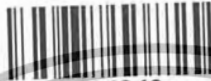


สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือและอินเทอร์เน็ต

Electric Appliances Controlled by Mobile Phone and Internet



T104340



เลขหมู่.....  
เลขทะเบียน.....104340  
วัน,เดือน,ปี.....- 2 พ.ศ. 2552

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

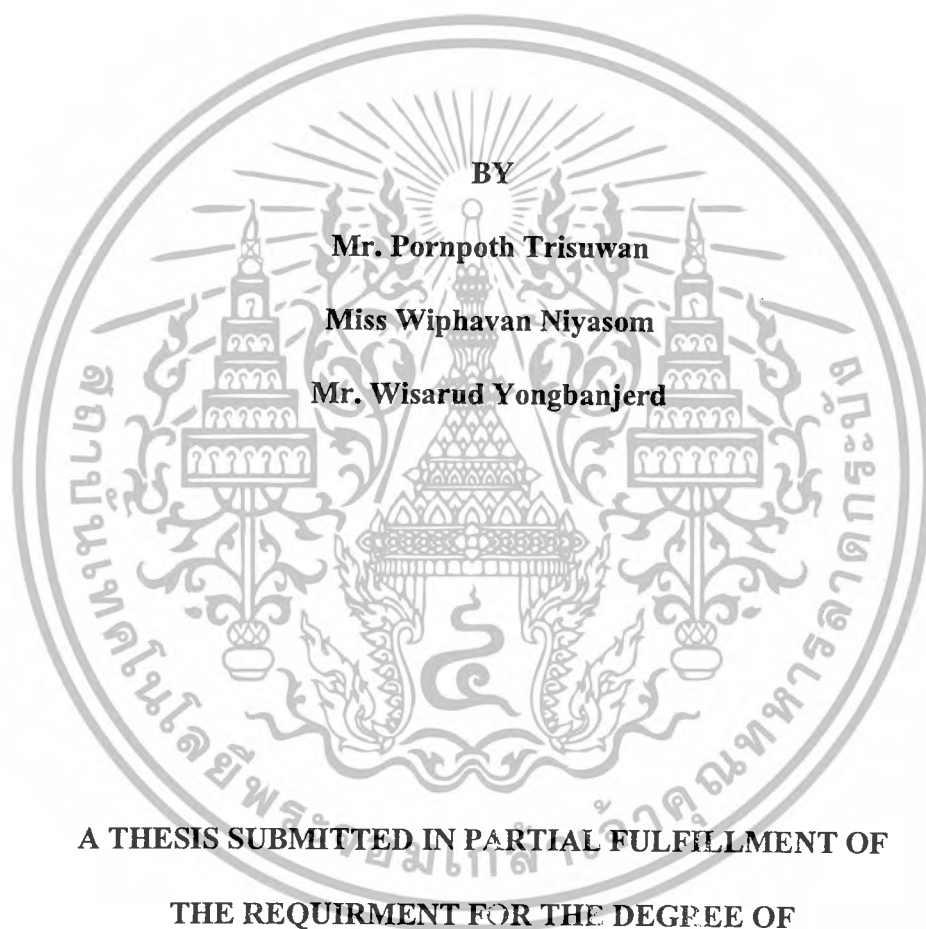
คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2551

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Electric Appliances Controlled by Mobile Phone and Internet**



**BY**

**Mr. Pornpoth Trisuwan**

**Miss Wiphavan Niyasom**

**Mr. Wisarud Yongbanjerd**

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF**

**THE REQUIREMENT FOR THE DEGREE OF**

**BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING**

**FACULTY OF ENGINEERING**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**2018**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือและอินเทอร์เน็ต  
Electric Appliances Controlled by Mobile Phone and Internet

นักศึกษา นายพรพจน์ ไตรสุวรรณ รหัสนักศึกษา 48012102  
นางสาววิภาวรรณ นิยะโสเม รหัสนักศึกษา 48012118  
นายวิศรุต ยงบรรเจิด รหัสนักศึกษา 48012119

อาจารย์ที่ปรึกษา ผศ. ไพศาล สิริโยภาสกุล  
ดร. พนารัตน์ เชิญถนอมวงศ์

ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต

ภาควิชา วิศวกรรมสารสนเทศ

ปีการศึกษา 2551

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้  
นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

อาจารย์ที่ปรึกษา

(ผศ. ไพศาล สิริโยภาสกุล)

 อาจารย์ที่ปรึกษา

(ดร.พนารัตน์ เชิญถนอมวงศ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือและอินเทอร์เน็ต  
Electric Appliances Controlled by Mobile Phone and Internet

นักศึกษา นายพรพจน์ ไตรสุวรรณ รหัสนักศึกษา 48012102  
นางสาววิภาวรรณ นิยะโสสม รหัสนักศึกษา 48012118  
นายวิศรุต ขงบรรเจิด รหัสนักศึกษา 48012119

อาจารย์ที่ปรึกษา ผศ.ไพศาล สิทธิโยภาสกุล

ดร.พนารัตน์ เชิญถนอมวงศ์

ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต

สาขา วิศวกรรมสารสนเทศ

ภาควิชา วิศวกรรมสารสนเทศ

ปีการศึกษา 2551

บทคัดย่อ

โครงการนี้เป็นโครงการที่จัดทำขึ้นเพื่อพัฒนาอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆโดยคำนึงถึงการอำนวยความสะดวกให้แก่ผู้ใช้งาน ซึ่งนำเอาอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆมาประยุกต์ใช้กับโทรศัพท์มือถือและอินเทอร์เน็ต(Internet) เพื่อใช้ควบคุมอุปกรณ์เหล่านี้ผ่านไมโครคอนโทรลเลอร์ (AR7 LPC2118) และโปรโตคอล(TCP/IP)ซึ่งเป็นตัวกลางในการควบคุม โดยอุปกรณ์เครื่องใช้ไฟฟ้าเหล่านี้จะติดต่อกับโทรศัพท์มือถือและอินเทอร์เน็ตผ่าน Serial Port (RS-232)

**Thesis Title** Electric Appliances Controlled by Mobile Phone and Internet

**Student** Mr.Pornpoth Trisuwan Student ID. 48012102

MissWiphavan Niyasom Student ID. 48012118

Mr.Wisarud Yongbanjerd Student ID. 48012119

**Advisor** Asst. Prof. Pisan Sidthiyopasakul

Dr. Panarat Chemtanomwng

**Graduate Level** Bachelor Degree of Information Engineering

**Department** Information Engineering

**Academic Year** 2008



**ABSTRACT**

This project proposes the electronic devices control system that is convenient for the user. This system is controlled by internet and the mobile phone via the TCP/IP protocol together with the microcontroller MCS-51. The electronic device will be connected to mobile phone and internet through serial port (RS-232)

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ไม่สามารถสำเร็จรุด่วงได้หากขาดความช่วยเหลือจากอาจารย์ที่ปรึกษา ทั้งสอง ผศ. ไพศาล สิริโยภาสกุล และ ดร.พนารัตน์ เจริญนอมวงศ์ ที่ได้ให้คำปรึกษาในด้านต่างๆ รวมไปถึงแนวคิดและวิธีการในการแก้ปัญหาและขอขอบคุณ บิดา มารดา ที่ให้ความช่วยเหลือ และเป็นกำลังใจให้ข้าพเจ้าเสมอมา ขอขอบคุณเพื่อนร่วมห้อง และ เพื่อนกลุ่ม FS ที่มีกันและกันตลอดไปคอยช่วยเหลือ เป็นกำลังใจ ซึ่งกันและกัน

และจะขาดเสียมิได้ขอขอบคุณเพื่อนคณะผู้จัดทำที่ได้ให้ความร่วมมือด้วยดีเสมอมาและจะเป็นความทรงจำแก่พวกเราตลอดไป



## สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญรูป	ฉ
สารบัญตาราง	ณ
บทที่ 1 บทนำ	1
1.1 แนวคิดและที่มา	1
1.2 วัตถุประสงค์ของโครงการ	2
1.3 ขอบเขตของโครงการ	2
1.4 โครงสร้างโดยรวมของระบบ	3
บทที่ 2 ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง	5
2.1 โครงสร้างการทำงานของ WEB	5
2.2 จีพีอาร์เอส (GPS)	5
2.3 เอ็ดจ์ (EDGE)	8
2.4 Macromedia Dreamweaver 8	19
2.5 Appserv	22
2.6 ความรู้เบื้องต้นเกี่ยวกับ Visual Studio 2008	25
2.7 ASP.NET	33
2.8 Microsoft Access	40

## สารบัญ(ต่อ)

เรื่อง	หน้า
2.9 Flash Magic	48
2.10 ไอซีไมโครคอนโทรลเลอร์ MCS-51	53
2.11 ET-OPTO RELAY4	69
2.12 การใช้งาน RS 232	72
<b>บทที่ 3 การออกแบบ</b>	<b>75</b>
3.1 ภาพโดยรวมของระบบ	75
3.2 การออกแบบ	76
3.3 การออกแบบ Web Site โดยใช้ภาษา VB.NET	76
3.4 การออกแบบฐานข้อมูล	85
<b>บทที่ 4 การทดลองและผลการทดลอง</b>	<b>86</b>
4.1 ขั้นตอนการทดลอง	86
<b>บทที่ 5 สรุปผลการทดลองและแนวทางในการพัฒนา</b>	<b>91</b>
5.1 สรุปผลการพัฒนาโครงการ	91
5.2 ปัญหาที่เกิดขึ้นระหว่างการพัฒนา	91
5.3 แนวทางในการพัฒนาโครงการ	92
<b>บรรณานุกรม</b>	<b>92</b>
<b>ภาคผนวก</b>	<b>93</b>

## สารบัญรูปภาพ

รูป	หน้า
รูปที่ 1.1 โครงสร้างโดยรวมของระบบ	4
รูปที่ 2.1 โมเดลการทำงานของ WEB	5
รูปที่ 2.2 โครงสร้างของ จีพีอาร์เอส	7
รูปที่ 2.3 เครื่องข่ายโทรศัพท์เคลื่อนที่ GSM	8
รูปที่ 2.4 การรวมกันของ FDMA และ TDMA	9
รูปที่ 2.5 เทคโนโลยี EDGE ในเครือข่ายโทรศัพท์เคลื่อนที่	10
รูปที่ 2.6 แผนภาพแทนสัญลักษณ์ 8 PSK เทียบกับตำแหน่งบิตข้อมูล	11
รูปที่ 2.7 การเข้ารหัส MCS-1 ถึง MCS-4	12
รูปที่ 2.8 การเข้ารหัส MCS-5 และ MCS-6	13
รูปที่ 2.9 การเข้ารหัส MCS-7, MCS-8 และ MCS-9	13
รูปที่ 2.10 โปรแกรม Dreamweaver 8	21
รูปที่ 2.11 โครงสร้างระบบจัดเก็บไฟล์หลังการติดตั้ง Appserv	23
รูปที่ 2.12 หน้าต่างของ Visual Studio 2008	27
รูปที่ 2.12 ก. ส่วนที่ใช้ออกแบบการแสดงผลเพื่อติดต่อใช้งานกับผู้ใช้	28
รูปที่ 2.12 ข. Project Explorer แบบ โปรเจกต์เดียวและแบบหลายโปรเจกต์	28
รูปที่ 2.12 ค. Properties Window	29
รูปที่ 2.13 การทำงานของหน้าเว็บแบบ ASP.NET	39
รูปที่ 2.14 ตัวอย่าง Microsoft Access	40
รูปที่ 2.15 หน้าต่างของ โปรแกรม	42
รูปที่ 2.16 หน้าต่างการสร้างฐานข้อมูลใหม่	43
รูปที่ 2.17 ที่เก็บไฟล์ข้อมูล	44
รูปที่ 2.18 หน้าต่างบันทึก	45
รูปที่ 2.19 ตรวจสอบสถานะชื่อเพิ่มข้อมูล	45
รูปที่ 2.20 การเชื่อมโยงตาราง	47
รูปที่ 2.21 หน้าต่างวิธีการเรียกใช้	49

## สารบัญรูปภาพ(ต่อ)

รูป	หน้า
รูปที่ 2.22 ลักษณะ โปรแกรม Flash Magic	50
รูปที่ 2.23 หน่วยความจำสำหรับเก็บ โปรแกรม (Program Memory)	55
รูปที่ 2.24 ตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน (Internal Data Memory)	56
รูปที่ 2.25 พื้นที่ในหน่วยความจำข้อมูล (RAM)	58
รูปที่ 2.26 การจัดการหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ	59
รูปที่ 2.27 แฟล็กต่างๆของรีจิสเตอร์ PSW	60
รูปที่ 2.28 ก. Carry Flag	61
รูปที่ 2.28 ข. Auxiliary Carry Flag	61
รูปที่ 2.29 Flow Chart การเขียน โปรแกรมภาษาแอสเซมบลี MCS-51	64
รูปที่ 2.30 Flow Chart การเขียนภาษาแอสเซมบลี	65
รูปที่ 2.31 โครงสร้างของบอร์ด ET-OPTO RELAY4	70
รูปที่ 2.32 วงจรของ ET-OPTO RELAY4 ขนาด 1 ช่อง	70
รูปที่ 2.34 การจัดเรียงสัญญาณของขั้ว IDE-10 Pin และ CPA-6 Pin	72
รูปที่ 2.35 วงจรของ ET-OPTO RELAY4	72
รูปที่ 2.36 ช่องสัญญาณใช้เชื่อมต่อกับ RS 232 เพื่อรับส่งข้อมูล	73
รูปที่ 2.37 ตัวอย่าง โปรแกรมกำหนดค่า UART	73
รูปที่ 2.38 วงจร Cable สำหรับ RS 232	74
รูปที่ 3.1 Diagram ภาพโดยรวมของโครงการ	75
รูปที่ 3.2 ขั้นตอนการทำงานของ Web Application	77
รูปที่ 3.3 Flow Chart การทำงานของการเข้าสู่ระบบที่งานหลัก	78
รูปที่ 3.4 Flow Chart การทำงานในส่วนของการเปลี่ยนแปลงสถานะของ Device	79
รูปที่ 3.5 Flow Chart การทำงานในส่วนของ Schedule	80
รูปที่ 3.6 Flow Chart การทำงานในส่วนของการเปลี่ยนสถานะของ Device แบบตั้งเวลา	81
รูปที่ 3.7 Flow Chart การทำงานในส่วนของการ Add Schedule	82
รูปที่ 3.8 Flow Chart การทำงานในส่วนของการ View Schedule	83
รูปที่ 3.9 Flow Chart การทำงานในส่วนของ History	84
รูปที่ 4.1 หน้าต่างเว็บ no-ip.com เพื่อขอใช้บริการ NO-IP Plus	85

## สารบัญรูปภาพ(ต่อ)

รูป	หน้า
รูปที่ 4.2 การเชื่อมต่อของเครื่องเซิร์ฟเวอร์	87
รูปที่ 4.3 แสดงหน้า Index	87
รูปที่ 4.4 หน้าควบคุมอุปกรณ์ไฟฟ้าออนไลน์	88
รูปที่ 4.5 อุปกรณ์ไฟฟ้าสามารถเปิดปิด	89
รูปที่ 4.6 หน้าระบบข้อมูลย้อนหลัง	89
รูปที่ 4.7 หน้า Index บน PDA Phone	90
รูปที่ 4.8 หน้าควบคุมอุปกรณ์ไฟฟ้าผ่าน PDA Phone และ การแสดงผลของเครื่องใช้ไฟฟ้า	90



## สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 แสดงค่าอัตราการรับส่งกับวิธีการ Modulation เทียบกับวิธีการเข้ารหัสข้อมูล	12
ตารางที่ 2.2 เปรียบเทียบมาตรฐาน โทรศัพท์เคลื่อนที่ตั้งแต่ยุค 2.5G ขึ้น ไปทุกระบบ	16
ตารางที่ 2.3 ส่วนประกอบของ โปรเจค	29



# บทที่ 1

## บทนำ

ในยุคปัจจุบันที่การสื่อสารมีวิวัฒนาการความก้าวหน้าอย่างต่อเนื่องจึงไม่อาจหลีกเลี่ยงข้อมูลและข่าวสารที่มีการกระจายจากที่หนึ่งไปยังที่หนึ่งอย่างรวดเร็ว เพื่อให้ทันกับความต้องการของผู้ใช้ซึ่งสิ่งหนึ่งที่เป็นตัวแปรผลักดันที่สำคัญในการรับส่งและกระจายข้อมูลข่าวสารนั้น คือ อินเทอร์เน็ต จะเห็นได้ว่าอินเทอร์เน็ตได้เข้ามามีบทบาทกับชีวิตประจำวัน ไม่ว่าจะเป็นการรับ-ส่งE-mail การค้นหาข้อมูลต่างๆ การสั่งซื้อสินค้าและบริการผ่านทางเว็บไซต์ (Web site) เป็นต้น ซึ่งนับวันอินเทอร์เน็ตได้ทำหน้าที่อำนวยความสะดวกในการสื่อสารข้อมูลข่าวสารได้อย่างรวดเร็วและมีประสิทธิภาพ แต่อินเทอร์เน็ตยังมีข้อจำกัดบางประการในการใช้งานโดยเฉพาะอย่างยิ่งเมื่อไม่สามารถที่จะนำคอมพิวเตอร์ส่วนบุคคล (PC) และคอมพิวเตอร์แบบพกพา (Notebook) ไปได้ในทุกที่จึงได้เห็นความสำคัญของปัญหาและเกิดแนวคิดที่จะนำเอาโทรศัพท์มือถือ (PDA Phone) ที่ปัจจุบันสามารถพกพาไปได้ทุกที่มาเป็นเทคโนโลยีในการเชื่อมต่องานร่วมกับนำเทคโนโลยีของ เอ็ดจ์ (Edge) หรือ จีพีอาร์เอส (GPRS) ที่ใช้งานบนโทรศัพท์มือถือ เพื่อให้เกิดความคล่องตัวในการติดต่อสื่อสารแทนอินเทอร์เน็ตที่อาจมีข้อจำกัดบางประการในการใช้งาน

### 1.1 แนวคิดและที่มา

เนื่องจากข้อจำกัดของการควบคุมอุปกรณ์ไฟฟ้าภายในที่พักอาศัยหรือออฟฟิศ สามารถที่จะกระทำได้ในบริเวณที่ที่พักอาศัยหรือบริเวณที่สัญญาณรีโมทไปถึงเท่านั้น ไม่สามารถควบคุมจากระยะไกลได้ ซึ่งเป็นอุปสรรคที่สำคัญ ในกรณีที่ลืมปิดหรือเปิดอุปกรณ์ไฟฟ้าเหล่านั้นเมื่อเราไม่อยู่ที่ที่พักอาศัยหรือออฟฟิศจึงได้เกิดแนวคิดที่จะนำเอาระบบอินเทอร์เน็ต ซึ่งเป็นระบบการสื่อสารที่มีเครือข่ายเชื่อมโยงไปทั่วโลก มาใช้ในการควบคุมอุปกรณ์ไฟฟ้าภายในที่พักอาศัยหรือในออฟฟิศ เนื่องจากสามารถเข้าถึงการควบคุมอุปกรณ์ไฟฟ้าได้หลายทางและหลายรูปแบบ เช่น ผ่านทางโทรศัพท์มือถือโดยใช้ เอ็ดจ์ หรือ จีพีอาร์เอส หรือ ผ่านทาง Web Application ทำให้เกิดความสะดวกและความปลอดภัย ในกรณีที่เราลืมปิดหรือเปิดอุปกรณ์ไฟฟ้า อีกทั้งยังสามารถตั้งเวลาเปิด-ปิดอุปกรณ์ไฟฟ้าได้ตามต้องการ นับ เป็นสิ่ง ที่มีประโยชน์อย่างมากสำหรับผู้ใช้งานอุปกรณ์ไฟฟ้าในปัจจุบันที่อยู่ในเวลาเร่งด่วน

## 1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 วิเคราะห์และออกแบบการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านหรือออฟฟิศผ่านอินเทอร์เน็ตและPDAphone
- 1.2.2 ศึกษาไมโครคอนโทรลเลอร์ตระกูล MCS-51 เพื่อนำมาใช้ควบคุมอุปกรณ์ไฟฟ้า
- 1.2.3 ออกแบบและศึกษาการทำงานของภาษา VB.NET เพื่อนำมาพัฒนาเป็น โปรแกรมที่ใช้ติดต่อและสั่งงานอุปกรณ์ไฟฟ้า
- 1.2.4 ออกแบบโปรแกรม MS Access เพื่อใช้เก็บข้อมูลลงฐานข้อมูล
- 1.2.5 เพื่อความสะดวกและประหยัดเวลาในการควบคุมอุปกรณ์ไฟฟ้าเมื่อไม่ได้อยู่บ้าน
- 1.2.6 เพื่อที่จะนำความรู้ที่ได้จากการศึกษาโครงการนี้ไปพัฒนาและประยุกต์ใช้ในอนาคตต่อไป

## 1.3 ขอบเขตของโครงการ

- 1.3.1 ควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าภายในบริเวณบ้านหรือออฟฟิศผ่านอินเทอร์เน็ตและโทรศัพท์มือถือได้
- 1.3.2 สามารถตั้งเวลาเปิด-ปิดอุปกรณ์ไฟฟ้าผ่าน Web Application เพื่อส่งคำสั่งไปยัง Hardware แล้วทำการควบคุมอุปกรณ์ไฟฟ้า
- 1.3.3 สามารถแสดงสถานะปัจจุบันของอุปกรณ์ไฟฟ้า ได้ว่ามีสถานะใดเปิดหรือปิดอยู่ผ่านทาง Web Browser
- 1.3.4 สามารถเก็บข้อมูล และเรียกดูการสั่งงานย้อนหลังที่ทำลงไป แล้วบันทึกลงฐานข้อมูลเพื่อเรียกดูรายการที่ได้ทำงานไปแล้วมาดูได้

โครงการนี้แบ่งออกได้เป็น 2 ส่วนใหญ่ๆ คือ

### 1. ฮาร์ดแวร์(Hardware)

- ไมโครคอนโทรลเลอร์ตระกูล MCS-51ต่อเข้ากับ Computer ที่เป็น Server ผ่านสาย LAN และสายอนุกรม(Serial Port) เพื่อทำหน้าที่ควบคุมสวิสต์เปิด – ปิด อุปกรณ์ไฟฟ้า

### 2. ซอร์ฟแวร์(Software)

- โปรแกรมควบคุมอุปกรณ์ไฟฟ้า
- โปรแกรม VB.net
- Web Application
- เอ็ดจ์/จีพีอาร์เอส
- Keil uVision3
- Flash Magic

### 1.4 โครงสร้างโดยรวมของระบบ

โครงสร้างโดยรวมของระบบจะเริ่มจากผู้ใช้งานที่ต้องการควบคุมการเปิด – ปิด อุปกรณ์ไฟฟ้าในระยะไกลผ่านเครือข่ายอินเทอร์เน็ต โดยผ่าน Web Brower ซึ่งหลักการทำงานคือการสร้าง Web page ขึ้นมาเพื่อใช้ติดต่อกับ Web Server จากนั้น Web Server ก็จะติดต่อไมโครคอนโทรลเลอร์ตระกูล MCS-51 และดาต้าเบส (Database) โดยอาศัยภาษา VB.net เพื่อช่วยในการติดต่อที่รับคำสั่งจาก MS Access เพื่อไปสั่งงานไมโครคอนโทรลเลอร์ให้ควบคุมอุปกรณ์ไฟฟ้าผ่านอินเทอร์เน็ต โดยก่อนที่จะเข้าสู่การควบคุมผู้ใช้จะต้องกรอกชื่อ(User) และรหัสผ่าน (Password) ให้ถูกต้องซึ่งจะมีการแสดงสถานะการควบคุมกลับมายังผู้ใช้พร้อมทั้งมีการบันทึกผลงานข้อมูล



รูปที่ 1.1 โครงสร้างโดยรวมของระบบ

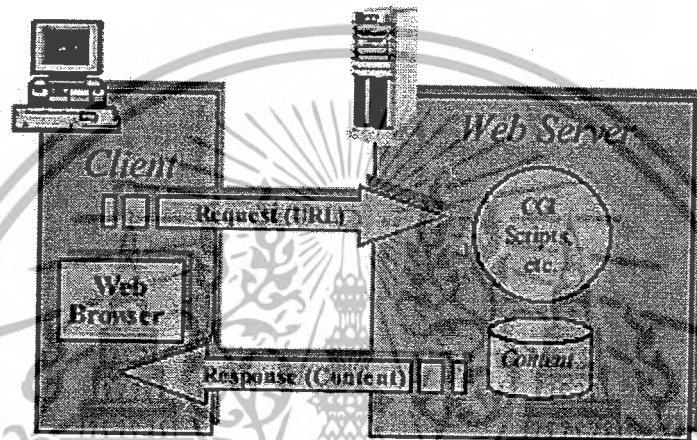
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### ทฤษฎีและเทคโนโลยีที่เกี่ยวข้อง

#### 2.1 โครงสร้างการทำงานของ WEB

โครงสร้างการทำงานของ Web จะเป็นแบบ Client/Server เมื่อ Client ร้องขอ (Request) ข้อมูลไปยัง เซิร์ฟเวอร์ (Server) แล้วเซิร์ฟเวอร์จะส่งข้อมูลตอบกลับ (Response) ไปให้ ดังรูป 2.1



รูปที่ 2.1 โมเดลการทำงานของ WEB

การติดต่อระหว่าง Client หรือ Server ของ Web จะใช้โปรโตคอลมาตรฐานของอินเทอร์เน็ต เช่น HTTP และ TCP/IP เป็นต้น โดย Client หรือ Server จะอยู่ในเครือข่ายเดียวกันหรือเชื่อมกันได้ผ่านโปรโตคอลมาตรฐานของอินเทอร์เน็ต

#### 2.2 จีพีอาร์เอส

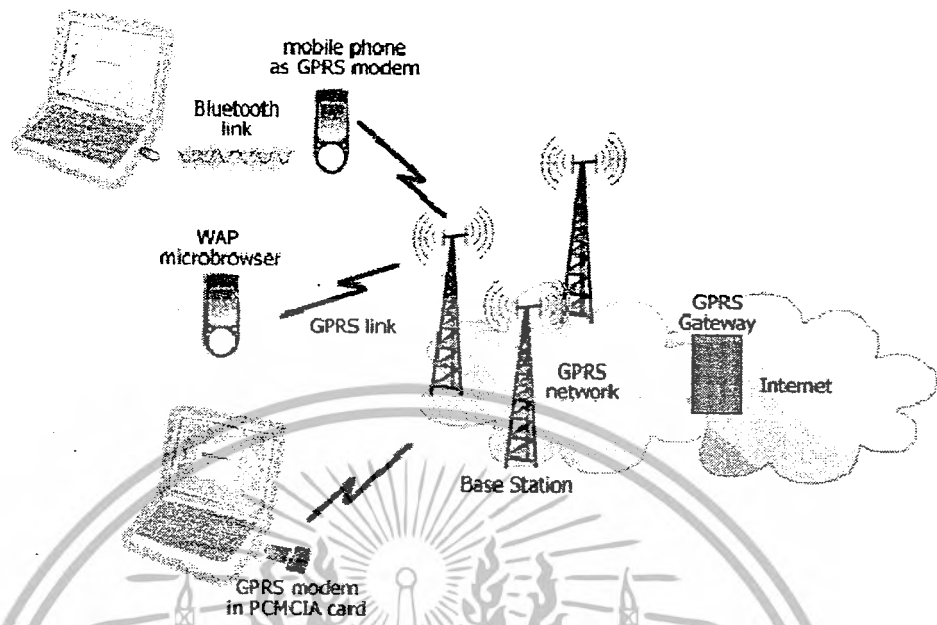
จีพีอาร์เอส หรือ GPRS ย่อมาจาก General Packet Radio Service คือบริการส่งข้อมูลสำหรับโทรศัพท์มือถือแบบ GSM โดย จีพีอาร์เอส มักถูกเรียกเป็นโทรศัพท์มือถือยุค 2.5G ซึ่งอยู่ระหว่าง 2G และ 3G โดยตามทฤษฎีแล้ว จีพีอาร์เอส ใช้ช่องสัญญาณแบบ TDMA ของเครือข่าย GSM ในการส่งข้อมูล และความเร็วสูงสุดของ จีพีอาร์เอส อยู่ที่ประมาณ 171.2 kbps

จีพีอาร์เอส ให้บริการผ่านทาง Radio Interface ระหว่างผู้ใช้งานและปลายทางโดยการให้บริการในรูปแบบ Application Server โดยตัวโทรศัพท์เคลื่อนที่จะถูกแปลงเป็น Packet ซึ่งมี IP Address กำกับอยู่ภายใน ทำให้ไม่เหมือนการส่งแบบเก่า (ระบบ-Radio Frame ที่ใช้กันในการส่งข้อมูลเสียงบนระบบ GSM )

อย่างไรก็ตาม จีพีอาร์เอส ไม่ได้เป็นลักษณะที่จะสามารถให้บริการได้ด้วยตัวของระบบเอง แต่ตัวจีพีอาร์เอส เป็นเพียงแค่ Bearer ให้กับ Application ต่างๆ ที่ต้องการใช้ความเร็วที่เพิ่มมากกว่าปกติในระบบ GSM ที่เคยรองรับอยู่เดิมมาก่อน และระบบ จีพีอาร์เอส ต้องทำการติดต่อไปยัง Packet Data Network ที่เป็น IP Network อีกต่อหนึ่ง ดังนั้นผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ ที่ต้องการเปิดให้ใช้ในระบบ จีพีอาร์เอส ต้องทำการติดตั้งระบบเครือข่าย ที่ประกอบด้วยหน่วยหลักๆ 2 หน่วยด้วยกันคือ

1. SGSN(Serving GPRS Supports Node)
2. GGSN(Gateway GPRS Support Node)

โดยทั้งสองหน่วยหลักขององค์ประกอบนี้จะถูกเชื่อมต่อเข้าด้วยกัน โดยมีอุปกรณ์อื่นๆ เป็นตัวช่วยเพื่อไปร่วมใช้ Radio Interface จาก Base Station โดยผ่านตัวควบคุมที่เรียกว่า PCU (Packet Control Unit ) ที่ติดตั้งไว้ที่ BSC (Base Station Controller) อันทั้งนี้อาจมองได้ว่า จีพีอาร์เอส Network เป็นอีก Network หนึ่ง ซึ่งเข้าถึง Mobile Phone ผ่านทาง Radio Interface ของระบบ GSM Network เดิม โดยเป็นบริการที่เกี่ยวข้องเนื่องกับการรับส่งข้อมูลเป็น Packet โดยตรง



รูปที่ 2.2 โครงสร้างของ จีพีอาร์เอส

### ประโยชน์ของ จีพีอาร์เอส

- การเชื่อมต่อเป็นลักษณะ Always On
- รับข้อมูลในรูปแบบ Video
- รับข้อมูลในรูปแบบ MMS
- ความเร็วในการทำงานดีกว่า GSM
- รับส่งข้อมูลต่างๆ เช่น ข้อมูลธนาคาร e-mail หุ่น ข้อมูล ซอฟต์แวร์ และอื่นๆ อีกมากมาย

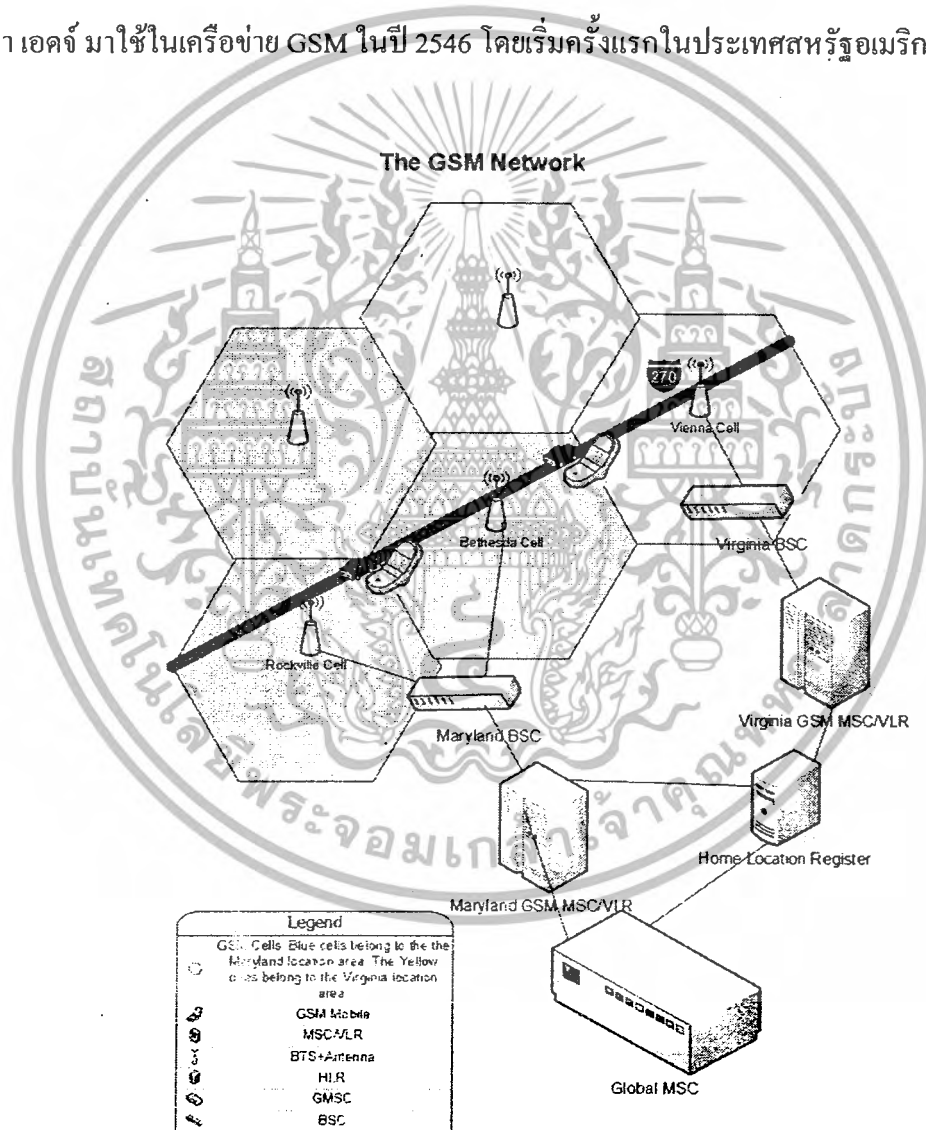
### การคิดค่าใช้จ่ายสำหรับบริการ จีพีอาร์เอส

การคิดค่าบริการนี้ขึ้นอยู่กับผู้ให้บริการซึ่งโดยทั่วไป จะคิดค่าบริการในส่วนของการรับและส่งข้อมูลเท่านั้น โดยคิดเป็นจำนวนต่อ kilobyte (kb) อย่างไรก็ตาม ผู้ให้บริการโทรศัพท์บางค่าย มีการให้บริการแบบเหมาจ่ายด้วย ดังนั้นสามารถเปิดระบบ จีพีอาร์เอส ได้ตลอดเวลาโดยไม่ต้องกลัวว่าจะเสียค่าใช้จ่าย ปัจจุบันผู้ให้บริการ จีพีอาร์เอส คือ AIS, DTAC และ ORANGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3 เอดจ์ (EDGE)

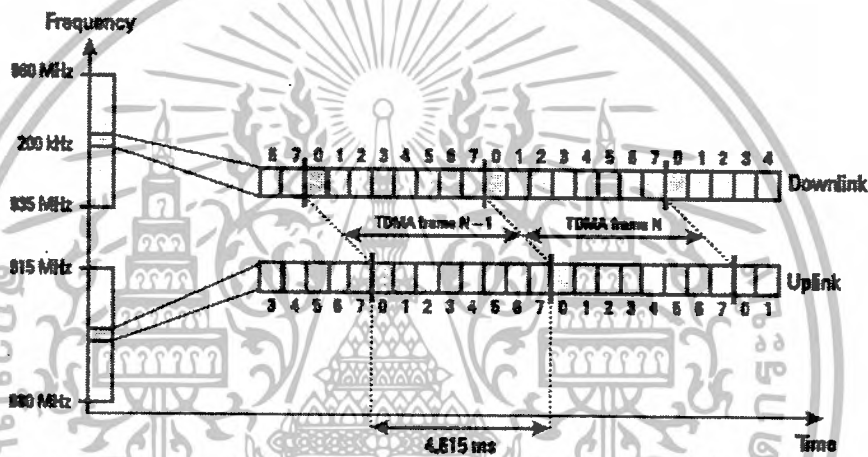
ย่อมาจาก Enhanced Data rates for GSM Evolution (EDGE) หรือเรียกอีกอย่างหนึ่งว่า Enhanced GPRS (EGPRS, Enhanced General Packet Radio Switch) เป็นเทคโนโลยีหนึ่งสำหรับโทรศัพท์เคลื่อนที่ เพื่อใช้ในการเพิ่มอัตราการรับส่งข้อมูล และเพิ่มความสำเร็จของการรับส่งข้อมูล ในทางเทคนิคแล้วเมื่อเปรียบเทียบ เอดจ์ กับเทคโนโลยีเครือข่าย 3G จะถูกจัดให้อยู่ในมาตรฐาน 2.75G อย่างไรก็ตามเป็นทางการ อันเนื่องมาจากความเร็วในเครือข่ายที่ช้ากว่านั้นและ ได้มีการนำเอดจ์ มาใช้ในเครือข่าย GSM ในปี 2546 โดยเริ่มครั้งแรกในประเทศสหรัฐอเมริกา



รูปที่ 2.3 เครือข่ายโทรศัพท์เคลื่อนที่ GSM

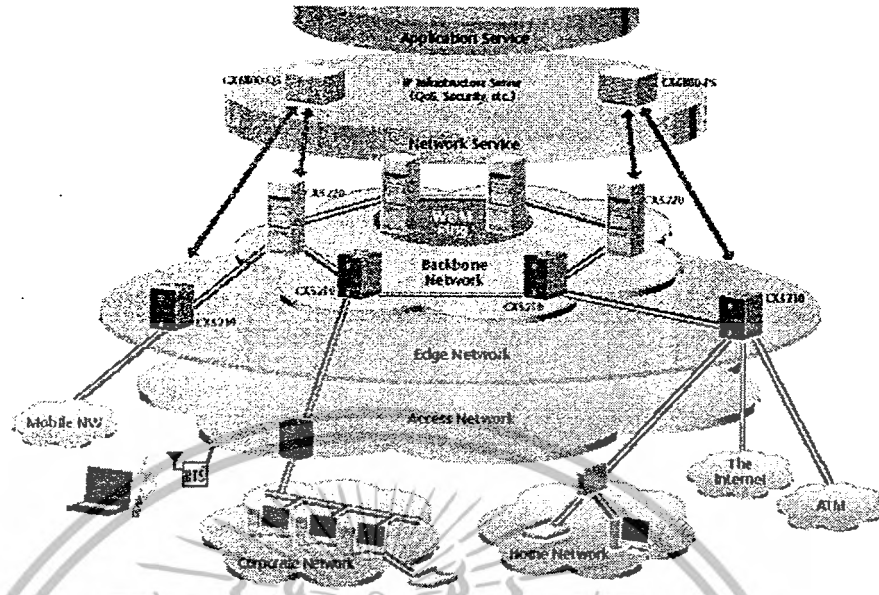
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอดจ์ ให้ความจุของข้อมูลที่สูงกว่า จีพีอาร์เอส ประมาณ 3 เท่า โดยที่ EDGE สามารถใช้โครงสร้างเฟรม TDMA ในช่องสัญญาณ 200 kHz ของระบบเครือข่าย GSM ทำให้ค่าใช้จ่ายในการใช้ EDGE ไม่สูงมาก นั่นคือ เอดจ์ เป็นการปรับปรุงซอฟต์แวร์ เพื่อให้ได้ความสามารถที่เพิ่มขึ้น (software-upgrade) นั่นเอง



รูปที่ 2.4 การรวมกันของ FDMA และ TDMA

เอดจ์ สามารถใช้ได้กับแอปพลิเคชันที่เป็น packet switch ใดๆก็ได้ อาทิเช่น การเชื่อมต่ออินเทอร์เน็ต หรือแอปพลิเคชันที่ต้องการการรับส่งข้อมูลความเร็วสูงเช่น การให้บริการวิดีโอ เพลง อีเมลล์หรือโปรแกรมมัลติมีเดียอื่นๆ ที่ต้องการปริมาณของข้อมูลที่เพิ่มขึ้นมาอันเนื่องจากเทคโนโลยีนี้ และวงจรไฟฟ้าของ เอดจ์ นี้ ก็สามารถพัฒนาเพิ่มเติมได้ในอนาคต

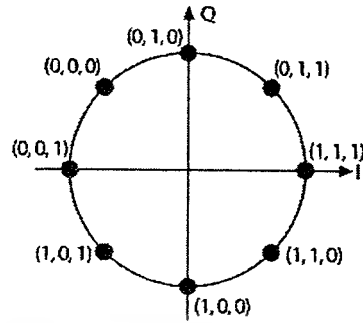


### รูปที่ 2.5 เทคโนโลยี เอจ ในเครือข่ายโทรศัพท์เคลื่อนที่

ในปัจจุบันนี้วิวัฒนาการของ เอจ ได้มาถึงในริลีสที่ 7 ตามมาตรฐานที่จัดทำโดย 3GPP (3rd Generation Partnership Project) เพื่อใช้ในการเพิ่มประสิทธิภาพ เช่น การทำให้เป็น High-Speed Packet Access (HSPA)

### การพัฒนาของ จีพีอาร์เอส ยูเอจ

แนวความคิดเบื้องหลัง เอจ ก็คือการเพิ่มอัตราการรับส่งข้อมูลด้วยสัญญาณ GSM ในช่วงความถี่พาหะ 200 KHz โดยการเปลี่ยนประเภทของการมอดดูเลชันที่มีอยู่ในโหนดของเครือข่าย GSM และ จีพีอาร์เอส วิธีการมอดดูเลชันแบบใหม่ที่น่าสนใจคือ 8-PSK (eight state phase shift keying) โดยมีหลักการพื้นฐานว่าจะต้องมีผลกระทบต่อแกนหลักของเครือข่ายน้อยที่สุด



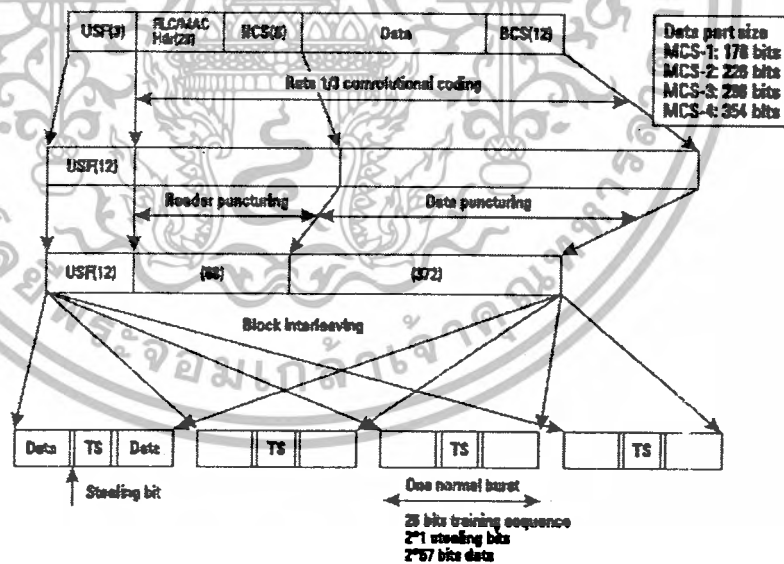
รูปที่ 2.6 แผนภาพแทนสัญลักษณ์ 8 PSK เทียบกับตำแหน่งบิตข้อ

เอ็ดจ์ ถูกพิจารณาว่าเป็นมาตรฐาน 2.75G นั่นคือผู้ให้บริการ โทรศัพท์เคลื่อนที่ไม่จำเป็นต้องขอใบอนุญาต เพราะว่าคุณสมบัติ เอ็ดจ์ นี้ยังคงใช้แถบความถี่เดิม และมีราคาค่าใช้จ่ายในการลงทุนต่ำสำหรับผู้ให้บริการที่ต้องการให้บริการมัลติมีเดียบนเครือข่าย GSM/จีพีอาร์เอส เดิมที่มีอยู่

เอ็ดจ์ จะอยู่ในส่วนบนของบริการ HSCSD (High-Speed Circuit-Switched Data) ซึ่งใช้ในการรับส่งข้อมูลในโหมดเซอร์กิตสวิทช์ของช่องสัญญาณต่างๆ โดยเริ่มจากวิวัฒนาการของ EDGE นี้เรียกว่า enhanced circuit-switched data (ESCD) ซึ่ง ESCD นี้สนับสนุนอัตราการรับส่งข้อมูลของระบบ GSM ที่มีอยู่ในปัจจุบัน (2.5 Kbps, 4.8 Kbps, 9.6 Kbps, 14.4 Kbps) และ CSs แบบใหม่ มาผสมผสานกับวิธีการมอดูเลชันแบบใหม่ ทำให้เกิดอัตรารับส่งข้อมูลที่ 28.8 Kbps 32 Kbps และ 43.2 Kbps ต่อช่องสัญญาณ ดังนั้นการใช้งาน ESCD ของ 4 ช่องสัญญาณรวมกันจะได้เป็น 172.8 Kbps ESCD ที่มีการปรับเปลี่ยนของโครงสร้างและวิวัฒนาการจาก จีพีอาร์เอส ไปยัง เอ็ดจ์ เรียกว่า EGPRS หรือบางครั้งเรียกว่า เอ็ดจ์ Classic ซึ่ง EGPRS มีการทำงานจะตั้งอยู่บนพื้นฐานสถาปัตยกรรมของเครือข่าย GPRS เดิม โดยสามารถรับส่งข้อมูลได้ถึง 475 Kbps สำหรับอุปกรณ์ที่สามารถรับส่งสัญญาณได้ 8 ช่องสัญญาณพร้อมกัน ซึ่งการมอดูเลชันสัญญาณวิทยุนี้จะมีผลกระทบ ในส่วนของ BSS (Base Station Subsystem) และ MS (Mobile Station) เท่านั้น

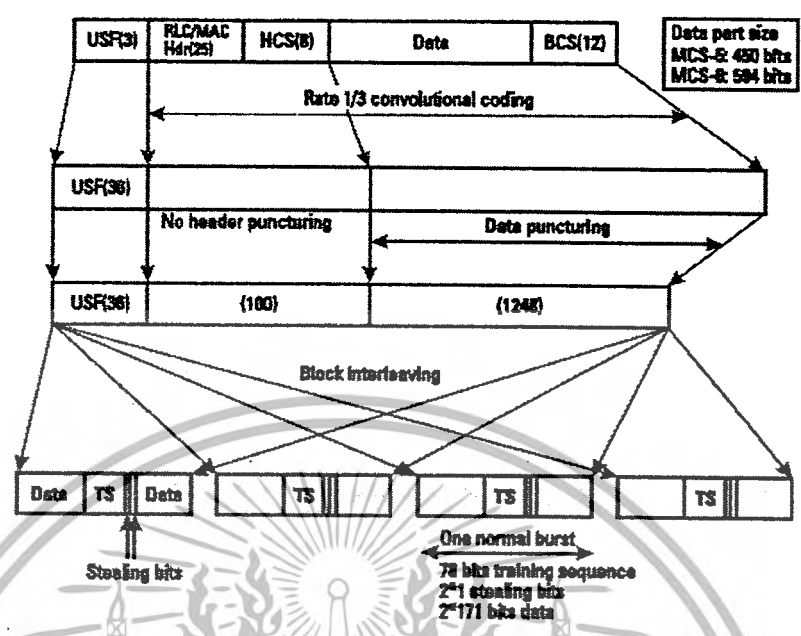
Modulation and Coding Scheme	Modulation	Maximum Throughput (Kbps)
MCS-9	8-PSK	59.2
MCS-8	8-PSK	54.4
MCS-7	8-PSK	44.8
MCS-6	8-PSK	29.6
MCS-5	8-PSK	22.4
MCS-4	GMSK	17.6
MCS-3	GMSK	14.8
MCS-2	GMSK	11.2

ตารางที่ 1 แสดงค่าอัตราการรับ-ส่งกับวิธีการ Modulation เทียบกับวิธีการเข้ารหัสข้อมูล

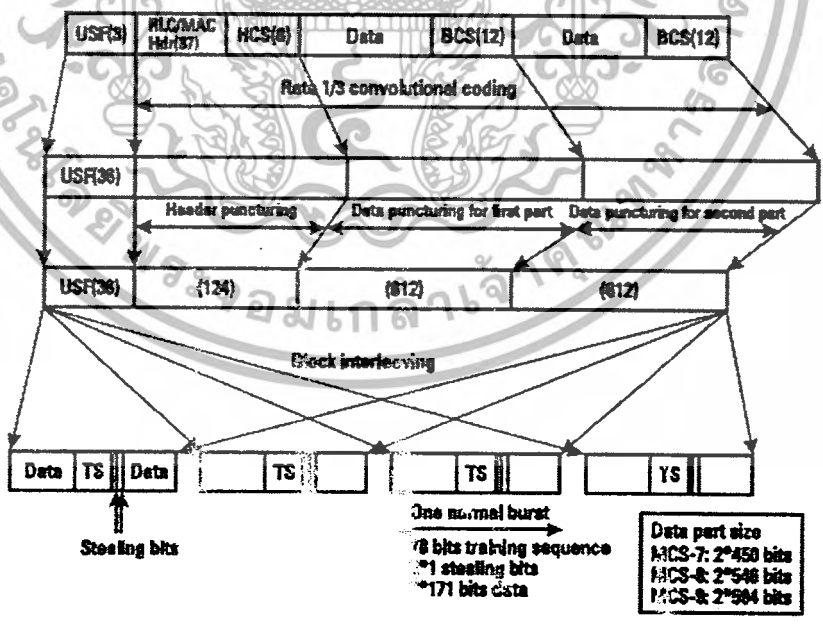


รูปที่ 2.7 การเข้ารหัส MCS-1 ถึง MCS-4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 การเข้ารหัส MCS-5 และ MCS-6



รูปที่ 2.9 การเข้ารหัส MCS-7, MCS-8 และ MCS-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## มาตรฐานของ GSM/เอดจ์

เอดจ์ นั้นถูกกำหนดให้เป็นวิธีหนึ่ง ในการขยายขีดความสามารถของการรับส่งข้อมูลต่อช่องสัญญาณทั้ง HSCSD (High-Speed Circuit-Switched Data) และ จีพีอาร์เอส โดยที่การขยายขีดความสามารถของ HSCSD จะถูกเรียกว่า ECSD (Enhanced Circuit-Switched Data) และการขยายขีดความสามารถของ จีพีอาร์เอส ก็จะเรียกว่า EGPRS (Enhanced General Packet Radio Service) โดยที่ใน ECSD อัตราการรับส่งข้อมูลสูงสุดไม่ได้เริ่มจาก 64 kbps เนื่องจากข้อจำกัดในการเชื่อมต่อแบบเอ (A-Interface) แต่ว่าอัตราการรับส่งข้อมูลต่อช่องสัญญาณจะเพิ่มขึ้น 3 เท่า เช่นเดียวกับใน EGPRS อัตราการรับส่งข้อมูลต่อช่องสัญญาณจะเพิ่มขึ้น 3 เท่าเช่นกัน โดยถ้ารวมทั้ง 8 ช่องสัญญาณในการเชื่อมต่อทางสัญญาณวิทยุก็จะมีค่าถึง 473 kbps

### 1. การมอดดูเลชัน 8-PSK ในมาตรฐาน GSM/เอดจ์

8-PSK (Octagonal Phase Shift Keying) คือวิธีที่ใช้ในการขยายขีดความสามารถของการรับส่งข้อมูลที่เพิ่มเติมขึ้นจากการรับส่งแบบเดิม GMSK (Gaussian Minimum Shift Keying) โดยที่สัญญาณของ 8-PSK มีความสามารถที่จะเก็บข้อมูลได้ขนาด 3 บิตต่อการมอดดูเลชันในสัญญาณวิทยุขณะที่ GMSK สามารถเก็บได้เพียง 1 บิตเท่านั้น เมื่ออัตราสัญญาณพาหะ 270.833 kbps ของมาตรฐาน GSM จะเก็บไว้ในรูปของ 8-PSK ซึ่งมีรูปของสัญญาณคลื่นเช่นเดียวกับที่ใช้ใน GMSK ซึ่งการเพิ่มอัตราการรับส่งข้อมูลนี้จะทำให้สูญเสียความไวในการรับสัญญาณ (sensitivity) ลงไปบ้าง ดังนั้นการรับส่งที่อัตราสูงสุดที่จะทำได้จะทำในบางพื้นที่ที่ครอบคลุมถึงเท่านั้น

แถบสเปกตรัมของ GMSK จะเป็นจุดเริ่มต้นของแถบสเปกตรัมของสัญญาณ 8-PSK แต่ด้วยการกำหนดมาตรฐานที่เริ่มต้นมาอย่างยาวนาน จึงมีการกำหนดให้แถบสเปกตรัมของ 8-PSK ผ่อนคลายสัญญาณลง 2-3 เดซิเบลและมีค่าออฟเซตห่างออกจากศูนย์กลางความถี่ 400 kHz ซึ่งถูกพบว่าเป็นการกระทำให้ระบบโดยรวมมีประสิทธิภาพดีที่สุด

## 2. การขยายขีดความสามารถของ จีพีอาร์เอส (EGPRS)

EGPRS นั้นถูกสร้างอยู่บน GPRS ที่ซึ่งก็คือการให้บริการข้อมูลแบบ แพ็คเก็ตสวิตซ์ในระบบ GSM นั้นเอง ตัวอย่างก็ได้แก่ การรับส่งข้อมูลความเร็วสูง รวมทั้งการ รับส่งไฟล์ การใช้บริการอินเทอร์เน็ตต่างๆ การใช้เว็บเบราว์เซอร์และการใช้อีเมล EGPRS มีผลกระทบต่อคลื่นความถี่และการเชื่อมต่อความถี่ในชั้น Physical layer พอๆกับผลต่อโปรโตคอล RLC/MAC (Radio Link Control / Medium Access Control) แต่ผลกระทบต่อโปรโตคอลอื่นๆนั้นเล็กน้อยมาก

## 3. การขยายขีดความสามารถของข้อมูลในเซอร์กิตสวิตซ์ (ECSD)

ECSD นั้นใช้ HSCSD เป็นพื้นฐาน อัตราการรับส่งข้อมูลของผู้ใช้จะไม่เพิ่มขึ้นเมื่อเทียบกับ HSCSD แต่อัตราการรับส่งนี้จะใช้ช่องสัญญาณที่น้อยลงและทำให้การจัดเตรียม MS ได้ง่ายกว่ามาก ถึงแม้ว่าอัตราการรับส่งข้อมูลที่จัดให้โดย ECSD จะถูกจำกัดที่ 64 kbps ซึ่งยังคงเพียงพอต่อการให้บริการต่างๆที่หลากหลาย ECSD สามารถใช้ในการทำงานแบบ inter-working ด้วย audio modems ซึ่งมีอัตราการรับส่งดีกว่าเครือข่าย GSM ในปัจจุบัน ตัวอย่างเช่นการให้บริการ Video Conference

### 1. วิธีการรับส่งข้อมูลแบบสองส่วนในคลาสเอ (Class A Dual Transfer Mode, DTM)

ด้วยสถานีโทรศัพท์เคลื่อนที่คลาสเอ (Class A mobile station) จะทำให้การรับส่งข้อมูลและสัญญาณเสียงสามารถรับส่งพร้อมกันได้

### 2. การบีบอัดข้อมูล เอคจี (EDGE Compact)

การบีบอัดข้อมูลเป็น โหมดเฉพาะสำหรับ EGPRS ซึ่งถูกออกแบบมาสำหรับการใช้งานในบางความถี่เท่านั้น ในความถี่ที่แคบความต้องการขั้นต่ำในการใช้งานจะต้องใช้สเปกตรัมน้อยกว่า 1 MHz ซึ่งเป็นที่รู้จักกันดีว่า ความจุของเครือข่าย GSM นั้นถูกจำกัดโดยการนำกลับมาใช้ใหม่ของ BCCH (Broadcast Control Channel)

## เอคจี กับ วิธีการรับส่งข้อมูลอื่นๆ

การรับส่งข้อมูลอื่นๆที่ใช้สำหรับ None Voice Application นั้นที่เราสามารถเห็นได้ในปัจจุบัน ได้แก่

1. Short Message Service (SMS) เป็นการรับส่งข้อความสั้นๆ ที่ได้รับความนิยมมากในทุกวันนี้

2. Circuit Switched Data (CSD) คือ WAP หรือ Wireless application Protocol ที่เชื่อมต่อกับ WAP Server ต่างๆในรูปของ Wireless Internet ซึ่งมีข้อจำกัดในด้านความเร็วในการรับส่งข้อมูลในช่วง 9.6 – 28.8 kbps

3. General Packet Radio Service (GPRS) มีความสามารถในการรับส่งข้อมูลได้สูงถึง 9-40 kbps ซึ่งสามารถรับส่งข้อมูลที่เป็น VDO Mail หรือภาพเคลื่อนไหวต่างๆได้ และเชื่อมต่อกับเครือข่ายอินเทอร์เน็ตได้เร็วขึ้น โดยมีลักษณะเป็น Always On คือการเชื่อมต่อกับเครือข่าย และการรับส่งข้อมูลสามารถดำเนินต่อไปถึงแม้ว่าจะมีการติดต่อเข้ามาก็ตาม ซึ่งการรับ-ส่งจะไม่ขาดตอนลง

4. Enhanced Data Rate for GSM Evolution (EDGE) เป็นการขยายขีดความสามารถของเครือข่ายโทรศัพท์เคลื่อนที่ GSM/GPRS ให้รองรับการรับ-ส่งข้อมูลในอัตราที่สูงขึ้น

เมื่อเปรียบเทียบมาตรฐานโทรศัพท์เคลื่อนที่ตั้งแต่ยุค 2.5G ขึ้นไปทุกระบบจะเปรียบเทียบกันได้ด้วยตารางที่ 2

เทคโนโลยี	อัตราเร็ว	เริ่มใช้งาน	จุดเด่น	จุดด้อย	หมายเหตุ
จีพีอาร์เอส	171.2 kbps	พ.ศ. 2544	มาตรฐานแรกของตระกูล GSM ที่มีการรับส่งข้อมูลแบบแพ็คเกจ	อัตราเร็วในการสื่อสารต่ำมาก	
HSCSD	115 kbps	พ.ศ. 2542	ใช้ช่องสื่อสารแบบสวิตช์วงจรหลายช่องในการรับส่งข้อมูล ส่งได้ด้วยอัตราเร็วสูง	มีต้นทุนของช่องสัญญาณสูงมากแทบไม่มีเปิดให้บริการ	เป็นเทคโนโลยีที่ยุติทางหลงไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

			พอสมควร		
EDGE Classic	384 kbps	พ.ศ. 2546	เพิ่มอัตราเร็วในการรับส่งข้อมูลบนเครือข่าย GPRS มากขึ้น	มีต้นทุนการปรับปรุงสถานีฐานค่อนข้างสูง เครื่องลูกข่ายมีน้อย	ประสิทธิภาพและความคุ้มทุนต่ำมากเมื่อเทียบกับโทรศัพท์เคลื่อนที่ W-CDMA
EDGE Compact	250 kbps	พ.ศ. 2545	ออกแบบพิเศษให้ใช้กับเครือข่าย TDMA ในประเทศสหรัฐอเมริกา	ผู้ให้บริการในสหรัฐอเมริกาเปลี่ยนทิศทางไปลงทุนในเครือข่าย 3G แทน	ไม่ประสบความสำเร็จ ปัจจุบันยุติโครงการลงแล้ว
CDMA/IS-95B	115 kbps	พ.ศ. 2538	เทคโนโลยีการสื่อสารข้อมูลแบบ फैคกัคที่ ใช้ในช่วงเปลี่ยนผ่านจากยุค CDMA สู่ CDMA 2000	มีเฉพาะใช้ประเทศญี่ปุ่นและเกาหลีใต้	
CDMA2000 1xMC	307 kbps	พ.ศ. 2545	อัตราเร็วในการสื่อสารข้อมูลสูง สามารถพัฒนาขึ้นจาก	มีผู้เปิดให้บริการทั่วโลก ้อยมาก	ไม่ประสบความสำเร็จ ปัจจุบันยุติโครงการลงแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			เครือข่าย CDMA		
PDC-P	9.6 kbps		ใช้โดยบริษัท NTT DoCoMo ประเทศญี่ปุ่น ภายใต้ เครื่องหมาย การค้า i-mode	มีใช้เฉพาะใน ประเทศญี่ปุ่น เท่านั้น อัตราเร็วใน การรับส่งต่ำ	เป็นเทคโนโลยี การสื่อสารข้อมูล แบบสวิตซ์ แพ็คเกจที่มีฐาน ผู้ใช้งานสูงสุดใน โลก
W-CDMA	2 Mbps	พ.ศ. 2544	เป็นมาตรฐานที่ ผลิตขึ้นเพื่อการ ใช้งานในเชิง พาณิชย์ในการ เข้าสู่ยุค 3G ของ มาตรฐาน GSM	มีต้นทุนด้าน ใบอนุญาต ประกอบการ และการ ประมูล ความถี่สูง มาก	เป็นมาตรฐาน ระดับโลกสำหรับ เทคโนโลยี 3G
CDMA2000 3xMC	2 Mbps	พ.ศ. 2547	รองรับ มาตรฐาน CDMA 2000- 1xMC และ IS- 95A	ไม่มีการ พัฒนาต่อ	ไม่ประสบ ความสำเร็จ ปัจจุบันยุติ โครงการลงแล้ว
CDMA EV-DV	2.4 Mbps	พ.ศ. 2546	เป็นเทคโนโลยี ที่ใช้เข้าสู่ยุค 3G สำหรับ มาตรฐาน CDMA โดยมี ต้นทุนต่ำที่สุด	มีผู้เปิด ให้บริการทั่ว โลกน้อยมาก	ยังคงอยู่ แต่คาด ว่าจะไม่เป็น มาตรฐานสำคัญ ในยุค 3G

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CDMA XTREAME	5.2 Mbps	พ.ศ. 2547	อัตราเร็วในการ สื่อสารสูงมาก	ยังเป็น มาตรฐาน เฉพาะของ บริษัท Nokia และ Motorola	ยังไม่มีทิศ ทางการพัฒนา ชัดเจน
-----------------	----------	--------------	---------------------------------	---	--------------------------------------

ตารางที่ 2 เปรียบเทียบมาตรฐาน โทรศัพท์เคลื่อนที่ตั้งแต่ยุค 2.5G ขึ้น ไปทุกระบบ

#### 2.4 Macromedia Dreamweaver 8

Macromedia Dreamweaver 8 เป็นอีกโปรแกรมหนึ่ง ที่มีการออกแบบมาเพื่อที่จะใช้ในการจัดการกับเอกสารที่ใช้ สำหรับการสร้างเว็บเพจ ซึ่งในสมัยก่อนหากจะมี การสร้างเว็บเพจ ขึ้นแต่ละเว็บเพจนั้น ต้องให้ผู้ที่มีความรู้ในภาษา HTML มาเขียนรหัสคำสั่ง (Code) ให้ แต่ในปัจจุบันโปรแกรม Macromedia Dreamweaver 8 สามารถที่จะสร้างรหัสคำสั่งให้กับผู้ใช้โดยอัตโนมัติ ซึ่งผู้ใช้ไม่จำเป็นต้องมีความรู้ด้านของภาษา HTML เนื่องจากโปรแกรม Macromedia Dreamweaver 8 นั้นจะมีลักษณะ การทำงานที่คล้ายๆ กับโปรแกรมพิมพ์เอกสารที่เราเคยใช้และรู้จักกันดี ซึ่งจะมีเครื่องมือและแถบคำสั่งให้เราเลือกใช้ได้ เหมือนกับ Word Processor จึงช่วยให้สามารถเว็บเพจด้วยความสะดวก และรวดเร็ว

##### 2.4.1 จุดเด่นของโปรแกรม Macromedia Dreamweaver 8

1. โปรแกรมจะทำการแปลงรหัสให้เป็นภาษา HTML โดยอัตโนมัติดังนั้น ผู้ใช้ที่ไม่มีความรู้ด้านนี้ก็สามารถทำได้
2. มีแถบเครื่องมือหรือแถบคำสั่ง ที่ใช้ในการควบคุมการทำงาน แบ่งออกเป็นหมวดหมู่จึงช่วยในการทำงานที่ดีขึ้น และรวดเร็วยิ่งขึ้น
3. สนับสนุนเว็บเพจที่เป็นภาษาไทยได้ดี
4. มีคุณสมบัติที่สามารถจัดการกับรูปภาพเคลื่อนไหว โดยไม่ต้องอาศัย Plugin
5. สามารถเรียกใช้ตารางจากภายนอก โดยการอิมพอร์ตจาก Text File
6. เป็นโปรแกรมที่สามารถสนับสนุนการใช้งาน CSS (Cascading Style Sheet)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

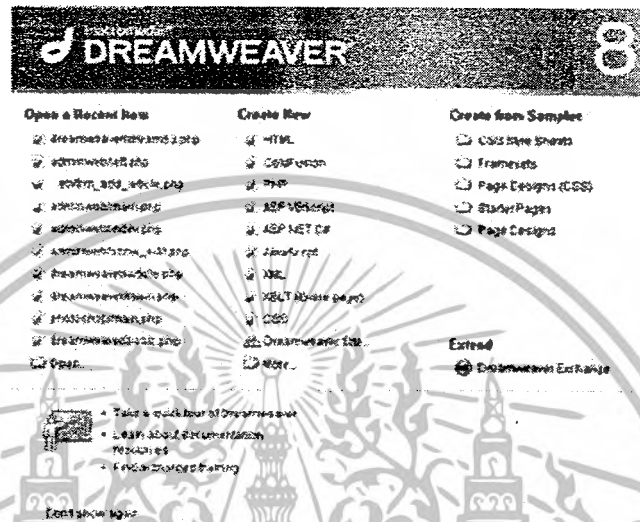
7. มีความสามารถในการทำ Drop Down Menu รวมไปถึงการทำให้รูปภาพเปลี่ยนเมื่อนำเมาส์ไปชี้ เป็นต้น

#### 2.4.2 ความสามารถของ Macromedia Dreamweaver 8

โปรแกรม Macromedia Dreamweaver 8 เป็นโปรแกรมที่ใช้งานได้ง่ายคล้ายกับโปรแกรมประเภทเดียวกับ Front Page ซึ่งเพียงแค่เราใช้งานโปรแกรมคอมพิวเตอร์ทั่วไปและแต่ละภาคองค์ประกอบ ของหน้าเว็บเพจที่ต้องการ เรียกว่า Object ไปวางบนหน้าเอกสาร ซึ่งความสามารถของ Macromedia Dreamweaver 8 มีดังต่อไปนี้

1. สนับสนุนการทำงานแบบ WYSIWYG (What You See Is What You Get) อะไรก็ตามที่ทำบนหน้าจอ Dreamweaver ก็จะปรากฏผลแบบเดียวกันบนเว็บเพจ ซึ่งช่วยให้การปรับปรุงแก้ไขเว็บเพจนั้นทำได้ง่าย ไม่ต้องมีความรู้ในภาษา HTML
2. มีเครื่องมือในการช่วยสร้างรูปแบบหน้าจอเว็บเพจและมีความยืดหยุ่นในการใช้งานสูง
3. สนับสนุนภาษาสคริปต์ต่างๆ เช่น Java, ASP, PHP, CGI, VBScript
4. มีเครื่องมือในการอัปโหลด (Upload) หน้าเว็บเพจไปที่เครื่อง Server เพื่อทำการเผยแพร่ งานที่เราสร้างในอินเทอร์เน็ต โดยการส่งผ่าน FTP หรือโดยการใช้โปรแกรมภายนอกช่วย
5. รองรับมัลติมีเดีย เช่น เสียง กราฟิก และแอนิเมชัน ที่สร้างโดยโปรแกรม Macromedia Flash
6. สร้างเว็บเพจได้อย่างง่ายดายแบบ WYSIWYG (What-See-Is-What-You-Get) โดยไม่จำเป็นต้องรู้โค้ดภาษา HTML หรือ JavaScript, VBScript มาก่อนเลยก็ได้
7. ออกแบบเว็บเพจด้วยเลเยอร์ (Layer) ก่อนแล้วจึงแปลงให้เป็นตารางได้อย่างง่ายดาย
8. ปรับเปลี่ยนตารางให้เป็นเลเยอร์เพื่อสะดวกในการจัดลำดับของการซ้อนภาพหรือข้อความได้
9. สร้างปุ่มกดแบบ Flash ได้ง่ายๆ สามารถนำไปใช้ได้สะดวก

Dreamweaver เป็นโปรแกรมประเภท Web Design ซึ่งมีคุณสมบัติในการใช้งานในแบบ WYSIWYG (What You See Is What You Get)



รูปที่ 2.10 โปรแกรม Dreamweaver 8

ความสามารถเพิ่มเติมในด้านอื่นๆ

- Zoom tool and guides
- Visual XML data binding
- New CSS Styles panel
- CSS layout visualization
- Code collapse
- Coding toolbar
- Background file transfer
- Insert Flash Video command

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 Appserv

สำหรับโปรแกรม AppServ นี้ไม่ได้เกิดการสนับสนุนจากหน่วยงานรัฐบาลหรือหน่วยงานเอกชน หรือองค์กรอิสระใดๆเลยทั้งสิ้น แต่โปรแกรม AppServ ได้กำเนิดจากแรงบันดาลใจจากผู้พัฒนาที่ได้เริ่มศึกษาภาษา PHP และฐานข้อมูล MySQL และมีปัญหาทุกครั้งที่การติดตั้ง กว่าจะติดตั้งได้ก็ใช้เวลาไม่น้อยกว่า 3 ชั่วโมง บางทีทำได้บ้างไม่ได้บ้าง และทุกครั้งที่ติดตั้งไม่ได้ก็จะมาขอความช่วยเหลือ จากผู้พัฒนาเป็นประจำทุกครั้ง จึงทำให้ผู้พัฒนาได้สร้างโปรแกรมที่สะดวกในการติดตั้งเพื่อให้เพื่อนของผู้พัฒนาสามารถนำไปใช้งานได้ทันที โดยไม่ต้องมาวุ่นวายกับการติดตั้งที่ยุ่งยากอีกต่อไป ในช่วงแรกที่แจกจ่ายนั้นผู้พัฒนาได้แจกจ่ายในเว็บไซด์ที่เป็นภาษาอังกฤษ ผู้ใช้งานต่างประเทศให้ความสนใจและมีการใช้งานเป็นจำนวนมาก และในปัจจุบันได้เพิ่มเติมในส่วนของเว็บไซด์ภาษาไทย ในอนาคตผู้พัฒนาจะจัดทำเว็บไซด์สามารถรองรับทุกภาษาและเข้าถึงผู้ใช้งานทุกคนทั่วโลก

### 2.5.1 รู้จักกับ Appserv

AppServ คือโปรแกรมที่รวบรวมเอา Open Source Software หลากๆ อย่างมารวมกัน โดยมี Package หลักดังนี้

- Apache
- PHP
- MySQL
- phpMyAdmin

โปรแกรมต่างๆ ที่นำมารวบรวมไว้ทั้งหมดนี้ ได้ทำการดาวน์โหลดจาก Official Release ทั้งหมด โดยตัว AppServ จึงให้ความสำคัญว่าทุกสิ่งทุกอย่างจะต้องให้เหมือนกับต้นฉบับ จึงไม่ได้คัดลอกหรือเพิ่มเติมอะไรที่แปลกไปกว่า Official Release แต่อย่างใด เพียงแต่มีบางส่วนเท่านั้นที่ได้เพิ่มประสิทธิภาพการติดตั้งให้สอดคล้องกับการทำงานแต่ละคน โดยที่การเพิ่มประสิทธิภาพนี้ไม่ได้ไปยุ่งในส่วนของ Original Package แม้แต่น้อยเพียงแค่เป็นการกำหนดค่า Config เท่านั้น เช่น Apache ก็จะเป็นในส่วนของ httpd.conf, PHP ก็จะเป็นในส่วนของ php.ini, MySQL ก็จะเป็นในส่วนของ my.ini ดังนั้นเราจึงรับประกันได้ว่าโปรแกรม AppServ สามารถทำงานและความเสถียรของระบบ ได้เหมือนกับ Official Release ทั้งหมด

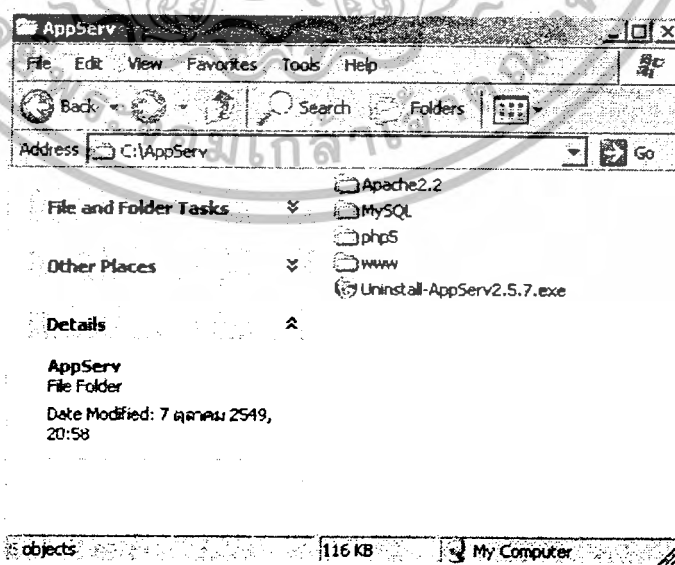
จุดประสงค์หลักของการรวบรวม Open Source Software เหล่านี้เพื่อทำให้การติดตั้งโปรแกรมต่างๆ ที่ได้กล่าวมาได้ง่ายขึ้น เพื่อลดขั้นตอนการติดตั้งที่แสนจะยุ่งยากและใช้เวลานาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยผู้ใช้งานเพียงดับเบิลคลิก setup ภายในเวลา 1 นาที ทุกอย่างก็ติดตั้งเสร็จสมบูรณ์ระบบต่างๆ ก็พร้อมทำงานได้ทันทีทั้ง Web Server, Database Server เหตุผลนี้จึงเป็นเหตุผลหลักที่หลายๆ คนทั่วโลก ได้เลือกใช้โปรแกรม AppServ แทนการที่จะต้องมาติดตั้งโปรแกรมต่างๆ ที่ละส่วนไม่ว่าจะเป็นผู้ที่ความชำนาญในการติดตั้ง Apache, PHP, MySQL ก็ไม่ได้เป็นเรื่องง่ายเสมอไป เนื่องจากการติดตั้งโปรแกรมที่แยกส่วนเหล่านี้ให้มารวมเป็นชิ้นอันเดียวกัน ก็ใช้เวลาค่อนข้างมากพอสมควร แม้แต่ตัวผู้พัฒนา AppServ เอง ก่อนที่จะ Release แต่ละเวอร์ชันให้ดาวน์โหลด ต้องใช้ระยะเวลาในการติดตั้ง ไม่น้อยกว่า 2 ชั่วโมง เพื่อทดสอบความถูกต้องของระบบ ดังนั้นจึงจะเห็นว่าเราเองนั้นเป็นมือใหม่หรือมือเก่า ไม่ใช่เรื่องง่ายเลยที่จะติดตั้ง Apache, PHP, MySQL ได้อย่างรวดเร็ว

นอกจากนี้ AppServ ยังสามารถนำไปเป็น Web Server หรือ Database Server ได้ทันทีแต่ทางผู้พัฒนาเองขอแนะนำว่า ระบบจัดการ Memory และ CPU บน Windows ที่ทำงานเกี่ยวกับ Web Server หรือ Database Server ไม่เหมาะกับการใช้งานหนักๆ เป็นอย่างยิ่ง เพราะ Windows นั้นใช้ทรัพยากรมาก และหากเทียบอัตราการรับระบบงานกับ OS ตัวอื่นเช่น Linux/Unix จะยิ่งเห็นได้ชัดว่า OS ที่เป็น Windows ที่มีขนาด Memory และ CPU ที่เท่าๆ กัน ส่วน OS ที่เป็น Linux/Unix นั้น จะรองรับงานได้น้อยกว่ามากพอสมควร เช่น Windows รับได้ 1000 คนพร้อมๆ กัน แต่ Linux/Unix อาจรับได้ถึง 5000 คนพร้อมๆ กัน หากต้องรองรับผู้ใช้บริการ ทางผู้พัฒนาแนะนำให้เลือกใช้ Linux/Unix OS จึงจะเหมาะสมกว่า

## 2.5.2 วิธีการใช้งาน Appserv และระบบจัดเก็บไฟล์



รูปที่ 2.11 โครงสร้างระบบจัดเก็บไฟล์หลังการติดตั้ง Appserv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### โครงสร้างระบบการจัดเก็บไฟล์ของ Apache Web Server

- apache/bin จัดเก็บ โปรแกรมการทำงานหลักของ Apache
- apache/conf จัดเก็บค่าคอนฟิกระบบของ Apache
- apache/error จัดเก็บการแจ้ง Error Template (มีเฉพาะ Apache 2)
- apache/icons จัดเก็บ ไอคอนต่างๆ
- apache/logs จัดเก็บ Log การทำงานของ Apache
- apache/modules จัดเก็บ Module เสริมของ Apache

### โครงสร้างระบบการจัดเก็บไฟล์ของ MySQL Database

- mysql/bin จัดเก็บ โปรแกรมการทำงานหลักของ MySQL
- mysql/data จัดเก็บฐานข้อมูลต่างๆของ MySQL โดยแต่ละ Table จะแยกเป็น Directory ย่อย
- mysql/share จัดเก็บที่เก็บ Error Message แยกตาม Charset

### โครงสร้างระบบการจัดเก็บไฟล์ของ PHP

- php จัดเก็บ php Command line Execution และ DLL Library
- php/ext จัดเก็บ PECL PHP Extension ของ PHP (มีเฉพาะ PHP 5)
- php/extension จัดเก็บ PECL PHP Extension ของ PHP (มีเฉพาะ PHP 4)
- php/PEAR จัดเก็บ PEAR Framework Components ของภาษา PHP

### โครงสร้างระบบการจัดเก็บไฟล์เว็บไซต์ไคเรกทอรี www

- www จัดเก็บ Directory Root สำหรับเก็บไฟล์ต่างๆ ของเว็บไซต์
- www/cgi-bin จัดเก็บ ไฟล์โปรแกรมประเภท CGI
- www/phpMyAdmin จัดเก็บ โปรแกรม phpMyAdmin
- www/appserv จัดเก็บไฟล์ของ appserv สามารถลบทิ้งได้ทันทีหลังจากติดตั้ง
- www/index.php ไฟล์หน้าแรกของเว็บไซต์ โดยเมื่อติดตั้งโปรแกรม AppServ สามารถลบ ไฟล์ index.php นี้ทิ้งได้ทันที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ข้อแตกต่างของ AppServ ในแต่ละเวอร์ชัน

AppServ ได้แบ่งเวอร์ชันออกเป็น 2 ส่วนด้วยกัน คือ 2.5.x คือเวอร์ชันที่นำ Package ใหม่ ๆ นำมาใช้งานโดยเฉพาะ เหมาะสำหรับนักพัฒนาที่ต้องการระบบใหม่ ๆ หรือต้องการทดสอบ ทดลอง ใช้งานฟังก์ชันใหม่ ซึ่งอาจจะไม่ได้ความเสถียรของระบบได้ 100% เนื่องจากว่า Package ของนักพัฒนายังอยู่ในช่วงของขั้นทดสอบ เพื่อหาข้อผิดพลาดอยู่ 2.4.x คือเวอร์ชันที่นำ Package ที่มีความเสถียรเป็นหลัก เหมาะสำหรับผู้ที่ต้องการความมั่นคงของระบบ โดยไม่ได้มุ่งเน้นที่จะใช้ฟังก์ชันใหม่

### คำแนะนำในการเลือกใช้งาน AppServ

AppServ ไม่จำเป็นต้องใช้เวอร์ชันใหม่เสมอไป แต่ถึงอย่างไรก็ดี ทางผู้พัฒนาเองอยากจะทำให้ผู้ใช้งานได้ทดสอบ ทดลองเวอร์ชันใหม่ทุกครั้งที่มีการแจกจ่าย เพื่อช่วยในการตรวจสอบปัญหา หรือข้อผิดพลาด ซึ่งจะเป็นประโยชน์ต่อตัวผู้พัฒนาเอง และผู้ใช้งานอื่นๆ ที่จะได้ใช้งานระบบมีข้อผิดพลาดที่น้อยที่สุด สำหรับผู้ที่ใช้งานระบบ ที่ไม่มีอะไรเปลี่ยนแปลงมากหรือไม่ได้ต้องการสิ่งใหม่ๆ แต่ต้องการความเสถียรเป็นหลัก ในการเลือกใช้ AppServ ให้เลือกใช้เวอร์ชันที่เหมาะสมกับระบบงานของท่าน เช่น หากใช้เวอร์ชัน 1.8.0 ได้อย่างไม่มีปัญหา ไม่จำเป็นต้องอัปเดตเวอร์ชันให้ใหม่อยู่เสมอ เพื่อที่จะให้การทำงานของระบบเป็นไปได้อย่างราบรื่น

### 2.6 ความรู้เบื้องต้นเกี่ยวกับ Visual Studio 2008

โปรแกรม Visual Studio 2008 เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่นิยมใช้อยู่ในปัจจุบัน โปรแกรม Visual Basic เป็นโปรแกรมที่ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงานมีเครื่องมือต่างๆ ที่เรียกกันว่า คอนโทรล (Controls) ไว้สำหรับช่วยในการออกแบบโปรแกรม โดยเน้นการออกแบบหน้าจอแบบกราฟฟิก หรือที่เรียกว่า Graphic User Interface (GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event-Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์ (Event) เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้คลิกปุ่มบนคีย์บอร์ด ผู้ใช้คลิกปุ่มเมาส์ เป็นต้น เครื่องมือหรือคอนโทรลต่าง ๆ ที่ Visual Basic ได้เตรียมไว้ให้ ไม่ว่าจะเป็น Form TextBox Label ฯลฯ ถือว่าเป็นวัตถุ (Object ในที่นี้ขอใช้คำว่า ออบเจกต์) นั้นหมายความว่า ไม่ว่าจะเป็นเครื่องมือใด ๆ ใน Visual Studio 2008 จะเป็นออบเจกต์ทั้งสิ้น สามารถที่จะควบคุมการทำงาน แกะไขคุณสมบัติของออบเจกต์นั้นได้โดยตรง ในทุกๆ ออบเจกต์จะมีคุณสมบัติ (properties)

และเมธอด (Methods) ประจำตัว ซึ่งในแต่ละออบเจกต์ อาจจะมีคุณสมบัติและเมธอดที่เหมือน หรือ ต่างกันก็ได้ ขึ้นอยู่กับชนิดของออบเจกต์

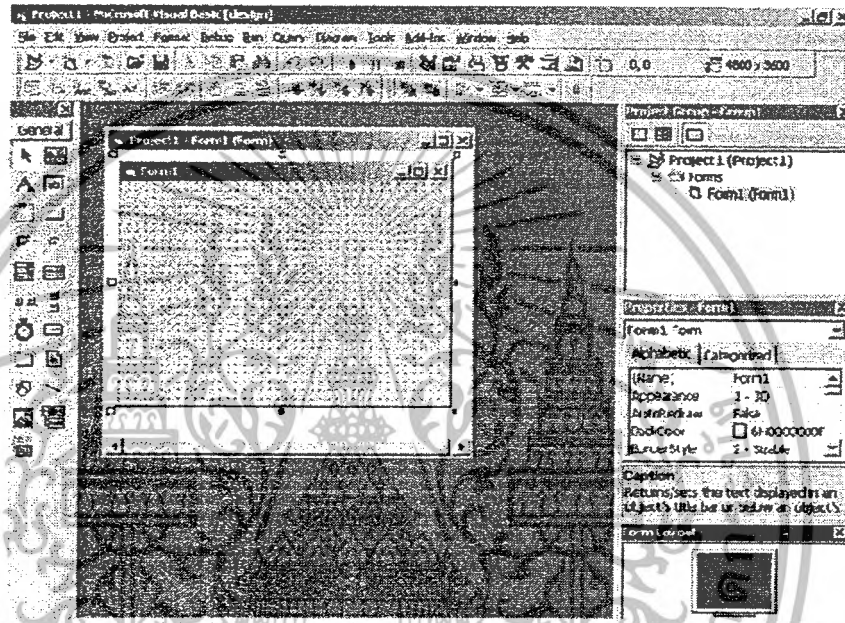
ในการพัฒนาโปรแกรมประยุกต์ด้วย Visual Studio 2008 การเขียนโค้ดจะถูกแบ่งออกเป็น ส่วนๆ เรียกว่า โพรซีเจอร์ (procedure) แต่ละโพรซีเจอร์จะประกอบไปด้วย ชุดคำสั่งที่พิมพ์เข้าไป แล้ว ทำให้คอนโทรลหรือออบเจกต์นั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียน โปรแกรมเชิงวัตถุ (Object Oriented Programming-OOP) แต่ตัวภาษา Visual Basic ยังไม่ถือว่าเป็น การเขียนโปรแกรมแบบ OOP อย่างแท้จริง เนื่องจากข้อจำกัดหลายๆ อย่างที่ Visual Studio 2008 ไม่ สามารถทำได้

วิซวลเบสิกดอทเน็ต (VB.NET) คือ เครื่องมือสำหรับพัฒนาโปรแกรมเป็นภาษาหนึ่งใน กลุ่มไมโครซอฟท์วิซวลสตูดิโอดอทเน็ต (Microsoft Visual Studio .NET) เป็นโปรแกรมที่มี สภาพแวดล้อมแบบกราฟิกสำหรับระบบปฏิบัติการวินโดวส์ (Windows Operating System) โดยมี รากฐานภาษามาจากภาษาเบสิก และทำงานบนดอทเน็ตเฟรมเวิร์ค (Dotnet Framework) ถูก ออกแบบให้มีความสามารถในการพัฒนาโปรแกรมเชิงวัตถุอย่างแท้จริง และรองรับการออกแบบ ด้วยยูเอ็มแอล (UML = Unified Modeling Language)

วิซวลสตูดิโอดอทเน็ต (Visual Studio .NET) คือ เครื่องมือสำหรับพัฒนาโปรแกรมแบบ ครบวงจร เป็นการรวมเครื่องมือที่จำเป็นต่อการพัฒนาโปรแกรมอย่างครบถ้วน (IDE = Integrated Development Environment) ซึ่งรวมบริการการพัฒนาภาษาโปรแกรม บริการคลาสิฟิเคชันให้ นำมาใช้งานร่วมกันได้อย่างเป็นระบบ เช่น เอสคิวแอลเซิร์ฟเวอร์ (SQL Server) วิซวลเบสิกดอท เน็ต (VB.NET) วิซวลซีชาร์ป (VC#) วิซวลเจชาร์ป (VJ#) วิซวลซีพลัสพลัส (VC++) และเอเอส พีดอทเน็ต (ASP .NET) เป็นต้น โดยทั้งหมดทำงานอยู่บนซีแอลอาร์ (CLR = Common Language Runtime) ที่รองรับการประมวลผลและเข้าใช้ทรัพยากรในเครื่องได้อย่างมีประสิทธิภาพ

## 2.6.1 เข้าสู่โปรแกรม Visual Studio 2008

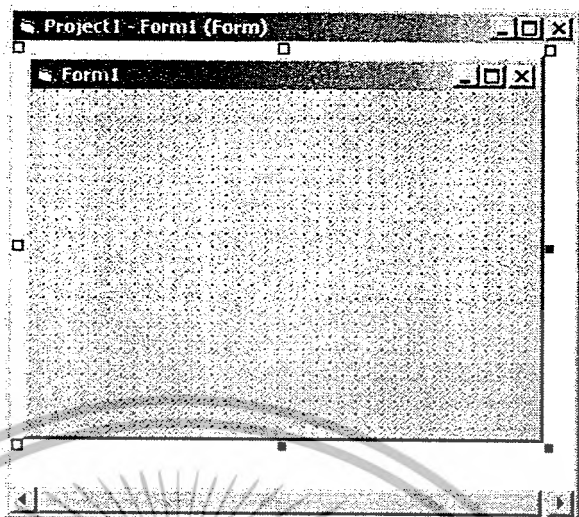
เมื่อเข้าสู่โปรแกรม Visual Studio 2008 จะแสดงกรอบโต้ตอบสำหรับเลือกชนิดของโปรแกรมประยุกต์ ที่ต้องการ เมื่อเลือกชนิดของโปรแกรมประยุกต์เป็นแบบ Standard EXE เข้าสู่หน้าต่างของ Visual Studio 2008 ในแต่ละส่วนของ Visual Studio 2008 มีหน้าที่แตกต่างกันไป ซึ่งในระหว่างการพัฒนาโปรแกรมประยุกต์ ต้องใช้ส่วนต่าง ๆ เหล่านี้ ในการพัฒนาโปรแกรมประยุกต์



รูปที่ 2.12 หน้าต่างของ Visual Studio 2008

### 1. Form Designer

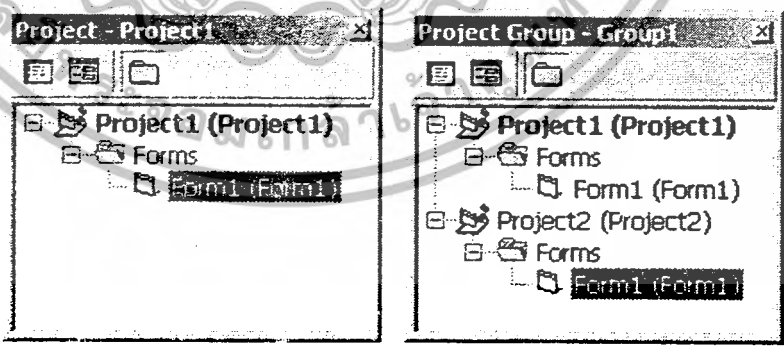
เป็นส่วนที่ใช้ออกแบบการแสดงผลส่วนที่ใช้ติดต่อกับผู้ใช้ ฟอรัมเป็นออบเจ็กต์แรกที่ถูกเตรียมไว้ให้ใช้งาน คอลโทรลทุกตัวที่ต้องการใช้งาน ต้องนำไปบรรจุไว้ในฟอรัมแล้วนำคอลโทรลมาประกอบกันขึ้นเป็นโปรแกรมประยุกต์ ทุกครั้งที่เปิด Visual Studio 2008 ขึ้นมา หรือ สร้าง Project ใหม่จะมีฟอรัมว่าง 1 ฟอรัมถูกสร้างเตรียมไว้เสมอ



รูปที่ 2.12 ก. ส่วนที่ใช้ออกแบบการแสดงผลเพื่อติดต่อใช้งานกับผู้ใช้

### 2. Project Explorer

Project Explorer ใช้สำหรับบริหารและจัดการ โปรเจกต์ โดยจะแสดงองค์ประกอบของแต่ละโปรเจกต์แบบโครงสร้างต้นไม้ (tree-view) ตัวโปรเจกต์จะหมายถึงโปรแกรมประยุกต์ซึ่งจะอยู่ส่วนบนสุด ถัดมาแสดงส่วนประกอบต่าง ๆ ของโปรเจกต์นั้นๆ ประกอบด้วยอะไรบ้าง เช่น ฟอร์มโมดูล รายงาน เป็นต้น ถ้ามี 2 โปรเจกต์ขึ้นไปจะแสดงแยกออกเป็นหน้าต่างหากอีกโปรเจกต์ ถ้าต้องการใช้งานส่วนใด ของโปรเจกต์ไหนก็สามารถคลิกเลือกได้ทันที



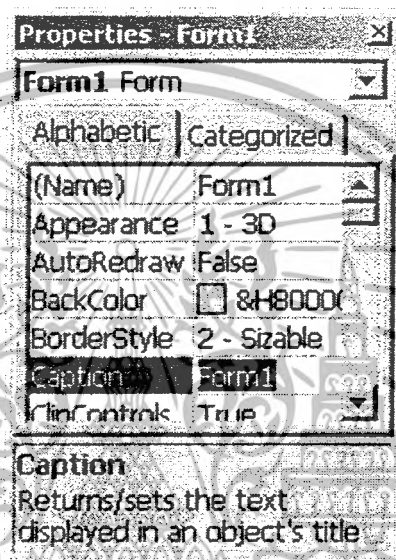
รูปที่ 2.12 ข. Project Explorer: แบบ โปรเจกต์เดี่ยวและแบบหลายโปรเจกต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3. ส่วนประกอบของโปรเจก

#### Properties Window

หน้าต่างคุณสมบัติเป็นส่วนที่ใช้กำหนดคุณสมบัติ ของออบเจกต์ที่ถูกเลือก (active) หรือได้รับความสนใจ (focus) อยู่ขณะนั้น ซึ่งสามารถที่จะปรับเปลี่ยนค่าต่าง ๆ ของคอลโทรลเพื่อให้เกิดความเหมาะสมและตรงกับความต้องการใช้งานได้ทันที



รูปที่ 2.12 ค. Properties Window

Project(n)	คือ โปรแกรมประยุกต์ที่พัฒนาอยู่ มีนามสกุล .vbp
Form(n).frm	เป็นฟอร์มที่มีอยู่ในโปรเจกนั้นๆ ใน 1 โปรเจกอาจมีมากกว่า 1 ฟอร์มก็ได้ มีนามสกุล .frm
Modules	เป็นที่เก็บชุดคำสั่งที่เขียนขึ้นมาโดยจะเก็บชุดคำสั่งที่ใช้บ่อยๆ มีนามสกุล .bas
Class Modules	เป็นโมดูลชนิดพิเศษที่มีลักษณะเป็นอ็อบเจกต์ ที่สามารถสร้างขึ้นมาได้ จะมีนามสกุล .cls
User control	เป็นส่วนที่เก็บคอนโทรล ActiveX ที่คุณสร้างขึ้นมา มีนามสกุล .ctl
Designers	เป็นส่วนของรายงานที่ถูกสร้างขึ้นมีนามสกุลเป็น .dsr

#### ตารางที่ 3 ส่วนประกอบของโปรเจก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในหน้าต่างคุณสมบัติ จะประกอบไปด้วยแท็บ 2 แท็บ คือ

1. แท็บ Alphabetic เป็นแท็บที่แสดงรายการคุณสมบัติ เรียงตามตัวอักษรในภาษาอังกฤษ
2. แท็บ Categorized เป็นแท็บที่แสดงรายการคุณสมบัติ โดยการจัดกลุ่มของคุณสมบัติที่มีหน้าที่คล้ายกัน หรือมีความสัมพันธ์กัน

## 2.6.2 ความสามารถของ Visual Basic 2008

เราพอที่จะสรุปความสามารถของ Visual Studio 2008 ได้ตามประเภทดังนี้

### ความสามารถพื้นฐาน

- ความสามารถในการติดต่อกับระบบ Hardwear
- สำหรับการเขียน โปรแกรมเพื่อควบคุมอุปกรณ์ภายนอกผ่านทาง Parallel Port ด้วย

Visual Studio 2008 Basic นั้น VB สามารถทำได้ดี แต่การเขียน โปรแกรมควบคุมอุปกรณ์ภายนอกนั้น Visual Studio 2008 ไม่มีฟังก์ชันสำหรับติดต่อกับ Parallel Port ได้โดยตรง แต่สามารถเขียนโปรแกรมให้ติดต่อกับ Parallel Port ได้โดยใช้ฟังก์ชันที่เตรียมมาในรูปของฟังก์ชัน API โดยเราสามารถอ่านข้อมูลและส่งข้อมูลไปยังพอร์ตที่ต้องการได้

### ความสามารถในการติดต่อกับฐานข้อมูล

การพัฒนาโปรแกรมประยุกต์ Visual Studio 2008 ส่วนใหญ่นำมาใช้กับฐานข้อมูล และการประยุกต์แบบ client / server ซึ่ง Visual Studio 2008 เป็นเครื่องมือที่อำนวยความสะดวกให้กับผู้พัฒนาโปรแกรมการเข้าถึงข้อมูล ส่วนความสามารถใหม่ที่สัมพันธ์กับฐานข้อมูลของ Visual Studio 2008 มีพื้นฐานจาก ActiveX Data Object (ADO) ซึ่งเป็นเทคโนโลยีที่ให้ผู้ใช้เข้าถึงฐานข้อมูลหรือแหล่งข้อมูล เมื่อมีการใช้ OLE DB provider ติดต่อกับแหล่งข้อมูล Visual Studio 2008 มีวิธีการติดต่อกับฐานข้อมูลได้หลายวิธี

## ODBC

ODBC ย่อมาจาก Open Data Connectivity และตั้งคำกรทำงานให้ผู้ใช้ติดต่อกับฐานข้อมูลทั้งภายในพื้นที่และระยะไกล Microsoft เสนอเทคโนโลยีที่เป็นวิธีการเข้าถึงฐานข้อมูลหลายประเภท เช่น dBase, Microsoft FoxPro, Microsoft Access, Microsoft SQL Server, Oracle รวมไปถึง text แบบ comma-delimited ด้วยการใช้ API ร่วมกัน ส่วนโปรแกรมประยุกต์ทำงานกับ DLL เรียกว่า ODBC driver manager ซึ่งจะส่งคำสั่งไปที่ไคร์ฟเวอร์ ODBC ในการรับฐานข้อมูลที่ผู้ใช้ต้องการสั่งทำทายของ ODBC เป็นการให้ติดต่อกับฐานข้อมูลประเภทต่างๆ ในทางทฤษฎี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้เขียนโปรแกรมประยุกต์สามารถเตรียมใช้ ODBC ติดต่อกับฐานข้อมูล Access และเปลี่ยนขนาดไปที่ฐานข้อมูล SQL Server โดยการเปลี่ยนไคลเอนต์ back-end ของ ODBC และมีคำสั่งไม่มาก การทำสิ่งเหล่านี้ได้เนื่องจากคำสั่งที่ส่งไปยังฐานข้อมูล คือ คำสั่งมาตรฐาน SQL ภาษา SQL (Structure Query Language) เป็นภาษาโปรแกรมที่ใช้กับฐานข้อมูล ในการปฏิบัติเลเยอร์ ODBC สามารถแปลงคำสั่ง SQL ให้เป็นภาษาเฉพาะของฐานข้อมูล ผู้เขียนโปรแกรม ODBC มักจะข้าม engine การแปลของ ODBC และส่งคำสั่งโดยตรงกับฐานข้อมูล ODBC มีประสิทธิภาพเมื่อเปรียบเทียบกับเทคนิคการเข้าถึงข้อมูล ความได้เปรียบของ ODBC คือสนับสนุน API ทั้งประเภท 16 บิต และ 32 บิต ใน ODBC เวอร์ชัน 3 เพิ่มเทคนิคการบูตให้ดีขึ้น เช่น การติดต่อแบบ pool ทำให้โปรแกรมประยุกต์ตอบสนองได้ดีขึ้น Microsoft Transaction Server ใช้การติดต่อแบบ pool เพื่อความเร็วในการเปิดการติดต่อโดย component ของ ActiveX ที่ทำงานอยู่ภายใต้ ODBC

การใช้ ODBC ไม่ใช่เรื่องง่ายโดยเฉพาะอย่างยิ่งสำหรับผู้พัฒนาโปรแกรมด้วย Visual Studio 2008 กลุ่มของฟังก์ชัน API มีความซับซ้อนและถ้ามีความผิดพลาดเกิดขึ้นอาจจะให้โปรแกรมประยุกต์เสียหาย ด้วยเหตุนี้ผู้พัฒนาโปรแกรม Visual Studio 2008 น้อยมากที่จะเขียนคำสั่งเรียกฟังก์ชัน ODBC โดยตรง เทคนิคการแก้ไขข้อมูลอื่น ๆ ส่วนใหญ่ให้ Visual Studio 2008 ใช้ไคลเอนต์ ODBC ในฐานะเลเยอร์กลาง ในบางครั้งอาจใช้เทคนิคอื่น ๆ กับการเรียก API โดยตรง แต่ไม่สามารถใช้วิธีการนี้กับ ADO ได้ถึงแม้ว่าภายใน ADO จะใช้ไคลเอนต์ ODBC แนวคิดพื้นฐานของเทคโนโลยี เช่น เมื่อมีการทำงานกับ ADO มีความเกี่ยวข้องกับ Data Source Name (DSN) โดย DSN เป็นกลุ่มของค่าที่โปรแกรมประยุกต์ในการติดต่อกับฐานข้อมูล โดยทั่วไปจะรวมถึงชื่อไคลเอนต์ของ ODBC ที่ต้องการใช้ ชื่อของไคลเอนต์เป็น host ของแม่ข่ายฐานข้อมูล (การทำงาน กับ Client-Server engine เช่น SQL Server หรือ Oracle), ชื่อพาร์ทของฐานข้อมูลที่เจาะจง, เวลาสิ้นสุดของการติดต่อ, ชื่อของการเรียกเวิร์คสเตชัน หรือโปรแกรมประยุกต์ เป็นต้น การสร้าง DSN ทำได้หลายวิธี ทั้งภายในและภายนอก Visual Studio 2008 ศูนย์กลางของคำสั่ง ODBC อยู่ใน control Panel ของระบบปฏิบัติการ Windows ที่ให้ผู้ใช้ติดต่อสร้าง DSN และคอนฟิก ODBC ซึ่งมี DSN หลายประเภท DSN ของผู้ใช้จะเก็บไว้ใน Registry ของระบบ และสามารถใช้ได้เฉพาะผู้ใช้ปัจจุบัน โดยไม่สามารถแบ่งให้ผู้ใช้ได้ DSN ของระบบเก็บไว้ใน Registry เช่นกัน และสามารถเห็นได้โดยผู้ใช้อื่น ๆ รวมถึงแม่ข่าย Windows NT ไฟล์ DSN เก็บในไฟล์ dsn และสามารถใช้ร่วมกันโดยผู้ใช้ทั้งหมด ไฟล์ DSN สามารถคัดลอกได้ง่าย ทำให้การติดตั้งทำได้สะดวก อีกด้านหนึ่งโปรแกรมประยุกต์ที่ต้องการที่เก็บ DSN ต้องใช้คำสั่งที่ระบุพาร์ทของไฟล์ DSN ที่สมบูรณ์ การทำงานกับ ODBC โดยไม่เกี่ยวข้องกับ DSN ทำได้โดยการให้สารสนเทศที่ต้องการสำหรับการติดต่อ เช่น ไคลเอนต์, ชื่อฐานข้อมูล และพาร์ท เป็นต้นให้ถูกตั้งในคำสั่ง วิธีนี้เรียกว่า DSN-Less

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Connections ซึ่งมีประสิทธิภาพเพราะประหยัดของไดร์ฟเวอร์ ODBC ในการเดินทางไป Registry หรือไฟล์ DSN แต่วิธีการนี้ ผู้พัฒนาโปรแกรมต้องทำงานมากขึ้น

## DAO

DAO หรือ Data Access Object เป็นสิ่งสำคัญของโปรแกรมประยุกต์ Visual Basic 3 ในพัฒนาการประยุกต์กับฐานข้อมูล DAO เป็นการคิดต่อแบบ Object-oriented ไปยัง Microsoft Jet ที่เป็น engine ที่มีความสามารถสูง ผู้พัฒนาโปรแกรมสามารถออกแบบฐานข้อมูล MDB ด้วย Access และใช้ DAO จากโปรแกรมประยุกต์ Visual Basic ในการเปิดฐานข้อมูล เพิ่มและลบเรคคอร์ด และจัดการทรานแซคชัน สิ่งที่ที่ที่สุดของ DAO คือไม่จำกัดผู้ใช้กับ Jet database เพราะผู้ใช้สามารถเปิดฐานข้อมูลทุกชนิดที่มีไดร์ฟเวอร์ ODBC ได้โดยตรงหรือผู้ใช้สามารถใช้ Jet attached table ซึ่งเป็น table เสมือนที่ปรากฏตามฐานข้อมูล MDB แต่การดึงและเก็บข้อมูลจริงในแหล่งอื่นของ ODBC

## RDO

RDO หรือ Remote Data Objects เป็นความพยายามครั้งแรกของ Microsoft ในการรวมความง่ายของ DAO กับความสามารถระดับสูงของ Direct ODBC API Programming โดย RDO เป็นแบบจำลองออบเจกต์ที่ไม่ชัดเจน ภายหลัง DAO แต่ใช้การข้าม Jet engine และ DLL ของ DAO และทำงานโดยตรงกับไดร์ฟเวอร์ ODBC โปรแกรมประยุกต์ที่มาจาก RDO โหลดเฉพาะ DLL จำนวนหนึ่งแทนการใช้ทรัพยากรจำนวนมากของ Jet engine โดยสิ่งสำคัญอยู่ที่การออกแบบเฉพาะของ RDO ให้ทำงานกับทรัพยากรของ ODBC ทำให้สามารถทำงานที่ไม่สามารถเข้าถึงได้โดย DAO เทคโนโลยีของ RDO เป็นเทคโนโลยี 32 บิต จึงไม่สามารถใช้ได้จากโปรแกรมประยุกต์ 16 บิต

## OLE DB

OLE DB เป็นเทคโนโลยีการเข้าถึงข้อมูลระดับล่างที่ Microsoft ผู้พัฒนาให้เข้ามาแทนที่ ODBC ในฐานะวิธีการหลักในการติดต่อกับฐานข้อมูล ส่วนของ OLE DB ที่เทียบได้กับไดร์ฟเวอร์ของ ODBC คือ OLE DB provider ซึ่งทำงานเป็นการเชื่อมโยงระหว่างโปรแกรมประยุกต์กับฐานข้อมูล OLE DB เป็นเทคโนโลยีใหม่ แต่ OLE DB provider พบได้ในฐานข้อมูลจำนวนมาก ถึงแม้ว่า ODBC และ OLE DB มีความคล้ายคลึงกัน แต่มีข้อแตกต่างกัน คือ OLE DB มีพื้นฐานจาก COM ซึ่งเป็นสถาปัตยกรรมที่ให้การถ่ายโอนข้อมูลจำนวนมากผ่านเครือข่าย ประการต่อมา OLE

DB ทำหน้าที่ในการติดต่อกับแหล่งข้อมูล ไม่เหมือนกับไดร์ฟเวอร์ ODBC ที่เป็นการสัมพันธ์กับฐานข้อมูล หรือ ISAM ( Indexed sequential access mode)

OLE DB เป็นยุทธศาสตร์ Microsoft's Universal Data Access ซึ่งเป็นการให้ผู้ใช้อ่านและประมวลผลข้อมูลโดยไม่ต้องแปลง และนำเข้ามายังฐานข้อมูลแบบแผน การใช้ OLE DB provider ทำให้ผู้ใช้สามารถประมวลผลข้อมูลใน E-Mail, Message, HTML, กระจาย, เอกสารข้อความและแหล่งข้อมูลภายนอก Visual Basic 6 มี provider ของ Microsoft Jet, SOL Server, FoxPro, ไฟล์ข้อความและฐานข้อมูล Oracle สำหรับ OLE DB provider อื่น ๆ สามารถดาวน์โหลดจากเว็บของ Microsoft

## ADO

ADO หรือ ActiveX Data Object เป็นการติดต่อระดับสูงของ OLE DB มีบทบาทใกล้เคียง RDO ในทำงานกับ API ของ ODBC ในขณะที่ OLE DB คล้ายกับ API ของ ODBC ที่เป็นการติดต่อระดับล่างที่ไม่สามารถเข้าถึงได้ง่ายจากภาษาระดับสูง เช่น Visual Basic เป็นต้น ADO สร้างบน OLE DB เพื่อให้การทำงานที่ไม่ให้ติดต่อโดยตรง ODBC หรือทำให้ผู้ใช้เขียนคำสั่งที่มีความสามารถ ADO สามารถเปรียบเทียบความสามารถกับ ADO คือทั้งคู่สามารถสร้างคิวรีแบบ asynchronous และการติดต่อ ADO เพิ่มส่วนใหญ่มาก เช่น File-Based และ Stand-Alone Recordset, Hierarchical Recordset และอื่นๆ

## 2.7 ASP.NET

โลกในยุคปัจจุบันนี้กำลังพัฒนาไปสู่โลกแห่งการสื่อสารไร้พรมแดน ซึ่งอาจได้เคยใช้เทคโนโลยีและการบริการพิเศษต่างๆมาบ้างแล้ว ไม่ว่าจะจากสื่อโฆษณา โทรทัศน์ การเช็คเมลล์ เล่นอินเทอร์เน็ต รวมทั้งของตัวชมภาพยนตร์ผ่านทางโทรศัพท์มือถือ สิ่งเหล่านี้ล้วนเป็นเทคโนโลยีที่ทันสมัยและอำนวยความสะดวกสบายต่อผู้ใช้เป็นอย่างมาก

### 1. Web Service

Web Service คือการใช้บริการต่างๆผ่านทางเว็บ ซึ่งก็ตรงตามจุดประสงค์ของไมโครซอฟท์ที่ต้องการจะทำให้ทุกอย่างสามารถใช้งานเชื่อมต่อกับอินเทอร์เน็ตได้ Web Service คือการเรียกใช้งานชุดคำสั่งในระยะไกล ซึ่งชุดคำสั่งเหล่านี้ไม่ใช่แค่ชุดคำสั่งธรรมดา แต่เป็นชุดคำสั่งที่เขียนขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นโปรแกรมเหมือนกับการใช้งานของ Client ทำให้ผู้ใช้ไม่ต้องไปติดตั้งโปรแกรมต่างๆภายในเครื่องคอมพิวเตอร์ของผู้ใช้ ผู้ใช้สามารถเข้าอินเทอร์เน็ตใช้บริการในเว็บไซด์ผู้ผลิตได้ทันทีโดยอาจมีการเรียกเก็บค่าบริการเป็นครั้งๆไป ซึ่งจะช่วยลดปัญหาในการละเมิดลิขสิทธิ์ และชุดคำสั่งเหล่านี้จะทำให้ ASP.NET มีบทบาทมาก

ไมโครซอฟท์ต้องการที่จะสร้างอะไรที่เป็นมาตรฐานขึ้นมา เพื่อให้ทุกสิ่งทุกอย่างสามารถติดต่อสื่อสารกันได้หมด โดยคิดค้นระบบซึ่งให้เป็นระบบมาตรฐาน ระบบนี้คือ .NET Framework ซึ่งระบบนี้ไม่ใช่ระบบปฏิบัติการ (OS) แต่เปรียบเสมือนโปรแกรมหนึ่งที่จะสามารถสร้างสภาวะแวดล้อมหนึ่ง ซึ่งสามารถทำงานในระบบ .NET นี้ได้

เอเอสพีคอตเน็ต (อังกฤษ: ASP.NET) คือเทคโนโลยีสำหรับพัฒนาเว็บไซด์ เว็บแอปพลิเคชัน และเว็บเซอร์วิส ซึ่งเป็นส่วนหนึ่งของคอตเน็ตเฟรมเวิร์ก พัฒนาโดยไมโครซอฟท์

ASP.NET เป็นรุ่นถัดจาก Active Server Pages (ASP) แม้ว่า ASP.NET นั้นจะใช้ชื่อเดิมจาก ASP แต่ทั้งสองเทคโนโลยีนั้นแตกต่างกัน โดยไมโครซอฟท์ได้สร้าง ASP.NET ขึ้นมาใหม่หมดบนพื้นฐานจาก Common Language Runtime (CLR) ซึ่งทำให้ผู้พัฒนาสามารถเลือกใช้ภาษาใดก็ได้ที่รองรับ โดย NET Framework เช่น C# และ VB.NET เป็นต้น ปัจจุบันรุ่นล่าสุดคือ ASP.NET 2.0 ซึ่งรวมอยู่ใน .NET Framework 2.0. และ .NET Framework 3.0.

ไมโครซอฟท์หวังที่จะนำเอาระบบนี้ไปติดตั้งลงบนอุปกรณ์ทุกชนิด เพื่อให้ใช้อุปกรณ์ทุกอย่างมีระบบๆหนึ่งที่เหมือนกันหมด โดย .NET Framework นั้นมี ส่วนประกอบภายในแบ่งออกเป็น 3 ชั้นใหญ่ๆคือ

1. Programming Language : เป็นรูปแบบของภาษาที่ออกแบบมาเพื่อให้สามารถทำงานในสภาวะที่เป็น .NET ได้โดยที่ทางไมโครซอฟท์ได้เปิดตัวภาษาหลักๆที่จะใช้พัฒนามบน .NET นี้ 3 ภาษา

- C# เป็นภาษาใหม่ที่ไมโครซอฟท์พัฒนามาจาก C++ กับ JAVA เป็นหลัก
- VB.NET เป็นภาษาที่พัฒนามาจาก Visual Basic ในเวอร์ชัน 6.0
- JScript.net เป็นภาษาที่พัฒนามาจาก JScript ซึ่งเป็น JavaScript ในเวอร์ชันของ

ไมโครซอฟท์

2. Base Classes Library : Library เปรียบเสมือนชุดคำสั่งสำเร็จรูปย่อยๆที่เพิ่มเข้ามา ซึ่งส่วนใหญ่เป็นชุดคำสั่งที่ต้องใช้งานอยู่เป็นประจำ ดังนั้นจึงมีผู้คิดค้นเพื่ออำนวยความสะดวกในการเขียนโปรแกรม ซึ่ง Library ในภาษาต่างๆส่วนใหญ่อยู่ในรูปแบบไฟล์ include แต่ถ้าเป็น ASP สิ่งที่เป็น Library ก็คือ คอมโพเนนต์ต่างๆ ซึ่งภายในระบบ .NET จะสร้างสิ่งที่เรียกว่าเป็น Library พื้นฐานขึ้น ทำให้ไม่ว่าจะใช้ภาษาใดในการพัฒนาโปรแกรมก็สามารถที่จะเรียกใช้ Library ที่เป็นตัวเดียวกันได้หมด

3. Common Language Runtime (CLR) : นับเป็นสิ่งสำคัญที่สุดของระบบ .NET นี้ก็ว่าได้ เพราะ CLR ที่ว่านี้มีหน้าที่ทำให้โปรแกรมที่เขียนขึ้นมาด้วยภาษาต่างๆกัน กลายเป็นภาษารูปแบบมาตรฐานเดียวกัน ทั้งหมด เราเรียกภาษาที่ว่านี้ว่า Intermediate language (IL) ซึ่งเมื่อต้องการรันโปรแกรมใด CLR จะทำการตรวจสอบเครื่องที่รันว่ามีสถานะแวดล้อมการทำงานเช่นใดหลังจากนั้น คอมไพล์เป็นโปรแกรมที่เหมาะสมต่อการทำงานของเครื่องนั้น ทำให้เราสามารถใช้งาน โปรแกรมต่างๆได้อย่างมีประสิทธิภาพสูงสุดในแต่ละเครื่อง

## 2. ประโยชน์และข้อดีของ .NET Framework มีดังนี้

1. เป็นระบบที่มีไลบรารีที่เป็นมาตรฐานเดียวกัน เนื่องจากมีไลบรารีที่เป็นมาตรฐานเดียวกันทั้งหมดทำให้เราไม่ต้องกังวลว่าภาษาที่ใช้เขียนนั้นมีไลบรารีตัวนั้นตัวนี้หรือไม่ รวมทั้งไม่ต้องคอยกังวลว่าถ้าใช้ไลบรารีของภาษาหนึ่งแล้วอีกภาษาหนึ่งจะไม่มีไลบรารีตัวนั้น

2. ไม่ขึ้นกับระบบปฏิบัติการ (OS) เนื่องจากระบบปฏิบัติการ ที่แต่ละบุคคลหรือองค์กรใช้นั้นย่อมไม่เหมือนกัน แต่ภายใน .NET Framework จะไม่มีปัญหานี้ของเพียงแค่มีระบบ .NET Framework ก็จะทำให้สามารถใช้งานโปรแกรมต่างๆได้ ซึ่งเป็นข้อดีตรงที่เราจะสามารถใช้โปรแกรมต่างๆได้ทุกระบบปฏิบัติการ

3. ใช้ในการพัฒนาได้ทุกภาษา ทำให้เราไม่ต้องคอยมาศึกษาภาษาใหม่ๆเมื่อต้องการสร้างโปรแกรมในแต่ละครั้ง นอกจากนั้นเรายังสามารถเลือก ใช้ภาษาที่ถนัดที่สุดในการพัฒนาโปรแกรมต่างๆ

4. มีการควบคุมสิ่งแวดล้อมในการทำงานเป็นอย่างดี เนื่องจากเป็นระบบที่เป็นมาตรฐาน ทำให้การควบคุมจัดสรรระบบต่างๆ ทำได้ง่ายขึ้น ไม่ว่าจะเป็นการจัดสรรหน่วยความจำด้านการใช้งานเครื่องก็มีความรวดเร็วมากขึ้นลดโอกาสที่เครื่องจะแฮกก็ได้เป็นอย่างดี

5. ความปลอดภัยที่มีมากขึ้น .NET Framework สามารถกำหนดสิทธิ์การใช้งานหรือ permission ของผู้ใช้งานได้มากขึ้นทำให้สามารถกำหนดว่า จะให้โปรแกรมในส่วใดใช้งานได้หรือไม่ได้ แล้วแต่เฉพาะบุคคล

ASP.NET หรืออีกชื่อหนึ่งว่า ASP+ ซึ่งเป็นชื่อที่ไม่โครซอฟท์ใช้เรียกในตอนแรก ถือว่าเป็น ASP เวอร์ชันล่าสุดต่อจาก ASP 3.0 แต่คงไม่สามารถพูดได้เต็มปากว่า ASP.NET พัฒนามาจาก ASP เพราะรูปแบบ และไวยากรณ์ต่างๆ และภาษาที่นำมาใช้งานนั้นต่างจากเดิมแทบทั้งสิ้น แทบจะเรียกได้ว่ายกเครื่องใหม่เลยทีเดียว น่าจะพูดได้ว่า ASP.NET เป็นอีก Generation หนึ่งของ ASP มากกว่า ข้อแตกต่างระหว่างทั้งสองอย่างนี้คือ

1. ใช้ภาษาใดๆในการเขียนสคริปต์ก็ได้ จากเดิมที่สามารถใช้ได้เฉพาะภาษาที่เป็นสคริปต์ของ VBScript และ JScript แต่ใน ASP.NET สามารถที่จะใช้ภาษาที่มีรูปแบบของภาษาเต็มๆ ซึ่ง ในเบื้องต้น มี 3 ภาษาคือ C#, VB.NET และ JScript.Net ที่ออกมาเป็นมาตรฐาน แต่ในอนาคต ไมโครซอฟท์มีแผนที่จะเพิ่มตัวแปลภาษาให้ครบทุกภาษา

2. มีความยืดหยุ่นในการเขียนโปรแกรม โดยสามารถใช้ภาษาในการเขียน ASP.NET ได้มากกว่า 1 ภาษาภายในไฟล์เดียวกัน ทำให้สามารถเลือกรูปแบบของภาษาที่ง่ายที่สุดต่อการเขียน ในแต่ละส่วนได้

3. ลักษณะการแปลภาษาและนามสกุลไฟล์เปลี่ยนไป ใน ASP เวอร์ชันก่อนๆ มีลักษณะการแปลภาษาเป็นแบบอินเตอร์พรีเตอร์ (Interpreter) คือการจะทำคำสั่งใดค่อยแปลคำสั่งนั้น แต่ในเวอร์ชัน .NET นี้จะมี ลักษณะเป็นคอมไพเลอร์ (Compiler) คือการแปลคำสั่งรวมทั้งโปรแกรม นอกจากนี้นามสกุลของไฟล์ก็มีการเปลี่ยนแปลงจากเดิมที่ใช้นามสกุลไฟล์เป็น “.asp” เป็น “.aspx “

4. รูปแบบและการใช้งานคอมโพเนนต์ที่ง่ายขึ้น รูปแบบของคอมโพเนนต์จะเน้นไปที่ XML มากที่สุด และที่สำคัญคือการใช้งานคอมโพเนนต์ใน ASP.NET นั้นเราสามารถอัพโหลดไฟล์ไปไว้ในไคลเอนต์ที่ผู้ดูแลเซิร์ฟเวอร์ (Admin) กำหนดหลังจากนั้นคอมโพเนนต์จะติดตั้งตัวเองโดยอัตโนมัติ ลดปัญหาที่เกิดจาก ASP เวอร์ชันก่อนๆได้เป็นอย่างดี เนื่องจากใน ASP เวอร์ชันก่อนนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดตั้งคอมโพเนนต์กระทำได้เพียงผู้ดูแลเซิร์ฟเวอร์เพียงคนเดียวเท่านั้น ทำให้เวลาต้องการใช้คอมโพเนนต์ต่างๆที่เซิร์ฟเวอร์ไม่มี จึงเป็นเรื่องที่ลำบาก

5. มีไลบรารีให้เลือกใช้ได้มากขึ้น ใน ASP เวอร์ชันก่อนๆนั้นแอปพลิเคชันบางอย่างสร้างได้ไม่สะดวกนัก ต้องอาศัยคอมโพเนนต์ต่างๆมากมาย แต่ใน ASP.NET นั้นได้เพิ่มไลบรารีในส่วนเหล่านี้ให้กลายเป็นพื้นฐานของการใช้งาน

6. มีคอนโทรลทำให้การใช้งานในบางสิ่งง่ายขึ้น เป็นส่วนพิเศษที่เพิ่มเติมมาจาก ASP รุ่นก่อนๆที่ไม่มีส่วนที่เรียกว่า คอนโทรล ซึ่งคอนโทรลนี้จะช่วยให้เราสามารถสร้างเว็บไซต์ได้อย่างง่ายและมีประสิทธิภาพมากขึ้น จึงไม่ต้องกังวลว่าบราวเซอร์รุ่นนั้นรุ่นนี้จะรองรับกับภาษาที่เราเขียนหรือไม่

7. สามารถเรียกขอข้อมูลจากเซิร์ฟเวอร์ได้ ใน ASP เวอร์ชันก่อนๆเซิร์ฟเวอร์สามารถเรียกขอข้อมูลได้จากเครื่องผู้ใช้เท่านั้นแต่ใน ASP.NET เครื่องเซิร์ฟเวอร์สามารถเรียกขอข้อมูลจากเครื่องเซิร์ฟเวอร์ด้วยกันได้

8. ไม่ต้องต่อฮาร์ดแวร์ (Hardware) เนื่องจากเป็นระบบใน .NET Framework ดังนั้นจึงมีคุณสมบัติของ Common Language Runtime (CLR) ทำให้มีการคอมไพล์โปรแกรมเป็นภาษามาตรฐานที่เรียกว่า IL ก่อน ดังนั้นไม่ว่าจะเล่นเครื่องปาล์มหรือโน้ตบุ๊ก PDA ก็ไม่เกิดปัญหา

9. ง่ายต่อการหาจุดผิดพลาดในการเขียนโปรแกรม หากเป็น ASP รุ่นก่อนเวลาเกิดความผิดพลาด (error) เครื่องจะบอกแค่ว่าเป็นความผิดพลาดชนิดใดบรรทัดไหน แต่ใน ASP.NET นี้เครื่องจะแสดงรายละเอียดที่มากขึ้น พร้อมแนวทางแก้ไข

10. มีการตรวจสอบเหตุการณ์ต่างๆได้ภายในเว็บเพจ มีการตรวจสอบเหตุการณ์ต่างๆตั้งแต่โหลดหน้าเว็บเพจไปจนถึงปิดหน้าเว็บเพจลง ทำให้เราสามารถเขียนโปรแกรมกำหนดเหตุการณ์ต่างๆได้ง่ายขึ้น

11. แยกส่วนที่เป็น HTML กับ ASP ออกมาอย่างชัดเจน ในเวอร์ชันก่อนๆส่วนที่เป็น HTML กับ ASP จะเขียนปนกันไปมา แต่ในเวอร์ชันนี้จะแยกส่วนกันอย่างชัดเจนว่าส่วนไหนเป็น HTML และส่วนไหนเป็น ASP

### 3. ข้อได้เปรียบหลักของ ASP.NET ระหว่าง ASP

- โค้ดจะได้รับการ compiled ทำให้การทำงานรวดเร็วขึ้นและช่วยจับข้อผิดพลาดในช่วงการออกแบบได้
- ระบบการจัดการข้อผิดพลาด (Exception handling) ที่ดีขึ้นกว่าเดิม
- ใช้วิธีการพัฒนาเว็บแอปพลิเคชัน อย่างการใช้ Controls หรือ Events ซึ่งทำให้การพัฒนาง่าย และดูดีขึ้น
- มีหลากหลาย controls และไลบรารีพร้อมในการใช้งานให้เลือกเพื่อการพัฒนาที่สะดวก และรวดเร็วขึ้น
- สามารถพัฒนาได้หลากหลายภาษาที่รองรับคอตเน็ต เช่น C# VB.NET J# เป็นต้น
- สามารถทำการแคชได้ทั้งหน้า หรือส่วนหนึ่งของหน้าที่ต้องการ
- สามารถแยกส่วนโค้ดออกมาต่างหากจากหน้ารูปแบบ
- Session สามารถเลือกที่จะบันทึกในฐานข้อมูลได้ ทำให้ session ไม่สูญหายหากมีการรีเซตเว็บเซิร์ฟเวอร์ เป็นต้น
- รองรับมาตรฐานเว็บดีซึ้นกว่าเดิม รวมถึงการทำงานร่วมกับ CSS

### รูปแบบไฟล์ ASPX

ASPX เป็นชื่อรูปแบบไฟล์ของหน้าแบบฟอร์ม ASP.NET โดยทั่วไปแล้วในไฟล์จะมีรหัสแบบ HTML หรือ XHTML ซึ่งใช้กำกับรูปแบบฟอร์ม หรือ เนื้อหาในหน้าเว็บ และในส่วนของโค้ดนั้น อาจจะอยู่ในหน้าเดียวกันในแท็ก หรือ บล็อก `<% -- รหัสที่ใช้ -- %>` (โดยในกรณีนี้จะคล้ายกับเทคโนโลยีที่ใช้พัฒนาเว็บ อย่าง PHP และ JSP) หรือแยกอยู่ในหน้าโค้ดออกมาต่างหาก (Code behind) ASP.NET รองรับการเขียน โค้ดในบรรทัดเดียวกันทั้งหมดในไฟล์ ASPX แต่วิธีนี้นั้นเป็นวิธีที่ไม่แนะนำ

### ไฟล์อื่นๆ

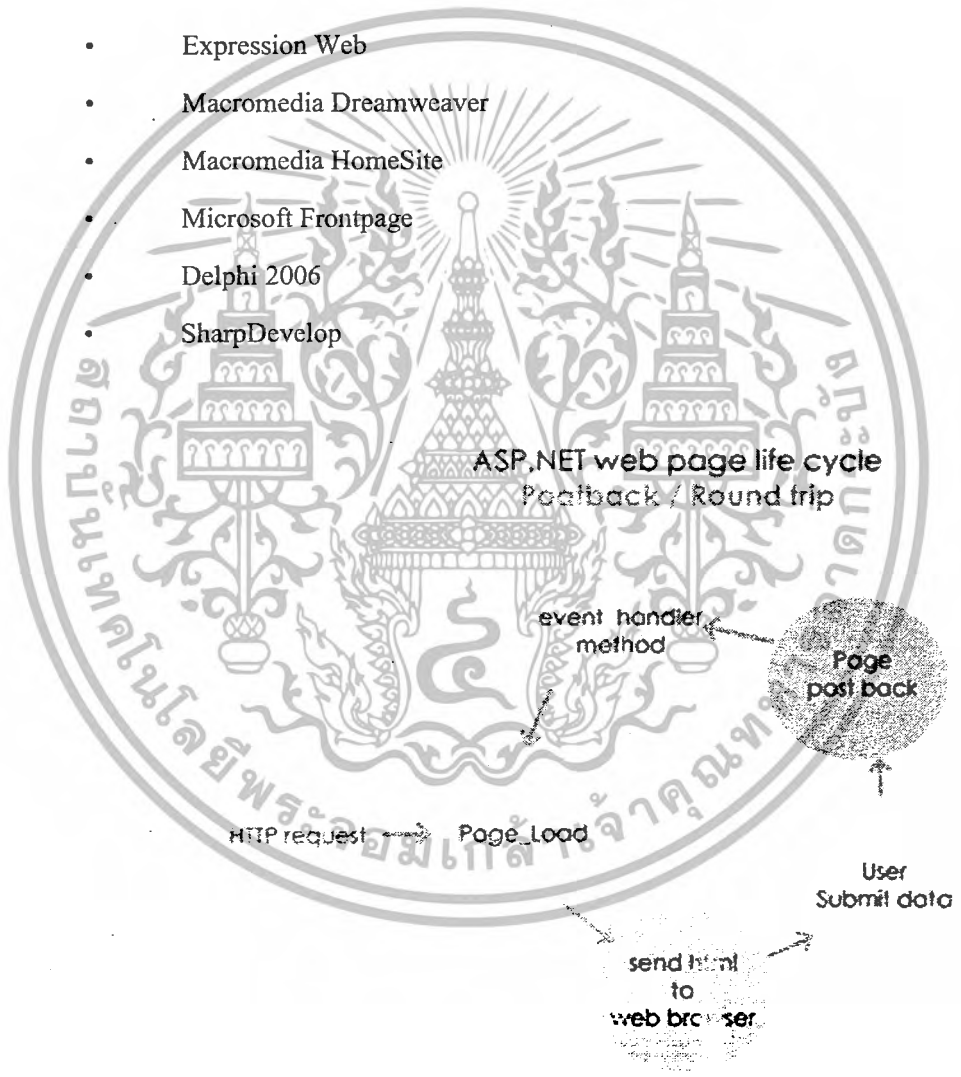
ไฟล์รูปแบบอื่นๆ ที่เกี่ยวกับ ASP.NET มีดังนี้

- ASCX - control ที่พัฒนาขึ้นมาเอง
- ASMX - หน้าเว็บเซอร์วิส
- ASHX - ตัวจัดการ HTTP ด้วยตนเอง

เครื่องมือในการพัฒนา

รายชื่อโปรแกรมที่สามารถใช้ในการพัฒนา ASP.NET

- Visual Studio 2008
- Visual Studio 2005
- Visual Studio 2003
- Visual Web Developer 2005 Express Edition
- ASP.NET Web Matrix
- Expression Web
- Macromedia Dreamweaver
- Macromedia HomeSite
- Microsoft Frontpage
- Delphi 2006
- SharpDevelop

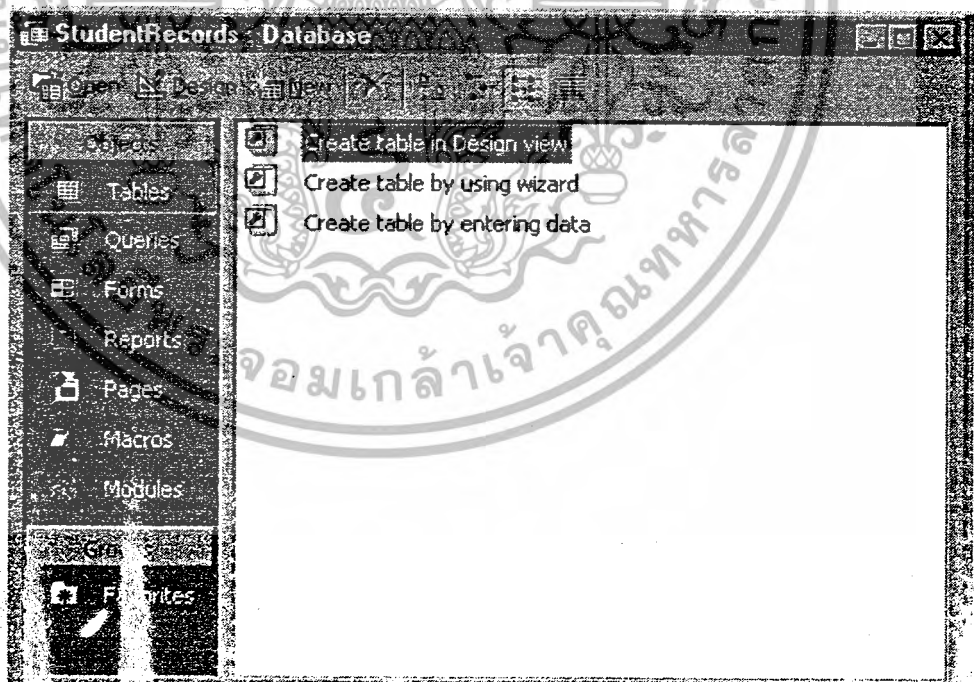


รูปที่ 2.13 การทำงานของหน้าเว็บแบบ ASP.NET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.8 Microsoft Access

ไมโครซอฟท์แอคเซส (Microsoft Access) คือ โปรแกรมเพื่อพัฒนาระบบฐานข้อมูล มีตารางเก็บข้อมูลและสร้างคิวรีได้ มีส่วนคอนโทรลให้เรียกใช้ในรายงานและฟอร์ม สร้างมาโครและโมดูลด้วยภาษาเบสิกเพื่อประมวลผลตามหลักภาษาโครงสร้าง หรือจะใช้เป็นเพียงระบบฐานข้อมูลให้โปรแกรมจากภายนอกเรียกใช้ ซึ่งง่ายสำหรับผู้ที่มีประสบการณ์ ในเรื่องการเขียนโปรแกรมหรือผู้พัฒนาระบบฐานข้อมูลมาแล้ว ช่วยให้การพัฒนาระบบงานเสร็จได้อย่างรวดเร็ว ไมโครซอฟท์แอคเซส (Microsoft Access) ต่างกับ วิววลเบสิก (Visual Basic) เพราะ วิววลเบสิกสามารถพัฒนาโปรแกรมได้หลากหลาย เช่น พัฒนาโปรแกรมควบคุมอุปกรณ์ โปรแกรมประยุกต์ เกมส์ หรือเชื่อมต่อกับระบบฐานข้อมูลภายนอก เป็นภาษาที่เหมาะสมกับการพัฒนาโปรแกรมประยุกต์ ส่วนไมโครซอฟท์แอคเซส เหมาะสำหรับผู้พัฒนาระบบฐานข้อมูลที่ไม่ต้องการระบบที่ซับซ้อนและต้อง การพัฒนาให้เสร็จอย่างรวดเร็ว มีเครื่องมือที่อำนวยความสะดวกในการพัฒนาระบบฐานข้อมูลอย่างครบถ้วน



รูปที่ 2.14 ตัวอย่าง Microsoft Access

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## การทำงานของ Microsoft Access

1. เป็นฐานข้อมูล
2. สร้างรายงาน (ในลักษณะ ของจดหมายเวียน)
3. สร้างตัวเองให้เป็นโปรแกรม สำหรับเก็บข้อมูลต่างๆ
4. ฟังก์ชันในการใช้งาน SQL หรือ เรียกแบบไทยๆ ง่ายๆ คือการดึงข้อมูลเฉพาะเจาะจงของฐานข้อมูลนั้นๆ

เนื่องจาก Access ทุก version ยกเว้น 2007 จะมีหน้าแบบเดียวกันหมด

เริ่มจากซ้ายบน เมนู Objects ---> Tables

- เมนูนี้คือการสร้างฐานข้อมูล ซึ่งปกติถ้าผู้ใช้ที่เคยได้เรียนเกี่ยวกับ ฐานข้อมูล Mysql Oracle SQL Server ฐานข้อมูลใช้จริงไม่ได้ใช้เยอะมากนัก ยกเว้นว่าจะใช้ฐานข้อมูลเดี่ยวๆที่ลงลึกรายพวกนั้น

ชนิดของข้อมูล ( Data Type )

- Number คือ ตัวเลข ชนิดของข้อมูลในต้องเป็นตัวเลขอย่างเดียว ไม่สามารถพิมพ์ ภาษาอะไรลงไปได้ -- เช่น -121 , 200.14 เป็นต้น
- Text คือ ถ้า พยางค์ ของภาษาต่างๆ รวมถึงตัวเลขด้วย เช่น สวัสดีครับ , Hey How r u ? , นาวา 34 เป็นต้น
- Date/Time คือ ข้อมูลจะถูกเก็บในรูปของวันและเวลา ซึ่งลักษณะของวันและเวลา จะเป็นรูปแบบตามนี้ ปี ค.ศ : เดือน : วันที่ ชั่วโมง : นาที : วินาที เช่น 2008-08-18 14:20:21 โปรแกรมก็อาจจะแสดงผลออกมาให้ดูว่า 18/08/08 14:20:21

- Memo คือ คล้ายกับการจดบันทึก ข้อมูลตรงจะไม่มีกำหนดความยาวสั้นจะไม่มีโชว์ข้อมูลให้เห็น แต่ถ้าเราเอาเมาส์ไปชี้ ข้อมูลก็โชว์ออกมาให้

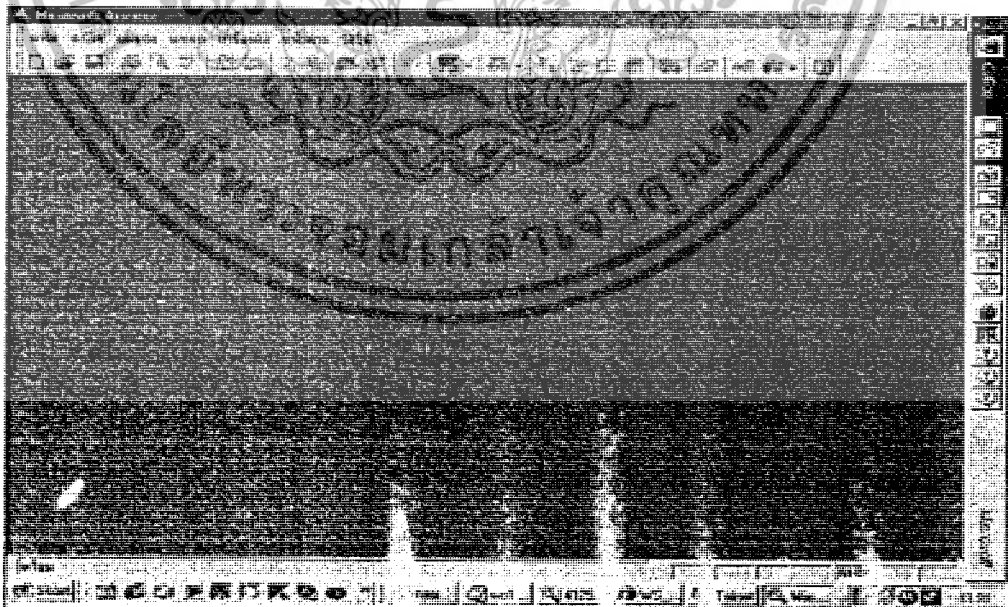
**\*\* ข้อมูลทั้ง ๖ ชนิดนี้เป็นข้อมูลที่ใช้ร่วมกันทุกฐานข้อมูล\*\***

### 2.8.1 ส่วนต่างๆของ Access

1. Table เป็นการเก็บข้อมูลในแนว Row และ Column สามารถออกแบบ Table เอง หรือสร้างจาก Table Wizard
2. Query เป็นการแสดงข้อมูลใน Table โดยสามารถเลือกแสดงเฉพาะ Field ที่ต้องการได้ นอกจากนี้สามารถสร้างฟิลด์ใหม่และคำนวณเพิ่มเติมได้
3. Form เป็นการสร้างฟอร์มในการเพิ่ม แก้ไข ลบ และดูข้อมูล
4. Report เป็นการสร้างรายงาน เพื่อนำเสนอข้อมูลในรูปแบบที่ต้องการ
5. Macro เป็นการนำคำสั่งมาเขียนเป็นลำดับขั้นตอนเพื่อเพิ่มประสิทธิภาพของโปรแกรม
6. Module เป็นการเขียนโปรแกรมโดยใช้ ภาษา Access Basic

### 2.8.2 การใช้โปรแกรม Microsoft Access

1. คลิกปุ่ม เริ่ม-Start
2. เลื่อน Cursor มาที่ไปที่ Program
3. คลิกที่รายการเมนู Microsoft Access



รูปที่ 2.1 หน้าต่างแรกของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าจอเริ่มแรกของโปรแกรม จะเป็นหน้าจอหลักในการใช้งาน โปรแกรม Microsoft Access เป็นโปรแกรมจัดการฐานข้อมูลที่จะช่วยอำนวยความสะดวก ดังต่อไปนี้

1. ช่วยจัดเก็บข้อมูลได้มากโดยไม่จำกัด
  2. ช่วยจัดการกับข้อมูลในรูปแบบที่ง่ายต่อการทำงาน
  3. ช่วยรับข้อมูลตามเงื่อนไขที่ต้องการ
  4. ช่วยสร้างฟอร์มเพื่อให้รับข้อมูลได้ง่ายกว่า
  5. ช่วยสร้างรายงานที่สื่อความหมายได้ง่าย และประกอบไปด้วยข้อมูลที่หลากหลาย เช่น ข้อความ กราฟิก และ อื่นๆ
  6. ช่วยให้นำข้อมูลจากโปรแกรมไปใช้งานในโปรแกรมอื่นๆ เช่น บนเว็บ เป็นต้น
- ผู้ใช้โปรแกรม Microsoft Access จึงจำเป็นต้องรู้เกี่ยวกับฐานข้อมูล วิธีการดำเนินการสร้าง ในโปรแกรมจะมีตัวช่วยทำงาน และอธิบายในขั้นตอนที่ผู้ใช้ไม่คุ้นต่อการใช้งานในเรื่องนั้นๆ ได้

### 2.8.3 การสร้างฐานข้อมูลใหม่

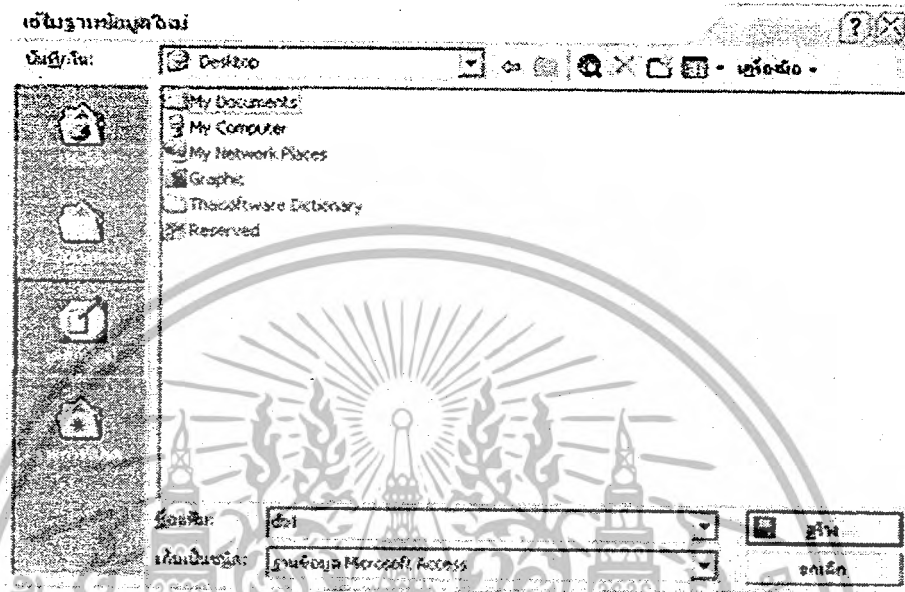
เปิดโปรแกรม MS Access ขึ้นมาเลือกสร้างฐานข้อมูลเปล่า Blank Access Database แล้วกด ตกลง (OK) ปรากฏภาพให้ตั้งชื่อ



รูปที่ 2.16 หน้าต่างการสร้างฐานข้อมูลใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดไคเร็คทอรีที่ต้องการเก็บฐานข้อมูลในช่อง Save in และตั้งชื่อฐานข้อมูลในช่อง File Name

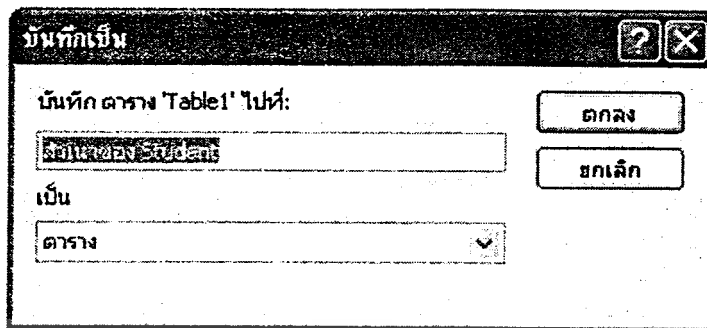


รูปที่ 2.17 เลือกที่เก็บไฟล์ข้อมูล

#### 2.8.4 การบันทึกฐานข้อมูล

ปกติเมื่อสร้างฐานข้อมูลขึ้นมา มีการสร้างแฟ้มต่างๆ ขึ้นมาภายในฐานข้อมูล เช่น ตารางฟอร์ม แบบสอบถาม และมีข้อความเตือนให้ทำการบันทึกแฟ้มเหล่านั้นไว้เสมอ เมื่อมีการป้อนหรือลบข้อมูลในแต่ละแฟ้ม ข้อมูลเหล่านั้นจะถูกบันทึกไว้อย่างอัตโนมัติ เมื่อเลิกใช้โปรแกรมจึงไม่ต้องบันทึกแฟ้มอีก นอกจากต้องการบันทึกเป็นแฟ้มชื่ออื่นหรือส่งแฟ้มในฐานข้อมูลเดิม ไปเก็บในฐานข้อมูลอื่นที่มีอยู่แล้ว

- บันทึกเป็น สำหรับบันทึกแฟ้มหรือตารางข้อมูลเดิม ไปเป็นชื่อใหม่ หรือทำสำเนาแฟ้มหรือตารางเอาไว้ป้องกันการเสียหาย โดยการคลิกที่ชื่อแฟ้ม ในตัวอย่างเป็นการเลือกตารางชื่อ Student แล้วคลิกคำสั่งแฟ้ม บันทึกเป็น

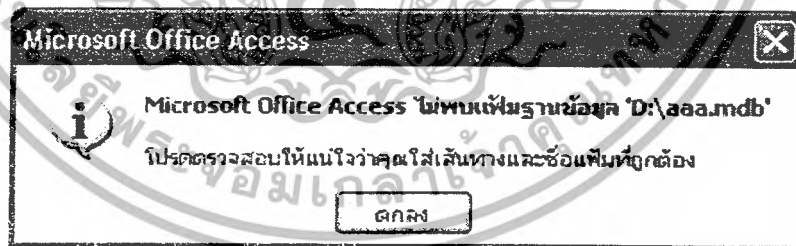


รูปที่ 2.18 หน้าต่างบันทึก

มีข้อความในกรอบ บันทึกตาราง 'Student' ไปที่และตั้งชื่อให้เป็น ลำเนาของ Student เป็นเพิ่มชนิดตาราง ถ้าต้องการเปลี่ยนชื่อให้พิมพ์ชื่อใหม่ลงไปแล้วคลิกที่ปุ่ม ตกลง ทำให้ได้ตารางชื่อใหม่ที่มีข้อมูลเหมือนเพิ่มเติมในฐานข้อมูลเดียวกันนี้

#### 2.8.5 ส่งออก

เป็นการส่งเพิ่มในฐานข้อมูลที่กำลังใช้งานอยู่ไปเก็บหรือเพิ่มในฐานข้อมูลใหม่ เมื่อเลือกจะมีกรอบให้เลือกชื่อฐานข้อมูลที่มีอยู่แล้ว ถ้าไม่มีฐานข้อมูลเดิมอยู่จะมีข้อความแจ้งว่าไม่พบฐานข้อมูล ดังรูปตัวอย่างต่อไปนี้



รูปที่ 2.19 ตรวจสอบสถานะชื่อเพิ่มข้อมูล

จากตัวอย่างเป็นการเลือกส่งตารางข้อมูลชื่อ Student ไปเก็บในแฟ้มชื่อ aaa ที่ไม่มีไอซีเก็บ เมื่อคลิกที่ปุ่ม ส่งออก จึงมีข้อความแจ้งว่าไม่พบเพิ่มฐานข้อมูลส่งออกไม่ได้ ใ้คลิกที่ปุ่มตกลง

### 2.8.6 การเปิดฐานข้อมูลเดิม

ในคำสั่ง แฟ้ม มีคำสั่งเปิด เพื่อเปิดฐานข้อมูลที่บันทึกไว้เป็นแฟ้มของไมโครซอฟต์เอกเซล หรือเปิดแฟ้มชนิดอื่นที่ไม่ได้สร้างด้วยไมโครซอฟต์เอกเซล เช่น แฟ้มข้อมูลชนิด .DBF ที่เป็นแฟ้มข้อมูลรุ่นเก่า แฟ้มชนิดข้อความที่สร้างเป็นตารางในไมโครซอฟต์เวิร์ดหรือเอกเซล เป็นต้น ส่วนอีกคำสั่งหนึ่งคือ รับข้อมูลภายนอก เป็นคำสั่งรับแฟ้มข้อมูลจากฐานข้อมูลอื่นเข้ามาในฐานข้อมูลที่กำลังใช้งานอยู่

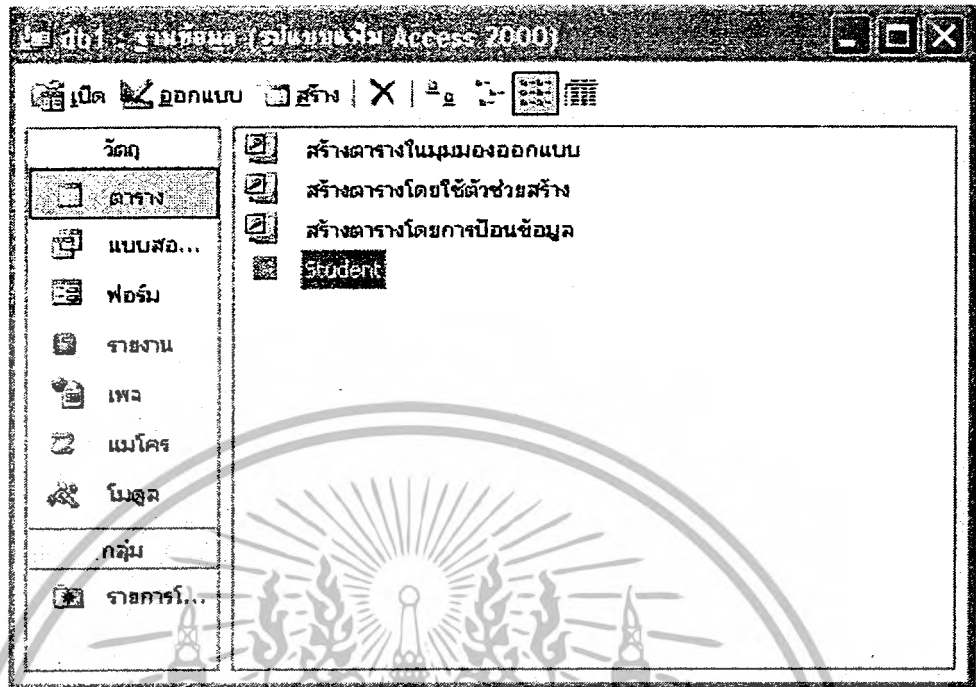
แฟ้ม (File) เปิด (Open) แล้วคำสั่งเปิดแฟ้มชนิด .MDB ของไมโครซอฟต์เอกเซลที่เคยบันทึกไว้แล้วมาแก้ไขหรือใช้งาน การเปิดฐานข้อมูลแต่ละครั้งจะเปิดได้เพียงฐานข้อมูลเดียว ถ้าเปิดซ้อนกันหลายครั้งแฟ้มฐานข้อมูลเดิมจะถูกปิดโดยอัตโนมัติ ดังนั้นถ้าต้องการถ่ายโอนข้อมูลจากตารางหรือข้อมูลอื่นให้ใช้วิธีรับข้อมูลภายนอกและเชื่อมโยงตารางแทน

การรับข้อมูลภายนอก คำสั่งนี้ประกอบด้วยคำสั่งย่อยสองคำสั่งคือ นำเข้า และเชื่อมโยงตาราง

การนำข้อมูลเข้า เป็นการรับแฟ้มข้อมูลจากฐานข้อมูลอื่นมาบรรจุในฐานข้อมูลที่กำลังใช้งาน เช่น ต้องการนำตารางข้อมูลชื่อ Student จากฐานข้อมูล Sample1 มาใส่ในฐานข้อมูล Sample2 ที่สร้างขึ้นใหม่ เป็นต้น

เมื่อใช้คำสั่งแฟ้ม รับข้อมูลภายนอกแล้วคลิกที่ฐานข้อมูลชื่อ Sample1 ระบบจะตรวจและอ่านชื่อแฟ้มทั้งหมดที่มีในฐานข้อมูล เช่น พบตารางชื่อ Student หนึ่งตารางจะแสดงชื่อไว้ในกรอบ นอกจากนี้ยังนำเข้าแฟ้มชนิดอื่นๆ ได้ทุกชนิด โดยคลิกที่ชื่อตารางแล้วคลิกปุ่มตกลง ถ้ามีหลายตารางคลิกที่ปุ่มเลือกทั้งหมดก็ได้

เชื่อมโยงตาราง เป็นการเชื่อมโยงฐานข้อมูลที่เปิดอยู่กับตารางข้อมูลในฐานข้อมูลอื่น โดยชื่อตารางข้อมูลที่เลือกไว้แสดงในฐานข้อมูลที่เปิดอยู่มีเครื่องหมายลูกศรชี้หน้าตารางข้อมูล เพื่อแจ้งให้ทราบว่าเป็นตารางที่เชื่อมโยงกับฐานข้อมูลอื่นตารางจริงยังคงอยู่ที่ฐานข้อมูลเดิม เมื่อมีการปรับปรุงแก้ไขข้อมูลในตารางปัจจุบันข้อมูลในตารางที่เชื่อมโยงไปจะถูกปรับปรุงตามไปด้วย ถ้าต้องการให้เป็นตารางจริงในฐานข้อมูลนี้ต้องใช้คำสั่งนำเข้า



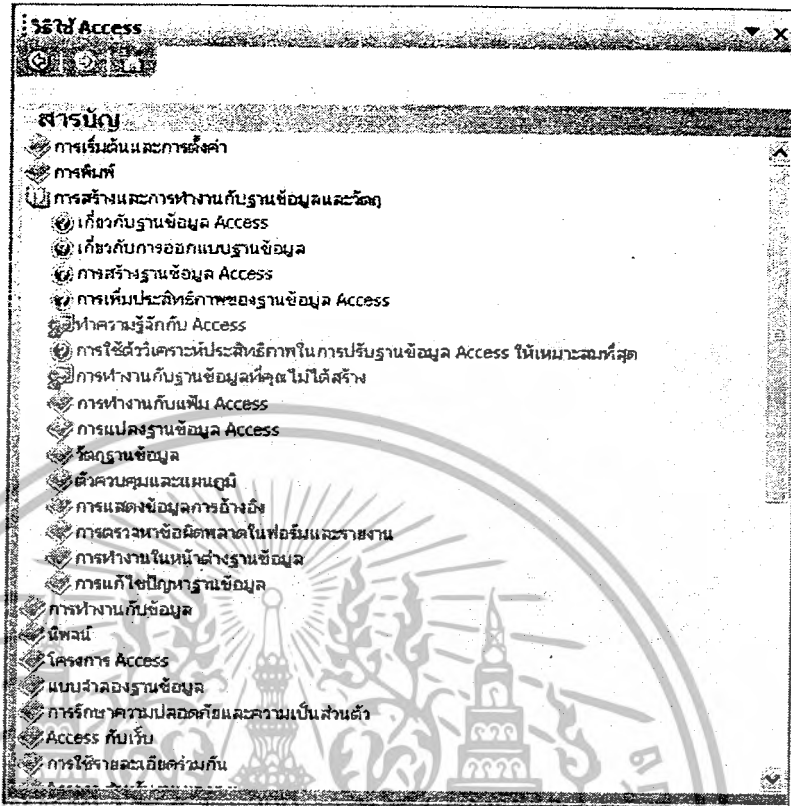
รูปที่ 2.20 การเชื่อมโยงตาราง

### 2.8.7 การออกจากโปรแกรม

ดังได้กล่าวไว้แล้วว่าการปรับปรุงข้อมูลในตารางใดๆ ข้อมูลในตารางจะถูกจัดเก็บให้อย่างอัตโนมัติโดยไม่ต้องตั้งบันทึกอีก เมื่อต้องการออกจากโปรแกรมจึงสามารถคลิกที่ปุ่มปิดหรือคลิกคำสั่งเพิ่มจบการทำงานได้ทันที

### 2.8.9 การเรียกดูวิธีใช้

ในกรณีที่ไม่เข้าใจการใช้งานเพิ่มหรือคำสั่งต่างๆ สามารถเรียกดูความช่วยเหลือ โดยกดเป็นฟังก์ชัน F1 หรือคลิกที่ วิธีใช้ ในแถบเมนูในขณะที่ยังไม่ได้เปิดหรือสร้างเพิ่มใดๆจะมีกรอบให้เลือกรูปร่างอธิบายวิธีใช้ต่างๆ ดังรูป



รูปที่ 2.21 หน้าต่างวิธีการเรียกใช้

## 2.9 Flash Magic

Flash Magic เป็นโปรแกรมสำหรับใช้ Download HEX File ให้กับ หน่วยความจำ โปรแกรมภายในตัวของไมโครคอนโทรลเลอร์ ตระกูล MCS51 ของ Philips ในกลุ่มเบอร์ที่รองรับ การ Download ข้อมูลด้วยวิธีการแบบ ISP Download (In System Programming Download) ซึ่งได้ รับการพัฒนาขึ้นโดย Embedded System Academy ซึ่งได้รับการสนับสนุนจาก Philips โดยผู้ใ้ สามารถทำการ Download โปรแกรมตัวนี้มาใช้งานได้ฟรีโดยไม่เสียค่าใช้จ่ายใดๆ จากเว็บไซต์ของ WWW.ESACADEMY.COM โดยในปัจจุบัน (มีนาคม 2547) โปรแกรม Flash Magic จะเป็นรุ่น 1.87 ซึ่งในรุ่นนี้จะสนับสนุนการใช้งานร่วมกับไมโครคอนโทรลเลอร์ของ Philips ได้หลายเบอร์ เช่น

- 89C51RX2 เช่น 89C51RA2XX,89C51R32XX,89C51RC2XX,89C51RD2XX
- 89C60X2,89C61X2

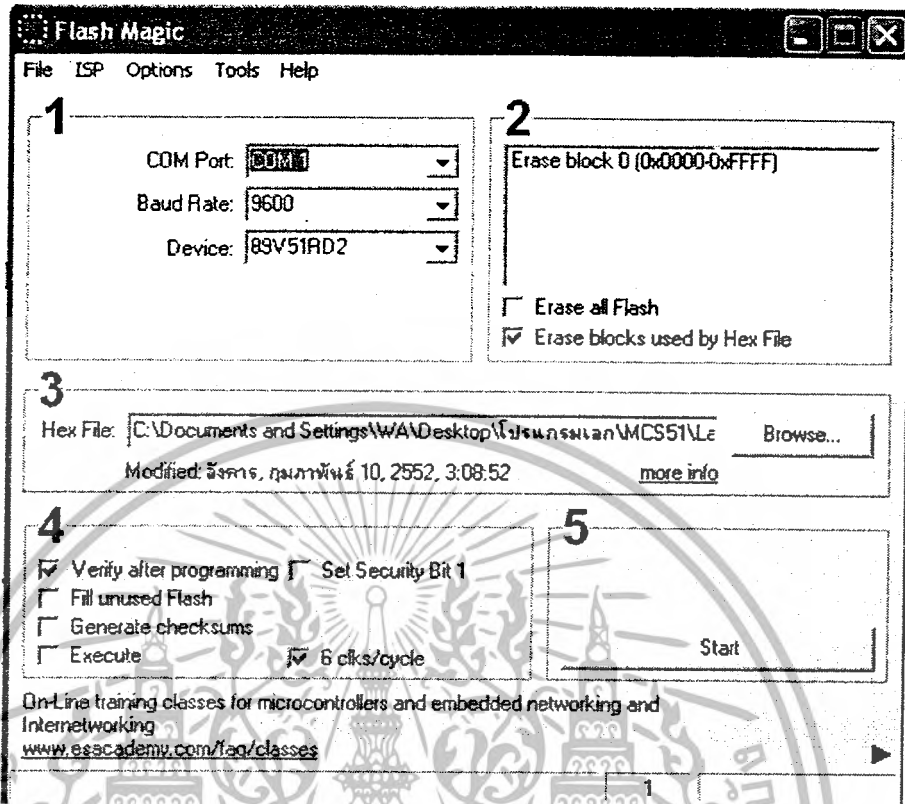
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 89C51RX2H เช่น 89C51RB2HXX,89C51RC2HXX,89C51RD2HXX
- 89C66X เช่น 89C660,89C662,89C664,89C668,89C6699
- 89C51RX+ เช่น 89C51RB+,89C51RC+,89C51RD+
- XA-G39,XA-G49
- 89LPC9XX เช่น 89LPC901,02,03,06,07,08,12,13,14,20,21,22,30,31,32,35
- 89LV51RD2
- 89V51RD2

โดยการทำงานของ โปรแกรม Flash Magic นั้นจะสนับสนุนการ Download โปรแกรมแบบ HEX File จากคอมพิวเตอร์ PC ผ่านทางพอร์ตสื่อสารอนุกรมแบบ RS232 ไปให้กับหน่วยความจำ โปรแกรม Flash Memory ภายในตัว CPU ตระกูล MCS51 เฉพาะที่ผลิตขึ้นโดย Philips เท่านั้น โดยการ Download สามารถทำได้ทั้งแบบ Manual และ อัตโนมัติ (Auto Download) ซึ่งในกรณีของการ Download แบบอัตโนมัตินั้น จะต้องออกแบบวงจรสำหรับเลือกโหมดการทำงานของ CPU โดยใช้ สัญญาณ RTS และ DTR ของพอร์ตสื่อสารอนุกรม RS232 ได้ด้วย โดยจะใช้สัญญาณ DTR สำหรับ ทำหน้าที่ควบคุมการทำงานของสัญญาณ Reset ของ CPU และใช้สัญญาณ RTS สำหรับกำหนดสถานะโลจิกให้กับขาสัญญาณ PSEN ของ CPU

### 2.9.1 การใช้งาน Flash Magic

สำหรับการเรียกใช้งานโปรแกรม Flash Magic นั้นสามารถเรียกจาก Icon ของโปรแกรม หรือจะเรียกผ่านเมนูคำสั่งของ Windows ก็ได้ตามต้องการ โดยลักษณะของโปรแกรมเป็นดังรูป



รูปที่ 2.22 ลักษณะ โปรแกรม Flash Magic

โดยในการใช้งานโปรแกรมนั้นจะต้องทำการกำหนดค่าพารามิเตอร์ต่างๆให้กับโปรแกรมให้ถูกต้องเสียก่อน โดย Flash Magic จะแบ่งหัวข้อสำหรับกำหนดค่าพารามิเตอร์ให้กับโปรแกรมไว้ทั้งหมด 5 ส่วนด้วยกันคือ

1. ใช้สำหรับเลือกกำหนดการเชื่อมต่อระหว่างพอร์ตสื่อสารอนุกรมของคอมพิวเตอร์ PC โดยมีค่าให้กำหนดทั้งหมด 4 ส่วน คือ

- **Com Port** ใช้สำหรับเลือกกำหนดหมายเลข Comport ของ คอมพิวเตอร์ PC ที่นำมาเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ โดยต้องเลือกตามความเป็นจริง
- **Baud rate** ใช้สำหรับเลือกกำหนดค่าความเร็วของอัตรา Baud rate ที่จะใช้ในการสื่อสารระหว่างคอมพิวเตอร์ PC กับไมโครคอนโทรลเลอร์ในบอร์ด ซึ่งต้องเลือกค่าที่ CPU สามารถรองรับได้ด้วย โดยตามปกติจะกำหนดเป็น 9600 แต่ถ้าใช้ค่าความถี่ที่เป็นค่ามาตรฐาน เช่น 11.0592MHz หรือ 18.432MHz อาจสามารถกำหนดค่าอัตราของความเร็ว Baud rate สูงสุดได้ถึง 115200
- **Device** ใช้สำหรับเลือกกำหนดความเร็วของ CPU ในบอร์ดที่จะทำการ Download

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Oscillator Freq.** ใช้สำหรับกำหนดค่าความถี่ของ Oscillator หรือ X-TAL ที่ใช้กับ CPU ในบอร์ดไมโครคอนโทรลเลอร์ โดยต้องกำหนดตามความเป็นจริง โดยกำหนดค่าเป็นหน่วยแบบ MHz

2. ใช้สำหรับกำหนดรูปแบบการลบข้อมูลในหน่วยความจำ Flash ของ CPU ก่อนที่จะทำการ Download ข้อมูลใหม่ให้กับหน่วยความจำของ Flash Memory ของ CPU โดยสามารถกำหนดรูปแบบการล้างข้อมูลได้ 2 แบบด้วยกันคือ

- **Erase all Flash + Security** ใช้สำหรับเลือกรูปแบบการลบข้อมูล โดยสั่งให้ลบข้อมูลทั้งหมดในตัว CPU พร้อมกับค่าใน Security Bit ด้วย
- **Erase Block Used By HEX File** ใช้สำหรับสั่งลบข้อมูลแบบ Block ซึ่งโปรแกรมจะทำการตรวจสอบตำแหน่งของหน่วยความจำที่ต้อง Download จาก Hex File เพื่อสั่งลบข้อมูลเฉพาะ Block ที่มีตำแหน่งตรงกับข้อมูลที่ระบุไว้ใน HEX File เท่านั้น ซึ่งการสั่งลบแบบนี้จะใช้เวลาน้อยกว่า แต่ไม่สามารถใช้กับ CPU ที่มีการสั่งกำหนด Security ไว้แล้วได้ ซึ่งถ้ามีการสั่ง Security ไว้แล้ว ต้องเลือกการลบเป็นแบบ Erase all Flash เท่านั้น ซึ่งในขั้นตอนของการพัฒนาโปรแกรมนั้นขอแนะนำให้ยังไม่ต้องการสั่ง Security เพื่อจะได้สั่งลบข้อมูลเป็นแบบ Block ได้

3. ใช้สำหรับกำหนดชื่อและตำแหน่งที่อยู่ของ Hex File ที่จะใช้ในการ Download ให้กับหน่วยความจำของ CPU โดยสามารถเลือกกำหนดได้ 2 วิธีคือ การพิมพ์ตำแหน่งของ Directory และชื่อของ Hex File ในช่องรับข้อมูลเอง หรือเลือกจากปุ่มคำสั่ง Browse เพื่อค้นหาชื่อและตำแหน่งไฟล์เองก็ได้

4. ใช้สำหรับกำหนดเงื่อนไขพิเศษให้กับโปรแกรม โดยสามารถเลือกกำหนดหัวข้อที่ต้องการได้เอง โดยต้องการเลือกหัวข้อใดก็ให้เลือกเครื่องหมาย (✓) ที่หัวข้อนั้นๆตามต้องการ

- **Verify After Program** ใช้สำหรับสั่งให้โปรแกรมตรวจสอบความถูกต้องของข้อมูลที่เขียนให้กับหน่วยความจำ Flash ด้วยหลังจากสั่ง Download หรือ โปรแกรม ขึ้นมาเรียบร้อยแล้ว

- **Fill unused Flash** ใช้สำหรับสั่งให้โปรแกรม Flash Magic พิมพ์ข้อมูลใน Buffer ตำแหน่งแอดเดรสที่ไม่มีข้อมูลจาก HEX File ให้ด้วยค่า 00H ทั้งหมด ซึ่งไม่มีการเลือกรหัสข้อนี้เพราะจะทำให้การทำงานของโปรแกรมช้ามากเนื่องจากโปรแกรม Flash จะทำการสั่ง Download ข้อมูลให้กับ CPU ตลอดช่วงตำแหน่งแอดเดรสของหน่วยความจำทั้งหมดที่อยู่ แต่ถ้าไม่เลือกหัวข้อนี้ โปรแกรม Flash Magic จะสั่ง Download เฉพาะข้อมูลที่มีอยู่ใน Hex file เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Generate Checksum** ใช้สำหรับสั่งให้โปรแกรมทำการคำนวณค่า Checksum ของข้อมูลใน Hex File ให้ด้วย เพื่อใช้ตรวจสอบความถูกต้องของข้อมูล
- **Execute** ใช้สำหรับกำหนดให้โปรแกรม Flash Magic สั่งให้ CPU เริ่มต้นทำงานตามคำสั่งที่อยู่ในโปรแกรมที่ Download ให้ทันที หลังการสั่ง Download ข้อมูลเสร็จเรียบร้อยแล้ว
- **Set Security Bit** ใช้สำหรับสั่งให้โปรแกรม Flash Magic ทำการสั่ง Set Security Bit เพื่อป้องกันการอ่านข้อมูลจากหน่วยความจำ Flash ของ CPU ให้ด้วยหลังจากสั่ง Download ข้อมูลเสร็จเรียบร้อยแล้ว ซึ่งถ้าเลือกค่าตัวเลือกนี้ไว้ ต้องกำหนดเงื่อนไขในการสั่งลบ (หัวข้อที่2) ให้เป็นแบบ Erase All Flash + Security ด้วยในการสั่ง Download ครั้งต่อไป

5. ใช้สำหรับสั่งให้โปรแกรม Flash Magic เริ่มต้นทำการ Download HEX ไฟล์ให้กับ CPU โดยการทำงานของคำสั่งนี้จะมีลำดับขั้นตอนดังนี้

- **Erase Flash** ตามเงื่อนไขที่กำหนดไว้ในข้อ 2
- **Program** ข้อมูลให้กับหน่วยความจำ Flash ของ CPU
- **Verify** ถ้าเลือกหัวข้อการทำงานหัวข้อนี้ไว้
- **Fill unused Flash** ถ้าเลือกการทำงานหัวข้อนี้ไว้
- **Generate Checksum** ถ้าเลือกหัวข้อการทำงานหัวข้อนี้ไว้
- **Program Lock bit** ถ้าเลือกหัวข้อการทำงานหัวข้อนี้ไว้
- **Program Security bit** ถ้าเลือกหัวข้อการทำงานหัวข้อนี้ไว้
- **Execute Firmware** ถ้าเลือกหัวข้อการทำงานหัวข้อนี้ไว้

## 2.10 MCS-51 AT89C4051 ไอซีไมโครคอนโทรลเลอร์ MCS-51

จากความก้าวหน้าของเทคโนโลยี พบว่าอุปกรณ์บางส่วนที่นำมาควบคุมเครื่องใช้ไฟฟ้า ระบบควบคุมทางอิเล็กทรอนิกส์ หรือ โครงการในวารสารต่าง ๆ จะนำเสนอในรูปแบบของไอซีไมโครคอนโทรลเลอร์ ซึ่งเทคโนโลยีดังกล่าวมีบริษัทต่างๆ ที่ผลิตออกมาแข่งขันด้านคุณสมบัติ และการใช้งานกันในตลาด อาทิเช่น Intel, Atmel, Microchip, Motorola ส่วนทางด้านการศึกษาและการเรียนรู้ นั้น มีการพัฒนาทางด้านการเรียนการสอนวิชาไมโครคอนโทรลเลอร์ โดยบริษัทต่างๆ แข่งขันกันเพื่อสร้างชุดฝึกตอบสนองต่อความต้องการเรียนรู้ ความสะดวกในการสร้างชุดทดลองของนักศึกษาและผู้สนใจ ในขณะเดียวกันเป็นผู้ตามเพื่อศึกษาเทคโนโลยีที่ทิ้งห่างทุกขณะ จึงต้องใช้ชุดทดลองที่ถูกกำหนดรูปแบบและราคา ทั้งที่อุปกรณ์ที่เป็นอุปกรณ์ร่วมกับเทคโนโลยีบางอย่างกลับทิ้งไปและไม่ได้นำมาใช้งาน ยกตัวอย่าง เช่น Main Board ,CD-ROM หรือ Hard disk ที่ใช้งานไม่ได้แล้ว นำอุปกรณ์ภายในมาใช้งานได้ก็ขึ้นบ้าง ตลอดจนตัวต้านทานและตัวเก็บประจุ ที่เป็นอุปกรณ์ติดบนผิวหน้าแบบ SMD (Surface mounted device) หน่วยความจำชนิดแรม และรวม แผ่นระบายความร้อนสเตปปีงมอเตอร์ (Stepping Motor) คอนเนคเตอร์ (connector) ฯลฯ ซึ่งพร้อมที่จะใช้งาน มากกว่าที่จะได้เพิ่มมาจากการซื้อชุดทดลองใหม่

ดังนั้น ไมโครคอนโทรลเลอร์จึงถูกเลือกใช้ในการทดลองด้วยเหตุผลในการใช้อุปกรณ์ที่ต่อรวมในวงจรน้อยชิ้น ราคาถูก หาแหล่งข้อมูลได้ง่ายและมีการพัฒนาประสิทธิภาพ ในการประมวลผล เทียบเท่าไมโครโพรเซสเซอร์ขนาด 8-16บิต และใช้ในการพัฒนากับงานที่ไม่ต้องการความซับซ้อนซึ่งจะทำให้มีความสะดวกมากขึ้นปัจจุบันมีไอซีของสองบริษัทที่เรานิยมใช้งาน ออกแบบ และสำหรับในการเรียนการสอนในสถาบันหลายแห่ง คือ ไอซีไมโครคอนโทรลเลอร์ของบริษัท Microchip ตระกูล 16FXXX และไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 จากหลายบริษัทที่ผลิตออกมามากมายหลายเบอร์ แต่คำสั่งจะใกล้เคียงกัน ดังนั้น การทดลองในโครงการนี้จึงเลือกใช้งานไอซี ไมโครคอนโทรลเลอร์ของตระกูล MCS-51 ด้วยเหตุผลเพราะมีข้อมูล และตัวอย่างที่หลากหลายสามารถใช้งานได้ง่าย แต่ก็ไม่ได้ตัดสินใจดีกว่าไอซีในตระกูลอื่นๆ ดังนั้น ในการออกแบบชุดทดลองที่ใช้งานร่วมกับบอร์ดไมโครคอนโทรลเลอร์ของ MCS-51 ที่จะแนะนำให้สร้างขึ้น สามารถที่จะนำมาใช้งานร่วมกับไอซีไมโครคอนโทรลเลอร์ได้ทุกเบอร์ และทุกตระกูล

### 2.10.1 การจัดหน่วยความจำของไอซีไมโครคอนโทรลเลอร์

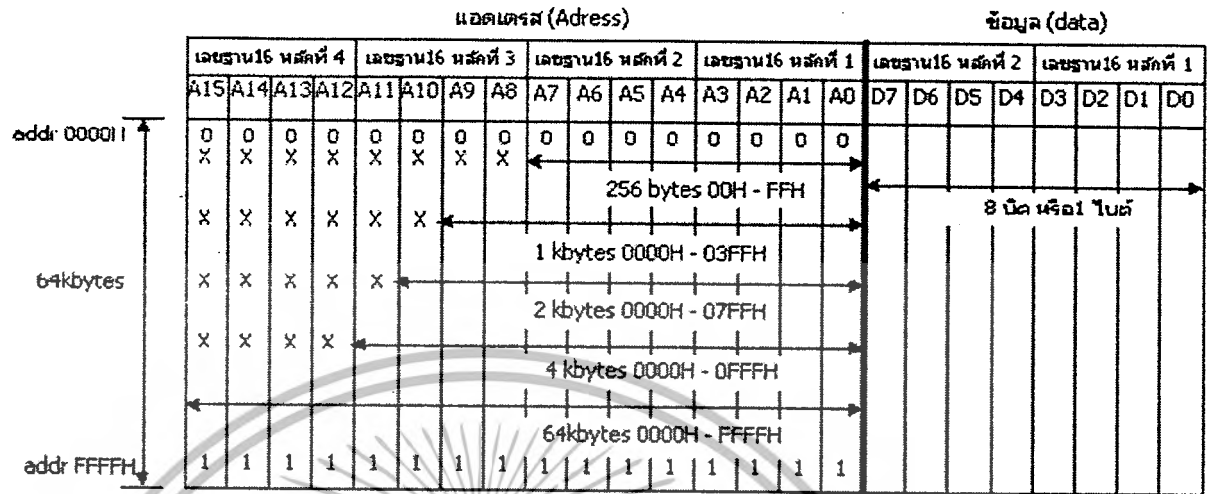
หน้าที่การทำงานของหน่วยความจำจะทำหน้าที่เก็บ โปรแกรมคำสั่ง และข้อมูลที่จะใช้ในการกำหนดค่าต่างๆ ให้กับไมโครคอนโทรลเลอร์หรือใช้เก็บค่าต่างๆ ที่ไมโครคอนโทรลเลอร์กระทำตามคำสั่ง การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 สามารถแบ่งหน่วยความจำออกเป็น 3 กลุ่มคือ

1. หน่วยความจำสำหรับเก็บ โปรแกรม (Program Memory) หรือ (Code Memory)
2. หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)
3. รีจิสเตอร์ที่ทำหน้าที่เฉพาะ (Special Function

#### การจัดหน่วยความจำของไอซีไมโครคอนโทรลเลอร์

##### 1 หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory) หรือหน่วยความจำรหัสคำสั่ง (Code Memory)

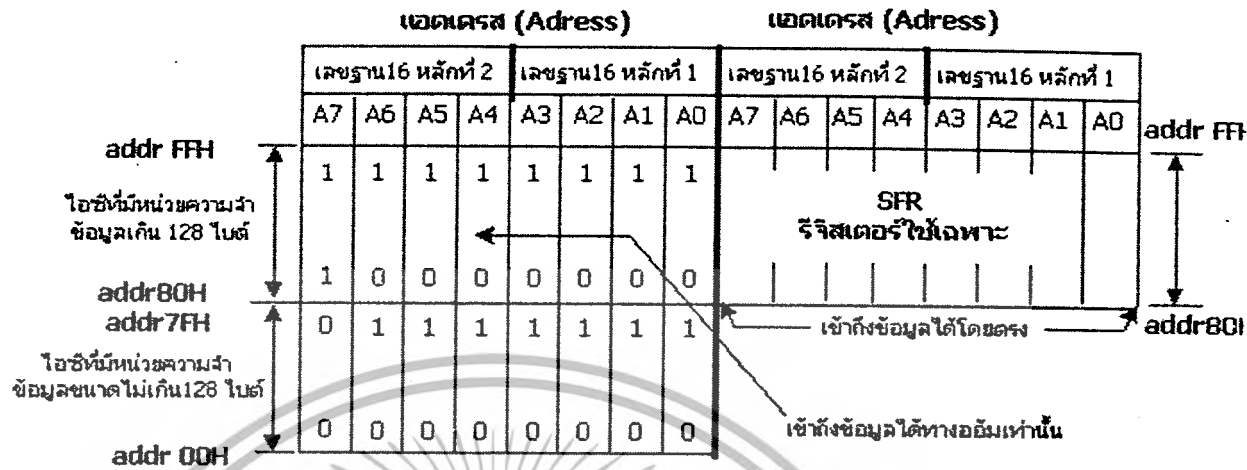
หน่วยความจำสำหรับเก็บ โปรแกรม(ทำหน้าที่เช่นเดียวกับรอม) หรือหน่วยความจำรหัสคำสั่ง (Code Memory) จะทำหน้าที่เก็บชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ปฏิบัติตามคำสั่งนั้นๆ ยกตัวอย่าง ในขณะที่เปิดเครื่องไมโครเวฟ จะมีการแสดงผลรายการหลักที่หน้าจอ LCD เพื่อคอยให้ป้อนค่าเวลาที่ต้องการจะอุ่นอาหาร คำสั่งที่จ่อ LCD เพื่อให้ป้อนข้อมูลนั้นจะเขียนคำสั่งอยู่ในส่วนของหน่วยความจำโปรแกรม ถึงจะเปิดเครื่องไมโครเวฟกี่ครั้ง การแสดงผลที่ LCD ให้ป้อนค่าเวลาที่ต้องการเหมือนเดิม ภายในตัวไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C1051AT89C2051 และ AT89C4051 จะมีหน่วยความจำที่เก็บโปรแกรมได้ 1 Kbytes, 2 Kbytes และ 4 Kbytes ตามลำดับ หน่วยความจำจะเป็นลักษณะแบบแฟลช ที่มีคุณสมบัติในการใช้งานโดยสามารถจะทำการลบข้อมูลด้วยไฟฟ้า และเก็บข้อมูลเข้าเก็บไว้ในตัวไมโครคอนโทรลเลอร์ ได้กว่า 1000 ครั้ง โดยใช้เครื่องโปรแกรมที่ไม่ยุ่งยากและราคาไม่แพง (สามารถรักษาข้อมูลไว้ได้นานหลายปี) ข้อสังเกต ส่วนของแอดเดรส(ADDRESS) ไม่สามารถที่จะใช้ตำแหน่งเดียวกันได้ แต่ข้อมูล(DATA)สามารถที่จะมีข้อมูลเหมือนกันได้ จากตารางเหมือนกับมีกระดาดจำนวนเท่ากับ  $2^n$  บรรทัดมาให้ ดังนั้นหากต้องการเขียนข้อมูลใด ลงในแต่ละบรรทัด จะต้องมิตำแหน่งของบรรทัดที่ไม่ซ้ำกัน และการที่จะเลือกจำนวนของบรรทัด ขึ้นอยู่กับปริมาณของข้อมูลที่ต้องการจะเขียน ยกตัวอย่างเช่นหากต้องการขนาดของข้อมูลในกาเขียนโปรแกรมเพียง 2,000 บรรทัด อาจจะใช้หน่วยความจำขนาด 2Kbytes โดยมีแอดเดรสตั้งแต่ 0000H - 07FFH



รูปที่ 2.23 หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory)

### 2 หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

หน่วยความจำข้อมูล(RAM) จะทำหน้าที่เก็บรักษาข้อมูล โดยข้อมูลอาจจะเป็นค่าหลังจากไมโครคอนโทรลเลอร์ ทำการการประมวลผล หรือเก็บค่าข้อมูลที่จะให้กับไมโครคอนโทรลเลอร์ประมวลผลในขณะนั้น และจะทำหน้าที่เป็น สแตก (Stack) บางส่วน (ส่วนของสแตกจะอธิบายในลำดับต่อไป) ยกตัวอย่าง เช่น ถ้าเป็นเครื่องไมโครเวฟที่ใช้สำหรับอุ่นอาหาร ก็คือส่วนที่เราป้อนข้อมูลเช่นเวลา หรืออุณหภูมิที่เป็นปัจจุบัน หลังจากหน่วยความจำโปรแกรมแสดงรายการ หลักที่ LCD นั้นเอง ตั้งเกตุว่าหากเราปิดเครื่อง แล้วเปิดเครื่องใหม่อีกครั้งหนึ่ง ค่าข้อมูลที่เป็นเวลา และอุณหภูมิเดิมที่เรากำหนดไว้ในครั้งแรกก็จะหายไป และจะให้เราป้อนค่าข้อมูลใหม่อีกครั้ง ดังนั้นการที่จะรักษาข้อมูลเดิมไว้ได้ จะต้องมีแหล่งจ่ายไฟสำรองไว้สำหรับเพื่อเลี้ยงให้กับตัวไอซีตลอดเวลา หรือที่ เรียกว่า Battery backup ถ้าหรับไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C1051 จะมีหน่วยความจำที่เก็บข้อมูลได้ 64 bytes ส่วนAT89C2051 และ AT89C405; จะมีหน่วยความจำที่เก็บข้อมูลได้ 128 bytes



รูป 2.24 ตารางตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน (Internal data memory)

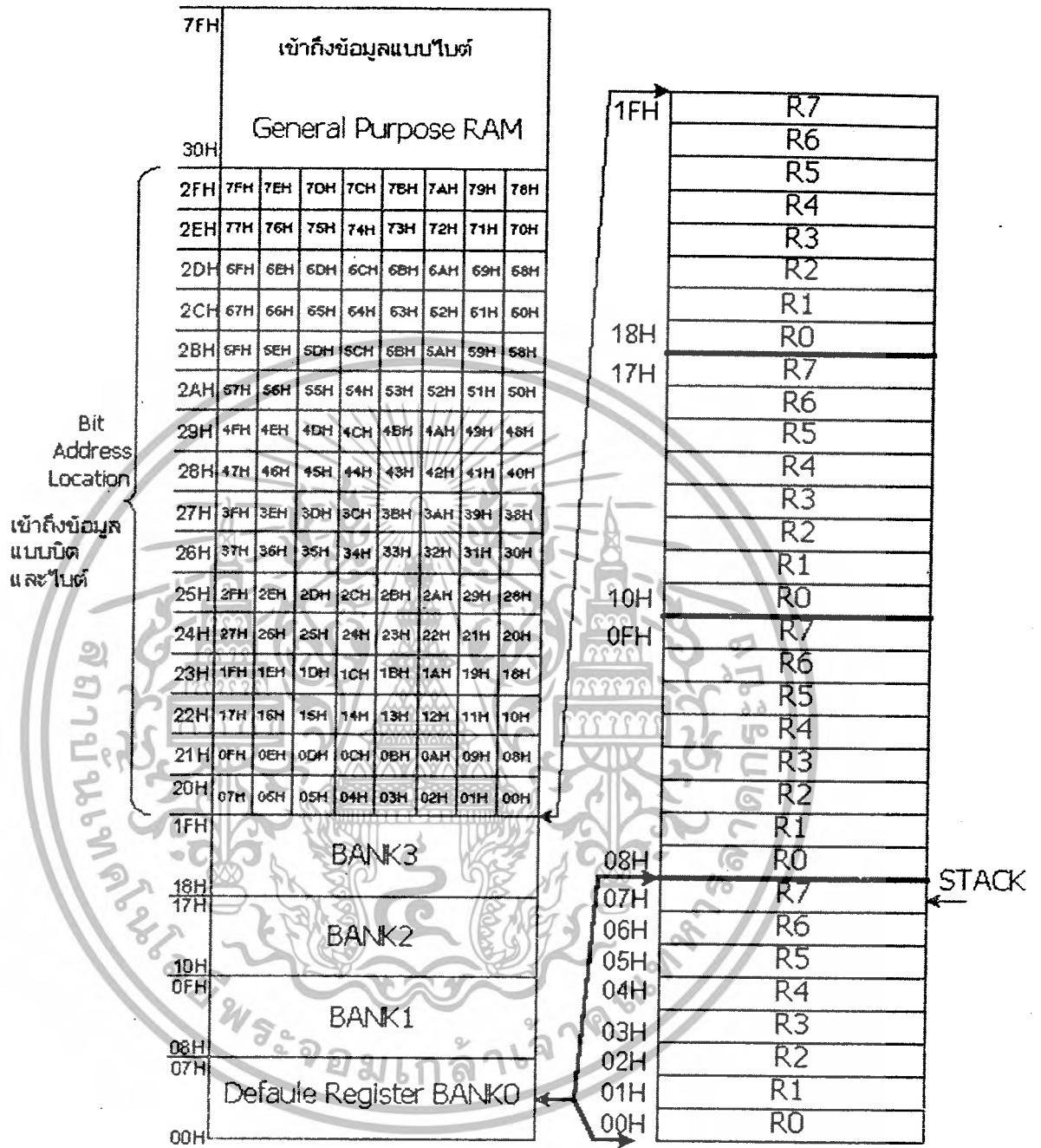
หน่วยความจำข้อมูลภายในยังแบ่งส่วนของการใช้งานได้อีกเป็นสองส่วนคือ หน่วยความจำข้อมูลภายใน 128 ไบต์จะเป็นหน่วยความจำที่ใช้งานทั่วไปอยู่ที่ตำแหน่งแอดเดรส 00H-7FH และหน่วยความจำในตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งจะเป็นส่วนของรีจิสเตอร์เฉพาะ (Special Function Register) ในส่วนของหน่วยความจำที่ใช้งานทั่วไป โดยพื้นที่ของหน่วยความจำข้อมูลภายในที่ตำแหน่งแอดเดรส 00H-7FH สามารถที่จะแบ่งออกเป็นส่วนย่อยได้ดังนี้

1. พื้นที่ในหน่วยความจำข้อมูล (แรม) ตำแหน่งที่ 00H-1FH จำนวน 32 ไบต์ จะถูกแบ่งออกเป็น 4 กลุ่ม เรียกว่า แบงก์ (Bank) และในแต่ละแบงก์ จะมี 8 ไบต์ พื้นที่ในแต่ละแบงก์จะถูกใช้งานเป็นรีจิสเตอร์ที่ใช้งานทั่วไป (รีจิสเตอร์ R0-R7 เป็นรีจิสเตอร์ที่มีขนาด 8 บิตหรือ 1 ไบต์) โดยที่รีจิสเตอร์ R0 จะอยู่ในตำแหน่งแรกของแต่ละแบงก์ และ รีจิสเตอร์ R7 จะอยู่ในตำแหน่งสุดท้ายของแต่ละแบงก์ ในการนำไปใช้งาน จะเลือกใช้รีจิสเตอร์ R0-R7 ได้เพียงแบงก์เดียว และเลือกใช้พื้นที่ของรีจิสเตอร์ R0-R7 ในแบงก์ใดๆก็ได้ โดยการกำหนดค่าข้อมูลที่รีจิสเตอร์ PSW ในส่วนของรีจิสเตอร์เฉพาะ (Special Function Register) หากไม่ได้กำหนดค่าใดๆเลย เมื่อทำการรีเซตให้กับไอซีไมโครคอนโทรลเลอร์ จะถูกกำหนด ให้เริ่มต้นใช้งานที่รีจิสเตอร์ R0-R7 ในหน่วยความจำตำแหน่งแบงก์ 0 ให้เอง ดังนั้นในการทดลองเริ่มต้นในส่วนแรกๆ เรา ยังไม่กำหนดค่าใดๆ ในการเลือกใช้รีจิสเตอร์แบงก์อื่นๆ (จะใช้เพียงรีจิสเตอร์ R0-R7 ในหน่วยความจำตำแหน่งแบงก์ 0 ที่ถูกกำหนดมาให้เท่านั้นก่อน)

2. พื้นที่ในหน่วยความจำข้อมูลภายใน(แรม) ตำแหน่งแอดเดรสที่ 20H-2FH จำนวน 16 ไบต์ เป็นส่วนที่สามารถใช้งานในลักษณะการเข้าข้อมูลแบบ ไบต์หรือแบบบิตได้ และสามารถอ้างตำแหน่งแบบบิตได้โดยตรง เพียงแต่ระบุตำแหน่งหรือชื่อของบิตนั้นๆ มีอยู่จำนวนทั้งหมด 128 บิต แต่ละบิตจะมีหมายเลขตำแหน่งของบิตคือ 00H-7FH โดยตำแหน่งบิตที่ 00H คือข้อมูลของบิตต่ำสุดในตำแหน่งแอดเดรสที่ 20H หรือ เรียกว่า(20H.1) และตำแหน่งของบิตที่ 7FH คือข้อมูลบิตสูงสุดในตำแหน่งแอดเดรสที่ 2FH หรือ เรียกว่า (20H.7) การอ้างตำแหน่งแบบบิตจะทำให้โปรแกรมทำงานได้รวดเร็วขึ้น

3. พื้นที่บริเวณหน่วยความจำข้อมูลในตำแหน่งที่ 30H-7FH เป็นพื้นที่ของหน่วยความจำใช้งานทั่วไป และการติดต่อกับข้อมูลในตำแหน่งต่างๆ ของหน่วยความจำส่วนนี้จะอ้างตำแหน่งข้อมูลได้ในลักษณะของแบบไบต์เท่านั้น และพื้นที่ส่วนนี้จะใช้เป็นสแตกได้ (รายละเอียดของสแตก อยู่ในเรื่องหน่วยความจำแบบ SFR)





รูปที่ 2.25 พื้นที่ในหน่วยความจำข้อมูล (RAM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.10.2 หน่วยความจำสำหรับเก็บค่ารีจิสเตอร์เฉพาะ (Special Function Register)

รีจิสเตอร์เฉพาะหรือ รีจิสเตอร์พิเศษ (Special Function Register) ที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ จะอยู่ในหน่วยความจำตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งสามารถจะเรียกใช้ชื่อของรีจิสเตอร์ได้โดยตรง หรืออาจจะเรียกชื่อตามตำแหน่งแอดเดรสก็ได้ รีจิสเตอร์เฉพาะจะประกอบด้วย

Byte Address	Bit Address								
FFH									
FOH	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	B
EOH	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	ACC
	CY	AC	F0	RS1	RS0	OV	F1	P	
DOH	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	PSW
B8H	BFH	BEH	BDH	BDH	B8H	BAH	B9H	B8H	IP
BOH	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	P3
	EA		ET2	ES	ET1	EX1	ET0	EX0	
A8H	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	IE
A0H	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	P2
99H	ไม่สามารถเข้าถึงได้ในระดับบิต								SBUF
	SM0	SM1	SM2	REN	TB8	TB8	T1	R1	
98H	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	SCON
90H	97H	96H	95H	94H	93H	92H	91H	90H	PI
8DH	ไม่สามารถเข้าถึงได้ในระดับบิต								TH1
8CH	ไม่สามารถเข้าถึงได้ในระดับบิต								TH0
8BH	ไม่สามารถเข้าถึงได้ในระดับบิต								TL1
8AH	ไม่สามารถเข้าถึงได้ในระดับบิต								TL0
89H	ไม่สามารถเข้าถึงได้ในระดับบิต								TMOD
88H	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	TCON
87H	ไม่สามารถเข้าถึงได้ในระดับบิต								PCON
83H	ไม่สามารถเข้าถึงได้ในระดับบิต								DPH
82H	ไม่สามารถเข้าถึงได้ในระดับบิต								DPL
81H	ไม่สามารถเข้าถึงได้ในระดับบิต								SP
80H	87H	86H	85H	84H	83H	82H	81H	80H	PC

Special Function Registers

### รูปที่ 2.26 การจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ (Special Function Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.26 ส่วนต่างๆของหน่วยความจำและตำแหน่งของรีจิสเตอร์ แบ่งออกเป็น

ACC (Accumulator) หรือ รีจิสเตอร์ A เป็นรีจิสเตอร์ขนาด 8 บิต และมีแอดเดรสอยู่ที่ตำแหน่ง EOH สามารถที่จะเข้าถึงข้อมูล ในระดับบิตได้ ใช้งานเป็นตัวกระทำร่วมทางคณิตศาสตร์ เช่น การบวก ลบ คูณ หาร เป็นต้น และทำหน้าที่เป็นตัวเก็บผลลัพธ์ที่ได้จากการคำนวณ ทางคณิตศาสตร์ ซึ่งรีจิสเตอร์นี้จะถูกใช้งานบ่อยมากในการเขียนโปรแกรม

รีจิสเตอร์ B เป็นรีจิสเตอร์ขนาด 8 บิตที่มีแอดเดรสอยู่ที่ตำแหน่ง FOH ใช้ในการกระทำในคำสั่งคูณหรือหารข้อมูล โดยใช้ร่วมกับรีจิสเตอร์ A จะทำหน้าที่เก็บค่าผลลัพธ์ที่เป็นเศษของการหาร และเก็บผลลัพธ์ของค่าผลคูณไปด้านบน และยังใช้เก็บข้อมูลทั่วไปได้ในกรณีไม่ได้ทำคำสั่งในการคูณหรือหาร

PSW: (Program Status Word) เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่แสดงสถานะ การทำงานของโปรแกรม ซึ่งจะมีการเปลี่ยนแปลง หลังจากมีการทำงานในคำสั่งต่างๆ และยังใช้เป็นตัวเลือกตำแหน่งแบงค์ของรีจิสเตอร์ (Register Bank) R0-R7 อีกด้วย ผลของบิตต่างๆ สามารถนำไปเป็นเงื่อนไขในการกระโดด (Jump) และค่าของบิตต่างๆ ใน PSW สามารถที่จะเซตหรือเคลียร์บิต ด้วยคำสั่งทางซอฟต์แวร์ได้

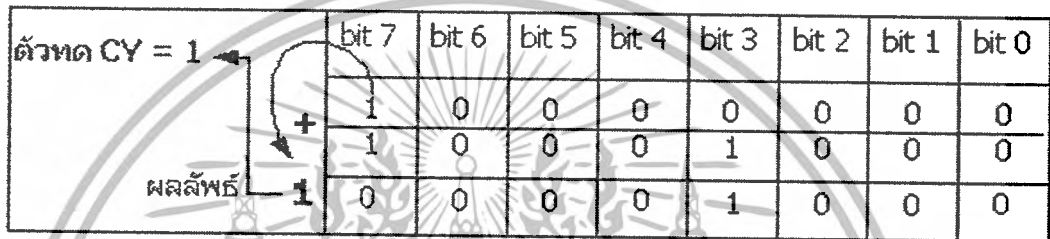
แฟล็ก ต่าง ๆ ของรีจิสเตอร์ PSW จะอยู่ในตำแหน่งของบิตต่างๆ ดังนี้

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	FO	RS1	RS0	OV	-	P

Register	RS0	RS1	ตำแหน่งหน่วยความจำ
BANK 0	0	0	00H - 07H
BANK 1	0	1	07H - 0FH
BANK 2	1	0	10H - 17H
BANK 3	1	1	18H - 1FH

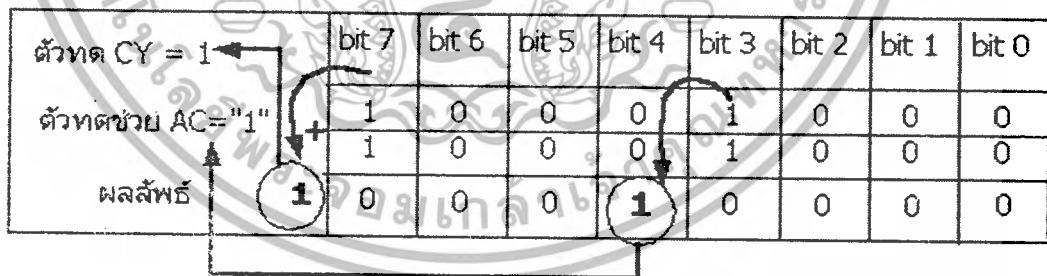
รูปที่ 2.27 แฟล็กต่าง ๆ ของรีจิสเตอร์ PSW

CY: (Carry Flag) เป็นบิตที่ทำหน้าที่แสดงสถานะของตัวทศ เช่นในกรณีของการบวก หากนำเลข 8 บิต 2 จำนวนมาบวกกัน แล้วปรากฏว่า ผลบวกที่ได้มีค่ามากกว่า 8 บิต ก็จะทำให้ สถานะของบิต CY ถูกเซตเป็น 1 แต่หากผลบวกไม่เกิน 8 บิตป้ายบอกสถานะที่บิต CY จะยังเป็น 0 และในทำนองเดียวกัน จะทำหน้าที่เป็นตัวยืมในกรณีของการลบ ใช้เป็นตัวร่วมกับแอดคิวิตูเลเตอร์ (Register A) ในการหมุนบิต และสามารถใช้ค่าของ CY เป็นเงื่อนไขในการเขียนโปรแกรมในการกระโดด (Jump) ได้



รูปที่ 2.28 ก. Carry Flag

AC: (Auxiliary Carry Flag) เป็น แฟล็ก ตัวทศช่วยในกรณีที่มีการบวกเลขสองจำนวน แล้วมีการทดระหว่างบิตที่ 3 ไปบิตที่ 4 ทำให้มีการเซตค่าที่บิต AC เป็น "1" ดังตัวอย่าง



รูปที่ 2.28 ข. Auxiliary Carry Flag

FO : (Flag 0) เป็น แฟล็ก ที่ใช้งานทั่วไปซึ่งสามารถใช้เป็นแฟล็กสถานะ (Status flag) ของโปรแกรมเมอร์ หรือรีเซตด้วยคำสั่ง ทางซอฟต์แวร์(สำรองไว้สำหรับผู้ใช้) RS1-RS0 (Register Bank Select) เป็นตัวกำหนดการเลือกพื้นที่ใช้งานของกลุ่มรีจิสเตอร์ R0 - R7 ในฟังก์ชัน

ต่างๆโดยการกำหนดสถานะที่บิต RS0 และ RS1 ตามตาราง Register RS0 RS1 ตำแหน่งหน่วยความจำ

OV : (Overflow Flag) เป็นบิตที่แสดงสถานะโอเวอร์โฟลว์ ซึ่งจะถูกเซตหรือเคลียร์จากการทำงานของคำสั่งทางคณิตศาสตร์ แล้วเกิดการทศข้ามจากบิตที่ 6มายังบิตที่ 7 เช่นในการนำเลขสองจำนวนมารวมกัน แล้วได้ผลลัพธ์มากกว่า +127(ฐานสิบ) หรือต่ำกว่า -128(ฐานสิบ) ในบิตที่ 7 (ซ้ายมือสุด) จะแสดงเป็นบิตสถานะของค่าบวก หรือลบโดยถ้าสถานะเป็น 1 จะเป็นค่าบวก ถ้าสถานะเป็น 0 จะเป็นค่าลบ ดังนั้นเมื่อมีการเกิดโอเวอร์โฟลว์ขึ้น จะทำให้แฟล็ก OV ถูกเซตเป็น "1"

P : (Parity Flag) เป็นบิตแสดงสถานะที่ใช้ตรวจสอบจำนวนบิตที่เป็น "1" ในข้อมูลของแอดคิวมูลเตอร์ (Register A) โดยบิต P จะถูกเซตเป็น "1" เมื่อสถานะทั้ง 8 บิตมีเลข 1 เป็นจำนวนคี่ (odd) และบิต P จะถูกเซตเป็น "0" เมื่อสถานะของทั้ง 8 บิตในแอดคิวมูลเตอร์ (Register A) มีจำนวนเลข 1 เป็นจำนวนคู่ (even) หรือ นับจำนวนเลข 1 ของข้อมูลในรีจิสเตอร์ A ทั้ง 8 บิตนั่นเอง

ตัวชี้สแต็ก SP (Stack Pointer) เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง 81H ใช้ในการเก็บค่าของตัวชี้บอก ตำแหน่งแอดเดรส เมื่อรีเซตระบบของไมโครคอนโทรลเลอร์ ค่าของตัวชี้สแต็กจะถูกกำหนดให้เริ่มต้นชี้ที่ตำแหน่งแอดเดรส 07H (ข้อมูลที่รีจิสเตอร์ SP จะมีค่าเท่ากับ 07H) ซึ่งจะเป็นตำแหน่งแอดเดรสของรีจิสเตอร์ R7 ในแบงค์ 0 สแต็ก(Stack) คือการจองพื้นที่หน่วยความจำข้อมูล บริเวณหนึ่งขึ้นมา เป็นตำแหน่งที่ใช้ในการเก็บข้อมูล โดยให้ รีจิสเตอร์ SP (Stack Pointer) เป็นตัวชี้บอกตำแหน่งแอดเดรสว่าข้อมูลนั้นเก็บไว้ที่ตำแหน่งใดของ หน่วยความจำข้อมูล และหลังจากที่นำข้อมูลไปเก็บไว้ในหน่วยความจำพื้นที่ๆ จองไว้แล้ว ค่าข้อมูลในตัวชี้สแต็ก (รีจิสเตอร์ SP) จะค่าไปยังตำแหน่งแอดเดรสใหม่ต่อไป (ขึ้นอยู่กับจำนวนของไบต์ข้อมูลที่จะนำไปเก็บ) ดังนั้นถ้าหากนำข้อมูลค่าต่อไปจัดเก็บอีก ซึ่งเป็นตำแหน่งแอดเดรสที่อยู่ถัดไปสแต็กถูกใช้งานในขณะที่มีการเรียกใช้โปรแกรมย่อยโดยคำสั่ง CALL หลังจากนั้นนำแอดเดรสที่ตำแหน่งถัดจากคำสั่ง CALL (โปรแกรมแกนเตอร์) ไปเก็บไว้ที่หน่วยความจำในพื้นที่ๆจองไว้ และหลังจากที่ไมโครคอนโทรลเลอร์ทำงานตามคำสั่งในโปรแกรมย่อยจนเสร็จสิ้นแล้ว โปรแกรมย่อยก็จะปิดท้ายด้วยคำสั่ง RET ดังนั้นไมโครคอนโทรลเลอร์จะกลับไปตำแหน่งเดิมได้ ก็จะต้องไปดูข้อมูลในตัวชี้สแต็ก (Stack Pointer) ว่าชี้อยู่ที่ตำแหน่งแอดเดรสใด ซึ่งค่าข้อมูลของสแต็กในตำแหน่งนั้น คือแอดเดรสที่โปรแกรมจะต้องกลับไปทำงานต่อ ดังนั้นการเรียกโปรแกรมย่อย ซ้อนกันหลายๆครั้ง ค่าในสแต็กก็จะซ้อนทับกัน การที่จะกลับค่าไปที่ตำแหน่งแอดเดรสเดิมได้ ต้องออกจากสแต็กที่ล้นขึ้นไป ซึ่งเรียกวิธีการนี้ว่า FILO (First In Last Out) หรือเข้าก่อนออกทีหลัง ในส่วนของสแต็กยัง

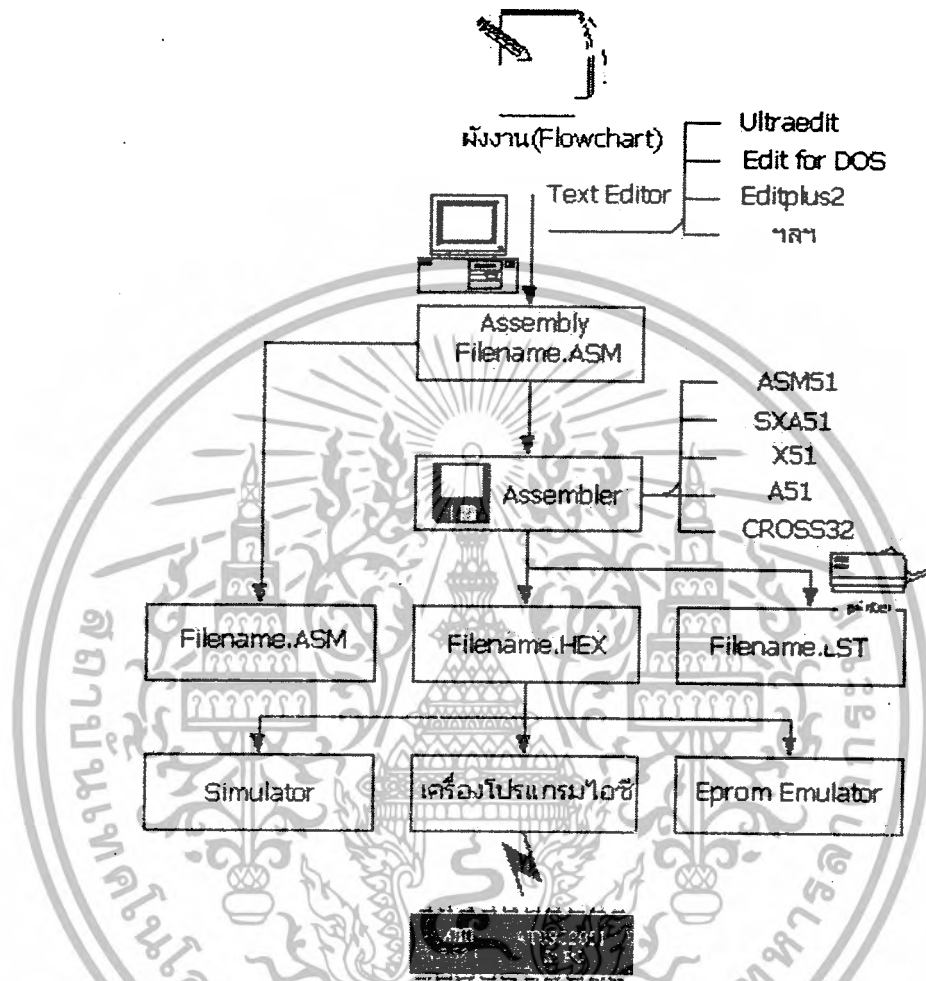
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถเก็บข้อมูลของรีจิสเตอร์ต่างๆได้ด้วย เช่น กรณีไมโครคอนโทรลเลอร์ข้ามไปทำที่โปรแกรมย่อย และในส่วนของโปรแกรมน้อยมีคำสั่งโดยเรียกใช้รีจิสเตอร์ที่เราใช้ในโปรแกรมหลักด้วย เช่น รีจิสเตอร์ PSW รีจิสเตอร์ A หรือ รีจิสเตอร์ R0 ฯลฯ ซึ่งถ้าหากไม่เก็บค่าข้อมูลเดิมของรีจิสเตอร์ไว้ก่อน ค่าข้อมูลในรีจิสเตอร์ดังกล่าวอาจจะเปลี่ยนแปลงข้อมูลไป ดังนั้นจึงต้องนำข้อมูลเดิมไปเก็บไว้ที่สแต็กก่อนด้วยคำสั่ง PUSH หลังจากนั้นเมื่อออกจากโปรแกรมน้อยจึงทำการคืนค่าของรีจิสเตอร์ในสแต็กกลับมา ให้เป็นข้อมูลเดิมโดยคำสั่ง POP ทุกครั้งที่เริ่มรีเซตระบบ ไอซีไมโครคอนโทรลเลอร์ เริ่มทำงานที่ตำแหน่งแอดเดรส 0000H ของหน่วยความจำโปรแกรมแฟลช (Flash memory) และเริ่มปฏิบัติตามคำสั่งที่ถูกเขียน เป็นข้อมูลขนาด 8 บิต(1 ไบต์)ในแต่ละแอดเดรสของหน่วยความจำ โปรแกรม ซึ่งบางคำสั่งอาจจะใช้เนื้อที่ในการเก็บข้อมูลเพียง 1 แอดเดรส และบางคำสั่งอาจจะต้องใช้เนื้อที่ 2-3 แอดเดรส จำนวนของแอดเดรสที่เก็บข้อมูลในแต่ละคำสั่งสามารถเปิดดูได้จากตารางคำสั่งของไมโครคอนโทรลเลอร์ ดังนั้นไอซีไมโครคอนโทรลเลอร์ที่มีหน่วยความจำโปรแกรมขนาด 1Kbytes ซึ่งเหมือนมีกระดาษไว้เขียนคำสั่ง 1,024 บรรทัด หากมีการเขียนคำสั่งที่ยาวกว่า 1,024 บรรทัด แต่ละบรรทัดมีขนาด 8 บิต (1 ไบต์) ซึ่งต้องเลือกไอซีที่มีหน่วยความจำโปรแกรมให้มากกว่านี้

DPTR (Data Pointer) เป็นรีจิสเตอร์ขนาด 16 บิตที่ประกอบด้วยรีจิสเตอร์ขนาด 8 บิต 2 ตัว คือรีจิสเตอร์ DPL และ DPH ซึ่งสามารถเลือกการใช้งานในลักษณะ 8 บิต 2 ตัวหรือ 16 บิต 1 ตัวก็ได้ มีแอดเดรสอยู่ที่ตำแหน่ง 82H,83H ตามลำดับ ใช้สำหรับเป็นตัวชี้ ตำแหน่งของหน่วยความจำหรือตำแหน่งของอุปกรณ์อินพุตและเอาต์พุตที่ไมโครคอนโทรลเลอร์ต้องการติดต่อกับ และใช้เป็นตัวกำหนดตำแหน่งเริ่มต้น (Base)

รีจิสเตอร์พอร์ต P0-P3 พอร์ต P1และพอร์ต P3 เป็นรีจิสเตอร์ขนาด 8 บิต ซึ่งพอร์ต P0 มีแอดเดรสอยู่ที่ตำแหน่ง 80H พอร์ต P1 มีแอดเดรสอยู่ที่ตำแหน่ง 90H พอร์ต P2มีแอดเดรสอยู่ที่ตำแหน่ง A0H และพอร์ต P3มีแอดเดรสอยู่ที่ตำแหน่ง B0H ข้อมูลที่อยู่ในรีจิสเตอร์ เหล่านี้จะเป็นค่าเดียวกับค่าของสัญญาณที่ขาต่างๆของพอร์ต เป็นรีจิสเตอร์ที่สามารถเข้าถึงข้อมูลแบบบิตได้ เป็นพอร์ตแบบสองทิศทางคือเป็นได้ทั้ง พอร์ตอินพุต และพอร์ตเอาต์พุต สามารถคงสถานะ(Latch) ขณะเป็นเอาต์พุต สำหรับเบอร์ AT89C1051 พอร์ต P3 จะมีขาภายนอกเพียง 7 ขา ส่วนขาที่ P3.6 จะเป็นขาที่อยู่ภายใน ไอซีเป็นขาเอาต์พุตของวงจรเรียบบีตระหว่างขา P1.0 และ P1.1

### 2.10.3 การเขียนโปรแกรมภาษาแอสเซมบลี MCS51



รูปที่ 2.29 flow chart การเขียนโปรแกรมภาษาแอสเซมบลี MCS51

#### การเขียนโปรแกรมแอสเซมบลี

##### โครงสร้างของโปรแกรมแอสเซมบลี

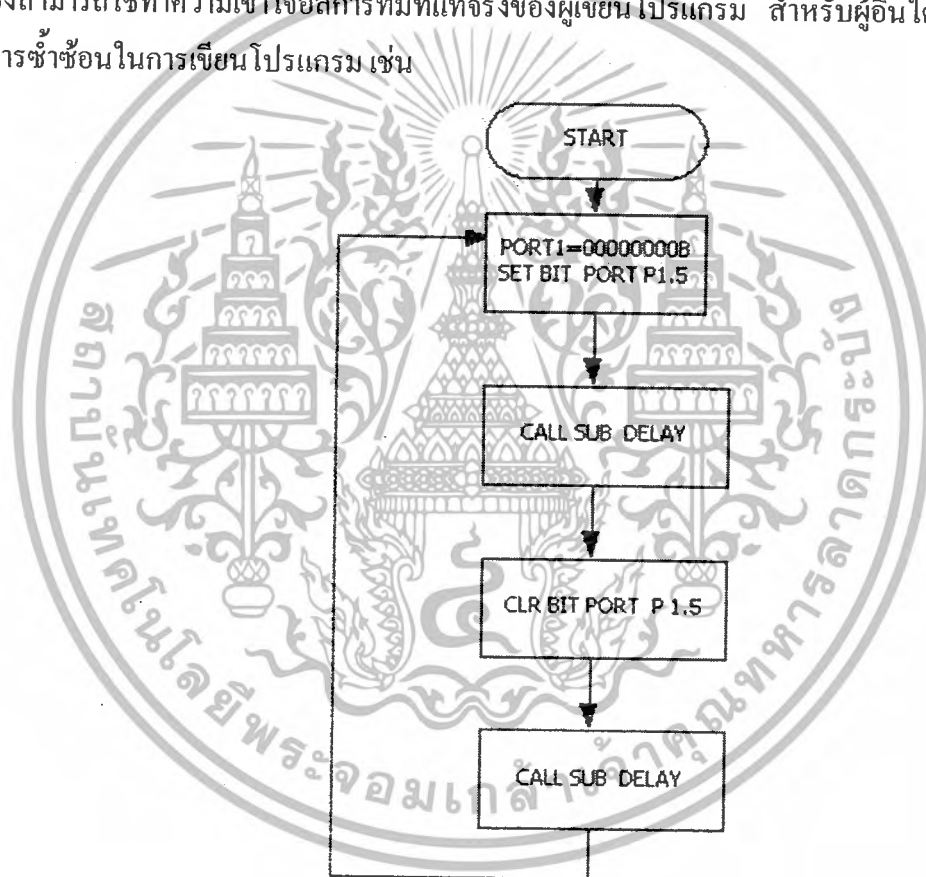
โปรแกรมจะถูกเก็บไว้ในหน่วยความจำ ซึ่งอยู่ในรูปแบบของเลขไบนารี เรียกว่าภาษา Machine ซึ่งเป็นภาษาที่สามารถ ติดต่อกับคอมพิวเตอร์ เข้าใจได้ ภาษา Machine นี้จะจัดให้อยู่ในรูปแบบที่เป็นเลขฐานสิบหก (HEX) เช่น คำสั่ง 8 บิต 11101011B (B-ไบนารี) เขียนได้เป็น 0EBH(H-ฐานสิบหก) แต่เป็นการที่จะเข้าใจความหมาย ยากในการใช้งาน การที่จะทำความเข้าใจภาษา Machine จะมีการใช้สัญลักษณ์ (Symbol) ที่เรียกว่า Mnemonics เพื่อแทนความหมายของคำสั่ง เช่น MOV A,#67H หมายความว่า นำข้อมูลค่าคงที่ 67H ไปเก็บไว้ใน reg. A) โปรแกรมที่เขียน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น เมื่อผู้ผู้ใดเห็นเป็นประโยชน์ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์รหัส Mnemonics เรียกว่า ภาษา Assembly และก่อนที่จะให้ CPU ทำงานตามโปรแกรมที่เขียนด้วยภาษา Assembly ได้ แต่ต้องเปลี่ยนให้เป็นภาษา Machine ก่อน โดยใช้ โปรแกรมแอสเซมเบลอร์

### 1. Flowchart โฟลว์ชาร์ต

โฟลว์ชาร์ตใช้แสดงลักษณะและอัลกอริทึมของโปรแกรมในรูปแบบผังกราฟที่เหมาะสมเป็นการลำดับงานแบบเป็นขั้นตอน โฟลว์ชาร์ต มีความสำคัญมากในการทำ Program-Debugging ซึ่งสามารถใช้ทำความเข้าใจอัลกอริทึมที่แท้จริงของผู้เขียนโปรแกรม สำหรับผู้อื่นได้ดี และไม่เกิดการซ้ำซ้อนในการเขียนโปรแกรม เช่น



รูปที่ 2.30 FLOW CHART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. Statement

รูปแบบโปรแกรมภาษา Assembly ประกอบด้วย ส่วนต่างๆ ( Field) 4 ส่วน คือ Label, Op-code, Operand และ Comment ในการเขียนในแต่ละคำสั่ง เรียกว่า Statement เช่น

Label	Op-code	Operand	Comment
LOOP1:	MOV	A,#0FFH	; REG A= 11111111B
	MOV	P1,A	;นำค่าข้อมูลในรีจิสเตอร์ A ออกไปที่พอร์ต P1
	ACALL	DELAY	;เรียกโปรแกรมย่อยหน่วงเวลา
	MOV	P1,#00000000B	;นำค่าคงที่ 00000000B ออกไปที่พอร์ต P1
	ACALL	DELAY	;เรียกโปรแกรมย่อยหน่วงเวลา
	SJMP	LOOP	;วนไปที่เลเบล LOOP ใหม่

## 3. Label

เลเบลจะต้องขึ้นต้นด้วยตัวอักษร “A-Z” หรือเครื่องหมาย “\_”, “.” จะเป็นตัวอักษรจาก A ถึง Z หรือตัวเลขจาก 0 ถึง 9 หรือเครื่องหมาย “?” เลเบลจะยาวกี่ตัวอักษรก็ได้ แต่เมื่อรวมทั้งบรรทัดแล้วต้องไม่เกิน 255 ตัวอักษร ในการเขียนเลเบลจะต้องมีเครื่องหมาย “:” (Colon) ตามหลังเลเบลด้วยทุกครั้ง เช่น LOOP:

แนะนำ การใช้เลเบลควรจะให้สื่อต่อตัวโปรแกรมที่จะเรียกใช้ เช่น ชื่อและหน้าที่ยกตัวอย่าง โปรแกรมย่อยหน่วงเวลาควรตั้งชื่อเป็น DELAY

## 4. Op-code

เป็นส่วนของนิโม่ทิกที่จะแสดงจุดมุ่งหมายของการกระทำแต่ละคำสั่ง เช่น MOV ADD SUB ในช่องว่างระหว่าง Op-Code และ โอเปอร์เรนด์จะต้องมีช่องว่างกันอยู่อย่างน้อย 1 ช่อง เช่น MOV A,R1

## 5. Operand

โอเปอร์เรนด์เป็นส่วนที่แสดงถึงส่วนที่ถูกกระทำ อาจประกอบด้วย 2 ส่วนที่มีกา: “ ” หรือการแสดงถึงตำแหน่งที่จะกระทำ

## 6. Comment

เป็นการเขียนคำอธิบายแต่ละบรรทัดหรือหมายเหตุนั้นและที่สำคัญจะต้องใช้เครื่องหมาย “;” (Semi-Colon) นำหน้าด้วยทุกครั้ง

## โปรแกรมAssembler

โปรแกรมที่เขียนด้วยรหัส Mnemonics เรียกว่า ภาษา Assembly และก่อนที่จะให้ CPU ทำงานตามโปรแกรมที่เขียนด้วยภาษา Assembly ได้ต้องเปลี่ยนให้เป็นภาษา Machine ก่อน โดยใช้โปรแกรมพิเศษช่วยเปลี่ยนเรียกว่า Assembler โดยปกติ โปรแกรมจะเขียนในรูป Assembly ซึ่งมีข้อดีกว่าเขียนด้วยภาษา Machine กล่าวคือ เขียนได้รวดเร็วกว่าจำได้ง่ายและ ตัวโปรแกรม Assembler มีส่วนช่วยในการตรวจสอบโปรแกรมด้วย จึงต้องใช้ตัว Assembler และระบบพัฒนาบนคอมพิวเตอร์ โปรแกรมที่ใช้งานแอสเซมบลอร์มีอยู่หลายโปรแกรมอาทิ เช่น CROSS32, X51, ASM51 และ SXA51

### 1. SXA51

SXA51 เป็นโปรแกรมที่ใช้ในการ Assembler ซีพียู MCS-51 โดยเขียนโปรแกรมที่เป็น Source โปรแกรม ด้วยโปรแกรมอิดิเตอร์ตัวใดก็ได้ (Q-EDIT, SK, EditPlus2, EDIT For Dos หรือ UltraEdit-32) แล้วให้ทำการบันทึกเป็นไฟล์นามสกุล .ASM ไว้โดยเขียนเฉพาะคำสั่งเท่านั้นแล้วตัวโปรแกรม SXA51 จะทำการแปลภาษาเครื่องออกมา เป็นเลขฐาน 16 สามารถนำมาโปรแกรมให้กับ ไอซีได้เลย

C:\> SXA51 - <Option> File name.ASM <CR>

- <OPTION> แสดงตัวเลือกให้ดังรายการ

-L = ให้สร้างไฟล์แสดงรายละเอียด Address Op-code โดยจะได้ไฟล์นามสกุล .LST

-N = ไม่สร้างไฟล์เอาต์พุตใดๆ เพื่อความรวดเร็วในการทำงาน

-C = สร้าง Symbol cross-reference ไว้ที่ท้ายของโปรแกรมเพื่อความสะดวกของการหาคำแทน่งของ LABEL

-D = แสดง Process ขณะกำลังทำการ Assembler โดยโปรแกรมจะแสดงเลขบรรทัดที่กำลังแปลอยู่

## โปรแกรมแอสเซมเบลอร์ Cross32

โปรแกรม Cross32 เป็นโปรแกรมครอส - แอสเซมเบลอร์ (Cross - Assembler) แบบตรวจสอบสองครั้ง (two pass) โดยมีทางเลือกแบบตรวจสอบครั้งที่สาม (third pass) ถ้าหากตรวจสอบพบ phase error ในโปรแกรมต้นแบบ การใช้งานโปรแกรม Cross32 จะต้องมีการกำหนดเบอร์ของไมโครโพรเซสเซอร์ที่ต้องการ โดยใช้คำสั่ง CPU directive ในส่วนหัวของโปรแกรม เช่น CPU "8051.TBL" จะเป็นการระบุให้โปรแกรม Cross32 ทราบว่าต้องการใช้ตารางคำสั่งของ CPU เบอร์ 8051 และต้องกำหนดรูปแบบของไฟล์แสดงรหัสภาษาเครื่อง โดยการใช้คำสั่ง HOF directive เพื่อระบุให้โปรแกรม ครอส 32 ทราบว่า จะต้องวางไฟล์ตามมาตรฐานใด เช่น HOF "INT8" เป็นการกำหนดว่าให้วางไฟล์ในรูปแบบตามมาตรฐานของบริษัทอินเทล เป็นต้น

การเขียนโปรแกรมจะใช้เท็กซ์ เอดิเตอร์ (Text Editor) สร้างโปรแกรมต้นแบบ โดยพิมพ์ที่ส่วนหัวของโปรแกรมที่มีรายละเอียดดังนี้

CPU "8051.TBL"	; ระบุตารางคำสั่งของไมโครคอนโทรลเลอร์เบอร์ 8051
HOF "INT8"	; กำหนดรูปแบบของไฟล์แสดงรหัสมาตรฐานอินเทล
ORG 0000H	; ระบุตำแหน่งแอดเดรสในหน่วยความจำ
	เพื่อที่จะให้เริ่มต้นการทำงานของโปรแกรมส่วนของโปรแกรมที่เป็นคำสั่ง(แอสเซมบลี)
END	; จบโปรแกรม

รูปแบบของคำสั่งที่ใช้งานจะเป็นดังนี้

**C32 Filename.asm [-L LISTFILE] [-H HEXFILE]**

โดยที่ส่วนที่อยู่ใน [ ] เป็นทางเลือก จะใช้หรือไม่ใช้ก็ได้

Filename.asm หมายถึง โปรแกรมต้นแบบ (Source Program) ที่เขียนขึ้นโดยมีนามสกุลเป็น .ASM

- L list file หมายถึง ให้ Cross32 สร้างไฟล์แสดงรายละเอียดการแอสเซมเบลอร์ (assemble listing file) โดยใช้ชื่อไฟล์ที่ตามหลังมา

- H hex file หมายถึง ให้ Cross32 สร้างไฟล์รหัสภาษาเครื่อง (machine code) ออกมาในไฟล์ที่ ตามหลังมา

ถ้าไม่ใช้ทางเลือก - L หรือ - H โปรแกรม Cross32 จะไม่สร้างไฟล์เหล่านี้ ขึ้นมา ในกรณีเช่นนี้รหัสที่แสดงการผิดพลาด (error code) จะแสดงออกมาจอภาพ (ถ้ามี) แต่ถ้าเลือกทางเลือก - L

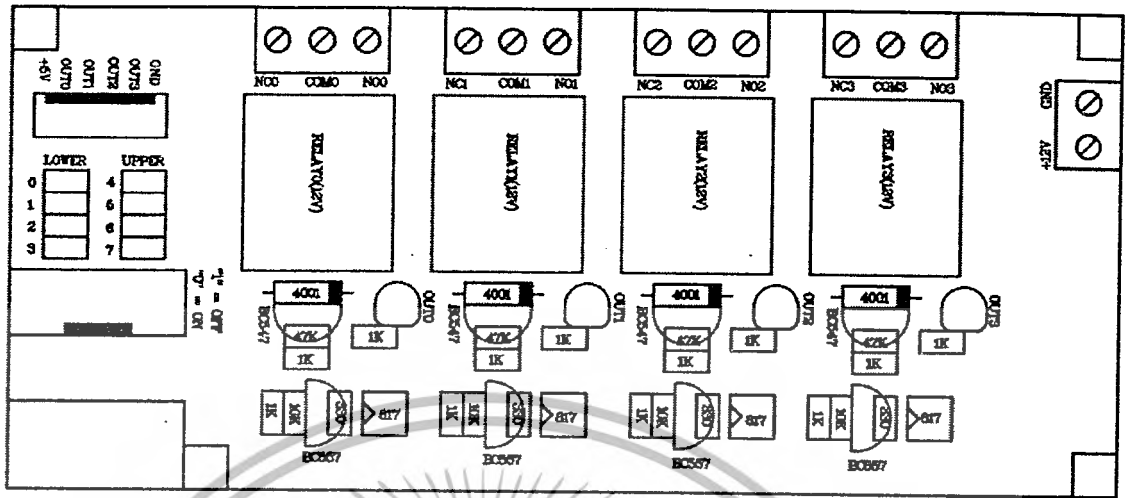
แล้ว รหัสผิดพลาดจะถูกสร้างขึ้นใน list file ด้วย (ใช้โปรแกรม Tex Editor ตรวจสอบรายละเอียดของข้อผิดพลาดในตำแหน่งต่างๆได้)

### โปรแกรมแอสเซมเบลอร์อื่นๆ

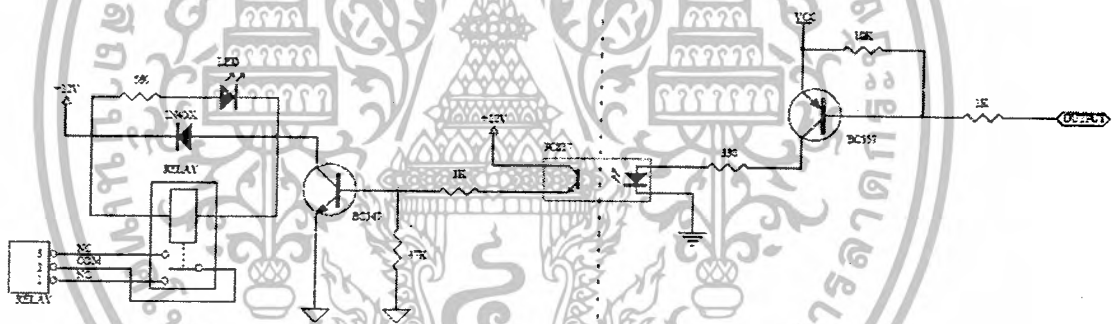
โปรแกรมที่เป็นแอสเซมเบลอร์ตัวอื่นๆ อาจจะมีวิธีการใช้กำกับมาด้วย การใช้งานและคำสั่งที่เป็นแอสเซมเบลอร์ไครีททีฟ (Assembler Directives) จะเป็นของแอสเซมเบลอร์แต่ละตัว แต่จะใกล้เคียงกันมากส่วนคำสั่งของ MCS-51 จะใช้เหมือนกันทุกประการ

## 2.11 ET-OPTO RELAY4

บอร์ด ET-OPTO RELAY4 เป็นชุด OPTO-ISOLATE OUTPUT แบบหน้าสัมผัส RELAY ขนาด 4 ช่อง ใช้สำหรับควบคุมการทำงานของอุปกรณ์ไฟฟ้าต่างๆ ในลักษณะของหน้าสัมผัส ON/OFF โดยใช้สัญญาณในการควบคุมการทำงานของ RELAY ด้วยสัญญาณลอจิก TTL ซึ่งวงจรสำหรับควบคุมการทำงานของ RELAY นั้น จะใช้วงจรแบบ OPTO-ISOLATE ในการควบคุมการทำงาน เพื่อป้องกันสัญญาณรบกวนต่างๆที่เกิดจากการทำงานของ RELAY และอุปกรณ์ Output ที่ควบคุมการทำงานจากหน้าสัมผัส RELAY ที่จะย้อนกลับมารบกวนการทำงานของวงจรควบคุมหรือไมโครคอนโทรลเลอร์ต่างๆ ทำให้ประสิทธิภาพการทำงานของระบบไมโครคอนโทรลเลอร์ต่างๆมีความน่าเชื่อถือมากยิ่งขึ้น โดยการทำงานของวงจรมันจะออกแบบให้การทำงานของ RELAY ทำงานด้วยสถานะลอจิก “0” และหยุดทำงานด้วยสถานะลอจิก “1” เพื่อป้องกันไม่ให้ RELAY ทำงานในขณะที่ระบบไมโครคอนโทรลเลอร์เกิดการ RESET หรือสัญญาณควบคุมของบอร์ด ET-OPTO RELAY4 ถูกปล่อยลอยอยู่



รูปที่ 2.31 โครงสร้างของบอร์ด ET-OPTO RELAY4



รูปที่ 2.32 วงจรของ ET-OPTO RELAY4 ขนาด 1 ช่อง

2.11.1 การทำงานของบอร์ด

ในการใช้งานบอร์ด ET-OPTO RELAY4 เพื่อควบคุมการทำงานของอุปกรณ์ต่างๆ ในลักษณะของหน้าสัมผัสสวิตช์นั้น จะต้องมีการป้อนแรงดันไฟตรงขนาด +12 VDC เพื่อใช้เป็นไฟเลี้ยงขดลวดของ RELAY ด้วย ซึ่งแรงดัน +12VDC นี้ควรเป็นแรงดันไฟตรงที่มาจากหม้อแปลงซึ่งแยกขดลวดกับชุดที่จะจ่ายให้กับบอร์ดไมโครคอนโทรลเลอร์ด้วย เพื่อป้องกันการรบกวนที่จะเกิดขึ้นกับวงจรไมโครคอนโทรลเลอร์ในขณะที่ RELAY หรืออุปกรณ์ Output ที่ถูกควบคุมจาก RELAY ทำงาน ซึ่งการทำงานของบอร์ดจะเริ่มจากเมื่อมีการเปลี่ยนสถานะ โลจิก " 0 " ให้กับ OUTPUT ของวงจร ซึ่งจะส่งผลให้ขา Base ของทรานซิสเตอร์ BC557 ที่อยู่คู่กับภาค Input ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

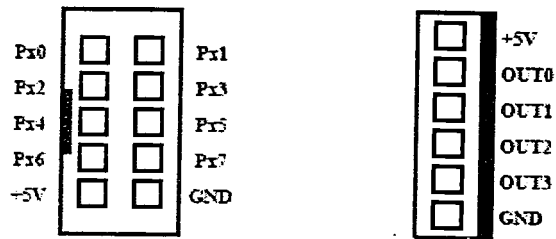
OPTO ISOLATE (PC817) ได้รับการไบอัสที่ถูกต้อง ทำให้ทรานซิสเตอร์นำ กระแส ซึ่งจะทำให้มีแรงดัน +VCC(+5VDC)ไหลออกจากขา Emitter ไปยัง Collector ผ่านตัวต้านทาน 330 โอห์มเพื่อจำกัดกระแสให้กับ OPTO ISOLATE (PC817) ซึ่งส่งผลให้วงจร OPTO ISOLATE ทำงานเป็นผลให้เกิดมีกระแสไหลจากแหล่งจ่าย +12VDC ผ่านภาค OUTPUT ของ OPTO ISOLATE (PC817) โดยแรงดัน +12VDC นี้จะถูกจำกัดขนาดของกระแสด้วยตัวต้านทานค่า 1k OHM ก่อนที่จะป้อนให้กับขา Base ของทรานซิสเตอร์ BC547 ที่ต่ออยู่กับวงจรภาค Output ของ OPTO ISOLATE เป็นผลให้ทรานซิสเตอร์ BC547 นำกระแสซึ่งก็จะทำให้มีกระแสไฟไหลผ่านขดลวดของ RELAY ทำให้ RELAY และ LED ทำงาน โดยในการประยุกต์ใช้งานบอร์ดนั้น จะใช้สำหรับควบคุมการทำงานของอุปกรณ์ต่างๆที่ต้องการควบคุมการทำงานด้วยวิธีการ ON/OFF ของหน้าสัมผัส โดยลักษณะของหน้าสัมผัส RELAY นั้นจะมีให้เลือกใช้งาน 2 แบบ คือ NO และ NC โดยในการต่อใช้งานจะต้องเลือกแบบใดแบบหนึ่งโดยจะต่อใช้งานร่วมกับขา COM(Common)

- NO หรือ Normal Open ซึ่งหมายถึงหน้าสัมผัสแบบปกติเปิด และจะทำงานเมื่อ RELAY เริ่มต้นทำงาน

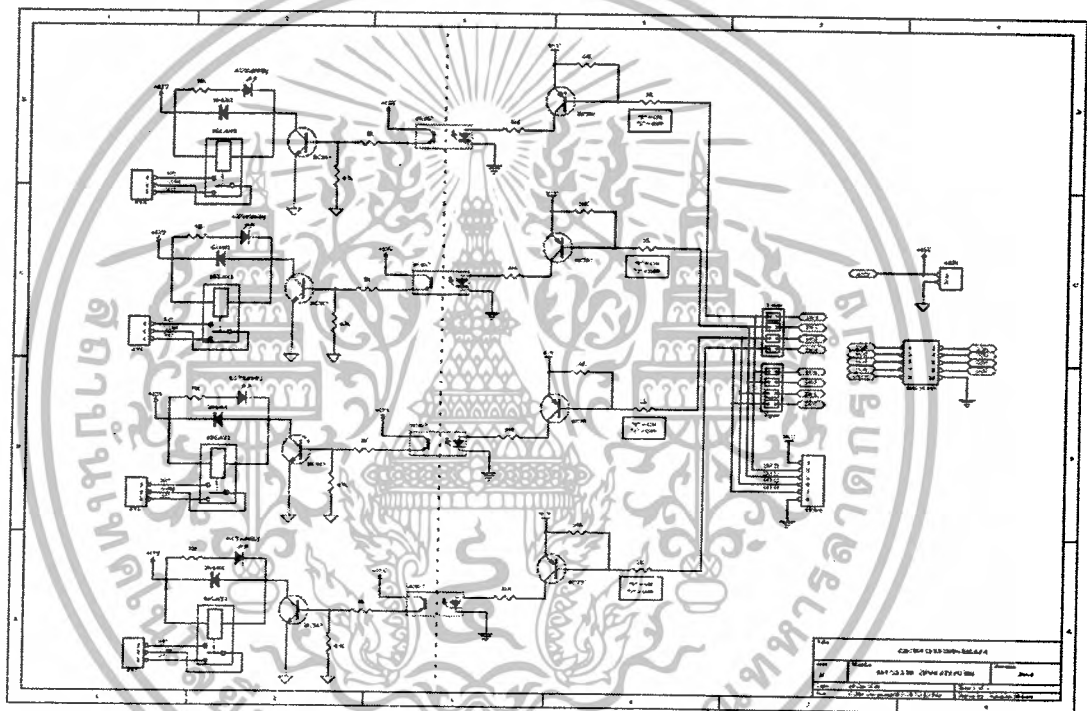
- NC หรือ Normal Close ซึ่งหมายถึงหน้าสัมผัสแบบปกติปิด โดยจะหยุดทำงานเมื่อ RELAY เริ่มทำงานสำหรับการเชื่อมต่อสัญญาณ TTL ไปให้กับวงจรภาค Output ของไมโครคอนโทรลเลอร์นั้น จะสามารถกระทำ ได้ 2 แบบ ด้วยกัน คือต่อผ่านขั้วต่อแบบ IDE ขนาด 10 Pin โดยใช้สายแพรเป็นสายนำสัญญาณ โดยสามารถเลือกกำหนดสัญญาณ TTL ที่ได้จากการทำงานของบอร์ด ได้ 2 กลุ่ม คือ LOWER และ UPPER ส่วนอีกวิธีหนึ่งจะสามารถกระทำได้โดยต่อสัญญาณผ่านขั้วต่อแบบ CPA ขนาด 6 Pin ดังตารางที่ 2.33

ขั้วต่อ Output (Terminal)	LOWER		UPPER	
	IDE 10 Pin	CPA 6 Pin	IDE 10 Pin	CPA 6 Pin
RELAY0	Px0	OUT0	Px4	OUT0
RELAY1	Px1	OUT1	Px5	OUT1
RELAY2	Px2	OUT2	Px6	OUT2
RELAY3	Px3	OUT3	Px7	OUT3

ตารางที่ 2.3 จุดเชื่อมต่อสัญญาณของบอร์ด ET-OPTO RELAY4 กับไมโครคอนโทรลเลอร์



รูปที่ 2.34 การจัดเรียงสัญญาณของขั้ว IDE-10 Pin และ CPA-6 Pin

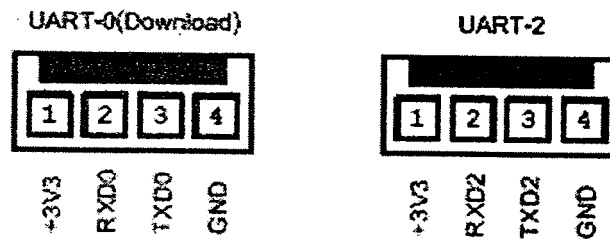


รูปที่ 2.35 วงจรของ ET-OPTO RELAY4

## 2.12 การใช้งาน RS232

พอร์ต RS232 เป็นสัญญาณ RS232 ซึ่งผ่านวงจรแปลงระดับสัญญาณ MAX3232 เรียบร้อยแล้ว โดยมีจำนวน 2 ช่อง คือ UART-0 และ UART-2 โดยทั้ง 2 ช่องสามารถใช้เชื่อมต่อกับสัญญาณ RS232 เพื่อรับส่งข้อมูลได้นอกจากนี้แล้ว UART-0 ยังสามารถใช้งานเป็น ISP Download สำหรับทำการ Download Hex File ให้กับ MCU ได้ด้วย โดยในกรณีนี้ต้องใช้งานร่วมกับ SW1 (ISP\_LOAD) และ SW2 (RESET) เพื่อ Reset ให้คอมพิวเตอรื เริ่มต้นทำงานใน Boot-Loader Mode เพื่อทำการ Download Hex File ให้กับคอมพิวเตอรืได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.36 ช่องสัญญาณใช้เชื่อมต่อกับ RS232 เพื่อรับส่งข้อมูล

- UART-0 ใช้หาสัญญาณจาก P0.2(TXD0) และ P0.3(RXD0)
- UART-2 ใช้หาสัญญาณจาก P0.10(TXD2) และ P0.11(RXD2)

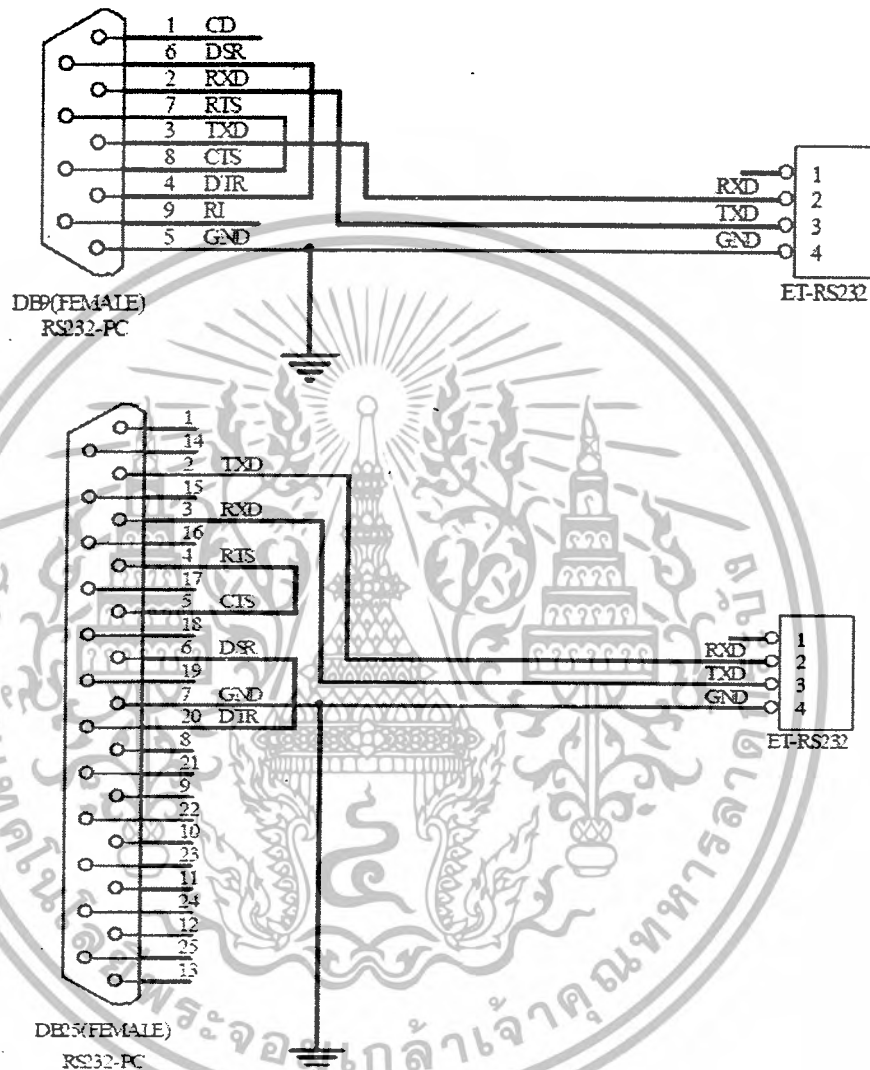
เนื่องจากระบบ Hardware UART ของ LPC2368 นั้นจะสามารถกำหนดหาสัญญาณในการเชื่อมต่อได้หลายจุด ตัวอย่างเช่น UART-2 สามารถเลือกใช้หาสัญญาณ P0[10] กับ P0[11] หรือ P2[8] กับ P2[9] ก็ได้ ซึ่งบอร์ด CP-JR ARM7 LPC2368 นั้นเลือกใช้หาสัญญาณชุด P0[10] กับ P0[11] เป็นจุดเชื่อมต่อกับ UART-2 ดังนั้น ผู้ใช้ต้องกำหนดคำสั่งในการเลือกใช้หาสัญญาณให้ถูกต้องด้วย และข้อควรระวังอีกประการหนึ่งในการใช้งาน UART ก็คือ ค่า Default ของ UART-2 จะถูกปิดการทำงานไว้ ดังนั้นผู้ใช้ต้องสั่งเปิดการทำงานของวงจร UART-2 ก่อนที่จะสั่ง Initial ค่าต่างๆให้กับ UART ด้วย ไม่เช่นนั้นจะไม่สามารถสั่งงาน UART ได้ สำหรับ Code ตัวอย่างการกำหนดค่า UART ในส่วนเริ่มต้นเป็นดังนี้

```
// Config UART-0 Connect to P0[2]:P0[3]
PINSEL0 &= 0xFFFFF0F; // Reset P0.2, P0.3 Pin Config
PINSEL0 |= 0x0000010; // Select P0.2 = TXD(UART 0)
PINSEL0 |= 0x0000040; // Select P0.3 = RXD(UART 0)

// Config UART-2 Connect to P0[10]:P0[11]
PINSEL0 &= 0xFF0FFFFF; // Reset P0.10, P0.11 Pin Config
PINSEL0 |= 0x00100000; // Select P0.10 = TXD(UART-2)
PINSEL0 |= 0x00400000; // Select P0.11 = RXD(UART-2)
PCONP |= 0x01000000; // UART2 Power-ON
```

รูปที่ 2.37 ก. ตัวอย่างโปรแกรมกำหนดค่า UART

สำหรับ Cable ที่จะใช้ในการเชื่อมต่อ RS232 ระหว่าง Comport ของเครื่องคอมพิวเตอร์เข้า  
กับขั้วต่อ UART-0 และ UART-2 ของบอร์ด CP-JR ARM7 LPC2368 นั้น เป็นดังนี้



รูปที่ 2.38 วงจรสาย Cable สำหรับ RS232

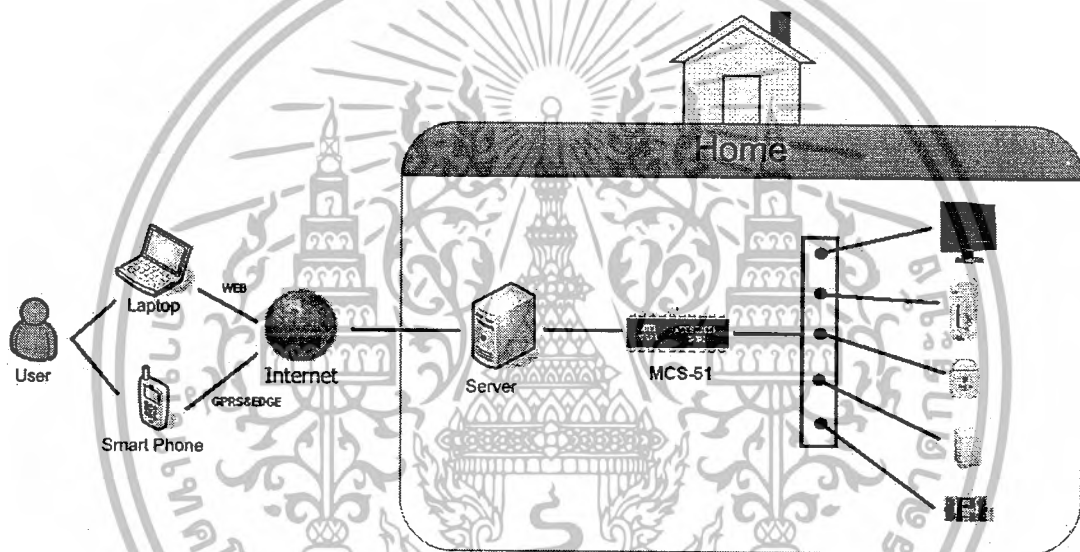
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 3

### การออกแบบ

#### 3.1 ภาพโดยรวมของระบบ

ระบบนี้เป็นการนำเอาความสามารถของ Internet มาประยุกต์ใช้กับระบบควบคุมภายในบ้าน ทำให้สามารถควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้อย่างสะดวก รวดเร็ว และประหยัดเวลาในการควบคุมและไม่จำกัดอยู่เพียงแค่คอมพิวเตอร์ตั้งโต๊ะเพียงอย่างเดียว โดยได้มีการนำเอาโทรศัพท์มือถือมาช่วยในการควบคุมให้การใช้งานได้สะดวกยิ่งขึ้น ภาพโดยรวมของระบบแสดงได้ดังในรูปข้างล่าง



รูปที่ 4.1 Diagram แสดงภาพ โดยรวมของ โครงการ

การทำงานโดยรวมของระบบ เริ่มโดย User ใช้เครื่องคอมพิวเตอร์ หรือ PDA Phone ทำการ log in เข้าหน้าหลักของการใช้งานผ่านเว็บไซต์ ซึ่งจะทำการติดต่อกับ เว็บเซิร์ฟเวอร์ โดยที่เครื่องเซิร์ฟเวอร์ทำการเชื่อมต่อ และส่งข้อมูลไปที่ ไมโครคอนโทรลเลอร์ MCS-51 จากนั้นไมโครคอนโทรลเลอร์ MCS-51 ทำการเชื่อมต่อกับวงจร Relay เพื่อควบคุมอุปกรณ์ เครื่องใช้ไฟฟ้าต่างๆ ดังแสดงออกมาในหน้าหลักของเว็บไซต์

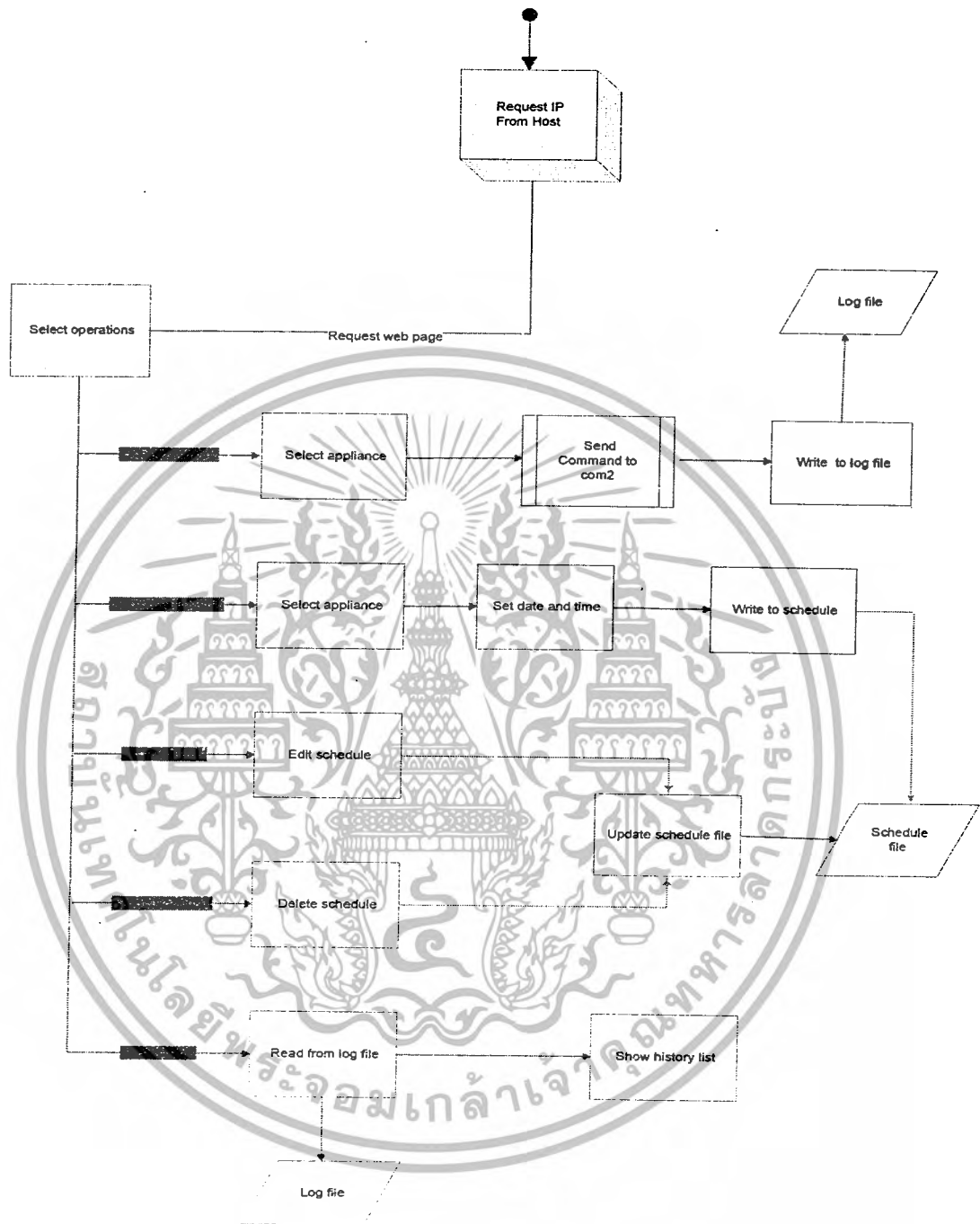
### 3.2 การออกแบบ

ในปฏิญญาฉบับนี้เป็นการออกแบบระบบควบคุมเครื่องใช้ไฟฟ้าภายในบ้าน โดยใช้คอมพิวเตอร์ผ่านเครือข่ายinternet เป็นหลัก ในส่วนเพิ่มเติมก็คือการควบคุมโดยใช้โทรศัพท์มือถือ ซึ่งเป็นการสั่งงานโดยใช้เทคโนโลยีของ จีพีอาร์เอส และ เอคซ์ ซึ่งจะทำการติดต่อมาที่ Web Server เมื่อทำการติดต่อมาที่ Web Server แล้วก็จะทำการส่งคำสั่งในรูปแบบข้อมูลในการควบคุมอุปกรณ์มาให้แก่โปรแกรมควบคุมที่เขียนด้วย Visual Basic ประมวลผลไปสั่งงานส่วนฮาร์ดแวร์ซึ่งมีตัวไมโครคอนโทรลเลอร์คอยรับคำสั่งและควบคุม ส่วนของเซิร์ฟเวอร์เมื่อทำการสั่งงานแล้วก็จะทำการเก็บข้อมูลต่างๆ คือ ข้อมูลการสั่งงาน สถานะปัจจุบัน เวลาของการสั่งงาน ลงในฐานข้อมูลเพื่อที่จะสามารถกลับมาตรวจสอบภายหลังได้ว่าได้ทำการสั่งงานอะไรไปบ้างแล้ว ในส่วนของไมโครคอนโทรลเลอร์ก็จะมี การติดต่อกับฐานข้อมูลเพื่อเก็บสถานะของอุปกรณ์ไฟฟ้าและข้อมูลการสั่งงานเช่นกัน เพียงแต่ไม่กระทำผ่านเว็บเซิร์ฟเวอร์

### 3.3 การออกแบบ Web Site โดยใช้ภาษา VB.NET

ในการออกแบบ Web Site เพื่อที่จะให้โต้ตอบกับผู้ใช้ทั้งการรับข้อมูลเพื่อนำไปประมวลผลควบคุมอุปกรณ์ไฟฟ้าและแสดงผลการทำงานและข้อมูลที่เก็บไว้ในฐานข้อมูลนั้นได้ทำการใช้ ภาษา VB.NET ซึ่งการเขียนโปรแกรมของภาษาเหล่านี้ได้ใช้โปรแกรม Macromedia Dreamweaver 8 และ Visual Studio 2008 ซึ่งในการเขียน โดยรายละเอียดการออกแบบการทำงานของ Web Site จะแสดงโดยใช้ Flow Chart มีดังนี้

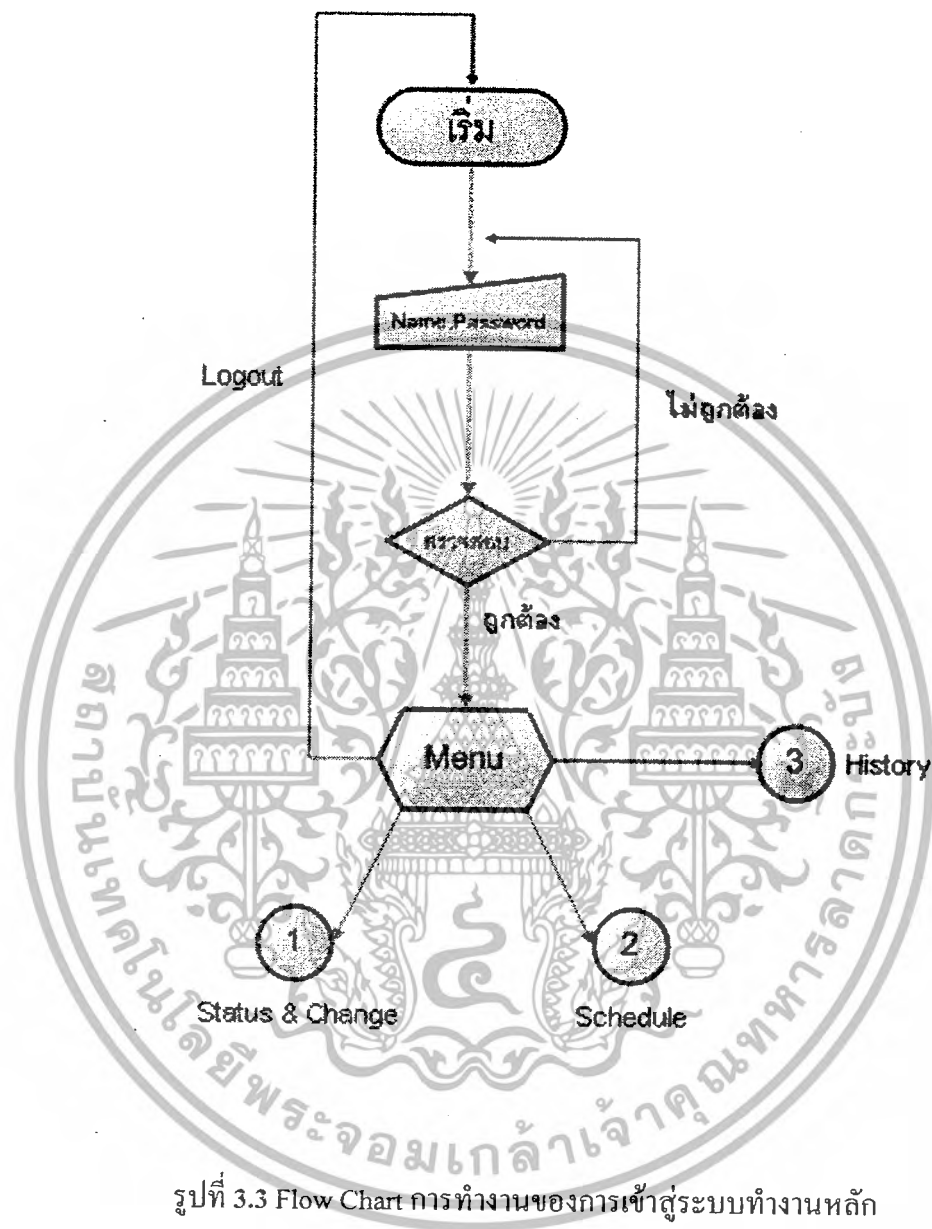
การทำงานของเว็บ จะมีคำสั่งต่างๆที่ให้ User เลือกใช้ในการควบคุมอุปกรณ์ไฟฟ้า และจะเก็บการทำงานไว้ใน history ดังรูปที่ 3.2



รูปที่ 3.2 ขั้นตอนการทำงานของ Web Application

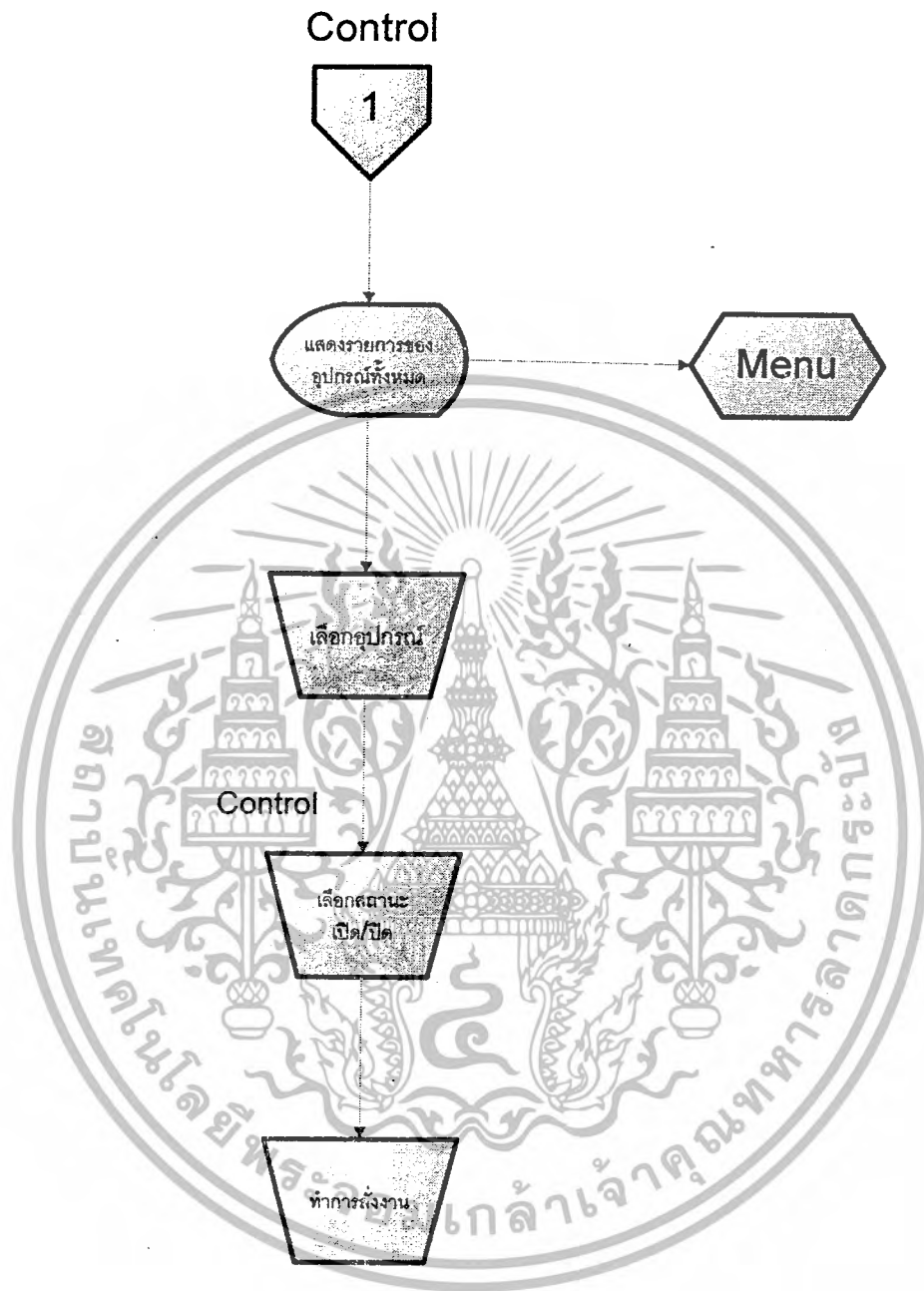
การทำงานของ การเข้าสู่ระบบการทำงานหลักจะต้องทำการ Log in โดยที่ต้องใส่ User Name และ Password ที่ถูกต้องจึงเข้าสู่ระบบการทำงานหลักได้ ดังรูปที่ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ส่วนการควบคุมจะแสดงสถานะของอุปกรณ์ไฟฟ้าปัจจุบัน แล้วทำการเลือกสถานะให้กับอุปกรณ์ตัวนั้นแล้วทำการสั่งงาน ดังรูปที่ 3.4

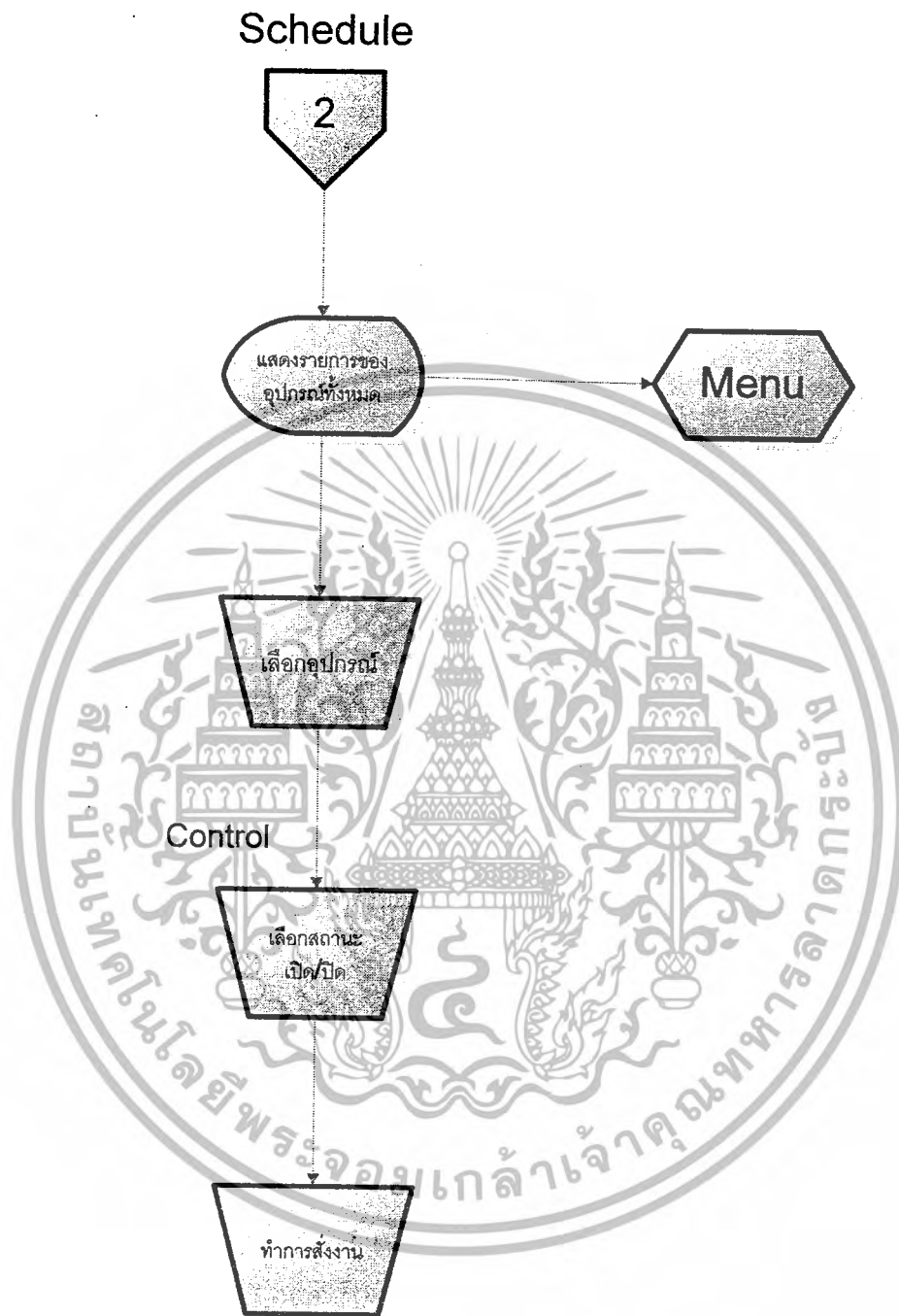
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 Flow Chart การทำงานในส่วนของการเปลี่ยนแปลงสถานะของ Device

การแสดงการทำงานในส่วนของ Schedule เป็นการแสดงสถานะปัจจุบันหลังมีการสั่งการทำงาน ดังรูปที่ 3.5

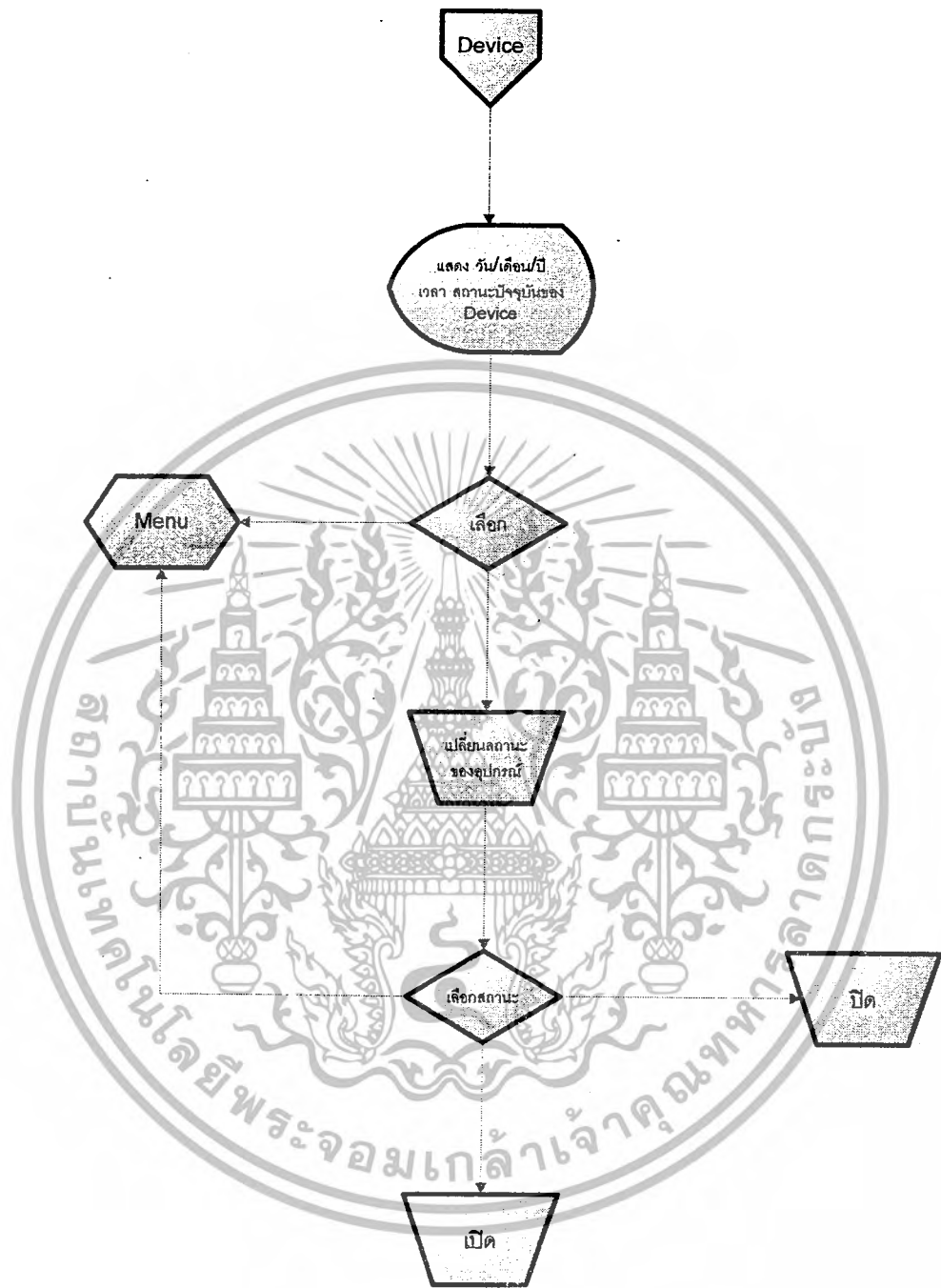
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Flow Chart การทำงานในส่วนของ Schedule

การทำงานในส่วนของการเปลี่ยนสถานะของ Device หลังจากที่ได้ทำการเลือกสถานะให้กับอุปกรณ์ไฟฟ้า ดังรูปที่ 3.6

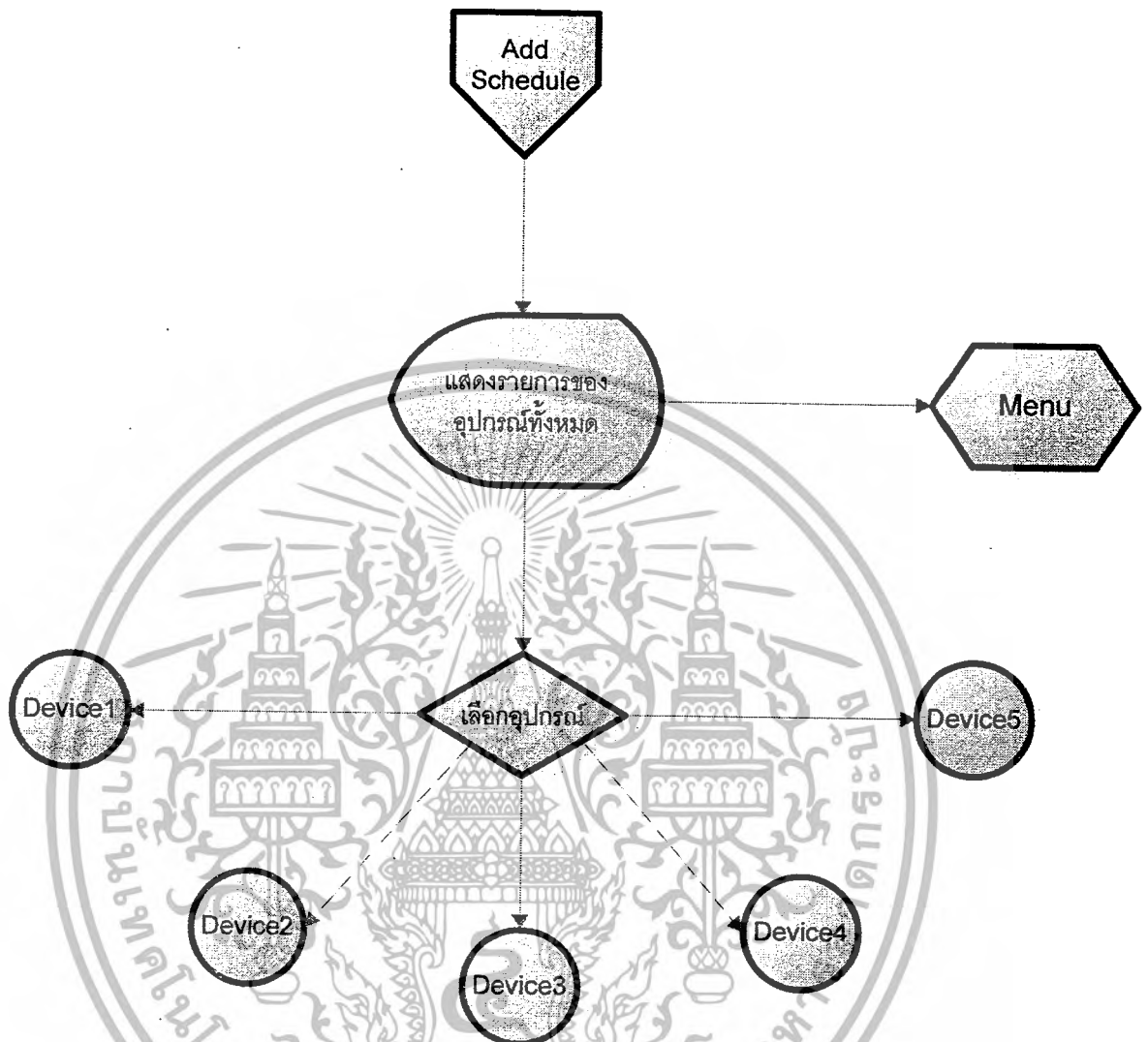
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Flow Chart การทำงานในส่วนของการเปลี่ยนสถานะของ Device

การทำงานในส่วนของการ Add Schedule เป็นการเลือกสถานะปัจจุบันหลังจากที่มีการเลือกสถานะให้กับอุปกรณ์ไฟฟ้า ดังรูปที่ 3.7

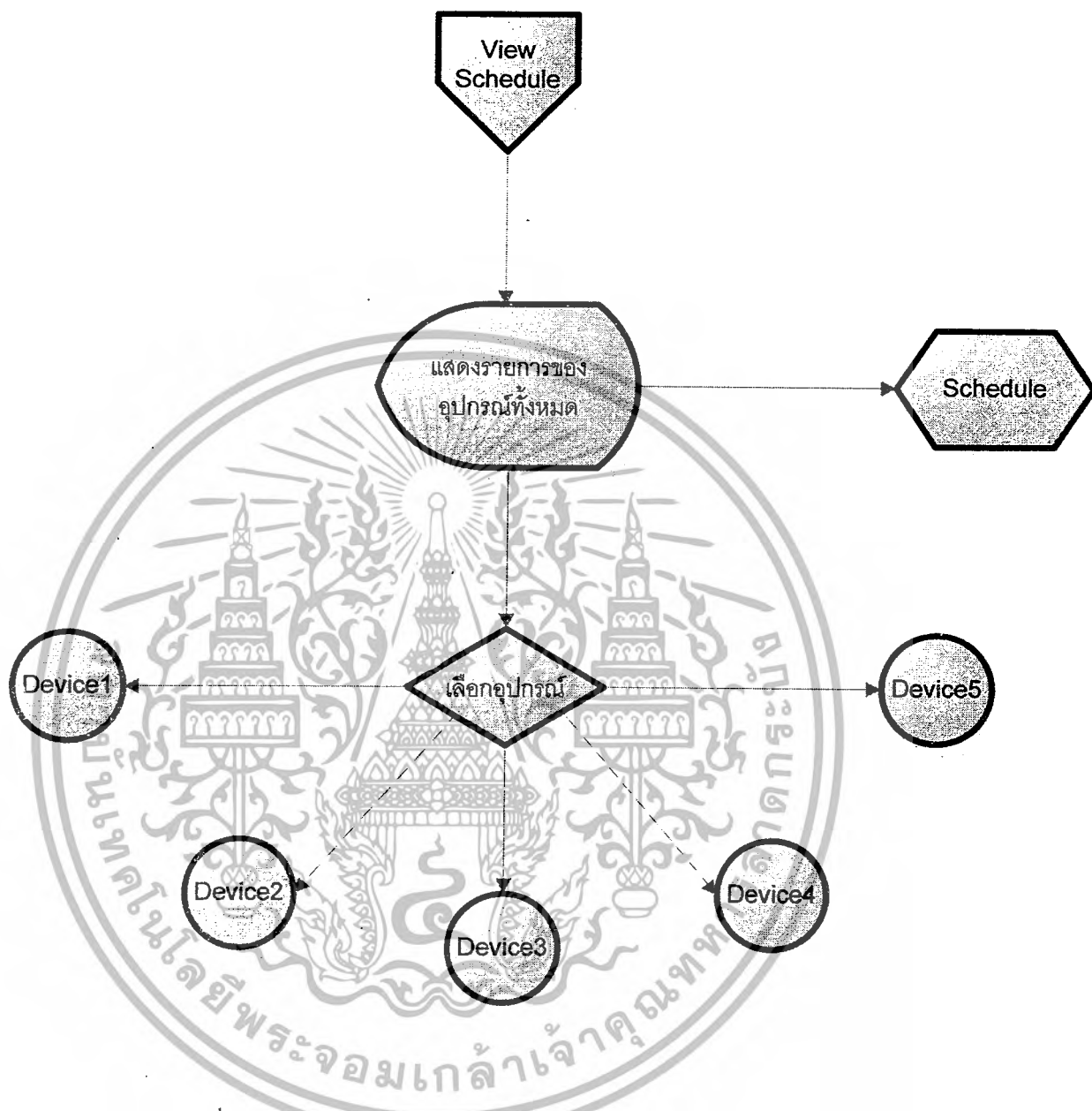
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 Flow Chart การทำงานในส่วนของการ Add Schedule

ก. ทำงานในส่วนของการ View Schedule เป็นการแสดงสถานะปัจจุบันหลัก จากที่มีคนเลือกสถานะให้กับ อุปกรณ์ไฟฟ้า ดังรูปที่ 3.8

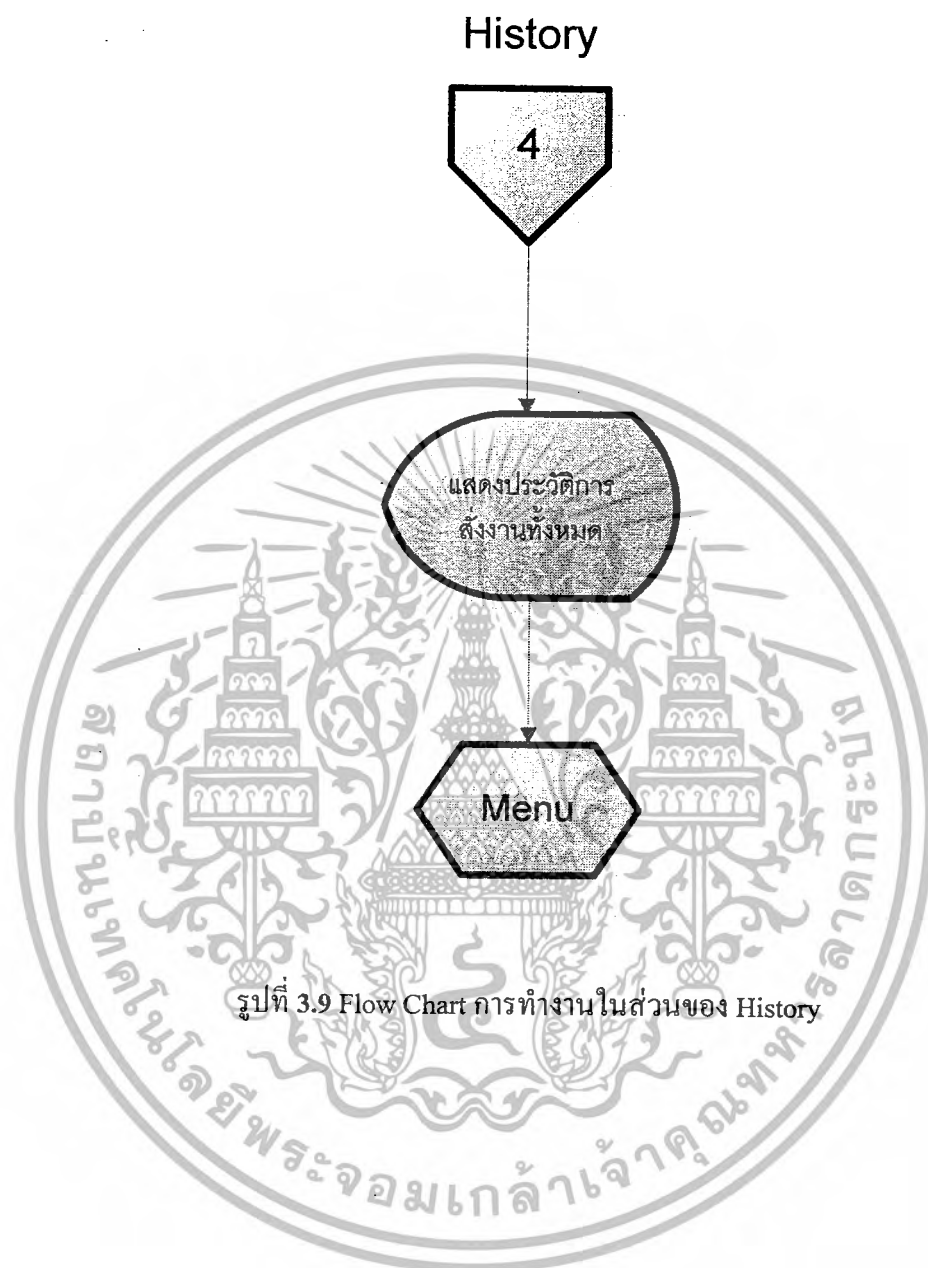
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Flow Chart การทำงานในส่วนของการ View Schedule

การทำงานในส่วนของ History เป็นการรวมข้อมูลการสั่งงานทั้งหมดที่ได้เลือกสถานะเปิดหรือปิดให้กับอุปกรณ์ไฟฟ้า ดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.5 การออกแบบฐานข้อมูล

ในโครงการนี้ถึงจะมีข้อมูลที่จะเก็บไม่มากนักจึงใช้ Access มาใช้ในการเก็บข้อมูลเพื่อสะดวกในการติดต่อและการส่งงานไม่ค่อยมีความยุ่งยาก อีกทั้งยังมีผู้นิยมใช้ร่วมกับ ASP.NET เป็นจำนวนมากและประเด็นสำคัญก็คือเป็นฟรีแวร์ซึ่งไม่ต้องเสียค่าใช้จ่ายในการใช้งานสามารถดาวน์โหลดจากอินเทอร์เน็ต ในการออกแบบ การใช้งานข้อมูลมีเพียงการเก็บข้อมูลสถานะ รายการคำสั่งเพียงเล็กน้อย ไม่ซับซ้อน มีการดึงข้อมูลออกมาโชว์ที่บราวเซอร์ และบันทึกเท่านั้น โดยแบ่งออกเป็นตารางทั้งหมด 7 ตารางดังนี้

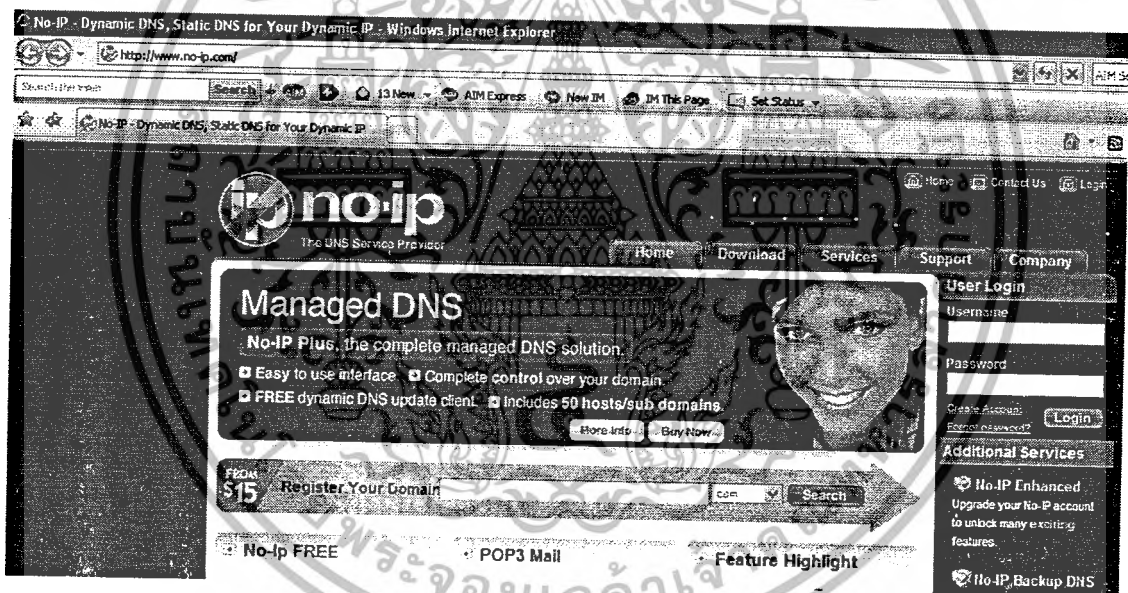
- Current Status มีฟิลด์ 2 ฟิลด์คือ Tool และ Status โดยกำหนดเป็นข้อมูลชนิด char ทั้งสองฟิลด์
- Device1 – Device5 รวม 5 ตารางมีฟิลด์ในแต่ละตาราง 5 ฟิลด์คือ Status , Day , Month , Year , Hour
- History มีฟิลด์ 4 ฟิลด์ คือ Tool, Status, Date, Time
- Date เป็นฟังก์ชันที่เก็บค่าของ วัน เดือน ปี พร้อมกันที่มีการกระทำต่อข้อมูลในฐานข้อมูลในที่นี้คือการ Insert ข้อมูลลงในฐานข้อมูล
- Time เป็นฟังก์ชันที่เก็บค่าของเวลาที่ เป็น ชั่วโมง นาที พร้อมกันที่มีการกระทำต่อข้อมูลในฐานข้อมูลในที่นี้คือการ Insert ข้อมูลลงในฐานข้อมูล

## บทที่ 4

### การทดลองและผลการทดลอง

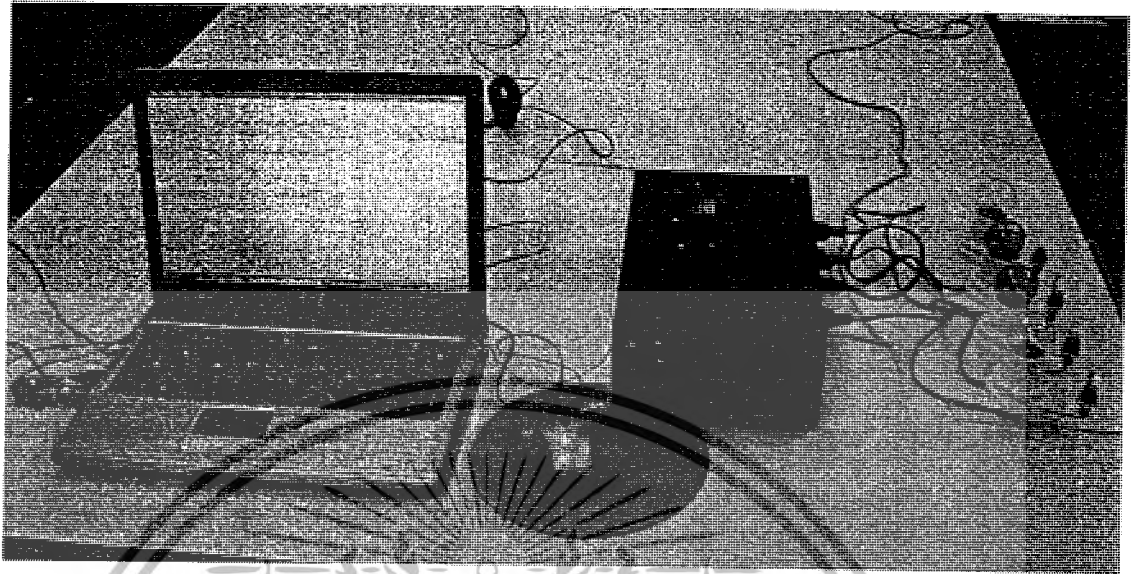
#### 4.1 ขั้นตอนการทดลอง

1. ทำการเขียน โปรแกรมลงในไมโครคอนโทรลเลอร์
2. จำลองเครื่องคอมพิวเตอร์เป็นเครื่องเซิร์ฟเวอร์โดยระบุเป็น NO-IP
3. สร้างหน้าเว็บเบราว์เซอร์จากโปรแกรม VB.NET
4. กำหนด Username และ Password ลงฐานข้อมูลโดยใช้ Access
5. ทำการเชื่อมต่อข้อมูลไปยังเซิร์ฟเวอร์
6. ทำการลงทะเบียนเพื่อขอใช้บริการ NO-IP Plus จาก www.no-ip.com
7. ทำการเชื่อมต่อผ่านเว็บแล้ว สามารถเปิด-ปิดไฟได้



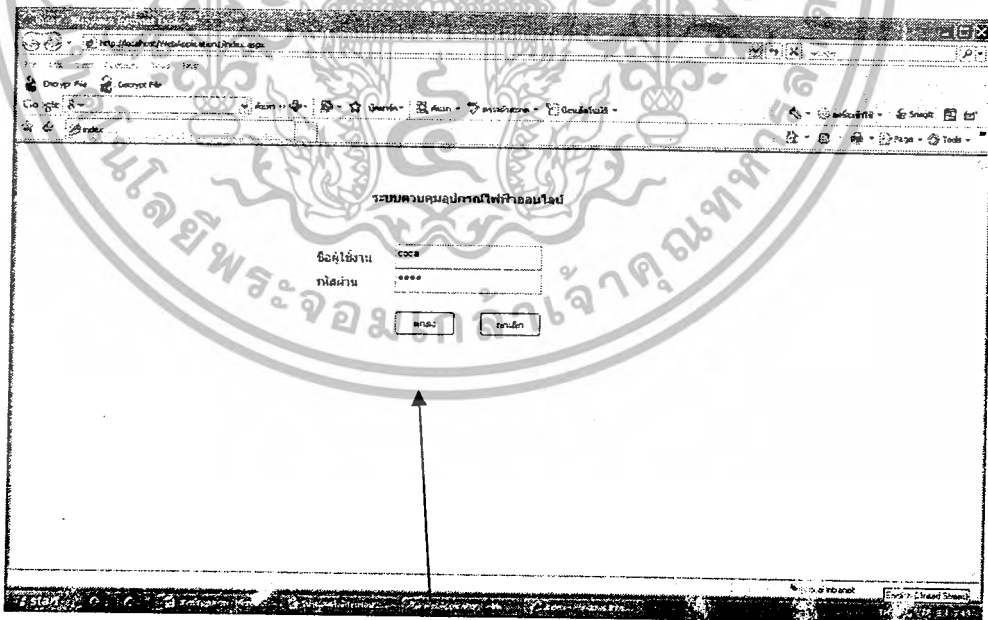
รูปที่ 4.1 หน้าหลักเว็บ no-ip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 การเชื่อมต่อของเครื่องเซิร์ฟเวอร์

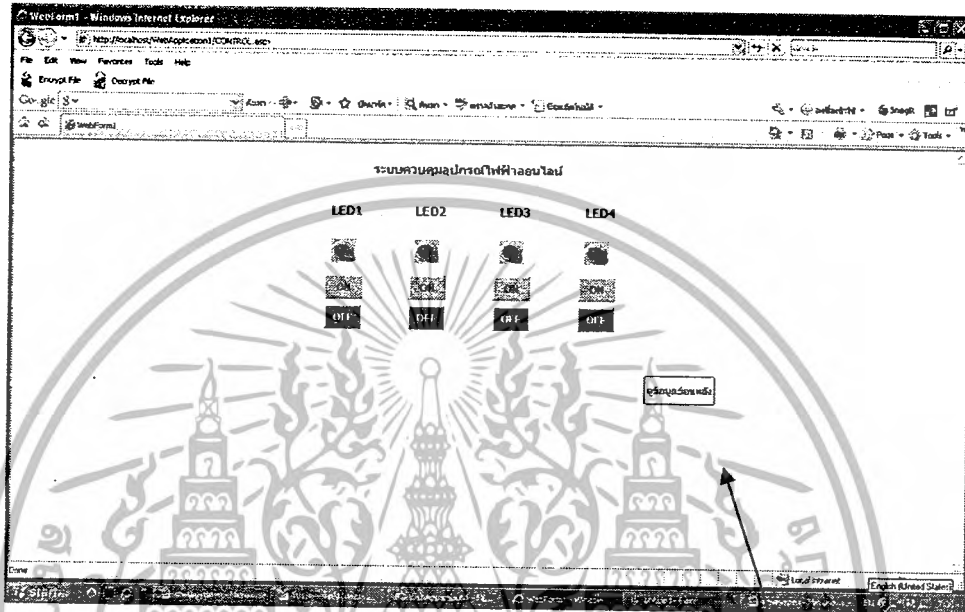
ทดลองเข้าเว็บจะปรากฏหน้าเว็บ Home Web Server



รูปที่ 4.3 แสดงหน้า index

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

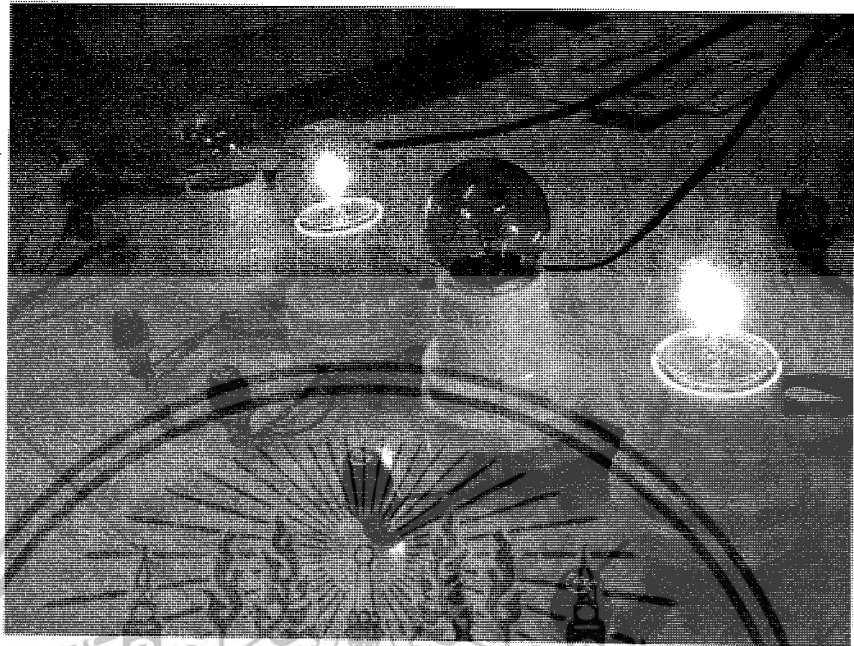
ระบบให้ทำการเข้าสู่ระบบผู้ใช้งาน เมื่อเข้าสู่หน้าควบคุมอุปกรณ์ไฟฟ้าออนไลน์ซึ่งหน้านี้เป็นหน้าในส่วนควบคุมอุปกรณ์ไฟฟ้า



รูปที่ 4.4 หน้าควบคุมอุปกรณ์ไฟฟ้าออนไลน์

เมื่อเข้าสู่ระบบควบคุมอุปกรณ์ไฟฟ้าออนไลน์ ในโครงการนี้ใช้หลอดไฟแทนอุปกรณ์ไฟฟ้าสามารถเปิด-ปิดได้ และสามารถดูข้อมูลย้อนหลังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 อุปกรณ์ไฟฟ้าสามารถเปิด-ปิด

ซึ่งเมื่อคู่มือข้อย้อนหลังแล้วสามารถกลับไปสู่โหมดควบคุมการใช้งานเริ่มต้นได้

ระบบข้อมูลย้อนหลัง

LED1	TIME	STATUS
LED1	10/2/2552 20:10:26	OFF
LED2	10/2/2552 20:10:26	OFF
LED3	10/2/2552 20:10:26	OFF
LED4	10/2/2552 20:10:26	ON
LED1	10/2/2552 20:11:55	ON
LED1	10/2/2552 21:13:39	ON
LED2	10/2/2552 21:13:46	ON
LED3	10/2/2552 21:13:50	OFF
LED4	10/2/2552 21:13:55	ON

กลับสู่หน้าจอโหมด

รูปที่ 4.6 หน้าระบบข้อมูลย้อนหลัง

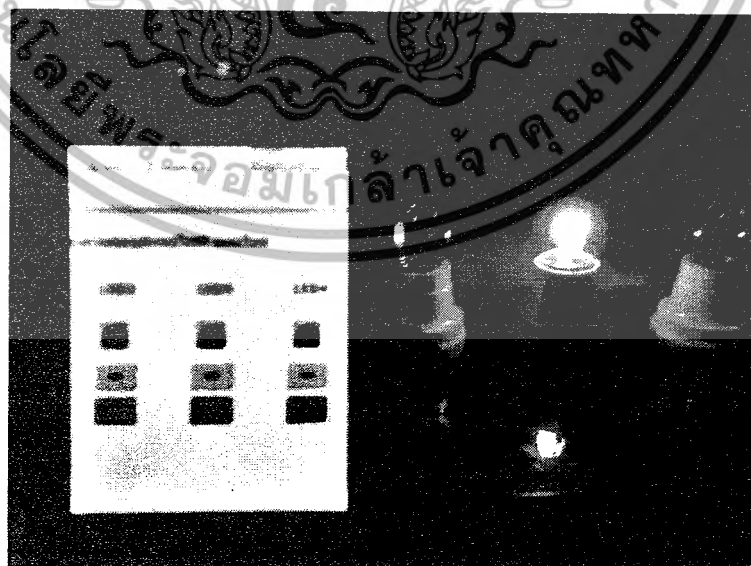
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ PDA Phone ทำการเข้าหน้า Log in และควบคุมอุปกรณ์ไฟฟ้าผ่าน GPRS/EDGE โดยใช้ web browser ในหน้า Log In ต้องใส่ User Name และ Password ให้ถูกต้อง



รูปที่ 4.7 หน้า index บน PDA Phone

ทำการ Log In เข้าสู่หน้าควบคุมอุปกรณ์ไฟฟ้า สามารถเลือกสถานะเปิด-ปิดอุปกรณ์ไฟฟ้าได้



รูปที่ 4.8 หน้าควบคุมอุปกรณ์ไฟฟ้าผ่าน PDA Phone และ การแสดงผลของเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปผลการทดลองและแนวทางในการพัฒนา

#### 5.1 สรุปผลการพัฒนาโครงการ

จุดมุ่งหมายของการพัฒนาโครงการ คือ สามารถควบคุมการเปิด - ปิด อุปกรณ์ไฟฟ้าภายในบริเวณที่พักอาศัย หรือ ออฟฟิศผ่านอินเทอร์เน็ตและ โทรศัพท์มือถือ (PDA Phone) รวมไปถึงการตั้งเวลาเปิด - ปิดอุปกรณ์ไฟฟ้า และ แสดงสถานะปัจจุบันของอุปกรณ์ไฟฟ้าว่ามีสถานะใด อีกทั้งสามารถตรวจเช็ค หรือ เรียกดูการสั่งงานย้อนหลังได้

จากการดำเนินการพัฒนาโครงการทำให้ได้รับความรู้และประสบการณ์ที่เพิ่มมากขึ้นในส่วนของ การศึกษาไมโครคอนโทรลเลอร์ MCS-51 และ วงจร Relay รวมไปถึงโปรแกรม และภาษาทางคอมพิวเตอร์ ที่ใช้ในการศึกษาโครงการนี้ อีกทั้งความพร้อมที่จะแก้ไขปัญหาที่เกิดขึ้น

#### 5.2 ปัญหาที่เกิดขึ้นระหว่างการพัฒนา

การใช้งานไมโครคอนโทรลเลอร์ MCS-51 และวงจร RELAY ต้องใช้เวลาในการศึกษาเป็น เวลาพอสมควร มีการเกิด Bug และ Error ในบางส่วนของโปรแกรม งบประมาณที่มีอยู่อย่างจำกัด รวมทั้ง ความไม่ชำนาญในการใช้งานอุปกรณ์ต่างๆของผู้จัดทำ

#### 5.3 แนวทางในการพัฒนาโครงการ

สามารถทำการพัฒนา จียูไอ (GUI) ให้มีการใช้งานที่ง่ายขึ้น รวมไปถึงเรื่องของรูปลักษณะของ ก่อตั้งอุปกรณ์ และการดึงเอาประสิทธิภาพของบอร์ด MCS-51 ออกมาใช้งานที่หลากหลายมากขึ้น

## บรรณานุกรม

ศุภชัย สมพานิช. 2549 Database Programming ด้วย 2005&VC#2005. นนทบุรี :  
ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์.

อภิชาติ ภูพลับ. 2546 เริ่มต้นเขียนโปรแกรมติดต่อควบคุมฮาร์ดแวร์ด้วย Visual Basic.  
นนทบุรี : ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์.

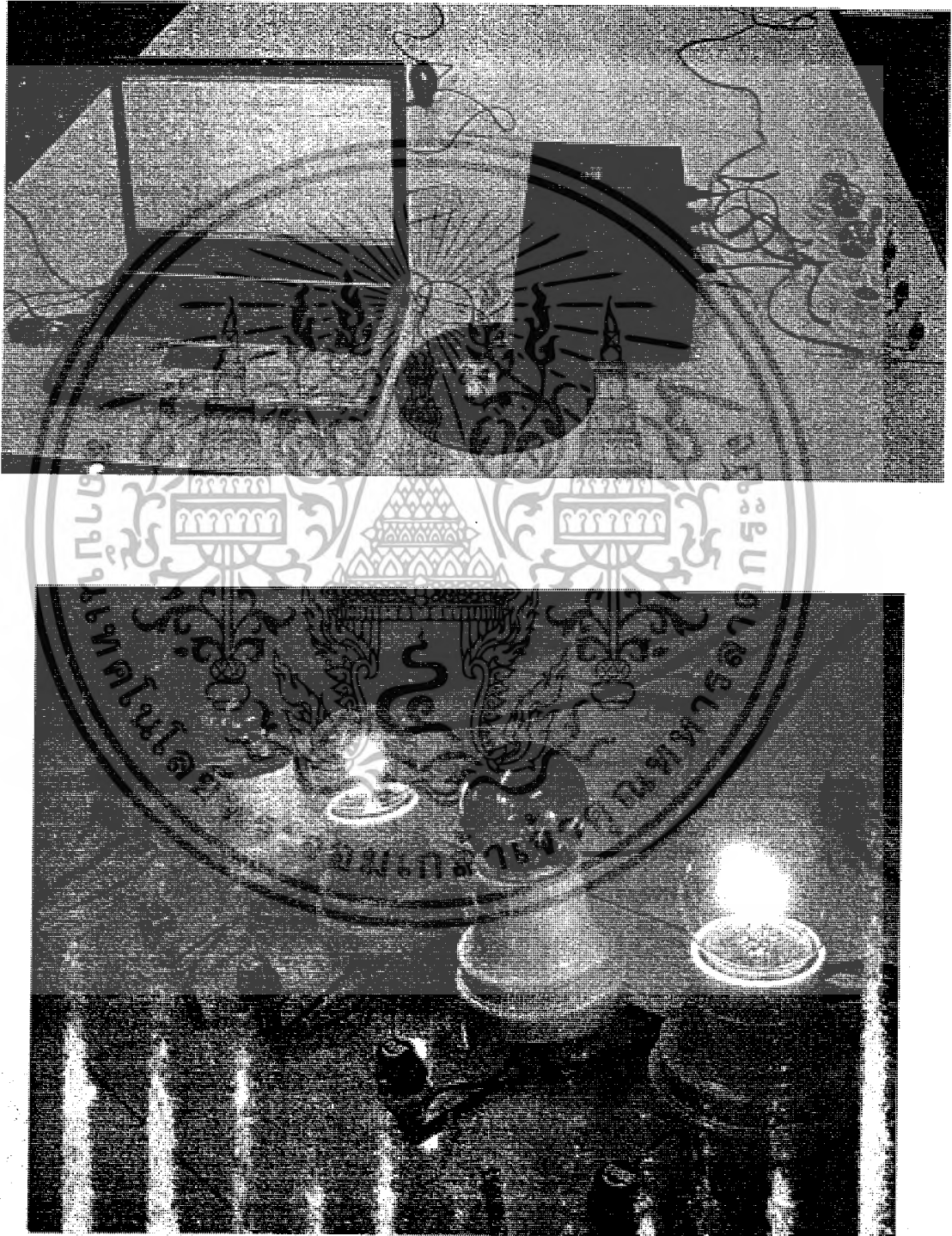
กลุ่มยุทธศาสตร์และสารสนเทศ สำนักส่งเสริมและพัฒนาการเกษตรเขตที่ 2,2551.

อดิศักดิ์ ชินะวงศ์. การใช้งาน Dreamweaver 8. [ออนไลน์]. เข้าถึงได้จาก [http://  
www.adisak51.com/index1.html](http://www.adisak51.com/index1.html) (วันที่ค้นข้อมูล : 20 ธันวาคม 2551)



## ภาคผนวก

ลักษณะการต่อวงจรเข้ากับเครื่องเซิร์ฟเวอร์ และไฟแสดงสถานะว่าอุปกรณ์พร้อมใช้งาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Visual Basic .Net ในหน้าการทำงานและควบคุมปุ่มสั่งการอุปกรณ์ เพื่อสั่งการอุปกรณ์เครื่องใช้ไฟฟ้า

```
Imports System.Data
```

```
Imports System.Data.OleDb
```

```
Public Class WebForm1
```

```
Inherits System.Web.UI.Page
```

```
#Region " Web Form Designer Generated Code "
```

```
'This call is required by the Web Form Designer.
```

```
<System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
```

```
End Sub
```

```
Protected WithEvents Label1 As System.Web.UI.WebControls.Label
```

```
Protected WithEvents Image1 As System.Web.UI.WebControls.Image
```

```
Protected WithEvents Image2 As System.Web.UI.WebControls.Image
```

```
Protected WithEvents Image3 As System.Web.UI.WebControls.Image
```

```
Protected WithEvents Image4 As System.Web.UI.WebControls.Image
```

```
Protected WithEvents Label2 As System.Web.UI.WebControls.Label
```

```
Protected WithEvents Label3 As System.Web.UI.WebControls.Label
```

```
Protected WithEvents Label4 As System.Web.UI.WebControls.Label
```

```
Protected WithEvents Label5 As System.Web.UI.WebControls.Label
```

```
Protected WithEvents Button13 As System.Web.UI.WebControls.Button
```

```
Protected WithEvents Button14 As System.Web.UI.WebControls.Button
```

```
Protected WithEvents Button15 As System.Web.UI.WebControls.Button
```

```
Protected WithEvents Button16 As System.Web.UI.WebControls.Button
```

```
Protected WithEvents Button29 As System.Web.UI.WebControls.Button
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Protected WithEvents Button30 As System.Web.UI.WebControls.Button

Protected WithEvents Button31 As System.Web.UI.WebControls.Button

Protected WithEvents Button32 As System.Web.UI.WebControls.Button

Protected WithEvents Button1 As System.Web.UI.WebControls.Button

Protected WithEvents Button2 As System.Web.UI.WebControls.Button

'NOTE: The following placeholder declaration is required by the Web Form Designer.

'Do not delete or move it.

Private designerPlaceholderDeclaration As System.Object

Private Sub Page\_Init(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles MyBase.Init

'CODEGEN: This method call is required by the Web Form Designer

'Do not modify it using the code editor.

InitializeComponent()

End Sub

#End Region

Dim strConn As String = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & Server.MapPath("control.mdb") & ""

Private Sub Page\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)

Handles MyBase.Load

Dim Sql As String

Sql = "SELECT \* FROM control1"

Dim MyConn As New OleDbConnection(strConn)

MyConn.Open()

Dim mReader As OleDbDataReader

Dim mCmd As OleDbCommand = New OleDbCommand(Sql, MyConn)

mReader = mCmd.ExecuteReader

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim a As String
mReader.Read()
a = Convert.ToString(mReader.Item("data1"))
    If a = "0" Then
        Me.Image1.ImageUrl = "led_off.jpg"
    ElseIf a = "1" Then
        Me.Image1.ImageUrl = "led_on.jpg"
    End If
a = Convert.ToString(mReader.Item("data2"))
    If a = "0" Then
        Me.Image2.ImageUrl = "led_off.jpg"
    ElseIf a = "1" Then
        Me.Image2.ImageUrl = "led_on.jpg"
    End If
a = Convert.ToString(mReader.Item("data3"))
    If a = "0" Then
        Me.Image3.ImageUrl = "led_off.jpg"
    ElseIf a = "1" Then
        Me.Image3.ImageUrl = "led_on.jpg"
    End If
a = Convert.ToString(mReader.Item("data4"))
    If a = "0" Then
        Me.Image4.ImageUrl = "led_off.jpg"
    ElseIf a = "1" Then
        Me.Image4.ImageUrl = "led_on.jpg"
    End If

MyConn.Close()

End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Button13_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button13.Click
```

```
Dim MySQL As String = "UPDATE control1 SET data4='1'"
```

```
Dim MyConn As New OleDbConnection(strConn)
```

```
Dim cmd As New OleDbCommand(MySQL, MyConn)
```

```
MyConn.Open()
```

```
cmd.ExecuteNonQuery()
```

```
MyConn.Close()
```

```
Me.Image4.ImageUrl = "led_on.jpg"
```

```
End Sub
```

```
Private Sub Button14_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button14.Click
```

```
Dim MySQL As String = "UPDATE control1 SET data3='1'"
```

```
Dim MyConn As New OleDbConnection(strConn)
```

```
Dim cmd As New OleDbCommand(MySQL, MyConn)
```

```
MyConn.Open()
```

```
cmd.ExecuteNonQuery()
```

```
MyConn.Close()
```

```
Me.Image3.ImageUrl = "led_on.jpg"
```

```
End Sub
```

```
Private Sub Button15_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
```

```
Handles Button15.Click
```

```
Dim MySQL As String = "UPDATE control1 SET data2='1'"
```

```
Dim MyConn As New OleDbConnection(strConn)
```

```
Dim cmd As New OleDbCommand(MySQL, MyConn)
```

```
MyConn.Open()
```

```
cmd.ExecuteNonQuery()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

MyConn.Close()
Me.Image2.ImageUrl = "led_on.jpg"
End Sub

```

```

Private Sub Button16_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button16.Click

```

```

    Dim MySQL As String = "UPDATE control1 SET status2='1'"
    Dim MyConn As New OleDbConnection(strConn)
    Dim cmd As New OleDbCommand(MySQL, MyConn)
    MyConn.Open()
    cmd.ExecuteNonQuery()
    MyConn.Close()
    Me.Image1.ImageUrl = "led_on.jpg"
End Sub

```

```

Private Sub Button29_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button29.Click

```

```

    Dim MySQL As String = "UPDATE control1 SET data4='0'"
    Dim MyConn As New OleDbConnection(strConn)
    Dim cmd As New OleDbCommand(MySQL, MyConn)
    MyConn.Open()
    cmd.ExecuteNonQuery()
    MyConn.Close()
    Me.Image4.ImageUrl = "led_off.jpg"
End Sub

```

```

Private Sub Button30_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)

```

```

Handles Button30.Click

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Dim MySQL As String = "UPDATE controll1 SET data3='0'"
Dim MyConn As New OleDbConnection(strConn)
Dim cmd As New OleDbCommand(MySQL, MyConn)
MyConn.Open()
cmd.ExecuteNonQuery()
MyConn.Close()
Me.Image3.ImageUrl = "led_off.jpg"
End Sub

```

```

Private Sub Button31_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button31.Click
Dim MySQL As String = "UPDATE controll1 SET data2='0'"
Dim MyConn As New OleDbConnection(strConn)
Dim cmd As New OleDbCommand(MySQL, MyConn)
MyConn.Open()
cmd.ExecuteNonQuery()
MyConn.Close()
Me.Image2.ImageUrl = "led_off.jpg"
End Sub

```

```

Private Sub Button32_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button32.Click
Dim MySQL As String = "UPDATE controll1 SET data1='0'"
Dim MyConn As New OleDbConnection(strConn)
Dim cmd As New OleDbCommand(MySQL, MyConn)
MyConn.Open()
cmd.ExecuteNonQuery()
MyConn.Close()
Me.Image1.ImageUrl = "led_off.jpg"
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
    Response.Redirect("historyLED.aspx")
End Sub
```

```
Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
    Response.Redirect("webtime.aspx")
End Sub
End Class
```

