ระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์

The Air Condition Control Wifi System On the Website

นายจตุภูมิ เกิดประกอบ
นายสุริยา นิลเปี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏนครปฐม
ปีการศึกษา 2557

ระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์

นายจตุภูมิ เกิดประกอบ
นายสุริยา นิลเปี่ยม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
สาขาวิชาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม คณะวิทยาศาสตร์และเทคโนโลยี
มหาวิทยาลัยราชภัฏนครปฐม
ปีการศึกษา 2557

The Air Condition Control Wifi System On the Website

Mr.Jatupoom Kerdprakob
Mr.Suriya Ninpiam

This Project Report Submitted in Partial Fulfillment of the Requirement
For Degree Of Bachelor of Science Program
Program in Industrial Computer Technology
Faculty of Science and Technology
Nakonpathom Rajabhat University
Academic Year 2014

ชื่อเรื่องปริญญานิพนธ์	ระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์ The Air Condition Control Wifi System On the Website
โดย	นาย จตุภูมิ เกิดประกอบ นาย สุริยา นิลเปี่ยม
สาขาวิชา	เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
อาจารย์ที่ปรึกษา	อาจารย์ดร.บุญธง วสุริย์ อาจารย์ขนิษฐา แซ่ลิ้ม
ปีการศึกษา	2557

สาขาวิชาเทคโนโลยีคอมพิวเตอร์อุตสาหกรรม คณะวิทยาศาสตร์และเทคโนโลยี

มหาวิทยาลัยราชภัฏนครปฐม

อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต

.....

(รองศาสตราจารย์ ดร.ปิยะ โควินท์ทวีวัฒน์)

คณบดีคณะวิทยาศาสตร์และเทคโนโลยี

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ

(อาจารย์วีระศักดิ์ ชื่นตา)

.....กรรมการ

(อาจารย์อรรถพล พลานนท์)

.....กรรมการ

(อาจารย์บพิตร ไชยนอก)

.....อาจารย์ที่ปรึกษา

(อาจารย์ ดร.บุญธง วสุริย์)

ชื่อเรื่องปริญญานิพนธ์ ระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์
The Air Condition Control Wifi System On the Website
โดย นายจตุภูมิ เกิดประกอบ
สาขาวิชา นายสุริยา นิลเปี่ยม
อาจารย์ที่ปรึกษา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
อาจารย์ดร.บุญธง วสุริย์
ปีการศึกษา อาจารย์ขนิษฐา แซ่ลิ้ม
2557

บทคัดย่อ

ปริญญานิพนธ์นี้ มีวัตถุประสงค์เพื่อสร้างระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บซึ่งสามารถควบคุมผ่านหน้าเว็บคอลโทลได้จริงเนื่องจากในปัจจุบันอินเตอร์เน็ตได้เข้ามามีบทบาทต่อชีวิตประจำวันของผู้คนที่อาศัยอยู่ในเมืองนักเรียนนักศึกษาคนทำงานต่างก็ใช้ระบบอินเตอร์เน็ตกันเช่นใช้เป็นตัวช่วยในการศึกษาและหาแหล่งของมูลติดต่อเพื่อนและติดต่องานเนื่องจากระบบอินเตอร์เน็ตเป็นระบบที่สามมารถใช้ติดต่อสื่อสารถึงกันได้อย่างรวดเร็วและครอบคลุมพื้นที่อย่างกว้างขวางใช้งานระบบอินเตอร์เน็ตได้จากทุกที่โดยที่ใช้ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติใช้ในการเก็บข้อมูลของรีโมทเครื่องปรับอากาศประเภท Intronics Digital Room Thermostatและใช้แผงวงจรประเภท ET-BASE AVR EASY328 เป็นตัวเชื่อต่อผ่านรีโมทรีโมทเครื่องปรับอากาศ ไปยังสัญญาณอินเตอร์เน็ตโดยมีจุดเชื่อมต่อไร้สายเป็นตัวกลางในการรับส่งสัญญาณโดยควบคุมผ่านหน้าจอเว็บที่สร้างขึ้นมาเพื่อควบคุมโดยเฉพาะ

จากาการทดสอบระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บสามารถใช้งานได้จริงทั้งการควบคุมผ่านแบบมีสายโดยใช้สายสัญญาณไขว้สายและการควบคุมไร้สายผ่านจุดเชื่อมต่อไร้สายผลการทดลองทั้งสองแบบสามารถควบคุมเครื่องประอากาศได้อย่างมีประสิทธิภาพ

กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้เป็นอย่างดีด้วยความกรุณาจากอาจารย์ที่ปรึกษาอาจารย์ ดร.บุญธง วสุริย์ อาจารย์ ขนิษฐาแซ่ลิ้ม ที่ได้สละเวลาอันมีค่ายิ่งที่คอยให้คำปรึกษาและได้ให้คำแนะนำด้านเอกสาร

สำหรับคุณงามความดีอันใดเกิดจากปริญญานิพนธ์ฉบับนี้คณะผู้จัดทำขอมอบให้กับบิดามารดา และญาติพี่น้อง ซึ่งเป็นที่รักและเคารพยิ่ง ซึ่งคอยให้กำลังใจเสมอมาตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสาทวิชาความรู้และถ่ายทอดประสบการณ์ที่ดีให้แก่คณะผู้จัดทำรวมทั้งมหาวิทยาลัยราชภัฏนครปฐม ซึ่งเป็นสถานที่ที่ให้การศึกษาแก่คณะผู้จัดทำ คณะผู้จัดทำขอขอบพระคุณทุกท่านเป็นอย่างยิ่ง ณ โอกาสนี้ ด้วยความจริงใจ

จตุภูมิ เกิดประกอบ
สุริยา นิลเปี่ยม

สารบัญ

	หน้า
บทคัดย่อ	ง
กิตติกรรมประกาศ	จ
สารบัญตาราง	ฉ
สารบัญภาพ	ซ
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงงาน	1
1.2 วัตถุประสงค์ของโครงงาน	2
1.3 ขอบเขตของโครงงานวิจัย	2
1.4 ขั้นตอนการวิจัย	2
1.5 แผนการดำเนินงาน	2
1.6 เครื่องมือที่ใช้วิจัย	4
1.7 ประโยชน์ที่คาดว่าจะได้รับ	4
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	5
2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)	5
2.2 ทฤษฎีเกี่ยวกับรีโมทเครื่องปรับอากาศ	16
2.3 อุปกรณ์รับส่งข้อมูล	17
2.4 สร้างเว็บไซต์เพจแบบเสมือนจริง (Adobe Dreamweaver CS5)	20
2.5 การออกแบบฐานข้อมูล (Data Base Design)	21
บทที่ 3 ขั้นตอนการดำเนินงาน	23
3.1 การดำเนินงานส่วนตัวเครื่อง	23
3.2 กระบวนการทำงานของระบบ	31

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการดำเนินงาน	46
4.1 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์(การกดผ่านเว็บไซต์ควบคุมแบบไร้สาย)	46
4.2 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์ แบบสายสัญญาณไขว้สาย (Cross Over Cable)	47
4.3 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์(การกดผ่านเว็บไซต์ควบคุมแบบไร้สาย)	49
4.4 สรุปผลการทดลอง	50
บทที่ 5 สรุปผลและข้อเสนอแนะ	52
5.1 สรุปผลการทดลอง	52
5.2 สรุปผลการดำเนินงาน	52
5.3 ปัญหาและข้อจำกัด	52
5.4 แนวทางการนำไปใช้	53
5.5 ข้อเสนอแนะ	53
บรรณานุกรม	54
ภาคผนวก ก คู่มือการใช้งานระบบ	55
ภาคผนวก ข ชุดคำสั่งการควบคุม	61

สารบัญภาพ

รูปที่	หน้า
2.1 โครงสร้างทั่วไปของไมโครของไมโครคอนโทรลเลอร์	5
2.2 โครงสร้างของแผงวงจรไมโครคอนโทรลเลอร์อุปกรณ์ต่อพ่วง	7
2.3 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51	8
2.4 โครงสร้างไมโครคอนโทรลเลอร์ประเภทควบคุมแรงดันไฟฟ้าอัตโนมัติ	9
2.5 แผงวงจร ET-BASE AVR EASY328	10
2.6 แผงวงจร ET-BASE MEGA128	11
2.7 ประเภทไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ	12
2.8 เครื่องโปรแกรมแบบ ISP (In-System Programming)	14
2.9 โครงสร้างไมโครคอนโทรลเลอร์อาดุยโน่	14
2.10 โครงสร้างโปรแกรมของอาดุยโน่	15
2.11 รีโมทเครื่องปรับอากาศ Intronics Digital Room Thermostat	16
2.12 แผงวงจร ET-MINI W5100	17
2.13 การต่อใช้งานแผงวงจร ET-MINI W5100	19
2.14 Access point รุ่น ALFA	19
2.15 Adobe Dreamweaver CS5.5	20
2.16 การสร้างเว็บเซิร์ฟเวอร์สำเร็จรูป	21
3.1 แบบเครื่องระบบควบคุมเครื่องปรับอากาศ	23
3.2 ส่วนของจุดเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับไมโครคอลโทรเลอร์	24
3.3 ไมโครคอลโทรเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328	24
3.4 ขั้วต่อสัญญาณจาก Digital[0-7]	25
3.5 ขั้วต่อสัญญาณ Digital[8-13]	25
3.6 แผงวงจรรีโมทเครื่องปรับอากาศ	26
3.7 ไมโครคอลโทรเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128	26
3.8 ขั้วต่อสัญญาณจาก Digital[0-7]	27
3.9 แผงวงจรรีโมทเครื่องปรับอากาศ	27
3.10 ขั้วต่อสัญญาณจาก Digital[0-7]	28
3.11 ขั้วต่อสัญญาณจาก Digital[0-7]	29
3.12 หมายเลข 13 และ 14 คือพอร์ต Uart	29

สารบัญภาพ(ต่อ)

รูปที่	หน้า
3.13 แผงวงจรเชื่อมต่ออีเธอร์เน็ต	30
3.14 พอร์ตเชื่อมต่อระหว่าง ET-Mini กับแผงวงจรระบบเครือข่ายสัญญาณแบบไร้สาย	31
3.15 แผนภาพความสัมพันธ์การทำงานของระบบ	31
3.16 การใช้งานโดยการควบคุมผ่านเว็บบราวเซอร์	32
3.17 แผนภาพความสัมพันธ์ส่วนของการรับคำสั่งของการใช้งาน	33
3.18 แผนภาพความสัมพันธ์ส่วนตรวจสอบสถานะอุปกรณ์ในการทำงาน	34
3.19 แผนภาพความสัมพันธ์ส่วนแสดงผลสถานะของการทำงาน	37
3.20 เป็นการแสดงผลบนเว็บบราวเซอร์	38
3.21 ปุ่มควบคุม TEMP_UP	41
3.22 ปุ่มควบคุม TEMP_DOWN	42
3.23 ปุ่มควบคุม FAN	42
3.24 ปุ่มควบคุม MODE	43
3.25 ปุ่มควบคุม POWER	43
4.1 การทดลองประสิทธิภาพการระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ	47
4.2 การทดลองประสิทธิภาพการทำงาน (การกดผ่านเว็บควบคุมแบบไร้สาย)	47
4.3 การทดลองประสิทธิภาพการทำงาน แบบสายสัญญาณไขว้สาย	49
4.4 การทดลองประสิทธิภาพการทำงาน (การกดผ่านเว็บควบคุมแบบไร้สาย)	50
4.5 แผนภูมิแท่งการทดลองประสิทธิภาพการทำงาน	51

สารบัญตาราง

ตารางที่		หน้า
1.1	ขั้นตอนการทำงานและระยะการปฏิบัติงาน	3
2.1	ทฤษฎีเกี่ยวกับรีโมทเครื่องปรับอากาศ รุ่น Intronics Digital Room Thermostat	16
4.1	การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศ ไร้สายผ่านเว็บไซต์ (การกดผ่านเว็บควบคุมแบบไร้สาย)	48
4.2	การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศ ไร้สายผ่านเว็บไซต์ (แบบสายสัญญาณไขว้สาย)	49
4.3	การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศ ไร้สายผ่านเว็บไซต์	50

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากสภาพอากาศโดยรวมในประเทศไทยมีลักษณะที่ร้อนชื้น และยังมีปัญหามลพิษทางสิ่งแวดล้อม จึงมีการใช้เครื่องปรับอากาศในอาคารที่พักอาศัย ซึ่งเครื่องปรับอากาศเป็นอุปกรณ์ไฟฟ้าที่มีอัตราการใช้พลังงานสูงเมื่อเทียบกับอุปกรณ์ไฟฟ้าภายในครัวเรือน เครื่องปรับอากาศส่วนใหญ่เป็นระบบคอมเพรสเซอร์ ได้แก่ เครื่องปรับอากาศชนิดติดหน้าต่าง เครื่องปรับอากาศชนิดแยกส่วน และเครื่องปรับอากาศขนาดใหญ่ที่ใช้ในศูนย์การค้าหรือโรงงานอุตสาหกรรม เครื่องปรับอากาศแบบแยกส่วน (Split Type) จะประกอบด้วยอุปกรณ์ 2 ชุดภายในระบบ คือ ส่วนที่เป็นตัวจ่ายลมเย็น (Fan Coil) ที่ติดตั้งภายในอาคารหรือพื้นที่ที่ต้องการปรับอุณหภูมิ และส่วนระบายความร้อน (Condensing Unit) ที่ถูกติดตั้งภายนอกอาคาร และเครื่องปรับอากาศในปัจจุบัน ได้แบ่งการควบคุมออกเป็น 2 แบบ ทั้งมีสายและแบบไม่มีสายหรือไร้สาย ซึ่งควบคุมได้ใกล้ และแบบมีสายต้องเดินมาควบคุมที่รีโมทในการควบคุมการทำงาน

ปัจจุบันมีการประยุกต์ใช้เทคโนโลยีในหลายๆ ด้านได้มีการพัฒนาทั้งในด้วนระบบอินเตอร์เน็ต (Internet) และอุปกรณ์อิเล็กทรอนิกส์ (Electronics) ที่สามารถนำระบบอินเตอร์เน็ต และอุปกรณ์อิเล็กทรอนิกส์ช่วยในการควบคุมการควบคุม (Control) เครื่องปรับอากาศยี่ห่อ Focus ประเภท Intronic Digital Room Thermostat ได้นำมาใช้ในการศึกษาทดลอง เพราะเครื่องปรับอากาศประเภทการควบคุม เป็นรีโมทชนิดมีสายติดผนังจอแสดงผลดิจิตอลมีตัวรับรู้ (Sensor’s) ตรวจจับอุณหภูมิในตัวการควบคุมเครื่องปรับอากาศมี 3 ระดับในการควบคุม ได้แก่ เย็น (Cool) พัดลม (Fan) และแห้ง (Dry) เป็นต้น ใช้ในการควบคุมปรับเพิ่ม-ลด อุณหภูมิของเครื่องปรับอากาศ

จากศึกษาค้นคว้าข้อมูลยังไม่พบผู้ผลิตเครื่องปรับอากาศบริษัทใดประเภทใดที่ควบคุมการเปิดปิดปรับอุณหภูมิเครื่องปรับอากาศผ่านเว็บไซต์ได้ ดังนั้นงานวิจัยนี้จึงขอเสนอ การพัฒนาระบบควบคุมเครื่องปรับอากาศผ่านเว็บไซต์ โดยการศึกษาสภาพปัญหาการควบคุมการเปิดปิดของเครื่องปรับอากาศออกแบบ และสร้างระบบควบคุมแบบไร้สายผ่านเว็บไซต์ที่สามารถควบคุมเครื่องปรับอากาศในระยะไกล โดยสั่งการผ่านหน้าเว็บไซต์ที่ใช้งานได้จริง รวมทั้งนำเทคโนโลยีสารสนเทศเข้ามาประยุกต์ใช้ในการควบคุมแทนการควบคุมแบบเดิมในปัจจุบัน สามารถสั่งควบคุมได้ระยะไกล โดยอาศัยระบบการสื่อสารไร้สายเป็นสื่อกลาง และช่องทางในการควบคุมผ่านหน้าเว็บไซต์อีกทั้งยังสามารถนำมาประยุกต์ใช้กับอุปกรณ์ไฟฟ้าชนิดอื่นๆ ได้ต่อไป

1.2 วัตถุประสงค์ของโครงงาน

เพื่อควบคุมเครื่องปรับอากาศผ่านเครือข่ายไร้สาย

1.3 ขอบเขตของโครงงาน

1.3.1 สามารถการเปิด – ปิดเครื่องปรับอากาศได้

1.3.2 สามารถปรับระดับความเย็นอุณหภูมิได้

1.3.3 สามารถปรับรูปแบบ (Mode) ได้

1.3.2 ระบบการสั่งการควบคุมเครื่องปรับอากาศเปิด – ปิดปรับระดับความเย็นอุณหภูมิโดยใช้ไมโครคอนโทรเลอร์ควบคุม

1.4 ขั้นตอนการวิจัย

1.4.1 รวบรวมข้อมูลและศึกษาหัวข้อปริญญานิพนธ์ที่เกี่ยวข้อง

1.4.2 นำเสนอหัวข้อปริญญานิพนธ์

1.4.3 ศึกษาการทำงานของอุปกรณ์

1.4.4 ศึกษาเกี่ยวกับเทคโนโลยีที่ใช้ในการเขียนโปรแกรมและการติดต่อกับอุปกรณ์

1.4.5 ออกแบบ และสร้างฮาร์ดแวร์

1.4.6 จัดทำโปรแกรมและส่วนต่างๆ ของโปรแกรม และทดสอบสอบการทำงานของโปรแกรม

1.4.7 ทดสอบการทำงานโดยมีการเชื่อมต่อผ่านทางอินเตอร์เน็ต

1.4.8 บันทึกผลการทดสอบการทำงาน

1.4.9 วิเคราะห์สรุปเมื่อเกิดปัญหาจึงหาทางแก้ปัญหาที่เกิดขึ้น

1.4.10 จัดทำคู่มือปริญาญานิพนธ์

1.5 แผนการดำเนินงาน

จากการดำเนินงานของการพัฒนาระบบควบคุมระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไซต์ สามารถลำดับเป็นแผนการดำเนินงาน ได้ดังนี้

ตารางที่ 1.1 แสดงขั้นตอนการทำงานและระยะการปฏิบัติงาน

ระยะเวลา / ปฏิบัติงาน / ขั้นตอนการดำเนินงาน	พ.ศ.2556									พ.ศ.2557		
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.
1.ศึกษาความเป็นมาของโครงงาน	◄	———	►									
2.รวบรวมข้อมูลที่เกี่ยวข้องเพื่อเสนอโครงงานเข้าพบอาจารย์ที่ปรึกษา	◄	———	———	———	►							
3.จัดทำโครงการเสนออาจารย์ประจำวิชา		◄	———	———	———	►						
4.ค้นคว้าข้อมูลและการทำงานของโปรแกรม			◄	———	———	———	►					
5.ออกแบบโครงสร้างและเขียนโปรแกรมอ่านป้ายทะเบียน					◄	———	———	———	►			
6.ทดสอบและแก้ไขปัญหาของระบบทั้งหมด						◄	———	———	———	►		
7.ประเมินผลของการทดลอง							◄	———	———	►		
8.จัดทำรูปเล่มปริญญานิพนธ์										◄	———	►

1.6 เครื่องมือที่ใช้วิจัย

1.6.1 แผงวงจรไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ

1.6.2 รีโมทเครื่องปรับอากาศ ประเภท Intronics Digital Room Thermostat

1.7 ประโยชน์ที่คาดว่าจะได้รับ

สามารถควบคุมเครื่องปรับอากาศผ่านหน้าเว็บไซต์ได้จริง

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

ในการศึกษาทฤษฎีที่เกี่ยวข้องกับงานวิจัย เรื่องพัฒนาระบบควบคุมเครื่องปรับอากาศผ่านเว็บไซต์ โดยมีหลักการและทฤษฎีต่างๆ ที่เกี่ยวข้องกับฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ดังนี้

2.1 ไมโครคอนโทรลเลอร์ (Microcontroller)

2.2 รีโมทเครื่องปรับอากาศ รุ่น Intronics Digital Room Thermostat

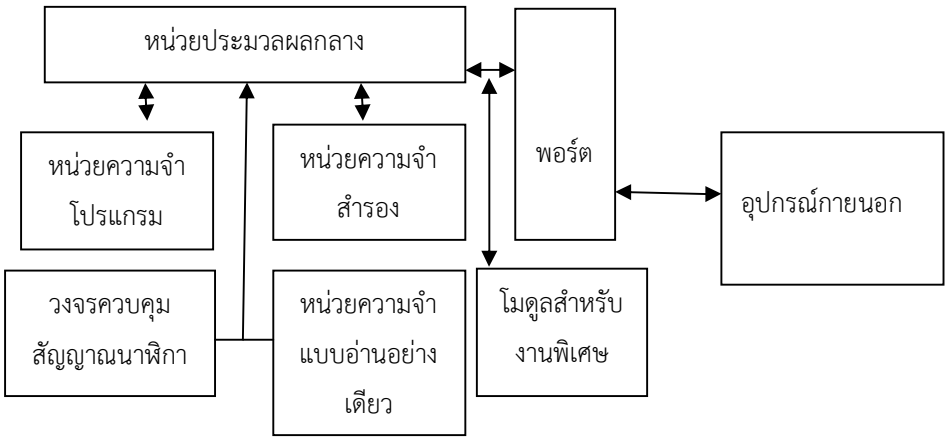
2.3 อุปกรณ์รับส่งข้อมูล

2.4 โปรแกรมสร้างเว็บเพจแบบเสมือนจริง (Dreamwwaver)

2.5 การออกแบบฐานข้อมูล (Data Base Design)

2.1 ไมโครคอนโทรลเลอร์ (Micro Controller)

ไมโครคอนโทรลเลอร์[1] หมายถึง อุปกรณ์ควบคุมขนาดเล็กที่ตัวการควบคุมหรืออุปกรณ์ควบคุม โดยรวมความสามารถของ ระบบคอมพิวเตอร์ซึ่งไมโครคอนโทรลเลอร์ ประกอบด้วย ประมวลผลกลาง หน่วยความจำ และส่วนติดต่อกับอุปกรณ์ภายนอก ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์บรรจุเข้าไว้ด้วยกัน ไมโครคอนโทรลเลอร์สร้างขึ้นเพื่อรองรับการออกแบบระบบที่มีขนาดเล็กและสามารถป้อนชุดคำสั่งให้สามารถทำงานได้อย่างอัตโนมัติ ดังรูปที่ 2.1



รูปที่ 2.1 โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์

จากรูปที่ 2.1 โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์ สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ ๆ ดังต่อไปนี้

- หน่วยประมวลผลกลาง (CPU :Central Processing Unit) เป็นหน่วยประมวลผลกลางซึ่งภายในมี หน่วยควบคุม (Control Unit) เป็นหน่วยที่ใช้สร้างสัญญาณเพื่อควบคุมการทำงานภายในไมโครคอนโทรลเลอร์

- หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ (Hard Disk) ของเครื่องคอมพิวเตอร์ อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) ใช้ในการคำนวณของหน่อยประมวลผลกลางสำรอง และเป็นที่พักข้อมูลชั่วคราวขณะทำงานแต่ไม่มีไฟเลี้ยง ข้อมูลหายคล้ายกับหน่วยความจำแรม (RAM: Random Access Memory) ในเครื่องคอมพิวเตอร์

- ส่วนติดต่อกับอุปกรณ์ภายนอกหรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ในการเชื่อมต่อกับอุปกรณ์ภายนอก

- ช่องทางเดินของสัญญาณหรือบัส (Bus) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างหน่วยประมวลผลกลาง หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากภายในตัวไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล (Data Bus) บัสที่อยู่ (Address Bus) และบัสควบคุม (Control Bus)

- วงจรควบคุมสัญญาณนาฬิกาเป็นส่วนสำคัญที่ใช้กำหนดจังหวะให้กับไมโครคอนโทรลเลอร์ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานไมโครคอนโทรลเลอร์มีความเร็วในการประมวลผลสูง

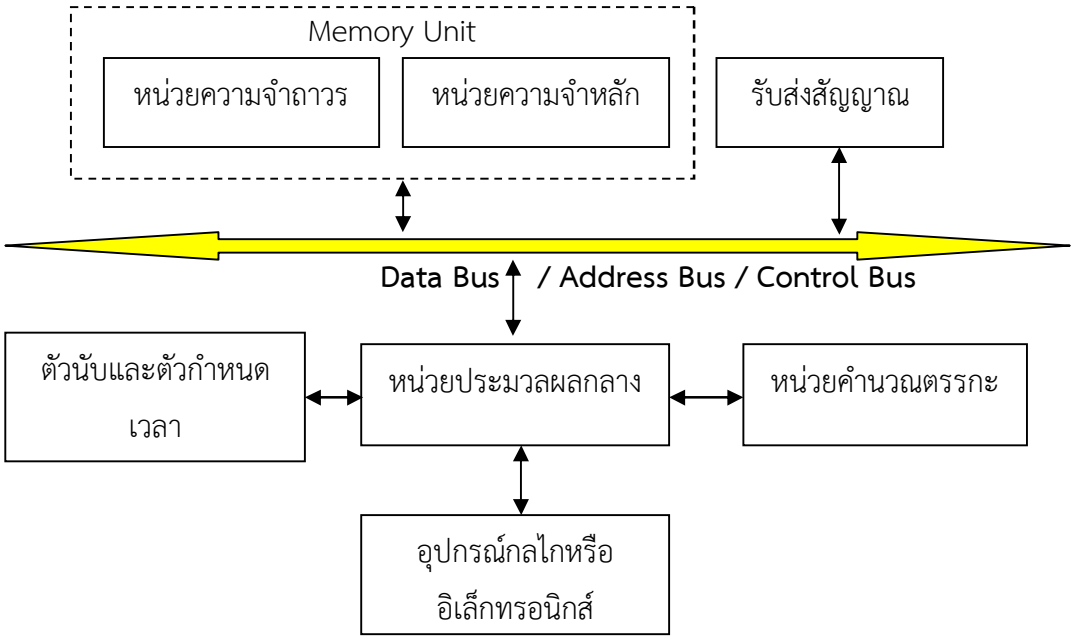
- โมดูลสำหรับงานพิเศษ ไมโครคอนโทรลเลอร์ถูกบรรจุวงจรการทำงานพิเศษ เช่น วงจรแปลงสัญญาณอนาล็อก (Analog) เป็นสัญญาณดิจิตอล (A/D Converter : Analog to Digital Converter) วงจรสร้างสัญญาณควบคุมพัลซ์ (PWM:Pulse Width Modulation) วงจรส่งข้อมูลแบบอนุกรม (Serial Transportation) วงจรตรวจจับสัญญาณอินพุต (Input Detection) เป็นต้น

2.1.1 ประเภทของไมโครคอนโทรลเลอร์

2.1.1.1 ไมโครคอนโทรลเลอร์อุปกรณ์ต่อพ่วงแบบควบคุม[2] (PIC : Peripheral Interface Controller)

ประกอบไปด้วยโปรแกรมเก็บหน่วยความจำหลัก (ProgramMemory), หน่วยความจำหลัก (Main Memory),หน่วยส่วนความจำแบบอ่านอย่างเดียว (EEPROM : Erasable Electrically Read-Only Memory) การสื่อสารอนุกรม (I2C : Inter Integrate Circuit),การปรับสัญญาณให้กว้าง (PWM : Pulse Width Modulation),คือการแปลงสัญญาณอนาล็อกเป็นดิจิตอล

โดยต้องต่ออุปกรณ์เสริมจากภายนอกอุปกรณ์ต่อพ่วงแบบควบคุม มีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ ซึ่งทำให้อุปกรณ์ต่อพ่วงแบบควบคุม เป็นหน่วยประมวลผลกลาง ดังรูปที่ 2.2



รูปที่ 2.2 โครงสร้างของแผงวงจรไมโครคอนโทรลเลอร์อุปกรณ์ต่อพ่วง

จากรูปที่ 2.2 โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์อุปกรณ์ต่อพ่วง คอนโทรลเลอร์อินเตอร์เฟซ ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

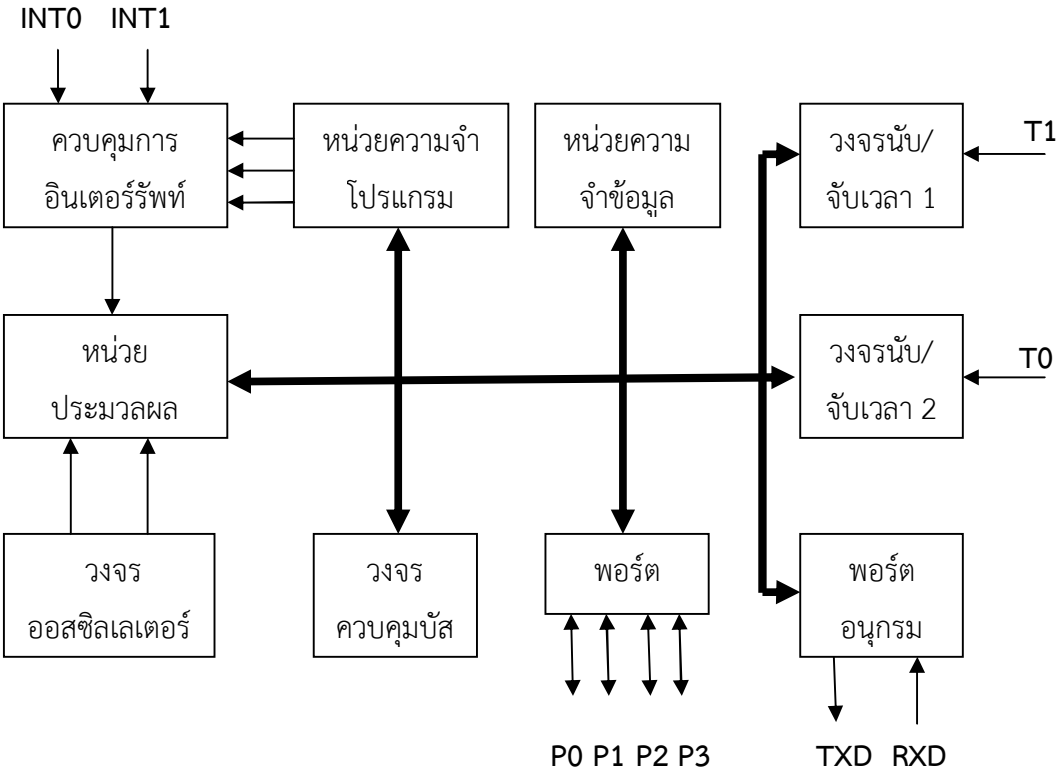
- หน่วยประมวลผลกลางประกอบด้วยวงจรที่จำเป็นสำหรับประมวลผลและการคำนวณ เช่น วงจรถอดรหัสคำสั่ง (Command Decoder) วงจรควบคุมสัญญาณนาฬิกา (Clock Control Circuit), วงจรควบคุมการทำงาน (Cycle Control), วงจรตั้งเวลา (Cycle Timer) และหน่วยคำนวณทางลอจิกและคณิตศาสตร์ (ALU: Arithmetic and Logical Unit) เป็นต้น

- หน่วยความจำ (Memory Unit) การเขียนโปรแกรมด้วยภาษาซีให้กับไมโครคอนโทรลเลอร์ที่มีขนาดความจุขนาดของหน่วยความจำโปรแกรมไมโครคอนโทรลเลอร์ ไม่สามารถบรรจุโปรแกรมลงครบสมบูรณ์ ใช้หน่วยความจำแบบอื่นในไมโครคอนโทรลเลอร์

- พอร์ตอินพุตและเอาท์พุต (I/O Port) ไมโครคอนโทรลเลอร์ทุกประเภทต้องมีขาใช้งาน และสามารถเป็นขาสัญญาณออกและขาสัญญาณเข้า สามารถติดต่อกับอุปกรณ์ภายนอก หรือนำไปใช้ เช่น จอแสดงผล, ปุ่มกด, อุปกรณ์ตรวจจับสัญญาณ เป็นต้น ตามมาตรฐานพอร์ตหนึ่งมี 8 บิต ขาต่อใช้งาน 8 ขา

2.1.1.2 ไมโครคอนโทรลเลอร์ประเภทเอ็มซีเอสระบบควบคุมขนาดเล็ก (MCS : Micro Controller Series)

ไมโครคอนโทรลเลอร์ระบบควบคุมขนาดเล็ก เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิตที่มีอุปกรณ์สนับสนุนประกอบอยู่ภายหน่วยความจำสำหรับเก็บข้อมูล หน่วยความจำสำหรับเก็บโปรแกรม ตัวตั้งเวลา/ตัวนับ อุปกรณ์รับส่งข้อมูลแบบอนุกรม เนื่องจากโครงสร้างของไมโครคอนโทรลเลอร์มีอุปกรณ์สนับสนุนประกอบอยู่ภายใน ทำให้การใช้งานมีประสิทธิภาพโดยไม่ต้องมีการเชื่อมต่ออุปกรณ์ภายนอกเพิ่มเติมมากเหมือนกับ ตัวไมโครโปรเซสเซอร์ทั่วไป ใช้งานไมโครคอนโทรลเลอร์ร่วมกับ อุปกรณ์อื่นเพิ่มเติมเช่น วงจรรวม (IC: Integrated Circuit) หรือหน่วยความจำภายนอก มาเชื่อมต่อเพิ่มเติมเข้ากับไมโครคอนโทรลเลอร์ ดังรูปที่ 2.3



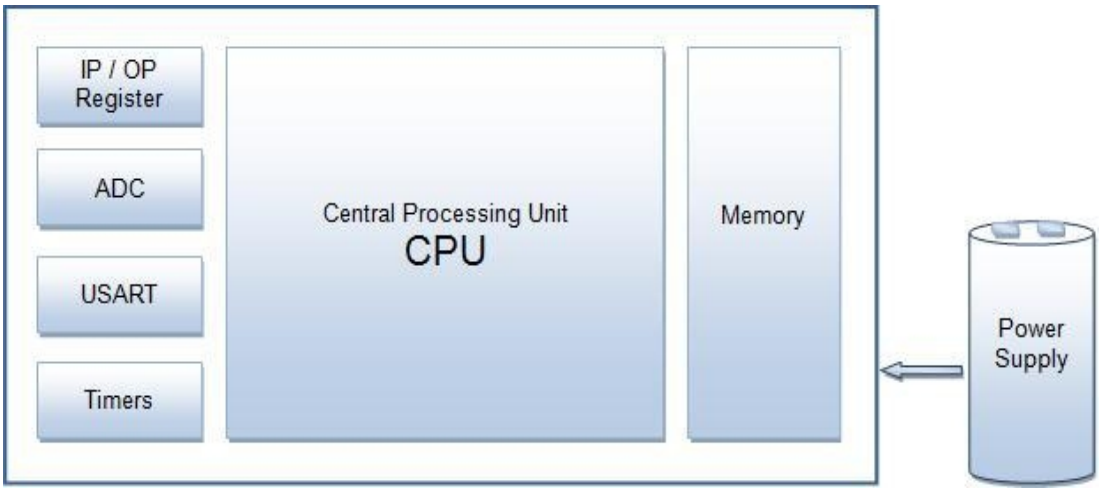
รูปที่ 2.3 แสดงโครงสร้างภายในของไมโครคอนโทรลเลอร์ MCS-51

จากรูปที่ 2.2 โครงสร้างภายในพื้นฐานของไมโครคอนโทรลเลอร์ประเภทเอ็มซีเอสระบบควบคุมขนาดเล็ก ประกอบด้วยอุปกรณ์ต่างๆ ดังนี้

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต

- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์
- หน่วยความจำภายใน
- พอร์ตรับส่งสัญญาณแบบขนาน ซึ่งสามารถแยกทำงานอย่างอิสระ
- วงจรนับ (Counter) จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรม (Serial Communication)
- วงจรควบคุมวงจรการขัดจังหวะ (Interrupt) จากแหล่งกำเนิดสัญญาณ พร้อมการกำหนดลำดับ
- วงจรผลิตสัญญาณนาฬิกาภายในโครงสร้างการทำงานทั้งหมดของไมโครคอนโทรลเลอร์อาศัยหลักการทำงานที่เกี่ยวข้องกัน โดยอาศัยหลักการทำงานที่เป็นไป ตามโครงสร้าง

2.1.1.3 ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ (AVR : Automatic Voltage Regulator) มีความเร็วในการประมวลผล 1 คำสั่ง ต่อ 1 สัญญาณนาฬิกา ดังรูปที่ 2.4

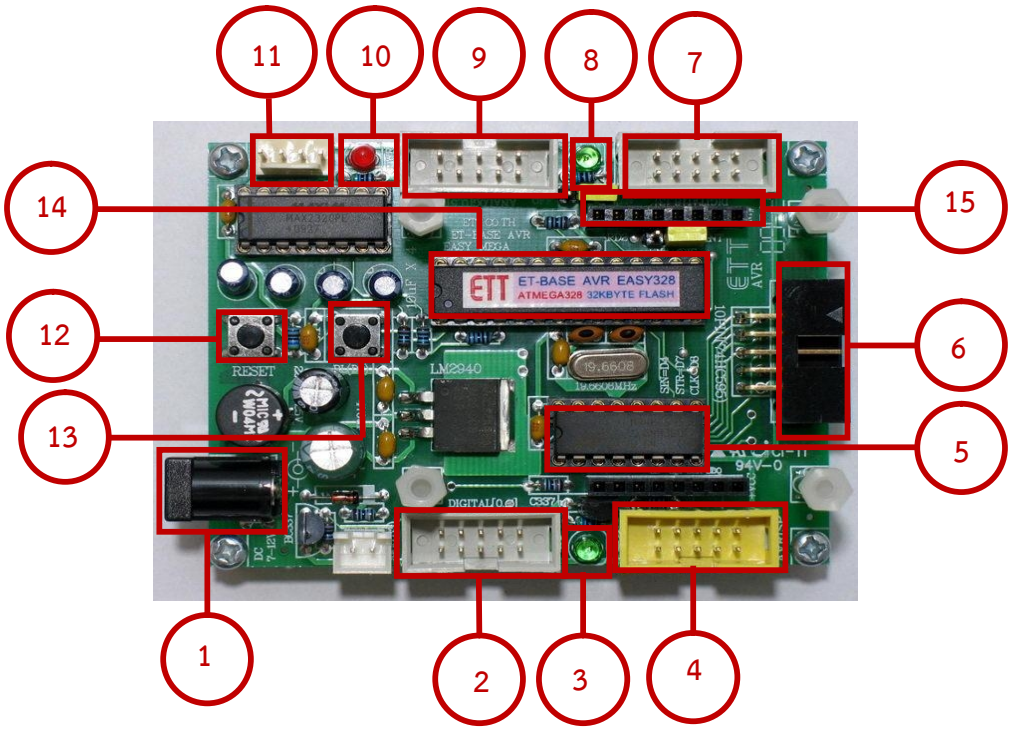


รูปที่ 2.4 โครงสร้างไมโครคอนโทรลเลอร์ประเภทควบคุมแรงดันไฟฟ้าอัตโนมัติ

การนำไปใช้งานไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328

เป็นไมโครคอนโทรลเลอร์ประเภทควบคุมแรงดันไฟฟ้าอัตโนมัติ[9] โดยมีการออกแบบพัฒนามาสำหรับงานที่ต้องใช้รับส่งสัญญาณโดยใช้ไมโครคอนโทรลเลอร์เบอร์ ATMEGA 2560-16 ซึ่งมี อุปกรณ์เก็บข้อมูลถาวรสำหรับคอมพิวเตอร์ที่สามารถลบและเขียนหน่วยความจำชั่วคราว 8

กิโลไบต์ และหน่วยความจำแบบอ่าน 4 กิโลไบต์ ซึ่งตัวของแผงวงจรมีขนาด 5.3 x 10.2 x 2 เซนติเมตร (Cm: Centimeter) ดังรูปที่ 2.5



รูปที่ 2.5 แผงวงจร ET-BASE AVR EASY328

จากรูปที่ 2.3 แผงวงจร ET-BASE AVR EASY328 เป็นการแสดงโครงสร้างการใช้งานแผงวงจรตามหมายเลขดังต่อไปนี้

ตำแหน่งที่ 1 ขั้วต่อแหล่งจ่ายไฟเลี้ยงวงจรของบอร์ด ใช้กับแหล่งจ่าย

ตำแหน่งที่ 2 ขั้วต่อสัญญาณจากช่องทางการสื่อสาร

ตำแหน่งที่ 3 ใช้แสดงสถานะของขาสัญญาณ

ตำแหน่งที่ 4 ขั้วต่อ AVRISP ใช้สำหรับดาวน์โหลด

ตำแหน่งที่ 5 วงจรรวมเบอร์ 74HC595 ซึ่งใช้ขยาย ช่องการสื่อสารส่งออก

ตำแหน่งที่ 6 ขั้วต่อสัญญาณ ส่งข้อมูลออก

ตำแหน่งที่ 7 ขั้วต่อสัญญาณซึ่งในกรณีใช้การพัฒนาโปรแกรม

ตำแหน่งที่ 8 แสดงสถานะของขาสัญญาณหรือขาสัญญาณเชิงตัวเลขไมโครคอนโทรลเลอร์

ตำแหน่งที่ 9 ขั้วต่อสัญญาณในกรณีใช้การพัฒนา

ตำแหน่งที่ 10 ใช้แสดงสถานะของแหล่งจ่ายไฟกระแสตรง

ตำแหน่งที่ 11 ขั้วต่อสำหรับใช้งานทั่วไป และการโอนย้ายไฟล์หรือข้อมูลจากที่หนึ่งไปอีกที

ตำแหน่งที่ 12 สวิตช์ตั้งใหม่ใช้สำหรับเริ่มใหม่การทำงาน

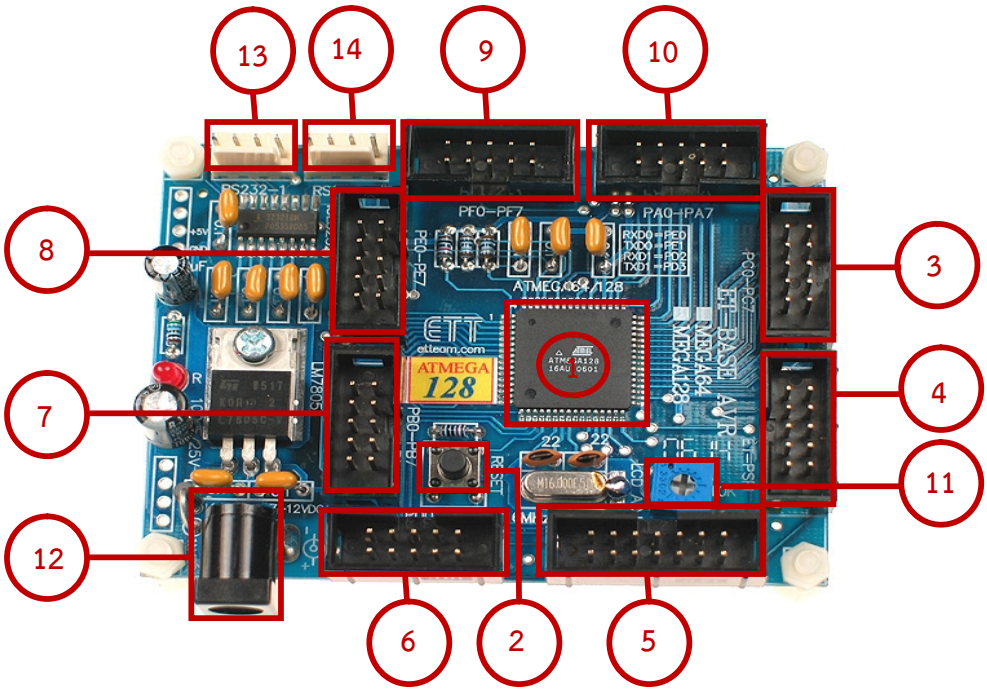
ตำแหน่งที่ 13 การดาวน์โหลดข้อมูลชุดคำสั่งระบบโดยต่อผ่านขาสัญญาณ

ตำแหน่งที่ 14 การจัดการระบบภาพและเสียงประจำ

ตำแหน่งที่ 15 หัวสำหรับรองรับการเชื่อมต่อสัญญาณกับแผงวงจร ET-MINI ENC28J60

การนำไปใช้งานไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128

ไมโครคอนโทรลเลอร์ประเภทควบคุมแรงดันไฟฟ้าอัตโนมัติเป็นวงจรรวมไมโครคอนโทรลเลอร์ของบริษัท Atmel ไมโครโพรเซสเซอร์ที่มีชุดคำสั่ง โดยใช้สัญญาณนาฬิกาเพียง 1 ลูกคลื่นในการปฏิบัติงานใน 1 คำสั่ง โดยประกอบด้วยหน่วยความจำโปรแกรมภายในอุปกรณ์เก็บข้อมูลถาวรโปรแกรมข้อมูลได้แบบในระบบโปรแกรม (ISP:In-System programmable) และบางประเภทสามารถกำหนดตำแหน่งของหน่วยความจำที่สร้าง การเขียนโปรแกรมเพื่อติดต่อกับ คอมพิวเตอร์ส่วนบุคคล มีขนาดของหน่วยความจำตามประเภทของวงจรรวม โครงสร้างของแผงวงจร ET-BASE MEGA128[4] ดังรูปที่ 2.6



รูปที่ 2.6 แผงวงจร ET-BASE MEGA128

จากรูปที่ 2.3 คือโครงสร้างการนำไปใช้งานไมโครคอนโทรลเลอร์ ET-BASE MEGA128 เป็นการแสดงโครงสร้างบอร์ดตามหมายเลขดังต่อไปนี้

ตำแหน่งที่ 1 ใช้จัดเก็บระบบภาพและเสียงจากหลายแหล่งมารวมกัน (MCU:Multipoint Control Unit) เบอร์ ATmega128 ประเภทไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติจาก ATMEL

ตำแหน่งที่ 2 สวิตช์เริ่มใหม่ (Switch RESET) ใช้สำหรับเริ่มใหม่

ตำแหน่งที่ 3 พอร์ตซี มีขนาด 8 บิต ประเภท อินพุต/เอาท์พุต

ตำแหน่งที่ 4 พอร์ต ISP LOAD ใช้สำหรับดาวน์โหลด Hex File

ตำแหน่งที่ 5 พอร์ต ET-CLCD สำหรับเชื่อมต่อกับจอแสดงผลซึ่งใช้การเชื่อมต่อแบบ 4 บิต

ตำแหน่งที่ 6 พอร์ตดี มีขนาด 8 บิต ประเภท อินพุต/เอาท์พุต

ตำแหน่งที่ 7 พอร์ตบี มีขนาด 8 บิต ประเภท อินพุต/เอาท์พุต

ตำแหน่งที่ 8 พอร์ตอี มีขนาด 8 บิต ประเภท อินพุต/เอาท์พุต

ตำแหน่งที่ 9 พอร์ตเอฟ มีขนาด 8 บิต ประเภท อินพุต/เอาท์พุต

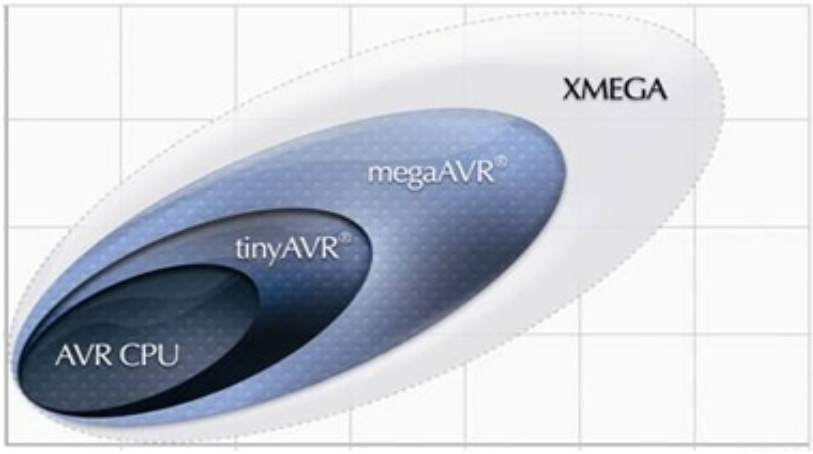
ตำแหน่งที่ 10 พอร์ตเอ มีขนาด 8 บิต ประเภท อินพุต/เอาท์พุต

ตำแหน่งที่ 11 ตัวต้านทานสำหรับปรับค่าความสว่างให้จอภาพ

ตำแหน่งที่ 12 ขั้วต่อแหล่งจ่ายไฟสำหรับเลี้ยงวงจรของบอร์ด

ตำแหน่งที่ 13,14 ขั้วต่อ RS232 สำหรับใช้งานทั่วไป

ประเภทของไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ ได้แบ่งออกเป็น 4 กลุ่ม ดังรูปที่ 2.7



รูปที่ 2.7 ประเภทไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ

จากรูปที่ 2.7 ประเภทของไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ ได้แบ่งออกเป็น 4 กลุ่ม ดังต่อไปนี้

ก) TinyAVR -ATtiny series เช่นเบอร์ Tiny13 , Tiny2313

- มีหน่วยความจำโปรแกรมขนาด 1– 8 กิโลไบต์
- มีจำนวนขาใช้งาน 6–32 ขา
- มีส่วนของอุปกรณ์เสริมที่จำกัด

ข) Mega AVR - ATmega series เช่นเบอร์ ATmega8 ,ATmega16, ATmega32, ATmega64

- มีหน่วยความจำโปรแกรมขนาด 4 – 256 กิโลไบต์
- มีจำนวนขาใช้งาน 28 –100 ขา
- มีชุดคำสั่งที่สามารถจัดการกับหน่วยความจำที่มีขนาดใหญ่
- มีส่วนของอุปกรณ์เสริมมากในตัววงจรรวม

ค) XMEGA - ATxmega series เช่นเบอร์ ATxmega64A1, ATxmega128A1

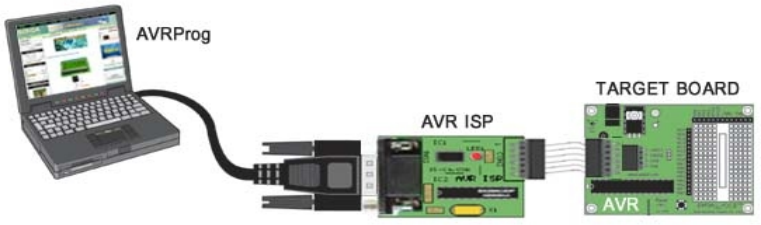
- มีหน่วยความจำโปรแกรมขนาด 16– 384 กิโลไบต์
- มีจำนวนขาใช้งาน 44–64–100 ขา
- มีชุดคำสั่งที่สามารถจัดการกับระบบ ของบัสคอมพิวเตอร์ให้ข้อมูลสามารถส่งโดยตรงจากอุปกรณ์ที่ติดต่อ (DMA : Direct Memory Access) และการเข้าถึงเหตุการณ์ โดยใช้การสื่อสารในแบบต่างๆ หลายรูปแบบ
- มีส่วนของอุปกรณ์เสริมในตัววงจรรวม โดยการใช้งานกับ ดิจิตอลเป็นอนาล็อก และสามารถเขียนรหัสเฉพาะ โดยเข้ากันได้กับไฟล์แบบการเข้ารหัสขั้นสูง (AES : Advanced Encryption Standard) และการเข้ารหัสข้อมูล (DES: Data Encryption Standard)

ง) Application Specific Automatic Voltage Regulator

- เป็นวงจรรวมที่สร้างเพื่อใช้งานเฉพาะ เช่น พื้นที่ควบคุมเครือข่าย (CAN AVR: Controller Area Network AVR)

เครื่องโปรแกรมแบบ ในระบบการเขียนโปรแกรม ISP

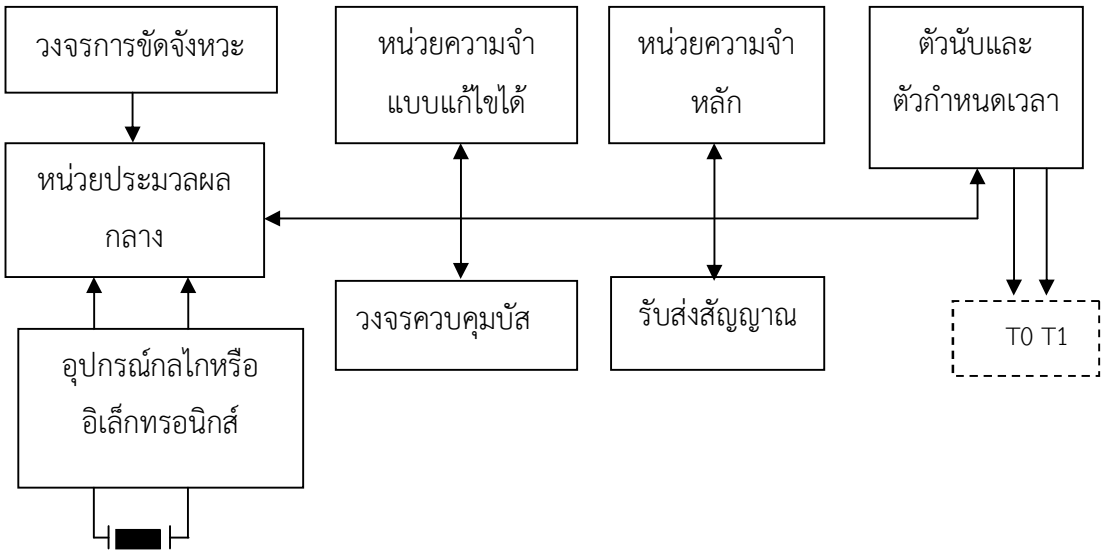
สามารถโปรแกรมข้อมูลลงในหน่วยความจำโปรแกรมโดยตรง โดยไม่ต้องถอดวงจรรวมจากแผงวงจร การแก้ไขข้อมูลทำได้ง่าย ดังรูปที่ 2.8



รูปที่ 2.8 เครื่องโปรแกรมแบบ ISP (In-System Programming)

2.1.1.4 ไมโครคอนโทรลเลอร์ประเภท อาดุยโน่

ไมโครคอนโทรลเลอร์ อาดุยโน่ คือเครื่องมือที่ทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้ ซึ่งใช้แบบซอฟต์แวร์ที่เปิดเผย (OSS : Open Source Software) และยังสามารถสร้างคำสั่งและไลบรารี (Library) ขึ้นมาใช้งานได้ เช่น การรับสัญญาณจากสวิทช์ หรือตัวรับรู้ และควบคุมหลอดไฟ, มอเตอร์ (Motor), หรืออุปกรณ์อื่น ตัวแผงวงจรออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยว (Single Chip) ดังรูปที่ 2.9



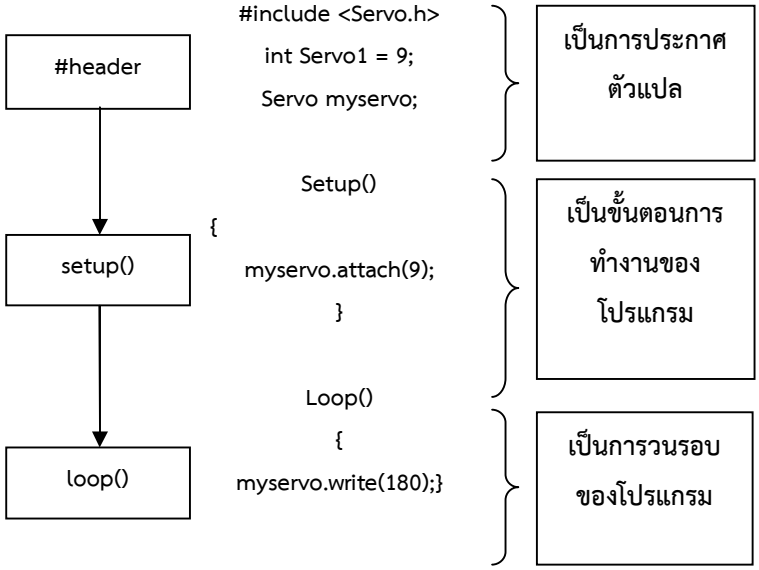
รูปที่ 2.9 โครงสร้างไมโครคอนโทรลเลอร์อาดุยโน่

จากรูปที่ 2.9 เป็นแผงวงจร อาดุยโน่ ที่มีช่องเสียบ (port) เพื่อการต่อวงจรที่มีการเชื่อมต่อกับอุปกรณ์ภายนอกโปรแกรมภาษาของอาดุยโน่ ใช้ภาษาซีซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์ในการเขียนโปรแกรม โครงสร้างพื้นฐานของภาษาซีที่ใช้กับอาดุยโน่ประกอบไปด้วย 3 ส่วนใหญ่ คือ

1. ส่วนหัว (Header) ต้องกำหนดไว้ในส่วนเริ่มต้นของโปรแกรม ซึ่งส่วนของ ส่วนหัวรวมไปถึงส่วนของการประกาศตัวแปร และค่าคงที่ต่างๆที่ใช้ในโปรแกรม

2. การติดตั้ง (Setup) เป็นฟังก์ชั่น (Function) บังคับที่ต้องกำหนดให้มีในโปรแกรม บางโปรแกรมไม่ต้องการใช้งานแต่คงการประกาศไว้ เพื่อให้ครบรูปแบบในการเป็นโปรแกรมเพียงแต่ไม่ต้องเขียนคำสั่งใดๆไว้ในระหว่างวงเล็บปีกกา { } ที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชั่น โดยฟังก์ชั่นใช้สำหรับบรรจุคำสั่งในส่วนที่ต้องการให้โปรแกรมทำงานเพียงรอบเดียว เท่านั้น โดยคำสั่งที่บรรจุไว้ในฟังก์ชั่น ได้แก่ คำสั่งสำหรับกำหนดโหมดการทำงานของคำสั่งสำหรับการกำหนดคุณสมบัติของช่องทางสื่อสารอนุกรม เป็นต้น

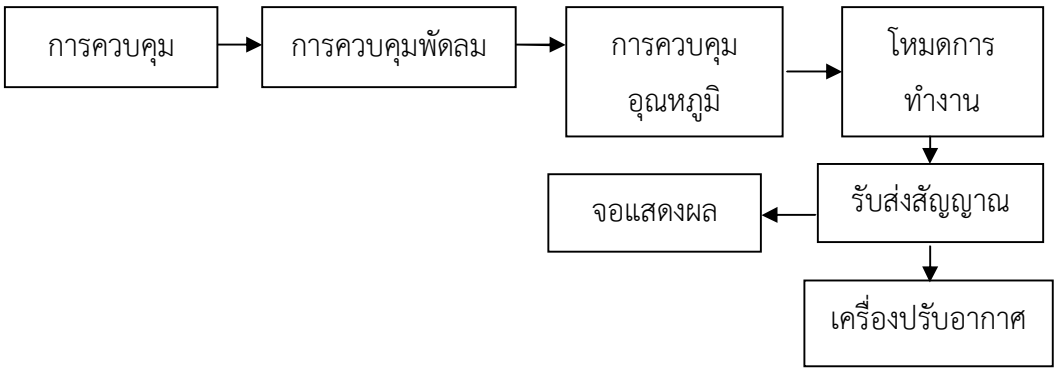
3. วนรอบ (Loop) เป็นฟังก์ชั่นบังคับที่กำหนดให้มีในโปรแกรมกับฟังก์ชั่น Setup() โดยฟังก์ชั่น Loop() ใช้บรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวนรอบตามลำดับและเงื่อนไขที่กำหนดคำสั่งจากภายนอก ที่มีการสร้างและเก็บรวบรวมเป็นไฟล์ในรูปแบบฟังก์ชั่นไลบรารี (Library Function) เก็บอยู่ภายนอกโปรแกรม เมื่อต้องการใช้งานสั่งไฟล์เข้ามาใช้งานในโปรแกรม ใช้งานคำสั่งต่างๆ ที่สร้างเก็บไว้ในไฟล์ได้ ดังรูปที่ 2.10



รูปที่ 2.10 โครงสร้างโปรแกรมของอาดุยโน่

2.2 อุปกรณ์ควบคุมเครื่องปรับอากาศรุ่น Intronics Digital Room Thermostat

รีโมทเครื่องปรับอากาศ ของ Focus ชนิดมีสายที่นำมาใช้ในการทดลอง รุ่น Intronics Digital Room Thermostat[3] ดังรูปที่ 2.11



รูปที่ 2.11 รีโมทเครื่องปรับอากาศ Intronics Digital Room Thermostat

จากรูปที่ 2.11 รีโมทเครื่องปรับอากาศ รุ่น Intronics Digital Room Thermostat ดังตารางที่ 2.1

ตารางที่ 2.1 การทำงานของรีโมทเครื่องปรับอากาศ รุ่น Intronics Digital Room Thermostat

รูปแบบ Model	**โหมดการทำงาน Mode of operation**	**ระบายความร้อน Cooling/Heating**	**คอมเพรสเซอร์แอร์ Compressor delay**
01	cool + sweep	Compressor	yes
02	cool + sweep	Chilled water	no
03	cool – fan – dry	Compressor	yes
04	cool – fan – dry	Chilled water	no
05	cool – fan – dry – heat – auto	Compressor/Heatpump	yes
06	cool – fan – dry – heat – auto	Compressor/Electricheater	yes, only for cool

ตารางที่ 2.1 (ต่อ) การทำงานของรีโมทเครื่องปรับอากาศ รุ่น Intronics Digital Room Thermostat

รูปแบบ Model	**โหมดการทำงาน Mode of operation**	**ระบายความร้อน Cooling/Heating**	**คอมเพรสเซอร์แอร์ Compressor delay**
07	cool – fan – dry – heat – auto	Hot/Chilled water	no

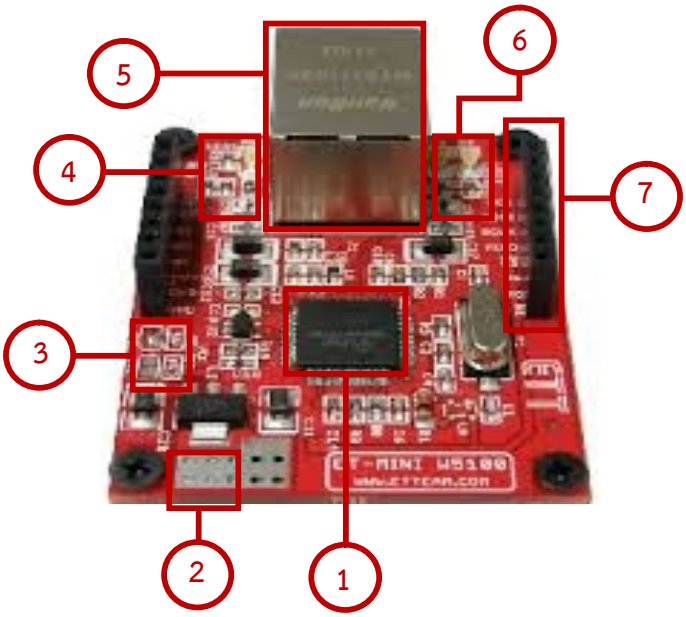
จากตารางที่ 2.1 เป็นทฤษฎีเกี่ยวกับรีโมทเครื่องปรับอากาศ รุ่น Intronics Digital Room Thermostat ที่นำมาใช้ในการทดลองระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ

2.3 อุปกรณ์รับส่งข้อมูล

การสื่อสารข้อมูล หมายถึง การโอนถ่าย (Transmission) ข้อมูลหรือการแลกเปลี่ยนข้อมูลระหว่างต้นทางกับปลายทาง โดยใช้อุปกรณ์ทางอิเล็กทรอนิกส์หรือเครื่องคอมพิวเตอร์ ซึ่งมีตัวกลาง เช่น ซอฟต์แวร์คอมพิวเตอร์สำหรับควบคุมการส่งและการไหลของข้อมูลจากต้นทางไปถึงปลายทาง อาจมีผู้รับผิดชอบกำหนดกฎเกณฑ์ในการส่งหรือรับข้อมูลตามรูปแบบ

2.3.1 แผงวงจร ET-MINI W5100

แผงวงจรเชื่อมต่อเครือข่ายเป็นวงจรของที่สามารถเชื่อมต่อกับอินเตอร์เน็ตโดยผ่านสายสัญญาณเครือข่ายท้องถิ่นหรือหัวต่อ RJ45 เป็นการเชื่อมต่อโดยผ่านทางไอพี (IP : Internet Protocol) ติดต่อกับอุปกรณ์ แผงวงจรเชื่อมต่อเครือข่ายเป็นอุปกรณ์แบบเชื่อมต่ออีเธอร์เน็ต (Ethernet Interface) สำหรับการสื่อสารผ่านเครือข่าย (TCP/IP: Transmission Control Protocol/Internet Protocol) โดยใช้ หน่วยของชุดวงจรรวมของคอมพิวเตอร์ จากแผงวงจรเชื่อมต่อเครือข่าย ดังรูปที่ 2.12



รูปที่ 2.12 แผงวงจร ET-MINI W5100[5]

จากรูปที่ 2.12 ส่วนประกอบของแผงวงจร ET-MINI W5100 เป็นการแสดงโครงสร้างแผงวงจรตามหมายเลขดังต่อไปนี้

ตำแหน่งที่ 1 หน่วยของชุดวงจรรวมของคอมพิวเตอร์ เบอร์ W5100 ของบริษัท WIZnet ซึ่งเป็นวงจรรวม Ethernet Controlle

ตำแหน่งที่ 2 ภาคจ่ายไฟ 3.3 โวลต์ สำหรับจ่ายไฟเลี้ยงแผงวงจร

ตำแหน่งที่ 3 จั้มเปอร์สำหรับเลือกแรงดัน (VCC: Voltage control circuit) ของแผงวงจร

ตำแหน่งที่ 4 จอแสดงผลสถานะไฟเลี้ยงของแผงวงจร

ตำแหน่งที่ 5 คอนเน็คเตอร์ RJ45 สำหรับเชื่อมต่อกับระบบการเชื่อมต่อเครือข่ายคอมพิวเตอร์

ตำแหน่งที่ 6 จอแสดงผลสถานะการรับส่งข้อมูลขาส่ง (TX: Transmitter) , ขารับ (RX: Receiver)

ตำแหน่งที่ 7 อุปกรณ์สำหรับต่อไฟฟ้าเพื่อใช้เลี้ยงแผงวงจรสำหรับเชื่อมต่อกับบอร์ดภายนอก

คุณสมบัติแผงวงจร ET-MINI W5100

- หน่วยของชุดวงจรรวมของคอมพิวเตอร์ เบอร์ W5100 (80 ช่อง LQFP TYPE) ของบริษัท WIZnet เป็นวงจรรวม เครือข่ายคอมพิวเตอร์อีเธอร์เน็ต
- คอนโทรเลอร์ซึ่งมี การเดินสาย (Hardwired TCP/IP stack) ในตัว

- รองรับการเชื่อมต่อแบบ ระบบโปรโตคอล การสื่อสารพื้นฐานของระบบอินเตอร์เน็ต หนึ่งในโปรโตคอลหลักในเครือข่ายอินเทอร์เน็ต

- รองรับการเชื่อมต่อแบบ 10BaseT/100 BaseTX

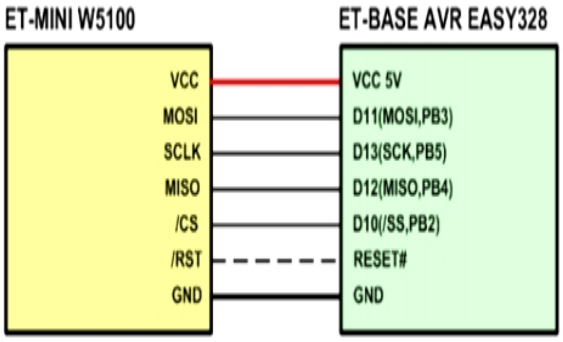
- การเชื่อมต่อกับแผงวงจรเป็นการสื่อสารแบบอนุกรมการส่งข้อมูลตอบและรับระหว่างคอมพิวเตอร์ (SPI:Serial Peripheral Interface) สามารถนำไปต่อกับแผงวงจรคอนโทรลเลอร์

- ใช้ไฟเลี้ยงได้ทั้ง 3.3 โวลต์ และ 5 โวลต์ โดยเลือกจากตัวเชื่อมสัญญาณไฟฟ้า (Jumper)

- ใช้งานร่วมกับแผงวงจร ET-BASE AVR EASY88/168/328 โดยตรง

2.3.2 การต่อใช้งาน ET-MINI W5100 ร่วมกับแผงวงจร ET-BASE AVR EASY328

การใช้งานแผงวงจร ET-MINI W5100 สามารถเชื่อมต่อกับไมโครคอนโทรลเลอร์ผ่านทางขา SPI ของไมโครคอนโทรลเลอร์การต่อใช้งานร่วมกับแผงวงจรซึ่งไดอะแกรมการเชื่อมต่อซึ่งการประกอบแผงวงจรทั้งสองสามารถนำแผงวงจร ET-MINI W5100 เสียบขั้วสำหรับเชื่อมต่อกับสายนำสัญญาณของแผงวงจร ET-BASE AVR EASY328 ดังรูปที่ 2.13



รูปที่ 2.13 การต่อใช้งานแผงวงจร ET-MINI W5100

จากรูปที่ 2.13 แสดงไดอะแกรมการเชื่อมต่อแผงวงจร ET-MINI W5100 กับแผงวงจร ET-BASE AVR EASY328

2.3.3 จุดเชื่อมต่อแบบไร้สาย (Access Point)

ทำหน้าที่เป็นจุดกระจายและเชื่อมต่อสัญญาณไร้สาย เพื่อเชื่อมต่ออุปกรณ์ไร้สายทุกชนิดเข้าด้วยกัน ทำหน้าที่เป็นจุดเชื่อมต่อการสื่อสารให้สามารถติดต่อสื่อสารกันได้โดยจุดเชื่อมต่อที่ใช้ในงานวิจัยนี้ใช้จุดเชื่อมต่อไร้สายของ ALPA รุ่น AWAP-411 ดังรูปที่ 2.14

รูปที่ 2.14 จุดเชื่อมต่อแบบไร้สาย ALFA รุ่น AWAP-411 [8]

คุณสมบัติ จุดเชื่อมต่อแบบไร้สาย

- รองรับ การขยายสัญญาณ (Repeater)
- รองรับโปรโตคอลที่อยู่ระดับบนในชั้นการประยุกต์
- การเชื่อมโยงเครือข่ายคอมพิวเตอร์ถึงกันทั้งหมดโดยอาศัยครือข่ายท้องถิ่น (LAN: Local Area Network)
- ระบบควบคุมการผ่านเข้าออก (ACL: Access Control)

2.4 สร้างเว็บเพจแบบเสมือนจริง (Adobe Dreamweaver CS5)

Dreamweaver คือโปรแกรมสร้างเว็บเพจแบบเสมือนจริง[7] ของค่าย Adobe ซึ่งช่วยให้ผู้ที่ต้องการสร้างเว็บเพจไม่ต้องเขียนภาษาหลักที่ใช้ในการเขียนเว็บเพจ (HTML : Hyper Text Markup Language) คำสั่งโปรแกรมหรือที่ศัพท์เทคนิคและโปรแกรม Dreamweaver มีความสัมพันธ์ที่ทำให้ผู้ใช้สามารถจัดวางข้อความ,รูปภาพ,ตาราง,ฟอร์ม,และวิดีโอ รวมถึงองค์ประกอบอื่นภายในเว็บเพจผู้ใช้ต้องการ โดยไม่ต้องใช้ภาษาสคริปต์ Dreamweaver มีทั้งในระบบปฏิบัติการลีนุกซ์และไมโครซอฟต์วินโดวส์ Dreamweaver ดังรูปที่ 2.15

รูปที่ 2.15 Adobe Dreamweaver CS5.5

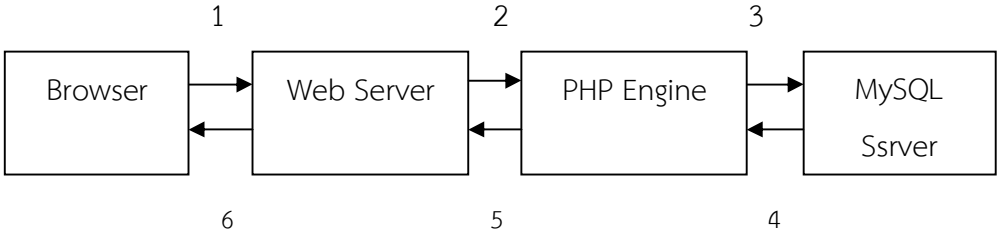
คุณสมบัติ Adobe Dreamweaver CS5.5

Adobe Dreamweaver CS5.5 สามารถสนับสนุนการสร้างเว็บเพจเทคโนโลยี DHTML (Dynamic HTML) และ XML (Extensible Markup Language) สามารถใส่ส่วนเพิ่มเติม เช่น Active X Applet และ สามารถใช้ภาษาที่ใช้เป็นส่วนของการจัดรูปแบบการแสดงผล (CSS : Cascading Style Sheet) ซึ่งเป็นเทคโนโลยีใหม่ของการสร้างเว็บเพจเพื่อให้เว็บเพจ Dreamweaver พื้นฐานเกี่ยวกับการสร้างโฮมเพจ

2.5 การออกแบบฐานข้อมูล (Data Base Design)

การสร้างเว็บเซิร์ฟเวอร์สำเร็จรูป[6] (AppServ) คือโปรแกรมที่รวบรวมเอาซอฟต์แวร์ลิขสิทธิ์ที่มีซอฟต์แวร์ที่เปิดเผยหลักการหรือแหล่งที่มา (Open Source Software) รวมกันมี ส่วนประกอบทั้งหมด (Package) หลัก ดังนี้

- ซอฟต์แวร์สำหรับเปิดให้บริการเซิร์ฟเวอร์บนโปรโตคอล (Apache)
- โปรแกรมช่วยสร้างเว็บไซต์ต่างๆ (PHP: Personal Home Page Too)
- ระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (MySQL: My Structured Query Language)
- ส่วนต่อประสานที่สร้างโดยภาษาพีเอชพี ซึ่งใช้จัดการฐานข้อมูล (phpMyAdmin) มีโครงสร้างดังรูปที่ 2.16



รูปที่ 2.16 การสร้างเว็บเซิร์ฟเวอร์สำเร็จรูป

ตำแหน่งที่ 1 ซอฟต์แวร์โปรแกรมคอมพิวเตอร์เข้าถึงข้อมูลและติดต่อสื่อสารกับระบบสารสนเทศที่อยู่ในรูปแบบของเว็บเพจ (Web Page) ทำคำขอโปรโตคอลในระดับชั้นโปรแกรมประยุกต์ (HTTP: Hypertext Transfer Protocol) สำหรับเว็บเพจ

ตำแหน่งที่ 2 เครื่องคอมพิวเตอร์ซึ่งให้บริการที่เก็บเว็บไซต์รับคำขอสำหรับดึงไฟล์และส่งผ่านการประมวลผล

ตำแหน่งที่ 3 PHP Engine กระจายสคริปต์ ภายในสคริปต์ คำสั่งการเชื่อมต่อกับฐานข้อมูล และ การประมวลผลด้วยภาษาสอบถามเปิดการเชื่อมต่อ โปรแกรมระบบจัดการฐานข้อมูล

ตำแหน่งที่ 4 โปรแกรมระบบจัดการฐานข้อมูลรับการนำข้อมูลจากฐานข้อมูลมาแสดงออกทาง จอภาพ และประมวลผลจากนั้นส่งผลลัพธ์ไปยังโปรแกรมเพื่อจะทำหน้าที่แปลภาษา PHP

ตำแหน่งที่ 5 โปรแกรมที่จะทำหน้าที่แปลภาษา PHP เสร็จสิ้นการเรียกใช้ สคริปต์เกี่ยวข้องกับการจัดรูปแบบผลลัพธ์กลับไปยัง PHP engine ส่งออกผลลัพธ์ภาษาหลักที่ใช้ในการเขียนเว็บเพจไปยังเครื่องแม่ข่าย

ตำแหน่งที่ 6 เครื่องคอมพิวเตอร์ซึ่งให้บริการที่เก็บเว็บไซต์ส่งผ่านภาษาหลักที่ใช้ในการเขียนเว็บเพจกลับไปยังโปรแกรมซอฟต์แวร์ที่ใช้ท่องเว็บหรือใช้ดูข้อมูลที่อยู่ในเว็บไซต์

บทที่ 3
ขั้นตอนการดำเนินงาน

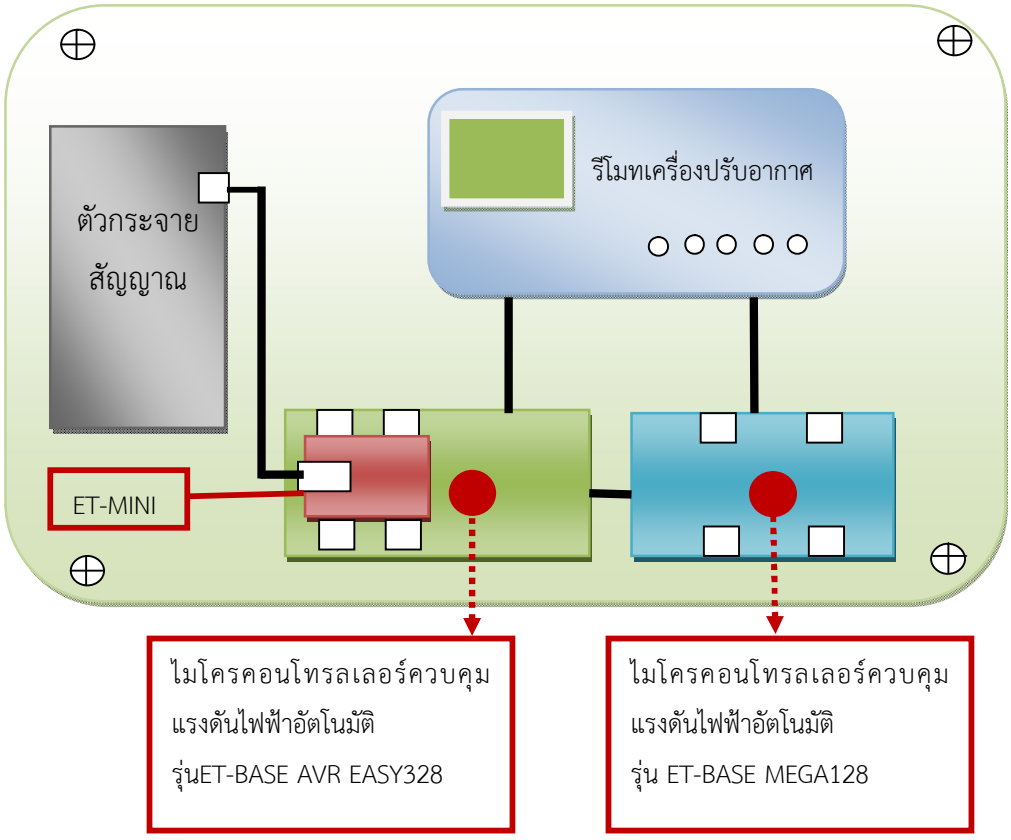
ในการดำเนินงานทำการพัฒนา ระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ มีการวางแผนและปฏิบัติงานโดยมีการแบ่งการทำงานออกเป็น 2 ส่วน คือ

3.1 การดำเนินงานส่วนตัวเครื่อง

3.2 การดำเนินงานส่วน โปรแกรม

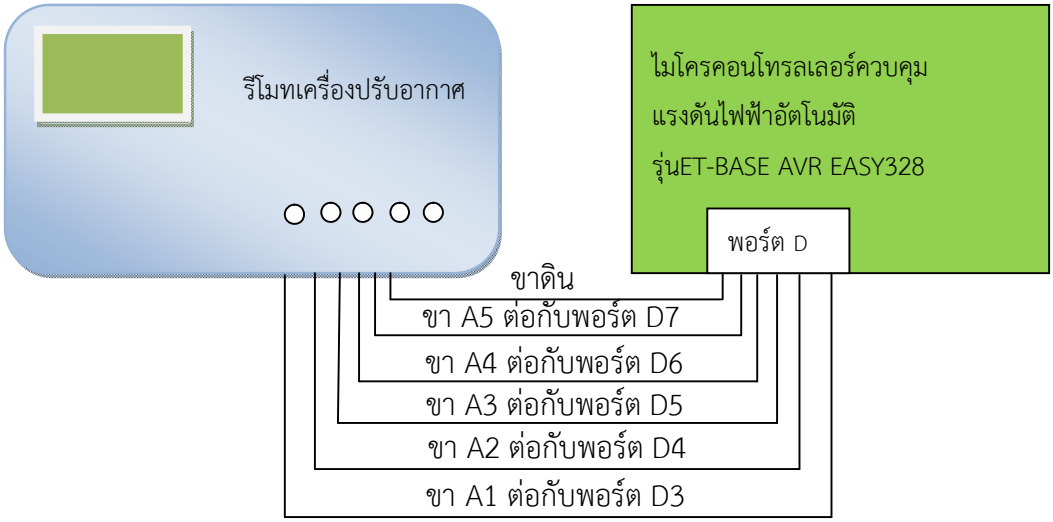
3.1 การดำเนินงานส่วนตัวเครื่อง

การออกแบบโครงสร้าง และสร้างการพัฒนาระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บแบบตามลำดับเพื่อนำเทคโนโลยีที่สำหรับการควบคุมมาใช้เพื่อควบคุมเครื่องปรับอากาศ และทำการออกแบบ อุปกรณ์แสดงภาพ ดังรูปที่ 3.1



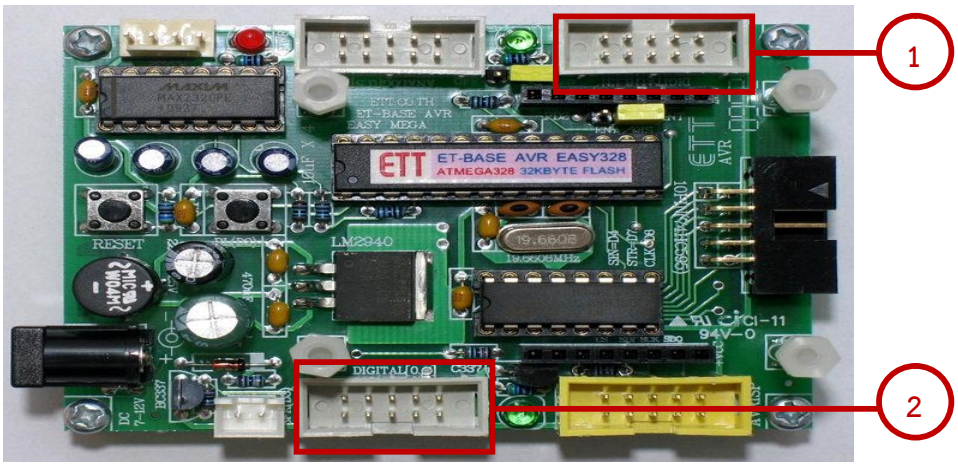
รูปที่ 3.1 แบบเครื่องระบบควบคุมเครื่องปรับอากาศ

3.1.1 ส่วนของการเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328 ดังรูปที่ 3.2



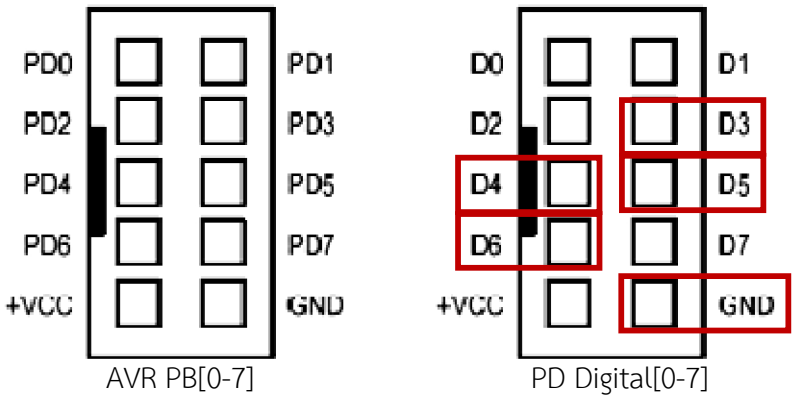
รูปที่ 3.2 ส่วนของจุดเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับไมโครคอนโทรลเลอร์

จากรูป 3.2 พอร์ต D หมายเลข 1 และ 2 ต่อกับแผงวงจรรีโมทเครื่องปรับอากาศเพื่อที่ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328 สั่งงานที่สวิตช์ของรีโมทเครื่องปรับอากาศ ดังรูปที่ 3.3



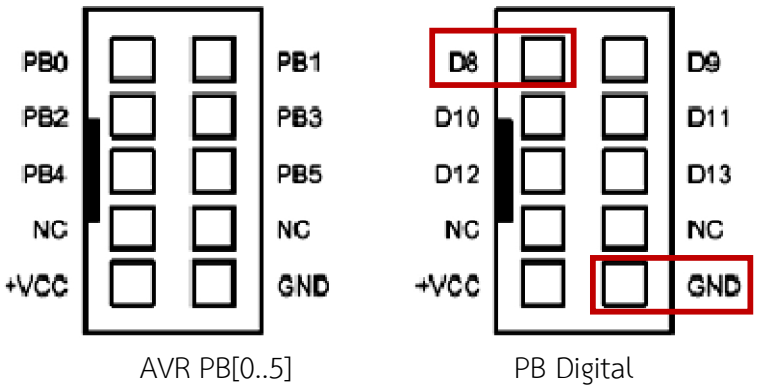
รูปที่ 3.3 ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328

จากรูป 3.3 หมายเลข 2 เป็นขั้วต่อสัญญาณจาก PD[0-7] ซึ่งในกรณีการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328 เป็นขาสัญญาณของ Digital [0-7] ดังรูปที่ 3.4



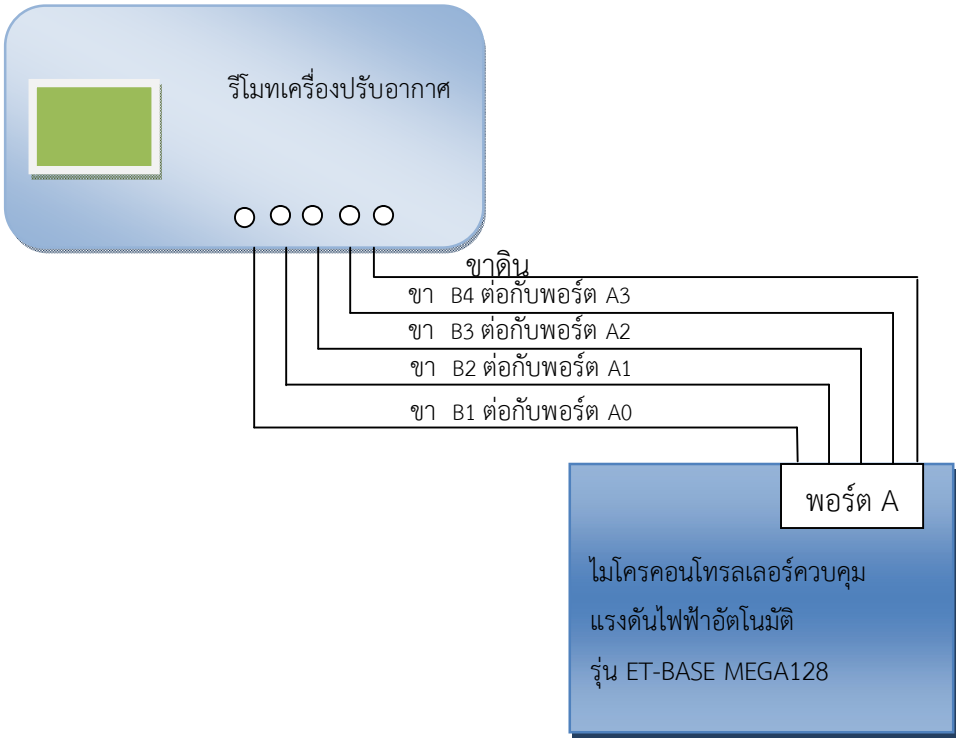
รูปที่ 3.4 ขั้วต่อสัญญาณจาก Digital[0-7]

จากรูป 3.3 หมายเลข 1 เป็นขั้วต่อสัญญาณจาก PD[0-5] ซึ่งในงานวิจัยนี้ใช้ขาสัญญาณของ Digital[8-13] ดังรูปที่ 3.5



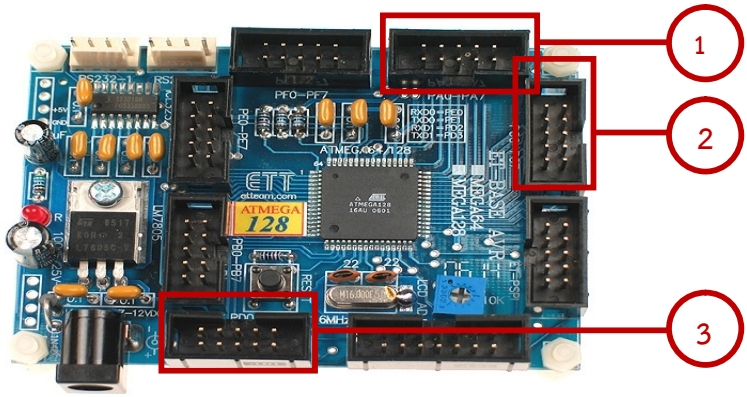
รูปที่ 3.5 ขั้วต่อสัญญาณ Digital[8-13]

3.1.2 ส่วนของการเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติส่วนของจุดเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ ดังรูปที่ 3.6



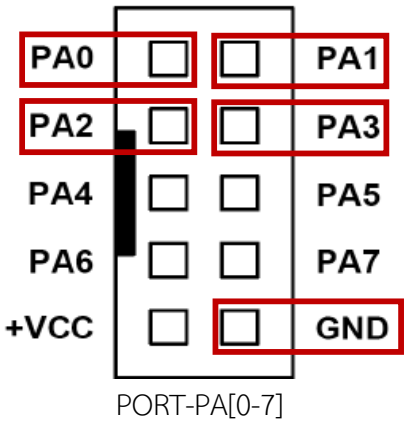
รูปที่ 3.6 แผงวงจรรีโมทเครื่องปรับอากาศ

จากรูปที่ 3.6 พอร์ต A หมายเลข 1 ต่อกับรีโมทเครื่องปรับอากาศเพื่อใช้อ่านค่าจากขาที่ไปสั่งงานของสวิตซ์ตัดต่อวงจรความเร็วพัดลมและคอมเพรสเซอร์หมายเลข 2 และ 3 ใช้สำหรับต่อกับรีโมทเครื่องปรับอากาศเพื่อต่อกับวงจรขับ 7Seg ตัวหน้าและตัวหลัง ดังรูปที่ 3.7



รูปที่ 3.7 ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128

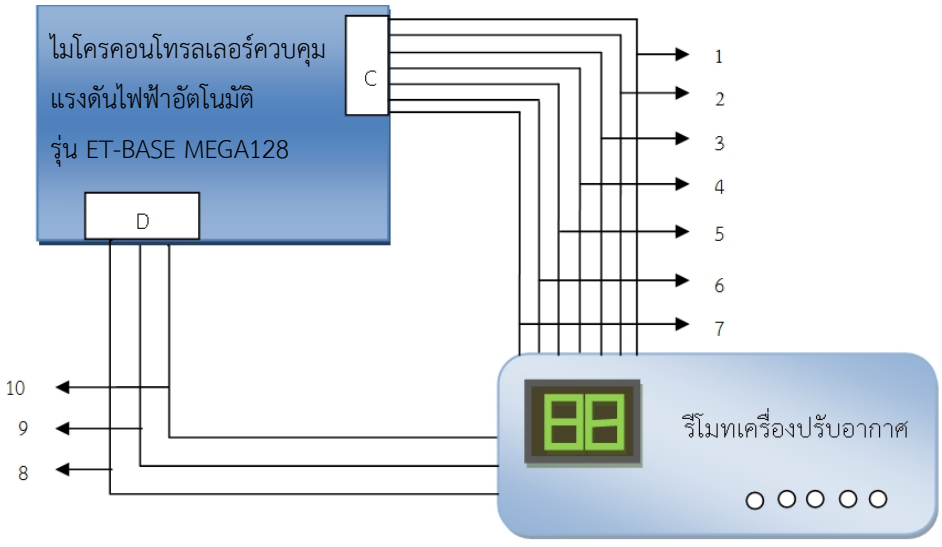
จากรูป 3.7 หมายเลข 1 เป็นขั้วต่อสัญญาณจาก PA [0-7] ซึ่งในกรณีการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ เป็นขาสัญญาณของ Digital[0-7] ดังรูป 3.8



รูปที่ 3.8 ขั้วต่อสัญญาณจาก Digital[0-7]

3.1.2.1 ส่วนของการเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ

พอร์ต D0 ต่อกับขาวงจรขับ 7seg ตัวหน้าพอร์ต D1 ต่อกับขาวงจรขับ 7seg ตัวหลัง และพอร์ต C ไปอ่านค่าที่สแกนตัวเลขบน 7seg ไม่ต่อพอร์ต C5 ทำให้รีโมทเครื่องปรับอากาศตอนเปิดเครื่องโหมดไม่ปกติเมื่อเอาพอร์ต C5 ออกยังได้ข้อมูลจากของตัวเลขแต่ละตัวได้ไม่ซ้ำกัน ดังรูปที่ 3.9



รูปที่ 3.9 แผงวงจรรีโมทเครื่องปรับอากาศ

จากรูป 3.9 เป็นจุดเชื่อมต่อพอร์ต C และ D ที่ใช้ตามหมายเลขต่อไปนี้

ตำแหน่งที่ 1 C0 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

ตำแหน่งที่ 2 C1 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

ตำแหน่งที่ 3 C2 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

ตำแหน่งที่ 4 C3 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

ตำแหน่งที่ 5 C4 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

ตำแหน่งที่ 6 C6 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

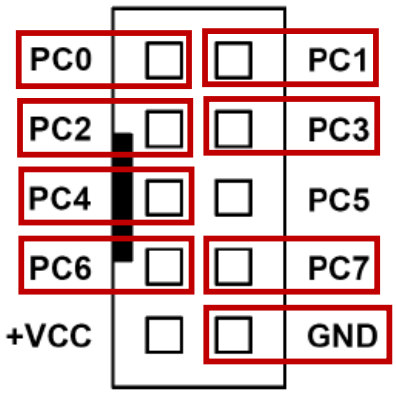
ตำแหน่งที่ 7 C7 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต C

ตำแหน่งที่ 8 D0 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต D

ตำแหน่งที่ 9 D1 ต่อกับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ พอร์ต D

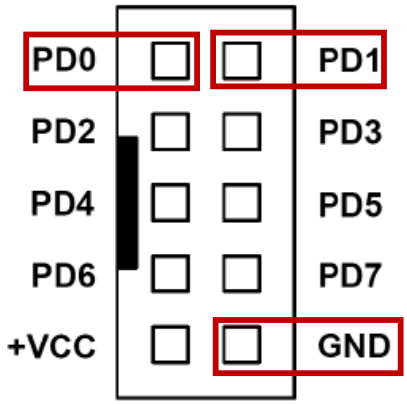
ตำแหน่งที่ 10 GROUND

ซึ่งจากรูป 3.7 หมายเลข 2 เป็นขั้วต่อสัญญาณจาก PC[0-7] โดยในกรณีการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ ใช้ขาสัญญาณของ Digital[0-7] ดังรูป 3.10



รูปที่ 3.10 ขั้วต่อสัญญาณจาก Digital[0-7]

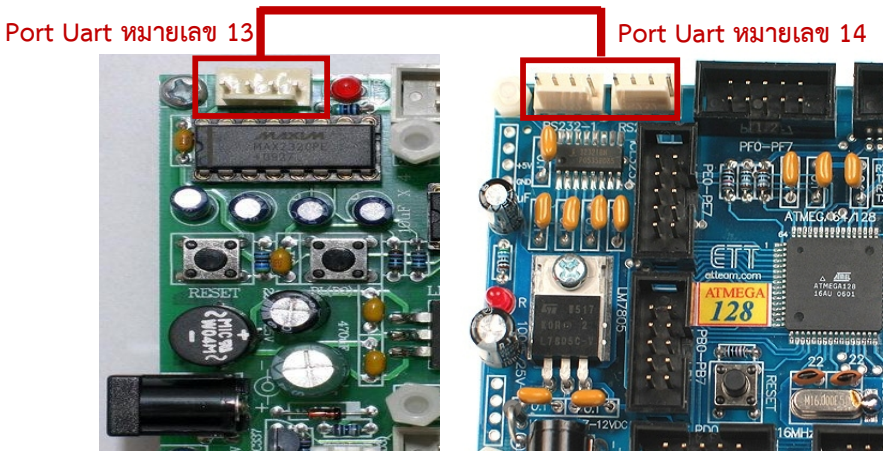
จากรูป 3.10 หมายเลข 3 เป็นขั้วต่อสัญญาณจาก PD[0-7] โดยในกรณีการพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ เป็นขาสัญญาณของ Digital[0-7] ดังรูป 3.7



รูปที่ 3.11 ขั้วต่อสัญญาณจาก Digital[0-7]

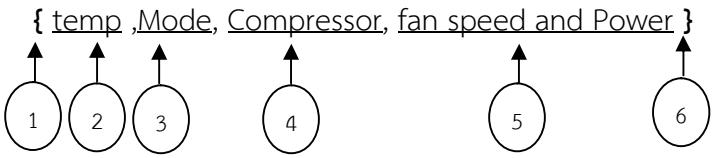
3.1.3 ส่วนของไมโครคอนโทรลเลอร์แรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328 กับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128

จากหมายเลข 13 และ 14 ใช้เพื่อเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์แรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY328 กับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128 โดยที่ไมโครคอนโทรลเลอร์แรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128 เป็นตัวส่งข้อมูลให้กับไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY3 EASY328 พอร์ต Uart ดังรูปที่ 3.13



รูปที่ 3.12 หมายเลข 13 และ 14 คือพอร์ต Uart

จากรูปที่ 3.12 ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE MEGA128 เป็นตัวส่ง ให้ ไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY3 EASY328 อย่างเดียวสองแผงวงจรมีการคุยกันด้วยพอร์ต Uart คือการส่งข้อมูล แบบอนุกรม ประกอบด้วย



ตำแหน่งที่ 1 ‘{‘ เพื่อแจ้งว่าเริ่มการส่งข้อมูล

ตำแหน่งที่ 2 Temp เป็นการส่งค่าของอุณหภูมิที่อ่านได้ Data[1]

ตำแหน่งที่ 3 Mode เป็นการส่งค่าของการอ่านโหมดว่าเป็นโหมดใด Data[2]

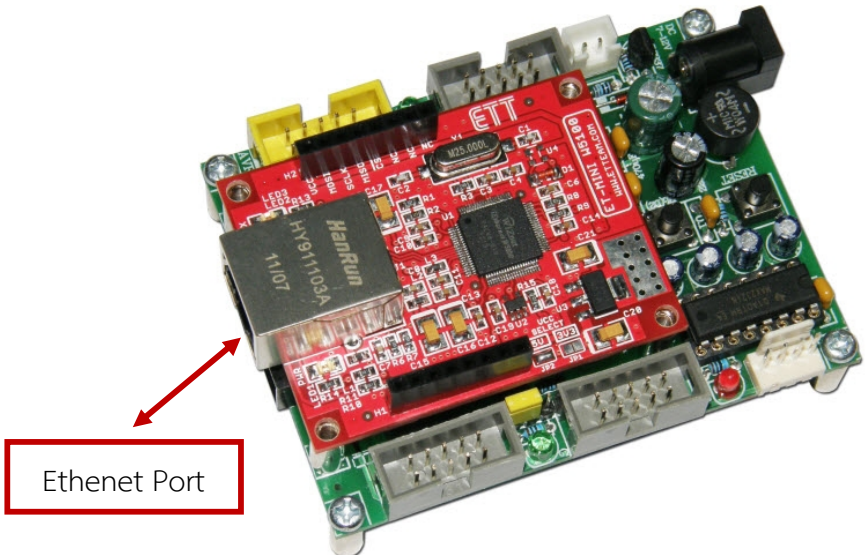
ตำแหน่งที่ 4 คอมเพรสเซอร์ เป็นการส่งค่าว่า Compressor ทำงาน หรือ ไม่ Data[3]

ตำแหน่งที่ 5 fan speed and Power ส่งระดับความเร็วถ้าระดับความเร็วที่อ่านมาไม่มีระดับความเร็วแสดงว่าเครื่องปิด Data[4]

ตำแหน่งที่ 6 ส่ง ‘}’ เพื่อแจ้งว่าสิ้นสุดการส่งข้อมูล

3.1.4 ส่วนของแผงวงจรเชื่อมต่ออีเธอร์เน็ต

เป็นตัวกลางในการเชื่อมต่อการรับส่งข้อมูลระหว่างไมโครคอนโทรลเลอร์ควบคุมแรงดันไฟฟ้าอัตโนมัติ รุ่น ET-BASE AVR EASY3 EASY328 ไปยังแผงวงจรรับส่งสัญญาณอินเตอร์เน็ต ดังรูปที่ 3.13

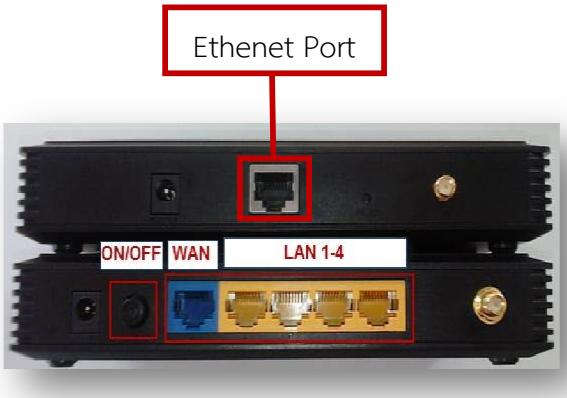


รูปที่ 3.13 แผงวงจรเชื่อมต่ออีเธอร์เน็ต

จากรูป 3.13 รับและส่งสัญญาณข้อมูลการควบคุมระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บไปยังแผงวงจรรับส่งสัญญาณอินเตอร์เน็ต

3.1.5 ส่วนของแผงวงจรรับส่งสัญญาณอินเตอร์เน็ต

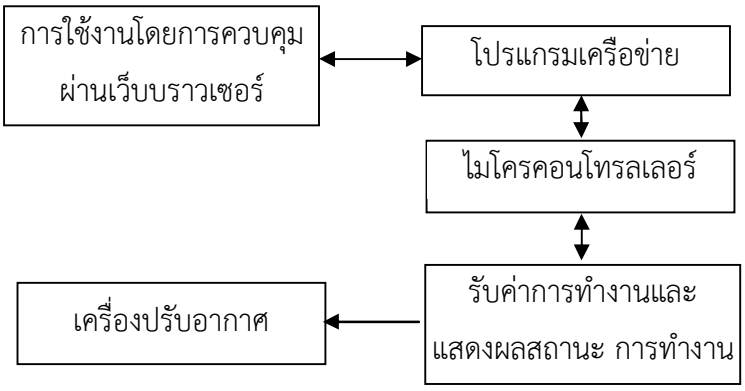
รับส่งสัญญาณจากแผงวงจรไมโครคอนโทรลเลอร์อาดุยโน่จากระบบอีเธอร์เน็ตให้เป็นระบบเครือข่ายสัญญาณแบบไร้สาย ดังรูปที่ 3.14



รูปที่ 3.14 พอร์ตเชื่อมต่อระหว่าง ET-Mini กับแผงวงจรระบบเครือข่ายสัญญาณแบบไร้สาย

3.2 กระบวนการทำงานของระบบ

การทำงานของระบบสามารถออกแบบแผนภาพความสัมพันธ์ ดังรูปที่ 3.15

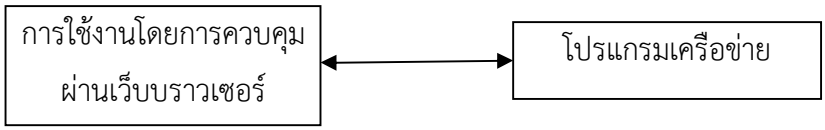


รูปที่ 3.15 แผนภาพความสัมพันธ์การทำงานของระบบ

จากรูป 3.15 แผนภาพความสัมพันธ์การทำงานของระบบสามารถอธิบายการทำงานของส่วนต่างๆ ดังนี้

3.3.1 การใช้งานโดยการควบคุมผ่านเว็บบราวเซอร์

ทำหน้าที่ในการคอนโทรลคำสั่งให้เครื่องปรับอากาศทำงาน โดยมีการทำงานดังรูปที่ 3.16



รูปที่ 3.16 การใช้งานโดยการควบคุมผ่านเว็บบราวเซอร์

จากรูป 3.16 การใช้งานโดยผ่านการควบคุมผ่านหน้าเว็บบราวเซอร์ โดยส่งข้อมูลไปยังไมโครคอนโทรลเลอร์เพื่อทำการประมวลผลของการควบคุม

ส่วนที่ 1 ส่วนของการเซต IP address หน้าเว็บดังนี้

```
#define refresh_web()                          // การสร้าง IP address
client.print("<meta http-                     // พิมพ์หน้าเว็บเป็น http
equiv=\"refresh\"                              // รีเฟลชหน้าเว็บ
content=\"1;url=http://192.168.1.201/\"/>      //ที่อยู่หน้าเว็บ
")
IPAddressip(192,168,1,201);                    //ที่อยู่ไอพีบนหน้าเว็บ
```

ส่วนที่ 2 ส่วนของการส่งข้อมูลส่งข้อมูลที่ละตัวโดยเริ่มจากการส่ง “{”

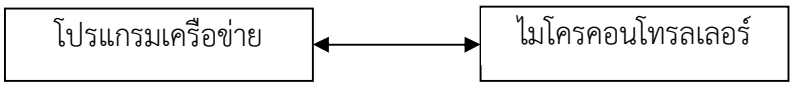
```
else
  {   if(temp_receiver == '}                   //เมื่อได้ ‘}’ แสดงว่าสิ้นสุด Package ก็ทำการ
                                                 เก็บค่าลง register ต่างเพื่อเอาไปประมวลผล
                                                 ต่อ ไป
                  {
temp_val = data_receiver[0];                   //เก็บอุณหภูมิไว้ใน Data0
mode_val = data_receiver[1];                   //เก็บโหมดไว้ใน Data1
com_val = data_receiver[2];                    //การทำงานเครื่องอัดไอเก็บไว้ใน Data2
fan_power_val = data_receiver[3];              //ความเร็วพัดลมเก็บไว้ใน Data3
count_addrUart = 0;                            //การส่งข้อมูลทางพอร์ต Uart
```

```
flag_receiver_start = 0;}                    // เมื่อ พบ‘{‘เป็นการเก็บข้อมูลลงใน Array

                    {
 data_receiver[count_addrUart] = (temp_receiver - 48);    //ถ้ามีการรับค่าแปลกๆหรือไม่
                                                          ครบรูปแบบ package ก็ให้
                                                          Reset และรอรับ ‘{‘ ใหม่
 count_addrUart++;                              //ส่งข้อมูลออกพอร์ต Uart
 }
 if(count_addrUart> 5)
                    {
 count_addrUart = 0;
 flag_receiver_start =  } } }
```

จากชุดคำสั่งโปรแกรมด้านบนคือ การส่งข้อมูลส่งข้อมูลที่ละตัวโดยเริ่มจากการส่ง “{” และเก็บมูล temp , mode , com , fan power ไว้ใน Data และเมื่อส่ง “}”แสดงว่าสิ้นสุดการส่งข้อมูล

3.3.2 โปรแกรมเครือข่าย

ทำเชื่อมต่อเครือข่ายของไมโครคอนโทรลเลอร์เครื่องผู้ใช้บริการโดยมีการทำงาน ดังรูปที่ 3.17



รูปที่ 3.17 แผนภาพความสัมพันธ์ส่วนของการรับคำสั่งของการใช้งาน

จากรูป 3.17 แผนภาพความสัมพันธ์ส่วนของการรับคำสั่งการใช้งานไมโครคอนโทรลเลอร์ไปยังการตรวจสอบสถานะอุปกรณ์ในการทำงาน

ส่วนที่ 3 เป็นการเซ็ตพอร์ต Uart 9600 เป็น 8 บิต

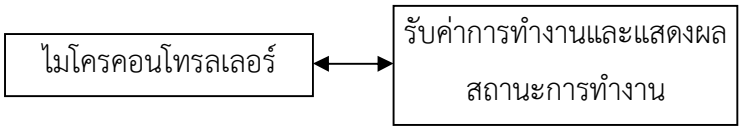
```
VoidUSART_Init(void)                          //การเซต Uart 9600 เป็น8 บิต
{
   UBRR0L = BAUD_PRESCALE;
   UBRR0H = (BAUD_PRESCALE >> 8);             //baudrate 9600 bps
  UCSR0B = (1<<TXEN0);                        //Tx Enable
```

```
  UCSR0C = (3<<UCSZ00)                          // 8 bit sending
}
voidUSART_Transmit( unsigned char data )      //ฟังก์ชั่นในการส่ง Data ออกทาง Uart ที
                                               ละ 1 byte
{
while ( !( UCSR0A & (1<<UDRE0)) );
UDR0 = data;                                  //ฟังก์ชั่นในการส่ง Data ออกทาง Uart ทีละ
1 byte
}
```

จากชุดคำสั่งโปรแกรมด้านบนเป็นการเซตพอร์ต Uart 9600 เป็น 8 บิต และเป็นคำสั่งการส่งข้อมูลออกทาง Uart

3.3.3 ตรวจสอบสถานะอุปกรณ์ในการทำงาน

ทำหน้าที่ตรวจสอบสถานะอุปกรณ์ในการทำงานของปุ่มคอนโทรลแต่ละปุ่มเพื่อรับค่าการทำงานและแสดงผลสถานะการทำงาน ดังรูปที่ 3.18



รูปที่ 3.18 แผนภาพความสัมพันธ์ส่วนตรวจสอบสถานะอุปกรณ์ในการทำงาน

จากรูป 3.18 แผนภาพความสัมพันธ์ส่วนตรวจสอบสถานะอุปกรณ์ในการทำงานเพื่อไปยังการควบคุมเครื่องปรับอากาศ

ส่วนที่ 4 เป็นการตรวจหา Start Pulse ดักจับสัญญาของ 7seg ตัวหน้าที่ PIND 0 และตัวหลังที่ PIND 1

```
External interrupt0
ISR(INT0_vect)                                // ดักจับสัญญาณ Scan 7seg ตัว หน้าที่
                                               PIND0
    {
    if(flag_start == 0)                       // ถ้ายังไม่เจอ Start pulse ของสัญญาณ scan
                                               ก็ให้หา start ก่อน
    {
```

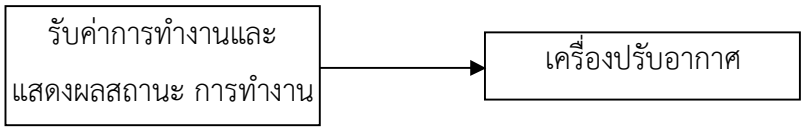
```
        _delay_us(100);                 //หน่วงเวลา 100 วินาที เพื่อหา Start pulse
                if(READ_LOW_D0)         // อ่านว่าเป็น start pulse หรือไม่
                {
EICRA &= ~(1<<ISC00);                   // กำหนด PD0 เกิด interrupt ที่ ขอบขาลง
                        flag_start = 1;
temp_data_digit_front = 0xFF;           //เก็บข้อมูล 7seg ตัวหน้าไว้ใน Digit front
temp_data_digit_black = 0xFF;           //เก็บข้อมูล 7seg ตัวหลังไว้ใน Digit black
count_data8 = 0;
                }}
        Else                            // เมื่อเจอ start pulse แล้วทำอ่านค่าจาก
                                        PORTC ทั้ง หมด 8 ครั้ง(0 - 7)
        {
        temp_data_digit_front&= PINC;
                count_data8++;          // ถ้าครบแล้วจัดเก็บ Data ลง
                                        data_digit_front เพื่อเอาไปประมวลผลต่อไป
                                        ใน Main function

if(count_data8 == 8)  {
DIS_INT0();                             // external input interrupt PD0 Disable
EN_INT1();                              external input interrupt PD1 Enable
data_digit_front =
temp_data_digit_front& 0b11001111;      //นำเอาข้อมูลที่ได้จาก 7seg มาคูณกับ
                                        11001111
                }}}
External interrupt1                     // ดักจับ สัญญาณ Scan 7seg ตัวหลังที่ PIND1
ISR(INT1_vect)
{
temp_data_digit_black&= PINC;
                count_data8++;          // ทำอ่านค่าจาก PORTC ทั้งหมด 8 ครั้ง (8 -
                                        15)
                if(count_data8 == 16)   // เมื่อ count ครบ 16 ทำการรีเซตค่าเพื่อรอ
                                        การ scan ของ 7segment อีกครั้ง
```

```
                {
                        DIS_INT1();             / /external input interrupt PD1 Disable
data_digit_black =                              // ถ้าครบแล้วก็จัดเก็บ Data ลง
temp_data_digit_black& 0b11010111;              data_digit_black เพื่อไปประมวลผลต่อไปใน
                                                Main function
flag_start = 0;
EICRA |= (1<<ISC00);                            // กำหนด PD0 เกิดการขัดจังหวะที่ขอบขาขึ้น
                        EN_INT0();              // external input interrupt PD0 Enable
        }}
```

จากชุดคำสั่งโปรแกรมด้านบนเป็นการดักจับสัญญาณ 7seg ตัวหน้าและตัวหลังและเก็บข้อมูล 7seg ตัวหน้าไว้ใน Digit front และ 7seg ตัวหลังไว้ใน Digit black และนำไปประมวลผลใน Main function

3.3.4 รับค่าการทำงานและแสดงผลสถานะ การทำงาน

ทำหน้าที่ รับค่าการทำงานและแสดงผลสถานการณ์ทำงาน ดังรูปที่ 3.19



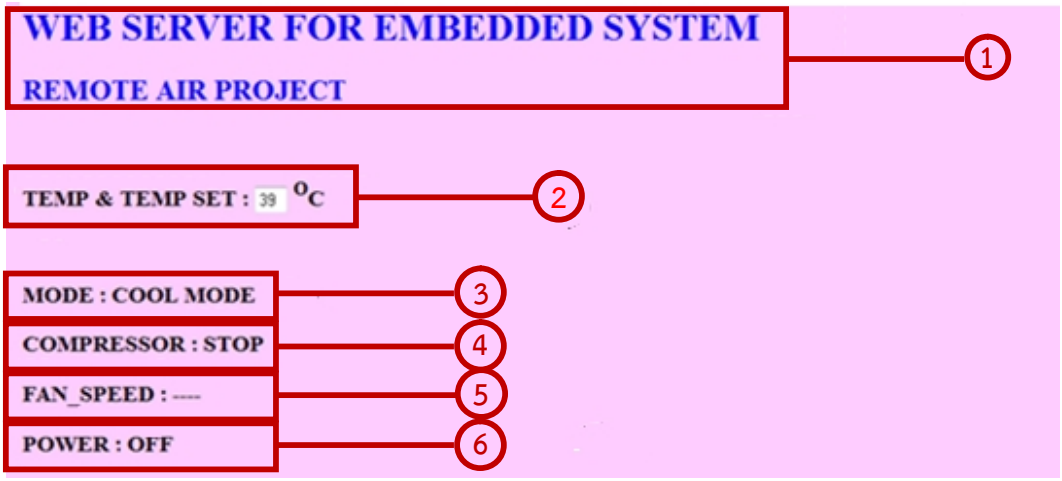
รูปที่ 3.19 แผนภาพความสัมพันธ์ส่วนแสดงผลสถานะของการทำงาน

ส่วนที่ 5 เป็นการแสดงผลบนเว็บบราวเซอร์เป็น html โดยระบายสีให้เห็นว่าแสดงอะไรบ้างบนหน้าเว็บ

```
if (c == '\n' &&currentLineIsBlank) {          //  การแสดงผลบนเว็บบราวเซอร์ เป็น html
client.println("HTTP/1.1 200 OK");
client.println("Content-Type:
text/html");
client.println();
client.println("<html>");                      //พิมพ์หน้าเว็บเป็น html
client.println("<h1><font                      //ขนาดตัวอักษร
```

```
color=\"blue\">WEB SERVER FOR          //สีตัวอักษร WEB SERVER FOR EMBEDDED
EMBEDDED SYSTEM</font></h1>");         SYSTEM
client.println("<h2><font              //ขนาดตัวอักษร
color=\"blue\">REMOTE AIR              //สีตัวอักษร REMOTE AIR PROJECT
PROJECT</font></h2>");
client.print("<FORM action=\"\">");
refresh_web();
client.println("<body bgcolor          // ใส่สีให้หน้าเว็บ
=\"#FFCCFF\"></body>");
sprintf((char *)data,"%u",temp_val);   // ส่งค่าจาก temp เก็บไว้ใน data
        client.print("<strong>TEMP &  // ทำการแสดงผล TEMP & TEMP SET :และ
TEMP SET : <input type=\"text\" size=2 ช่องว่างสี่เหลี่ยม
value=\"");
        client.print(data);            // ทำการพิมพ์ค่าจาก data ลงในช่องสี่เหลี่ยม
        client.print("\"><sup>O</sup>C\
r\n<br>");
```

จากชุดคำสั่งโปรแกรมด้านบน เป็นการแสดงผลบนเว็บบราวเซอร์เป็น html โดยระบายสีแสดงบนหน้าเว็บ ดังรูป 3.20



รูปที่ 3.20 เป็นการแสดงผลบนเว็บบราวเซอร์

จากรูป 3.22 เป็นส่วนการออกแบบหน้าเว็บควบคุม คือ

ตำแหน่งที่ 1 WEB SERVER FOR EMBEDDED SYSTEM REMOTE AIR PROJECT เป็นการแสดงชื่อของโปรเจ็คบนหน้าเว็บ

ตำแหน่งที่ 2 แสดงสถานะของ MODE

ตำแหน่งที่ 3 แสดงสถานะของอุณหภูมิเมื่อทำการปรับเพิ่มหรือลดอุณหภูมิ

ตำแหน่งที่ 4 แสดงสถานของการทำงานคอมเพรสเซอร์

ตำแหน่งที่ 5 แสดงสถานของการทำงานของระดับของแรงของพัดลม

หมายเลข 6 แสดงสถานของการทำงานเปิดปิด

ส่วนที่ 6 แสดงผลสถานะการทำงานโหมดคอมเพรสเซอร์ความเร็วพัดลมมีค่าเท่าไรและออกแบบใส่สีตัวอักษรบนหน้าเว็บ

```
Switch (mode_val)                                   // (mode_val)
        {
case(0):  client.println("<br><h3><font             //แสดงขนาดตัวอักษรและสีของ MODE : FAN
color=\"black\">MODE : FAN                          MODE
MODE</font></h3>");
break;
case(1):  client.println("<br><h3><font             //แสดงขนาดตัวอักษรและสีของ MODE : DRY
color=\"black\">MODE : DRY                          MODE
MODE</font></h3>");
break;
case(2):  client.println("<br><h3><font             //แสดงขนาดตัวอักษรและสีของ MODE : COOL
color=\"black\">MODE : COOL                         MODE
MODE</font></h3>");
break;
default: break;
        }

Switch (com_val)                                    (com_val)
        {
case(0):  client.println("<h3><font                 //แสดงขนาดตัวอักษรและสีของ COMPRESSOR
```

```
color=\"black\">COMPRESSOR  :                    : STOP
STOP</font></h3>");
break;
case(1):  client.println("<h3><font               //แสดงขนาดตัวอักษรและสีของ COMPRESSOR
color=\"black\">COMPRESSOR :                     : RUN
RUN</font></h3>");
break;
default: break;
          }
Switch (fan_power_val)                          // (Fan_Power_Val)
          {
case(0):  client.println("<h3><font               //แสดงขนาดตัวอักษรและสีของ
color=\"black\">FAN_SPEED :                      FAN_SPEED : LOW และ
LOW</font></h3>");
client.println("<h3><font                        แสดง POWER : ON
color=\"black\">POWER :
ON</font></h3>");
break;
case(1):  client.println("<h3><font               //แสดงขนาดตัวอักษรและสีของ
color=\"black\">FAN_SPEED :                      FAN_SPEED : MID และ
MID</font></h3>");
client.println("<h3><font                        POWER : ON
color=\"black\">POWER :
ON</font></h3>");
break;
case(2):  client.println("<h3><font               //แสดงขนาดตัวอักษรและสีของ
color=\"black\">FAN_SPEED :                      FAN_SPEED : HIGH และ
HIGH</font></h3>");
client.println("<h3><font                        POWER : ON
color=\"black\">POWER :
ON</font></h3>");
```

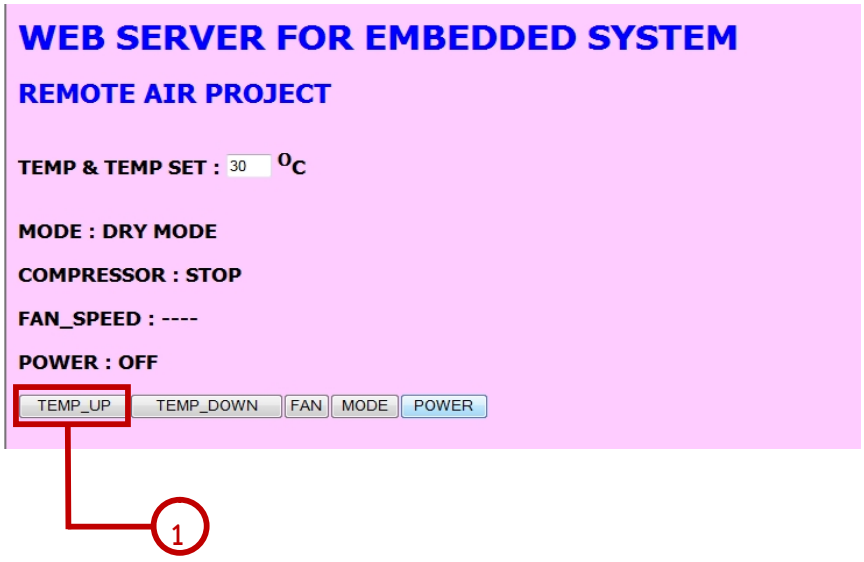
```
break;
case(3):
client.println("<h3><font                    //แสดงขนาดตัวอักษรและสีของ
color=\"black\">FAN_SPEED : ----             FAN_SPEED และ
</font></h3>");
client.println("<h3><font                    POWER  : OFF
color=\"black\">POWER  :
OFF</font></h3>");
break;
default: break; }
```

จากชุดคำสั่งโปรแกรมด้านบนเป็นการแสดงขนาดตัวอักษรและสีของและพิมพ์หน้าเว็บ

ส่วนที่ 6 เป็นการสร้างปุ่ม Button ชื่อ TEMP_UP, TEMP_DOWN, FAN, MOD , POWER

```
client.print("<P><INPUT type=\"submit\"  name=\"status\"  value=\"TEMP_UP\">");
```

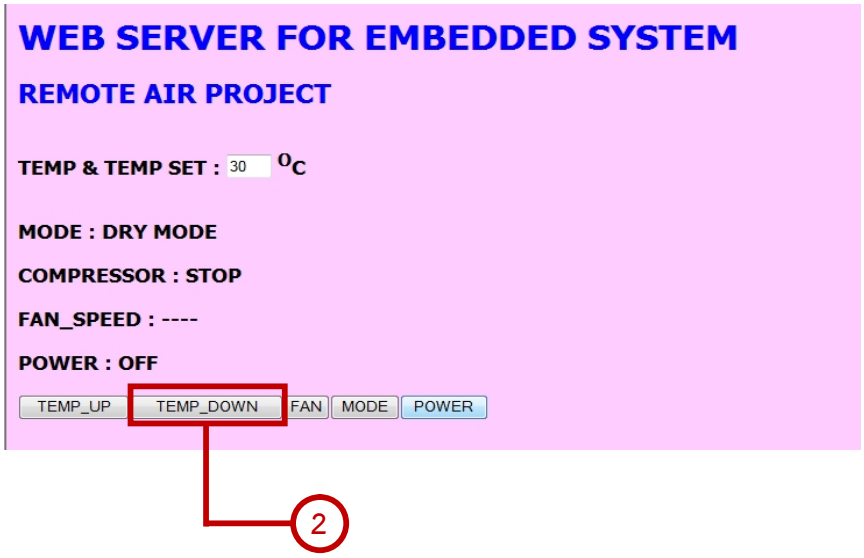
จากชุดคำสั่งโปรแกรมด้านบนตำแหน่งหมายเลข 1 เป็นการสร้างปุ่ม Button ชื่อ TEMP_UP เพื่อใช้ในการควบคุมแทนการกดที่รีโมทเครื่องปรับอากาศและพิมพ์หน้าเว็บ ดังรูปที่ 3.21



รูปที่ 3.21 ปุ่มควบคุม TEMP_UP

```
client.print("<INPUT type=\"submit\"  name=\"status\"  value=\"TEMP_DOWN\">");
```

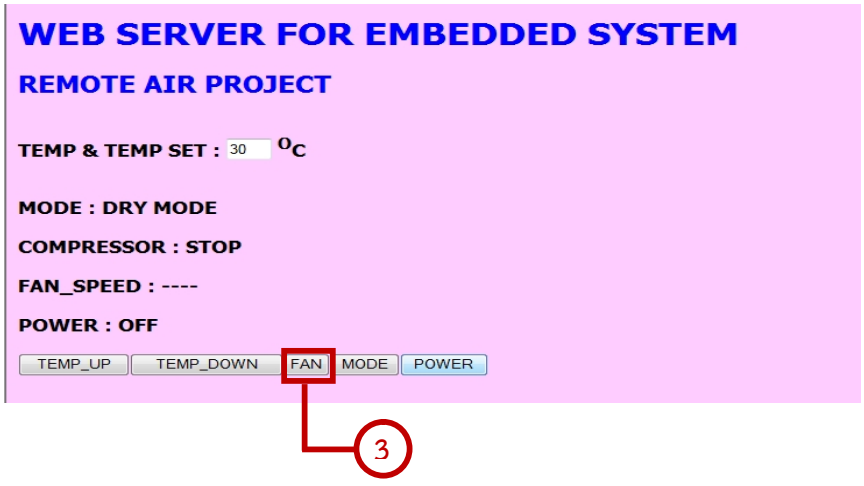
จากชุดคำสั่งโปรแกรมด้านตำแหน่งหมายเลข 2 เป็นการสร้างปุ่ม Button ชื่อ TEMP_DOWN เพื่อใช้ในการควบคุมแทนการกดที่รีโมทเครื่องปรับอากาศและพิมพ์หน้าเว็บ ดังรูปที่ 3.22



รูปที่ 3.22 ปุ่มควบคุม TEMP_DOWN

client.print ("<INPUT type=\"submit\" name=\"status\" value=\"FAN\">");

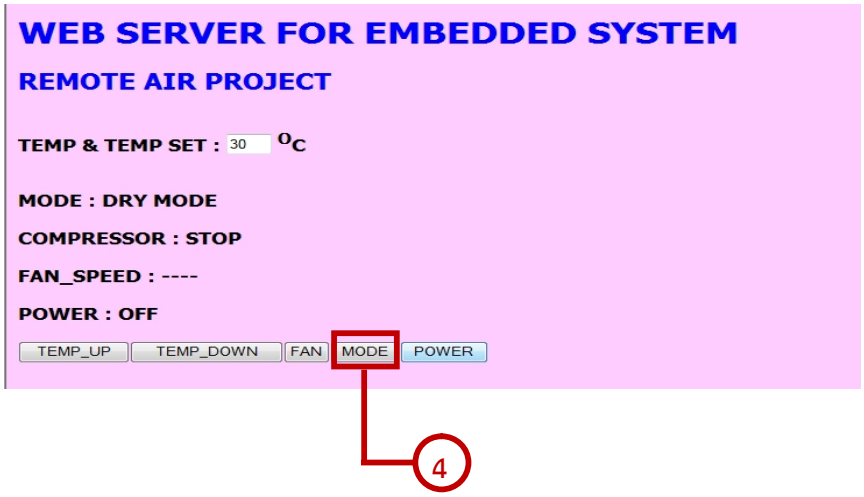
จากชุดคำสั่งโปรแกรมด้านตำแหน่งหมายเลข 3 เป็นการสร้างปุ่ม Button ชื่อ FAN เพื่อใช้ในการควบคุมแทนการกดที่รีโมทเครื่องปรับอากาศ และพิมพ์หน้าเว็บ ดังรูปที่ 3.23



รูปที่ 3.23 ปุ่มควบคุม FAN

client.print("<INPUT type=\"submit\" name=\"status\" value=\"MODE\">");

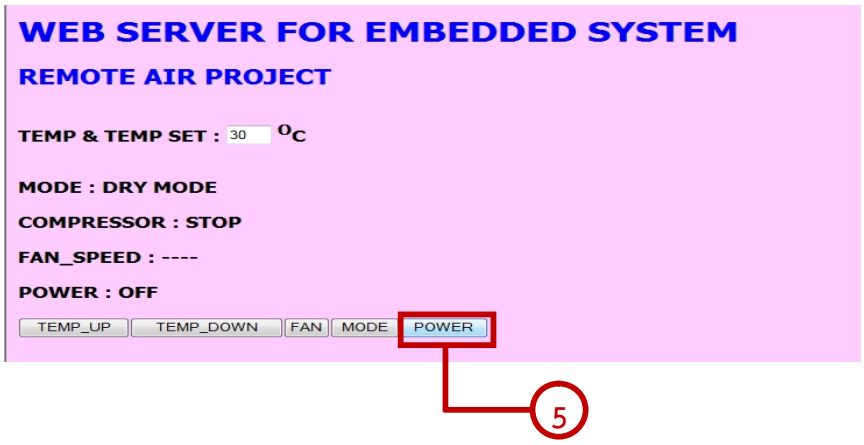
จากชุดคำสั่งโปรแกรมด้านตำแหน่งหมายเลข 4 เป็นการสร้างปุ่ม Button ชื่อ MODE เพื่อใช้ในการควบคุมแทนการกดที่รีโมทเครื่องปรับอากาศ และพิมพ์หน้าเว็บ ดังรูปที่ 3.24



รูปที่ 3.24 ปุ่มควบคุม MODE

client.print("<INPUT type=\"submit\" name=\"status\" value=\"POWER\"></FORM>");
client.println("</html>"); break; }

จากชุดคำสั่งโปรแกรมด้านตำแหน่งหมายเลข 5 เป็นการสร้างปุ่ม Button ชื่อ POWER เพื่อใช้ในการควบคุมแทนการกดที่รีโมทเครื่องปรับอากาศ และพิมพ์หน้าเว็บ ดังรูปที่ 3.25



รูปที่ 3.25 ปุ่มควบคุม POWER

if (c == '\n') {
currentLineIsBlank = true;

```
buffer="";
 }
```

ส่วนที่ 7 การอ่าน text บน link web ถ้ากดปุ่มใดไปขึ้นข้อความบน link web ดังนั้นเราจึงต้องมา scan เพื่อที่ ดูว่าเรากดปุ่มใดไปเพื่อทำงานเมื่อกดปุ่มนั้นๆ

```
else if (c == '\r') {
if(buffer.indexOf("GET                          //ถ้า เจอ GET /สถานะ=TEMP_UP
/?status=TEMP_UP")>=0)
          {
digitalWrite(8,LOW);                           //pin8 ส่ง logic 0 แล้วหน่วงเวลา100
                                               ไมโครวินาที (MS: Microsecond) และ pin2 ส่ง
                                               logic 1
delay(100);
digitalWrite(8,HIGH);                          //ส่งสัญญาณออกจาก pin8 ไปที่บอร์ด แอร์
                                               เปรียบเหมือนกับเราไปกดปุ่มเพิ่มอุณหภูมิ 1 ครั้ง


          }
if(buffer.indexOf("GET                          //ถ้า เจอ GET /สถานะ=TEMP_DOWN
/?status=TEMP_DOWN")>=0)
           {
digitalWrite(3,LOW);                           //pin3 ส่ง logic 0 แล้ว หน่วงเวลา100
                                               ไมโครวินาทีและ pin3 ส่ง logic 1
delay(100);
digitalWrite(3,HIGH);
                                               //ส่งสัญญาณออกจาก pin3 ไปที่บอร์ด
                                               เครื่องปรับอากาศเปรียบเหมือนกับเราไปกดปุ่มลด
                                               อุณหภูมิ 1 ครั้ง
          }
if(buffer.indexOf("GET /?status=FAN")>=0)      //ถ้า เจอ GET /สถานะ=FAN
           {
```

```
digitalWrite(4,LOW);                          //pin4 ส่ง logic0แล้วหน่วงเวลา 100
delay(100);                                   ไมโครวินาที
                                              //และ pin4 ส่ง logic 1

digitalWrite(4,HIGH);                         //ส่งสัญญาณออกจาก pin4 ไปที่บอร์ดเครื่อง
                                              ปรับเปรียบเหมือนกับเราไปกดปุ่ม fan 1 ครั้ง


          }
if(buffer.indexOf("GET /?status=MODE")>=0)    //ถ้า เจอ GET /สถานะ=TEMP_MODE
 {
digitalWrite(5,LOW);                          //pin5 ส่ง logic 0 แล้วหน่วงเวลา100
                                              ไมโครวินาทีและ pin5 ส่ง logic 1
delay(100);
digitalWrite(5,HIGH);                         //ส่งสัญญาณออกจาก pin5 ไปที่บอร์ด
                                              เครื่องปรับอากาศเปรียบเหมือนกับเราไปกดปุ่ม
                                              mode 1 ครั้ง
 }
if(buffer.indexOf("GET /?status=POWER")>=0)    //ถ้าเจอ GET /สถานะ=TEMP_POWER
    {
digitalWrite(6,LOW);                          //pin6 ส่ง logic 0 แล้ว หน่วงเวลา 100
                                              ไมโครวินาทีและ pin6 ส่ง logic 1
delay(100);
digitalWrite(6,HIGH);                         //ส่งสัญญาณออกจาก pin6ไปที่แผงวงจร
                                              เครื่องปรับอากาศเปรียบเหมือนกับเราไปกดปุ่ม
                                              power1 ครั้ง


          }}
else {
currentLineIsBlank = false;
}}}
delay(1);
```

```
client.stop(); }
```

จากชุดคำสั่งด้านบน เป็นจากชุดคำสั่งโปรแกรมด้านบนเป็นการการอ่าน text บน link web ถ้ากดปุ่มใดไปขึ้นข้อความบน link web ดังนั้นเราจึงต้องมาสแกนเพื่อที่ดูว่ากดปุ่มใดไปเพื่อทำงานเมื่อกดปุ่มนั้นๆ

บทที่ 4
ผลการดำเนินงาน

การทดลองประสิทธิภาพการระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บทำการทดลองการทำงานของระบบ โดยการทดลองการทำงานของเครื่องผู้ใช้บริการเครื่อง เพื่อหาประสิทธิภาพการทำงาน ดังรูปที่ 4.1

รูปที่ 4.1 การทดลองประสิทธิภาพการระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ

4.1 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ (การกดผ่านเว็บควบคุมแบบไร้สาย)

การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บการกดผ่านเว็บควบคุมแบบไร้สาย ตรวจสอบการส่งของระบบควบคุมเครื่องปรับอากาศว่าสามารถทำงานได้มีประสิทธิภาพ ดังรูปที่ 4.2

รูปที่ 4.2 การทดลองประสิทธิภาพการทำงาน (การกดผ่านเว็บควบคุมแบบไร้สาย)

จากรูปที่ 4.2 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ(การกดผ่านเว็บควบคุมแบบไร้สาย) ดังตารางที่ 4.1

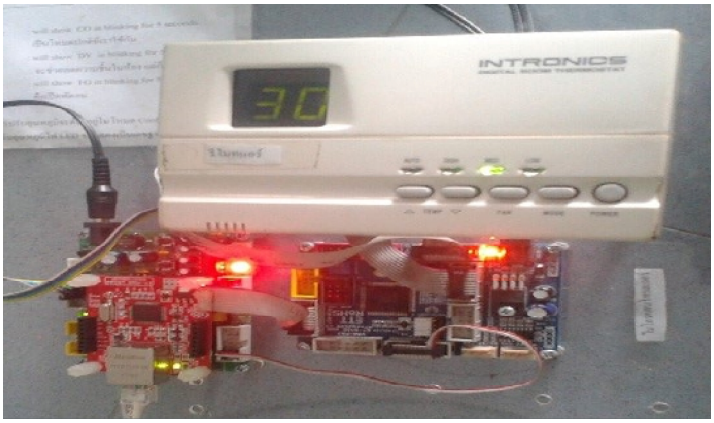
ตารางที่ 4.1 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ (การกดผ่านเว็บควบคุมแบบไร้สาย)

ปุ่มที่กด	**กดปุ่ม 1 ครั้ง**	**กดปุ่ม 2 ครั้ง (ต่อเนื่อง)**	**กดปุ่ม 3 ครั้ง (ต่อเนื่อง)**	**กดปุ่ม 4ครั้ง (ต่อเนื่อง)**
Temp up / Temp down	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1.5 วินาที/ ในการประมวลผล	ใช้เวลา 2 วินาที/ ในการประมวลผล	ใช้เวลา 2.5 วินาที/ ในการประมวลผล
Fan	ใช้เวลา 1 วินาที / ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล
Mode	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล

จากตารางที่ 4.1 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ การกดผ่านเว็บควบคุมแบบไร้สาย สามารถส่งข้อมูลได้แต่อาจช้าเพราะมีการหน่วงเวลาสามารถเชื่อมต่อและสื่อสารกันอย่างมีประสิทธิภาพ

4.2 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ แบบสายสัญญาณไขว้สาย (Cross Over Cable)

การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บแบบสายสัญญาณไขว้สายตรวจสอบการส่งของระบบควบคุมเครื่องปรับอากาศว่าสามารถทำงานได้มีประสิทธิภาพ ดังรูปที่ 4.3



รูปที่ 4.3 การทดลองประสิทธิภาพการทำงาน แบบสายสัญญาณไขว้สาย

จากรูปที่ 4.3 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ(แบบสายสัญญาณไขว้สาย) ดังตารางที่ 4.2

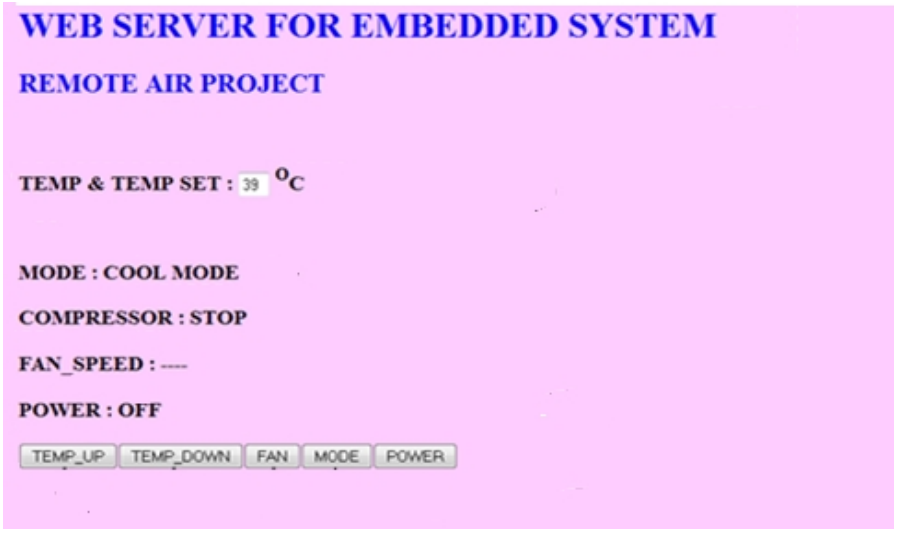
ตารางที่ 4.2 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ (แบบสายสัญญาณไขว้สาย)

ปุ่มที่กด	กดปุ่ม 1 ครั้ง	กดปุ่ม 2 ครั้ง (ต่อเนื่อง)	กดปุ่ม 3 ครั้ง (ต่อเนื่อง)	กดปุ่ม 4ครั้ง (ต่อเนื่อง)
Temp up / Temp down	ใช้เวลา 0.2 วินาที/ ในการประมวลผล	ใช้เวลา 0.3 วินาที/ ในการประมวลผล	ใช้เวลา 0.5 วินาที/ ในการประมวลผล	ใช้เวลา 0.7 วินาที/ ในการประมวลผล
Fan	ใช้เวลา 0.6 วินาที/ ในการประมวลผล	ใช้เวลา 0.5 วินาที/ ในการประมวลผล	ใช้เวลา 0.5 วินาที/ ในการประมวลผล	ใช้เวลา 0.5วินาที/ ในการประมวลผล
Mode	ใช้เวลา 0.6 วินาที/ ในการประมวลผล	ใช้เวลา 0.5วินาที/ ในการประมวลผล	ใช้เวลา 0.5 วินาที/ ในการประมวลผล	ใช้เวลา 0.5 วินาที/ ในการประมวลผล

จากตารางที่ 4.2 จากการทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ แบบสายสัญญาณไขว้สาย สามารถเชื่อมต่อและสื่อสารกันได้อย่างมีประสิทธิภาพ

4.3 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ (การกดผ่านเว็บควบคุมแบบไร้สาย)

การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ(การกดผ่านเว็บควบคุมแบบไร้สาย) ตรวจสอบการส่งของระบบควบคุมเครื่องปรับอากาศว่าสามารถทำงานได้มีประสิทธิภาพ ดังรูปที่ 4.4



รูปที่ 4.4 การทดลองประสิทธิภาพการทำงาน (การกดผ่านเว็บควบคุมแบบไร้สาย)

จากรูปที่ 4.4 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ(การกดผ่านเว็บควบคุมแบบไร้สาย) ดังตารางที่ 4.3

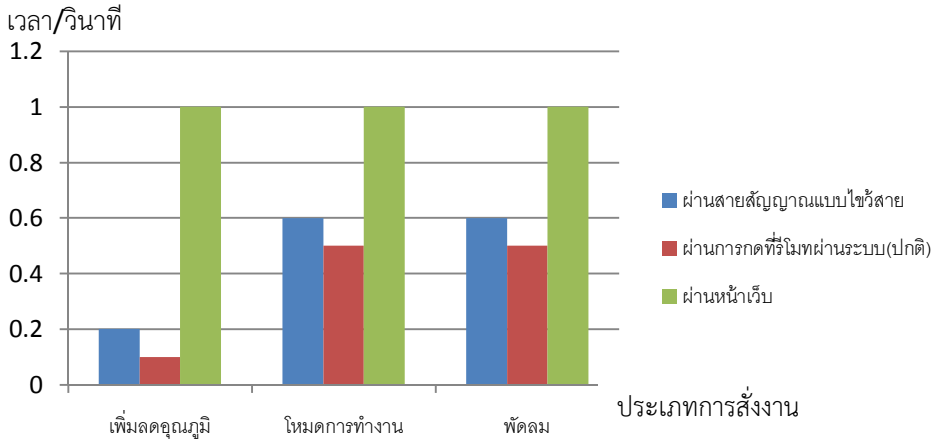
ตารางที่ 4.3 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ (การกดผ่านเว็บควบคุมแบบไร้สาย)

ปุ่มที่กด	กดปุ่ม 1 ครั้ง	กดปุ่ม 2 ครั้ง (ต่อเนื่อง)	กดปุ่ม 3 ครั้ง (ต่อเนื่อง)	กดปุ่ม 4ครั้ง (ต่อเนื่อง)
Temp up / Temp down	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1.5 วินาที/ ในการประมวลผล	ใช้เวลา 2วินาที/ ในการประมวลผล	ใช้เวลา 2.5 วินาที/ ในการประมวลผล
Fan	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1วินาที/ ในการประมวลผล	ใช้เวลา 1วินาที/ ในการประมวลผล
Mode	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1วินาที/ ในการประมวลผล	ใช้เวลา 1 วินาที/ ในการประมวลผล	ใช้เวลา 1วินาที/ ในการประมวลผล

จากตารางที่ 4.3 การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ การกดผ่านเว็บควบคุมแบบไร้สาย สามารถส่งข้อมูลได้แต่อาจช้าเพราะจะมีการหน่วงเวลาสามารถเชื่อมต่อและสื่อสารกันได้อย่างมีประสิทธิภาพ

4.4 สรุปผลการทดลอง

การทดลองประสิทธิภาพการทำงานของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ การกดผ่านรีโมทเครื่องปรับอากาศ สามารถเชื่อมต่อและสื่อสารกันได้อย่างมีประสิทธิภาพและสามารถนำไปใช้งานได้จริง แสดงดังรูปที่ 4.5



รูปที่ 4.5 แผนภูมิแท่งการทดลองประสิทธิภาพการทำงาน

สรุปผลการทดลอง จากการทดสอบหาประสิทธิภาพของระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ ได้ผลดังนี้

- การกดปุ่มแบบไร้สายสามารถใช้งานได้จริงแต่ใช้เวลาในการประมวลค่าหน่วงเวลาเป็นอาการหน่วงของเวลาเนื่องจากส่งสัญญาณแบบไร้สายต้องส่งสัญญาณตัวกลางเช่นอุปกรณ์ต่างและสภาพอากาศทำให้การส่งสัญญาณไม่เต็มที่เวลาที่ใช้ในการกดปุ่ม 1 ครั้งใช้เวลา 1 วินาที ต่อการกดหากกดต่อเนื่อง 4 ครั้ง ใช้เวลา 2.5 วินาที ในการประมวลผล

- การทดลองการกดปุ่มแบบมีสายสามารถใช้งานได้ดีเนื่องจากเป็นส่งผ่านสายโดยตรงเช่นกดปุ่ม Temp up / temp down 1 ครั้งใช้เวลาเพียง 0.5 วินาที ในการประมวลผล

- การทดลองการกดปุ่มรีโมทโดยตรงสามารถใช้งานได้ดีเนื่องจากเป็นส่งโดยตรง เช่นกดปุ่ม Temp up / temp down 1 ครั่งใช้เวลาเพียง 0.1 วินาที ในการประมวลผล

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองสรุปผลได้ว่าการพัฒนาระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บสามารถทำงานได้ตามเงื่อนไข ดังนี้

5.1.1 การควบคุมในรูปแบบไร้สายโดยมีหน้าเว็บเป็นตัวควบคุมการทำงาน เมื่อเปิดโปรแกรมต้องตั้งค่าไอพีเพื่อให้เครื่องคอมพิวเตอร์กับเครื่องควบคุมเชื่อมต่อกัน

5.1.2 ขณะทำการเชื่อมต่อระหว่างรีโมทเครื่องปรับอากาศกับหน้าเว็บควบคุมแล้วค่าที่หน้าจอกับรีโมทเครื่องปรับอากาศไม่ตรงกันให้ทำการกดปุ่มเริ่มใหม่ที่แผงวงจรควบคุมเสมอ

5.2 สรุปผลการดำเนินงาน

การพัฒนาระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ ได้มีผลจากการทำงานดังนี้

5.2.1 สามารถพัฒนาระบบควบคุมอุปกรณ์เครื่องปรับอากาศได้จริงเช่น การเพิ่มและลดอุณหภูมิของเครื่องปรับอากาศได้ ปรับโหมดของเครื่องปรับอากาศและยังเปิดปิดเครื่องปรับอากาศโดยผ่านหน้าเว็บได้

5.2.2 สามารถควบคุมเครื่องปรับอากาศผ่านหน้าคอมพิวเตอร์ของผู้ใช้โดยไม่ต้องอาศัยรีโมทเครื่องปรับอากาศ

5.3 ปัญหาและข้อจำกัด

5.3.1 ปัญหา

5.3.1.1 ในการควบคุมการระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บต้องใช้ระบบเครือข่ายท้องถิ่นเดียวกัน แต่ในกรณีที่ใช้ในอาคารเรียนมีหลายวงเครือข่ายท้องถิ่นซึ่งยากในการตั้งค่าการเชื่อมต่อ

5.3.1.2 ในส่วนของหน้าเว็บควบคุมเครื่องปรับอากาศมีการเริ่มใหม่หน้าจอตลอดเวลาเพื่อการปรับให้เป็นปัจจุบันสาถานะของรีโมทเครื่องปรับอากาศ

5.3.2 ข้อจำกัด

5.3.2.1 ไม่สามารถควบคุมการระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บในระยะที่เกินกำหนด 100 เมตร เนื่องจากสัญญาณของการปล่อยมีขีดในระยะที่จำกัด

5.3.2.2 เมื่อขยายเป็นระบบที่ใหญ่ขึ้นทำให้มีค่าใช้จ่ายสูงขึ้น

5.3.2.3 ในการควบคุมเครื่องปรับอากาศสามารถควบคุมได้เฉพาะในเครือข่ายท้องถิ่นเดียวกัน

5.4 แนวทางการนำไปใช้

การสร้างระบบควบคุมการระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ เพื่อนำเทคโนโลยีที่ทันสมัยมาประยุกต์ใช้ในชีวิตประจำวันได้จัดทำขึ้นมานั้นสามารถนำไปใช้งานได้จริงเพื่อลดปริมาณการใช้พลังงานและสิ่งไม่จำเป็นเช่นรีโมทเครื่องปรับอากาศ

5.5 ข้อเสนอแนะ

5.5.1 สามารถนำไปพัฒนาเป็นการเชื่อมต่อแบบระบบเครือข่ายไร้สายหลายๆเครื่องได้

5.5.2 สามารถนำเอารูปแบบควบคุมไปประยุกต์ใช้กับเครื่องใช้ไฟฟ้าชนิดอื่นได้

5.5.3 สามารถนำไปพัฒนาต่อโดยเปลี่ยนจากระบบใช้เทคโนโลยีสำหรับเครือข่ายท้องถิ่นเป็นใช้เครือข่ายคอมพิวเตอร์ขนาดใหญ่

บรรณานุกรม

[1] รศ. ธีรวัฒน์ ประกอบผล กันทะวงศ์ และ ศุภรักษ์ สมศรี .(2550) **.ไมโครคอนโทรเลอร์.** รายงานการวิจัย มหาวิทยาลัยเกษตรศาสตร์.

[2] โอภาส ศิริศรชิตถาวร วรพจน์ กรแก้ววัฒนกุล กฤษฎา ใจเย็น และ ชัยวัฒน์ ลิ้มพรจิตรวิไล.(2555).**พัฒนาโครงงานสร้างสรรค์โดยใช้ไมโครคอนโทรลเลอร์**

[3] **“รีโมทเครื่องปรับอากาศ รุ่น (Intronics Digital Room Thermostat”** [ออนไลน์]. เข้าถึงได้จาก : http://www.vtronix.com/products.php?c=2 (วันที่ค้นข้อมูล : 27 เมษายน 2557).

[4] **“แผงวงจร ET -BASE AVR ATMEGA 128”**ออนไลน์].เข้าถึงได้จาก: http://ett.co.th/product/03A21.html (วันที่ค้นข้อมูล : 25 มีนาคม 2557).

[5] **“แผงวงจร ET-MINI W5100”** [ออนไลน์]. เข้าถึงได้จาก : http://ett.co.th/prod2013/et-mini- w5100/et-mini-w5100.html (วันที่ค้นข้อมูล : 3 มีนาคม 2557).

[6] **“โครงสร้างการใช้งาน AppServ”** [ออนไลน์].เข้าถึงได้จาก:http://www.appservnetwork.com/ (วันที่ค้นข้อมูล : 11 เมษายน 2557).

[7] **“Adobe Dreamweaver CS5.5”** [ออนไลน์]. เข้าถึงได้จาก : https:// sites.google.com/site/ onnichacom/dreamweaver-cs5 (วันที่ค้นข้อมูล : 25 เมษายน 2557).

[8] **“Access point รุ่น ALFA”** [ออนไลน์]. เข้าถึงได้จาก : http://www.wifi4you.com/ (วันที่ค้นข้อมูล : 18 เมษายน 2557).

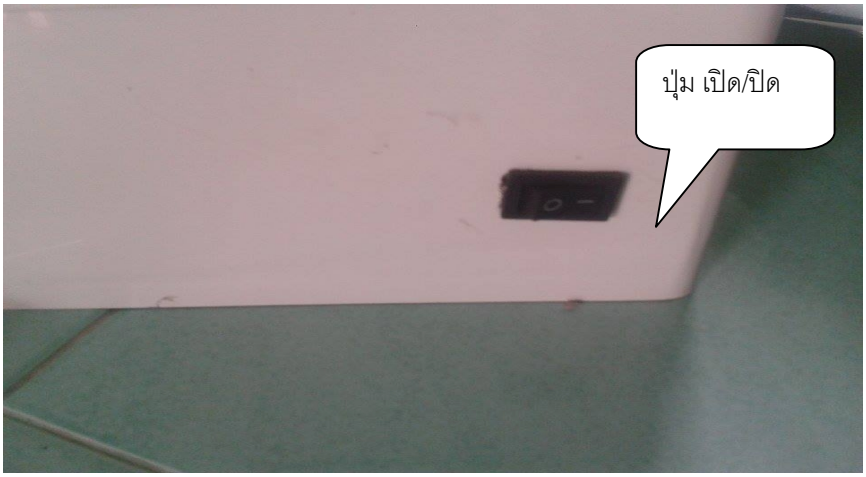
[9] **“AVR Microcontroller”**[ออนไลน์].เข้าถึงได้จาก :http://www.adisak51.com/avr2.htm (วันที่ค้นข้อมูล : 3 มิถุนายน 2557).

[10] **“AVR Microcontroller”** [ออนไลน์]. เข้าถึงได้จาก : http://ett.co.th/ (วันที่ค้นข้อมูล : 12 พฤษภาคม 2557).

ภาคผนวก ก
คู่มือการใช้งานระบบ

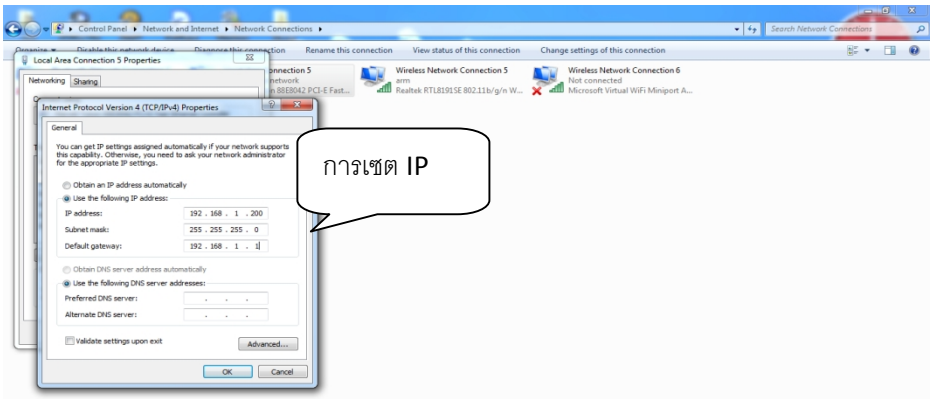
ขั้นตอนการใช้งานระบบ

1.เปิดเครื่องควบคุมเครื่องปรับอากาศ ดังรูปที่ ก.1



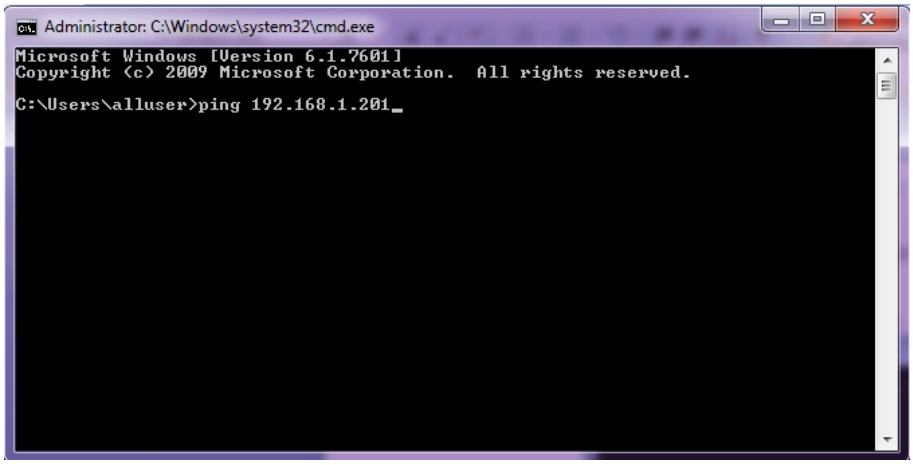
รูปที่ ก.1 เปิดเครื่องควบคุมเครื่องปรับอากาศ

2. การเซทไอพีคอมพิวเตอร์กับไอพีเครื่องควบคุมเครื่องปรับอากาศระหว่างใกล้เคียงกันเช่น 192.168.1.200 และ 192.168.1.201 ดังรูปที่ ก.2



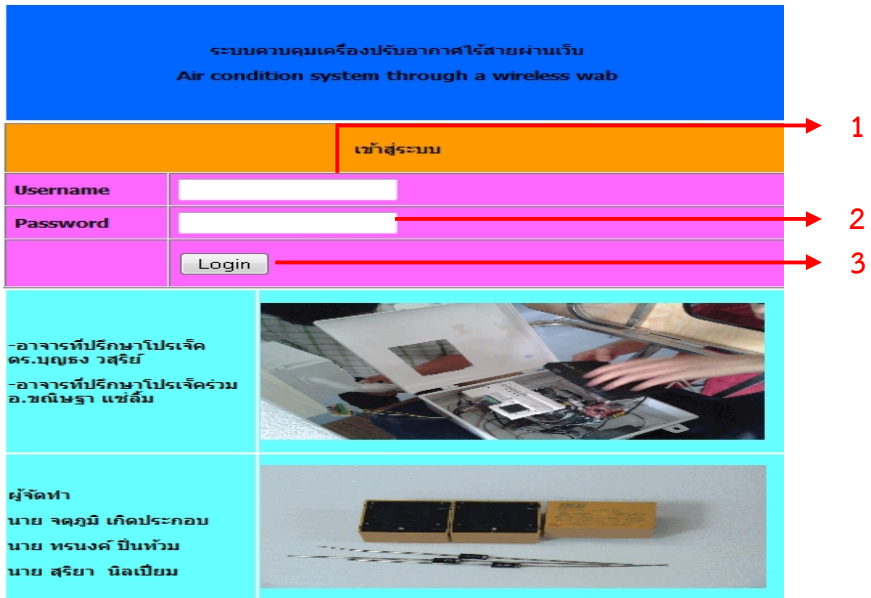
รูปที่ ก.2 เซทไอพีคอมพิวเตอร์กับไอพีเครื่องควบคุมเครื่องปรับอากาศ

3. แล้วทำการ PING ว่าคอมพิวเตอร์กับเครื่องปรับอากาศเชื่อมต่อกันหรือยัง ดังรูปที่ ก.3



รูปที่ ก.3 การ PING ว่าคอมพิวเตอร์กับเครื่องปรับอากาศเชื่อมต่อกัน

4. เข้าหน้าเว็บควบคุมเครื่องปรับอากาศ ดังรูปที่ ก.4



รูปที่ ก.4 เป็นส่วนของการเข้าหน้าเว็บควบคุมเครื่องปรับอากาศ

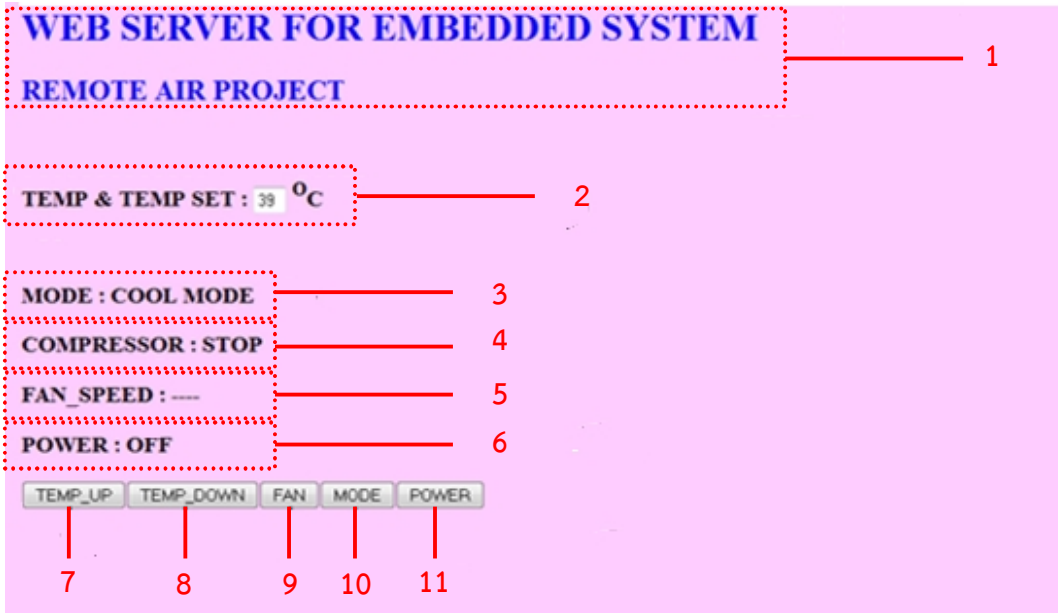
ปุ่มที่ใช้ในการทำงาน

ตำแหน่งที่ 1 ชื่อผู้ใช้ เช่น armjung , yajung

ตำแหน่งที่ 2 รหัสผ่าน 1234

ตำแหน่งที่ 3 เข้าสู่ระบบ

5.จะปรากฏหน้าเว็บคอนโทรลขึ้นมา ดังรูปที่ ก.5



รูปที่ ก.5 เป็นส่วนควบคุมการใช้งานเครื่องผู้ใช้บริการ

ตำแหน่งที่ 1 WEB SERVER FOR EMBEDDED SYSTEM REMOTE AIR PROJECT เป็นการแสดงชื่อของโปรเจ็คบนหน้าเว็บ

ตำแหน่งที่ 2 จะแสดงสถานะของโหมดในขณะปรับเลือกโหมด เช่น เย็น,พัดลม,แห้ง เป็นต้น

ตำแหน่งที่ 3 จะแสดงสถานะของอุณหภูมิเมื่อทำการปรับเพิ่มหรือลดอุณหภูมิ

ตำแหน่งที่ 4 จะแสดงสถานของการทำงานคอมเพรสเซอร์ เช่น หยุดและทำงาน

ตำแหน่งที่ 5 จะแสดงสถานของการทำงานของระดับของแรงของพัดลม เช่น HGH, MED, LOW

ตำแหน่งที่ 6 จะแสดงสถานของการทำงานเปิดปิด

ตำแหน่งที่ 7 ทำหน้าที่ในการปรับเพิ่มอุณหภูมิ

ตำแหน่งที่ 8 หน้าที่ในการปรับลดอุณหภูมิ

ตำแหน่งที่ 9 ทำหน้าปรับจาก MODE เครื่องปรับอากาศปกติให้เป็น MODE FAN

ตำแหน่งที่ 10 ทำหน้าที่ในการปรับ เช่น Cool Mode, Dry mode, Fan Mode เป็นต้น
ตำแหน่งที่ 11 ทำหน้าที่เปิดปิดเครื่องปรับอากาศ

ภาคผนวก ข
ชุดคำสั่งของการควบคุม

ชุดคำสั่งของการควบคุม

ในการศึกษาและพัฒนาชุดคำสั่งที่โปรแกรมลงไปที่แผงวงจรควบคุมต่างๆ โดยมีชุดคำสั่งดังต่อไปนี้

1. ชุดคำสั่งในการล๊อกอินหน้าเว็บควบคุมเครื่องปรับอากาศ
2. ชุดคำสั่งของวงจรควบคุมเครื่องเครื่องปรับอากาศ (ใช้ในการเก็บข้อมูลของรีโมท)
3. ชุดคำสั่งของวงจรควบคุมเครื่องเครื่องปรับอากาศ (ใช้การเชื่อมต่อระหว่างคอมพิวเตอร์กับเครื่องปรับอากาศแล้วปลิ้นหน้าเว็บควบคุม)

1. ชุดคำสั่งในการล๊อกอินหน้าเว็บควบคุมเครื่องปรับอากาศ

```
<?php include("connect.php");
if($_POST['cmdLogin']<>""){
	$user=$_POST['username'];
	$pass=$_POST['password'];
	$strQuery=mysql_query("select * from tb_admin where username='$user' and password='$pass'");
	$numrow=mysql_num_rows($strQuery);
	if($numrow<>""){
		$_SESSION['user_login']=$user;
		echo"<script>alert('สวัสดี $user');
window.location='http://192.168.1.201/';</script>";
	}else{
		echo"<script>alert('ล็อกอินไม่ถูกต้อง'); window.back();</script>";
	}
}
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

```
<title>Untitled Document</title>
<style type="text/css">
body,td,th {
        font-size: 12px;
        color: #000;
}
</style>
</head>
<body>
<form action="#" method="post">
  <p>
    <label></label>
  </p>
  <table width="490" border="0" align="center">
    <tr>
      <th width="483" height="78" bgcolor="#0066FF" scope="row"> <p> </p>
        <p><strong>ระบบควบคุมเครื่องปรับอากาศไร้สายผ่านเว็บ</strong></p>
        <p><strong> Air condition system through a wireless wab</strong></p>
        <p> </p>      </th>
    </tr>
  </table>
  <table width="490" border="1" align="center" cellpadding="5" cellspacing="0"
bordercolor="#CCCCCC" bgcolor="#EEEEEE">
    <tr>
    <td height="39" colspan="2" align="center" bgcolor="#FF9900"><strong>เข้าสู่ระบบ
</strong></td>
  </tr>
  <tr>
    <td width="88" bgcolor="#FF66FF"><strong>Username</strong></td>
    <td width="364" bgcolor="#FF66FF"><input type="text" name="username"
id="username" /></td>
```

```
  </tr>
  <tr>
    <td bgcolor="#FF66FF"><strong>Password</strong></td>
    <td bgcolor="#FF66FF">  <input type="password" name="password"
id="password" /></td>
  </tr>
  <tr>
    <td height="37" bgcolor="#FF66FF"> </td>
    <td bgcolor="#FF66FF"><input type="submit" name="cmdLogin" id="cmdLogin"
value="Login" /></td>
  </tr>
</table>
  <table width="490" height="297" border="0" align="center">
    <tr>
    <td width="153" height="146" bgcolor="#66FFFF"><p><strong>-อาจารที่ปรึกษาโปร
เจ็ค</strong><strong> ดร.บุญธง วสุริย์</strong></p>
      <p><strong>-อาจารที่ปรึกษาโปรเจ็คร่วม</strong><strong> อ.ขณิษฐา แซ่ลิ้ม
</strong><strong></strong></p></td>
    <td width="325" bgcolor="#66FFFF"><p> <img src="anigif.gif" width="313"
height="140" alt=" " /></p></td>
    </tr>
  <tr>
    <td height="145" bgcolor="#66FFFF"><p><strong>    ผู้จัดทำ</strong></p>
      <p><strong> นาย  จตุภูมิ เกิดประกอบ
      </strong></p>
      <p><strong>นาย ทรนงค์ ปิ่นท้วม  </strong></p>
      <strong> นาย สุริยา  นิลเปี่ยม  </strong></td>
    <td bgcolor="#66FFFF"><img src="1.gif" width="314" height="126" alt="ฝุ่" /></td>
    </tr>
</table>
</form>
```

```
</body>
</html>
```

2. ชุดคำสั่งของวงจรควบคุมเครื่องเครื่องปรับอากาศ (ใช้ในการเก็บข้อมูลของรีโมท)

การประกาศตัวแปร

```
#include<avr/io.h>
#include<util/delay.h>
#include<avr/interrupt.h>

#define F_CPU 16000000UL
#define USART_BAUDRATE 9600
#define BAUD_PRESCALE (((F_CPU / (USART_BAUDRATE * 16UL))) - 1)

#define EN_INT0()       EIMSK  |=  (1<<INT0)
#define DIS_INT0()  EIMSK  &=  ~(1<<INT0)
#define EN_INT0()       EIMSK  |=  (1<<INT0)
#define DIS_INT0()  EIMSK  &=  ~(1<<INT0)
#define READ_LOW_D0                    (((PIND >> PIND0 ) & 0x01) == 1)
#define EN_INT1()       EIMSK  |=  (1<<INT1)
#define DIS_INT1()  EIMSK  &=  ~(1<<INT1)
#define EN_INT1()       EIMSK  |=  (1<<INT1)
#define DIS_INT1()  EIMSK  &=  ~(1<<INT1)
volatile unsigned char temp_data_digit_front = 0xFF;
volatile unsigned char data_digit_front = 0x00;
volatile unsigned char count_data8 = 0;
volatile unsigned char temp_data_digit_black = 0xFF;
volatile unsigned char data_digit_black = 0x00;
//volatile unsigned char count_data8 = 0;
unsigned char           data_front = 0x00;
unsigned char           data_black = 0x00;
```

```
unsigned char  data_all = 0x00;
void initial_port(void);
void USART_Init(void);
void USART_Transmit( unsigned char data );
void USART_SendString(unsigned char * str);
char Data[20] = {'{',0,0,0,0,'}'};
unsigned char temp_val = 0;
unsigned char mode_val = 0;
unsigned char com_val = 0;
unsigned char fan_power_val = 0;
unsigned char flag_start = 0;
unsigned char flag_toggle = 0;
External interrupt0
ISR(INT0_vect)
{
        if(flag_start == 0)
        {
        _delay_us(100);
                if(READ_LOW_D0)
                {
                        EICRA &= ~(1<<ISC00);
                        flag_start = 1;
                        temp_data_digit_front = 0xFF;
                        temp_data_digit_black = 0xFF;
                        count_data8 = 0;
                }
        }
        Else
        {
                temp_data_digit_front &= PINC;
                count_data8++;
```

```
                if(count_data8 == 8)
{
                        DIS_INT0();
                        EN_INT1();
                        data_digit_front = temp_data_digit_front  & 0b11001111;
                }
        }
}
External interrupt1
ISR(INT1_vect)
{
                temp_data_digit_black &= PINC;
                count_data8++;
                        if(count_data8 == 16)
                {
                        DIS_INT1();
                        data_digit_black = temp_data_digit_black  & 0b11010111;
                        flag_start = 0;
                        EICRA |= (1<<ISC00);
                        EN_INT0();
                }
                }

int main (void)
{
        USART_Init();
        PORTA = 0xFF;
        DDRA = 0x00;
        PORTC = 0xFF;
        DDRC = 0x00;
```

```
PORTD = 0xFF;
DDRD = 0xF0;
EICRA = 0x00;
EICRA |= (1<<ISC01) | (1<<ISC00);
EICRA |= (1<<ISC11);
EN_INT0();
sei();
while(1)
{
        fan_power_val = PINA;
        fan_power_val &= 0x07;
        if(fan_power_val == 0x06)
        {
                fan_power_val = 2;
        }
        else if(fan_power_val == 0x05)

        {
                fan_power_val = 1;
        }
        else if(fan_power_val == 0x03)
        {
                fan_power_val = 0;
        }
        else if(fan_power_val == 0x07)
        {
                fan_power_val = 3;
        }
        Data[4] = fan_power_val;
        com_val = PINA;
        com_val &= 0x08;
```

```
                if(com_val == 0x00)
                {
                        com_val = 1;
                }
                Else
                {
                        com_val = 0;
                }
                Data[3] = com_val;
USART_Transmit('{');
                _delay_ms(20);
                for(int i = 1 ; i < 5 ; i++)
                {
                        USART_Transmit((48+Data[i]));
                        _delay_ms(20);
                }
                USART_Transmit('}');
                _delay_ms(20);
                switch
                {
                        case(0x04): data_front = 0x00;
                                                break;
                        case(0xCE): data_front = 0x01;
                                                break;
                        case(0x08): data_front = 0x02;
                                                break;
                        case(0x88): data_front = 0x03;
                                                break;
                        case(0xC2): data_front = 0x04;
                                                break;
                        case(0x81): data_front = 0x05;
```

```
                                    break;
            case(0x01): data_front = 0x06;
                                    break;
            case(0xCC): data_front = 0x07;
                                    break;
            case(0x00): data_front = 0x08;
                                    break;
case(0x80): data_front = 0x09;
                                    break;
case(0x41): Data[2]  = 0;
                                    break;
case(0x0A): Data[2] = 1;
                                    break;
case(0x05): Data[2] = 2;
                                    break;
}
switch(data_digit_black)
{
            case(0x40): data_black = 0x00;
                                    break;
            case(0x53): data_black = 0x01
                                    break;
            case(0x81): data_black = 0x02;
                                    break;
            case(0x11): data_black = 0x03;
                                    break;
            case(0x12): data_black = 0x04;
                                    break;
            case(0x14): data_black = 0x05;
                                    break;
            case(0x04): data_black = 0x06;
```

```
                                                    break;
                                case(0x51): data_black = 0x07;
                                                    break;
                                            case(0x00): data_black = 0x08;
                                                    break;
                                case(0x10): data_black = 0x09;
                                                    break;
                        }
                        Data[1] =  (data_front*10) + data_black;
                }
        }
void USART_Init(void)
{
   UBRR0L = BAUD_PRESCALE;
   UBRR0H = (BAUD_PRESCALE >> 8);

  UCSR0B = (1<<TXEN0);
  UCSR0C = (3<<UCSZ00)
}
void USART_Transmit( unsigned char data )
{
while ( !( UCSR0A & (1<<UDRE0)) );
UDR0 = data;
}
```

3. ชุดคำสั่งของวงจรควบคุมเครื่องเครื่องปรับอากาศ (ใช้การเชื่อมต่อระหว่างคอมพิวเตอร์กับเครื่องปรับอากาศแล้วปลิ้นหน้าเว็บควบคุม)

```
#include <SPI.h>
#include <Ethernet.h>
byte mac[] = {
  0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
```

```
#define refresh_web()  client.print("<meta http-equiv=\"refresh\"
content=\"1;url=http://192.168.1.201/\"/>")
IPAddress ip(192,168,1,201);
EthernetServer server(80);
unsigned char temp_val = 0;
unsigned char mode_val = 0;
unsigned char com_val = 0;
unsigned char fan_power_val = 0;
unsigned char data_receiver[11] = {0};
unsigned char temp_receiver = 0;
unsigned char flag_receiver_start = 0;
int incomingByte = 0;   // for incoming serial data
int count_addrUart = 0;
int flag_send_web = 0;

void setup()
{
        Serial.begin(9600);

  pinMode(8, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);

  Ethernet.begin(mac, ip);
  server.begin();

  digitalWrite(8,HIGH);
  digitalWrite(3,HIGH);
  digitalWrite(4,HIGH);
```

```
  digitalWrite(5,HIGH);
  digitalWrite(6,HIGH);
}
void loop()
{
   if (Serial.available() > 0)
 {
                temp_receiver = Serial.read();
                 if(flag_receiver_start == 0)
                 {
                    if(temp_receiver == '{')
                    {
                        flag_receiver_start = 1;

                    }

                 }
                 else
                 {
                    if(temp_receiver == '}')
                    {
                        temp_val = data_receiver[0];
                        mode_val = data_receiver[1];
                        com_val = data_receiver[2];
                        fan_power_val = data_receiver[3];
                        count_addrUart = 0;
                        flag_receiver_start = 0;
                    }
                    Else
                    {
                        data_receiver[count_addrUart] = (temp_receiver - 48);
                        count_addrUart++;
```

```
                    }
                    if(count_addrUart > 5)
                    {
                      count_addrUart = 0;
                        flag_receiver_start = 0;
                    }

                }
        }
  EthernetClient client = server.available();
  if (client) {
    boolean currentLineIsBlank = true;
    String buffer = "";
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.print(c);
        buffer+=c;
        if (c == '\n' && currentLineIsBlank) {
          client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println();
          client.println("<html>");
          client.println("<h1><font color=\"blue\">WEB SERVER FOR EMBEDDED
SYSTEM</font></h1>");
          client.println("<h2><font color=\"blue\">REMOTE AIR
PROJECT</font></h2>");
          client.print("<FORM action=\"\">");
          client.println("<body bgcolor =\"#FFCCFF\"></body>");

          sprintf((char *)data,"%u",temp_val);
```

```
            client.print("<strong>TEMP & TEMP SET : <input type=\"text\" size=2
value=\"");
            client.print(data);
            client.print("\"> <sup>O</sup>C\r\n<br>");
  switch(mode_val)
            {
                case(0):  client.println("<br><h3><font color=\"black\">MODE : FAN
MODE</font></h3>");
                        break;
                case(1):  client.println("<br><h3><font color=\"black\">MODE : DRY
MODE</font></h3>");
                        break;
               case(2):  client.println("<br><h3><font color=\"black\">MODE : COOL
MODE</font></h3>");
                        break;
                default: break;

            }
            switch(com_val)
            {
                case(0):  client.println("<h3><font color=\"black\">COMPRESSOR  :
STOP</font></h3>");
                        break;
                case(1):  client.println("<h3><font color=\"black\">COMPRESSOR :
RUN</font></h3>");
                        break;
                default: break;
            }
switch(fan_power_val)
            {
```

```
            case(0):  client.println("<h3><font color=\"black\">FAN_SPEED :
LOW</font></h3>");
                      client.println("<h3><font color=\"black\">POWER :
ON</font></h3>");
                    break;
            case(1):  client.println("<h3><font color=\"black\">FAN_SPEED :
MID</font></h3>");
                      client.println("<h3><font color=\"black\">POWER :
ON</font></h3>");
                    break;
             case(2):  client.println("<h3><font color=\"black\">FAN_SPEED : HIGH
</font></h3>");
                       client.println("<h3><font color=\"black\">POWER :
ON</font></h3>");
                    break;

                     case(3):  client.println("<h3><font color=\"black\">FAN_SPEED : ----
</font></h3>");

                      client.println("<h3><font color=\"black\">POWER  :
OFF</font></h3>");
                    break;
            default: break;
          }
        client.print("<P><INPUT type=\"submit\"  name=\"status\"
value=\"TEMP_UP\">");
        client.print("<INPUT type=\"submit\"  name=\"status\"
value=\"TEMP_DOWN\">");
        client.print("<INPUT type=\"submit\"  name=\"status\"  value=\"FAN\">");
        client.print("<INPUT type=\"submit\"  name=\"status\"  value=\"MODE\">");
        client.print("<INPUT type=\"submit\"  name=\"status\"  value=\"POWER\">
</FORM>");
```

```
  client.println("</html>");
  break;
}
if (c == '\n') {
  currentLineIsBlank = true;
  buffer="";
}
else if (c == '\r') {
  if(buffer.indexOf("GET /?status=TEMP_UP")>=0)
    {
      digitalWrite(8,LOW);
      delay(100);
      digitalWrite(8,HIGH);
            }
  if(buffer.indexOf("GET /?status=TEMP_DOWN")>=0)
     {
      digitalWrite(3,LOW);
      delay(100);
      digitalWrite(3,HIGH);
     }
   if(buffer.indexOf("GET /?status=FAN")>=0)
     {
      digitalWrite(4,LOW);
      delay(100);
      digitalWrite(4,HIGH);
     }
  if(buffer.indexOf("GET /?status=MODE")>=0)
     {
      digitalWrite(5,LOW);
      delay(100);
      digitalWrite(5,HIGH);
```

```
            }
                    if(buffer.indexOf("GET /?status=POWER")>=0)
             {
              digitalWrite(6,LOW);
              delay(100);
              digitalWrite(6,HIGH);
            }
        }
        else {
          currentLineIsBlank = false;
        }
      }
    }
    delay(1);
    client.stop();
  }
}
}
```

ประวัติผู้จัดทำปริญญานิพนธ์

ชื่อ – สกุล	นายสุริยา นิลเปี่ยม
รหัสประจำตัวนักศึกษา	524282150
สาขาวิชา	เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
โปรแกรมวิชา	เทคโนโลยีอุตสาหกรรม
สถานศึกษา	มหาวิทยาลัยราชภัฏนครปฐม
ที่อยู่	9809 หมู่ 9 ตำบลดอนคา อำเภออู่ทอง จังหวัดสุพรรณบุรี 72160
อีเมล์ :	ya_jung789@hotmail.com
ประวัติการศึกษา	
- เบอร์โทร	090-791-7494
- ระดับมัธยมต้น	โรงเรียนอู่ทองศึกษาลัย
- ระดับมัธยมปลาย	โรงเรียนอู่ทองศึกษาลัย
- ระดับปริญญาตรี	มหาวิทยาลัยราชภัฏนครปฐม

ประวัติผู้จัดทำปริญญานิพนธ์

ชื่อ – สกุล	นายจตุภูมิ เกิดประกอบ
รหัสประจำตัวนักศึกษา	524282106
สาขาวิชา	เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม
สถานศึกษา	มหาวิทยาลัยราชภัฏนครปฐม
ที่อยู่	1940/18 หมู่ 6 ตำบลอู่ทอง อำเภออู่ทอง จังหวัดสุพรรณบุรี 72160
อีเมล์ :	arm_jk34@hotmail.com
ประวัติการศึกษา	
- เบอร์โทร	082-075-0765
- ระดับมัธยมต้น	โรงเรียนอู่ทอง
- ระดับมัธยมปลาย	โรงเรียนอู่ทอง
- ระดับปริญญาตรี	มหาวิทยาลัยราชภัฏนครปฐม