

เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านทางอินเทอร์เน็ต

ELECTRIC EQUIPMENT CONTROL

VIA

INTERNET



โดย

นายธรรมนุญ

ยะไชยศรี

นายวิเชียร

มัจฉา

นายสมาน

ไชยชิต

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....

เลขทะเบียน..... 37093

วัน, เดือน, ปี - 4 ก.ย. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านทางอินเทอร์เน็ต

ELECTRIC EQUIPMENT CONTROL

VIA

INTERNET



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2542

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
เรื่อง เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านทางอินเทอร์เน็ต

**ELECTRIC EQUIPMENT CONTROL VIA INTERNET**

ผู้จัดทำ

- |                        |          |
|------------------------|----------|
| 1. นายธรรมบุญ ยะไชยศรี | 40013008 |
| 2. นายวิเชียร มัจฉา    | 40013022 |
| 3. นายสมาน ไชยจิต      | 40013028 |

  
..... อาจารย์ที่ปรึกษา  
( รศ.ดร. กอบชัย เดชหาญ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านทางอินเทอร์เน็ต  
ELECTRIC EQUIPMENT CONTROL VIA INTERNET

โดย นายธรรมนุญ ยะไชยศรี 40013008  
นายวิเชียร มัจฉา 40013022  
นายสมาน ไชยชิต 40013028

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ

**บทคัดย่อ**

โครงการนี้เป็นการเขียนโปรแกรมประยุกต์ เพื่อใช้ในการสั่งงานเครื่องใช้ไฟฟ้าได้ 8 ช่องสัญญาณ ซึ่งแบ่งการทำงานออกเป็น 3 ส่วนทางด้านโปรแกรม คือ โปรแกรมโฮมเพจ โดยใช้ HTML ร่วมกับ เอเอสพี (ASP : Active Server Pages) โปรแกรมควบคุมอุปกรณ์ไฟฟ้าทางด้านเซิร์ฟเวอร์ โดยใช้ วินโดวส์ เอนที เซิร์ฟเวอร์ และ ไอไอเอส (IIS : Internet Information Server) ร่วมกับ วิววลเบสิก 6 โปรแกรมควบคุมไมโครคอนโทรลเลอร์ ในการพัฒนาทางด้านโปรแกรม ส่วนทางด้านฮาร์ดแวร์คือวงจรควบคุมอุปกรณ์ไฟฟ้า โดยใช้ ไมโครคอนโทรลเลอร์ MCS-51 ในการพัฒนา

**ABSTRACT**

This project presents an applied software to control the 8 channel electrical equipments. The program is divided into 3 parts, homepage program using HTML incorporating with ASP (Active Server Pages), electrical equipment controlled program at server and using Windows NT Server and IIS (Internet Information Server) with Visual Basic 6, microcontroller controlled program. The programs are developed for using in this project. The hardware part is electrical equipment controlled circuit using MCS-51 to develope.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการทำงาน	3
2.1 เว็บเซิร์ฟเวอร์	3
2.2 เอเอสพี	9
2.3 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	14
2.4 การสื่อสารพอร์ตอนุกรม RS 232	21
บทที่ 3 การคำนวณและการสร้าง	27
3.1 แสดงหลักการทำงาน	27
3.2 แผนผังการทำงานเครื่องใช้ไฟฟ้า	28
3.2.1 บล็อกไดอะแกรมแสดงการทำงานในส่วน เอเอสพี	28
3.2.2 แผนผังการทำงานในส่วน เอเอสพี	31
3.3 แสดงหลักการทำงานในส่วน วิชาวลเบติก	33
3.3.1 บล็อกไดอะแกรมการทำงานในส่วนวิชาวลเบติก	33
3.3.2 แผนผังการทำงานในส่วน วิชาวลเบติก	34
3.4 แสดงหลักการทำงานของไมโครคอนโทรลเลอร์	38
3.4.1 แผนผังการทำงานของไมโครคอนโทรลเลอร์ตระกูล MCS-51	38
3.4.2 บล็อกไดอะแกรมในส่วนไมโครคอนโทรลเลอร์ตระกูล MCS-51	39
3.4.3 วงจรการทำงานของไมโครคอนโทรลเลอร์	40
3.4.4 ส่วนของวงจรควบคุมอุปกรณ์ไฟฟ้า	42
3.5 ส่วนของวงจรภาคจ่ายไฟ	43
3.6 แสดงการเชื่อมโยงฐานข้อมูล OBDC	43
3.7 ส่วนการออกแบบฐานข้อมูล	44
บทที่ 4 การทดลองและผลการทดลอง	48
4.1 ส่วนของโสมเพจ	48
4.1.1 หน้าต่างล็อกอิน	48
4.1.2 หน้าต่างเปลี่ยนพาสเวิร์ด	49
4.1.3 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า	50
4.1.4 หน้าต่างพาสเวิร์ดผิดพลาด	51
4.1.5 หน้าต่างล็อกเอาท์	51
4.2 ส่วนของโปรแกรมวิชาวลเบติก	52
4.2.1 หน้าต่างพาสเวิร์ด	52
4.2.2 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
4.2.3 หน้าต่างกำหนดคุณสมบัติ	53
4.2.4 หน้าต่างเปลี่ยนพาสเวิร์ด	53
บทที่ 5 บทวิจารณ์และสรุป	54
ภาคผนวก ก	56
ภาคผนวก ข	80
ภาคผนวก ค	108
กิตติกรรมประกาศ	109
หนังสืออ้างอิง	110



## สารบัญรูป

	หน้า
รูปที่ 2.1 แสดงภาพรวมของ เวิร์ด ไวค์ เว็บ	3
รูปที่ 2.2 ขบวนการทำงานของ เอเอสพี	10
รูปที่ 2.3 โครงสร้างทางด้าน เอเอสพี	11
รูปที่ 2.4 การจัดวางฯฯ 8051	16
รูปที่ 2.5 แสดง โครงสร้างภายใน MCS 51	18
รูปที่ 2.6 การจัดหน่วยความจำ	20
รูปที่ 3.1 บล็อก ไดอะแกรมรวมของ โครงการงาน	27
รูปที่ 3.2 บล็อก ไดอะแกรมการทำงานในส่วน เอเอสพี	30
รูปที่ 3.3 ก แสดงผังการทำงาน เอเอสพี	31
รูปที่ 3.3 ข แสดงผังการทำงาน เอเอสพี	32
รูปที่ 3.4 แสดงบล็อก ไดอะแกรมการทำงานของวิซวลเบสิก	33
รูปที่ 3.5 ก ผังการทำงานในส่วนวิซวลเบสิก	34
รูปที่ 3.5 ข ผังการทำงานในส่วนวิซวลเบสิก	35
รูปที่ 3.5 ค ผังการทำงานในส่วนวิซวลเบสิก	36
รูปที่ 3.5 ง ผังการทำงานในส่วนวิซวลเบสิก	37
รูปที่ 3.6 ผังการทำงานในส่วนไมโครคอนโทรลเลอร์	38
รูปที่ 3.7 แสดงบล็อก ไดอะแกรมระบบไมโครคอนโทรลเลอร์	39
รูปที่ 3.8 วงจรควบคุมอุปกรณ์ไฟฟ้าทางด้านส่ง	41
รูปที่ 3.9 วงจรควบคุมอุปกรณ์ไฟฟ้าทางด้านรับ	42
รูปที่ 3.10 แสดงวงจรภาคจ่ายไฟ	43
รูปที่ 3.11 แสดงการตั้งค่าการเชื่อมโยงฐานข้อมูล ODBC	43
รูปที่ 3.12 แสดงการตั้งค่าฐานข้อมูลเมื่อเสร็จสมบูรณ์	44
รูปที่ 3.13 แสดงโครงสร้างของฐานข้อมูล	45
รูปที่ 3.14 แสดงตาราง Users	45
รูปที่ 3.15 แสดงตาราง Channel	46
รูปที่ 3.16 แสดงตาราง คุณสมบัติ	47
รูปที่ 4.1 หน้าต่างล็อกอิน	48
รูปที่ 4.2 หน้าต่างการเปลี่ยนพาสเวิร์ด	49
รูปที่ 4.3 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า	50
รูปที่ 4.4 หน้าต่างเมื่อพาสเวิร์ดผิดพลาด	51
รูปที่ 4.5 หน้าต่างล็อกเอาท์	51
รูปที่ 4.6 หน้าต่าง พาสเวิร์ด	52

## สารบัญรูป (ต่อ)

	หน้า
รูปที่ 4.7 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า	52
รูปที่ 4.8 หน้าต่างคุณสมบัติ	53
รูปที่ 4.9 หน้าต่างเปลี่ยนพาสเวิร์ด	53



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 1

### บทนำ

สาขาโทรคมนาคมเป็นสาขาที่เจริญก้าวหน้าไปค่อนข้างรวดเร็ว โดยเฉพาะทางด้านอินเทอร์เน็ต ซึ่งจะเป็นผลโดยตรงในการนำไปใช้ในชีวิตประจำวัน และปัจจุบันก็ถูกนำมาใช้อย่างกว้างขวาง เพราะฉะนั้นถ้าเรามีโอกาสศึกษา ก็จะเป็นประโยชน์ต่อตัวเราในอนาคต และเป็นการนำเอาความรู้ที่ได้ศึกษา มาประยุกต์ใช้กับชีวิตประจำวัน ซึ่งอนาคตอันใกล้นี้คาดว่าจะมีผู้ใช้อินเทอร์เน็ตมากขึ้น เราอาจนำเครื่องควบคุมอุปกรณ์ไฟฟ้าไปใช้ในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านของเราในขณะที่เราไม่อยู่บ้านได้ ซึ่งถ้าเรามีความรู้ทางด้านนี้เราสามารถคัดแปลงโครงงานนี้ไปใช้งานทางด้านอื่นได้ เช่น การค้าอิเล็กทรอนิกส์ หรือเราสามารถเข้าถึงข้อมูลในเครื่องของเราได้ในขณะที่เราอยู่นอกบ้าน

โดยโครงสร้างของโครงงานมี 2 ส่วนใหญ่ๆ ดังนี้ คือ

#### 1. ส่วนซอฟต์แวร์ ประกอบด้วย

- 1.1 โปรแกรมเชื่อมต่ออินเทอร์เน็ต
- 1.2 โปรแกรมเขียน โสมเพจ
- 1.3 โปรแกรมควบคุมอุปกรณ์ไฟฟ้า
- 1.4 โปรแกรมควบคุม แอลซีดี
- 1.5 โปรแกรมฐานข้อมูล

#### 2. ส่วนฮาร์ดแวร์ ประกอบด้วย

- 2.1 ส่วนของวงจรควบคุมการเปิดเปิดอุปกรณ์ไฟฟ้า
- 2.2 ส่วนของระบบไมโครคอนโทรลเลอร์
- 2.3 ส่วนของวงจรแสดงผลแบบแอลซีดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ขั้นตอนการดำเนินงาน

ภาคการศึกษา 1/2542 ทำในแต่ละส่วน ดังต่อไปนี้

#### ส่วนฮาร์ดแวร์

- ส่วนของวงจรควบคุมปิด เปิดอุปกรณ์ไฟฟ้า
- ส่วนของระบบไมโครคอนโทรลเลอร์

#### ส่วนซอฟต์แวร์

- โปรแกรมควบคุมอุปกรณ์ไฟฟ้า

ภาคการศึกษา 2/2542 ทำในแต่ละส่วน ดังต่อไปนี้

#### ส่วนฮาร์ดแวร์

- เพิ่มเติมแก้ไขวงจรควบคุมการปิด เปิดอุปกรณ์ไฟฟ้า
- ประกอบลงกล่อง
- ทดลองใช้งานจริงกับอินเทอร์เน็ต

#### ส่วนซอฟต์แวร์

- ส่วนของโฮมเพจ
- ส่วนฐานข้อมูล
- ส่วนควบคุมอุปกรณ์ไฟฟ้าทางดี้นเซิร์ฟเวอร์

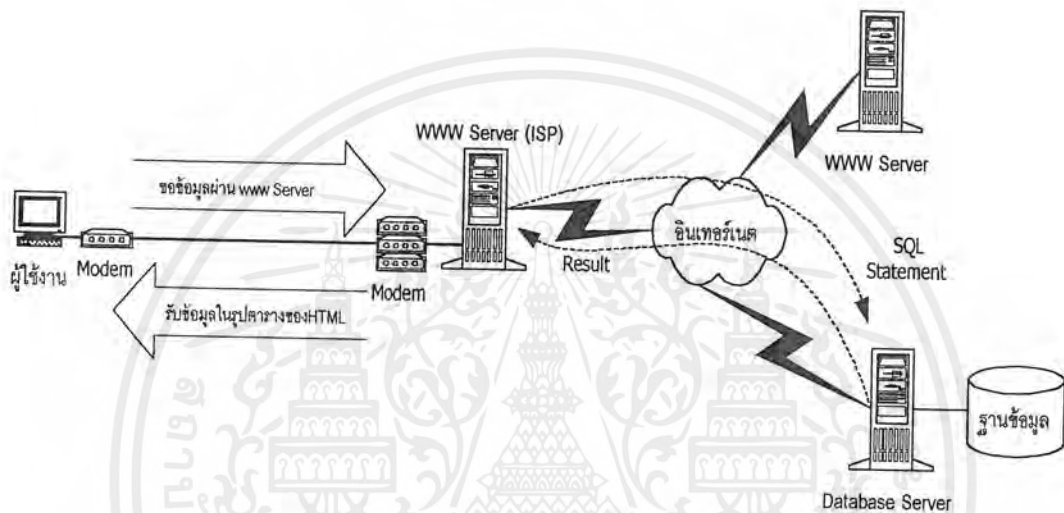
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและหลักการทำงาน

#### 2.1 เว็บเซิร์ฟเวอร์

เว็บเซิร์ฟเวอร์ คือ แอปพลิเคชันทำหน้าที่รับ และประมวลผลเอกสาร ที่ถูกร้องขอจากผู้ให้บริการอินเทอร์เน็ต จากนั้นเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารกลับไปแสดงผลให้ผู้ให้บริการผ่านบราวเซอร์ นอกจากนี้เว็บเซิร์ฟเวอร์จะถูกนำมาให้บริการในอินเทอร์เน็ตแล้ว แต่อาจมีการประยุกต์ให้นำมาใช้กับเครือข่ายภายในองค์กร หรือ อินเทอร์เน็ตได้เช่นกัน



รูปที่ 2.1 แสดงภาพรวมของ เวิลด์ ไวด์ เว็บ  
ซึ่งเป็นบริการที่ได้รับความนิยมสูงสุดของอินเทอร์เน็ต

แต่เดิมนั้นเว็บเซิร์ฟเวอร์มักจะอยู่ในเครื่องคอมพิวเตอร์ ยูนิกซ์ ที่มีประสิทธิภาพสูง และราคาแพง ต่อมาเมื่ออินเทอร์เน็ตขยายความนิยมมาสู่ผู้ใช้ พีซี ทำให้มีการพัฒนาซอฟต์แวร์ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์บนพีซี ซึ่งสามารถรันได้ทั้งวินโดวส์ 95/98 และ วินโดวส์เอ็นที เซิร์ฟเวอร์/Workstation ตัวอย่างเช่น

- NCSA Web Server จาก NSCA เป็นเว็บเซิร์ฟเวอร์ที่สามารถดาวน์โหลดได้ฟรีจากเว็บไซต์ที่ให้บริการ ดาวน์โหลด ฟรีทั่วไป
- Net Server จาก Netscape เป็น เวิลด์ ไวด์ เว็บ (World Wide Web : www) เซิร์ฟเวอร์ (Server) ที่มีความสามารถรองรับ จาวา (JAVA) ได้อย่างเต็มรูปแบบ
- ออราเคิลเว็บเซิร์ฟเวอร์ (Oracle Web Server) จากออราเคิล เป็นเว็บเซิร์ฟเวอร์ที่เน้นความสามารถด้านการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูล โดยเฉพาะการใช้งานร่วมกับระบบจัดการฐานข้อมูลของ ออราเคิล ปัจจุบันเราจะได้ยินชื่อที่ตั้งขึ้นใหม่เป็น แอปพลิเคชัน เซิร์ฟเวอร์ ( Application Server ) (คือ ออราเคิล เว็บ เซิร์ฟเวอร์ ตัวเดิมที่ถูกพัฒนาขึ้นมา)
- Personal Web Server จากไมโครซอฟท์เป็นเว็บเซิร์ฟเวอร์ที่สามารถใช้งานได้กับ วินโดวส์ 95/98 หรือ วินโดวส์ เอ็นที เวิร์คสแตชัน และรองรับการใช้งานร่วมกับ เอเอสพี (Active Server Page) โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถดาวน์โหลดมาใช้ได้ฟรีได้จากเว็บไซต์ของไมโครซอฟท์ แต่ถ้าใช้งาน วินโดวส์ 98 ก็ สามารถเลือกติดตั้งได้ ซึ่งมักใช้ในการทดสอบเว็บเพจ หรือแอปพลิเคชันอินเทอร์เน็ต ก่อนจะนำไป ใช้งานจริง

- IIS (Internet Information Server) จากไมโครซอฟท์ เป็นอินเทอร์เน็ต เซิร์ฟเวอร์เวอร์ชัน 2 ที่แถม มากับ วินโดวส์ เอนที เซิร์ฟเวอร์ 4.0 มีความสามารถให้บริการได้ทั้ง เวิลด์ ไวด์ เว็บ ,เอฟทีพี (FTP:File Transfer Protocol) และ Gopher ส่วนเวอร์ชันที่ใช้งานกับเอเอสพี ได้จะเป็นเวอร์ชัน 3.0 ขึ้นไป สำหรับบั้นนี้จะแสดงการใช้งานกับเวอร์ชัน 4.0

### ระบบเครือข่ายอินเทอร์เน็ต

ในองค์กรหนึ่ง ๆ อาจมีการติดตั้งระบบเครือข่ายแบบไกลตั้งแต่ 2 ระบบขึ้นไปแต่เป็นระบบที่อยู่ ห่างไกลกันมาก เช่น อยู่คนละจังหวัด ระบบเครือข่ายแบบไกลแต่ละระบบก็สามารถถูกเชื่อมโยงเข้าด้วยกันจนกลายเป็นเครือข่ายที่เรียกว่า เครือข่ายระยะไกล (WAN : Wide Area Network) และนอกจากนี้ระบบ เครือข่ายระยะไกล ที่หนึ่งก็สามารถเชื่อมกับ เครือข่ายระยะไกล ที่อยู่ห่างไกลกันออกไปมาก ๆ ได้อีก เช่น อยู่คนละประเทศหรือคนละทวีป ทำให้เกิดระบบเครือข่ายขนาดใหญ่ที่เรียกว่า “Internetworking” ซึ่งเป็นระบบเครือข่ายขนาดใหญ่และเป็นหลักการที่กลายมาเป็นระบบเครือข่ายอินเทอร์เน็ตในที่สุด

อินเทอร์เน็ต เป็นเครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่มาก เกิดจากการเชื่อมต่อเครือข่าย คอมพิวเตอร์จำนวนมากมายในโลกลงเข้าด้วยกัน ไม่ว่าจะ เป็นเครือข่ายขนาดเล็ก เช่น ระบบเครือข่ายแบบ ใกล้เคียงหรือระบบเครือข่ายขนาดใหญ่เช่นระบบเครือข่ายของมินิหรือเมนเฟรมคอมพิวเตอร์ซึ่งแต่ละเครือข่าย ก็จะมีเครื่องคอมพิวเตอร์ที่เป็นเครื่องแม่ข่าย หรือ โฮสต์ (Host) ซึ่งมีอยู่หลายชนิดหลายยี่ห้อ โดยจะมีการ กำหนดข้อตกลงใจการสื่อสารที่เรียกว่า โปรโตคอล (Protocol) ขึ้นมา เพื่อให้คอมพิวเตอร์แต่ละชนิด สามารถติดต่อสื่อสารกันได้ โปรโตคอลมาตรฐานที่ใช้ในการสื่อสารบนอินเทอร์เน็ตจะมีชื่อเรียกว่า TCP/IP (Transmission Control Protocol/Internet Protocol)

ถ้าเราเปรียบให้โปรโตคอลเหมือนภาษาที่ใช้ในการสื่อสารของมนุษย์ซึ่งภาษามนุษย์นั้นมีมาก มายหลายภาษาเช่นเดียวกับโปรโตคอล ซึ่งก็จะมีโปรโตคอลอยู่หลายแบบเช่นกัน TCP/IP นั้นจะเปรียบได้ กับภาษาอังกฤษเนื่องจากเป็นโปรโตคอลที่เครื่องคอมพิวเตอร์ของทุกระบบเข้าใจและมีการใช้โปรโตคอล นี้ร่วมกันเพื่อการเชื่อมต่อเข้าสู่โลกอินเทอร์เน็ต

### เวิลด์ ไวด์ เว็บ คืออะไร

เมื่อไม่กี่ปีก่อนหน้านี้มีผู้ที่สนใจใช้งานอินเทอร์เน็ตไม่มากนัก เนื่องจากการใช้บริการอินเทอร์เน็ต ไม่ว่าจะเป็นการค้นหาข่าวสารข้อมูล การรับส่งจดหมายอิเล็กทรอนิกส์ การสำเนาไฟล์ ฯลฯ จะอยู่ในรูปแบบของตัวอักษร (Text Mode) เท่านั้น ไม่มีการแสดงที่เป็นรูปภาพ เสียง และไม่มีตัวอักษรแบบต่าง ๆ ปรากฏให้เห็นแต่อย่างใด นอกจากนี้ผู้ใช้ต้องเรียนรู้และจดจำคำสั่งคอมพิวเตอร์มากมาย เช่น ต้องเรียนรู้คำสั่งเบื้องต้นของยูนิกซ์ (UNIX) เนื่องจากเมื่อจะมีการเรียกใช้อินเทอร์เน็ต เครื่องคอมพิวเตอร์ของผู้ใช้จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนให้เป็นเทอร์มินัลของโฮสต์คอมพิวเตอร์ที่ให้บริการอินเทอร์เน็ต และโฮสต์ส่วนมากจะทำงานอยู่ภายใต้ระบบปฏิบัติการยูนิกซ์ ดังนั้นผู้ใช้จึงจะต้องเรียนรู้คำสั่งเบื้องต้นของยูนิกซ์ (เช่นเกี่ยวกับการเรียนรู้คำสั่งคอส (DOS) บนเครื่องพีซี) เพื่อทำการป้อนคำสั่งที่เป็นตัวอักษรด้วยตนเองให้โฮสต์คอมพิวเตอร์ทำงานตามที่เรากำลังการใช้

จนกระทั่งมีบริการที่เรียกว่า เวิลด์ ไวด์ เว็บ หรือเครือข่ายใยแมงมุมเกิดขึ้น ทำให้ความนิยมการใช้อินเทอร์เน็ตสูงขึ้นเป็นทวีคูณ เนื่องจาก เวิลด์ ไวด์ เว็บ เป็นบริการอันหนึ่งที่มีอยู่ในอินเทอร์เน็ต ทำให้การใช้งานอินเทอร์เน็ตเป็นสิ่งที่ง่ายขึ้น ผู้ใช้ไม่ต้องจดจำคำสั่งของยูนิกซ์อีกต่อไป การอ่านและค้นหาข่าวสารข้อมูล ทำได้โดยการกดปุ่มบนเมาส์เพียงอย่างเดียว นอกจากนี้ข่าวสาร ข้อมูลหรือเอกสารที่เก็บอยู่ในอินเทอร์เน็ตจะอยู่ในรูปแบบของสื่อผสม (Multimedia) ที่เรียกว่า “เว็บเพจ” (web page) อันประกอบไปด้วยอักษรข้อความที่มีรูปแบบต่าง ๆ รูปภาพ เสียง ภาพเคลื่อนไหว วิดีโอและไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นการแสดงข้อความที่การเชื่อมโยงถึงกัน ได้เหมือนกับการเรียกใช้ Help ในโปรแกรม วินโดวส์ โดยผู้ใช้สามารถเรียกดูเอกสารหนึ่งจากเอกสารหนึ่งได้ ซึ่งเป็นลักษณะการเชื่อมต่อที่คล้ายกับ ใยแมงมุม จึงมีการตั้งชื่อบริการนี้ว่าเป็นเครือข่ายใยแมงมุม

### ส่วนประกอบของ เวิลด์ ไวด์ เว็บ

การจะใช้บริการ เวิลด์ ไวด์ เว็บ ได้นั้นจำเป็นต้องมีส่วนประกอบดังนี้

- แหล่งข้อมูลหรือเว็บไซต์
- โปรแกรมเว็บเบราว์เซอร์

#### 1. แหล่งข้อมูลหรือเว็บไซต์

เว็บไซต์ หรือ เว็บเซอร์ฟเวอร์ คือ ระบบคอมพิวเตอร์ที่เป็นแหล่งเก็บเว็บเพจที่ผู้ใช้บริการสามารถเรียกดูเว็บเพจที่เก็บอยู่ในเว็บไซต์นั้นได้ ซึ่งเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเว็บไซต์อาจจะใช้ระบบปฏิบัติการยูนิกซ์หรือวินโดวส์ เอนที ก็ได้ และจะต้องมีการติดตั้งโปรแกรมจัดการ ที่จะทำให้เครื่องคอมพิวเตอร์เครื่องนั้นทำหน้าที่เป็นเว็บไซต์ ตัวอย่างเช่น ในกรณีที่เครื่องคอมพิวเตอร์นั้นใช้ระบบปฏิบัติการ วินโดวส์ เอนที จะมีซอฟต์แวร์เว็บเซอร์ฟเวอร์ให้เลือกใช้ เช่น

โปรแกรม Purveyor HTTP Server ของสถาบัน European Microsoft Windows NT Academic Centre (EMWAC) โปรแกรม เว็บไซต์ ของบริษัท O'Reilly and Associate หรือถ้าใช้ระบบปฏิบัติการยูนิกซ์ ก็อาจใช้เว็บเซอร์ฟเวอร์เป็น NCSA httpd, Apache เป็นต้น

ผู้ที่เป็นเจ้าของเว็บไซต์จะจัดสร้างเว็บเพจของคนเก็บไว้ที่เว็บไซต์นั้นเพื่อให้ผู้ใช้คนอื่น ๆ ทั่วโลกสามารถเข้ามาดูเก็บไว้ในเว็บไซต์นั้นได้ ตัวอย่างเช่น เว็บเพจของสำนักข่าว CNN จะเก็บอยู่ที่เว็บไซต์ [www.cnn.com](http://www.cnn.com) เป็นต้น

#### 2. โปรแกรมเว็บเบราว์เซอร์

เป็นโปรแกรมหรือซอฟต์แวร์ที่ใช้ในการเข้าสู่ เวิลด์ ไวด์ เว็บ จะเปิดดูเว็บเพจที่เก็บอยู่ในเว็บไซต์ ใดๆ ตัวอย่างของโปรแกรมเว็บเบราว์เซอร์ ที่มีให้ดาวน์โหลดใช้กันได้ฟรีได้แก่ โปรแกรม

Netscape Navigator จากบริษัท Netscape Communications และ โปรแกรม Microsoft Internet Explorer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ชื่อโฮสต์คอมพิวเตอร์.ชื่อเครือข่ายท้องถิ่น[.ชื่อโดเมนย่อย].[ชื่อโดเมนระดับบนสุด

ชื่อ โดเมนจะมีการแบ่งออกเป็นระดับชั้น โดยใช้เครื่องหมายจุดคั่นชื่อ โดเมนที่อยู่ทางด้านขวาสุดจะเรียกว่าชื่อ โดเมนระดับบนสุด ซึ่งจะแบ่งออกได้อีก 2 ประเภทคือ

- ชื่อ โดเมนระดับบนสุดที่เป็นชื่อย่อประเภทขององค์กร ในสหรัฐอเมริกา เช่น

com	commercial	กลุ่มองค์กรเอกชน
edu	educational	กลุ่มสถาบันการศึกษา
gov	governmental	กลุ่มองค์กรของรัฐทั่วไป
mil	military	กลุ่มองค์กรทหาร
net	network services	กลุ่มองค์กรบริการเครือข่าย
org	non-commercial organization	กลุ่มองค์กรที่ไม่แสวงหาผลกำไร

ตัวอย่างเช่น ชื่อ โดเมน biz.zd.com โดเมนระดับบนสุดคือ com ที่แสดงถึงองค์กรเอกชนที่อยู่ในประเทศสหรัฐอเมริกา สำหรับระดับถัดไปคือ zd จะแสดงถึงชื่อเครือข่ายของบริษัทเอกชนนั้น และ biz จะหมายถึงชื่อ โฮสต์คอมพิวเตอร์ที่ให้บริการ เป็นต้น

- ชื่อ โดเมนระดับบนสุดที่เป็นชื่อย่อของประเทศต่าง ๆ เช่น

au	ออสเตรเลีย	, .jp	ญี่ปุ่น
ca	แคนาดา	, .th	ไทย
fr	ฝรั่งเศส	, .uk	อังกฤษ

จะมีสับโดเมน (Subdomain) ที่แสดงถึงประเภทขององค์กร ในประเทศนั้น ๆ เช่น

ac	สถาบันการศึกษา	, .go	องค์กรรัฐบาล
co	องค์กรเอกชน	, .or	องค์กรไม่แสวงหาผลกำไร

ตัวอย่างเช่น ชื่อโดเมน kku1.kku.ac.th มีชื่อโดเมนระดับบนสุดคือ th ที่หมายถึงประเทศไทย ระดับรองลงมาคือโดเมนย่อยคือ ac หมายถึงสถาบันการศึกษา ระดับถัดไปคือ kku หมายถึงมหาวิทยาลัยขอนแก่นซึ่งโดยทั่วไปจะหมายถึง เครือข่ายมหาวิทยาลัยของแก่นสำหรับ kku1 จะเป็นชื่อโฮสต์คอมพิวเตอร์ที่ให้บริการ เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## URL

นอกจากที่อยู่อีเมลล์แล้ว ชื่อโดเมนยังนำมาใช้เป็นส่วนหนึ่งของข้อมูล URL (Uniform Resource Locator) ซึ่งโปรแกรมเว็บเบราว์เซอร์จะใช้ข้อมูล URL นี้ในการค้นหาที่อยู่บนเว็บไซต์ที่ผู้ใช้บริการจะไปเรียกดูข้อมูล รูปแบบของ URL มีดังนี้

**โปรโตคอล://ชื่อโดเมน/ไดรกทอรีที่เก็บไฟล์ในโฮสต์/ชื่อไฟล์ในโฮสต์**

โปรโตคอล หมายถึง โปรโตคอลที่ใช้เรียกบริการ ในอินเทอร์เน็ต เช่น http:// หมายถึง โปรโตคอลที่เรียกใช้บริการ เวิลด์ ไวด์ เว็บ ที่มีข้อมูลเป็น Hypertext  
ftp:// หมายถึง โปรโตคอลที่เรียกใช้บริการ FTP เป็นต้น

ตัวอย่างเช่น ถ้าผู้ใช้ต้องการเรียกใช้บริการ เวิลด์ ไวด์ เว็บ เพื่อเข้าไปยังโฮมเพจของบริษัทการบินไทยจะทำได้โดยการระบุ URL เป็น <http://www.thaiair.com/> ในช่องแอดเดรส ของเว็บเบราว์เซอร์ที่ใช้โดยระบุเฉพาะชื่อโดเมนอย่างเดียว

แต่ถ้าต้องการจะเรียกดูข้อมูล ตารางเที่ยวบิน ตารางเที่ยวบินของบริษัท จะสามารถระบุ URL เป็น <http://www.thaiair.com/flying/schedule.htm> หรือทำการคลิกเลือกหัวข้อตารางเที่ยวบินที่อยู่ภายในโฮมเพจนั้นได้เลย

## HTML

โฮมเพจและเว็บเพจใด ๆ โดยส่วนใหญ่จะถูกสร้างขึ้นมาจากภาษาที่เรียกว่า HTML (Hyper Text markup Language) หรือ “ภาษาสำหรับทำเครื่องหมาย” ประกอบด้วยคำสั่ง (Tags) ที่ใช้ในการกำหนดว่าเว็บเพจจะมีข้อความอะไร มีการแสดงรูปภาพ เสียง และภาพวิดีโอที่ตำแหน่งใด นอกจากนี้ ยังมีคำสั่งสำหรับการเชื่อมโยงเว็บเพจหนึ่งไปยังอีกเว็บเพจหนึ่งและไปยังบริการอื่น ๆ ในอินเทอร์เน็ต อีกด้วย

การสร้างเว็บเพจนอกจากจะสร้างโดยการเขียน HTML บน โปรแกรม Editor ตัวใดตัวหนึ่ง เช่น โปรแกรม Notepad บน วินโดวส์ แล้วยังสามารถใช้โปรแกรมช่วยสร้างเว็บเพจกำหนดรูปร่างหน้าตาของเว็บเพจได้โดยตรง โดยที่ผู้สร้างไม่จำเป็นต้องเขียนหรือเรียนรู้ HTML แต่อย่างใด ซึ่งในปัจจุบันจะมีโปรแกรมช่วยสร้างเว็บเพจหลายตัว เช่น โปรแกรม Microsoft FrontPage โปรแกรม Netscape Navigator Gold เป็นต้น

## HTML Form และ CGI สคริปต์

ถึงแม้ว่าเว็บเพจที่สร้างจะได้รับการออกแบบให้เป็นที่น่าดึงดูดความสนใจ เต็มไปด้วยสีสันรูปภาพการสร้าง Link เชื่อมโยงข้อมูล ฯลฯ แต่สิ่งหนึ่งที่จะทำให้เว็บเพจยิ่งน่าสนใจมากขึ้นอีกคือ การเปิดโอกาสให้มีการโต้ตอบจากผู้อ่านทั่วไป ซึ่งทำได้โดยการสร้างแบบฟอร์ม (Form) ที่มีลักษณะเหมือนกับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบสอบถามทั่ว ๆ ไปลงเว็บเพจ เพื่อให้ผู้อ่านกรอกข้อมูลลงในแบบฟอร์มแล้วส่งกลับมายังเครื่องบริการที่เป็นเครื่องเซิร์ฟเวอร์

ในการนี้จำเป็นต้องมีการเขียนโปรแกรมย่อยที่เรียกว่า สคริปต์ เก็บไว้ในเครื่องเซิร์ฟเวอร์เพื่อทำหน้าที่รวบรวมข้อมูลที่ส่งมาจากแบบฟอร์มและนำไปประมวลผลเก็บไว้ หรืออาจส่งผลลัพธ์กลับไปยังผู้อ่านอีกทีหนึ่ง

สคริปต์ ที่ถูกเขียนขึ้นนี้อาจสร้างภาษาใดภาษาหนึ่งเช่น ภาษา C, Perl, Java, Pascal หรือแม้แต่ Unix Shell

### Static และ Dynamic HTML

เอกสาร HTML โดยทั่วไปจะมีอยู่ 2 ลักษณะคือ

- Static HTML
- Dynamic HTML

#### Static HTML

จะเป็นเอกสาร HTML ที่ถูกเขียนขึ้นและเก็บอยู่ในเครื่องเซิร์ฟเวอร์ เมื่อมีการเรียกใช้งานจากผู้ใช้งานทาง URL เครื่องเซิร์ฟเวอร์ก็จะส่งเอกสารนี้ไปให้กับผู้ใช้ ซึ่งเอกสารชนิดนี้จะถูกเขียนขึ้นโดยใช้ HTML tag ธรรมดา ไม่จำเป็นต้องมีการสร้างสคริปต์ ใดๆ

#### Dynamic HTML

สามารถเรียกได้อีกอย่างว่า “HTML on-the-fly” เอกสารประเภทนี้จำเป็นต้องมีการเขียน สคริปต์ เก็บไว้ในเครื่องเซิร์ฟเวอร์ เซิร์ฟเวอร์จะทำการประมวลผล สคริปต์ นั้นและ สคริปต์ จะทำการสร้างผลลัพธ์ที่เป็นเอกสาร HTML ที่มีรูปแบบของผลลัพธ์ขึ้นอยู่กับข้อมูลที่ผู้ใช้งานป้อนจากแบบฟอร์ม จึงเรียกเอกสารประเภทนี้ว่า “Dynamic HTML” เพื่อให้เซิร์ฟเวอร์ส่งผลลัพธ์นั้นกลับ ไปยังผู้ใช้ทันที

## 2.2 เอเอสพี (Active Server Page : ASP)

เอเอสพี เป็นคำที่ย่อมาจาก Active Server Pages ซึ่งเป็นโปรแกรมที่พัฒนาขึ้นโดยบริษัท ไมโครซอฟท์ เพื่อใช้งานทางด้านอินเทอร์เน็ต โดย เอเอสพี จะทำหน้าที่ตีความเอกสารที่เขียนด้วยภาษาสคริปต์ เช่น VBสคริปต์ โดยที่มี เอเอสพี tag (คือ คำสั่งที่มีเครื่องหมาย <% %>) กำกับอยู่ ซึ่งเบราว์เซอร์ทั่วไป เช่น Netscape Navigator หรือ Internet Explorer ไม่สามารถนำไปแสดงผล จากนั้นจึงสร้างเอกสารผลลัพธ์เป็นเอกสาร HTML อันเป็นเอกสารที่ประกอบด้วย HTML tag ต่างๆ(คือ คำสั่งที่มีเครื่องหมาย <>) กำกับอยู่ ซึ่งเบราว์เซอร์ทั่วไปสามารถนำไปสร้างเป็นเว็บเพจขึ้นเพื่อใช้แสดงผลได้

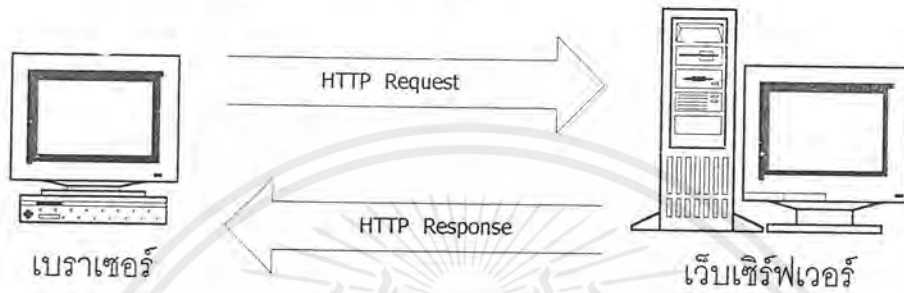
การทำงานของโปรแกรม เอเอสพี จะเกิดขึ้นเฉพาะทางฝั่งเซิร์ฟเวอร์เท่านั้น เราจึงเรียกว่า เป็นการทำงานแบบ เซิร์ฟเวอร์ไซด์ จากนั้นผลลัพธ์ที่ได้จะถูกส่งไปที่เว็บเซิร์ฟเวอร์ แล้วเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารดังกล่าวต่อไปยังเบราว์เซอร์อีกทีหนึ่ง เมื่อเบราว์เซอร์ได้รับเอกสารนั้นแล้ว เบราว์เซอร์ก็จะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แสดงผลได้ถูกต้องครบถ้วน โดยการทำงานของบราวเซอร์ทางฝั่งของผู้ใช้ เราเรียกว่าเป็นการทำงานแบบ ไคลเอนไซด์ (client side)

เข้าใจขบวนการทำงานของ เอเอสพี

เพื่อจะได้ศึกษา เอเอสพี ในระดับเบื้องต้นก่อนอื่นขอให้ลองทำความเข้าใจพื้นฐานขบวนการทำงานของ เอเอสพี (เอเอสพี model) จากรูปที่ 2.2 ดังนี้



รูปที่ 2.2 ขบวนการทำงานของ เอเอสพี

การทำงานทั้งหมดจะเริ่มจาก บราวเซอร์ร้องขอเอกสาร HTML ไปยังเว็บเซิร์ฟเวอร์ผ่านทาง HTTP (HTTP request) โดยที่เอกสารที่ขอไปจะเป็นแฟ้มข้อมูลที่มีนามสกุลเป็น .ASP (เช่น search.ASP ฯลฯ) เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอดังกล่าว ก็จะส่งเอกสารนั้น ไปให้ เอเอสพี ตีความจากนั้น เอเอสพี ก็จะสร้างเอกสาร HTML ส่งกลับไปให้เว็บเซิร์ฟเวอร์เพื่อส่งต่อไปยังบราวเซอร์และใช้แสดงผลทางฝั่งผู้ใช้ต่อไป (HTTP response) ซึ่งการทำงานของ เอเอสพี นี้ แทบไม่แตกต่างไปจากหลักการทำงานโปรแกรม CGI (Common Gateway Interface) จนอาจกล่าวได้ว่า เอเอสพี ก็เป็น โปรแกรม CGI ประเภทหนึ่งเช่นกัน

การเขียนโปรแกรมเพื่อที่จะสร้างเอกสารที่จะทำงานกับ เอเอสพี นั้น (ในที่นี้ขอเรียกว่า เอกสาร เอเอสพี) ไม่จำเป็นต้องอาศัยโปรแกรมเฉพาะในการเขียน เราสามารถนำ โปรแกรมประเภท text editor มาใช้งานได้ทันที เช่น โปรแกรม Notepad ฯลฯ หรือจะใช้โปรแกรมที่เขียนเอกสาร เอเอสพี โดยเฉพาะก็ได้ เช่น Visual InterDev เป็นต้น

เอกสาร เอเอสพี แตกต่างกับเอกสาร HTML ทั่วไปตรงที่มีส่วนของคำสั่ง เอเอสพี อยู่ในเอกสารด้วย โดยทั่วไปหากเรานำเอกสาร HTML มาเปลี่ยนเป็นเอกสาร เอเอสพี เลยก็ทำได้ นั่นคือวิธีการสร้างเอกสาร เอเอสพี แบบง่ายๆ เช่น เราสามารถเปลี่ยนเอกสาร HTML ที่ชื่อ index.html ไปเป็น index.asp ได้เลย โดยที่เมื่อโปรแกรม เอเอสพี ตีความส่วนใดของเอกสารที่มี HTML tag กำกับอยู่ ก็จะไม่เกิดการเปลี่ยนแปลงใด ๆ ในเอกสารนั้นเลย แต่หากว่าส่วนใดมี เอเอสพี tag กำกับ เอเอสพี ก็จะเปลี่ยนเอกสารส่วนดังกล่าว ไปอยู่ในรูปข้อความทั่วไปหรือเป็น HTML tag แทน เช่น หากในเอกสารมีคำสั่งนี้

```
<br> <%response.write("Hello " & now)%>
```

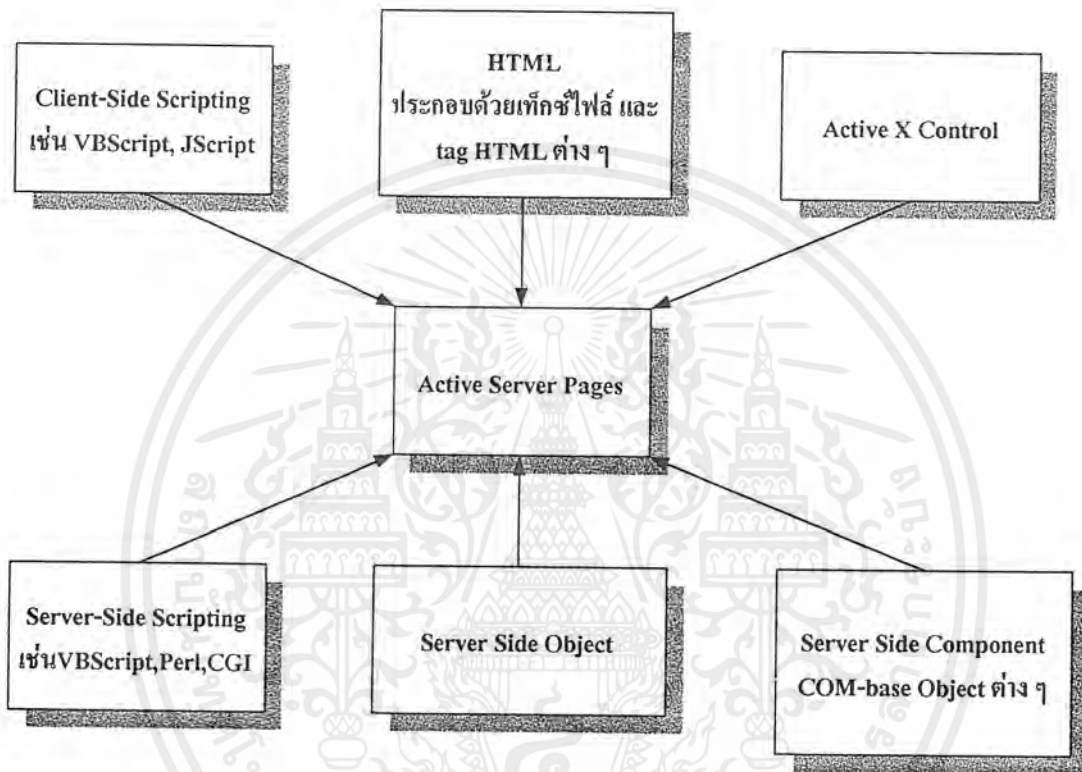
ก็จะถูกเปลี่ยนเป็น...

```
<br>Hello แล้วตามด้วยวันและเวลาปัจจุบัน
```

เอกสารนี้เป็นเอกสารที่ส่งมาเพื่อขอการเซ้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### เอเอสพี (Active Server Pages)

เอเอสพี เป็นชื่อของเทคโนโลยีในการพัฒนาแอปพลิเคชันสำหรับอินเทอร์เน็ตที่ไม่โครซอฟท์คิดขึ้นมา โดยเน้นไปที่การพัฒนา และจัดการแอปพลิเคชันที่เว็บเซิร์ฟเวอร์ โดยเราจะเรียกแอปพลิเคชันที่สร้างจากเทคโนโลยีเอเอสพีว่า “แอปพลิเคชัน เอเอสพี”



รูปที่ 2.3 โครงสร้างทางด้านเอเอสพี

แอปพลิเคชัน เอเอสพี (ASP Application) ก็คือ เท็กซ์ไฟล์ที่บรรจุเอาคำสั่งสคริปต์ ต่าง ๆ ผสมรวมกับเอกสาร HTML ซึ่งจะถูกเก็บไว้ที่เว็บเซิร์ฟเวอร์ เมื่อมีบราวเซอร์เรียกใช้งานก็จะถูกแปล (โดย ASP Interpreter) และถูกเอ็กซ์คิวต์ที่เว็บเซิร์ฟเวอร์ โดยผลลัพธ์ที่ได้จากการเอ็กซ์คิวต์แอปพลิเคชันเอเอสพี จะเก็บในรูปเอกสาร HTML แล้วถูกส่ง กลับไปให้บราวเซอร์ที่เรียกใช้แอปพลิเคชัน เอเอสพี นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 2.1 ตัวอย่างไฟล์เอเอสพี

```

ตัวอย่างไฟล์ เอเอสพี
<HTML>
<HEAD> <TITLE> ยินดีต้อนรับเข้าสู่โลกของ เอเอสพี </TITLE> </HEAD>
<BODY>
<Script RUNAT=SERVER LANGUAGE = VBScript>
<%
Sub GreetingASP()
    Response.Write(“ยินดีต้อนรับสู่โลกของ เอเอสพี”)
End Sub
%>
<% Call GreetingASP %>
<%= “<BR> ขณะนี้เวลา” & Time & “เวลานี้เป็นเวลาที่ เซิร์ฟเวอร์ นะครับ” %>
</SCRIPT>
</BODY>
</HTML>

```

เพราะฉะนั้นจึงไม่มีการรันแอปพลิเคชัน เอเอสพี ที่บราวเซอร์ โดยต้องรันที่เว็บเซิร์ฟเวอร์เท่านั้น สังเกตได้จากตัวอย่างที่ Tag <Script> จะกำหนดให้แอตทริบิวต์ RUNAT = SERVER

### การเขียนสคริปต์ในรูปแบบของ เอเอสพี

สำหรับการเขียนสคริปต์ในรูปแบบ เอเอสพี เราสามารถใช้ได้ตั้งแต่ โน้ตแพ็ค, เวิร์ดโปรเซสเซอร์ หรือแม้กระทั่งเครื่องมือพัฒนาแอปพลิเคชันสำหรับอินเทอร์เน็ตโดยเฉพาะ เช่น Visual Interdev

สำหรับสคริปต์ที่ใช้เขียน เอเอสพี ได้ นั้น ณ ขณะนี้สามารถใช้ได้ 2 ภาษา ได้แก่ VBScript กับ Jscript (เป็น จาวาสคริปท์ ในเวอร์ชันของไมโครซอฟท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 2.2 ตัวอย่างไฟล์เอเอสพี ร่วมกันทั้ง JScript และ VBScript

```

ตัวอย่างไฟล์ ASP ร่วมกันทั้ง JScript และ VBScript
<%@ LANGUAGE="VBScript" %>
<HTML>
<HEAD> <TITLE> ทดลอง JScript กับ เอเอสพี </TITLE> </HEAD>
<BODY>
< Script RUNAT=Server LANGUAGE= "JScript">
function J Script Sample()
{
    var DateObject = new Date ( )
    Response.Write("<H1> ทดลอง JScript </H1>")
    Response.Write("<BR> ขณะนี้เวลา" +DateObject.getHours( ) + "." +
DateObject.getMinute( ))
}
</Script>
<% Call J Script Sample %>
</BODY>
</HTML>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

คุณสมบัติของ MCS-51 คุณสมบัติที่สำคัญๆ ของชิปไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว
- มีหน่วยความจำสำหรับเก็บ โปรแกรมควบคุมการทำงานอยู่ภายในชิป จำนวน 4 กิโลไบต์ (เบอร์ 8031, 8032 ไม่มีหน่วยความจำส่วนนี้ ส่วนเบอร์ 8052 มีหน่วยความจำส่วนนี้ 8 กิโลไบต์ และสำหรับเบอร์ 83C51FB จะมีหน่วยความจำส่วนนี้รวมทั้งสิ้น 16 กิโลไบต์)
- มีหน่วยความจำสำหรับ โปรแกรมและข้อมูล (RAM) อยู่ภายในชิปจำนวน 128 ไบต์ (ในเบอร์ 8031, 8051) หรือ 256 ไบต์ (ในเบอร์ 8031, 8052)
- สามารถให้หน่วยความจำสำหรับ โปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้ อย่างละ 64 กิโลไบต์ แยกจากกัน
- คำสั่งส่วนใหญ่ใช้เวลาทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกกะเฮิร์ตซ์
- มีพอร์คที่สามารถรับหรือส่งข้อมูลได้ทั้ง 2 ทิศทาง จำนวน 4 พอร์ค ๆ ละ 8 บิต หรือสามารถใช้งานเป็นพอร์คขนาด 1 บิตแยกจากกัน ทำให้เสมือนมีพอร์คขนาด 1 บิต ใช้งานรวมทั้งสิ้น 32 พอร์ค
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดอัตราเร็วในการรับและส่งข้อมูล (Baud Rate) ได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
- จัดลำดับความสำคัญของสัญญาณอินเตอร์รัปต์ได้ 2 ระดับ
- มีรีจิสเตอร์สำหรับใช้งานเป็น ไทม์เมอร์หรือเคาน์เตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิปหรือนับเปลี่ยนแปลงสถานะของสัญญาณภายนอกขนาด 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์วัดความกว้างของพัลส์หรือใช้วัดช่วงเวลา (ในเบอร์ 8052 จะมี 3 ตัว)
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วน สามารถเข้าถึงข้อมูลได้ทั้งระดับบิต เพื่อให้การออกแบบ โปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น
- มีคำสั่งคูณและหารเลขขนาด 8 บิตในตัวเอง
- สามารถประมวลผลแบบบวกลบเพื่อใช้งานควบคุมโดยเฉพาะ
- ใช้โปรแกรมของไมโครคอนโทรลเลอร์ตระกูลMCS-48 (upwardly compatible) ได้

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ที่จัดได้ว่าเป็นเบอร์พื้นฐานในตระกูลนี้ คือ เบอร์ 8051, 8751 และ 8031 ซึ่งมีจำนวนขาภายนอก 40 ขาเหมือนกัน ใช้เวลาในการปฏิบัติคำสั่งแต่ละคำสั่งเท่ากัน (มีไทม์มิ่งไดอะแกรมเหมือนกัน) ใช้แรงดันไฟเท่ากัน สิ่งที่แตกต่างกันระหว่างสาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบอร์นี้ก็คือขนาดของหน่วยความจำสำหรับเก็บลงในชิป(onchip program memory) ซึ่งมีไว้เพื่อตอบสนองความต้องการที่ไม่เหมือนกัน

### คุณสมบัติของไมโครคอนโทรลเลอร์ในตระกูล MCS-51

: เบอร์ 8751 มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเป็น EPROM ขนาด 4 กิโลไบต์ ซึ่งจะสามารถทำการลบข้อมูลเดิมแล้วเขียนซ้ำลงไปใหม่ได้ ทั้งนี้เพื่อความสะดวกในการแก้ไขหรือปรับปรุงโปรแกรม

: เบอร์ 8051 หลังจากทดสอบโปรแกรมจนไม่พบข้อผิดพลาดแล้ว จะเป็นช่วงของการผลิตจริง ซึ่งต้องคำนึงถึงต้นทุนเป็นอันดับแรก ในการผลิตจริงจะใช้ไมโครคอนโทรลเลอร์เบอร์ 8051 ซึ่งมีหน่วยความจำภายในเป็น ROM ขนาด 4 กิโลไบต์แทน เพราะราคาต่ำกว่ามาก แต่มีข้อจำกัดที่ไม่สามารถทำการแก้ไขโปรแกรมที่บรรจุไปแล้วได้ ไม่ว่าจะด้วยวิธีใดก็ตาม

:เบอร์ 8031 เบอร์นี้จะไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป แต่สามารถใช้หน่วยความจำเพื่อเก็บโปรแกรมภายนอกชิปได้มากถึง 64 กิโลไบต์ ซึ่งอาจจะใช้เป็น ROM, PROM หรือ EPROM ตามความต้องการของผู้ใช้

:เบอร์ 8751 และ8051 จะใช้โปรแกรมจากหน่วยความจำภายนอกได้เอง เมื่อมีโปรแกรมมีความยาวเกิน 4 กิโลไบต์ หรืออาจบังคับให้ไมโครคอนโทรลเลอร์ทั้งสองเบอร์ใช้โปรแกรมจากหน่วยความจำภายนอกเพียงอย่างเดียวด้วยการต่อขา 31 (EA) ลงกราวด์ ทำให้มีคุณสมบัติเหมือนกับเบอร์ 8031 ที่ไม่มีหน่วยความจำสำหรับโปรแกรมภายในชิป

### โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีสมาชิกในตระกูลหลายเบอร์ด้วยกัน แต่ละเบอร์จะมีคุณสมบัติพิเศษบางอย่างแตกต่างกัน เช่น มีหน่วยความจำภายในสำหรับเก็บโปรแกรมและข้อมูลภายในชิปเพิ่มขึ้น มีวงจรเปลี่ยนคำสั่งสัญญาณอนาล็อกเป็นดิจิทัลในตัว สามารถรับสัญญาณอินเทอร์รัปต์ได้หลายชนิด ทำกระบวนการ DMA (Direct Memory Access) ได้ในตัว มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์เพิ่มขึ้นคุณสมบัติพิเศษที่แตกต่างกันของไมโครคอนโทรลเลอร์แต่ละเบอร์ในตระกูลนี้ดังแสดงในตารางที่ผ่านมา

ไมโครคอนโทรลเลอร์เบอร์ที่นับได้ว่าเป็นเบอร์พื้นฐานสำหรับตระกูล MCS-51 นี้ ได้แก่ เบอร์ 8051, 8031, 8751 โดยเบอร์ 8051 จัดเป็นสมาชิกตัวแรกในตระกูล ซึ่งมีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป เป็น ROM ขนาด 4 กิโลไบต์ และหน่วยความจำสำหรับเก็บข้อมูลทั่วไปภายใน MCS-51 (RAM) เองจำนวน 128 ไบต์ มีพอร์ตขนาด 8 บิต 4 พอร์ต มีรีจิสเตอร์สำหรับใช้เป็นไทม์เมอร์หรือเคาน์เตอร์ขนาด 16 บิตรวม 2 ตัว รับสัญญาณอินเทอร์รัปต์จากภายนอกได้ 2 ชนิด สามารถรับและส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารข้อมูลแบบอนุกรม มีวงจรออสซิลเลเตอร์เพื่อสร้างสัญญาณนาฬิกาควบคุมการทำงานในตัวเอง ส่วนเบอร์ 8751 จะมีคุณสมบัติเหมือน เบอร์ 8051 ทุกอย่าง ต่างกันเพียงชนิดของหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป คือ เบอร์ 8751 จะเป็น EPROM แทนที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น ROM ส่วนเบอร์ 8031 จะเหมือนกับเบอร์ 8051 ต่างกันเพียงในเบอร์ 8031 ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเท่านั้น

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ใช้แรงดันไฟเพียง 5 โวลต์ในการทำงาน ส่วนกระแสไฟฟ้าที่ใช้จะแตกต่างกันไปตามชนิดของเทคโนโลยีที่ใช้ในการผลิตเบอร์ของไมโครคอนโทรลเลอร์ตระกูลนี้ที่มีตัวอักษร C อยู่ตรงกลางเบอร์ เช่น 80C31, 80C51 จะเป็นเบอร์ของชิปที่ผลิตโดยอาศัยเทคโนโลยี CMOS ซึ่งใช้พลังงานในการทำงานน้อยกว่าและสามารถควบคุมการใช้พลังงานของตัวชิปได้จากโปรแกรมเพื่อการประหยัดพลังงานในระบบ

MCS-51 เป็นตระกูลของไมโครคอนโทรลเลอร์ที่ถูกพัฒนาขึ้นมาจากตระกูล MCS-48 ดังนั้นจึงมีความสามารถเหนือกว่าหลายอย่าง ข้อดีของ MCS-51 คือสามารถใช้ความถี่ได้ถึง 12 เมกกะเฮิร์ตซ์ หรือสำหรับบางเบอร์ในตระกูลสามารถใช้ได้ถึง 16 เมกกะเฮิร์ตซ์ ทำให้ช่วงเวลาในการทำงานแต่ละคำสั่งน้อยมาก เมื่อใช้ความถี่ 12 เมกกะเฮิร์ตซ์ คำสั่งที่ใช้เวลาน้อยที่สุดจะใช้เวลาเพียง 1 ไมโครวินาที ส่วนคำสั่งที่ใช้เวลามากที่สุดจะใช้เวลาเพียง 4 ไมโครวินาทีเท่านั้น



รูปที่ 2.4 การจัดวางขาของ 8051

### ตำแหน่งขาของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์ จะมีตำแหน่งขาพื้นฐานที่เหมือนกันดังแสดงในรูปที่ 2.6

หน้าที่การใช้งานแต่ละขาของไมโครคอนโทรลเลอร์ในตระกูล MCS-51 มีดังนี้

- ขา VSS (ขา 20) สำหรับต่อลงกราวด์
- ขา VCC (ขา 40) สำหรับต่อแหล่งจ่ายแรงดันกระแสตรงขนาด 5 โวลต์
- ขาพอร์ต 0 (ขา 32-39) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 0 ขนาด 8 บิต (P0.0-P0.7)

แบบโอเพ่นเดรน ไบไดเรกชันแนล (Open Drain Bidirectional) พอร์ตนี้สามารถใช้งานเป็นอินพุต-เอาต์พุตพอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ตต้องโหนดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อบังคับให้ขา

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยไม่คิดค่าลิขสิทธิ์และสงวนลิขสิทธิ์ในเอกสารนี้ไว้เพื่อวัตถุประสงค์ในการศึกษาเท่านั้น ไม่สามารถนำเอกสารนี้ไปใช้เพื่อการค้าหรือเผยแพร่ในที่สาธารณะได้ หากมีการนำเอกสารนี้ไปใช้โดยไม่ได้รับอนุญาตจากผู้จัดทำเอกสารนี้ ผู้ใช้จะมีความผิดตามกฎหมายลิขสิทธิ์

แล้ว พอร์ต 0 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกชิปได้ด้วย โดยส่งค่าแอดเดรสไบต์ค่า (A0-A7) และมัลติเพล็กซ์กับการรับส่งข้อมูล (D0-D7) จากหน่วยความจำภายนอก ในระหว่างการเขียนหรืออ่านข้อมูลโดยมีวงจรถูกปิดภายใน

- ขาพอร์ต 1 (ขา 1-8) ใช้เป็นขาสำหรับพอร์ต (P1.0-P1.7) สามารถใช้งานเป็นอินพุตหรือเอาต์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มีสถานะถูกปล่อยลอย โดยมีวงจรถูกปิดภายใน

- ขาพอร์ต 2 (ขา 21-28) ใช้เป็นขาสำหรับพอร์ต 2 (P2.0-P2.7) ขนาด 8 บิต แบบโอเพ่น ดเรน ไบไดเร็กชั่นแนล พอร์ตสามารถใช้งานเป็นอินพุต-เอาต์พุตพอร์ตทั่วไปได้ โดยหากใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อบังคับให้ขาอยู่ในสถานะถูกปล่อยลอย นอกจากจะใช้งานเป็นอินพุต-เอาต์พุตพอร์ตทั่วไปแล้ว พอร์ต 2 ยังใช้ในการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลภายนอกด้วย โดยใช้สำหรับค่าแอดเดรสไบต์สูง (A8-A15) และมีวงจรถูกปิดภายใน

- ขาพอร์ต 3 (ขา 10-17) มี 8 ขา ใช้เป็นขาสำหรับพอร์ต 3 (P3.0-P3.7) สามารถใช้งานเป็นอินพุต-เอาต์พุตพอร์ตทั่วไปได้ หากต้องการใช้งานเป็นอินพุตพอร์ต ต้องโหลดค่า 1 ไปยังแต่ละบิตของพอร์ตนี้ เพื่อให้มีสถานะถูกปล่อยลอย โดยใช้วงจรถูกปิดภายใน นอกจากนี้ยังใช้งานในหน้าที่พิเศษต่าง ๆ อีกหลายอย่างดังนี้

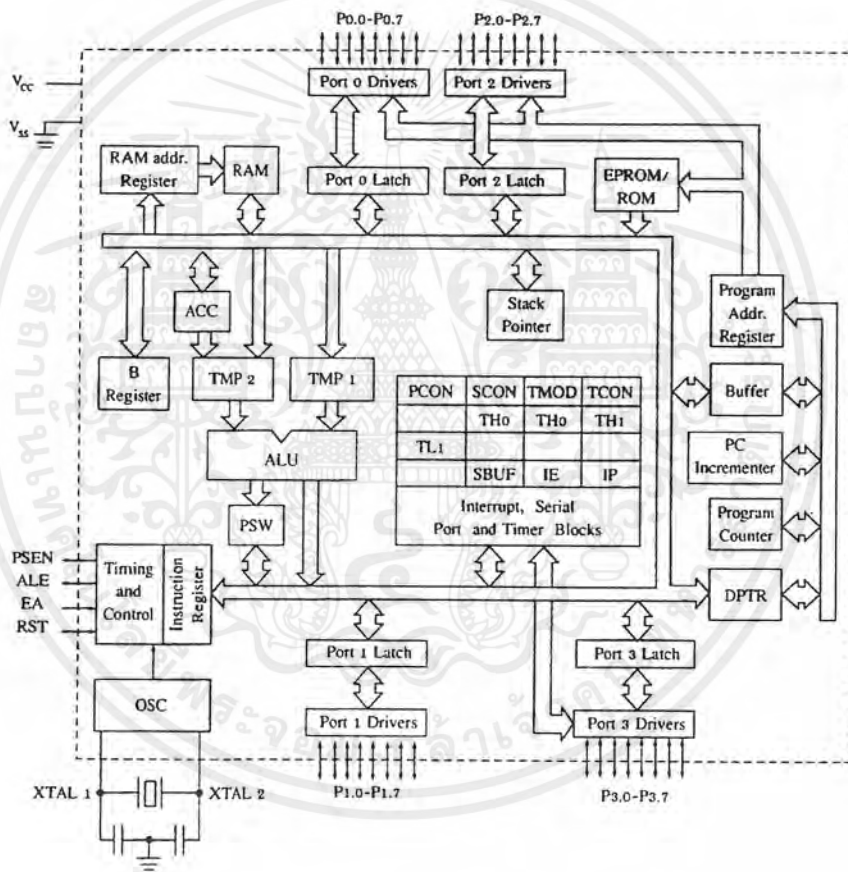
- : ขา P3.0 ใช้รับข้อมูลจากภายนอกแบบอนุกรม
- : ขา P3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม
- : ขา P3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเทอร์รัปต์ชนิดที่ 0
- : ขา P3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเทอร์รัปต์ชนิดที่ 1
- : ขา P3.4 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 0
- : ขา P3.5 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 1
- : ขา P3.6 ใช้เป็นสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับเก็บข้อมูลภายนอกชิป
- : ขา P3.7 ใช้เป็นสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำ สำหรับเก็บข้อมูลภายนอกชิป

การใช้งานพอร์ต 3 ในหน้าที่พิเศษดังกล่าวนี้จะต้องโหลดค่า 1 ไปยังแต่ละบิตที่ต้องการใช้ก่อนทุกครั้ง

- ขา RST (ขา 9) ใช้สำหรับการรีเซ็ตวงจรทุกอย่างภายในชิปเพื่อเริ่มต้นการทำงานใหม่ การรีเซ็ตใช้เมื่อเริ่มจ่ายพลังงานหรือเมื่อโปรแกรมเกิดทำงานผิดพลาด เมื่อต้องการรีเซ็ตชิป MCS-51 ขานี้ ต้องมีสถานะ 1 เป็นเวลาอย่างน้อย 2 แมกซ์ซีไนเซกิลระหว่างที่ออสซิลเลเตอร์ยังทำงานอยู่ โดยต้องต่อตัวต้านทานค่า 8.2 กิโลโอห์ม เพื่อทำหน้าที่พูลดาวน์ (รักษาค่าแรงดันไฟฟ้าให้มีสถานะเป็นกราวด์) และเพื่อให้ตัวรีเซ็ตเองเมื่อเริ่มจ่ายพลังงานให้ต่อตัวเก็บประจุขนาด 10 ไมโครฟาร์ัดคร่อมระหว่างขา RST กับ VCC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา ALE/PROG (ขา 30) เป็นขาสำหรับใช้ส่งสัญญาณออกไปภายนอก เพื่อควบคุมการแลตช์ค่าแอดเดรสไบต์ค่า (address latch enable) จากพอร์ต 0 ในระหว่างการติดต่อหน่วยความจำสำหรับเก็บโปรแกรมหรือข้อมูลภายนอกปกติเมื่อไม่มีการติดต่อหน่วยความจำภายนอก ขานี้จะส่งสัญญาณพัลส์ออกมาด้วยความถี่ 1/8 ของความถี่ออสซิลเลเตอร์ที่ใช้ตลอดเวลา ดังนั้นเราสามารถใช้เวลาที่ได้ออกไปใช้งานอย่างอื่นได้ แต่ความถี่ที่ขานี้จะลดลงครึ่งหนึ่งในระหว่างการติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลที่อยู่นอกชิป นอกจากนี้ ALE ยังใช้สำหรับควบคุมการเขียนโปรแกรมลงไปใน EPROM สำหรับ MCS-51 เบอร์ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปเป็น EPROM



รูปที่ 2.5 แสดง โครงสร้างภายในของ MCS-51

- ขา EA/Vpp (ขา 31) เป็นขาสำหรับใช้เลือกให้ MCS-51 ทำงานจากโปรแกรมที่อยู่ในภายในหรือภายนอกชิป โดยหากขานี้มีสถานะเป็น 0 หมายถึงให้ใช้โปรแกรมจากหน่วยความจำที่เก็บโปรแกรมภายนอกหากขานี้มีสถานะเป็น 1 หมายถึงบังคับให้ MCS-51 ใช้โปรแกรมจากหน่วยความจำสำหรับเก็บโปรแกรมภายในชิป และสำหรับ MCS-51 ที่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปสามารถเลือกให้ทำงานได้ทั้งจากโปรแกรมที่เก็บในหน่วยความจำภายในชิปหรือจากโปรแกรมที่เก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ผ่านการอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน่วยความจำนอกชิปด้วยกับไฟเลี้ยงหรือกราวด์ตามลำดับ ส่วนใน MCS-51 ที่ไม่มีหน่วยความจำสำหรับเก็บโปรแกรมภายในชิปให้ค่อขาลงกราวด์เสมอ

- ขา XTAL 1 (ขา 19) ใช้ต่อคริสตอลภายนอก โดยเป็นอินพุตเข้าสู่วงจร ออสซิลเลเตอร์
- ขา XTAL 2 (ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาต์พุตออกจากวงจร ออสซิลเลเตอร์

ไมโครคอนโทรลเลอร์ที่เลือกใช้ในครั้งนี้จะเป็นไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ซึ่งมีอยู่ด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายในของมัน เบอร์พื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วย 8051, 8031, 8751 ซึ่งแตกต่างกันที่ชนิดและหน่วยความจำภายใน โดยในโครงการนี้จะใช้เบอร์ 8031 ซึ่งมีหน่วยความจำสำหรับเก็บข้อมูลภายในจำนวน 128 ไบต์ มีพอร์ตขนาด 8 บิต 4 พอร์ต มีรีจิสเตอร์สำหรับใช้เป็นไทมเมอร์หรือเคาน์เตอร์ขนาด 16 บิตรวม 2 ตัวรับสัญญาณอินพุตรีปต์จากภายนอกได้ 2 ชนิด สามารถรับส่งข้อมูลแบบอนุกรมผ่านทางพอร์ตสื่อสารแบบอนุกรม มีวงจรออสซิลเลเตอร์เพื่อสร้างสัญญาณนาฬิกาควบคุมการทำงานในตัวเอง

### โครงสร้างของหน่วยความจำใน MCS-51

แบ่งออกเป็น 2 ส่วนคือ

1. หน่วยความจำโปรแกรม
2. หน่วยความจำข้อมูล

MCS-51 ทุกเบอร์จะมีหน่วยความจำสำหรับเก็บข้อมูลภายในอย่างน้อย 128 ไบต์ไปจนถึง 256 ไบต์ หน่วยความจำสำหรับเก็บข้อมูลภายในบริเวณ 128 ไบต์แรกมีชื่อเรียกว่า Lower 128 ไบต์และในบริเวณ 128 ไบต์หลังที่มีเพิ่มในบางเบอร์มีชื่อเรียกว่า Upper 128 ดังแสดงในรูปที่ 2.8

หน่วยความจำสำหรับเก็บข้อมูลภายในบริเวณ 128 ไบต์แรกและหลังจะมีวิธีการ เข้าถึงข้อมูลในหน่วยความจำทั้งสองส่วนไม่เหมือนกัน

FFH	หน่วยความจำสำหรับเก็บข้อมูล ภายในส่วนนี้มีใน MCS-51 ของเบจอร์เท่านั้น
80H 7FH	
2FH	บริเวณหน่วยความจำที่ใช้ได้ถึงระดับบิต จำนวน 16 ไบท์ X 8 = 128
20H	รีจิสเตอร์ใช้งานเท่าไป R0-R7 กลุ่ม 4
18H	รีจิสเตอร์ใช้งานเท่าไป R0-R7 กลุ่ม 3
10H	รีจิสเตอร์ใช้งานเท่าไป R0-R7 กลุ่ม 2
08H	รีจิสเตอร์ใช้งานเท่าไป R0-R7 กลุ่ม 1
00H	

รูปที่ 2.6 การจัดหน่วยความจำ

### การทำงานของ 8051

คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่าฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวไม่ได้ จะต้องมีการโปรแกรมหรือคำสั่งที่จัดเรียงกันไว้ให้คอมพิวเตอร์ทำงานตามลำดับใน 8051 ก็เช่นกัน ผู้ใช้จะต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลขฐาน 2 เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งของ 8051 อาจประกอบด้วย 1, 2 หรือ 3 ไบท์แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไร คอมพิวเตอร์ก็จะเหมือนกับคนที่จะต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วก็กลับมารับคำสั่งต่อไป

เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Power on set ค่อยอยู่จะมีการรีเซ็ตเกิดขึ้น การทำงานภายใน 8051 จะเริ่มจากบัสล็อก Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่ง ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปยังบัส (Bus) หมายเลข 1 บัสนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ จะปรากฏที่บัส 16 บิตหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซ็ต ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายในหรือภายนอก 8051 โดยการป้อนสถานะลอจิกเข้าไปที่ 8051 ทางขา  $\overline{EA}$  ซึ่งค่อยอยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไปถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา  $\overline{EA}$  จะเป็นการเลือกใช้ ROM ภายใน 8051 โดยที่วงจร Timing and Control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาจากบัสหมายเลข 2 ข้อมูลจาก ROM จะถูกส่งลงไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไว้ที่ Instruction Register (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆ ต่อไปแล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไร ในกรณีที่เลือก ROM ภายนอก 8051 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ขา  $\overline{EA}$  จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และพอร์ท 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่ เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วไปเก็บที่ Instruction Register เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจาก ROM ภายใน การทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่ เป็นคำสั่งกลับเข้ามาเก็บไว้ใน Instruction Register เรียกว่าเป็นช่วงของการ Fetch (Fetch Cycle) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์ Accumulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM ตำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งค่าตำแหน่งหน่วยความจำ 23H ลงไปยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจร Latch) ซึ่งจะทำหน้าที่เก็บสถานะ ผลลัพธ์ของการทำงานใน ALU เช่นผลลัพธ์การบวกมีค่าเกิน 8 บิต ก็จะทำให้บิตหนึ่งใน PSW ถูก SET เป็น 1

#### 2.4 การสื่อสารพอร์ตอนุกรม RS-232

ลักษณะของการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะส่งออกมาทีละบิต จากตัวส่งไปตัวรับข้อมูล ช่องสัญญาณในการส่งข้อมูลอาจใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารจะถูกกว่าแบบขนาน แต่อัตราการรับ-ส่งข้อมูลจะช้ากว่าแบบขนาน ในการส่งข้อมูลแบบอนุกรม ข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็น ไบต์จะทยอยส่งทีละบิต และทางตัวรับจะต้องรับข้อมูลเข้ามาทีละบิต แล้วมารวมกันเป็น ไบต์ซึ่งทางตัวรับต้องคอยตรวจสอบว่าบิตใดเป็นบิตเริ่มต้นหรือบิตสุดท้ายของข้อมูล การตรวจสอบ นั้นจะขึ้นอยู่กับรูปแบบของรหัสของบิตข้อมูลที่ใช้ ซึ่งในการรับส่งข้อมูลแบบอนุกรมระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น จำเป็นจะต้องมีมาตรฐานในการรับส่งข้อมูล ซึ่งมาตรฐานที่นิยมมากที่สุดก็คือ มาตรฐาน RS-232

##### มาตรฐาน RS-232

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันอย่างกว้างขวางที่สุดคือ RS-232C ซึ่งโดยปรกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรมอยู่ในตัวอยู่แล้ว และจะทำหน้าที่รับส่งข้อมูลในแบบอนุกรม

ตามจุดประสงค์ของมาตรฐาน RS-232C นั้นเพื่อจะสามารถเชื่อมต่อกันระหว่างอุปกรณ์รับส่งปลายทาง (Data Terminal Equipment : DTE) เช่น พอร์ตของคอมพิวเตอร์หลักหรืออุปกรณ์ปลายทางกับอุปกรณ์สื่อสาร RS-232 เป็นข้อกำหนดของการอินเตอร์เฟซมาตรฐาน และสามารถใช้เพื่อจุดประสงค์อื่นๆ ต่างๆกันไป เช่นการสื่อสารแบบซิงโครนัส (synchronous communication) และรูปแบบการสื่อสาร

ที่ต้องการสัญญาณนาฬิกา และสัญญาณกำหนดจังหวะเพิ่มเติมขึ้นมา ในความเป็นจริงแล้วเราสามารถทำเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปเผยแพร่บนอินเทอร์เน็ต ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ให้มีการสนทนากันระหว่าง DTE และ DCE โดยการใส่สายสัญญาณเพียง 3 เส้นเท่านั้น คือ ใช้สาย TD สาย RD และสายกราวด์เท่านั้น

หมายเลขขาสัญญาณ	ชื่อของสายสัญญาณ
1	Protective Ground
2	Transmitted Data
3	Received Data
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	Signal Common
8	Received Line Signal Detect
9	Reserved For Testing
10	Reserved For Testing
11	Unassigned
12	Secondary Received Line Signal Detect
13	Secondary Clear To Send
14	Secondary Transmitted Data
15	Transmission Signal Element Timing
16	Secondary Received Data
17	Receiver Signal Element Timing
18	Unassigned
19	Secondary Request To Send
20	Data Terminal Ready
21	Signal Quality Detector
22	Ring Indicator
23	Data Signal Rate Detector
24	Transmitter Signal Element Timing
25	Unassigned

ตารางที่ 2.3 ชื่อกำหนดของขาสัญญาณต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลขขาสัญญาณ	ชื่อของสายสัญญาณ
1	Protective Ground
2	Transmitted Data
3	Received Data
4	Request To Send
5	Clear To Send
6	Data Set Ready
7	Signal Common
8	Data Carrier Detect
20	Data Terminal Ready
22	Ring Indicator
23	Data Signal Rate Detector

ตารางที่ 2.4 แสดงรายละเอียดของสัญญาณที่ต่อจาก DTE ไปยัง DCE โดยใช้คอนเน็กเตอร์แบบ DB-25

หมายเลขขาสัญญาณ	ชื่อของสายสัญญาณ
1	Data Carrier Detect
2	Received Data
3	Transmitted Data
4	Data Terminal Ready
5	Signal Common
6	Data Set Ready
7	Request To Send
8	Clear To Send
9	Ring Indicator

ตารางที่ 2.5 รายละเอียดการต่อคอนเน็กเตอร์แบบ DB9 มาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 1 (Protective Ground Circuit ,AA) ขานี้จะต่อเข้ากับตัวถังของอุปกรณ์ และสามารถต่อเข้ากับกราวด์ภายนอกถ้าอุปกรณ์ต้องใช้งานขานี้
- ขา 12 (Transmitted Data Circuit BA,TD) เป็นขาสัญญาณข้อมูลที่ออกจากอุปกรณ์ DTE กระแสบิตอนุกรมจากขานี้ คือข้อมูลที่ถูกลำเลียงไปโดยโมเด็ม หรือถูกถอดรหัสโดยอุปกรณ์ DCE ที่มี
- ขา 13 (Receive Data Circuit BB,RD) สัญญาณที่ขานี้จะถูกสร้างจากอุปกรณ์ DCE กระบิตอนุกรมนี้จะกำเนิดขึ้นที่อุปกรณ์ DTE ปลายทางและเป็นผลผลิตของวงจรรับข้อมูลของอุปกรณ์ DCE สัญญาณนี้มักเป็นข้อมูลที่ถูกรับขึ้นโดยอุปกรณ์ DCE
- ขา 14 (Request to Send Circuit CA,RST) สัญญาณนี้จะเตรียมพร้อมอุปกรณ์ DCE สำหรับการทำการส่งข้อมูลเมื่อสัญญาณ RTS นี้อยู่ในสถานะ “ON” จะทำให้อุปกรณ์ DCE อยู่ในโหมดส่งข้อมูล (Transmit mode) ในขณะที่สัญญาณนี้อยู่ในสถานะ “OFF” ทำให้อุปกรณ์ DCE อยู่ในโหมดรับข้อมูล (Receive mode) อุปกรณ์ DCE ควรจะตอบสนองต่อสัญญาณ RST ON โดยการทำให้สัญญาณ Clear to Send (CTS) อยู่ในสถานะ “ON” ด้วยเมื่อ RTS อยู่ในสถานะ “OFF” สัญญาณนี้ไม่ควรจะ “ON” ขึ้นอีก จนกว่าสัญญาณ CTS อยู่ในสถานะ “OFF” เสียก่อน สัญญาณนี้จะถูกใช้ร่วมกับสัญญาณ DTR DSR และ DCD ขาสัญญาณ RTS จะถูกใช้อย่างมากในการควบคุมการไหลของข้อมูล
- ขา 15 (Clear to Send Circuit CB,CTS) สัญญาณนี้จะตอบรับกลับไปยังอุปกรณ์ DTE เมื่อได้รับสัญญาณ RTS และข้อมูลสามารถถูกส่งออกไปได้ ข้อมูลจะถูกส่งออกไปตามคircuitที่ใช้สื่อสารได้ ก็ต่อเมื่อสัญญาณ CTS นี้ อยู่ในสถานะ “ON” เท่านั้น สัญญาณนี้จะใช้ร่วมกับสัญญาณ DTR DSR และ DCD ขาสัญญาณนี้จะใช้ร่วมกับขา RTS สำหรับควบคุมการไหลข้อมูล
- ขา 16 (Data Set Ready Circuit CC,DSR) สัญญาณ DSR จะบอกต่ออุปกรณ์ DTE ว่าอุปกรณ์ DCE ได้ต่อกับตัวกลางการสื่อสารที่ถูกต้องแล้ว และในบางกรณีจะบ่งชี้ว่าสายโทรศัพท์ที่อยู่ในสถานะ “OFF HOOK” สถานะ “OFF HOOK” จะเป็นตัวบ่งชี้ว่าอุปกรณ์ DCE กำลังอยู่ในโหมด dialing หรือกำลังติดต่อกับอุปกรณ์ DCE อีกตัวหนึ่งอยู่ เมื่อสัญญาณ DSR นี้อยู่ในสถานะ “OFF” อุปกรณ์ DTE ก็ควรจะถูกกำหนดให้ไม่สนใจสัญญาณอื่นๆ ทั้งหมดจากอุปกรณ์ DCE ถ้าสัญญาณนี้ถูกทำให้ อยู่ในสถานะ “OFF” ก่อนอุปกรณ์ DTR แล้วอุปกรณ์ DCE ก็จะสรุปว่าการสื่อสารนั้นสิ้นสุดลง
- ขา 17 (Signal Common Circuit AB) สายนี้จะให้สัญญาณอ้างอิงของกราวด์ร่วมกันสำหรับวงจรการแลกเปลี่ยนข้อมูลทั้งหมด ยกเว้นวงจร AA หรือ Protective Ground ข้อกำหนด RS-232B จะอนุญาตให้วงจรนี้ถูกตัดต่อเพิ่มเติมกับ Protective Ground ภายในอุปกรณ์ DCE ได้ ถ้าจำเป็น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท ทรานซิสเตอร์ จำกัด ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 8 (Data Carrier Detect Circuit CF,DCD) ขานี้รู้จักกันในนามของ Receive Line Signal Detect (RLSD) หรือขา Carrier Detect (CD) สัญญาณนี้จะเกิด Active เมื่อเกิดสัญญาณพาหะที่เหมาะสมระหว่างอุปกรณ์ DCE ที่สถานีกับที่อยู่ระยะไกล เมื่อสัญญาณนี้อยู่ในสถานะ “OFF” สัญญาณที่ขา RD ควรจะถูกทำให้ค้างอยู่ในสถานะ “Mark” (สถานะ “1” ในเลขฐานสอง)
- ขา 20 (Data Terminal Ready Circuit CD,DTR) สัญญาณ DTR ถูกใช้ในการควบคุมการสวิตช์ อุปกรณ์ DCE เข้ากับตัวกลางในการสื่อสารสัญญาณ DTR ON บ่งชี้ว่าอุปกรณ์ DCE ที่กำลังต่อเชื่อมกันอยู่ ก็ยังต่อร่วมกัน และถ้ายังไม่มีการเชื่อมต่อกันก็สามารถทำให้มีการเชื่อมต่อกันครั้งใหม่ได้ ปกติแล้วสัญญาณ DTR จะอยู่ในสถานะ “OFF” เพื่อกระตุ้นให้เกิดภาวะ “ON HOOK” (วางสาย) อุปกรณ์ DCE โดยปกติแล้วจะตอบสนองต่อการกระตุ้นจากสัญญาณ DTR โดยการทำให้สัญญาณ DSR แยกทีฟ
- ขา 22 (Ring Indicator Circuit CE,RI) สถานะ “ON” ของขานี้จะบ่งชี้ว่า ได้รับสัญญาณเรียกสายโทรศัพท์ที่จากตัวกลางในการสื่อสาร โดยปกติแล้วจะขึ้นอยู่กับโปรแกรมควบคุม ในการที่จะทำให้เกิดสัญญาณนี้ขึ้นหรือไม่
- ขา 23 Data Signal Rate Detector Circuit CH/CI DSRD) วงจร CH เป็นส่วนประกอบของ DTE และวงจร CI เป็นส่วนประกอบของ DCE สัญญาณนี้ถูกใช้ในการเลือกใช้อัตราในการส่งข้อมูลค่าใดค่าหนึ่งในสองค่า ในกรณีที่ใช้โมเด็มที่มีอัตราการส่งข้อมูลได้ 2 ค่า (Dual Rate Modems) ถ้าสัญญาณนี้เป็น “ON” ก็จะเป็นการเลือกอัตราการส่งข้อมูลที่มีค่าสูงสุดในสองค่านั้น

#### ขั้นตอนการติดต่อระหว่างอุปกรณ์ DTE และ DCE

1. เมื่อจ่ายกำลังงานให้กับ DTE และอุปกรณ์ก็จะส่งสัญญาณ DTR ออกมา
2. อุปกรณ์ DCE ถูกเปิดขึ้นและรับรู้สัญญาณ DTR ที่ส่งมาจากอุปกรณ์ DTE
3. อุปกรณ์ DCE ส่งสัญญาณ DSR ออกมา และโมเด็มก็กระทำกระบวนการ OFF HOOK
4. ถ้าสายสัญญาณอยู่ในสภาพดีและปลายทางอีกด้านหนึ่งก็พร้อมจะรับข้อมูลแล้ว โดยจะตรวจจับพบสัญญาณพาหะ แล้วอุปกรณ์ DCE จะส่งสัญญาณ DCD ออกมา
5. อุปกรณ์ DCE จะตอบสนองด้วยการ ส่งสัญญาณ CTS ออกมา
6. การติดต่อสื่อสารก็เริ่มขึ้น โปรแกรมควบคุมจะทำการส่งหรือรับข้อมูล

#### ส่วนลำดับขั้นในการตอบรับก็เป็นในทำนองนี้

1. อุปกรณ์ DTE จะส่งสัญญาณ DTR ออกมา
2. อุปกรณ์ DCE จะอยู่ในโหมดตอบรับอัตโนมัติ(auto answer mode)โดยมีสัญญาณ DSR ออกมา
3. สถานีปลายทางส่งสัญญาณเรียกอุปกรณ์ DCE และอุปกรณ์ DCE ส่งสัญญาณ RI ออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. อุปกรณ์ DTE รับรู้ถึงอุปกรณ์ RI ที่ส่งมาจากเครื่องปลายทาง และอุปกรณ์ DCE ก็เข้าสู่สถานะ OFF HOOK
5. อุปกรณ์ DCE ทำการแลกเปลี่ยนข้อมูลกับอุปกรณ์ DCE ที่อีกปลายทางหนึ่ง และมีการส่งสัญญาณ DCD ออกมา
6. อุปกรณ์ DTE จะส่งสัญญาณ RTS ออกมาหรืออาจจะรอข้อมูลก็ได้ ขึ้นอยู่กับโปรแกรมควบคุม
7. อุปกรณ์ DCE จะตอบสนองด้วยการส่งสัญญาณ DTS กลับออกมา
8. การติดต่อสื่อสารก็จะเริ่มขึ้น

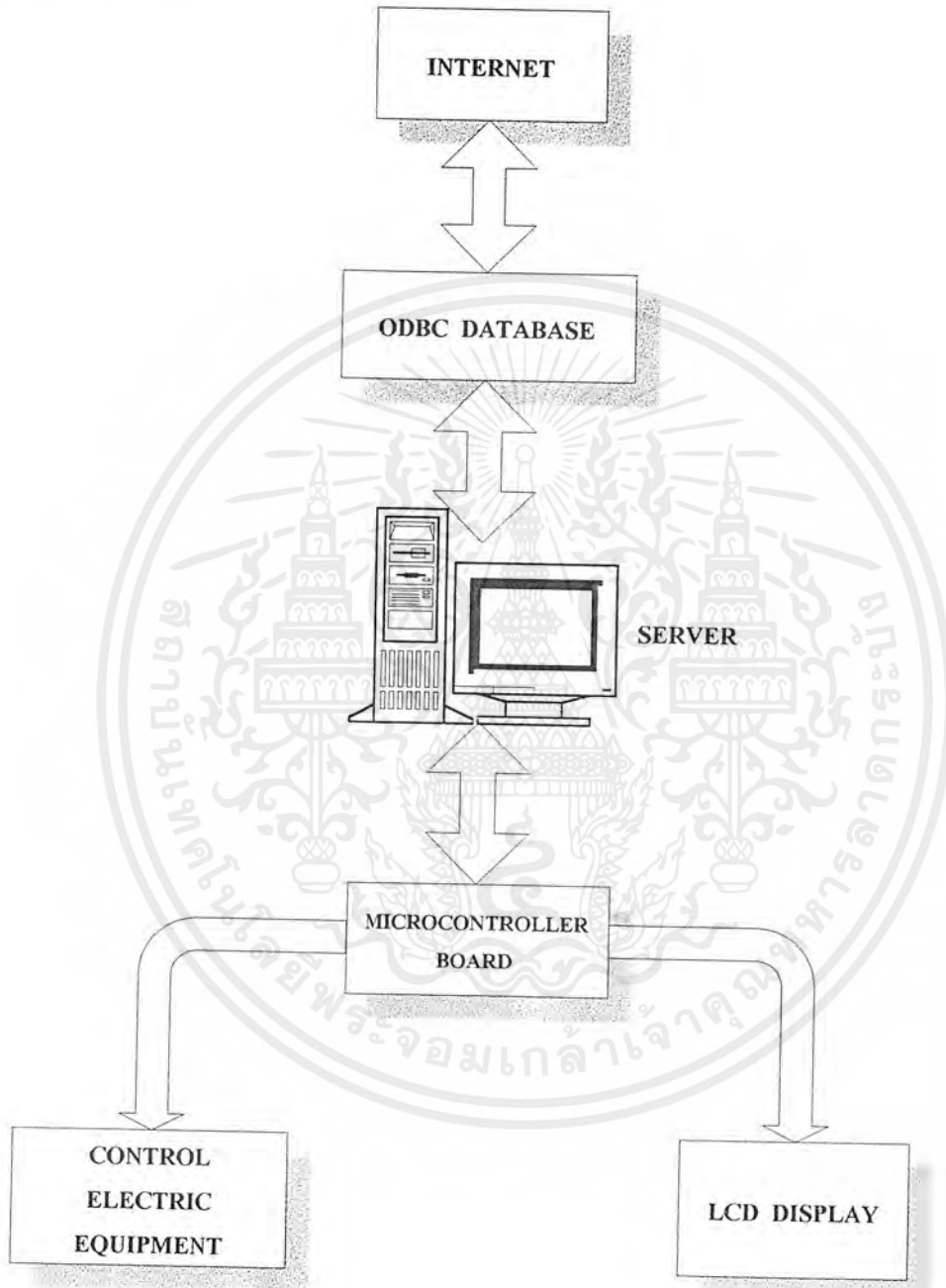


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

## การคำนวณและการสร้าง

## 3.1 แสดงหลักการทำงาน



รูปที่ 3.1 แสดงบล็อกไดอะแกรมรวมของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### หลักการการทำงานของโครงการ

จากบล็อกไดอะแกรมแสดงการทำงานของโครงการเครื่องควบคุมอุปกรณ์ไฟฟ้าที่เครื่องลูกข่าย (CLIENT) ผ่านอินเทอร์เน็ต โดยรับข้อมูลการสั่งงานจากบราวเซอร์ เช่น INTERNET EXPLORER ,NETSCAPE โดยใช้ เอเอสพี จะติดต่อฐานข้อมูลผ่านทาง ODBC (Open Database Connectivity) โดยต้องสร้างส่วนในการติดต่อกับฐานข้อมูล DSN (Data Source Name) นำข้อมูลที่ได้ไปทำการตรวจสอบและแก้ไขฐานข้อมูล

เครื่องลูกข่ายจะส่งสัญญาณร้องขอ(Request) เพื่อขอควบคุมไฟฟ้าไปยังเครื่องเซิร์ฟเวอร์ (Server) เครื่องเซิร์ฟเวอร์ก็จะส่งหน้าค่างที่ใช้ในการควบคุมอุปกรณ์ไฟฟ้าไปให้เครื่องลูกข่าย โดยเครื่องเซิร์ฟเวอร์ จะทำการแปลงข้อมูลแบบ เอเอสพี ให้เป็น HTML ส่งไปให้เครื่องลูกข่าย ( ในการทำโครงการครั้งนี้จะเขียนหน้าค่างที่ใช้ในการส่งควบคุมอุปกรณ์ไฟฟ้าด้วย เอเอสพี )

เมื่อมีการส่งควบคุมอุปกรณ์ไฟฟ้า เครื่องลูกข่ายก็จะทำการส่งข้อมูลการควบคุมมาเก็บไว้ในดาต้าเบส ที่เครื่องเซิร์ฟเวอร์ โดยติดต่อฐานข้อมูลผ่าน ODBC

หลังจากนั้นเครื่องเซิร์ฟเวอร์ก็จะนำข้อมูลการควบคุมอุปกรณ์ไฟฟ้าจากฐานข้อมูลส่งออกทางพอร์ทอนุกรม(SerialPort) ไปส่วนควบคุมอุปกรณ์ไฟฟ้าซึ่งควบคุมการทำงานโดยไม่โครคอนโทรลเลอร์ MCS-51 ซึ่งโครงการนี้ใช้โปรแกรมวิซวลเบสิก ในการเขียนโปรแกรมติดต่อกับไมโครคอนโทรลเลอร์ MCS-51

## 3.2 แผนผังการทำงานของเครื่องสั่งงานเครื่องใช้ไฟฟ้า

### 3.2.1 บล็อกไดอะแกรมแสดงการทำงานในส่วนเอเอสพี

#### หลักการทำงาน

จากบล็อกไดอะแกรมของส่วนของโฮมเพจจะเห็นว่าประกอบด้วย หน้าค่างหลักๆ 6 หน้าค่าง ซึ่งประกอบด้วย

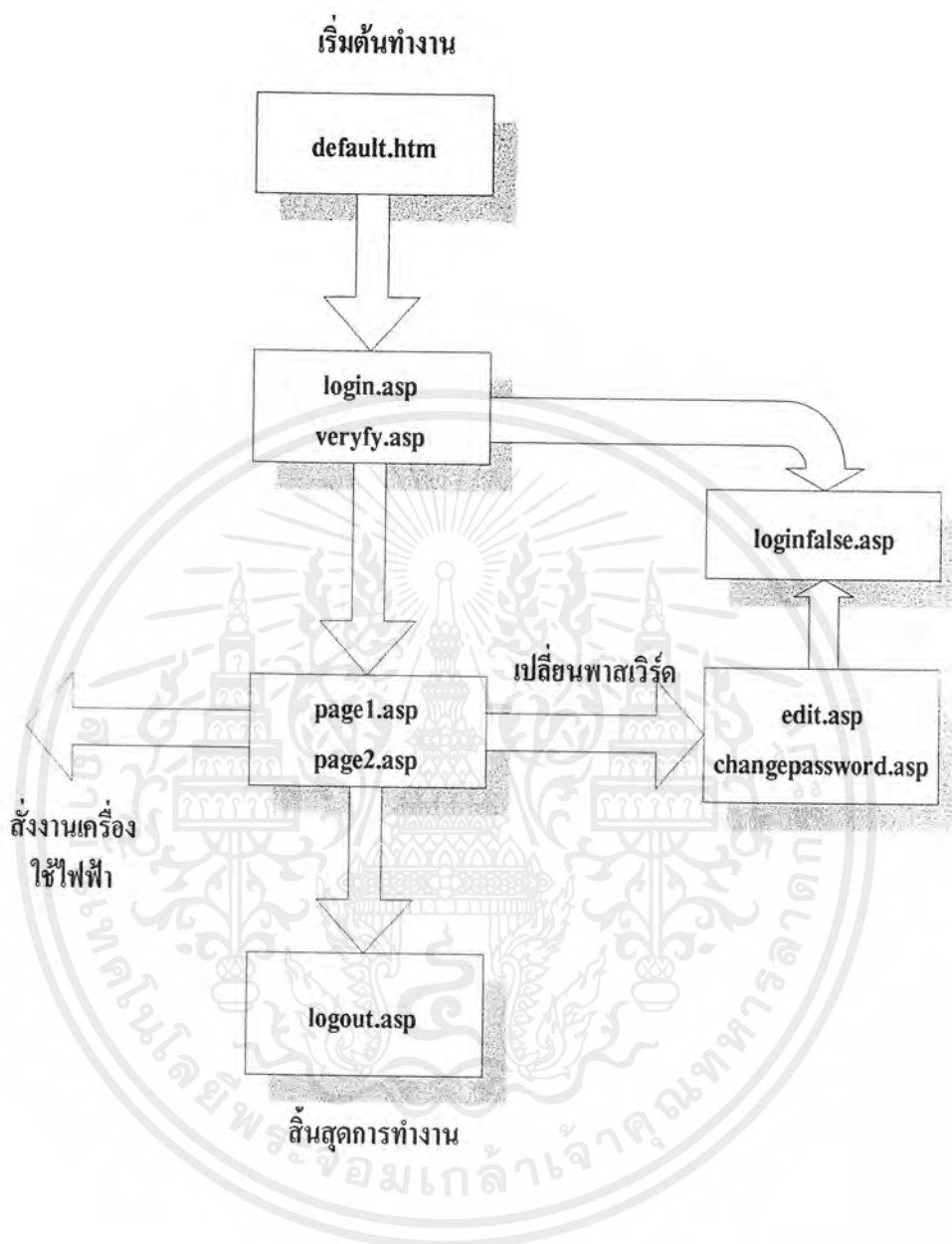
1. หน้าค่าง default ทำหน้าที่รีเฟรช (refresh) หน้าค่าง login เพื่อความปลอดภัยของโครงการ (security)
2. หน้าค่าง login ทำหน้าที่ป้อน login, พาสเวิร์ด (password) โดยถ้า พาสเวิร์ด ไม่ถูกต้องจะถูกหลุดไปที่หน้าค่าง loginfalse.asp ซึ่งต้องป้อน ล็อกอินพาสเวิร์ดใหม่ โดยประกอบด้วย เอเอสพี 2 หน้า คือ login.asp และ verify.asp โดย login.asp ทำหน้าที่เป็นตัวคอยรับ ล็อกอิน พาสเวิร์ด และ โปสท์ (post) ค่าให้กลับ verify.asp ส่วน verify.asp ทำหน้าที่ในการตรวจสอบ ล็อกอิน พาสเวิร์ด ว่าถูกต้องหรือไม่และคอยติดต่อกับฐานข้อมูลที่ เซิร์ฟเวอร์
3. หน้าค่าง สั่งงานเครื่องใช้ไฟฟ้า ซึ่งประกอบด้วย page1.asp,page2.asp ซึ่ง page1.asp สั่งงาน อุปกรณ์ไฟฟ้าและตรวจสอบความปลอดภัย (ตรวจสอบว่าเข้ามาถูกต้องหรือไม่) และคอย โปสท์ ค่าให้กับ page1.asp โดย page2.asp จะทำการตรวจสอบค่าต่าง ๆ ถูกต้องหรือไม่ (เข้ามาถูกต้องหรือไม่,ค่าช่องอุปกรณ์ 1-8 หรือไม่ สถานนะ เปิด (1) ปิด (0) หรือไม่) ถ้าผิดพลาดให้หลุดไปที่ หน้าค่าง loginfalse.asp และอ่านค่าบันทึกฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. หน้าต่างเปลี่ยน รหัสเวิร์ด ซึ่งประกอบไปด้วย edit.asp,changepassword.asp ซึ่ง edit.asp ทำหน้าที่รับค่าในการเปลี่ยน รหัสเวิร์ด และ โปสท์ ค่าให้กับ changepassword.asp เพื่อในการตรวจสอบและบันทึกฐานข้อมูล
5. หน้าต่าง ล็อกเอาท์ ประกอบด้วยหน้าต่าง logout.asp ซึ่งทำหน้าที่เป็นหน้าต่างในการสิ้นสุดการทำงานและเคลียร์ค่าตัวแปร
6. หน้าต่าง ล็อกอิน ผิดพลาดประกอบด้วยหน้าต่าง loginfalse.aspเป็นหน้าต่างในการป้อน ล็อกอิน และ รหัสเวิร์ด ใหม่ โดยเมื่อเกิดการผิดพลาด ( ล็อกอิน และ รหัสเวิร์ด เข้ามาโดยไม่ถูกต้อง ) จะหลุดมาที่หน้าค่างนี้เสมอ



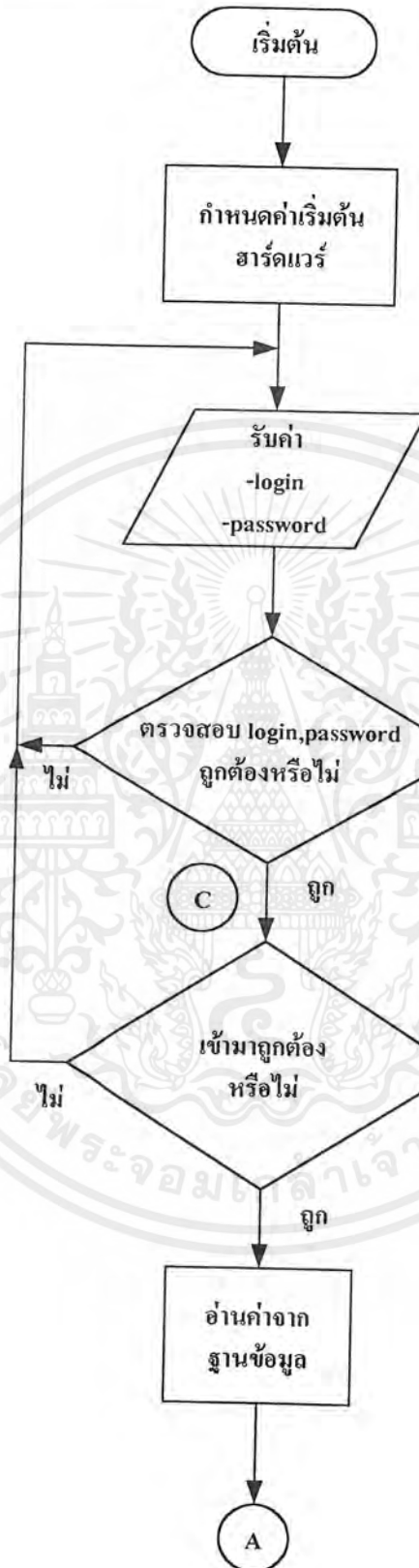
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงบล็อกไดอะแกรมการทำงานของ เอเอสพี

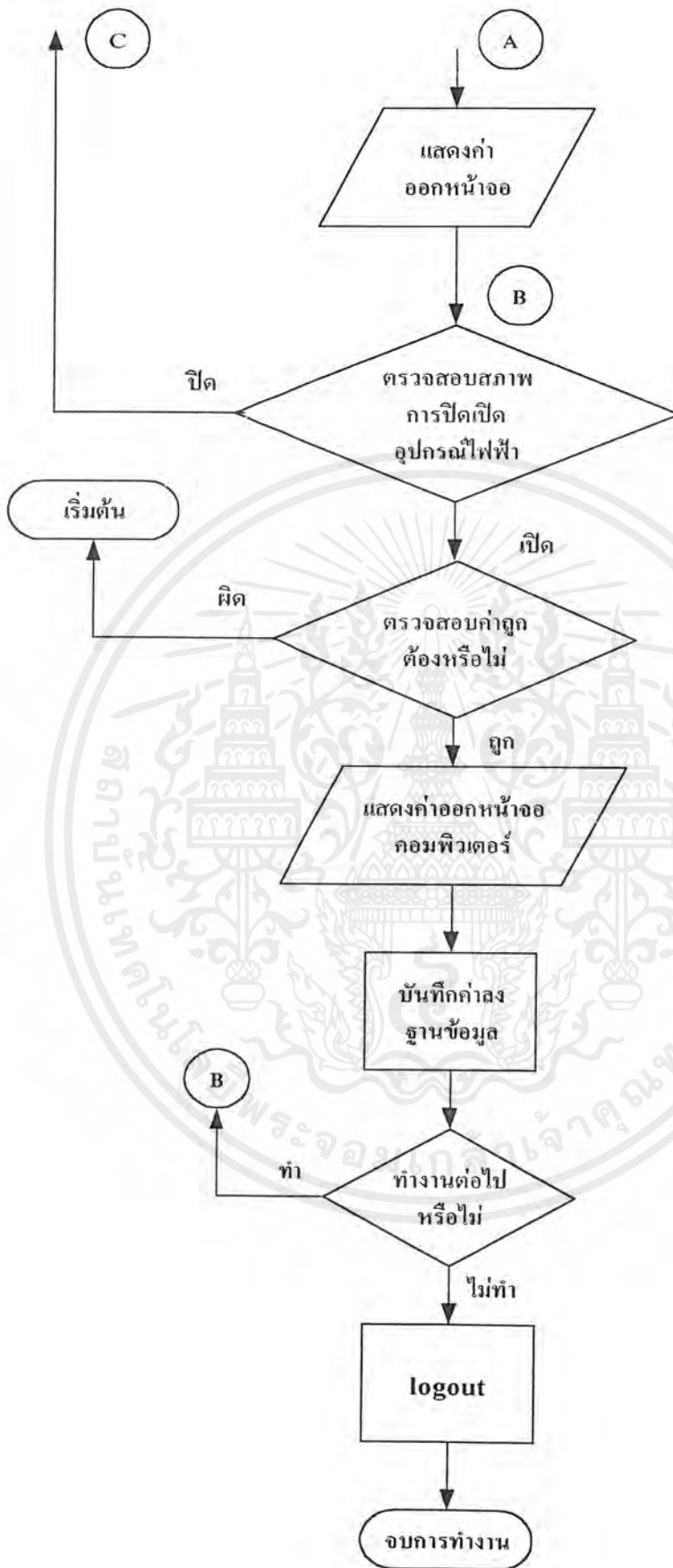
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2.2 แผนผังการทำงานในส่วน เอเอสพี



รูปที่ 3.3 ก แสดงผังการทำงานของ เอเอสพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ข แสดงผังการทำงานของ เอเอสพี

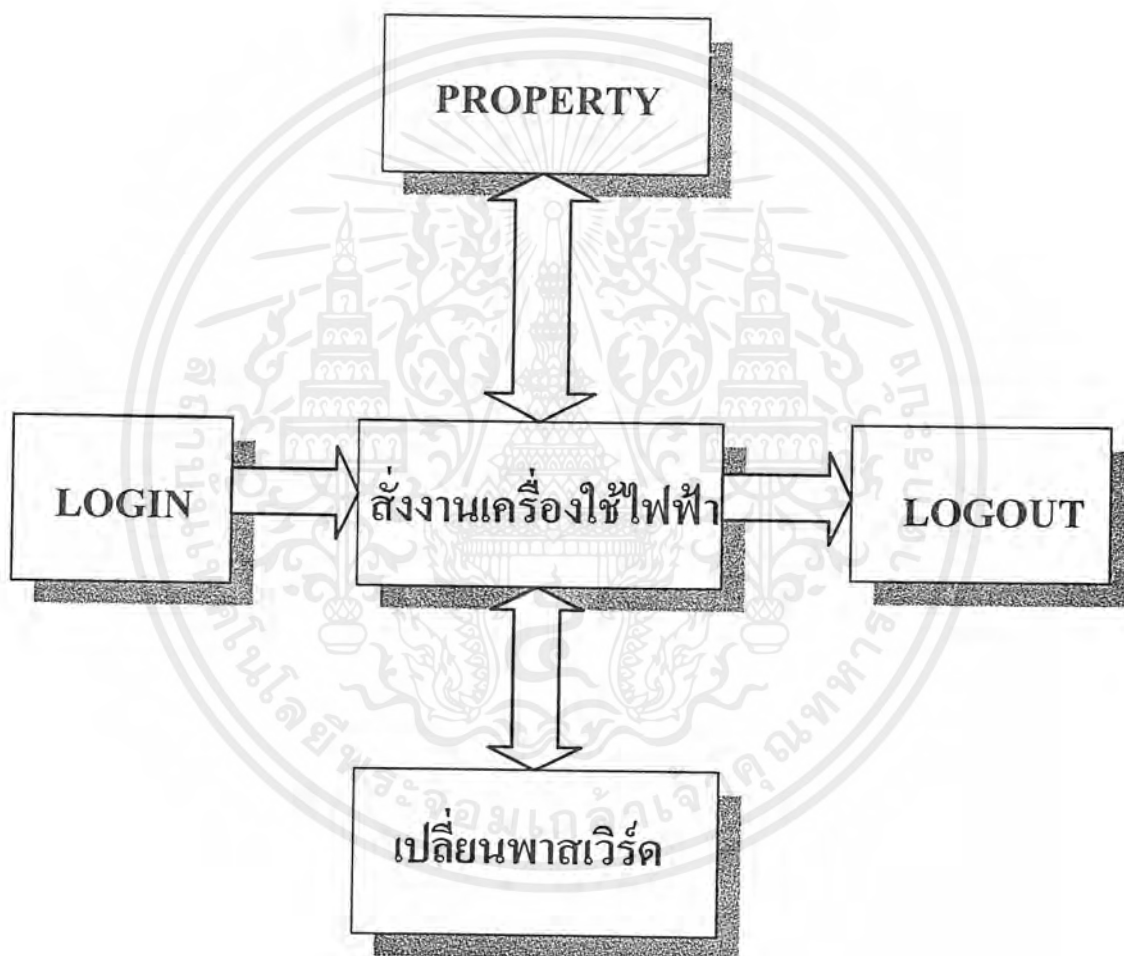
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 แสดงหลักการทำงานในส่วน วิชาเว็บลิค

#### 3.3.1 บล็อกไดอะแกรมการทำงานของ วิชาเว็บลิค

##### หลักการทำงาน

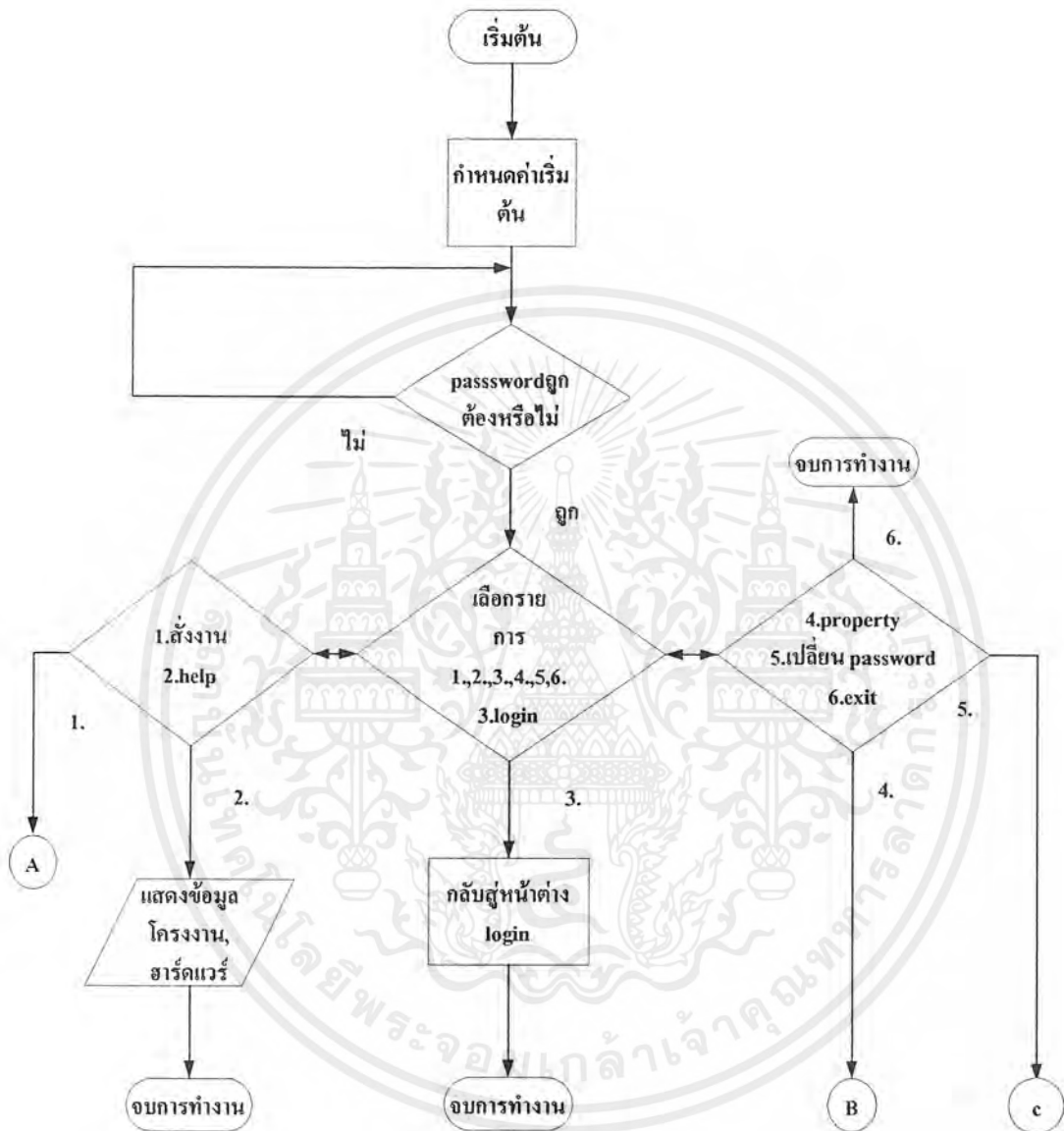
จากบล็อกไดอะแกรมของ วิชาเว็บลิคจะเห็นได้ว่าเมื่อเราต้องการสั่งงานจะต้องผ่านหน้าต่าง login ก่อน จากนั้นจึงเข้าสู่หน้าต่างสั่งงานเครื่องใช้ไฟฟ้าซึ่งสามารถสั่งอุปกรณ์ไฟฟ้าได้ 8 ช่องโดยในหน้าต่างนี้เราสามารถทำการเปลี่ยนแปลงพาสเวิร์ดได้และก็สามารถที่จะกำหนดคุณสมบัติของอุปกรณ์ต่างๆ ที่เราทำการควบคุมรวมทั้งกำหนด พอร์ตได้ด้วย



รูปที่ 3.4 แสดงบล็อกไดอะแกรมในการทำงานของ วิชาเว็บลิค

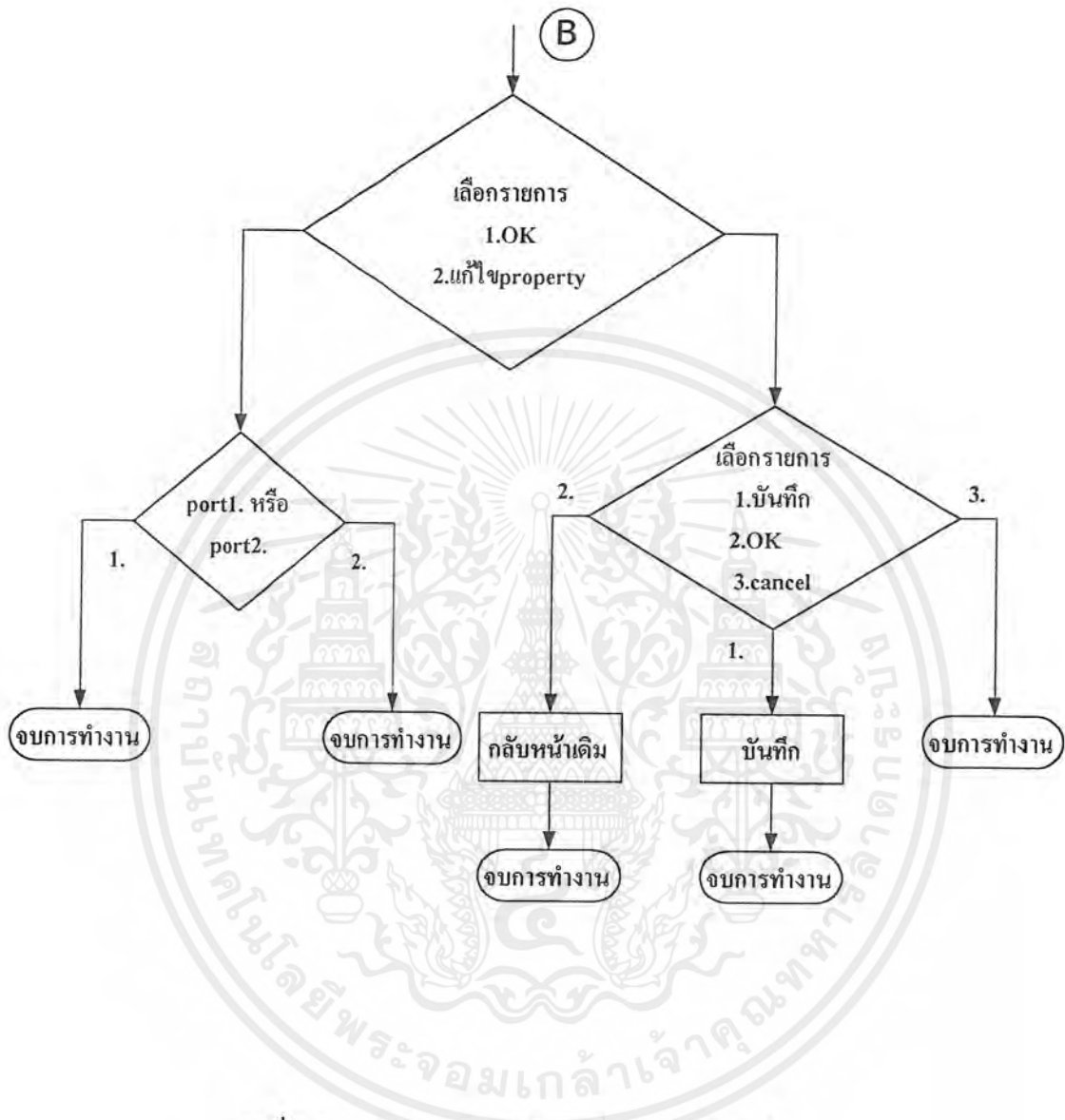
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.3.2 ฟังก์ชันการทำงานในส่วนวิชวลเบสิก



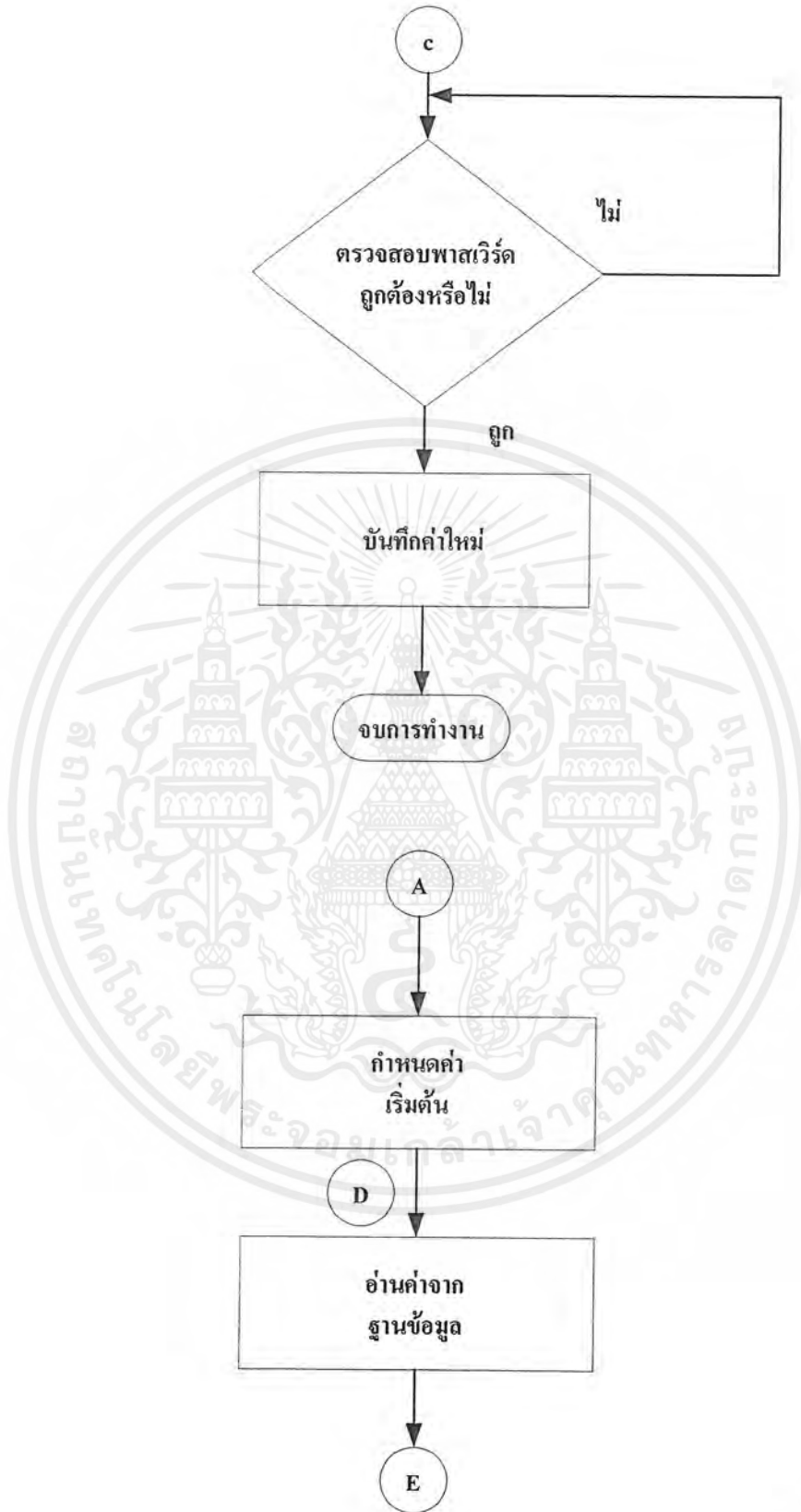
รูปที่ 3.5 ก แสดงฟังก์ชันการทำงานในส่วนของ โปรแกรมวิชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



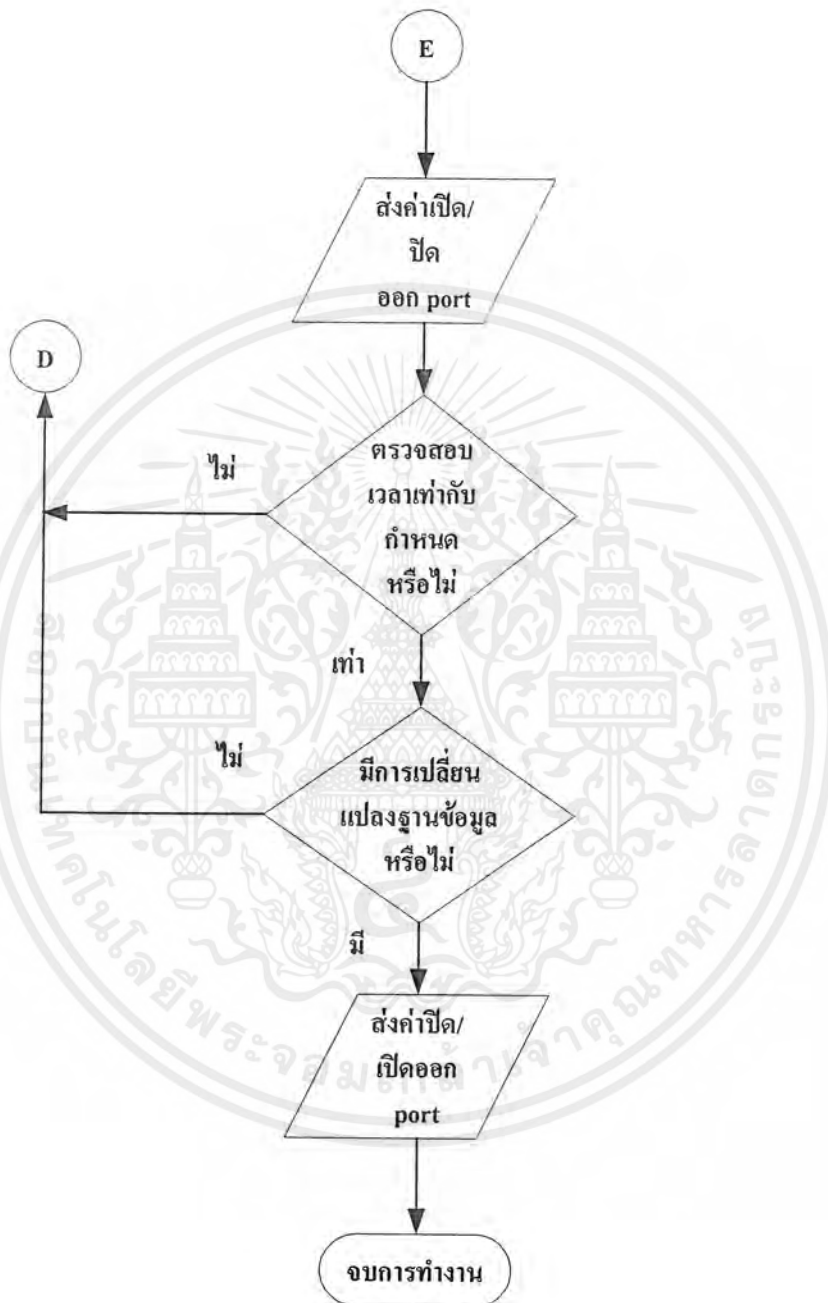
รูปที่ 3.5 ข แสดงแผนผังการทำงานในส่วน วิชาลเวสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 ค แสดงผังการทำงานในส่วน วิชาลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

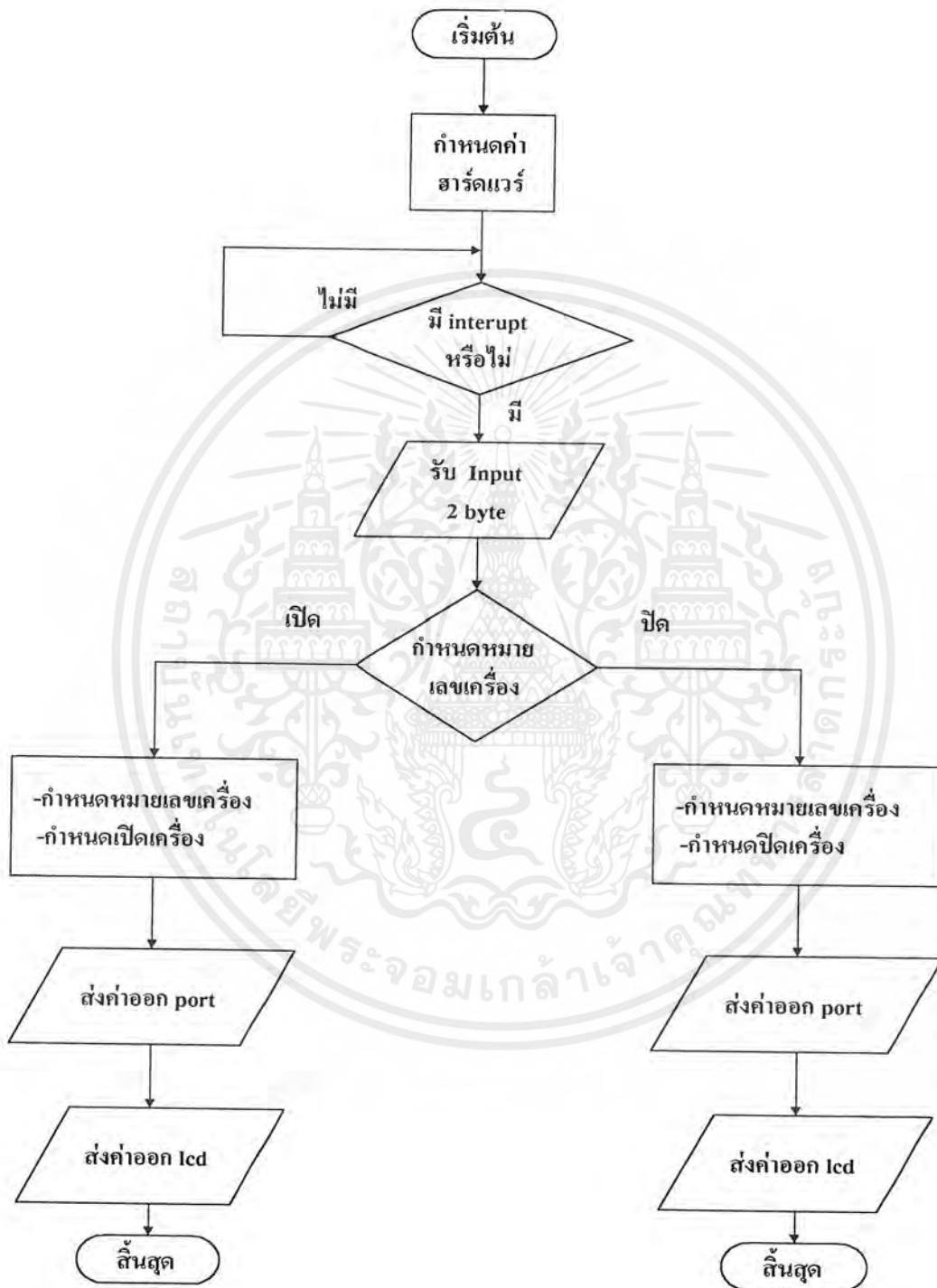


รูปที่ 3.5 ง แสดงผังการทำงานในส่วน วิชาลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4 แสดงหลักการทำงานของไมโครคอนโทรลเลอร์

#### 3.4.1 ฟังก์ชันการทำงานของไมโครคอนโทรลเลอร์



รูปที่ 3.6 แสดงฟังก์ชันการทำงานในส่วนไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

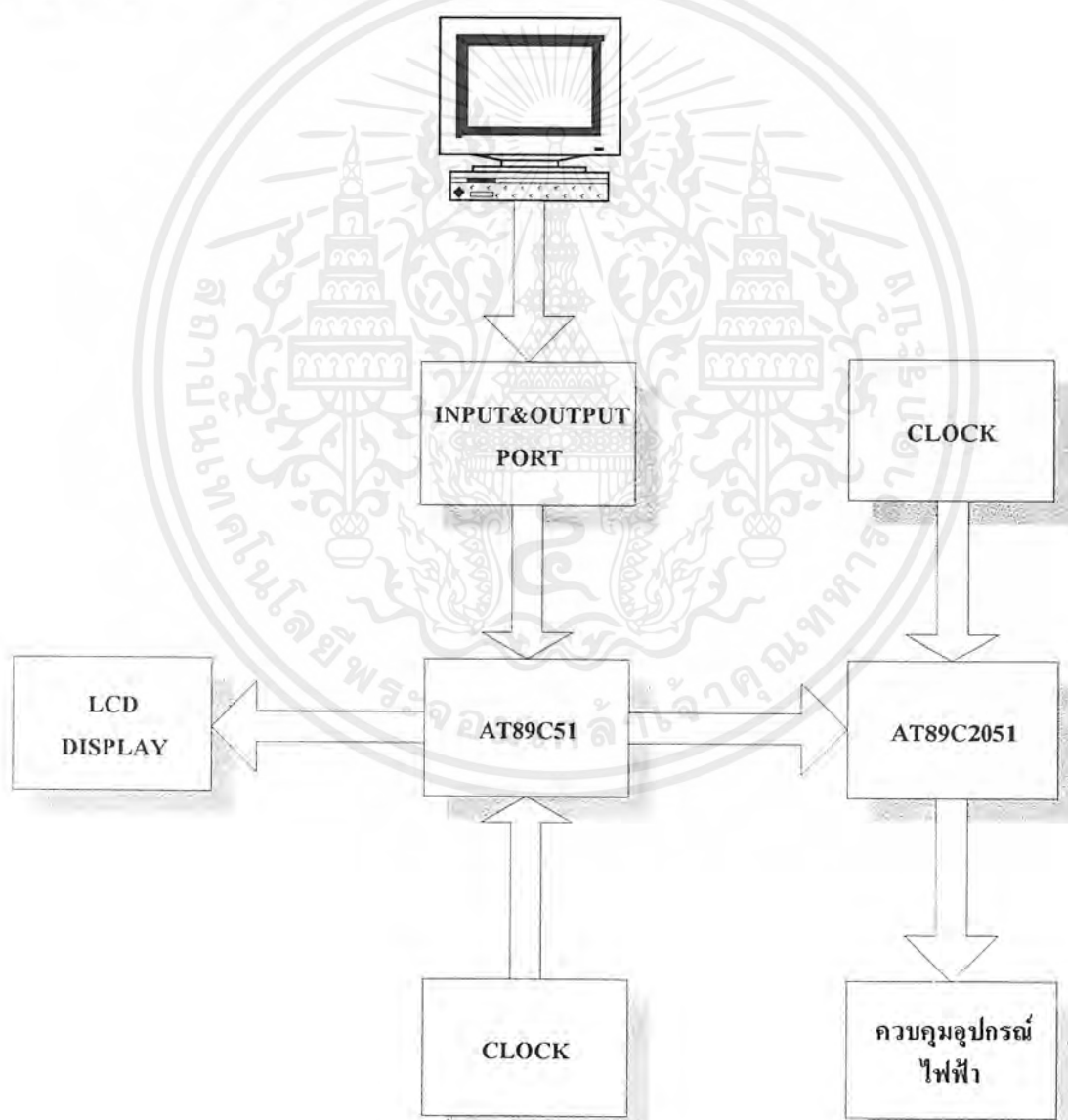
### 3.4.2 บล็อกไดอะแกรมในส่วนระบบไมโครคอนโทรลเลอร์

ในโครงการนี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51 ทั้งนี้เพื่อสะดวกในการทดลองและใช้งานจริงได้ ซึ่งส่วนประกอบของไมโครคอนโทรลเลอร์นี้มีส่วนประกอบดังบล็อกไดอะแกรมดังนี้

- ส่วนของไมโครคอนโทรลเลอร์เอง ซึ่งมีส่วนเชื่อมต่อกันระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอกในตัวอยู่แล้ว

- ส่วนของโปรแกรมเมโมรี่ซึ่งใช้ในการเก็บข้อมูลซึ่งมีอยู่แล้วภายในตัวไมโครคอนโทรลเลอร์เอง

- ส่วนของสัญญาณนาฬิกาซึ่งใช้เป็นตัวสร้างสัญญาณนาฬิกาให้ตัวไมโครคอนโทรลเลอร์เอง โดยตัวไมโครคอนโทรลเลอร์จะเป็นตัวคอยรับคำสั่งจากคอมพิวเตอร์จากนั้นจึงแปลงสัญญาณเป็นสัญญาณอนุกรมเพื่อจะได้ส่งสัญญาณไปควบคุมได้ไกลๆ จากนั้นจะมีไมโครคอนโทรลเลอร์คอยแปลงสัญญาณอนุกรมมาเป็นสัญญาณขนานอีกครั้งเพื่อควบคุมปิดเปิดอุปกรณ์ไฟฟ้า



รูปที่ 3. 7 แสดงบล็อกไดอะแกรมระบบไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.3 วงจรการทำงานของไมโครคอนโทรลเลอร์

#### หลักการการทำงานของวงจร

เมื่อไมโครคอนโทรลเลอร์ MCS-51(AT89C51) ถูกอินเทอร์รัพ ไมโครคอนโทรลเลอร์ก็จะรับข้อมูลเข้ามาทางขาที่ 10 และขาที่ 11 โดยที่ขา 10 เป็นขารับข้อมูล (RXD) ขาที่ 11 เป็นขาส่งข้อมูล (TXD) แล้วนำข้อมูลไปตรวจสอบว่าข้อมูลที่รับได้ส่งควบคุมอุปกรณ์ไฟฟ้าขึ้นใดหรือช่องสัญญาณใดหลังจากนั้นไมโครคอนโทรลเลอร์ MCS-51 (AT89C51) ก็จะทำงานสองอย่างคือ

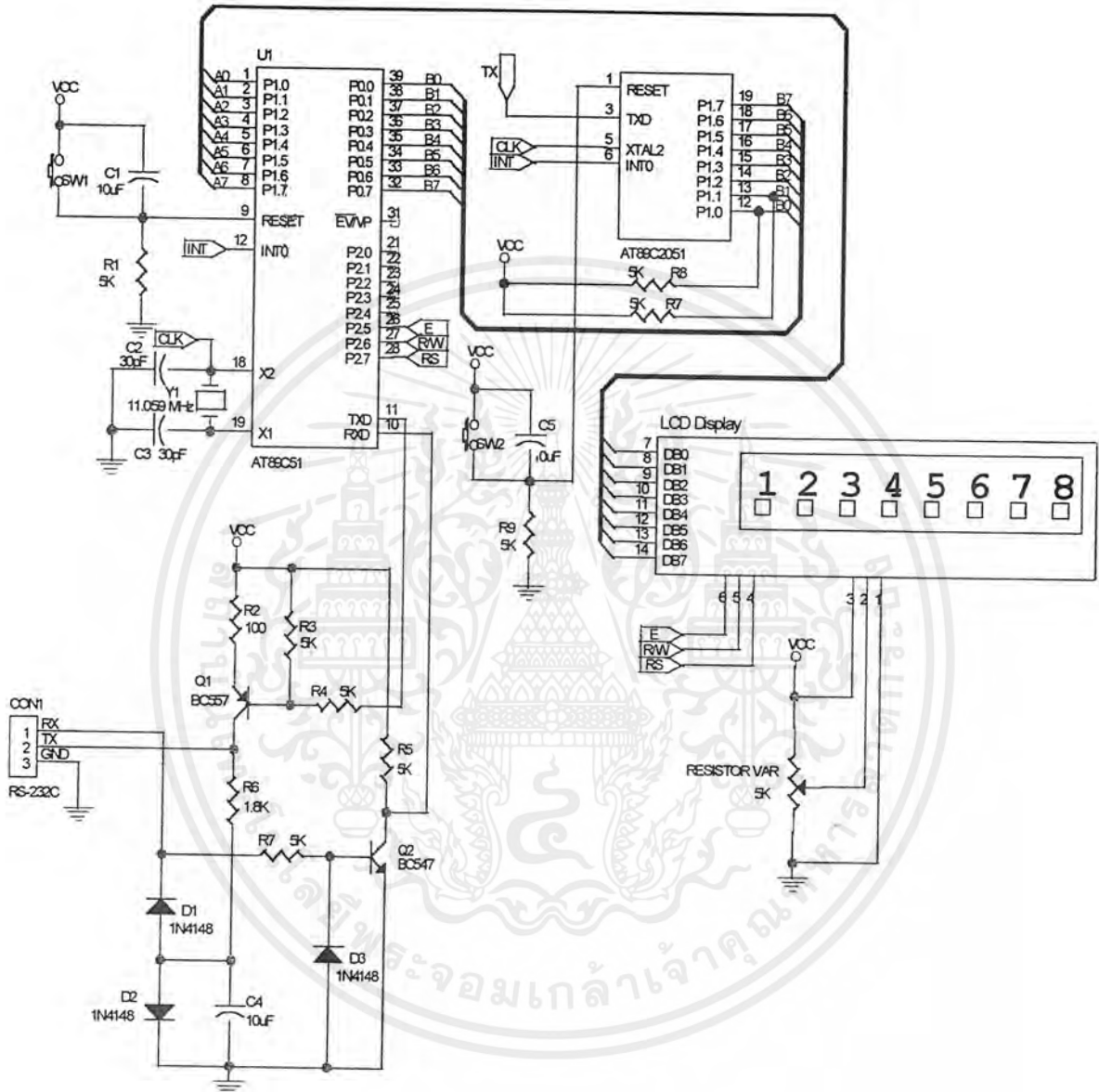
1. ส่งข้อมูลควบคุมออกไปที่ พอร์ทบี ( Port B ) ออกมาในลักษณะข้อมูลแบบขนาน ไปให้ไมโครคอนโทรลเลอร์ MCS-51 (AT89C2051) ซึ่งไมโครคอนโทรลเลอร์ตัวนี้จะทำหน้าที่แปลงข้อมูลที่เป็นข้อมูลแบบขนาน ให้เป็นข้อมูลแบบอนุกรม (เหตุที่ต้องแปลงให้เป็นข้อมูลแบบอนุกรม ก็เพื่อประโยชน์ในกรณีที่อุปกรณ์ไฟฟ้าที่ต้องการจะควบคุมอยู่ไกลจากเครื่องควบคุม ถ้าไม่แปลงเป็นข้อมูลอนุกรม เราก็ต้องเดินสายควบคุมทั้ง 8 เส้นไปยังอุปกรณ์ไฟฟ้าที่ต้องการควบคุม ทำให้สิ้นเปลืองในด้านค่าใช้จ่ายในการซื้อสาย) เมื่อไมโครคอนโทรลเลอร์ MCS-51 (AT89C2051) แปลงข้อมูลเป็นแบบอนุกรมแล้วก็จะส่งข้อมูลออกทางขา 3 (TXD) ไปควบคุมอุปกรณ์ไฟฟ้าที่อยู่ไกลๆออกไป โดยที่ปลายทางก็จะมีวงจรที่ทำหน้าที่แปลงข้อมูลที่เป็นแบบอนุกรมให้กลับมาเป็นข้อมูลแบบขนานอีกครั้งหนึ่ง ซึ่งหลักการทำงานของวงจรจะกล่าวโดยละเอียดในหัวข้อต่อไป

2. ส่งข้อมูลออกทางพอร์ทเอ (port A) ไปแสดงสถานะของอุปกรณ์ไฟฟ้าทั้งหมดที่ส่งควบคุมว่าช่องสัญญาณใดมีสถานะ เปิด หรือ ปิด

จากรูปจะเห็นได้ว่า สัญญาณที่รับมาจากเซิร์ฟเวอร์ซึ่งเป็นสัญญาณตามมาตรฐาน RS232C ก่อนที่สัญญาณจะเข้าที่ขา 10 (TXD) และขา 11 (RXD) ของไมโครคอนโทรลเลอร์ MCS-51(AT89C51) จะผ่านวงจรชุดหนึ่งดังรูป ซึ่งจะทำหน้าที่แปลงสัญญาณแบบ RS232C ให้เป็นสัญญาณที่ใช้ได้กับไมโครคอนโทรลเลอร์ MCS-51 คือ ลอจิก 1 เท่ากับ +5 โวลต์ ลอจิก 0 เท่ากับ 0 โวลต์

ส่วนตัวต้านทานแบบปรับค่าได้ ( Resistor Var 5K ) ที่ต่ออยู่กับขา 1 , 2 และขา 3 ของ แอล ซี ดี นั้นทำหน้าที่ปรับค่าความสว่างของจอ แอล ซี ดี

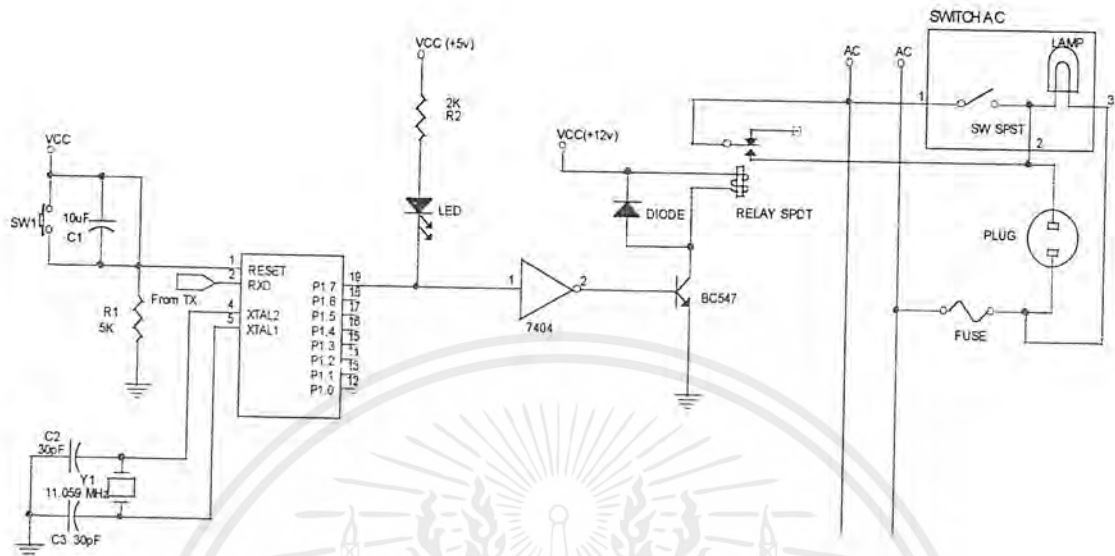
### 3.4.3 ส่วนของวงจรควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 3.8 วงจรควบคุมอุปกรณ์ไฟฟ้าทางด้านส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.4 ส่วนของวงจรควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 3.9 วงจรควบคุมอุปกรณ์ไฟฟ้าทางค่านับ

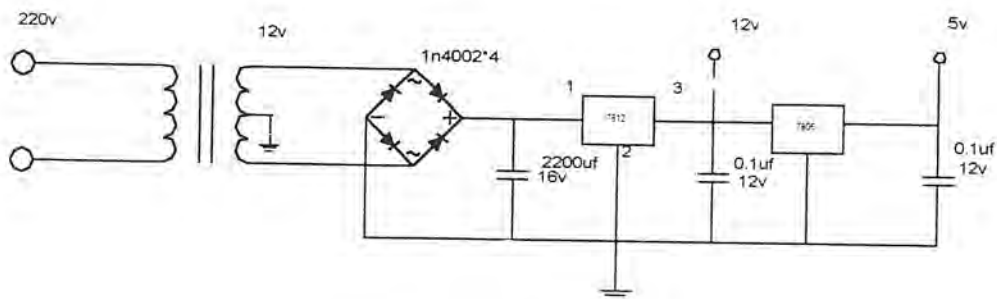
#### การทำงานของวงจร

ไมโครคอนโทรลเลอร์ MCS-51 (AT89C51) จะทำหน้าที่รองรับสัญญาณที่ส่งมาจากภาคส่ง ซึ่งเป็นสัญญาณแบบอนุกรมหลังจากนั้นก็แปลงให้เป็นข้อมูลแบบขนาน แล้วส่งออกพอร์ทที่ 1 (ขาที่ 12 ถึงขาที่ 19) หลังจากนั้นก็นำสัญญาณแต่ละขาไปควบคุมการปิดเปิดอุปกรณ์ไฟฟ้า จากวงจรจะเห็นได้ว่า เราสามารถควบคุมอุปกรณ์ไฟฟ้าได้ทั้งหมด 8 ช่องสัญญาณ เนื่องจากพอร์ท 1 มีขาสัญญาณ 8 ขานั่นเอง

จากรูปจะเป็นการแสดงการควบคุมอุปกรณ์ไฟฟ้าเพียงช่องสัญญาณเดียวเท่านั้น โดยมีการทำงานดังนี้ คือ โดยปกติสัญญาณที่ส่งออกมาที่ Port 1 จะเป็นลอจิก 1 (ลอจิก 1 หมายถึงการสั่งปิดอุปกรณ์ไฟฟ้า) ถ้ามีการสั่งให้มีการเปิดอุปกรณ์ไฟฟ้า ไมโครคอนโทรลเลอร์ MCS-51(AT89C51) ก็จะส่งลอจิก 0 ออกมาซึ่งจะทำให้ แอล อี ดี ที่ทำหน้าที่แสดงสถานะของอุปกรณ์ไฟฟ้าติดด้วย หลังจากนั้นสัญญาณควบคุม (ลอจิก 0) ก็จะผ่านวงจร NOT GATE กลับสัญญาณให้เป็นลอจิก 1 เพื่อนำไปไบอัสให้ทรานซิสเตอร์นำกระแส (BC547) เมื่อทรานซิสเตอร์นำกระแส ก็จะทำให้มีกระแสไหลผ่านขดลวดของรีเลย์(Relay) ทำให้หน้าสัมผัสของรีเลย์เปลี่ยนตำแหน่ง(ไดโอดที่ต่อคร่อมรีเลย์ทำหน้าที่ป้องกันแรงดันย้อนกลับ BackEMF) ถ้ามีการกด สวิตช์ (SW1) ก็จะทำให้อุปกรณ์ไฟฟ้าที่ต่อกับปลั๊กทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 ส่วนของภาคจ่ายไฟ



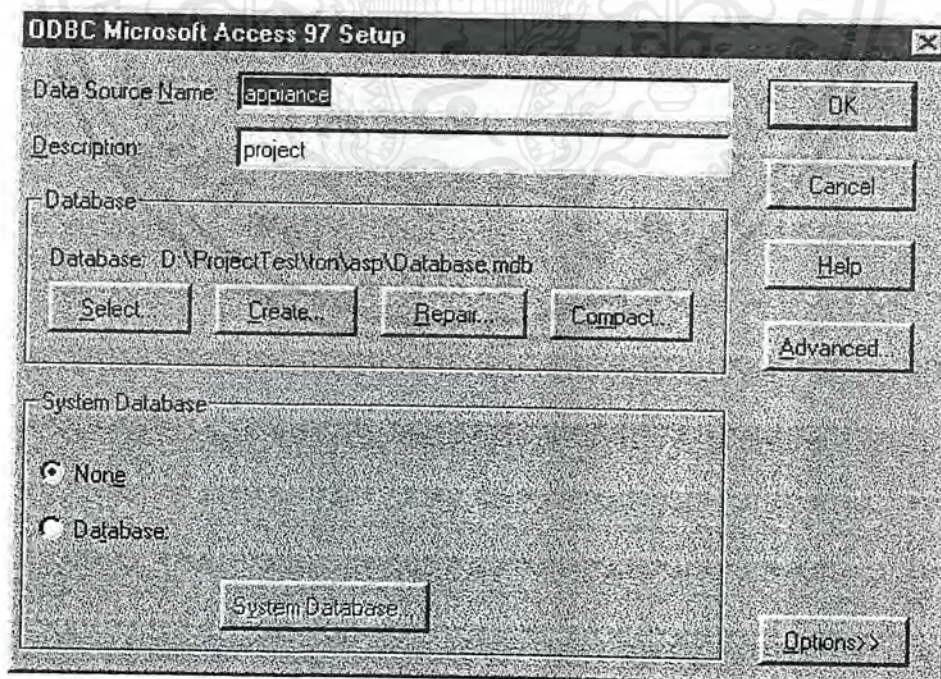
รูปที่ 3.10 แสดงวงจรภาคจ่ายไฟ

### 3.6 ส่วนการกำหนดการเชื่อมโยงฐานข้อมูล ODBC

เป็นส่วนในการเชื่อมโยงและติดต่อกับฐานข้อมูล

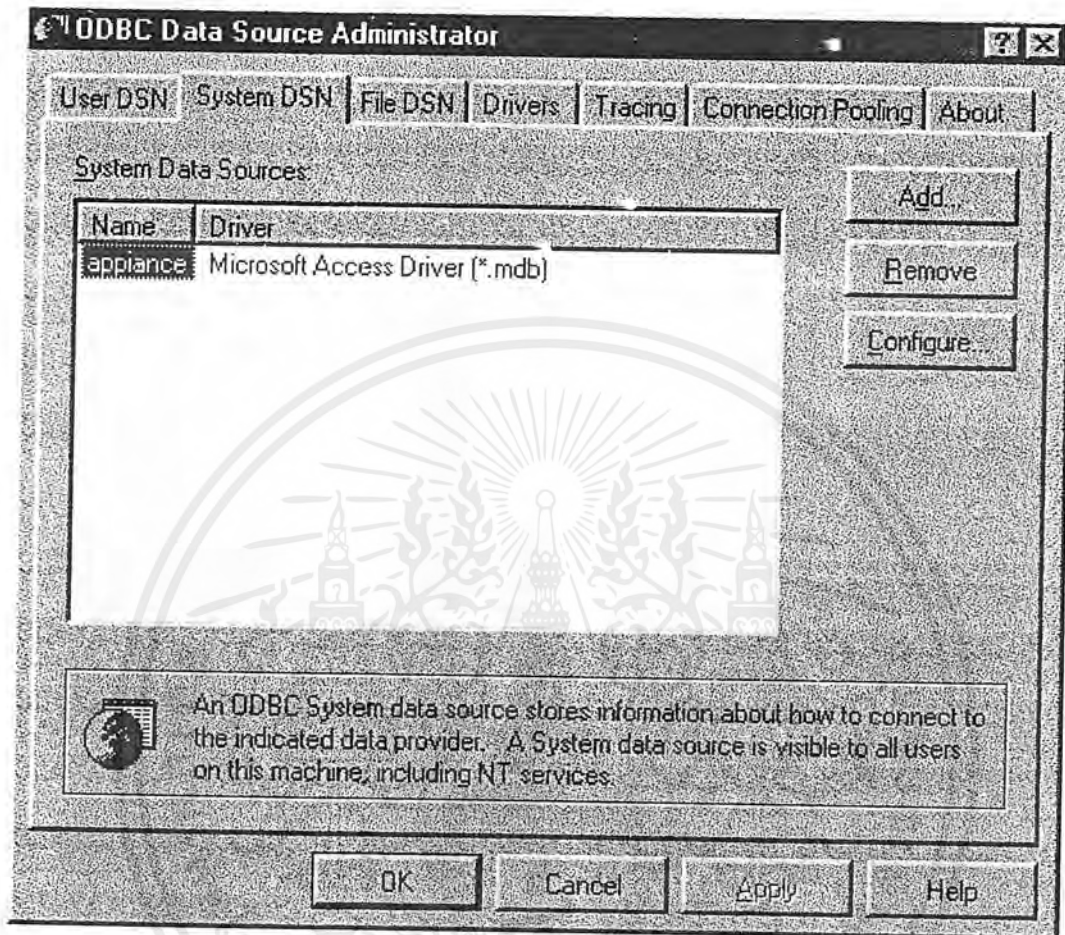
ขั้นตอนการออกแบบ

1. เข้า control panel
2. เข้าไปที่ ODBC32 เพื่อทำการสร้างการเชื่อมโยง
3. ทำการตั้งค่าฐานข้อมูลที่เราต้องการ ดังรูป 3.11



รูปที่ 3.11 แสดงการตั้งค่าฐานข้อมูล ODBC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.12 แสดงการตั้งค่าการเชื่อมโยงเมื่อเสร็จสมบูรณ์

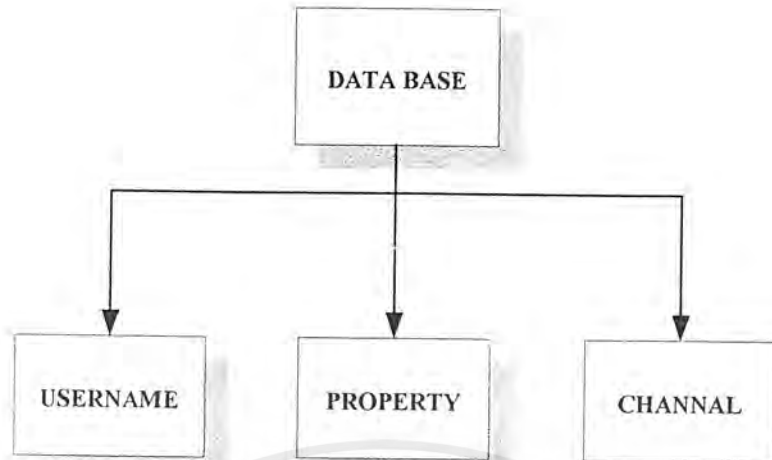
### 3.7 ส่วนการออกแบบฐานข้อมูล

ฐานข้อมูลเป็นส่วนในการเก็บข้อมูลดังนี้

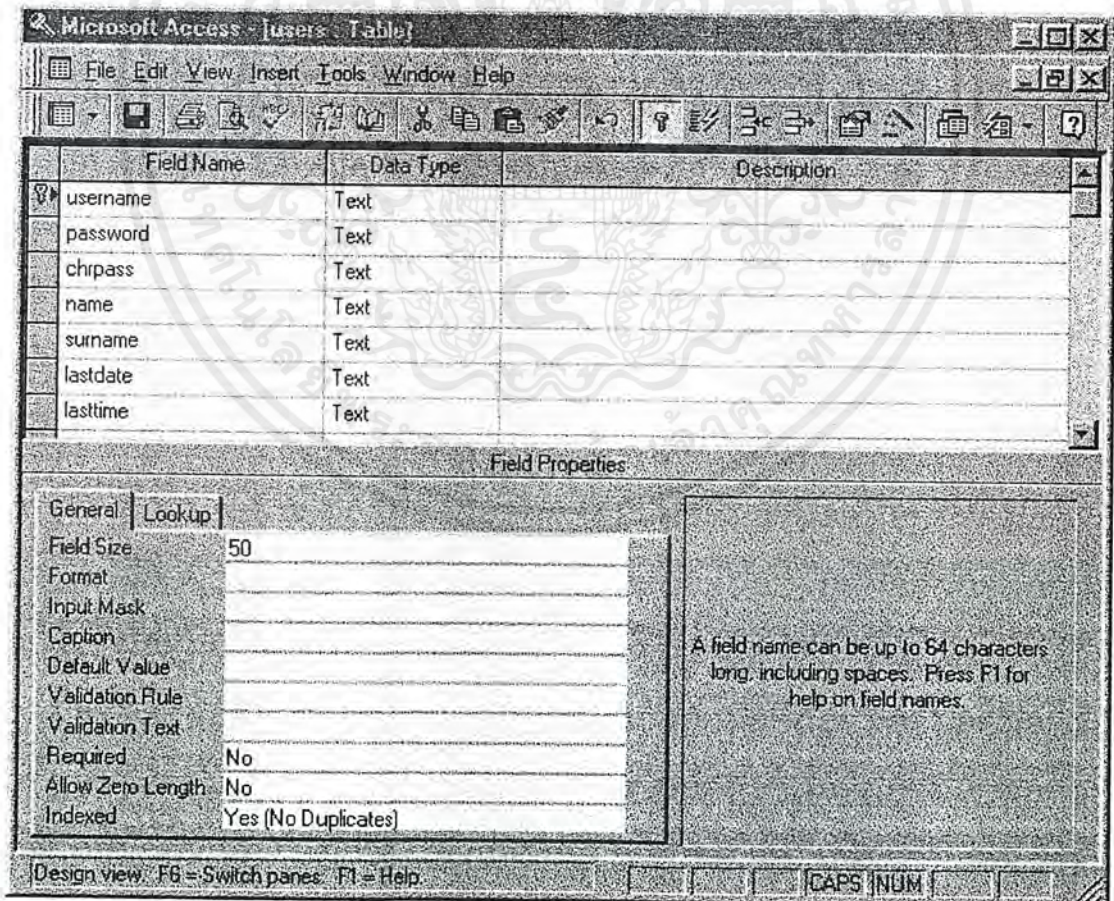
1. สถานะของผู้ใช้งานเอง (ตารางชื่อ USERS)
2. สถานะการเปิด-ปิดของอุปกรณ์ (ตารางชื่อ CHANNEL)
3. สถานะชนิด-ของอุปกรณ์นั้น (ตารางชื่อ PROPERTY)

โดยแบ่งเป็น 3 ตารางเช่นกันดังที่กล่าวมาโดยเราใช้โปรแกรม ACCESS ในการออกแบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

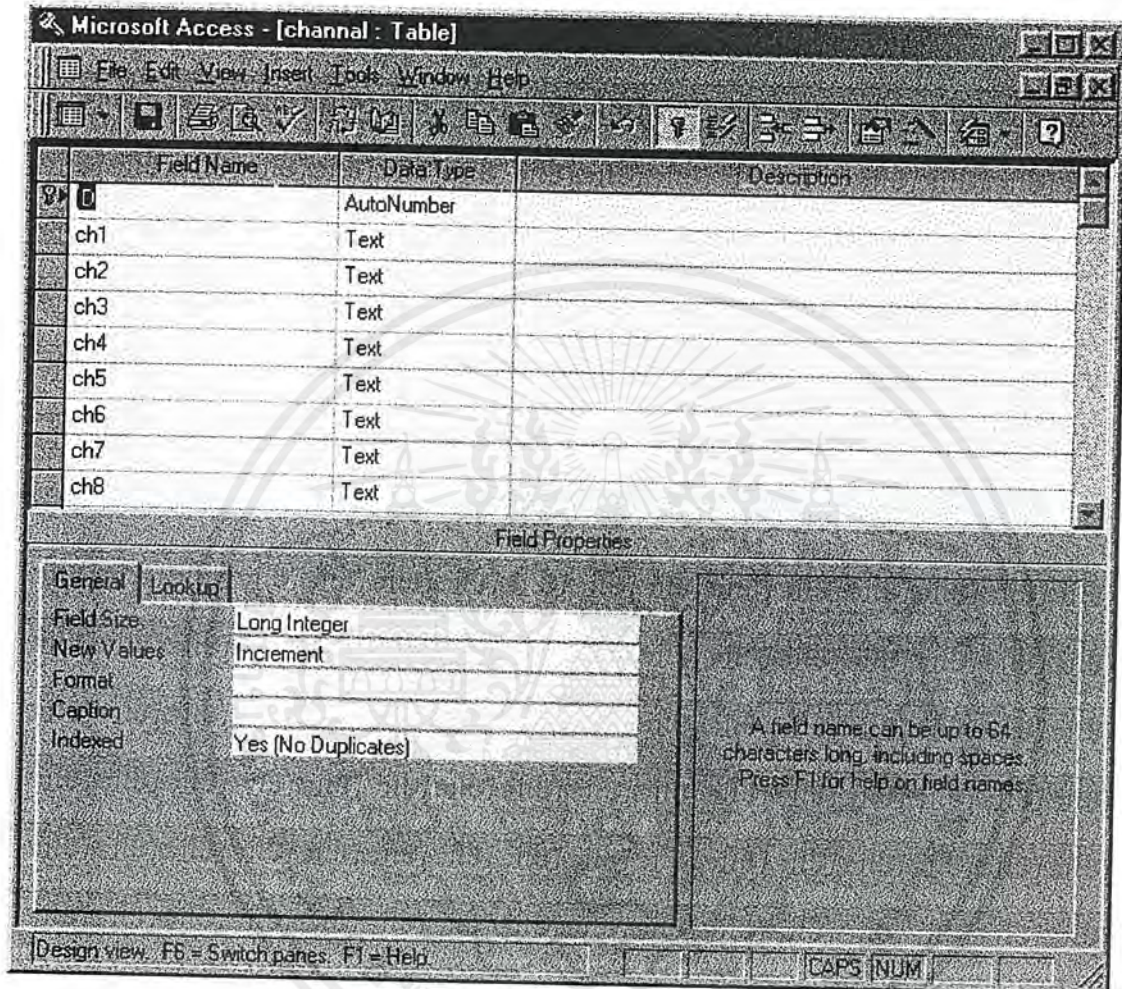


รูปที่ 3.13 แสดงโครงสร้างของฐานข้อมูล



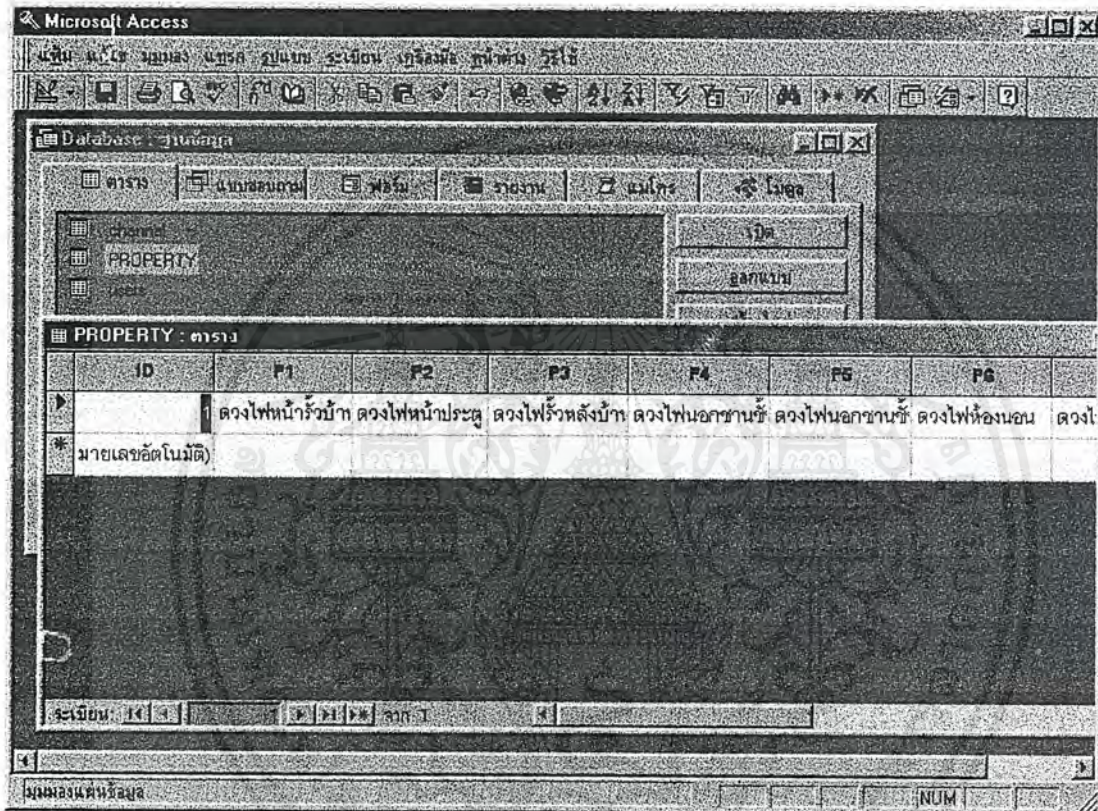
รูปที่ 3.14 แสดงตาราง USER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.15 แสดงตาราง CHANNEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 ตาราง PROPERTY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

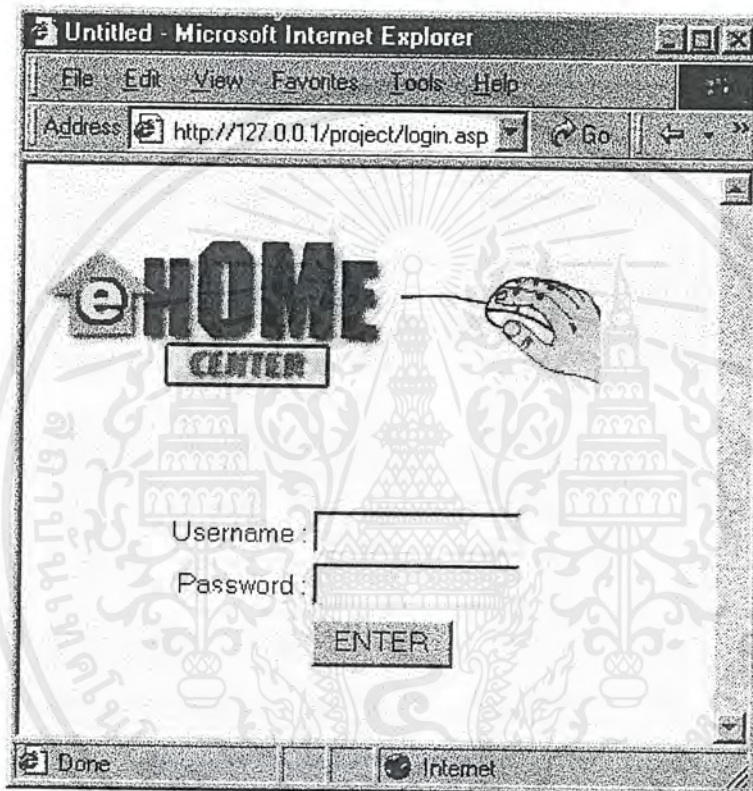
## บทที่ 4

### การทดลองและผลการทดลอง

#### 4.1 ส่วนของโฮมเพจ

##### 4.1.1 หน้าต่าง ล็อกอิน

เมื่อเข้าโฮมเพจหน้าแรกจะเป็นการถาม ล็อกอิน และ พาสเวิร์ด



รูปที่ 4.1 หน้าต่าง ล็อกอิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.2 หน้าต่าง เปลี่ยน รหัสเว็บ

เมื่อต้องการเปลี่ยน รหัสเว็บ สามารถทำได้โดยการคลิกที่ change password ในหน้าต่างสั่งงาน เครื่องใช้ไฟฟ้าจะได้หน้าต่างแสดงผลดังนี้ โดยเรากำหนด รหัสเว็บ ต้องมากกว่า 5 ตัว

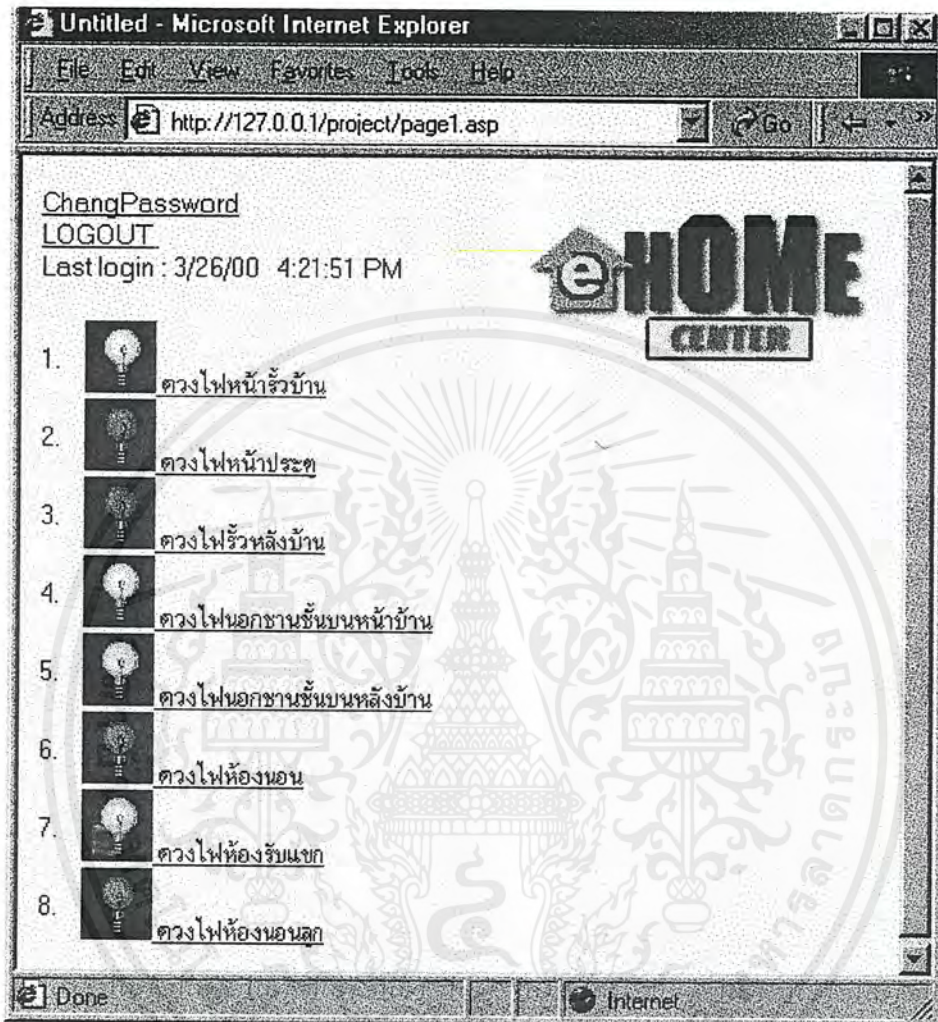
The screenshot shows a Microsoft Internet Explorer window titled "Change Password TCTMAIL - Microsoft Internet Explorer". The address bar displays "http://127.0.0.1/project/editpass.asp". The main content area features the "eHOME CENTER" logo and the heading "CHANGE PASSWORD". Below the heading are four input fields labeled "User Name:", "Old Password:", "New Password:", and "Confirm Password:". A "Change" button is positioned below the "Confirm Password" field. At the bottom of the form, there is a note in Thai: "หมายเหตุ - ความยาวของรหัสผ่านที่ใช้อย่างน้อย 6 ตัวอักษร". The browser's status bar at the bottom shows "Internet".

รูปที่ 4.2 หน้าต่าง เปลี่ยน รหัสเว็บ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.3 หน้าต่างการสั่งงานเครื่องใช้ไฟฟ้า

เมื่อ ล็อกอิน และ พาสเวิร์ด ผ่านแล้ว จะเข้าสู่หน้าต่างหลักคือหน้าต่างสั่งงาน

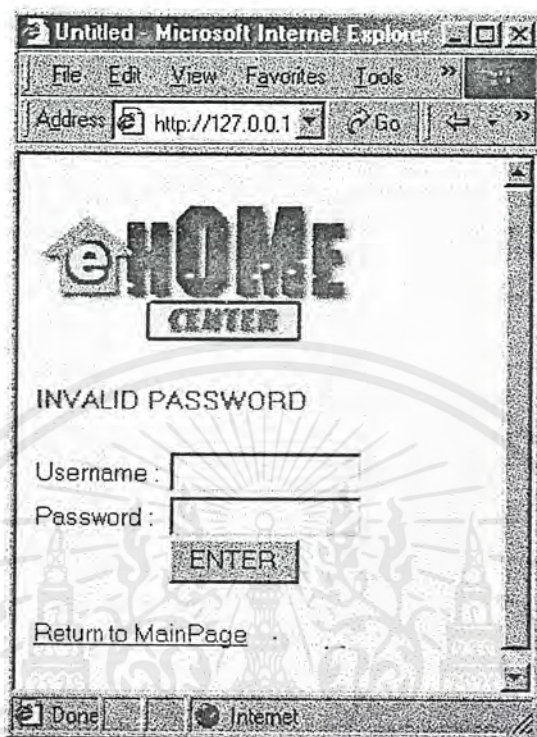


รูปที่ 4.3 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.4 หน้าต่าง พาสเวิร์ด ผิดพลาด

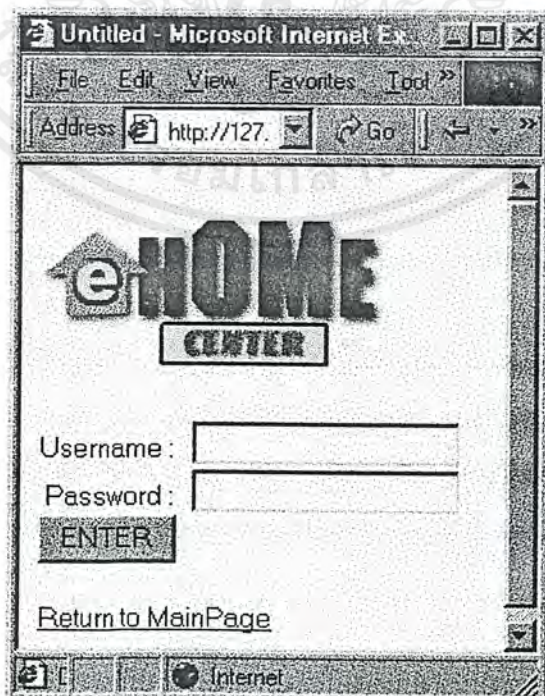
เมื่อ พาสเวิร์ด ผิดพลาด จะเข้าสู่หน้าต่างแสดงผลต่างผิดพลาดดังนี้



รูปที่ 4.4 หน้าต่างเมื่อ พาสเวิร์ด ผิดพลาด

#### 4.1.5 หน้าต่าง ล็อกเอาท์

เมื่อเราต้องการออกจาก โปรแกรมก็จะเข้าสู่หน้าต่าง ล็อกเอาท์



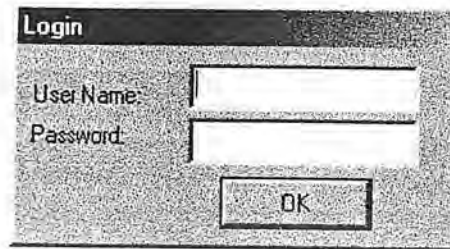
รูปที่ 4.5 หน้าต่าง ล็อกเอาท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 ส่วนของโปรแกรม วิชาพลศึกษา

### 4.2.1 หน้าต่าง พาสเวิร์ด

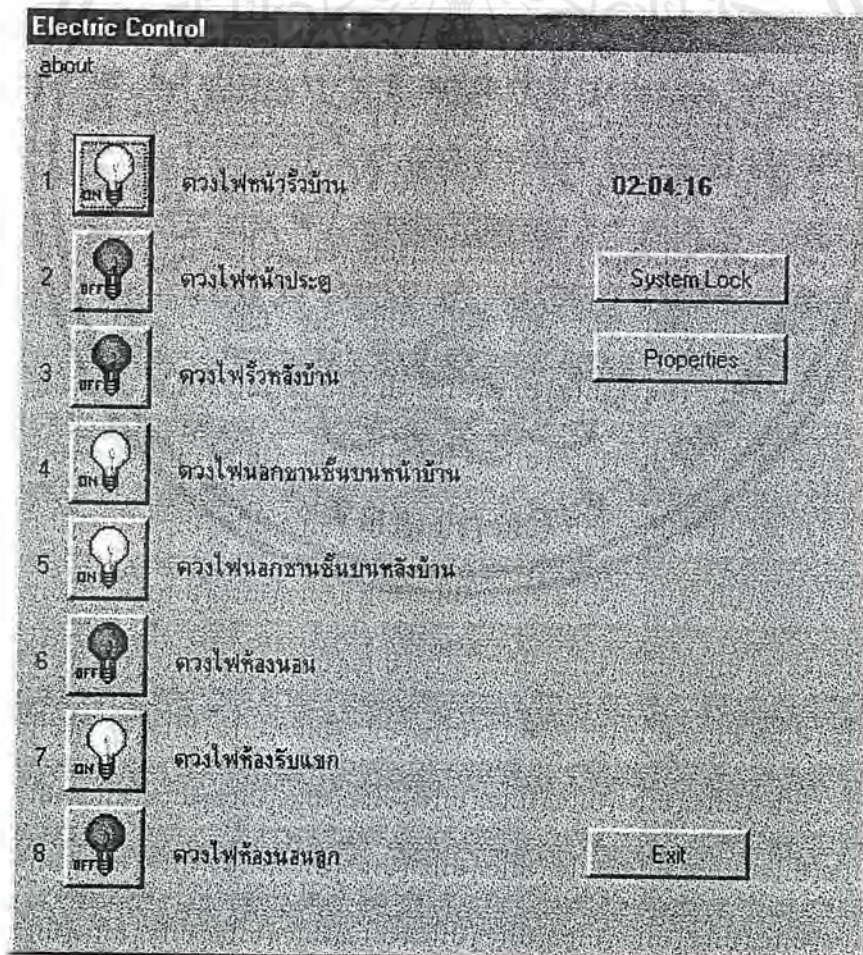
เช่นเดียวกับทางด้าน เอเอสที เราต้องมีการป้อน ล็อกอิน และ พาสเวิร์ด ดังหน้าต่างต่อไปนี้



รูปที่ 4.6 หน้าต่าง พาสเวิร์ด

### 4.2.2 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า

โดยเราสามารถสั่งงานเครื่องใช้ไฟฟ้าได้ 8 ช่อง

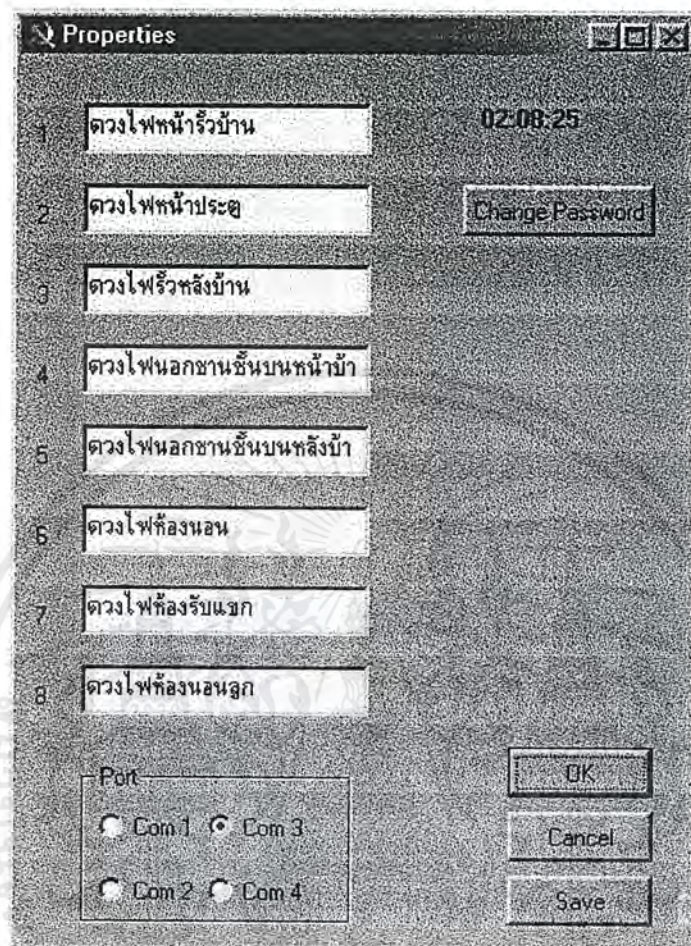


รูปที่ 4.7 หน้าต่างสั่งงานเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2.3 หน้าต่างกำหนดคุณสมบัติ

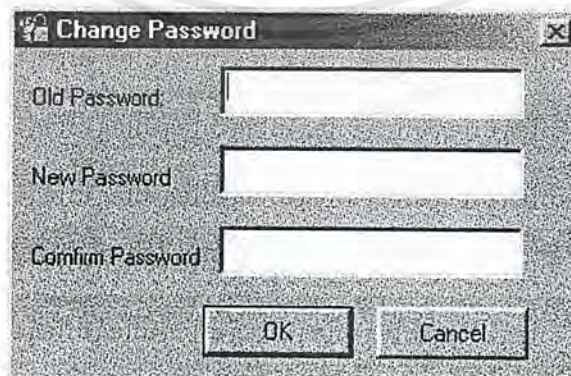
เป็นหน้าต่างกำหนดคุณสมบัติของเครื่องใช้ไฟฟ้า เปลี่ยน พาสเวิร์ด และเลือก คอมพอร์ต



รูปที่ 4.8 หน้าต่างกำหนดคุณสมบัติ

#### 4.2.4 หน้าต่าง เปลี่ยนพาสเวิร์ด

เมื่อเราต้องการเปลี่ยน พาสเวิร์ด ก็คลิกที่ change password จะเข้าสู่หน้าต่างเปลี่ยน พาสเวิร์ด



รูปที่ 4.9 หน้าต่างเปลี่ยนพาสเวิร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทวิจารณ์และบทสรุป

#### 5.1 ปัญหาในการทำโครงการ มีดังต่อไปนี้

- 5.1.1 ส่วนของ เอเอสพี ก่อนข้างจะมีซับซ้อน และมีการดีบั๊ก(Debug)ค่อนข้างยาก เนื่องจากไม่สามารถดีบั๊กทีละบรรทัดได้ จึงทำให้ยากต่อการวิเคราะห์ข้อผิดพลาดที่เกิดขึ้นขณะเขียน โปรแกรม
- 5.1.2 เมื่อเชื่อมส่วนของ โสมเพจกับ เซิร์ฟเวอร์ ก่อนข้างมีปัญหาในการเชื่อมโยงข้อมูล
- 5.1.3 โปรแกรม เซิร์ฟเวอร์ มีข้อจำกัดเมื่อนำมาใช้กับ วิน โคว์ 98 ในเรื่องเสถียรภาพของระบบ

#### 5.2 วิธีการแก้ไขปัญหา

- 5.2.1 เรื่องของ เอเอสพี และ วิซวลเบสิก ควรจะศึกษาควบคู่กันไปเนื่องจากโครงสร้างเป็นภาษาเบสิกเหมือนกันไม่ควรมองข้ามพื้นฐาน โครงสร้างของภาษาเมื่อ โครงสร้างภาษาไม่แน่นจะมีปัญหาในการเขียน โปรแกรม
- 5.2.2 เรื่องความปลอดภัยของระบบควรใช้ระบบที่มีความปลอดภัยสูงเช่น วิน โคว์ เอนที
- 5.2.3 ต้องเข้าหาเพื่อนที่มีความถนัดในด้านนั้นๆ เพื่อที่จะช่วยเหลือเราได้เนื่องจากผู้ที่ผ่านการเขียนมาก่อนย่อมมีประสบการณ์และรู้ปัญหาและวิธีในการแก้ไข

#### 5.3 แนวทางการพัฒนาต่อไป

- 5.3.1 ควรมี การตรวจสอบว่าอุปกรณ์เช่นเซอร์กระแสว่าอุปกรณ์ไฟฟ้าทำงานจริงหรือไม่แล้วผ่านค่ามาบันทึก ที่ฐานข้อมูล (ตาราง channel) ซึ่งได้ออกแบบโครงการรองรับไว้แล้ว
- 5.3.2 ควรมี การเพิ่มส่วนในการบันทึก รายงาน กราฟ และ รายการผลออกทางเครื่องพิมพ์
- 5.3.3 ดัดแปลงโครงการไปประยุกต์ใช้เกี่ยวกับทางด้าน พาณิชย์อิเล็กทรอนิกส์ ( E-Commerce ) ซึ่งเป็นจุดประสงค์หลักของทำโครงการนี้ เนื่องจากจะเป็นส่วนสำคัญมากในอนาคตและเป็นเรื่องใหม่ที่ควรศึกษา

#### 5.4 ข้อเสนอแนะเกี่ยวกับโครงการ

- 5.4.1 นอกจาก เอเอสพี แล้วอาจใช้ CGI, Visual Interdev, Script ต่างๆ ในการเขียน โปรแกรมได้
- 5.4.2 โปรแกรมในการเขียน HTML อาจใช้ Home Side, Dream weaver, Front Page ฯลฯ ช่วยในการสร้าง โสมเพจได้
- 5.4.3 โปรแกรมทำหน้าที่เป็น เซิร์ฟเวอร์ อาจใช้ IIS,PWS,OMNIHTTP ( แต่ OMNIHTTP ไม่สนับสนุน เอเอสพี )
- 5.4.4 ไมโครคอนโทรลเลอร์ควรใช้บอร์ดที่มีหน่วยความจำแบบแฟลช (Flash Memory) ซึ่งสามารถโปรแกรมซ้ำแล้วซ้ำอีกได้หลายครั้งทำให้สะดวกและง่ายต่อการออกแบบโปรแกรม และประหยัด ในงานที่ไม่ใหญ่มากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 5.4.5 อาจใช้ พีเอชพี ( Personal Home Page :PHP ) แทน เอเอสพี ได้เนื่องจากเอเอสพีไม่สามารถทำงานได้กับ เซิร์ฟเวอร์ ที่เป็น ยูนิกซ์ ได้ แต่ พีเอชพี สามารถทำงานได้ทั้งเซิร์ฟเวอร์ที่เป็น วินโดวส์ และ ยูนิกซ์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ก

## ส่วนโปรแกรม ( Source Code )

## ส่วนวิหวลเขลิค

## โปรแกรมหน้าต่างหลัก

```

Option Explicit

Dim ch1, ch2, ch3, ch4, ch5, ch6, ch7, ch8 As Integer
Dim c1 As String
Dim d1, d2, d3, d4, d5, d6, d7, d8 As Integer
Dim Response, Style

Private Sub btnProperties_Click()
    frmMain.Hide
    frmProp.Show
    MSComm1.PortOpen = False
End Sub

Private Sub CmdExit_Click()
    End
End Sub

'*****
'(ABCD) AB = On or Off , CD = CHANNEL
'CH1 ON(0101H) And OFF(0001H)
Private Sub Command1_Click()
    Dim a As Integer
    Data1.Refresh
    a = Val(Text1.Text)
    If a = 0 Then
        Text1.Text = "1"
        Data1.Recordset.Edit
        Data1.Recordset.Update
        Debug.Print a
    Else
        Debug.Print a
        Text1.Text = "0"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    End If
End Sub

' Command2.Picture = LoadPicture(App.Path & "/" & "on.ico")
' CH2 ON(0103H) And OFF(0003H)
Private Sub Command2_Click()
    Dim a As Integer
    Data1.Refresh
    a = Val(Text2.Text)
    If a = 0 Then
        Text2.Text = "1"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    Else
        Text2.Text = "0"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    End If
End Sub

' CH3 ON(0105H) And OFF(0005H)
Private Sub Command3_Click()
    Dim a As Integer
    Data1.Refresh

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

a = Val(Text3.Text)
If a = 0 Then
    Text3.Text = "1"
    Data1.Recordset.Edit
    Data1.Recordset.Update
Else
    Text3.Text = "0"
    Data1.Recordset.Edit
    Data1.Recordset.Update
End If
End Sub

```

```
' CH4 ON(0107H) And OFF(0007H)
```

```
Private Sub Command4_Click()
```

```
    Dim a As Integer
```

```
    Data1.Refresh
```

```
    a = Val(Text4.Text)
```

```
    If a = 0 Then
```

```
        Text4.Text = "1"
```

```
        Data1.Recordset.Edit
```

```
        Data1.Recordset.Update
```

```
    Else
```

```
        Text4.Text = "0"
```

```
        Data1.Recordset.Edit
```

```
        Data1.Recordset.Update
```

```
    End If
```

```
End Sub
```

```
' CH5 ON(0109H) And OFF(0009H)
```

```
Private Sub Command5_Click()
```

```
    Dim a As Integer
```

```
    Data1.Refresh
```

```
    a = Val(Text5.Text)
```

```
    If a = 0 Then
```

```
        Text5.Text = "1"
```

```
        Data1.Recordset.Edit
```

```
        Data1.Recordset.Update
```

```
    Else
```

```
        Text5.Text = "0"
```

```
        Data1.Recordset.Edit
```

```
        Data1.Recordset.Update
```

```
    End If
```

```
End Sub
```

```
' CH6 ON(010BH) And OFF(000BH)
```

```
Private Sub Command6_Click()
```

```
    Dim a As Integer
```

```
    Data1.Refresh
```

```
    a = Val(Text6.Text)
```

```
    If a = 0 Then
```

```
        Text6.Text = "1"
```

```
        Data1.Recordset.Edit
```

```
        Data1.Recordset.Update
```

```
    Else
```

```
        Text6.Text = "0"
```

```
        Data1.Recordset.Edit
```

```
        Data1.Recordset.Update
```

```
    End If
```

```
End Sub
```

```
' CH7 ON(010DH) And OFF(000DH)
```

```
Private Sub Command7_Click()
```

```
    Dim a As Integer
```

```
    Data1.Refresh
```

```
    a = Val(Text7.Text)
```

```
    If a = 0 Then
```

เอกสารนี้เป็นเอกสารต้นฉบับไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        Text7.Text = "1"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    Else
        Text7.Text = "0"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    End If
End Sub

' CH8 ON(010FH) And OFF(000FH)
Private Sub Command8_Click()
    Dim a As Integer
    Data1.Refresh
    a = Val(Text8.Text)
    If a = 0 Then
        Text8.Text = "1"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    Else
        Text8.Text = "0"
        Data1.Recordset.Edit
        Data1.Recordset.Update
    End If
End Sub

Private Sub CmdLock_Click()
    MSComm1.PortOpen = False
    frmMain.Hide
    frmLogin.Show
End Sub

Private Sub Form_Activate()
    Select Case comport
        Case 1
            MSComm1.CommPort = 1
            MSComm1.Settings = "9600,N,8,1"
            MSComm1.InputLen = 10
            MSComm1.PortOpen = True
        Case 2
            MSComm1.CommPort = 2
            MSComm1.Settings = "9600,N,8,1"
            MSComm1.InputLen = 10
            MSComm1.PortOpen = True
        Case 3
            MSComm1.CommPort = 3
            MSComm1.Settings = "9600,N,8,1"
            MSComm1.InputLen = 10
            MSComm1.PortOpen = True
        Case 4
            MSComm1.CommPort = 4
            MSComm1.Settings = "9600,N,8,1"
            MSComm1.InputLen = 10
            MSComm1.PortOpen = True
    End Select
End Sub

Private Sub Form_Load()
    comport = 3
End Sub

Private Sub mnuaboutMy_Click()
    frmAbout.Show
    MSComm1.PortOpen = False
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Timer2_Timer()
'wait 0.1 second to send data
Dim ch1, ch2, ch3, ch4, ch5, ch6, ch7, ch8 As Integer
Data1.Refresh
'วงส่งแต่ละ channel
ch1 = Val(Text1.Text)
If ch1 = 1 And ch1 <> d1 Then
d1 = d1 + 1
Command1.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
MSComm1.Output = Chr(1) & Chr(1) ' Decimal(1) = Hex(01)
Else
If ch1 = 0 And ch1 <> d1 Then
d1 = d1 - 1
Command1.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
MSComm1.Output = Chr(0) & Chr(1) ' Decimal(0) = Hex(00)

Else
End If
End If
'ch2
'.....
ch2 = Val(Text2.Text)
If ch2 = 1 And ch2 <> d2 Then
d2 = d2 + 1
Command2.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
MSComm1.Output = Chr(1) & Chr(3) ' Decimal(1) = Hex(01)
Else
If ch2 = 0 And ch2 <> d2 Then
d2 = d2 - 1
Command2.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
MSComm1.Output = Chr(0) & Chr(3) ' Decimal(0) = Hex(00)
Else
End If
End If
'ch3
'.....
ch3 = Val(Text3.Text)
If ch3 = 1 And ch3 <> d3 Then
d3 = d3 + 1
Command3.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
MSComm1.Output = Chr(1) & Chr(5) ' Decimal(1) = Hex(01)
Else
If ch3 = 0 And ch3 <> d3 Then
d3 = d3 - 1
Command3.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
MSComm1.Output = Chr(0) & Chr(5) ' Decimal(0) = Hex(00)
Else
End If
End If
'ch4
'.....
ch4 = Val(Text4.Text)
If ch4 = 1 And ch4 <> d4 Then
d4 = d4 + 1
Command4.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
MSComm1.Output = Chr(1) & Chr(7) ' Decimal(1) = Hex(01)
Else
If ch4 = 0 And ch4 <> d4 Then

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        d4 = d4 - 1
        Command4.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
        MSComm1.Output = Chr(0) & Chr(7) ' Decimal(0) = Hex(00)
        Else
        End If
    End If
End If
'ch5
'.....
ch5 = Val(Text5.Text)
If ch5 = 1 And ch5 <> d5 Then
    d5 = d5 + 1
    Command5.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
    MSComm1.Output = Chr(1) & Chr(9) ' Decimal(1) = Hex(01)
    Else
    If ch5 = 0 And ch5 <> d5 Then
        d5 = d5 - 1
        Command5.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
        MSComm1.Output = Chr(0) & Chr(9) ' Decimal(0) = Hex(00)
        Else
        End If
    End If
End If
'ch6
'.....
ch6 = Val(Text6.Text)
If ch6 = 1 And ch6 <> d6 Then
    d6 = d6 + 1
    Command6.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
    MSComm1.Output = Chr(1) & Chr(11) ' Decimal(1) = Hex(01)
    Else
    If ch6 = 0 And ch6 <> d6 Then
        d6 = d6 - 1
        Command6.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
        MSComm1.Output = Chr(0) & Chr(11) ' Decimal(0) = Hex(00)
        Else
        End If
    End If
End If
'ch7
'.....
ch7 = Val(Text7.Text)
If ch7 = 1 And ch7 <> d7 Then
    d7 = d7 + 1
    Command7.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
    MSComm1.Output = Chr(1) & Chr(13) ' Decimal(1) = Hex(01)
    Else
    If ch7 = 0 And ch7 <> d7 Then
        d7 = d7 - 1
        Command7.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
        MSComm1.Output = Chr(0) & Chr(13) ' Decimal(0) = Hex(00)
        Else
        End If
    End If
End If
'ch8
'.....
ch8 = Val(Text8.Text)
If ch8 = 1 And ch8 <> d8 Then
    d8 = d8 + 1
    Command8.Picture = LoadPicture(App.Path & "/" & "on.ico")
'ON(0101H)
    MSComm1.Output = Chr(1) & Chr(15) ' Decimal(1) = Hex(01)

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของโรงเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ไปภายนอก

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
If ch8 = 0 And ch8 <> d8 Then
d8 = d8 - 1
Command8.Picture = LoadPicture(App.Path & "/" & "off.ico")
'OFF(0001H)
MSComm1.Output = Chr(0) & Chr(15) ' Decimal(0) = Hex(00)
Else
End If
End If

'close port
'MSComm1.PortOpen = False

End Sub

```

```

Private Sub Timer1_Timer()
Label1.Caption = Format(Now, "hh:mm:ss")
End Sub
'*****
' (ABCD) AB = On or Off , CD = CHANNEL
'Private Sub Timer2_Timer()

'End Sub

```

```

Private Sub Timer3_Timer()
'.....
d1 = Val(Text1.Text)
Text9.Text = Text1.Text
d2 = Val(Text2.Text)
Text10.Text = Text2.Text
d3 = Val(Text3.Text)
Text11.Text = Text3.Text
d4 = Val(Text4.Text)
Text12.Text = Text4.Text
d5 = Val(Text5.Text)
Text13.Text = Text5.Text
d6 = Val(Text6.Text)
Text14.Text = Text6.Text
d7 = Val(Text7.Text)
Text15.Text = Text7.Text
d8 = Val(Text8.Text)
Text16.Text = Text8.Text
Timer3.Enabled = False
End Sub

```

โปรแกรมหน้าต่างพรอพเพอร์ตี้

```

Private Sub Timer1_Timer()
LabelTime.Caption = Format(Now, "hh:mm:ss")
End Sub

```

```

Private Sub changepassword_Click()
frmLogin1.Show
End Sub

```

```

Private Sub cmdCancel_Click()
Me.Hide
frmMain.Show
End Sub

```

```

Private Sub cmdOK_Click()
Me.Hide
frmMain.Show
frmMain.nameCH1.Caption = frmProp.Text1.Text
frmMain.nameCH2.Caption = frmProp.Text2.Text

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของโรงเรียนเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

frmMain.nameCH3.Caption = frmProp.Text3.Text
frmMain.nameCH4.Caption = frmProp.Text4.Text
frmMain.nameCH5.Caption = frmProp.Text5.Text
frmMain.nameCH6.Caption = frmProp.Text6.Text
frmMain.nameCH7.Caption = frmProp.Text7.Text
frmMain.nameCH8.Caption = frmProp.Text8.Text
End Sub

```

```

Private Sub cmdSave_Click()
    Data1.Recordset.Edit
    Data1.Recordset.Update
End Sub

```

```

Public Sub Option1_Click()
    If Option1.Value = True Then
        comport = 1
    End If
End Sub

```

```

Public Sub Option2_Click()
    If Option1.Value = True Then
        comport = 2
    End If
End Sub

```

```

Public Sub Option3_Click()
    If Option1.Value = True Then
        comport = 3
    End If
End Sub

```

```

Public Sub Option4_Click()
    If Option1.Value = True Then
        comport = 4
    End If
End Sub

```

โปรแกรมหน้าต่างล็อก-อิน

Option Explicit

Public LoginSucceeded As Boolean

```

Private Sub cmdCancel_Click()
    'set the global var to false
    'to denote a failed login
    LoginSucceeded = False
    Me.Hide
End Sub

```

```

Private Sub cmdOK_Click()
    'check for correct password
    If Text3.Text = txtUserName.Text And Text4.Text =
txtPassword.Text Then

```

```

        LoginSucceeded = True
        Me.Hide
        frmMain.Show
    Else
        MsgBox "The Password is incorrect, Please try again!",
vbCritical, "Invalid Password"
        txtPassword.SetFocus
        SendKeys "{Home}+{End}"
    End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Form_Activate()
    txtPassword.Text = ""
    txtUserName.Text = ""
End Sub

```

โปรแกรมหน้าต่างเปลี่ยนพาสเวิร์ด

```
Option Explicit
```

```
Public LoginSucceeded As Boolean
```

```

Private Sub cmdCancel_Click()
    'set the global var to false
    'to denote a failed login
    LoginSucceeded = False
    Me.Hide
End Sub

```

```

Private Sub cmdOK_Click()
    'check for correct password
    If Text5.Text = txtOldPasswd Then
        If txtNewPasswd.Text = txtComfirmPasswd.Text Then
            If (Len(txtNewPasswd.Text) And
                Len(txtComfirmPasswd.Text)) > 5 Then
                Text5.Text = txtNewPasswd.Text
                MsgBox "Password change Complete", vbOKOnly
                Data1.Refresh
                LoginSucceeded = True
                Me.Hide

            Else
                MsgBox "The Character password must > 5", vbExclamation,
                    "Change Password"
                txtNewPasswd.SetFocus
                txtComfirmPasswd.SetFocus
                SendKeys "{Home}+{End}"
                txtNewPasswd.SetFocus
                txtComfirmPasswd.SetFocus

            End If
        Else
            MsgBox "The password you typed do not match",
                vbExclamation, "Change Password"
            txtNewPasswd.SetFocus
            txtComfirmPasswd.SetFocus
            SendKeys "{Home}+{End}"
            txtNewPasswd.SetFocus
            txtComfirmPasswd.SetFocus

        End If
    Else
        MsgBox "The password you typed do not match",
            vbExclamation, "Change Password"
        txtOldPasswd.SetFocus
        SendKeys "{Home}+{End}"
        txtOldPasswd.SetFocus

    End If
End Sub

```

```

Private Sub Form_Activate()
    txtNewPasswd.Text = ""
    txtComfirmPasswd.Text = ""
    txtOldPasswd.Text = ""
End Sub

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ส่วนไมโครคอนโทรลเลอร์

## โปรแกรมหลัก

```

RS      EQU  P2.7
R_W    EQU  P2.6
ENABLE EQU  p2.5

ORG     0000H
AJMP   START

ORG     0023H
AJMP   MAIN

START:  ORG     0100H
        MOV     SP,#5FH
        MOV     IE,#10010000B ;ENABLE /INT0
        MOV     IP,#00010000B ;SET HIGH PRIORITY
        MOV     TMOD,#00101100B ;SET SERIAL PORT
        MOV     SCON,#01010000B
        MOV     PCON,#0
        MOV     TH1,#253
        SETB    TR1
;===== INIT =====
        MOV     A,#38H ;8bit, 2 line, 5x7 dot
        ACALL   W_INSC
        MOV     A,#0CH
        ACALL   W_INSC
        MOV     A,#6 ;increment cursor
        ACALL   W_INSC
        MOV     A,#1 ;clear and home
        ACALL   W_INSC
        CLR     ENABLE
;=====SHOW START LCD=====
STR1:   MOV     R0,#16
        MOV     DPTR,#TAB1
        CLR     A
        MOVC    A,@A+DPTR
        ACALL   W_DATA
        INC     DPTR
        DJNZ    R0,STR1
        MOV     A,#0C0H ;New line
        ACALL   W_INSC
        MOV     R0,#16
        MOV     DPTR,#TAB2
STR2:   CLR     A
        MOVC    A,@A+DPTR
        ACALL   W_DATA
        INC     DPTR
        DJNZ    R0,STR2
        SJMP    $ ;Wait For Interrupt
;=====
MAIN:   CLR     RI
        MOV     A,SBUF
        MOV     20H,A
        JNB    RI,$
        CLR    RI
        MOV     A,SBUF
;CH1=====
        CJNE   A,#01H,CH2
        JB     00,OPEN1
        SETB   P0.0
        ACALL  INTER
        ADD    A,#0C0H
        ACALL  LCDCLOSE
        RETI

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;=====
OPEN1:   CLR     P0.0
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN2:   CLR     P0.1
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN3:   CLR     P0.2
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN4:   CLR     P0.3
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN5:   CLR     P0.4
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN6:   CLR     P0.5
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN7:   CLR     P0.6
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
OPEN8:   CLR     P0.7
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDOPEN
         RETI

;=====
CH2:    CJNE   A,#03H,CH3
         JB    00,OPEN2
         SETB  P0.1
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDCLOSE
         RETI

;=====
CH3:    CJNE   A,#05H,CH4
         JB    00,OPEN3
         SETB  P0.2
         ACALL  INTER
         ADD    A,#0C0H
         ACALL  LCDCLOSE
         RETI

;=====

```

เอกสารนี้เป็นเอกสารที่กรมส่งเสริมการค้าระหว่างประเทศจัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        JB      00,OPEN4
        SETB   P0.3
        ACALL  INTER
        ADD    A,#0C0H
        ACALL  LCDCLOSE
        RETI
;=====
CH5:    CJNE   A,#09H,CH6
        JB     00,OPEN5
        SETB   P0.4
        ACALL  INTER
        ADD    A,#0C0H
        ACALL  LCDCLOSE
        RETI
;=====
CH6:    CJNE   A,#0BH,CH7
        JB     00,OPEN6
        SETB   P0.5
        ACALL  INTER
        ADD    A,#0C0H
        ACALL  LCDCLOSE
        RETI
;=====
CH7:    CJNE   A,#0DH,CH8
        JB     00,OPEN7
        SETB   P0.6
        ACALL  INTER
        ADD    A,#0C0H
        ACALL  LCDCLOSE
        RETI
;=====
CH8:    JB     00,OPEN8
        SETB   P0.7
        ACALL  INTER
        ADD    A,#0C0H
        ACALL  LCDCLOSE
        RETI
;=====
;INTERRUPT0 FOR 2051 (P3.3)
INTER:  CLR    P3.2
        NOP
        NOP
        NOP
        NOP
        NOP
        SETB   P3.2
        RET
;=====
LCDOPEN: ACALL  W_INSC
        MOV    A,#"•"
        ACALL  W_DATA
        RET
;=====
LCDCLOSE: ACALL  W_INSC
        MOV    A,#"•"
        ACALL  W_DATA
        RET
;=====
W_INSC:  ACALL  WAIT
        MOV    P1,A      ;move data command at port 1
        CLR    RS        ;set low RS
        CLR    R_W       ;set low R_W
        SETB   ENABLE    ;set high Enable
        CLR    ENABLE    ;set low Enable again for read data
        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

W_DATA:   ACALL   WAIT
          MOV     P1,A
          SETB   RS
          CLR    R_W
          SETB   ENABLE
          CLR    ENABLE
          RET

WAIT:     MOV     P1,#0FFH
          CLR    RS
          SETB   R_W
          SETB   ENABLE
          JB     P1.7,$
          CLR    ENABLE
          RET

TAB1:    DB     " 1 2 3 4 5 6 7 8"
TAB2:    DB     " . . . . . . . ."

          END

```

## โปรแกรมส่ง

```

ORG      0000H
AJMP     START

ORG      0003H
AJMP     READ

START:   ORG      0100H
          MOV     IE,#10000001B
          MOV     TMOD,#00101100B
          MOV     PCON,#00H
          MOV     SCON,#01000000B
          MOV     TH1,#253
          SETB   TR1
READ:    MOV     A,P1
          MOV     SBUF,A
          JNB   TI,$
          CLR    TI
          RETI
          END

```

## โปรแกรมรับ

```

ORG      0000H
AJMP     START

ORG      0023H
AJMP     READ

START:   ORG      0100H
          MOV     IE,#10010000B
          MOV     TMOD,#00101100B
          MOV     PCON,#00H
          MOV     SCON,#01010000B
          MOV     TH1,#253
          SETB   TR1
          SJMP   $
READ:    MOV     A,SBUF
          CLR    RI
          MOV     P1,A
          RETI
          END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ส่วนของ HTML และ เอเอสซี

โปรแกรมเชื่อมโยงหน้าต่างล็อก-อิน (default.html)

```
<HTML>
<HEAD>
  <TITLE>home page</TITLE>
</HEAD>
<META HTTP-EQUIV="Refresh" CONTENT="5; URL=login.asp">
</HTML>
```

โปรแกรมหน้าต่างล็อก-อิน (login.asp, verify.asp)

login.asp

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
  <title>login</title>
</head>
<body bgcolor="#FFFFFF">
<!--use method "post" to sent variable to verify.asp-->
<p><a href=" ../default.asp"></a><object classid="clsid:D27CDB6E-AE6D-11cf-
96B8-444553540000"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swfla
sh.cab#version=4,0,2,0" width="100" height="100">
  <param name=movie value=" ../flash/test.swf">
  <param name=quality value=high>
  <embed src=" ../flash/test.swf" quality=high
pluginspage="http://www.macromedia.com/shockwave/download/index.cgi?
P1_Prod_Version=ShockwaveFlash" type="application/x-shockwave-flash"
width="100" height="100">
  </embed>
</object> </p>
<form action="verify.asp" method="post">
  <table bgcolor="#FFFFFF" cellspacing="0" cellpadding="0" border="0"
frame="box" rules="all" style = "HEIGHT: 120px; WIDTH: 300px"
width="431" align="left" height="158">
  <tr>
    <td colspan="2" align="left" valign="middle" height="31"><font
color=#ff0000 face=AngsanaUPC size=6>
  </font>
  <tr>
    <td width="135" align="left" valign="middle" height="2">
      <div align="right">Username : </div>
    <td width="159" height="2">
      <input
        name=uid
        maxlength="10"
        value=""
        size="15">
  <tr>
    <td width="135" align="left" valign="middle" height="2">
      <div align="right">Password : </div>
    <td width="159" height="2">
      <input maxlength=10 name=pwd
        type=password value="" size="15">
  <tr>
    <td align="left" valign="middle" height="30">
      <div align="center"><!--chang passwd to call pageeditpass.asp-->
</div>
  <td>
  <td align="left" valign="middle" height="30">
    <input type="submit" value="ENTER" name="submit">
  </td>
</tr>
</table>
</form>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<div align="center"></div>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p>&nbsp;</p>
<p><a href=" ../default.asp">กลับสู่หน้าหลัก</a></p>
<p>&nbsp;</p>
</body>
</html>

```

### verify.asp

```

<%
    'define check=session variable
    'if check=0 mean you come by interrupt
        Session("Check") = "0"
    'receive uid from login.asp
        User = Request("uid")
    'replace " " by A
        User = replace(User, " ", "A")
    'receive pwd from login.asp
        Password = Request("pwd")
    'show the length of password
        countpwd=Len(Password)
        MidString = Password
    'check password
        For i=1 To countpwd
    'change to ascii code
            chang_Asc = chang_Asc & Asc(Mid(MidString,i,1))
        Next
    'create odbc name =appliance
        Set dbmail = server.CreateObject("ADODB.Connection")
        dbmail.Open "appliance"
        Set RS = dbmail.Execute("SELECT * FROM users
where username = '" & User & "' and Password = '" & chang_Asc & "'")

        On Error Resume Next
        rs.movefirst
        Do While Not RS.EOF
            Session("Check") = "1"
            Session("lastDate") = RS("lastdate")
            Session("lastTime") = RS("lasttime")
            rs.movenext
        loop
        rs.close
        dbmail.close

    if Session("Check") = "1" then
        Set newlogin = Server.CreateObject("ADODB.Recordset")
        newlogin.Open "SELECT * FROM users where username = '" &
User & "' and Password = '" & chang_Asc
& "'", "DATABASE=Database;DSN=appliance", 1, 3
        newlogin("LastTime") = time
        newlogin("LastDate") = date
        newlogin.Update
        Response.Redirect "page1.asp"
    else
        Response.Redirect "loginfalse.asp"
    end if
%>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

else
    response.Write "<META HTTP-EQUIV=""Refresh"" CONTENT=""0";
URL=login.asp"">"
end if

%>
</form>
<table width="100%" border="0" cellspacing="0" cellpadding="0">
<tr>
<td width="11%">1.</td>
<td width="89%">
<%
    i=0
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
<tr>
<td width="11%">2.</td>
<td width="89%">
<%
    i=1
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
<tr>
<td width="11%">3.</td>
<td width="89%">
<%
    i=2
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
<tr>
<td width="11%">4.</td>
<td width="89%"><%
    i=3
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%>
</td>
</tr>
<tr>
<td width="11%">5.</td>
<td width="89%">
<%
    i=4
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
<tr>
<td width="11%">6.</td>
<td width="89%">
<%
    i=5
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
<tr>
<td width="11%">7.</td>
<td width="89%">
<%
    i=6
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
<tr>
<td width="11%">8.</td>
<td width="89%">
<%
    i=7
    if ch(i) = "0" then
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=1> <img src=""off.jpg""> " & property( i)&" </a>"
    else
        Response.Write "<a href=page2.asp?strMsgNo=" & i + 1 &
"&rndno=0> <img src=""on.jpg""> "& property( i)&" </a>"
    end if
%>
</td>
</tr>
</table>
</td>
<td width="55%" height="343" valign="top">

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการใช้งานเพื่อการค้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    <div align="right">
      <p></p>
      <p align="center"><font color="#FF3333"><b>ข้อเสนอแนะ</b></font>
</p>
      <p align="center"><font color="#0033CC">1.หลังจากที่คลิก เปิดหรือปิด
แล้วควรรอให้เกิดการเปลี่ยนแปลงก่อนแล้วค่อยคลิกอันต่อไป</font></p>
      <p align="center"><font color="#0033CC">2.หลังจากที่เสร็จสิ้นการใช้งาน
</font><b>
        <font color="#FF0000">ต้องคลิก LOGOUT</font> </b><font
color="#0033CC">เพื่อที่จะออกไปหน้าหลักของเว็บไซต์</font></p>
      <p align="center">&nbsp;</p>
    </div>
  </td>
</tr>
</table>
<p><font color="#0033CC">อย่าลืมคลิกที่ LOGOUT เมื่อต้องการออกจากส่วนการควบคุมไฟฟ้า
</font></p>
</body>
</html>

```

โปรแกรมหน้าต่างสั่งงานเครื่องใช้ไฟฟ้า (page2.asp)

```

<%
channel = Request.QueryString("strMsgNo")
valueSW = Request.QueryString("rndno")
on error resume next
if cint(channel) > 8 or cint(channel) < 1 then
  Response.Redirect "login.asp"
end if
if cint(valueSW) > 1 or cint(valueSW) < 0 then
  Response.Redirect "login.asp"
end if
if Session("Check") = "1" then
  chan = "ch" & channel
  Session(chan) = valueSW
  Set new1 = Server.CreateObject("ADODB.Recordset")
  new1.Open "SELECT * FROM channel
", "DATABASE=Database;DSN=appliance", 1, 3
  new1(chan) = valueSW
  new1.Update
  Response.Redirect "page1.asp"
else
  Response.Write "<META HTTP-EQUIV=" "Refresh" " CONTENT=" "0;
URL=loginfalse.asp">"
end if
%>

```

โปรแกรมหน้าต่างเปลี่ยนพาสเวิร์ด (editpass.asp, changepassword.asp)

editpass.asp

```

<html>
<head>
  <title>Change Password TCTMAIL</title>
</head>
<body leftmargin="0" topmargin="0" bgcolor="#FFFFFF">
<FORM action=changepassword.asp method=post>
  <center>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





```

MidString = newpass
For i=1 To countpwd
    newAsc=newAsc & Asc(Mid(MidString,i, 1 ))
Next
Set rsmail = Server.CreateObject

("ADODB.Recordset")
rsmail.Open "SELECT * FROM users where
username = '& user &'" and Password = '& chang_Asc
& " "', "DATABASE=Database;DSN=appiance",1,3
rsmail("password") = newAsc
rsmail("chrpass") = newpass
rsmail.Update
response.write "<TR ALIGN=CENTER
VALIGN=CENTER>"
response.write "<TD BGCOLOR=""#00A4A4""
height=25><B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>การเปลี่ยนแปลงข้อมูลของคุณ
เสร็จเรียบร้อยแล้วครับ</FONT></B></TD> </TR>"
response.write "</table><BR><a
href=pagel.asp>BACK</CENTER></a>"
else
response.write "<TR ALIGN=CENTER
VALIGN=CENTER>"
response.write "<TD BGCOLOR=""#FC0303""
height=25><B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณป้อน Confirm Password ไม่
ตรงกับ New Password กรุณาแก้ไขใหม่</FONT></B></TD> </TR>"
response.write "</table><BR><a
href=pagel.asp>BACK</CENTER></a>"
end if
else
Session.abandon
response.write "<TR ALIGN=CENTER VALIGN=CENTER>"
response.write "<TD BGCOLOR=""#FC0303"" height=25>
<B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณไม่ได้ข้อมูล ในช่อง NEW PASSWORD และ
ในช่อง CONFIRM NEW PASSWORD</FONT></B></TD> </TR>"
response.write "</table><BR><a href=login.asp>BACK
</CENTER></a>"
end if
else
Session.abandon
response.write "<TR ALIGN=CENTER VALIGN=CENTER>"
response.write "<TD BGCOLOR=""#FC0303"" height=25>
<B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณได้ NEW PASSWORD น้อยกว่า 6 ตัวอักษร
</FONT></B></TD> </TR>"
response.write "</table><BR><a href=login.asp>BACK
</CENTER></a>"
end if
else
Session.abandon
response.write "<TR ALIGN=CENTER VALIGN=CENTER>"
response.write "<TD BGCOLOR=""#FC0303"" height=25>
<B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณไม่ได้ข้อมูลในช่อง OLD PASSWORD หรือคุณ
ใส่ข้อมูลผิด</FONT></B></TD> </TR>"
response.write "</table><BR><a href=login.asp>BACK
</a></CENTER>"
end if
else
Session.abandon

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Response.Write "<META HTTP-EQUIV=""Refresh"" CONTENT=""0;
URL=loginfalse.asp"">"
end if

%>
</div>
<p>&nbsp;&nbsp;&nbsp;</p>
</FORM>
</BODY>
</HTML>

```

โปรแกรมหน้าต่างแสดงความผิดพลาด(loginfalse.asp)และสิ้นสุดการทำงาน(changepassword.asp)  
loginfalse.asp

```

<html>
<head>
  <title>Untitled</title>
</head>

<body bgcolor="#FFFFFF">
<form action="verify.asp" method="post">
  <p><a href=" ../index.html"></a></p>
  <p><b><font color="#FF3333">INVALID PASSWORD</font></b></p>
  <table width="30%" border="0" cellspacing="0" cellpadding="0">
    <tr>
      <td width="34%">Username : &nbsp;&nbsp;&nbsp;</td>
      <td width="66%">
        <input name="uid" maxlength="10" size="15" >
      </td>
    </tr>
    <tr>
      <td width="34%">Password :&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</td>
      <td width="66%">
        <input type="password" name="pwd" maxlength="10" size="15">
      </td>
    </tr>
    <tr>
      <td width="34%">&nbsp;&nbsp;&nbsp;</td>
      <td width="66%">
        <input type="submit" value="ENTER" name="submit">
      </td>
    </tr>
  </table>
  <p><a href=" ../ ../index.html">Return to MainPage</a><br><br>
</p>
</form>
</body>
</html>
changepassword.asp

```

```

<HTML>
<HEAD>
<TITLE>CHANG PASSWORD</TITLE>

</HEAD>
<body leftmargin="0" topmargin="0" bgcolor="#FFFFFF">
  <div align="center"><FORM > <CENTER><TABLE BORDER=1 COLS=1 WIDTH="550"
cellspacing="0" cellpadding="0">
<%

```

```

  if Session("Check") = "1" then
    user = Request("uid")
    oldpass = Request("oldpassword")
    newpass = Request("newpassword")

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยอาจารย์ผู้สอน ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

confirm = Request("confirmpass")
countpwd=Len(oldpass)
MidString = oldpass
For i =1 To countpwd
    chang_Asc = chang_Asc & Asc(Mid(MidString,i, 1 ))
Next
chkpass = "0"
Set dbmail = Server.CreateObject("ADODB.Connection")
dbmail.Open "appiance"
Set RS = dbmail.Execute("SELECT * FROM users where username
= '& user &' and Password = '& chang_Asc &'")

On Error Resume Next
rs.movefirst
Do While Not RS.EOF
    chkpass = "1"
    rs.movenext
loop
rs.close
dbmail.close
if chkpass = "1" then
    if len(newpass) > 5 and len(confirm) > 5 then
        if newpass <> "" and confirm <> "" then
            if newpass = confirm then
                countpwd=Len(newpass)
                MidString = newpass
                For i=1 To countpwd
                    newAsc = newAsc & Asc(Mid(MidString,i, 1 ))
                Next
                Set rsmail = Server.CreateObject
("ADODB.Recordset")
                rsmail.Open "SELECT * FROM users where
username = '& user &' and Password = '& chang_Asc
&'", "DATABASE=Database;DSN=appiance",1,3
                rsmail("password") = newAsc
                rsmail("chrpass") = newpass
                rsmail.Update
                response.write "<TR ALIGN=CENTER
VALIGN=CENTER>"
                response.write "<TD BGCOLOR=""#00A4A4""
height=25><B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>การเปลี่ยนแปลงข้อมูลของคุณ
เสร็จเรียบร้อยแล้วครับ</FONT></B></TD> </TR>"
                response.write "</table><BR><a href=pagel.asp>
BACK</CENTER></a>"
            else
                response.write "<TR ALIGN=CENTER
VALIGN=CENTER>"
                response.write "<TD BGCOLOR=""#FC0303""
height=25><B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณป้อน Confirm Password ไม่
ตรงกับ New Password กรุณาแก้ไขใหม่</FONT></B></TD> </TR>"
                response.write "</table><BR><a href=pagel.asp>
BACK</CENTER></a>"
            end if
        else
            Session.abandon
            response.write "<TR ALIGN=CENTER VALIGN=CENTER>"
            response.write "<TD BGCOLOR=""#FC0303"" height=25>
<B><FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณไม่ได้ข้อมูล ในช่อง NEW PASSWORD และ
ในช่อง CONFIRM NEW PASSWORD</FONT></B></TD> </TR>"
            response.write "</table><BR><a href=login.asp>BACK
</CENTER></a>"

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        end if
    else
        Session.abandon
        response.write "<TR ALIGN=CENTER VALIGN=CENTER>"
        response.write "<TD BGCOLOR=""#FC0303"" height=25><B>
<FONT COLOR=""#FFFFFF"" FONT SIZE=-1>คุณใส่ NEW PASSWORD น้อยกว่า 6 ตัวอักษร
</FONT></B></TD> </TR>"
        response.write "</table><BR><a href=login.asp>BACK
</CENTER></a>"
    end if
    else
        Session.abandon
        response.write "<TR ALIGN=CENTER VALIGN=CENTER>"
        response.write "<TD BGCOLOR=""#FC0303"" height=25><B><FONT
COLOR=""#FFFFFF"" FONT SIZE=-1>คุณไม่ได้ข้อมูลในช่องของ OLD PASSWORD หรือคุณใส่ข้อมูลผิด
</FONT></B></TD> </TR>"
        response.write "</table><BR><a href=login.asp>BACK</a>
</CENTER>"
    end if
    else
        Session.abandon
        <META HTTP-EQUIV="Refresh" CONTENT="0; URL=login.
        Response.Write "<META HTTP-EQUIV=""Refresh"" CONTENT=""0;
        URL=loginfalse.asp"">"
    end if
%>
</div>
<p>&nbsp;&nbsp;&nbsp;</p>
</FORM>
</BODY>
</HTML>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข  
หนังสือคู่มือ

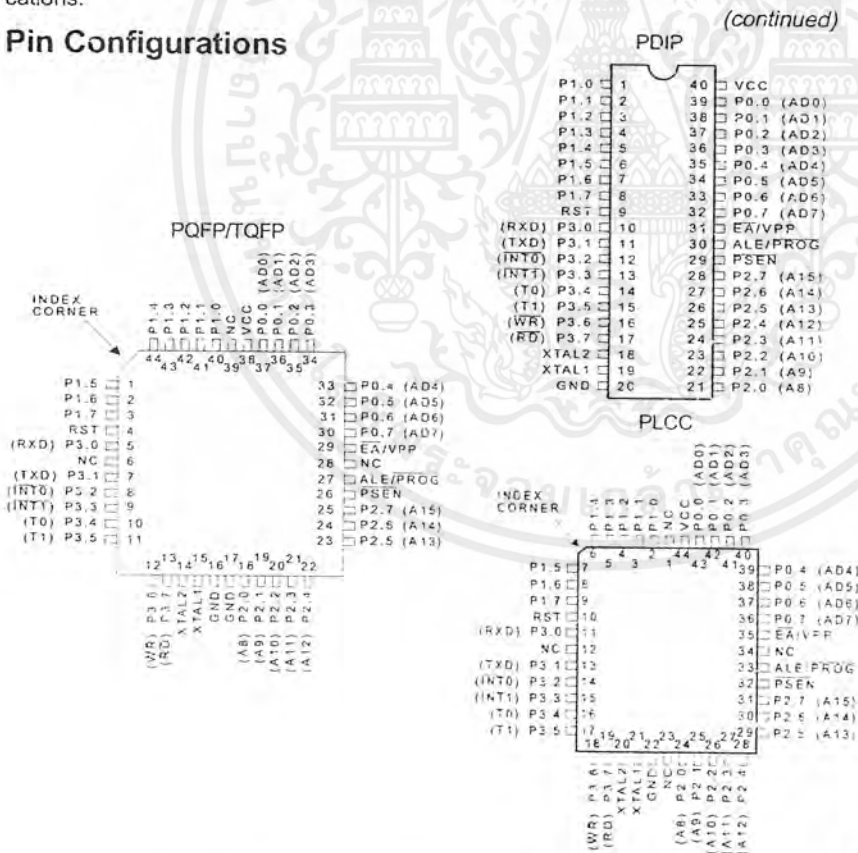
Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ 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 Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

Pin Configurations



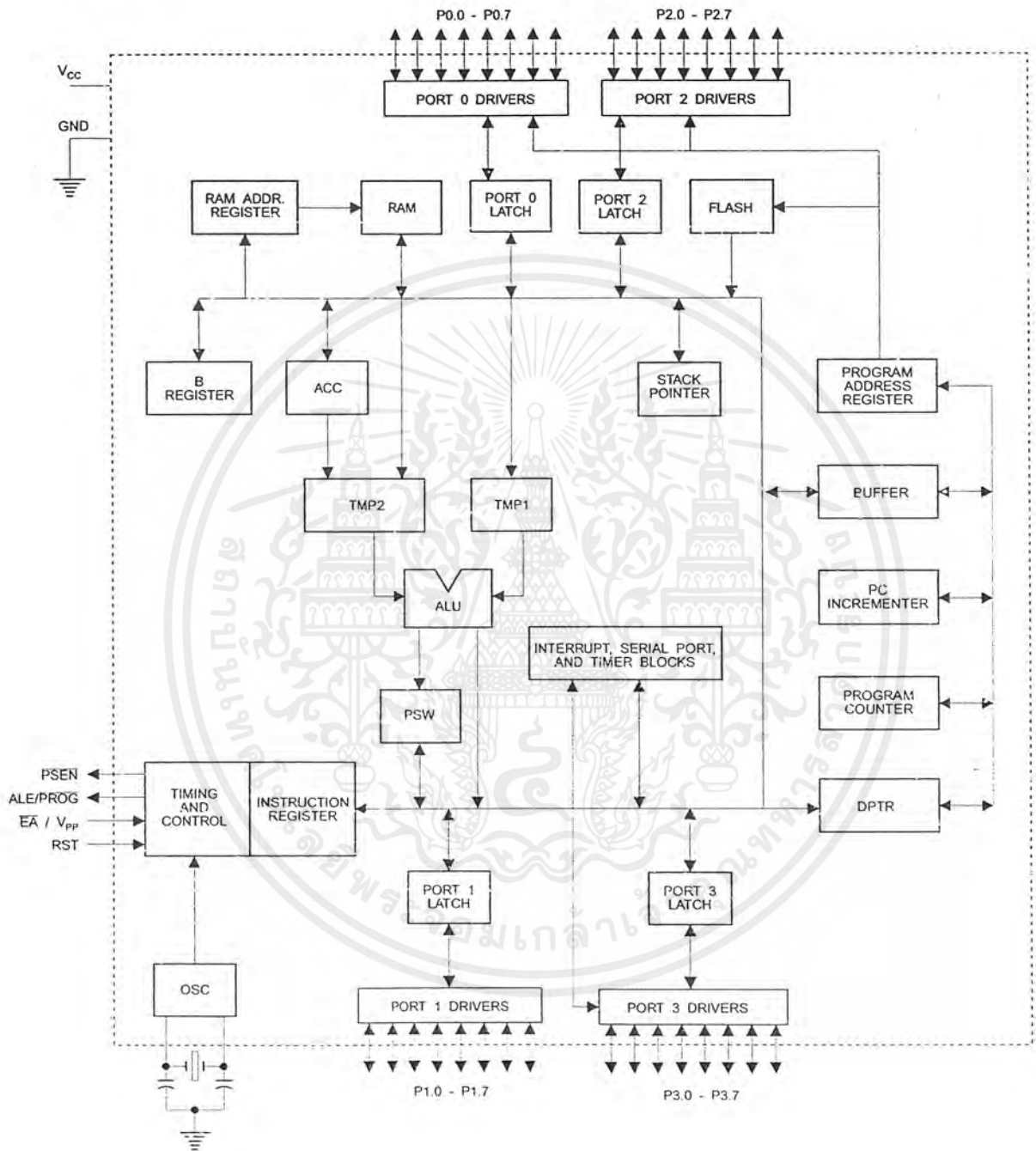
0265F-A-12/97



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT89C51

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 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 hardware reset.

### Pin Description

#### V<sub>CC</sub>

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 may 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.

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

#### 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 it 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 AT89C51 as listed below:

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

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

#### RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

#### ALE/PROG

Address Latch Enable 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 weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

#### PSEN

Program Store Enable is the read strobe to external program memory.





When the AT89C51 is executing code from external program memory,  $\overline{PSEN}$  is activated twice each machine cycle, except that two  $\overline{PSEN}$  activations are skipped during each access to external data memory.

#### $\overline{EA}/V_{PP}$

External Access Enable.  $\overline{EA}$  must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed,  $\overline{EA}$  will be internally latched on reset.

$\overline{EA}$  should be strapped to  $V_{CC}$  for internal program executions.

This pin also receives the 12-volt programming enable voltage ( $V_{PP}$ ) during Flash programming, for parts that require 12-volt  $V_{PP}$ .

#### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

#### XTAL2

Output from the inverting oscillator amplifier.

### Oscillator Characteristics

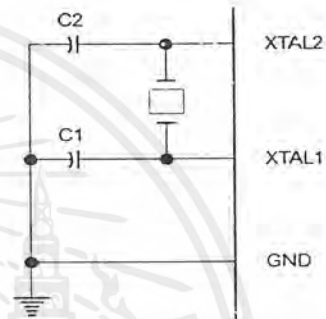
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. 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.

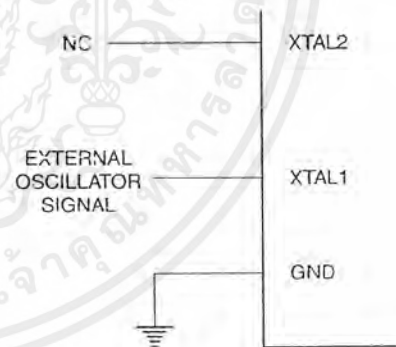
It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port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

Figure 1. Oscillator Connections



Note: C1, C2 = 30 pF  $\pm$  10 pF for Crystals  
= 40 pF  $\pm$  10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



### Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT89C51

### 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. The only exit from power down is a hardware reset. 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.

### 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 <sub>C</sub> instructions executed from external program memory are disabled from fetching code bytes from internal memory. EA is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

### Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage ( $V_{CC}$ ) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	$V_{PP} = 12V$	$V_{PP} = 5V$
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

### Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

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. It is necessary that the latched value of  $\overline{EA}$  be in agreement with the current logic level at that pin in order for the device to function properly.

**Programming Algorithm:** Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, 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 for the high-voltage programming mode.
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 1.5 ms. 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 AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are 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. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 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 lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:** The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/ $\overline{\text{PROG}}$  low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulsed to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel

(031H) = 51H indicates 89C51

(032H) = FFH indicates 12V programming

(032H) = 05H indicates 5V programming

## Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased 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.

## Flash Programming Modes

Mode	RST	$\overline{\text{PSEN}}$	ALE/ $\overline{\text{PROG}}$	$\overline{\text{EA}}/\text{V}_{\text{PP}}$	P2.6	P2.7	P3.6	P3.7
Write Code Data	H	L		H/12V	L	H	H	H
Read Code Data	H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H
	Bit - 2	H	L		H/12V	H	H	L
	Bit - 3	H	L		H/12V	H	L	L
Chip Erase	H	L	(1)	H/12V	H	L	L	L
Read Signature Byte	H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

## AT89C51

Figure 3. Programming the Flash

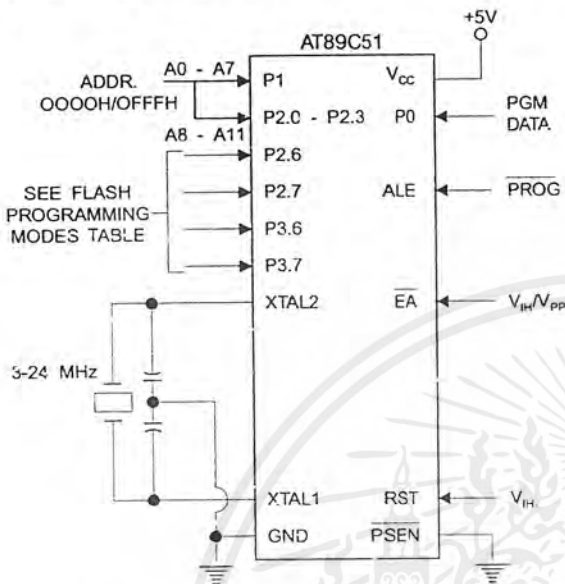
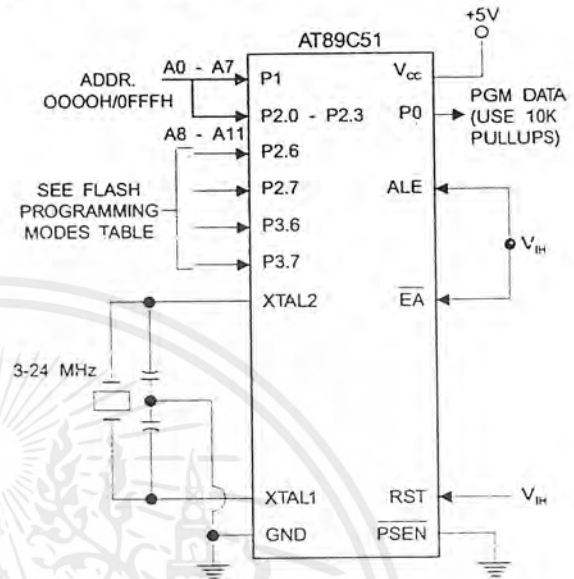


Figure 4. Verifying the Flash



### Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C}$  to  $70^\circ\text{C}$ ,  $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{PP}^{(1)}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}^{(1)}$	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
$t_{AVGL}$	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
$t_{GHAX}$	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
$t_{DVGL}$	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
$t_{GHDX}$	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
$t_{EHS}$	P2.7 (ENABLE) High to $V_{PP}$	$48t_{CLCL}$		
$t_{SHGL}$	$V_{PP}$ Setup to $\overline{PROG}$ Low	10		$\mu\text{s}$
$t_{GHSL}^{(1)}$	$V_{PP}$ Hold After $\overline{PROG}$	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{PROG}$ Width	1	110	$\mu\text{s}$
$t_{AVQV}$	Address to Data Valid		$48t_{CLCL}$	
$t_{ELCV}$	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
$t_{EHOZ}$	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
$t_{GHBL}$	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	$\mu\text{s}$
$t_{WC}$	Byte Write Cycle Time		2.0	ms

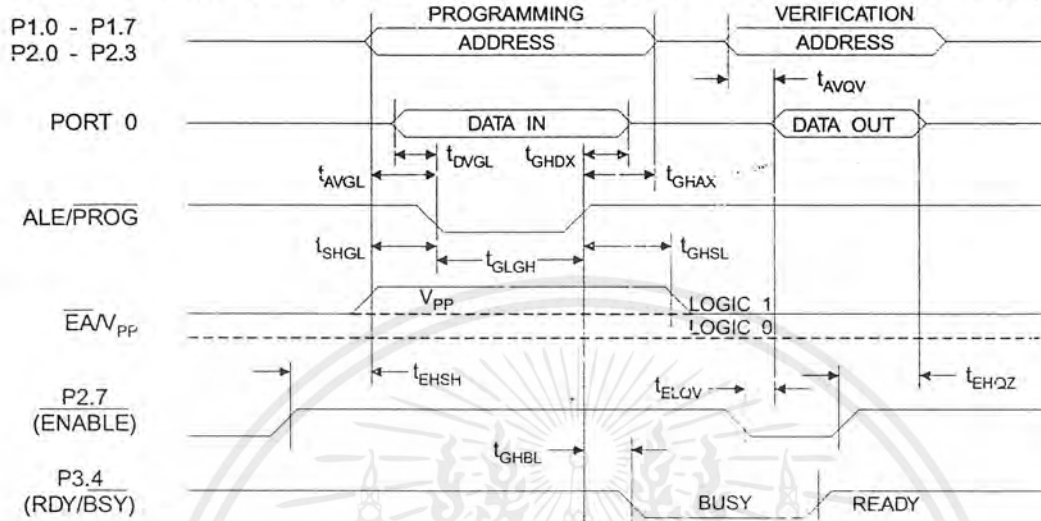
Note: 1. Only used in 12-volt programming mode.



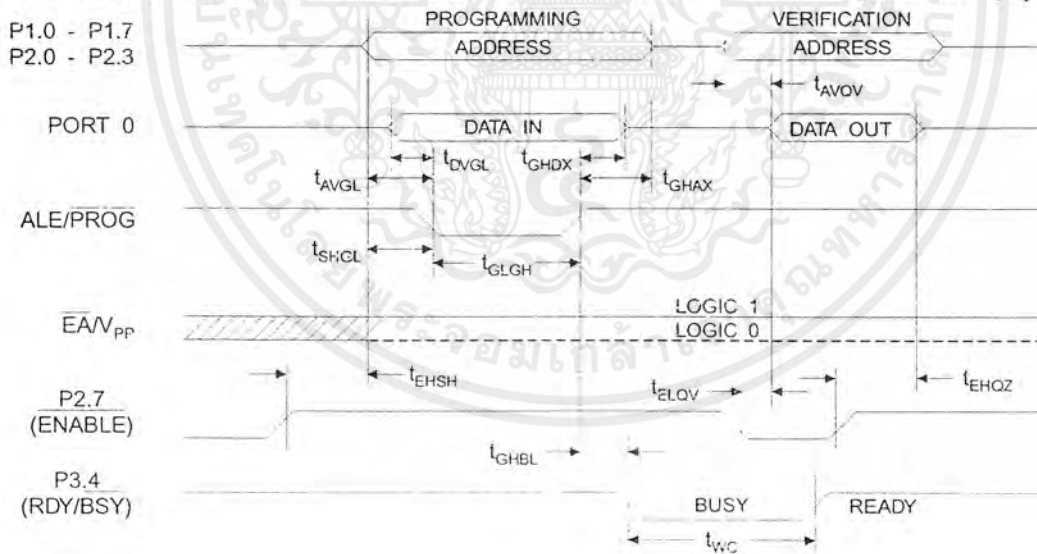
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### Flash Programming and Verification Waveforms - High Voltage Mode ( $V_{PP} = 12V$ )



### Flash Programming and Verification Waveforms - Low Voltage Mode ( $V_{PP} = 5V$ )



## AT89C51

### 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

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 5.0\text{V} \pm 20\%$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low Voltage	(Except $\overline{EA}$ )	-0.5	$0.2 V_{CC} - 0.1$	V
$V_{IL1}$	Input Low Voltage ( $\overline{EA}$ )		-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 <sup>(1)</sup> (Ports 1,2,3)	$I_{OL} = 1.6 \text{ mA}$		0.45	V
$V_{OL1}$	Output Low Voltage <sup>(1)</sup> (Port 0, ALE, PSEN)	$I_{OL} = 3.2 \text{ mA}$		0.45	V
$V_{OH}$	Output High Voltage (Ports 1,2,3, ALE, PSEN)	$I_{OH} = -60 \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -25 \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -10 \mu\text{A}$	$0.5 V_{CC}$		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
$I_{IL}$	Logical 0 Input Current (Ports 1,2,3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1,2,3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-650	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port 0, $\overline{EA}$ )	$0.45 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$		40	$\mu\text{A}$

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)

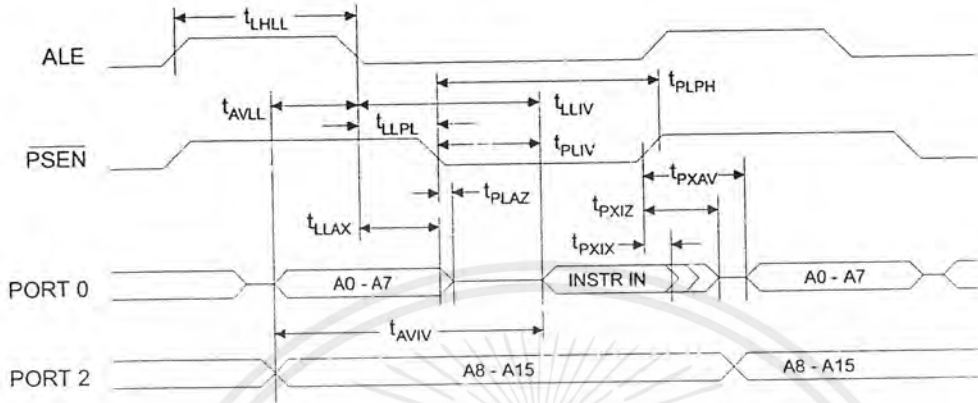
### External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{\text{CLCL}}$	Oscillator Frequency			0	24	MHz
$t_{\text{LHLL}}$	ALE Pulse Width	127		$2t_{\text{CLCL}}-40$		ns
$t_{\text{AVLL}}$	Address Valid to ALE Low	43		$t_{\text{CLCL}}-13$		ns
$t_{\text{LLAX}}$	Address Hold After ALE Low	48		$t_{\text{CLCL}}-20$		ns
$t_{\text{LLIV}}$	ALE Low to Valid Instruction In		233		$4t_{\text{CLCL}}-65$	ns
$t_{\text{LLPL}}$	ALE Low to $\overline{\text{PSEN}}$ Low	43		$t_{\text{CLCL}}-13$		ns
$t_{\text{PLPH}}$	$\overline{\text{PSEN}}$ Pulse Width	205		$3t_{\text{CLCL}}-20$		ns
$t_{\text{PLIV}}$	$\overline{\text{PSEN}}$ Low to Valid Instruction In		145		$3t_{\text{CLCL}}-45$	ns
$t_{\text{PXIX}}$	Input Instruction Hold After $\overline{\text{PSEN}}$	0		0		ns
$t_{\text{PXIZ}}$	Input Instruction Float After $\overline{\text{PSEN}}$		59		$t_{\text{CLCL}}-10$	ns
$t_{\text{PXAV}}$	$\overline{\text{PSEN}}$ to Address Valid	75		$t_{\text{CLCL}}-8$		ns
$t_{\text{AVIV}}$	Address to Valid Instruction In		312		$5t_{\text{CLCL}}-55$	ns
$t_{\text{PLAZ}}$	$\overline{\text{PSEN}}$ Low to Address Float		10		10	ns
$t_{\text{RLRH}}$	$\overline{\text{RD}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
$t_{\text{WLWH}}$	$\overline{\text{WR}}$ Pulse Width	400		$6t_{\text{CLCL}}-100$		ns
$t_{\text{RLDV}}$	$\overline{\text{RD}}$ Low to Valid Data In		252		$5t_{\text{CLCL}}-90$	ns
$t_{\text{RHDX}}$	Data Hold After $\overline{\text{RD}}$	0		0		ns
$t_{\text{RHDX}}$	Data Float After $\overline{\text{RD}}$		97		$2t_{\text{CLCL}}-28$	ns
$t_{\text{LLDV}}$	ALE Low to Valid Data In		517		$8t_{\text{CLCL}}-150$	ns
$t_{\text{AVDV}}$	Address to Valid Data In		585		$9t_{\text{CLCL}}-165$	ns
$t_{\text{LLWL}}$	ALE Low to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	200	300	$3t_{\text{CLCL}}-50$	$3t_{\text{CLCL}}+50$	ns
$t_{\text{AVWL}}$	Address to $\overline{\text{RD}}$ or $\overline{\text{WR}}$ Low	203		$4t_{\text{CLCL}}-75$		ns
$t_{\text{QVWX}}$	Data Valid to $\overline{\text{WR}}$ Transition	23		$t_{\text{CLCL}}-20$		ns
$t_{\text{QVWH}}$	Data Valid to $\overline{\text{WR}}$ High	433		$7t_{\text{CLCL}}-120$		ns
$t_{\text{WHOX}}$	Data Hold After $\overline{\text{WR}}$	33		$t_{\text{CLCL}}-20$		ns
$t_{\text{RLAZ}}$	$\overline{\text{RD}}$ Low to Address Float		0		0	ns
$t_{\text{WHLH}}$	$\overline{\text{RD}}$ or $\overline{\text{WR}}$ High to ALE High	43	123	$t_{\text{CLCL}}-20$	$t_{\text{CLCL}}+25$	ns

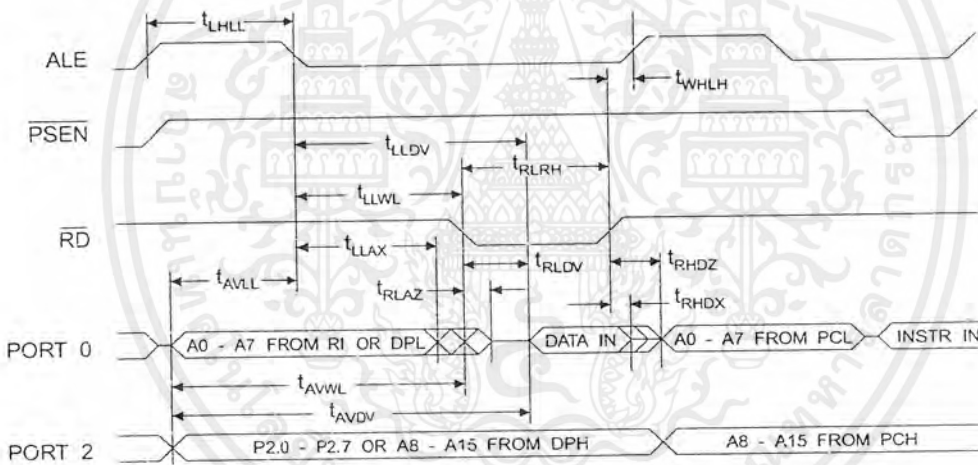
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**AT89C51**

**External Program Memory Read Cycle**



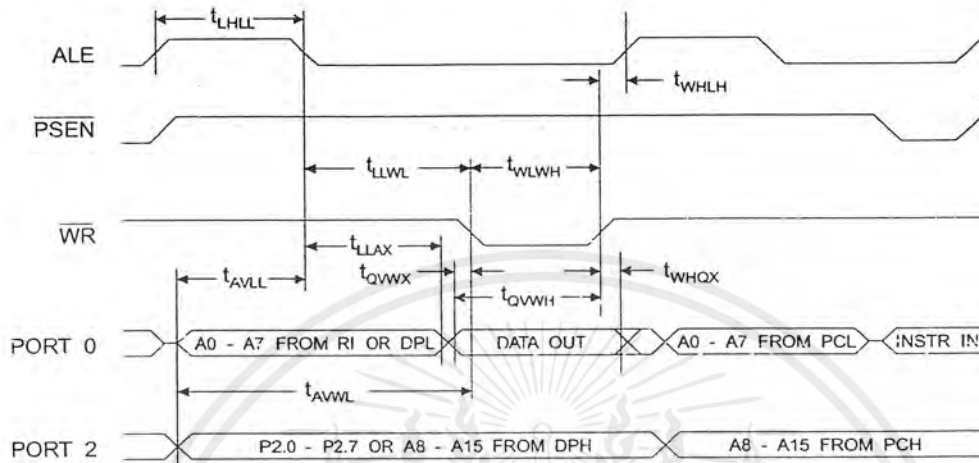
**External Data Memory Read Cycle**



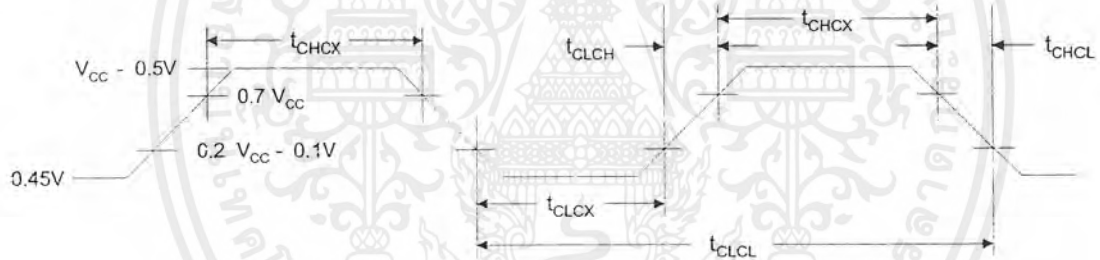
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## External Data Memory Write Cycle



## External Clock Drive Waveforms



## External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
$t_{CLCL}$	Clock Period	41.6		ns
$t_{CHCX}$	High Time	15		ns
$t_{CLCX}$	Low Time	15		ns
$t_{CLCH}$	Rise Time		20	ns
$t_{CHCL}$	Fall Time		20	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

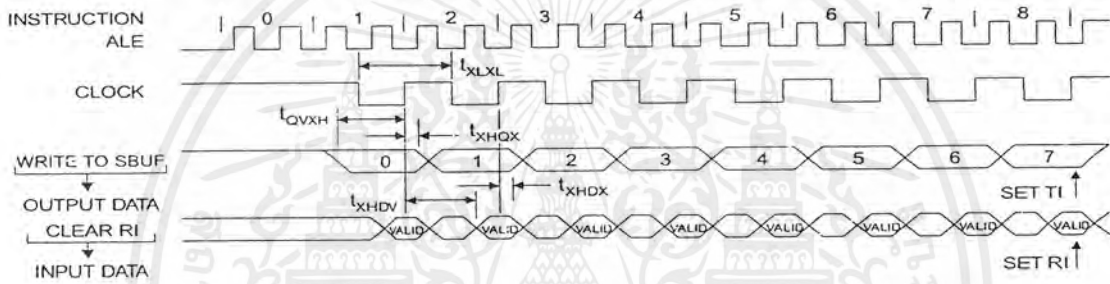
# AT89C51

## Serial Port Timing: Shift Register Mode Test Conditions

( $V_{CC} = 5.0\text{ V} \pm 20\%$ ; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
$t_{XLXL}$	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		$\mu\text{s}$
$t_{QVXH}$	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
$t_{XHDX}$	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
$t_{XHDX}$	Input Data Hold After Clock Rising Edge	0		0		ns
$t_{XHDV}$	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

### Shift Register Mode Timing Waveforms



### AC Testing Input/Output Waveforms<sup>(1)</sup> Float Waveforms<sup>(1)</sup>



Note: 1. AC Inputs during testing are driven at  $V_{CC} - 0.5\text{V}$  for a logic 1 and  $0.45\text{V}$  for a logic 0. Timing measurements are made at  $V_{IH}$  min. for a logic 1 and  $V_{IL}$  max. for a logic 0.

Note: 1. For timing purposes, a port pin is no longer floating when a  $100\text{ mV}$  change from load voltage occurs. A port pin begins to float when  $100\text{ mV}$  change from the loaded  $V_{OH}/V_{OL}$  level occurs.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial (0°C to 70°C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial (-40°C to 85°C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P5	
		AT89C51-12QI	44Q	
		AT89C51-12AA	44A	Automotive (-40°C to 105°C)
		AT89C51-12JA	44J	
		AT89C51-12PA	40P6	
		AT89C51-12QA	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial (0°C to 70°C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial (-40°C to 85°C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
		AT89C51-16AA	44A	Automotive (-40°C to 105°C)
		AT89C51-16JA	44J	
		AT89C51-16PA	40P6	
		AT89C51-16QA	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial (0°C to 70°C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial (-40°C to 85°C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AT89C51

### Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5V ± 20%	AT89C51-24AC	44A	Commercial (0°C to 70°C)
		AT89C51-24JC	44J	
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial (-40°C to 85°C)
		AT89C51-24JI	44J	
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	



Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Features

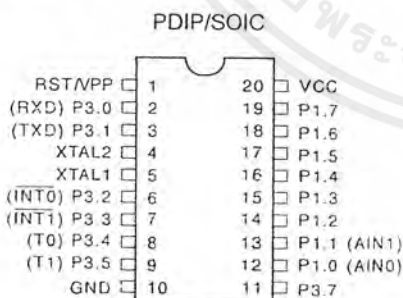
- Compatible with MCS-51™ Products
- 2K Bytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes

## Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2K Bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2K Bytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 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 hardware reset.

## Pin Configuration



## 8-Bit Microcontroller with 2K Bytes Flash

### AT89C2051

0368D-B-12/97

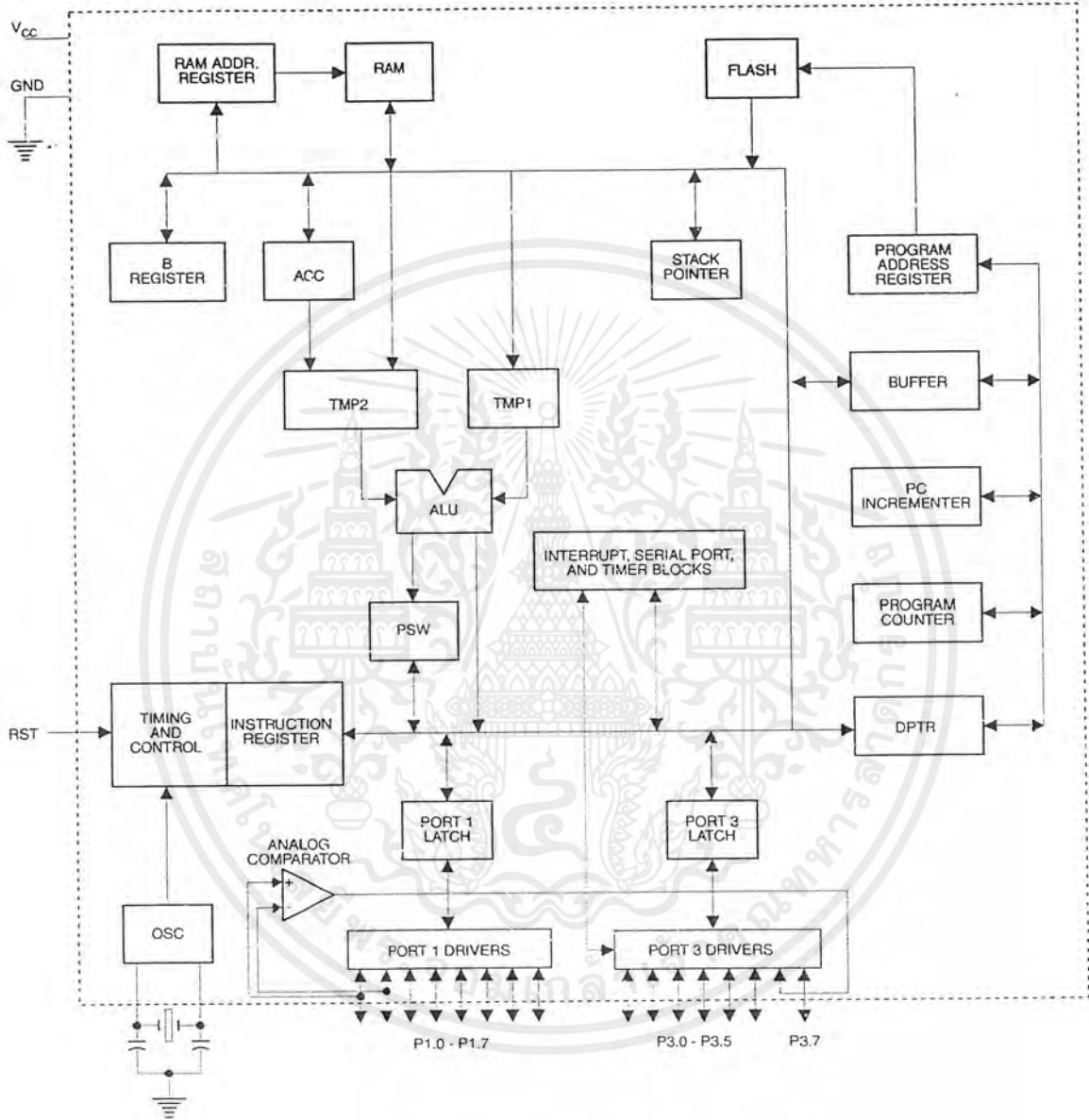


4-15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# AT89C2051

## Pin Description

**V<sub>CC</sub>**  
Supply voltage.

**GND**  
Ground.

### Port 1

Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current ( $I_{IL}$ ) because of the internal pullups.

Port 1 also receives code data during Flash programming and verification.

### Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. 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 AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

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

### RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

### XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

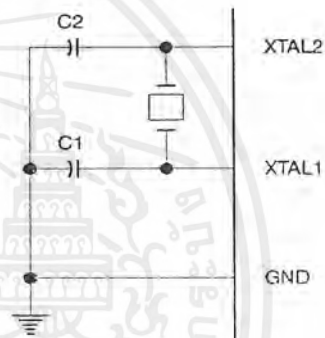
### XTAL2

Output from the inverting oscillator amplifier.

## Oscillator Characteristics

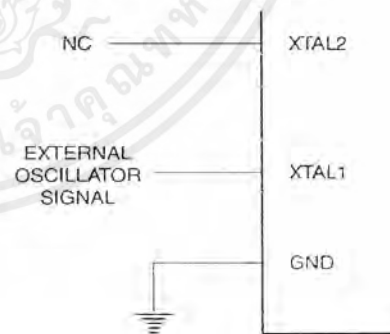
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. 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.

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration





## Special Function Registers

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

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.

**Table 1.** AT89C2051 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H								0CFH
0C0H								0C7H
0B8H	IP XXX00000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0XX00000							0AFH
0A0H								0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H		SP 00000111	DPL 00000000	DPH 00000000			PCON 0XXX0000	87H

## AT89C2051

### Restrictions on Certain Instructions

The AT89C2051 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2K bytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

#### 1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

#### 2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

### Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

#### Lock Bit Protection Modes<sup>(1)</sup>

Program Lock Bits			Protection Type
	LB1	LB2	
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation.

### 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.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle 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 is terminated by reset, the instruction following the one that invokes Idle should not be one that writes 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. The only exit from power down is a hardware reset. 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.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.





## Programming The Flash

The AT89C2051 is shipped with the 2K bytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

**Internal Address Counter:** The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

**Programming Algorithm:** To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:  
Apply power between  $V_{CC}$  and GND pins  
Set RST and XTAL1 to GND
2. Set pin RST to 'H'  
Set pin P3.2 to 'H'
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming.
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropriate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2K bytes array or until the end of the object file is reached.
10. Power-off sequence:  
set XTAL1 to 'L'  
set RST to 'L'  
Turn  $V_{CC}$  power off

**Data Polling:** The AT89C2051 features  $\overline{\text{Data}}$  Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin.  $\overline{\text{Data}}$  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/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

**Program Verify:** If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

**Chip Erase:** The entire PEROM array (2K bytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

**Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel

(001H) = 21H indicates 89C2051

## Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased 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.

# AT89C2051

## Flash Programming Modes

Mode		RST/VPP	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data <sup>(1)(3)</sup>		12V		L	H	H	H
Read Code Data <sup>(1)</sup>		H	H	L	L	H	H
Write Lock	Bit - 1	12V		H	H	H	H
	Bit - 2	12V		H	H	L	L
Chip Erase		12V		H	L	L	L
Read Signature Byte		H	H	L	L	L	L

- Notes:
1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL 1 pin.
  2. Chip Erase requires a 10-ms PROG pulse.
  3. P3.1 is pulled Low during programming to indicate RDY/BSY.

Figure 3. Programming the Flash Memory

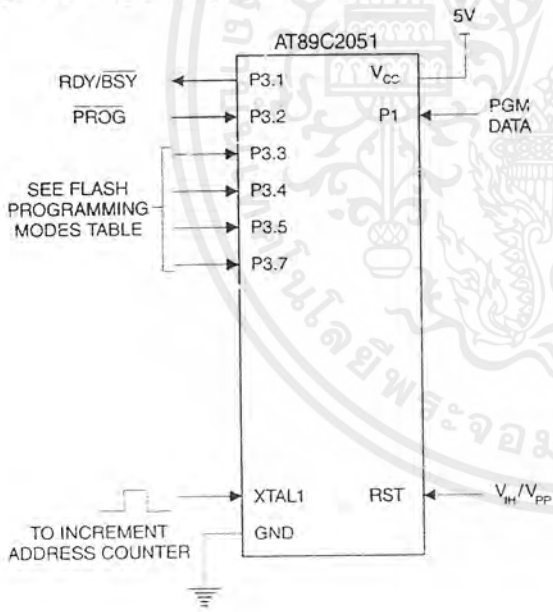
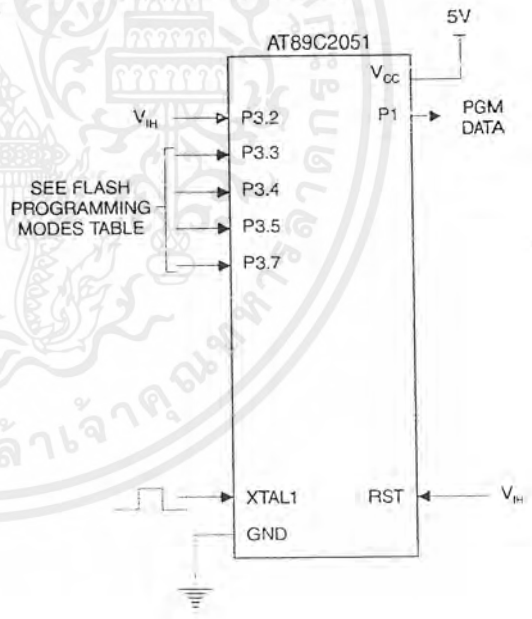


Figure 4. Verifying the Flash Memory



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



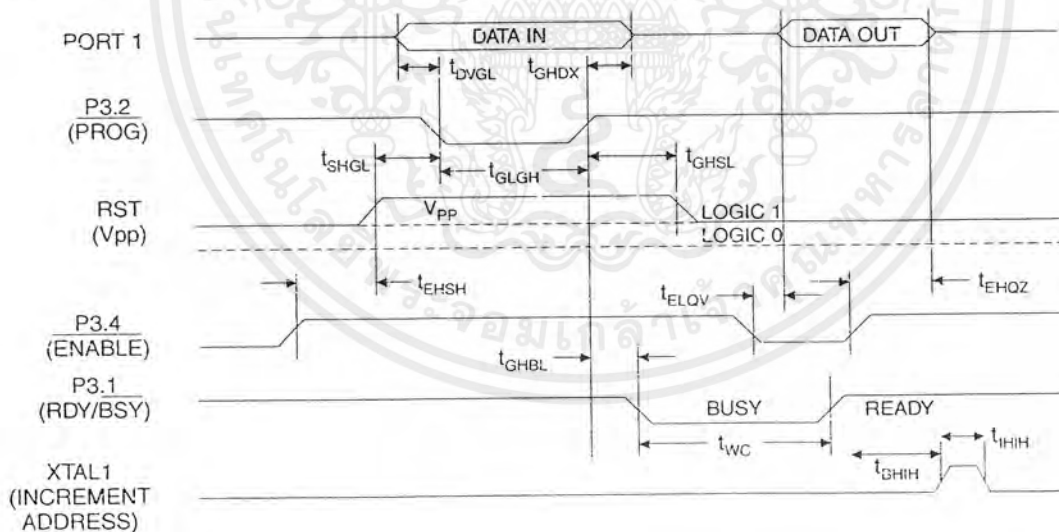
## Flash Programming and Verification Characteristics

$T_A = 0^\circ\text{C to } 70^\circ\text{C}$ ,  $V_{CC} = 5.0 \pm 10\%$

Symbol	Parameter	Min	Max	Units
$V_{PP}$	Programming Enable Voltage	11.5	12.5	V
$I_{PP}$	Programming Enable Current		250	$\mu\text{A}$
$t_{DVGL}$	Data Setup to $\overline{\text{PROG}}$ Low	1.0		$\mu\text{s}$
$t_{GHDX}$	Data Hold After $\overline{\text{PROG}}$	1.0		$\mu\text{s}$
$t_{EHS}$	P3.4 ( $\overline{\text{ENABLE}}$ ) High to $V_{PP}$	1.0		$\mu\text{s}$
$t_{SHGL}$	$V_{PP}$ Setup to $\overline{\text{PROG}}$ Low	10		$\mu\text{s}$
$t_{GHSL}$	$V_{PP}$ Hold After $\overline{\text{PROG}}$	10		$\mu\text{s}$
$t_{GLGH}$	$\overline{\text{PROG}}$ Width	1	110	$\mu\text{s}$
$t_{ELQV}$	$\overline{\text{ENABLE}}$ Low to Data Valid		1.0	$\mu\text{s}$
$t_{EHQZ}$	Data Float After $\overline{\text{ENABLE}}$	0	1.0	$\mu\text{s}$
$t_{GHBL}$	$\overline{\text{PROG}}$ High to $\overline{\text{BUSY}}$ Low		50	ns
$t_{WC}$	Byte Write Cycle Time		2.0	ms
$t_{BHIH}$	$\text{RDY}/\overline{\text{BSY}}$ to Increment Clock Delay	1.0		$\mu\text{s}$
$t_{IHIL}$	Increment Clock High	200		ns

Note: 1. Only used in 12-volt programming mode.

## Flash Programming and Verification Waveforms



## AT89C2051

### 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.....	25.0 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

$T_A = -40^\circ\text{C}$  to  $85^\circ\text{C}$ ,  $V_{CC} = 2.0\text{V}$  to  $6.0\text{V}$  (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
$V_{IL}$	Input Low Voltage		-0.5	$0.2 V_{CC} - 0.1$	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 <sup>(1)</sup> (Ports 1, 3)	$I_{OL} = 20\text{ mA}$ , $V_{CC} = 5\text{V}$ $i_{OL} = 10\text{ mA}$ , $V_{CC} = 2.7\text{V}$		0.5	V
$V_{OH}$	Output High Voltage (Ports 1, 3)	$I_{OH} = -80\ \mu\text{A}$ , $V_{CC} = 5\text{V} \pm 10\%$	2.4		V
		$I_{OH} = -30\ \mu\text{A}$	$0.75 V_{CC}$		V
		$I_{OH} = -12\ \mu\text{A}$	$0.9 V_{CC}$		V
$I_{IL}$	Logical 0 Input Current (Ports 1, 3)	$V_{IN} = 0.45\text{V}$		-50	$\mu\text{A}$
$I_{TL}$	Logical 1 to 0 Transition Current (Ports 1, 3)	$V_{IN} = 2\text{V}$ , $V_{CC} = 5\text{V} \pm 10\%$		-750	$\mu\text{A}$
$I_{LI}$	Input Leakage Current (Port P1.0, P1.1)	$0 < V_{IN} < V_{CC}$		$\pm 10$	$\mu\text{A}$
$V_{OS}$	Comparator Input Offset Voltage	$V_{CC} = 5\text{V}$		20	mV
$V_{CM}$	Comparator Input Common Mode Voltage		0	$V_{CC}$	V
RRST	Reset Pulldown Resistor		50	300	$\text{K}\Omega$
$C_{IO}$	Pin Capacitance	Test Freq. = 1 MHz, $T_A = 25^\circ\text{C}$		10	pF
$I_{CC}$	Power Supply Current	Active Mode, 12 MHz, $V_{CC} = 6\text{V}/3\text{V}$		15/5.5	mA
		Idle Mode, 12 MHz, $V_{CC} = 6\text{V}/3\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		5/1	mA
	Power Down Mode <sup>(2)</sup>	$V_{CC} = 6\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		100	$\mu\text{A}$
		$V_{CC} = 3\text{V}$ P1.0 & P1.1 = 0V or $V_{CC}$		20	$\mu\text{A}$

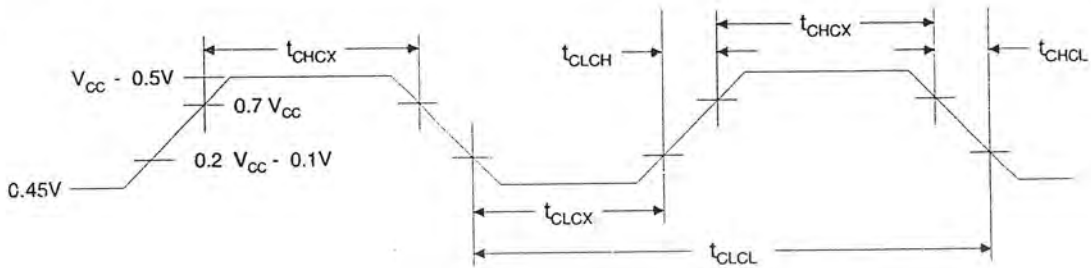
- Notes: 1. Under steady state (non-transient) conditions,  $I_{OL}$  must be externally limited as follows:  
 Maximum  $I_{OL}$  per port pin: 20 mA  
 Maximum total  $I_{OL}$  for all output pins: 80 mA  
 If  $I_{OL}$  exceeds the test condition,  $V_{OL}$  may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum  $V_{CC}$  for Power Down is 2V.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## External Clock Drive Waveforms



## External Clock Drive

Symbol	Parameter	$V_{CC} = 2.7V \text{ to } 6.0V$		$V_{CC} = 4.0V \text{ to } 6.0V$		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
$t_{CLCL}$	Clock Period	83.3		41.6		ns
$t_{CHCX}$	High Time	30		15		ns
$t_{CLCX}$	Low Time	30		15		ns
$t_{CLCH}$	Rise Time		20		20	ns
$t_{CHCL}$	Fall Time		20		20	ns

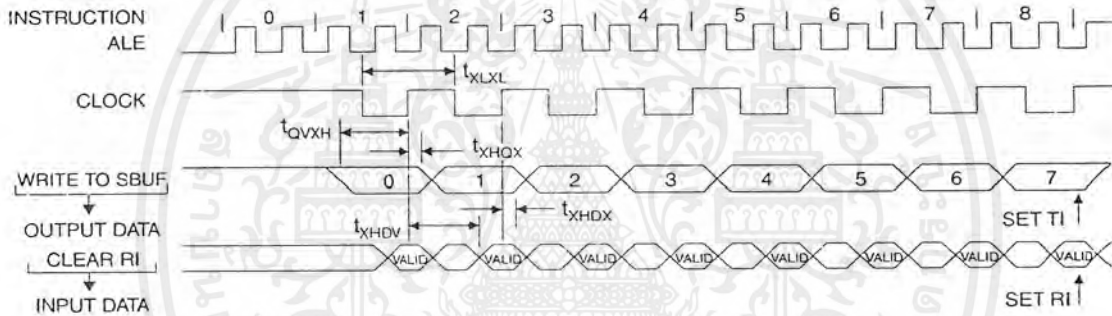
# AT89C2051

## Serial Port Timing: Shift Register Mode Test Conditions

( $V_{CC} = 5.0V \pm 20\%$ ; Load Capacitance = 80 pF)

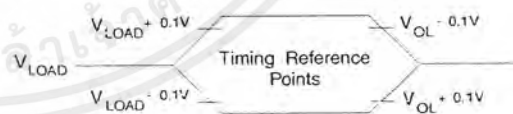
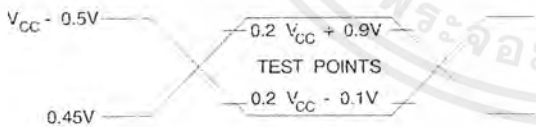
Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
$t_{XLXL}$	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		$\mu s$
$t_{QVXH}$	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}-133$		ns
$t_{XHOX}$	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}-117$		ns
$t_{XHDX}$	Input Data Hold After Clock Rising Edge	0		0		ns
$t_{XHDV}$	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}-133$	ns

## Shift Register Mode Timing Waveforms



## AC Testing Input/Output Waveforms<sup>(1)</sup>

## Float Waveforms<sup>(1)</sup>



Note: 1. AC Inputs during testing are driven at  $V_{CC} - 0.5V$  for a logic 1 and  $0.45V$  for a logic 0. Timing measurements are made at  $V_{IH}$  min. for a logic 1 and  $V_{IL}$  max. for a logic 0.

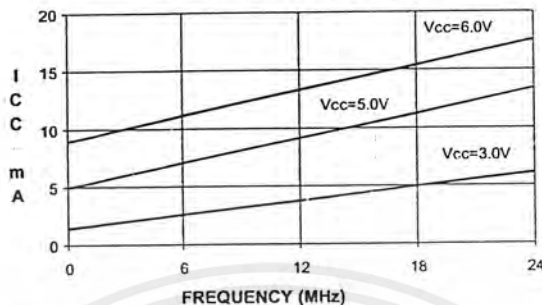
Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded  $V_{OH}/V_{OL}$  level occurs.



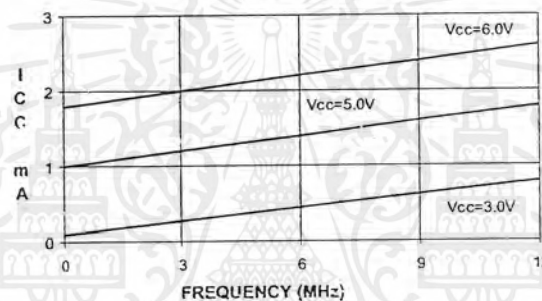
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



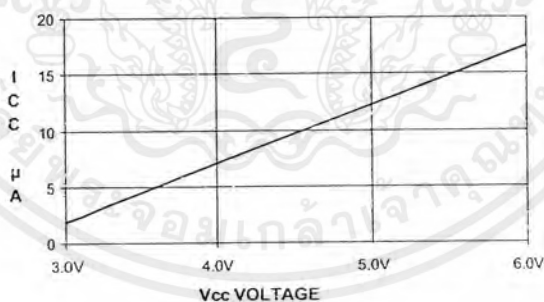
**AT89C2051**  
TYPICAL ICC - ACTIVE (85°C)



**AT89C2051**  
TYPICAL ICC - IDLE (85°C)



**AT89C2051**  
TYPICAL ICC vs. VOLTAGE- POWER DOWN (85°C)



- Notes:
1. XTAL1 tied to GND for  $I_{CC}$  (power down)
  2. P1.0 and P1.1 =  $V_{CC}$  or GND
  3. Lock bits programmed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Ordering Information**

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7V to 6.0V	AT89C2051-12PC	20P3	Commercial (0°C to 70°C)
		AT89C2051-12SC	20S	
		AT89C2051-12PI	20P3	Industrial (-40°C to 85°C)
		AT89C2051-12SI	20S	
		AT89C2051-12PA	20P3	Automotive (-40°C to 105°C)
		AT89C2051-12SA	20S	
24	4.0V to 6.0V	AT89C2051-24PC	20P3	Commercial (0°C to 70°C)
		AT89C2051-24SC	20S	
		AT89C2051-24PI	20P3	Industrial (-40°C to 85°C)
		AT89C2051-24SI	20S	



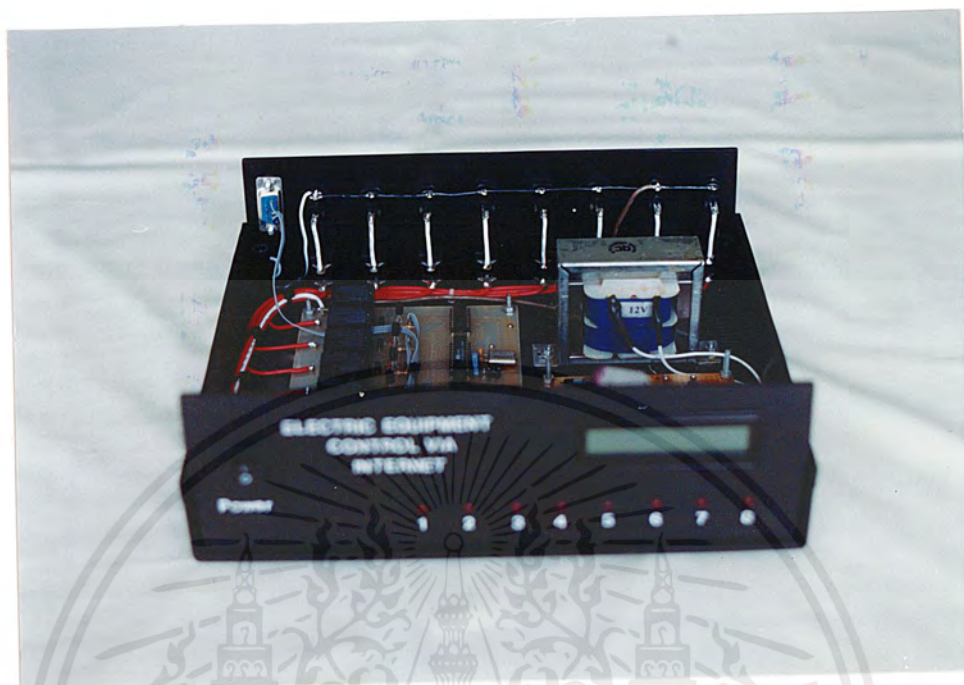
Package Type	
20P3	20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ค

แสดงรูปลักษณะของเครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านอินเทอร์เน็ต



รูปแสดงลักษณะด้านหน้าเครื่อง



รูปแสดงลักษณะด้านหลังเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### กิตติกรรมประกาศ

- รศ.ดร. กอบชัย เดชหาญ และอาจารย์ทุกท่านที่คอยให้คำปรึกษาด้านต่างๆที่เป็นประโยชน์จนทำให้โครงการสำเร็จลุล่วงไปได้ด้วยดี
- พี่ปริญญ์โทที่อยู่ในห้องโปรเจกต์ทุกคนที่คอยอำนวยความสะดวกในด้านอุปกรณ์ต่างๆ
- นาย รัชชชัย เอี่ยมคารา ที่ให้คำปรึกษาเกี่ยวกับ เอเอสพี
- นางสาว พนิดา ศรีไพโรสนธิ ที่ช่วยเหลือในการจัดพิมพ์งานบางส่วน

จึงแสดงความขอบคุณอย่างสูงมาก ณ ที่นี้

จาก คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## หนังสืออ้างอิง

1. รศ. สมยศ จุณณะปิยะ การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 ภาควิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ, 2541
2. สัจจะ จรัสรุ่งรวีวร คู่มือการสร้างแอปพลิเคชันด้วย Visual Basic 6 Basic & Advance : สำนักพิมพ์ อินโฟเพรส, 2542
3. สัจจะ จรัสรุ่งรวีวร Internet Programming ด้วย Visual Basic 6 และ ASP : สำนักพิมพ์ อินโฟเพรส, 2542
4. สัจจะ จรัสรุ่งรวีวร , สมพร จิวรสกุล Active Server Page และ แอปพลิเคชัน ฐานข้อมูลบนอินเทอร์เน็ต : สำนักพิมพ์ อินโฟเพรส, 2541
5. วีรยุทธ ประเสริฐศิริกุล สร้างสรรค์กราฟิกด้วย Microsoft Front Page 98 : สำนักพิมพ์ คอมกราฟเพรส, 2541
6. วีรศักดิ์ เหลืองอุไร คัมภีร์การใช้งาน การสื่อสารอนุกรมบน PC เรียบเรียงจาก Mastering Serial Communications : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 2538
7. จิตเกษม พัฒนาศิริ เสริมแต่งโฮมเพจให้มีชีวิตชีวาด้วย Java Script : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 2541
8. ทรงเกียรติ ภาวดี แกะรอย CGI เพื่อเขียนเรียกสคริปต์เรียกเพจเจอร์/มือถือ ผ่านเว็บ : บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 2542
9. กิตติภูมิ วรฉัตร เพิ่มพลังอินเทอร์เน็ตที่ไฟให้เว็บเพจ ด้วย ASP : บริษัท วิคตี กรุ๊ป จำกัด, 2542
10. รศ.ดร. สุชาย ธนวเสถียร , อาจารย์ นรินทร์ อัครพิเชษฐ Fundamental Of Visual Basic Client-Server Programming : Sum System Company limited, 2541
11. งามนิจ อาจอินทร์ การเขียนโปรแกรมบนเว็บ WEB Programming ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยขอนแก่น, 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้