ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

ระบบควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต

ELECTRICAL EQUIPMENT VIA INTERNET

H006597

ศมาพงศ์ มกรารัตต์

ทศพล รอดความทุกข์

เลขหมู่.....
เลขทะเบียน.....06597
วัน, เดือน, ปี.....27 ก.พ. 2555

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

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต

ELECTRICAL EQUIPMENT VIA INTERNET

โดย

ศมาพงศ์ มกรารัตต์

ทศพล รอดความทุกข์

อาจารย์ที่ปรึกษา

ดร.ปานวิทย์ ธุวะนุติ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาคเรียนที่ 2 ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL EQUIPMENT VIA INTERNET

SAMAPONG MOKKARARAT

THOSAPHOL RODKWAMTUK

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF SCIENCE PROGRAM IN INFORMATION TECHNOLOGY

FACULTY OF INFORMATION TECNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2/2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใบรับรองปริญญานิพนธ์ ประจำปีการศึกษา 2553

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต

ELECTRICAL EQUIPMENT VIA INTERNET

ผู้จัดทำ

1. นายศมาพงศ์ มกรารัตต์ รหัสประจำตัว 50070059
2. นายทศพล รอดความทุกข์ รหัสประจำตัว 50070062

.....อาจารย์ที่ปรึกษา

(ดร.ปานวิทย์ ธุวะนุติ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2011

FACULTY OF INFORMATION TECHNOLOGY

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อโครงงาน	ระบบควบคุมอุปกรณ์ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต
นักศึกษา	นายศมาพงศ์ มกรารัตต์ รหัสนักศึกษา 50070059
	นายทศพล รอดความทุกข์ รหัสนักศึกษา 50070062
ระดับการศึกษา	วิทยาศาสตรบัณฑิต
สาขาวิชา	เทคโนโลยีสารสนเทศ
ปีการศึกษา	2553
อาจารย์ที่ปรึกษา	ดร. ปานวิทย์ ธุวะนุติ

บทคัดย่อ

บางครั้งการควบคุมและตัดการเครื่องใช้ไฟฟ้าภายในบ้าน ก็ทำได้ยาก เช่น อุปกรณ์อยู่ในที่ๆไม่สามารถเข้าถึง อุปกรณ์อยู่ห่างไกล หรืออยู่ในที่ๆอันตราย ซึ่งอาจจะทำให้เป็นสาเหตุของการเกิดอุบัติเหตุ หรืออันตรายต่างๆได้ เราจึงได้เสนอโครงงาน การสั่งการอุปกรณ์ไฟฟ้าภายในบ้านระยะไกลผ่านอินเตอร์เน็ต โดยใช้ไมโครคอนโทรเลอร์เป็นตัวควบคุมอุปกรณ์ไฟฟ้า และรับคำสั่งการควบคุมอุปกรณ์ไฟฟ้าจากคอมพิวเตอร์ ผ่านทางอินเตอร์เน็ต จึงทำให้สามารถทำให้สั่งการอุปกรณ์ไฟฟ้าในบ้านที่ไหนก็ได้ แม้ว่าจะอยู่ไกลจากอุปกรณ์นั้นก็ตาม ในวิทยานิพนธ์ฉบับนี้ เราจะได้อธิบายในรายละเอียดเกี่ยวกับฮาร์ดแวร์และซอฟต์แวร์ที่ต้องใช้เพื่อการพัฒนาหลักการและอุปกรณ์เหล่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Title	Electrical equipment via internet
Student	Mr. Samapong Mokkararat Student ID 50070059
	Mr. Thosaphol Rodkwamtuk Student ID 50070062
Degree	Bachelor's degree of science
Program	Information Technology
Year	2010
Advisor	Dr.Panwit Tuwanut

ABSTRACT

Occasionally, control and manage home appliances is difficult in such devices cannot access the other, remote device or located in any danger. This may be the cause of the accident and dangers. We proposed thesis, Electrical equipment via internet. We using microcontroller to control electrical, Command and control electrical devices from your computer via the Internet. We can control the electrical equipment in the house anywhere, although far from any device. In this thesis, we explain in detail about the hardware and software required to develop these principles and equipment.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ได้รับความช่วยเหลือและการให้คำปรึกษาจากอาจารย์ที่ปรึกษา ดร. ปานวิทย์ ธุวะนุติ ที่ให้คำแนะนำในทุกขั้นตอนที่ได้ทำการศึกษาปริญญานิพนธ์ รวมทั้งการตรวจสอบ แก้ไขข้อบกพร่องต่างๆ ของปริญญานิพนธ์ฉบับ และช่วยอำนวยเหลือทางด้านวัสดุอุปกรณ์ รวมถึงห้องที่ใช้ในการจัดทำปริญญานิพนธ์นี้ด้วย หากขาดท่านอาจารย์ไป คงเป็นไปได้ยากที่ปริญญานิพนธ์นี้จะเสร็จสมบูรณ์ได้ ขอขอบพระคุณท่านอาจารย์ที่ให้ความช่วยเหลือเสมอมา

ขอบพระคุณคณาจารย์ประจำคณะเทคโนโลยีสารสนเทศ ณ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่คอยให้ความรู้ และให้คำแนะนำต่างๆ ทั้งในด้านวิชาการหรือในด้านการทำงาน ซึ่งสิ่งเหล่านี้ทำให้เกิดปริญญานิพนธ์ ที่เกิดจากการอบรม สั่งสอนจากทุกท่าน

ขอบคุณเพื่อนๆ และพี่น้อง คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่คอยช่วยเหลือและให้กำลังใจเสมอมา ไม่ว่าด้านใดก็ตาม

สุดท้ายคุณค่าอันจะเป็นประโยชน์ คุณงามความดีทั้งปวงที่เกิดขึ้นจากปริญญานิพนธ์นี้ ขอมอบแด่บิดา มารดา และครอบครัวที่รักยิ่ง หากมีข้อบกพร่องด้วยประการใดๆ ผู้จัดทำขอน้อมรับไว้ด้วยความขอบคุณยิ่ง

ศมาพงศ์ มกรารัตต์

ทศพล รอดความทุกข์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ I

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญรูป VI

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตของงาน 2

1.4 ขั้นตอนการพัฒนาระบบ 3

1.5 เครื่องมือที่ใช้ในการพัฒนาระบบ 4

1.6 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 5

2.1 MCS51(AT89C51) 5

2.2 เอเอสพีดอทเน็ต (ASP.NET) 11

2.3 ซีชาร์ป (C#) 13

2.4 เอแจ็กซ์ (AJAX) 14

2.5 MAX232 16

2.6 ZIGBEE 17

2.7 เครือข่ายแบบ CLIENT-SERVER 20

2.8 อินฟราเรด (INFRARED) 21

บทที่ 3 การวิเคราะห์และออกแบบ 23

3.1 หลักการทำงาน และ การออกแบบในภาพรวม 23

3.2. ตัวอย่างส่วนติดต่อกับผู้ใช้ (GUI) ของหน้าเว็บ 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

บทที่ 4 ผลการศึกษาและดำเนินงาน 27

4.1 หน้าจอการเข้าสู่ระบบ 27
4.2 หน้าจอแสดงสถานะของอุปกรณ์ 27
4.3 หน้าจอการแก้ไขรายละเอียดอุปกรณ์ 27
4.4 หน้าจอการเพิ่มอุปกรณ์ 28
4.5 ส่วนการทำงานและควบคุมระหว่างคอมพิวเตอร์ กับอุปกรณ์ไฟฟ้า 29

บทที่ 5 สรุปและข้อเสนอแนะ 37

บรรณานุกรม 39
ภาคผนวก 40
ภาคผนวก ก. 41
ภาคผนวก ข 47
ภาคผนวก ค 51
ประวัติผู้เขียน 61

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.2 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโครลเลอร์ MCS-51 7
รูปที่ 2.3 โครงสร้างของ เอแจ๊กซ์ (AJAX) 15
รูปที่ 2.4 ตำแหน่งขาสัญญาณและวงจรภายในไอซีปรับระดับแรงดัน MAX 232 16
รูปที่ 2.5 แสดงโครงสร้างในรูป ZIGBEE STACK 18
รูปที่ 2.6 แสดงโครงสร้าง ส่วนประกอบ และรูปแบบของ ZIGBEE 19
รูปที่ 3.1 ภาพโครงสร้างการทำงานอย่างง่าย 23
รูปที่ 3.2 ภาพวงจรระหว่าง MAX232 และ ZIGBEE 24
รูปที่ 3.3 ภาพวงจรระหว่างโปรโตคอล ZIGBEE , MCS-51 และ RELAY ในเบื้องต้น 24
รูปที่ 3.4 แสดงหน้าเข้าสู่ระบบ 25
รูปที่ 3.5 แสดงหน้าแรกของอุปกรณ์ 25
รูปที่ 3.6 แสดงหน้าเพิ่มอุปกรณ์ 26
รูปที่ 3.7 แสดงหน้าแก้ไขอุปกรณ์ 26
รูปที่ 4.1 แสดงหน้าจอสำหรับการเข้าใช้งานระบบ 27
รูปที่ 4.2 แสดงหน้าจอสถานะของอุปกรณ์ 27
รูปที่ 4.3 แสดงหน้าจอการแก้ไขรายละเอียดอุปกรณ์ 28
รูปที่ 4.4 แสดงหน้าจอการเพิ่มอุปกรณ์ 28
รูปที่ 4.5 แสดงโปรแกรมที่ใช้ในการส่งและรับค่าระหว่างเครื่องเซิร์ฟเวอร์กับ ไมโครคอนโทรลเลอร์ 29
รูปที่ 4.6 ส่วนตัวส่งสัญญาณจากคอมพิวเตอร์ 30
รูปที่ 4.7 ส่วนรับสัญญาณจากคอมพิวเตอร์ 30
รูปที่ 4.8 รูปของส่วนรับสัญญาณโดยเริ่มต้นที่ปิดไฟทั้งหมด 31
รูปที่ 4.9 ผลการทดลองจากการสั่งการโดยเว็ป โดยเปิดไฟทั้ง 2 เครื่อง 32
รูปที่ 4.10 ผลการทดลองจากการสั่งโดยเว็ป โดยเปิดไฟเพียงเครื่องเดียว 32
รูปที่ 4.11 รูปตัวรับสัญญาณอินฟราเรด 33
รูปที่ 4.12 รูปตัวส่งสัญญาณอินฟราเรด 34
รูปที่ 4.13 รูปผลทดลองเมื่อนำวัตถุมาวางขวางสัญญาณอินฟราเรด 34
รูปที่ 4.14 รูปของปุ่มที่ใช้เปิด-ปิดไฟฟ้าโดยตรง 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

หน้า

รูปที่ 4.15 รูปของอุปกรณ์หลังจากถูกสั่งจากคอมพิวเตอร์ให้ปิดการทำงาน 36
รูปที่ 4.16 รูปของผลจากการกดปุ่มสั่งการอุปกรณ์โดยตรง โดยเปิดอุปกรณ์ 36
รูปที่ ก.1 ที่ตั้งของส่วน USB SERIAL PORT 42
รูปที่ ก.2 ภาพในส่วนของ PORT SETTING 43
รูปที่ ก.3 รูปภาพใน ADVANCED 43
รูปที่ ก.4 ภาพหลังจากเรียก SXA51.EXE เพื่อสร้าง TEST01.ASM ให้เป็น TEST01.HEX 44
รูปที่ ก.5 ภาพของ ET-AFP V 1.0 ที่เชื่อมต่อกับ USB TO RS232 44
รูปที่ ก.6 ภาพหลังจากเลือก COMPORT และ CPU SEARCH แล้ว จะได้รุ่นของ IC ออกมา 45
รูปที่ ก.7 ภาพการเลือก HEX FILE 45
รูปที่ ก.8 ภาพหลังจาก AUTO แล้ว ถ้าสำเร็วจะขึ้นว่า PASS 46
รูปที่ ข.1 รูปภาพส่วน PC SETTINGS 48
รูปที่ ข.2 แสดงภาพเมื่อ XBEE สามารถเชื่อมต่อกับคอมพิวเตอร์ได้สำเร็จหลังจาก TEST/QUERY 48
รูปที่ ข.3 ภาพในส่วนของ MODEM CONFIGURATION 50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ทุกครัวเรือน หรือองค์กร ย่อมต้องมีสิ่งอำนวยความสะดวกไว้ใช้งาน ส่วนใหญ่จะเป็นพวกอุปกรณ์อิเล็คทรอนิคต่างๆ เช่น ฟัดลม หลอดไฟ โทรทัศน์ ไปจนกระทั่ง เครื่องจักรที่ใช้ในการผลิตขนาดใหญ่ ที่มีผลกระทบต่อคนส่วนมาก ซึ่งจะมีผลต่อการใช้ชีวิตประจำวัน หรือการประกอบอาชีพทั้งทางตรง และทางอ้อม จึงจำเป็นต้องมีการจัดการ และควมคุมให้ดี มิฉะนั้น อาจจะเกิดปัญหา ที่สร้างความสียหายอย่างใหญ่หลวงตามมาก็ได้ ทว่าการจัดการและป้องกัน ระบบเครื่องใช้ไฟฟ้าต่างๆ จะมีความสำคัญ แต่ก็ไม่ใช่เรื่องสามารถจัดการได้ง่ายๆ โดยเฉพาะองค์กรที่มีขนาดใหญ่ มีครเองมือมาก และมีผลต่อธุรกิจนั้น ยิ่งเป็นเรื่องยากขึ้นไปอีก อีกทั้งส่วนหนึ่ง เกิดมาจากการที่ตัวบุคคล ไม่สามารถเข้าถึง หรือความคุมอุปกรณ์ได้โดยตรงได้ยาก ได้แก่ ตัวผู้ชำนาญการหรือช่างซ่อมบำรุง ไม่อยู่ หรืออยู่ห่างไกลจากเครื่องที่มีปัญหา หรือแม้การทั้ง เครื่องมือที่อยู่ในสถานที่ ที่อันตรายจนเกินไป

เนื่องมาจากสาเหตุข้างต้นนี้เอง จึงจำเป็นต้องมีการริเริ่ม พัฒนา ระบบการจัดการควบคุมเครื่องใช้ไฟฟฟ้า เพื่อให้สามารถจัดการป้องกัน และแก้ไขปัญหาได้ทันท่วงที โดยการนำเทคโนโลยีในด้านไมโครคอนโทรเลอร์ (Microcontroller) ซึ่งสามารถนำมาประยุกต์ได้มากมาย และหลากหลาย ได้แก่ เซ็นเซอร์ต่างๆ สวิตซ์ และอื่นๆอีกมากมาย และทางด้านเน็ตเวิร์ค (Network) มาประยุกต์ ซึ่งจะทำให้สามารถควมคุม และจัดการได้ในระยะไกล และทันท่วงที โดยจะนำมาประยุกต์กับอินเตอร์เน็ต (Internet) ที่เป็นเครือข่ายที่มีทั่วโลก ทำให้สามารถใช้งานที่ไหนก็ได้ โดยผ่านทางเว็บ (Web- Base Application) โดยใช้เอเอสพีดอทเน็ต(ASP.NET)โดยเขียนด้วย ซีชาร์ฟ (C#) เพราะเป็นภาษาที่เข้าใจได้ง่าย และมีความรวดเร็ว โดยจะสามารถสั่งการให้เครื่องใช้ไฟฟ้าทำงานได้ตามต้องการ ในเบื้องต้นผ่านอินเตอร์เน็ต ได้แก่ การเปิด ปิดไฟภายในบ้าน การบอกสถานะต่างๆของเครื่องใช้ไฟฟ้าในบ้าน เข้ามาทางอินเตอร์เน็ต ทำให้สามารถจัดการได้ดียิ่งขึ้น ซึ่งระบบนี้ จะเป็นทางหน่งที่จะช่วยป้องกัน และลดความเสี่ยงเกี่ยวกับเครื่องใช้ไฟฟ้าที่อาจจะเกิดปัญหาได้ตลอดเวลา หรืออาจจะต้องใช้งานในกรณีจำเป็นได้ทันท่วงที

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาความรู้ด้านฮาร์ดแวร์ (Hardware) และไมโครคอร์นโทรเลอร์ (Microcontroller) ทั้งในด้านพื้นฐาน และการประยุกต์ใช้จริง กับเครื่องใช้ไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยสามารถทำให้เครื่องใช้ไฟฟ้าที่ได้ติดตั้งตัวฮาร์ดแวร์(Hardware) หรือไมโครคอนโทรเลอร์ (Microcontroller) นี้แล้ว สามารถควบคุม หรือจัดการเครื่องใช้ไฟฟ้าเหล่านี้ได้

1.2.2 เพื่อศึกษาความรู้ทางด้านระบบเครือข่าย (Network) ในด้านต่างๆ เช่น ความรู้พื้นฐานของอินเตอร์เน็ต (Internet) พื้นฐานด้านโปรโตคอล (Protocal) ที่ต้องใช้ด้านการจัดการสัญญาณต่างๆ ข้อจำกัดต่างๆ เพื่อเป็นตัวเชื่อมโยง ระหว่างอุปกรณ์ไฟฟ้า และคอมพิวเตอร์ได้

1.2.3 สามารถผนวกรวม และประยุกต์ให้ฮาร์ดแวร์ และเน็ตเวิร์ค ทำงานร่วมกันได้อย่างมีประสิทธิภาพ

1.2.4 เพื่อศึกษาถึงหลักการและทฤษฎีต่างๆที่เกี่ยวข้องได้แก่

1.2.4.1 ทฤษฎีและหลักการในการทำงานของไฟฟ้าทั่วไป

1.2.4.2 ทฤษฎีและหลักการของเว็บ (Web-Base Application) คือเอเอสพีดอทเน็ต (ASP.NET) ด้วยภาษาซีชาร์ป (C#) เพื่อใช้ในการพัฒนาให้สามารถใช้งานบนเว็ปได้

1.3 ขอบเขตของงาน

โครงการนี้เป็นการศึกษาการนำเอาความรู้ทางด้านไมโครคอนโทรลเลอร์ (Microcontroller) และด้านเน็ตเวิร์ค (Network) มาประยุกต์ และพัฒนาเครื่องใช้ไฟฟ้า ให้สามารถควบคุม จัดการได้ในระยะไกล ผ่านระบบเว็บ (Web-Base Application) บนอินเตอร์เน็ต (Internet) เพื่อป้องกัน ดูแลและสั่งการเครื่องใช้ไฟฟ้า ให้เป็นไปด้วยความเรียบร้อย

1.3.1 นายทศพล รอดความทุกข์ รหัสนักศึกษา 50070062

- วิเคราะห์ ออกแบบ และพัฒนาเว็บ (Web-Base Application) ในการติดต่อสื่อสารระหว่างเครื่อง, ฮาร์ดแวร์และผู้ใช้งาน
- ศึกษาหลักการและทฤษฎีของ เน็ตเวิร์คและอินเตอร์เน็ต ทั้งในด้านพื้นฐานและการประยุกต์จริง
- ศึกษาหลักการและทฤษฎีของ ฮาร์ดแวร์และไมโครคอนโทรเลอร์ ทั้งในด้านพื้นฐาน และการประยุกต์ใช้จริง
- ทำการทดสอบและวิเคราะห์ผลที่ได้ออกมา ตรงตามวัตถุประสงค์ หรือไม่

1.3.2 นายศมาพงศ์ มกรารัตต์ รหัสนักศึกษา 50070059

- การวิเคราะห์ ออกแบบ และพัฒนา ในส่วนของฮาร์ดแวร์และไมโครคอนโทรเลอร์ ที่จะต้องประยุกต์ใช้กับเครื่องใช้ไฟฟ้าจริง และประยุกต์กับเว็บ (Web-Base Application)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ศึกษาหลักการและทฤษฎีของ เน็ตเวิร์คและอินเตอร์เน็ต ทั้งในด้านพื้นฐานและการประยุกต์จริง
- ศึกษาหลักการและทฤษฎีของ ฮาร์ดแวร์และไมโครคอนโทรเลอร์ ทั้งในด้านพื้นฐาน และการประยุกต์ใช้จริง
- ทำการทดสอบและวิเคราะห์ผลที่ได้ออกมา ตรงตามวัตถุประสงค์ หรือไม่

1.4 ขั้นตอนการพัฒนาระบบ

เริ่มจากศึกษาการหลักทำงานของไอซี MCS-51 และอุปกรณ์ทุกอย่างที่เกี่ยวข้อง วิธีการใช้งาน ข้อควรระวัง เบื้องต้น จากนั้นจึงเริ่มออกแบบวงจร โดยให้ไมโครอนโทรเลอร์สามารถเชื่อมต่อระหว่างคอมพิวเตอร์และไมโครคอนโทรเลอร์ เชื่อมต่อระหว่างเครื่องใช้ไฟฟ้าได้ จากนั้นก็เริ่มพัฒนาให้สามารถรับ ส่งสัญญาณระยะไกลจากอินเตอร์เน็ต ผ่านเว็บ (Web-Base Application) ได้ โดยช่วงนี้จะเริ่มสร้างส่วนเว็บเพจ (Webpage) ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์และอุปกรณ์ไฟฟ้า โดยใช้เว็บ (Web-Base Application) คือเอเอสพีดอทเน็ต(ASP.NET) ด้วยภาษาซีชาร์ป (C#) โดยมีขั้นตอนการดำเนินงานดังนี้

1.4.1 การเตรียมการเบื้องต้น

1.4.1.1 ศึกษาเอกสารและระบบงานที่เกี่ยวข้องกับไอซี MCS-51 จากหนังสือ วารสารอินเตอร์เน็ตและเอกสารงานวิจัยต่าง ๆ

1.4.1.2 ศึกษาเอกสารและระบบงานที่เกี่ยวข้องกับ การสั่งการระยะไกลผ่านอินเตอร์ จากหนังสือ วารสาร อินเตอร์เน็ตและผู้เชี่ยวชาญ

1.4.1.3 ศึกษาซอฟต์แวร์และฮาร์ดแวร์ที่ใช้พัฒนา ว่ามีความเหมาะสมกับระบบที่จะสร้างหรือไม่

1.4.1.4 การจัดหาอุปกรณ์ทั้งซอฟต์แวร์ และฮาร์ดแวร์ที่จะมาใช้ในการพัฒนา

1.4.2 การเก็บรวบรวมข้อมูล โดยข้อมูลที่จะใช้ในการออกแบบวงจร และโปรแกรม

1.4.3 การวิเคราะห์และออกแบบระบบ

1.4.4 สร้างส่วนข้อเท็จจริงและกฎที่เกี่ยวข้อง

1.4.5 ทำการพัฒนาระบบเพื่อดูการทำงานและผลกระทบที่เกิดขึ้น

1.4.6 ปรับปรุงรูปแบบ ขั้นตอนการค้นหาคำตอบและส่วนติดต่อผู้ใช้ให้สามารถใช้งานได้คล่องตัวมากขึ้น

1.4.7 ประเมินผลการทำงาน

1.4.8 สรุปผลและข้อเสนอแนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 เครื่องมือที่ใช้ในการพัฒนาระบบ

1.5.1 ฮาร์ดแวร์

1.5.1.1. ไอซี AT89C51 ซึ่งอยู่ในตระกูลของ MCS-51

1.5.1.2. ไอซี MAX232

1.5.1.3. ไอซี 74HC541

1.5.1.4. XBee PRO

1.5.1.5. รีเลย์ (Relay)

1.5.1.6. ทรานซิสเตอร์ (Transistor)

1.5.1.7. อุปกรณ์ตรวจจับเซ็นเซอร์

1.5.1.8. คอมพิวเตอร์ทั่วไป ที่สามารถเชื่อมต่อกับอินเตอร์เน็ตได้

1.5.1.9. เครื่องใช้ไฟฟ้าทั่วไป

1.5.2 ซอฟต์แวร์

1.5.2.1. AFP34 เป็นโปรแกรมสำหรับการบันทึกโปรแกรม ลงสู่ AT89C51

1.5.2.2. SXA51 เป็นโปรแกรมสำหรับเรียบเรียงแปลโปรแกรม (Compiler) ภาษาเอซเซมบีล

1.5.2.3. NetBean6.5 ขึ้นไป, EditPlus, JDK 6 update6

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1.6.1 สามารถจัดการกับเครื่องใช้ไฟฟ้าได้ในระยะไกล ผ่านเว็ป (Web-Base Application) ได้

1.6.2 มีความรู้ ความเข้าใจในด้านฮาร์ดแวร์และไมโครคอนโทรเลอร์ ทั้งในด้านพื้นฐานและการประยุกต์ใช้จริง

1.6.3 มีความรู้ ความเข้าใจในด้านอินเตอร์เน็ตและเน็ตเวิร์ค ทั้งในด้านพื้นฐาน และการประยุกต์ใช้จริง

1.6.4 มีความรู้ ความเข้าใจในการสร้างเว็บ (Web-Base Application) ทั้งในด้านพื้นฐานและการประยุกต์ใช้จริง

1.6.5 สามารถเป็นต้นแบบ หรือพัฒนาในระบบการควบคุมเครื่องใช้ไฟฟ้าอื่นๆ ที่มีความซับซ้อน หรือขนาดใหญ่มากขึ้นต่อไปได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 MCS51(AT89C51)

ไอซี MCS51 เป็นไมโครคอนโทรเลอร์ ที่ผลิตโดย INTEL ซึ่งเป็น บริษัทชั้นนำจึงทำให้ไอซี MCS51 ค่อนข้างเป็นมาตรฐานและได้รับความนิยมในปัจจุบัน อีกทั้งมีบริษัทอื่นที่ผลิตไอซี (IC) โดยใช้มาตรฐานของ MCS51 เช่น ATMEL ทำให้ผู้ใช้สามารถเลือกหามาใช้งานได้ง่าย และ MCS51 ก็เป็นไมโครคอนโทรเลอร์ ที่ใช้งานได้ง่ายและตอบสนองงาน ด้านระบบควบคุมขนาดเล็กได้ดี

คุณสมบัติทั่วไปของ MCS51 (AT89C51)

- ทำงานที่ความถี่ 1 - 24 MHz Operation
- 3-Level Program Memory Lock (ป้องกันการ copy program)
- Boolean Processor
- Internal Program 4 KB (AT89C51)
- 128-Byte Data RAM
- 32 Programmable I/O Lines (P0 - P3)
- Two 16-Bit Timer/Counters
- 1 Serial Port
- Extended Temperature Range (-40 'C to +85 'C)
- 5 Interrupt Sources
- TTL- and CMOS-Compatible Logic Levels
- 64K External Program Memory Space
- 64K External Data Memory Space
- Power Control Modes [-Idle , -Power Down]

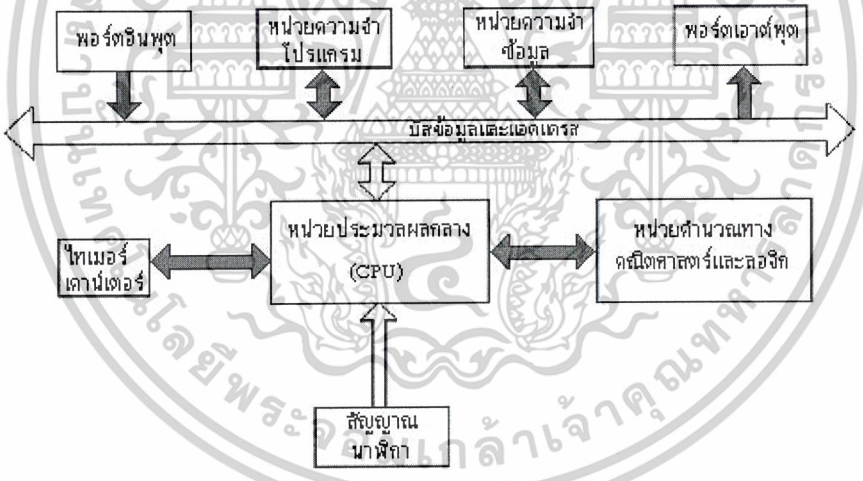
2.1.1 โครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 แสดงในรูปที่ 2.1 ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต (BOOLEAN PROCRSSOR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบ อีพรอม (เบอร์ 8451)
- หน่วยความจำแบบ แรม ภายในจำนวน 128 ไบต์
- พอร์ตอินพุต/เอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถแยกทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรมแบบดูเพล็กเต็ม(FULL DUPLEX)
- วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภท พร้อมการกำหนด ลำดับ
- วงจรผลิตสัญญาณนาฬิกาภายในซึ่งโครงสร้างการทำงานทั้งหมดของไมโครคอนโทรลเลอร์จะอาศัยหลักการทำงานที่เกี่ยวข้องกัน โดยอาศัยหลักการทำงานที่เป็นไป ตามโครงสร้างเสมอ

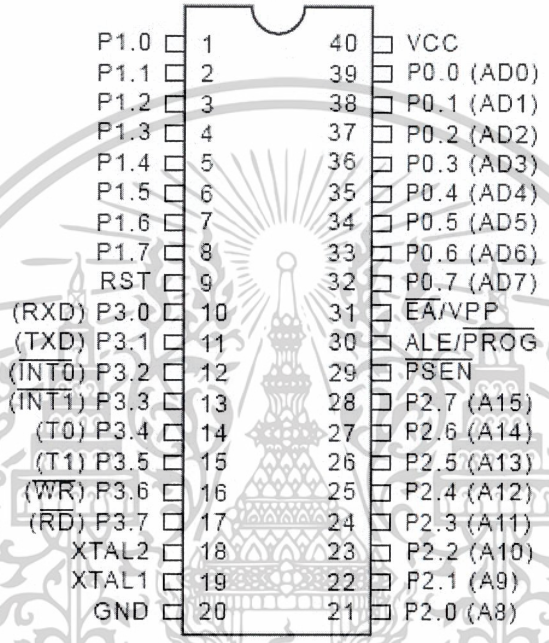


รูปที่ 2.1 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้มักจะมีรูปร่างของไอซีเป็นแบบขนาด 40 ขา ดังแสดงในรูปที่ 1 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อ ที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า ALTERNATE FUNCTION ในรูปที่ 2) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ เช่นขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อเป็น P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุต หรืออินพุตตามปกติ ภายในไมโครคอนโทรลเลอร์ MCS-51 ซึ่งประกอบด้วยหน่วยการทำงานต่างๆ ภายในไอซีMCS-51 จำนวนมาก โดยแต่ละบล็อกซึ่งเป็นวงจรควบคุมรีจิสเตอร์ (REGISTER) หรือหน่วยความจำภายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของไอซี MCS-51 จะถูกเชื่อมต่อเข้าด้วยกันผ่านทางเส้นสัญญาณที่เรียกว่าบัสข้อมูลภายใน รีจิสเตอร์และหน่วยความจำเหล่านี้จะถูกนำไปใช้ระหว่างการประมวลผลคำสั่ง หน้าที่ของโปรแกรมที่ผู้ใช้สร้างขึ้นมาก็เป็นการควบคุมการรับหรือส่งข้อมูล ระหว่างรีจิสเตอร์เหล่านี้ ซึ่งอาจจะมีการดำเนินการร่วมกับหน่วยการดำเนินงานประมวลผลทางคณิตศาสตร์และ ลอจิก หรือเรียกว่า ARITHMATIC AND LOGIC UNIT :ALU



รูปที่ 2.2 แสดงรูปร่างและการจัดวางขาต่างๆ ของไมโครคอนโครลเลอร์ MCS-51

2.1.2 โครงสร้างหน่วยความจำภายในไมโครคอนโครลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 แยกการจัดการหน่วยความจำออกเป็นสองส่วนอย่างชัดเจน คือ หน่วยความจำโปรแกรม (PROGRAM MEMORY) และหน่วยความจำข้อมูล (DATA MEMORY) หน่วยความจำทั้งสองนี้ มีหน้าที่แตกต่างไปจากกัน และใช้วิธีการอ้างแอดเดรสสัญญาณการติดต่อแยกออกจากกัน

2.1.2.1 หน่วยความจำโปรแกรม (PROGRAM MEMORY)

หน่วยความจำโปรแกรมของ MCS-51 เป็นบริเวณหน่วยความจำ สำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้กับระบบข้อมูลเหล่านี้ก็ยังคงอยู่ไม่สูญหายในไมโครคอนโทรลเลอร์ MCS-51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมนี้ได้สูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะ ตามตำแหน่งของหน่วยความจำนั้น คือ

- หน่วยความจำโปรแกรมภายใน (INTERNAL PROGRAM MEMORY) ซึ่งเป็นหน่วยความจำรอม หรือ อีพรอม ที่อยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง
- หน่วยความจำโปรแกรมภายนอก (EXTERNAL PROGRAM MEMORY) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

2.1.2.2 หน่วยความจำข้อมูล (DATA MEMORY)

หน่วยความจำข้อมูล (DATA MEMORY) ซึ่งโดยพื้นฐานแล้วเป็นหน่วยความจำแรมสามารถเขียนหรืออ่านข้อมูลได้ (READ OR WRITE MEMORY) ใช้สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว ซึ่งโดยพื้นฐานแล้วหน่วยความจำข้อมูลจัดเป็นหน่วยความจำแรมแบบสแตติกดังนั้น เมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ภายใน หน่วยความจำนี้สูญไป พื้นที่ของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์ MCS-51 มีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น ตามลักษณะของหน่วยความจำโปรแกรมภายในซึ่งก็เป็นแรมที่อยู่ภายในตัวไอซีใน ตระกูลของไมโครคอนโทรลเลอร์ และหน่วยความจำข้อมูลภายนอกซึ่งเป็นการใช้ไอซีหน่วยความจำแรมมาเพิ่มเติม เข้าไปในวงจรลักษณะเดียวกับ การนำไอซีอีพรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

2.1.3 รีจิสเตอร์ที่เกี่ยวกับไมโครคอนโทรลเลอร์ MCS-51

รีจิสเตอร์ในกลุ่มนี้จะเป็นรีจิสเตอร์ขนาด 16บิตที่ใช้งานเพื่อเก็บข้อมูลของตัวแอดเดรสเป็นสำคัญโดยค่าที่อยู่ภายในแอด เดรสนี้จะนำไปเป็นค่าของข้อมูลที่ส่งออกไปทางบัสแอดเดรส ในส่วนของไมโครคอนโทรลเลอร์ เพื่อบอกตำแหน่งที่ต้องการติดต่อ รีจิสเตอร์ที่จัดในกลุ่มนี้ประกอบด้วย

2.1.3.1 รีจิสเตอร์ใช้งานทั่วไป (GENERAL-PURPUSE REGISTERS) รีจิสเตอร์ในกลุ่มนี้จัดเป็นพื้นที่หน่วยความจำที่ใช้ในการสนับสนุนในการประมวลผล การทำงานจากหน่วยประมวลผลทางคณิตศาสตร์และลอจิก (ALU) เพื่อให้สามารถจัดการข้อมูลให้เร็วที่สุด นอกจากนี้โปรแกรมที่ไม่ได้ใช้คำสั่งเหล่านี้ก็ยังใช้เป็นการเก็บข้อมูลตัว แปรภายในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3.2 รีจิสเตอร์หน้าที่พิเศษ เป็นรีจิสเตอร์หน้าที่พิเศษ (SFR) เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่ และการทำงานของอุปกรณ์หรือพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 ทั้งหมด ตำแหน่งของรีจิสเตอร์เหล่านี้จะจัดอยู่ในบริเวณแอดเดรส 80H - FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรส ที่เป็นของรีจิสเตอร์นั้นก็ได้ การจัดพื้นที่หน่วยความจำสำหรับรีจิสเตอร์หน้าที่พิเศษเหล่านี้ โดยมีข้อสังเกตว่ารีจิสเตอร์ที่อยู่ในตำแหน่งแอดเดรสที่มีจำนวนเป็นทวีคูณ ของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H 88H 90H A0H A8H B0H B8H D0H E0H และ F0H)

2.1.3.3 แอกคิวมูเลเตอร์ (ACCUMULATOR) หรือ ACC เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในหน่วยประมวลผลกลาง และเก็บผลลัพธ์ที่ได้จากการทำงานเท่านั้น การทำงานของรีจิสเตอร์นั้มีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซ ส-เซอร์ทั่วไป การใช้งานในโปรแกรมซึ่งใช้เรียกเป็น รีจิสเตอร์ A

2.1.4 ชุดคำสั่งของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ประกอบด้วยคำสั่งทั้งหมดจำนวนมาก ซึ่งสามารถจะจัดกลุ่มคำสั่งเหล่านี้ตามลักษณะและหน้าที่การทำงานที่คล้าย คลึงกัน เพื่อความสะดวกต่อการศึกษาทำความเข้าใจและใช้งาน ดังนี้

2.1.4.1 กลุ่มการถ่ายเทข้อมูล คือ กลุ่มคำสั่งในการโอนย้ายข้อมูล ทำหน้าที่ในโอนย้ายข้อมูลระหว่างรีจิสเตอร์ หรือหน่วยความจำภายในแรม โดยมีรายละเอียดดังนี้

- MOV: จะทำงานในลักษณะเป็นการถ่ายเทข้อมูลที่มีขนาดเป็น ไบต์หรือบิตก็ได้จากแหล่งกำเนิดเข้าสู่ตัวรับข้อมูลในฟิลด์โอเปอร์แรนด์

- PUSH: จะทำงานโดยเพิ่มค่ารีจิสเตอร์ SP ก่อนแล้วจึงทำการถ่ายเทข้อมูล 1 ไบต์จากแหล่งกำเนิดไปบริเวณสแต็กตามตำแหน่งที่รีจิสเตอร์ SP กำหนด

- POP: การถ่ายเทข้อมูลขนาด 1 ไบต์จากบริเวณตำแหน่งที่รีจิสเตอร์ SP กำหนดไปยังรีจิสเตอร์ที่โอเปอร์แรนด์ กำหนดและหลังจากนั้นรีจิสเตอร์ SP จะลดค่าลง

- XCH: คำสั่งแลกเปลี่ยนไบต์ระหว่างแหล่งกำเนิดโอเปอร์แรนด์กับรีจิสเตอร์ AXCHD คำสั่งในการแลกเปลี่ยนขนาดนิบเบิลทางอันดับต่ำของแหล่งกำเนิดโอเปอร์แรนด์กับนิบเบิลอันดับต่ำลงของแอกคิวมูเลเตอร์

2.1.4.2 กลุ่มคำสั่งทางคณิตศาสตร์ เช่น การบวก ลบ คูณ และหารข้อมูลภายในตัวรีจิสเตอร์ต่างๆ ช่วงเวลาการทำงาน ของแต่ละคำสั่งนั้นจะกำหนดที่ความถี่ของสัญญาณนาฬิกาที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมกะเฮิรตซ์ คำสั่งทางคณิตศาสตร์ส่วนใหญ่ใช้เวลา 1 ms ยกเว้นคำสั่ง INC DPTR ซึ่งใช้เวลา 2 ms โดยที่คำสั่งการคูณและหารใช้เวลา 4 ms โดยมีรายละเอียดการใช้คำสั่งมีดังนี้

- INC: เป็นการบวกหนึ่งกับโอเปอร์แรนด์และใส่ค่าใหม่กลับเข้าที่ตัวโอเปอร์แรนด์นั้นๆ

- DEC: เป็นการลบออกจากตัวเลขที่อยู่ในแหล่งกำเนิดโอเปอร์แรนด์ และนำผลลัพธ์ที่ได้มาเก็บไว้ที่ตัวโอเปอร์แรนด์นั้น

- ADD: เป็นการบวกในแอกคิวมูเลเตอร์เข้ากับค่าในแหล่งกำเนิดโอเปอร์แรนด์

- ADDC: เป็นการบวกค่าต่างๆ ในแอกคิวมูเลเตอร์เข้ากับค่าในแหล่งกำเนิดโอเปอร์-แรนด์และบวกกับบิตทดด้วย

- SUBB: เป็นการนำเลขที่แหล่งกำเนิดโอเปอร์แรนด์ ลบออกจากตัวเลขใน A และนำค่าบิตตัวทดมาลบออกอีกและผลลัพธ์ที่ได้นำมาใส่ลงในแอกคิวมูเลเตอร์ A

- MUL: เป็นการคูณแบบไม่คิดตัวเครื่องหมายของตัวเลขที่อยู่ใน แอกคิวมูเลเตอร์กับเลขใน รีจิสเตอร์ B แล้วได้ผลลัพธ์ 2 ไบต์ นำมาเก็บไว้ที่ AB โดย A จะรับอันดับต่ำส่วน B จะรับอันดับสูง

- DIV: เป็นคำสั่งในการหารแบบไม่คิดเครื่องหมายที่อยู่ในแอกคิวมูเลเตอร์แล้ว หารตัวเลขในรีจิสเตอร์ B แล้วนำผลลัพธ์ไปเก็บในแอกคิวมูเลเตอร์และเศษของการหารตัวเลข จะเก็บไว้ในรีจิสเตอร์ B

- DA: สำหรับการบวกกันทางตัวเลข BCD เป็นการปรับค่ารวม ซึ่งเป็นผลมาจากการบวกกันทางไบนารี่ของระบบตัวเลข BCD ขนาด 2 หลักสองจำนวน การปรับค่าตัวเลขผลรวมด้วยการใช้คำสั่ง DA จะได้ผลลัพธ์กลับมาที่แอกคิวมูเลเตอร์

2.1.4.3 กลุ่มคำสั่งทางตรรกศาสตร์หรือ แบบลอจิก ทำหน้าที่เกี่ยวกับการประมวลผลแบบลอจิก ต่างๆ เช่น การ AND OR หรือ EX-OR ระหว่างข้อมูลในรีจิสเตอร์ A นั่นเอง โดยมีการใช้คำสั่งดังนี้ CPL: เป็นการใช้คำสั่งกลับค่าหรือคอมพลีเมนต์ ข้อมูลในแอกคิวมูเลเตอร์จะไม่มีผลใดๆ ต่อค่าของแฟลก หรือการอ้างถึงตำแหน่งแอดเดรสนั้นตามบิตนั้นๆ

- RL, RLC, RR, RRC, SWAP: ทั้ง 5 คำสั่งนี้เป็นคำสั่งในการทำงานการวนบิตบนตัวของแอกคิวมูเลเตอร์ซึ่ง RL เป็นการวนบิตทางขวา, RLC เป็นการทำการวนทางซ้ายผ่านบิตทด, RRC เป็นการวนขวาผ่านบิตทด และ SWAP เป็นการวนซ้ายสี่ครั้ง

- ANL: เป็นการ ADD กันทางตรรกศาสตร์ ระหว่างแหล่งกำเนิดสองโอเปอร์แรนด์ ซึ่งจะสั่งให้ทำงานในรูปแบบของตรรกศาสตร์ทางข้อมูลขนาดเป็นไบต์หรือบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.4.4 กลุ่มคำสั่งแบบบูลีนหรือแบบบิต ซึ่งเป็นความสามารถของไมโครคอนโทรลเลอร์ MCS-51 ที่จะดำเนินการประมวลผลแบบบิต แทนที่จะเป็นข้อมูลทั้งไบต์เช่นปกติ โดยมีชุดคำสั่งที่จัดการโดยตรง ทุกคำสั่งจะเข้าถึงข้อมูลโดยตรงในระดับบิต โดยมีการบิตแอดเดรสได้ตั้งแต่ 00H - 7FH ในพื้นที่ 128 บิต หน่วยความจำข้อมูลภายในและบิตแอดเดรส 80H - FFH ในบริเวณกลุ่มรีจิสเตอร์ฟังก์ชั่นพิเศษ (SFR)

2.1.4.5 กลุ่มคำสั่งในการกระโดดไปยังตำแหน่งต่างๆภายในโปรแกรม ซึ่งจะเปลี่ยนลำดับของการประมวลผลภายในโปรแกรมไปยังส่วนต่างๆแทนที่จะดำเนิน การไปเป็นลำดับ ต่อเนื่อง โดยที่คำสั่ง JMP จะแบ่งเป็น 3 ลักษณะ คือ SJMP, LJMP, AJMP ซึ่งในแต่ละคำสั่ง จะมีข้อแตกต่างของการกระโดดไปยังแอดเดรสไกลสุดที่ต่างกัน คำสั่ง JMP ซึ่งเป็นแบบโมนีชิก ที่สามารถจะใช้ได้ โดยมีรายละเอียดการใช้งานของคำสั่งดังต่อไปนี้

- SMP: จะเป็นการกระโดดแบบการย้ายอันดับตำแหน่งของแอดเดรสตำแหน่งเดิมซึ่งจะสามารถกระโดดได้ - 128 ถึง +127 ไบต์
- AJMP: ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 2 กิโลไบต์ ซึ่งจะใช้หน่วยความจำเพียง 2 ไบต์เท่านั้นในการกำหนด
- LJMP: ลักษณะแบบนี้จะสามารถกระโดดได้ไกลสุดประมาณ 64 กิโลไบต์ ซึ่งจะใช้หน่วยความจำเพียง 3 ไบต์เท่านั้นในการกำหนด
- JMP @A+DPTR: เป็นการควบคุมการกระโดดไปยังโปรแกรมที่ต้องการเฉพาะภายในส่วนต่างๆ

2.2 เอเอสพีดอทเน็ต (ASP.NET)

เอเอสพีดอทเน็ต (ASP.NET) คือเทคโนโลยีสำหรับพัฒนาเว็บไซต์ เว็บแอปพลิเคชัน (Web Application) และเว็บเซอร์วิส (Web Service) ซึ่งเป็นส่วนหนึ่งของดอทเน็ตเฟรมเวิร์ก พัฒนาโดยไมโครซอฟท์ เอเอสพีดอทเน็ต (ASP.NET) เป็นรุ่นถัดจาก Active Server Pages (ASP) แม้ว่าเอเอสพีดอทเน็ต (ASP.NET) นั้นจะใช้ชื่อเดิมจากเอเอสพี (ASP) แต่ทั้งสองเทคโนโลยีนั้นแตกต่างกันอย่างสิ้นเชิง โดยไมโครซอฟท์นั้นได้สร้างเอเอสพีดอทเน็ต ขึ้นมาใหม่หมดบนฐานจาก Common Language Runtime (CLR) ซึ่งทำให้ผู้พัฒนาสามารถเลือกใช้ภาษาใดก็ได้ที่รองรับโดยดอทเน็ตเฟรมเวิร์ก เช่นซีชาร์ฟ (C#) และวีบีดอทเน็ต (VB.NET) เป็นต้น ปัจจุบันรุ่นล่าสุดคือเอเอสพีดอทเน็ต 4.0 ซึ่งรวมอยู่ใน ดอทเน็ตเฟรมเวิร์ค 4.0 (.NET Framework 4.0)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 รูปแบบไฟล์เอเอสพีเอ็ก (ASPX)

เอเอสพีเอ็ก (ASPX) เป็นชื่อรูปแบบไฟล์ของหน้าแบบฟอร์ม เอเอสพีดอทเน็ต โดยทั่วไปแล้วในไฟล์จะมีรหัสแบบ HTML หรือ XHTML ซึ่งใช้กำกับรูปแบบฟอร์ม หรือ เนื้อหาในหน้าเว็บ และในส่วนของโค้ดนั้น อาจจะอยู่ในหน้าเดียวกันในแท็ค หรือ บล็อก <% -- รหัสที่ใช้ -- %> (โดยในกรณีนี้จะคล้ายกับเทคโนโลยีที่ใช้พัฒนาเว็บ อย่าง PHP และ JSP) หรือแยกอยู่ในหน้าโค้ดออกมาต่างหาก (Code behind) ASP.NET รองรับการเขียนโค้ดในบรรทัดเดียวกันทั้งหมดในไฟล์ ASPX แต่วิธีนี้นั้นเป็นวิธีที่ไม่แนะนำ

2.2.2 ASP.NET กับ คุณสมบัติพิเศษ และ ข้อดี

ASP.NET หรืออีกชื่อหนึ่งว่า ASP+ ซึ่งเป็นชื่อที่ Microsoft ใช้เรียกในตอนแรก ถือว่าเป็น ASP เวอร์ชั่น ล่าสุดต่อจาก ASP 3.0 แต่ได้เปลี่ยนรูปแบบ และไวยากรณ์ต่างๆ และ ภาษา ที่นำมาใช้งานนั้นต่างจากเดิมไปมาก ความแตกต่างระหว่างเอเอสพีดอทเน็ต (ASP.NET) และเอเอสพี (ASP) รุ่นก่อนสรุปได้ดังนี้

2.2.2.1 ใช้ ภาษา ใดๆในการเขียนสคริป (Script) จากเดิมที่ใช้ได้เฉพาะภาษา ที่เป็นสคริป (Script) ของ VBScript และ JScript แต่ใน ASP.NET สามารถที่จะใช้ ภาษาที่มีรูปแบบของภาษาเต็มๆ ซึ่ง ในเบื้องต้น มี 3 ภาษา คือ C#, VB.NET และ JScript.Net ที่ออกมาเป็น มาตรฐาน แต่ในอนาคต Microsoft มีแผนที่จะเพิ่ม ตัวแปลภาษาให้ครบ ทุกภาษา

2.2.2.2 มีความยืดหยุ่นใน การเขียนโปรแกรม มากขึ้น โดยที่สามารถใช้ ภาษาในการเขียน ASP.NET ได้มากกว่า 1 ภาษาภายในไฟล์เดียวกัน ทำให้สามารถเลือกรูปแบบของภาษาที่ง่ายที่สุดต่อการเขียน ในแต่ละส่วนได้

2.2.2.3 ลักษณะ การแปลภาษา และ นามสกุล ไฟล์เปลี่ยนไป ใน ASP เวอร์ชั่นก่อน มีลักษณะการแปลภาษาเป็นแบบ อินเตอร์พรีเตอร์ (Interpreter) คือการจะทำคำสั่งใดค่อยแปลคำสั่งนั้น แต่ในเวอร์ชั่น .NET นี้จะมี ลักษณะเป็น คอมไพเลอร์ (Compiler) คือการแปลคำสั่งรวมทั้งโปรแกรม นอกจากนี้ นามสกุล ของไฟล์ก็มี การเปลี่ยนแปลง จากเดิมที่ใช้นามสกุลไฟล์เป็น "*.asp" เป็น "*.aspx"

2.2.2.4. รูปแบบและการใช้งานในส่วนต่างๆ (Component) ที่ง่ายขึ้น รูปแบบของ Component จะเน้นไปที่ XML มากที่สุด และที่สำคัญคือการใช้งาน component ใน ASP.NET นั้น เราสามารถอัพโหลดไฟล์ไปไว้ใน Directory ที่ผู้ดูแลเซิฟเวอร์ (Admin) กำหนดหลังจากนั้น Component จะติดตั้งตัวเองโดย อัตโนมัติ ลดปัญหาที่เกิดจาก ASP เวอร์ชั่นก่อนๆได้เป็นอย่างดี เนื่องจากใน ASP เวอร์ชั่นก่อนนั้น การติดตั้ง Component กระทำได้เพียงผู้ดูแลเซิฟเวอร์ เพียงคนเดียวเท่านั้น ทำให้เวลาต้องการใช้คอมโพเนนต์ต่างๆที่ Server ไม่มีจึงเป็นเรื่องที่ลำบาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.5 มี Library ให้เลือกใช้ได้มากขึ้น ใน ASP เวอร์ชั่นก่อนๆนั้น Application บางอย่างสร้างได้ไม่สะดวก ต้องอาศัย component ต่างๆมากมาย แต่ใน ASP.NET นั้นได้เพิ่ม Libraryในส่วนเหล่านี้ให้กลายเป็น พื้นฐาน ของการใช้งาน

2.2.2.6 มีคอนโทรล (Control) ทำให้การใช้งานในบางสิ่งง่ายขึ้น เป็น ส่วนพิเศษ ที่เพิ่มเติมมาจาก ASP รุ่นก่อนๆที่ไม่มีส่วนที่เรียกว่า คอนโทรล ซึ่งคอนโทรลนี้จะช่วยให้เราสามารถ สร้างเว็บไซท์ ได้อย่างง่ายและมีประสิทธิภาพมากขึ้น

2.2.2.7 สามารถเรียกขอข้อมูลจากเซิฟเวอร์ได้ ใน ASP เวอร์ชั่นก่อน Server สามารถเรียกขอข้อมูลได้จาก เครื่องผู้ใช้ เท่านั้นแต่ใน ASP.NET เครื่อง Server สามารถเรียกขอข้อมูลจากเครื่อง Server ด้วยกันได้

2.2.2.8 ไม่ต้องต่อฮาร์ดแวร์ (Hardware) เนื่องจากเป็นระบบใน .NET Framework ดังนั้นจึงมีคุณสมบัติของ Common Language Runtime (CLR) ทำให้มี การคอมไพล์ โปรแกรม เป็น ภาษามาตรฐาน ที่เรียกว่า IL ก่อน ดังนั้นไม่ว่าจะใช้ ปาล์ม, โน้ตบุ๊ค หรือ PDA ก็ไม่เกิดปัญหา

2.2.2.9 ง่ายต่อการหา ข้อผิดพลาด ใน การเขียนโปรแกรม หากเป็น ASP รุ่นก่อนเวลาเกิดความผิดพลาด (Error) เครื่องจะบอกแค่ว่าเป็น ความผิดพลาด ชนิดใดบรรทัดไหน แต่ใน ASP.NET นี้เครื่องจะแสดงรายละเอียดที่มากขึ้น พร้อมแนวทางแก้ไข

2.2.2.10 มีการ ตรวจสอบ เหตุการณ์ต่างๆได้ภายในเว็บเพจ (Webpage) มีการตรวจสอบเหตุการณ์ ต่างๆ ตั้งแต่โหลด หน้าเว็บเพจ ไปจนถึง ปิดหน้าเว็บ เพจลง ทำให้เราสามารถ เขียนโปรแกรม กำหนดเหตุการณ์ต่างๆได้ง่ายขึ้น

2.2.2.11 แยกส่วนที่เป็น HTML กับ ASP ออกมาอย่างชัดเจน ในเวอร์ชั่นก่อน ส่วนที่เป็น HTML กับ ASP จะเขียนปนกันไปมา แต่ใน เวอร์ชั่น นี้จะแยกส่วนกันอย่างชัดเจนว่าส่วนไหนเป็น HTML และส่วนไหนเป็น ASP

2.3 ซีชาร์ป (C#)

ภาษาซีชาร์ป(C#) เป็นภาษาที่ใหม่มากๆ ปรากฏตัวเป็นครั้งแรกในปี 2000 และถูกอัพเดตเวอร์ชันอยู่เสมอ มันเป็นภาษาที่ได้รับอธิพลจากภาษาก่อนหน้าเช่นภาษา Delphi ภาษา C++ ภาษา Java และภาษา Eiffel ในตอนต้นภาษานี้ถูกออกแบบ และกำหนดลักษณะโดยบริษัทไมโครซอฟท์ ต่อมาได้ถูกรับรองจากหน่วยงาน ECMA (หน่วยงานกำหนดมาตรฐานสากลด้านสารสนเทศ) และ ISO แต่ปัจจุบันไมโครซอฟท์ยังพัฒนาภาษานี้อย่างต่อเนื่อง (ปัจจุบันเป็นเวอร์ชัน 3.0)

ภาษาซีชาร์ป (C#) ถูกพัฒนาขึ้นโดยเป็นส่วนหนึ่งในการพัฒนาโครงสร้างพื้นฐานของ .NET Framework เป็นการการนำข้อดีของภาษาต่างๆ (เช่นภาษา Delphi ภาษา C++) มาปรับปรุง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้มีความเป็น OOP อย่างถึงที่สุด ขณะเดียวกันก็ลดความซับซ้อนในโครงสร้างของภาษาลง (เรียบง่ายกว่าภาษา C++) และมีเครื่องแต่งตัวน้อยลง (เมื่อเทียบกับ Java)

ผู้สร้างภาษา C# คือบริษัทไมโครซอฟท์ แต่บิดาของภาษา C# คือ Anders Hejlsberg (แอนเดรส ฮาเยสเบิร์ก) ไมโครซอฟท์ต้องการให้ภาษา C# เป็น "อะไรที่จะอยู่ไปอีกนาน" เหมือนบริษัทรถยนต์โฟลค์ที่จ้าง Ferdinand Porsche (เฟอร์ดินันด์ พอร์ช) นักออกแบบรถยนต์มือดีมาออกแบบรถโฟลคเต่า (เป็นเหตุการณ์ที่เกิดขึ้นก่อนสงครามโลกครั้งที่สอง) ทำให้มันกลายเป็นรถยนต์คลาสสิกมาจนถึงปัจจุบัน ในทำนองเดียวกันเมื่อต้องการภาษาที่ต้องการให้กลายเป็นภาษา "คลาสสิก" บริษัทไมโครซอฟท์ตัดสินใจมอบหมายให้ Hejlsberg นักออกแบบภาษา ผู้เคยสร้างภาษาที่กลายเป็นตำนานมาแล้วเช่น Turbo Pascal และผู้นำในทีมสร้างภาษา Delphi

2.4 เอแจ็กซ์ (Ajax)

ชุดของเทคโนโลยีต่างๆ เอแจ็กซ์ (Ajax) ย่อมาจาก Asynchronous JavaScript And XML; ซึ่งหมายถึงการทำงานร่วมกันของ JavaScript และ XML แบบไม่ประสานกัน (Asynchronous) มีหลักการทำงาน 2 ประเด็น คือ การอัพเดทหน้าจอแบบบางส่วน และการติดต่อสื่อสารกับเซิฟเวอร์ (Server) โดยใช้หลักการ Asynchronous ทำให้ผู้ใช้ไม่ต้องหยุดการทำงาน เพื่อรอการประมวลผลจากเซิฟเวอร์ (Server) รวมถึงการโหลดและการรีเฟรชหน้าจอ ของบราวเซอร์ทางฝั่ง Client มีการใช้ เอแจ็กซ์ (Ajax) โดยการเพิ่มเลเยอร์ระหว่างผู้ใช้ บราวเซอร์ กับ Server ทำให้ผู้ใช้สามารถทำงานได้โดยไม่ต้องรอให้ไคลแอนด์ (Client) ติดต่อไปยังเซิฟเวอร์ รวมถึงการโหลดและการรีเฟรชหน้าจอทั้งหมดด้วย ดังนั้นผู้ใช้สามารถใช้งานแอปพลิเคชั่น (Application) ได้อย่างมีประสิทธิภาพมากขึ้น

เอแจ็กซ์ (AJAX) จึงไม่ใช่เทคโนโลยีในตัวของมันเอง แต่ว่าเป็นการนำเทคโนโลยีหลายๆ ตัวมารวมกันเช่น JavaScript, DHTML, XML, Css, Dom และ XMLHTTPRequest

Ajax engine ทำหน้าที่เป็นตัวกลางระหว่าง Client และ Server ฉะนั้นเมื่อ Client มี requestแทนที่จะส่ง HTTP request ไปยัง Server โดยตรง Client จะส่ง JavaScript call ไป Ajax engine เพื่อโหลดข้อมูลที่ผู้ใช้ต้องการ และหาก Ajax engine ต้องการข้อมูลเพิ่มเติมในการตอบสนองต่อผู้ใช้ Ajax engine จะส่งรีเควสไปยัง Server โดยใช้ XML

การ์เรตได้กล่าวถึงเทคโนโลยีต่าง ๆ ที่เป็นส่วนประกอบของ เอแจ็กซ์ (Ajax) ซึ่งได้แก่

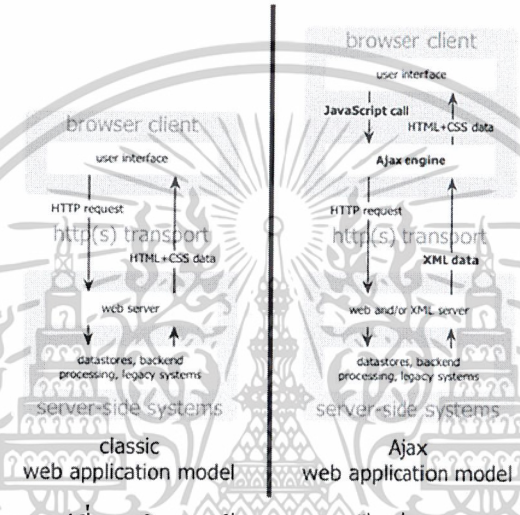
- HTML/XHTML เป็นภาษาในการจัดแสดงข้อมูล
- CSS เป็นรูปแบบการจัดแต่ง XHTML
- Document Object Model (DOM) สำหรับ dynamic display and interaction
- XML เป็นรูปแบบการแลกเปลี่ยนดาต้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- XSLT สำหรับ แปลง XML เป็น XHTML
- XMLHTTPRequest สำหรับ asynchronous data retrieval
- JavaScript เป็นภาษาในการใช้งาน Ajax engine

โดยส่วนประกอบจำเป็นขั้นพื้นฐานที่ขาดไม่ได้ในเอแจ็กซ์ (Ajax) ได้แก่ HTML/XHTML DOM และ JavaScript เพราะ XHTML

2.4.1 เปรียบเทียบการทำงานแบบเดิม กับ เอแจ็กซ์ (Ajax)



รูปที่ 2.3 โครงสร้างของ เอแจ็กซ์ (Ajax)

มุมมองของโครงสร้างทาง Software ของ เอแจ็กซ์ (Ajax) ต่างจากเว็บแอพลิเคชั่นในทุกวันนี้ เนื่องจากมีการเพิ่ม engine ทางฝั่ง Client

2.4.2 การทำงานของ เอแจ็กซ์ (Ajax)

เอแจ็กซ์ (AJAX) จะช่วยลดการติดต่อระหว่าง Client กับ Server โดยในการโหลดหน้าเว็บนั่น บราวเซอร์จะโหลดข้อมูลจาก เอแจ็กซ์ (AJAX) engine แทนการร้องขอข้อมูลจาก Server โดยตรง ดังนั้น เอแจ็กซ์ (Ajax) จะทำหน้าที่ทั้งการ render ส่วนติดต่อกับผู้ใช้และติดต่อไปยัง Server แล้ว เอแจ็กซ์ (AJAX) engine อนุญาติให้การกระทำต่างๆ ใน web application เป็นแบบ Asynchronous คือความเป็นอิสระในการติดต่อไปยัง Server นั่นเอง ดังนั้นผู้ใช้จะไม่พบกับบราวเซอร์หน้าขาวๆ อีกต่อไป และไม่ต้องรอการโหลดข้อมูลต่างๆ จาก Server

2.4.3 ข้อดีของ เอแจ็กซ์ (Ajax)

- ตอบสนองต่อผู้ใช้ได้อย่างรวดเร็วเนื่องจากการ update แบบบางส่วน
- ผู้ใช้ไม่ต้องหยุดรอคอยการประมวลของ Server เนื่องจากการติดต่อแบบ Asynchronous

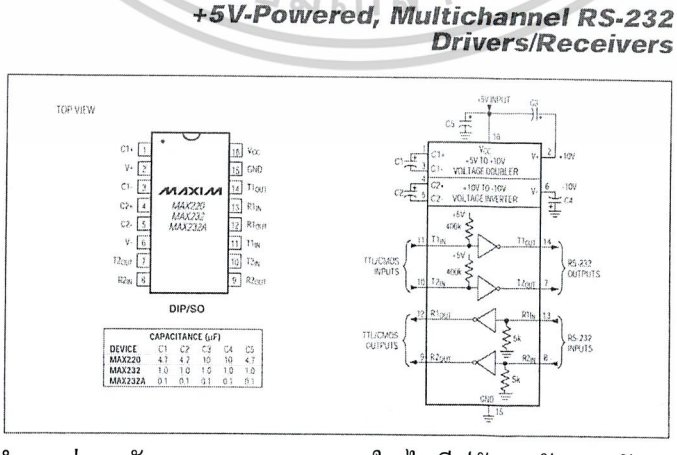
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รองรับกับบราวเซอร์หลักๆที่สามารถใช้ JavaScript ได้
- ทำให้การประมวลผลที่ Server มีความรวดเร็วขึ้นเนื่องจากการประมวลผลที่ Server ลดลง
- ไม่ต้องทำการติดตั้ง หรือใช้ Plugs-in
- ไม่ยึดติดกับ Platform หรือภาษาที่ใช้ในการเขียนโปรแกรม
- เป็นเ ทคโนโลยีใหม่ที่ไม่ได้เป็นของนักพัฒนาเว็บแอพลิเคชั่นคนใด นั่นคือทุกคนมีสิทธิ์เข้ามาพัฒนาแอพลิเคชั่นตัวนี้

2.5 MAX232

ไอซี MAX232 เป็นตัว Driver เพื่อทำหน้าที่แปลงระดับแรงดัน RS-232 จากฝั่งตอมพิวเตอร์ให้อยู่ในระดับแรงดัน TTL จะนำไปต่อกับขา TX, RX ของ IC4 เพื่อใช้รับส่งข้อมูลระหว่างคอมพิวเตอร์และ IC โดยผ่าน J2 DB9 แบบ Female

ไอซีปรับระดับแรงดัน MAX 232 เป็นไอซี 16 ขาที่ใช้ทำหน้าที่ในการรับส่งข้อมูลแบบอนุกรม (Serial) ระหว่างอุปกรณ์ TTL/CMOS กับพอร์ท RS-232 โดยไอซีปรับระดับแรงดัน MAX 232 มีช่องทางในการรับข้อมูลแบบอนุกรม (Serial) 2 ช่องทางและมีช่องทางในการส่งข้อมูลแบบอนุกรม (Serial) 2 ช่องทาง โครงสร้างภายในของไอซีปรับระดับแรงดัน MAX 232 ยังมีวงจรทวีแรงดัน (Voltage Doublers) และวงจรอินเวอร์สแรงดัน (Voltage Inverter) วงจรทวีแรงดันเป็นวงจรที่ทำหน้าที่ยกระดับแรงดันจาก 5 โวลท์เป็น 10 โวลท์ และวงจรอินเวอร์สแรงดันเป็นวงจรที่ทำหน้าที่กลับสัญญาณจาก +10 โวลท์เป็น –10 โวลท์ วงจรทั้งสองทำหน้าที่หลักก็คือขยายสัญญาณก่อนที่จะส่งออกไปยังพอร์ท RS-232 รูปตำแหน่งขาสัญญาณและวงจรเทียบการทำงานของ MAX 232 ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 ตำแหน่งขาสัญญาณและวงจรภายในไอซีปรับระดับแรงดัน MAX 232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ห้องสมุดคณะเทคโนโลยีสารสนเทศ พระจอมเกล้าลาดกระบัง

2.5.1 การทำงานของไอซีปรับระดับแรงดัน MAX 232

2.5.1.1. เมื่อส่งข้อมูลออก โดยไอซีปรับระดับแรงดัน MAX 232 จะรับข้อมูลจากอุปกรณ์ TTL/CMOS เข้ามาที่ขา TX IN จากนั้นไอซีปรับระดับแรงดัน MAX 232 จะทำการยกระดับแรงดันของข้อมูลให้เป็น +10 โวลท์หรือ –10 โวลท์ และส่งออกมาที่ขา TX OUT ผ่านพอร์ท RS-232 เพื่อส่งข้อมูลต่อไปยังอุปกรณ์ภายนอก

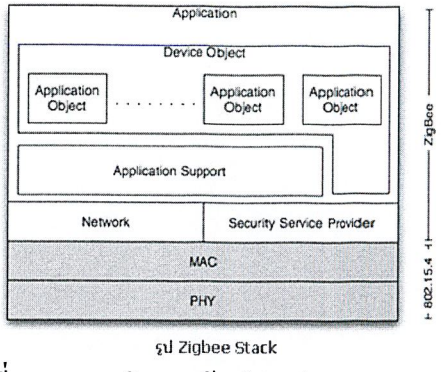
2.5.1.2. เมื่อรับข้อมูลเข้า โดยไอซีปรับระดับแรงดัน MAX 232 จะรับข้อมูลที่ส่งมาจากอุปกรณ์ภายนอกผ่านพอร์ท RS-232 มายังขา RX IN ของไอซีปรับระดับแรงดัน MAX 232 จากนั้นไอซีปรับระดับแรงดัน MAX 232 จะทำการแปลงระดับแรงดันของข้อมูลให้เป็น 0 โวลท์หรือ 5 โวลท์ และส่งออกที่ขา RX OUT เพื่อทำการส่งข้อมูลต่อไปที่อุปกรณ์ TTL/CMOS

2.6 ZigBee

ZigBee มาตรฐานสากล กำหนดโดย ZigBee Alliance เป็น การสื่อสารแบบไร้สาย ที่มีอัตราการรับส่งข้อมูลต่ำ ใช้พลังงานต่ำ ราคาถูก จุดประสงค์ก็เพื่อให้สามารถสร้างระบบที่เรียกว่า Wireless Sensor Network ได้ ซึ่งระบบนี้ จะสามารถทำงาน ในร่ม กลางแจ้ง ทนแดด ทนฝน และอยู่ได้ด้วยแบตเตอรี่ก้อนเล็ก (เช่นถ่าน AA 2 ก้อน) นานเป็นเดือน เป็นปี เหมาะสมใช้งานกับพวก monitoring ต่าง ๆ

ZigBee ถูกสร้างขึ้นในการทำระบบเครือข่ายไร้สายส่วนบุคคล(WPAN) อยู่ภายใต้มาตรฐาน IEEE 802.15.4 โดยมาตรฐานนี้ใช้งานสำหรับการสื่อสารความเร็วต่ำ ใช้กำลังไฟฟ้าน้อย อุปกรณ์ราคาถูก และมีคุณสมบัติการจัดการตัวเองได้ เป็นเทคโนโลยีไร้สายที่ร่วมกันสื่อสารข้อมูลผ่านเซ็นเซอร์ขนาดเล็กจิ๋ว จำนวนเป็นพันๆ หมื่นๆ ชิ้นที่ฝังอยู่ตามส่วนต่างๆ ในอาคาร สำนักงาน โรงงาน หรือแม้แต่ในบ้าน การทำงานของ ZigBee จะเป็นการรับ-ส่งคลื่นสัญญาณข้อมูล ผ่านชิปเล็กจิ๋วนี้จุดต่อจุดไปเรื่อยๆ จนถึงปลายทางที่ต้องการดาวน์โหลดข้อมูลลงในเครื่องคอมพิวเตอร์เพื่อใช้ในการ วิเคราะห์ข้อมูล ข้อมูลที่ได้อาจจะเป็นการวัดอุณหภูมิ การเคลื่อนไหวของสิ่งมีชีวิต จับปริมาณมลพิษในอากาศ ปริมาณน้ำ ท่อแก๊สโดยใช้พลังงานแสงอาทิตย์หรือแบตเตอรี่ขนาดเล็กที่กินไฟน้อยมาก ZigBee นำ Physical Layer และ MAC Layer ของ IEEE 802.15.4 ซึ่งเป็นมาตรฐานการกำหนดการสื่อสารไร้สายแบบ WPAN (Wireless Personal Area Network) มาทำงานใน Layer ที่ต่ำกว่า เช่น เรื่องของ ระดับกำลังสัญญาณ , Link Quality , Access control , Security ฯลฯ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
06597



รูปที่ 2.5 แสดงโครงสร้างในรูป ZigBee Stack

ZigBee สามารถสร้างเป็นเครือข่ายได้ ทั้งนี้ ZigBee ได้อ้างอิงมาตรฐานตาม IEEE 802.15.4 โดยที่ IEEE 802.15.4 จะแบ่งชนิดอุปกรณ์ในเครือข่ายออกเป็น 2 ประเภท คือ FFD (Full Function Device) ซึ่งหมายถึงอุปกรณ์ที่สามารถทำงานได้ทุกอย่างในเครือข่ายและ RFD (Reduce Function Device) ซึ่งหมายถึงอุปกรณ์ที่ถูกลดความสามารถการทำงานในเครือข่าย

2.6.1 ชนิดอุปกรณ์ของ ZigBee มีอยู่ 2 ชนิดคือ แบบ Physical Device และ Logical Device

2.6.1.1แบบ Physical Device มี 2 ประเภท คือ

2.6.1.1.1 Full Function Device: FFD เป็นเราเตอร์ที่เป็นสื่อกลางในการส่งข้อมูลจากอุปกรณ์อื่นๆ ใช้พลังงานจาก power line ทำงานได้ในทุก Topology และสามารถทำเป็นจุดเชื่อมต่อกันได้

2.6.1.1.2 Reduced Function Device: RFD เหมาะแก่การเชื่อมต่อภายในเครือข่าย ใช้พลังงานจากแบตเตอรี่ ไม่สามารถถ่ายทอดข้อมูลจากอุปกรณ์อื่นๆ ได้ ทำได้ง่ายในเครือข่ายที่เป็นแบบ star

2.6.1.2 แบบ Logical Device มี 3 ประเภท คือ

2.6.1.2.1 ZigBee Coordinators เป็นจุดที่ประสานเชื่อมต่อกัน ทำหน้าที่ในการจัดเก็บข้อมูลในเครือข่าย

2.6.1.2.2 ZigBee Routers ทำหน้าที่จัดการเส้นทางของข้อความที่ส่งผ่านภายในโครงข่ายระหว่างคู่ของโนดใด ๆ และ

2.6.1.2.3 ZigBee End Devices เป็นโหนดที่อยู่ในส่วนของผู้ใช้งาน โดยสามารถเป็นได้ทั้งแบบ RFD และ FFD

2.6.2 ZigBee ได้แบ่งตามลักษณะการทำงาน 3 แบบ คือ

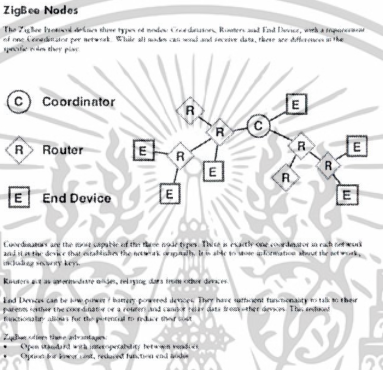
2.6.2.1. Coordinator มีหน้าที่สร้างการสื่อสารเชื่อมโยงเครือข่ายระหว่าง End

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Device กับ Router หรือระหว่าง Coordinator กับ Coordinator ด้วยกัน หรือระหว่าง Coordinator กับ Router กำหนด Address ให้กับ Device ที่อยู่ในวงเครือข่าย ไม่ให้ซ้ำกัน ดูแลจัดการเรื่องการ Routing เส้นทาง ซึ่งเทียบได้กับ FFD

2.6.2.2. End Device เป็นอุปกรณ์สุดปลายทาง ซึ่งจะใช้รับสัญญาณมาจาก Sensor ที่ปลายทาง โดยที่ใช้พลังงานต่ำในการทำงาน เทียบได้กับ RFD หรือ FFD บางกรณีขึ้นอยู่กับ Sensor ที่ใช้

2.6.2.3. Router มีหน้าที่รับส่งข้อมูล และส่งต่อไปในเส้นทางต่าง ๆ ของเครือข่าย ซึ่งเทียบได้กับ FFD



รูปที่ 2.6 แสดงโครงสร้าง ส่วนประกอบ และรูปแบบของ ZigBee

นอกจากนี้ยังสามารถประยุกต์ใช้กับอุปกรณ์พื้นฐานที่หลากหลายในชีวิตประจำวัน ถ้าเรานำมาตรฐานเครือข่ายแบบไร้สาย IEEE 802.15.4 มาประยุกต์ใช้แบบยูบิควิตัส โดยเป็นการสื่อสารระหว่างอุปกรณ์ กับอุปกรณ์ หรือ อุปกรณ์ กับมนุษย์ ที่ผ่านระบบเครือข่ายไร้สาย ทั้งนี้ประโยชน์ที่ได้รับมีดังนี้

- ระบบการควบคุมอัตโนมัติ ที่บ้าน โรงงาน และ โกดังเก็บสินค้า เป็นต้น
- ระบบการติดตามสำหรับ ความปลอดภัย ชีวะอนามัย และสิ่งแวดล้อม เป็นต้น
- การตรวจหาตำแหน่งที่นำไปใช้ใน การปฏิบัติการทางทหาร การทำงานของนักผจญเพลิง และ บริษัทที่ต้องการการตรวจหาตำแหน่งแบบเวลาจริง
- ให้ความบันเทิง เช่น เกมฝึกทักษะ และของเล่นแบบ interactive

มาตรฐาน 802.15.4 หรือ ZigBee ถ้ามีการใช้งานเกิดขึ้นจริงนั้นจะมีผลต่อการใช้ชีวิตประจำวันของเราอย่างมาก เช่น ในทางชีวะอนามัย นอกจากนี้ระบบ 802.15.4 จะช่วยเตือนภัยจากสิ่งแวดล้อม รวมถึงอุบัติภัยต่างๆ เช่น ไฟไหม้ น้ำท่วม แผ่นดินไหว เป็นต้น ทั้งนี้ระบบเตือนภัยในปัจจุบันไม่ได้เชื่อมต่อกันเป็นระบบเครือข่าย และตัวอุปกรณ์เองมีช่วงการใช้งานจากแบตเตอรี่สั้น นอกจากนี้ยังมีราคาแพงอีกด้วย แต่ในระบบ 802.15.4 สามารถใช้ได้กับอุปกรณ์พื้นฐานเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซนเซอร์ และ Actuators ที่มีราคาถูก ทำให้สามารถติดตามเหตุการณ์ต่างๆ และอุปกรณ์จะทำงานอย่างอัตโนมัติตามที่เราต้องการ

2.7 เครือข่ายแบบ Client-Server

ระบบ Client-Server เป็นสถาปัตยกรรมซอฟท์แวร์ที่ได้รับการออกแบบให้แยกออกเป็น 2 ส่วน ส่วนแรกเรียกว่าส่วน Client และอีกส่วนเรียกว่า Server โดยปกติแล้วข้อมูลจะอยู่ข้าง Server ในฐานข้อมูล ซึ่งอาจเป็น ฐานข้อมูล MS Access, MS SQL Server, Oracle ในส่วนของข้าง Client จะส่งคำสั่ง SQL Statement แล้วดึง ข้อมูลจากฐานข้อมูล ที่ส่งไปยัง Client

2.7.1 กระบวนการทำงานของ Client - Server

2.7.1.1 Client คือซอฟต์แวร์ส่วนที่เป็นกระบวนงานในการขอบริการ ข้อมูล (Launcher/Requester process)

2.7.1.2 Client Application จะติดต่อกับ Client application อื่นๆได้ และใช้ทรัพยากรร่วมกันได้และติดต่อขอใช้ข้อมูลและ บริการจาก Server ต่างๆทำให้เพิ่มขีดความสามารถของผู้ใช้งาน Client สามารถมีหน้าจอของตัวเอง ได้รับการออกแบบมาให้ผู้ใช้สามารถใช้งานได้สะดวกโดยที่ไม่ต้องมีความรู้ด้าน กลไกลที่อยู่เบื้องหลังนั่นคือ Client จะซ่อนความซับซ้อนของระบบปฏิบัติการเครือข่าย (Network Operating System) กรรมวิธีการนำเสนอข้อมูลมาใช้ ทำให้ผู้ใช้รู้สึกว่าสามารถทำงานได้อย่างสะดวกสบายตาม Business Rule ที่ตัวเองเข้าใจ

2.7.1.3 Server เป็นซอฟต์แวร์ที่สามารถตอบสนองต่อการขอบริการ และข้อมูลของ Client มีหน้าที่คือการตีความ Request ของ Client การจัดการกับขั้นตอนการ Access ข้อมูล หลังการให้บริการข้อมูล เฉพาะที่ต้องใช้ซอฟต์แวร์ Server อาจอยู่บนเครื่องคอมพิวเตอร์ เครื่องเดียวกันได้ (เป็น Multiple Server Processor) หรือบน ต่างเครื่องกันก็ได้

2.7.2 ข้อดีของ Client – Server

ในอดีตถ้าเราต้องการระบบคอมพิวเตอร์ที่มีเครื่องแม่ข่าย ที่คอยจัดการควบคุม เครื่องลูกข่ายที่เป็น Terminal เราต้องใช้ สถาปัตยกรรมแบบ Mainframe ซึ่งระบบ Mainframe นี้ต้องใช้ทรัพยากรในการจัดการสูงมากพูดง่ายๆก็คือ มีราคาที่แพงมาก และยังมีข้อเสียอีกคือความยืดหยุ่นในการใช้งานค่อนข้างต่ำ แต่ในปัจจุบันระบบงานคอมพิวเตอรีที่ได้รับความนิยมมากขึ้นเรื่อยๆ และกำลัง เข้ามาแทนเครื่องแบบ Mainframe นั้นก็คือ ระบบ Client-Server เนื่องจาก มีความคล่องตัวในการใช้งานสูง และที่สำคัญคือ ราคาค่อนข้างต่ำเนื่องจากระบบ นี้ไม่ต้องการ เครื่องแม่ข่ายหรือ Server ที่มีความสามารถสูงนัก เป็นเพียงเครื่อง ระดับ Pentium ก็พอแล้ว เพราะระบบ Client-Server นี้ข้อมูลและ โปรแกรมต่างๆ จะไม่รวมกันอยู่ที่ศูนย์กลางเพียงแห่งเดียวทำให้การ Overhead ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 อินฟราเรด (Infrared)

Infrared หรือเรียกอีกอย่างหนึ่งว่า คลื่นความถี่สั้น (Millimeter waves) ซึ่งจะมีย่านความถี่คาบเกี่ยวกับย่านความถี่ของคลื่นไมโครเวฟอยู่บ้าง วัตถุร้อนจะแผ่รังสีอินฟราเรดที่มีความยาวคลื่นสั้นกว่า 10-4 เมตรออกมา ประสาทสัมผัสทางผิวหนังของมนุษย์สามารถรับรังสีอินฟราเรด

2.8.1 รังสีอินฟราเรด สามารถอธิบายได้ดังนี้

- เป็นคลื่นแม่เหล็กไฟฟ้าที่มีความถี่อยู่ในระหว่างแสงที่ตามองเห็น
- ลำแสงอินฟราเรดเดินทางเป็นเส้นตรง ไม่สามารถผ่านวัตถุทึบแสง และสามารถสะท้อนแสงในวัสดุผิวเรียบได้เหมือนกับแสงทั่วไป
- ใช้มากในการสื่อสารระยะใกล้ เช่น รีโมทคอนโทรลของเครื่องรับโทรทัศน์
- ปัจจุบันถูกพัฒนาใช้ในการสื่อสารไร้สาย สำหรับเครือข่ายเฉพาะบริเวณ
- เมื่อใช้ในการสื่อสารข้อมูลในเครือข่ายสามารถส่งสัญญาณได้ในระยะ 30-80 ฟุต หรือ 10-30 เมตร
- เป็นสื่อที่มีช่องสัญญาณกว้างพอประมาณในระดับสูงเมื่อเปรียบเทียบกับสายยูทีพี
- อุปกรณ์คอมพิวเตอร์ที่มีช่องสื่อสารอินฟราเรด เรียกว่า IrDa (Infrared Data Association) สามารถสั่งงานระยะไกลได้ประมาณ 1 – 5 เมตร
- เป็นระบบสำหรับควบคุมการทำงานของอุปกรณ์ต่าง ๆ จากระยะไกล โดยรังสีอินฟราเรดจะเป็นตัวนำคำสั่งจากเครื่องควบคุมไปยังเครื่องรับ

2.8.2 คุณสมบัติเด่นของ Infrared

- คลื่นสั้น ทางเดินของแสงเป็นแนวตรง
- ราคาถูก, ง่ายต่อการผลิต
- ปลอดภัยต่อการดักสัญญาณ
- ไม่สามารถทะลุผ่านวัตถุ ทำให้สามารถติดตั้ง Infrared ในห้องทำงานติดกันได้

2.8.3 หลักการทำงานของอินฟราเรดมี ดังนี้

2.8.3.1. จัดตำแหน่ง: ในการพิมพ์ไฟล์จากโน้ตบุ๊ค ให้วางอุปกรณ์นั้น 3 ฟุตจากเครื่องพิมพ์ที่เหมาะสมกับอินฟราเรด ชี้ไปที่พอร์ตอินฟราเรด (หรือที่เรียกว่าโฟโตไดโอด) โดยตรงไปยังโฟโตไดโอดของเครื่องพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.3.2. ส่ง: พัลส์ของแสงอินฟราเรดจะถูกส่งไปกลับระหว่างอุปกรณ์สองตัวเพื่อขนถ่ายแพ็กเก ตของข้อ มูลที่ประกอบกันเป็นแพ็กเกตจะถูกสื่อสารด้วยพัลส์เปิด/ปิดของแสงอินฟราเรด โดยพัลส์จะถูกอ่าน ในรปของรหัสไบนารี่

2.8.3.3. รับ: โฟโตไดโอดจะรับแพ็กเกต ซึ่งจะถูกแปรกลับไปเป็นข้อมูลอีกครั้งเครื่องพิมพ์หรือพีซีในด้าน รับจะประมวลผลข้อมูลที่ได้มาจากการเชื่อมต่อเครือข่ายที่ใช้สายเคเบิล

2.8.3.4. การขัดขวาง: ถ้ามีวัตถุมาขัดขวางลำของพัลส์ของอินฟราเรดขณะ ที่ข้อมูลกำลังถูกส่งสัญญาณจะ ถูกบล็อก อย่างไรก็ตามอุปกรณ์ด้านส่งจะรับรู้ข้อผิดพลาดและทำการส่งข้อมูลที่ขาดหาย ไปใหม่

2.8.4 ตัวอย่างอุปกรณ์ที่ใช้อินฟราเรด (Infrared)

2.8.4.1. Remote Control ของโทรทัศน์

2.8.4.2. โทรศัพท์มือถือ

2.8.4.3. PDA, Palm

2.8.4.4. คอมพิวเตอร์โน้ตบุ๊ค

2.8.4.5. เครื่องคอมพิวเตอร์มือถือ (Hand held) หรือเครื่องขนาดฝ่ามือ (palmtop)

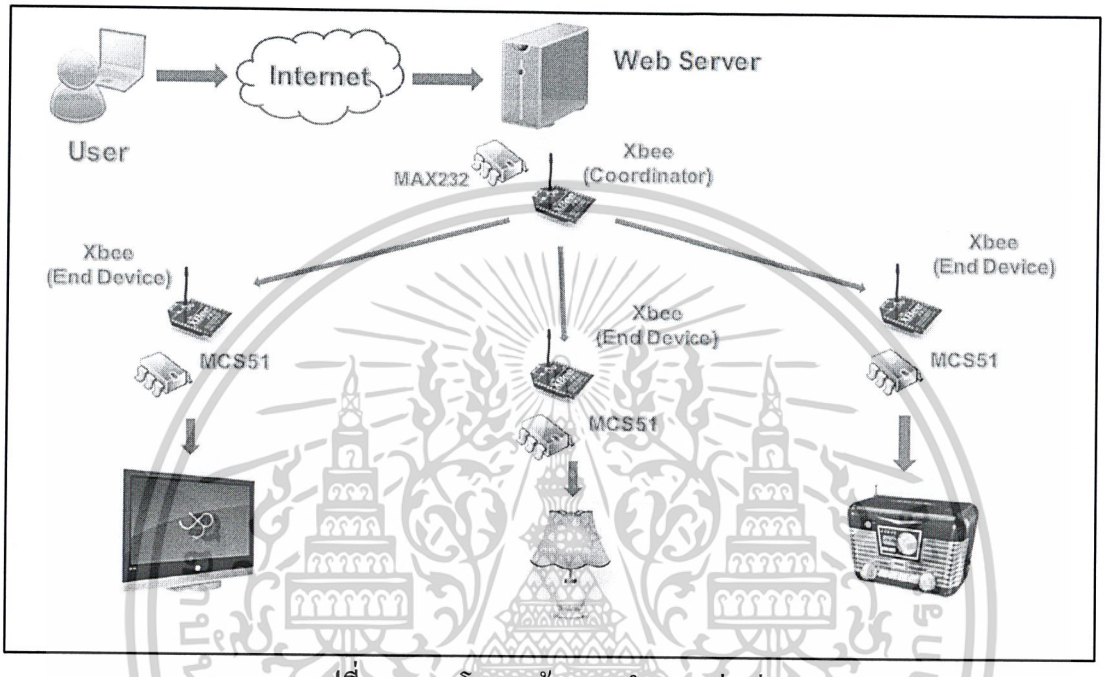
2.8.4.6. เครื่องเลเซอร์ในการรักษาโรค

2.8.4.7. กล้องส่องทางไกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การวิเคราะห์และออกแบบ

3.1 หลักการทำงาน และ การออกแบบในภาพรวม

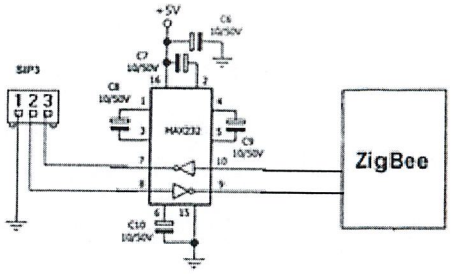


รูปที่ 3.1 ภาพโครงสร้างการทำงานอย่างง่าย

เมื่อผู้ใช้ (User) สั่งการจากคอมพิวเตอร์ (Computer) การผ่านอินเตอร์เน็ต (Internet) ไปยังเครื่องที่เป็นเซิร์ฟเวอร์ (Server) เครื่องที่เป็นเซิร์ฟเวอร์ จะทำการรับข้อมูล โดยจะเก็บข้อมูลในรูปเท็กไฟล์ (Text File) ซึ่งบันทึกสถานะต่างของเครื่องใช้ไฟฟ้าเอาไว้ แล้วส่งผ่านสายซีเรียลพอร์ต (Serial Port) ต่อไปยังส่วนของ MAX232 โดย MAX232 ทำหน้าที่แปลงระดับสัญญาณ RS-232 (ซึ่งหมายถึง Serial Port จะมีระดับสัญญาณที่ -12 V ถึง + 12 V) ให้เป็นสัญญาณ TTL (มีระดับสัญญาณที่ 0 ถึง + 5 V) เมื่อแปลงแล้ว จึงส่งข้อมูลผ่าน ZigBee ซึ่งเป็นการส่งแบบไร้สาย (Wireless) โดยค่าสถานะที่ประมวลผลนั้น จะมีส่งที่ระบุว่า ไปยังเครื่องใช้ไฟฟ้าไหน (ซึ่งจะบันทึกใน Text File ด้วยเช่นกัน) โดย ZigBee จะส่งสัญญาณกระจายไปทั่วบริเวณ โดยที่เครื่องใช้ไฟฟ้า ก็จะมี ZigBee ซึ่งเป็นตัวรับสัญญาณอยู่ ถ้าสัญญาณที่ได้รับ เป็นของตัวเอง ก็จะเอาสัญญาณนั้นมาประมวลผลผ่าน MCS51 เพื่อจะเปลี่ยนสถานะของอุปกรณ์ไฟฟ้า เมื่อเปลี่ยนแล้ว ก็จะส่งค่าที่เปลี่ยนนั้นไปยัง Relay เพื่อแปลงให้เข้ากับอุปกรณ์ไฟฟ้า ให้อุปกรณ์ไฟฟ้าใช้งานได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

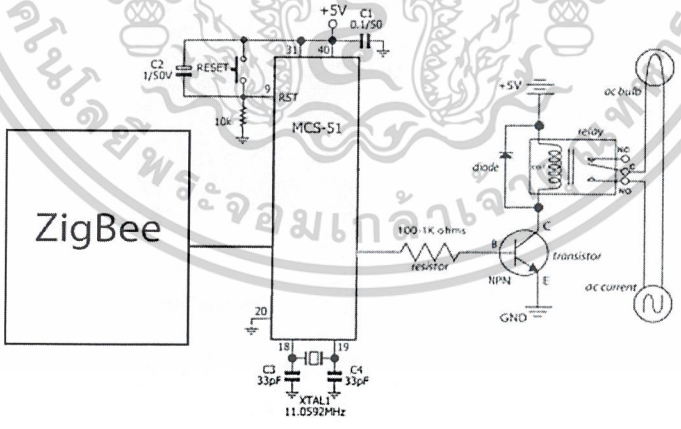
3.1.1 รายละเอียดในส่วนของ MCS-51 และ MAX232



รูปที่ 3.2 ภาพวงจรระหว่าง MAX232 และ ZigBee

เมื่อรับคำสั่งมาจากเซิร์ฟเวอร์ผ่านมายังซีเรียลพอร์ต(Serial Port) ไปยังไอซี MCS-51 จำเป็นต้องผ่านไอซี MAX232 ก่อน เนื่องจากว่าระดับสัญญาณจาก RS-232 (ซึ่งหมายถึง Serial Port) จะมีระดับสัญญาณที่ -12 V ถึง + 12 V (โลจิก +12 V มีค่าเท่ากับ 0 และโลจิก – 12 V มีค่าเท่ากับ 1) ทว่าไอซี MCS-51 เป็นแบบ TTL จะมีระดับสัญญาณที่ 0 ถึง + 5 V เท่านั้น ถ้าหากเราต่อ RS-232 โดยตรงกับตัวไอซี MCS-51 จะทำให้ไม่สามารถรับสัญญาณกันได้ เพราะระดับสัญญาณไม่ตรงกัน ไอซี MAX232 จึงมีหน้าที่เปลี่ยนระดับสัญญาณ RS-232 ให้เป็นระดับสัญญาณ TTL เมื่อระดับสัญญาณเป็นแบบ TTL แล้ว แล้วต่อไปยังส่วนโปรโตคอล ZigBee เพื่อส่งต่อไปยังเครื่องรับต่อไป

3.1.2 รายละเอียดส่วน MCS51 และ Relay Set



รูปที่ 3.3 ภาพวงจรระหว่างโปรโตคอล ZigBee , MCS-51 และ Relay ในเบื้องต้น

เนื่องจากอุปกรณ์ไฟฟ้า มีหลายอย่าง จึงต้องมีการตรวจว่า ข้อมูลที่ส่งมา เป็นเครื่องที่เราต้องการหรือไม่ ถ้าข้อมูลที่ส่งมาเป็นเครื่องใช้ที่เราต้องการผ่านโปรโตคอล ZigBee ก็จะรับข้อมูลแล้วส่งมาประมวลผลที่ไอซี MCS-51 เพื่อรับค่าสถานะ แล้วเปลี่ยนสถานะตามต้องการ ไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์ (Transistor) เปรียบเหมือนสวิชต์เปิด-ปิด ต่อไปยัง รีเลย์ (Relay) ซึ่งทำหน้าที่ตัดต่อวงจร และเชื่อมต่อกับอุปกรณ์ไฟฟ้า

กรณีต้องการเปิดเครื่องใช้ไฟฟ้าไอซี MCS-51 จะส่งสัญญาณออกไปยังทรานซิสเตอร์ (Transistor) โดยเมื่อจ่ายไฟฟ้าเข้าไปยังตัวทรานซิสเตอร์ (ซึ่งก็คือ สัญญาณที่ได้มาจากไอซี MCS51 นั้นเอง) ทำให้อุปกรณ์ไฟฟ้าครบวงจร อุปกรณ์รีเลย์ (Relay) ก็จะเริ่มการทำงาน เมื่อรีเลย์ได้รับไฟฟ้าตามที่กำหนด รีเลย์ก็จะเชื่อมต่อวงจรของเครื่องใช้ไฟฟ้าให้ครบวงจร เครื่องใช้ไฟฟ้าก็สามารถทำงานได้

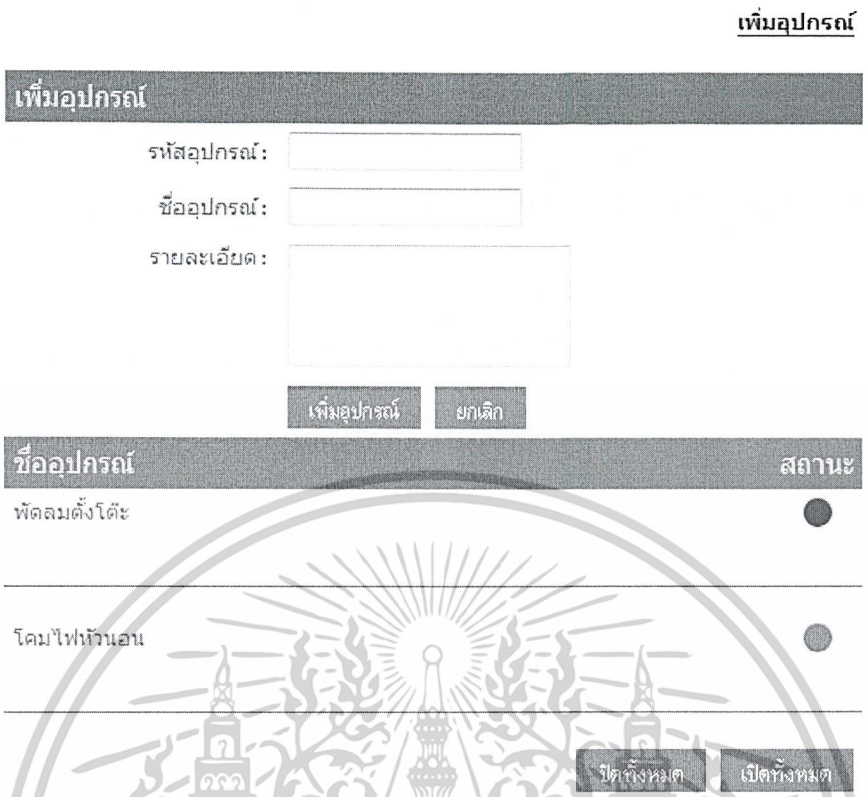
กรณีที่ต้องการปิดเครื่องใช้ไฟฟ้าไอซี MCS-51 จะหยุดส่งสัญญาณ ทำให้ทรานซิสเตอร์ไม่ทำงาน ส่งผลให้รีเลย์ตัดการเชื่อมต่อของวงจรเครื่องใช้ไฟฟ้า ทำให้ไม่ครบวงจร เมื่อไม่ครบวงจร เครื่องใช้ไฟฟ้าก็จะหยุดทำงาน จนกระทั่งไอซี MCS-51 ส่งสัญญาณมายังทรายซิสเตอร์อีกครั้งหนึ่ง

3.2. ตัวอย่างส่วนติดต่อกับผู้ใช้ (GUI) ของหน้าเว็บ

รูปที่ 3.4 แสดงหน้าเข้าสู่ระบบ

รูปที่ 3.5 แสดงหน้าแรกของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงหน้าเพิ่มอุปกรณ์

รูปที่ 3.7 แสดงหน้าแก้ไขอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการศึกษาและดำเนินงาน

4.1 หน้าจอการเข้าสู่ระบบ

ในการเข้าสู่ระบบ ผู้ใช้งานต้องกรอกรหัสให้ถูกต้องตามที่ได้ตั้งไว้แต่แรก มิเช่นนั้นจะไม่สามารถเข้าไปใช้งานได้

รูปที่ 4.1 แสดงหน้าจอสำหรับการเข้าใช้งานระบบ

4.2 หน้าจอแสดงสถานะของอุปกรณ์

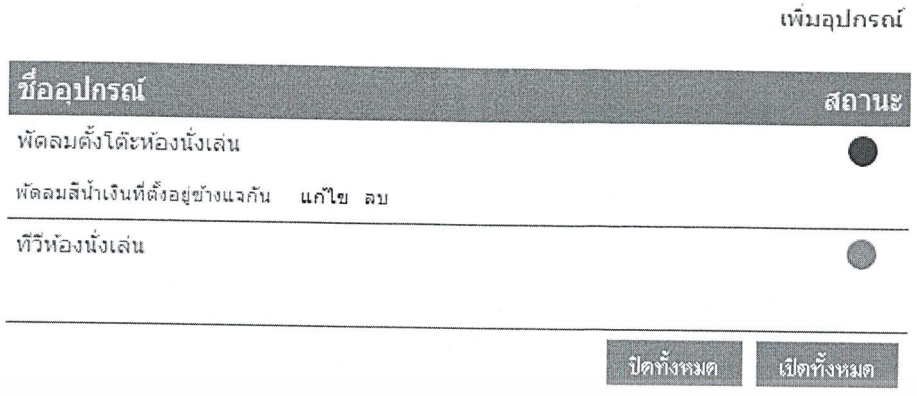
หน้าจอนี้จะแสดงรายชื่ออุปกรณ์ และสถานะของอุปกรณ์ที่ได้ทำการเพิ่มไว้ก่อนหน้า หากอุปกรณ์ใดถูกเปิดอยู่จะมีสัญลักษณ์เป็นรูปสีเขียว แต่หากอุปกรณ์ใดถูกปิดอยู่จะมีสัญลักษณ์เป็นรูปสีแดง แสดงอยู่ข้างหลังรายชื่อของอุปกรณ์นั้นๆ

รูปที่ 4.2 แสดงหน้าจอสถานะของอุปกรณ์

4.3 หน้าจอการแก้ไขรายละเอียดอุปกรณ์

หน้าจอนี้จะแสดงรายชื่ออุปกรณ์ และสถานะของอุปกรณ์ที่ได้ทำการเพิ่มไว้ก่อนหน้า หากอุปกรณ์ใดถูกเปิดอยู่จะมีสัญลักษณ์เป็นรูปสีเขียว แต่หากอุปกรณ์ใดถูกปิดอยู่จะมีสัญลักษณ์เป็นรูปสีแดง แสดงอยู่ข้างหลังรายชื่อของอุปกรณ์นั้นๆ

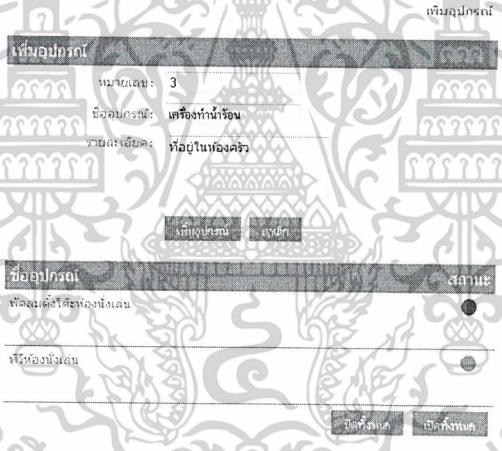
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงหน้าจอการแก้ไขรายละเอียดอุปกรณ์

4.4 หน้าจอการเพิ่มอุปกรณ์

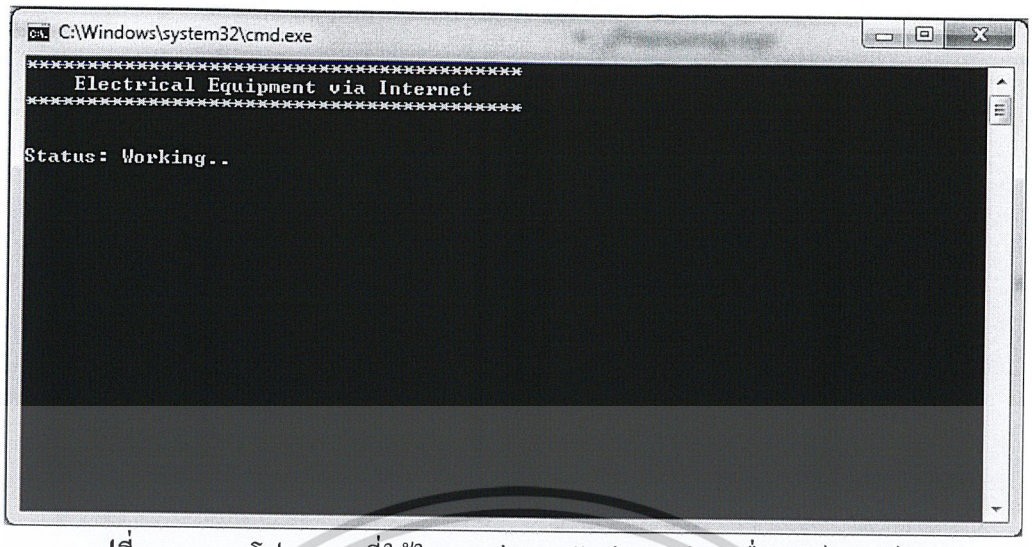
หากต้องการเพิ่มอุปกรณ์เข้าไปในระบบ ให้คลิกที่ เพิ่มอุปกรณ์ ระบบจะแสดงข้อมูลมาให้กรอก หลังจากกรอกเสร็จให้กดปุ่ม ตกลง หรือกดปุ่ม ยกเลิก หากไม่ต้องการเพิ่มอุปกรณ์



รูปที่ 4.4 แสดงหน้าจอการเพิ่มอุปกรณ์

ในการใช้งาน จะต้องเปิดโปรแกรมควบคุมในการส่งและรับค่าระหว่างเครื่องเซิร์ฟเวอร์กับตัวไมโครคอนโทรลเลอร์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงโปรแกรมที่ใช้ในการส่งและรับค่าระหว่างเครื่องเซิร์ฟเวอร์กับไมโครคอนโทรลเลอร์

4.5 ส่วนการทำงานและควบคุมระหว่างคอมพิวเตอร์ กับอุปกรณ์ไฟฟ้า

ในการศึกษา และทดลองนี้ ตั้งแต่การเชื่อมต่อไมโครคอนโรเลอร์กับ อุปกรณ์ไฟฟ้า และเชื่อมต่อระหว่างคอมพิวเตอร์ โดยจะแบ่งออกเป็น 3 ส่วน คือ

การทดลองที่ 1: การเชื่อมต่อ และส่งข้อมูลระหว่าง Computer และอุปกรณ์ไฟฟ้า ผ่านทาง XBee

การทดลองที่ 2: การสั่งการอุปกรณ์ไฟฟ้าด้วยอินฟาเรด (Infrared)

การทดลองที่ 3: การควบคุม, ตรวจสอบอุปกรณ์ไฟฟ้าโดยตรง และส่งสถานะกลับไปหาคอมพิวเตอร์

การทดลองที่ 1 การเชื่อมต่อ และส่งข้อมูลระหว่าง Computer และอุปกรณ์ไฟฟ้า ผ่านทาง XBee

วัตถุประสงค์

- เพื่อให้คอมพิวเตอร์ สามารถเชื่อมต่อกับอุปกรณ์ไฟฟ้าผ่านทางคอมพิวเตอร์ โดยผ่านทาง XBee

สมมุติฐาน

- คอมพิวเตอร์สามารถสั่งการอุปกรณ์ไฟฟ้า โดยใช้ XBee ได้

ขั้นตอนการทดลอง

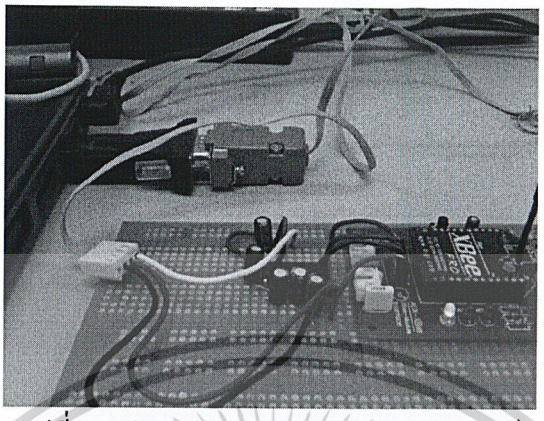
ขั้นตอนนี้ จะแบ่งออกเป็น 2 ส่วนย่อย ประกอบด้วย

- ส่วนส่งสัญญาณจากคอมพิวเตอร์
- ส่วนรับสัญญาณจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่วนส่งสัญญาณจากคอมพิวเตอร์

ในส่วนนี้จะทำหน้าที่ส่งสัญญาณจากคอมพิวเตอร์ผ่านไปยัง XBee



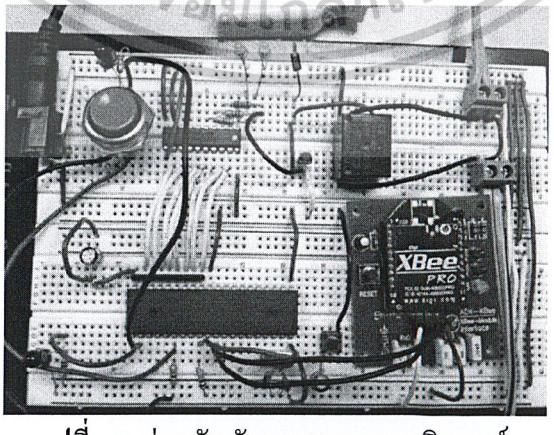
รูปที่ 4.6 ส่วนตัวส่งสัญญาณจากคอมพิวเตอร์

อุปกรณ์และแผงวงจรที่เกี่ยวข้อง

- MAX232
- XBee PRO
- ADX-XBee (ฐานรอง XBee)
- สายซีเรียลพอร์ต (ในที่นี้ใช้ RS-232)
- โปรแกรม X-CTU ให้สำหรับกำหนดค่าและรูปแบบให้ XBee

2. ส่วนรับสัญญาณจากคอมพิวเตอร์

ส่วนนี้จะเป็นที่เชื่อมต่อกับอุปกรณ์ไฟฟ้า ประกอบด้วยตัวรับสัญญาณ XBee และแผงวงจรที่เชื่อมต่อกับอุปกรณ์ไฟฟ้า โดยมีวงจรดังนี้



รูปที่ 4.7 ส่วนรับสัญญาณจากคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์และแผงวงจรที่เกี่ยวข้อง

- MCS51
- MAX232
- XBee PRO
- ADX-XBee
- Relay
- Diode
- ปลั๊กตัวผู้
- ปลั๊กตัวเมีย
- ตัวต้านทาน (Resister)
- 74HC54 (Buffer IC)

เมื่อต่อวงจรทุกอย่างเสร็จแล้ว ก็เริ่มทำการทดลอง โดยมีวางตำแหน่งของฝั่งส่งสัญญาณและฝั่งรับสัญญาณให้ห่างกัน จากนั้น ก็เริ่มสั่งการผ่านทางเว็ปโดนเริ่มต้นจากปิดไฟทั้งหมด

รูปที่ 4.8 รูปของส่วนรับสัญญาณโดยเริ่มต้นที่ปิดไฟทั้งหมด

จากนั้นก็ให้สั่งการผ่านเว็ปโดยทดลองเปิดไฟทั้งหมด ทั้ง 2 เครื่อง และต่อมาทดลองโดยเปิดเพียงแค่เครื่องเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิเคราห์และสรุปผลการทดลอง

จากการทดลองที่ 3 ซึ่งพัฒนามาจาการทดลองที่ 1 และ 2 สามารถสั่งการอุปกรณ์ไฟฟ้าจากคอมพิวเตอร์ โดยให้ XBee ได้ ตามสมมุติฐานที่ตั้งเอาไว้ โดยการที่ใช้ XBee เพิ่มขึ้นมา ทำให้สามารถสั่งการอัปกรณ์ไฟฟ้าในระยะที่ไกลได้

รูปที่ 4.9 ผลการทดลองจากการสั่งการโดยเว็ป โดยเปิดไฟทั้ง 2 เครื่อง

รูปที่ 4.10 ผลการทดลองจากการสั่งโดยเว็ป โดยเปิดไฟเพียงเครื่องเดียว

การทดลองที่ 2 การสั่งการอุปกรณ์ไฟฟ้าด้วยอินฟาเรด (Infrared)

วัตถุประสงค์

- ทำให้สามารถควบคุมอุปกรณ์ผ่านอินฟราเรดได้

สมมุติฐาน

- เมื่อมีวัตถุมากั้นขวานระหว่าง ตัวอินฟราเรด และตัวส่งอินฟราเรด อุปกรณ์ไฟฟ้า จะถูกเปลี่ยนสถานะ

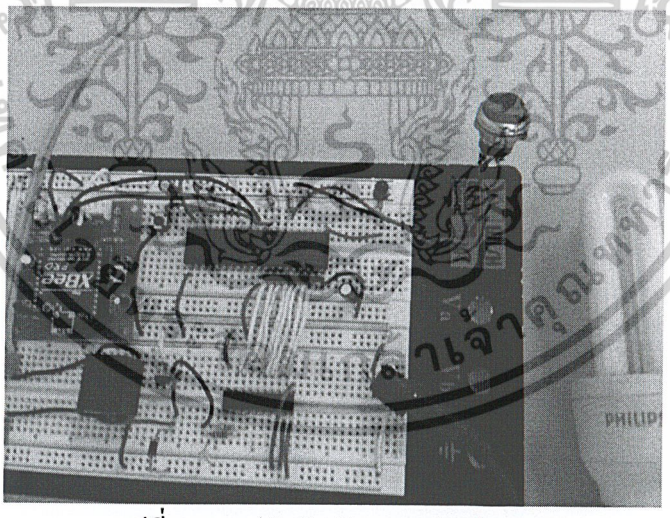
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนนี้ เป็นการพัฒนาต่อมาจากการทดลองที่ 1 โดยจากเดิม โดยทดลองติดตั้งสัญญาณอินฟาเรด เพิ่มขึ้นมา หลักการของอินฟาเรดคือ เมื่อฝั่งส่ง ส่งสัญญาณไปหาฝั่งรับได้ จะมีค่า logic เป็น 1 ตลอดการส่ง แต่ถ้าหากสัญญาณถูกบัง หรือถูกขวาง ค่า logic จะถูกเปลี่ยนเป็น 0 หลักการเบื้องต้นนี้ สามารถนำมาประยุกต์ใช้ในการทำงานในแบบต่างๆได้มากมาย เช่นการตรวจสอบการเข้า ออกจากห้องโดยอินฟาเรด โดยเมื่อมีคนเดินผ่านประตู ก็ทำการสั่งให้เครื่องใช้ไฟฟ้าทำงานได้

อุปกรณ์ที่ใช้ในการทดลอง

- ตัวส่งอินฟาเรด
- ตัวรับอินฟาเรด
- Resistor
- ถ่านไฟฉายขนาด AA

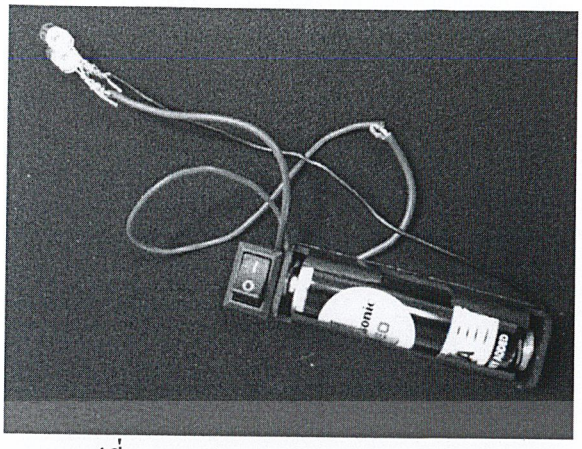
การติดตั้ง จะเพิ่มเติมมาจากการทดลองที่ 3 ในส่วนการรับสัญญาณจากคอมพิวเตอร์ ซึ่งเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ไฟฟ้าอยู่ ตัวรับอินฟาเรดด้านหนึ่งต่อกับ MCS51 เข้าที่ Port 1 โดยผ่าน Resistor เพื่อป้องกันกระแสที่อาจจะแรงเกินไป จนทำให้อินฟาเรดเสียหายได้ อีกขาหนึ่งต่อเข้ากับกราวด์



รูปที่ 4.11 รูปตัวรับสัญญาณอินฟราเรด

ต่อมาจะเป็นในส่วนของตัวส่งอินฟาเรด วิธีอย่างง่ายคือการทดลองเอาตัวส่งอินฟาเรดต่อเข้ากับรางถ่านไฟฉาย AA ก่อน จากนั้นจึงใส่ถ่านลงไป (ไม่สามารถนำขาอินฟาเรดต่อกับถ่านไฟฉาย AA ได้โดยตรง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

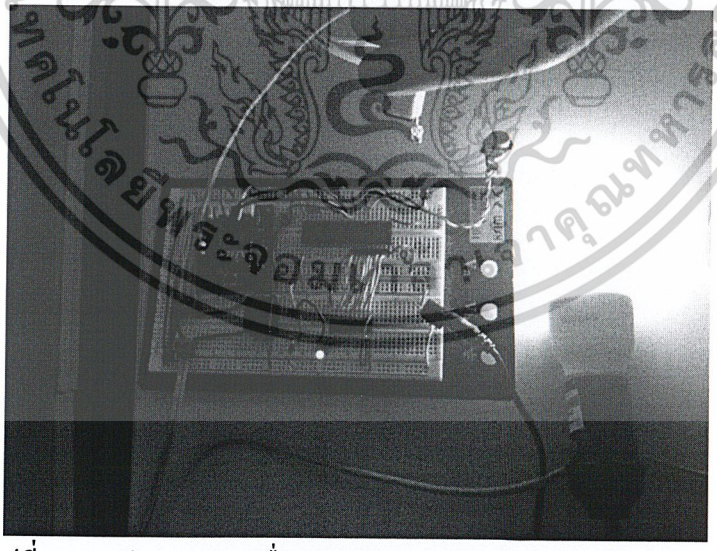


รูปที่ 4.12 รูปตัวส่งสัญญาณอินฟราเรด

การทดลอง เริ่มโดยการตั้งตัวรับอินฟาเรด และตัวส่งอินฟาเรดให้ตรงกัน และห่างกันพอสมควร โดยโปรแกรมที่ใส่เข้าไปใน MCS51 จะกำหนดให้ อุปกรณ์จะเปลี่ยนสถานะทุกครั้งที่มีสัญญาณอินฟาเรดจากตัวส่งผ่านเข้ามายังตัวรับ

วิเคราะห์และสรุปผลการทดลอง

จากการทดลองพบว่า สามารถเปลี่ยนสถานะของอุปกรณ์ไฟฟ้าจากอินฟาเรดได้ แต่ยังพบปัญหาบ้างเช่น สัญญาณอินฟาเรดอ่อนลง เพราะพลังงานการส่งไม่พอ ทำให้การสั่งการทำได้ยากขึ้น



รูปที่ 4.13 รูปผลทดลองเมื่อนำวัตถุมาวางขวางสัญญาณอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3 การควบคุม, ตรวจสอบอุปกรณ์ไฟฟ้าโดยตรง และส่งสถานะกลับไปหาคอมพิวเตอร์

วัตถุประสงค์

- เพื่อทำให้ส่วนของอุปกรณ์ไฟฟ้า สามารถส่งสถานะกลับมายังคอมพิวเตอร์ และเปลี่ยนแปลงสถานะของอุปกรณ์ไฟฟ้า ในกรณีถูกใช้งานโดยตรงได้

สมมุติฐาน

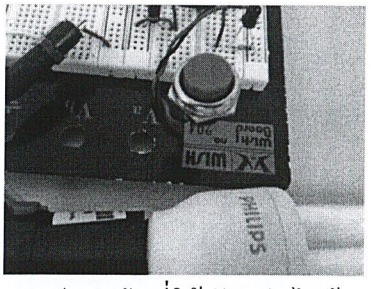
- เมื่อมีการกดปุ่มเปิด-ปิดที่อุปกรณ์ อุปกรณ์จะเปลี่ยนสถานะได้ถูกต้องและสส่งสาถนะที่ถูกต้องนั้น กลับไปยังคอมพิวเตอร์

ตอนนี้เราสามารถควบคุมอุปกรณ์ไฟฟ้าจากคอมพิวเตอร์ได้แล้ว แต่ในความเป็นจริงการเปิด-ปิดอุปกรณ์ไฟฟ้า ก็อาจจะถูกเปิด-ปิดโดยตรงได้เช่นกันทำให้เกิดปัญหาคือ สถานะที่แสดงในคอมพิวเตอร์กับอุปกรณ์จริงไม่ตรงกัน การแก้ปัญหาก็คือ ต้องมีการตรวจสอบสถานะ และส่งสถานะกลับไปยังคอมพิวเตอร์ ส่วนกรณีการเปิด-ปิดอุปกรณ์ไฟฟ้าโดยตรง วิธีหนึ่งที่จะตรวจสอบได้ คือการวัดกระแสจากอุปกรณ์ไฟฟ้า ซึ่งค่อนข้างวุ่นวาย อีกทั้งลักษณะทางกายภาพของอุปกรณ์ไฟฟ้าเอง ก็ไม่สามารถดัดแปลงไปจากเดิมได้มากนัก จึงมีอีกวิธีคือ การสร้างสวิตซ์ขึ้นมาใหม่ ซึ่งทำหน้าที่แทนสวิตซ์จริง โดยเครื่องใช้ไฟฟ้า จะให้เปิดตลอดเวลา แต่จะถูกควบคุมโดย Relay และ MCS51 ผ่านสวิตซ์ใหม่ที่ถูกสร้างขึ้นมา

อุปกรณ์ที่ใช้ในการทดลอง

- สวิตซ์
- ตัวต้านทาน (Resistor)

ขั้นตอนการพัฒนาคือ ส่วนของสวิตซ์ด้านที่ต่อกับ MCS51 Port 1 และต่อกับ Resistor ส่วนอีกด้านต่อเข้ากราวด์ ส่วนการตรวจสอบการทำงาน และตัดต่อวงจร จะอยู่ในส่วนของ MCS51 ทั้งหมด เมื่อมีการเปลี่ยนสถานะจากอุปกรณ์ไฟฟ้าผ่านสวิตซ์นี้จะทำการควบคุมและส่งสถานะใหม่กลับไปยังคอมพิวเตอร์ได้



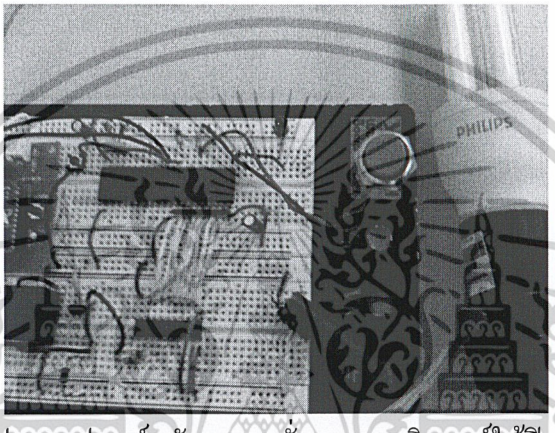
รูปที่ 4.14 รูปของปุ่มที่ใช้เปิด-ปิดไฟฟ้าโดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทดลอง โดยเริ่มจากสั่งการอุปกรณ์ไฟฟ้าผ่านอินเตอร์เน็ต โดยให้ปิดไฟก่อน จากนั้น ให้กดปุ่มเพื่อเปิดอุปกรณ์ไฟฟ้าโดยตรง เพื่อให้หลอดไฟติด

วิเคราะห์และสรุปผลการทดลอง

จากการทดลองทำให้เราสามารถควบคุมอุปกรณ์ไฟฟ้าได้โดยตรง และส่งสถานะกลับไปยังคอมพิวเตอร์ และกลับไปยังอินเตอร์เน็ตได้ ซึ่งทำให้สามารถแก้ไขปัญหาเรื่องสถานะที่แสดงในคอมพิวเตอร์และ อุปกรร์ไฟฟ้าจริงไม่ตรงกันได้



รูปที่ 4.15 รูปของอุปกรณ์หลังจากถูกสั่งจากคอมพิวเตอร์ให้ปิดการทำงาน

รูปที่ 4.16 รูปของผลจากการกดปุ่มสั่งการอุปกรณ์โดยตรง โดยเปิดอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลโครงงาน

ในการศึกษาและพัฒนาระบบควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ต (Electrical Equipment via Internet) โดยศึกษาจากแนวทางการทำงานร่วมกันระหว่างอุปกรณ์ฮาร์ดแวร์ (Hardware) และซอฟท์แวร์ (Software) จากนั้นจึงออกแบบวงจรไมโครคอนโทรลเลอร์ (Microcontroller) พร้อมกับพัฒนาแอพพลิเคชั่น และเว็บแอพพลิเคชั่น เพื่อให้ทั้งฮาร์ดแวร์และซอฟท์แวร์สามารถติดต่อสื่อสาร และรับส่งข้อมูลถึงกันได้

จากการที่ได้ศึกษาและพัฒนามา ดังที่ได้กล่าวมาแล้วข้างต้น จะเห็นได้ว่าระบบนี้ทำงานได้ตรงตามที่ต้องการ กล่าวคือ เมื่อมีการเปิดหรือปิดอุปกรณ์ผ่านทางเว็บแอพพลิเคชั่นแล้ว แอพพลิเคชั่นที่ทำงานบนเครื่องเซิร์ฟเวอร์จะส่งข้อมูลไปยังอุปกรณ์เครื่องใช้ไฟฟ้านั้นๆ เพื่อเปิดปิดอุปกรณ์ หากมีการกดเปิดปิดที่สวิตช์ตัวอุปกรณ์ หรือใช้รีโมตอินฟาเรดในการเปิดปิด ตัวไมโครคอนโทรลเลอร์ ที่ต่ออยู่กับอุปกรณ์นั้นๆ จะส่งข้อมูลสถานะเปิดหรือปิดปัจจุบันของอุปกรณ์กลับมายังเครื่องเซิร์ฟเวอร์ เพื่อทำการบันทึกสถานะ พร้อมกับแสดงสถานะปัจจุบันบนเว็บแอพพลิเคชั่น

จากการศึกษาและทดลองทำโครงงานนี้ ทำให้ได้เรียนรู้เกี่ยวกับการทำงานร่วมกันระหว่างฮาร์ดแวร์กับซอฟแวร์ โดยสามารถนำมาออกแบบและพัฒนาระบบควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าผ่านเครือข่ายอินเตอร์เน็ตได้อย่างเหมาะสม และสามารถนำไปใช้ได้จริง ซึ่งระบบนี้จะช่วยให้ชีวิตผู้คนมีความสะดวกสบายมากยิ่งขึ้น โดยเฉพาะผู้ที่เร่งรีบในการไปทำงานนอกบ้าน จะได้ไม่ต้องกังวลว่าได้เปิดเครื่องใช้ไฟฟ้าภายในบ้านทิ้งไว้หรือเปล่า เพราะหากเปิดทิ้งไว้แล้วลืมปิด สามารถที่จะเข้ามาปิดผ่านเว็บแอพพลิเคชั่นได้ตลอดเวลา โครงงานนี้จึงเป็นอีกหนึ่งแนวทางสำหรับผู้ที่ต้องการความสะดวกสบายเหมาะกับวิถีชีวิตในปัจจุบันที่มีแต่ความเร่งรีบอยู่ตลอดเวลา

5.2 ปัญหาและอุปสรรค

5.2.1 ปัญหาด้านฮาร์ดแวร์

5.2.1.1 การต่อวงจรไมโครคอนโทรลเลอร์เข้ากับอุปกรณ์เครื่องใช้ไฟฟ้า เนื่องจากอุปกรณ์เครื่องใช้ไฟฟ้าต้องต่อเข้ากับไฟบ้านซึ่งมีแรงดันไฟฟ้า 220 โวลต์ ในขณะที่วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ใช้แรงดันไฟฟ้า 5 โวลต์ ฉะนั้นในการเชื่อมต่อกันต้องใช้ความระมัดระวังเป็นพิเศษ ห้ามต่อสลับสายหรือห้ามให้สายไฟในส่วนของทองแดงสัมผัสกัน

5.2.1.2 แรงดันไฟฟ้าของวงจรไมโครคอนโทรลเลอร์ โดยตัววงวรจะต้องใช้แรงดันไฟฟ้า 5 โวลต์ ซึ่งแหล่งพลังงานของเราคือไฟบ้านมีแรงดันไฟฟ้า 220 โวลต์ ฉะนั้นจึงต้องทำการแปลงแรงดันไฟฟ้าจาก 220 โวลต์ เป็น 5 โวลต์ ซึ่งถ้าไม่ได้ทำด้วยความระมัดระวังอาจเกิดอันตรายถึงขั้นเสียชีวิตได้

5.2.1.3 ระยะห่างของอุปกรณ์อินฟราเรดระหว่างตัวรับและตัวส่ง ค่อนข้างจะมีระยะสั้น ทั้งยังไม่ค่อยมีความเสถียรและแม่นยำ เมื่อให้ตัวรับและตัวส่งอยู่ห่างกัน ฉะนั้นจึงต้องเพิ่มอุปกรณ์ไมโครคอนโทรเลอร์เพื่อช่วยให้อุปกรณ์อินฟราเรดมีสัญญาณที่ดีขึ้น สามารถส่งได้ในระยะที่ไกลขึ้น มีความแม่นยำมากยิ่งขึ้น

5.2.1.4 การรับส่งข้อมูลระหว่างเครื่องเซิร์ฟเวอร์กับอุปกรณ์ไมโครคอนโทรลเลอร์นั้น จะรับส่งได้เพียง 8 บิต หรือ 1 ตัวอักษรเท่านั้น ฉะนั้นเวลาต้องการรับส่งข้อมูลที่มีหลายตัวอักษร จึงต้องทำการรับส่งทีละ 1 ตัวอักษร

5.2.2 ปัญหาด้านซอฟแวร์

การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ MCS51 ต้องใช้ภาษาแอสเซมบลี (Assembly) ในการเขียน ทำให้ค่อนข้างมีความยุ่งยากในการเขียนพอสมควร เพราะไม่ค่อยรู้คำสั่งของภาษานี้มากเท่าที่ควร ทั้งยังไม่มีโปรแกรมที่ใช้ในการเขียนภาษานี้โดยเฉพาะ หากต้องการให้โปรแกรมทำงานตามที่เราต้องการ ต้องเขียนคำสั่งขึ้นมาเองโดยใช้คำสั่งพื้นฐานที่มีอยู่

นอกจากนี้เมื่อเขียนโปรแกรมเสร็จ เรายังไม่ทราบว่าโปรแกรมที่เขียนขึ้นทำงานได้ตรงตามที่เราต้องการหรือไม่ จนกว่าจะได้เขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์ MCS51 แล้วทดสอบ ซึ่งหากโปรแกรมที่เขียนขึ้นไม่ตรงตามที่เราต้องการ เราต้องกลับมาแก้ไขใหม่ แล้วเขียนโปรแกรมลงบนไมโครคอนโทรลเลอร์ MCS51 ใหม่อีกครั้ง ซึ่งนับว่าค่อนข้างยุ่งยากและเสียเวลาเป็นอย่างมาก เนื่องจากไม่มีโปรแกรมช่วยในการทดสอบคำสั่ง

5.3 ข้อเสนอแนะ

เนื่องจากสัญญาณอินฟาเรดมีระยะค่อนข้างสั้น หากต้องการเพิ่มระยะของสัญญาณอินฟาเรด สามารถต่อส่วนขยายสัญญาณได้ที่ตัวรับสัญญาณอินฟาเรด ซึ่งต้องมีค่าใช้จ่ายเพิ่มเติม ยิ่งต้องการให้สัญญาณอินฟาเรดมีระยะไกลมากเท่าไร ยิ่งต้องใช้ค่าใช้จ่ายที่สูงมากเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ThaiEasyElec.com. 2551. **Zigbee and Xbee BASIC ตอน การใช้งาน Xbee เบื้องต้น**. เข้าถึงได้จาก http://www.thaieasyelec.com/index.php?lay=show&ac=article&Id=538707978&Ntype=7

[2] Rob Faludi, 2008. **Upgrading Firmware for XBee Radios.** [Online]. Available: www.faludi.com/itp_coursework/meshnetworking/XBee/XBee_firmware_upgrade.html

[3] STK@TEE, 2551. **Xbee Basic Configuration in Network Application**. เข้าถึงได้จาก http://www.thaieasyelec.com/Embedded-Electronics-Application/Xbee-Basic-Configuration-in-Network-Application.html

[4] NECTEC. 2551. **Ajax (Asynchronous JavaScript and XML).** เข้าถึงได้จาก http://wiki.nectec.or.th/giti/Knowledge/Ajax

[5] Artificial Motion. 2544. **Microcontroller.** เข้าถึงได้จาก http://www.school.net.th/library/webcontest2003/100team/dlnes137/am/Microcontroller.html

[6] วิกิพิเดีย. 2554. **รังสีอินฟราเรด**. เข้าถึงได้จาก http://th.wikipedia.org/wiki/รังสีอินฟราเรด

[7] Embed4u. 2010. **Relay Working Principle.** [Online].Available: www.embed4u.com/?p=380

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

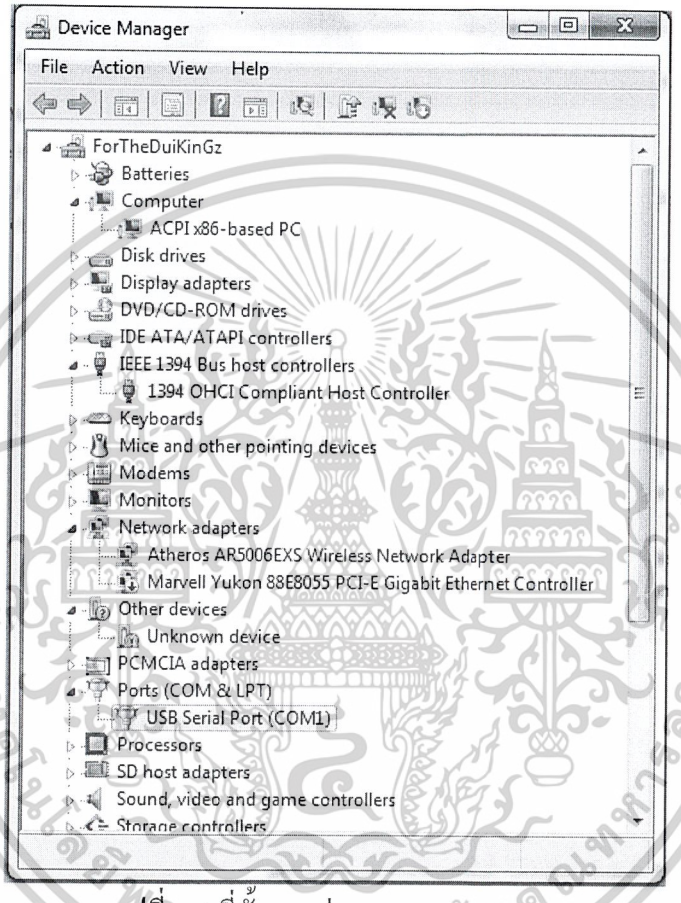
ภาคผนวก ก.

การติดตั้ง และเซตค่าเพื่อเขียนโปรแกรมลงไป MCS51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการติดตั้ง Driver และตั้งค่า Port ให้กับ USB to RS232 (Serial Port)

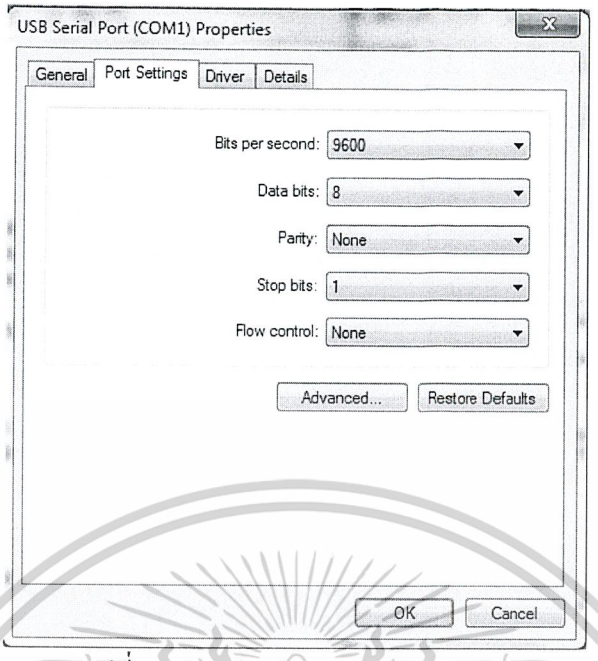
1. เชื่อมต่อ USB to RS232 เข้ากับคอมพิวเตอร์
2. ติดตั้ง Driver Z-TEK USB to Serial ลงไป
3. จากนั้นเข้าไปที่ My Computer -> Properties -> Device Manager -> COM (COM & LPT)



รูปที่ ก.1 ที่ตั้งของส่วน USB Serial Port

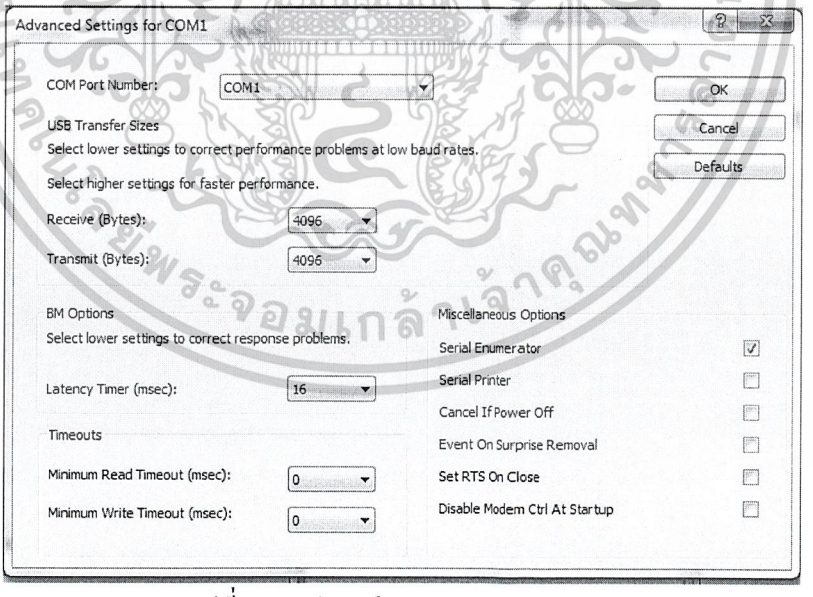
4. คลิ๊กขวาที่ USB Serial Port เลือกที่ Properties
5. เลือกที่ Tab Port Setting ที่ Bits Per Second ให้เลือกค่าตามอุปกรณ์ที่จะเชื่อมต่อกับ Serial Port (ในที่นี้ จะเลือกที่ 9600) จากนั้นให้กดที่ Advanced

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.2 ภาพในส่วนของ Port Setting

6. เมื่อเข้ามาแล้ว ให้เลือกที่ Com Port Number ให้เลือกเป็น COM1 แล้วกด OK จึงเสร็จวิธีการติดตั้ง



รูปที่ ก.3 รูปภาพใน Advanced

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

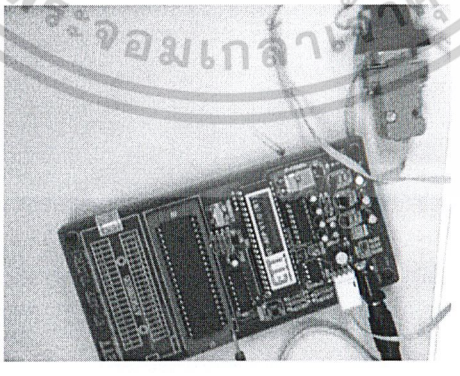
วิธีการเขียนภาษา Assembly เข้าไปใน MCS51

1. ทำการเขียนภาษา Assembly ลงใน Notepad
2. เมื่อเขียนเสร็วแล้ว ให้ Save As และเปลี่ยนนามสกุลเป็น .asm และเก็บเอาไว้ในที่เดียวกับ SXA51.exe
3. มาเลือกที่ RUN-> พิมพ์ CMD เพื่อเรียก Command Line ออกมา
4. ทำการเปลี่ยน Path ใน Command Line ให้ไปอยู่ที่เดียวกับ Folder เดียวกับ SXA51.exe
5. ทำการเรียกใช้ SXA51.exe แล้วเว้นวรรค ต่อด้วยโปรแกรม .asm ที่ต้องการสร้าง HEX File ออกมา เช่น

   C:\SXA51\SXA51.exe test01.asm ซึ่ง HEX File จะอยู่ในที่เดียวกับ SXA51.exe

รูปที่ ก.4 ภาพหลังจากเรียก SXA51.exe เพื่อสร้าง test01.asm ให้เป็น TEST01.HEX

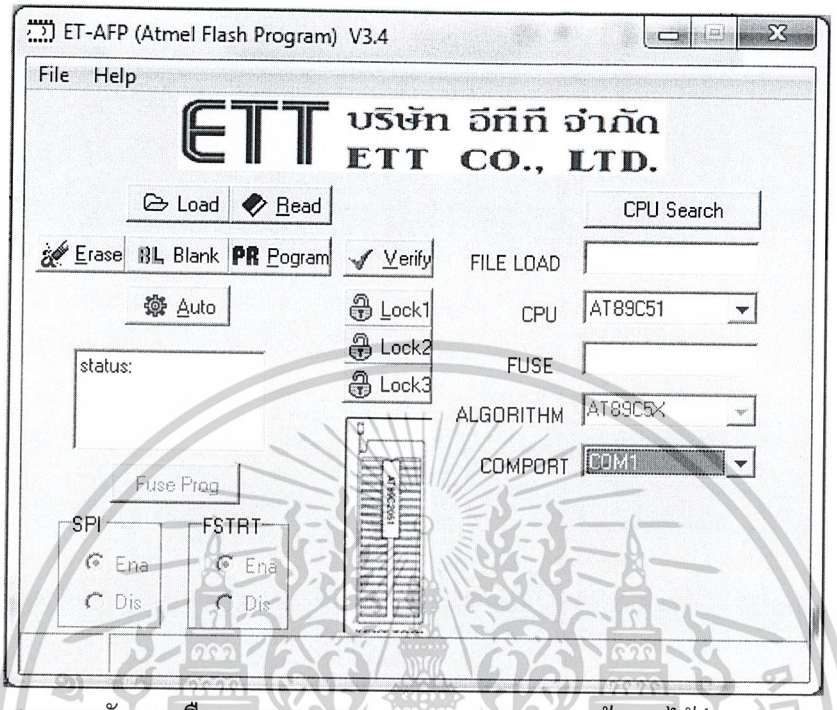
6. เชื่อมต่อ USB to RS232 เข้ากับ ET-AFP V 1.0 ที่ใส่ MCS51 ไว้แล้ว จากนั้นเปิดโปรแกรม AFX34 ขึ้นมา



รูปที่ ก.5 ภาพของ ET-AFP V 1.0 ที่เชื่อมต่อกับ USB to RS232

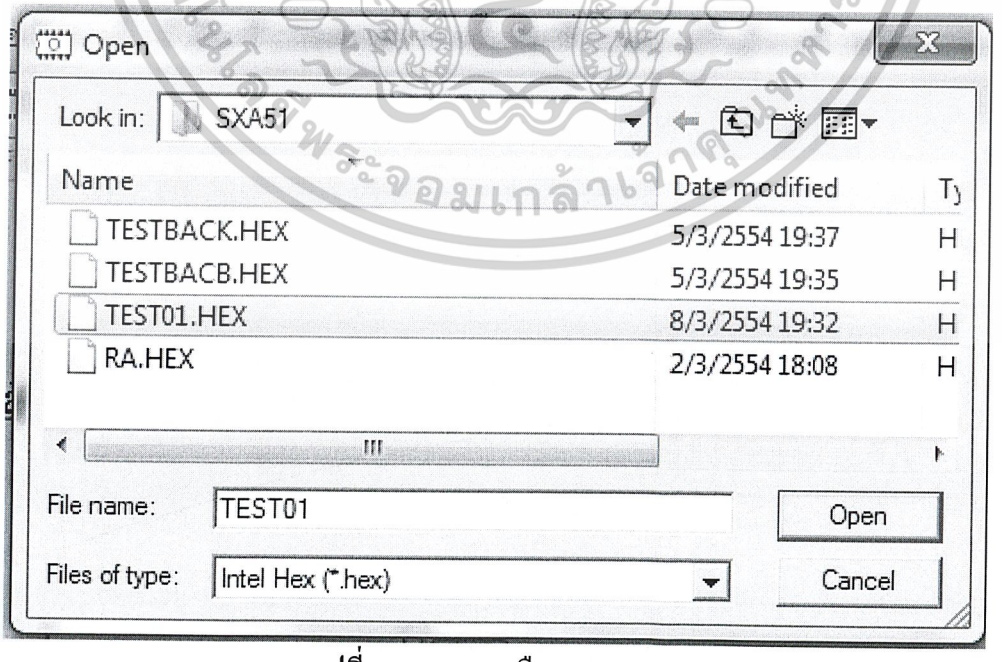
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เมื่อเปิดมาแล้ว ให้เลือกที่ COMPORT จากนั้นเลือกที่ COM1 เสร็จแล้วกดที่ CPU Search เพื่ออ่านค่ารุ่นของ IC ออกมา



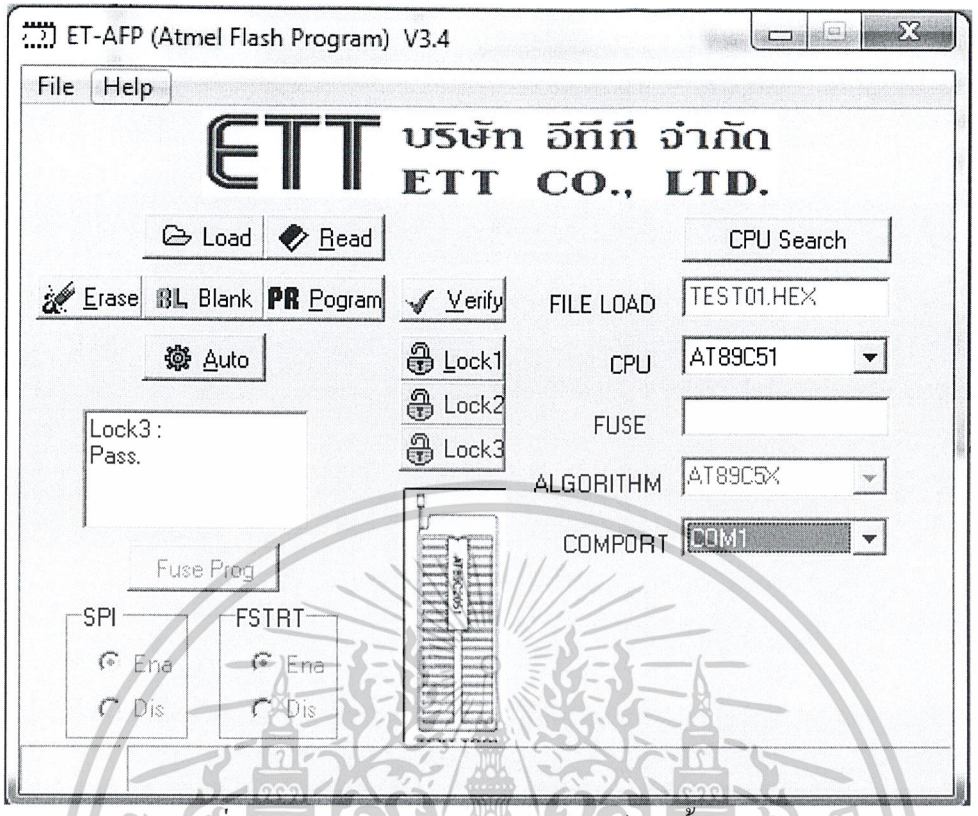
รูปที่ ก.6 ภาพหลังจากเลือก COMPORT และ CPU Search แล้ว จะได้รุ่นของ IC ออกมา

8. ให้กดที่ Load แล้วเลือก HEX File ที่ได้สร้างจาก SXA51 ออกมา แล้วกด Auto จึงเสร็จสิ้นวิธีใช้งาน



รูปที่ ก.7 ภาพการเลือก HEX File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.8 ภาพหลังจาก Auto แล้ว ถ้าสำเร็จจะขึ้นว่า Pass

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

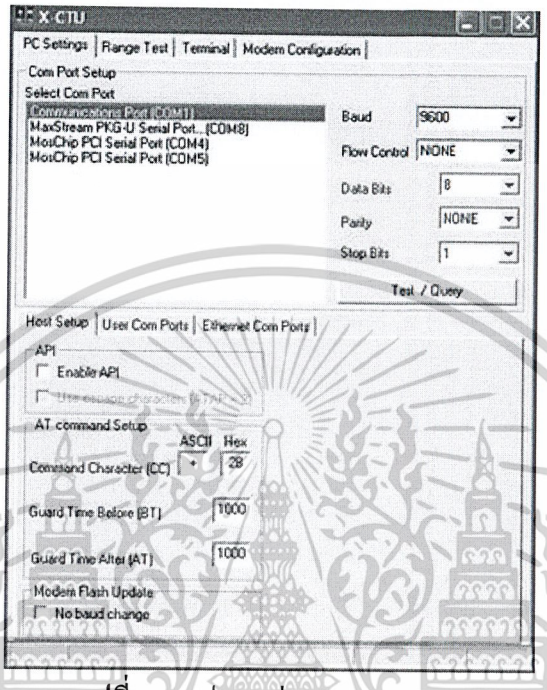
ภาคผนวก ข

การตั้งค่าและใช้งาน XBee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

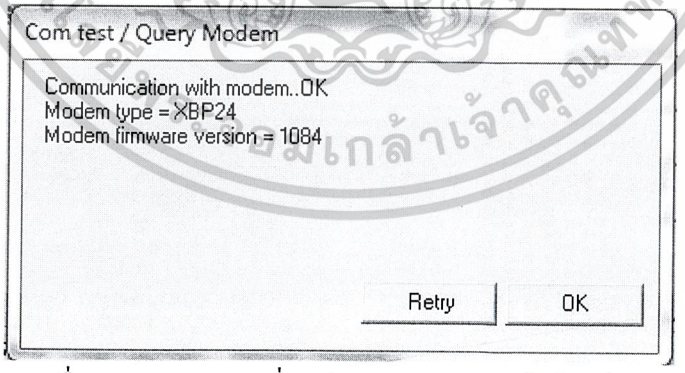
การตั้งค่า XBee เบื้องต้น

1. ทำการต่อ XBee เข้ากับเครื่องประมวลผล และต่อเข้ากับคอมพิวเตอร์
2. เปิดโปรแกรม X-CTU จากนั้นให้เลือก COM Port ที่เชื่อมต่อกับ XBee อยู่



รูปที่ ข.1 รูปภาพส่วน PC Settings

3. ให้กดที่ Test/Query เพื่อเป็นการทดสอบว่า XBee เชื่อมต่อกับคอมพิวเตอร์หรือไม่ อีกทั้งยังบอกถึงรุ่น และ Firmware ที่ใช้งาน



รูปที่ ข.2 แสดงภาพเมื่อ XBee สามารถเชื่อมต่อกับคอมพิวเตอร์ได้สำเร็จหลังจาก Test/Query

4. ในส่วนหน้า PC Settings นี้ จะเป็นหน้าเกี่ยวกับการตั้งค่า Com Port ต่างๆ ซึ่งต้องปรับให้ตรงกับ XBee ในที่นี้ จะตั้งค่าไว้ดังนี้
   - Buad = 9600
   - Flow Control = NONE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Data Bits = 8
- Parity = NONE
- Stop Bits = 1

5. ต่อไปให้เลือกที่แท็บ Modern Configuration ส่วนนี่จะเป็นส่วนที่กำหนดค่าต่างๆใน XBee ให้กดที่ Read เพื่อทำการ Load ข้อมูลในตัว XBee ออกมา
6. เมื่อกำหนดค่าเสร็จแล้ว ให้กดที่ Write จึงเสร็จสิ้นการตั้งค่า

หากต้องการตั้งค่าใน XBee จะต้องทราบความหมายของค่าแต่ละตัว ค่าที่ใช้ในการตั้งค่า XBee ให้ใช้งานได้จริงที่เราต้องทราบมีดังนี้

- รีจิสเตอร์ ID (PAN ID/Personal Area Network): ใช้กำหนดเลขหมายเครือข่าย โดยเลือกค่าตั้งแต่ 0x0000 ถึง 0xFFFE โดยไม่สามารถส่งข้ามเครือข่ายอื่นได้ (ในที่นี้จะให้ ID = 100 ซึ่ง XBee ทุกตัวที่จะใช้งานในเครือข่ายนี้ ต้องให้ ID เป็น 100 ทั้งหมด)
- รีจีสเตอร์ MY (16-bit source Address) : ใช้กำหนด Address 16 bit ของแต่ละโมดูล ตั้งค่าได้ตั้งแต่ 0x0000 ถึง 0xFFFD ถ้าตั้งค่าเป็น 0xFFFF จะเป็นการ
- ฟังก์ชั่น CE (Coordinator Enable) : ใช้กำหนดโหมดการทำงานของ XBee โดยโหมดการทำงานของ XBee จะแบ่งออกเป็น 2 โหมด
  - Coordinator (CE=1) เปรียบเสมือนศูนย์กลางของเครือข่าย
  - End Device (CE=0) เป็นตัวรับ

  ในเครือข่าย จะต้องมี Coordinator และ End Device อย่างน้อย 1 คู่
- รีจีสเตอร์ DH และ DL (Destination High/Low) : ใช้กำหนด Address ของโมดูลตัวรับ โดยกำหนดให้ DH เป็น 0x0000 และ DL เป็นค่า MY ของ Module ตัวรับ ถ้ากำหนดเป็น 0xFFFF ซึ่งหมายความว่า ให้ส่งไปหาทุกตัวที่อยู่ใน ID เดียวกัน
- รีจีสเตอร์ SH และ SL (Serial Number High/Low) เป็นตัวเลขที่ระบุลงใน XBee แต่ละตัว เป็นเลขที่มาจากโรงงาน ไม่สามารถแก้ไขได้ สามารถนำมาใช้เป็น Address ได้

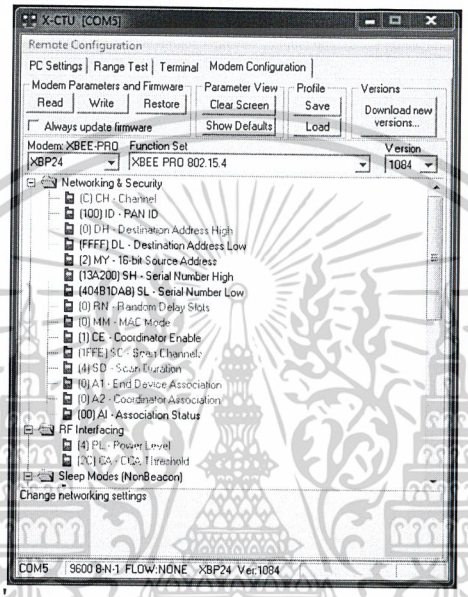
การตั้งค่า XBee ให้เป็น Coordinator

ขั้นตอนการตั้งค่า Coordinator (ในส่วนนี้จะใช้การตั้งค่าที่ใช้ในการทดลองเป็นหลัก)

1. เข้าไปที่ Modem Configuration
2. ให้กดที่ Read เพื่อทำการเรียกข้อมูล XBee ออกมา
3. ตั้งค่า ID โดยถ้าต้องการอยู่ในเน็ตเวิร์คเดียวกัน ต้องตั้งให้ ID ตรงกัน (ในการทดลองใช้ ID = 100)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ตั้งค่า MY (ในการทดลองใช้ 2)
5. ตั้งค่า CE โดยเลือก CE = 1 เพื่อกำหนดให้เป็น Coordinator
6. กำหนด DH และ DL ที่เราต้องการ (ในการทดลอง เราต้องการให้ Coordinator ส่งข้อมูลไปหาลูกข่ายทุกตัว จึงตั้งค่าเป็น 0xFFFF ซึ่งจะให้ผลเหมือน Boardcast)
7. เมื่อตั้งค่าเสร็จ ให้กด Wirte จึงเสร็จสิ้นการตั้งค่า



รูปที่ ข.3 ภาพในส่วนของ Modem Configuration

การตั้งค่า XBee ให้เป็น End Device

ขั้นตอนการตั้งค่า End Device (ในส่วนนี้จะใช้การตั้งค่าที่ใช้ในการทดลองเป็นหลัก)

1. เข้าไปที่ Modem Configuration
2. ให้กดที่ Read เพื่อทำการเรียกข้อมูล XBee ออกมา
3. ตั้งค่า ID โดยถ้าต้องการอยู่ในเน็ตเวิร์คเดียวกัน ต้องตั้งให้ ID ตรงกัน (ในการทดลองใช้ ID = 100)
4. ตั้งค่า MY (ในการทดลองใช้ 1)
5. ตั้งค่า CE โดยเลือก CE = 0 เพื่อกำหนดให้เป็น End Device
6. กำหนด DH และ DL ที่เราต้องการ (ในการทดลอง เราต้องการให้ End Device นี้ สามารถส่งค่ากลับมาหา Coordinator ได้ด้วย วิธีที่ง่ายที่สุด คือการนำค่า SH และ SL ของฝั่ง Coordinator มาใส่ใน DH และ DL ของฝั่งรับ)
7. เมื่อตั้งค่าเสร็จ ให้กด Wirte จึงเสร็จสิ้นการตั้งค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

ตัวอย่างโปรแกรมที่ใช้ในการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างโปรแกรมที่ใช้รับค่าจากคอมพิวเตอร์

ซึ่งรวมถึง การสั่งการอุปกรณ์ไฟฟ้า การรับค่าอินฟราเรด การสั่งอุปกรณ์ไฟฟ้าโดยตรง และการส่งค่ากลับไปยังคอมพิวเตอร์ (โปรแกรมนี้ จะใช้เหมือนกันทุกอุปกรณ์ จะต่างกันที่บางค่าสำหรับ ระบุรหัสของอุปกรณ์ แต่โดยส่วนใหญ่แล้ว จะเหมือนกัน)

```
                ORG     0000H
                device  EQU     30H
                status  EQU     50H
                sensor  EQU     70H
                cycle   EQU     90H
                MOV     device,#41H             ; เก็บชื่ออุปกรณ์ในตัวแปร
device ในที่นี้อุปกรณ์ชื่อว่า A โดยที่ถ้าเป็นที่อุปกรณ์ตัวอื่น ค่าจะเปลี่ยนไป
Start:          MOV     cycle,#00B              ; เก็บค่า 0 ในตัวแปร cycle
ReceivingSetup: MOV     PCON,#00H               ; SMOD=0
                MOV     SCON,#50H               ; Mode=1, Ren=1, TI=0
                MOV     TMOD,#20H               ; Time 1 Mode 2
                MOV     TH1,#0FDH               ; 9600 Baud Rate
                SETB    TR1                     ; เริ่มรับข้อมูล
ReceiveWaiting: JNB     RI,SensorWaiting        ; ตรวจสอบว่า ถ้ายังไม่มี
ข้อมูลเข้ามาให้ไปที่ SensorWaiting
                MOV     B,SBUF                  ; นำข้อมูลที่รับมาเก็บใน
รีจิสเตอร์ B
                CLR     RI                      ; ลบค่าใน RI
                CLR     TR1                     ; ลบค่าใน TR1
                LJMP    IsMeCycle               ; กระโดดไปยัง IsMeCycle
SensorWaiting:  JB      P1.0,SwitchWaiting      ; ตรวจสอบว่า ถ้ายังไม่มี
ข้อมูลเข้ามาให้ไปที่ SwitchWaiting
                LCALL   Delay                   ; เรียกใช้ Delay
                LJMP    IsSensor                ; กระโดดไปยัง IsSensor
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SwitchWaiting:      JB      P1.5,ReceiveWaiting          ; ตรวจสอบว่า ถ้ายังไม่มี
ข้อมูลเข้ามาให้ย้อนกลับไปที่ ReceiveWaiting
                    LCALL   Delay
                    LJMP    Toggle                       ; กระโดดไปยัง Toggle

IsMeCycle:          MOV     A,cycle                      ; เก็บค่าจำนวนรอบในตัวแปร
cycle ลงรีจิสเตอร์ A
                    CJNE    A,#00B,IsSensorCycle         ; ตรวจสอบว่าการรับข้อมูล
รอบนี้ใช่รอบแรกหรือเปล่า ถ้าไม่ใช่ให้กระโดดไปยัง IsSensorCycle

ChkMe:              MOV     A,B
                    CJNE    A,device,Start               ; ตรวจสอบว่าชื่ออุปกรณ์ที่
รับมาใช่ของตัวเองหรือเปล่า ถ้าไม่ใช่ให้กระโดดไปยัง Start
                    MOV     cycle,#01B                   ; เก็บค่า 1 ลงตัวแปร cycle
                    LJMP    ReceivingSetup               ; กระโดดไปยัง
ReceivingSetup

IsSensorCycle:      CJNE    A,#01B,SwitchOff             ; ตรวจสอบว่าการรับข้อมูล
รอบนี้ใช่รอบที่สองหรือไม่ ถ้าไม่ใช่ให้กระโดดไปยัง SwitchOff
                    MOV     sensor,B                     ; เก็บค่าที่รับมาลงในตัวแปร
sensor
                    MOV     cycle,#10B                   ; เก็บค่า 2 ลงตัวแปร cycle
                    LJMP    ReceivingSetup               ; กระโดดไปยัง
ReceivingSetup

SwitchOff:          MOV     A,B
                    CJNE    A,#30H,SwitchOn              ; ตรวจสอบสถานะที่รับเข้ามา
ว่าปิดหรือไม่ ถ้าไม่ให้กระโดดไปยัง SwitchOn
                    MOV     P0,#00000000B                ; กำหนดให้ปิดไฟ
                    MOV     status,#30H                  ; เก็บค่า 0 ลงในตัวแปร status
                    LJMP    Start
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SwitchOn:       MOV   P0,#11111111B        ; กำหนดให้เปิดไฟ
                MOV   status,#31H          ; เก็บค่า 1 ลงในตัวแปร status
                LJMP  Start

IsSensor:       MOV   A,sensor
                CJNE  A,#31H,Start         ; ตรวจสอบว่าได้เปิดให้
เซนเซอร์ทำงานอยู่หรือไม่ ถ้าไม่ให้กระโดดไปยัง Start

Toggle:         MOV   A,status
                CJNE  A,#31H,ToggleOn      ; ตรวจสอบว่าสถานะปัจจุบัน
เปิดอยู่หรือไม่ ถ้าเปิดอยู่ให้กระโดดไปยัง ToggleOn
ToggleOff:      MOV   status,#30H          ; เปลี่ยค่าสถานะปัจจุบันเป็น
0 ซึ่งก็คือปิด
                LCALL SendingSetup         ; เรียกใช้ SendingSetup
                LJMP  Start
ToggleOn:       MOV   status,#31H          ; เปลี่ยค่าสถานะปัจจุบันเป็น
1 ซึ่งก็คือเปิด
                LCALL SendingSetup         ; เรียกใช้ SendingSetup
                LJMP  Start
SendingSetup:   MOV   PCON,#00H            ; SMOD=0
                MOV   SCON,#40H            ; Mode 1, REN=0, TI=0
                MOV   TMOD,#20H            ; Time 1 Mode 2
                MOV   TH1,#0FDH            ; 9600 Baud rate
                SETB  TR1                  ; เริ่มส่งข้อมูล
                MOV   SBUF,device
                LCALL Sending

                SETB  TR1
                MOV   SBUF,status
                LCALL Sending
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Sending:    JNB   TI,Sending      ; ส่งข้อมูลจนกว่าจะส่งเสร็จ
            CLR   TI
            CLR   TR1
            RET
Delay:      MOV   R1,#300H
Delay2:     MOV   R0,#300H
Delay1:     DJNZ  R0,Delay1
            DJNZ  R1,Delay2
            RET
            END
```

ตัวอย่างโปรแกรมที่ใช้ในการรับ – ส่งข้อมูลระหว่าง Server และ Microcontroller

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.IO.Ports;
using System.Threading;
using System.Collections;

namespace CDvI_Control
{
   class Program
   {
      private static ArrayList listDevice;
      private static ArrayList listDeviceNew;
      private static string status;
      private static string statusNew;
      private static string statusNewReceiving;
      private static string[] device;
      private static string[] deviceSending;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private static string deviceReceiving;
        static void Main(string[] args)
        {

            Console.WriteLine("****************************************");
            Console.WriteLine("    Electrical Equipment via Internet    ");
            Console.WriteLine("****************************************");
            Console.WriteLine("");
            Console.WriteLine("");
            Console.Write("Status: Working..");

            SerialPort sp = new SerialPort("COM1", 9600, Parity.None, 8, StopBits.One);
// เชื่อมต่อ Serial port ผ่านทาง COM1, 9600 Baud rate
            sp.Open();

            listDeviceNew = new ArrayList();
            status = "";

            while (true)
            {
                listDevice = FileControl.ReadFile("D:\\_senior project\\CDvI-WEB\\CDvI-
WEB\\data\\devices.txt");  // โหลดข้อมูลของอุปกรณ์ทั้งหมดที่เก็บไว้ในไฟล์
                statusNew = "";

                for (int i = 0; i < listDevice.Count; i++)
                {
                    device = listDevice[i].ToString().Split('|');          // แยกข้อมูลของแต่ละ
อุปกรณ์ด้วยเครื่องหมาย |
                    statusNew += device[0] + device[3] + device[4];        // นำค่ามาเก็บลงในตัว
แปร
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (i + 1 != listDevice.Count)
            {
                statusNew += "|";
            }
        }

        if (!status.Equals(statusNew))                    // ตรวจสอบว่าสถานะอุปกรณ์
ในปัจจุบันเหมือนกับสถานะก่อนหน้าหรือไม่
        {
            status = statusNew;
            deviceSending = status.Split('|');

            for (int i = 0; i < deviceSending.Length; i++)
            {
                sp.Write(deviceSending[i][0].ToString());    // ส่งชื่ออุปกรณ์ผ่านทาง
Serial port
                sp.Write(deviceSending[i][1].ToString());    // ส่งสถานะเซ็นเซอร์ผ่าน
ทาง Serial port
                sp.Write(deviceSending[i][2].ToString());    // ส่งสถานะอุปกรณ์ผ่านทาง
Serial port
            }

        }

        deviceReceiving = sp.ReadExisting();              // รับข้อมูลจาก Serial port

        if (!deviceReceiving.Equals(""))                  // ตรวจสอบว่าค่าที่รับมามีข้อมูล
หรือไม่
        {
            listDeviceNew.Clear();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            for (int i = 0; i < listDevice.Count; i++)
            {
              device = listDevice[i].ToString().Split('|');

              if (deviceReceiving[0].ToString() == device[0].ToString())    // ตรวจสอบ
ชื่ออุปกรณ์
              {
                statusNewReceiving = deviceReceiving[1].ToString();      // นำค่า
สถานะอุปกรณ์ที่รับมาทาง Serial port เก็บลงในตัวแปร
              }
              else
              {
                statusNewReceiving = device[4];
              }

              listDeviceNew.Add(device[0] + "|" + device[1] + "|" + device[2] + "|" +
device[3] + "|" + statusNewReceiving);

            }

            FileControl.WriteFile("D:\\_seniorproject\\CDvI-WEB\\CDvI-
WEB\\data\\devices.txt", listDeviceNew); //  บันทึกข้อมูล
          }


          Thread.Sleep(100);  // หน่วงเวลาในการทำงานรอบต่อไป
        }

        sp.Close();

      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
}
```

ตัวอย่างที่โปรแกรมการอ่านข้อมูล และการบันทึกจากไฟล์

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Collections;
using System.IO;

namespace CDvI_Control
{
   class FileControl
   {
      public static ArrayList ReadFile(string fileName)
      {
         StreamReader reader;
         StreamWriter writer;
         ArrayList line = new ArrayList();
         try
         {
            if (!(File.Exists(fileName)))
            {
               writer = File.CreateText(fileName);
               writer.Close();
            }

            reader = File.OpenText(fileName);
            while (!(reader.EndOfStream))
            {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                line.Add(reader.ReadLine());
            }
            reader.Close();
        }
        catch (Exception e)
        {
        }
        return line;
    }
    public static void WriteFile(string fileName, ArrayList data)
    {
        StreamWriter writer;
        try
        {
            writer = File.CreateText(fileName);
            for (int i = 0; i < data.Count; i++)
            {
                writer.WriteLine(data[i].ToString());
            }
            writer.Close();
        }
        catch (Exception e)
        {
        }
    }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ-นามสกุล	นายศมาพงศ์ มกรารัตต์
วัน เดือน ปีเกิด	21 เมษายน 2532 ที่กรุงเทพฯ
ที่อยู่	676 ซอยกรุงเทพ-นนท์บุรี 13 เขตบางซื่อ
	ถนน กรุงเทพ-นนทบุรี กรุงเทพฯ 10800
ประวัติการศึกษา	2553 วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชื่อ-นามสกุล	นายทศพล รอดความทุกข์
วัน เดือน ปีเกิด	3 กุมภาพันธ์ 2532 ที่กรุงเทพฯ
ที่อยู่	71/16 หมู่ 11 ซอยธนสิทธิ์ ตำบลบางปลา
	อำเภอบางพลี จังหวัดสมุทรปราการ 10540
ประวัติการศึกษา	2553 วิทยาศาสตรบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ
	สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้