สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องควบคุมอุณหภูมิเครื่องปรับอากาศและวัดความชื้นสัมพัทธ์

TEMPERATURE REMOTE CONTROL FOR AIR CONDITIONER AND MEASURING HUMIDITY

โดย

นายวิรุฬ กู้เกียรติโอพาร

นายสมประสงค์ เอี่ยมผ่องใส

นายอรรถพร ทวีธนะลาภ

เลขหมู่.....
เลขทะเบียน 73166
วัน,เดือน,ปี 6 ก.ค. 2550

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

.b 11486202
.i.....

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุณหภูมิเครื่องปรับอากาศและวัดความชื้นสัมพัทธ์

TEMPERATURE REMOTE CONTROL
FOR AIR CONDITIONER AND MEASURING HUMIDITY

โดย

นายวิรุฬ	กู้เกียรติโอฬาร	เลขประจำตัว 45010717
นายสมประสงค์	เอี่ยมผ่องใส	เลขประจำตัว 45010800
นายอรรถพร	ทวีธนะลาภ	เลขประจำตัว 45010939

อาจารย์ที่ปรึกษา

ร.ศ. พิชัย คูศิริวานิชกร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2548

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมอุณหภูมิเครื่องปรับอากาศและวัดความชื้นสัมพัทธ์

TEMPERATURE REMOTE CONTROL FOR AIR CONDITIONER AND MEASURING HUMIDITY

ผู้จัดทำ

1. นายวิรุฬ	กู้เกียรติโอฬาร	เลขประจำตัว 45010717
2. นายสมประสงค์	เอี่ยมผ่องใส	เลขประจำตัว 45010800
3. นายอรรถพร	ทวีธนะลาภ	เลขประจำตัว 45010939

ผศ. พิจัย คูศิริวานิชกร อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุณหภูมิเครื่องปรับอากาศและวัดความชื้นสัมพัทธ์

นายวิรุฬ กู้เกียรติโอพาร รหัส 45010717
นายสมประสงค์ เอี่ยมผ่องใส รหัส 45010800
นายอรรถพร ทวีธนะลาภ รหัส 45010939
ร.ศ.พิชัย คูศิริวานิชกร อาจารย์ที่ปรึกษา
ปีการศึกษา 2548

บทคัดย่อ

เครื่องควบคุมเครื่องปรับอากาศโดยใช้ความสัมพันธ์ระหว่างอุณหภูมิกับความชื้นสัมพัทธ์ที่เหมาะสม โดยใช้เซ็นเซอร์ SHT15 ที่สามารถตรวจจับอุณหภูมิกับความชื้นสัมพัทธ์ได้ในตัวเดียวกัน ซึ่งมีการเชื่อมต่อกับไมโครคอนโทรเลอร์ PIC 16F877 แบบ 2-wire และในตัวไมโครคอนโทรเลอร์จะทำหน้าที่วิเคราะห์ความสัมพันธ์ระหว่างอุณหภูมิกับความชื้นสัมพัทธ์ เพื่อส่งสัญญาณอินฟราเรดไปควบคุมเครื่องปรับอากาศให้มีอุณหภูมิที่เหมาะสมเพื่อเป็นการประหยัดพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEMPERATURE REMOTE CONTROL

FOR AIR CONDITIONER AND MEASURING HUMIDITY

Mr.Virun Kukiatoran ID 45010717

Mr.Somprasong Eiampongsai ID 45010800

Mr.Auttaporn Tawcetanalarp ID 45010939

Assist. Prof. Pichai Kusiriwanichakorn Advisor

Education Year 2005

Abstract

This project invents the remote control for air conditioner by relative appropriately between humidity and temperature. It can detect the degree of temperature and humidity by a single SHT15 sensor. And, it connect to microcontroller PIC 16F877, 2-wire interface type which analyze the relationship between temperature and humidity. Later, air conditioner will be controlled by Infrared signal for saving energy.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ ร.ศ.พิชัย คูศิริวานิชกร ที่ท่านเมตตารับเป็นอาจารย์ที่ปรึกษากลุ่มโครงงานของพวกข้าพเจ้า รวมทั้งยังได้มอบแนวความคิดเริ่มต้นของโครงงานนี้ และยังยอมสละเวลาอันมีค่าในการให้คำปรึกษาประกอบการทำโครงงานจนประสบผลสำเร็จลุล่วงไปได้ด้วยดี หากมีสิ่งใดที่ทำให้อาจารย์ขุ่นข้องหมองใจ ก็ต้องกราบขออภัยอาจารย์เป็นอย่างสูง

ขอขอบพระคุณ ท่านอาจารย์ในภาควิชาอิเล็กทรอนิกส์ทุก ๆ ท่าน ที่ได้มีส่วนสอนสั่งให้พวกข้าพเจ้ามีความรู้ สามารถนำมาใช้ให้เกิดประโยชน์ รวมทั้งการทำโครงงานชิ้นนี้ด้วย

ขอขอบพระคุณ ผู้เขียนหนังสือทุก ๆ เล่มที่พวกข้าพเจ้าได้อ่าน

ขอขอบพระคุณ เว็บไซต์ทุก ๆ เว็บไซต์ และผู้สรรสร้างเครือข่ายอินเตอร์เน็ตขึ้นมา ทำให้สามารถสืบค้นข้อมูลได้อย่างสะดวก รวดเร็ว

และสุดท้าย ขอขอบพระคุณ คุณพ่อคุณแม่ทุกท่านของพวกข้าพเจ้า ที่ทำให้พวกข้าพเจ้ามีวันนี้

นายวิรุฬ กู้เกียรติโอพาร

นายสมประสงค์ เอี่ยมผ่องใส

นายอรรถพร ทวีธนะลาภ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1	**บทนำ**	**1**
	1.1 ที่มา	**1**
	1.2 แนวความคิด	**1**
	1.3 หลักการทำงาน	**1**
บทที่ 2	**การวัดอุณหภูมิและความชื้นด้วย SHT 15**	**3**
	2.1 คุณสมบัติของเซ็นเซอร์ SHT 15	**3**
	2.1.1 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ SHT 15	**3**
	2.1.1.1 Serial Clock Input (SCK)	**3**
	2.1.1.2 Serial Data (DATA)	**3**
	2.1.2 การวัดอุณหภูมิและความชื้นสัมพัทธ์	**4**
	2.1.3 การแปลงข้อมูลอุณหภูมิและความชื้นเป็นหน่วยมาตรฐาน	**6**
	2.1.3.1 การแปลงข้อมูลความชื้นสัมพัทธ์	**6**
	2.1.3.2 การแปลงข้อมูลอุณหภูมิ	**6**
	2.2 ตัวอย่างในการวัดความชื้นสัมพัทธ์	**7**
บทที่ 3	**ส่วนควบคุมการทำงาน**	**9**
	3.1 สถาปัตยกรรมและโครงสร้างทางด้านฮาร์ดแวร์ ของ PIC 16F877	**9**
	3.2 การจัดสรรพื้นที่หน่วยความจำและรีจิสเตอร์ต่างๆ	**12**
	3.2.1 หน่วยความจำโปรแกรม	**12**
	3.2.2 หน่วยความจำข้อมูล	**13**
	3.3 รีจิสเตอร์ที่สำคัญต่าง ๆ ของ PIC 16F877	**15**
	3.3.1 รีจิสเตอร์ STATUS	**15**
	3.3.1.1 IRP0 (Indirect Register Bank Select bit)	**15**
	3.3.1.2 RP1,RP0 (Register Bank Select)	**16**
	3.3.1.3 TO (Time-out bit)	**16**
	3.3.1.4 PD (Power-down)	**16**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

3.3.1.5 Z (Zero bit)	16
3.3.1.6 DC (Digit carry/borrow)	16
3.3.1.7 C (Carry/borrow)	16
3.3.2 รีจิสเตอร์ W (Working register)	16
3.3.3 โปรแกรมเคาน์เตอร์ (PC)	17
3.4 PORT ใน PIC16F877	17
3.4.1 PORTA	17
3.4.2 PORTB	20
3.4.3 PORTC	23
3.4.4 PORTD และ PORTE	26
3.5 โฟลว์ชาร์จแสดงโปรแกรมควบคุมการวัดอุณหภูมิและความชื้น	31
3.5.1 โปรแกรมหลัก	31
3.5.2 โปรแกรมย่อยวัดอุณหภูมิและความชื้น	32
3.5.3 โปรแกรมย่อยเขียนค่าและอ่านค่า	33
3.5.4 โปรแกรมปรับอุณหภูมิให้เหมาะสมกับความชื้นสัมพัทธ์	34
บทที่ 4 การสื่อสารด้วยอินฟราเรดและภาคแสดงผล	35
4.1 ลักษณะของช่องสัญญาณที่ใช้เป็นรหัสควบคุม	35
4.2 สัญญาณโทนเบิรสต์	36
4.3 วงจรจ่ายกำลังหรือวงจรขับเอาท์พุท	37
4.4 ภาคการแสดงผล	38
บทที่ 5 การทดลองและสรุปผล	40
5.1 การทดสอบคุณสมบัติของความชื้นสัมพัทธ์ในห้องติดเครื่องปรับอากาศ	40
5.1.1 การทดลองที่ 1	40
5.1.2 ผลการทดลองที่ 1	40
5.1.3 สรุปผลการทดลองที่ 1	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

5.2 การทดสอบประสิทธิภาพของเซ็นเซอร์ **41**

5.2.1 การทดลองที่ 2 **41**

5.2.2 ผลการทดลองที่ 2 **42**

5.2.3 สรุปผลการทดลองที่ 2 **42**

5.3 การทดสอบการส่งสัญญาณด้วยอินฟราเรด **43**

5.3.1 การทดลองที่ 3 **43**

5.3.2 ผลการทดลองที่ 3 **43**

5.3.3 สรุปผลการทดลองที่ 3 **43**

ภาคผนวก **44**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

บทที่ 1 บทนำ

รูปที่ 1.1 แสดงความสัมพันธ์ระหว่างอุณหภูมิและความชื้นสัมพัทธ์ 2

บทที่ 2 การวัดอุณหภูมิและความชื้นด้วย SHT 15

รูปที่ 2.1 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ SHT 15 4

รูปที่ 2.2 แสดงคำสั่งให้ SHT 15 พร้อมที่จะทำงาน 4

รูปที่ 2.3 แสดงคำสั่งให้ SHT 15 ลบข้อมูลเพื่อที่พร้อมจะทำงานใหม่ 5

รูปที่ 2.4 แสดงตัวอย่างการวัดความชื้นสัมพัทธ์ 7

บทที่ 3 ส่วนควบคุมการทำงาน

รูปที่ 3.1 แสดงภาพบล็อกไดอะแกรมแสดงการทำงาน 9

รูปที่ 3.2 แสดงสถาปัตยกรรมภายในของ PIC 16F877 11

รูปที่ 3.3 แสดงตำแหน่งขาสัญญาณต่าง ๆ ของ PIC 16F877 12

รูปที่ 3.4 การจัดสรรพื้นที่หน่วยความจำของ PIC 16F877 13

รูปที่ 3.5 การจัดวางพื้นที่ของหน่วยความจำข้อมูล 14

รูปที่ 3.6 รายละเอียดในแต่ละบิต 15

รูปที่ 3.7 แสดงบล็อกไดอะแกรมขา RA3:RA0 18

รูปที่ 3.8 แสดงบล็อกไดอะแกรมขา RA4 19

รูปที่ 3.9 แสดงบล็อกไดอะแกรมขา RB3:RB0 21

รูปที่ 3.10 แสดงบล็อกไดอะแกรมขา RB7:RB4 22

รูปที่ 3.11 แสดงบล็อกไดอะแกรมขา RC2:RC0 และ RC7:RC5 24

รูปที่ 3.12 แสดงบล็อกไดอะแกรมขา RC4:RC3 25

รูปที่ 3.13 แสดงบล็อกไดอะแกรมขา RD ทั้งหมด 27

รูปที่ 3.14 แสดงบล็อกไดอะแกรมขา RE ทั้งหมด 29

รูปที่ 3.15 แสดงการทำงานของโปรแกรมหลัก 31

รูปที่ 3.16 แสดงการทำงานของโปรแกรมย่อยวัดอุณหภูมิและความชื้น 32

รูปที่ 3.17 แสดงการทำงานของโปรแกรมย่อยเขียนค่าและอ่านค่า 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ 3.18 แสดงการปรับอุณหภูมิให้เหมาะสมกับความชื้นสัมพัทธ์ **34**

บทที่ 4 การสื่อสารด้วยอินฟราเรดและภาคแสดงผล

รูปที่ 4.1 แสดงวงจรเครื่องควบคุมอุณหภูมิและวัดความชื้นสัมพัทธ์ **35**

รูปที่ 4.2 แสดงสัญญาณพัลส์ที่ใช้เป็นรหัสควบคุม **36**

รูปที่ 4.3 (ก) สัญญาณพัลส์ธรรมดา (ข) สัญญาณโทนเบิร์สต์ **37**

รูปที่ 4.4 แสดงวงจรส่งสัญญาณผ่าน LED อินฟราเรด **38**

รูปที่ 4.5 แสดงวงจรเชื่อมต่อระหว่างโมดูล LCD กับไมโครคอนโทรลเลอร์ **39**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

บทที่ 2 การวัดอุณหภูมิและความชื้นด้วย SHT 15

ตารางที่ 2.1 แสดงคุณสมบัติของเซ็นเซอร์ SHT 15 **3**

ตารางที่ 2.2 แสดงรหัสคำสั่ง 8 บิต **5**

ตารางที่ 2.3 **6**

ตารางที่ 2.4 **6**

ตารางที่ 2.5 **7**

ตารางที่ 2.6 **7**

บทที่ 3 ส่วนควบคุมการทำงาน

ตารางที่ 3.1 การกำหนดค่าในรีจิสเตอร์ **15**

ตารางที่ 3.2 แสดงฟังก์ชันการทำงานของ PORTA **20**

ตารางที่ 3.3 แสดงฟังก์ชันการทำงานของ PORT B **23**

ตารางที่ 3.4 แสดงฟังก์ชันการทำงานของ PORTC **26**

ตารางที่ 3.5 แสดงฟังก์ชันการทำงานของ PORTD **28**

ตารางที่ 3.6 แสดงฟังก์ชันการทำงานของ PORTE **30**

บทที่ 5 การทดลองและสรุปผล

ตารางที่ 5.1 แสดงค่าความชื้นสัมพัทธ์ในแต่ละชั่วโมง ณ อุณหภูมิ 25 องศาเซลเซียส ครั้งที่ 1 **40**

ตารางที่ 5.2 แสดงค่าความชื้นสัมพัทธ์ในแต่ละชั่วโมง ณ อุณหภูมิ 25 องศาเซลเซียส ครั้งที่ 2 **40**

ตารางที่ 5.3 แสดงค่าความชื้นสัมพัทธ์ในแต่ละชั่วโมง ณ อุณหภูมิ 25 องศาเซลเซียส ครั้งที่ 3 **41**

ตารางที่ 5.4 แสดงค่าอุณหภูมิเปรียบเทียบระหว่างการวัดด้วยเทอร์โมมิเตอร์ กับเซ็นเซอร์ SHT15 **42**

ตารางที่ 5.5 แสดงระยะที่สามารถส่งสัญญาณได้จากการเปลี่ยนค่าความต้านทาน **43**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ที่มา

เนื่องจากขณะอยู่ในห้องปรับอากาศเป็นระยะเวลานาน ผู้ที่อยู่ในห้องจะมีความรู้สึกหนาวขึ้นกว่าตอนที่เพิ่งเปิดเครื่องปรับอากาศใหม่ ทั้งที่อุณหภูมิยังคงที่อยู่

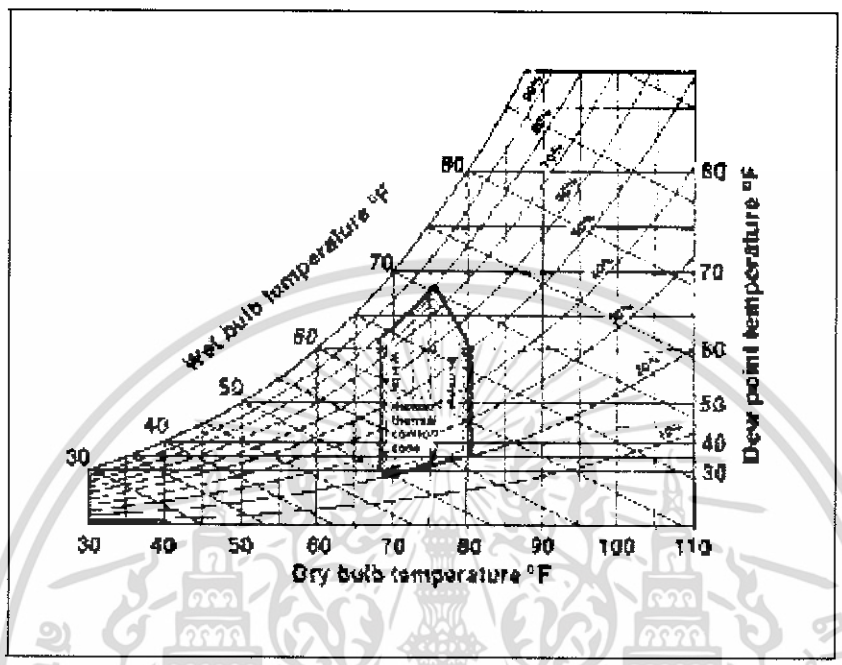
1.2 แนวความคิด

การที่ผู้อยู่ในห้องปรับอากาศเป็นระยะเวลานานแล้วรู้สึกหนาวขึ้นกว่าเมื่อตอนเพิ่งเปิดเครื่องปรับอากาศใหม่ๆนั้น เป็นเพราะมีปัจจัยอื่นที่เกี่ยวข้องกับอากาศ นั่นคือความชื้นสัมพัทธ์ (Humidity) ซึ่งส่งผลต่ออุณหภูมิ(Temperature)เมื่อความชื้นสัมพัทธ์ในอากาศมีค่ามากจะทำให้มีความรู้สึกร้อนเนื่องจากเหงื่อที่ผิวหนังระเหยได้ช้า แต่ในขณะที่ความชื้นสัมพัทธ์น้อย เหงื่อที่ผิวหนังจะระเหยได้ดีทำให้ร่างกายรู้สึกเย็นสบาย ในที่นี้เมื่อตอนเปิดเครื่องปรับอากาศใหม่ๆ ความชื้นสัมพัทธ์ยังมากอยู่ แต่เมื่อเวลาผ่านไปความชื้นสัมพัทธ์ในห้องจะลดลงเนื่องจากถูกพัดลมดูดความชื้นในห้องออกไป จึงทำให้ผู้อยู่ในห้องรู้สึกเย็นกว่าเดิมทั้งที่อุณหภูมิคงเดิมอยู่

1.3 หลักการทำงาน

เมื่อทำการวัดความชื้นสัมพัทธ์ในห้องที่มีเครื่องปรับอากาศลดลงจนถึงจุดหนึ่ง ไมโครคอนโทรเลอร์จะทำการส่งสัญญาณให้มีการเพิ่มอุณหภูมิไปยังเครื่องปรับอากาศ จากนั้นอีก 10 นาทีจะทำการตรวจสอบความชื้นสัมพัทธ์และอุณหภูมิมีความเหมาะสมกันหรือยัง ถ้าไม่เหมาะสมก็จะทำการปรับอุณหภูมิขึ้นอีกรอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.1 แสดงความสัมพันธ์ระหว่างอุณหภูมิและความชื้นสัมพัทธ์

จากรูปแสดงความสัมพันธ์ระหว่างอุณหภูมิและความชื้นสัมพัทธ์ช่วงที่ทำให้มนุษย์มีความสบาย ซึ่งสามารสรุปความสัมพันธ์นี้ได้ว่า เมื่อความชื้นสัมพัทธ์ลดลง แม้ว่าอุณหภูมิเพิ่มขึ้นก็ยังอยู่ในช่วงของ comfort zone ดังนั้นเราจึงใช้ความสัมพันธ์นี้มาใช้ในเครื่องปรับอากาศเพื่อเป็นการช่วยประหยัดพลังงาน โดยอ้างอิงจากกราฟประมาณได้ว่า

ณ อุณหภูมิ 25 องศาเซลเซียส ความชื้นสัมพัทธ์ที่เหมาะสมควรเป็นช่วงระหว่าง 20 – 80 %

ณ อุณหภูมิ 26 องศาเซลเซียส ความชื้นสัมพัทธ์ที่เหมาะสมควรเป็นช่วงระหว่าง 20 – 60 %

ณ อุณหภูมิ 27 องศาเซลเซียส ความชื้นสัมพัทธ์ที่เหมาะสมควรเป็นช่วงระหว่าง 20 – 50 %

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
การวัดอุณหภูมิและความชื้นด้วย SHT 15

SHT 15 เป็นเซ็นเซอร์ประเภทคาปาซิทีฟ(Capacitive) ที่สามารถตรวจจับความชื้นและอุณหภูมิได้ในตัวเดียวกัน ซึ่งภายในมีวงจรแปลง Analog เป็น Digital โดยมีการเชื่อมต่อแบบ 2-wire ส่งสัญญาณจากเซนเซอร์ให้กับไมโครคอนโทรลเลอร์กับ SHT 15

2.1 คุณสมบัติของเซ็นเซอร์ SHT 15

ตารางที่ 2.1 แสดงคุณสมบัติของเซ็นเซอร์ SHT 15

	ค่าน้อยสุด	ค่ามากสุด	หน่วย
แหล่งจ่ายไฟ	2.4	5.5	โวลท์
ความชื้น(HUMIDITY)			
ช่วงที่วัดได้	0	100	%RH
เวลาในการวัด	4	4	วินาที
ความแม่นยำ	±2.0	±2.0	%RH
อุณหภูมิ(TEMPERATURE)			
ช่วงที่วัดได้	-40	254.9	องศาฟาเรนไฮต์
	-40	123.8	องศาเซลเซียส
เวลาในการวัด	5	30	วินาที
ความแม่ยำ	±0.3	±0.3	เคลวิน

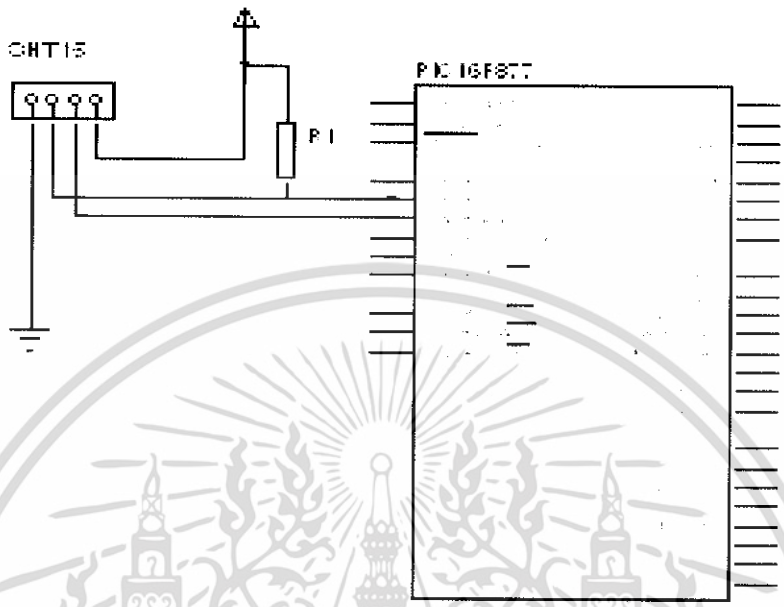
2.1.1 การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ SHT 15

การเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ SHT 15 เป็น แบบ 2-wire Interface ประกอบด้วยสายสัญญาณ 2 เส้น คือ

2.1.1.1 Serial Clock Input (SCK) เป็นการเชื่อมต่อแบบซิงโครนัสระหว่างไมโครคอนโทรล เลอร์กับ SHT 15 โดยที่สัญญาณ SCK ไม่มีข้อจำกัดความถี่ต่ำสุด

2.1.1.2 Serial Data (DATA) เป็นสายส่งข้อมูลรับส่งระหว่างไมโครคอนโทรลเลอร์กับ SHT 15 ซึ่งจะมีความต้านทาน 10 กิโลโอห์มต่อ pull up ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดงการเชื่อมต่อระหว่างไมโครคอนโทรลเลอร์กับ SHT 15

2.1.2 การวัดอุณหภูมิและความชื้นสัมพัทธ์

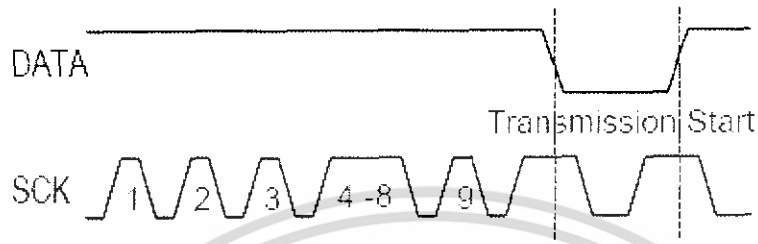
2.1.2.1 ในการส่งคำสั่ง จะต้องมีความสอดคล้องกันระหว่าง SCK กับ DATA โดยที่สัญญาณ DATA จะเปลี่ยนก็ต่อเมื่อสัญญาณ SCK อยู่ในช่วงขอบขาลงจนช่วงขอบขาขึ้นต่อไป

2.1.2.1.1 คำสั่งให้ SHT 15 พร้อมที่จะทำงาน

รูปที่ 2.2 แสดงคำสั่งให้ SHT 15 พร้อมที่จะทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.2.1.2 คำสั่งให้ SHT 15 ลบข้อมูลเพื่อที่พร้อมจะทำงานใหม่



รูปที่ 2.3 แสดงคำสั่งให้ SHT 15 ลบข้อมูลเพื่อที่พร้อมจะทำงานใหม่

2.1.2.1.3 คำสั่งวัดอุณหภูมิและความชื้น เมื่อ SHT 15 พร้อมที่จะทำงานแล้ว เราสามารถที่จะส่งคำสั่งวัดอุณหภูมิหรือความชื้นได้ โดยส่งข้อมูล 8 บิต ได้ตามตารางข้างล่าง

ตารางที่ 2.2 แสดงรหัสคำสั่ง 8 บิต

Command	Code
Reserved	0000000x
Measure Temperature	00000011
Measure Humidity	00000101
Read Status Register	00000111
Write Status Register	00000110
Soft Reset	00011110

เมื่อสัญญาณ SCK clock ถึงครั้งที่ 8 SHT 15 จะส่งสัญญาณ DATA บิต 0 ในสัญญาณ SCK clock ครั้งที่ 9 จนถึงขอบขาลงของ SCK clock สัญญาณ DATA จึงมีบิตเป็น 1

2.1.2.2 หลังจากเสร็จสิ้นการส่งคำสั่งแล้ว จะต้องคอยให้ SHT 15 ทำการวัดสมบูรณ์ก่อน โดยใช้เวลาประมาณ 55 , 210 มิลลิวินาที ในการส่งข้อมูล 12 , 14 บิต ตามลำดับ โดยเมื่อการวัดเสร็จสมบูรณ์แล้ว SHT 15 จะส่งสัญญาณ DATA บิต 0 มาให้ไมโครคอนโทรลเลอร์รับรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการวัดอุณหภูมิและความชื้น ค่าที่อ่านได้จะเป็นข้อมูล 14 บิต และ 12 บิต ตามลำดับ โดยส่งข้อมูลมาทีละ 8 บิต ไมโครคอนโทรลเลอร์ต้องส่งข้อมูลบิต 0 ไป ใน SCK clock ครั้งที่ 9 เพื่อแสดงว่าได้รับข้อมูลแล้ว จากนั้น SHT 15 จะส่งข้อมูลมาอีก 8 บิต และ ไมโครคอนโทรลเลอร์ต้องส่งข้อมูลบิต 0 ไปใน SCK clock ครั้งที่ 18 เพื่อแสดงว่าได้รับข้อมูลแล้ว

2.1.3 การแปลงข้อมูลอุณหภูมิและความชื้นเป็นหน่วยมาตรฐาน

2.1.3.1 การแปลงข้อมูลความชื้นสัมพัทธ์

$$RH_{linear} = C_1 + C_2 . SO_{RH} + C_3 . (SO_{RH})^2$$

โดยที่

SO_{RH} เป็นค่าที่ได้จากการแปลงเลขฐานสองจากข้อมูลที่วัดได้จาก SHT 15 เป็นเลขฐานสิบ

ตารางที่ 2.3

SO_{RH}	C_1	C_2	C_3
12 bit	-4	0.0405	-2.8×10^{-6}
8 bit	-4	0.648	-7.2×10^{-4}

จากสมการเป็นการวัดสัญญาณที่อุณหภูมิ 25 องศาเซลเซียส (77 F) ในกรณีที่อุณหภูมิเปลี่ยนแปลงจาก 25 องศาเซลเซียส จะมีผลต่อความชื้นสัมพัทธ์

$$RH_{true} = (T_C . 25) . (t_1 + t_2 . SO_{RH}) + RH_{linear}$$

โดยที่

ตารางที่ 2.4

SO_{RH}	t_1	t_2
12 bit	0.01	0.00008
8 bit	0.01	0.00128

2.1.3.2 การแปลงข้อมูลอุณหภูมิ

$$Temperature = d_1 + d_2 . SO_T$$

โดยที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.5

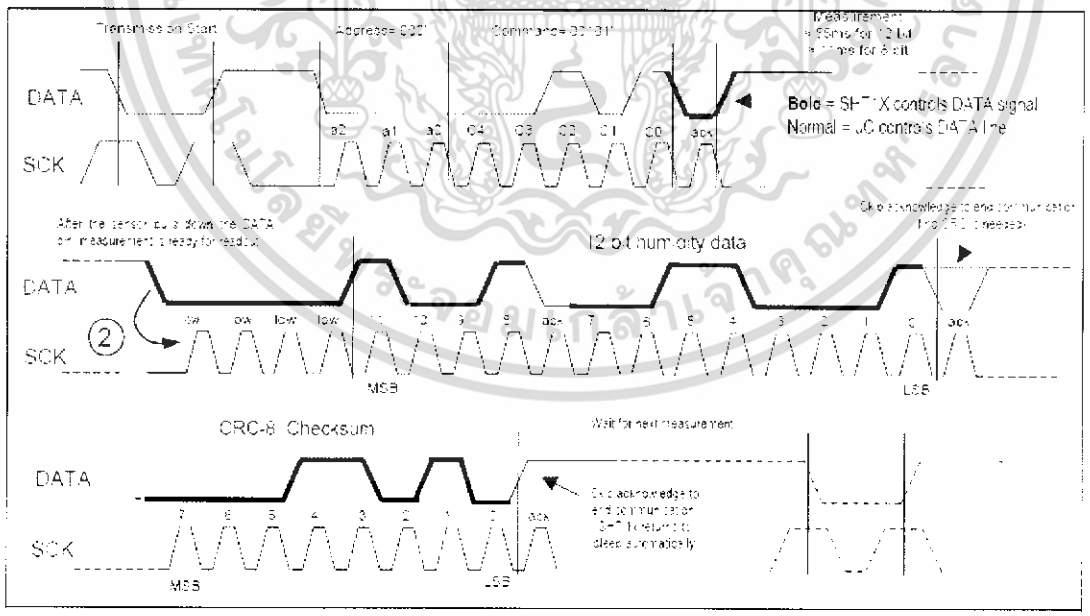
SO_T	d_2 (C)	d_2 (F)
14 bit	0.01	0.018
12 bit	0.04	0.072

และ

ตารางที่ 2.6

V_{dd}	d_1 (C)	d_1 (F)
5 V	-40.00	-40.00
4 V	-39.75	-39.50
3.5 V	-39.66	-39.35
3 V	-39.60	-39.28
2.5 V	-39.55	-39.23

2.2 ตัวอย่างในการวัดความชื้นสัมพัทธ์



รูปที่ 2.4 แสดงตัวอย่างการวัดความชื้นสัมพัทธ์

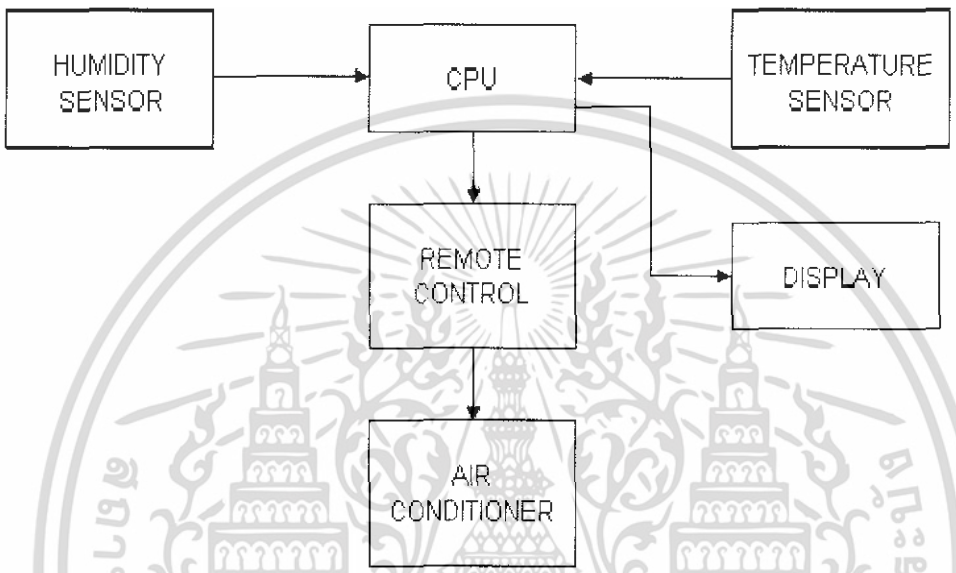
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าที่ได้จากการวัด “0000 1001 0011 0001” ทำเป็นเลขฐานสิบเท่ากับ 2353 แทนในสมการ

$$RH_{true} = (T_C . 25) . (t_1 + t_2 . SO_{RH}) + RH_{linear} = 75.79\%RH$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ส่วนควบคุมการทำงาน



รูปที่ 3.1 แสดงภาพบล็อกไดอะแกรมแสดงการทำงาน

3.1 สถาปัตยกรรมและโครงสร้างทางด้านฮาร์ดแวร์ ของ PIC 16F877

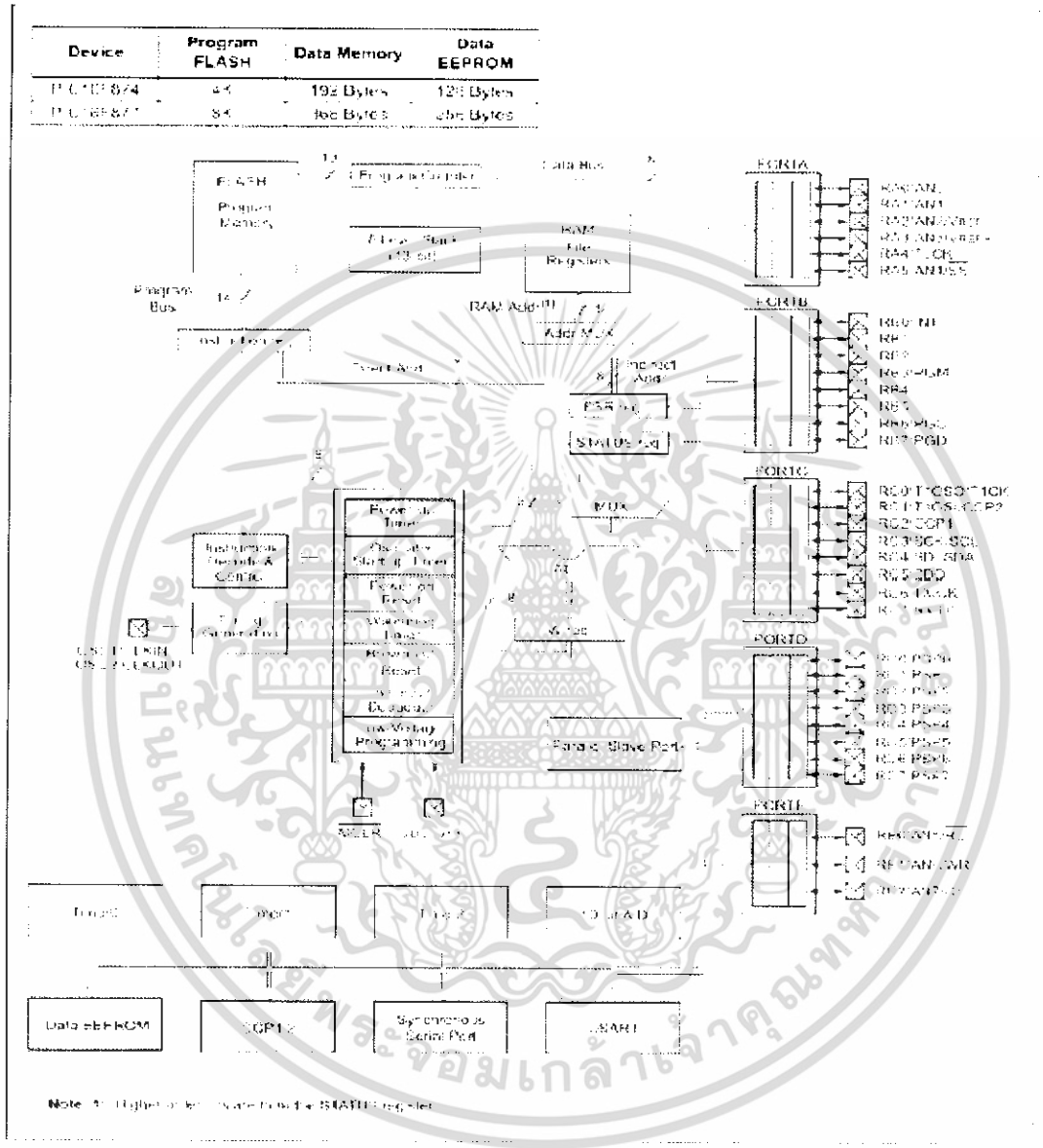
PIC 16F877 เป็น CPU ของบริษัท Microchip Technology ซึ่งเป็นผู้ผลิต CPU ตระกูล PIC โดยในเบอร์ 16F877 นี้เป็นเบอร์ที่มีความสามารถสูงพอสมควร ประกอบไปด้วยฟังก์ชันการทำงานต่างๆ มากมาย พอจะสรุปคุณสมบัติคร่าวๆ ได้ดังนี้

- มี 35 Instruction คำสั่ง
- ในการปฏิบัติงานคำสั่งต่างๆจะใช้ Cycle เดียว และ 2 Cycle ในคำสั่งที่เป็นการกระโดด
- ความถี่สูงสุดที่ทำงานได้คือ 20 MHz (16F877-20/P)
- การทำงานจะเป็นลักษณะ Pipeline ทำให้มีการทำงานที่เร็วขึ้น
- หน่วยความจำโปรแกรม FLASH Program Memory มีขนาด 8k (14-Bit Words)
- หน่วยความจำข้อมูล (RAM) 368 Bytes

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- หน่วยความจำข้อมูล (EEPROM) 256 Bytes
- สามารถตอบสนองการอินเตอร์รับป์ได้ถึง 14 แหล่ง
- มี STACK 8 ระดับ
- มีเพาเวอร์ออนรีเซต(POR),เพาเวอร์อัปไทเมอร์(PWRT) และ Oscillator Start-Up Timer
- Watchdog Timer
- สามารถเลือกการป้องกันข้อมูลได้ (Code Protection)
- มีโหมดประหยัดพลังงาน (Sleep Mode)
- เลือกโหมดของ สัญญาณนาฬิกาได้หลายโหมด
- สามารถโปรแกรมโดยใช้แรงดัน +5V ได้
- มีฟังก์ชันการโปรแกรมแบบ ICSP (In-Circuit Serial Programing)
- ทำงานที่ไฟเลี้ยง 2.0V ถึง 5.5V
- กระแสทั้งซิงก์และซอร์สของพอร์ตคือ 25mA
- มี Timer/Counter จำนวน 3 ตัว คือ Timer0,Timer1 และ Timer2
- มีโมดูล Capture/Compare/PWM จำนวน 2 ชุด
- มี Analog to Digital Converter ความละเอียด 10 บิต 8 แชนเนล ภายในตัว
- มีโมดูลการสื่อสาร USART
- มีโมดูลตรวจจับระดับไฟเลี้ยง Brown – out reset (BOR)
- มีพอร์ต I/O 5 พอร์ตประกอบด้วย A,B,C,D และ E แต่ละพอร์ตจะมีจำนวนบิตไม่เท่ากันซึ่งรวมแล้วจะมี I/O จำนวน 33 บิต

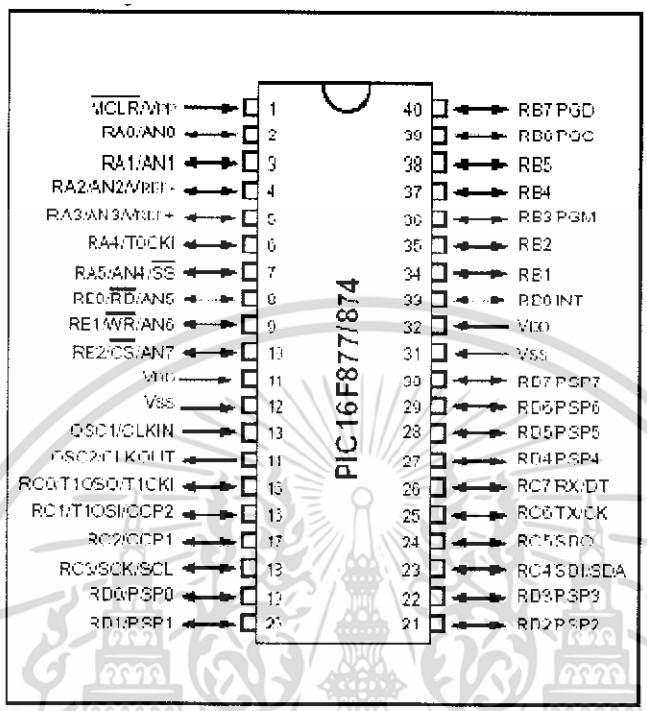
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes

รูปที่ 3.2 แสดงสถาปัตยกรรมภายในของ PIC 16F877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงตำแหน่งขาสัญญาณต่าง ๆ ของ PIC 16F877

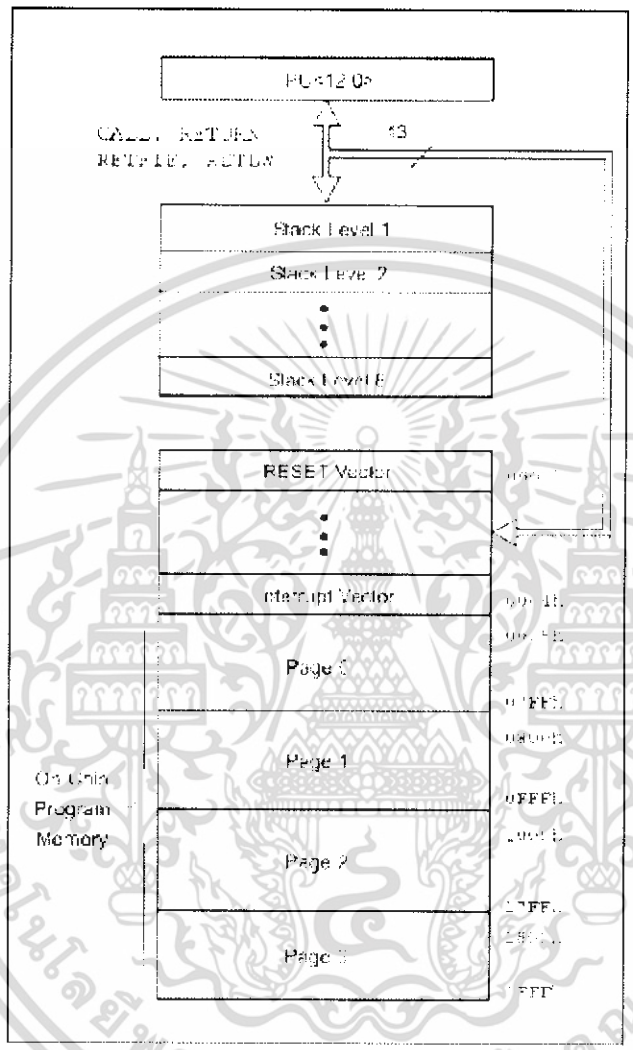
3.2 การจัดสรรพื้นที่หน่วยความจำและรีจิสเตอร์ต่างๆ

การจัดหน่วยความจำของ PIC 16F877 นี้จะแบ่งออกเป็น 3 ส่วนคือ หน่วยความจำโปรแกรม , หน่วยความจำข้อมูล(RAM) และหน่วยความจำข้อมูลที่เป็น EEPROM ซึ่งเราจะมาดูในแต่ละส่วนดังนี้

3.2.1 หน่วยความจำโปรแกรม

PIC 16F877 นี้มี Program Counter ขนาด 13 บิตซึ่งสามารถอ้างถึงตำแหน่งข้อมูลได้ถึง 8 กิโลเวิร์ด โดยจะมีตำแหน่ง Reset Vector ที่ 0000h และ Interupt Vector ที่ 0004h ดังนั้นในการเขียนโปรแกรมจึงควรสงวนพื้นที่ส่วนนี้ไว้สำหรับการใช้งานอินเตอร์รัพท์ จากรูปด้านล่างจะเห็นว่ามีพื้นที่ของ Stack 8 ระดับ และ หน่วยความจำโปรแกรมแบ่งออกเป็น 4 Page (8 kwords) ซึ่งพื้นที่ในส่วนนี้มีไว้สำหรับเก็บข้อมูลคำสั่งทั้งหมดโดยโครงสร้างจะเป็นแบบแฟลช (flash memory) ทำให้ลบและเขียนใหม่ได้หลายครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

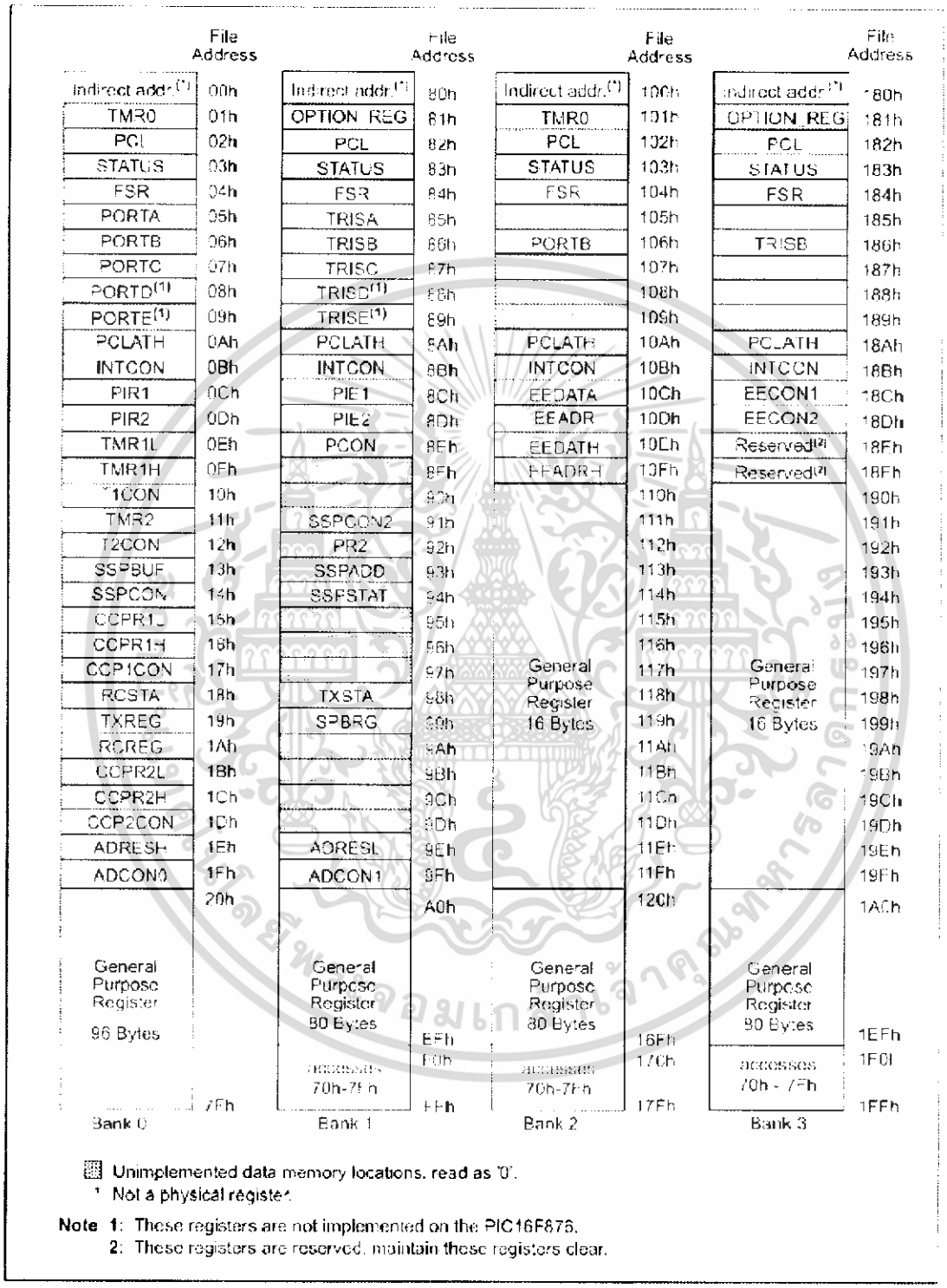


รูปที่ 3.4 การจัดสรรพื้นที่หน่วยความจำของ PIC 16F877

3.2.2 หน่วยความจำข้อมูล

ใน PIC 16F877 นี้หน่วยความจำข้อมูลจะแบ่งออกเป็นพื้นที่ของ RAM หน่วยความจำใช้งานทั่วไป (Gerneral Purpose Register) ขนาด 368 Bytes และ พื้นที่ของ รีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Registers) ในการจัดวางพื้นที่จะแบ่งออกเป็น 4 แบงค์ ตั้งแต่แอดเดรส 00h ถึง 1FFh ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การจัดวางพื้นที่ของหน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งจากรูป ในการเข้าถึงข้อมูลในแต่ละส่วน หรือ แต่ละแบงก์ สามารถทำได้โดยการกำหนดค่าในรีจิสเตอร์ STATUS ในบิตที่ 5 และ 6 (RP0,RP1) ซึ่งมีความหมายดังนี้

ตารางที่ 3.1 การกำหนดค่าในรีจิสเตอร์

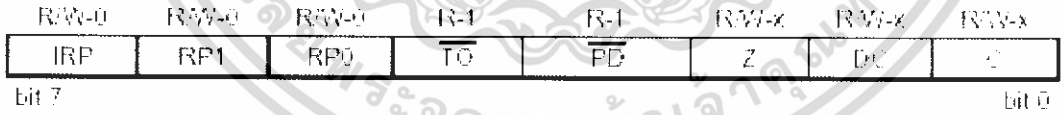
RP1	RP0	Bank Select
0	0	Bank0 : 00h – 7Fh
0	1	Bank1 : 80h – FFh
1	0	Bank2 : 100h – 17Fh
1	1	Bank3 : 180h – 1FFh

3.3 รีจิสเตอร์ที่สำคัญต่าง ๆ ของ PIC 16F877

รีจิสเตอร์จัดได้ว่าเป็นส่วนประกอบที่มีความสำคัญต่อการพัฒนาโปรแกรมเป็นอย่างมาก ซึ่งใน PIC 16F877 นี้มีรีจิสเตอร์ต่างๆ มากมาย เราจะมาดูเฉพาะรีจิสเตอร์ที่ใช้งานหลักๆ ดังนี้

3.3.1 รีจิสเตอร์ STATUS

เป็นรีจิสเตอร์ที่ใช้เก็บข้อมูลสถานะ การทำงานของไมโครคอนโทรลเลอร์ ไม่ว่าจะเป็นแฟลกสถานะต่างๆ ที่ใช้บอกผลลัพธ์การทำงานของส่วนต่างๆ เช่น การคำนวณทางคณิตศาสตร์ การเลือกแบงก์ข้อมูล ซึ่งมีรายละเอียดในแต่ละบิตดังนี้



รูปที่ 3.6 รายละเอียดในแต่ละบิต

3.3.1.1 IRP0 (Indirect Register Bank Select bit – บิต 7) ใช้เลือกแบงก์ในหน่วยความจำชนิด RAM ในกรณีที่ใช้การอ้างถึงข้อมูลแบบทางอ้อม (indirect addressing mode)

"0" : เลือกแบงก์ 0 และ 1

"1" : เลือกแบงก์ 2 และ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1.2 RP1,RP0 (Register Bank Select) ใช้เลือกแบงก์ของหน่วยความจำข้อมูลแรมและ รีจิสเตอร์ไฟล์เมื่อเป็นการเข้าถึงข้อมูลแบบโดยตรง (direct addressing mode)

"00" เลือกแบงก์ 0

"01" เลือกแบงก์ 1

"10" เลือกแบงก์ 2

"11" เลือกแบงก์ 3

3.3.1.3 TO (Time-out bit) เป็นบิตที่แสดงการเกิดไทม์เอาต์ของวอตช์ด็อกไทเมอร์ แอกทีฟลอจิก "0"

3.3.1.4 PD (Power-down) บิตแสดงการทำงานในโหมดสลีป (Sleep mode) คือ เมื่อเข้าสู่ Sleep mode บิตนี้จะกลายเป็น "0"

3.3.1.5 Z (Zero bit) เป็นบิตแสดงสถานะการทำงานทางคณิตศาสตร์

"0" เมื่อผลลัพธ์ทางคณิตศาสตร์ลอจิกไม่เป็นศูนย์

"1" เมื่อผลลัพธ์ทางคณิตศาสตร์ลอจิกเป็นศูนย์

3.3.1.6 DC (Digit carry/borrow) เป็นบิตทดหรือยืมระหว่างหลัก

"0" เมื่อไม่เกิดการทดจากบิต 3 ไปบิต 4 หรือ หากเกิดการยืมจากบิต 4 มาบิต 3

"1" เมื่อเกิดการทดจากบิต 3 ไปบิต 4 หรือ หากไม่เกิดเกิดการยืมจากบิต 4 มาบิต 3

3.3.1.7 C (Carry/borrow) บิตทดหรือยืม ใช้แสดงสถานะการทด หรือการยืมทางคณิตศาสตร์

"0" เมื่อไม่มีการทดบิต 7 (MSB) หรือ เกิดการยืมจากบิต 7 (MSB)

"1" เมื่อมีการทดจากบิต 7 (MSB) หรือ ไม่เกิดการยืมค่าของบิต 7 (MSB)

3.3.2 รีจิสเตอร์ W (Working register)

เป็นรีจิสเตอร์ที่มีความสำคัญมากที่สุดตัวหนึ่งของไมโครคอนโทรลเลอร์ PIC ซึ่งรีจิสเตอร์ W เป็น รีจิสเตอร์ขนาด 8 บิต จะทำหน้าที่เหมือนเป็นรีจิสเตอร์ แอกคิวมูลเลเตอร์ ในการทำคำสั่งทางคณิตศาสตร์หรือการโอนย้ายข้อมูล จะต้องผ่านรีจิสเอตร์ W ทั้งสิ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

3.3.3 โปรแกรมเคาน์เตอร์ (PC)

เป็นรีจิสเตอร์ที่สำคัญอีกตัวหนึ่ง ซึ่งทำหน้าที่เก็บตำแหน่งแอดเดรสที่จะให้ CPU ไปทำงาน โดยรีจิสเตอร์ PC ของ PIC 16F877 จะมีขนาด 13 บิต โดย 8 บิตล่าง PC< 7:0 > จะอยู่ที่รีจิสเตอร์ PCL สามารถอ่านและเขียนได้เหมือนรีจิสเตอร์ทั่วไป ส่วน PC<12 :8 > จะไม่สามารถเข้าถึงได้ตามปกติ การเข้าถึงจะกระทำ ผ่านรีจิสเตอร์ PCLATH < 4 :0 > และเมื่อเกิดการรีเซต PCLATH จะเคลียร์ สถานะเป็น "0" ในการใช้คำสั่งในการกระโดดเช่น CALL หรือ GOTO จะเป็นการนำค่าแอดเดรสมาใส่ในรีจิสเตอร์ PC <10:0 > เพียง 11 บิตเท่านั้นส่วน PC<12:11> จะไม่เปลี่ยนแปลง ทำให้การใช้คำสั่ง CALL หรือ GOTO ไปได้ไกลเพียง 2Kwords (2048 ตำแหน่ง) แต่หน่วยความจำทั้งหมดมี 8 Kwords แบ่งเป็น 4 Page ฉะนั้นในการกระโดดข้าม Page จะต้องมีการกำหนดค่าให้กับ PC<12:11> ด้วยโดยผ่านทางรีจิสเตอร์PCLATH

3.4 PORT ใน PIC16F877

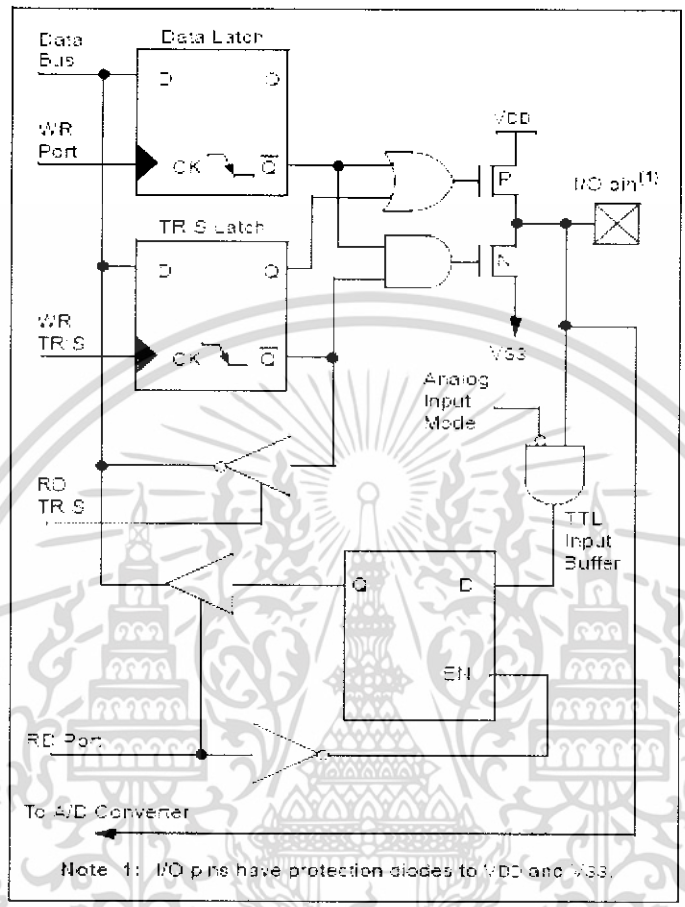
PIC16F877 นี้จะมีพอร์ตให้ใช้งานทั้งหมด 5 Port คือ PORTA(7Bit) , PORTB(8Bit) , PORTC(8Bit) , PORTD(8Bit) และ PORTE(3Bit) โดยแต่ละบิตสามารถกำหนดการทำงานให้เป็นอินพุตหรือเอาท์พุตได้อย่างอิสระ

3.4.1 PORTA

PORTA มีขนาด 6 bit ซึ่งเป็น port ที่เป็นได้ทั้ง Input และ Output โดยต้องเลือกแบบใดแบบหนึ่ง สามารถเลือกได้จาก register ที่มีชื่อว่า TRISA ซึ่งถ้า TRISA bit ถูก set เป็น '1' PORTA ที่มีหมายเลขบิตเดียวกันนั้นก็ทำงานเป็น input (ทำให้ port นั้นอยู่ในสถานะ hi-impedance) ส่วนถ้า TRISA bit ถูก set เป็น '0' PORTA ที่มีหมายเลขบิตเดียวกันนั้นก็จะทำงานเป็น output (port จะอยู่ในสถานะ output latch) การอ่านค่า PORTA register คือการอ่านค่าสถานะของขา PORTA ในขณะนั้น ส่วนการเขียนค่าไปยัง PORTA คือการเขียนไปยัง latch ของ port ลักษณะการเขียนจะเป็นแบบ read-modify-write operations ซึ่งหมายความว่า ในการเขียนไปยัง port จะเริ่มด้วยการอ่านค่า port นั้นมาก่อนแล้วทำการเปลี่ยนแปลงค่า จากนั้นก็ทำการเขียนกลับไปยัง port latch อีกครั้งหนึ่ง

73166

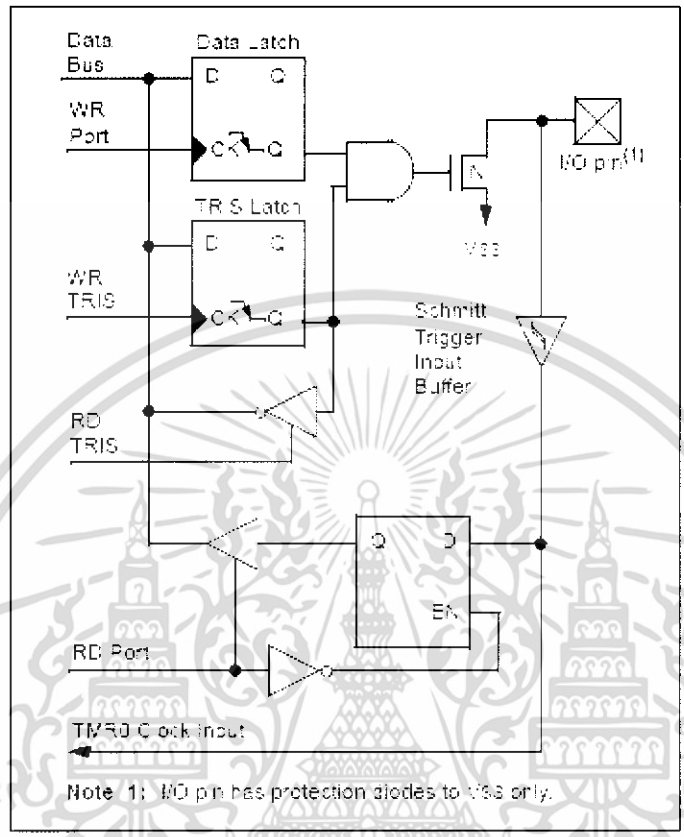
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 แสดงบล็อกไดอะแกรมขา RA3:RA0

ขา RA4 สามารถทำหน้าที่เป็น RA4/T0CKI โดยที่ขา RA4/T0CKI จะเป็นลักษณะ Input แบบ Schmitt Trigger และมี Output แบบ open drain โดย Port RA อื่นๆ จะมี TTL input level และ มี output แบบ full CMOS drivers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงบล็อกไดอะแกรมขา RA4

ในขณะที่เกิด Power-on Reset ขาเหล่านี้จะถูกกำหนดให้เป็น analog input และจะอ่านค่าได้เป็น ‘0’

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 แสดงฟังก์ชันการทำงานของ PORTA

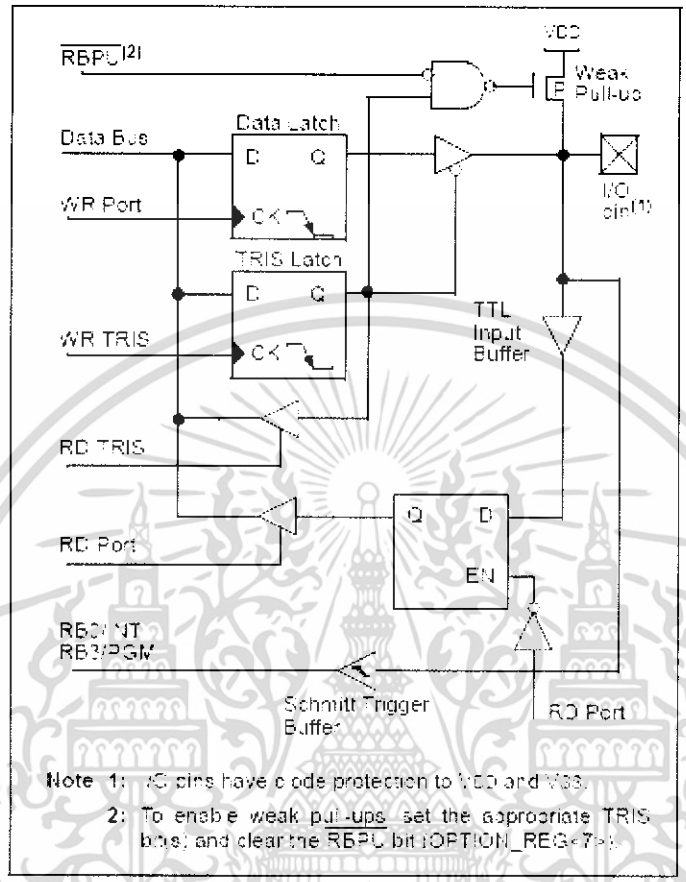
Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/SS/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

Legend: TTL = TTL input, ST = Schmitt Trigger input

3.4.2 PORTB

PORTB เป็นลักษณะแบบ Port แบบสองทิศทาง ซึ่ง register ที่จะเป็นตัวกำหนดว่า port ใดจะเป็นแบบ input/output จะถูกกำหนดโดย TRISB register ถ้า set TRISB bit ใด (=1) PORTB ที่บิตนั้นก็จะเป็น input ถ้า clear TRISB bit ใด (=0) PORTB ที่บิตนั้นก็จะเป็น output ขาสามขาของ PORTB จะ multiplexed กับ Low Voltage Programming function ซึ่งได้แก่ RB3/PGM, RB6/PGC และ RB7/PGD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

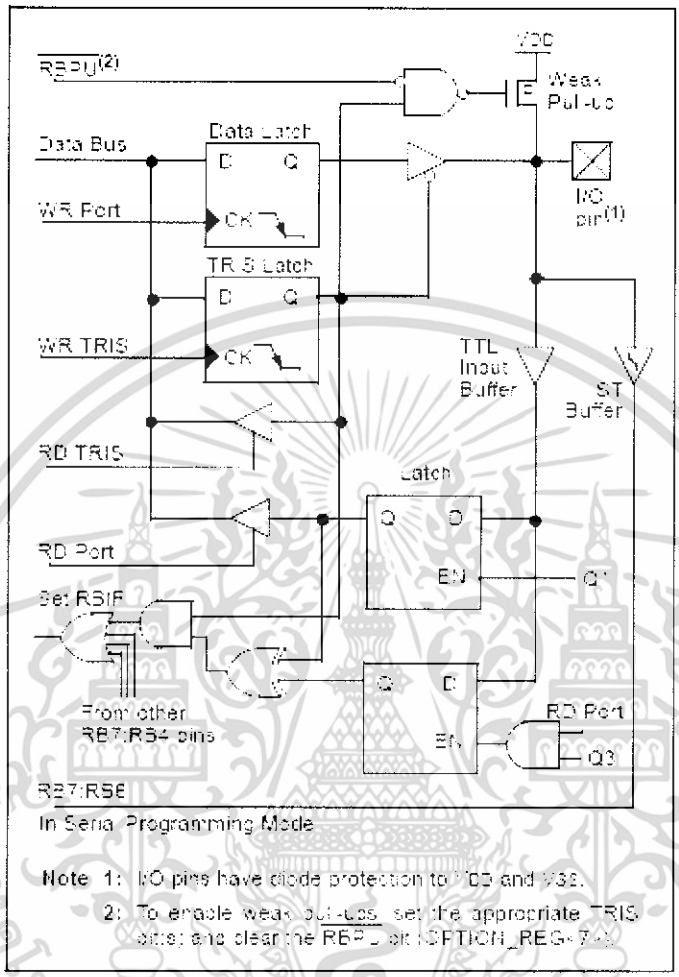


รูปที่ 3.9 แสดงบล็อกไดอะแกรมขา RB3:RB0

PORTB แต่ละ port จะมี weak pull-up อยู่ภายใน (ถ้าต้องการ pull-up ต้องต่อวงจรภายนอก) เราสามารถกำหนดว่าจะใช้ pull-up ภายในหรือจากการ set หรือ clear RBPU\ (OPTION register บิต 7) โดยถ้าเรา clear RBPU\ จะหมายถึง เราทำการ disable pull-up ภายใน และถ้าเรากำหนดให้ PORTB เป็น OUTPUT แล้ว pull-up จะถูก disable โดยอัตโนมัติ

สำหรับ ขา RB4-RB7 จะมีลักษณะเพิ่มเติมก็คือ การกำหนดให้เกิด Interrupt เมื่อเกิดการเปลี่ยนแปลงของสถานะของสัญญาณไฟฟ้าที่ขา RB4-RB7 (โดยถ้าขาใดขาหนึ่งเกิดเปลี่ยนสถานะก็จะทำให้เกิด RB Port Change Interrupt ขึ้น ซึ่งจะทำให้ RBIF (INTCON.0) flag ถูก set โดยที่ Interrupt ประเภทนี้สามารถทำการ “wake” microcontroller จากสถานะ sleep mode ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงบล็อกไดอะแกรมขา RB7:RB4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 แสดงฟังก์ชันการทำงานของ PORT B

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST(1)	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM(3)	bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST(2)	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST(2)	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

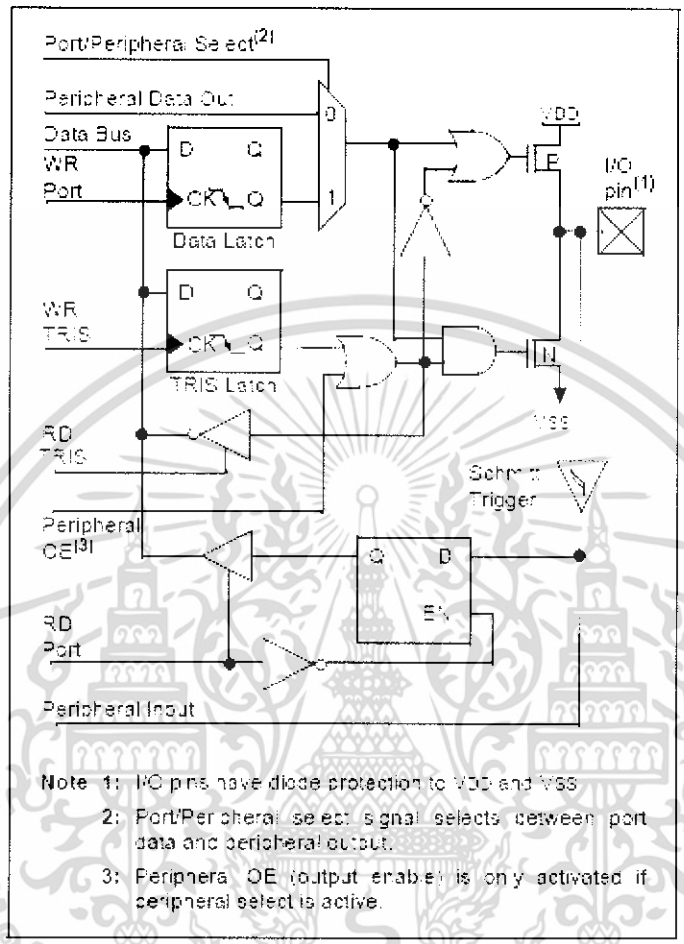
Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: Low Voltage ICSP Programming (LVP) is enabled by default, which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

3.4.3 PORTC

PORTC เป็นลักษณะแบบ Port แบบสองทิศทาง ซึ่ง register ที่จะเป็นตัวกำหนดว่า port ใดจะเป็นแบบ input/output จะถูกกำหนดโดย TRISC register ถ้า set TRISC bit ใด (=1) PORTC ที่บิตนั้นก็จะเป็น input ถ้า clear TRISC bit ใด (=0) PORTC ที่บิตนั้นก็จะเป็น output ที่ PORTC จะมีคุณสมบัติเพิ่มเติม เช่น IIC, UART, SPI, PWM, CAPTURE ขึ้นอยู่กับการเลือกใช้งาน โดยเมื่อเราทำการ enable คุณสมบัติเพิ่มเติมต่างๆ ที่ PORTC เราต้องระวังในเรื่องของการตั้งค่า TRISC ของแต่ละขาของ PORTC เพราะในการ enable คุณสมบัติบางตัวที่อยู่ที่ PORTC (เช่น UART) ตัวมันเองก็จะทำการเปลี่ยน bit TRISC โดยอัตโนมัติ ดังนั้นไม่ควรที่จะตั้งค่า TRISC โดยตรงกับขาใดของ PORTC ที่ทำการ enable PORTC แต่ละ port จะ Schmitt Trigger input buffers อยู่ภายในแต่ละขา

ลักษณะโครงสร้างของ PORT จะแบ่งเป็น 2 กลุ่มคือ PORTC0-2,5-7 และอีกกลุ่มหนึ่งก็คือ PORTC3-4

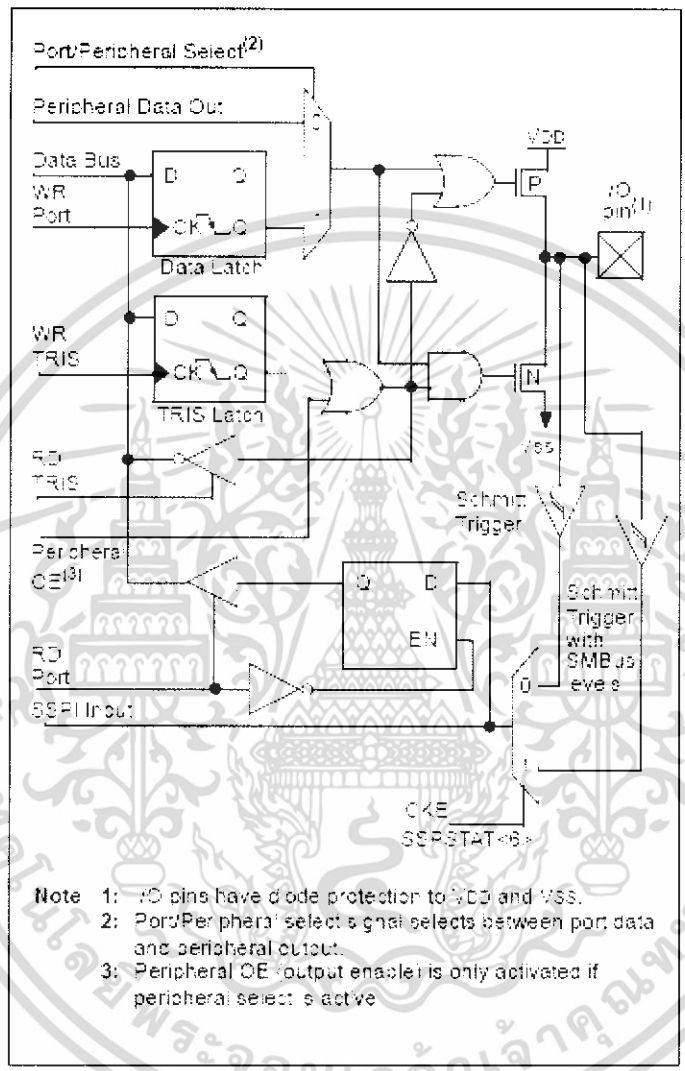
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงบล็อกไดอะแกรมขา RC2:RC0 และ RC7:RC5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORTC3-4 เหมาะสำหรับการเชื่อมต่อแบบ IIR



รูปที่ 3.12 แสดงบล็อกไดอะแกรมขา RC4:RC3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 แสดงฟังก์ชันการทำงานของ PORTC

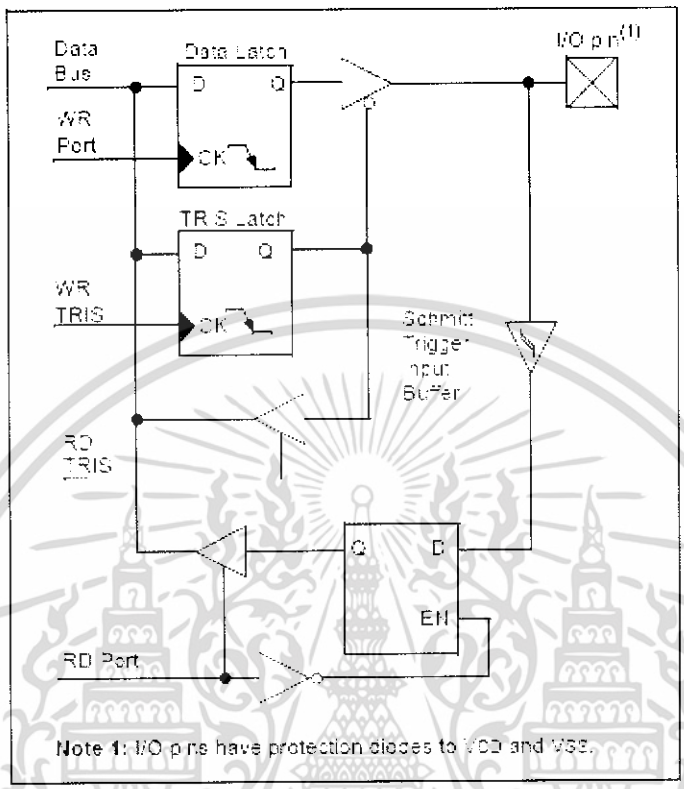
Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/ Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/ PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I^2C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

Legend: ST = Schmitt Trigger input

3.4.4 PORTD และ PORTE

สำหรับ PORTD และ PORTE นั้นจะไม่มีอยู่ใน PIC ในตระกูลนี้ที่มีขนาดขา 28 ขา โดยที่ PORTD จะเป็น port ขนาด 8 bits ซึ่งจะมี Schmitt Trigger input buffer อยู่ในตัว โดยที่เราสามารถกำหนดแต่ละบิตของ port ให้เป็น input หรือ output ได้โดยอิสระจากกัน PORTD สามารถที่จะทำตัวเป็น parallel slave port ได้อีกด้วย โดยทำได้โดยการ set PSPMODE bit (TRISE<4>) ซึ่งใน mode นี้ buffer ภายในจะกลายเป็นแบบ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.13 แสดงบล็อกไดอะแกรมขา RD ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.5 แสดงฟังก์ชันการทำงานของ PORTD

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL(1)	Input/output port pin or parallel slave port bit0.
RD1/PSP1	bit1	ST/TTL(1)	Input/output port pin or parallel slave port bit1.
RD2/PSP2	bit2	ST/TTL(1)	Input/output port pin or parallel slave port bit2.
RD3/PSP3	bit3	ST/TTL(1)	Input/output port pin or parallel slave port bit3.
RD4/PSP4	bit4	ST/TTL(1)	Input/output port pin or parallel slave port bit4.
RD5/PSP5	bit5	ST/TTL(1)	Input/output port pin or parallel slave port bit5.
RD6/PSP6	bit6	ST/TTL(1)	Input/output port pin or parallel slave port bit6.
RD7/PSP7	bit7	ST/TTL(1)	Input/output port pin or parallel slave port bit7.

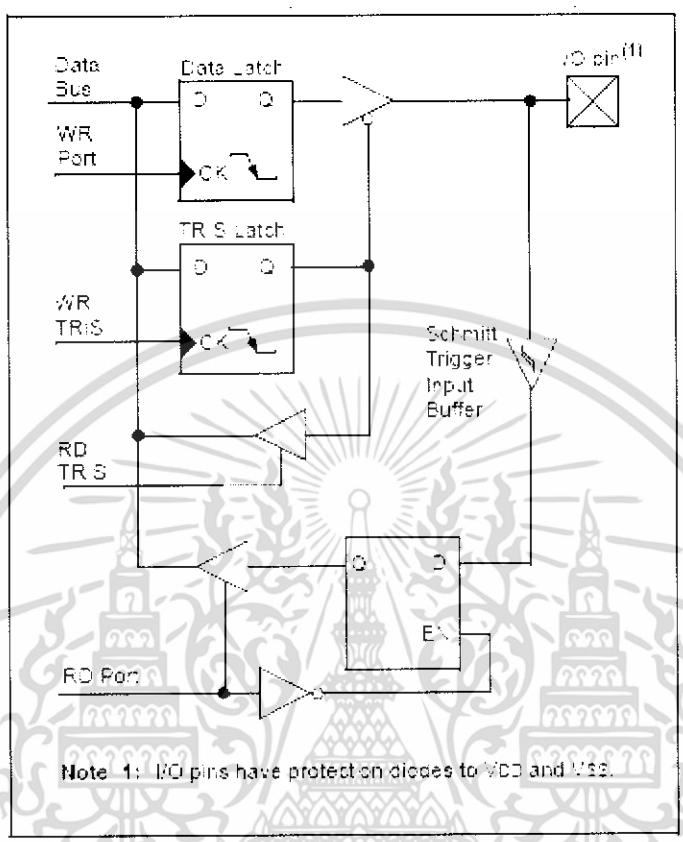
Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

PORTE จะมีทั้งหมด 3 ขา คือ RE0/(RD\)/AN5, RE1/(WR\)/AN6 และ RE2/(CS\)/AN7 ซึ่งจะมี Schmitt Trigger input buffer อยู่ในตัว โดยที่เราสามารถกำหนดแต่ละบิตของ port ให้เป็น input หรือ output I/O PORTE สามารถกลายเป็น control input สำหรับ microprocessor port เมื่อ ทำการ set PSPMODE(TRISE<4>) bit ข้อควรระวังเมื่ออยู่ในโหมดนี้ก็คือ ต้องตรวจดูให้ดีว่า TRISE ตั้งแต่บิต 0-2 ถูก set (อยู่ในสถานะ input) และต้องแน่ใจว่า ADCON1 ถูก set ให้อยู่ใน mode digital I/O ซึ่งใน mode นี้ input buffer จะเป็น TTL

PORT E จะมีลักษณะคือ จะ multiplex กับ analog inputs โดยเมื่อ PORTE ถูก set เป็น analog inputs แล้ว ขาเหล่านี้เมื่อทำการอ่านค่าจะมีค่าเป็น 0 ส่วน TRISE ซึ่งเป็น control register นั้นจะต้อง set ให้เป็น input เมื่อ set ให้อยู่ใน mode analog input

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงบล็อกไดอะแกรมขา RE ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.6 แสดงฟังก์ชันการทำงานของ PORTE

Name	Bit#	Buffer Type	Function
RE0/$\overline{RD}$/AN5	bit0	ST/TTL(1)	I/O port pin or read control input in Parallel Slave Port mode or analog input: $\overline{RD}$ 1 = Idle 0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected)
RE1/$\overline{WR}$/AN6	bit1	ST/TTL(1)	I/O port pin or write control input in Parallel Slave Port mode or analog input: $\overline{WR}$ 1 = Idle 0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected)
RE2/$\overline{CS}$/AN7	bit2	ST/TTL(1)	I/O port pin or chip select control input in Parallel Slave Port mode or analog input: $\overline{CS}$ 1 = Device is not selected 0 = Device is selected

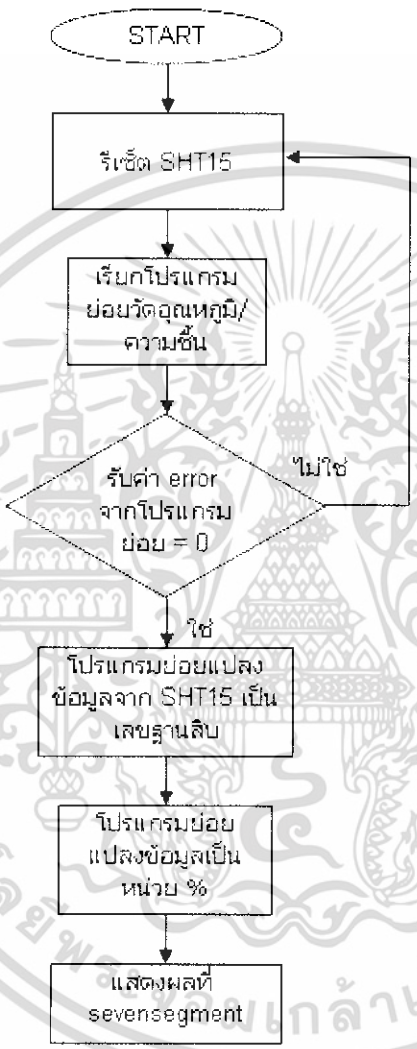
Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 โฟลว์ชาร์จแสดงโปรแกรมควบคุมการวัดอุณหภูมิและความชื้น

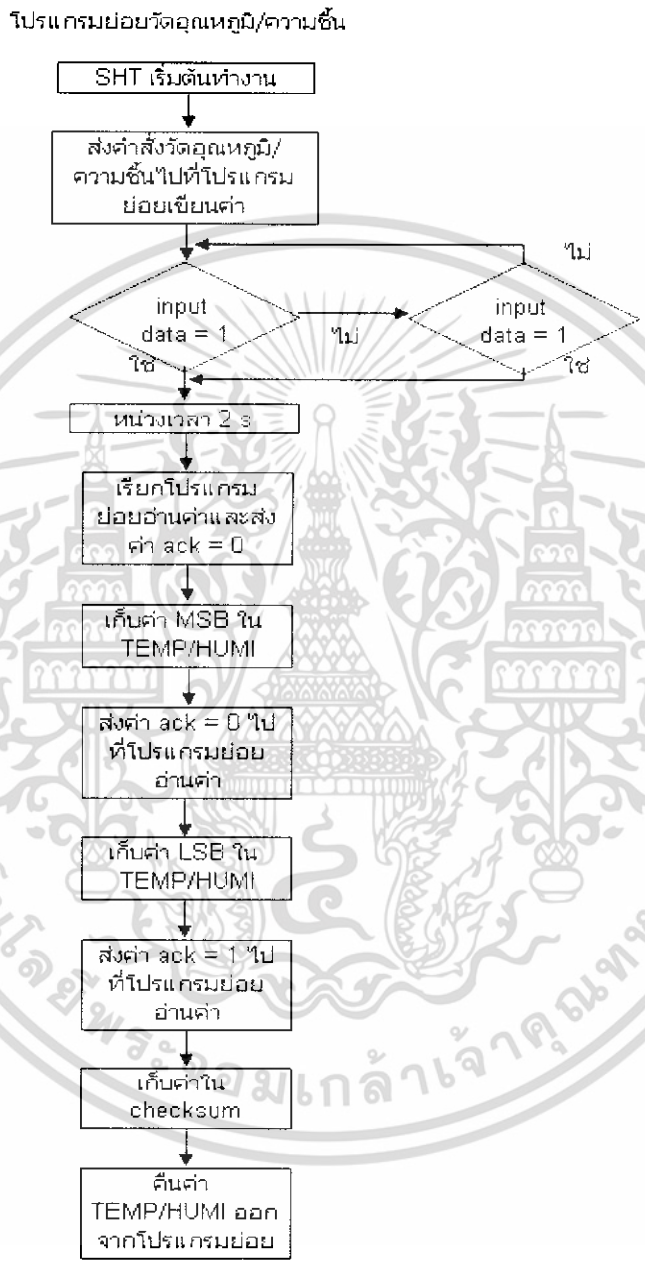
3.5.1 โปรแกรมหลัก



รูปที่ 3.15 แสดงการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

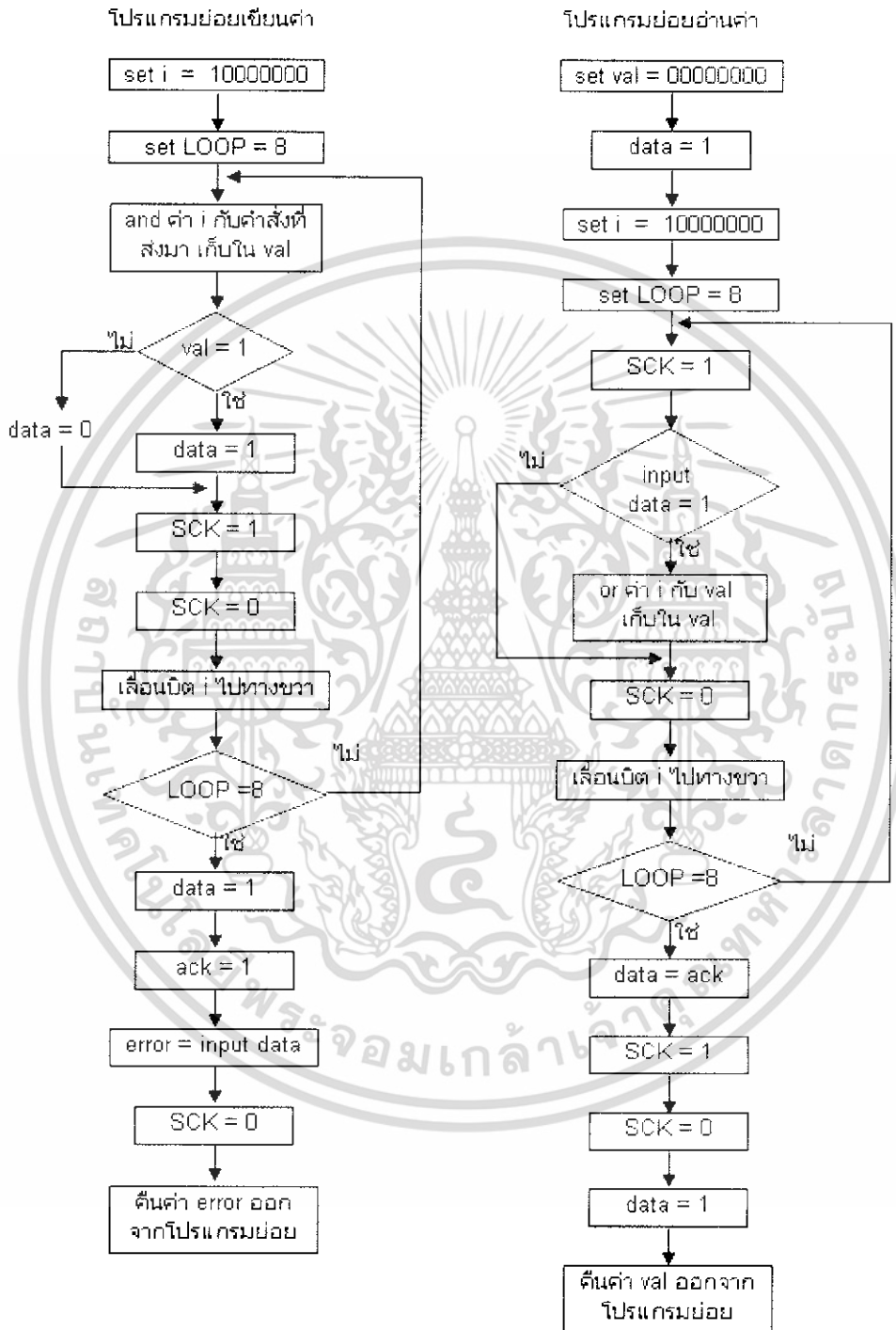
3.5.2 โปรแกรมย่อยวัดอุณหภูมิและความชื้น



รูปที่ 3.16 แสดงการทำงานของโปรแกรมย่อยวัดอุณหภูมิและความชื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

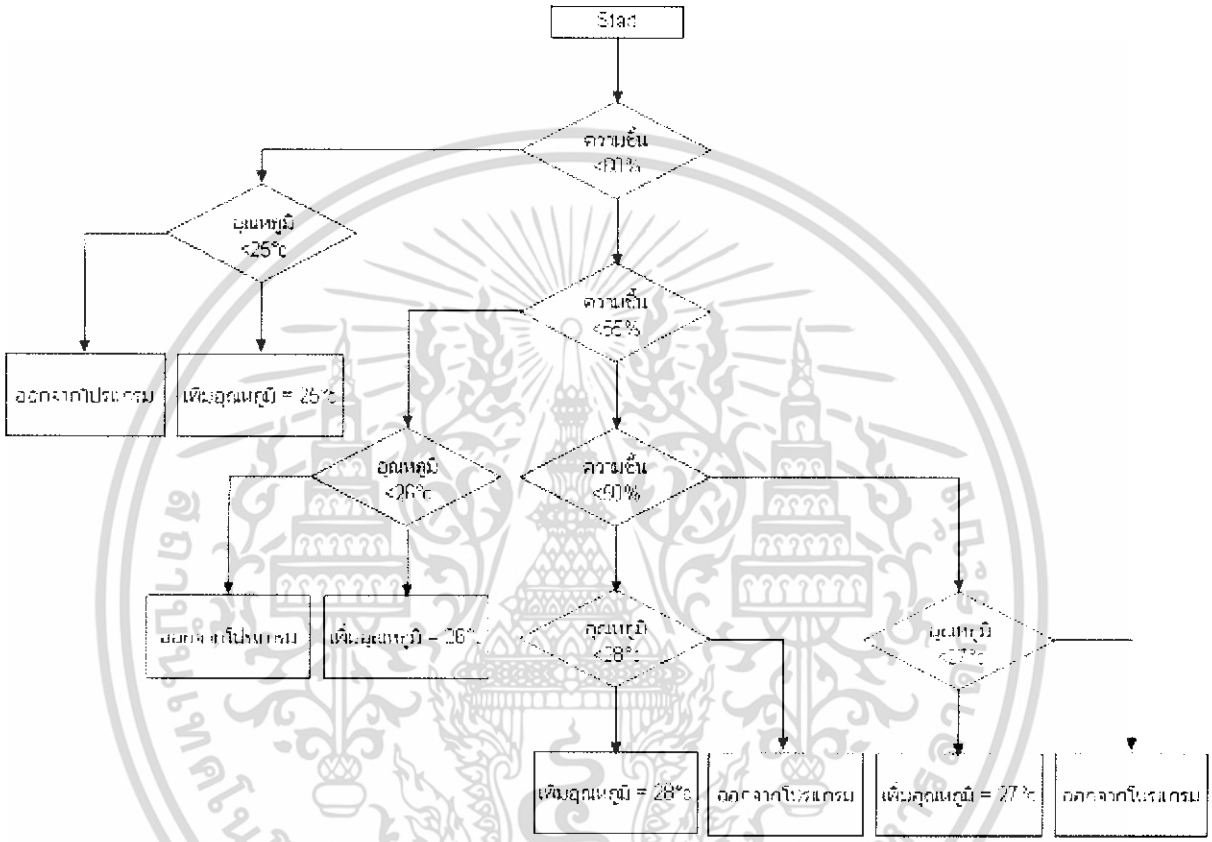
3.5.3 โปรแกรมย่อยเขียนค่าและอ่านค่า



รูปที่ 3.17 แสดงการทำงานของโปรแกรมย่อยเขียนค่าและอ่านค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.4 โปรแกรมปรับอุณหภูมิให้เหมาะสมกับความชื้นสัมพัทธ์

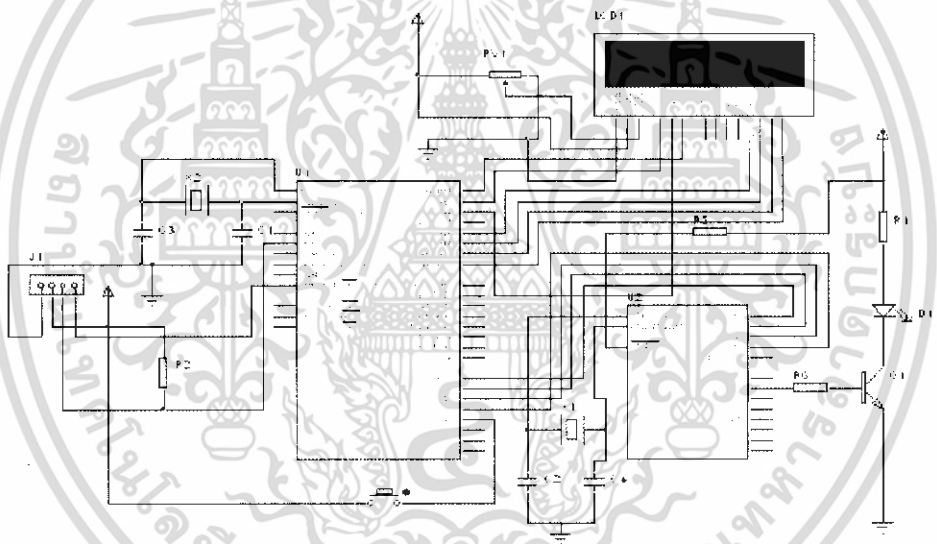


รูปที่ 3.18 แสดงการปรับอุณหภูมิให้เหมาะสมกับความชื้นสัมพัทธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การสื่อสารด้วยอินฟราเรดและภาคแสดงผล

ในส่วนควบคุมอุณหภูมิได้ทดลองกับเครื่องปรับอากาศยี่ห้อ Mitsubishi โดยใช้อุณหภูมิระหว่าง 24 – 28 องศาเซลเซียส ซึ่งเป็นการส่งสัญญาณที่ความถี่ 40 กิโลเฮิร์ท เป็นแบบ PPM (Pulse Position Modulate) เพื่อกำหนดชนิดของข้อมูล โดยสัญญาณมีคาบเวลาประมาณ 26 ไมโครวินาที เมื่อสัญญาณเป็นลอจิก “1” สัญญาณพัลส์จะเป็นลอจิกสูงเวลาประมาณ 13 ไมโครวินาที และเป็นลอจิกต่ำเวลาประมาณ 13 ไมโครวินาทีถ้าสัญญาณเป็นลอจิก “0” สัญญาณพัลส์จะเป็นลอจิกต่ำต่อเนื่องเวลาประมาณ 26 ไมโครวินาที



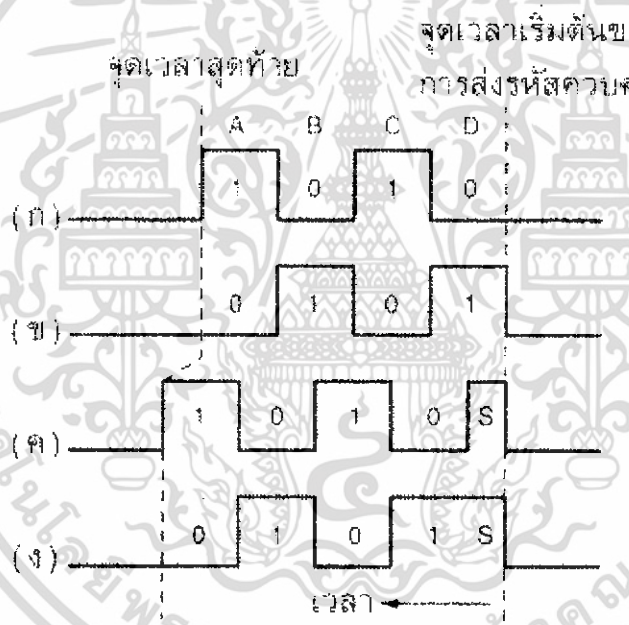
รูปที่ 4.1 แสดงวงจรเครื่องควบคุมอุณหภูมิและวัดความชื้นสัมพัทธ์

4.1 ลักษณะของช่องสัญญาณที่ใช้เป็นรหัสควบคุม

ลักษณะของรหัสที่กำหนดเมื่อถูกส่งจะจัดลำดับเป็นอนุกรมออกไป ดังเช่นรหัส 0101 และ 1010 ในรูปที่ 4.1 (ก) และ 4.1 (ข) ตามลำดับ โดยเรียงความสำคัญจากมากที่สุดไปหาน้อยที่สุด คือจะส่งบิตแรกเป็น D, C, B และ A ข้อสังเกตจากการส่งสัญญาณในรูปที่ 4.1 (ก) และ (ข) แม้ว่าเกิดจากรหัสที่แตกต่างกัน แต่เครื่องรับอาจมีความผิดพลาดในการตีความได้ หากช่วงเวลาในการส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสควบคุม ซึ่งกำหนดด้วยเส้นประที่แสดงจุดเริ่มต้นและจุดสุดท้ายของการส่งข้อมูลเกิดความผิดพลาดไป อาจส่งผลให้เครื่องรับตีความหมายของรูป 4.1 (ก) สลับกับรูป 4.1 (ข) ได้ จึงจำเป็นต้องมีการส่งสัญญาณชิงโครนัส หรือสัญญาณที่ระบุให้ทราบจุดแสดงเวลาเริ่มต้นที่แน่นอนของรหัสสัญญาณควบคุมไปพร้อมกัน โดยทั่วไปมักใช้สัญญาณพัลส์ที่แสดงสภาวะสูงเป็นตัวบอกจุดเริ่มต้นของเวลาในการส่งสัญญาณ ดังแสดงในรูปที่ 4.1 (ค) และ (ง) อันเป็นรูปแบบของรหัสจากรูป 4.1 (ก) และ 4.1 (ข) ตามลำดับ แต่ในการส่งรหัสจากรีโมทของมิตซูบิชิจะใช้สัญญาณเริ่มต้นเป็นสัญญาณพัลส์ที่แสดงสภาวะสูงเป็นเวลา 3400ไมโครวินาทีและใช้สัญญาณพัลส์ที่แสดงสภาวะต่ำเป็นเวลา 1700 ไมโครวินาที



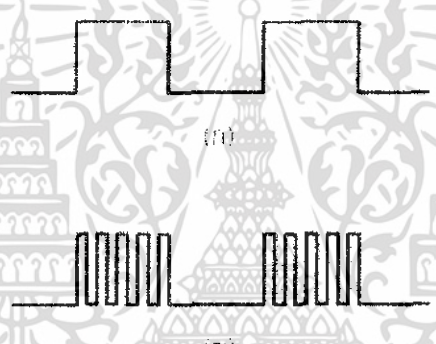
รูปที่ 4.2 แสดงสัญญาณพัลส์ที่ใช้เป็นรหัสควบคุม

4.2 สัญญาณโทนเบิรสต์

สัญญาณโทนเบิรสต์ประกอบด้วยพัลส์ความถี่สูงแบบต่อเนื่องตลอดช่วงความกว้างของบิตที่เป็น “1” ในขณะที่บิตข้อมูลอยู่ในสภาวะต่ำสัญญาณจะคงเดิมไม่มีการเปลี่ยนแปลงแต่อย่างใด ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การส่งข้อมูลของรีโมทจึงต้องสร้างความถี่โทนเบิรสต์ที่เหมาะสมกับวงจรภาครับของเครื่องปรับอากาศ ข้อดีของสัญญาณแบบโทนเบิรสต์คือช่วยป้องกันการรบกวนจากสัญญาณภายนอกได้ดี เช่น การติดตั้งอุปกรณ์ที่รับ - ส่งสัญญาณแบบไร้สายไว้ในพื้นที่ใกล้เคียงกัน ถ้ามีความถี่ของสัญญาณโทนเบิรสต์แตกต่างกันก็จะทำให้ไม่รบกวนซึ่งกันและกัน หรือระบบที่ควบคุมด้วยแสงอินฟราเรดที่ส่งด้วยพัลส์แบบธรรมดา อาจจะมีแสงจากภายนอก ทั้งแสงจากหลอดไฟหรือแสงจากดวงอาทิตย์เข้าไปรบกวนที่เครื่องรับจนทำให้เกิดข้อผิดพลาดได้ แต่ถ้าหากมีการใช้สัญญาณโทนเบิรสต์แล้ว สิ่งรบกวนต่าง ๆ เหล่านี้จะถูกขจัดออกไปโดยสิ้นเชิง

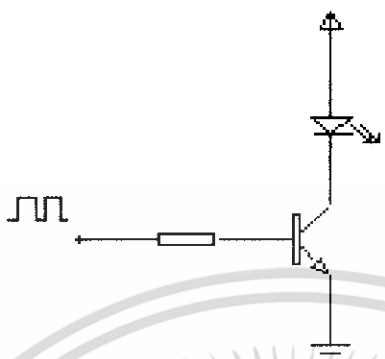


รูปที่ 4.3 (ก) สัญญาณพัลส์ธรรมดา (ข) สัญญาณโทนเบิรสต์

4.3 วงจรจ่ายกำลังหรือวงจรขับเอาท์พุท

สัญญาณที่ได้จะเข้าขาเบสของทรานซิสเตอร์ เพื่อทำให้ทรานซิสเตอร์อยู่ในสภาวะ เปิด - ปิด อุปกรณ์ เพื่อกำหนดให้กระแสไหลผ่าน LED อินฟราเรด โดย LED อินฟราเรด จะทำงานเมื่อสัญญาณควบคุมทางขาเบสของทรานซิสเตอร์เป็น “1” และจะหยุดการทำงานเมื่อสัญญาณควบคุมทางขาเบสของทรานซิสเตอร์เป็น “0” ซึ่งสัญญาณที่ออกมาจาก LED อินฟราเรด จะอยู่ในรูปดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 แสดงวงจรส่งสัญญาณผ่าน LED อินฟราเรด

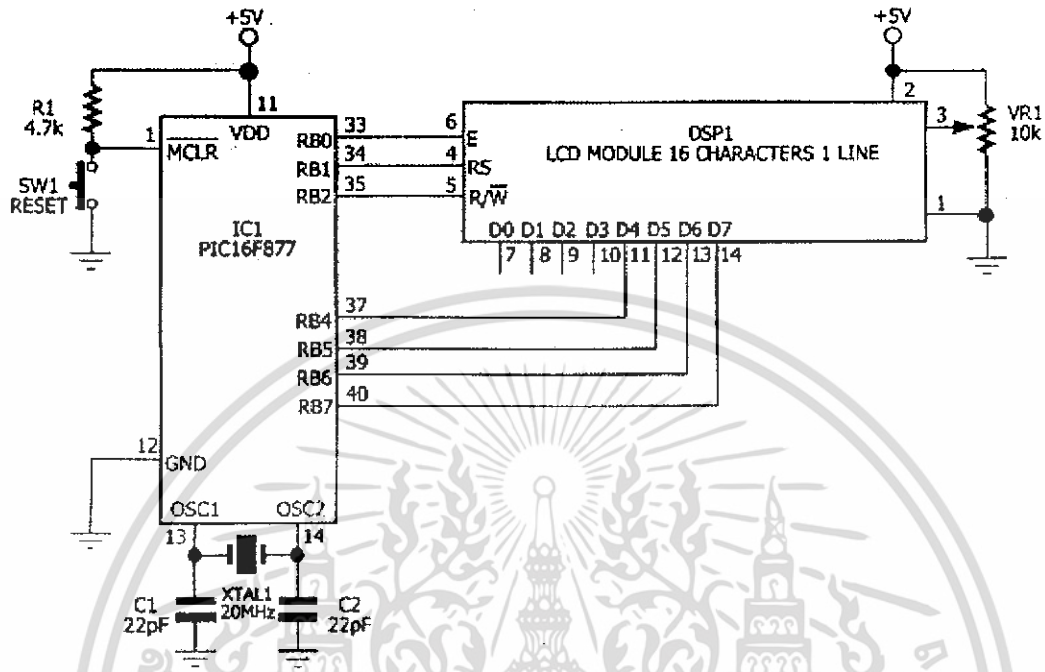
ในวงจรภาคส่งจะใช้สัญญาณโทนเบิรสต์ที่สร้างสัญญาณจากไมโครคอนโทรลเลอร์ PIC16F84A โดยสัญญาณมีคาบเวลาทั้งหมดประมาณ 62.5 มิลลิวินาที ความถี่ 40 กิโลเฮิร์ท ส่งไปที่ขาเบสของวงจรขับเอาท์พุท เพื่อควบคุมให้ LED อินฟราเรด ส่งสัญญาณต่อไปยังวงจรภาครับของเครื่องปรับอากาศ

4.4 ภาคการแสดงผล

ในปัจจุบัน LCD (Liquid Crystal Display module) ได้รับความนิยมในการใช้งานกับไมโครคอนโทรลเลอร์มาก เนื่องจากสามารถแสดงผลได้หลากหลายรูปแบบ นอกจากนี้ยังมีขนาดเล็ก แต่การใช้งานต้องอยู่ในมุมมองที่เหมาะสม

ในส่วนแสดงผลได้ใช้โมดูล LCD ในโหมด 4 บิต แบบตัวอักษรขนาด 16 ตัวอักษร 2 บรรทัด โดยเชื่อมต่อกับไมโครคอนโทรลเลอร์ที่ Port B ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงวงจรเชื่อมต่อระหว่างโมดูล LCD กับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
การทดลองและสรุปผล

5.1 การทดสอบคุณสมบัติของความชื้นสัมพัทธ์ในห้องติดเครื่องปรับอากาศ

5.1.1 การทดลองที่ 1

ทำการทดลองวัดความชื้นสัมพัทธ์ในห้องติดเครื่องปรับอากาศทุก ๆ 1 ชั่วโมง ตั้งแต่เริ่มเปิดเครื่องปรับอากาศ เป็นเวลาทั้งหมด 3 ชั่วโมง ที่อุณหภูมิ 25 องศาเซลเซียส จำนวน 3 ครั้ง

5.1.2 ผลการทดลองที่ 1

ตารางที่ 5.1 แสดงค่าความชื้นสัมพัทธ์ในแต่ละชั่วโมง ณ อุณหภูมิ 25 องศาเซลเซียส ครั้งที่ 1

เวลา(ชั่วโมง)	ความชื้นสัมพัทธ์(%)
0	61.7
1	53.7
2	52.6
3	50.6

ตารางที่ 5.2 แสดงค่าความชื้นสัมพัทธ์ในแต่ละชั่วโมง ณ อุณหภูมิ 25 องศาเซลเซียส ครั้งที่ 2

เวลา(ชั่วโมง)	ความชื้นสัมพัทธ์(%)
0	60.8
1	54.2
2	53.5
3	51.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3 แสดงค่าความชื้นสัมพัทธ์ในแต่ละชั่วโมง ณ อุณหภูมิ 25 องศาเซลเซียส ครั้งที่ 3

เวลา(ชั่วโมง)	ความชื้นสัมพัทธ์(%)
0	59.7
1	51.6
2	48.6
3	49.3

5.1.3 สรุปผลการทดลองที่ 1

จากการทดลองแสดงให้เห็นว่า เมื่อเริ่มเปิดเครื่องปรับอากาศ ความชื้นสัมพัทธ์มีค่าลดลง แสดงว่าไอน้ำในอากาศถูกดูดออกไป โดยพัดลมดูดอากาศ

5.2 การทดสอบประสิทธิภาพของเซ็นเซอร์

5.2.1 การทดลองที่ 2

ทำการวัดอุณหภูมิ เปรียบเทียบระหว่างเทอร์โมมิเตอร์กับเซ็นเซอร์ SHT15 เริ่มตั้งแต่อุณหภูมิ 22 องศาเซลเซียส จนถึง 32 องศาเซลเซียส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2.2 ผลการทดลองที่ 2

ตารางที่ 5.4 แสดงค่าอุณหภูมิเปรียบเทียบระหว่างการวัดด้วยเทอร์โมมิเตอร์กับเซ็นเซอร์ SHT15

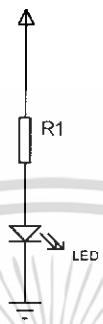
วัดอุณหภูมิจากเทอร์โมมิเตอร์ (˚C)	วัดอุณหภูมิจากเซนเซอร์ SHT15 (˚C)
22	22.0
23	23.2
24	24..1
25	25.2
26	26.2
27	27.3
28	28.5
29	29.4
30	30.5
31	31.6
32	32.5

5.2.3 สรุปผลการทดลองที่ 2

จากการทดลอง ผลจากการวัดอุณหภูมิทั้งสองแบบแสดงให้เห็นว่า การวัดอุณหภูมิด้วยเซ็นเซอร์ SHT15 มีความคลาดเคลื่อนจากการวัดด้วยเทอร์โมมิเตอร์เล็กน้อย โดยมีแนวโน้มที่จะมีความคลาดเคลื่อนมากขึ้น เมื่ออุณหภูมิสูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 การทดสอบการส่งสัญญาณด้วยอินฟราเรด



รูปที่ 5.1 แสดงวงจรขับ LED อินฟราเรด

5.3.1 การทดลองที่ 3

ทำการทดลองโดยต่อวงจรตามรูปที่ 5.1 เมื่อให้แรงดัน 5 โวลท์ ปรับค่าความต้านทานให้กระแสที่ไหลผ่าน LED อินฟราเรดเปลี่ยน เพื่อทดสอบความสามารถในการส่งสัญญาณด้วยอินฟราเรด โดยปรับค่าความต้านทานดังตารางที่ 5.5

5.3.2 ผลการทดลองที่ 3

ตารางที่ 5.5 แสดงระยะที่สามารถส่งสัญญาณได้จากการเปลี่ยนค่าความต้านทาน

ความต้านทาน	แรงดันคร่อม R(V)	กระแส(mA)	ระยะทางที่ส่งได้(m)
330	4	12.12	3
500	4	8	2.8
830	4	4.82	2.1
1k	4	4	1.3

5.3.3 สรุปผลการทดลองที่ 3

จากการทดลองส่งสัญญาณผ่าน LED อินฟราเรด เมื่อมีกระแสไหลผ่าน LED อินฟราเรดมากขึ้น จะทำให้สามารถส่งสัญญาณได้ระยะทางไกลมากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ประภากร สุวรรณะ และ สมศักดิ์ ชุมช่วย, " วิศวกรรมอิเล็กทรอนิกส์ 1 ", คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 456 หน้า, 2545

Sensirion, " SHT1x/SHT7x Relative Humidity & Temperature Sensor System ", 2002.

Microchip Technology, " PIC16F87XA Data Sheet ", 2001.

Microchip Technology, " PIC16F8X Data Sheet ", 2001.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 1: #include <16F877.h>
 2: //#include <math.h>
 3: //#include <stdio.h>
 4: #include <string.h>
 5: #define txd pin_c6
 6: #define rxd pin_c7
 7: #define CLOCK_SP 20000000
 8: #fuses HS
 9: #fuses NOPROTECT,nowdt,put
10: #use delay(clock=CLOCK_SP)
11: typedef union
12: { unsigned int i;
13: float f;
14: } value;
15: enum {TEMP,HUMI,one,two};
16: #define t25 pin_d0
17: #define t26 pin_d1
18: #define t27 pin_d2
19: #define t28 pin_d3
20: #define match pin_d4
21: #define no_match pin_d5
22: #define sht_noACK 0
23: #define sht_ACK 1
24: #define sht_STATUS_REG_W 0x06 //000 0011 0
25: #define sht_STATUS_REG_R 0x07 //000 0011 1
26: #define sht_MEASURE_TEMP 0x03 //000 0001 1
27: #define sht_MEASURE_HUMI 0x05 //000 0010 1
28: #define sht_RESET 0x1e //000 1111 0
29: #define sht11_data pin_A1
30: #define sht11_sck pin_A5
31: /*#define t23 pin_d0
32: #define t24 pin_d1
33: #define t25 pin_d2
34: #define t26 pin_d3
35: #define t27 pin_d4
36: #define t28 pin_d5*/
37: //#byte port_b=6
38: #define  use_portb_lcd
39: #include "lcd.c"
40: int ple = 0;
41: int count = 0;
42: //-----------------------------------------------------------------------
43: void lcd_show(float rh,float tc)
44: //-----------------------------------------------------------------------
45: {
46: lcd_send_byte(0,0x80);
47: printf(lcd_putc,"    humi:%5.2f",rh);
48: delay_ms(500);
49: lcd_send_byte(0,0xc0);
50: printf(lcd_putc,"    temp:%5.2f",tc);
51: delay_ms(500);
52:
53: }
54: //-----------------------------------------------------------------------
55: void lcd_show_off(float rh,float tc)
56: //-----------------------------------------------------------------------
57: {
58: lcd_send_byte(0,0x80);
59: printf(lcd_putc,"off humi:%5.2f",rh);
60: delay_ms(500);
61: lcd_send_byte(0,0xc0);
62: printf(lcd_putc,"    temp:%5.2f",tc);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 63: delay_ms(500);
 64: }
 65: void lcd_show25(float rh,float tc)
 66: //-------------------------------------------------------------------------
 67: {
 68: lcd_send_byte(0,0x80);
 69: printf(lcd_putc," 25 humi:%5.2f",rh);
 70: delay_ms(500);
 71: lcd_send_byte(0,0xc0);
 72: printf(lcd_putc,"    temp:%5.2f",tc);
 73: delay_ms(500);
 74: }
 75: void lcd_show26(float rh,float tc)
 76: //-------------------------------------------------------------------------
 77: {
 78: lcd_send_byte(0,0x80);
 79: printf(lcd_putc," 26 humi:%5.2f",rh);
 80: delay_ms(500);
 81: lcd_send_byte(0,0xc0);
 82: printf(lcd_putc,"    temp:%5.2f",tc);
 83: delay_ms(500);
 84: }
 85: void lcd_show27(float rh,float tc)
 86: //-------------------------------------------------------------------------
 87: {
 88: lcd_send_byte(0,0x80);
 89: printf(lcd_putc," 27 humi:%5.2f ",rh);
 90: delay_ms(500);
 91: lcd_send_byte(0,0xc0);
 92: printf(lcd_putc,"    temp:%5.2f ",tc);
 93: delay_ms(500);
 94: }
 95: void lcd_show28(float rh,float tc)
 96: //-------------------------------------------------------------------------
 97: {
 98: lcd_send_byte(0,0x80);
 99: printf(lcd_putc," 28 humi:%5.2f",rh);
100: delay_ms(500);
101: lcd_send_byte(0,0xc0);
102: printf(lcd_putc,"    temp:%5.2f",tc);
103: delay_ms(500);
104: }
105: //-------------------------------------------------------------------------
106: char compare_temp(float rh_true,float t_C)
107: //-------------------------------------------------------------------------
108: {
109: char pa = 0x80;
110: float rh;
111: float tc;
112: rh = rh_true;
113: tc = t_C;
114:
115: if(rh_true<60){
116:    if(rh_true<55){
117:       if(rh_true<50){
118:       if(t_C<28){//temperature no match at 28
119:       //output_low(match);
120:       output_high(no_match);
121:       lcd_show28(rh,tc);
122:       output_high(t28);
123:       delay_ms(1300);
124:       output_low(t28);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
125:        return pa;}
126:        else{ // temperature match at 28
127:        if(t_C>29){
128:        output_high(no_match);
129:        lcd_show28(rh,tc);
130:        output_high(t28);
131:        delay_ms(1300);
132:        output_low(t28);
133:        return pa;}
134:        else{
135:        output_low(no_match);
136:        return pa;}}}
137:     if(t_C<27){ // temparature no match at 27
138:     output_high(no_match);
139:     lcd_show27(rh,tc);
140:     output_high(t27);
141:     delay_ms(1300);
142:     output_low(t27);
143:     return pa;}
144:     else{ //temperature match at 27
145:     if(t_C>28){
146:     output_high(no_match);
147:     lcd_show27(rh,tc);
148:     output_high(t27);
149:     delay_ms(1300);
150:     output_low(t27);
151:     return pa;}
152:     else{
153:     output_low(no_match);
154:     //output_high(match);
155:     return pa;}}}
156: if(t_C<26){ //temperature no match at 26
157: //output_low(match);
158: output_high(no_match);
159: lcd_show26(rh,tc);
160: output_high(t26);
161: delay_ms(1300);
162: output_low(t26);
163: return pa;}
164: else{ //temperature match at 26
165: if(t_C>27){
166: output_high(no_match);
167: lcd_show26(rh,tc);
168: output_high(t26);
169: delay_ms(1300);
170: output_low(t26);
171: return pa;}
172: else{
173: output_low(no_match);
174: return pa;}}}
175:    else{
176:       if(t_C<26){//temperature match at 25
177:          if(t_C>24){
178:          output_low(no_match);
179:          //output_high(match);
180:          return pa;}
181:          else{
182:          output_high(no_match);
183:          lcd_show25(rh,tc);
184:          output_high(t25);
185:          delay_ms(1300);
186:          output_low(t25);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
187:          return pa;}}
188:       else{//temperature no match at 25
189:          //output_low(match);
190:          output_high(no_match);
191:          lcd_show25(rh,tc);
192:          output_high(t25);
193:          delay_ms(1300);
194:          output_low(t25);
195:       return pa;}}
196: }
197:
198: //-----------------------------------------------------------------------
199: char sht11_write_byte(unsigned char value)
200: //-----------------------------------------------------------------------
201: {
202: unsigned char i,error=0;
203: for (i=0x80;i>0;i/=2) //shift bit for masking
204: {
205: if (i & value)
206: output_high(sht11_data); //masking value with i , write to SENSI-BUS
207: else
208: output_low(sht11_data);
209: output_high(sht11_sck); //clk for SENSI-BUS
210: delay_us(5); //pulswith approx. 5 us
211: output_low(sht11_sck);
212: delay_us( 5); //pulswith approx. 5 us
213: }
214: output_high(sht11_data); //release DATA-line//////////////////////////////
215: output_high(sht11_sck); //clk #9 for ack
216: delay_us(5); //pulswith approx. 5 us
217: error=input(sht11_data); //check ack (DATA will be pulled down by SHT11)
218: output_low(sht11_sck);
219: return error; //error=1 in case of no acknowledge
220: }
221: //-----------------------------------------------------------------------
222: char sht11_read_byte(unsigned char ack)
223: //-----------------------------------------------------------------------
224: // reads a byte form the Sensibus and gives an acknowledge in case of "ack=
225: {
226: unsigned char i,val=0x00;
227: output_high(sht11_data); //release DATA-line
228: for (i=0x80;i>0;i/=2) //shift bit for masking
229: {
230: output_high(sht11_sck); //clk for SENSI-BUS
231: delay_us( 5); //pulswith approx. 5 us
232: if (input(sht11_data)==1)
233: val=(val | i); //read bit
234: output_low(sht11_sck);
235: }
236: output_bit(sht11_data,!ack); //in case of "ack==1" pull down DATA-Line
237: output_high(sht11_sck); //clk #9 for ack
238: delay_us( 5); //pulswith approx. 5 us
239: output_low(sht11_sck);
240: output_high(sht11_data); //release DATA-line
241: return val;
242: }
243: //-----------------------------------------------------------------------
244: void sht11_transstart(void)
245: //-----------------------------------------------------------------------
246: // generates a transmission start
247: //       _____         ____
248: // DATA:      |_______|
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
249: // ___ ___
250: // SCK : ___| |___| |______
251: {
252: output_high(sht11_data);
253: output_low(sht11_sck); //Initial state
254: delay_us( 1);
255: output_high(sht11_sck);
256: delay_us( 1);
257: output_low(sht11_data);
258: delay_us( 1);
259: output_low(sht11_sck);
260: delay_us( 3);
261: output_high(sht11_sck);
262: delay_us( 1);
263: output_high(sht11_data);
264: delay_us( 1);
265: output_low(sht11_sck);
266: }
267: //----------------------------------------------------------------------------
268: void sht11_connectionreset(void)
269: //----------------------------------------------------------------------------
270: // communication reset: DATA-line=1 and at least 9 SCK cycles
271: // followed by transstart
272: //
273: // DATA: |_______|
274: //
275: // SCK : __| |__| |__| |__| |__| |__| |__| |__| |__| |______| |___| |_____
276: {
277: unsigned char i;
278: output_high(sht11_data);
279: output_low(sht11_sck); //Initial state
280: for(i=0;i<9;i++) //9 SCK cycles
281: {
282: output_high(sht11_sck);
283: output_low(sht11_sck);
284: }
285: sht11_transstart(); //transmission start
286: }
287: //----------------------------------------------------------------------------
288: void sht11_calc(unsigned int16 bath,unsigned int16 dolla)
289: //----------------------------------------------------------------------------
290: {
291: char pik;
292: int16 rh,t,m1,m2,n1,n2,j;
293: int   rh_true;
294: float rh_lin,t_C;
295: rh_lin=0;
296: t_C=0;
297: rh=bath; // rh: Humidity [Ticks] 12 Bit
298: t=dolla; // t: Temperature [Ticks] 14 Bit
299: t_C=(t*0.01) - 40; //calc. temperature from ticks to [?C]
300: rh_lin=0.0405*rh - 0.0000028*rh*rh - 4; //calc. humidity from ticks to [%RH
301: //rh_true=(t_C-25)*(T1+T2*rh)-rh_lin; //calc. temperature compensated humid
302: if(rh_lin>100)rh_lin=99; //cut if the value is outside of
303: if(rh_lin<0.1)rh_lin=0; //the physical possible range
304: //----------------------------------------------------------------------------
305: if(ple == 1){
306:
307: if(++count == 9){
308: count = 0;
309: pik=compare_temp(rh_lin,t_C);
310: }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
311: lcd_show(rh_lin,t_C);
312: }
313: else
314: lcd_show_off(rh_lin,t_C);
315: }
316: //-----------------------------------------------------------------------
317: char sht11_measure(unsigned char mode,less,unsigned int16 *price)
318: //-----------------------------------------------------------------------
319: // makes a measurement (humidity/temperature) with checksum
320: //lcd_show(t1);
321: {
322: unsigned char p_value,p_val,p_checksum;
323: unsigned error=0;
324: unsigned int i;
325: char a,b,c;
326: unsigned int16 h1;
327:
328: sht11_transstart(); //transmission start
329: switch(mode)
330: { //send command to sensor
331: case TEMP : error+=sht11_write_byte(sht_MEASURE_TEMP); break;
332: case HUMI : error+=sht11_write_byte(sht_MEASURE_HUMI); break;
333: default : break;
334: }
335:
336: for (i=0;i<65535;i++)
337: if(input(sht11_data)==0)
338: break; //wait until sensor has finished the measurement
339: if(input(sht11_data)==1)
340: error+=1;
341:
342: if(error==0)
343: {
344: p_value =sht11_read_byte(sht_ACK); //read the first byte (MSB)
345: p_val=sht11_read_byte(sht_ACK); //read the second byte (LSB)
346: p_checksum =sht11_read_byte(sht_noACK); //read checksum
347: switch(less)
348: {case one:
349: h1=1;
350: c=0x08;
351: if(c & p_value)
352: {for (a=0x04;a>0;a/=2) //shift bit for masking
353: {if(a & p_value)
354: h1=(2*h1)+1;
355: else
356: h1=2*h1;}}
357: else
358: {for (a=0x02;a>0;a/=2) //shift bit for masking
359: {if(a & p_value)
360: h1=(2*h1)+1;
361: else
362: h1=2*h1;}}
363: break;
364: case two:
365: h1=1;
366: for (a=0x08;a>0;a/=2) //shift bit for masking
367: {if(a & p_value)
368: h1=(2*h1)+1;
369: else
370: h1=2*h1;
371: }
372: break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
373: default:break;}
374: for (b=0x80;b>0;b/=2) //shift bit for masking
375: {
376: if (b & p_val)
377: h1=2*h1+1;
378: else
379: h1=2*h1;
380: }
381:
382: *price=h1;
383: //lcd_show(*price);
384: return error;
385: }
386: else
387: return error;
388: }
389:
390: //--------------------------------------------------------------------------
391: void main()
392: //--------------------------------------------------------------------------
393: {
394: value humi_val,temp_val;
395: float value;
396:   unsigned char error,checksum;
397:   unsigned int16 i,dolla,bath;
398:   int k,g;
399:   float x;
400: lcd_init();
401:   output_low(match);
402:   //output_low(pin_d1);
403: sht11_connectionreset();
404:
405: while(true)
406:   {
407:   bath=0;
408:   dolla=0;
409:   //lcd_show(x);
410:     output_low(match);
411:   if(input(match))
412:   ple = 1;
413:
414: error=0;
415: error+=sht11_measure(HUMI,one,&bath);  //measure humidity
416:     if(error!=0)
417:     //lcd_show(x,x);
418:     sht11_connectionreset();
419:
420: error=0;
421: error+=sht11_measure(TEMP,two,&dolla); //measure temperature
422:     if(error!=0)
423:     //lcd_show(x,x);
424:
425:     sht11_connectionreset();
426:
427:
428:       //lcd_show(bath,dolla);
429:       sht11_calc(bath,dolla);
430: /*
431: if(input(t28))
432: lcd_show28(rh,tc);
433: if(input(t27))
434: lcd_show27(rh,tc);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
435: if(input(t26))
436: lcd_show26(rh,tc);
437: if(input(t25))
438: lcd_show25(rh,tc);
439: */
440:    }
441:    }
442:
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 1:                          /* TEMPERATURE SIGNAL */
 2:
 3: #include <16F84a.h>
 4: #define clock_sp 4000000
 5: #fuses hs
 6: #fuses noprotect,nowdt,put
 7: #define over pin_b0
 8: #use delay(clock = clock_sp)
 9: #define xxx pin_a3 //28
10: #define yyy pin_a2 //27
11: #define zzz pin_a1 //26
12: #define aaa pin_a0 //25
13: #define bbb pin_a4 //24
14: int k,r;
15: int l,s ;
16: int high;
17: //---------------------------------------------------------------------
18: void begin()
19: //---------------------------------------------------------------------
20: {
21: for (l=0; l<131; l++){
22: output_high(over);
23: delay_us(9);
24: output_low(over);
25: delay_us(2);}
26: output_low(over);
27: delay_us(1650);
28: }
29: //---------------------------------------------------------------------
30: void data(int j)
31: //---------------------------------------------------------------------
32: {
33: int i,x;
34: for(i=0;i<j;i++){
35: for(x=0;x<17;x++){
36: output_high(over);
37: delay_us(9);
38: output_low(over);
39: delay_us(3);}
40: output_low(over);
41: delay_us(395);
42: }
43: output_low(over);
44: delay_us(790);
45: }
46: //---------------------------------------------------------------------
47: void  temp28()
48: //---------------------------------------------------------------------
49: {
50: int t28[25] = {1,1,4,3,1,2,3,1,2,1,3,3,18,3,3,1,7,1,47,1,1,1,1,1,3};
51: k=0;
52: begin();
53: for(r=0;r<25;r++){
54: high = t28[k];
55: data(high);
56: k-+;}
57: }
58: //---------------------------------------------------------------------
59: void  temp27()
60: //---------------------------------------------------------------------
61: {
62: int t27[19] = {1,1,4,3,1,2,3,1,2,1,3,3,18,3,3,1,9,52,2};
```

เอกสารนี้เป็นการที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 63: k=0;
 64: begin();
 65: for(r=0;r<19;r++){
 66: high = t27[k];
 67: data(high);
 68: k++;}
 69: }
 70: //-------------------------------------------------------------------
 71: void  temp26()
 72: //-------------------------------------------------------------------
 73: {
 74: int t26[21] = {1,1,4,3,1,2,3,1,2,1,3,3,18,3,3,1,7,2,46,6,2};
 75: k=0;
 76: begin();
 77: for(r=0;r<21;r++){
 78: high = t26[k];
 79: data(high);
 80: k++;}
 81: }
 82: //-------------------------------------------------------------------
 83: void  temp25()
 84: //-------------------------------------------------------------------
 85: {
 86: int t25[21] = {1,1,4,3,1,2,3,1,2,1,3,3,18,3,3,1,8,1,47,5,2};
 87: k=0;
 88: begin();
 89: for(r=0;r<21;r++){
 90: high = t25[k];
 91: data(high);
 92: k++;}
 93: }
 94: //-------------------------------------------------------------------
 95: void option(int pic)
 96: //-------------------------------------------------------------------
 97: {
 98: if(pic == 25)
 99: temp25();
100: if(pic == 26)
101: temp26();
102: if(pic == 27)
103: temp27();
104: if(pic == 28)
105: temp28();
106: }
107: //-------------------------------------------------------------------
108: void main(){
109: //-------------------------------------------------------------------
110: int pik;
111: while(true){
112: pik = 0;
113:
114: if(input(xxx))
115: pik = 28;
116:
117: if(input(yyy))
118: pik = 27;
119:
120: if(input(zzz))
121: pik = 26;
122:
123: if(input(aaa))
124: pik = 25;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

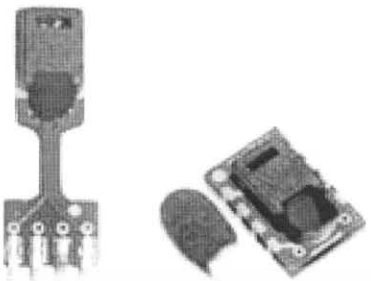
```
125:
126: option(pik);
127: };
128: //-------------------------------------------------------------
129:
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SENSIRION
THE SENSOR COMPANY

SHT1x / SHT7x

Humidity & Temperature Sensmitter



_ Relative humidity and temperature sensors
_ Dew point
_ Fully calibrated, digital output
_ No external components required
_ Ultra low power consumption
_ Surface mountable or 4-pin fully interchangeable
_ Excellent long-term stability
_ Small size
_ Automatic power down

Preliminary Information June 2002

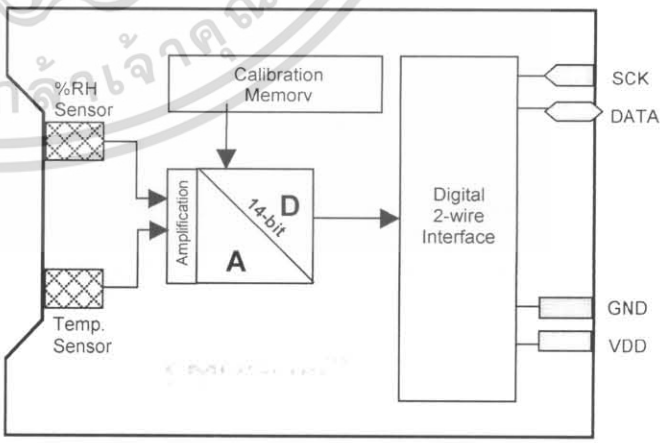
SHT1x Product Summary

The SHT1x / SHT7x is a single chip relative humidity and temperature multi sensor module comprising a calibrated digital output. Application of industrial CMOS processes with customized post processing (CMOSens® technology) ensures highest reliability and excellent long term stability. The device includes two calibrated microsensors for relative humidity and temperature which are seamlessly coupled to a 14bit analog to digital converter and a serial interface circuit on the same chip. This results in superior signal quality, a fast response time and insensitivity to external disturbances (EMC) at a very competitive price.
Each sensor is calibrated in a precision humidity chamber and the calibration coefficients are programmed into the OTP memory. These coefficients are used internally during measurements to calibrate the signals from the sensors.
The 2-wire serial interface and internal voltage regulation allows easy and fast system integration. Its tiny size and low power consumption makes it the ultimate choice for even the most demanding applications including automotive, instrumentation, medical equipment, heating, ventilation and air conditioning systems (HVAC), portable consumer electronics and battery-operated controllers.
The device is supplied in either a surface-mountable LCC (SHT1x) or as a 4-pin single-in-line type package (SHT7x). Customer specific packaging options may be available on request.

Applications

_ Consumer Appliances
_ Automotive
_ HVAC
_ Weather stations
_ Test & Measurement
_ Data Logging
_ Automation
_ White Goods

Schematic Diagram



Ordering information

Part Number	Humidity accuracy	Temperature accuracy	Package	
SHT11	±3.5%RH	±0.5°C@25°C	SMT	
SHT15	±2.0%RH	±0.5°C	SMT	
SHT71	±3.5%RH	±0.5°C@25°C	4-pin single-in-line	Available Q3 2002
SHT75	±2.0%RH	±0.5°C	4-pin single-in-line	Available Q3 2002

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 Sensor Performance Specifications[1]

Parameter	Conditions	Min.	Typ.	Max.	Units
Humidity					
Resolution		0.5	0.03	0.03	% RH
		8	12	12	bit
Repeatability			±0.1		% RH
Accuracy[2] & Interchangeability		see figure 1			
Nonlinearity	10 - 90 %RH	<1[5]		±3[5]	% RH
Range		0		100	% RH
Response time	1/e (63%) slowly moving air		4		s
Hysteresis			±1		% RH
Long term stability	Typical		< 1		% RH/yr
Temperature					
Resolution		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		see figure 1			
Range		-40		123.8	°C
		-40		254.9	°F
Response Time	1/e (63%)	5		30	s

Table 1 Sensor Performance Specifications

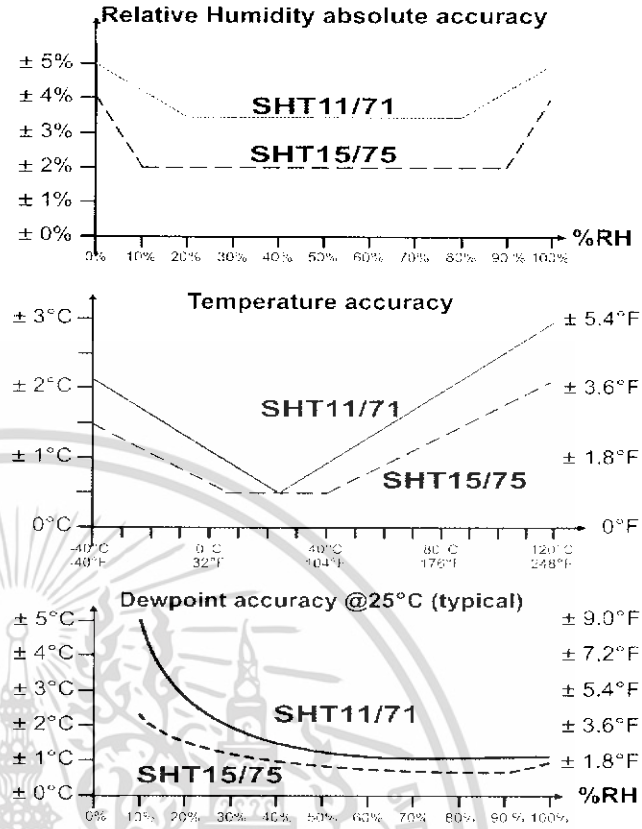


Figure 1 Rel. Humidity, Temperature and Dewpoint accuracies

1.1 Converting the digital output to physical values

1.1.1 Humidity

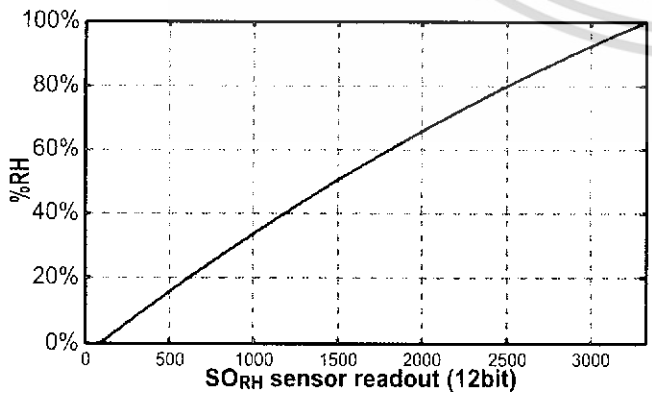
To compensate for the non-linearity of the humidity sensor and to obtain the full accuracy it is recommended to convert the readout with the following formula:

$$RH_{linear} = c_1 + c_2 \bullet SO_{RH} + c_3 \bullet SO_{RH}^2$$

$c_1 = -4$ $c_2 = 0.0405$ $c_3 = -2.8 * 10^{-6}$ for 12bit SO_{RH}

$c_1 = -4$ $c_2 = 0.648$ $c_3 = -7.2 * 10^{-4}$ for 8bit SO_{RH}

For simplified, less computation intense conversion formulas see application note "RH Non-Linearity Compensation".



For temperatures significantly different from 25°C (~77°F) the temperature coefficient of the RH sensor should be considered:

$$RH_{true} = (T_{°C} - 25) \bullet (t_1 + t_2 \bullet SO_{RH}) + RH_{linear}$$

with $t_1 = 0.01$; $t_2 = 0.00008$ for 12bit SO_{RH} ; $t_2 = 0.00128$ for 8bit SO_{RH}

This equals ~0.12%RH / °C @ 50%RH

1.1.2 Temperature

The temperature sensor is very linear by design. Use the following formula to convert from digital readout to temperature: Temperature = $d_1 + d_2 * SO_T$

Use the appropriate table entries for 5V or 3V.

	Celsius		Fahrenheit	
SO_T	d_1	d_2	d_1	d_2
14bit 5V	-40	0.01	-40	0.018
12bit 5V	-40	0.04	-40	0.072
14bit 3V	-38.4	0.0098	-37.1	0.0176
12bit 3V	-38.4	0.0392	-37.1	0.0704

This equals a voltage dependency of ~ -0.2°C/V @ 25°C

1.1.3 Dewpoint

See application note "Dewpoint calculation" for more information.

[1] For operation within normal operation range as described in Chapter 3, RH accuracy at 25°C

[2] Not including non-linearity

[3] The default measurement resolution of 14bit (temperature) and 12bit (humidity) can be reduced to 12 and 8 bit through the status register.

[4] Where SO_{RH} is the sensor output for relative humidity

[5] Min. value after compensation with formula in chapter 1.1.1, Max. value without any compensation

เอกสารนี้จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 Serial Interface

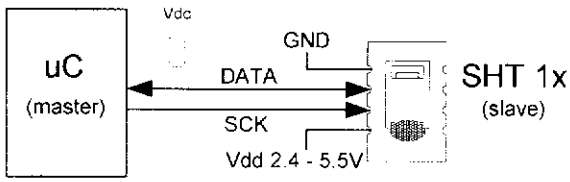


Figure 2 Typical application circuit

2.1 Power Pins

The device requires a voltage supply between 2.4V and 5.5V. After powerup the device requires 11ms to reach its "sleep" state. No commands should be sent before that time. Power supply pins (VDD, GND) may be decoupled with a 100 nF capacitor.

2.2 I/O Pins (Bidirectional 2-wire Interface)

See Table 5 for a detailed IO characteristics.

2.2.1 Serial clock input (SCK)

The SCK is used to synchronize the communication between a master and the SHT1x/SHT7x. Since the device contains fully static logic there is no minimum SCK frequency.

2.2.2 Serial data (DATA)

The DATA tristate pin is used to transfer data in and out of the device. DATA **changes at the falling edge** and is **valid on the rising edge** of the serial clock SCK. An external pull-up resistor is required to pull the signal high. (See Figure 2). Pull-up resistors are often included in I/O circuits of microcontrollers.

2.2.3 Command sequence

To initiate a transmission a "Transmission Start" sequence has to be issued. It consists of a lowering of the DATA line while SCK is high, followed by a low pulse on SCK and raising DATA again while SCK is still high.

Figure 3 "Transmission Start" sequence

The subsequent command sequence consists of three address bits (only "000" is currently supported) and five command bits. The SHT1x/SHT7x indicates the proper reception of a command by pulling the DATA pin low (ACK bit) after the falling edge of the 8th SCK clock and the DATA line is released (and goes high) after the falling edge of the 9th SCK clock.
See 2.2.5 "Measurement Sequence" for an application of the command sequence

2.2.4 Connection reset sequence

If communication with the device is lost the following signal sequence will reset its serial interface:
While leaving DATA high toggle SCK 9 or more times. This must be followed by a "Transmission Start" sequence preceding the next command. This sequence resets the interface only. The status register preserves its contend.

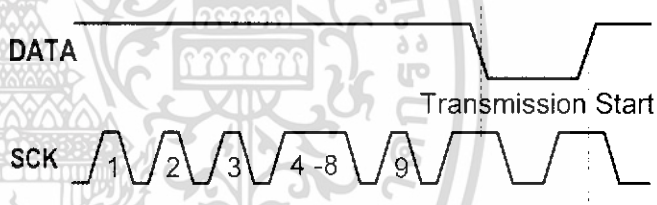
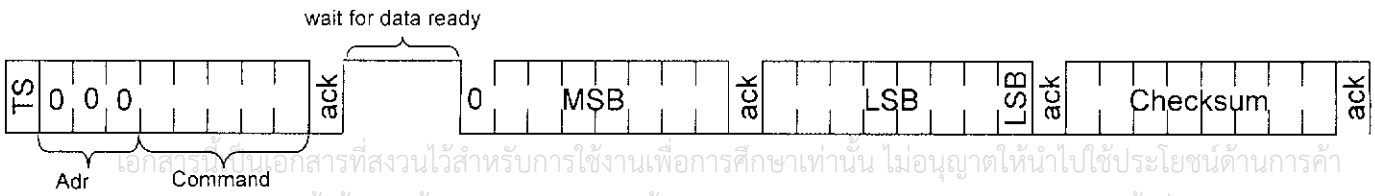


Figure 4 Connection reset sequence

Command	Code	Description
Reserved	0000x	Reserved
Measure Temperature	**00011**	**Temperature measurement**
Measure Humidity	**00101**	**Humidity measurement**
Status Register Read	00111	Read access to the status register (see application note)
Status Register Write	00110	Write access to the status register (see application note)
Reserved	0101x-1110x	Reserved
Soft reset	**11110**	**resets the chip, clears the status register to default values** wait 11ms before next command

Table 2 SHT1x/SHT7x list of commands



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 *Measurement sequence (T and RH)*

After issuing a measurement command ('00000101' for RH, '00000011' for Temperature) the controller has to wait for the measurement to complete. This takes approximately 11/55/210ms for a 8/12/14bit measurement. The exact time varies by up to ±15% with the speed of the internal oscillator. To signal the completion of a measurement, the SHT1x pulls down the data line (2) and the controller must restart SCK.
Two bytes of measurement data and one byte of CRC checksum will then be transmitted. The uC must acknowledge each byte by pulling the DATA line low. All values are MSB first, right justified. (e.g. the 5th SCK is MSB for a 12bit value, for a 8bit result the first byte is not used).
Communication terminates after the acknowledge bit of the CRC data. If CRC-8 Checksum is not used the controller may terminate the communication after the measurement data LSB by keeping ack high.
The device automatically returns to sleep mode after the measurement and communication have finished.

Warning: To keep heat up of the SHT1x/SHT7x below 0.1°C it should not be active for more than 15% of the time (e.g. max. 3 measurements / second for 12bit accuracy).

2.2.6 *CRC-8 Checksum Calculation*

Please consult application note "CRC-8 Checksum Calculation" for information on how to calculate the CRC.

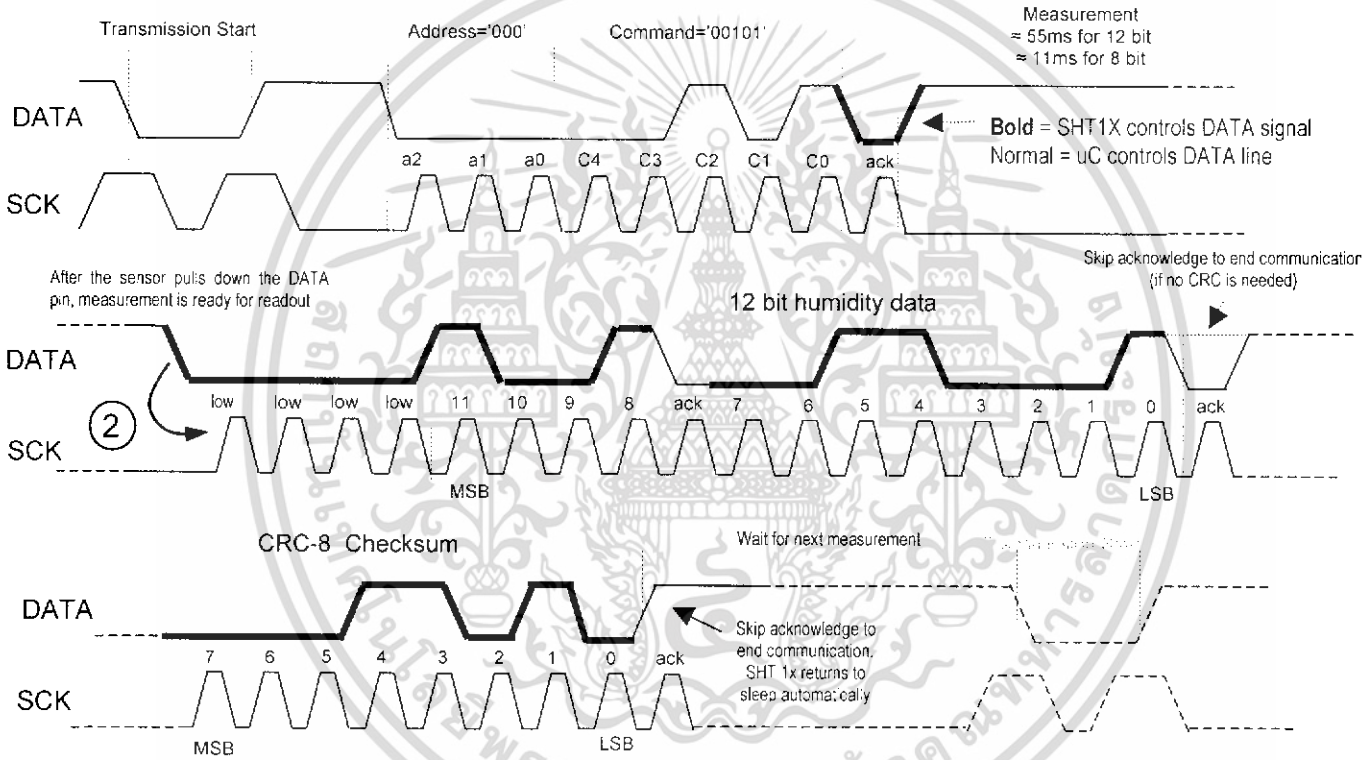


Figure 5 Example RH measurement sequence for value "0000'1001 ' 0011'0001"= 2353 = 75.79%RH

2.3 Status Register

Some of the advanced functions of the SHT1x/SHT7x are available through the status register. The following section gives a brief overview of these features. Please consult application note "Status Register" for more information.

2.3.1 *Measurement resolution*

The default measurement resolution of 14bit (temperature) and 12bit (humidity) can be reduced to 12 and 8 bit. This is especially useful in high speed or extreme low power applications.

2.3.2 *End of Battery*

The "End of Battery" function detects VDD voltages below 2.47V. Accuracy is ±0.05V

2.3.3 *Heater*

An on chip heating element can be switched on. It will increase the temperature of the sensor by approximately 5°C. Power consumption will increase by 8mA @ 5V.
Applications:

- By comparing temperature and humidity values before and after switching on the heater, proper functionality of both sensors can be verified.
- In high RH environments heating the sensor element will avoid condensation.

Warning: The built-in calibration is not correct while the sensmitter is heated!

Please consult application note "Status Register" for more information on how to access and use these features.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3 Specifications SHT1x/SHT7x

3.1 Absolute Maximum Ratings

Ambient Storage Temperature: -40°C to 120°C

3.2 Operating Conditions

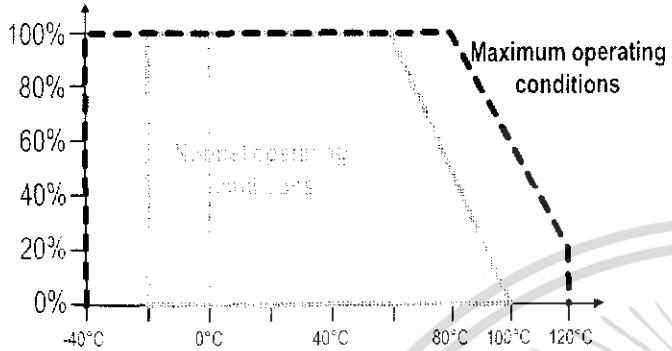


Figure 6 Recommended operating conditions

Conditions outside the recommended range may temporarily offset the RH signal up to ±3%RH. After return to normal conditions it will slowly return close to calibration state by itself.
To accelerate this process we recommend the following reconditioning procedure:
90°C at <5%RH for 24h followed by
20-30°C at >74%RH for 48h

Prolonged exposure to extreme conditions may accelerate ageing.

3.3 Special Conditions

Extensive tests were performed in various environments.

Environment	Norm	Results[4]
Temperature Cycles	JESD22-A104-A -40 +125°C, 1000cy	Within Specifications
Pressure Cooker	JESD22-A110-B 2.3bar 125°C 85%RH	Reversible shift by +2% RH
Salt Atmosphere	DIN-50021ss	Within Specifications
Freezing cycles fully submerged	-20 +90°C, 100cy 30min dwell time	Reversible shift by +2% RH

Table 3 Qualification tests

Please contact SENSIRION for additional qualification information.

3.4 Electrical Specifications[1]

3.4.1 ESD (Electrostatic Discharge)

ESD immunity is qualified according to MIL STD 883E, method 3015 (Human Body Model at ±2kV)).
Latch-up immunity is provided at a force current of ±100 mA with T_{amb}=80°C according to JEDEC 17.
See application note "ESD, Latchup and EMC" for more information.

3.4.2 DC Characteristics

VDD=5V, Temperature= 25°C unless otherwise noted

Parameter	Conditions	Min.	Typ.	Max.	Units
Power supply DC		2.4	5	5.5	V
Supply current	measuring		550		µA
	average	2[2]	28[3]		µA
	sleep		0.3	1	µA
Low level output voltage		0		20%	Vdd
High level output voltage		75%		100%	Vdd
Low level input voltage	Negative going	0		20%	Vdd
High level input voltage	Positive going	80%		100%	Vdd
Input current on pads				1	µA
Output peak current	on			4	mA
	Tristated (off)		10		µA

Table 4 SHT1x/SHT7x DC Characteristics

3.4.3 I/O Characteristics

	Parameter	Conditions	Min	Typ.	Max.	Unit
F_{SCK}	SCK frequency	VDD > 4.5 V			10	MHz
		VDD < 4.5 V			1	MHz
T_{RFO}	DATA fall time	Output load 5 pF	3.5	10	20	ns
		Output load 100 pF	30	40	200	ns
T_{CLH}	SCK high time		100			ns
T_{CLL}	SCK low time		100			ns
T_V	DATA valid time			250		ns
T_{SU}	DATA set up time		100			ns
T_{HO}	DATA hold time		0	10		ns
T_R/T_F	SCK rise/fall time				200	ns

Table 5 SHT1x/SHT7x I/O Signals Characteristics

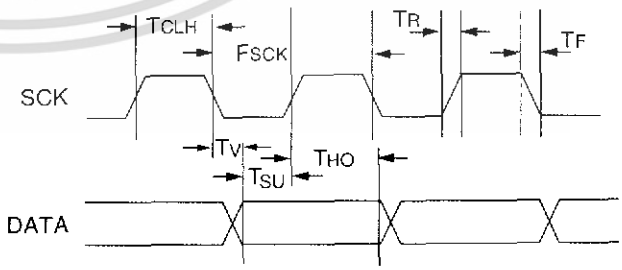


Figure 7 Timing Diagram

[1] Parameters are periodically sampled and not 100% tested
[2] With one measurement of 8 bit accuracy without OTP reload per second
[3] With one measurement of 12bit accuracy per second
[4] The temperature sensor passed all tests without any drift Package and electronics also passed 100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา รวมถึงการลบข้อความนี้ออกจากของเอกสารทุกครั้งที่มีการนำไปใช้

4 Physical Dimensions and Mounting Information

4.1 SHT1x (surface mountable)

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial data bidirectional
3	SCK	Serial clock input
4	VDD	Supply 2.4 – 5.5V
5-8	nc	Do not connect pins on right side

Table 6 SHT1x Pin Description

4.1.1 Package type

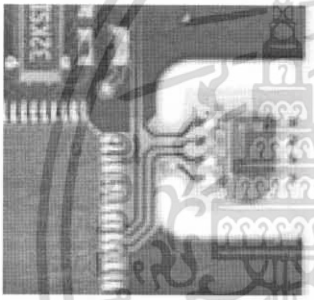
The SHT1x is supplied in a surface-mountable LCC type package. The sensors housing consists of a Liquid Crystal Polymer (LCP) cap with epoxy glob top on a standard 0.8mm FR4 substrate.
Device size is 7.62 x 5.08 x 2.5 mm. Weight 100mg

4.1.2 Soldering Information

The SHT1x can be soldered using standard reflow ovens at maximum 225°C for 20 seconds. For manual soldering contact time must be limited to 5 seconds at up to 350°C.
After soldering the devices must be stored at >74%RH for at least 24h to allow the polymer to recover.
Please consult the application note "Soldering procedure" for detailed instructions.

4.1.3 Delivery Conditions

The SHT1x will be delivered in standard IC tubes by 80 pieces per tube. Other delivery options may be available on request.



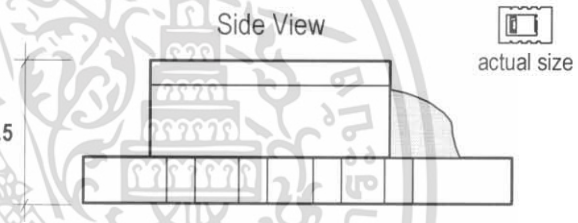
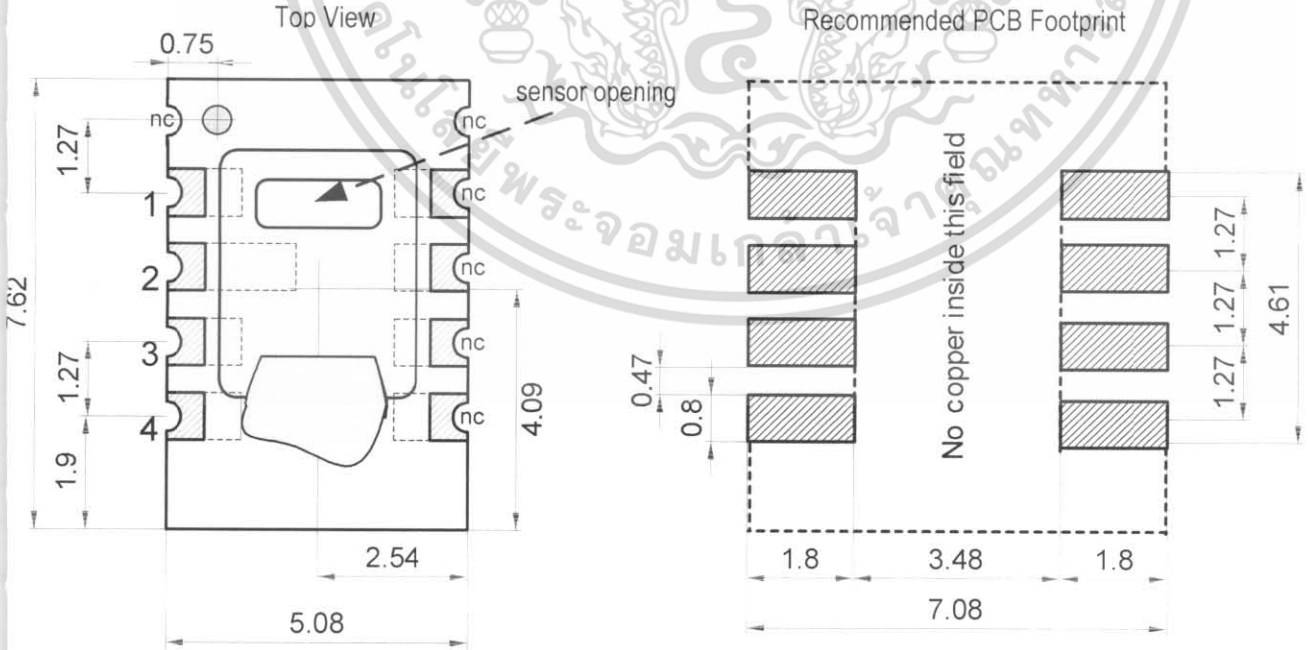


Figure 8 SHT1x Mounting example



igure 9 SHT1x drawing and footprint dimensions in mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 SHT7x (4-pin single-in-line)

SHT7x are available starting Q3 2002

Pin	Name	Comment
1	*SCK*	*Serial clock input*
2	*VDD*	*Supply 2.4 – 5.5V*
3	*GND*	*Ground*
4	*DATA*	*Serial data bidirectional*

Table 7 SHT7x Pin Description

4.2.1 Package type

The device is supplied in a single-in-line pin type package.
The sensors housing consists of a Liquid Crystal Polymer (LCP) cap with epoxy glob top on a standard 0.6mm FR4 substrate.
The sensor head is connected to the pins by a small bridge to minimize heat conduction and response times.
A 100nF capacitor is mounted on the back side between VDD and GND.
Weight 168mg, Weight of sensor head 73mg
All pins are gold plated to avoid corrosion.
Pins mate with most 1.27mm (0.05") sockets
e.g.: Preci-dip / Mill-Max 851-93-004-20-001 or similar

4.2.2 Soldering Information (Preliminary)

The SHT7x may be soldered using standard wave soldering systems at maximum 225°C for 20 seconds.
For manual soldering contact time must be limited to 5 seconds at up to 350°C.
After soldering the devices must be stored at >74%RH for at least 24h to allow the polymer to recover.
Please consult the application note "Soldering procedure" for detailed instructions.

4.2.3 Delivery Conditions

The SHT7x will be delivered in trays by *xx* pieces per tray.
Other delivery options may be available on request.

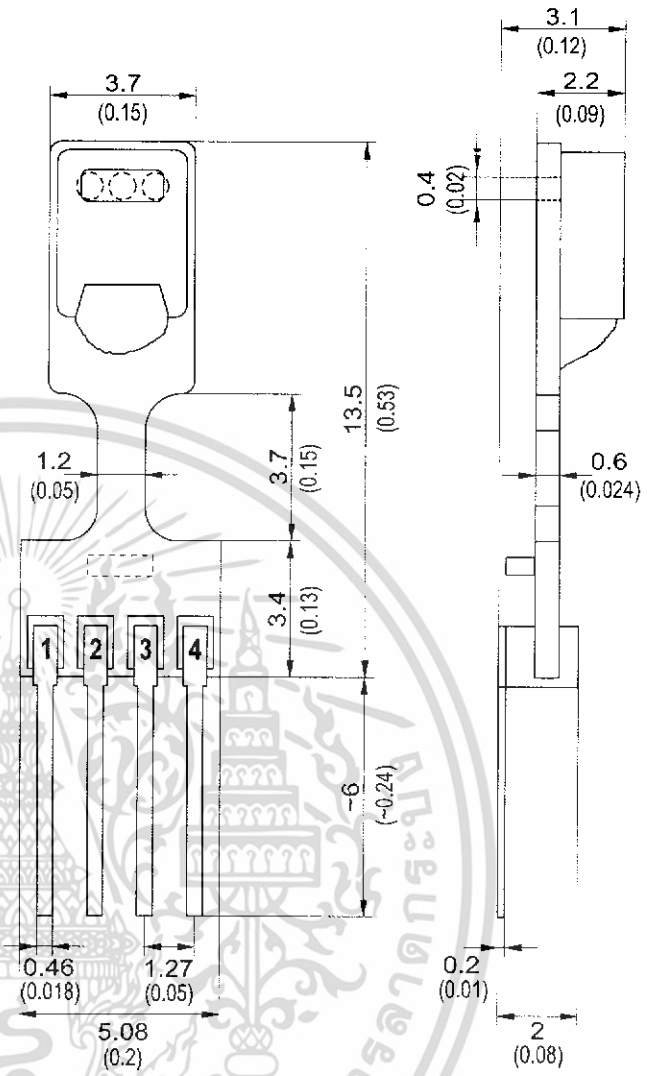


Figure 10 SHT7x dimensions in mm (inch), preliminary

4.3 Other Packages

Other packaging options may be available on request.

4.4 Production date

The production date is printed onto the cap in white numbers in the form wwy. e.g. "351" = week 35, 2001.

4.5 Mounting Recommendations

The relative humidity of a gas strongly depends on its temperature. It is therefore essential to keep the sensor at the same temperature as the air of which the humidity is to be measured.
If the sensmitter shares a PCB with heating electronic components it should be mounted below the heat source and the housing must remain well ventilated. To reduce heat conduction copper layers between the SHT1x and the rest of the PCB should be minimized and a slit may be milled in between.
Prolonged direct exposure of the SHT1x/SHT7x to sunshine or UV radiation should be avoided.

4.6 Wiring considerations and signal integrity

Carrying the SCK and DATA signal parallel and in close proximity (e.g. in wires) for more than 10cm may result in cross talk and loss of communication. This may be resolved by routing VDD and/or GND between the two data signals.

Please see the application note "ESD, Latchup and EMC" for more information.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 Revision history

Date	Page	Changes
February 2002	1-9	First public release
February (2) 2002	4	Corrected CRC information to match application note
March 2002	2	Extended SHT11 3.5 accuracy range to 20%-80%
	8	Added image of mounting example
	2	Changed coefficients of temperature conversion formula
		Various small modifications
June 2002	6	Extended "no copper" area in figure 9
	5	Corrected timing diagram and table 5 with setup times for DATA
	5	Added qualification data table
	7	Added SHT7x information
	4	Renamed "End of Life" function to "End of Battery"

The latest version of this document and all application notes can be found at:
www.sensirion.com/en/download/humiditysensor/SHT11.htm

6 Important Notices

The warranty for each SENSIRION AG product comes in the form of a written warranty which governs sale and use of such product. Such warranty is contained in the printed terms and conditions under which such product is sold, or in a separate written warranty supplied with the product. Please refer to such written warranty with respect to its applicability to certain applications of such product.

These products may be subject to restrictions on use. Please contact SENSIRION AG for a list of the current additional restrictions on these products. By purchasing these products, the purchaser of these products agrees to comply with such restrictions. Please contact SENSIRION AG for clarification of any restrictions described herein.
SENSIRION AG reserves the right, without further notice, to change the SENSIRION SHT1x/SHT7x Relative Humidity and Temperature Sensor product specifications and/or information in this document and to improve reliability, functions and design.

SENSIRION AG assumes no responsibility or liability for any use of SENSIRION SHT1x/SHT7x product. Application examples and alternative uses of the SENSIRION SHT1x/SHT7x are for illustration purposes only and SENSIRION AG makes no representation or warranty that such applications shall be suitable for the use specified.

Copyright© 2001-2002, SENSIRION AG.
All rights reserved.

7 Caution

The inherent design of this component causes it to be sensitive to electrostatic discharge (ESD). To prevent ESD-induced damage and/or degradation, take normal ESD precautions when handling this product.
See application note "ESD, Latchup and EMC" for more information.

8 Warning, personal injury

Do not use this product as safety or emergency stop devices or in any other application where failure of the product could result in personal injury. **Failure to comply with these instructions could result in death or serious injury.**

Headquarters and Sales Office

SENSIRION AG
Eggbühlstr. 14
P.O. Box
CH-8052 Zürich
Switzerland

Phone: + 41 (0)1 306 40 00
Fax: + 41 (0)1 306 40 30
e-mail: info@sensirion.com
http://www.sensirion.com/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้