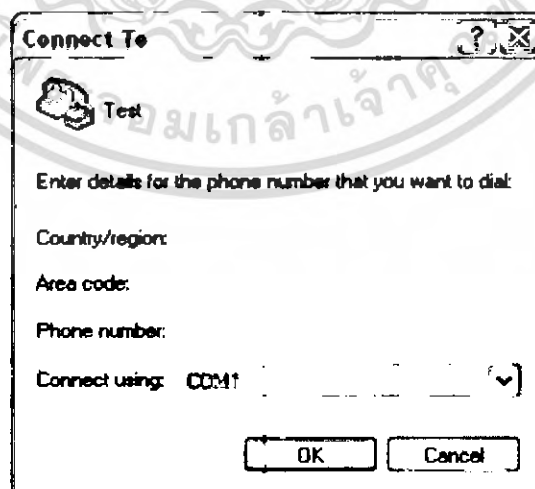
4.1 การทดลองส่วนของ Hardware

4.1.1 ส่วนของขั้นตอนการทำงานบนเครื่องเดสก์ทอป

1. คลิกที่ปุ่ม Start แล้วเปิดโปรแกรม HyperTerminal โดยไปที่
Programs>>Accessories>>Communications>>HyperTerminal
2. ตั้งชื่อ

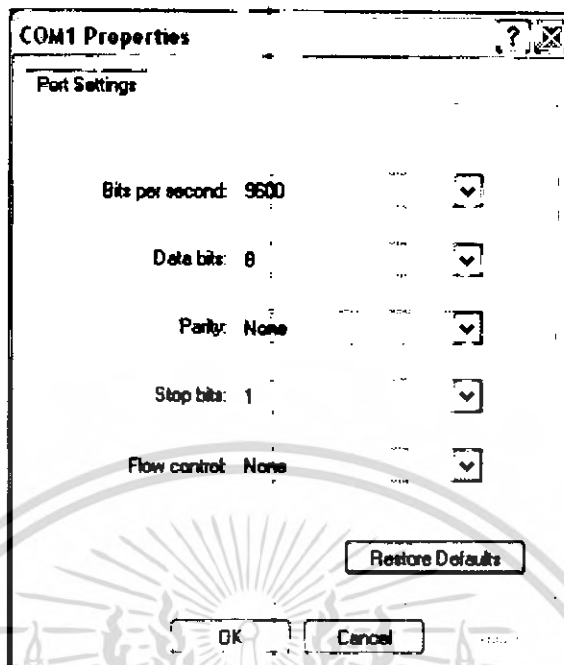


3. เลือกพอร์ต

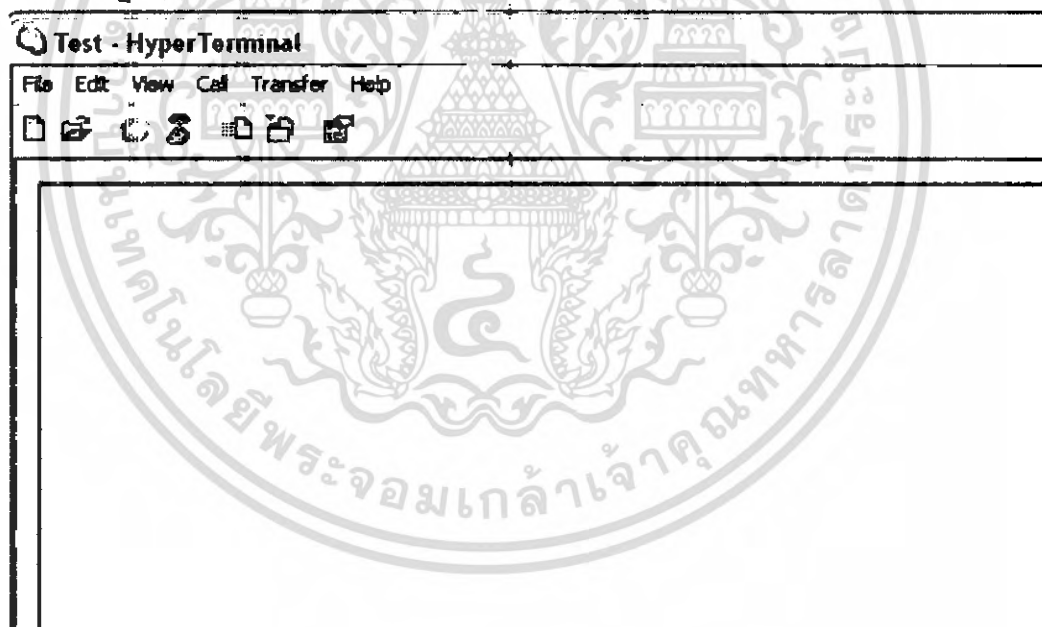


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตั้งค่าพารามิเตอร์ตามนี้



5. จะปรากฏหน้าต่างตามนี้



6. ขั้นตอนที่ 1

ทำการเสียบบัตรสมาร์ตการ์ดจะปรากฏไฟสีส้มขึ้น และจะส่งหมายเลขบัตรประจำตัวประชาชน 13 หลักที่เสียบไปมายังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๐๘ . ๘ ๐๘ ๘

:01234567890123

:0XXXXXXXXXXXX

ข้อมูลที่ส่งกลับมาอธิบายได้ดังนี้ :0 คือบอกเป็นการทำกระบวนการขั้นตอนแรกที่ติดต่อกับคอมพิวเตอร์

ส่วน **XXXXXXXXXXXX** คือหมายเลขบัตรประจำตัวประชาชน 13 หลัก ซึ่งก็คือ 1234567890123 นั่นเอง

ซึ่งต้องการตรวจสอบกับฐานข้อมูลที่คอมพิวเตอร์ว่าหมายเลขบัตรประจำตัวประชาชนนี้สามารถเลือกตั้งได้หรือไม่

ให้พิมพ์ :1 ตามด้วย Enter ลงไปที่คอมพิวเตอร์เพื่อเป็นการตอบกลับว่าหมายเลขบัตรประจำตัวประชาชนนี้สามารถเลือกตั้งได้ ก็จะสามารทำขั้นตอนที่ 2 ของการเลือกตั้งได้ **LED ดวงที่ 1 จะติด**

ถ้าสมมุติว่าบัตรนี้ไม่สามารถเลือกตั้งได้ก็ ให้พิมพ์ :0 ตามด้วย Enter ลงไปที่คอมพิวเตอร์จะทำให้เครื่องกลับสู่สถานะเริ่มต้นใหม่ ต้องดึงบัตรออกจากหัวอ่าน

7. ขั้นตอนที่ 2

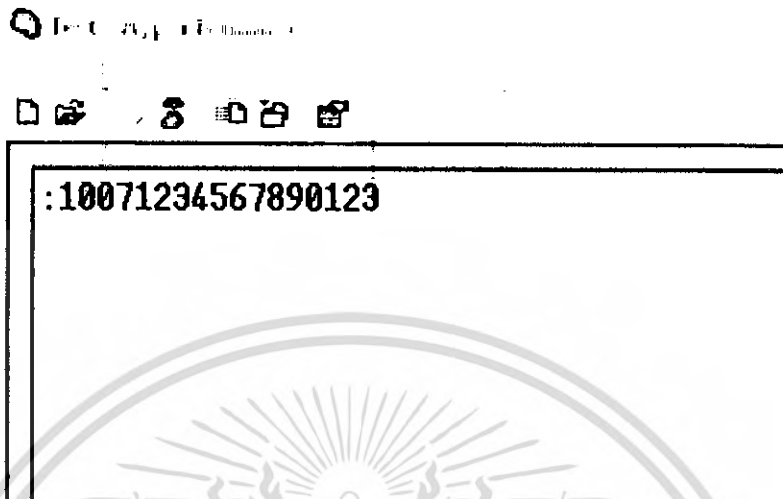
ที่ชุดเครื่องเลือกตั้ง ให้กดหมายเลขผู้สมัครที่ต้องการเลือกลงไป ตามกดปุ่ม Enter เมื่อต้องการเลือกผู้สมัครคนนี้ เมื่อกดแล้วก็ไม่สามารถกดหมายเลขเพิ่มเติมได้นอกจากกด Ok หรือ Cancel เท่านั้น

-ในขั้นตอนนี้ถ้าต้องการยกเลิกหมายเลขที่ผู้สมัครและต้องการเลือกใหม่ให้กดปุ่ม Cancel ให้เลย

-สมมุติว่าต้องการเลือกผู้สมัครเบอร์ 7 ให้กดหมายเลข 7 แล้วกด Enter **ที่ชุดเครื่องเลือกตั้ง LED ดวงที่ 2 จะติด**

8. ขั้นตอนที่ 3

กดปุ่ม OK ที่ชุดเครื่องเลือกตั้ง เมื่อต้องการเลือกผู้สมัครคนนี้จริงๆ ที่ชุดเครื่องเลือกตั้งจะส่งข้อมูลกลับมาดังนี้ตามรูป



:1ZZZXXXXXXXXXXXXXX

ข้อมูลที่ส่งกลับมาอธิบายได้ดังนี้ :1 คือบอกเป็นการทำกระบวนการขั้นตอนสุดท้ายที่ติดต่อกับคอมพิวเตอร์

ส่วน ZZZ คือหมายเลขผู้สมัคร ที่ทำการเลือกในขั้นตอนที่ 2 ได้ทำการเลือกหมายเลข 7 ก็คือ 007

ส่วน XXXXXXXXXXXXXXX คือหมายเลขบัตรประจำตัวประชาชน 13 หลัก ซึ่งก็คือ 1234567890123 นั่นเอง

การส่งข้อมูลนี้มาเพื่อทำการเลือกผู้สมัครคนนี้โดยตรวจสอบว่าสามารถเลือกผู้สมัครคนนี้ได้หรือไม่

สมมุติถ้าสามารถเลือกได้ ให้พิมพ์ :1 ตามด้วย Enter ลงไปที่คอมพิวเตอร์ LED ดวงที่ 3 จะติด

และ LED สีเหลืองจะดับ ถ้าดึงบัตรออก LED สีเขียวทั้ง 3 ดวงจะดับ

สมมุติถ้าไม่สามารถเลือกได้ ให้พิมพ์ :1 ตามด้วย Enter ลงไปที่คอมพิวเตอร์ LED สีเหลืองจะ

ดับ และ LED สีเขียวทั้งหมดจะดับ

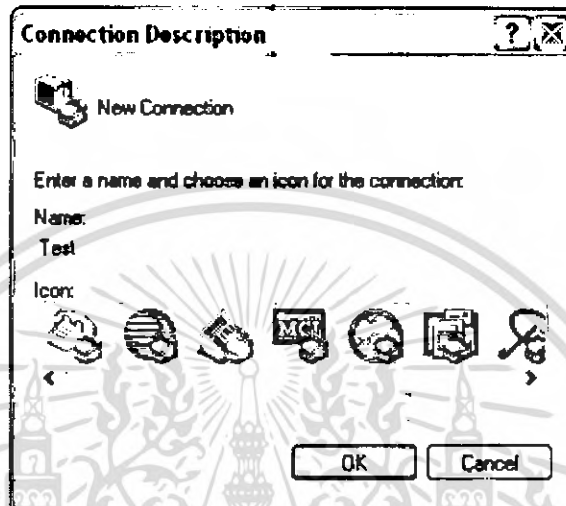
4.1.2 ส่วนของขั้นตอนการทำงานของการอ่านเขียนบัตรสมาชิกการ์ด

1. ชุดอ่านเขียนบัตรสมาชิกการ์ด

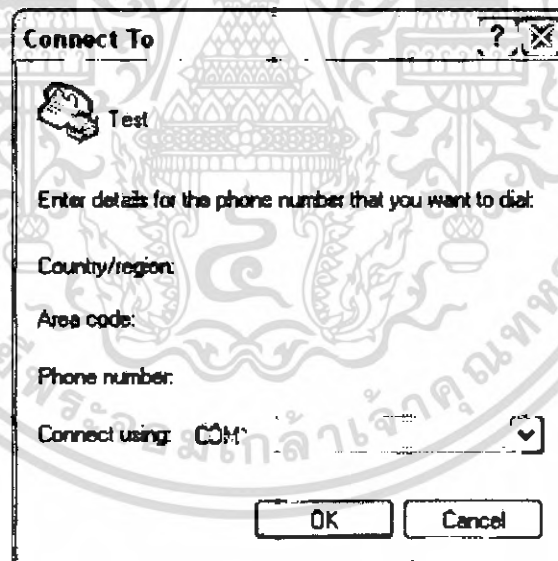
1.1 คลิกที่ปุ่ม Start แล้วเปิด โปรแกรม HyperTerminal โดยไปที่

Programs>>Accessories>>Communications>>HyperTerminal

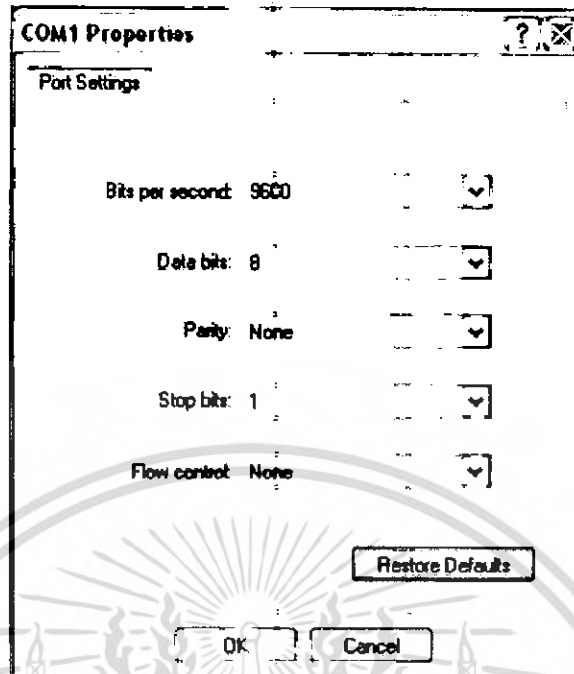
1.2 ตั้งชื่อ



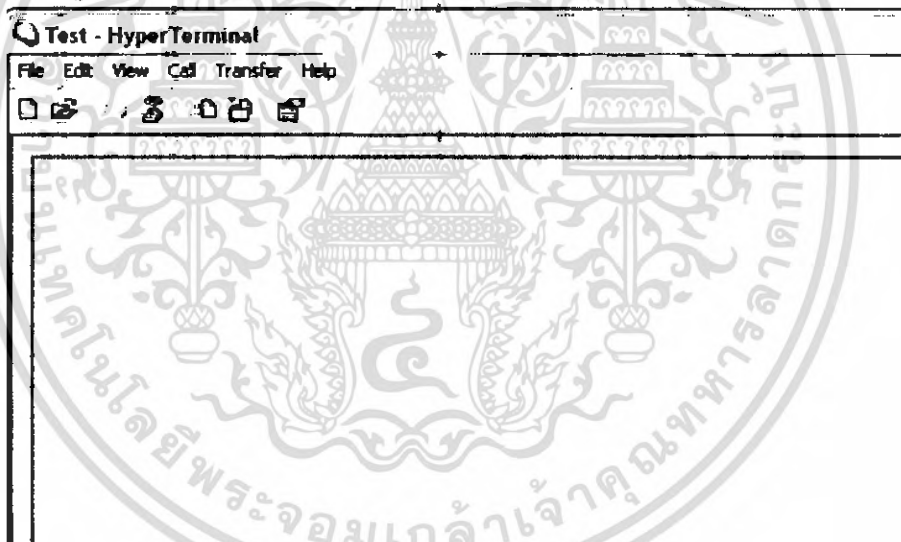
1.3 เลือกพอร์ต



1.4 ตั้งค่าพารามิเตอร์ตามนี้



1.5 จะปรากฏหน้าต่างตามนี้



1.6 ทำการเทียบบัตรสมาร์ตการ์ดจะปรากฏไฟสีเขียวขึ้น

1.7 ถ้าต้องการเขียนหมายเลขบัตรประจำตัวประชาชน 13 หลักลงบัตรสมาร์ตการ์ด ให้พิมพ์คำสั่งดังนี้


:1XXXXXXXXXXXXX ตามด้วย Enter

หมายเหตุ XXXXXXXXXXXX คือหมายเลขบัตรประจำตัวประชาชน 13 หลักเป็นตัวเลขเท่านั้น

ตัวอย่างพิมพ์ สมมุติ 1234567890123 คือหมายเลขบัตรประจำตัวประชาชน 13 หลัก

:11234567890123 ตามด้วย Enter

ถ้าสามารถเขียนหมายเลขบัตรประจำตัวประชาชน 13 หลักลงบัตร ได้จะส่งคำนี้ตามรูปกลับมายังคอมพิวเตอร์

 Print Hypertext Location

:1

ถ้าไม่สามารถเขียนหมายเลขบัตรประจำตัวประชาชน 13 หลักลงบัตร ได้จะส่งคำนี้ตามรูปกลับมายังคอมพิวเตอร์

 Print Hypertext Location

:9

1.8 ถ้าต้องการอ่านหมายเลขบัตรประจำตัวประชาชน 13 หลักจากบัตรสมัครพรรค ให้พิมพ์คำสั่งดังนี้

:0 ตามด้วย Enter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าสามารถอ่านหมายเลขบัตรประจำตัวประชาชน 13 หลักจากบัตร ได้จะส่งคำนี้ตามรูปกลับมายังคอมพิวเตอร์

0 : 1 2 3 4 5 6 7 8 9 0 1 2 3

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ ๐ ๑ ๒ ๓

:01234567890123

หมายเหตุ 1234567890123 คือหมายเลขบัตรประจำตัวประชาชน 13 หลัก ที่ได้ทำการเขียนลงไป
ในบัตร อาจเป็นคำอื่นก็ได้ ที่ได้ทำการเขียนลงไป

ถ้าไม่สามารถอ่านหมายเลขบัตรประจำตัวประชาชน 13 หลักจากบัตร ได้จะส่งคำนี้ตามรูปกลับมา
ยังคอมพิวเตอร์

0 : 1 2 3 4 5 6 7 8 9 0 1 2 3

๐ ๑ ๒ ๓ ๔ ๕ ๖ ๗ ๘ ๙ ๐ ๑ ๒ ๓

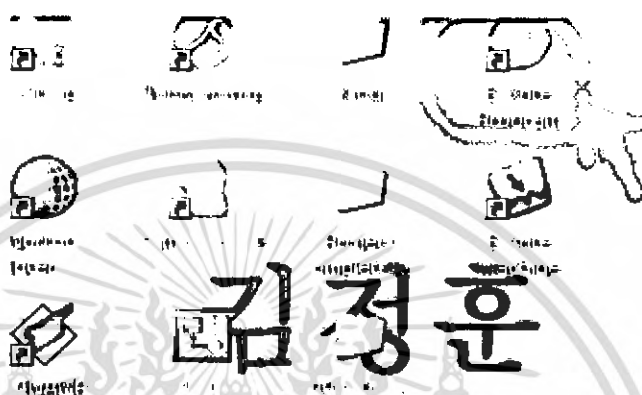
:9

4.2 การทดลองส่วนของ Software

4.2.1 ฟังก์ชันสำหรับผู้ดูแลการเลือกตั้ง

ฟังก์ชันสำหรับผู้ดูแลการเลือกตั้งจะเป็นฟังก์ชันเพื่อใช้ในการดูแลการเลือกตั้งต่างๆ เช่น การเพิ่ม การแก้ไข การลบ ข้อมูลเกี่ยวกับการเลือกตั้ง ผู้มีสิทธิเลือกตั้ง บัตรเลือกตั้ง และเจ้าหน้าที่ประจำหน่วยเลือกตั้ง

4.2.1.1 ฟังก์ชันที่ใช้ในการเข้าระบบ



รูป แสดงโปรแกรมที่เจ้าหน้าที่ใช้ในการล็อกอิน
เมื่อเข้าไปที่โปรแกรมนี้จะปรากฏหน้าต่างนี้ขึ้นมา ซึ่งหน้าต่างนี้จะใช้ในการล็อกอินเข้าสู่ระบบ

รูปที่ 4.1 หน้าต่างการล็อกอินเข้าสู่ระบบ

ผู้ดูแลการเลือกตั้งจะต้องป้อน ชื่อ และรหัสผ่านให้ถูกต้องจึงจะสามารถเข้าสู่ระบบได้

4.2.1.2 ฟังก์ชันที่ใช้เป็นฐานข้อมูลสำหรับเจ้าหน้าที่ประจำหน่วยเลือกตั้ง ประกอบด้วย

3 ส่วน คือ

1. ข้อมูลของผู้สมัคร(Applicant)
2. ข้อมูลของเจ้าหน้าที่ที่ดูแลในการลงคะแนน (Office Group)
3. ข้อมูลของผู้มาลงคะแนน(Uservote)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง : ข้อมูลของผู้สมัคร

| ap id | ap idcard | ap name | ap surname | ap address | ap HOG | ap locality | ap county | ap province | ap phone | ap party | ap score |
|-------|----------------|--------------|------------|------------------|---------------|-------------|-----------|-------------|------------|----------|----------|
| 1 | 1234567894567 | นาย ธน | ทองดี | 58/22ต.นาดีบึง 1 | ก.ต.ต.บึง 32 | ต.ต.พริก | ไทย | กอน. 10230 | 02-6441882 | รักไทย | |
| 2 | 1234567894567 | นาย ธา | ทองดี | 33/1129 ม.10 | ก.ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10230 | 02-9114657 | เพื่อไทย | |
| 3 | 12456789456232 | นาย ธีร | ธีรชาติ | 237 ซ.พริก 24 | ต.พริก | ต.พริก | ไทย | กอน. 10950 | 02-4546123 | เพื่อไทย | |
| 4 | 1245752365441 | นางสาว สัน | สุขุม | 757/10 ซ.ปตจ | ก.ต.ต.พริก 2 | บ.ต.พริก | ไทย | กอน. 10120 | 02-4561237 | ชาติไทย | |
| 5 | 1276668866123 | นาย มาลี | ทองดี | 52 ม.บึง 1 | ก.ต.ต.พริก 2 | บ.ต.พริก | ไทย | กอน. 10230 | 02-3456795 | ก้าวไกล | |
| 6 | 1234547895513 | นาย สมศักดิ์ | ทองดี | 121 ซ.พริก 17 | ต.พริก | ต.พริก | ไทย | กอน. 10900 | 02-5463978 | เพื่อไทย | |
| 7 | 1257895456267 | นาย สมาน | ใจดี | 40/465 ม.10 | ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10230 | 02-3264597 | เพื่อไทย | |
| 8 | 1245668732269 | นาย สมิน | มากมายน | 511/787 | ก.ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10700 | 02-6547894 | เพื่อไทย | |
| 9 | 1234567894564 | นางสาว สัน | ทองดี | 146 | ก.ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10110 | 02-4568925 | เพื่อไทย | |
| 10 | 1234567899554 | นาย ธน | สมใจ | 195 ซ.พริก 17 | ต.พริก | ต.พริก | ไทย | กอน. 10110 | 02-4386788 | ก้าวไกล | |
| 11 | 1245786665125 | นางสาว สัน | ทองดี | 363/2 หมู่ 4 | บ.ต.พริก | บ.ต.พริก | ไทย | กอน. 10160 | 02-3668958 | เพื่อไทย | |
| 12 | 1234568756829 | นางสาว พุฒิ | ใจดี | 36 ซ.พริก 8 | ก.ต.ต.พริก | ต.ต.พริก | ไทย | กอน. 10300 | 02-4587455 | เพื่อไทย | |
| 13 | 1254784895611 | นาย สมสุข | ใจดี | 100/307 ซ.8 | บ.ต.พริก | บ.ต.พริก | ไทย | กอน. 10250 | 02-2668887 | เพื่อไทย | |
| 14 | 1234567844621 | นาย สัน | ใจดี | 81 ต.พริก 42 | ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10310 | 02-3548792 | เพื่อไทย | |
| 15 | 1245678112245 | นาง สมใจ | ใจดี | 107 ม.บึง 1 | ก.ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10240 | 02-7894561 | เพื่อไทย | |
| 16 | 1256487945661 | นางสาว สัน | ทองดี | 1581 หมู่ 10 | ก.ต.ต.พริก 15 | บ.ต.พริก | ไทย | กอน. 10230 | 02-4567843 | เพื่อไทย | |
| 17 | 1234567815551 | นาย สัน | ใจดี | 1918 ต.พริก | บ.ต.พริก | บ.ต.พริก | ไทย | กอน. 10600 | 02-1564782 | เพื่อไทย | |
| 18 | 1224789452345 | นาง พุฒิ | ใจดี | 573/125 | ก.ต.ต.พริก 39 | บ.ต.พริก | ไทย | กอน. 10310 | 02-1234567 | เพื่อไทย | |
| 19 | 124578961221 | นางสาว ใจ | ใจดี | 12 ซ.พริก 4 | ต.ต.พริก 83 | บ.ต.พริก | ไทย | กอน. 10110 | 02-4587974 | เพื่อไทย | |
| 20 | 1257489632148 | นาย สม | ใจดี | 254/108 | ม.บึง 1 | บ.ต.พริก | ไทย | กอน. 10240 | 02-6668888 | เพื่อไทย | |
| 21 | 1245789666667 | นาง สัน | ใจดี | 8048 | บ.ต.พริก | บ.ต.พริก | ไทย | กอน. 10150 | 02-4784569 | เพื่อไทย | |
| 22 | 1234744554188 | นาย สัน | ใจดี | 230/3 | ต.ต.พริก 23/2 | บ.ต.พริก | ไทย | กอน. 10118 | 02-5441882 | เพื่อไทย | |
| 23 | 1546789452236 | นางสาว ใจ | ใจดี | 200 ต.พริก 49 | ก.ต.ต.พริก | บ.ต.พริก | ไทย | กอน. 10110 | 02-1124578 | เพื่อไทย | |
| 24 | 16457815668956 | นาย ใจ | ใจดี | 1213389 | ก.ต.ต.พริก 94 | บ.ต.พริก | ไทย | กอน. 10310 | 02-1564785 | เพื่อไทย | |
| 25 | 154675522155 | นางสาว ใจ | ใจดี | 440/15 | ก.ต.ต.พริก 13 | บ.ต.พริก | ไทย | กอน. 10600 | 02-1456786 | เพื่อไทย | |
| 26 | 1457525225244 | นาง สม | ใจดี | 74 ม.3 | บ.ต.พริก | บ.ต.พริก | ไทย | กอน. 10210 | 02-4468117 | เพื่อไทย | |
| 27 | 122244414447 | นาย สม | ใจดี | 8 ต.พริก 93 | บ.ต.พริก | บ.ต.พริก | ไทย | กอน. 10110 | 02-7758556 | เพื่อไทย | |
| 999 | No Vote | No Vote | No Vote | No Vote | No Vote | No Vote | No Vote | No Vote | No Vote | No Vote | No Vote |

รูปที่ 1
รูปที่ 4.2 แสดงฐานข้อมูลของผู้สมัคร

ตัวอย่าง : ข้อมูลของเจ้าหน้าที่ที่ดูแลการเลือกตั้ง

| off index | off user | off password | off idcard | off name | off surname | off station | off position |
|-----------|----------|--------------|------------|----------|-------------|-------------|--------------|
| 1 | admin1 | admin | admin | admin | admin | admin | admin |
| 2 | admin2 | admin | admin | admin | admin | admin | admin |
| 3 | admin3 | admin | admin | admin | admin | admin | admin |
| 4 | admin4 | admin | admin | admin | admin | admin | admin |
| 5 | admin5 | admin | admin | admin | admin | admin | admin |
| 6 | admin6 | admin | admin | admin | admin | admin | admin |
| 7 | admin7 | admin | admin | admin | admin | admin | admin |
| 8 | admin8 | admin | admin | admin | admin | admin | admin |
| 9 | admin9 | admin | admin | admin | admin | admin | admin |
| 10 | admin10 | admin | admin | admin | admin | admin | admin |

รูปที่ 4.3 แสดงฐานข้อมูลของเจ้าหน้าที่ที่ดูแลหน่วยลงคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง : ข้อมูลของผู้มาลงคะแนน

1. กรณีที่ผู้มาใช้สิทธิสามารถลงคะแนนได้

| วot index | vot icard | vot name | vot surname | vot address | vot NOG | vot locality | vot county | vot province | vot state | vot complete |
|-----------|---------------|-------------|-------------|-------------------|-------------|--------------|------------|--------------|-----------|--------------|
| 1 | 111111111111 | นาย คิล | ป๋อทอง | 14 ซ.ดอนมู่เตล 1 | เช็กทุ่ง 57 | บางนาท | ไทย | กชน. 10120 | NO | |
| 2 | 111122244444 | นาย อุดม | ทวีโร | 14 ซ.ติงตู่ป๋อป๋อ | สามสมมต | ห้วยทวง | ไทย | กชน. 10320 | NO | |
| 3 | 111122334555 | นาย เดก | โหว | 58/22 ซ.ตงอิม 1 | ก.ตงอิม 32 | จระงัก | ไทย | กชน. 10230 | YES | |
| 4 | 111123355666 | นาย อรทงิม | อิมตง | 67/180 | สทงทัก | อู่จัก | ไทย | กชน. 10900 | NO | |
| 5 | 1112123314454 | นาย ตงม | อู่ทง | 66 ม.ปะ 4 | ก.ทงทง 9 | สทงทง | ไทย | กชน. 10250 | YES | |
| 6 | 1112444777662 | นาย ตงม | อู่ทง | 97/89 ม.ปะ | ตงอิม | บงทง | ไทย | กชน. 10240 | YES | |
| 7 | 1112454556677 | นาย ตงม | อู่ทง | 15 ซ.ตงอิม | ก.ตงอิม | บงทง | ไทย | กชน. 10120 | YES | |
| 8 | 1121455155526 | นางทง อู่ทง | อู่ทง | 4 ม.ตงอิม | ก.ตงอิม | บงทง | ไทย | กชน. 10220 | YES | |
| 9 | 1121477799533 | นาย สทงทง | อู่ทง | 36 ซ. ตงอิม 34 | ก.ตงอิม | บงทง | ไทย | กชน. 10800 | YES | |
| 10 | 1122122335599 | นาย ตงม | อู่ทง | 514 ซ.ตงอิม 1 | ก.ตงอิม | บงทง | ไทย | กชน. 10900 | YES | |
| 11 | 1122144448775 | นาย สทง | อู่ทง | 50/54-55 ซ. 8 | ก.ตงอิม | บงทง | ไทย | กชน. 10320 | YES | |
| 12 | 1122255155538 | นางทง อู่ทง | อู่ทง | ม.ตงอิม 127 | อู่ทง | บงทง | ไทย | กชน. 10120 | YES | |
| 13 | 1122323213788 | นาย สทง | อู่ทง | 23 ซ.ตงอิม 11 | ตงอิม | บงทง | ไทย | กชน. 10240 | YES | |
| 14 | 1122444775575 | นาย อู่ทง | อู่ทง | 77 ก.ตงอิม | อู่ทง | จระงัก | ไทย | กชน. 10400 | YES | |
| 15 | 1123123444546 | นาย อู่ทง | อู่ทง | 302/208 | ก.ตงอิม 9 | ห้วยทวง | ไทย | กชน. 10320 | YES | |
| 16 | 1123222555555 | นาย อู่ทง | อู่ทง | 822/15 ซ.ตงอิม | อู่ทง | บงทง | ไทย | กชน. 10120 | YES | |
| 17 | 1123455845587 | นาย ตงม | อู่ทง | 59 ซ.ตงอิม | ก.ตงอิม | จระงัก | ไทย | กชน. 10400 | NO | |
| 18 | 1124455795552 | นาย อู่ทง | อู่ทง | 662 ก.ตงอิม 4 | บงทง | บงทง | ไทย | กชน. 10500 | NO | |
| 19 | 1124555465556 | นาย อู่ทง | อู่ทง | 273 ม.ตงอิม | อู่ทง | สทงทง | ไทย | กชน. 10240 | NO | |
| 20 | 1144455577789 | นาย อู่ทง | อู่ทง | 75/53 ซ.ตงอิม | อู่ทง | บงทง | ไทย | กชน. 10700 | YES | |
| 21 | 1144555778976 | นาย อู่ทง | อู่ทง | 17 ซ.ตงอิม 25 | อู่ทง | ตงอิม | ไทย | กชน. 10110 | YES | |

2. กรณีที่ผู้มาลงคะแนนไม่มีสิทธิในการลงคะแนน

| วot index | vot icard | vot name | vot surname | vot address | vot NOG | vot locality | vot county | vot province | vot state | vot complete |
|-----------|---------------|-------------|-------------|-------------------|-------------|--------------|------------|--------------|-----------|--------------|
| 1 | 111111111111 | นาย คิล | ป๋อทอง | 14 ซ.ดอนมู่เตล 1 | เช็กทุ่ง 57 | บางนาท | ไทย | กชน. 10120 | NO | |
| 2 | 111122244444 | นาย อุดม | ทวีโร | 14 ซ.ติงตู่ป๋อป๋อ | สามสมมต | ห้วยทวง | ไทย | กชน. 10320 | NO | |
| 3 | 111122334555 | นาย เดก | โหว | 58/22 ซ.ตงอิม 1 | ก.ตงอิม 32 | จระงัก | ไทย | กชน. 10230 | YES | |
| 4 | 111123355666 | นาย อรทงิม | อิมตง | 67/180 | สทงทัก | อู่จัก | ไทย | กชน. 10900 | NO | |
| 5 | 1112123314454 | นาย ตงม | อู่ทง | 66 ม.ปะ 4 | ก.ทงทง 9 | สทงทง | ไทย | กชน. 10250 | YES | |
| 6 | 1112444777662 | นาย ตงม | อู่ทง | 97/89 ม.ปะ | ตงอิม | บงทง | ไทย | กชน. 10240 | YES | |
| 7 | 1112454556677 | นาย ตงม | อู่ทง | 15 ซ.ตงอิม | ก.ตงอิม | บงทง | ไทย | กชน. 10120 | YES | |
| 8 | 1121455155526 | นางทง อู่ทง | อู่ทง | 4 ม.ตงอิม | ก.ตงอิม | บงทง | ไทย | กชน. 10220 | YES | |
| 9 | 1121477799533 | นาย สทงทง | อู่ทง | 36 ซ. ตงอิม 34 | ก.ตงอิม | บงทง | ไทย | กชน. 10800 | YES | |
| 10 | 1122122335599 | นาย ตงม | อู่ทง | 514 ซ.ตงอิม 1 | ก.ตงอิม | บงทง | ไทย | กชน. 10900 | YES | |
| 11 | 1122144448775 | นาย สทง | อู่ทง | 50/54-55 ซ. 8 | ก.ตงอิม | บงทง | ไทย | กชน. 10320 | YES | |
| 12 | 1122255155538 | นางทง อู่ทง | อู่ทง | ม.ตงอิม 127 | อู่ทง | บงทง | ไทย | กชน. 10120 | YES | |
| 13 | 1122323213788 | นาย สทง | อู่ทง | 23 ซ.ตงอิม 11 | ตงอิม | บงทง | ไทย | กชน. 10240 | YES | |
| 14 | 1122444775575 | นาย อู่ทง | อู่ทง | 77 ก.ตงอิม | อู่ทง | จระงัก | ไทย | กชน. 10400 | YES | |
| 15 | 1123123444546 | นาย อู่ทง | อู่ทง | 302/208 | ก.ตงอิม 9 | ห้วยทวง | ไทย | กชน. 10320 | YES | |
| 16 | 1123222555555 | นาย อู่ทง | อู่ทง | 822/15 ซ.ตงอิม | อู่ทง | บงทง | ไทย | กชน. 10120 | YES | |
| 17 | 1123455845587 | นาย ตงม | อู่ทง | 59 ซ.ตงอิม | ก.ตงอิม | จระงัก | ไทย | กชน. 10400 | NO | |
| 18 | 1124455795552 | นาย อู่ทง | อู่ทง | 662 ก.ตงอิม 4 | บงทง | บงทง | ไทย | กชน. 10500 | NO | |
| 19 | 1124555465556 | นาย อู่ทง | อู่ทง | 273 ม.ตงอิม | อู่ทง | สทงทง | ไทย | กชน. 10240 | NO | |
| 20 | 1144455577789 | นาย อู่ทง | อู่ทง | 75/53 ซ.ตงอิม | อู่ทง | บงทง | ไทย | กชน. 10700 | YES | |
| 21 | 1144555778976 | นาย อู่ทง | อู่ทง | 17 ซ.ตงอิม 25 | อู่ทง | ตงอิม | ไทย | กชน. 10110 | YES | |

รูปที่ 4.4 แสดงฐานข้อมูลของผู้มาใช้สิทธิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.13 ฟังก์ชันของการกรอกข้อมูลผู้สมัคร

หน้า 2 of (28)

ส่วนการจัดการข้อมูลผู้สมัคร

Cancel

หมายเลขผู้สมัคร

รหัสบัตรประชาชน

ชื่อ

นามสกุล

พรรค

ชื่อผู้

เลขที่

หมู่ที่

ตำบล

อำเภอ

จังหวัด

โทรศัพท์

| op_id | op_idcard | op_กตธอ | op_หมายเลข^ |
|-------|-------------|--------------|-------------|
| 1 | 12345678... | นาง ดวง | ทองดี |
| ▶ | 12345678... | นาย ทอง | สุโข |
| 11 | 12457965... | นางสาว ศิริ | ช่วยเจริญ |
| 12 | 12345687... | นางสาว พุทธิ | วังทอง |
| 13 | 12547849... | นาย สมบูรณ์ | ฉวีใจ |
| 14 | 12345678... | นาย พิเศษ | สังข์ |
| 15 | 12456781... | นาง สมศรี | สังข์ |
| 16 | 12564878... | นาง ศาณี | กาบแก้ว |
| 17 | 12345678... | นาย ศาณี | ภริยธรรม |
| 18 | 12247894... | นาง จารุวรรณ | แดงธรรม |
| 19 | 12457489... | นางสาว ไชรี | สุดสวย |
| 2 | 12349876... | นาย คำ | คพม่า |
| 20 | 12574898... | นาย สมธิ | แม่สง |
| 21 | 12457855... | นาง คำพญา | บัวแก้ว |
| 22 | 12347445... | นาย อัมพันธ์ | สุภาพ |
| 23 | 15487894... | นางสาว นิษา | ศรีอัมพันธ์ |
| 24 | 18457815... | นาย กิ่ง | เจริญผล |
| 25 | 15482755... | นางสาว เสน | สว่าง |
| 28 | 14575252... | นาง ชานแก้ว | วรงค์ |

รูปที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 of (28) Add

ส่วนการจัดการข้อมูลผู้สมัคร

หมายเลขผู้สมัคร 10
 รหัสบัตรประชาชน 1234567899554
 ชื่อ นาย ทน
 นามสกุล สมใจ
 พรรค 195 ส.ชาตวิวัฒน์
 ชื่อผู้ เลขที่ จุฬาลงกรณ์ 56
 หมู่ที่ พระโขนง
 ตำบล โข
 อำเภอ กทม. 10110
 จังหวัด 02-4386788
 โทรศัพท์ วมใจ

| ap_id | ap_idcard | ap_name | ap_gender |
|-------|-------------|-------------------------------|-----------|
| 1 | 12345678... | นาย ทน | ชาย |
| 10 | 12345678... | นาย ทน | ชาย |
| 11 | 12457896... | นางสาว ศิ | ช่วยเหลือ |
| 12 | 12345687... | นางสาว สุ | วังของ |
| 13 | 12547849... | นาย สม | มีใจ |
| 14 | 12345678... | นาย ส | มี |
| 15 | 12456781... | นาง ส | ดี |
| 18 | 12584879... | นาง ค | กาชแก้ว |
| 17 | 12345678... | นาย ส | กวีธรรม |
| 18 | 12247894... | นาง อ | แสงธรรม |
| 18 | 12457489... | นางสาว ไ | สุภาพ |
| 2 | 12349878... | นาย ต | คชฉ้า |
| 20 | 12574896... | นาย ส | แฟง |
| 21 | 12457855... | นาง ค | วิ้วแก้ว |
| 22 | 12347445... | นาย ส | สุภาพ |
| 23 | 15487894... | นางสาว น | พรหมพันธ์ |
| 24 | 16457815... | นาย ก | เจริญผล |
| 25 | 15482755... | นางสาว เ </td <td>คำรุ่ง</td> | คำรุ่ง |
| 26 | 14575252... | นาง ช | วาทศ |

รูปที่ 2

รูปที่ 4.5 แสดงการกรอกข้อมูลประวัติของผู้สมัครโดยเจ้าหน้าที่หน่วยลงคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.4 ฟังก์ชันของการกรอกข้อมูลผู้ลงคะแนน

ส่วนการจัดการข้อมูลผู้ลงคะแนน

| ชื่อผู้ลงคะแนน | หมายเลขบัตรประชาชน | ชื่อ | นามสกุล | เพศ | สถานะ | ที่อยู่ | เลขที่ | หมู่ที่ | ตำบล | อำเภอ | จังหวัด |
|----------------|--------------------|------|---------|-----|-------|---------|--------|---------|------|-------|---------|
| | | | | Yes | No | | | | | | |

| vote_index | vote_idcard | vote_name | vote_summa |
|------------|-------------|---------------|-------------|
| 1 | 11111111... | นาย สิต | ประสงค์ |
| 10 | 11221223... | นาง เสนอณี | สิงโตสาร |
| 100 | 13020201... | นาง วราวุธ | ธีระกุล |
| 101 | 12101301... | นางสาว รัง... | ไธสงกุล |
| 102 | 10257048... | นาง วิภาห | สุภาไพฑูรณ์ |
| 103 | 10754800... | นาง รนิช | ไกรบุตร |
| 104 | 10230053... | นาง ดาวใจ | อุมารินทร์ |
| 105 | 14525078... | นาย นิช | บุญศักดิ์ |
| 106 | 12110134... | นาย สาธารณ | ชานธีชน |
| 107 | 12405789... | นาย วุฒิชัย | ธีระชัยกุล |
| 108 | 14709070... | นาย สุศักดิ์ | เศรษฐกร |
| 109 | 11470460... | นาง ศวรรณ | เกษณี |
| 11 | 11221444... | นาง อมร | สีดวง |
| 110 | 14375010... | นาย รณ | พาศิตกุล |
| 12 | 11222501... | นางสาว ไชยม | ศรีแก้ว |
| 13 | 11223232... | นาย สิชย์ | ธนากร |
| 14 | 11224447... | นาย สุรชุต | ไพฑูรินทร์ |
| 15 | 11201234... | นาย สนิสา | อินทุสมบัติ |
| 10 | 11202225... | นาย อาทิตย์ | ศิวาพันธ์ |

รูปที่ 1

ส่วนการจัดการข้อมูลผู้ลงคะแนน

| ชื่อผู้ลงคะแนน | หมายเลขบัตรประชาชน | ชื่อ | นามสกุล | เพศ | สถานะ | ที่อยู่ | เลขที่ | หมู่ที่ | ตำบล | อำเภอ | จังหวัด |
|----------------|--------------------|---------|---------|-----|-------|---------|--------|---------|------|-------|---------|
| 1 | | นาย สิต | ประสงค์ | Yes | No | | | | | | |

| vote_index | vote_idcard | vote_name | vote_summa |
|------------|-------------|---------------|-------------|
| 1 | 11111111... | นาย สิต | ประสงค์ |
| 10 | 11221223... | นาง เสนอณี | สิงโตสาร |
| 100 | 13020201... | นาง วราวุธ | ธีระกุล |
| 101 | 12101301... | นางสาว รัง... | ไธสงกุล |
| 102 | 10257048... | นาง วิภาห | สุภาไพฑูรณ์ |
| 103 | 10754800... | นาง รนิช | ไกรบุตร |
| 104 | 10230053... | นาง ดาวใจ | อุมารินทร์ |
| 105 | 14525078... | นาย นิช | บุญศักดิ์ |
| 106 | 12110134... | นาย สาธารณ | ชานธีชน |
| 107 | 12405789... | นาย วุฒิชัย | ธีระชัยกุล |
| 108 | 14709070... | นาย สุศักดิ์ | เศรษฐกร |
| 109 | 11470460... | นาง ศวรรณ | เกษณี |
| 11 | 11221444... | นาง อมร | สีดวง |
| 110 | 14375010... | นาย รณ | พาศิตกุล |
| 12 | 11222501... | นางสาว ไชยม | ศรีแก้ว |
| 13 | 11223232... | นาย สิชย์ | ธนากร |
| 14 | 11224447... | นาย สุรชุต | ไพฑูรินทร์ |
| 15 | 11201234... | นาย สนิสา | อินทุสมบัติ |
| 10 | 11202225... | นาย อาทิตย์ | ศิวาพันธ์ |

รูปที่ 2

รูปที่ 4.6 แสดงการกรอกข้อมูลประวัติของผู้มาลงคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 ฟังก์ชันการสมัครตรงคะแนน

1. Click ที่โปรแกรมสมัครตรงคะแนน หน้าจะเป็นดังนี้

สมัครตรงคะแนน

ชื่อผู้สมัคร
จริง ชื่อจริงภาษาไทย

ชื่อผู้สมัครจริง

ตรวจสอบประวัติการลงทะเบียน

สมัครตรงคะแนน

ดูข้อมูลจากบัตร

บ้านหอ
ศิษย์
คณาฯ
วิจัย
เกษียณ
ครูพี่
ต่าง
จำเลย
งดรับ

2. จากนั้นเราจะป้อนเลขรหัสบัตรประชาชนของผู้มาใช้สิทธิ์

ชื่อผู้สมัครจริง
รหัสบัตรประชาชน
162247852245

ชื่อผู้สมัครจริง
คำศัพท์ของคะแนน

บ้านหอ
ศิษย์
คณาฯ
วิจัย
เกษียณ
ครูพี่
ต่าง
จำเลย
งดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อป้อนเลขรหัสบัตรประชาชนแล้วจะทำการตรวจสอบสิทธิการลงคะแนน

ปิดไฟ ปิดระบบ ประกาศผล ออกค่าใช้สอย

| | |
|---|---|
| ค้นหาข้อมูล รหัสบัตรประชาชน 1632478952245 | ข้อมูลค้นหาได้ ลำดับผู้ลงคะแนน 98 ชื่อ นางสาว จันทา นามสกุล ศศิประภา สังกัด สถานะ No ที่อยู่ เลขที่ 50บ.วชิรธรรมสาริต หมู่ที่ ชุมวิท 101/1 ตำบล บางนา อำเภอ โขง จังหวัด กทม. 10260 |
|---|---|

กรุณาทำการออกบัตรได้ !!

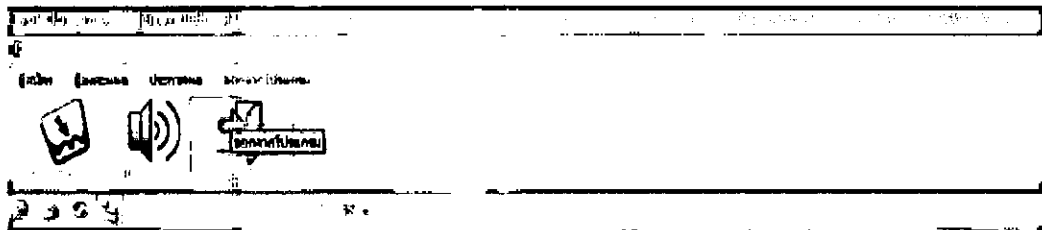
4. ซึ่งผลที่ได้ก็คือ ผู้มาใช้สิทธิคนนี้มีรายชื่อในฐานข้อมูลของผู้มีสิทธิลงคะแนน เจ้าหน้าที่ประจำหน่วยเลือกตั้งจึงทำการออกบัตรลงคะแนนให้แก่ผู้มาใช้สิทธิได้

ปิดไฟ ปิดระบบ ประกาศผล ออกค่าใช้สอย

| | |
|--------------------------------|---|
| ค้นหาข้อมูล รหัสบัตรประชาชน | ข้อมูลค้นหาได้ ลำดับผู้ลงคะแนน 98 ชื่อ นางสาว จันทา นามสกุล ศศิประภา สังกัด สถานะ No ที่อยู่ เลขที่ 50บ.วชิรธรรมสาริต หมู่ที่ ชุมวิท 101/1 ตำบล บางนา อำเภอ โขง จังหวัด กทม. 10260 |
|--------------------------------|---|

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. และเมื่อครบกำหนดเวลา 15.00 น. ซึ่งเป็นเวลาสิ้นสุดการลงคะแนนเสียง เราก็จะสามารถทราบคะแนนของผู้ที่มาใช้สิทธิได้ทันทีเมื่อทำการปิดหีบ



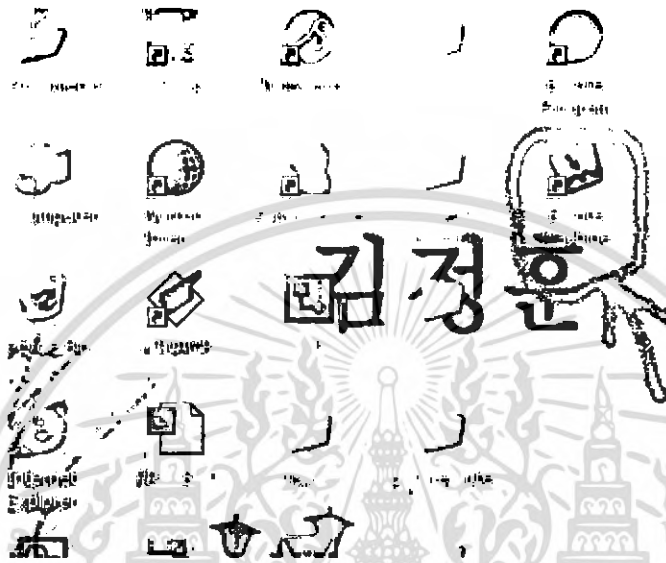
| ผลการเลือกตั้ง | | | | | 33/2550 |
|----------------|--------------|------------|------------|-------|---------|
| หมายเลข | ชื่อ | นามสกุล | สังกัดพรรค | คะแนน | |
| 1 | ชาย สม | ทองดี | ชาติไทย | 0 | |
| 2 | ชาย คำ | คนคำ | ชาติไทย | 1 | |
| 3 | ชาย นิลา | จิลาภา | ชาติไทย | 1 | |
| 4 | นางสาว อิ่ม | สุเมธ | ชาติไทย | 0 | |
| 5 | นาง นารี | สุเมธภรรยา | ก้าวหน้า | 0 | |
| 6 | นาย สมศักดิ์ | ทองดี | พลังใจ | 0 | |
| 7 | นาย สมาน | ใจดี | มั่นคง | 0 | |
| 8 | นาย สมัย | ภาพฉาย | พลังไทย | 0 | |
| 9 | นางสาว สมพร | ทองดี | ชาติไทย | 0 | |
| 10 | นาย สม | สมใจ | รวมใจ | 1 | |
| 11 | นางสาว สิริ | สุเมธณี | ชาติพัฒนา | 0 | |
| 12 | นางสาว สุเมธ | วิทอง | นิภา | 0 | |
| 13 | นาย สมบูรณ์ | มีใจ | ชนะใจ | 0 | |
| 14 | นาย สมชาย | มีชัย | เพื่อชาติ | 0 | |
| 15 | นาง สมศรี | ศิริ | เมืองใหม่ | 1 | |
| 16 | นาง ศรณี | ภรณ์แก้ว | รวมชาติ | 0 | |
| 17 | นาย สมศักดิ์ | แก้วทอง | พลังไทย | 0 | |
| 18 | นาง จุฬารัตน | แสงธรรม | พลังไทย | 0 | |
| 19 | นางสาว ไชว | สุทธยา | พัฒนา | 0 | |
| 20 | นาย สม | สมใจ | ชาติไทย | 0 | |
| 21 | นาง คำคำ | มีชัย | ชาติไทย | 0 | |
| 22 | นาย สมศักดิ์ | สุภาพ | กิจพัฒนา | 0 | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

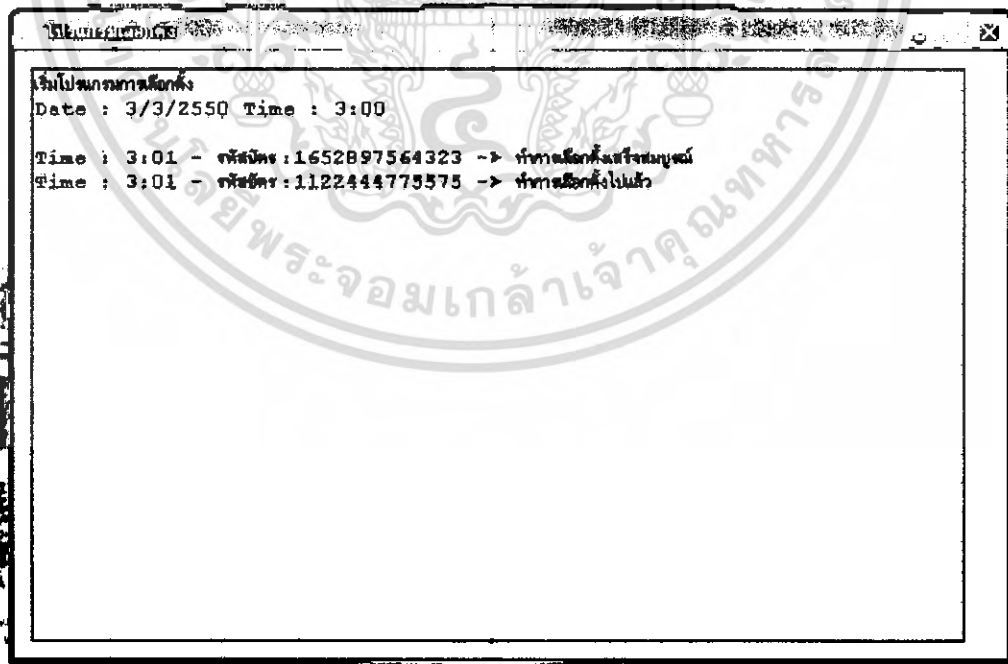
4.3 ฟังก์ชันสำหรับผู้ลงทะเบียนเลือกตั้ง

ฟังก์ชันสำหรับผู้ลงทะเบียนเลือกตั้ง จะใช้ในการเก็บบันทึกข้อมูลของผู้มาใช้สิทธิ และ เป็นฟังก์ชันที่ใช้ในการเก็บคะแนนของผู้มาใช้สิทธิ และยังเป็นการตรวจสอบว่าผู้ที่มาใช้สิทธินั้น ได้ลงคะแนนหรือไม่

4.3.1 ฟังก์ชันที่ใช้ในการเข้าระบบ

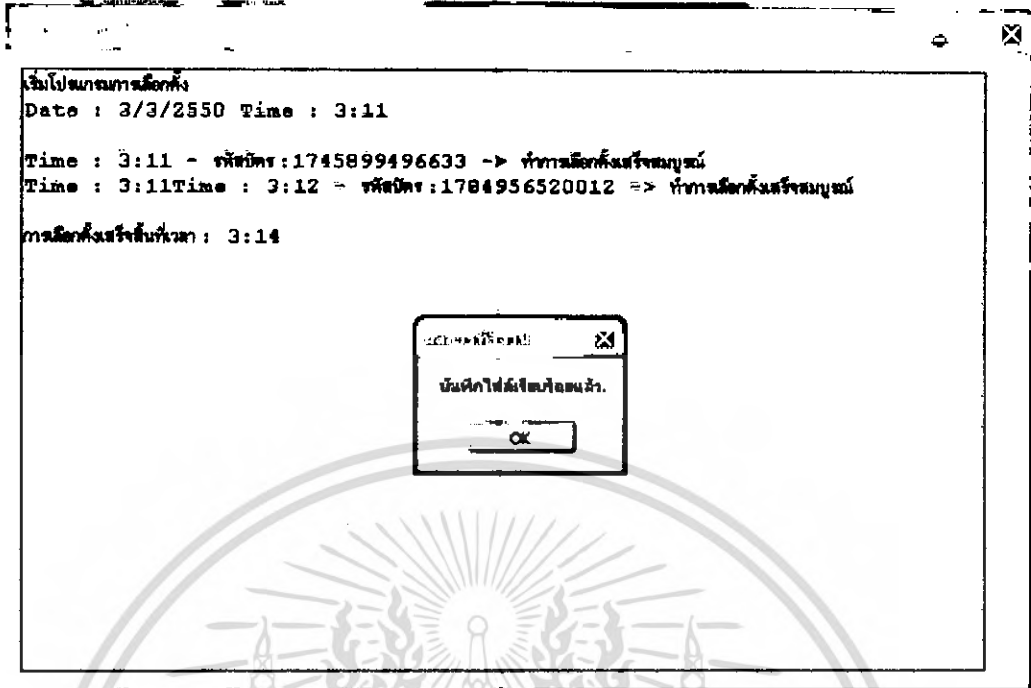


4.3.2 โปรแกรมที่ใช้ในการบันทึกผล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 เมื่อทำการเลือกตั้งเสร็จแล้ว เราจะทำการบันทึกผลข้อมูลทั้งหมดของผู้มาใช้วิธีการลงคะแนน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง

โครงการนี้เป็น การนำเอาระบบอิเล็กทรอนิกส์และระบบฐานข้อมูลมาใช้ในระบบการเลือกตั้งผ่านเครือข่าย Computer ซึ่งโครงการนี้แบ่งระบบการทำงานออกเป็น 2 ระบบ คือ ระบบของเครื่องลงคะแนนและระบบฐานข้อมูลของการเลือกตั้งผ่านเครือข่าย Computer ซึ่งในส่วนของระบบเครื่องลงคะแนนนั้นจะประกอบไปด้วยเครื่อง Client และเครื่องลงคะแนน โดยเครื่องลงคะแนนนั้นเป็นการประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์มาควบคุมการทำงาน โดยสามารถส่งและรับข้อมูลผ่านเครื่องคอมพิวเตอร์ Client ซึ่งใช้สำหรับการเลือกตั้ง

ในส่วนผลการทดลองนั้นเครื่องลงคะแนน สามารถลงคะแนนได้ตามวัตถุประสงค์แต่ก็ยังมีปัญหาบางประการดังต่อไปนี้ คือ ในส่วนของตัวเครื่องลงคะแนนมีปัญหาที่พบก็คือ การหาซื้ออุปกรณ์บางตัวหาซื้อยาก, ในการออกแบบ print ของวงจรเกิดความผิดพลาด เนื่องจากมี library ที่ออกแบบไม่ตรงกับรูปวงจรทำให้ในการเรียก file ข้อมูลที่ต้องการผิดพลาด จึงทำให้ต้องกริดลาย print ออกแล้วใช้สายไฟจัมพ์แทน และในส่วนของวงจรที่แสดงผลตัวเลข (7 - segment) จะต้องมี R เพิ่มเข้าไป และในส่วนของซอฟต์แวร์มีปัญหาที่เกิดขึ้นดังนี้ การ ในการส่งข้อมูลผ่านเครือข่าย ไม่มีการเข้ารหัสที่เครื่องผู้ทำให้สามารถถูกดักข้อมูลได้ , และในการถืออินของโปรแกรมสำหรับดูแลการลงคะแนนจะไม่สามารถเปลี่ยนแปลงได้เนื่องจากถืออินนี้ถูกฝังในระบบด้วย แต่ถืออินของโปรแกรมสำหรับการตรวจสอบรายชื่อผู้ลงคะแนนสามารถเปลี่ยนแปลงได้ และปัญหาการติดต่อผ่านเครือข่ายการใช้งาน ไม่เสถียรอาจจะใช้งานได้บางครั้งบางคราวขึ้นอยู่กับระบบ Network

หนังสืออ้างอิง

1. เทคโนโลยีสารสนเทศการรีด
2. ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 , วรพจน์ กรแก้ววัฒนกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล
3. ระบบฐานข้อมูล , ศุภชัย สมพานิช
4. คู่มือการใช้งาน Visual C# .NET , ศุภชัย สมพานิช
5. Microsoft Access 2000 , วิภา เพิ่มทรัพย์ และ วกีน เพิ่มทรัพย์

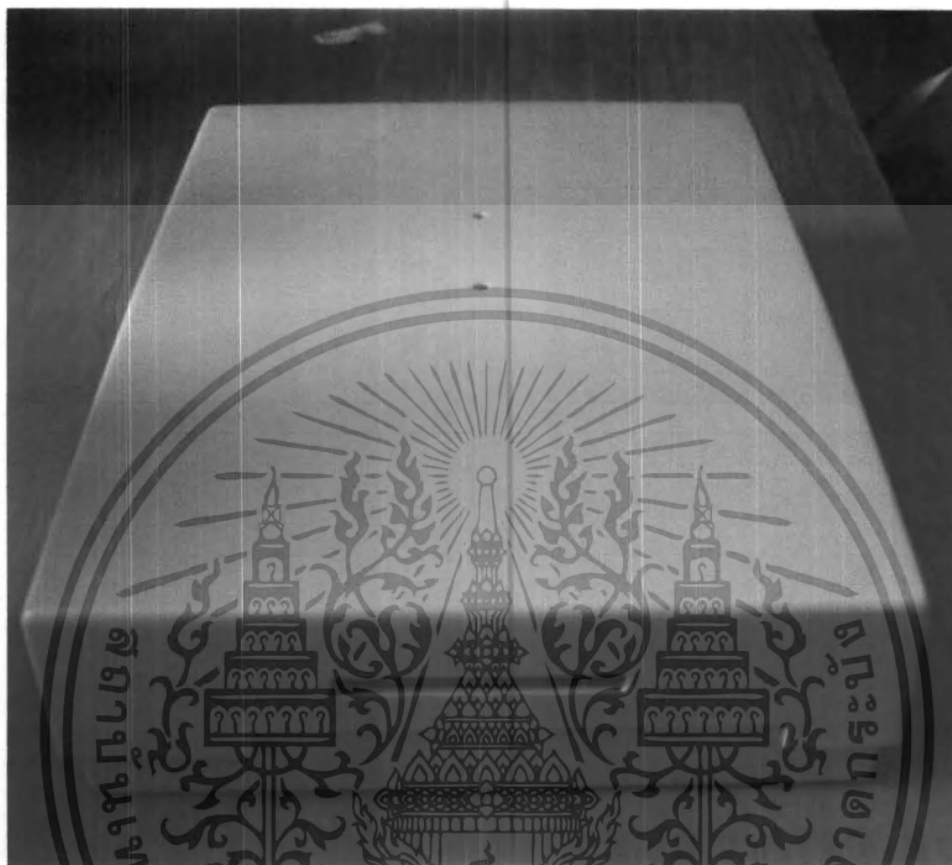


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



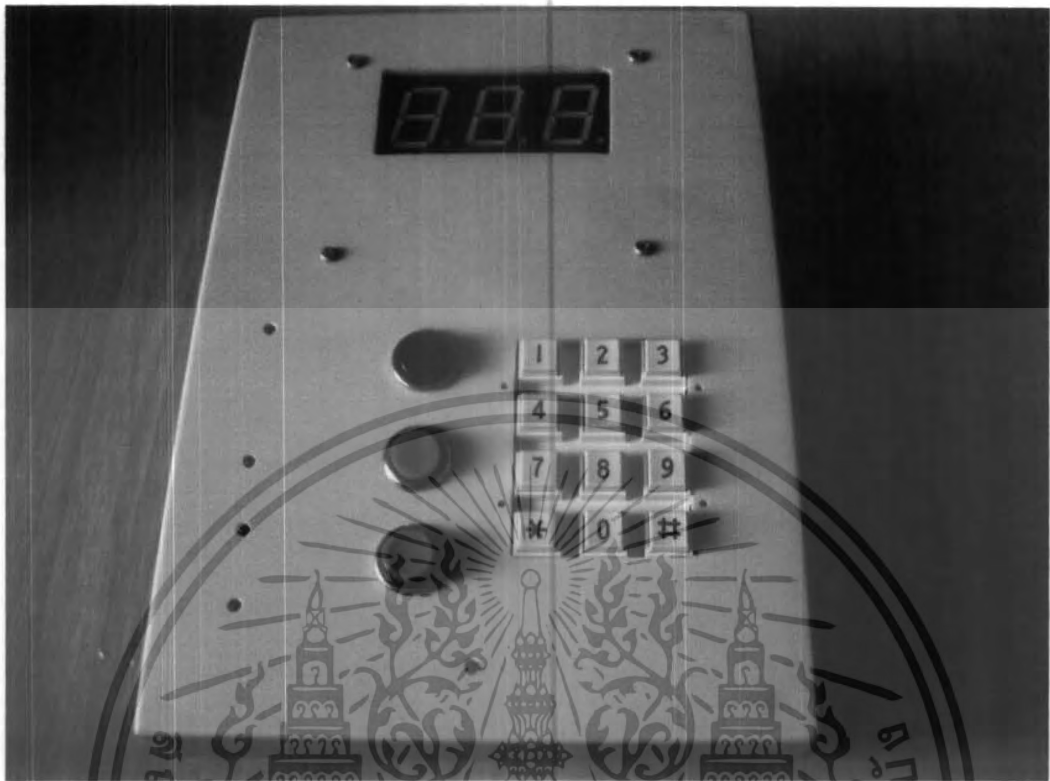
ภาพผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป เครื่องอ่าน - เขียนบัตรลงคะแนน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของ Card RW

```
#pragma code
#include "AT89S52.h"
#include <stdio.h>

sbit RST = P1^0;
sbit CLK = P1^1;
sbit IO = P1^2;
sbit VSMC = P1^6;
sbit C_SW = P1^7;
sbit BUZZ = P1^5;

#define rx_buffer_size 20
#define tx_buffer_size 20

unsigned char per_id[14];
const unsigned char code atr[ ] = {0xA2, 0x13, 0x10, 0x91};
const unsigned char psc[ ] = {0xFF, 0xFF, 0xFF};
static bit SMC_Status;

static char rx_buffer[rx_buffer_size];
static char idata tx_buffer[tx_buffer_size];
static unsigned char idata rx_index;
static unsigned char tx_index, tx_max;
static bit send_status;
static bit recv_status;

void serial_isr(void) interrupt 4 using 2
{
    static unsigned char rx_dat;
    if(RI)
    {
        RI=0;
        rx_dat = SBUF;
        if (!recv_status)
        {
            if (rx_dat == ':')
                rx_index = 0;
            else if (rx_dat == '\r')
                recv_status = 1;
            rx_buffer[rx_index++] = rx_dat;
            if (rx_index == rx_buffer_size)
                rx_index = 0;
        }
    }

    if(TI)
    {
        TI=0;
        if(send_status)
        {
            if(tx_index < tx_max)
                SBUF = tx_buffer[tx_index++];
            else
                send_status = 0;
        }
    }
}

void clr_rx_buffer(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    unsigned char i;
    for (i=0;i<rx_buffer_size;i++)
        rx_buffer[i] = 0;
}

void clr_per_id(void)
{
    unsigned char i;
    for (i=0;i<13;i++)
        per_id[i] = '0';
    per_id[13] = '\0';
}

void init (void)
{
    C_SW = 1;
    RST = 0;
    CLK = 0;
    IO = 1;
    BUZZ = 1;
    clr_rx_buffer();
    clr_per_id();
    SMC_Status = 0;
    tx_max = 0;
    tx_index = 0;
    rx_index = 0;
    send_status = 0;
    recv_status = 0;
}

void serial_init (void)
{
    PCON=PCON&0x7F;
    SCON=0x52;
    TMOD=0x20;
    TH1=0xFD;
    TR1=1;
    EA=1;
    RI=0;
    TI=0;
    ES=1;
}

void delay_msec (unsigned int count)
{
    unsigned char i;
    while (count)
    {
        for (i=1;i<=228;i++);
        count--;
    }
}

void beep(void)
{
    BUZZ = 0;
    delay_msec (250);
    BUZZ = 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void beep_error(void)
{
    BUZZ = 0;
    delay_msec (50);
    BUZZ = 1;
    delay_msec (50);
    BUZZ = 0;
    delay_msec (50);
    BUZZ = 1;
}

```

```

void smc_delay (void)
{
    unsigned char i;
    i = 20;
    while (i>0) i--;
}

```

```

void smcstart (void)
{
    CLK = 0;
    IO = 1;
    CLK = 1;
    smc_delay ();
    IO = 0;
    smc_delay ();
    CLK = 0;
    smc_delay ();
}

```

```

void smcstop (void)
{
    IO = 0;
    CLK = 1;
    smc_delay ();
    IO = 1;
    smc_delay ();
    CLK = 0;
    smc_delay ();
}

```

```

static bit smcchkatr (void)
{
    unsigned char i,io_data,atr_buf[4];

    RST = 1;
    smc_delay ();
    CLK = 1;
    smc_delay ();
    CLK = 0;
    smc_delay ();
    RST = 0;
    smc_delay ();

    io_data = 0;
    for (i=0;i<32;i++)
    {
        IO = 1;
        io_data >>= 1;
        if (IO) io_data |= 0x80; else io_data &= 0x7f;
        CLK = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    smc_delay ();
    CLK = 0;
    smc_delay ();
    if (i%8 == 7)
        {
            atr_buf[i/8] = io_data;
            io_data = 0;
        }
    }

    for (i = 0; i < 4; i++)
        if (atr_buf[i] != atr[i]) return 1;
    return 0;
}

```

```

void smcwrbyte (unsigned char io_data)

```

```

{
    unsigned char i;
    for (i=0;i<8;i++)
        {
            IO = (io_data & 0x1) ? 1 : 0;
            CLK = 1;
            smc_delay ();
            CLK = 0;
            smc_delay ();
            io_data >>= 1;
        }
}

```

```

unsigned char smcrdbyte (void)

```

```

{
    unsigned char i, io_data;
    io_data = 0;
    for (i=0;i<8;i++)
        {
            IO = 1;
            io_data >>= 1;
            if (IO) io_data |= 0x80; else io_data &= 0x7f;
            CLK = 1;
            smc_delay ();
            CLK = 0;
            smc_delay ();
        }
    return (io_data);
}

```

```

static bit smcread (unsigned char com, add, len)

```

```

{
    unsigned char i;
    smcstart ();
    smcwrbyte (com);
    smcwrbyte (add);
    smcwrbyte (0);
    smcstop ();
    for (i=0;i<len;i++)
        {
            if(C_SW)
                per_id[i] = smcrdbyte ();
            else
                return 1;
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    RST = 1;
    smc_delay ();
    RST = 0;
    return 0;
}

void smcwrite (unsigned char com,add,dat)
{
    smcstart ();
    smcwrbyte (com);
    smcwrbyte (add);
    smcwrbyte (dat);
    smcstop ();
    do {
        CLK = 1;
        smc_delay ();
        CLK = 0;
        smc_delay ();
        IO = 1;
    } while (!IO);
}

unsigned char smcverpsc (unsigned char *pscbuf)
{
    unsigned char mask,i;
    mask = 1;
    smcread (0x31,0,4);
    if (per_id[0]>0)
    {
        for (i=0;i<3;i++)
        {
            if (per_id[0] & mask) break;
            mask <<= 1;
        }
        per_id[0] ^= mask;
        smcwrite (0x39,0,per_id[0]);
        smcwrite (0x33,1,pscbuf[0]);
        smcwrite (0x33,2,pscbuf[1]);
        smcwrite (0x33,3,pscbuf[2]);
        smcwrite (0x39,0,0xFF);
        smcread (0x31,0,4);
    }
    mask = 0;
    for (i=0;i<3;i++)
    {
        if ((per_id[0] & 0x01) ==0) mask++;
        per_id[0] >>= 1;
    }
    return (mask);
}

static bit write_id(void)
{
    unsigned char i,addr,vdata;
    for (i=0;i<13;i++)
    {
        addr = i+0x20;
        smcwrite (0x38,addr,per_id[i]);
        vdata = per_id[i];
        smcread (0x30,addr,1);
        if(per_id[i] != vdata)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return 1;
    }
    return 0;
}

void send_data(unsigned char send_command)
{
    if(!send_status)
    {
        if(send_command == 0)
            tx_max = sprintf(tx_buffer,":0%s\r",per_id);
        else if(send_command == 1)
            tx_max = sprintf(tx_buffer,":1\r");
        else
            tx_max = sprintf(tx_buffer,":9\r");
        tx_index = 0;
        send_status = 1;
        TI=1;
    }
}

void get_command(void)
{
    unsigned char i;
    if(rx_buffer[0] == ':' && rx_buffer[rx_index-1] == '\r')
    {
        if(rx_buffer[1] == '0')
        {
            if(smcread (0x30,0x20,13))
            {
                VSMC = 1;
                send_data(9);
                beep_error();
            }
            else
            {
                send_data(0);
                beep();
            }
        }
        else if (rx_buffer[1] == '1')
        {
            i = smcverpsc (psc);
            if (i == 0)
            {
                for (i=0;i<13;i++)
                    per_id[i] = rx_buffer[i+2];
                if(write_id())
                {
                    VSMC = 1;
                    send_data(9);
                    beep_error();
                }
                else
                {
                    send_data(1);
                    beep();
                }
            }
        }
        else
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        VSMC = 1;
        beep_error();
    }
    else beep_error();
}
}

void process(void)
{
    while (1)
    {
        if(C_SW)
        {
            if(!SMC_Status)
            {
                SMC_Status = 1;
                VSMC = 0;
                smc_delay ();
                if(smcchkatr ())
                {
                    VSMC = 1;
                    beep_error();
                }
                else
                {
                    if(!VSMC)
                    {
                        while(C_SW)
                        {
                            if(recv_status)
                            {
                                get_command();
                                clr_rx_buffer();
                                rx_index = 0;
                                tx_index = 0;
                                recv_status = 0;
                                VSMC = 1;
                            }
                        }
                    }
                }
            }
            VSMC=1;
        }
        else
            SMC_Status = 0;
    }
}

void main (void)
{
    init();
    delay_msec (300);
    serial_init ();
    process();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมของ E-VOTING

```
#pragma code
#include "AT89S52.h"
#include <stdio.h>

sbit R0 = P0^0;
sbit R1 = P0^1;
sbit R2 = P0^2;
sbit R3 = P0^3;
sbit C0 = P0^4;
sbit C1 = P0^5;
sbit C2 = P0^6;

sbit RST = P1^0;
sbit CLK = P1^1;
sbit IO = P1^2;
sbit BUZZ = P1^5;
sbit C_SW = P1^7;
sbit VSMC = P1^6;

sbit Ok_SW = P2^0;
sbit Cancel_SW = P2^1;
sbit Enter_SW = P2^2;
sbit Disp_EN = P2^4;
sbit Disp_Lt = P2^5;
sbit Disp_CLK = P2^6;
sbit Disp_Si = P2^7;

sbit Led_Status1 = P3^4;
sbit Led_Status2 = P3^5;
sbit Led_Status3 = P3^6;

#define rx_buffer_size 5
#define tx_buffer_size 25

const unsigned char Disp [] =
    { 0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F };
unsigned char Disp_Show[3];
unsigned char per_id[14];
const unsigned char code atr[] = {0xA2, 0x13, 0x10, 0x91};

static unsigned char n_digit;
static char rx_buffer[rx_buffer_size];
static char idata tx_buffer[tx_buffer_size];
static unsigned char idata rx_index;
static unsigned char tx_index, tx_max;
static bit send_status;
static bit recv_status;
static bit SMC_Status;

void serial_isr(void) interrupt 4 using 2
{
    static unsigned char rx_dat;
    if(RI)
    {
        RI=0;
        rx_dat = SBUF;
        if (!recv_status)
        {
            if (rx_dat == ':')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        rx_index = 0;
        else if (rx_dat == '\r')
            recv_status = 1;
        rx_buffer[rx_index++] = rx_dat;
        if (rx_index == rx_buffer_size)
            rx_index = 0;
    }
}

if(TI)
{
    TI=0;
    if(send_status)
    {
        if(tx_index < tx_max)
            SBUF = tx_buffer[tx_index++];
        else
            send_status = 0;
    }
}

void clr_rx_buffer(void)
{
    unsigned char i;
    for (i=0;i<rx_buffer_size;i++)
        rx_buffer[i] = 0;
}

void clr_per_id(void)
{
    unsigned char i;
    for (i=0;i<13;i++)
        per_id[i] = '0';
    per_id[13] = '\0';
}

void init (void)
{
    Ok_SW = 1;
    Cancel_SW = 1;
    Enter_SW = 1;
    Led_Status1 = 0;
    Led_Status2 = 0;
    Led_Status3 = 0;
    C_SW = 1;
    RST = 0;
    CLK = 0;
    IO = 1;
    BUZZ = 1;
    clr_rx_buffer();
    clr_per_id();
    SMC_Status = 0;
    tx_max = 0;
    tx_index = 0;
    rx_index = 0;
    send_status = 0;
    recv_status = 0;
}

void serial_init (void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    PCON=PCON&0x7F;
    SCON=0x52;
    TMOD=0x20;
    TH1=0xFD;
    TR1=1;
    EA=1;
    RI=0;
    TI=0;
    ES=1;
}

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

void delay_msec (unsigned int count)
{
    unsigned char i;
    while (count)
    {
        for (i=1;i<=228;i++);
        count--;
    }
}

void beep(void)
{
    BUZZ = 0;
    delay_msec (250);
    BUZZ = 1;
}

void beep_error(void)
{
    BUZZ = 0;
    delay_msec (50);
    BUZZ = 1;
    delay_msec (50);
    BUZZ = 0;
    delay_msec (50);
    BUZZ = 1;
}

void Disp_Shift (unsigned char Seg_Data)
{
    unsigned char i;
    Disp_Si = 0;
    Disp_Lt = 0;
    Disp_CLK = 0;

    for (i=0; i < 8; i++)
    {
        Disp_Si = (Seg_Data & 0x80) ? 1 : 0;
        Disp_CLK = 1;
        delay ();
        Disp_CLK = 0;
        Disp_Si = 0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        delay ();
        Seg_Data <<= 1;
    }
    Disp_Lt = 1;
    delay ();
    Disp_Lt = 0;
}

void Clr_Displ (void)
{
    Disp_Shift(0x00);
    Disp_Shift(0x00);
    Disp_Shift(0x00);
}

unsigned char read_key (void)
{
    P0=0x0f;

    C2=1;
    C1=1;
    C0=0;

    if (R0 == 0)
    {
        delay_msec (50);
        while (R0==0);
        return '1';
    }
    if (R1 == 0)
    {
        delay_msec (50);
        while (R1==0);
        return '4';
    }
    if (R2 == 0)
    {
        delay_msec (50);
        while (R2==0);
        return '7';
    }
    if (R3 == 0)
    {
        delay_msec (50);
        while (R3==0);
        return '*';
    }
    C2=1;
    C0=1;
    C1=0;

    if (R0 == 0)
    {
        delay_msec (50);
        while (R0==0);
        return '2';
    }
    if (R1 == 0)
    {
        delay_msec (50);
        while (R1==0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return '5';
    }
    if (R2 == 0)
    {
        delay_msec (50);
        while (R2==0);
        return '8';
    }
    if (R3 == 0)
    {
        delay_msec (50);
        while (R3==0);
        return '0';
    }
    C0=1;
    C1=1;
    C2=0;
    if (R0 == 0)
    {
        delay_msec (50);
        while (R0==0);
        return '3';
    }
    if (R1 == 0)
    {
        delay_msec (50);
        while (R1==0);
        return '6';
    }
    if (R2 == 0)
    {
        delay_msec (50);
        while (R2==0);
        return '9';
    }
    if (R3 == 0)
    {
        delay_msec (50);
        while (R3==0);
        return '#';
    }
    return 0;
}

```

```

static bit read_sw (unsigned char sw)
{
    if (sw == 1 && !Enter_SW)
    {
        delay_msec (20);
        while (Enter_SW==0);
        return 1;
    }
    if (sw == 2 && !Ok_SW)
    {
        delay_msec (20);
        while (Ok_SW==0);
        return 2;
    }
    if (sw == 3 && !Cancel_SW)
    {
        delay_msec (20);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        while (Cancel_SW==0);
        return 3;
    }
    return 0;
}

void smcstart (void)
{
    CLK = 0;
    IO = 1;
    CLK = 1;
    delay ();
    IO = 0;
    delay ();
    CLK = 0;
    delay ();
}

void smcstop (void)
{
    IO = 0;
    CLK = 1;
    delay ();
    IO = 1;
    delay ();
    CLK = 0;
    delay ();
}

static bit smcchkatr (void)
{
    unsigned char i,io_data,atr_buf[4];

    RST = 1;
    delay ();
    CLK = 1;
    delay ();
    CLK = 0;
    delay ();
    RST = 0;
    delay ();

    io_data = 0;
    for (i=0;i<32;i++)
    {
        IO = 1;
        io_data >>= 1;
        if (IO) io_data |= 0x80; else io_data &= 0x7f;
        CLK = 1;
        delay ();
        CLK = 0;
        delay ();
        if (i%8 == 7)
        {
            atr_buf[i/8] = io_data;
            io_data = 0;
        }
    }

    for (i = 0; i < 4; i++)
        if (atr_buf[i] != atr[i]) return 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return 0;
    }

void smcwrbyte (unsigned char io_data)
{
    unsigned char i;
    for (i=0;i<8;i++)
    {
        IO = (io_data & 0x1) ? 1 : 0;
        CLK = 1;
        delay ();
        CLK = 0;
        delay ();
        io_data >>= 1;
    }
}

unsigned char smcrdbyte (void)
{
    unsigned char i,io_data;
    io_data = 0;
    for (i=0;i<8;i++)
    {
        IO = 1;
        io_data >>= 1;
        if (IO) io_data |= 0x80; else io_data &= 0x7f;
        CLK = 1;
        delay ();
        CLK = 0;
        delay ();
    }
    return (io_data);
}

static bit smcread (unsigned char com,add,len)
{
    unsigned char i;
    smcstart ();
    smcwrbyte (com);
    smcwrbyte (add);
    smcwrbyte (0);
    smcstop ();
    for (i=0;i<len;i++)
    {
        if(C_SW)
            per_id[i] = smcrdbyte ();
        else
            return 1;
    }
    RST = 1;
    delay ();
    RST = 0;
    return 0;
}

static bit get_response (void)
{
    if(rx_buffer[0] == ':' && rx_buffer[rx_index-1] == '\r')
    {
        if(rx_buffer[1] == '0')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        beep_error();
        return 0;
    }
    else if (rx_buffer[1] == '1')
    {
        beep();
        return 1;
    }
}
return 0;
}

void send_data(unsigned char send_command)
{
    if(!send_status)
    {
        if(send_command == 0)
            tx_max = sprintf(tx_buffer,":0%s\r", per_id);
        else if(send_command == 1)
            tx_max = sprintf(tx_buffer,":1%s%s\r",Disp_Show,
per_id);
        tx_index = 0;
        send_status = 1;
        TI=1;
    }
    while(!recv_status)
    {
        if(!C_SW)
        {
            VSMC = 1;
            break;
        }
    }
    if(!get_response())
    {
        VSMC = 1;
        beep_error();
    }
    else
        beep();
    recv_status = 0;
    clr_rx_buffer();
}

```

```

void process(void)
{
    unsigned char Disp_Temp;//, i 0;

    Disp_EN = 1;
    Clr_Disb ();
    delay ();
    Disp_EN = 0;

    while (1)
    {
        Disp_Show[0] = ' ';
        Disp_Show[1] = ' ';
        Disp_Show[2] = ' ';
        n_digit = 3;
        if(C_SW)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    if(!SMC_Status)
    {
        SMC_Status = 1;
        VSMC = 0;

        delay ();
        if(smcchkatr ())
        {
            VSMC = 1;
            beep_error();
        }
        else
        {
            if(smcread (0x30,0x20,13))
            {
                VSMC = 1;
                beep_error();
            }
            else
            {
                send_data(0);
                if(!VSMC)
                {
                    Led_Status1 = 1;
                    while(C_SW)
                    {
                        if(!Led_Status2)
                        {
                            Disp_Temp =
read_key();
n_digit > 0)
Disp_Show[1];
Disp_Show[2];
Disp_Temp;
Disp_Shift(Disp[Disp_Temp - 0x30]);
n_digit < 3)
                            n_digit--;
                            if(read_sw(1) &&
                                {
                                    Led_Status2 = 1;
                                    beep();
                                }
                            }
                        }
                    }
                    else
                    {
                        if(read_sw(2))
                        {
                            send_data(1);
                            if(!VSMC)
                            {
                                VSMC = 1;
                                Led_Status3 = 1;
                            }
                        }
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        break;
    }
}
if(read_sw(3))
{
    VSMC = 1;
    beep();
    break;
}
}
}
if(!C_SW)
    beep_error();
}
}
VSMC = 1;
}
Clr_Displ ();
}
else
{
    SMC_Status = 0;
    Led_Status1 = 0;
    Led_Status2 = 0;
    Led_Status3 = 0;
}
}
}
void main (void)
{
    init();
    delay_msec (300);
    serial_init ();
    process();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51® Products
- 8K Bytes of In-System Programmable (ISP) Flash Memory
 - Endurance: 1000 Write/Erase Cycles
- 4.0V to 5.5V Operating Range
- Fully Static Operation: 0 Hz to 33 MHz
- Three-level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Eight Interrupt Sources
- Full Duplex UART Serial Channel
- Low-power Idle and Power-down Modes
- Interrupt Recovery from Power-down Mode
- Watchdog Timer
- Dual Data Pointer
- Power-off Flag

Description

The AT89S52 is a low-power, high-performance CMOS 8-bit microcontroller with 8K bytes of in-system programmable Flash memory. The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard 80C51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with in-system programmable Flash on a monolithic chip, the Atmel AT89S52 is a powerful microcontroller which provides a highly-flexible and cost-effective solution to many embedded control applications.

The AT89S52 provides the following standard features: 8K bytes of Flash, 256 bytes of RAM, 32 I/O lines, Watchdog timer, two data pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power-down mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.



**8-bit
Microcontroller
with 8K Bytes
In-System
Programmable
Flash**

AT89S52

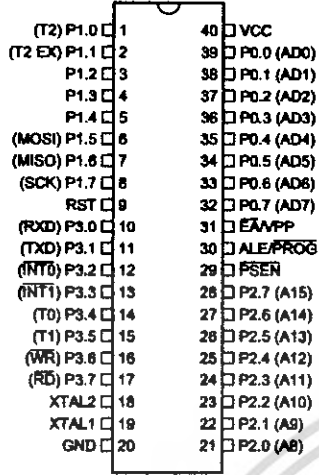
Rev. 1919A-07/01



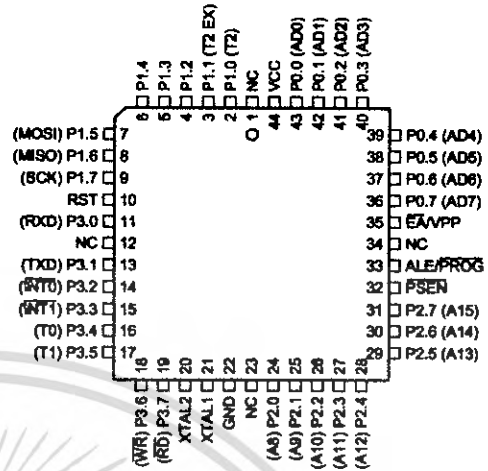
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Configurations

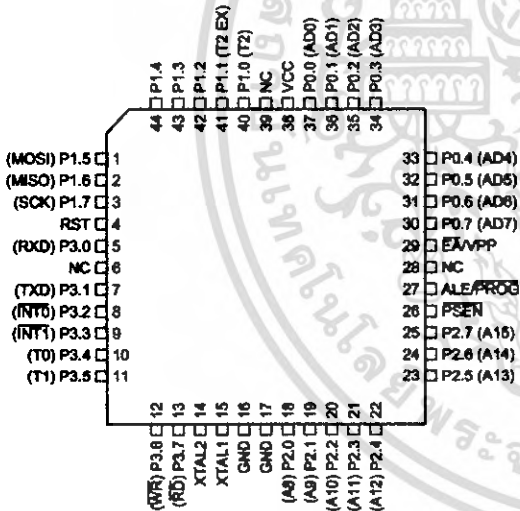
PDIP



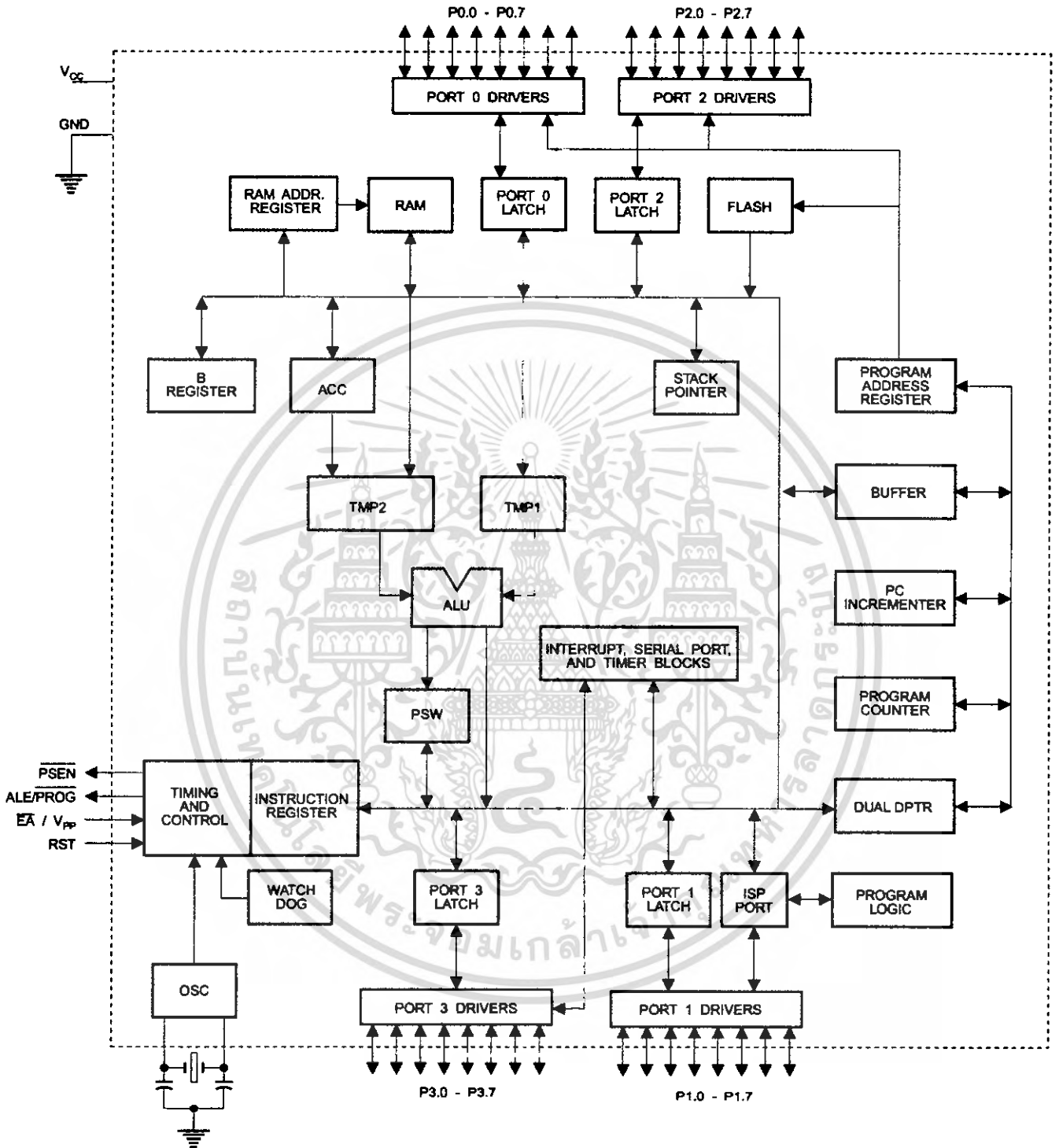
PLCC



TQFP



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port 1 also receives the low-order address bytes during Flash programming and verification.

| Port Pin | Alternate Functions |
|----------|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.5 | MOSI (used for In-System Programming) |
| P1.6 | MISO (used for In-System Programming) |
| P1.7 | SCK (used for In-System Programming) |

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to

external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89S52, as shown in the following table.

Port 3 also receives some control signals for Flash programming and verification.

| Port Pin | Alternate Functions |
|----------|---|
| P3.0 | RXD (serial input port) |
| P3.1 | TXD (serial output port) |
| P3.2 | $\overline{INT0}$ (external interrupt 0) |
| P3.3 | $\overline{INT1}$ (external interrupt 1) |
| P3.4 | T0 (timer 0 external input) |
| P3.5 | T1 (timer 1 external input) |
| P3.6 | \overline{WR} (external data memory write strobe) |
| P3.7 | \overline{RD} (external data memory read strobe) |

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device. This pin drives High for 96 oscillator periods after the Watchdog times out. The DISRTO bit in SFR AUXR (address 8EH) can be used to disable this feature. In the default state of bit DISRTO, the RESET HIGH out feature is enabled.

ALE/PROG

Address Latch Enable (ALE) is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke

new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers: Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 3) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

Interrupt Registers: The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Table 2. T2CON – Timer/Counter 2 Control Register

| | | | | | | | | |
|----------------------|-----|------|------|------|--------------------------|-----|------|--------|
| T2CON Address = 0C8H | | | | | Reset Value = 0000 0000B | | | |
| Bit Addressable | | | | | | | | |
| Bit | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | C/T2 | CP/RL2 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|--------|--|
| TF2 | Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1. |
| EXF2 | Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1). |
| RCLK | Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock. |
| TCLK | Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock. |
| EXEN2 | Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX. |
| TR2 | Start/Stop control for Timer 2. TR2 = 1 starts the timer. |
| C/T2 | Timer or counter select for Timer 2. C/T2 = 0 for timer function. C/T2 = 1 for external event counter (falling edge triggered). |
| CP/RL2 | Capture/Reload select. CP/RL2 = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/RL2 = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow. |

Table 3a. AUXR: Auxiliary Register

| | | | | | | | | | |
|------|---------------------|------------------------|---|---|--------|--------|---|---|--------|
| AUXR | Address = 8EH | Reset Value = XXX0XX0B | | | | | | | |
| | Not Bit Addressable | | | | | | | | |
| | | - | - | - | WDIDLE | DISRTO | - | - | DISALE |
| Bit | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

- Reserved for future expansion

DISALE Disable/Enable ALE

DISALE Operating Mode

0 ALE is emitted at a constant rate of 1/6 the oscillator frequency

1 ALE is active only during a MOVX or MOVC instruction

DISRTO Disable/Enable Reset out

DISRTO

0 Reset pin is driven High after WDT times out

1 Reset pin is input only

WDIDLE Disable/Enable WDT in IDLE mode

WDIDLE

0 WDT continues to count in IDLE mode

1 WDT halts counting in IDLE mode

Dual Data Pointer Registers: To facilitate accessing both internal and external data memory, two banks of 16-bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at 84H-85H. Bit DPS = 0 in SFR AUXR1 selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the

appropriate value before accessing the respective Data Pointer Register.

Power Off Flag: The Power Off Flag (POF) is located at bit 4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by reset.

Table 3b. AUXR1: Auxiliary Register 1

| | | | | | | | | | |
|-------|---------------------|-------------------------|---|---|---|---|---|-----|---|
| AUXR1 | Address = A2H | Reset Value = XXXXXXX0B | | | | | | | |
| | Not Bit Addressable | | | | | | | | |
| | | - | - | - | - | - | - | DPS | |
| Bit | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

- Reserved for future expansion

DPS Data Pointer Register Select

DPS

0 Selects DPTR Registers DP0L, DP0H

1 Selects DPTR Registers DP1L, DP1H



Watchdog Timer (One-time Enabled with Reset-out)

The WDT is intended as a recovery method in situations where the CPU may be subjected to software upsets. The WDT consists of a 13-bit counter and the Watchdog Timer Reset (WDTRST) SFR. The WDT is defaulted to disable from exiting reset. To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, it will increment every machine cycle while the oscillator is running. The WDT timeout period is dependent on the external clock frequency. There is no way to disable the WDT except through reset (either hardware reset or WDT overflow reset). When WDT overflows, it will drive an output RESET HIGH pulse at the RST pin.

Using the WDT

To enable the WDT, a user must write 01EH and 0E1H in sequence to the WDTRST register (SFR location 0A6H). When the WDT is enabled, the user needs to service it by writing 01EH and 0E1H to WDTRST to avoid a WDT overflow. The 13-bit counter overflows when it reaches 8191 (1FFFH), and this will reset the device. When the WDT is enabled, it will increment every machine cycle while the oscillator is running. This means the user must reset the WDT at least every 8191 machine cycles. To reset the WDT the user must write 01EH and 0E1H to WDTRST. WDTRST is a write-only register. The WDT counter cannot be read or written. When WDT overflows, it will generate an output RESET pulse at the RST pin. The RESET pulse duration is $96 \times TOSC$, where $TOSC = 1/FOSC$. To make the best use of the WDT, it should be serviced in those sections of code that will periodically be executed within the time required to prevent a WDT reset.

WDT During Power-down and Idle

In Power-down mode the oscillator stops, which means the WDT also stops. While in Power-down mode, the user does not need to service the WDT. There are two methods of exiting Power-down mode: by a hardware reset or via a level-activated external interrupt which is enabled prior to entering Power-down mode. When Power-down is exited with hardware reset, servicing the WDT should occur as it normally does whenever the AT89S52 is reset. Exiting Power-down with an interrupt is significantly different. The interrupt is held low long enough for the oscillator to stabilize. When the interrupt is brought high, the interrupt is serviced. To prevent the WDT from resetting the device while the interrupt pin is held low, the WDT is not started until the interrupt is pulled high. It is suggested that the WDT be reset during the interrupt service for the interrupt used to exit Power-down mode.

To ensure that the WDT does not overflow within a few states of exiting Power-down, it is best to reset the WDT just before entering Power-down mode.

Before going into the IDLE mode, the WDIDLE bit in SFR AUXR is used to determine whether the WDT continues to count if enabled. The WDT keeps counting during IDLE (WDIDLE bit = 0) as the default state. To prevent the WDT from resetting the AT89S52 while in IDLE mode, the user should always set up a timer that will periodically exit IDLE, service the WDT, and reenter IDLE mode.

With WDIDLE bit enabled, the WDT will stop to count in IDLE mode and resumes the count upon exit from IDLE.

UART

The UART in the AT89S52 operates the same way as the UART in the AT89C51 and AT89C52. For further information on the UART operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89S52 operate the same way as Timer 0 and Timer 1 in the AT89C51 and AT89C52. For further information on the timers' operation, refer to the ATMEL Web site (<http://www.atmel.com>). From the home page, select 'Products', then '8051-Architecture Flash Microcontroller', then 'Product Overview'.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit $C/T\bar{2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

Table 3. Timer 2 Operating Modes

| RCLK +TCLK | CP/RL2 | TR2 | MODE |
|------------|--------|-----|---------------------|
| 0 | 0 | 1 | 16-bit Auto-reload |
| 0 | 1 | 1 | 16-bit Capture |
| 1 | X | 1 | Baud Rate Generator |
| X | X | 0 | (Off) |



In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON.

Figure 5. Timer in Capture Mode

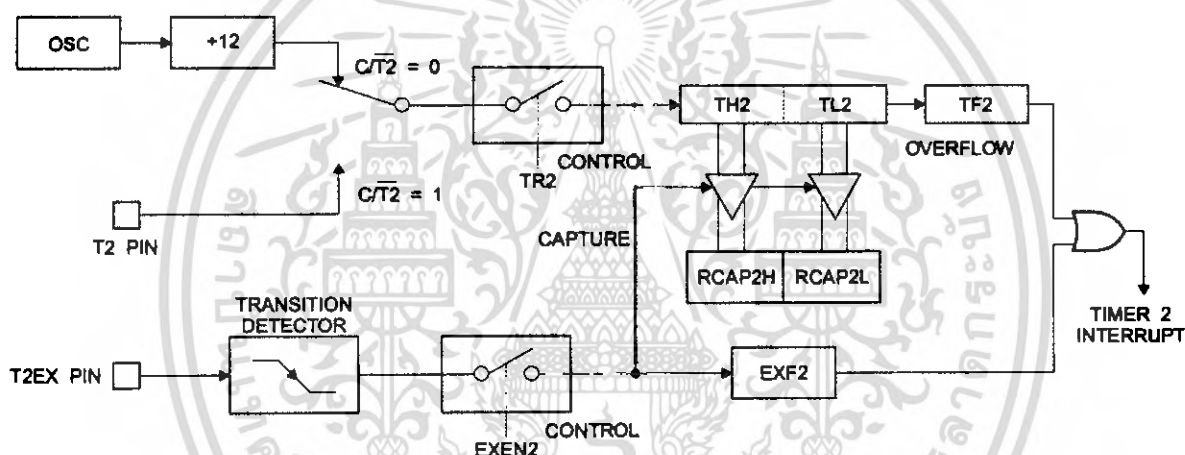


Figure 6 shows Timer 2 automatically counting up when DCEN=0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in Timer in Capture Mode RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 6. In this mode, the T2EX pin controls

This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 5.

Auto-reload (Up or Down Counter)

Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 9. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit $C/\bar{T}2$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

Interrupts

The AT89S52 has a total of six interrupt vectors: two external interrupts ($\overline{\text{INT0}}$ and $\overline{\text{INT1}}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 5 shows that bit position IE.6 is unimplemented. In the AT89S52, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

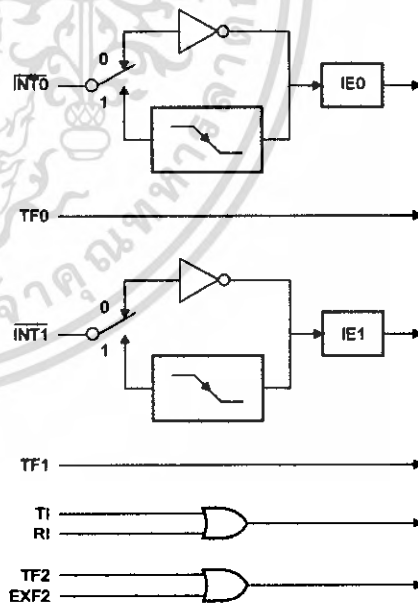
Table 5. Interrupt Enable (IE) Register

| (MSB) | | | | | | | | (LSB) |
|--|---|-----|----|-----|-----|-----|-----|-------|
| EA | — | ET2 | ES | ET1 | EX1 | ET0 | EX0 | |
| Enable Bit = 1 enables the interrupt. | | | | | | | | |
| Enable Bit = 0 disables the interrupt. | | | | | | | | |

| Symbol | Position | Function |
|--------|----------|---|
| EA | IE.7 | Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit. |
| — | IE.6 | Reserved. |
| ET2 | IE.5 | Timer 2 interrupt enable bit. |
| ES | IE.4 | Serial Port interrupt enable bit. |
| ET1 | IE.3 | Timer 1 interrupt enable bit. |
| EX1 | IE.2 | External interrupt 1 enable bit. |
| ET0 | IE.1 | Timer 0 interrupt enable bit. |
| EX0 | IE.0 | External interrupt 0 enable bit. |

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

Figure 10. Interrupt Sources



Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

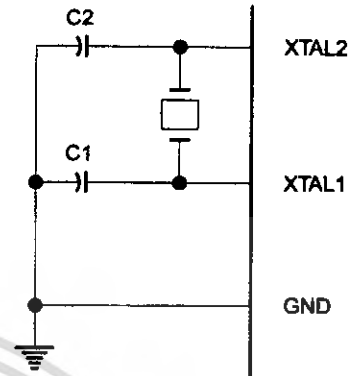
Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power-down Mode

In the Power-down mode, the oscillator is stopped, and the instruction that invokes Power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the Power-down mode is terminated. Exit from Power-down mode can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held

active long enough to allow the oscillator to restart and stabilize.

Figure 11. Oscillator Connections



Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 12. External Clock Drive Configuration

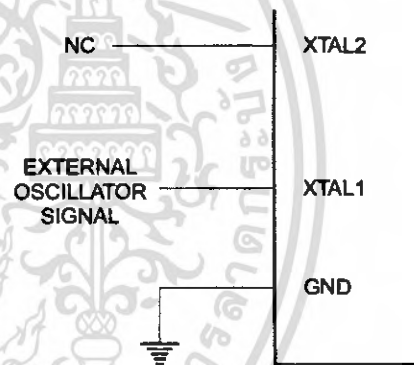


Table 6. Status of External Pins During Idle and Power-down Modes

| Mode | Program Memory | ALE | PSEN | PORT0 | PORT1 | PORT2 | PORT3 |
|------------|----------------|-----|------|-------|-------|---------|-------|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power-down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power-down | External | 0 | 0 | Float | Data | Data | Data |



Program Memory Lock Bits

The AT89S52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

Table 7. Lock Bit Protection Modes

| Program Lock Bits | | | | Protection Type |
|-------------------|-----|-----|---|--|
| LB1 | LB2 | LB3 | | |
| 1 | U | U | U | No program lock features |
| 2 | P | U | U | MOV _C instructions executed from external program memory are disabled from fetching code bytes from internal memory, \overline{EA} is sampled and latched on reset, and further programming of the Flash memory is disabled |
| 3 | P | P | U | Same as mode 2, but verify is also disabled |
| 4 | P | P | P | Same as mode 3, but external execution is also disabled |

When lock bit 1 is programmed, the logic level at the \overline{EA} pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of \overline{EA} must agree with the current logic level at that pin in order for the device to function properly.

Programming the Flash – Parallel Mode

The AT89S52 is shipped with the on-chip Flash memory array ready to be programmed. The programming interface needs a high-voltage (12-volt) program-enable signal and is compatible with conventional third-party Flash or EPROM programmers.

The AT89S52 code memory array is programmed byte-by-byte.

Programming Algorithm: Before programming the AT89S52, the address, data, and control signals should be set up according to the Flash programming mode table and Figures 13 and 14. To program the AT89S52, take the following steps:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise \overline{EA}/V_{PP} to 12V.
5. Pulse ALE/ \overline{PROG} once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 50 μ s.

Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89S52 features Data Polling to indicate the end of a byte write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P0.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The progress of byte programming can also be monitored by the RDY/ \overline{BSY} output signal. P3.0 is pulled low after ALE goes high during programming to indicate \overline{BSY} . P3.0 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The status of the individual lock bits can be verified directly by reading them back.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 100H, and 200H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (000H) = 1EH indicates manufactured by Atmel
- (100H) = 52H indicates 89S52
- (200H) = 06H

Chip Erase: In the parallel programming mode, a chip erase operation is initiated by using the proper combination of control signals and by pulsing ALE/ \overline{PROG} low for a duration of 200 ns - 500 ns.

In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 500 ms.

During chip erase, a serial read from any address location will return 00H at the data output.

Programming the Flash – Serial Mode

The Code memory array can be programmed using the serial ISP interface while RST is pulled to V_{CC} . The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before other operations can be executed. Before a reprogramming sequence can occur, a Chip Erase operation is required.

The Chip Erase operation turns the content of every memory location in the Code array into FFH.

Either an external system clock can be supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK)

frequency should be less than 1/16 of the crystal frequency. With a 33 MHz oscillator clock, the maximum SCK frequency is 2 MHz.

Serial Programming Algorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
Apply power between VCC and GND pins.
Set RST pin to "H".
If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.
2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time by supplying the address and data together with the

appropriate Write instruction. The write cycle is self-timed and typically takes less than 1 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

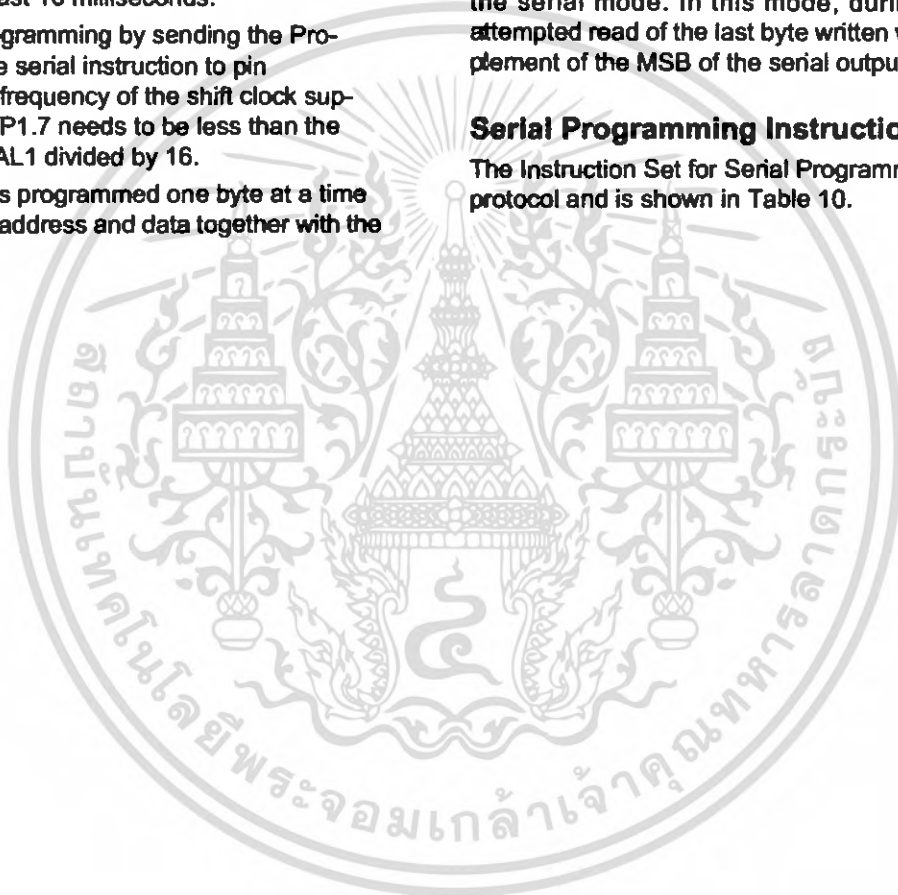
Power-off sequence (if needed):

- Set XTAL1 to "L" (if a crystal is not used).
- Set RST to "L".
- Turn V_{CC} power off.

Data Polling: The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

Serial Programming Instruction Set

The Instruction Set for Serial Programming follows a 4-byte protocol and is shown in Table 10.



Programming Interface – Parallel Mode

Every code byte in the Flash array can be programmed by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Table 8. Flash Programming Modes

| Mode | V _{cc} | RST | PSEN | ALE/ PROG | EA/ V _{pp} | P2.6 | P2.7 | P3.3 | P3.6 | P3.7 | P0.7-0 Data | Address | |
|---------------------------|-----------------|-----|------|--------------|------------------------|------|------|------|------|------|------------------------|---------|--------|
| | | | | | | | | | | | | P2.4-0 | P1.7-0 |
| Write Code Data | 5V | H | L | | 12V | L | H | H | H | H | D _{IN} | A12-8 | A7-0 |
| Read Code Data | 5V | H | L | H | H | L | L | L | H | H | D _{OUT} | A12-8 | A7-0 |
| Write Lock Bit 1 | 5V | H | L | | 12V | H | H | H | H | H | X | X | X |
| Write Lock Bit 2 | 5V | H | L | | 12V | H | H | H | L | L | X | X | X |
| Write Lock Bit 3 | 5V | H | L | | 12V | H | L | H | H | L | X | X | X |
| Read Lock Bits 1, 2, 3 | 5V | H | L | H | H | H | H | L | H | L | P0.2, P0.3, P0.4 | X | X |
| Chip Erase | 5V | H | L | | 12V | H | L | H | L | L | X | X | X |
| Read Atmel ID | 5V | H | L | H | H | L | L | L | L | L | 1EH | X 0000 | 00H |
| Read Device ID | 5V | H | L | H | H | L | L | L | L | L | 62H | X 0001 | 00H |
| Read Device ID | 5V | H | L | H | H | L | L | L | L | L | 06H | X 0010 | 00H |

- Notes:
1. Each $\overline{\text{PROG}}$ pulse is 200 ns - 500 ns for Chip Erase.
 2. Each $\overline{\text{PROG}}$ pulse is 200 ns - 500 ns for Write Code Data.
 3. Each $\overline{\text{PROG}}$ pulse is 200 ns - 500 ns for Write Lock Bits.
 4. $\text{RDY}/\overline{\text{BSY}}$ signal is output on P3.0 during programming.
 5. X = don't care.

Figure 13. Programming the Flash Memory (Parallel Mode)

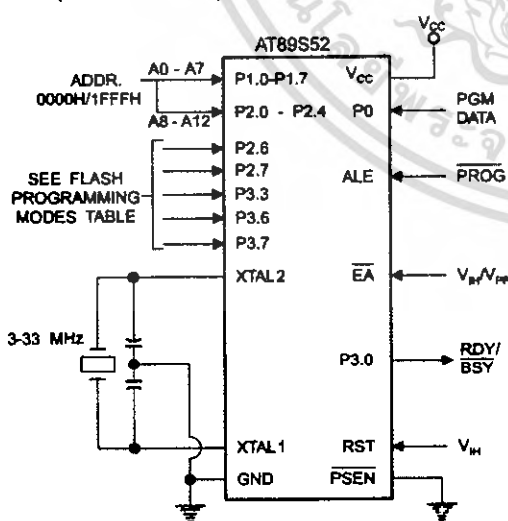
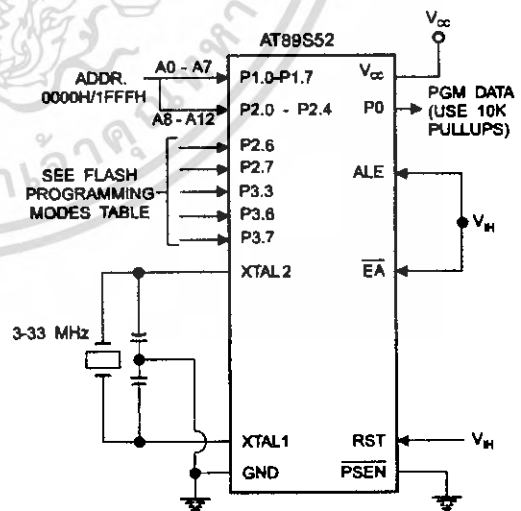


Figure 14. Verifying the Flash Memory (Parallel Mode)

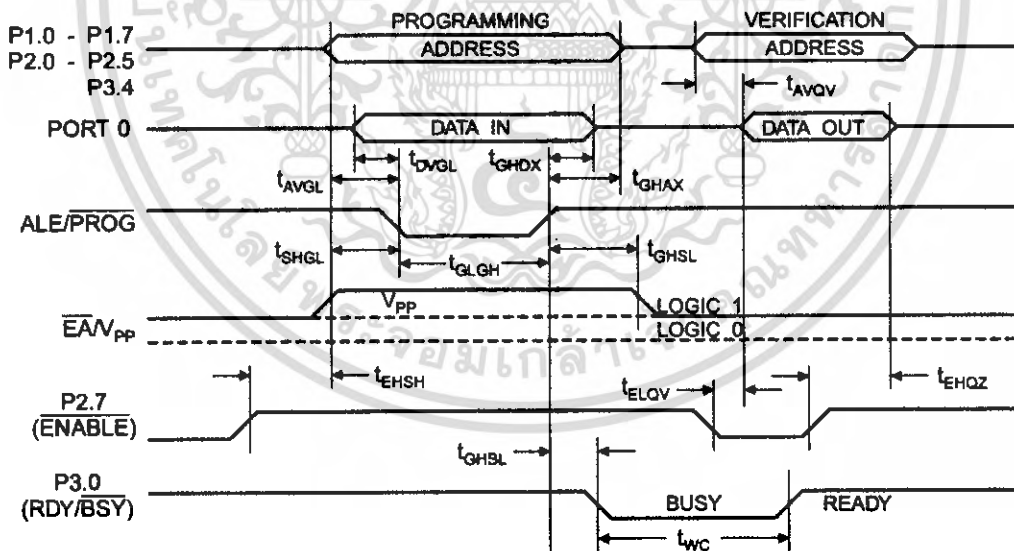


Flash Programming and Verification Characteristics (Parallel Mode)

$T_A = 20^\circ\text{C}$ to 30°C , $V_{CC} = 4.5$ to 5.5V

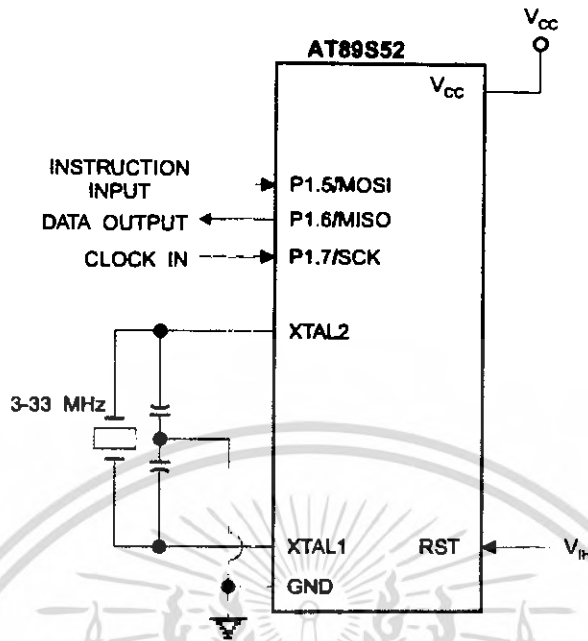
| Symbol | Parameter | Min | Max | Units |
|--------------|---|--------------|--------------|---------------|
| V_{PP} | Programming Supply Voltage | 11.5 | 12.5 | V |
| I_{PP} | Programming Supply Current | | 10 | mA |
| I_{CC} | V_{CC} Supply Current | | 30 | mA |
| $1/f_{CLCL}$ | Oscillator Frequency | 3 | 33 | MHz |
| t_{AVGL} | Address Setup to $\overline{\text{PROG}}$ Low | $48t_{CLCL}$ | | |
| t_{GHAX} | Address Hold After $\overline{\text{PROG}}$ | $48t_{CLCL}$ | | |
| t_{DVGL} | Data Setup to $\overline{\text{PROG}}$ Low | $48t_{CLCL}$ | | |
| t_{GHDX} | Data Hold After $\overline{\text{PROG}}$ | $48t_{CLCL}$ | | |
| t_{EHS} | P2.7 (ENABLE) High to V_{PP} | $48t_{CLCL}$ | | |
| t_{SHGL} | V_{PP} Setup to $\overline{\text{PROG}}$ Low | 10 | | μs |
| t_{GHSL} | V_{PP} Hold After $\overline{\text{PROG}}$ | 10 | | μs |
| t_{GLGH} | $\overline{\text{PROG}}$ Width | 0.2 | 1 | μs |
| t_{AVQV} | Address to Data Valid | | $48t_{CLCL}$ | |
| t_{ELQV} | ENABLE Low to Data Valid | | $48t_{CLCL}$ | |
| t_{EHQZ} | Data Float After ENABLE | 0 | $48t_{CLCL}$ | |
| t_{GHBL} | $\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low | | 1.0 | μs |
| t_{WC} | Byte Write Cycle Time | | 50 | μs |

Figure 15. Flash Programming and Verification Waveforms – Parallel Mode



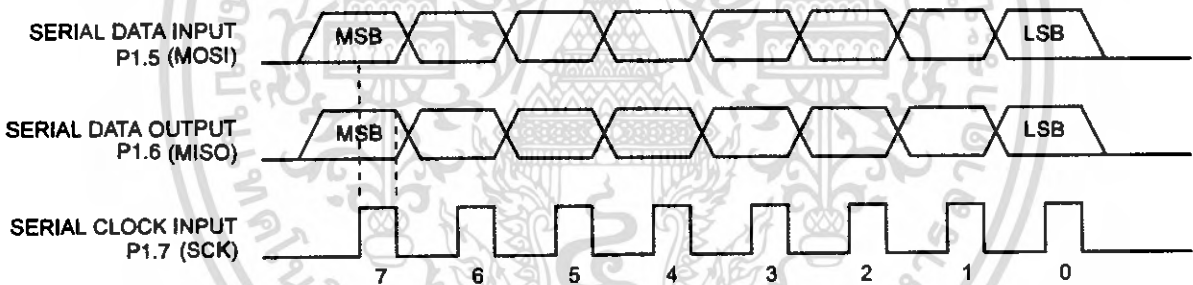
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 16. Flash Memory Serial Downloading



Flash Programming and Verification Waveforms – Serial Mode

Figure 17. Serial Programming Waveforms





Serial Programming Characteristics

Figure 18. Serial Programming Timing

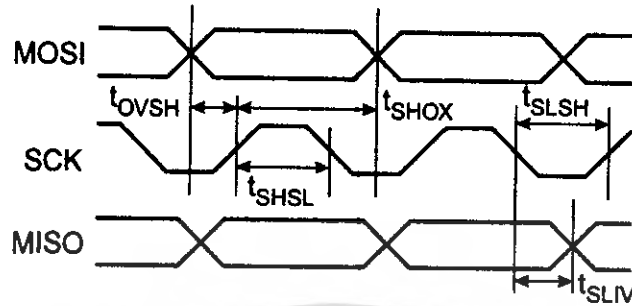


Table 10. Serial Programming Characteristics, $T_A = -40^\circ\text{C}$ to 85°C , $V_{CC} = 4.0 - 5.5\text{V}$ (Unless otherwise noted)

| Symbol | Parameter | Min | Typ | Max | Units |
|--------------|-----------------------------------|--------------|-----|---------------------|---------------|
| $1/t_{CLCL}$ | Oscillator Frequency | 0 | | 33 | MHz |
| t_{CLCL} | Oscillator Period | 30 | | | ns |
| t_{SHSL} | SCK Pulse Width High | $2 t_{CLCL}$ | | | ns |
| t_{SLSH} | SCK Pulse Width Low | $2 t_{CLCL}$ | | | ns |
| t_{OVSH} | MOSI Setup to SCK High | t_{CLCL} | | | ns |
| t_{SHOX} | MOSI Hold after SCK High | $2 t_{CLCL}$ | | | ns |
| t_{SLIV} | SCK Low to MISO Valid | 10 | 16 | 32 | ns |
| t_{ERASE} | Chip Erase Instruction Cycle Time | | | 500 | ms |
| t_{SWC} | Serial Byte Write Cycle Time | | | $64 t_{CLCL} + 400$ | μs |

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..... | 6.6V |
| DC Output Current..... | 15.0 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.

DC Characteristics

The values shown in this table are valid for $T_A = -40^\circ\text{C}$ to 85°C and $V_{CC} = 4.0\text{V}$ to 5.5V , unless otherwise noted.

| Symbol | Parameter | Condition | Min | Max | Units |
|-----------|--|--|------------------|------------------|------------------|
| V_{IL} | Input Low Voltage | (Except $\bar{E}A$) | -0.5 | $0.2 V_{CC}-0.1$ | V |
| V_{IL1} | Input Low Voltage ($\bar{E}A$) | | -0.5 | $0.2 V_{CC}-0.3$ | V |
| V_{IH} | Input High Voltage | (Except XTAL1, RST) | $0.2 V_{CC}+0.9$ | $V_{CC}+0.5$ | V |
| V_{IH1} | Input High Voltage | (XTAL1, RST) | $0.7 V_{CC}$ | $V_{CC}+0.5$ | V |
| V_{OL} | Output Low Voltage ⁽¹⁾ (Ports 1,2,3) | $I_{OL} = 1.6 \text{ mA}$ | | 0.45 | V |
| V_{OL1} | Output Low Voltage ⁽¹⁾ (Port 0, ALE, $\bar{P}SEN$) | $I_{OL} = 3.2 \text{ mA}$ | | 0.45 | V |
| V_{OH} | Output High Voltage (Ports 1,2,3, ALE, $\bar{P}SEN$) | $I_{OH} = -80 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$ | 2.4 | | V |
| | | $I_{OH} = -25 \mu\text{A}$ | $0.75 V_{CC}$ | | V |
| | | $I_{OH} = -10 \mu\text{A}$ | $0.9 V_{CC}$ | | V |
| V_{OH1} | Output High Voltage (Port 0 in External Bus Mode) | $I_{OH} = -800 \mu\text{A}, V_{CC} = 5\text{V} \pm 10\%$ | 2.4 | | V |
| | | $I_{OH} = -300 \mu\text{A}$ | $0.75 V_{CC}$ | | V |
| | | $I_{OH} = -80 \mu\text{A}$ | $0.9 V_{CC}$ | | V |
| I_L | Logical 0 Input Current (Ports 1,2,3) | $V_{IN} = 0.45\text{V}$ | | -50 | μA |
| I_{TL} | Logical 1 to 0 Transition Current (Ports 1,2,3) | $V_{IN} = 2\text{V}, V_{CC} = 5\text{V} \pm 10\%$ | | -650 | μA |
| I_{LI} | Input Leakage Current (Port 0, $\bar{E}A$) | $0.45 < V_{IN} < V_{CC}$ | | ± 10 | μA |
| RRST | Reset Pulldown Resistor | | 10 | 30 | $\text{K}\Omega$ |
| C_{IO} | Pin Capacitance | Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$ | | 10 | pF |
| I_{CC} | Power Supply Current | Active Mode, 12 MHz | | 25 | mA |
| | | Idle Mode, 12 MHz | | 6.5 | mA |
| | Power-down Mode ⁽¹⁾ | $V_{CC} = 5.5\text{V}$ | | 50 | μA |

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:

Maximum I_{OL} per port pin: 10 mA

Maximum I_{OL} per 8-bit port:

Port 0: 26 mA Ports 1, 2, 3: 15 mA

Maximum total I_{OL} for all output pins: 71 mA

If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.

2. Minimum V_{CC} for Power-down is 2V.



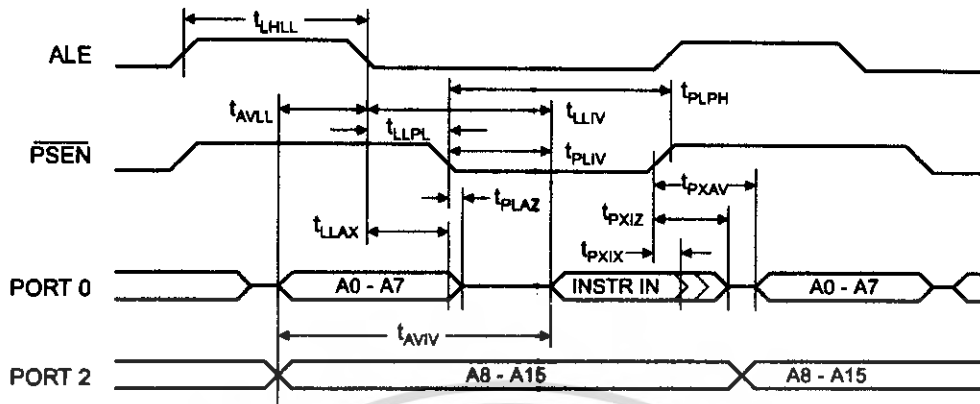
AC Characteristics

Under operating conditions, load capacitance for Port 0, ALE/ $\overline{\text{PROG}}$, and $\overline{\text{PSEN}}$ = 100 pF; load capacitance for all other outputs = 80 pF.

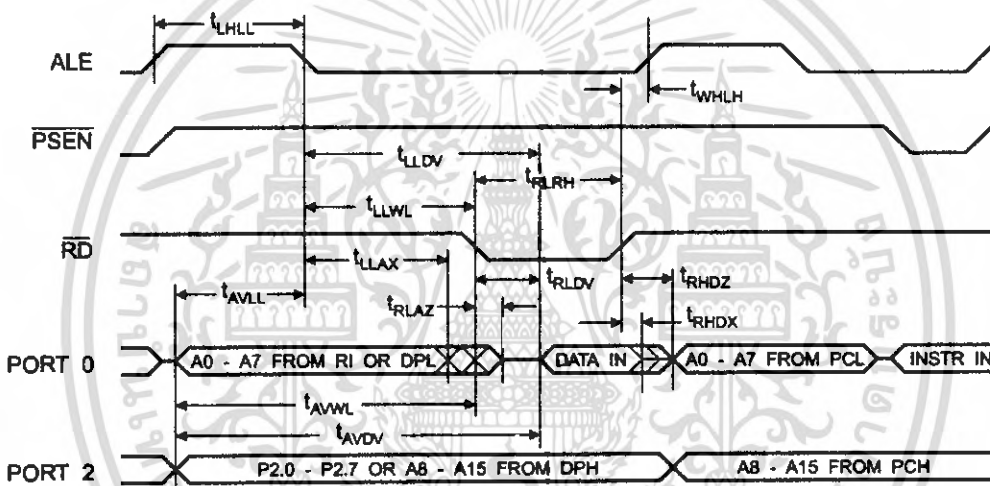
External Program and Data Memory Characteristics

| Symbol | Parameter | 12 MHz Oscillator | | Variable Oscillator | | Units |
|---------------------|---|-------------------|-----|------------------------|------------------------|-------|
| | | Min | Max | Min | Max | |
| $1/t_{\text{CLCL}}$ | Oscillator Frequency | | | 0 | 33 | MHz |
| t_{LHL} | ALE Pulse Width | 127 | | $2t_{\text{CLCL}}-40$ | | ns |
| t_{AVLL} | Address Valid to ALE Low | 43 | | $t_{\text{CLCL}}-25$ | | ns |
| t_{LAX} | Address Hold After ALE Low | 48 | | $t_{\text{CLCL}}-25$ | | ns |
| t_{LLIV} | ALE Low to Valid Instruction In | | 233 | | $4t_{\text{CLCL}}-65$ | ns |
| t_{LLPL} | ALE Low to $\overline{\text{PSEN}}$ Low | 43 | | $t_{\text{CLCL}}-25$ | | ns |
| t_{PLPH} | $\overline{\text{PSEN}}$ Pulse Width | 205 | | $3t_{\text{CLCL}}-45$ | | ns |
| t_{PLIV} | $\overline{\text{PSEN}}$ Low to Valid Instruction In | | 145 | | $3t_{\text{CLCL}}-60$ | ns |
| t_{PIX} | Input Instruction Hold After $\overline{\text{PSEN}}$ | 0 | | 0 | | ns |
| t_{PIXZ} | Input Instruction Float After $\overline{\text{PSEN}}$ | | 59 | | $t_{\text{CLCL}}-25$ | ns |
| t_{PXAV} | $\overline{\text{PSEN}}$ to Address Valid | 75 | | $t_{\text{CLCL}}-8$ | | ns |
| t_{AVIV} | Address to Valid Instruction In | | 312 | | $5t_{\text{CLCL}}-80$ | ns |
| t_{PLAZ} | $\overline{\text{PSEN}}$ Low to Address Float | | 10 | | 10 | ns |
| t_{RLRH} | $\overline{\text{RD}}$ Pulse Width | 400 | | $6t_{\text{CLCL}}-100$ | | ns |
| t_{WLWH} | $\overline{\text{WR}}$ Pulse Width | 400 | | $6t_{\text{CLCL}}-100$ | | ns |
| t_{RLDV} | $\overline{\text{RD}}$ Low to Valid Data In | | 252 | | $5t_{\text{CLCL}}-90$ | ns |
| t_{RHDX} | Data Hold After $\overline{\text{RD}}$ | 0 | | 0 | | ns |
| t_{RHDXZ} | Data Float After $\overline{\text{RD}}$ | | 97 | | $2t_{\text{CLCL}}-28$ | ns |
| t_{LLDV} | ALE Low to Valid Data In | | 517 | | $8t_{\text{CLCL}}-150$ | ns |
| t_{AVDV} | Address to Valid Data In | | 585 | | $9t_{\text{CLCL}}-165$ | ns |
| t_{LLWL} | ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low | 200 | 300 | $3t_{\text{CLCL}}-50$ | $3t_{\text{CLCL}}+50$ | ns |
| t_{AVWL} | Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low | 203 | | $4t_{\text{CLCL}}-75$ | | ns |
| t_{QVWX} | Data Valid to $\overline{\text{WR}}$ Transition | 23 | | $t_{\text{CLCL}}-30$ | | ns |
| t_{QVWH} | Data Valid to $\overline{\text{WR}}$ High | 433 | | $7t_{\text{CLCL}}-130$ | | ns |
| t_{WHQX} | Data Hold After $\overline{\text{WR}}$ | 33 | | $t_{\text{CLCL}}-25$ | | ns |
| t_{RLAZ} | $\overline{\text{RD}}$ Low to Address Float | | 0 | | 0 | ns |
| t_{WHLH} | $\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High | 43 | 123 | $t_{\text{CLCL}}-25$ | $t_{\text{CLCL}}+25$ | ns |

External Program Memory Read Cycle



External Data Memory Read Cycle



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้