

# สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

## การควบคุมระบบแสงสว่างระยะไกลผ่านระบบอินเทอร์เน็ต และเครือข่าย ZigBee Remote Lighting Control via Internet and ZigBee Network



เลขหมู่.....  
เลขทะเบียน..... 81786  
วัน,เดือน,ปี 24 ส.ย. 2551

b..... 11436417  
i.....

ปฏิญานีพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2550  
การควบคุมระบบแสงสว่างระยะไกลผ่านระบบอินเทอร์เน็ต  
และเครือข่าย ZigBee  
Remote Lighting Control via Internet and ZigBee Network



อาจารย์ที่ปรึกษา

ดร. สมภาพ ผลไม้  
ผศ.ดร. สุพัฒน์ กิตติรัตน์สัจจา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2550

ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมระบบแสงสว่างระยะไกลผ่านระบบอินเทอร์เน็ตและเครือข่าย ZigBee

ผู้จัดทำ

1. นาย ฐิฐนนท์ เหล่าสมบัติ
2. นาย ฐิฐพงศ์ อุดมประเสริฐ
3. นางสาว ทิพย์วิมล กரியอาภรณ์



.....  
( ดร. สมภพ ผลไม้ )

..... อาจารย์ที่ปรึกษา  
( ผู้ช่วยศาสตราจารย์ ดร. สุพัฒน์ กิตติรัตน์สัจจา )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การควบคุมระบบแสงสว่างระยะไกลผ่านระบบอินเทอร์เน็ตและเครือข่าย

## ZigBee

นาย ณัฐนนท์	เหล่าสมบัติ	
นาย ณัฐพงศ์	อุดมประเสริฐ	
นางสาว ทิพย์วิมล	กริยาอาภรณ์	
ดร. สมภพ	ผลไม้	อาจารย์ที่ปรึกษา
ผศ.ดร. สุพัฒน์	กิตติรัตน์สัจจา	อาจารย์ที่ปรึกษา
ปีการศึกษา 2550		

## บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้นำเสนอการควบคุมระบบส่องสว่างระยะไกลผ่านระบบอินเทอร์เน็ตและเครือข่าย ZigBee โดยประกอบด้วย เว็บเซิร์ฟเวอร์แบบฝังตัวทำหน้าที่ติดต่อผู้ใช้ผ่านทางหน้าเว็บทางอินเทอร์เน็ต ชุดรับส่งข้อมูลไร้สาย ZigBee จำนวนสามตัว ซึ่งถูกกำหนดให้ทำงานแบบเครือข่ายและหน้าที่รับส่งข้อมูลระหว่าง เว็บเซิร์ฟเวอร์ และชุดควบคุมการส่องสว่าง การควบคุมการส่องสว่างในปริญญาานิพนธ์นี้ประกอบด้วย การควบคุมแบบ เปิด-ปิด สำหรับหลอดไฟชนิดเผาไส้ การควบคุมการหรี่ไฟด้วยเทคนิค PWM สำหรับหลอดไฟ LED และ การควบคุมการผสมสีสำหรับหลอด LED สามสีหลักเพื่อให้ได้สีที่ต้องการ ในปริญญาานิพนธ์ฉบับนี้จะอธิบายถึงขั้นตอนการออกแบบระบบและวงจรการโปรแกรมอุปกรณ์ที่เกี่ยวข้องต่าง ๆ การประยุกต์ใช้งาน ตลอดจนผลการทดลองและแนวทางการพัฒนาต่อไปในอนาคต

# Remote Lighting Control via Internet and ZigBee Network

Nattanon	Laosombat	
Nuttapong	Udomprasert	
Thipwimon	Kreeya-arporn	
Dr. Somphob	Polmai	Supervisor
Asst. Prof. Dr. Supat	Kittiratsatcha	Supervisor

2007

## ABSTRACT

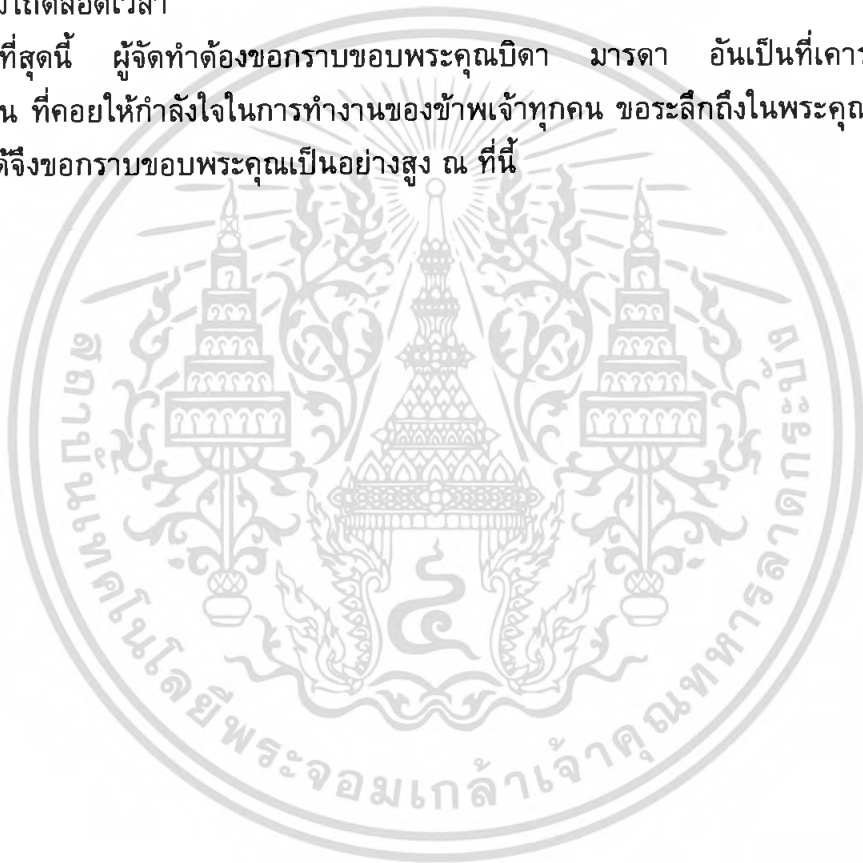
This thesis presents remote lighting control via internet and ZigBee network. The system is composed of an embedded web server that interacts with user from the web page, 3 ZigBee modems configured as a wireless network performing data exchange between the web server and the local lighting control systems. In this thesis, there are two local lighting control systems; one for the on-off control of the incandescent lamps and the other for dimming a white-light LED lamp and color mixing of a tri-color LED lamp. In this thesis, the design of the system and circuits are described first, and then the programming of each component and applications are explained. The experiment and its results as well as the comment for further development are also presented.

## กิตติกรรมประกาศ

ผู้จัดทำโครงการนัฐศึกษาซึ่งต่อความกรุณาที่ ดร. สมภพ ผลไม้ และ ผศ.ดร. สุพัฒน์ กิตติรัตนัสัจจา อาจารย์ที่ปรึกษา ที่ได้ให้คำแนะนำ ตลอดจนอาจารย์ในภาควิชาที่ประสิทธิประสาทความรู้ให้กับผู้จัดทำในครั้งนี้ ถ้าปราศจากท่านโครงการนี้ก็คงไม่สำเร็จลุล่วงไปได้

นอกจากนั้นผู้จัดทำต้องขอขอบคุณ คุณแอกวี่ฉวี สุขประจันทร์ ที่คอยช่วยเหลือให้คำปรึกษาทางด้าน ฮาร์ดแวร์ และคุณไทรสลัน เหลืองอ่อน คุณศิริวัฒน์ อ่วมงามทรัพย์ ที่คอยช่วยเหลือคำปรึกษาด้านโปรแกรมต่างๆ ตลอดจนเจ้าหน้าที่ห้องธุรการที่ให้คำแนะนำเกี่ยวกับการเบิกใบเสร็จของโครงการ และผู้จัดทำต้องขอบคุณเพื่อนร่วมห้องปฏิบัติการ (lab) ที่คอยเอาใจใส่คอยถามไถ่ตลอดเวลา

ท้ายที่สุดนี้ ผู้จัดทำต้องขอกราบขอบพระคุณบิดา มารดา อันเป็นที่เคารพรักของข้าพเจ้าทุกคน ที่คอยให้กำลังใจในการทำงานของข้าพเจ้าทุกคน ขอระลึกถึงในพระคุณอันหาสิ่งตอบแทนมิได้จึงขอกราบขอบพระคุณเป็นอย่างสูง ณ ที่นี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา III ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VI
สารบัญตาราง.....	IX
บทที่ 1 บทนำ.....	1
1.1 ชื่อโครงการ.....	1
1.2 ส่วนนำ.....	1
1.2.1 ที่มาและความสำคัญ.....	1
1.2.2 โครงการที่นำเสนอ.....	2
1.3 วัตถุประสงค์ของโครงการ.....	3
1.4 ผลที่คาดว่าจะได้รับ.....	3
1.5 ขอบเขตของโครงการ.....	3
1.6 วิธีนำเสนอโครงการ.....	4
1.7 ฝั่งเวลาการดำเนินโครงการ.....	4
1.8 การจัดโครงสร้างของเนื้อหาในปฏิญานินทร์.....	5
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	7
2.1 ส่วนนำ.....	7
2.2 ทฤษฎี.....	7
2.2.1 รูปแบบระบบเครือข่าย TCP/IP.....	8
2.2.2 โครงสร้างของโปรโตคอล TCP/IP.....	9
2.2.3 Encapsulation.....	10
2.2.4 Demultiplexing.....	11
2.2.5 โปรโตคอล IP (Internet Protocol) .....	12
2.2.6 โปรโตคอล TCP (Transmission Control Protocol).....	18
2.2.7 โปรโตคอล HTTP (Hyper text transfer protocol).....	24
2.2.8 HTML (Hypertext Markup Language).....	26
2.2.9 ความสัมพันธ์ระหว่าง HTTP กับ HTML.....	28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ(ต่อ)

หน้า

2.2.10 โปรแกรม Dynamic C.....	30
2.2.11 Rabbit3720 Microprocessor.....	32
2.2.12 ZigBee Modem.....	36
2.2.13 บอร์ด Controller.....	36
<b>บทที่ 3 การออกแบบโปรแกรม.....</b>	<b>43</b>
3.1 การออกแบบโดยรวมของโครงการ.....	46
3.2 การออกแบบในส่วนของโปรแกรม Dynamic C เวอร์ชัน 9.25.....	46
3.2.1 การตั้งค่า IP Address ให้กับบอร์ด Rabbit จะใช้คำสั่งต่อไปนี้.....	47
3.2.2 การเรียกใช้ชื่อเอกสาร Web page ที่สร้างไว้เก็บลง ในบอร์ด Rabbit จะใช้คำสั่งต่อไปนี้.....	48
3.3 การเชื่อมต่อ ZigBee ทั้ง 2 ตัวโดยใช้โปรแกรม X-CTU.....	49
3.3.1 การตั้งค่าพารามิเตอร์ของ Zig Bee.....	50
3.4 การสร้างเว็บไซต์เพจ.....	50
3.5 การออกแบบโปรแกรม Microcontroller.....	53
<b>บทที่ 4 ระบบการทดลองและผลการทดลอง.....</b>	<b>55</b>
4.1 การทดสอบการทำงานของชุด เปิด ปิด หลอดไฟ.....	55
4.2 การทดสอบการทำงานของชุดปรับแสงสว่างและชุดผสมสี.....	56
4.3 การทดลองการทำงานจริง.....	59
<b>บทที่ 5 สรุปและวิจารณ์โครงการ.....</b>	<b>76</b>
5.1 สรุปโครงการ.....	76
5.2 ปัญหา อุปสรรค และการแก้ไข.....	76
5.3 แนวทางการพัฒนา.....	77
<b>บรรณานุกรม.....</b>	<b>78</b>
<b>ภาคผนวก .....</b>	<b>79</b>
<b>ประวัติผู้เขียน .....</b>	<b>109</b>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ

หน้า

ภาพที่ 1.1 แสดงภาพรวมของโครงการ.....	2
ภาพที่ 2.1 ระบบผู้ให้บริการและผู้ใช้บริการ.....	7
ภาพที่ 2.2 การแบ่งชั้นของ TCP/IP.....	8
ภาพที่ 2.3 แสดงให้เห็นถึงความสัมพันธ์ระหว่างโปรโตคอลต่างๆใน TCP/IP.....	9
ภาพที่ 2.4 ขั้นตอนการ Encapsulation เมื่อข้อมูลถูกส่งผ่านโปรโตคอลต่างๆ.....	11
ภาพที่ 2.5 การ Demultiplexing ข้อมูล.....	12
ภาพที่ 2.6 การกำหนด IP Address ในคลาสต่างๆ.....	13
ภาพที่ 2.7 IP Header.....	13
ภาพที่ 2.8 network ตัวอย่าง.....	17
ภาพที่ 2.9 TCP Header.....	20
ภาพที่ 2.10 Connection Establishment.....	22
ภาพที่ 2.11 Connection Termination.....	23
ภาพที่ 2.12 HTTP กับ HTML.....	29
ภาพที่ 2.13 รูปแบบการทำงาน HTTP.....	29
ภาพที่ 2.14 การทำงานแบบไม่จองสายของ HTTP.....	30
ภาพที่ 2.15 โปรแกรม Dynamic C.....	30
ภาพที่ 2.16 ไฟล์ที่อยู่ใน Lib.....	31
ภาพที่ 2.17 ไฟล์ที่อยู่ใน TCP/IP.....	32
ภาพที่ 2.18 บอร์ด Rabbit.....	33
ภาพที่ 2.19 แสดงระบบย่อยของ RCM 3700.....	34
ภาพที่ 2.20 แสดง pinout สำหรับ RJ-45 Ethernet port (J3).....	34
ภาพที่ 2.21 Rabbit 3000 TFBGA.....	35
ภาพที่ 2.22 ZigBee Modem.....	36
ภาพที่ 2.23 บอร์ด ET-BASE-PIC40/458.....	38
ภาพที่ 2.24 โครงสร้างของบอร์ด ET-BASE PIC40.....	39
ภาพที่ 2.25 การจัดเรียงขาสัญญาณของบอร์ด ET-BASE PIC40.....	40
ภาพที่ 2.26 การจัดเรียงสัญญาณของพอร์ต RS232.....	40
ภาพที่ 2.27 ขาสัญญาณ PORTE.....	40
ภาพที่ 2.28 จัมป์เปอร์เลือกการใช้งาน PIC MCU.....	41
ภาพที่ 2.29 ET-CAB 10P.....	41
ภาพที่ 3.1 ภาพรวมของโครงการ.....	43
ภาพที่ 3.2 รายละเอียดของอุปกรณ์แต่ละส่วนที่ใช้ในการทำโครงการ.....	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

หน้า

ภาพที่3.3 หน้าโปรแกรม Dynamic C.....	46
ภาพที่3.4 หน้าต่างโปรแกรมการทำงานของการ์ดตั้งค่า IP Address ให้กับ RCM 3720.....	47
ภาพที่3.5 แสดงหน้าต่างโปรแกรม Dynamic C ที่มีการเขียนคำสั่งควบคุมการทำงานของ อุปกรณ์ไฟฟ้า.....	48
ภาพที่3.6 หน้าโปรแกรม X-CTU.....	49
ภาพที่3.7 แสดงหน้าต่างโปรแกรม X-CTU ในการตั้งค่าพารามิเตอร์ให้กับ Zig Bee.....	50
ภาพที่3.8 แสดงหน้าต่างพร้อมใช้งานของโปรแกรม Adobe Dreamwaver CS3.....	51
ภาพที่3.9 แสดงหน้าต่าง Design ของโปรแกรม Adobe Dreamwaver CS3.....	51
ภาพที่3.10 แสดงหน้าต่างแสดง code ของโปรแกรม Adobe Dreamwaver CS3.....	52
ภาพที่3.11 แสดงหน้าต่างการทำงานของโปรแกรม Adobe Dreamwaver CS3 ทั้งส่วน ที่เป็นหน้าต่าง Design และหน้าต่าง code HTML.....	52
ภาพที่3.12 แสดงหน้าต่างเว็บเพจที่ใช้ควบคุมอุปกรณ์ไฟฟ้า.....	53
ภาพที่3.13 แสดง Flowchart การทำงานของ Microcontroller.....	54
ภาพที่4.1 แสดงสถานะของหลอดไฟวงจรขับเมื่อมีการส่งคำสั่งให้ Microcontroller.....	55
ภาพที่4.2 แสดงสถานการณ์ทำงานที่ 0%.....	56
ภาพที่4.3 แสดงสถานการณ์ทำงานที่ 20%.....	57
ภาพที่4.4 แสดงสถานการณ์ทำงานที่ 50%.....	57
ภาพที่4.5 แสดงสถานการณ์ทำงานที่ 90%.....	58
ภาพที่4.6 แสดงการ Log in หน้าเว็บเพจ.....	59
ภาพที่4.7 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1.....	60
ภาพที่4.8 แสดงสถานะหลอดไฟหลอดที่ 1.....	60
ภาพที่4.9 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1 และ 2.....	61
ภาพที่4.10 แสดงสถานะหลอดไฟหลอดที่ 1 และ 2.....	61
ภาพที่4.11 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1, 2 และ 3.....	62
ภาพที่4.12 แสดงสถานะหลอดไฟหลอดที่ 1, 2 และ 3.....	62
ภาพที่4.13 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1, 2, 3 และ 4.....	63
ภาพที่4.14 แสดงสถานะหลอดไฟหลอดที่ 1, 2, 3 และ 4.....	63
ภาพที่4.15 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 0 %.....	64
ภาพที่4.16 แสดงสถานะของโคมหลอดขาวที่ 0%.....	64
ภาพที่4.17 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 10 %.....	65
ภาพที่4.18 แสดงสถานะของโคมหลอดขาวที่ 10%.....	65
ภาพที่4.19 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 50 %.....	66

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

	หน้า
ภาพที่4.20 แสดงสถานะของโคมหลอดขาวที่ 50%.....	66
ภาพที่4.21 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 100 %.....	67
ภาพที่4.22 แสดงสถานะของโคมหลอดขาวที่ 100%.....	67
ภาพที่4.23 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีแดง).....	68
ภาพที่4.24 แสดงสถานะโคมหลอด 3 สี (สีแดง).....	68
ภาพที่4.25 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีเขียว).....	69
ภาพที่4.26 แสดงสถานะโคมหลอด 3 สี (สีเขียว).....	69
ภาพที่4.27 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีน้ำเงิน).....	70
ภาพที่4.28 แสดงสถานะโคมหลอด 3 สี (สีน้ำเงิน).....	70
ภาพที่4.29 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีส้ม).....	71
ภาพที่4.30 แสดงสถานะโคมหลอด 3 สี (สีส้ม).....	71
ภาพที่4.31 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีชมพู).....	72
ภาพที่4.32 แสดงสถานะโคมหลอด 3 สี (สีชมพู).....	72
ภาพที่4.33 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีขาว).....	73
ภาพที่4.34 แสดงสถานะโคมหลอด 3 สี (สีชมพู).....	73
ภาพที่4.35 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 1.....	74
ภาพที่4.36 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 2.....	74
ภาพที่4.37 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 3.....	74
ภาพที่4.38 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 4.....	75
ภาพที่4.39 แสดงภาพอุปกรณ์ที่ใช้ทดลองทั้งหมด.....	75

## สารบัญตาราง

	หน้า
ตารางที่ 1.1 ตารางผังเวลาการทำงาน.....	4
ตารางที่ 2.1 แสดงช่วงของ IP Address แต่ละคลาส.....	13
ตารางที่ 2.2 อธิบายตำแหน่งต่างๆ ของ IP Header.....	14
ตารางที่ 2.3 รายละเอียด TCP Header.....	20
ตารางที่ 2.4 รายละเอียดคำสั่งของ HTTP.....	25
ตารางที่ 2.5 รหัสแสดงสถานะ การทำงานของโปรโตคอล HTTP.....	26



# บทที่ 1

## บทนำ

ในปัจจุบันได้มีการนำเทคโนโลยีที่ทันสมัยเข้ามาใช้มากขึ้น ทั้งด้านการสื่อสาร อุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ ซึ่งได้มีการพัฒนามาจากในอดีต โดยเฉพาะทางด้านการสื่อสาร ในปัจจุบันระบบ Wireless LAN ได้เป็นสื่อที่รู้จักกันอย่างแพร่หลายและเป็นสื่อที่ได้มีการพัฒนาอย่างต่อเนื่องควบคู่ไปกับการดำเนินชีวิตของมนุษย์ในปัจจุบันทางกลุ่มได้รับหัวข้อโครงการเกี่ยวกับการควบคุมระบบไฟฟ้าผ่านระบบเครือข่าย Wireless LAN ซึ่งเป็นอีกแนวทางหนึ่งในการนำเทคโนโลยีสมัยใหม่มาประยุกต์ใช้กับการควบคุมการทำงานของอุปกรณ์ไฟฟ้าภายในที่พักอาศัย และอาคารสำนักงาน โดยจะเพิ่มความความสะดวกสบายให้แก่ผู้ใช้งาน ตามความต้องการซึ่งในอดีตการควบคุมอุปกรณ์ไฟฟ้าแต่ละชนิด จะต้องควบคุมด้วยมือโดยตรงในระยะทางที่ใกล้กับอุปกรณ์ แต่ในปัจจุบันมีการควบคุมหลายรูปแบบ แต่ในโครงการนี้จะใช้การควบคุมระบบไฟฟ้าผ่านเครือข่าย wireless ซึ่งสามารถพัฒนาการใช้งานให้มีความหลากหลายมากขึ้นอีกในอนาคต

### 1.1 ชื่อโครงการ

Remote Lighting Control via Internet and ZigBee Network

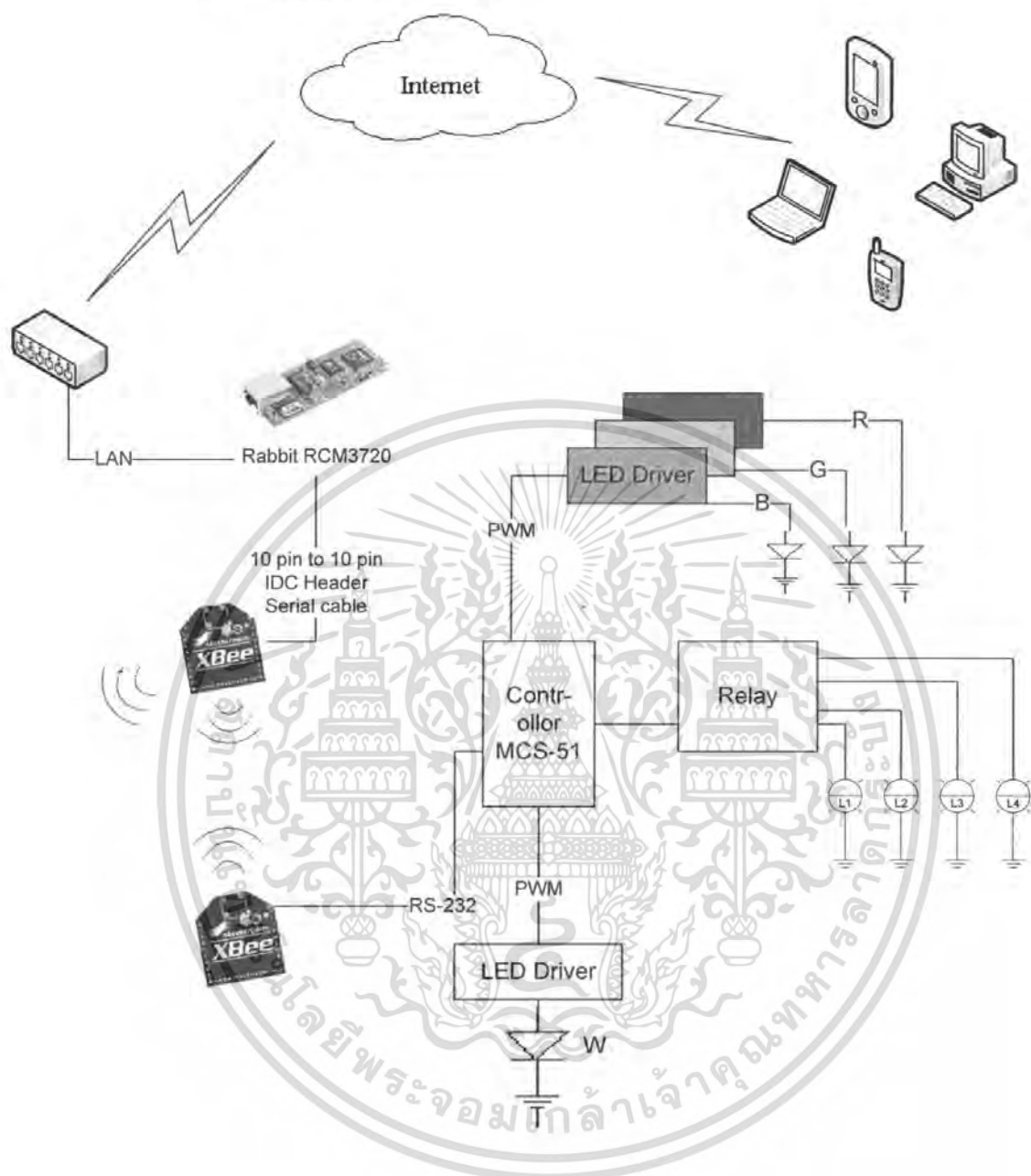
### 1.2 ส่วนนำ

#### 1.2.1 ความสำคัญและที่มา

การควบคุมอุปกรณ์ไฟฟ้าในอาคารที่พักอาศัย แต่ก่อนจะต้องทำการควบคุมการเปิดปิด ด้วยมือของผู้ใช้งานในอุปกรณ์ไฟฟ้านั้น ๆ โดยตรง และเมื่อผู้ใช้งานออกจากที่ทำงานหรือที่พักอาศัย อาจเกิดปัญหาการหลงลืมเปิดปิดอุปกรณ์ไฟฟ้าทิ้งไว้ ทำให้เกิดการสูญเสียและสิ้นเปลืองพลังงานไฟฟ้าโดยใช่เหตุ และถ้าอยู่ในที่ห่างไกลจากสถานที่ทำงานหรือที่พักอาศัยทำให้ผู้ใช้งานต้องเสียเวลาในการเดินทางกลับมาเพื่อปิดด้วยตัวเอง

อุปกรณ์ชุดนี้จะเป็นชุดทดลอง เพื่อการพัฒนาให้สามารถควบคุมอุปกรณ์ไฟฟ้าได้ในระยะไกลแบบไร้สาย โดยควบคุมอุปกรณ์ไฟฟ้าผ่านทางเครือข่าย Internet สามารถควบคุมการทำงานของอุปกรณ์ไฟฟ้าได้โดยไม่เสียเวลาในการเดินทาง ทำให้มีความสะดวกสบายในการใช้งานระบบไฟฟ้ามากขึ้น อีกทั้งยังสามารถทำให้เกิดการใช้งานหลากหลายมากยิ่งขึ้น

## 1.2.2 โครงการที่นำเสนอ



ภาพที่ 1.1 แสดงภาพรวมของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงการนี้นำเสนอการควบคุมอุปกรณ์ไฟฟ้าผ่านระบบ Internet โดยใช้ IP Address , User name และ Password ที่ถูกต้องก็สามารถ Access เข้าสู่หน้า Web page ที่ใช้ควบคุมอุปกรณ์ไฟฟ้าได้ เพียงแค่มีอุปกรณ์ที่สามารถเชื่อมต่อ Internet ได้ เช่น คอมพิวเตอร์ มือถือ PDA ฯลฯ โดยหลักการทำงานจะส่งคำสั่งผ่าน Web page ในระบบ Internet ไปที่ Rabbit ที่ทำหน้าที่เป็น server ซึ่งออนไลน์อยู่ตลอดเวลา จากนั้น Rabbit จะส่งคำสั่งไปที่ Coordinator ZigBee (เป็นโมเด็มที่มีตัวรับ-ส่งสัญญาณ Wireless ในตัวเดียวกัน) ที่เชื่อมต่ออยู่กับ Rabbit เพื่อให้ Coordinator ZigBee ส่งข้อมูลผ่านระบบ Wireless ไปที่ Children ZigBee ที่ต่ออยู่กับ Microcontroller เพื่อประมวลคำสั่งและ ควบคุมอุปกรณ์ไฟฟ้า

โดยในโครงการนี้จะทำการควบคุมการทำงานของหลอด LED 3 แบบ คือ เปิด – ปิด ปรับความเข้มแสง และผสมสี โดยตัว Microcontroller จะทำการตรวจสอบสถานะการทำงานของอุปกรณ์ไฟฟ้าอยู่ตลอดเวลาและส่งข้อมูลสถานะการทำงานของอุปกรณ์ไฟฟ้ากลับไปแสดงที่ Web page เพื่อเป็นการแสดงสถานะการทำงานของอุปกรณ์ไฟฟ้า ณ ปัจจุบัน

### 1.3 วัตถุประสงค์ของโครงการ

1. ศึกษาการทำงานของระบบเครือข่าย wireless
2. ศึกษาการเขียนโปรแกรม Dynamic C (Software ที่ใช้ในการควบคุม)
3. นำความรู้ที่ได้ศึกษามาประยุกต์ใช้สร้างชุดควบคุมอุปกรณ์ไฟฟ้า
4. เพื่อความสะดวกในการใช้งานและควบคุมการทำงานของเครื่องใช้ไฟฟ้าต่าง ๆ ในระยะไกล โดยผ่าน เครือข่ายของระบบ Internet

### 1.4 ผลงานที่คาดว่าจะได้รับ

1. อุปกรณ์ที่ทำให้เราสามารถควบคุมอุปกรณ์ไฟฟ้าได้ในระยะไกลเพียงแต่เราต้องมีอุปกรณ์ที่สามารถต่ออินเทอร์เน็ตได้เท่านั้น
2. สามารถตรวจสอบเช็คได้ว่าในขณะนี้มียุอุปกรณ์ไฟฟ้าตัวใด ทำงานหรือไม่ได้ทำงานอยู่บ้างจากรีโมท หรืออุปกรณ์ที่สามารถต่ออินเทอร์เน็ตได้

### 1.5 ขอบเขตของโครงการ

ขอบเขตของโครงการคือการเขียนโปรแกรมการควบคุมการทำงานของ LED ผ่านออกทางระบบ wireless เพื่อส่งข้อมูลให้กับชุดไมโครคอนโทรลเลอร์ เพื่อสั่งการทำงานให้กับชุด LED โดยใช้สัญญาณ PWM และสั่งเปิด ปิด อุปกรณ์ไฟฟ้าทั่วไปด้วย

## 1.6 วิธีการดำเนินโครงการ

1. เลือกหัวข้อในการดำเนินงาน
2. ศึกษาวัตถุประสงค์และขอบเขตการทำงานของโครงการ
3. ศึกษาและค้นคว้าข้อมูลที่เกี่ยวข้องกับโครงการ
4. เขียนโปรแกรมควบคุมการเปิด ปิด อุปกรณ์ไฟฟ้า และความสว่างของ Power LED โดยใช้โปรแกรม Dynamic C
5. ออกแบบและสร้างวงจรทดลองเพื่อศึกษาการทำงาน
6. ศึกษาผลการทดลองที่ได้แล้วนำผลการทดลองที่ได้มาวิเคราะห์ผลการทดลอง
7. สรุปผลการทดลอง รวมถึงปัญหาและอุปสรรคในการทำงาน

## 1.7 ผังเวลาดำเนินโครงการ

ตารางที่ 1.1 ตารางผังเวลาการทำงาน

ระยะเวลา(เดือน)	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
แผนงาน										
1.ศึกษาทฤษฎีและโปรแกรมที่เกี่ยวข้องของการทำโครงการ	←→									
2.ส่งรายงานความคืบหน้าการทำโครงการครั้งที่ 1			←→							
3.เขียนโปรแกรม dynamic c เพื่อให้โมดูล ZigBee ส่งสัญญาณไวร์เลสถึงกันได้			←→							
4.เขียนหน้าเวปเพจที่ใช้ในการควบคุมแสงสว่างผ่านอินเตอร์เน็ต			←→							
5.เขียนโปรแกรมเพื่อให้ไมโครคอนโทรลเลอร์รับคำสั่งจากโมดูล ZigBee แล้วประมวลผลสั่งอุปกรณ์ทำงานได้						←→				

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



- Rabbit 3720 Microprocessor
- ZigBee Modem
- บอร์ด Controller

### บทที่ 3 การออกแบบโปรแกรม

- การออกแบบโดยรวมของโครงการ
- การออกแบบในส่วนของโปรแกรม Dynamic C
- การเชื่อมต่อ ZigBee ทั้ง 2 ตัว โดยใช้โปรแกรม X-CTU
- การสร้างหน้าเว็บเพจ
- การออกแบบโปรแกรม Microcontroller

### บทที่ 4 การทดลองและผลการทดลอง

- การทดสอบการทำงานของชุด เปิด ปิด หลอดไฟ
- การทดสอบการทำงานของชุดปรับแสงสว่างและชุดผสมสี
- การทดลองการทำงานจริง

### สรุปและวิจารณ์โครงการ

- สรุปโครงการ
- ปัญหา อุปสรรค และการแก้ไข
- แนวทางการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

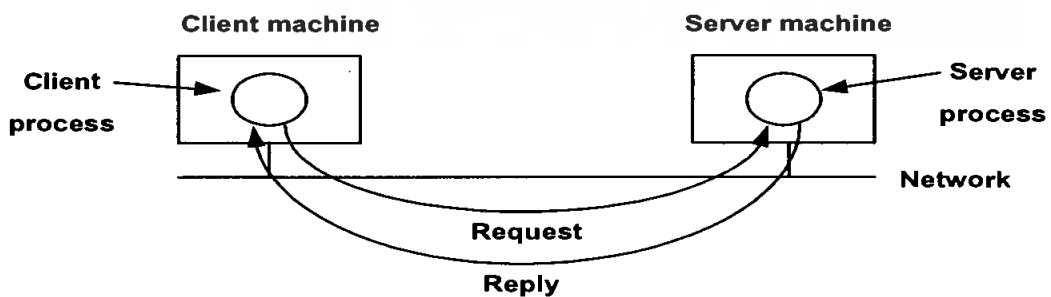
### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 ส่วนหน้า

โครงการนี้เป็นการศึกษาการควบคุมแสงสว่างผ่านระบบสื่อสารไร้สายเครือข่ายไวร์เลสแลน (Wireless LAN) โดยใช้ชุดทดลองใช้งาน ZigBee<sup>™</sup>/802.15.4 ประกอบด้วยโมเด็ม RF รุ่น XBee ของ MaxStream เข้ากับโมดูล RabbitCore RCM3720 ซึ่งเป็นรุ่นที่ได้รับความนิยมพร้อมด้วยการเชื่อมต่อที่ทำได้ง่าย ซึ่งการควบคุมจะกระทำผ่านหน้าเว็บเพจ ที่ได้สร้างขึ้นบน RabbitCore RCM 3720 การเรียกหน้าเว็บเพจจะเรียกผ่าน IP Address ที่ได้สร้างขึ้น โดยจะพิมพ์ลงไปในช่วง Internet Address หน้าเว็บที่เป็นแผงควบคุมก็จะแสดงออกมา เมื่อมีการเปลี่ยนแปลงแผงควบคุมที่หน้าเว็บเพจและมีการส่งคำสั่งผ่านเว็บ คำสั่งจะถูกส่งไปที่ RabbitCore RCM3720 จากนั้นคำสั่งจะถูกส่งไปยัง โมดูล ZigBee ตัวหนึ่งซึ่งถูกเชื่อมต่อกับ Ethernet ซึ่งทำให้ RabbitCore Module RCM3720 ทำงานได้ แล้ว โมดูล ZigBee ตัวดังกล่าว จะส่งคำสั่งนั้นผ่านระบบ Wireless LAN ไปยังโมดูล ZigBee อีกตัวหนึ่งที่ต่ออยู่กับชุดไมโครคอนโทรลเลอร์เพื่อควบคุมการแสงสว่างตามคำสั่งที่ได้รับ

#### 2.2 ทฤษฎี

ระบบเครือข่ายคอมพิวเตอร์ (Computer Network) หมายความว่าเครื่องคอมพิวเตอร์ตั้งแต่สองเครื่องขึ้นไปที่เป็นอิสระต่อกัน นำมาเชื่อมต่อถึงกันได้โดยไม่คำนึงถึงระยะทางระหว่างเครื่องทั้งสอง หมายถึงการเชื่อมต่อถึงกันนั้นก็ไม่ได้จำกัดเอาไว้ว่าจะต้องใช้แบบใดขอเพียงให้แลกเปลี่ยนข่าวสารข้อมูลกันได้ ดังรูปที่ 2.1 เป็นระบบผู้ให้บริการและใช้บริการ การสื่อสารในระบบนี้ “ผู้ใช้บริการ” จะส่งความต้องการในลักษณะของ การร้องขอ (request) ไปยัง “ผู้ให้บริการ” ซึ่งเมื่อผู้ให้บริการทำงานตามความต้องการนั้นๆ เสร็จแล้วจะส่งผลที่ได้กลับมายังผู้ใช้บริการในลักษณะของ การตอบสนอง (reply) สัดส่วนของผู้ให้บริการต่อผู้ใช้บริการนี้ขึ้นอยู่กับลักษณะของงานและประสิทธิภาพที่ต้องการเป็นหลัก



ภาพที่ 2.1 ระบบผู้ให้บริการและผู้ให้บริการ ( Client/Server )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเครือข่ายที่ใช้อยู่ในปัจจุบันคือระบบอินเทอร์เน็ตที่มีต้นกำเนิดมาจากระบบเครือข่ายชื่อ ARPANET โดยใช้สายโทรศัพท์เป็นหลัก ต่อมาเมื่อระบบการสื่อสารแบบคลื่นวิทยุ ความถี่สูงและการสื่อสารดาวเทียมเริ่มเข้ามามีบทบาทและนำมาใช้ในระบบมากขึ้น ทำให้กฎการสื่อสาร ( protocol ) ที่เคยใช้ได้ผลดีนั้นเกิดปัญหาไม่สามารถใช้งานได้อีกต่อไป กฎการสื่อสารรุ่นต่อมาจึงได้รับการออกแบบเพื่อนำมาใช้ทดแทนแบบเก่าโดยมีวัตถุประสงค์ในการเชื่อมการติดต่อระหว่างระบบที่มีความแตกต่างกันเป็นเรื่องหลัก ผลที่ได้รับคือกฎการสื่อสารที่เรียกว่า กฎการสื่อสารมาตรฐานแบบ TCP/IP ซึ่งได้รับการปรับปรุงจนนำมาใช้งานจริงได้ในปี ค.ศ. 1974 การปรับปรุงรุ่นต่อมา สำเร็จในปี ค.ศ. 1988

กฎการสื่อสารมาตรฐานแบบ TCP/IP ยังมีวัตถุประสงค์หลักอีกสองข้อสำคัญคือ ความสามารถในการแก้ไขปัญหาที่เกิดขึ้นในระบบเครือข่าย เช่นในกรณีที่ผู้ส่งและผู้รับยังคงมีการติดต่อกันอยู่ แต่โหนดกลางที่ใช้เป็นผู้ช่วยรับ - ส่งข้อมูลเกิดเสียหายใช้การไม่ได้หรือสายสื่อสารบางช่วงถูกตัดขาด กฎการสื่อสารนี้จะต้องสามารถจัดการหาทางเลือกอื่นเพื่อให้การสื่อสารดำเนินต่อไปได้โดยอัตโนมัติ ข้อที่สองคือจะต้องมีความอ่อนตัวต่อการสื่อสารข้อมูลได้หลายชนิดทั้งแบบที่ไม่มีความเร่งด่วน และแบบที่ต้องการรับประกันความเร่งด่วนของข้อมูล

### 2.2.1 รูปแบบระบบเครือข่าย TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นชุดของโปรโตคอลที่ใช้ในการสื่อสารผ่านเครือข่ายอินเทอร์เน็ต ได้รับการพัฒนามาตั้งแต่ปี 1960 ซึ่งถูกใช้เป็นครั้งแรกในเครือข่าย ARPANET ซึ่งต่อมาได้ขยายการเชื่อมต่อไปทั่วโลกเป็นเครือข่ายอินเทอร์เน็ต ทำให้ TCP/IP เป็นที่ยอมรับอย่างกว้างขวางจนถึงปัจจุบัน

#### การแบ่งชั้นของ TCP/IP

TCP/IP เป็นชุดโปรโตคอลที่ประกอบด้วยโปรโตคอลย่อยหลายตัวโดยแต่ละตัวก็จะทำหน้าที่ในแต่ละเลเยอร์ ซึ่งรับผิดชอบและแปลความหมายของข้อมูลในแต่ละระดับของการสื่อสารซึ่งในภาพรวมแล้ว TCP/IP แบ่งออกเป็น 4 เลเยอร์ดังรูป

Application
Transport
Network Layer
Link

ภาพที่ 2.2 การแบ่งชั้นของ TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### ชั้นสื่อสารเชื่อมต่อข้อมูล (Link Layer)

โพรโตคอลสำหรับการควบคุมการสื่อสารในชั้นนี้เป็นสิ่งที่ไม่มีการกำหนดรายละเอียดอย่างเป็นทางการ หน้าที่หลักคือการรับข้อมูลจากชั้น Network มาแล้วส่งไปยังโหนดที่ระบุไว้ในเส้นทางเดินข้อมูล ทางด้านผู้รับก็จะทำงานในทางกลับกันคือรับข้อมูลจากสายสื่อสารแล้วนำส่งให้กลับโปรแกรมในชั้น Network

### ชั้นสื่อสารควบคุมเครือข่าย (Network Layer)

ชั้นสื่อสารอินเทอร์เน็ต จากความต้องการทั้งหมดที่กล่าวถึงทำให้เกิดเป็นวิธีการส่งข้อมูลแบบหนึ่งเรียกว่าระบบเครือข่ายแบบสลับช่องสื่อสารระดับแพ็กเก็ต (packet – switching) ซึ่งเป็นการติดต่อแบบไม่ต่อเนื่อง ( connectionless ) หลักการทำงานคือการปล่อยให้ข้อมูลขนาดเล็กที่เรียกว่าแพ็กเก็ตสามารถไหลจากโหนดผู้ส่งแพ็กเก็ตไปตามโหนดต่าง ๆ ในระบบ จนถึงจุดหมายปลายทางได้โดยอิสระ

### ชั้นจัดการนำส่งข้อมูล ( Transport Layer)

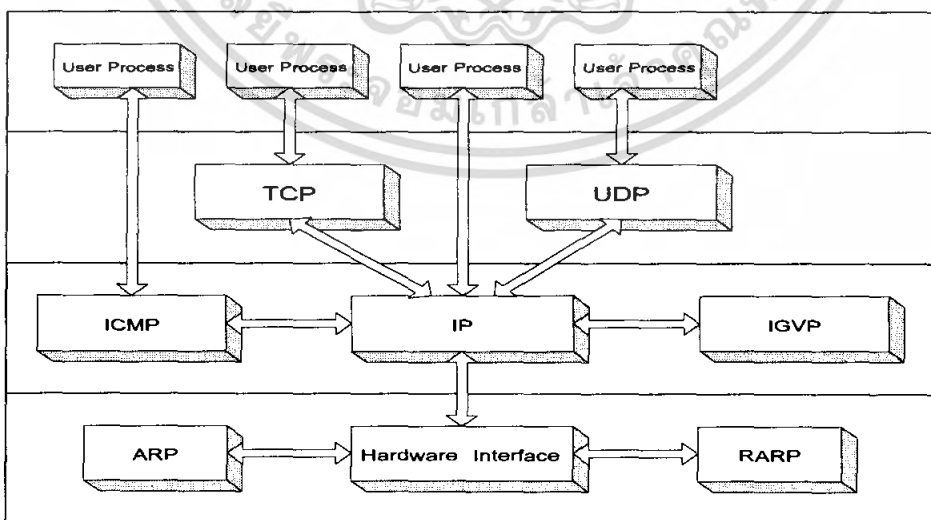
ชั้นสื่อสารนำส่งข้อมูล รับผิดชอบในการส่งข้อมูลระหว่างเครื่องหนึ่ง ไปยังอีกโฮสต์หนึ่ง และจะส่งข้อมูลขึ้นไปให้ Application Layer นำไปใช้งานต่อ

### ชั้นสื่อสารการประยุกต์ ( Application Layer)

เป็นเลเยอร์ที่แอปพลิเคชันเรียกใช้โพรโตคอลระดับล่างๆ ลงไปเพื่อวัตถุประสงค์ที่แตกต่างกัน เพื่อให้บริการต่างๆ เช่น FTP , SMTP , Telnet , HTTP , POP

## 2.2.2 โครงสร้างของโพรโตคอล TCP/IP

ในชุดของโพรโตคอล TCP/IP ประกอบด้วยโพรโตคอลหลายตัวทำงานร่วมกันในแต่ละเลเยอร์จึงมีหน้าที่แตกต่างกันออกไป รูป แสดงให้เห็นถึงโพรโตคอลในแต่ละเลเยอร์ที่เมื่อรวมกันเป็นชุดของ TCP/IP



ภาพที่ 2.3 แสดงให้เห็นถึงความสัมพันธ์ระหว่างโพรโตคอลต่างๆใน TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP (Transmission Control Protocol) อยู่ใน Transport Layer ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล และมี กลไกความคุมการ รับส่งข้อมูลให้มีความถูกต้อง ( reliable )

UDP (User Datagram Protocol) อยู่ใน Transport Layer ทำหน้าที่จัดการและควบคุมการรับส่งข้อมูล แต่ไม่มีกลไกควบคุมการรับ ส่งข้อมูลให้มีเสถียรภาพและเชื่อถือได้ ( unreliable, connectionless ) โดยปล่อยให้เป็นที่ของแอฟพลิเคชันเลยอร์

IP ( Internet Protocol ) อยู่ใน Internetwork Layer เป็นโปรโตคอลที่ทำหน้าที่รับภาระในการนำข้อมูลไปส่งยังผู้รับ ที่เชื่อมต่ออยู่ในระบบ network ซึ่งทั้งสองฝั่งอาจอยู่คนละเน็ตเวิร์กกันก็

ICMP ( Internet Control Message Protocol ) อยู่ใน Internetwork Layer เป็นโปรโตคอลที่ใช้ในการตรวจสอบและรายงานสถานะภาพของดาต้าแกรม โปรโตคอลนี้ทำงานในระดับ Network Layer เช่นเดียวกับ IP ในกรณีที่เกิดปัญหากับดาต้าแกรม ICMP จะถูกส่งออกไปยังโฮสต์ต้นทางเพื่อรายงานข้อผิดพลาด ที่เกิดขึ้น

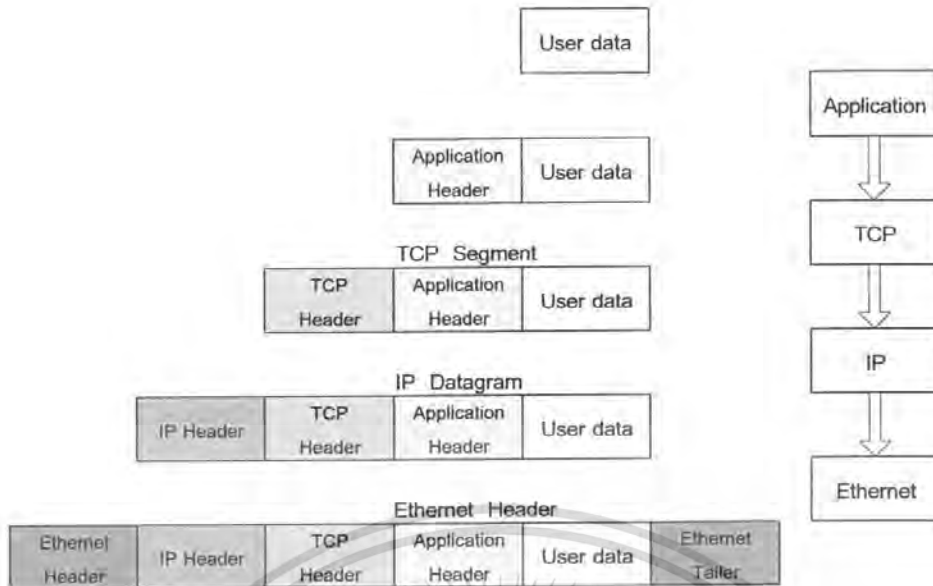
IGMP ( Internet Group Management Protocol ) อยู่ใน Internetwork Layer ทำหน้าที่ในการส่ง UDP ดาต้าแกรมไปยังกลุ่มของโฮสต์ หรือโฮสต์หลายๆตัวพร้อมกัน

ARP ( Address Resolution Protocol ) อยู่ใน Link Layer ทำหน้าที่เปลี่ยนระหว่างแอดเดรสที่ใช้โดย IP ให้เป็นแอดเดรสของ Network Interface

RARP ( Reverse ARP ) อยู่ใน Link Layer ทำหน้าที่กลับกันกับ ARP คือเปลี่ยนระหว่างแอดเดรสของ Network Interface ให้เป็นแอดเดรสที่ใช้โดย IP

### 2.2.3 Encapsulation

การ Encapsulate คือการนำข้อมูลที่ต้องการส่งมาประกอบรวมกับข้อมูลที่เป็นส่วนควบคุมของโปรโตคอล โดยข้อมูลส่วนที่เป็นส่วนควบคุมนั้นจะถูกนำมาไว้ในส่วนหัวของข้อมูล เรียกว่า เฮดเดอร์ (header) ซึ่งในการรับข้อมูลนั้นผู้ที่รับข้อมูลจะได้รับเฮดเดอร์ก่อนจากนั้นนำเฮดเดอร์ไปแปลและทราบว่าข้อมูลที่ตามมานั้นมีลักษณะอย่างไรจะได้จัดการได้อย่างถูกต้อง ภายในเฮดเดอร์ของโปรโตคอลส่วนใหญ่จะประกอบด้วยข้อมูลหลักที่สำคัญของโปรโตคอลที่ทำการ Encapsulate มาคือ แอดเดรสต้นทาง, แอดเดรสปลายทาง, ความยาวข้อมูล, รหัสตรวจสอบความผิดพลาดข้อมูล



ภาพที่ 2.4 ขั้นตอนการ encapsulation เมื่อข้อมูลถูกส่งผ่านโปรโตคอลต่างๆ

ข้อมูลที่ผ่านการ Encapsulate ในแต่ละระดับมีชื่อเรียกแตกต่างกัน ข้อมูลที่มาจาก User หรือก็คือข้อมูลที่ User เป็นผู้ป้อนให้กับ Application เรียกว่า (User Data)

เมื่อ Application ได้รับข้อมูลจาก user ก็จะนำมาประกอบกับส่วนหัวของ Application เรียกว่า Application Data และส่งต่อไปยังโปรโตคอล TCP

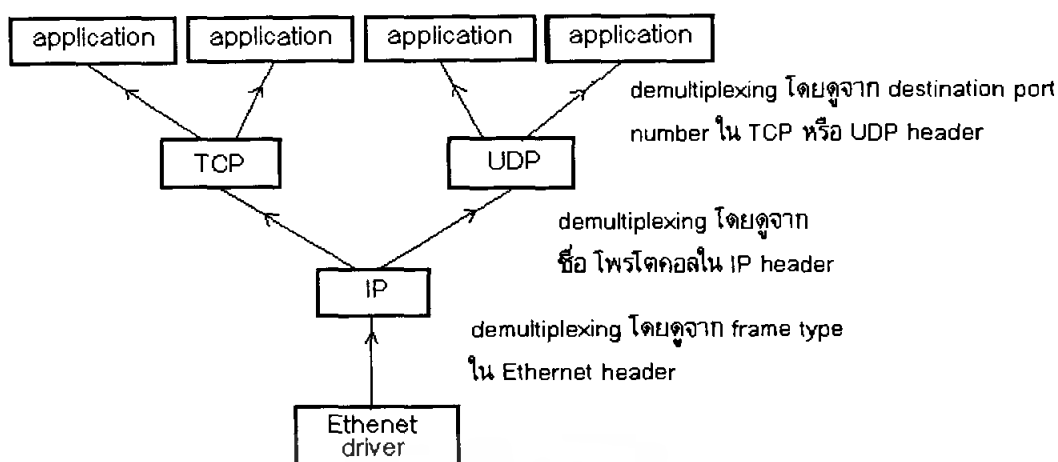
เมื่อโปรโตคอล TCP ได้รับ Application Data ก็จะนำมาพร้อมกับ Header ของโปรโตคอล TCP เรียกว่า (TCP Segment) และส่งต่อไปยังโปรโตคอล IP

เมื่อโปรโตคอล IP ได้รับ TCP Segment ก็จะนำมาพร้อมกับ Header ของโปรโตคอล IP เรียกว่า (IP Datagram) และส่งต่อไปยังเลเยอร์ Datalink Layer

ในระดับ Datalink จะนำ IP Datagram มาเพิ่มส่วน Error Correction และ flag เรียกว่า Ethernet Frame ก่อนจะแปลงข้อมูลเป็นสัญญาณไฟฟ้า ส่งผ่านสายสัญญาณที่เชื่อมโยงอยู่ต่อไป

### 2.2.4 Demultiplexing

การ Demultiplexing คือกระบวนการย้อนกลับของการ Encapsulation นั่นเอง การ Demultiplexing ก็คือการรับข้อมูลที่ถูกส่งมา และในแต่ละเลเยอร์ก็จะทำการถอดสแตคเดอริ่งออก แล้วส่งไปบนเลเยอร์ที่สูงกว่าที่ละชั้นตามเลเยอร์จนถึงเลเยอร์บนสุดคือแอปพลิเคชันเลเยอร์จึงได้ข้อมูลจริงที่ถูกส่งมา



ภาพที่ 2.5 การ Demultiplexing ข้อมูล

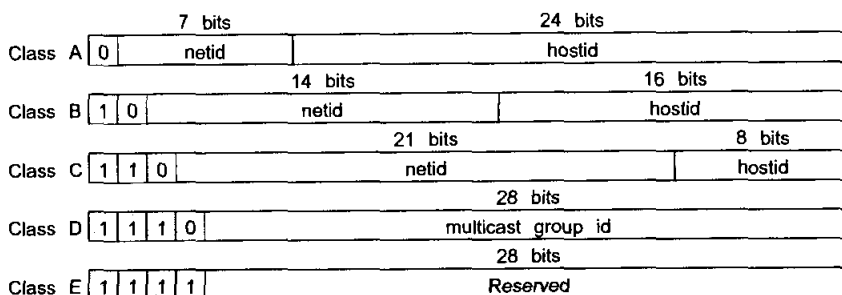
การ Demultiplex และ Encapsulate เป็นสิ่งคู่กันและสอดคล้องกัน อุปกรณ์ที่จะสื่อสารกันบนเน็ตเวิร์กได้ จะต้องมีส่วนที่ทำหน้าที่ทั้งสอง โดยการ Demultiplexing ใช้ในตอนที่ได้รับข้อมูลจากเน็ตเวิร์ก และการ Encapsulate ใช้ตอนที่ทำการส่งข้อมูลในทุกเลเยอร์ของโปรโตคอล

### 2.2.5 โพรโตคอล IP (Internet Protocol)

IP เป็นโปรโตคอลที่ทำหน้าที่รับภาระในการนำข้อมูลไปส่งยังจุดหมายปลายทางไม่ว่าที่ใดๆ ในอินเทอร์เน็ตโปรโตคอลต่างๆ ใน TCP/IP Suite ทั้ง TCP, UDP, ICMP ต่างก็ต้องอาศัยระบบนี้ทั้งสิ้น เนื่องจากตัวโปรโตคอล IP มีกลไกที่ค่อนข้างฉลาดในการหาเส้นทางขนส่งข้อมูล

#### IP Addressing

ทุกอินเทอร์เน็ตเฟซที่ต่ออยู่บนอินเทอร์เน็ตจะต้องมีหมายเลขประจำตัวเพื่อใช้ในการสื่อสารข้อมูล เรียกว่า Internet Address หรือเรียกย่อๆว่า IP Address โดยค่า IP Address นี้จะเป็นหมายเลขจำนวน 32 บิต แต่แทนที่จะกำหนดให้เลขทั้ง 32 บิตนั้นถูกนับต่อเนื่องกันไป ก็จะใช้วิธีการแบ่งหมายเลขดังกล่าวออกเป็นกลุ่มของเลขขนาด 8 บิตจำนวน 4 ชุด และคั่นแต่ละชุดด้วยจุด ตัวอย่างเช่น 172.17.3.12 นอกจากนี้ใน IP Address นั้นยังถูกแบ่งออกเป็น 2 ส่วนคือ ส่วนที่เป็นแอดเดรสของเน็ตเวิร์ก (Network ID) และส่วนที่เป็นแอดเดรสของโฮสต์ (Host ID) ซึ่งข้อมูลในส่วนนี้จะถูกใช้สำหรับ ค้นหาเส้นทางของ IP ในการที่จะขนส่งข้อมูลจากต้นทางให้ถึงปลายทางอย่างถูกต้อง เพื่อเป็นการกำหนดขนาดของเน็ตเวิร์ก สำหรับ IP Address ต่างๆ ดังนั้นจึงมีการจัด IP Address ในแต่ละช่วงออกเป็นคลาส (class) ต่างๆกันจาก A ถึง E เพื่อจะได้ทำการจัดสรร IP Address ได้อย่างเหมาะสมกับขนาดของเน็ตเวิร์ก



ภาพที่ 2.6 การกำหนด IP Address ในคลาสต่างๆ

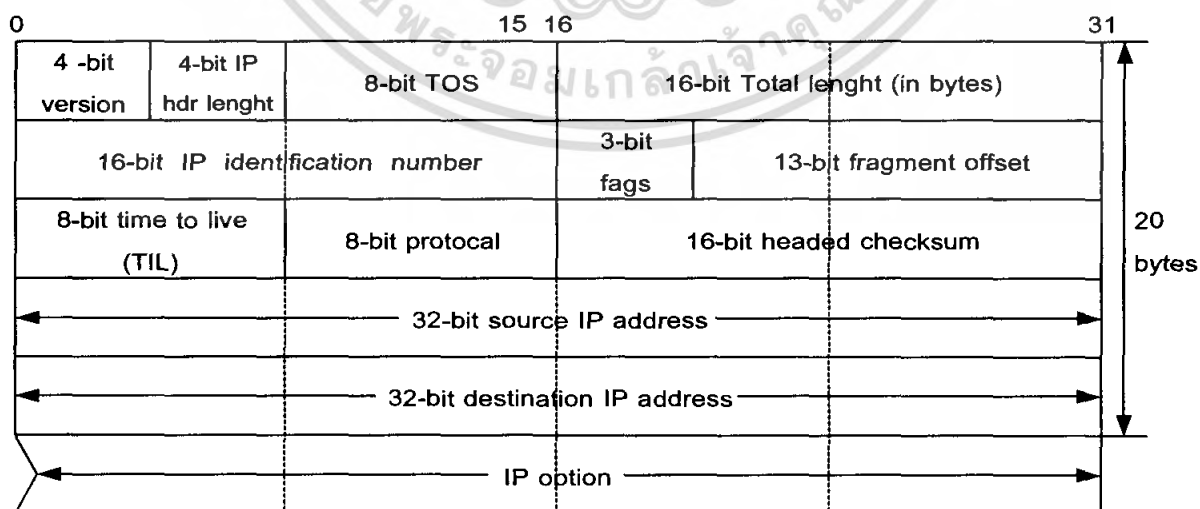
จากข้อกำหนดในการแบ่งคลาสของ IP Address หากลองนำบิตที่อยู่ในตอนต้นของ IP Address ในแต่ละคลาสมาแปลงเป็น IP address ในเลขฐานสิบ จะเห็นว่าแต่ละคลาสครอบคลุม IP Address ช่วงต่างๆ ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงช่วงของ IP Address แต่ละคลาส

Class	IP Range
A	0.0.0.0 - 127.255.255.255
B	128.0.0.0 - 191.255.255.255
C	192.0.0.0 - 223.255.255.255
D	224.0.0.0 - 239.255.255.255
E	240.0.0.0 - 255.255.255.255

IP Header

เมื่อข้อมูลถูกส่งลงมาจากชั้น Transport Layer สู่ชั้น Network Layer กระบวนการ Encapsulate ของ IP Protocol จะทำการเพิ่มส่วน Header ลงไป Header ของ IP datagram มีขนาด 20-32 ไบต์ มีส่วนประกอบต่างๆ ดังแสดงในรูป



ภาพที่ 2.7 IP Header

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ตารางที่ 2.2 อธิบายตำแหน่งต่างๆ ของ IP Header

ตำแหน่ง	ชื่อ	อธิบาย	
0-3	Version	มีขนาด 4 บิตเป็นเวอร์ชันของ IP ปัจจุบันค่านี้ถูกกำหนดให้เป็น 4	
4-7	Length	มีขนาด 4 บิตเป็นค่าความยาวของ Header นี้ โดยปกติจะเป็น 5 หมายความว่า $5 \times 32$ บิต = 20 ไบต์	
8-15	Type of Service	เป็นข้อมูลขนาด 8 บิต ปัจจุบันไม่ได้ใช้งานแล้ว	
16-31	Total length	เป็นฟิลด์ที่บอกจำนวนไบต์ทั้งหมดของ IP Datagram ด้วยขนาด 16 บิตทำให้ Datagram มีขนาดสูงสุดไม่เกิน 65535 ไบต์ และมีขนาดเล็กสุดไม่ต่ำกว่า 512 ไบต์	
32-47	Identification	ใช้ในกรณีที่มีการแบ่งดาต้าแกรมออกเป็นแฟรกเมนต์ เมื่อนำกลับมารวมกันใหม่จะได้รู้ว่ามาจากดาต้าแกรมเดียวกัน	
48-50	Flag	ใช้ในกรณีที่มีการแบ่งข้อมูลออกเป็นแฟรกเมนต์ มีความหมายดังนี้	
		บิต 0 : reserved	เป็น 0 เสมอ
		บิต 1 (DF)	0 = May Fragment, 1 = Don't Fragment
บิต 2 (MF)	0 = Last Fragment, 1 = More Fragments		
51-63	fragment offset	เป็นส่วนระบุข้อมูลที่ใส่แยกรวมข้อมูล เพื่อให้ข้อมูลที่ถูกแยกออกเป็นแฟรกเมนต์กลับมารวมกันได้อย่างถูกต้องตามลำดับ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

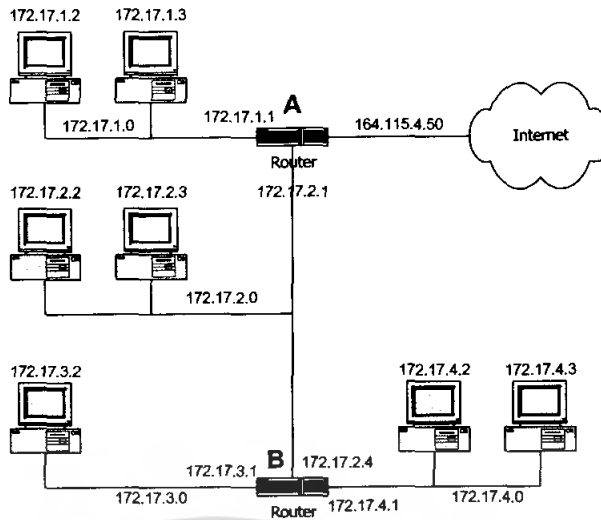
## ตารางที่ 2.2 อธิบายตำแหน่งต่างๆ ของ IP Header ( ต่อ )

ตำแหน่ง	ชื่อ	อธิบาย		
64-71	Time to Live (TTL)	เป็นจำนวนครั้งสูงสุดที่ดาต้าแกรมนี้จะถูกส่งผ่านเครือข่ายไปยังปลายทางได้ เพื่อ ป้องกันไม่ให้ดาต้าแกรมถูกเราต์ไปเรื่อยๆอย่างไม่สิ้นสุด ปกติค่านี้จะเริ่มต้นที่ 32 และจะถูกลดค่าลงทีละ 1 เมื่อมีการเราต์ จนค่านี้มีค่าเป็น 0 ก็จะไม่ถูกเราต์อีกต่อไป		
72-79	Protocol	เป็นข้อมูลที่ระบุโปรโตคอลที่ส่งดาต้าแกรมนี้มา ตัวอย่างโปรโตคอลที่ใช้บ่อยๆ ได้แก่		
		โปรโตคอล	ค่าในฟิลด์ Protocol	อธิบาย
		ICMP	1	Internet Control Message Protocol
		TCP	6	Transmission Control Protocol
		UDP	17	User Datagram Protocol
80-95	Header Checksum	เป็นส่วนตรวจสอบความถูกต้องของข้อมูลใน Header โดยไม่เกี่ยวกับส่วนข้อมูลที่อยู่ภายใน payload ค่านี้จะถูกคำนวณใหม่ทุกครั้งที่มีการเปลี่ยนแปลงข้อมูลใน Header ( เช่น TTL ที่มีการเปลี่ยนแปลงทุกครั้ง ที่ IP datagram ถูกส่งผ่านเราต์เตอร์ )		
86-127	Source IP Address	คือ IP Address ของผู้ส่งดาต้าแกรม		
128-163	Destination IP Address	คือ IP Address ของผู้รับดาต้าแกรม		
ไม่แน่นอน	Option	มีขนาดข้อมูลไม่แน่นอน ใช้สำหรับกำหนดค่าพารามิเตอร์ปลีกย่อย ซึ่งส่วนใหญ่ไม่มีการนำไปใช้งาน		
ขึ้นอยู่กับ Option	Padding	มีข้อมูลว่างเปล่า ใช้เป็นส่วนเติมเต็มของฟิลด์ Option ให้ครบ 32 ไบต์		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP Routing เป็นกระบวนการค้นหาเส้นทางในการส่งผ่านข้อมูลจากต้นทางไปยังที่หมายปลายทางโดยผ่านการส่งต่อของอุปกรณ์ IP ที่อยู่ในเน็ตเวิร์กซึ่งจะช่วยกันทำหน้าที่ส่งต่อข้อมูลไปจนกว่าข้อมูลจะถึงปลายทาง กลไกสำคัญที่ทำให้ IP เป็นโปรโตคอลสำหรับขนส่งข้อมูลไปยังทุกๆ ที่ในโลกอินเทอร์เน็ตที่ดีที่สุดขณะนี้คือการที่ IP มีกระบวนการ IP Routing นี้เอง สิ่งที่น่าสนใจที่สุดของ IP Routing คือการที่ต้นทางและปลายทางของการสื่อสารนั้นในบางโอกาสต่างก็อยู่กันแสนไกล การสื่อสารข้อมูลแต่ละครั้ง ข้อมูลจะต้องเดินทางผ่านโครงข่ายอันสลับซับซ้อนมากมาย แต่ในที่สุดข้อมูลก็สามารถส่งถึงกันได้ในเวลาอันรวดเร็ว โครงข่ายอินเทอร์เน็ตคงไม่อาจเกิดขึ้นได้หากไม่มีโปรโตคอล IP หลักการพื้นฐานของ IP Routing เริ่มต้นด้วยส่วนประกอบต่างๆ ของเน็ตเวิร์ค ในแง่ของ IP Routing กันก่อน

- Host โฮสต์เป็นอุปกรณ์ที่ทำหน้าที่ให้กำเนิดข้อมูลในกรณีเป็นผู้ส่ง หรือทำหน้าที่รับข้อมูลไปใช้งานในกรณีเป็นผู้รับ การสื่อสารข้อมูลใดๆ จะต้องเป็นการสื่อสารจากโฮสต์ไปยังโฮสต์เสมอ สำหรับ IP Packet แล้วข้อมูลในเฮดเดอร์ที่ปรากฏอยู่ในฟิลด์ Source Address และ Destination Address ซึ่งเรียกว่า IP Addresss จะเป็นหมายเลขระบุตำแหน่งของโฮสต์ต้นทางและโฮสต์ปลายทางเท่านั้น
- Router เราเตอร์เป็นอุปกรณ์สำคัญสำหรับ IP ที่จะทำให้การขนส่งข้อมูลเป็นไปอย่างสมบูรณ์ เราเตอร์ทำหน้าที่ในการส่งผ่านข้อมูลจากเน็ตเวิร์กหนึ่งไปยังอีกเน็ตเวิร์กหนึ่ง ตำแหน่งของเราเตอร์จะอยู่ในจุดที่เชื่อมต่อระหว่างสองเน็ตเวิร์กเข้าด้วยกัน ด้วยข้อกำหนดของ IP ข้อมูลจะส่งไปถึงกันโดยตรงข้ามเน็ตเวิร์กไม่ได้ จะต้องอาศัยเราเตอร์เป็นผู้ทำหน้าที่ส่งผ่านไปให้ ดังนั้นเน็ตเวิร์กของ IP ถึงแม้ไม่ได้ต่อกันในทางกายภาพแต่ก็สามารถสื่อสารกันได้โดยอาศัยเราเตอร์เป็นตัวเชื่อมประสานเข้าด้วยกัน
- Network เน็ตเวิร์คเป็นเครือข่ายที่มีการเชื่อมต่อกันของโฮสต์ 2 ตัวขึ้นไป โฮสต์แต่ละตัวในเน็ตเวิร์คเดียวกันสามารถเชื่อมต่อถึงกันได้โดยตรง



ภาพที่ 2.8 network ตัวอย่าง

การ Routing จะเป็นไปตามขั้นตอนดังนี้

1. ถ้าโฮสต์ต้นทางและปลายทางต่อเชื่อมรวมอยู่ในเน็ตเวิร์กเดียวกัน มีการเชื่อมต่อถึงกันโดยตรง เช่น อีเทอร์เน็ตหรือโทเค็นริง ดังแสดงในภาพที่ 2.3 เป็นการติดต่อระหว่าง 172.17.2.2 และ 172.17.2.3 (เครื่องสีแดง) IP ดาต้าแกรมก็จะถูกส่งไปยังโฮสต์ปลายทางโดยตรง

2. หากโฮสต์ต้นทางและปลายทางไม่ได้อยู่ในเน็ตเวิร์กเดียวกัน IP ดาต้าแกรมจะถูกส่งไปยังดีฟอลต์เราท์เตอร์

3. เมื่อเราเตอร์ได้รับ IP ดาต้าแกรมจากข้อ 2 แล้วตรวจสอบดู หากพบว่าโฮสต์ปลายทางต่อรวมอยู่บนเน็ตเวิร์กเดียวกันกับเราเตอร์ ให้ทำการส่งดาต้า แกรมไปที่โฮสต์นั้น เช่น หาก 172.17.3.2 ต้องการส่งดาต้าแกรมไปยัง 172.17.4.2 (เครื่องสีเหลือง) จะต้องส่งดาต้าแกรมไปที่ Router B Router B จะส่งดาต้าแกรมต่อไปยังโฮสต์ปลายทาง

4. หากไม่ได้ต่อรวมกันก็ส่งดาต้าแกรมไปที่เราเตอร์ตัวต่อไป โดย Router จะเป็นผู้เลือกเส้นทาง ซึ่งมีอยู่ 2 กรณีคือ

ถ้ามีข้อมูลของโฮสต์ปลายทางอยู่ใน Routing Table Router จะส่งดาต้าแกรมไปยัง router ตัวที่ระบุไว้ใน routing table

ถ้าไม่มีข้อมูลของโฮสต์ปลายทางอยู่ใน Routing Table Router จะส่งดาต้าแกรมไปยัง default router และกลับไปขั้นตอนในข้อ 3 ใหม่ จนกว่า IP ดาต้าแกรมจะเดินทางถึงปลายทางหรือหมดเวลาในการส่ง (TTL=0)

สมมติว่าเครื่อง 172.17.1.3 ต้องการติดต่อกับ 172.17.4.3 จะต้องส่ง ip datagram ไปยัง Router A หาก Router A มีข้อมูลเกี่ยวกับ 172.17.4.3 อยู่ ก็จะมีรู้ว่าต้องส่งดาต้าแกรมไปยัง Router B คือ 172.17.2.4 และ Router B ก็จะส่ง ip datagram ไปยังโฮสต์ปลายทางได้สำเร็จ

### Subnet Addressing / Subnet Mask

ในการใช้งาน โพรโทคอล TCP/IP ใน internet นั้นการแบ่ง IP Address ออกเป็น แอดเดรสของเน็ตเวิร์ก (net id) และแอดเดรสของ โฮสต์ ตามที่ระบุของแต่ละคลาสก่อนข้างจะ ขาดประสิทธิภาพ คือในเน็ตเวิร์กคลาส A และ B แต่ละเน็ตเวิร์กนั้น สามารถมีจำนวนโฮสต์ได้ มาก ซึ่งการที่จะนำ IP Address มาใช้อย่างทั่วถึงนั้นมีโอกาสเป็นไปได้ยากมากทั้งคลาส A และ คลาส B เพราะมีโอกาสน้อยมากที่จะมีเน็ตเวิร์ก ใดในโลกมีจำนวนโฮสต์มากมายขนาดนั้นอยู่ ภายในเน็ตเวิร์กเดียว ดังนั้น IP Address ที่จัดสรรให้ไปในแต่ละเน็ตเวิร์กของ คลาสเหล่านี้จึง ถูกใช้ไม่หมดและไม่สามารถนำไปใช้ประโยชน์อื่นได้เลย ดังนั้นเพื่อให้การจัดสรร ip เป็นไป อย่างมีประสิทธิภาพ จึงมีการนำส่วนของ host id มาแบ่งย่อยเป็นสองส่วนคือ subnet id และ host id ทำให้ได้เน็ตเวิร์กย่อยหลายๆเน็ตเวิร์ก โดยในแต่ละเน็ตเวิร์ก มีจำนวนโฮสต์ไม่มาก เกินไปและเพียงพอต่อการใช้งาน

### 2.2.6 โพรโทคอล TCP (Transmission Control Protocol)

TCP เป็นโพรโทคอลที่ใช้สื่อสารระหว่างโฮสต์ที่มีความน่าเชื่อถือ จะเห็นได้ว่าโพรโทคอล ในระดับ IP หรือแม้กระทั่ง UDP จะสนใจ ข้อมูลเพียง 1 ดาต้าแกรม กลไกของโพรโทคอลจะมี หน้าที่ตรวจสอบความถูกต้องเพียงเฉพาะดาต้าแกรมนั้น ๆ เมื่อจะทำ การส่งดาต้าแกรมใหม่ก็ จะถือว่าเป็นข้อมูลชุดใหม่ที่ไม่มีความสัมพันธ์ใด ๆ กับข้อมูลดาต้าแกรมอื่น (การสื่อสาร 1 ครั้ง จึงใช้เพียง 1 ดาต้าแกรม) แต่สำหรับ TCP แล้วจะเห็นว่าข้อมูลนั้นเป็น stream คือมี ความสัมพันธ์ต่อเนื่องกัน มีกลไกในการตรวจสอบทั้งด้านส่ง และด้านรับเพื่อให้แน่ใจว่าสามารถ สื่อสารกันได้จริงจึงจะมีการส่งรับข้อมูลเกิดขึ้น ตลอดจนการยกเลิกการติดต่อก็มีกลไกสำหรับ แจ้งให้อีกฝั่งทราบ ทำให้การสื่อสารด้วย TCP จึงเสมือนว่าทั้ง 2 ฝ่ายคือฝ่ายรับและฝ่ายส่งได้ทำ การต่อสาย เน็ตเวิร์กถึงกัน (connected) ตลอดเวลาที่มีการรับส่งข้อมูลจนกระทั่งการสื่อสาร ทั้งหมดเสร็จสิ้นจึงจะทำการยกเลิกการเชื่อมต่อนั้นเสีย

จุดเด่นประการสำคัญของ TCP ที่กล่าวถึงอยู่เสมอคือ ความมีเสถียรภาพและความ ถูกต้องของการสื่อสารซึ่งมีความเชื่อถือได้สูง คุณสมบัติที่ทำให้ TCP มีข้อดีดังกล่าวคือ

1. ข้อมูลที่จะส่งผ่าน TCP จะถูกนำมาแตกย่อยออกเป็นส่วน ๆ ให้มีขนาดเหมาะสม สำหรับการส่งข้อมูล โดย TCP มีกลไกในการพิจารณาว่าขนาดเท่าใดจะทำให้การรับ - ส่งนั้นมี ประสิทธิภาพและน่าเชื่อถือสูงสุด โดยข้อมูลแต่ละชุดที่แบ่งออกและ ทำการส่งโดย TCP แต่ละ ครั้งจะเรียกว่า TCP เซกเมนต์

2. ในการส่งข้อมูลแต่ละครั้ง TCP จะมีการจับเวลาไว้เสมอ เพื่อรอการตอบรับจากผู้รับ ว่าได้รับข้อมูลถูกต้อง หากหมดเวลาแล้วไม่มีการตอบรับ TCP จะถือว่าข้อมูลไปไม่ถึงและทำ การแก้ปัญหาที่เกิดขึ้น เช่น ยกเลิกการติดต่อ , ส่งข้อมูลซ้ำ ทำให้ Application ทราบสถานะการ ส่งข้อมูลตลอดเวลา

3. TCP มี checksum ซึ่งจะครอบคลุมทั้ง TCP Header และ TCP Data เพื่อเป็นการป้องกันและตรวจสอบว่าข้อมูลที่ส่งมานั้นถูกต้อง และไม่ได้ถูกแก้ไขระหว่างทาง หาก TCP ได้รับข้อมูลที่ทำการตรวจสอบกับ checksum แล้วปรากฏว่า มีความผิดพลาดเกิดขึ้น TCP จะทิ้งข้อมูลที่ได้รับและจะไม่ทำการตอบรับข้อมูลนั้นกลับไปยังผู้ส่ง คือถือเสมือนว่าไม่ได้รับ ข้อมูลนั้นเพื่อให้ทางฝ่ายผู้ส่งทำการส่งใหม่หรือหาข้อบกพร่องและพยายามแก้ไขตามแต่แอปพลิเคชันทางฝ่ายผู้ส่งเห็นสมควร

4. เนื่องจาก TCP อาศัย IP ในการส่งข้อมูล ซึ่ง IP เองอาจจะถูกแฟรกเมนต์ได้ และทำให้ข้อมูลที่ถูกแฟรกเมนต์นั้นส่งถึง ปลายทางในลำดับที่ไม่ถูกต้องได้ หน้าที่ของ TCP เมื่อรับข้อมูลที่แฟรกเมนต์มานั้นจะต้องนำข้อมูลแต่ละส่วนมาประกอบ รวมกันให้ถูกต้องสมบูรณ์ก่อนจะส่งไปยัง Application Layer ต่อไป

5. การส่ง - รับข้อมูลด้วย IP อาจจะมีกรณีที่ IP Datagram นั้นถูกส่งซ้ำขึ้นได้ TCP ที่รับข้อมูลซ้ำดังกล่าวจะต้องทราบว่า เป็น IP Datagram ที่ซ้ำและไม่นำข้อมูลไปใช้งาน

6. TCP มีกลไกควบคุมการไหลของข้อมูล (Flow Control) โดยการควบคุมนี้จะต้องอาศัยลำดับของการรับส่งที่ถูกต้อง และสัมพันธ์กันทั้ง 2 ฝ่าย ในขณะที่เดียวกันข้อมูลที่ส่งนั้นจะต้องอาศัย IP หลายดาด้าแกรมจึงจะได้รับข้อมูลครบทั้งหมด ดังนั้นในการรับข้อมูลทางฝ่ายรับจึงต้องเตรียมบัฟเฟอร์ไว้จำนวนหนึ่งเพื่อรอรับข้อมูลและรวบรวมข้อมูลทั้งหมดให้อยู่ในบัฟเฟอร์ก่อนที่จะทำการจัดเรียงข้อมูล ตรวจสอบความถูกต้องแล้วจึงส่งต่อไปยังแอปพลิเคชันด้วยเหตุผลดังกล่าวจะเห็น ได้ว่าขนาดของข้อมูลมิได้ถูกจำกัดที่ขนาดของดาด้าแกรมใด ๆ ข้อมูลที่ส่งอาจจะมีความยาวมากอยู่ในหลายดาด้าแกรม ก็เป็นได้ ดังนั้นเพื่อป้องกันการส่งข้อมูลขนาดใหญ่เร็วเกินไปจนทำให้ทางฝ่ายรับไม่มีหน่วยความจำเพียงพอที่จะเป็น บัฟเฟอร์ที่พักข้อมูล การส่งข้อมูลจึงถูกจำกัดโดยจะอนุญาตให้ทำการส่งข้อมูลได้ เท่าที่ฝ่ายรับมีบัฟเฟอร์เพียงพอเท่านั้น

#### TCP Header

ใน TCP Header จะเริ่มต้นระบุที่หมายเลข Port ต้นทางและหมายเลข Port ปลายทาง แต่อันที่จริงแล้วข้อมูลอีกส่วนหนึ่งที่ใช้ในการสื่อสาร คือ IP Address ของต้นทางและปลายทาง ก็ต้องระบุด้วยเช่นกัน แต่ได้ถูก Encapsulate ไว้ใน Layer ของ IP และค่าของ IP Address ทั้งคู่จะอยู่ใน IP Header ส่วนคู่ของ IP Address และหมายเลข Port นั้นจะเรียกว่า Socket ซึ่งในการสื่อสารแต่ละครั้งจะต้องมีทั้ง Socket ของต้นทางและปลายทางข้อมูลจึงจะถูกส่ง-รับไปถูกที่ และถูก Application รายละเอียดมีดังนี้

16-bit source port number				16-bit destination port				
32-bit sequence number								
32-bit acknowledge number								
header length	6-bit reserved	URG	ACK	PUSH	RESET	SYN	FIN	16-bit windows size
16-bit TCP checksum				16-bit urgent size				
TCP Option								
Data								

ภาพที่ 2.9 TCP Header

ตารางที่ 2.3 รายละเอียด TCP Header

ชื่อ	อธิบาย
Source Port Number	หมายถึงพอร์ตที่โฮสต์ต้นทางใช้ในการสื่อสารกันของเซสชันนี้ และ TCP/IP จะใช้พอร์ตนั้นไป ตลอดตรวจดูที่การสื่อสารในเซสชันนี้ยังไม่ยุติลง โดยทั่วไปพอร์ตนี้จะเรียกว่า "ไคลเอนต์พอร์ต" คือพอร์ตที่ไคลเอนต์เปิดขึ้น มาเพื่อรอการตอบรับจากเซิร์ฟเวอร์ (พิจารณาจากทิศทางของแพ็กเก็ตที่ส่งมาจากไคลเอนต์ไปยังเซิร์ฟเวอร์) ไคลเอนต์พอร์ต จะมีหมายเลขไม่แน่นอนและเปลี่ยนไปทุกครั้งที่มีการเริ่มการเชื่อมต่อใหม่ เป็นพอร์ตที่ถูกเปิดไว้ในระยะเวลาสั้น ๆ (ephemeral port) ค่าที่เป็นไปได้ของพอร์ตนี้ขึ้นอยู่กับการจัดสรรของระบบปฏิบัติการ ในการกำหนดขอบเขตของพอร์ตเหล่านี้ส่วนใหญ่จะมีค่า อยู่ในช่วง 1024 – 5000
Destination Port Number	หมายถึงหมายเลขพอร์ตบนโฮสต์ปลายทางที่โฮสต์ต้นทางต้องการติดต่อด้วย โดยนัยแล้วจะ หมายถึงแอปพลิเคชันที่ให้บริการอยู่พอร์ตนั้นที่โฮสต์ปลายทางนั่นเอง พอร์ตนั้นจะเรียกอีกอย่างหนึ่งว่า "เซิร์ฟเวอร์พอร์ต" หมายเลขพอร์ตที่เปิดไว้จะขึ้นอยู่กับแอปพลิเคชันที่ให้บริการ โดยทั่วไปแอปพลิเคชันแต่ละประเภทจะมีหมายเลขพอร์ต เป็นมาตรฐานสำหรับให้ไคลเอนต์ได้เรียกใช้บริการ
Sequence Number	เป็นฟิลด์ที่ระบุถึงหมายเลขลำดับที่ใช้อ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อให้ทั้ง 2 ฝ่าย จะได้รับทราบตรงกันว่าเป็นข้อมูลของชุดใด การนำไปใช้งานจะได้ไม่ปะปนกัน และมีลำดับที่ถูกต้อง เนื่องจากการสื่อสารข้อมูลผ่าน TCP นั้นจังหวะและลำดับเป็นส่วนสำคัญของโปรโตคอลไม่ยิ่งหย่อนไปกว่าข้อมูลใน TCP Header รวมไปถึง การที่ข้อมูลในแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	TCP Segment อาจจะถูกทำการแฟร็กเมนต์ในเลเยอร์ของ IP ถัดลงไป ทำให้ข้อมูลถูกแบ่งออกและส่งไปในลำดับที่ไม่เรียงกัน หากไม่มี จุดอ้างอิงของข้อมูลก็จะไม่สามารถอ่านข้อมูลกลับใหม่ได้อย่างสมบูรณ์ และถูกต้อง การส่งข้อมูลและการตอบรับจะใช้ฟิลด์ นี้เป็นตัวยืนยัน ระหว่างกันเสมอ
Acknowledge Number	ทำหน้าที่เช่นเดียวกับ Sequence Number ต่างกันตรงที่เป็น Sequence Number ซึ่งในการตอบรับ กล่าวคือ เนื่องจาก Sequence Number ที่ใช้ในการอ้างอิงนั้นผู้ที่เริ่มส่งข้อมูลจะเป็นผู้กำหนดเลขขึ้น มาและส่งไป พร้อมกับการสร้างการเชื่อมต่อครั้งใหม่แต่สำหรับฝ่ายที่ถูกติดต่อก็ จำเป็นต้องกำหนดหมายเลขสำหรับใช้อ้างอิง ในการตอบรับเช่นกัน ค่าที่อยู่ใน Acknowledge Number ก็คือหมายเลขที่ใช้อ้างอิงในการตอบรับนี้

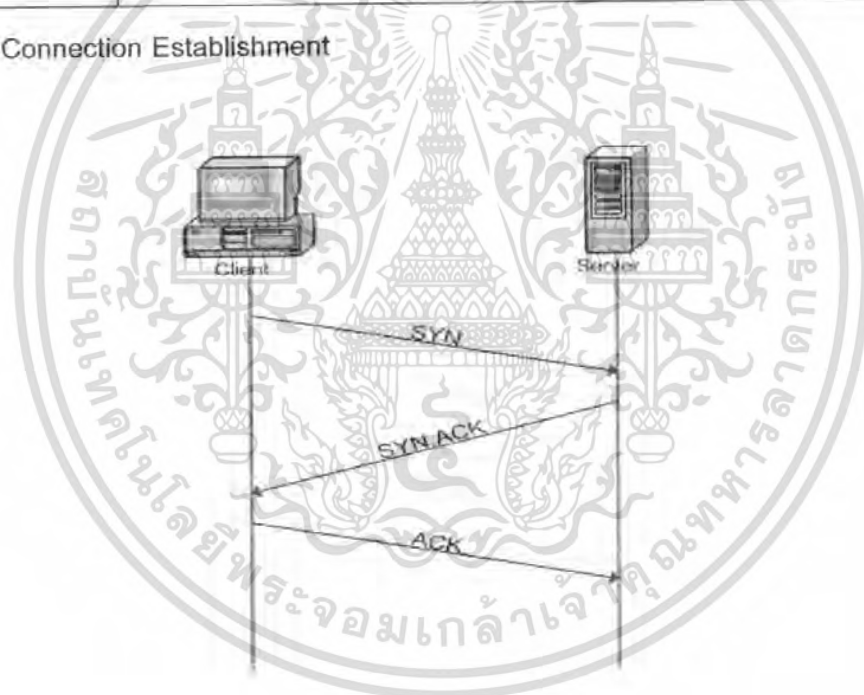
### ตารางที่ 2.3 รายละเอียด TCP Header ( ต่อ )

Header Length	โดยปกติความยาวของ TCP Header จะเท่ากับ 20 ไบต์ แต่ถ้าหากมีการใช้ Option อาจจะทำให้ ขนาดของเฮดเดอร์ยาวขึ้นตามข้อมูลที่ต้องเพิ่มมาจาก Option นั้น แต่ทั้งหมดแล้วจะไม่เกิน 60 ไบต์	
Flag	เป็นข้อมูลในระดับบิตที่ใช้เป็นตัวบอกคุณสมบัติของ TCP Segment ที่กำลังส่งอยู่นั้น และใช้เป็นตัวควบคุมจังหวะ การรับส่งข้อมูลด้วย ซึ่ง Flag ทั้งหมดมีอยู่ 6 บิต แต่ละบิตมีชื่อและมีความหมายดังนี้	
	<b>Flag</b>	<b>อธิบาย</b>
	URG	ใช้บอกความหมายว่าเป็นข้อมูลด่วน และมีข้อมูลพิเศษมาด้วย (อยู่ใน Urgent pointer)
	ACK	แสดงว่าข้อมูลในฟิลด์ Acknowledge Number นำมาใช้งานได้
	DSH	เพื่อแจ้งให้ผู้รับข้อมูลทราบว่า ควรจะส่งข้อมูล Segment นี้ไปยังโพรเซสที่กำลังรออยู่ที่
	RST	ใช้ในกรณีที่เกิดการสับสน ขึ้นด้วยเหตุผลต่างๆ เช่น โฮสต์มีปัญหา ให้เริ่มต้นสื่อสารกันใหม่
	SYN	ใช้ในการเริ่มต้นขอติดต่อกับปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	FIN	ใช้ส่งเพื่อแจ้งให้ปลายทางทราบว่ายุติการติดต่อ
Window Size		เป็นขนาดของการรับ – ส่งข้อมูลในแต่ละครั้งที่ทางฝ่ายผู้รับจะสามารถรับได้ เนื่องจากในการรับข้อมูลนั้น ทางผู้รับจะต้องจัดเตรียมหน่วยความจำในการพักข้อมูลที่มาจาก TCP และทำการ Demultiplex ออกมา หากไม่มีการตกลง ถึงขนาดที่ทางฝ่ายผู้รับสามารถรับได้ ก็จะทำให้การสื่อสารข้อมูลไม่สมดุล และฝ่ายรับอาจจะประมวลผลทัน ซึ่งจะส่งผลให้ต้องส่ง ข้อมูลซ้ำหลายครั้ง
Checksum		ฟิลด์ที่ใช้ในการตรวจสอบความถูกต้องของข้อมูลใน TCP เชกเมนต์ใช้ระบุหมายเลข

### Connection Establishment



ภาพที่ 2.10 Connection Establishment

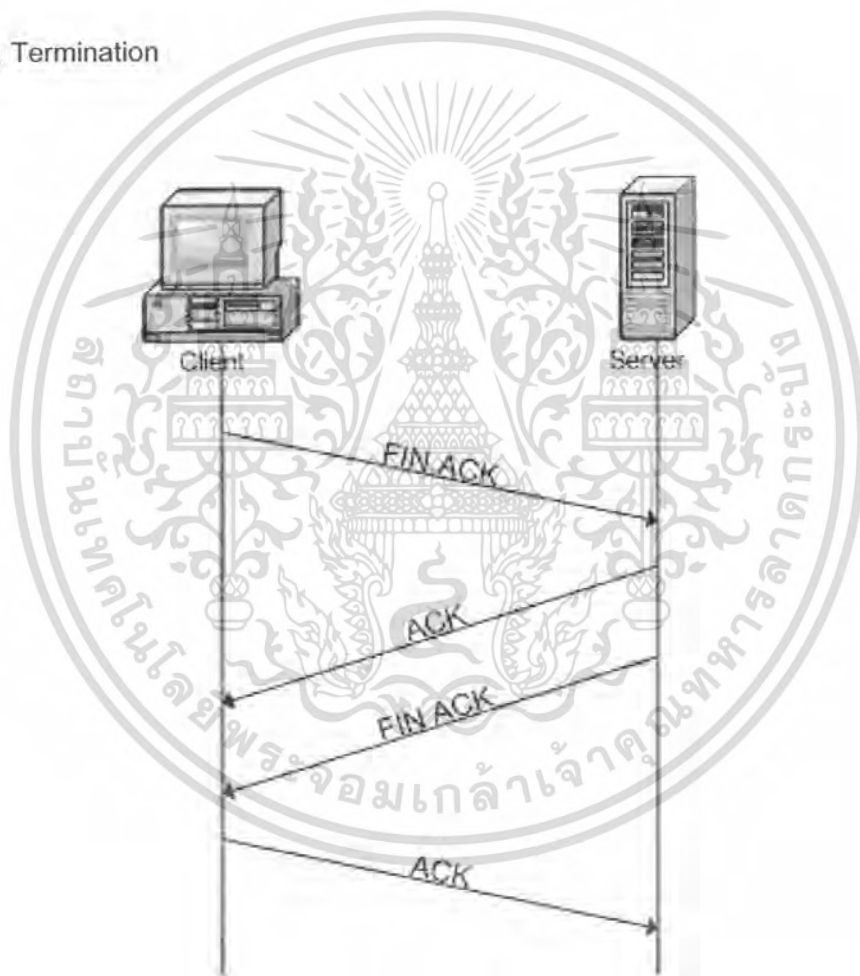
ก่อนที่จะเริ่มต้นการสื่อสาร จะต้องมีการส่งสัญญาณเพื่อบอกโฮสต์อีกฝั่งหนึ่งให้เตรียมตัวติดต่อ ซึ่งกระบวนการที่ใช้มีชื่อเรียกว่า 3-Way Hand Shake มีขั้นตอนคือ

1. เครื่องไคลเอนต์จะทำการส่งเซกเมนต์ โดยเปิด SYN Flag ระบุหมายเลขพอร์ตที่ต้องการติดต่อบนเซิร์ฟเวอร์และระบุหมายเลข ลำดับของข้อมูล (ISN - Initial Sequence Number)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เครื่องเซิร์ฟเวอร์เมื่อได้รับข้อมูลเซกเมนต์จากข้อ 1 ก็จะตอบกลับด้วยการเพิ่มค่า ISN ที่ได้รับขึ้นอีก 1 พร้อมทั้งระบุหมายเลขลำดับ (ISN) ของตนเอง และเปิด SYN กับ ACK Flag
3. ไคลเอนต์เมื่อได้รับการตอบกลับจากเซิร์ฟเวอร์ตามข้อ 2 ก็จะทำการตอบรับกลับไป โดยการเพิ่มค่า ISN ของเซิร์ฟเวอร์ขึ้นอีก 1 และเปิด ACK Flag เมื่อผ่านการสร้าง connection ทั้ง 3 ขั้นตอนแล้ว ตอนนี้ทั้งไคลเอนต์ และเซิร์ฟเวอร์เปรียบเสมือนมีการเชื่อมต่อถึงกันแล้ว สถานะของการเชื่อมต่อในขณะนี้เรียกว่า Established

### Connection Termination



ภาพที่ 2.11 Connection Termination

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อการสื่อสารของทั้งสองฝั่งจบลง และไม่ต้องการรับส่งข้อมูลอีกต่อไป จะต้องทำตามขั้นตอนการยุติการสื่อสารเพื่อให้การสื่อสารจบลงอย่างสมบูรณ์ ซึ่งมีอยู่ 4 ขั้นตอนคือ

1. ไคลเอนต์ทำการส่ง ISN พร้อมกับ FIN ACK Flag ไปยังเซิร์ฟเวอร์
2. เซิร์ฟเวอร์ทำการตอบรับ ISN และบวกค่า ISN อีก 1 พร้อม ACK Flag
3. เซิร์ฟเวอร์ทำการส่ง ISN พร้อมกับ FIN ACK Flag ไปยังไคลเอนต์
4. ไคลเอนต์ทำการตอบรับ ISN และบวกค่า ISN อีก 1 พร้อม ACK Flag

การยุติการเชื่อมต่อ โดยส่ง FIN ACK ออกไปมีความหมายคือ โฮสต์ที่ส่งไม่มีข้อมูลจะส่งไปอีก มิใช่ต้องการปิดการสื่อสารทั้งหมดในทันที ดังนั้นจึงต้องทำทั้งสองทาง การสื่อสารจึงจะยุติลงอย่างสมบูรณ์ ในการใช้งานจริง อาจมีการยุติการสื่อสารเพียงด้านเดียว คือหยุดส่งข้อมูลแต่ยังคงเปิดพอร์ตไว้รอรับข้อมูลจากอีกด้านหนึ่ง ทั้งนี้ขึ้นอยู่กับลักษณะการใช้งาน การปิดพอร์ตสื่อสารเพียงด้านเดียวเช่นนี้ เรียกว่า Half-Close

## 2.2.7 โพรโตคอล HTTP ( Hiper text transfer protocol )

โพรโตคอล HTTP เป็นโพรโตคอลพื้นฐานของเว็บไซต์เว็บที่ใช้ติดต่อกันระหว่างเว็บเซิร์ฟเวอร์และโปรแกรมบราวเซอร์, HTML ซึ่งเป็นภาษาพื้นฐานที่ใช้ในการสร้างเว็บเพจ และสุดท้ายเป็นเรื่องที่เกี่ยวกับ CGI ซึ่งเป็นแนวทางการพัฒนาโปรแกรมหรือภาษาสคริปต์ต่างๆ เพื่อร่วมใช้งานกับเว็บไซต์เว็บ และรายละเอียดเหล่านี้สามารถใช้เป็นพื้นฐานในการศึกษาส่วนที่ซับซ้อนมากขึ้น HTTP (ซึ่งเป็นส่วนย่อยของโพรโตคอล TCP/IP) เป็นกลไกหรือโพรโตคอลหลักที่ใช้แลกเปลี่ยนข้อมูลกันระหว่างเซิร์ฟเวอร์และไคลเอนต์ของเว็บไซต์เว็บ โดยถูกออกแบบมาให้มีความกะทัดรัด สามารถทำงานได้รวดเร็ว มีกระบวนการทำงานที่ไม่ซับซ้อน และมีคำสั่งที่ใช้งานไม่มากนัก แต่สามารถรองรับข้อมูลได้ทุกแบบ ไม่ว่าจะเป็นข้อมูลทั่วไปที่เข้ารหัสแบบ MIME หรือข้อมูลที่เป็นกราฟิก เช่น ไฟล์ที่เป็น GIF หรือ JPEG เป็นต้น

### โครงสร้างข้อมูลของ HTTP

โครงสร้างข้อมูลของ HTTP จะแบ่งออกเป็น 2 ส่วนใหญ่ๆคือ ส่วนเฮดเดอร์หรือเรียกว่า metadata จะเป็นส่วนเก็บข้อมูลที่ต้องใช้ภายในโพรโตคอล ส่วนที่สองเป็นส่วนข้อมูลจริงที่ต้องการรับส่ง ทั้งนี้ HTTP ถูกออกแบบมาให้สามารถรับส่งข้อมูลผ่าน ProxyหรือFirewall ต่างๆได้ โดยการทำงาน HTTP จะอาศัยโพรโตคอลพื้นฐาน TCP/IP ซึ่งทั่วไปจะใช้หมายเลขพอร์ตที่ 80

โพรโตคอล HTTP ในปัจจุบันได้พัฒนาขึ้นมาเป็นเวอร์ชัน 1.1 (จากเดิมคือ เวอร์ชัน 1.0) ซึ่งโปรแกรมบราวเซอร์ที่แพร่หลายทั่วไปนั้นจะสามารถรองรับโพรโตคอลในเวอร์ชันใหม่นี้ได้ และได้กำหนดไว้เป็นมาตรฐานใน RFC 2068 แล้ว โดยใน HTTP เวอร์ชัน 1.1 นี้ได้เพิ่มประสิทธิภาพการทำงานให้สูงขึ้น และปรับปรุงในด้านต่างๆที่ทำให้มีความสามารถมากขึ้นดังนี้

- ลดภาระของการเชื่อมต่อผ่านโพรโตคอล TCP และสามารถประสิทธิภาพของ TCP ได้เต็มที่
- สามารถทำการบีบอัดข้อมูลที่รับส่งระหว่างเซิร์ฟเวอร์และไคลเอนต์ได้

- รองรับการทำงานแบบ virtual host หมายถึง เว็บเซิร์ฟเวอร์เครื่องหนึ่งๆ มีชื่อโดเมนมากกว่าหนึ่งชื่อได้
- สามารถรองรับการทำงานได้หลายภาษา
- โอนไฟล์ข้อมูลเฉพาะบางส่วนได้ ซึ่งคุณสมบัตินี้จะมีประโยชน์มากในกรณีที่มีการโอนไฟล์ข้อมูลขนาดใหญ่ และเกิดปัญหาขึ้นระหว่างการทำงาน ซึ่งโปรโตคอล HTTP 1.1 มีจุดเด่นที่สามารถตรวจสอบได้ และโอนไฟล์ข้อมูลต่อจากส่วนที่เคยโอนมาแล้วได้

#### คำสั่งของ โปรโตคอล HTTP

HTTP มีคำสั่งต่างๆไม่มากนัก เพื่อให้สามารถใช้งานได้อย่างสะดวกและรวดเร็ว โดยมีคำสั่งที่ใช้งานแพร่หลายอยู่เพียง 3 คำสั่ง คือ GET, HEAD และ POST ส่วนคำสั่งอื่นๆอีก 4 คำสั่ง คือ PUT, DELETE, LINK และ UNLINK มีให้ใช้งานเช่นกัน แต่ไม่เป็นที่นิยมมากนัก รายละเอียดคำสั่งของ HTTP มีดังนี้

ตารางที่ 2.4 รายละเอียดคำสั่งของ HTTP

คำสั่ง	รายละเอียด
GET	ใช้อ่านข้อมูลจากเว็บเซิร์ฟเวอร์และส่งไปยังไคลเอนต์โดยมีรูปแบบดังนี้ GET <URL>HTTP/1.0 ตัวอย่างเช่น ต้องการให้เว็บเซิร์ฟเวอร์ส่งไฟล์ sale.html จากโดเมน www.netcorp.com ไปยังไคลเอนต์จะใช้รูปแบบของคำสั่ง GET ดังนี้ GET www.netcorp.com/sale.html/1.0 นอกจากนี้คำสั่ง GET ยังสามารถกำหนดเงื่อนไขให้อ่านข้อมูลจากเว็บเซิร์ฟเวอร์เฉพาะที่มีการเปลี่ยนแปลงแก้ไขได้ด้วย
HEAD	คำสั่งนี้จะทำงานคล้ายกับคำสั่ง GET แต่เว็บเซิร์ฟเวอร์จะส่งข้อมูลกลับมาให้เฉพาะในรายละเอียดของ metadata หรือข้อมูลในเฮดเดอร์เท่านั้น ส่วนข้อมูลที่เป็น HTML จะไม่ถูกส่งมาด้วย ซึ่งคำสั่ง HEAD นี้จะใช้เพื่อทดสอบว่าข้อมูลตาม URL นั้น มีการเปลี่ยนแปลงหรือไม่เท่านั้น
POST	เป็นคำสั่งที่ตรงข้ามกับคำสั่ง GET และ HEAD โดยทำหน้าที่ส่งข้อมูลจากไคลเอนต์ไปยังเว็บเซิร์ฟเวอร์นั้นจะไม่ค่อยมีใช้งาน นอกจากในกรณีที่ HTML ทำงานในลักษณะที่ให้ผู้ใช้อกรอกข้อมูลตามแบบฟอร์ม (เช่น รายละเอียดส่วนตัวของผู้ใช้งาน) และส่งข้อมูลนี้กลับมาเก็บที่เว็บเซิร์ฟเวอร์
PUT	เป็นคำสั่งที่ทำงานเหมือนกับคำสั่ง POST แต่ไม่เป็นที่นิยม
DELETE	เพื่อให้ไคลเอนต์สั่งเว็บเซิร์ฟเวอร์ลบ URL ที่กำหนดไว้ออกจากเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	แต่เป็นคำสั่งที่ไม่นิยมใช้มากนัก เนื่องจากเว็บเซิร์ฟเวอร์ทั่วไปมักจะทำงานในแบบอ่านข้อมูลได้เท่านั้น (read-only)
LINK	เป็นคำสั่งที่เชื่อม URL ที่ต้องการไปยังเว็บเซิร์ฟเวอร์อื่น
UNLINK	ยกเลิกคำสั่ง LINK ให้กลับมาใช้เว็บเซิร์ฟเวอร์เดิมตามที่กำหนดไว้ใน URL

### สถานะการทำงานของ HTTP

โปรโตคอล HTTP ได้กำหนดรหัสแสดงสถานะการทำงานของโปรโตคอลไว้ โดยแบ่งกลุ่มของรหัสสถานะออกไว้เป็น 5 กลุ่มคือ

#### ตารางที่ 2.5 รหัสแสดงสถานะการทำงานของโปรโตคอล HTTP

รหัสสถานะ	ประเภท	รายละเอียด
100 - 199	Information	เป็นรหัสสถานะกลุ่มที่เปิดให้โปรแกรมประยุกต์ต่างๆกำหนดใช้งานได้เอง
200 - 299	Successful	กลุ่มรหัสที่แสดงว่าการทำงานสำเร็จ
300 - 399	Redirection	กลุ่มรหัสนี้จะใช้ภายในโปรโตคอล HTTP เอง โดยเป็นการทำงานที่ต่อเนื่องมาจากโปรเซสก่อนหน้า ซึ่งโคลเอนต์เป็นผู้ส่งงาน
400 - 499	Client Error	ใช้แสดงการปัญหาที่เกิดขึ้นกับโคลเอนต์
500 - 599	Server Error	ใช้แสดงการปัญหาที่เกิดขึ้นกับเซิร์ฟเวอร์

### 2.2.8 HTML (Hypertext Markup Language )

HTML เป็นสิ่งที่ผู้พัฒนาโฮมเพจคุ้นเคยกันดี และเป็นภาษาที่ออกแบบมาเพื่อให้โปรแกรมบราวเซอร์สามารถเข้าใจ และทำงานได้ในแบบของไฮเปอร์เท็กซ์ ซึ่งผู้สร้างเว็บเพจจะใช้ภาษา HTML นี้ในการสร้างเว็บเพจและเก็บไว้ในเว็บเซิร์ฟเวอร์ เมื่อมีผู้ใช้งานติดต่อผ่านโปรแกรมบราวเซอร์ที่เครื่องโคลเอนต์ โดยระบุ URL ของเว็บเซิร์ฟเวอร์นั้นๆ ไฟล์ html ที่เก็บไว้ในเว็บเซิร์ฟเวอร์ก็จะถูกส่งไปยังโคลเอนต์โดยใช้โปรโตคอล HTTP และแสดงผลให้ผู้ใช้งานเห็นโดยผ่านโปรแกรมบราวเซอร์ html มีพื้นฐานมาจากภาษาที่เรียกว่า SGML (Standard Generalize Markup Language) และ HTML ได้รับการออกแบบมาให้ใช้งานกับเว็บเพจที่ไม่มีการเปลี่ยนแปลงหรือเคลื่อนไหวในพจนานุกรม ซึ่งในกรณีที่ต้องการพัฒนาให้เว็บเพจสามารถเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนแปลงหรือเคลื่อนไหว (Dynamic 8html) ได้นั้น จะต้องใช้ภาษาอื่นเข้ามาช่วยด้วย เช่น Visual Basic หรือภาษาที่เป็นสคริปต์ต่างๆ เช่น Vbscript เป็นต้น

#### - แท็ก (Tag)

ความเข้าใจในเรื่องของ Tag เป็นส่วนสำคัญของการเรียนรู้ HTML โดย Tag จะทำหน้าที่กำหนดขอบเขต หรือแบ่งแยกการสั่งงานต่างๆในไฟล์ HTML ให้ตัวแปลภาษาสามารถเข้าใจได้ หน้าที่ Tag คือใช้ในการสร้างเฮดดิ้ง, กำหนดย่อหน้า, สร้างลิสต์, กำหนดรูปแบบ (Formatting) และการเชื่อมโยงเว็บอื่นๆ โดยการเริ่มต้น Tag จะใช้เป็นตัวอักษรอยู่ภายในสัญลักษณ์ "<" และ ">" และปิดท้ายด้วยตัวอักษรที่อยู่ระหว่างสัญลักษณ์ "</" และ ">" ตัวอย่างเช่น ถ้าต้องการให้แสดงคำว่า "Welcome" เป็นอักษรตัวหนา (boldface) ก็จะต้องกำหนด Tag เป็น <B>Welcome</B> เป็นต้น ซึ่ง Tag บางประเภทก็ไม่จำเป็นต้องปิดท้าย เช่น <BR> ซึ่งหมายถึง Line Break ตัวอย่างเช่น ให้แสดงว่า Good Morning และต่อท้ายด้วย Line Break จะทำได้โดย Good Morning<BR> เป็นต้น ส่วน Tag ที่เป็นคำอธิบายโปรแกรม (Comments) จะเริ่มต้นด้วย"<!--" และปิดท้ายด้วย "-->"

#### โครงสร้างโดยทั่วไปของไฟล์ HTML

```
<HTML>
<HEAD>
<----- รายละเอียดของ title ----->
</TITLE>
<----- Header อื่นๆ ----->
</HEAD>
<BODY>
<----- ส่วนที่เป็นเนื้อหาของเอกสาร ----->
</BODY>
</HTML>
```

ในบางครั้งคำสั่งใน HTML ก็ไม่สามารถครอบคลุมการทำงานได้เพียงพอ จึงจำเป็นต้องใช้ภาษาอื่นๆเข้าทำงานร่วมด้วย ซึ่งเรียกว่า Script Language ตัวอย่างเช่น JavaScript , VBScript หรือ Jscript เป็นต้น Tag ที่ใช้กำหนดสคริปต์นี้ก็คือ <SCRIPT LANGUAGE> ตัวอย่างเช่น < SCRIPT LANGUAGE=JAVASCRIPT> หมายถึงกำหนดเนื้อหาในส่วนที่จะเป็นสคริปต์ของ JavaScript เป็นต้น

#### - Image Map

เป็นเว็บเพจต่างๆบางครั้งจะสังเกตเห็นได้ว่าไม่ได้มีแต่ข้อความเท่านั้นที่สามารถคลิกต่อไปได้ รูปภาพบางรูปก็สามารถคลิกเมาส์เพื่อแสดงข้อความอื่นๆหรือลิงค์ต่อไปยังเพจอื่นๆได้เช่นกัน นอกจากนี้รูปภาพรูปเดียวกันยังสามารถแบ่งออกเป็นส่วนย่อยๆที่สามารถคลิกเมาส์แล้วลิงค์ไปยังที่ต่างกันได้อีกด้วย ตัวอย่างเช่น ในเว็บเพจที่มีรูปแผนที่แสดงที่ตั้งของจังหวัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่างๆในประเทศ เมื่อคลิกที่จังหวัดใดก็จะแสดงรายละเอียดของจังหวัดนั้นๆเป็นต้น ซึ่งความสามารถลักษณะนี้เรียกว่า Image Map

#### - CGI (Common Gateway Interface)

ในการใช้งาน HTTP ที่ต้องการติดต่อกับเซิร์ฟเวอร์อื่นที่ไม่ใช่ HTTP ด้วยกัน เช่น การกรอกข้อมูลผ่านฟอร์ม และต้องการให้ไปค้นหาข้อมูลเซิร์ฟเวอร์อื่นที่เก็บฐานข้อมูลไว้ การทำงานในลักษณะนี้สามารถทำได้หลายวิธี แต่กลไกพื้นฐานที่ง่ายและสะดวกที่สุดคือการทำงานโดยใช้ CGI หรือ Common Gateway Interface ซึ่งหลักการทำงานของ CGI นี้ไคลเอนต์จะกำหนดไฟล์ CGI ที่ต้องการเรียกใช้งานโดยระบุในลักษณะของ URL เมื่อ HTTP หรือเซิร์ฟเวอร์ได้รับคำสั่งก็จะเรียกไฟล์ CGI นั้นๆขึ้นมาทำงาน จากนั้น HTTP หรือเว็บเซิร์ฟเวอร์ก็จะส่งผลลัพธ์การทำงานกลับไปให้ไคลเอนต์ตัวอย่างการเรียกใช้งาน CGI เช่น ต้องการเรียกใช้

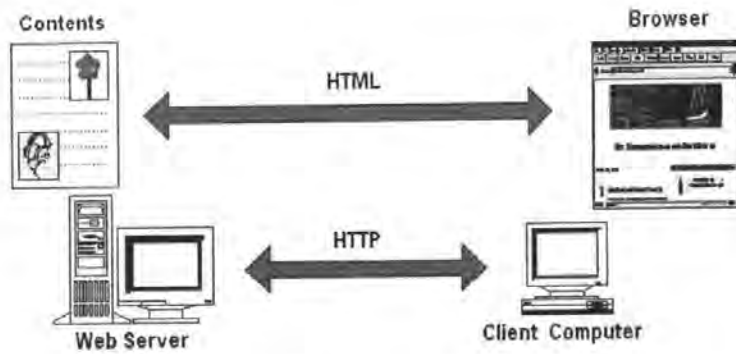
งาน CGI ชื่อ getcust.cgi อยู่ในไดเรกทอรี cgi-bin ที่เว็บไซต์ www.netcorp.com จะอ้างถึงได้ดังนี้ <http://www.netcorp.com/cgi-bin/getcust.cgi>

ในบางครั้งต้องมีที่ส่งค่าที่เป็นพารามิเตอร์เข้าไปใน CGI จะกำหนดได้ด้วยสัญลักษณ์ ? ถ้าหากจะเปรียบเทียบกับกรเขียนด้วยภาษาซีก็คือ argc/arve ของฟังก์ชัน main() ตัวอย่างเช่น <http://www.netcorp.com/cgi-bin/getcust.cgi?a=1>

ซึ่งภาษาที่จะใช้ในการเขียน CGI นั้นก็เป็นประเภทภาษาสคริปต่างๆ เช่น Perl หรืออาจจะเขียนเป็นโปรแกรมด้วยภาษาระดับสูงต่างๆก็ได้ เช่น ภาษา C เป็นต้น โดยให้โปรแกรมไปทำงานตามที่กำหนดไว้ และสร้างเป็นไฟล์ HTML ขึ้นมา ส่วนในการติดต่อกับ HTTP หรือเว็บเซิร์ฟเวอร์ของ CGI นั้น จะรับข้อมูลจากบราวเซอร์ที่ผ่านมาที่เว็บเซิร์ฟเวอร์ โดย CGI ต้องอ่านข้อมูลจากอินพุตมาตรฐาน เช่น ในภาษาซี จะอ้างถึง stdin และในทำนองเดียวกันการแสดงผลเอาต์พุตนั้น CGI จะเขียนข้อมูลลงเอาต์พุตมาตรฐานหรือ stdout ในภาษา C จากนั้นเอาต์พุตก็จะถูกส่งมาแสดงผลที่จอภาพของบราวเซอร์โดยอัตโนมัติ

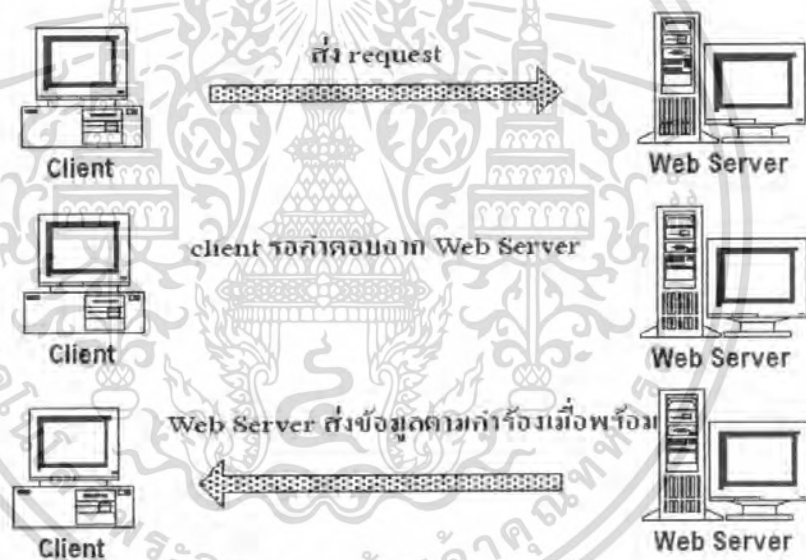
### 2.2.9 ความสัมพันธ์ระหว่าง HTTP กับ HTML

HTTP คือโปรโตคอลที่ใช้สื่อสารระหว่าง client computer กับ server computer ทำให้ทั้งสองเครื่องรู้ว่าจะจัดการส่งข้อมูลไปอย่างไร ส่วน HTML คือสื่อภาษาที่ทำให้เอกสารหรือ contents ที่อยู่บนเครื่อง server computer เมื่อถูกส่งมาที่ client computer แล้วจะนำไปแสดงได้อย่างไร เราเรียกซอฟต์แวร์ที่ใช้แสดงนี้ว่า Browser



ภาพที่ 2.12 HTTP กับ HTML

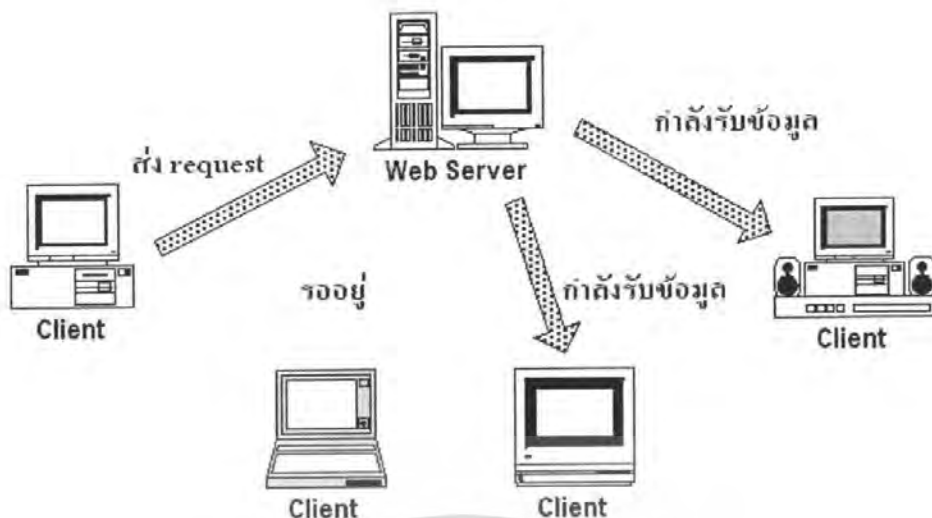
โปรโตคอล HTTP จะนี้่วงอยู่บน TCP/IP อีกชั้นหนึ่ง รูปแบบการทำงานจะไม่มี การจองสาย โดย client จะเรียกข้อมูลจาก server โดยการส่ง request ไปแล้วจะตัดการติดต่อทันที จากนั้นจะรอจนกระทั่ง server ส่งข้อมูลมาให้



ภาพที่ 2.13 รูปแบบการทำงาน HTTP

ประโยชน์ของการทำงานแบบไม่จองสายของ HTTP ทำให้ WWW server สามารถให้บริการ client ได้หลายๆ คนพร้อมๆ กัน การสื่อสารของ WWW จึงมีประสิทธิภาพมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.14 การทำงานแบบไม่จองสายของ HTTP

### 2.2.10 โปรแกรม Dynamic C

Dynamic C คือระบบการพัฒนาการเขียนซอฟต์แวร์ สร้างมาจากระบบคอมพิวเตอร์ของ IBM ซึ่งออกแบบให้เข้ากับคอมพิวเตอร์ทั่วไป ให้ใช้งานได้



ภาพที่ 2.15 โปรแกรม Dynamic C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Dynamic C นั้นมีการใช้ในเครือข่ายตั้งแต่ปี 1989 ถูกออกแบบสำหรับการเขียนโปรแกรมที่ฝังระบบ และมีความสามารถตรวจสอบในตัวของมันเองอย่างรวดเร็ว สำหรับไมโครโปรเซสเซอร์ Rabbit 2000 ที่ใช้งานกันทั่วไป สามารถติดต่อกันโดย สายแพ 10 สายที่ พอร์ต C โปรแกรมพื้นฐานของระบบ มีข้อมูลประมาณ 1,000 ไบท์ ที่ใช้ในการจัดเตรียม Debugging และการติดต่อข้อมูลต่างๆ Dynamic C ต้องการ BIOS เพื่อใช้ในการตรวจสอบโปรแกรมเพื่อที่จะใช้งานได้สะดวกถ้าผู้ใช้หยุดการ Run โปรแกรมและใช้โปรแกรมใหม่ BIOS ก็ จะเช็คการทำงานใหม่ตลอด Dynamic C ออกแบบให้เข้ากับภาษา Assembly หรือใช้ได้กับโปรแกรมภาษาซี การ Interrupt อาจะเกิดขึ้นได้กับการเขียนในภาษา Dynamic C หรือภาษา Assembly ซึ่งซอฟต์แวร์ Dynamic C เป็นซอฟต์แวร์ที่มีประสิทธิภาพในการเขียนโปรแกรมเพื่อควบคุมการทำงานของ ไมโครโปรเซสเซอร์ Rabbit2000 โดยซอฟต์แวร์มีโครงสร้างการใช้งาน คล้ายคลึงกับ ภาษา C ลักษณะของตัวซอฟต์แวร์โปรแกรม Dynamic C จะสนับสนุน TCP/IP โดยจะมี Libraies หลักที่สนับสนุนคือ DCRTCP.LIB ,DNS.LIB, IP.LIB, TCP.LIB, UDP.LIB, NET.LIB ส่วนในการติดต่อชั้นเครือข่ายของโปรโตคอล TCP/IP จะมี Libraies ARP.LIB กับ ICMP.LIB

ผู้ใช้โปรแกรม Dynamic C มีตัวเลือกในการที่จะพัฒนาซอฟต์แวร์ภาษาเขียน ใน flash memory ขนาด 256 K ไบท์ หรือใน Static Ram ขนาด 128 K ไบท์ ผลการทำงานในหน่วยความจำ คือการบันทึกข้อมูลสามารถบันทึกได้ถึง 100,000 ครั้งของการเขียน

ข้อเสียของการใช้ Flash Memory เมื่อมีการดีบั๊กโปรแกรมเพื่อขจัดจั้งหะการทำงาน จะทำให้ Interrupt เกิดข้อผิดพลาด การทำงานของโปรแกรมก็จะหยุดตามไปด้วย

#### Dynamic C Libraries

การรัน Dynamic C File > Open และ เลือก Lib จะมีชื่อไฟล์ที่เป็นนามสกุล lib ที่จะใช้งาน ซึ่งบนจะบอร์ดมีชิพ RS-232 ใช้สำหรับ Link กับที่ Libraries ของ Rabbit RS232.LIB ซึ่งเป็นไฟล์ที่ใช้สำหรับจัดเตรียมฟังก์ชันเพื่อใช้ในการส่งและรับข้อมูล

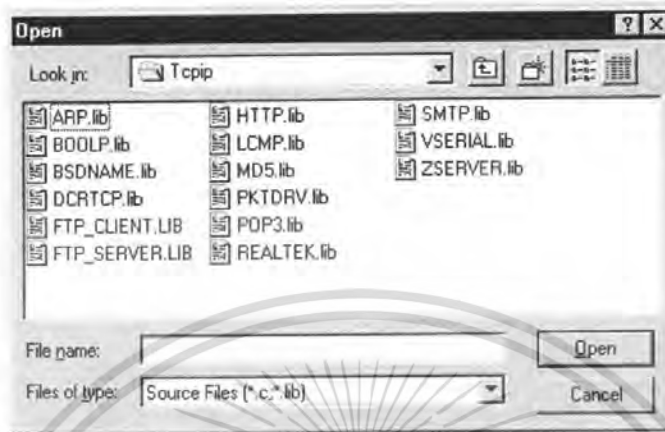


ภาพที่ 2.16 ไฟล์ที่อยู่ใน Lib

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP / IP Drivers

TCP / IP Drivers ที่อยู่ใน TCPIP directory



ภาพที่ 2.17 ไฟล์ที่อยู่ใน TCP/IP

### 2.2.11 Rabbit3720 Microprocessor

#### RCM3720

คือ คอร์ - ไมโคร ในตระกูลของ RabbitCore ออกแบบมาสำหรับงานการเชื่อมต่อกับ Ethernet / Internet RCM3720 มีหน่วยความจำโปรแกรมแบบ Flash ขนาด 512K/SRAM ขนาด 512K หรือ Flash ขนาด 256K /SRAM ขนาด 256K, 4 พอร์ตอนุกรมและมีบอร์ดขนาดเล็ก (2.95"x1.20"/ 75 x 30 mm) RCM3720 เป็นรุ่นที่มี Ethernet/ Internet ทำงานด้วย Rabbit3000 ที่มีราคาถูก แลมีโปรแกรมตัวอย่างและซอฟต์แวร์ ชุดพัฒนาที่สามารถนำมาพัฒนาการเชื่อมต่อ Secure Web Browser สำหรับระบบควบคุมแบบฝังตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



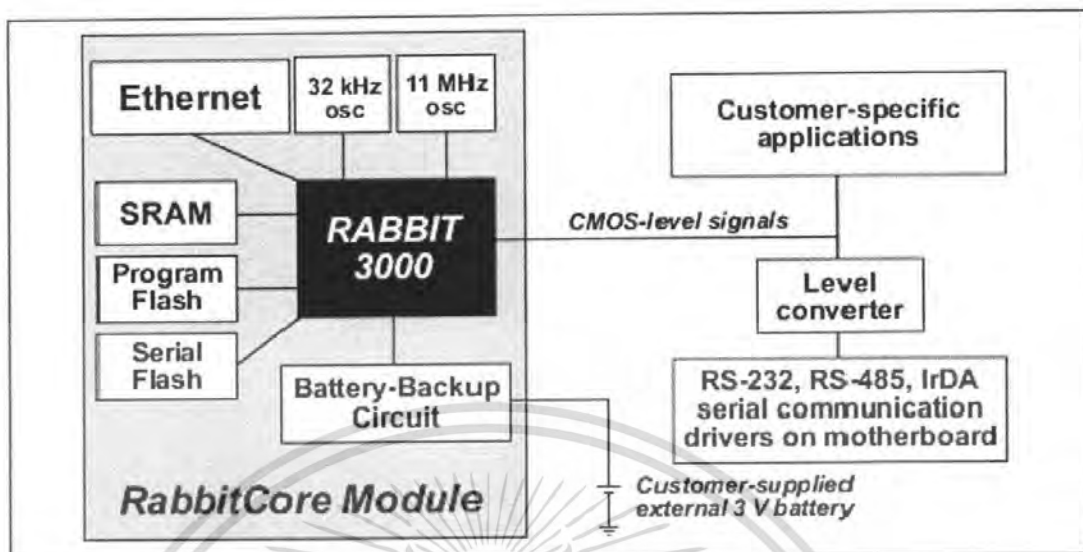
ภาพที่ 2.18 บอร์ด Rabbit

#### คุณสมบัติเด่น

1. อ่านและเขียนตัวแปรจากระยะไกลได้โดยตัดความยุ่งยากการโปรแกรมแบบ CGI
2. สร้างการควบคุมได้ง่ายเช่น เมนูแบบ Pull-Down หรือปุ่มคอนโทรล มั่นใจได้ว่าค่าอินพุตที่ ป้อนไปเป็นค่าที่ถูกต้องและมีการตรวจสอบสิทธิ์ผู้ใช้งาน
3. การแสดงผลความผิดพลาดมีความชัดเจนเพื่อง่ายต่อการแก้ไข ย่อเวลาในการโปรแกรมและตรวจสอบแบบ CGI 10X
4. ทำงานด้วย Dynamic C HTTP Server เพื่ออัปเดตค่าได้อย่างถูกต้อง
5. หน่วยเก็บข้อมูล : ฐานข้อมูล (Data Base) และ Web Page สามารถใช้แบดเตอร์ี่สำรองเพื่อป้องกันระบบไฟล์ข้อมูลในขณะไฟเลี้ยงหาย
6. สามารถทำงานได้เร็วกับกระบวนการเข้ารหัสที่ซับซ้อนสูงถึง 120 Kbit/ วินาที
7. สามารถใช้ HTTP ด้วย SSL Version3 และ Transport Layer Security (TLS) Version 1 ความปลอดภัยในการประยุกต์ใช้งานเว็บด้วยความละเอียดของคำสั่ง <math>< 10</math> บรรทัด

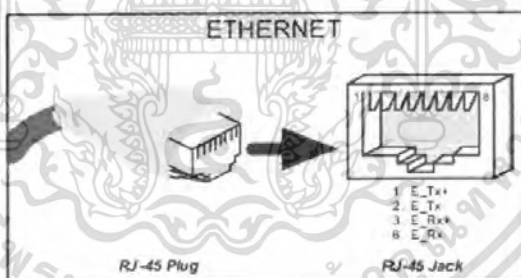
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hardware อ้างอิง



ภาพที่ 2.19 แสดงระบบย่อยของ RCM 3700

Ethernet Port



ภาพที่ 2.20 แสดง pinout สำหรับ RJ-45 Ethernet port (J3).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Rabbit 3000 TFBGA



ภาพที่ 2.21 Rabbit 3000 TFBGA

ไมโครโปรเซสเซอร์ราคาถูก ออกแบบมาเป็นพิเศษสำหรับระบบการฝังตัว

1. ขนาดบอร์ด 10x10x1.2 mm
2. อินพุต/เอาต์พุต 56 ขา แบบดิจิตอล
3. แรงดันไฟฟ้า 1.5-3.6 V
4. พอร์ตสื่อสารอนุกรม 6 พอร์ต
5. สัญญาณนาฬิกา 54 MHz
6. เป็น RabbitCore โมดูล ปฏิบัติการที่สัญญาณนาฬิกา 22.1 MHz และ 10 Base-T Ethernet และมีหน่วยความจำในตัว
7. ได้แก่แบบ Flash ขนาด 512 K, SRAM ขนาด 256 K และ Serial Flash อีก 1 MB, เส้น GPIO 33 เส้น
8. มีโมดูลไร้สายรุ่น ZigBee ของ MaxStream 3 โมดูล และสายสัญญาณสำหรับการเชื่อมต่อได้สะดวก
9. มีโปรแกรมตัวอย่าง และไลบรารี เพื่อตอบสนองการทำงาน อินพุต/เอาต์พุตดิจิตอล และการควบคุมได้ง่าย
10. มีGateway ทำหน้าที่เชื่อมต่อ Ethernet ไปยัง ZigBee และมีการติดต่อสื่อสารของโครงข่ายแบบ Mesh
11. Rabbitweb ที่ทำให้ง่ายต่อการโปรแกรมซอฟต์แวร์ HTML ด้วยกลไกรีเฟรช
12. มีการเชื่อมต่อ web ที่ง่ายสำหรับการติดตั้ง(setup) โครงข่าย ZigBee / 802.15.4 มีโครงสร้างโมเด็มรูปแบบ RF และฟังก์ชันต่างๆที่ใช้ในการอัปเดต Firmware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2.12 ZigBee Modem



ภาพที่ 2.22 ZigBee Modem

ชุดอุปกรณ์ของ ZigBee™/802.15.4 ประกอบด้วย โมเด็ม MaxStream's XBee™ RF และ โมดูล RabbitCore ในส่วนของโมเด็ม XBee™ มีการเชื่อมต่อเข้ากับโมดูลแบบ RF ซึ่งมี power supply และ พอร์ตอนุกรม RS-232 ที่มี flow control ด้วย โมดูลแบบ RF ทั้ง 3 จะประกอบไปด้วย โมเด็ม XBee™ RF จำนวน 3 ชุด โดยคุณสามารถที่จะกำหนดให้ชุดหนึ่งเป็น coordinator ที่คอยควบคุมระบบ Rabbit-based และ อีก 2 ชุด เป็น children ที่ระยะไม่เกิน 100 ฟุต (30 เมตร)

ชุดอุปกรณ์ของ ZigBee™/802.15.4 ทำหน้าที่เป็นแบบสำหรับ ระบบ Rabbit-based ซึ่งผสมผสานโมเด็ม XBee™ แบบไร้สายกับระบบ Rabbit-based เข้าไว้ด้วยกัน ตัวอย่างโปรแกรมประกอบด้วยชุดโปรแกรมประยุกต์ ที่ทำหน้าที่เป็นแบบสำหรับโปรแกรมประยุกต์ไร้สาย ซึ่งต้องการกำลังและอัตราโอนถ่ายข้อมูลที่ต่ำ โปรแกรมประยุกต์มีส่วนที่เป็น การตรวจจับ ระยะไกลอย่างง่าย การอ่านเซนเซอร์ระยะไกล การควบคุมอินพุต/เอาต์พุตแบบไร้สาย และการโอนถ่ายข้อมูล ส่วนที่เป็น user-configurable จะกำหนดให้คุณสามารถที่จะติดตั้งเครือข่าย ระบุ และสำรวจเครือข่ายสำหรับโมเด็ม XBee™ RF ชุดอื่นๆ อีกทั้ง คุณสามารถควบคุม LED และ สวิตช์ ผ่านโมดูล RF ด้วยโมเด็ม XBee™ RF ที่ทำหน้าที่เป็น coordinator

## 2.2.13 บอร์ด Controller

ET-BASE51 V2.0 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล MCS51 ขนาด 40 Pin ซึ่งเลือกใช้ไมโครคอนโทรลเลอร์เบอร์ AT89451ED2 ของ ATMEL เป็น MCU ประจำบอร์ด โดย MCU รุ่นนี้จะบรรจุอยู่ในตัวถังแบบ DIP40 โดย MCU ตัวนี้จะมีจุดเด่น คือ เรื่องของความเร็วในการประมวลผล ซึ่งสามารถทำงานได้ด้วยความถี่สูงสุด 60 MHz ที่ 12 Clock/1 Machine Cycle นอกจากนี้แล้วยังมีความยืดหยุ่นพร้อมด้วยอุปกรณ์พื้นฐานต่างๆที่จำเป็นต่อการใช้งาน ไม่ว่าจะเป็นหน่วยความจำสำหรับเก็บข้อมูลแบบ EEPROM ขนาด 2 กิโลบิต เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KByte หรือหน่วยความจำใช้งานแบบ RAM ซึ่งมีมากถึง 1792 Byte ส่วนในด้านของอุปกรณ์ Peripheral นั้นก็นับว่าครบถ้วนเหมาะสมแก่การนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมและประมวลผลต่างๆได้เป็นอย่างดี โดยจะมีทั้ง SPI, UART, Watchdog, Timer/Counter, PWM ฯลฯ โดยการออกแบบโครงสร้างของบอร์ดนั้นจะเน้นเรื่องขนาดของบอร์ดให้มีขนาดเล็กเพื่อให้ง่ายต่อการนำไปประยุกต์ใช้งาน และสะดวกต่อการพัฒนาโปรแกรม

### คุณสมบัติของบอร์ด

เลือกใช้ MCU ตระกูล MCS51 เบอร์ AT89C51ED2 ของ ATMEEL เป็น MCU ประจำบอร์ด โดยเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกาแบบ Oscillator Module ค่า 29.4912 MHz ซึ่งสามารถกำหนดการทำงานของ MCU ให้ทำงานในโหมดความเร็ว 2 เท่า (x2 Mode) ได้ ทำให้ MCU สามารถประมวลผลด้วยความเร็วสูงสุดที่ 58.9824 MHz โดยคุณสมบัติเด่นๆของ MCU ได้แก่

1. มีหน่วยความจำ Flash สำหรับเขียนโปรแกรมขนาด 64 KByte
2. มี EEPROM ขนาด 2 KByte สำหรับเก็บข้อมูลและสามารถเขียนซ้ำได้กว่า 1 ล้านครั้ง
3. มีพอร์ต I/O ขนาด 8 บิต จำนวน 4 พอร์ต (P0,P1,P2 และ P3)
4. มี RAM ใช้งาน 1792 Byte เพียงพอต่อการใช้งาน
5. มีวงจรรีโมตสื่อสารอนุกรม UART จำนวน 1 พอร์ต และมีวงจรรีโมตสื่อสาร SPI จำนวน 1 พอร์ต
6. มีวงจรรีโมตสื่อสาร Timer/Counter ขนาด 16 บิต จำนวน 3 ชุด
7. มีวงจรรีโมตสื่อสาร Watchdog, Power-ON Reset, Capture/Compare, PWM

มีพอร์ตสื่อสารอนุกรมแบบ RS232 จำนวน 1 ช่อง สำหรับใช้ในการ Download โปรแกรมให้กับบอร์ด และประยุกต์ใช้งานทั่วไป

- มีขั้วต่อสัญญาณ I/O แบบ TTL แบบ Header ขนาด 2x5 จำนวน 4 ชุด (P0, P1, P2 และ P3)
- มี LED แสดงสถานะแหล่งจ่าย Power และ Self-Test สำหรับใช้ทดสอบการทำงานของบอร์ด
- ใช้แหล่งจ่ายไฟขนาด +5 VDC
- ขนาด PCB Size เล็กเพียง 8x6 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โครงสร้างบอร์ด ET-BASE-PIC40/458



ภาพที่ 2.23 บอร์ด ET-BASE-PIC40/458

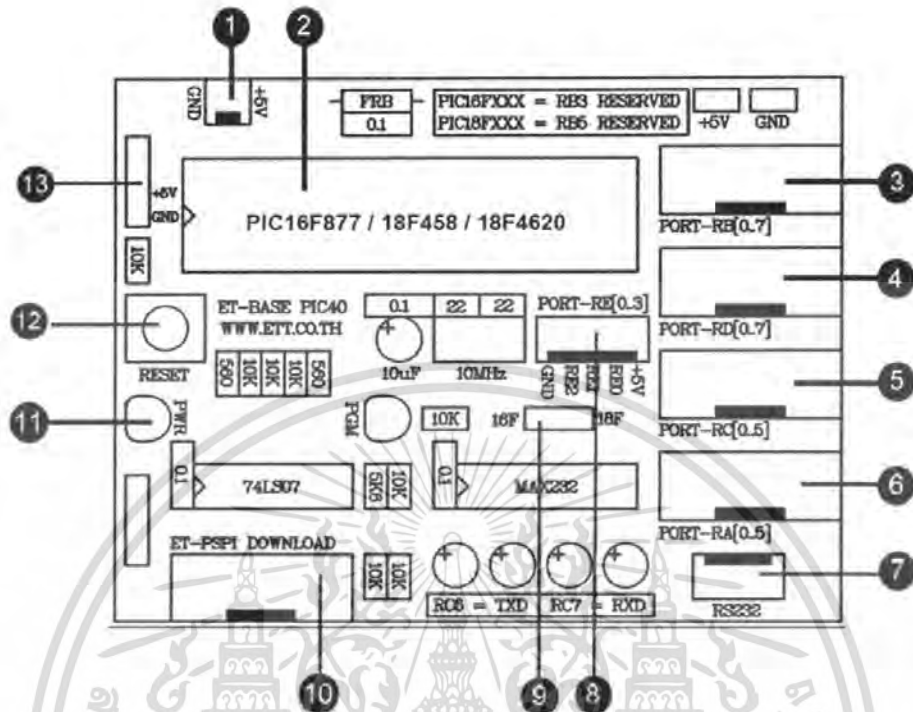
ET-BASE PIC40 เป็นบอร์ดไมโครคอนโทรลเลอร์ในตระกูล PIC ขนาด 40 PIN ของบริษัท Microchip ซึ่งในเวอร์ชันนี้ได้นำเอา PIC MCU มาจัดวงจรใช้งานให้มีขนาดกะทัดรัดโดยเน้นการใช้งานทรัพยากรของ PIC MCU เป็นหลัก นอกจากนี้ยังออกแบบให้สนับสนุนการนำไปใช้งานร่วมกับบอร์ดทดลอง "ET-BASIC IO" อีกด้วย

### คุณสมบัติของบอร์ด

- รองรับการใช้งานไมโครคอนโทรลเลอร์ขนาด 40 PIN คือ PIC16F877 , PIC18F458 และ PIC18F4620
- สัญญาณนาฬิกาคริสตอลอสซิลเลเตอร์ขนาด 10 MHz
- I/O Port ขนาด 10 PIN (จัดเรียงตามมาตรฐานของ อีทีที) จำนวน 4 พอร์ต
- I/O Port ขนาด 5 PIN จำนวน 1 พอร์ต
- ชุดวงจรไดรฟ์เวอ์ RS232 จำนวน 1 พอร์ต
- ชุดวงจรดาวน์โหลดแบบแรงดันต่ำ (Low Voltage Programming)
- ขั้วต่อแรงดันไฟ VCC และ GND

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โครงสร้างบอร์ด ET-BASE PIC40

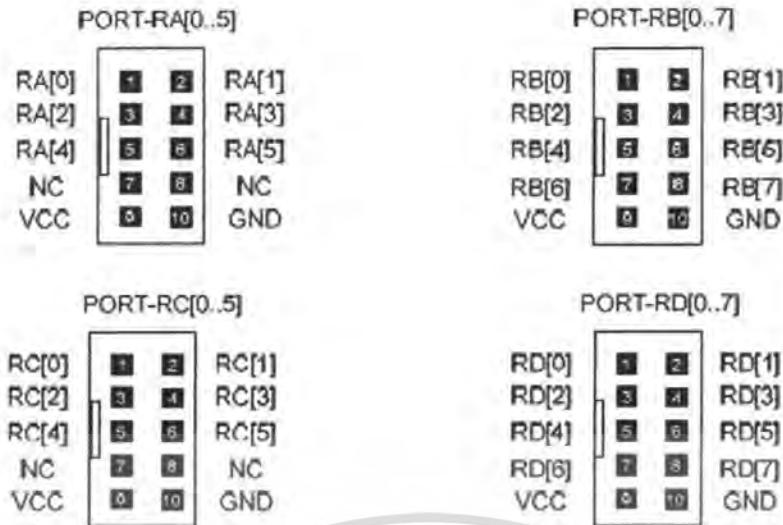


ภาพที่ 2.24 โครงสร้างของบอร์ด ET-BASE PIC40

อธิบายรายละเอียดตามหมายเลขต่าง ๆ ดังนี้

- หมายเลข 1 ขั้วต่อแหล่งจ่ายไฟ
- หมายเลข 2 ไมโครคอนโทรลเลอร์ PIC 40 PIN รองรับ PIC16F877, PIC18F458 และ PIC18F4620
- หมายเลข 3 ,4 ,5 และ 6 เป็นพอร์ตของขาสัญญาณ I/O ของ PIC MCU คือ PORT-RA , PORT-RB, PORT-RC และ PORT-RD ตามลำดับ โดยจะมีการจัดเรียงขาสัญญาณดังรูปต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.25 การจัดเรียงขาสัญญาณของบอร์ด ET-BASE PIC40

หมายเหตุ ขาสัญญาณ RB3 (กรณี PIC16F877) หรือ RB5 (กรณี PIC18F458 และ PIC18F4620) จะถูกสงวนไว้สำหรับฟังก์ชันการโปรแกรมไม่สามารถใช้งานได้ ส่วน RB6 และ RB7 ให้ถอดสายสัญญาณคาว์นโหลดออกก่อนจึงจะใช้งานได้

- หมายเลข 7 พอร์ต RS232 จัดเรียงสัญญาณดังนี้

RS-232 Port



TXD=RC6 RXD=RC7

ภาพที่ 2.26 การจัดเรียงสัญญาณของพอร์ต RS232

- หมายเลข 8 ขาสัญญาณ PORTE



ภาพที่ 2.27 ขาสัญญาณ PORTE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเลข 9 จัมป์เปอร์เลือกการใช้งาน PIC MCU โดยสามารถเลือกได้ 2 แบบดังนี้

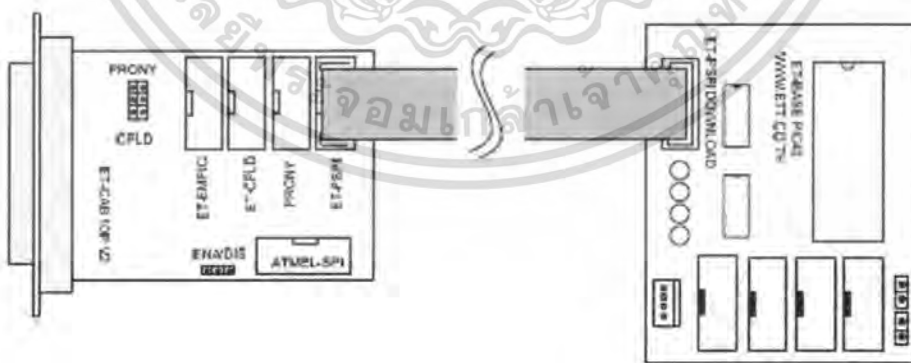
- 16F คือ ไมโครคอนโทรลเลอร์ 40 PIN เบอร์ PIC16F877
- 18F คือ ไมโครคอนโทรลเลอร์ 40 PIN เบอร์ PIC18F458 และ PIC18F4620

16F  18F

ภาพที่ 2.28 จัมป์เปอร์เลือกการใช้งาน PIC MCU

- หมายเลข 10 พอร์ตสัญญาณสำหรับดาวน์โหลดโปรแกรม
- หมายเลข 11 LED แสดงสถานะของแหล่งจ่ายไฟภายในบอร์ด
- หมายเลข 12 สวิตช์ RESET โปรแกรม
- หมายเลข 13 ขั้วต่อแหล่งจ่ายไฟ สำหรับใช้ร่วมกับบอร์ด ET-BASIC I/O
- การโปรแกรมซอร์สโค้ด (Hex File)

จะใช้ซอฟต์แวร์ WinPic800 โดยจะต้องทำการเชื่อมต่อสายสัญญาณดาวน์โหลด ระหว่างบอร์ด กับคอมพิวเตอร์ โดยผ่าน ET-CAB 10P ดังนี้



ภาพที่ 2.29 ET-CAB 10P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

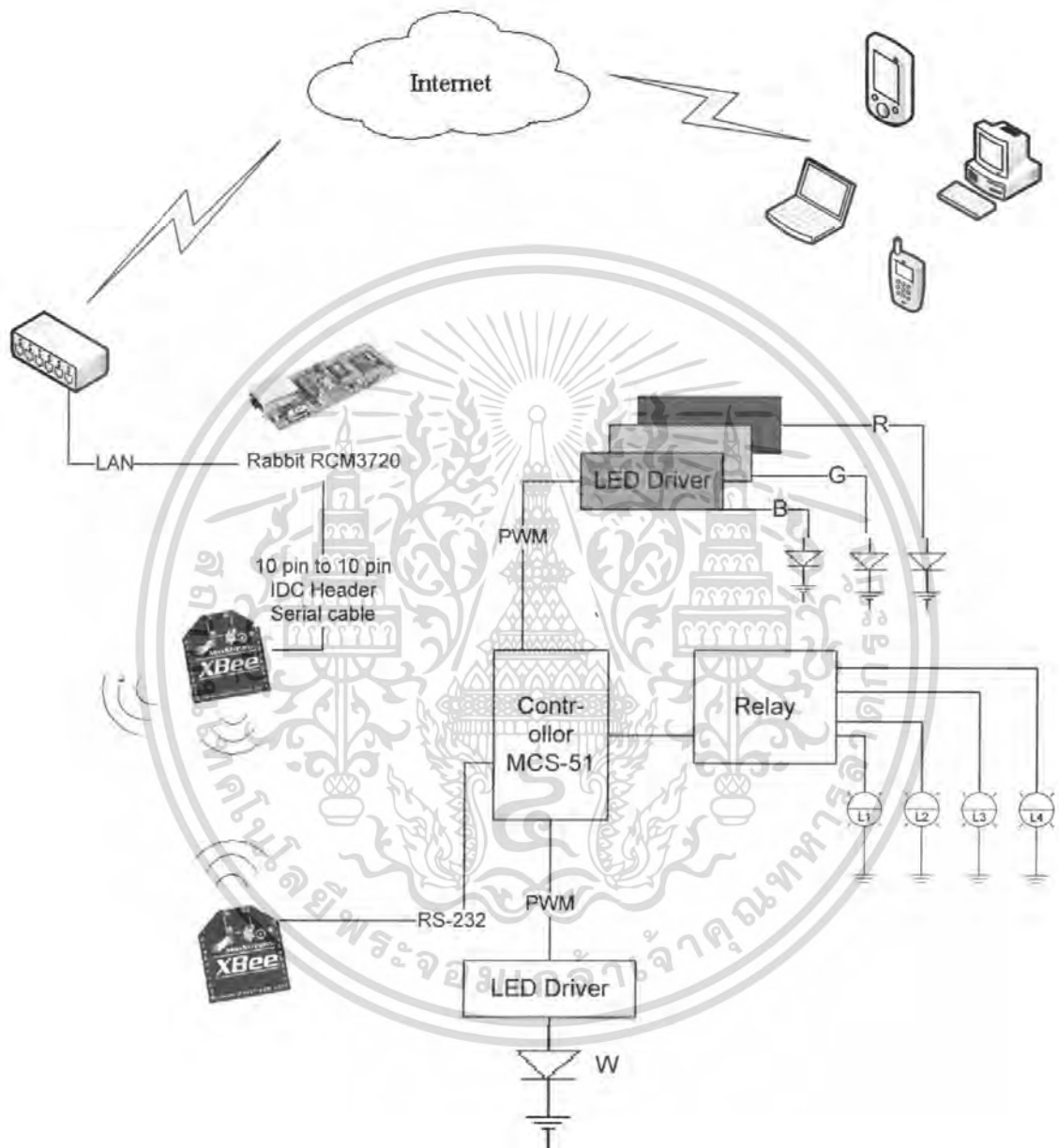
### ข้อแนะนำเบื้องต้น

- ตรวจสอบการเชื่อมต่อของสายสัญญาณต่างๆ
- ตรวจสอบการจ่ายพลังงานให้กับบอร์ด
- ตรวจสอบการเลือกจัมป์เปอร์ 16F/18F ว่าถูกต้องตรงตามเบอร์ที่ใช้หรือไม่
- บอร์ดรับแรงดันไฟได้ 5 VDC ระวังห้ามจ่ายไฟเกิน
- ขาสัญญาณ RB6,RB7 หากไม่สามารถใช้งานได้ให้ถอดสายสัญญาณตาวาน์โหลดออก  
จึงจะสามารถ  
ใช้งานได้



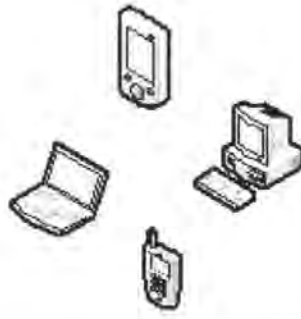
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3 การออกแบบโปรแกรม



ภาพที่ 3.1 แสดงภาพรวมของโครงการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



อุปกรณ์ที่สามารถต่ออินเทอร์เน็ตได้



ผ่านทางระบบ Wireless



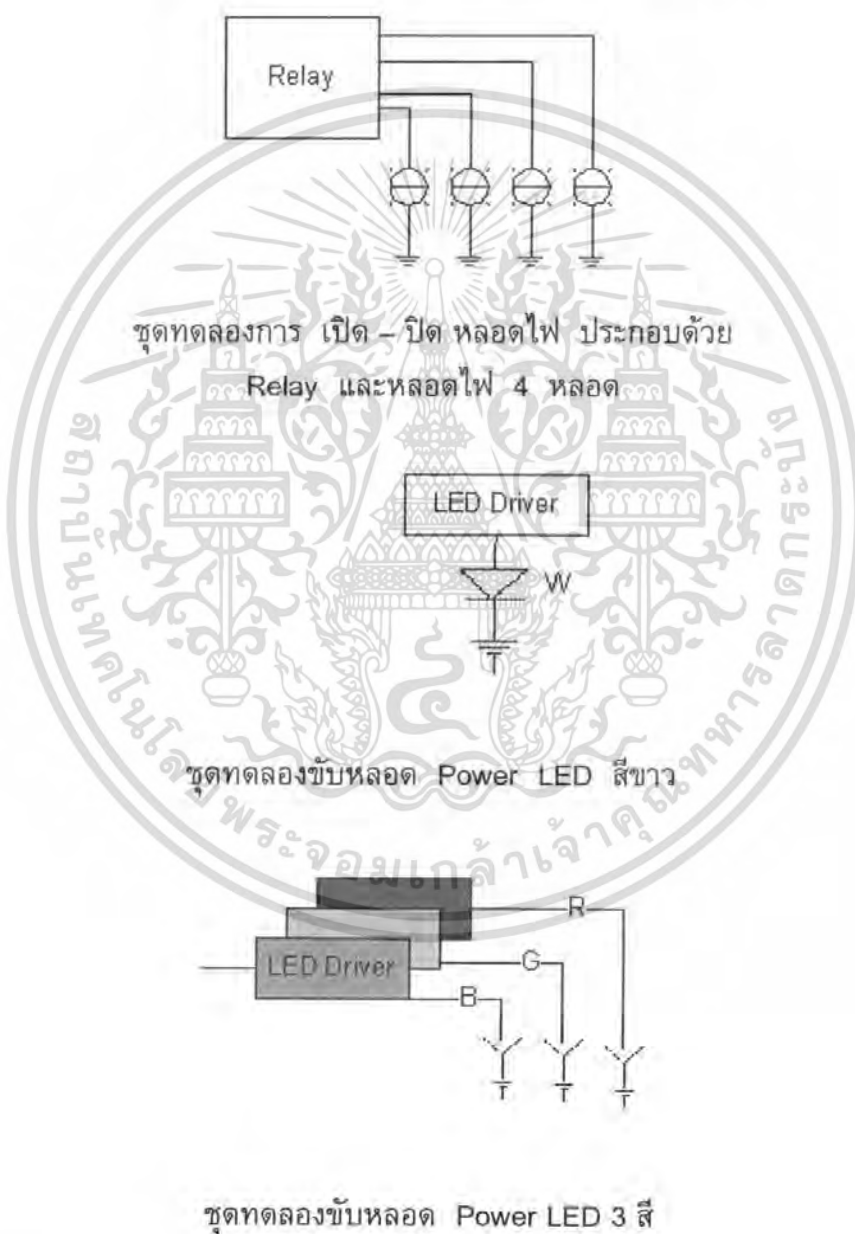
Rabbit RCM3720

Rabbit RCM 3720 เป็นอุปกรณ์ที่สามารถเป็นเว็บเซิร์ฟเวอร์ และทำงานเป็นไมโครคอนโทรลเลอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Contr-  
oller  
MCS-51

ไมโครคอนโทรลเลอร์ เป็นอุปกรณ์ที่รับคำสั่งมา  
ประมวลผลและสั่งอุปกรณ์ต่างๆ ให้ทำงานตามคำสั่ง



ภาพที่ 3.2 แสดงรายละเอียดของอุปกรณ์แต่ละส่วนที่ใช้ในการทำโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1 การออกแบบโดยรวมของโครงการ

เริ่มจากการเขียน Web page สั่งงานด้วยโปรแกรม Dynamic C เวอร์ชัน 9.25 ลงในบอร์ด Rabbit RCM 3720 ซึ่งเชื่อมต่อกับ Local Internet ผ่านระบบ LAN การทำงานจะส่งผ่านคำสั่งไปยัง Coordinator ZigBee โดยผ่าน 10 Pin to 10 Pin IDC Header Serial Cables ที่ตั้งค่าต่างๆโดยใช้โปรแกรม Dynamic C คำสั่งจะถูกส่งจาก Coordinator ZigBee ไปยัง Router ZigBee โดยผ่านระบบ Wireless LAN เมื่อ Router ZigBee ได้รับคำสั่งจะส่งคำสั่งผ่าน 10 Pin to 10 Pin IDC Header Serial Cables ซึ่งต่ออยู่กับ Microcontroller ควบคุมอุปกรณ์ไฟฟ้าอีกทีหนึ่ง แล้วบอร์ดจะทำการตรวจสอบว่าอุปกรณ์ไฟฟ้าทำงานตามคำสั่งแล้วหรือไม่ เมื่อทำแล้วบอร์ดจะส่งคำสั่งไปยัง Router ZigBee และ Router ZigBee จะส่งคำสั่งของบอร์ดไปยัง Coordinator ZigBee โดยผ่านระบบ Wireless LAN จากนั้น Coordinator ZigBee จะส่งคำสั่งไปยัง Rabbit เพื่อเปลี่ยนสถานะการทำงานของอุปกรณ์ไฟฟ้าใน Web page

### 3.2 การออกแบบในส่วนของโปรแกรม Dynamic C เวอร์ชัน 9.25

ในส่วนของการออกแบบโปรแกรม Dynamic C เวอร์ชัน 9.25 เป็นส่วนที่สำคัญในการเขียน Web page และเชื่อมต่อ Rabbit กับ Coordinator ZigBee



ภาพที่ 3.3 หน้าโปรแกรม Dynamic C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

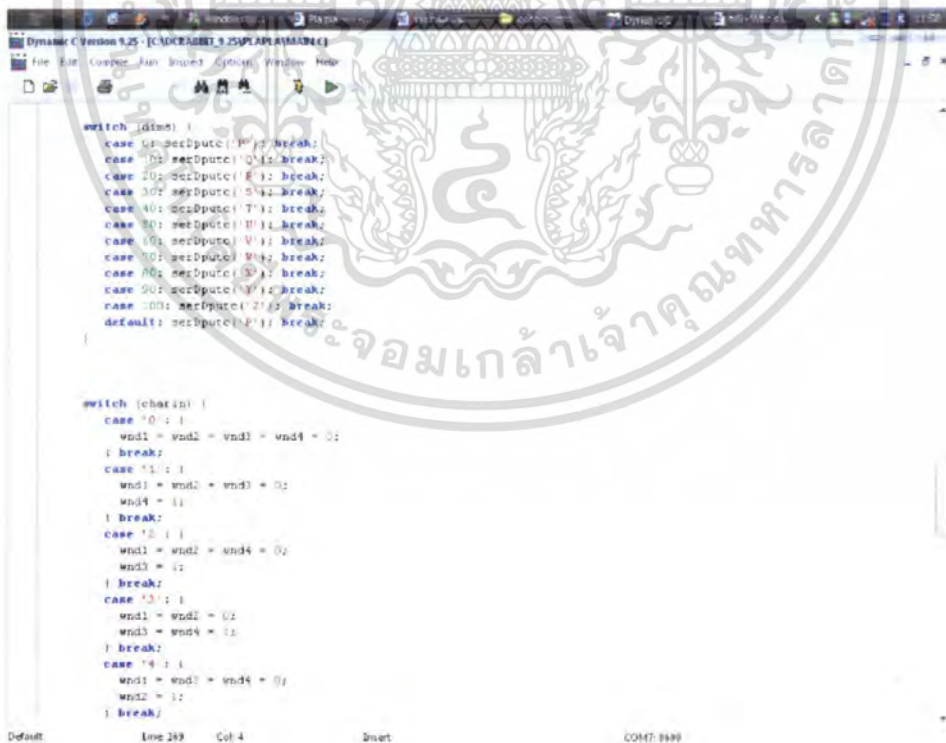


### 3.2.2 การเรียกใช้ชื่อเอกสาร Web page ที่สร้างไว้เก็บลงในบอร์ด Rabbit จะใช้คำสั่งต่อไปนี้

```

SSPEC_RESOURCETABLE_START
SSPEC_RESOURCE_XMEMFILE("/", index_html),
SSPEC_RESOURCE_XMEMFILE("/index.shtml", index_html),
SSPEC_RESOURCE_XMEMFILE("/showsrc.shtml", showsrc_shtml),
SSPEC_RESOURCE_XMEMFILE("/rabbit1.gif", rabbit1_gif),
SSPEC_RESOURCE_XMEMFILE("/ledon.gif", ledon_gif),
SSPEC_RESOURCE_XMEMFILE("/ledoff.gif", ledoff_gif),
SSPEC_RESOURCE_XMEMFILE("/button.gif", button_gif),
SSPEC_RESOURCE_XMEMFILE("browseled.c", browseled_c),
SSPEC_RESOURCE_ROOTVAR("led1", led1, PTR16, "%s"),
SSPEC_RESOURCE_ROOTVAR("led2", led2, PTR16, "%s"),
SSPEC_RESOURCE_FUNCTION("/led1tog.cgi", led1toggle),
SSPEC_RESOURCE_FUNCTION("/led2tog.cgi", led2toggle),
SSPEC_RESOURCETABLE_END

```



```

switch (dimen) {
case 0: serputc('A'); break;
case 01: serputc('B'); break;
case 20: serputc('E'); break;
case 30: serputc('G'); break;
case 40: serputc('T'); break;
case 50: serputc('I'); break;
case 60: serputc('V'); break;
case 70: serputc('W'); break;
case 80: serputc('X'); break;
case 90: serputc('Z'); break;
case 100: serputc(' '); break;
default: serputc(' '); break;
}

switch (charin) {
case '0': {
wnd1 = wnd2 = wnd3 = wnd4 = 0;
} break;
case '1': {
wnd1 = wnd2 + wnd3 + 0;
wnd4 = 1;
} break;
case '2': {
wnd1 = wnd2 = wnd4 = 0;
wnd3 = 1;
} break;
case '3': {
wnd1 = wnd2 = 0;
wnd3 = wnd4 = 1;
} break;
case '4': {
wnd1 = wnd2 = wnd4 = 0;
wnd3 = 1;
} break;
}

```

**ภาพที่ 3.5** แสดงหน้าต่างโปรแกรม Dynamic C ที่มีการเขียนคำสั่งควบคุมการทำงานของเอกสารนี้เป็นเอกสารที่ส่งอุปกรณ์ไฟฟ้าการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

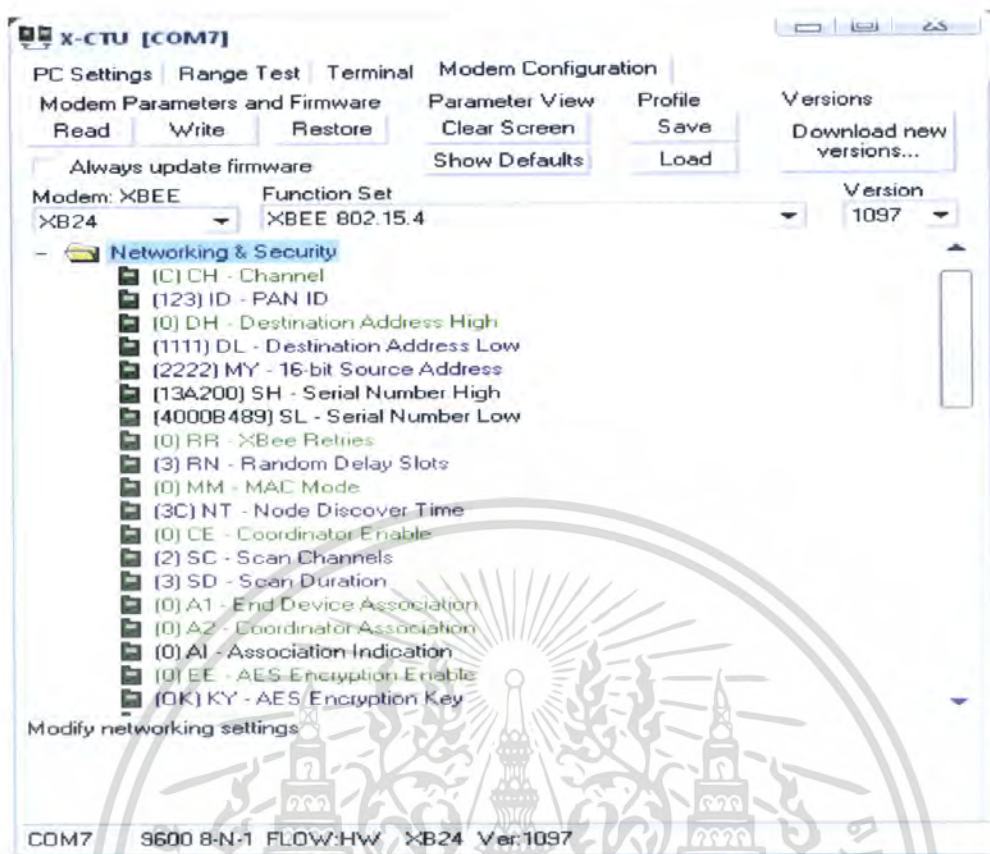
### 3.3 การเชื่อมต่อ ZigBee ทั้ง 2 ตัวโดยใช้โปรแกรม X-CTU

ในส่วนการออกแบบโปรแกรม X-CTU เป็นส่วนที่สำคัญในการทำให้ ZigBee ทั้ง 2 ตัว สามารถส่งผ่านข้อมูลถึงกันได้โดยใช้ระบบ Wireless LAN



ภาพที่ 3.6 หน้าโปรแกรม X-CTU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.7 แสดงหน้าต่างโปรแกรม X-CTU ในการตั้งค่าพารามิเตอร์ให้กับ Zig Bee

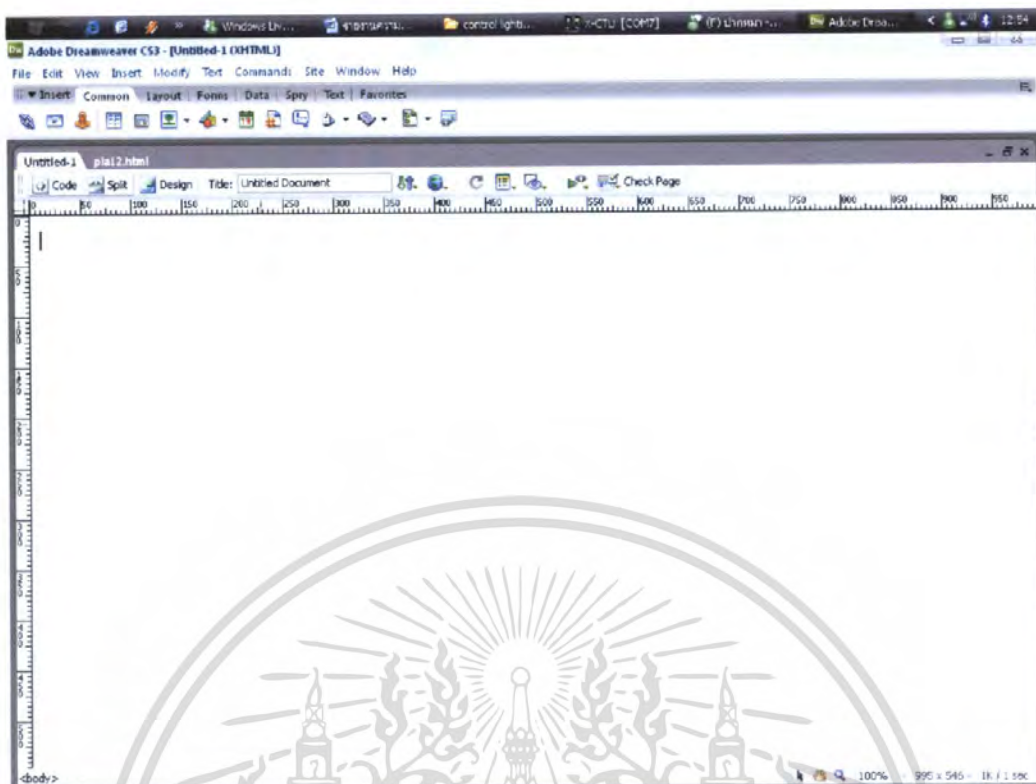
### 3.3.1 การตั้งค่าพารามิเตอร์ของ Zig Bee

ในโครงงานนี้เราจะใช้ Zig Bee ทั้งหมดสามตัวด้วยกัน ซึ่งเราจะทำการตั้งค่าให้การส่งข้อมูลของ ZigBee เป็นแบบตัวต่อตัว คือจะกำหนดให้ coordinator ZigBee ส่งข้อมูลให้กับ router ZigBee ทั้งสองตัวซึ่งถูกกำหนดค่า mind ให้เป็นชื่อเดียวกัน เพื่อให้ coordinator ZigBee มองเห็นเป็นตัวเดียวกัน ซึ่งค่าพารามิเตอร์ต่างๆแสดงดังภาพที่ 3.7

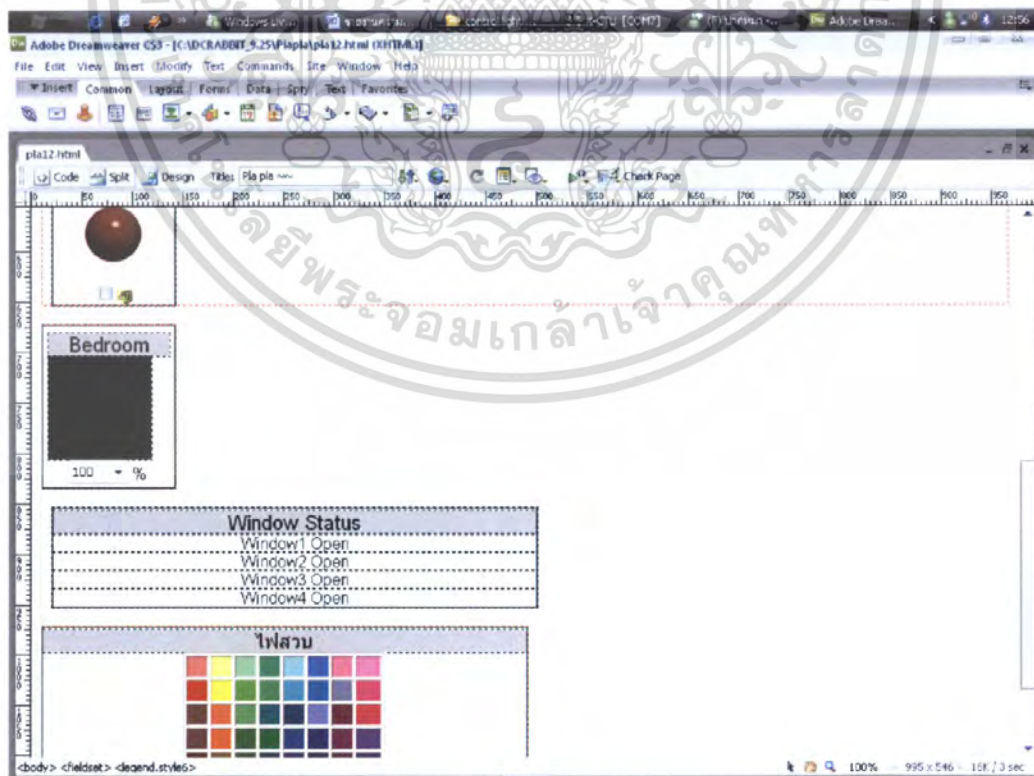
### 3.4 การสร้างหน้าเวปเพจ

ในการสร้างหน้าเวปเพจ จะใช้โปรแกรม Adobe Dreamwaver CS3 ซึ่งเป็นโปรแกรมที่ง่ายสำหรับผู้ที่เริ่มต้นสร้างเวปเพจ เนื่องจากมีเครื่องมือและฟังก์ชันช่วยมากมาย ง่ายในการเรียกใช้และยังมีหน้าต่างการทำงานสองแบบคือ แบบที่เป็นส่วนของการออกแบบ หรือที่เรียกว่า ส่วน Design ซึ่งจะแสดงผลเป็นรูปแบบต่างๆที่เราสร้างขึ้น และอีกส่วนหนึ่งคือส่วนที่เป็นหน้าต่างสำหรับแสดงโค้ด HTML ในส่วนนี้ก็จะเป็นการง่ายสำหรับผู้ที่สามารถเขียนภาษา HTML ได้อย่างคล่องแคล่ว และง่ายในการแก้ไขโค้ดอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

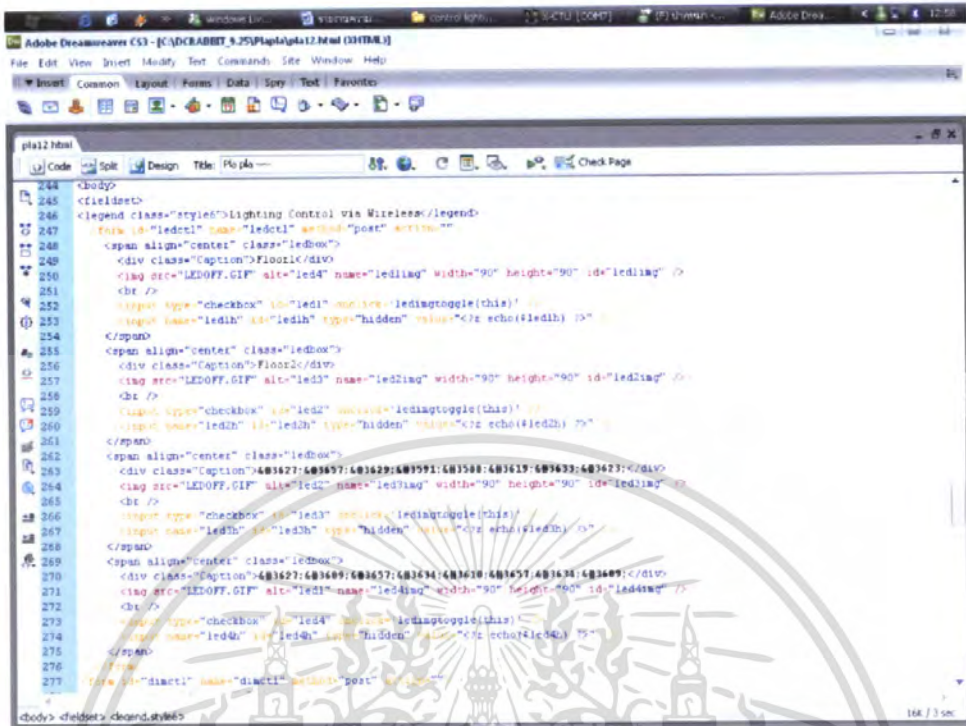


ภาพที่ 3.8 แสดงหน้าต่างพร้อมใช้งานของโปรแกรม Adobe Dreamweaver CS3

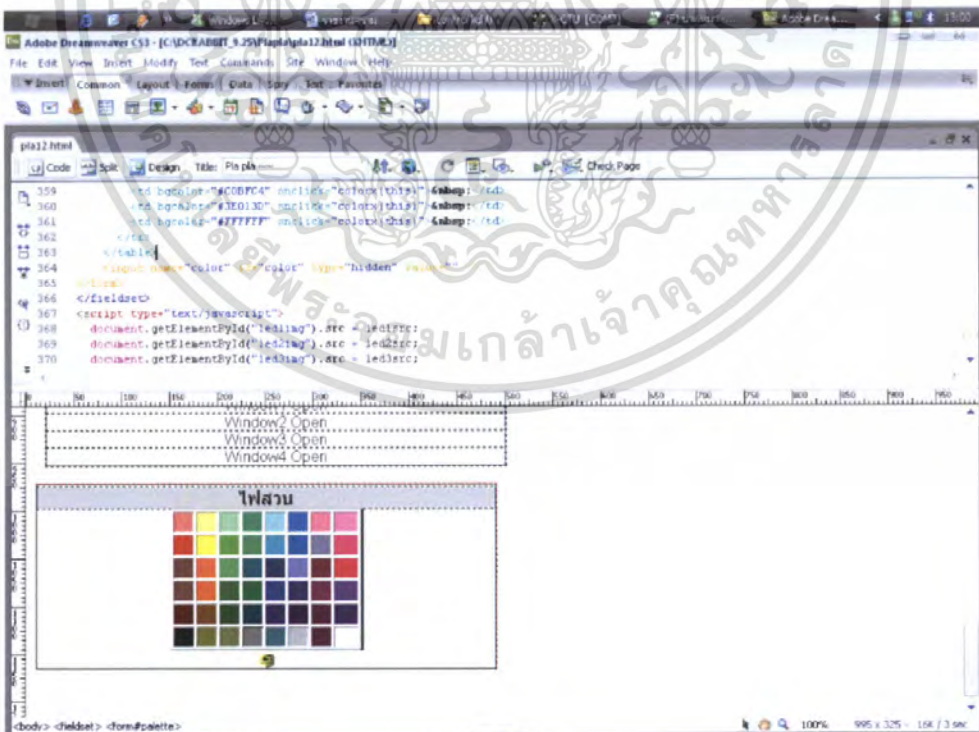


ภาพที่ 3.9 แสดงหน้าต่าง Design ของโปรแกรม Adobe Dreamweaver CS3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



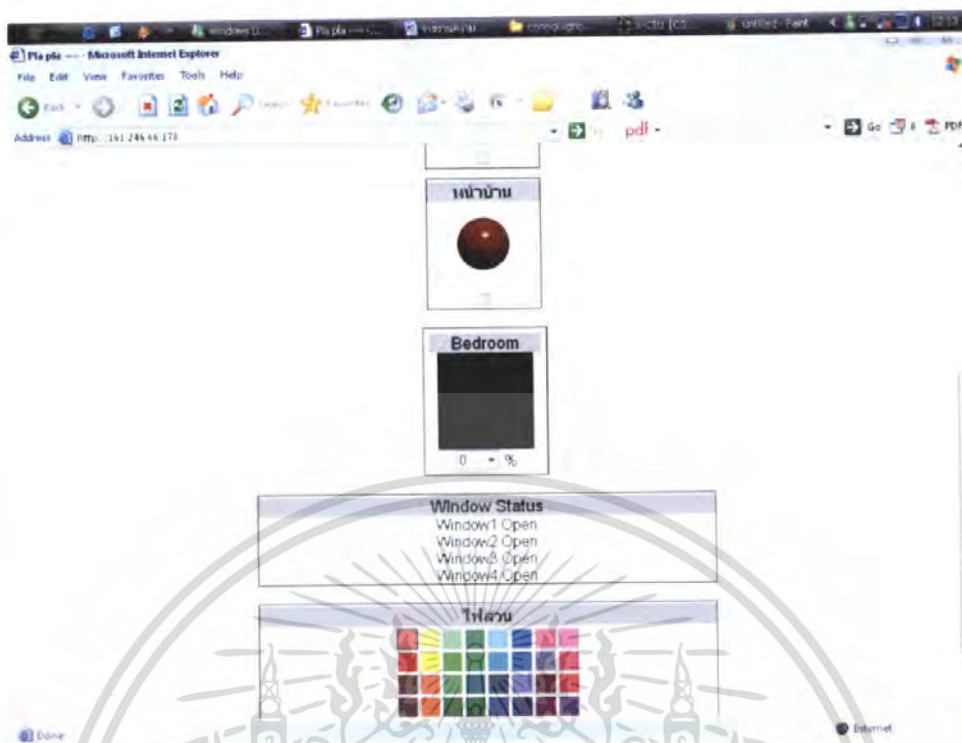
ภาพที่ 3.10 แสดงหน้าต่างแสดง code ของโปรแกรม Adobe Dreamwaver CS3



ภาพที่ 3.11 แสดงหน้าต่างการทำงานของโปรแกรม Adobe Dreamwaver CS3 ทั้งส่วน

ที่เป็นหน้าต่าง design และหน้าต่าง code HTML

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.12 แสดงหน้าต่างเว็บเพจที่ใช้ควบคุมอุปกรณ์ไฟฟ้า

### 3.5 การออกแบบโปรแกรม Microcontroller

โดยเริ่มจากการรับข้อมูลขนาดแปดบิต จากรูปแบบสัญญาณอนุกรมที่ทำการส่งงานจากหน้าเว็บเพจ เมื่อรับค่ามาแล้วตรวจสอบเงื่อนไขเพื่อทำให้โรแกรมสามารรถสั่งงานการทำงานของ Output ได้ตามต้องการ

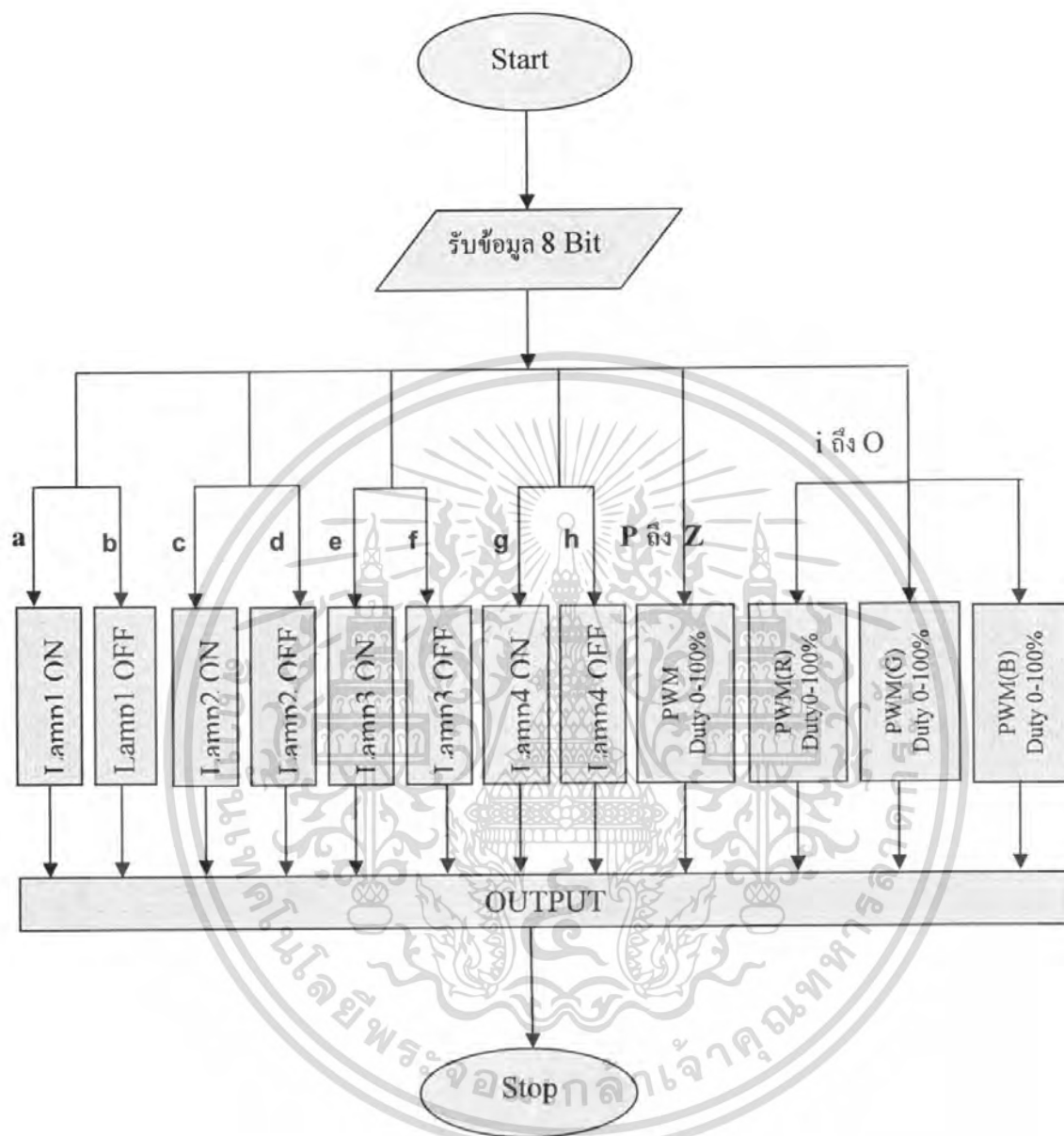
ข้อมูลจากหน้าเว็บ ในช่วง a ถึง h จะเป็นการ เปิด-ปิด อุปกรณ์การทำงาน 4 ตัว ซึ่งสามารถนำไปใช้กับอุปกรณ์ที่มีฟังก์ชันเปิด ปิด เท่านั้น

ข้อมูลจากหน้าเว็บ ในช่วง i ถึง r จะเป็นการควบคุม LED ที่สามารถปรับความสว่างได้ โดยการควบคุมสัญญาณ PWM ที่ออกจาก ไมโครคอนโทรลเลอร์ โดยสัญญาณที่รับมาจะมาควบคุม Duty Cycle จาก 0-100%

ข้อมูลจากหน้าเว็บ ในช่วง s ถึง Z จะใช้ควบคุมการทำงานของหลอด LED สามสีในโคมเดียวกัน จึงทำให้สามารถควบคุมสีได้ตามใจชอบเพราะทั้งสามชุดสามารถทำการปรับความสว่างได้จึงเกิดสีได้หลากหลายมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flowchart การทำงานของ Microcontroller



ภาพที่ 3.13 แสดง Flowchart การทำงานของ Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การทดลองและผลการทดลอง

จากการออกแบบการทำงานของระบบและสร้างชิ้นงานขึ้นมาจึงได้ทดสอบการทำงานของอุปกรณ์โดยการทำงาน Test Run จริง ว่ามีประสิทธิภาพการทำงานเป็นอย่างไร และได้ทำการเก็บค่าบางส่วนเพื่อการวิเคราะห์การทำงานของอุปกรณ์ว่าเป็นไปตามความต้องการที่ได้ออกแบบไว้หรือไม่

#### 4.1 การทดสอบการทำงานของชุด เปิด ปิด หลอดไฟ

จากการทดสอบการทำงานจะเห็นได้ว่าชุด Microcontroller สามารถทำตามคำสั่งของการควบคุมผ่านหน้าเว็บ ได้ทุกคำสั่งทั้ง 4 ชุด



ภาพที่ 4.1 แสดงสถานะของหลอดไฟบนวงจรรับเมื่อมีการส่งคำสั่งให้ Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4.2 การทดสอบการทำงานของชุดปรับแสงสว่างและชุดผสมสี

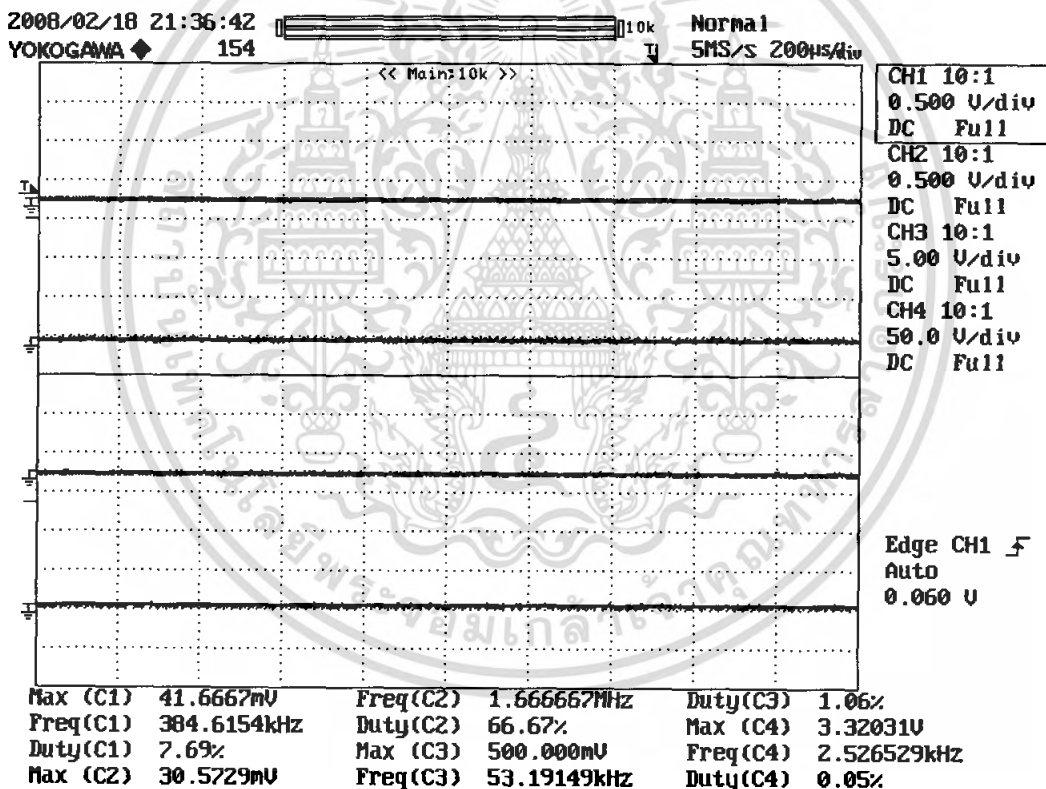
จากการทดสอบโดยทำการต่อกับวงจรขับ LED ทั้งสองแบบแล้วสั่งการทำงานโดยปรับค่า Duty Cycle ของสัญญาณ PWM ที่จ่ายออกมาจาก Microcontroller ตามการสั่งงานจากหน้าเว็บเพจ และได้ทำการเก็บรูปสัญญาณ PWM บางค่ามาเพื่อวิเคราะห์การทำงานของ Program ว่ามีประสิทธิภาพมากน้อยเพียงใด โดย

กราฟเส้นที่ 1(CH1) คือ สัญญาณ PWM ของหลอด LED ที่สามารถปรับความสว่างได้

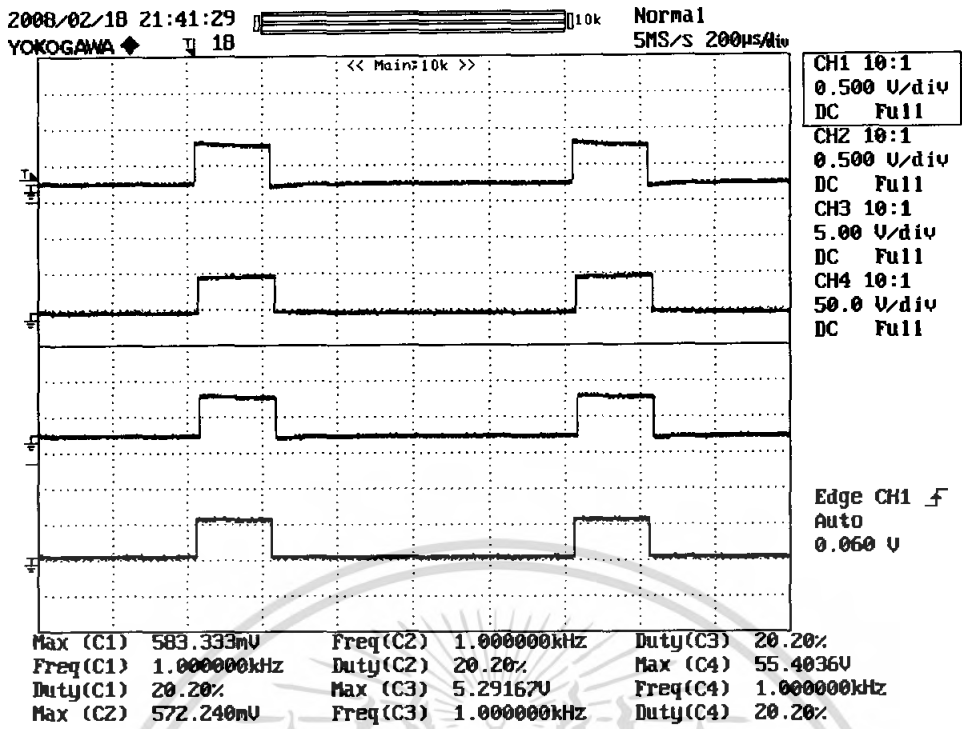
กราฟเส้นที่ 2(CH2) คือ สัญญาณ PWM ของชุดผสมสี ของสีแดง

กราฟเส้นที่ 3(CH3) คือ สัญญาณ PWM ของชุดผสมสี ของสีเขียว

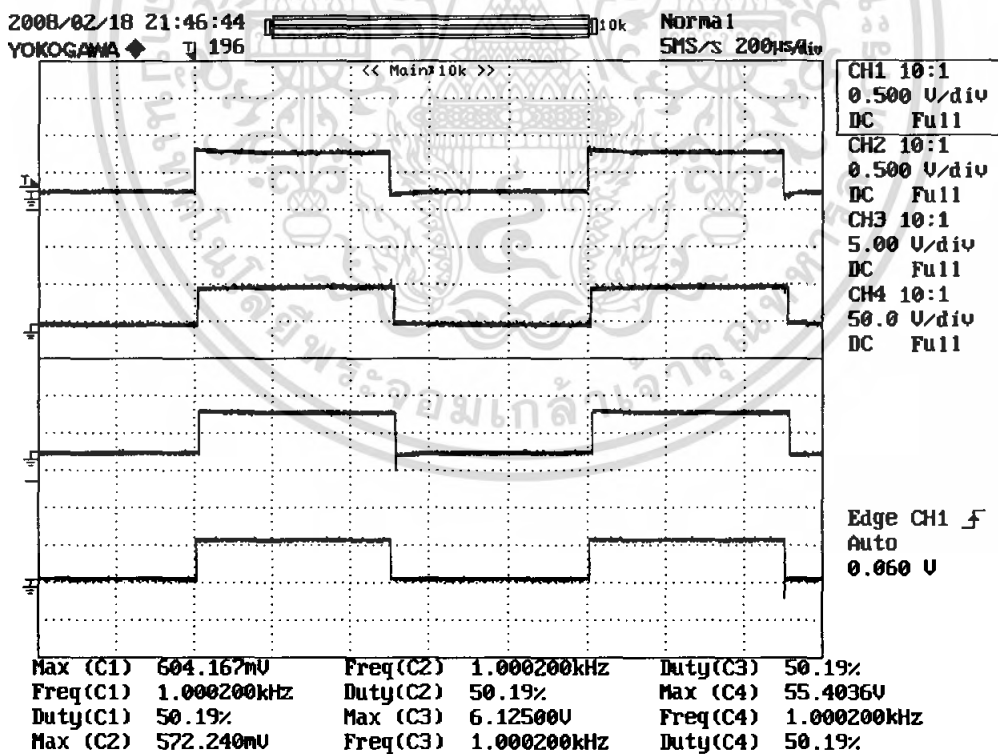
กราฟเส้นที่ 4(CH4) คือ สัญญาณ PWM ของชุดผสมสี ของน้ำเงิน



ภาพที่ 4.2 แสดงสถานการณ์ทำงานที่ 0%

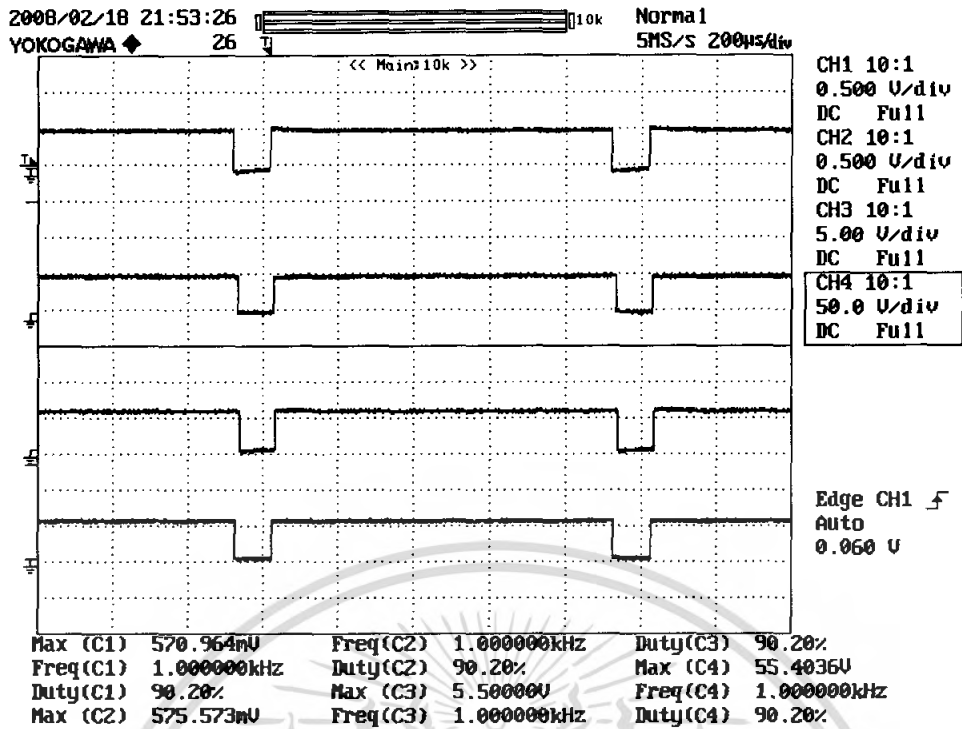


ภาพที่ 4.3 แสดงสถานการณ์ทำงานที่ 20%



ภาพที่ 4.4 แสดงสถานการณ์ทำงานที่ 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.5 แสดงสถานการณ์ทำงานที่ 90%

จากการทดลองและเก็บผลสัญญาณ PWM มาวิเคราะห์ผลการทำงานของโปรแกรมควบคุม ที่ทำงานร่วมกับ Microcontroller พบว่า ชุดทดลองทั้งสองชุดมีค่า Error เพียงเล็กน้อยเท่านั้น ทำให้ผลที่ได้ใกล้เคียงกับความต้องการของผู้ควบคุม

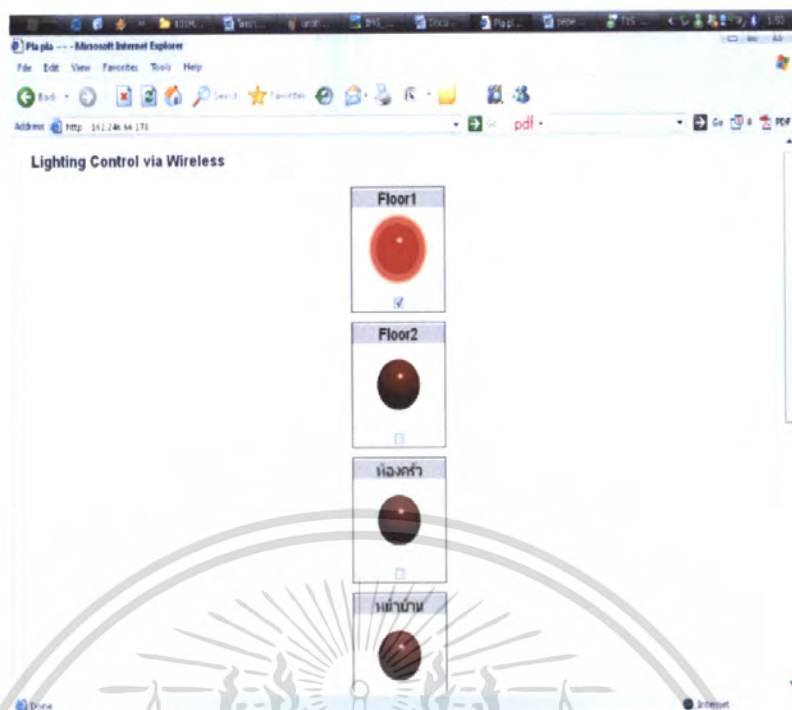
### 4.3 การทดลองการทำงานจริง

การทำงาน ก็จะต้องเข้า Internet Explorer แล้วกรอก ค่า IP Address ของเว็บเพจที่ตั้งไว้ คือ 161.246.66.178 แล้วจะมี User name และ Password ซึ่งในโครงการนี้ ได้ทำการตั้งชื่อเข้าใช้ (Username) คือ “rabbit” และรหัสเข้าใช้ (Password) เป็น 1234 ดังรูป

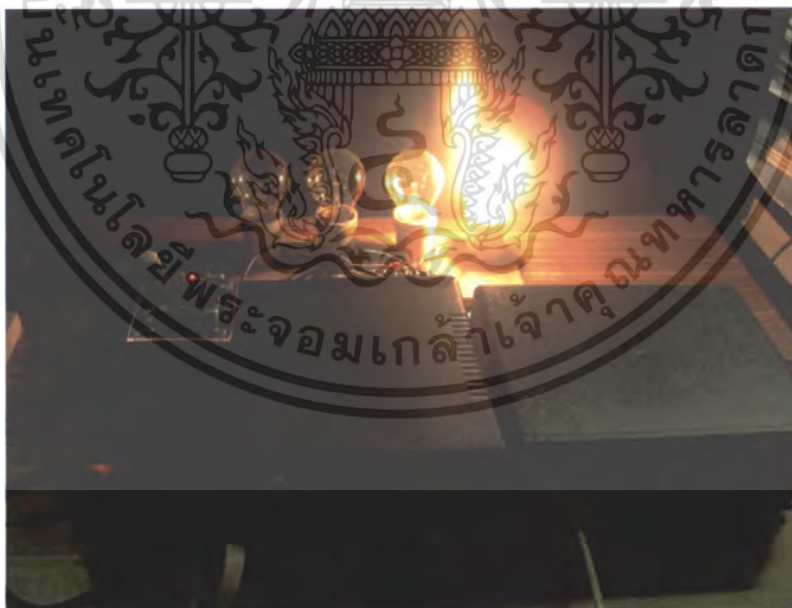


ภาพที่ 4.6 แสดงการ Log in หน้าเว็บเพจ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

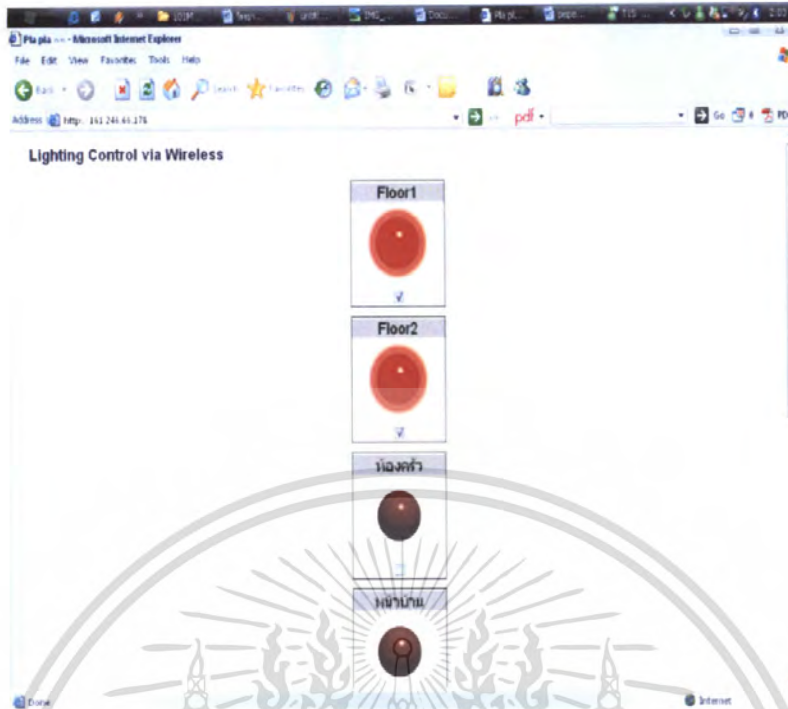


ภาพที่ 4.7 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1

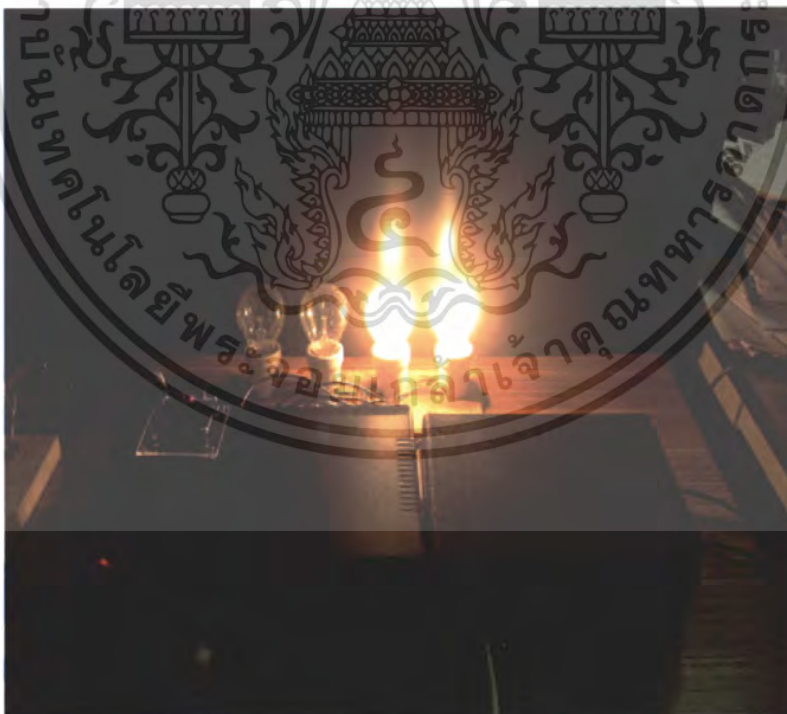


ภาพที่ 4.8 แสดงสถานะหลอดไฟหลอดที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

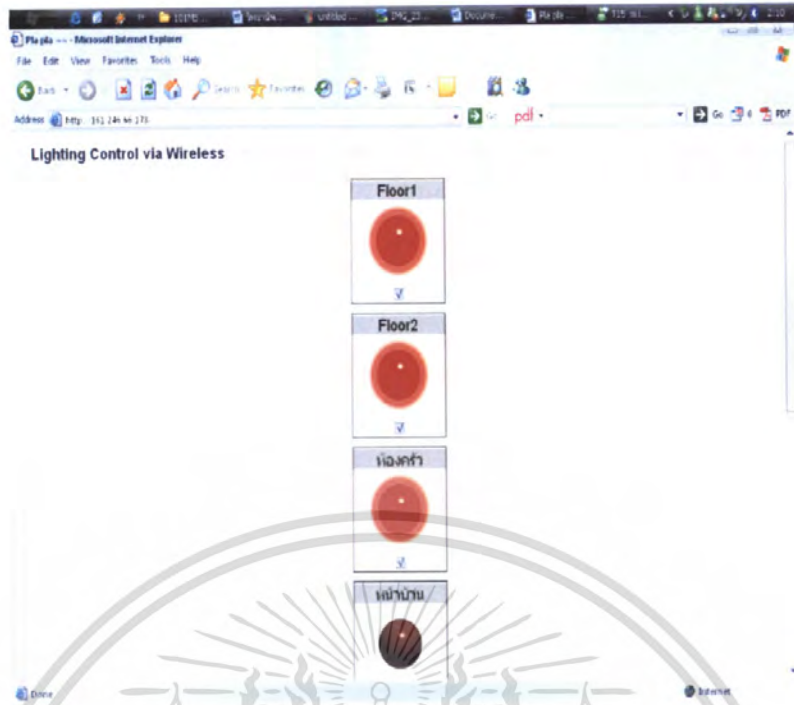


ภาพที่ 4.9 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1 และ 2



ภาพที่ 4.10 แสดงสถานะหลอดไฟหลอดที่ 1 และ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

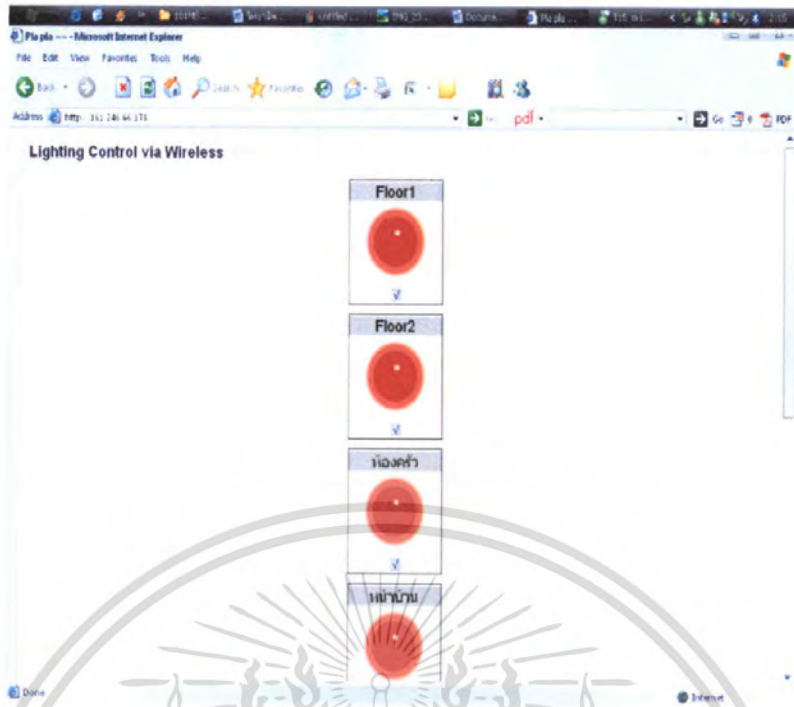


ภาพที่ 4.11 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1 , 2 และ 3



ภาพที่ 4.12 แสดงสถานะหลอดไฟหลอดที่ 1 , 2 และ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.13 แสดงหน้าเว็บเมื่อสั่งเปิดหลอดไฟหลอดที่ 1 , 2 , 3 และ 4

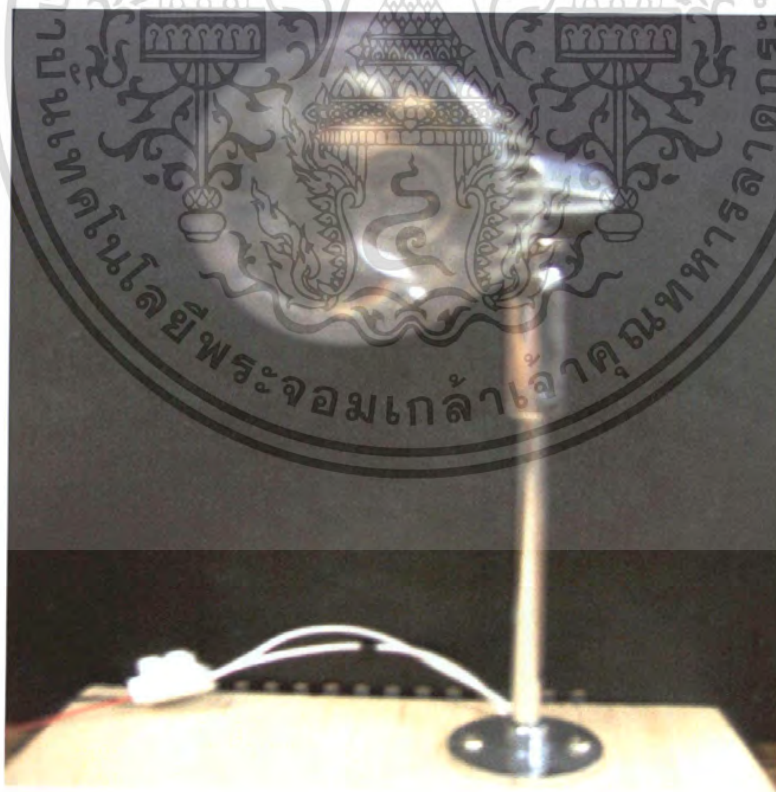


ภาพที่ 4.14 แสดงสถานะหลอดไฟหลอดที่ 1 , 2 , 3 และ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

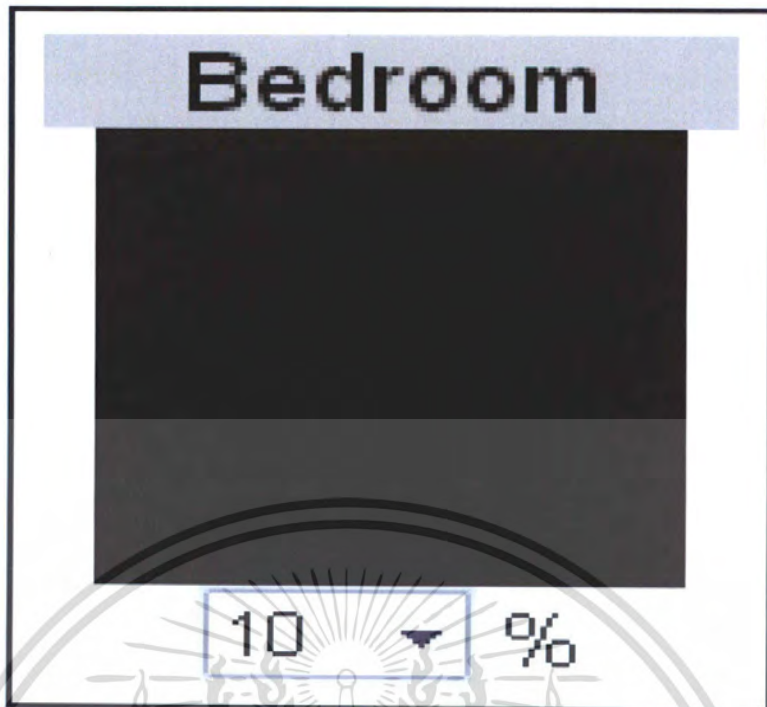


ภาพที่ 4.15 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 0 %



ภาพที่ 4.16 แสดงสถานะของโคมหลอดขาวที่ 0%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.17 แสดงหน้าเว็บเมื่อสั่งการทำงานของคอมพิวเตอร์ขาว 10 %



เอกสารนี้เป็นเอกสารที่สงวนไว้ภาพที่ 4.18 แสดงสถานะของคอมพิวเตอร์ขาวที่ 10% ถ้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

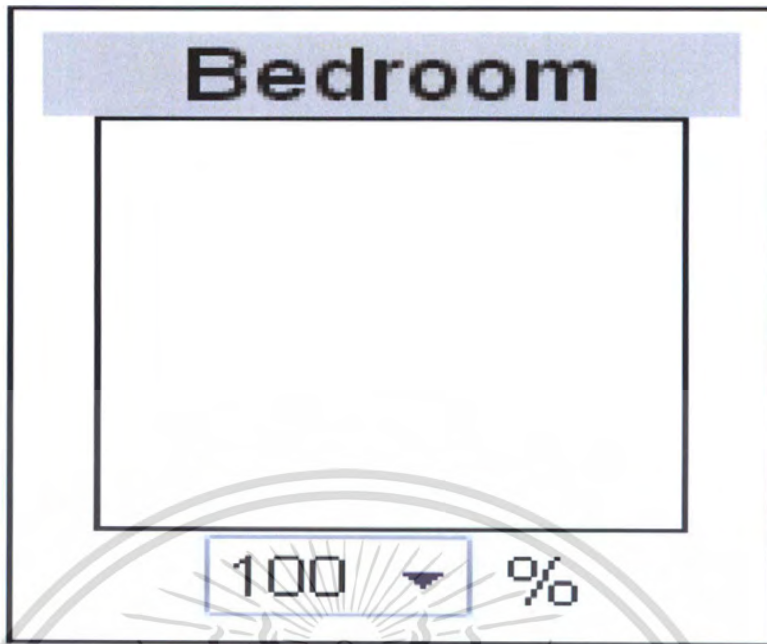


ภาพที่ 4.19 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 50 %



ภาพที่ 4.20 แสดงสถานะของโคมหลอดขาวที่ 50%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.21 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอดขาว 100 %

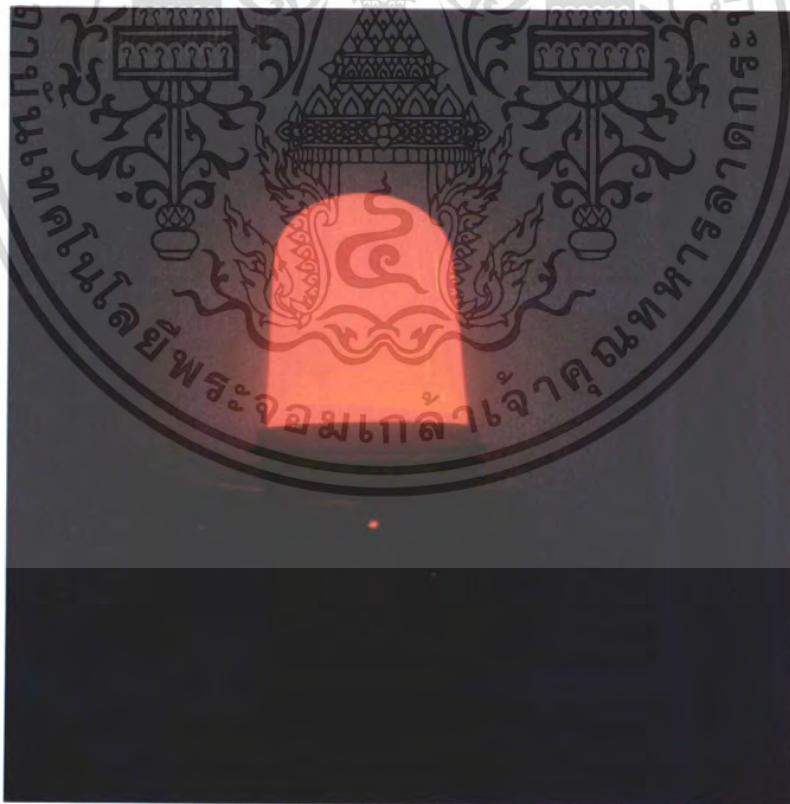


ภาพที่ 4.22 แสดงสถานะของโคมหลอดขาวที่ 100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

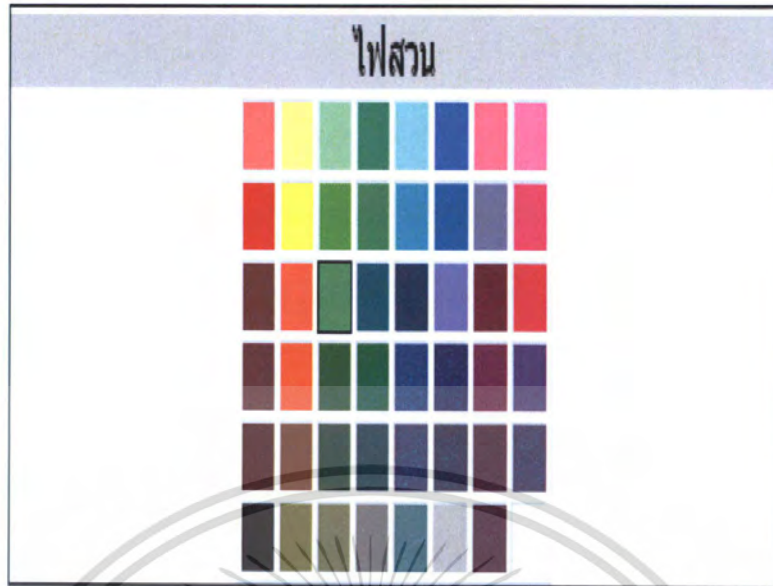


ภาพที่ 4.23 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีแดง)

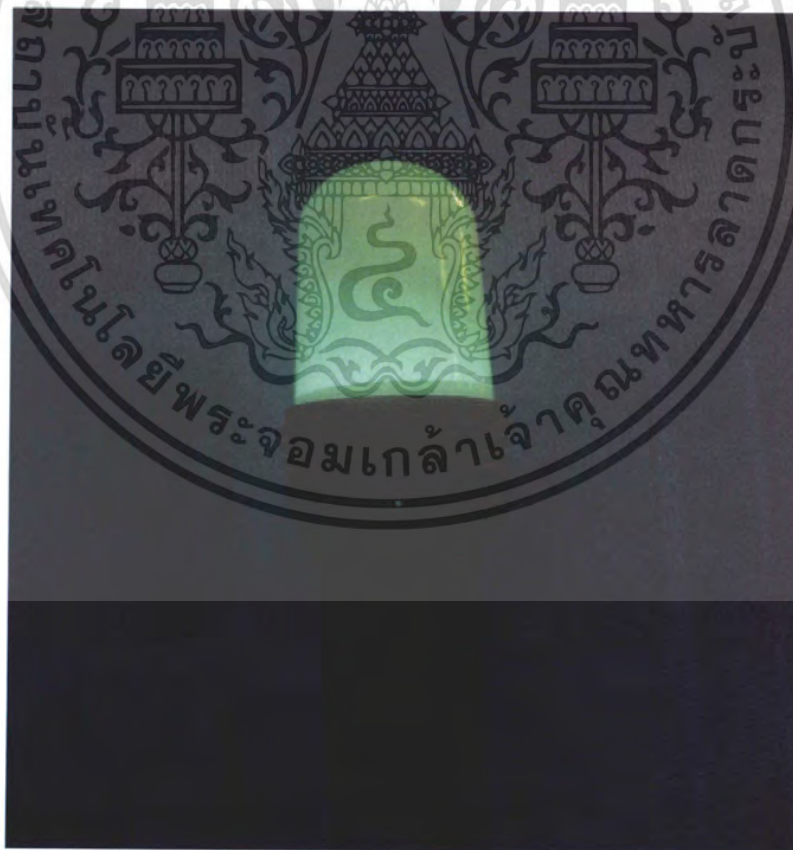


ภาพที่ 4.24 แสดงสถานะโคมหลอด 3 สี (สีแดง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.25 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีเขียว)

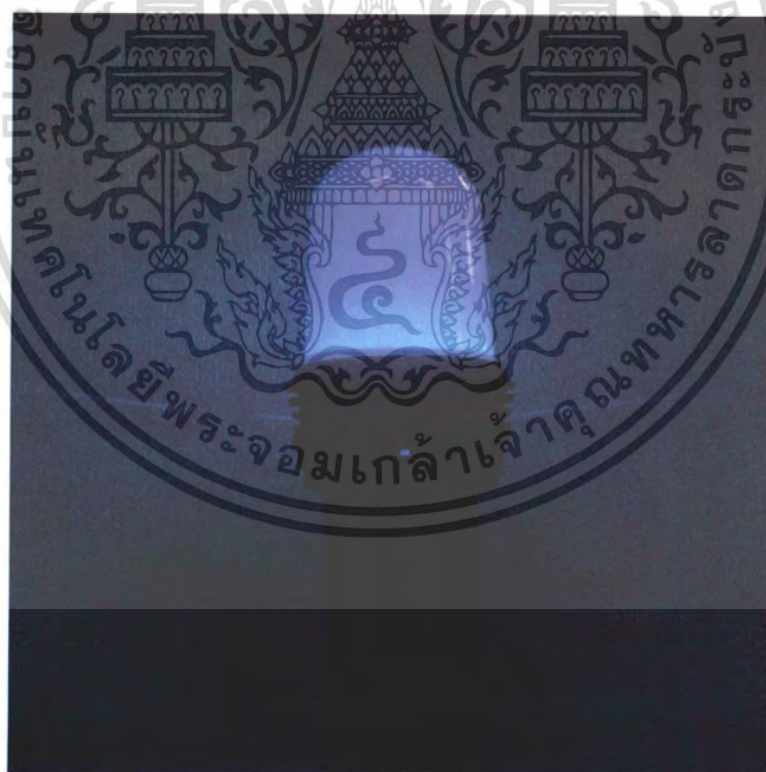


ภาพที่ 4.26 แสดงสถานะโคมหลอด 3 สี (สีเขียว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.27 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีน้ำเงิน)

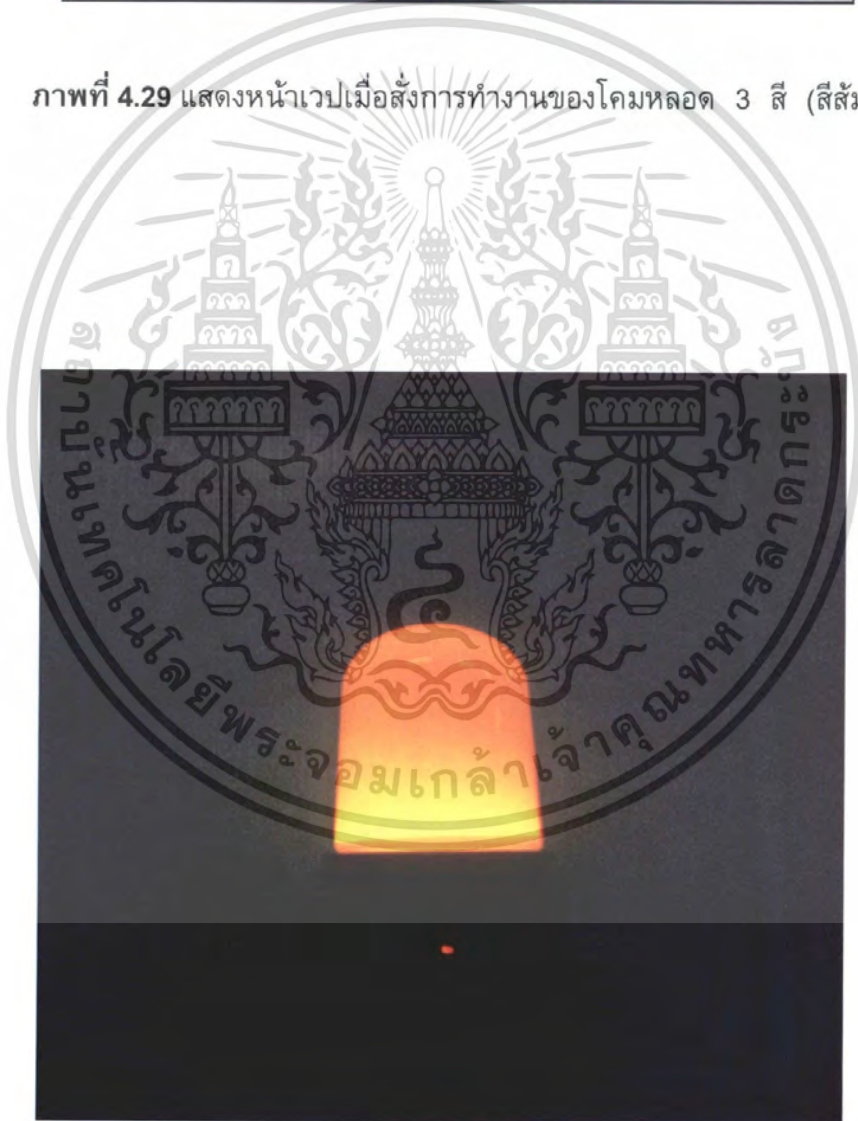


ภาพที่ 4.28 แสดงสถานะโคมหลอด 3 สี (สีน้ำเงิน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

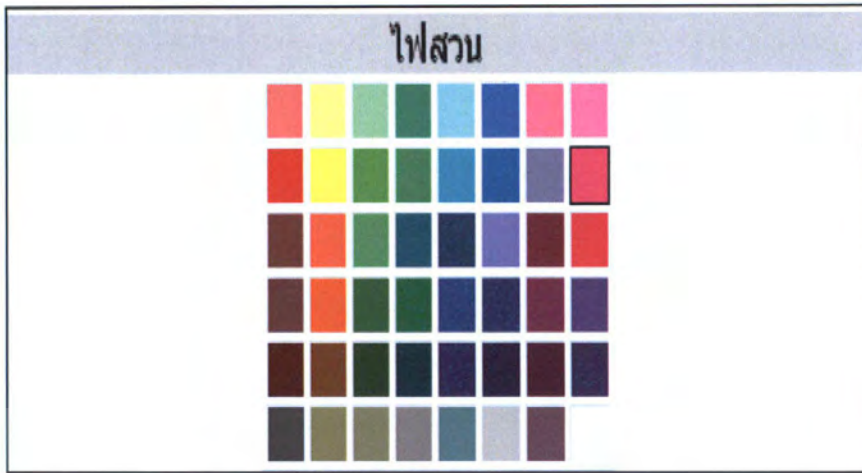


ภาพที่ 4.29 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีส้ม)

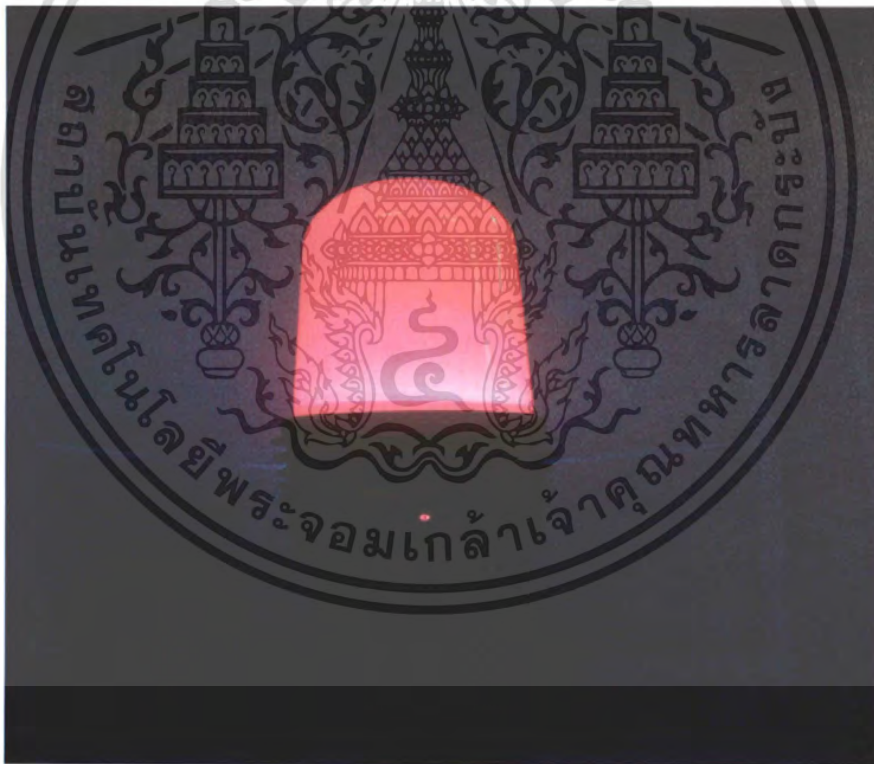


ภาพที่ 4.30 แสดงสถานะโคมหลอด 3 สี (สีส้ม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.31 แสดงหน้าเว็บเมื่อสั่งการทำงานของโคมหลอด 3 สี (สีชมพู)

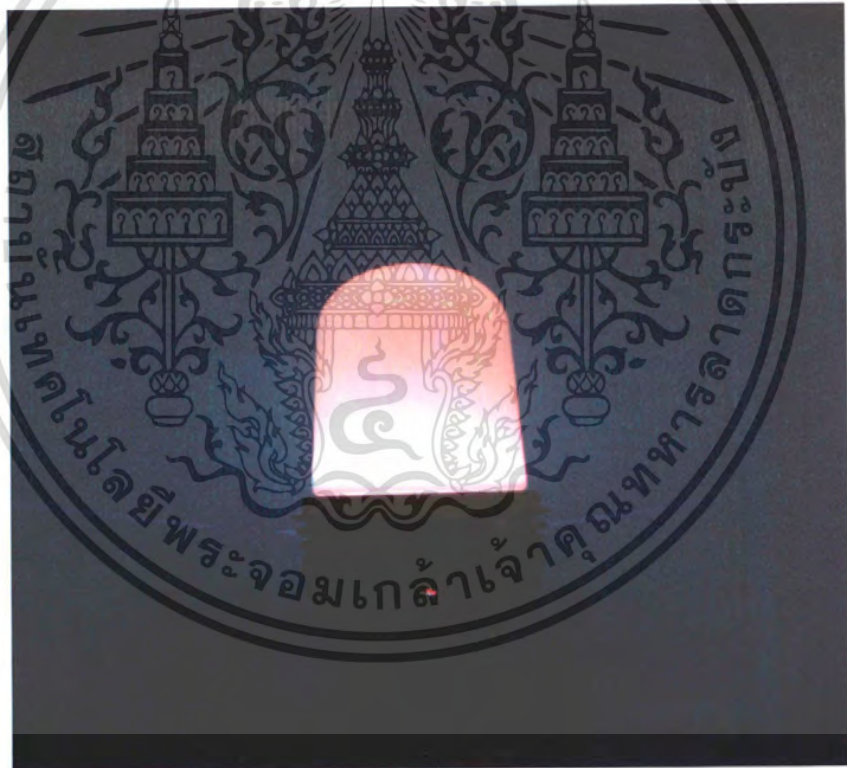


ภาพที่ 4.32 แสดงสถานะโคมหลอด 3 สี (สีชมพู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.33 แสดงหน้าเว็บเมื่อส่งการทำงานของคอมหลอด 3 สี (สีขาว)



ภาพที่ 4.34 แสดงสถานะคอมหลอด 3 สี (สีชมพู)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Window Status
Window1 Close
Window2 Open
Window3 Open
Window4 Open

ภาพที่ 4.35 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 1

Window Status
Window1 Open
Window2 Close
Window3 Open
Window4 Open

ภาพที่ 4.36 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 2

Window Status
Window1 Open
Window2 Open
Window3 Close
Window4 Open

ภาพที่ 4.37 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Window Status
Window1 Open
Window2 Open
Window3 Open
Window4 Close

ภาพที่ 4.38 แสดงหน้าเว็บเมื่อกดสวิตช์ที่ 4



ภาพที่ 4.39 แสดงภาพอุปกรณ์ที่ใช้ทดลองทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุปและวิจารณ์โครงงาน

#### 5.1 สรุปโครงงาน

การควบคุมในโครงงานที่ทำนี้ จะช่วยในเรื่องของความเสถียรสบายในการสั่งเปิด - ปิด การทำงานของอุปกรณ์ไฟฟ้า เพื่อให้มีความรวดเร็วและสะดวกยิ่งขึ้น

จากการทดลองที่ผ่านมาทางกลุ่มสามารถเปิด - ปิด หลอดไฟ ควบคุมความสว่างของ หลอด Power LED สีขาว และผสมสีหลอด Power LED ทั้ง 3 สี ได้ด้วยการสั่งจากหน้า เวปเพจทางอินเตอร์เน็ต ซึ่งได้สร้างขึ้นและให้ออนไลน์ตลอดเวลาด้วยโปรแกรม Dynamic C และทำการส่งคำสั่งไปที่ ZigBee คำสั่งจะถูกส่งไปที่ ZigBee อีกตัวผ่านทางระบบ Wireless คำสั่งจะถึงตัวไมโคร ซึ่งสามารถตอบสนองความต้องการในการสั่งงานได้หลากหลาย โดยสั่ง ผ่านทางหน้าเวปเพจ ได้เป็นอย่างดี

ในส่วนของการรับคำสั่งทางไมโคร แล้วส่งคำสั่งกลับไปยังหน้าเวปเพจนั้น ทางกลุ่มได้ ทางกลุ่มได้ทดลองทำแล้วได้รับผลการทดลองที่ดี จึงทำให้โครงงานนี้สามารถพัฒนาไปในทาง ตรวจเช็คระบบความปลอดภัยได้อีกด้วย

เมื่อมองโครงงานนี้โดยรวมจะเห็นว่าการศึกษาและค้นคว้าเนื้อหาของโครงงานและการ ประกอบชิ้นงานและทำการทดลองนั้น ผลที่ได้สามารถที่จะยืนยันการสั่งงาน โดยควบคุมผ่าน ทางหน้าเวปเพจ ได้เป็นอย่างดี เป็นไปตามวัตถุประสงค์และขอบเขตของโครงงานที่วางไว้ทุก ประการ

สุดท้ายนี้ จากการเริ่มต้นทำโครงงานและทดลองในส่วนต่างๆนั้น ทำให้ผู้ทำได้รับความรู้และประสบการณ์ในการทำงานใหม่ๆ ได้มองเห็นถึงคุณสมบัติของอุปกรณ์ต่างๆ และความผิดพลาดรวมไปถึงการแก้ปัญหาในการทำงาน ซึ่งจะนำไปปรับแก้และประยุกต์ใช้ในการ ทำงานและการดำเนินชีวิตประจำวันต่อไปในอนาคต

#### 5.2 ปัญหา อุปสรรค และการแก้ไข

จากการทำงานที่ผ่านมาสามารถที่จะสรุปปัญหา และอุปสรรค รวมถึงการแก้ไขดังนี้ เนื่องจากอุปกรณ์ Rabbit และ ZigBee เป็นอุปกรณ์ใหม่ที่ทางผู้จัดทำไม่เคยใช้มาก่อน ประกอบกับการเขียนโปรแกรมด้วยภาษา Dynamic C ผู้จัดทำไม่มีความรู้และความ ชำนาญในการเขียนมาก่อนเลย จึงทำให้เสียเวลาในการศึกษาค่อนข้างมาก ทำให้การ ดำเนินงานในขั้นตอนต่อไปนั้นช้ากว่ากำหนดที่คาดเอาไว้มาก

ทางด้านปัญหาของการจัดทำปริญญาณิพนธ์นั้น ส่วนใหญ่มาจากการรวบรวมข้อมูล การคัดเลือกข้อมูลที่ถูกต้องเหมาะสมกับโครงงาน ซึ่งข้อมูลของอุปกรณ์ มีหลากหลายเนื้อหาที่ ต่างกันจึงต้องนำมาประยุกต์ให้ได้ข้อมูลที่ดีและเหมาะสมซึ่งทั้งหมดนี้ก็เป็นปัญหาอีกทางหนึ่ง ที่ทำให้เกิดความล่าช้าในการทำรูปเล่มปริญญาณิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากปัญหาและอุปสรรคที่เกิดขึ้นนั้น มีผลทำให้โครงการสะดุดและล่าช้า ตามที่ได้กล่าวมาแล้วนั้น แนวทางการแก้ไขของกลุ่ม อันดับแรกคือการศึกษาหาข้อมูลเพิ่มเติมในขอบเขตที่พอจะสามารถศึกษาและค้นคว้าด้วยตัวเองได้ แต่ถ้าไม่สามารถแก้ไขได้ ก็จะขอคำปรึกษาจากเพื่อนๆ ภาควิชาวิศวกรรมคอมพิวเตอร์และภาควิชาวิศวกรรมสารสนเทศ พี่ๆ ปริญญาโท สุดท้ายถ้ายังไม่ได้อีกก็จะทำการขอคำปรึกษา คำอธิบาย หรือการแก้ไขปัญหาจากอาจารย์ที่ปรึกษา ซึ่งจากการปรึกษาและขอคำแนะนำ ทำให้ข้อผิดพลาดของโครงการลดน้อยลง และประสบความสำเร็จในที่สุด

### 5.3 แนวทางการพัฒนา

สามารถที่จะนำไปใช้ความคุมการทำงานของอุปกรณ์ไฟฟ้าอย่างอื่นในโรงงานอุตสาหกรรม ไร่ สวน ฟาร์ม ได้ สามารถทำการควบคุมรับส่งข้อมูล และนำข้อมูลมาแสดงผล ได้หลายอย่างในโปรแกรมเดียวกัน และยังสามารพัฒนาไปทางด้านระบบรักษาความปลอดภัยได้อีกด้วย เช่น แสดงสถานการณ์เข้าออกและบันทึกเวลาของหอสมุด โดยการติดตั้ง Sensor และ Zigbee เมื่อเวลาคนผ่านก็จะใช้ Sensor นับจำนวนและส่งข้อมูลผ่าน Zigbee และบันทึกเวลาโดย Real Time Clock ของ Rabbit เมื่อเราเข้าตรวจเช็คหน้าเว็บ ก็จะแสดง Log ของการเข้าหอสมุด จำนวนคน และเวลา เป็นต้น

จากการทดลองพบว่าระบบอินเตอร์เน็ตบางเครือข่ายไม่รองรับภาษาสคริปอาจทำให้เกิดความผิดพลาดที่หน้าเว็บจึงจำเป็นที่จะต้องพัฒนาในขั้นต่อไป

## บรรณานุกรม

- [1] นคร ภักดีชาติ และคณะ, “คู่มือการทดลองเบื้องต้น dsPIC microcontroller ด้วยโปรแกรมภาษา C กับ MPLAB C30”, กรุงเทพฯ, บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด, 2550
- [2] เกียรติกร แสงภูเขียว, คุณาวุฒิ บุญมาก, เฉลิมเกียรติ สายยศ, นิวัฒน์ กุลครอง, “ชุดควบคุมระบบแสงสว่างผ่านระบบ LAN(IP Lighting)”, ปรินญาณิพนธ์ ปี 2549 ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2549
- [3] [www.rabbitsemiconductor.com](http://www.rabbitsemiconductor.com)
- [4] โสรัถย์ อุณหะวารการ, “เซมิคอนดักเตอร์อิเล็กทรอนิกส์.” ฉบับที่ 235 กรุงเทพฯ
- [5] ปริมาภรณ์ เนตรวิกรม, “เซมิคอนดักเตอร์อิเล็กทรอนิกส์.” ฉบับที่ 264 กรุงเทพฯ
- [6] Z-World, “Dynamic C [CD-ROM]”
- [7] อุดม รานอก, “ภาษา C สำหรับงานควบคุมไมโครคอนโทรลเลอร์”, นนทบุรี : ไอทีซี , 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมคอนโทล (Dynamic C)

```

#define TCPCONFIG 1
#define USE_RABBITWEB 1

#define memmap xmem
#define use "dcrtcp.lib"
#define use "http.lib"

#define ximport "pla12.html" index_html
#define ximport "window.html" window_html
#define ximport "LEDON.GIF" LEDON_GIF
#define ximport "LEDOFF.GIF" LEDOFF_GIF
#define ximport "main.c" main_c

#define DS1 0x40 //led, port F bit 6 bitmask
#define DS2 0x80 //led, port F bit 7 bitmask

#define TCP_BUF_SIZE 2048
#define USE_HTTP_DIGEST_AUTHENTICATION 1

#define REDIRECTHOST __PRIMARY_STATIC_IP
#define REDIRECTTO "http://" REDIRECTHOST "/index.html"

#define SERD_RTS_PORT PCDR
#define SERD_RTS_SHADOW PCDRShadow
#define SERD_RTS_BIT 2 //PC2; TxC on protoboard
#define SERD_CTS_PORT PCDR
#define SERD_CTS_BIT 3 //PC3; RxC on protoboard

// serial D buffer size
#define DINBUFSIZE 15
#define DOUTBUFSIZE 15

// serial baud rate

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#ifdef _232BAUD
#define _232BAUD 9600
#endif
/*
int update_led(HttpState* state) {
    printf("%d\n", led1h);
    printf("%d\n", led2h);
    printf("%d\n", led3h);
    printf("%d\n", led4h);

    cgi_redirectto(state,REDIRECTTO);
    return 0;
} */

SSPEC_MIMETABLE_START
    SSPEC_MIME_FUNC(".html", "text/html", zhtml_handler),
    SSPEC_MIME(".gif", "image/gif")
SSPEC_MIMETABLE_END

SSPEC_RESOURCETABLE_START
    SSPEC_RESOURCE_XMEMFILE("/", index_html),
    SSPEC_RESOURCE_XMEMFILE("/index.html", index_html),
    SSPEC_RESOURCE_XMEMFILE("/window.html", window_html),
    SSPEC_RESOURCE_XMEMFILE("/LEDON.GIF", LEDON_GIF),
    SSPEC_RESOURCE_XMEMFILE("/LEDOFF.GIF", LEDOFF_GIF),
    SSPEC_RESOURCE_XMEMFILE("main.c", main_c)/*
    SSPEC_RESOURCE_FUNCTION("/updateled.cgi", update_led) */
SSPEC_RESOURCETABLE_END

int led1h;
int led2h;
int led3h;
int led4h;
int o1, o2, o3, o4;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int color;
int dims;
int wnd1, wnd2, wnd3,wnd4;
```

```
#web led1h
#web led2h
#web led3h
#web led4h
#web color;
#web dims;
#web wnd1;
#web wnd2;
#web wnd3;
#web wnd4;
```

```
void sends(char *cset) {
    serDputc (cset[0]);
    serDputc (cset[1]);
    serDputc (cset[2]);
}
```

```
void update_outputs()
{
    auto int value;
    auto int charin;
    value=PFDRShadow&0x3F;
    charin = serDgetc();
```

```
    if (led1h==0) {
        value|=DS1;
        serDputc ('b');
    } else if (led1h!=0) {
        serDputc ('a');
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

    if (led2h==0) {
        value|=DS2;
    serDputc ('d');
    } else if (led2h!=0) {
    serDputc ('c');
}

if (led3h==0) {
    serDputc ('f');
    } else if (led3h!=0) {
    serDputc ('e');
}

    if (led4h==0) {
    serDputc ('h');
    } else if (led4h!=0) {
    serDputc ('g');
}

switch (color) {
    case 1: sends ("syJ"); break;
    case 2: sends ("sDJ"); break;
    case 3: sends ("nDJ"); break;
    case 4: sends ("Dji"); break;
    case 5: sends ("nDO"); break;
    case 6: sends ("yOi"); break;
    case 7: sends ("syM"); break;
    case 9: sends ("stE"); break;
    case 10: sends ("sDE"); break;
    case 11: sends ("nDE"); break;
    case 12: sends ("DHi"); break;
    case 13: sends ("DOi"); break;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

case 14: sends ("yMi"); break;  
 case 15: sends ("nyM"); break;  
 case 16: sends ("sOt"); break;  
 case 17: sends ("nwH"); break;  
 case 18: sends ("syH"); break;  
 case 19: sends ("DiE"); break;  
 case 20: sends ("yJi"); break;  
 case 21: sends ("wJi"); break;  
 case 22: sends ("nyO"); break;  
 case 23: sends ("nHt"); break;  
 case 24: sends ("sJt"); break;  
 case 25: sends ("ntE"); break;  
 case 26: sends ("syE"); break;  
 case 27: sends ("yiE"); break;  
 case 28: sends ("yHi"); break;  
 case 29: sends ("Oit"); break;  
 case 30: sends ("Kit"); break;  
 case 31: sends ("nJt"); break;  
 case 32: sends ("nOt"); break;  
 case 33: sends ("ItE"); break;  
 case 34: sends ("nwE"); break;  
 case 35: sends ("wiE"); break;  
 case 36: sends ("wHi"); break;  
 case 37: sends ("yiE"); break;  
 case 38: sends ("Hit"); break;  
 case 39: sends ("IHt"); break;  
 case 40: sends ("IJt"); break;  
 case 41: sends ("itE"); break;  
 case 42: sends ("nyE"); break;  
 case 43: sends ("nyH"); break;  
 case 44: sends ("nyJ"); break;  
 case 45: sends ("lyJ"); break;  
 case 46: sends ("qBM"); break;  
 case 47: sends ("IHt"); break;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 48: sends ("sDO"); break;
default: sends ("sDO"); break;
}

```

```

switch (dims) {
    case 0: serDputc('P'); break;
    case 10: serDputc('Q'); break;
    case 20: serDputc('R'); break;
    case 30: serDputc('S'); break;
    case 40: serDputc('T'); break;
    case 50: serDputc('U'); break;
    case 60: serDputc('V'); break;
    case 70: serDputc('W'); break;
    case 80: serDputc('X'); break;
    case 90: serDputc('Y'); break;
    case 100: serDputc('Z'); break;
    default: serDputc('P'); break;
}

```

```

switch (charin) {
    case '0': {
        wnd1 = wnd2 = wnd3 = wnd4 = 0;
    } break;
    case '1': {
        wnd1 = wnd2 = wnd3 = 0;
        wnd4 = 1;
    } break;
    case '2': {
        wnd1 = wnd2 = wnd4 = 0;
        wnd3 = 1;
    } break;
    case '3': {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

wnd1 = wnd2 = 0;
wnd3 = wnd4 = 1;
} break;
case '4': {
    wnd1 = wnd3 = wnd4 = 0;
    wnd2 = 1;
} break;
case '5': {
    wnd1 = wnd3 = 0;
    wnd2 = wnd4 = 1;
} break;
case '6': {
    wnd1 = wnd4 = 0;
    wnd2 = wnd3 = 1;
} break;
case '7': {
    wnd1 = 0;
    wnd2 = wnd3 = wnd4 = 1;
} break;
case '8': {
    wnd1 = 1;
    wnd2 = wnd3 = wnd4 = 0;
} break;
case '9': {
    wnd1 = wnd4 = 1;
    wnd2 = wnd3 = 0;
} break;
case '!': {
    wnd1 = wnd3 = 1;
    wnd2 = wnd4 = 0;
} break;
case '@': {
    wnd1 = wnd3 = wnd4 = 1;
    wnd2 = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

} break;
case '#': {
    wnd1 = wnd2 = 1;
    wnd3 = wnd4 = 0;
} break;
case '$': {
    wnd1 = wnd2 = wnd4 = 1;
    wnd3 = 0;
} break;
case '%': {
    wnd1 = wnd2 = wnd3 = 1;
    wnd4 = 0;
} break;
case '^': {
    wnd1 = wnd2 = wnd3 = wnd4 = 1;
} break;
//default: wnd1 = wnd2 = wnd3 = wnd4 =0; break;
}
/*
printf("%c", charin);
printf("%d%d%d%d", wnd1,wnd2,wnd3,wnd4);
*/

    WrPortl(PFDR, &PFDRShadow, value);
}

```

```

void main() {
    auto int nIn;
    auto char cOut;

    int admin, page1;

    brdInit();
    BitWrPortl(PEDR, &PEDRShadow, 0, 5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

serDopen(_232BAUD);           //initialize TxD, RxD
    serDflowcontrolOn();      //enable flow control
    serDwrFlush();           //clear buffers
    serDrdFlush();

    sock_init();

http_init();
http_setauthentication(HTTP_DIGEST_AUTH);
admin = sauth_adduser("rabbit", "1234", SERVER_HTTP);
page1 = sspec_addxmemfile("/", index_html, SERVER_HTTP);
sspec_adduser(page1, admin);
sspec_setrealm(page1, "Admin");

led1h = led2h = led3h = led4h = 0;
o1 = o2 = o3 = o4 = 0;
wnd1 = wnd2 = wnd3 = wnd4 = 0;
color = 1;
dims = 0;

for (;;) {
    http_handler();
    update_outputs();
}
}

```

## โปรแกรมสร้างหน้าเว็บเพจ

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<META HTTP-EQUIV="Pragma" CONTENT="no-cache">
<title>Pla pla ~~</title>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<script type="text/javascript">
```

```
led1st = <?z echo($led1h) ?>;
```

```
led2st = <?z echo($led2h) ?>;
```

```
led3st = <?z echo($led3h) ?>;
```

```
led4st = <?z echo($led4h) ?>;
```

```
led1src = (led1st) ? ("LEDON.GIF") : ("LEDOFF.GIF") ;
```

```
led2src = (led2st) ? ("LEDON.GIF") : ("LEDOFF.GIF") ;
```

```
led3src = (led3st) ? ("LEDON.GIF") : ("LEDOFF.GIF") ;
```

```
led4src = (led4st) ? ("LEDON.GIF") : ("LEDOFF.GIF") ;
```

```
dims = <?z echo($dims) ?>;
```

```
color_curr=<?z echo($color)?>;
```

```
function dec(objname) {
    obj = document.getElementById(objname);
    if (parseInt(obj.value) >= 10) obj.value = parseInt(obj.value) - 10;
    colorvalue = parseInt(obj.value);
    colorvalue = parseInt(255 * colorvalue / 100);
    colorvalue = colorvalue.toString(16).toUpperCase();
    colorvalue = '#' + colorvalue + colorvalue + colorvalue;
    document.getElementById('dim1').style.backgroundColor = colorvalue ;
}
```

```
function inc(objname) {
    obj = document.getElementById(objname);
    if (parseInt(obj.value) <= 90) obj.value = parseInt(obj.value) + 10;
    colorvalue = parseInt(obj.value);
    colorvalue = parseInt(255 * colorvalue / 100);
    colorvalue = colorvalue.toString(16).toUpperCase();
    colorvalue = '#' + colorvalue + colorvalue + colorvalue;
    document.getElementById('dim1').style.backgroundColor = colorvalue ;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
function dim(value, submit) {
    colorvalue = parseInt(value);
    colorvalue = parseInt(255 * colorvalue / 100);
    colorvalue = colorvalue.toString(16).toUpperCase();
    colorvalue = '#' + colorvalue + colorvalue + colorvalue;
    document.getElementById('dim1').style.backgroundColor = colorvalue ;
    if (submit == 1) document.getElementById('dimctl').submit();
}
```

```
function ledsrctoggle(getid) {
    ledsrc = document.getElementById( getid ); /*
    if (ledsrc.src.indexOf("LEDOFF.GIF") != -1) {
        ledsrc.src = "LEDON.GIF";
    }
    else {
        ledsrc.src = "LEDOFF.GIF";
    }
    */
    ledxid = getid.substring(0,4) + 'h';
    ledxst = getid.substring(0,4);
    document.getElementById(ledxid).value = (document.getElementById(ledxst).checked)
? (1) : (0) ;
}
function ledimgtoggle(ledimg) {
    switch (ledimg.id) {
        case "led1": ledsrctoggle("led1img"); break;
        case "led2": ledsrctoggle("led2img"); break;
        case "led3": ledsrctoggle("led3img"); break;
        case "led4": ledsrctoggle("led4img"); break;
        default: break;
    }
    frm = document.getElementById('ledctl');
    frm.submit();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function colormix(colorid) {
    mix = (colorid.checked == true) ? "FF" : "00";
    oldmix = document.getElementById("result");
    str = oldmix.style.backgroundColor; // alert(str);
    getred = str.substring(1, 3); //alert(getred);
    getgreen = str.substring(3, 5); //alert(getgreen);
    getblue = str.substring(5, 7); //alert(getblue);

    switch (colorid.id) {
        case "red" : getred = mix; break;
        case "green": getgreen = mix; break;
        case "blue" : getblue = mix; break;
        default: break;
    }
    oldmix.style.backgroundColor = "#" + getred + getgreen + getblue;
}

function colorx(td) {
    var colorCode = td.getAttribute('bgcolor');
    var tdCount = document.getElementById('palette').getElementsByTagName('td').length;
    var tdArray = document.getElementById('palette').getElementsByTagName('td');
    for (i = 0; i != tdCount; ++i) {
        if (tdArray[i].getAttribute('bgcolor') == colorCode) {
            document.getElementById('color').value = parseInt(i) + 1;
            //alert(document.getElementById('color').value);
            break;
        }
    }
    document.getElementById('palette').submit();
}

var xmlhttp;
var wndstr;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function loadXMLDoc(url)
{
xmlhttp=null;
if (window.XMLHttpRequest)
    { // code for Firefox, Opera, IE7, etc.
    xmlhttp=new XMLHttpRequest();
    }
else if (window.ActiveXObject)
    { // code for IE6, IE5
    xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
    }
if (xmlhttp!=null)
{
    var mytime= "?&ms="+new Date().getTime();
    url += mytime;
    xmlhttp.onreadystatechange=state_Change;
    xmlhttp.open("GET",url,true);
    xmlhttp.send(null);
}
else
    {
    alert("Your browser does not support XMLHttpRequest.");
    }
}

```

```

function state_Change()
{
if (xmlhttp.readyState==4)
    { // 4 = "loaded"
    if (xmlhttp.status==200)
        { // 200 = "OK"
        wndstr = xmlhttp.responseText; //alert(wndstr);
        wnd1 = wndstr.substring(0, 1) * 1;
        wnd2 = wndstr.substring(1, 2) * 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

wnd3 = wndstr.substring(2, 3) * 1;
wnd4 = wndstr.substring(3, 4) * 1;

if (wnd1 == 0) document.getElementById('window1').innerHTML = "Window1
Open";
else document.getElementById('window1').innerHTML = "Window1 Close";
if (wnd2 == 0) document.getElementById('window2').innerHTML = "Window2
Open";
else document.getElementById('window2').innerHTML = "Window2 Close";
if (wnd3 == 0) document.getElementById('window3').innerHTML = "Window3
Open";
else document.getElementById('window3').innerHTML = "Window3 Close";
if (wnd4 == 0) document.getElementById('window4').innerHTML = "Window4
Open";
else document.getElementById('window4').innerHTML = "Window4 Close";
}
else
{
//alert("Problem retrieving data:" + xmlhttp.statusText);
}
}
}
function wndLoadData() {
loadXMLDoc('window.html');
var t = setTimeout("wndLoadData()", 2000);
}
</script>
<style type="text/css">
<!--
.color {
margin: 0px;
padding: 0px;
height: 100px;
width: 100px;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

border: 1px solid black;
}
.color#red {
background-color: #FF0000;
}
.color#green {
background-color: #00FF00;
}
.color#blue {
background-color: #0000FF;
}
.color#result {
background-color: #000000;
}
.style1 {font-size: larger}
.style6 {
font-size: large;
font-weight: bold;
}
body,td,th {
font-family: Arial, Helvetica, sans-serif;
}
#dim1 {
width: 100px;
height: 100px;
border: 1px solid black;
background-color: #000000;
}
.Caption {font-size: larger; color: #000000; text-align: center; background-color:
#CCCCCC;
margin: 2px 0 0 0;
}
#dimctl {
display: block;

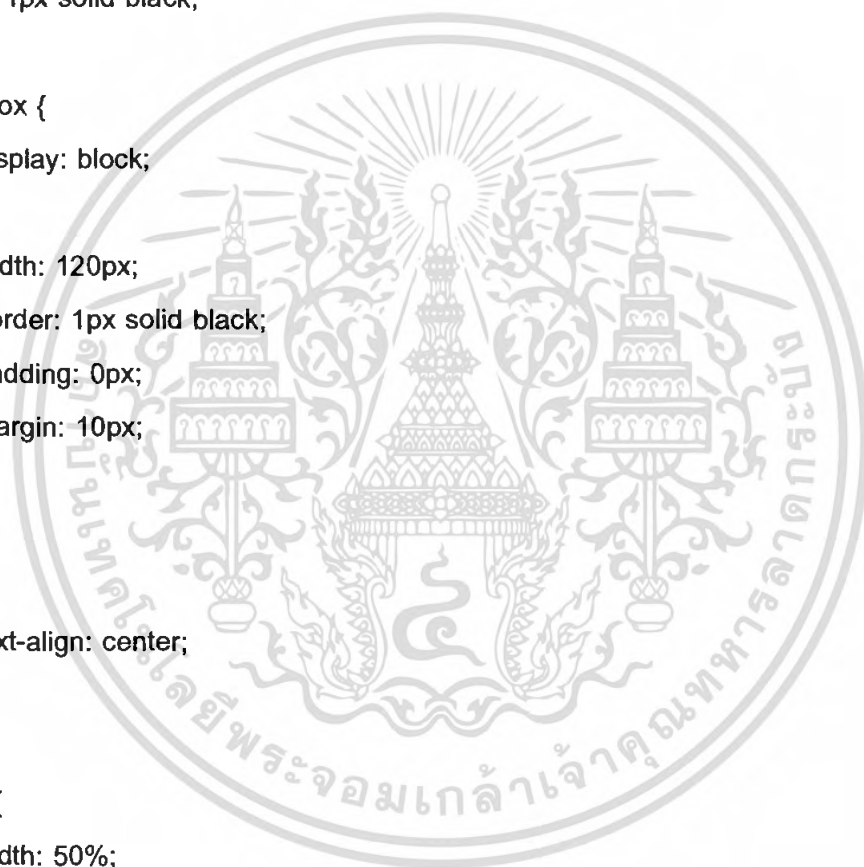
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* float: left; */
width: 120px;
border: 1px solid black;
padding: 5px;
}
#palette {
display: block;
/* float: left; */
width: 50%;
border: 1px solid black;
}
span.ledbox {
display: block;
width: 120px;
border: 1px solid black;
padding: 0px;
margin: 10px;
}
fieldset {
text-align: center;
}
#window {
width: 50%;
border: 1px solid black;
padding: 0px;
margin: 10px;
}
#window span {
display: block;
width: 100%;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

-->
</style>
</head>

<body>
<fieldset>
<legend class="style6">Lighting Control via Wireless</legend>
<form id="ledctl" name="ledctl" method="post" action="">
  <span align="center" class="ledbox">
    <div class="Caption">Floor1</div>
    
    <br />
    <input type="checkbox" id="led1" onclick='ledimgtoggle(this)' />
    <input name="led1h" id="led1h" type="hidden" value="<?z echo($led1h) ?>" />
  </span>
  <span align="center" class="ledbox">
    <div class="Caption">Floor2</div>
    
    <br />
    <input type="checkbox" id="led2" onclick='ledimgtoggle(this)' />
    <input name="led2h" id="led2h" type="hidden" value="<?z echo($led2h) ?>" />
  </span>
  <span align="center" class="ledbox">
    <div
class="Caption">&#3627;&#3657;&#3629;&#3591;&#3588;&#3619;&#3633;&#3623;</div
>
    
    <br />
    <input type="checkbox" id="led3" onclick='ledimgtoggle(this)' />

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<input name="led3h" id="led3h" type="hidden" value="<?z echo($led3h) ?>" />
</span>
<span align="center" class="ledbox">
  <div
class="Caption">&#3627;&#3609;&#3657;&#3634;&#3610;&#3657;&#3634;&#3609;</div
>
  
  <br />
  <input type="checkbox" id="led4" onclick='ledimgtoggle(this)' />
  <input name="led4h" id="led4h" type="hidden" value="<?z echo($led4h) ?>" />
</span>
</form>
<form id="dimctl" name="dimctl" method="post" action="">
  <div class="Caption">Bedroom</div>
  <div class="dim" id="dim1"></div>
  <select name="dime" id="dime" onchange="dim(this.options[this.selectedIndex].value,
1)">
  <option value="0">0</option>
  <option value="10">10</option>
  <option value="20">20</option>
  <option value="30">30</option>
  <option value="40">40</option>
  <option value="50">50</option>
  <option value="60">60</option>
  <option value="70">70</option>
  <option value="80">80</option>
  <option value="90">90</option>
  <option value="100">100</option>
  </select>
  <label for="dim">%</label>
</form>
  <div align="center" id="window"><div class="Caption">Window Status</div>
  <span id='window1'>Window1 Open</span>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<span id='window2'>Window2 Open</span>
<span id='window3'>Window3 Open</span>
<span id='window4'>Window4 Open</span> </div>
<form id="palette" name="palette" method="post" action="">
  <div class="Caption">&#3652;&#3615;&#3626;&#3623;&#3609;</div>
  <table width="200px" border="1" id="palettetbl">
    <tr>
      <td bgcolor="#F28683" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#FFFF7E" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#81FF79" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#00FF84" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#7EFFFF" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#0D7CEF" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#FC7FC3" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#FE81F9" onclick="colorx(this)">&nbsp;</td>
    </tr>
    <tr>
      <td bgcolor="#FE000A" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#FEFE00" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#87FA09" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#08FA44" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#02FDFF" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#1E74AD" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#8584B0" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#FF00FF" onclick="colorx(this)">&nbsp;</td>
    </tr>
    <tr>
      <td bgcolor="#81453B" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#FF7F3A" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#38DE36" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#0A8080" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#0A3C6F" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#807FF7" onclick="colorx(this)">&nbsp;</td>
      <td bgcolor="#7B0343" onclick="colorx(this)">&nbsp;</td>
  </table>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<td bgcolor="#FF007F" onclick="colorx(this)">&nbsp;</td>
</tr>
<tr>
<td bgcolor="#7E4241" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#FF7C02" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#097B01" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#00853F" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#0005EB" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#0001A0" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#7C0674" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#8500F4" onclick="colorx(this)">&nbsp;</td>
</tr>
<tr>
<td bgcolor="#440001" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#804100" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#004400" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#004144" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#000087" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#000047" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#3D023C" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#3D0B6A" onclick="colorx(this)">&nbsp;</td>
</tr>
<tr>
<td bgcolor="#000104" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#7C7F00" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#7D833D" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#808080" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#3B8284" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#C0BFC4" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#3E013D" onclick="colorx(this)">&nbsp;</td>
<td bgcolor="#FFFFFF" onclick="colorx(this)">&nbsp;</td>
</tr>
</table>
<input name="color" id="color" type="hidden" value="" />

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

</form>
</fieldset>
<script type="text/javascript">
    document.getElementById("led1img").src = led1src;
    document.getElementById("led2img").src = led2src;
    document.getElementById("led3img").src = led3src;
    document.getElementById("led4img").src = led4src;
    document.getElementById("led1").checked = led1st;
    document.getElementById("led2").checked = led2st;
    document.getElementById("led3").checked = led3st;
    document.getElementById("led4").checked = led4st;/*
    document.getElementById("led1h").value = (led1st) ? (0) : (1) ;
    document.getElementById("led2h").value = (led2st) ? (0) : (1) ;
    document.getElementById("led3h").value = (led3st) ? (0) : (1) ;
    document.getElementById("led4h").value = (led4st) ? (0) : (1) ;*/

    document.getElementById("dim1").style.backgroundColor = "#000000";
    document.getElementById("dims").selectedIndex = parseInt(dims) / 10;
    document.getElementById("color").value = color_curr;
    dim(dims, 0);
    var tdCount = document.getElementById('palette').getElementsByTagName('td').length;
    var tdArray = document.getElementById('palette').getElementsByTagName('td');
    for (i = 0; i != tdCount; ++i) {
        if (i == (color_curr-1)) {
            tdArray[i].style.borderColor = "black";
            break;
        }
    }
    wndLoadData();
</script>
</body>
</htm

```

## โปรแกรม Microcontroller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมควบคุมการเปิด-ปิดหลอดไฟและสวิตช์ทั้งสี่

```

#include <16F877.h>
#define CLOCK_SP 10000000
#fuses HS,NOWDT,NOPROTECT,NOPUT,NOBROWNOUT,NOLVP
#use delay(clock=10000000)
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)

#define led1 PIN_A0
#define led2 PIN_A1
#define led3 PIN_A2
#define led4 PIN_A3

////////////////////////////////////
int8 input=0,sum=0,output=0;
////////////////////////////////////
int key[10] = {'0','1','2','3','4','5','6','7','8','9'};
////////////////////////////////////
#INT_RDA
void interruptrx232()
{

input = getc();
putc(input);

if (input == 97)
output_high(led1);
if (input == 98)
output_low(led1);
if (input == 99)
output_high(led2);
if (input == 100)
output_low(led2);
if (input == 101)
output_high(led3);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (input == 102)
output_low(led3);
if (input == 103)
output_high(led4);
if (input == 104)
output_low(led4);

}

```

```

#INT_RB
void interruptrb()
{
if (input(PIN_B4))
sum = sum + 8;
if (input(PIN_B5))
sum = sum + 4;
if (input(PIN_B6))
sum = sum + 2;
if (input(PIN_B7))
sum = sum + 1;

switch (sum)
{
case 0 : putc('^'); break;
case 1 : putc('%'); break;
case 2 : putc('$'); break;
case 3 : putc('#'); break;
case 4 : putc('@'); break;
case 5 : putc('!'); break;
case 6 : putc('9'); break;
case 7 : putc('8'); break;
case 8 : putc('7'); break;
case 9 : putc('6'); break;
case 10 : putc('5'); break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case 11 : putc('4'); break;
case 12 : putc('3'); break;
case 13 : putc('2'); break;
case 14 : putc('1'); break;
case 15 : putc('0'); break;
}

```

```
sum = 0;
```

```
}
```

```
void main()
```

```
{
```

```
set_tris_a(0x00);
```

```
set_tris_b(0xff);
```

```
enable_interrupts(GLOBAL);
```

```
enable_interrupts(INT_RDA);
```

```
enable_interrupts(INT_RB);
```

```
while (TRUE)
```

```
{
```

```
}
```

```
}
```

โปรแกรมควบคุมโคมหลอดขาวและหลอดผสมสี

```
#include <16F877.h>
```

```
#define CLOCK_SP 10000000
```

```
#fuses HS,NOWDT,NOPROTECT,NOPUT,NOBROWNOUT,NOLVP
```

```
#use delay(clock=10000000)
```

```
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
```

```
#define led1 PIN_A0
```

```
#define led2 PIN_A1
```

```
#define led3 PIN_A2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#define led4 PIN_A3
#define led5 PIN_B0
#define led6 PIN_B1
#define led7 PIN_B2
#define led8 PIN_B4

/////////////////////////////////////////////////////////////////
int8 input=0,a=10,b=10,c=10,d=10,round=0,none=0;
/////////////////////////////////////////////////////////////////
int key[10] = {'0','1','2','3','4','5','6','7','8','9'};
/////////////////////////////////////////////////////////////////
#INT_TIMER0
void interrupt()
{
  disable_interrupts(INT_TIMER0);
  set_timer0(43);

  round++;

  if (round <= 10)
  {

    if (a >= round)
      output_high(led5);
    if (a < round)
      output_low(led5);
    if (b >= round)
      output_high(led6);
    if (b < round)
      output_low(led6);
    if (c >= round)
      output_high(led7);
    if (c < round)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

output_low(led7);
if (d >= round)
output_high(led8);
if (d < round)
output_low(led8);
}
if (round == 10)
{
round = 0;
}
//output_high(led5);
//output_low(led5);
enable_interrupts(INT_TIMER0);
}
void main()
{
set_tris_a(0x00);
set_tris_b(0x00);
enable_interrupts(GLOBAL);
enable_interrupts(INT_TIMER0);
setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
set_timer0(43);
while (TRUE)
{
input = getc();
putc(input);
if (input == 97)
output_high(led1);
if (input == 98)
output_low(led1);
if (input == 99)
output_high(led2);
if (input == 100)
output_low(led2);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (input == 101)
output_high(led3);
if (input == 102)
output_low(led3);
if (input == 103)
output_high(led4);
if (input == 104)
output_low(led4);
if (input == 105)
a = 10;
if (input == 106)
a = 9;
if (input == 107)
a = 8;
if (input == 108)
a = 7;
if (input == 109)
a = 6;
if (input == 110)
a = 5;
if (input == 111)
a = 4;
if (input == 112)
a = 3;
if (input == 113)
a = 2;
if (input == 114)
a = 1;
if (input == 115)
a = 0;
if (input == 116)
b = 10;
if (input == 117)
b = 9;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (input == 118)
b = 8;
if (input == 119)
b = 7;
if (input == 120)
b = 6;
if (input == 121)
b = 5;
if (input == 122)
b = 4;
if (input == 65)
b = 3;
if (input == 66)
b = 2;
if (input == 67)
b = 1;
if (input == 68)
b = 0;
if (input == 69)
c = 10;
if (input == 70)
c = 9;
if (input == 71)
c = 8;
if (input == 72)
c = 7;
if (input == 73)
c = 6;
if (input == 74)
c = 5;
if (input == 75)
c = 4;
if (input == 76)
c = 3;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (input == 77)
c = 2;
if (input == 78)
c = 1;
if (input == 79)
c = 0;
if (input == 80)
d = 10;
if (input == 81)
d = 9;
if (input == 82)
d = 8;
if (input == 83)
d = 7;
if (input == 84)
d = 6;
if (input == 85)
d = 5;
if (input == 86)
d = 4;
if (input == 87)
d = 3;
if (input == 88)
d = 2;
if (input == 89)
d = 1;
if (input == 90)
d = 0;
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน



ชื่อ นายณัฐนนท์ เหล่าสมบัติ , ชื่อเล่น นัต  
 วันเดือนปีเกิด 30 สิงหาคม 2528  
 ที่อยู่ 398 ถ.เจริญเมือง ต.โนเวียง อ. เมือง จ.แพร่  
 54000  
 E-mail phoenix\_nad@hotmail.com  
 การศึกษา :มัธยมศึกษาตอนปลาย พิริยาลัย จังหวัดแพร่  
 :ปริญญาตรีสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ  
 ทหารลาดกระบัง



ชื่อ นายณัฐพงศ์ อุดมประเสริฐ , ชื่อเล่น เพชร  
 วันเดือนปีเกิด 4 พฤศจิกายน 2528  
 ที่อยู่ 697/5 ค. ถ. สุรัชย์ ต. มะขามหย่ง อ. เมือง จ.  
 ชลบุรี 20000  
 E-mail nutt\_petch@hotmail.com  
 การศึกษา :มัธยมศึกษาตอนปลาย จุฬารัตน์ราชวิทยาลัย  
 ชลบุรี  
 :ปริญญาตรี สถาบันเทคโนโลยีพระจอมเกล้าเจ้า  
 คุณทหารลาดกระบัง



ชื่อ นางสาวทิพย์วิมล กริยาอาภรณ์ , ชื่อเล่น ปลา  
 วันเดือนปีเกิด 6 พฤศจิกายน 2528  
 ที่อยู่ 4/158 ม.15 ต.วัดไทร อ.เมืองนครสวรรค์ จ.  
 นครสวรรค์ 60000  
 E-mail blackbox\_plalek@hotmail.com  
 การศึกษา :มัธยมศึกษาตอนปลาย นวมินทรราชูทิศ มัชฌิม  
 นครสวรรค์  
 :ปริญญาตรี สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณ  
 ทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้