สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอุณหภูมิและความชื้นในห้องจำลอง

Humidity and Temperature Control system

โดย

นายชัยวัฒน์ พุ่มฉัตร์ 48015285

นายศักดา กาละศรี 48015313

นายวิทยา สุขวิทยากุล 48015308

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่ 83069

วัน,เดือน,ปี - 5 ส.ค. 2551

b 11964258

i

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุณหภูมิและความชื้นในห้องจำลอง

Humidity and Temperature Control system

โดย

นายชัยวัฒน์	พุ่มจัตรี	48015285
นายศักดา	กาละศรี	48015313
นายวิทยา	สุขวิทยากุล	48015308

อาจารย์ที่ปรึกษา

รศ.ดร. เกียรติศักดิ์ คมวัชระ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2550

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอุณหภูมิและความชื้นในห้องจำลอง

Humidity and Temperature Control system

ผู้จัดทำ

1. นาย ชัยวัฒน์ พุ่มฉัตร์
2. นาย ศักดา กาละศรี
3. นาย วิทยา สุขวิทยากุล

..... อาจารย์ที่ปรึกษา

(รศ.ดร เกียรติศักดิ์ คมวัชระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอุณหภูมิและความชื้นในห้องจำลอง

Humidity and Temperature Control system

ชัยวัฒน์ พุ่มฉัตร์ 48015285

ศักดา กาละศรี 48015313

วิทยา สุขวิทยากุล 48015308

อาจารย์ที่ปรึกษา

รศ.ดร. เกียรติศักดิ์ คมวัชระ

ปีการศึกษา 2550

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นฉบับปรับปรุง และพัฒนาการควบคุมอุณหภูมิและความชื้นภายในห้องทดลอง โดยการประยุกต์ไมโครคอนโทลเลอร์ตระกูล PIC เบอร์ 18F458 โดยจะใช้ sensor SHT 15 ซึ่งเป็นเซ็นเซอร์ตรวจวัดอุณหภูมิและความชื้นภายในตัวเดียวมาตรวจวัดอุณหภูมิและความชื้นภายในห้องทดลอง และทำการเขียนโปรแกรมควบคุมให้กับชุดควบคุมย่อยเพื่อจะสามารถควบคุมความชื้นภายในห้องทดลองได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Humidity and Temperature Control system

By

Mr.Chaiwat pumchat

Mr.Wittaya sukwittayakul

Mr. Sakda kalasri

Advisor

Dr.Kiattisak Komwatchara

Academic Year 2007

Abstract

This version of dissertation has being modified and the development of temperature and humidity control in laboratory. The method is being applied by using PIC no. 18F458 Controller which its SHT 15 sensor will check and measure temperature and humidity in the laboratory. The method is also to write a program to manage subset controller in order to control humidity in laboratory as requirement.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ผู้จัดทำปริญญานิพนธ์ขอขอบพระคุณอาจารย์ที่ปรึกษา รศ.ดร เกียรติศักดิ์ คมวัชระ ที่ให้คำปรึกษา แนะนำ ตลอดจนแนวคิดเห็นและแนวทางในการปฏิบัติงานและการแก้ไขปัญหาที่เป็นอุปสรรคต่างๆเกี่ยวกับโครงงานนี้ และขอขอบพระคุณอาจารย์ทุกๆท่านที่ได้ให้ความรู้ในด้านต่างๆ ซึ่งสามารถนำมาประยุกต์ใช้เกี่ยวกับการทำโครงงานได้แทบทั้งสิ้น และขอบคุณพี่ๆ เพื่อนๆ ที่คอยช่วยเหลือให้คำแนะนำทำให้โครงงานฉบับนี้สำเร็จลุล่วงไปด้วยดีในภาคเรียนที่ 2 ของการทำโครงงาน ผู้จัดทำขอระลึกถึงด้วยความขอบพระคุณอย่างยิ่ง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง		หน้า
บทคัดย่อ		I
สารบัญ		II
สารบัญภาพ		IV
สารบัญตาราง		VI
บทที่ 1 บทนำ		
	1.1 ความเป็นมาและความสำคัญ	1
	1.2 วัตถุประสงค์	1
	1.3 แนวคิดและขอบเขตของโครงงาน	2
	1.4 แสดงผังการทำงานหลักของห้องจำลองการควบคุมอุณหภูมิและความชื้น	2
	1.5 โครงสร้างของเนื้อหาของโครงงาน	3
	1.6 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 หลักการและทฤษฎี		
	2.1 ระบบการเชื่อมต่ออุปกรณ์แบบ I^2C Bus	4
	2.2 ตัวประมวลผลไมโครคอนโทรเลอร์	6
	2.3 ข้อมูลเบื้องต้นของ PCF8591	23
	2.4 DS1307 ไอซีร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)	24
	2.5 วงจรการสร้างสัญญาณกระตุ้น	26
	2.6 เซนเซอร์ที่ใช้วัดอุณหภูมิแล้วความชื้น	27
	2.7 ส่วนแสดงผลทาง LCD Module	30
	2.8 คีย์แพด (Keypad)	30
	2.9 ตัวทำความชื้นชนิดอัลตราโซนิค (ULTRASONIC HUMIDIFIERS)	38
	2.10 ฮีทเตอร์อากาศแบบครีบ Fin heater แบบตัว U (CAL-P)	39
	2.11 ตัวควบคุมสัดส่วน	40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
บทที่ 3 โครงสร้างและการออกแบบระบบ	
3.1 รูปวงจรรวมส่วนควบคุมการทำงานทั้งหมด	41
3.1.1 วงจรภาคแหล่งจ่ายไฟและ Vsync control	42
3.1.2 วงจรส่วนไมโครคอนโทรเลอร์	42
3.1.3 ส่วนของ Keypad and LCD	43
3.1.4 สวนของวงจรเฟสคอนโทล	43
3.1.5 วงจรควบคุมพัดลมความชื้น	44
3.1.6 วงจรควบคุมพัดลม I/O	44
3.1.7 วงจรควบคุมตัวทำความชื้น	45
3.1.8 วงจรส่งผ่าน RS 232	45
3.2 รูปห้องจำลองและห้องควบคุมรวม	46
3.2.1 ห้อง CONTROL หลัก	46
3.2.2 ห้องจำลองการทดลอง	47
3.2.3 ห้องทำความชื้น	47
3.2.4 การวางวงจรภายในตู้ Control	48
บทที่ 4 ขั้นตอนการทดลองและผลการทดลอง	
4.1 การทดลอง การควบคุมความชื้นเพียงอย่างเดียว	49
4.2 การทดลอง การควบคุมอุณหภูมิเพียงอย่างเดียว	49
4.3 การทดลอง การควบคุมอุณหภูมิและความชื้นพร้อมกัน	50
บทที่ 5 บทสรุป ปัญหาที่พบ และ แนวทางการพัฒนา	
5.1 บทสรุป	55
5.2 ปัญหาที่พบ	55
5.3 แนวทางการพัฒนา	55
ภาคผนวก ก. โปรแกรมต่างๆที่ใช้ในการควบคุม	
ภาคผนวก ข. เอกสารอ้างอิง	
กิตติกรรมประกาศ	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพ	หน้า
1.1 บล็อกไดอะแกรมของระบบ	2
2.1 ไดอะแกรมแสดงสถานะต่าง ๆ บนระบบบัส I2C	6
2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC แบบ OTP	7
2.3 ไมโครคอนโทรลเลอร์ตระกูล PIC แบบ EPROM	8
2.4 ไมโครคอนโทรลเลอร์ตระกูล PIC แบบ Flash หรือ EEPROM	8
2.5 ขาสัญญาณของไมโครคอนโทรลเลอร์ตระกูล PIC18F458	11
2.6 โครงสร้างภายในของไมโคคอนโทรลเลอร์ PIC เบอร์ PIC18F458	15
2.7 เซลามิคเรโซเนเตอร์	17
2.8 การต่อคริสตอลและตัวเก็บประจุเข้ากับ PIC	18
2.9 โครงสร้างขาของ Crystal Square-wave	18
2.10 วงจรการเชื่อมต่อ PIC กับวงจรกำเนิดสัญญาณ	19
2.11 การจัดการพื้นที่หน่วยความจำของ PIC18F458	21
2.12 การจัดขาของอซี ADC/DAC ขนาด 8 บิตผ่านบัส I²C เบอร์ PCF8591	24
2.13 การจัดขาของไอซี DS1307 ไอซีสร้างฐานเวลาจริง (RTC)	25
2.14 บล็อกไดแกรมของ TCA 785	26
2.15 รูป IC SHT 15	27
2.16 แสดง Block Diagram ของ SHT15	29
2.17 แสดการต่อ SHT15 กับไมโรคอนโทรเลอร์	29
2.18 แสดง Timing Diagram ในช่วงการส่งข้อมูลของ SHT 15	30
2.19 แสดงรูปของจอ LCD	30
2.20 การทำงานของสวิตช์	36
2.21 แสดงการเกิดพัลส์เมื่อทำการกดสวิตช์	37
2.22 ตัวทำความชื้นชนิดอัลตร้าโซนิค (ultrasonic humidifiers)	37
2.23 รูปแสดงการเชื่อมต่อคีย์แพดเข้ากับตัวไมโครคอนโทรลเลอร์	38
3.1 รูปวงจรรวมส่วนควบคุมการทำงานทั้งหมด	41
3.1.1วงจรภาคแหล่งจ่ายไฟและ Vsync control	42
3.1.2วงจรส่วนไมโครคอนโทรเลอร์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพ	หน้า
3.1.3 ส่วนของ Keypad and LCD	43
3.1.4 ส่วนของวงจรเฟสคอนโทล	43
3.1.5 วงจรควบคุมพัดลมความชื้น	44
3.1.6 วงจรควบคุมพัดลม I/O	44
3.1.7 วงจรควบคุมตัวทำความชื้น	45
3.1.8 วงจรส่งผ่าน RS 232	45
3.2 รูปห้องจำลองและห้องควบคุมรวม	46
3.2.1 ห้อง CONTROL หลัก	46
3.2.2 ห้องจำลองการทดลอง	47
3.2.3 ห้องทำความชื้น	47
3.2.4 การวางวงจรควบคุมภายในตู้ CONTROL	48
4.1 การทดลอง การควบคุมความชื้นเพียงอย่างเดียว	49
4.2 การควบคุมอุณหภูมิเพียงอย่างเดียว	49
4.3 การควบคุมอุณหภูมิและความชื้นพร้อมกัน	50
4.4 ภาพกราฟแสดงการเปรียบเทียบสัญญาณระหว่างมุมกระตุ้น Triac กับแรงดันตกคร่อม Heater	51
4.5 รูปกราฟแสดงรูปสัญญาณของ TCA 785	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตาราง	หน้า
2.1 หน้าที่ของขาสัญญาณของพอร์ต A	12
2.2 หน้าที่ของขาสัญญาณของพอร์ต B	13
2.3 หน้าที่ของขาสัญญาณของพอร์ต C	13
2.4 หน้าที่ของขาสัญญาณของพอร์ต D	14
2.5 หน้าที่ของสัญญาณของพอร์ต E	14
2.6 โหมดของสัญญาณนาฬิกาเมื่อใช้เซรามิคเรโซเนเตอร์	20
2.7 โหมดของสัญญาณนาฬิกาเมื่อใช้คริสตอล	20
2.8 แสดงคุณสมบัติการทำงานของ SHT 15	28
2.9 แสดงลักษณะของ SHT15	28
4.1 ตารางแรงดันพัดลมความชื้น	50
4.2 ตารางมุมเฟสและแรงดันของ Heater	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

โดยปกติอุณหภูมิและความชื้นจะมีอยู่ทั่วๆไปทุกหนทุกแห่ง รวมไปถึงบริเวณรอบๆตัวเราด้วย ดังนั้น เราอาจพูดได้ว่าอุณหภูมิและความชื้นจะส่งผลกระทบโดยตรงกับความเป็นอยู่ของสิ่งที่มีชีวิตและไม่มีชีวิต ซึ่งมากน้อยเพียงใดขึ้นอยู่กับสภาพแวดล้อมนั้นๆ เพราะฉะนั้นถ้าเราสามารถควบคุมอุณหภูมิและความชื้นได้ ก็สามารถที่จะนำประยุกต์ใช้ได้ตามสภาพความเหมาะสมกับความต้องการ และจากอดีตจนถึงปัจจุบันก็ได้มีการพัฒนาเทคโนโลยีต่างๆ รวมถึงการพัฒนาทางด้านอุตสาหกรรมซึ่งส่งผลโดยตรงกับชีวิตความเป็นอยู่ของมนุษย์ ดังนั้นการควบคุมอุณหภูมิและความชื้นจึงเข้ามามีบทบาทสำคัญในการดำรงชีวิตของมนุษย์ ทั้งทางด้านการเกษตร ด้านการแพทย์ และยังรวมไปถึงทางด้านอุตสาหกรรมการผลิตต่างอีกด้วยๆ

ในปัจจุบันเทคโนโลยีทางด้านอิเล็กทรอนิกส์และคอมพิวเตอร์ มีสูงขึ้นมากมนุษย์จึงมีทางเลือกวิธีที่จะนำเอาเทคโนโลยีที่ทันสมัยมาใช้ในการควบคุมอุณหภูมิและความชื้น เช่น ระบบสมองกลฝังตัวโดยการใช้ไมโครคอนโทลเลอร์มาใช้ประยุกต์และเป็นตัวประมวลผลในการควบคุมหลัก และใช้อุปกรอิเล็กทรอนิกส์กำลังมาเป็นตัวควบคุมทางด้านกำลังขาออก เช่น อุปกรณ์ในกลุ่มของไทริสเตอร์ ซึ่งจะถูกควบคุมการทำงานจากไมโครคอลโทรลเลอร์อีกทีหนึ่ง

1.2 วัตถุประสงค์

โครงงานนี้เป็นการประยุกต์เทคโนโลยีไมโครโปรเซสเซอร์มาใช้เป็นตัวควบคุมระบบการทำงานแบบอัตโนมัติของห้องทดลองการควบคุมความชื้นและอุณหภูมิ โดยมีวัตถุประสงค์ดังนี้

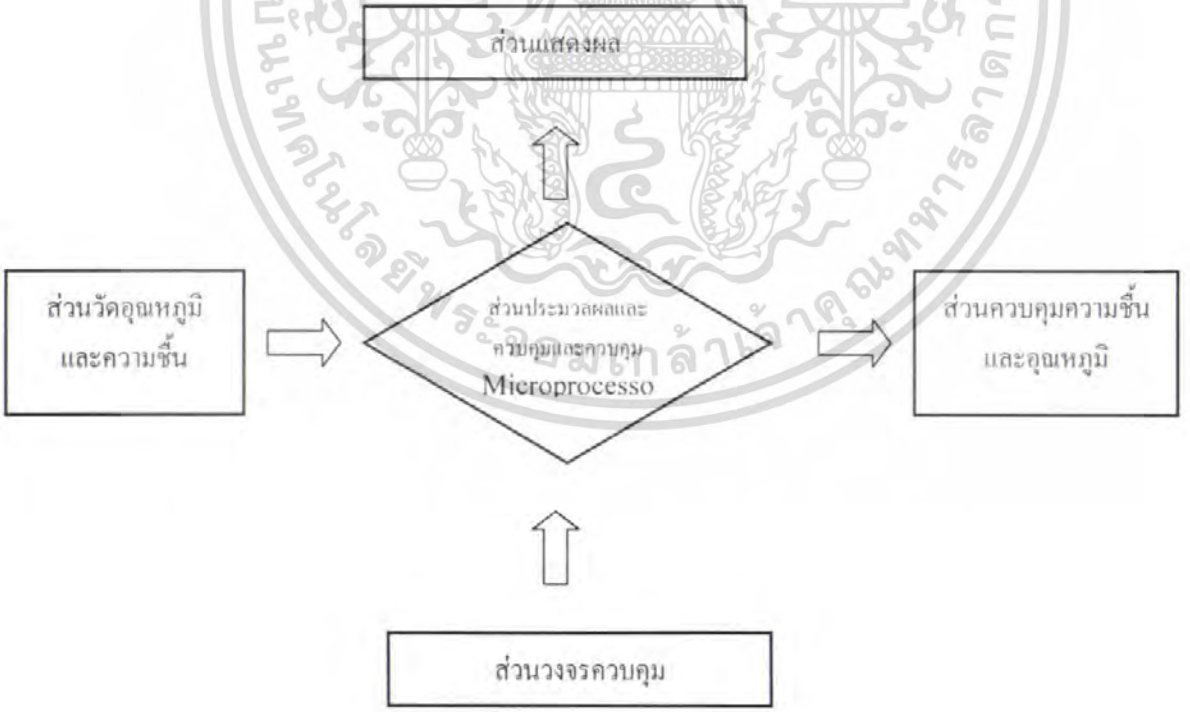
1. เพื่อศึกษาการออกแบบระบบอัตโนมัติที่เป็นไปได้มากที่สุดที่มีตัวไมโครโปรเซสเซอร์เป็นตัว ควบคุมระบบ
2. เพื่อออกแบบระบบวัดอุณหภูมิและความชื้น
3. เพื่อออกแบบระบบควบคุมอุณหภูมิและความชื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 แนวคิดและขอบเขตของโครงงาน

ระบบควบคุมอุณหภูมิและความชื้นประกอบด้วย ส่วนของเซนเซอร์อ่านค่าอุณหภูมิและความชื้น ซึ่งจะทำการวัดค่าอุณหภูมิและความชื้นแล้วส่งค่ามายังไมโครคอนโทรเลอร์เพื่อประมวลผล แล้วส่งค่ามาแสดงผลยังน่าจอ LCD เพื่อให้รู้ถึงค่าของอุณหภูมิและความชื้นที่วัดได้ ณ เวลานั้น และค่าของข้อมูลส่วนหนึ่งจะถูกส่งไปยังไมโครคอนโทรเลอร์ เพื่อเป็นตัวเปรียบเทียบข้อมูลระหว่างค่าที่วัดได้กับค่าที่ต้องการจะควบคุม โดยค่าที่ต้องการจะควบคุมนั้นจะถูกป้อนค่าผ่านทางคีย์สวิตช์ โดยท่าหากว่าค่าที่วัดได้ไม่ตรงกับค่าที่ต้องการควบคุมไมโครคอนโทรเลอร์จะกำหนดให้อุปกรณ์ทำงานควบคุมอุณหภูมิและความชื้น โดยที่การควบคุมอุณหภูมิเราจะใช้ ฮีตเตอร์ในการทำความร้อนและความชื้นเราจะใช้ตัวทำอัลตร้าโซนิคในการควบคุม เมื่อค่าอุณหภูมิและความชื้นที่วัดได้มีค่าตรงกับค่าที่ตั้งไว้ก็จะสั่งให้อุปกรณ์ควบคุมอุณหภูมิและความชื้นหยุดทำงาน การที่จะควบคุมอุปกรณ์นั้นไมโครคอลโทรเลอร์จะส่งให้อุปกรณ์ควบคุมทำงานมากน้อยตามค่า ERROR ของค่าจริงและค่าที่ต้องการจะควบคุม

1.4 แสดงผังการทำงานหลักของห้องจำลองการควบคุมอุณหภูมิและความชื้น



รูปที่ 1.1 บล็อกไดอะแกรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบของห้องจำลองการควบคุมอุณหภูมิและความชื้น

1. ส่วนตัวประมวลผลควบคุม

เป็นส่วนที่ทำหน้าที่ควบคุมการทำงานทุกส่วนของห้องจำลองการควบคุมอุณหภูมิและความชื้น ตั้งแต่การรับข้อมูลจากส่วนอินพุทของตัวห้องจำลองเข้ามายังตัวประมวลผล และทำการส่งข้อมูลที่ทำการประมวลผลแล้วส่งออกไปยังส่วนแสดงผล

2. ส่วนวัดอุณหภูมิและความชื้น

ตรวจจับค่าอุณหภูมิและความชื้นโดยใช้เซนเซอร์ SHT 15 ซึ่งจะสามารถวัดค่าได้ทั้งอุณหภูมิและความชื้นภายในตัวเดียวกัน โดยจะให้เอาท์พุทออกมาเป็นสัญญาณดิจิตอลส่งออกมายังตัวประมวลผล

3. ส่วนควบคุม

รับการตั้งค่าอุณหภูมิที่กำหนดการทำงานให้กับห้องจำลอง เพื่อส่งไปยังตัวประมวลผลควบคุมการทำงานของห้องจำลอง

4. ส่วนแสดงผล

รับค่าที่ได้จากส่วนแสดงผลออกไปแสดงผลทาง Module LCD

1.5 โครงสร้างของเนื้อหาของโครงงาน

รายละเอียดของโครงงานนี้แบ่งเป็นเนื้อหาในส่วนต่างๆดังนี้

บทที่ 1 บทนำ กล่าวถึง แนวคิด ที่มา วัตถุประสงค์ คุณสมบัติทั่วไปและประโยชน์ที่คาดว่าจะได้รับจากการทำโครงงาน

บทที่ 2 กล่าวถึง ทฤษฎีที่เกี่ยวข้องในการทำโครงงานนี้

บทที่ 3 กล่าวถึง การออกแบบและการสร้างระบบ

บทที่ 4 กล่าวถึง การทดลองและผลการทดลอง

บทที่ 5 กล่าวถึง สรุป วิจารณ์ผลการทดลอง และแนวทางการพัฒนาระบบ

1.6 ประโยชน์ที่คาดว่าจะได้รับ

- สามารถศึกษาการทำงานของห้องจำลองการควบคุมความชื้นและอุณหภูมิได้
- สามารถศึกษาเทคโนโลยีไมโครคอนโทรเลอร์ที่นำมาประยุคใช้กับชิ้นงานได้
- สามารถศึกษาเกี่ยวกับระบบควบคุมแบบต่างๆ ที่นำมาประยุกต์ใช้กับการการควบคุมอุณหภูมิและความชื้นได้
- สามารถออกแบบระบบการวัดและควบคุมอุณหภูมิและความชื้นเพื่อใช้งานกับระบบกึ่งอัตโนมัติหรือเป็นแบบอัตโนมัติให้มากที่สุดเท่าที่จะเป็นไป
- เป็นข้อมูลสนับสนุนการวิจัยพัฒนาของผู้สนใจได้นำไปใช้ประโยชน์ในการพัฒนาเทคโนโลยีทางด้านเครื่องมือและอุปกรณ์ทางการแพทย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการและทฤษฎี

2.1 ระบบการเชื่อมต่ออุปกรณ์แบบ I² C Bus

2.1.1 ความรู้เบื้องต้นเกี่ยวกับ I^2C

I^2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซีโดยบัส I^2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philps) ด้วยจุดมุ่งหมายหลัก คือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อสั่งงาน และควบคุมภายใต้สัญญาณเพียง 2 เส้น เส้นหนึ่ง คือ สายข้อมูล อีกเส้นหนึ่ง คือสายสัญญาณนาฬิกา ที่ใช้ในการกำหนดจังหวะการทำงาน การต่อรวมกันของอุปกรณ์บนบัส I^2Cทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณของอุปกรณ์แต่ละตัวขนานหรือพ่วงกันไป ส่วนการกำหนดแอดเดรสหรือตำแหน่งสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลและการกำหนดสภาวะลอจิกที่ขาแอดเดรสของอุปกรณ์แต่ละตัว

สายข้อมูลบนบัส I^2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรมหรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line)

อุปกรณ์ที่ทำการเชื่อมต่อบนบัส I^2C มีหลายชนิด ไม่ว่าจะเป็นไอซีขยายพอร์ตอินพุตเอาท์พุต (I/O Expander), ไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอล (ADC) และแปลงดิจิตอลเป็นอนาล็อก (DAC), ไอซีรีลไทม์คล็อก (RTC), ไอซีขับโมดูล LCD, หน่วยความจำอีอีพรอมและไมโครคอนโทรลเลอร์

2.1.2 คุณสมบัติโดยทั่วไปของบัส I^2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (bi-directional line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน + 5v ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาท์พุทของอุปกรณ์ที่อยู่บนบัส I^2C ต้องมีลักษณะเป็นวงจรเดรนเปิด (open-drain) หรือคอลเล็กเตอร์เปิด (open-collector)

อัตราการถ่ายทอดข้อมูลบนบัส I^2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (standard mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (fast mode) อุปกรณ์ที่ต่อร่วมอยู่บนบัส I^2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400 pF การเข้าถึงอุปกรณ์บนบัส I^2C ใช้ข้อมูลการเข้าถึง 2 ค่า คือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

ข้อเด่นอีกประการหนึ่งของบัส I^2C คือ สามารถเชื่อมต่ออุปกรณ์ที่ใช้ไฟเลี้ยงไม่เท่ากันให้สามารถติดต่อสื่อสารกันได้ โดยอุปกรณ์บนบัส I^2C ตัวหนึ่งอาจใช้ไฟเลี้ยง +5V ในขณะที่อีกตัวหนึ่งใช้ไฟเลี้ยง +12V การต่อรวมกันบนบัส I^2C สามารถกระทำได้ในลักษณะเดียวกันกับกรณีที่อุปกรณ์ทั้งสองใช้ไฟเลี้ยงเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กล่าวคือ ให้ต่อสาย SDA และ SCL ของอุปกรณ์แต่ละตัวเข้าด้วยกัน และต้องตัวต้านทานพลูอัพ(R_p) เข้ากับแรงดัน +5V ไว้ด้วยเสมอ

ในกรณีที่อาจมีแรงดันไฟกระชากขนาดใหญ่ปะปนเข้ามาในบัส I^2C ที่ขา SDA และ SCL ของอุปกรณ์แต่ละตัว ต้องต่อตัวต้านทานอนุกรมกับขา SDA และ SCL เรียกว่า R_s ก่อนต่อเข้าสู่บัส I^2C

2.1.3 หลักการของบัส I^2C

บัส I^2C ประกอบด้วยสายสัญญาณ 2 เส้น ดังที่ได้กล่าวมาแล้ว คือ SDA และ SCL อุปกรณ์ที่ต่อพ่วงบนบัสสามารถมีได้มากมาย ดังนั้น จึงต้องมีการกำหนดรูปแบบของการติดต่อบนบัส หรือเรียกว่า โปรโตคอล (protocal) เพื่อให้ผู้ใช้สามารถทราบได้ว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ใดเป็นตัวรับตัวส่ง ต่อไปนี้จะอธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I^2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I^2C ต่อไป

อุปกรณ์ที่เป็นผู้สร้างข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (transmitter)

อุปกรณ์ที่เป็นผู้สร้างข้อมูล เรียกว่า ตัวรับ (receiver) ในอุปกรณ์บนบัส I^2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนบัส I^2C ที่ทำหน้าที่เป็นตัวส่งเพียงอย่างเดียว

อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการติดต่อบนบัส I^2C เรียกว่า มาสเตอร์ (master)

อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนบัส I^2C เรียกว่า สเลฟ (slave)

ข้อกำหนด 2 ประการสำคัญของการติดต่อบนบัส I^2C คือ

1. การถ่ายทอดข้อมูลจะเกิดขึ้นได้เมื่อบัสว่างเท่านั้น

2. ในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้ อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

2.1.4 สภาวะที่เกิดขึ้นบนบัส I^2C

มีด้วยกัน 5 สภาวะ ดังนี้

1. บัสว่าง (BUS not busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้ง คู่ นั่นหมายความว่าการถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้

2. เริ่มต้นการถ่ายทอดข้อมูล (Start data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากสูงไปต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น(START)

3. หยุดการถ่ายทอดข้อมูล (Stop data transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจากต่ำไปสูง ในขณะที่สาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. ข้อมูลดำรงอยู่บนบัส (Data valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA คือ ข้อมูลที่ทำการถ่ายทอดเมื่อสาย SCL เป็นลอจิกสูง สถานะที่สาย SDA ต้องคงที่เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น 0 หรือ 1 ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบรูณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ช่วงเวลาที่สาย SCL มีสถานะลอจิกสูงหากมีการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูลจะแปลความหมายเป็นสภาวะหยุด หรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น

5. รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากการถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบรูณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิต เรียกว่า บิตรับรู้ (Acknowledge) สถานะเป็นลอจิกสูง หลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษ ซึ่งสัมพันธ์ กับสัญญาณนาฬิกา เพื่อตอบสนองบิตรับรู้ที่ส่งมาจากตัวส่ง ทางด้านตัวรับจัดส่งบิตรับรู้ที่มีสถานะลอจิกต่ำลงบนบัส อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อ หรือกำลังติดต่ออยู่ในขณะนั้น ก็จะกำเนิดบิตรับรู้ เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว

รูปที่ 2.1 ไดอะแกรมแสดงสถานะต่าง ๆ บนระบบบัส I2C

2.2 ตัวประมวลผลไมโครคอนโทรเลอร์

PIC คือไมโครคอนโทรลเลอร์ตระกูลหนึ่ง ผลิตขึ้นมาโดยบริษัทไมโครชิปต่างๆ โดย PIC ย่อมาจากคำว่า (Peripheral Interface Controller) ซึ่งภายใน PIC ประกอบด้วย หน่วยความจำโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Program Memory) หน่วยความจำข้อมูล (Data Memory) พอร์ตอินพุต (Input Port) พอร์ตเอาต์พุต (Output Port) ทำให้ PIC เหมือนเป็นไมโครคอนโทรลเลอร์ตัวหนึ่ง นอกจากนี้ภายใน PIC ยังมี I^2C , PWM , A/D ซึ่งได้ถือว่าเป็นคุณสมบัติพิเศษของ PIC ที่แตกต่างไปจากไมโครคอนโทรลเลอร์ตัวอื่นๆ การรวมทุกสิ่งทุกอย่างไว้ในตัว PIC ทำให้สามารถนำไปทำงานได้อย่างง่ายและสะดวก เพียงต่อแหล่งจ่ายไฟ ป้อนสัญญาณนาฬิกา และเขียนโปรแกรมควบคุม PIC ก็สามารถควบคุมอุปกรณ์ภายนอกผ่านพอร์ตอินพุตและพอร์ตเอาต์พุตได้

2.2.1 ชนิดของไมโครคอนโทรลเลอร์ตระกูล PIC

ไมโครคอนโทรลเลอร์ตระกูล PIC ที่นิยมใช้กันอยู่ในปัจจุบันมีอยู่หลายเบอร์ด้วยกัน โดยสามารถเป็นกลุ่มต่างๆได้ดังนี้ กลุ่มที่ขึ้นต้นด้วย PIC10, PIC12, PIC14, PIC16, PIC17 และ PIC18 ซึ่งในแต่ละกลุ่มยังแยกเป็นเบอร์ต่างๆ อีกหลายเบอร์ แต่กลุ่มที่ได้รับความนิยมมากที่สุดมีอยู่ 3 กลุ่มคือ กลุ่มที่ขึ้นต้นด้วย PIC16 PIC17 และ PIC18 สำหรับทั้ง 3 กลุ่มนี้จะมีคุณสมบัติที่แตกต่างกันออกไป เช่น ขนาดของหน่วยความจำโปรแกรม ขนาดของหน่วยความจำข้อมูล จำนวนคำสั่งแอสแซมบลี และจำนวนพอร์ต แต่โครงสร้างและสถาปัตยกรรมจะคล้ายกัน สำหรับรายละเอียดดูได้จากเว็บไซต์ www.microchip.com ไมโครคอนโทรลเลอร์ตระกูล PIC แบ่งตามชนิดของหน่วยความจำ แบ่งออกได้ 3 กลุ่ม คือ

2.2.1.1 กลุ่มที่ 1 PIC ที่มีหน่วยความจำโปรแกรมได้ครั้งเดียว

หน่วยความจำโปรแกรมกลุ่มนี้ เรียกว่า OTP (one time Programmable) เป็นชิปที่ราคาถูกที่สุด เนื่องจากชิปแบบ OPT สามารถโปรแกรมไปได้ครั้งเดียวเท่านั้น ไม่สามารถแก้ไขได้อีก ดังนั้น การนำชิปประเภทนี้มาใช้งานต้องแก้ไขข้อผิดพลาดให้ถูกต้องก่อน ชิปประเภทนี้จะเหมาะกับงานที่ได้รับการพัฒนาจนไม่พบจุดบกพร่องต่างๆ อีก เพราะต้นทุนจะต่ำกว่าหน่วยความจำประเภทอื่น ชิปแบบ OTP จะมีตัวอักษรตัว C แสดงบนตัวชิป เช่นเบอร์ PIC16C62, PIC16C74, PIC16C84

40-LEAD PDIP

"P" OR "PL"

รูปที่ 2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC แบบ OTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1.2 กลุ่มที่ 2 PIC ที่มีหน่วยความจำโปรแกรมได้หลายครั้งแบบอีพรอม

หน่วยความจำโปรแกรมกลุ่มนี้ เรียกว่า EPROM (Erasable Programmable ROM) เป็นชิปที่หน่วยความจำโปรแกรม เมื่อโปรแกรมไปแล้วสามารถลบ และเขียนเข้าไปใหม่ได้อีก โดยใช้แสงอัลตราไวโอเลตหรือแสงยูวี (UV : Ultra Violet) ซึ่งด้านบนของชิปจะมีขอบกระจกเพื่อให้แสงยูวีส่องผ่านเข้าไปในตัวชิป โดยใช้เวลาประมาณ 5-10 นาที สำหรับชิปแบบนี้จะมีข้อจำกัดในเรื่องการลบโปรแกรม เนื่องจากการลบโปรแกรมด้วยแสงยูวีหลายๆครั้งจะเกิดอาการด้าน ทำให้โปรแกรมไม่เข้า ชิปแบบ EPROM จะมีตัวอักษร JW แสดงบนตัวชิป หรือมีกรอบกระจกอยู่บนชิป เช่นเบอร์ PIC12C508

40-LEAD CERDIP

"JW"

รูปที่ 2.3 ไมโครคอนโทรลเลอร์ตระกูล PIC แบบ EPROM

2.2.1.3 กลุ่มที่ 3 PIC ที่มีหน่วยความจำโปรแกรมได้หลายครั้งแบบแฟลตหรืออีอีพรอม

หน่วยความจำโปรแกรมกลุ่มนี้ เรียกว่า Flash หรือ EEPROM (Electronically Erasable Programmable ROM) เป็นชิปที่ได้รับความนิยมมากที่สุดเนื่องจากหน่วยโปรแกรมสามารถอ่านเขียนและลบด้วยสัญญาณไฟฟ้า ซึ่งใช่เวลาในการลบไม่มาก และสมามารถเขียนและลบได้หลายพันครั้ง ทำให้สะดวกในการแก้ไขปรับปรุงและเปลี่ยนแปลงโปรแกรม ชิปแบบแฟลตจะมีตัวอักษร F แสดงบนตัวชิป เช่นเบอร์ PIC16F84 และ PIC18F458

40-LEAD PDIP

"P" OR "PL"

รูปที่ 2.4 ไมโครคอนโทรลเลอร์ตระกูล PIC แบบ Flash หรือ EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล PIC ในแต่ละกลุ่ม

การแบ่งกลุ่มของ PIC จะแบ่งตามตัวเลขที่ขึ้นต้น เช่น PIC12, PIC16, และ PIC17 โดยแต่ละกลุ่มจะมีเบอร์ต่าง แยกออกไปอีกหลายเบอร์ต่างๆ แยกออกไปอีกหลายเบอร์ ซึ่งสามารถแยกออกเป็น 6 กลุ่ม คือ

2.2.2.1 ไมโครคอนโทรลเลอร์ตระกูล PIC16C5X

เป็นชิปที่ผลิตออกมาในยุคแรกๆ เหมาะกับงานที่ไม่ยุ่งยากและซับซ้อน มีคำสั่งใช้งานเป็นภาษาแอสแซมบลี 33 คำสั่ง มี I/O Timer และ Watchdog แต่ไม่มี I^2C หรือ Serial ซึ่งต้องเขียนโปรแกรมขึ้นมาเอง ชิปในกลุ่มนี้มีหลายเบอร์ แต่ละเบอร์จะมีจำนวนขาและหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลต่างกัน

2.2.2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC16CXXX

จากข้อจำกัดหลายๆอย่างของ PIC16C5X บริษัทไมโครชิปจึงได้ทำการพัฒนาและปรับปรุงเป็นตระกูล PIC16CXXX โดยเป็นชิปแบบ 18 ขา มีคำสั่งใช้งานเป็นภาษาแอสแซมบลี 35 คำสั่ง มี Timer เพิ่มขึ้น บางเบอร์มีมากกว่า 1 ตัว มีพอร์ต I^2C และพอร์ต USART ทำให้สามารถติดต่อกับอุปกรณ์ภายนอกได้ง่ายขึ้น เขียนโปรแกรมควบคุมได้ง่ายขึ้น นอกจากนี้ยังได้ทำการเพิ่มขนาดของหน่วยความจำให้มีขนาดใหญ่ขึ้นด้วย

2.2.2.3 ไมโครคอนโทรลเลอร์ตระกูล PIC12CXXX และ PIC17CXXX

เป็นชิปขนาดเล็กที่มีเพียง 8 ขา เท่านั้น ซึ่งเหมาะกับงานเล็กๆ ขา มีคำสั่งใช้งานเป็นภาษาแอสแซมบลี 33 และ 35 คำสั่ง สำหรับจุดเด่นของPIC กลุ่มนี้คือ มีสัญญาณนาฬิกาหรือออสซิลเลเตอร์ (Oscillator) ขนาด 4 MHz อยู่ภายในชิปทำให้ไม่ต้องต่อออสซิลเลเตอร์ภายนอก และมีหน่วยความจำข้อมูลภายในเป็นแบบ EEPROM แต่ส่วนของหน่วยความจำโปรแกรมยังเป็นแบบ OTP และ EPROM หลังจากได้ผลิต PIC16CXXX และทางบริษัทไมโครชิปก็ได้ผลิต PIC12FXXX ขึ้นมาซึ่งเป็นแบบแฟลชทำให้สามารถเขียนโปรแกรมและลบโปรแกรมได้หลายครั้ง

2.2.2.4 ไมโครคอนโทรลเลอร์ตระกูล PIC17CXXX

PIC17CXXX เป็นชิปที่ออกมาพร้อมกันกับ PIC16CXXX แต่ต่างกันที่ PIC17CXXX จะมีความสามารถสูง มีจำนวนขากกว่า มีคำสั่งใช้งานเป็นภาษาแอสแซมบลี 58 คำสั่ง มีคำสั่งการคูณ หาร รวมทั้งขนาดของหน่วยความจำโปรแกรม หน่วยความจำโปรแกรมที่มีขนาดใหญ่ขึ้น และยังสามารถต่อกับหน่วยความจำภายนอกได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2.5 ไมโครคอนโทรลเลอร์ตระกูล PIC16FXXX

เป็น PIC ที่ได้รับความนิยมมาก เพราะเป็นชิปรุ่นแรกที่มีหน่วยความจำโปรแกรมเป็นแบบแฟลต และมีหน่วยความจำข้อมูลแบบ EEPROM ทำให้สามารถพัฒนาโปรแกรมได้ง่าย ซึ่ง PIC เบอร์ PIC16FXXX สนับสนุนการทำงานแบบอินเซอร์กิตดีบักเกอร์ (In Circuit Debugger) ทำให้ไม่ต้องซื้ออีพรอมอีมูลเตอร์ (EPROM Emulator) ซึ่งมีราคาแพง กว่า มีคำสั่งใช้งานเป็นภาษาแอสแซมบลี 35 คำสั่ง และมีวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (A/D) ขนาด 10 บิตอยู่ภายในด้วย

2.2.2.6 ไมโครคอนโทรลเลอร์ตระกูล PIC18CXXX และ PIC18FXXX

เป็น PIC อีกแบบหนึ่งที่ได้รับความนิยม เนื่องจาก PIC18CXXX และ PIC18FXXX ไม่ว่าจะเป็นคำสั่งภาษาแอสแซมบลี 77 คำสั่ง หรือหน่วยความจำโปรแกรมที่มีขนาดใหญ่ขึ้น ทำให้สามารถรองรับและการเขียนโปรแกรมภาษาซีได้

2.2.3 สถาปัตยกรรมไมโครคอนโทรลเลอร์ตระกูล PIC18F458

จากที่ได้กล่าวมาแล้วว่า ไมโครคอนโทรลเลอร์ตระกูล PIC18F458 ได้รับความนิยมมากที่สุดเพราะถือว่าเป็นชิปรุ่นแรกที่มีหน่วยความจำโปรแกรมเป็นแบบแฟลต และมีหน่วยความจำข้อมูลแบบ EEPROM แต่ชิปในตระกูลนี้ก็มีหลายเบอร์ให้เลือกใช้ตามขนาดของงาน และความสามารถของชิปแต่ละตัว เบอร์ PIC18F458 เป็นชิปที่มีการใช้กันอย่างแพร่หลาย หาซื้อง่ายและราคาไม่แพงนัก และมีคุณสมบัติหลายประการที่เหมาะสม ซึ่งคุณสมบัติหลักๆ ของ PIC18F458 มีดังนี้

1. มีคำสั่งที่เป็นภาษาแอสแซมบลี 35 คำสั่ง
2. ใน 1 คำสั่งใช้เวลาทำงาน 1 ถึง 2 ไซเคิล
3. ทำงานได้สูงสุดที่สัญญาณนาฬิกาตั้งแต่ไฟตรงถึง 40 MHz
4. ทำงานแบบ Pipe-line สามารถทำงาน 2 อย่างในเวลาเดียวกันได้
5. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 32 Kword (1 word = 14 บิต)
6. มีหน่วยความจำข้อมูล (Data Memory RAM) ขนาด 15k ไบต์
7. มีหน่วยความจำข้อมูลแบบ EEPROM ขนาด 256 ไบต์
8. ตอบสนองการอินเทอร์รัปได้ทั้งหมด 21 แหล่ง
9. มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
10. มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up และ Watchdog Timer
11. มีระบบ Code Protection กันการคัดลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

12. มีโหมดประหยัดพลังงาน (Sleep Mode)

13. สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้

14. สามารถโปรแกรมด้วยไฟ +5 VDC ได้

15. ใช้การโปรแกรมแบบ In Circuit Serial Programming

16. ทำงานที่ไฟเลี้ยง 2 VDC ถึง 5.5 VDC

17. Current Sink และ Current Source อยู่ที่ 25 mA

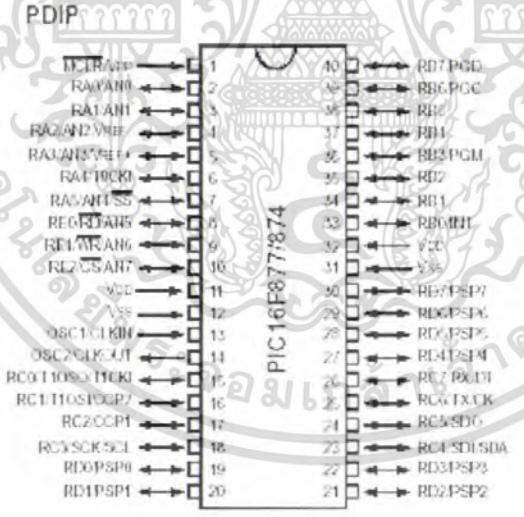
18. มี Timer/ Counter 4 ตัว คือ Timer 0 ขนาด 8 บิต , Timer 1 ขนาด 16 บิต และ Timer 2 ขนาด 8 บิต,Timer 3

19. มีโมดูล Capture/compare/PWM (Pulse Width Modulation) 2 ชุด

20. มีวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (A/D) ขนาด 10 บิต

21. มี I/O พอร์ตทั้ง 5 พอร์ต แต่ละพอร์ตมีจำนวนบิตไม่เท่ากัน

2.2.4 โครงสร้างขาสัญญาณของไมโครคอนโทรลเลอร์ตระกูล PIC18F458



รูปที่ 2.5 ขาสัญญาณของไมโครคอนโทรลเลอร์ตระกูล PIC18F458

ไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC18F458 เป็นไมโครคอนโทรลเลอร์ขนาด 40 ขา มีขาสัญญาณต่างๆดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. MCLR/Vpp : Mater Clear (Reset) Input / Programming Voltage ทำหน้าที่ เป็นขาสัญญาณรีเซท (Reset) เมื่อขานี้ได้รับลอจิก 0 ไมโครคอนโทรลเลอร์จะถูกรีเซต และทำหน้าที่เป็นขาสัญญาณรับแรงดัน ขณะทำการบันทึกโปรแกรมลงหน่วยความจำของไมโครคอนโทรลเลอร์
2. VDD : Positive Supply (+2.00 ถึง 5.5 V) ทำหน้าที่เป็นขาไฟเลี้ยงของไมโครคอนโทรลเลอร์
3. VSS : Ground ทำหน้าที่เป็นขากราวด์
4. OSC1/CLKIN : Oscillator Crystal Input / External Clock Source Input
5. OSC2/CLKOUT ; Oscillator Crystal Output / External Clock Source Output
6. RA0 – RA5 : พอร์ต A มีจำนวน 6 ขา เป็นพอร์ตแบบสองทิศทาง (Bi Directional I/O) คือเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุตใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.1 ดังนี้

ตารางที่ 2.1 หน้าที่ของขาสัญญาณของพอร์ต A

พอร์ต	สัญญาณ	หน้าที่
RA0	AN0	รับสัญญาณอินพุตสำหรับ ADC ช่อง 0
RA1	AN1	รับสัญญาณอินพุตสำหรับ ADC ช่อง 1
RA2	AN2	รับสัญญาณอินพุตสำหรับ ADC ช่อง 2
RA3	AN3	รับสัญญาณอินพุตสำหรับ ADC ช่อง 3
RA4	TOCK1	รับสัญญาณ Input Clock ของ Timer
RA5	AN4	รับสัญญาณอินพุตสำหรับ ADC ช่อง 4
	SS	รับสัญญาณ Slave Select จากการติดต่อของ Serial Port

7. RB0-RB7 : พอร์ต B มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการรับส่งข้อมูล นอกจากนี้ยังทำหน้าที่รับสัญญาณอินพุตจากการอินเตอร์รัป (Interrupt) จากภายนอกด้วย แสดงดังตารางที่ 2.2 นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 หน้าที่ของขาสัญญาณของพอร์ต B

พอร์ต	สัญญาณ	หน้าที่
RB0	INT	รับสัญญาณอินพุตจากการอินเตอร์รัปจากภายนอก
RB3	PGM	รับสัญญาณอินพุตแรงดันต่ำในการบันทึกโปรแกรม (ถ้ามีการ Enable)
RB6	PGC	ขาสัญญาณนาฬิกาในการบันทึกโปรแกรม
RB7	PGD	ขาสัญญาณข้อมูลในการบันทึกโปรแกรม

8. RC0 – RC7 : พอร์ต C มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการรับส่งข้อมูล นอกจากนี้ยังทำหน้าที่อื่น ๆ แสดงดังตารางที่ 2.3

ตารางที่ 2.3 หน้าที่ของขาสัญญาณของพอร์ต C

พอร์ต	สัญญาณ	หน้าที่
RC0	T1OSO	ขาสัญญาณเอาต์พุตของวงจรออสซิลเลเตอร์ของ Timer 1
	T1CK1	ขาสัญญาณเอาต์พุตของสัญญาณนาฬิกาของ Timer 1
RC1	T1OSI	ขาสัญญาณอินพุตของวงจรออสซิลเลเตอร์ของ Timer 1
	CCP2	ขาสัญญาณเอาต์พุตของโมดูล CCP2 (Capture2,compare2,PWM2)
RC2	CCP1	ขาสัญญาณเอาต์พุตของโมดูล CCP2 (Capture1,compare1,PWM1)
RC3	SCK	ขาสัญญาณนาฬิกาของ SPI
	SCL	ขาสัญญาณนาฬิกาของ I^2C
RC4	SD1	ขาสัญญาณอินพุตและ Serial Data ของระบบ SPI
	SDA	ขาข้อมูลของระบบบัส I^2C
RC5	SDO	ขาสัญญาณเอาต์พุตและ Serial Data ของระบบ SPI
RC6	TxD	ขาส่งข้อมูลแบบ Serial Port
	CK	ขาสัญญาณนาฬิกา แบบ Synchronize
RC7	RxD	ขารับข้อมูลแบบ Serial Port
	DT	ขารับข้อมูลแบบ Synchronize

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. RD0 – RD7 : พอร์ต D มีจำนวน 8 ขา ขนาด 8 บิต เป็นพอร์ตแบบสองทิศทาง ใช้ในการรับส่งข้อมูล นอกจากนี้ยังทำหน้าที่อื่น ๆ แสดง
ดังตารางที่ 2.4

ตารางที่ 2.4 หน้าที่ของขาสัญญาณของพอร์ต D

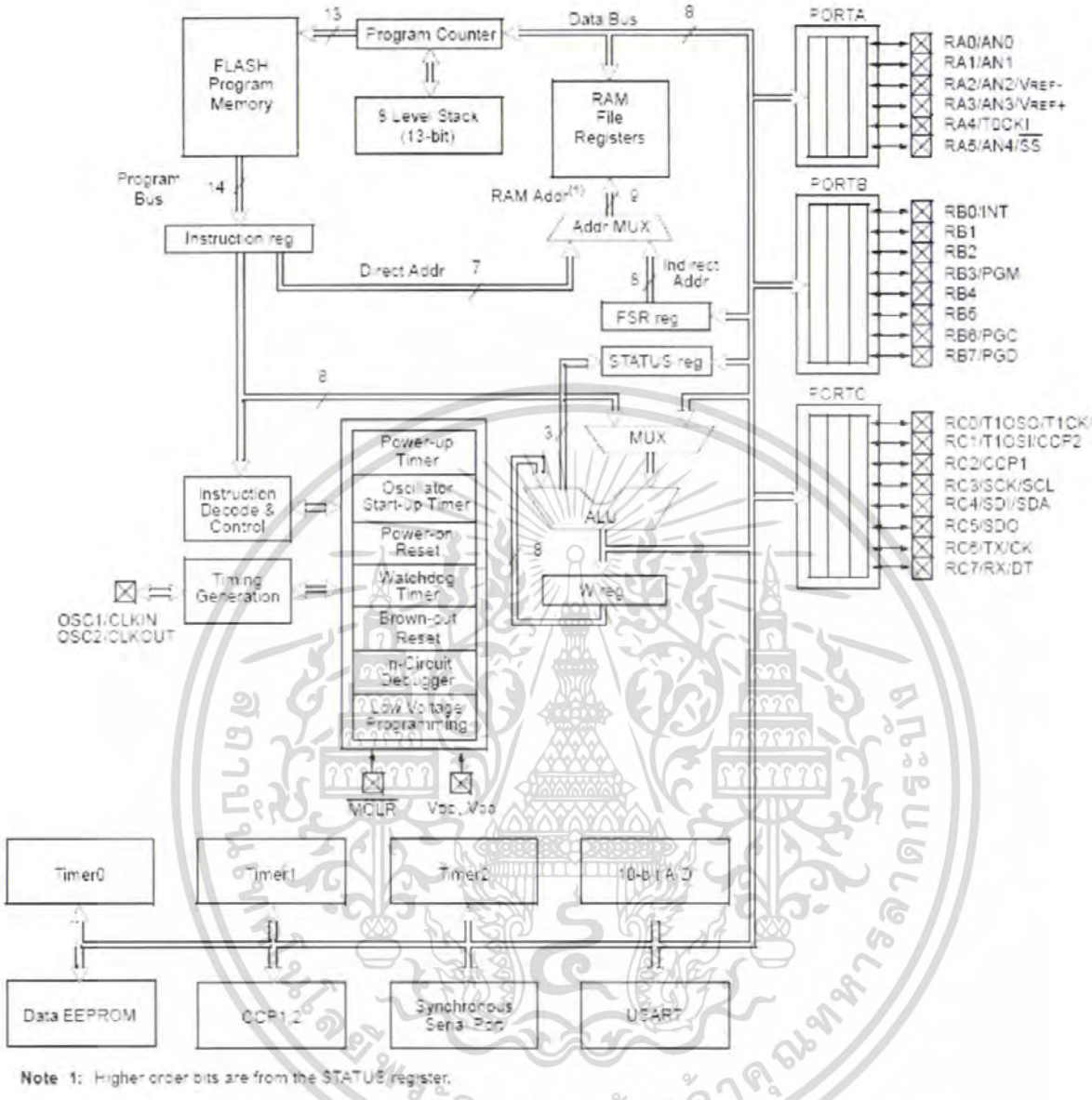
พอร์ต	สัญญาณ	หน้าที่
RD0	SPS0	ขาสัญญาณขยายพอร์ตขนาน บิต 0
RD1	SPS1	ขาสัญญาณขยายพอร์ตขนาน บิต 1
RD2	SPS2	ขาสัญญาณขยายพอร์ตขนาน บิต 2
RD3	SPS3	ขาสัญญาณขยายพอร์ตขนาน บิต 3
RD4	SPS4	ขาสัญญาณขยายพอร์ตขนาน บิต 4
RD5	SPS5	ขาสัญญาณขยายพอร์ตขนาน บิต 5
RD6	SPS6	ขาสัญญาณขยายพอร์ตขนาน บิต 6
RD7	S7PS	ขาสัญญาณขยายพอร์ตขนาน บิต 7

10. RE0-RE2 : พอร์ต E มีจำนวน 3 ขา เป็นพอร์ตแบบสองทิศทาง ใช้ในการส่งและรับข้อมูล นอกจากนี้ยังทำหน้าที่อื่นๆ แสดงดังตารางที่ 2.5

ตารางที่ 2.5 หน้าที่ของสัญญาณของพอร์ต E

พอร์ต	สัญญาณ	หน้าที่
RE0	AN5	ขาสัญญาณอินพุตสำหรับ ADC ช่อง 5
	RD	ขาสัญญาณขยายพอร์ตขนานควบคุมการอ่าน
RE1	AN6	ขาสัญญาณอินพุตสำหรับ ADC ช่อง 6
	WR	ขาสัญญาณขยายพอร์ตขนานควบคุมการเขียน
RE2	AN7	ขาสัญญาณอินพุตสำหรับ ADC ช่อง 7
	CS	ขาสัญญาณขยายพอร์ตขนานควบคุมการเลือกอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Note 1: Higher order bits are from the STATUS register.

รูปที่ 2.6 โครงสร้างภายในของไมโครคอนโทรลเลอร์ PIC เบอร์ PIC18F458

สำหรับพอร์ตทั้ง 5 พอร์ตของไมโครคอนโทรลเลอร์ PIC คือ พอร์ต A พอร์ต B พอร์ต C พอร์ต D และพอร์ต E เป็นพอร์ตแบบสองทิศทาง สามารถเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุต ในการนำไปใช้งานต้องมีการกำหนดให้ขาสัญญาณของพอร์ตเป็นอินพุตหรือเอาต์พุต ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ SET_TRIS_ พอร์ต (ข้อมูลเลขฐานสิบหก)

ข้อมูลเลขฐานสิบหก หมายถึงถ้าต้องการให้บิตไหนของพอร์ตเป็นเอาต์พุตให้กำหนดข้อมูล 0 ที่บิตนั้น และถ้าต้องการให้บิตไหนของพอร์ตเป็นตัวอินพุตให้กำหนดข้อมูล 1 ที่บิตนั้น

ตัวอย่าง SET_TRIS_ A (0x0FF)

หมายถึง กำหนดให้พอร์ต A ทั้ง 6 บิตเป็นพอร์ตอินพุต

บิต	7	6	5	4	3	2	1	0
พอร์ต B	-	-	RA5	RA4	RA3	RA2	RA1	RA0
ข้อมูล	1	1	1	1	1	1	1	1

ตัวอย่าง SET_TRIS_B (0x0F)

หมายถึง กำหนดให้พอร์ต B บิต RB0-RB3 เป็นพอร์ตอินพุต

บิต RB4-RB7 เป็นพอร์ตเอาต์พุต

บิต	7	6	5	4	3	2	1	0
พอร์ต B	RB7	RB6	RB5	RB	RB3	RB2	RB1	RB0
ข้อมูล	0	0	0	0	1	1	1	1

สำหรับพอร์ต A และพอร์ต E นอกจากจะทำหน้าที่เป็นพอร์ตอินพุตและเอาต์พุตแล้ว ยังทำหน้าที่รับสัญญาณอินพุตของวงจรแปลงสัญญาณแอนะล็อกเป็นสัญญาณดิจิตอลด้วย แต่ถ้าต้องการนำพอร์ต A และพอร์ต B มาทำหน้าที่เป็นพอร์ตอินพุตและพอร์ตเอาต์พุตเพียงอย่างเดียว ต้องทำการกำหนดการปิดพอร์ตแอนะล็อกก่อทุกครั้งโดยใช้คำสั่ง 2 คำสั่ง ดังนี้

- SETUP_ADC_PORTS(NO_ANALOGS);
- SETUP_ADC(ADC_OFF);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.2.5วงจรสร้างสัญญาณนาฬิกา

ไมโครคอนโทรลเลอร์จะมีสัญญาณนาฬิกาเป็นตัวควบคุมก่อนการทำงาน ดังนั้นหากต้องการให้ไมโครคอนโทรลเลอร์ทำงานต้องต่อวงจรกำเนิดสัญญาณนาฬิกาหรือออสซิลเลเตอร์ ซึ่งไมโครคอนโทรเลอร์ตระกูล PIC ก็เช่นกัน โดย PIC สามารถเลือกใช้ออสซิลเลเตอร์ภายในหรือภายนอกได้ สำหรับออสซิลเลเตอร์ภายในจะเป็นประเภท RC ออสซิลเลเตอร์ ซึ่งมีความถี่คงที่ 4 MHz ที่แรงดัน 5 โวลต์ อุณหภูมิที่ 25 องศาเซลเซียส โดยวงจรออสซิลเลเตอร์ประเภท RC ออสซิลเลเตอร์ ความถี่จะเปลี่ยนแปลตามอุณหภูมิ แต่ถ้าเป็นวงจรออสซิลเลเตอร์ภายนอกความถี่จะขึ้นอยู่กับอุปกรณ์ภายนอกที่นำมาต่อ วงจรสร้างสัญญาณนาฬิกาหรือวงจรออสซิลเลเตอร์ภายนอกมี 3 แบบดังนี้

แบบที่ 1 สร้างจากเซลามิคเรโซเนเตอร์ (Ceramic Resonators) เหมาะสำหรับงานที่ใช้ความถี่ไม่สูงมากนัก เพราะมีความถูกเมื่อเทียบกับออสซิลเลเตอร์ประเภทอื่น โดยการต่อใช้งาน ขากลางจะเป็นขากราวด์ (Ground ; GND) ส่วนอีกสองขาต่อเข้ากับ CLKIN และ CLKOUT ของ PIC

รูปที่ 2.7 เซลามิคเรโซเนเตอร์

แบบที่ 2 สร้างจากผลิตแร่คริสตอล (Quartz Crystal Oscillator) ใช้ตัวคริสตอลกำเนิดสัญญาณความถี่กำลังต่ำออกมา โดยจะต่อกับตัวเก็บประจุ (Capacitor) ลงกราวด์ (GND) ทั้ง 2 ขาดังรูป และทั้ง 2 ขาต่อเข้ากับขา OSC1/ CLKIN และ OSC2/ CLKOUT ของ PIC
ซึ่งคริสตอลจะมีราคาแพงกว่าเซลามิคเรโซเนเตอร์ แต่จะให้ความถี่ที่มีความเที่ยงตรงมากกว่า

83069
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

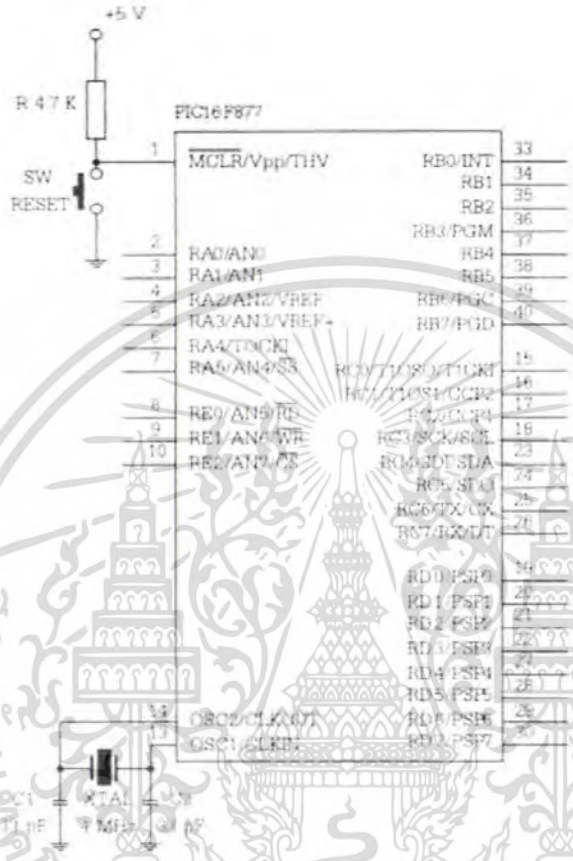
รูปที่ 2.8 การต่อคริสตอลและตัวเก็บประจุเข้ากับ PIC

แบบที่ 3 สร้างจากวงจรกำเนิดความถี่สำเร็จรูป (Crystal Square-wave Oscillator) ซึ่งจะมีคริสตอลและวงจรอยู่ภายในตัว ทำให้ความถี่ที่ออกมามีความเที่ยงตรงมาก แต่ราคาแพง โดยมีการต่อเข้ากับ PIC ดังนี้ ขา 14 ต่อเข้ากับไฟ 5 โวลต์ ขา 7 ต่อกับกราวด์ (GND) ขา 8 ต่อกับขา CLKIN ของ PIC ขา 1 ไม่ต่อ และขา CLKOUT ของ PIC ก็ไม่ต่อเช่นกัน

รูปที่ 2.9 โครงสร้างขาของ Crystal Square-wave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 การต่อเชื่อม PIC18F458 กับวงจรกำเนิดสัญญาณนาฬิกา



รูปที่ 2.10 วงจรการเชื่อมต่อ PIC กับวงจรกำเนิดสัญญาณนาฬิกา

จากรูปที่ 1.9 วงจรกำเนิดสัญญาณนาฬิกาที่นิยมใช้กันอย่างแพร่หลาย จะเป็นการใช้ตัวคริสตอลต่อกับตัวเก็บประจุลงกราวด์ทั้งสองขา และทั้ง 2 ขาของคริสตอลต่อเข้ากับขา OSC1/ CLKIN และ OSC2/ CLKOUT ของ PIC สำหรับเลือกใช้ความถี่ของคริสตอลจะมีผลต่อโหมดของสัญญาณนาฬิกาในการเขียนโปรแกรม แสดงดังตาราง 1.6 โดยมีโหมดต่างๆ ดังนี้

โหมด LP (Low power Crystal) ทำงานที่ความถี่ตั้งแต่ 32 KHz ถึง 100 KHz

โหมด XT (Crystal/Resonator) ทำงานที่ความถี่ตั้งแต่ 100 KHz ถึง 4 MHz

โหมด HS (High power Crystal/ Resonator) ทำงานที่ความถี่ 4 MHz ถึง 40 MHz

โหมด RC (External Resistor/Capacitor) มี 2 โหมด สามารถกำหนดความถี่ได้จากตัวต้านทานและตัวเก็บประจุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 โหมดของสัญญาณนาฬิกาเมื่อใช้เซรามิคเรโซเนเตอร์

โหมด	ความถี่	C1	C2
XT	455 KHz	68-100 pF	68-100 pF
	2.0 MHZ	15-68 pF	15-68 pF
	4.0 MHZ	15-68 pF	15-68 pF
HS	8.0 MHZ	10-68 pF	10-68 pF
	16.0 MHZ	10-22 pF	10-22 pF

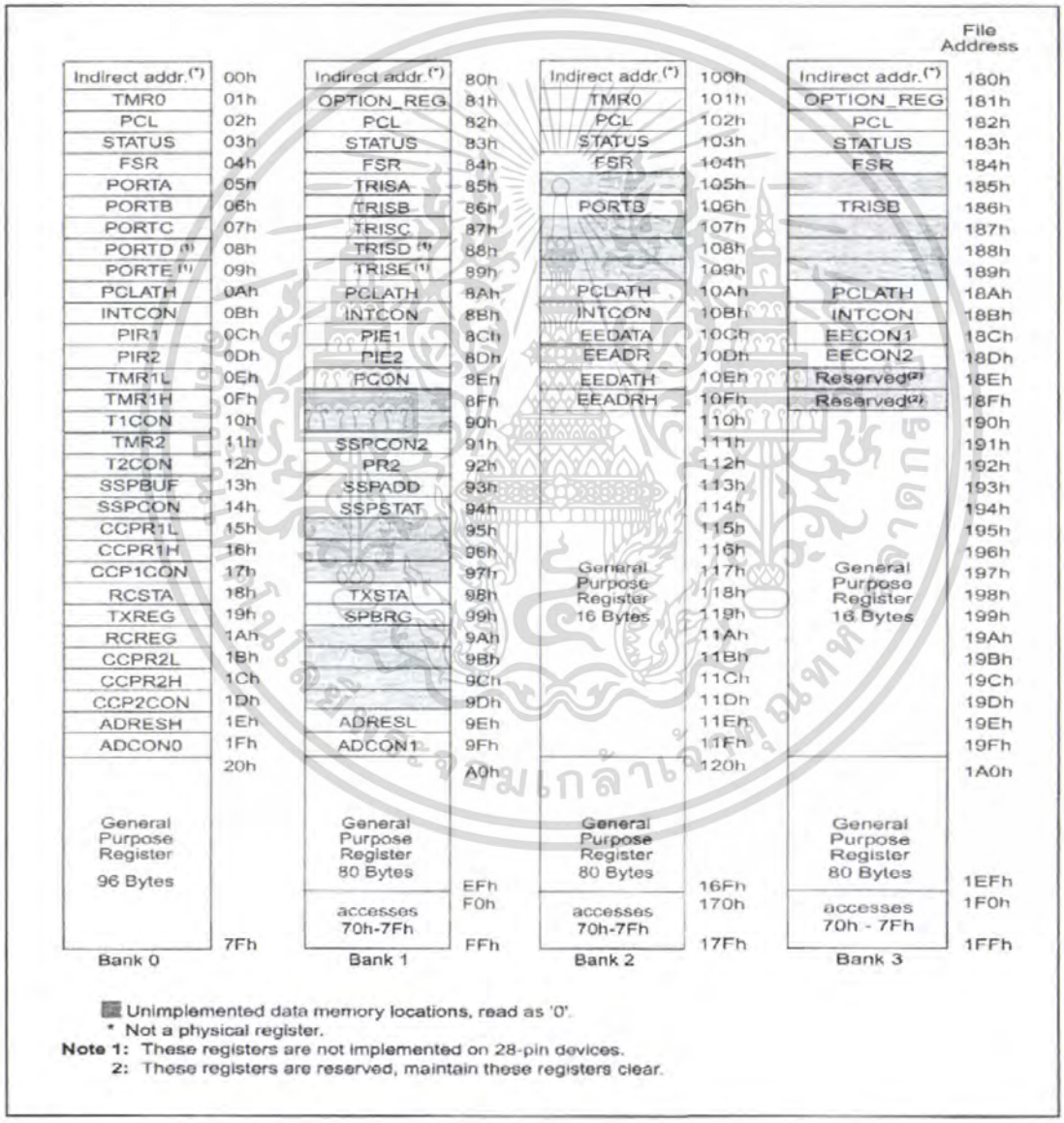
ตารางที่ 2.7 โหมดของสัญญาณนาฬิกาเมื่อใช้คริสตอล

โหมด	ความถี่	C1	C2
LP	32 KHz	33 pF	33 pF
	200 MHZ	15 pF	15 pF
XT	200 MHZ	47-68 pF	47-68 pF
	1.0 MHZ	15 pF	15 pF
	4.0 MHZ	15 pF	15 pF
HS	4.0 MHZ	15-33 pF	15-33 pF
	8.0 MHZ	15-33 pF	15-33 pF
	20.0 MHZ	15-33 pF	15-33 pF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.7 การจัดการหน่วยความจำของ PIC18F458

ไมโครคอนโทรลเลอร์ PIC18F458 มีหน่วยความจำโปรแกรมแบบแฟลต ซึ่งใช้สำหรับเก็บโปรแกรมควบคุมการทำงานของ PIC มีความจำข้อมูล (Data Memory RAM) ขนาด 15k ไบต์ หน่วยความจำข้อมูลแบบ EEPROM (Data Memory EEPROM) ขนาด 256 ไบต์ และมีการจัดการหน่วยความจำของ PIC ออกเป็นแบงก์ต่าง ๆ ได้ทั้งหมด 4 แบงก์ คือ Bank 0 - Bank 3 แต่ละแบงก์มีขนาด 128 ไบต์ ดังรูปที่ 1.10



รูปที่ 2.11 การจัดการพื้นที่หน่วยความจำของ PIC18F458

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบงก์ 0 และ แบงก์ 1 จะแบ่งออกเป็น 2 ส่วนเหมือนกันคือ ส่วนที่ 1 ขนาด 32 ไบต์ เป็นพื้นที่ของริจีสเตอร์ไฟล์ที่เรียกว่า SFR (Special Function Register) ซึ่งใช้สำหรับกำหนดเงื่อนไขการทำงานและบันทึกสถานการณ์ทำงานของไมโครคอนโทรลเลอร์ ส่วนที่ 2 ขนาด 96 ไบต์ เป็นพื้นที่หน่วยความจำสำหรับใช้งานทั่วไป ซึ่งใช้สำหรับเก็บผลลัพธ์และเงื่อนไขต่างๆ ของโปรแกรม

แบงก์ 2 และ แบงก์ 3 จะแบ่งออกเป็น 2 ส่วน แต่มีความแตกต่างจากแบงก์ 0 และ แบงก์ 1 ส่วนที่ 1 ขนาด 16 ไบต์ เป็นพื้นที่ของริจีสเตอร์ไฟล์ ส่วนที่ 2 ขนาด 112 ไบต์ เป็นพื้นที่หน่วยความจำสำหรับใช้งานทั่วไป

2.2.8 กระบวนการรีเซต

การรีเซต (Reset) ของไมโครคอนโทรลเลอร์ คือการทำให้ไมโครคอนโทรลเลอร์เริ่มต้นทำงานใหม่ ซึ่งเมื่อไมโครคอนโทรลเลอร์ถูกรีเซต จะกระโดดไปทำงานที่ตำแหน่ง 0000H ดังนั้นตำแหน่งของโปรแกรมจะต้องเก็บไว้ที่ตำแหน่งเริ่มต้นคือ 0000H การรีเซตจะช่วยแก้ปัญหาและข้อผิดพลาดของโปรแกรม ซึ่งการรีเซตของ PIC18F458 สามารถเกิดขึ้นได้ 6 กรณี คือ

1. เพาเวอร์ออนรีเซต (Power On Reset)
2. การรีเซตที่ขา MCLR ในโหมดปกติ
3. การรีเซตที่ขา MCLR ในโหมดประหยัดพลังงาน
4. การรีเซตจากวอตช์ดอกไทเมอร์ ในโหมดปกติ
5. การรีเซตจากวอตช์ดอกไทเมอร์ ในโหมดประหยัดพลังงาน
6. การรีเซตเนื่องจากไม่มีไฟเลี้ยงหรือไฟเลี้ยงลดต่ำลง

2.2.9 โหมดประหยัดพลังงาน

โหมดประหยัดพลังงาน (Sleep Mode) เป็นโหมดที่เกิดขึ้นเมื่อไมโครคอนโทรลเลอร์ไม่มีการทำงานหรือประมวลผลโปรแกรมใดๆ เพื่อช่วยลดกระแสไฟฟ้าในตัวชี เมื่อเข้าสู่โหมดนี้ ไมโครคอนโทรลเลอร์จะกินกระแสไฟต่ำมาก และจะออกจากโหมดประหยัดพลังงานเมื่อ

1. มีการรีเซตที่ขา MCLR
2. มีการอินเตอร์รัปจากไมโครคอนโทรลเลอร์
3. มีสัญญาณวอตช์ดอกไทเมอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.10 วอตช์ดอกไทเมอร์

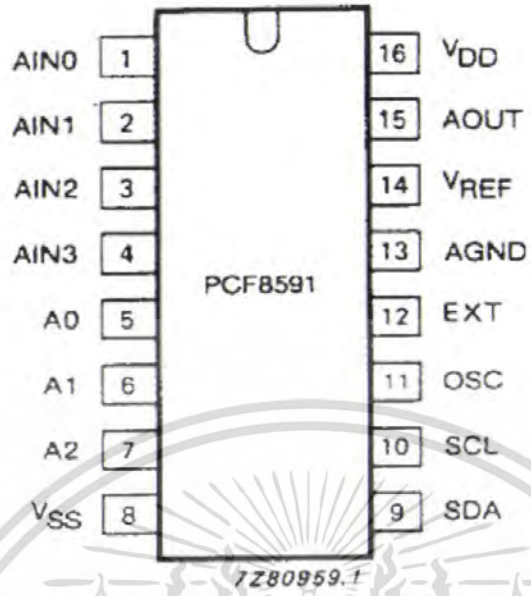
ในการทำงานของไมโครคอนโทรเลอร์บางครั้งอาจเกิดการผิดพลาดขึ้นได้ วอตช์ดอกไทเมอร์จึงทำหน้าที่เป็นตัวตรวจสอบความผิดพลาดที่เกิดขึ้น เช่น ถ้าไมโครคอนโทรเลอร์ทำงานอยู่ในเงื่อนไขบางอย่างนานเกินไปวอตช์ดอกไทเมอร์จะนับค่าเวลาของตัวเอง ถ้าเกิดข้อผิดพลาดขึ้นก็จะส่งสัญญาณไปรีเซตไมโครคอนโทรลเลอร์

2.3 ข้อมูลเบื้องต้นของ PCF8591

ในการทดลองนี้จะใช้ ไอซี ที่มีความสามารถสูงเบอร์ PCF8591 เนื่องจากในตัวมันมีวงจร ADC แบบซักเซสซีฟแอปพรอกซิเมชั่นขนาด 8 บิต สูงถึง 4 ช่องทั้งยังมีวงจร DAC อีก 1 ช่องด้วย ระบบการเชื่อมต่อเป็นแบบบัส I^2C ทำให้ใช้สายสัญญาณเพียง 2 เส้น ทั้งยังสามารถต่อพ่วงกันได้สูงสุด 8 ตัว ทำให้ได้วงจร ADC รวมสูงถึง 32 ช่อง และวงจร DAC รวม 4 ช่อง สามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง มีรายละเอียดคุณสมบัติทางเทคนิคดังนี้

- ทำงานโดยใช้แหล่งจ่ายไฟเพียงชุดเดียว
- ทำงานที่แรงดัน 2.5 V ถึง 6 V
- กินกระแสขณะอยู่ในสภาวะแสตนบายต่ำ
- ติดต่อกับไมโครคอมพิวเตอร์หรือไมโครคอนโทรลเลอร์ผ่านระบบบัส I^2C
- เลือกตำแหน่งแอดเดรสทางฮาร์ดแวร์จากขา A0 , A1 , A2 ทำให้สามารถต่อพ่วงกันได้สูงถึง 8 บิต
- อัตราการสุ่มข้อมูล (samping) ขึ้นอยู่กับความเร็วของสัญญาณนาฬิกาบนบัส I^2C
- วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล (ADC) สามารถรับสัญญาณอนาลอกได้ 4 ช่อง ทั้งยังเลือกได้ว่าทำงานแบบแยกช่องหรือทำงานเป็นวงจรดิฟเฟอร์เรนเชียล
- การอ่านค่าสามารถกำหนดให้เลื่อนช่องอินพุตโดยอัตโนมัติ
- สัญญาณอนาลอกเป็นดิจิตอลเป็นแบบซักเซสซีฟแอปพรอกซิเมชั่นขนาด 8 บิต
- มีวงจรแปลงสัญญาณดิจิตอลเป็นอนาลอกขนาด 8 บิต 1 ช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การจัดขาของอซี ADC/DAC ขนาด 8 บิตผ่านบัส I² C เบอร์ PCF8591

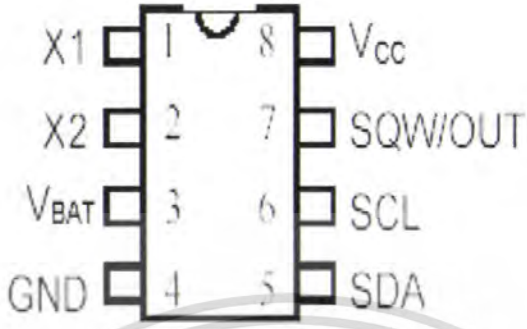
สามารถทำหน้าที่เป็นไอซีแปลงสัญญาณอนาลอกเป็นดิจิตอลขนาด 8 บิต 4 ช่องและทำหน้าที่เป็นไอซีแปลงสัญญาณดิจิตอลเป็นอนาลอกได้ในคราวเดียวกัน ด้านการควบคุมผ่านระบบบัส I² C ทำให้สามารถต่อพ่วงไอซี PCF8591 ได้สูงสุดถึง 8 ตัว รองรับการอ่านค่าสัญญาณอนาลอกอินพุตได้สูงสุดถึง 32 ช่อง ด้วยการกำหนดแอดเดรสจากขา A0 , A1 และ A2 การจัดขา PCF8591 แสดงได้ดังรูป

2.4 DS1307 ไอซีร้างฐานเวลาจริงหรือรีลไทม์คล็อก (RTC)

ผู้ผลิตคือ ดัลลัสเซมิคอนดักเตอร์ (Dallas semiconductor) มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรเลอร์โดย DS1307 จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที, นาที , ชั่วโมง , วันที่ (DATE) วันในสัปดาห์ (DAY), เดือนและปี โดยสามารถปรับวันเดือนและปีให้ตรงตามปฏิทินได้อย่างถูกต้องรวมถึงการกำหนดวันในปีอธิกสุรทินด้วย คุณสมบัติทางเทคนิคที่สำคัญดังนี้

- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปีรวมถึงการปรับวันในปีอธิกสุรทินด้วย สามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสศักราช 2100
- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายในสามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส I² C
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาไว้ได้แม้ไม่มีไฟเลี้ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 การจัดขาของไอซี DS1307 ไอซีสร้างฐานเวลาจริง (RTC)

2.4.1การทำงานของ DS1307

ไอซี จัดการเชื่อมต่อในแบบบัส I² C โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ I² C ในรูปที่แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างข้อมูลเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUTจะมีสัญญาณพัลส์เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการอีนาเบิลวงจรกำหนดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุมค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่า คือ 1Hz , 4.096kHz , 32 kHz พร้อมกันนั้นจะมีการเก็บค่าของเวลาไว้ในหน่วยความจำนอนโวลาไทล์แรม ซึ่งมีขนาดรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า 1.25 x V_{BAT} ก็จะควบคุมให้หยุดการทำงาน รีเซตค่าตัวนับแอดเดรสภายใน ทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นในการใช้งาน DS1307 ต้องระมัดระวังอย่าให้ไฟเลี้ยงต่ำกว่า 1.25 x V_{BAT} หรือประมาณ 3.75 V ในกรณีที่ใช้ V_{BAT} เท่ากับ 3 V ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า V_{BAT} ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อค่าของเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

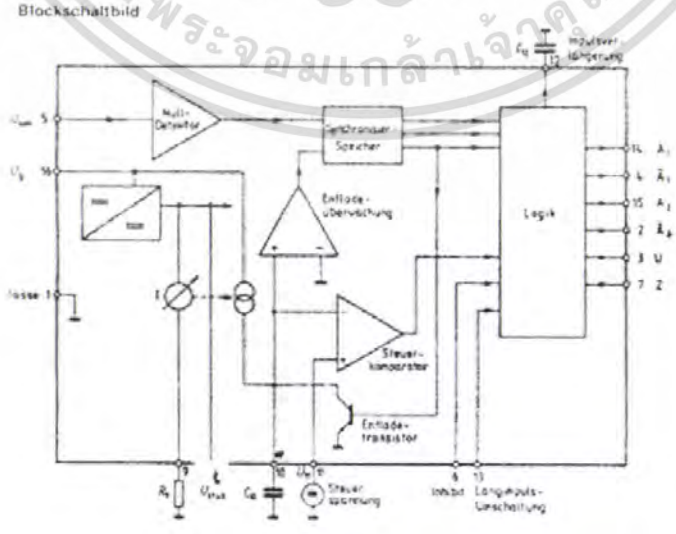
วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัส I²C เป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบบัส I²C

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 วงจรการสร้างสัญญาณกระตุ้น

ไอซี TCA 785 ใช้ในการควบคุมลำดับของการทริกซึ่งมีวงจรหลัก คือวงจรการควบคุมแรงดันภายใน (Vref)ดังนั้นแหล่งจ่ายไฟที่จ่ายให้ TCA 785 สามารถรับแรงดันได้ในช่วง คือ V = 8-18 V โดยขา 1 จะต้องลงกราวด์และขา 16 ต่อ แหล่งจ่ายไฟ 15 V เมื่อไม่มีโหลดจะกินกระแสประมาณ 10 mA ที่ขา 8 จะต่อ Capacitor และต่อกราวด์เพื่อป้องกันสัญญาณรบกวน สัญญาณซิงโครไนซ์เซชั่น (Synchronization) ซึ่งจะนำแรงดันไฟสลับจากเฟสภายนอกที่ต้องการควบคุมที่ต่อผ่านตัวต้านทาน 220 kΩ ต่อเข้าที่ขา 9 เป็นตัวปรับกระแสและต่อตัวคาปาซิเตอร์กับขา 10 ขนาด 100 nF เป็นตัวชาร์จให้ให้เป็นเชิงเส้น ตัวต้านทานและคาปาซิเตอร์ที่ขา 9 และ 10 จะทำหน้าที่เป็น RC Network ใช้กำหนดความชันของสัญญาณฟันเลื่อยซึ่งจะต้องใช้ในการเปรียบเทียบค่าแรงดันไฟตรงจากภายนอกที่มาจากวงจรควบคุมที่ขา 11 นั่นคือแรงดันที่ขา 11 จะไปตัดกับสัญญาณสามเหลี่ยมที่ขา 10 ซึ่งเป็นสัญญาณเอาท์พุตที่ออกมาจะเป็นพัลส์ขนาดความกว้างเท่าขนาดช่วงตัดยอดสัญญาณสามเหลี่ยม ซึ่งเป็นสัญญาณทริกเกอร์ที่ขา 14 และขา 15 ซึ่งเราสามารถสร้างสัญญาณทริกตั้งแต่ 0 – 180 องศาโดยปรับแรงดันที่ขา 11 ซึ่งถ้าปรับแรงดันต่ำพัลส์ที่ได้จะเริ่มต้นเข้าใกล้ 0 องศา หรือถ้าปรับแรงดันให้สูงโดยปรับเพียงปลายยอดของสัญญาณพัลส์ที่ได้จะเริ่มต้นเข้าใกล้ 180 องศา

จะเห็นว่าสัญญาณควบคุมแรงดัน (Control Voltage) ที่ขา ตัดกับสัญญาณฟันเลื่อยที่ขา 10 พัลส์เอาท์พุตที่ได้จะมีความกว้างเท่ากับช่วงตัดสัญญาณ ซึ่งเราสามารถรถขนาดความกว้าง ของพัลส์ลงได้โดยค่ามุมทริกไม่เปลี่ยนแปลง ซึ่งทำได้โดยปรับค่าคาปาซิเตอร์ 12 ซึ่งมีขนาดพัลส์จะได้ขนาดตามตารางที่

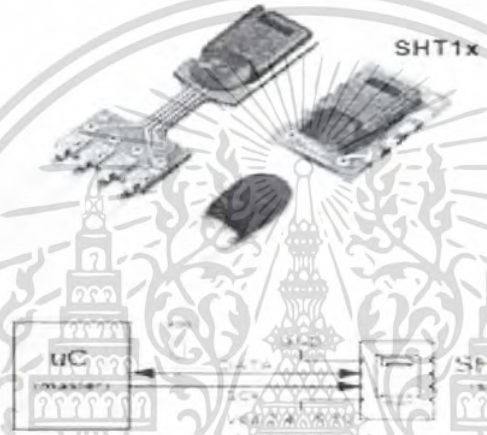


รูปที่ 2.14 บล็อกไดแกรมของ TCA 785

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 เซนเซอร์ที่ใช้วัดอุณหภูมิและความชื้น

ส่วนวัดผลทางอุณหภูมิจะวัดค่าอุณหภูมิโดยใช้เซนเซอร์ SHT 15 ซึ่ง SHT 15 สามารถใช้เป็นเซนเซอร์วัดได้ทั้งอุณหภูมิและมีตัวแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิตัล (A/D) ภายในตัว SHT 15 เป็นอุปกรณ์ที่ให้ค่าเอาท์พุทเป็นข้อมูลดิจิตอล ซึ่งค่าที่ได้ต้องนำมาผ่านการคำนวณตามสมการมาตรฐานของอุปกรณ์จึงจะได้ค่าเอาท์พุทที่แท้จริงเป็นค่าของอุณหภูมิและความชื้น



รูปที่ 2.15 รูป IC SHT 15

2.6.1 คุณสมบัติของ SHT 15 มีดังนี้

- มี Package แบบ LLC (Leadless Chip Carrier)
- สามารถวัดได้ทั้งอุณหภูมิและความชื้น
- สามารถวัดอุณหภูมิได้ตั้งแต่ 0-125 °C ความละเอียดในการวัด 0.1%RH
- สามรถวัดความชื้นสัมพัทธ์ได้ตั้งแต่ 1- 99 % RH ความละเอียดในการวัด 0.1 % RH
- ใช้แหล่งจ่ายไฟ + 5 V กินกระแสต่ำ
- ใช้สัญญาณในการควบคุม 2 เส้น คือ DATA และ CLOOK ภายใต้มาตรฐาน I^2C
- มีความแม่นยำในการวัดอุณหภูมิที่ 0.5 °C ส่วนความชื้นที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

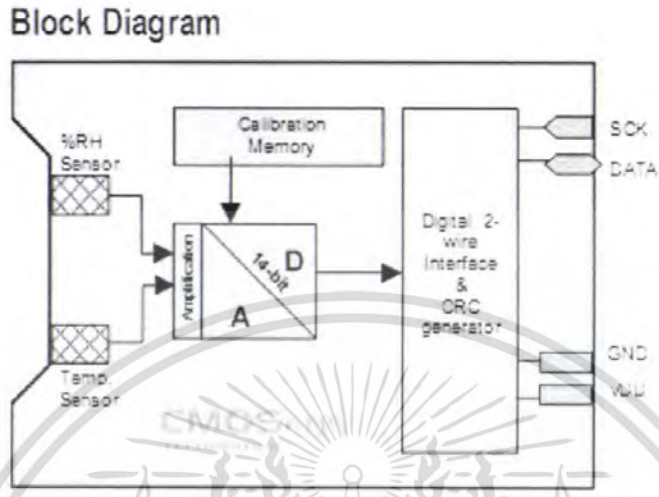
Parameter	Conditions	Min.	Typ.	Max.	Units
Humidity					
Resolution [1]		0.5	0.03	0.03	%RH
		8	12	12[3]	bit
Repeatability			±0.1		%RH
Accuracy [3] Uncertainty	linearized	see figure 1			
Interchangeability		Fully interchangeable			
Nonlinearity	raw data		±3		%RH
	linearized		<<1		%RH
Range		0		100	%RH
Response time	1/e (63%) at 25°C, 1m/s air	6	8	10	s
Hysteresis			±1		%RH
Long term stability	typical		< 0.5		%RH/yr
Temperature					
Resolution [1]		0.04	0.01	0.01	°C
		0.07	0.02	0.02	°F
		12	14	14	bit
Repeatability			±0.1		°C
			±0.2		°F
Accuracy [3]	see figure 1				
Range		-40		123.8	°C
		-40		254.9	°F
Response Time	1/e (63%)	5		30	s

ตารางที่ 2.8 แสดงคุณสมบัติการทำงานของ SHT 15

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial data, bidirectional
3	SCK	Serial clock, input
4	VDD	Supply
	NC	Remaining pins must be left unconnected

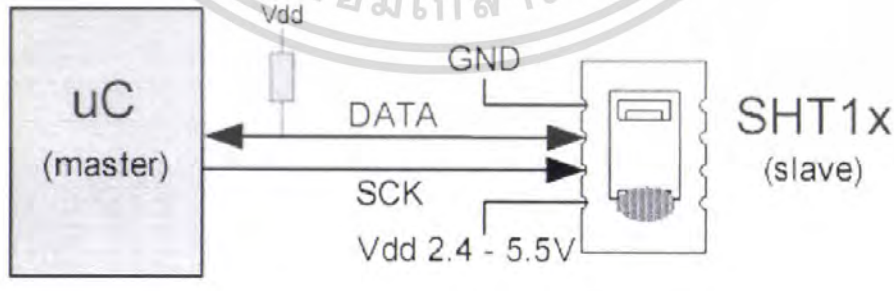
ตารางที่ 2.9 แสดงลักษณะของ SHT15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 แสดง Block Diagram ของ SHT15

2.6.2 การนำไปใช้งานจะต่อวงจรลักษณะดังรูป



รูปที่ 2.17 แสดงการต่อ SHT15 กับไมโครคอนโทรเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การต่อขา Vcc กับ GND ต้องต่อไฟเลี้ยงให้อยู่ระหว่าง 2.4 – 5.5 V แล้วหลังจากที่จ่ายไฟเข้าที่ตัว SHT 15 แล้ว SHT 15 จะใช้เวลาประมาณ 11 ms เพื่อเข้าสู่โหมด Sleep ดังนั้นต้องส่งข้อมูลก่อนที่ IC จะเข้าสู่โหมด Sleep ในการต่อ Vcc กับ GND ควรที่จะต่อ C 100 nF คร่อมระหว่างขา Vcc กับ GND
- การต่อขา DATA และขา SCK จะเป็นการต่อ Serial Interface(Bidirectional 2 - wire) ซึ่งจะเป็นการต่อในลักษณะที่คลายกับ I²C แต่ไม่เหมือน I²C โดยทั่วไป
- การต่อขา SCK จะเป็นการต่อแบบตรงระหว่างไมโครคอนโทรเลอร์กับ SHT 15
- การต่อขา DATA จะมีการต่อ Pull-up เพื่อให้ได้สัญญาณที่มีค่าสูง ซึ่งการต่อ Pull-up จะต่อกันบ่อยๆในการใช้ไมโครคอนโทรเลอร์ และในช่วงที่ทำการส่ง DATA จำเป็นที่จะต้องทำให้ DATA มีความเสถียรในขณะที่ SCK high ซึ่งแสดง Timing Diagram ได้ดังรูป

รูปที่ 2.18 แสดง Timing Diagram ในช่วงการส่งข้อมูลของ SHT 15

2.7 ส่วนแสดงผลทาง LCD Module

รูปที่ 2.19แสดงรูปของจอ LCD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นจอแสดงผลแบบผลึกเหลวซึ่งเป็นสารที่รวมตัวกันอย่างได้สัดส่วนระหว่างของเหลวกับผลึก LCD มีข้อดีหลายประการ

- กินพลังงานน้อย ต้องการกำลังงานน้อย แรงดันต่ำ
- เชื่อถือได้ใช้งานได้ในช่วงอุณหภูมิที่กว้างและมีอายุการใช้งาน
- ง่ายที่จะทำการควบคุมการแสดงผลข้อมูล
- ราคาถูก ใช้งานได้กว้างขวาง

ซึ่งด้วยเหตุผลดังกล่าวจึงทำให้ LCD กลายเป็นส่วนประกอบที่สำคัญของอุปกรณ์ต่างๆ ที่ผลิตออกสู่ท้องตลาดในยุคปัจจุบัน

ในยุคแรกๆ จะเป็นส่วนประกอบของนาฬิกา เครื่องคิดเลข หรือเครื่องมือวัดต่างๆและต่อได้มีการพัฒนาเข้ามามีบทบาทในระบบคอมพิวเตอร์ อุปกรณ์อิเล็กทรอนิกส์ในส่วนของการแสดงผลข้อมูลในปัจจุบัน LCD ได้เข้ามาแทนที่จอภาพแบบ CRT ในที่สุด

ในโครงงานนี้จะใช้ตัวแสดงผลแบบ LCD Module แบบแสดงผลตัวอักษร (Charecters) ขนาด 20 X 4 (ขนาด 20 ตัวอักษร 4 บรรทัด)

2.7.1 การเชื่อมต่อกับโมดูลแอลซีดี (LCD Module)

ในโมดูลแอลซีดีนั้นจะมีส่วนประกอบ 3 ส่วนหลักดังนี้

- ตัวแสดงผล(Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นได้โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงบนจอแอลซีดี
- ตัวควบคุม(Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอก มาควบคุมการทำงานของโมดูลแอลซีดี เช่น ลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์เป็นต้น ตัวควบคุมนี้ใช้ชิบควบคุมโดยเฉพาะชิบ ที่นิยมใช้คือเบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ และ HD61830 ใช้ควบคุม LCD แบบกราฟฟิก
- ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำงานเป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

2.7.1.1 โครงสร้างภายในของตัวควบคุมโมดูล LCD

ในการใช้งานโมดูล LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งในการควบคุมให้ดีเสียก่อน โดยในที่นี้จะทำการแสดงตัวอย่าง โมดูล LCD แบบอักขระ เพราะสามารถเข้าใจง่าย โดยบล็อกไดอะแกรมภายในของชิบควบคุม LCD เบอร์ HD44780H ซึ่งใช้ในโมดูล LCD แบบอักขระประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บัฟเฟอร์อินพุตเอาท์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register:IR) เป็นรีจิสเตอร์ที่ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล(Display Data RAM:DDRAM) เป็นหน่วยความจำแบบแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up table) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM:CGROM)เป็นหน่วยความจำแบบรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7200 บิต โดยจะถูกอ่านด้วยของข้อมูลใน DRAM

แรมเก็บตัวอักษร(Character Generator RAM:CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGROM เอง

แฟลกบิซซี(Busy Flag) เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่าตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนที่จะส่งข้อมูลหรือคำสั่งมายังโมดูล LCD ขนาด 20 ตัวอักษร 4 บรรทัด

สำหรับโมดูล LCD ที่ใช้ในโครงงาน เป็นขนาด 20 ตัวอักษร 4 บรรทัด เนื่องจากเป็นขนาดที่เหมาะสมที่สุดในการใช้การเป็นจอแสดงผลสำหรับการสั่งอาหารซึ่งพิจารณาจากประโยชน์ใช้สอยและความประหยัด ซึ่งประกอบด้วยขาทั้งหมด 14 ขา มีการจัดวางตามรูป

รูปที่ แสดงการจัดขาของโมดูลซีดีแบบอักขระ

Vss(ขา1) : ต่อกราวน์

Vdd(ขา2) : ต่อไฟเลี้ยง +5 v

Vo(ขา4) : เป็นอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS(ขา4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่งแต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R/W(ขา5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลจากโมดูล LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น "1" จะเป็นการอ่านข้อมูล

E(ขา6) : เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเอเบิลโมดูล LCD ให้ทำงาน

D0-D7 : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

(ขา 7-14) : อนึ่งขา RS,R/W และ E จะทำงานร่วมกัน โดยมีความสัมพันธ์แสดงดังตาราง

2.7.1.2 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น "0" แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล(Clear display) มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุด ของจอแสดงผล แล้วเซ็ทบิต I/D ให้เป็น "1"

2. คำสั่ง(Return home) ต้องกำหนดให้บิต 1 ของข้อมูลเป็น "1" เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode set) มีรายละเอียดของรูปแบบข้อมูลคำสั่ง ดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	1	I/D	S

- บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น "1" เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น "0" เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ
- บิต I/D เป็นบิตที่ใช้กำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น "1" แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0" แอดเดรสจะลดลง ดังนั้นข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H-07H (4 ข้อมูลคำสั่งและที่ใช้บ่อยคือ 06H หมายถึงกำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1.3 คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	0	0	D	C	B

- บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น "1" จะเป็นการเปิดจอแสดงผล ถ้าเป็น "0" จะเป็นการปิดจอแสดงผล
- บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น "1" ถ้ากำหนดให้เป็น "0" จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์
- บิต B ใช้ควบคุมการกระพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น "1" เคอร์เซอร์จะกระพริบดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H-0FH ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กระพริบ

2.7.1.4 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผล ขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.1.5 คำสั่งกำหนดฟังก์ชั่นการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
0	0	1	DL	N	F	*	*

- บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น "0" จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าบิตนี้เป็น "1" จะเป็นแบบ 8 บิต
- บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น "0" จะแสดงผล 1 บรรทัด ถ้าเป็น "1" จะแสดงผล 2 บรรทัด ในกรณี ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น "1" จุดที่น่าสังเกตคือ โมดูล LCD แบบ 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่จะต้องกำหนด N ให้เป็น "1" เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่อง คือ 00H และ 40H
- บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น "0" จะเป็นการแสดงผลแบบ 5x7 และถ้าเป็น "1" จะแสดงเป็นแบบ 5x10 จุด ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 4 บรรทัด และเลือกความละเอียดเป็น 5x7 จุด

คำสั่งเลือกแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น "0" บิต 6 เป็น "1" ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทากรกำหนดแอดเดรสด้วยคำสั่งนี้ ก่อนจะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H-3FH

2.7.1.6 คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น "1" และข้อมูลอีก 7 บิตที่เหลือจะเป็นคู่แอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะบิตที่ N ด้วย เป็น "0" แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมีสองช่วงคือ 8CH-87H และ 0C0H-0C7H

2.7.1.7 คำสั่งอ่านบิตซีแฟลตและแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านบิซซีแฟลก(BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น "0" แสดงว่าพร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น "1" แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง เมื่อต้องการอ่านค่าแฟลกต้องกำหนดให้ขา R/W เป็น "1" ด้วย แต่สัญญาณที่ RS ยังต้องเป็น "0" อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูล แอดเดรสของ CGRAM และ DDRAM ด้วย

2.8 คีย์แพด (Keypad)

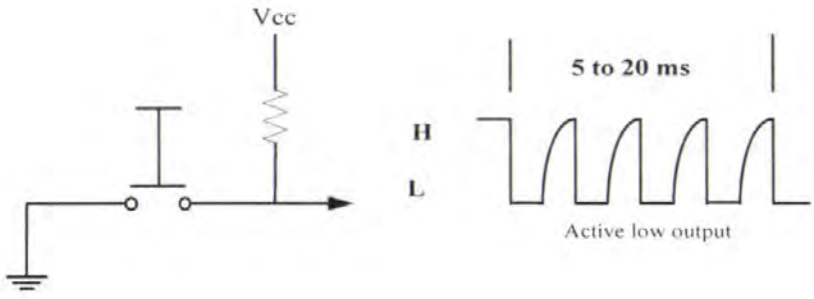
คีย์แพด (Keypad) คืออุปกรณ์อินพุตที่เกิดจากการนำสวิตช์หลายๆ ตัวมาต่อด้วยกันแบบเมทริกซ์เสมือนเป็นคีย์บอร์ดขนาดเล็กอันหนึ่ง เช่น ต้องการคีย์แพดขนาด 4×4 จะใช้สวิตช์ทั้งหมด 16 ตัว โดยกลไกการทำงานของสวิตช์เมื่อกดปุ่มสวิตช์ จะทำให้เส้นลวดทองแดงสัมผัสกันเกิดการครบวงจรขึ้นดังรูปที่ 2.17

(1) Push-botton Switch (2) Toggle Switch

รูปที่ 2.20 การทำงานของสวิตช์

จากโครงสร้างการทำงานของสวิตช์จะมีปัญหาที่เรียกว่า Contact Bounce หรือการกดสัมผัส นั่นคือเมื่อเรากดสวิตช์ เอาต์พุตที่ได้จะเกิดเป็นพัลส์ขึ้นมาในช่วงแรกๆ ที่เป็นเช่นนี้เพราะว่าเวลาเรากดสวิตช์หน้าสัมผัสของสวิตช์ไม่ได้เชื่อมต่อทันทีทันใด แสดงดังรูปที่ 2.21

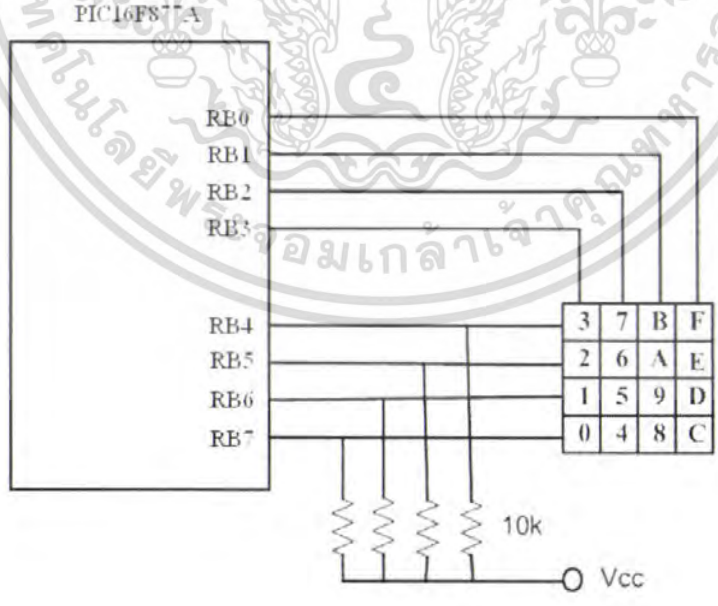
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แสดงการเกิดพัลส์เมื่อทำการกดสวิตช์

เพื่อให้ได้เอาต์พุตที่ถูกต้องจากการกดสวิตช์ ก่อนส่งไปยังไมโครคอนโทรลเลอร์เราต้องหน่วงเวลาไว้ระยะหนึ่งก่อนเพื่อแน่ใจว่าสวิตช์สัมผัสกันคงที่แล้ว กระบวนการนี้เรียกว่า Debouncing ดังนั้นในการใช้งานคีย์แพดจึงประกอบไปด้วย 3 ขั้นตอน

2.8.1 Keypad Scanning สำหรับตรวจสอบว่าสวิตช์ไหนถูกกด เนื่องจากคีย์แพดประกอบด้วยสวิตช์จำนวนหลายตัวมาต่อกันเป็นเมทริกซ์ และใช้เทคนิคการถอดรหัส และการเลือกว่าคีย์ไหนถูกกดให้ดูคีย์ในแนวที่แถวและคอลัมน์ตรงกัน



รูปที่ 2 .22 รูปแสดงการเชื่อมต่อคีย์แพดเข้ากับตัวไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 Keypad Debouncing เพื่อความแน่ใจว่าสวิตช์นั้นถูกกดจริง เมื่อเรากดสวิตช์เอาต์พุตที่ได้จะเกิดเป็นพัลส์ขึ้นมาในช่วง 5 ms แรก และเกิดพัลส์เมื่อไม่กดหรือปล่อยสวิตช์ภายใน 20 ms ดังนั้นจึงให้คิดว่าสวิตช์ถูกกดหลังจากแรงดันเป็น low ประมาณ 10 ms และสวิตช์ถูกปล่อยหลังจากแรงดันเป็น high ประมาณ 10 ms

2.8.3 Table Lookup สำหรับหารหัสแอสกี้ (ASCII Code) ของคีย์ที่กดนั้น เพื่อส่งให้ไมโครคอนโทรลเลอร์ หลังจากคีย์ถูกกดโปรแกรมก็จะหารหัสแอสกี้ (ASCII) ของคีย์ที่เรากดในตาราง รหัสแอสกี้ (ASCII Code Table) จากนั้นจะส่งรหัสที่ได้ไปยังซีพียู

2.9 ตัวทำความชื้นชนิดอัลตราโซนิค (ULTRASONIC HUMIDIFIERS)

รูปที่ 2.23 ตัวทำความชื้นชนิดอัลตร้าโซนิค (ultrasonic humidifiers)

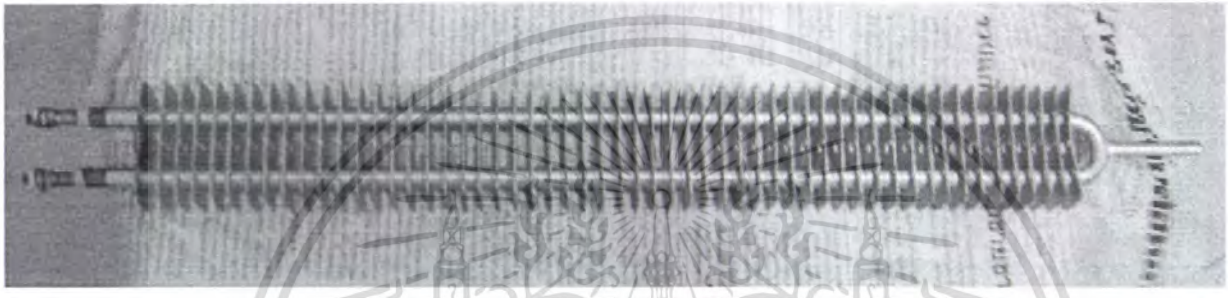
ตัวทำความชื้นถูกใช้ในการปรับระดับความชื้น ให้อยู่ในช่วงที่สามารถควบคุมได้ในระบบการทำงานของโรงงานอุตสาหกรรม เช่น การลดอัตราการส่งผ่านพาหะ ซึ่ง เป็นสาเหตุที่ทำให้เกิดเชื้อแบคทีเรียในโรงพยาบาล , การลดไฟฟ้าสถิตย์ในห้องคอมพิวเตอร์ และ ห้องที่ควบคุมเรื่องความสะอาดของโรงงานอุตสาหกรรม

ตัวทำความชื้นชนิด อัลตร้าโซนิค จะสร้างกลุ่มหมอกไอน้ำที่ปราศจากการเพิ่มขึ้นของอุณหภูมิโดยใช้การสั่นของอุปกรณ์อิเล็กทรอนิกส์ที่ถูกเปลี่ยนให้เป็นการสั่นเชิงกล โดใช้ตัวแผ่นเปียโซ (a piezo disk) จุ่มลงในน้ำ การสั่นเชิงกลจะมีผลโดยตรงที่พื้นผิวน้ำ ซึ่งทำให้เกิดความถี่สูงที่บริเวณผิวน้ำทำให้เกิดกลุ่มหมอกของหยดน้ำบริเวณนั้น ขบวนการนี้ไม่สามารถส่งผ่านความร้อนได้ การลดลงของตัวทำความชื้นใช้พลังงานประมาณ 90-93 % เมื่อเทียบกับระบบที่ทำการต้มน้ำ ตัวทำความชื้นชนิดอัลตร้าโซนิค แสดงเห็นถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

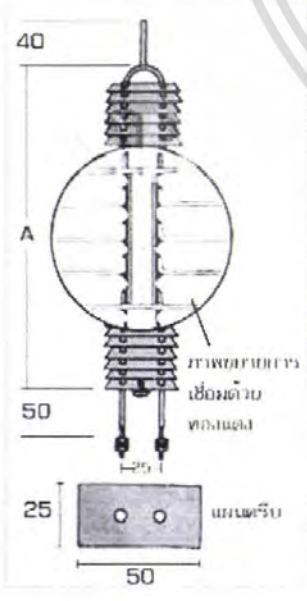
ประสิทธิภาพสูง และการซ่อมบำรุงต่ำกว่าตัวทำความชื้นแบบ electrode canisters , quartz lamps and indirect stream-to-stream กระบวนการนี้จะไม่ทำให้เกิดการกัดกร่อน ซึ่งส่งผลกระทบกับคุณภาพของอากาศภายในอาคาร เมื่อเทียบกับตัวทำความชื้นแบบสร้างไอน้ำโดยตรง (direct stream humidifiers

2.10 ฮีทเตอร์อากาศแบบครีบ Fin heater แบบตัว U (CAL-P)



ฝาเข้าจากยุโรป ประกอบด้วย แท่งฮีทเตอร์ขนาด 8.0 มม. เชื่อมด้วยทองแดงกับแผ่นครีบ 25x50 มม. ระยะห่างระหว่างครีบ 7 มม. พ่นด้วยสีอีพ็อกซี่ มีความแข็งแรง ทนทานเป็นเลิศ ให้ความร้อนฉับไว มีสองรุ่น:

- Series20 <สามารถใช้โดยไม่ต้องมีพัดลมเป่า>
- Series30 <ต้องใช้พัดลมเป่า> ความเร็วลมที่ผ่านฮีทเตอร์ครีบ ไม่ควรต่ำกว่า 3 เมตร / วินาที



Series 20 (2000w/m) 220v			
รุ่น	วัตต์	ความยาว (A) มม.	น้ำหนัก (กก.)
CAL-P 1060*	250	200	0.21
CAL-P 1061	600	300	0.27
CAL-P 1062	800	400	0.40
CAL-P 1063	1000	500	0.62
CAL-P 1064	1250	625	0.77
CAL-P 1065	1500	750	0.95
CAL-P 1066	1750	850	1.10
CAL-P 1067	2000	1000	1.25
Series 30 (3000w/m) 220v			
รุ่น	วัตต์	ความยาว (A) มม.	น้ำหนัก
CAL-P 1071	1000	300	0.27
CAL-P 1072	1500	400	0.40
CAL-P 1073	1750	500	0.62
CAL-P 1074	2000	625	0.77
CAL-P 1075	2500	750	0.95
CAL-P 1076	3000	875	1.10
CAL-P 1077	3500	1000	1.25

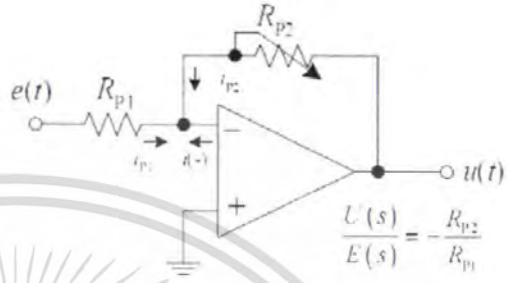
* ไม่มีแท่งแกน 40 มม. ทั้งปลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 ตัวควบคุมสัดส่วน

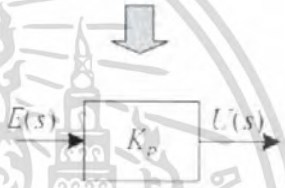
ตัวควบคุมสัดส่วน

- ตัวควบคุมสัดส่วนหรือตัวควบคุม P
- ประกอบด้วยพารามิเตอร์เดียว คือ อัตราการขยาย K_P



- ฟังก์ชันถ่ายโอนของตัวควบคุมสัดส่วน

$$\frac{U(s)}{E(s)} = K_P$$



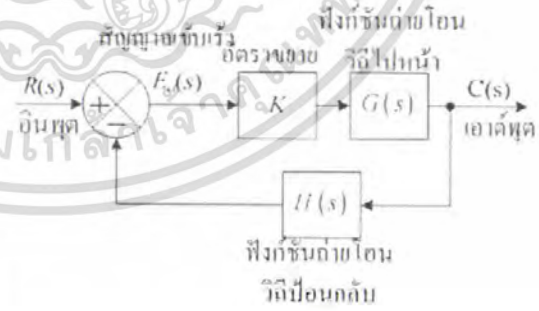
โพลของฟังก์ชันถ่ายโอนวงปิด

- ฟังก์ชันถ่ายโอนวงปิด $G_n(s)$

$$G_n(s) = KG(s)H(s) = KGH(s)$$

- ฟังก์ชันถ่ายโอนวงปิด $T(s)$

$$T(s) = \frac{KG(s)}{1 + KGH(s)}$$

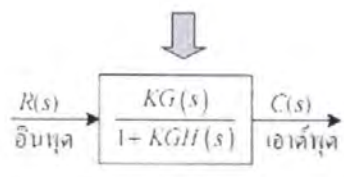


- กำหนด

$$G(s) = \frac{N_G(s)}{D_G(s)} \quad \text{และ} \quad H(s) = \frac{N_H(s)}{D_H(s)}$$

- สมการลักษณะเฉพาะของระบบ

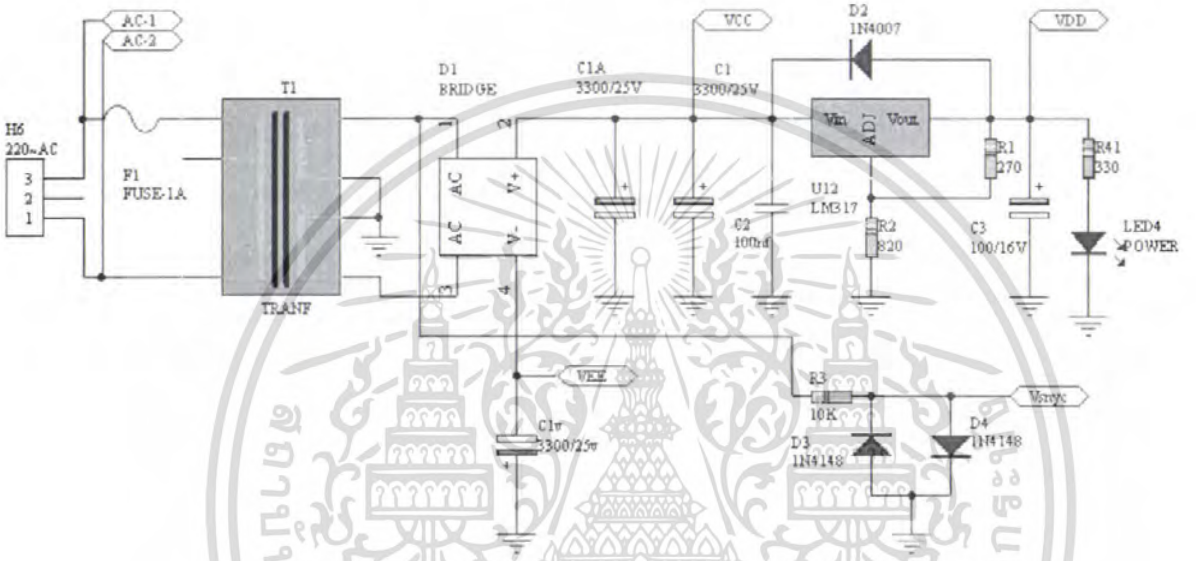
$$F(s) = D_G(s)D_H(s) + KN_G(s)N_H(s)$$
$$= 1 + KGH(s)$$



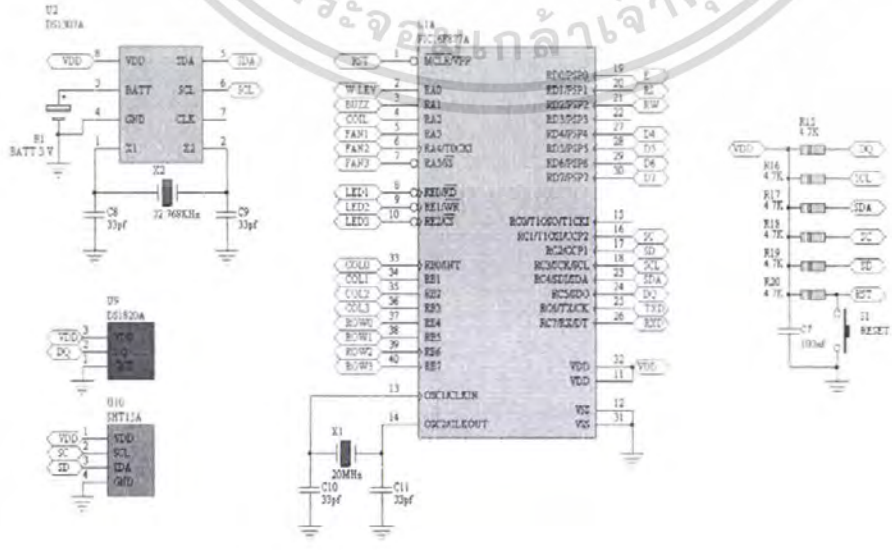
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างและการออกแบบระบบ

3.1.1 > วงจรภาคแหล่งจ่ายไฟและ Vsync control

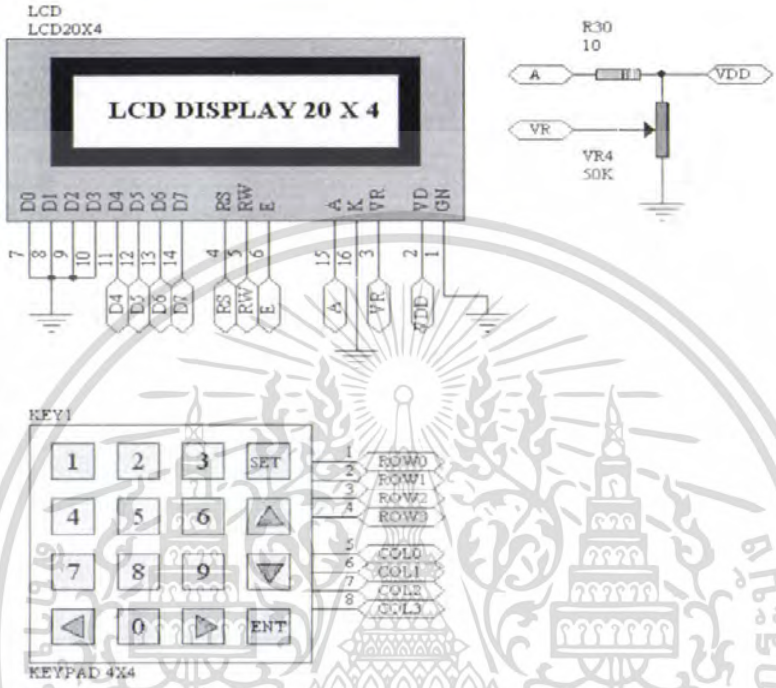


3.1.2 > วงจรส่วนไมโครคอนโทรเลอร์

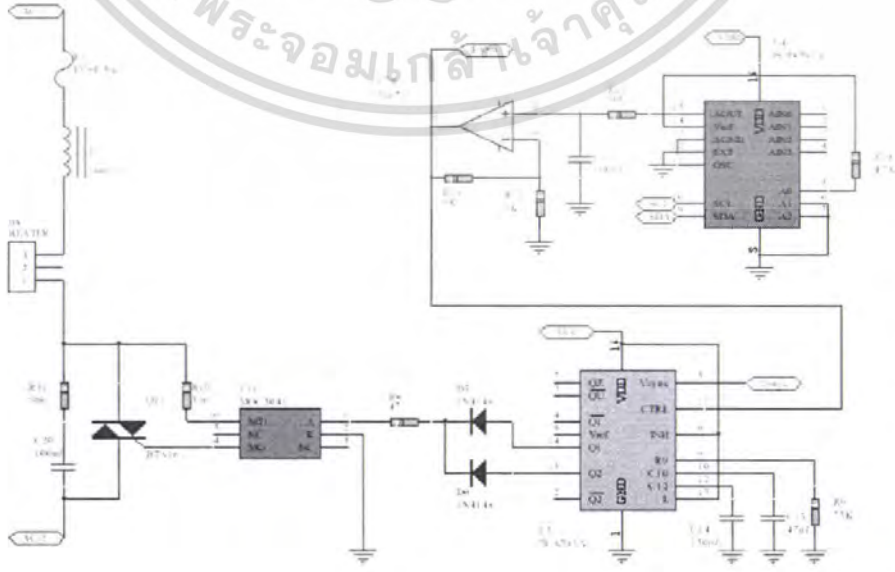


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 > ส่วนของ Keypad and LCD

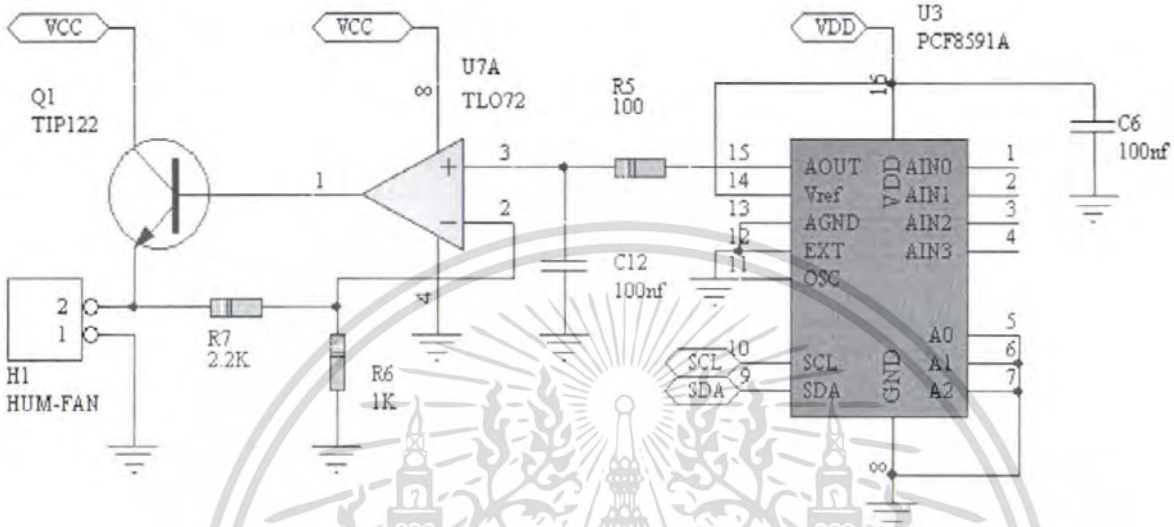


3.1.4 > ส่วนของวงจรเฟสคอนโทล

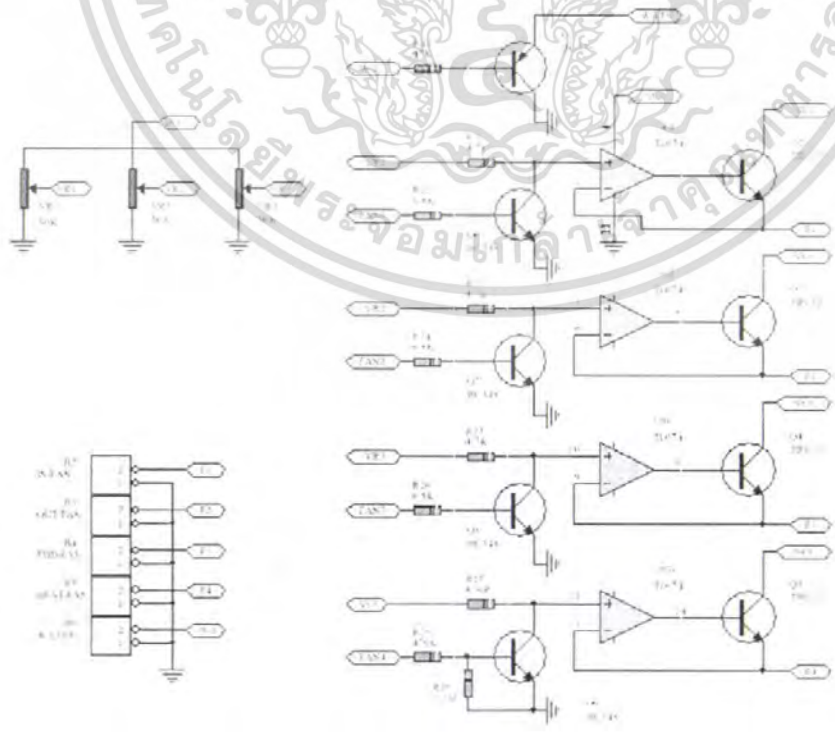


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 > วงจรควบคุมพัดลมความชื้น

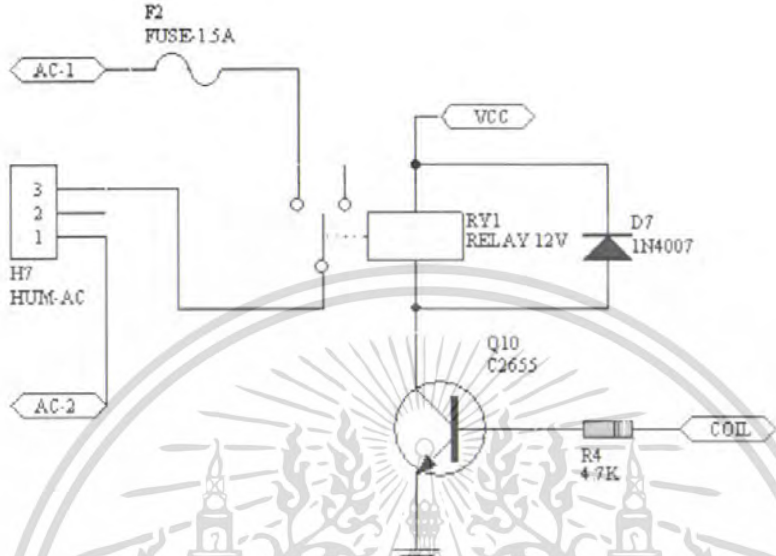


3.1.6 > วงจรควบคุมพัดลม I/O

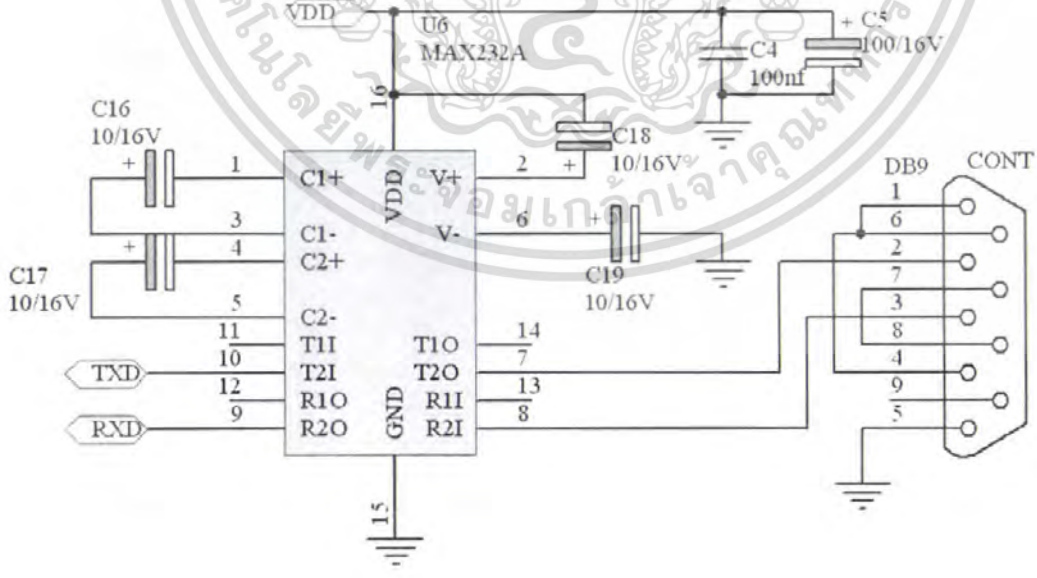


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.7 > วงจรควบคุมตัวทำความชื้น



3.1.8 > วงจรส่งผ่าน RS 232



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

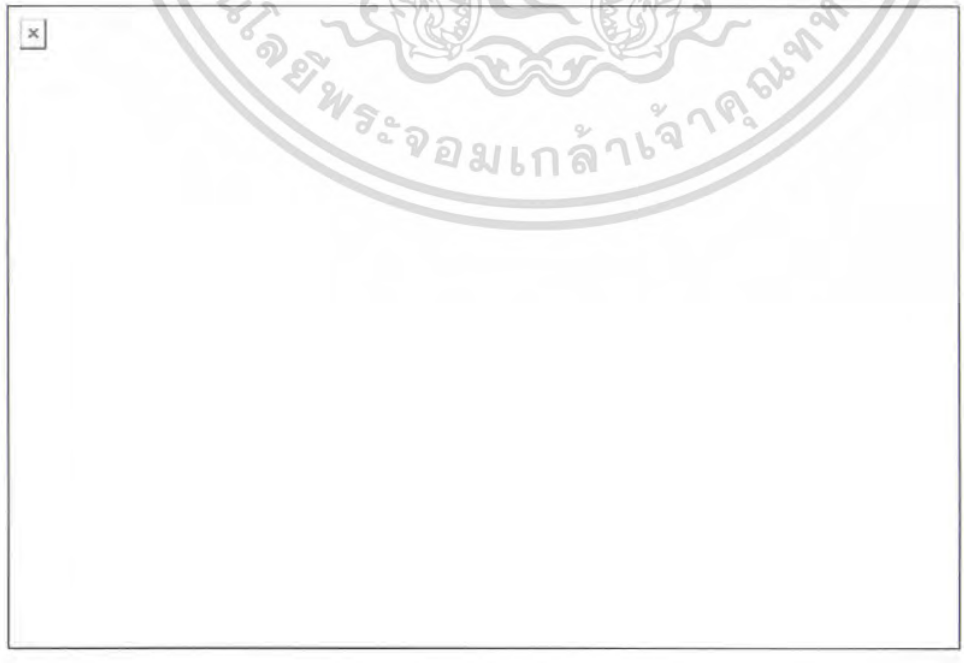
3.2 > รูปห้องจำลองและห้องควบคุมรวม

3.2.1 > ห้อง CONTROL หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 > ห้องจำลองการทดลอง

3.2.3 > ห้องทำความชื้น



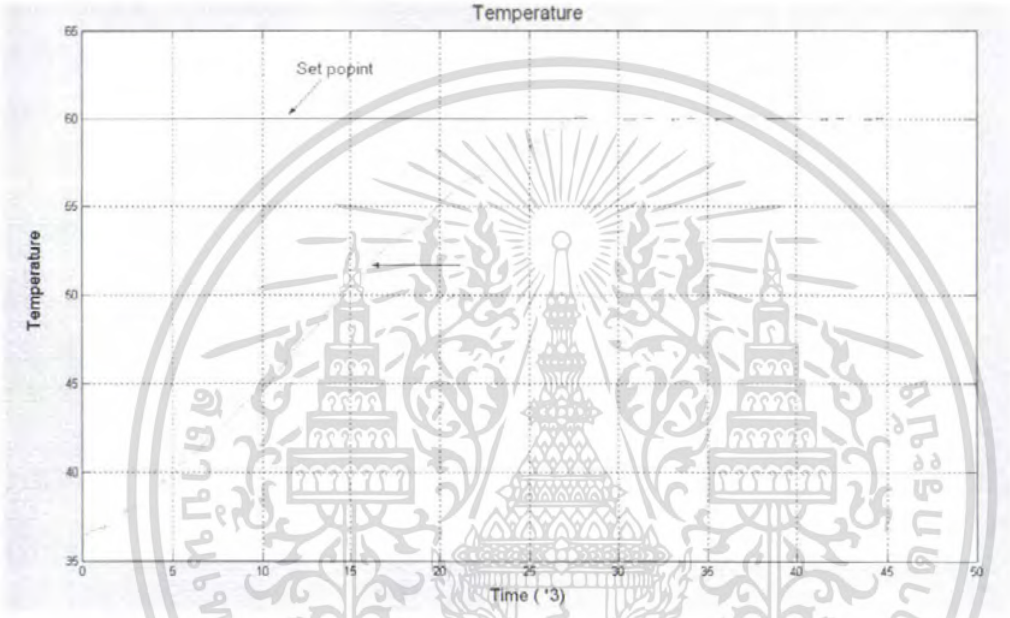
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 > การวางวงจรควบคุมภายในตู้ CONTROL

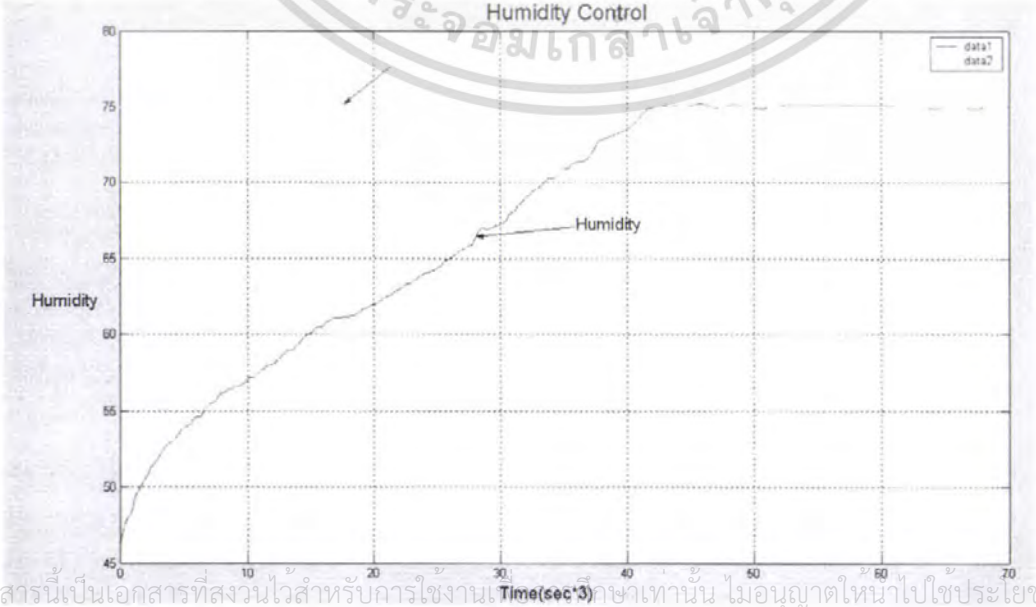
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ขั้นตอนการทดลองและผลการทดลอง

4.1> การทดลอง การควบคุมความชื้นเพียงอย่างเดียว

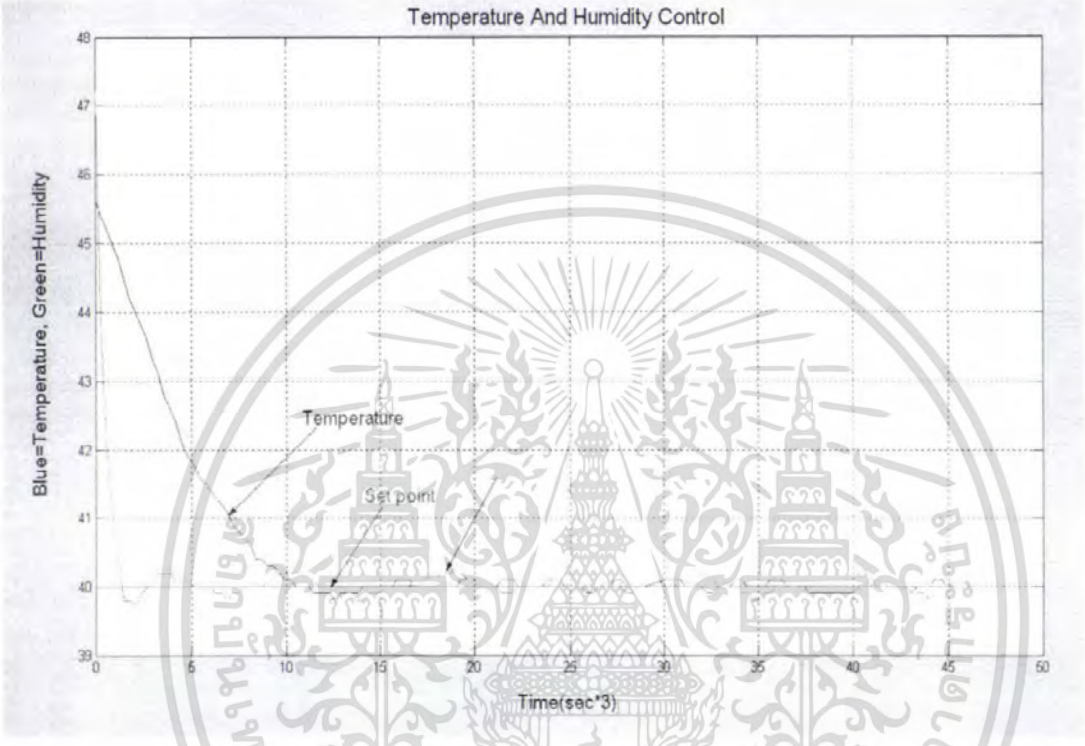


4.2 > การควบคุมอุณหภูมิเพียงอย่างเดียว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 > การควบคุมอุณหภูมิและความชื้นพร้อมกัน



4.1 > ตารางแรงดันพัดลมความชื้น

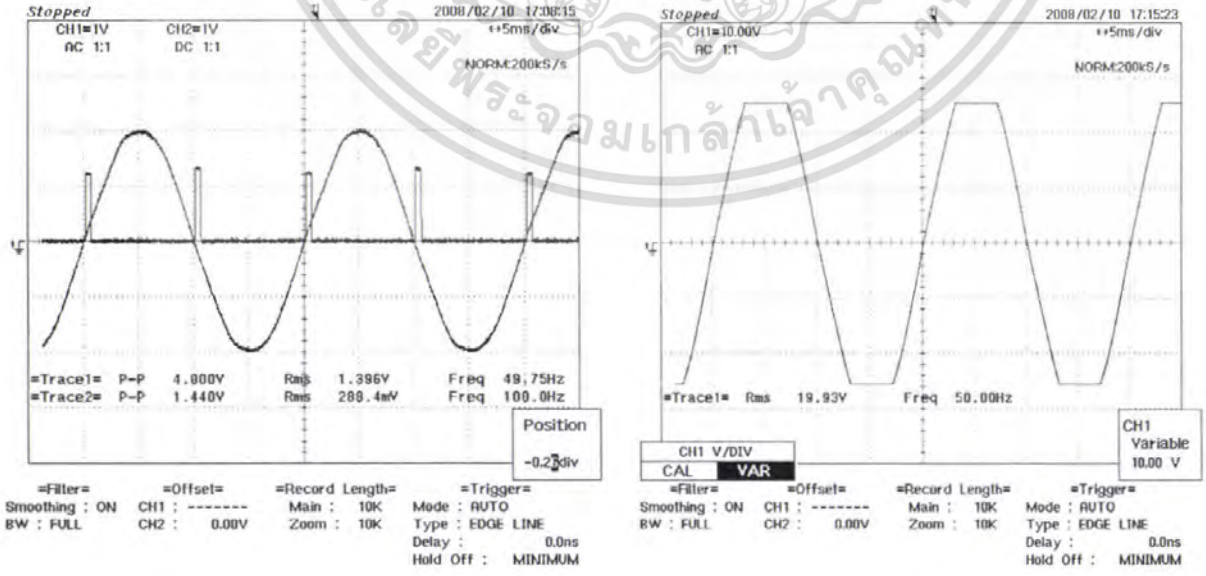
DAC	VDACout	V_fan
0X00 H	0.02 V	0.15 V
0X36 H	1.02 V	2.05 V
0X50 H	1.52 V	3.02 V
0X80 H	2.55 V	5.03 V
0XB3 H	3.51 V	7.08 V
0XE6 H	4.51V	9.05 V
0XFF H	5.04V	10.04 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 > ตารางมุมเฟสและแรงดันของ Heater

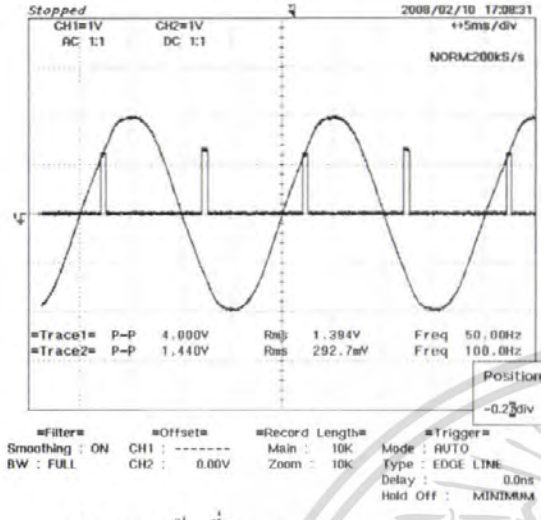
DAC	VDACout	V11	Phase	V-Heater
0X00 H	0.01 V	0.1 V	1.8 °	220 VAC
0X36 H	1.02 V	2.05 V	37 °	195 VAC
0X50 H	1.52 V	3.02 V	55 °	175 VAC
0X80 H	2.53 V	5.03 V	90 °	120 VAC
0XB3 H	3.51 V	7.08 V	127 °	55 VAC
0XE6 H	4.53 V	9.07 V	163 °	10 VAC
0XFF H	5.04 V	10.04 V	180 °	0.5 VAC

4.4 > ภาพกราฟแสดงการเปรียบเทียบสัญญาณระหว่างมุมกระตุ้น Triac กับแรงดันตกคร่อม Heater

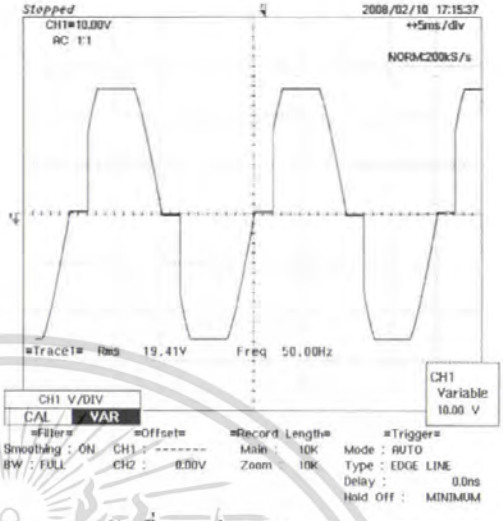


- กระตุ้นที่มุม 1.8 °
- แรงดันที่ตกคร่อม Heater = 220 VAC

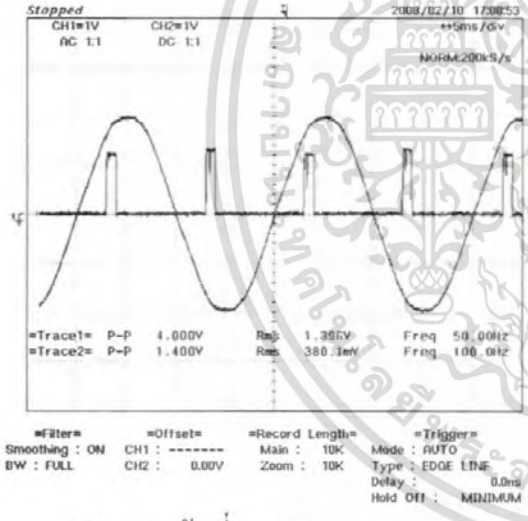
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



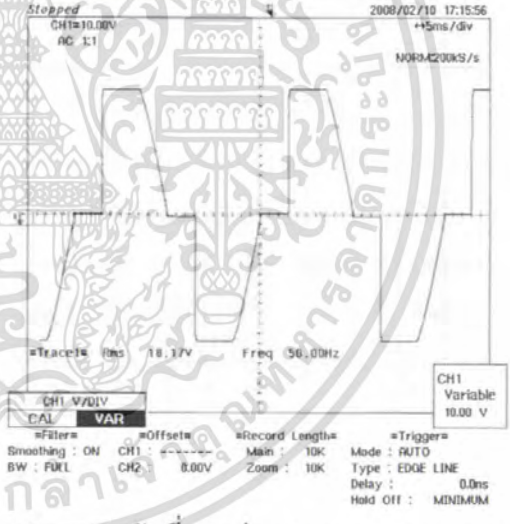
● กระตุ้นที่มุม 37 °



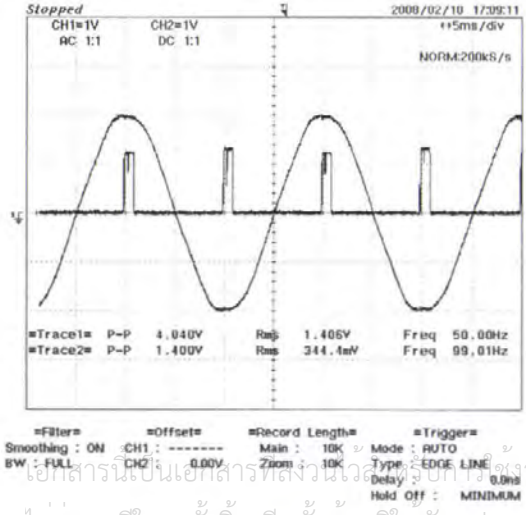
● แรงดันที่ตกคร่อม Heater = 195 VAC

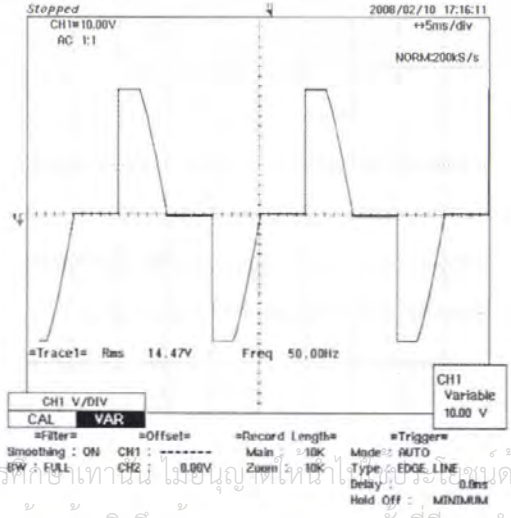


● กระตุ้นที่มุม 55 °



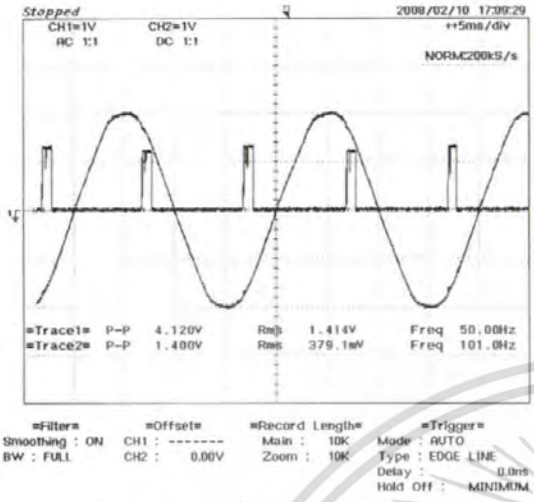
● แรงดันที่ตกคร่อม Heater = 175 VAC



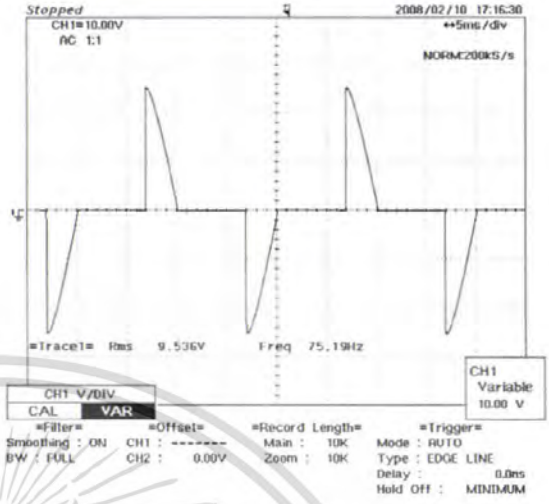


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

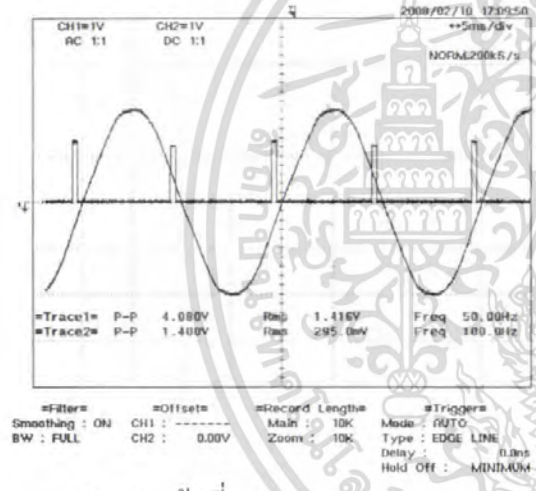
- กระตุ้นที่มุม 90 °



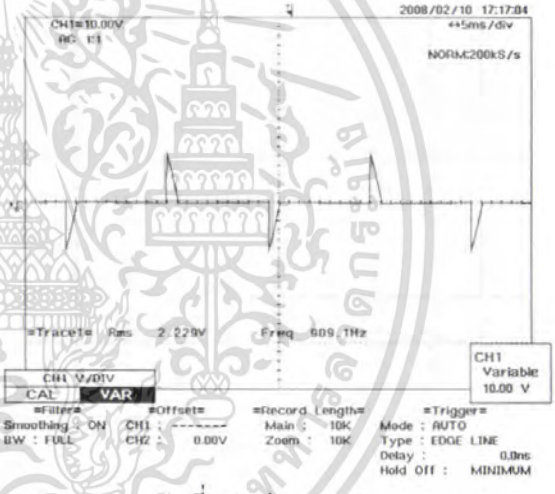
- แรงดันที่ตกคร่อม Heater = 120 VAC



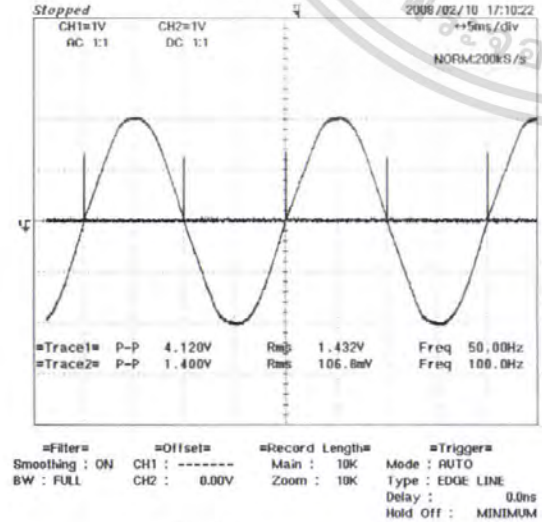
- กระตุ้นที่มุม 127 °



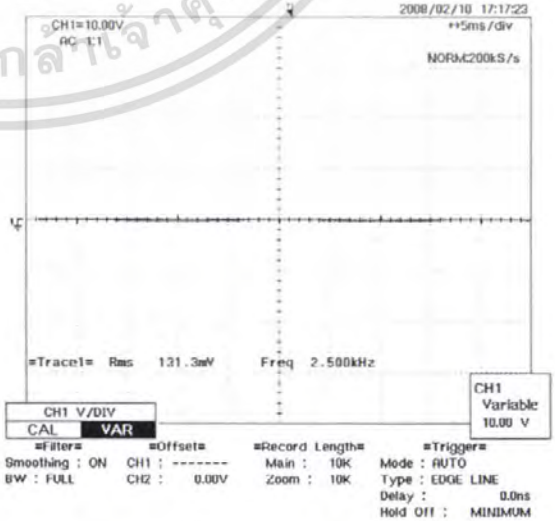
- แรงดันที่ตกคร่อม Heater = 55 VAC



- กระตุ้นที่มุม 163 °



- แรงดันที่ตกคร่อม Heater = 10 VAC

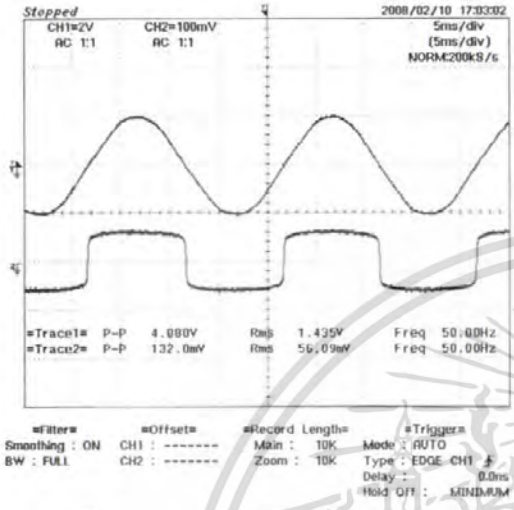


- กระตุ้นที่มุม 180 °

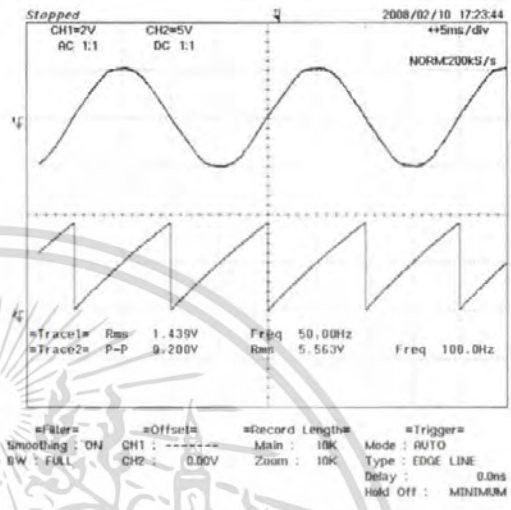
- แรงดันที่ตกคร่อม Heater = 0.5 VAC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปกราฟแสดงรูปสัญญาณของ TCA 785



● สัญญาณV_syncเทียบกับVac 50Hz



● สัญญาณ saw_tooth เทียบกับสัญญาณVac 50Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

5.1 > บทสรุป

จากการออกแบบวงจรที่นำเอาเซนเซอร์แบบวัดอุณหภูมิและความชื้นในตัวเดียวกัน (SHT15) มาใช้ในการทดลองนี้จะเห็นได้ว่ามีความสะดวกและมีความละเอียดมากในการวัดค่าและไม่ต้องยุ่งยากในการแปลงสัญญาณที่ได้จากเซนเซอร์ เพราะเอาท์พุทของ (SHT15) ออกมาเป็นสัญญาณแบบดิจิตอลเลย โดยการแสดงค่าของเอาท์พุทจะแสดงที่จอ LCD ขนาด 4 บรรทัด 20 ตัวอักษร

ในการทดลองนี้ได้นำเอาตัวทำความชื้นแบบอัลตร้าโซนิค มาทำความชื้นให้แก่ระบบ เพราะความชื้นที่ได้จากตัวทำความชื้นชนิดนี้มีลักษณะเป็นละอองไอน้ำขนาดเล็กๆ ขนาดประมาณ 0.4-1.4 ไมครอน ดังนั้นละอองไอน้ำจึงสามารถแพร่กระจายและรวมกับอากาศจนกลายเป็นความชื้นรวมกับอากาศได้เป็นอย่างดี

จากการทดลองที่ได้จากกราฟนั้น จะเห็นว่ากราฟของอุณหภูมินั้นเมื่ออุณหภูมิเพิ่มหรือลดจำเป็นต้องใช้เวลาในการทำงานพอสมควรเพราะอุณหภูมินั้นมีการเปลี่ยนแปลงที่ช้ามากดังนั้นในการลดอุณหภูมิเราจึงใช้ความชื้นเข้าช่วยเพื่อให้อุณหภูมิลดหรือเพิ่มได้เราขึ้น (ต้องการ Temp ลดให้เพิ่มความชื้นเข้าห้อง ต้องการ Temp เพิ่มดูดความชื้นออกจากห้อง

จากการทดลองการเปลี่ยนแปลงความชื้นนั้นจะมีความไวมากกว่าการเปลี่ยนแปลงอุณหภูมิ เพราะความชื้นจะมีการเปลี่ยนแปลงที่เร็ว แต่ในการทดลองเราต้องการให้ค่าความชื้นเข้าถึงจุดที่ต้องการเร็วที่สุดเราจึงใช้อุณหภูมิเข้าช่วย จะทำให้ความชื้นเข้าถึงจุดที่กำหนดได้ไว (ต้องการความชื้นมาก ลดอุณหภูมิลง เมื่อต้องการความชื้นน้อยเพิ่มอุณหภูมิภายในห้อง)

5.2 > ปัญหาที่พบ

จากการทดลองจะเห็นว่าอุณหภูมิกับความชื้นนั้นเมื่อนำมาคุมพร้อมกันจะสามารถคุมได้ในช่วงต่ำเท่านั้น ประมาณช่วง 45 ของทั้งอุณหภูมิและความชื้น ถ้าหากควบคุมในช่วงสูงๆนั้นจะไม่สามารถควบคุมไปในจุดที่ต้องการได้ และตู้จำลองอาจมีลอยรั่วทางด้านพัดลม I/O จึงทำให้ค่าของอุณหภูมิและความชื้นเข้าถึงจุดที่กำหนดได้ช้า

5.3 > แนวทางการพัฒนา

ระบบจำลองการควบคุมที่ทำขึ้นนั้นจะควบคุมแบบ P ซึ่งจากการทดลองนั้นมีความเสถียรภาพเป็นที่น่าพอใจ ซึ่งในอนาคตนั้นอาจพัฒนาให้ควบคุมแบบ PID ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ธีรวัฒน์ ประกอบผล. 2542. "การประยุกต์ใช้ไมโครคอลโทรลเลอร์". สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น). กรุงเทพฯ.
2. ทะนง โชติสรยุทธ์.2537 "เทคนิคการใช้อัตราโซนิคทรานสดิวเซอร์ สำหรับนักทดลอง". ซีเอ็ดยูเคชั่น. กรุงเทพฯ.
3. ประจิน พลังสันติกุล.2521 " การเรียนรู้และใช้งาน CCS คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC" . อินโนเวทีฟ.กรุงเทพฯ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในการควบคุม

```
//#include <16F877A.h>
#include <18F458.h>
#fuses HS,NOWDT,PUT,NOPROTECT,NOLVP
#use delay(clock=20000000)
#include <lcd20x4.c>
#include <sht15.c>
#include <stdlib.h>
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)  // Serial port
#use I2C(master, sda=PIN_C4, scl=PIN_C3)        // I2C Bus module
#define use_portb_kbd
#include <kbd44.c>
#include <ds1307.c>
#include "input.c"
#define PCF8591_ID  0x90  // 10010000, PCF8591:000 Slave Address (bit1-bit3)
#define W_lev      PIN_A0
#define Buz        PIN_A1
#define H_coil     PIN_A2
#define F_in       PIN_A3
#define F_aux      PIN_A4
#define F_out      PIN_A5
#define H_led      PIN_E0
#define Fan        PIN_E1
#define Hu_led     PIN_E2
unsigned int16     a1,b1,c1,d1,e1,a2,b2,c2,d2,e2;
unsigned  char     a,c,d,e,i,j,k,l,x,time,kp1,kp2,kp3,kp4;
signed   int16     s,s1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
float           temp_set,humi_set,restemp,truehumid,rs1,rs2;

//************************** RS-232 Interrup ************************
#INT_TIMER1
void Timer1_ISR()
{
 time++;
 if(time>=60)
 {
  DS1307_ReadDate();
  DS1307_ReadTime();
  //sht_rd(restemp,truehumid);
  printf("\r%3.1fC      %3.1fC      %3.1f%%RH      %3.1f%%RH      %2x/%2x/%2x      %2x:%2x:%2x
\n",temp_set,rs1,
  humi_set,rs2,RTC.date,RTC.month,RTC.year,RTC.hr,RTC.min,RTC.sec);
  time=0;
 }
}
//************************** PCF8591_Write ************************//
void Disp_Temp();
// Description : PCF8591 Write
void PCF8591_DAC(int selec,dat)
{
   i2c_start();          // Start condition
   output_high(PIN_A1);
   i2c_write(PCF8591_ID+selec);    // Device address
   i2c_write(0x44);         // enable DAC (command)
   i2c_write(dat);         // Send data
   i2c_stop();             // Stop condition
}
//************************SW BUZZER*********************************//
void sw_Bz()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  output_high(Buz);
  delay_ms(10);
  output_low(Buz);
 }
//************************Show Process*******************************//
 void Disp_process()
 {
  //DS1307_ReadDate();
  //DS1307_ReadTime();
  lcd_putc('\f');
  lcd_gotoxy(2,1);
  lcd_putc("Temp & Humi Control");
  lcd_gotoxy(1,2);
  lcd_putc("--------------------");
  lcd_gotoxy(1,3);
  printf(lcd_putc,"SetTemp:%3.1f%cC<%3.1f>",temp_set,223,restemp);
  lcd_gotoxy(1,4);
  printf(lcd_putc,"SetHumi:%3.1f%%R<%3.1f>",humi_set,truehumid);
  lcd_gotoxy(2,20);
  lcd_send_byte(0,0x0C);
 }
//************************SHOW DATE**********************************//
void Disp_Show_date()
{
 lcd_putc('\f');
 lcd_gotoxy(2,1);
 lcd_putc("SET : DATE & TIME");
 lcd_gotoxy(2,2);
 lcd_putc("-----------------");
 lcd_gotoxy(2,3);
 lcd_putc("IF SET:press<SET>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 lcd_gotoxy(2,4);
 lcd_putc("NO SET:press<ENT>");
}
//************************SET DATE & TIME*****************************//
void Disp_Set_date()
{
 x=0;
 while(x!=12)
 {
  k=kbd_getc();
  if(k!=0)
  {
   sw_Bz();
   switch(k)
    {
  case 'A' : lcd_putc('\f');
          lcd_gotoxy(2,1);
          lcd_putc("SET : DATE & TIME");
          lcd_gotoxy(2,2);
          lcd_putc("-----------------");
          lcd_gotoxy(2,3);
          lcd_putc("DATE : __/__/__ ");
          lcd_gotoxy(2,4);
          lcd_putc("TIME : __:__:__ ");
          i=9;
          j=9;
          l=3;
          a=0;
          e=0;
          k=0;
          while(a!=12)
          {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(i<=16)
 lcd_gotoxy(i,l);
else
 {
 lcd_gotoxy(j,l);
 e=1;
 }
k=kbd_getc();
if(k!=0)
 {
 sw_Bz();
 switch(i)
 {
 case 9  : lcd_putc(k);
          c=0x0F&k;
          c=c<<4;
          i++;
          break;
 case 10 : lcd_putc(k);
          d=0x0F&k;
          RTC.date=c|d;
          i+=2;
          break;
 case 12 : lcd_putc(k);
          c=0x0F&k;
          c=c<<4;
          i++;
          break;
 case 13 : lcd_putc(k);
          d=0x0F&k;
          RTC.month=c|d;
          i+=2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        break;
 case 15 : lcd_putc(k);
        c=0x0F&k;
        c=c<<4;
        i++;
        break;
 case 16 : lcd_putc(k);
        d=0x0F&k;
        RTC.year=make16(c,d);
        l=4;
        i++;
        break;
 }
if(e==1)
 {
  switch(j)
  {
 case 9  : lcd_putc(k);
        c=0x0F&k;
        c=c<<4;
        j++;
        break;
 case 10 : lcd_putc(k);
        d=0x0F&k;
        RTC.hr=c|d;
        j+=2;
        break;
 case 12 : lcd_putc(k);
        c=0x0F&k;
        c=c<<4;
        j++;
        break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        case 13 : lcd_putc(k);
                  d=0x0F&k;
                  RTC.min=c|d;
                  j+=2;
                  break;
        case 15 : lcd_putc(k);
                  c=0x0F&k;
                  c=c<<4;
                  j++;
                  break;
        case 16 : lcd_putc(k);
                  d=0x0F&k;
                  RTC.sec=c|d;
                  lcd_gotoxy(20,4);
                  j++;
                  break;
          }
         }
        if(j==17)
          a=12;
         }
        }
        break;
case 'D' : x=12;
        lcd_putc('\f');
        break;
 }
 }
}
 DS1307_WriteDate();
 DS1307_WriteTime();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//**********************SHOW TEMP**********************//
void Disp_Temp()
{
 lcd_putc('\f');
 lcd_gotoxy(2,1);
 lcd_putc("SET : Temperature");
 lcd_gotoxy(2,2);
 lcd_putc("-----------------");
 lcd_gotoxy(2,3);
 lcd_putc("Temp_Rang<30-60>C");
 lcd_gotoxy(2,4);
 lcd_putc("IF SET:Press<SET>");
}
//**********************SET TEMP**********************//
void Disp_SetTemp()
{
 x=0;
 while(x!=13)
 {
  k=kbd_getc();
  if(k!=0)
   {
    sw_Bz();
    switch(k)
    {
     case 'A' :  lcd_putc('\f');
                 lcd_gotoxy(2,1);
                 lcd_putc("SET : Temperature");
                 lcd_gotoxy(2,2);
                 lcd_putc("-----------------");
                 lcd_gotoxy(2,3);
                 lcd_putc("Set_Temp:<__.00C>");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lcd_gotoxy(2,4);
lcd_putc("Finish:Press<ENT>");
i=12;
a=0;
e=0;
k=0;
while(a!=13)
 {
  lcd_gotoxy(i,3);
  k=kbd_getc();
  if(k!=0)
   {
    sw_Bz();
    switch(i)
     {
      case 12 : lcd_putc(k);
                c=0x0F&k;
                c=c*10;
                i++;
                break;
      case 13 : lcd_putc(k);
                d=0x0F&k;
                temp_set=c+d+0.1;
                lcd_gotoxy(20,4);
                i++;
                a=13;
                break;
     }
   }
 }
  break;
case 'D' : x=13;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            lcd_putc('\f');
            break;
      }
    }
  }
}

//**********************SHOW HUMID*****************************//
void Disp_Humid()
{
 lcd_putc('\f');
 lcd_gotoxy(2,1);
 lcd_putc("SET : Humidity ");
 lcd_gotoxy(2,2);
 lcd_putc("----------------");
 lcd_gotoxy(2,3);
 lcd_putc("Hum_Rang<30-80>%RH");
 lcd_gotoxy(2,4);
 lcd_putc("IF SET:Press<SET>");
}
//**********************SET HUMID*****************************//
void Disp_SetHumid()
{
 x=0;
 while(x!=14)
 {
  k=kbd_getc();
  if(k!=0)
  {
   sw_Bz();
   switch(k)
   {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
case 'A' : lcd_putc('\f');
        lcd_gotoxy(2,1);
        lcd_putc("SET : Humidity");
        lcd_gotoxy(2,2);
        lcd_putc("----------------");
        lcd_gotoxy(2,3);
        lcd_putc("Set_Hum:<__.0%RH>");
        lcd_gotoxy(2,4);
        lcd_putc("Finish:Press<ENT>");
        i=11;
        a=0;
        c=0;
        k=0;
        while(a!=14)
         {
          lcd_gotoxy(i,3);
          k=kbd_getc();
          if(k!=0)
           {
            sw_Bz();
            switch(i)
             {
              case 11 : lcd_putc(k);
                        c=0x0F&k;
                        c=c*10;
                        i++;
                        break;
              case 12 : lcd_putc(k);
                        d=0x0F&k;
                        humi_set=c+d+0.1;
                        lcd_gotoxy(20,4);
                        i++;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    a=14;
                    break;
                }
               }
              }
            break;
    case 'D' : x=14;
            lcd_putc('\f');
            break;
    }
  }
 }
}
//*******************************SUB KBD*************************//
char Sub_kbd()
{
 unsigned int16 q=0;
 char s,t;
 while(q!=1000)
 {
  s=kbd_getc();
  if(s!=0)
   {
    sw_Bz();
    return s;
    q=1400;
   }
   else
   {
    q++;
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
//*******************************NEW SET*************************//
void New_set()
{
 k=Sub_kbd();
 switch(k)
  {
   case 'A' : i=24;
              break;
   case 'D' : lcd_putc('\f');
              lcd_gotoxy(3,1);
              lcd_putc(" Date & Time \n");
              lcd_putc(" ------------------ ");
              lcd_gotoxy(3,3);
              DS1307_ReadDate();
              delay_ms(10);
              DS1307_ReadTime();
              printf(lcd_putc,"Date:%2x/%2x/%2x ",RTC.date,RTC.month,RTC.year);
              lcd_gotoxy(3,4);
              printf(lcd_putc,"Time:%2x:%2x:%2x ",RTC.hr,RTC.min,RTC.sec);
              delay_ms(1000);
              break;
   }
 }
//*************************HUMID_CONTROL**********************//
void Humid_Control()
{
 i=0;
 while(i!=24)
 {
   New_set();
   sht_rd (restemp, truehumid);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
rs1=restemp;
rs2=truehumid;
d1=humi_set*10;
e1=truehumid*10;
a1=d1-e1;
s=((a1*0.35)*kp2)+70;
if(s>=255)
{
 s=255;
}
Disp_Process();
if((humi_set>truehumid)&&(restemp>45))
{
 output_low(F_in);
 output_low(F_out);
 output_high(H_coil);
 PCF8591_DAC(2,0xFF);
 PCF8591_DAC(0,s);
}
else if(humi_set>truehumid)
{
 output_high(F_in);
 output_high(F_out);
 output_high(H_coil);
 PCF8591_DAC(2,0xFF);
 PCF8591_DAC(0,s);
}
if(truehumid>humi_set)
{
 New_set();
 a2=e1-d1;
 output_high(H_coil);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(a2>=50)
{
 output_low(H_coil);
 PCF8591_DAC(0,0x40);
 PCF8591_DAC(2,0x40);
 //output_low(F_in);
 output_low(F_out);
}
else if((a2>=5)&&(a2<50))
{
 output_low(H_coil);
 PCF8591_DAC(0,0x40);
 PCF8591_DAC(2,0x90);
 output_high(F_in);
 output_high(F_out);
}
else if((a2>=4)&&(a2<=5))
{
 PCF8591_DAC(0,0x00);
 output_low(F_out);
 //output_low(F_in);
 New_set();
 delay_ms(300);
 output_high(F_in);
 output_high(F_out);
 delay_ms(300);
 PCF8591_DAC(2,0xA0);
}
else if(a2<=4)
{
 PCF8591_DAC(2,0xE0);
 PCF8591_DAC(0,0x40);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    output_high(F_out);
    output_high(F_in);
   }
  }
 }
}
//*****************************TEMP_CONTROL************************//
void Temp_Control()
{
 i=0;
 while(i!=24)
 {
  New_set();
  sht_rd (restemp, truehumid);
  rs1=restemp;
  rs2=truehumid;
  d1=temp_set*10;
  e1=restemp*10;
  a1=d1-e1;
  s=255-(((a1*1.5)*kp1)+90);
  if(s<=0)
   {
   s=0;
   }
   c1=s;
   Disp_Process();
  if(temp_set>restemp)
   {
   PCF8591_DAC(2,c1);
   PCF8591_DAC(0,0x00);
   output_low(H_coil);
   output_high(F_out);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 output_high(F_in);
 }
else
 {
 a1=e1-d1;
 if(a1>=30)
 {
  New_set();
  output_high(H_coil);
  PCF8591_DAC(0,0xE0);
  PCF8591_DAC(2,0xFF);
  output_low(F_in);
  output_low(F_out);
 }
 if(a1>=5)
 {
 New_set();
 PCF8591_DAC(0,0xB0);
 output_high(H_coil);
 PCF8591_DAC(2,0xFF);
 output_low(F_in);
 output_low(F_out);
 }
 else if(a1<5)
 {
 output_low(H_coil);
 PCF8591_DAC(2,0xFD);
 PCF8591_DAC(0,0x00);
 output_high(F_out);
 output_high(F_in);
 }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
}
//*****************************TEMP&HUMID_CONTROL*********************//
void TempHumi_Control()
{
 i=0;
 while(i!=24)
 {
  New_set();
  Disp_process();
  sht_rd(restemp,truehumid);
  rs1=restemp;
  rs2=truehumid;
  d1=temp_set*10;
  d2=humi_set*10;
  e1=restemp*10;
  e2=truehumid*10;
  a1=d1-e1;
  a2=d2-e2;
  s=255-(((a1*1.5)*kp3)+75);
  s1=((a2*0.5)*kp4)+65;
  if(s<=0)
  {
   s=0;
   c1=s;
  }
  else
  {
   c1=s;
  }
  if(s1>=255)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 s1=255;
 c2=s1;
 }
 else
 {
 c2=s1;
 }
if((temp_set>restemp)&&(humi_set>truehumid))
{
 output_high(F_out);
 output_high(F_in);
 output_high(H_coil);
 PCF8591_DAC(2,c1);
 PCF8591_DAC(0,c2);
}
else if((temp_set<restemp)&&(humi_set<truehumid))
{
 output_low(F_out);
 output_low(F_in);
 output_high(H_coil);
 PCF8591_DAC(2,0xFF);
 PCF8591_DAC(0,c2);
}
else if((temp_set==restemp)&&(humi_set>truehumid))
{
 output_high(F_out);
 output_high(F_in);
 PCF8591_DAC(0,c2);
 PCF8591_DAC(2,c1);
 output_high(H_coil);
}
else if((temp_set==restemp)&&(humi_set<truehumid))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
 output_low(H_coil);
 PCF8591_DAC(0,0x00);
 PCF8591_DAC(2,c1);
 output_high(F_out);
 output_high(F_in);
}
else if((temp_set>restemp)&&(humi_set==truehumid))
{
 output_low(F_out);
 output_low(F_in);
 PCF8591_DAC(2,c1);
 output_high(H_coil);
 PCF8591_DAC(0,c2);
}
else if((temp_set<restemp)&&(humi_set==truehumid))
{
 PCF8591_DAC(2,0xFF);
 output_low(F_out);
 output_low(F_in);
 output_high(H_coil);
 PCF8591_DAC(0,c2);
}
else if((temp_set==restemp)&&(humi_set==truehumid))
{
 output_high(F_out);
 output_high(F_in);
 output_high(H_coil);
 PCF8591_DAC(0,c2);
 PCF8591_DAC(2,c1);
}
else if((temp_set>restemp)&&(humi_set<truehumid))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  {
   output_low(H_coil);
   PCF8591_DAC(0,0x00);
   PCF8591_DAC(2,c1);
   output_low(F_out);
   output_high(F_in);
  }
  else if((temp_set<restemp)&&(humi_set>truehumid))
  {
   PCF8591_DAC(2,0xFF);
   output_low(F_out);
   output_low(F_in);
   output_high(H_coil);
   PCF8591_DAC(0,c2);
  }
 }
}
//**********************MAIN**********************//
void main()
{
    lcd_init();
    kbd_init();
    sht_init();
    PCF8591_DAC(0,0x00);
    PCF8591_DAC(2,0xFF);
    lcd_send_byte(0,0x0F);
    Disp_Show_date();
    output_low(Buz);
    Disp_Set_date();
    kp1=25;
    kp2=15;
    kp3=30;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    kp4=25;
Nset:  lcd_send_byte(0,0x0F);
    PCF8591_DAC(0,0x00);
    PCF8591_DAC(2,0xFF);
    Disp_Temp();
    Disp_SetTemp();
    Disp_Humid();
    Disp_SetHumid();
    printf("\f\r********** TEMPERTURE & HUMIDITY CONTROL SYSTEM **********\n\n");
    printf("\rS_temp  T_temp  S_humi  T_humid  Date     Time\n");
    enable_interrupts(GLOBAL);
    enable_interrupts(INT_TIMER1);
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_4);
    set_timer1(0);
while(1)
 {
 if((temp_set>=30)&&(humi_set<30))
 {
 humi_set=0x00;
 Temp_control();
 goto Nset;
 }
 if((temp_set<30)&&(humi_set>=30))
 {
 temp_set=0x00;
 Humid_Control();
 goto Nset;
 }
 if((temp_set>=30)&&(humi_set>=30))
 {
 TempHumi_Control();
 goto Nset;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  }
 }
}
```

KBD 44.C

```
//******************************** KEYPAD 4X4*************************//
//                                                                    //
//********************************************************************//
//Keypad connection:
#define row0 PIN_B4
#define row1 PIN_B5
#define row2 PIN_B6
#define row3 PIN_B7
#define col0 PIN_B0
#define col1 PIN_B1
#define col2 PIN_B2
#define col3 PIN_B3

// Keypad layout:
char const KEYS[4][4] =
{{'1','2','3','A'},
 {'4','5','6','B'},
 {'7','8','9','C'},
 {'*','0','#','D'}};
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define KBD_DEBOUNCE_FACTOR 33 // Set this number to apx n/333 where
// n is the number of times you expect
// to call kbd_getc each second

void kbd_init()
{
//set_tris_b(0xF0);
//output_b(0xF0);
port_b_pullups(true);
}

short int ALL_ROWS (void)
{
if(input (row0) & input (row1) & input (row2) & input (row3))
  return (0);
else
  return (1);
}
char kbd_getc()
{
static byte kbd_call_count;
static short int kbd_down;
static char last_key;
static byte col;

byte kchar;
byte row;

kchar='\0';

if(++kbd_call_count>KBD_DEBOUNCE_FACTOR)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
switch (col)
 {
  case 0:
   output_low(col0);
   output_high(col1);
   output_high(col2);
   output_high(col3);
   break;

  case 1:
   output_high(col0);
   output_low(col1);
   output_high(col2);
   output_high(col3);
   break;

  case 2:
   output_high(col0);
   output_high(col1);
   output_low(col2);
   output_high(col3);
   break;

  case 3:
   output_high(col0);
   output_high(col1);
   output_high(col2);
   output_low(col3);
   break;
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(kbd_down)
 {
  if(!ALL_ROWS())
   {
    kbd_down=false;
    kchar=last_key;
    last_key='\0';
   }
 }
else
 {
  if(ALL_ROWS())
   {
    if(!input (row0))
      row=0;
    else if(!input (row1))
      row=1;
    else if(!input (row2))
      row=2;
    else if(!input (row3))
      row=3;

    last_key =KEYS[row][col];
    kbd_down = true;
   }
  else
   {
    ++col;
    if(col==4)
      col=0;
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  kbd_call_count=0;
 }
return(kchar);
}
```

LCD 20X4.C

```
////////////////////////////////////////////////////////////////////////////
////                     LCDD.C                                         ////
////             Driver for common LCD modules                          ////
////                                                                    ////
//// lcd_init()   Must be called before any other function.             ////
////                                                                    ////
//// lcd_putc(c)  Will display c on the next position of the LCD.       ////
////                 The following have special meaning:                ////
////                  \f  Clear display                                 ////
////                  \n  Go to start of second line                    ////
////                  \b  Move back one position                        ////
////                                                                    ////
//// lcd_gotoxy(x,y) Set write position on LCD (upper left is 1,1)      ////
////                                                                    ////
//// lcd_getc(x,y)   Returns character at position x,y on LCD           ////
////                                                                    ////
////////////////////////////////////////////////////////////////////////////
////      (C) Copyright 1996,1997 Custom Computer Services              ////
//// This source code may only be used by licensed users of the CCS C   ////
//// compiler.  This source code may only be distributed to other       ////
//// licensed users of the CCS C compiler.  No other use, reproduction  ////
//// or distribution is permitted without written permission.           ////
//// Derivative programs created using this software in object code     ////
//// form are not restricted in any way.                                ////
////////////////////////////////////////////////////////////////////////////
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// As defined in the following structure the pin connection is as follows:
//     D0  enable
//     D1  rs
//     D2  rw
//     D4  D4
//     D5  D5
//     D6  D6
//     D7  D7
//
//   LCD pins D0-D3 are not used and PIC D3 is not used.

// Un-comment the following define to use port B
// #define use_portb_lcd TRUE


struct lcd_pin_map {                // This structure is overlayed
        boolean enable;             // on to an I/O port to gain
        boolean rs;                 // access to the LCD pins.
        boolean rw;                 // The bits are allocated from
        boolean unused;             // low order up.  ENABLE will
        int     data : 4;           // be pin B0.
      } lcd;


#if defined(__PCH__)
#if defined use_portb_lcd
  #byte lcd = 0xF81                 // This puts the entire structure
#else
  #byte lcd = 0xF83                 // This puts the entire structure
#endif
#else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#if defined use_portb_lcd
  #byte lcd = 6              // on to port B (at address 6)
#else
  #byte lcd = 8              // on to port D (at address 8)
#endif
#endif

#if defined use_portb_lcd
  #define set_tris_lcd(x) set_tris_b(x)
#else
  #define set_tris_lcd(x) set_tris_d(x)
#endif



#define lcd_type 2          // 0=5x7, 1=5x10, 2=2 lines
#define lcd_line_two 0x40   // LCD RAM address for the second line


byte CONST LCD_INIT_STRING[4] = {0x20 | (lcd_type << 2), 0xc, 1, 6};
                    // These bytes need to be sent to the LCD
                    // to start it up.


                    // The following are used for setting
                    // the I/O port direction register.

STRUCT lcd_pin_map const LCD_WRITE = {0,0,0,0,0}; // For write mode all pins are out
STRUCT lcd_pin_map const LCD_READ = {0,0,0,0,15}; // For read mode data pins are in
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
byte lcd_read_byte() {
    byte low,high;
    set_tris_lcd(LCD_READ);
    lcd.rw = 1;
    delay_cycles(1);
    lcd.enable = 1;
    delay_cycles(1);
    high = lcd.data;
    lcd.enable = 0;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(1);
    low = lcd.data;
    lcd.enable = 0;
    set_tris_lcd(LCD_WRITE);
    return( (high<<4) | low);
}


void lcd_send_nibble( byte n ) {
    lcd.data = n;
    delay_cycles(1);
    lcd.enable = 1;
    delay_us(2);
    lcd.enable = 0;
}


void lcd_send_byte( byte address, byte n ) {

    lcd.rs = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    while ( bit_test(lcd_read_byte(),7) ) ;
    lcd.rs = address;
    delay_cycles(1);
    lcd.rw = 0;
    delay_cycles(1);
    lcd.enable = 0;
    lcd_send_nibble(n >> 4);
    lcd_send_nibble(n & 0xf);
}

void lcd_init() {
  byte i;
  set_tris_lcd(LCD_WRITE);
  lcd.rs = 0;
  lcd.rw = 0;
  lcd.enable = 0;
  delay_ms(15);
  for(i=1;i<=3;++i) {
    lcd_send_nibble(3);
    delay_ms(5);
  }
  lcd_send_nibble(2);
  for(i=0;i<=3;++i)
    lcd_send_byte(0,LCD_INIT_STRING[i]);
}

void lcd_gotoxy( byte x, byte y) {
  byte address;
  switch (y)
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    case 1 : address=0x00 ; break;
    case 2 : address=0x40 ; break;
    case 3 : address=0x14 ; break;
    case 4 : address=0x54 ; break;
    }
   address+=x-1;
   lcd_send_byte(0,0x80|address);
}

void lcd_putc( char c) {
   switch (c) {
     case '\f'   : lcd_send_byte(0,1);
                   delay_ms(2);
                                 break;
     case '\n'   : lcd_gotoxy(1,2);      break;
     case '\b'   : lcd_send_byte(0,0x10);  break;
     default     : lcd_send_byte(1,c);     break;
   }
}

char lcd_getc( byte x, byte y) {
   char value;

    lcd_gotoxy(x,y);
    lcd.rs=1;
    value = lcd_read_byte();
    lcd.rs=0;
    return(value);
}
```

SHT15.C

```
////////////////////////////////////////////////////////////////////////////
```

// เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Driver file for SHT15 Temperature & Humidity Sensor          //
//                                                              //
// ***** To initialise SHT15 sensor upon power up *****         //
//                                                              //
// Function : sht_init()                                        //
// Return   : none                                              //
//                                                              //
//                                                              //
// ***** To measure and caluculate SHT15 temp & real RH *****  //
//                                                              //
// Function : sht_rd (temp, truehumid)                          //
// Return   : temperature & true humidity in float values       //
//                                                              //
//////////////////////////////////////////////////////////////////

#define sht_data_pin   PIN_C2
#define sht_clk_pin    PIN_C1


//***** Function to alert SHT15 *****

void comstart (void)
{
 output_float(sht_data_pin);  //data high
 output_bit(sht_clk_pin, 0);  //clk low
 delay_us(1);
 output_bit(sht_clk_pin, 1);  //clk high
 delay_us(1);
 output_bit(sht_data_pin, 0); //data low
 delay_us(1);
 output_bit(sht_clk_pin, 0);  //clk low
 delay_us(2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
output_bit(sht_clk_pin, 1);  //clk high
delay_us(1);
output_float(sht_data_pin);  //data high
delay_us(1);
output_bit(sht_clk_pin, 0);  //clk low
}


//***** Function to write data to SHT15 *****

int1 comwrite (int8 iobyte)
{
int8 i, mask = 0x80;
int1 ack;

//Shift out command
delay_us(4);
for(i=0; i<8; i++)
 {
  output_bit(sht_clk_pin, 0);                  //clk low
  if((iobyte & mask) > 0) output_float(sht_data_pin);  //data high if MSB high
  else output_bit(sht_data_pin, 0);            //data low if MSB low
  delay_us(1);
  output_bit(sht_clk_pin, 1);                  //clk high
  delay_us(1);
  mask = mask >> 1;                            //shift to next bit
 }

//Shift in ack
output_bit(sht_clk_pin, 0);  //clk low
delay_us(1);
ack = input(sht_data_pin);   //get ack bit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 output_bit(sht_clk_pin, 1);  //clk high
 delay_us(1);
 output_bit(sht_clk_pin, 0);  //clk low
 return(ack);
}


//***** Function to read data from SHT15 *****

int16 comread (void)
{
 int8 i;
 int16 iobyte = 0;
 const int16 mask0 = 0x0000;
 const int16 mask1 = 0x0001;

 //shift in MSB data
 for(i=0; i<8; i++)
  {
   iobyte = iobyte << 1;
   output_bit(sht_clk_pin, 1);          //clk high
   delay_us(1);
   if (input(sht_data_pin)) iobyte |= mask1;  //shift in data bit
   else iobyte |= mask0;
   output_bit(sht_clk_pin, 0);          //clk low
   delay_us(1);
  }

//send ack 0 bit
output_bit(sht_data_pin, 0); //data low
delay_us(1);
output_bit(sht_clk_pin, 1); //clk high
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
delay_us(2);
output_bit(sht_clk_pin, 0);  //clk low
delay_us(1);
output_float(sht_data_pin);  //data high

//shift in LSB data
for(i=0; i<8; i++)
 {
  iobyte = iobyte << 1;
  output_bit(sht_clk_pin, 1);              //clk high
  delay_us(1);
  if (input(sht_data_pin)) iobyte |= mask1;  //shift in data bit
  else iobyte |= mask0;
  output_bit(sht_clk_pin, 0);              //clk low
  delay_us(1);
 }

//send ack 1 bit
output_float(sht_data_pin);  //data high
delay_us(1);
output_bit(sht_clk_pin, 1);  //clk high
delay_us(2);
output_bit(sht_clk_pin, 0);  //clk low

return(iobyte);
}


//***** Function to wait for SHT15 reading *****

void comwait (void)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int16 sht_delay;

output_float(sht_data_pin);                //data high
output_bit(sht_clk_pin, 0);                //clk low
delay_us(1);
for(sht_delay=0; sht_delay<30000; sht_delay++)  // wait for max 300ms
 {
  if (!input(sht_data_pin)) break;          //if sht_data_pin low, SHT75 ready
  delay_us(10);
 }
}


//***** Function to reset SHT15 communication *****

void comreset (void)
{
 int8 i;

 output_float(sht_data_pin);    //data high
 output_bit(sht_clk_pin, 0);    //clk low
 delay_us(2);
 for(i=0; i<9; i++)
  {
   output_bit(sht_clk_pin, 1);  //toggle clk 9 times
   delay_us(2);
   output_bit(sht_clk_pin, 0);
   delay_us(2);
  }
 comstart();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//***** Function to soft reset SHT15 *****

void sht_soft_reset (void)
{
 comreset();          //SHT75 communication reset
 comwrite(0x1e);      //send SHT15 reset command
 delay_ms(15);        //pause 15 ms
}


//***** Function to measure SHT15 temperature *****

int16 measuretemp (void)
{
 int1 ack;
 int16 iobyte;

 comstart();          //alert SHT15
 ack = comwrite(0x03);  //send measure temp command and read ack status
 if(ack == 1) return;
 comwait();           //wait for SHT15 measurement to complete
 iobyte = comread();    //read SHT15 temp data
 return(iobyte);
}


//***** Function to measure SHT15 RH *****

int16 measurehumid (void)
{
 int1 ack;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int16 iobyte;

comstart();          //alert SHT15
ack = comwrite(0x05);  //send measure RH command and read ack status
if(ack == 1) return;
comwait();           //wait for SHT15 measurement to complete
iobyte = comread();    //read SHT15 temp data
return(iobyte);
}

//***** Function to calculate SHT15 temp & RH *****

void calculate_data (int16 temp, int16 humid, float & tc, float & rhlin, float & rhtrue)
{
float truehumid1, rh;

//calculate temperature reading
tc = ((float) temp * 0.01) - 40.0;

//calculate Real RH reading
rh = (float) humid;

rhlin = (rh * 0.0405) - (rh * rh * 0.0000028) - 4.0;

//calculate True RH reading
rhtrue = ((tc - 25.0) * (0.01 + (0.00008 * rh))) + rhlin;
}

//***** Function to measure & calculate SHT15 temp & RH *****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void sht_rd (float & temp, float & truehumid)
{
 int16 restemp, reshumid;
 float realhumid;
 restemp = 0; truehumid = 0;

 restemp = measuretemp();    //measure temp
 reshumid = measurehumid();  //measure RH
calculate_data (restemp, reshumid, temp, realhumid, truehumid);  //calculate temp & RH
}
//***** Function to initialise SHT15 on power-up *****

void sht_init (void)
{
 comreset();    //reset SHT15
 delay_ms(20);  //delay for power-up
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้