

ระบบควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านโครงข่าย
NETWORK CONTROL SYSTEM



โดย

นายธนกฤต ยິงรัมย์
นางสาวธิราภรณ์ โล่ห์วิสัย
นายนราทิช ณ ลำปาง



ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

Handwritten notes in blue ink

เลขหมู่.....
เลขทะเบียน... 46471
วัน, เดือน, ปี... 2 เม.ย. 2546

Box with fields .b..... and .i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านโครงข่าย
NETWORK CONTROL SYSTEM

โดย

นายธนกฤต ยิ่งรัมย์ 41014165

นางสาวธิราภรณ์ โล่ห์วิสัย 41014188

นายนราทิช ณ ลำปาง 41014211

อาจารย์ที่ปรึกษา

รศ. สมยศ จุณณะปิยะ

อ. นภัทร สระเอี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2544

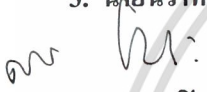
ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง ระบบควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านโครงข่าย

NETWORK CONTROL SYSTEM

ผู้จัดทำ

1. นายชนกฤต ยิ่งรัมย์ 41014165
2. นางสาวธิราภรณ์ โล่ห์วิสัย 41014188
3. นายนราทิช ณ ลำปาง 41014211


(รศ. สมยศ จุณณะปิยะ)


(อ. นภัทร สระเอี่ยม)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านโครงข่าย
NETWORK CONTROL SYSTEM

โดย นายชนกฤต ยิ่งรัมย์ 41014165
นางสาวชริราภรณ์ โล่ห์วีลีย์ 41014188
นายนราทิช ฅ ลำปาง 41014211

อาจารย์ที่ปรึกษา รศ. สมยศ จุณณะปี
และ อ. นภัทร สระเอี่ยม

บทคัดย่อ

โครงการที่จะนำเสนอนี้ เป็นการศึกษาและสร้างระบบควบคุมอุปกรณ์อิเล็กทรอนิกส์ผ่านโครงข่ายคอมพิวเตอร์ โดยการทำงานคอมพิวเตอร์ต้นทางจะติดต่อไปยังคอมพิวเตอร์ปลายทางเพื่อที่จะควบคุมอุปกรณ์อิเล็กทรอนิกส์ พร้อมทั้งแสดงสถานะและรายงานผลการทำงานกลับไป

ABSTRACT

This project presents a design of Network control system. The system uses the computer connection to control, check status of Electronic equipment and report the result back.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ	I
สารบัญ	II
สารบัญรูป	III
	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 Visual Basic	3
2.1.1 คำสั่งสำหรับเขียนโปรแกรม	3
2.1.2 ขอบเขตการใช้งานของตัวแปร	4
2.1.3 ฟังก์ชันสำหรับเขียนโปรแกรม	4
2.1.4 การเขียนโปรแกรมกับ Control	5
2.1.5 Method ที่ใช้ในการจัดเก็บข้อมูล	7
2.1.6 Data Environment Designer	8
2.1.7 การประยุกต์ใช้งาน Visual Basic	10
2.2 Assembly	11
2.2.1 การใช้งานไมโครคอนโทรลเลอร์ MCS-51	11
2.2.2 การใช้งานพอร์ตอนุกรม	11
2.2.3 การกำหนดเริ่มต้นและการใช้งานรีจิสเตอร์ของพอร์ตอนุกรม	15
2.2.4 การใช้งานไทม์เมอร์ / เคา์นเตอร์	16
2.2.5 การอินเตอร์เฟสแบบอนุกรม RS-232	17
2.3 ASP และ WAP	18
2.3.1 ASP	18
2.3.1.1 Active Server Page	18
2.3.1.2 The Active Server Page Model & Architecture	19
2.3.1.3 ADO Object Model	21
2.3.2 WAP	22
2.3.2.1 มารู้อักกับ WAP	22
2.3.2.2 ODBC กับ WAP	27
2.3.2.3 การส่งข้อมูลเข้าหา WAP	29
บทที่ 3 ขั้นตอนการสร้างและการดำเนินงาน	31
3.1 แนวคิดและการดำเนินงาน	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
- BLOCK DIAGRAM	31
- FLOW CHART	34
3.2 การสร้างวงจรควบคุมอุปกรณ์ในบ้าน	37
3.2.1 การสร้างวงจรและอุปกรณ์	37
3.2.2 ชิ้นงาน	40
3.3 การสร้างซอฟต์แวร์เพื่อใช้ในการควบคุม	44
3.3.1 Source Code เขียนด้วยภาษา Assembly	44
3.3.1.1 Source Code	44
3.3.1.2 ขั้นตอนการใช้งาน	44
3.3.2 โปรแกรม Project 1 เขียนด้วย Visual Basic 6.0	45
3.3.3 WAP และ ASP	61
บทที่ 4 การทดลองและผลการทดลอง	65
4.1 ขั้นตอนการปฏิบัติ	65
4.1.1 ขั้นตอนการปฏิบัติในการใช้งานอุปกรณ์	65
4.1.2 การติดตั้งอุปกรณ์	66
4.2 ผลการทดลองการควบคุมจากคอมพิวเตอร์ภายในบ้าน	69
4.3 ผลการทดลองการควบคุมโดยผ่านอินเทอร์เน็ต	76
4.4 การแสดงการใช้งานอุปกรณ์ด้วย DATA REPORT	83
4.5 การติดต่อสื่อสารระหว่าง Client - Server	84
บทที่ 5 บทวิจารณ์และบทสรุป	85
ภาคผนวก	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
บทที่ 2	
รูปที่ 2.1 การทำงานของ WAP	23
รูปที่ 2.2 การใช้โปรแกรมจำลองในการชม WAP	24
รูปที่ 2.3 แสดงการให้บริการ WAP	25
รูปที่ 2.4 ระบบเครือข่ายของ WAP	26
รูปที่ 2.5 การสร้าง ODBC ใน Control Panel	28
รูปที่ 2.6 แสดงการเข้าถึงข้อมูลของ WAP	29
บทที่ 3	
รูปที่ 3.1 Block Diagram ของระบบทั้งหมด	32
รูปที่ 3.2 แสดง Block Diagram การทำงานของระบบควบคุมที่บ้าน	33
รูปที่ 3.3 แสดง Block Diagram การทำงานของระบบควบคุมผ่านอินเทอร์เน็ต	33
รูปที่ 3.4 Flow Chart ของระบบควบคุมที่บ้าน	34
รูปที่ 3.5 Flow Chart ของระบบควบคุมผ่านอินเทอร์เน็ต 1	35
รูปที่ 3.6 Flow Chart ของระบบควบคุมผ่านอินเทอร์เน็ต 2	36
รูปที่ 3.7 แผนผังวงจร	38
รูปที่ 3.8 ลายปริ้นท์	39
รูปที่ 3.9 แสดงด้านหน้าของเครื่องควบคุมอุปกรณ์	40
รูปที่ 3.10 แสดงด้านหลังของเครื่องควบคุมอุปกรณ์	41
รูปที่ 3.11 แสดงภายในของเครื่องควบคุมอุปกรณ์ (ด้านบน)	42
รูปที่ 3.12 แสดงภายในของเครื่องควบคุมอุปกรณ์ (ด้านข้าง)	43
รูปที่ 3.13 แสดงหน้าจอของ โปรแกรม Login	45
รูปที่ 3.14 แสดงหน้าจอของเลือกโหมด Select Form	46
รูปที่ 3.15 แสดงหน้าจอของ โปรแกรม HOUSE CONTROLLER	47
รูปที่ 3.16 แสดงหน้าจอการกำหนดเปิด - ปิดหลอดไฟ	48
รูปที่ 3.17 แสดงการยืนยันการทำงาน CONFIRM FORM	50
รูปที่ 3.18 แสดงการนับเวลาถอยหลังเพื่อปิดอุปกรณ์ OPEN - CLOSE FORM	52
รูปที่ 3.19 แสดงสถานะของการทำงานของอุปกรณ์ PROCESS FORM	55
รูปที่ 3.20 แสดงการตรวจจับสถานะการเปลี่ยนแปลงข้อมูล WATCHING FORM	58
รูปที่ 3.21 แสดงโปรแกรม NETWORK	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 3.22 แสดงโปรแกรม CHAT	60
รูปที่ 3.23 แสดงหน้าต่างของโปรแกรม WAP	62
บทที่ 4	
รูปที่ 4.1 แสดงการเชื่อมต่อ CONNECTOR สำหรับติดต่อกับเครื่องคอมพิวเตอร์	65
รูปที่ 4.2 แสดงการต่อสาย DB9 เข้ากับ Serial Port ของคอมพิวเตอร์	66
รูปที่ 4.3 แสดงการทำงานของเครื่องควบคุมอุปกรณ์	67
รูปที่ 4.4 แสดงการทดลองการ LOGIN	68
รูปที่ 4.5 แสดงการทดลองการ เลือกโหมดการทำงาน HOUSE CONTROLLER	68
รูปที่ 4.6 แสดงการทดลองการ ทำงาน HOUSE CONTROLLER	69
รูปที่ 4.7 แสดงการทดลองการ กำหนดเวลาให้หลอดไฟฟ้า	69
รูปที่ 4.8 แสดงการทดลองการ ยืนยันการกำหนดเวลาให้หลอดไฟฟ้า	70
รูปที่ 4.9 แสดงการเปลี่ยนแปลงของข้อมูลในฐานข้อมูล	70
รูปที่ 4.10 แสดงการทดลองการนับเวลาถอยหลังให้หลอดไฟฟ้า	71
รูปที่ 4.11 แสดงการทดลองเมื่อ Microcontroller สั่งงานให้ Relay ทำงาน	71
รูปที่ 4.12 แสดงการทำงานของหลอดไฟฟ้า	72
รูปที่ 4.13 แสดงสถานะบน PROCESS FORM ของหลอดไฟฟ้า	72
รูปที่ 4.14 แสดงเวลาการนับถอยหลังของ OPEN – CLOSE FORM	73
รูปที่ 4.15 แสดงการทำงานของหลอดไฟฟ้า	73
รูปที่ 4.16 แสดงสถานะบน PROCESS FORM ของหลอดไฟฟ้า	74
รูปที่ 4.17 หน้าตาของโปรแกรม Nokia Mobile Internet Toolkit 3.0	75
รูปที่ 4.18 แสดงการเปลี่ยนแปลงของข้อมูลจากการบันทึกของ WAP	79
รูปที่ 4.19 แสดงสถานะบน WATCHING FORM ที่ตรวจจับเจอ	79
รูปที่ 4.20 แสดงการนับเวลาถอยหลังของ OPEN – CLOSE FORM	80
รูปที่ 4.21 แสดงสถานะบน PROCESS FORM ของทีวี	80
รูปที่ 4.22 แสดงการทำงานของ OPEN – CLOSE FORM	81
รูปที่ 4.23 แสดงสถานะบน PROCESS FORM ของทีวี	81
รูปที่ 4.24 แสดงรายงานการใช้อุปกรณ์ โดย DATA REPORT	82
รูปที่ 4.25 แสดงการพุดคุยกันระหว่างเครื่อง	83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ที่มาของโครงการ

ปัจจุบันเทคโนโลยีด้านการสื่อสารได้รุดหน้าไปอย่างรวดเร็ว ซึ่งมีอิทธิพลต่อการเปลี่ยนแปลงวิถีชีวิตของมนุษย์อย่างมากทีเดียว โทรศัพท์เคลื่อนที่ (Mobile) กลายเป็นปัจจัยที่สำคัญอย่างหนึ่งในการดำรงชีวิตประจำวัน

แนวโน้มนี้เองที่นำไปสู่การพัฒนาการใช้งาน โทรศัพท์เคลื่อนที่ให้เกิดประสิทธิภาพสูงสุด ดังเช่นเทคโนโลยีในปัจจุบัน เช่น Bluetooth ซึ่งสามารถควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ได้โดยผ่านโทรศัพท์เคลื่อนที่ แนวโน้มของอินเทอร์เน็ต (Internet) ผ่านโทรศัพท์เคลื่อนที่ และสิ่งอำนวยความสะดวกมากมาย รองรับแนวโน้มเทคโนโลยีและวิถีชีวิตของอนาคต จึงเป็นที่มาของโครงการนี้ ซึ่งสามารถใช้อินเทอร์เน็ตควบคุมอุปกรณ์ไฟฟ้าภายในบ้านโดยผ่านอินเทอร์เน็ต ได้แก่ บริการ World Wide Web และโทรศัพท์เคลื่อนที่ โดยใช้ WAP ในการทำงาน

ขอบเขตของโครงการ

1. เพื่อศึกษาการควบคุมอุปกรณ์ไฟฟ้า ซึ่งทำการเปิด-ปิด Relay โดยอาศัยการควบคุมของ Microcontroller
2. ศึกษาและทดลองการเชื่อมต่อและใช้งานพร้อมกันระหว่างโปรแกรม Visual Basic กับ Microcontroller ผ่านพอร์ตอนุกรม RS-232
3. ศึกษาและทดลองการเชื่อมต่อและใช้งานร่วมกันระหว่างโปรแกรม Visual Basic กับ HTML (ASP,WAP) โดยอาศัยการทำงานแบบเครือข่าย Client-Server

ขั้นตอนการดำเนินงาน

1. แบ่งโครงการออกเป็น 3 ส่วน
 - 1.1 Assembly ใช้ควบคุม Microcontroller
 - 1.2 Visual Basic ใช้ควบคุมเครื่องคอมพิวเตอร์
 - 1.3 HTML, ASP, WAP การทำงานผ่านอินเทอร์เน็ต ผ่านเครือข่ายแบบ Client-Server
2. ออกแบบวงจรและเขียน Flow Chart
3. ออกแบบ Block Diagram ของระบบ
4. ดำเนินการเขียน Assembly และทดลองใช้งานกับ Microcontroller (AT89C51)
5. เขียนโปรแกรม Visual Basic ชื่อว่า "Project 1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ทดสอบการทำงานระหว่าง Visual Basic และ Microcontroller ผ่าน RS-232 เก็บบันทึกผลข้อมูล
7. ทำการปรับปรุงโปรแกรม Project 1 ให้มีประสิทธิภาพโดยอาศัยระบบการตั้งเวลาและสั่งปิดอัตโนมัติ สามารถทำการ chat ระหว่างเครื่อง Client-Server ได้
8. เขียนโปรแกรม ASP และ WAP
9. ทดสอบการทำงานของ ASP และ WAP ในการเปลี่ยนแปลงฐานข้อมูลกับเครื่อง Server โดยอาศัยโปรแกรม PWS ในการทำงาน
10. ทดสอบการทำงานร่วมกันระหว่าง WAP และ Visual Basic
11. ทำการทดสอบระบบทั้งหมด
 - 11.1 ระหว่าง Visual Basic กับ Microcontroller
 - 11.2 การทำงานร่วมกันทั้ง WAP, Visual Basic และ Microcontroller
12. ลงโปรแกรม Windows 2000 เพื่อทำการเซต Server จริงโดยอาศัยโปรแกรม IIS

ประโยชน์ที่ได้รับจากโครงการ

1. ได้รับความรู้ทางด้านวงจรไฟฟ้า และอิเล็กทรอนิกส์
2. ทำให้ทราบขั้นตอนการทำงานของ Microcontroller
3. ทำให้ทราบการทำงานและการเชื่อมต่อของ Microcontroller กับเครื่องคอมพิวเตอร์ผ่าน Serial Port
4. ทำให้ทราบหลักการการทำงานของ โปรแกรมภาษา Visual Basic
5. ทำให้ทราบถึงการทำงานของระบบเครือข่ายแบบ Client – Server
6. ทำให้ทราบถึงหลักการและมาตรฐานในการติดต่อสื่อสารแบบโปรโตคอล TCP/IP
7. ทำให้ทราบหลักการการทำงานของ Application ประยุกต์กับอินเทอร์เน็ต เช่น ASP, WAP
8. ทำให้ทราบถึงความแตกต่างของการทำงานระหว่าง PWS และ IIS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 Visual Basic

2.2 Assembly

2.3 ASP และ WAP

2.1 Visual Basic

2.1.1 คำสั่งสำหรับเขียนโปรแกรม

การ Declare คำตัวแปร

คำตัวแปรต่าง ๆ ที่ใช้ใน Visual Basic จะต้องมีกำหนดประเภทของตัวแปรนั้น โดยทำได้ 2 วิธี คือ

วิธีที่ 1 *Explicit Declaration* กำหนดประเภทของตัวแปรโดยใช้คำสั่ง

```
Dim varname [As type] [,varname [As type]]...
```

โดยที่ varname หมายถึง ชื่อตัวแปร
type หมายถึง ประเภทข้อมูล

การกำหนดในลักษณะ สามารถกำหนดได้ทั้งในส่วน General และต้นโปรแกรมของแต่ละ Procedure หรือฟังก์ชัน แต่ทั้ง 2 วิธีจะมีข้อแตกต่างกันคือ ตัวแปรที่ Declare ไว้ในส่วน General จะเป็นตัวแปรส่วนกลางที่ Procedure หรือฟังก์ชันต่าง ๆ ภายใน Form เดียวกันสามารถเรียกใช้งานได้ แต่กรณีที่ Declare ไว้ใน Procedure หรือฟังก์ชัน ตัวแปรที่ Declare ไว้ จะใช้ได้ภายใน Procedure หรือฟังก์ชันที่มีการ Declare ตัวแปรนั้นไว้เท่านั้น จะไม่สามารถเรียกใช้ใน Procedure หรือฟังก์ชันอื่นได้เนื่องจากจะถือว่าเป็นตัวแปรคนละตัวกัน ถึงแม้จะมีชื่อเดียวกันก็ตาม

สำหรับ type จะกำหนดตามประเภทข้อมูล หรืออาจกำหนดเป็นประเภทของ Object ที่ตัวแปรนั้นจะทำหน้าที่แทน หรือไม่กำหนด ในกรณีที่ไมกำหนด จะถือว่าเป็นตัวแปรที่ใช้เก็บข้อมูลประเภท Variant และคำสั่งในการ Declare จะเหลือเพียงคำสั่ง Declare และตามด้วยชื่อตัวแปร ดังนี้

```
Dim varname
```

เช่น Dim TempVar ซึ่งในที่นี้ตัวแปร TempVar จะเป็นตัวแปรที่ใช้เก็บข้อมูลประเภท Variant

วิธีที่ 2 *Implicit Declaration* การกำหนดประเภทของตัวแปรจะทำอยู่ในรูป

```
Varname <Type Identifier>
```

เช่น Mradius!, Vamount@ เป็นต้น ซึ่งการกำหนดประเภทตัวแปรลักษณะนี้ ตัวแปรนั้นจะต้องเขียนในลักษณะนี้ตลอดการใช้งานใน Procedure นั้น และจะถือว่าเป็นตัวแปรที่ถูกกำหนดให้ใช้งานเฉพาะใน Procedure หรือฟังก์ชันเท่านั้น

สำหรับการกำหนดค่าตัวแปร Constant ให้ใช้ประโยคคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้เข้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[Public 1 Private] Const constname = expression

โดยที่ constname หมายถึง ชื่อตัวแปร Constant

expression หมายถึง ค่าที่กำหนดให้กับตัวแปร Constant

คำสั่ง Public (ค่า Default) ใช้ในกรณีที่ต้องการให้ตัวแปร Constant ที่กำหนดขึ้นสามารถใช้ได้ เฉพาะ Procedure ที่มีการ Declare ตัวแปร Constant นั้น

การใช้ตัวแปรในโปรแกรม บ่อยครั้งที่ผู้พัฒนาโปรแกรมจะลืม Declare ตัวแปรไว้ก่อนมีการใช้งาน ตัวแปรนั้น ซึ่งอาจทำให้ประเภทของตัวแปรที่ใช้ผิดไปจากที่ต้องการได้ ดังนั้นเราสามารถกำหนดให้ Visual Basic คอยตรวจสอบว่า มีตัวแปรใดบ้างที่ใช้ในโปรแกรมแล้วยังไม่ได้มีการ Declare ได้โดยเลือกจากเมนู Tools และ Options...ตามลำดับ ให้เลือก "Require Variable Declaration" แล้วคลิกที่ปุ่ม OK จะปรากฏ ประโยคคำสั่ง

Option Explicit

ในส่วน Declaration ของ Editor เสมอ เพื่อกำหนดให้กับ Editor คอยตรวจสอบว่ามีตัวแปรใดที่ถูก ใช้งาน โดยไม่มีการ Declare หรือไม่

2.1.2 ขอบเขตการใช้งานของตัวแปร

ตัวแปรที่ถูกกำหนดขึ้นด้วยคำสั่ง Declare จะมีขอบเขตในการใช้งานภายใต้ Procedure หรือ Function ต่าง ๆ ขึ้นอยู่กับว่าการ Declare ตัวแปรนั้นกระทำ ณ ที่ใดของโปรแกรม ดังนั้นจึงสามารถแบ่ง ประเภทของตัวแปรตามขอบเขตการใช้งาน ได้ออกเป็น 2 แบบดังนี้

1. **ตัวแปรแบบ Private** ตัวแปรชนิดนี้จะมีขอบเขตการใช้งานอยู่ในเฉพาะ Procedure หรือ Function ที่ Declare ไว้ ดังนั้นตัวแปรชื่อเดียวกันจึงสามารถ Declare ในต่าง procedure หรือ Function กันได้ โดยที่ Visual Basic จะมองตัวแปรเหล่านั้นเป็นตัวแปรคนละตัวกัน เมื่อ เปลี่ยนแปลงค่าของตัวแปรประเภทนี้ใน Procedure หรือ Function อื่น ๆ แต่อย่างไร ในการ Declare ตัวแปรประเภทนี้ ทำได้ทั้งแบบ Implicit และแบบ Explicit แต่ต้องกำหนดใน Procedure หรือ Function ที่ต้องการใช้งาน
2. **ตัวแปรแบบ Public** ตัวแปรประเภทนี้จะแตกต่างจากแบบแรก กล่าวคือ จะมีขอบเขตการ ใช้งานอยู่ในทุก ๆ Procedure หรือ Function ภายใต้ Form เดียวกัน เมื่อเปลี่ยนแปลงค่าของตัวแปรประเภทนี้ ไม่ว่าจะที่ Procedure หรือ Function ใดภายใต้ Form เดียวกัน จะส่งผลต่อตัวแปรชื่อเดียวกันใน Procedure หรือ Function อื่นด้วย ในการ Declare จึงกระทำในส่วน ของ General

2.1.3 ฟังก์ชันสำหรับเขียนโปรแกรม

คำสั่งและฟังก์ชัน Time, Time\$

Time, Time\$ จะมีลักษณะเช่นเดียวกับฟังก์ชัน Date และ Date\$ แต่จะใช้กับเวลา โดยที่ฟังก์ชัน Time\$ จะคืนค่ากลับมาเป็นเวลาในรูปแบบของ String ขนาด 8 ตัวอักษรในรูปแบบ hh:mm:ss ในขณะที่ ฟังก์ชัน Time จะให้ค่า เดียวกันแต่ในรูปแบบของ Date สำหรับคำสั่ง Time\$ และ Time ใช้สำหรับกำหนดค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของเวลาปัจจุบัน โดยจะใช้รับข้อมูลในรูปของ String แต่คำสั่ง Time จะใช้รับข้อมูลได้หลายรูปแบบมากกว่า ซึ่งมักจะใช้ร่วมกับฟังก์ชัน TimeSerial

ฟังก์ชัน TimeSerial

ฟังก์ชันนี้จะมีลักษณะเช่นเดียวกับฟังก์ชัน DateSerial แต่ฟังก์ชันนี้จะใช้กับเวลาแทน กล่าวคือใช้สำหรับแปลงค่าที่เป็นตัวเลขให้อยู่ในรูปของเวลา โดยมีรูปแบบดังนี้

TimeSerial(hour, minute, second)

โดยที่ hour หมายถึง ตัวเลขที่มีค่าระหว่าง 0 - 23

minute หมายถึง ตัวเลขที่มีค่าระหว่าง 0 - 59

second หมายถึง ตัวเลขที่มีค่าระหว่าง 0 - 59

ฟังก์ชัน TimeValue

ฟังก์ชันนี้จะมีลักษณะเช่นเดียวกับฟังก์ชัน DateValue แต่ฟังก์ชันนี้จะใช้กับเวลาแทน กล่าวคือใช้สำหรับแปลงค่าที่เป็น String ให้อยู่ในรูปของเวลา โดยมีรูปแบบดังนี้

TimeValue(time)

โดยที่ time จะหมายถึงเวลาที่ในรูปแบบของ String เช่น "3:05", "3:05:20", "3:05 pm" เป็นต้น

ฟังก์ชัน Hour, Minute และ Second

ใช้สำหรับแสดงค่าของ Date ในรูปของชั่วโมงโดยใช้ฟังก์ชัน Hour ในรูปของนาฬิกาโดยใช้ฟังก์ชัน Minute และในรูปของวินาทีโดยใช้ฟังก์ชัน Second

Hour(time) | Minute(time) | Second(time)

โดยที่ time หมายถึง เวลาที่ในรูปแบบของ Date เช่น

A = Hour(Time)	'A จะมีค่าเป็น 14 (เวลาปัจจุบันเป็น 14:20:39)
B = Minute(time)	'B จะมีค่าเป็น 20
C = Second(Time)	'C จะมีค่าเป็น 39

หมายเหตุ : ค่าที่ปรากฏในวงเล็บของทั้ง 3 ฟังก์ชันนี้คือ เวลา ซึ่งในที่นี้ใช้ฟังก์ชัน Time เพื่อบอกเวลาปัจจุบัน

2.1.4 การเขียนโปรแกรมกับ Control

การเขียนโปรแกรมฐานข้อมูลด้วย Data Control

Visual Basic เป็นภาษาคอมพิวเตอร์ที่ใช้ในการพัฒนาโปรแกรมในลักษณะของ Front-End ซึ่งผู้พัฒนาโปรแกรมสามารถที่จะกำหนดรูปแบบของจอภาพให้ง่ายต่อการใช้งาน เพื่อช่วยลดความซับซ้อนในการทำงานของ Database Management System (DBMS) ซึ่งเป็นโปรแกรมที่ใช้ในการจัดการข้อมูลในฐานข้อมูล ไม่ว่าจะเป็นการจัดเก็บ การค้นคืน การเพิ่ม การลบ หรือการแก้ไขข้อมูล

ในการแก้ไขข้อมูล Visual Basic จะกระทำผ่านตัว Database Engine ซึ่งมีความสามารถในการเข้าถึงฐานข้อมูลได้หลายรูปแบบ เช่น Microsoft Access, dbase, Foxpro, Paradox หรือ Btrieve โดยแต่ละฐานข้อมูลจะมีโครงสร้างที่แตกต่างกันไป แต่ที่นิยมใช้กับ Visual Basic ได้แก่ Microsoft Access ซึ่งมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างแบบ Relational Database ดังนั้นในบทนี้เราจะใช้ Microsoft Access เป็นฐานข้อมูลตัวอย่างในการทำงาน

Data Control

Visual Basic จะอาศัย Control ชื่อ “Data” (มักจะเรียกว่า “Data Control”) ในการทำงานร่วมกับ Database โดยที่ 1 Data Control จะอ้างถึง 1 table หรือมากกว่าในฐานข้อมูล (กรณีที่อ้างถึงมากกว่า 1 Table จะหมายถึงการนำเอา Table ต่าง ๆ ที่สัมพันธ์กันมา Join กัน) ข้อมูลที่ถูกอ่านจาก Table มาเก็บไว้ใน Data Control จะเรียกว่า “Recordset” ดังนั้นจึงอาจกล่าวได้ว่า Recordset ได้แก่ กลุ่มของ Recordset ได้แก่ กลุ่มของ Record ใน Table ซึ่งอาจมาจาก 1 หรือมากกว่า 1 Table ที่ถูกอ่านเข้ามาเก็บไว้ในตัว Data Control ตามเงื่อนไขที่กำหนด ในการอ้างถึง Recordset ใน Data Control ให้เขียนอยู่ในรูปแบบ ดังนี้

`Datacontrol .Recordset.methodproperty`

โดยที่ datacontrol หมายถึง ชื่อของ Data Control

methodproperty หมายถึง Method หรือ Property ของ Data Control

ในการทำงานกับ Data Control จะต้องอาศัย Property ต่าง ๆ ดังนี้

1. Connect ใช้สำหรับกำหนดประเภทฐานข้อมูล สำหรับ Visual Basic 6.0 จะสามารถเข้าถึงข้อมูลที่เป็น dBase (dBase 3, dBase4 และ dBase 5), Excel (Version 3.0, 4.0, 5.0 และ 8.0), Foxpro (Version 2.0, 2.5, 2.6 และ 3.0), Lotus (WK1, WK3 และ WK4), Paradox (Version 3.X, 4.X และ 5.X) และ Text File ทั่วไป

2. DatabaseName ใช้สำหรับกำหนด Path และชื่อของ Database (สำหรับ Microsoft Access จะเป็น file.MDB)

3.RecordsetType ใช้สำหรับกำหนดประเภทของ Recordset ซึ่งประกอบด้วย

1.1 Table เป็น Recordset ที่กระทำกับ Table เพียง Table

1.2 Dynaset เป็น Recordset ที่กระทำกับ Table ตั้งแต่ 1 Table ขึ้นไป Recordset ประเภทนี้จะสามารถแก้ไขข้อมูลได้ โดยข้อมูลที่ได้รับการแก้ไขจะถูกส่งผ่านไปยัง Table ที่อ้างถึงโดยอัตโนมัติ

1.3 Snapshot เป็น Recordset ที่กระทำกับ Table ตั้งแต่ 1 Table ขึ้นไป แต่ Recordset ประเภทนี้จะไม่สามารถแก้ไขข้อมูลได้ ส่วนใหญ่จะใช้ในการแสดงผลเพียงอย่างเดียว

4. Recordsource ใช้สำหรับกำหนดชื่อของ Table เมื่อกดคลิกใน Property นี้จะปรากฏเป็น ListBox ที่ประกอบไปด้วยชื่อของ Table ต่าง ๆ ใน Database ที่กำหนดไว้ใน Property “DatabaseName”

5. BOFAction ใช้กำหนดการทำงานให้กับ Data Control เมื่อเกิดสถานะ “Begin of File” ขึ้น ซึ่งไม่สามารถกำหนดได้ 2 ลักษณะดังนี้

5.1 “0 - MoveFirst” (ค่า Default) เพื่อเลื่อน Pointer ไปยัง Record แรก

5.2 “1 - BOF” เพื่อกำหนดค่าของ Property “BOF” เป็น True ซึ่งจะส่งผลให้ Data Control ใช้

Method “MoveLast” ไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. EOFAction ใช้กำหนดการทำงานให้กับ Data Control เมื่อเกิดสถานะ “End of File” ขึ้น ซึ่งสามารถกำหนดได้ 3 ลักษณะดังนี้

6.1 “0 - MoveLast” (ค่า Default) เพื่อเลื่อน Pointer ไปยัง Record สุดท้าย

6.2 “1 - EOF” เพื่อกำหนดค่าของ Property “EOF” เป็น True ซึ่งจะส่งผลให้ Data Control ใช้ Method “MoveLast” ไม่ได้

7. ReadOnly เป็น Property ที่มีลักษณะข้อมูลแบบตรรกะ ใช้สำหรับกำหนดให้ Data Control อ่านได้อย่างเดียวเมื่อกำหนดให้มีค่าเป็น True และจะสามารถแก้ไขได้เมื่อกำหนดให้มีค่าเป็น False

Bound Control

Bound Control ได้แก่ Control ต่าง ๆ ที่สามารถใช้งานร่วมกับ Recordset ใน Data Control เพื่อใช้ในการแสดงผลและรับข้อมูลจากทางจอภาพ Bound Control มาตรฐานใน Visual Basic ได้แก่ TextBox, ListBox, CheckBox, Image, Label และ Picture

2.1.5 Method ที่ใช้ในการจัดเก็บข้อมูล

ในการทำงานกับ Database จะอาศัย Method “AddNew” เพื่อสร้าง Record วางต่อเพิ่มเข้าไปใน Recordset

Recordset. Addnew

โดยที่ Recordset หมายถึง ชื่อ Object ที่เป็นเจ้าของ Recordset ในกรณีที่ใช้ Data Control ให้กำหนดในรูป datacontrol.Recordset โดย datacontrol ได้แก่ ชื่อของ Data Control นั้น

การแก้ไขข้อมูล

Method ที่ใช้ในการบันทึกข้อมูลที่มีการแก้ไข ได้แก่ Method “Update” โดยจะบันทึกข้อมูลที่ปรากฏอยู่ใน Object ที่เป็น Bound Control กลับไปยัง Record ปัจจุบันใน Recordset รูปแบบของคำสั่งดังนี้

Recordset. Update

โดยที่ Recordset หมายถึง ชื่อ Object ที่เป็นเจ้าของ Recordset ในกรณีที่ใช้ Data Control ให้กำหนดในรูป datacontrol.Recordset โดย datacontrol ได้แก่ ชื่อของ Data Control นั้น

การลบข้อมูล

Method ที่ใช้สำหรับลบข้อมูล Record ปัจจุบันออกจาก Recordset ได้แก่ Method “Delete”

Recordset. Delete

โดยที่ Recordset หมายถึง ชื่อ Object ที่เป็นเจ้าของ Recordset ในกรณีที่ใช้ Data Control ให้กำหนดในรูป datacontrol.Recordset โดย datacontrol ได้แก่ ชื่อของ Data Control นั้น

ในการลบ Record ทุกครั้งหลังการลบ ควรที่จะมีการขยับ Pointer เนื่องจาก Record ปัจจุบันได้ถูกกลับไปเพื่อป้องกันการสับสนว่า Pointer ชี้ไปที่ Record ใด สำหรับ Method ที่ใช้ในการเลื่อน Pointer ได้แก่ Method “MoveFirst” , “MoveNext” ,”MoveLast” และ “MovePrevious”

Recordset.{ MoveFirst | MoveLast | MoveNext | MovePrevious}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ Recordset หมายถึง ชื่อ Object ที่เป็นเจ้าของ Recordset ในกรณีที่ใช้ Data Control ให้กำหนดในรูป datacontrol.Recordset โดย datacontrol ได้แก่ ชื่อของ Data Control นั้น ทั้ง 4 Method นี้ จะมีรูปแบบของคำสั่งเช่นเดียวกัน สำหรับหน้าที่ของแต่ละ Method มีดังนี้

1. MoveFirst ใช้สั่งให้ Data Control เลื่อน Pointer ไปยัง Record แรกใน Recordset
2. MoveLast ใช้สั่งให้ Data Control เลื่อน Pointer ไปยัง Record สุดท้ายใน Recordset
3. MoveNext ใช้สั่งให้ Data Control เลื่อน Pointer ไปยัง Record ถัดไปใน Recordset
4. MovePrevious ใช้สั่งให้ Data Control เลื่อน Pointer ไปยัง Record ก่อนหน้าใน Recordset

ในกรณีที่ Pointer ของ Record ปัจจุบันชื่ออยู่ที่ Record แรกของ Recordset เมื่อใช้ Method “MovePrevious” ค่าของ Property “BOF” ของ Data Control จะถูกกำหนดให้มีค่าเป็น True และก็เช่นเดียวกัน กรณีที่ Pointer ของ Record ปัจจุบันชื่ออยู่ที่ Record สุดท้ายของ Recordset เมื่อใช้ Method “MoveNext” ค่าของ Property “BOF” จะถูกกำหนดให้มีค่าเป็น True เช่นเดียวกัน และถ้าไม่ใช่ทั้ง 2 กรณีแล้ว ค่าของ Property “BOF” และ “EOF” จะถูกกำหนดให้มีค่าเป็น False เสมอ

2.1.6 Data Environment Designer

เป็น Tool ใหม่อีก Tool หนึ่ง que เพิ่มเติมเข้ามาใน Visual Basic 6.0 ที่สนับสนุนแนวความคิดทางด้าน Reusability กล่าวคือ เป็น Tool ที่ใช้กำหนด แสดง และปรับปรุง Data Environment ของแต่ละ Project ที่สร้างขึ้น

Data Environment

Data Environment เป็นสภาพแวดล้อมของข้อมูลที่ถูกนำมาใช้ใน Project โดยมองข้อมูลทั้งหมดที่ถูกนำมาใช้ใน Project เป็น Object หนึ่ง ๆ จะอ้างถึงข้อมูลใดข้อมูลหนึ่งเพียงข้อมูลเดียว เช่น การใช้ Data Control เพื่ออ้างถึงข้อมูลใน Table เป็นต้น ซึ่งด้วยแนวคิดนี้ ในกรณีที่มีการเปิดหลาย ๆ Table ใน Project จึงต้องใช้ Object ที่อ้างถึงข้อมูลเหล่านั้นหลาย Object ด้วยกันและยังในกรณีที่ Project อื่นมีการใช้ข้อมูลชุดเดียวกัน ยิ่งส่งผลให้เสียเวลาในการพัฒนาโดยไม่จำเป็น แต่ด้วยแนวความคิดของ Data Environment จะทำให้การติดต่อกับข้อมูลต่าง ๆ ที่เกิดขึ้นใน Project ไม่ว่าจะเป็นเพียง Table เดียวหรือมากกว่า หรืออาจอยู่ในรูปของ View ถูกมองเป็นเพียง Object เดียว และสามารถบันทึกอยู่ในรูปของไฟล์ เพื่อนำไปใช้งาน Project อื่นได้ ตามแนวความคิดของ Reusability สำหรับ Object ที่ใช้แทน Data Environment ที่สร้างขึ้นสามารถใช้แทนข้อมูลได้ทั้งในรูปของ Table, View, Stored Procedure และ ประโยคคำสั่ง SQL

การเรียกใช้ Data Environment Designer

การเรียกใช้ Data Environment Designer กระทำได้ 2 กรณีดังนี้

1. **กรณีเปิด Data Environment ใหม่** ให้คลิกที่ Icon “Add Data Environment” ในจอภาพ
2. **กรณีแก้ไข Data Environment เดิม** ให้ดับเบิลคลิกที่ Icon ของ Data Environment ที่ต้องการแก้ไขในจอภาพ Project Explorer

เมื่อสังเกตจอภาพของ Data Environment จะพบว่า มี Connection 1 ปรากฏอยู่ Connection นี้ได้แก่ Object “Command” เป็นต้น ซึ่ง Object ต่าง ๆ ที่สร้างขึ้นใน Connection นี้จะไปปรากฏในส่วน

Data Environment Connections ของจอภาพ Data View โดยอัตโนมัติ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่เปิด Data Environment ใหม่ทุกครั้ง จะปรากฏ Icon ที่แทน Object ของ Data Environment ที่ถูกสร้างขึ้นใหม่เสมอในจอภาพ Project Explorer โดยมีชื่อที่ขึ้นต้นด้วย DataEnvironment และตามด้วยลำดับที่ของ Data Environment นั้น เช่น DataEnvironment1, DataEnvironment2, Data Environment3 เป็นต้น

สำหรับ Data Environment ที่ถูกสร้างขึ้นนี้ จะปรากฏเป็นส่วนหนึ่งของ Project ที่สร้างขึ้นเสมอ ดังนั้นเมื่อมีการบันทึก Project ที่สร้างขึ้น จึงต้องบันทึก DataEnvironment ที่สร้างขึ้นด้วย

การสร้าง Table ใน Connection

ในการสร้าง Object ที่แทน Table ใน Connection สามารถทำได้ 2 วิธีดังนี้

1. ลาก Icon ของ Table ที่ต้องการจากส่วน Data Link ของจอภาพ Data View มาไว้ภายใต้

Connection ของจอภาพ Data Environment Designer

2. กำหนดขึ้นจากจอภาพ Data Link Properties เช่นเดียวกับการสร้าง Data Link โดยเลือกจากเมนู Properties

การกำหนด Object “Command” ใน Connection

Object “Command” ได้แก่ Object ที่ใช้แทนคำสั่งที่ใช้แทน Recordset หรือข้อมูลที่ได้จากการใช้คำสั่งอย่างใดอย่างหนึ่งกับข้อมูลใน Object ของข้อมูลที่ปรากฏอยู่ใน Data Environment ที่สร้างขึ้น เช่น Recordset ที่ได้จากการใช้คำสั่ง SQL เป็นต้น ในการกำหนด Object “Command” ใน Connection ทำได้ 2 วิธีดังนี้

1. เลือกจากเมนู Add Command ของ Connection
2. คลิก Icon ใน Toolbar ของจอภาพ Data Environment Designer

การบันทึก Data Environment

Connection ที่กำหนดในจอภาพ Data Environment Designer สามารถนำไปใช้ใน Project ที่สร้างขึ้นได้ทันที เนื่องจาก Data Environment ที่กำหนดขึ้นในจอภาพ Data Environment Designer จะถูกนำไปไว้ใน Project โดยอัตโนมัติ และเมื่อบันทึก Project จึงปรากฏจอภาพสำหรับบันทึก Data Environment ที่ประกอบอยู่ใน Project ไปด้วย

สำหรับ Data Environment จะบันทึกอยู่ในไฟล์นามสกุล DSR

การนำเข้า Data Environment ไปใช้งานใน Project

ในการนำไฟล์ Data Environment ที่บันทึกไว้ ไปใช้งานภายใน Project ต่าง ๆ สามารถกระทำได้โดยเลือกจากเมนู Project และ Add File ตามลำดับ จะปรากฏจอภาพ Add File ให้เลือกไฟล์นามสกุล DSR ที่ต้องการแล้วคลิกที่ปุ่ม Open ชื่อของ Data Environment ดังกล่าวจะมาปรากฏในจอภาพ Project Explorer

การนำ Connection ไปใช้งาน

ในการนำ Connection ที่กำหนดในจอภาพ Data Environment เช่น Table หรือ Object “Command” เป็นต้น ไปใช้งานกับ Bound Control ที่วาดไว้บน Form สามารถกระทำได้โดยตรง โดยไม่ต้องผ่าน ADO Data Control ด้วยการกำหนดค่าให้กับ Property “DataSource” และ “DataMember” ของ Bound Control นั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเชื่อมโยง DataReport กับ Visual Basic

สำหรับรายงานที่สร้างขึ้นด้วย Data Report Designer เราสามารถเรียกใช้จาก Project ของ Visual Basic ได้โดยใช้ Method “Show”

2.1.7 การประยุกต์ใช้งาน Visual Basic

เรื่อง การประยุกต์ไปใช้งาน

Serial Port(RS-232)

สามารถติดต่อกับอุปกรณ์ต่างๆที่มีการติดต่อกับอุปกรณ์ภายนอกผ่านทาง RS-232 เช่น เครื่องซึ่งนำรวมถึงโพลลเซล(เป็นเซ็นเซอร์ชนิดหนึ่งใช้สำหรับวัดน้ำหนักซึ่งที่ชุดแสดงผลภายในเป็นชุด ไมโครคอนโทรลเลอร์ จะมีสัญญาณรับส่งทาง RS-232),เครื่องวัดงานทางด้านไฟฟ้า, ไมโครคอนโทรลเลอร์, ควบคุมอุปกรณ์ไฟฟ้า, โอนถ่ายข้อมูลในฮาร์ดดิสค์ระหว่างเครื่องคอมพิวเตอร์ด้วยกัน, ควบคุมเสตปปีงมอเตอร์ เป็นต้น ข้อดีของการติดต่อข้อมูลกันผ่านทาง RS-232 ก็คือสามารถใช้ได้ในระยะทางไกลๆระหว่างอุปกรณ์ ที่ติดต่อกัน

เนื่องจากที่ Microsoft Visual Basic 5.6 จะมีตัวคอนโทรลชื่อ MS Comm ที่ใช้ติดต่อกับ Serial Port(RS-232) ทำให้การพัฒนาโปรแกรมในด้านนี้ได้เร็วและเป็นมาตรฐาน

TCP/IP

ใช้กับระบบเครือข่าย Internet เป็นโปรโตคอลมาตรฐานชื่อ TCP/IP ในการสื่อสารผ่านระบบเพื่อติดต่อกับเครื่องคอมพิวเตอร์อื่น โปรโตคอล TCP/IP ประกอบด้วยส่วนที่สำคัญ 2 ส่วนก็คือ

- TCP(Transmission Control Protocol)
- IP(Internet Protocol)

ในการติดต่อสื่อสารกันจะอธิบายโปรโตคอล TCP/IP มีส่วนประกอบดังนี้

- IP Address : สำหรับการรับส่งข้อมูลในระบบ Internet จะถูกกำหนดและอ้างอิงด้วยหมายเลขประจำเครื่อง นั่นก็คือเครื่อง IP Address ซึ่งในระบบ Internet จะมีเครื่องคอมพิวเตอร์เป็นจำนวนมากที่อยู่ในระบบ ในการที่จะใช้ IP Address อาจจะไม่สะดวก จึงได้มีการเปลี่ยนมาใช้เป็น ชื่อ ในความเข้าใจก็คือ Domain name โดยทั้งหมดนี้อยู่ในระบบ Name Services ซึ่งเป็นการอ้างอิงชื่อแทนหมายเลขนั่นเอง
- Routing Configuration : ข้อดีของโปรโตคอล TCP/IP ก็คือในการกำหนดเส้นทางสำหรับการรับส่ง ที่สามารถเลือกเส้นทางในการรับส่งข้อมูลได้อย่างอัตโนมัติหากถ้าเกิดเส้นทาง บางเส้นทางเสียหาย ระบบทั่วโลก ในการกำหนดเส้นทางสำหรับการรับส่งข้อมูลของโปรโตคอล TCP/IP ก็จะเลือกเส้นทางที่เหมาะสมถูกต้องให้สามารถรับส่งข้อมูลได้
- Protocol, Ports, Sockets : เป็นช่องทางสำหรับกำหนดทิศทางของการรับส่งข้อมูลนอกเหนือจากที่จะต้องกำหนดหลังจาก IP Address

ในงานควบคุมระยะไกลระบบนี้เป็นระบบที่มีประสิทธิภาพดี ซึ่งสามารถจะเขียนโปรแกรมไปควบคุม I/O ของระบบคอมพิวเตอร์ หรือฮาร์ดแวร์ที่มีการติดต่อกับระบบเดียวกัน สำหรับ Visual Basic มีเครื่องมือช่วยในการเขียนโปรแกรมด้านนี้มีชื่อว่า Winsock หรือจะใช้แบบฟังก์ชัน API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 Assembly

2.2.1 การใช้งานไมโครคอนโทรลเลอร์ MCS-51

ทฤษฎี MCS-8051

โดยปกติแล้ว เอ็มซีเอส-8051 ก็คือระบบคอมพิวเตอร์ซึ่งมีข้อมูลจากภายนอก และนำมาประมวลผล พร้อมทั้งสามารถส่งสัญญาณเพื่อการควบคุมการทำงานของอุปกรณ์ต่างๆ ได้อย่างดี แล้วในส่วนของการติดต่อสื่อสารข้อมูล (Data Communication) กับระบบภายนอกอื่นๆ ก็สามารถกระทำโดยผ่านพอร์ตอนุกรม (Serial Port) ซึ่งพอร์ตอนุกรมนี้นี้เป็นส่วนที่เหมาะสมในการรับหรือส่งข้อมูลในระยะทางไกลได้ดีกว่าพอร์ตนาน ซึ่งในการทำโครงการนี้เน้นเฉพาะการใช้โปรแกรมใช้งานพอร์ตอนุกรมของเอ็มซีเอส-8051

2.2.2 การใช้งานพอร์ตอนุกรม

รีจิสเตอร์ต่างๆ ที่เกี่ยวข้องในการใช้งานพอร์ตอนุกรม

- รีจิสเตอร์ควบคุม ไทม์เมอร์ เนื่องจากการใช้งานพอร์ตอนุกรมนั้นจะมีสิ่งที่จะต้องคำนึงถึง คือ อัตราการรับส่งข้อมูล หรือที่เรียกว่าอัตราบอด (Baud Rate) จริงๆ แล้วคือจังหวะการเลื่อนเข้าหรือออกของข้อมูลจากเอ็มซีเอส-8051 นั่นเอง โดยอัตราบอดนี้จะสามารถสร้าง ได้ภายในของเอ็มซีเอส-8051 ได้จากไทม์เมอร์ แชนแนล 1 โดยทำงานในโหมด 2 คือ โหลดข้อมูลกลับอัตโนมัติ ดังนั้นรีจิสเตอร์ที่จะต้องทำการ โปรแกรมจะมีดังนี้
 - TMOD ตำแหน่ง 89 H ทำหน้าที่เลือก โหมดของไทม์เทอร์
 - TCON ตำแหน่ง 88 H ทำหน้าที่เริ่มต้นการสร้างอัตราบอด
 - TH1 ตำแหน่ง 8C H ทำหน้าที่ใส่ข้อมูลการนับของ ไทม์เมอร์ 1 เพื่อสร้างอัตราบอดตามต้องการ
- รีจิสเตอร์ควบคุมการลัดกำลัง เนื่องจากการสร้างอัตราบอดนั้นจะต้องนำบิตในรีจิสเตอร์ PCON มาใช้ในการคำนวณข้อมูลของ TH1 ดังนั้น รีจิสเตอร์ที่ใช้คือ

7	6	5	4	3	2	1	0
SMOD	-	-	-	GF1	GFO	PD	IDL

บิต สัญลักษณ์

รายละเอียด

- 7 SMOD บิตที่สำหรับเปลี่ยนแปลง แกไขอัตราการส่งข้อมูลแบบอนุกรม โดยใช้โปรแกรม เซ็ตให้เป็น 1 จะเป็นการเพิ่มอัตราการส่งขึ้นเท่าตัวเมื่อใช้ไทม์เมอร์ 1 เข้าช่วย ถ้าเลือกให้การรับส่งข้อมูลในโหมด 1, 2 , และ 3 แต่ถ้าเคลียร์เป็น 0 จะใช้ไทม์เมอร์ 1 ตามปกติที่เป็นอยู่ตามการกำหนดในช่วงเริ่มต้นของ โปรแกรม
- 6-4 - ไม่ได้ใช้งาน

3 GF1 แฟล็กใช้งานทั่วไป กำหนดโดยผู้ใช้ ไม่เกี่ยวข้องกับการควบคุมใดๆทั้งสิ้น เซ็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่บนสื่อออนไลน์ การค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเคลียร์โดยโปรแกรม

- 2 GFO แฟล็กใช้งานทั่วไป กำหนดโดยผู้ใช้เช่นกัน
- 1 PD บิตที่แสดงการลดลงของกำลังไฟ (Power Down) เซ็ตเป็น 1 โดยโปรแกรมเพื่อเข้าสู่โหมดของการลดลงของกำลังไฟ ใช้ในไมโครโปรเซสเซอร์ที่เป็นแบบซิมอสเท่านั้น
- 0 IDL บิตที่แสดงโหมดไอเดิล (Idle) เซ็ตเป็น 1 โดยโปรแกรมเพื่อเข้าไอเดิลในไมโครโปรเซสเซอร์ที่เป็นแบบซิมอสเท่านั้น

3. รีจิสเตอร์ควบคุมอินเทอร์รัพต์ เนื่องจากว่าเอ็มซีเอส-8051 สามารถใช้งานพอร์ตอนุกรมในลักษณะของการอินเทอร์รัพต์ได้ จึงมีรีจิสเตอร์ที่เกี่ยวข้องดังนี้

- IE ตำแหน่ง A8H ทำหน้าที่ยอมให้เกิดการอินเทอร์รัพต์จากพอร์ตอนุกรมได้หรือไม่ ซึ่งมีรายละเอียดของบิตต่างๆ ดังนี้

7	6	5	4	3	2	1	0
EA	X	ET2	ES	ET1	EX1	ETO	EXO

บิต	สัญลักษณ์	รายละเอียด
7	EA	บิตที่ควบคุมทั้ง 6 บิต ถ้าข้อมูลในบิตนี้เป็น 0 จะเป็นการดิสเอเบิลทุกบิตทำให้เกิดการขัดจังหวะ ถ้าบิตนี้เป็น 1 การอินาเบิล หรือดิสเอเบิล ใน 6 บิตจะขึ้นกับข้อมูลในแต่ละบิตนั้น
6	=	บิตที่ผู้ใช้กำหนดการใช้งานเอง
5	ET2	บิตนี้จะใช้อินาเบิล หรือดิสเอเบิล สัญญาณขัดจังหวะที่มาจากวงจรไทม์เมอร์ 2 บิตนี้จะใช้อินาเบิล หรือดิสเอเบิลสัญญาณขัดจังหวะที่มาจากพอร์ตอนุกรม
4	ES	เมื่อมีข้อมูลเข้ามายัง SBUF หรือ ข้อมูลจาก SBUF ได้ถูกส่งออกไปทาง พอร์ตอนุกรม หมดแล้ว
3	ET1	บิตนี้จะใช้อินาเบิล หรือดิสเอเบิล สัญญาณขัดจังหวะที่มาจากวงจรไทม์เมอร์ 1
2	EX1	บิตนี้ใช้สำหรับการอินาเบิลสัญญาณที่เข้ามาทางขา INT1 ให้เกิดการขัดจังหวะ
1	ETO	บิตนี้จะใช้อินาเบิลหรือดิสเอเบิล สัญญาณขัดจังหวะที่มาจากวงจรไทม์เมอร์ 0
0	EXO	บิตนี้ใช้สำหรับการอินาเบิลสัญญาณที่เข้ามาทางขา INTO ให้เกิดการขัดจังหวะ

- IP ตำแหน่ง B8 H ทำหน้าที่จัดลำดับความสำคัญของการอินเทอร์รัพต์ ซึ่งมีรายละเอียดของบิตต่างๆ

ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7	6	5	4	3	2	1	0
X	X	PT2	PS	PT1	PX1	PTO	PXO

<u>บิต</u>	<u>สัญลักษณ์</u>	<u>รายละเอียด</u>
7	PCT	PCT = 1 จะมีเพียงหนึ่งระดับ
6	-	บิตที่ผู้ใช้กำหนดการใช้งานเอง
5	PT2	กำหนดระดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัปต์ของ ไทม์เมอร์ 2
4	PS	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัปต์ของ พอร์ตอนุกรม
3	PT1	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัปต์ของ ไทม์เมอร์ 1
2	PX1	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัปต์ภาย นอกที่ 1
1	PTO	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัปต์ของ ไทม์เมอร์ 2
0	PXO	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัปต์ภาย นอกที่ 0

4. รีจิสเตอร์ควบคุมพอร์ตอนุกรม การใช้งานพอร์ตอนุกรมนี้จะขึ้นอยู่กับรีจิสเตอร์นี้โดยตรง คือ
- SBUF ตำแหน่ง 99 H ทำหน้าที่เป็นบัฟเฟอร์การรับหรือการส่ง
 - SCON ตำแหน่ง 98 H ทำหน้าที่ควบคุมและกำหนดโหมดการใช้งานพอร์ตอนุกรมทั้งหมดซึ่งมีรายละเอียดของบิตต่างๆ ดังนี้

7	6	5	4	3	2	1	0
SMO	SM1	SM2	REN	TB8	RB8	TI	RI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต	สัญลักษณ์	รายละเอียด
7	SM0	โหมดการพอร์ตต่ออนุกรมบิต 0 ทำการเซ็ทหรือเคลียร์โดยใช้โปรแกรมสั่งงาน
6	SM1	โหมดการพอร์ตต่ออนุกรมบิต 1 ทำการเซ็ทหรือเคลียร์โดยใช้โปรแกรมสั่งงาน เช่นกัน รายละเอียดดังตาราง 2.1
5	SM2	ใช้เป็นการติดต่อระหว่างไมโครโปรเซสเซอร์หลายตัวด้วย (Multiprocessor Communication) เซ็ทและเคลียร์ โดยการ ใช้โปรแกรมควบคุมในกรณีนี้ควรจะใช้เฉพาะ โหมด 2 และ 3 เมื่อบิตนี้ถูกเซ็ทเป็น 1 การอินเทอร์รัพต์จะเกิดขึ้น ถ้าข้อมูลที่ได้รับเข้ามาเป็นบิต 9 เป็น 1 ในทำนองกลับกัน ถ้าข้อมูลที่จะทำการรับเข้ามาเป็น 0 การอินเทอร์รัพต์จะเกิดขึ้นเมื่อได้รับบิตหยุดที่ถูกต้องแล้วเท่านั้น แต่ถ้าใช้ในโหมด 0 บิตนี้จะถูกเคลียร์ให้เป็น 0
4	REN	บิตอีนามาเปิด การรับบิตนี้จะถูกเซ็ทเป็น 1 เมื่อต้องการรับสัญญาณอนุกรมและจะต้องถูกเคลียร์ให้เป็น 0 เมื่อ ไม่ต้องการรับสัญญาณอนุกรม
3	TB8	ใช้เลือกว่า จะให้ส่ง 8 บิตหรือไม่ และจะถูกเซ็ทหรือเคลียร์โดยโปรแกรมโหมด 2 หรือ 3
2	RB8	ใช้เลือกว่าจะรับบิต 8 หรือ ไม่ ใช้สำหรับกรณีรับข้อมูล โหมด 2 หรือ 3 หยุดในโหมด 1 ส่วนในโหมด 0 จะไม่ใช้งานบิตนี้
1	TI	แฟล็กอินเทอร์รัพต์เมื่อส่งข้อมูลในโหมด 0 จะถูกเซ็ทให้เป็น 1 หลังจากที่ส่งบิต 7 ออกไปแล้วในโหมดอื่นๆ จะถูกเซ็ทให้เป็น 1 เมื่อบิตหยุด เริ่มส่งออกไป การเคลียร์บิตนี้ จะทำได้โดยใช้โปรแกรม
0	RI	แฟล็กอินเทอร์รัพต์เมื่อรับข้อมูลในโหมด 0 จะถูกเซ็ทให้เป็น 1 หลักการรับบิต 7 เรียบร้อยแล้วในโหมดอื่นๆ จะถูกเซ็ทเป็น 1 เมื่อรับบิตหยุดได้ครั้งหนึ่ง การเคลียร์บิตนี้ จะทำได้โดยการ ใช้โปรแกรม

การอ้างอิงแบบบิตแอดเดรสของรีจิสเตอร์นี้คือ SCON.0 ถึง SCON.7

SM0	SM1	โหมด	รายละเอียด
0	0	0	รีจิสเตอร์แบบเลื่อนบิต อัตราการส่ง = $f / 12$
0	1	1	UART ชนิด 8 บิต อัตราการส่ง = เปลี่ยนแปลงได้
1	0	2	UART ชนิด 9 บิต อัตราการส่ง = $f / 32$ หรือ $f / 64$
1	1	3	UART ชนิด 9 บิต อัตราการส่ง = เปลี่ยนแปลงได้

ตารางที่ 2.1 แสดง โหมดการทำงานของพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณอัตราการบอดโหมด 1

ไทม์เมอร์ 1 จะถูกใช้เป็นตัวสร้างอัตราการบอด เมื่อกำหนดการรับส่งข้อมูลแบบอนุกรมในโหมด 1 โดยไทม์เมอร์จะถูกใช้ให้ทำงานในโหมด 2 ซึ่งจะโหลดค่าเข้าไปโดยอัตโนมัติ การคำนวณจะเป็นดังนี้

$$f_{\text{busd}} = \frac{2^{\text{SMOD}} \times \text{Oscillator frequency} = \text{ความถี่ของออสซิลเลเตอร์}}{32 \times 12 \times [256d - (\text{TH1})]}$$

เมื่อ SMOD จะเป็นบิตควบคุมอยู่ในรีจิสเตอร์ PCON ซึ่งจะเป็น 0 หรือ 1 ก็ได้ ถ้าเป็น 0 จะเป็นความถี่ปกติ ถ้าเป็น 1 ความถี่จะเพิ่มขึ้นเป็น 2 เท่า ดังได้กล่าวมาแล้วในรีจิสเตอร์ PCON

ถ้าไทม์เมอร์ 1 ไม่ถูกใช้งานในโหมด 2 ค่าอัตราการบอดจะเป็น

$$f_{\text{busd}} = \frac{2^{\text{SMOD}} \times (\text{time 1 overflow frequency})}{32}$$

ถ้าเลือกใช้อัตราบอดมาตรฐานควรเลือกความถี่ของคริสตัลให้ถูกต้องคือ 11.0592 เมกกะเฮิร์ตซ์

2.2.3 การกำหนดเริ่มต้นและการใช้งานรีจิสเตอร์ของพอร์ทอนุกรม

บิตต่างๆ ของ SCON

- บิต SMO (บิตที่ 7) และบิต SM1 (บิตที่ 6) เป็นการเลือกโหมด การทำงานพอร์ท อนุกรม สามารถเลือกได้ 4 โหมด

- บิต SM2 (บิตที่ 5) บิตนี้ถ้าเซตเป็น “ 1 ” จะใช้งานใน โหมดโปรเซสเซอร์หลายตัว (Multiprocessor) ดังนั้นการใช้งานโหมด 1 มิใช่เป็นการติดต่อ โปรเซสเซอร์หลายตัว จึงให้บิตนี้เป็น “ 0 ”

- บิต REN (บิตที่ 4)

ถ้าเซตเป็น “ 1 ” หมายถึงเอ็มซีเอส-51 สามารถรับข้อมูลจากพอร์ทอนุกรมได้

ถ้าเคลียร์เป็น “ 0 ” หมายถึงเอ็มซีเอส-51 ไม่สามารถรับข้อมูลจากพอร์ทอนุกรมได้

- บิต TB8 (บิตที่ 3) จะใช้งานในการติดต่อข้อมูลโหมด 2 และ โหมด 3

- บิต RB8 (บิตที่ 2) จะใช้งานในการติดต่อข้อมูลโหมด 2 และ โหมด 3

- บิต TI (บิตที่ 1) จะเซตเมื่อทุกบิตของข้อมูลส่งออกไปแล้ว เพื่อเป็นการแสดงว่าบัฟเฟอร์ของการส่งออกข้อมูลว่าง ถ้าซอฟต์แวร์ต้องการส่งข้อมูลออกไปยังอุปกรณ์ที่ต่ออยู่กับพอร์ทอนุกรม โปรแกรมจะต้องตรวจสอบว่าพอร์ทอนุกรมว่างหรือไม่ (ซึ่งก็คือตรวจสอบว่าตัวอักษรที่แล้วส่งออกไปแล้วหรือยัง) ถ้าว่างก็ส่งข้อมูลต่อไป

- บิต RI (บิตที่ 0) จะเซตเมื่อเสร็จสิ้นการรับข้อมูลแล้ว เพื่อเป็นการแสดงว่าบัฟเฟอร์ของการรับข้อมูลเต็มข้อกำหนดนี้จะตรวจสอบด้วยซอฟต์แวร์ หรือ โปรแกรมให้เกิดการอินเตอร์รัปท์ก็ได้ ถ้าซอฟต์แวร์ต้องการรับข้อมูลจากอุปกรณ์ที่ต่ออยู่กับพอร์ทอนุกรม เช่น เทอร์มินอล ตัวโปรแกรมจะรอจนกระทั่งบิต RI เซตแล้วเคลียร์ด้วย โปรแกรมพร้อมทั้งอ่านข้อมูลจาก SBUF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 การใช้งานไทม์เมอร์ / เคน์เตอร์

การใช้งานไทม์เมอร์ / เคน์เตอร์ภายในของเอ็มซีเอส-8051 จะต้องมีการกำหนดรูปแบบการใช้งานต่างๆ เสียก่อนจึงจะทำงานได้ถูกต้อง การกำหนดค่าเริ่มต้นและการอ่านค่าจากไทม์เมอร์รีจิสเตอร์ เพราะฉะนั้นการจะเริ่มต้นใช้งานไทม์เมอร์ได้จะต้องมีการกำหนดค่าในรีจิสเตอร์ ซึ่งรีจิสเตอร์ที่จะกำหนดจะมี ดังนี้

1. รีจิสเตอร์ TMOD เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ตำแหน่งที่ 89H ไม่สามารถอ้างตำแหน่งแบบบิตได้ รายละเอียดของการกำหนดค่าบิตต่างๆ จะเป็นดังนี้

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0
Timer 1				Time 0			

บิต	สัญลักษณ์	รายละเอียด
7/3	GATE	การอินทิเกรต OR เกท เพื่อควบคุมการทำงานของไทม์เมอร์ GATE = 1 ไทม์เมอร์จะทำงาน ถ้า สัญญาณ ขา INTn เป็นลอจิก 1 และบิต Tm ในรีจิสเตอร์ TCON เป็น 1 ด้วย
6/2	C/T	ถ้าบิต C/T = 0 เป็นการเลือกการทำงานแบบนับพัลส์จากภายในตัวเอ็มซีเอส-51 ถ้าบิต C/T = 1 เป็นการเลือกการทำงานแบบนับพัลส์จากภายนอกที่ป้อนเข้าทางขา P3.5 (T1) หรือ P3.4 (T0) (COUNTER)
5/1	M1	ใช้ในการเลือกโหมดการทำงานดังตารางที่ 2.2
4/0	M0	ใช้ในการเลือกโหมดการทำงานดังตารางที่ 2.2

M1	M0	โหมดที่ใช้งาน	ลักษณะการทำงาน
0	0	0	ไทม์เมอร์ขนาด 13 บิต
0	1	1	ไทม์เมอร์ขนาด 16 บิต
1	0	2	ไทม์เมอร์ขนาด 8 บิต แบบ โหลดค่ากลับอัตโนมัติ
1	1	3	ไทม์เมอร์แบบใช้งานอิสระ

ตารางที่ 2.2 แสดงโหมดการทำงานของไทม์เมอร์

2. รีจิสเตอร์ TH0 , TL0 และ TH1 , TL1 รีจิสเตอร์เหล่านี้เป็นรีจิสเตอร์ขนาด 8 บิต ใช้กำหนดค่าของไทม์เมอร์ ตำแหน่งของ TH0 , TL0 จะอยู่ที่ 8CH , 8AH และตำแหน่ง TH1 , TL 1 จะอยู่ที่ 8DH , 8BH และเนื่องจากว่า ไทม์เมอร์ภายในนี้เป็นไทม์เมอร์แบบนับขึ้นจนถึงค่าสูงสุด คือ FFH (ในกรณีที่เลือกการทำงานแบบ 8 บิต) , FFFH (ในกรณีที่เลือกการทำงานแบบ 16 บิต) แล้วจะเกิดโอเวอร์โฟลว์ (เปลี่ยนจาก FFH เป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FFFH เป็นค่า 0) ดังนั้นการกำหนดค่าการนับจะต้องนำค่าสูงสุดของโหมดการทำงานนั้นๆ มาลบออกด้วยค่าที่ต้องการ

3. รีจิสเตอร์ TCON เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ที่ตำแหน่งที่ 88H สามารถอ้างอิงตำแหน่งแบบบิตได้ ใช้งานในการควบคุมการทำงานของไทม์เมอร์ และ ควบคุมการทำงานอินเทอร์รัพต์ด้วยรายละเอียดที่เกี่ยวกับบิตต่าง ๆ เป็นดังนี้

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

บิต	สัญลักษณ์	รายละเอียด
7	TF1	บิตโอเวอร์โฟลว์ของไทม์เมอร์ 1 เมื่อไทม์เมอร์ 1 นับค่าจนเปลี่ยนจากค่าสูงสุด (FFH , FFFFH) เป็น 0 บิตนี้จะถูกเซตค่าเป็น 1 และบิตนี้จะเป็น 0 เมื่อกลับจากการทำงานในโปรแกรมบริการอินเทอร์รัพต์ (Interrupt Service Routine) หรือ จะใช้การเคลียร์ค่าจากซอฟต์แวร์
6	TR1	บิตควบคุมการเริ่มทำงาน หรือ หยุดการทำงาน ไทม์เมอร์ 1 กล่าวคือ ถ้าบิต TR1 = 1 หมายถึง เริ่มต้นการทำงานไทม์เมอร์ 1 TR1 = 0 หมายถึง หยุดการทำงานไทม์เมอร์ 1 <i>หมายเหตุ: การหยุดการทำงานนี้ไม่ได้หมายความว่าค่าภายในจะถูกเคลียร์</i>
5	TF0	ความหมาย และ การใช้งานเหมือนกับ TF1 แต่เป็นไทม์เมอร์ 0
4	TR0	ความหมาย และ การใช้งานเหมือนกับ TR1 แต่เป็นไทม์เมอร์ 0
3	IE1	บิตของแฟล็กจากการอินเทอร์รัพต์ภายนอกจากขา P3.3 (INT1)
2	IT1	บิตนี้ใช้สำหรับ เลือกวีธีการอินเทอร์รัพต์ภายนอก จาก ขา P3.3 (INT1) ถ้าบิต IT1 = 0 เป็นการเลือกการเกิดอินเทอร์รัพต์จากระดับลอจิก 0 IT1 = 1 เป็นการเลือกการเกิดอินเทอร์รัพต์จากขอบขา ลง
1	IE0	การใช้งานเหมือนกับ IE1 แต่เป็นการอินเทอร์รัพต์ภายนอกจากขา P3.2 (INT0)
0	IT0	การใช้งานเหมือนกับ IT1 แต่เป็นการอินเทอร์รัพต์ภายนอกจากขา P3.2 (INT0)

2.2.5 การอินเทอร์เฟซแบบอนุกรม RS-232

เทคนิคที่สำคัญอีกชนิดหนึ่งของการเชื่อมต่อ คือ การอินเทอร์เฟซแบบอนุกรม RS-232 โดยมีมาตรฐานที่ปรับปรุงใหม่เป็น RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดวางตัวอักษรในรูปแบบอนุกรม

ตัวอักษรจะถูกเก็บในคอมพิวเตอร์ในลักษณะของรูปแบบขนานขนาด 8 บิต ถูกกำหนดโดยแต่ละด้าน แต่ในการส่งแบบอนุกรมมันจะถูกแบ่งเป็น 8 ส่วนที่แยกจากกัน (ส่วนมากมักจะเป็น 7 บิต) ซึ่งแต่ละตัวจะส่งแยกจากกัน วงจรพิเศษทางอิเล็กทรอนิกส์ที่เรียกว่า UART (ใช้สำหรับการส่งและรับแบบอะซิงโครนัส) ใช้สำหรับการทำงานนี้ UART จะถูกสร้างในคอมพิวเตอร์หรือรวมเข้ากับส่วนของการอินเตอร์เฟซแบบอนุกรม UART รุ่นใหม่จะประกอบไปด้วยไอซีตัวเดียวและสามารถทำงานในหน้าที่ซับซ้อน UART ของคอมพิวเตอร์จะรับข้อมูลอักขระ 8 บิต นำมาแยกส่วนและส่งแต่ละบิตออกด้วยอัตราที่กำหนดไว้ก่อน ในการส่งข้อมูล 7 บิต UART จะส่ง 8 บิต ซึ่งมีพาริตีบิต (Parity Bit) และอีก 2 บิตพิเศษคือ บิตเริ่มต้น (Start Bit) และบิตสิ้นสุด (Stop Bit)

พาริตีบิต

บิตที่ 8 ที่ส่งโดย UART ตามหลังข้อมูล 7 บิตที่ถูกสร้างจากตัวอักษร โดยปกติมักจะเป็นพาริตีบิตซึ่งคือการตรวจสอบความผิดพลาด โดยจะมีการใช้พาริตี 2 ทางคือ พาริตีคู่ หรือ พาริตีคี่ สมมุติว่าพาริตีคี่ จะมีขบวนการทำงานดังนี้ ถ้ามีจำนวนบิต "1" เป็นจำนวนคู่ในตัวในข้อมูลอักขระ 7 บิต ดังนั้นพาริตีจะต้องเป็น "1" ซึ่งจะมีผลทำให้มีจำนวนบิต "1" เป็นคี่ตัว ถ้ามีจำนวนบิต "1" เป็นจำนวนคี่ตัว ดังนั้นพาริตีจะต้องถูกกำหนดเป็น "0" ซึ่งจะมีผลทำให้มีจำนวนบิตที่เป็น "1" เป็นคี่ตัว ดังนั้นข้อมูลที่ถูกส่งไปจะมีจำนวนบิต "1" เป็นจำนวนคี่ตัว UART จะตรวจสอบข้อมูลอักขระแต่ละตัวจะประกอบด้วยบิต "1" จำนวนคี่ตัว ถ้าในระหว่างการส่งมีบิตใดบิตหนึ่งในรูปแบบของตัวอักษรถูกเปลี่ยนแปลง บิตพาริตีจะไม่ตรงกับข้อมูล และ UART จะลดความผิดพลาดที่จะเกิดขึ้น ส่วนระบบของพาริตีคู่ก็เหมือนกัน เพียงแต่บิตนี้จะทำการส่งบิต "1" เป็นจำนวนคู่ตัว

บิตเริ่มต้นและบิตสิ้นสุด

UART จะต้องมีเวลาในการเตรียมตัวสำหรับการรับข้อมูล โดยเมื่อบิตเริ่มต้นถูกส่งมาก่อนที่จะส่งข้อมูลอักขระจริงมาตามสายสัญญาณ เมื่อตัวอักษรถูกส่งหมดแล้วบิตสิ้นสุดจะถูกส่งมา ซึ่งจะทำให้ UART สามารถแน่ใจได้ว่าการส่งสิ้นสุดลงแล้ว การใช้บิตเริ่มต้นและบิตสิ้นสุดนี้มีความจำเป็นมาก เพราะการส่งข้อมูลเป็นแบบอะซิงโครนัสไม่ใช่แบบซิงโครนัส ซึ่งคอมพิวเตอร์จะรู้แน่นอนว่าเมื่อใดข้อมูลอักขระจึงจะมาถึง

2.3 ASP และ WAP

2.3.1 ASP

2.3.1.1 Active Server Page

Active Server Page ก็คือ เอกสาร HTML ที่มีนามสกุลเป็น .asp แต่ ASP มีความสามารถที่สูงกว่า HTML ธรรมดาๆ ทั้งนี้เพราะ ASP สามารถสร้างเว็บเพจ ที่ติดต่อกับปฏิสัมพันธ์กับผู้ใช้ผ่านทางเบราว์เซอร์ได้ ที่เราเรียกว่า Dynamic และ Interactive ได้ ASP จะทำงานในลักษณะ Server - Side ถ้าคุณเคยใช้ฟังก์ชันของ JavaScript ในเอกสาร HTML สำหรับทำให้เกิดรูปแบบพิเศษต่าง ๆ หรือการตรวจสอบค่าต่าง ๆ ในฟอร์ม ซึ่งเราเรียกลักษณะการทำงานของ Script นี้ว่า Client - Side ทั้งนี้เพราะว่ามันจะทำการ download Script เหล่านั้นมาพักที่เอกสาร และทำการเอ็กซ์คิวต์ ในเบราว์เซอร์ฝั่ง Client แต่ถ้าเป็นลักษณะการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานของ Server - Side นั้น จะทำการเอ็กซ์คิวต์ ทางฝั่ง Server - Side โดยไม่ต้อง download Script มาที่เบราว์เซอร์เลย

ลักษณะที่สำคัญและโดดเด่นของ ASP มีด้วยกัน 4 อย่าง นั้นคือ

1. Active Server Page สามารถบรรจุ Script ที่ใช้ประมวลผลทาง Server ได้ การทำเช่นนี้ได้ทำให้เกิดประโยชน์มาก เพราะจะทำให้เราสามารถสร้างหน้าเอกสารที่เป็น dynamic ได้ตัวอย่างง่าย ๆ ที่แสดงถึงประโยชน์ของคุณลักษณะนี้คือ เราสามารถที่จะสร้างเอกสาร (Page) ที่แสดงข้อความทักทายที่แตกต่างกันไปในแต่ละเวลาของวันได้

2. Active Server Page ได้เตรียม built - in object มากมาย การที่มี built - in object ใน Active server page ช่วยให้ Script ของเรามีประสิทธิภาพในการใช้งานมากขึ้น นั้นเพราะ Object ต่างๆ เหล่านี้ จะทำให้เราสามารถรับ-ส่งข้อมูล (Data) ระหว่าง Server กับ Client (Browser) ได้ ตัวอย่างเช่น การใช้ Object "Request" เราสามารถรับข้อมูลจากผู้ใช้ที่ส่งผ่านมาจากฟอร์ม (Form) ของ HTML และข้อมูลนั้นต่อให้กับส่วนของ Script ที่ต้องการได้อย่างได้อย่างง่ายดาย

3. Active Server Page สามารถเพิ่ม Component ที่ต้องการได้ ไม่เฉพาะ Component มาตรฐานที่ Active Server Page ได้เตรียมไว้ตอนที่ Install เท่านั้น Active Server Page ยังสามารถทำการเพิ่ม Component ที่ผู้ใช้ต้องการเข้าไปได้อีก

4. Active Server Page สามารถทำการติดต่อกับฐานข้อมูล (Data Base) ดังเช่น Microsoft SQL Sever หรือ Microsoft Access ได้เป็นอย่างดี โดยการใช้ชุดของ Object พิเศษ (Object เหล่านี้มีเป็นมาตรฐานอยู่แล้วใน Active Server Page) เรียกว่า Active X Data Object (ADO) คุณลักษณะเหล่านี้ทำให้ Active Server Page มีประสิทธิภาพมากที่สุดในการที่จะนำไปใช้งาน

ดังนั้นด้วยคุณลักษณะเด่นทั้ง 4 ข้อ ที่กล่าวมาทำให้กล่าวได้ว่า Active Server Page นั้นคือหน้าเอกสาร HTML (page) มาตรฐานที่ได้เพิ่มการทำงานของ Script ที่ประมวลผลทาง Server โดยมี Object และ Component เพิ่มเข้ามาช่วยในการทำงานทำให้สามารถสร้าง Web Site ที่มีหน้าเอกสารแบบ Dynamic ได้

2.3.1.2 The Active Server Page Model & Architecture

การเขียน Asp

Active Server Page (ASP) คือไฟล์ที่สร้างขึ้นมีนามสกุลเป็น .asp ซึ่งภายในไฟล์ .asp นั้น ประกอบไปด้วยข้อความต่าง ๆ (text), คำสั่งของ HTML และคำสั่งของ Script ASP ไม่ใช่ภาษาสคริปต์ในรูปแบบของ Active Server page นั้นเราสามารถจะใช้โปรแกรม Notepad ในการสร้างก็ได้ หรือจะใช้เครื่องมือช่วยเช่น Visual Interdev ส่วนภาษาสคริปต์จะใช้ อะไรก็ได้เช่น JAVA, C++ , Visual Basic, Perl, VB Script, Jscript จะสังเกตได้ว่า ASP นั้นมีความยืดหยุ่นในการเขียนมาก ในเนื้อหาทั้งหมดนี้ผมจะใช้ VBScript เป็นหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ายังจำได้ว่าในการเขียนคำสั่ง HTML นั้น เราใช้ <และ> ปิดท้ายของ Tag แต่ใน ASP นั้น เราจะใช้ <%และ%> ปิดท้ายของสคริปต์ ซึ่งจะช่วยให้ ASP interpreter รู้ว่าเป็นสคริปต์ของ ASP ซึ่งจะถูกระมวลผลที่ Web server เท่านั้น แต่ถ้าเราต้องการให้ผลลัพธ์ของการประมวลผลสคริปต์ ไปแสดงที่เบราว์เซอร์ของผู้ใช้ เราจะปิดท้ายของ สคริปต์ด้วย <%= และ %> บางครั้งเราจะใช้ <%@ กับ %> ปิดท้าย ซึ่งเป็นการบอกให้ Web Server รู้ว่าเป็น แอปพลิเคชัน ASP

ออบเจกต์ภายใน Active Server Page

ที่ถูกนำมาใช้สร้างและจัดการแอปพลิเคชันทั้งหมด 5 ชนิด

1. Request Object

เป็นออบเจกต์ ที่เก็บข้อมูลทั้งหมดของผู้ใช้ผ่านทางเบราว์เซอร์ โดยข้อมูลที่ถูกส่งมานั้นมี 2 วิธี คือ POST ใช้ร่วมกับแบบฟอร์มซึ่งไม่จำกัดความยาวของข้อมูล และใช้วิธี GET ซึ่งข้อมูลสามารถส่งได้ โดยไม่ต้องสร้างแบบฟอร์มหรือจะสร้างก็ได้ ข้อมูลยาวไม่เกิน 255 ตัวอักษร โดยใช้ QueryString ในการรับค่าเช่น

Request.Form("name") เมื่อใช้สำหรับส่งข้อมูลแบบ โปส

Request.QueryString("Address") เมื่อใช้สำหรับส่งข้อมูลแบบ GET

นอกจากนี้ยังมี Collection ที่สำคัญอีก 1 ตัว คือ Server Variable ซึ่งมีหน้าที่เก็บตัวแปรของ WebServer ซึ่งมีข้อมูลหลายอย่างที่เป็ประโยชน์ในการสร้าง Application ASP

2. Respond object

เป็นออบเจกต์ ที่ใช้ส่งข้อมูลไปยังผู้ใช้ปรากฏทางเบราว์เซอร์ เมธอดของ Respond Object ที่ใช้กันมากที่สุดคือ Write Method สำหรับการเขียนสตริงส่งไปยังเบราว์เซอร์ เช่น

```
<%Respond.Write("Hello ASP")%>
```

นอกจากนี้ยังมีเมธอดอื่น ๆ บางตัวที่ใช้กันเช่น Redirect Method ในการย้ายเบราว์เซอร์ไปอ่านข้อมูลยัง Web Page อื่นเช่น

```
<%respond.redirect("search.ASP")%>
```

3. Sesssion Object

เป็นออบเจกต์ ที่ใช้ในการเก็บข้อมูลต่าง ๆ เกี่ยวกับผู้ใช้ของ Session นั้น โดยจะใช้ Cookie ในการติดตามการใช้งานของผู้ใช้แต่ละราย โดยจะส่งเลขที่ หรือ ID ที่กำหนดเฉพาะให้กับผู้ใช้แต่ละรายส่งไป-กลับ ระหว่างเบราว์เซอร์และ Web Server เพื่อคอยติดตามการใช้งานแอปพลิเคชัน ASP โดยมี Property ที่สำคัญคือ Session ID ใช้แสดง id เฉพาะที่ถูกกำหนดให้ผู้ใช้แต่ละราย และ Time out ซึ่งเป็นเวลาที่ WebServer จะจัดการติดต่อกับผู้ใช้ หากไม่สามารถติดต่อกับ Session ID ที่มันสร้างขึ้นมา มีหน่วยเป็น นาที ปกติคือ 20 นาที

4. Application Object

เป็นออบเจกต์เดี่ยว ๆ ทำหน้าที่จัดการแอปพลิเคชัน ASP ซึ่ง Application ASP จะถูกเรียกใช้ โดยการเรียกไฟล์ .asp มาใช้งาน นอกจากนี้ยังทำหน้าที่เสมือนกับแชร์ข้อมูลระหว่างส่วนต่าง ๆ ของแอปพลิเคชัน ASP โดยข้อมูลที่นำมาแชร์จะเก็บอยู่ในรูปของตัวแปรที่กำหนดขึ้น Property และ collection ต่าง ๆ ตัวอย่างข้าง

ล่างนี้เป็นตัวอย่างการใช้งานของ Collection content ซึ่งเป็นการแสดงค่าตัวแปรทุกตัวของ Application ออปเจ็ค โดยใช้ For Each.Next และแสดงจำนวนครั้งที่เข้ามาของเอกสาร ว่าเป็นครั้งที่เท่าไร

5. Server Object

เป็น ออปเจ็ค ที่ใช้จัดการและบริหารของ WebServer มีเมธอด ที่สำคัญ 2 ตัวดังนี้คือ

-CreateObject เป็นการเรียกใช้ Active Object โดยอาศัยแนวคิดของคอม หรือเรียกการใช้งาน ADO ใน Web server ดังตัวอย่างข้างล่างเป็นตัวอย่างการใช้ Server Object สร้างการติดต่อกับฐานข้อมูล

```
<%Set Conn=Server.CreateObject("ADODB.Connection")
Conn.open "DSN"
%>
```

-MapPath เป็นการกำหนด Virtual Path เข้ากับพาทที่มีจริงใน Web server เช่น

```
<%=Server.mappath("Scripts")%>
จะแสดง c:\inetpub\wwwroot\scripts
<%=server.mappath("/scripts")%>
จะแสดง c:\inetpub\scripts
```

2.3.1.3 ADO Object Model

ปัจจุบันการใช้งานอินเทอร์เน็ตโดยเฉพาะ WWW เป็นอะไรที่มากกว่าการขอข้อมูลจาก Server หนึ่งในการใช้งานที่เพิ่มขึ้น และมีความสำคัญมากขึ้น คือ การเรียกค้นข้อมูลจากฐานข้อมูล Active Server Page มีวิธีการติดต่อกับฐานข้อมูล โดยใช้ Server side component ที่เรียกว่า Active X Data Object (ADO) โดยมีตัวกลางการเชื่อมต่อระหว่างแอปพลิเคชัน และฐานข้อมูลนั้นคือ Open Database Connectivity หรือ ODBC

ADO Object Model

Active X Data Object คือชุดของ Object ที่ถูกนำมาใช้ในการติดต่อกับ Database Server สำหรับ ASP โดยใช้ ADODB สร้าง Object ที่เกี่ยวข้องกับการติดต่อและดึงข้อมูล จากฐานข้อมูลเพื่อนำมาใช้งาน โดยมี Object ต่าง ๆ ที่เกี่ยวข้องกับ ADO ดังนี้

1. Command Object

เป็น Object คำสั่งที่ถูกส่งไปเพื่อเอ็กซ์คิวทีฟ ให้ได้เรคคอร์ดเซต กลับมาเป็นผลลัพธ์ และยังสามารถในการกำหนดพารามิเตอร์ต่าง ๆ ไปกับการส่งคำสั่งปกติได้

2. Connection Object

เป็นออปเจ็ค ที่ถูกสร้างขึ้นเพื่อเชื่อมต่อกับฐานข้อมูล

3. Recordset Object

เป็นออปเจ็ค ที่ได้มาจากการส่งคำสั่งเพื่อให้ได้เรคคอร์ดเซตเป็นผลลัพธ์กลับมา

การเชื่อมต่อฐานข้อมูล

ก่อนที่จะใช้งาน ADO เราจำเป็นต้องสร้าง Data Source Name (DSN) เสียก่อน โดยเซต Open DataBase Connectivity (ODBC) ให้เชื่อมไปยังฐานข้อมูลที่เราเตรียมไว้ ซึ่งขั้นตอนการเซตมีดังนี้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. เปิด Control Panel → 32 bit ODBC ของเครื่องที่ทำ Web server
2. เลือกแท็บ System DSN แล้วคลิกปุ่ม Add
3. เลือกไดรเวอร์ของฐานข้อมูลที่ต้องการเชื่อมต่อ ในที่นี้เลือก Microsoft Access driver (*.mdb) แล้วคลิก Finish
4. กรอกรายละเอียดในช่อง Data Source Name ให้ตั้งชื่อของ DSN แล้วคลิกปุ่ม Select เลือก MS

การใช้งาน Connection Object

การที่จะติดต่อกับฐานข้อมูล เราต้องสร้างส่วนการเชื่อมต่อหรือ Connection ก่อน โดยสร้าง วัตถุ เชื่อมต่อ สำหรับการติดต่อ แล้วใช้เมธอด Open เปิดการเชื่อมต่อ เช่น

```
Set conn=Server.CreateObject("ADODB.Connection")
```

```
Conn.Open"MYDSN"
```

บรรทัดแรกเป็นการสร้างวัตถุ เชื่อมต่อ สำหรับการติดต่อชื่อ Conn ไว้ก่อน จากนั้นใช้เมธอด Open เปิดการเชื่อมต่อ กับ Data Source Name ชื่อ MYDSN

การใช้งาน Recordset Object

เมื่อสร้างวัตถุ เชื่อมต่อแล้ว เราสามารถสร้าง Record Object เพื่อเอ็กซ์คิวต์ คำสั่งในการดึง ข้อมูลจากฐานข้อมูลมาได้ เช่น

```
Set Rs=Server.Create ("ADODB.Recordset")
```

```
Rs.Open"Select*From mytable Where id<10"
```

2.3.2 WAP

2.3.2.1 มารู้อีกกับ WAP

WAP เป็นมาตรฐานที่สร้างขึ้นจากความนิยม หรือ ที่เรียกกันว่า de-facto standard เพื่อทำหน้าที่นำเสนอ และ รับ-ส่งข้อมูลข่าวสาร ผ่านระบบการสื่อสารแบบไร้สาย ซึ่งอยู่ภายใต้การควบคุม และดูแล โดย WAP Forum ซึ่งเป็นกลุ่มของทีมงานที่ถูกจัดตั้งขึ้นมาจาก บริษัท Ericsson, Motorola, Nokia และ Phone.Com (ชื่อเก่าคือ Unwired Planet) ในเดือนมิถุนายน ปี 1997 และนับจากนั้นจำนวนสมาชิกก็เพิ่มขึ้นเรื่อยๆ ทั้งที่มา จาก ผู้ให้บริการ โทรศัพท์มือถือ, ผู้ผลิต, หรือ แม้แต่กลุ่มนักพัฒนาโปรแกรมเองเช่นกัน

WAP Forum เป็นผู้ที่ร่างและออกแบบรายละเอียดของมาตรฐาน WAP โดยพยายามที่จะใช้มาตรฐานเก่าที่มีอยู่ ให้ได้ประโยชน์มากที่สุด และ จะเพิ่มเติมในรายละเอียดเมื่อจำเป็นเท่านั้น ซึ่งเป็นสิ่งที่ดีที่จะช่วยให้ผู้ผลิต โทรศัพท์มือถือ, ผู้วางระบบเครือข่าย, ผู้ให้บริการ โทรศัพท์ และ นักพัฒนาโปรแกรมทั้งหลาย สามารถทำงาน ร่วมกันได้อย่างไม่มีอุปสรรค และ ไม่จำเป็นที่จะต้องเปลี่ยน โครงสร้างของเทคโนโลยีมากนัก ซึ่งเราสามารถ เขียนจุดประสงค์ของ WAP Forum ได้เป็นหัวข้อย่อยๆ ได้ดังนี้

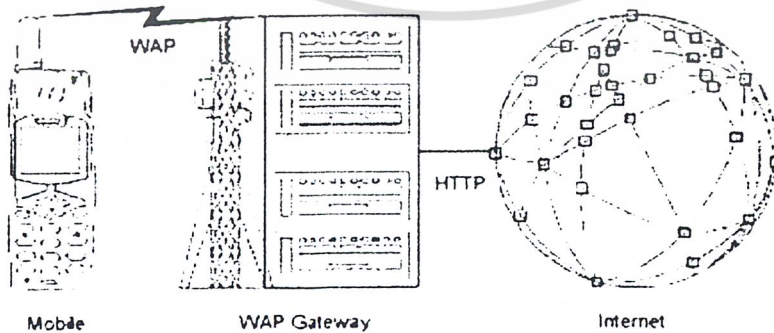
1. ความพยายามที่จะนำเนื้อหา และบริการที่มีอยู่ในอินเทอร์เน็ต ออกนำเสนอบนเครื่อง โทรศัพท์มือถือ และ อุปกรณ์ไร้สายอื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. มีจุดประสงค์ในการออกแบบ โครงสร้างของโปรโตคอลที่ใช้กับการสื่อสารแบบไร้สาย ที่สามารถใช้งานได้ดีกับเครือข่ายไร้สายในทุกเทคโนโลยี
3. ความพยายามที่จะสนับสนุนให้เนื้อหาและบริการที่ต้องการ ถูกนำออกมาใช้ได้กับระบบการสื่อสารแบบไร้สายชนิดใดก็ได้ โดยไม่จำกัด รวมทั้งไม่จำกัดชนิดของเครื่องโทรศัพท์มือถือด้วย เช่น GSM, CDMA หรือ IDEN
4. ความพยายามที่จะรวบรวมนำเทคโนโลยีที่มีอยู่มาใช้ เช่นการทำงานในฝั่งของเซิร์ฟเวอร์ และ จะเพิ่มเติมในรายละเอียดเมื่อจำเป็นเท่านั้น

หลักการการทำงานของ WAP

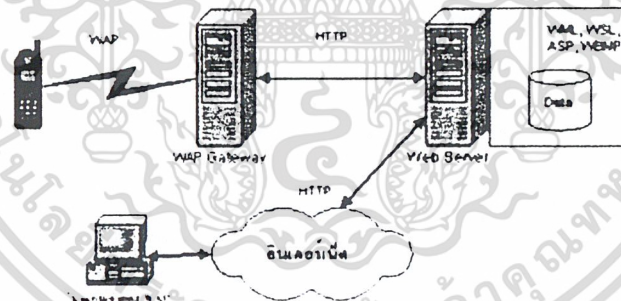
WAP มีการทำงานในลักษณะเดียวกับเว็บ คือ เป็นแบบไคลเอ็นท์ และ เซิร์ฟเวอร์ (Client-Server) แต่ WAP นั้นจะมีเครื่องมือพิเศษเข้ามาทำงานเป็นตัวกลางเชื่อมต่อระหว่าง โปรโตคอล WAP และ HTTP ซึ่งก็คือ WAP เกตเวย์นั่นเอง



เอกสารนี้เป็นเอกสารที่สงวน **รูปที่ 2.1 การทำงานของ WAP** ษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของ WAP บนโทรศัพท์มือถือสามารถอธิบายเป็นขั้นตอนย่อยๆ ได้ดังนี้

1. เมื่อผู้ใช้ต้องการเปิด WAP ไซต์ที่อยู่ในอินเทอร์เน็ตขึ้นมาชมในโทรศัพท์มือถือ ผู้ใช้จำเป็นต้องใส่ค่า URL (Uniform Resource Locator) ที่ต้องการลงไป เช่น <http://www.thaiwarp.com>
2. เครื่องโทรศัพท์มือถือจะทำหน้าที่แปลงข้อมูลของ URL พร้อมทั้งบีบอัดไฟล์ ให้เป็นเลขฐานสอง หรือไบนารีไฟล์ เช่น <http://www.thaiwarp.com> 10011010111 ซึ่งจะช่วยในการประหยัดแบนวิดธ์
3. เครื่องโทรศัพท์มือถือส่งไบนารีไฟล์นั้นไปยัง WAP เกตเวย์ และเกตเวย์เปลี่ยนคำร้องขอข้อมูล (Request) ของ WSP เป็น HTTP
4. จากนั้นจึงเกตเวย์จะส่ง HTTP Request ไปในเครือข่ายอินเทอร์เน็ต
5. ทางด้านฝั่งผู้ให้บริการ WAP ก็จะมีเซิร์ฟเวอร์ที่มี WAP :เพจ เขียนด้วย WML เมื่อเซิร์ฟเวอร์ของ www.thaiwarp.com ได้รับคำร้องขอ หรือ HTTP Request จากเกตเวย์ ก็ทำการส่งไฟล์ WML กลับไป
6. เมื่อเกตเวย์ได้รับไฟล์ WML กลับมาจากเซิร์ฟเวอร์ ก็ทำการเปลี่ยนแปลงข้อมูลเป็นไบนารีไฟล์ เพื่อเตรียมส่งข้อมูลกลับ ไปหาเครื่องโทรศัพท์มือถือนั้นโดยใช้โปรโตคอล WAP
7. เมื่อเครื่องโทรศัพท์มือถือรับไบนารีไฟล์นั้นแล้ว ก็ทำการแปลงข้อมูลเป็นภาษา WML ที่ WAP บราวเซอร์ในเครื่องสามารถอ่านได้



รูปที่ 2.2 การใช้โปรแกรมจำลองในการชม WAP เพจ

แต่ที่จริงแล้ว นอกจากเราจะสามารถเข้าชม WAP ไซต์ โดยใช้โทรศัพท์มือถือแล้ว เรายังสามารถใช้โปรแกรมจำลอง หรือ ที่เรียกกันว่า Emulator มาท่องโลกของ WAP ก็ได้ เพราะมีโปรแกรม WAP บราวเซอร์อยู่ในโปรแกรมเหล่านั้นเอง หรือ แม้แต่เราจะเข้าไปชมจากในเว็บ ไซต์ที่มีโปรแกรมเหล่านี้ก็ใช้ได้

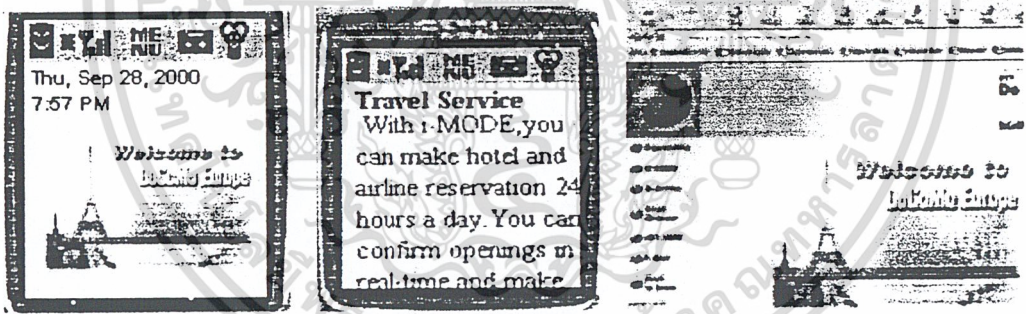


WAP เป็นเทคโนโลยีใหม่เช่นเดียวกับ อินเทอร์เน็ต ปัจจุบันโปรแกรมที่ใช้งานกับ WAP เริ่มที่จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากขึ้นอย่างเป็นลำดับใน อินเทอร์เน็ต ถึงแม้ว่าในบางครั้ง นักพัฒนาโปรแกรมทั้งหลายจะรู้สึกถึงความยากลำบากบ้างในการเขียน โปรแกรม WAP ก็ตาม หรือแม้ว่า มันจะเป็นไปได้โดยยากที่จะทำให้ระบบการสื่อสารแบบไร้สายนี้สามารถรับรอง การทำงานแบบมัลติมีเดีย (Multimedia) ได้ WAP ก็ยังมีความสามารถอื่นๆ อีกที่จะนำพาตัวเองไปสู่ความสำเร็จ โดยเฉพาะอย่างยิ่งความสามารถในการพกพา(Mobility) ที่สะดวกสบาย และพร้อมเสมอที่จะนำพาผู้ใช้เข้าสู่โลกอินเทอร์เน็ต และ ความสามารถในการตรวจสอบผู้ใช้บริการ (Authentication) ที่จะสามารถเรียกเก็บเงินค่าบริการต่างๆที่ผู้ใช้ ใช้ได้อย่างถูกต้องและรวดเร็ว ซึ่งถือว่าเป็นจุดขายหลักของ WAP นั่นเอง เช่น เมื่อเรากรอจองตั๋วภาพยนตร์โดยใช้ WAP แล้ว เราก็สามารถจ่ายค่าตั๋วผ่านทางผู้ให้บริการโทรศัพท์มือถือ โดยรวมไปกับค่าโทรรายเดือนเลยก็ได้ โดยไม่จำเป็นต้องให้เลขเครดิตการ์ดเหมือนในอินเทอร์เน็ตอีกเลย

สิ่งหนึ่งที่เป็นสิ่งสำคัญอย่างยิ่งที่จะผลักดันให้อินเทอร์เน็ตของ WAP สดใส ก็คือ การที่ WAP มีโปรแกรมในการให้บริการแก่ผู้ใช้ที่หลากหลายนั่นเอง ดังเช่น ในปัจจุบันนี้มีเว็บไซต์หลายแห่งได้สร้าง WAP ไซต์ขึ้นมาเพื่อให้บริการควบคู่ไปกับ เว็บไซต์ของตนเองที่มีอยู่แล้ว เช่น <http://www.wopwap.com> ก็จะมี WAP ไซต์ <http://wap.wopwap.com> หรือ <http://www.yahoo.com> ก็จะมี <http://wap.yahoo.com> นั่นเอง ซึ่งอาจจะสังเกตได้ว่า WAP ไซต์นั้น จะมีคำว่า wap นำหน้าชื่อโดเมนเนม ซึ่งจะบอกว่าถ้าเราเปิดดู URL นี้เราก็จะได้ข้อมูลที่ เป็น WAP ไซต์ แต่อย่างไรก็ตามบางเว็บไซต์ก็ยังคงชื่อเดิมของเขาไว้แต่จะสร้าง WAP ไซต์ ในไคลเอนทอร์ย่อยต่อมาเป็น www.domainname.com/wap ก็ได้



รูปที่ 2.3 แสดงการให้บริการ WAP

เมื่อพูดถึงอนาคตของ WAP แล้ว เราก็อดที่จะกล่าวถึงเทคโนโลยีคู่แข่งที่น่ากลัวอันหนึ่งมิได้ เทคโนโลยี นั้นก็คือ i-mode นั่นเอง i-mode เป็นเทคโนโลยีที่เปิดตัวในประเทศญี่ปุ่น โดยบริษัท NTT DoCoMo เป็นผู้บุกเบิก ซึ่งมีผู้ใช้บริการไปแล้วมากกว่า 10 ล้านคน ตั้งแต่เริ่มเปิดตัวเมื่อ เดือน กุมภาพันธ์ 1999 i-mode มีบริการที่คล้ายคลึงกับ WAP บนเครื่องโทรศัพท์มือถือ เช่น การจองตั๋วภาพยนตร์, การตรวจสอบสภาพอากาศ และ การทำค้นหาข้อมูลในอินเทอร์เน็ต

ภาษาที่ใช้ในการพัฒนาระบบ i-mode ก็คือ ภาษา cHTML (Compact HTML) ซึ่งมีลักษณะคล้ายคลึงกับภาษา HTML แต่จะตัดแค่บางส่วนที่ที่ไม่จำเป็นต้องออกไป ดังนั้นข้อดีของ i-mode ก็คือ เราสามารถใช้เว็บเบราว์เซอร์ที่ดูง่าย ทั้งสิ้น อีกทั้งยังมีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บราวเซอร์ เปิด i-mode คูได้เช่นเดียวกับ บราวเซอร์ของ i-mode เอง แต่อย่างไรก็ตามข้อเด่นของ WAP ที่มีเหนือ i-mode ก็คือ ในอนาคตเว็บ ไซต์ส่วนใหญ่จะเริ่มเปลี่ยนแปลงตัวเองเข้าสู่ภาษา XML และ WAP เอง ก็เกิดจาก WML ที่ส่วนที่คล้ายคลึงกับ XML มาก ดังนั้นถ้าเราทราบ WML แล้ว การทำงานในอนาคตร่วมกับเว็บ ไซต์ที่เขียนขึ้นโดย XML ก็เป็นไปได้โดยง่าย ท่าทีของไมโครซอฟต์ (MicroSoft) ที่มีต่อ WAP ก็เป็นอีกหนึ่งตัวแปรที่เราไม่ควรมองข้ามเลย ถึงแม้ว่าไมโครซอฟต์นั้นจะเป็นสมาชิกหนึ่งของ WAP Forum แต่ก็ได้พยายามที่จะสร้าง โปรแกรมของตัวเองขึ้นมาเพราะ ไมโครซอฟต์พยายามที่จะนำเอาโปรแกรม Window CE ไปเป็นระบบปฏิบัติการบนเครื่องอุปกรณ์ไร้สายด้วยเช่นกัน แต่สิ่งที่เป็นอุปสรรคความคิดของไมโครซอฟต์นั้น ก็คือความสามารถของตัวเครื่องโทรศัพท์มือถือที่ตัวเอง ที่ว่าจะเป็นไปได้มากน้อยเพียงใดที่จะสามารถรับระบบปฏิบัติการดังกล่าวได้ และถ้าเป็นไปได้ HTML ก็ยังคงครองตลาดอินเตอร์เน็ต และ เครื่องอุปกรณ์ไร้สายอยู่เช่นเดิม



รูปที่ 2.4 ระบบเครือข่ายของ WAP

การใช้ CGI(Common Gateway Interface) ทำงานร่วมกับ WML จะทำให้ข้อมูลที่เรากำลังต้องการเปลี่ยนแปลงบ่อยๆ สามารถทำงาน ได้ง่ายขึ้น โดยเฉพาะอย่างยิ่งการแสดงข้อมูลที่มีลักษณะของการต้องการที่จะ update อยู่ตลอดเวลา

โปรแกรมพวก CGI หรือในที่นี้เราจะกล่าวถึง ASP(Active Server Page) นั้น จะยังมีความสำคัญมากขึ้น เพราะเมื่อมีการเปลี่ยนแปลงของข้อมูล โปรแกรมนี้จะทำหน้าที่สร้างข้อมูลออกมา ให้ผู้เยี่ยมชมเห็น โดยอัตโนมัติ ดังเช่น โปรแกรมแสดงผลของการ Vote ซึ่งสามารถเปลี่ยนแปลงข้อมูลได้ และอีกหนึ่งสิ่งที่โปรแกรม CGI มีความสำคัญก็คือ ความสามารถในการติดต่อกับฐานข้อมูล เพื่อนำข้อมูลที่สำคัญ มาแสดงออก หรือ ทำหน้าที่เก็บข้อมูลที่สำคัญลงฐานข้อมูลนั่นเอง (ดังในรูปด้านบนนั้น เมื่อบราวเซอร์ของเราต้อง ทำการติดต่อกับเซิร์ฟ

เอกซามป์เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ใช้เฉพาะที่องค์กรนั้นเท่านั้น ไม่สามารถนำข้อมูลไปใช้ประโยชน์อื่น การค้าไม่ว่ากรณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวอร์ CGI จะทำหน้าที่เก็บและแสดงข้อมูลที่ได้รับเปลี่ยนแปลงเหล่านั้นออกมา) ซึ่งเราจะรู้จักกันดีถ้าเราได้เคยเขียน Web Based Application ที่ทำงานเกี่ยวกับ Form

ในส่วนนี้ขอพูดถึงคำสั่ง ASP อย่างง่าย ๆ ก่อน ที่มีความจำเป็นในการรับ-ส่ง ข้อมูล
<%...%> แท็ก ASP ทุกแท็กจำเป็นต้องอยู่ภายใต้แท็กนี้

Request เป็น Object ที่ใช้ในการเรียกขอข้อมูลจาก Browser เข้าหา Server มีฟังก์ชันที่สำคัญดังนี้
Request.Form เป็นการเรียกเก็บข้อมูลที่มาจากการ Post ข้อมูลของ Form ที่มากับ HTTP มีการใช้งานดังนี้
<%ชื่อตัวแปร=Request.Form("ชื่อวัตถุ")%> เป็นการเรียกเก็บข้อมูลจากวัตถุในฟอร์มมาเก็บไว้ในตัวแปร
Request.Query เป็นการรับค่าของตัวแปรที่มากับ HTTP Header ที่ตามหลัง เครื่องหมาย?

นอกจากนี้เรายังสามารถเก็บค่าตัวแปรของ ServerVariables, ClientCertificate, และ Cookies ซึ่ง รายละเอียดคลิก
ซึ่งลงไปเราคงไปศึกษาเองได้จาก Help หรือ หนังสือ ASP ทั่วๆ ไป

อีกหนึ่ง Object ที่ควรทราบก็คือ Response ที่ใช้ในการเขียน Output สู่อุปกรณ์ลูกข่าย เช่น Cookie, ข้อความ, หรือ
ค่าของตัวแปร

Response.write ใช้ในการเขียนข้อความ หรือ ค่าของตัวแปร (เราสามารถให้ Server แสดงค่าของตัวแปร

ได้เช่นเดียวกัน โดยใช้แท็ก<%=ชื่อตัวแปร%> เช่น <%=Time%> จะเป็นการแสดงค่าของเวลาออกมา
สู่หน้าจอ ซึ่ง Time เป็นฟังก์ชันที่ ASP มี (เราสามารถเรียกใช้ฟังก์ชันของ VBScript ใน ASP ได้ด้วยนะครับ)

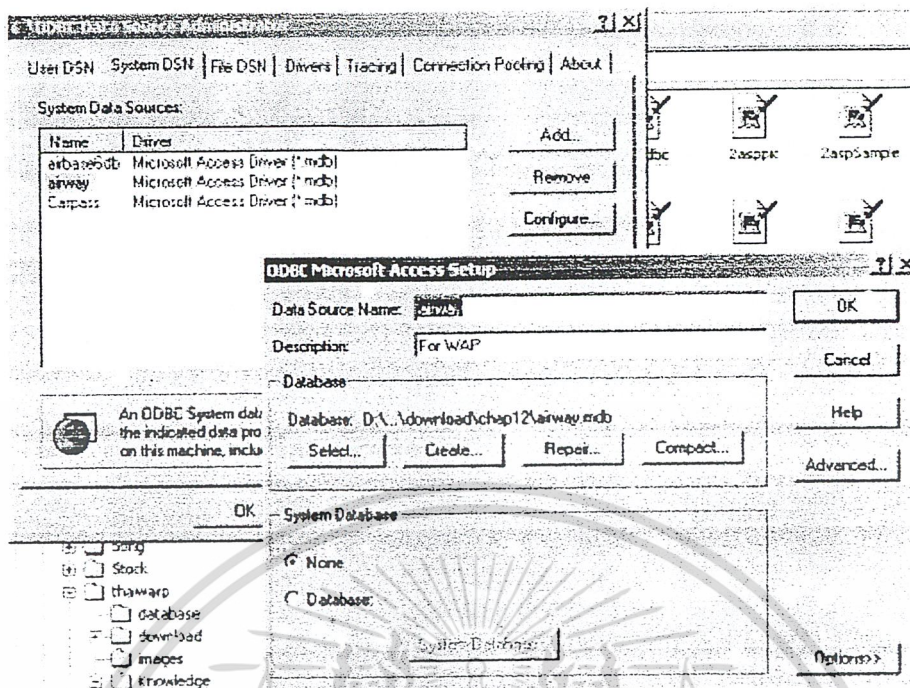
Response.ContentType="text/vnd.wap.wml"

อีกหนึ่งคุณสมบัติที่สำคัญของ Response ใน WML ก็คือ เราสามารถส่งชนิดของไฟล์ ไปกับ HTTP ได้ เช่น
เอกสารเราเป็นไฟล์ .asp แต่สามารถเปิดใน WAP Browser ได้ก็เพราะเราได้บอก Browser ก่อนว่า Server จะ
แปลเอกสาร .asp นี้ไปเป็นไฟล์ชนิด "text/vnd.wap.wml" ให้นั่นเองครับ

2.3.2.2 ODBC กับ WAP

การนำเสนอข้อมูลจากฐานข้อมูลที่เราสามารถ Update ได้ตลอดเวลา ออกสู่ WAP เป็นเรื่องที่น่าสนใจ
ยิ่ง การเชื่อมต่อฐานข้อมูลเข้ากับ ASP นั้นเราสามารถทำได้ทั้งแบบ DSN (Data Source Name) และ DSNLess

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 การสร้าง ODBC ใน Control Panel

อย่างไรก็ตามการเชื่อมต่อฐานข้อมูลแบบไม่ผ่าน DSN จะเป็นวิธีที่สะดวกต่อการใช้งานมากกว่าโดยเฉพาะ ผู้ที่ไม่มีเซิร์ฟเวอร์เป็นของตัวเอง แต่อาจจะเช่าพื้นที่จากผู้ให้บริการ web hosting เพราะการพื้นฐานข้อมูลผ่าน DSN จะต้องขออนุญาตจากผู้ดูแลเซิร์ฟเวอร์ เพื่อขอให้เขาสร้าง DSN มายังไฟล์ที่ต้องการ ซึ่งจะสร้างความยุ่งยาก และเสียเวลาของทั้งสองฝ่าย

ในกรณีที่ web hosting บางแห่ง เช่น www.thaihostserver.net หรือที่อื่นๆทั่วไป ซึ่งสามารถใช้งานฐานข้อมูลของโปรแกรม Microsoft Access ได้ และสามารถเชื่อมต่อฐานข้อมูลแบบ DSNLess ได้ เราก็น่าจะใช้การเชื่อมต่อแบบนี้

```
Set Conn=Server.CreateObject("ADODB.Connection")
Conn.Open "DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=" &
Server.MapPath("database/warp.mdb"), "", ""
แต่ถ้าเราไม่ใช่คำสั่ง Server.MapPath เราก็จะต้องเขียน Path เต็มๆของที่อยู่ของฐานข้อมูล ดังนี้ Conn.Open
"DRIVER={Microsoft Access Driver (*.mdb)}; DBQ=C:/inetput/wwwroot/wap/database/warp.mdb", "", ""
```

ซึ่งทั้งสองตัวอย่างดังกล่าวสามารถนำมาเปรียบเทียบกับแบบเชื่อมต่อกับ DSN ได้ดัง

```
Set Conn=Server.CreateObject("ADODB.Connection")
Conn.Open "ชื่อ DSN"
```

2.3.2.3 การส่งข้อมูลเข้าหา WAP

การส่งข้อมูลจาก Form บนโทรศัพท์มือถือของเราเข้าสู่เครื่องเซิร์ฟเวอร์ ซึ่งเราอาจจะต้องใช้ไฟล์ที่เป็น CGI มาเป็นตัวรับข้อมูล อีกทีหนึ่ง เพื่อนำข้อมูลนั้น ไปใส่ลงในฐานข้อมูล(database) หรือ ส่งต่อไปหาอีเมลล์ของเจ้าของ WAP หน้าอื่นๆก็ได้ ในบทความนี้เราจะมาศึกษาถึง วิธีการส่งข้อมูลซึ่งมีอยู่หลักๆ 2 วิธีคือ Get และ Post

การส่งข้อมูลแบบ GET

การส่งข้อมูลแบบ GET สามารถทำได้สองวิธีคือ ส่ง ไปในแอตทริบิวต์ href ของแท็ก go หรือส่ง โดย ใช้แท็ก postfield การส่งข้อมูล ไปในแอตทริบิวต์ href มีรูปแบบดังนี้

```
<go href="ชื่อไฟล์จัดการข้อมูล?ชื่อตัวแปร1=ค่าของตัวแปร1&ชื่อตัวแปร2=ค่าของตัวแปร2&..."/>
```

เช่น

```
<go href="doLogin.asp?name=Jane&password=ffgrio"/>
```

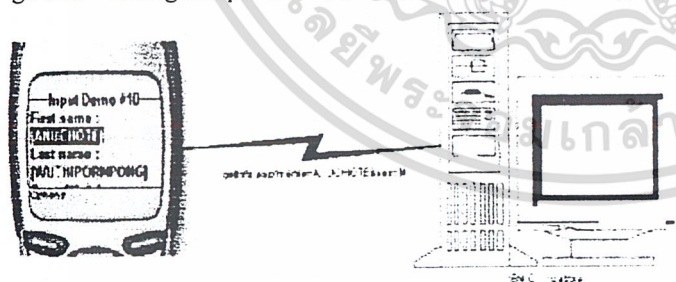
เราจะเห็นว่าตัวแปรแรกจะต้องต่อจากเครื่องหมายคำถาม(?) และตัวแปรต่อๆ มาจะต้องขึ้นด้วยเครื่องหมาย และ(&)

เนื่องจากการระบุค่าตัวแปร ในเอกสาร WML มีรูปแบบดังนี้ \$(ชื่อตัวแปร) และเครื่องหมายและ(&) ก็ เป็นเครื่องหมายพิเศษซึ่งจะต้องใช้กลุ่มอักษรแทนคือ & amp; ดังนั้นรูปแบบการส่งข้อมูลจึงเป็นดังนี้

```
<go href="ชื่อไฟล์จัดการข้อมูล?ชื่อตัวแปร1=$(ชื่อตัวแปร1)& amp; ชื่อตัวแปร2=$(ชื่อตัวแปร2)& amp;..."/>
```

เช่น

```
<go href="doLogin.asp?name=$(name)& amp;password=ffgrio"/>
```



รูปที่ 2.6 แสดงการเข้าถึงข้อมูลของ WAP

ตัวอย่าง

```
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.2//EN"
"http://www.wapforum.org/DTD/wml12.dtd">
<wml>
<card id="card1" title="Input Demo #10">
<do type="accept" label="Submit">
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<go href="getInfo.asp?name=${(name)&sex=${(sex)"/>
</do>
<p>
Input your info :<br/>
<fieldset title="name">
First name : <input type="text" name="fname"/><br/>
Last name : <input type="text" name="lname"/><br/>
</fieldset>
<fieldset title="sex">
Sex :
<select name="sex" multiple="false">
<option value="M">Male</option>
<option value="F">Female</option>
</select>
</fieldset>
</p>
</card>
</wml>

```

การส่งข้อมูลโดยใช้แท็ก postfield

เราสามารถ ใช้แท็ก postfield ที่มีพิเศษ ในภาษา WML ในการส่งข้อมูลแบบ GET ได้ด้วย โดยมีรูปแบบให้ดังนี้

```

<go href="ชื่อไฟล์จัดการข้อมูล" method="get">
<postfield name="ชื่อตัวแปร" value="ค่าตัวแปร"/>
</go>

```

เราจะเห็นว่าการระบุวิธีการส่งนั้นจะทำได้ในแท็ก go ด้วยแอตทริบิวต์ method ส่วนแท็ก postfield นั้น เป็นที่ระบุชื่อตัวแปรและค่าของมัน

การส่งข้อมูลแบบ POST

การส่งข้อมูลแบบ POST ทำได้โดยใช้แท็ก postfield และ go เช่นเดียวกับการส่งแบบ GET แต่เปลี่ยนค่าแอตทริบิวต์ method ของแท็ก go เป็น post ดังนี้

```

<go href="ชื่อไฟล์จัดการข้อมูล" method="post">
<postfield .../>

```

...

```

</go>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ขั้นตอนการสร้างและการดำเนินงาน

3.1 แนวคิดและการดำเนินงาน

3.2 การสร้างวงจรควบคุมอุปกรณ์ในบ้าน

3.3 การสร้างซอฟต์แวร์เพื่อใช้ในการควบคุม

3.1 แนวคิดและการดำเนินงาน

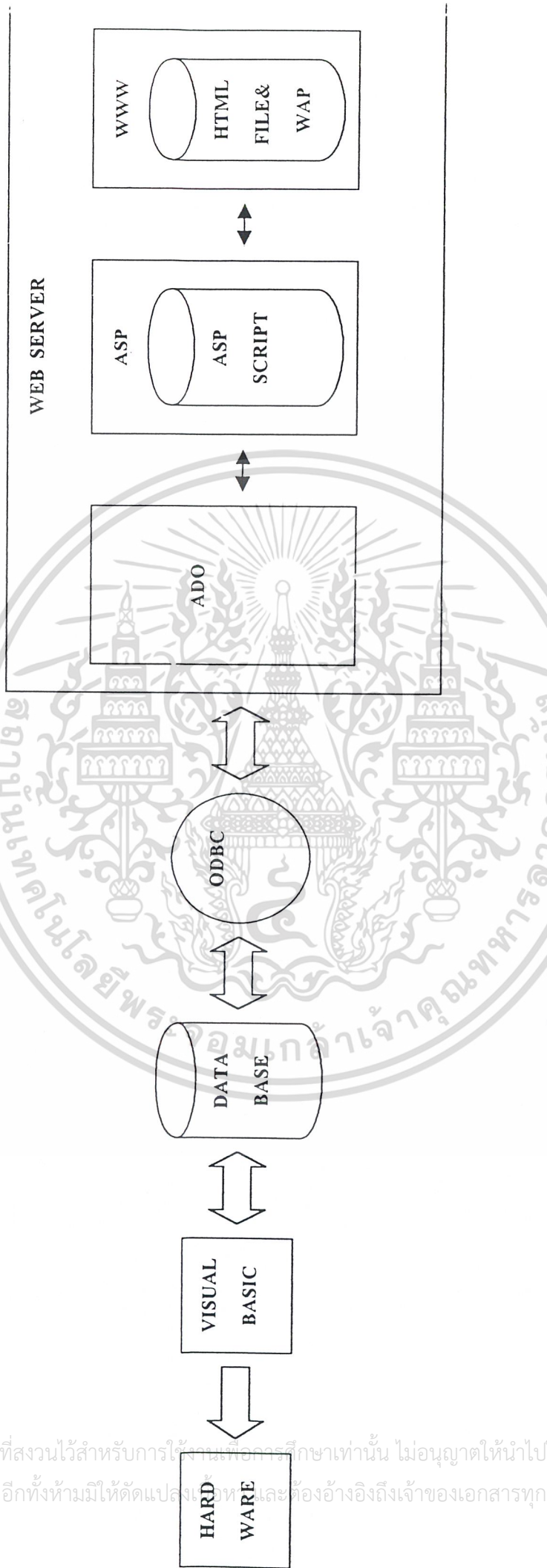
3.1.1 FLOW CHART และ BLOCK DIAGRAM แนวคิดการทำงาน

- **BLOCK DIAGRAM**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

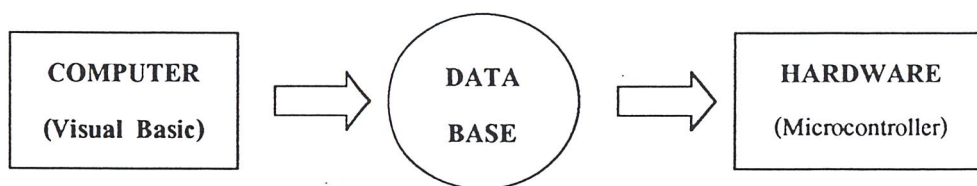
OVERVIEW - SYSTEM



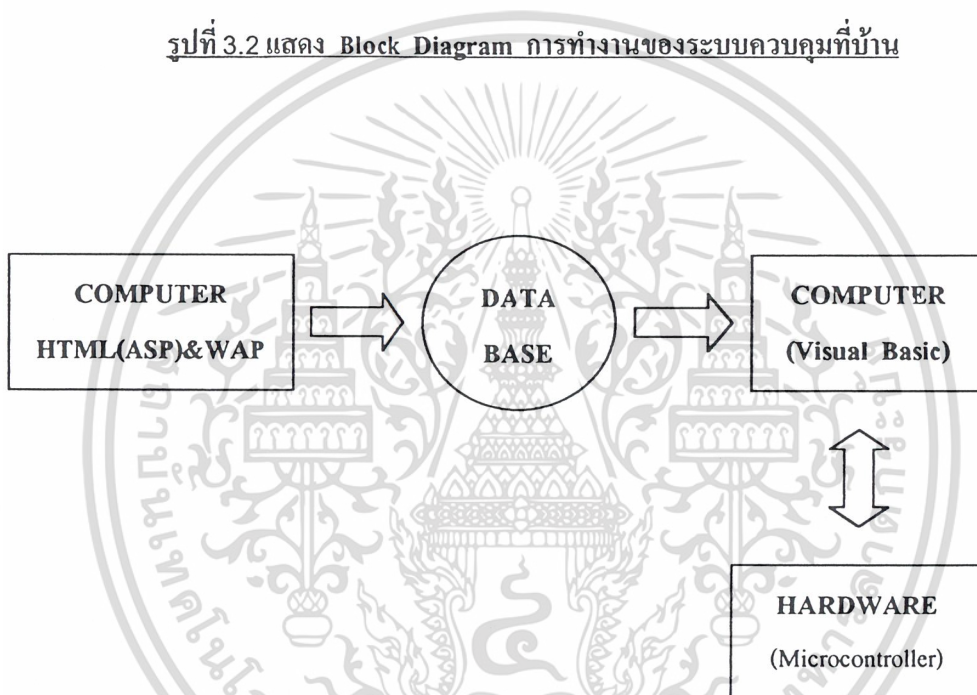
รูปที่ 3.1 แสดง BLOCK DIAGRAM ของระบบทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงแก้ไข และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• **BLOCK DIAGRAM** ของการควบคุมการทำงาน



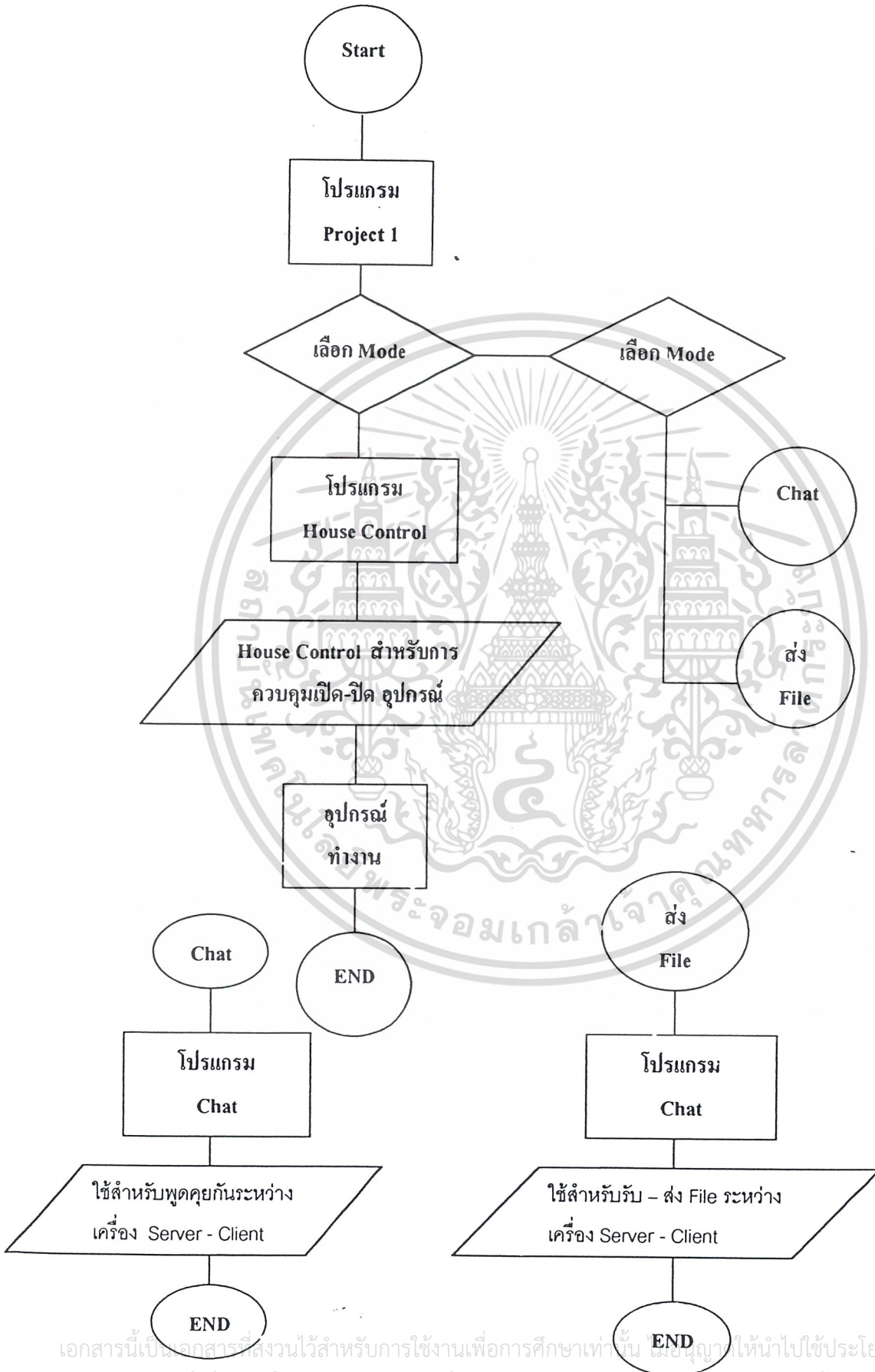
รูปที่ 3.2 แสดง Block Diagram การทำงานของระบบควบคุมที่บ้าน



รูปที่ 3.3 แสดง Block Diagram การทำงานของระบบควบคุมผ่านอินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

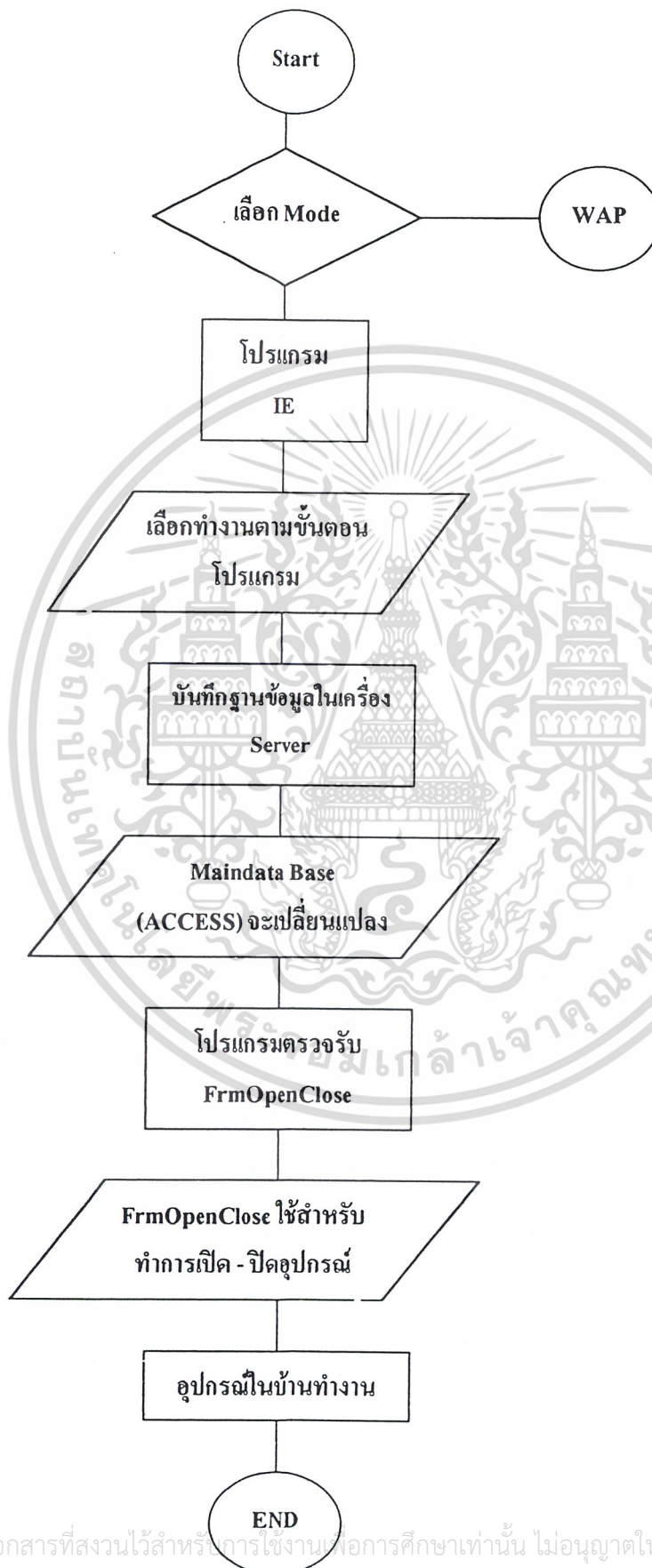
1. การควบคุมคอมพิวเตอร์ที่บ้าน



รูปที่ 3.4 FLOW CHART สำหรับระบบควบคุมที่บ้าน

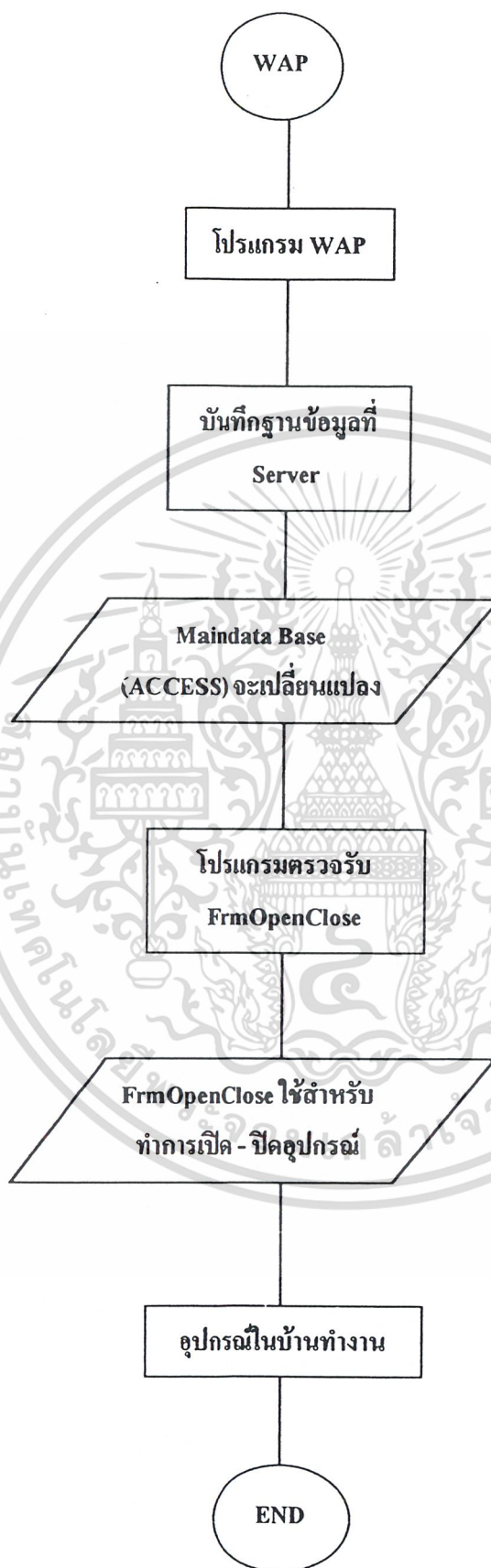
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ความคุมจาก Internet



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 FLOW CHART ของการควบคุมผ่านอินเทอร์เน็ต 1



รูปที่ 3.6 FLOW CHART ของระบบควบคุมผ่านอินเทอร์เน็ต 2

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการศึกษาค้นคว้าเท่านั้น เมื่อผู้จัดทำเห็นว่าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การสร้างวงจรมอบอุปกรณ์ในบ้าน

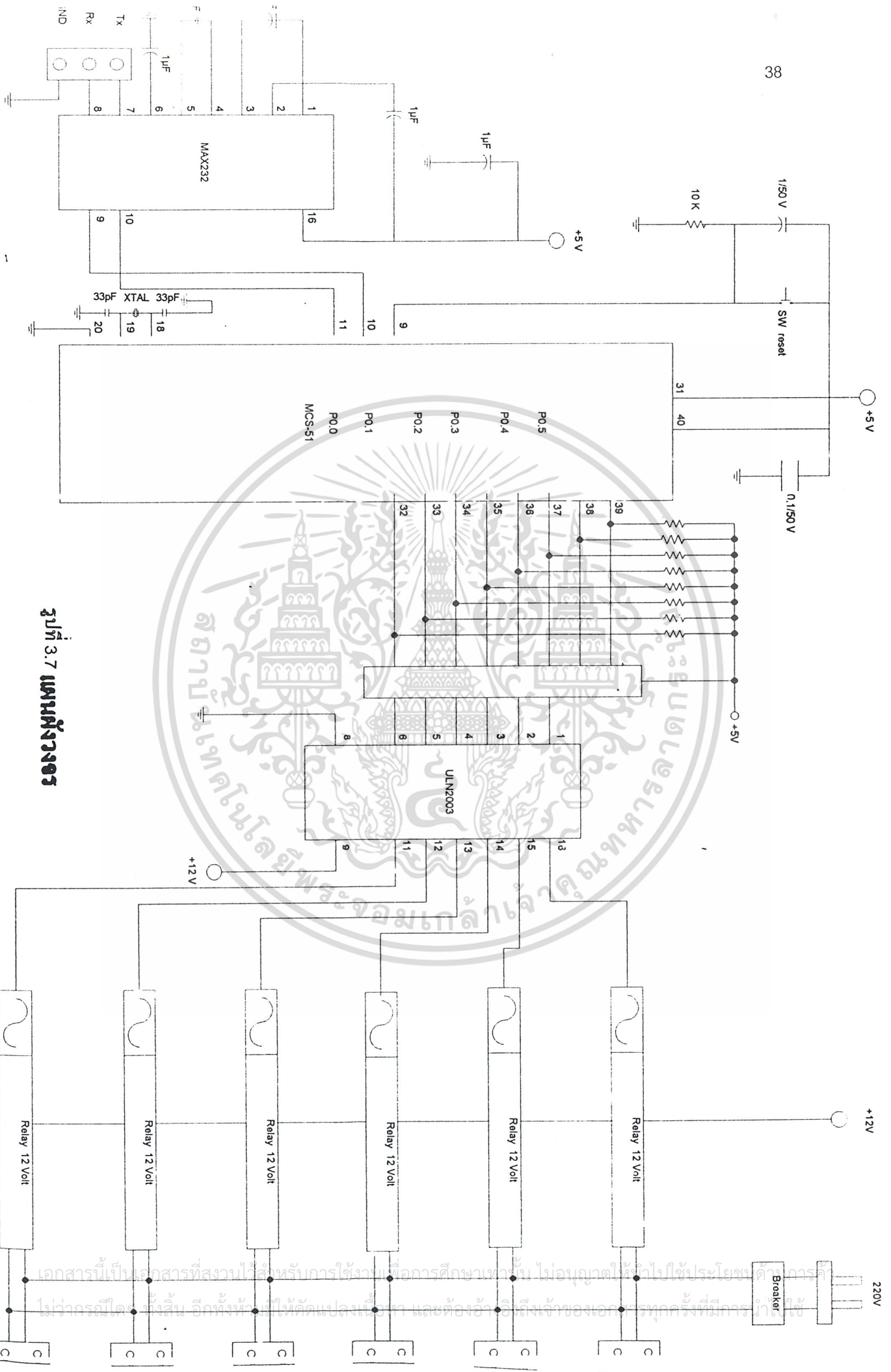
3.2.1 การสร้างวงจรและอุปกรณ์

อุปกรณ์ในการสร้างวงจรมอบอุปกรณ์ในบ้าน

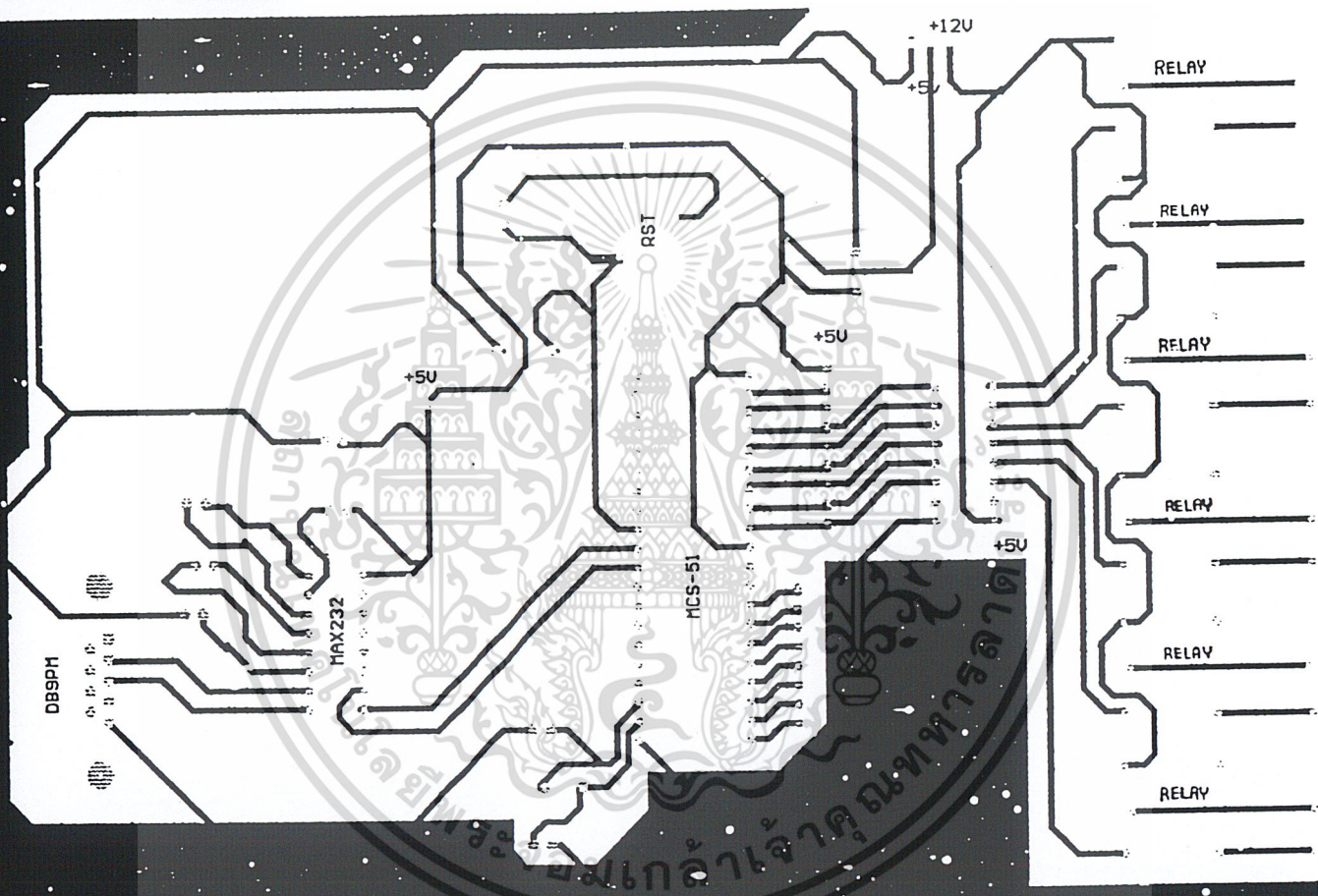
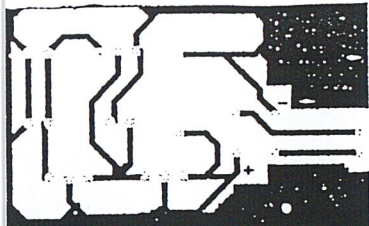
1. AT89C 51	1	ตัว
2. ULN 2003	2	ตัว
3. MAX 232	1	ตัว
4. IC 7812	2	ตัว
5. IC7805	2	ตัว
6. RELAY (แบบ 5 ขา 12 Volt)	6	ตัว
7. ปลั๊กตัวเมีย	6	ตัว
8. หม้อแปลง (12 Volt 500 mA)	2	ตัว
9. DB 9 (ตัวเมีย)	1	ตัว
10. XTAL 11.0592 MHz	1	ตัว
11. Switch Reset	1	ตัว
12. Bridge	1	ตัว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



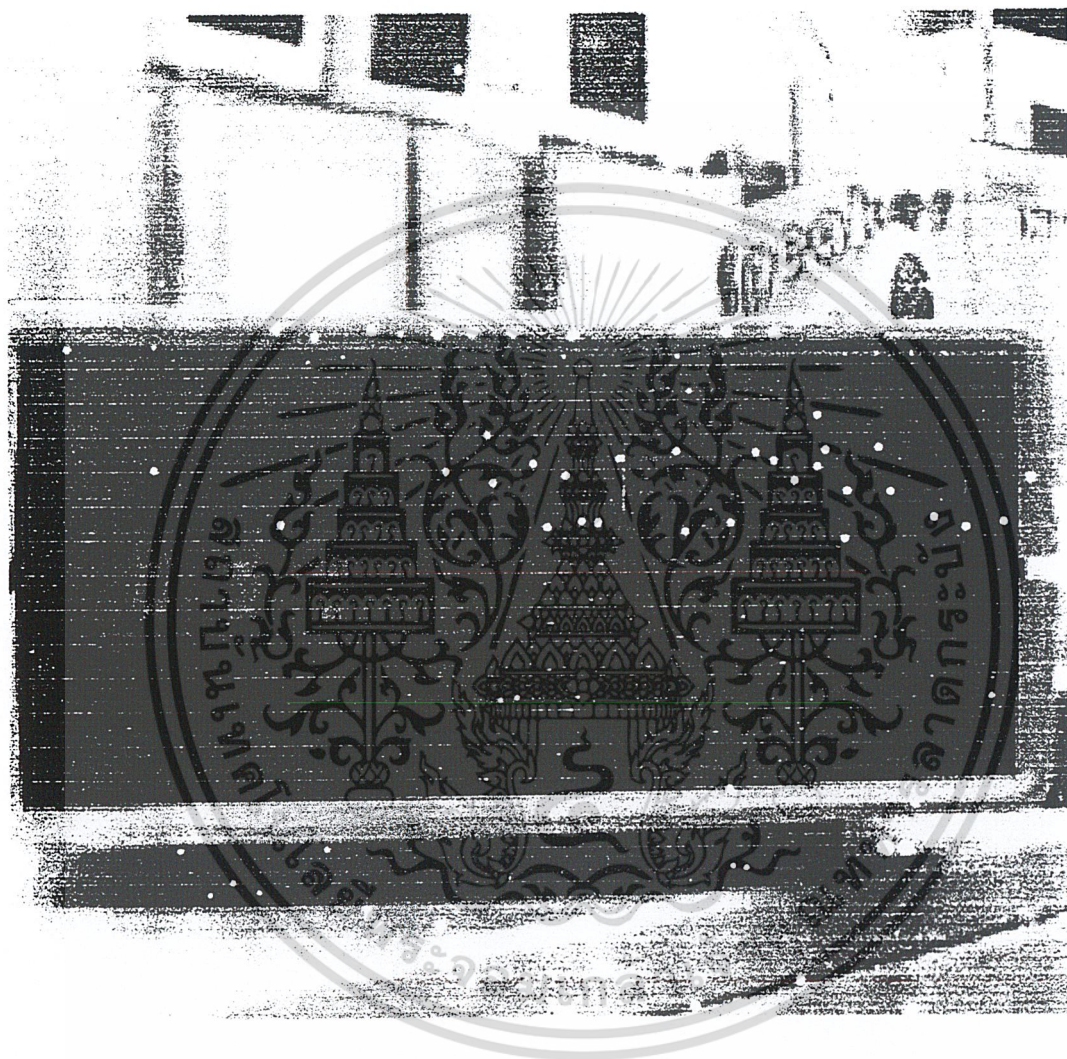
รูปที่ 3.7 แผงรีเลย์



รูปที่ 3.8 ลายปริ้นต์

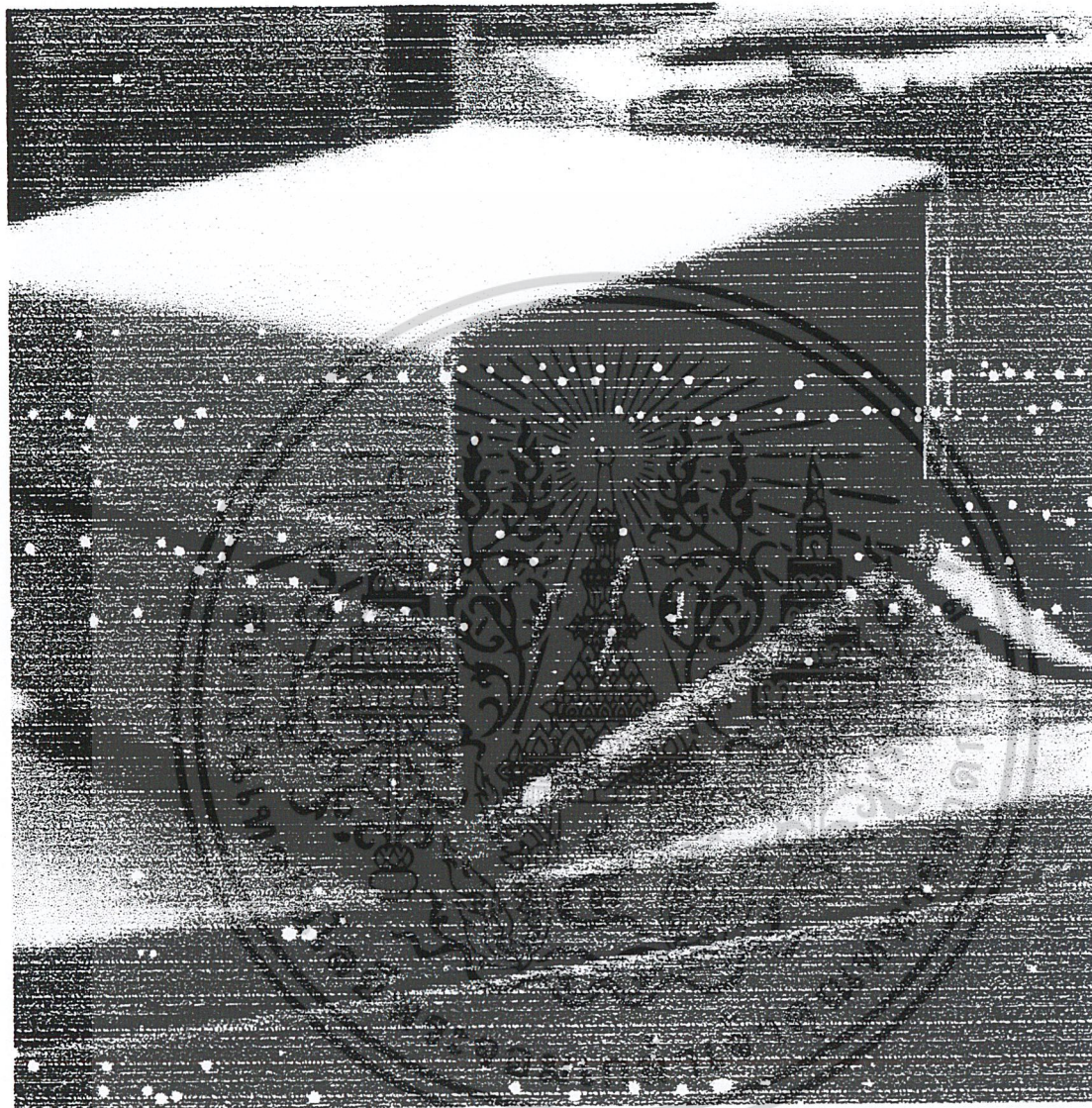
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ชิ้นงาน



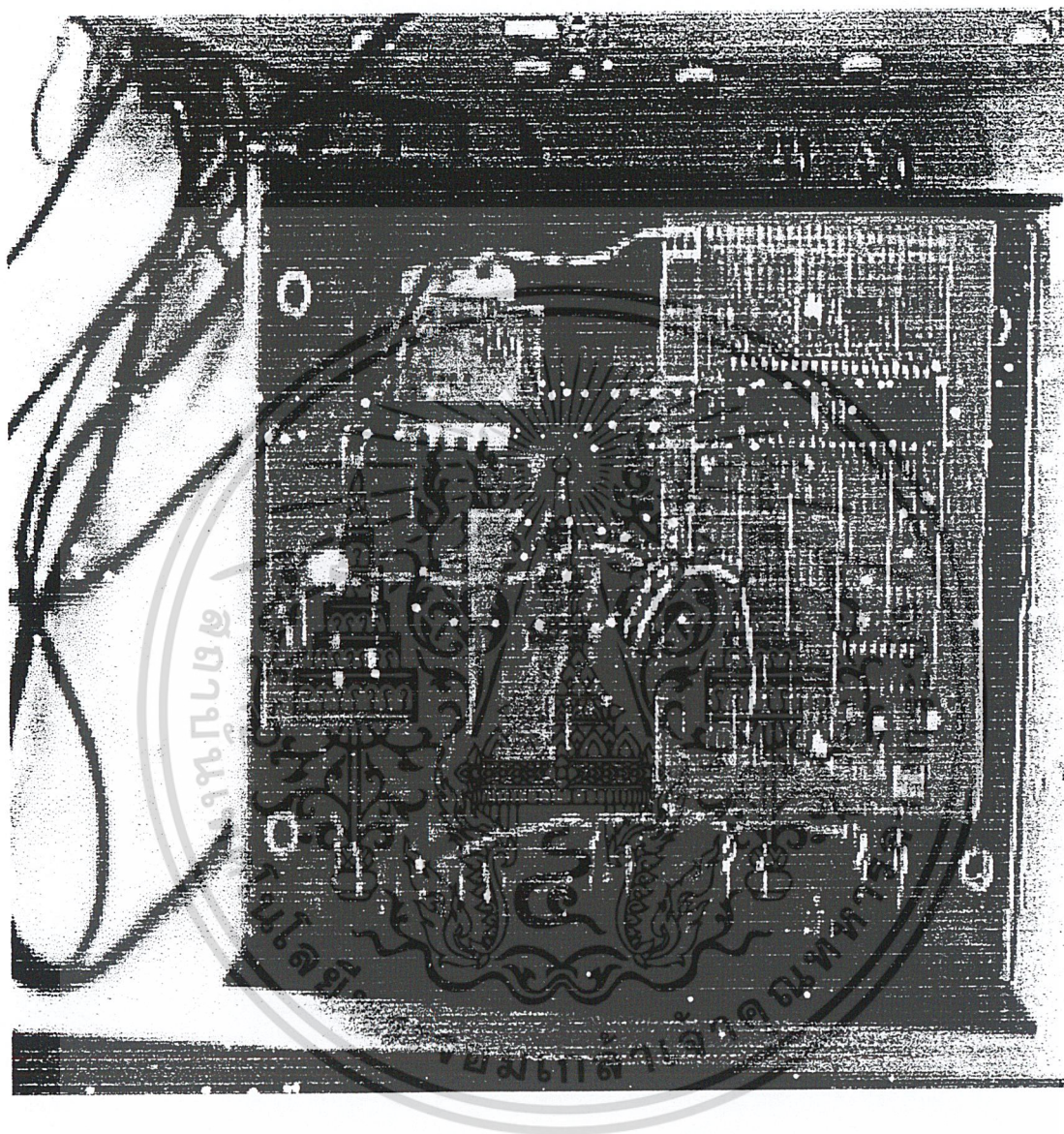
รูปที่ 3.9 แสดงด้านหน้าของเครื่องควบคุมอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



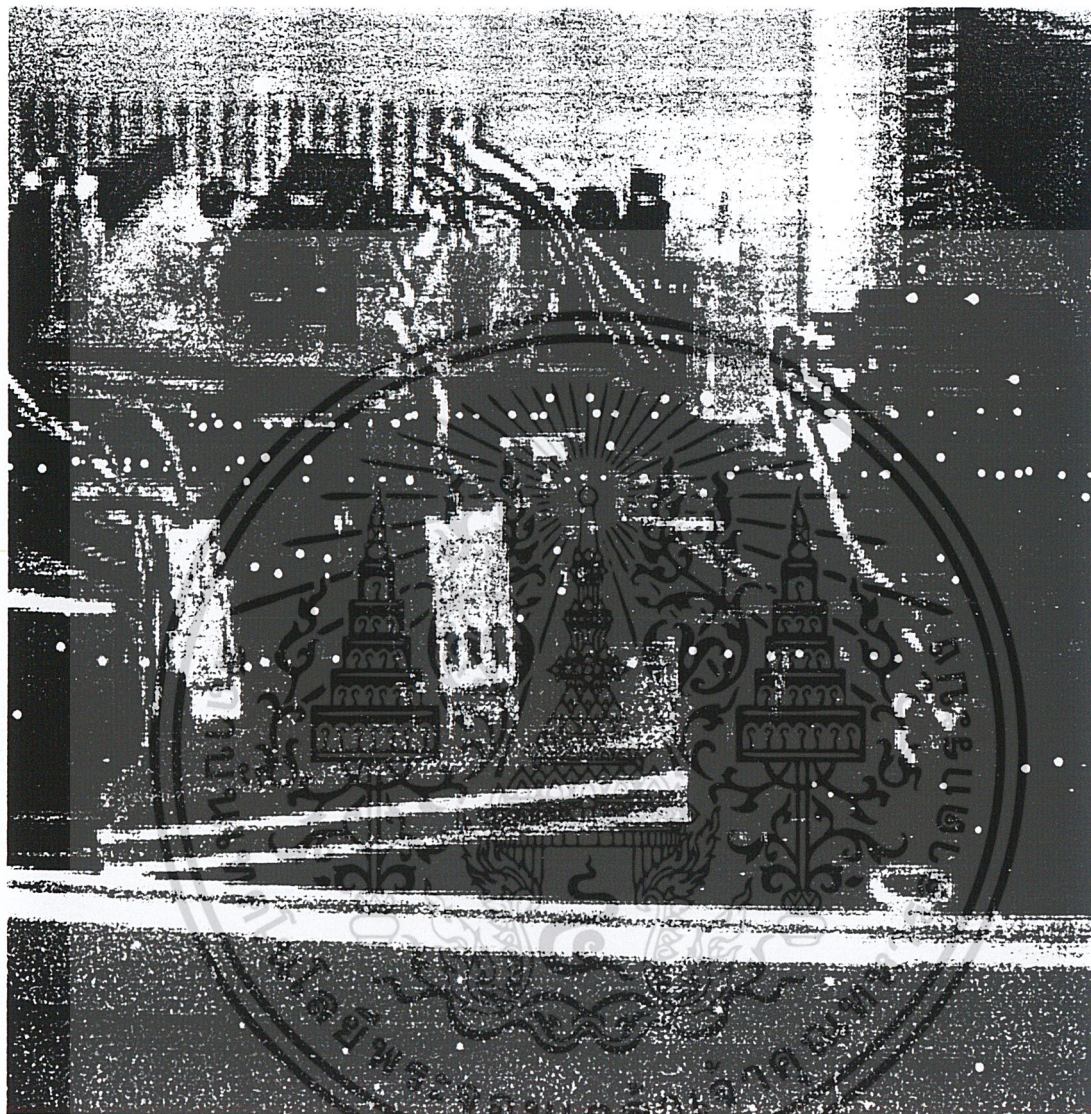
รูปที่ 3.10 แสดงด้านหลังของเครื่องควบคุมอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงภายในของเครื่องควบคุมอุณหภูมิ(ด้านบน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงด้านในของเครื่องควบคุมอุปกรณ์(ด้านข้าง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การสร้างซอฟต์แวร์สำหรับควบคุมวงจร

- Source Code เขียนด้วยภาษา Assembly
- โปรแกรม Project 1 เขียนด้วย Visual Basic 6.0
- WAP และ ASP

3.3.1 Source Code เขียนด้วยภาษา Assembly

3.3.1.1 Source Code

```

                                ORG 0000H

                                MOV  PCON, #000H
                                MOV  SCON, #50H
                                MOV  TMOD, #20H
                                MOV  TH1, #0FDH
                                SETB TR1
                                MOV  P1, #00H

MAIN:  MOV  A, #00H
TEST:  JNB  RI, TEST
       MOV  A, SBUF
       CLR  RI
LED_1: CJNE A, #31H, LED_2
       CPL  P1.0
LED_2: CJNE A, #32H, LED_3
       CPL  P1.1
LED_3: CJNE A, #33H, LED_4
       CPL  P1.2
LED_4: CJNE A, #34H, LED_5
       CPL  P1.3
LED_5: CJNE A, #35H, LED_6
       CPL  P1.4
LED_6: CJNE A, #36H, LED_7
       CPL  P1.5
LED_7: CJNE A, #37H, LED_8
       CPL  P1.6
LED_8: CJNE A, #38H, TXD_1
       CPL  P1.7
TXD_1: MOV  SBUF, A
WAIT:  JNB  TI, WAIT
       CLR  TI
       LJMP MAIN
END

```

3.3.1.2 ขั้นตอนการใช้งาน

1. บันทึกโปรแกรมเป็น 23.asm
2. นำไปทดสอบกับโปรแกรม Rchip51
3. ใช้โปรแกรม et-afpv ในการ burn โปรแกรมลงใน microcontroller (AT89C51)
4. นำไปใช้กับ"วงจรควบคุมอุปกรณ์ในบ้าน"(ดังที่แสดงในรูปที่ 13

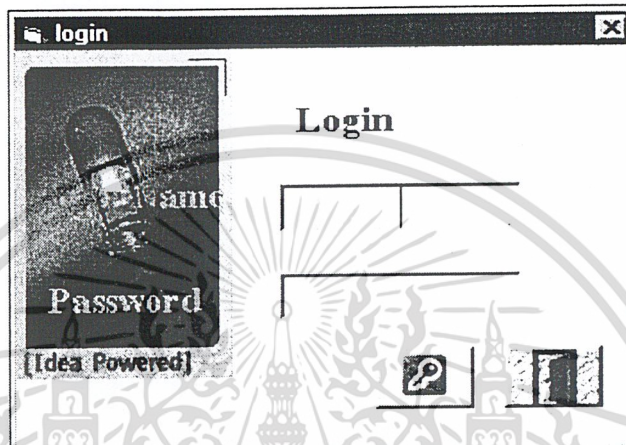
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การสร้างโปรแกรม Project 1 เขียนด้วย Visual Basic 6.0

3.3.2.1 หน้าตาของโปรแกรมและ Source Code

1. LOGIN FORM(frmlogin.frm)

- เป็นหน้าจอของโปรแกรมเพื่อเข้าสู่ระบบการใช้งาน



รูปที่ 3.13 แสดงหน้าจอของโปรแกรม Login

Source Code

```
Option Explicit
Dim answer As Variant

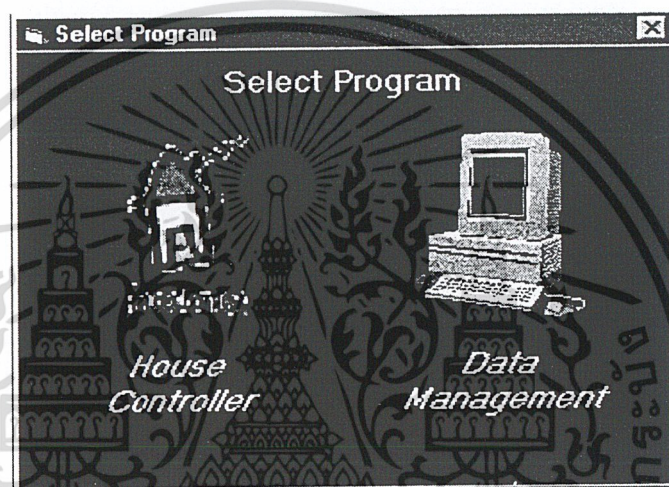
Private Sub cmdExit_Click()
End
End Sub

Private Sub cmdlogin_Click()
If txtinputusername.Text = "Tanagrit" Then
    If txtinputpassword.Text = "attoas" Then
        Unload frmlogin
        frmselectprogram.Show vbModal
    Else
        answer = MsgBox("Password Error please try agian", vbYesNo
+ vbQuestion, "Login Error")
        If answer = vbNo Then
            Unload frmlogin
            End If
        End If
    Else
        answer = MsgBox("UserName Error please try agian", vbYesNo +
vbQuestion, "Login Error")
        If answer = vbNo Then
            Unload frmlogin
            End If
        End If
    End Sub
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. SELECTPROGRAM FORM(frmsselectprogram.frm)

- เป็นหน้าจอของโปรแกรมในการเลือกโหมดการทำงานซึ่งมี 2 โหมด
 1. HOUSE CONTROLLER คือการควบคุมการเปิด-ปิดอุปกรณ์ภายในบ้าน
 2. NETWORK CONTROL คือการสื่อสารระหว่างเครื่อง Server-Client



รูปที่ 3.14 แสดงหน้าของเลือกโหมด Select Form

Source Code

```
Option Explicit
Private Sub imgdatamanagement_Click()
answer2 = MsgBox("คุณต้องการเข้าสู่ ระบบการจัดการข้อมูลภายในเครื่องหรือไม่?", vbYesNo +
vbQuestion, "Are you sure?")
If answer2 = vbYes Then
Unload frmsselectprogram
frmNetwork.Show vbModal
End If
End Sub
Private Sub imghousecontroller_Click()
answer = MsgBox("คุณต้องการเข้าสู่ การควบคุมบ้านหรือไม่?", vbYesNo + vbQuestion, "Are you
sure?")
If answer = vbYes Then
Unload frmsselectprogram
frmhousecontroller.Show vbModeless
End If
End Sub
```

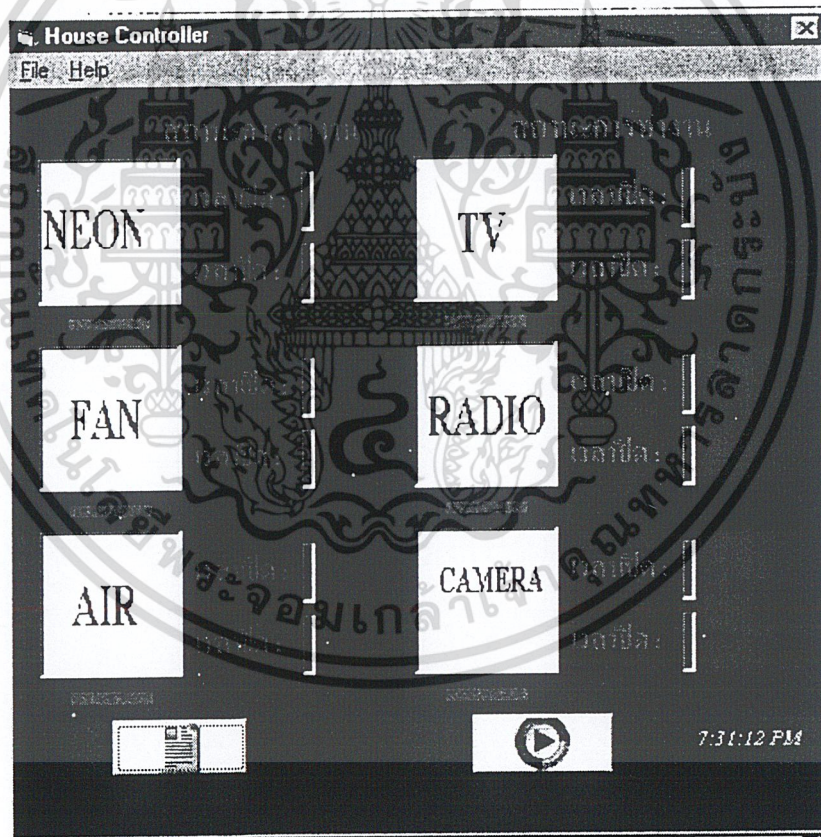
3. HOUSE CONTROLLER FORM(frmhousecontroller.frm)

- เป็นหน้าจอของโปรแกรมในการทำงาน ตั้งเวลาเปิด-ปิด อุปกรณ์จากเครื่องคอมพิวเตอร์ที่บ้าน

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ไม่ว่ากรรมใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. FORM ของการตั้งเวลาของอุปกรณ์นั้น เช่น frmneon.frm
2. FORM ของการยืนยันสถานะของการทำงาน เรียกว่า "CONFIRM FORM"(frmconfirm.frm)
3. FORM ของการตรวจสอบและนับเวลาถอยหลังเพื่อทำการปิดอุปกรณ์นั้นๆ เรียกว่า"OPEN-CLOSE FORM"(frmOpenClose.frm) ซึ่งเป็นตัวกลางของการเชื่อมต่อกับระบบอินเทอร์เน็ต ทำงานร่วมกับ WATCHING FORM เรียกว่า "frmwatch.frm"
4. FORM ของการเชื่อมต่อและสั่งงานกับ Microcontroller เรียกว่า "PROCESS FORM"(frmprocess.frm)
5. FORM ของการตรวจจับการเปลี่ยนแปลงฐานข้อมูลที่เกิดจากการใช้งาน ASP และ WAP เรียกว่า "frmwatch.frm"

หน้าจอแสดงโปรแกรม HOUSE CONTROLLER



รูปที่ 3.15 แสดงหน้าจอโปรแกรม HOUSE CONTROLLER

Source Code

```
Option Explicit
Private Sub cmdconfirm_Click()
frmprocess.Show
frmhousecontroller.Hide
End Sub
Private Sub cmddisplay_Click()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับผู้ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DataReport1.Show
End Sub
-----
Private Sub Command1_Click()
frmtimedelay.Show
End Sub
-----
Private Sub Form_Load()
lblcloseneon.Caption = " "
lblclosefan.Caption = " "
lblcloseair.Caption = " "
lblclosetv.Caption = " "
lblclosecamera.Caption = " "
lblcloseradio.Caption = " "
frmprocess.Show
frmOpenClose.Show
frmwatch.Show
End Sub
-----
Private Sub imgair_Click(Index As Integer)
frmair.Show vbModeless
End Sub
-----
Private Sub imgcamera_Click()
frmcamera.Show vbModeless
End Sub
-----
Private Sub imgfan_Click()
frmfan.Show vbModeless
End Sub
-----
Private Sub imgneon_Click()
frmneon.Show vbModal
End Sub
Private Sub imgradio_Click(Index As Integer)
frmradio.Show vbModeless
End Sub
Private Sub imgtv_Click()
frmtv.Show vbModal
End Sub
-----
Private Sub Select_Click()
frmselectprogram.Show vbModal
End Sub
-----
Private Sub Timer1_Timer()
Label27.Caption = Time()
End Sub
-----

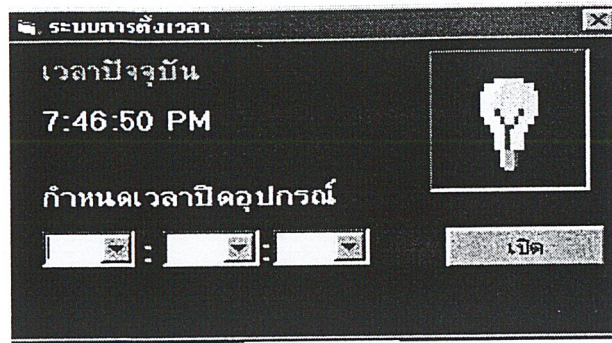
```

3.1 FORM ในการกำหนดเวลาปิด-เปิดอุปกรณ์

- เป็นหน้าจอที่ใช้กำหนด เวลาในการเปิด-ปิดอุปกรณ์ เช่น การเปิด-ปิด หลอดไฟจะใช้

frmneon.frm ในการกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงหน้าจอการกำหนดการเปิด-ปิดหลอดไฟ

Source Code

```
Option Explicit
Public t2
Public t1

Private Sub cmdclose_Click()
frmhousecontroller.lblcloseon.Caption = t1
frmconfirm.cmdesdit.Enabled = False
frmhousecontroller.Shapel.FillColor = &HFF&
frmprocess.MSComm1.Output = "1"
frmOpenClose.Timer1.Enabled = False
Unload Me
frmconfirm.txtdevice.Text = "NEON"
frmconfirm.txttimeon.Text = frmhousecontroller.lblopenneon.Caption
frmconfirm.txttimeoff.Text = frmhousecontroller.lblcloseon.Caption

frmconfirm.Show vbModal
'close neon device 1 port pl.0
End Sub

Private Sub cmdenter_Click()
Dim answer As Variant
Dim h As Integer
Dim m As Integer
Dim s As Integer
Dim timeoff1 As Long
Dim timeoff2 As Integer
Dim timeopen1 As Long
Dim timeopen2 As Integer
Dim alltime As Long
h = Val(cmbhour.Text)
m = Val(cmbminute.Text)
s = Val(cmbsecond.Text)
t2 = TimeSerial(h, m, s)
t1 = Time
If t1 > t2 Then
answer = MsgBox("Your time is invalid", vbQuestion, "Error time")
Else
frmhousecontroller.lblopenneon.Caption = t1
frmhousecontroller.lblcloseon.Caption = t2
frmconfirm.txtdevice.Text = "NEON"
frmconfirm.txttimeoff.Text = t2
frmhousecontroller.Shapel.FillColor = &HFF00&
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

timeoff1 = h * CLng(3600)
timeoff2 = m * 60
timeopen1 = Hour(t1) * CLng(3600)
timeopen2 = Minute(t1) * 60
alltime = timeoff1 + timeoff2 - timeopen2 - timeopen1
frmconfirm.Show vbModal
frmprocess.MSComm1.Output = "1" 'open neon device 1 port pl.0
End If
Unload Me
End Sub

```

```

Private Sub Form_Load()
Dim i As Integer
If frmhousecontroller.Shapel.FillColor = &HFF& Then
cmdenter.Enabled = True
cmbhour.Enabled = True
cmbminute.Enabled = True
cmbsec.Enabled = True
cmdclose.Visible = False
Else
cmdenter.Enabled = False
cmbhour.Enabled = False
cmbminute.Enabled = False
cmbsec.Enabled = False
cmdclose.Visible = True
End If
For i = 1 To 24
cmbhour.AddItem i
Next i
For i = 1 To 60
cmbminute.AddItem i
cmbsec.AddItem i
Next i
End Sub
Private Sub Timer1_Timer()
lbltime1.Caption = Time()
End Sub

```

3.2 CONFIRM FORM(frmconfirm.frm)

- เป็นหน้าจอในการยืนยันคำสั่งในการทำงานจากการกำหนดของ frmneon.frm
- ทำงานร่วมกับ frmOpenClose.frm ในการจับเวลา
- ทำงานร่วมกับ frmProcess.frm ในการควบคุม Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยืนยันหรือทำการเปลี่ยนแปลง

DEVICE

DATE 3/23/02

TIME ON 7:54:52 PM

TIME OFF

▶

⏏

รูปที่ 3.17 แสดงการยืนยันการทำงาน CONFIRM FORM

Source Code

```
Option Explicit
Private Sub cmdadd_Click()

frmOpenClose.txttimeon.Text = txttimeon.Text
frmOpenClose.txttimeoff.Text = txttimeoff.Text
frmOpenClose.txtdevice.Text = txtdevice.Text
Adodcl.Recordset.AddNew
Unload frmconfirm
End Sub

Private Sub cmdesdit_Click()
If txtdevice.Text = "NEON" Then
frmhousecontroller.lblopenneon.Caption = " "
frmhousecontroller.lblcloseonon.Caption = " "
End If
If txtdevice.Text = "AIR" Then
frmhousecontroller.lblopenair.Caption = " "
frmhousecontroller.lblcloseair.Caption = " "
End If
If txtdevice.Text = "TV" Then
frmhousecontroller.lblopentv.Caption = " "
frmhousecontroller.lblclosetv.Caption = " "
End If
If txtdevice.Text = "FAN" Then
frmhousecontroller.lblopenfan.Caption = " "
frmhousecontroller.lblclosefan.Caption = " "
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

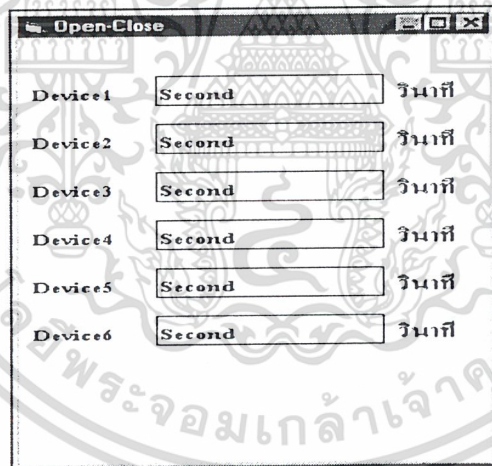
```

If txtdevice.Text = "CAMERA" Then
frmhousecontroller.lblopencamera.Caption = " "
frmhousecontroller.lblclosecamera.Caption = " "
End If
If txtdevice.Text = "RADIO" Then
frmhousecontroller.lblopenradio.Caption = " "
frmhousecontroller.lblcloseradio.Caption = " "
End If
Unload frmconfirm
End Sub
Private Sub Form_Load()
Dim t2
Adodcl.Recordset.MoveLast
Adodcl.Recordset.AddNew
txtdate.Text = Date
txttimeon.Text = Time
txttimeoff.Text = t2
End Sub

```

3.3 OPEN-CLOSE FORM(frmOpenClose.frm)

- ทำหน้าที่ในการเปิด-ปิดอุปกรณ์และนับเวลาถอยหลังเพื่อสั่งปิดอุปกรณ์
- ทำงานร่วมกับ frmprocess.frm และ housecontroller.frm เพื่อแจ้งสถานะและการควบคุม Microcontroller



รูปที่ 3.18 แสดงการนับเวลาถอยหลังเพื่อปิดอุปกรณ์ OPEN-CLOSE FORM

Source Code

```

Private Sub Timer1_Timer()
txtneon.Text = Val(txtneon.Text) - 1
If txtneon.Text = 0 Then
txtttest.Text = "setting device1"
frmprocess.MSComm1.Output = "1"
frmhousecontroller.Shape1.FillColor = &HFF&
Timer1.Enabled = False
End If
End Sub

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Private Sub Timer2_Timer()
txtfan.Text = Val(txtfan.Text) - 1
If txtfan.Text = 0 Then
txtttest.Text = "setting device2"
Timer2.Enabled = False
frmprocess.MSComm1.Output = "2"
frmhousecontroller.Shape2.FillColor = &HFF&
End If
End Sub

```

```

Private Sub Timer3_Timer()
txtair.Text = Val(txtair.Text) - 1
If txtair.Text = 0 Then
txtttest.Text = "setting device3"
Timer3.Enabled = False
frmprocess.MSComm1.Output = "3"
frmhousecontroller.Shape3.FillColor = &HFF&
End If
End Sub

```

```

Private Sub Timer4_Timer()
txttv.Text = Val(txttv.Text) - 1
If txttv.Text = 0 Then
txtttest.Text = "setting device4"
Timer4.Enabled = False
frmprocess.MSComm1.Output = "4"
frmhousecontroller.Shape4.FillColor = &HFF&
End If
End Sub

```

```

Private Sub Timer5_Timer()
txtradio.Text = Val(txtradio.Text) - 1
If txtradio.Text = 0 Then
txtttest.Text = "setting device5"
Timer5.Enabled = False
frmprocess.MSComm1.Output = "5"
frmhousecontroller.Shape5.FillColor = &HFF&
End If
End Sub

```

```

Private Sub Timer6_Timer()
txtcamera.Text = Val(txtcamera.Text) - 1
If txtcamera.Text = 0 Then
txtttest.Text = "setting device6"
Timer6.Enabled = False
frmprocess.MSComm1.Output = "6"
frmhousecontroller.Shape6.FillColor = &HFF&
End If
End Sub

```

```

Private Sub txtadd_Change()
txtttest.Text = Val(txtttest.Text) + 10
End Sub

```

```

Private Sub txtttest_Change()
Dim hour1 As Integer
Dim minutel As Integer
Dim second1 As Integer

```

เอกสารนี้เป็นเอกสารที่ออกนอกระบบการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim hour2 As Integer
Dim minute2 As Integer
Dim second2 As Integer
```

```
If txtdevice.Text = "NEON" Then
A = Hour(txttimeoff.Text) - Hour(txttimeon.Text)
hour1 = A * 3600
B = Minute(txttimeoff.Text) - Minute(txttimeon.Text)
minutel = B * 60
C = Second(txttimeoff.Text) - Second(txttimeon.Text)
second1 = C
txtneon.Text = (hour1 + minutel + second1)
Timer1.Enabled = True
End If
```

```
-----
If txtdevice.Text = "FAN" Then
A = Hour(txttimeoff.Text) - Hour(txttimeon.Text)
hour1 = A * 3600
B = Minute(txttimeoff.Text) - Minute(txttimeon.Text)
minutel = B * 60
C = Second(txttimeoff.Text) - Second(txttimeon.Text)
second1 = C
txtfan.Text = (hour1 + minutel + second1)
Timer2.Enabled = True
End If
```

```
-----
If txtdevice.Text = "AIR" Then
A = Hour(txttimeoff.Text) - Hour(txttimeon.Text)
hour1 = A * 3600
B = Minute(txttimeoff.Text) - Minute(txttimeon.Text)
minutel = B * 60
C = Second(txttimeoff.Text) - Second(txttimeon.Text)
second1 = C
txtair.Text = (hour1 + minutel + second1)
Timer3.Enabled = True
End If
```

```
-----
If txtdevice.Text = "TV" Then
A = Hour(txttimeoff.Text) - Hour(txttimeon.Text)
hour1 = A * 3600
B = Minute(txttimeoff.Text) - Minute(txttimeon.Text)
minutel = B * 60
C = Second(txttimeoff.Text) - Second(txttimeon.Text)
second1 = C
```

```
txttv.Text = (hour1 + minutel + second1)
Timer4.Enabled = True
End If
```

```
-----
If txtdevice.Text = "RADIO" Then
A = Hour(txttimeoff.Text) - Hour(txttimeon.Text)
hour1 = A * 3600
B = Minute(txttimeoff.Text) - Minute(txttimeon.Text)
minutel = B * 60
C = Second(txttimeoff.Text) - Second(txttimeon.Text)
second1 = C
txtradio.Text = (hour1 + minutel + second1)
Timer5.Enabled = True
End If
```

```
-----
If txtdevice.Text = "CAMERA" Then
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

A = Hour(txttimeoff.Text) - Hour(txttimeon.Text)
hour1 = A * 3600
B = Minute(txttimeoff.Text) - Minute(txttimeon.Text)
minutel = B * 60
C = Second(txttimeoff.Text) - Second(txttimeon.Text)
second1 = C
txtcamera.Text = (hour1 + minutel + second1)
Timer6.Enabled = True
End If
'-----

```

```
End Sub
```

```

Private Sub txtdevice_Change()
If txtdevice.Text = "NEON" Then
txttest.Text = "1"
End If
'-----

```

```

If txtdevice.Text = "FAN" Then
txttest.Text = "2"
End If
'-----

```

```

If txtdevice.Text = "AIR" Then
txttest.Text = "3"
End If
'-----

```

```

If txtdevice.Text = "TV" Then
txttest.Text = "4"
End If
'-----

```

```

If txtdevice.Text = "RADIO" Then
txttest.Text = "5"
End If
'-----

```

```

If txtdevice.Text = "CAMERA" Then
txttest.Text = "6"
End If
'-----

```

```
End Sub
```

3.4 PROCESS FORM(frmprocess.frm)

- ทำหน้าที่ในการสั่งงานให้ Microcontroller สั่งงานให้อุปกรณ์นั้นทำการเปิดหรือปิด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 แสดงสถานะของการทำงานของอุปกรณ์ PROCESS FORM

Source Code

```

Public atto1
Public atto2
Public atto3
Public atto4
Public atto5
Public atto6
Public atto7
Public atto8
Private Sub Command1_Click()
frmhousecontroller.Show vbModeless
End Sub
Private Sub Form_Load()
MSComm1.CommPort = 1
MSComm1.PortOpen = True
MSComm1.Settings = "9600,n,8,1"
MSComm1.RThreshold = 1
MSComm1.InputLen = 0
atto1 = False
atto2 = False
atto3 = False
atto4 = False
atto5 = False
atto6 = False
atto7 = False
atto8 = False
End Sub
Private Sub MSComm1_OnComm()
Select Case MSComm1.CommEvent
Case comEvReceive
buffer = MSComm1.Input
If buffer = "1" Then
atto1 = Not atto1
If atto1 = True Then
Shapel(1).FillColor = &HFFFFFF0
Else
Shapel(1).FillColor = &HFF8080
End If
End If
If buffer = "2" Then
atto2 = Not atto2
If atto2 = True Then
Shapel(2).FillColor = &HFFFFFF0
Else
Shapel(2).FillColor = &HFF8080
End If
End If
If buffer = "3" Then
atto3 = Not atto3
If atto3 = True Then
Shapel(3).FillColor = &HFFFFFF0
Else
Shapel(3).FillColor = &HFF8080
End If
End If
If buffer = "4" Then
atto4 = Not atto4
If atto4 = True Then
Shapel(4).FillColor = &HFFFFFF0

```

เอกสารนี้เป็นเอกสารที่ลิขสิทธิ์สงวนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Else
    Shapel(4).FillColor = &HFF8080
End If
End If

If buffer = "5" Then
    atto5 = Not atto5
    If atto5 = True Then
        Shapel(5).FillColor = &HFFFFFFC0
    Else
        Shapel(5).FillColor = &HFF8080
    End If
End If

If buffer = "6" Then
    atto6 = Not atto6
    If atto6 = True Then
        Shapel(6).FillColor = &HFFFFFFC0
    Else
        Shapel(6).FillColor = &HFF8080
    End If
End If

If buffer = "7" Then
    atto7 = Not atto7
    If atto7 = True Then
        Shapel(7).FillColor = &HFFFFFFC0
    Else
        Shapel(7).FillColor = &HFF8080
    End If
End If

If buffer = "8" Then
    atto8 = Not atto8
    If atto8 = True Then
        Shapel(8).FillColor = &HFFFFFFC0
    Else
        Shapel(8).FillColor = &HFF8080
    End If
End If

'Text1.Text = attol
End Select
End Sub

```

3.5 WATCHING FORM(frmwatch.frm)

- ทำหน้าตรวจสอบสถานะการเปลี่ยนของข้อมูล maindatabase
- เหมือนเกิดการเปลี่ยนฐานข้อมูล frmprocess จะทำงานร่วมกับ frmOpenCloseเพื่อทำการเปิด-ปิด อุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.20 แสดงการตรวจจับสถานะการเปลี่ยนแปลงข้อมูล WATCHING FORM

Source Code

```

Private Sub Form_Load()
    Adodcl.Recordset.MoveLast
    txtdevice2.Text = txtdevice1.Text
    txttimeon2.Text = txttimeon1.Text
    txttimeoff2.Text = txttimeoff1.Text
    txtdate2.Text = txtdate1.Text
End Sub

Private Sub Timer1_Timer()
    Adodcl.Refresh
    Adodcl.Recordset.MoveLast
    txtdevice1.Text = Adodcl.Recordset.Fields("DEVICE")
    txttimeon1.Text = Adodcl.Recordset.Fields("TIME ON")
    txttimeoff1.Text = Adodcl.Recordset.Fields("TIME OFF")
    txtdate1.Text = Adodcl.Recordset.Fields("DATE")
    If txtdevice1.Text <> txtdevice2.Text Then
        If txttimeon1.Text <> txttimeon2.Text Then

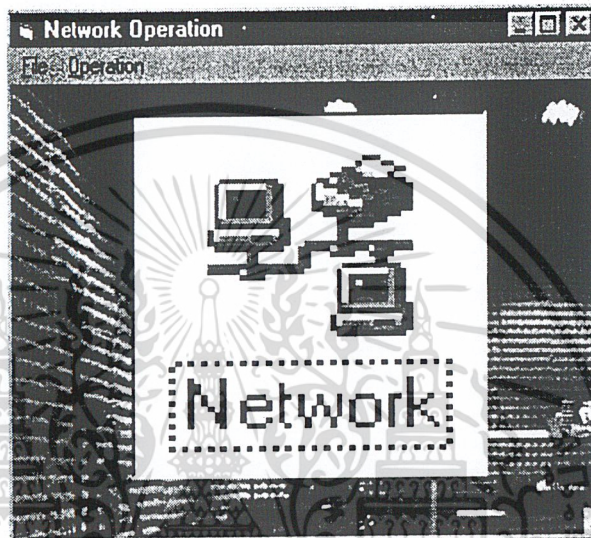
            txtttest.Text = Val(txtttest.Text) + 1
            txtdevice2.Text = txtdevice1.Text
            txttimeon2.Text = txttimeon1.Text
            txtdate2.Text = txtdate1.Text
            txttimeoff2.Text = txttimeoff1.Text
            '-----
            frmOpenClose.txtdevice.Text = txtdevice2.Text
            frmOpenClose.txttimeon.Text = txttimeon2.Text
            frmOpenClose.txttimeoff.Text = txttimeoff2.Text
        End If
    End If
End Sub

```

เอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. NETWORK FORM(frmnetwork.frm)

- ทำหน้าที่สื่อสารระหว่างเครื่องคอมพิวเตอร์ Server-Client
- ประกอบด้วย 2 FORM
 1. CHAT FORM
 2. SEND FILE FORM



รูปที่ 3.21 แสดงโปรแกรม NETWORK

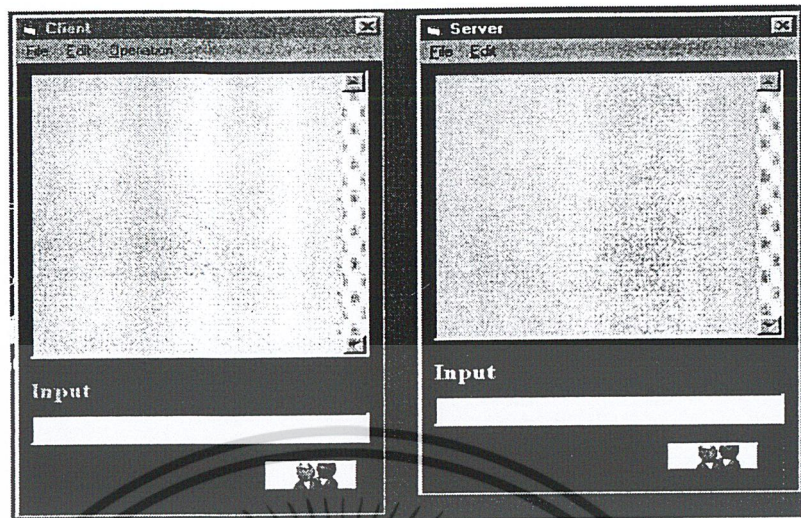
Source Code

```
Private Sub Chat_Click()
Unload Me
frmServer.Show
End Sub
Private Sub Select_Click()
Unload Me
frmselectprogram.Show vbModal
End Sub
```

4.1 CHAT FORM(frmchat.frm)

- ทำหน้าที่ในการสื่อสาร พูดคุย ระหว่างเครื่อง Server-Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 แสดงโปรแกรม CHAT

Source Code

1. Server Code

```
Private Sub cmdSend_Click()
On Error GoTo errlabel:
TCPSTServer.SendData txtInput.Text
txtOutput.Text = txtOutput.Text & vbCrLf & "Server:" & txtInput.Text
txtInput.Text = ""
Exit Sub
errlabel:
ans = MsgBox("เลือก chat จาก frmclient ใช้ operate", vbYesNo + vbQuestion,
"Login Error")
End Sub

Private Sub exit_Click()
Unload FrmClient
Unload frmServer
frmselectprogram.Show
End Sub

Private Sub Form_Load()
TCPSTServer.LocalPort = 1200
TCPSTServer.Listen
FrmClient.Show vbModeless
End Sub

Private Sub TCPSTServer_ConnectionRequest(ByVal requestID As Long)
If TCPSTServer.State <> sckClosed Then
TCPSTServer.Close
TCPSTServer.Accept requestID
End If
End Sub
```

```
Private Sub TCPSTServer_DataArrival(ByVal bytesTotal As Long)
```

Dim strData As String

เอกสารนี้เป็นเอกสารประกอบการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TCPClient.GetData strData
If txtOutput.Text <> "" Then
txtOutput.Text = txtOutput.Text & vbCrLf & "Client:" & strData
Else
txtOutput.Text = "Client:" & strData
End If
End Sub

```

2.Client Code

```

Private Sub Chat_Click()
TCPClient.Connect
End Sub
Private Sub cmdSend_Click()
On Error GoTo errlabel:
TCPClient.SendData txtInput2.Text
txtOutput2.Text = txtOutput2.Text & vbCrLf & "Client:" & txtInput2.Text
txtInput2.Text = ""
Exit Sub
errlabel:
ans = MsgBox("เลือก chat ใช้ operation", vbYesNo + vbQuestion, "Login
Error")
End Sub
Private Sub exit_Click()
Unload FrmClient
Unload frmServer
frmselectprogram.Show
End Sub

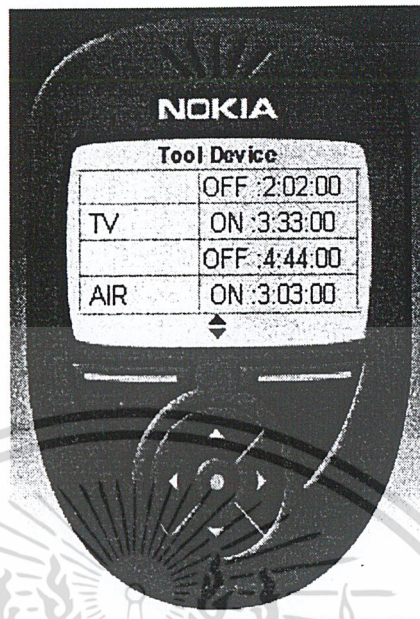
Private Sub Form_Load()
TCPClient.RemoteHost = "ATTO"
TCPClient.RemotePort = 1200
End Sub
Private Sub TCPClient_DataArrival(ByVal bytesTotal As Long)
Dim strData As String
TCPClient.GetData strData
If txtOutput2.Text <> "" Then
txtOutput2.Text = txtOutput2.Text & vbCrLf & "Server:" & strData
Else
txtOutput2.Text = "Server:" & strData
End If
End Sub

```

3.3.3 ขั้นตอนการสร้างและโปรแกรม ASP และ WAP

- โปรแกรม WAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 แสดงหน้าตาของโปรแกรม WAP

Source Code

1. WAP.ASP

```
<%
Response.ContentType="text/vnd.wap.wml"
Dim Conn, sql

'Making Connection
Set Conn = Server.CreateObject("ADODB.Connection")
Conn.Open "maindatabase", "", ""
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="main" title="Tool Device">
    <p align="center">
      <table columns="2">
        <% sql="SELECT * FROM table1 where device = 'NEON' ORDER
BY date DESC"
          set rs=Conn.Execute(sql) %>
        <tr> <td> NEON </td> <td> ON :<%=rs("time on")%> </td> </tr>
        <tr> <td></td> <td>OFF :<%=rs("time off")%></td> </tr>
        <% sql="SELECT * FROM table1 where device = 'TV' ORDER BY
date DESC"
          set rs=Conn.Execute(sql) %>
        <tr> <td> TV </td> <td> ON :<%=rs("time on")%> </td> </tr>
        <td></td> <td>OFF :<%=rs("time off")%></td> </tr>
        <% sql="SELECT * FROM table1 where device = 'AIR' ORDER
BY date DESC"
          set rs=Conn.Execute(sql) %>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    <tr> <td > AIR </td> <td> ON :<%=rs("time on")%> </td> </tr>    <tr>
<td></td> <td>OFF :<%=rs("time off")%></td> </tr>
    <% sql="SELECT * FROM table1 where device = 'RADIO' ORDER
BY date DESC"
    set rs=Conn.Execute(sql) %>
    <tr> <td > RADIO </td> <td> ON :<%=rs("time on")%> </td> </tr>
<tr> <td></td> <td>OFF :<%=rs("time off")%></td> </tr>
    <% sql="SELECT * FROM table1 where device = 'FAN' ORDER
BY date DESC"
    set rs=Conn.Execute(sql) %>
    <tr> <td > FAN </td> <td> ON :<%=rs("time on")%> </td> </tr>    <tr>
<td></td> <td>OFF :<%=rs("time off")%></td> </tr>
    <% sql="SELECT * FROM table1 where device = 'CAMERA'
ORDER BY date DESC"
    set rs=Conn.Execute(sql) %>
    <tr> <td > CAMERA </td> <td> ON :<%=rs("time on")%> </td> </tr>
<tr> <td></td> <td>OFF :<%=rs("time off")%></td> </tr>

<% Rs.Close %>
</table>

    <anchor title="Set Time">Set Time Device
    <go href="settime.asp" >
    </go>
    </anchor>
</p>
</card>
</wml>

```

2. SETTIME.ASP

```

<%
Response.ContentType="text/vnd.wap.wml"
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
    <card id="main" title="Set Time">
    <p align="center">
<b>TOOL</b><br/>
<select name="device">
<option value="-">Select Device</option>
    <option value='NEON'>NEON</option>
    <option value='TV'>TV</option>
    <option value='AIR'>AIR</option>
    <option value='RADIO'>RADIO</option>
    <option value='FAN'>FAN</option>
    <option value='CAMERA'>CAMERA</option>
</select>
</p>
<p>ON time<br/>
<input name="ontime" size="5" type="text"/>
</p>
<p>OFF Time</p>
<p>
<input name="offtime" size="5" type="text"/>
</p>
<p align="center"/><p>
<anchor title="Set Timer">Submit Set Timer
<go href="savetime.asp" method="post">

```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<postfield name="device" value="$device"/>
<postfield name="ontime" value="$ontime"/>
<postfield name="offtime" value="$offtime"/>
</go>
</anchor>
<br/><br/>
<anchor title="View Timer">View Timer
<go href="wap.asp">
</go>
</anchor>
    </p>
    </card>
</wml>

```

3. SETTIME2.ASP

```

<%
Response.ContentType="text/vnd.wap.wml"
%>
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
  <wml>
    <card id="main" title="Set Time">
      <p align="center">
<b>TOOL</b><br/>
<select name="device">
<option value="-">Select Device</option>
  <option value='NEON'>NEON</option>
  <option value='TV'>TV</option>
  <option value='AIR'>AIR</option>
<option value='RADIO'>RADIO</option>
  <option value='FAN'>FAN</option>
  <option value='CAMERA'>CAMERA</option>
</select>
</p>
<p>ON time<br/>
<input name="ontime" size="5" type="text"/>
</p>
<p>OFF Time</p>
<p>
<input name="offtime" size="5" type="text"/>
</p>
<p align="center"/><p>
<anchor title="Set Timer">Submit Set Timer
<go href="savetime.asp" method="post">
<postfield name="device" value="$device"/>
<postfield name="ontime" value="$ontime"/>
<postfield name="offtime" value="$offtime"/>
</go>
</anchor>
<br/><br/>
<anchor title="View Timer">View Timer
<go href="wap.asp">
</go>
</anchor>
    </p>
    </card>
  </wml>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

แบ่งการทดลองออกเป็น 2 ส่วน คือ

1. การทดลองการควบคุมจากคอมพิวเตอร์ภายในบ้าน
2. การทดลองการควบคุมโดยผ่านอินเทอร์เน็ต

****หมายเหตุ**** ผลการทดลองการควบคุมโดยอินเทอร์เน็ตอาศัย Software ที่ชื่อว่า “PWS”(Personal Web Server)

4.1 ขั้นตอนการปฏิบัติ

4.1.1 ขั้นตอนการปฏิบัติในการใช้งานอุปกรณ์

1. ติดตั้งเครื่องควบคุมอุปกรณ์ภายในบ้าน
2. ต่อสายคอมพิวเตอร์(DB9)กับ Serial Port หลังคอมพิวเตอร์ และต่อสาย ลงในกล่องควบคุมอุปกรณ์
3. เปิดสวิตช์อุปกรณ์
4. ทำการรีเซ็ตวงจร(สวิตช์รีเซ็ตของ Microcontroller) ให้ผู้สถานะพร้อมในการทำงาน
5. เปิดโปรแกรม Project 1.exe
6. ใช้งาน โปรแกรมตามต้องการ

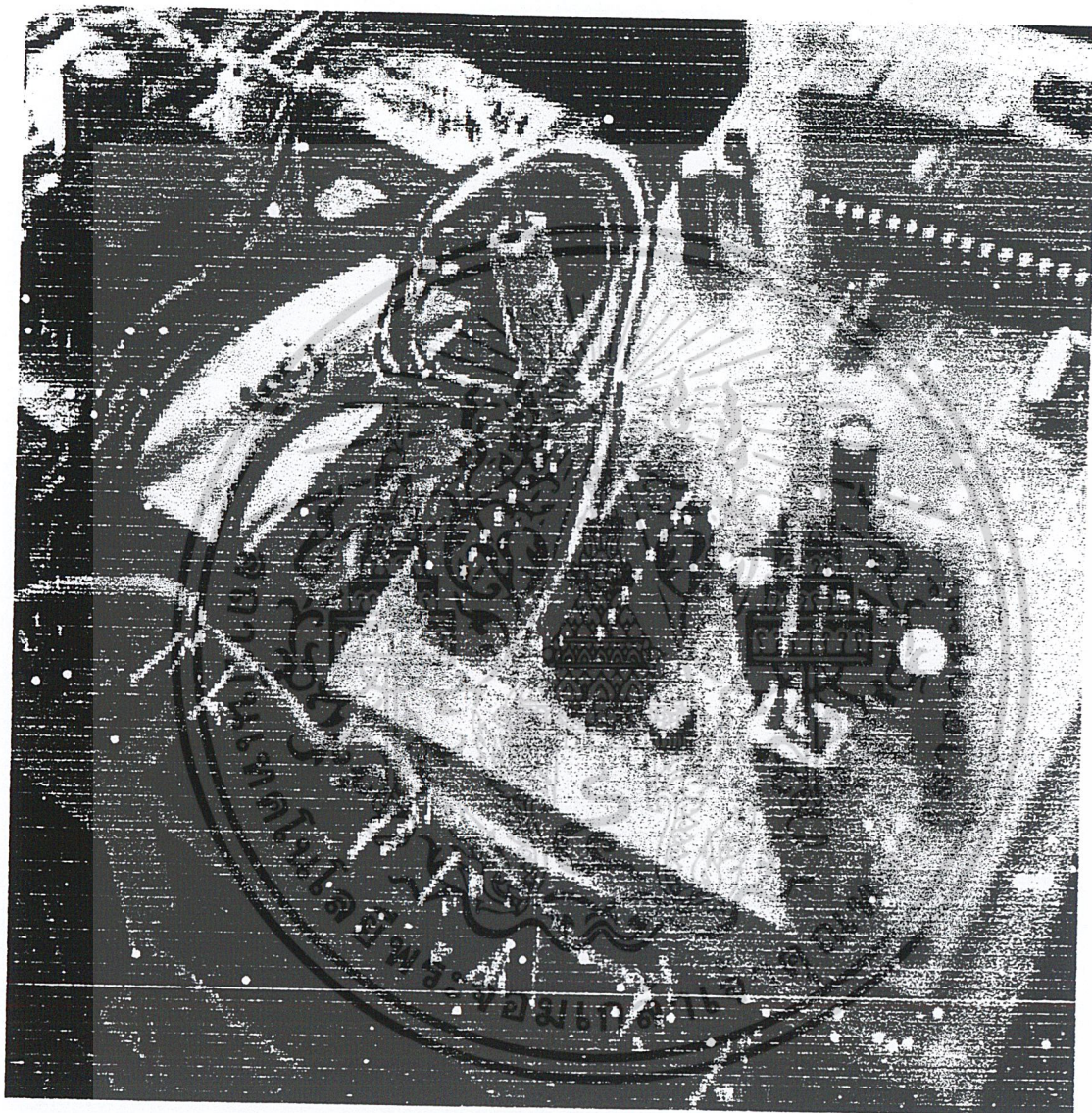
****หมายเหตุ****

ข้อควรระวัง

1. ควรเปิดคอมพิวเตอร์ให้เรียบร้อยก่อนทำการ ต่อสาย Serial Port
2. ควรทำการรีเซ็ตอุปกรณ์ทุกครั้งเมื่อจะทำการ ใช้งาน
3. หลังเสร็จการ ใช้งานควรถอนสายต่างๆ และปิดสวิตช์ให้เรียบร้อย
4. ควรถอดสาย Serial Port ออกจากเครื่องควบคุมอุปกรณ์ ก่อนทำการปิดเครื่องคอมพิวเตอร์

4.1.2 การติดตั้งอุปกรณ์

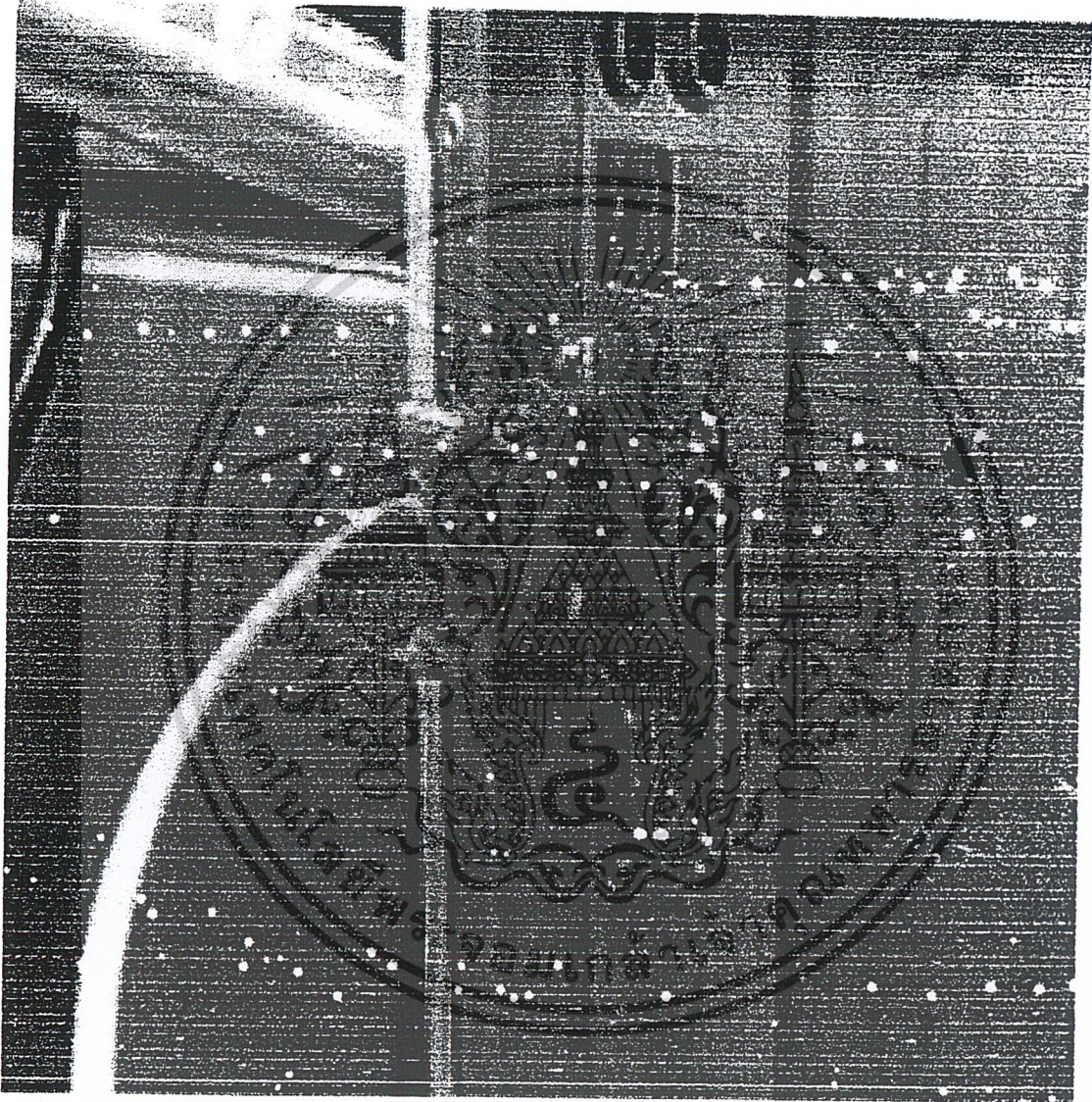
1. ต่อสาย Connector ให้กับเครื่องควบคุม



รูปที่ 4.1 แสดงการเชื่อมต่อ CONNECTOR สำหรับติดต่อกับเครื่องคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

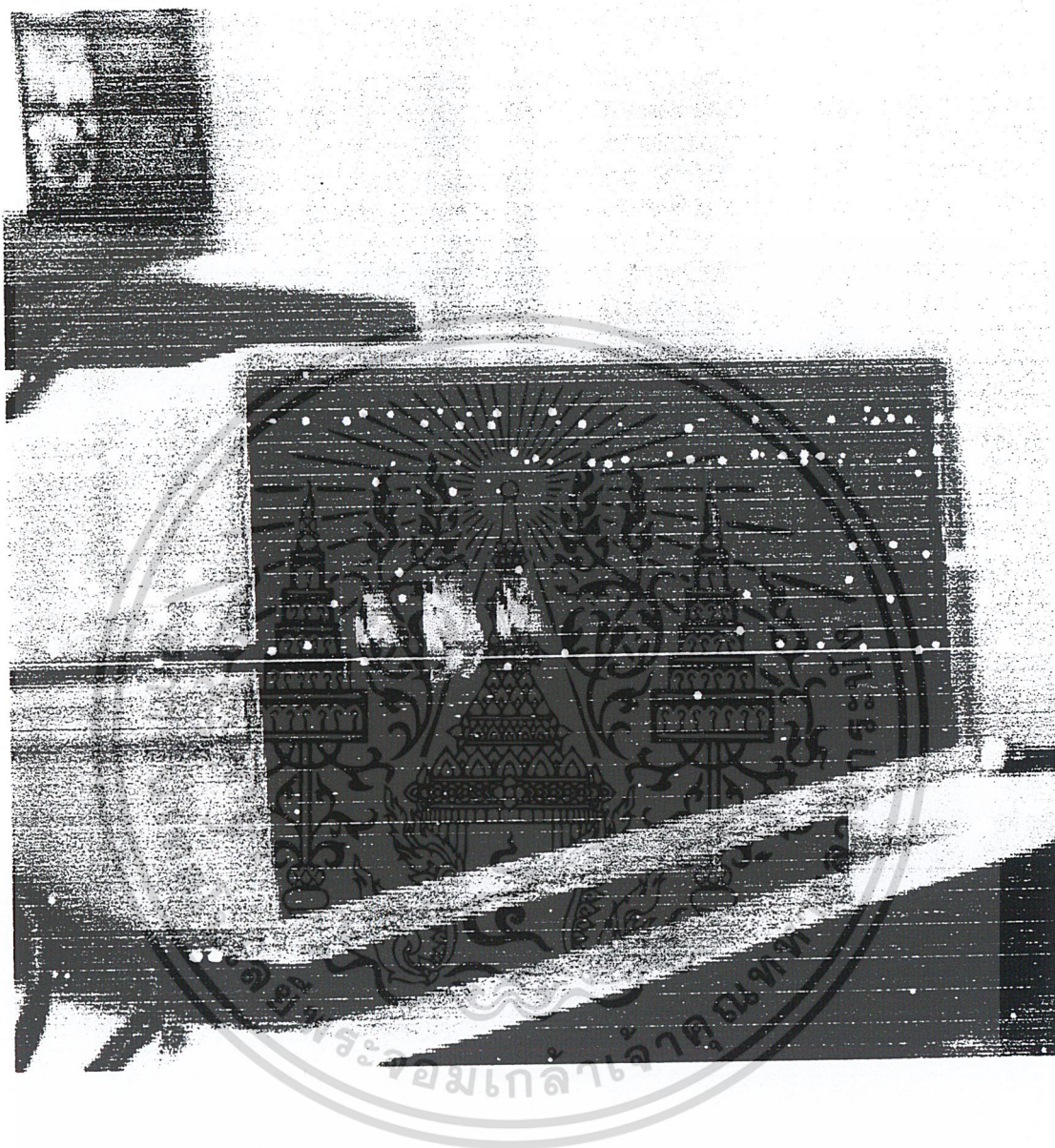
2. ต่อสาย คอมพอร์ท (DB9) เข้ากับ Serial Port ของเครื่องคอมพิวเตอร์



รูปที่ 4.2 แสดงการต่อสายDB9 เข้ากับ Serial Port ของคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เปิดสวิตช์ให้อุปกรณ์ทำงาน

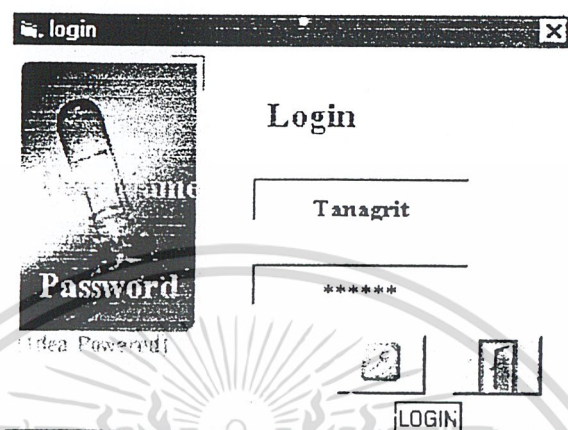


รูปที่ 4.3 แสดงการทำงานของเครื่องควบคุมอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

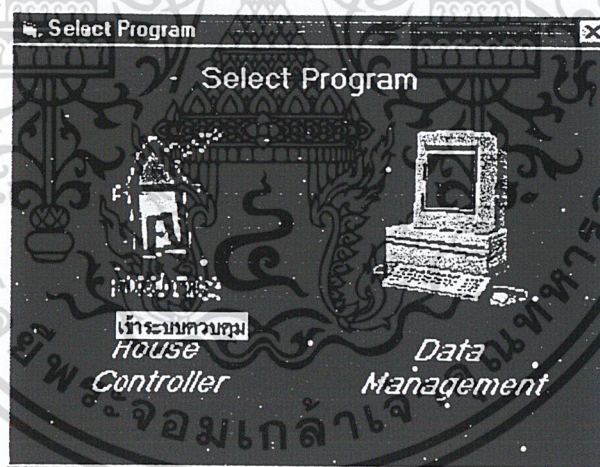
4.2 ผลการทดลองการควบคุมจากคอมพิวเตอร์ภายในบ้าน

- ทดลองทำการควบคุมการเปิด-ปิด หลอดไฟฟ้า
 1. เปิดโปรแกรม Project 1
 2. LOGIN เข้าสู่ระบบการใช้งาน



รูปที่ 4.4 แสดงการทดลองการ LOGIN

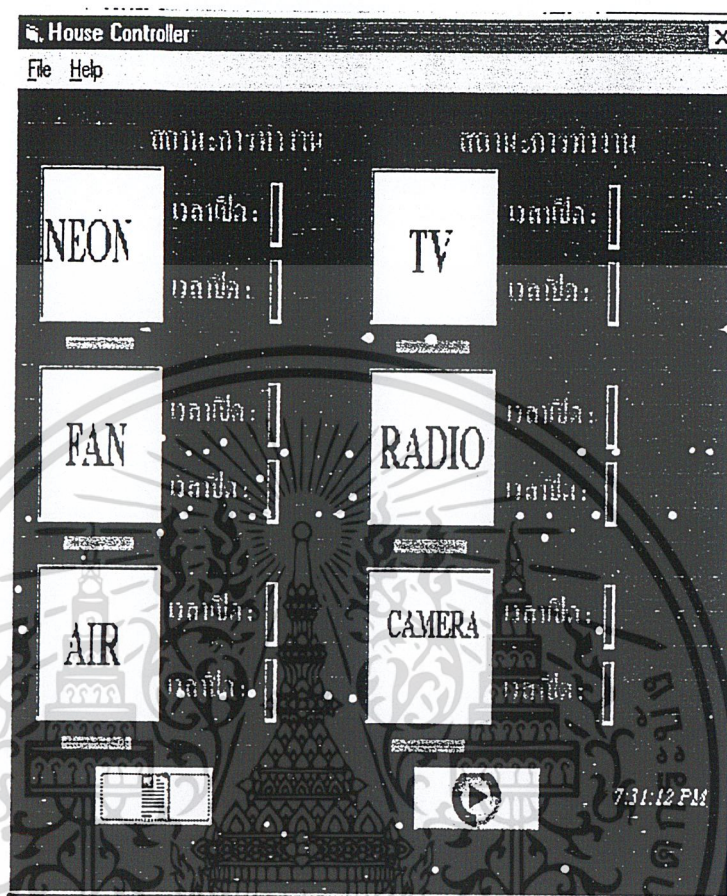
3. เลือก HOUSE CONTROLLER



รูปที่ 4.5 แสดงการทดลองการ เลือกโหมดการทำงาน HOUSE CONTROLLER

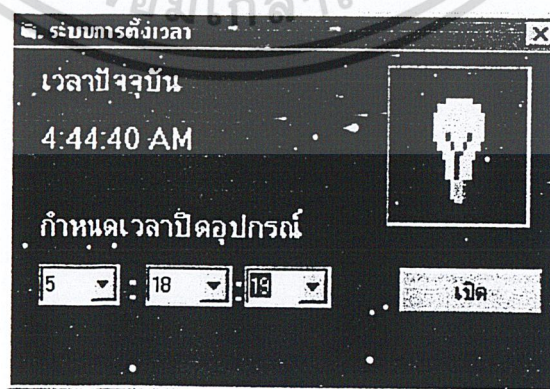
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เลือก icon เพื่อทำการกำหนดเวลาให้อุปกรณ์



รูปที่ 4.6 แสดงการทดลองการทำงาน HOUSE CONTROLLER

5. กำหนดเวลาให้อุปกรณ์



รูปที่ 4.7 แสดงการทดลองการ กำหนดเวลาให้หลอดไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ยืนยันสถานะการทำงานด้วย firmconfirm

ยืนยันหรือทำการเปลี่ยนแปลง

DEVICE: NEON

DATE: 3/24/02

TIME ON: 4:49:40 AM

TIME OFF: 5:18:05 AM

รูปที่ 4.8 แสดงการทดลองการ ยืนยันการกำหนดเวลาให้หลอดไฟฟ้า

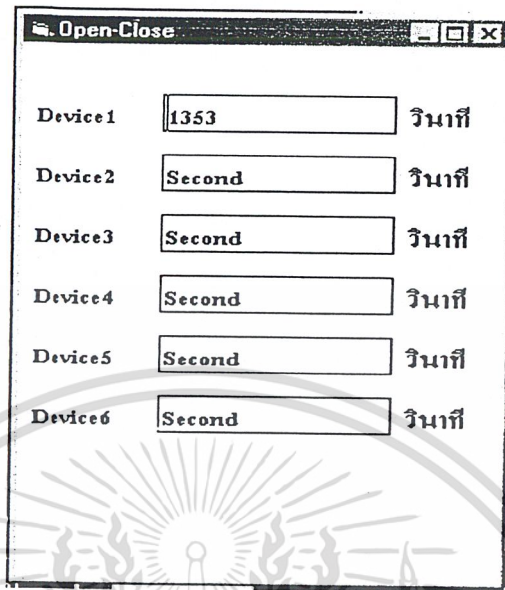
7. ข้อมูลจะถูกบันทึกในฐานข้อมูล

DATE	DEVICE	TIME ON	TIME OFF
21/3/02	CAMERA	4:51:14	6:00:06
21/3/02	TV	7:24:20	8:00:08
24/3/02	NEON	4:49:40	5:18:05
24/3/02	NEON	4:55:19	5:18:05

รูปที่ 4.9 แสดงการเปลี่ยนแปลงของข้อมูลในฐานข้อมูล

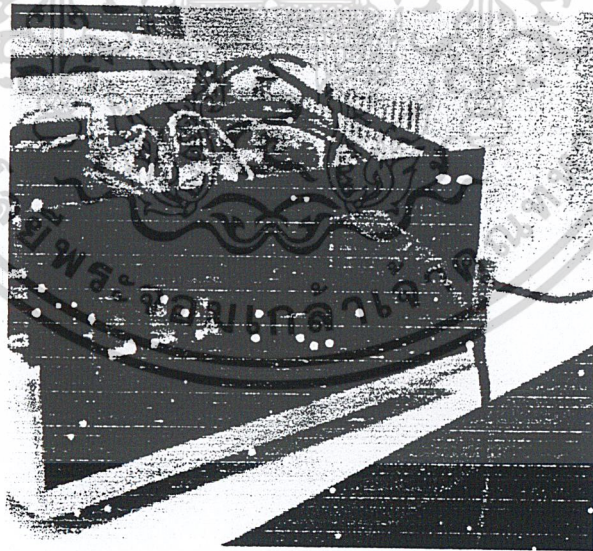
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. เวลาจะถูกส่ง ไปจัดการยัง frmOpenClose เพื่อทำการเริ่มนับเวลา



รูปที่ 4.10 แสดงการทดลองการนับเวลาถอยหลังให้หลอดไฟฟ้า

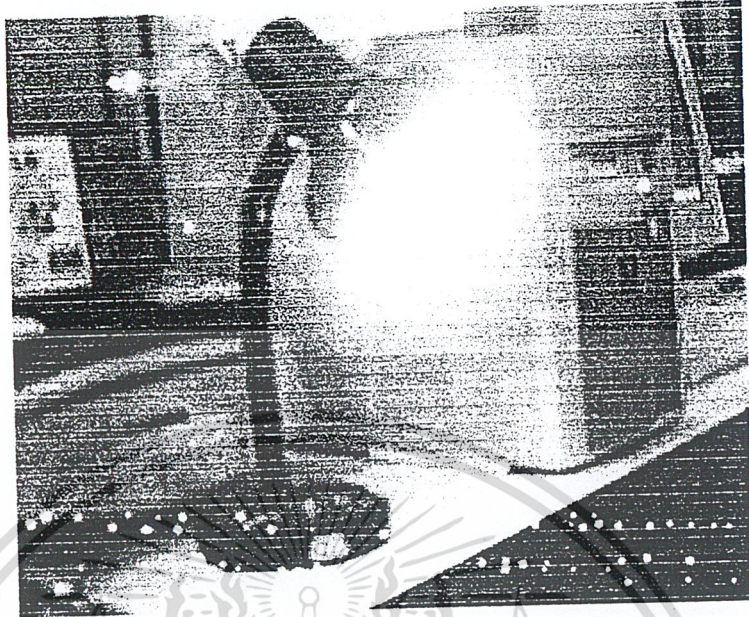
9. frmprocess ได้รับข้อมูลแล้วจะจัดการส่งค่าให้ Microcontroller



รูปที่ 4.11 แสดงการทดลองเมื่อ Microcontroller สั่งงานให้ RELAY ทำงาน

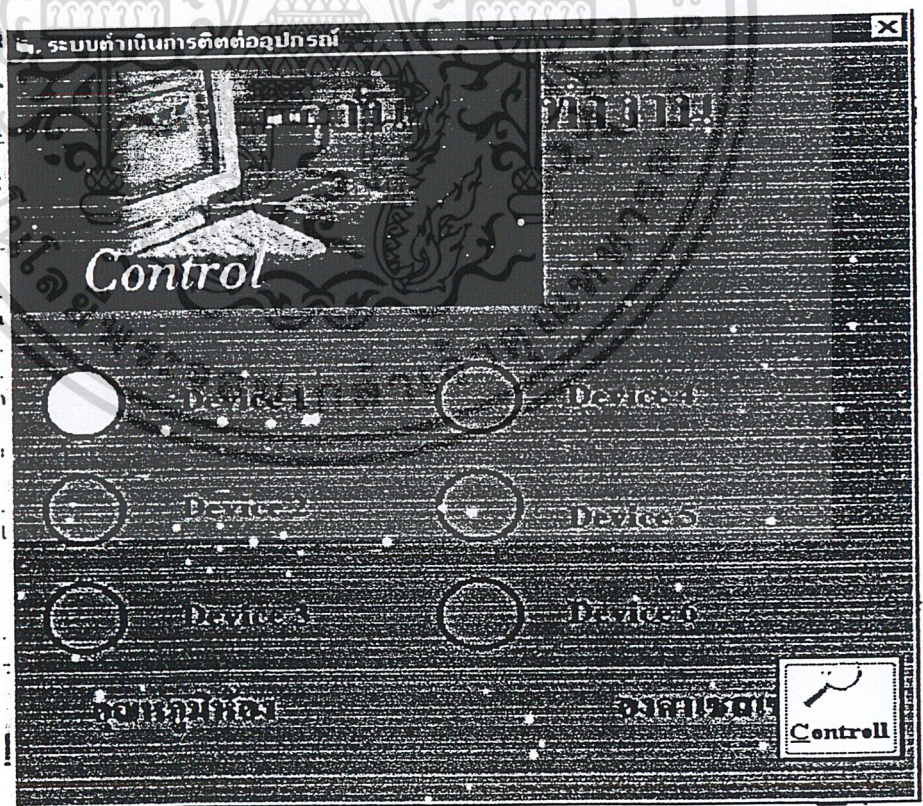
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. Microcontroller สั่งให้ RELAY ทำงาน มีผลให้อุปกรณ์ไฟฟ้าติด



รูปที่ 4.12 แสดงการทำงานของหลอดไฟฟ้า

11. Microcontroller ส่ง feedback กลับมายัง fmrprocess แจ้งว่าอุปกรณ์(คิด)ทำงานแล้ว



รูปที่ 4.13 แสดงสถานะบน PROCESS FORM ของหลอดไฟฟ้า

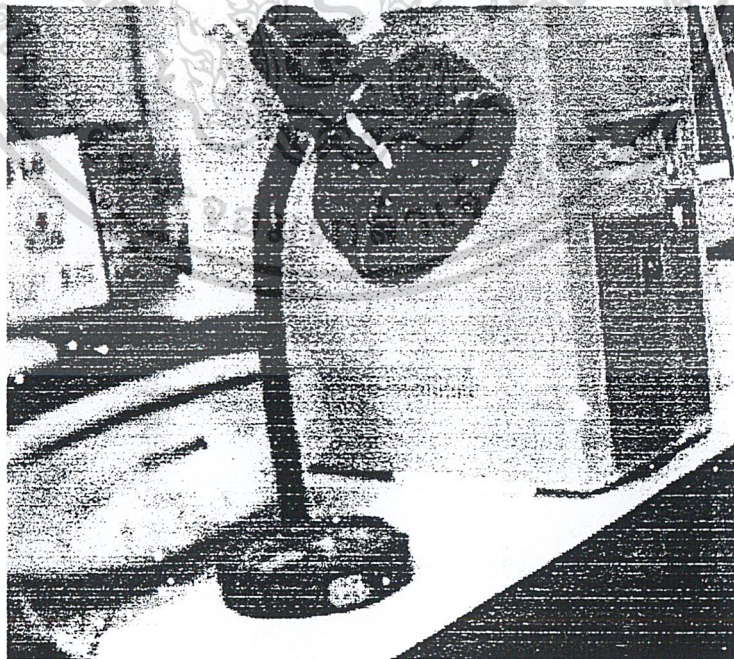
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. FrmOpenClose ทำการนับเวลาตลอดเวลาเมื่อ เวลานั้นๆเป็นศูนย์ จะส่งให้frmprocces ทำการส่งค่า ไปยัง Microcontroller อีกครั้ง

Device	Value	Unit
Device 1	0	วินาที
Device 2	Second	วินาที
Device 3	Second	วินาที
Device 4	Second	วินาที
Device 5	Second	วินาที
Device 6	Second	วินาที

รูปที่ 4.14 แสดงเวลาการนับถอยหลังของ OPEN-CLOSE FORM

13. Microcontroller สั่งให้ RELAY ทำงาน มีผลให้อุปกรณ์ไฟฟ้าดับ



รูปที่ 4.15 แสดงการทำงานของหลอดไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. Microcontroller ส่ง feedback กลับมายัง firmprocess แจ้งว่าอุปกรณ์(คืบ)ทำงานแล้ว



รูปที่ 4.16 แสดงสถานะบน PROCESS FORM ของหลอดไฟฟ้า

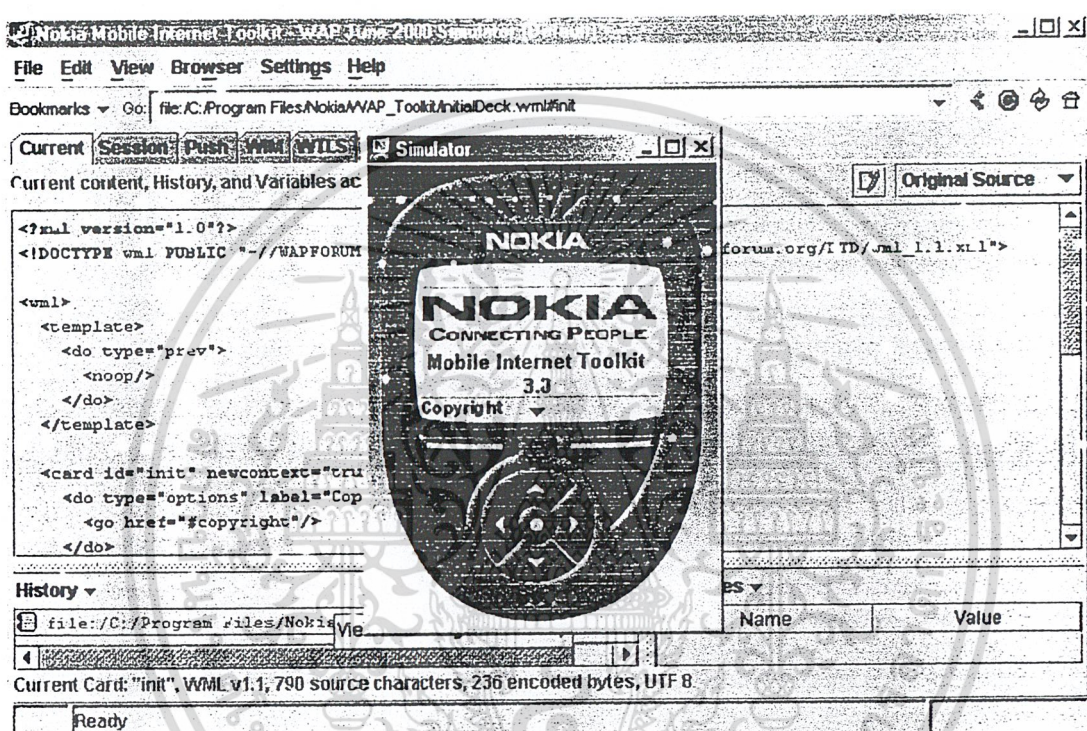
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดลองการควบคุมโดยผ่านอินเทอร์เน็ต

1. เปิดโปรแกรม WAP ตามขั้นตอนต่อไปนี้

การทดลองการอัปเดตข้อมูลผ่านทาง WAP

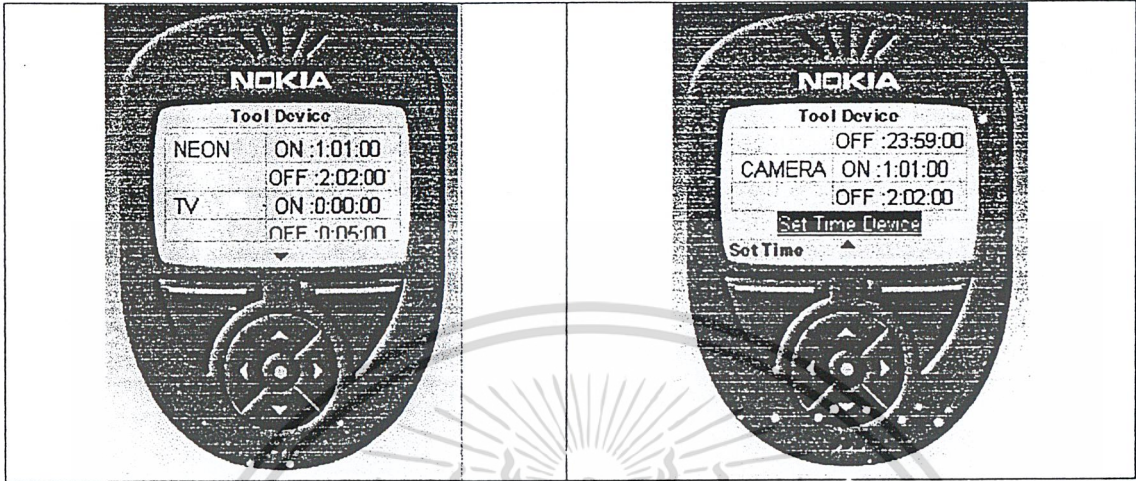
ในการทดลองนี้เราจะใช้โปรแกรม Nokia Mobile Internet Toolkit 3.0 เป็นโปรแกรมทดสอบการแก้ไขข้อมูลลง database server ผ่านทาง wab



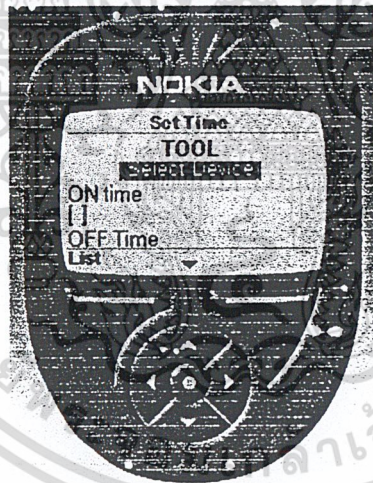
รูปที่ 4.17 หน้าตาของโปรแกรม Nokia Mobile Internet Toolkit 3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มการทดลอง จะใช้โปรแกรมทดสอบเรียกไปยัง host wap ในที่นี้คือ <http://net/pro/wap/wap.asp>
จะปรากฏหน้าของเวบดังรูป

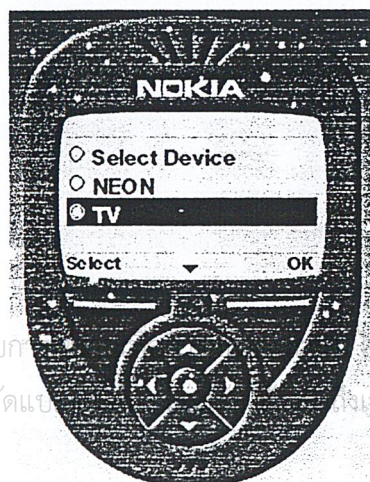


ซึ่งหน้าจะดึงข้อมูลที่บันทึกไว้มาโชว์
ต่อไปจะเป็นการตั้งเวลาเปิดปิดอุปกรณ์ เลือกเมนู set time device แล้วค้ปุ่ม settime จะขึ้นมาดังนี้



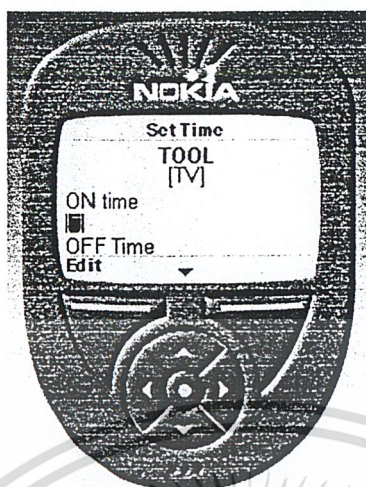
ต่อไปทำการเลือกอุปกรณ์ที่จะตั้งเวลา เลือกเมนู Select Device แล้วค้ปุ่ม List จะมีรายการขึ้นมาให้

เลือก

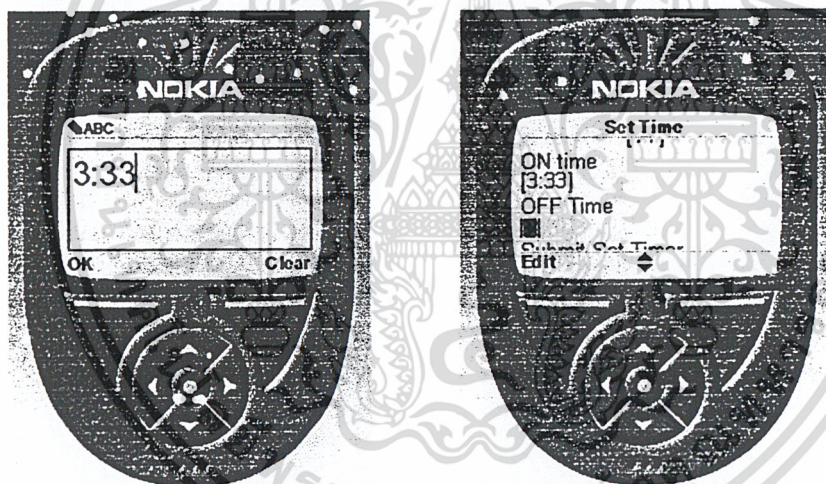


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับค... อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดเ... เจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะทดลอง โดยการตั้งเวลาอุปกรณ์คือ TV เลือกมาเมนู TV กดปุ่ม select แล้ว กดปุ่ม OK



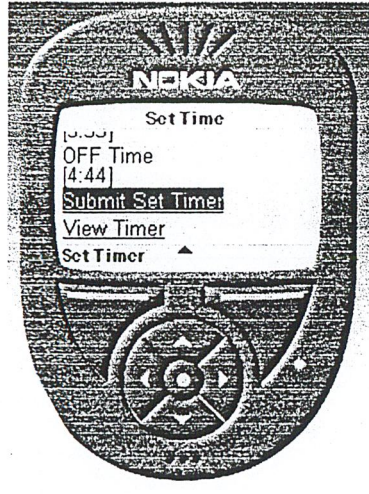
เมื่อเลือกแล้วก็จะปรากฏคำว่า TV ในช่อง device ขึ้นมาจากนั้นเลื่อนเมนู ไปตั้งค่า ON time แล้วกด Edit จะมีกล่องให้ป้อนข้อมูล ป้อนเวลาเปิดไป 3:33 แล้วกด OK



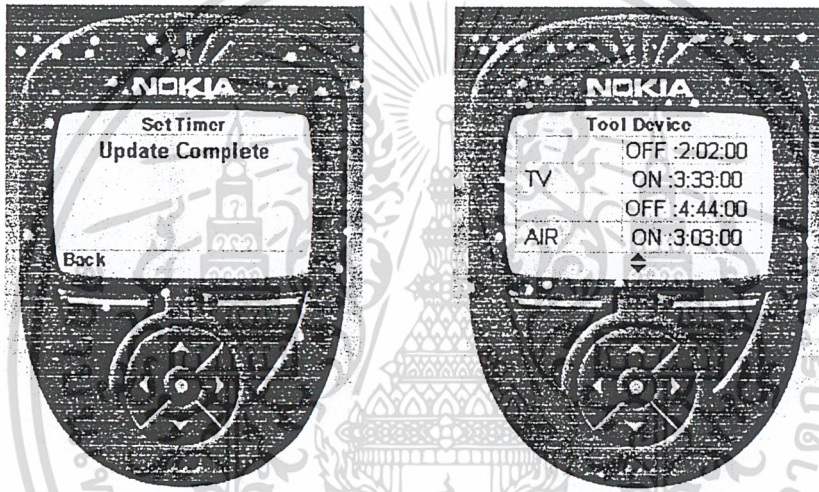
จากนั้นเราจะตั้งเวลาปิด โดยก็จะคล้ายเคมกับการตั้งเวลาเปิด เลื่อนเมนู ไปยัง OFF time แล้วกด Edit ป้อนเวลาปิดไป เป็น 4:44 แล้วกด OK



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลง... ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เวลาที่ตั้งไปจะ ปรากฏขึ้นมา จากนั้นกด Submit Set Time เมื่อ update ได้จะปรากฏหน้าจอขึ้นมา แล้ว กด Back กลับไปดูยังหน้าจอลูกเวลา



ดูในรายการเปิดปิดอุปกรณ์ ค่าของ TV เวลาเปิดคือ 3:33 และเวลาปิดคือ 4:44 การตั้งค่าสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เกิดการเปลี่ยนแปลงข้อมูลของฐานข้อมูลหลังจากทำการใช้ WAP

	DEVICE	TIME ON	TIME OFF
21/3/02	NEON	4:50:42	5:00:05
21/3/02	FAN	4:50:50	5:00:05
21/3/02	AIR	4:50:56	5:00:05
21/3/02	RADIO	4:51:06	5:00:05
21/3/02	TV	4:51:10	5:00:05
21/3/02	CAMERA	4:51:14	6:00:06
21/3/02	TV	7:24:20	8:00:08
24/3/02	NEON	4:49:40	5:18:05
24/3/02	NEON	4:55:19	5:18:05
24/3/02	TV	3:33:00	4:44:00

รูปที่ 4.18 แสดงการเปลี่ยนแปลงของข้อมูลจากการบันทึกของ WAP

3. หลังบันทึกข้อมูลแล้ว fimwatch จะทำการตรวจจับข้อมูลที่เข้ามาใหม่ตลอดเวลา

Watching System

เรกอร์ดสุดท้ายของตารางข้อมูล

Device: TV

Date: 3/24/02

Time ON: 3:33:00 AM

Time OFF: 4:44:00 AM

เรกอร์ดล่าสุดที่เกิดการ Update

Device: TV

Date: 24 March 2002

Time ON: 03:33:00

Time OFF: 04:44:00

รูปที่ 4.19 แสดงสถานะบน WATCH FORM ที่ตรวจจับเจอ

4. fimwatch จะส่งเวลาให้ fimOpenClose ในการจัดการด้านเวลาและการตั้งงานเครื่องควบคุมอุปกรณ์ Microcontroller ทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จ. Open-Close		
Device 1	Second	วินาที
Device 2	Second	วินาที
Device 3	Second	วินาที
Device 4	4260	วินาที
Device 5	Second	วินาที
Device 6	Second	วินาที

รูปที่ 4.20 แสดงการนับเวลาถอยหลังของ OPEN-CLOSE FORM

5. frmprocess ทำงานตามข้อมูลที่ได้รับมา

จ. ระบบดำเนินการผลิตคอมพิวเตอร์

Control

Device 1 Device 4

Device 2 Device 5

Device 3 Device 6

องคมนตรี องค์กรบริหาร Control

รูปที่ 4.21 แสดงสถานะบน PROCESS FORM ของทีวี

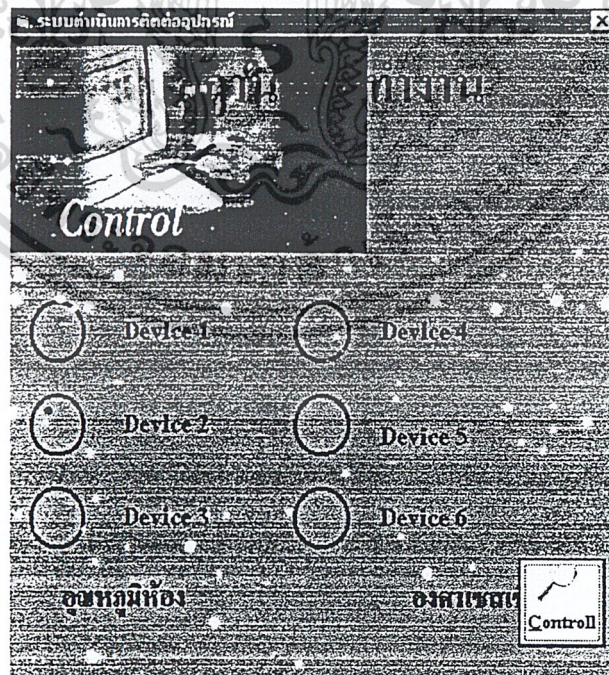
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. เมื่อเวลาที่ firmOpenClose ตรวจจับเท่ากับศูนย์ จะสั่งงานให้ Microcontroller ทำการปิด อุปกรณ์นั้น

Device	Value	Unit
Device1	Second	วินาที
Device2	Second	วินาที
Device3	Second	วินาที
Device4	0	วินาที
Device5	Second	วินาที
Device6	Second	วินาที

รูปที่ 4.22 แสดงการทำงาน ของ OPEN-CLOSE FORM

7. firmProcess รับ feedback จาก Microcontroller ก็จะแสดงสถานะบนหน้าจอ



รูปที่ 4.23 แสดงสถานะบน PROCESS FORM ของที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 กรแสดงการใช้งานอุปกรณ์ด้วย DATA REPORT

- เลือกผลการใช้งานย้อนหลังจาก icon ใน house controller



- จะได้ผลดังนี้

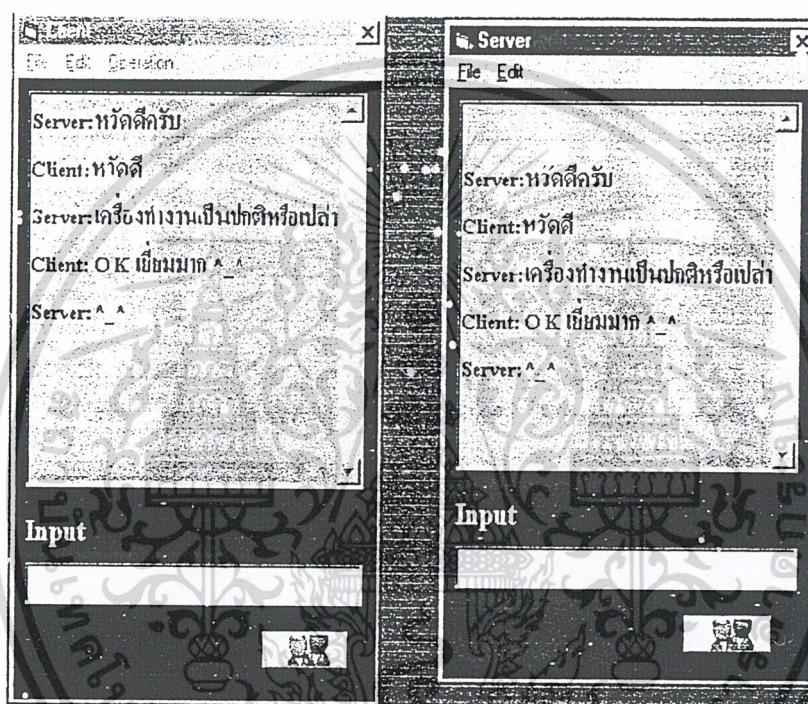
DATE	DEVICE	TIME ON	TIME OFF
24 March	FAN	12:00:00	13:00:00
24 March	NEON	12:00:00	13:13:00
24 March	TV	03:33:00	04:44:00

รูปที่ 4.24 แสดงรายงานการใช้งานอุปกรณ์ โดย DATA REPORT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การติดต่อสื่อสารระหว่าง Client - Server

- เป็นการติดต่อสื่อสาร หรือพูดคุยกันระหว่าง เครื่องคอมพิวเตอร์หรือที่เรียกทั่วไปว่าการ “Chat”
 - ทำได้โดยเลือก โหมด network control จาก selectprogram form
 - แล้วเลือก “Operation>Chat to Client-Server”
- จะได้ผลการทดลองดังนี้



รูปที่ 4.25 แสดงการพูดคุยกันระหว่างเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

สรุปและวิเคราะห์

1. สรุปผลการดำเนินงาน

- ในการทำงานระหว่าง Visual Basic และ Microcontroller ทำงานได้อย่างมีประสิทธิภาพ สามารถเปิด-ปิดอุปกรณ์ได้จริง
- การทำงานเปิด-ปิดอุปกรณ์โดยผ่าน Internet สามารถทำงานได้ดี ในการใช้โปรแกรมจำลอง Server อย่าง PWS

2. ปัญหาและแนวทางการแก้ไข

- ขณะที่ Microcontroller เบอร์ AT89C51 ทำงาน ต้องการระดับแรงดันที่คงที่ การเกิดปัญหาแรงดันตกจะทำให้ AT89C51 พังได้
การแก้ไขปัญหาคือ
 1. ควรคำนวณโหลดของการใช้งานจริงของอุปกรณ์ทั้งหมด
 2. ไม่ควรเชื่อมต่อ Connector กับวงจร ขณะทำการเปิดเครื่องคอมพิวเตอร์เพราะมีผลต่อแรงดัน
- การทำงานของวงจรควบคุมอุปกรณ์เก็บ Relay ซึ่งต้องใช้ไฟ 220 Volt ก่อนข้างอันตราย ดังนั้นควรต่อ Fuse ที่เหมาะสมให้กับวงจร
การแก้ไขปัญหาคือ
 1. ในโครงงานนี้แนะนำการใช้อุปกรณ์ครบทั้ง 6 อุปกรณ์ให้ใช้ Fuse ขนาด 5 Ampere และระดับกำลังไฟฟ้าของอุปกรณ์ทั้งหมดไม่เกิน 1,000 Watt (ตามมาตรฐาน มอก.)
 2. ในโครงงานนี้ใช้หม้อแปลง 2 ตัว คือแบบ 12 Volt 500 mA 2 ตัว เนื่องจากใช้เป็นแหล่งจ่ายให้แก่วงจร Microcontroller ที่ติดต่อกับ Serial Port และอีกตัวใช้จ่ายให้แก่วงจร Relay แบบ 12 Volt 6 ตัว ทั้งนี้เป็นการป้องกันการเกิดปัญหา
 1. การช็อตกันของวงจรภายใน โดยเฉพาะวงจร Relay ซึ่งใช้กับไฟบ้าน (ขนาด 220 Volt)
 2. เพื่อจะสามารถทดสอบและหาจุดผิดพลาดของวงจรง่ายขึ้น
- ตัว IC 7812 และ 7805 จะเกิดความร้อนขณะใช้งาน ดังนั้นควรติด Sink เพื่อลดความร้อนให้แก่ IC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แนวทางการพัฒนา

โครงการนี้ทำการศึกษาและทดลองการควบคุมอุปกรณ์ไฟฟ้าภายในบ้าน ดังนั้นสามารถประยุกต์ใช้งานกับองค์กร หรือบริษัทขนาดเล็กได้ และเนื่องจากโปรแกรม Visual Basic เป็นโปรแกรมที่สามารถประยุกต์ใช้งานได้หลากหลาย เช่น สามารถทำการถ่ายโอน File ระหว่างเครื่อง ทำการพูดคุยกันระหว่างเครื่อง (chat) สามารถทำการใช้ Remote PC Control คือการควบคุม PC ระยะไกลได้อีกด้วย การประยุกต์ใช้งานภายในบ้านเราสามารถออกแบบวงจรอื่น ๆ เช่น วงจรตรวจอุณหภูมิเพื่อป้องกันและแจ้งเตือนไฟไหม้ เครื่อง Webcam ซึ่งสามารถดูความเคลื่อนไหวภายในบ้านผ่านอินเทอร์เน็ต ซึ่งการประยุกต์ใช้งานกับวงจรต่าง ๆ สามารถทำได้โดยสะดวก เพราะ Visual Basic มี Control ต่าง ๆ ใช้รองรับการทำงานอยู่แล้ว อีกทั้งทางด้านการพัฒนาทั้ง Application จำพวก ASP, WAP ก็ทำได้เพราะ Visual Basic รองรับการทำงานแบบ Client-Server และรองรับโปรโตคอลแบบ TCP/IP และ UDP ซึ่งต่อไปในอนาคตการใช้งานและควบคุมอุปกรณ์อิเล็กทรอนิกส์ระยะไกล ผ่านโทรศัพท์เคลื่อนที่ที่มีแนวโน้มความนิยมสูงขึ้น

โครงการนี้จึงเป็นส่วนหนึ่งของการเปลี่ยนแปลง และสามารถพัฒนาให้เดินก้าวไปกับเทคโนโลยีด้านโทรคมนาคมในอนาคต





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

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

8-Bit Microcontroller with 4 Kbytes Flash

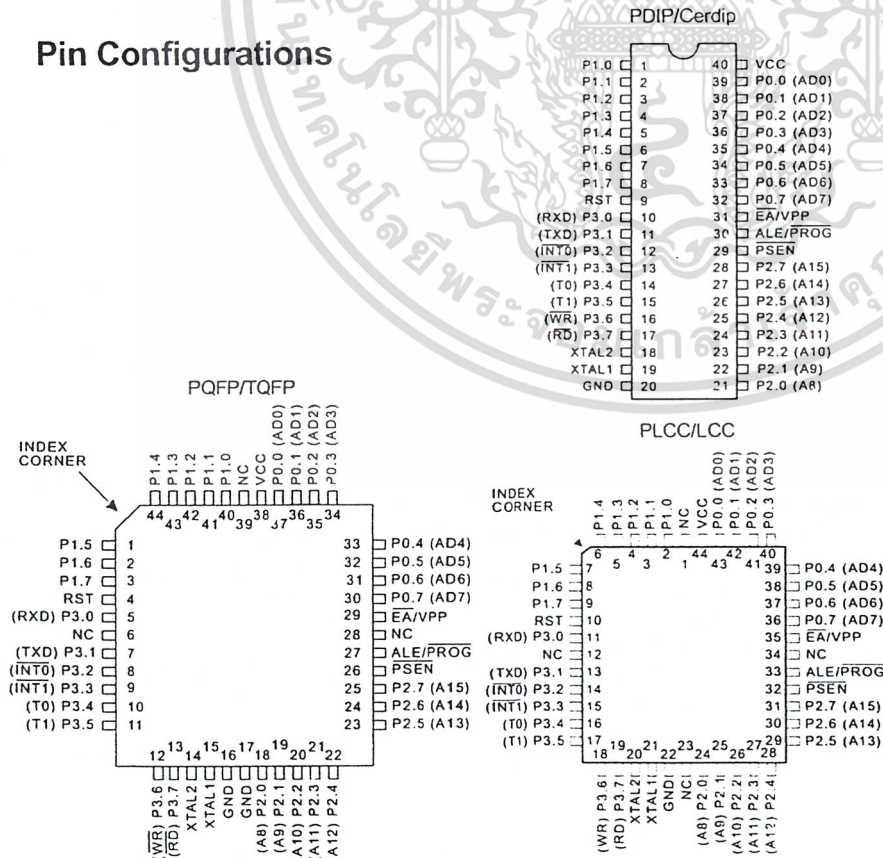
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash, Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

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

(continued)

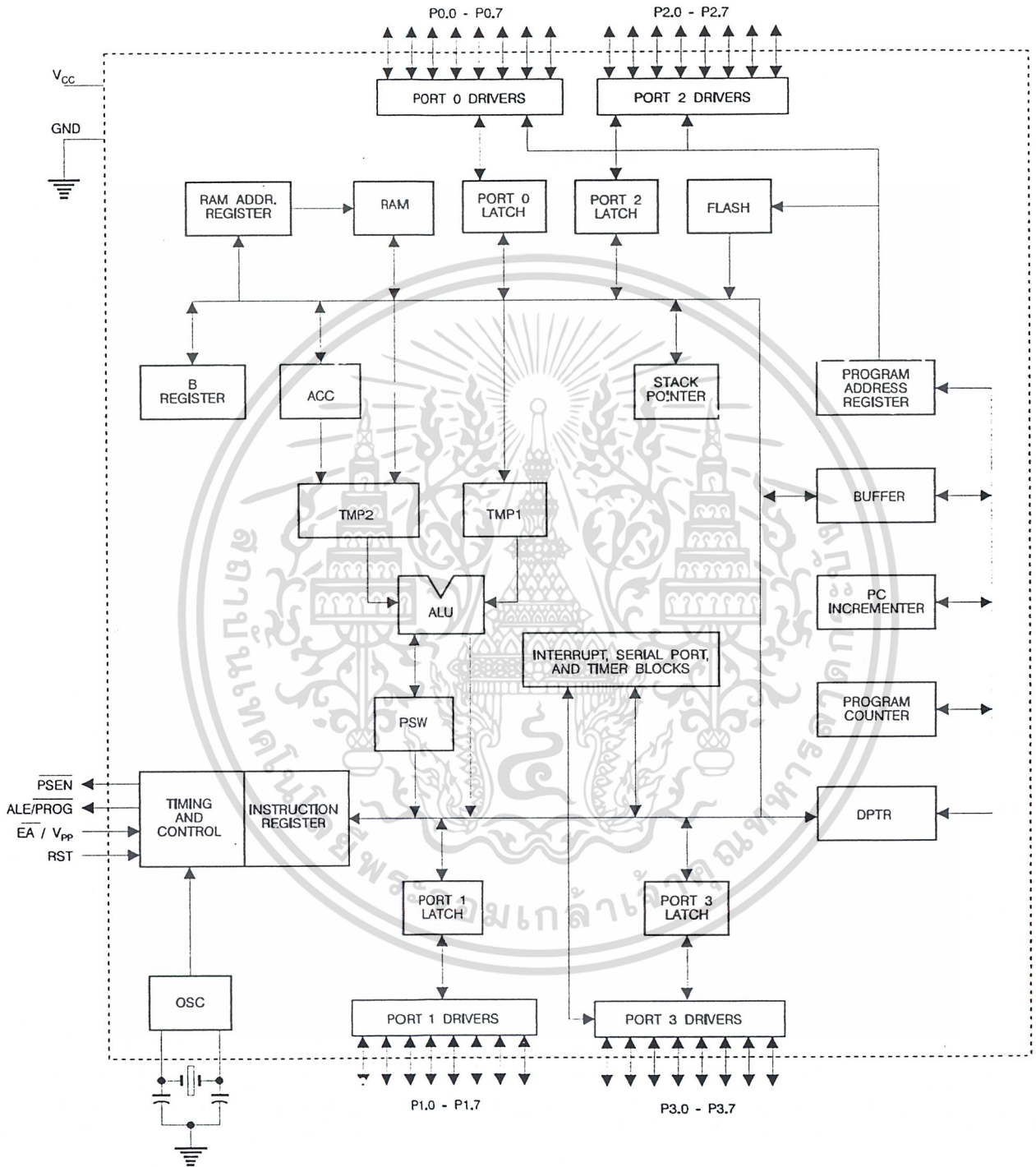
Pin Configurations



0265E



Block Diagram





HIN231, HIN232, HIN236, HIN237, HIN238, HIN239, HIN240, HIN241

Data Sheet

August 2001

File Number 3138.10

+5V Powered RS-232 Transmitters/Receivers

The HIN231-HIN241 family of RS-232 transmitters/receivers interface circuits meet all EIA RS-232E and V.28 specifications, and are particularly suited for those applications where $\pm 12V$ is not available. They require a single +5V power supply (except HIN231 and HIN239) and feature onboard charge pump voltage converters which generate +10V and -10V supplies from the 5V supply. The family of devices offer a wide variety of RS-232 transmitter/receiver combinations to accommodate various applications (see Selection Table).

The drivers feature true TTL/CMOS input compatibility, slew-rate-limited output, and 300 Ω power-off source impedance. The receivers can handle up to $\pm 30V$, and have a 3k Ω to 7k Ω input impedance. The receivers also feature hysteresis to greatly improve noise rejection.

Features

- Meets All RS-232E and V.28 Specifications
- Requires Only Single +5V Power Supply
 - (+5V and +12V - HIN231 and HIN239)
- High Data Rate..... 120kbps
- HIN233 and HIN235 Require No External Capacitors
- Onboard Voltage Doubler/Inverter
- Low Power Consumption
- Low Power Shutdown Function
- Three-State TTL/CMOS Receiver Outputs
- Multiple Drivers
 - $\pm 10V$ Output Swing for 5V Input
 - 300 Ω Power-Off Source Impedance
 - Output Current Limiting
 - TTL/CMOS Compatible
 - 30V/ μs Maximum Slew Rate
- Multiple Receivers
 - $\pm 30V$ Input Voltage Range
 - 3k Ω to 7k Ω Input Impedance
 - 0.5V Hysteresis to Improve Noise Rejection

Applications

- Any System Requiring RS-232 Communication Ports
 - Computer - Portable, Mainframe, Laptop
 - Peripheral - Printers and Terminals
 - Instrumentation
 - Modems

Selection Table

PART NUMBER	POWER SUPPLY VOLTAGE	NUMBER OF RS-232 DRIVERS	NUMBER OF RS-232 RECEIVERS	EXTERNAL COMPONENTS	LOW POWER SHUTDOWN/TTL THREE-STATE	NUMBER OF LEADS
HIN231	+5V and +7.5V to 13.2V	2	2	2 Capacitors	NO/NO	16
HIN232	+5V	2	2	4 Capacitors	NO/NO	16
HIN236	+5V	4	3	4 Capacitors	YES/YES	24
HIN237	+5V	5	3	4 Capacitors	NO/NO	24
HIN238	+5V	4	4	4 Capacitors	NO/NO	24
HIN239	+5V and +7.5V to 13.2V	3	5	2 Capacitors	NO/YES	24
HIN240	+5V	5	5	4 Capacitors	YES/YES	44
HIN241	+5V	4	5	4 Capacitors	YES/YES	28

HIN231 thru HIN241

Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
HIN231IB	-40 to 85	16 Ld SOIC	M16.3
HIN232CP	0 to 70	16 Ld PDIP	E16.3
HIN232CB	0 to 70	16 Ld SOIC	M16.3
HIN232IP	-40 to 85	16 Ld PDIP	E16.3
HIN232IB	-40 to 85	16 Ld SOIC	M16.3
HIN236CP	0 to 70	24 Ld PDIP	E24.3
HIN236CB	0 to 70	24 Ld SOIC	M24.3
HIN236IB	-40 to 85	24 Ld SOIC	M24.3
HIN237CB	0 to 70	24 Ld SOIC	M24.3
HIN238CP	0 to 70	24 Ld PDIP	E24.3

Ordering Information (Continued)

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
HIN238CB	0 to 70	24 Ld SOIC	M24.3
HIN238IB	-40 to 85	24 Ld SOIC	M24.3
HIN239CB	0 to 70	24 Ld SOIC	M24.3
HIN239CP	0 to 70	24 Ld PDIP	E24.3
HIN239IB	-40 to 85	24 Ld SOIC	M24.3
HIN240CN	0 to 70	44 Ld MQFP	Q44.10X10
HIN241CB	0 to 70	28 Ld SOIC	M28.3
HIN241IB	-40 to 85	28 Ld SOIC	M28.3
HIN241CA	0 to 70	28 Ld SSOP	M28.209

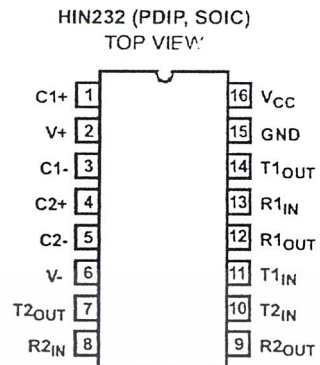
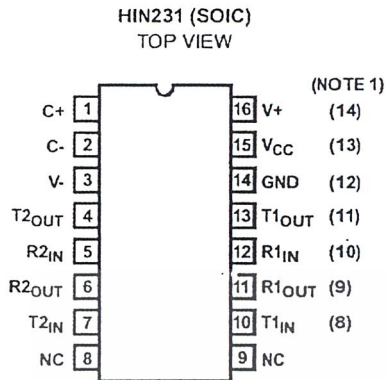
NOTE: Many of the surface mount devices are available on tape and reel; add -T to suffix.

Pin Descriptions

PIN	FUNCTION
V _{CC}	Power Supply Input 5V ±10%.
V+	Internally generated positive supply (+10V nominal), HIN231 and HIN239 require +7.5V to +13.2V.
V-	Internally generated negative supply (-10V nominal).
GND	Ground lead. Connect to 0V.
C1+	External capacitor (+ terminal) is connected to this lead.
C1-	External capacitor (- terminal) is connected to this lead.
C2+	External capacitor (+ terminal) is connected to this lead.
C2-	External capacitor (- terminal) is connected to this lead.
T _{IN}	Transmitter Inputs. These leads accept TTL/CMOS levels. An internal 400kΩ pull-up resistor to V _{CC} is connected to each lead.
T _{OUT}	Transmitter Outputs. These are RS-232 levels (nominally ±10V).
R _{IN}	Receiver Inputs. These inputs accept RS-232 input levels. An internal 5kΩ pull-down resistor to GND is connected to each input.
R _{OUT}	Receiver Outputs. These are TTL/CMOS levels.
\overline{EN}	Enable input. This is an active low input which enables the receiver outputs. With $\overline{EN} = 5V$, the outputs are placed in a high impedance state.
SHUTDOWN	Shutdown Input. With SHUTDOWN = 5V, the charge pump is disabled, the receiver outputs are in a high impedance state and the transmitters are shut off.
NC	No Connect. No connections are made to these leads.

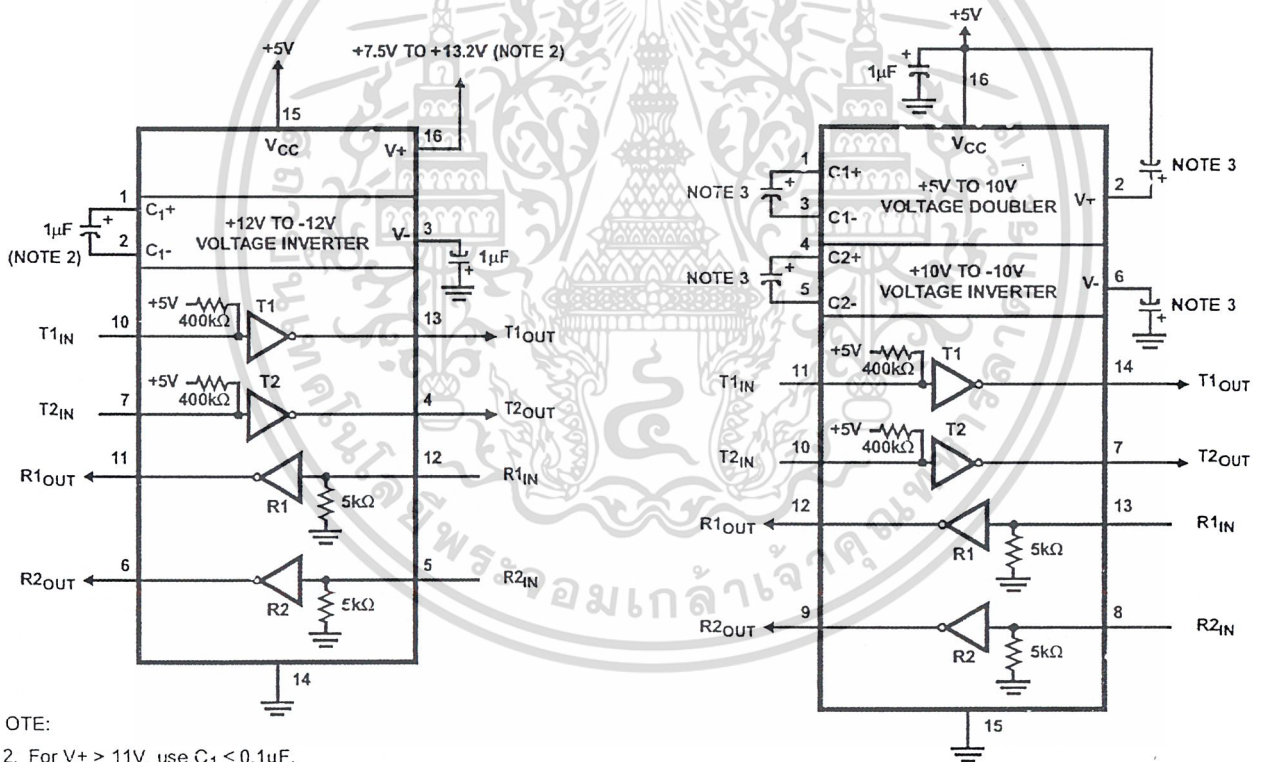
HIN231 thru HIN241

Pinouts



NOTE:

1. Pin numbers in parentheses are for PDIP Package.



OTE:

2. For $V+ > 11V$, use $C_1 \leq 0.1\mu F$.

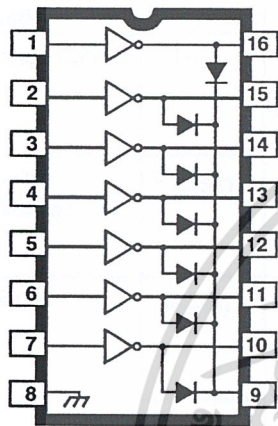
NOTE:

3. Either $0.1\mu F$ or $1\mu F$ capacitors may be used. The $V+$ capacitor may be terminated to V_{CC} or to GND.

2003 THRU 2024

Data Sheet
29304F

HIGH-VOLTAGE, HIGH-CURRENT DARLINGTON ARRAYS



Dwg. No. A-9594

Note that the ULN20xxA series (dual in-line package) and ULN20xxL series (small-outline IC package) are electrically identical and share a common terminal number assignment.

ABSOLUTE MAXIMUM RATINGS

Output Voltage, V_{CE}	
(ULN200xA and ULN200xL)	50 V
(ULN202xA and ULN202xL)	95 V
Input Voltage, V_{IN}	30 V
Continuous Output Current,	
I_C	500 mA
Continuous Input Current, I_{IN}	25 mA
Power Dissipation, P_D	
(one Darlington pair)	1.0 W
(total package)	See Graph
Operating Temperature Range,	
T_A	-20°C to +85°C
Storage Temperature Range,	
T_S	-55°C to +150°C

Ideally suited for interfacing between low-level logic circuitry and multiple peripheral power loads, the Series ULN20xxA/L high-voltage, high-current Darlington arrays feature continuous load current ratings to 500 mA for each of the seven drivers. At an appropriate duty cycle depending on ambient temperature and number of drivers turned ON simultaneously, typical power loads totaling over 230 W (350 mA x 7, 95 V) can be controlled. Typical loads include relays, solenoids, stepping motors, magnetic print hammers, multiplexed LED and incandescent displays, and heaters. All devices feature open-collector outputs with integral clamp diodes.

The ULN2003A/L and ULN2023A/L have series input resistors selected for operation directly with 5 V TTL or CMOS. These devices will handle numerous interface needs — particularly those beyond the capabilities of standard logic buffers.

The ULN2004A/L and ULN2024A/L have series input resistors for operation directly from 6 to 15 V CMOS or PMOS logic outputs.

The ULN2003A/L and ULN2004A/L are the standard Darlington arrays. The outputs are capable of sinking 500 mA and will withstand at least 50 V in the OFF state. Outputs may be paralleled for higher load current capability. The ULN2023A/L and ULN2024A/L will withstand 95 V in the OFF state.

These Darlington arrays are furnished in 16-pin dual in-line plastic packages (suffix "A") and 16-lead surface-mountable SOICs (suffix "L"). All devices are pinned with outputs opposite inputs to facilitate ease of circuit board layout. All devices are rated for operation over the temperature range of -20°C to +85°C. Most (see matrix, next page) are also available for operation to -40°C; to order, change the prefix from "ULN" to "ULQ".

FEATURES

- TTL, DTL, PMOS, or CMOS-Compatible Inputs
- Output Current to 500 mA
- Output Voltage to 95 V
- Transient-Protected Outputs
- Dual In-Line Plastic Package or Small-Outline IC Package

x = digit to identify specific device. Characteristic shown applies to family of devices with remaining digits as shown. See matrix on next page.

หนังสืออ้างอิง

1. ชัยวัฒน์ ลิ้มพรจิตรวิไล, วรพจน์ กรแก้ววัฒนกุล, “เรียนรู้และปฏิบัติการ ไมโครคอนโทรลเลอร์ MCS-51”, กรุงเทพฯ, 2538
2. สมยศ จุณณะปิยะ, “ไมโครคอนโทรลเลอร์ MCS-51”, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543
3. จิรศักดิ์ เหลืองอุไร, “การสื่อสารอนุกรมบน PC”, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), กรุงเทพฯ, 2538
4. สุรศักดิ์ สวงพงษ์, “สถาปัตยกรรมและโปรโตคอลที่ซีพี/ไอพี”, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), กรุงเทพฯ, 2543
5. พิพัฒน์ หิรัณย์วนิชชากร, “ระบบการสื่อสารข้อมูล และเครือข่ายคอมพิวเตอร์”, บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน), กรุงเทพฯ, 2542
6. กิตติ ภักดีวัฒนกุล, จำลอง ครูอุตสาหะ, “Visual Basic 6 ฉบับโปรแกรมเมอร์”, KTP COMP&CONSULT ISBN:974-7042-99-1, 2543
7. ร้อยโทอนุโชต วุฒิพรพงษ์, “สร้าง WAP ด้วย WMLSCRIPT”, Infopress, 2543
8. สัจจะ จรัสรุ่งวีวรร, สมพร จิวรสกุล, “ASP และ E-Commerce ฉบับสมบูรณ์”, Infopress, 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้