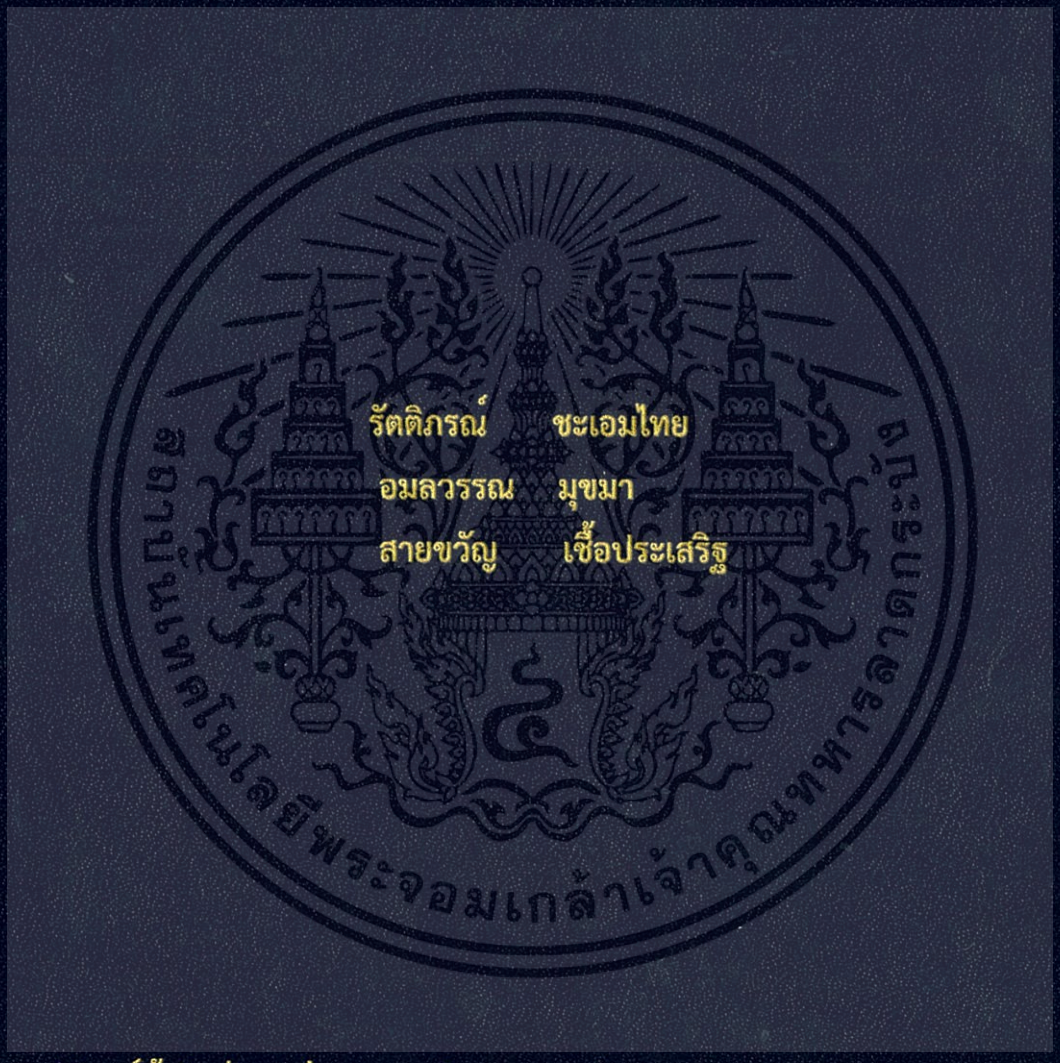
อุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนด

THE AIR CONDITIONER CONTROLLED DEVICE FOLLOWS THE SETTING TIME



รัตติภรณ์ ชะเอมไทย
อมลวรรณ มุขมา
สายขวัญ เชื้อประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

อุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนด

THE AIR CONDITIONER CONTROLLED DEVICE FOLLOWS THE SETTING TIME

รัตติภรณ์ ชะเอมไทย

อมลวรรณ มุขมา

สายขวัญ เชื้อประเสริฐ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE AIR CONDITIONER CONTROLLED DEVICE FOLLOWS THE SETTING TIME

Rattiporn Cha-emthai
Amonwan Mukma
Saikwan Chueprasert

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2014

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2557

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์	อุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนด THE AIR CONDITIONER CONTROLLED DEVICE FOLLOWS THE SETTING TIME	
นักศึกษาผู้จัดทำ	นางสาวรัตติภรณ์ ชะเอมไทย	**รหัสนักศึกษา** 54011081
	นางสาวอมลวรรณ มุขมา	**รหัสนักศึกษา** 54011507
	นางสาวสายขวัญ เชื้อประเสริฐ	**รหัสนักศึกษา** 54011355
ปริญญา	วิศวกรรมศาสตรบัณฑิต	
สาขาวิชา	วิศวกรรมการวัดคุม	
ปีการศึกษา	2557	

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
ผศ. เชื้อ นกอยู่	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	อุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนด THE AIR CONDITIONER CONTROLLED DEVICE FOLLOWS THE SETTING TIME
นักศึกษาผู้จัดทำ	นางสาวรัตติภรณ์ ชะเอมไทย **รหัสนักศึกษา** 54011081 นางสาวอมลวรรณ มุขมา **รหัสนักศึกษา** 54011507 นางสาวสายขวัญ เชื้อประเสริฐ **รหัสนักศึกษา** 54011355
อาจารย์ที่ปรึกษา	ผศ. เชื้อ นกอยู่
ปริญญา	วิศวกรรมศาสตรบัณฑิต
สาขาวิชา	วิศวกรรมการวัดคุม
ปีการศึกษา	2557

บทคัดย่อ

ในปัจจุบันเทคโนโลยีมีความก้าวหน้าไปอย่างรวดเร็ว เพื่ออำนวยความสะดวกสบายต่างๆ ให้กับมนุษย์ เครื่องปรับอากาศเป็นเครื่องใช้ไฟฟ้าชนิดหนึ่งที่อำนวยความสะดวกสบายให้กับมนุษย์อีกเช่นกัน เพราะประเทศไทยตั้งอยู่ในภูมิประเทศร้อนชื้น มนุษย์จึงต้องเปิดเครื่องปรับอากาศเพื่อทำให้อากาศเย็นสบาย และปิดเมื่อไม่มีความจำเป็นต้องใช้ ดังนั้นคณะผู้จัดทำจึงได้คิดค้นอุปกรณ์เสริมที่ช่วยควบคุมการเปิด-ปิดเครื่องปรับอากาศเพื่อเพิ่มความสะดวกสบายในการใช้งานให้กับผู้ใช้ โดยใช้บอร์ดไมโครคอนโทรเลอร์ตระกูล AVR หรือ ที่รู้จักในชื่อ Arduino ถูกใช้เป็นตัวควบคุม (Controller) ทำงานตามโปรแกรมที่เขียนไว้ ให้สามารถเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนดไว้ได้ผ่านทางระบบอินเตอร์เน็ต ภายในเครือข่าย LAN เดียวกัน โดยผู้ใช้สามารถตั้งเวลาเปิด –ปิดเครื่องปรับอากาศที่หน้าเว็บบราวเซอร์ และมีการสั่งการเปิด-ปิดเครื่องปรับอากาศผ่านหน้าจอทัชสกรีนได้อีกช่องทางหนึ่ง ซึ่งอุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนดนี้ สามารถนำไปใช้ในชีวิตประจำวันได้จริงอย่างมีประสิทธิภาพ เพื่ออำนวยความสะดวกสบายให้กับผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	THE AIR CONDITIONER CONTROLLED DEVICE FOLLOWS THE SETTING TIME
Authors	Ms. Rattiporn Cha-emthai
	Ms. Amonwan Mukma
	Ms. Saikwan Chueprasert
Thesis Advisor	Asst. Prof. Chuae Nokyoo
Year	2014

ABSTRACT

Nowadays, technology is advancing rapidly. To facilitate comfortable for human, air conditioner is kind of Electric appliance that human use for convenience. Because Thailand is situated in tropical landscapes. People need to turn on air conditioner to keep cool and turn off when not needed. Therefore, the device was invented accessories that help control the on - off air conditioner to maximize convenience of use to the users. The board's AVR family of microcontrollers, also known as Arduino was used as a control (Controller) which works follow the program is written to turn - off the air conditioner according to the setting time via the Internet within the same LAN network. User can set time to turn on-off air conditioner in the web. User also can turn on-off air conditioner via touch screen display. The air conditioner controlled device follows the setting time can be used in everyday life and have efficiency. To facilitate comfort to the user.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้จัดทำสำเร็จลุล่วงได้ด้วยความเมตตากรุณาอย่างสูงจากผู้ช่วยศาสตราจารย์ เชื้อ นกอยู่ อาจารย์ที่ปรึกษาปริญญานิพนธ์ แลรศ.ทรงชัย วีระทวีมาศที่ได้ให้คำแนะนำมาโดยตลอดอีกทั้งเอื้อเฟื้ออุปกรณ์และเครื่องมือต่างๆในการทำปริญญานิพนธ์นี้ ผู้วิจัยรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง ณ ที่นี้

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกท่านที่ให้คำแนะนำและเป็นกำลังใจในการทำวิจัยตลอดมา

ขอขอบพระคุณภาควิชาวิศวกรรมการวัดคุมคณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่เอื้อเฟื้อสถานที่ในการทำวิจัย

สุดท้ายนี้ขอกราบขอบพระคุณคุณพ่อคุณแม่ ที่ให้การสนับสนุน คอยให้กำลังใจ และเป็นแรงบัลดาลใจในการทำปริญญานิพนธ์เสมอมา คุณความดีที่พึงมีจากการทำปริญญานิพนธ์ ผู้วิจัยขอมอบแด่ คุณพ่อ คุณแม่ รวมทั้งคณาจารย์และผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง IX

สารบัญรูป X

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 ความมุ่งหมายและวัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 ขั้นตอนการศึกษา 2

บทที่ 2 ทฤษฎีและโครงงานที่เกี่ยวข้อง 3

2.1 ความรู้เบื้องต้นเกี่ยวกับ Arduino MEGA 2560 R3 board 3

2.1.1 โครงสร้างพื้นฐานของ Arduino MEGA 2560 R3 board 4

2.1.1.1 หน้าที่และประโยชน์การใช้งานของพอร์ทต่างๆ 4

2.1.2 คุณสมบัติเบื้องต้นของ Arduino MEGA 2560 R3 board 6

2.1.3 Arduino Software 6

2.2 ขั้นตอนการเริ่มต้นใช้งาน Arduino MEGA 2560 R3 board 7

2.3 วิธีการติดตั้ง Arduino software 1.0.6 10

2.4 ตัวอย่างฟังก์ชันการทำงานต่างๆของ Arduino MEGA 2560 R3 ที่เกี่ยวข้องกับโครงงาน 12

2.4.1 Temperature Sensor โมดูลเพื่อวัดค่าจากสิ่งแวดล้อม 12

2.4.1.1 วิธีการต่อใช้งาน DHT11 กับบอร์ด Ardiono 13

2.4.1.2 วิธีการส่งข้อมูล DHT11 กับบอร์ด Arduino 13

2.4.2 โมดูล Real Time Clock (RTC) 14

2.4.2.1 คุณสมบัติ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.4.2.2 วิธีต่อใช้งาน 15
2.4.3 Ethernet Shield W5100 R3 16
2.4.4 Touch screen (3.2 inch TFT LCD module Display with touch panel SD Card) 17
2.4.4.1 การใช้งานหน้าจอทัชกรีนนี้กับบอร์ด Arduino 17
2.4.4.2 ข้อดีของการใช้ TFT LCD Mega Shield V2.2 Adapter Module 17
2.4.4.3 การใช้งาน 17
2.4.5 รีเลย์ (8 Channel Relay) 20
2.5 หลักการทำงานของ Arduino Mega 2560 board ที่ถูกควบคุมผ่านระบบอินเตอร์เน็ต 21
2.5.1 การทำเว็บเซิฟเวอร์ด้วยบอร์ด Arduino Mega 2560 และ Ethernet W5100 21
2.5.1.1 เพิ่มไลบรารี่ในโปรแกรม Adruino IDE 1.0.1 22
2.5.1.2 เปิดตัวอย่างเว็บเซิฟเวอร์ 22
2.5.1.3 ผลของโปรแกรม 23
2.6 ความรู้เบื้องต้นเกี่ยวกับเครื่องปรับอากาศ 24
2.6.1 หลักการทำความเย็นของเครื่องปรับอากาศ 24
2.6.2 อุปกรณ์ระบบทำความเย็นแบบคอมเพรสเซอร์อัดไอ 25
2.6.2.1 คอมเพรสเซอร์ (Compressor) 25
2.6.2.2 คอยล์ร้อน (Condenser) 25
2.6.2.3 คอยล์เย็น (Evaporator) 25
2.6.2.4 อุปกรณ์ลดความดัน (Throttling Device) 27
2.6.3 BTU (British Thermal Unit) 27
2.6.3.1 เหตุผลที่ต้องเลือก BTU ให้พอเหมาะ 28
2.6.3.2 การคำนวณ BTU 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.6.3.3 ตัวแปรความร้อน 29
2.6.4 ระบบ Inverter 29
2.7 การประยุกต์ใช้งานเครื่องปรับอากาศ 30
2.7.1 ภาระในระบบปรับอากาศ 30
2.7.2 การเลือกระบบปรับอากาศ 31
2.7.2.1 ระบบปรับอากาศแบบแยกส่วน (Split Type) 31
2.7.2.2 ระบบปรับอากาศแบบชุดหรือแพ็คเกจ (Package) 31
2.7.2.3 ระบบปรับอากาศแบบใช้เครื่องทำน้ำเย็น (Chiller) 31
2.7.3 ประสิทธิภาพระบบปรับอากาศ 33
2.7.3.1 อัตราส่วนประสิทธิภาพพลังงาน (EnergyEfficientRatio,EER) 33
2.7.3.2 ประสิทธิภาพของเครื่องทำน้ำเย็น (ChillerPerformance,ChP) 33
2.7.4 การควบคุมระบบปรับอากาศอย่างมีประสิทธิภาพ 33
2.7.5 การควบคุมระบบทำความเย็นที่เครื่องทำน้ำเย็น 34
2.7.6 การควบคุมความเร็วรอบของเครื่องสูบน้ำเย็น 35
2.7.7 การควบคุมระบบส่งจ่ายลมเย็น 37
2.7.8 การควบคุมหอระบายความร้อน 38
2.7.9 แนวทางในการประหยัดพลังงานในระบบปรับอากาศ 38
2.7.9.1 การออกแบบปรับปรุงอาคาร 38
2.7.9.2 การเลือกใช้อุปกรณ์ 38
2.7.9.3 การใช้งานและการควบคุม 39

บทที่ 3 หลักการออกแบบ 42
3.1 โครงสร้างของระบบควบคุมด้วย Arduino 42
3.1.1 Arduino 43
3.1.2 หน้าที่ของบอร์ด Arduino 44
3.1.3 การเชื่อมต่อบอร์ด Arduino กับอุปกรณ์ 45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

3.2 อุปกรณ์ที่ใช้ในการควบคุมการเปิด-ปิดเครื่องปรับอากาศ 48
3.2.1 Temperature Sensor (DHT11) 48
3.2.2 Real Time Clock (DS1307) 49
3.2.3 Ethernet Shield (W5100 R3) 50
3.2.4 Touch screen (3.2 inch TFT LCD module Display with touch panel SD Card) 51
3.2.5 Relay (8 Channel Relay) 53
3.3 โครงสร้างของระบบควบคุมเครื่องปรับอากาศ 56
3.3.1 ระบบควบคุมเครื่องปรับอากาศ 56
3.3.1.1 การทำงานของเครื่องปรับอากาศ 56
3.3.1.1.1 ระบบควบคุมแบบอัตโนมัติ 56
3.3.1.1.2 ระบบควบคุมด้วยมือ 56
3.3.1.2 การควบคุมการเปิด-ปิดเครื่องปรับอากาศ 56
3.3.1.2.1 ควบคุมการเปิด – ปิดจากการตั้งเวลา 57
3.3.1.2.2 ควบคุมการเปิดปิดผ่านระบบอินเตอร์เน็ต 57
3.3.1.2.3 ควบคุมการเปิดปิดผ่านหน้าจอทัชสกรีน 59
3.3.2 ระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด 62
3.3.2.1 ควบคุมการเปิด-ปิดเครื่องปรับอากาศโดยการตั้งเวลาผ่านระบบอินเตอร์เน็ต 63
3.3.2.2 ควบคุมการเปิด-ปิดเครื่องปรับอากาศผ่านหน้าจอทัชสกรีน 65

บทที่ 4 การทดลอง 74
4.1 กล่าวนำ 74
4.2 การทดลองการเปิด-ปิดเครื่องปรับอากาศแบบภาระโหลด 74
4.2.1 การทดลองเปิด-ปิดเครื่องปรับอากาศโดยการตั้งเวลาผ่านระบบอินเตอร์เน็ต 74
4.2.1.1 ขั้นตอนการทดลอง 74
4.2.1.2 ผลการทดลอง 77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2.1.2.1 ผลการทดลองของการกดปุ่มคำสั่ง OPEN และ CLOSE 77

4.2.1.2.2 ผลการทดลองของการตั้งค่าเวลาเปิด-ปิดเครื่องปรับอากาศ 78

4.2.2 การทดลองเปิด-ปิดเครื่องปรับอากาศโดยผ่านหน้าจอทัชสกรีน 78

4.2.2.1 ขั้นตอนการทดลอง 78

4.2.2.2 ผลการทดลองของการกดปุ่มคำสั่ง OPEN และ SHUT 80

บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ 81

5.1 สรุปผลการทดลอง 81

5.2 ข้อเสนอแนะ 81

บรรณานุกรม 82

ภาคผนวก 84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตารางแสดงขนาด BTU ที่เหมาะสมกับขนาดของห้อง....28

2.2 ตารางแสดงค่าไฟฟ้าจากการติดตั้งเครื่องปรับอากาศต่างสถานที่กัน....29

2.3 ตารางแสดงการสรุปการใช้งานของเครื่องปรับอากาศในลักษณะต่างๆ....32

3.1 ตารางแสดงการใช้ช่อง Input ของ Arduino....45

3.2 ตารางแสดงการใช้ช่อง Output ของ Arduino....45

3.3 ตารางแสดงรายละเอียดในการต่ออุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศ....46
เข้ากับบอร์ด Arduino (MEGA 2560)

3.4 ตารางแสดงการทำงานของพัดลมตามเงื่อนไขอุณหภูมิ....48

4.1 ตารางแสดงการทดลองกดปุ่มเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัวผ่านเว็บบราวเซอร์....77

4.2 ตารางแสดงการทดลองตั้งค่าเวลาเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัว....78

4.3 ตารางแสดงการทดลองกดปุ่มเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัวผ่านหน้าจอทัชสกรีน....80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างพื้นฐานของ Arduino MEGA 2560 R3 board....4
2.2 Pinout Diagram ของ Arduino MEGA 2560 R3 board....5
2.3 Pinout Diagram ของ MCU (Microcontroller) Atmega2560....5
2.4 ตัวอย่างต่อบอร์ด Arduino MEGA 2560 R3 เข้ากับคอมพิวเตอร์ผ่านทาง USB PORT....7
2.5 หน้าต่างการหาไดร์เวอร์ของ windows....8
2.6 การแก้ปัญหาเมื่อ windows หาไดร์เวอร์ไม่เจอ....8
2.7 ตัวอย่างหน้าต่างการเลือกเมนู browse my computer....9
2.8 การ browse ไปที่ไดร์เวอร์ Arduino....9
2.9 การ Install this driver software anyway....10
2.10 หน้าหลักของ เว็บไซต์ http://arduino.cc....10
2.11 หน้าต่างที่ได้หลังจากกดเมนู Download....11
2.12 หน้าดาวน์โหลด ARDUINO 1.0.5....11
2.13 หน้าต่างแจ้งข้อมูลเรื่องลิขสิทธ์ก่อนทำการติดตั้ง....12
2.14 วิธีการต่อใช้งาน DHT11 กับบอร์ด Arduino....13
2.15 Timing diagram แสดงการ ส่งข้อมูล DHT11 กับ MCU....14
2.16 ตัวอย่าง Tiny RTC I2C 24C32 DS1307....15
2.17 ตัวอย่าง Ethernet Shield W5100 R3....16
2.18 ตัวอย่างหน้าจอทัชสกรีน....17
2.19 การระบุตำแหน่งและโครงสร้างเบื้องต้นของ Resistive Touch Screen....18
2.20 การหาตำแหน่งระยะจากขอบจอจากสมการ....19
2.21 หน้าสัมผัสแต่ละชั้นของ Resistive Touch Screen....19
2.22 การถ่ายประจุจากจอไปที่มือของผู้ใช้....20
2.23 8-Channel Relay....20
2.24 circuit diagram ของ Relay....21
2.25 การต่ออุปกรณ์ระหว่าง Arduino Board และ Ethernet Shield....22
2.26 การเปิดตัวอย่างเว็บเซิฟเวอร์....22
2.27 ตัวอย่าง Sketch ในการเปิดเว็บเซิฟเวอร์....23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ หน้า

2.28 ผลของโปรแกรม....23
2.29 ตัวอย่าง Refrigeration Cycle....24
2.30 Semi flood evaporator....26
2.31 Full- flooded evaporator....26
2.32 Liquid overfeed....27
2.33 ภาระในระบบปรับอากาศ....30
2.34 ไดอะแกรมระบบปรับอากาศแบบใช้เครื่องทำน้ำเย็น....34
2.35 ไดอะแกรมระบบปฐมภูมิ....35
2.36 ไดอะแกรมระบบปฐมภูมิ-ทุติยภูมิ....35
2.37 ไดอะแกรมระบบปฐมภูมิ-ทุติยภูมิที่มีการติดตั้ง VSD....36
2.38 ภาพรวมระบบส่งจ่ายลม....37
2.39 ระบบส่งจ่ายลมเย็นและการควบคุม....37
3.1 ภาพรวมการติดตั้งอุปกรณ์ของระบบควบคุมการเปิด-ปิดเครื่องปรับอากาศ....42
ตารางเวลาที่กำหนดอย่างอัตโนมัติ
3.2 การเลือกรุ่นของบอร์ด Arduino....43
3.3 ตัวอย่างการเขียนโปรแกรมในโปรแกรม Arduino....44
3.4 ตัวอย่างการเขียนโปรแกรมในโปรแกรม Arduino (ต่อ)....44
3.5 อุปกรณ์ DHT11 (Humidity and Temperature Sensor)....48
3.6 การต่อ DHT11 กับ Arduino Mega 2560 R3 Board....49
3.7 Real Time Clock (DS1307)....49
3.8 การต่อ RTC กับ Arduino Mega 2560 R3 Board....50
3.9 Ethernet Shield W5100 R3....50
3.10 การต่อ Ethernet Shield กับ Arduino Mega 2560 R3 board....51
3.11 การต่อ Ethernet Shield กับ Arduino Mega 2560 R3 board (ต่อ)....51
3.12 3.2 inch TFT LCD module Display with touch panel SD Card....52
3.13 แสดงการต่อ Touch screen กับ Arduino Mega 2560 R3 board....52
3.14 TFT LCD Mega Shield V2.2 Adapter Module....52
3.15 การต่อรีเลย์กับ Arduino Mega 2560 R3 board....53

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.16	กล่องอุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศ	54
3.17	อุปกรณ์ฮาร์ดแวร์ภายในกล่อง	54
3.18	ภาพด้านหน้าของกล่อง	54
3.19	ภาพด้านล่างของกล่อง	55
3.20	ภาพด้านบนของกล่อง	55
3.21	ภาพด้านซ้ายของกล่อง	55
3.22	ตั้งค่าเวลาเปิดปิดเครื่องปรับอากาศผ่านหน้าเว็บบราวเซอร์	57
3.23	หน้าเว็บบราวเซอร์	58
3.24	หน้าเว็บบราวเซอร์ (ต่อ)	58
3.25	หน้าจอทัชสกรีนในระบบควบคุมด้วยมือ	59
3.26	ค่าอุณหภูมิ Setpoint	59
3.27	สถานะโหมดเครื่องปรับอากาศ	59
3.28	หน้าจอทัชสกรีน	61
3.29	หน้าจอทัชสกรีนแสดงตอนเริ่มทำงาน	61
3.30	กราฟ Cooling load แสดงความสัมพันธ์ระหว่าง Cooling load % กับ Solar Time	62
3.31	การตั้งค่าเวลาเปิดปิดเครื่องปรับอากาศผ่านหน้าเว็บบราวเซอร์	63
3.32	หน้าเว็บบราวเซอร์	63
3.33	หน้าเว็บบราวเซอร์ที่ใช้ควบคุมเครื่องปรับอากาศทั้งหมด (5 เครื่อง)	64
3.34	หน้าจอทัชสกรีน	65
3.35	เครื่องปรับอากาศตัวที่ 1 – 5	65
3.36	สถานะของเครื่องปรับอากาศ	66
3.37	ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศ	67-70
3.38	ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด	71-73
4.1	การติดตั้งอุปกรณ์ควบคุมกับคอมพิวเตอร์	75
4.2	การเข้า IP address 161.246.20.64	75
4.3	การกดปุ่มคำสั่ง OPEN ของเครื่องปรับอากาศตัวที่ 1 (Air no.1)	75
4.4	การตั้งเวลาเปิด-ปิดเครื่องปรับอากาศ	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.5 การทำงานของเครื่องปรับอากาศตัวที่ 1 (LED 1 ติด).....77
4.6 การติดตั้งอุปกรณ์ควบคุมกับคอมพิวเตอร์.....79
4.7 การกดปุ่มคำสั่ง OPEN ที่หน้าจอทัชสกรีน.....79
4.8 การกดปุ่มคำสั่ง SHUT ที่หน้าจอทัชสกรีน.....79

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันการใช้พลังงานเป็นปัญหาสำคัญของประเทศและมีบทบาทเกี่ยวข้องกับการดำเนินชีวิตในหลายรูปแบบ ไฟฟ้าเป็นอีกพลังงานที่มีความสำคัญและมีการใช้งานเพิ่มมากขึ้นเป็นอย่างมากในปัจจุบัน ทั้งนี้เนื่องจากประเทศไทยตั้งอยู่ในเขตภูมิอากาศที่มีอากาศร้อนเกือบตลอดทั้งปี ทำให้ต้องอาศัยการใช้งานจากเครื่องปรับอากาศเพื่อทำให้เกิดความรู้สึกเย็นสบายซึ่งมีการใช้พลังงานที่สูงและทำให้เสียค่าใช้จ่ายมากเมื่อมีการใช้งานเป็นเวลานาน อีกทั้งการใช้งานเครื่องปรับอากาศในที่สาธารณะ เช่น ภายในอาคาร ร้านอาหารสำนักงาน โรงเรียน มหาวิทยาลัย เป็นต้น ยากต่อการควบคุมการใช้งานทั้งอาคาร ซึ่งปัญหาเบื้องต้นในการใช้งานเครื่องปรับอากาศที่ติดตั้งอยู่ ภายในห้องหรือสถานที่ต่างๆ มีดังนี้ เช่น เมื่อห้องไม่มีตารางการใช้งาน จะมีผู้ใช้งานบางกลุ่มเข้าไปใช้งานห้องและทำการเปิด-ปิด เครื่องปรับอากาศ หรือ เมื่อผู้ใช้งานใช้งานห้องเสร็จแล้ว ไม่ทำการปิดเครื่องปรับอากาศ เป็นต้น จึงทำให้เราสูญเสียพลังงานไปเป็นอย่างมาก อีกทั้งการติดตั้งเครื่องปรับอากาศนั้นจะต้องมีการติดตั้งสวิทซ์เพื่อใช้สำหรับเปิด-ปิดให้อุปกรณ์สามารถทำงานได้ ซึ่งหากมีการติดตั้งจำนวนเครื่องมากเพิ่มขึ้นจำนวนสวิทซ์ก็จะต้องเพิ่มขึ้น และยิ่งหากแต่ละตำแหน่งที่ติดตั้งอยู่ไกลกันการดูแลการทำงานนั้นก็ค่อนข้างจะไม่สะดวกและเป็นการยากในการควบคุมการเปิด-ปิดและตรวจสอบความผิดพลาดได้ยาก จากปัญหาที่ได้กล่าวมา ทำให้สิ้นเปลืองพลังงานและค่าใช้จ่ายที่ใช้กับเครื่องปรับอากาศเป็นอย่างมาก

ดังนั้น เพื่อลดการใช้พลังงานไฟฟ้าของระบบปรับอากาศ และสะดวกในการเข้าควบคุมดูแลระบบเครื่องปรับอากาศ จึงออกแบบระบบควบคุมการเปิด-ปิดเครื่องปรับอากาศให้เป็นระบบ โดยสามารถควบคุมการทำงานได้จากตัวควบคุมส่วนกลางเพียงตัวเดียว (master control) ในห้องควบคุมส่วนกลางรวมทั้งเราสามารถดูการทำงาน สั่งการควบคุมเครื่องปรับอากาศแต่ละตัวผ่านทางเว็บบราวเซอร์ และผ่านทางหน้าจอ HMI ได้อีกด้วย ซึ่งจะมีส่วนช่วยทำให้การใช้งานเครื่องปรับอากาศมีประสิทธิภาพมากและสะดวกในการควบคุมดูแลตรวจสอบการทำงานมากขึ้น สามารถแสดงสถานะความพร้อมในการใช้งาน สามารถเปิดปิดใช้งานตัวเครื่องปรับอากาศจากการควบคุมด้วย Arduino ซึ่งจะต้องทำงานอย่างอัตโนมัติตามเวลาที่ผู้ใช้งานกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 ความมุ่งหมายและวัตถุประสงค์ของโครงงาน

1. เขียนโปรแกรมเพื่อให้ Arduino Board ควบคุมการเปิด – ปิดเครื่องปรับอากาศตามเวลาที่กำหนดซึ่งรวมถึงการต่อวงจรประกอบชุดควบคุมการเปิด – ปิดกับเครื่องปรับอากาศได้
2. เพื่อให้ผู้ใช้สะดวกในการดูแลควบคุมการใช้งานเปิด – ปิดเครื่องปรับอากาศโดยผ่านทางเว็บบราวเซอร์ที่สามารถทำได้ทุกเวลา และควบคุมผ่านจอทัชสกรีนที่อยู่ในห้องควบคุมส่วนกลาง

1.3 ขอบเขตของโครงงาน

1. เขียนโปรแกรมให้ Arduino board สามารถสั่งเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนด
2. สามารถควบคุมการเปิด-ปิดเครื่องปรับอากาศผ่านระบบอินเตอร์เน็ต ด้วยเว็บบราวเซอร์
3. สามารถควบคุมการเปิด-ปิดเครื่องปรับอากาศผ่านหน้าจอ HMI ด้วยการใช้หน้าจอทัชสกรีน
4. สามารถต่อควบคุมระดับการทำความเย็นของเครื่องปรับอากาศได้อัตโนมัติโดยผ่านตัวเซนเซอร์อุณหภูมิที่ติดไว้ในห้องนั้นๆ

1.4 ขั้นตอนการศึกษา

1. ศึกษาการนำไปใช้งานต่างๆของ Arduino และอุปกรณ์ที่ใช้งานร่วมกับบอร์ด Arduino
2. ศึกษาเงื่อนไขการทำงาน และออกแบบฮาร์ดแวร์
3. ทำการจัดซื้ออุปกรณ์ต่างๆ และทำการต่อฮาร์ดแวร์
4. ศึกษาการเขียนโปรแกรม Arduino
5. ทำการเขียนโปรแกรมลง Arduino และทดสอบการทำงานเบื้องต้น
6. ทำการเขียนโปรแกรมทดสอบการติดต่อระหว่าง Arduino กับ อุปกรณ์ที่ออกแบบไว้ทีละตัวให้ทำงานตามที่ต้องการ
7. ทำการเชื่อมต่อ Arduino กับ อุปกรณ์ทุกตัวที่ใช้ และรวมโปรแกรมเขียนไว้เพื่อ สั่งเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและโครงงานที่เกี่ยวข้อง

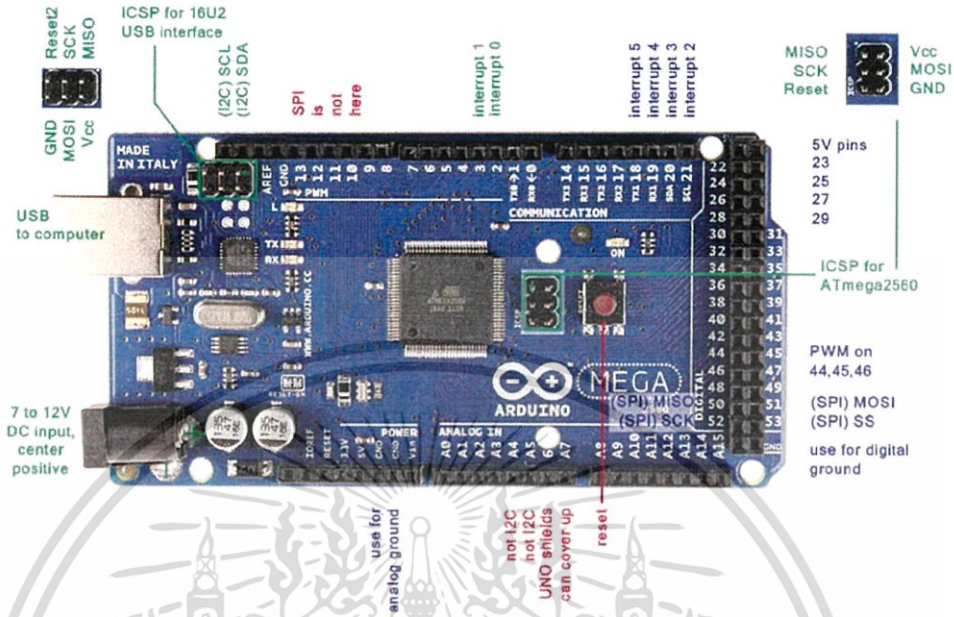
2.1 ความรู้เบื้องต้นเกี่ยวกับ Arduino MEGA 2560 R3 board

เป็นบอร์ดไมโครคอนโทรเลอร์ตระกูล AVR ที่มีการพัฒนาแบบ Open Sourceคือมีการเปิดเผยข้อมูลทั้งด้าน Hardware และ Software ซึ่งตัวบอร์ดถูกออกแบบมาให้ใช้งานได้ง่ายจึงเหมาะสำหรับผู้เริ่มต้น รวมทั้งผู้ใช้งานยังสามารถดัดแปลง,เพิ่มเติมและพัฒนาต่อยอดทั้งตัวบอร์ด หรือโปรแกรมต่อได้บอร์ด Arduino รุ่นนี้สามารถต่ออุปกรณ์เสริมต่างๆได้สะดวก คือผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขา I/O ของบอร์ดหรือเพื่อความสะดวกสามารถเลือกต่อกับบอร์ดเสริม (Shield) ประเภทต่างๆ เช่น XBee Shield, Music Shield, Relay Shield, Wireless Shield, GPRS Shield เป็นต้น มาต่อกับบอร์ด Arduino แล้วเขียนโปรแกรมพัฒนาต่อได้เลย

Arduino Mega 2560 ถือเป็นบอร์ดรุ่นใหญ่ในของตระกูล Arduino มีคุณสมบัติต่างๆ เพิ่มขึ้นจาก Arduino Uno R3 โดยจะใช้ชิพ ATmega2560 ที่มีหน่วยความจำแฟลช 256 KB แรม 8 KB ใช้ไฟเลี้ยง 7 ถึง 12 V แรงดันของระบบอยู่ที่ 5 V มี Digital Input / Output มากถึง 54 ขา (เป็น PWM ได้ 14 ขา) มี Analog Input 16 ขา Serial UART 4 ชุด I2C 1 ชุด SPI 1 ชุด เขียนโปรแกรมบน Arduino IDE และโปรแกรมผ่าน USB จะเห็นได้ว่าเป็น Arduino ที่มีหน่วยความจำและขาสัญญาณต่างให้ต่อใช้งานมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 โครงสร้างพื้นฐานของ Arduino MEGA 2560 R3 board



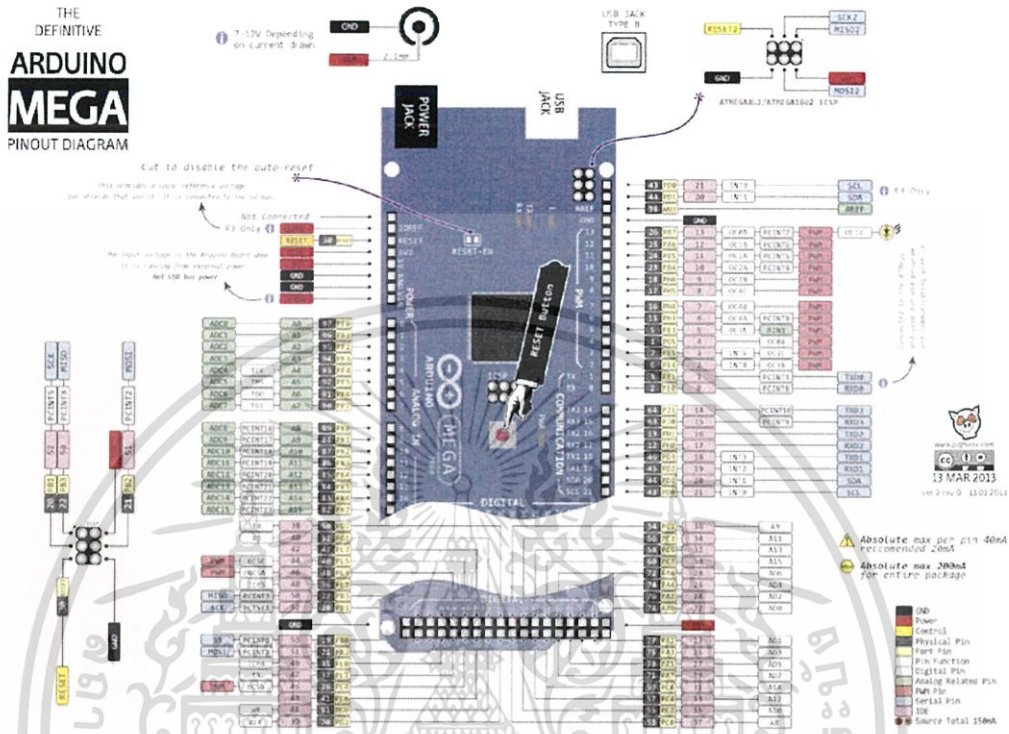
รูปที่ 2.1 โครงสร้างพื้นฐานของ Arduino MEGA 2560 R3 board

2.1.1.1 หน้าที่และประโยชน์การใช้งานของพอร์ทต่างๆ

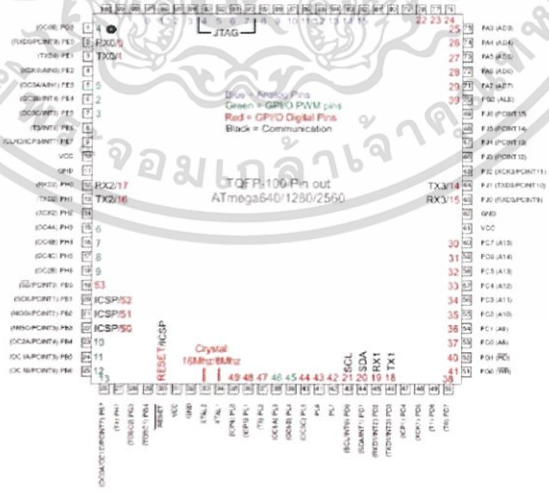
- **USB JACK** ใช้สาหรับต่อกับ Computer เพื่ออัพโหลดโปรแกรมเข้า MC และจ่ายไฟให้กับบอร์ด
- **Reset Button** เป็นปุ่ม Reset ใช้กดเมื่อต้องการให้ MCU เริ่มการทางานใหม่
- **ICSP Port (In-circuit serial programming) Atmega2560** เป็นพอร์ตที่ใช้โปรแกรม Bootloader
- **Digital Pins** Digital I/O ตั้งแต่ขา D0 ถึง D13 นอกจากนี้ บาง Pin จะทำหน้าที่อื่นๆ เพิ่มเติมด้วย เช่น Pin0,1 เป็นขา Tx,Rx Serial, Pin3,5,6,9,10 และ 11 เป็นขา PWM
- **MCU** (Microcontroller) Atmega 2560 เป็น MCU ที่ใช้บนบอร์ด Arduino
- **Analog Pins** นอกจากจะเป็น Digital I/O แล้ว ยังเปลี่ยนเป็น ช่องรับสัญญาณอนาล็อก ตั้งแต่ขา A0-A5
- **Power Pins** ไฟเลี้ยงของบอร์ดเมื่อต้องการจ่ายไฟให้กับวงจรภายในประกอบด้วยขาไฟเลี้ยง +3.3 V, +5V, GND, Vin
- **Power Jack** รับไฟจาก Adapter โดยที่แรงดันอยู่ระหว่าง 7-12 V
- **Power LED** ตัวแสดงสถานะว่าตอนนี้มีไฟเลี้ยงจ่ายเข้าที่บอร์ด
- **Pins 13(LED)** ตัวแสดงสถานะว่า Pin ใดๆใน13 Pins ตอนนี้ถูกใช้งานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **Rx,Tx (LED)** ตัวแสดงสถานะว่า Pin Rx,Tx ตอนนี้ถูกใช้งานอยู่



รูปที่ 2.2 Pinout Diagram ของ Arduino MEGA 2560 R3 board



รูปที่ 2.3 Pinout Diagram ของ MCU (Microcontroller) Atmega2560

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดทดลองไมโครคอนโทรลเลอร์ Arduino MEGA 2560 R3 board จะมีLEDแสดงผลต่อไว้กับตำแหน่งขา Digital 13 สามารถนำไปเขียนโปรแกรมทดสอบการทำงานของบอร์ดทดลองได้

2.1.2 คุณสมบัติเบื้องต้นของ Arduino MEGA 2560 R3 board

Arduino MEGA 2560 R3 board เป็น Microcontroller board ที่ใช้ ATmega 2560 เป็น MCU หลักซึ่งตัวนี้จะมีขา Digital 54 ขา อินพุท/เอาต์พุท (สามารถทำเป็น PWM ได้ถึง 14 ขา) และมีขา Analog อินพุทได้อีก 16 ขา, รันที่ความถี่ 16 MHz มี USB Connector และ Power Jack DC ซึ่ง Concept ของ Arduino Board นี้ทำมาเพื่อความสะดวก ง่ายในการเชื่อมต่อเข้ากับคอมพิวเตอร์ สามารถต่อ USB เข้ากับช่องคอมพอร์ต ก็สามารถ Run โปรแกรมที่ Board ได้ เหมาะสำหรับผู้ที่กำลังเริ่มต้นศึกษาด้าน
โดยมีรายละเอียดดังนี้

Microcontroller	ATmega 2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output , 4 UART TTL)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader used by boot loader
SRAM	8 KB (ATmega2560)
EEPROM	4 KB (ATmega2560)
Clock Speed	16 MHz

2.1.3 Arduino Software

- ใช้ซอฟต์แวร์ Arduino IDE ในการเขียนโปรแกรม มีสองเวอร์ชั่น 1.0.x และ 1.5.x และใช้สำหรับระบบปฏิบัติการ Windows, Linux, Mac OS X รวมทั้งใช้ภาษา C/C++ ในการเขียนโปรแกรม (เรียกว่า Arduino Sketch) ดังนั้นสามารถเขียนโปรแกรมเชิงวัตถุได้ (Object-oriented Programming: OOP) คลาส (Class) หรือ ออปเจค (Object) เกี่ยวข้องกับการใช้งานฮาร์ดแวร์ต่างๆ ของตัวประมวลผล หรือเรียกว่า ไมโครคอลโทรลเลอร์ (Microcontroller)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีชุดคำสั่งสำหรับใช้งานของ Arduino โดยมีการสร้างฟังก์ชั่น (Functions) หรือคำสั่งไว้ให้เรียกใช้งานหลายคำสั่ง และจัดทำในรูปของไลบลารี่ในภาษา C++ (C++ Library) ต่างๆ

ภาษาซีของ Arduino จะจัดแบ่งรูปแบบโครงสร้างของการเขียนโปรแกรมเป็นส่วนย่อยๆ หลายๆ ส่วนโดยเรียกแต่ละส่วนว่า "ฟังก์ชั่น" และเมื่อนำฟังก์ชันมารวมเข้าด้วยกัน ก็จะเรียกว่า "โปรแกรม" โดยโครงสร้างการเขียนโปรแกรมของ Arduino ทุกๆ โปรแกรมจะประกอบไปด้วยฟังก์ชั่นจำนวนเท่าใดก็ได้แต่อย่างน้อยสุดต้องมี 2 ฟังก์ชั่นคือ setup() และ loop()

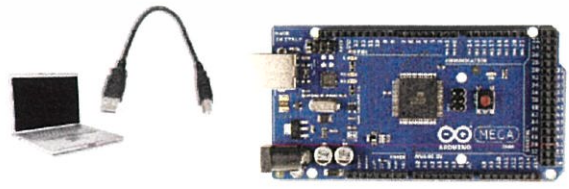
- **Setup()** : เป็นฟังก์ชั่นบังคับที่ต้องกำหนดให้มีทุกๆโปรแกรม ถึงแม้ว่าบางโปรแกรมจะไม่ต้องการใช้งานก็ยังจำเป็นต้องประกาศไว้เสมอ เพียงแต่ไม่ต้องเขียนคาสั่งใดๆไว้หลังวงเล็บปีกกา {} ที่ใช้เป็นตัวกำหนดขอบเขตของฟังก์ชัน โดยฟังก์ชันนี้จะใช้สาหรับบรรจุคำสั่งที่ต้องการให้โปรแกรมทางานเพียงรอบเดียว ตอนเริ่มต้นทำงานของโปรแกรมครั้งแรกเท่านั้น ซึ่งได้แก่คำสั่งเกี่ยวกับการ Setup ค่าการทำงานต่างๆ เช่นการกำหนดหน้าที่ของการใช้งานของ Pin Mode และค่า Baud rate สำหรับการใช้งานสื่อสารพอร์ตอนุกรม เป็นต้น

- **Loop()** : เป็นส่วนฟังก์ชั่นบังคับที่ต้องกำหนดให้มีในทุกโปรแกรม เช่นเดียวกันกับฟังก์ชั่น Setup() โดยฟังก์ชั่น Loop() นี้จะใช้ในการบรรจุคำสั่งที่ต้องการให้โปรแกรมทำงานเป็นวนรอบซ้ำๆ กันไปไม่รู้จบ ซึ่งเปรียบเทียบได้กับ ฟังก์ชั่น main () ใน ANSCI-C

2.2 ขั้นตอนการเริ่มต้นใช้งาน Arduino MEGA 2560 R3 board

1. ดาวน์โหลด Arduino IDE มาไว้ที่เครื่องคอมพิวเตอร์ เพราะไดร์เวอร์สำหรับบอร์ดนั้นรวมอยู่ในไฟล์นี้ด้วย โดยดาวน์โหลดได้จาก http://arduino.cc/en/Main/Software พร้อมทั้ง Unzip ไว้ที่ใดก็ได้ตามต้องการ

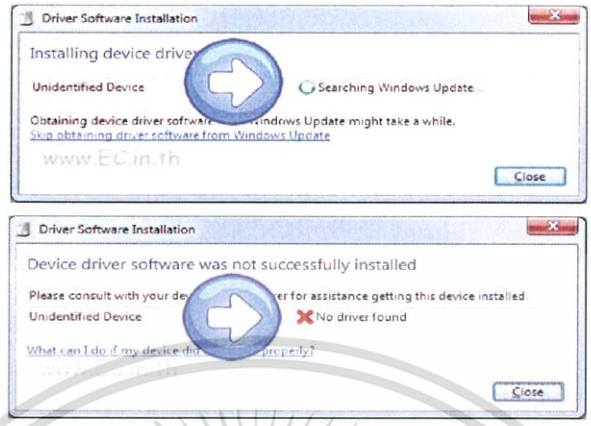
2. ต่อบอร์ด Arduino MEGA 2560 R3 เข้ากับคอมพิวเตอร์ผ่านทาง USB PORT ดังรูปที่ 2.4



รูปที่ 2.4 ตัวอย่างต่อบอร์ด Arduino MEGA 2560 R3 เข้ากับคอมพิวเตอร์ผ่านทาง USB PORT

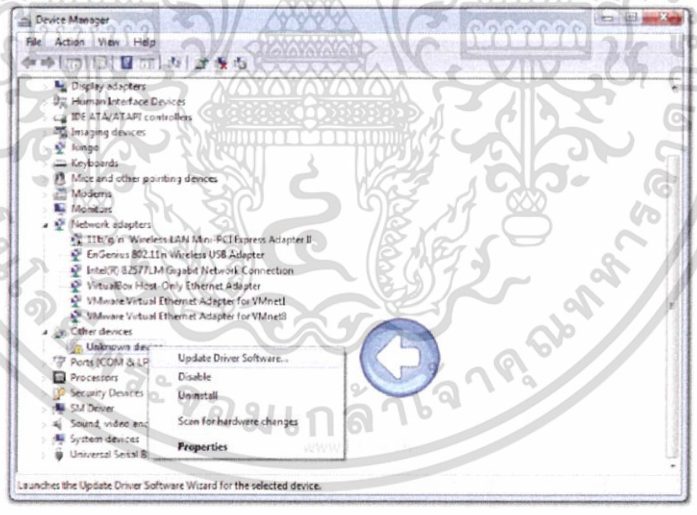
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. จากนั้น windows จะทำการหาไดร์เวอร์แต่จะไม่พบ ดังรูปที่ 2.5



รูปที่ 2.5 หน้าต่างการหาไดร์เวอร์ของ windows

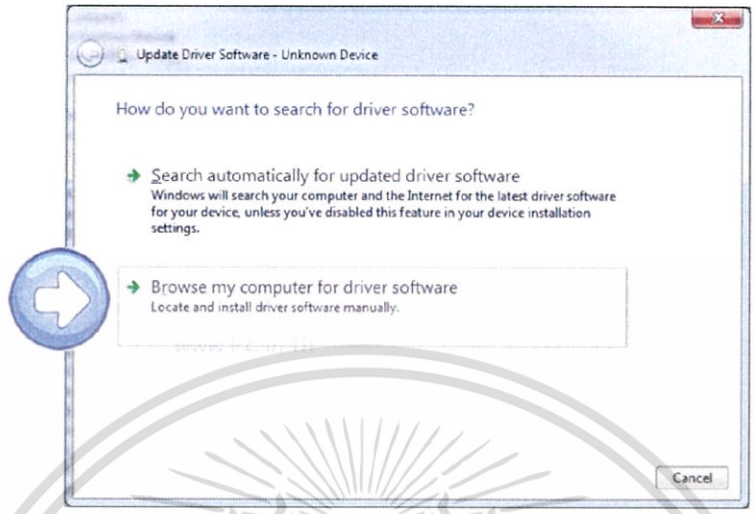
4. ให้เข้าไปที่ device manager แล้วคลิกขวาที่ unknown แล้วเลือก update driver ดังรูปที่ 2.6



รูปที่ 2.6 การแก้ปัญหาเมื่อ windows หาไดร์เวอร์ไม่เจอ

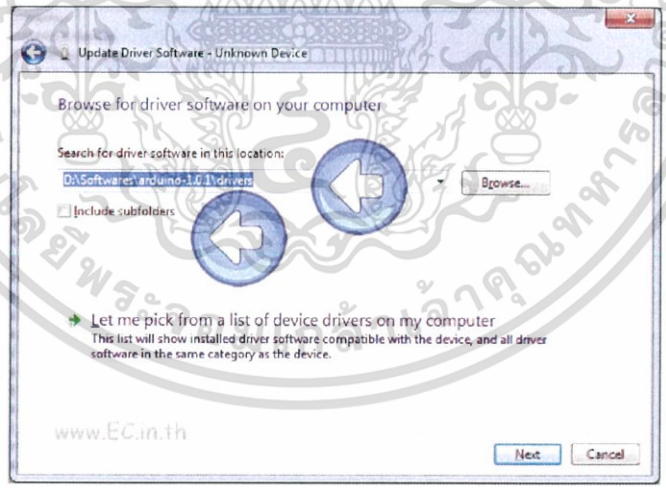
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. เลือกเมนู browse my computer ดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างหน้าต่างการเลือกเมนู browse my computer

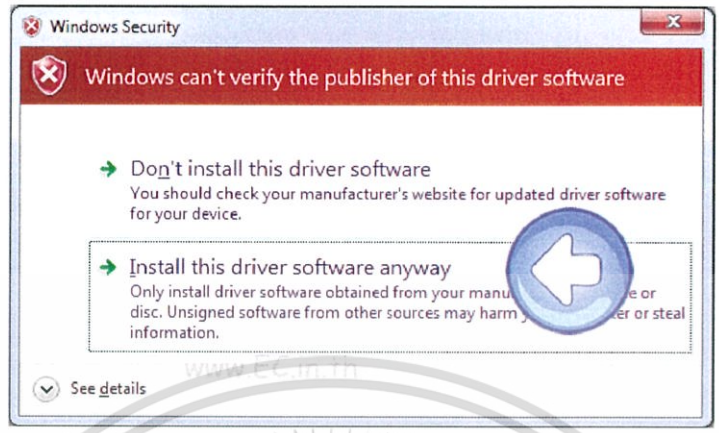
6. เลือก browse ไปที่ไดร์เวอร์ Arduino โดยจะอยู่ที่ X:xxx\arduino-x.x.x\drivers และเอาเครื่องหมายที่ช่อง Include subfolders ออก ดังรูปที่ 2.8



รูปที่ 2.8 การ browse ไปที่ไดร์เวอร์ Arduino

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. เลือก Install this driver software anyway ดังรูปที่ 2.9



รูปที่ 2.9 การ Install this driver software anyway

8. เมื่อสิ้นสุดขั้นตอนนี้ Windows ก็จะรู้จักบอร์ด Arduino MEGA 2560 R3

2.3 วิธีการติดตั้ง Arduino software 1.0.5

การดาวน์โหลด Arduino IDE มาไว้ที่เครื่องคอมพิวเตอร์ดาวน์โหลดได้จาก http://arduino.cc/en/Main/Software โดยมีขั้นตอนวิธีการดังนี้

1. เข้าที่หน้าเว็บไซต์ http://arduino.cc จะได้หน้าเว็บขึ้นมา ดังรูปที่ 2.10

รูปที่ 2.10 หน้าหลักของ เว็บไซต์ http://arduino.cc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 กดเลือกที่เมนู Download จะได้หน้าเว็บขึ้นมา ดังรูปที่ 2.11

รูปที่ 2.11 หน้าต่างที่ได้หลังจากกดเมนู Download

3. กดเลือกที่ PREVIOUS RELEASES เพื่อจะดาวน์โหลด ARDUINO 1.0.5 จากนั้นกดเลือกที่ Windows **หรือ** Windows installer เมื่อดาวน์โหลดเสร็จสิ้นจะได้เป็น zipไฟล์ ดังรูปที่ 2.12

รูปที่ 2.12 หน้าดาวน์โหลด ARDUINO 1.0.5

4. เมื่อกดเลือกที่ Windows ZIP file for non admin install แล้วในกรณีที่เลือกรายการ Windows Installer จะทำการดาวน์โหลด EXE ไฟล์ (arduino-1.0.5.exe) และ เมื่อทำการดาวน์โหลดเสร็จ จะขึ้นหน้าจอถามว่าจะ run โปรแกรมหรือ save โปรแกรม ให้เลือก save โปรแกรมเพื่อนำ โปรแกรมไปใช้ติดตั้งในภายหลัง หลังจากนั้นให้ทำการดับเบิลคลิก โปรแกรมที่เก็บไว้ จะปรากฏหน้าจอแจ้งข้อมูล เรื่องลิขสิทธิ์ให้คลิก I Agree ดังรูปที่ 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.13 หน้าต่างแจ้งข้อมูลเรื่องลิขสิทธ์ก่อนทำการติดตั้ง

5. หลังจากนั้นจะปรากฏหน้าจอสอบถามตำแหน่งโฟลเดอร์ที่จะติดตั้งโปรแกรมให้กดปุ่ม Browseแล้วเลือกตำแหน่งโฟลเดอร์ที่ต้องการแล้วกดปุ่ม Install เมื่อโปรแกรมทำการติดตั้งเสร็จจะปรากฏ shortcut สัญลักษณ์ของ Arduino ที่หน้าจอ Desktop

2.4 ตัวอย่างฟังก์ชันการทำงานต่างๆของ Arduino MEGA 2560 R3 ที่เกี่ยวข้องกับโครงงาน

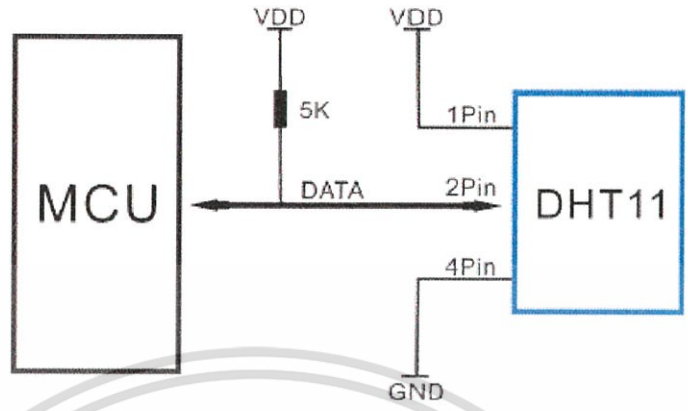
2.4.1 Temperature Sensor โมดูลเพื่อวัดค่าอุณหภูมิและความชื้นจากสิ่งแวดล้อม

โดยจะใช้ DHT11 Humidity and Temperature Sensor ร่วมกับบอร์ด Arduino เป็น Module ที่ใช้ในการวัดอุณหภูมิกับความชื้นในอากาศ โดยคุณสมบัติของ DHT11 มีดังนี้

- เป็นย่านวัดความชื้น 20 – 90% RH โดยมีค่าความแม่นยำ +- 5% RH ความละเอียดในการวัด 1 % แสดงผลแบบ 8 บิต
- ย่านวัดอุณหภูมิ 0 -50 องศาเซลเซียส โดยมีค่าความแม่นยำ +- 2 องศาเซลเซียส ความละเอียดในการวัด 1 องศาเซลเซียส แสดงผลแบบ 8 บิต
- มี PIN 4 ขารายละเอียดดังรูปด้านบน
- กินกระแส 0.5 - 2.5 mA (ขณะทำการวัดค่า) ที่ระดับแรงดัน 3 - 5.5 VDC
- อ่านค่าสัญญาณ (Sample Rate) ทุก 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1.1 วิธีการต่อใช้งาน DHT11 กับบอร์ด Ardiono



รูปที่ 2.14 วิธีการต่อใช้งาน DHT11 กับบอร์ด Arduino

ในการการต่อวัดแบบปกติ คือ ระยะห่างระหว่าง Sensor กับตัว Arduino ห่างกันไม่เกิน 20 เมตรต้องใช้ Pull up resistor ขนาด 5 kΩ (ต่อตัวต้านทาน 5k ไว้กับแหล่งจ่ายแรงดันและต่อเข้าไปที่ขา DATA)

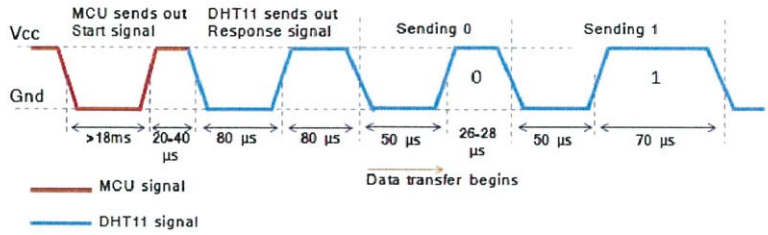
- Pin 1 ต่อกับ VDD
- Pin 2 ต่อเป็นขา DATA
- Pin 3 ไม่ได้ใช้
- Pin 4 ลงกราวด์

โดยใช้แหล่งจ่ายแรงดัน VDD ขนาด 3-5.5 VDC โดยข้อดีคือจะทำให้ DHT11 นี้สามารถใช้งานได้กับ Arduino หลายรุ่น เช่น Arduino Due หรือรุ่น UNO และ Mega/Mega ADK

2.4.1.2 วิธีการส่งข้อมูล DHT11 กับบอร์ด Arduino

การสื่อสารกับ Arduino โดยผ่าน MCU ด้วยวิธี Single-wire Two-way Serial interface หรือการสื่อสารอนุกรมสองทางโดยใช้สายเส้นเดียวโดยการสื่อสารแบบนี้จะใช้สายสื่อสารเพียงเส้นเดียวและส่งข้อมูลได้ทั้งจาก MCU ไปที่ตัว DHT11 และในทางกลับกันก็สามารถส่งข้อมูลได้ทั้งจาก DHT11 ไปที่ตัว MCU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 Timing diagram แสดงการ ส่งข้อมูล DHT11 กับ MCU

2.4.2 โมดูล Real Time Clock (RTC)

โมดูล Real Time Clock (RTC) ก็คือ อุปกรณ์ที่ให้ค่าเวลาตามจริง โดยมีหลักทำงานโดยการจับสัญญาณนาฬิกาที่ได้มาจาก Crystal ซึ่งบางรุ่นจะมีถ่านสำรองมาเพื่อทำหน้าที่ในการบันทึกเวลาอย่างต่อเนื่องถึงแม้ว่าจะไม่มีไฟเลี้ยงมาที่ตัวบอร์ดทำให้เวลาก็ยังคงนับได้ต่อได้ ทำให้ไม่ต้องเสียเวลามาตั้งเวลาใหม่หลังจากที่หยุดจ่ายไฟเลี้ยง

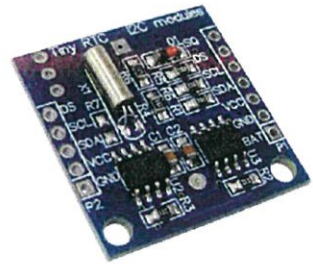
โมดูล RTC นี้จำเป็นอย่างยิ่งกับการใช้งานที่ต้องมีการบันทึกเวลา (Time Stamp) เช่น อุปกรณ์ Data logger เป็นต้น

สาเหตุที่ต้องการ RTC module นี้ถึงแม้ว่า Arduino Board มีตัวจับเวลา เช่น millis() อยู่แล้วคือ ไมโครโปรเซสเซอร์ที่เป็นหัวใจในการทำงานของ Arduino Board ทั้งหลายนั้นต้องทำงานหลายฟังก์ชัน ไม่ว่าจะเป็นคำสั่งพื้นฐาน เช่น บวก ลบ คูณและหาร ไปจนถึงการติดต่อกับอุปกรณ์ภายนอก คำสั่งมากมายซึ่งเราเป็นผู้เขียนลงใน Sketch นั้นจะทำงานแบบ อนุกรม (Serial) ก็คือ ทำงานทีละบรรทัด ทำให้การทำงานของคำสั่งจับเวลานั้นจะถูกรบกวนไปด้วย ทำให้เกิดผลเสียต่อการจับเวลา (จับเวลาได้ไม่ต่อเนื่อง) เพราะถูกแทรกแซง (Interrupt) จากคำสั่งอื่นๆ เวลาที่ได้จากการใช้คำสั่งนี้ จึงไม่สามารถนำมาเป็นเวลาตามจริงที่ต้องการบันทึกไปพร้อมกับค่าอื่นๆ ที่ต้องการวัดได้

ดังนั้น ในการประยุกต์ใช้กับงานที่ต้องการเวลาที่แม่นยำ และเป็นเวลาตามนาฬิกา ที่บอก วันที่ เดือน ปี ชั่วโมง นาที วินาทีจึงจำเป็นต้องใช้อุปกรณ์ที่ทำหน้าที่จับเวลาแยก ซึ่งก็ทำให้ต้องมีสัญญาณนาฬิกาจาก Crystal แยกต่างหากด้วยเช่นกัน โดยมีการออกแบบ Chip หลายแบบ ที่ทำหน้าที่นี้ และวิธีใช้งานคือ ติดต่อผ่านไปที่บอร์ดโดยใช้การสื่อสารแบบ I2C หรือ Inter - Integrated Circuit ที่ใช้ SDA SCL VCC และ GND เท่านั้น

แต่อย่างไรก็ตามการที่มีหลายรุ่นก็ทำให้เกิดความสับสนได้ โมดูล RTC นั้นมีอยู่หลายแบบ เช่น DS3231, DS1302, Tiny RTC I2C 24C32 DS1307 แต่ละรุ่นก็จะมีความต่างกันในเรื่องของความละเอียดในการจับเวลา, การมีแบตเตอรี่สำรอง, ขนาด และราคา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 ตัวอย่าง Tiny RTC I2C 24C32 DS1307

2.4.2.1 คุณสมบัติ

- ประกอบด้วย DS1307 RTC IC
- ประกอบด้วยชิป 24C32AN EEPROM สำหรับกรณีมีการใช้งานที่เพิ่มขึ้น
- ประกอบด้วย แบตเตอรี่ LIR2032 ลิเธียมไอออน 3.6V
- ประกอบด้วยส่วนประสาน DS1820 เครื่องวัดอุณหภูมิที่ 1 สาย (ที่มีขาที่แยกต่างหากที่จะควบคุมมันไม่เป็นไปตามมาตรฐาน I2C) แต่มี "BAT" ขาที่จะเชื่อมต่อเข้ากับขา ADC เพื่อวัดแรงดันแบตเตอรี่
- ds1307 ทำงานในรหัสไบนารีทศนิยมหรือ BCD
- สามารถใช้เป็นนาฬิกาและ EEPROM การจัดเก็บตัวแปรทั้งหมดนี้ผ่านทางอินเตอร์เฟสอนุกรมเพียง 2 สาย สายข้อมูล SDA และ SCL มีความจำเป็น และสามารถเชื่อมต่อโดยตรงกับ Arduino MEGA 2560 R3 โดยไม่ต้องมีตัวต้านทานเพิ่ม แบตเตอร์รี่ก็ยังทำงานได้

2.4.2.2 วิธีต่อใช้งาน

Tiny RTC I2C modules 24C32 memory DS1307 clock กับบอร์ด Arduino โมดูลนาฬิกา RTC DS1307 มีอินเตอร์เฟสการใช้งานแบบ I2C ต่อสายใช้งานดังนี้

- พอร์ท VCC ต่อเข้ากับแรงดันไฟฟ้า 5V
- พอร์ท GND ต่อเข้ากับ GND
- พอร์ท SCL ต่อเข้ากับ A5
- พอร์ท SDA ต่อเข้ากับ A4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 Ethernet Shield W5100 R3

ถ้าต้องการให้ Arduino สามารถติดต่อกับเครือข่าย หรือเชื่อมต่อกับอินเตอร์เน็ตได้ เพื่อที่จะได้สะดวกในการควบคุมและติดตามอุปกรณ์ต่าง ๆ เช่น การดึงค่าเซนเซอร์ต่าง ๆ ที่เราติดตั้งไว้ มาดูบนโทรศัพท์มือถือ หรือการเปิดปิดเครื่องใช้ไฟฟ้าผ่านทางอินเตอร์เน็ตเราทำได้โดย Ethernet Shield W5100 R3 ก็สามารถเชื่อมต่อ Arduino อุปกรณ์ต่างๆ ที่ต้องการผ่านทาง Ethernet หรือ Internet ได้

บอร์ด Module Ethernet Shield W5100 ใช้งานโดยเสียบลงบนตัวบอร์ด Arduino Mega ก็สามารถใช้งานได้เลยรวมทั้งยังมีช่องเสียบ SD Card ทำให้สะดวกในการบันทึกข้อมูลจำนวนมาก ๆ ลงบน SD Card ได้และ มีไลบารีมาตรฐานพร้อมใช้งาน

รูปที่ 2.17 แสดงตัวอย่าง Ethernet Shield W5100 R3

การเชื่อมกับ Ethernet Shield นี้จะใช้สาย RJ45 อาจจะใช้ CAT5 หรือ CAT6 โดยสามารถใช้ DHCP ได้ แต่การเชื่อมต่อระหว่างสองจุดยังต้องใช้สาย Cross Over อยู่เพราะไม่มีวงจร Cross Over ภายใน ความเร็วในการสื่อสารของบอร์ดนี้ จะอยู่ที่ 10/50 Mbps หรือ 10/100 Mbps

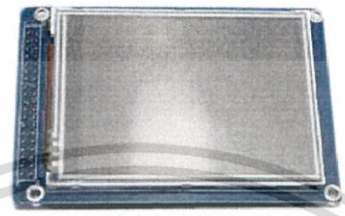
ในส่วนของ Ethernet Shield นี้มีโมดูลที่สามารถรองรับ Power over Ethernet (PoE) ซึ่งสามารถใช้แหล่งจ่ายไฟของบอร์ดได้เลย นอกจากนี้ยังมีไฟแสดงผลซึ่งสามารถอธิบายได้ดังนี้

PWR : ไฟแสดงสัญญาณ Power
LINK : ไฟแสดงสถานการณ์อัพโหลดและดาวน์โหลดข้อมูลผ่านเครือข่าย
FULLD : ไฟแสดงสถานะของการเชื่อมต่อแบบ Full Duplex
100M : ไฟแสดงสถานะเมื่อมีการเชื่อมต่อเครือข่ายได้ถึง 100 Mbps
RX : ไฟแสดงสถานะเมื่อ Ethernet shield มีการรับข้อมูล
TX : ไฟแสดงสถานะเมื่อ Ethernet shield มีการส่งข้อมูล
COLL : ไฟแสดงสถานะเมื่อมี IP ชนกันของเครือข่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 Touch screen (3.2 inch TFT LCD module Display with touch panel SD Card)

เป็นหน้าจอทัชสกรีนที่มีขนาด 3.2 นิ้ว ที่ประกอบไปด้วย SD card ในตัว ดังรูปที่ 2.18



รูปที่ 2.18 ตัวอย่างหน้าจอทัชสกรีน

2.4.4.1 การใช้งานหน้าจอทัชสกรีนนี้กับบอร์ด Arduino

จอชนิดทัชสกรีนทั่วไป โดยมากจะมีแรงดันทำงาน (Operating Voltage) ที่ระดับ 3.3 V แต่บอร์ดบางรุ่น โดยเฉพาะรุ่นยอดนิยมเช่น Arduino UNO R3 และ Arduino Mega 2560 นั้น มีแรงดันที่แต่ละ PIN ที่ 5 V ถ้าเรานำจอ TFT ไปต่อโดยตรงกับบอร์ดจะทำให้เสียหายได้เราจึงมีวิธีแก้ไขคือการใช้ตัวต้านทานมาต่อให้เป็น Voltage Divider ให้กับทุกขาของ TFT แต่วิธีนี้จะใช้เวลามากกว่าวิธีที่สอง ซึ่งคือ การใช้ TFT LCD Mega Shield V2.2 Adapter Module

2.4.4.2 ข้อดีของการใช้ TFT LCD Mega Shield V2.2 Adapter Module

ไม่ต้องกังวลเรื่องการต่อขาให้ตรงกับตัวบอร์ด เพราะ TFT LCD Mega Shield V2.2 Adapter Module ถูกออกแบบมาให้ใช้คู่กับบอร์ด และ Library

2.4.4.3 การใช้งาน

การใช้งานแบ่งออกเป็น 2 ส่วนดังนี้

- **การใช้งานเพื่อแสดงผลเป็นจออย่างเดียว** (Universal TFT : UTFT)

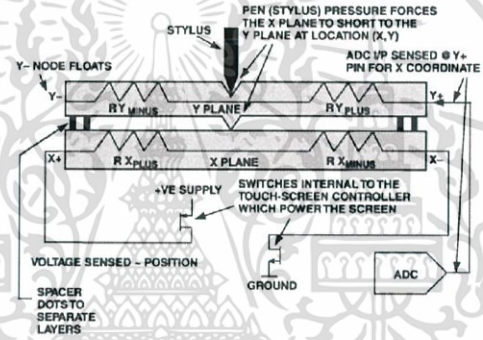
ในการติดต่อคุยกับจอ TFT จะต้องคุยผ่าน Controller Chip ซึ่งจะใช้การสื่อสารแบบ SPI (Serial Peripheral Interface) การสื่อสารแบบนี้มีมาให้เป็นขามาตรฐานของบอร์ด Arduino อยู่แล้ว และเราสามารถสั่งตัว Controller chip กับจอ TFT นี้โดยเรียกใช้ Library ได้เลยโดยไม่ต้องสิ้นเปลือง เวลาในการเริ่มต้นเขียนเอง การสื่อสารและควบคุมบอร์ดก็จะมีทั้งแบบ 8 บิต และ 16 บิต แต่ที่เราจะใช้กับบอร์ดรุ่นเล็กๆ อย่าง Arduino MEGA 2560 R3 ก็จะเป็นแบบ 16 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การใช้งานเป็น Touch Screen**

จอ Touch Screen นั้นทำหน้าที่สำคัญในการติดต่อแสดงผล และ รับคำสั่งจากผู้ใช้งานโดยการสัมผัสที่หน้าจอ Touch Screen จะมีอยู่สองประเภทด้วยกันคือ Resistive Touch Screen และ Capacitive Touch Screen

Resistive Touch Screen เป็นการสัมผัสแล้วจะระบุตำแหน่งโดยใช้การหาค่าทีละแกน (สแกนบนแกน X และ สแกนบนแกน Y ทีละแกน) โครงสร้างของจอ Resistive Touch Screen จะเป็นแผ่นที่ทำจากวัสดุใส 2 ชั้นโดยมีช่องว่างอยู่ตรงกลางแผ่นวัสดุใสดังกล่าวมีคุณสมบัติเป็นตัวต้านทานเมื่อมีการกดหรือสัมผัสลงไปบนหน้าจอโดยการใช้ Stylus หรือใช้นิ้วกดลงไปที่จอจะทำให้แผ่นวัสดุชั้นบนถูกกดลงไปสัมผัสกับชั้นล่าง

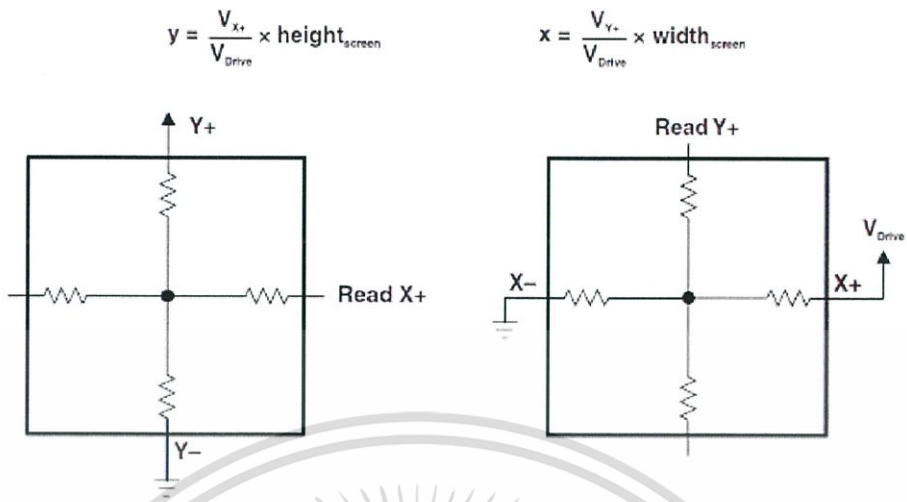


รูปที่ 2.19 การระบุตำแหน่งและโครงสร้างเบื้องต้นของ Resistive Touch Screen

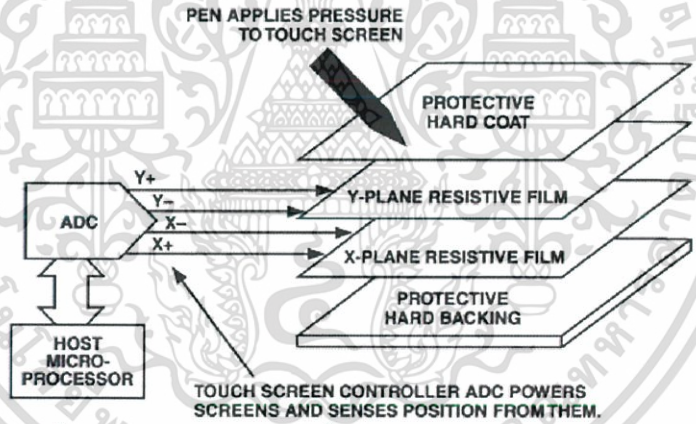
สามารถหาค่าตำแหน่ง (ระยะจากขอบจอ) โดยใช้หลักการแบ่งแรงดันตามสมการในรูปที่ 2.20 เสร็จการสแกนแกนแรกก็ย้ายไปหาค่าในแกนที่เหลือภายในเวลาสั้นๆ

ส่วนที่ใช้ในการควบคุมการสลับแกนและคำนวณระยะห่างจากขอบจอของแกนทั้งสองดังในรูปที่ 2.21 ส่วนที่ใช้ในการระบุตำแหน่งบนจอสัมผัส จะซ้อนอยู่ด้านบนของจอแสดงภาพอีกที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.20 การหาตำแหน่งระยะจากขอบจอจากสมการ



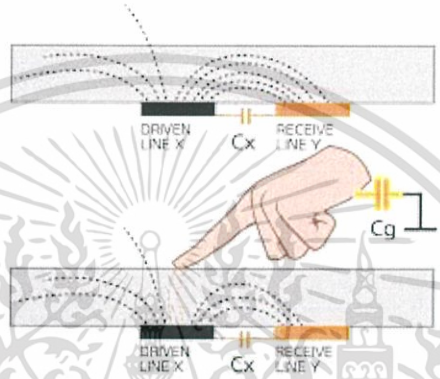
รูปที่ 2.21 หน้าสัมผัสแต่ละชั้นของ Resistive Touch Screen

ดังนั้น Resistive Touch Screen คือ การที่จอสามารถระบุตำแหน่งโดยไม่ต้องมีการสัมผัสโดยตรงกับผู้ใช้งาน (ไม่เหมือนกับ Capacitive Touch Screen) แต่สามารถกดโดยใช้วัสดุใดก็ได้บนหน้าจอทำให้เหมาะสำหรับการใช้ในโรงงานอุตสาหกรรมที่ผู้ปฏิบัติงานต้องใส่ถุงมือ หรือ ในงานที่เกี่ยวกับการแพทย์ซึ่งไม่ต้องการให้ผู้ปฏิบัติงานสัมผัสโดยตรงกับหน้าจอ

ส่วนข้อเสีย คือความละเอียดของจอไม่สูงทำให้การระบุตำแหน่งมีความแม่นยำน้อยกว่าจอแบบอื่นๆ เช่น Capacitive Touch Screen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

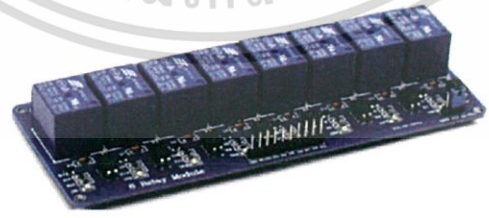
Capacitive Touch Screen หรือที่เราเห็นกันทั่วไปจะใช้กับอุปกรณ์ประเภท Smart Phone ในปัจจุบันโดยการสัมผัสจะมีลักษณะคือ ตำแหน่งที่สัมผัสบนจอจะถูกระบุโดยการสแกนหาตำแหน่งที่มีการเปลี่ยนแปลงของสนามไฟฟ้าเนื่องจากมีนิ้วมือมารับการถ่ายประจุออกไปจากหน้าจอ ส่วนที่เราเห็นว่าการใช้ Stylus ก็สามารถใช้แทนนิ้วมือได้ เพราะด้วยเหตุที่ว่าStylus ที่สามารถใช้ได้กับจอ Capacitive Touch Screen นั้น จะต้องทำจากวัสดุนำไฟฟ้า เนื่องจากต้องมีความสามารถในการถ่ายประจุจากจอไปที่มือของผู้ใช้



รูปที่ 2.22 การถ่ายประจุจากจอไปที่มือของผู้ใช้

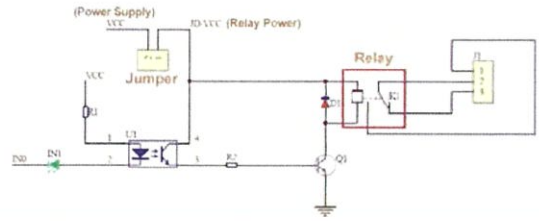
2.4.5 รีเลย์ (8 Channel Relay)

ประกอบด้วยรีเลย์ 8 ตัว เพื่อใช้งานในการควบคุมอุปกรณ์ไฟฟ้า โดยรับกระแสได้สูงถึง 10 A ใช้งานได้ทั้งไฟฟ้ากระแสตรงและกระแสสลับรวมถึงรับแรงดันระดับ 5 V ตรงจาก Arduino board มี LED แสดงสถานะการทำงานของรีเลย์ซึ่งมีการออกแบบให้ป้องกันวงจรด้านควบคุมออกจากด้านกำลังโดยใช้การส่งผ่านด้วยแสง (Optocoupler) ในรีเลย์ทุกตัว



รูปที่ 2.23 8-Channel Relay

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 circuit diagram ของ Relay

2.5 หลักการทำงานของ Arduino Mega 2560 board ที่ถูกควบคุมผ่านระบบอินเตอร์เน็ต

2.5.1 การทำเว็บเซิฟเวอร์ด้วยบอร์ด Arduino Mega 2560 และ Ethernet W5100

การใช้งานบอร์ด Arduino สร้างเป็นเว็บเซิฟเวอร์โดยจะต้องเชื่อมต่อกับ Ethernet W5100 ผ่าน SPI port ซึ่งการสร้างเป็นเว็บเซิฟเวอร์ได้นั้น ต้องนำซอร์สโค้ดมาจาก https://github.com/jcw/ethercard จากนั้น Extract ไฟล์และเพิ่มเข้าไปใน Arduino IDE เป็นการเพิ่มไลบรารี่ (Library) ให้กับ Arduino IDE 1.0.5

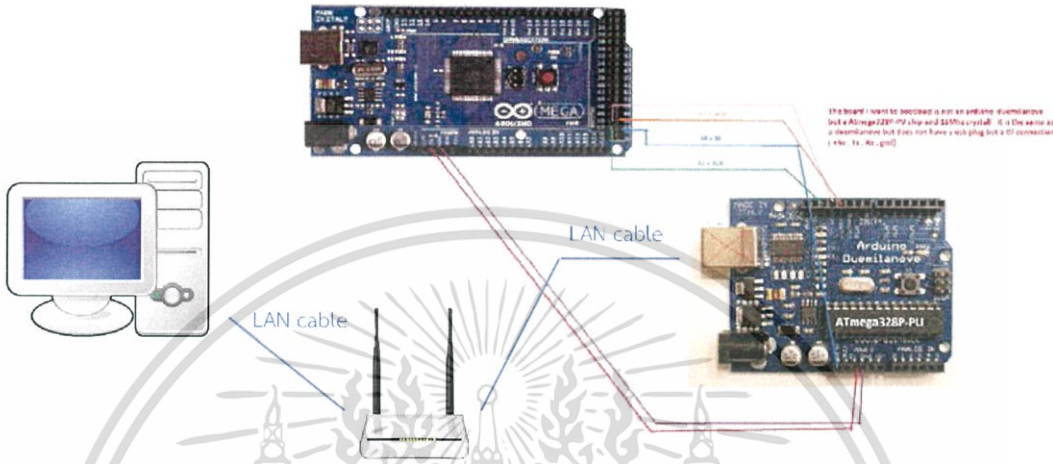
อุปกรณ์ที่จำเป็นต้องใช้มีดังนี้:

1. บอร์ด Arduino Mega 2560
2. Ethernet W5100
3. Adruino IDE 1.0.5
4. การเพิ่มไลบรารี่ (Library) ให้กับ Arduino IDE 1.0.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.1 เพิ่มไลบรารี่ในโปรแกรม Adruino IDE 1.0.5

หลังจากเพิ่มไลบรารี่ในโปรแกรมเรียบร้อยแล้ว ทำการต่ออุปกรณ์ให้เหมือนดังรูปที่ 2.25



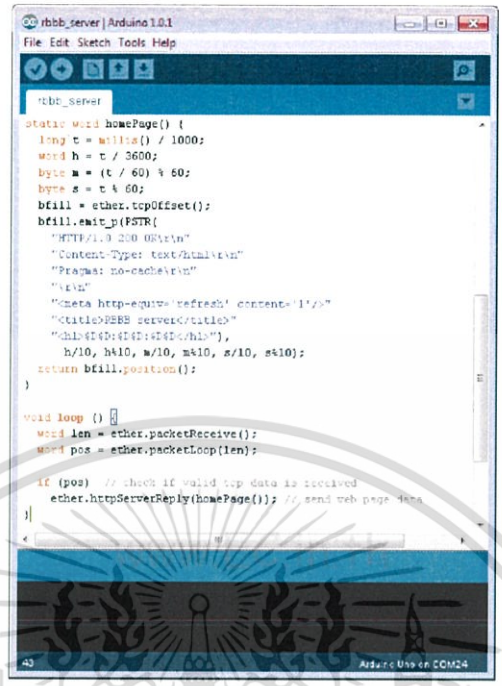
รูปที่ 2.25 การต่ออุปกรณ์ระหว่าง Arduino Board และ Ethernet Shield

2.5.1.2 เปิดตัวอย่างเว็บเซิฟเวอร์

เปิดตัวอย่างจาก File->Examples->jcw_ethercard-rbbb_server ดังรูปที่ 2.26 จะปรากฏ Sketch ดังรูปที่ 2.27 และคอมไพล์

รูปที่ 2.26 การเปิดตัวอย่างเว็บเซิฟเวอร์

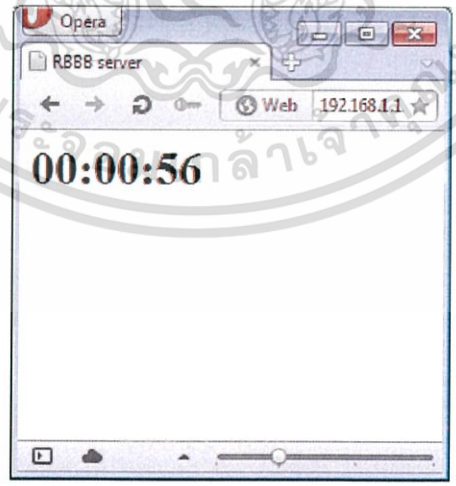
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.27 ตัวอย่าง Sketch ในการเปิดเว็บเซิฟเวอร์

2.5.1.3 ผลของโปรแกรม

โหลดโปรแกรมลงบอร์ดจากนั้นเปิดเว็บบราวเซอร์และเข้าเว็บเพจตามเลข IP(192.168.203) ในโปรแกรม และสังเกตผลของโปรแกรม



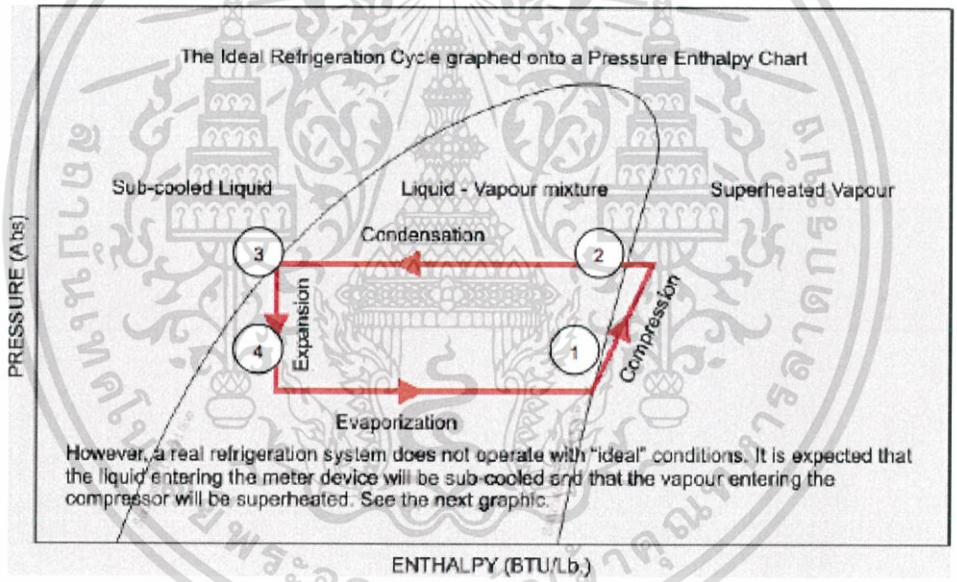
รูปที่ 2.28 ผลของโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 ความรู้เบื้องต้นเกี่ยวกับเครื่องปรับอากาศ

2.6.1 หลักการทำความเย็นของเครื่องปรับอากาศ

หลักการทำงานของระบบปรับอากาศแต่ละประเภทจะแตกต่างกันตามลักษณะการออกแบบ การติดตั้งและใช้งานแต่ทุกระบบโดยส่วนใหญ่จะใช้วัฏจักรการทำความเย็นแบบวงจรอัดไอ โดยมีสารทำความเย็น เช่น R22 หรือ R134a และอื่นๆ เป็นสารที่ทำหน้าที่ดูดและคายความร้อนจากสารตัวกลางอัน ได้แก่อากาศหรือน้ำให้ได้อุณหภูมิตามต้องการเมื่อสารตัวกลางได้รับความเย็นจะถูกส่งไปยังอุปกรณ์ แลกเปลี่ยนความร้อน (ในกรณีที่สารตัวกลางเป็นน้ำ) หรืออากาศเย็นไปยังพื้นที่ปรับอากาศโดยตรง (ในกรณีที่สารตัวกลางเป็นอากาศ) ส่วนความร้อนที่เกิดขึ้นจะถูกส่งไประบายออกที่ชุดระบายความร้อน ซึ่งอาจจะเป็นการระบายความร้อนด้วยอากาศหรือระบายความร้อนด้วยน้ำขึ้นอยู่กับระบบที่เลือกใช้งาน สำหรับส่วนประกอบของวัฏจักรการทำความเย็นนั้นมีส่วนประกอบดังนี้



รูปที่ 2.29 ตัวอย่าง Refrigeration Cycle

- 1 – 2 การอัด (Compression) อัดสารทำความเย็นสถานะก๊าซความดันต่ำ ให้เป็นก๊าซร้อนความดันสูง
- 2 – 3 การควบแน่น (Condensing) สารทำความเย็นสถานะก๊าซควบแน่นเป็นของเหลว และคายความร้อนออก
- 3 – 4 การขยายตัว (Expansion) จากสารทำความเย็นความดันสูงไปเป็นความดันต่ำ พร้อมทั้งลด อุณหภูมิลงและเปลี่ยนสถานะจากของเหลวเป็นของเหลวผสมก๊าซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4 – 1 การระเหย (Evaporation) ความร้อนจากสารตัวกลาง (อากาศหรือน้ำ) จะถูกดูดเพื่อใช้ในการระเหยของสารทำความเย็นเหลวให้เป็นก๊าซ

ระบบการทำความเย็นที่เรากำลังกล่าวถึงคือ ระบบอัดไอ (Vapor-Compression Cycle) ซึ่งมีหลักการทำงานง่ายๆ โดยการทำให้สารทำความเย็น (น้ำยา) ไหลวนไปตามระบบ โดยผ่านส่วนประกอบหลักทั้ง 4 อย่างต่อเนื่องเป็น วัฏจักรการทำความเย็น (Refrigeration Cycle) โดยมีกระบวนการดังนี้

1. เริ่มต้นโดยคอมเพรสเซอร์ทำหน้าที่ดูดและอัดสารทำความเย็นเพื่อเพิ่มความดันและอุณหภูมิของน้ำยา แล้วส่งต่อเข้าคอยล์ร้อน

2. น้ำยาจะไหลวนผ่านแผงคอยล์ร้อนโดยมีพัดลมเป่าเพื่อช่วยระบายความร้อน ทำให้น้ำยาที่จะออกจากคอยล์ร้อนมีอุณหภูมิลดลง (ความดันคงที่) จากนั้นจะถูกส่งต่อให้อุปกรณ์ลดความดัน

3. น้ำยาที่ไหลผ่านอุปกรณ์ลดความดันจะมีความดันและอุณหภูมิที่ต่ำมาก จากนั้นจะไหลเข้าสู่คอยล์เย็น (หรือที่นิยมเรียกกันว่า การฉีดน้ำยา)

4. น้ำยาจะไหลวนผ่านแผงคอยล์เย็นโดยมีพัดลมเป่าเพื่อช่วยดูดซับความร้อนจากภายในห้อง เพื่อทำให้อุณหภูมิห้องลดลง ซึ่งทำให้น้ำยาที่ออกจากคอยล์เย็นมีอุณหภูมิที่สูงขึ้น (ความดันคงที่) จากนั้นจะถูกส่งกลับเข้าไปยังคอมเพรสเซอร์เพื่อทำการหมุนเวียนน้ำยาต่อไป

2.6.2 อุปกรณ์ระบบทำความเย็นแบบคอมเพรสเซอร์อัดไอ

ส่วนประกอบที่สำคัญของระบบการทำการความเย็น (Refrigeration Cycle) มีดังนี้

2.6.2.1 คอมเพรสเซอร์ (Compressor)

คอมเพรสเซอร์ของเครื่องปรับอากาศ ทำหน้าที่ขับเคลื่อนสารทำความเย็นหรือน้ำยา (Refrigerant) ในระบบโดยทำให้สารทำความเย็นมีอุณหภูมิ และความดันสูงขึ้น

2.6.2.2 คอยล์ร้อน (Condenser)

ทำหน้าที่ระบายความร้อนของสารทำความเย็น

2.6.2.3 คอยล์เย็น (Evaporator)

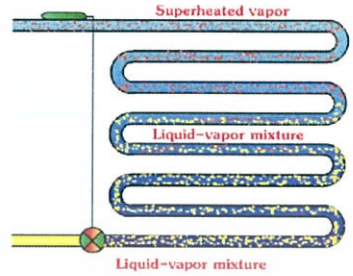
ทำหน้าที่ดูดซับความร้อนภายในห้องมาสู่สารทำความเย็นซึ่งมีการผลิตออกมาหลายรูปแบบแตกต่างกันตามความเหมาะสมของการใช้งานแบ่งได้ 4 ชนิดด้วยกัน

- **Semi flood evaporator หรือแบบแห้ง (dry expansion)**

แบบ Semi flood evaporator หรือแบบแห้ง (dry expansion) นั้นจะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริมาของสารความเย็นที่ป้อนให้คอยล์ถูกกำจัดที่ปริมาณหนึ่งเพื่อให้เกิดการกลายเป็นไอเท่าที่ถูกดูดเข้าเครื่องอัดคอมเพรสเซอร์ ดังรูปที่ 2.30



รูปที่ 2.30 Semi flood evaporator

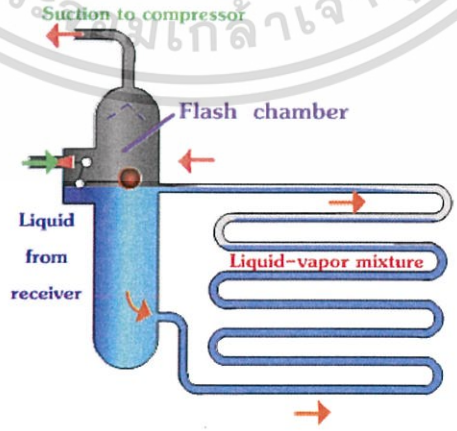
ดังนั้นจะเห็นว่าอีวาโปรเรเตอร์แบบนี้ น้ำยาที่เป็นของเหลวจะผ่านตลอดจนกระทั่งเป็นแก๊ส ไม่มีการแยกระหว่างน้ำยาที่เป็นของเหลวและแก๊สเลยดังนั้นจึงเรียกว่า Dry expansion evaporator

- Flooded evaporator

แบบนี้จะมีถังทำความเย็นซึ่งจะเป็นแหล่งจ่ายสารทำความเย็นโดยใช้แรงโน้มถ่วงของโลกให้กับคอยล์ระดับของเหลวในถังจะถูกควบคุมโดยด้านความดันต่ำหรือสูงก็ได้และไอของสารทำความเย็นที่มาจากคอยล์จะถูกแยกอยู่ส่วนบนของถังเก็บ และถูกดูดออกไปเข้าเครื่องอัด

- Full- flooded evaporator

แบบ Full-flooded นั้นทำงานได้อย่างสมบูรณ์เนื่องจากในท่อบรรจุเต็มไปด้วยสารความเย็นเหลว เพื่อให้ผิวในเปียกเสมออันจะทำให้การถ่ายเทความร้อนมากที่สุดที่จะเป็นไปได้ ดังรูปที่ 2.31

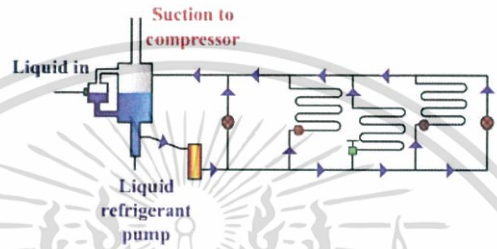


รูปที่ 2.31 Full- flooded evaporator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Liquid overfeed

แบบ liquid overfeed แบบนี้สารทำความเย็นจะถูกป้อนให้คอยล์มากกว่าที่มันจะสามารถระเหยเป็นไอหมดซึ่งส่วนนี้ยังไม่ระเหยจะไหลเวียนในคอยล์ต่อไปส่วนไอจะถูกดูดเข้าเครื่องอัดระบบแบบ overfeed นี้ส่วนนิยมใช้กับระบบที่มีอีวาโปรเรเตอร์หลาย ๆ ชุด ดังรูปที่ 2.32 ในแบบนี้จะมีปัญหาในการแบ่งสารทำความเย็นในแต่ละคอยล์แต่จะลดปัญหาได้ ถ้าเพิ่มอัตราการไหล หรือป้อนมาก ๆและป้องกันการมากจนเกินไปจะมี bypass วาล์วติดตั้งอยู่



รูปที่ 2.32 Liquid overfeed

2.6.2.4 อุปกรณ์ลดความดัน (Throttling Device)

ทำหน้าที่ลดความดันและอุณหภูมิของสาร ทำความเย็น โดยทั่วไปจะใช้เป็นแค็ปพิลลารี่ทิ้วบ์ (Capillary tube) หรือ เอ็กสแปนชั่นวาล์ว (Expansion Valve)

2.6.3 BTU (British Thermal Unit)

คือ ขนาดทำความเย็นของ เครื่องปรับอากาศ มีหน่วยดังนี้ 1 ตันความเย็น เท่ากับ 12000 BTU/hr. เราควรเลือกขนาด BTU ให้เหมาะสมกับขนาด ของห้อง ที่จะทำการติดตั้ง โดยสามารถเลือก ได้จากตารางที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 ตารางแสดงขนาด BTU ที่เหมาะสมกับขนาดของห้อง

Btu/h	ขนาดห้อง (Square metre)	
	Normal Room	ห้องโดนแดด
9,000	12-14	11-13
12,000	16-20	14-18
18,000	20-28	21-27
21,000	28-35	25-32
24,000	32-40	28-35
26,000	35-44	30-39
30,000	40-50	35-45
36,000	48-60	42-54
40,000	56-65	52-60
48,000	64-80	56-72
60,000	80-1,000	70-90

2.6.3.1 เหตุผลที่ต้องเลือก BTU ให้พอเหมาะ เนื่องจาก

กรณีที่ 1 BTU สูงเกินไป คอมเพรสเซอร์ทำงานตัดบ่อยเกินไป ทำให้ประสิทธิภาพในการทำงานลดน้อยลง ทำให้ความชื้นในห้องสูง ไม่สบายตัว และที่สำคัญราคาแพง และสิ้นเปลืองพลังงาน

กรณี 2 BTU ต่ำเกินไป คอมเพรสเซอร์ทำงานตลอดเวลา เพราะความเย็นห้องไม่ได้ตามอุณหภูมิที่ตั้งไว้ สิ้นเปลืองพลังงาน และมีผลให้เครื่องปรับอากาศเสียเร็ว

2.6.3.2 การคำนวณ BTU

เครื่องปรับอากาศ เป็น อุปกรณ์ไฟฟ้าที่ใช้พลังงานมากและสิ้นเปลืองค่าใช้จ่ายสูงโดยเห็นได้จากพลังงานไฟฟ้าโดยรวมของเครื่องใช้ไฟฟ้า ในบ้าน 50-80 เปอร์เซ็นต์ เป็นพลังงานที่ใช้กับเครื่องปรับอากาศแต่อย่างไรก็ตามด้วยอุณหภูมิบ้านเราร้อนขึ้นเรื่อยๆ เครื่องปรับอากาศก็มีความจำเป็นอย่างหลีกเลี่ยงไม่ได้ ดังนั้นเราควรจำเป็นทำความเข้าใจเรื่องค่าใช้จ่ายพลังงานเพื่อนำไปพิจารณาประกอบการเลือกซื้อเครื่องปรับอากาศที่ใช้พลังงานไฟฟ้าอย่างเหมาะสมและไม่เกินความจำเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ตารางแสดงค่าไฟฟ้าจากการติดตั้งเครื่องปรับอากาศต่างสถานที่กัน

Type	Btu/h	Watt	จำนวนหน่วยที่ใช้/1 ชั่วโมง	ระยะเวลา/วัน	หน่วย/เดือน	ค่าไฟ
ติดหน้าต่าง	9000	920	0.92	8 ชั่วโมง คอมเพรสเซอร์ทำงาน 6 ชั่วโมง	165.6	418.97
	12000	1150	1.15		207.0	523.71
	24000	2900	2.99		538.2	1,361.65
ติดผนัง	9000	680	0.68		122.4	309.67
	12000	1130	1.13		203.4	514.60
	24000	2490	2.49		448.2	1,133.92
ตั้ง/แขวน	12000	1330	1.33		239.4	605.68
	24000	2710	2.71		487.8	1,234.13

BTU = พื้นที่ห้อง (กว้าง*ยาว)*ตัวแปร

2.6.3.3 ตัวแปรความร้อน

แบ่งได้ 2 ระดับดังนี้

1. ระดับ 700 คือห้องที่มีความร้อนน้อย ใช้เฉพาะเวลากลางคืน

2. ระดับ 800 คือห้องที่มีความร้อนสูง ใช้กลางวันมาก กรณีเพดานสูงกว่า 2.5 เมตร ให้บวกเพิ่มจากเดิม 5% เพิ่มเติม หากฝ้าเพดานสูงกว่า 2.5 เมตร มีจำนวนคนในห้องมาก หรือมีคอมพิวเตอร์ควรบวกค่า BTU เพิ่มขึ้นอีก 5% จากค่าปกติ

2.6.4 ระบบ Inverter

คือ ระบบที่ควบคุมการปรับอากาศ ให้เป็นอย่างราบเรียบและคงที่ ด้วยการปรับเปลี่ยนรอบการหมุนของคอมเพรสเซอร์โดยการเปลี่ยนความถี่ของกระแสไฟที่จ่ายให้กับมอเตอร์ของคอมเพรสเซอร์แทนการทำงานแบบ ติด-ดับ-ติด-ดับ ในเครื่องปรับอากาศแบบเก่าทำให้ระบบอินเวอร์เตอร์สามารถควบคุมอุณหภูมิได้อย่างแม่นยำมากขึ้นและที่สำคัญ คือ ประหยัดพลังงาน กำลังงานที่ใช้ในการทำความร้อนหรือทำความเย็นจะถูกเปลี่ยนโดยอัตโนมัติอุณหภูมิของห้องคงที่กว่าเมื่อเทียบกับระบบเก่าเนื่องจากระบบนี้จะไม่มีการหยุดการทำงานของคอมเพรสเซอร์ อุณหภูมิของห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่อนข้างคงที่ เนื่องจากระบบนี้จะปรับกำลังในการทำความร้อนหรือทำความเย็นโดยอัตโนมัติ อ้างอิงกับภาวะ(Workload) ที่มีอยู่ในห้องให้กำลังที่สูงกว่าการทำงานในช่วงเริ่มต้น ทำให้ห้องเย็นหรืออุ่นได้เร็วขึ้นผลิตภัณฑ์ที่มีประสิทธิภาพพลังงานมากอาจมีต้นทุนที่สูงในตอนแรก (ด้านราคา) แต่ผลตอบแทนคือค่าไฟฟ้าที่ได้รับจะถูกกว่า

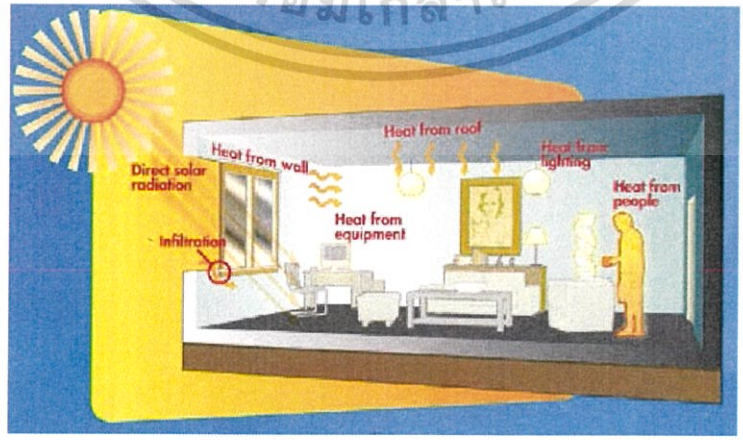
เราอาจสรุปได้ว่าผลิตภัณฑ์ที่มีประสิทธิภาพพลังงานสูงกว่าจะช่วยประหยัดเงินและพลังงานมากกว่าในระยะยาว

2.7 การประยุกต์ใช้งานเครื่องปรับอากาศ

2.7.1 ภาระในระบบปรับอากาศ

องค์ประกอบของความร้อนที่ห้องได้รับนั้นสามารถแบ่งออกได้เป็น 2 ส่วนคือ ความร้อนจากภายนอกอาคารและความร้อนจากภายในอาคารโดย

- ความร้อนจากภายนอกประกอบด้วย
  1. การนำความร้อนผ่านจากผนังหลังคาและกระจกด้านนอก
  2. การนำความร้อนผ่านผนังเบาเพดานและพื้นด้านใน
  3. การแผ่รังสีความร้อนจากดวงอาทิตย์ผ่านกระจก
- ความร้อนจากภายในประกอบด้วย
  1. ความร้อนจากแสงสว่าง
  2. คน และสัตว์
  3. อุปกรณ์ต่างๆ เช่น อุปกรณ์ไฟฟ้ามอเตอร์ของเครื่องเป่าลมเป็นต้น
  4. การรั่วของอากาศ
  5. ที่อุณหภูมิสูงเข้าสู่ห้องหรืออาคารที่ปรับอากาศอาจจะจัดแยกอยู่อีกประเภท



รูปที่ 2.33 ภาระในระบบปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาถึงแหล่งที่มาของความร้อนของระบบปรับอากาศ จะพบว่าประมาณ 60% เป็นผลจากความร้อนที่ถ่ายเทจากภายนอกอาคาร ผ่านผนังทึบและผนังโปร่งแสงเข้าสู่ภายในตัวอาคาร ส่วนที่เหลืออีก 40% เป็นภาระความร้อนที่เกิดขึ้นจากภายในตัวอาคารเอง เช่น ความร้อนจากหลอดไฟฟ้า อุปกรณ์ สำนักงาน และความร้อนจากผู้อยู่ในอาคาร เพื่อให้ผู้ที่อยู่ในห้องปรับอากาศ มีความรู้สึกสบายดังนั้น ในการออกแบบระบบปรับอากาศจะต้องคำนึงถึงตัวประกอบดังนี้

- อุณหภูมิของอากาศ
- ความชื้นของอากาศ
- ความเร็วของอากาศ
- คุณภาพของอากาศ
- ปริมาณของอากาศหมุนเวียน
- การควบคุมเสียง

2.7.2 การเลือกระบบปรับอากาศ

การเลือกประเภทของระบบปรับอากาศที่จะติดตั้งขึ้นอยู่กับความต้องการและรูปแบบการใช้งานของอาคาร โดยทั่วไปประเภทของระบบปรับอากาศที่สามารถเลือกใช้ได้มีดังนี้

2.7.2.1 ระบบปรับอากาศแบบแยกส่วน (Split Type)

เป็นระบบปรับอากาศที่ติดตั้งใช้งานง่ายมีความยืดหยุ่นในการใช้งานสูงแต่ประสิทธิภาพต่ำกว่า เหมาะสำหรับอาคารที่แบ่งเป็นพื้นที่ขนาดเล็กหลายๆ ส่วนเช่นอาคารชุดพักอาศัยในบางอาคาร อาจจะติดตั้งเครื่องปรับอากาศประเภทนี้เป็นบางห้องเพื่อที่ว่าห้องนั้นอาจจะมีคนมาใช้นอกเวลา โดยที่ไม่ต้องขึ้นกับเครื่องปรับอากาศชนิดทำน้ำเย็นที่ส่งน้ำเย็นมายังห้องต่างๆเมื่อเครื่องทำน้ำเย็นหยุดทำงานก็ยังสามารถใช้เครื่องปรับอากาศแบบแยกส่วนได้

2.7.2.2 ระบบปรับอากาศแบบชุดหรือแพ็คเกจ (Package)

เป็นระบบที่ติดตั้งง่าย แต่สำหรับเครื่องที่มีขนาดใหญ่ อาจจำเป็นต้องมีห้องเครื่องและระบบ ส่งจ่ายลมเย็น โดยทั่วไปมีประสิทธิภาพสูงกว่าระบบปรับอากาศแบบแยกส่วน เหมาะ สำหรับอาคารที่แบ่งพื้นที่เป็นชั้นและต้องการเปิด - ปิดใช้งานอย่างอิสระ

2.7.2.3 ระบบปรับอากาศแบบใช้เครื่องทำน้ำเย็น (Chiller)

เป็นระบบปรับอากาศขนาดใหญ่เหมาะสำหรับ อาคารที่ต้องการปรับอากาศทั้งอาคารมี ความยุ่งยากซับซ้อนในการออกแบบและติดตั้งมากกว่าระบบอื่นทำให้มีความจำเป็นต้องมี การออกแบบทางวิศวกรรม โดยมีส่วนประกอบคือเครื่องทำน้ำเย็น ระบบระบายความร้อน ระบบท่อและอื่นๆ เป็นระบบปรับอากาศที่มีประสิทธิภาพสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ตารางแสดงการสรุปการใช้งานของเครื่องปรับอากาศในลักษณะต่างๆ

ลักษณะเครื่องปรับอากาศ	ขนาดทำความเย็น (ตัน)	ประสิทธิภาพโดยเฉลี่ย (กิโลวัตต์ต่อตัน)	ลักษณะการใช้งาน
แบบหน้าต่าง (Window Type)	0.5-3.0	1.3-1.5	บ้านพักอาศัย สำนักงาน
แบบแยกส่วน (Split Type)	0.75-3.0	1.3-1.5	บ้านพักอาศัย สำนักงาน
แบบแพ็คเกจระบายความร้อนด้วย อากาศ (Packaged Air-Cooled Air conditioner)	3.0-30.0	1.3-1.5	คอนโดมิเนียม สำนักงาน
แบบแพ็คเกจระบายความร้อนด้วยน้ำ (Packaged Water-Cooled Air conditioner)	1.0-50.0	1.2	คอนโดมิเนียม สำนักงาน
เครื่องทำน้ำเย็นระบายชนิดความร้อนด้วยอากาศ (Air-Cooled Water Chiller)	3.0-10.0 10.0-500.0	1.4-1.6 1.4-1.6 (ปริมาณการกิน ไฟทั้งระบบ)	บ้านพักอาศัย ศูนย์คอมพิวเตอร์ขนาดเล็ก ศูนย์คอมพิวเตอร์ โรงแรมขนาดกลาง ห้องส่งสถานีโทรทัศน์ โรงพยาบาลขนาดกลาง
เครื่องทำน้ำเย็นชนิดระบายความร้อนด้วยน้ำ(Water-Cooled Water Chiller)	500-10,000	0.8-1.0 (ปริมาณการกินไฟ ทั้งระบบ)	ศูนย์การค้าขนาดใหญ่ สำนักงานขนาดใหญ่ โรงแรม โรงพยาบาล ศูนย์คอมพิวเตอร์ขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 ประสิทธิภาพระบบปรับอากาศ

ประสิทธิภาพของระบบปรับอากาศสามารถคำนวณและระบุได้ 2 รูปแบบ

2.7.3.1 อัตราส่วนประสิทธิภาพพลังงาน(EnergyEfficientRatio,EER)

เช่นเดียวกับสัมประสิทธิ์ใน การทำงานเพียงแต่พลังงานความเย็นใช้มีหน่วยเป็น บีทียู / ชั่วโมง แต่พลังงานไฟฟ้าที่ใช้มีหน่วยเป็นวัตต์เพราะฉะนั้น

$$\text{ค่าอัตราส่วนประสิทธิภาพพลังงาน (EER)} = \frac{\text{อัตราการทำความเย็น (บีทียูต่อชั่วโมง)}}{\text{กำลังไฟฟ้าป้อนเข้า (วัตต์)}}$$

สำหรับค่าอัตราส่วนประสิทธิภาพพลังงานจะใช้บอกประสิทธิภาพของระบบปรับอากาศขนาดเล็กเช่นระบบปรับอากาศแบบแยกส่วนและระบบปรับอากาศแบบแพ็คเกจขนาดเล็ก

2.7.3.2 ประสิทธิภาพของเครื่องทำน้ำเย็น (ChillerPerformance,ChP)

เป็นค่าที่แสดงประสิทธิภาพการทำความเย็นคือ อัตราส่วนระหว่างพลังงานที่เครื่องสามารถทำความเย็นได้ต่อพลังงานที่ต้องใช้ (พลังงานไฟฟ้า)

TON = ความสามารถในการทำความเย็นที่ภาระเต็มพิกัด (ตันความเย็น) หาได้จาก (FxT) / 50.4

F = ปริมาณน้ำเย็นที่ไหลผ่านส่วนทำน้ำเย็น (ลิตรต่อนาที)

T = อุณหภูมิแตกต่างของน้ำเย็น ที่ไหลเข้าและไหลออกจากส่วนทำน้ำเย็น (องศาเซลเซียส)

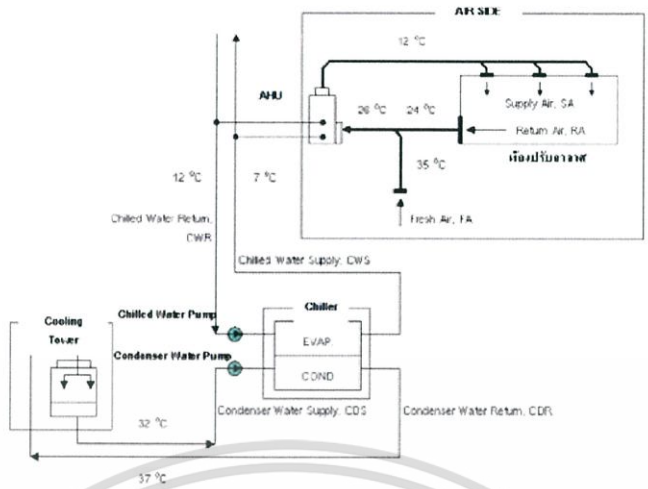
kW = กำลังไฟฟ้าที่ใช้ของส่วนทำน้ำเย็น (กิโลวัตต์)

$$\text{ประสิทธิภาพของเครื่องทำน้ำเย็น (ChP)} = \frac{\text{กำลังไฟฟ้าป้อนเข้า (กิโลวัตต์)}}{\text{อัตราการทำความเย็น (ตัน)}}$$

2.7.4 การควบคุมระบบปรับอากาศอย่างมีประสิทธิภาพ

ระบบปรับอากาศที่มีการใช้งานโดยส่วนใหญ่ในอาคารธุรกิจนั้นมีทั้งประเภทระบบปรับอากาศ แบบแยกส่วนระบบปรับอากาศแบบแพ็คเกจและระบบปรับอากาศแบบใช้เครื่องทำน้ำเย็นขึ้นอยู่กับลักษณะการใช้งาน สำหรับอาคารธุรกิจขนาดใหญ่โดยทั่วไปจะเป็นระบบปรับอากาศแบบใช้เครื่องทำ น้ำเย็นเป็นหลักดังแสดงไว้ในไดอะแกรมดังรูปที่ 2.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.34 ไดอะแกรมระบบปรับอากาศแบบใช้เครื่องทำน้ำเย็น

ส่วนประกอบหลักของระบบปรับอากาศควรคำนึงถึง 4 อย่างคือ การควบคุมระบบทำความเย็นที่เครื่องทำน้ำเย็น,การควบคุมความเร็วรอบของเครื่องสูบน้ำเย็น,การควบคุมระบบส่งจ่ายลม,การควบคุมหอระบายความร้อน

2.7.5 การควบคุมระบบทำความเย็นที่เครื่องทำน้ำเย็น

โดยปกติเครื่องทำน้ำเย็นที่ออกแบบและติดตั้งในอาคารจะถูกออกแบบให้มีขนาดทำความเย็น พิกัดมากกว่าภาระจริงเนื่องจากการเปลี่ยนแปลงของสภาพภูมิอากาศภายนอก และภาระภายในอาคาร ที่มีการเปลี่ยนแปลง ทั้งนี้เพื่อให้อุณหภูมิการปรับอากาศเป็นไปตามความต้องการตลอดเวลา ดังนั้นขนาดของเครื่องทำน้ำเย็นจะใหญ่กว่าภาระจากการคำนวณเนื่องจากการเผื่อไว้

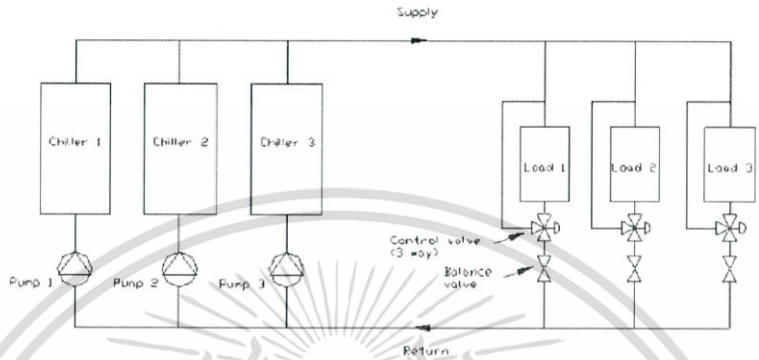
นอกจากนี้การติดตั้งเครื่องทำน้ำเย็นยังต้องมีการติดตั้งชุด Standby เพื่อใช้ในกรณีที่ภาระการปรับ อากาศสูงขึ้นและกรณีทำการบำรุงรักษาเครื่องทำน้ำเย็นโดยไม่ส่งผลกระทบต่อการปรับอากาศในอาคาร ดังนั้นการควบคุมการทำงานของเครื่องทำน้ำเย็นตามความต้องการภาระการทำความเย็นที่มีการเปลี่ยนแปลงตลอดเวลาจึงมีส่วนในเรื่องการใช้ระบบปรับอากาศอย่างมีประสิทธิภาพ ระบบควบคุมจะทำการตรวจสอบความต้องการ ของภาระทำความเย็น (Load) ณ ขณะใดขณะหนึ่ง จากนั้นจะทำการควบคุมการเดินเครื่องของเครื่องทำน้ำเย็นให้เหมาะสมกับภาระจริง เนื่องจากโดยปกติเครื่องทำน้ำเย็นทุกประเภทจะถูกออกแบบให้มี ประสิทธิภาพสูงสุดเมื่อเครื่องทำน้ำเย็นเดินที่พิกัดภาระสูงสุด (Full Load) หากภาระของเครื่องทำน้ำเย็นลดลงประสิทธิภาพจะลดลงไปด้วยหรือการใช้พลังงานจะสูงขึ้น (Part Load) ดังนั้นเครื่องทำน้ำเย็นจึง ควรเดินที่ภาระสูงที่สุดตลอดเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.6 การควบคุมความเร็วรอบของเครื่องสูบน้ำเย็น

เครื่องสูบน้ำเย็นที่ออกแบบและติดตั้งเพื่อจ่ายน้ำเย็นให้กับระบบปรับอากาศ โดยทั่วไปมีอยู่ 2 ระบบ คือ

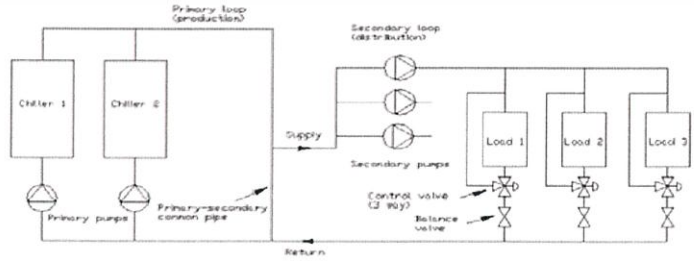
- ระบบปฐมภูมิ (Primary Chilled Water System)



รูปที่ 2.35 แสดงไดอะแกรมระบบปฐมภูมิ

เครื่องสูบน้ำเย็นในระบบปฐมภูมิ (Primary Chilled Water System) จะมีเครื่องสูบน้ำเย็น 1 ชุด ทำหน้าที่สูบน้ำเย็นผ่านเครื่องทำน้ำเย็นแต่ละเครื่องและภาระการทำความเย็น (Load) โดยปริมาณน้ำเย็นที่ไหลผ่านเครื่องทำน้ำเย็นจะคงที่ตลอดเวลา (Constant Flow) การควบคุมปริมาณน้ำเย็นที่ไหลผ่าน ภาระการทำความเย็นจะถูกควบคุมด้วยวาล์วควบคุม 3 ทาง (Three Way Valve) ในระบบ นี้เครื่องสูบน้ำเย็นจะทำงานพร้อมกับเครื่องทำน้ำเย็นซึ่งขึ้นอยู่กับภาระการทำความเย็น สำหรับการติดตั้งชุดควบคุมความเร็วรอบของเครื่องสูบน้ำเย็นจะทำได้เพียงปริมาณน้ำเย็น ส่วนเกินที่ไหลผ่านเครื่องทำน้ำเย็นเนื่องจากเครื่องทำน้ำเย็นที่ถูกออกแบบมาต้องการอัตราการไหลของน้ำเย็นคงที่

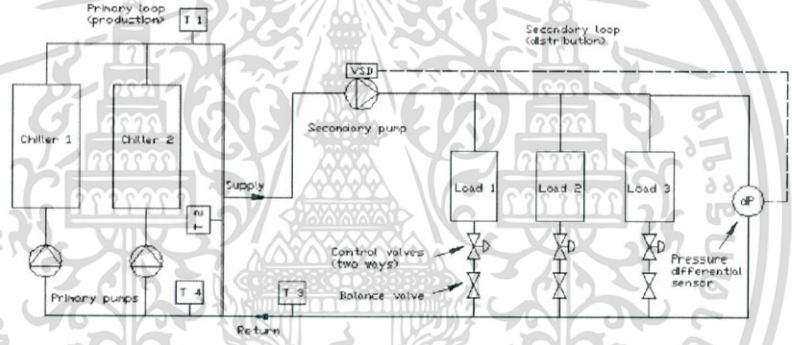
- ระบบปฐมภูมิ-ทุติยภูมิ (Primary-Secondary Chilled Water System)



รูปที่ 2.36 ไดอะแกรมระบบปฐมภูมิ-ทุติยภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

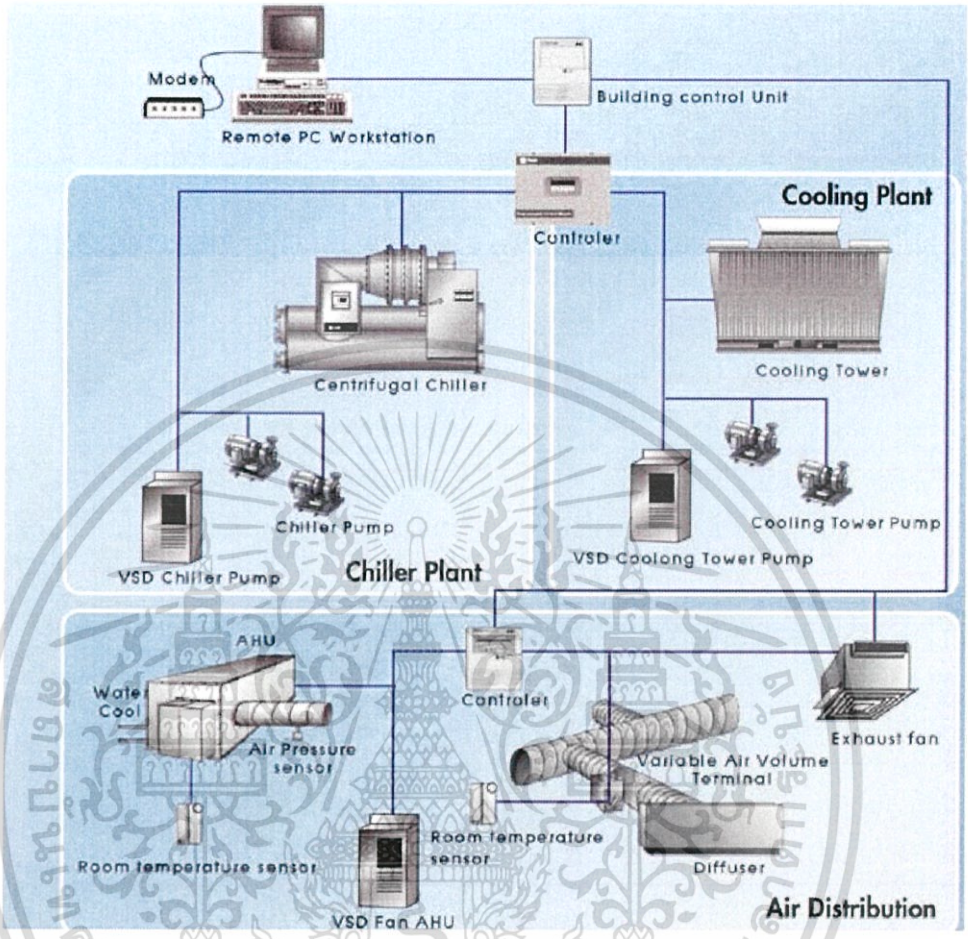
เครื่องสูบน้ำเย็นในระบบปฐมภูมิ - ทุติยภูมิ (Primary-Secondary Chilled Water System) จะมี เครื่องสูบน้ำเย็น 2 ชุด ชุดที่ 1 สำหรับ วงจรปฐมภูมิ (Primary Loop) ทำหน้าที่สูบน้ำเย็นผ่านเครื่องทำน้ำเย็นแต่ละเครื่อง โดยปริมาณน้ำเย็นที่ไหลผ่านเครื่องทำน้ำเย็นจะคงที่ตลอดเวลา (Constant Flow) และชุดที่ 2 สำหรับวงจรทุติยภูมิ (Secondary Loop) ทำหน้าที่สูบน้ำเย็นไปยังภาระการทำความเย็น (Load) โดยปริมาณน้ำเย็นที่ไหลผ่านเครื่องทำน้ำเย็นจะคงที่ตลอดเวลา (Constant Flow) การควบคุมปริมาณน้ำเย็นที่ไหลผ่านภาระการทำความเย็นจะถูกควบคุมด้วยวาล์วควบคุม 3 ทาง (Three Way Valve) สำหรับการติดตั้งชุดควบคุมความเร็วรอบของเครื่องสูบน้ำเย็นในวงจรปฐมภูมิจะทำได้เพียงปริมาณน้ำเย็นส่วนเกินที่ไหลผ่านเครื่องทำน้ำเย็น เนื่องจากเครื่องทำน้ำเย็นที่ถูกออกแบบมาต้องการอัตราการไหลของน้ำเย็นคงที่ส่วนการติดตั้งชุดควบคุมความเร็วรอบของเครื่องสูบน้ำเย็นในวงจรทุติยภูมิสามารถติดตั้งได้โดยการติตตั้งวาล์ว 2 ทางแทนวาล์ว 3 ทาง อัตราการไหลของน้ำเย็นที่ลดลงหากภาระลดลงทำให้เครื่องสูบน้ำเย็นปรับลดรอบลง



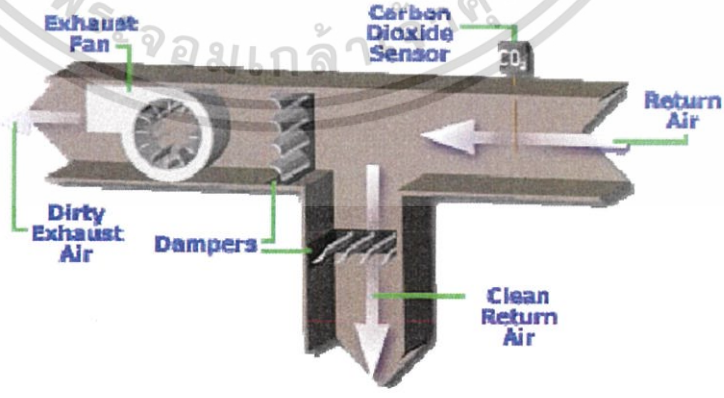
รูปที่ 2.37 ไดอะแกรมระบบปฐมภูมิ-ทุติยภูมิที่มีการติดตั้ง VSD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.7 การควบคุมระบบส่งจ่ายลมเย็น



รูปที่ 2.38 ภาพรวมระบบส่งจ่ายลม



รูปที่ 2.39 ระบบส่งจ่ายลมเย็นและการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบปรับอากาศที่ใช้งานในอาคารส่วนใหญ่มักจะเป็นระบบปรับอากาศชนิดปริมาณลมเย็นส่งออกคงที่ (Constant Air Volume System) คือปริมาณลมเย็นที่ส่งออกจากชุดส่งลมเย็น (Air - Handling Unit) เข้าสู่พื้นที่บริเวณปรับอากาศในโซนต่างๆ มีปริมาณลมเย็นส่งออกคงที่ไม่แปรเปลี่ยนตามภาระความร้อนที่เกิดขึ้นในบริเวณปรับอากาศนั้น ๆ เพียงแต่อุณหภูมิลมเย็นส่งออกแปรเปลี่ยนได้เพื่อคงสภาวอุณหภูมิห้องให้เหมาะตามที่ต้องการ ทั้งนี้โดยอาศัยเทอร์โมสแตทเป็นตัววัดและควบคุมอุณหภูมิภายในบริเวณปรับอากาศ ปัจจุบันยังมีระบบปรับอากาศอีกชนิดหนึ่งซึ่งเริ่มนิยมใช้กันมากคือระบบปรับอากาศชนิดปริมาณลมเย็นส่งออกแปรเปลี่ยน(VariableAirVolumeSystem)คือปริมาณลมเย็นที่ส่งออกจากชุดส่งลมเย็นเข้าสู่บริเวณปรับอากาศโซนต่าง ๆ สามารถแปรเปลี่ยนได้ตามภาระความร้อนที่เกิดขึ้นในบริเวณปรับอากาศโซนนั้น ๆ ส่วนอุณหภูมิภายในบริเวณปรับอากาศแต่ละโซนควบคุมให้คงที่

นอกจากการควบคุมอุณหภูมิอุณหภูมิและความชื้นภายในอาคารแล้ว ในพื้นที่ปรับอากาศที่มีคน อยู่จำนวนมากจำเป็นต้องมีการระบายอากาศออกในระดับที่เหมาะสมเพื่อรักษาระดับความเข้มข้นของ คาร์บอนไดออกไซด์ที่เกิดขึ้นจากกิจกรรมของคน ดังนั้นระบบปรับอากาศในอาคารบางครั้งต้องทำการติดตั้ง CO_2 SENSOR เพื่อใช้วัดค่าปริมาณความเข้มข้นของคาร์บอนไดออกไซด์และส่งสัญญาณกลับไปยังตัวควบคุมเพื่อปรับปริมาณการเติมอากาศบริสุทธิ์เข้าสู่ระบบปรับอากาศ

2.7.8 การควบคุมหอระบายความร้อน

การควบคุมหอระบายความร้อนสำหรับระบบปรับอากาศโดยทั่วไปจะต้องเดินจำนวนของหอระบายความร้อนให้เหมาะสมกับภาระการระบายความร้อนที่เครื่องทำน้ำเย็นโดยพิจารณาที่อุณหภูมิกระเปาะเปียกของอากาศยิ่งต่ำเท่าใดเราจะยิ่งได้น้ำหล่อเย็นที่มีอุณหภูมิต่ำยิ่งขึ้น

2.7.9 แนวทางในการประหยัดพลังงานในระบบปรับอากาศ

2.7.9.1 การออกแบบปรับปรุงอาคาร

- ปรับปรุงสภาพแวดล้อมโดยรอบ
- จัดทิศทางการวางตัวอาคารให้เหมาะสม (สำหรับอาคารใหม่)
- ใช้วัสดุก่อสร้างที่เป็นฉนวนกันความร้อน
- ป้องกันการรั่วซึมของอากาศภายนอก
- ใช้การบังเงาเพื่อหลีกเลี่ยงแสงโดยตรงจากดวงอาทิตย์

2.7.9.2 การเลือกใช้อุปกรณ์

- เลือกประเภทและขนาดของอุปกรณ์ให้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เลือกใช้อุปกรณ์ประสิทธิภาพสูง

2.7.9.3 การใช้งานและการควบคุม

- พื้นที่ปรับอากาศ

1. ตั้งค่าอุณหภูมิให้เหมาะสม ไม่เย็นจนเกินไปโดยทั่วไปอุณหภูมิใช้งานในห้องทำงานไม่ควรต่ำกว่า 24° c
2. หลีกเลี่ยงการใช้งานที่ไม่จำเป็น
3. หลีกเลี่ยงการใช้อุปกรณ์ที่มีความร้อนสูงในพื้นที่ปรับอากาศ
4. ใช้อุปกรณ์ตรวจวัดและควบคุมที่มีความเที่ยงตรงและแม่นยำเช่น เทอร์โมสตัท (Thermostat)

- เครื่องทำน้ำเย็น

1. อุณหภูมิของน้ำเย็นเข้าและออกจากเครื่องทำน้ำเย็นไม่ควรต่างกันเกิน 7°C ถ้าค่าออกแบบอยู่ที่ 5.5° C
2. ในช่วงที่ภาระการทำความเย็นลดลงควรปรับอุณหภูมิของน้ำเย็นที่ออกจากเครื่องระเหย (Evaporator) สูงขึ้น 1-2° C
3. น้ำเย็นที่ออกจากเครื่องระเหยควรมีอุณหภูมิสูงกว่าอุณหภูมิอิ่มตัวของสารทำความเย็น(Refrigerant) ไม่เกิน 1.5 ° C สำหรับเครื่องระเหยแบบ Flooded Shell and Tube และ 3 ° C สำหรับเครื่องระเหยแบบ DX Shell and Tube หากผลต่างอุณหภูมิสูงกว่าน้ำ
4. แสดงว่าเครื่องระเหยสกปรก ทำให้ประสิทธิภาพการถ่ายเทความร้อนลดลง น้ำระบายความร้อนที่ออกจากเครื่องควบแน่นควรมีอุณหภูมิสูงกว่าอุณหภูมิควบแน่น อิ่มตัวของสารทำความเย็น (Refrigerant) ไม่เกิน 3 ° C หากผลต่างอุณหภูมิสูงกว่าน้ำแสดงว่าเครื่องควบแน่นสกปรก ทำให้ประสิทธิภาพการถ่ายเทความร้อนลดลง
5. ความดันของสารทำความเย็นด้านต่ำ (เครื่องระเหย) และความเย็นด้านสูง (เครื่องควบแน่น) ไม่ควรต่ำกว่ามาตรฐาน

- เครื่องส่งลมเย็น

1. อุณหภูมิผิวท่อของแผงคอล์ยเย็นต้องต่ำกว่าอุณหภูมิของจุดน้ำค้างของอากาศที่เข้ามารับความเย็น
2. อุณหภูมิอากาศขาออกจากแผงคอล์ยเย็นต้องสูงกว่าอุณหภูมิน้ำเย็นที่ออกจากแผงคอล์ยเย็นไม่เกิน 3.3 ° C (6° C)
3. อัตราการไหลของอากาศที่ผ่านแผงคอล์ยเย็นควรอยู่ที่ 300-400 ลบ.ฟุต ต่อนาทีต่อตัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ปรับสมดุลน้ำที่วาล์วปรับสมดุลน้ำเย็น (Balancing Valve) เพื่อให้อัตราการไหลของน้ำเย็นเข้าแผงคอล์ยเย็นเหมาะสมกับขนาดของแผงคอล์ยเย็นเพื่อป้องกันอัตราการ ไหลของน้ำเย็นบางจุดเกินและบางจุดขาด เพราะจะทำให้เครื่องส่งลมเย็นไม่สามารถจ่ายลมเย็นได้อย่างเพียงพอกรณีที่อัตราการไหลน้อยเกินไป หรือบางจุดอาจเย็นเกินไป เนื่องจากอัตราการไหลเกิน

- หอระบายความร้อน

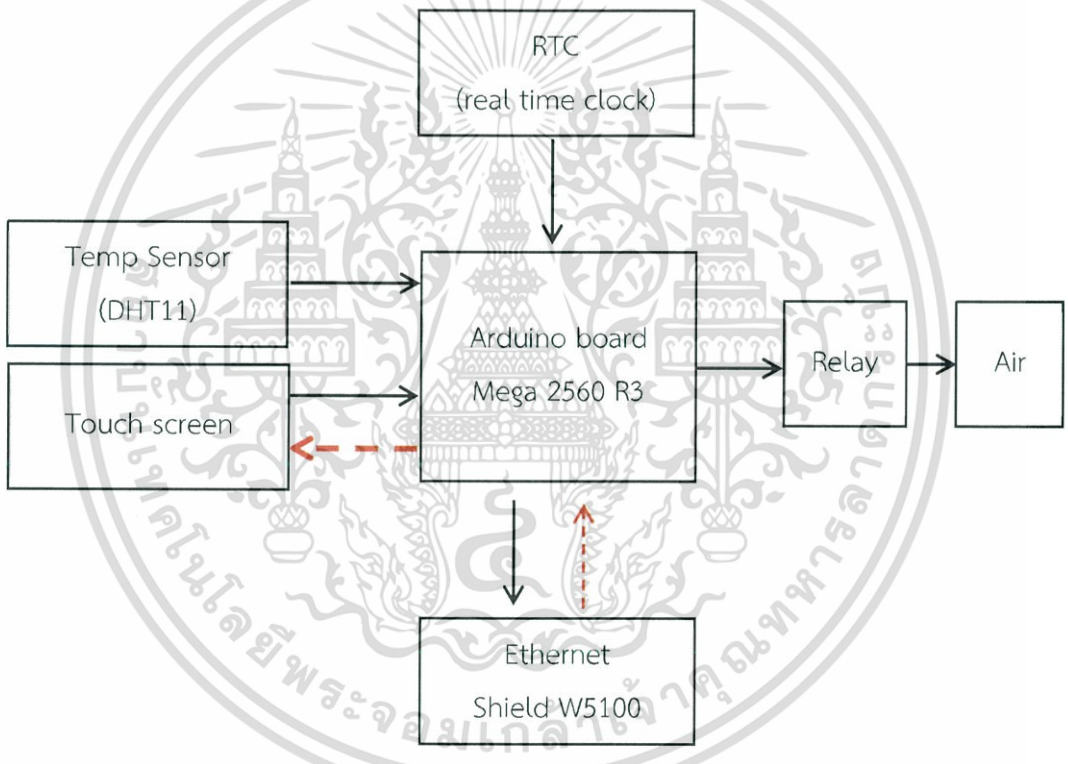
1. อุณหภูมิของน้ำที่ออกจากหอระบายความร้อนไม่ควรสูงกว่าอุณหภูมิกระเปาะเปียกของอากาศเกิน 3.3 ° C (6° C)

2. อัตราการไหลของอากาศต้องไม่น้อยกว่าพิกัดออกแบบโดยทั่วไปอยู่ที่ประมาณ 180- 250 ลบ.ฟุตต่อนาทีต่อตัน มิฉะนั้นจะระบายความร้อนให้กับน้ำได้น้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
หลักการออกแบบ

การออกแบบสำหรับควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลาที่กำหนดอย่างอัตโนมัติ โดยควบคุมด้วย Arduino ในเบื้องต้นต้องเข้าใจถึงหลักการและการทำงานภาพรวมของระบบควบคุมก่อนว่า มีหลักการทำงาน อย่างไร จากนั้นจึงค่อยพิจารณาในส่วนประกอบย่อยๆ โครงสร้างการทำงานทั้งหมดระบบ สามารถอธิบายได้ด้วยแผนภาพ ดังรูปที่ 3.1



รูปที่ 3.1 ภาพรวมการติดตั้งอุปกรณ์ของระบบควบคุมการเปิด-ปิดเครื่องปรับอากาศตามตารางเวลาที่กำหนดอย่างอัตโนมัติ

3.1 โครงสร้างของระบบควบคุมด้วย Arduino

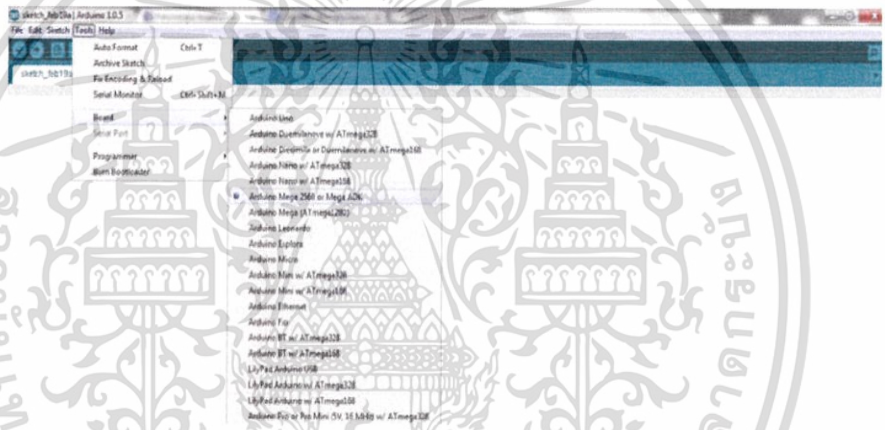
ในการออกแบบวงจรที่ใช้ในการควบคุมโครงสร้างที่เป็นวงจรและอุปกรณ์อิเล็กทรอนิกส์ ได้เลือกใช้ไมโครคอนโทรลเลอร์ Arduino เป็นตัวควบคุมอุปกรณ์ต่างๆ ภายในโครงสร้างซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถและคุณสมบัติของบอร์ด Arduino สามารถควบคุม รวมทั้ง รีเลย์, จอทัสกรีน, Ethernet shield , real time clock และ Temperature sensor

3.1.1 Arduino

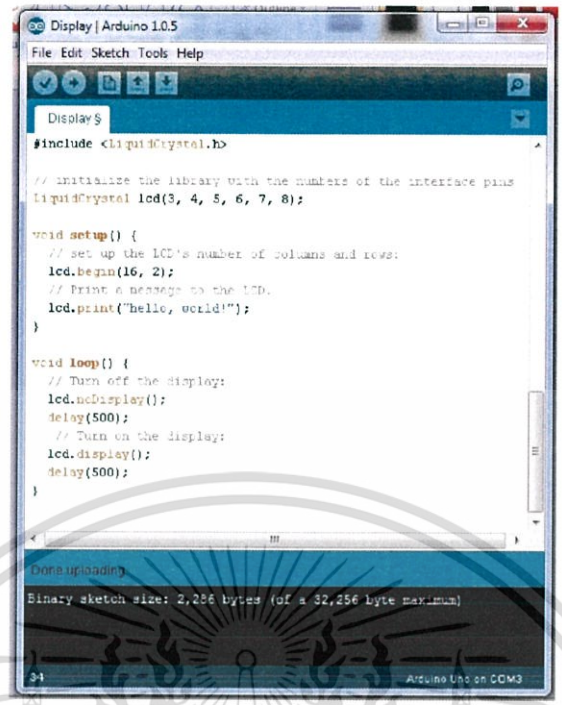
อุปกรณ์อิเล็กทรอนิกส์ต่างๆ จะมีวิธีการใช้งานหรือคุณสมบัติแตกต่างกันออกไป ตัวอย่าง เช่นการเริ่มต้นการใช้งานบอร์ด Arduino อันดับแรกคือ ทำการลงไดร์เวอร์ของ Arduino ลงในคอมพิวเตอร์เพื่อ ใช้สำหรับการเขียนโปรแกรมในการควบคุมในเวอร์ชั่น 1.0.5 และลงไดร์เวอร์ของพอร์ต USB เพื่อใช้ในการอัพโหลดข้อมูลลงบอร์ด Arduino หลังจากลงไดร์เวอร์เป็นที่เรียบร้อยแล้ว ต่อมาต้องเลือกรุ่นของ บอร์ด Arduino ซึ่งบอร์ดที่เลือกใช้นี้มีชื่อว่า Arduino Mega 2560 R3 board ดังรูปที่ 3.2



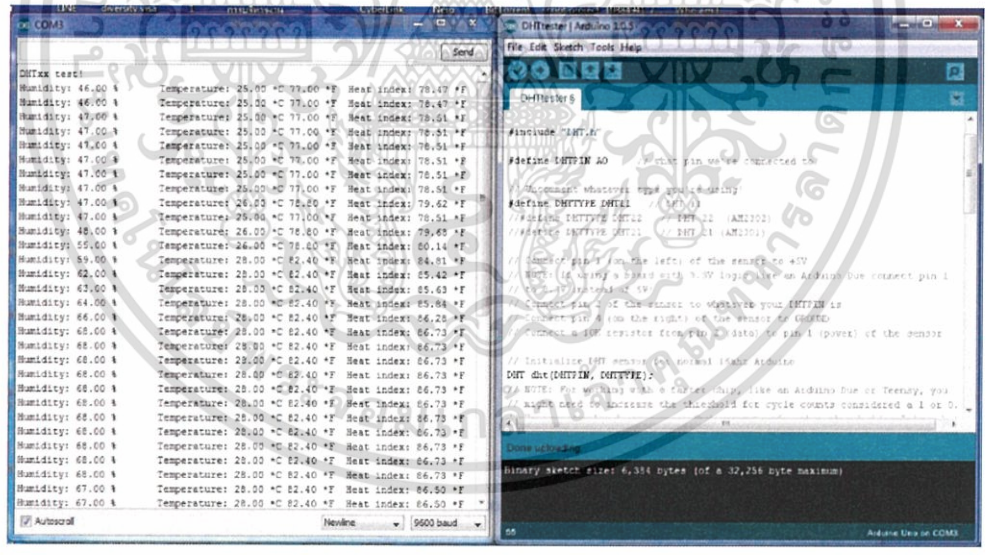
รูปที่ 3.2 การเลือกรุ่นของ บอร์ด Arduino

จากนั้นเริ่มทำการเขียนโปรแกรมเพื่อทดสอบอุปกรณ์ต่างๆ ที่ต้องการใช้บอร์ด Arduino ในการควบคุม เมื่อเขียนโปรแกรมได้ตามต้องการ จึงทำการทดสอบโดยการอัพโหลดข้อมูลที่เขียนลงใน บอร์ด Arduino เมื่ออัพโหลดเสร็จ ที่บอร์ด Arduino จะแสดงไฟติด หมายความว่า อัพโหลดเสร็จสมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ตัวอย่างการเขียนโปรแกรมในโปรแกรม Arduino



รูปที่ 3.4 ตัวอย่างการเขียนโปรแกรมในโปรแกรม Arduino (ต่อ)

3.1.2 หน้าที่ของบอร์ด Arduino

ในการควบคุมบอร์ด Arduino เพียงบอร์ดเดียวสามารถใช้ควบคุมการ เปิด - ปิดตามเวลาที่กำหนดโดยจะควบคุมรวมทั้ง Relay, Touch screen , Ethernet shield และ Temperature sensor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 การเชื่อมต่อบอร์ด Arduino กับอุปกรณ์

ตารางที่ 3.1 ตารางแสดงการใช้ช่อง Input ของ Arduino

ชนิดของอุปกรณ์	ช่อง INPUT
Real time clock	D20, D21
Touch screen display	5V, GND, D0-D7, D21-D53
Ethernet shield	5V, GND, D10-D13 (used for internet) สามารถต่อคร่อมบนบอร์ด Arduino ได้เลย
Temperature sensor	5V, GND, A0

ตารางที่ 3.2 ตารางแสดงการใช้ช่อง Output ของ Arduino

ชนิดของอุปกรณ์	ช่อง OUTPUT (Arduino)
รีเลย์	14, 15, 16, 17, 19

จากตารางที่ 3.1 อธิบายได้ดังนี้

- Real time clock เป็นตัวตั้งเวลาให้กับ Arduino board และให้ค่าเวลาตามจริง
- Touch screen display เป็นหน้าจอสัมผัสที่ทำหน้าที่

  รับคำสั่ง เปิด–ปิดเครื่องปรับอากาศ

  รับคำสั่ง เปิด-ปิดพัดลมตัวที่ 1,2 และ 3 ของเครื่องปรับอากาศ

  รับคำสั่ง เพิ่ม-ลด Set point value ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Ethernet shield ทำหน้าที่ให้บอร์ด arduino เชื่อมต่อกับอินเตอร์เน็ต เพื่อให้บอร์ด arduino สามารถควบคุมการเปิด - ปิดเครื่องปรับอากาศผ่านระบบอินเตอร์เน็ตได้
- Temperature sensor ทำหน้าที่วัดอุณหภูมิและความชื้นของห้อง ณ ขณะนั้น

จากตารางที่ 3.2 อธิบายได้ดังนี้

- รีเลย์ ทำหน้าที่ควบคุมการเปิด-ปิดเครื่องปรับอากาศ

ตารางที่ 3.3 ตารางแสดงรายละเอียดในการต่ออุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศ เข้ากับบอร์ด Arduino (MEGA 2560)

อุปกรณ์ควบคุมการเปิดปิดเครื่องปรับอากาศ	บอร์ด Arduino (MEGA 2560)
1. Temperature Sensor (DHT11) VCC GND Data	 ไฟ 5 V GND A0 (Analog 0)
2. Real Time Clock (DS1307) SCL SDA GND Vcc	 D21 D20 GND 5V
3. Ethernet Shield W5100 R3	สามารถต่อคร่อมกับบอร์ด Arduino ได้โดยตรง โดยให้ตำแหน่งของ pin ที่บอร์ด Arduino กับ Ethernet Shield ตรงกัน pin ที่จำเป็นสำหรับใช้กับ internet คือ digital pin 10, 11, 12 และ 13 * ไม่ควรใช้ digital pin ดังกล่าวเป็น I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ควบคุมการเปิดปิดเครื่องปรับอากาศ	บอร์ด Arduino (MEGA 2560)
4. Touch screen (3.2 inches TFT LCD module Display with touch panel SD Card) หน้าจอ Touch screen ไม่สามารถรับไฟ 5V จากบอร์ด Arduino ได้โดยตรง จึงต้องมีการต่อ Shield เพิ่มขึ้นมา เพื่อทำการแปลงไฟจาก 5V เป็น 3V สำหรับการใช้งานของหน้าจอโดยขา pin ที่ออกจากบอร์ด shield ของ TFT สามารถต่อเข้ากับบอร์ด Arduino ได้อย่างพอดี โดย pin ที่จำเป็น คือ pin ที่ใช้รับไฟ 5V จาก Arduino	
Pin รับไฟ 5V	5V
GND	GND
	จากบอร์ด TFT LCD Mega Shield V2.2 Adapter Module ต่อเข้ากับขา D0-D7 และ D21-D53 ของ Arduino
5. Relay (8 Channel Relay)	
Vcc	5V
GND	GND
IN1	D14
IN2	D15
IN3	D16
IN4	D17
IN5	D19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 อุปกรณ์ที่ใช้ในการควบคุมการเปิด-ปิดเครื่องปรับอากาศ

3.2.1 Temperature Sensor (DHT11)

ออกแบบให้ใช้ Temperature Sensor (DHT11) เพื่อวัดอุณหภูมิห้อง ห้องที่ต้องการติดตั้งเครื่องปรับอากาศ เพื่อนำค่าอุณหภูมิที่วัดได้นั้นมากำหนดการเปิด-ปิดพัดลมในเครื่องปรับอากาศ ซึ่งเครื่องปรับอากาศนั้นมีพัดลม 3 ตัว ทำงานในระดับที่ต่างกัน ระดับที่ต่างกันนี้มีค่าอุณหภูมิ (T) ที่วัดได้เป็นตัวกำหนด ดังตารางที่ 3.4

ตารางที่ 3.4 ตารางแสดงการทำงานของพัดลมตามเงื่อนไขอุณหภูมิ

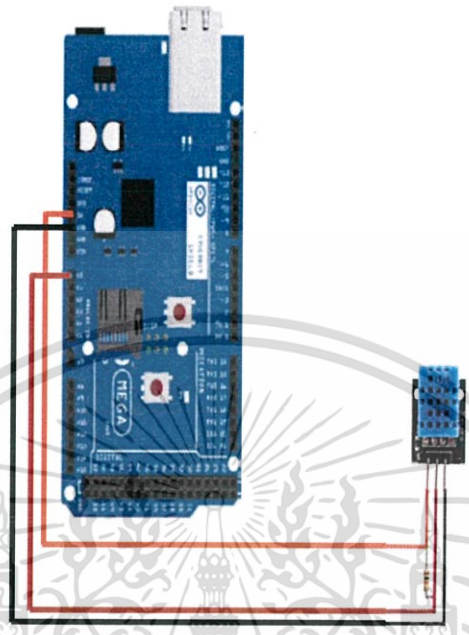
พัดลม	อุณหภูมิ (T) °C
ตัวที่ 1 ทำงาน	T <= Set point
ตัวที่ 2 ทำงาน	T >= Set point อยู่ 1 °C และ T<= Set point อยู่ 2 °C
ตัวที่ 3 ทำงาน	T > Set point อยู่ 2 °C

รูปที่ 3.5 อุปกรณ์ DHT11 (Humidity and Temperature Sensor)

ใช้ DHT11 (Humidity and Temperature Sensor) ต่อร่วมกับบอร์ด Arduino เป็น Module ที่ใช้ในการวัดอุณหภูมิกับความชื้นในอากาศ เนื่องจากวัดอุณหภูมิในย่าน 0 - 50 องศาเซลเซียส มีความละเอียดในการวัด 1 องศาเซลเซียส เพียงพอกับความต้องการที่จะใช้วัดอุณหภูมิที่อุณหภูมิห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การต่ออุปกรณ์ DHT11 กับ Arduino Mega 2560 R3 Board ต้องใช้ตัวต้านทาน 10 kΩ ต่อดังรูปที่ 3.6



รูปที่ 3.6 การต่อ DHT11 กับ Arduino Mega 2560 R3 Board

3.2.2 Real Time Clock (DS1307)

Real Time Clock (RTC) ก็คือ อุปกรณ์ที่ให้ค่าเวลาตามจริง โดยมีหลักทำงานโดยการจับสัญญาณนาฬิกาที่ได้มาจาก Crystal ซึ่งบางรุ่นจะมีถ่านสำรองมาเพื่อทำหน้าที่ ในการบันทึกเวลาอย่างต่อเนื่องถึงแม้ว่าจะไม่มีไฟเลี้ยงมาที่ตัวบอร์ดทำให้เวลาก็ยังคงนับได้ต่อได้ ทำให้ไม่ต้องเสียเวลามาตั้งเวลาใหม่หลังจากที่หยุดจ่ายไฟเลี้ยง

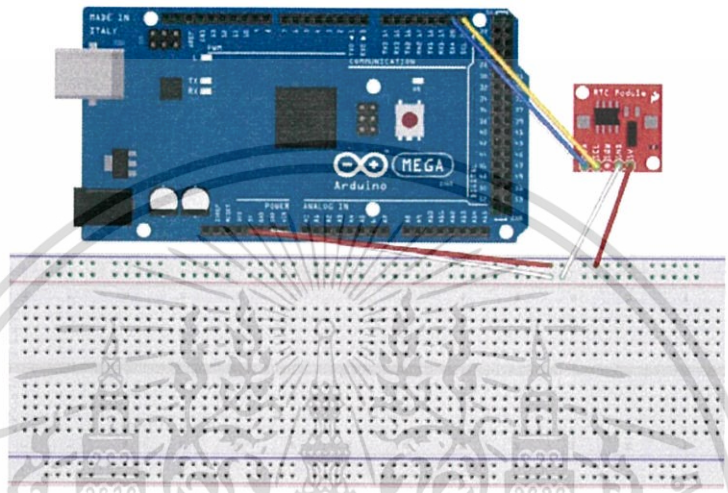
ถึงแม้ว่า Arduino Board มีตัวจับเวลาอยู่แล้ว แต่คำสั่งจาก Sketch ที่เขียนนั้น ทำให้เกิดผลเสียต่อการจับเวลา (จับเวลาได้ไม่ต่อเนื่อง) เพราะถูกแทรกแซง (Interrupt) จากคำสั่งอื่นๆ เวลาที่ได้จากการใช้คำสั่งนี้ จึงไม่สามารถนำมาเป็นเวลาตามจริงที่ต้องการบันทึกไปพร้อมกับค่าอื่นๆ ที่ต้องการวัดได้

รูปที่ 3.7 Real Time Clock (DS1307)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกใช้ RTC ในรุ่น Tiny RTC I2C 24C32 DS1307 เพราะเป็นนาฬิกาแบบเรียวไทม์ สามารถเก็บข้อมูลได้ทั้งวัน เวลา พร้อมแสดงผลเวลาได้ทั้งแบบ 12 และ แบบ 24 ชั่วโมง รวมทั้งมีไลบารีมาตรฐานพร้อมใช้งาน

ทำการต่ออุปกรณ์ RTC กับ Arduino Mega 2560 R3 Board ดังรูปที่ 3.8



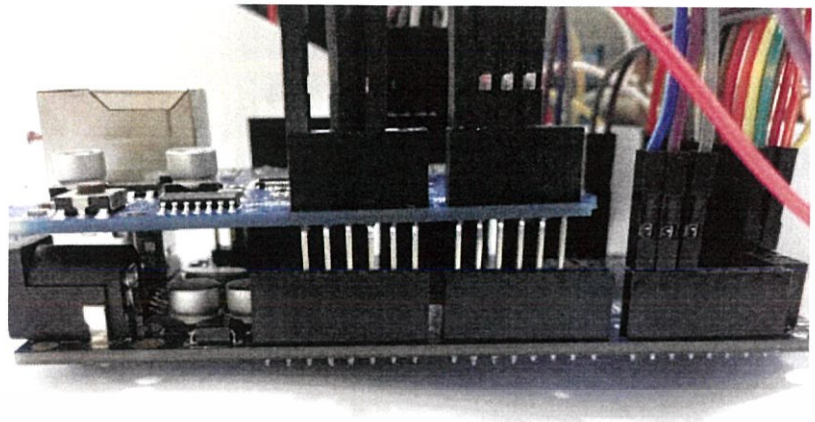
รูปที่ 3.8 การต่อ RTC กับ Arduino Mega 2560 R3 Board

3.2.3 Ethernet Shield (W5100 R3)

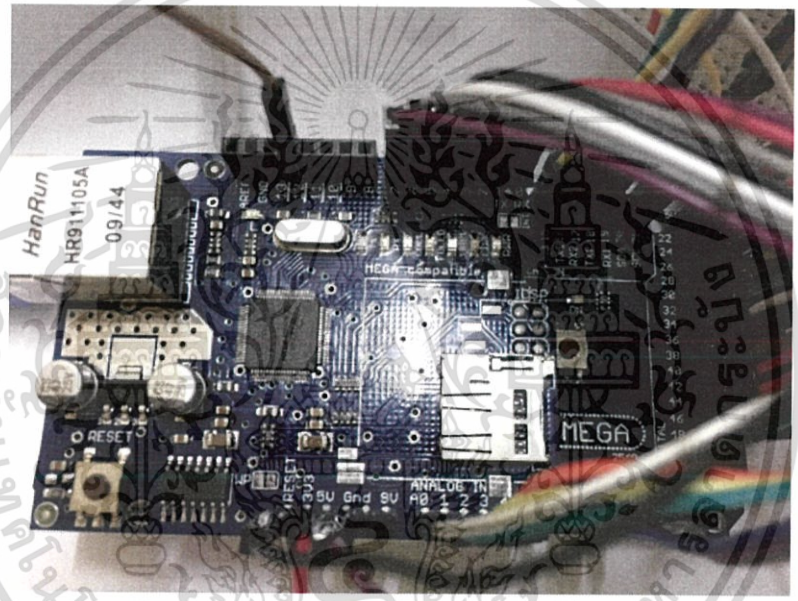
Ethernet Shield คือ อุปกรณ์ที่สามารถติดต่อกับเครือข่าย หรือเชื่อมต่อกับอินเตอร์เน็ตได้ เพื่อควบคุมการเปิด-ปิดเครื่องปรับอากาศผ่านเว็บบราวเซอร์ และแสดงค่าต่างๆ ดังรูปที่ 3.9 โดยทำการเชื่อมต่อกับ Arduino Mega 2560 R3 board ดังรูปที่ 3.10 และ 3.11

รูปที่ 3.9 Ethernet Shield W5100 R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 การต่อ Ethernet Shield กับ Arduino Mega 2560 R3 board



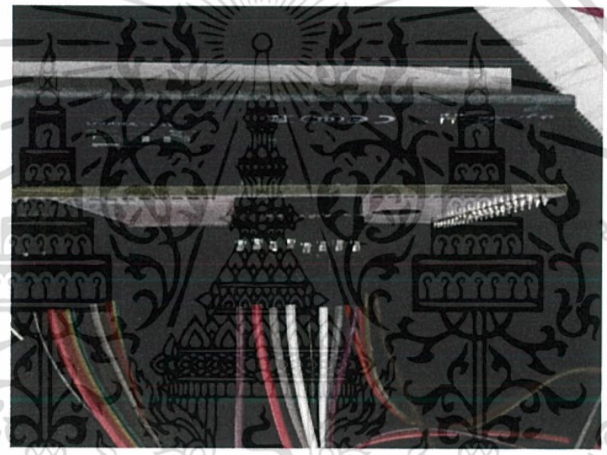
รูปที่ 3.11 การต่อ Ethernet Shield กับ Arduino Mega 2560 R3 board (ต่อ)

3.2.4 Touch screen (3.2 inches TFT LCD module Display with touch panel SD Card)

ออกแบบให้ใช้หน้าจอทัชสกรีนที่มีขนาด 3.2 นิ้ว ประเภท Resistive touch screen ดังรูปที่ 3.12 เพื่อเป็นหน้าจอควบคุมการเปิดปิดเครื่องปรับอากาศแบบแมนนวล ภายในห้องที่ติดตั้งเครื่องปรับอากาศ โดยทำการต่อกับ Arduino Mega 2560 R3 board ดังรูปที่ 3.13

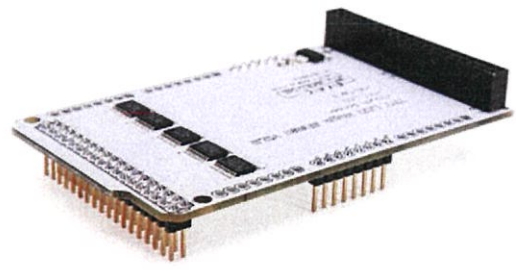
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 3.2 inch TFT LCD module Display with touch panel SD Card



รูปที่ 3.13 แสดงการต่อ Touch screen กับ Arduino Mega 2560 R3 board

เนื่องจากจอทัชสกรีนทั่วไปนั้น มีแรงดันที่ระดับ 3.3 V ซึ่งน้อยกว่าแรงดันทำงานของ Arduino Mega 2560 R3 ที่มีแรงดันทำงานเท่ากับ 5 V จึงต้องใช้ TFT LCD Mega Shield V2.2 Adapter Module ดังรูปที่ 3.14 มาต่อเพิ่ม

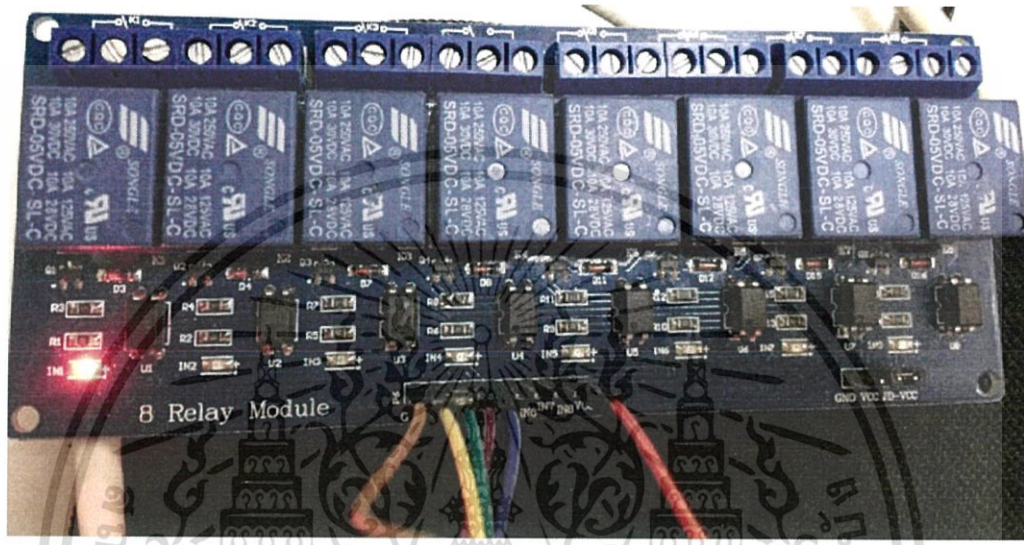


รูปที่ 3.14 TFT LCD Mega Shield V2.2 Adapter Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Relay (8 Channel Relay)

เนื่องจากรีเลย์เป็นตัวควบคุมอุปกรณ์ไฟฟ้า จึงออกแบบให้ใช้รีเลย์เพื่อควบคุมการเปิดปิดเครื่องปรับอากาศตามคำสั่งของ Arduino Mega 2560 R3 board โดยที่รีเลย์มีการเชื่อมต่อกับ Arduino Mega 2560 R3 board ดังรูปที่ 3.15



รูปที่ 3.15 การต่อรีเลย์กับ Arduino Mega 2560 R3 board

ออกแบบใช้รีเลย์แบบ NO (Normally Open) คือ ที่สภาวะปกติหน้าสัมผัสกับขั้วแยกจากกัน ถ้าไม่มีกระแสไฟฟ้าเครื่องปรับอากาศไม่เปิด แต่หากต้องการให้เครื่องปรับอากาศเปิดต้องป้อนกระแสไฟฟ้าเข้ามา ซึ่ง Arduino Mega 2560 R3 board ทำหน้าที่ในการป้อนกระแสไฟฟ้าตามโปรแกรมที่เขียนไว้

จากการออกแบบให้ใช้อุปกรณ์ทั้งหมดนี้ ทำให้เราได้กล่องอุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศ ซึ่งอธิบายได้ดังนี้

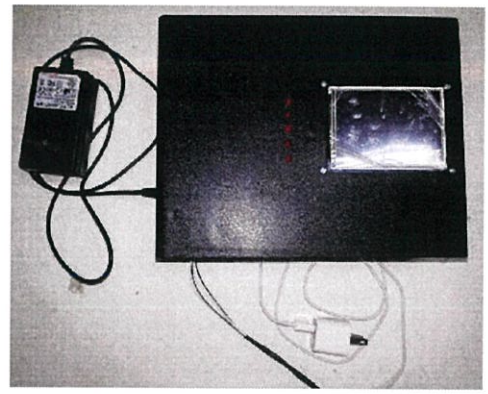
ด้านหน้าของกล่องแสดงหน้าจอทัชสกรีนรับคำสั่งจากผู้ใช้งาน

ด้านบนของกล่องเป็นรีเลย์ 8 ช่องที่ต่อกับเครื่องปรับอากาศ

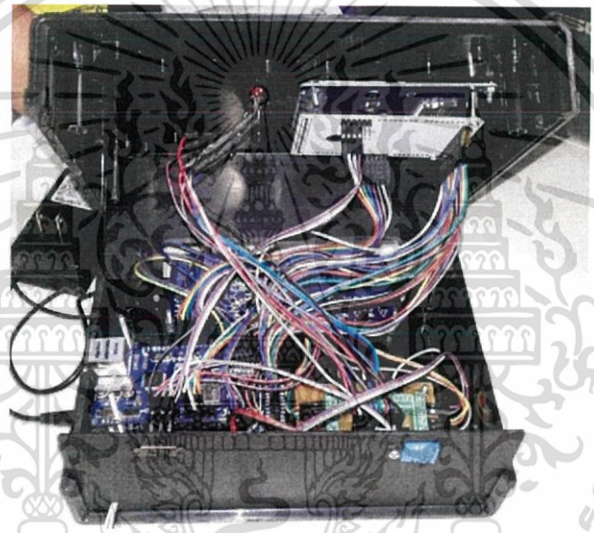
ด้านล่างของกล่องมีสวิตช์สับเปลี่ยนระหว่างระบบอินเตอร์เน็ตกับระบบหน้าจอทัชสกรีน อุปกรณ์ตรวจรู้อุณหภูมิ (Temperature) เพื่อวัดค่าอุณหภูมิห้อง

ด้านซ้ายของกล่องเป็นสายไฟต่อไฟเลี้ยง สายเชื่อมต่ออินเตอร์เน็ตเพื่อเข้าบอร์ด Arduino Mega 2560
ดังรูปต่อไปนี้

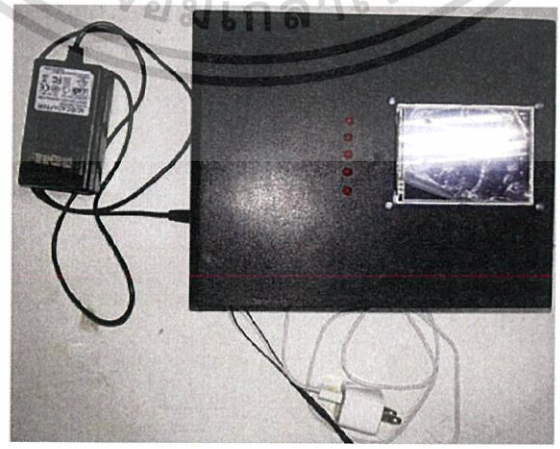
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 กล่องอุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศ

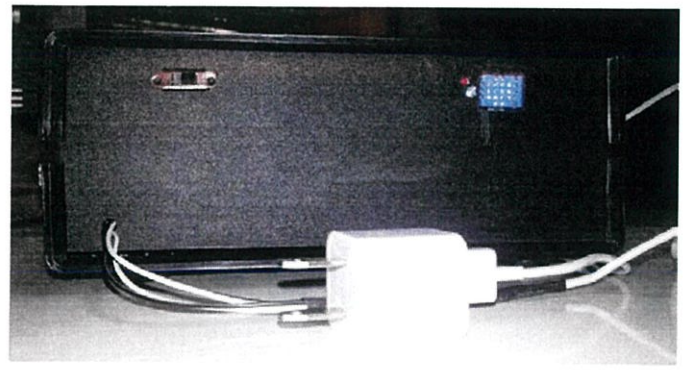


รูปที่ 3.17 อุปกรณ์ฮาร์ดแวร์ภายในกล่อง

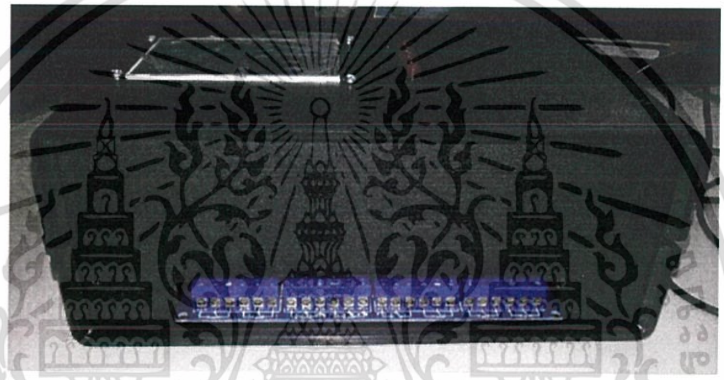


รูปที่ 3.18 ภาพด้านหน้าของกล่อง

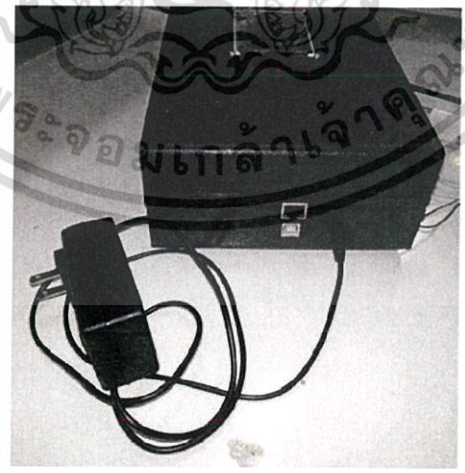
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.19 ภาพด้านล่างของกล่อง



รูปที่ 3.20 ภาพด้านบนของกล่อง



รูปที่ 3.21 ภาพด้านซ้ายของกล่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 โครงสร้างของระบบควบคุมเครื่องปรับอากาศ

3.3.1 ระบบควบคุมเครื่องปรับอากาศ

การควบคุมเครื่องปรับอากาศนั้น เปรียบเสมือนการที่ผู้ใช้ควบคุมเครื่องปรับอากาศโดยใช้รีโมทคอนโทรล ที่มีการทำงานแบบอัตโนมัติ (Auto) และ แบบปรับค่าอุณหภูมิเอง (Manual)

3.3.1.1 การทำงานของเครื่องปรับอากาศ

เครื่องปรับอากาศมี 2 โหมดการทำงาน คือ 1. ระบบคุมแบบอัตโนมัติ 2. ระบบควบคุมด้วยมือ ซึ่งกำหนดให้เครื่องปรับอากาศมีพัดลม 3 ตัว

3.3.1.1.1 ระบบควบคุมแบบอัตโนมัติ

เมื่อสั่งเปิดเครื่องปรับอากาศ เครื่องปรับอากาศเริ่มทำงานในระบบควบคุมแบบอัตโนมัติมีหลักการทำงานดังนี้

เมื่อเปิดเครื่องปรับอากาศพัดลมตัวที่ 3 ทำงานโดยอัตโนมัติ แต่

- ถ้าอุณหภูมิห้องสูงกว่าค่า Set point ที่กำหนดไว้ 2 °C ขึ้นไป ให้พัดลมตัวที่ 3 ทำงาน
- ถ้าอุณหภูมิห้องสูงกว่าค่า Set point ที่กำหนดไว้ 2 °C หรือ 1 °C ให้พัดลมตัวที่ 2 ทำงาน
- ถ้าอุณหภูมิห้องต่ำกว่าหรือเท่ากับค่า Set point ที่กำหนดไว้ให้พัดลมตัวที่ 1 ทำงาน

โดยที่ค่า Set point คือ ค่าอุณหภูมิที่ผู้ใช้งานต้องการ ซึ่งสามารถเปลี่ยนค่านี้ได้ เช่น ถ้าต้องการให้อากาศภายในห้องมีอุณหภูมิ 25 °C ต้องปรับค่า Set point ให้มีค่า 25 °C

3.3.1.1.2 ระบบควบคุมด้วยมือ

โหมดนี้ทำงานก็ต่อเมื่อผู้ใช้งานกดคำสั่งระบบควบคุมด้วยมือจากหน้าจอทัชสกรีนที่ติดอยู่กับเครื่องปรับอากาศ และเว็บบราวเซอร์ โดยที่ผู้ใช้งานเป็นผู้สั่งให้พัดลมตัวที่ 1, 2 หรือ 3 ทำงานเอง ในโหมดนี้ผู้ใช้ต้องเป็นผู้กำหนดค่า Set point เช่นเดียวกับระบบควบคุมแบบอัตโนมัติ

3.3.1.2 การควบคุมการเปิด-ปิดเครื่องปรับอากาศ

ออกแบบให้มีการควบคุมการเปิด - ปิดเครื่องปรับอากาศจาก 3 การควบคุม คือ 1. ควบคุมการเปิดปิดจากการตั้งเวลา 2. ควบคุมการเปิดปิดผ่านระบบอินเตอร์เน็ต 3. ควบคุมการเปิดปิดผ่านหน้าจอทัชสกรีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2.1 ควบคุมการเปิด – ปิดจากการตั้งเวลา

การควบคุมการเปิดปิดเครื่องปรับอากาศจากการตั้งเวลานั้น ผู้ใช้งานสามารถตั้งเวลาเปิด เวลาปิด ผ่านเว็บบราวเซอร์ซึ่งใน IP address ขึ้นอยู่กับผู้ใช้งานแต่ในที่นี้ตั้งให้ IP address คือ 192.168.1.64 ดังรูปที่ 3.17 โดยที่การตั้งค่าเวลานั้น สามารถตั้งค่าได้ในหน่วยชั่วโมง และนาที ตามความต้องการของผู้ใช้งาน เช่น ถ้าต้องการให้เครื่องปรับอากาศเปิดในเวลา 10.30 น. และต้องการให้ปิดในเวลา 21.30 น. กดปุ่มบวกที่ Hour เป็นเลข 10, Minute เป็น 30 ในส่วนของ Time to open และกดปุ่มบวกที่ Hour เป็นเลข 21, Minute เป็น 30 ในส่วนของ Time to close ดังรูปที่ 3.16

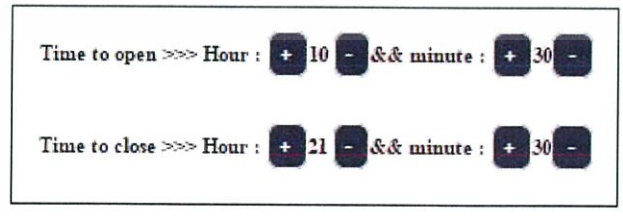
ถ้าไม่มีการเปลี่ยนแปลงเวลาเปิด-ปิดเครื่องปรับอากาศที่หน้าเว็บ เครื่องปรับอากาศจะทำการเปิดปิดตามเวลาเดิมที่ตั้งไว้ทุกวัน โดยอัตโนมัติหากผู้ใช้ต้องการเปลี่ยนแปลงเวลาการเปิดปิดเครื่องปรับอากาศสามารถเข้าเว็บบราวเซอร์นี้ได้ จากภายในวง LAN เดียวกัน

3.3.1.2.2 ควบคุมการเปิดปิดผ่านระบบอินเตอร์เน็ต

การควบคุมการเปิดปิดเครื่องปรับอากาศผ่านระบบอินเตอร์เน็ตนั้น ผู้ใช้งานสามารถเข้าเว็บแอดเดรส 192.168.164 จากเครือข่ายอินเตอร์เน็ตที่อยู่ในวง LAN เดียวกัน เพื่อทำการควบคุมการเปิดปิด โดยที่ผู้ใช้งานสามารถควบคุมผ่านหน้าเว็บได้ มีดังนี้

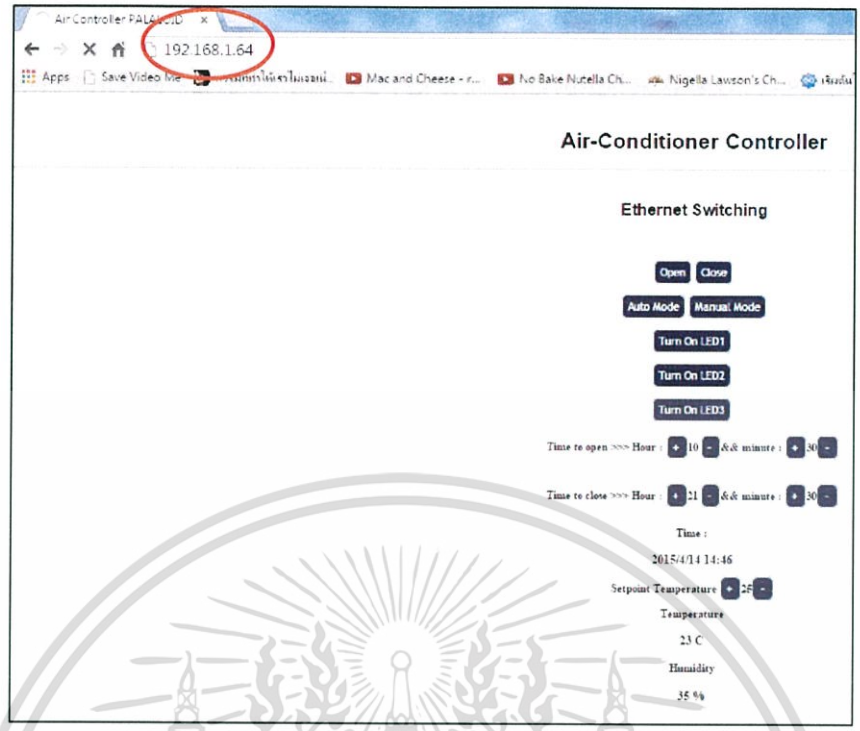
1. สั่งเปิด-ปิดเครื่องปรับอากาศ
2. เลือกโหมดการทำงานของเครื่องปรับอากาศ โหมดอัตโนมัติ และโหมดแมนนวล
3. สั่งเปิดปิดพัดลมตัวที่ 1, 2 และ 3
4. ตั้งค่าเวลาเปิดปิดเครื่องปรับอากาศ
5. ตั้งค่า Set point Temperature ได้

นอกจากนี้หน้าเว็บบราวเซอร์สามารถแสดงค่าอุณหภูมิห้อง ค่าความชื้นที่ อุปกรณ์ตรวจวัดอุณหภูมิ (Temperature Sensor) อ่านได้ และแสดงวัน เวลาแบบเรียลไทม์ ดังรูปที่ 3.24

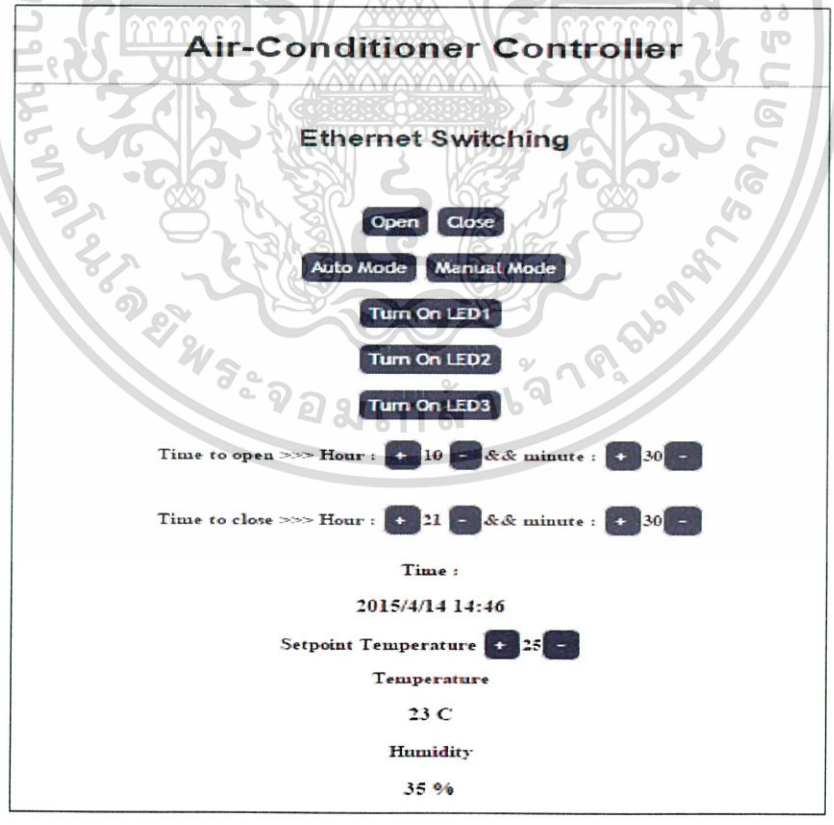


รูปที่ 3.22 ตั้งค่าเวลาเปิดปิดเครื่องปรับอากาศผ่านหน้าเว็บบราวเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 หน้าเว็บบราวเซอร์



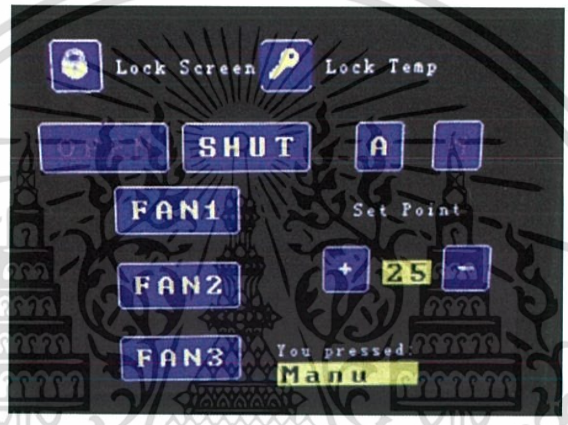
รูปที่ 3.24 หน้าเว็บบราวเซอร์ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2.3 ควบคุมการเปิดปิดผ่านหน้าจอทัชสกรีน

การควบคุมการเปิดปิดเครื่องปรับอากาศผ่านหน้าจอทัชสกรีนนั้น ผู้ใช้งานสามารถควบคุมการเปิดปิดเครื่องปรับอากาศผ่านหน้าจอทัชสกรีนที่อยู่ในห้องควบคุม โดยที่ผู้ใช้งานสามารถควบคุมผ่านหน้าจอทัชสกรีนได้ มีดังนี้

1. สั่งเปิด-ปิดเครื่องปรับอากาศ
2. สั่งเปิดปิดพัดลมตัวที่ 1, 2 และ 3
3. เลือกระบบควบคุมแบบอัตโนมัติ หรือ ระบบควบคุมด้วยมือ
4. ตั้งค่า Set point Temperature ได้

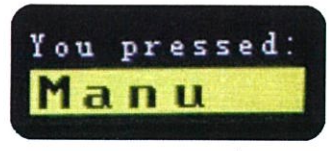


รูปที่ 3.25 หน้าจอทัชสกรีนในระบบควบคุมด้วยมือ

- **ค่าแสดงสถานะบนหน้าจอทัชสกรีน**

หน้าจอทัชสกรีนแสดงค่า Set point ที่ผู้ใช้สามารถตั้งค่าได้ (กำหนดเอง) ดังรูปที่ 3.26 และแสดงสถานะโหมดว่าตอนนี้ผู้ใช้งานเลือกโหมดใด ดังรูปที่ 3.27

รูปที่ 3.26 ค่าอุณหภูมิ Set point



รูปที่ 3.27 สถานะโหมดเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

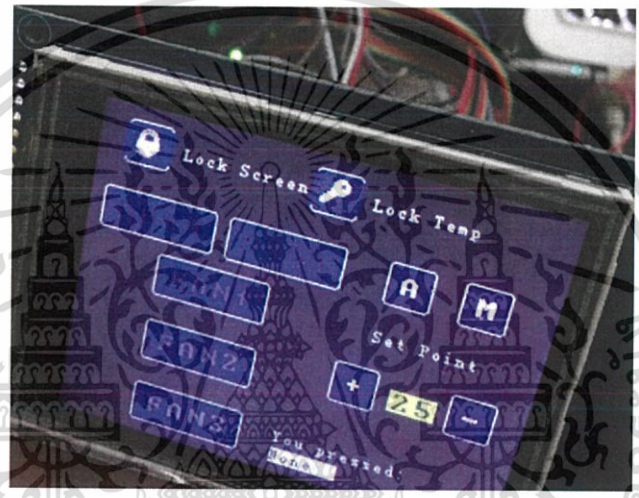
- **อธิบายปุ่มบนหน้าจอทัชสกรีน**

	Lock Screen ปุ่มล็อคหน้าจอ	ล็อคหน้าจอทัชสกรีนผู้ใช้งานไม่กดปุ่มได้
	Lock Temp ปุ่มล็อคค่าSet point	ผู้ใช้งานไม่สามารถเปลี่ยนค่า Set point ได้
	ปุ่มระบบควบคุมแบบอัตโนมัติ (Auto)	เครื่องปรับอากาศทำงานในระบบควบคุมแบบอัตโนมัติ
	ปุ่มระบบควบคุมด้วยมือ (Manual)	เครื่องปรับอากาศทำงานในระบบควบคุมด้วยมือ
	ปุ่มเพิ่มค่า Set point	ค่าอุณหภูมิ Set point มีค่าเพิ่มขึ้น
	ปุ่มลดค่า Set point	ค่าอุณหภูมิ Set point มีค่าลดลง
	ปุ่มเปิดเครื่องปรับอากาศ	เปิดเครื่องปรับอากาศ
	ปุ่มปิดเครื่องปรับอากาศ	ปิดเครื่องปรับอากาศ
	ปุ่มเปิด-ปิดพัดลมตัวที่ 1	เปิด-ปิดพัดลมในระบบควบคุมด้วยมือ
	ปุ่มเปิด-ปิดพัดลมตัวที่ 2	เปิด-ปิดพัดลมในระบบควบคุมด้วยมือ
	ปุ่มเปิด-ปิดพัดลมตัวที่ 3	เปิด-ปิดพัดลมในระบบควบคุมด้วยมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- **การใช้งานหน้าจอทัชสกรีน**

เริ่มแรกหน้าจอทัชสกรีนปรากฏดังรูปที่ 3.28 ผู้ใช้ต้องกดปุ่ม Lock Screen ก่อน เพื่อปลดล็อคหน้าจอ แล้วกดปุ่ม OPEN เมื่อเครื่องปรับอากาศเริ่มทำงาน เครื่องปรับอากาศจะเริ่มทำงานในระบบควบคุมแบบอัตโนมัติก่อน ดังนั้นหน้าจอทัชสกรีนแสดงดังรูปที่ 3.29 เห็นได้ว่าไม่สามารถกดปุ่มเปิด ปุ่มพัดลมตัวที่ 1, 2 และ 3 ได้ แต่ผู้ใช้งานสามารถกดปุ่มปิด เลือกระบบควบคุมด้วยมือ และสามารถเปลี่ยนค่า Set point ได้ ส่วนปุ่มล็อคหน้าจอ และปุ่มล็อคค่าอุณหภูมิ Set point นั้นสามารถกดได้ตลอดเวลาตามความต้องการของผู้ใช้งาน



รูปที่ 3.28 หน้าจอทัชสกรีน

รูปที่ 3.29 หน้าจอทัชสกรีนแสดงตอนเริ่มทำงาน

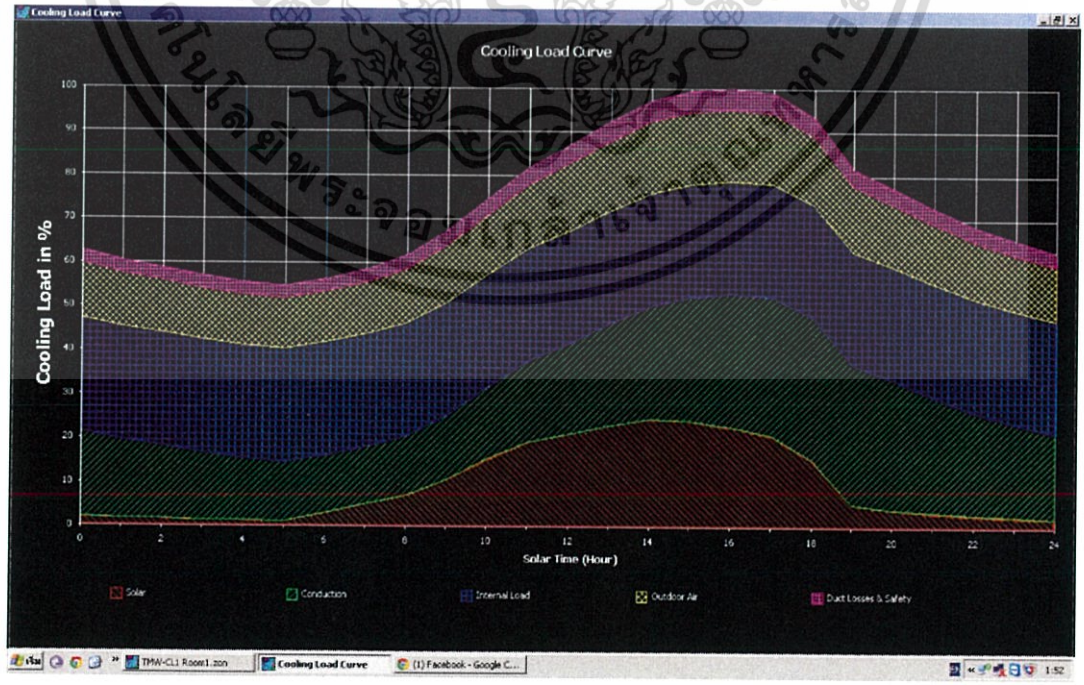
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากผู้ใช้งานต้องการเปลี่ยนเป็นระบบควบคุมด้วยมือ ให้กดปุ่มระบบควบคุมด้วยมือ หน้าจอทัชสกรีนจะปรากฏเป็นดังรูปที่ 3.25 ผู้ใช้งานสามารถเลือกกดปุ่มพัดลมตัวที่ 1, 2 และ 3 ได้ตามความต้องการ และยังสามารถกดปุ่มปิดเครื่องปรับอากาศและ เปลี่ยนค่า Set point ได้ตามต้องการ

3.3.2 ระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด

ระบบโหลดบาลานซ์ เป็นระบบที่จะช่วยแก้ไขปัญหาการทำงานของระบบที่มีการใช้งานหนักได้เป็นอย่างดี โดยการทำบาลานซ์ คือการจัดกลุ่ม ของตัวควบคุม (Server) เพื่อแบ่งงาน กันทำ หรือ กระจายโหลดการใช้งานของผู้ใช้ไปยังตัวควบคุมหลักเครื่องต่างๆ ภายในกลุ่ม สามารถใช้กับฮาร์ดแวร์ ,ซอฟต์แวร์

ดังนั้นระบบควบคุมเครื่องปรับอากาศแบบโหลดบาลานซ์ คือ ระบบควบคุมที่จะทำการบาลานซ์โหลด เพื่อเปิดเครื่องปรับอากาศเป็นจำนวนที่เหมาะสมกับโหลด ซึ่งขึ้นอยู่กับภาระโหลดในห้อง กำหนดให้ในห้องขนาดใหญ่มีเครื่องปรับอากาศเป็นจำนวน 5 เครื่อง เครื่องปรับอากาศแต่ละตัวถูกตั้งเวลาให้เปิด-ปิด ตามเวลาที่สัมพันธ์กับเปอร์เซ็นต์ Cooling load ดังรูปที่ 3.24 เช่น ในเวลา 16.00 น. จะมีเปอร์เซ็นต์ Cooling load สูงที่สุดดังนั้น ในเวลานี้ผู้ใช้งานห้องนี้ ต้องทำการตั้งเวลาให้เปิดเครื่องปรับอากาศทั้ง 5 ตัวให้ทำงานพร้อมกัน แต่ช่วงเวลาอื่น เช่น ช่วงเช้าของวันจะมี เปอร์เซ็นต์ Cooling load น้อย หากผู้ใช้งานต้องการใช้งานช่วงนี้ เครื่องปรับอากาศต้องเปิด 2 หรือ 3 เครื่อง ขึ้นอยู่กับโหลดที่กำหนด



รูปที่ 3.30 กราฟ Cooling load แสดงความสัมพันธ์ระหว่าง Cooling load % กับ Solar Time

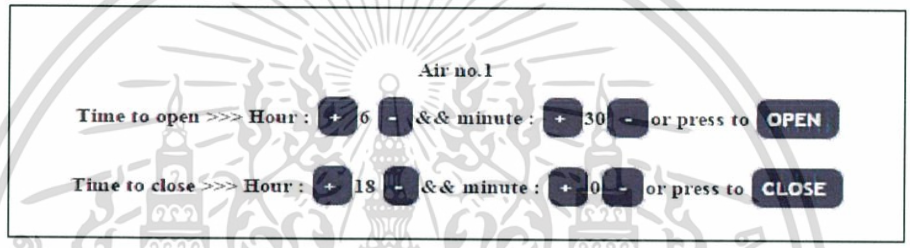
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.1 ควบคุมการเปิด-ปิดเครื่องปรับอากาศโดยการตั้งเวลาผ่านระบบอินเตอร์เน็ต

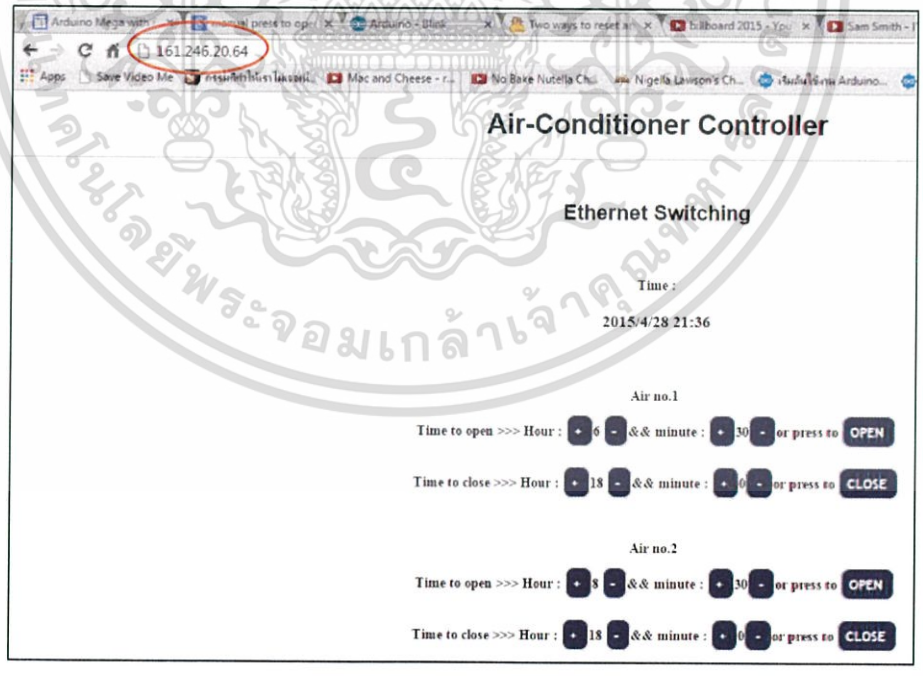
การควบคุมการเปิดปิดเครื่องปรับอากาศผ่านระบบอินเตอร์เน็ตนั้น ผู้ควบคุมสามารถเข้าเว็บแอดเดรส 161.246.20.64 ดังรูปที่ 3.32 จากเครือข่ายอินเตอร์เน็ตที่อยู่ในวง LAN เดียวกัน เพื่อทำการควบคุมการเปิดปิด โดยที่ผู้ควบคุมสามารถควบคุมผ่านหน้าเว็บได้ มีดังนี้

1. สั่งเปิด-ปิดเครื่องปรับอากาศ เครื่องที่ 1 ถึง 5
2. ตั้งค่าเวลาเปิดปิดเครื่องปรับอากาศ เครื่องที่ 1 ถึง 5

นอกจากนี้หน้าเว็บบราวเซอร์สามารถแสดงค่าวัน เวลาแบบเรียลไทม์ (Real Time) ดังรูปที่ 3.33

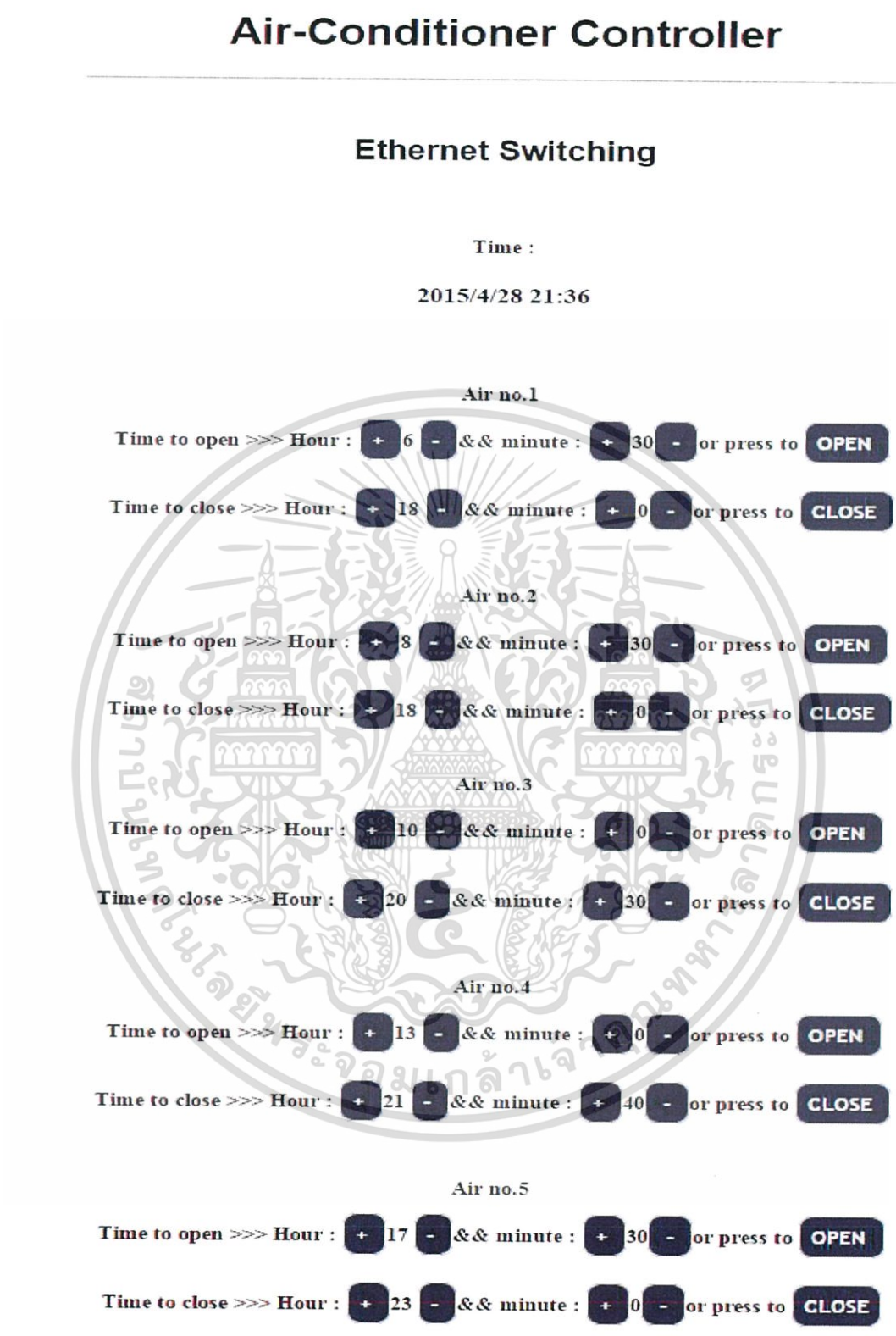


รูปที่ 3.31 การตั้งค่าเวลาเปิดปิดเครื่องปรับอากาศผ่านหน้าเว็บบราวเซอร์



รูปที่ 3.32 หน้าเว็บบราวเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

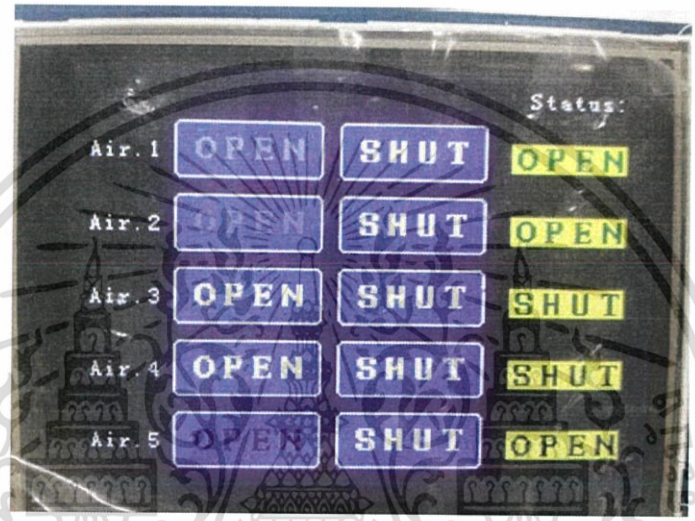


รูปที่ 3.33 หน้าเว็บบราวเซอร์ที่ใช้ควบคุมเครื่องปรับอากาศทั้งหมด (5 เครื่อง)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.2 ควบคุมการเปิด-ปิดเครื่องปรับอากาศผ่านหน้าจอทัชสกรีน

ระบบควบคุมการเปิด-ปิดเครื่องปรับอากาศผ่านหน้าจอทัชสกรีนนั้น ออกแบบมาเพื่อ ให้ระบบควบคุมนี้ยังสามารถทำงานได้อยู่ ในกรณีที่ระบบควบคุมผ่านอินเตอร์เน็ตล้มเหลว (fail) โดยควบคุมผ่านหน้าจอทัชสกรีนแทนระบบควบคุมผ่านอินเตอร์เน็ต โดยที่หน้าจอทัชสกรีนนี้ติดตั้งที่ห้องควบคุม ผู้ใช้งานสามารถสั่งเปิด-ปิดเครื่องปรับอากาศตัวที่ 1 ถึง 5 ผ่านหน้าจอทัชสกรีนได้ และมีการแสดงสถานะว่าเครื่องปรับอากาศนั้นเปิดหรือปิด ดังรูปที่ 3.34



รูปที่ 3.34 หน้าจอทัชสกรีน

- ค่าแสดงสถานะบนหน้าจอทัชสกรีน

เครื่องปรับอากาศตัวที่ 1

เครื่องปรับอากาศตัวที่ 2

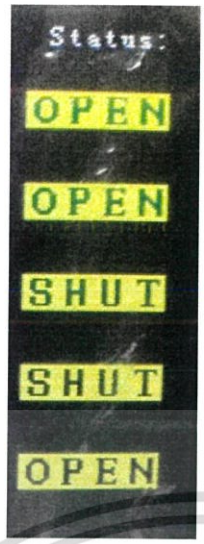
เครื่องปรับอากาศตัวที่ 3

เครื่องปรับอากาศตัวที่ 4

เครื่องปรับอากาศตัวที่ 5

รูปที่ 3.35 เครื่องปรับอากาศตัวที่ 1 – 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สถานะเปิด

สถานะเปิด

สถานะปิด

สถานะปิด

สถานะเปิด

รูปที่ 3.36 สถานะของเครื่องปรับอากาศ

- อธิบายปุ่มบนหน้าจอทัชสกรีน

ปุ่มเปิดเครื่องปรับอากาศ

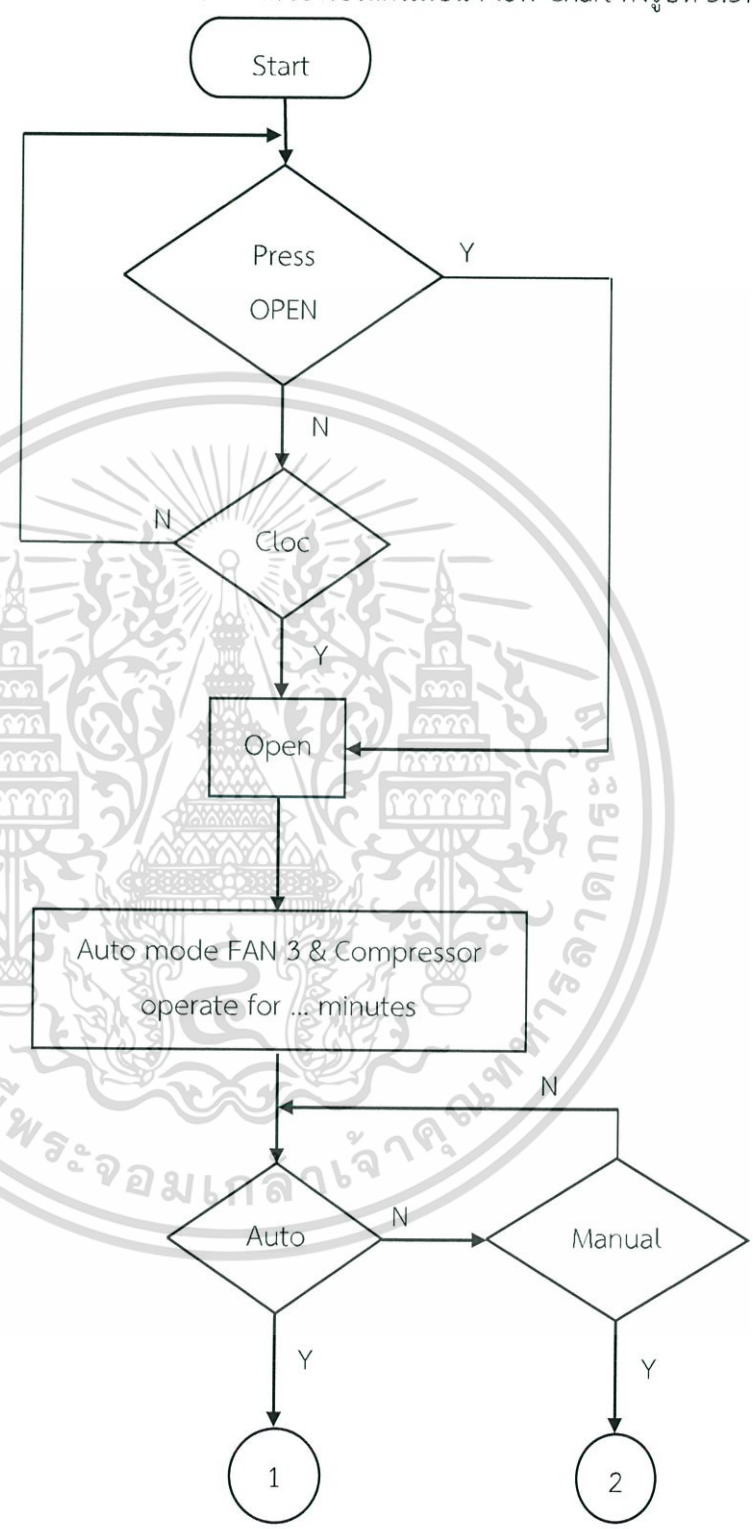
ปุ่มปิดเครื่องปรับอากาศ

- การใช้งานหน้าจอทัชสกรีน

เนื่องจากหน้าจอทัชสกรีนมี 2 ปุ่มจึงใช้ได้แค่ในการ เปิด – ปิด เครื่องปรับอากาศตัวที่ 1 ถึง 5 เท่านั้น

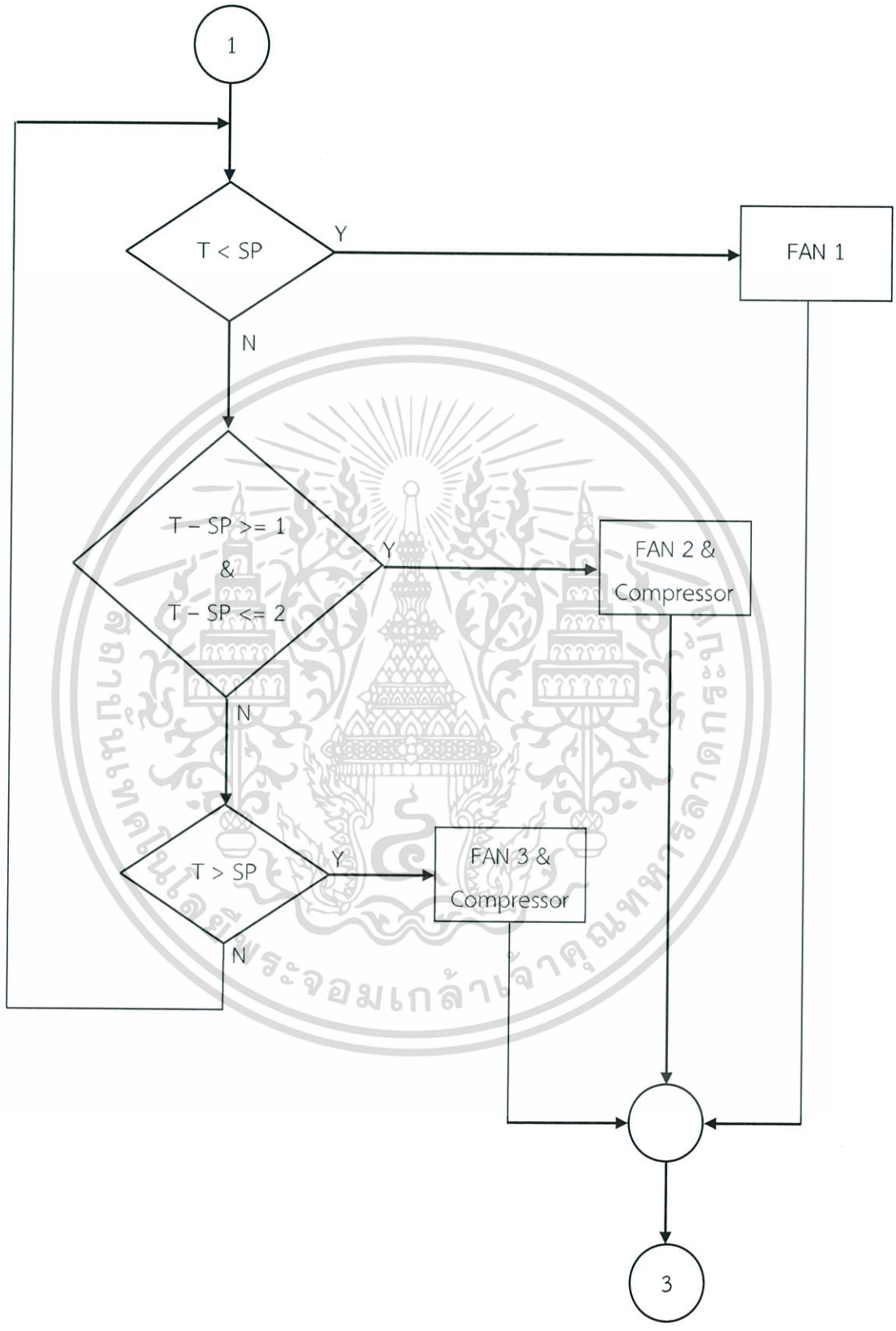
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการออกแบบระบบควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลา ของระบบควบคุมเครื่องปรับอากาศ สามารถเขียนขั้นตอนการทำงานของโปรแกรมเป็น Flow Chart ดังรูปที่ 3.37



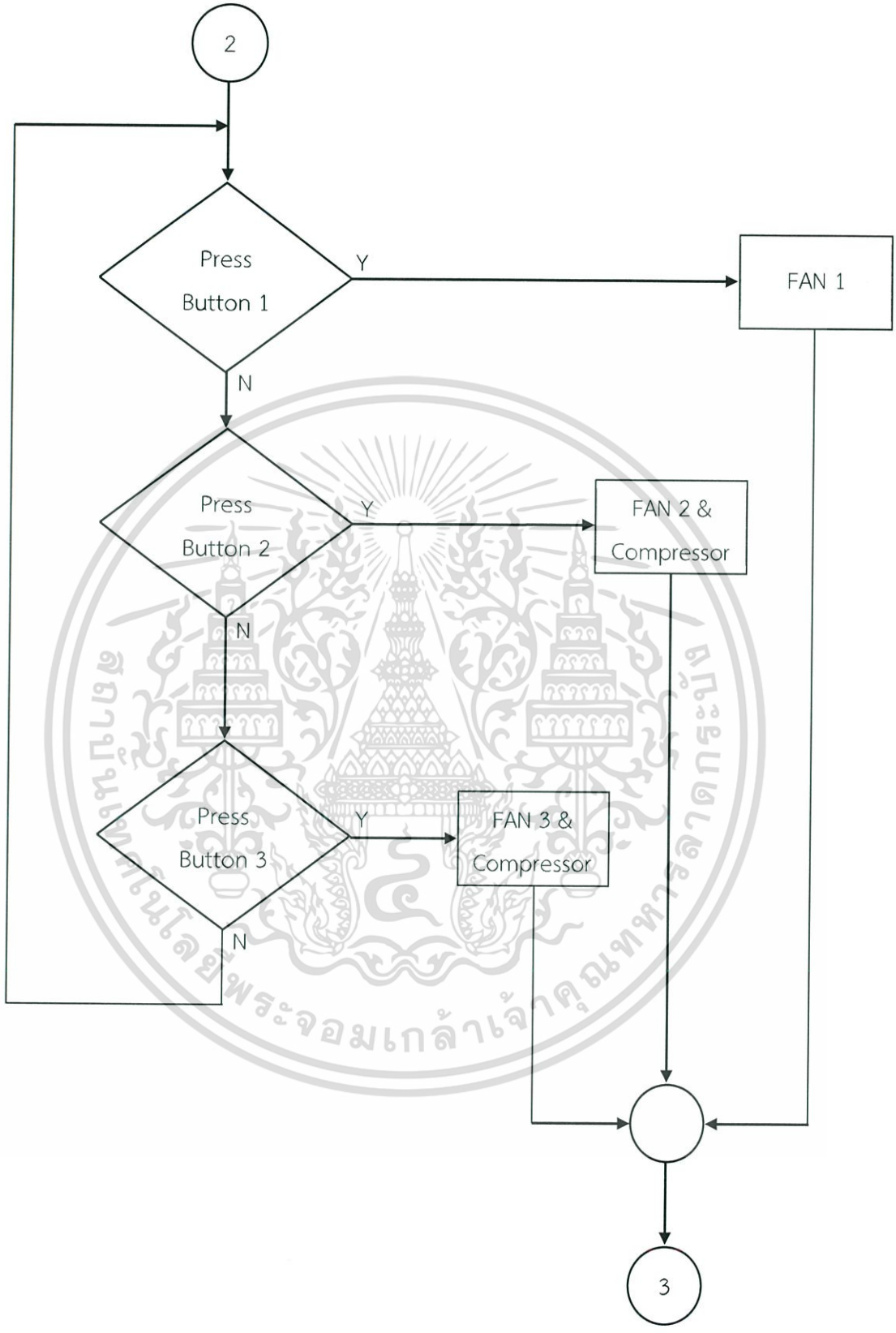
รูปที่ 3.37 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



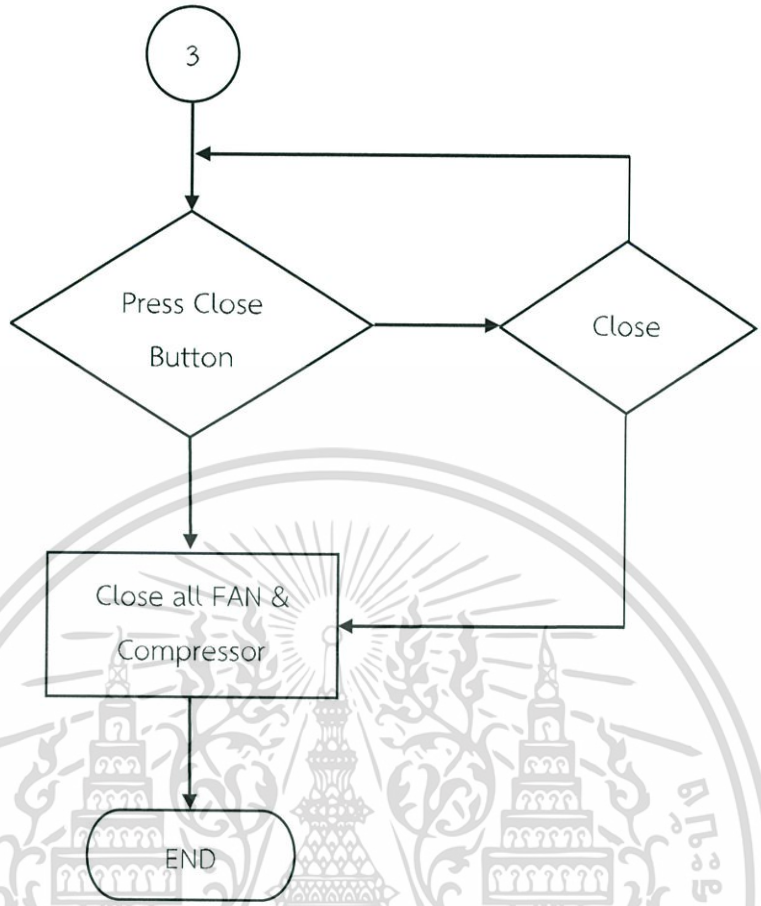
รูปที่ 3.37 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.37 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศ (ต่อ)

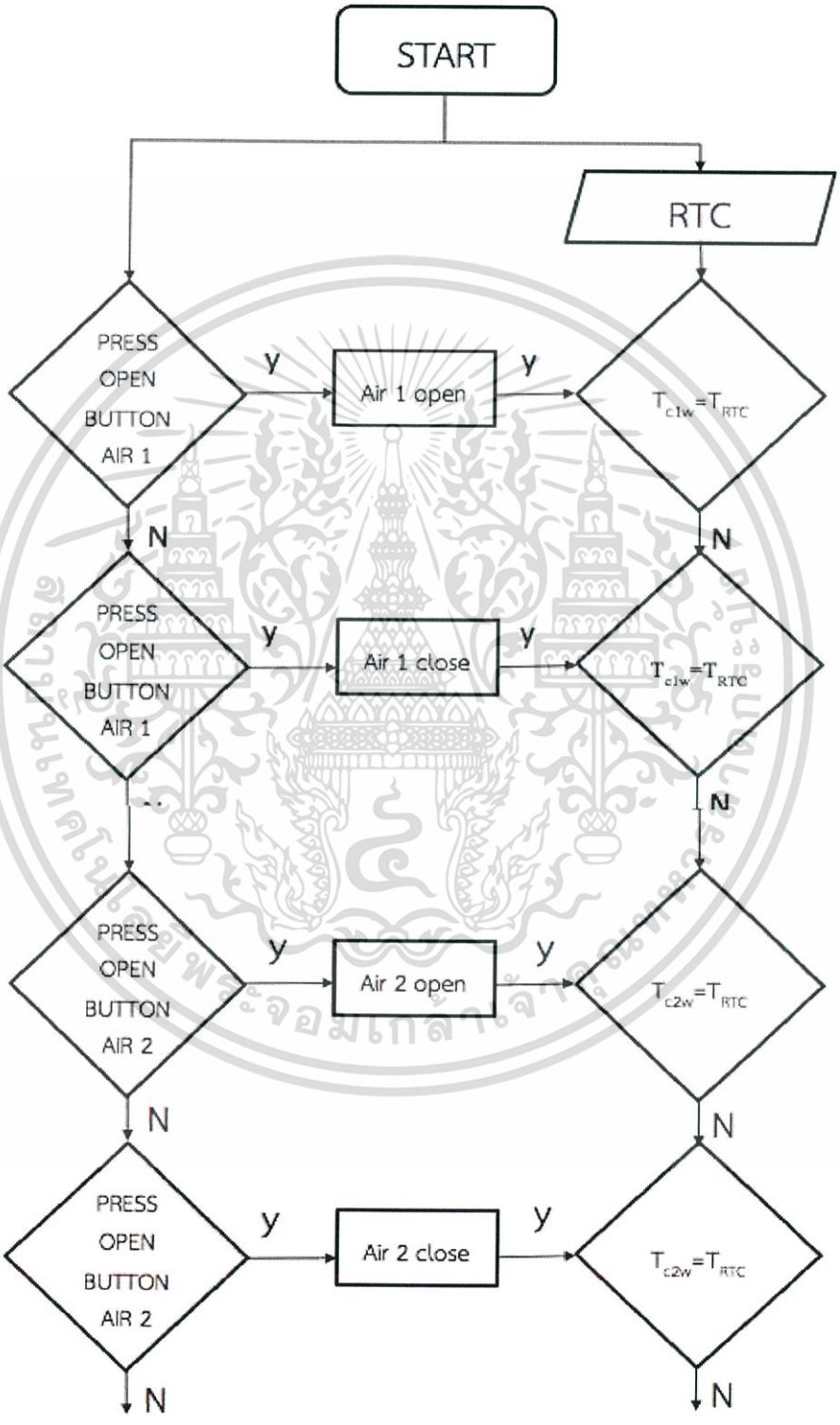
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.37 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศ (ต่อ)

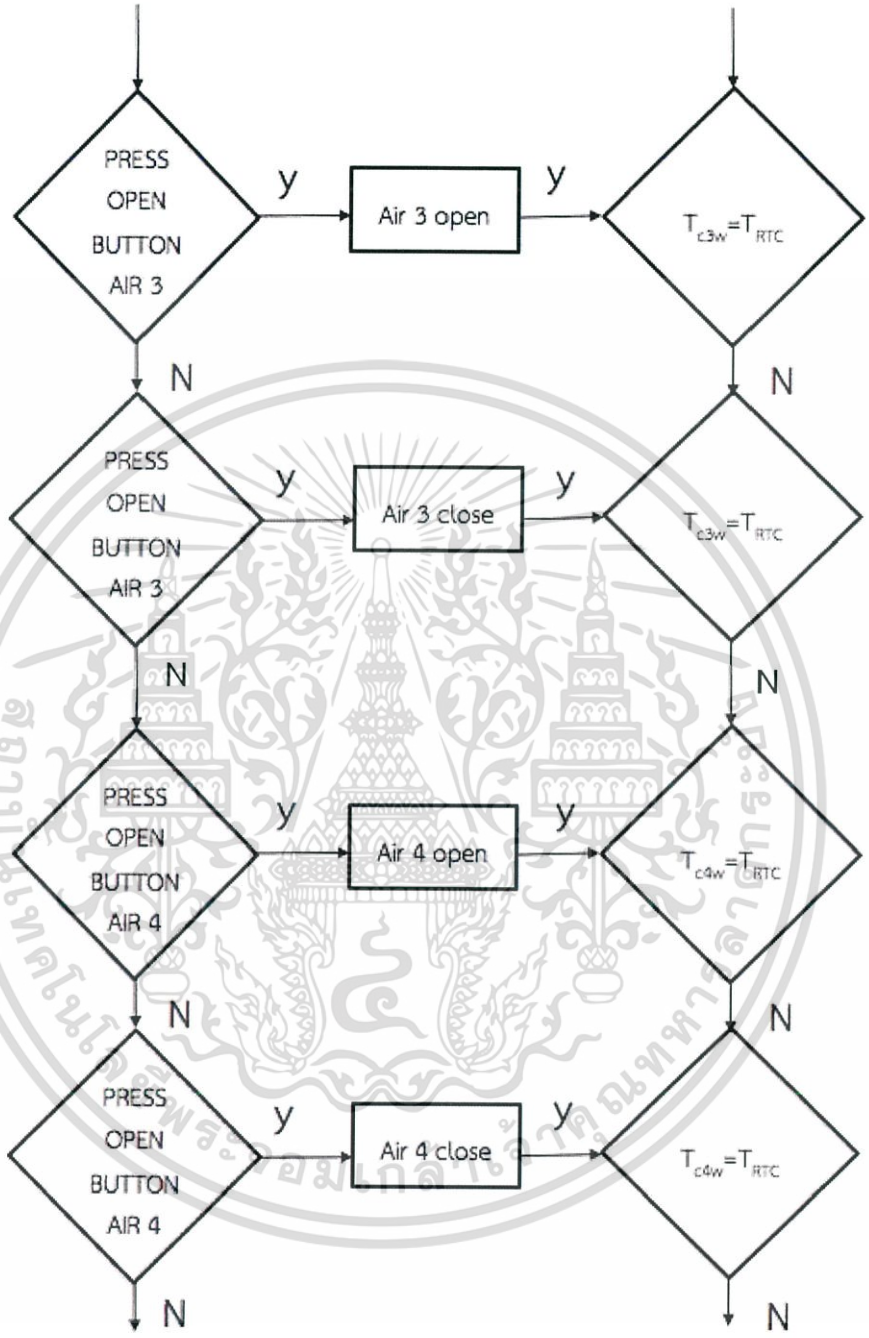
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการออกแบบระบบควบคุมการเปิด-ปิดเครื่องปรับอากาศตามเวลา ของระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด สามารถเขียนขั้นตอนการทำงานของโปรแกรมเป็น Flow Chart ดังรูปที่ 3.38



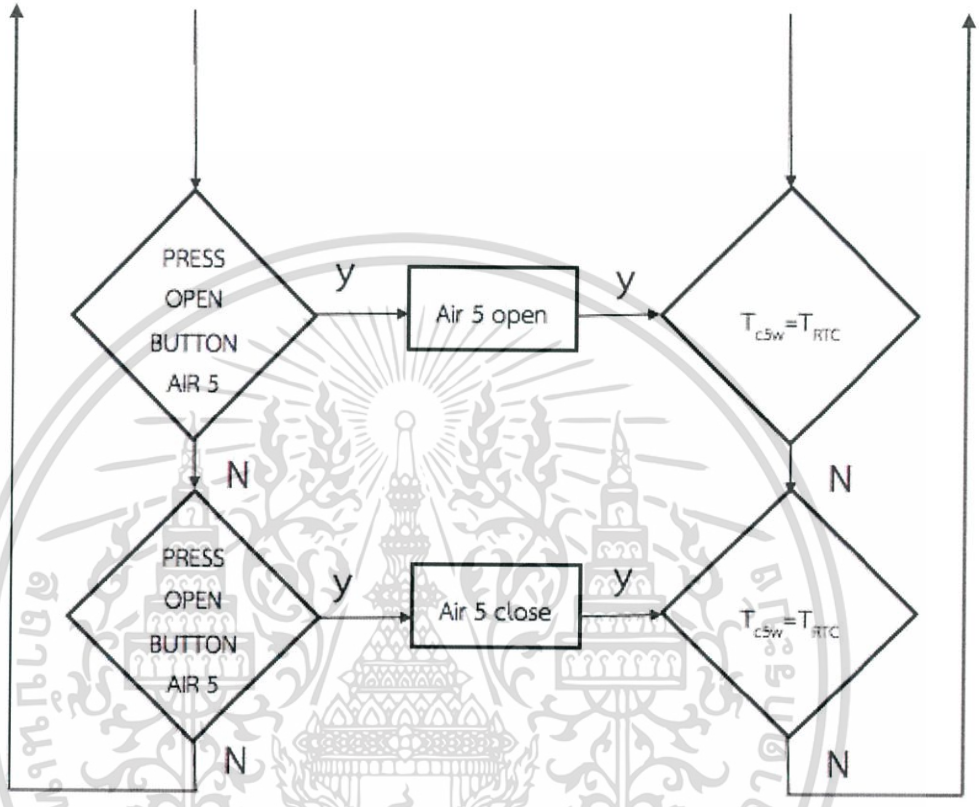
รูปที่ 3.38 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 ขั้นตอนการทำงานของระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลอง

4.1 กล่าวนำ

การทดลองการเปิด-ปิดเครื่องปรับอากาศให้เปิด-ปิดแบบภาระโหลด ตามเวลาที่ตั้งไว้ผ่านระบบอินเตอร์เน็ตและ เปิด-ปิดผ่านหน้าจอทัชสกรีน โดยทำการต่ออุปกรณ์ควบคุมนี้ (ด้านฮาร์ดแวร์) กับหลอด LED เพื่อแสดงสถานะการเปิด-ปิดของเครื่องปรับอากาศ และตรวจสอบดูว่าเครื่องปรับอากาศนั้นเปิด-ปิดจริงตรงกับที่ตั้งเวลาไว้ หรือ เปิด-ปิดจริงตรงกับการกดปุ่มคำสั่งหรือไม่

4.2 การทดลองการเปิด-ปิดเครื่องปรับอากาศแบบภาระโหลด

การทดลองนี้จำลองให้ทดลองกับการต่ออุปกรณ์ควบคุมนี้กับหลอดไฟ LED ซึ่งหลอด LED แสดงสถานะการเปิด-ปิด ของเครื่องปรับอากาศ เช่น ถ้าหลอดไฟติดแสดงให้รู้ว่ามีกระแสไหลผ่านไปที่ Relay ซึ่ง Relay แต่ละตัวเป็นอุปกรณ์ควบคุมการเปิด-ปิดของอุปกรณ์ไฟฟ้าหรือเครื่องปรับอากาศ ถ้ามีกระแสไหลผ่านไปที่ Relay ตัวใด Relay ตัวนั้นจะทำหน้าที่เปิดเครื่องปรับอากาศ แต่หากไม่มีกระแสไหลผ่าน Relay จะไม่ทำงานเครื่องปรับอากาศจึงไม่ทำงานเช่นกัน

4.2.1 การทดลองเปิด-ปิดเครื่องปรับอากาศโดยการตั้งเวลาผ่านระบบอินเตอร์เน็ต

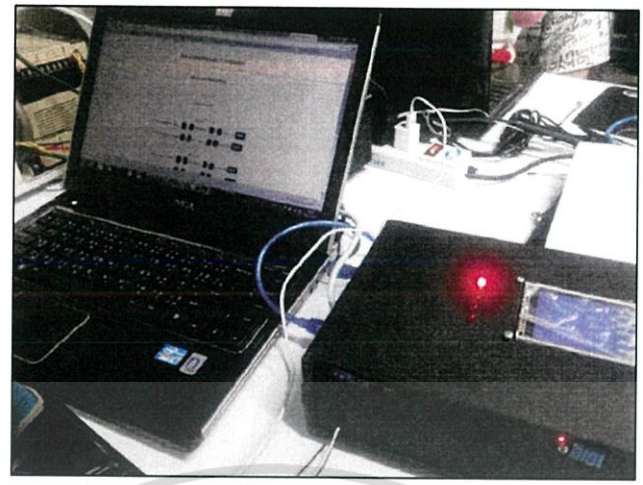
จากการออกแบบให้มีการเปิด-ปิดโดยการตั้งเวลาผ่านระบบอินเตอร์เน็ตนั้น ผู้ใช้สามารถกดปุ่มคำสั่งเปิด หรือปิดเครื่องปรับอากาศได้ และสามารถตั้งค่าเวลาเปิด-ปิดเครื่องปรับอากาศได้ จึงออกแบบการทดลองเป็น 2 แบบ

1. ทดลองให้ผู้ใช้งานกดปุ่มคำสั่งหน้าเว็บบราวเซอร์ผ่านระบบอินเตอร์เน็ต
2. ทดลองให้ผู้ใช้งานตั้งเวลาเปิด-ปิดที่หน้าเว็บบราวเซอร์ผ่านระบบอินเตอร์เน็ต

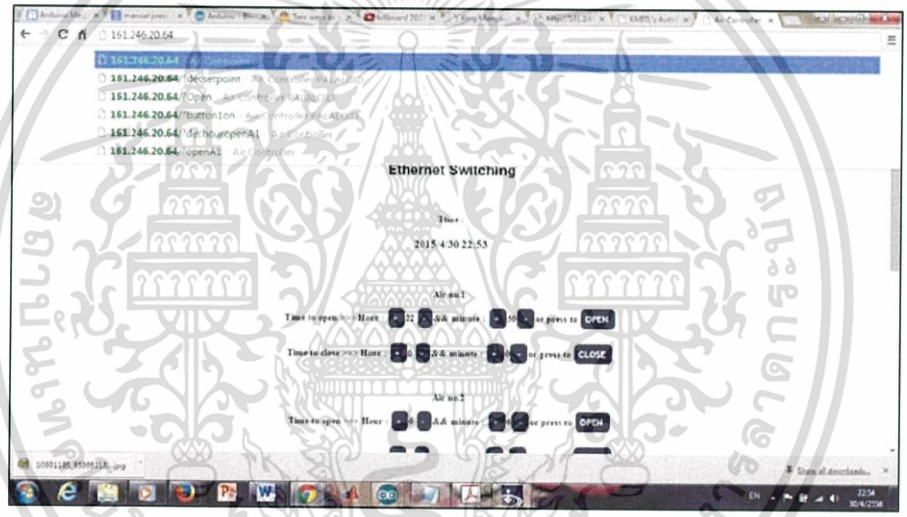
4.2.1.1 ขั้นตอนการทดลอง

1. ติดตั้งอุปกรณ์ควบคุม
2. เข้า IP address 161.246.20.64
3. กดปุ่มคำสั่ง OPEN ของเครื่องปรับอากาศตัวที่ 1 (Air no.1)
4. ตั้งเวลาเปิด – ปิดเครื่องปรับอากาศ
5. สังเกตผลการทดลองจากหลอดไฟ LED

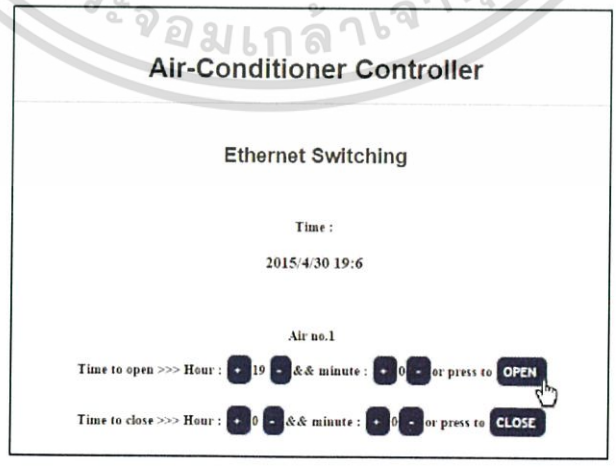
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 การติดตั้งอุปกรณ์ควบคุมกับคอมพิวเตอร์

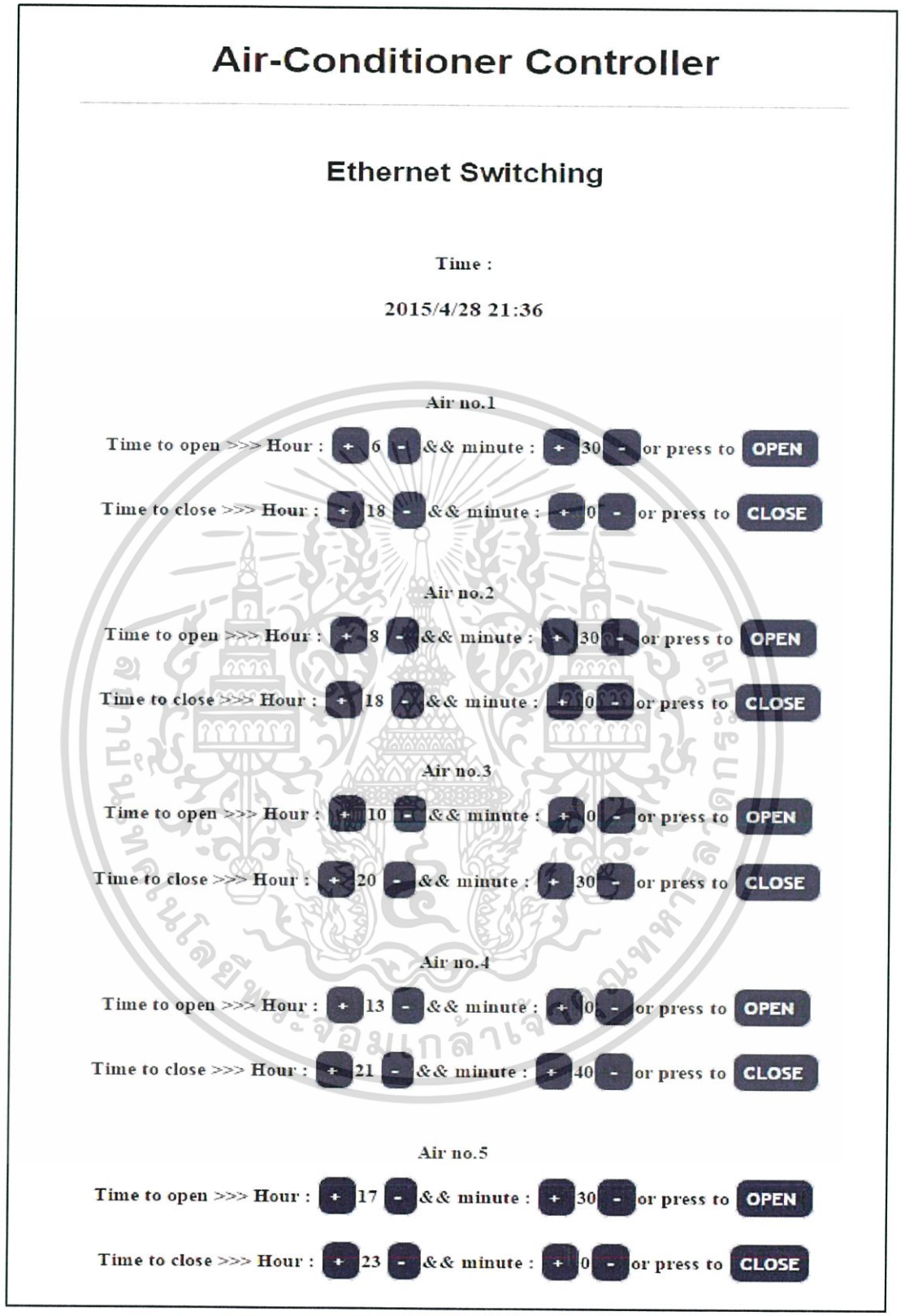


รูปที่ 4.2 การเข้า IP address 161.246.20.64



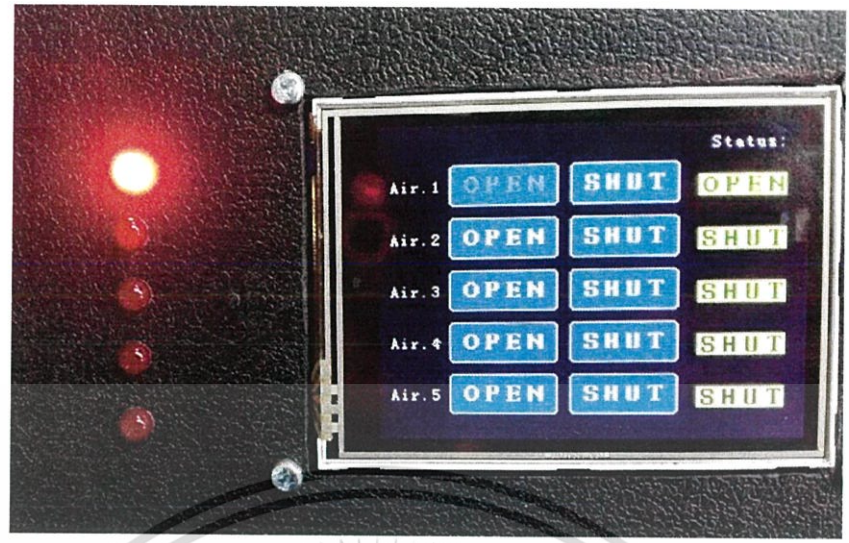
รูปที่ 4.3 การกดปุ่มคำสั่ง OPEN ของเครื่องปรับอากาศตัวที่ 1 (Air no.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 การตั้งเวลาเปิด-ปิดเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 การทำงานของเครื่องปรับอากาศตัวที่ 1 (LED 1 ติด)

4.2.1.2 ผลการทดลอง

4.2.1.2.1 ผลการทดลองของการกดปุ่มคำสั่ง OPEN และ CLOSE

เมื่อกดปุ่มคำสั่ง OPEN ของเครื่องปรับอากาศตัวที่ 1 (Air no.1) หลอดไฟ LED หลอดที่ 1 แสดงสถานะเป็นไฟสีแดง แสดงให้เห็นว่าเครื่องปรับอากาศตัวที่ 1 เปิด จากนั้นกดปุ่มคำสั่ง CLOSE ของเครื่องปรับอากาศตัวที่ 1 (Air no.1) หลอดไฟ LED หลอดที่ 1 ไฟดับลง แสดงให้เห็นว่าเครื่องปรับอากาศตัวที่ 1 ปิด

จากตารางที่ 4.1 แสดงให้เห็นผลการทดสอบของเครื่องปรับอากาศทั้ง 5 ตัวว่าเมื่อกดปุ่มคำสั่ง OPEN เครื่องปรับอากาศจะเปิด และกดปุ่มคำสั่ง CLOSE เครื่องปรับอากาศจะปิด

ตารางที่ 4.1 ตารางแสดงการทดลองกดปุ่มเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัวผ่านเว็บบราวเซอร์

เครื่องปรับอากาศ	OPEN	CLOSE
Air no.1	✓	✓
Air no.2	✓	✓
Air no.3	✓	✓
Air no.4	✓	✓
Air no.5	✓	✓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.2.2 ผลการทดลองของการตั้งค่าเวลาเปิด-ปิดเครื่องปรับอากาศ

เมื่อทำการตั้งค่าเวลาให้เครื่องปรับอากาศตัวที่ 1 (Air no.1) ทำงานในช่วงเวลา 06.30 น. ถึง 18.00 น. ผลการทดลองที่ได้คือ เครื่องปรับอากาศสามารถทำงานในช่วงเวลา 06.30 น. ถึง 18.00 น. ได้จริง

ตารางที่ 4.2 ตารางแสดงการทดสอบการตั้งค่าเวลาเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัว

เครื่องปรับอากาศ	Time to OPEN	Time to CLOSE
Air no.1	06.30	18.00
Air no.2	08.30	18.00
Air no.3	10.00	20.30
Air no.4	13.00	21.40
Air no.5	17.30	23.00

จากตารางที่ 4.2 แสดงให้เห็นผลการทดสอบของเครื่องปรับอากาศทั้ง 5 ตัว ว่าเครื่องปรับอากาศจะทำงานในช่วงเวลาเปิด-ปิด ที่ตั้งไว้จริง โดยสังเกตจากหลอดไฟ LED ว่าไฟติดและดับตามเวลาที่ตั้งไว้หรือไม่

4.2.2 การทดลองเปิด-ปิดเครื่องปรับอากาศโดยผ่านหน้าจอทัชสกรีน

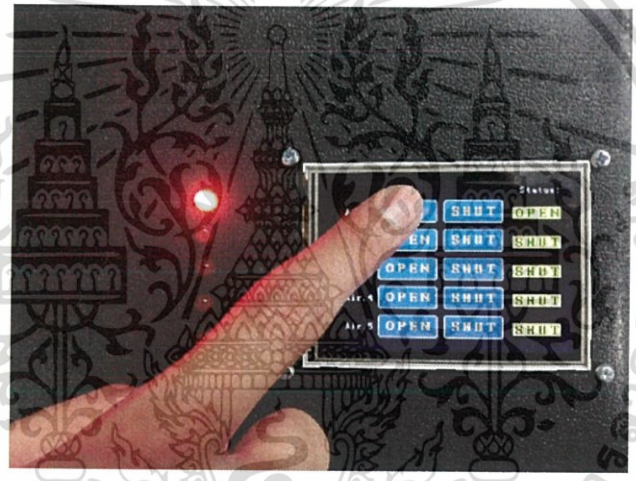
จากการออกแบบหน้าจอทัชสกรีนดังรูปที่ 3.28 เห็นได้ว่าผู้ใช้งานสามารถควบคุมการเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัวได้ โดยการกดปุ่มคำสั่งเปิด-ปิด (OPEN-SHUT) แต่ไม่สามารถควบคุมการเปิด-ปิดจากการตั้งเวลาได้ ดังนั้นจึงออกแบบการทดลองเป็นการกดปุ่มคำสั่งที่หน้าจอทัชสกรีน แล้วดูผลการทดลองว่าเครื่องปรับอากาศเปิด-ปิดตามที่ผู้ใช้งานกดปุ่มคำสั่งหรือไม่

4.2.2.1 ขั้นตอนการทดลอง

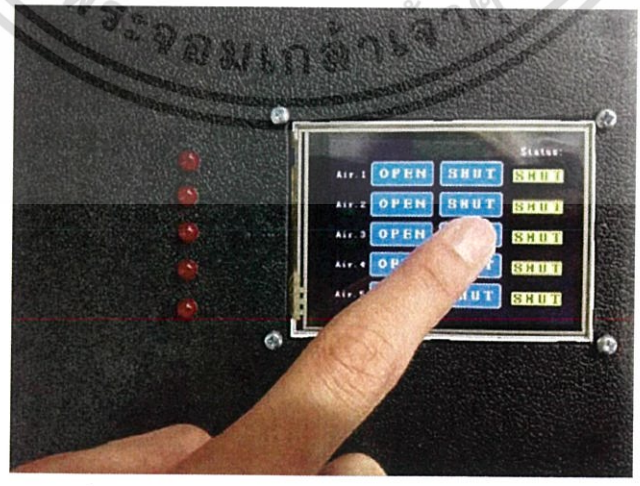
1. ติดตั้งอุปกรณ์ควบคุม
2. กดปุ่มคำสั่ง OPEN และ SHUT ที่หน้าจอทัชสกรีน
3. สังเกตผลการทดลองจากหลอดไฟ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.6 การติดตั้งอุปกรณ์ควบคุมกับคอมพิวเตอร์



รูปที่ 4.7 การกดปุ่มคำสั่ง OPEN ที่หน้าจอทัชสกรีน



รูปที่ 4.8 การกดปุ่มคำสั่ง SHUT ที่หน้าจอทัชสกรีน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.2 ผลการทดลองของการกดปุ่มคำสั่ง OPEN และ SHUT

จากตารางที่ 4.3 แสดงให้เห็นผลการทดสอบของเครื่องปรับอากาศทั้ง 5 ตัวว่า เมื่อกดปุ่มคำสั่ง OPEN เครื่องปรับอากาศจะเปิด และกดปุ่มคำสั่ง SHUT เครื่องปรับอากาศจะปิด และหน้าจอทัชสกรีนแสดงสถานะ OPEN หรือ SHUT ตามที่ผู้ใช้งานกด

ตารางที่ 4.3 ตารางแสดงการทดลองกดปุ่มเปิด-ปิดเครื่องปรับอากาศทั้ง 5 ตัวผ่านหน้าจอทัชสกรีน

เครื่องปรับอากาศ	OPEN	SHUT
Air .1	✓	✓
Air .2	✓	✓
Air .3	✓	✓
Air .4	✓	✓
Air .5	✓	✓

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

จากการทดลองทำอุปกรณ์ควบคุมการเปิด-ปิดเครื่องปรับอากาศให้สามารถเปิด-ปิดตามเวลาที่กำหนดได้ โดยใช้ Arduino Mega 2560 board เป็นบอร์ดควบคุมหลัก ใช้ Ethernet Shield W5100 เป็นอุปกรณ์สร้างเว็บบราวเซอร์เพื่อควบคุมการเปิดปิดเครื่องปรับอากาศผ่านอินเตอร์เน็ต ใช้หน้าจอทัชสกรีนในการควบคุมการเปิดปิดเครื่องปรับอากาศผ่านระบบแมนนวลที่ห้องควบคุม และใช้อุปกรณ์ RTC (Real Time Clock DS1307) ในการควบคุมการเปิดปิดเครื่องปรับอากาศโดยการตั้งเวลา สามารถสรุปได้ ดังนี้

ในด้านฮาร์ดแวร์ของอุปกรณ์ควบคุมการเปิดปิดเครื่องปรับอากาศนั้น เมื่อทำการทดลองการเปิดปิดเครื่องปรับอากาศ พบว่าสามารถใช้อุปกรณ์นี้สั่งเปิด สั่งปิดเครื่องปรับอากาศทั้ง 5 ตัว ตามเงื่อนไขของโปรแกรมได้ และอุปกรณ์แต่ละตัวที่ต่อกับบอร์ด Arduino Mega 2560 สามารถทำงานตามฟังก์ชันของแต่ละอุปกรณ์ได้ เช่น หน้าจอทัชสกรีน สามารถกดปุ่มคำสั่งเปิดปิดเครื่องปรับอากาศได้จริง

ในด้านซอฟท์แวร์ของอุปกรณ์ควบคุมการเปิดปิดเครื่องปรับอากาศนั้น เมื่อทำการทดลองการเปิดปิดเครื่องปรับอากาศ พบว่าเครื่องปรับอากาศสามารถเปิดปิดตามเวลาที่กำหนดได้ สามารถสั่งการเปิดปิดผ่านระบบอินเตอร์เน็ตได้ และสามารถสั่งการเปิดปิดผ่านระบบแมนนวลที่หน้าจอทัชสกรีนได้จริง

5.2 ข้อเสนอแนะ

จากการศึกษาออกแบบ และทำการทดลอง ทำให้ทราบถึงปัญหาที่เกิดขึ้น ดังนี้

1. ควรศึกษาการเขียนโปรแกรมให้ชัดเจนก่อนว่า โปรแกรมที่จะทำการเขียนขึ้นนั้นใช้พื้นที่หน่วยความจำขนาดเท่าไหร่ ก่อนเลือกซื้อบอร์ดควบคุม เพราะบอร์ด Arduino ในแต่ละรุ่นมีขนาดหน่วยความจำไม่เท่ากัน
2. ควรออกแบบให้ชัดเจนว่าเลือกใช้อุปกรณ์ต่อกับบอร์ด Arduino กี่ตัว ใช้จำนวนช่องอินพุตเอาท์พุตกี่ช่อง เพื่อเลือกบอร์ด Arduino ที่มีช่องอินพุต และเอาท์พุตให้เพียงพอกับความต้องการ
3. ในการต่อสายระหว่างอุปกรณ์ ควรต่อสายให้แน่น เพราะหากต่อสายไม่แน่นจะมีผลในการส่งสัญญาณระหว่างอุปกรณ์ เช่น หากต่อสายหน้าจอทัชสกรีนไม่แน่นพอ ทำให้หน้าจอแสดงภาพไม่ชัดเจน
4. หากนำอุปกรณ์ควบคุมนี้ไปใช้งานจริง ควรเลือกจอทัชสกรีนรุ่นที่ดีกว่าเพื่อความทันสมัย ความเหมาะสม และความสะดวกในการเปิดปิดเครื่องปรับอากาศผ่านระบบแมนนวล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Arduino. [Online]. Available : http://arduino.cc/

[2] Arduino UNO. [Online]. Available :
http://www.arduitronics.com/article/%E0%B9%80%E0%B8%A3%E0%B8%B4%E0%B9%88%E0%B8%A1%E0%B8%95%E0%B9%89%E0%B8%99%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99-arduino-uno-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-1-%E0%B9%81%E0%B8%99%E0%B8%B0%E0%B8%99%E0%B8%B3%E0%B8%95%E0%B8%B1%E0%B8%A7%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B8%81%E0%B9%88%E0%B8%AD%E0%B8%99/

[3] RTC DS1307. [Online]. Available :
http://nicuflorica.blogspot.ro/2013/06/ceas-de-timp-real-rtc-cu-ds1307-si.html/

[4] Arduino and online data logger using ENC28J60 Ethernet Module. [Online]. Available :
http://www.arduitronics.com/article/arduino-and-online-data-logger-using-enc28j60-ethernet-module/

[5] Arduino Temp/Humidity with LCD and Web Interface. [Online]. Available :
http://www.instructables.com/id/Arduino-TempHumidity-with-LCD-and-Web-Interface/?ALLSTEPS/

[6] Arduino LCD Thermometer , temperature control with TMP36GZ Temp sensor added 2 chanel relay, added RTC ds1307 , added a case.based on WWC Guide. [Online]. Available :
http://www.instructables.com/id/Arduino-LCD-Thermometer-with-TMP36GZ-Temp-Sensor-b/

[7] Arduino Webserver Control Lights, Relays, Servos, etc. [Online]. Available :
http://www.instructables.com/id/Arduino-Webserver-Control-Lights-Relays-Servos-etc/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[8] Pin control over the Internet-Arduino+Ethernet. [Online]. Available : http://bildr.org/2011/06/arduino-ethernet-pin-control/

[9] Henning Karlsen. 2014. UTFT Arduino and chipKit Universal TFT display library Manual : Rinky-Dink Electronics

[10] Henning Karlsen. 2014. UTouch Arduino and chipKit Universal TFT display library Manual : Rinky-Dink Electronics

[11] Henning Karlsen. 2014. UTFT_Buttons Add-on library for UTFT, Buttons Manual : Rinky-Dink Electronics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก1.

โปรแกรมระบบควบคุมเครื่องปรับอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <SPI.h> //library for Ethernet
#include <Ethernet.h> //library for Ethernet
#include "DHT.h"  //library for DHT11
#include <UTFT.h>   //library for TFT
#include <UTouch.h> //library for TFT
#include <UTFT_Buttons.h> //library for TFT
#define DHTPIN A0
// what pin we're connected to DHT11
// Uncomment whatever type you're using!
#define DHTTYPE DHT11
// DHT 11
 // #define DHTTYPE DHT22
 // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21
(AM2301)
DHT dht(DHTPIN, DHTTYPE);
#include <Wire.h>
#include "RTClib.h"  //ไลบรารี่สำหรับ ds1307
RTC_DS1307 RTC;          // for ds1307
#define RELAY1 14        // pin for relay
#define RELAY2 15        // pin for relay
#define RELAY3 16        // pin for relay
#define RELAY4 17        // pin for relay
#define power A1 // pin for led status
#define auto A2 // pin for led status
#define manual A3  // pin for led status
#define fan1 A8 // pin for led status
#define fan2 A9 // pin for led status
#define fan3 A10// pin for led status
// Connect the pole of the switch to pin 18
const int SW_PIN = 18;
EthernetClient client;
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
0xED };
//physical mac address : do not change
// ip in lan (that's what you need to use in
your     browser)
byte ip[] = { 161, 246, 29, 64 };
//manual setup
byte gateway[] = { 161, 246, 29, 1 };
// internet access via router, manual setup
byte subnet[] = { 255, 255, 255, 0 };
//subnet mask, manual setup
EthernetServer server(80);
//server port
String readString ;
int refreshPage = 5;
// refresh page every 5 sec.
int sp = 25; //initial set point value
int ho = 10;  // time to open : initial value :
hour  10.30 am
int mo = 30;  // time to open : initial value :
minute 10.30 am
int hc = 22;// time to close : initial value :
hour  22.30 pm
int mc = 30;  // time to close : initial value :
minute 22.30 pm
int but1, but2, but3, but4, but5, but6, but7,
but8, but9, but10, but11, butX,
// define buttons to show on TFT
pressed_button;
int led = 13;      // use for reset board after
//close air conditioner
int resetPin = 12;     //  use for reset board
//after close air conditioner
// Declare which fonts we will be using for
//TFT
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
//extern uint8_t SevenSegNumFont // too
big
extern uint8_t Dingbats1_XL[ ];
// Set up UTFT...
// Set the pins to the correct ones for your
development board
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// ----------------------------------------
// Standard Arduino 2009/Uno/Leonardo
shield      : <display model>,19,18,17,16
// Standard Arduino Mega/Due shield
: <display model>,38,39,40,41
// CTE TFT LCD/SD Shield for Arduino Due
: <display model>,25,26,27,28
// Standard chipKit Uno32/uC32
: <display model>,34,35,36,37
// Standard chipKit Max32
: <display model>,82,83,84,85
// AquaLEDSource All in One Super Screw
//Shield   : <display model>,82,83,84,85
// Remember to change the model
//parameter to suit your display module!
UTFT          myGLCD (ITDB32S,38,39,40,41);
// Set up UTouch...
// Set the pins to the correct ones for your
//development board
// ----------------------------------------
// Standard Arduino 2009/Uno/Leonardo
//shield    : 15,10,14,9,8
// Standard Arduino Mega/Due shield
//: 6,5,4,3,2
// CTE TFT LCD/SD Shield for Arduino Due
//: 6,5,4,3,2
// Standard chipKit Uno32/uC32
//: 20,21,22,23,24
// Standard chipKit Max32
//: 62,63,64,65,66
// AquaLEDSource All in One Super Screw
//Shield    : 62,63,64,65,66
UTouch        myTouch(6,5,4,3,2);
// Finally we set up UTFT_Buttons :)
UTFT_Buttons  myButtons (&myGLCD,
&myTouch);
void setup ( )
{
  Serial.begin(9600);   //board rate
  Wire.begin ( );  // for RTC
  dht.begin ( ); // for DHT11
  RTC.begin ( );  //start RTC
    if (! RTC.isrunning ( ) )
        {
          Serial.println ("RTC is NOT
running!");
 // following line sets the RTC to the date &
time this sketch was compiled
          RTC.adjust(DateTime(2015, 4, 23,
16, 51, 0));
        }
// define font size, font color and initial
command to //TFT
  myGLCD.InitLCD ( );
  myGLCD.clrScr ( );
  myGLCD.setFont (SmallFont);
  myTouch.InitTouch ( );
  myTouch.setPrecision (PREC_MEDIUM);
  myButtons.setTextFont (BigFont);
  myButtons.setSymbolFont
(Dingbats1_XL);
  myButtons.setButtonColors
(VGA_YELLOW, VGA_GRAY, VGA_WHITE,
VGA_RED, VGA_BLUE);
 // (text and symbol, text and symbol when
//disable, boarder, boarder when select,
//back ground)
//define button size, coordinate and name
 but1 = myButtons.addButton ( 10,  60, 80,
30, "OPEN", BUTTON_DISABLED);
  but2 = myButtons.addButton ( 100,  60, 80,
30, "SHUT", BUTTON_DISABLED);
  but3 = myButtons.addButton ( 55, 100, 80,
30, "FAN1", BUTTON_DISABLED);
  but4 = myButtons.addButton ( 55, 150, 80,
30, "FAN2", BUTTON_DISABLED);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  but5 = myButtons.addButton ( 55, 200, 80,
30, "FAN3", BUTTON_DISABLED);
  but6 = myButtons.addButton (190, 140, 30,
30, "+");
  but7 = myButtons.addButton ( 270, 140,
30,  30, "-"); // minus button
  but8 = myButtons.addButton ( 20, 10, 30,
30, "Y", BUTTON_SYMBOL); // Lock screen
  but9 = myButtons.addButton ( 150, 10, 30,
30, "Z", BUTTON_SYMBOL); // Lock SP
  but10 = myButtons.addButton ( 210,  60,
30,  30, "A");
  but11 = myButtons.addButton ( 260,  60,
30,  30, "M");
  myButtons.drawButtons ( ); // use for draw
//button, necessary !
    {
      myGLCD.print("You pressed:", 160,
205);
      myGLCD.print("Set Point", 210, 110);
      myGLCD.print("Lock Screen", 60,20);
      myGLCD.print("Lock Temp", 190, 20);
      myGLCD.setColor(VGA_BLACK);
      myGLCD.setBackColor(VGA_WHITE);
       myGLCD.print("None  ", 160, 220);
     }
     {
      myGLCD.setFont(BigFont);
      myGLCD.setColor(VGA_GREEN);
      myGLCD.setBackColor(VGA_YELLOW);
      myGLCD.printNumI(sp, 230, 150);
     }
  pinMode(RELAY1, OUTPUT);   // define pin
either OUTPUT or INPUT
  pinMode(RELAY2, OUTPUT);   // define pin
either OUTPUT or INPUT
  pinMode(RELAY3, OUTPUT);   // define pin
either OUTPUT or INPUT
  pinMode(RELAY4, OUTPUT);   // define pin
either OUTPUT or INPUT
   pinMode(power, OUTPUT);   // define pin
either OUTPUT or INPUT
  pinMode(auto, OUTPUT);     // define pin
either OUTPUT or INPUT
  pinMode(manual, OUTPUT);  // define pin
either OUTPUT or INPUT
  pinMode(fan1, OUTPUT);     // define pin
either OUTPUT or INPUT
  pinMode(fan2, OUTPUT);     // define pin
either OUTPUT or INPUT
  pinMode(fan3, OUTPUT);     // define pin
either OUTPUT or INPUT
  pinMode(SW_PIN, INPUT);    // define pin
either OUTPUT or INPUT
  digitalWrite(RELAY1, 1);   // connected to
NO of RELAY, "1" means active low, clear
status.
  digitalWrite(RELAY2, 1);   // connected to
NO of RELAY, "1" means active low, clear
status.
  digitalWrite(RELAY3, 1);   // connected to
NO of RELAY, "1" means active low, clear
status.
  digitalWrite(RELAY4, 1);   // connected to
NO of RELAY, "1" means active low, clear
status.
// end of defining to TFT
  Ethernet.begin(mac, ip, gateway, subnet);
  server.begin ( );
  Serial.print("server is at ");
  Serial.println ( Ethernet.localIP ( ) ); //fine
youe ip address on serial mornitor
  EthernetClient client;
  digitalWrite (resetPin, HIGH);
  delay (200);
  pinMode (resetPin, OUTPUT);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  pinMode (led, OUTPUT);
  delay(200);
}
void loop( )
{
    if (digitalRead(SW_PIN) == LOW)
          {
               if (myTouch.dataAvailable( )
== true )
                    {
                        touchscreen  ( );
                    }
          }

    if (digitalRead(SW_PIN) == HIGH)
          {
              internet( );
          }
}
int touchscreen( )
{
//defining initial value for character
          static unsigned char ledState =
LOW;
          static unsigned char buttonStateOn
= LOW;
          static unsigned char
lastButtonState = LOW;
          static unsigned long ledCameOn =
0;
          static unsigned char Autopower =
LOW;
          static unsigned char internalrelay =
LOW;
          static unsigned char startrelay3 =
LOW;
          static unsigned char Manualpower
= LOW;
          static unsigned char Supply = LOW;
          static unsigned char firststatus =
LOW ;
          static unsigned char swapmtoa =
LOW;
          static unsigned char afterclose =
LOW;
// reading value, temperature and humidity,
from DHT11  keep in integer h and t
               int h = dht.readHumidity( );
               int t = dht.readTemperature (
);
               DateTime now = RTC.now( );

     pressed_button =
myButtons.checkButtons( ); // used to check
buttons are pressed or not
     if (pressed_button==but8)
     {
          if
(myButtons.buttonEnabled(but1) &&
myButtons.buttonEnabled(but2)  )
          {

myButtons.disableButton(but1, true);

myButtons.disableButton(but2, true);

          }
     else
     {
          myButtons.enableButton(but1,
true);
          myButtons.enableButton(but2,
true);
     }
   }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if (pressed_button==but9)
            {
                if
(myButtons.buttonEnabled(but6) &&
myButtons.buttonEnabled(but7) )
 myButtons.disableButton(but6, true);
myButtons.disableButton(but7, true);
                }
            else
                {
myButtons.enableButton(but6, true);
myButtons.enableButton(but7, true);
                }
            }
 if (pressed_button==but1 ||
pressed_button==but10 &&
myButtons.buttonEnabled(but10) )
        {
         myButtons.disableButton(but10,
true); // choose auto = disable
         myButtons.enableButton(but11,
true); // manual enable
         myButtons.disableButton(but1, true);
// disable open
         myButtons.enableButton(but2, true);
// enable close
         myButtons.disableButton(but3, true);
// disable fan1
         myButtons.disableButton(but4, true);
// disable fan2
         myButtons.disableButton(but5, true);
// disable fan3
         myButtons.enableButton(but6, true);
// enable +
         myButtons.enableButton(but7, true);
// enable -
        }
 else if (pressed_button==but11 &&
myButtons.buttonEnabled(but11))
        {
          myButtons.disableButton(but11,
true);  // choose manual
 myButtons.enableButton(but10, true);  //
auto enable
          myButtons.disableButton(but1,
true);  // disable open
          myButtons.enableButton(but2,
true);  // enable close
          myButtons.enableButton(but3,
true);  // enable fan1
          myButtons.enableButton(but4,
true);  // enable fan2
          myButtons.enableButton(but5,
true);  // enable fan3
          myButtons.enableButton(but6,
true);  // enable +
          myButtons.enableButton(but7,
true);  // enable -
        }
    else if (pressed_button==but2)
        {
          myButtons.enableButton(but1,
true);
          myButtons.enableButton(but3,
true);
          myButtons.enableButton(but4,
true);
          myButtons.enableButton(but5,
true);
          myButtons.enableButton(but6,
true);
          myButtons.enableButton(but7,
true);
          myButtons.enableButton(but10,
true);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            myButtons.enableButton(but11,
true);
        }
      if  (pressed_button==but1)
       {
            myGLCD.print("OPEN", 160, 220);
            myButtons.disableButton(but1,
true);
            internalrelay =  HIGH;
             Supply = HIGH;
             analogWrite(power, 255);
         }
            buttonStateOn = internalrelay ;
            if (buttonStateOn !=
lastButtonState)
            {
                lastButtonState =
buttonStateOn;
                if ((buttonStateOn == HIGH)
&& (ledState == LOW))
                    {
                      startrelay3 = HIGH;
                       ledState = HIGH;
                       ledCameOn = millis( );
// the maximum value it can hold is
4,294,967,295
                    }
            }
 // If the LED has been on for at least 5
seconds  or more than then turn it off.
          If (ledState == HIGH)
            {
              If (millis()-ledCameOn >
5000)
                {
                  startrelay3 = LOW;
                  Autopower = HIGH;
                  ledState = LOW;
                  firststatus = HIGH;
                }
              }
if (startrelay3 == HIGH && Autopower ==
LOW && firststatus == LOW && swapmtoa
== LOW)
        {
            digitalWrite(RELAY3, LOW);
            digitalWrite(RELAY1, HIGH);
            digitalWrite(RELAY2, HIGH);
            digitalWrite(RELAY4, LOW)
        }
if (pressed_button==but2)
  {
    myGLCD.print("SHUT", 160,
220);
    Manualpower = LOW;
    Autopower = LOW;
    buttonStateOn = LOW;
    lastButtonState = LOW;
    internalrelay = LOW;
    Supply = LOW;
    startrelay3 = LOW;
    firststatus == LOW;
    ledState = LOW;
    ledCameOn = 0;
    swapmtoa = LOW;
    afterclose = HIGH;
digitalWrite(RELAY1, HIGH);
digitalWrite(RELAY2, HIGH);
digitalWrite(RELAY3, HIGH);
digitalWrite(RELAY4, HIGH);
analogWrite(power, 0);
analogWrite(auto, 0);
analogWrite(manual, 0);
analogWrite(fan1, 0);
 analogWrite(fan2, 0);
 analogWrite(fan3, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            delay(10);  //for  reset
            digitalWrite(led, HIGH);  // for reset
            delay(1000);  //reset
            digitalWrite(led, LOW); //reset
           delay(1000); //reset
           Serial.println("resetting"); //reset
            delay(10); // reset
           digitalWrite(resetPin, LOW); // reset
              }
 if ((startrelay3 == LOW && Autopower ==
HIGH ) || (pressed_button==but10 &&
Manualpower == HIGH))
    {
                  myGLCD.print("Auto", 160,
220);
                  Autopower = HIGH;
                  Manualpower = LOW;
                  startrelay3 = LOW;
                  AutoMode();
                  analogWrite(auto, 255);
                  analogWrite(manual, 0);
                 }
 if  ( (pressed_button==but11 && Supply ==
HIGH && startrelay3 == LOW ) ||
(pressed_button==but11
     && Supply == HIGH && startrelay3 ==
HIGH ) )
        {
                 myGLCD.print("Manu", 160,
220);
                 Autopower = LOW;
                 Manualpower = HIGH;
                 startrelay3 = HIGH;
                 analogWrite(manual, 255);
                 analogWrite(auto, 0);
        }
    if (pressed_button==but3  &&
Autopower == LOW && Manualpower ==
HIGH  )
          {
                 myGLCD.print("FAN1", 160,
220);
                 digitalWrite(RELAY1, LOW);
                 digitalWrite(RELAY2, HIGH);
                 digitalWrite(RELAY3, HIGH);
                 digitalWrite(RELAY4, HIGH);
                 analogWrite(fan1, 255);
                 analogWrite(fan2, 0);
                 analogWrite(fan3, 0);
          }
    if (pressed_button==but4 && Autopower
== LOW && Manualpower == HIGH )
          {
                 myGLCD.print("FAN2", 160,
220);
                 digitalWrite(RELAY2, LOW);
                 digitalWrite(RELAY1, HIGH);
                 digitalWrite(RELAY3, HIGH);
                 digitalWrite(RELAY4, LOW);
                 analogWrite(fan1, 0);
                 analogWrite(fan2, 255);
                 analogWrite(fan3, 0);
          }
    if (pressed_button==but5 && Autopower
== LOW && Manualpower == HIGH )
          {
                 myGLCD.print("FAN3", 160,
220);
                 digitalWrite(RELAY3, LOW);
                 digitalWrite(RELAY1, HIGH);
                 digitalWrite(RELAY2, HIGH);
                 digitalWrite(RELAY4, LOW);
                 analogWrite(fan1, 0);
                 analogWrite(fan2, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    analogWrite(fan3, 255);
                }
  if ( Manualpower == HIGH && t < sp )
        }
                digitalWrite(RELAY4, HIGH);
        }
   if (Manualpower == HIGH && t > sp )
        {
                digitalWrite(RELAY4, LOW);
         }
   if (pressed_button==but6 && sp <= 29)
        {
                Incrsetpoint( );
                myGLCD.printNumI(sp, 230,
150);
         }
    else if (pressed_button==but7 && sp >=
21)
         {
                Decrsetpoint( );
                myGLCD.printNumI(sp, 230,
150);
         }
    if (pressed_button==-1)
          myGLCD.print("None  ", 160, 220);
    if (pressed_button==but6)
          myGLCD.print("Incr", 160, 220);
   if (pressed_button==but7)
          myGLCD.print("Decr", 160, 220);
}
int AutoMode ( )
{
          int h = dht.readHumidity( );
          int t = dht.readTemperature ( );
          static unsigned char startrelay3 =
LOW ;
          static unsigned char Autopower =
HIGH ;
    if ( t <= sp && startrelay3 == LOW &&
Autopower == HIGH  )  // RELAY1 turn on
        {
          digitalWrite(RELAY1, LOW);
          digitalWrite(RELAY2, HIGH);
          digitalWrite(RELAY3, HIGH);
          digitalWrite(RELAY4, HIGH);
          analogWrite(fan1, 255);
          analogWrite(fan2, 0);
          analogWrite(fan3, 0);
        }
    else if ( ( t - sp >= 1 && t - sp <= 2 &&
startrelay3 == LOW && Autopower == HIGH )
        )
        {
          digitalWrite(RELAY2, LOW);
          digitalWrite(RELAY1, HIGH);
          digitalWrite(RELAY3, HIGH);
          digitalWrite(RELAY4, LOW);
          analogWrite(fan1, 0);
          analogWrite(fan2, 255);
          analogWrite(fan3, 0);
        }
    else if ((t - sp > 2 && startrelay3 ==
LOW && Autopower == HIGH) )
          {
          digitalWrite(RELAY3, LOW);
          digitalWrite(RELAY1, HIGH);
          digitalWrite(RELAY2, HIGH);
          digitalWrite(RELAY4, LOW);
          analogWrite(fan1, 0);
          analogWrite(fan2, 0);
          analogWrite(fan3, 255);
          }
}
int internet( )
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        static unsigned char ledState =
LOW;
        static unsigned char buttonStateOn
= LOW;
        static unsigned char
lastButtonState = LOW;
        static unsigned long ledCameOn =
0;
        static unsigned char Autopower =
LOW;
        static unsigned char internalrelay =
LOW;
        static unsigned char startrelay3 =
LOW;
        static unsigned char Manualpower
= LOW;
        static unsigned char Supply = LOW;
        static unsigned char firststatus =
LOW ;
        static unsigned char swapmtoa =
LOW;
        static unsigned char afterclose =
LOW;
        int h = dht.readHumidity( );
        int t = dht.readTemperature( );
        DateTime now = RTC.now( );
    EthernetClient client = server.available(
);
      if (client) {
      while (client.connected(  ) ) {
      if (client.available( ) ) {
     char c = client.read( );
     if (readString.length() < 100)
       {
          //store characters to string
        readString += c;
        }
        //if HTTP request has ended
            if (c == '\n') {
              Serial.println(readString); //print
to serial monitor for debuging
              client.println("HTTP/1.1 200 OK");
//send new page
              client.println ("Content-Type:
text/html");
              client.println( );
              client.println ("<HTML>");
              client.println ("<HEAD>");
              client.println ("<meta
name='apple-mobile-web-app-capable'
content='yes' />");
              client.println ("<meta
name='apple-mobile-web-app-status-bar-
style' content='black-
                                translucent' />");
              client.println("<link rel='stylesheet'
type='text/css' href = 'http: //
randomnerdtutorials .com
                                /ethernetcss.css'
/>");
              client.println("<TITLE>Air Controller
PALALOID</TITLE>");
              client.println("</HEAD>");
              client.println("<BODY>");
              client.println("<H1>Air-Conditioner
Controller</H1>");
              client.println("<hr />");
              client.println("<br />");
              client.println("<H2> Ethernet
Switching</H2>");
              client.println("<br />");
              client.println("<br />");
              client.println("<br />");
              client.println("<a
href=\"/?Open\"\">Open</a>");  // button
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            client.println("<a
href=\"/?Close\"\">Close</a><br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<a
href=\"/?AutoMode\"\">Auto Mode</a>");
            client.println("<a
href=\"/?ManualMode\"\">Manual
Mode</a><br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<a
href=\"/?button1on\"\">Turn On LED1</a>");
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<a
href=\"/?button2on\"\">Turn On LED2</a>");
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<a
href=\"/?button3on\"\">Turn On LED3</a>");
            client.println("<br />");
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to open >>>
Hour : "));
            client.println("<a
href=\"/?inchouropen\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(ho);
            client.println(" ");
            client.println("<a
href=\"/?dechouropen\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
href=\"/?incminopen\"\"> + </a>\t\t\n");
            client.print(mo);
            client.println("<a
href=\"/?decminopen\"\"> - </a>\t\t\n");
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to close >>>
Hour : "));
            client.println("<a
href=\"/?inchourclose\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(hc);
            client.println(" ");
            client.println("<a
href=\"/?dechourclose\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
href=\"/?incminclose\"\"> + </a>\t\t\n");
            client.print(mc);
            client.println("<a
href=\"/?decminclose\"\"> - </a>\t\t\n");
            client.println("<br />");
            client.println("<br />");
            client.print(F("<tr>\n"));
            client.print(F("<td><h4>"));
            client.print(F(" Time : \n "));
            client.print(F("</h4></td>"));
            client.print(F("<td></td>"));
            client.print(F("<td>"));
            client.print(F("<h3>"));
            client.print(now.year(), DEC);
            client.print('/');
            client.print(now.month(), DEC);
            client.print('/');
            client.print(now.day(), DEC);
            client.print(' ');
            client.print(now.hour(), DEC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            client.print(':');
            client.print(now.minute(), DEC);
            client.print(F("<td><h4>"));
            client.print(F("Setpoint Temperature
\n "));
            client.println("<a
href=\"/?incsetpoint\"\"> + </a>\t\t");
            client.print(sp);
            client.println("<a
href=\"/?decsetpoint\"\"> - </a><br />");
            client.print(F("<tr>\n"));
            client.print(F("<td><h4>"));
            client.print(F("Temperature"));
            client.print(F("</h4></td>"));
            client.print(F("<td></td>"));
            client.print(F("<td>"));
            client.print(F("<h3>"));
            client.print(t);
            client.print(F(" C </h3></td>\n"));
            client.print(F("<td></td>"));
            client.print(F("</tr>\n"));
            client.print(F("<tr>\n"));
            client.print(F("<td><h4>"));
            client.print(F("Humidity"));
            client.print(F("</h4></td>"));
            client.print(F("<td></td>"));
            client.print(F("<td>"));
            client.print(F("<h3>"));
            client.print(h);
            client.print(F(" % </h3></td>"));
            client.print(F("<td></td>"));
            client.print(F("</tr>"));
            client.print(F("<meta http-
equiv=\"refresh\" content=\""));
            client.print(refreshPage);
            client.println(F("; url=/\">"));
            delay(1);
           //stopping client
            client.stop( );
             if
(readString.indexOf("?incsetpoint")>0 && sp
<= 29)
                  {
                    Incrsetpoint( );
                  }
          else if
(readString.indexOf("?decsetpoint")>0 && sp
>= 21)
                  {
                    Decrsetpoint( );
                  }
          ///set time .....time to open---hour
          else  if
(readString.indexOf("?inchouropen")>0 && ho
<= 24)
                  {
                    ho++;
                    if
(readString.indexOf("?inchouropen")>0 && ho
== 24)
                      {
                        ho = 0;
                      }
                  }
           else if
(readString.indexOf("?dechouropen")>0 &&
ho >= -1)
                {
                  ho--;
                  if
(readString.indexOf("?dechouropen")>0 &&
ho == -1)
                    {
                      ho = 23;
                    }
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        // set time.....time to open----minute
        else  if
(readString.indexOf("?incminopen")>0 && mo
<= 50)
                {
                        mo = mo+10;
                       if
(readString.indexOf("?incminopen")>0 && mo
== 60)
                        {
                           mo = 0;
                        }
                }
        else if
(readString.indexOf("?decminopen")>0 &&
mo >= -10)
                {
                    mo =  mo-10;
                    if
(readString.indexOf("?decminopen")>0 &&
mo == -10)
                    {
                        mo = 50;
                    }
                }
    // set time.....time to close----hour
      else  if
(readString.indexOf("?inchourclose")>0 && hc
<= 24)
                {
                    hc++;
                    if
(readString.indexOf("?inchourclose")>0 && hc
== 24)
                    {
                       hc = 0;
                    }
                }
        else if
(readString.indexOf("?dechourclose")>0 &&
hc >= -1)
                {
                    hc--;
                   if
(readString.indexOf("?dechourclose")>0 &&
hc == -1)
                    {
                        hc = 23;
                    }
                }
        //set time..... time to close ----minute
            else  if
(readString.indexOf("?incminclose")>0 && mc
<= 50)
                {
                    mc = mc+10;
                    if
(readString.indexOf("?incminclose")>0 && mc
== 60)
                    {
                       mc = 0;
                    }
                }
            else if
(readString.indexOf("?decminclose")>0 && mc
>= -10)
                {
                    mc =  mc-10;
                    if
(readString.indexOf("?decminclose")>0 && mc
== -10)
                    {
                        mc = 50;
                    }
                }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      if  (readString.indexOf("?Open")>0 || (
now.hour()== ho && now.minute() == mo ))
                {
                      analogWrite(power, 255);
                      internalrelay =  HIGH;
                      Supply = HIGH;
                 }
                       buttonStateOn =
internalrelay ;
                    if (buttonStateOn !=
lastButtonState)
                {
                      lastButtonState =
buttonStateOn;
                      if ((buttonStateOn ==
HIGH) && (ledState == LOW))
                        {
                          startrelay3 = HIGH;
                         ledState = HIGH;
                         ledCameOn =
millis();
                        }
                 }
 ///// If the LED has been on for at least 5
seconds or more than then turn it off.
        If (ledState == HIGH)
                {
                    If (millis()-ledCameOn >
5000)
                    {
                       startrelay3 = LOW;
                       Autopower = HIGH;
                        ledState = LOW;
                       firststatus = HIGH;
                     }
                }
                    if (startrelay3 == HIGH &&
Autopower == LOW && firststatus == LOW
&& swapmtoa == LOW)
                    {
                        digitalWrite(RELAY3, LOW);
                        digitalWrite(RELAY1, HIGH);
                        digitalWrite(RELAY2, HIGH);
                        digitalWrite(RELAY4, LOW);
                    }
      if (readString.indexOf("?Close")>0 || (
now.hour()== hc && now.minute() == mc ) )
                    {
                      Manualpower = LOW;
                      Autopower = LOW;
                      buttonStateOn = LOW;
                      lastButtonState = LOW;
                      internalrelay = LOW;
                      Supply = LOW;
                      startrelay3 = LOW;
                      firststatus = LOW;
                      ledState = LOW;
                      ledCameOn = 0;
                      swapmtoa = LOW;
                      afterclose = HIGH;

                      unsigned long currentMillis =
millis( );
                  if ((unsigned long)(currentMillis -
previousMillis) >= interval)
                        {
                            previousMillis =
currentMillis;
                            ledCameOn = 0;
                        }
                  digitalWrite(RELAY1, HIGH);
                  digitalWrite(RELAY2, HIGH);
                  digitalWrite(RELAY3, HIGH);
                  digitalWrite(RELAY4, HIGH);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                analogWrite(power, 0);
                analogWrite(auto, 0);
                analogWrite(manual, 0);
                analogWrite(fan1, 0);
                analogWrite(fan2, 0);
                analogWrite(fan3, 0);
                delay(10); // reset
                digitalWrite(led, HIGH);  //reset
                delay(1000);  //reset
                digitalWrite(led, LOW); //reset
               delay(1000); //reset
                Serial.println("resetting");
//reset
                delay(10); // reset
                digitalWrite(resetPin, LOW);
// reset
            }
  if ((startrelay3 == LOW && Autopower ==
HIGH ) || (readString.indexOf("?AutoMode")>0
&&
     Manualpower == HIGH   ))
            {
                analogWrite(auto, 255);
                analogWrite(manual, 0);
                Autopower = HIGH;
                Manualpower = LOW;
                startrelay3 = LOW;
                swapmtoa = HIGH;
                AutoMode( );
            }
            if  ( (
readString.indexOf("?ManualMode")>0 &&
Supply == HIGH && startrelay3 == LOW ) ||
     (readString.indexOf("?ManualMode")>0
&& Supply == HIGH && startrelay3 == HIGH )
)
             {
                  Autopower = LOW;
                  Manualpower = HIGH;
                  startrelay3 = HIGH;
                  analogWrite(manual, 255);
                  analogWrite(auto, 0);
             }
     if (readString.indexOf("?button1on") >0
&& Autopower == LOW && Manualpower ==
HIGH )
          {
                  digitalWrite(RELAY1,
LOW);
                  digitalWrite(RELAY2,
HIGH);
                  digitalWrite(RELAY3,
HIGH);
                  digitalWrite(RELAY4,
HIGH);
                  analogWrite(fan1,
255);
                  analogWrite(fan2, 0);
                  analogWrite(fan3, 0);
          }
     if (readString.indexOf("?button2on")>0
&& Autopower == LOW && Manualpower ==
HIGH)
          {
                  digitalWrite(RELAY1,
HIGH);
                  digitalWrite(RELAY2,
LOW);
                  digitalWrite(RELAY3,
HIGH);
                  digitalWrite(RELAY4,
LOW);
                  analogWrite(fan1, 0);
                  analogWrite(fan2,
255);
                  analogWrite(fan3, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                }                                                    }
        if (readString.indexOf("?button3on")>0
&& Autopower == LOW && Manualpower ==
HIGH)
                {
                        digitalWrite(RELAY3,
LOW);
                        digitalWrite(RELAY1,
HIGH);
                        digitalWrite(RELAY2,
HIGH);
                        digitalWrite(RELAY4,
LOW);
                        analogWrite(fan1, 0);
                        analogWrite(fan2, 0);
                        analogWrite(fan3, 255);
                }
      if ( Manualpower == HIGH && t < sp )
            {
                digitalWrite(RELAY4, HIGH);
            }
       if (Manualpower == HIGH && t > sp )
            {
                digitalWrite(RELAY4, LOW);
            }
      readString="";
            }
       }
      }
   }
}
int Incrsetpoint( )
  {
      sp++;
   }
int Decrsetpoint( )
 {
      sp--;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก2.

โปรแกรมระบบควบคุมเครื่องปรับอากาศแบบภาระโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <SPI.h> // library for Ethernet
#include <Ethernet.h> // library for Ethernet
#include <UTFT.h> //library for TFT
#include <UTouch.h> //library for TFT
#include <UTFT_Buttons.h> // library for TFT
#include <Wire.h> //library for clock
#include "RTClib.h"  // library for clock
RTC_DS1307 RTC;  // library for clock
#define RELAY1 14 // pin for relay
#define RELAY2 15 // pin for relay
#define RELAY3 16 // pin for relay
#define RELAY4 17 // pin for relay
#define RELAY5 19 // pin for relay
#define powerA1 A1  // Air no.1  status pin
#define powerA2 A2  // Air no.2 status pin
#define powerA3 A3  // Air no.3 status pin
#define powerA4 A8  // Air no.4 status pin
#define powerA5 A9 // Air no.5 status pin
// Connect the pole of the switch to pin 18
const int SW_PIN = 18;
EthernetClient client;
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
0xED };   //physical mac address : do not chang
byte ip[] = { 161, 246, 20, 64 };
// ip in lan (that's what you need to use in
//your browser. ("161.246.20.64")
byte gateway[] = { 161, 246, 20, 143 };
 // internet access via router
byte subnet[] = { 255, 255, 255, 0 };
//subnet mask
EthernetServer server(80);
//server port
String readString;
int refreshPage = 5;
int ho1 = 0; // time to open : initial value : hour
: Air1
int mo1 = 0; // time to open : initial value :
minute : Air1
int hc1 = 0; // time to close : initial value : hour :
Air1
int mc1 = 0; // time to close : initial value :
minute  : Air1
int ho2 = 0; // time to open : initial value : hour
: Air2
int mo2 = 0; // time to open : initial value :
minute : Air2
int hc2 = 0; // time to close : initial value : hour:
Air2
int mc2 = 0; // time to close : initial value :
minute : Air2
int ho3 = 0; // time to open : initial value : hour
: Air3
int mo3 = 0; // time to open : initial value :
minute : Air3
int hc3 = 0; // time to close : initial value : hour :
Air3
int mc3 = 0; // time to close : initial value :
minute : Air3
int ho4 = 0; // time to open : initial value : hour
: Air4
int mo4 = 0; // time to open : initial value :
minute  : Air4
int hc4 = 0; // time to close : initial value : hour
: Air4
int mc4 = 0; // time to close : initial value :
minute  : Air4
int ho5 = 0; // time to open : initial value : hour
: Air5
int mo5 = 0; // time to open : initial value :
minute  : Air5
int hc5 = 0; // time to close : initial value : hour
: Air5
int mc5 = 0; // time to close : initial value :
minute  : Air5
int but1, but2, but3, but4, but5, but6, but7,
but8, but9, but10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Define buttons to show on TFT
// Declare which fonts we will be using
extern uint8_t SmallFont[];
extern uint8_t BigFont[];
//extern uint8_t SevenSegNumFont // too big
extern uint8_t Dingbats1_XL[];
// Set up UTFT...
// Set the pins to the correct ones for your
development board
// ---------------------------------------------
// Standard Arduino 2009/Uno/Leonardo shield
: <display model>,19,18,17,16
// Standard Arduino Mega/Due shield          :
//<display model>,38,39,40,41
// CTE TFT LCD/SD Shield for Arduino Due     :
//<display model>,25,26,27,28
// Standard chipKit Uno32/uC32               :
//<display model>,34,35,36,37
// Standard chipKit Max32                    :
//<display model>,82,83,84,85
// AquaLEDSource All in One Super Screw Shield
//: <display model>,82,83,84,85
// Remember to change the model parameter
//to suit your display module!
UTFT          myGLCD(ITDB32S,38,39,40,41);
// Set up UTouch...
// Set the pins to the correct ones for your
//development board
// ---------------------------------------------
// Standard Arduino 2009/Uno/Leonardo shield
//: 15,10,14,9,8
// Standard Arduino Mega/Due shield          :
//6,5,4,3,2
// CTE TFT LCD/SD Shield for Arduino Due     :
//6,5,4,3,2
// Standard chipKit Uno32/uC32               :
//20,21,22,23,24
// Standard chipKit Max32                    :
//62,63,64,65,66
// AquaLEDSource All in One Super Screw Shield
//: 62,63,64,65,66
UTouch        myTouch(6,5,4,3,2);
// Finally we set up UTFT_Buttons :)
UTFT_Buttons  myButtons(&myGLCD,
&myTouch);
void setup( )
{
  // Open serial communications and wait for
port to open:
    Serial.begin(9600);
    Wire.begin( );
    RTC.begin();
    if (! RTC.isrunning())
    {
    Serial.println("RTC is NOT running!");
    // following line sets the RTC to the date &
  time this sketch was compiled
    RTC.adjust(DateTime(2015, 4, 28, 21, 9, 0));
    }
    myGLCD.InitLCD( );
    myGLCD.clrScr( );
    myGLCD.setFont(SmallFont);
    myTouch.InitTouch( );
    myTouch.setPrecision(PREC_MEDIUM);
    myButtons.setTextFont(BigFont);
   myButtons.setSymbolFont(Dingbats1_XL);
    myButtons.setButtonColors(VGA_YELLOW,
VGA_GRAY, VGA_WHITE, VGA_RED, VGA_BLUE);
   // (text and symbol, text and symbol when
//disable, boarder, boarder when select, back
ground)
  but1 = myButtons.addButton( 55,  30, 80,  30,
"OPEN");          // for Air1
  but2 = myButtons.addButton( 145, 30, 80,  30,
"SHUT");          // for Air1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  but3 = myButtons.addButton( 55, 70, 80,  30,
"OPEN");          // for Air2
  but4 = myButtons.addButton( 145, 70, 80,  30,
"SHUT");          //for Air2
  but5 = myButtons.addButton( 55, 110, 80,  30,
"OPEN");          //for Air3
  but6 = myButtons.addButton(145, 110, 80, 30,
"SHUT");          //for Air3
  but7 = myButtons.addButton( 55, 150, 80,  30,
"OPEN");          // for Air4
  but8 = myButtons.addButton( 145, 150, 80,  30,
"SHUT");       // for Air4
  but9 = myButtons.addButton( 55, 190, 80,  30,
"OPEN");          // for Air5
  but10 = myButtons.addButton( 145,  190, 80,
30, "SHUT");     // for Air5
  myButtons.drawButtons( );
  {
     myGLCD.print("Status:", 250, 10);
     myGLCD.print("Air.1", 10, 40);
     myGLCD.print("Air.2", 10, 80);
     myGLCD.print("Air.3", 10, 120);
    myGLCD.print("Air.4", 10, 160);
    myGLCD.print("Air.5", 10, 200);
    myGLCD.setColor(VGA_BLACK);
    myGLCD.setBackColor(VGA_WHITE);
 }
 {
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_GREEN);
    myGLCD.setBackColor(VGA_YELLOW);
 }
 // Set the pin modes
    pinMode(RELAY1, OUTPUT);
    pinMode(RELAY2, OUTPUT);
    pinMode(RELAY3, OUTPUT);
    pinMode(RELAY4, OUTPUT);
    pinMode(RELAY5, OUTPUT);
    pinMode(powerA1, OUTPUT);
    pinMode(powerA2, OUTPUT);
    pinMode(powerA3, OUTPUT);
    pinMode(powerA4, OUTPUT);
    pinMode(powerA5, OUTPUT);
    pinMode(SW_PIN, INPUT);
    digitalWrite(RELAY1, 1);
    digitalWrite(RELAY2, 1);
    digitalWrite(RELAY3, 1);
    digitalWrite(RELAY4, 1);
    digitalWrite(RELAY5, 1);
     analogWrite(powerA1, 0);
     analogWrite(powerA2, 0);
     analogWrite(powerA3, 0);
     analogWrite(powerA4, 0);
     analogWrite(powerA5, 0);
    // start the Ethernet connection and the
    //server:
    Ethernet.begin(mac, ip, gateway, subnet);
    server.begin( );
    Serial.print("server is at ");
    Serial.println(Ethernet.localIP());  //find your ip
   //address on serial monitor
   EthernetClient client;
  }
  void loop()
  {
    if (digitalRead(SW_PIN) == LOW)
   {
    if (myTouch.dataAvailable() == true )
      {
        touchscreen ( );
      }
   }
   if (digitalRead(SW_PIN) == HIGH)
      {
        internet( );
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
int touchscreen( )
{
     pressed_button = myButtons.checkButtons(
); //check if any buttons are pressed
      if (pressed_button==but1 &&
myButtons.buttonEnabled(but1))
       {
            myButtons.disableButton(but1, true);
       }
      if (pressed_button==but2)
       {
            myButtons.enableButton(but1, true);
       }
     if (pressed_button==but3 &&
myButtons.buttonEnabled(but3))
      {
           myButtons.disableButton(but3, true);
      }
     if (pressed_button==but4)
      {
           myButtons.enableButton(but3, true);
      }
    if (pressed_button==but5 &&
myButtons.buttonEnabled(but5))
      {
           myButtons.disableButton(but5, true);
      }
    if (pressed_button==but6)
      {
           myButtons.enableButton(but5, true);
      }
    if (pressed_button==but7 &&
myButtons.buttonEnabled(but7))
      {
           myButtons.disableButton(but7, true);
      }
    if (pressed_button==but8)
      {
           myButtons.enableButton(but7, true);
      }
   if (pressed_button==but9 &&
myButtons.buttonEnabled(but9))
    {
         myButtons.disableButton(but9, true);
    }
   if (pressed_button==but10)
    {
         myButtons.enableButton(but9, true);
    }
    if  (pressed_button==but1)
      {
        myGLCD.print("OPEN", 240, 40);
        digitalWrite(RELAY1, LOW);
        analogWrite(powerA1, 255);
      }
    if (pressed_button==but2)
     {
        myGLCD.print("SHUT", 240, 40);
        digitalWrite(RELAY1, HIGH);
        analogWrite(powerA1, 0);
     }
    if  (pressed_button==but3)
     {
        myGLCD.print("OPEN", 240, 80);
        digitalWrite(RELAY2, LOW);
        analogWrite(powerA2, 255);
     }
   if (pressed_button==but4)
     {
        myGLCD.print("SHUT", 240, 80);
        digitalWrite(RELAY2, HIGH);
        analogWrite(powerA2, 0);
     }
    if  (pressed_button==but5)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            myGLCD.print("OPEN", 240, 120);
            digitalWrite(RELAY3, LOW);
            analogWrite(powerA3, 255);
        }
      if (pressed_button==but6)
        {
            myGLCD.print("SHUT", 240, 120);
            digitalWrite(RELAY3, HIGH);
            analogWrite(powerA3, 0);
        }
       if  (pressed_button==but7)
        {
            myGLCD.print("OPEN", 240, 160);
            digitalWrite(RELAY4, LOW);
            analogWrite(powerA4, 255);
        }
      if (pressed_button==but8)
        {
            myGLCD.print("SHUT", 240, 160);
            digitalWrite(RELAY4, HIGH);
            analogWrite(powerA4, 0);
        }
       if  (pressed_button==but9)
        {
            myGLCD.print("OPEN", 240, 200);
            digitalWrite(RELAY5, LOW);
            analogWrite(powerA5, 255);
        }
      if (pressed_button==but10)
        {
            myGLCD.print("SHUT", 240, 200);
            digitalWrite(RELAY5,HIGH);
            analogWrite(powerA5, 0);
        }
}
int internet( )
{
        DateTime now = RTC.now( );
        EthernetClient client = server.available( );
        if (client) {
        while (client.connected( )) {
        if (client.available( )) {
         char c = client.read( );
         //read char by char HTTP request
         if (readString.length() < 100) {
          //store characters to string
          readString += c;
         }
         //if HTTP request has ended
         if (c == '\n') {
          Serial.println(readString); //print to serial
monitor for debuging
          client.println("HTTP/1.1 200 OK");  //send
new page
          client.println("Content-Type: text/html");
          client.println( );
          client.println("<HTML>");
          client.println("<HEAD>");
          client.println("<meta name='apple-
mobile-web-app-capable' content='yes' />");
          client.println("<meta name='apple-
mobile-web-app-status-bar-style' content='black-
translucent' />");
          client.println("<link rel='stylesheet'
type='text/css
href='http://randomnerdtutorials.com/ethernetcs
s.css' />");
          client.println("<TITLE>Air Controller
</TITLE>");
          client.println("</HEAD>");
          client.println("<BODY>");
          client.println("<H1>Air-Conditioner
Controller</H1>");
          client.println("<hr />");
          client.println("<br />");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        client.println("<H2> Ethernet
Switching</H2>");
        //////// Time////////
        client.println("<br />");
        client.print(F("<tr>\n"));
        client.print(F("<td><h4>"));
        client.print(F(" Time : \n "));
        client.print(F("</h4></td>"));
        client.print(F("<td></td>"));
        client.print(F("<td>"));
        client.print(F("<h3>"));
        client.print(now.year( ), DEC);
        client.print('/');
        client.print(now.month( ), DEC);
        client.print('/');
        client.print(now.day( ), DEC);
        client.print(' ');
        client.print(now.hour( ), DEC);
        client.print(':');
        client.print(now.minute(), DEC);
        client.println("<br />");
        client.println("<br />");
        client.println("<br />");
        //////air.1///////
        client.print(F("<tr>\n"));
        client.print(F("<td><h4>"));
        client.print(F("Air no.1 "));
        client.println("<br />");
        client.print(F("<tr>\n"));
        client.print(F("<td><h4>"));  // time to
open // Air 1
        client.print(F("Time to open >>> Hour :
"));  //set time to open
        client.println("<a
href=\"/?inchouropenA1\"\"> + </a>\t\t\n");
        client.println(" ");
        client.print(ho1);
        client.println(" ");
        client.println("<a
href=\"/?dechouropenA1\"\"> - </a>\t\t\n");
        client.print(F("&& minute :  "));
        client.println("<a
href=\"/?incminopenA1\"\"> + </a>\t\t\n");
        client.print(mo1);
        client.println("<a
href=\"/?decminopenA1\"\"> - </a>\t\t\n"); //
Air1
        client.print(F("    or press to   "));
        client.println("<a href=\"/?openA1\"\">
OPEN </a>\t\t\n"); // open button
        client.println("<br />");
        client.println("<br />");
        client.print(F("<td><h4>")); // time to
close Air1
        client.print(F("Time to close >>> Hour :
"));  //set time to open
        client.println("<a
href=\"/?inchourcloseA1\"\"> + </a>\t\t\n");
        client.println(" ");
        client.print(hc1);
        client.println(" ");
        client.println("<a
href=\"/?dechourcloseA1\"\"> - </a>\t\t\n");
        client.print(F("&& minute :  "));
        client.println("<a
href=\"/?incmincloseA1\"\"> + </a>\t\t\n");
        client.print(mc1);
        client.println("<a
href=\"/?decmincloseA1\"\"> - </a>\t\t\n");
        client.print(F("    or press to   "));
        client.println("<a href=\"/?closeA1\"\">
CLOSE </a>\t\t\n");
        /////Air.2///////
        client.println("<br />");
        client.println("<br />");
        client.println("<br />");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            client.println("<br />");
            client.print(F("Air no.2 "));
            client.println("<br />");
            client.print(F("<td><h4>"));  // time to
 open // Air 2
            client.print(F("Time to open >>> Hour :
 "));  //set time to open
            client.println("<a
 href=\"/?inchouropenA2\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(ho2);
            client.println(" ");
            client.println("<a
 href=\"/?dechouropenA2\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
 href=\"/?incminopenA2\"\"> + </a>\t\t\n");
            client.print(mo2);
            client.println("<a
 href=\"/?decminopenA2\"\"> - </a>\t\t\n");
            client.print(F("    or press to   "));
            client.println("<a href=\"/?openA2\"\">
 OPEN </a>\t\t\n");
            client.println("<br />");
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to close >>> Hour :
 "));
            client.println("<a
 href=\"/?inchourcloseA2\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(hc2);
            client.println(" ");
            client.println("<a
 href=\"/?dechourcloseA2\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
 href=\"/?incmincloseA2\"\"> + </a>\t\t\n");
            client.print(mc2);
            client.println("<a
 href=\"/?decmincloseA2\"\"> - </a>\t\t\n");
            client.print(F("    or press to   "));
            client.println("<a href=\"/?closeA2\"\">
 CLOSE </a>\t\t\n");
             /////Air.3////////
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.print(F("Air no.3 "));
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to open >>> Hour :
 "));
            client.println("<a
 href=\"/?inchouropenA3\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(ho3);
            client.println(" ");
            client.println("<a
 href=\"/?dechouropenA3\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
 href=\"/?incminopenA3\"\"> + </a>\t\t\n");
            client.print(mo3);
            client.println("<a
 href=\"/?decminopenA3\"\"> - </a>\t\t\n");
            client.print(F("    or press to   "));
            client.println("<a href=\"/?openA3\"\">
 OPEN </a>\t\t\n");
            client.println("<br />");
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to close >>> Hour :
 "));
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            client.println("<a
 href=\"/?inchourcloseA3\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(hc3);
            client.println(" ");
            client.println("<a
 href=\"/?dechourcloseA3\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
 href=\"/?incmincloseA3\"\"> + </a>\t\t\n");
            client.print(mc3);
            client.println("<a
 href=\"/?decmincloseA3\"\"> - </a>\t\t\n");
            client.print(F("    or press to   "));
            client.println("<a href=\"/?closeA3\"\">
 CLOSE </a>\t\t\n");
             ///////Air.4/////////
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.print(F("Air no.4 "));
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to open >>> Hour :
 "));
            client.println("<a
 href=\"/?inchouropenA4\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(ho4);
            client.println(" ");
            client.println("<a
 href=\"/?dechouropenA4\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
 href=\"/?incminopenA4\"\"> + </a>\t\t\n");
            client.print(mo4);
            client.println("<a
 href=\"/?decminopenA4\"\"> - </a>\t\t\n");
            client.print(F("    or press to   "));
            client.println("<a href=\"/?openA4\"\">
 OPEN </a>\t\t\n");
            client.println("<br />");
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to close >>> Hour :
 "));
            client.println("<a
 href=\"/?inchourcloseA4\"\"> + </a>\t\t\n");
            client.println(" ");
            client.print(hc4);
            client.println(" ");
            client.println("<a
 href=\"/?dechourcloseA4\"\"> - </a>\t\t\n");
            client.print(F("&& minute  :  "));
            client.println("<a
 href=\"/?incmincloseA4\"\"> + </a>\t\t\n");
            client.print(mc4);
            client.println("<a
 href=\"/?decmincloseA4\"\"> - </a>\t\t\n");
            client.print(F("    or press to   "));
            client.println("<a href=\"/?closeA4\"\">
 CLOSE </a>\t\t\n");
            ///////Air.5/////////
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.println("<br />");
            client.print(F("Air no.5 "));
            client.println("<br />");
            client.print(F("<td><h4>"));
            client.print(F("Time to open >>> Hour :
 "));
            client.println("<a
 href=\"/?inchouropenA5\"\"> + </a>\t\t\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        client.println(" ");
        client.print(ho5);
        client.println(" ");
        client.println("<a
href=\"/?dechouropenA5\"\"> - </a>\t\t\n");
        client.print(F("&& minute  :  "));
        client.println("<a
href=\"/?incminopenA5\"\"> + </a>\t\t\n");
        client.print(mo5);
        client.println("<a
href=\"/?decminopenA5\"\"> - </a>\t\t\n");
        client.print(F("    or press to   "));
        client.println("<a href=\"/?openA5\"\">
OPEN </a>\t\t\n");
        client.println("<br />");
        client.println("<br />");
        client.print(F("<td><h4>"));
        client.print(F("Time to close >>> Hour :
"));
        client.println("<a
href=\"/?inchourcloseA5\"\"> + </a>\t\t\n");
        client.println(" ");
        client.print(hc5);
        client.println(" ");
        client.println("<a
href=\"/?dechourcloseA5\"\"> - </a>\t\t\n");
        client.print(F("&& minute  :  "));
        client.println("<a
href=\"/?incmincloseA5\"\"> + </a>\t\t\n");
        client.print(mc5);
        client.println("<a
href=\"/?decmincloseA5\"\"> - </a>\t\t\n");
        client.print(F("    or press to   "));
        client.println("<a href=\"/?closeA5\"\">
CLOSE </a>\t\t\n");
        // add a meta refresh tag, so the
browser pulls again every x seconds:
        client.print(F("<meta http-
equiv=\"refresh\" content=\""));
        client.print(refreshPage);
        client.println(F("; url=/\">"));
        delay(1);
        //stopping client
        client.stop( );
       ///set time A1/// 0pen hour
        if
(readString.indexOf("?inchouropenA1")>0 && ho1
<= 24)  .
                {
                    ho1++;
                if
(readString.indexOf("?inchouropenA1")>0 && ho1
== 24)
                    {
                        ho1 = 0;
                    }
                }
            else if
(readString.indexOf("?dechouropenA1")>0 && ho1
>= -1)
                {
                    ho1--;
                if
(readString.indexOf("?dechouropenA1")>0 && ho1
== -1)
                    {
                      ho1 = 23;
                    }
                }
            else  if
(readString.indexOf("?incminopenA1")>0 && mo1
<= 50)
                {
                    mo1 = mo1+10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            if
(readString.indexOf("?incminopenA1")>0 && mo1
== 60)
                {
                  mo1 = 0;
                }
              }
        else if
(readString.indexOf("?decminopenA1")>0 && mo1
>= -10)
            {
              mo1 =  mo1-10;
          if
(readString.indexOf("?decminopenA1")>0 && mo1
== -10)
                {
                  mo1 = 50;
                }
            }
        //////////  set time/// close hour A1
         else  if
(readString.indexOf("?inchourcloseA1")>0 && hc1
<= 24)
            {
              hc1++;
            if
(readString.indexOf("?inchourcloseA1")>0 && hc1
== 24)
                {
                  hc1 = 0;
                }
            }
        else if
(readString.indexOf("?dechourcloseA1")>0 && hc1
>= -1)
            {
              hc1--;
            if
(readString.indexOf("?dechourcloseA1")>0 && hc1
== -1)
                {
                  hc1 = 23;
                }
            }
        ////////  set time/// close minute
          else  if
(readString.indexOf("?incmincloseA1")>0 && mc1
<= 50)
            {
              mc1 = mc1+10;
            if
(readString.indexOf("?incmincloseA1")>0 && mc1
== 60)
              {
                mc1 = 0;
              }
            }
          else if
(readString.indexOf("?decmincloseA1")>0 && mc1
>= -10)
            {
              mc1 =  mc1-10;
            if
(readString.indexOf("?decmincloseA1")>0 && mc1
== -10)
                {
                  mc1 = 50;
                }
            }
        ///set time A2/// 0pen hour
          else  if
(readString.indexOf("?inchouropenA2")>0 && ho2
<= 24)
            {
              ho2++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if
(readString.indexOf("?inchouropenA2")>0 && ho2
== 24)
                    {
                      ho2 = 0;
                    }
                }
            else if
(readString.indexOf("?dechouropenA2")>0 && ho2
>= -1)
                {
                    ho2--;
                  if
(readString.indexOf("?dechouropenA2")>0 && ho2
== -1)
                    {
                      ho2 = 23;
                    }
                }
              else if
(readString.indexOf("?incminopenA2")>0 && mo2
<= 50)
                {
                    mo2 = mo2+10;
                  if
(readString.indexOf("?incminopenA2")>0 && mo2
== 60)
                    {
                      mo2 = 0;
                    }
                }
            else if
(readString.indexOf("?decminopenA2")>0 && mo2
>= -10)
                {
                  mo2 =  mo2-10;
```

```
                if
(readString.indexOf("?decminopenA2")>0 && mo2
== -10)
                    {
                      mo2 = 50;
                    }
                }
              else  if
(readString.indexOf("?inchourcloseA2")>0 && hc2
<= 24)
                    {
                      hc2++;
                    if
(readString.indexOf("?inchourcloseA2")>0 && hc2
== 24)
                    {
                      hc2 = 0;
                    }
                    }
                else if
(readString.indexOf("?dechourcloseA2")>0 && hc2
>= -1)
                {
                    hc2--;
                  if
(readString.indexOf("?dechourcloseA2")>0 && hc2
== -1)
                    {
                      hc2 = 23;
                    }
                }
              else  if
(readString.indexOf("?incmincloseA2")>0 && mc2
<= 50)
                {
                    mc2 = mc2+10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if
(readString.indexOf("?incmincloseA2")>0 && mc2
== 60)
                    {
                      mc2 = 0;
                    }
                }
            else if
(readString.indexOf("?decmincloseA2")>0 && mc2
>= -10)
                {
                  mc2 =  mc2-10;
                if
(readString.indexOf("?decmincloseA2")>0 && mc2
== -10)
                    {
                      mc2 = 50;
                    }
                }
        else  if
(readString.indexOf("?inchouropenA3")>0 && ho3
<= 24)
                {
                    ho3++;
                  if
(readString.indexOf("?inchouropenA3")>0 && ho3
== 24)
                    {
                      ho3 = 0;
                    }
                }
            else if
(readString.indexOf("?dechouropenA3")>0 && ho3
>= -1)
                {
                    ho3--;
                if
(readString.indexOf("?dechouropenA3")>0 && ho3
== -1)
                    {
                      ho3 = 23;
                    }
                }
            else  if
(readString.indexOf("?incminopenA3")>0 && mo3
<= 50)
                {
                    mo3 = mo3+10;
                if
(readString.indexOf("?incminopenA3")>0 && mo3
== 60)
                    {
                      mo3 = 0;
                    }
                }
            else if
(readString.indexOf("?decminopenA3")>0 && mo3
>= -10)
                {
                  mo3 =  mo3-10;
                if
(readString.indexOf("?decminopenA3")>0 && mo3
== -10)
                    {
                      mo3 = 50;
                    }
                }
            else  if
(readString.indexOf("?inchourcloseA3")>0 && hc3
<= 24)
                {
                    hc3++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if
(readString.indexOf("?inchourcloseA3")>0 && hc3
== 24)
                    {
                      hc3 = 0;
                    }
                }
            else if
(readString.indexOf("?dechourcloseA3")>0 && hc3
>= -1)
                {
                    hc3--;
            if
(readString.indexOf("?dechourcloseA3")>0 && hc3
== -1)
                    {
                      hc3 = 23;
                    }
                }
        else  if
(readString.indexOf("?incmincloseA3")>0 && mc3
<= 50)
                {
                    mc3 = mc3+10;
                  if
(readString.indexOf("?incmincloseA3")>0 && mc3
== 60)
                    {
                      mc3 = 0;
                    }
                }
            else if
(readString.indexOf("?decmincloseA3")>0 && mc3
>= -10)
                {
                  mc3 =  mc3-10;
```

```
                if
(readString.indexOf("?decmincloseA3")>0 && mc3
== -10)
                    {
                      mc3 = 50;
                    }
                }
            else  if
(readString.indexOf("?inchouropenA4")>0 && ho4
<= 24)
                {
                    ho4++;
                  if
(readString.indexOf("?inchouropenA4")>0 && ho4
== 24)
                    {
                      ho4 = 0;
                    }
                }
            else if
(readString.indexOf("?dechouropenA4")>0 && ho4
>= -1)
                {
                  ho4--;
                  if
(readString.indexOf("?dechouropenA4")>0 && ho4
== -1)
                    {
                      ho4 = 23;
                    }
                }
            else  if
(readString.indexOf("?incminopenA4")>0 && mo4
<= 50)
                {
                    mo4 = mo4+10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if
(readString.indexOf("?incminopenA4")>0 && mo4
== 60)
                    {
                      mo4 = 0;
                    }
                }
            else if
(readString.indexOf("?decminopenA4")>0 && mo4
>= -10)
                {
                  mo4 =  mo4-10;
              if
(readString.indexOf("?decminopenA4")>0 && mo4
== -10)
                    {
                      mo4 = 50;
                    }
                }
            else  if
(readString.indexOf("?inchourcloseA4")>0 && hc4
<= 24)
                {
                    hc4++;
                if
(readString.indexOf("?inchourcloseA4")>0 && hc4
== 24)
                    {
                      hc4 = 0;
                    }
                }
            else if
(readString.indexOf("?dechourcloseA4")>0 && hc4
>= -1)
                {
                    hc4--;
                if
(readString.indexOf("?dechourcloseA4")>0 && hc4
== -1)
                    {
                      hc4 = 23;
                    }
                }
            else  if
(readString.indexOf("?incmincloseA4")>0 && mc4
<= 50)
                {
                    mc4 = mc4+10;
                if
(readString.indexOf("?incmincloseA4")>0 && mc4
== 60)
                    {
                      mc4 = 0;
                    }
                }
            else if
(readString.indexOf("?decmincloseA4")>0 && mc4
>= -10)
                {
                  mc4 =  mc4-10;
                if
(readString.indexOf("?decmincloseA4")>0 && mc4
== -10)
                    {
                      mc4 = 50;
                    }
                }
            else  if
(readString.indexOf("?inchouropenA5")>0 && ho5
<= 24)
                {
                    ho5++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if
(readString.indexOf("?inchouropenA5")>0 && ho5
== 24)
                    {
                      ho5 = 0;
                    }
                }
            else if
(readString.indexOf("?dechouropenA5")>0 && ho5
>= -1)
                {
                    ho5--;
                  if
(readString.indexOf("?dechouropenA5")>0 && ho5
== -1)
                    {
                      ho5 = 23;
                    }
                }
            else  if
(readString.indexOf("?incminopenA5")>0 && mo5
<= 50)
                {
                   mo5 = mo5+10;
                  if
(readString.indexOf("?incminopenA5")>0 && mo5
== 60)
                    {
                      mo5 = 0;
                    }
                }
            else if
(readString.indexOf("?decminopenA5")>0 && mo5
>= -10)
                {
                  mo5 =  mo5-10;
                if
(readString.indexOf("?decminopenA5")>0 && mo5
== -10)
                    {
                      mo5 = 50;
                    }
                }
            else  if
(readString.indexOf("?inchourcloseA5")>0 && hc5
<= 24)
                {
                    hc5++;
                  if
(readString.indexOf("?inchourcloseA5")>0 && hc5
== 24)
                    {
                      hc5 = 0;
                    }
                }
            else if
(readString.indexOf("?dechourcloseA5")>0 && hc5
>= -1)
                {
                    hc5--;
                  if
(readString.indexOf("?dechourcloseA5")>0 && hc5
== -1)
                    {
                      hc5 = 23;
                    }
                }
            else  if
(readString.indexOf("?incmincloseA5")>0 && mc5
<= 50)
                {
                   mc5 = mc5+10;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if
(readString.indexOf("?incmincloseA5")>0 && mc5
== 60)
                    {
                      mc5 = 0;
                    }
                }
          else if
(readString.indexOf("?decmincloseA5")>0 && mc5
>= -10)
              {
                mc5 =  mc5-10;
              if
(readString.indexOf("?decmincloseA5")>0 && mc5
== -10)   {
                     mc5 = 50;
                   }
              }
 if  (readString.indexOf("?openA1")>0 || (
now.hour( )== ho1 && now.minute( ) == mo1 ))
                  {
                    digitalWrite(RELAY1, LOW);
                    analogWrite(powerA1, 255);
                  }
    if (readString.indexOf("?closeA1")>0 || (
now.hour( )== hc1 && now.minute( ) == mc1 ) )
          {
            digitalWrite(RELAY1, HIGH);
            analogWrite(powerA1, 0);
          }
  if  (readString.indexOf("?openA2")>0 || (
now.hour( )== ho2 && now.minute( ) == mo2 ))
                  {
                    digitalWrite(RELAY2, LOW);
                    analogWrite(powerA2, 255);
                  }
   if (readString.indexOf("?closeA2")>0 || (
now.hour( )== hc2 && now.minute( ) == mc2 ) )
                  {
                    digitalWrite(RELAY2, HIGH);
                    analogWrite(powerA2, 0);
                  }
    if  (readString.indexOf("?openA3")>0 || (
now.hour( )== ho3 && now.minute( ) == mo3 ))
                  {
                    digitalWrite(RELAY3, LOW);
                    analogWrite(powerA3, 255);
                  }
    if (readString.indexOf("?closeA3")>0 || (
now.hour( )== hc3 && now.minute( ) == mc3 ) )
                  {
                    digitalWrite(RELAY3, HIGH);
                    analogWrite(powerA3, 0);
                  }
    if  (readString.indexOf("?openA4")>0 || (
now.hour( )== ho4 && now.minute( ) == mo4 ))
                  {
                    digitalWrite(RELAY4, LOW);
                    analogWrite(powerA4, 255);
                  }
    if (readString.indexOf("?closeA4")>0 || (
now.hour( )== hc4 && now.minute( ) == mc4 ) )
                  {
                    digitalWrite(RELAY4, HIGH);
                    analogWrite(powerA4, 0);
                  }
if  (readString.indexOf("?openA5")>0 || ( now.hour(
)== ho5 && now.minute( ) == mo5 ))
                  {
                    digitalWrite(RELAY5, LOW);
                    analogWrite(powerA5, 255);
                  }
if (readString.indexOf("?closeA5")>0 || ( now.hour(
)== hc5 && now.minute( ) == mc5 ) )       {
                    digitalWrite(RELAY5, HIGH);
                    analogWrite(powerA5, 0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
    readString="";
            }
       }
    }
 }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTFT

Arduino and chipKit Universal TFT display library

Manual

(C)2014 Rinky-Dink Electronics, Henning Karlsen

Introduction:

This library was originally the continuation of my ITDB02_Graph, ITDB02_Graph16 and RGB_GLCD libraries for Arduino and chipKit. As the number of supported display modules and controllers started to increase I felt it was time to make a single, universal library as it will be much easier to maintain in the future.

Basic functionality of this library was origianlly based on the demo-code provided by ITead studio (for the ITDB02 modules) and NKC Electronics (for the RGB GLCD module/shield).

This library supports a number of 8bit, 16bit and serial graphic displays, and will work with both Arduino and chipKit boards. For a full list of tested display modules and controllers, see the document **UTFT_Supported_display_modules_&_controllers.pdf**.

You can always find the latest version of the library at **http://www.RinkyDinkElectronics.com/**

For version information, please refer to **version.txt**.

IMPORTANT:

When using 8bit and 16bit display modules there are some requirements you must adhere to. These requirements can be found in the document **UTFT_Requirements.pdf**.
There are no special requirements when using serial displays.

Since most people have only one or possibly two different display modules a lot of memory has been wasted to keep support for many unneeded controller chips.
As of v1.1 you now have the option to easily remove this unneeded code from the library. By disabling the controllers you don't need you can reduce the memory footprint of the library by several Kb. For more information, please refer to **memorysaver.h**.

If you are using the **"AquaLEDSource All in One Super Screw Shield"** on a chipKit Max32, please read the comment in **hardware/pic32/HW_PIC32_defines.h**

If you are using the **"CTE TFT LCD/SD Shield for Arduino Due"** or the **"ElecHouse TFT LCD Screen Shield for Arduino DUE /Taijiuino"**, please read the comment in **hardware/arm/HW_ARM_defines.h**

8 bit display shields designed for use on Arduino Uno (and similarly sized boards) can now be used on Arduino Megas. Please read the comment in **hardware/avr/HW_AVR_defines.h**

The 7" display modules have not been tested on the chipKit boards due to the high current requirement for the LED backlight.

This library is licensed under a **CC BY-NC-SA 3.0** (Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported) License.

For more information see: **http://creativecommons.org/licenses/by-nc-sa/3.0/**

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DEFINED LITERALS:

Alignment

For use with print(), printNumI() and printNumF()

```
LEFT:   0
RIGHT:  9999
CENTER: 9998
```

Orientation

For use with InitLCD()

```
PORTRAIT:  0
LANDSCAPE: 1
```

VGA Colors

Predefined colors for use with setColor() and setBackColor()

VGA_BLACK	VGA_SILVER	VGA_GRAY	VGA_WHITE
VGA_MAROON	VGA_RED	VGA_PURPLE	VGA_FUCHSIA
VGA_GREEN	VGA_LIME	VGA_OLIVE	VGA_YELLOW
VGA_NAVY	VGA_BLUE	VGA_TEAL	VGA_AQUA

VGA_TRANSPARENT (only valid for setBackColor())

Display model

For use with UTFT()

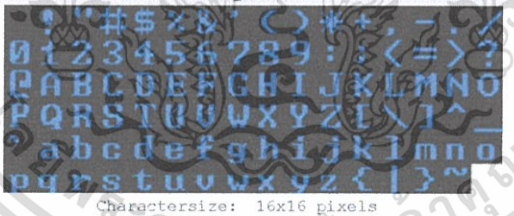
Please see **UTFT_Supported_display_modules_&_controllers.pdf**

INCLUDED FONTS:

SmallFont

```
Charactersize:  8x12 pixels
Number of characters:  95
```

BigFont



```
Charactersize:  16x16 pixels
Number of characters:  95
```

SevenSegNumFont

```
Charactersize:  32x50 pixels
Number of characters:  10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า แสวงหาผลประโยชน์ทางการค้า ทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTIONS:

UTFT(Model, RS, WR, CS, RST[, ALE]);

The main class constructor when using 8bit or 16bit display modules.

Parameters:	Model:	See the separate document for the supported display modules
	RS:	Pin for Register Select
	WR:	Pin for Write
	CS:	Pin for Chip Select
	RST:	Pin for Reset
	ALE:	**<optional>** Only used for latched 16bit shields Pin for Latch signal
Usage:	UTFT myGLCD(ITDB32S,19,18,17,16); // Start an instance of the UTFT class	

UTFT(Model, SDA, SCL, CS, RST[, RS]);

The main class constructor when using serial display modules.

Parameters:	Model:	See the separate document for the supported display modules
	SDA:	Pin for Serial Data
	SCL:	Pin for Serial Clock
	CS:	Pin for Chip Select
	RST:	Pin for Reset
	RS:	**<optional>** Only used for 5pin serial modules Pin for Register Select
Usage:	UTFT myGLCD(ITDB18SP,11,10,9,12,8); // Start an instance of the UTFT class	

InitLCD([orientation]);

Initialize the LCD and set display orientation.

Parameters:	Orientation:	**<optional>** PORTRAIT LANDSCAPE **(default)**
Usage:	myGLCD.initLCD(); // Initialize the display	
Notes:	This will reset color to white with black background. Selected font will be reset to *none*.	

getDisplayXSize();

Get the width of the screen in the current orientation.

Parameters:	None
Returns:	Width of the screen in the current orientation in pixels
Usage:	Xsize = myGLCD.getDisplayXSize(); // Get the width

getDisplayYSize();

Get the height of the screen in the current orientation.

Parameters:	None
Returns:	Height of the screen in the current orientation in pixels
Usage:	Ysize = myGLCD.getDisplayYSize(); // Get the height

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

lcdOff();

Turn off the LCD. No commands will be executed until a lcdOn(); is sent.

Parameters:	None
Usage:	myGLCD.lcdOff(); // Turn off the lcd
Notes:	This function is currently only supported on PCF8833 and CPLD-based displays. CPLD-based displays will only turn off the backlight. It will accept further commands/writes.

lcdOn();

Turn on the LCD after issuing a lcdOff()-command.

Parameters:	None
Usage:	myGLCD.lcdOn(); // Turn on the lcd
Notes:	This function is currently only supported on PCF8833 and CPLD-based displays. CPLD-based displays will only turn on the backlight.

setContrast(c);

Set the contrast of the display.

Parameters:	c: Contrast-level (0-64)
Usage:	myGLCD.setContrast(64); // Set contrast to full (default)
Notes:	This function is currently only supported on PCF8833-based displays

setBrightness(br);

Set the brightness of the display backlight.

Parameters:	br: Brightness-level (0-16)
Usage:	myGLCD.setBrightness(16); // Set brightness to maximum (default)
Notes:	This function is currently only supported on CPLD-based displays

setDisplayPage(pg);

Set which memory page to display.

Parameters:	pg: Page (0-7) (0 is default)
Usage:	myGLCD.setDisplayPage(4); // Display page 4
Notes:	This function is currently only supported on CPLD-based displays

setWritePage(pg);

Set which memory page to use for subsequent display writes.

Parameters:	pg: Page (0-7) (0 is default)
Usage:	myGLCD.setWritePage(2); // Use page 2 for subsequent writes
Notes:	This function is currently only supported on CPLD-based displays

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

clrScr();

Clear the screen. The background-color will be set to black.

Parameters: None
Usage:
```
myGLCD.clrScr(); // Clear the screen
```

fillScr(r, g, b);

Fill the screen with a specified color.

Parameters:
```
r: Red component of an RGB value (0-255)
g: Green component of an RGB value (0-255)
b: Blue component of an RGB value (0-255)
```
Usage:
```
myGLCD.fillScr(255,127,0); // Fill the screen with orange
```

fillScr(color);

Fill the screen with a specified pre-calculated RGB565 color.

Parameters: color: RGB565 color value
Usage:
```
myGLCD.fillScr(VGA_RED); // Fill the screen with red
```

setColor(r, g, b);

Set the color to use for all draw*, fill* and print commands.

Parameters:
```
r: Red component of an RGB value (0-255)
g: Green component of an RGB value (0-255)
b: Blue component of an RGB value (0-255)
```
Usage:
```
myGLCD.setColor(0,255,255); // Set the color to cyan
```

setColor(color);

Set the specified pre-calculated RGB565 color to use for all draw*, fill* and print commands.

Parameters: color: RGB565 color value
Usage:
```
myGLCD.setColor(VGA_AQUA); // Set the color to aqua
```

getColor();

Get the currently selected color.

Parameters: None
Returns: Currently selected color as a RGB565 value (word)
Usage:
```
Color = myGLCD.getColor(); // Get the current color
```

setBackColor(r, g, b);

Set the background color to use for all print commands.

Parameters:
```
r: Red component of an RGB value (0-255)
g: Green component of an RGB value (0-255)
b: Blue component of an RGB value (0-255)
```
Usage:
```
myGLCD.setBackColor(255,255,255); // Set the background color to white
```

setBackColor(color);

Set the specified pre-calculated RGB565 background color to use for all print commands.

Parameters: color: RGB565 color value
Usage:
```
myGLCD.setBackColor(VGA_LIME); // Set the background color to lime
```

getBackColor();

Get the currently selected background color.

Parameters: None
Returns: Currently selected background color as a RGB565 value (word)
Usage:
```
BackColor = myGLCD.getBackColor(); // Get the current background color
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

drawPixel(x, y);

Draw a single pixel.

```
Parameters:     x: x-coordinate of the pixel
                y: y-coordinate of the pixel
Usage:          myGLCD.drawPixel(119,159); // Draw a single pixel
```

drawLine(x1, y1, x2, y2);

Draw a line between two points.

```
Parameters:     x1: x-coordinate of the start-point
                y1: y-coordinate of the start-point
                x2: x-coordinate of the end-point
                y2: y-coordinate of the end-point
Usage:          myGLCD.drawLine(0,0,239,319); // Draw a diagonal line
```

drawRect(x1, y1, x2, y2);

Draw a rectangle between two points.

```
Parameters:     x1: x-coordinate of the start-corner
                y1: y-coordinate of the start-corner
                x2: x-coordinate of the end-corner
                y2: y-coordinate of the end-corner
Usage:          myGLCD.drawRect(119,159,239,319); // Draw a rectangle
```

drawRoundRect(x1, y1, x2, y2);

Draw a rectangle with slightly rounded corners between two points. The minimum size is 5 pixels in both directions. If a smaller size is requested the rectangle will not be drawn.

```
Parameters:     x1: x-coordinate of the start-corner
                y1: y-coordinate of the start-corner
                x2: x-coordinate of the end-corner
                y2: y-coordinate of the end-corner
Usage:          myGLCD.drawRoundRect(0,0,119,159); // Draw a rounded rectangle
```

fillRect(x1, y1, x2, y2);

Draw a filled rectangle between two points.

```
Parameters:     x1: x-coordinate of the start-corner
                y1: y-coordinate of the start-corner
                x2: x-coordinate of the end-corner
                y2: y-coordinate of the end-corner
Usage:          myGLCD.fillRect(119,0,239,159); // Draw a filled rectangle
```

fillRoundRect(x1, y1, x2, y2);

Draw a filled rectangle with slightly rounded corners between two points. The minimum size is 5 pixels in both directions. If a smaller size is requested the rectangle will not be drawn.

```
Parameters:     x1: x-coordinate of the start-corner
                y1: y-coordinate of the start-corner
                x2: x-coordinate of the end-corner
                y2: y-coordinate of the end-corner
Usage:          myGLCD.fillRoundRect(0,159,119,319); // Draw a filled, rounded rectangle
```

drawCircle(x, y, radius);

Draw a circle with a specified radius.

```
Parameters:     x:      x-coordinate of the center of the circle
                y:      y-coordinate of the center of the circle
                radius: radius of the circle in pixels
Usage:          myGLCD.drawCircle(119,159,20); // Draw a circle with a radius of 20 pixels
```

fillCircle(x, y, radius);

Draw a filled circle with a specified radius.

```
Parameters:     x:      x-coordinate of the center of the circle
                y:      y-coordinate of the center of the circle
                radius: radius of the circle in pixels
Usage:          myGLCD.fillCircle(119,159,10); // Draw a filled circle with a radius of 10 pixels
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

print(st, x, y[, deg]);

Print a string at the specified coordinates.
You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen.

Parameters:
```
st:  the string to print
x:   x-coordinate of the upper, left corner of the first character
y:   y-coordinate of the upper, left corner of the first character
deg: <optional>
     Degrees to rotate text (0-359). Text will be rotated around the upper left corner.
```
Usage:
```
myGLCD.print("Hello, World!",CENTER,0); // Print "Hello, World!"
```
Notes:
```
CENTER and RIGHT will not calculate the coordinates correctly when rotating text.
The string can be either a char array or a String object
```

printNumI(num, x, y[, length[, filler]]);

Print an integer number at the specified coordinates.
You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen.

Parameters:
```
num:    the value to print (-2,147,483,648 to 2,147,483,647) INTEGERS ONLY
x:      x-coordinate of the upper, left corner of the first digit/sign
y:      y-coordinate of the upper, left corner of the first digit/sign
length: <optional>
        minimum number of digits/characters (including sign) to display
filler: <optional>
        filler character to use to get the minimum length. The character will be inserted in front
        of the number, but after the sign. Default is ' ' (space).
```
Usage:
```
myGLCD.printNumI(num,CENTER,0); // Print the value of "num"
```

printNumF(num, dec, x, y[, divider[, length[, filler]]]);

Print a floating-point number at the specified coordinates.
You can use the literals LEFT, CENTER and RIGHT as the x-coordinate to align the string on the screen.
WARNING: Floating point numbers are not exact, and may yield strange results when compared. Use at your own discretion.

Parameters:
```
num:     the value to print (See note)
dec:     digits in the fractional part (1-5) 0 is not supported. Use printNumI() instead.
x:       x-coordinate of the upper, left corner of the first digit/sign
y:       y-coordinate of the upper, left corner of the first digit/sign
divider: <optional>
         Single character to use as decimal point. Default is '.'
length:  <optional>
         minimum number of digits/characters (including sign) to display
filler:  <optional>
         filler character to use to get the minimum length. The character will be inserted in front
         of the number, but after the sign. Default is ' ' (space).
```
Usage:
```
myGLCD.printNumF(num, 3, CENTER,0); // Print the value of "num" with 3 fractional digits
```
Notes:
```
Supported range depends on the number of fractional digits used.
Approx range is +/- 2^(10^(9-dec))
```

setFont(fontname);

Select font to use with print(), printNumI() and printNumF().

Parameters:
```
fontname: Name of the array containing the font you wish to use
```
Usage:
```
myGLCD.setFont(BigFont); // Select the font called BigFont
```
Notes:
```
You must declare the font-array as an external or include it in your sketch.
```

getFont();

Get the currently selected font.

Parameters:
```
None
```
Returns:
```
Currently selected font
```
Usage:
```
CurrentFont = myGLCD.getFont(); // Get the current font
```

getFontXsize();

Get the width of the currently selected font.

Parameters:
```
None
```
Returns:
```
Width of the currently selected font in pixels
```
Usage:
```
Xsize = myGLCD.getFontXsize (); // Get font width
```

getFontYsize();

Get the height of the currently selected font.

Parameters:
```
None
```
Returns:
```
Height of the currently selected font in pixels
```
Usage:
```
Ysize = myGLCD.getFontYsize (); // Get font height
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

drawBitmap (x, y, sx, sy, data[, scale]);

Draw a bitmap on the screen.

Parameters:
- x: x-coordinate of the upper, left corner of the bitmap
- y: y-coordinate of the upper, left corner of the bitmap
- sx: width of the bitmap in pixels
- sy: height of the bitmap in pixels
- data: array containing the bitmap-data
- scale: **<optional>**
  Scaling factor. Each pixel in the bitmap will be drawn as <scale>x<scale> pixels on screen.

Usage: `myGLCD.drawBitmap(0, 0, 32, 32, bitmap); // Draw a 32x32 pixel bitmap`

Notes: You can use the online-tool "*ImageConverter 565*" or "*ImageConverter565.exe*" in the Tools-folder to convert pictures into compatible arrays. The online-tool can be found on my website.
Requires that you *#include <avr/pgmspace.h>* when using an Arduino other than Arduino Due.

drawBitmap (x, y, sx, sy, data, deg, rox, roy);

Draw a bitmap on the screen with rotation.

Parameters:
- x: x-coordinate of the upper, left corner of the bitmap
- y: y-coordinate of the upper, left corner of the bitmap
- sx: width of the bitmap in pixels
- sy: height of the bitmap in pixels
- data: array containing the bitmap-data
- deg: Degrees to rotate bitmap (0-359)
- rox: x-coordinate of the pixel to use as rotational center relative to bitmaps upper left corner
- roy: y-coordinate of the pixel to use as rotational center relative to bitmaps upper left corner

Usage: `myGLCD.drawBitmap(50, 50, 32, 32, bitmap, 45, 16, 16); // Draw a bitmap rotated 45 degrees around its center`

Notes: You can use the online-tool "*ImageConverter 565*" or "*ImageConverter565.exe*" in the Tools-folder to convert pictures into compatible arrays. The online-tool can be found on my website.
Requires that you *#include <avr/pgmspace.h>* when using an Arduino other than Arduino Due.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTFT_Buttons

Add-on Library for UTFT: Buttons

Manual

(C)2014 Rinky-Dink Electronics, Henning Karlsen

Introduction:

This library is an add-on to UTFT and will not work on its own.
This add-on library also requires the UTouch library.

This library adds simple but easy to use buttons to extend the use of the UTFT and UTouch libraries.

You can always find the latest version of the library at **http://www.RinkyDinkElectronics.com/**

For version information, please refer to **version.txt**.

IMPORTANT:

The library defaults to a maximum of 20 simultaneous buttons.

This number can be adjusted according to your needs by changing the number on the line:

```
#define MAX_BUTTONS 20
```

In the **UTFT_Buttons.h** file.

You should note that every possible button will reserve a small amount of RAM, 13-15 bytes depending on what development board you are using, whether it is used or not so you should not increase the number beyond what you actually need.

This library is licensed under a **CC BY-NC-SA 3.0** (Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported) License.

For more information see: **http://creativecommons.org/licenses/by-nc-sa/3.0/**

เอกสารนี้ผลิตขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

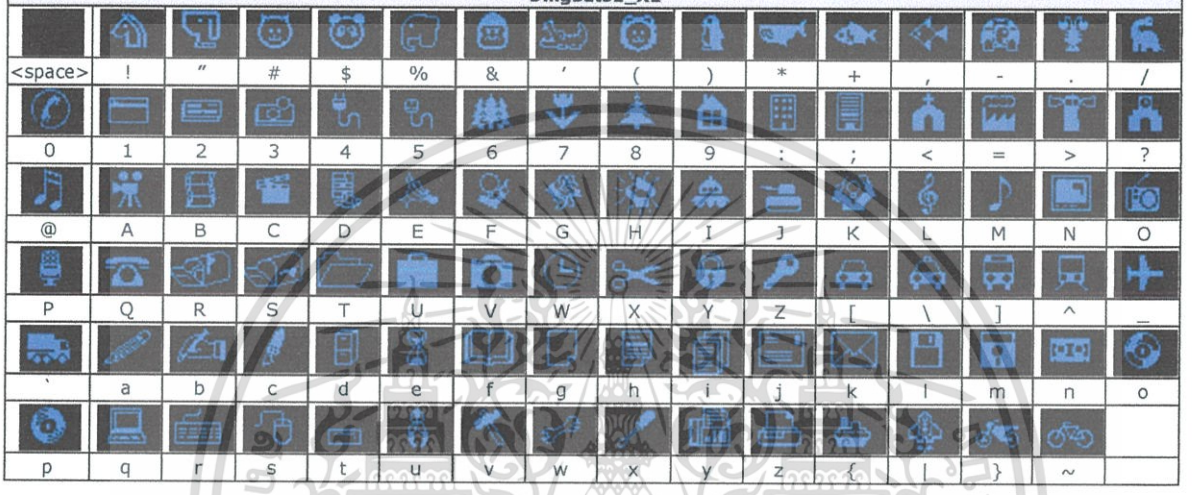
DEFINED LITERALS:

Status flags

BUTTON_DISABLED:	0x0001	
BUTTON_SYMBOL:	0x0002	
BUTTON_SYMBOL_REP_3X:	0x0004	
BUTTON_BITMAP:	0x0008	(Should not be used manually)
BUTTON_NO_BORDER:	0x0010	(Only valid for bitmap buttons)
BUTTON_UNUSED:	0x8000	(Should not be used manually)

INCLUDED FONTS:

Dingbats1_XL



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTIONS:

UTFT_Buttons(UTFT, UTouch);

The main class constructor.

Parameters:
```
UTFT  : A reference to an already created UTFT object
UTouch: A reference to an already created UTouch object
```

Usage:
```
UTFT_Buttons myButtons(&myGLCD, &myTouch); // Start an instance of the UTFT_Buttons class
```

Notes: Remember the '&' in front of the object names

addButton(x, y, width, height, label[, flags]);

Add a new text or symbol button.

Parameters:
```
X     : x-coordinate for the upper left corner of the button
y     : y-coordinate for the upper left corner of the button
width : width of the button in pixels
height: height of the button in pixels
label : button text or character for symbol
flags : <optional>
        Can use any combination of BUTTON_DISABLED, BUTTON_SYMBOL and BUTTON_SYMBOL_REP_3X.
        Use | to combine. Default is <none>.
```

Returns: (INT) buttonID, -1 if no button could be added

Usage:
```
int but1 = myButtons.addButton( 10,  20, 300,  30, "Button 1"); // add a new button "Button 1"
```

Notes: Buttons will not be drawn on the screen until drawButton() or drawButtons() is called.

addButton(x, y, width, height, data[, flags]);

Add a new bitmap button.

Parameters:
```
X     : x-coordinate for the upper left corner of the button
y     : y-coordinate for the upper left corner of the button
width : width of the bitmap in pixels
height: height of the bitmap in pixels
data  : array containing the bitmap-data
flags : <optional>
        Can use any combination of BUTTON_DISABLED or BUTTON_NO_BORDER.
        Use | to combine. Default is <none>.
```

Returns: (INT) buttonID, -1 if no button could be added

Usage:
```
int but1 = myButtons.addButton( 10,  20, 300,  30, bitmap); // add a new bitmap button
```

Notes: Buttons will not be drawn on the screen until drawButton() or drawButtons() is called.
You can use the online-tool "ImageConverter 565" or "ImageConverter565.exe" supplied with UTFT to convert pictures into compatible arrays. The online-tool can be found on my website.

drawButtons();

Draw all currently added buttons on the screen.

Parameters: None

Usage:
```
myButtons.drawButtons(); // Draw all buttons
```

drawButton(buttonID);

Draw a single button on the screen.

Parameters:
```
buttonID: ID of the button to draw
```

Usage:
```
myButtons.drawButton(but1); // Draw button with buttonID but1
```

enableButton(buttonID[, redraw]);

Set button state to enabled/clickable.

Parameters:
```
buttonID: ID of the button to enable
redraw  : <optional>
          true : redraw button immediately
          false: do not redraw button yet (Default)
```

Usage:
```
myButtons.enableButton(but1, true); // Enable button with buttonID but1 and redraw it
```

disableButton(buttonID[, redraw]);

Set button state to disabled/unclickable.

Parameters:
```
buttonID: ID of the button to disable
redraw  : <optional>
          true : redraw button immediately
          false: do not redraw button yet (Default)
```

Usage:
```
myButtons.disableButton(but1); // Disable button with buttonID but1 but do not redraw it
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

buttonEnabled(buttonID);

Check the enabled/disabled status of a button.

Parameters: buttonID: ID of the button to disable

Returns: (BOOLEAN) **true** if button is enabled, otherwise **false**

Usage:
```
boolean state = myButtons.buttonEnabled(but1); // Check if the button with ButtonID but1 is enabled
```

relabelButton(buttonID, label[, redraw]);

Relabel a button.

Parameters:
```
buttonID: ID of the button to enable
label   : new button text or character for symbol
redraw  : <optional>
          true : redraw button immediately
          false: do not redraw button yet (Default)
```

Usage:
```
myButtons.relabelButton(but1, "New Label"); // Relabel button with buttonID but1 but do not redraw
```

deleteButton(buttonID);

Delete a button.

Parameters: buttonID: ID of the button to delete

Usage:
```
myButtons.deleteButton(but1); // Delete button with buttonID but1
```

Notes: Already drawn buttons will not be deleted from the screen, but they will no longer be detected by calling checkButtons()

deleteAllButtons();

Delete all current buttons.

Parameters: None

Usage:
```
myButtons.deleteAllButtons(); // Delete all buttons
```

Notes: Already drawn buttons will not be deleted from the screen, but they will no longer be detected by calling checkButtons()

checkButtons();

Check if any button is being pressed.

Parameters: None

Returns: (INT) buttonID of pressed button, -1 if no button is pressed

Usage:
```
int pressed = myButtons.checkButtons(); // Check if any buttons are pressed
```

setTextFont(fontname);

Select which font to use for button labels.

Parameters: fontname: Name of the array containing the font you wish to use

Usage:
```
myButtons.setTextFont(BigFont); // Select the font called BigFont
```

Notes: You must declare the font-array as an external or include it in your sketch.

setSymbolFont(fontname);

Select which font to use for button symbols.

Parameters: fontname: Name of the array containing the font you wish to use

Usage:
```
myButtons.setSymbolFont(Dingbats1_XL); // Select the font called Dingbats1_XL
```

Notes: You must declare the font-array as an external or include it in your sketch.

setButtonColors(text, inactive, border, highlight, background);

Set the colors used to draw the buttons.

Parameters:
```
text      : RGB565-encoded color to use for button text and symbols
inactive  : RGB565-encoded color to use for button text and symbols on disabled buttons
border    : RGB565-encoded color to use for button borders
highlight : RGB565-encoded color to use for button borders when selected
background: RGB565-encoded color to use for button background
```

Usage:
```
myButton.setButtonColors(VGA_WHITE, VGA_GRAY, VGA_WHITE, VGA_RED, VGA_BLUE); // Set default colors
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

UTouch

Arduino and chipKit Universal TFT touchscreen library

Manual

(C)2014 Rinky-Dink Electronics, Henning Karlsen

Introduction:

This library was made to complement UTFT to provide touch screen functionality.

You can always find the latest version of the library at **http://www.RinkyDinkElectronics.com/**

For version information, please refer to **version.txt**.

REGARDING CALIBRATION:

All touch screens will have slight variations. It is therefore important that you calibrate your particular touch screen for the best possible performance. The default calibration data supplied with the library *may* work on your screen, but ***only if*** your screen has a 320x240 resolution. Screens with other resolutions **MUST** be calibrated.

To calibrate your touch screen you will need to run the **UTouch_Calibration** sketch supplied in the examples of the library.

Before you compile and upload the sketch there are a couple of things you must do.

1. Make sure you have uncommented the correct section for your development board
2. Make sure the UTFT display model code is correct for your display module
3. Make sure the TOUCH_ORIENTATION define is correct. You can find a list of the correct parameter for all the tested displays in the ***UTouch_Supported_display_modules*** PDF.

Further instructions will be given on screen when you run the sketch.

Remember that if you have more than one touch display module you may have to run the calibration on each module.

An on-line tool to verify your calibration data can be found at
http://www.RinkyDinkElectronics.com/t_cal_verify.php

Some touch screens, espcially the larger ones (4.3" and larger), have some flaws where they have problems registering touch near the edges. The calibration sketch tries to take this into account when calibrating. Because of this some calibration points may take longer to register.

It is also recommended that you power your Arduino/chipKit using an external power source when running the calibration on 4.3" and larger screens.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Defined Literals:

Orientation
For use with InitTouch()
PORTRAIT: 0
LANDSCAPE: 1

Precision
For use with setPrecision()
PREC_LOW: 1
PREC_MEDIUM: 2
PREC_HI: 3
PREC_EXTREME: 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functions:

UTouch(TCLK, TCS, TDIN, TDOUT, IRQ);

The main class of the interface.

Parameters:
```
TCLK:  Pin for Touch Clock (D_CLK)
TCS:   Pin for Touch Chip Select (D_CS)
TDIN:  Pin for Touch Data input (D_DIN)
TDOUT: Pin for Touch Data output (D_OUT)
IRQ:   Pin for Touch IRQ (DPenirq)
```
Usage:
```
UTouch myTouch(15,10,14,9,8); // Start an instance of the UTouch class
```

InitTouch([orientation]);

Initialize the touch screen and set display orientation. If the library is used together with UTFT the orientation should be set to the same orientation for both libraries.

Parameters:
```
orientation: <optional>
             PORTRAIT
             LANDSCAPE (default)
```
Returns: Nothing

Usage:
```
myTouch.InitTouch();// Initialize the touch screen
```

dataAvailable();

Check to see if new data from the touch screen is waiting.

Parameters: None

Returns: Boolean: true means data is waiting, otherwise false

Usage:
```
check = myTouch.dataAvailable() // See if data is waiting
```

read();

Read waiting data from the touch screen. This function should be called if dataAvailable() is true. Use getX() and getY() to get the coordinates.

Parameters: None

Returns: Nothing

Usage:
```
myTouch.read(); // Read data from touch screen
```
Notes: After calling read(), raw data from the touch screen is available in the variables TP_X and TP_Y. Do not use these if you do not know how to handle the raw data. Use getX() and getY() instead.

getX();

Get the x-coordinate of the last position read from the touch screen.

Parameters: None

Returns: Integer

Usage:
```
x = myTouch.getX(); // Get the x-coordinate
```

getY();

Get the y-coordinate of the last position read from the touch screen.

Parameters: None

Returns: Integer

Usage:
```
y = myTouch.getY(); // Get the y-coordinate
```

setPrecision(precision);

Set the precision of the touch screen.

Parameters: precision: **PREC_LOW**, **PREC_MEDIUM**, **PREC_HI**, **PREC_EXTREME**

Returns: Nothing

Usage:
```
myTouch.setPrecision(PREC_MEDIUM); // Set precision to medium
```
Notes: Higher precision data will take longer to read, so take care when using PREC_HI or PREC_EXTREME with fast-moving input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้