

เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านอินเทอร์เน็ต

Controlling the Electric Device via Internet Network



โดย
นาย ธีรวัฒน์ พถมินทร์
นาย ทศน์ ศิริโมฆดารา
นาย ศิริพงศ์ เมฆรุ่งเรืองกุล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เลขหมู่.....

เลขทะเบียน...50106

วัน,เดือน,ปี...1.12.45

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านอินเทอร์เน็ต

Controlling the Electric Device via Internet Network

โดย

นาย ธีรวัฒน์ พลมณตรี 42010109

นาย ทศน์ ศิริโมชดารา 42010129

นาย ศิริพงศ์ เมฆรุ่งเรืองกุล 42010346

อาจารย์ที่ปรึกษา

รศ.ดร. สุวิพล ลิทธิชีวกาศ

ผศ. เกียรติกร วงศ์โรจนภรณ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2545

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2545

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านอินเทอร์เน็ต

Controlling the electric device via Internet Network

ผู้จัดทำ

1. นาย ณัฐวิทย์ พฤษมนตร์ 42010109
2. นาย ทศน์ ศิริโมหาคารา 42010129
3. นาย ศิริพงษ์ เมฆรุ่งเรืองกุล 42010346


..... อาจารย์ที่ปรึกษา
(รศ.ดร. สุวิมล สิริธิชิวภาค)


..... อาจารย์ที่ปรึกษา
(ผศ.เกรียงไกร วงศ์โรจนภรณ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าผ่านอินเทอร์เน็ต

Controlling the electric device via Internet Network

โดย

นายฉวีวัฒน์ พถมนตร์ 42010109

นายทัศนีย์ ศิริโมฆคารา 42010129

นายศิริพงษ์ เมฆรุ่งเรืองกุล 42010346

อาจารย์ที่ปรึกษา

รศ.ดร.สุวิมล สิริชีวะภาค

ผศ.เกรียงไกร วงศ์โรจนภรณ์

บทคัดย่อ

วัตถุประสงค์ของโครงการนี้คือ การพัฒนาอุปกรณ์ที่สามารถควบคุมอุปกรณ์ไฟฟ้าโดยผ่านโครงข่าย internet อุปกรณ์ชิ้นนี้จะนำไปใช้เพื่อเป็นอุปกรณ์ที่เชื่อมต่อระหว่างผู้ใช้และอุปกรณ์ไฟฟ้าโดยผู้ใช้สามารถควบคุมเครื่องใช้เหล่านี้ได้แม้ว่าจะไม่อยู่บ้านของตัวเอง โครงการงานชิ้นนี้ได้มีการจัดทำอุปกรณ์ที่ใช้เชื่อมต่อขึ้นมาและมีการเขียนโปรแกรมคอมพิวเตอร์เพื่อใช้ควบคุมอุปกรณ์เหล่านี้ด้วย ขั้นตอนของการทำโครงการงานชิ้นนี้คือ เริ่มจากการศึกษาข้อมูลของอุปกรณ์อิเล็กทรอนิกส์ที่นำมาสร้างอุปกรณ์ วงจรที่ใช้ในอุปกรณ์ และการศึกษาโปรแกรมคอมพิวเตอร์เพื่อที่จะใช้ควบคุมอุปกรณ์เหล่านั้นนอกจากนี้ยังมีการนำวงจรสำเร็จรูปมาประกอบเข้าไปในเครื่องมือเพื่อทำงานได้มีประสิทธิภาพมากขึ้นด้วยและนำสิ่งที่ศึกษามานั้นมาสร้างเป็นชิ้นงาน จากการผลิตอุปกรณ์ที่ใช้เชื่อมต่อเมื่อสำเร็จ อุปกรณ์สามารถใช้ควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้จริง เช่น ควบคุมการเปิด-ปิดหลอดไฟ

ABSTRACT

The subject of this project is to develop a controller that can control the electric devices on internet network. The controller is used for connecting between a user and the electric devices. Although the user is far away from home, he can still control the electric devices. The project is to create the controller for connecting and study the computer language for installing in that controller. The practice of this project is to study DATA of Ics, circuits in the controller, the computer language for controlling the controller. Besides the controller is contained by some complete circuit for more efficient work. Then to take the knowledge to make the instrument. When the controller is complete. The controller can supervise the electric devices, such as controlling the light switches.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51	2
2.1.1 คุณสมบัติของMCS-51	2
2.1.2 คุณสมบัติทั่วไปของ AT89C2051	2
2.1.3 หน้าที่และหลักการ ใช้งานของ AT89C2051	3
2.2 IC MAX 232	4
2.2.1 การสื่อสารพอร์ตอนุกรม RS-232	5
2.2.2 มาตรฐานRS-232	5
2.3 รีเลย์	8
2.4 เว็บเซิร์ฟเวอร์	8
2.5 ระบบเครือข่ายอินเทอร์เน็ต	9
2.6 เวิลด์ไวด์เว็บคืออะไร	9
2.7 โสมเพจ	11
2.8 IP Address	11
2.9 TCP/IP Protocol	11
2.9.1 ส่วนประกอบของ TCP/IP	12
2.9.1.1 กลุ่มข้อกำหนดรูปแบบการขนส่ง	12
2.9.1.2 กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง	12
2.9.1.3 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย	13
2.9.1.4 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางสื่อสารระหว่างเครือข่าย	13
2.9.1.5 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้	13
2.9.2 สถาปัตยกรรม TCP/IP	14
2.10 DNS	15
2.11 HTML	15
2.12 ASP	16
2.12.1 ขบวนการทำงานของ เอเอสพี	16
2.12.2 เอกสาร ASP	17
2.12.3 ความต้องการของระบบในการใช้งาน ASP	18
2.12.4 การแบ่งแยกโค้ด ASP ออกจากโค้ด HTML	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 การคำนวณและการสร้าง	20
3.1 หลักการทำงานของโครงงาน	21
3.2 ส่วนของ Hardware	21
3.2.1 ภาควัดจ่ายไฟ	22
3.2.2 วงจร MAX232 และ AT89C2051	23
3.2.3 IC AT89C2051	24
3.2.4 IC MAX 232	24
3.2.5 การเชื่อมต่อผ่านทางพอร์ตอนุกรม	25
3.2.6 วงจรขั้วรีเลย์ (Relay)	25
3.3 ส่วนของ Software	26
3.3.1 โปรแกรมย่อยส่วนที่1(เอเอสพี)	27
3.3.2 บล็อกไดอะแกรมแสดงการทำงานในส่วนเอเอสพี	28
3.3.3 ฟังก์ชันการทำงานในส่วนเอเอสพี	29
3.3.4 โปรแกรมย่อยส่วนที่ 2 (โปรแกรม Inet)	30
3.3.5 ฟังก์ชันการทำงานในส่วนของ Inet	32
3.4 หลักการทำงานในส่วนของโปรแกรมในไมโครคอนโทรลเลอร์	34
3.4.1 ฟังก์ชันการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์	35
3.5 ส่วนการออกแบบฐานข้อมูล	36
บทที่ 4 การทดลองและผลการทดลอง	40
4.1 ส่วนของโฮมเพจ	40
4.1.1 หน้าต่าง ล็อกอิน	40
4.1.2 หน้าต่าง เมนู	41
4.1.3 หน้าต่าง member	42
4.2 ส่วนของโปรแกรม visual basic (INET)	43
4.2.1 หน้าต่างควบคุมเครื่องใช้ไฟฟ้า	43
บทที่ 5 บทวิจารณ์และบทสรุป	44
5.1 ปัญหาที่พบจากการทดลอง	44
5.2 แนวทางแก้ไขปัญหา	44
5.3 แนวทางในการพัฒนา	44
5.4 ข้อเสนอแนะเกี่ยวกับโครงงาน	44

ภาคผนวก

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ 1.1 แสดงระบบที่ใช้ในการควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านเครือข่ายอินเทอร์เน็ต	1
รูปที่ 2.1 แสดงตำแหน่งขาของ AT89C2051	3
รูปที่ 2.2 แสดงการต่อคริสตัลภายนอกของ AT89C2051	4
รูปที่ 2.3 แสดงภาพรวมของ www	8
รูปที่ 2.4 แสดงการทำงานของ ASP	16
รูปที่ 2.5 แสดงองค์ประกอบในเอกสาร ASP ไฟล์หนึ่งๆ	18
รูปที่ 3.1 Block diagram ของโครงการทั้งหมด	20
รูปที่ 3.2 Block Diagram ในส่วนระบบไมโครคอนโทรลเลอร์	21
รูปที่ 3.3 แสดงวงจรภาคจ่ายไฟในบอร์ดรีเลย์	22
รูปที่ 3.4 แสดงวงจรของบอร์ดรีเลย์ซึ่งใช้ควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า	23
รูปที่ 3.5 แสดงตำแหน่งขาของ IC AT89C2051	24
รูปที่ 3.6 แสดงตำแหน่งขาและ Functional Diagram ของ MAX 232	24
รูปที่ 3.7 แสดงการต่อสายระหว่างพอร์ตอนุกรม DB9 กับบอร์ดรีเลย์	25
รูปที่ 3.8 แสดงวงจรขั้วรีเลย์	25
รูปที่ 3.9 แสดงการทำงานของโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเทอร์เน็ต	27
รูปที่ 3.10 แสดงการทำงานในส่วนเอเอสพี	28
รูปที่ 3.11 แสดงผังการทำงานในส่วนเอเอสพี	29
รูปที่ 3.12 แสดงตำแหน่ง Icon ของโปรแกรม Inet	30
รูปที่ 3.13 แสดงการติดตั้ง MSComm 6.0 ใน toolbox ของโปรแกรม Visual Basic 6.0	31
รูปที่ 3.14 ก แสดงผังการทำงานในส่วน วิชาลเบสิก	32
รูปที่ 3.14 ข แสดงผังการทำงานในส่วน วิชาลเบสิก	33
รูปที่ 3.14 ค แสดงผังการทำงานในส่วน วิชาลเบสิก	34
รูปที่ 3.15 แสดงผังการทำงานในส่วนของไมโครคอนโทรลเลอร์	35
รูปที่ 3.16 แสดงโครงสร้างฐานข้อมูล	36
รูปที่ 3.17 แสดงตาราง member	36
รูปที่ 3.18 แสดงตาราง outlet	37
รูปที่ 3.19 แสดงตาราง status	38
รูปที่ 3.20 แสดงตาราง Timer	39
รูปที่ 4.1 แสดงหน้าต่าง login	40
รูปที่ 4.2 แสดงหน้าต่างเมนู	41
รูปที่ 4.3 แสดงหน้าต่าง member	42
รูปที่ 4.4 แสดงหน้าต่างควบคุมของ INET	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

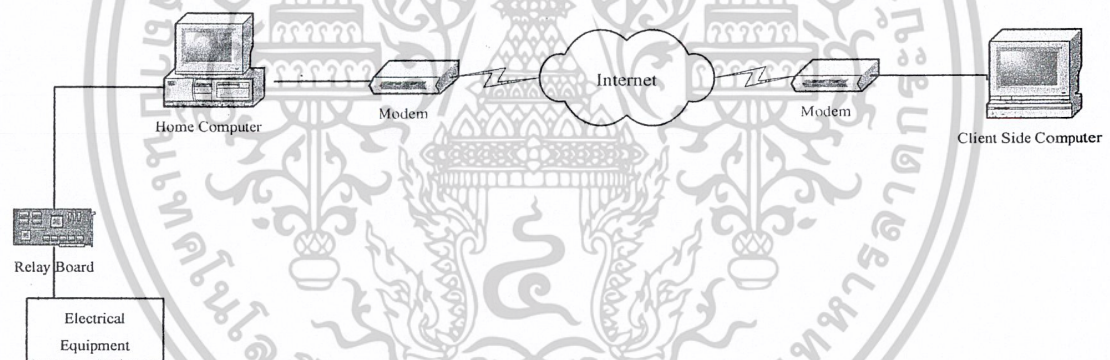
บทที่ 1

บทนำ

ในปัจจุบันคอมพิวเตอร์และอินเทอร์เน็ต ได้เข้ามามีส่วนสำคัญในชีวิตของเราเป็นอย่างมาก และสามารถนำไปประยุกต์ใช้ในงานได้หลากหลายประเภท ในโครงการนี้จึงได้นำคอมพิวเตอร์มาประยุกต์ใช้ในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน โดยคอมพิวเตอร์เป็นเครื่องรับส่งสัญญาณควบคุมผ่านทางระบบเครือข่ายอินเทอร์เน็ตที่มีอยู่ ซึ่งแสดงดังรูป เนื่องจากการใช้งานผ่านเครือข่ายอินเทอร์เน็ตจะช่วยลดค่าใช้จ่ายในการสื่อสาร ซึ่งเดิมจะใช้เครือข่ายโทรศัพท์ ซึ่งจะเสียค่าใช้จ่ายที่สูงกว่าและเปลี่ยนแปลงตามตามระยะทางที่ใช้ในการควบคุม

เนื่องจากการควบคุมอุปกรณ์ไฟฟ้าอยู่ในลักษณะการควบคุมแบบเปิดปิด ไม่ได้นำไปใช้กับงานที่ต้องการความเร็วสูงมาก และเป็นการส่งข้อมูลขนาดเล็กดังนั้น จึงเลือกใช้พอร์ตอนุกรมเป็นตัวกลางในการเชื่อมต่อระหว่างบอร์ดรีเลย์กับคอมพิวเตอร์ เพราะสามารถควบคุมได้ง่ายกว่าพอร์ตนานและ USB

และเพื่อให้ง่ายต่อการควบคุมจึงใช้โปรแกรม web browser สร้างการติดต่อไปยังคอมพิวเตอร์ที่บ้านซึ่งโดยส่วนมากโปรแกรม web browser มักมีการติดตั้งมาพร้อมกับระบบปฏิบัติการอยู่แล้วจึงสะดวกต่อการใช้งาน และไม่ต้องติดตั้งเพิ่มที่ฝั่ง Client



รูปที่ 1.1 แสดงระบบที่ใช้ในการควบคุมอุปกรณ์ไฟฟ้าในบ้านผ่านเครือข่ายอินเทอร์เน็ต โดยโครงสร้างของโครงการมีสองส่วนใหญ่ๆ ดังนี้

ส่วน Software ประกอบด้วย

1. โปรแกรมเชื่อมต่ออินเทอร์เน็ต
2. โปรแกรมเขียน โสแมเพจ
3. โปรแกรมควบคุมอุปกรณ์ไฟฟ้า
4. โปรแกรมฐานข้อมูล

ส่วน Hardware ประกอบด้วย

1. ส่วนของวงจรควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า
2. ส่วนของระบบไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการ

2.1 ไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.1.1 คุณสมบัติของMCS-51 ที่สำคัญๆมีดังนี้

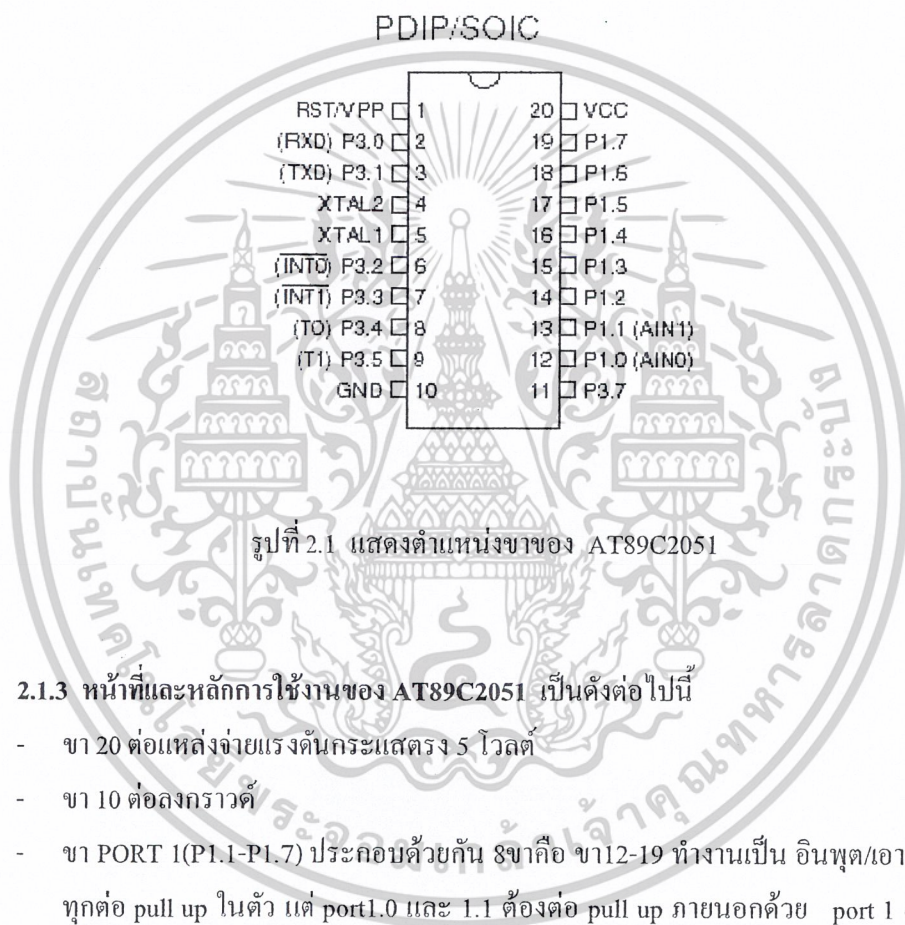
- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว
- มีหน่วยความจำ สำหรับเก็บโปรแกรมควบคุมการทำงานอยู่ในชิป จำนวน 4 กิโลไบต์ (เบอร์ 8031 , 8032 ไม่มีหน่วยความจำส่วนนี้ ส่วนเบอร์ 8052 มีหน่วยความจำส่วนนี้ 8 กิโลไบต์ และเบอร์ 83C51FB จะมีหน่วยความจำส่วนนี้รวมทั้งสิ้น 16 กิโลไบต์)
- มีหน่วยความจำสำหรับโปรแกรมและข้อมูล (RAM) อยู่ในชิปจำนวน 128 ไบต์ (ในเบอร์8031 , 8051) หรือ 256 ไบต์(ในเบอร์ 8031, 8052)
- สามารถใช้หน่วยความจำสำหรับ โปรแกรมและข้อมูลที่อยู่ภายนอกชิปได้ อย่างละ 64 กิโลไบต์ แยกจากกัน
- คำสั่งส่วนใหญ่ใช้เวลาเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกกะเฮิร์ตซ์
- มีพอร์ตที่สามารถรับหรือส่งข้อมูลได้ทั้ง 2 ทิศทาง จำนวน 4 พอร์ต พอร์ตละ 8 บิต หรือสามารถใช้งานเป็นพอร์ตขนาด 1 บิต แยกจากกัน ทำให้เหมือนมีพอร์ตขนาด 1 บิต ใช้งานรวมทั้งสิ้น 32 พอร์ต
- รับและส่งข้อมูลแบบอนุกรมได้ในตัว โดยสามารถกำหนดความเร็วในการรับและส่งข้อมูล (Baud Rate) ได้ตั้งแต่ 300 ถึง 375 กิโลบิตต่อวินาที
- จัดลำดับความสำคัญของสัญญาณอินเทอร์รัปต์ได้ 2 ระดับ
- มีรีจิสเตอร์สำหรับใช้งานเป็นไทม์เมอร์ หรือเคาน์เตอร์ เพื่อนับจำนวนสัญญาณนาฬิกาภายในชิปหรือนับเปลี่ยนสถานะของสัญญาณภายนอก 16 บิต จำนวน 2 ตัว เพื่อใช้สำหรับนับจำนวนพัลส์ วัดความกว้างของพัลส์ หรือใช้วัดช่วงเวลา (ในเบอร์ 8052 จะมี 3 ตัว)
- หน่วยความจำสำหรับเก็บข้อมูลภายในบางส่วน สามารถเข้าถึงข้อมูลได้ทั้งระดับบิต เพื่อให้การออกแบบโปรแกรมและการควบคุมระบบทำได้ง่ายขึ้น
- มีคำสั่งคูณและหารเลขในตัวเอง
- สามารถประมวลผลแบบบูลีน เพื่อใช้งานควบคุมโดยเฉพาะ

2.1.2 คุณสมบัติทั่วไปของ AT89C2051

- เป็น MCS-51 ที่ผลิตโดยบริษัท ATMEL
- มีขนาด 8 บิต 20 ขา
- หน่วยความจำ Flash Memory 2 กิโลไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมและลบได้ 1000 ครั้ง
- ใช้งานในความถี่ 0-24 MHz
- ไฟเลี้ยง 2.7-6 V
- RAM ภายใน 128*8 บิต
- สามารถขับ LED ได้โดยตรง
- ใช้งานคำสั่งเหมือนตัว 8051



รูปที่ 2.1 แสดงตำแหน่งขาของ AT89C2051

2.1.3 หน้าทีและหลักการใช้งานของ AT89C2051 เป็นดังต่อไปนี้

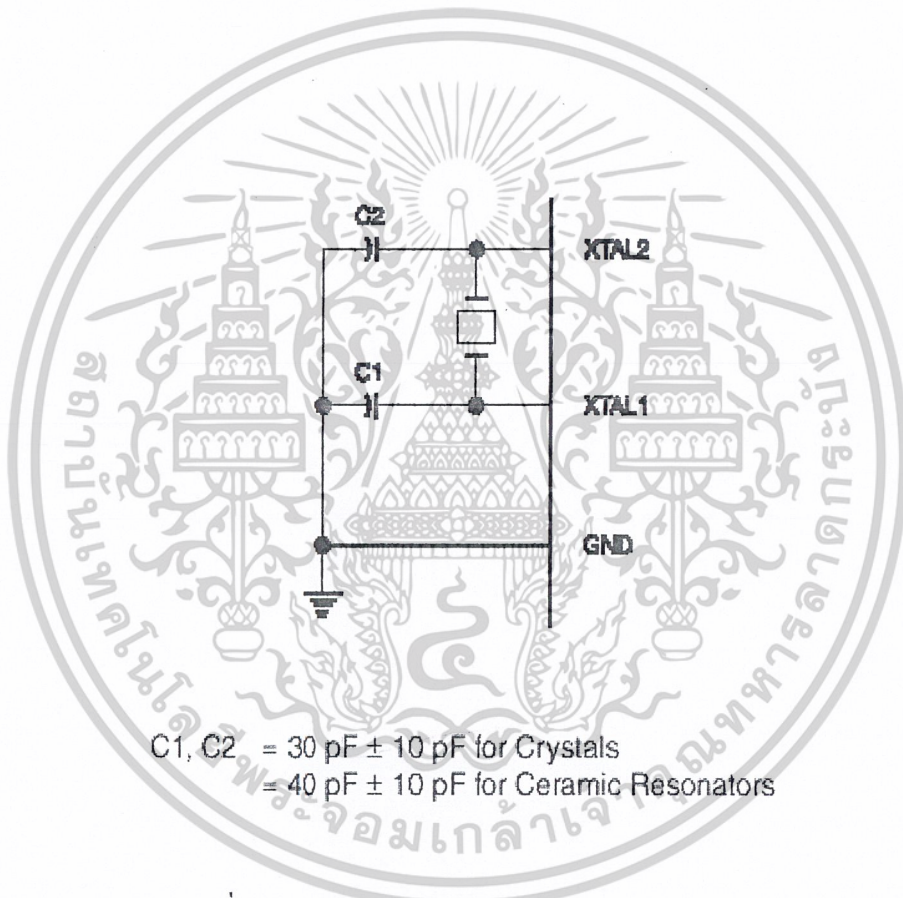
- ขา 20 ต่อแหล่งจ่ายแรงดันกระแสตรง 5 โวลต์
- ขา 10 ต่อลงกราวด์
- ขา PORT 1(P1.1-P1.7) ประกอบด้วยกัน 8ขา คือ ขา12-19 ทำงานเป็น อินพุต/เอาท์พุต พอร์ต ทุกต่อ pull up ในตัว แต่ port1.0 และ 1.1 ต้องต่อ pull up ภายนอกด้วย port 1 ยังใช้ในการติดต่อหน่วยความจำสำหรับ โปรแกรมและข้อมูลภายนอกชิปด้วย และยังสามารถขับ LED ได้โดยตรง
- ขา PORT 3 (P3.0-3.5,P3.7) มี 7 ขา คือ ขา2,3,6-9 และ 11 แต่ละ PIN จะมี pull up อยู่ข้างในแล้ว และสามารถใช้งานเป็น I/O port ทั่วไปได้ แต่ถ้าจะใช้งานเป็นอินพุตพอร์ตจะต้องต่อโหนดค่าหนึ่งลงกราวด์ (pull down) เพื่อลดค่า Pull up ภายใน นอกจากนี้ยังใช้งานตามหน้าที่ต่างๆดังนี้

ขา P 3.0 ใช้รับข้อมูลจากภายนอกแบบอนุกรม

ขา P 3.1 ใช้ส่งข้อมูลออกไปภายนอกแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา P 3.2 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเทอร์รัปชนิดที่ 0
- ขา P 3.3 ใช้เป็นอินพุตเพื่อรับสัญญาณอินเทอร์รัปชนิดที่ 1
- ขา P 3.4 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 0
- ขา P 3.5 สัญญาณอินพุตให้เคาน์เตอร์ของไทม์เมอร์ 1
- ขา RST (ขา 9) ใช้สำหรับการรีเซ็ตวงจรทุกอย่างภายในชิปเพื่อเริ่มต้นการทำงานใหม่ การรีเซ็ตใช้เมื่อเริ่มจ่ายพลังงานหรือเมื่อโปรแกรมเกิดทำงานผิดพลาด เมื่อต้องการรีเซ็ตชิป MCS-51 ขา นี้ ต้องมีสถานะ 1 เป็นอย่างน้อย 2 แมกซ์วินไซเคิลระหว่างที่ออสซิลเลเตอร์ยังทำงานอยู่
- ขา XTAL1 (ขา 5) ใช้ต่อคริสตัลภายนอก โดยเป็น input เข้าสู่วงจร Oscillator
- ขา XTAL2 (ขา 4) ต่อคริสตัลภายนอกโดยเป็น output ออกจากวงจร Oscillator ดังรูป



รูปที่ 2.2 แสดงการต่อคริสตัลภายนอกของ AT89C2051

2.2 IC MAX 232

เป็น IC ที่ทำหน้าที่เปลี่ยนแรงดันที่เข้ามาจาก serial port ไปเป็นแรงดันตามมาตรฐานของ RS-232 โดยเปลี่ยนระดับแรงดัน TTL เพื่อให้ใช้ได้กับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การสื่อสารพอร์ตอนุกรม RS-232

ลักษณะของการส่งข้อมูลแบบอนุกรมนี้ ข้อมูลที่ส่งออกมาที่ละบิต จากตัวส่งไปตัวรับข้อมูล ช่องสัญญาณในการส่งข้อมูลอาจใช้เพียง 1 หรือ 2 ช่องสัญญาณเท่านั้น ทำให้ค่าใช้จ่ายในการสื่อสารจะถูกกว่าแบบขนาน ในการส่งข้อมูลแบบอนุกรม ข้อมูลที่ต้องการส่งจะอยู่ในลักษณะเป็นไบต์จะทยอยส่งทีละบิต และทางตัวรับจะต้องรับข้อมูลเข้ามาทีละบิต แล้วมารวมกันเป็นไบต์ซึ่งทางตัวรับต้องคอยตรวจสอบว่าบิตใดเป็นบิตเริ่มต้นหรือบิตสุดท้ายของข้อมูล การตรวจสอบ นั้นจะขึ้นอยู่กับรูปแบบของรหัสของบิตข้อมูลที่ใช่ ซึ่งในการรับส่งข้อมูลแบบอนุกรมระหว่างไมโครคอมพิวเตอร์กับอุปกรณ์ภายนอกนั้น จำเป็นต้องมีมาตรฐานในการรับส่งข้อมูล ซึ่งมาตรฐานที่นิยมมากที่สุดก็คือ มาตรฐาน RS-232

2.2.2 มาตรฐาน RS-232

เพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันทำงานร่วมกันได้ มาตรฐานหลายชนิดจึงได้รับการออกแบบขึ้น มาตรฐานที่ใช้กันอย่างกว้างขวางมากที่สุดคือ RS-232C ซึ่งโดยปกติไมโครคอมพิวเตอร์จะมีพอร์ตที่เป็นแบบอนุกรมอยู่ในตัวแล้ว

ตามจุดประสงค์ของมาตรฐาน RS-232C นั้นเพื่อจะสามารถเชื่อมต่อกันระหว่างอุปกรณ์รับส่งปลายทาง(DATA TERMINAL EQUIPMENT : DTE) เช่นพอร์ตของคอมพิวเตอร์หลักหรืออุปกรณ์ปลายทางกับอุปกรณ์ RS-232 เป็นข้อกำหนดของการอินเตอร์เฟสมาตรฐาน และสามารถใช้เพื่อจุดประสงค์อื่นๆต่างๆกันไป เช่นการสื่อสารแบบซิงโครนัส (Synchronous Communication) และรูปแบบการสื่อสารที่ต้องการสัญญาณนาฬิกา และสัญญาณกำหนดจังหวะเพิ่มเติมขึ้นมาในความเป็นจริงแล้วเราสามารถทำให้มีการสนทนาระหว่าง DTE และ DCE โดยการให้สายสัญญาณเพียง 3 เส้นเท่านั้น คือ ใช้สาย TD สาย RD และสายกราวด์เท่านั้น

หมายเลขสัญญาณ	ชื่อของสายสัญญาณ
1	Data Carrier
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 2.1 แสดงรายละเอียดการต่อคอนเน็กเตอร์แบบ DB9 มาตรฐาน RS-232.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตเห็นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 1 (Data Carrier Detect Circuit CF,DCD) ขานี้รู้จักกันในนามของ Receive Line Signal Detect (RLSD) หรือขา Carrier Detect(CD) สัญญาณนี้จะเกิด Active เมื่อเกิดสัญญาณนี้พازهที่เหมาะสมระหว่างอุปกรณ์ DCE ที่สถานีกับที่อยู่ระยะไกล เมื่อสัญญาณนี้อยู่ในสภาวะ OFF สัญญาณที่ขา RD ควรทำให้อยู่ในสภาวะ “MARK”
- ขา 2 (Receive Data Circuit BB,RD) สัญญาณที่ขานี้จะถูกสร้างจากอุปกรณ์ DCE กระแสบิตอนุกรมนี้จะกำเนิดที่อุปกรณ์ DTE ปลายทางและเป็นผลผลิตของวงจรรับข้อมูล DCE สัญญาณนี้มักเป็นข้อมูลที่ถูกสร้างขึ้นโดยอุปกรณ์ DCE
- ขา 3 (Transmitted Data circuit BA,TD) เป็นขาสัญญาณข้อมูลที่ออกจากอุปกรณ์ DTE กระแสบิตอนุกรมจากขานี้ คือข้อมูลที่ถูกถ่ายทอดไปโดยโมเด็ม หรือถูกถอดรหัสโดยอุปกรณ์ DCE ที่มี
- ขา 4 (Data Terminal Ready Circuit CD,DTR) สัญญาณ DTR ถูกใช้ในการควบคุมการสวิตช์ อุปกรณ์ DCE เข้ากับตัวกลางในการสื่อสารสัญญาณ DTR ON บ่งชี้ว่าอุปกรณ์ DCE ที่กำลังเชื่อมต่อกันอยู่ ก็ยังต่อร่วมกัน และถ้ายังไม่มีการเชื่อมต่อกันก็สามารถทำให้มีการเชื่อมต่อกันครั้งใหม่ได้ ปกติแล้วสัญญาณ DTR จะอยู่ในสภาวะ OFF เพื่อการกระตุ้นให้อยู่ในสภาวะ “ON HOOK” (วางสาย) อุปกรณ์ DCE โดยปกติแล้วจะตอบสนองต่อการกระตุ้นจากสัญญาณ DTR โดยการทำให้อุปกรณ์ DSR แยกที่ฟ
- ขา 5 (Signal Common Circuit AB) สายนี้จะให้สัญญาณอ้างอิงของกราวด์ร่วมกันสำหรับวงจรการแลกเปลี่ยนกันทั้งหมด ยกเว้นวงจร AA หรือ Protective Ground ข้อกำหนด RS-232 จะอนุญาตให้วงจรนี้ถูกตัดต่อเพิ่มเติมกับ Protective Ground ภายในอุปกรณ์ DCE ได้ถ้าจำเป็น
- ขา 6 (Data Set Ready Circuit CC,DSR) สัญญาณ DSR จะบอกต่ออุปกรณ์ DTE ว่าอุปกรณ์ DCE ได้ต่อกับตัวกลางการสื่อสารที่ถูกต้องแล้ว และในบางกรณีจะบ่งชี้ว่าสายโทรศัพท์ที่อยู่ในสภาวะ “OFF HOOK” สภาวะ “OFF HOOK” จะเป็นตัวบ่งชี้ว่าอุปกรณ์ DCE กำลังอยู่ในโหมด dialing หรือกำลังติดต่อกับอุปกรณ์ DCE อีกตัวหนึ่งอยู่ เมื่อสัญญาณ DSR นี้อยู่ในสภาวะ “OFF” อุปกรณ์ DTE ก็ควรจะถูกกำหนดให้ไม่สนใจสัญญาณอื่นๆทั้งจากอุปกรณ์ DCE ถ้าสัญญาณนี้ถูกกำหนดให้อยู่ในสภาวะ “OFF” ก่อนอุปกรณ์ DTR แล้วอุปกรณ์ DCE ก็ จะสรุปว่าการสื่อสารนี้สิ้นสุดลง
- ขา 7 (Request to Send Circuit CA,RST) สัญญาณนี้จะเตรียมพร้อมอุปกรณ์ DCE สำหรับการทำการส่งข้อมูลเมื่อสัญญาณ RST นี้อยู่ในสภาวะ “ON” จะทำให้อุปกรณ์ DCE อยู่ในโหมดส่งข้อมูล (Transmit mode) ในขณะที่สัญญาณนี้อยู่ในสภาวะ “OFF” ทำให้อุปกรณ์ DCE อยู่ในโหมดรับข้อมูล (Receive mode) อุปกรณ์ DCE ควรจะตอบสนองต่อสัญญาณ RST ON โดยการทำให้สัญญาณ Clear to Send(CTS) อยู่ในสภาวะ “ON” ด้วยเมื่อ RST อยู่ในสภาวะ “OFF” เสียก่อน สัญญาณนี้ไม่ควรจะ “ON” ขึ้นอีก จนกว่าสัญญาณ CTS อยู่ในสภาวะ “OFF”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เสียก่อน สัญญาณนี้จะถูกใช้ร่วมกับสัญญาณ DTR DSR และ DCD ขาสัญญาณ RTS จะถูกใช้
อย่างมากในการควบคุมการไหลของข้อมูล

- ขา 8 (Clear to Send Circuit CB,CTS)สัญญาณนี้จะตอบรับกลับไปยังอุปกรณ์ DTE เมื่อได้รับ
สัญญาณ RST และข้อมูลสามารถถูกส่งออกไปได้ ข้อมูลจะถูกส่งออกไปตามตัวกลางที่ใช้
สื่อสารได้ ก็ต่อเมื่อสัญญาณ CTS นี้ อยู่ในสภาวะ “ON” เท่านั้น สัญญาณนี้จะใช้ร่วมกับ
สัญญาณ DTR DSR และ DCD ขาสัญญาณนี้จะใช้ร่วมกับขา RTS สำหรับควบคุมการไหล
ข้อมูล
- ขา 9 (Ring Indicator Circuit CE,RI) สภาวะ “ON” ของขานี้จะบ่งชี้ว่าได้รับสัญญาณ โทรศัพท์
ตัวกลางในการสื่อสาร โดยปกติแล้วจะขึ้นอยู่กับโปรแกรมควบคุม ในการที่จะทำให้เกิด
สัญญาณนี้ขึ้นหรือไม่

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. ถ้าสายสัญญาณอยู่ในสภาพดีและอุปกรณ์อีกข้างหนึ่งก็พร้อมจะรับข้อมูลแล้ว โดยจะ
ตรวจจับพบสัญญาณพาหะแล้วอุปกรณ์ DCE จะส่งสัญญาณ DCD ออกมา
5. อุปกรณ์ DCEจะตอบสนองด้วยการส่งสัญญาณ CTS ออกมา
6. การติดต่อสื่อสารก็เริ่มขึ้น โปรแกรมควบคุมจะทำการส่งหรือรับข้อมูล

ส่วนลำดับขั้นในการตอบรับที่เป็นดังนี้

1. อุปกรณ์ DTE จะส่งสัญญาณ DTR ออกมา
2. อุปกรณ์ DCE จะอยู่ในโหมดตอบรับอัตโนมัติ (auto answer mode) โดยมีสัญญาณ DSR
ออกมา
3. สถานีปลายทางส่งสัญญาณเรียกอุปกรณ์ DCE และอุปกรณ์ DCE ส่งสัญญาณ RI ออกมา
4. อุปกรณ์ DTE รับรู้ถึง RI ที่ส่งมาจากเครื่องปลายทาง และอุปกรณ์ DCE ก็เข้าสู่สภาวะ OFF
HOOK
5. อุปกรณ์ DCE ทำการแลกเปลี่ยนข้อมูลกับอุปกรณ์ DCE ที่อีกปลายทางหนึ่ง และมีการส่ง
สัญญาณ DCD ออกมา
6. อุปกรณ์ DTE จะส่งสัญญาณ RTS ออกมาหรือจะรอข้อมูลก็ได้ ขึ้นอยู่กับ โปรแกรมควบคุม
7. อุปกรณ์ DCE จะตอบสนองด้วยการส่งสัญญาณ DTS ออกมาการติดต่อสื่อสารก็จะเริ่มขึ้น

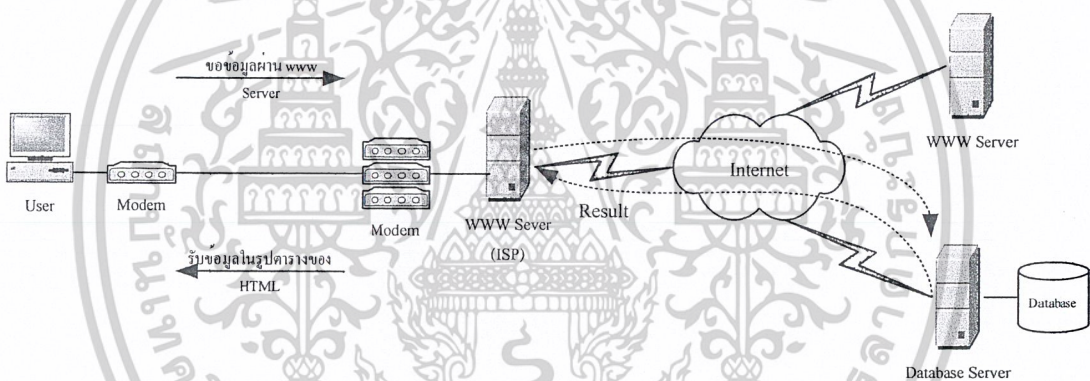
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 รีเลย์ (Relay)

เป็นอุปกรณ์ที่ใช้กระแสต่ำๆ เพื่อควบคุมสวิตช์ให้ตัดต่อโหลดที่มีกระแสสูงๆ โครงสร้างส่วนประกอบของรีเลย์ การทำงานของรีเลย์คือ เมื่อมีแรงดันคร่อมขดลวดจะทำให้เกิดกระแสไหลผ่านขดลวด ซึ่งจะทำให้น้ำสัมผัสเคลื่อนที่ (moving contact) เกิดสนามแม่เหล็กไฟฟ้าดูดหน้าสัมผัส NO (ปกติเปิดวงจร :normally open) ให้ต่อวงจร และเมื่อปลดแรงดันออก สนามแม่เหล็กก็จะหมดลง น้ำสัมผัสเคลื่อนที่ก็จะดีดกลับมาต่อยังหน้าสัมผัส NC (ปกติต่อวงจร :normally close)

2.4 เว็บเซิร์ฟเวอร์

เว็บเซิร์ฟเวอร์ คือ แอปพลิเคชันทำหน้าที่รับ และประมวลผลเอกสาร ที่ถูกร้องขอจากผู้ใช้บริการอินเทอร์เน็ต จากนั้นเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารกลับไปแสดงผลให้ผู้ใช้บริการผ่านเบราว์เซอร์ นอกจากนี้เว็บเซิร์ฟเวอร์จะถูกนำมาให้บริการในอินเทอร์เน็ตแล้ว แต่อาจมีการประยุกต์ให้นำมาใช้กับเครือข่ายภายในองค์กร หรือ อินเทอร์เน็ตได้เช่นกัน



รูปที่ 2.3 แสดงภาพรวมของ www

ซึ่งเป็นบริการที่ได้รับความนิยมสูงสุดของอินเทอร์เน็ต

แต่เดิมนั้นเว็บเซิร์ฟเวอร์มักจะอยู่ในเครื่องคอมพิวเตอร์ ยูนิกซ์ ที่มีประสิทธิภาพสูง และราคาแพงต่อมา เมื่ออินเทอร์เน็ตขยายความนิยมมาสู่ผู้ใช้ พีซี ทำให้มีการพัฒนาซอฟต์แวร์ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์บนพีซี ซึ่งสามารถรันได้ทั้งวินโดวส์ 95/98 และ วินโดวส์เอ็นที เซิร์ฟเวอร์/Workstation ตัวอย่างเช่น

- NCSA Web Server จาก NCSA เป็นเว็บเซิร์ฟเวอร์ที่สามารถดาวน์โหลดได้ฟรีจากเว็บไซต์ที่ให้บริการดาวน์โหลดฟรีทั่วไป
- Net Server จาก Netscape เป็น เวิลด์ ไวด์ เว็บ (World Wide Web : www) เซิร์ฟเวอร์ (Server) ที่มีความสามารถรองรับ จาวา (JAVA) ได้อย่างเต็มรูปแบบ
- ออราเคิลเว็บเซิร์ฟเวอร์ (Oracle Web Server) จากออราเคิล เป็นเว็บเซิร์ฟเวอร์ที่เน้น

ความสามารถด้านการติดต่อกับเซิร์ฟเวอร์ฐานข้อมูลโดยเฉพาะการใช้งานร่วมกับระบบจัดการฐานข้อมูล
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น เมื่อผู้เช่าได้เห็น แบบเซิร์ฟเวอร์อินเทอร์เน็ตค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ ออราเคิล ปัจจุบันเราอาจได้ยินชื่อที่ตั้งขึ้นใหม่เป็น แอปพลิเคชัน เซิร์ฟเวอร์ (Application Server) (คือ ออราเคิล เว็บเซิร์ฟเวอร์ ตัวเดิมที่ถูกพัฒนาขึ้นมา)

- Personal Web Server จากไมโครซอฟท์เป็นเว็บเซิร์ฟเวอร์ที่สามารถใช้งานได้กับ วินโดวส์ 95/98 หรือ วินโดวส์เอ็นที เวอร์กซ์เตชั่น และรองรับการใช้งานร่วมกับ เอเอสพี (Active Server Page) โดยเราสามารถดาวน์โหลดมาใช้ได้ฟรีจากเว็บไซต์ของไมโครซอฟท์ แต่ถ้าใช้งาน วินโดวส์ 98 ก็ สามารถเลือกติดตั้งได้ ซึ่งมักใช้ในการทดสอบเว็บเพจ หรือ แอปพลิเคชันอินเทอร์เน็ต ก่อนจะนำไปใช้งานจริง

- IIS (Internet Information Server) จากไมโครซอฟท์ เป็นอินเทอร์เน็ต เซิร์ฟเวอร์ เวอร์ชัน 2 ที่ แดมมากับ วินโดวส์ เอ็นที เซิร์ฟเวอร์ 4.0 มีความสามารถให้บริการได้ทั้ง เวิลด์ ไวด์ เว็บ, เอฟ ที พี (FTP: File Transfer Protocol) และ Gopher ส่วนเวอร์ชันที่ใช้งานกับเอเอสพีได้จะเป็นเวอร์ชัน 3.0 ขึ้นไป สำหรับบนนี้ จะแสดงการใช้งานกับเวอร์ชัน 4.0

2.5 ระบบเครือข่ายอินเทอร์เน็ต

ในองค์กรหนึ่งๆ อาจมีการติดตั้งระบบเครือข่ายแบบใกล้เคียงตั้งแต่ 2 ระบบขึ้นไปแต่เป็นระบบที่อยู่ห่างไกลกันมาก เช่น อยู่คนละจังหวัด ระบบเครือข่ายแบบใกล้เคียงแต่ละระบบก็สามารถเชื่อมโยงเข้าด้วยกันจนกลายเป็นเครือข่ายที่เรียกว่า เครือข่ายระยะไกล (WAN: Wide Area Network) และนอกจากนี้ ระบบเครือข่ายระยะไกลที่หนึ่ง ก็สามารถเชื่อมกับเครือข่ายระยะไกลที่อยู่ห่างไกลกันออกไปมากๆ ได้อีก เช่น อยู่คนละประเทศหรือคนละทวีป ทำให้เกิดเครือข่ายขนาดใหญ่ที่เรียกว่า "Internetworking" ซึ่งเป็นระบบเครือข่ายขนาดใหญ่และเป็นหลักการที่กลายมาเป็นระบบเครือข่ายอินเทอร์เน็ตในที่สุด

อินเทอร์เน็ต เป็นเครือข่ายคอมพิวเตอร์ที่มีขนาดใหญ่มาก เกิดจากการเชื่อมต่อเครือข่ายคอมพิวเตอร์จำนวนมากมายในโลกเข้าด้วยกัน ไม่ว่าจะเป็เครือข่ายขนาดเล็ก เช่น ระบบเครือข่ายแบบ ใกล้เคียง หรือ ระบบเครือข่ายขนาดใหญ่ เช่น ระบบเครือข่ายของมินิหรือเมนเฟรมคอมพิวเตอร์ซึ่งแต่ละ เครือข่ายก็จะมีเครื่องคอมพิวเตอร์ที่เป็นเครื่องแม่ข่ายหรือ โฮสต์ (Host) ซึ่งมีอยู่หลายชนิดหลายยี่ห้อ โดย จะมีการกำหนดข้อตกลงใจในการสื่อสารที่เรียกว่า โพรโตคอล (Protocol) ขึ้นมา เพื่อให้คอมพิวเตอร์แต่ละ ชนิดสามารถติดต่อสื่อสารกันได้ โพรโตคอลมาตรฐานที่ใช้ในการสื่อสารบนอินเทอร์เน็ตจะมีชื่อ เรียกว่า TCP/IP (Transmission Control Protocol / Internet Protocol)

2.6 เวิลด์ ไวด์ เว็บ คืออะไร

เมื่อไม่กี่ปีก่อนหน้านี้มีผู้สนใจใช้งานอินเทอร์เน็ตไม่มากนัก เนื่องจากการใช้บริการอินเทอร์เน็ต ไม่ว่าจะเป็ การค้นหาข่าวสารข้อมูล การรับส่งจดหมายอิเล็กทรอนิกส์ การสำเนาไฟล์ ฯลฯ จะอยู่ในรูปแบบของตัวอักษร (Text Mode) เท่านั้น ไม่มีการแสดงที่เป็น รูปภาพ เสียง และไม่มีตัวอักษรแบบต่างๆ ปรากฏให้เห็นแต่อย่างใด นอกจากนี้ผู้ใช้ต้องเรียนรู้และจดจำคำสั่งคอมพิวเตอร์มากมาย เช่น ต้องเรียนรู้คำสั่งเบื้องต้นของยูนิกซ์ (UNIX) เนื่องจากเมื่อมีการเรียกใช้อินเทอร์เน็ต เครื่องคอมพิวเตอร์ของผู้ใช้จะถูกเปลี่ยนให้เป็นเทอร์มินอล ของโฮสต์คอมพิวเตอร์ที่ให้บริการอินเทอร์เน็ต และโฮสต์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนมากจะทำงานอยู่ภายใต้ระบบปฏิบัติการยูนิกซ์ ดังนั้นผู้ใช้จึงจะต้องเรียนรู้คำสั่งเบื้องต้นของยูนิกซ์ (เช่นเดียวกับการเรียนรู้คำสั่งดอส (DOS) บนเครื่องพีซี) เพื่อทำการป้องกันคำสั่งที่เป็นตัวอักษรด้วยตัวเองให้โฮสต์คอมพิวเตอร์ทำงานตามที่เราต้องการใช้

จนกระทั่งมีบริการที่เรียกว่า เวิลด์ ไรด์ เว็บ หรือเครือข่ายใยแมงมุมเกิดขึ้น ทำให้ความนิยมการใช้อินเทอร์เน็ตสูงขึ้นเป็นทวีคูณ เนื่องจาก เวิลด์ ไรด์ เว็บ เป็นบริการอันหนึ่งที่มีอยู่ในอินเทอร์เน็ต ทำให้การใช้งานอินเทอร์เน็ตเป็นเรื่องที่ง่ายขึ้น ผู้ใช้ไม่ต้องจดจำคำสั่งของยูนิกซ์อีกต่อไป การอ่านและค้นหาข่าวสารข้อมูล ทำได้โดยการกดปุ่มบนเมาส์เพียงอย่างเดียว นอกจากนี้ ข่าวสาร ข้อมูล หรือเอกสารที่เก็บอยู่ในอินเทอร์เน็ตจะอยู่ในรูปแบบของสื่อผสม (Multimedia) ที่เรียกว่า "เว็บเพจ" (web page) อันประกอบไปด้วยอักษรข้อความที่มีรูปแบบต่างๆ รูปภาพ เสียง ภาพเคลื่อนไหว วิดีโอ และไฮเปอร์เท็กซ์ (Hypertext) ซึ่งเป็นการแสดงข้อความที่มีการเชื่อมโยงถึงกันได้เหมือนกับการเรียกใช้ Help ในโปรแกรมวินโดวส์ โดยผู้ใช้สามารถเรียกดูเอกสารหนึ่งจากอีกเอกสารหนึ่งได้ ซึ่งเป็นลักษณะการเชื่อมต่อที่คล้ายกับใยแมงมุม จึงมีการตั้งชื่อบริการนี้ว่าเป็นเครือข่ายใยแมงมุม

ส่วนประกอบของ เวิลด์ ไรด์ เว็บ

การจะใช้บริการ เวิลด์ ไรด์ เว็บ ได้นั้น จำเป็นต้องมีส่วนประกอบเหล่านี้

- แหล่งข้อมูลหรือเว็บไซต์
- โปรแกรมเว็บเบราว์เซอร์

1. แหล่งข้อมูลหรือเว็บไซต์

เว็บไซต์ หรือ เว็บเซิร์ฟเวอร์ คือ ระบบคอมพิวเตอร์ที่เป็นแหล่งเก็บเว็บเพจที่ผู้ใช้บริการสามารถเรียกดูเว็บเพจที่เก็บอยู่ในเว็บไซต์นั้นได้ ซึ่งเครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเว็บไซต์อาจจะใช้ระบบปฏิบัติการยูนิกซ์ หรือวินโดวส์ เอ็นที ก็ได้ และจะต้องมีการติดตั้งโปรแกรมจัดการ ที่จะทำให้เครื่องคอมพิวเตอร์เครื่องนั้นทำหน้าที่เป็นเว็บไซต์ ตัวอย่างเช่น ในกรณีที่เครื่องคอมพิวเตอร์เครื่องนั้นใช้ระบบปฏิบัติการวินโดวส์ เอ็นที จะมีซอฟต์แวร์เว็บเซิร์ฟเวอร์ให้เลือกใช้

2. โปรแกรมเว็บเบราว์เซอร์

เป็นโปรแกรมหรือซอฟต์แวร์ที่ใช้ในการเข้าสู่ เวิลด์ ไรด์ เว็บ จะเปิดดูเว็บเพจที่เก็บอยู่ในเว็บไซต์ใดๆ ตัวอย่างของ โปรแกรมเว็บเบราว์เซอร์ ที่มีให้ดาวน์โหลดใช้กันได้ฟรีได้แก่ โปรแกรม Netscape Navigator จากบริษัท Netscape Communications และโปรแกรม Microsoft Internet Explorer จากบริษัท ไมโครซอฟท์ ผู้ผลิตโปรแกรมวินโดวส์ที่ผู้คนรู้จักกันดี เป็นต้น ซึ่งโปรแกรมเว็บเบราว์เซอร์ส่วนใหญ่นั้น จะทำงานร่วมกันกับโปรแกรมวินโดวส์ และนอกจากจะใช้เพื่อดูเว็บเพจจากเว็บไซต์ใดๆแล้ว หลายโปรแกรมยังมีความสามารถอื่นๆ เช่น บริการส่งจดหมายอิเล็กทรอนิกส์ การค้นหาข้อมูล การโอนถ่ายโปรแกรมด้วย FTP เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 โฮมเพจ

โฮมเพจ (Home Page) จะหมายถึงหน้าแรกของเว็บเพจทั้งหมดที่ผู้ใช้บริการอินเทอร์เน็ตจะพบเมื่อมีการเข้าไปยังเว็บไซต์ใดๆ โฮมเพจเปรียบเสมือนกับสารบัญ และคำนำที่เจ้าของเว็บไซต์สร้างขึ้นเพื่อจะใช้ประชาสัมพันธ์องค์กรของตนว่าให้บริการในสิ่งใดบ้าง นอกจากนี้แล้วภายในโฮมเพจก็อาจมีเอกสารข้อความอื่นๆที่เชื่อมโยงต่อจากโฮมเพจนั้นๆได้อีก ที่เรียกว่า เว็บเพจ ซึ่งโฮมเพจหนึ่งๆก็อาจมีการเชื่อมกับเว็บเพจอื่นๆเป็นจำนวนมากก็ได้

2.8 IP Address

เนื่องจากในระบบเครือข่ายอินเทอร์เน็ตจะใช้โปรโตคอล TCP/IP เป็นมาตรฐานในการสื่อสารข้อมูลซึ่งจะมีการกำหนดหมายเลขประจำตัวที่ไม่ซ้ำกันให้กับเครื่องคอมพิวเตอร์ทุกเครื่องที่มีการเชื่อมต่อกันอยู่ในระบบเครือข่าย หมายเลขประจำตัวนี้จะถูกเรียกว่า IP Address หรือหมายเลข IP โดยมีรูปแบบเป็นชุดของตัวเลข 4 ชุดที่คั่นด้วยเครื่องหมายจุด เช่น 202.44.192.43 ตัวเลขในแต่ละชุดจะมีขนาด 8 บิต ดังนั้นแต่ละชุดจะมีค่าได้ตั้งแต่ 0 ถึง 255 เท่านั้น ซึ่งจะทำให้สามารถกำหนดหมายเลข IP ให้กับเครื่องคอมพิวเตอร์ได้ทั้งหมดถึง 4 พันล้านเลขหมายที่ไม่ซ้ำกันเลย

ดังนั้นเมื่อมีการติดตั้งเครื่องคอมพิวเตอร์ที่เป็น โฮสต์คอมพิวเตอร์เพื่อเชื่อมต่อเข้ากับอินเทอร์เน็ต และให้บริการต่างๆจำเป็นต้องขอเลขหมาย IP ประจำหน่วยงาน Internet Network Information Center (InterNIC) ขององค์กร Network Solution Incorporated (NSI) ที่รัฐเวอร์จิเนีย สหรัฐอเมริกา แต่ถ้าผู้ใช้สมัครเป็นสมาชิกกับหน่วยงานที่ทำหน้าที่เป็นผู้ให้บริการอินเทอร์เน็ต (Internet Service Provider : ISP) (ในประเทศไทยมีอยู่หลายหน่วยงาน) ก็ไม่ต้องขอหมายเลข IP เนื่องจาก ISP จะเป็นผู้ส่งหมายเลข IP ให้แก่ผู้ใช้เอง

2.9 โพรโทคอลที่ซีพี / ไอพี (TCP / IP Protocol)

ในหัวข้อนี้จะกล่าวถึงลักษณะการทำงานของอินเทอร์เน็ต ซึ่งประกอบไปด้วยเรื่องเกี่ยวกับ TCP / IP, การกำหนดชื่อ และ เลข IP อินเทอร์เน็ตนับได้ว่าเป็นเครือข่ายที่เปิดโอกาสให้เครือข่ายคอมพิวเตอร์อื่น ๆ เชื่อมโยงเข้ามาใช้งาน หรือเป็นศูนย์กลางเชื่อมโยงเครือข่ายคอมพิวเตอร์อื่น ๆ อีก แต่ปัญหาที่เกิดขึ้นในการเชื่อมโยงเครือข่ายคอมพิวเตอร์เข้าด้วยกันคือ แต่ละเครือข่ายใช้คอมพิวเตอร์ต่างชนิด ต่างยี่ห้อ และต่างระบบปฏิบัติการ มาตรฐานของ TCP / IP จึงถูกใช้เป็นกุญแจสำคัญในการแก้ปัญหาเหล่านี้ โดยจะกลายเป็นระบบเปิดที่สมบูรณ์และที่มีการเชื่อมโยงคอมพิวเตอร์ได้ตั้งแต่พีซีจนถึงเมนเฟรม และไม่จำกัดระบบปฏิบัติการที่ใช้ TCP / IP จึงเป็นมาตรฐานที่ทั่วโลกยอมรับ มีอุปกรณ์และซอฟต์แวร์ผลิตออกมาสนับสนุน TCP/IP มากมาย ดังนั้นจึงนับได้ว่า TCP / IP เป็นหัวใจของอินเทอร์เน็ตเลยทีเดียว

TCP / IP คืออะไร

TCP / IP เป็นข้อกำหนดเกี่ยวกับรูปแบบการเชื่อมโยงในเครือข่าย (Networking Protocol) จัดทำขึ้นเพื่อใช้เป็นเกณฑ์ให้เครื่องคอมพิวเตอร์ใช้งานร่วมกัน ในลักษณะของระบบเปิด (Open System) คือไม่ว่าจะเป็นคอมพิวเตอร์ชนิดใดหรือระบบใดก็ตาม จะสามารถติดต่อสื่อสารและแลกเปลี่ยนข้อมูลกันได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น เครื่องคอมพิวเตอร์ของ Digital Equipment ซึ่งเป็นระบบ Mini ติดต่อสื่อสารกับ Compaq ซึ่งเป็นเครื่อง PC ได้เมื่อดำเนินการด้วย TCP / IP

TCP / IP เป็นการกำหนดรูปแบบการสื่อสารระหว่างซอฟต์แวร์ การจัดการโอนย้ายข้อมูล การแสดงสถานะของเครื่องคอมพิวเตอร์ที่อยู่บนเครือข่าย ตลอดจนกฎระเบียบต่าง ๆ ที่กำหนดให้ทำเมื่อเกิดความผิดพลาดหรือต้อง ทำเพื่อป้องกันไม่ให้เกิดความผิดพลาด

TCP / IP เกิดจากการนำข้อกำหนดของรูปแบบต่างๆ กันมาใช้ร่วมกัน TCP และ IP ต่างก็เป็นรูปแบบหนึ่งของชุดข้อกำหนดนี้ แต่เรียกชุดข้อกำหนดรูปแบบนี้ว่า TCP / IP ถูกออกแบบมาเพื่อใช้รับส่งหรือโอนย้ายข้อมูลระหว่างคอมพิวเตอร์ที่อยู่บนระบบเครือข่ายเดียวกัน หรือต่างเครือข่ายกันก็ได้และมีการจัดเตรียมข้อมูลสถานะของเครือข่ายขึ้น ได้ภายในดังข้อกำหนดรูปแบบเอง ในการสร้างซอฟต์แวร์ของระบบเครือข่ายจะใช้ TCP / IP เป็นส่วนสนับสนุนได้ทั้งระบบเครือข่ายเฉพาะบริเวณ (Local Area Network) และเครือข่ายบริเวณกว้าง (Wide Area Network) ไม่ได้ใช้งานเฉพาะกับอินเทอร์เน็ตเท่านั้น

2.9.1 ส่วนประกอบของ TCP / IP

จากที่กล่าวมาแล้วข้างต้นว่า TCP / IP ประกอบไปด้วยชุดข้อกำหนดรูปแบบต่าง ๆ ซึ่งแบ่งเป็นกลุ่มได้ดังนี้

2.9.1.1 กลุ่มข้อกำหนดรูปแบบการขนส่ง (Transport Protocols)

ทำหน้าที่ควบคุมการเคลื่อนย้ายข้อมูลระหว่างคอมพิวเตอร์สองเครื่อง แบ่งย่อยออกได้เป็นสองชนิดคือ

ก. TCP (Transmission Control Protocols) เป็นการบริการแบบ Connection Based Service ซึ่งคอมพิวเตอร์ด้านผู้รับและผู้ส่งต้องต่อถึงกันอยู่ตลอดเวลาในระหว่างการสื่อสาร ถ้าเปรียบเทียบกับคล้ายกับระบบโทรศัพท์ที่ต้องติดต่อกันให้ ได้ก่อนจะพูดคุยกันได้

ข. UDP (User Datagram Protocol) เป็นการให้บริการแบบ Connections Service คอมพิวเตอร์ด้านผู้ส่งไม่จำเป็นต้องติดต่อกับด้านผู้รับก่อน เพียงรู้ที่อยู่ของด้านผู้รับแล้วใส่ที่อยู่นั้นไปกับข้อมูลที่ส่งออก ข้อมูลจะเดินทางตามเส้นทางต่าง ๆ เพื่อไปถึงปลายทางตามที่อยู่คล้ายกับการส่งจดหมายที่ไปรษณีย์จะส่งให้ตามที่อยู่ที่อยู่ปลายทาง โดยผู้ส่งและผู้รับไม่ต้องติดต่อกัน

2.9.1.2 กลุ่มข้อกำหนดเกี่ยวกับรูปแบบเส้นทาง (Routing Protocol)

ทำหน้าที่พิจารณาเส้นทางที่ดีที่สุดที่ใช้ส่งข้อมูลและถ้ามีข้อมูลเป็นจำนวนมากหรือมีขนาดใหญ่ กลุ่มข้อกำหนดรูปแบบนี้ก็จะทำการแบ่งย่อยข้อมูลให้มีขนาดเหมาะสม แล้วส่งออกไป เมื่อถึงผู้รับปลายทาง กลุ่มข้อกำหนดนี้ก็จะทำหน้าที่ตรงข้าม คือรวบรวมข้อมูลย่อยให้ถูกต้องก่อนการแสดงผลกลุ่มข้อกำหนดรูปแบบกลุ่มนี้ประกอบด้วย

ก. IP (Internet Protocol) เป็นข้อกำหนดรูปแบบกลุ่มนี้ประกอบด้วย

ข. ICMP (Internet Control Message Protocol) เป็นข้อกำหนดรูปแบบของข้อมูลข่าวสารเกี่ยวกับสถานะของ IP เช่น ข่าวสารความผิดพลาดและผลกระทบต่อเส้นทางเมื่อมีการเปลี่ยนแปลงฮาร์ดแวร์ในเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก. RIT (Routing Information Protocol) ข้อกำหนดรูปแบบหนึ่งที่ใช้สำหรับการพิจารณาวิธีการเลือกเส้นทางเพื่อให้ได้เส้นทางที่ดีที่สุดเหมาะสมกับข้อมูลมากที่สุด

ง. PE (Open Shortest Path First) ข้อกำหนดรูปแบบอีกประเภทหนึ่ง ที่ใช้ตัดสินเลือกเส้นทางโดยพิจารณาจากเส้นทางที่สั้นที่สุดก่อน

2.9.1.3 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับที่อยู่เครือข่าย (Network Address)

ทำหน้าที่พิจารณาที่อยู่ของเครือข่ายและเครื่องคอมพิวเตอร์ไม่ว่าจะเป็นลักษณะตัวเลขหรือชื่อก็ตาม เพื่อความถูกต้องของข้อมูลที่จะไปยังผู้รับปลายทาง โดยที่ไม่ว่าเครือข่ายจะใหญ่โตสักเพียงใดหรือมีเครื่องคอมพิวเตอร์จำนวนมากก็ตามที่อยู่จะต้องไม่ซ้ำกัน กลุ่มข้อกำหนดรูปแบบกลุ่มนี้มีดังนี้

ก. ARP(Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เพื่อไม่ให้เกิดที่อยู่ซ้ำกัน

ข. DNS (Domain Name System) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เมื่อรู้ชื่อของเครือข่ายหรือเครื่องคอมพิวเตอร์ เพราะในการใช้งานจริงนั้นใช้เพียงที่อยู่ที่เป็นตัวเลข แต่ระบบชื่อจัดทำขึ้นเพื่อให้สะดวกต่อการใช้งานของผู้ใช้

ค. RARP (Reverse Address Resolution Protocol) ข้อกำหนดรูปแบบที่พิจารณาตัวเลขที่อยู่เช่นเดียวกับ ARP แต่จะทำตรงกันข้ามกับ ARP

2.9.1.4 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับเส้นทางการสื่อสารระหว่างเครือข่าย (Gateway Protocols)

สนับสนุนข้อมูลสถานะเพื่อนำไปใช้เลือกเส้นทางที่เหมาะสม ข้อกำหนดรูปแบบเหล่านี้ประกอบด้วย

ก. EGP (Exterior Gateway Protocol) ข้อกำหนดรูปแบบนี้จะทำการถ่ายโอน ข้อมูลเส้นทางกันระหว่าง Gateway กับเครือข่ายภายนอกเพื่อทำการสื่อสาร

ข. GGKP (Gateway – to – Gateway Protocol) เป็นข้อกำหนดรูปแบบที่ทำงานถ่ายโอนข้อมูลเส้นทางกันระหว่าง Gateway Protocol

ค. IGP (Interior Gateway Protocol) ข้อกำหนดรูปแบบที่ถ่ายโอนข้อมูลเส้นทางกันภายในเครือข่ายเดียวกัน

Gateway เป็นอุปกรณ์หรือเครื่องคอมพิวเตอร์ที่ทำหน้าที่เสมือนประตูสื่อสารเป็นช่องสัญญาณเข้าหรือออกไปยังระบบสื่อสารอื่น หรือภายในเครือข่ายเดียวกันติดต่อกัน

2.9.1.5 กลุ่มข้อกำหนดรูปแบบเกี่ยวกับการบริการผู้ใช้ (User Services)

ผู้ใช้สามารถใช้ข้อกำหนดรูปแบบได้โดยตรง ข้อกำหนดรูปแบบนี้ประกอบด้วย

ก. BOOTP (BOOT Protocol) เมื่อผู้ใช้เปิดเครื่องคอมพิวเตอร์บนเครือข่ายให้เริ่มทำงาน ข้อกำหนดรูปแบบนี้จะอ่านโปรแกรมควบคุมการทำงานจากคอมพิวเตอร์ให้บริการ (Server Computer) มาให้

ข. FTP(File Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนไฟล์ข้อมูลระหว่างเครื่องคอมพิวเตอร์ ซึ่งอาจจะอยู่บนเครือข่ายเดียวกันหรือต่างเครือข่ายกันก็ได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก. TELNET เป็นข้อกำหนดรูปแบบที่ให้บริการเกี่ยวกับการควบคุมการติดต่อระยะทางไกล (Remote Login)

กลุ่มข้อกำหนดรูปแบบอื่น ที่นอกเหนือจากกลุ่มที่จัดไว้และบริการที่สำคัญ ๆ จัดทำไว้ให้บนระบบเครือข่ายที่สนใจมีดังนี้

- NFS (Network File System) เป็นข้อกำหนดรูปแบบที่ทำให้ผู้ใช้คอมพิวเตอร์เครื่องหนึ่งสามารถเข้าไปดูไฟล์ข้อมูลและใช้งานไฟล์ข้อมูล ซึ่งอยู่ในคอมพิวเตอร์เครื่องอื่นได้
- NIS (Network Information Services) เป็นข้อกำหนดรูปแบบที่ให้บริการกับ User Accounts ข้ามเครือข่าย เช่น Logins และ Password
- RPC (Remote Procedure Call) ข้อกำหนดรูปแบบที่อำนวยความสะดวกให้กับโปรแกรมประยุกต์ที่ใช้งานกับการควบคุมระยะทางไกล
- SMTP(Simple Mail Transfer Protocol) ข้อกำหนดรูปแบบที่ให้บริการถ่ายโอนจดหมายอิเล็กทรอนิกส์ (Electronic Mail) ระหว่างคอมพิวเตอร์

2.9.2 สถาปัตยกรรม TCP/IP

TCP /IP ออกแบบเป็นชุดกำหนดรูปแบบ แบ่งการบริหารออกเป็นกลุ่ม ๆ ภายในชุดข้อกำหนดรูปแบบเช่น กลุ่มผู้ใช้บริการ กลุ่มการขนส่ง และกลุ่มเกี่ยวกับเครือข่าย ดังกล่าวมาแล้ว จากกลุ่มต่างๆ ที่แบ่งไว้นำไปพัฒนาเป็นสถาปัตยกรรมซึ่งมีลักษณะเป็นระดับชั้น (Layer) ตามกลุ่มการบริการ

สถาปัตยกรรมแบบระดับชั้น มีข้อดีอยู่หลายประการ แต่ละชั้นมีอิสระไม่ขึ้นต่อกันทำให้การเปลี่ยนแปลงการบริการของชั้นใด ๆ ไม่ก่อปัญหาต่อบริการชั้นอื่น การเพิ่มเติมการบริการใหม่ทำได้โดยไม่ต้องเปลี่ยนแปลงโปรแกรมระบบเดิมและสิ่งสำคัญก็คือการทำงานระดับชั้นนั้นทำให้ตัวโปรแกรมมีขนาดเล็ก สามารถระบุส่วนที่จะต้องปรับปรุงได้แน่นอน ไม่ต้องวิตกกังวลถึงโปรแกรมส่วนอื่นทำให้การพัฒนาประสิทธิภาพของระบบทำได้ง่ายและดียิ่งขึ้น

ชุดข้อกำหนดรูปแบบ TCP/IP เมื่อนำไปใช้งานจะอยู่ระหว่างระดับชั้นที่เป็นฮาร์ดแวร์กับระดับชั้นที่เป็นซอฟต์แวร์อื่น เมื่อติดตั้งกับบางระบบ

TCP /IP เมื่อนำไปใช้งานกับระบบ Windows ต้องเพิ่ม Drivers พิเศษเข้าไปด้วยเพื่อให้ Windows รู้จักข่าวสารที่เป็นแพ็กเกจ (Packet) และส่งแพ็กเกจเข้าสู่ระบบเครือข่ายได้ แต่ถ้าเป็นซอฟต์แวร์ที่ใช้ชุดข้อกำหนดรูปแบบ TCP /IP อยู่แล้วข่าวสารจากโปรแกรมประยุกต์จะผ่านมายังระดับชั้น TCP /IP ทำการประมวลส่งเข้าเครือข่ายโดยให้ระบบปฏิบัติการทำการส่ง กำหนดชื่อและเลข IP

เครือข่ายอินเทอร์เน็ตใช้รหัสหมายเลขมากำหนดให้แต่ละเครือข่ายและเครื่องคอมพิวเตอร์ภายในเครือข่ายที่เชื่อมโยง เรียกรหัสหมายเลขเหล่านี้ว่า อินเทอร์เน็ตแอดเดรส (Internet Address) หรือนิยมเรียกกันทั่วไปว่า IP แอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP แอดเดรส ประกอบด้วย เลขฐานสองจำนวน 32 บิตแบ่งออกเป็น 4 ส่วน แต่ละส่วนมี 8 บิต เมื่อคูณเฉพาะแต่ละส่วนเป็นฐานสิบจะได้เลขจำนวน 256 ค่าไม่ซ้ำกัน (0 – 256) IP แอดเดรส จะนำเอา หมายเลขทั้ง 4 ส่วนมารวมกัน โดยแยกแต่ละส่วนด้วยจุด ดังนั้นหมายเลขทั้งหมดที่เป็นไปได้ โดยไม่ซ้ำกัน คือ 256^4 หรือ 4294967296 จำนวนมีค่าหมายเลขจาก 000.000.000.000 จนถึง 255.255.255.255 หมายเลขเหล่านี้เองที่อินเทอร์เน็ต ใช้กำหนดให้กับเครือข่าย และเครื่องคอมพิวเตอร์เพื่อใช้อ้างอิงถึง IP แอดเดรส บางหมายเลขสงวนไว้ใช้ด้วยจุดหมายกรณีพิเศษ ทำให้ IP แอดเดรส ที่ใช้งานทั่วไป ลดลง จากจำนวนที่เป็นไปได้ ความหมายของ IP แอดเดรส จะแบ่งได้เป็น 2 กลุ่มดังนี้

- กลุ่มที่ใช้เป็นรหัสประจำเครือข่าย
- กลุ่มที่ใช้เป็นรหัสประจำเครื่องคอมพิวเตอร์ที่อยู่ภายในเครือข่าย (Host Computers)

IP แอดเดรส ในกลุ่มรหัสประจำเครื่องคอมพิวเตอร์ สามารถซ้ำกันได้ แต่ละกลุ่มรหัสประจำเครือข่ายจะซ้ำกัน ไม่ได้ ดังนั้นรหัสเครื่องที่ซ้ำกันจึงไม่มีผลต่อการอ้างอิงถึง

นอกจากนี้เพื่อความเหมาะสมในการกำหนด IP แอดเดรสให้กับผู้ขอ ทางผู้บริหารอินเทอร์เน็ต แบ่งคลาสของผู้ขอ IP แอดเดรสตามขนาดของเครือข่าย เพื่อให้ทรัพยากรส่วนนี้ถูกใช้อย่างคุ้มค่าที่สุด องค์การขนาดใหญ่จะจัดให้อยู่ในคลาสที่สามารถกำหนด IP แอดเดรสให้กับเครื่องคอมพิวเตอร์ในเครือข่ายได้มาก

การกำหนดหลายเลขของเครือข่ายและเครื่องคอมพิวเตอร์ นอกจากจะแบ่ง IP แอดเดรสเป็นคลาส ทั้ง 5 ประเภทแล้ว ยังมีข้อกำหนดปลีกย่อยอีกหลายประการที่ผู้วางระบบต้องรู้ไว้ เพื่อที่จะกำหนดแอดเดรสใช้งาน ได้อย่างถูกต้องไม่ผิดพลาด

2.10 ดีเอ็นเอส (Domain Name System : DNS)

การใช้หมายเลข IP ซึ่งเป็นตัวเลขสั้นๆ ในการอ้างอิงถึงเครื่องคอมพิวเตอร์แต่ละเครื่องนั้น จะมีข้อเสียคือ จำยากและก่อให้เกิดความสับสนได้ง่าย จึงมีการพัฒนาวิธีการอ้างอิงถึงหมายเลข IP แบบใหม่ ที่เรียกว่า DNS ขึ้นมา DNS เป็นเทคนิคการเปลี่ยนหมายเลข IP ที่เป็นตัวเลขให้เป็นตัวอักษรแทน

2.11 HTML

โฮมเพจและเว็บเพจใดๆ โดยส่วนใหญ่จะถูกสร้างขึ้นมาจากภาษาที่เรียกว่า HTML (Hyper Text Markup Language) หรือ "ภาษาสำหรับทำเครื่องหมาย" ประกอบด้วยคำสั่ง (Tags) ที่ใช้ในการกำหนดว่าเว็บเพจจะมีข้อความอะไร มีการแสดงรูปภาพ เสียง และภาพวิดีโอที่ตำแหน่งใด นอกจากนี้ ยังมีคำสั่งสำหรับการเชื่อมโยงเว็บเพจหนึ่งไปยังอีกเว็บเพจหนึ่งและไปยังบริการอื่นๆ ในอินเทอร์เน็ตอีกด้วย

การสร้างเว็บเพจนอกจากจะสร้างโดยการเขียน HTML บนโปรแกรม Editor ตัวใดตัวหนึ่ง เช่น โปรแกรม Notepad บนวินโดวส์แล้ว ยังสามารถใช้โปรแกรมช่วยสร้างเว็บเพจ กำหนดรูปร่างหน้าตาของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เว็บเพจได้โดยตรง โดยที่ผู้สร้างไม่จำเป็นต้องเขียนหรือเรียนรู้ HTML แต่อย่างใด ซึ่งในปัจจุบันจะมีโปรแกรมช่วยสร้างเว็บเพจหลายตัว เช่น โปรแกรม Microsoft Frontpage โปรแกรม Netscape Navigator Gold เป็นต้น

2.12 เอเอสพี (Active Sever Page : ASP)

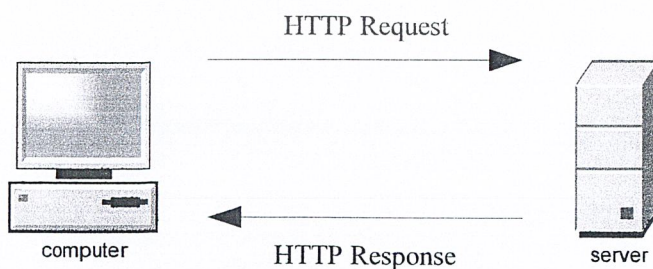
เอเอสพี เป็นคำที่ย่อมาจาก Active Sever Pages ซึ่งเป็นโปรแกรมที่พัฒนาขึ้นโดยบริษัท ไมโครซอฟท์ เพื่อใช้งานทางด้านอินเทอร์เน็ต โดย เอเอสพี จะทำหน้าที่ตีความเอกสารที่เขียนด้วยภาษา สคริปต์ เช่น VBสคริปต์ โดยที่มี เอเอสพี tag (คือคำสั่งที่มีเครื่องหมาย <%> กำกับอยู่ซึ่งบราวเซอร์ทั่วไป เช่น Netscape Navigator หรือ Internet Explorer ไม่สามารถนำไปแสดงผล จากนั้นจึงสร้างเอกสารผลลัพธ์เป็นเอกสาร HTML อันเป็นเอกสารที่ประกอบด้วย HTML tag ต่างๆ (คือ คำสั่งที่มีเครื่องหมาย < >) กำกับอยู่ซึ่งบราวเซอร์ทั่วไปสามารถนำไปสร้างเป็นเว็บเพจขึ้นเพื่อใช้แสดงผลได้

การทำงานของโปรแกรม เอเอสพี จะเกิดขึ้นเฉพาะทางฝั่งเซิร์ฟเวอร์เท่านั้น เราจึงเรียกว่า เป็นการทำงานแบบ เซิร์ฟเวอร์ไซด์ จากนั้นผลลัพธ์ที่ได้จะถูกส่งไปให้เว็บเซิร์ฟเวอร์ แล้วเว็บเซิร์ฟเวอร์ก็จะส่งเอกสารดังกล่าวต่อไปยังบราวเซอร์อีกทีหนึ่ง เมื่อบราวเซอร์ได้รับเอกสารนั้นแล้ว บราวเซอร์ก็จะสามารถแสดงผลได้ถูกต้องครบถ้วน โดยการทำงานของบราวเซอร์ทางฝั่งของผู้ใช้ เราเรียกว่าเป็นการทำงานแบบ ไคลเอนไซด์ (client side)

2.12.1 ขบวนการทำงานของ เอเอสพี

เพื่อ ได้ศึกษา เอเอสพี ในระดับเบื้องต้นก่อนอื่นขอให้ลองทำความเข้าใจพื้นฐานขบวนการทำงานของ เอเอสพี (เอเอสพี model)

การทำงานทั้งหมดจะเริ่มจาก บราวเซอร์ร้องขอเอกสาร HTML ไปยังเว็บเซิร์ฟเวอร์ผ่านทาง HTTP (HTTP request) โดยที่เอกสารที่ขอไปจะเป็นแฟ้มข้อมูลที่มีนามสกุลเป็น ASP (เช่น search.ASP ฯลฯ)เมื่อเว็บเซิร์ฟเวอร์ได้รับการร้องขอดังกล่าว ก็จะส่งเอกสารนั้นไปให้ เอเอสพีตีความ จากนั้น เอเอสพี ก็จะสร้างเอกสาร HTML ส่งกลับไปที่เซิร์ฟเวอร์เพื่อส่งต่อไปยังบราวเซอร์และใช้แสดงผลทางฝั่งผู้ใช้ต่อไป(HTTP response) ซึ่งการทำงานของ เอเอสพีนี้แทบไม่แตกต่างไปจากหลักการทำงานโปรแกรม CGI(Common Gateway Interface) จนอาจกล่าวได้ว่า เอเอสพี ก็เป็นโปรแกรม CGI ประเภทหนึ่งเช่นกัน



รูปที่ 2.4 แสดงการทำงานของ ASP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

เอกสารเอเอสพี 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 ก็จะไม่เกิดการเปลี่ยนแปลงใดๆในเอกสารนั้นเลย แต่หากว่าส่วนใดมี เอเอสพี tag กำกับ เอเอสพี ก็จะเปลี่ยนแปลงเอกสารส่วนดังกล่าวไปอยู่ในรูปข้อความทั่วไปหรือเป็น HTML tag แทน เช่น หากในเอกสารมีคำสั่งนี้

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

ก็จะถูกเปลี่ยนเป็น...

```
<br>Hello แล้วตามด้วยวันและเวลาปัจจุบัน
```

2.12.2 เอกสาร ASP

เอกสาร ASP คือ เท็กซ์ไฟล์ที่บรรจุเอาคำสั่งสคริปต์ต่างๆ (อาจเขียนด้วยภาษาเดียวหรือ หลายภาษาผสมกันก็ได้) แทรกอยู่กับแท็ก HTML และสคริปต์ที่ทำงานฝั่งไคลเอนต์ เมื่อบราวเซอร์เรียกใช้งานเอกสาร ASP เอกสาร ASP นั้น ก็จะถูกแปล โดย ASP Interpreter ที่เซิร์ฟเวอร์ และถูกส่งให้ทำงานที่เว็บเซิร์ฟเวอร์ โดยผลลัพธ์ที่ได้จากการรันแอปพลิเคชัน ASP จะถูกส่งมายังบราวเซอร์ที่เรียกใช้ในรูปแบบเอกสาร HTML

เอกสาร ASP ประกอบด้วย

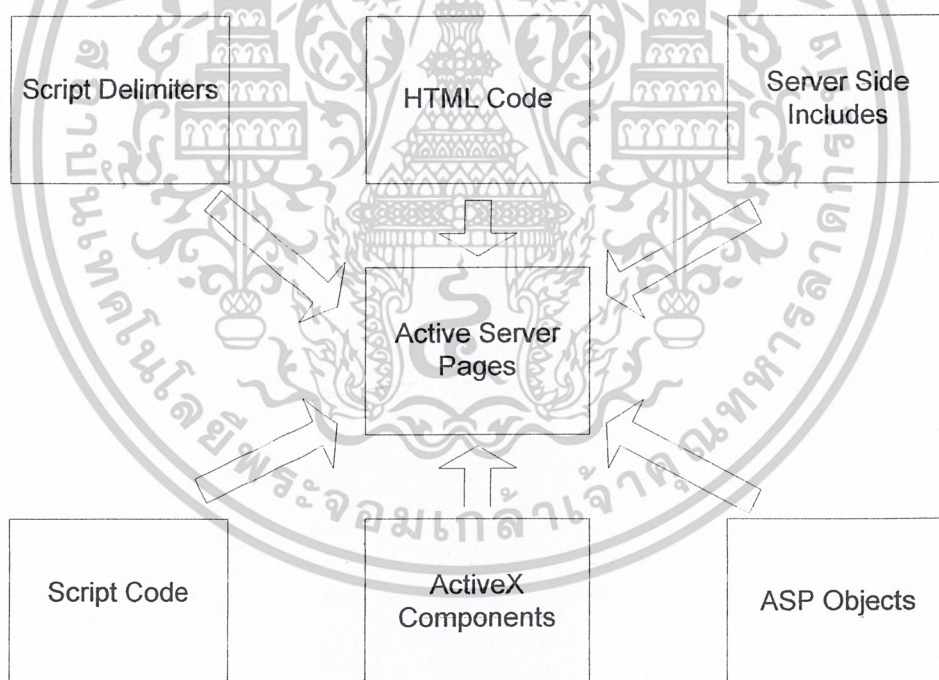
1. Server Side Include (อาจมีหรือไม่ก็ได้)
2. HTML Code
3. Script delimiters คือ เครื่องหมายที่แยกโค้ด ASP ออกจากโค้ด HTML
4. Script Code เช่น VBScript ,Jscript, JaVaScript เป็นต้น
5. ActiveX Components
6. ActiveX Objects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.12.3 ความต้องการของระบบในการใช้งาน ASP

ในการที่จะพัฒนาและใช้งานแอปพลิเคชัน ASP นั้น มีความต้องการด้าน Software และ Hardware ดังนี้

1. หน่วยประมวลผล (CPU) ควรจะเป็น Pentium ขึ้นไป
2. Ram อย่างต่ำ 32 เมกะไบต์
3. พื้นที่ว่างใน HardDisk อย่างน้อย 150 เมกะไบต์
4. ระบบปฏิบัติการ Windows NT Server 4.0 , Windows NT Workstation 4.0 หรือ Windows 95/98/ME และต้องมีการติดตั้ง โพรโทคอล TCP/IP ด้วย
5. โปรแกรม Web Server ที่สนับสนุน ASP เช่น Internet Information Server 3.0/4.0 สำหรับ Windows NT และ Windows 2000 หรือ Personal Web Server สำหรับ Windows 95/98/ME
6. ถ้ามีการพัฒนาระบบฐานข้อมูล เราต้องมีโปรแกรมฐานข้อมูลที่รองรับ ODBC ด้วย เช่น Microsoft Access หรือ Microsoft SQL Server เป็นต้น



รูปที่ 2.5 แสดงองค์ประกอบในเอกสาร ASP ไฟล์หนึ่งๆ

2.12.4 การแบ่งแยกโค้ด ASP ออกจากโค้ด HTML

เนื่องจากในเอกสาร ASP นั้น โค้ด ASP จะถูกแทรกอยู่กับโค้ด HTML การที่ตัวแปลโค้ด ASP (ASP Interpreter) ที่เว็บเซิร์ฟเวอร์จะสามารถแยกออกเป็นส่วนไหนเป็นโค้ด HTML และส่วนไหนที่จะเป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ด ASP ต้องอาศัยเครื่องหมายที่ใช้แบ่งแยก ที่เราเรียกว่า ASP Delimiters โดย ASP Delimiters มีอยู่ 3 รูปแบบ ซึ่งมีประโยชน์และความเหมาะสมในการใช้งานต่าง ๆ ดังนี้

1. เครื่องหมายที่ใช้กำหนดขอบเขตของโค้ด ASP ที่แทรกอยู่ตามโค้ด HTML

โค้ดที่เป็น ASP จะต้องอยู่ระหว่างเครื่องหมาย <% และ %> โดยมีรูปแบบการใช้งานดังนี้

```
<%
<โค้ด คำสั่งการทำงานของ ASP>
%>
```

2. เครื่องหมายที่ใช้แสดงค่าของนิพจน์หรือตัวแปร

สำหรับค่าที่ต้องการแสดงออกให้ผู้ใช้เห็น หรือต้องการให้ปรากฏในโค้ด HTML นั้น สามารถทำได้ง่ายๆ โดยการใส่ค่าตัวแปรหรือนิพจน์ที่ต้องการแสดงระหว่างเครื่องหมาย <%= และ %> โดยมีรูปแบบการใช้งานดังนี้

```
<%=<สิ่งที่ต้องการแสดง>%>
```

3. การกำหนดขอบเขตของ ASP โดยการระบุภาษาที่จะใช้

วิธีนี้จะใช้แอตทริบิวต์ RUNAT=SERVER แทรกในแท็ก <Script> เพื่อเป็นการบ่งบอกว่าสคริปต์ที่อยู่ระหว่างแท็ก <Script> และ </Script> ต่อไปนี้จะทำงานที่ฝั่งเซิร์ฟเวอร์เท่านั้น ซึ่งจะไม่แสดงโค้ดให้เห็นในฝั่งไคลเอนต์ด้วย

เนื่องจาก ASP นั้น อนุญาตให้เราสามารถเขียนสคริปต์คำสั่งได้หลากหลายภาษา (ซึ่งเราอาจเลือกใช้ภาษาใดภาษาหนึ่ง หรือใช้ผสมกันหลายภาษาก็ได้) ในกรณีที่เราไม่ได้ระบุภาษาที่เราจะใช้เขียน ASP จะถือว่าเราใช้ VB Script โดยปริยาย แต่ถ้าในกรณีที่เราต้องการใช้ภาษาอื่นที่ไม่ใช่ภาษา VB Script เราต้องระบุภาษาที่เราต้องการใช้แทรกเพิ่มเติม โดยแทรกแอตทริบิวต์ LANGUAGE = ภาษาที่เราจะใช้ภายในแท็ก <Script> ที่ใช้ในการแบ่งแยกโค้ด ASP ออกจากโค้ด HTML ด้วยรูปแบบการใช้งานดังนี้

```
< Script RUNAT = SERVER LANGUAGE =language>
< โค้ดคำสั่งการทำงาน >
</Script>
```

โดยที่ language หมายถึง ภาษาที่จะใช้เขียนสคริปต์ เช่น JavaScript, Jscript เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การคำนวณและการสร้าง



รูปที่ 3.1 Block diagram ของโครงการทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 หลักการทำงานของโครงการ

จากบล็อกไดอะแกรมแสดงการทำงานของโครงการที่เครื่องลูกข่าย (CLIENT) ผ่านอินเทอร์เน็ต โดยรับข้อมูลส่งงานจากบราวเซอร์ เช่น INTERNET EXPLORER ,NETSCAPE โดย เอเอสพี จะทำการติดต่อฐานข้อมูลผ่านทาง ODBC (Open Data Connectivity) โดยต้องสร้างส่วนในการติดต่อกับฐานข้อมูล DSN (Data Source Name) นำข้อมูลที่ได้ไปทำการตรวจและแก้ไขฐานข้อมูล

เครื่องลูกข่ายจะทำส่งสัญญาณร้องขอ (Request) เพื่อขอควบคุมอุปกรณ์ไฟฟ้าไปยังเซิร์ฟเวอร์ (Sever) เครื่องเซิร์ฟเวอร์ก็จะส่งหน้าที่ใช้ในการควบคุมอุปกรณ์ไฟฟ้าไปยังเครื่องลูกข่ายโดยเครื่องเซิร์ฟเวอร์จะทำการแปลงข้อมูลแบบ ASP ให้เป็น HTML ส่งไปให้เครื่องลูกข่าย

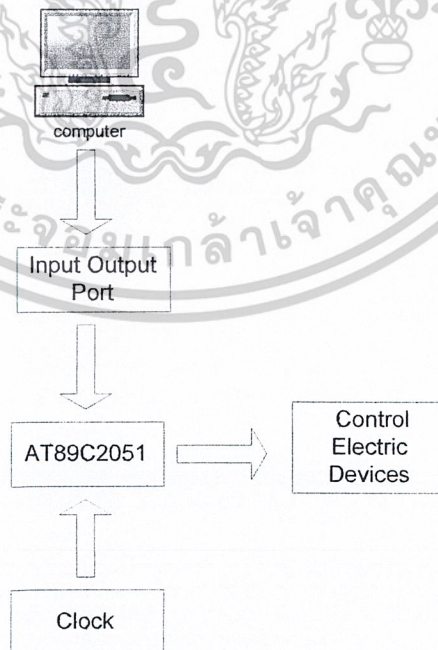
เมื่อมีการควบคุมอุปกรณ์ไฟฟ้า เครื่องลูกข่ายจะมีการส่งข้อมูลมาเก็บไว้ใน Database ที่เครื่องเซิร์ฟเวอร์

หลังจากนั้นเครื่องเซิร์ฟเวอร์ก็จะทำการนำข้อมูลการควบคุมอุปกรณ์ไฟฟ้าจากฐานข้อมูลส่งออกพอร์ทอนุกรม (Serial Port) ไปยังส่วนควบคุมอุปกรณ์ไฟฟ้า (Relay Board) ซึ่งจะควบคุมการทำงานโดยไมโครคอนโทรลเลอร์ MCS-51 ในโครงการนี้ใช้ IC AT89C2051

3.2 ส่วนของ hardware

วงจรเปิดปิดอุปกรณ์ไฟฟ้า(บอร์ดรีเลย์)

บอร์ดรีเลย์เป็นอุปกรณ์ที่รับสัญญาณจากพอร์ทอนุกรมของคอมพิวเตอร์และจะนำสัญญาณที่ได้ผ่านวงจรไฟฟ้าบนบอร์ดเพื่อไปควบคุมหน้าสัมผัสรีเลย์เพื่อเปิดปิดอุปกรณ์ไฟฟ้าที่ต้องการต่อไป



รูปที่ 3.2 Block Diagram ในส่วนระบบไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโครงงานนี้ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C2051 ซึ่งไมโครคอนโทรลเลอร์มีส่วนประกอบตาม Block Diagram ดังนี้

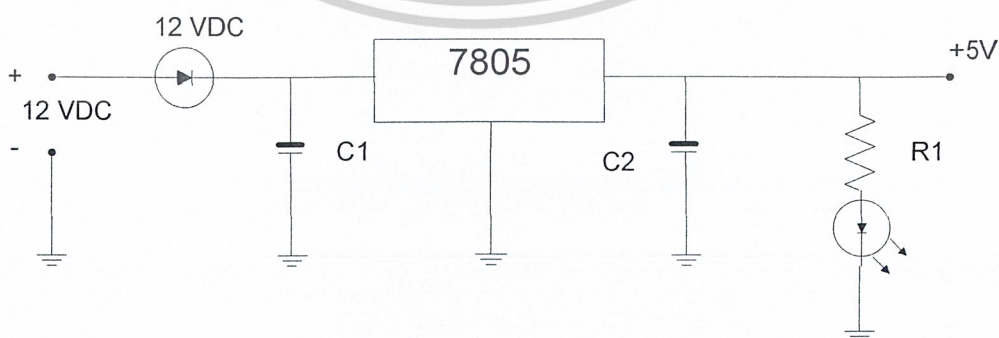
- ส่วนของไมโครคอนโทรลเลอร์เอง ซึ่งมีส่วนเชื่อมโยงต่อกัน ระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอกในตัวอยู่แล้ว
- ส่วนของโปรแกรม Memory ที่ใช้ในการเก็บข้อมูล ในตัวไมโครคอนโทรลเลอร์
- ส่วนของสัญญาณนาฬิกาซึ่งใช้เป็นตัวสร้างสัญญาณนาฬิกาให้ตัวไมโครคอนโทรลเลอร์ โดยตัวไมโครคอนโทรลเลอร์ จะเป็นตัวรับคำสั่งจากคอมพิวเตอร์เพื่อใช้ควบคุมเปิดปิดรีเลย์ ที่ต่ออยู่กับอุปกรณ์ไฟฟ้า

หลักการทำงานและรูปแบบวงจรของ (Relay Board)

จากรูป แสดงวงจรของบอร์ดรีเลย์ โดยเริ่มจากพอร์ต com2 ของเครื่องคอมพิวเตอร์ มาต่อเข้ากับบอร์ดรีเลย์ เมื่อโปรแกรมควบคุมได้รับคำสั่งจากผู้ใช้ โปรแกรมจะส่งข้อมูลจากคอมพิวเตอร์ไปเข้าบอร์ดรีเลย์โดยสัญญาณที่เข้ามาจะผ่าน IC MAX232 ก่อนเพื่อทำการเปลี่ยนระดับแรงดันที่ได้จาก พอร์ตอนุกรมของคอมพิวเตอร์ให้เป็นระดับแรงดันของ TTL กล่าวคือ ระดับแรงดันที่ได้มาจากคอมพิวเตอร์ซึ่งถูกส่งผ่านมาทางพอร์ตอนุกรมจะมีระดับแรงดันอยู่ที่ 3 โวลต์ ถึง 12 โวลต์ ซึ่งไม่สามารถนำมาต่อใช้โดยตรงกับ IC AT89C2051 ได้ดังนั้นจึงต้องใช้ IC MAX232 เพื่อเปลี่ยนระดับแรงดันเสียก่อน เมื่อผ่าน IC MAX232 มาแล้วจึงมาเข้า AT89C2051 ไมโครคอนโทรลเลอร์ขนาด 8 บิต ซึ่งจะประมวลผลตามโปรแกรมที่ได้โปรแกรมไว้ในตัว

3.2.1 ภาคจ่ายไฟ

ภาคจ่ายไฟจะได้มาจาก adapter ขนาด 12 โวลต์จากภายนอกโดยผ่าน ไดโอดตัวที่ 1 ซึ่งป้องกันการต่อแรงดันผิดขั้วและจะไปเข้ารีเลย์ทั้ง 8 ตัวและอีกส่วนหนึ่งจะผ่านเข้าไปยังเรกูเลเตอร์ เบอร์ 7805 ซึ่งทำหน้าที่ลดระดับแรงดันจาก 12 โวลต์ ให้เหลือ 5 โวลต์ ซึ่งแสดงดังรูป



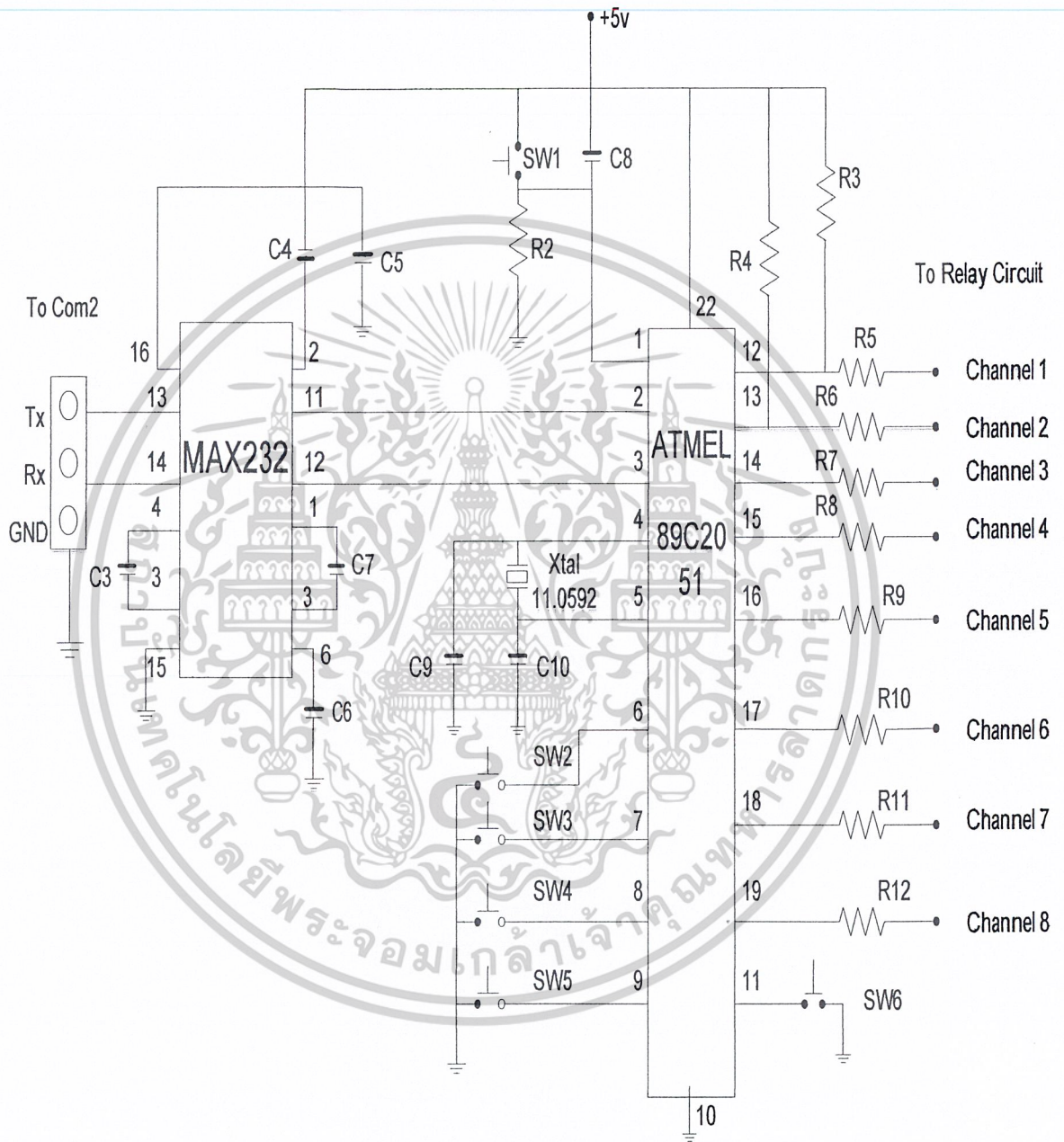
รูปที่ 3.3 แสดงวงจรภาคจ่ายไฟในบอร์ดรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 วงจร MAX 232 และ AT89C2051

จากวงจรภาคจ่ายไฟทำการจ่ายไฟเลี้ยงให้แก่ IC MAX 232 และ AT89C2051 โดยต่อตามวงจร

ดังรูป



รูปที่ 3.4 แสดงวงจรของบอร์ดรีเลย์ซึ่งใช้ควบคุมการเปิดปิดอุปกรณ์ไฟฟ้า

เมื่อทำการประมวลผลแล้วจะส่งผลออกมา 8 บิต ซึ่งแต่ละบิตจะต่ออยู่กับวงจรขั้วรีเลย์ 1 ชุด

ดังนั้นเราจึงสามารถควบคุมรีเลย์แต่ละตัวได้อย่างอิสระต่อกัน

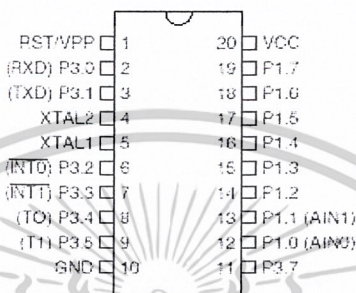
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 IC AT89C2051

IC AT89C2051 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต ผลิตโดย บริษัท ATMEL มีขนาดเล็กเพียง 20 ขาเท่านั้นแรงดันไฟเลี้ยงสามารถใช้ได้ตั้งแต่ 2.7 – 6.0 โวลต์ มีหน่วยความจำแฟลชขนาด 2 กิโลไบต์ ขนาดหน่วยความจำแรม 128 ไบต์สามารถขับ LED ได้โดยตรงมีโหมดประหยัดพลังงานทำให้กินกระแสเพียงน้อยลงมีตำแหน่งขา ดังรูป

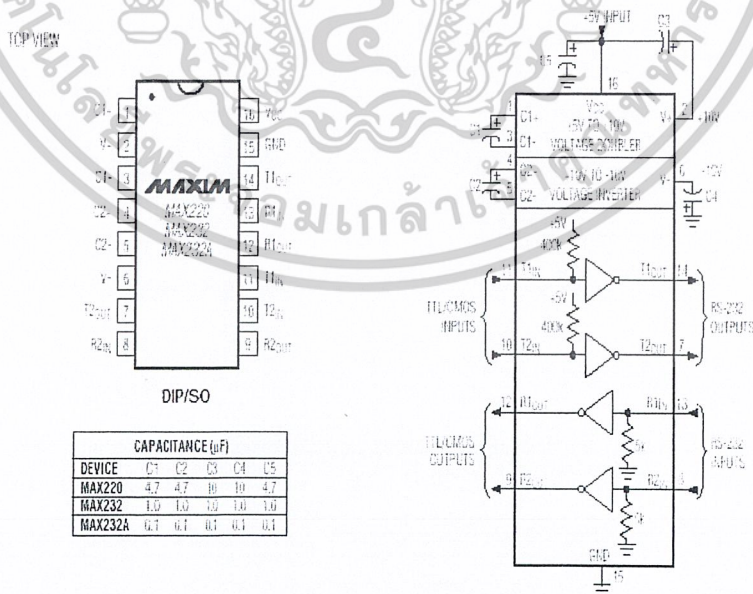
PDIP/SOIC



รูปที่ 3.5 แสดงตำแหน่งขาของ IC AT89C2051

3.2.4 IC MAX 232

เป็นไอซีที่รองรับการส่งข้อมูลแบบอนุกรมตามมาตรฐาน RS – 232C และแปลงไปเป็นระดับแรงดันของ TTL หรือ CMOS มีขนาด 16 ขา ต้องการไฟเลี้ยงขนาด 5 โวลต์มี Voltage Doubler และ Voltage Inverter กินไฟน้อยมี Driver และ Receiver อย่างละ 2 ชุด แสดงตำแหน่งขาและ Function Diagram ดังรูป

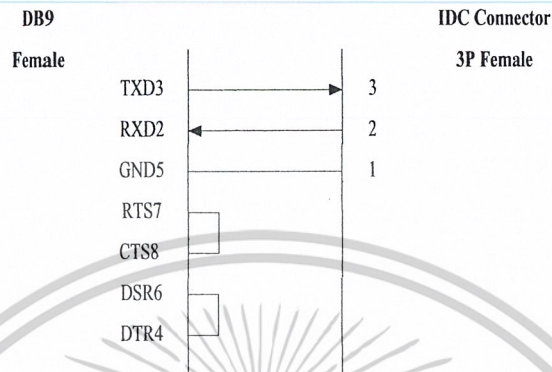


รูปที่ 3.6 แสดงตำแหน่งขาและ Functional Diagram ของ MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 การเชื่อมต่อผ่านทางพอร์ตอนุกรม

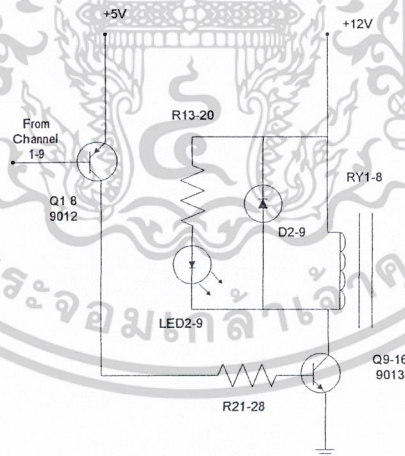
ในการนำรีเลย์บอร์ดไปใช้งานเราจะต่อรีเลย์บอร์ดด้วยสายนำสัญญาณที่มีหัวต่อแบบ DB9 เพื่อนำไปต่อกับพอร์ต Com2 ของคอมพิวเตอร์ส่วนด้านที่ลงบอร์ดจะเป็น Connector 3 ขา แสดงดังรูป



รูปที่ 3.7 แสดงการต่อสายระหว่างพอร์ตอนุกรม DB9 กับบอร์ดรีเลย์

3.2.6 วงจรขั้วรีเลย์ (Relay)

จากไมโครคอนโทรลเลอร์ (AT89C2051) เมื่อมีการสั่งงานเปิดปิดเครื่องใช้ไฟฟ้าสัญญาณจะส่งออกจากไมโครคอนโทรลเลอร์ตาม channel 1-8 ซึ่งแต่ละ channel ต่อกับวงจรขั้วรีเลย์ ดังรูป



รูปที่ 3.8 แสดงวงจรขั้วรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ส่วนของ software

โปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเทอร์เน็ต

โปรแกรมควบคุมผ่านทางเครือข่ายอินเทอร์เน็ตเป็น โปรแกรมที่สำคัญที่สุดในระบบนี้ซึ่งจะการติดต่อผ่านทางโปรโตคอล HTTP- Hypertext Transfer Protocol ซึ่งเป็นมาตรฐานในการติดต่อสื่อสารของเอกสาร HTML โดยรายละเอียดของการเชื่อมต่อโดยใช้ โปรโตคอล HTTP มีดังต่อไปนี้

บทบาทของโปรโตคอล

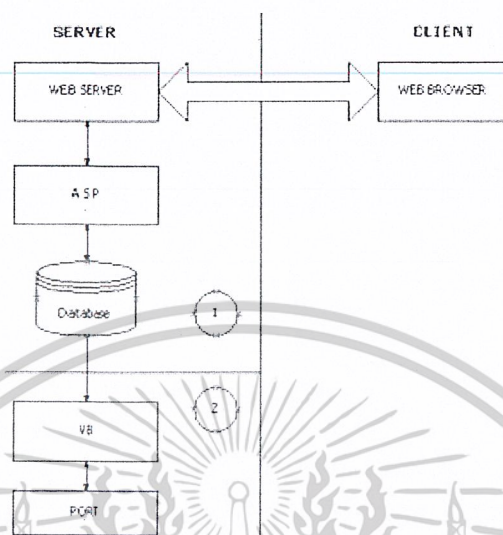
ในการเชื่อมต่อเพื่อรับ-ส่งข้อมูลเพื่อใช้ควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน นั้นต้องมีการสื่อสารข้อมูล ผ่านอินเทอร์เน็ต ซึ่งสิ่งที่สำคัญอย่างหนึ่งคือโปรโตคอล ซึ่งเปรียบเหมือนระบบที่เป็นข้อตกลงว่าจะมีการติดต่อ และรับส่งข้อมูลผ่านทางพอร์ตขนานกันอย่างไรระหว่างไคลเอนต์กับเซิร์ฟเวอร์ซึ่งเป็นคอมพิวเตอร์ตัวแม่และตัวลูก ซึ่งที่ใช้โปรโตคอลที่ใช้ในการทำอุปกรณ์ควบคุมอุปกรณ์ไฟฟ้าภายในบ้านผ่านอินเทอร์เน็ตคือ HTTP ซึ่งโปรโตคอล HTTP ซึ่งอยู่บนพื้นฐานการติดต่อแบบไคลเอนต์/เซิร์ฟเวอร์ ที่ต้องมีการร้องขอและตอบสนอง โดยโปรโตคอลนี้ อาศัยการเชื่อมต่อผ่านทางโปรโตคอล TCP/IP อีกทีหนึ่งโดยใช้พอร์ตหมายเลข 80 เป็นช่องทาง ในการติดต่อ ในทางปฏิบัติจะใช้พอร์ตหมายเลขอื่นก็ได้ แต่จะทำให้เกิดความลำบากแก่ผู้ใช้ ที่ต้องระบุหมายเลข พอร์ตลงใน URL ด้วย

ขั้นตอนการทำงานโดยคร่าวของโปรโตคอล HTTP มีดังต่อไปนี้

- 1.สร้างการเชื่อมต่อกับเซิร์ฟเวอร์ผ่านชื่อเกิด จากไคลเอนต์มาเซิร์ฟเวอร์
- 2.เซิร์ฟเวอร์จะตอบกลับยังไคลเอนต์ว่าการเชื่อมต่อสำเร็จหรือไม่
- 3.ไคลเอนต์จะส่งคำร้องขอข้อมูลไปยังเซิร์ฟเวอร์ และเซิร์ฟเวอร์จะทำการไปหาข้อมูลที่ไคลเอนต์ต้องการ
- 4.ข้อมูลจากตอบสนองจะถูกส่งจากเซิร์ฟเวอร์มาไคลเอนต์
- 5.การเชื่อมต่อจะสิ้นสุดลง

การเชื่อมต่อของโปรโตคอล HTTP นั้นมีข้อดีคือ โปรโตคอล HTTP มีการเชื่อมต่อในระยะเวลาสั้นๆ หรือที่เรียกว่าเป็นโปรโตคอลแบบ connectionless ข้อมูลจะถูกส่งเป็นชุดๆ และส่งข้อมูลเมื่อมีการร้องขอ และ จะตัดการติดต่อเมื่อไคลเอนต์ได้ข้อมูลที่ต้องการครบ ไม่ได้ติดต่อกันตลอดเวลา ซึ่งทำให้รับรองไคลเอนต์ จำนวนมากในเวลาเดียวกัน

หลักการทํางานของโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเทอร์เน็ต



รูปที่ 3.9 แสดงการทํางานของโปรแกรมควบคุมการเปิดปิดอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเทอร์เน็ต

โปรแกรมที่ใช้ผ่านเครือข่ายนี้ประกอบด้วยส่วนย่อย 2 ส่วนคือ

ส่วนที่ 1 เป็นส่วนของหน้าจอ ที่ใช้ในการ ควบคุมผ่าน web browser ที่อยู่บนฝั่ง client มีหน้าที่เป็นส่วนติดต่อผู้ใช้ โดยมีส่วนติดต่อผู้ใช้ โปรแกรมจะประมวลผลคำสั่ง และส่งข้อมูลที่ได้ออกทางพอร์ตอนุกรมทันที แต่โปรแกรม นี้ จะต้องนำค่าที่ใช้ ในการ ควบคุมต่างๆ ไปบันทึกไว้ในฐานข้อมูลที่สร้างขึ้นมาก่อน

ส่วนที่ 2 เป็นส่วนของโปรแกรม INET ทำหน้าที่ในการสั่งงานออกทางพอร์ต อนุกรม และการควบคุม ซึ่งเป็นโปรแกรมที่ทำงานอยู่บน ฝั่ง server เพียงอย่างเดียว ไม่มีการ ติดต่อกับ ระบบอินเทอร์เน็ต โดยในส่วนนี้จะทำหน้าที่ในการ ไปอ่านค่าจากฐานข้อมูลที่ ได้ถูกบันทึกค่าไว้แล้ว มาประมวลผลคำสั่งและ ส่งข้อมูลที่ ต้องการควบคุมไปให้ รีเลย์บอร์ดผ่านทางพอร์ตอนุกรม โดยจะต้องมีการวนมาอ่านค่าจากฐานข้อมูลทุกๆ 8 มิลลิวินาที

3.3.1 โปรแกรมย่อยส่วนที่ 1

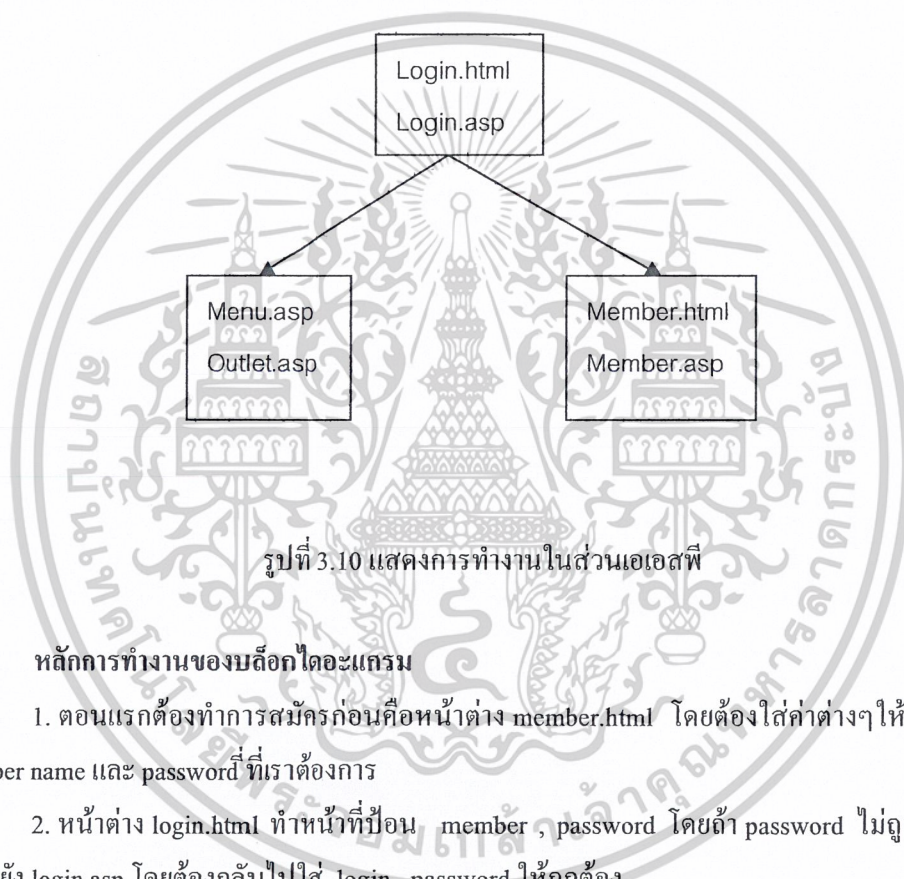
ในการทำงานร่วมกับเว็บเพจหนึ่งๆเราจะต้องไปยังที่อยู่ของ URL นั้นๆซึ่งทำงานอยู่บนเซิร์ฟเวอร์ โดย หากเว็บเพจที่สร้างไว้มีแค่หน้าที่ในการนำเสนอเพียงอย่างเดียว เอกสารดังกล่าวก็จะถูกสร้างจากภาษา HTML แต่หากเพจดังกล่าวมีการทำงานร่วมกับเรา เช่น การบันทึกแบบฟอร์มและส่งไปยังเซิร์ฟเวอร์ การทํางานจะไม่ได้จบแค่ที่เพจนี้ แต่จะต้องมีการทํางานต่อบนเซิร์ฟเวอร์ อีกทั้งโปรแกรมนี้ยังทำงานอยู่บนฝั่งของ server ซึ่งไม่เหมือนกับเอกสาร HTML ที่ทำงานอยู่บนเครื่องของเรา เช่นเดียวกับในโครงการนี้ ดังนั้นใน โครงการนี้จึงได้เลือกใช้ ASP-Active Server Page เป็นภาษาที่ใช้ในการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจาก ASP มีโครงสร้าง คล้ายกับ ภาษา Visual Basic โดยเฉพาะ VBScript และเป็นภาษา ที่มีความ สะดวกในการใช้ติดต่อ ฐานข้อมูล

ในระหว่างการพัฒนาโปรแกรม ASP จำเป็นต้องมีเว็บเซิร์ฟเวอร์ เพื่อใช้ในการดูการทำงาน ของ โปรแกรม ที่ได้พัฒนาขึ้น ซึ่งในโปรแกรมนี้ได้เลือกใช้โปรแกรม PWS - Personal Web Server ของ Microsoft เป็นตัวจำลองการทำงานบนเครื่องของเรา

3.3.2 บล็อกไดอะแกรมแสดงการทำงานในส่วนเอเอสพี



รูปที่ 3.10 แสดงการทำงานในส่วนเอเอสพี

หลักการทํางานของบล็อกไดอะแกรม

1. ตอนแรกต้องทำการสมัครก่อนคือหน้าต่าง member.html โดยต้องใส่ค่าต่างๆให้ครบก็จะได้ member name และ password ที่เราต้องการ
2. หน้าต่าง login.html ทำหน้าที่ป้อน member , password โดยถ้า password ไม่ถูกต้อง จะถูกส่งไปยัง login.asp โดยต้องกลับไปใส่ login , password ให้ถูกต้อง
3. หน้าต่าง menu.asp ใช้สั่งงานการเปิดปิดอุปกรณ์ไฟฟ้า โดยจะตรวจสอบค่าจากฐานข้อมูลแล้วแสดงผลออกมาว่า ch1 -8 เปิดหรือปิดอยู่และเวลาที่ตั้งไว้ ส่วนoutlet.asp จะถูกเรียกใช้เมื่อเราต้องการเปิดปิดอุปกรณ์ไฟฟ้าหรือต้องการตั้งเวลาโดยจะนำค่าไปเก็บไว้ในฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ฟังก์การทำงานในส่วนเอเอสพี

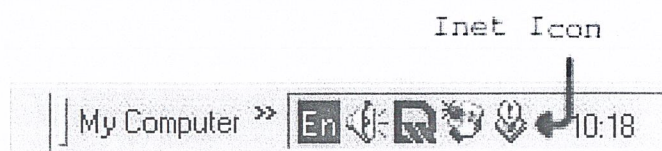


รูปที่ 3.11 แสดงฟังก์การทำงานในส่วนเอเอสพี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 โปรแกรมย่อยส่วนที่ 2 (โปรแกรม Inet)

โปรแกรมย่อยในส่วนที่ 2 มีหน้าที่ในการอ่านค่าจากฐานข้อมูล โดยจะวนอ่านฐานข้อมูลไปเรื่อยๆ และส่งค่าที่ได้ออกทางพอร์ต อนุกรมเพื่อทำการควบคุมต่อไปโดยที่โปรแกรมดังกล่าวใช้ภาษา Visual Basic 6.0 ในการพัฒนา

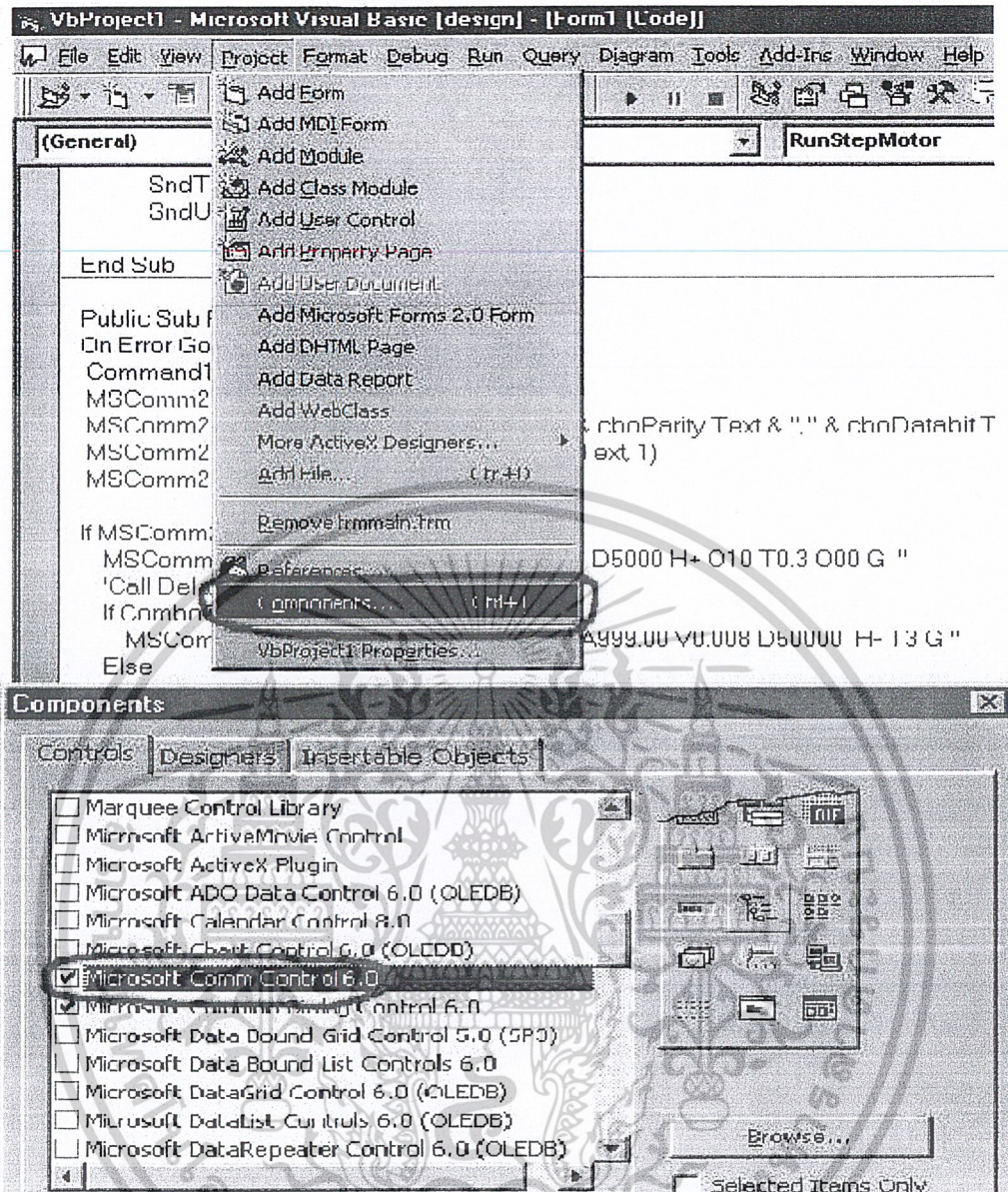


รูปที่ 3.12 แสดงตำแหน่ง Icon ของโปรแกรม Inet

ซึ่งเป็นโปรแกรมที่ใช้ในการสร้างโปรแกรมประยุกต์ ซึ่งพัฒนาโดยบริษัท Microsoft มีการใช้งานง่าย ทำให้เรียนรู้ได้อย่างรวดเร็ว คุณสมบัติอย่างหนึ่งของ Visual Basic 6.0 ก็คือสามารถนำไปติดต่อกับอุปกรณ์ ภายนอกได้ โดยผ่านทางพอร์ตขนาน หรือพอร์ตอนุกรมของเครื่องคอมพิวเตอร์ การติดต่อกับอุปกรณ์ ภายนอก ด้วยพอร์ตขนานนั้น เราสามารถทำได้โดยใช้ไฟล์ที่มีชื่อว่า INPOUT32.DLL โดยจะต้องเพิ่มเข้าไปใน General Module จึงสามารถติดต่อกับอุปกรณ์ภายนอกโดยผ่านพอร์ตขนานได้

ส่วนในโครงการนี้ จะใช้งานผ่าน พอร์ต อนุกรมดังนั้นจะใช้ Control MSComm32.ocx ซึ่งมีมาพร้อม กับโปรแกรม Visual Basic อยู่แล้วแต่จะไม่แสดงไว้ใน toolbox ตั้งแต่แรก ซึ่งเราจะต้องนำมาแสดงไว้ใน toolbox ได้โดยการเข้าไปติดตั้งจาก Project > Components > Controls โดยมีชื่อว่า Microsoft Comm Control 6.0 แสดงไว้ดังรูปที่ 3.13 เมื่อติดตั้งแล้วก็สามารถใช้งานได้ตามปกติเช่นเดียวกับเครื่องมืออื่นๆ ใน toolbox ซึ่งทำให้การใช้งานการใช้งานของพอร์ตอนุกรมใช้งาน ได้สะดวกมากกว่า พอร์ตขนาน และใช้สายสัญญาณ ที่มีความยาวมากกว่าโดยเกิดการสูญเสียสัญญาณ ดังนั้นการติดต่อกับอุปกรณ์ภายนอกผ่านทางพอร์ต อนุกรม ของคอมพิวเตอร์ จึงได้รับความนิยมเพิ่มมากขึ้น

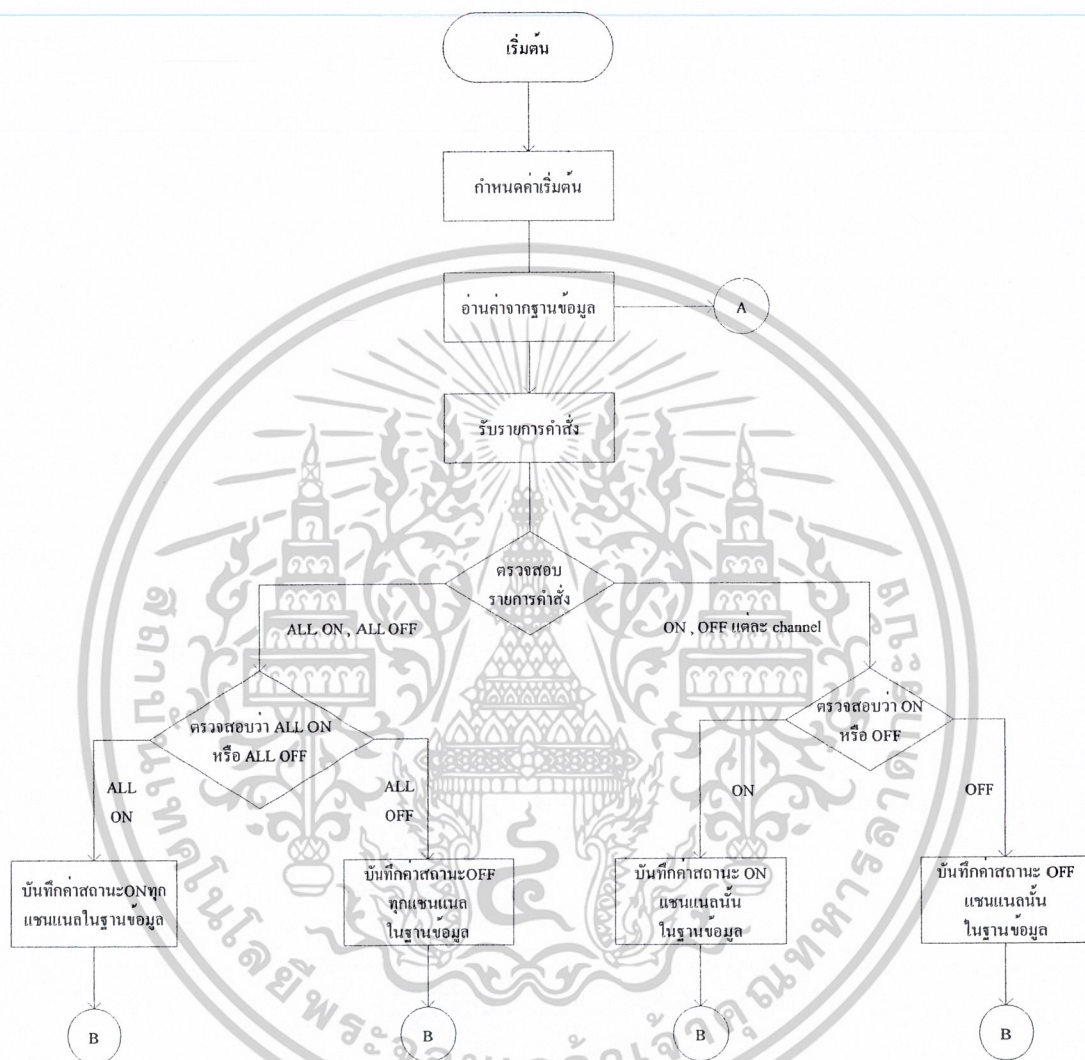
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงการติดตั้ง MSComm 6.0 ใน toolbox ของโปรแกรม Visual Basic 6.0

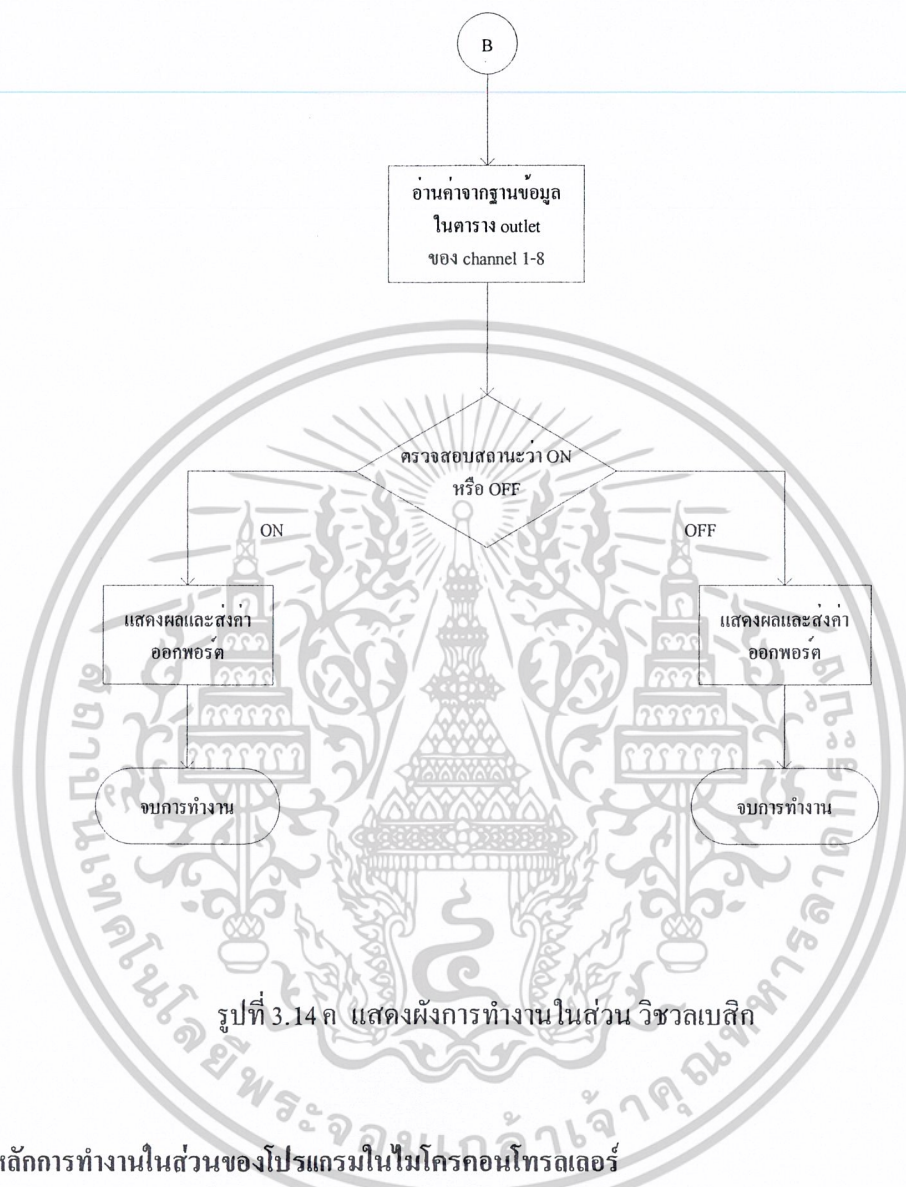
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5 ฟังก์การทำงานในส่วนของ Inet



รูปที่ 3.14 ก แสดงฟังก์การทำงานในส่วน วิชาลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงผังการทำงานในส่วน วิชาลเบสิก

3.4 หลักการทำงานในส่วนของโปรแกรมในไมโครคอนโทรลเลอร์

ในโครงงานนี้ใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C2051 เชื่อมต่อกับอุปกรณ์ต่างๆตามที่กล่าวมาแล้วในหัวข้อ 3.2 ส่วนของโปรแกรมที่ใช้จะอธิบายในหัวข้อนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 ฟังก์ชันการทำงานของโปรแกรมในไมโครคอนโทรลเลอร์



รูปที่ 3.15 แสดงฟังก์ชันการทำงานในส่วนของไมโครคอนโทรลเลอร์

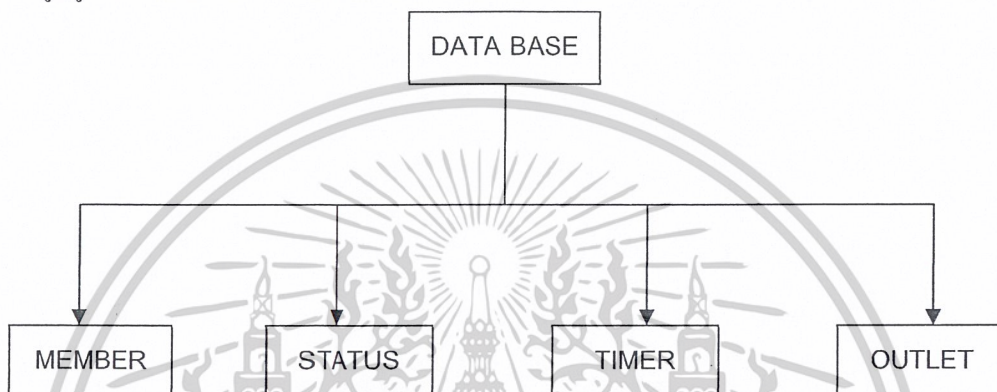
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ส่วนการออกแบบฐานข้อมูล

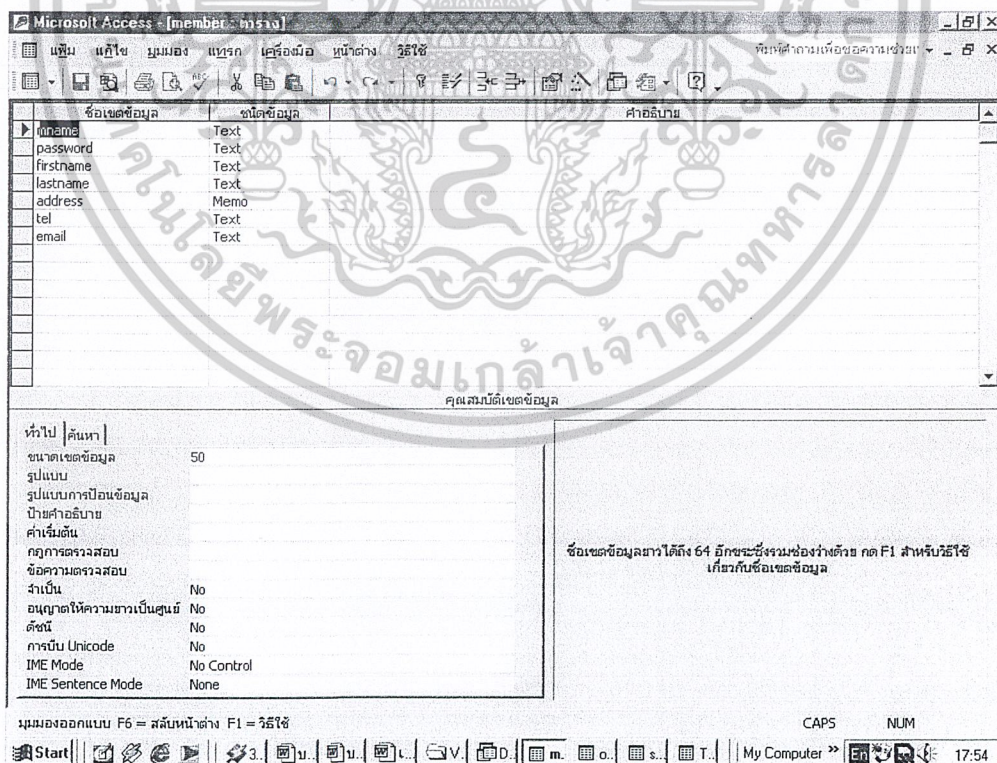
ฐานข้อมูลประกอบด้วย 4 ส่วนในการเก็บข้อมูลดังนี้

1. แสดงสถานะของผู้ใช้งาน (ตารางชื่อ member)
2. แสดงสถานะการเปิด-ปิด ของอุปกรณ์ (ตาราง status)
3. แสดงสถานะการณ้ตั้งเวลาเปิด- ปิด อุปกรณ์ (ตาราง timer)
4. แสดงสถานะของแต่ละ channel (ตาราง outlet)

ฐานข้อมูลถูกออกแบบโดยใช้โปรแกรม ACCESS

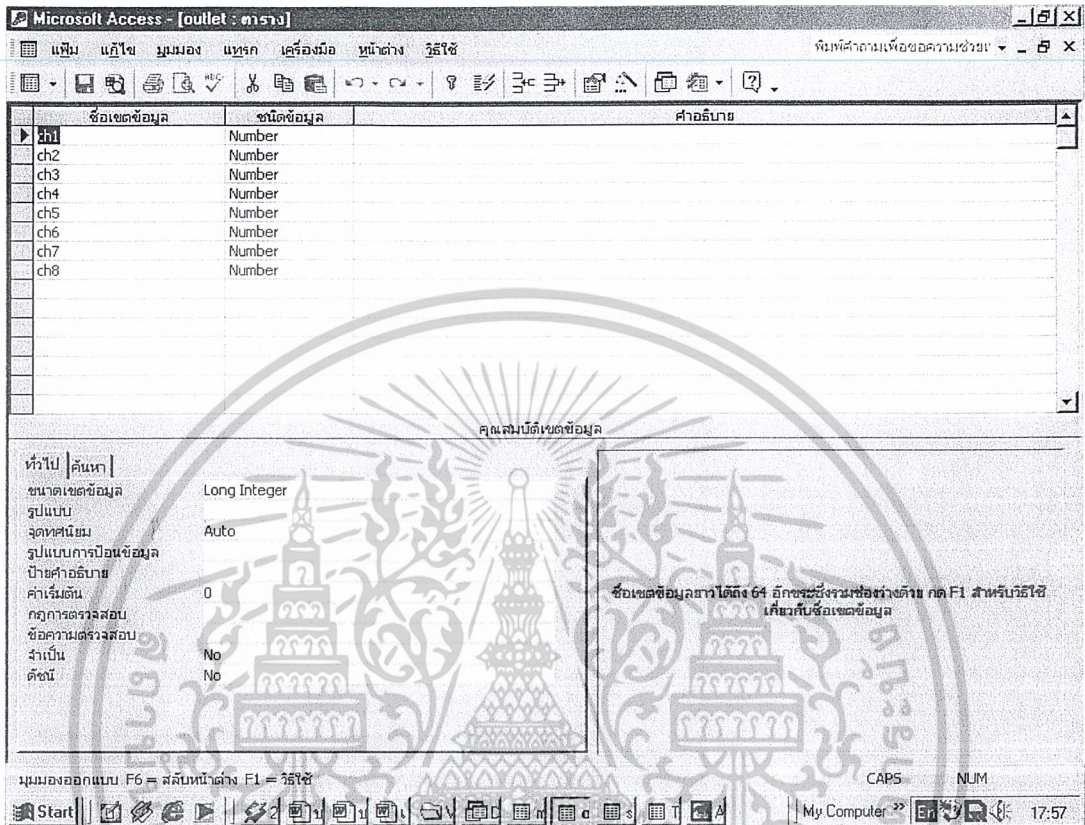


รูปที่ 3.16 แสดงโครงสร้างฐานข้อมูล



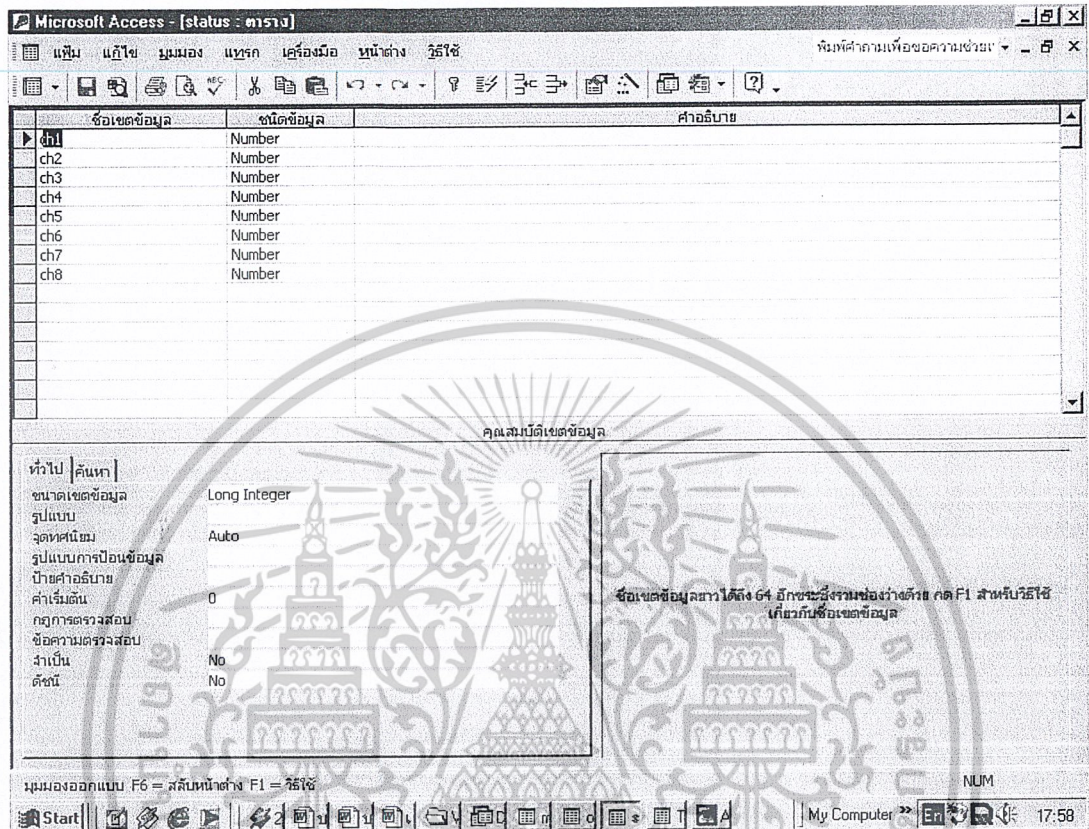
รูปที่ 3.17 แสดงตาราง member

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



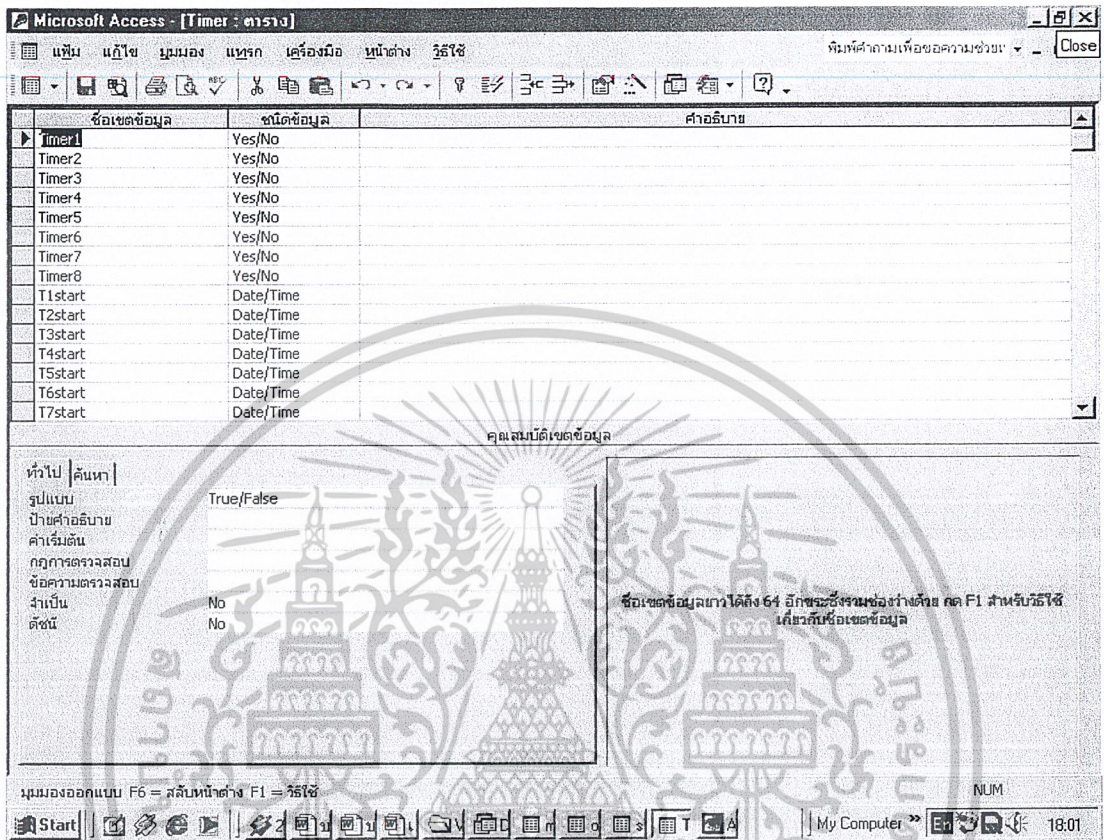
รูปที่ 3.18 แสดงตาราง outlet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 แสดงตาราง Status

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 แสดงตาราง Timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

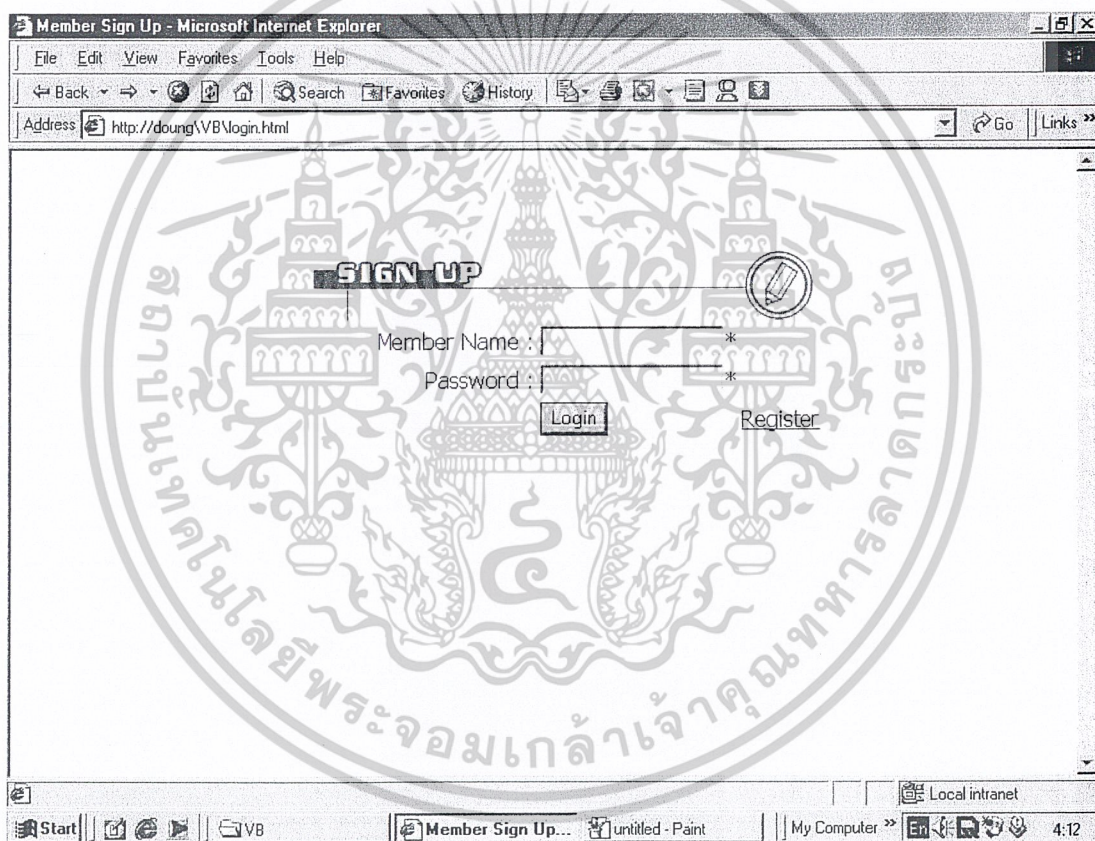
บทที่ 4

การทดลองและผลการทดลอง

4.1 ส่วนของโฮมเพจ

4.1.1 หน้าต่าง ล็อกอิน

เมื่อเข้าสู่โฮมเพจ หน้าต่างแรกที่จะพบคือ หน้าต่างล็อกอิน จะเป็นการถาม Username และ Password สำหรับผู้ที่ลงทะเบียนแล้ว และมีปุ่ม Register สำหรับผู้ที่ยังไม่ลงทะเบียน



รูปที่ 4.1 แสดงหน้าต่าง Login

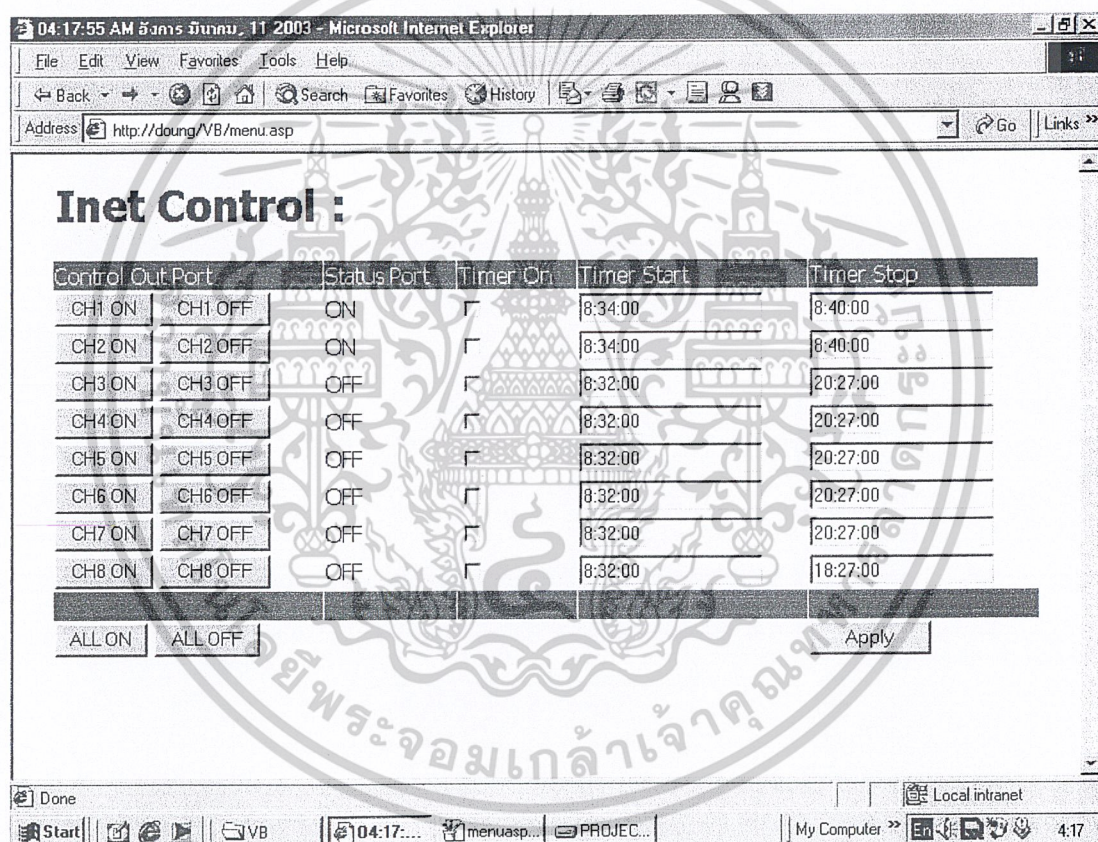
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 หน้าต่าง เมนู

คือ หน้าต่างที่ใช้ควบคุมอุปกรณ์ จะถัดจากหน้าต่าง Login ก็คือต้องใส่ username และ password ให้ถูกต้องก่อน จึงจะปรากฏหน้าต่างนี้

ส่วนควบคุมแบ่งออกเป็น 2 ส่วน คือ

1. ควบคุมจากปุ่ม ON, OFF ในแต่ละ channel หรือปุ่ม ALL ON, ALL OFF
2. ควบคุมโดยใช้เวลา ใช้งาน โดยใส่เครื่องหมายในช่อง TIMER ON และใส่เวลาเปิดปิดลงในช่อง Timer Start และ Timer Stop และคลิกไปที่ปุ่ม Apply



รูปที่ 4.2 แสดงหน้าต่าง Menu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 หน้าต่าง Member

คือ หน้าต่างสำหรับผู้ที่ยังไม่ลงทะเบียน เมื่อเข้ามาที่หน้าต่าง login แล้วคลิกไปที่ปุ่ม register จะปรากฏหน้าต่างนี้ขึ้น

หลังจากกรอกรายละเอียดทั้งหมดแล้วให้คลิกไปที่ปุ่ม sign me up เพื่อส่งเก็บข้อมูลไปยังฐานข้อมูลในส่วนตาราง member

The screenshot shows a web browser window titled "Member Sign Up - Microsoft Internet Explorer". The address bar contains "http://dong/VB/member.html". The main content area displays a "SIGN-UP" form with the following fields and labels:

- Member Name : *
- Password : *
- Re-password : *
- ชื่อ : *
- นามสกุล : *
- ที่อยู่ : *
- โทรศัพท์ : *
- E-mail : *

At the bottom of the form, there are two buttons: "RESET" and "SIGN ME UP". The background of the page features a large, faint watermark of a university seal.

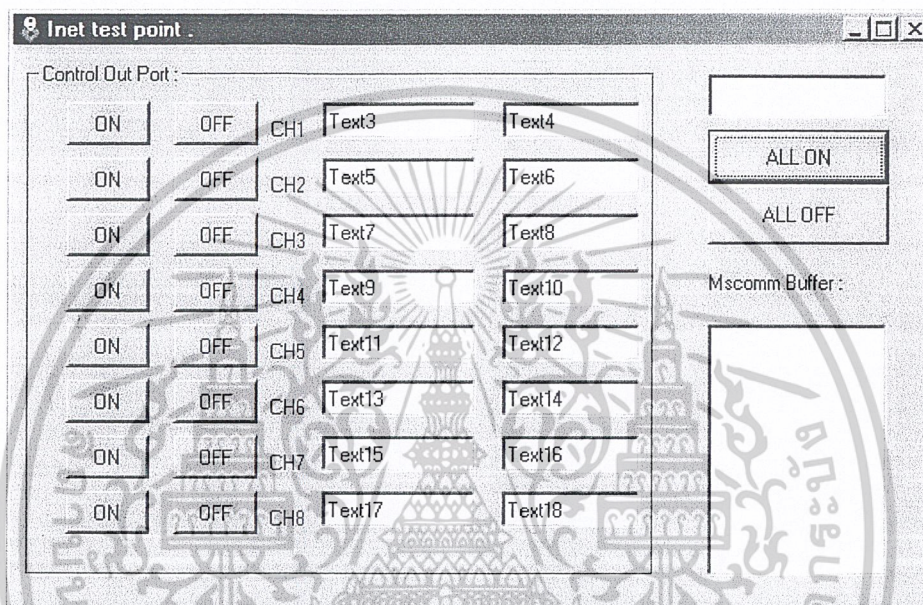
รูปที่ 4.3 แสดงหน้าต่าง Member

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ส่วนของโปรแกรม Visual basic (I NET)

4.2.1 หน้าต่างควบคุมเครื่องใช้ไฟฟ้า

มีหน้าที่แสดงสถานะและเวลาที่ตั้งไว้ของ Ch 1-8 ผ่านทาง Mscomm Buffer และทำหน้าที่ควบคุมอุปกรณ์ไฟฟ้าได้โดยตรง



รูปที่ 4.4 แสดงหน้าต่างควบคุมของ INET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และบทสรุป

5.1 ปัญหาที่พบจากการทดลอง

1. แนวความคิดที่ใช้ออกแบบวงจรผิดพลาดทำให้เสียเวลาในการออกแบบใหม่
2. เสียเวลาในการศึกษาค้นคว้าโปรแกรมใหม่ๆหลายโปรแกรม
3. อุปกรณ์เกิดความผิดพลาด ทำให้ยากในการตรวจสอบวงจร
4. ส่วนของ ASP ค่อนข้างซับซ้อน หากจุดผิดพลาดลำบากทำให้ยากต่อการวิเคราะห์ข้อผิดพลาดที่เกิดขึ้น
5. โปรแกรม SERVER มีปัญหาเรื่องความเสถียรและระบบรักษาความปลอดภัยในการนำไปใช้งาน

5.2 แนวทางแก้ไขปัญหา

1. ควรศึกษาคุณสมบัติของอุปกรณ์แต่ละตัวให้ดีก่อนจะออกแบบวงจร
2. ควรใช้โปรแกรมที่เราเคยศึกษาหรือมีผู้ที่คอยแนะนำได้
3. ควรเลือกใช้ระบบที่มีความเสถียรสูงเช่น Window NT หรือ IIS

5.4 แนวทางในการพัฒนา

1. ควรพัฒนาอุปกรณ์ที่สามารถเปิดเครื่องคอมพิวเตอร์ได้อัตโนมัติ ทำให้ไม่ต้อง Online ตลอดเวลา
2. พัฒนาขนาดของอุปกรณ์ในส่วนของบอร์ดรีเลย์ให้มีขนาดเล็ก เพื่อสะดวกในการติดตั้งมากขึ้น

5.3 ข้อเสนอแนะเกี่ยวกับโครงการ

1. นอกจาก ASP แล้วสามารถใช้ CGI, PHP ในการเขียนโปรแกรมได้
2. โปรแกรมในการเขียน HTML อาจใช้ Dream Weaver, Front Page ในการเขียนโฮมเพจ
3. โปรแกรม SERVER อาจใช้ IIS, PWS, OMNIHTTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

SOURCE CODE ส่วน Visual Basic

Option Explicit

'ประกาศตัวแปรสำหรับ connect database

Private Const strConn = "Provider=Microsoft.Jet.OLEDB.4.0;Persist Security Info=False"

Dim conn As New ADODB.Connection

Dim CH As Integer

Dim rsstatus As New ADODB.Recordset

Private Clicked As Boolean

Private Sub GoSystemTray()

'ใช้สำหรับให้โปรแกรม show ที่ system tray

VBGTray.cbSize = Len(VBGTray)

VBGTray.hwnd = Me.hwnd

VBGTray.uId = vbNull

VBGTray.uFlags = NIF_ICON Or NIF_TIP Or NIF_MESSAGE

VBGTray.ucallbackMessage = WM_MOUSEMOVE

VBGTray.hIcon = Me.Icon 'เปลี่ยน Icon ตรงนี้ได้นะ

'Tool Tip

VBGTray.szTip = Me.Caption & vbNullChar

Call Shell NotifyIcon(NIM_ADD, VBGTray)

App.TaskVisible = False 'ซ่อนโปรแกรมจาก Taskbar

Me.Hide

End Sub

Private Sub CH1_Timer()

' Timer สำหรับอ่านคำสั่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที

' ใช้สำหรับ Channel 1

On Error GoTo err_1

CH1.Enabled = False ' ปิด timer ch1

Dim rsCommand As New ADODB.Recordset

Dim strSql As String

strSql = "select * from outlet" ' อ่านคำสั่ง ch1 จากตาราง outlet

rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText

rsCommand.MoveFirst

MSComm1.Output = Chr(rsCommand("ch1")) ' ส่งคำสั่งออก port

Text1 = Time

Dim rs As New ADODB.Recordset

strSql = "select * from timer" ' เช็คเวลาที่ตั้งไว้

rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText

Text3 = rs("tlstart")

Text4 = rs("tlstop")

If rs("timer1") Then

If rs("tlstart") = Time Then

rsCommand.Update "ch1", 11

End If

If rs("tlstop") = Time Then

rsCommand.Update "ch1", 1

End If

End If

rs.Close

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vb

```
rsCommand.Close
```

```
CH2.Enabled = True ' เปิด timer ch2  
Exit Sub
```

```
err_1:  
MsgBox Err.Number & vbNewLine & Err.Description
```

```
End Sub
```

```
Private Sub CH2_Timer()  
' Timer สำหรับอ่านค่าสิ่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที  
' ใช้สำหรับ Channel 2
```

```
On Error GoTo err_1  
CH2.Enabled = False  
Dim rsCommand As New ADODB.Recordset  
Dim strSql As String  
strSql = "select * from outlet"  
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText  
rsCommand.MoveFirst
```

```
MSComml.Output = Chr(rsCommand("ch2"))  
Dim rs As New ADODB.Recordset  
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้  
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText  
Text5 = rs("t2start")  
Text6 = rs("t2stop")  
If rs("timer2") Then  
If rs("t2start") = Time Then  
rsCommand.Update "ch2", 12  
End If  
If rs("t2stop") = Time Then  
rsCommand.Update "ch2", 2  
End If  
End If  
rs.Close  
rsCommand.Close
```

```
CH3.Enabled = True  
Exit Sub
```

```
err_1:  
MsgBox Err.Number & vbNewLine & Err.Description
```

```
End Sub
```

```
Private Sub CH3_Timer()  
' Timer สำหรับอ่านค่าสิ่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที  
' ใช้สำหรับ Channel 3
```

```
On Error GoTo err_1  
CH3.Enabled = False  
Dim rsCommand As New ADODB.Recordset  
Dim strSql As String  
strSql = "select * from outlet"  
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText  
rsCommand.MoveFirst  
MSComml.Output = Chr(rsCommand("ch3"))  
Dim rs As New ADODB.Recordset  
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้  
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText  
Text7 = rs("t3start")  
Text8 = rs("t4stop")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 2 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vb

```
If rs("timer3") Then
    If rs("t3start") = Time Then
        rsCommand.Update "ch3", 13
    End If
    If rs("t3stop") = Time Then
        rsCommand.Update "ch3", 3
    End If
End If
rs.Close
rsCommand.Close

CH4.Enabled = True
Exit Sub

err_1:
    MsgBox Err.Number & vbNewLine & Err.Description

End Sub

Private Sub CH4_Timer()
' Timer สำหรับอ่านค่าส่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที
' ใช้สำหรับ Channel 4

On Error GoTo err_1
CH4.Enabled = False
Dim rsCommand As New ADODB.Recordset
Dim strSql As String
strSql = "select * from outlet"
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText
rsCommand.MoveFirst
    MSComm1.Output = Chr(rsCommand("ch4"))
Dim rs As New ADODB.Recordset
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText
Text9 = rs("t4start")
Text10 = rs("t4stop")
If rs("timer4") Then
    If rs("t4start") = Time Then
        rsCommand.Update "ch4", 14
    End If
    If rs("t4stop") = Time Then
        rsCommand.Update "ch4", 4
    End If
End If
rs.Close
rsCommand.Close

CH5.Enabled = True
Exit Sub

err_1:
    MsgBox Err.Number & vbNewLine & Err.Description

End Sub

Private Sub CH5_Timer()
' Timer สำหรับอ่านค่าส่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที
' ใช้สำหรับ Channel 5

On Error GoTo err_1
CH5.Enabled = False
Dim rsCommand As New ADODB.Recordset
Dim strSql As String
strSql = "select * from outlet"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

vb
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText
rsCommand.MoveFirst
    MScmm1.Output = Chr(rsCommand("ch5"))
Dim rs As New ADODB.Recordset
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText
Text11 = rs("t5start")
Text12 = rs("t5stop")
If rs("timer5") Then
    If rs("t5start") = Time Then
        rsCommand.Update "ch5", 15
    End If
    If rs("t5stop") = Time Then
        rsCommand.Update "ch5", 5
    End If
End If
rs.Close
rsCommand.Close

CH6.Enabled = True
Exit Sub

```

```

err_1:
    MsgBox Err.Number & vbNewLine & Err.Description
End Sub

```

```

Private Sub CH6_Timer()
' Timer สำหรับอ่านค่าส่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที
' ใช้สำหรับ Channel 6

On Error GoTo err_1
CH6.Enabled = False
Dim rsCommand As New ADODB.Recordset
Dim strSql As String
strSql = "select * from outlet"
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText
rsCommand.MoveFirst
    MScmm1.Output = Chr(rsCommand("ch6"))
Dim rs As New ADODB.Recordset
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText
Text13 = rs("t6start")
Text14 = rs("t6stop")
If rs("timer6") Then
    If rs("t6start") = Time Then
        rsCommand.Update "ch6", 16
    End If
    If rs("t6stop") = Time Then
        rsCommand.Update "ch6", 6
    End If
End If
rs.Close
rsCommand.Close

```

```

CH7.Enabled = True
Exit Sub

```

```

err_1:
    MsgBox Err.Number & vbNewLine & Err.Description
End Sub

```

```

Private Sub CH7_Timer()

```

vb

```
' Timer สำหรับอ่านค่าส่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที  
' ใช้สำหรับ Channel 7
```

```
On Error GoTo err_1  
CH7.Enabled = False  
Dim rsCommand As New ADODB.Recordset  
Dim strSql As String  
strSql = "select * from outlet"  
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText  
rsCommand.MoveFirst  
MSComm1.Output = Chr(rsCommand("ch7"))  
Dim rs As New ADODB.Recordset  
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้  
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText  
Text15 = rs("t7start")  
Text16 = rs("t7stop")  
If rs("timer7") Then  
    If rs("t7start") = Time Then  
        rsCommand.Update "ch7", 17  
    End If  
    If rs("t7stop") = Time Then  
        rsCommand.Update "ch7", 7  
    End If  
End If  
rs.Close  
rsCommand.Close  
  
CH8.Enabled = True  
Exit Sub
```

```
err_1:  
    MsgBox Err.Number & vbNewLine & Err.Description  
End Sub
```

```
Private Sub CH8_Timer()  
' Timer สำหรับอ่านค่าส่งจาก Database แล้วส่งออกที่ Port ทำงานเองตามเวลาที่ตั้งไว้ทุก 8/1000 วินาที  
' ใช้สำหรับ Channel 8  
  
On Error GoTo err_1  
CH8.Enabled = False  
Dim rsCommand As New ADODB.Recordset  
Dim strSql As String  
strSql = "select * from outlet"  
rsCommand.Open strSql, conn, adOpenDynamic, adLockOptimistic, adCmdText  
rsCommand.MoveFirst  
MSComm1.Output = Chr(rsCommand("ch8"))  
Dim rs As New ADODB.Recordset  
strSql = "select * from timer" ' เช็ควเวลาที่ตั้งไว้  
rs.Open strSql, conn, adOpenKeyset, adLockReadOnly, adCmdText  
Text17 = rs("t8start")  
Text18 = rs("t8stop")  
If rs("timer8") Then  
    If rs("t8start") = Time Then  
        rsCommand.Update "ch8", 18  
    End If  
    If rs("t8stop") = Time Then  
        rsCommand.Update "ch8", 8  
    End If  
End If  
rs.Close  
rsCommand.Close
```

```
CH1.Enabled = True
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vb

Exit Sub

err_1:

MsgBox Err.Number & vbNewLine & Err.Description

End Sub

Private Sub Command1_Click()

' ทดลองส่งคำสั่งเก็บลง database ให้ ON ทุก Channel

Dim rsCommand As New ADODB.Recordset

Dim strSql As String

strSql = "select * from outlet"

rsCommand.Open strSql, conn, adOpenKeyset, adLockOptimistic, adCmdText

If rsCommand.EOF Then Exit Sub

' เก็บค่าเหล่านี้ลง database เพื่อสั่ง on

rsCommand("ch1") = "11"

rsCommand("ch2") = "12"

rsCommand("ch3") = "13"

rsCommand("ch4") = "14"

rsCommand("ch5") = "15"

rsCommand("ch6") = "16"

rsCommand("ch7") = "17"

rsCommand("ch8") = "18"

rsCommand.Update

rsCommand.Close

End Sub

Private Sub Command2_Click()

Dim rsCommand As New ADODB.Recordset

Dim strSql As String

strSql = "select * from outlet"

rsCommand.Open strSql, conn, adOpenKeyset, adLockOptimistic, adCmdText

If rsCommand.EOF Then Exit Sub

' เก็บค่าเหล่านี้ลง database เพื่อสั่ง off

rsCommand("ch1") = "1"

rsCommand("ch2") = "2"

rsCommand("ch3") = "3"

rsCommand("ch4") = "4"

rsCommand("ch5") = "5"

rsCommand("ch6") = "6"

rsCommand("ch7") = "7"

rsCommand("ch8") = "8"

rsCommand.Update

rsCommand.Close

End Sub

Private Sub Form_Load()

On Error GoTo err_1

Call GoSystemTray

MSComm1.Settings = "9600,n,8,1"

MSComm1.InputLen = 1

MSComm1.RThreshold = 1

' อ่านค่า comm port ที่เก็บลง Registry มา set Mscomm1

MSComm1.CommPort = GetSetting(App.EXENAME, "Setting", "CommPort", 1)

MSComm1.PortOpen = True

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 6 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vb

```
' Connect database
OpenDB

' เปิด Timer chl
CH1.Enabled = True
Exit Sub

err_1:
If Err.Number = 8005 Then
    MsgBox "Port already opened by another program " & vbNewLine & "Select
new port please ..", vbInformation, "Mscomm Error "
Else
    MsgBox Err.Number & vbNewLine & Err.Description, vbInformation

End If

End Sub
Private Sub OpenDB()
On Error GoTo err_1
    With conn
        If .State = adStateOpen Then .Close
        .ConnectionString = strConn & ";Data Source=" & App.Path & "\data.mdb"
        .CursorLocation = adUseClient
        .ConnectionTimeout = 90

        .Open
    End With
    rsstatus.Open "status", conn, adOpenKeyset, adLockOptimistic, adCmdTable

Exit Sub
err_1:
    MsgBox Err.Description, vbInformation, "OpenDB error "

End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
    Static lngMsg As Long
    Static blnFlag As Boolean
    Dim result As Long
    lngMsg = X / Screen.TwipsPerPixelX
    If blnFlag = False Then
        blnFlag = True
        Select Case lngMsg

            Case WM_LBUTTONDOWNCLICK 'เปลี่ยนเป็นอย่างอื่นได้นะ ดูจาก Module1 หรือ API Text Viewer
                'แสดง form เมื่อ ดับเบิลคลิกที่ icon
                Me.Show
            Case WM_RBUTTONDOWNUP 'เปลี่ยนเป็นอย่างอื่นได้นะ ดูจาก Module1 หรือ API Text Viewer
                ' แสดง menu เมื่อ click ขวาที่ icon
                result = SetForegroundWindow(Me.hwnd)
                Me.PopupMenu mPopupMenu

        End Select
        blnFlag = False
    End If

End Sub

Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
On Error Resume Next
MSComm1.PortOpen = False
VBGTray.cbSize = Len(VBGTray)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 7 การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

vb

```
VBGTray.hwnd = Me.hwnd
VBGTray.uId = vbNull
Call Shell_NotifyIcon(NIM_DELETE, VBGTray)
```

End Sub

```
Private Sub mAbout_Click()
' MsgBox "About code goes here.", vbInformation, "About"
frmAbout.Show vbModal
```

End Sub

```
Public Sub mExit_Click()
```

```
Unload Me
```

End Sub

```
Private Sub MSComm1_OnComm()
```

```
On Error GoTo com_err
```

```
Dim buffer As Variant
```

```
Select Case MSComm1.CommEvent
```

```
Case comEvReceive
```

```
buffer = MSComm1.Input
```

```
Select Case Asc(buffer)
```

```
Case 1, 11
```

```
rsstatus.Update "ch1", Asc(buffer)
```

```
Case 2, 12
```

```
rsstatus.Update "ch2", Asc(buffer)
```

```
Case 3, 13
```

```
rsstatus.Update "ch3", Asc(buffer)
```

```
Case 4, 14
```

```
rsstatus.Update "ch4", Asc(buffer)
```

```
Case 5, 15
```

```
rsstatus.Update "ch5", Asc(buffer)
```

```
Case 6, 16
```

```
rsstatus.Update "ch6", Asc(buffer)
```

```
Case 7, 17
```

```
rsstatus.Update "ch7", Asc(buffer)
```

```
Case 8, 18
```

```
rsstatus.Update "ch8", Asc(buffer)
```

```
End Select
```

```
End Select
```

```
If CH > 8 Then
```

```
Text2 = ""
```

```
CH = 0
```

```
End If
```

```
Text2 = Asc(buffer) & vbNewLine & Text2
```

```
CH = CH + 1
```

Exit Sub

```
com_err:
```

```
MsgBox Err.Number & Err.Description, vbInformation, "Comm Error "
```

End Sub

```
Private Sub mSetting_Click()
```

```
frmSetting.Show vbModal
```

End Sub

```
Private Sub OFF_Click(Index As Integer)
```

```
MSComm1.Output = Chr(1 + Index)
```

End Sub

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 8 การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
vb
Private Sub ON_Click(Index As Integer)
MSComm1.Output = Chr(11 + Index)

End Sub

Private Sub Timer2_Timer()
If Me.WindowState = vbMinimized Then
Call GoSystemTray
End If

End Sub
```



html

SOURCE CODE ส่วน HTML


```
<html>
<head>
  <title>Member Sign Up</title>
</head>
<SCRIPT language=Javascript>

  function Validator() {

    if (document.FormAddress.mname.value == "") {
      alert("กรุณากรอกข้อมูลลงในช่อง Member Name ด้วยครับ.");
      document.FormAddress.mname.focus();
      return (false);
    }

    if (document.FormAddress.password.value == "") {
      alert("กรุณากรอกข้อมูลลงในช่อง Password ด้วยครับ.");
      document.FormAddress.password.focus();
      return (false);
    }

    return (true);
  }
</SCRIPT>

<body>

<form name="FormAddress" onsubmit="return Validator()" action="login.asp"
method="post">
<br>
<br>
<br>
<center>
<img src ="image/signup.gif" width="350" height="50"><br>
<table border="0">
<tr><td width="200" align="right">Member Name :</td><td>
  <input type="text" name ="mname" size="20"><FONT
  color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">Password :</td><td>
  <input type="password" name="password" size="20"><FONT
  color=#ff0000>*</FONT></td></tr>

<tr><td width="200" align="right"></td><td><input type="submit"
value="Login"></td><td><a href="member.html">Register</a></td></tr>
</table>
<br>

</center>
</form>
</body>
</html>
```

```
<html>
<head>
  <title>Member Sign Up</title>
</head>
<SCRIPT language=Javascript>
```

```
function Validator() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
if (document.FormAddress.mname.value == "") {
    alert("กรุณากรอกข้อมูลลงในช่อง Member Name ด้วยครับ.");
    document.FormAddress.mname.focus();
    return (false);
}

if (document.FormAddress.password.value == "") {
    alert("กรุณากรอกข้อมูลลงในช่อง Password ด้วยครับ.");
    document.FormAddress.password.focus();
    return (false);
}

if (document.FormAddress.repassword.value == "") {
    alert("กรุณากรอกข้อมูลลงในช่อง Re-Password ด้วยครับ.");
    document.FormAddress.repassword.focus();
    return (false);
}

if (document.FormAddress.password.value !=
document.FormAddress.repassword.value) {
    alert("กรุณากรอก Password และ Re-Password ให้ตรงกัน");
    document.FormAddress.repassword.focus();
    return (false);
}

if (document.FormAddress.fname.value == "") {
    alert("กรุณากรอกชื่อด้วยครับ.");
    document.FormAddress.fname.focus();
    return (false);
}

if (document.FormAddress.lname.value == "") {
    alert("กรุณากรอกนามสกุลด้วยครับ.");
    document.FormAddress.lname.focus();
    return (false);
}

if (document.FormAddress.tel.value == "") {
    alert("กรุณากรอกที่อยู่ด้วยครับ.");
    document.FormAddress.tel.focus();
    return (false);
}

if (document.FormAddress.tel.value == "") {
    alert("กรุณากรอกหมายเลขโทรศัพท์ด้วยครับ.");
    document.FormAddress.tel.focus();
    return (false);
}

if (document.FormAddress.email.value == "") {
    alert("กรุณากรอก E-mail ด้วยครับ.");
    document.FormAddress.email.focus();
    return (false);
}

return (true);
}
</SCRIPT>

<body>

<form name="FormAddress" onsubmit="return Validator()" action="member.asp"
method="post">
<br>
<br>
<br>
<center>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 2 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
html
<img src ="image/signup.gif" width="350" height="50"><br>
<table border="0">
<tr><td width="200" align="right">Member Name :</td><td>
<input type="text" name ="mname" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">Password :</td><td>
<input type="password" name="password" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width="200" align="right">Re-password :</td><td>
<input type="password" name="repassword" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">ชื่อ :</td><td>
<input type="text" name ="fname" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">นามสกุล :</td><td>
<input type="text" name ="lname" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">ที่อยู่ :</td><td>
<textarea name ="address" rows="3" cols="20"></textarea><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">โทรศัพท์ :</td><td>
<input type="text" name ="tel" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width ="200" align="right">E-mail :</td><td>
<input type="text" name ="email" size="20"><FONT
color=#ff0000>*</FONT></td></tr>
<tr><td width="200" align="right"><input type="image" src="image/bt_reset.gif"
width="81" height="22"></td><td>
<input type="image" src="image/bt_signup.gif" width="81"
height="22"></td></tr>
</table>
</center>
</form>
</body>
</html>
```

```
*****
*****
SOURCE CODE ส่วนASP
*****
*****
```

```
<% response.buffer=true %>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">

<html>
<head>
<title>Member Login</title>
</head>

<body>
<%
mname=request ("mname")
password=request ("password")

sql="select mname,password from member where mname='" & Mname & "' and
password='" & password & "'"
set conn=server.createobject ("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath
("\vb\data.mdb" )

set rs=server.createobject ("adodb.recordset")
rs.open sql,conn,1,3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 3 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
if rs.eof then
    response.write("<center>Login ใหม่</center>")
else
    session("x")=mname
    response.redirect("menu.asp")
end if
%>
```

```
</body>
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<html>
<head>
    <title>Member Area</title>
</head>
```

```
<body>
<%
```

```
mname=request("mname")
password=request("password")
Fname=request("fname")
Lname=request("lname")
address=request("address")
tel=request("tel")
email=request("email")
s="insert into member values('" & mname & "','" & password & "','" & fname &
"', '" & lname & "','" & address & "','" & tel & "','" & email & "')"
sql="select mname from member where mname='" & Mname & "'"
set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath
("\vb\data.mdb" )
```

```
set rs=server.createobject("adodb.recordset")
rs.open sql,conn,1,3
```

```
if rs.eof then
    set rs2=server.createobject("adodb.recordset")
    rs2.open s,conn,1,3
    response.write("<center>ได้รับข้อมูลของคุณ " & fname & " เรียบร้อยแล้ว</center> ")
else
```

```
    response.write("<center>มีคนอื่นแล้ว กรุณาเปลี่ยน Member Name ใหม่</center>")
end if
```

```
%>
```

```
</body>
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<html>
<head>
<script language="JavaScript">
<!--
```

```
function resetIt() {
// Calculate Time
var timerID = null;
var timerRunning = false;
if(timerRunning)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 4 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
clearTimeout(timerID);
timerRunning = false;
// getTime
var timeNow = new Date();
var hours = timeNow.getHours();
var minutes = timeNow.getMinutes();
var seconds = timeNow.getSeconds();
var timeValue = "" + ((hours >12) ? hours -12 :hours)
timeValue = ((timeValue <10)? "0":"") + timeValue
timeValue += ((minutes < 10) ? ":0" : ":") + minutes
timeValue += ((seconds < 10) ? ":0" : ":") + seconds
timeValue += (hours >= 12) ? " PM" : " AM"
timerID = setTimeout("resetIt()",100);
timerRunning = true;
// getDate
var dateNow = new Date();
var days = new Array('อาทิตย์', 'จันทร์', 'อังคาร', 'พุธ', 'พฤหัสบดี', 'ศุกร์', 'เสาร์');
var months = new Array('มกราคม', 'กุมภาพันธ์', 'มีนาคม', 'เมษายน', 'พฤษภาคม', 'มิถุนายน', 'กรกฎาคม', 'สิงหาคม', 'กันยายน', 'ตุลาคม', 'พฤศจิกายน', 'ธันวาคม');
var date = ((dateNow.getDate()<10) ? "0" : "")+ dateNow.getDate();
function y2k(number){return (number < 1000) ? number + 1900 : number;}
// compileIt
today = timeValue + " " + days[dateNow.getDay()] + " " +
months[dateNow.getMonth()] + ", " +
date + " " +
(y2k(dateNow.getYear()));
if(document.all || document.getElementById){ // Browser Check
document.title = today.toString();
}else{
self.status = today.toString(); // Default to status.
}
}
resetIt();
//-->
</script>
<title>Inet Control</title>
</head>
<body>
<%
set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath
("\vb\data.mdb" )
' Load ค่าTimer
set rs=server.createobject("adodb.recordset")
sql="select * from Timer"
rs.open sql,conn,1,3

if rs("timer1") then t1="checked"
if rs("timer2") then t2="checked"
if rs("timer3") then t3="checked"
if rs("timer4") then t4="checked"
if rs("timer5") then t5="checked"
if rs("timer6") then t6="checked"
if rs("timer7") then t7="checked"
if rs("timer8") then t8="checked"

' เช็ค Status ของPort
set rsStatus=server.createobject("adodb.recordset")
sql="select * from status"
rsstatus.open sql,conn,1,3

if rsstatus("ch1")=11 then
ch1="ON"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 5 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
else
    ch1="OFF"
end if

if rsstatus("ch2")=12 then
    ch2="ON"
else
    ch2="OFF"
end if

if rsstatus("ch3")=13 then
    ch3="ON"
else
    ch3="OFF"
end if

if rsstatus("ch4")=14 then
    ch4="ON"
else
    ch4="OFF"
end if

if rsstatus("ch5")=15 then
    ch5="ON"
else
    ch5="OFF"
end if

if rsstatus("ch6")=16 then
    ch6="ON"
else
    ch6="OFF"
end if

if rsstatus("ch7")=17 then
    ch7="ON"
else
    ch7="OFF"
end if

if rsstatus("ch8")=18 then
    ch8="ON"
else
    ch8="OFF"
end if
```

%>

```
<FORM action=outlet.asp method=post>
<center>
<table width=95% cellspacing="1" cellpadding="1" border="0">
<tr >
    <td colspan=4 ><strong><b><font color="Blue"><h1>Inet Control
: </h1></font></b></strong></td>

</tr>

<tr >
    <td bgcolor=#ff6600><font color="White">Control Out Port</font></td>
    <td bgcolor=#ff6600><font color="White">Status Port</font></td>
    <td bgcolor=#ff6600><font color="White">Timer On</font></td>
    <td bgcolor=#ff6600><font color="White">Timer Start</font></td>
    <td bgcolor=#ff6600><font color="White">Timer Stop</font></td>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 6 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
</tr>
<tr>
  <td><input type="submit" value="CH1 ON" name="method"> <input type
="submit" value="CH1 OFF" name="method"></td>
  <td><%=ch1%></td>
  <td><input type="checkbox" name="Timer1" value="1" <%=t1%>></td>
  <td><input type="text" name="T1start" value='<%=rs("t1start")%>'></td>
  <td><input type="text" name="T1stop" value='<%=rs("t1stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH2 ON" name="method"> <input type
="submit" value="CH2 OFF" name="method"></td>
  <td><%=ch2%></td>
  <td><input type="checkbox" name="Timer2" value="2" <%=t2%>></td>
  <td><input type="text" name="T2start" value='<%=rs("t2start")%>'></td>
  <td><input type="text" name="T2stop" value='<%=rs("t2stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH3 ON" name="method"> <input type
="submit" value="CH3 OFF" name="method"></td>
  <td><%=ch3%></td>
  <td><input type="checkbox" name="Timer3" value="3" <%=t3%>></td>
  <td><input type="text" name="T3start" value='<%=rs("t3start")%>'></td>
  <td><input type="text" name="T3stop" value='<%=rs("t3stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH4 ON" name="method"> <input type
="submit" value="CH4 OFF" name="method"></td>
  <td><%=ch4%></td>
  <td><input type="checkbox" name="Timer4" value="4" <%=t4%>></td>
  <td><input type="text" name="T4start" value='<%=rs("t4start")%>'></td>
  <td><input type="text" name="T4stop" value='<%=rs("t4stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH5 ON" name="method"> <input type
="submit" value="CH5 OFF" name="method"></td>
  <td><%=ch5%></td>
  <td><input type="checkbox" name="Timer5" value="5" <%=t5%>></td>
  <td><input type="text" name="T5start" value='<%=rs("t5start")%>'></td>
  <td><input type="text" name="T5stop" value='<%=rs("t5stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH6 ON" name="method"> <input type
="submit" value="CH6 OFF" name="method"></td>
  <td><%=ch6%></td>
  <td><input type="checkbox" name="Timer6" value="6" <%=t6%>></td>
  <td><input type="text" name="T6start" value='<%=rs("t6start")%>'></td>
  <td><input type="text" name="T6stop" value='<%=rs("t6stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH7 ON" name="method"> <input type
="submit" value="CH7 OFF" name="method"></td>
  <td><%=ch7%></td>
  <td><input type="checkbox" name="Timer7" value="7" <%=t7%>></td>
  <td><input type="text" name="T7start" value='<%=rs("t7start")%>'></td>
  <td><input type="text" name="T7stop" value='<%=rs("t7stop")%>'></td>
</tr>
<tr>
  <td><input type="submit" value="CH8 ON" name="method"> <input type
="submit" value="CH8 OFF" name="method"></td>
  <td><%=ch8%></td>
```

```

html
<td><input type="checkbox" name="Timer8" value="8" <%=t8%>></td>
<td><input type="text" name="T8start" value='<%=rs("t8start")%>'></td>
<td><input type="text" name="T8stop" value='<%=rs("t8stop")%>'></td>
</tr>
<tr >
<td bgcolor=#ff6600><font color=#ff6600>.</font></td>
<td bgcolor=#ff6600><font color=#ff6600>.</font></td>
<td bgcolor=#ff6600><font color=#ff6600>.</font></td>
<td bgcolor=#ff6600><font color=#ff6600>.</font></td>
<td bgcolor=#ff6600><font color=#ff6600>.</font></td>
</tr>
<tr>
<td><input type ="submit" value="ALL ON" name="method"> <input type ="submit"
value="ALL OFF" name="method"> </td>
<td></td>
<td></td>
<td></td>
<td><input type ="submit" value=" Apply " name="method"></td>
</tr>
</table>
<br>
</FORM>
</center>
</body>
</html>

```

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
<title>Inet Control</title>
</head>
<body>
<%
method=request("method")
set conn=server.createobject("adodb.connection")
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath
("\vb\data.mdb" )

set rs=server.createobject("adodb.recordset")
sql="select * from outlet"
rs.open sql,conn,1,3

select case method
case "ALL ON"
allon()
case "ALL OFF"
alloff()
case " Apply "
apply()
case "CH1 ON"
chl1on()
case "CH1 OFF"
chl1off()
case "CH2 ON"
ch2on()
case "CH2 OFF"
ch2off()

```

html

```
case "CH3 ON"
    ch3on()
case "CH3 OFF"
    ch3off()
case "CH4 ON"
    ch4on()
case "CH4 OFF"
    ch4off()
case "CH5 ON"
    ch5on()
case "CH5 OFF"
    ch5off()
case "CH6 ON"
    ch6on()
case "CH6 OFF"
    ch6off()
case "CH7 ON"
    ch7on()
case "CH7 OFF"
    ch7off()
case "CH8 ON"
    ch8on()
case "CH8 OFF"
    ch8off()
```

end select

```
response.write("<center>บันทึกคำสั่งเรียบร้อยแล้ว<br><a href='Menu.asp'>กลับ</a></center>")
```

```
sub apply()
```

```
set rst=server.createobject("adodb.recordset")
```

```
sql="select * from timer"
```

```
rst.open sql,conn,1,3
```

```
if request("timer1")="1" then
```

```
    rst("timer1")=true
```

```
else
```

```
    rst("timer1")=false
```

```
end if
```

```
if request("timer2")="2" then
```

```
    rst("timer2")=true
```

```
else
```

```
    rst("timer2")=false
```

```
end if
```

```
if request("timer3")="3" then
```

```
    rst("timer3")=true
```

```
else
```

```
    rst("timer3")=false
```

```
end if
```

```
if request("timer4")="4" then
```

```
    rst("timer4")=true
```

```
else
```

```
    rst("timer4")=false
```

```
end if
```

```
if request("timer5")="5" then
```

```
    rst("timer5")=true
```

```
else
```

```
    rst("timer5")=false
```

```
end if
```

```
if request("timer6")="6" then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
rst("timer6")=true
else
rst("timer6")=false
end if

if request("timer7")="7" then
rst("timer7")=true
else
rst("timer7")=false
end if

if request("timer8")="8" then
rst("timer8")=true
else
rst("timer8")=false
end if
```

'บันทึกเวลา

```
rst("t1start")=request("T1start")
rst("t2start")=request("T2start")
rst("t3start")=request("T3start")
rst("t4start")=request("T4start")
rst("t5start")=request("T5start")
rst("t6start")=request("T6start")
rst("t7start")=request("T7start")
rst("t8start")=request("T8start")
```

```
rst("t1stop")=request("T1stop")
rst("t2stop")=request("T2stop")
rst("t3stop")=request("T3stop")
rst("t4stop")=request("T4stop")
rst("t5stop")=request("T5stop")
rst("t6stop")=request("T6stop")
rst("t7stop")=request("T7stop")
rst("t8stop")=request("T8stop")
```

```
rst.update
end sub
```

```
sub ALLON()
```

```
rs("ch1")=11
rs("ch2")=12
rs("ch3")=13
rs("ch4")=14
rs("ch5")=15
rs("ch6")=16
rs("ch7")=17
rs("ch8")=18
```

```
rs.update
```

```
end sub
```

```
sub ALLOFF()
```

```
rs("ch1")=1
rs("ch2")=2
rs("ch3")=3
rs("ch4")=4
rs("ch5")=5
rs("ch6")=6
rs("ch7")=7
rs("ch8")=8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 10 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
rs.update
```

```
end sub
```

```
sub ch1on()  
rs("ch1")=11  
rs.update  
end sub
```

```
sub ch1off()  
rs("ch1")=1  
rs.update  
end sub
```

```
sub ch2on()  
rs("ch2")=12  
rs.update  
end sub
```

```
sub ch2off()  
rs("ch2")=2  
rs.update  
end sub
```

```
sub ch3on()  
rs("ch3")=13  
rs.update  
end sub
```

```
sub ch3off()  
rs("ch3")=3  
rs.update  
end sub
```

```
sub ch4on()  
rs("ch4")=14  
rs.update  
end sub
```

```
sub ch4off()  
rs("ch4")=4  
rs.update  
end sub
```

```
sub ch5on()  
rs("ch5")=15  
rs.update  
end sub
```

```
sub ch5off()  
rs("ch5")=5  
rs.update  
end sub
```

```
sub ch6on()  
rs("ch6")=16  
rs.update  
end sub
```

```
sub ch6off()  
rs("ch6")=6  
rs.update  
end sub
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 11 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

```
sub ch7on()  
rs("ch7")=17  
rs.update  
end sub
```

```
sub ch7off()  
rs("ch7")=7  
rs.update  
end sub
```

```
sub ch8on()  
rs("ch8")=18  
rs.update  
end sub
```

```
sub ch8off()  
rs("ch8")=8  
rs.update  
end sub
```

```
%>  
</body>  
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<html>  
<head>  
  <title>Inet Control</title>  
</head>
```

```
<body>  
<%  
method=request("method")  
  
set conn=server.createobject("adodb.connection")  
conn.open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" & Server.MapPath  
("data.mdb" )
```

```
set rs=server.createobject("adodb.recordset")  
sql="select * from outlet"  
rs.open sql,conn,1,3
```

```
if method="ALL ON" then  
  ALLON()  
else  
  ALLOFF()  
end if
```

```
sub ALLON()
```

```
rs("ch1")=11  
rs("ch2")=12  
rs("ch3")=13  
rs("ch4")=14  
rs("ch5")=15  
rs("ch6")=16  
rs("ch7")=17  
rs("ch8")=18
```

```
rs.update  
response.write("บันทึกคำสั่งเรียบร้อยแล้ว")
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

html

end sub

```
sub ALLOFF()  
rs("ch1")=1  
rs("ch2")=2  
rs("ch3")=3  
rs("ch4")=4  
rs("ch5")=5  
rs("ch6")=6  
rs("ch7")=7  
rs("ch8")=8
```

```
rs.update  
response.write("บันทึกคำสั่งเรียบร้อยแล้ว")  
end sub
```

%>

```
<form action="menu.asp" method="post">
```

```
<input type="submit" value="ALL ON" name="method">
```

```
<input type="submit" value="ALL OFF" name="method">
```

```
</form>
```

```
</body>
```

```
</html>
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งาน Page 13 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. ชัยวัฒน์ ลิ้มพรจิตรวิไลย์, “คู่มืออิเล็กทรอนิกส์”, บริษัท ซีเอ็ดยูเคชั่นจำกัด(มหาชน)
2. ปรมะรัฐ ประณยานันท์ และ ปิยะพงษ์ เผ่าวณิช, “คู่มือและการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ MCS-51”, บริษัท ซีเอ็ดยูเคชั่นจำกัด(มหาชน)
3. “PC-Serial Port Interfacing text-lab manual”, Innovative Experiment
4. AT89C2051 Datasheet (*.pdf)
5. Maxim Multichannel RS-232 Datasheet (*.pdf)
6. ธนพล ฉันทวีชัย, “การพัฒนา ASP ด้วยฐานข้อมูล Access97/2000”, บริษัท ซีเอ็ดยูเคชั่นจำกัด (มหาชน)
7. ชาริน สิริธรรมชารี, “คู่มือการเขียนโปรแกรม Microsoft Visual Basic 6.0”, บริษัท ซัคเซส มีเดีย จำกัด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้