สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือ

HOME ELECTRONIC EQUIPMENT CONTROLLING BY MOBILE TELEPHONE

T 046552

โดย

นายวรรณรัตน์ เสระพันธุ์

นายวิลาส ทีเจริญ

นายวิศิษฏ์ บุท์ทะกะพันธ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เลขหมู่
เลขทะเบียน 46552
วัน, เดือน, ปี 4 เม.ย. 2546
.b.....
.i.....
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือ

HOME ELECTRONIC EQUIPMENT CONTROLLING BY MOBILE TELEPHONE

โดย

นายวรรณรัตน์ เสระพันธุ์

นายวิลาส ทีเจริญ

นายวิศิษฏ์ ขุทักะพันธ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต

สาขาเทคโนโลยีโทรคมนาคม ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	การควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือ
TITLE	HOME ELECTRONIC EQUIPMENT CONTROLLING BY MOBILE TELEPHONE
โดย	นายวรรณรัตน์ เสระพันธุ์
	นายวิลาส ทีเจริญ
	นายวิศิษฏ์ ขุท์ทะกะพันธ์
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์ พนารัตน์ ระวิวรรณ
	อาจารย์ มนต์ชัย แช่มช้อย
ระดับการศึกษา	ปริญญาอุตสาหกรรมศาสตรบัณฑิต
	สาขาวิชาเทคโนโลยีโทรคมนาคม
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2544

ปริญญานิพนธ์ฉบับนี้ได้รับการอนุมัติให้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรอุตสาหกรรมศาสตรบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

.....

(อาจารย์ พนารัตน์ ระวิวรรณ)

อาจารย์ผู้ควบคุมวิทยานิพนธ์

.....

(อาจารย์ มนต์ชัย แช่มช้อย)

อาจารย์ผู้ควบคุมวิทยานิพนธ์

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	**การควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือ** **HOME ELECTRONIC EQUIPMENT CONTROLLING BY MOBILE TELEPHONE**
นักศึกษา	นายวรรณรัตน์ เสระพันธุ์ 43015837 นายวิลาส ทีเจริญ 43015839 นายวิศิษฏ์ ขุท์ทะกะพันธ์ 43015840
อาจารย์ผู้ควบคุมวิทยานิพนธ์	อาจารย์ พนารัตน์ ระวิวรรณ อาจารย์ มนต์ชัย แช่มช้อย
ระดับการศึกษา	ปริญญาอุตสาหกรรมศาสตรบัณฑิต สาขาวิชาเทคโนโลยีโทรคมนาคม
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2544

บทคัดย่อ

โครงงานนี้ นำเสนอการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือ ซึ่งประกอบด้วย 3 ส่วนสำคัญคือ ภาษา WML , ภาษา Visual Basic, Microcontroller ส่วนที่ 1 คือ ภาษา WML ในการเขียน WAP Site เพื่อให้ผู้ใช้สามารถใช้งานอินเตอร์เน็ตทางโทรศัพท์มือถือได้ ส่วนที่ 2 ภาษา Visual Basic ใช้สำหรับออกแบบเชื่อมโยงการทำงานระหว่างโทรศัพท์มือถือและไมโครคอนโทรลเลอร์ นอกจากนี้ยังสามารถแสดงผลการควบคุมอุปกรณ์ไฟฟ้าบนหน้าจอคอมพิวเตอร์ได้อีกด้วย ส่วนที่3 คือส่วนของไมโครคอนโทรลเลอร์ เบอร์ AT90S8535 ผลิตโดยบริษัท ATMEL ใช้สำหรับสั่งงานอุปกรณ์ไฟฟ้าและสามารถตอบสนองกลับมายังผู้ใช้ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROJECT TITLE	**HOME ELECTRONIC EQUIPMENT CONTROLING BY MOBILE TELEPHONE**	
STUDENT	**Mr. Wannarat Sarapun**	**No.43015837**
	Mr. Wilas Teechalearn	**No.43015839**
	Mr. Wisit Kuttakaphan	**No.43015840**
ADVISOR	**Miss. Panarat Rawiwan**	
	Mr. Monchai Chamchoy	
COURSE	**Bachelor of Industrial Technology Telecommunication**	
DEPARTMENT	**Industrial Technology**	
YEAR	**2001**	

ABSTRACT

In this project presents home electronic equipment controlling by mobile phone. It consists of three important parts that are wireless markup language (WML) , Visual Basic language and Microcontroller. For the first part, WML is used to create the WAP Site for user's access to the Internet via mobile phone. The second part is the part of Visual basic language which function as interface between mobile phone and microcontroller .In addition, it is able to display the results of electronic equipment controlling through the computer for the find part, microcontroller which its number is AT90S8535 produced by ATMEL CO, LD is used to control electrical equipment and is also able to response to user.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้ได้จัดทำขึ้นเป็นผลสำเร็จ ทางคณะผู้จัดทำจักต้องขอขอบคุณบิดา มารดาที่คอยช่วยเหลือและให้กำลังใจ ขอขอบพระคุณบูรพาจารย์ทั้งหลายท่านผู้เขียนเอกสารและตำราอ้างอิงต่างๆทุกท่าน โดยเฉพาะอาจารย์ที่ปรึกษา อาจารย์ พนารัตน์ ระวิวรรณ, อาจารย์ มนต์ชัย แช่มช้อย, อาจารย์ พิชญ สุพรรณกุล และเพื่อนทุกคน ที่คอยให้คำปรึกษาแนะนำ และช่วยเหลือในการหาข้อมูลต่างๆ ทั้งทางด้านทฤษฎีและปฏิบัติเป็นอย่างดียิ่ง ทั้งนี้คณะผู้จัดทำต้องขอขอบพระคุณภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้คณะผู้จัดทำได้มีโอกาสเข้ามาศึกษา ณ สถาบันแห่งนี้

สุดท้ายนี้ทางคณะผู้จัดทำ ขอบพระคุณคณะอาจารย์ทุกท่านที่กรุณาประสิทธิ์ประสาทวิชาให้แก่กลุ่มผู้จัดทำ จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย

บทคัดย่อภาษาอังกฤษ

กิตติกรรมประกาศ

สารบัญ I

สารบัญรูป IV

สารบัญตาราง VII

บทที่ 1 บทนำ 1

1.1 บทนำ 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของงาน 2

1.4 โครงประกอบของโครงงาน 3

บทที่ 2 ทฤษฎีเกี่ยวกับ WAP, PHP และการใช้งาน Server 4

2.1 บทนำ 4

2.2 แบบโครงสร้างของ WAP 5

2.3 โพลโตคอล 8

2.4 โครงสร้างการติดต่อสื่อสารให้บริการ WAP 10

2.5 Apache Web Server 12

2.5.1 การดาวน์โหลด Apache Web Server 12

2.5.2 การติดตั้ง Apache Web Server 13

2.6 PHP 14

2.6.1 การดาวน์โหลด PHP และการติดตั้ง 15

2.6.2 การใช้งาน PHP บน Server 16

2.7 การใช้งาน WAP บน Apache Web Server 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 ความรู้พื้นฐานเกี่ยวกับ AVR MICROCONTROLLER 20

3.1 คุณสมบัติและขาต่อใช้งานของ AVR 20

3.1.1 รายละเอียดของ MCU (Microcontroller) 21

3.1.2 โครงสร้างภายในของ MCU 22

3.2 สถาปัตยกรรมภายในและรีจีสเตอร์ใช้งานทั่วไป 24

3.2.1 สถาปัตยกรรมภายใน 24

3.2.2 รีจิสเตอร์ใช้งานทั่วไป 27

3.3 การสื่อสารแบบอนุกรมของ AVR 30

3.4 พอร์ตอินพุตและเอาท์พุต 38

บทที่ 4 การวิเคราะห์และการออกแบบ 50

4.1 การออกแบบระบบโดยรวม 51

4.2 การออกแบบ WAP Site โดยใช้ ภาษา WML และ PHP 53

4.3 พอร์ตอนุกรม 57

4.3.1 การสื่อสารแบบอนุกรม 57

4.3.2 การสื่อสารแบบอะซิงโครนัส 58

4.3.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232 60

4.3.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ 61

4.3.5 UART (Universal Asynchronous Receiver Transmitter) 64

4.3.6 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232 66

4.3.7 ลักษณะสัญญาณอินพุตและเอาท์พุตของพอร์ต RS-232 67

4.4 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม 69

4.5 บล็อกไดอะแกรมในส่วนระบบไมโครคอนโทรลเลอร์ AVR 77

4.5.1 Flow Chart การทำงานในส่วนของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ 78

4.6 วงจรการทำงานของไมโครคอนโทรลเลอร์ 82

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 5 การทดลองและผลการทดลอง 85

5.1 การทดลองเขียน WAP Site โดยใช้ภาษา WML ร่วมกับ PHP 84

5.2 การทดลองการเขียนภาษา Visual Basic เพื่อใช้ติดต่อกับไมโครคอนโทรลเลอร์ 89

5.3 การทดลองในส่วนของ Hard ware 96

5.3.1การติดต่อระหว่าง Visual Basic กับ ไมโครคอนโทรลเลอร์ 96

5.3.2 การทำงานของส่วนอุปกรณ์ภายนอก 96

5.4 โปรแกรมทดสอบไมโครคอนโทลเลอร์ (AVR) 96

บทที่6 สรุปและวิจารณ์ผลการทดลอง 99

6.1 ปัญหาในการทดลอง 99

6.2 แนวทางการแก้ไข 99

6.3 แนวทางการพัฒนา 100

หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่ **หน้า**

รูปที่ 2.1 วิธีที่อุปกรณ์ WAP สื่อสารกับเซอร์เวอร์ของข้อมูล 6
รูปที่ 2.2 แบบโครงสร้าง WAP 7
รูปที่ 2.3 โปรโตคอล WAP 9
รูปที่ 2.4 โทรศัพท์มือถือติดต่อกับ WAP Server ผ่านเครือข่ายไร้สาย 11
รูปที่ 3.1 โครงสร้างภายนอกและตำแหน่งขาของ AVR 21
รูปที่ 3.2 บล็อคไดอะแกรม 23
รูปที่ 3.3 สถาปัตยกรรมแบบ RISC ของ AT90S4434/8535 25
รูปที่ 3.4 โครงสร้างของหน่วยความจำ 26
รูปที่ 3.5 โครงสร้างของรีจิสเตอร์ใช้งานทั่วไป 27
รูปที่ 3.6 รีจิสเตอร์ X ,Y และ Z 27
รูปที่ 3.7 รีจิสเตอร์ของ SREG 28
รูปที่ 3.8 โครงสร้างของ SPI 31
รูปที่ 3.9 SPI Master – Slave Interconnection 32
รูปที่ 3.10 SPI Transfer Format with CPHA 33
รูปที่ 3.11 รีจิสเตอร์ของ UDR 34
รูปที่ 3.12 รีจิสเตอร์ของ USR 34
รูปที่ 3.13 รีจิสเตอร์ของ UCR 36
รูปที่ 3.14 รีจิสเตอร์ของ UBRR 37
รูปที่ 3.15 รีจิสเตอร์ของ PORTA 38
รูปที่ 3.16 รีจิสเตอร์ของ DDRA 39
รูปที่ 3.17 รีจิสเตอร์ของ PINA 39
รูปที่ 3.18 ลักษณะโครงสร้างขาของพอร์ต A (Pins PA0 – PA7) 40
รูปที่ 3.19 รีจิสเตอร์ของ PORTB 41
รูปที่ 3.20 รีจิสเตอร์ของ DDRB 42
รูปที่ 3.21 รีจิสเตอร์ของ PINB 42
รูปที่ 3.22 โครงสร้างขาของพอร์ต B (Pins PB0 และ PB1) 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูป **หน้า**

รูปที่ 3.23 รีจิสเตอร์ของ PORTC 44
รูปที่ 3.24 รีจิสเตอร์ของ DDRC 44
รูปที่ 3.25 รีจิสเตอร์ของ PINC 45
รูปที่ 3.26 โครงสร้างขาของพอร์ต C (PC0 – PC5) 46
รูปที่ 3.27 รีจิสเตอร์ของ PORTD 47
รูปที่ 3.28 รีจิสเตอร์ของ DDRD 47
รูปที่ 3.29 รีจิสเตอร์ของ PIND 48
รูปที่ 3.30 โครงสร้างขาของพอร์ต D (PD0) 50
รูปที่ 4.1 การทำงานโดยรวมของระบบ 52
รูปที่ 4.2 Flow Chart โปรแกรมในหน้าเมนูหลัก 53
รูปที่ 4.3 Flow chart โปรแกรมในส่วนของการแสดงสถานะ 54
รูปที่ 4.4 Flow chart โปรแกรมในส่วนของการควบคุมเครื่องใช้ไฟฟ้า 55
รูปที่ 4.5 Flow chart โปรแกรมในส่วนของการควบคุมเครื่องใช้ไฟฟ้า(ต่อ) 56
รูปที่ 4.6 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม 58
รูปที่ 4.7 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส 59
รูปที่ 4.8 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 62
รูปที่ 4.9 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ 63
รูปที่ 4.10 บล๊อกไดอะแกรมในส่วนของไมโครคอนโทรเลอร์ 77
รูปที่ 4.11 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (1) 78
รูปที่ 4.12 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (2) 79
รูปที่ 4.13 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (3) 80
รูปที่ 4.14 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (4) 81
รูปที่ 4.15 วงจรไมโครคอนโทรลเลอร์และการต่อร่วมกับอุปกรณ์ภายนอก 83
รูปที่ 5.1 หน้าจอการเข้าสู่ WAP Site 85
รูปที่ 5.2 หน้าจอให้ผู้ใช้กดคีย์ 85

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูป **หน้า**

รูปที่ 5.3 หน้าจอการทำรายการหลัก 86
รูปที่ 5.4 หน้าจอแสดงสถานะของหลอดไฟ 86
รูปที่ 5.5 หน้าจอการขอรหัสผ่าน 87
รูปที่ 5.6 หน้าจอรหัสผ่านไม่ถูกต้อง 87
รูปที่ 5.7 หน้าจอการเข้าสู่การควบคุมหลอดไฟ 88
รูปที่ 8.5 หน้าจอของการป้อนข้อมูลและการตอบสนอง 88
รูปที่ 5.9 หน้าจอแสดงการป้อนสถานะผิดพลาด 89
รูปที่ 5.10 การเข้าสู่โปรแกรม 90
รูปที่ 5.11 การเริ่มต้นของโปรแกรมสถานะ 91
รูปที่ 5.12 การเริ่มต้นการทำงานโดยที่สถานะเริ่มต้นอยู่ที่สถานะ "1" 92
รูปที่ 5.13 การกดสถานะ "4" ที่อุปกรณ์ภายนอก 93
รูปที่ 5.14 การตอบสนองกลับของไมโครคอนโทรลเลอร์ 93
รูปที่ 5.15 การคีย์สถานะไม่ถูกต้อง 94
รูปที่ 5.16 โปรแกรมแสดงการเตือนการป้อนสถานะผิดพลาด 94
รูปที่ 5.17 การแสดงหลอดขาด 95

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง **หน้า**

ตารางที่ 2.1 ตารางเปรียบเทียบรูปแบบของข้อมูลระหว่าง WAP กับ WWW 7

ตารางที่ 3.1 SPI Pin Overrides 32

ตารางที่ 3.2 DDAn Effects on PORTA Pins 39

ตารางที่ 3.3 การใช้งานเป็นฟังก์ชั่นอื่นๆ ของพอร์ต B 41

ตารางที่ 3.4 การใช้งานพอร์ต B เป็นอินพุตและเอาท์พุต 42

ตารางที่ 3.5 การใช้งานของพอร์ต C 45

ตารางที่ 3.6 การใช้งานเป็นฟังก์ชั่นอื่นๆ ของพอร์ต D 47

ตารางที่ 3.7 การใช้งานพอร์ต D 48

ตารางที่ 4.1 บิตพาริตี้ของข้อมูล 60

ตารางที่ 4.2 การจัดขา DB-9 และ DB-25 62

ตารางที่ 4.3 ข้อมูลในแอดเดรส 0000 : 0411 H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม 68

ตารางที่ 5.1 รหัสและสถานะของอุปกรณ์ที่จะนำมาแสดงผล 84

ตารางที่ 5.2 การกดปุ่มคีย์บอร์ดแต่ละระดับให้กับ VB 97

ตารางที่ 5.3 การกดสวิตซ์แต่ละระดับให้กับ VB 97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 บทนำ

ในปัจจุบันนี้ มีเทคโนโลยีใหม่ๆ ผ่านเข้ามาในชีวิตของเราอย่างมากมาย โดยเฉพาะอย่างยิ่งเมื่อเราอยู่ในยุคที่ข้อมูลและข่าวสารซึ่งสามารถกระจายจากที่หนึ่งไปสู่ที่อื่นๆ ได้อย่างรวดเร็ว และสิ่งหนึ่งซึ่งเราต้องยอมรับว่าเป็นตัวขับเคลื่อนและผลักดันที่สำคัญอันนำมาสู่การเปลี่ยนแปลงของเทคโนโลยีที่ใช้ในการสื่อสารก็คือ อินเตอร์เน็ต ทุกวันนี้เราสามารถเข้าอินเตอร์เน็ตเพื่อไปค้นหาข้อมูล หรือซื้อสินค้าที่เราต้องการได้ตั้งแต่ราคาเพียง ร้อยกว่าบาท จนเป็นหลายร้อยล้านบาทก็มี และในปัจจุบันดูเหมือนว่าร้านค้า และธุรกิจหลายๆ ประเภทต่างเข้ามาเปิดร้านค้าอิเล็กทรอนิกส์ในอินเตอร์เน็ตเป็นจำนวนมาก หรือที่เราเรียกว่า E-Commerce ซึ่งสาเหตุหลักๆ ก็มาจากการลงทุนที่ไม่มากนัก และความคุ้มค่ากับการได้เปิดสินค้าของตัวเองไปให้ใครๆก็ได้ในโลกใบนี้ได้ชม และสามารถสั่งซื้อได้

เราจะได้เห็นว่าอินเตอร์เน็ตได้เข้ามามีบทบาทกับชีวิตประจำวันของเราอย่างมากมาย ตั้งแต่การเข้าไปรับ-ส่ง อีเมล์ , พูดคุยกับเพื่อนๆ , การดูโปรแกรมภาพยนตร์ หรือการค้นหาข้อมูลเพื่อนำมาทำรายงาน แต่อย่างไรก็ตามการเชื่อมต่อเข้ากับอินเตอร์เน็ต ในชีวิตจริงก็มีบางสิ่งที่เป็นอุปสรรคเช่นกัน โดยเฉพาะอย่างยิ่งเมื่อเรามีความจำเป็นต้องเดินทาง ไม่สามารถนำ PC ไปกับเราได้ หรือถ้าเรามี Notebook เราก็จำเป็นต้องซื้ออุปกรณ์ร่วมต่างๆ เช่น โมเด็ม และโทรศัพท์มือถือ นอกจากนี้ ขนาดของ Notebook เองก็ไม่ได้ให้ความคล่องตัวกับเรามากนัก จุดนี้เองเป็นสิ่งที่ WAP (Wireless Application Protocol) จะเข้ามาช่วยเหลือ และขจัดอุปสรรคของการเข้าถึงโลกอินเตอร์เน็ตของเราออกไป

ในปริญญานิพจน์ฉบับนี้ ได้ทำการศึกษาเรื่องการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือซึ่งประกอบด้วยส่วนสำคัญ 3 ส่วนคือ WAP , Visual Basic และไมโครคอนโทรเลอร์ AVR ในส่วนของ WAP ที่ใช้เป็นแบบไคลเอนท์ และเซิร์ฟเวอร์นั้น ภาษาที่ใช้เขียนคือ WML, WMLscript ส่วนของ Visual Basic เป็นตัวรับข้อมูลจาก WAP เข้ามาและส่งข้อมูลที่ได้มานี้ไปยังตัวไมโครคอนโทรลเลอร์ ซึ่งเป็นตัวควบคุมชุดอุปกรณ์ไฟฟ้า รวมถึงสามารถนำสัญญาณที่สั่งไปให้ชุดอุปกรณ์ไฟฟ้า ตอบสนองกลับมายัง Visual Basic เพื่อแสดงผลใน WAP Browser ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อวิเคราะห์และศึกษาการทำงานเกี่ยวกับการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือ
2. ศึกษาการทำงานของภาษา WML , PHP , Apache เพื่อนำมาใช้ในการติดต่อเชื่อมโยงข้อมูลระหว่างโปรแกรม
3. ศึกษาโปรแกรม Visual Basic ไมโครคอนโทรลเลอร์ เพื่อนำมาใช้ในการออกแบบการควบคุมอุปกรณ์ไฟฟ้า
4. เพื่อสามารถนำข้อมูลที่ได้จากศึกษาครั้งนี้ไปพัฒนาและปรับปรุงในครั้งต่อไป

1.3 ขอบเขตของงาน

1. สามารถวิเคราะห์ควบคุมการทำงานสำหรับระบบไฟฟ้าภายในบ้านด้วยโทรศัพท์มือถือได้
2. สามารถเขียนภาษา WML ร่วมกับ php เพื่อใช้ในการออกแบบ WAP Site ได้
3. สามารถเขียนภาษา Visual Basic เพื่อใช้ในการแสดงผลในการทำงานออกสู่หน้าจอคอมพิวเตอร์ได้
4. สามารถออกแบบวงจรเพื่อควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านโดยใช้ไมโครคอนโทรเลอร์
5. เข้าใจขั้นตอนและวิธีการปฏิบัติ รวมถึงการใช้งานและโปรแกรมคอมพิวเตอร์สำหรับออกแบบและควบคุมอุปกรณ์ไฟฟ้าได้
6. สามารถใช้โทรศัพท์มือถือเรียก URL ที่ Server เพื่อใช้งานในการควบคุมอุปกรณ์ไฟฟ้าภายในบ้านได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 โครงประกอบของโครงงาน

โครงงานฉบับนี้ได้แบ่งเนื้อหาออกเป็น 6 บท โดยมีบทนำ ในบทที่ 1 เป็นการกล่าวแนะนำหลักการเบื้องต้น และวัตถุประสงค์ในการทำโครงงานนี้ ส่วนรายละเอียดของบทต่างๆอีก 5 บท มีดังต่อไปนี้

บทที่ 2 ทฤษฎีเกี่ยวกับ WAP, PHP และการใช้งาน Server

บทที่ 3 ทฤษฎีพื้นฐาน ไมโครคอนโทรลเลอร์ AVR

บทที่ 4 การออกแบบ

บทที่ 5 ผลการทดลอง

บทที่ 6 สรุปผลการทดลอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเกี่ยวกับ WAP, PHP และการใช้งาน Server

2.1 บทนำ

WAP ย่อมาจาก Wireless Application Protocol เป็นข้อกำหนดในการสื่อสารในเครือข่ายแบบไร้สาย ในการใช้งานอินเตอร์เน็ตด้วยเครื่องคอมพิวเตอร์ ทำให้คุ้นเคยกับคำว่า "HTTP" , "TCP" หรือ "IP" สิ่งเหล่านี้ก็คือ ข้อกำหนดในการสื่อสารระหว่างคอมพิวเตอร์ในโลกของอินเตอร์เน็ต หรือ "โพรโตคอล" (Protocol) ซึ่งถือเป็นกลไกสำคัญที่ทำให้ติดต่อสื่อสารผ่านเครือข่ายระดับโลกนี้ได้ แต่เมื่อเปลี่ยนจากการใช้เครื่องคอมพิวเตอร์มาเป็นอุปกรณ์ ที่อาศัยเครือข่ายแบบไร้สายเป็นสื่อกลาง (เช่นเครือข่ายระบบโทรศัพท์ GSM ที่เรารู้จักกันดี) ก็ต้องมีการตั้งข้อกำหนดหรือกฎเกณฑ์ขึ้นมาใหม่ สำหรับใช้ในสภาพแวดล้อมใหม่นี้ (สภาพแวดล้อมใหม่ในที่นี้ก็คือ โทรศัพท์มือถือและเครือข่ายไร้สาย) ซึ่งข้อกำหนดหรือโพรโตคอลที่ขึ้นใหม่นี้ ก็มีชื่อว่า "WAP" นั่นเอง

ไม่ว่า HTTP , TCP , IP หรือ WAP ก็ล้วนเป็นโพรโตคอลทั้งสิ้นแต่มีหน้าที่ต่างๆ กันไป โดย WAP มีลักษณะแปลกไปจากโพรโตคอลอื่น ๆ ตรงที่ทำงานในสภาพแวดล้อมแบบไร้สาย ส่วนผู้ที่คิดค้นหรือสร้างข้อกำหนด-กฎเกณฑ์ที่ชื่อว่า WAP ก็คือกลุ่มบริษัทชั้นนำ เช่น Phone.com (ชื่อเดิม Unwired Planet) , อีริคสัน , โนเกีย , โมโตโรลา ฯลฯ ซึ่งเรียกว่า "WAP Forum"

บริษัทที่มาร่วมกับ Phone.com ในยุคแรกๆ ก็คือ อีริคสัน, โนเกีย และ โมโตโรลา จนถึงกลางปี 2000 ก็ได้เพิ่มเป็นกว่า 400 บริษัท กลุ่มนี้ได้คิดค้นแนวทางใหม่ในการพัฒนาเทคโนโลยีอินเทอร์เน็ตในเครือข่ายไร้สายและก็ได้เสนอโพรโตคอล WAP (Wireless Application Protocol) ซึ่งประกอบไปด้วยโพโตคอลย่อย 4 ชั้น คือชั้น Session Layer , ชั้น Transaction Layer , ชั้น Security Layer และชั้นTransport Layer อีกทั้งยังคิดค้นภาษาใหม่ คือ WML (Wireless Markup Language) ซึ่งเป็นภาษาที่ใช้สร้างแอปพลิเคชั่นสำหรับ WAP โดยเฉพาะ คุณลักษณะของภาษานี้ส่วนหนึ่งได้นำมาจากภาษาHTML เดิม บางส่วนก็นำมาจากภาษา HTML ด้วย

เมื่อต้นปี 1990 ในประเทศญี่ปุ่น มีบริษัทผู้ให้บริการเครือข่ายโทรศัพท์เคลื่อนที่ยักษ์ใหญ่ ชื่อ NTT DoCoMo ได้ให้บริการที่เรียกว่า I-Mode ซึ่งเป็นการใช้งานอินเตอร์เน็ตผ่านทางโทรศัพท์มือถือที่ใช้ระบบเครือข่ายของตนเอง บริการนี้แพร่หลายมากในประเทศญี่ปุ่นและบางส่วนในฮ่องกง แต่อย่างไรก็ดี ระบบนี้เป็น "ระบบปิด" เพราะจำกัดอยู่เพียงแค่โทรศัพท์มือถือที่ใช้เครือข่ายของ NTT DoCoMo เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวคิดของ โพรโตคอล WAP แนวคิดของโพรโตคอลนี้ก็คือการทำงานที่เหมาะสมกับเครือข่ายแบบไร้สาย ซึ่งมีข้อจำกัดมากมายเมื่อเทียบกับเครือข่ายอินเตอร์เน็ต ไม่ว่าจะเป็นอัตราเร็วในการรับ-ส่งข้อมูลซึ่งช้ากว่าแบนด์วิดธ์ (Bandwidth) หรือช่วงความถี่ในการสื่อสารซึ่งแคบ หรือการเชื่อมต่อ (Connection) ที่มีความเสถียรน้อยกว่า ดังนั้นการออกแบบโพรโตคอล WAP จึงต้องพยายามให้เกิดความเหมาะสมกับข้อจำกัดเหล่านี้ ข้อมูลที่ส่งไปในเครือข่ายชนิดนี้ก็จะเป็นแบบไบนารี (Binary) เพื่อลดขนาดของข้อมูลให้เหลือน้อยที่สุด และที่สำคัญจะต้องให้ WAP เป็นระบบเปิด ซึ่งหมายถึงไม่ได้จำกัดอยู่ที่เครือข่ายแบบใดแบบหนึ่ง ตัวอย่างของเครือข่ายไร้สายที่รองรับระบบ WAP ก็เช่น GSM , CDMA, TDMA, SMS ,CDPD เป็นต้น

สำหรับข้อมูลอื่นๆ ของ WAP Forum หรือข่าวคราวความเคลื่อนไหวเกี่ยวกับ WAP สามารถติดตามได้จากเว็บไซต์ของ WAP Form ที่ http://www.wapforum.org/

2.2 แบบโครงสร้างของ WAP (WAP Model)

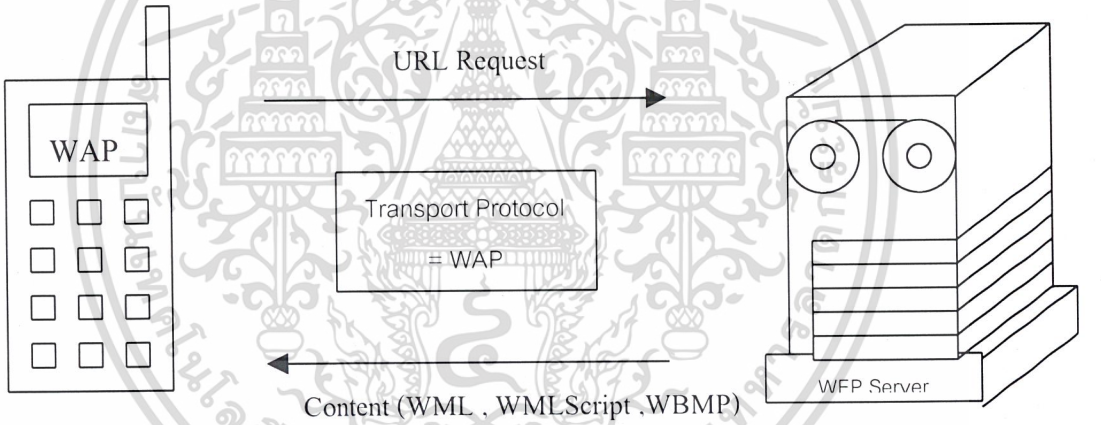
หลักการในการทำงานทั่วไปของ WAP และ WWW มีความใกล้เคียงกันอยู่มาก แต่สิ่งที่แตกต่างกันอย่างชัดเจนที่สุดระหว่าง WAP กับ WWW คือการทำงานของ WWW นั้นจะใช้สมมติฐานว่าทุก ๆ องค์ประกอบของ WWWทำงานอยู่ในสิ่งแวดล้อมเดียวกัน นั้นคือบนเครือข่าย TCP/IP ซึ่งเป็นเครือข่ายแบบมีสาย (Fixed Network) นั่นเอง แต่ในขณะที่ WAP มีสองสิ่งแวดล้อมที่ต่างกัน คือจะมีทั้งในส่วนของเครือข่ายแบบมีสาย และเครือข่ายแบบไม่มีสาย (Wireless Network) หรือเครือข่ายโทรศัพท์เคลื่อนที่ (Cellular Network) ซึ่งต้องให้ความสำคัญกับมันเป็นพิเศษ จึงใช้มาตรฐาน 3 อย่างคือโปรโตคอล (Protocol) ,การกำหนดชื่อที่เป็นมาตรฐาน (Naming Standard) และการจัดรูปแบบของข้อมูล (Content Format) เดียวกับอินเตอร์เน็ตไม่ได้ ทั้งนี้เนื่องจากข้อจำกัดต่างๆ และเนื่องจาก WAP Forum ได้กำหนดว่า WAP จะต้องใช้มาตรฐานที่มีอยู่แล้วบนอินเตอร์เน็ตให้มากที่สุด ดังนั้น URL จึงถูกนำมาใช้โดยไม่มีการดัดแปลงใดๆ ทั้งสิ้นเพื่อที่จะทำให้ WAP สามารถเข้าถึงข้อมูลที่มีอยู่แล้วบนอินเตอร์เน็ตได้ ส่วน MIME (Multipurpose Internet Mail Extensions) ก็สามารถนำมาใช้กับ WAP ได้โดยไม่ต้องดัดแปลงอะไรทั้งสิ้น แต่เพื่อเป็นการเพิ่มประสิทธิภาพในการทำงานให้ดีขึ้น ได้มีการกำหนดรูปแบบของข้อมูลขึ้นมาใหม่ ซึ่งจะมีพื้นฐานมาจาก XML คือ

- WML (Wireless Markup Language) และ WMLScript (Wireless Markup Language Script) เพื่อใช้ในการจัดรูปแบบของข้อมูลแล้วทำ User Interface ซึ่งก็จะเปรียบเทียบได้กับ HTML และ JavaScript บนอินเตอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- WBMP (Wireless Bitmap) เพื่อใช้ในการแสดงรูปภาพ (Image) ต่างๆ ซึ่งก็เทียบเท่ากับ GIF

ส่วนมาตรฐานส่วนที่เหลือคือ HTTP ซึ่งทำเป็นมาตรฐานในการส่งรับข้อมูล (Transport Protocol) หรือเป็นภาษาที่ใช้สื่อสาร ปัญหาเนื่องจาก HTTP เป็นโปรโตคอลที่มี "ความฟุ่มเฟือย" สูง จึงไม่เหมาะกับการนำมาใช้บนเครือข่ายที่ส่งรับข้อมูลได้อย่างช้าอย่างเครือข่ายโทรศัพท์เคลื่อนที่ ดังนั้นจำเป็นที่ WAP จะต้องแนะนำมาตรฐานใหม่ให้ได้เป็นที่รู้จัก ซึ่งก็ไม่ได้เป็นมาตรฐานแบบใหม่ แต่เป็นมาตรฐานเก่าที่นำมาปรับปรุงใหม่ โดยตั้งชื่อใหม่ว่า WSP หรือ Wireless Session Protocol ซึ่งเป็นเวอร์ชันดัดแปลงของ HTTP



รูปที่ 2.1 วิธีที่อุปกรณ์ WAP สื่อสารกับเซิร์ฟเวอร์ของข้อมูล

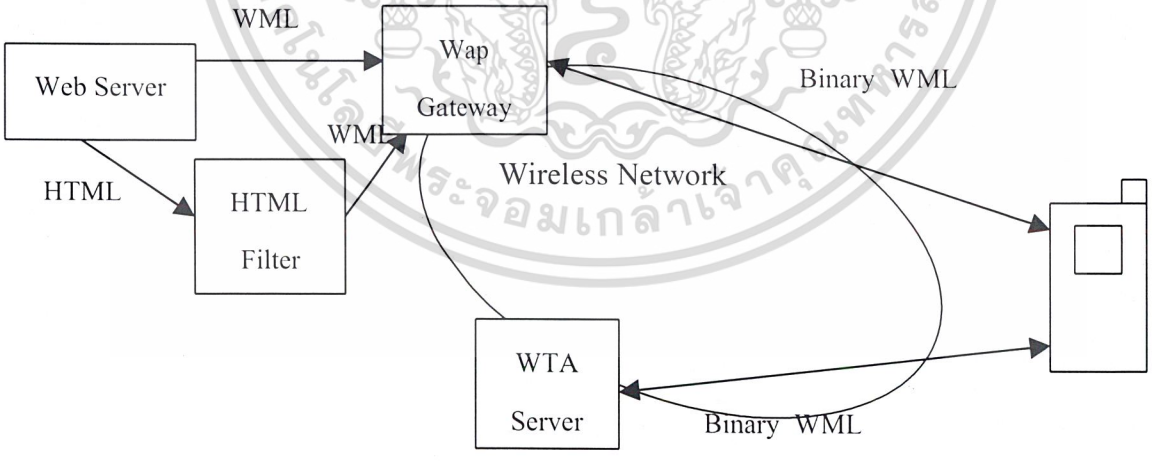
จากรูปที่ 2.1 จะเห็นได้ว่าอุปกรณ์ WAP ส่ง URL Request ไปยังเซิร์ฟเวอร์ของข้อมูลโดยใช้ WSP เป็นโปรโตคอลในการรับส่ง เมื่อได้รับ Request แล้วเซิร์ฟเวอร์ของข้อมูลก็จะส่งข้อมูลกลับมาโดยมีรูปแบบของข้อมูลดังตารางต่อไปนี้ (เปรียบเทียบกับ WWW)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางเปรียบเทียบรูปแบบของข้อมูลระหว่าง WAP กับ WWW

	WAP	WWW
Page description	WML	HTML
Script	WMLScript	JavaScript
Images	WBMP	GIG
Transport Protocol	WSP	HTTP

การแปลงข้อมูลที่มีอยู่แล้วมาใช้บน WAP จะต้องใช้ ASP (Active Server Page) หรือ CGI (Common Gateway Interface) หรือ JavaScript นอกจากนั้นแล้ว WAP Gateway บางตัวยังสามารถทำการแปลง HTML ให้เป็น WML ได้ด้วย และยังมีซอฟต์แวร์ที่ผลิตหลายรายได้ผลิตขึ้นมาสำหรับการแปลงข้อมูลจาก Web กับ WAP



รูปที่ 2.2 แบบโครงสร้าง WAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAP Gateway จะรับข้อมูล WML มาจาก Web Server โดยตรงถ้า Web Server มีข้อมูล WML หรือข้อมูล XML เพื่อให้บริการ อย่างไรก็ตามถ้า Web Server สามารถส่งได้เฉพาะ HTML เท่านั้น จะต้องนำเอา HTML Filter ซึ่งก็คือ ASP หรือ CGI มาใช้ในการแปลง HTML ให้เป็น WML ก่อนที่จะส่งไปยัง WAP Gateway เมื่อ WAP Gateway ได้รับข้อมูลที่เป็น WML แล้วจะทำการ Encode ข้อมูลเหล่านี้ให้อยู่ในรูปแบบของไบนารี (Binary) เพื่อทำให้มีขนาดเล็กลง ก่อนที่จะส่งไปให้อุปกรณ์ WAP

2.3 โพรโตคอล

Protocol (โพรโตคอล) คือ ข้อกำหนดหรือกติกาเกี่ยวกับการสื่อสารข้อมูลในระบบคอมพิวเตอร์ เนื่องจากระบบคอมพิวเตอร์มีอยู่หลากหลาย แต่ละระบบก็มีระเบียบวิธีในการทำงานที่แตกต่างกัน ถ้าจะให้ระบบที่แตกต่างกันติดต่อสื่อสารกันรู้เรื่อง ดังนั้นโพรโตคอลจึงเป็นสิ่งสำคัญมากในการสื่อสารข้อมูล

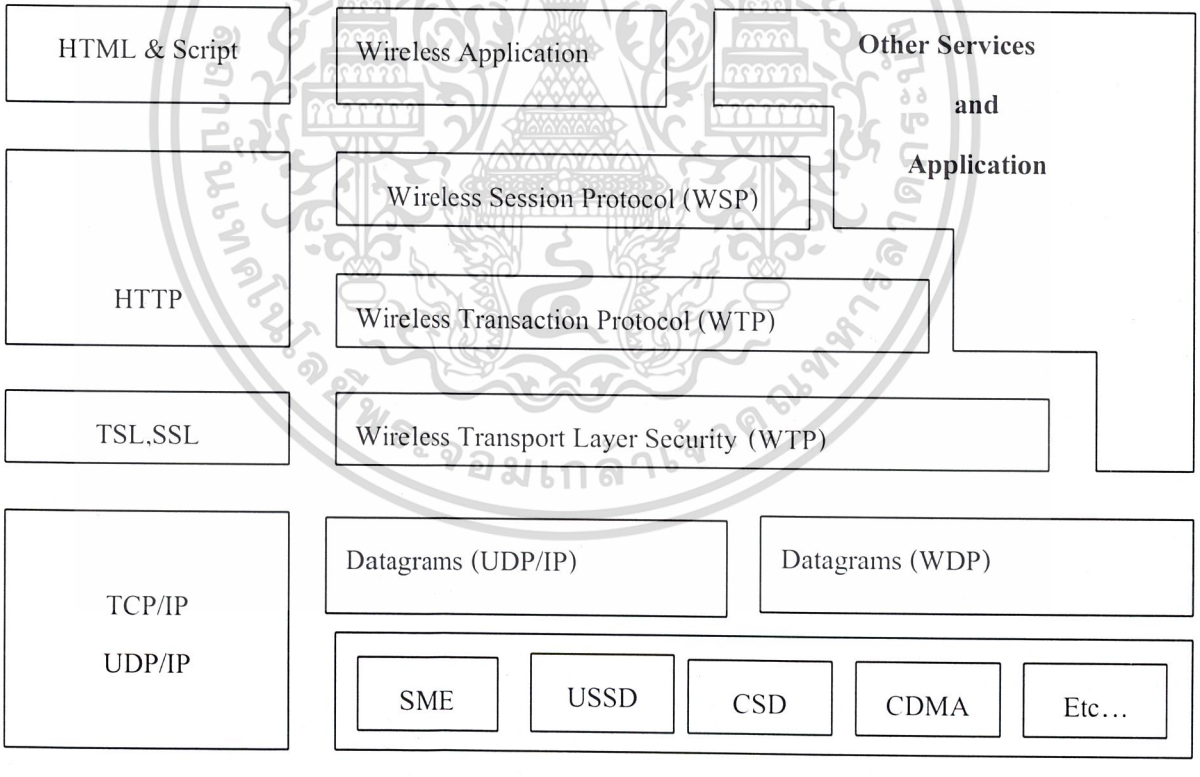
WAP เป็นโปรโตคอลของการสื่อสารแบบไร้สายที่มีหลายๆ ส่วนเหมือนกับโปรโตคอลที่มีใช้อยู่แล้วในปัจจุบัน เช่น โปรโตคอล HTTP และ HTML ซึ่งโครงสร้างของ WAP สามารถเขียนแสดงได้ดังในรูปที่ 2.3 และในแต่ละเลเยอร์สามารถนำมาเปรียบเทียบกับเทคโนโลยีที่มีอยู่ในปัจจุบัน และอธิบายอย่างคร่าวๆ ได้ ดังนี้

1. เลเยอร์ Wireless Application Environment (WAE) เป็นส่วนที่มีการทำงานเหมือนกับ HTML และ Java Script ก็คือ การกำหนดภาษา WML และ WML Script ที่ทำหน้าที่แสดงข้อมูล และข้อมูลที่จำเป็นให้กับผู้ใช้ผ่านทาง WAP บราวเซอร์
2. เลเยอร์ Wireless Session Protocol (WSP) และเลเยอร์ wireless Transaction Protocol (WTP) เป็นส่วนประกอบที่มีการทำงานคล้ายกับโปรโตคอล HTTP ก็คือส่วนที่ทำหน้าที่เชื่อมต่อการสื่อสารระหว่างเครื่องโทรศัพท์มือถือ และเซิร์ฟเวอร์หรือเกตเวย์ ในส่วนนี้จะเกี่ยวกับข้องกับเทคโนโลยีการสื่อสารแบบไร้สายโดยตรง เช่น ความเร็วของข้อมูลที่สามารถส่งได้ หรือเวลาที่ใช้ในการต่อเข้ากับเซิร์ฟเวอร์นั่นเอง
3. เลเยอร์ Wireless Transport Layer Security (WTLS) เป็นเลเยอร์ที่สร้างมาจากโปรโตคอลมาตรฐาน Transport Layer Security (TLS) ซึ่งจะรู้จักกันดีในชื่อของ Secure Sockets Layer (SSL) ซึ่งจะทำหน้าที่ในการรักษาความปลอดภัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของข้อมูล และป้องกันภัยจากผู้บุกรุกชนิดต่างๆ (attacker) โดยมีบริการของ Data integrity , Confidential , Authentication และป้องกัน Denial of Service เป็นเครื่องมือที่ใช้ในการป้องกันดังกล่าว

4. เลเยอร์ Datagram หรือ Transport Layer (WDR) มีการทำงานเหมือนโปรโตคอล TCD หรือ UDP ที่ทำหน้าที่ในการส่งข้อมูลระหว่างเลเยอร์ชั้นบนกับเครื่องเซิร์ฟเวอร์หรือเครื่องโทรศัพท์มือถือ (client) ผ่านระบบเครือข่ายไร้สายชนิดต่างๆ



รูปที่ 2.3 โปรโตคอล WAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 โครงสร้างการติดต่อสื่อสารให้บริการ WAP

อุปกรณ์ที่มีความสามารถในการติดต่อสื่อสารผ่าน WAP ได้แก่

1.Web Gateway ทำหน้าที่เป็นเครื่องคอมพิวเตอร์ซึ่งทำงานเป็นตัวกลางระหว่างโทรศัพท์มือถือและเครื่องเว็ปเซิร์ฟเวอร์ที่เก็บข้อมูลของ WAP Site ในอินเตอร์เน็ต สาเหตุที่ต้องอาศัย WAP Gateway เป็นตัวกลางก็เพราะว่าโทรศัพท์มือถือทำงานอยู่ในระบบไร้สาย (ในประเทศไทยก็คือ GSM , 900 /1800) จะใช้โปรโตคอล WAP ซึ่งเป็นโปรโตคอลต่างระบบกับโปรโตคอลที่ใช้ใน WAP Site ที่อยู่ในเครือข่ายอินเตอร์เน็ต ดังนั้นจึงต้องอาศัย WAP Gateway เป็นตัวกลางเชื่อมต่อให้และทำการแปลงโปรโตคอลเพื่อจะทำให้สามารถติดต่อสื่อสารกันได้ WAP Gateway ติดต่อได้ 2 ทาง คือใช้ WAP Gateway จากผู้ให้บริการเครือข่ายโทรศัพท์มือถือ และ ใช้ WAP Gateway สาธารณะ

1) WAP Gateway จากผู้ให้บริการเครือข่ายโทรศัพท์มือถือ ผู้ใช้สามารถสอบถามการตั้งค่า WAP Gateway ในโทรศัพท์ WAP Phone จากผู้ขาย

2) WAP Gateway สาธารณะเป็นวิธีหนึ่งโดยผู้ใช้โทรศัพท์มือถือโทรติดต่อเข้าไปยังผู้ให้บริการอินเทอร์เน็ต (ISP) แล้วระบุ IP Address ของ WAP Gateway สาธารณะ (ผู้ใช้ระบุค่าลงในโทรศัพท์มือถือ) IP Address หาได้จาก WAP Site WAP of Thailand ซึ่งรวบรวมรายชื่อ WAP Gateway สาธารณะเอาไว้ โดยเข้าไปหาได้ที่ WWW.Wapofthailand.com

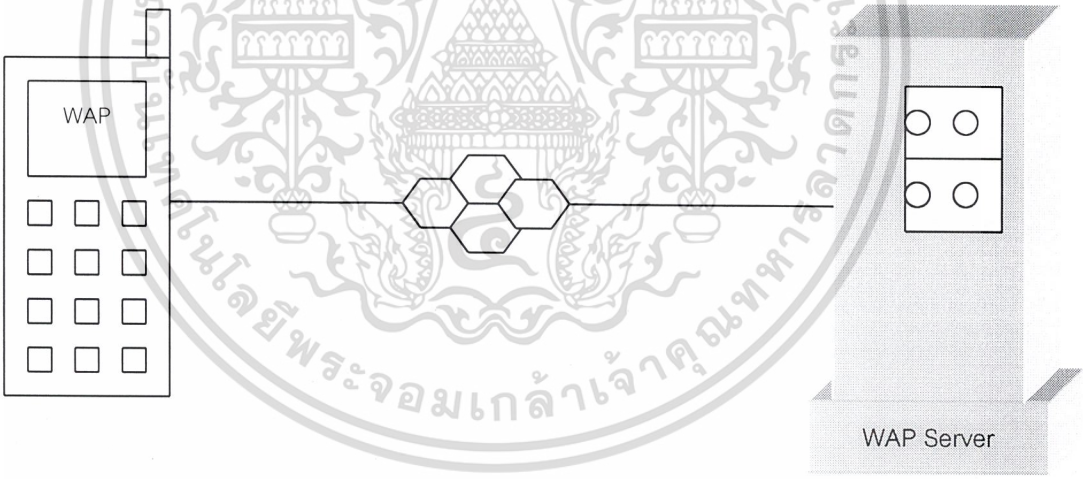
2. WAP Browser หรือไมโครบราวน์เซอร์ ที่ใช้ในการดู WAP Site ต่างๆ เปรียบเสมือน เว็บบราวน์เซอร์ในเครื่องพีซีที่ใช้เรียกดูเว็บไซต์ในระบบ WWW เช่น Netscape หรือ IE WAP brownser ที่อยู่ในโทรศัพท์มือถือ ต้องอิงมาตรฐานของ WAP เช่น WAP 1.1 นอกจากนี้ยังสามารถเข้าไปเยี่ยมชมเว็บไซต์ที่มีบริการของ WAP ผ่านทางการให้บริการของโทรศัพท์มือถือโดยทั่วไป ยังสามารถใช้โปรแกรมจำลองการทำงานของเครื่องโทรศัพท์มือถือ (Simulator) ที่มีอยู่บน Internet เข้าไปเยี่ยมชมเว็บไซต์ได้อีกด้วย เช่น เว็บไซต์ www.gelon.net ที่จะอนุญาติให้สามารถใส่ URL ของเว็บไซต์ที่ให้บริการ WAP เพจและโปรแกรมก็จะแสดงข้อมูลของเว็บไซต์นั้น ออกมาทางหน้าต่างโปรแกรมจำลอง นอกจากนี้ยังสามารถใช้โปรแกรม WAP Brownser บนเครื่องพีซีก็ได้ ซึ่ง Nokia Toolkit ก็สามารถทำหน้าที่เป็น WAP บราวน์เซอร์ได้เหมือนกัน โดยพิมพ์ URL ที่ต้องการลงในช่อง location และกด Enter บราวน์เซอร์ก็จะพยายามติดต่อกับ WAP Gateway ทางฝั่งเซิร์ฟเวอร์เพื่อขอข้อมูลมาแสดงแสดงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. WAP Server มีความหมายสั้นๆ คือ เครื่องคอมพิวเตอร์ที่เป็นทั้งเว็บเซิร์ฟเวอร์และ WAP Gateway คือรวมฟังก์ชันการทำงานของทั้ง 2 อย่างเข้าด้วยกัน หรือแทนด้วย WAP Server = เว็บเซิร์ฟเวอร์ + WAP Gateway

4. WAP Proxy คือเครื่องคอมพิวเตอร์ที่เป็นตัวกลางระหว่างไคลเอนต์และเว็บเซิร์ฟเวอร์ในระบบ WAP ประโยชน์ของ WAP Proxy ก็คือเก็บเอกสารที่ไคลเอนต์ (โทรศัพท์มือถือ) เรียกใช้จากเว็บเซิร์ฟเวอร์บ่อยๆ ไว้ในตัว เมื่อไคลเอนต์ร้องขอเอกสารเหล่านั้น WAP Proxy ก็จะส่งเอกสารไปให้แก่ไคลเอนต์เอง โดยไม่ต้องเสียเวลาให้กับเว็บเซิร์ฟเวอร์ส่งเอกสารนั้นๆ มาใหม่ทุกครั้งที่ไคลเอนต์ร้องขอ

WAP Gateway ก็มีคุณสมบัติอย่างหนึ่ง คือ เก็บเอกสารที่โทรศัพท์มือถือร้องขอไปยังเว็บเซิร์ฟเวอร์บ่อยๆ เช่นกัน ดังนั้นส่วนมากแล้ว WAP Gateway และ WAP Proxy จึงมักจะเป็นเครื่องคอมพิวเตอร์เครื่องเดียวกัน หรือจะเขียนว่า WAP Gateway/Proxy ก็ไม่ผิด



รูปที่ 2.4 โทรศัพท์มือถือติดต่อกับ WAP Server ผ่านเครือข่ายไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 Apache Web Server

การฝึกเขียน WAP Application ด้วยภาษา WML และ WMLScript เบื้องต้น อาจไม่จำเป็นต้องติดตั้งเว็บเซิร์ฟเวอร์ แต่ถ้าต้องการเพิ่มศักยภาพของ WAP Application ด้วยการใช้ CGI/Perl, ASP, PHP จำเป็นที่จะต้องติดตั้งเว็บเซิร์ฟเวอร์ และที่สำคัญก็คือ ที่อยู่ เอกสาร HTML, WML ในอินเตอร์เน็ตก็คือ เครื่องมือที่เป็นเว็บเซิร์ฟเวอร์ เมื่อเป็นเช่นนี้แล้วหากจะพัฒนาแอปพลิเคชั่นขึ้นมา ไม่ว่าเป็น แอปพลิเคชั่นสำหรับเว็บหรือสำหรับ WAP ก็ตาม ก็ควรทำความเข้าใจและคุ้นเคยกับเว็บเซิร์ฟเวอร์เสียก่อน และวิธีที่สะดวกที่สุดในการทำความเข้าใจการทำงานของมัน ก็คือจำลองเครื่องพีซีให้เป็นเว็บเซิร์ฟเวอร์ เพื่อใช้ฝึกเล่นๆ, ฝึกทดลอง, ฝึกทดลอง, ฝึกสร้างแอปพลิเคชั่น และถือเป็นการทดสอบการทำงานของแอปพลิเคชั่น ก่อนที่จะอัปโหลดไปยังเว็บเซิร์ฟเวอร์จริงๆ ต่อไป

การเลือกใช้โปรแกรมเว็บเซิร์ฟเวอร์ เช่น Omni HTTPd, PWS (Personal Web Server), IIS4 (Internet Information Server) หรือ Xitami เป็นต้น แล้วแต่ความถนัดของแต่ละคน แต่อีกเหตุผลหนึ่งในการเลือกโปรแกรมเว็บเซิร์ฟเวอร์ มักจะอยู่ที่การรองรับสคริปต์มากกว่า เช่น ถ้าจะเขียนสคริปต์ ASP (Active Server Pages) ก็ควรใช้ PWS หรือ IIS4 ของไมโครซอฟต์เพื่อความสะดวก แต่ถ้าเขียนสคริปต์ PHP ก็ควรใช้ Apache (ถึงแม้ว่า PHP จะทำงานภายใต้เว็บเซิร์ฟเวอร์ชนิดอื่นๆ ได้ด้วย เช่น IIS4) เพราะ PHP และ Apache เป็นของฟรีเหมือนกัน อีกทั้งถ้าใช้งานร่วมกับโปรแกรมดาต้าเบสเซิร์ฟเวอร์อย่าง MySQL ด้วยแล้ว ถือว่าโปรแกรมทั้งสามเข้ากันได้ดีทีเดียว แต่ในโครงงานนี้ไม่มีการใช้งานเกี่ยวกับโปรแกรมดาต้าเบสเซิร์ฟเวอร์หรือการเก็บข้อมูลในปริมาณมากๆ ฉะนั้นจึงไม่มีความจำเป็นที่จะต้องใช้งานโปรแกรมเกี่ยวกับดาต้าเบสเซิร์ฟเวอร์แต่ไม่ได้มีการศึกษาต่อในโครงงานนี้

2.5.1 การดาวน์โหลด Apache Web Server

โปรแกรม Apache Web Server เป็นฟรีแวร์ซึ่งสามารถดาวน์โหลดได้จากเว็บของผู้จัดทำโปรแกรมเอง ซึ่งด้วยเหตุผลที่โปรแกรมนี้เปิดให้ดาวน์โหลดได้ฟรีจึงเป็นอีกเหตุผลหนึ่งที่เลือกใช้ตัวโปรแกรมนี้เพราะเหมาะสำหรับผู้พัฒนาเว็บได้สามารถใช้โปรแกรม Web Server ที่ไม่ผิดกฎหมายทำให้สามารถทดลองรันเว็บของตนเองก่อนที่จะอัฟโหลดขึ้นไปยัง Server และยังสามารถใช้งานเป็น Sever เพื่อให้บุคคลอื่นสามารถเข้าถึงเว็บไซต์ของผู้เขียนโดยไม่ต้องเช่า Server จากผู้ให้บริการอีกด้วย

การดาวน์โหลดไฟล์เริ่มจากเข้าไปที่ http://www.apache.org แล้วเลือกเข้าไปที่ HTTP Server เลือก download ในโครงงานนี้เลือกใช้ เวอร์ชั่น 1.3.12 ซึ่งเป็นเวอร์ชั่นเก่าเลือกที่โฟลเดอร์ binaries

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือก win32 เลือก old ตอนนี้จะอยู่ที่ http://www.apache.org/dist/httpd/binaries/win32/old จะเจอกับไฟล์ ใช้สำหรับติดตั้งมีดาวน์โหลดที่ apache_1_3_12_win32.exe ซึ่งเป็นไฟล์ที่สำหรับติดตั้งในเครื่องไมโครคอมพิวเตอร์ได้ทันที ไฟล์มีขนาดประมาณ 3M

2.5.2 การติดตั้ง Apache Web Server

เมื่อดาวน์โหลดโปรแกรมเรียบร้อยแล้วก็ทำการติดตั้งไฟล์ของ Apache Web Server ลงในเครื่องพีซีที่จะนำมาทำเป็น Server การลงโปรแกรม Sever ควรติดตั้งไว้แยกจากโปรแกรมตัวอื่นๆ หรือแยกมาทำเป็นโฟลเดอร์ย่อยต่างหากเพราะจะได้สามารถเข้าถึงและใช้งานโปรแกรมได้ง่าย ในโครงงานนี้ได้แยกส่วนของโปรแกรมไว้ที่ไดร์ D เพื่อสะดวกต่อการจัดการโปรแกรม เมื่อทำการติดตั้งเสร็จสมบูรณ์แล้วให้เข้าไปดูไฟล์และโฟลเดอร์ต่างๆ ที่ Apache สร้างขึ้นเองในระหว่างการติดตั้ง โดยสังเกตที่โฟลเดอร์ conf และ htdocs ให้ดีเพราะต่อไปต้องเข้ามาแก้ไขเกี่ยวกับโฟลเดอร์ทั้งสองบ่อยๆ

โฟลเดอร์หรือไดเรกทอรี htdocs ตามข้อกำหนดในการทำงานของเซิร์ฟเวอร์ ไฟล์เอกสารต่างๆ ซึ่งจะถูกส่งไปยังผู้ร้องขอเอกสาร เช่น HTML, GIF, JPEG เป็นต้น จะต้องเก็บไว้ในไดเรกทอรี htdocs ของเซิร์ฟเวอร์เท่านั้น ดังนั้นหากสร้างไฟล์ WML, WMLScript, WBMP ขึ้นมา ก็ต้องเก็บไว้ในไดเรกทอรีนี้เหมือนกัน แต่เพื่อความเป็นระเบียบ ไม่ให้ไฟล์ของระบบปะปนกันจึงได้ต้องสร้างไดเรกเทรีเพื่อเก็บเอกสารต่างๆ ไว้แยกต่างหากออกไปโดยเฉพาะ

สำหรับไดเรกทอรี conf มีไฟล์สำคัญในการทำงานของเซิร์ฟเวอร์อยู่ไฟล์หนึ่งคือ httpd.conf หลังจากติดตั้ง Apache แล้วต้องเข้าไปแก้ไขข้อมูลในไฟล์ httpd.conf เพื่อบอกให้ Apache รู้จักเครื่องที่จะรันเซิร์ฟเวอร์ก่อน ซึ่งก็คือเครื่องพีซีเองโดยใช้โปรแกรมเท็กซ์เอดิเตอร์ (เช่น NotePad) เปิดไฟล์ httpd.conf แล้วค้นหาคำว่า ServerName new.host.name เมื่อค้นเจอแล้ว หากข้างหน้าคำๆนี้มีเครื่องหมาย # กำกับอยู่ ให้ลบเครื่องหมาย # ออก (เพราะเครื่องหมายนี้ใช้สำหรับระบุว่าข้อความที่ตามมาไม่มีผลกับการทำงาน นั่นคือ ทำหน้าที่เหมือนกับการกำหนดข้อความที่เป็น comment) จากนั้นให้แก้ไขข้อความดังกล่าวเป็น ServerName localhost หรือ ServerName 127.0.0.1 แทนก็ได้ โดย 127.0.0.1 เป็นหมายเลข IP address ของตัวเซิร์ฟเวอร์เอง ซึ่งก็คือเครื่องพีซีที่ติดตั้ง Apache หรือคำว่า localhost ก็คือคีย์เวิร์ดซึ่งหมายถึงตัวเซิร์ฟเวอร์เองเช่นเดียวกัน

เมื่อแก้ไขข้อมูลในไฟล์ httpd.conf เสร็จแล้ว ให้เริ่มรันโปรแกรม Apache Web Server โดยสั่งทางเมนูในระบบ Windows คือ **Start -> Programs -> Apache Web Server -> start Apache** ... ก็จะพบกับหน้าต่างซึ่งมีข้อความลักษณะดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Apache/1.3.12 (Win32) running…

จากนั้นถ้าไม่ต้องการให้หน้าต่างนี้รบกวนการทำงาน ก็ย่อ (minimize) หน้าต่างนี้เอาไว้ แต่ห้ามปิดหน้าต่างนี้ เพราะการปิดหน้าต่างนี้ เท่ากับปิดการทำงานของเว็บเซิร์ฟเวอร์ แถมยังเป็นการปิดที่ไม่ปกติ หรือผิดวิธี ซึ่งมีผลให้เมื่อเปิดเซิร์ฟเวอร์ครั้งต่อไป จะพบข้อความแสดงออกมา เมื่อถึงตอนนี้ ก็ถือได้ว่าติดตั้ง Apache Web Server สำเร็จแล้วเพียงต่อ Apache จะรู้จักเฉพาะเอกสารในตระกูลของเว็บ เช่น HTML กับไฟล์ภาพจำพวก GIF หรือ JPEG ยังไม่รู้จักเอกสารตระกูล WAP เช่น WML, WMLScript หรือ WBMP ดังนั้นต้องเข้าไปแนะนำให้ Apache รู้จักเพื่อที่ว่า Apache จะได้สามารถให้สามารถให้บริการ WAP Site ด้วย

2.6 PHP

สำหรับผู้ที่เคยสร้างเว็บแอปพลิเคชัน (Web application) อาจเคยใช้ CGI/Perl, ASP, PHP หรือ JSP (Java Server Page) อย่างใดอย่างหนึ่ง ซึ่งทั้งหมดเหล่านี้จัดเป็นภาษาสคริปต์ทางฝั่งเซิร์ฟเวอร์ (server-side script) ที่ช่วยสร้างแอปพลิเคชันให้มีคุณสมบัติและความสามารถเพิ่มขึ้นมาจากภาษา HTML รวมทั้งเหนือกว่าภาษาสคริปต์ทางฝั่งไคลเอนต์ (client-side script) อย่าง JavaScript หรือ VBScript ด้วยเพราะสามารถติดต่อกับฟังก์ชันต่างๆ ทางฝั่งเซิร์ฟเวอร์โดยตรง เช่น การติดต่อกับฐานข้อมูลเป็นต้น ระบบ WAP ภาษาสคริปต์ทางฝั่งเซิร์ฟเวอร์เหล่านั้นก็ยังนำมาใช้ได้อยู่ ทั้งนี้เพราะ WAP ก็คล้ายกับเว็บ เพียงแต่มาใช้ภาษา WML ซึ่งสิ่งสำคัญที่เพิ่มขึ้นมา และถือเป็นหัวใจสำหรับการใช้ภาษาสคริปต์ทางฝั่งเซิร์ฟเวอร์เหล่านี้ก็คือ การระบุ Content-type เป็น “text/vnd.wap.wml” เพื่อให้ User Agent (ซึ่งก็คือ WAP Browser) รู้ว่าเป็นเอกสาร WML จะได้แสดงผลอย่างถูกต้อง และต้องเพิ่มส่วนของ XML Declaration และ Document Type Definition ตรงส่วนต้นของเอกสารด้วย

สำหรับ PHP (Professional Home Page) เป็นโปรแกรมประเภท Server Side Scripting ประเภทหนึ่ง คล้ายๆ กับ SSI และ ASP ซึ่งสามารถสร้างเว็บไซต์แบบไดนามิกได้ สามารถดึงข้อมูลจากฐานข้อมูลได้ และการทำงานของ PHP จะเหมือนกับการรวมเอาความสามารถของภาษา C, Java, Peal เข้าไว้ด้วยกัน หรือกล่าวอีกนัยหนึ่ง PHP คือการนำข้อดีของภาษา C, Perl และ Java มารวมกัน

PHP 1.0 ถูกสร้างขึ้นมาเพื่อใช้ในการตรวจสอบข้อมูลผู้เข้าชมเว็บเพจ Rasmus Lerdorfs online และช่วยในการจัดการระบบข้อมูล SQL ต่อมาได้ถูกพัฒนาเป็น PHP 2.0 ซึ่งออกแบบให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถใช้งานร่วมกับแท็ก HTML ได้ สามารถรับค่าที่ได้จากการ Submit จากผู้ใช้ได้ เข้าถึงฐานข้อมูลได้ง่ายกว่าเวอร์ชั่นก่อน PHP 2.0 จึงเป็นที่นิยมอย่างรวดเร็ว

PHP ถูกนำมาใช้ในเว็บไซต์ธุรกิจในปี 1996 โดยเว็บไซต์ SuperCuts (www.supercuts.com) ซึ่งใช้ PHP ในการสร้างฐานข้อมูลสำหรับลูกค้า ต่อมาประมาณกลางปี 1998 PHP 3 ได้ถูกสร้างขึ้นโดยได้รับการแก้ไขข้อบกพร่องจากรุ่นก่อน จนกระทั่งปี 1999 มีเว็บไซต์กว่า 100,000 ไซต์ที่หันมาใช้ PHP ช่วยในการดำเนินธุรกิจบนเว็บของตน PHP (PHP: Hypertext Preprocessor) เป็นไฟล์นามสกุล .php ที่ทำหน้าที่เป็น CGI ตัวหนึ่งภายใน Web Server ภายในไฟล์นี้ส่วนมากจะเหมือนกับเอกสาร HTML เพราะสามารถเขียนแท็ก HTML ลงในโปรแกรม PHP ได้ และเราสามารถบรรจุแท็กซึ่งเป็นคำสั่งของ PHP ลงไปได้ด้วยโดยใส่เอาไว้ภายในเครื่องหมาย <?...?> จะเห็นได้ว่า PHP จะเป็นการรวมกันระหว่าง HTML และ Server Side Script อาจกล่าวได้ว่า PHP ก็คือ HTML ที่บรรจุคำสั่ง PHP ลงไปให้สามารถประมวลผลที่ Server ได้นั่นเอง

เนื่องจาก PHP ต่างจากการเขียน CGI ด้วยภาษาอื่นตรงที่มีขอบเขตของโปรแกรม เพื่อแบ่งพื้นที่กันระหว่าง HTML กับโค้ด PHP ซึ่งขอบเขตของโปรแกรมนี้ จะละเอาไว้ในส่วนใดก็ได้ เพราะฉะนั้น ถ้าดูภาพรวมของ PHP แล้ว PHP มีลักษณะของภาษาคล้ายกับ Perl และสามารถเขียนเป็น CGI ที่ดีได้ พร้อมกับความสามารถทางด้านการจัดการโปรแกรม โดยให้อยู่ภายในขอบเขตที่กำหนด จะเห็นได้ว่าในส่วนของ PHP ได้รวมรูปแบบที่ดีทางไวยากรณ์ของภาษา Perl, C, Java ไว้ภายในขอบเขตของสัญลักษณ์ <?....?> เพื่อแบ่งแยกขอบเขตระหว่าง HTML และโค้ด PHP โดยที่ไม่ต้องยุ่งกับ HTML หรือตัวโปรแกรมเลย ดังนั้นจึงสามารถแทรกโค้ด JavaScript หรือแทรกแท็ก HTML มาตรฐานได้ ในขณะเดียวกัน ก็สามารถสั่งให้โปรแกรมรันงานบางอย่างได้อีกด้วย นอกจากนี้ PHP ยังมีข้อดีอีกเช่น มีการ Complile และ Execute ได้อย่างรวดเร็ว สามารถทำงานได้ทั้งบน Web Server ในระบบ UNIX, Macintosh, Windows (NT/2000/98/95/me) เพราะได้รองรับการทำงานแบบ Cross Platform

2.6.1 การดาวน์โหลด PHP และการติดตั้ง

ข้อดีอีกประการหนึ่งของ PHP ก็คือเป็นฟรีแวร์ ซึ่งสามารถดาวน์โหลดได้ฟรีจากเว็บไซต์ของ PHP ได้เลยที่ www.php.net ทั้ง Source Code และเอกสารอ้างอิงที่มีมากกว่า 1Mbit นอกจากนั้นใช้ทำงานในด้านการเข้าถึงฐานข้อมูลได้หลายประเภทอีกด้วย

การดาวน์โหลดโปรแกรมเพื่อนำมาติดตั้งเริ่มเมื่อ เข้าไปยังเว็บไซต์ของ PHP ที่เว็บไซต์ของ PHP นี้จะมี help อยู่ด้วยไว้สำหรับอธิบายความหมายของคำสั่งรวมทั้งตัวอย่างของโปรแกรมให้ดาวน์โหลดด้วย เมื่อเข้าไปแล้วให้คลิกที่ downloads จากนั้น ให้ดูที่ Windows Binaries จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมให้ดาวน์โหลดอยู่ ให้เลือกดาวน์โหลดที่ PHP 4.1.1 installer ซึ่งมีขนาด 909kbit จะได้ไฟล์ที่ชื่อว่า php-4.1.1-installer.exe มาแล้วจึงนำมาติดตั้ง

เมื่อดาวน์โหลดมาแล้ว ขั้นตอนแรกจะต้องทำการ install โปรแกรม PHP 1.1.1 ลงในเครื่องก่อนซึ่งก็จะเหมือนกับการลงโปรแกรมทั่วไป การวางโปรแกรมควรวางในตำแหน่งที่เป็นสัดส่วนเพื่อสะดวกต่อการใช้งาน หรือวางรวมไว้ในโฟลเดอร์เดียวกับ Apache Web Server ก็ได้ ต่อมาโปรแกรม install ก็จะให้กำหนด Web Server ที่จะนำ PHP ไปใช้งานจากนั้นก็ทำการติดตั้งโปรแกรมตามปกติ

2.6.2 การใช้งาน PHP บน Server

เมื่อทำการติดตั้งเรียบร้อยแล้ว ขั้นตอนต่อไปก็คือทำให้ Apache รู้จัก PHP เสียก่อน วิธีการก็เริ่มจาก ให้ทำการ ก๊อบปี้ไฟล์ที่มีนามสกุล .dll ยกเว้นไฟล์ที่มีชื่อขึ้นต้นด้วย php_ ไปใส่ไว้ใน c:\windows\system\ จากนั้นก๊อบปี้ไฟล์ php.ini-dist ไปใส่ใน c:\windows\ แล้วเปลี่ยนชื่อไฟล์เป็น php.ini ซึ่งเป็น text file ธรรมดา จากนั้นให้ใช้โปรแกรม NotePad เปิดไฟล์นี้ขึ้นมาแล้วแก้ไขข้อมูลในไฟล์ 3 จุด ซึ่งอยู่ในหัวข้อ Paths and Directories ดังนี้

doc_root = c:\apache\htdocs;

user_dir = c:\apache\users;

extension_dir = c:\php;

เข้าไปแก้ไขข้อมูลในไฟล์ c:\apache\conf\httpd.conf ค้นหาคำว่า ScriptAlias ซึ่งจะเจอข้อความ ScriptAlias /cgi-bin/ "c:/apache/cgi-bin/" แล้วจึงเปลี่ยนให้เป็น ScriptAlias /php/ "c:/apache/php/" ค้นหาบรรทัดที่มีคำว่า AddType ซึ่งจะเจอข้อความ

#AddType application/x-httpd-php .phtml

#AddType application/x-httpd-php-source .phps

ให้ลบเครื่องหมาย # ออก (เครื่องหมายนี้เป็นสัญลักษณ์บ่งบอกว่า ข้อความในบรรทัดที่มีเครื่องหมายนี้กำกับอยู่ จะไม่มีผลต่อการทำงานเพราะมีสถานะเป็น comment จึงต้องลบออกเพื่อให้ Apache รู้จักไฟล์ในตระกูล PHP) จากนั้นเพิ่มข้อมูลอีก 1 บรรทัดเข้าไป โดยให้ผลออกมาเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

AddType application/x-httpd-php .php

AddType application/x-httpd-php .phtml

AddType application/x-httpd-php-source .phps

การเพิ่มข้อความใหม่ดังกล่าว หมายความว่า เวลาใช้งานเราจะต้องตั้งชื่อไฟล์ของ PHP ให้มีนามสกุลเป็น .php ต่อมาค้นหาคำว่า Format:Action ซึ่งจะเจอข้อความ

#Format: Action media/type /cgi-script/location

#format: Action handler-name /cgi-script/location

แล้วเพิ่มข้อความ **Action application/x-httpd-php "/php/php.exe"** save ไฟล์ httpd.conf แล้วทดลองรันโปรแกรม Apache ผ่านทาง Browser ว่าสามารถใช้งานได้หรือไม่

2.7 การใช้งาน WAP บน Apache Web Server

การบอกให้ Apache รู้จักเอกสารตระกูล WAP เพื่อให้บริการ WAP Site ทำได้ 2 วิธีคือ

1. แก้ไขข้อมูลในไฟล์ httpd.conf
2. เพื่อไฟล์ htaccess ลงในไดเรกทรอรีที่ใช้จัดเก็บเอกสาร WAP

ทั้ง 2 วิธีมีข้อจำกัดแตกต่างกัน คือกรณีของการเข้าไปแก้ไฟล์ httpd.conf ทำได้ก็ต่อเมื่อมีสิทธิในการควบคุม Apache ซึ่งหมายถึงเป็น System Admin. เท่านั้น อย่างไรก็ดีในกรณีที่ติดตั้ง Apache ในเครื่องพีซี สามารถใช้วิธีแก้ไฟล์ httpd.conf ได้เลย แต่ถ้าถูกจำกัดสิทธิ อย่างในกรณีที่เช่าพื้นที่เว็บไซต์ (web hosting service) จะไม่มีสิทธิแตะต้องไฟล์ httpd.conf เลย จึงต้องเลี่ยงมาใช้วิธีที่สอง คือ การเพิ่มไฟล์ .htaccess ลงไปในไดเรกทอรีที่ใช้เก็บเอกสารตระกูล WAP แทน

1) แก้ไขข้อมูลในไฟล์ httpd.conf

การที่ Apache รู้จักไฟล์ HTML, GIF, JPEG เนื่องมาจากมีการกำหนด MIME type ให้รู้จักในไฟล์ httpd.conf ดังนั้นหากต้องการให้ Apache รู้จักไฟล์ในตระกูล WAP ก็ต้องบอกผ่านทาง MIME type เช่นกัน โดยการเพิ่ม MIME type ในไฟล์ httpd.conf

ให้ใช้โปรแกรมเท็กซ์เอดิเตอร์เปิดไฟล์ httpd.conf แล้วหาบรรทัดที่มีคำว่า AddType ซึ่งจะพบบรรทัดที่มีข้อความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

46552

#AddType application/x-httpd-php. .phtml

#AddType application/x-httpd-php-source .phps

AddType application/x-tar .tgz

จากนั้นให้เพิ่มข้อความแทรกลงไป เพื่อกำหนดให้เว็บเซิร์ฟเวอร์รู้จักเอกสารต่างๆ ในตระกูลของ WAP โดยข้อความที่จะเพิ่ม ก็คือ...

AddType text/vnd.wap.wml wml

AddType application/vnd.wap.wmlc wmlc

AddType text/vnd.wap.wmlscript wmls

AddType application/vnd.wap.wmlscriptc wmlsc

AddType image/vnd.wap.wbmp wbmp

เสร็จแล้วจึง save ไฟล์ httpd.conf และ restart โปรแกรม Apache หลังจากนั้นโปรแกรม Apache Web Server รู้จักไฟล์ตระกูล WAP

2) เพื่อไฟล์ htaccess ลงในไดเรกทรอรีที่ใช้จัดเก็บเอกสาร WAP

การเพิ่มไฟล์ .htaccess ทำได้ง่ายๆ คือ ใช้โปรแกรมเท็กซ์เอดิเตอร์อย่าง NotePad สร้างไฟล์นี้ขึ้นมาก่อน โดยพิมพ์เนื้อความลงในไฟล์ดังนี้

DirectoryIndex index.wml

AddType text/vnd.wap.wml wml

AddType application/vnd.wap.wmlc wmlc

AddType text/vnd.wap.wmlscript wmls

AddType application/vnd.wap.wmlscriptc wmlsc

AddType image/vnd.wap.wbmp wbmp

จะเห็นได้ว่ามีลักษณะคล้ายกับข้อความที่เพิ่มในไฟล์ Httpd.conf ต่างกันก็ตรงที่มีบรรทัด DirectoryIndex index.wml เพิ่มขึ้นมา ซึ่งเป็นตัวกำหนดว่าให้ไฟล์ index.wml เป็นไฟล์โฮมเพจของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดเรกทอรีนี้ หมายความว่า หากเว็บไซต์มีชื่อโดเมนเป็น mydomain.com แล้วเมื่อเรียก URL เป็น http://www.mydomain.com/wap/ ก็จะหมายถึง http://www.mydomain.com/wap/index.wml โดยอัตโนมัติ จากนั้นก็เก็บไฟล์นี้ไว้ในไดเรกทอรี ในกรณีที่เป็น Apache ในเครื่องพีซี (ถ้าไม่ได้เลือกใช้วิธีเพิ่ม MIME type ลงในไฟล์ httpd.conf แล้วหันมาเลือกใช้วิธีเพิ่มไฟล์ .htaccess นี้แทน)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ความรู้พื้นฐานเกี่ยวกับ AVR MICROCONTROLLER

3.1 คุณสมบัติและขาต่อใช้งานของ AVR

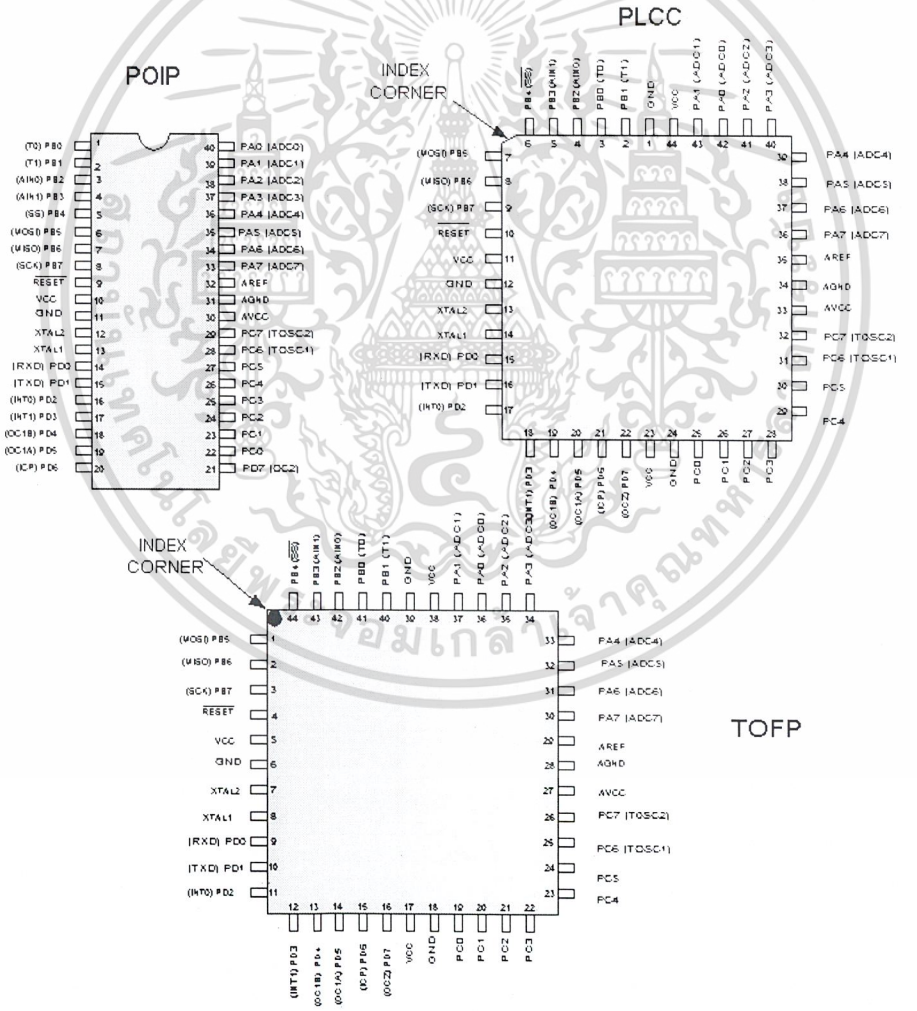
- สถาปัตยกรรมภายในถูกออกแบบให้มีโครงสร้างแบบ RISC
- มีหน่วยคำสั่งในการควบคุมการทำงานจำนวน 188 คำสั่ง
- มีหน่วยความจำแบบ FLASH สำหรับบันทึก PROGRAM MEMORY ขนาด 8 KByte(AT90S/LS8535)
- มีหน่วยความจำแบบ FLASH สำหรับบันทึก PROGRAM MEMORY ขนาด 4KByte(AT90S/LS4434)
- มีหน่วยความจำแบบ EEPROM สำหรับบันทึก DATA MEMORY ขนาด 512 KByte(AT90S/LS8535)
- มีหน่วยความจำแบบ EEPROM สำหรับบันทึก DATA MEMORY ขนาด 256KByte(AT90S/LS4434)
- มีหน่วยความจำแบบ RAM ขนาด 512Byte(AT90S/LS8535)
- มีหน่วยความจำแบบ RAM ขนาด 256Byte(AT90S/LS4434)
- ระบบการเปลี่ยนสัญญาณ A/D ขนาด 10 บิท จำนวน 8 แชนแนล
- กลุ่มรีจิสเตอร์สำหรับใช้งานทั่วไป ขนาด 8บิท จำนวน 32 ตัว
- พอร์ท I/Oขนาด 8 บิท จำนวน 4 พอร์ท
- ระบบการสื่อสารข้อมูลดิจิตอลแบบอะซิงโครนัส (UART) จำนวน 1 CHANNEL
- ระบบการสื่อสารข้อมูลดิจิตอลแบบซิงโครนัส (SPI) จำนวน 1 CHANNEL
- ความถี่สัญญาณนาฬิกา 0-8 MHz(AT90S/LS8535)
- ความถี่สัญญาณนาฬิกา 0-4 MHz(AT90S/LS4434)
- ระบบ POWER ON RESET
- ระบบการกำเนิดความถี่สัญญาณแบบ PWM จำนวน 3 CHANNEL
- ระบบการตรวจจับสัญญาณอนาล็อก(Analog Comparater)
- 3 SLEEP MODE: IDLE,POWER SAVE and POWER DOWN
- ระบบป้องกันการคัดลอกข้อมูลภายในหน่วยความจำ(LOCK FOR SOFTWARE SECURITY)
- ระบบตรวจจับการทำงานผิดพลาดของ CPU(WATCHDOG TIMER WITH ON-CHIP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- OSCILATOR)
- ระบบการ อินเตอร์รัพท์จากภายนอก (EXTERNAL INTERRUPT)
- TIMER/COUNTER ขนาด 16 บิท 1 CHANNEL
- TIMER/COUNTER ขนาด 8 บิท 2 CHANNEL

3.1.1 รายละเอียดของ MCU (Microcontroller)

AT90S4434/AT90S8535 เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิท ที่มีสถาปัตยกรรมแบบ RISC (Reduce Instruction Set Computer) ซึ่งทำให้การประมวลผลมีความเร็ว 1 คำสั่ง /1 clock หรือ CPU สามารถประมวลคำสั่งได้ 1 MIPS/1 MHz



รูปที่ 3.1 โครงสร้างภายนอกและตำแหน่งขาของ AVR

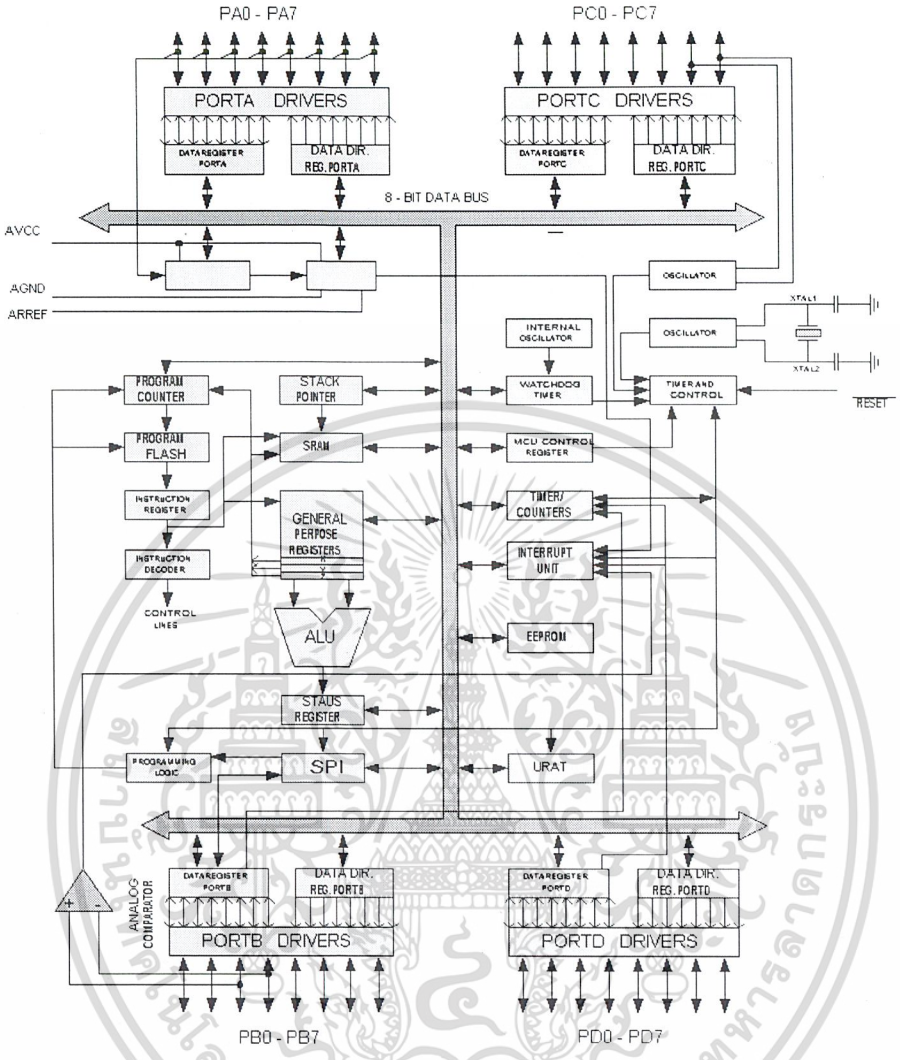
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในประกอบด้วยรีจิสเตอร์ใช้งานทั่วไป ขนาด 8 บิท จำนวน 32 ตัว ซึ่งแต่ละตัวจะต่อเข้ากับหน่วยประมวลผลทางคณิตศาสตร์โดยตรง ทำให้การประมวลผลต่อ 1 คำสั่งมีความเร็วกว่า CPU ที่มีสถาปัตยกรรมแบบ CISC

3.1.2โครงสร้างภายในของ MCU

AT90S8535 จะมีหน่วยความจำสำหรับ PROGRAM MEMORY แบบ FLASH ขนาด 8 KByte หน่วยความจำสำหรับ DATA MEMERY แบบ EEPROMขนาด 512 byte และหน่วยความจำแบบ RAM ขนาด 512 byte มีพอร์ทที่สามารถทำงานได้ 2 ทิศทางจำนวน 32 เส้นสัญญาณ และระบบ TIMER/COUNTER จำนวน 3 ชุด ที่มีโหมดการทำงานเสริมในส่วนของการสร้างสัญญาณ PWM และส่วนของการตรวจจับสัญญาณ INPUT CAPTURE มีอุปกรณ์สื่อสารข้อมูลอนุกรมแบบอะซิงโครนัส (UART) และระบบการสื่อสารข้อมูลดิจิตอลแบบซิงโครนัส (SPI) และยังเสริมด้วยระบบการแปลงสัญญาณอนาลอกเป็นดิจิตอล ขนาด 10 บิท จำนวน 8 ช่องสัญญาณที่มาพร้อมกับ MCU มี Watchdog Timer เพื่อตรวจสอบการทำงานของไมโครคอนโทรลเลอร์ และมีระบบประหยัดพลังงานอีก 3 ระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 บล็อคไดอะแกรมของ MCU

รายละเอียดของขาสัญญาณ

- VCC ขาจ่ายไฟให้กับ CPU

- GND ขากราวด์

- Port A(PA7…PA0) เป็นพอร์ท 2 ทิศทาง ขนาด 8 บิท โดยสามารถกำหนดให้แต่ละขาของพอร์ทสามารถ PULL UP ภายในแยกจากกัน ซึ่งสามารถรับกระแสซิงค์ได้ 20 มิลลิแอมป์ โดยพอร์ท A ยังใช้เป็นขาอินพุทเพื่อรับสัญญาณอนาล็อกในส่วนของการแปลงสัญญาณ ANALOG TO DIGITAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Port B(PB7...PB0) เป็นพอร์ท 2 ทิศทางขนาด 8 บิทโดยสามารถกำหนดให้แต่ละขาของพอร์ทสามารถ PULL UP ภายในอิสระแยกจากกันซึ่งแต่ละขาสามารถรับกระแสซิงค์ได้ 20 มิลลิแอมป์ โดยแต่ละขาสามารถถูกใช้งานในฟังก์ชั่นอื่นๆอีก

- Port C(PC7...PC0) เป็นพอร์ท 2 ทิศทางขนาด 8 บิท โดยสามารถกำหนดให้แต่ละขาของพอร์ทสามารถ PULL UP ภายในอิสระแยกจากกันซึ่งแต่ละขาสามารถรับกระแสซิงค์ได้ 20 มิลลิแอมป์ โดยแต่ละขาสามารถถูกใช้งานในฟังก์ชั่นอื่นๆอีก

- RESET เป็นขารีเซต

- XTAL เป็นขาอินพุตของ OSC

- XTAL2 เป็นขาเอาต์พุตของ OSC

- Avcc ใช้จ่ายไฟให้กับวงจร A/D ที่อยู่ภายใน MCU

- AREF เป็นขาแรงดันอ้างอิงที่ใช้งานในส่วนของ A/D

- AGND เป็นขากราวด์ของวงจร A/D

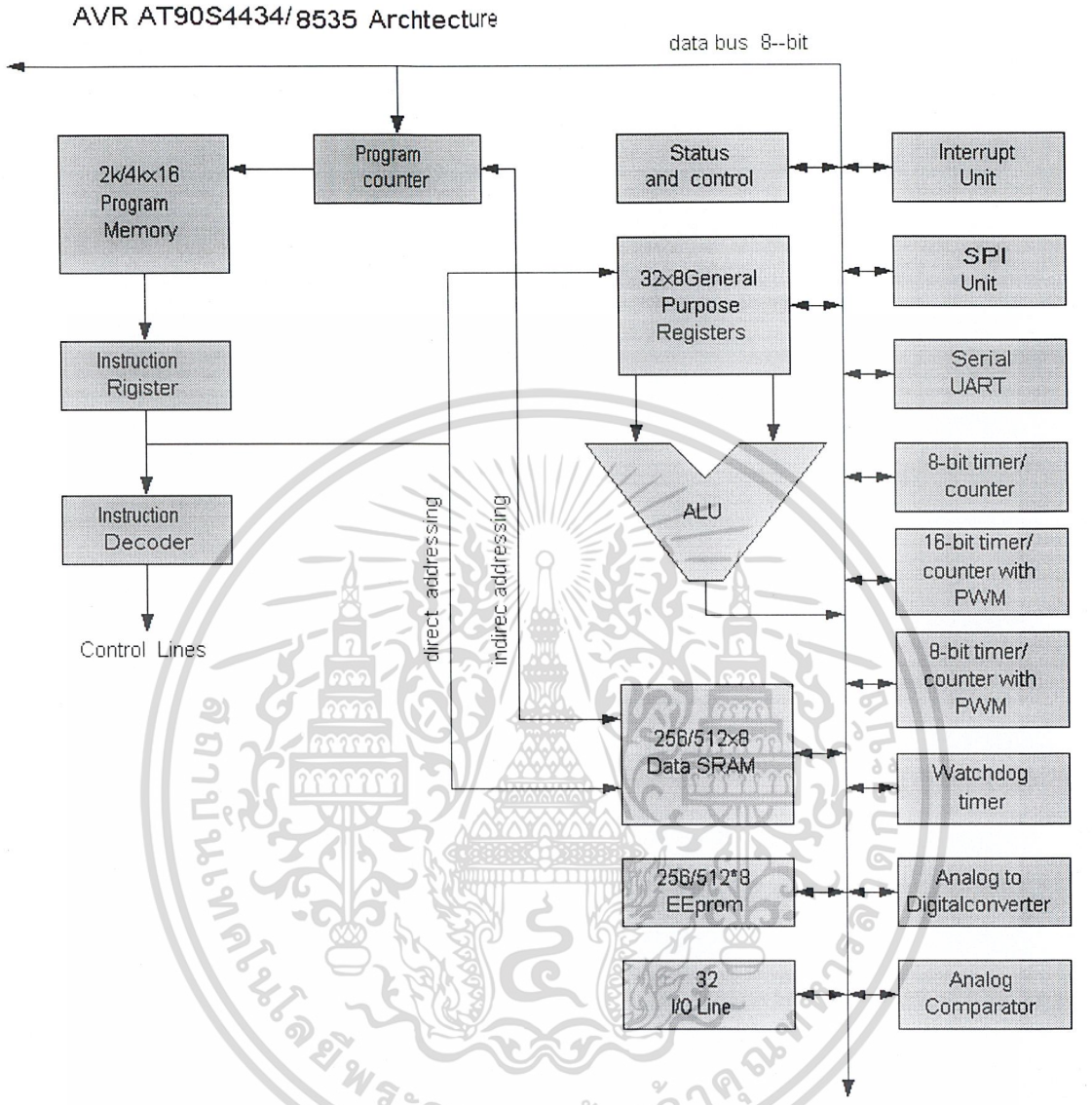
3.2 สถาปัตยกรรมภายในและรีจีสเตอร์ใช้งานทั่วไป

3.2.1 สถาปัตยกรรมภายใน

สถาปัตยกรรมภายในจะประกอบด้วยรีจิสเตอร์ใช้งานทั่วไป 8 บิท จำนวน 32 ตัว ที่สามารถเข้าถึงข้อมูลได้ใน 1 clock ของสัญญาณนาฬิกา

โดยรีจิสเตอร์ R26-31 เป็นรีจิสเตอร์ขนาด 8บิท จำนวน 6ตัวสามารถที่จะจับคู่เพื่อใช้เป็นรีจิสเตอร์ขนาด 16 บิท 3 ตัว โดยใช้ชื่อว่ารีจิสเตอร์ X , Y , Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 สถาปัตยกรรมแบบ RISC ของ AT90S4434/8535

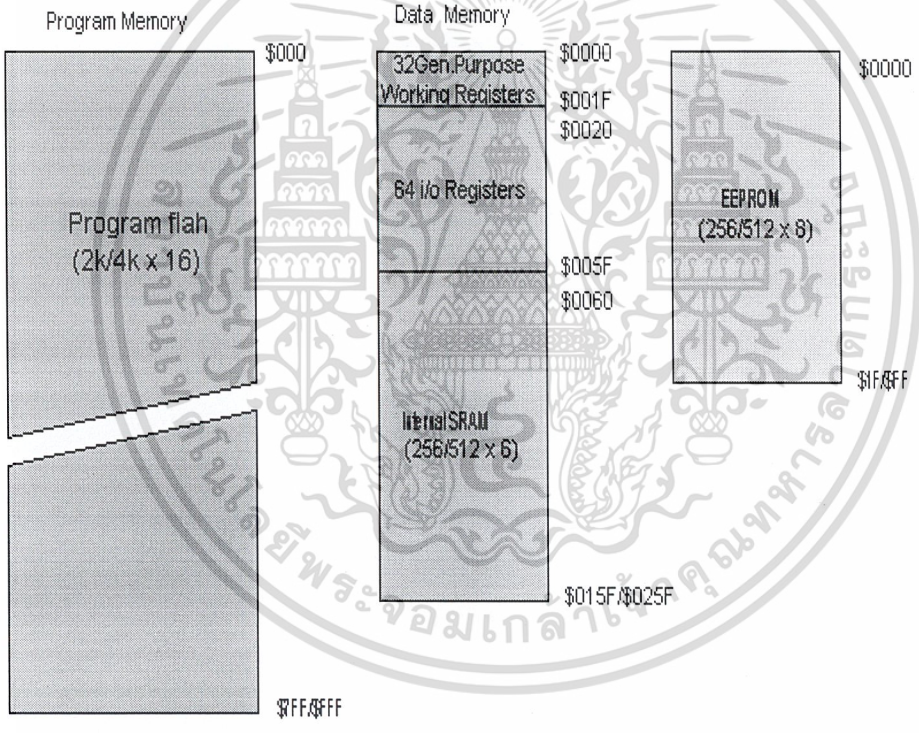
ALU จะสนับสนุนการกระทำทางคณิตศาสตร์และลอจิก ระหว่างรีจิสเตอร์กับรีจิสเตอร์ หรือระหว่างรีจีสเตอร์กับค่าคงที่ ซึ่งการเรียกใช้รีจิสเตอร์ใช้งานทั่วไป สามารถกระทำได้โดยการอ้างถึงหน่วยความจำภายในที่ตำแหน่ง $00 - $1F จำนวน 32 ตำแหน่งและใน MCU ได้จัดแบ่งให้มีรีจิสเตอร์ที่ใช้ควบคุมการทำงานของหน่วยอินพุตและเอาต์พุตต่างๆ อีก 64 ตำแหน่ง โดยสามารถใช้งานได้โดยการอ้างตำแหน่งหน่วยความจำที่ตำแหน่ง $20 - $5F

ระบบการทำงานของไมโครคอนโทรลเลอร์ใช้หลักการออกแบบของ HAVARD ด้วยการแยกระบบบัสของ PROGRAM และ DATA ออกจากกัน โดยโปรแกรมจะมีหน่วยประมวลผลด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SINGLE LEVEL PIPELINING ซึ่งทำให้ CPU สามารถ Fetch และ Executeคำสั่งได้ภายใน 1 คาบเวลา

ด้วยคำสั่ง JUMP และ CALL แบบ RELATIVE ซึ่งสามารกระโดดข้ามการทำงานได้ไกลถึง 2K/4K ซึ่งใน 1 คำสั่งจะใช้รหัสการทำงาน 16 บิท หรือ 1 WORD โดยทุกครั้งที่มีการอินเตอรัพท์ หรือการข้ามไปทำงานโปรแกรมย่อย ค่าของ PROGRAM COUNTER (PC) จะถูกเก็บลงใน STACK ซึ่งจะใช้พื้นที่หน่วยความจำใน SRAM บางส่วนเพื่อทำเป็นพื้นที่ของ STACK

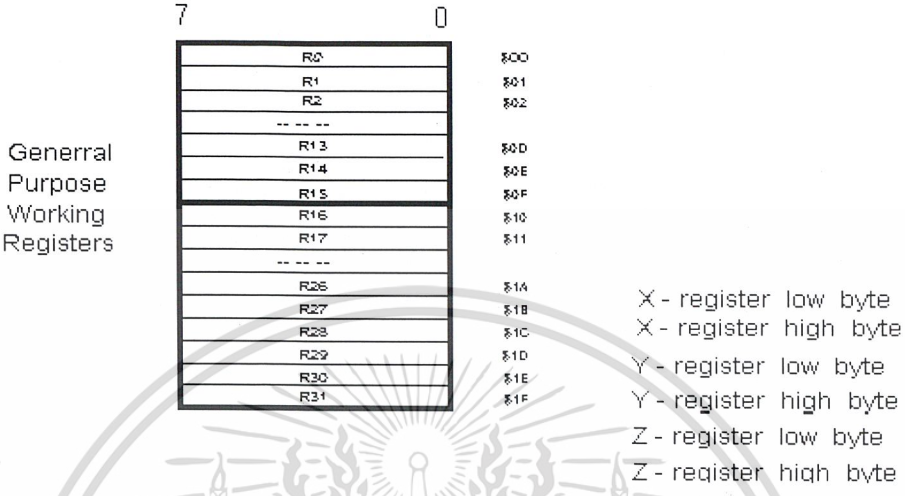


รูปที่ 3.4 โครงสร้างของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 รีจิสเตอร์ใช้งานทั่วไป

โครงสร้างของรีจิสเตอร์ใช้งานทั่วไปทั้ง 32 ตัว แสดงดังรูปที่ 3.5

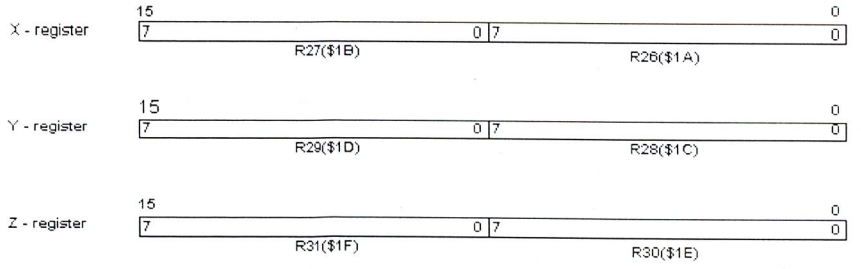


รูปที่ 3.5 โครงสร้างของรีจิสเตอร์ใช้งานทั่วไป

รีจิสเตอร์ทั้งหมดใช้ชุดคำสั่ง เพื่อเข้าถึงโดยตรงและจะใช้ช่วงเวลาการเข้าถึงเพียง 1 CLOCK โดยคำสั่ง SBCI , SUBI, CPI, ANDI และ ORI ซึ่งกระทำระหว่างรีจิสเตอร์กับค่าคงที่ และรีจิสเตอร์กับรีจิสเตอร์ และคำสั่ง LDI ที่ใช้โหลดค่าคงที่เข้าไปในรีจิสเตอร์ และต้องใช้งานกับรีจิสเตอร์ R16-R31 ส่วนคำสั่ง SBC, SUB, CP, AND และ OR และพวกคำสั่งใช้งานอื่นๆ สามารถใช้งานได้ในรีจิสเตอร์ทั่วไป

รีจิสเตอร์ X , รีจิสเตอร์ Y ,รีจิสเตอร์ Z

รีจิสเตอร์ R26 - R31 สามารถนำมาต่อกันเพื่อใช้เป็นรีจิสเตอร์คู่ เพื่อใช้งานเป็นตัวชี้ข้อมูล ในชื่อของรีจิสเตอร์ X ,Y ,Z



รูปที่ 3.6 รีจิสเตอร์ X ,Y และ Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยประมวลผลทางคณิตศาสตร์และลอจิก

ระบบการประมวลผลที่มีประสิทธิภาพของ AVR คือ ALU สามารถสื่อสารได้โดยตรงกับรีจิสเตอร์ใช้งานทั่วไปทั้ง 32 ตัว โดย ALU ได้จัดแบ่งระบบการจัดการข้อมูลไว้ 3 ส่วน คือ ส่วนของการจัดการทางคณิตศาสตร์ ส่วนของการกระทำทางลอจิก และในส่วนของการกระทำทางบิท

หน่วยความจำ SRAM

หน่วยความจำภายใน MCU จัดไว้ 352 ตำแหน่ง สำหรับ สำหรับ AT90S4434 และ 608 ตำแหน่งสำหรับ AT90S8535 โดยหน่วยความจำทั้งหมดถูกแบ่งออกเป็น พื้นที่ของรีจิสเตอร์ใช้งานทั่วไป , รีจิสเตอร์ใช้งาน I /O และหน่วยความจำภายใน SRAM โดย 96 ตำแหน่งแรกจะถูกแบ่งออกเป็นส่วนของรีจิสเตอร์ และอีก 256 ตำแหน่ง สำหรับ AT90S4434 และ 512 ตำแหน่ง สำหรับ AT90S8535 ถูกจัดไว้เป็นส่วนของหน่วยความจำภายใน SRAM

การเข้าถึงข้อมูลแบ่งออกเป็น 4 ส่วน คือ DIRECT , INDIRECT WITH DISPLACEMENT, INDIRECT WITH PREDISPLACEMENT และ INDIRECT WITH POST-INCREMENT

รีจิสเตอร์สถานะ SREG(The Status Register-SREG)

รีจิสเตอร์ SREG ถูกวางที่ตำแหน่ง $3F($5F)

	7	6	5	4	3	2	1	0
	I	T	H	S	V	N	Z	C
Read/Writ	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0

รูปที่ 3.7 รีจิสเตอร์ของ SREG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 7- I: Grobal Interrupt Enable

บิท I จะต้องถูกเซ็ตเป็น 1 เมื่อต้องการกำหนดให้ INTERRUPT ทั้งหมดทำงาน โดยที่ INTERRUPT แต่ละแหล่งจะสามารถแบ่งแยกกันในรีจิสเตอร์ GIMSK และ TIMSK แต่ถ้าบิท I ถูกเคลียร์เป็น 0 INTERRUPT ทุกตัวจะถูก DISABLE โดยบิท I จะถูกเคลียร์โดยอัตโนมัติเมื่อเกิดการ INTERRUPT และจะถูกเซ็ตโดยการใช้คำสั่ง RETI

Bit 6- T: Bit Copy Storage

การใช้คำสั่ง BLD และคำสั่ง BST จะเป็นคำสั่งที่ใช้บิท T ทำหน้าที่เป็น SOURCE หรือ DESTINATION โดยบิทต่างๆในรีจิสเตอร์สามารถ COPY ลงในบิท T ได้โดยการใช้คำสั่ง BLD และสามารถคัดลอกลงในบิทของรีจิสเตอร์ได้โดยใช้คำสั่ง BST

Bit 5- H : Half Carry Flag

บิท H แสดงการเกิด HALF CARRY FLAG

Bit 4- S:Sign Bit, S=N+V

บิทที่ใช้แสดงเครื่องหมาย

Bit 3- V: Two's Complement Overflow Flag

แสดงการทำ Two's Complement

Bit 2 - N :Negative Flag

จะแสดงการทำคำสั่งที่เป็นลบของการกระทำทางคณิตศาสตร์ และลอจิก

Bit 1- Z: Zero Flag

แสดงการกระทำคำสั่งทางคณิตศาสตร์และลอจิกที่ให้ผลเป็น 0

Bit 0-C :Carry Flag

แสดงการเกิด Carry Flag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

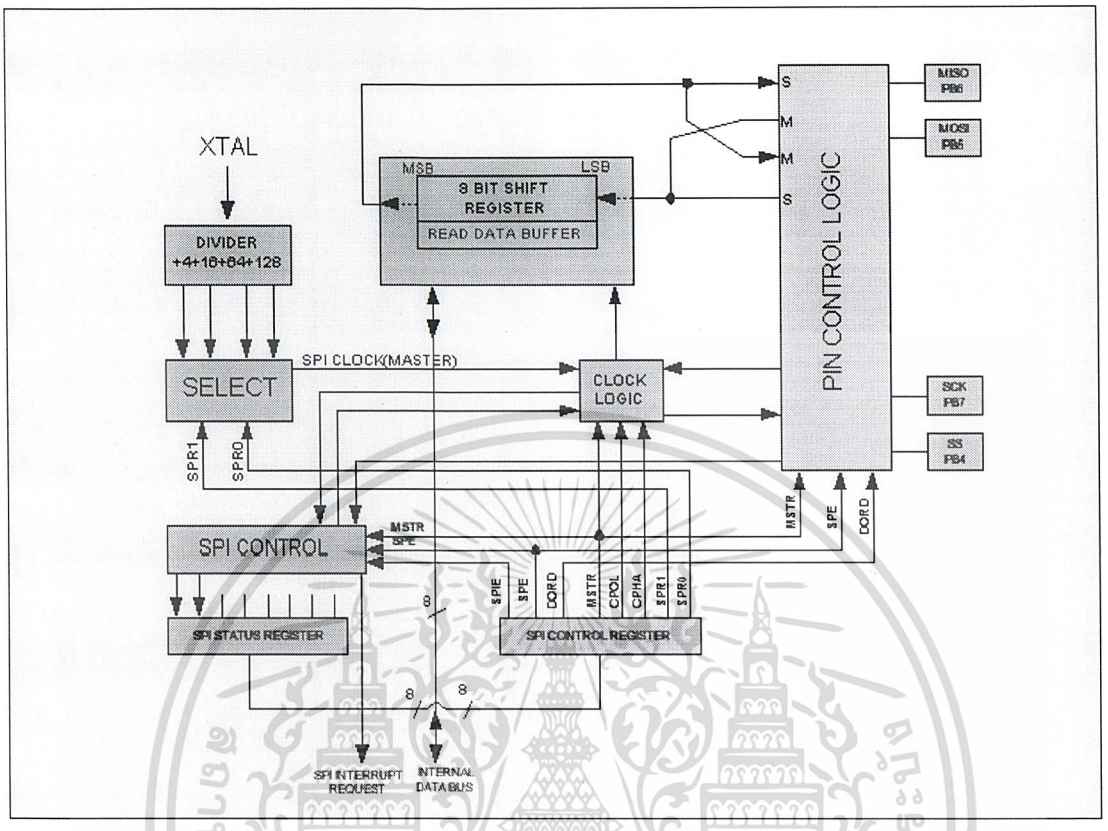
3.3 การสื่อสารแบบอนุกรมของ AVR

การสื่อสารอนุกรมแบบซิงโครนัส (Serial Peripheral Interface) ในไมโครคอนโทรลเลอร์ AT90S4434/AT90S8535 ได้จัดให้มีการสื่อสารแบบ SPI ซึ่งเป็นการสื่อสารแบบซิงโครนัสที่มีความเร็วสูง ซึ่งมีคุณสมบัติดังนี้

1. เป็นการสื่อสารแบบ FULL DUPLEX ซึ่งใช้สายสัญญาณ 3 เส้น ในการส่งถ่ายข้อมูล
2. สามารถทำงานได้เป็นทั้งตัวส่งข้อมูลและตัวรับข้อมูล
3. สามารถเลือกได้ว่าจะส่งให้บิท MSB หรือ LSB ออกก่อน
4. มี FLAG แสดงสถานะการรับส่งข้อมูลสิ้นสุด

การสื่อสารข้อมูลระหว่าง MCU ที่เป็น MASTER กับ MCU ที่เป็น SLAVE ด้วยการสื่อสารแบบ SPI โดยขา PB7(SCK) ซึ่งเป็นขา Clock จะถูกใช้งานเป็นขา OUTPUT เมื่อ MCU เป็น MASTER และถูกใช้งานเป็นขา INPUT เมื่อ MCU เป็น SLAVE โดยทุกครั้งที่เขียนข้อมูลลงใน SPI Data Register ของ MCU ที่เป็น MASTER และกำเนิด Clock ที่ขา SCK และข้อมูลจะถูกเลื่อนจากขา MOSI ของ MPU ที่เป็นMASTER เข้าสู่ขา MOSI ของตัวที่เป็น SLAVE และเมื่อมีการส่งข้อมูลครบ 1Byte สัญญาณ Clock จะหยุดและบิท SPIF จะถูกเซ็ต ซึ่งถ้าบิท SPIE ในรีจิสเตอร์ SPCR ถูกเซ็ตเป็น 1 จะทำให้เกิดอินเตอร์รัพท์

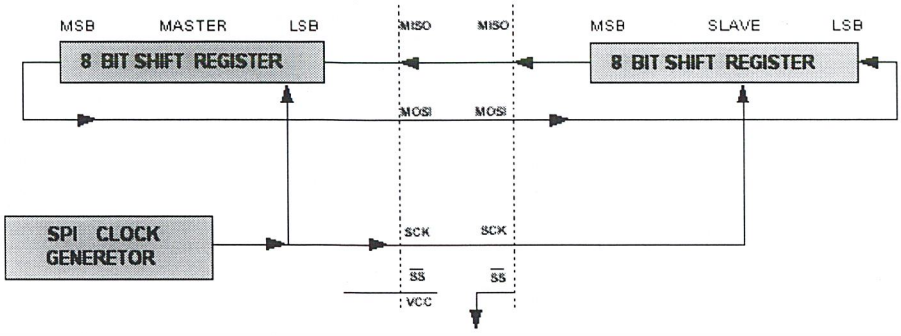
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 โครงสร้างของ SPI

ถ้าขา PB4(SS) มีสถานะเป็น LOW จะเป็นการกำหนดให้ MCU ทำหน้าที่เป็น SLAVE โดยเมื่อพิจารณาถึง SHIFT REGISTER ที่ทำหน้าที่เลื่อนข้อมูลใน MCU ที่ทำหน้าที่เป็น MASTER และ SLAVE สามารถพิจารณาให้เป็น Circular Shift Register ขนาด 16บิท ซึ่งแสดงดังรูปที่ 3.9 เมื่อต้องการส่งข้อมูลออกจาก MCU ที่เป็น MASTER เข้าไปใน MCU ที่เป็น SLAVE ข้อมูลจะถูกส่งถ่ายระหว่าง MCU ทั้ง 2 ตัวในทิศทางที่ตรงกันข้าม ซึ่ง 1 รอบของการส่งข้อมูลจะทำให้ข้อมูลใน MASTER และ SLAVE ถูกแลกเปลี่ยนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 SPI Master – Slave Interconnection

ในด้านส่งข้อมูลจะมี BUFFER เพียงชุดเดียว แต่ในส่วนของด้านรับจะมี BUFFER 2 ชุด เพราะฉะนั้นการส่งข้อมูลจะต้องรอให้ข้อมูลใน BUFFER ถูกส่งออกไปทางด้านรับเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลชุดใหม่ลงใน BUFFER ได้ ส่วนในการรับข้อมูลจะต้องอ่านข้อมูลออกจาก SPI Data Register ก่อนที่ข้อมูลชุดใหม่จะถูกส่งเข้ามา เพื่อไม่ให้ข้อมูลสูญหาย

สามารถกำหนดทิศทางของ MOSI, MISO, SCK และ SS จะถูกกำหนดตามตารางที่ 3.1

ตารางที่ 3.1 SPI Pin Overrides

Pin	Direction, Master SPI	Direction, Slave SPI
MOSI	User Defined	Input
MISO	Input	User Defined
SCK	User Defined	Input
SS	User Defined	Input

SS Pin Function

เมื่อ MCU ถูกกำหนดให้เป็น MASTER (โดยการเซ็ตบิท MSTR ในรีจิสเตอร์ SPCR) ซึ่งถ้า SS ถูกกำหนดให้เป็น เอาต์พุต (ซึ่งปกติจะเป็นเอาต์พุต) จะทำให้ MCU เป็น MASTER และพร้อมที่จะส่งข้อมูลออก แต่ถ้า SS ถูกกำหนดให้เป็นขาอินพุต จะต้องป้อนโลจิก HIGH ให้กับ SS เพื่อให้แน่ใจว่า MCU จะทำหน้าที่เป็น MASTER และพร้อมที่จะส่งข้อมูล แต่ถ้า SS ถูกกำหนดเป็น อินพุตและมีสถานะเป็น LOW จะทำให้ MCU ถูกควบคุมให้เป็น SLAVE ซึ่งทำให้ระบบ SPI ถูกเปลี่ยนแปลงดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

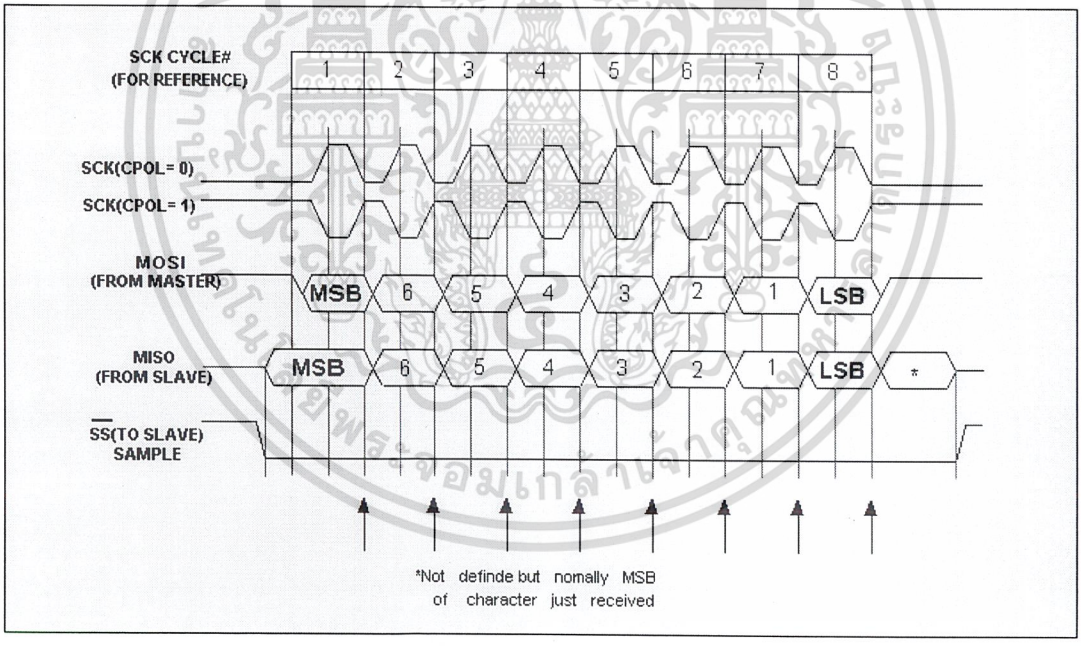
1. บิท MSTR ในรีจิสเตอร์ SPCR จะถูกเคลียร์ และ MCU จะกระทำตัวเป็น SLAVE ซึ่งทำให้ขา MOSI และ SCK เป็น อินพุต

2. บิท SPIF ในรีจิสเตอร์ SPSR ถูกเซ็ต และถ้าอินเตอร์รัพท์ของ SPI ได้ถูกกำหนดให้ Enable จะทำให้เกิดการ INTERRUPT ขึ้น

เมื่อ MCU ถูกกำหนดให้เป็น SLAVE (ขา SS เป็นอินพุต) และถูกป้อนลอจิก LOW จะทำให้ MCU พร้อมที่จะรับข้อมูล แต่ถ้าขา SS ถูกป้อนลอจิก HIGH จะทำให้MCU ไม่สามารถรับข้อมูลได้

DATA MODES

มีวิธีการ 4 อย่าง ที่จะกำหนดลักษณะสัญญาณที่ขา SCK เพื่อใช้ในการสื่อสารข้อมูลแบบ SPI โดยสามารถกำหนดได้จากบิท CPHA และบิท CPOL ซึ่งแสดงในรูปที่ 3.10



รูปที่ 3.10 SPI Transfer Format with CPHA

การสื่อสารข้อมูลแบบอซิงโครนัส (UART)

ไมโครคอนโทรลเลอร์ AT90S4434/8535 ถูกจัดให้มีฟังก์ชั่นในส่วนของการสื่อสารในแบบอะซิงโครนัส หรือ UART (Asynchronous Receiver and Transmitter)โดยจัดให้มีคุณสมบัติหลักๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. สามารถเปลี่ยนแปลง BOUD RATE ของการสื่อสารได้หลาย BOUD RATE
2. สามารถสื่อสารได้ในอัตรา BOUD RATE ที่สูงในขณะที่ความถี่ XTAL ต่ำ
3. สื่อสารข้อมูลได้ทั้ง 8 บิท และ 9 บิท
4. มีส่วนของการกำจัดสัญญาณรบกวน (NOISE FILTERING) ที่เกิดขึ้นในสายส่ง
5. ตรวจจับความผิดพลาดของการสื่อสารข้อมูล ซึ่งเกิดจากการที่ข้อมูลถูกส่งเข้ามาก่อนยังไม่ได้ถูกอ่านออก จากบัฟเฟอร์และมีข้อมูลใหม่เข้ามาทับ
6. ตรวจจับความผิดพลาดของกรอบข้อมูลผิดพลาด
7. ตรวจจับความผิดพลาดของบิทเริ่มต้น (False Start Bit Detection)
8. จัดให้มีการแยกระบบอินเตอรัพท์ของการสื่อสารออกเป็น 3 ส่วน คือ อินเตอรัพท์ซึ่งเกิดจากการส่งข้อมูล อินเตอรัพท์ซึ่งเกิดจากการรับข้อมูล และอินเตอรัพท์ซึ่งเกิดจากการไม่มีข้อมูลในรีจิสเตอร์
9. มี BUFFER ในการเก็บข้อมูลที่จะส่งออกและข้อมูลที่รับ

การควบคุม UART

The UART I/O Data Register- UDR

	7	6	5	4	3	2	1	0	
	MSB							LSB	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**UDR**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.11 รีจิสเตอร์ของ UDR

โครงสร้างถูกแยกเป็น 2 ส่วนแต่ใช้ตำแหน่งที่อยู่เดียวกัน ถ้าเป็นการเขียนข้อมูลลงในรีจีสเตอร์จะเป็นรีจีสเตอร์ที่ใช้ในการเก็บข้อมูลที่จะส่ง แต่ถ้าเป็นการอ่านข้อมูลจากรีจีสเตอร์ จะเป็นการอ่านข้อมูลที่รับได้จาก UART

The UART Status Register-USR

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	OR	-	-		
Read/Write	R	R/W	R	R	R	R	R	R	**USR**
Initial value	0	0	1	0	0	0	0	0	

รูปที่ 3.12 รีจิสเตอร์ของ USR

เป็นรีจีสเตอร์ที่สามารถอ่านได้อย่างเดียว โดยข้อมูลภายในจะแสดงสถานะต่างๆของUART

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Bit 7**-RXC : UART Receive complete

บิทนี้จะเซ็ตเป็น 1 เมื่อมีการรับข้อมูลจากพอร์ตอนุกรมและข้อมูลเก็บลงในรีจีสเตอร์ UDR เสร็จเรียบร้อยซึ่งบิท RXC จะเซ็ตโดยไม่คำนึงถึงการตรวจสอบความผิดพลาดของกรอบข้อมูลว่าเกิดความผิดพลาดหรือไม่โดยเมื่อบิท RXCIE ในรีจีสเตอร์ UCR และบิท RXC เซ็ตเป็น 1 จะทำให้เกิดสัญญาณอินเตอร์รัพท์เกิดขึ้น โดยบิท RXC จะถูกเคลียร์ให้เป็น 0 เมื่อค่าในรีจีสเตอร์ UDR ถูกอ่าน

- **Bit 6**- TXC :UART Transmit Complete

บิทนี้จะเซ็ตเป็น 1 เมื่อข้อมูลถูกส่งออก (รวมบิท STOP) ทางพอร์ตทั้งหมดและไม่มีข้อมูลชุดใหม่เขียนลงในรีจีสเตอร์ UDR ซึ่งสถานะของบิทนี้ จะมีประโยชน์มากในการส่งข้อมูลอนุกรมแบบ HALF-DUPLEX

เมื่อบิท TXCIE ในรีจีสเตอร์ UCR และบิท TXC เซ็ตเป็น 1 จะทำให้เกิดอินเตอร์รัพท์ ซึ่งบิท TXC สามารถเคลียร์ให้เป็น 0 ได้โดยการเขียนลอจิก 1 ลงในบิท TXC หรือเคลียร์โดยอัตโนมัติเมื่อ CPU ทำโปรแกรมบริการอินเตอร์รัพท์

- **Bit 5**- UDRE : UART Data Register Empty

บิทนี้จะถูกเซ็ตเป็น 1 เมื่อข้อมูลจาก Data register ถูกเขียนลงใน Shift Register ซึ่งหมายถึงไม่มีข้อมูลใน data register เพื่อแสดงให้ผู้เขียนข้อมูลชุดใหม่ลงใน Data registerเพื่อส่งข้อมูลชุดต่อไป

เมื่อบิท UDRIE และบิท UDRE เซ็ตเป็น 1 จะทำให้เกิดสัญญาณอินเตอร์รัพท์ โดยบิท UDRE จะเคลียร์เป็น 0 เมื่อเขียนข้อมูลลงในรีจีสเตอร์ UDR

- **Bit 4** –FE Framing Error

บิทนี้จะถูกเซ็ตเป็น 1 เมื่อมีการตรวจพบว่ากรอบข้อมูลผิดพลาด

- **Bit 3**- OR:Overrun

บิทนี้จะถูกเซ็ตเมื่อตรวจสอบได้ว่าเกิด Overrun คือการที่ข้อมูลชุดใหม่เข้ามาทับข้อมูลเดิม

- Bit 2..0-Res:Reseved bits

บิท 0-บิท 2 ถูกสงวนไว้โดยปกติจะเป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The UART Control Register- UCR

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIe	RXEN	TXEN	CHR9	RXB8	TXB8	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	W	**UCR**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.13 รีจิสเตอร์ของ UCR

- **Bit 7** –RXCIE:RX Complete Interrupt Enable

เป็นบิทที่ใช้ Enable อินเตอร์รัพท์ ของการรับข้อมูลคือ เมื่อการรับข้อมูลไม่มีความผิดพลาดจะทำให้ บิท RXC เซ็ตเป็น 1 จะทำเกิดอินเตอร์รัพท์ถ้าบิท RXCIE และบิท I เซ็ตเป็น 1

- **Bit 6** –TXCIE:TX Complete Interrupt Enable

เป็นบิทที่ใช้ Enable อินเตอร์รัพท์ของการส่งข้อมูล คือเมื่อมีการส่งข้อมูลครบทุกบิทจะทำให้บิท TXC เซ็ตเป็น 1 จะทำเกิดอินเตอร์รัพท์ถ้าบิท TXCIE และบิท I เซ็ตเป็น 1

- **Bit 5**- UDRIE: UART Data Register Empty Interrupt Enable

ถ้าไม่มีข้อมูลในรีจีสเตอร์ UDR จะทำให้บิทUDRE เซ็ตเป็น 1 จะทำเกิดอินเตอร์รัพท์ถ้าบิท UDRIE และบิท I เซ็ตเป็น 1

- **Bit 4** –RXEN: Receive Enable

ใช้กำหนดให้ UART เริ่มรับข้อมูล

- **Bit 3** –TXEN:Transmitter Enable

ใช้กำหนดให้ UART เริ่มส่งข้อมูล

- **Bit 2**- CHR9: Bit characters

เป็นบิทที่กำหนดการรับส่งข้อมูลขนาด 9 บิท ถ้าบิทนี้เซ็ตเป็น 1 จะมีการกำหนดให้มีการรับส่งข้อมูลขนาด 9 บิทแต่ถ้าบิท CHR 9 ถูกเคลียร์เป็น 0 จะเป็นการรับส่งข้อมูลขนาด 8 บิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Bit 1**- RXB8 : Recive Databit 8

บิท RXB8 จะทำหน้าที่ในการเก็บค่าบิทที่ 9 ของการรับข้อมูลขนาด 9 บิท

- **Bit 0**- TXB8 : Transmit Databit8

บิท TXB8 จะทำหน้าที่ในการเก็บค่าบิทที่ 9 ของการส่งข้อมูลขนาด 9 บิท

การกำหนด BAUD RATE

$$BAUD = \frac{fck}{16\,(UBRR+1)}$$

-BAUD = Baud-Rate

-fck = Crystal Clock frequency

-UBRR = Contents of UART Baud Rate register

-UBRR = (0-255)

รีจีสเตอร์ UBRR (UART Baud Rate Register)

เป็นรีจีสเตอร์ขนาด 8 บิท ที่ใช้ในการกำหนดอัตรา Baud Rate

	7	6	5	4	3	2	1	0	
	MSB							LSB	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**UBRR**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.14 รีจิสเตอร์ของ UBRR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 พอร์ตอินพุตและเอาท์พุต

พอร์ต A

เป็นพอร์ต 2 ทิศทางขนาด 8 บิทโดยมีหน่วยควบคุมการทำงานของพอร์ตต่างๆคือ รีจีสเตอร์ PORTA (DATA REGISTER) อยู่ที่ตำแหน่ง $1B($3B), รีจีสเตอร์ DDRA (DATA DIRECTION REISTER) อยู่ที่ตำแหน่ง $1A($3A) และ PINA (PORTA INPUT PINS) อยู่ที่ตำแหน่ง $19($39) โดย PIN A จะสามารถอ่านได้อย่างเดียว ไม่สามารถเขียนข้อมูลลงได้ ในขณะที่ PORTA และ DDRA จะสามารถอ่านและเขียนได้ โดยแต่ละขาสัญญาณของพอร์ต A สามารถกำหนดให้ความต้านทาน PULL UP ทำงานในสภาวะ ACTIVE หรือไม่ ACTIVE ก็ได้ โดยแต่ละขาสัญญาณของพอร์ต A จะสามารถรับกระแส (SINK CURRENT) ได้ 20 mA ซึ่งถ้าขาของพอร์ต A ใช้งานเป็น อินพุตและภายนอกมีความต้านทาน PULL –LOW ในขณะที่ความต้านทาน PULL UP ภายใน ACTIVE จะทำให้ MCU จ่ายกระแสออกมาภายนอก

พอร์ต A จะถูกนำไปใช้งานอีกอย่างหนึ่งคือ เป็นขาสัญญาณอินพุตของวงจร ANALOG TO DIGITAL โดยถ้าบางขาของพอร์ต A ถูกใช้งานเป็นขาเอาท์พุตของพอร์ต ในขณะที่บางขาของพอร์ต A ถูกใช้งานเป็นขาเอาท์พุตของ ANALOG TO DIGITAL ผู้ใช้ต้องไม่เปลี่ยนแปลงสภาวะการทำงานของพอร์ต A ในขณะที่ วงจร ANALOG TO DIGITAL กำลัง convertion สัญญาณซึ่งอาจจะทำให้การทำการแปลงสัญญาณถูกขัดจังหวะ จะมีผลให้การแปลงสัญญาณผิดพลาด

การทำงานในโหมดประหยัดพลังงานจะทำให้ SCHMITT TRIGGERของส่วน อินพุตถูกตัดออก

The Port A Data Register-PORT A

Bit $1B	7	6	5	4	3	2	1	0	
	PORT A7	PORT A6	PORT A5	PORT A4	PORT A3	PORT A2	PORT 1	PORT A0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**PORTA**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.15 รีจิสเตอร์ของ PORTA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Port A Data Direction Register-DDRA

Bit$1A	7	6	5	4	3	2	1	0	
	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**DDRA**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.16 รีจิสเตอร์ของ DDRA

The Port A Input Pin Address-PINA

Bit$19	7	6	5	4	3	2	1	0	
	PINA7	PINA6	PINA5	PINA4	PINA3	PINA2	PINA1	PINA0	
Read/Write	R	R	R	R	R	R	R	R	**PINA**
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

รูปที่ 3.17 รีจิสเตอร์ของ PINA

PINA ไม่ใช่รีจีสเตอร์ ซึ่งตำแหน่งที่กำหนดจะเป็นตำแหน่งของแต่ละขาของพอร์ต โดยเมื่ออ่าน PORTA จะเป็นการอ่านค่าที่ LATCH ส่วนการอ่านค่า PINA จะเป็นการอ่านค่าจริงๆของขาพอร์ต

การใช้งานพอร์ต A เป็นขาอินพุตเอาท์พุต

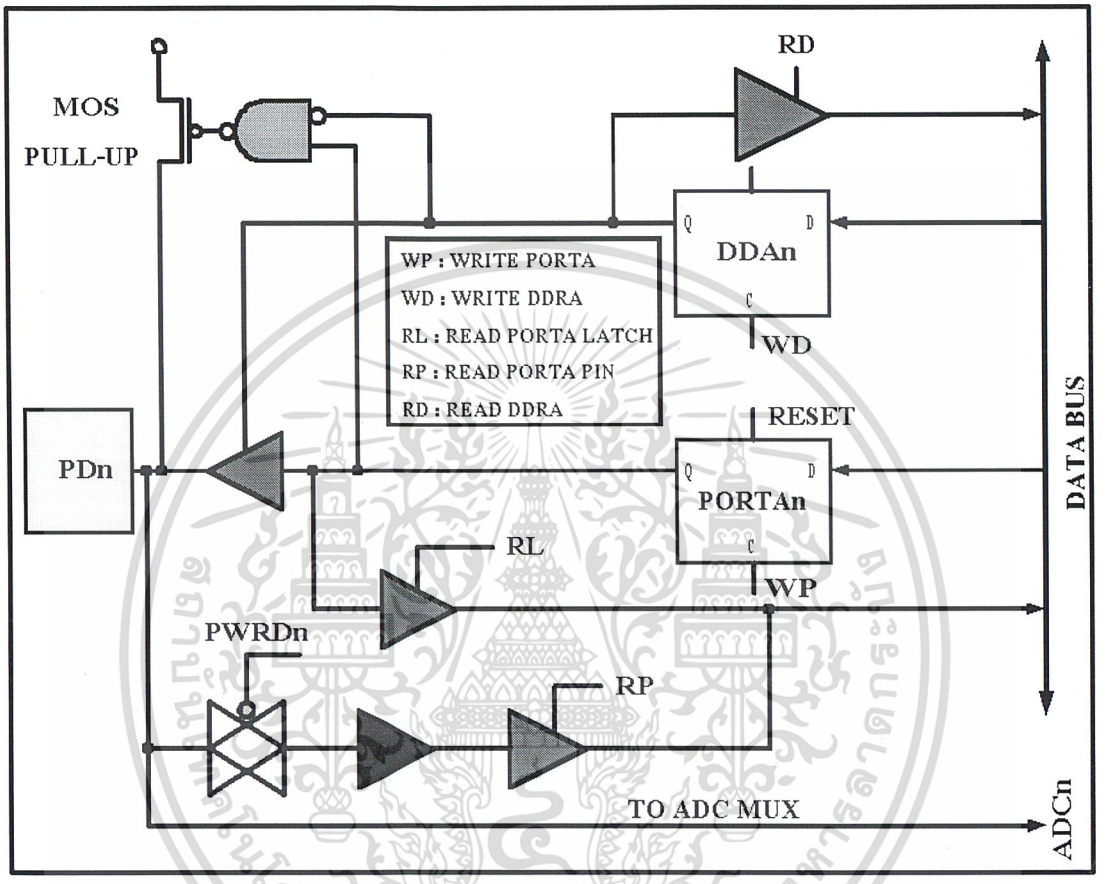
บิท DDAn ในรีจีสเตอร์ DDRA จะเป็นบิทที่ใช้ในการกำหนดทิศทางของแต่ละพอร์ต ถ้าบิท DDAn ถูกเซ็ตเป็น 1 จะทำให้ขาของพอร์ตนั้นเป็นเอาท์พุต แต่เมื่อ DDAn ถูกเคลียเป็น 0 จะทำให้ขานั้นถูกกำหนดเป็นอินพุต ซึ่งสภาวะของขาต่างๆแสดงดังตารางที่ 3.2

ตารางที่ 3.2 DDAn Effects on PORTA Pins

DDAn	PORTAn	I/O	Pull Up	Comment
0	0	Input	NO	Tri-state (Hi-Z)
0	1	Input	YES	PAn will source current if ext.pulled low
1	0	Output	NO	Push –pull Zero Output
1	1	Output	No	Push –pull One Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของพอร์ต A



รูปที่ 3.18 ลักษณะโครงสร้างขาของพอร์ต A (Pins PA0 – PA7)

P: WRITE PORTA

WD: WRITE DDRA

RL: READ PORTA LATCH

RP : READ PORTA PIN

RD: READ DDRA

N: 0-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต B

พอร์ต B เป็นพอร์ตสองทิศทางขนาด 8 บิท โดยมีรีจีสเตอร์ที่ใช้ควบคุมพอร์ต B อยู่ 3 ตัว คือรีจีสเตอร์ PORTB อยู่ที่ตำแหน่ง $18($38) รีจีสเตอร์ DDRB อยู่ที่ตำแหน่ง $17(&37(และ PINB อยู่ที่ตำแหน่ง $16($36) โดยพอร์ต A แต่ละขาสามารถแยกกำหนดให้มีความต้านทาน Pull Up ได้ตามต้องการ ซึ่งแต่ละขาสามารถรับกระแส (SINK CURRENT) ได้ 20 mA โดยขาของพอร์ต Bสามารถใช้งานเป็นฟังก์ชั่นอื่นๆได้ดังตารางที่ 3.3

ตารางที่ 3.3 การใช้งานเป็นฟังก์ชั่นอื่น ๆ ของพอร์ต B

Port Pin	Afternate Function
PB0	T0 (Timer / Counter 0 external counter input)
PB1	T1(Timer / Counter 1external counter input)
PB2	AIN0(Analog comparator positive input)
PB3	AIN1Analog comparator negative input)
PB4	SS (SPI Slave Selectinput)
PB5	MOSI (SPI Bus Master Output /Slave input)
PB6	MISO (SPI Bus Master input/Slave Output)
PB7	SCK (SPI Bus Serial Clock)

The Port B Data Register –PORTB

Bit $1B	7	6	5	4	3	2	1	0	
	PORT B7	PORT B6	PORT B5	PORT B4	PORT B3	PORT B2	PORT 1	PORT B0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**PORTB**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.19 รีจิสเตอร์ของ PORTB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Port BData Direction Register-DDRB

Bit$1A	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**DDRB**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.20 รีจิสเตอร์ของ DDRB

The Port BInput Pin Address-PINB

Bit$19	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	
Read/Write	R	R	R	R	R	R	R	R	**PINB**
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

รูปที่ 3.21 รีจิสเตอร์ของ PINB

การใช้งานพอร์ต B เป็นพอร์ตอินพุตและเอาท์พุต

ตารางที่ 3.4 การใช้งานพอร์ต B เป็นอินพุตและเอาท์พุต

DDBn	PORTBn	I/O	Pull Up	Comment
0	0	Input	NO	Tri-state (Hi-Z)
0	1	Input	YES	PBn will source current if ext.pulled low
1	0	output	NO	Push –pull Zero Output
1	1	output	No	Push –pull One Output

ฟังชั่นอื่นๆที่ใช้สัญญาณของ PORTB

- SCK – PORT B,Bit7

เป็นขาClock ที่ใช้งานในส่วนการสื่อสารแบบ SPI

- MISO-PORTB,Bit6

เป็นขาการรับข้อมูลของการสื่อสารแบบ SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-MOSI-PORTB,Bit 5

เป็นขาส่งข้อมูลในของการสื่อสารแบบ SPI

- SS-PORTB,Bit 4

ใช้เป็นขาควบคุมการทำงานในโหมด SPI

- AIN1-PORTB ,Bit 3

เป็นขาอินพุต 1 ของการทำงานในส่วนของ Analog Compare

- AIN0-PORTB,Bit 2

เป็นขาอินพุต 0 ของการทำงานในส่วนของ Analog Compare

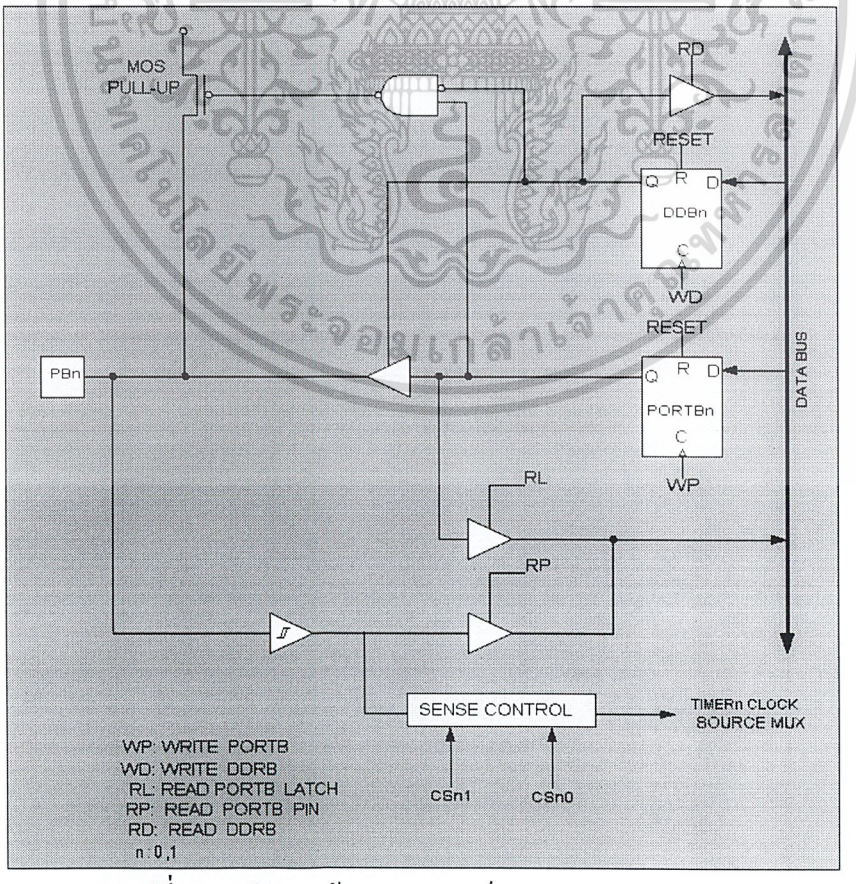
- T1-PORTB,Bit 1

เป็นขาอินพุตที่ใช้ในส่วนของ Timer 1

- T0-PORTB,Bit 0

เป็นขาอินพุตที่ใช้ในส่วนของ Timer 0

โครงสร้างของพอร์ต B



รูปที่ 3.22 โครงสร้างขาของพอร์ต B (Pins PB0 และ PB1)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ต C

พอร์ต C เป็นพอร์ต 2 ทิศทางขนาด 8 บิท โดยมีหน่วยควบคุมการทำงานของพอร์ตจำนวน 3 หน่วยคือ PORTC (Data Register- PORTC) อยู่ที่ตำแหน่งหน่วยความจำ $15($35), DDRC (Data Direction Register) อยู่ที่ตำแหน่งหน่วย $14($34) และ PINC (Port C Input Pin-PINC) อยู่ที่ตำแหน่ง $13($33) โดย PIN C จะสามารถอ่านได้อย่างเดียว ในขณะที่ PORTC แต่และDDRC จะสามารถทั้งอ่านและเขียนได้โดยแต่ละขาของพอร์ต C สามารถแยกกำหนดความต้านทาน PULL UP ได้ในขณะที่พอร์ต C แต่ละขาสามารถรับกระแส (SINK CURRENT) ได้ 20 mA โดยถ้าภายในกำหนดให้มีความต้านทาน PULL UP และภายนอกมีความต้านทาน PULL LOW จะทำให้ MCU จ่ายกระภายนอก

The Port C Data Register –PORTC

Bit $1B	7	6	5	4	3	2	1	0	
	PORT C7	PORT C6	PORT C5	PORT C4	PORT C5	PORT C2	PORT 1	PORT C0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**PORTC**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.23 รีจิสเตอร์ของ PORTC

The Port C Data Direction Register -DDRC

Bit$1A	7	6	5	4	3	2	1	0	
	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**DDRC**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.24 รีจิสเตอร์ของ DDRC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Port C Input Pin Address-PINC

Bit$19	7	6	5	4	3	2	1	0	
	PINC7	PINC6	PINC5	PINC4	PINC3	PINC2	PINC1	PINC0	
Read/Write	R	R	R	R	R	R	R	R	**PINC**
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

รูปที่ 3.25 รีจิสเตอร์ของ PINC

PINC ไม่ใช่ REGISTER ดังนั้นตำแหน่งที่กำหนดจะเป็นตำแหน่งของแต่ละขาของพอร์ต C โดยถ้าเป็นการอ่าน PORTC จะเป็นการอ่านค่าที่ LATCH ในขณะที่การอ่านค่าจาก PINC จะเป็นการอ่านค่าจริงของขาพอร์ต C

PORT C As General Digital I/O

ทั้ง 8 บิทของพอร์ต C จะมีลักษณะเหมือนกัน โดย PCn, แทนค่าต่างๆ ของพอร์ต C และ DDCn เป็นบิทที่ใช้ควบคุมทิศทางของขานั้นๆโดยถ้า DDCn, มีการเซ็ตเป็น 1 แสดงว่าเป็นขาเอาท์พุตแต่ถ้า DDCn ถูกเคลียเป็น 0 จะเป็นการกำหนดให้ขานั้นๆเป็นขาอินพุต ถ้าต้องการให้ MOS PULL UP เกิดการ ACTIVE จะต้องเซ็ต PORTCn มีค่าเป็น 1 และถ้าไม่ต้องการให้ MOS PULL UP เกิดการ ACTIVE จะเคลียร์ให้ PORTCn เป็น 0 โดยลักษณะการกำหนดการใช้งานของแต่ละขาดังตารางที่ 3.5

ตารางที่ 3.5 การใช้งานของพอร์ต C

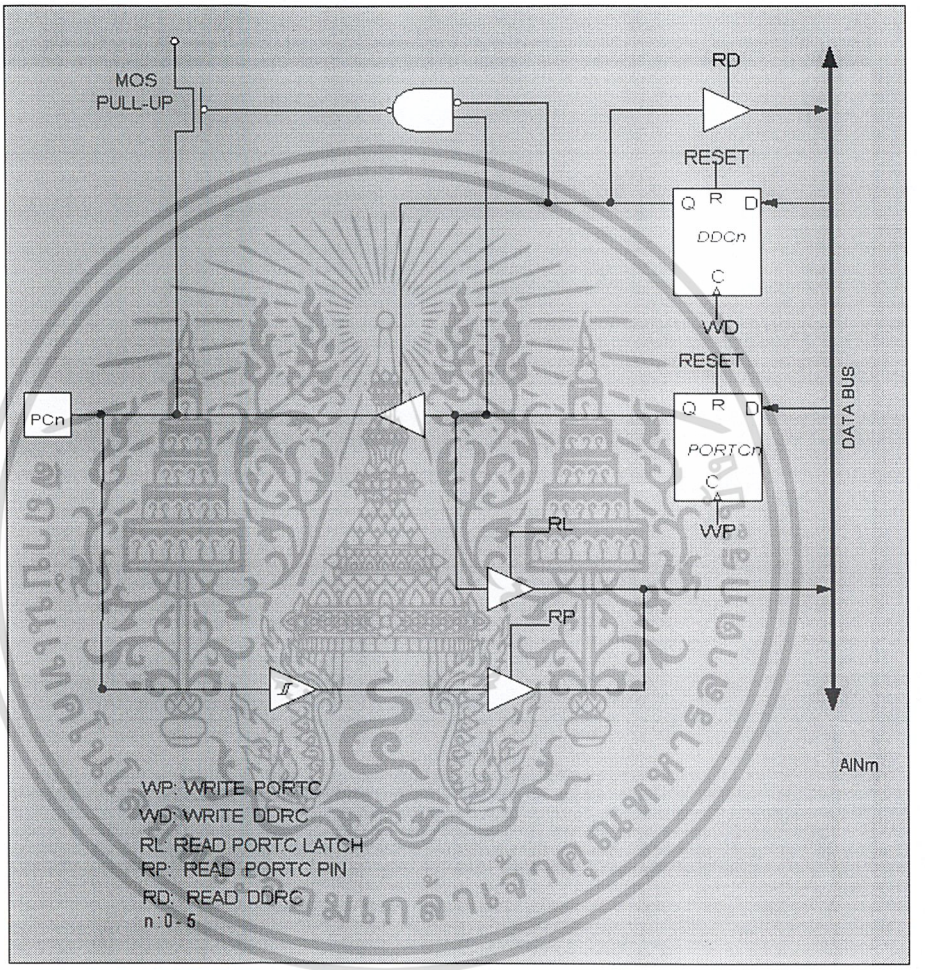
DDCn	PORTCn	I/O	Pull Up	Comment
0	0	Input	NO	Tri-state (Hi-Z)
0	1	Input	YES	PCn will source current if ext.pulled low
1	0	Output	NO	Push –pull Zero Output
1	1	Output	No	Push –pull One Output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ฟังชั่นอื่น ๆ ของพอร์ต C

โดยเมื่อบิท AS2 ในรีจีสเตอร์ ASSR ถูกเซ็ตให้เป็น 1 จะเป็นการกำหนดให้มีการรับ CLOCK จากภายนอกของ TIMER/COUNTER2 ที่ตำแหน่งขา PC6,PC

โครงสร้างของพอร์ต C



รูปที่ 3.26 โครงสร้างขาของพอร์ต C (PC0 – PC5)

พอร์ต D

พอร์ต D เป็นพอร์ต 2 ทิศทางขนาด 8 บิทที่มีหน่วยควบคุมการทำงานของพอร์ตคือ PORTD (Data Register –PORTD)อยู่ที่ตำแหน่ง $12 ($ 32), DDRD (Data Direction Regiter) อยู่ที่ตำแหน่ง $11($31), และ PIND (Port D Input pins) อยู่ที่ตำแหน่ง $10 ($ 30) โดย PIND สามารถอ่านได้อย่างเดียวในขณะที่ PORTD และ DDRD สามารถทั้งอ่านและเขียน โดยพอร์ตD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถรับกระแสได้ 20mA ซึ่งแต่ละขาของพอร์ต D สามารถเลือกฟังชั่นการทำงานอื่นๆได้อีกดังตารางที่ 3.6

ตารางที่ 3.6 การใช้งานเป็นฟังก์ชั่นอื่นๆ ของพอร์ต D

Port Pin	Afternate Function
PD0	RDX (UART input line)
PD1	TDX (UART output line)
PD2	INT0 (External interrupt 0 input)
PD3	INT1(External interrupt 1input)
PD4	OC1B (Timer / Counter1 output compare B match output
PD5	OC1A (Timer / Counter1 output compare A match output
PD6	ICP (Timer / Counter 1 input capture pin)
PD7	OC2 (Timer /counter 2 output compare match output

The Port D Data Register –PORTD

Bit $1D	7	6	5	4	3	2	1	0	
	PORT D7	PORT D6	PORT D5	PORT D4	PORT D3	PORT D2	PORT 1	PORT D0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**PORTD**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.27 รีจิสเตอร์ของ PORTD

The Port DData Direction Register -DDRD

Bit$1A	7	6	5	4	3	2	1	0	
	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	**DDRD**
Initial value	0	0	0	0	0	0	0	0	

รูปที่ 3.28 รีจิสเตอร์ของ DDRD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The Port DInput Pin Address-PIND

Bit$19	7	6	5	4	3	2	1	0	
	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0	
Read/Write	R	R	R	R	R	R	R	R	**PIND**
Initial value	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	

รูปที่ 3.29 รีจิสเตอร์ของ PIND

PIND ไม่ใช่รีจีสเตอร์ ฉะนั้นตำแหน่งที่กำหนดจะเป็นตำแหน่งจริงของขาพอร์ต โดยเมื่ออ่านค่าของ PORTD และDDRD จะเป็นการอ่านค่าที่ LATCH ส่วนการอ่านค่าของ PIND จะเป็นการอ่านค่าจริงๆที่เกิดขึ้นขาของพอร์ต D

การใช้งานของพอร์ต D เป็นพอร์ตอินพุตและเอาท์พุต

โดย PDn แทนขาใดๆของพอร์ต D และ DDDn เป็นตำแหน่งบิทในรีจีสเตอร์ DDRD ซึ่งรีจีสเตอร์ DDRD ทำหน้าที่ควบคุม ทิศทางของพอร์ต D โดยถ้าเซ็ตให้เป็น 1 จะเป็นการกำหนด ให้พอร์ตทำงานเป็นเอาท์พุตแต่ถ้าเคลียร์ให้เป็น 0 จะกำหนดให้พอร์ตทำงานเป็นอินพุต

ตารางที่ 3.7 การใช้งานพอร์ต D

DDDn	PORTDn	I/O	Pull Up	Comment
0	0	Input	NO	Tri-state (Hi-Z)
0	1	Input	YES	PDn will source current if ext.pulled low
1	0	output	NO	Push –pull Zero Output
1	1	output	NO	Push –pull One Output

การทำงานฟังชั่นอื่นๆของพอร์ต D

OC2- PORTD,BIT 7

ขา OC2 หรือบิท 7 ของพอร์ต D สามารถถูกนำไปใช้งานเป็นขาเอาท์พุตของ Compare Output Match ของ timer /counter2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานฟังชั่นอื่นๆของพอร์ต D

OC2- PORTD,BIT 7

ขา OC2 หรือบิท 7 ของพอร์ต D สามารถถูกนำไปใช้งานเป็นขาเอาท์พุตของ Compare Output Match ของ timer /counter2

ICP- PORTD, BIT 6

ขา ICP หรือ บิท 6 ของพอร์ต D สามารถถูกนำไปใช้งานเป็นขาอินพุตของ Input capture mode ของ timer /counter1

OC1A - PORTD,BIT 5

ขา OC1A หรือบิท 5 ของพอร์ต D สามารถถูกนำไปใช้งานเป็นขาเอาท์พุต ของ Output Compare Amatch ของ timer /counter1

OC1B- PORTD,BIT4

ขา OC1B หรือบิท 5 ของพอร์ต D สามารถถูกนำไปใช้งานเป็นขาเอาท์พุต ของ Output Compare B Match ของ timer /counter1

INT 1 – PORTD , BIT 3

ขา INT1 หรือบิท 3 ของพอร์ต D สามารถถูกใช้งานเป็นขาอินพุตของอินเตอร์รัพท์ จากภายนอก

INT 0– PORTD , BIT 2

ขา INT2 หรือบิท 2 ของพอร์ต D สามารถถูกใช้งานเป็นขาอินพุตของอินเตอร์รัพท์ จากภายนอก

TXD –PORTD ,BIT1

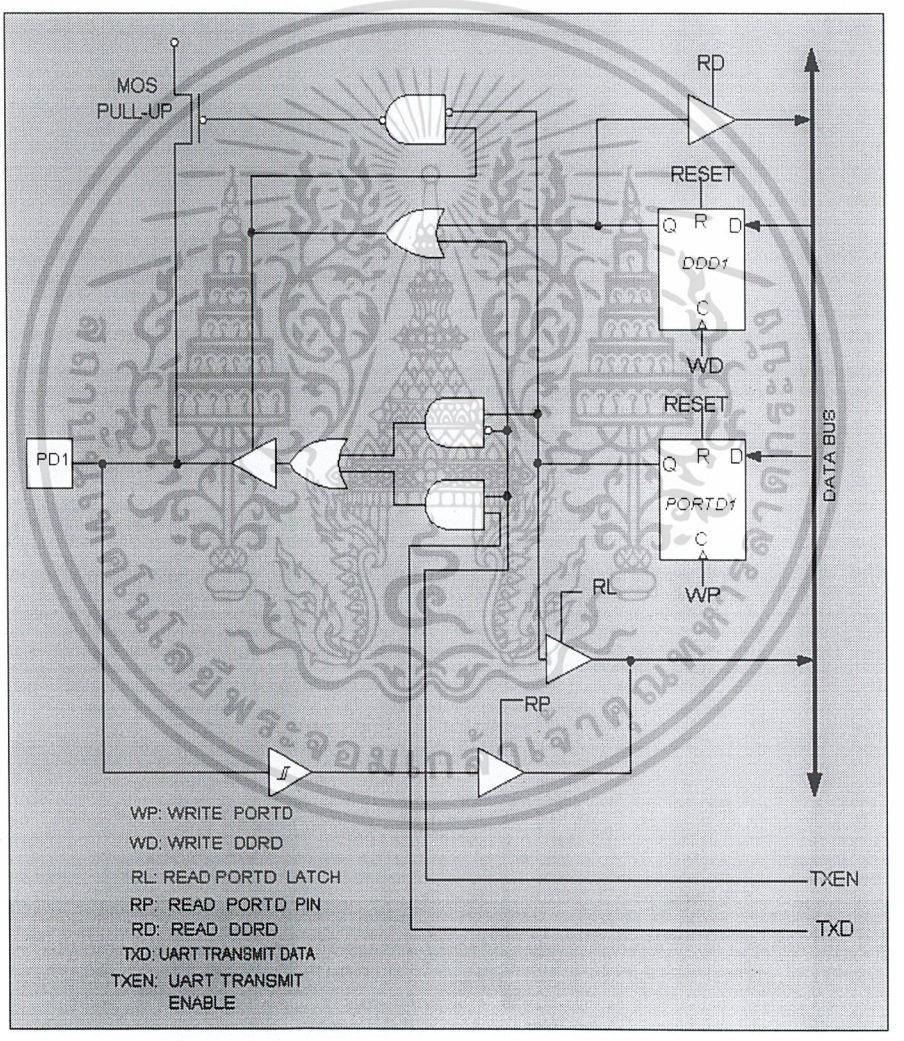
ขา TXD หรือบิท 1 ของพอร์ต D สามารถถูกใช้งานเป็นขาส่งสัญญาณของการสื่อสารอนุกรมแบบ UART

RXD –PORTD ,BIT0

ขา RXD หรือบิท 0 ของพอร์ต D สามารถถูกใช้งานเป็นขาส่งสัญญาณของการสื่อสารอนุกรมแบบ UART

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของพอร์ต D



รูปที่ 3.30 โครงสร้างขาของพอร์ต D (PD0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

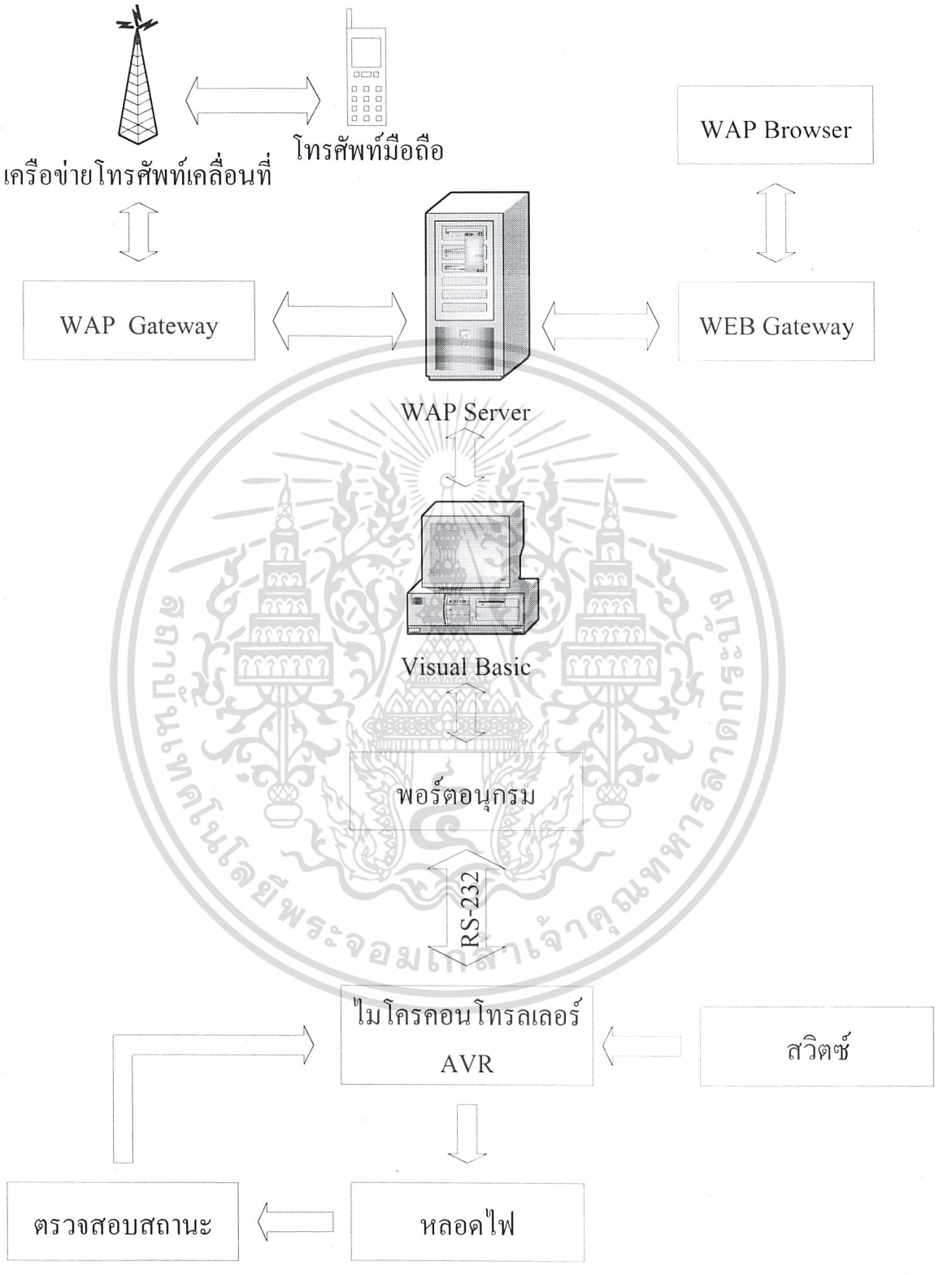
บทที่ 4
การออกแบบ

4.1 การออกแบบระบบโดยรวม

ในปริญญานิพนธ์นี้เป็นการออกแบบระบบควบคุมเครื่องใช้ไฟฟ้าภายในบ้านโดยใช้โทรศัพท์มือถือผ่านระบบอินเตอร์เน็ต โดยสามารถสั่งงานผ่านงานผ่านส่วนต่างๆ ได้หลายส่วน ในส่วนแรกคือการสั่งงานผ่านเครื่องโทรศัพท์มือถือ ซึ่งวิธีการที่จะใช้งานจะต้องใช้โทรศัพท์มือถือที่สามารถใช้งาน WAP ได้แล้วทำการเรียก IP Address ของ WAP Server ที่ต้องการเพื่อใช้ควบคุมเครื่องใช้ไฟฟ้าได้

นอกจากการควบคุมเครื่องใช้ไฟฟ้าโดยใช้โทรศัพท์มือถือแล้ว สามารถสั่งงานผ่านอุปกรณ์ WAP ซึ่งเป็นอุปกรณ์ที่สามารถใช้ WAP Browser ได้ เช่น Plam ,PocketPC เป็นต้น หรือใช้ WAP Browser ที่ดาวน์โหลดจากอินเตอร์เน็ตก็ได้ เมื่อทำการสั่งงานเรียบร้อยแล้วข้อมูลที่ได้จะถูกส่งไปยังคอมพิวเตอร์ซึ่งเชื่อมต่อกับอินเตอร์เน็ตเพื่อรอรับข้อมูลอยู่ตลอดเวลา โดยมีโปรแกรม Visual Basic เป็นตัวประมวลผลข้อมูลที่เข้ามาอยู่ตลอดเวลาและจะทำการส่งข้อมูลที่ประมวลผลเสร็จแล้วไปยังพอร์ตอนุกรมผ่าน RS-232 ไปยังไมโครคอนโทรลเลอร์ AVR ซึ่งไมโครคอนโทรลเลอร์ AVR นี้จะใช้ควบคุมเครื่องใช้ไฟฟ้าให้ทำการหรี่ไฟ เปิดไฟ ดับไฟและสารถตรวจว่าหลอดขาดได้ และเมื่อทำการเปลี่ยนสถานะของเครื่องไฟฟ้าแล้วไมโครคอนโทรลเลอร์จะทำการส่งค่าสถานะที่มีการเปลี่ยนนั้นกลับไปให้กับ Visual Basic เพื่อนำกลับไปแสดงบนโทรศัพท์มือถือหรืออุปกรณ์ WAP ต่อไป ดังรูปที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การทำงานโดยรวมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การออกแบบ WAP Site โดยใช้ ภาษา WML และ PHP

ในการออกแบบ WAP Site เพื่อที่จะให้ผู้ใช้สามารถรับรู้ข้อมูลจากโทรศัพท์มือถือและสามารถควบคุมอุปกรณ์ไฟฟ้าผ่านทางอินเตอร์เน็ตจะต้องทำความเข้าใจเกี่ยวกับภาษา WML และ PHP ซึ่งใช้ในการออกแบบก่อน ภาษา WML และ PHP สามารถเขียนจากเครื่องมือเขียนข้อความต่างๆ เช่น Text Editor, EditPlus หรือ Notepad อย่างไรก็ตาม WAP ยังมีเครื่องมือที่ใช้ช่วยในการเขียน WAP เช่นเดียวกับเครื่องมือที่ช่วยในการเขียน HTML ที่เรียกว่า WAP ToolKit ซึ่งมีอยู่หลายโปรแกรมที่มีความน่าสนใจและสามารถดาวน์โหลดโปรแกรมได้จากอินเตอร์เน็ต โดยส่วนประกอบสำคัญของการออกแบบ WAP Site ประกอบไปด้วยส่วนหลักๆ 2 ส่วนคือ

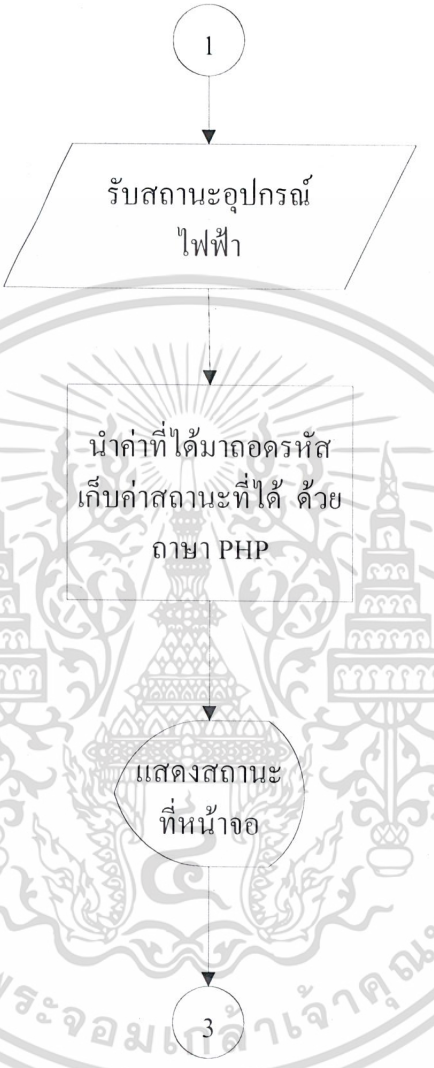
1. ส่วนของการแสดงสถานะให้ผู้ใช้สามารถตรวจสอบสถานะของเครื่องใช้ไฟฟ้า
2. ส่วนของการควบคุมเครื่องใช้ไฟฟ้าผ่านโทรศัพท์มือถือ

การทำงานของ WAP Site แสดงดังรูป



รูปที่ 4.2 Flow Chart โปรแกรมในหน้าเมนูหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



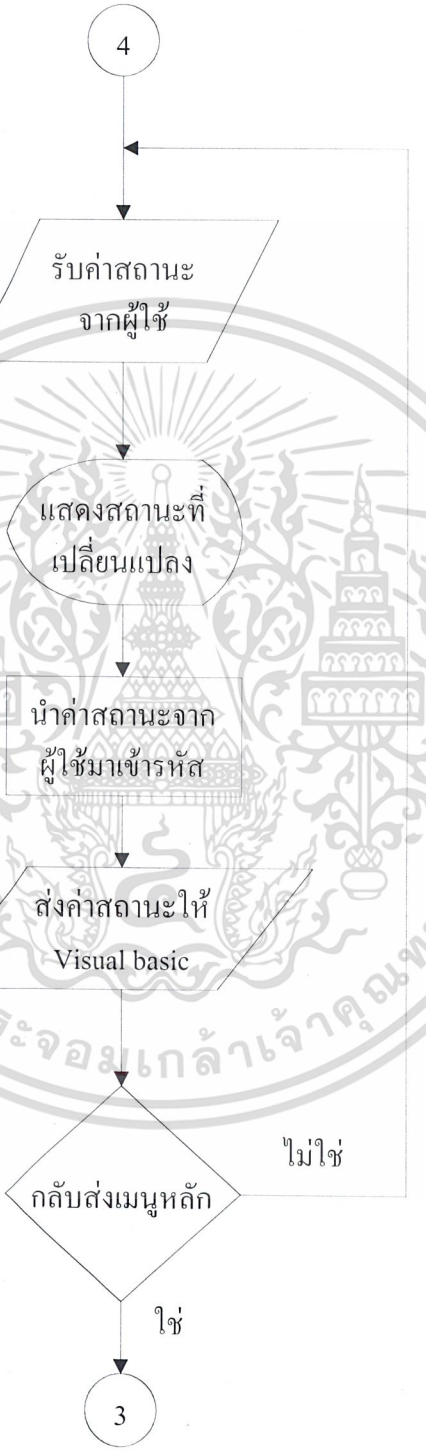
รูปที่ 4.3 Flow Chart โปรแกรมในส่วนของการแสดงสถานะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Flow Chart โปรแกรมในส่วนของการควบคุมเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 Flow Chart โปรแกรมในส่วนของการควบคุมเครื่องใช้ไฟฟ้า(ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 พอร์ตอนุกรม

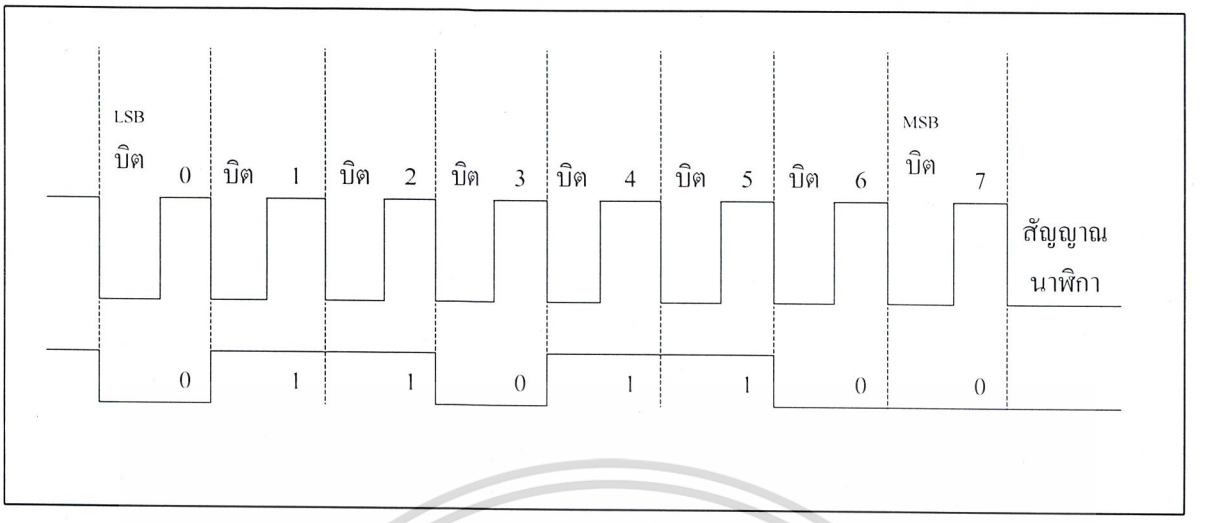
มีทางเลือกอยู่ 2 แนวทางในการที่จะเคลื่อนย้ายข้อมูลจากคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่นๆหรือคอมพิวเตอร์ด้วยกัน นั่นคือการรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรมการรับส่งข้อมูลแบบขนาน จะเป็นการรับหรือส่งข้อมูลคราวละ 4 หรือ 8 บิต ในเวลาเดียวกัน ซึ่งจะทำให้การรับและส่งข้อมูลทำได้ที่ความเร็วสูง ซึ่งก็หมายความว่าจำนวนของสายที่ใช้ในการส่งจะต้องมีมากเท่ากับจำนวนบิตของข้อมูลที่จะส่งด้วย นอกจากนี้ยังจะต้องรวมถึงสายที่ใช้สำหรับการควบคุมและการตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายมากเป็น 2 เท่าของจำนวนบิตข้อมูล ที่จะส่งก็ได้ ซึ่งก็เป็นปัญหาในเรื่องราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักจะมีราคาแพง

ในขณะที่การรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถรับส่งข้อมูลได้คราวละหลายๆบิต ได้ หากแต่จะต้องมีการตกลงกันระหว่างตัวส่งและตัวรับว่า จะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรอข้อมูลมาให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลอนุกรมอาจมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายที่น้อยกว่ามาก อย่างน้อยที่สุดใช้เพียง 2 – 3 เส้นเท่านั้น แต่อัตราเร็วในการรับส่งข้อมูลอาจต่ำกว่าแบบขนาน อย่างไรก็ตามการรับส่งข้อมูลแบบอนุกรมสามารถใช้สายสัญญาณที่มีความยาวมากกว่าแบบขนาน ทำให้ระยะทางในการสื่อสารข้อมูลแบบอนุกรมสามารถทำได้มากกว่า

4.3.1 การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมนั้นจะแบ่งออกได้เป็น 2 แบบ คือการสื่อสารอนุกรมแบบซิงโครนัสและการสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือคีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสายของสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา ,ข้อมูลและกราวด์ รูปที่ 4.6 แสดงให้เห็นถึงไทมิ่งไดอะแกรมของการส่งข้อมูลแบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรม

4.3.2 การสื่อสารข้อมูลแบบอะซิงโครนัส

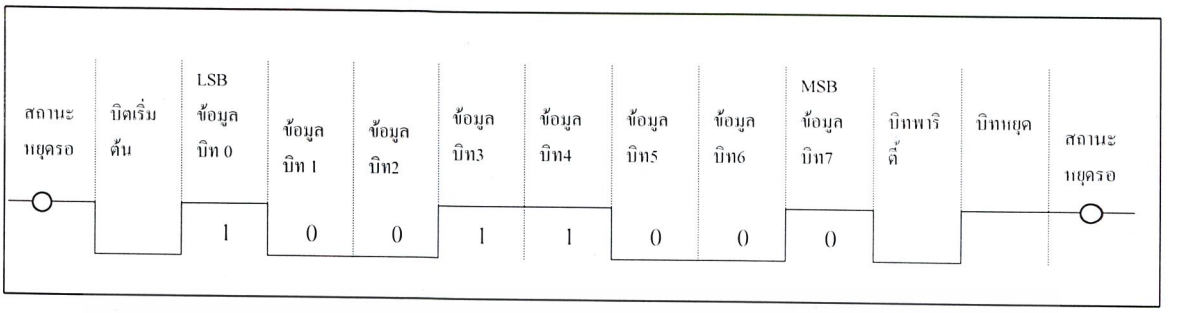
การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate)มีหน่วยเป็น บิตต่อวินาที (bit per second : bps)

รูปแบบของข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัสประกอบด้วย 4 ส่วน คือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต
2. บิตข้อมูลแบบอนุกรมจะมีขนาด 5,6,7 หรือ9 บิต
3. บิตตรวจสอบพาริตี้ (Parity Bit)จะมีขนาด 1 บิต หรือไม่มี
4. บิตปิดท้าย (Stop Bit) จะมีขนาด 1,1.5 หรือ 2 บิต

รูปที่ 4.7 แสดงรูปแบบของข้อมูลอนุกรมแบบอะซิงโครนัส ซึ่งเมื่อไม่มีข้อมูลที่จะส่งขา DATA จะมีสถานะ ลอจิก “1” ซึ่งจะเรียกสถานะนี้ว่าสถานะหยุดรอ (waiting stage) การเริ่มต้นส่งข้อมูลจะเริ่มจากการให้ขา DATA มีลอจิก “0”ด้วยช่วงระยะเวลา 1 บิต ซึ่งจะเรียกบิตนี้ว่าบิตเริ่มต้น จากนั้นบิตข้อมูลจะถูกส่งออกไป โดยเริ่มจากบิตที่มีนัยสำคัญต่ำสุด (LSB) ก่อน ซึ่งข้อมูลในไบต์ที่จะส่งอาจมีจำนวนบิต 5,6,7 หรือ 8 บิตก็ได้ จากนั้นจากตามด้วยบิตพาริตี้ ซึ่งใช้เพื่อตรวจสอบความผิดพลาดที่เกิดขึ้นจากการส่งข้อมูล บิตสุดท้ายที่จะส่งคือบิตปิดท้าย ซึ่งจะให้ขาดาต้ามีสถานะลอจิก 1 อีกครั้งด้วยระยะเวลาอย่างน้อย 1 บิต, 1.5 บิต หรือ 2 บิต เพื่อเป็นการแสดงว่าสิ้นสุดข้อมูลแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 รูปแบบอย่างง่ายที่สุดของข้อมูลอนุกรมแบบอะซิงโครนัส

อุปกรณ์พิเศษที่ได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัสเรียกว่า **Universal Asynchronous Receiver / Transmitter** หรือ UARTอัตราความเร็วในการรับและส่งข้อมูลของการรับส่งข้อมูลแบบอะซิงโครนัสคือ **ค่าบอดเรต** ซึ่งก็คือค่าจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูล บอดเรตมาตรฐานที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110,150,300,600,1200,2400,4800,9600 และ 19200 บิตต่อวินาที และมีค่าเพิ่มมากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะสามารถกำหนดค่าบิดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบิตเรตคือจำนวนบิตข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่าง ข้อมูลอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี้ มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บิดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลได้ด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตี้ความเร็วในการรับส่งข้อมูลจะเหลือเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตี้สามารถกำหนดในเป็นแบบคี่ (odd),แบบคู่ (even)หรือไม่มีการตรวจสอบพาริตี้ก็ได้ การตรวจสอบพาริตี้เป็นการตรวจสอบจำนวนของบิตที่เป็นลอจิก “1” ภายในข้อมูลที่ส่งไป 1 ไบต์ว่ามีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตี้เข้าไปด้วย ยกตัวอย่างข้อมูลที่จะทำการส่งมีขนาด 8 บิต และมีค่าเท่ากับ 99 ฐานสิบหก หรือ 10011001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวนลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ดังนั้นถ้ากำหนดค่าพาริตี้เป็นคู่ค่าในบิตพาริตี้ จะต้องมีลอจิกเป็น”0” แต่ถ้าพาริตี้เป็นคี่ ค่าบิตพาริตี้จะต้องเป็น”1” เพื่อให้ข้อมูล 1 ไบต์ รวมทั้งบิตพาริตี้มีจำนวนบิตที่เป็นลอจิก “1” มีจำนวนรวมกันเป็นเลขคี่ ในตารางที่ 4.1 แสดงตัวอย่างของบิตพาริตี้ในการรับส่งข้อมูลอนุกรม

บิตพาริตี้ถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องการกำหนดคุณสมบัติการตรวจสอบพาริตี้ให้ตรงกันว่าจะตรวจสอบพาริตี้คี่หรือพาริตี้คู่ จากนั้นภาครับของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UART จะทำการตรวจสอบค่าพาริตี้ที่เกิดขึ้นว่าเป็นคู่หรือคี่ โดยการนับจำนวนลอจิก "1"ทั้งหมด รวมทั้งบิตพาริตี้ด้วย ถ้ากำหนดพาริตี้ไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้ตัวเลขเป็นคี่ทั้งภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียว เท่านั้น ถ้าข้อมูลที่ทำการส่งมีบิตที่ผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตี้บิตเป็น NONE นั้นทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบพาริตี้

ตารางที่ 4.1 บิตพาริตี้ของข้อมูล

ข้อมูล	บิตพาริตี้คู่	บิตพาริตี้คี่
00000000	0	1
00000001	1	0
00000010	1	0
00000011	0	1
00000100	1	0
11111110	0	1
11111111	1	0

คอมพิวเตอร์ในรุ่น AT เกือบทั้งหมดจะใช้ UARTเบอร์ 16450 และ 16550 ส่วนคอมพิวเตอร์ในรุ่น XT ใช้ UARTเบอร์ 8250 UART ชิปเหล่านี้มีระดับแรงดันแบบทีทีแอล (1 และ –5 V)แต่เพื่อให้มีแรงดันเป็นไปตามมาตรฐาน RS-232 และเพื่อให้การรับส่งข้อมูลสามารถทำได้ที่ระยะทางไกลมากขึ้น ระดับแรงดันทีทีแอลจะถูกแปลงเป็นระดับแรงดันที่สูงขึ้น โดยลอจิก "0"มีระดับแรงดัน +3V ถึง +12 V ในขณะที่ลอจิก "1" มีระดับแรงดัน –3 V จนถึง –12 V

4.3.3 มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิก (Electronic Indrustries Association : EIA)ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS –232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ –3 V ถึง –12 V แสดงว่ามีข้อมูล (Mark) และ+3 V ถึง +12 V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE)กับวงจรข้อมูลปลายทาง (Data Cricuit Terminating : DCE)ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรเลอร์หรือไมโครคอมพิวเตอร์ ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCEจะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

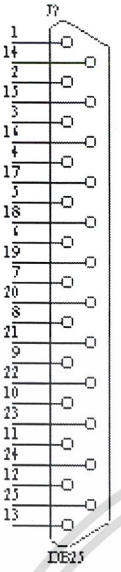
ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นได้ชัดคือ คอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE

สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักจะถูกใช้เชื่อมต่อกับโมเด็มหรือเมาท์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

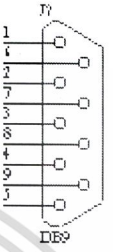
4.3.4 คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆที่เคยใช้งานในอดีต ปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไป โดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(ก) คอนเน็กเตอร์รอนุกรม25 ขาหรือแบบ DB-25
(มองจากด้านหลังคอมพิวเตอร์)



(ข) คอนเน็กเตอร์อนุกรม9 ขาหรือแบบ DB-9
(มองจากด้านหลังคอมพิวเตอร์)

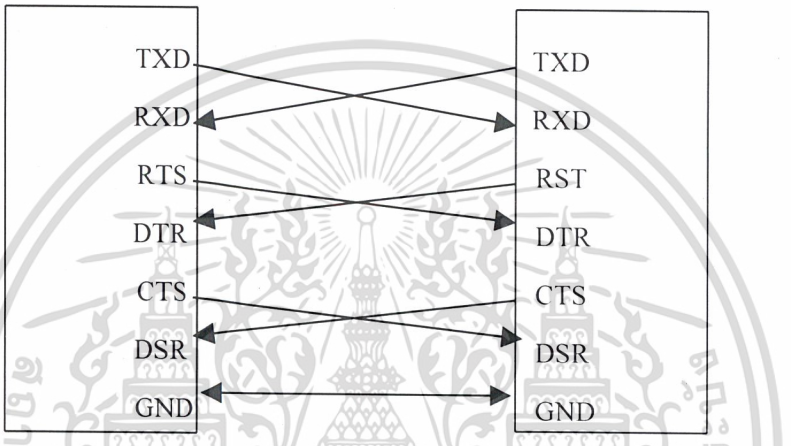
รูปที่ 4.8 การจัดขาของคอนเน็กเตอร์พอร์ตอนุกรมตามมาตรฐาน RS-232 ทั้งแบบ DB-9และDB-25

ตารางที่ 4.2 การจัดขา DB-9 และ DB-25

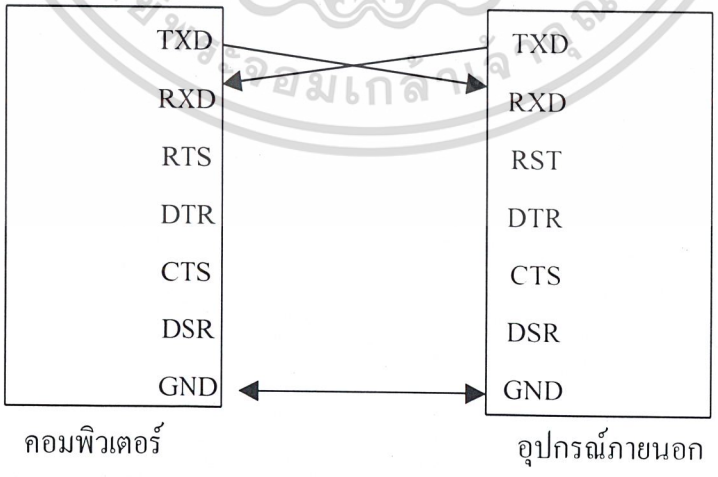
คอนเน็กเตอร์ DB9	คอนเน็กเตอร์ DB25	ชื่อของสายสัญญาณ	ชนิดของสายสัญญาณ
1	8	Data Carrier Detect:DCD	อินพุต
2	3	ReceicedData : RxD	อินพุต
3	2	Transmitted Data : TxD	เอาต์พุต
4	20	Data Terminal Teady : DTR	เอาต์พุต
5	7	Signal Ground : GND	-
6	6	Data Set Ready : DSR	อินพุต
7	4	Request To Send : RTS	เอาต์พุต
8	5	Clear To Send : CTS	อินพุต
9	22	Ring Indicator : RI	อินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกแสดงดังรูป 4.9 ลูกศรในรูปแสดงถึงทิศทางของข้อมูล ในรูปที่ 4.9 (ก)เป็นการเชื่อมต่อแบบ Null modem หรือการเชื่อมต่อโดยตรงโดยไม่ต้องผ่านโมเด็ม โดยมีการตรวจสอบหรือแฮนด์เช็กเต็มรูปแบบ ส่วนในรูปที่ 4.9 (ข) เป็นการเชื่อมต่อแบบ Null modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้น โดยเส้นหนึ่งสำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูล และเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ตอนุกรม RS-232 มีดังนี้



(ก) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ Null Modem



(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS 232 โดยใช้สัญญาณ 3 เส้น

รูปที่ 4.9 การต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-Data Carrier Detect : DCD หรืออาจเรียกว่า Carrier Detect : CD ขานี้จะแอกตีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็ม สำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

-Receive Data : RD หรือ RxD ขานี้ใช้เพื่อรับส่งสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์ โดยนำข้อมูลที่อ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์

-Transmitted Data : TD หรือ TxD ขานี้ใช้เพื่อส่งข้อมูลออกจากคอมพิวเตอร์โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลส่งออกไป

-Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์เพื่อให้อุปกรณ์ปลายทางรับรู้ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง และขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อขา DCDด้วยในกรณีที่โปรแกรมสื่อสารที่ใช้มีการตรวจจับสัญญาณพาห์

-Signal Ground : GND ขากราวด์ของระบบ

-Data Set Ready : DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSRนี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

-Request To Send : RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null modem 3 สาย จะต้องเชื่อมต่อขา RTS และ CTS ของตัวมันเองเข้าด้วยกัน เพื่อจะให้การรับและส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา

-Clear To Send :CTS ขานี้จะคอยรับสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไปสายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับโมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

4.3.5 UART

UART มาจากคำว่า Universal Asynchronous Receiver Transmitter ซึ่งหมายถึงอุปกรณ์ที่ทำหน้าที่รับและส่งข้อมูลแบบอะซิงโครนัสนั่นเอง สำหรับการสื่อสารอนุกรมบนคอมพิวเตอร์แล้ว UART ถือว่าเป็นหัวใจสำคัญของการสื่อสารอนุกรม

หน้าที่หลักของ UART คือทำหน้าที่แปลงข้อมูลที่อยู่ในรูปแบบขนานจากคอมพิวเตอร์ให้อยู่ในรูปแบบอนุกรมแบบอะซิงโครนัส แล้วส่งออกไป และทำหน้าที่แปลงสัญญาณอนุกรมแบบอะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซิงโครนัสที่ป้อนเข้ามายัง UART ให้เป็นแบบขนานก่อนที่จะส่งข้อเข้าสู่คอมพิวเตอร์ ซึ่งนอกจาก UART จะส่งข้อมูลไปยังคอมพิวเตอร์แล้ว ยังแจ้งข้อมูลอื่นๆ ให้คอมพิวเตอร์รับทราบด้วย เช่น อัตราเร็วในการรับส่งข้อมูล (บอดเรต), รูปแบบการส่งข้อมูล,ความผิดพลาดที่เกิดขึ้นระหว่างการถ่ายทอดข้อมูล (ผิดพลาดจากพาริตี้, เฟรมข้อมูล, โอเวอร์รัน) เป็นต้น

ภายใน UART จะมีส่วนของวงจรสร้างบอดเรตแบบโปรแกรมได้ (progrannable buadrate generator)โดยการกำหนดค่าตัวหารให้กับสัญญาณนาฬิกาของ UART โดยตัวหารนี้มีขนาด 16 บิต ดังนั้นจึงสามารถกำหนดตัวหารอยู่ในช่วง 1-65,535 UART สามารถรับส่งข้อมูลได้ทั้งแบบ ฮาล์ฟดูเพล็กซ์ (half duplex)และฟูลดูเพล็กซ์(full duplex) โดยการส่งแบบฮาล์ฟดูเพล็กซ์เป็นการส่งแบบทิศทางเดียว ส่วนการส่งแบบฟูลดูเพล็กซ์นั้นสามารถรับและส่งข้อมูลได้ในคราวเดียวกัน

ชนิดของ UART

ในเครื่องคอมพิวเตอร์ทั่วไปมี UART ที่ใช้งานกันอยู่ 2 เบอร์ คือ 8250 ซึ่งเป็น UART มาตรฐานที่มีใช้กันมายาวนาน UART เบอร์นี้จะมีบัฟเฟอร์สำหรับรับและส่งข้อมูลตำแหน่งเดียวกัน ทำให้การรับและส่งข้อมูลถูกจำกัดความเร็วอยู่ที่ 57.6 กิโลบิตต่อวินาทีเท่านั้น แต่ UART เบอร์นี้ถือว่าเป็นต้นแบบ ของ UART ที่ใช้ในคอมพิวเตอร์ โดยคอมพิวเตอร์ทุกๆรุ่นจะต้องสนับสนุนการทำงานตามรูปแบบของ UART เบอร์นี้

UART อีกเบอร์หนึ่งคือ 16450 มีความสามารถรับส่งข้อมูลได้ที่ความเร็ว 115,200 บิตต่อวินาที และเพิ่มรีจิสเตอร์สำหรับพักข้อมูลสำหรับ UART นอกจากนั้นยังเพิ่มส่วนของชิฟต์รีจิสเตอร์แบบ FIFO (First In First Out) ขนาด 16 ไบต์เข้าไป ทำให้สามารถสนับสนุนความเร็วในการรับส่งข้อมูลที่ 256 กิโลบิตต่อวินาทีได้ โดยคอมพิวเตอร์ในปัจจุบันใช้ UARTเบอร์นี้หรือใหม่กว่า เช่น เบอร์ TL15C750 ซึ่งมีรีจิสเตอร์แบบ FIFOขนาด 64 ไบต์ ทำงานได้ที่ระดับแรงดัน +5 V และ +3 V มีโหมดการประหยัดพลังงาน สามารถรับส่งข้อมูลได้ที่ความเร็ว 1 เมกะบิตต่อวินาทีเมื่อใช้สัญญาณนาฬิกา 16 MH_z

อย่างไรก็ตาม ความเร็วในการส่งข้อมูลที่มากมายของ UART เบอร์ใหม่ๆก็ไม่ได้ช่วยให้การรับส่งข้อมูลของคอมพิวเตอร์เร็วขึ้น เนื่องจากว่าคอมพิวเตอร์ยังใช้ความถี่ของสัญญาณนาฬิกาแบบในการแปลงข้อมูลเพียง 1.8432 MH_zเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 วงจรภายในและรีจิสเตอร์ของพอร์ตอนุกรม RS-232

เครื่องคอมพิวเตอร์โดยทั่งไปสามารถต่อพอร์ตอนุกรม RS-232 สูงสุดได้ 4 พอร์ต ซึ่งจะมีชื่อเรียกเป็น COM 1, COM2, COM3 และ COM4 ซึ่งพอร์ตอนุกรมแต่ละตัวต่างก็ใช้งาน UART ภายในคอมพิวเตอร์ในการติดต่อกับอุปกรณ์ภายนอกเช่นเดียวกัน

การทำงานภายในพอร์ตอนุกรม ซึ่งประกอบไปด้วยรีจิสเตอร์ขนาด 8 บิต 8 ตัวที่ใช้งานร่วมกับ UART แอดแดรสของรีจิสเตอร์ภายในพอร์ตอนุกรมสามารถคำนวณได้จากค่ารีจิสเตอร์พื้นฐานของพอร์ตอนุกรม ยกตัวอย่าง พอร์ตอนุกรม COM 1 มีแอดแดรสอยู่ที่ 3F8H ตำแหน่งของรีจิสเตอร์ต่างๆจะเป็นตำแหน่งที่บวกเข้าไปกับค่า 3F8H โดยรีจิสเตอร์ที่ใช้งานกับพอร์ตอนุกรมดังนี้

00H เป็นรีจีสเตอร์บัฟเฟอร์สำหรับเก็บข้อมูลที่รับเข้ามาหรือเตรียมข้อมูลก่อนที่จะส่งออกไป

01H รีจิสเตอร์อีนาเบิลการอินเตอร์รัปต์ ใช้ในการเซตโหมดการอินเตอร์รัปต์ของพอร์ตอนุกรม

02H รีจิสเตอร์แสดงโหมดการอินเตอร์รัปต์ ใช้เพื่อตรวจสอบโหมดของอินเตอร์รัปต์เมื่อมีการอินเตอร์รัปต์เกิดขึ้น

03H รีจีสเตอร์กำหนดรูปแบบของข้อมูล

04H รีจีสเตอร์ควบคุมโมเด็ม ใช้ตรวจสอบบิตสำหรับติดต่อกับโมเด็ม เช่น RTS หรือ DTR

05H รีจีสเตอร์แสดงสถานะการรับและการส่งข้อมูลแบบอนุกรม

06H รีจิสเตอร์แสดงสถานะของโมเด็ม ซึ่งจะแสดงสถานะของขา DCD, RI, DSRและ CTS

07H รีจิสเตอร์สำหรับการเก็บข้อมูลชั่วคราว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจีสเตอร์ตำแหน่ง 00H : รีจีสเตอร์บัฟเฟอร์

เป็นรีจีสเตอร์สำหรับเก็บข้อมูลที่รับเข้ามาและข้อมูลที่จะส่งออกไป โดยการติดต่อกับรีจีสเตอร์นี้เพื่อเก็บข้อมูลที่ต้องการจะส่งจะต้องกำหนดให้บิต DLAB ในรีจีสเตอร์กำหนดรูปแบบข้อมูล (03H)จะต้องมีสถานะเป็น 0 ซึ่งการเขียนข้อมูลมายังแอดแดรสนี้ เป็นการส่งข้อมูลไปยังรีจีสเตอร์ส่งข้อมูลและข้อมูลจะถูกส่งออกไปแบบอนุกรม สำหรับการรับข้อมูล เมื่อข้อมูลที่รับเข้ามาเรียบร้อยและแปลงเป็นแบบขนานแล้ว ข้อมูลจะถูกส่งมายังรีจีสเตอร์เก็บข้อมูล หลังจากมีการอ่านรีจีสเตอร์นี้ออกไปรีจีสเตอร์นี้จะถูกเคลียร์ และเตรียมพร้อมสำหรับการรับข้อมูลในไบต์ต่อไป

4.3.7 ลักษณะสัญญาณอินพุตและเอาท์พุตของพอร์ต RS-232

สัญญาณเอาท์พุตที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุต (CTS, DSRและ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาต์พุตออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

แอสเดรสของพอร์ตอนุกรม

แอสเดรสพื้นฐานของพอร์ตอนุกรม มี 4 ตำแหน่งดังนี้คือ

COM 1 : 3F8H

COM 2 : 2F8H

COM 3 : 3E8H

COM 4 : 2E8H

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM1จะเก็บไว้ที่แอดเดรส 0000 : 0400 H และ 0000 : 0401H ส่วนตำแหน่งอื่นๆมีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM 2 = 0000 : 0402 H – 0000 : 0403 H

COM 3 = 0000 : 0404 H – 0000 : 0405 H

COM 4 = 0000 : 0406 H – 0000 : 0407 H

นอกจากนี้ที่หน่วยความจำแอดแดรส 0000 : 0411 ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วยโดยมีรายละเอียดดังแสดงในตารางที่ 4.2

ตารางที่ 4.3 ข้อมูลในแอดเดรส 0000 : 0411 H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

บิท3	บิท2	บิท1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การเขียนโปรแกรมเพื่อใช้งานพอร์ตอนุกรม

คอนโทรล MSComm

สำหรับการใช้งาน Visual BASIC ตั้งแต่เวอร์ชัน 2 เป็นต้นมา ใน Visual BASIC จะมีคัสตอมคอนโทรลสำหรับการสื่อสารอนุกรมผ่านทางพอร์ตอนุกรมของคอมพิวเอตร์มาให้ โดยใน Visual BASIC เวอร์ชัน 2 และเวอร์ชัน 3 จะใช้ชื่อว่า **MSCOMM.VBX** ส่วน เวอร์ชัน 4 ใช้ชื่อว่า MSCOMM16.OCX สำหรับการทำงานกับระบบปฏิบัติการ 16 บิต และ **MACOMM32.OCX** สำหรับการทำงานกับระบบปฏิบัติการ 32 บิต สำหรับใน Visual BASIC เวอร์ชัน 5 จะมีเพียง MSCOMM32.OCX เท่านั้นเพราะถูกออกแบบมาให้ใช้งานกับระบบปฏิบัติการ 32 บิต

MSComm จัดเตรียมทางเลือกเอาไว้ 2 ทางเพื่อความสะดวกในการสื่อสารข้อมูล ทางแรกคือ การสื่อสารข้อมูลที่กระตุ้นด้วยเหตุการณ์ (event-driven communications) เป็นรูปแบบการใช้งานที่มีประสิทธิภาพมากสำหรับการตอบสนองแบบทันทีทันใด เช่น เมื่อตัวอักษรถูกส่งมาที่พอร์ตอนุกรมหรือเกิดการเปลี่ยนแปลงที่ขา Data Carrier (DCD)หรือขา Request To Send (RTS)เหตุการณ์ Oncomm ของ MSCommจะสามารถจับตรวจสัญญาณนั้นได้ทันที ซึ่งจะกล่าวถึงรายละเอียดในหัวข้อคุณสมบัติ CommEven ต่อไป ส่วนทางเลือกที่สองเป็นการคอยตรวจสอบค่าเหตุการณ์และความผิดพลาดที่เกิดขึ้นด้วยการดูค่าที่เปลี่ยนแปลงภายในคุณสมบัติ ConnEven หลังจากให้โปรแกรมทำงานในฟังก์ชั่นต่างๆ ไปเรียบร้อยแล้ว ซึ่งวิธีนี้ใช้งานได้ดีในกรณีที่โปรแกรมมีขนาดเล็ก

คอนโทรล MSComm 1 ตัวสามารถควบคุมการทำงานของพอร์ตอนุกรมได้ 1 พอร์ต ถ้าในโปรแกรมที่ใช้งานต้องการติดต่อกับพอร์ตอนุกรมมากกว่า 1 พอร์ต จะต้องใช้คอนโทรล MSComm มากกว่า 1 ตัวเพื่อควบคุมพอร์ตอนุกรมในแต่ละพอร์ต แอดเดรสของพอร์ตอนุกรมและแอดเดรสของการเกิดอินเตอร์รัปต์สามารถเปลี่ยนแปลงได้จากการแก้ไขค่าที่ Control Panel

CommPort

ใช้ในการกำหนดและอ่านค่าพอร์ตอนุกรมที่ติดต่ออยู่ (COM1, COM2, COM3, COM4)

รูปแบบการใช้งาน

object.CommPort [= value]

โดย Value เป็นค่าของพอร์ตอนุกรม ชนิดของข้อมูลเป็น Integer ค่า Value สามารถกำหนดได้ในช่วง 1-16 (ค่าเริ่มต้นกำหนดไว้ที่ 1)เมื่อมีการกำหนดค่าแล้วทำการเปิดพอร์ตโดยใช้คุณสมบัติ PortOpen แต่ว่าพอร์ตนั้นไม่มีอยู่ในระบบ MSComm จะสร้างสัญญาณแสดงข้อผิดพลาด error 68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นมา ซึ่งหมายถึง อุปกรณ์ตัวนี้ไม่มีอยู่ในระบบ ดังนั้นการเขียนโปรแกรมจึงจำเป็นต้องกำหนดตำแหน่งของพอร์ตอนุกรมก่อนที่ใช้คำสั่ง OpenPort

Setting

ใช้ในการกำหนดและอ่านค่าอัตราบอด , พาริตี้, จำนวนบิตข้อมูล, จำนวนบิตปิดท้าย

รูปแบบการใช้งาน

object.Setting [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบ String มีรูปแบบเป็น "BBBB,P,D,S" โดย BBBBเป็นค่าอัตราบด , P เป็นค่าพาริตี้ ,D เป็นจำนวนของบิตข้อมูล และ S เป็นจำนวนของบิตปิดท้าย ปกติแล้วค่านี้ถูกกำหนดไว้เป็น "9600, N, 8, 1"

ค่าบิดเรตมาตรฐานที่ใช้กับ MSComm มีดังนี้

110 บิตต่อวินาที
300 บิตต่อวินาที
600 บิตต่อวินาที
1,200 บิตต่อวินาที
2,400 บิตต่อวินาที
9,600 บิตต่อวินาที (ค่าปกติ)
14,400 บิตต่อวินาที
19,200 บิตต่อวินาที
28,800 บิตต่อวินาที
28,400 บิตต่อวินาที
38,400 บิตต่อวินาที (สงวน)
56,000 บิตต่อวินาที (สงวน)
128,000บิตต่อวินาที (สงวน)
256,000บิตต่อวินาที (สงวน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับค่ามาตรฐานในการกำหนด ค่าพาริตี้มีดังนี้

สัญลักษณ์	**รายละเอียด**
E	พาริตี้คู่ (Even)
M	ลอจิก “1” (Mark)
N	ไม่ใช้ (ค่าปกติ)
O	พาริตี้คี่ (Odd)
S	ลอจิก “0” (Space)

ค่าที่ใช้ในการกำหนดจำนวนบิตมี 5 ค่าคือ 4,5,6,7 และ 8 **(เป็นค่าปกติ)**

ค่าที่ระบุจำนวนบิตปิดท้าย มี 3 ค่าคือ **1 (เป็นค่าปกติ),**1.5และ 2

ตัวอย่างการใช้งานคำสั่ง Setting โดยจะเป็นการกำหนดค่าบิดเรตเท่ากับ 9600 ไม่มีพาริตี้ จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต สามารถเขียนโปรแกรมได้ดังนี้

MSCONN1.Setting = “9600 , N , 8 , 1 “

หมายเหตุ สามาเหตุที่ค่าที่กำหนดจะต้องอยู่ภายในเครื่องหมายคำพูด “ ” เนื่องจากค่าที่กำหนดนี้อยู่ในรูปตัวแปร String

PortOpen

ใช้ในการกำหนดและอ่านค่าสถานะของพอร์ตอนุกรม เพื่อเปิดและปิดพอร์ตอนุกรม

รูปแบบการใช้งาน

object.PortOpen [= value]

ค่า Value มีชนิดข้อมูลเป็นแบบบูลีน คือ True กับ False โดย True หมายถึงการเปิดพอร์ตอนุกรมและ Flaseหมายถึงการปิดพอร์ตอนุกรม สำหรับการปิดพอร์ตนั้นจะมีการเคลียร์บัฟเฟอร์รับข้อมูลด้วย คอนโทรล MSComm จะปิดพอร์ตอนุกรมโดยอัตโนมัติเมื่อออกจากโปรแกรม ก่อนที่จะใช้คุณสมบัติ PortOpen ต้องตรวจสอบให้แน่ใจก่อนว่าคุณสมบัติ CommPort นั้นได้ทำการกำหนดตำแหน่งของพอร์ตอนุกรมไว้ถูกต้องหรือไม่ มิเช่นนั้น MSComm จะแสดงข้อผิดพลาด Error 68 แจ้งแก่ผู้ใช้งาน หรือถ้าพอร์ตอนุกรมนั้นถูกเปิดเอาไว้แล้ว โปรแกรมก็จะแจ้งข้อผิดพลาดออกมาเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าคุณสมบัติ DTREnable หรือ RTSEnable ถูกกำหนดให้เป็น True ก่อนที่จะทำการเปิดพอร์ต ค่าคุณสมบัติของ DTREnable หรือ RTSEnable จะถูกเซตเป็น False หลังจากปิดพอร์ตแต่ถ้าเซตเป็น False หลังจากปิดโปรแกรมแล้ว ค่าที่กำหนดไว้จะเป็นค่าเดิม

ตัวอย่างการใช้คำสั่งเปิดพอร์ต เพื่อติดต่อสื่อสารกับพอร์ตอนุกรม COM1 และมีบอดเรต 9,600 บิตต่อวินาที ไม่มีพาริตี้ จำนวนบิตข้อมูล 8 บิต และบิตปิดท้าย 1 บิต มีดังนี้

```
MSComm1.Setting = "9600,n,8,1 "
MSComm1.CommPort  = 1
MSComm1.PortOpen    = True
```

Input

อ่านค่าและลบค่าขบวนข้อมูลจากบัฟเฟอร์ภาครับ

รูปแบบการใช้งาน

object . Input

คุณสมบัติ InputLen เป็นตัวกำหนดจำนวนของตัวอักษรที่จะอ่านโดยคุณสมบัติ Input การกำหนดค่าให้ InputLen เท่ากับ 0 เป็นการกำหนดให้คุณสมบัติ Input ทำให้การอ่านค่าข้อมูลในบัฟเฟอร์รับข้อมูลทั้งหมด

คุณสมบัติ InputMode เป็นตัวกำหนดชนิดของข้อมูลที่มีคุณสมบัติ Input รับเข้ามา ถ้า InputMode ถูกกำหนดเป็น comInputModeText คุณสมบัติ Input จะส่งค่าข้อมูลกลับมาในรูปแบบของข้อความชนิดข้อมูลเป็นแบบ Variant ถ้า InputMode กำหนดเป็น comInputModeBinary คุณสมบัติ Input จะส่งข้อมูลกลับมาในรูปของไบนารี่และชนิดข้อมูลเป็นแบบ Variant

ตัวอย่างโปรแกรมแสดงให้เห็นถึงวิธีในการรับข้อมูลจากบัฟเฟอร์ข้อมลทั้งหมด

```
Private Sub Commandl _Click ()
Dim InString as String
        MSComm1.InputLen  =  0                  Retrieve all available data.
        If MSComm1.InBufferCont Then Check for data
        InString  = MSComm1.Input               Read data
        End  If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

InbufferCount

ส่งค่าจำนวนของตัวอักษรที่อยู่ในบัฟเฟอร์ภาครับ

รูปแบบการใช้งานคำสั่ง

object.InBufferCount [= value]

คำสั่ง InBufferCount จะแสดงค่าจำนวนของตัวอักษร ซึ่งรับมาจากภายนอกและยังเก็บอยู่ในบัฟเฟอร์ภาครับ เพื่อให้ผู้ใช้งานอ่านค่าออกไป สำหรับการเคลียร์ค่าบัฟเฟอร์ภาครับทำได้โดยกำหนดให้ InBufferCount มีค่าเป็น 0

หมายเหตุ อย่างสับสนระหว่าง คำสั่ง InBufferSize และ InBufferCount คำสั่ง InBufferSizeนั้นใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ

InBufferSize

กำหนดและคืนค่าขนาดของบัฟเฟอร์ภาครับในหน่วยเป็นไบต์

รูปแบบการใช้งานคำสั่ง

object . InBufferSize [= value]

คำสั่ง InBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาครับ โดยค่าเริ่มต้นถูกกำหนดไว้ที่ 1,024ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาครับขนาดใหญ่จะทำให้ หน่วยความจำที่เหลือสำหรับการใช้งานส่วนอื่นๆจะเหลือน้อย อย่างไรก็ตามการกำหนดค่า บัฟเฟอร์ภาครับที่น้อยเกินไปจะทำให้เกิดการโอเวอร์โฟลวหรือข้อมูลล้นบัฟเฟอร์ เว้นแต่จะมีการใช้เฮนด์เช็ก ดังนั้นค่าปานกลางที่เหมาะสมก็คือ ค่า 1,024 ซึ่งเป็นค่าเริ่มต้นนั้นเอง แต่ถ้าโปรแกรมมีการเกิดโอเวอร์โฟลวแล้วจึงค่อยปรับเพิ่มค่าขนาดของบัฟเฟอร์ให้มีค่าเพิ่มมากขึ้น

```
Private Sub Command1_Click ()
Dim Buffer as Variant
Dim Arr () as Byte
MSComm1.CommPort  = 1 Set and open port
MSComm1.PortOpen = True
MSComm1.InputMode = comInputModeBinary   set InputMode to read binary data
Do Unit MSComm1.InBufferCoint < 10       Wait unit  10 bytes are in the
Input buffer
        DoEvents
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Loop
Buffer = MSComm1.Input                    Store binary data in buffer
Arr = Buffer                              Assign to byte array for processing
End Sub
```

Output

ใช้ในการส่งขบวนของข้อมูลไปยังบัฟเฟอร์ส่งข้อมูล

รูปแบบการใช้งาน

object.Output [= value]

ค่า value เป็นค่าของตัวอักษรที่เขียนไปยังบัฟเฟอร์ส่งข้อมูล คุณสมบัติ OutPut สามารถใช้ในการส่งข้อมูลตัวอักษรหรือข้อมูลไบนารี่ก็ได้ โดยการส่งข้อมูลเป็นรูปแบบตัวอักษรจะต้องกำหนดข้อมูลเป็นแบบ Variant และมีข้อมูลภายในเป็นแบบ byte

ตัวอย่างโปรแกรมการส่งค่าที่ป้อนจากคีย์บอร์ดไปยังพอร์ตอนุกรม โดยใช้คุณสมบัติ Output

```
Private Sub Form_KeyPress (keyAscii As Integer)
    Dim Buffer as Variant
    MSComm1.ComPort = 1 Use COM1
    MSComm1.PortOpen    =True   Open port
    Buffer = Chr$ (KeyAscii)
    MSComm1.Output = Buffer     Send DATA
End Sub
```

OutBufferCount

คืนค่าจำนวนข้อมูลตัวอักษรที่เก็บอยู่ในบัฟเฟอร์ภาคส่ง และสามารถใช้คำสั่งนี้เพื่อเคลียร์บัฟเฟอร์ภาคส่งได้ด้วย

รูปแบบการใช้งานคำสั่ง

object.OutBufferCount [= value]

ผู้ใช้งานสามารถเคลียร์บัฟเฟอร์ภาคส่งได้โดยกำหนดค่า OutBufferCount เท่ากับ “0”

หมายเหตุ ระวังการสับสนระหว่างคุณสมบัติ OutBufferCount กับ OutBufferSuze ซึ่ง OutBufferSize ใช้เพื่อกำหนดขนาดของบัฟเฟอร์ภาคส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OutBufferSize

กำหนดค่าและคืนค่าขนาดของบัฟเฟอร์ภาคส่ง ชนิดตัวแปรเป็นแบบไบต์

รูปแบบการใช้งานคำสั่ง

object.OutBufferSize [= object]

คุณสมบัติ OutBufferSize ใช้สำหรับกำหนดขนาดของบัฟเฟอร์ภาคส่ง โดยปกติที่ใช้งานจะมีค่าเท่ากับ 512 ไบต์

หมายเหตุ การกำหนดค่าบัฟเฟอร์ภาคส่งที่มากเกินไป จะทำให้มีหน่วยความจะเหลือให้ใช้งานน้อย แต่อย่างไรก็ตามถ้ากำหนดค่าน้อยเกินไป จะทำให้เกิดข้อมูลล้นบัฟเฟอร์ขึ้นได้ ยกเว้นจะมีการใช้ แฮนด์เช็ค วิธีการที่ถูกต้องในการกำหนดค่าคือ ทดลองใช้ค่าเริ่มต้นคือค่า 512ไบต์ดูก่อน ถ้าโปรแกรมทำงานแล้วเกิดการล้นของข้อมูลค่อยเพิ่มค่าของ OutBufferSize ให้มากขึ้น

เหตุการณ์ OnComm

เหตุการณ์ OnComm จะถูกสร้างขึ้นเมื่อค่าของคุณสมบัติ CommEvent มีการเปลี่ยนแปลงเพื่อแสดงผลการเปลี่ยนแปลงเหล่านั้นแบบทันทีทันใดหรือแสดงข้อผิดพลาดที่เกิดขึ้น ตัวอย่างโปรแกรมย่อย OnComm เพื่อนำเหตุการณ์ CommEventมาแสดง

```
Private Sub MSComm_OnComm ()
        Select Case MSComm1.CommEvent
        ' Handle each event or error by placing
        ' code below each case statement
        ' Error
                Case  comEventBreak             ' A Break was received.
                Case comEvenCDTO                ' CD (RLSD)Timeout.
                Case comEvenCTSTO               ' CTS Timeout .
                Case comEvenDSRTO               ' DSR Timeout .
                Case comEvenFrame               ' Framing Error .
                Case comEvenOverrun             ' Data Lost .
                Case comEvenRxOver              ' Receive buffer overflow .
                Case comEvenRxParity            ' Parity Error .
Case comEventTxFull                             ' Transmit buffer full .
        ' Events
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Case comEvCD                ' Change in the CD line .
        Case comEvCTS               ' Change in the CYS line .
        Case comEvDSR               ' Change in the DSR line .
        Case comEvRing              ' Change in the Ring Indicator .
        Case comEvReceive           ' Receivd Rthreshold # of chars .
        Case comEvSend          ' Sthreshold number in the 'transmit buffer.
Case comEvEof                       ' An EOF charater was found in the input stream
    End Select
End Sub
```

การใช้ MSComm เพื่อการติดต่อฮาร์ดแวร์

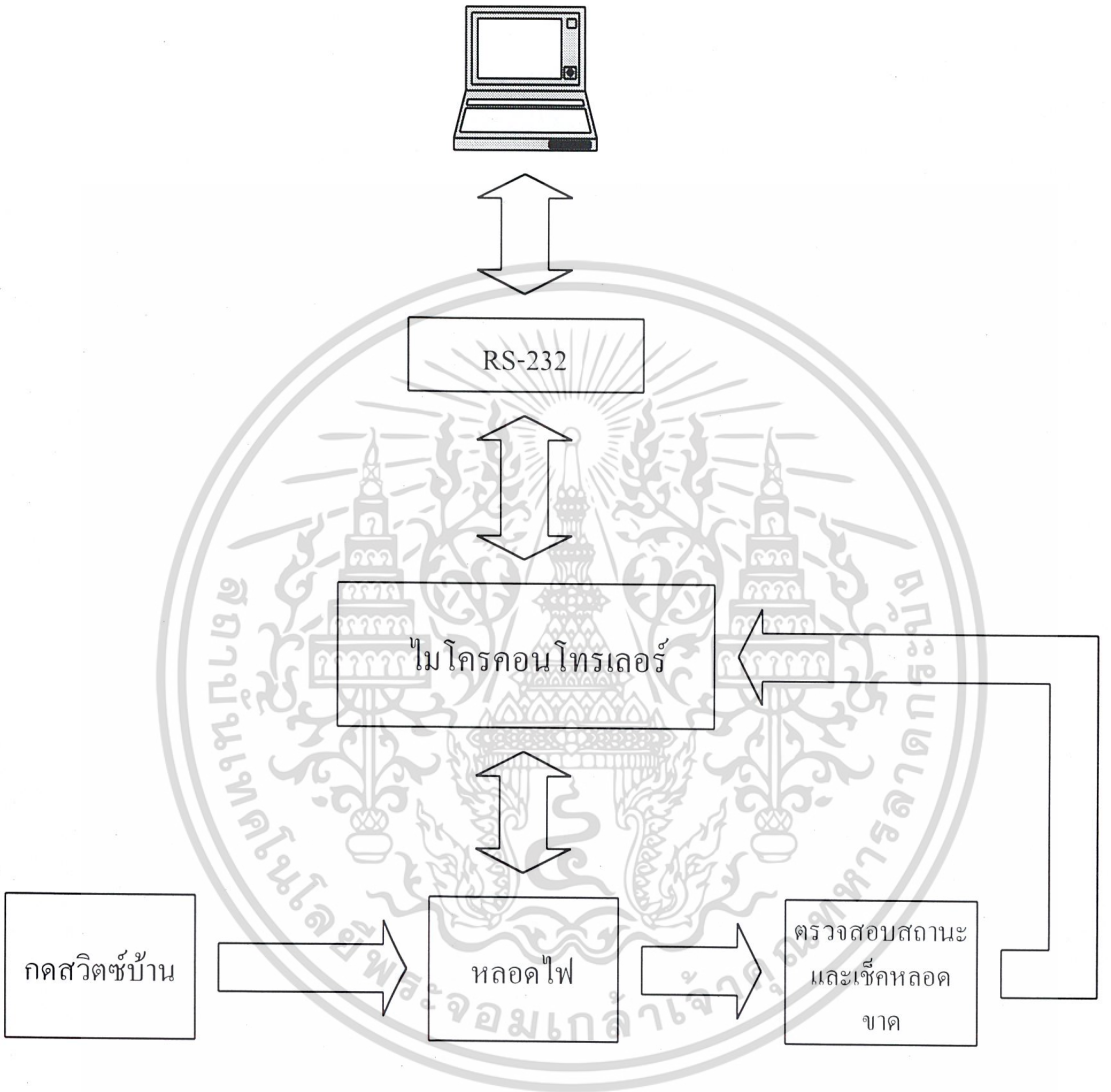
จากรายละเอียดของ MSComm ที่กล่าวไปในตอนต้นนั้น จะเห็นได้ว่าวิธีการที่จะอ่านค่าหรือเขียนค่าไปยังขาสถานะและขาควบคุมของพอร์ตอนุกรมสามารถทำได้ง่ายดายมาก โดยใช้คำสั่งเหล่านี้

DTREnable	สำหรับคำสั่งให้ขา DTR มีลอจิก "0" หรือ "1"
RTSEnable	สำหรับสั่งให้ขา RTS มีลอจิก "0" หรือ "1"
CTSHolding	สำหรับอ่านค่าสถานะจากขา CTS ว่ามีลอจิก "0" หรือ "1"
CDHolding	สำหรับอ่านค่าสถานะจากขา DCDว่ามีลอจิก "0" หรือ "1"
DSRHolding	สำหรับอ่านค่าสถานะจากขา DSR ว่ามีลอจิก "0" หรือ "1"
Break	สำหรับการสั่งให้ขา Txd มีลอจิก "0" หรือ "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 บล็อกไดอะแกรมการเชื่อมต่อ Visual Basic กับอุปกรณ์ภายนอก

โครงงานนี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ AT9038535 (AVR) ซึ่งการทำงานของไมโครคอนโทรลเลอร์มีส่วนประกอบดังบล็อกไดอะแกรมดังรูปที่ 4.10



รูปที่ 4.10 บล็อกไดอะแกรมในส่วนของไมโครคอนโทรเลอร์

การทำงานจะแบ่งเป็น 2 ส่วน คือ

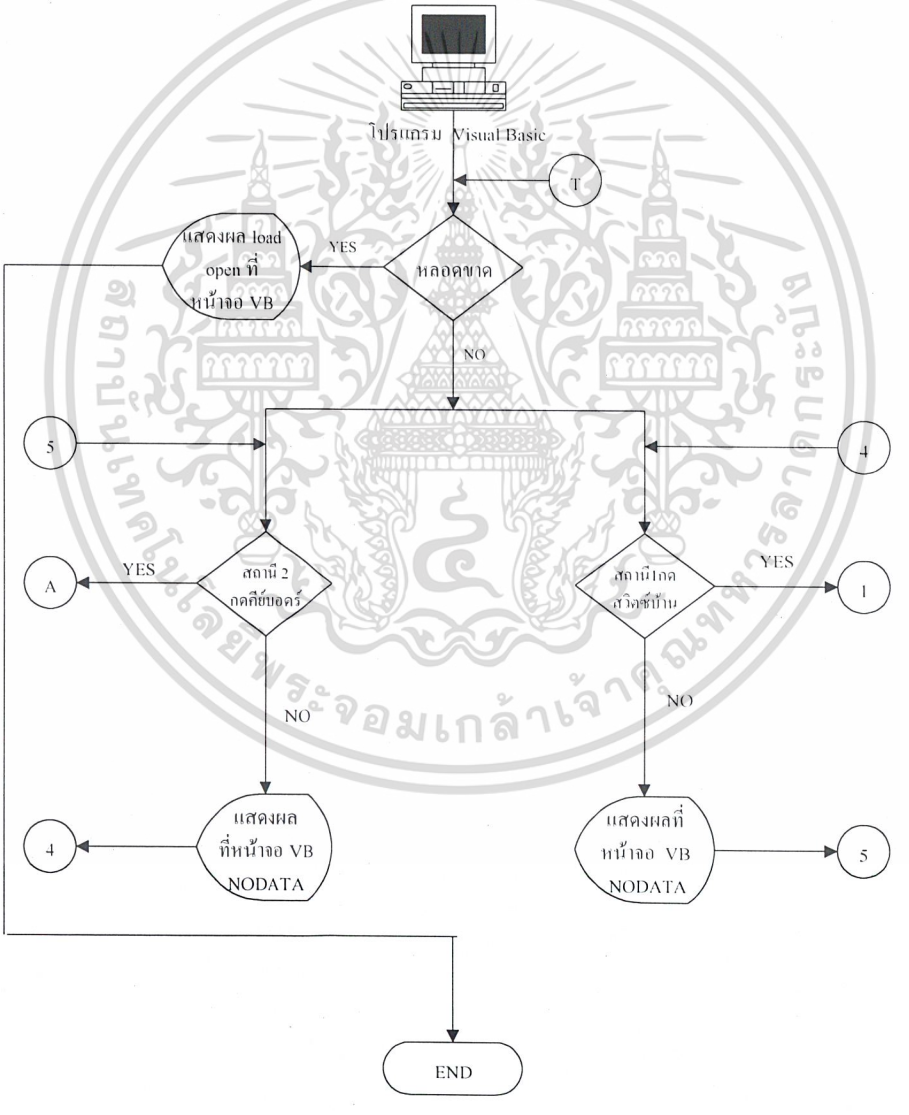
1. จากการจำลองคีย์บอร์ดจาก Computer ให้เป็นปุ่มโทรศัพท์มือถือ สั่งงานผ่านคอมพิวเตอร์โดยมีโปรแกรม Visual Basic จำลองการรับส่งค่าผ่านพอร์ตอนุกรม RS-232 สั่งงานให้ไมโครคอนโทรลเลอร์ (AVR) ไปควบคุมหลอดไฟให้สามารถเปิด, หรี่ไฟและดับไฟได้ จากนั้นวงจรตรวจสอบสถานะและเช็คหลอดไฟขาดก็ทำการตรวจเช็คสภาวะและสั่งค่ากลับไปยังไม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครคอนโทรลเลอร์ทำการประมวลผลและส่งค่าผ่านพอร์ตอนุกรม มาเข้าสู่จอ Computer ซึ่งเราสามารถรู้สถานะการเปิด, หรี่ไฟและดับได้ จากโปรแกรม Visual Basic

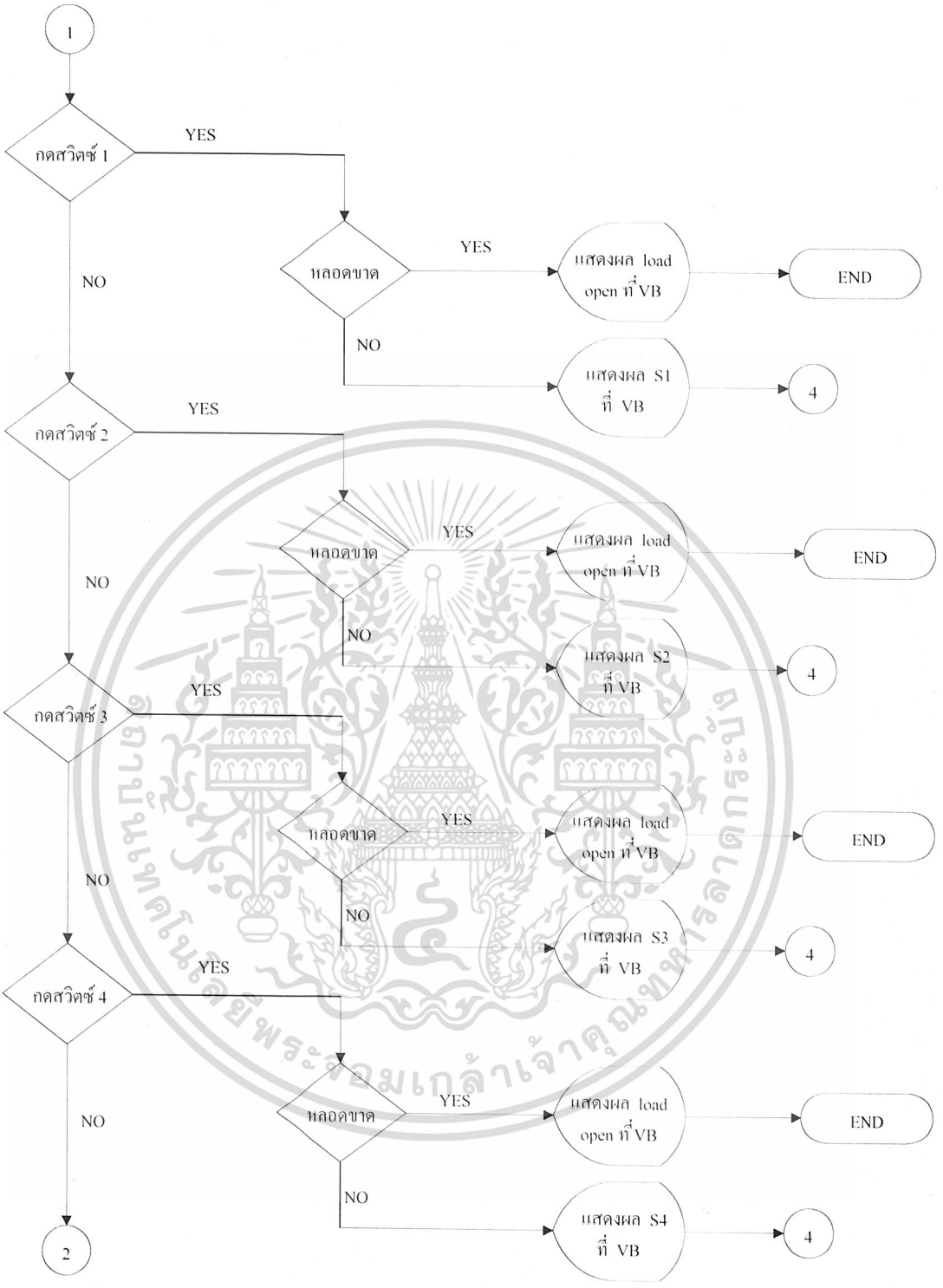
2. จากการกดสวิตซ์สมมุติถ้ามีการกดสวิตซ์ที่บ้านเกิดขึ้น หลอดไฟก็จะสว่างและจะมีวงจรตรวจสอบสถานะว่าหลอดติด, ดับ, หรี่ หรือถ้ามีการขาดของหลอดวงจรก็จะตรวจเช็คและส่งค่ากลับไปยัง ไมโครคอนโทรลเลอร์ (AVR) ผ่าน RS-232 กลับไปยัง Computer และดูสถานะต่าง ๆ ได้จาก Visual Basic

4.5.1 Flow Chart การทำงานในส่วนของ Visual Basic ร่วมกับไมโครคอนโทรลเลอร์



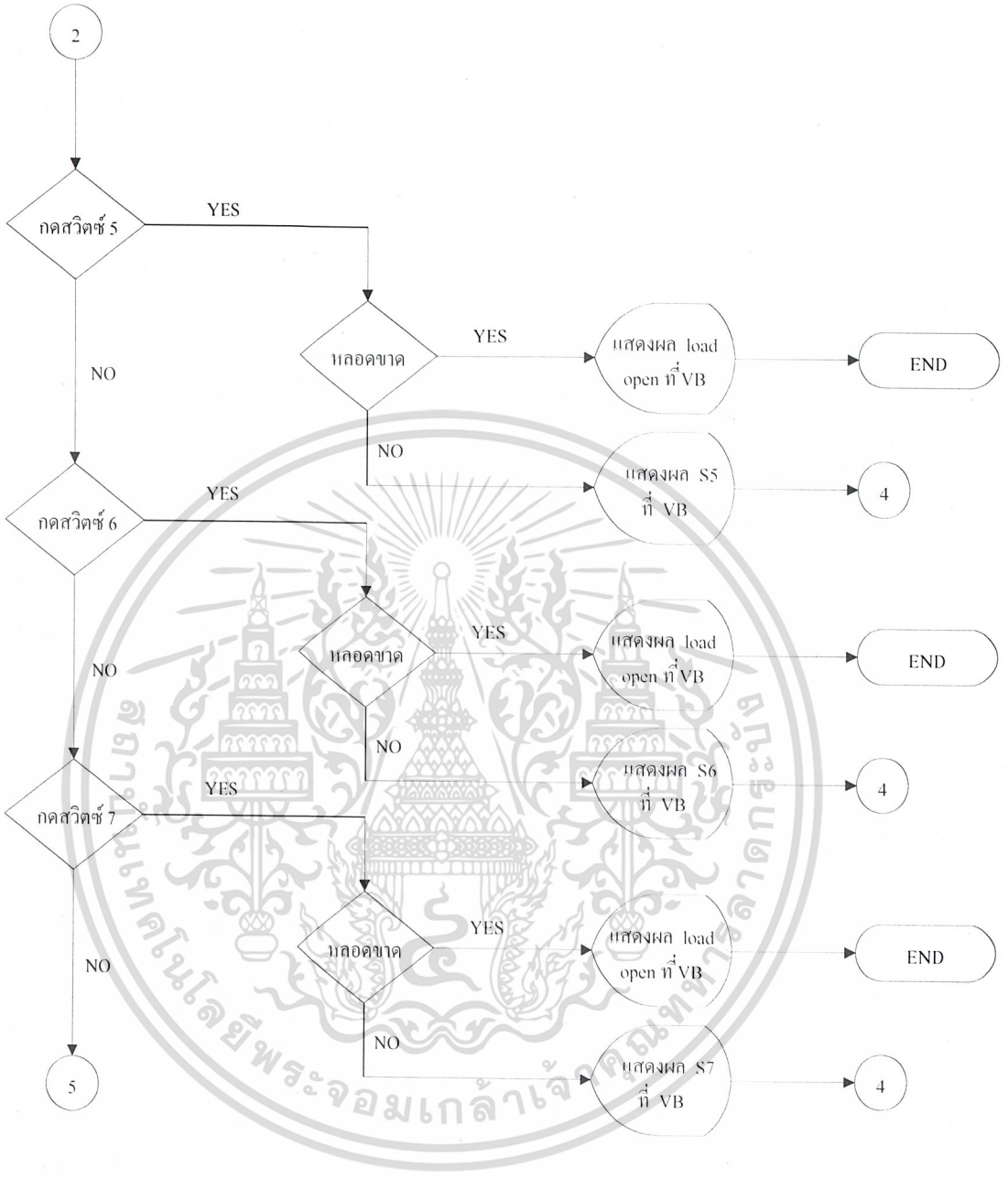
รูปที่ 4.11 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



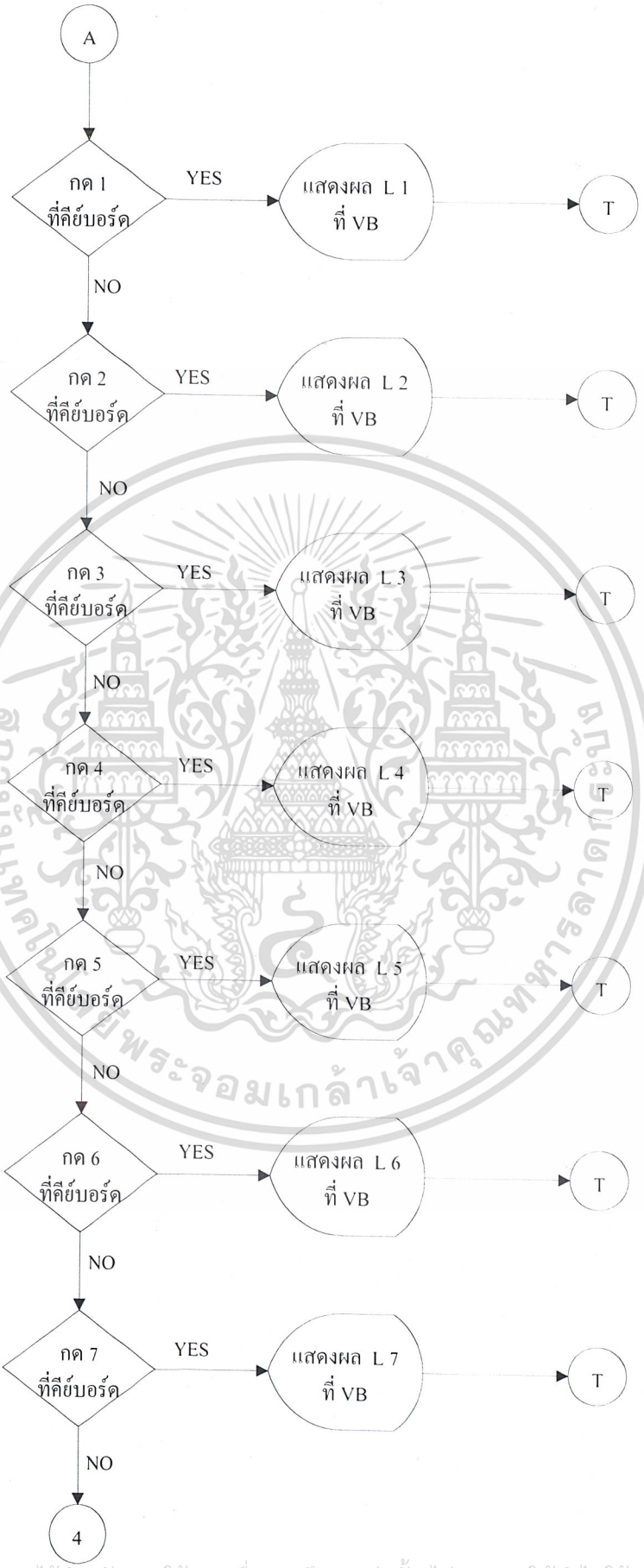
รูปที่ 4.12 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.14 Flow Chart การทำงานของ Visual Basic ร่วมกับ ไมโครคอนโทรลเลอร์ (4)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 วงจรการทำงานของไมโครคอนโทรเลอร์

หลักการทำงาน

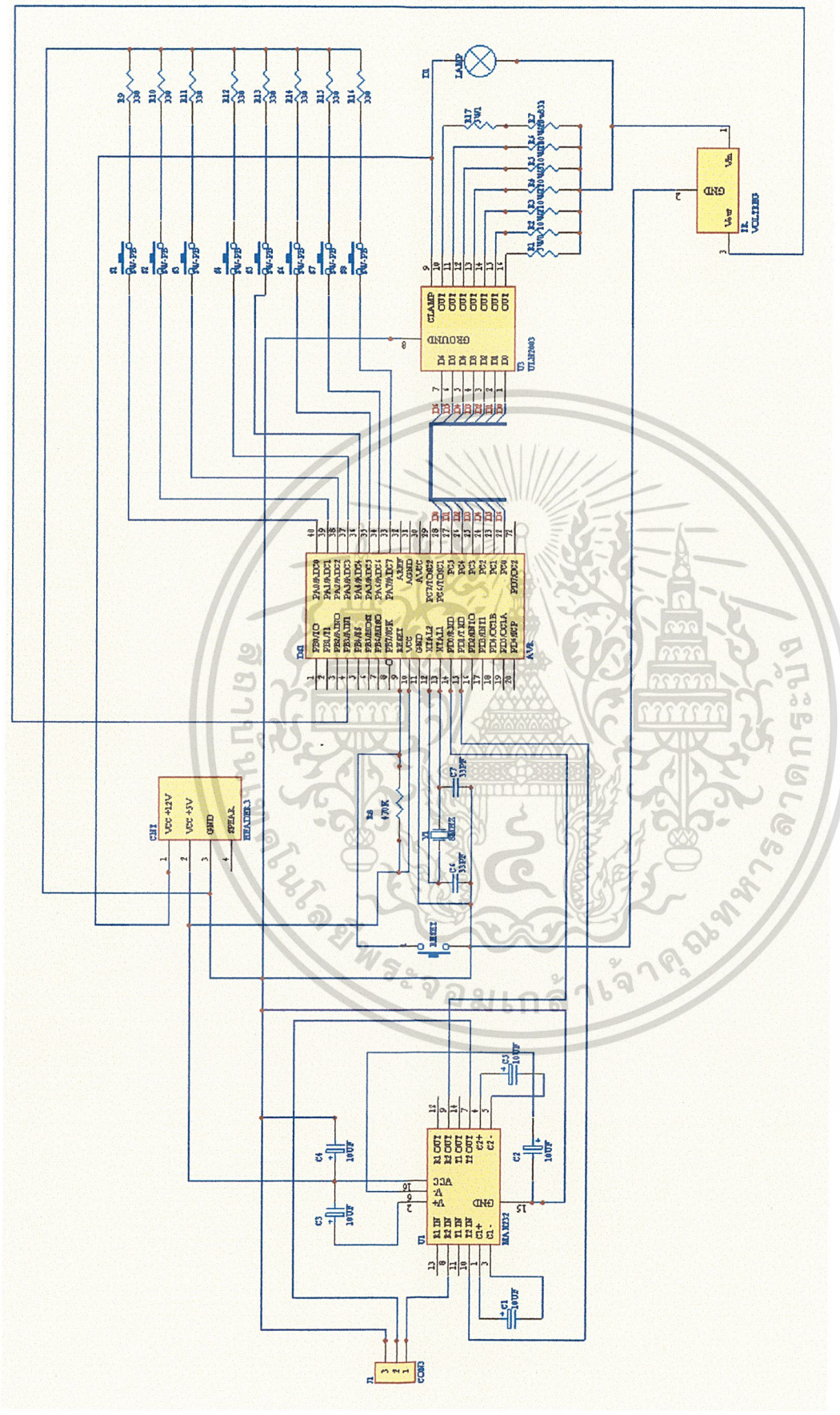
ไมโครคอนโทรเลอร์ AVR (AT 90S8535) ก็จะรับข้อมูลข้อมูลเข้ามาที่ขา 14 และ ขาที่ 15 โดยที่ขา 14 เป็นขาที่ รับข้อมูล (RXD) และขาที่ 15 เป็นขาส่งข้อมูล (TXD) แล้วนำข้อมูลไปตรวจสอบว่าข้อมูลที่รับได้ส่งควบคุมอุปกรณ์ไฟฟ้าหรือสัญญาณใดหลังจากนั้นไมโครคอรโทรเลอร์ AVR จะทำงานสามอย่างคือ

1. ส่งข้อมูลควบคุมออกไปที่ พอร์ต C เพื่อควบคุมอุปกรณ์หลอดแสงสว่าง ให้มีการติดดับและสามารถหรี่ไฟแต่ละระดับได้ โดยใช้คีย์บอร์ดคอมพิวเตอร์ควบคุมแทนโทรศัพท์มือถือ

2. ส่งข้อมูลออกทาง พอร์ต A เพื่อควบคุมอุปกรณ์หลอดแสงสว่าง ให้มีการติดดับและสามารถหรี่ไฟเช่น กัน แต่จะใช้ สวิตซ์ ในการควบคุมแทน

3. ส่วนของการเช็คหลอดขาด จะส่งข้อมูลออกไปที่พอร์ต A เพื่อควบคุมว่าหลอดขาดหรือไม่ และจะส่งข้อมูลกลับไปยัง AVR เพื่อประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 วงจรไมโครคอนโทรลเลอร์และการต่อร่วมกับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและผลการทดลอง

5.1 การทดลองเขียน WAP Site โดยใช้ภาษา WML ร่วมกับ PHP

การทดลองในส่วนนี้จะเป็นการทดลองโดยการเขียน WAP Site ขึ้นมา โดยใช้ภาษา WML ร่วมกับ PHP โดยในการทดลองนี้จำเป็นต้องใช้ Server เพื่อใช้ในการเก็บ WAP Site ก่อนที่จะนำมาประมวลผล สำหรับในการทดลองนี้ได้ใช้โปรแกรม Apache Web Server เป็น Server ที่ใช้งาน โดยที่ในการใช้งานจริงจะต้องทำการเก็บไฟล์ที่ทำการเขียนเรียบร้อยแล้ว นำไปใส่ไว้ในโฟลเดอร์ของโปรแกรมเสียก่อน เมื่อต้องการแสดงผลต้องทำการเปิด Server แล้วใช้ WAP Browser ทำการเรียก IP Address ของ Server ที่ต้องการ เมื่อมีการเรียก IP Address ของ Server แล้ว Server จะทำการประมวลผล WAP Site ที่เรียกมาผ่าน IP Address โดยจะทำการประมวลผลภาษา PHP เมื่อประมวลเสร็จแล้ว Serverจะทำการส่ง WAP Site ที่ทำการประมวลผลเสร็จสิ้นแล้วมาให้ยัง WAP Browser

สำหรับในการทดลองนี้ได้จำลองการเล่น WAP ผ่านระบบโดยอินเตอร์เน็ต โดยใช้โปรแกรม Nokia Mobile Internet Toolkit เวอร์ชั่น 3 ในการจำลองการทำงานโดยเริ่มจากการป้อน IP Address ของ WAP Server คือ http://161.246.26.254/ ซึ่งข้อมูลที่ใช้ในการทดลองได้บันทึกไว้เป็นไฟล์ตัวอักษรเก็บไว้ใน Server เรียบร้อยแล้ว ข้อมูลที่อยู่ใน Server จะถูกเก็บไว้เป็นรหัสไว้ดังตารางที่ 5.1 โดยรหัสนี้จะถูกถอดรหัสเพื่อมาแสดงสถานะโดยใช้ภาษา PHP และนำมาแสดงบน WAP Browser

ตารางที่ 5.1 รหัสและสถานะของอุปกรณ์ที่จะนำมาแสดงผล

รหัส	สถานะ
L1	Very high
L2	High
L3	Medium
L4	Medium low
L5	Low
L6	Very low
L7	Off line
z	Load open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของการแสดงผลที่ WAP Browser ได้มีการแบ่งการทำงานของ WAP Site ไว้ดังนี้

1) หน้าจอหลัก เมื่อผู้ใช้ทำการเรียก IP Address ของ WAP Site WAP Browser จะทำการร้องขอข้อมูลจาก WAP Server เมื่อ WAP Server ประมวลผล WAP Site เรียบร้อยแล้ว ก็จะส่งข้อมูลกลับมาให้ WAP Browser และปรากฏหน้าจอนี้ขึ้นมาเพื่อแสดงว่าพร้อมที่จะทำงานแล้ว หน้าจอนี้เป็นหน้าจอก่อนจะเข้าสู่เมนูหลัก แสดงดังรูปที่ 5.1



รูปที่ 5.1 หน้าจอการเข้าสู่ WAP Site

2) หน้าจอให้ผู้ใช้กดคีย์ เมื่อผู้ใช้เข้าสู่หน้าจอหลักแล้ว WAP Browser จะทำรอคอยจนครบเวลาที่กำหนดไว้ถ้าผู้ใช้ไม่มีการกดคีย์เพื่อเข้าสู่หน้าจอของการทำรายการ WAP Browser จะทำการเปลี่ยนหน้าจอเข้าจอสู่หน้าจอให้ผู้ใช้กดคีย์เพื่อร้อยขอให้ผู้ใช้กดคีย์เพื่อทำรายการต่อ แสดงดังรูปที่ 5.2



รูปที่ 5.2 หน้าจอให้ผู้ใช้กดคีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) หน้าจอการทำรายการหลัก ใช้แสดงในส่วนของรายการเพื่อให้ผู้ใช้เลือกทำรายการต่าง ๆ เมื่อผู้ใช้กดคีย์เพื่อเข้าสู่เมนูหลักจะปรากฏหน้าจอนี้ขึ้นที่ WAP Browser เพื่อให้ผู้ใช้เข้าสู่รายการที่แสดงไว้ใน ในรูปที่ 5.3 (ก) เป็นการเลือกรายการ Status ซึ่งเป็นการขอดูสถานะของหลอดไฟ ใน รูปที่ 5.3 (ข) เป็นการเลือกรายการ Control ซึ่งเป็นการทำงานในการควบคุมสถานะของเครื่องใช้หลอดไฟ ส่วนในรายการ Security เป็นแนวทางในการพัฒนาการทำงานของ WAP Site เพื่อที่จะสามารถตรวจดูสถาวะของอุปกรณ์หรือระบบรักษาความปลอดภัยภายในบ้าน ซึ่งในรายการนี้คือส่วนของแนวทางการพัฒนาการทำงานของระบบต่อไป



(ก) (ข)

รูปที่ 5.3 หน้าจอการทำรายการหลัก รูป (ก) หน้าจอการทำรายการ Status รูป (ข) หน้าจอการทำรายการ Control

4) หน้าจอแสดงสถานะของหลอดไฟ เมื่อผู้ใช้ทำการเลือกรายการ Status WAP Broser จะเข้าหน้าจอนี้ ซึ่งเป็นส่วนที่ใช้แสดงสถานะของหลอดไฟในการทดลอง โดยที่สถานะที่แสดงบนหน้าจอนี้จะมีระดับการแสดงผลดังตารางที่ 5.1 และจะแสดงผลขึ้นหน้าจอดังรูปที่ 5.4



(ก) (ข) (ค)

รูปที่ 5.4 หน้าจอแสดงสถานะของหลอดไฟ รูป (ก) หน้าจอสถานะ L2 รูป (ข) หน้าจอสถานะ L5 รูป (ค) หน้าจสถานะ L7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) หน้าจอแสดงการขอรหัสผ่าน เมื่อผู้ใช้ทำการกดคีย์เพื่อจะทำรายการ Control ซึ่งเป็นการทำงานในส่วนของการควบคุมเครื่องใช้ไฟฟ้า ก่อนจะเข้าสู่การทำงานในการควบคุมหลอดไฟจำเป็นต้องป้อนชื่อผู้ใช้ รวมทั้งรหัสผ่านให้ถูกต้องก่อน เมื่อผู้ใช้ทำการกดเลือก OK เรียบร้อยแล้ว WAP Site ส่งชื่อและรหัสผ่านมายัง WAP Server เพื่อประมวลผลว่ารหัสผ่านที่ผู้ใช้ทำงานส่งมาถูกต้องหรือไม่ ดังในรูปที่ 5.5 (ก) ซึ่งเป็นหน้าจอก่อนที่ผู้ใช้ทำการกรอกชื่อและรหัสผ่าน และในรูปที่ 5.5 (ข) เป็นหน้าจอที่ผู้ใช้ทำการใส่ชื่อและรหัสผ่านเรียบร้อยแล้ว



(ก) (ข)

รูปที่ 5.5 หน้าจอการขอรหัสผ่าน รูป (ก) หน้าจอที่ให้ผู้ใช้กรอกชื่อและรหัสผ่าน รูป (ข) หน้าจอที่ผู้ใช้กรอกชื่อและรหัสผ่านเรียบร้อยแล้ว

6) หน้าจอแสดงรหัสผ่านไม่ถูกต้อง เมื่อ WAP Server ทำการประมวลผลรหัสผ่านที่ผู้ใช้ส่งมาแล้วพบว่ารหัสผ่านไม่ถูกต้อง WAP Server จะส่งหน้าจอนี้ขึ้นเพื่อแสดงว่ารหัสผ่านที่ผู้ใช้ป้อนไม่ถูกต้อง แสดงดังรูปที่ 5.6



รูปที่ 5.6 หน้าจอรหัสผ่านไม่ถูกต้อง

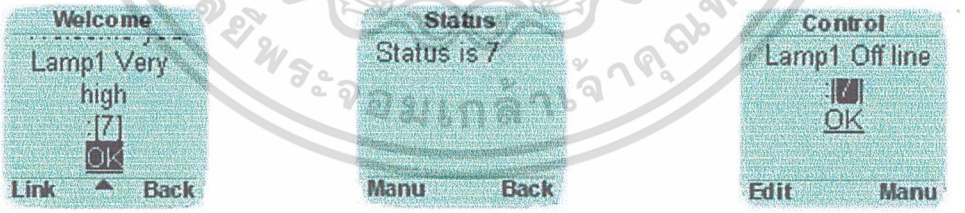
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) หน้าจอแสดงการเข้าสู่การควบคุมหลอดไฟ เมื่อ WAP Server รับชื่อและรหัสของผู้ใช้แล้วนำมาประมวลผล ถ้าพบว่ารหัสผ่านที่ผู้ใช้ส่งมานั้นถูกต้อง WAP Server จะทำการส่งข้อมูลมาให้กับ WAP Browser เพื่อนำมาแสดงผลที่หน้าจอ โดยที่หน้าจอแสดงผลจะบอกสถานะปัจจุบันของหลอดไฟ และให้ผู้ใช้ทำการส่งสถานะของหลอดไฟที่ต้องการเปลี่ยนแปลงกลับไปยัง WAP Server ดังรูปที่ 5.7



รูปที่ 5.7 หน้าจอการเข้าสู่การควบคุมหลอดไฟ

8) หน้าจอแสดงการป้อนข้อมูลที่หน้าจอ WAP Browser และ การตอบสนอง เมื่อทำการเข้าสู่หน้าจอของการควบคุมเครื่องใช้ไฟฟ้าเรียบร้อยแล้ว เมื่อผู้ใช้ทำการกดคีย์เพื่อป้อนสถานะใหม่ให้แก่หลอดไฟ ดังรูปที่ 5.8 (ก) WAP Server จะส่งข้อมูลกลับมายัง WAP Browser ว่าทำการส่งสถานะเรียบร้อยแล้ว ดังรูปที่ 5.8 (ข) และเมื่อกลับมาหน้าจอแสดงการควบคุมหลอดไฟอีกครั้ง WAP Browser จะแสดงผลดังรูปที่ 5.8 (ค)



(ก) (ข) (ค)

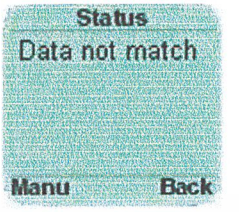
รูปที่ 5.8 หน้าจอของการป้อนข้อมูลและการตอบสนอง รูป (ก) การป้อนสถานะที่ใหม่ให้แก่หลอดไฟ รูป (ข) การตอบสนองการสั่งงานของ WAP Server รูป (ค) สถานะของหลอดไฟที่มีการปรับปรุงเรียบร้อยแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) หน้าจอแสดงการป้อนสถานะที่ผิดพลาด และ การตอบสนอง เมื่อผู้ใช้ทำการคีย์สถานะใหม่ที่ต้องการไม่ถูกต้อง ดังรูปที่ 5.9 (ก) จะมีการตอบสนองกลับจาก WAP Server โดยจะแสดงผลออกมาว่าสถานะที่ส่งมานั้นไม่ถูกต้องเพื่อให้ทำการใส่ค่าใหม่ลงไป ดังรูปที่ 5.9 (ข)



(ก)



(ข)

รูปที่ 5.9 หน้าจอแสดงการป้อนสถานะผิดพลาด รูป (ก) การป้อนสถานะที่ผิดพลาด รูป (ข) ผลการตอบสนองกลับจาก WAP Server

5.2 การทดลองการเขียนภาษา Visual Basic เพื่อใช้ติดต่อกับไมโครคอนโทรลเลอร์

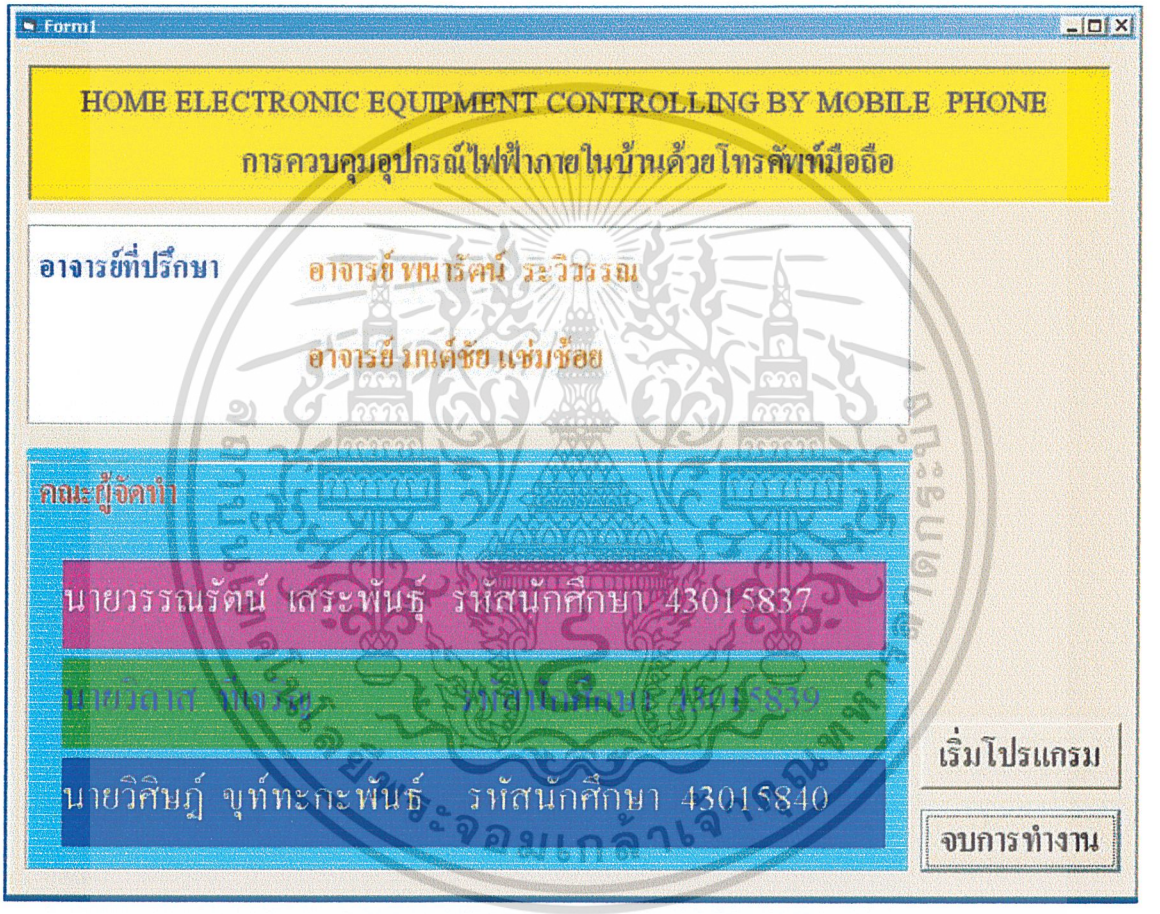
ในการทดลองนี้เป็นการทดลองในการใช้ภาษา Visual Basic ที่เขียนขึ้นสำหรับใช้ในการติดต่อกับหลอดไฟ เพื่อจะใช้ในการตรวจสอบสถานะของหลอดไฟ การสั่งงานผ่านโปรแกรม Visual Basic และให้นำข้อมูลแสดงสถานะที่ได้จากการติดต่อนี้นำไปเก็บไว้ยัง WAP Server เพื่อให้ WAP Server นำมาใช้ในการส่งข้อมูลแสดงสถานะให้กับ WAP Browser โดยที่โปรแกรมนี้จะทำการติดต่อกับไมโครคอนโทรลเลอร์ AVR ซึ่งเป็นอุปกรณ์ทางฮาร์ดแวร์ที่ใช้ในการติดต่อกับหลอดไฟอีกที และการสื่อสารระหว่างโปรแกรม Visual Basic กับ อุปกรณ์ไมโครคอนโทรลเลอร์ได้ใช้การติดต่อผ่านพอร์ตอนุกรม

เมื่อทำการเริ่มต้นการทำงานของโปรแกรมนี้จะต้องทำการติดตั้งอุปกรณ์ภายนอกให้เรียบร้อย เพื่อให้สามารถแสดงผลย้อนกลับมายังโปรแกรมได้ทันทีในส่วนของการทดลองนี้จะเป็นการควบคุมสถานะโดยใช้ Visual Basic การทำงานเริ่มต้นของโปรแกรมจะทำการดึงข้อมูลสถานะจาก WAP Server และนำมาสั่งงานอุปกรณ์ภายนอก ในส่วนของการโดยการสั่งงานผ่านโปรแกรม Visual Basic ต้องทำการป้อนข้อมูลลงไปที่หน้าจอ โปรแกรมจะทำการเก็บข้อมูลสถานะใหม่ลงไปใน WAP Server ทันทีเพื่อให้ข้อมูลถูกส่งยังหลอดไฟเพื่อแสดงสถานะที่โปรแกรมต้องการตลอดเวลา และในส่วนของการควบคุมสถานะด้วยมือก็จะทำโดยเมื่อมีการกดสถานะใหม่ ไมโครคอนโทรลเลอร์จะทำการส่งสถานะใหม่ไปให้กับโปรแกรม เมื่อโปรแกรมได้รับข้อมูลแล้วจะทำการเก็บข้อมูลสถานะใหม่ที่ WAP Server และจะทำการส่งสถานะกลับไปยังไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้ไมโครคอนโทรลเลอร์ทำการเปลี่ยนสถานะของหลอดไฟ การทำงานของโปรแกรมที่เขียนขึ้นนี้จะแบ่งการทำงานออกเป็นส่วนต่าง ๆ ดังนี้

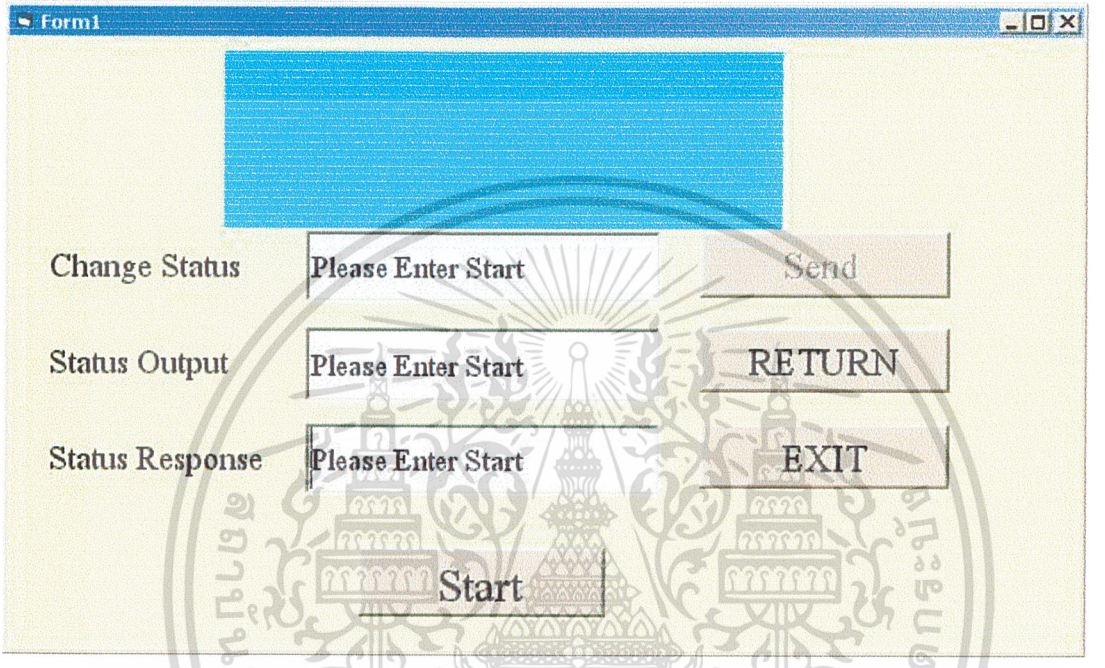
1) การเริ่มเข้าสู่โปรแกรม จะเริ่มต้นในการทำงานโดยจะกดเลือกที่ "เริ่มโปรแกรม" และยกเลิกการทำงานของโปรแกรมโดยกดที่ "จบการทำงาน"



รูปที่ 5.10 การเข้าสู่โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

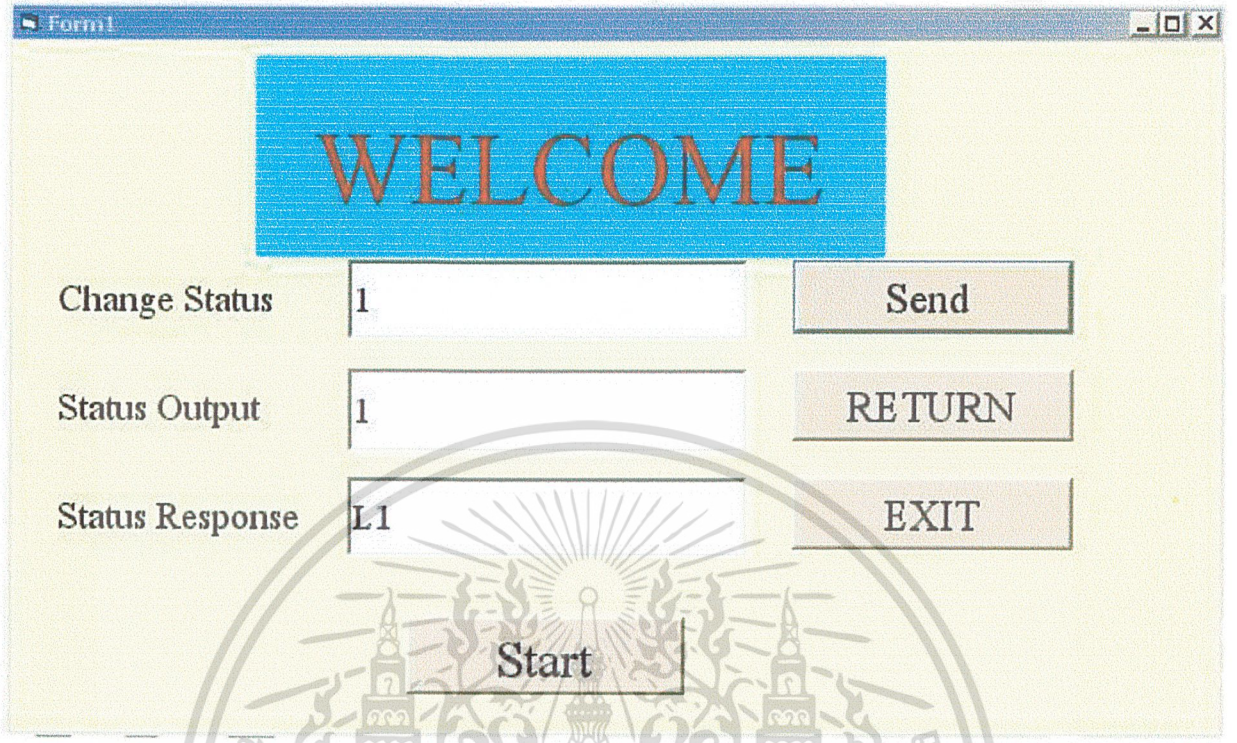
2) การเริ่มต้นของโปรแกรม เมื่อกดเลือก "เริ่มการทำงาน" จะเข้าสู่โปรแกรมในส่วนของการทำงานเปลี่ยนสถานะของอุปกรณ์ภายนอกโดยจะต้องทำการกดเลือก "Start" เพื่อให้ระบบเริ่มการทำงานได้



รูปที่ 5.11 การเริ่มต้นของโปรแกรมสถานะ

3) การเริ่มต้นทำงานของโปรแกรม การทำงานของโปรแกรมในส่วนนี้จะเริ่มขึ้นเมื่อมีการกดเลือกที่ "Start" แสดงดังรูปที่ 5.12 ซึ่งเป็นการแสดงการทำงานโดยที่โปรแกรมจะทำการดึงข้อมูลที่อยู่ใน WAP Server คือสถานะ "1" นำมาสั่งงานให้แก่หลอดไฟ และเมื่อไมโครคอนโทรลเลอร์ได้รับสถานะ "1" จาก Visual Basic ก็จะทำการสั่งงานหลอดไฟมีการเปลี่ยนแปลงสถานะเป็นสว่างสุดและไมโครคอนโทรลเลอร์ก็จะทำการส่งค่าสถานะสว่างสุดคือ "L1" กลับมาให้แก่โปรแกรมแล้ว โปรแกรมจะแสดงค่าที่ได้รับมาบนหน้าจอและเก็บค่าสถานะนี้ไว้ใน WAP Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.12 การเริ่มต้นการทำงานโดยที่สถานะเริ่มต้นอยู่ที่สถานะ "1"

4) การทำงานในด้านการควบคุมสถานะด้วยสวิตซ์ เมื่อมีการกดสวิตซ์ที่อุปกรณ์ภายนอก ในการแสดงผลนี้ยกตัวอย่างเป็นการกดสวิตซ์ที่สถานะ "4" ไมโครคอนโทรลเลอร์จะทำการส่งค่า "S4" มาให้แก่โปรแกรมดังรูปที่ 5.13 เมื่อโปรแกรมได้รับข้อมูลในการเปลี่ยนสถานะก็จะทำการส่งเก็บค่าไว้ยัง WAP Server และจะทำการส่งค่าสถานะ "4" กลับไปยังอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.13 การกดสถานะ "4" ที่อุปกรณ์ภายนอก

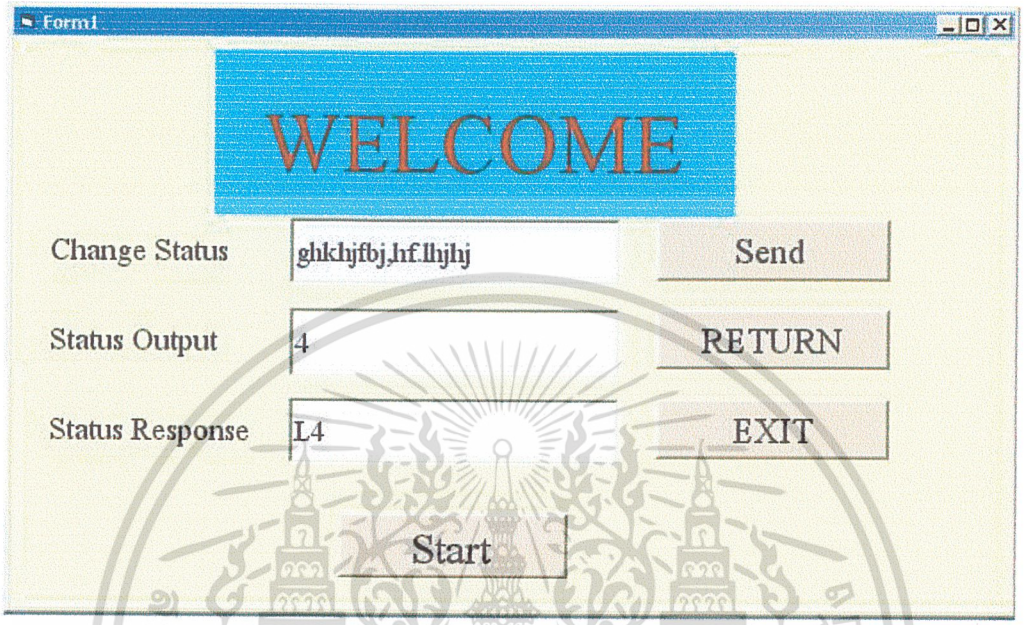
เมื่อมีการส่งข้อมูลไปสถานะ "4" ไปยังไมโครคอนโทรลเลอร์เรียบร้อยแล้วไมโครคอนโทรลเลอร์จะทำการเปลี่ยนสถานะของหลอดไฟและทำการส่งค่า "L4" กลับมายังโปรแกรม Visual Basic ดังรูปที่ 5.14



รูปที่ 5.14 การตอบสนองกลับของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การทำงานในด้านการป้อนข้อมูลผิดพลาด เมื่อมีการคีย์ข้อมูลสถานะและทำการเลือก "Send" โปรแกรมจะทำการเช็คข้อมูลสถานะที่ได้ว่าถูกต้องหรือไม่ถ้าไม่ถูกต้อง ดังรูปที่ 5.15 โปรแกรมจะทำการแสดงข้อความบอกผู้ใช้ว่ามีการป้อนข้อมูลผิดพลาดดังรูปที่ 5.16



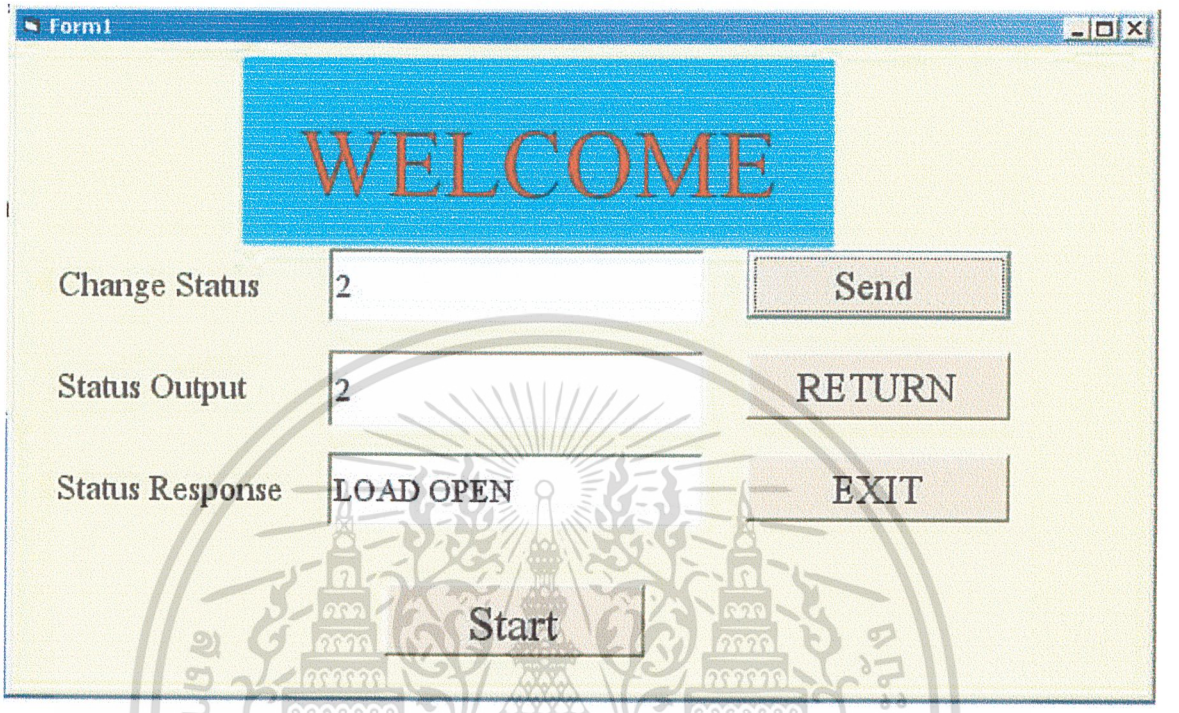
รูปที่ 5.15 การคีย์สถานะไม่ถูกต้อง



รูปที่ 5.16 โปรแกรมแสดงการเตือนการป้อนสถานะผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การทำงานในด้านการตรวจสอบหลอดขาด เมื่อเกิดปัญหาขึ้นกับหลอดไฟขาด ไมโครคอนโทรเลอร์จะส่งสถานะหลอดขาดมาให้กับโปรแกรม ดังรูปที่ 5.17



รูปที่ **5.17** การแสดงหลอดขาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทดลองในส่วนของ Hard ware

5.3.1การติดต่อระหว่าง Visual Basic กับ ไมโครคอนโทรลเลอร์

จากการทดลองส่งค่าจาก VB ควบคุมผ่าน RS 232 ให้ไมโครคอนโทรลเลอร์ (AVR) ที่พอร์ต C รับค่าและส่งค่าไปควบคุมชุดระดับของการเปิด ปิดและหรี่ไฟ โดยการส่งค่าที่ต้องการออกจาก VB จากนั้นทำการตรวจสอบสภาวะการทำงานของการหรี่ไฟของแต่ละระดับสภาวะยังคงทำงานได้อย่างถูกต้อง จากนั้นให้ไมโครคอนโทรลเลอร์ AVR รับค่าจากการเปิดปิดหรี่ไฟที่ Port D ทำการส่งค่าผ่าน RS-232 ให้คอมพิวเตอร์รับค่าออกมาแสดงผลทางหน้าจอของโปรแกรม VB จากการทดลองโปรแกรม VB สามารถรับค่าจากไมโครคอนโทรลเลอร์ (AVR) ได้อย่างถูกต้อง

5.3.2 การทำงานของส่วนอุปกรณ์ภายนอก

5.3.2.1 ชุดความต้านทาน

การทำงานของชุดความต้านทานจะทำตามการทำงานของไมโครคอนโทรลเลอร์ AVR โดยสั่งงานผ่าน IC ULN 2003 จากการวัดค่าแรงดันที่คร่อมหลอดไฟระดับแรงดันไฟสูงสุดที่หรี่ไฟได้สูงสุดได้ประมาณ 8 V และต่ำที่สุด3.2 V จากการทดลองการทำงานของชุดความต้านทานทำงานได้อย่างถูกต้อง

5.3.2.2 ชุดตรวจสอบสถานะและหลอดขาด

การทำงานของชุดตรวจสอบสถานะจะทำงานตามการทำงานของไมโครคอนโทรลเลอร์ที่ได้เขียนโปรแกรมไว้และการตรวจเช็คหลอดขาดจะใช้ชุด ไอซี 7805 เป็นตัวเช็คระดับแรงดันคือถ้าแรงดันที่ Port Aเป็น ลอจิก 0 จะเช็คว่าหลอดขาดแต่ถ้าเป็นลอจิก 1 ก็จะทำงานตามปกติจากการทดสอบการทำงานของชุดตรวจสถานะและหลอดขาดสามารถทำงานได้ตามต้องการ

5.4 โปรแกรมทดสอบไมโครคอนโทลเลอร์ (AVR)

จะแบ่งออกเป็นส่วนของโปรแกรมรับข้อมูลจากชุดตรวจสอบสถานะและการควบคุมด้วยมือ และโปรแกรมรับส่งข้อมูลจากพอร์ตอนุกรม ในส่วนของโปรแกรมรับข้อมูลจากชุดตรวจสอบสถานะและการควบคุมด้วยมือ ไมโครคอนโทรลเลอร์จะรับข้อมูลสถานะจากอุปกรณ์ภายนอกเข้ามาทางพอร์ต C ที่ขา 22 ถึง 28 การควบคุมด้วยมือไมโครคอนโทรลเลอร์จะรับข้อมูลการสั่งงานด้วยมือจากพอร์ต A ที่ขา 34 ถึง 40 และในการตรวจสอบหลอดขาดไมโครคอนโทรลเลอร์จะรับข้อมูลที่พอร์ต B ที่ขา 4 เมื่อได้ข้อมูลเรียบร้อยแล้วไมโครคอนโทรลเลอร์จะนำข้อมูลมาประมวลผลเพื่อส่งให้กับโปรแกรมรับส่งข้อมูลผ่านพอร์ตอนุกรม ซึ่งในการประมวลผลจะได้ข้อมูลเพื่อส่งให้กับโปรแกรมรับส่งข้อมูลผ่านพอร์ตอนุกรม 3 ส่วนคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. การรับสถานะของหลอดไฟ โดยจะรับข้อมูลผ่านพอร์ต C ที่ขา 22 ถึง 28 และจะนำมาประมวลผล โดยจะแบ่งสถานะการทำงานได้ 7 ชนิดดังตารางที่ 5.2 ซึ่ง L1 คือ ระดับความสว่างมากสุด และระดับอื่นๆก็จะลดระดับลงมาจนถึง L7 คือการดับสนิทของหลอดไฟ

ตารางที่ 5.2 การกดปุ่มคีย์บอร์ดแต่ละระดับให้กับ VB

กดปุ่มคีย์บอร์ด	แสดงสถานะที่ VB
1	L1
2	L2
3	L3
4	L4
5	L5
6	L6
7	L7

2. การควบคุมการทำงานด้วยมือ จะรับข้อมูลผ่านพอร์ต A ที่ขา 34 ถึง 40 ไมโครคอนโทรลเลอร์จะนำมาประมวลผลได้ดังตารางที่ 5.3 ซึ่ง S1 คือ ระดับความสว่างมากสุด และระดับอื่นๆก็จะลดระดับลงมาจนถึง S7 คือการดับสนิทหลอดไฟ

ตารางที่ 5.3 การกดสวิตซ์แต่ละระดับให้กับ VB

กดสวิตซ์	แสดงสถานะที่ VB
1	S1
2	S2
3	S3
4	S4
5	S5
6	S6
7	S7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 หากมีการขาดของหลอดเกิดขึ้น ไมโครคอนโทรลเลอร์จะมีการรับสถานะที่พอร์ต B ขา 4 ซึ่งมีระดับสัญญาณ logic “0” และนำมาประมวลผลเป็น “z” เพื่อส่งให้โปรแกรมรับส่งข้อมูลผ่านพอร์ตอนุกรมต่อไป

ในส่วนของโปรแกรมการรับส่งข้อมูลจากพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ จะรับส่งข้อมูลระหว่างโปรแกรมตรวจสอบสถานะและโปรแกรม Visual Basic ที่พอร์ต D ขา 14 และขา 15 ซึ่งเป็นขา RXD และ TXD ตามลำดับ โดยผ่านทางพอร์ตอนุกรม RS-232 เพื่อเข้าสู่คอมพิวเตอร์ทางพอร์ต com1 หรือ com2 ต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6
สรุปและวิจารณ์ผลการทดลอง

ในการทดลองของโครงงานนี้สามารถทำการติดต่อกันระหว่างโทรศัพท์มือถือและไมโครคอนโทรลเลอร์เพื่อควบคุมการทำงานของหลอดไฟได้ นอกจากนั้นยังสามารถใช้ WAP Browser เชื่อมต่อกับอินเตอร์เน็ตเพื่อควบคุม และใช้การควบคุมจากโปรแกรม Visual Basic ได้ และยังสามารถควบคุมการทำการตามปกติด้วยมือได้อีกด้วย ซึ่งทำให้โครงงานนี้ใช้งานได้อย่างหลากหลายและสามารถพัฒนาการทำงานได้

6.1 ปัญหาในการทดลอง

6.1.1 ในส่วนของการเขียน WAP Site โดยใช้ภาษา WML ร่วมกับ WMLscript มีความยุ่งยากในการใช้งาน

6.1.2 โปรแกรม Nokia Toolkit ที่ดาวน์โหลดมาใช้ในการทดลองมีระยะเวลาในการใช้งานจำกัดเมื่อครบกำหนดระยะเวลาแล้วไม่สามารถใช้งานได้

6.1.3 โปรแกรม Apache Web Server มีปัญหาในการติดตั้งเพราะใช้งานกับหลายภาษา

6.1.4 โปรแกรม Nokia Toolkit ใช้งานยากเพราะเป็นโปรแกรมต้นแบบที่เพิ่งพัฒนามาได้ไม่นาน

6.1.5 ในส่วนของไมโครคอนโทรลเลอร์ AVR ยุ่งยากในการหาข้อมูลเพราะยังไม่มีหนังสือภาษาไทยให้ศึกษา

6.1.6 ไมโครคอนโทรลเลอร์ AVR มีความเปราะบางสูง มีราคาแพงและเกิดความเสียหายได้ง่าย

6.1.7 ในโปรแกรม WAP Browser ภายในโทรศัพท์มือถือมีความไม่เสถียรภาพภายในตัวโปรแกรมทำให้การใช้งานเกิดความผิดพลาดในการใช้งาน

6.1.8 ในการทำงานของระบบโดยรวมยังมีปัญหาเกิดขึ้นในบางส่วน

6.2 แนวทางการแก้ไข

6.2.1 ได้เปลี่ยนภาษาสคริปต์จาก ภาษา WMLscript มาเป็นภาษา PHP ที่มีความยืดหยุ่นและใช้งานง่ายกว่า

6.2.2 ดาวน์โหลดโปรแกรม Nokia Toolkit เวอร์ชั่นใหม่มาใช้แทน

6.2.3 ทำการศึกษาหาข้อมูลเพิ่มเติมในการแก้ไข configuration เพื่อแก้ปัญหาในการใช้งานร่วมกันหลายภาษาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.2.4 ทำการศึกษาการทำงานในส่วนต่าง ๆ ของโปรแกรม Nokia Toolkit และทดลองใช้บ่อยๆ เพื่อเพิ่มความชำนาญในการใช้งาน

6.2.5 ซื้อบอร์ดทดลองในการเขียนโปรแกรมเพราะจะมีคู่มือภาษาไทยบางส่วนให้ทดลองโปรแกรม จึงสามารถที่จะเขียนโปรแกรมได้

6.2.6 เพิ่มความระมัดระวังในการใช้งานไมโครคอนโทรลเลอร์มากเป็นพิเศษ

6.2.7 ทำการโหลดข้อมูลใหม่อีกครั้ง เพื่อรับสถานะที่ส่งมาใหม่ได้อย่างถูกต้อง

6.2.8 ทำการพัฒนาซอฟท์แวร์และฮาร์ดแวร์เพื่อแก้ไขปัญหาของระบบ และทำให้ระบบมีเสถียรภาพมากขึ้น

6.3 แนวทางการพัฒนา

6.3.1 แก้ไขการทำงานของโครงงานเพื่อเพิ่มเสถียรภาพให้แก่ระบบ ทั้งในส่วนของ WAP Site, โปรแกรม Visual Basic ในส่วนของไมโครคอนโทรลเลอร์และอุปกรณ์ เพื่อตอบสนองการใช้งานได้อย่างเต็มที่

6.3.2 ทำการศึกษาอุปกรณ์ไฟฟ้าชนิดอื่น เพื่อนำมาพัฒนาโครงงาน ให้สามารถควบคุมอุปกรณ์ไฟฟ้าชนิดต่างๆ ได้

6.3.3 ถ้ามีการศึกษาการใช้อุปกรณ์ไฟฟ้าได้อย่างหลายหลาย สามารถรวมการทำงานในอุปกรณ์ต่าง ๆ ให้เป็นการใช้งานภายใต้ระบบเดียวกันได้

6.3.4 สามารถดัดแปลงโครงงานไปเป็นธุรกิจเพื่อนำไปใช้ในเชิงพาณิชย์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. กิตติ ภักดีวัฒนะกุล, จำลอง ครูอุตสาหะ, “Visual Basic 6 ฉบับโปรแกรมเมอร์” หจก. ไทยเจริญการพิมพ์, 2521

2 กฤษดา ใจเย็น, อรรถพล บุญยะโภคา, และชัยวัฒน์ ลิ้มพรจิตรวิไล, “เรียนรู้และปฏิบัติการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ภายนอกผ่านพอร์ตอนุกรม บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด” 2521.

3 นิรุธ อำนวยศิลป์, “PHPเพื่อการประยุกต์ใช้งาน” บริษัท ซัคเวส มีเดีย จำกัด, 2521.

4 สราวุธ อ้อยศรีสกุล, “เปิดมิติ Mobile Internet ด้วย WAP” บริษัท วิตตี้ กรุ๊ป จำกัด, 2537.

5 นิรันดร ทนงศักดิ์มนตรี WAP The World In Your Hand เล่มที่ 1 กรุงเทพฯ บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน), 2543.

6 ร.ท. อนุโชต วุฒิพรพงษ์,ร.ต. พันธุ์เทพ แก้วมงคล “สร้าง WAP Page ด้วย WML Script สำนักพิมพ์ อินโฟเพรส, 2543.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้