

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

คู่มือเด็กทารกแรกเกิด

INFANT INCUBATOR



โดย

นางสาว วาทีทยา โอพารัตน์มณี

นางสาว ศกุนต์ลาวัลย์ บุญเคติม

นางสาว สันติภาพ เอ็มพันธุ์

2548

เลขหมู่.....

เลขทะเบียน..... 73130

วัน,เดือน,ปี..... 4 ก.ค. 2550

b. 41๙๕12๙3
i. ....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตู้อบเด็กทารกแรกเกิด**  
**INFANT INCUBATOR**



โดย  
นางสาว วาทีศยา โอพารัตน์มณี รหัส 45010702  
นางสาว ศกุนต์ลาวัณย์ บุญเคลิ้ม รหัส 45010741  
นางสาว สันติภาพ เอ็มพันธ์ รหัส 45010817

อาจารย์ที่ปรึกษา  
ผศ.ดร.ชูชาติ ปิณฑวิรุจน์

**ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต**  
**สาขาอิเล็กทรอนิกส์**  
**คณะวิศวกรรมศาสตร์**  
**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**  
**ปีการศึกษา 2548**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโท ปีการศึกษา 2548

ภาควิชา สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง คู่มือเด็กทารกแรกเกิด

ผู้จัดทำ

1. นางสาว วาติทยา โอพารัตน์มณี รหัสประจำตัว 45010702
2. นางสาว ศกุนต์สาวณีย์ บุญเคลิ้ม รหัสประจำตัว 45010741
3. นางสาว สันติภาพ เอมพันธุ์ รหัสประจำตัว 45010817



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## คู่มือเด็กทารกแรกเกิด

นางสาว วาติทยา โอพารัตน์มณี รหัส 45010702

นางสาว ศกุนต์ลาวัณย์ บุญเคลิ้ม รหัส 45010741

นางสาว สันติภาพ เอ็มพันธุ์ รหัส 45010817

ผศ.ดร.ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2548

### บทคัดย่อ

รายงานฉบับนี้มีวัตถุประสงค์ เพื่อศึกษาและพัฒนาคุณภาพตู้อบเด็กทารกแรกเกิด (Infant Incubator) จากตู้อบเด็กเครื่องเก่า ให้มีประสิทธิภาพในการทำงานสูงขึ้น เนื่องจากตู้อบเด็กทารกแรกเกิดถือเป็นเครื่องมือทางการแพทย์ที่สำคัญอย่างหนึ่ง ที่จะช่วยบริหารร่างกายทารกให้สามารถปรับสภาพเข้ากับสิ่งแวดล้อมได้

โครงการนี้ได้นำเอาเทคโนโลยีไมโครคอนโทรลเลอร์ มาประยุกต์ใช้ในระบบควบคุมทั้งหมดของตู้อบ โดยสามารถควบคุมอุณหภูมิภายในตู้อบได้ตามความต้องการ และสั่งงานผ่าน keypad โดยนำอัลกอริทึมของระบบควบคุมอัตโนมัติแบบพีไอดี (PID Control) มาใช้ในการเขียนโปรแกรมควบคุมค่าอุณหภูมิและความชื้น ซึ่งใช้ตัวเซนเซอร์ SHT 15 เป็นอุปกรณ์ตรวจวัดค่าอุณหภูมิและความชื้นตลอดเวลา จากนั้นนำค่าที่ได้มาประมวลผลด้วยไมโครคอนโทรลเลอร์ ให้แสดงผลออกทางจอ LCD ในกรณีเกิดเหตุขัดข้อง ได้ติดตั้งระบบสัญญาณเตือน อย่างน้อย 3 กรณี คือ ไฟฟ้าขัดข้อง , อุณหภูมิเกิน และน้ำแห้ง

## Infant Incubator

Wathitaya Olarnratmanee ID. 45010702

Sakunlawan Boonklerm ID. 45010741

Santipap Ampun ID. 45010817

Assist. Prof. Dr. Chuchart Pintavirooj Advisor


Educational Year 2005

### Abstract


This report concerns about the design and construction of infant incubator, which is the essential medical device for a newborn where the controlled atmosphere is needed. The operation of the incubator is based on using the microcontroller. The input of the microcontroller includes the sensing temperature, the sensing humidity and the keypad; where the output of the microcontroller is the LCD display and the proportionally controlled signal for the heating element. The sensing temperature and the sensing humidity are detected by SHT15 Chip, which is capable of communicating with the microcontroller via a series protocol. The keypad is used to set up the controlled temperature, the alarmed temperature and the other alarm setting including, for example, the power-failure alarm. The LCD display shows the setting temperature, the sensing temperature and the humidity.

### กิตติกรรมประกาศ

คณะผู้จัดทำขอกราบขอบคุณ ผศ. ดร. ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา ที่กรุณาให้แนวคิดและคำปรึกษาแนะนำ ตลอดจนตรวจแก้ไขข้อบกพร่อง ทำให้โครงการสำเร็จลุล่วงด้วยดี ขอขอบคุณอาจารย์ และพี่ๆ นักศึกษาปริญญาโทภาควิชาอิเล็กทรอนิกส์ทุกท่าน ที่ได้กรุณาสละเวลาให้คำแนะนำ ตลอดจนให้การช่วยเหลือ ตั้งแต่เริ่มต้นจนเสร็จสมบูรณ์

  
.....  
(นางสาว วาতিทยา โอพารัตน์มณี)

  
.....  
(นางสาว สกุนต์ลาวัลย์ บุญเกลี่ยม)

  
.....  
(นางสาว สันติภาพ เอ็มพันธ์)

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
คำนำ	IV
สารบัญ	V
สารบัญตาราง	VII
สารบัญรูป	VIII
บทที่ 1 บทนำ	1
1.1 ความเป็นมาของโครงการและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 แนวคิดและขอบเขตของโครงการ	2
1.4 โครงสร้างของเนื้อหาของโครงการ	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 ทฤษฎีเกี่ยวกับทารกแรกเกิด	4
2.1.1 สรีระวิทยาของทารกแรกเกิด	5
2.1.2 ระบบหายใจ	5
2.1.3 การดูแลทารกแรกเกิด	5
2.1.4 การควบคุมอุณหภูมิกายทารกแรกเกิด	6
2.1.5 การเลือกใช้คู่ออบ	6
2.1.6 อุณหภูมิคู่ออบที่เหมาะสมกับอายุและน้ำหนักของทารกแรกเกิด	6
2.2 ระบบควบคุมของคู่ออบเด็กทารก	8
2.2.1 เซนเซอร์วัดอุณหภูมิและความชื้น	9
2.2.2 ส่วนแสดงผลทาง LCD Module	13
2.2.3 ส่วนควบคุมอุณหภูมิและความชื้น	14
2.2.4 ส่วนสั่งงานผ่านคีย์สวิตช์	16
2.2.5 ระบบควบคุมแบบอัตโนมัติชนิด PID	16
2.2.6 การส่งข้อมูลแบบ I <sup>2</sup> C ของ EEPROM	21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
2.2.7 ตัวประมวลผลควบคุม(Microcontroller)	23
บทที่ 3 ขั้นตอนการดำเนินงาน	37
3.1 การออกแบบวงจรต่างๆ ภายในตู้อบเด็ก	37
3.1.1 วงจรแหล่งจ่ายไฟของระบบควบคุม	37
3.1.2 ส่วนตรวจจับอุณหภูมิที่ใช้ SHT15	38
3.1.3 วงจรทรานซิสเตอร์และแยกโพล	39
3.1.4 วงจรตรวจจับสัญญาณชาयน์	39
3.1.5 วงจรส่วนแสดงผล	40
3.1.6 วงจรควบคุมการทำงานหลัก	41
3.2 โปรแกรมการทำงานของตู้อบเด็ก	44
3.2.1 Flow chart แสดงการทำงานของโปรแกรม	44
3.2.2 อธิบายโปรแกรมการทำงานของตู้อบเด็ก	46
3.3 การออกแบบส่วนประกอบภายนอก	47
3.3.1 ตู้ครอบตัวเด็กและแผ่นรองตัวเด็ก	47
3.3.2 Chamber	48
3.3.3 ตู้ฐานรอง	48
3.3.4 ส่วนแสดงผล	48
บทที่ 4 ขั้นตอนการทดลองและผลการทดลอง	50
4.1 ทดสอบความเป็นเชิงเส้นของช่วงเวลาที่เพิ่มขึ้นของอุณหภูมิเทียบกับเวลา	50
4.2 ปรับค่าค่าคงที่ (K) ต่างๆของระบบพีไอดี (PID) ที่เหมาะสมกับระบบมากที่สุด	51
บทที่ 5 สรุป วิจารณ์ผลการทดลอง และข้อเสนอแนะ	64
ภาคผนวก ก	65
ภาคผนวก ข	90
หนังสืออ้างอิง	98

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญตาราง

ตารางที่	หน้า
2.1 อุณหภูมิคู่อบที่เหมาะสมกับอายุและน้ำหนักของทารกแรกเกิด	7
2.2 แสดงคุณสมบัติ DC ของ SHT15	10
2.3 แสดงขาสัญญาณของ SHT15	11
2.4 แสดงคุณสมบัติของสัญญาณ SCK และ DATA	13
4.1 แสดงค่า Overshoot , Risetime และ Sattling Time ที่ $K_p$ ค่าต่างๆ	54
4.2 แสดงค่า Overshoot , Risetime และ Sattling Time ที่ $K_i$ ค่าต่างๆ	57
4.3 แสดงค่า Overshoot , Risetime และ Sattling Time ที่ $K_d$ ค่าต่างๆ	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญรูป

รูปที่	หน้า
2.1 บล็อกไดอะแกรมของระบบ	8
2.2 รูป IC SHT1X และ SHT7X	9
2.3 แสดงลักษณะของ SHT15	10
2.4 แสดง Block Diagram ของ SHT15	11
2.5 แสดงการต่อ SHT15 กับ ไมโครคอนโทรลเลอร์	12
2.6 แสดง Timing Diagram ในช่วงการส่งข้อมูลของ SHT15	12
2.7 แสดงคุณสมบัติของ Timing Diagram ของสัญญาณ DATA และ SCK ของ SHT15	13
2.8 แสดงการเชื่อมต่อ LCD Module กับ Microcontroller	14
2.9 แสดงลักษณะของ MOC3021	15
2.10 แสดงอุปกรณ์ BTA12 600B	15
2.11 แสดงการเชื่อมต่อ Keypad กับ Micro controller	16
2.12 แสดงระบบควบคุมแบบ PID	16
2.13 แสดงการเข้าสู่จุด Set point ของ Proportional (P)	17
2.14 แสดงการเข้าสู่จุด Set point ของ Integration (I)	18
2.15 แสดงการเข้าสู่จุด Set point ของ Derivative (D)	18
2.16 แสดงการเข้าสู่จุด Set point ของ Proportional + Integration + Derivative (PID)	19
2.17 แสดงการเปรียบเทียบการตอบสนองของระบบควบคุม 3 ชนิด	19
2.18 ความสัมพันธ์ระหว่าง $e(t)$ กับ $t$	20
2.19 การติดต่ออุปกรณ์หลายๆตัวของ I <sup>2</sup> C	21
2.20 แสดงสัญญาณ Acknowledge ของ I <sup>2</sup> C	22
2.21 โครงสร้างของไมโครคอนโทรลเลอร์	23
2.22 โครงสร้างของหน่วยความจำใน MCS-51	24
2.23 โปรแกรมเมมโมรี่ของ MCS 51	25
2.24 การต่อ External program memory และ External data memory ร่วมกัน	26
2.25 การต่อ External program memory	27
2.26 Data memory ของ MSC-51	28

## สารบัญรูป (ต่อ)

รูปที่	หน้า
2.27 Internal Data Memory	29
2.28 การต่อใช้งานของ External data memory	29
2.29 โครงสร้างภายในของ Microcontroller MCS-51	30
2.30 รายละเอียดของตำแหน่งข้อมูลต่างๆ ใน SFR	31
2.31 การต่อ CPU Timing	34
2.32 Machine Cycles	35
2.33 ขบวนการ Fetch / Execute	36
3.1 วงจรแหล่งจ่ายไฟของระบบควบคุม	38
3.2 ตัวอย่างการต่อใช้งานของ SHT15	38
3.3 แสดงวงจรทรานซิสเตอร์และแยกโหลด	39
3.4 แสดงวงจรตรวจจับสัญญาณชานน์	39
3.5 สัญญาณชานน์เมื่อผ่านวงจรคอมพาราเตอร์ที่ใช้ LM339	40
3.6 แสดงการเชื่อมต่ออุปกรณ์ส่วนแสดงผลข้อมูล	40
3.7 วงจรส่วนที่ใช้ในการวัดและประมวลผลอุณหภูมิและความชื้น	42
3.8 วงจรส่วนที่ควบคุมการแสดงผลผ่านทาง LCD และควบคุมการทำงานทางเฟสของขดลวดความร้อน	43
3.9 Flow chart แสดงการทำงานของโปรแกรมหลัก	44
3.10 Flow chart แสดงการทำงานของโปรแกรมน้อยการตั้งค่าอุณหภูมิ	45
3.11 Flow chart แสดงการทำงานของโปรแกรมการควบคุม PID	46
3.12 ส่วนกระโجمกรอบ	47
3.13 แผ่นรองตัวเด็ก	47
3.14 คู่มือรอง	48
3.15 ส่วนหน้าปิดแสดงผล และสัญญาณเตือน	48
3.16 แสดงลักษณะโดยรวมของตู้อบเค้ก	49
4.1 กราฟค่าความแตกต่างของอุณหภูมิที่วัดจากเซนเซอร์ กับวัดจากเทอร์โมมิเตอร์	50
4.2 กราฟแสดงการปรับค่าค่าคงที่ของพี ( $K_p=90$ )	51
4.3 กราฟแสดงการปรับค่าค่าคงที่ของพี ( $K_p=110$ )	52
4.4 กราฟแสดงการปรับค่าค่าคงที่ของพี ( $K_p=120$ )	52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5	กราฟแสดงการปรับค่าค่าคงที่ของพี ( $K_p=150$ )	53
4.6	กราฟแสดงการเปรียบเทียบอุณหภูมิ เมื่อใช้ค่า $K_p$ ต่างๆ	54
4.7	กราฟแสดงการปรับค่าค่าคงที่ของพีไอ ( $K_p=110, K_i=0.05$ )	55
4.8	กราฟแสดงการปรับค่าค่าคงที่ของพีไอ ( $K_p=110, K_i=0.1$ )	55
4.9	กราฟแสดงการปรับค่าค่าคงที่ของพีไอ ( $K_p=110, K_i=0.5$ )	56
4.10	กราฟแสดงการเปรียบเทียบอุณหภูมิ เมื่อใช้ค่า $K_p=110$ และ $K_i$ ต่างๆกัน	56
4.11	กราฟแสดงการปรับค่าค่าคงที่ของพีไอดี ( $K_p=110, K_i=0.1, K_d=0.05$ )	57
4.12	กราฟแสดงการปรับค่าค่าคงที่ของพีไอดี ( $K_p=110, K_i=0.5, K_d=0.1$ )	58
4.13	กราฟแสดงการปรับค่าค่าคงที่ของพีไอดี ( $K_p=110, K_i=0.7, K_d=0.5$ )	58
4.14	กราฟแสดงการเปรียบเทียบอุณหภูมิ เมื่อใช้ค่า $K_p = 110, K_i = 0.5$ และ $K_d$ ต่างๆกัน	59
4.15	กราฟแสดงการเปรียบเทียบการควบคุมแบบต่างๆ กัน	60
4.16	กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้ เท่ากับ 28C	61
4.17	กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้ เท่ากับ 30C	61
4.18	กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้ เท่ากับ 32C	62
4.19	กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้ เท่ากับ 34C	62
4.20	กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้ เท่ากับ 36C	63
4.21	กราฟแสดงอุณหภูมิค่าต่างๆ ภายในตู้ ที่สามารถควบคุมได้	63

## บทที่ 1

### บทนำ

#### 1.1 ความเป็นมาของโครงการและความสำคัญของปัญหา

ทารกแรกเกิด โดยเฉพาะทารกเกิดก่อนกำหนด มีความสามารถจำกัดในการปรับตัวให้อุณหภูมิร่างกายคงที่ (homeothermia) ส่งผลให้อุณหภูมิแกนกลางของร่างกาย (core body temperature) เปลี่ยนแปลงไปตามอุณหภูมิสิ่งแวดล้อมที่เปลี่ยนแปลง หากไม่ควบคุมอุณหภูมิกายของทารกให้อยู่ในเกณฑ์ปกติ จะทำให้อัตราการเกิดโรค (morbidity) และอัตราตาย (mortality) ในทารกเพิ่มขึ้น เนื่องจากภาวะอุณหภูมิร่างกายต่ำจะซ้ำเติมทารกที่ป่วยด้วยโรคใดโรคหนึ่งอยู่แล้ว ให้เจ็บป่วยรุนแรงขึ้น การป้องกันจึงเป็นสิ่งสำคัญที่สุด ผู้บดเด็กทารกแรกเกิดจัดเป็นเครื่องมือทางแพทย์ที่สำคัญ สำหรับช่วยทารกแรกเกิดให้รอดชีวิตและเจริญเติบโตได้อย่างปกติ

ผู้บดเด็กทารกแรกเกิดได้พัฒนาจากเครื่องที่ควบคุมด้วยระบบปฏิบัติงานเอง(manual) ถึงระบบควบคุมกึ่งอัตโนมัติ จนกระทั่งเป็นระบบควบคุมอัตโนมัติ(Automatic) การควบคุมด้วยระบบอัตโนมัติทำให้ประสิทธิภาพการทำงานของตู้บดดีขึ้น และสะดวกง่ายต่อการใช้งาน

เครื่องที่ควบคุมด้วยระบบอัตโนมัติ ได้นำเอาเทคโนโลยีไมโคร โปรเซสเซอร์มาประยุกต์ใช้ควบคุมการทำงานต่างๆของระบบ เพื่อให้ผู้บดเด็กทำงานอย่างมีประสิทธิภาพสูงสุด

#### 1.2 วัตถุประสงค์

1. เพื่อศึกษาระบบควบคุมอัตโนมัติที่มีตัวไมโคร โปรเซสเซอร์ ควบคุมระบบ
2. เพื่อออกแบบระบบวัดอุณหภูมิ และความชื้น
3. เพื่อออกแบบระบบควบคุมอัตโนมัติอุณหภูมิ และความชื้น
4. เพื่อพัฒนาประสิทธิภาพผู้บดเด็กแบบเก่าให้มีประสิทธิภาพการทำงานมากขึ้น
5. เพื่อช่วยชีวิตทารกแรกเกิดให้รอดชีวิต และเจริญเติบโตได้อย่างปกติ

### 1.3 แนวคิดและขอบเขตของโครงการ

โครงการนี้จัดทำขึ้นเพื่อศึกษาและพัฒนาตู้อบเด็กทารกแรกเกิด (Infant Incubator) รุ่นเก่าให้มีประสิทธิภาพในการทำงานสูงขึ้น ซึ่งตู้อบเด็กทารกแรกเกิด มีส่วนประกอบที่สำคัญหลายส่วน ทั้งส่วนควบคุมอัตโนมัติ และส่วนแสดงผล

ส่วนควบคุมอัตโนมัติประกอบด้วย ส่วนเซนเซอร์วัดอุณหภูมิและความชื้น ซึ่งจะทำการวัดอุณหภูมิและความชื้นตลอดเวลา และส่งค่าไปยังไมโครคอนโทรลเลอร์เพื่อประมวลผลเปรียบเทียบข้อมูลระหว่างค่าที่วัดได้กับค่าที่ต้องการควบคุม โดยค่าที่ต้องการควบคุมจะผ่านทางดีวีซีวีดีซ์ ถ้าหากค่าที่วัดได้ไม่ตรงกับค่าที่ต้องการควบคุม จะสั่งการให้อุปกรณ์ควบคุมอุณหภูมิทำงาน เมื่ออุณหภูมิมีค่าตรงกับค่าที่ตั้งไว้ อุปกรณ์ควบคุมอุณหภูมิจะหยุดทำงาน

ส่วนแสดงผล มีการติดตั้งจอ LCD Display เพื่อแสดงค่าอุณหภูมิและความชื้นที่วัดได้ ณ เวลานั้นๆ นอกจากนี้ ยังมีระบบสัญญาณเตือนอัตโนมัติทั้งแสงและเสียง ในกรณีที่ตู้อบเด็กทารกมีความผิดปกติเกิดขึ้น เช่น ระบบไฟฟ้าขัดข้อง, อุณหภูมิเกิน, น้ำแห้ง เป็นต้น ทั้งนี้เพื่อประสิทธิภาพในการทำงานของตู้อบเด็กทารกแรกเกิด

### 1.4 โครงสร้างเนื้อหาของโครงการ

- บทที่ 1 บทนำ กล่าวถึง ความเป็นมาของโครงการฯ วัตถุประสงค์ แนวคิดและขอบเขต โครงสร้างเนื้อหาโครงการ และประโยชน์ที่คาดว่าจะได้รับจากการทำโครงการ
- บทที่ 2 ทฤษฎีที่เกี่ยวข้อง
- บทที่ 3 ขั้นตอนการดำเนินงาน การออกแบบ และการสร้างระบบ
- บทที่ 4 การทดลอง และผลการทดลอง
- บทที่ 5 สรุป วิเคราะห์ผลการทดลอง และข้อเสนอแนะ

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- เกิดประสบการณ์เรียนรู้ และเข้าใจ การทำงานหลักของตู้อบเด็กทารกแรกเกิด
- สามารถนำเทคโนโลยีไมโครคอนโทรลเลอร์มาประยุกต์ใช้กับชิ้นงานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สามารถนำระบบควบคุมแบบ PID มาประยุกต์ใช้กับการควบคุมอุณหภูมิ และความชื้น
- สามารถออกแบบระบบวัด และควบคุมอุณหภูมิ ความชื้นเพื่อใช้งานกับระบบกึ่งอัตโนมัติ หรือแบบอัตโนมัติให้มากที่สุดเท่าที่จะเป็นไปได้
- ลดอัตราการเกิดโรค (Morbidity) และอัตราการตาย (mortality) ในทารกแรกเกิด
- เป็นข้อมูลสนับสนุนผู้วิจัย เพื่อใช้ประโยชน์ในการพัฒนาเทคโนโลยีทางด้านเครื่องมือและอุปกรณ์ทางการแพทย์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

ทฤษฎีที่เกี่ยวข้องแบ่งออกเป็น 2 ส่วนดังนี้

### 2.1 ทฤษฎีเกี่ยวกับทารกแรกเกิด

2.1.1 สรีระวิทยาของทารกแรกเกิด

2.1.2 ระบบหายใจ

2.1.3 การดูแลทารกแรกเกิด

2.1.4 การควบคุมอุณหภูมิกายทารกแรกเกิด

2.1.5 การเลือกใช้อุปกรณ์

2.1.6 อุณหภูมิคู่อุปที่เหมาะสมกับอายุและน้ำหนักของทารกแรกเกิด

neutral thermal environmental temperature

### 2.2 ระบบควบคุมของตู้อบเด็กทารกแรกเกิด

2.2.1 เซนเซอร์ตรวจจับอุณหภูมิและความชื้น

2.2.2 ส่วนแสดงผล LCD Module

2.2.3 ส่วนควบคุมอุณหภูมิและความชื้น

2.2.4 ส่วนสั่งงานผ่านคีย์สวิตช์

2.2.5 ระบบควบคุมแบบอัตโนมัติชนิด PID

2.2.6 การส่งข้อมูลแบบ I<sup>2</sup>C ของ EEPROM

2.2.7 ตัวประมวลผลควบคุม(Micro controller)

### 2.1 ทฤษฎีเกี่ยวกับทารกแรกเกิด

ทารกแรกเกิด (Newborn Infant) หมายถึง ทารกแรกเกิดจนถึงอายุ 1 เดือน ซึ่งถือว่าเป็นช่วงที่เสี่ยงอันตรายมาก แบ่งเป็น 3 ประเภทดังนี้

- ทารกที่คลอดครบกำหนด (Full Term Baby) หมายถึง ทารกที่คลอดจากครรภ์มารดา ในระหว่าง 38-42 สัปดาห์ ทารกประเภทนี้จะเติบโตเต็มที่และสามารถมีชีวิตอยู่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทารกที่คลอดก่อนกำหนด (Pre Term Baby) หมายถึง ทารกที่คลอดจากครรภ์มารดา ก่อนการตั้งครรภ์ครบ 37 สัปดาห์ และมีน้ำหนักตัวแรกคลอดต่ำกว่า 2500 กรัม

- ทารกคลอดเกินกำหนด (Post Term Baby) หมายถึง ทารกที่คลอดจากครรภ์มารดาภายหลังการตั้งครรภ์ 42 สัปดาห์

ทารกทั้ง 3 ประเภทนี้มีความแตกต่างกันทั้งรูปร่าง ลักษณะ การเจริญเติบโต และความต้านทานโรค โดยเฉพาะทารกที่คลอดเกินกำหนด จะมีความต้านทานโรคน้อย มีอันตราย (Mortality rate) สูงกว่า และมีอันตรายระหว่างคลอด (Birth Injury) ได้มากกว่าทารกที่คลอดก่อนกำหนด

### 2.1.1 ศรีระวิทยาของทารกแรกเกิด

ทารกเมื่ออยู่ในครรภ์มารดาจะถูกห่อหุ้มเลี้ยงดูโดยไม่ต้องหายใจและกินอาหาร เนื่องจากได้รับอาหารและออกซิเจนผ่านทางสายสะดือ พื้นที่ที่ทารกคลอดพ้นครรภ์มารดาสู่โลกภายนอก ซึ่งสภาพแวดล้อมต่างกัน ทารกจำเป็นต้องปรับตัวเพื่อให้มีชีวิตอยู่รอด

### 2.1.2 ระบบหายใจ

เป็นระบบแรกทำงานทันทีเมื่อคลอด ปอดของทารกขณะอยู่ในครรภ์มารดามีลักษณะแข็ง แฝบไม่ทำงาน ถุงลมปอดไม่พองตัว การหายใจของทารกจะเกิดขึ้นเมื่อร่างกายขาดออกซิเจน และมีคาร์บอนไดออกไซด์คั่งในกระแสโลหิต และไปกระตุ้นศูนย์ควบคุมการหายใจให้เริ่มหายใจ

นอกจากนี้การที่บริเวณทรวงอกของทารกถูกบีบรัดขณะคลอด และสัมผัสเสียดสีบริเวณ แขน ขา มือ เท้า จะเป็นการกระตุ้นให้ทารกหายใจและร้องไห้ การร้องไห้จะช่วยให้ออกขยายตัวดีขึ้น อัตราการหายใจเมื่อแรกคลอดใหม่ๆ ประมาณ 35-50 ครั้งต่อนาที เฉลี่ย 40 ครั้งต่อนาที ถ้าเกิน 60 ครั้งต่อนาทีแสดงว่าผิดปกติ

### 2.1.3 การดูแลทารกแรกเกิด

หลักการดูแลที่สำคัญมี 7 ประการ ดังนี้

1. การดูแลอุณหภูมิกายทางทวารหนักให้อยู่ที่ 37 °C
2. การดูแลทางเดินหายใจให้โล่ง และออกซิเจนในเลือดปกติ
3. การป้องกันการติดเชื้อ
4. การให้อาหาร ได้แก่ นมแม่ หรือสารน้ำในกรณีที่ได้รับนมไม่ได้
5. การรักษาเฉพาะโรค เมื่อมีการเจ็บป่วย
6. การส่งเสริมการสร้างสายสัมพันธ์ (maternal-infant bonding)
7. การดูแลด้านพัฒนาการ (development care)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.4 การควบคุมอุณหภูมิภายทารกแรกเกิด

พยาบาลมีบทบาทสำคัญในการป้องกันการสูญเสียความร้อน และควบคุมอุณหภูมิภายของทารก โดยมีหลักปฏิบัติดังนี้

#### 1. การพยาบาลทั่วไป

- ทันทีกที่คลอดรีบเช็ดผิวให้แห้ง ใส่เสื้อผ้า ห่มผ้า และให้ทารกนอนในเตียงที่อบอุ่น
- วัดอุณหภูมิภายทารกทุก 15 นาที จนกว่าอุณหภูมิจะปกติ และเปลี่ยนวัดทุก 4 ชั่วโมง
- ถ้าอุณหภูมิภายต่ำกว่าปกติคือ ต่ำกว่า 36.5 องศาเซลเซียส ไม่ต้องอาบน้ำทารก ควรทำความสะอาดร่างกายเฉพาะที่ เช่น หน้าและก้นก็พอ และห่มผ้าให้อบอุ่น
- ถ้าอุณหภูมิภายทารกยังต่ำ อยู่พิจารณาใช้ผ้าห่มไฟฟ้าหรือ โคมไฟฟ้า ตั้งส่องที่ทารก ระวังอันตรายจากความร้อน (Burn) และแสงไฟเข้าตาโดยตรง จะเกิดระคายเคืองต่อตาทารกได้
- ถ้าอุณหภูมิจึงขึ้นมาก โดยเฉพาะฤดูหนาวต้องใช้เครื่องทำความร้อน (Heater) ช่วย
- ถ้าทารกยังไม่สามารถควบคุมอุณหภูมิภายได้ควรให้อยู่ในตู้อบจนกว่าอุณหภูมิจะปกติ

#### 2. การใช้ตู้อบเด็ก ทารกแรกเกิดที่จำเป็นต้องนำเข้าตู้อบเด็ก ได้แก่

- น้ำหนักแรกเกิดน้อยกว่า 2,500 กรัม หรือ มากกว่า 4,000 กรัม
- เกิดก่อนอายุครรภ์ 37 สัปดาห์ หรือหลังอายุครรภ์ 42 สัปดาห์
- น้ำหนักผิดปกติเมื่อเทียบกับอายุครรภ์
- มีภาวะขาดออกซิเจน หายใจเร็ว หรือตัวเขียว
- พิกัดแร่ค่านีครุนแรง
- ภาวะซีด ภาวะเม็ดเลือดแดงเกิน จำเลือด

### 2.1.5 การเลือกใช้ตู้อบ

การเลือกใช้ตู้อบ มีหลักควรพิจารณาคุณสมบัติใหญ่ๆ คือ

- สามารถควบคุมอุณหภูมิและความชื้นให้คงที่ได้ง่าย
- ให้ออกซิเจนได้ง่าย
- ฝาครอบใสมองเห็นทารกได้ชัดเจนจากภายนอกตู้อบ
- มีสัญญาณเตือน เมื่อภายในตู้อบร้อนเกินไป หรือมีการถ่ายเทอากาศไม่เพียงพอ
- สามารถปรับระดับของที่นอนได้

### 2.1.6 อุณหภูมิตู้อบที่เหมาะสมกับอายุและน้ำหนักของทารกแรกเกิด ดังตารางที่ 2.1

ตารางที่ 2.1 แสดงค่าอุณหภูมิสิ่งแวดล้อมที่เหมาะสมกับอายุและน้ำหนักของทารกแรกเกิด

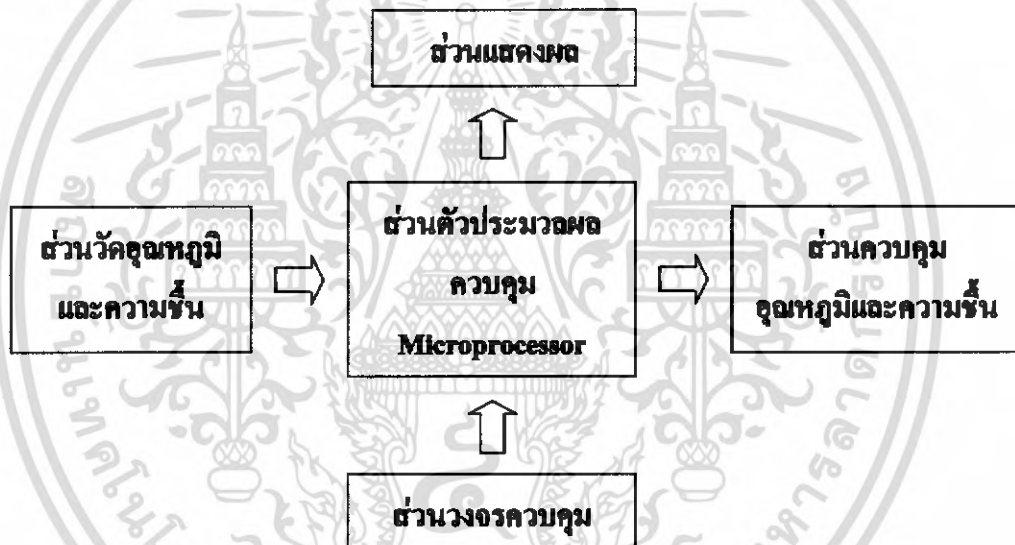
อายุน้ำหนัก (กรัม)	อุณหภูมิ (°C)	อายุน้ำหนัก (กรัม)	อุณหภูมิ (°C)
0-6 ชั่วโมง		72-96 ชั่วโมง	
ต่ำกว่า 1200	34.0-35.4	ต่ำกว่า 1200	34.0-35.0
1200-1500	33.9-34.4	1200-1500	33.0-34.0
1501-2500	32.8-33.8	1501-2500	33.1-33.2
เกิน 2500 (และ > 36 สัปดาห์)	32.0-33.8	เกิน 2500 (และ > 36 สัปดาห์)	29.8-32.8
6-12 ชั่วโมง		4-12 วัน	
ต่ำกว่า 1200	34.0-35.4	ต่ำกว่า 1500	33.0-34.0
1200-1500	33.5-34.4	1501-2500	31.0-33.2
1501-2500	32.2-33.8	เกิน 2500 (และ > 36 สัปดาห์)	
เกิน 2500 (และ > 36 สัปดาห์)	31.4-33.8	4-5 วัน	29.5-32.6
12-24 ชั่วโมง		5-6 วัน	29.4-32.3
ต่ำกว่า 1200	34.0-35.4	6-8 วัน	29.0-32.2
1200-1500	33.3-34.3	8-10 วัน	29.0-31.8
1501-2500	31.8-33.8	10-12 วัน	29.0-31.4
เกิน 2500 (และ > 36 สัปดาห์)	31.0-33.7	12-14 วัน	
24-36 ชั่วโมง		ต่ำกว่า 1500	32.6-34.0
ต่ำกว่า 1200	34.0-35.0	1501-2500	31.0-33.2
1200-1500	33.1-34.2	เกิน 2500 (และ > 36 สัปดาห์)	29.0-30.8
1501-2500	31.6-33.6	2-3 สัปดาห์	
เกิน 2500 (และ > 36 สัปดาห์)	30.7-33.5	ต่ำกว่า 1500	32.2-34.0
36-48 ชั่วโมง		1501-2500	30.5-33.0
ต่ำกว่า 1200	34.0-35.0	3-4 สัปดาห์	
1200-1500	33.0-34.1	ต่ำกว่า 1500	31.6-33.6
1501-2500	31.4-33.5	1501-2500	30.0-32.7
เกิน 2500 (และ > 36 สัปดาห์)	30.5-33.3	4-5 สัปดาห์	
48-72 ชั่วโมง		ต่ำกว่า 1500	31.2-33.0
ต่ำกว่า 1200	34.0-35.0	1501-2500	29.5-32.2
1200-1500	33.0-34.0	5-6 สัปดาห์	
1501-2500	31.2-33.4	ต่ำกว่า 1500	30.6-32.3
เกิน 2500 (และ > 36 สัปดาห์)	30.1-33.2	1501-2500	29.0-31.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การพิจารณาว่าทารกสามารถออกจากตู้อบเด็กมาอยู่ในอุณหภูมิห้องได้ พิจารณาจากการสังเกต อุณหภูมิคู่อบที่ทารกต้องการเพื่อรักษาอุณหภูมิกายให้ปกติ หากสามารถลดอุณหภูมิคู่อบลงจนใกล้เคียง กับอุณหภูมิห้องที่ทารกจะอยู่หลังออกจากตู้อบ และทารกยังมีอุณหภูมิกายปกติ ให้นำทารกออกจาก ตู้อบ สวมเสื้อผ้าและห่มผ้า ต่อจากนั้นให้ติดตามอุณหภูมิกายอย่างใกล้ชิด โดยวัดอุณหภูมิกายทุกครั้ง ชั่วโมง จนแน่ใจว่าทารกสามารถรักษาอุณหภูมิกายได้เอง จึงติดตามอุณหภูมิกายตามปกติ

## 2.2 ระบบควบคุมของตู้อบเด็กทารกแรกเกิด

ผังการทำงานหลักของตู้อบเด็กทารกแรกเกิด แสดง ได้ดังนี้



รูปที่ 2.1 บล็อกโคอะแกรมของระบบ

ตู้อบเด็กทารกแรกเกิด ประกอบด้วย

- ส่วนตัวประมวลผลควบคุม

เป็นส่วนที่ทำหน้าที่ควบคุมการทำงานทุกส่วนของตู้อบเด็กทารกแรกเกิด ตั้งแต่การรับข้อมูล จากส่วนอินพุทของตัวผู้เข้ามายังตัวประมวลผล และทำการส่งข้อมูลที่ทำการประมวลผลแล้วส่งออกไปยังส่วนแสดงผล

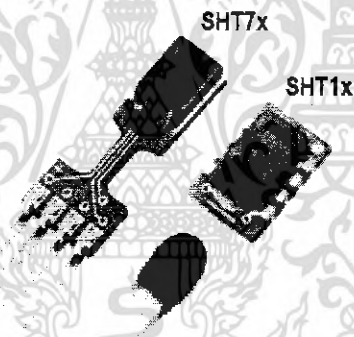
- ส่วนวัดอุณหภูมิและความชื้น

ตรวจจับค่าอุณหภูมิและความชื้นโดยใช้เซนเซอร์ SHT15 ซึ่งสามารถวัดได้ทั้งอุณหภูมิ และความชื้นในตัวเดียวกัน โดยให้เอาท์พุทเป็นสัญญาณดิจิตอลส่งผ่านมายังตัวประมวลผลเพื่อประมวลผล

- ส่วนควบคุม  
รับการตั้งค่าอุณหภูมิที่กำหนดการทำงานให้กับตู้อบเด็กทารกแรกเกิด เพื่อส่งไปยังตัวประมวลผลควบคุมการทำงานของตู้อบ
- ส่วนแสดงผล  
รับค่าที่ได้จากส่วนประมวลผลออกไปแสดงทาง LCD Module

### 2.2.1 เซนเซอร์วัดอุณหภูมิและความชื้น

ส่วนวัดผลทางอุณหภูมิจะวัดค่าอุณหภูมิโดยใช้เซนเซอร์ SHT15 ซึ่ง SHT15 สามารถใช้เป็นเซนเซอร์วัดได้ทั้งอุณหภูมิและความชื้น มีตัวแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอล (A/D) ภายในตัว จึงให้ค่าเอาต์พุตเป็นข้อมูลดิจิตอล ซึ่งค่าที่ได้ต้องนำมาผ่านการคำนวณตามสมการมาตรฐานของอุปกรณ์ จึงจะได้ค่าเอาต์พุตที่แท้จริงเป็นค่าของอุณหภูมิและความชื้น

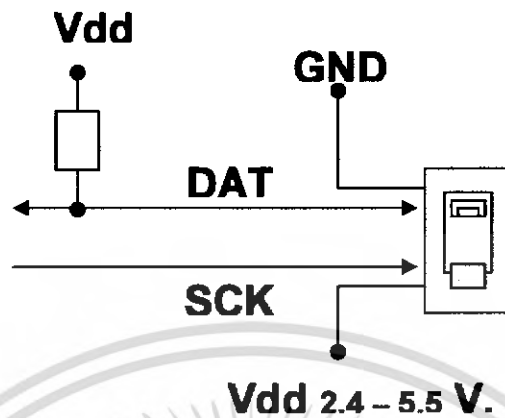


รูปที่ 2.2 รูป IC SHT1X และ SHT7X

ในโครงการนี้เราเลือกใช้ SHT15 มีคุณสมบัติดังนี้

- มี Package แบบ LCC (Leadless Chip Carrier)
- สามารถวัดได้ทั้งอุณหภูมิและความชื้น
- สามารถวัดอุณหภูมิได้ตั้งแต่ 0-125°C ความละเอียดในการวัด 0.1°C
- สามารถวัดความชื้นสัมพัทธ์ได้ตั้งแต่ 1-99.9%RH ความละเอียดในการวัด 0.1%RH
- ใช้แหล่งจ่ายไฟ +5 V กินกระแสต่ำ
- ใช้สัญญาณในการควบคุม 2 เส้น คือ DATA และ CLOCK
- มีความแม่นยำในการวัดอุณหภูมิที่ 0.5°C และความชื้นที่ 2.0%RH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงลักษณะของ SHT15

ตารางที่ 2.2 แสดงคุณสมบัติ DC ของ SHT15

Parameter	Conditions	Min.	Type	Max.	Units.
Power supply DC		2.4	5	5.5	V
Supply current	Measuring		550		$\mu\text{A}$
	Average	2 <sup>2</sup>	28 <sup>3</sup>		$\mu\text{A}$
	Sleep		0.3	1	$\mu\text{A}$
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	$\mu\text{A}$
Input peak current	On			4	mA
	Tristated (off)		10		$\mu\text{A}$

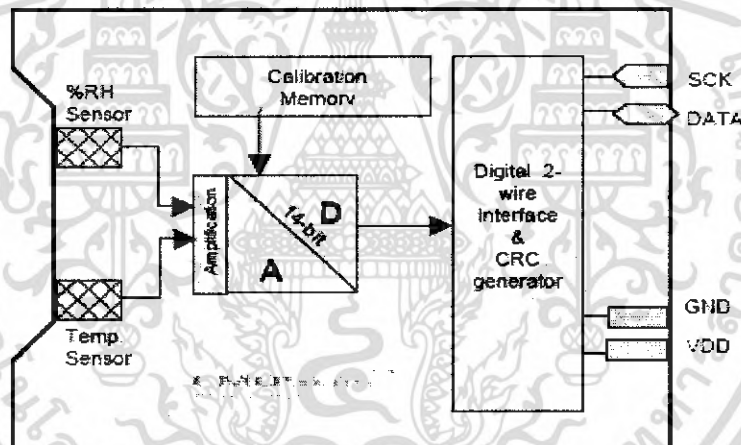
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 แสดงขาสัญญาณของ SHT15

Pin	Name	Comment
1	GND	Ground
2	DATA	Serial data , bi-directional
3	SCK	Serial clock , input
4	VDD	Supply 2.4 – 5.5 V
	NC	Remaining pins must be left unconnected

ลักษณะการทำงานของ SHT15 จะแสดงได้ดังรูปที่ 2.4

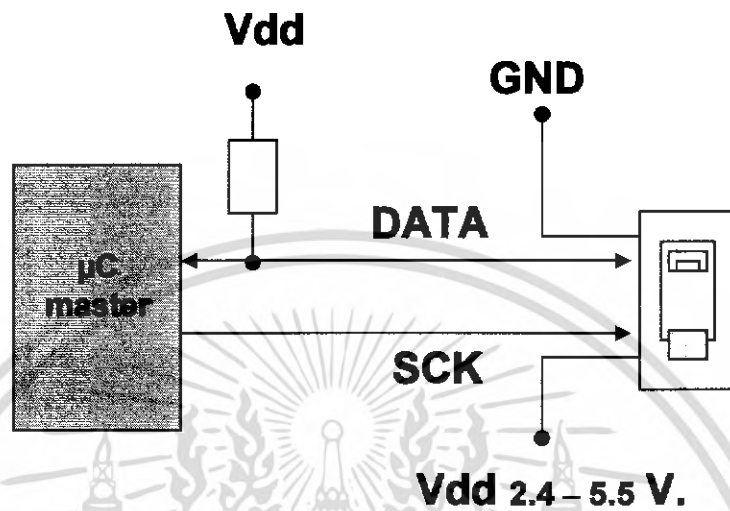
### Block Diagram



รูปที่ 2.4 แสดง Block Diagram ของ SHT15

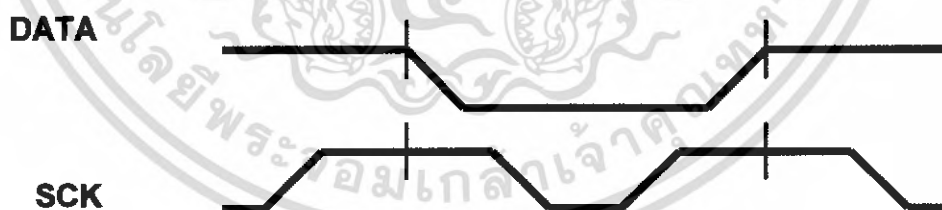
- การต่อขา Vcc กับ GND ต้องต่อไฟเลี้ยงให้อยู่ระหว่าง 2.4- 5.5 V แล้วหลังจาก ที่จ่ายไฟเข้าที่ตัว SHT15 แล้ว SHT15 จะใช้เวลาประมาณ 11ms เพื่อเข้าสู่โหมด Sleep ดังนั้นต้องส่งข้อมูลก่อนที่ IC จะเข้าสู่โหมด Sleep ในการต่อ Vcc กับ GND ควรที่จะต่อ C 100 nF คร่อมระหว่างขา Vcc กับ GND
- การต่อขา DATA และขา SCK จะต่อแบบ Serial Interface (Bi-directional 2 –wire) ซึ่งจะเป็นการต่อในลักษณะที่คล้ายกับ I2C แต่ไม่เหมือน I<sup>2</sup>C โดยทั่วไป

การนำไปใช้งานจะต้องวงจรลักษณะดังรูป 2.5



รูปที่ 2.5 แสดงการต่อ SHT15 กับ ไมโครคอนโทรลเลอร์

- การต่อขา SCK จะเป็นการต่อแบบตรงระหว่างไมโครคอนโทรลเลอร์กับ SHT15
- การต่อขา DATA จะมีการต่อ Pull-up เพื่อให้ได้สัญญาณที่มีค่าสูง ซึ่งการต่อ Pull-up จะค่อนข้างบ่อยในการใช้ ไมโครคอนโทรลเลอร์ และในช่วงที่ทำการส่ง DATA จำเป็นที่จะต้องทำให้ DATA มีความเสถียรในขณะที่ SCK high ซึ่งแสดง Timing Diagram ได้ดังรูป 2.6

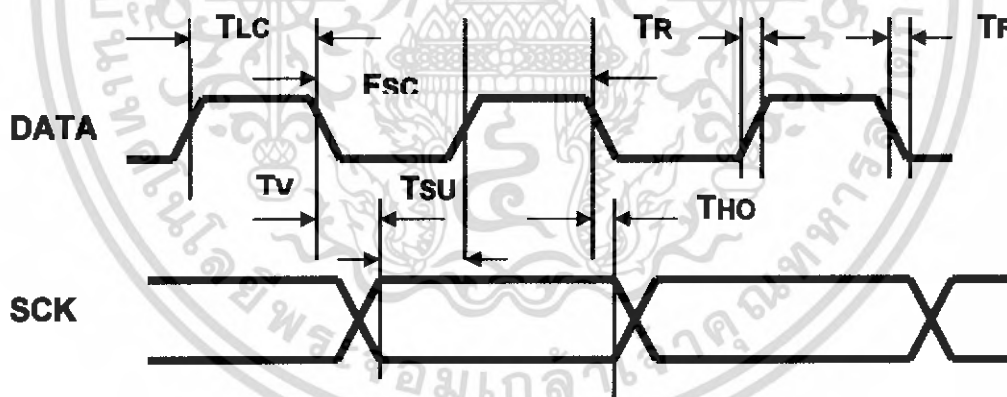


รูปที่ 2.6 แสดง Timing Diagram ในช่วงการส่งข้อมูลของ SHT15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.4 แสดงคุณสมบัติของสัญญาณ SCK และ DATA

	Parameters	Conditions	Min.	Type.	Max.	Units.
F <sub>SCK</sub>	SCK frequency	VDD > 4.5 V			10	MHz
		VDD < 4.5 V			0	MHz
T <sub>RFO</sub>	DATA fall time	Output load 5 pF	3.5	10	20	ns
		Output load 100 pF	30	40	200	ns
T <sub>CLX</sub>	SCK h/l time		100			ns
T <sub>V</sub>	DATA valid time			250		ns
T <sub>SU</sub>	DATA setup time		100			ns
T <sub>HO</sub>	DATA hold time		0	10		ns
T <sub>R</sub> /T <sub>F</sub>	SCK rise/fall time			200		ns



รูปที่ 2.7 แสดงคุณสมบัติของ Timing Diagram ของสัญญาณ DATA และ SCK ของ SHT15

### 2.2.2 ส่วนแสดงผลทาง LCD Module

เป็นจอแสดงผลแบบผลึกเหลวซึ่งเป็นสารที่รวมตัวกันอย่างได้สัดส่วนระหว่างของเหลวกับผลึก LCD ซึ่งมีข้อดีหลายประการ

- กินพลังงานน้อย ต้องการกำลังงานน้อย แรงดันต่ำ

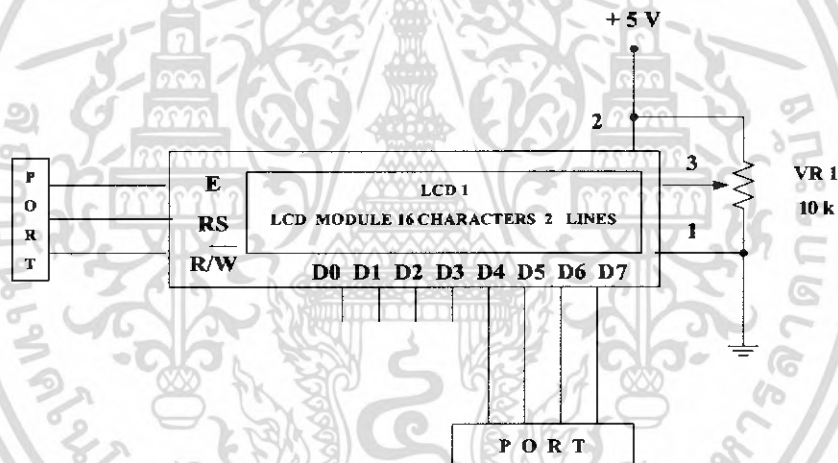
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เชื้อถือได้ และใช้งานได้ในช่วงอุณหภูมิที่กว้าง
- มีอายุการใช้งานยาวนาน
- ง่ายที่จะทำการควบคุมการแสดงผลข้อมูล
- ใช้งานได้กว้างขวาง

ซึ่งด้วยเหตุผลดังกล่าวจึงทำให้ LCD กลายเป็นส่วนประกอบที่สำคัญของอุปกรณ์ต่างๆ ที่ผลิตออกสู่ท้องตลาดในยุคปัจจุบัน

อุปกรณ์อิเล็กทรอนิกส์ในส่วนของแสดงผลข้อมูล ในปัจจุบัน LCD ได้เข้ามาแทนที่จอภาพแบบ CRT (Cathode Ray Tube) ในการแสดงผล

ในโครงการนี้จะใช้ตัวแสดงผลแบบ LCD Module แสดงผลตัวอักษร (Characters) ขนาด 16 X 2 (ขนาด 16 ตัวอักษร 2 บรรทัด) การต่อวงจรเพื่อใช้งาน แสดงดังรูปที่ 2.8



รูปที่ 2.8 แสดงการเชื่อมต่อ LCD Module กับ Micro controller

### 2.2.3 ส่วนควบคุมอุณหภูมิและความชื้น

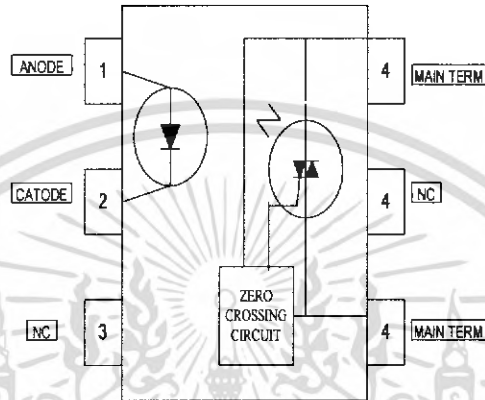
ขดลวดความร้อน(Heater) ที่นำมาใช้จะกินกำลังงาน 300 วัตต์ การทำงานของขดลวดจะเป็นการควบคุมทางเฟส แต่ตอนแรกจะใช้หัวแร้งที่กินกำลังงาน 20 วัตต์มาทำการทดลองแทนขดลวด โดยระบบการทำงานของตัวไมโครคอนโทรลเลอร์คือส่งสัญญาณไปควบคุมเฟสของการทำงานขดลวดความร้อน หรือหัวแร้ง ซึ่งวงจรควบคุมการทำงานของขดลวดความร้อนประกอบไปด้วย

1. MOC3021
2. BTA12 600B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**MOC3021**

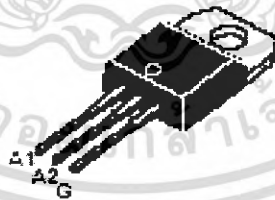
เป็นอุปกรณ์ Isolation ป้องกันแรงดันไฟกระแสสลับไหลเข้าวงจรในส่วนควบคุม



รูปที่ 2.9 แสดงลักษณะของ MOC3021

**BTA12 600B**

เป็นอุปกรณ์ที่ใช้ควบคุมการเปิดการทำงานของวงจรกระแสสลับ ในการควบคุมเฟสการทำงาน ของขดลวดความร้อน ซึ่งใช้คู่เป็นวงจรร่วมกับ MOC 3021 Zero crossing

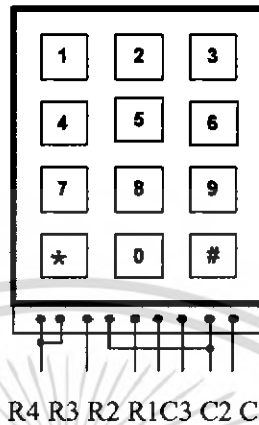


TO-220AB Insulated (BTA12)

รูปที่ 2.10 แสดงอุปกรณ์ BTA12 600B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

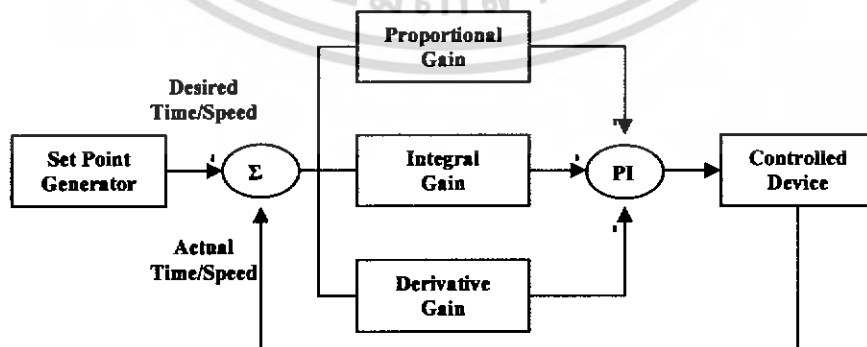
## 2.2.4 ส่วนทำงานผ่านคีย์บอร์ด



รูปที่ 2.11 แสดงการเชื่อมต่อ Keypad กับ Micro controller

## 2.2.5 ระบบควบคุมแบบอัตโนมัติชนิด PID

ระบบควบคุมแบบอัตโนมัติเป็นระบบควบคุมแบบป้อนกลับที่ใช้กันอย่างแพร่หลาย ซึ่งเป็นระบบควบคุมขั้นสูงระบบนี้จะให้ความถูกต้องที่สูงกว่าในกระบวนการทำงาน บางระบบจะสามารถทำได้โดยเพียงใช้ Proportional (P) หรือ Proportional Integral (PI) ก็เพียงพอสำหรับระบบแล้ว แต่เพื่อให้ได้ระบบที่สร้างขึ้นแล้วใกล้เคียงกับความเป็นจริงมากที่สุด โดยระบบส่วนมากจะต้องมีความสามารถในการคาดการณ์สมรรถภาพของระบบ ซึ่งระบบดังกล่าวสามารถทำได้โดยเพิ่ม Derivative (D) เข้าไปในระบบ Derivative (D) ที่เพิ่มเข้าไปในระบบป้อนกลับ จะป้อนกลับค่าเอาต์พุตที่กลับมาอยู่ที่ Loop PI ซึ่งจะกลายมาเป็นระบบที่เรียกว่า Proportional Integrate Derivative (PID) ระบบนี้จะให้ความแม่นยำและความถูกต้องสูงเหมาะกับระบบหรือกระบวนการที่มีความซับซ้อนสูงในยุคปัจจุบัน

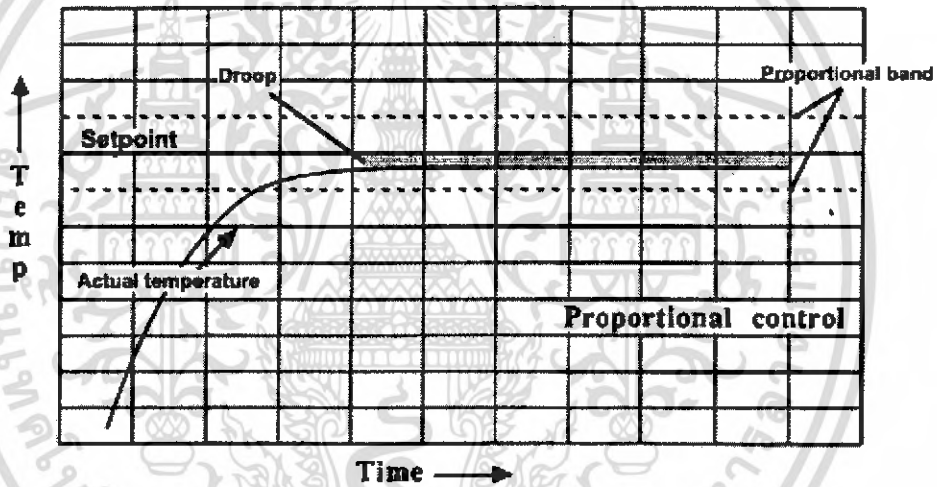


รูปที่ 2.12 แสดงระบบควบคุมแบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Proportional (P)

ระบบรับสัญญาณเข้ามาและเมื่อผ่านระบบก็จะถูกป้อนกลับมายังตัวคอนโทรลเลอร์ ซึ่งจะทำให้เกิดค่าผิดพลาด (Error) คือผลต่างระหว่างเอาต์พุตกับจุด Set point จากการรับสัญญาณจากภายนอกเข้ามาในระบบทำให้เกิด Displacement error โดย Proportional filter จะทำหน้าที่ลดค่าผิดพลาดนี้ อัตราเร็วในการเข้าสู่จุด Set point จะเป็น สัดส่วนเชิงเส้นกับค่าผิดพลาด การปรับค่าอัตราส่วนนี้ทำได้โดยปรับค่า Proportional gain คือค่า  $K_p$  นั่นเอง ซึ่งถ้าค่า  $K_p$  ต่ำอัตราเร็วในการเข้าสู่จุด Set point จะต่ำ ทำให้ผลตอบสนองของระบบช้า แต่ถ้าค่า  $K_p$  สูงอัตราเร็วในการเข้าสู่จุด Set point สูง ผลตอบสนองของระบบทำให้เกิด Overshoot และ Undershoot ขึ้น

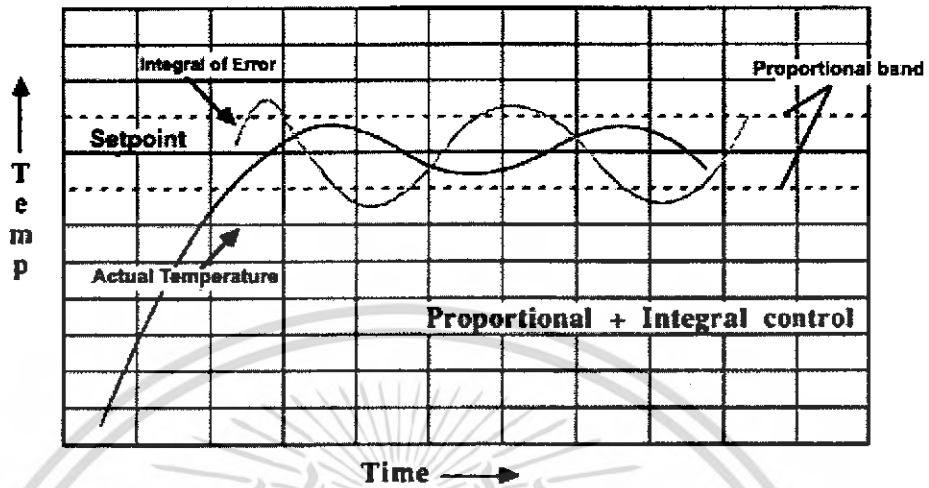


รูปที่ 2.13 แสดงการเข้าสู่จุด Set point ของ Proportional (P)

Integration (I)

การกำจัด Tracking error ค่า Integral จะเป็นสัดส่วนโดยตรงกับค่า Tracking error และเพิ่มอย่างเป็นเชิงเส้นกับเวลา การปรับค่าสามารถปรับได้โดยการปรับ ค่า Integral gain คือค่า  $K_i$  ซึ่งถ้าค่า  $K_i$  มากจะเกิดการชดเชยอย่างรวดเร็ว แต่จะเพิ่ม Overshoot และค่า Ringing ปกติค่า  $K_i$  ควรตั้งให้มีค่าสอดคล้องเพื่อทำให้คุณสมบัติของทั้ง 3 ระบบขอเซชซึ่งกันและกัน

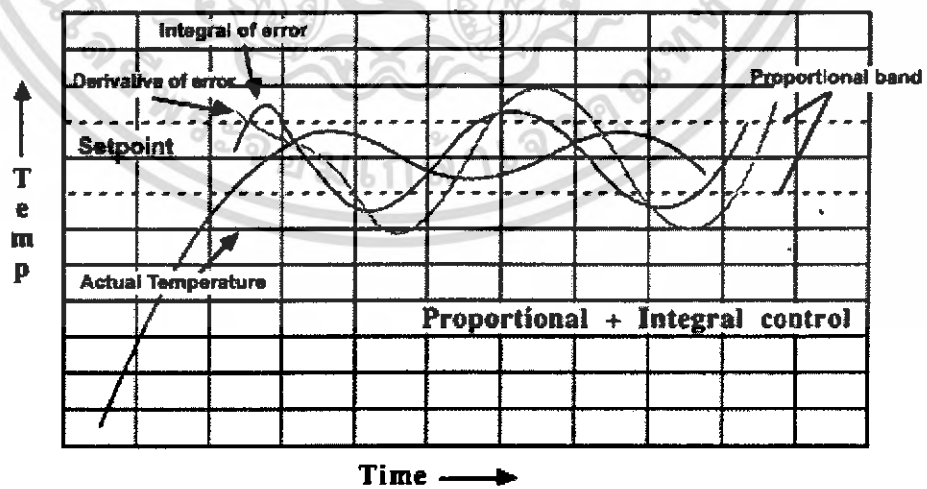
73130



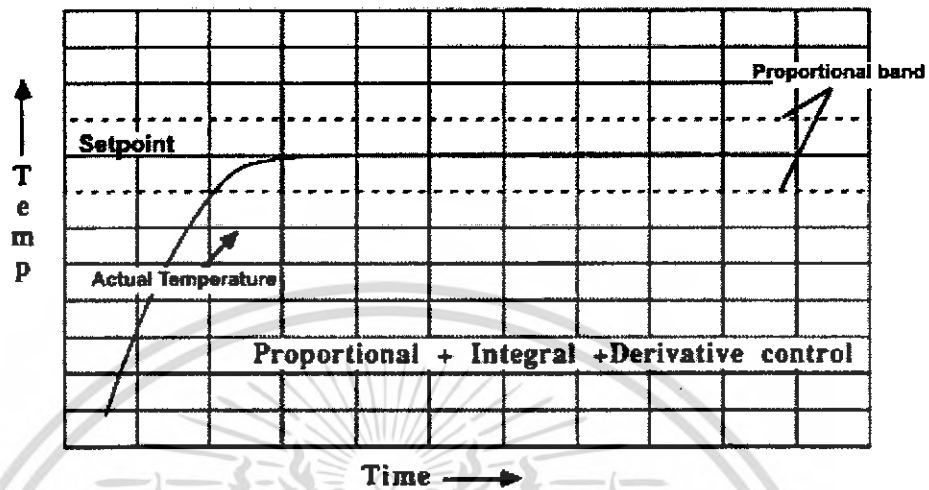
รูปที่ 2.14 แสดงการเข้าสู่จุด Set point ของ Integration (I)

Derivative (D)

การกำจัด Oscillation ของระบบและลด Overshoot กับ Ringing สามารถปรับค่าได้โดยการปรับค่า Derivative gain คือค่า  $K_d$  ซึ่งค่า  $K_d$  จะเป็นสัดส่วน โดยตรงกับอัตราการเปลี่ยนแปลงของ Tracking error ค่า  $K_d$  จะทำให้เกิดเสถียรภาพของทั้งระบบและยังเพิ่มความสามารถในการคาดคะเนผลตอบสนองของระบบ และยังสามารถทำให้เกิดความถูกต้องก่อนที่จะเกิด Tracking error ได้ ความสามารถในการคาดการณ์นี้จะเกิดขึ้นได้จากผลตอบสนอง ซึ่งเกิดจากการกระทำของระบบในช่วงเวลานั้นๆ



รูปที่ 2.15 แสดงการเข้าสู่จุด Set point ของ Derivative (D)



รูปที่ 2.16 แสดงการเข้าสู่จุด Set point ของ Proportional + Integration + Derivative (PID)

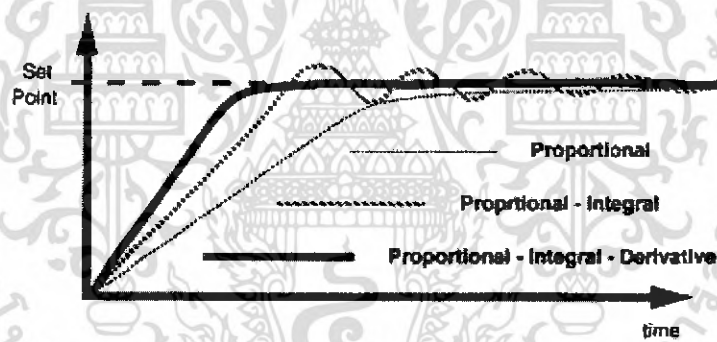


Figure 2 - Control Comparison

รูปที่ 2.17 แสดงการเปรียบเทียบการตอบสนองของระบบควบคุมชนิด

Proportional (P), Proportional + Integration (PI),

Proportional + Integration + Derivative (PID)

จากรูปที่ 2.17 แสดงการเปรียบเทียบการตอบสนองของระบบควบคุมที่เกิดขึ้น

- ระบบควบคุมชนิด Proportional (P) => จะมีผลตอบสนองที่ช้าสุด
- ระบบควบคุมชนิด Proportional + Integration (PI) => rise time ที่เร็วขึ้นแต่จะทำให้เกิดการ Oscillation ก่อนที่จะเข้าสู่จุด Set point

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบบควบคุมชนิด Proportional + Integration + Derivative (PID) => จะให้ rise time ที่เร็วที่สุด ความสามารถในการเข้าสู่จุด Set point เร็วและใกล้เคียงกับ ค่า Set point จริงมากที่สุด

### PID Controller สำหรับสัญญาณดิจิทัล

สมการควบคุมสัญญาณแบบ PID เป็นดังนี้

$$M(e(t)) = K_p \cdot e(t) + K_i \int e(t)dt + K_d \cdot (de(t)/dt)$$

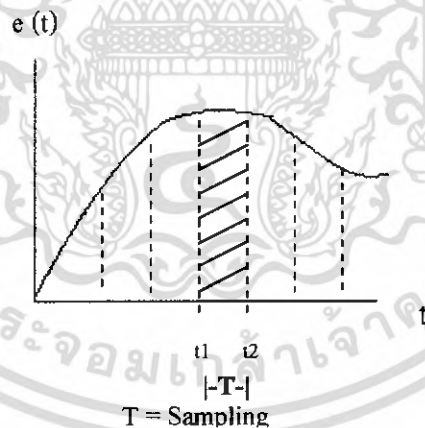
โดยที่  $e(t) = SP - PV(t) = \text{error}$

เมื่อ  $SP = \text{set point}$

$PV(t) = \text{ค่าที่วัดได้ในขณะนั้น}$

จะเห็นได้ว่ามีลักษณะเป็นสัญญาณแบบอนาลอก(Analog) แต่เนื่องจากสัญญาณที่ตัวไมโครคอนโทรลเลอร์นำมาคำนวณ เป็นสัญญาณดิจิทัล ดังนั้นสมการในการคำนวณ PID เดิม จึงไม่สามารถนำมาใช้ได้โดยตรง สามารถพิจารณาได้เป็น 3 ส่วน ดังนี้

$$M(t) = M_p(t) + M_i(t) + M_d(t)$$



รูปที่ 2.18 ความสัมพันธ์ระหว่าง  $e(t)$  กับ  $t$

เมื่อพิจารณาตามรูป 2.18 จะเห็นว่าค่าในส่วนของ Proportion Controller จะคงเดิม คือ

$$M_p(t_n) = K_p \cdot e(t_n)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของ Integral Controller โดยพิจารณาตามรูป จะเห็นว่าค่าในส่วนของ  $\int e(t)dt$  ก็คือพื้นที่ใต้กราฟนั่นเอง เมื่อกำหนด  $T$  เป็นคาบในการสุ่มสัญญาณ มีค่าน้อยมากๆแล้ว จะคิดพื้นที่เป็นสี่เหลี่ยมผืนผ้าได้ นั่นคือ

$$Mi(t_n) = Ki \int e(t)dt = T \cdot Ki \cdot e(t_n) + Mi(t_{n-1})$$

ค่าของ Derivative Controller จากรูป ค่าในส่วนของ  $de(t)/d(t)$  ก็คือความชัน (Slope) ของกราฟนั่นเอง เพราะฉะนั้น

$$Md(t_n) = Kd[e(t_n) - e(t_{n-1})]/T$$

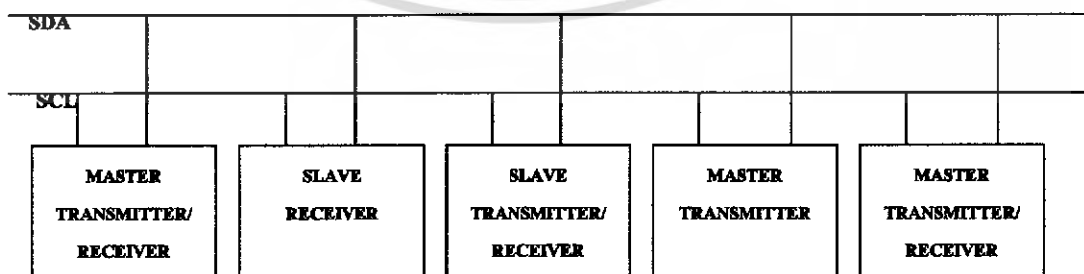
เมื่อรวมเข้าด้วยกัน จะได้ว่า สัญญาณควบคุมที่เป็นดิจิทัล สามารถคำนวณได้จากสมการ

$$M(t) = Kp \cdot e(t_n) + T \cdot Ki \cdot e(t_n) + Kd[e(t_n) - e(t_{n-1})]/T + Mi(t_{n-1})$$

สมการนี้จะอยู่ในส่วนของโปรแกรมที่ใช้ในการคำนวณหา Manipulated Value จากค่าผิดพลาด โดยมีความสัมพันธ์ของคาบเข้ามาเกี่ยวข้อง นั้นแสดงว่า ในการทำงานจริง คาบที่ใช้ในการคำนวณของคอนโทรลเลอร์ จะต้องเท่ากับคาบในการนำสัญญาณ Input เข้ามาของ A/D จริงๆ มิฉะนั้นค่าที่คำนวณออกมาจะผิดพลาด และอาจทำให้ระบบไม่เสถียร

## 2.2.6 การส่งข้อมูลแบบ I<sup>2</sup>C ของ EEPROM

ระบบ I<sup>2</sup>C ย่อมาจาก inter-IC Communication ซึ่ง พัฒนาโดยห้องวิจัย Phillips เมื่อปี 1980 โดยเป็น ระบบ bus ที่ส่ง ข้อมูล หรือสัญญาณ ระหว่าง IC โดยใช้สายเพียง 2 เส้น โดยสายข้อมูล 2 เส้นนั้นมี สาย SDA (Serial Data line) และ SCL (Serial Clock line) โดยสายทั้งสองนี้จะรับส่งสัญญาณไปได้ทั้ง 2 ทิศทาง (Bi-directional) และวงจรทาง output ของอุปกรณ์ที่ใช้ Bus ระบบนี้จะเป็นแบบวงจร open-drain หรือ open-collector ทำให้ต้องทำการ pull up สัญญาณ เมื่อต้องการใช้งาน และในการใช้งาน ระบบ Bus นี้เราสามารถใส่ สายเพียงสองสาย ต่อ กับ IC หลายๆ ชนิดได้ ดังภาพ



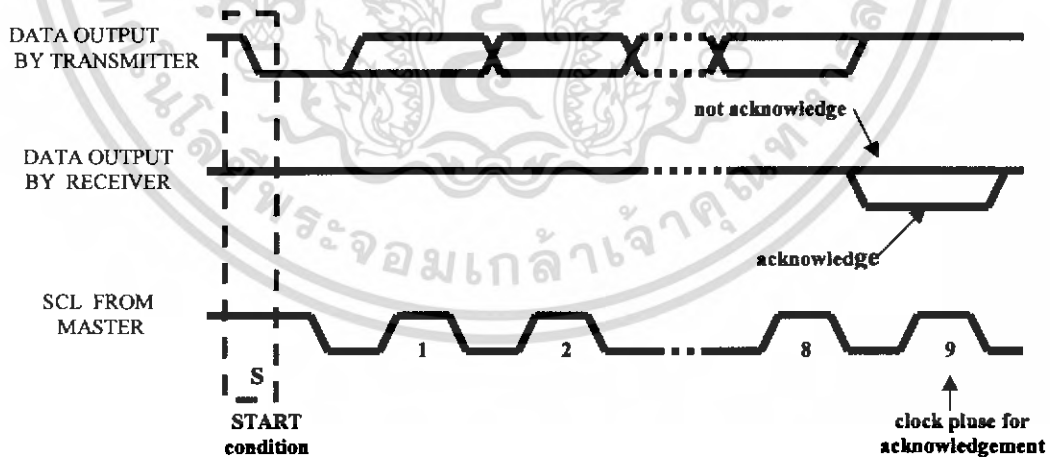
รูปที่ 2.19 การติดต่ออุปกรณ์หลายๆตัวของ I<sup>2</sup>C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเมื่อเราต้องการใช้ งานอุปกรณ์ต่างๆที่ต่ออยู่บนระบบ bus นี้ เราต้อง ทำการแจ้ง สถานะ ให้แก่ระบบ เพื่อแสดงว่า เราเป็นอุปกรณ์ Master ที่ต้องการจะส่งสัญญาณ ให้แก่ อุปกรณ์ Slave โดย สถานะ ต่างๆมีดังนี้

1. bus ว่าง คือสถานะนี้จะไม่มี ค่า logic บน bus ทั้ง 2 สายเลย
2. เริ่มส่งข้อมูล โดย สาย SCL จะมีค่า logic สูง แต่สาย SDA จะมีค่า logic เปลี่ยนจากสูงไปต่ำ เป็นสถานะที่เรียกว่า สถานะเริ่มต้น (start)
3. สถานะหยุด (stop) โดยสาย SCL จะเป็น logic สูง แต่ สาย SDA จะเปลี่ยนจาก logic ต่ำไป สูง
4. เป็นสถานะที่มีการรับส่ง ข้อมูล โดยข้อมูลบน SDA ต้องคงที่ ในขณะที่ SCL เป็น logic สูง และ bit ใน SDA จะเปลี่ยนแปลงได้ต่อเมื่อ SCL มี logic เป็น 0

- สถานะตอบรับ เป็นสถานะที่อุปกรณ์ Mater ส่งข้อมูลออกไปครบ 1 byte แล้ว ตัว master จะส่ง logic สูง ออกมา ถ้าตัว slave ได้รับข้อมูลครบ จะทำให้ logic เป็น 0 แต่ถ้าไม่ถูกต้อง จะรอทำการ รับข้อมูลใหม่



รูปที่ 2.20 แสดงสัญญาณ Acknowledge ของ I<sup>2</sup>C

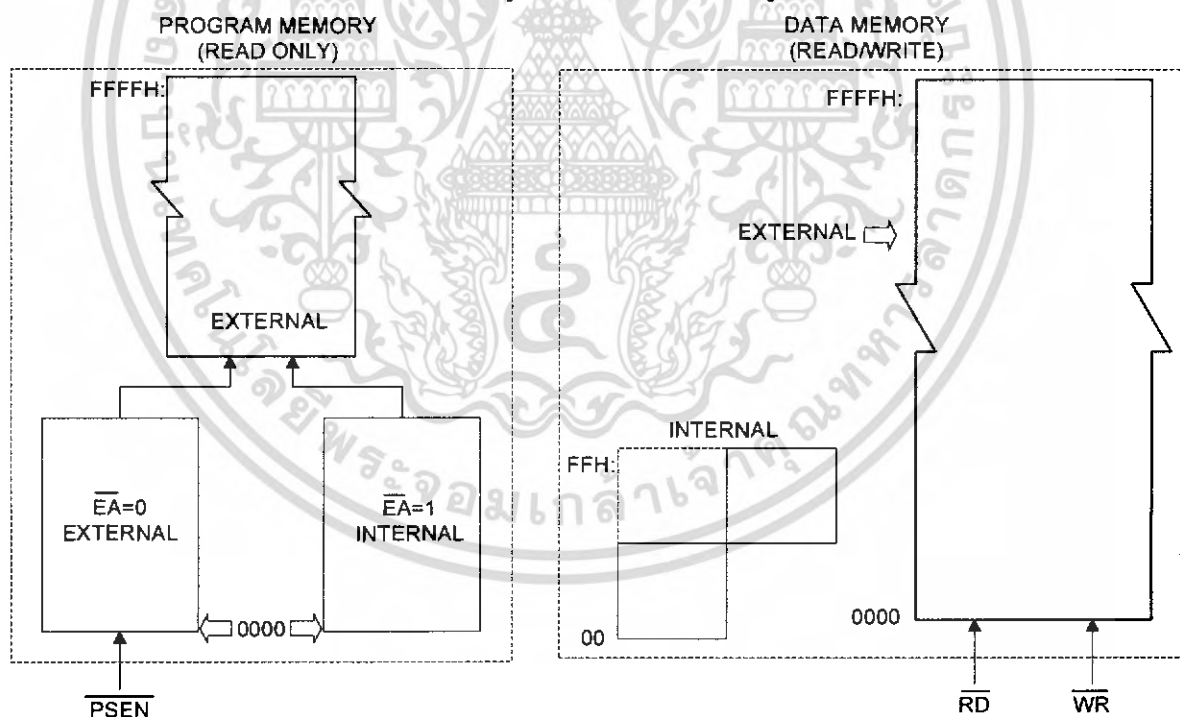


- มีหน่วยความจำ ข้อมูล (RAM) 128 Bytes ภายใน และรองรับการใช้งานของหน่วยความจำ ข้อมูล ได้ ถึง 64K
- มี Port ที่เป็น ได้ทั้ง I/O ทั้งหมด 4 port และสามารถใช้งานได้ในระดับ BIT
- มีส่วน Timer / Counter ขนาด 16 Bit สองชุด สำหรับใช้ในการจับเวลา หรือนับจำนวน
- มี Full duplex UART สำหรับใช้ รับ/ส่ง ข้อมูลแบบอนุกรม รับ Interrupt ได้จาก 6 แหล่งกำเนิด โดยมี 5 ตำแหน่งของ ISR และการ Interrupt โดยสามารถจัดระดับความสำคัญได้ 2 ระดับ
- มีตัวกำเนิดความถี่ Clock ภายใน

### โครงสร้างของหน่วยความจำใน MCS-51

#### Logical separation of program and data memory

สำหรับ Micro controller MCS-51 นั้นถูกออกแบบมาให้มีหน่วยความจำสำหรับเก็บ โปรแกรม (Op-code) และหน่วยความจำสำหรับเก็บข้อมูลที่แยกออกจากกันดังรูป

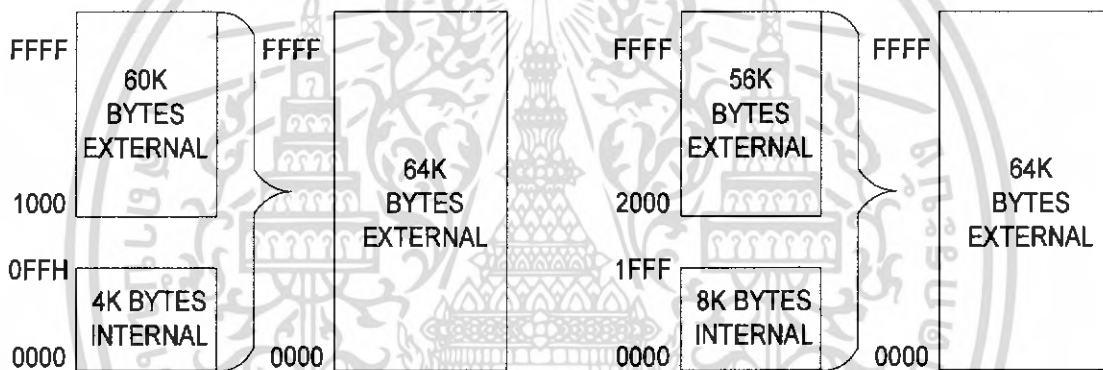


รูปที่ 2.22 โครงสร้างของหน่วยความจำใน MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบของ Data memory ที่แยกออกมา นี้ จะทำให้สามารถเรียกใช้งานได้โดยใช้ Address เพียง 8 Bit เท่านั้น ซึ่งจะทำได้อย่างรวดเร็วใน CPU ที่เป็น 8 Bit แต่การใช้ Address เพียง 8 Bit นี้ จะทำให้อ้างถึงตำแหน่งของหน่วยความจำได้เพียง 256 ตำแหน่งเท่านั้น ( 00h – FFh) ซึ่งก็เพียงพอสำหรับการอ้างถึงตำแหน่งของ Internal Data Memory อย่างไรก็ตามการอ้างถึงตำแหน่ง Data memory โดยใช้ Address แบบ 16 Bit สำหรับ External data memory ก็สามารถทำได้ โดยใช้ DPTR: Data pointer (Data memory address register)

ส่วนของ Program memory จะเป็นหน่วยความจำที่อ่านได้เพียงอย่างเดียว และสามารถมีได้ทั้งหมด 64K ตำแหน่ง

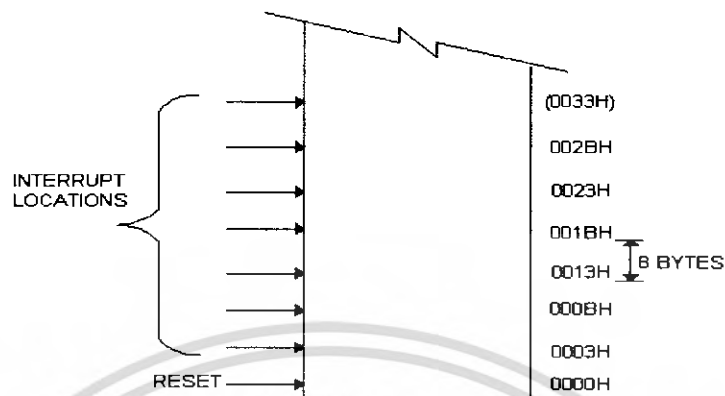


รูปที่ 2.23 โปรแกรม memory ของ MCS 51

สำหรับ MCS-51 ในแบบที่มี Program memory ภายใน ก็จะมีพื้นที่ในการเก็บโปรแกรมภายใน IC เอง 4K, 8K, 16K หรือ 32K Address (ตามเบอร์ของ IC ที่ใช้) สำหรับ Address ที่มากกว่านี้ ก็จะเป็น Program memory ภายนอก ส่วน MCS-51 ที่ไม่มี internal program memory ส่วนของ Program memory ทั้งหมดจะอยู่ภายนอก

สำหรับการอ่าน External program memory นั้น MCS-51 จะใช้ขาสัญญาณ PSEN (Program Store Enable)

สำหรับ Data memory ซึ่งสามารถที่จะอ่านหรือเขียนข้อมูลลงไปได้ ก็จะมี Address ที่แยกออกจาก Program memory และมีตำแหน่งของ External data memory ได้ทั้งหมด 64K Address ในการติดต่อกับ External data memory นั้น MCS-51 จะใช้ขาสัญญาณ RD และ WR Program memory



รูปที่ 2.24 การต่อ External program memory และ External data memory ร่วมกัน

ในกรณีที่ต้องการ ส่วนของ External program memory และ External data memory ร่วมกัน นั้น จะทำได้โดยการนำสัญญาณ PSEN และ RD มารวมกัน โดยใช้ AND gate ก็จะทำให้ได้สัญญาณที่เป็นการอ่าน External Program/Data memory

รูปแสดงส่วนของ Program memory ในตำแหน่งเริ่มต้น ซึ่งเมื่อ CPU เริ่มการทำงานหลังจาก การ Reset ก็จะมีการทำงานตามคำสั่งที่ Address 0000h และสำหรับ Address ที่แสดงต่อมานั้น จะ ตำแหน่งที่อยู่ของ ISR: Interrupt Service Routine โดยการทำงานของ Interrupt ใน MCS-51 เมื่อเกิด สัญญาณ Interrupt เข้ามา มันก็จะกระโดดการทำงานมายัง โปรแกรมในตำแหน่งที่กำหนดนี้ นั่นเอง ตัวอย่างเช่น สำหรับ Interrupt 0 จากภายนอก เมื่อ MCS-51 ได้รับสัญญาณ Interrupt นี้ มันก็จะกระโดด การทำงานมายัง โปรแกรมใน Address 0003h และสำหรับ Interrupt ที่เกิดจาก Timer 0 ก็จะกระโดด การทำงานมายัง โปรแกรมใน Address 000Bh และ Interrupt 0 จากภายนอก ก็จะกระโดดการทำงาน มายัง โปรแกรมใน Address 0013h . ในแต่ละ Address กำหนดให้สำหรับ ISR นั้นจะมีพื้นที่ในการ เก็บโปรแกรมได้ 8 Address ซึ่งถ้า ISR ที่ต้องการ เป็นโปรแกรมที่สั้นๆ ก็จะสามารถใส่เข้าไปได้ แต่ถ้า เป็น ISR ที่ยาวมากแล้วก็จะทำได้ โดยการใช้คำสั่ง Jump ไปยัง โปรแกรมที่ต้องการอีกที

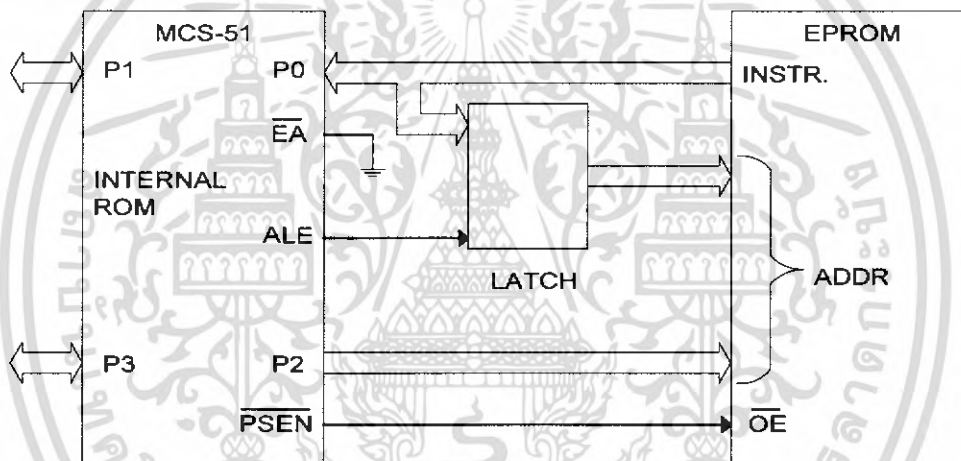
### **External Program Memory**

สำหรับ MCS-51 ที่มี Internal program memory นั้น ผู้ใช้สามารถที่จะเลือกได้ว่า จะใช้งาน ของ Internal program memory นั้น หรือไม่ โดยการต่อของขาสัญญาณ EA : External access เข้ากับ VCC หรือ GND เช่น ถ้า MCS-51 มี Internal program memory 4K (0000h-0FFFh) แล้วต่อขาสัญญาณ EA นี้เข้ากับ VCC การ Fetch คำสั่งที่ Address น้อยกว่า 0FFFh ก็จะได้จาก Internal program memory

และถ้าเป็น Address ตั้งแต่ 1000h ก็ จะเป็นการอ่านจาก External program memory นั้นเอง แต่ถ้าต่อขา EA เข้ากับ GND การ Fetch คำสั่งทั้งหมดจะกระทำกับ External program memory

สำหรับ MCS-51 ที่ไม่มี internal program memory แล้ว ขา EA จะต้องต่อ GND เสมอ สัญญาณ PSEN ซึ่งเป็นสัญญาณที่ใช้ในการ Fetch คำสั่งจาก External program memory นั้น จะไม่ทำงานเมื่อเป็นการ Fetch คำสั่งจาก internal program memory

#### การต่อ External program memory



รูปที่ 2.25 การต่อ External program memory

MCS-51 จะใช้ขาสัญญาณ ของ I/O port 16 bit (Port 0 และ Port 2) มาทำหน้าที่เป็น Bus ของระบบ โดยจะใช้งานของ Port 0 ทำหน้าที่เป็น Address และ Data bus สลับกัน คือส่งค่าของ Low byte ของ Program counter (PCL) ออกมาที่ Port 0 นี้ หลังจากส่งค่าของ PCL ออกมาแล้วมันจะเข้าสู่สถานะ Float เพื่อรอรับคำสั่งที่จะอ่านได้จาก External program memory ในระหว่างที่ ค่าของ PCL ออกมาที่ P0 นี้ สัญญาณ ALE: Address Latch Enable จะไปทำให้ค่าของ PCL ถูกเก็บเข้าที่ LATCH และในเวลา นั้น ค่าของ PCH ก็ถูกส่งออกมาที่ Port 2 เพื่อสร้างเป็น Address ขนาด 16 bit แล้ว สัญญาณ PSEN ก็จะเป็นตัวอ่านข้อมูลจาก Memory ที่ต้องการ

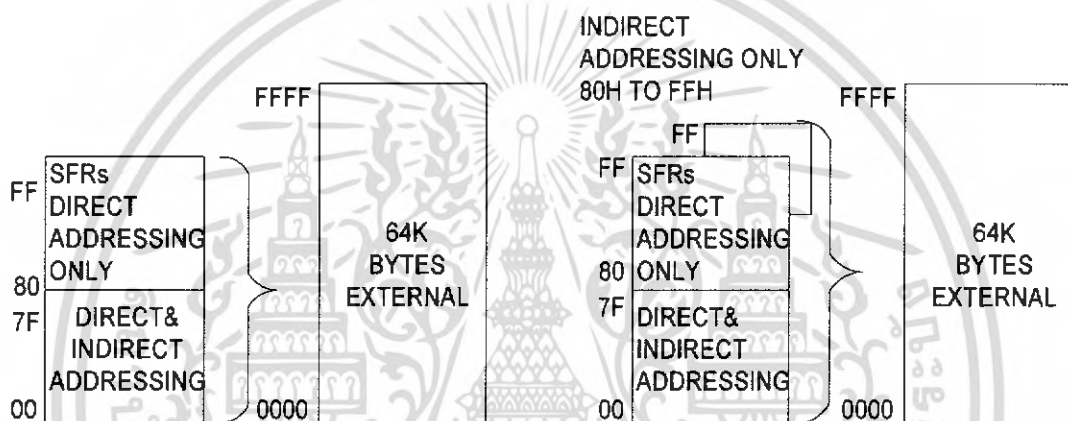
### Data memory

Data memory ของ MSC-51 นั้นจะแบ่งออกเป็น Internal data memory และ External data memory โดยการใช้งานของ Data memory ทั้งสองส่วนนี้จะมี Address ที่แยกจากกันด้วย

MCS-51 with 128 bytes Internal Data memory (ซ้าย)

MCS-51 with 256 bytes Internal Data memory (ขวา)

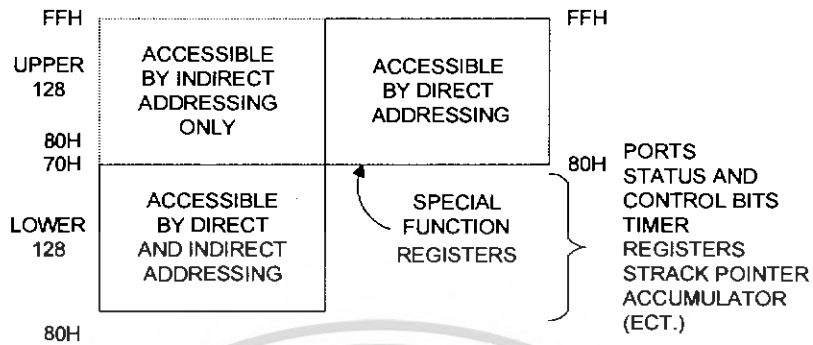
รายละเอียดดังรูปที่ 2.26



รูปที่ 2.26 Data memory ของ MSC-51

### Internal Data Memory

ผังการจัดแบ่งพื้นที่ของ Internal data memory จะเป็นดังรูป ซึ่งมันจะถูกแบ่งออกเป็น สามส่วนด้วยกัน คือ Lower 128, Upper 128 และ SFR และจากการที่ Internal data memory นั้นมีเพียง 256 ตำแหน่งเท่านั้น ทำให้การอ้าง Address สามารถทำได้ โดยใช้เพียง 8 Bit และจากการที่ Internal data memory นั้นมีเพียง 256 ตำแหน่งเท่านั้น ทำให้การอ้าง Address สามารถทำได้ โดยใช้เพียง 8 Bit



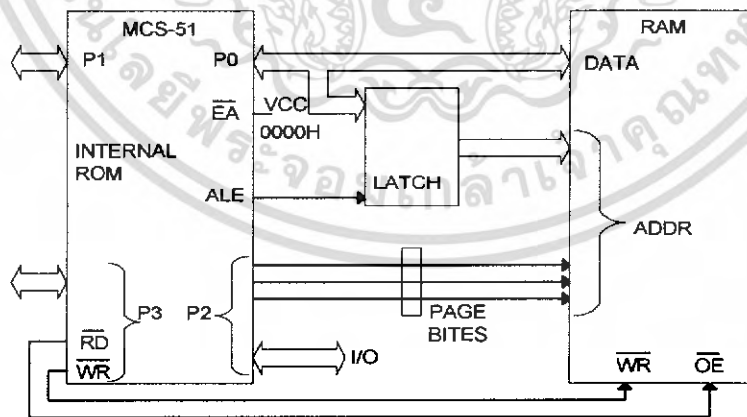
รูปที่ 2.27 Internal Data Memory

แต่ด้วยเทคนิคของการอ้างถึงตำแหน่งข้อมูล (Addressing mode) ทำให้ได้ตำแหน่งของ internal data memory ทั้งหมด 384 bytes

โดยสำหรับข้อมูลตั้งแต่ Address 80h – FFh ถ้าอ้างถึงข้อมูลที่ Address นั้นๆ ด้วยวิธีของ Direct addressing ก็จะได้ข้อมูลที่มาจากคนละส่วน กับการอ้างถึงข้อมูลที่ Address เดียวกันนั้น ด้วยวิธีของ indirect addressing จากรูปจะเห็นได้ว่า Memory ในส่วนของ Upper 128 ก็จะมี Address เดียวกับ Memory ในส่วนของ SFR แต่จะใช้วิธีการเข้าถึงข้อมูลที่แตกต่างกันนั่นเอง

External Data Memory

การต่อใช้งานของ External data memory จะทำได้ดังตัวอย่าง ซึ่งเป็นการต่อใช้งานของ RAM ขนาด 2K โดยให้ MCS-51 นี้ ทำงานจาก internal program memory



รูปที่ 2.28 การต่อใช้งานของ External data memory โดยให้ MCS-51 นี้ ทำงานจาก Internal program memory



**SFR**

รายละเอียดของตำแหน่งข้อมูลต่างๆ ใน SFR จะเป็นดังนี้

**8 Bytes**

F8	IOCON									FF
F0	B									F7
EB										EF
E0	ACC									E7
D8										DF
D0	PSW									D7
C8	T2CON		RCAP2L	RCAP2H	TL2	TH2				CF
C0										C7
B8	IP									BF
B0	P3									B7
A8	IE									AF
A0	P2									A7
98	SCON	SBUF								9F
90	P1									97
88	TCON	TMOD	TL0	TL1	TH0	TH1				8F
80	P0	SP	DPL	DPH					PCON	87

bit addressable

รูปที่ 2.30 รายละเอียดของตำแหน่งข้อมูลต่างๆ ใน SFR

จะเห็นได้ว่าตำแหน่งต่างๆ ใน SFR จะไม่ได้ถูกใช้ทั้งหมด การอ่านข้อมูลจากตำแหน่งที่ยังไม่ได้ กำหนดนี้ จะทำให้ได้ค่าสุ่มออกมา ส่วนการเขียนข้อมูลเข้าไปจะไม่มีผลใดๆ แต่อย่างไรก็ดี เราไม่ควรเขียนข้อมูล '1' เข้าไปในตำแหน่งที่ไม่ได้กำหนดนี้ เนื่องจากมันอาจถูกใช้งานในการควบคุม ฟังก์ชันการทำงานที่จะเพิ่มเข้ามาในอนาคต

**Accumulator**

ACC จะคือ accumulator ซึ่งจะมีตำแหน่งในหน่วยความจำภายในเท่ากับ 0E0H มีขนาด 8 Bit เป็น Register ที่ใช้กันมาก ในรหัสคำสั่งช่วยจำจะใช้อักษร A

**B Register**

จะมีตำแหน่งในหน่วยความจำภายในเท่ากับ 0F0H มีขนาด 8 Bit จะใช้ในการคูณและการหาร เท่านั้น หรืออาจใช้ในการเก็บข้อมูลอื่นๆ ที่ต้องการก็ได้

### Program status word (PSW)

จะมีตำแหน่งในหน่วยความจำภายในเท่ากับ D0H มีขนาด 8 Bit แต่ละ Bit จะบอกถึงสถานะต่างๆ ในการทำงานของ CPU (Flag) ซึ่งแต่ละ Bit ของ PSW สามารถกำหนดให้เป็น 1 หรือ 0 ได้โดยคำสั่ง SETB หรือ CLR B ตามลำดับตำแหน่ง Bit Address 0 ถึง Bit 7 ของ PSW เท่ากับ D0h ถึง D7h

### Stack Pointer

จะมีตำแหน่งในหน่วยความจำภายในเท่ากับ 081H มีขนาด 8 Bit Register นี้ใช้ชี้ตำแหน่งในหน่วยความจำภายใน 8051 ที่จะใช้สร้างเป็น Stack ในการทำงานของ MCS-51 ค่าของ SP นี้จะมีค่าที่เพิ่มขึ้น ก่อนที่จะมีการเก็บข้อมูลเข้าไปด้วยคำสั่ง PUSH หรือ CALL

การกำหนดตำแหน่งของ Memory ที่จะสร้างเป็น Stack นั้นสามารถกำหนดให้เป็นที่อยู่ใดก็ได้ใน internal data memory และเมื่อทำการ Reset ค่าเริ่มต้นของ SP จะค่าเป็น 07h ซึ่งจะทำให้การเก็บข้อมูลในตำแหน่งแรกของ Stack เริ่มที่ Internal data memory ตำแหน่งที่ 08h

### Data Pointer Register

Data Pointer (DPTR) จะอยู่ในตำแหน่งหน่วยความจำภายในเท่ากับ 82h และ 83h DPTR นี้ประกอบไปด้วย Register ขนาด 8 Bit 2 ตัวคือ DPH และ DPL ซึ่ง DPTR นี้ จะใช้ในการชี้ตำแหน่งของข้อมูลของ External data memory แบบ 16 Bit ในการแก้ไขข้อมูลใน Register DPTR จะทำได้ทีละ 16 Bit หรือกระทำทีละ 8 Bit ก็ได้ (DPH, DPL)

### PORT 0 ถึง 3

จะตำแหน่งในหน่วยความจำภายในเท่ากับ 80h, 90h, 0A0h, 0B0h เป็น Register ขนาด 8 Bit การเขียนข้อมูลไปยังหน่วยความจำแต่ละตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ทนั้นๆ ของ MCS-51 ข้อมูลที่เขียนออกไปจะถูก Latch ก้างไว้ที่ Register นี้ และปรากฏแต่ละ Bit ของ Port เช่นถ้าเขียนข้อมูล 18h ไปที่หน่วยความจำตำแหน่ง 80h ก็จะปรากฏ Logic 0001 1000 ที่ขา 7 ถึง 0 ของ Port 0 ในการอ่านข้อมูลจาก Register แต่ละตัวจะเป็นการอ่านสถานะ Logic ที่มีปรากฏอยู่แต่ละขาของ Port นั้นๆ

### Serial Data Buffer (SBUF)

จะตำแหน่งหน่วยความจำภายในเท่ากับ 99H มีขนาด 8 Bit แต่จากโครงสร้างภายในแล้วมันคือ Register 2 ตัวที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรม และอีกตัวหนึ่งสำหรับรับข้อมูลแบบอนุกรมที่เข้ามา ในการเขียนข้อมูลเข้าที่ SBUF มันจะถูกเขียนไปยังที่สำหรับเก็บข้อมูลสำหรับส่ง และเริ่มต้นการส่งข้อมูล ส่วนการอ่านข้อมูลจาก SBUF ก็จะเป็นการอ่านค่าของข้อมูลที่รับเข้ามาได้

### TIMER Register

คู่ของ Register (TH0, TL0) (TH1, TL1) และ (TH2, TL2) ซึ่งอยู่ในตำแหน่งหน่วยความจำภายใน (8Ch, 8Ah) (8Dh, 8Bh) และ (0CDh, 0CCh) ตามลำดับ ซึ่งจะใช้ในการเก็บค่าของการนับแบบ 16 Bit ในการใช้งานเป็น Timer หรือ Counter ใน 80C51 จะมี Timer อยู่ 2 ชุดคือ Timer 0 และ Timer 1 ใน Timer แต่ละชุดจะมี Register ขนาด 8 Bit อยู่ 2 ตัว เพื่อเก็บค่าการนับของ Timer ได้สูงสุดถึง 16 Bit

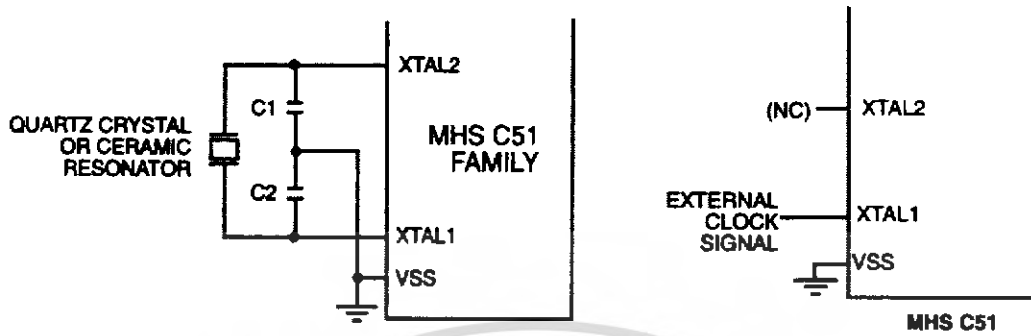
การกำหนดการทำงานของวงจร Timer ในโหมด Timer หรือ Counter ทำได้โดยการกำหนดใน Register TMOD (Timer/Counter Mode Control Register) การทำงานเป็น Timer จะให้ Register ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิลของสัญญาณนาฬิกา ในการให้วงจร Timer ทำงานเป็น Counter คือการใช้ Register THx และ TLx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1

### Control Register

SFR ที่ชื่อ IP, IE จะสำหรับกำหนดรูปแบบการทำงาน และสถานะของการ Interrupt TMOD, TCON, T2CON จะสำหรับกำหนดรูปแบบการทำงานของ Timer/Counter และ SCON จะสำหรับกำหนดรูปแบบการทำงานของ Serial port และ PCON จะสำหรับกำหนดรูปแบบการใช้พลังงานของตัว CPU เอง

### CPU Timing

ใน MCS-51 จะมีส่วนของวงจรกำเนิด Clock อยู่ภายในแล้ว ซึ่งสามารถใช้เป็นสัญญาณนาฬิกาให้กับ CPU ได้ ในการใช้งานของ On-Chip Oscillator นั้น ก็เพียงต่อ Crystal หรือ Ceramic resonator ที่ขา XTAL1 กับ XTAL2 และตัวเก็บประจุ ดังแสดง



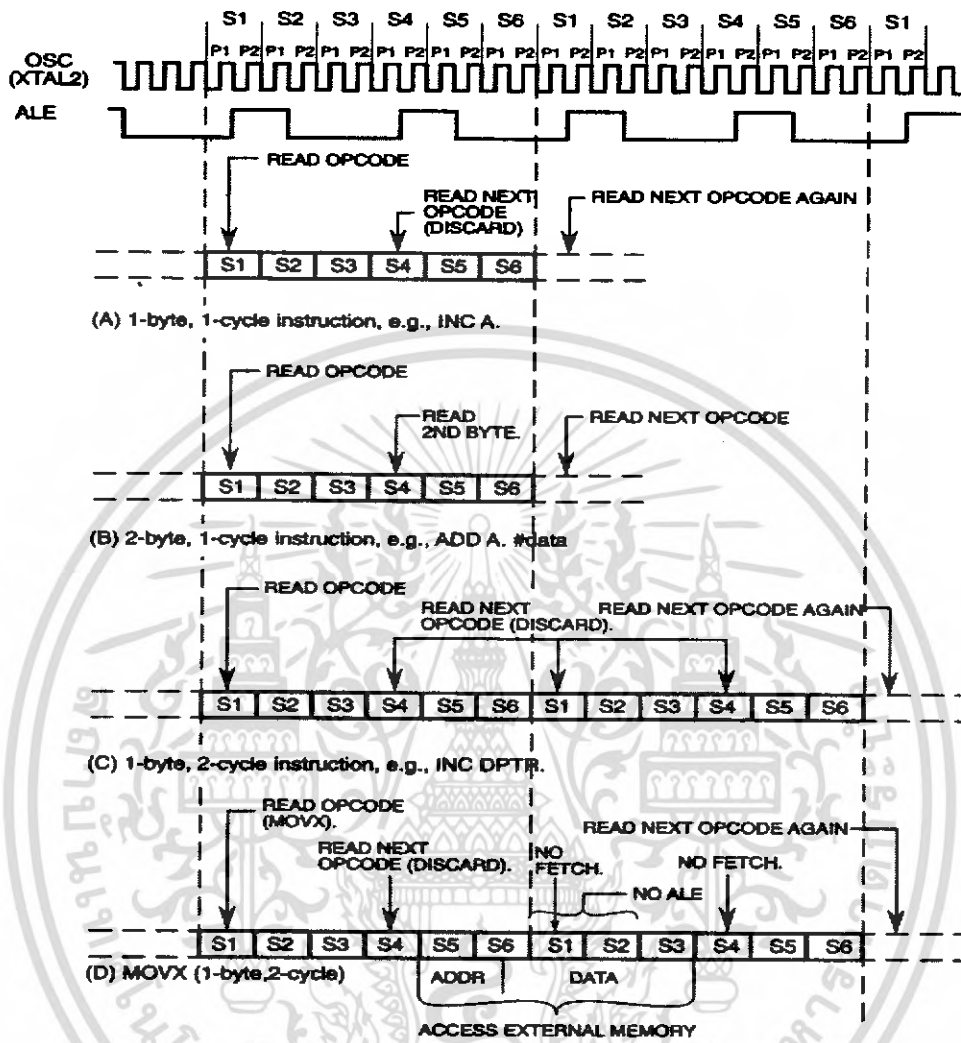
รูปที่ 2.31 การต่อ CPU Timing

อย่างไรก็ดี ถ้าต้องการใช้สัญญาณ Clock จากภายนอก จะทำได้โดยการต่อสัญญาณ Ext. Clock เข้าที่ขา XTAL1

Machine Cycles

สำหรับแต่ละ Machine cycle ของ MCS-51 จะประกอบด้วยการทำงาน 6 States (S1 – S6) ซึ่งในแต่ละ State ของการทำงาน จะใช้เวลา 2 Clocks ดังนั้นถ้าใช้ Clock 12 MHz ก็จะได้เวลาในการทำงานของ 1 Machine cycle คือ 1  $\mu$ S

ดังแสดงในรูป จะเป็นตัวอย่างของการ Fetch / Execute ของคำสั่ง ซึ่งจะในแต่ละ Machine cycle จะมีการ Fetch รหัสคำสั่ง 2 ครั้ง ถึงแม้ว่าคำสั่งนั้นๆ จะเป็นคำสั่งแบบ 1 Byte ก็ตาม ในกรณีที่คำสั่งนั้นๆ ไม่ต้องการข้อมูล Byte ที่สอง CPU ก็เพียงแต่ไม่สนใจข้อมูลที่ Fetch ได้เกินมา และค่าของ PC ก็จะไม่เพิ่มขึ้น

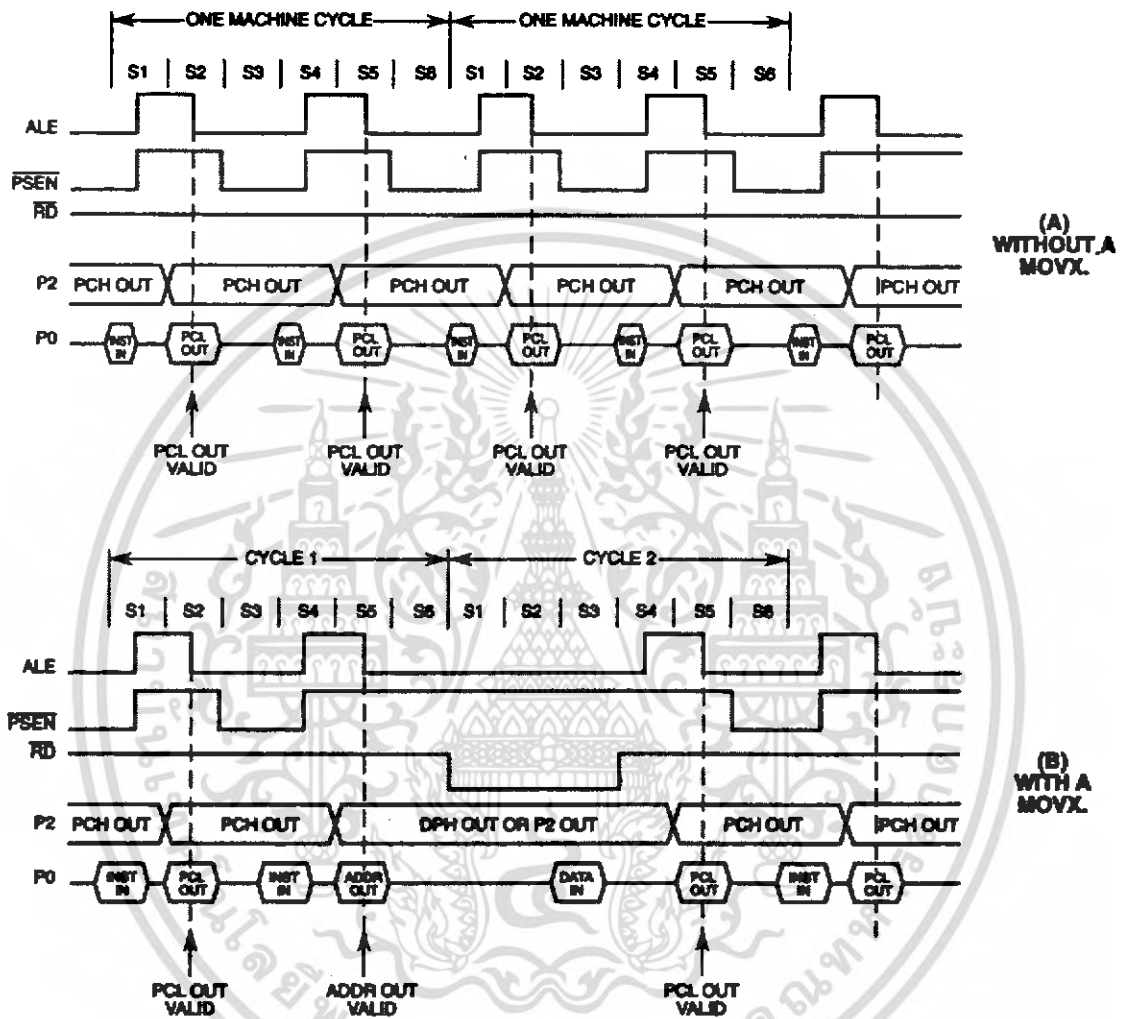


รูปที่ 2.32 Machine Cycles

จากรูป a และ b ซึ่งเป็นคำสั่งที่ใช้การทำงาน 1 Machine cycle ที่ S1 CPU ก็จะอ่าน Op-code ที่ต้องการเข้ามายัง Instruction register และที่ S4 จะมีการ Fetch ครั้งที่สองเกิดขึ้น ซึ่งในรูป a การ Fetch ครั้งที่สองนี้ CPU จะไม่สนใจข้อมูลที่ได้มา ส่วนในรูป b ข้อมูลที่ได้มาก็จะเป็น Byte ที่สองของคำสั่งนั่นเอง

จากรูป c ซึ่งเป็นการทำงานของคำสั่งที่ใช้การทำงาน 2 Machine cycles CPU จะไม่สนใจข้อมูลที่ได้จากการ Fetch 3 ครั้งด้วยกันและในรูป d ซึ่งเป็นการทำงานของคำสั่ง MOVX ซึ่งเป็นการติดต่อกับ External data memory ในกรณีนี้ จะไม่มีการ Fetch เกิดขึ้น เนื่องจาก CPU จะต้องใช้ระบบ BUS เดียวกันนี้ ในการติดต่อกับหน่วยความจำภายนอก

ขบวนการ Fetch / Execute นี้จะเหมือนกัน ไม่ว่าจะเป็นการทำงานของโปรแกรมที่อยู่ใน Internal หรือ External program memory ดังนั้นแล้วมันจะใช้เวลาในการทำงานของคำสั่งต่างๆ เท่ากัน



รูปที่ 2.33 ขบวนการ Fetch / Execute

จากรูป a จะเป็นการแสดงถึงจังหวะการทำงานของสัญญาณต่างๆ ของการ Fetch เมื่อคำสั่งนั้นอยู่ใน External program memory ซึ่งสัญญาณ จะเกิดขึ้น 2 ครั้งต่อ Machine cycle ในรูป b จะแสดงสัญญาณต่างๆ ที่เกิดขึ้นเมื่อกระทำคำสั่งที่มีการติดต่อกับ External data memory การ Fetch จะถูกข้ามไป 2 ครั้ง และจะเห็นว่าจังหวะเวลาของการอ่านข้อมูลจาก External data memory นั้นจะใช้เวลาที่มากกว่าการอ่านข้อมูลจาก Program memory

## บทที่ 3

### ขั้นตอนการดำเนินงาน

สามารถแบ่งขั้นตอนการดำเนินงานได้ดังนี้คือ

#### 3.1 การออกแบบวงจรต่างๆ ภายในตู้ของเด็ก

3.1.1 วงจรแหล่งจ่ายไฟของระบบควบคุม

3.1.2 ส่วนตรวจจับอุณหภูมิที่ใช้ SHT15

3.1.3 วงจรทรานซิสเตอร์และแยกโพล

3.1.4 วงจรตรวจจับสัญญาณชาวน์

3.1.5 วงจรส่วนแสดงผล

3.1.6 วงจรควบคุมการทำงานหลัก

#### 3.2 โปรแกรมการทำงานของตู้ของเด็ก

3.2.1 Flow chart แสดงการทำงานของโปรแกรม

3.2.2 อธิบายโปรแกรมการทำงานของตู้ของเด็ก

#### 3.3 การออกแบบส่วนประกอบภายนอก

3.3.1 ตู้ครอบตู้เด็กและแผ่นรองตู้เด็ก

3.3.2 Chamber

3.3.3 ตู้ฐานรอง

3.3.4 ส่วนแสดงผล

#### 3.1 การออกแบบวงจรส่วนต่างๆ ภายในตู้ของเด็ก

##### 3.1.1 วงจรแหล่งจ่ายไฟกระแสตรง

วงจรจ่ายไฟกระแสตรง มีการรักษาระดับแรงดันคงที่ การออกแบบจะใช้ไอซีแบบ 3 ขา ประกอบด้วยขาอินพุต เอาท์พุท และขากราวนด์ ซึ่งตัวเลขที่บอกเบอร์ไอซีจะเป็นตั้งบอกขนาดของแรงดันเอาท์พุท เบอร์ที่ให้แรงดันไฟบวกคือเบอร์ 78xx และให้แรงดันไฟลบคือเบอร์ 79xx

เบอร์ 7815 แรงดันเอาท์พุท + 15V

เบอร์ 7812 แรงดันเอาท์พุท + 12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เบอร์ 7915 แรงดันเอาต์พุต - 15V

เบอร์ 7912 แรงดันเอาต์พุต - 12V

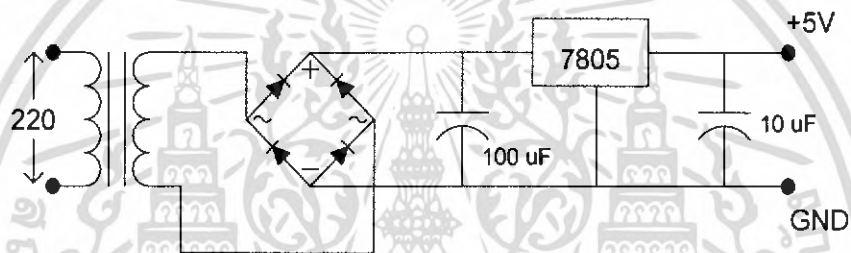
เบอร์ 7809 แรงดันเอาต์พุต + 9V

เบอร์ 7909 แรงดันเอาต์พุต - 9V

เบอร์ 7805 แรงดันเอาต์พุต + 5V

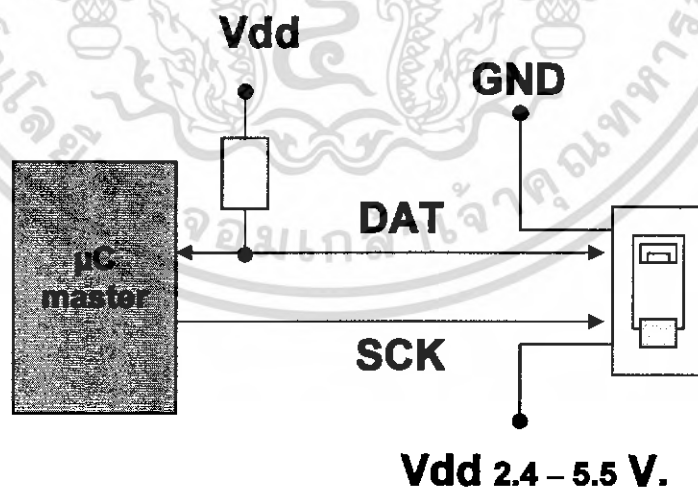
เบอร์ 7905 แรงดันเอาต์พุต - 5V

ขั้นแรกของการออกแบบเพื่อใช้ในการวัดค่าอุณหภูมิและความชื้น ระบบปิดเปิด (ON , OFF) ในส่วนแหล่งจ่ายไฟกระแสตรงจึงใช้เพียงไอซีเบอร์ 7805 เท่านั้น



รูปที่ 3.1 วงจรแหล่งจ่ายไฟของระบบควบคุม

### 3.1.2 ส่วนตรวจจับอุณหภูมิที่ใช้ SHT15



รูปที่ 3.2 ตัวอย่างการต่อใช้งาน SHT15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การต่อขา Vcc กับ GND ต้องต่อไฟเลี้ยงให้อยู่ระหว่าง 2.4- 5.5 V แล้วหลังจากที่จ่ายไฟเข้าที่ตัว SHT15 แล้ว SHT15 จะใช้เวลาประมาณ 11ms เพื่อเข้าสู่โหมด Sleep ดังนั้นต้องส่งข้อมูลก่อนที่ IC จะเข้าสู่โหมด Sleep ในการต่อ Vcc กับ GND ควรที่จะต่อ C 100 nF คร่อมระหว่างขา Vcc กับ GND

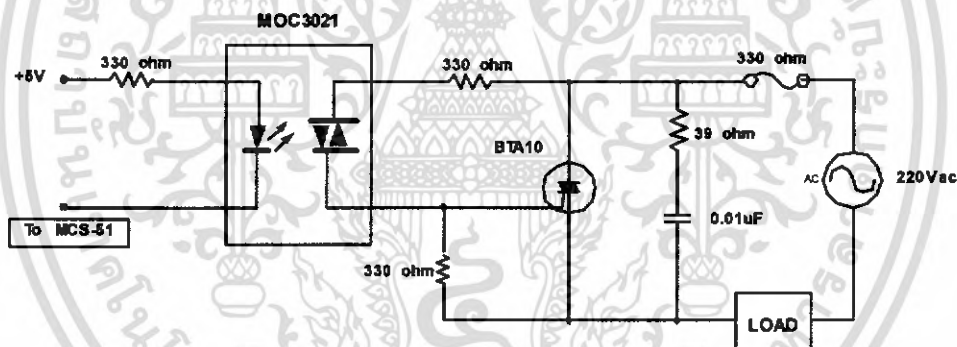
- การต่อขา DATA และขา SCK จะต่อแบบ Serial Interface(Bidirectional 2 –wire) ซึ่งจะเป็นการต่อในลักษณะที่คล้ายกับ I<sup>2</sup>C แต่ไม่เหมือน I<sup>2</sup>C โดยทั่วไป

- การต่อขา SCK จะเป็นการต่อแบบตรงระหว่างไมโครคอนโทรลเลอร์ กับ SHT15

- การต่อขา DATA จะมีการต่อ Pull-up เพื่อให้ได้สัญญาณที่มีค่าสูง ซึ่งการต่อ Pull-up จะต่อกันบ่อยๆ ในการใช้ไมโครคอนโทรลเลอร์ และในช่วงที่ทำการส่ง DATA จำเป็นที่จะต้องทำให้ DATA มีความเสถียรในขณะที่ SCK high ซึ่งแสดง Timing Diagram ได้

### 3.1.3 วงจรทริกและแยกโหนด

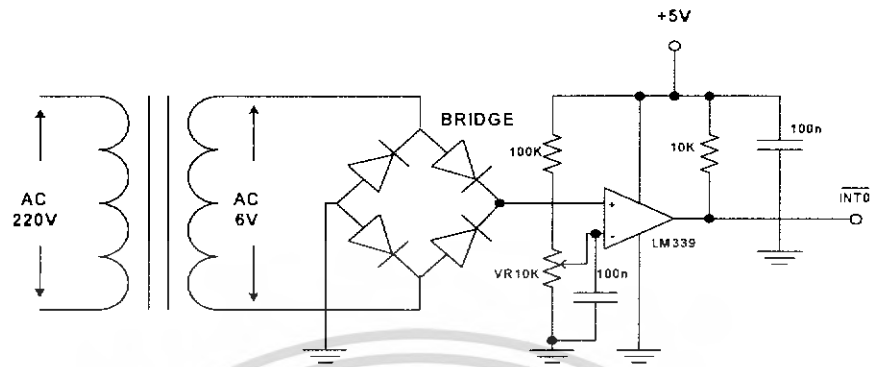
เป็นส่วนที่แยกวงจร High voltage ออกจากวงจรควบคุมต่างๆ เพื่อป้องกันแรงดัน 220 Vac โหลดเข้าสู่วงจรส่วนอื่นๆ



รูปที่ 3.3 แสดงวงจรทริกและแยกโหนด

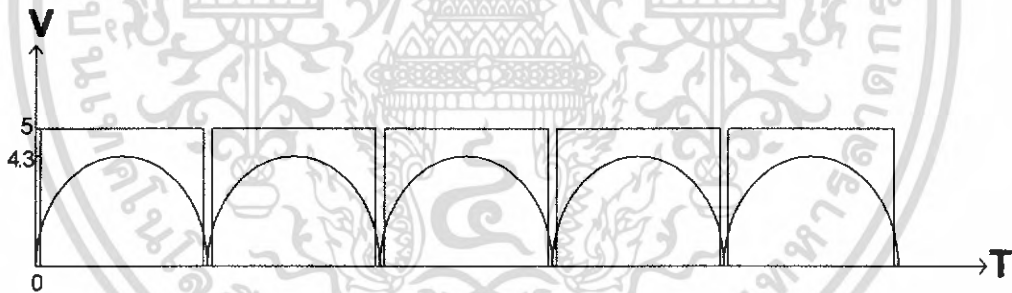
### 3.1.4 วงจรตรวจจับสัญญาณขาอิน

วงจรตรวจจับสัญญาณขาอิน เพื่อนำไปเป็นสัญญาณอินเทอร์รัพท์ให้กับไมโครคอนโทรลเลอร์ ในการเป็นสัญญาณเริ่มต้นของดีเลย์ สัญญาณทริกในการเริ่มทำงานของโหลด โดยเอาที่พุทของวงจรมีจะไปเข้าที่ขา INTO



รูปที่3.4 แสดงวงจรตรวจจับสัญญาณชานัน

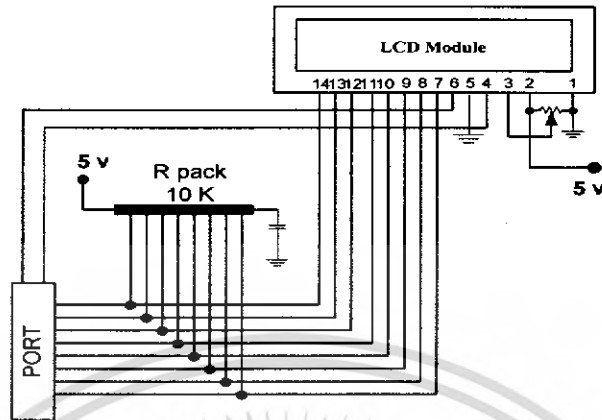
สัญญาณชานัน 5 โวลต์เมื่อผ่านวงจรบริดจ์เรกติไฟเออร์จะให้สัญญาณเอาต์พุต full wave นำไปเข้าวงจรคอมพาราเตอร์โดยใช้ LM339 และแรงดันเปรียบเทียบสามารถปรับได้โดยใช้ VR10K สามารถปรับช่วงที่เป็น 0 ของสัญญาณพัลส์เอาต์พุตได้ นำสัญญาณเอาต์พุตที่ได้จากวงจรคอมพาราเตอร์ไปเข้าขา INT0 ของไมโครคอนโทรลเลอร์เพื่อเป็นสัญญาณการอินเตอร์รัพท์ในการเริ่มดีเลย์



รูปที่3.5 สัญญาณชานันเมื่อวงจรคอมพาราเตอร์ที่ใช้ LM339

### 3.1.5 วงจรส่วนการแสดงผลออกทาง LCD Module

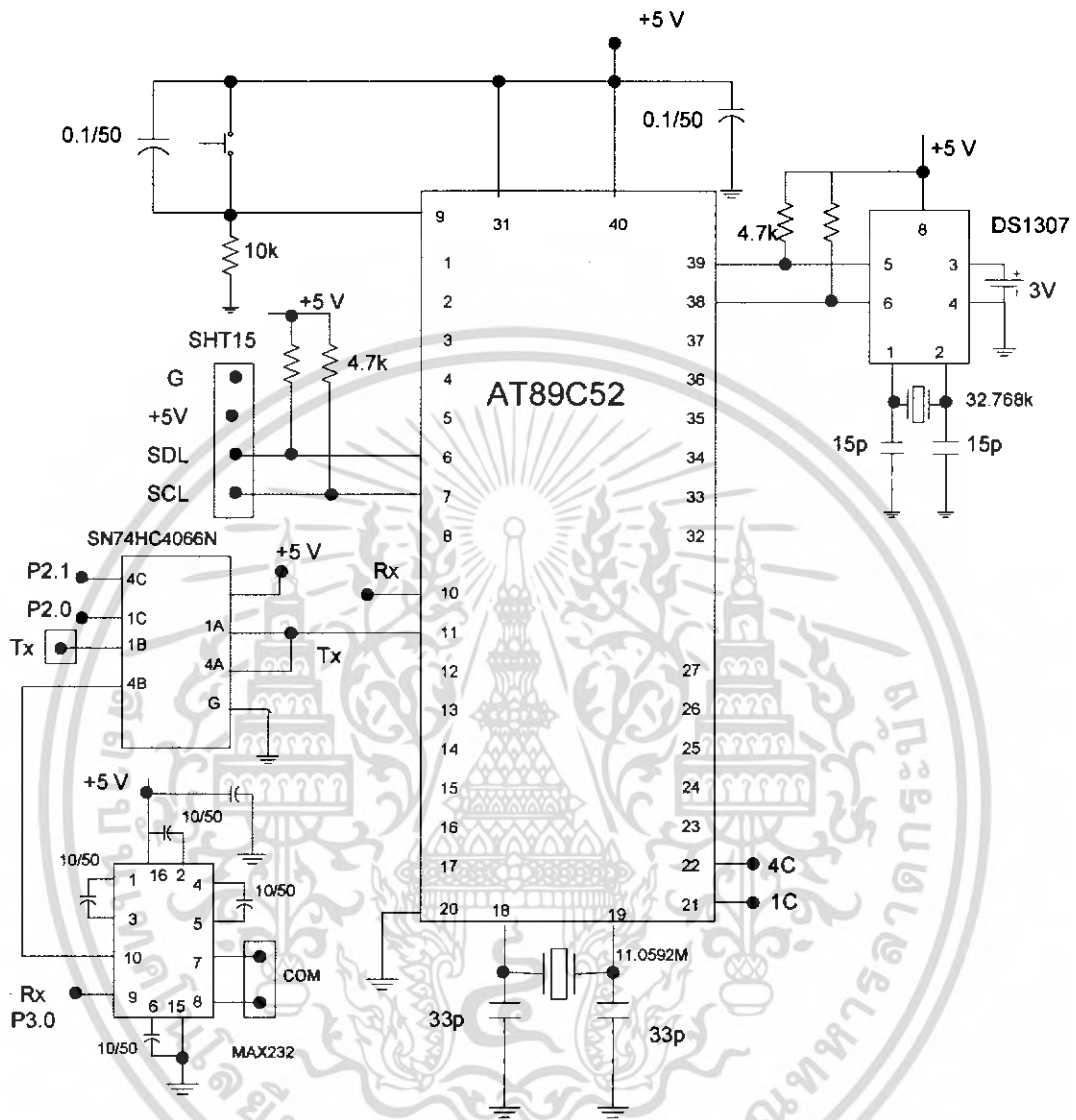
ส่วนแสดงผลทำหน้าที่ในการแสดงผลในขณะที่รับค่าจากภายนอกในกำหนดช่วงอุณหภูมิของตู้ การกดปุ่มเพื่อเพิ่มอุณหภูมิ หรือกดปุ่มเพื่อเปลี่ยนโหมดการทำงาน เมื่อทำการกดปุ่มที่ส่วนควบคุม ส่วนแสดงผลจะส่งข้อมูลไปยังไมโครคอนโทรลเลอร์ที่อยู่บนส่วนควบคุมหลักเพื่อให้ประมวลผลควบคุมอุปกรณ์ต่างๆ ตามค่าที่กำหนด แล้วส่งผลออกมาแสดงทางจอ LCD ซึ่งจะแสดงได้ดังรูปที่ 3.6



รูปที่ 3.6 แสดงการเชื่อมต่ออุปกรณ์ส่วนแสดงผลข้อมูล

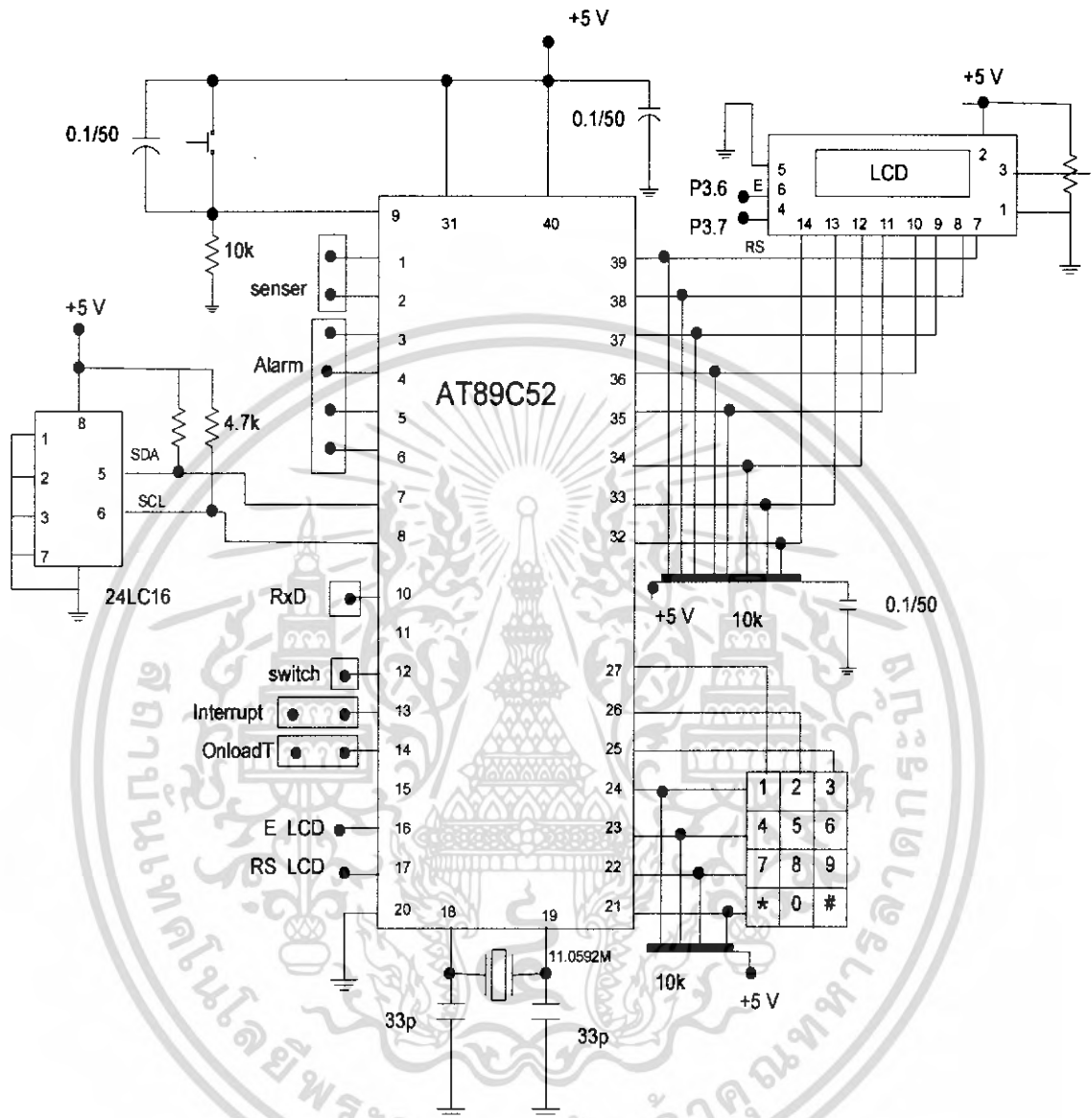
### 3.1.6 วงจรควบคุมการทำงาน

เป็นวงจรที่ควบคุมการทำงานของตู้อบเด็ก มีไมโครคอนโทรลเลอร์เป็นตัวประมวลผลควบคุมการทำงานทั้งหมด ประกอบด้วย 2 ส่วน ส่วนแรกเป็นส่วนที่ใช้ในการวัดและประมวลผลอุณหภูมิและความชื้นโดยใช้ SHT15 ในการตรวจจับ ส่วนที่ 2 เป็นส่วนควบคุมการแสดงผลผ่านทาง LCD และควบคุมการทำงานทางเฟสของขดลวดความร้อน



รูปที่ 3.7 วงจรส่วนที่ใช้ในการวัดและประมวลผลอุณหภูมิและความชื้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

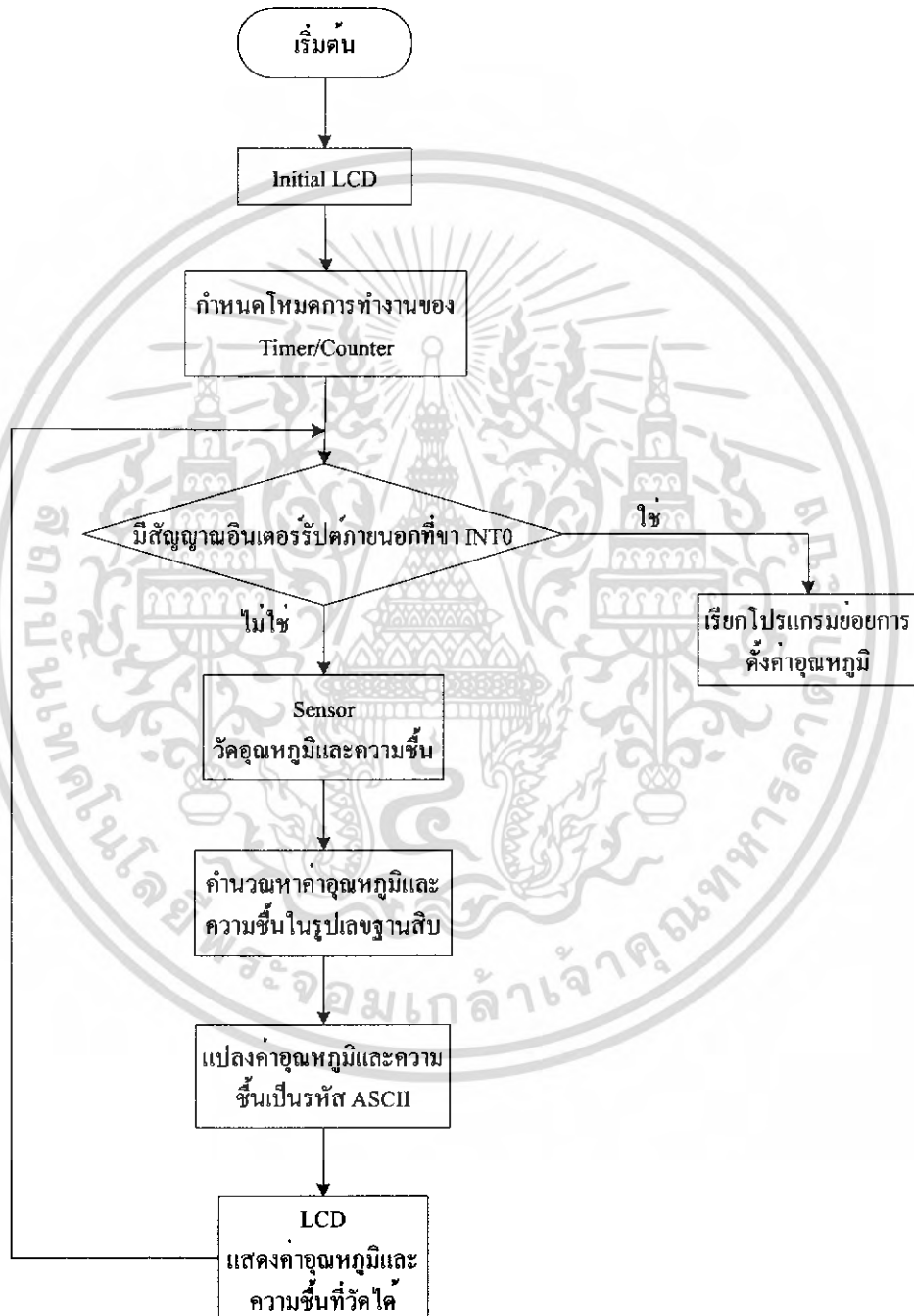


รูปที่ 3.8 วงจรส่วนที่ควบคุมการแสดงผลผ่านทาง LCD และควบคุมการทำงานทางเฟสของขดลวดความถี่อื่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

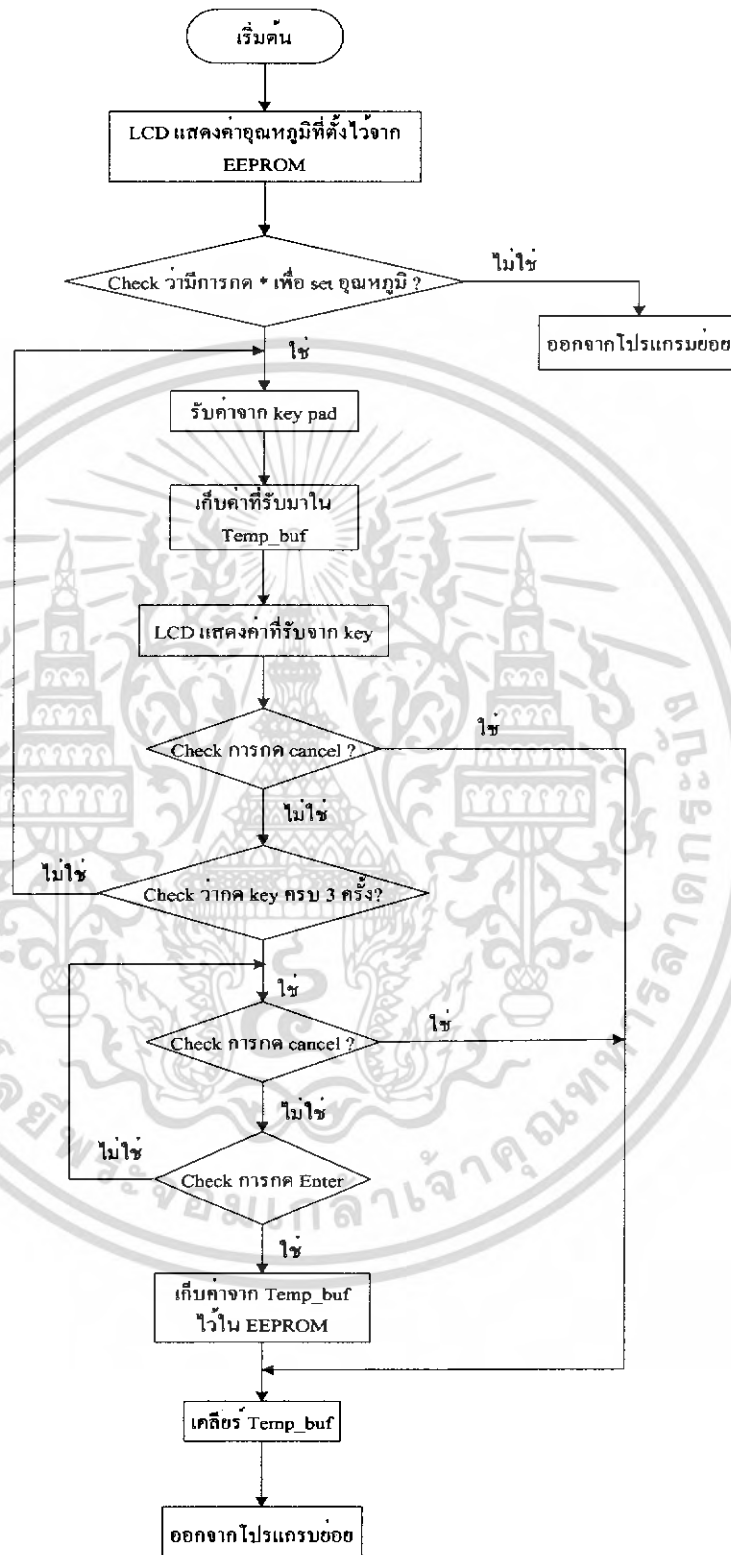
### 3.2 โปรแกรมการทำงานของตู้อบเด็ก

#### 3.2.1 Flow chart แสดงการทำงานของโปรแกรม



รูปที่ 3.9 Flow chart แสดงการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 Flow chart แสดงการทำงานของโปรแกรมย่อยการตั้งค่าอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 Flow chart แสดงการทำงานของโปรแกรมการควบคุม PID

### 3.2.2 อธิบายโปรแกรมการทำงานของตู้อบเค้ก

หลังจากเปิดสวิตช์ Power แล้วต้องทำการกดสวิตช์เพื่อตั้งค่าอุณหภูมิที่ต้องการเมื่อกดสวิตช์แล้วจอ LCD จะแสดงผลการตั้งค่าอุณหภูมิในครั้งก่อนหลังจากนั้นต้องทำการยืนยันค่าที่เคยตั้งไว้หรือยกเลิกค่าแล้วเข้าสู่โหมดการตั้งอุณหภูมิในครั้งใหม่ จากนั้นจึงทำการตั้งค่าอุณหภูมิ ซึ่งค่าอุณหภูมิที่ใช้ในการอบทาร์ตควรอยู่ในช่วงประมาณ 28-36 องศาเซลเซียส จากนั้นต้องยืนยันการตั้งค่าอุณหภูมิซึ่งค่าอุณหภูมิที่ตั้งใหม่นี้จะถูกเก็บไว้ใน EEPROM เมื่อทำการตั้งค่าอุณหภูมิเสร็จแล้วโปรแกรมจะออกจากโหมดการตั้งค่าอุณหภูมิเข้าสู่โหมดปกติซึ่งจะมีการแสดงผลอุณหภูมิและความชื้นที่วัดได้ในขณะนั้น ออกทางจอ LCD และในขณะเดียวกันจะมีการประมวลผลโดยระบบ PID เพื่อควบคุมอุณหภูมิให้เป็นไปตามค่าที่ได้ตั้งไว้

ระบบ PID เป็นระบบการควบคุมแบบป้อนกลับโดยจะนำค่าอุณหภูมิที่วัดได้ในการสุ่มแต่ละครั้งมาหาค่า error จากค่าอุณหภูมิที่ได้ตั้งไว้ จากนั้นเอาค่า error ที่ได้ ประมวลผลโดยสมการ

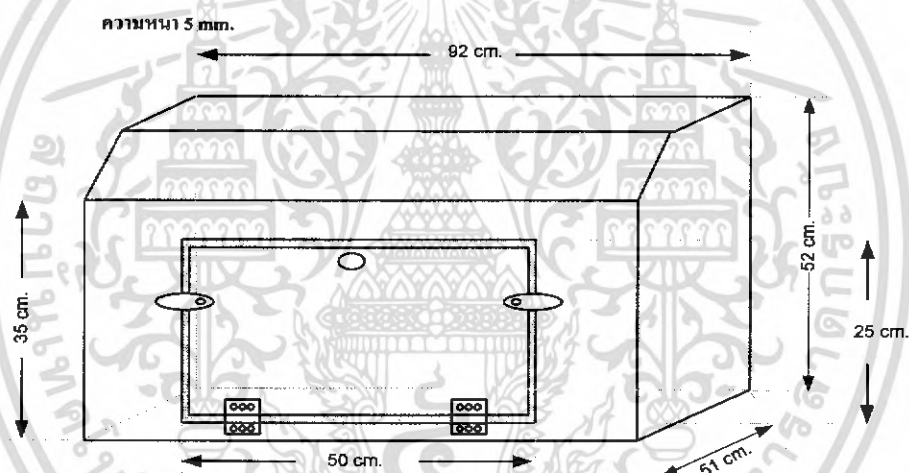
Proportional Integral Derivative Control จะได้ค่า % ในการควบคุม จะใช้ค่าที่ได้นี้ในการควบคุมเฟสของไฟสลับที่จะเข้าสู่ Heater ซึ่งจะเป็นตัวที่ให้ความร้อนภายในตู้อบ

### 3.3 การออกแบบส่วนประกอบภายนอก

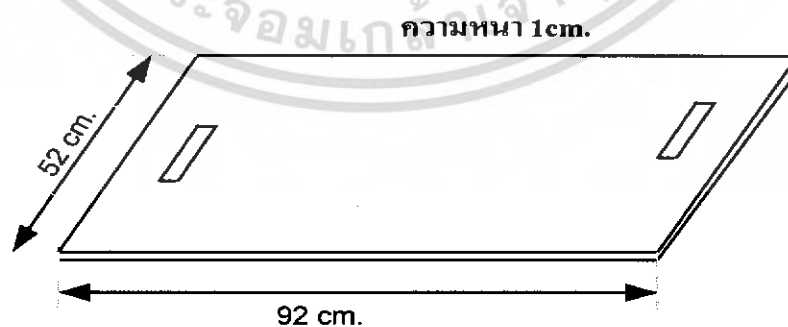
#### 3.3.1 กระจังครอบและแผ่นรองตัวเด็ก

ทำหน้าที่เปรียบเสมือนเป็นห้องที่ให้ทารกอาศัยอยู่ชั่วคราว เป็นส่วนที่ครอบตัวทารกไว้ทำให้เกิดการไหลเวียนของอากาศภายในตู้อบ ซึ่งมีอุณหภูมิตามที่ตั้งไว้เพื่อเหมาะสมกับสภาพร่างกายของทารก จนกว่าทารกจะแข็งแรงพอที่จะออกมาสู่สภาวะปกติภายนอกได้

กระจังครอบนั้น ทำจากพลาสติกขาวใส สามารถมองเห็นภายใน ได้อย่างชัดเจน



รูปที่ 3.12 ส่วนกระจังครอบ



รูปที่ 3.13 แผ่นรองตัวเด็ก

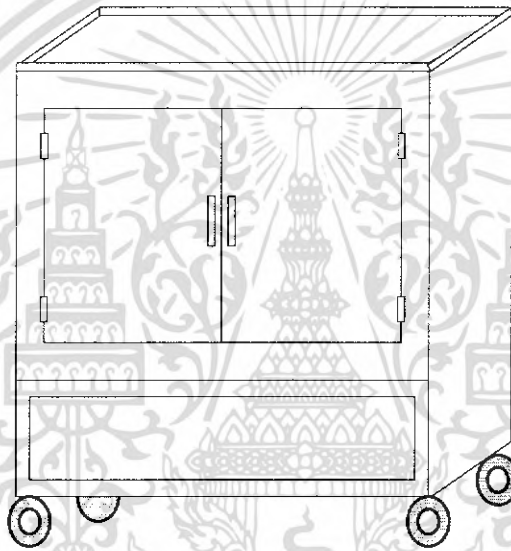
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3.2 Chamber

เป็นส่วนกำหนดทิศทางการไหลของอากาศภายในตู้อบ ซึ่งใช้ไฟเบอร์เป็นวัสดุในการสร้างส่วนประกอบส่วนนี้

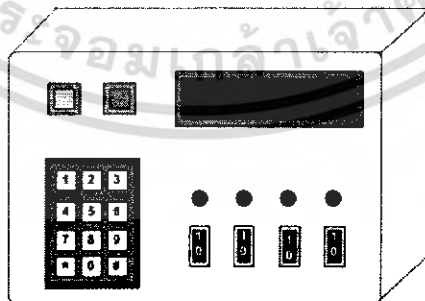
### 3.3.3 ตู้ฐานรอง

ใช้วางส่วนประกอบทุกอย่างของตู้อบเตี๊ยกทารกแรกเกิด เช่น กระจาอมครอบ, แผ่นรองตัวเตี๊ยก, Chamber, ชุดควบคุมระบบทั้งหมด และส่วนแสดงผล



รูปที่ 3.14 ตู้ฐานรอง

### 3.3.4 ส่วนแสดงผล



รูปที่ 3.15 ส่วนหน้าปัดแสดงผล และสัญญาณเตือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 แสดงลักษณะโดยรวมของตุ๋บเด็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

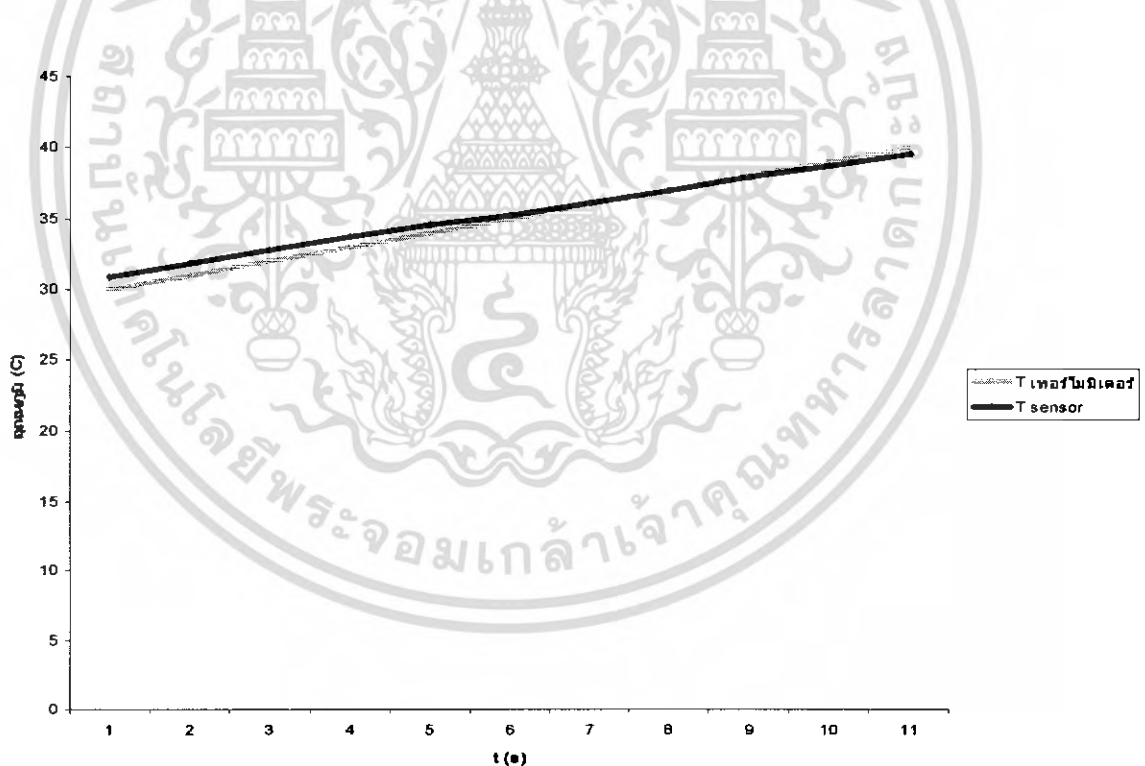
### การทดลองและผลการทดลอง

การทดลองนี้ทำเพื่อวัตถุประสงค์ดังต่อไปนี้

- 4.1 เพื่อทดสอบความเป็นเชิงเส้นของช่วงเวลาการเพิ่มขึ้นของอุณหภูมิเทียบกับเวลา
- 4.2 เพื่อปรับค่าค่าคงที่(K) ต่างๆของระบบพีไอดี (PID) ที่เหมาะสมกับระบบมากที่สุด

#### 4.1 ทดสอบความเป็นเชิงเส้นของช่วงเวลาการเพิ่มขึ้นของอุณหภูมิเทียบกับเวลา

โดยการให้โหลคทำงานเต็ม 100 เปอร์เซ็นต์ แล้วทำการบันทึกค่าอุณหภูมิทุกๆ 1 นาที ผลที่ได้ดังรูปที่ 4.1 ซึ่งจะเห็นได้ว่าอุณหภูมิเพิ่มขึ้นเกือบเป็นเชิงเส้น และค่าที่วัดได้จากเซนเซอร์กับจากเทอร์โมมิเตอร์มีค่าใกล้เคียงกัน

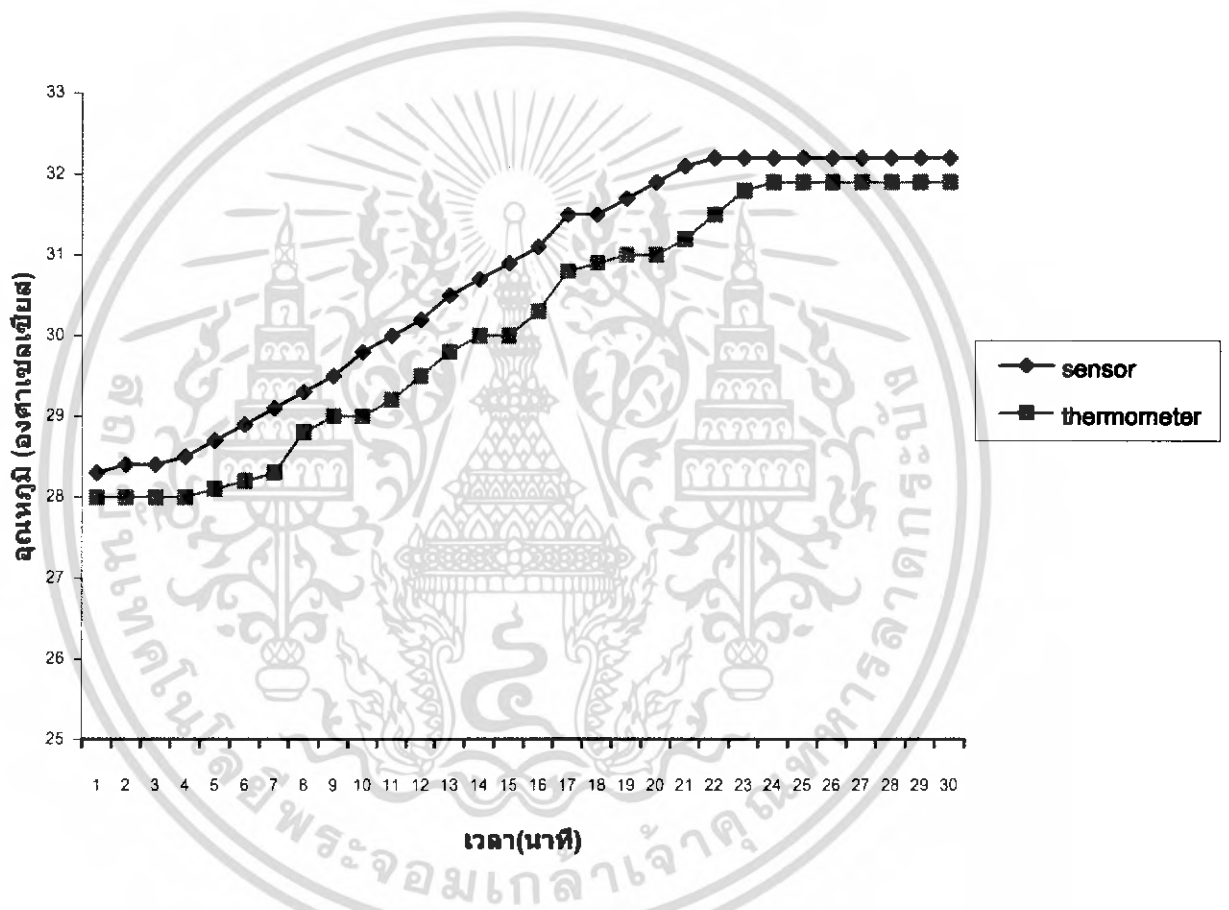


รูปที่ 4.1 กราฟแสดงความแตกต่างระหว่างค่าอุณหภูมิที่วัดจากเซนเซอร์กับวัดจากเทอร์โมมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

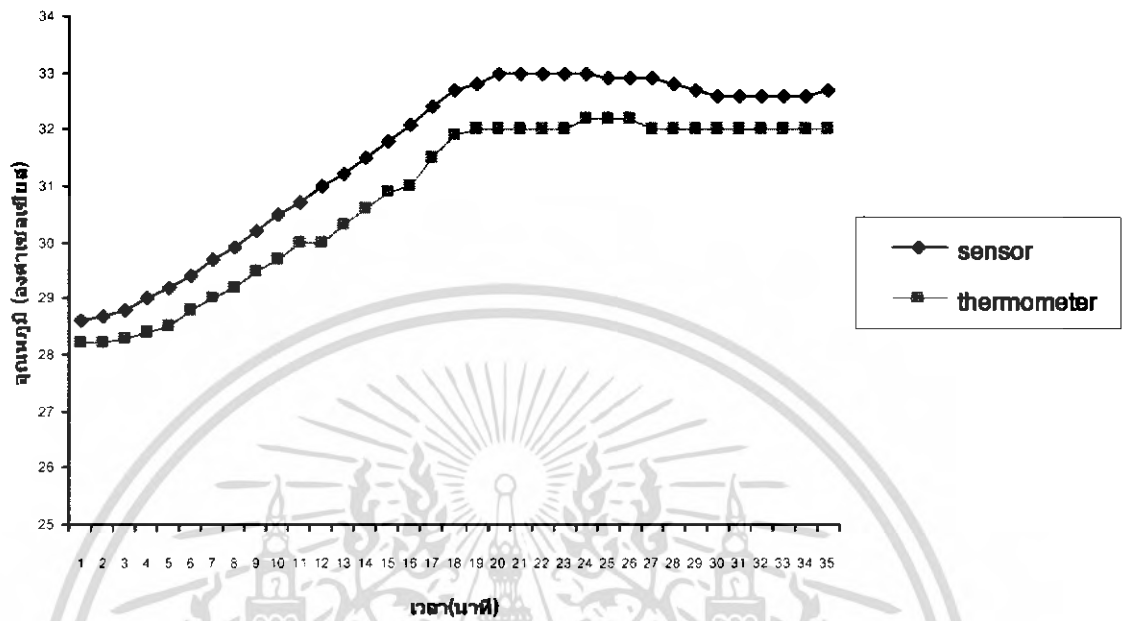
#### 4.2ปรับค่าค่าคงที่(K)ต่างๆของระบบพีไอดี (PID) ที่เหมาะสมกับระบบมากที่สุด

โดยเริ่มจากการปรับค่าค่าคงที่ของพี (Kp) ก่อนเป็นอันดับแรกซึ่งจะต้องทำการสุ่มเปลี่ยนค่าไปเรื่อยๆจนได้ค่าที่เข้าใกล้จุดกำหนดมากที่สุด ซึ่งค่าคงที่ได้คือ 90, 110, 120 และ 150 ตามลำดับทำการวัดค่าแล้ววาดกราฟได้ดังรูปที่ 4.2, 4.3, 4.4, 4.5 และ 4.6

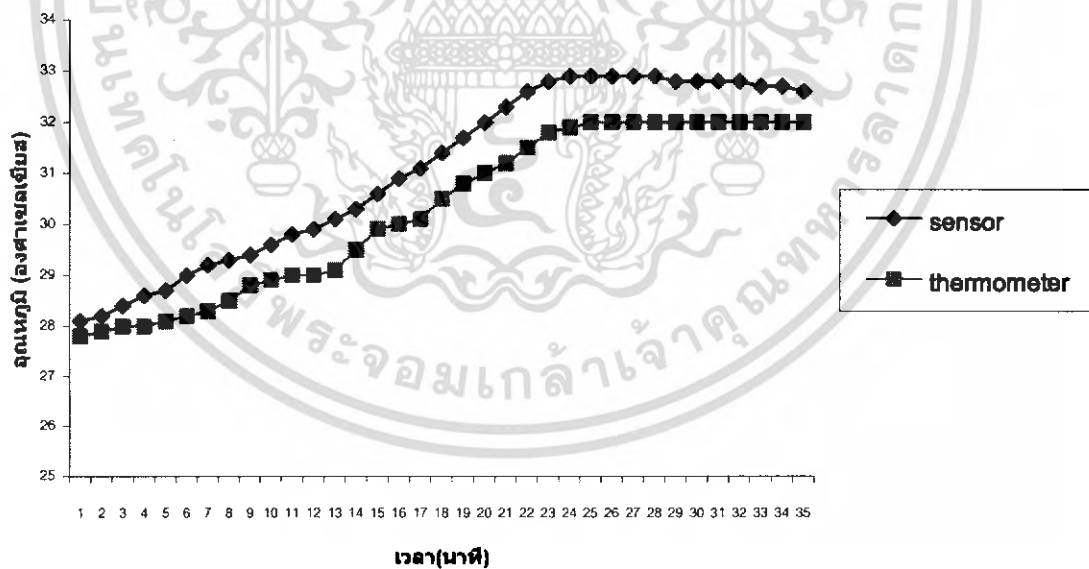


รูปที่ 4.2 กราฟแสดงการปรับค่าค่าคงที่ของพี (Kp = 90)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

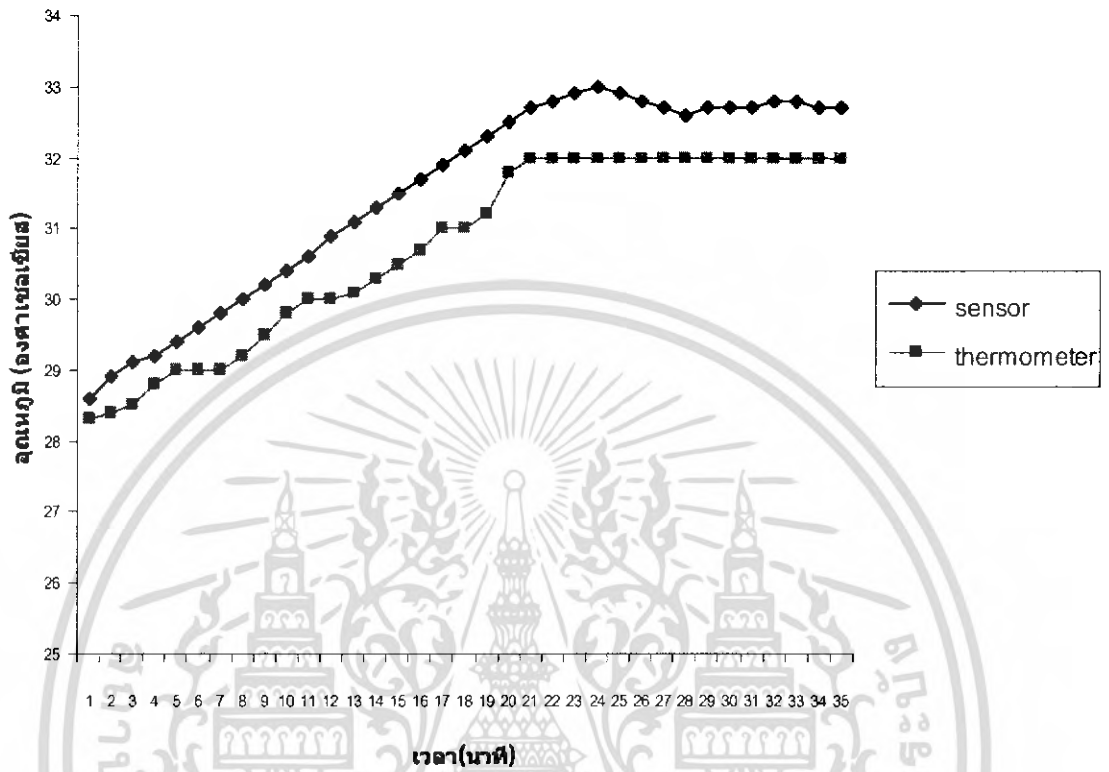


รูปที่ 4.3 กราฟแสดงการปรับค่าค่างของพี ( $K_p = 110$ )



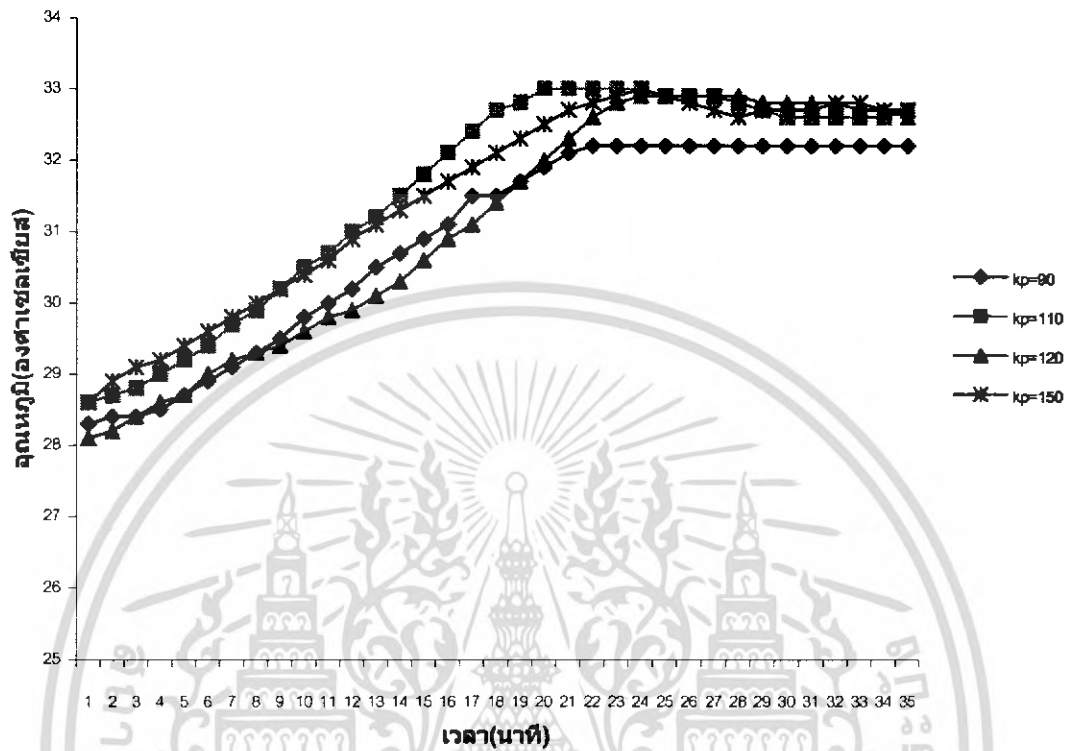
รูปที่ 4.4 กราฟแสดงการปรับค่าค่างของพี ( $K_p = 120$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 กราฟแสดงการปรับค่าค่างที่ของพี ( $K_p = 150$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



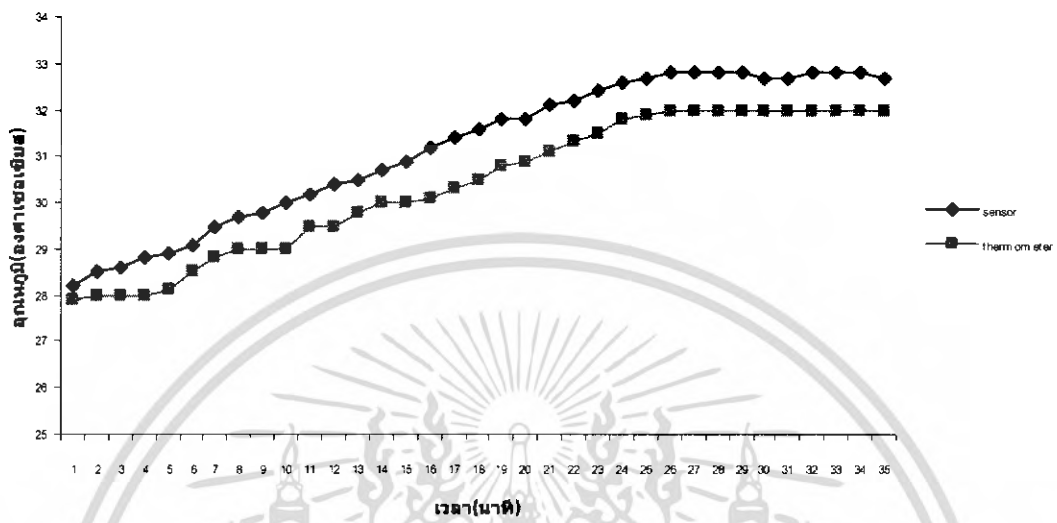
รูปที่ 4.6 กราฟแสดงการเปรียบเทียบจุดหมุมิ เมื่อใช้ค่า Kp ต่างๆ

ตารางที่ 4.1 แสดงค่า Overshoot , Risetime และ Sattling Time ที่ Kp ค่าต่างๆ

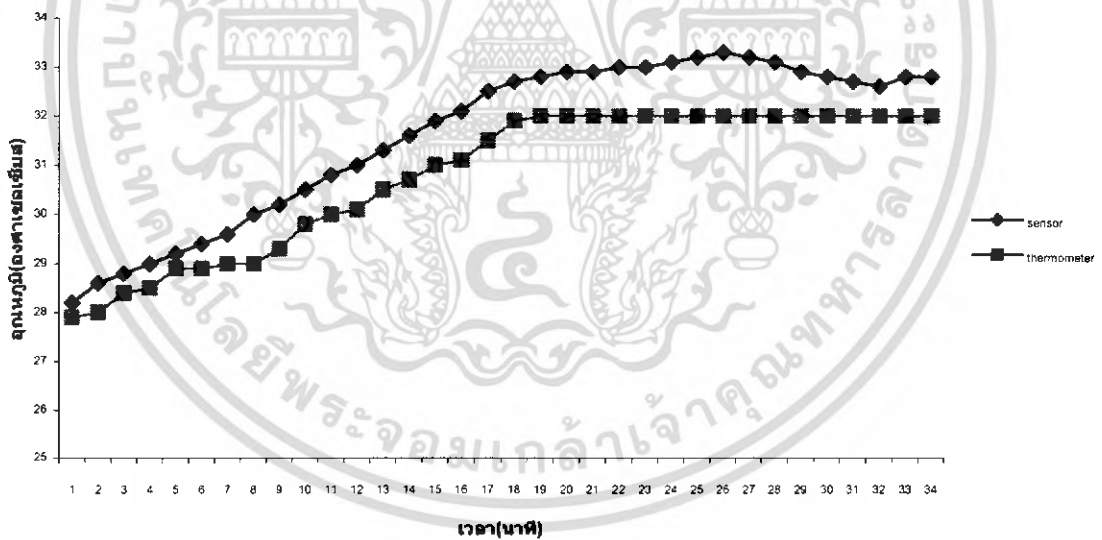
<b>Kp</b>	<b>Overshoot</b> (%)	<b>Risetime</b> (mins)	<b>Sattling Time</b> (mins)
<b>90</b>	0	20	24
<b>110</b>	1.21	18	29
<b>120</b>	0.91	21	31
<b>150</b>	0.90	20	30

จากรูปที่ 4.6 และตารางที่ 4.1 จะเห็นว่า ค่าคงที่ของพีที่ดีที่สุดคือ 110 เพราะ Risetime มีค่าน้อยที่สุด ดังนั้นจึงนำค่า Kp = 110 มาใช้ทำการปรับหาค่าคงที่ของไอ (Ki) โดยทำการสุ่มเช่นเดียวกัน ค่าคงที่ได้คือ 0.1, 0.5, 0.7 ตามลำดับ ทำการวัดค่าแล้ววาดกราฟได้ดังรูปที่ 4.7, 4.8, 4.9 และ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

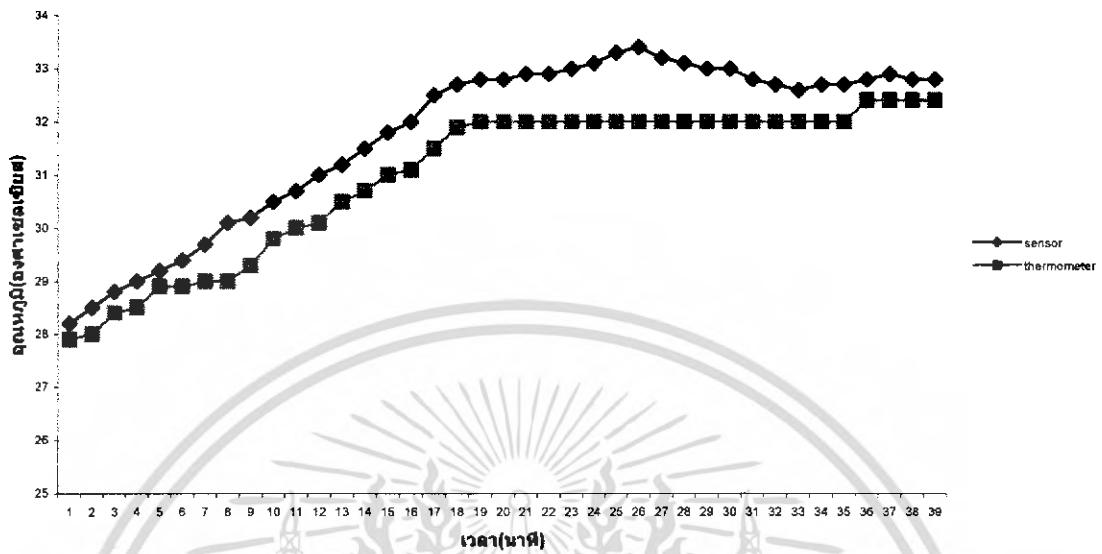


รูปที่ 4.7 กราฟแสดงการปรับค่าค่าคงที่ของพีไอ ( $K_p=110$ ,  $K_i=0.1$ )

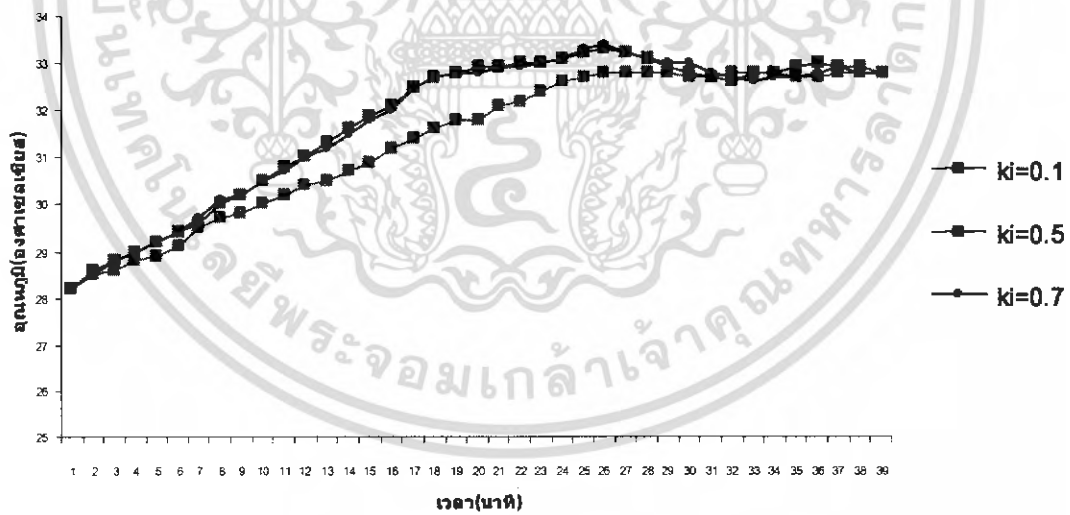


รูปที่ 4.8 กราฟแสดงการปรับค่าค่าคงที่ของพีไอ ( $K_p=110$ ,  $K_i=0.5$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 กราฟแสดงการปรับค่าค่างที่ของพีไอ ( $K_p = 110, K_i = 0.7$ )



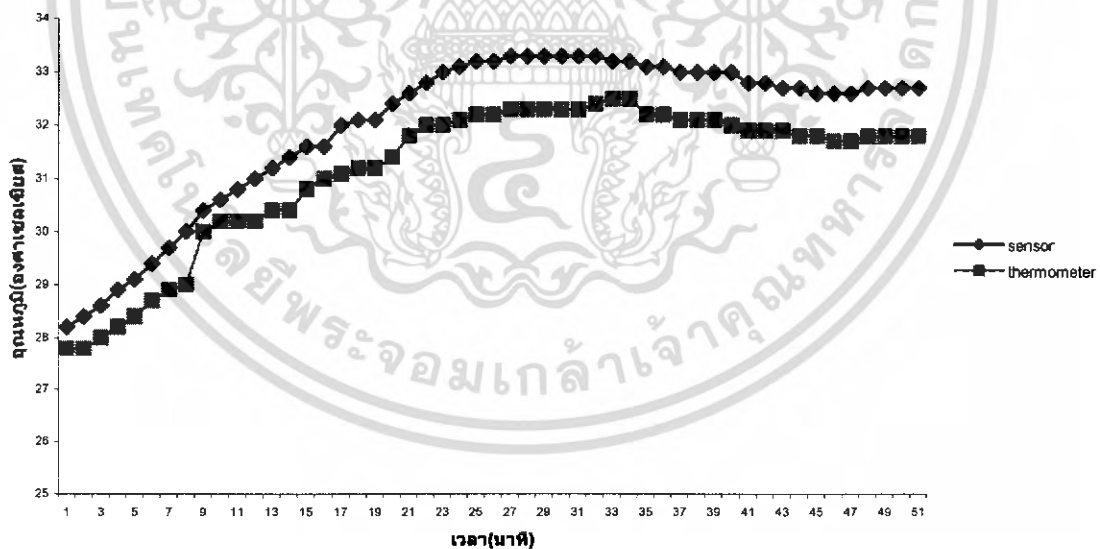
รูปที่ 4.10 กราฟแสดงการเปรียบเทียบอุณหภูมิ เมื่อใช้ค่า  $K_p = 110$  ส่วนค่า  $K_i$  ต่างๆกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงค่า Overshoot , Risetime และ Sattling Time ที่ Ki ค่าต่างๆ

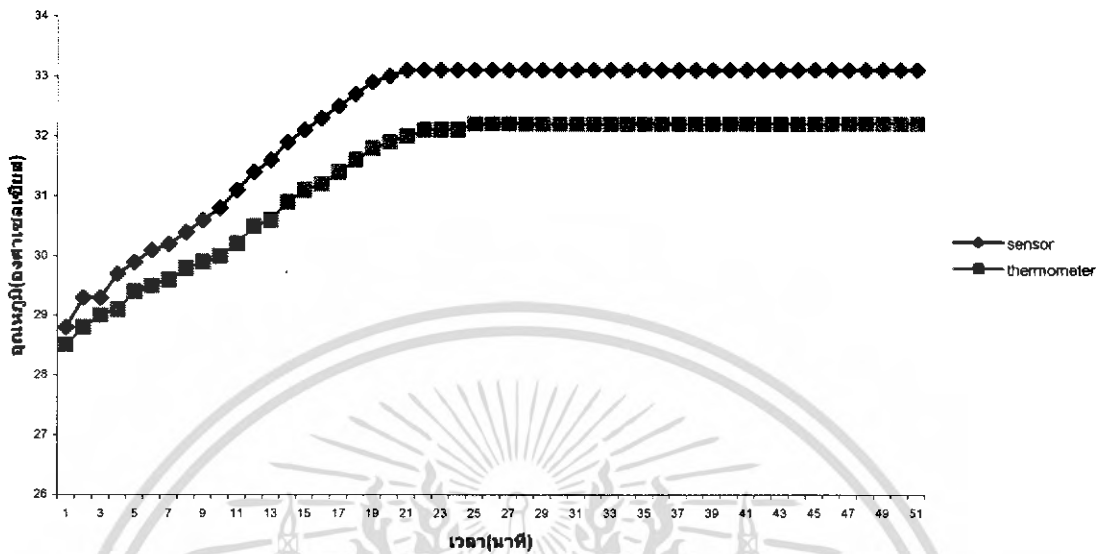
<b>Kp=110</b>	<b>Overshoot</b>	<b>Risetime</b>	<b>Sattling Time</b>
<b>Ki</b>	(%)	(mins)	(mins)
<b>0.1</b>	0.09	23	35
<b>0.5</b>	1.09	18	34
<b>0.7</b>	1.41	18	39

การปรับค่าคงที่ของพีไอ ( $K_p$  ,  $K_i$ ) สามารถเปรียบเทียบได้ดังรูปที่ 4.10 และตารางที่ 4.2 ค่าคงที่ของพีไอที่ดีที่สุดคือ  $K_p = 110$  และ  $K_i = 0.5$  เพราะค่า Risetime น้อยและค่า Sattling Time อยู่ในเกณฑ์ที่ดีกว่าเมื่อเปรียบเทียบกับ  $K_i$  ค่าอื่นๆ นำค่าที่ได้นี้มาทำการปรับค่าคงที่ดี ( $K_d$ ) โดยทำการสุ่มเช่นเดียวกันค่าคงที่ได้คือ 0.05 , 0.1 , 0.5 ตามลำดับ ทำการวัดค่าแล้ววาดกราฟได้ดังรูปที่ 4.11 , 4.12 , 4.13 และ 4.14



รูปที่ 4.11 กราฟแสดงการปรับค่าคงที่ของพีไอดี ( $K_p = 110$  ,  $K_i = 0.5$  ,  $K_d = 0.05$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

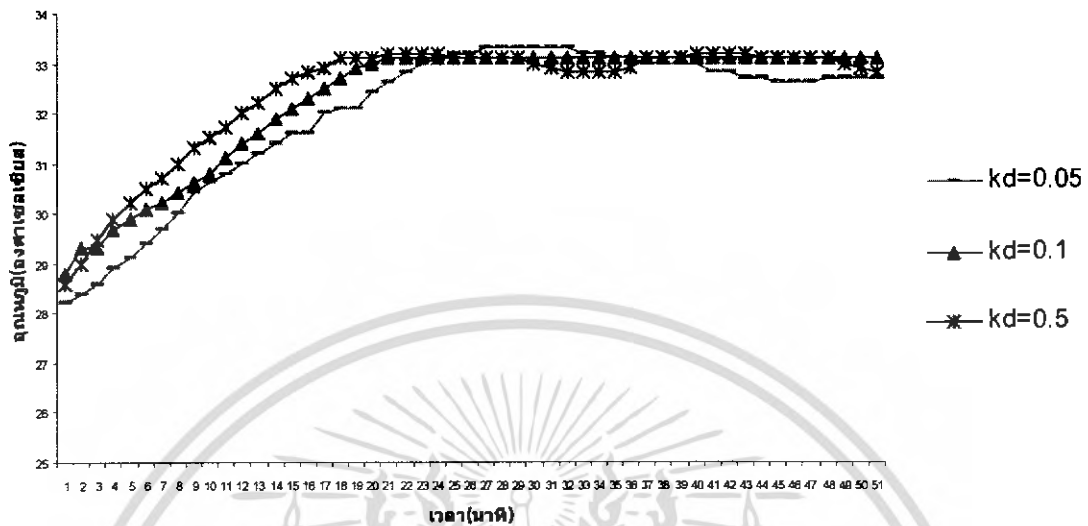


รูปที่ 4.12 กราฟแสดงการปรับค่าคงที่ของพีไอดี ( $K_p = 110$ ,  $K_i = 0.5$ ,  $K_d = 0.1$ )



รูปที่ 4.13 กราฟแสดงการปรับค่าคงที่ของพีไอดี ( $K_p = 110$ ,  $K_i = 0.5$ ,  $K_d = 0.5$ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



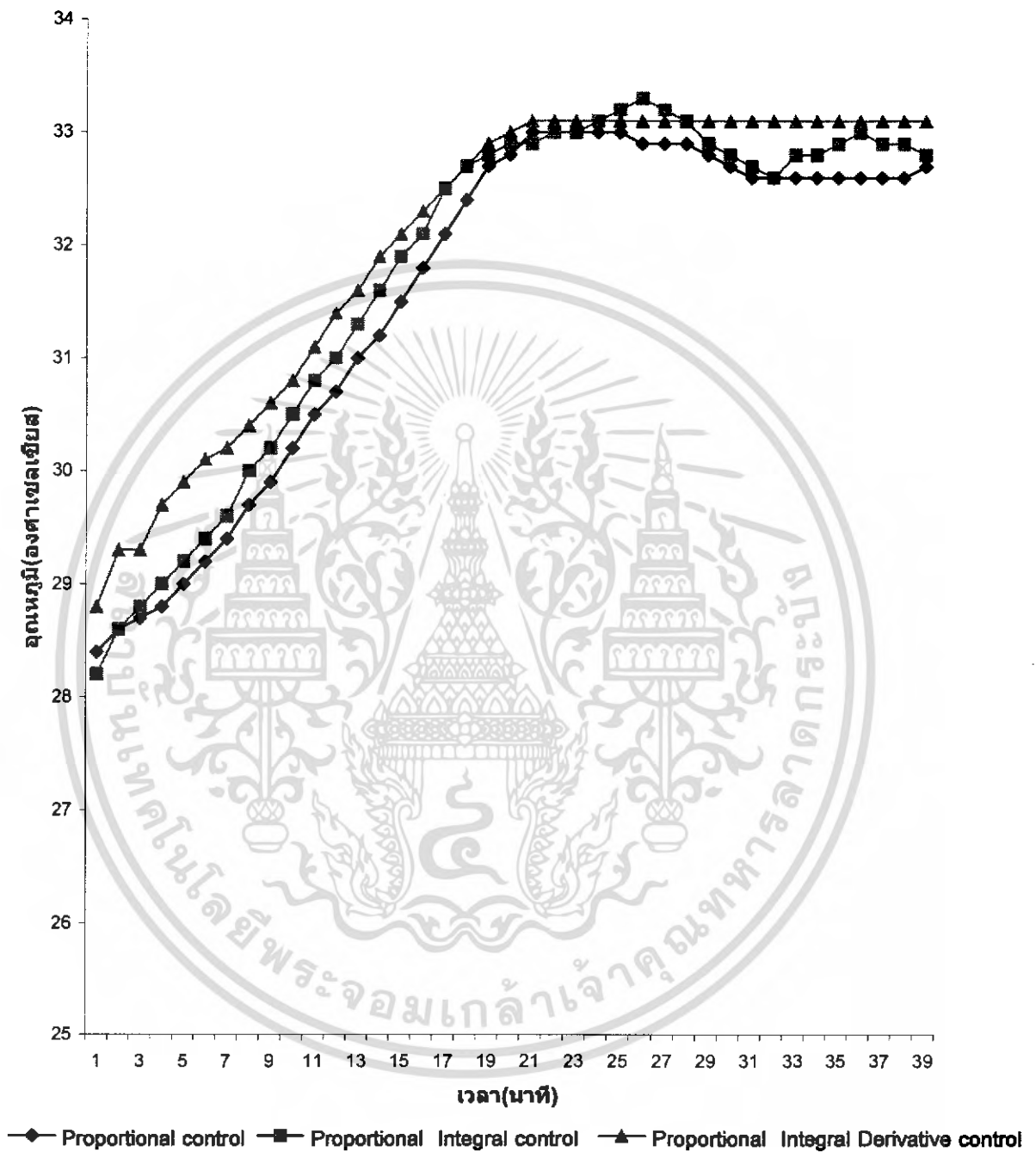
รูปที่ 4.14 กราฟแสดงการเปรียบเทียบอุณหภูมิ เมื่อใช้ค่า  $K_p = 110$ ,  $K_i = 0.5$  และ  $K_d$  ต่างๆกัน

ตารางที่ 4.3 แสดงค่า Overshoot , Risetime และ Sattling Time ที่  $K_d$  ค่าต่างๆ

<b><math>K_p=110, K_i=0.5</math></b>	<b>Overshoot</b>	<b>Risetime</b>	<b>Sattling Time</b>
<b><math>K_d</math></b>	(%)	(mins)	(mins)
<b>0.05</b>	0.96	22	48
<b>0.1</b>	0.01	18	26
<b>0.5</b>	0.44	15	39

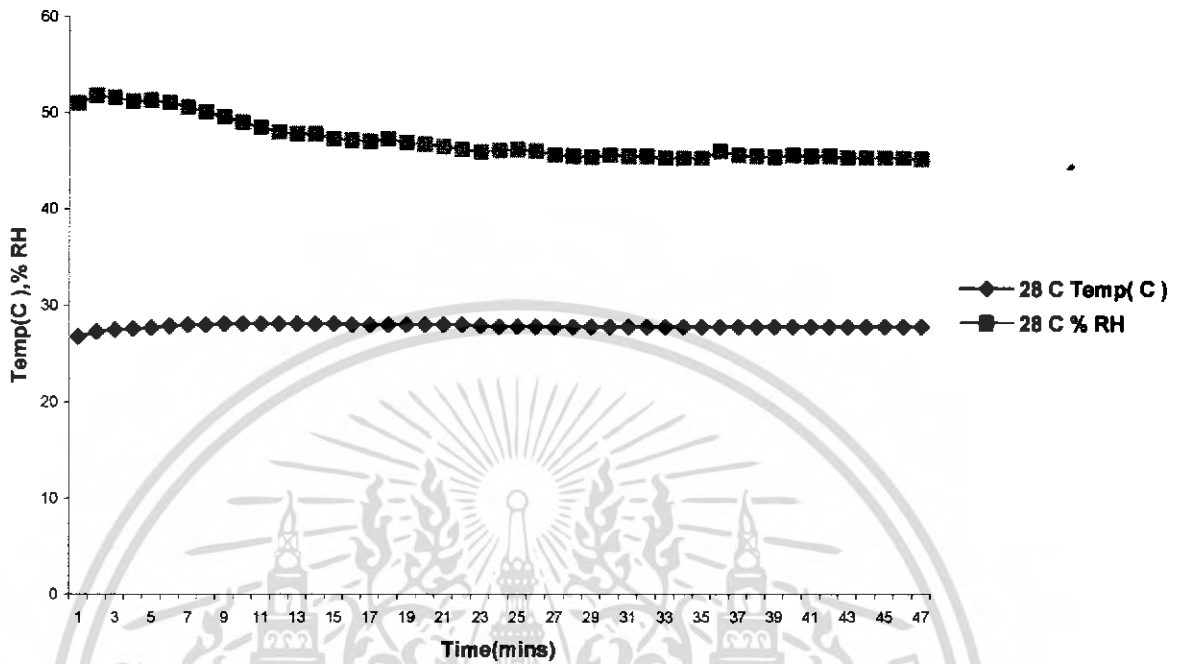
การปรับค่าคงที่ของพีไอดี ( $K_p$ ,  $K_i$ ,  $K_d$ ) สามารถเปรียบเทียบได้ดังรูปที่ 4.14 และตารางที่ 4.3 ซึ่งค่าคงที่ของพีไอดีที่ดีที่สุดคือ  $K_p = 110$ ,  $K_i = 0.5$  และ  $K_d = 0.1$  เพราะเปอร์เซ็นต์ Overshoot มีค่าน้อยมากเมื่อเปรียบเทียบกับ  $K_d$  ค่าอื่นๆ

เมื่อเราได้ค่าคงที่ที่เหมาะสมของสมการ การควบคุมแบบพีไอดีครบทั้งหมดทุกค่าแล้ว นำค่าที่ได้ไปใช้ในการเขียน โปรแกรมเพื่อควบคุมการทำงานของโหลด (ขดลวดความร้อน) ต่อไป

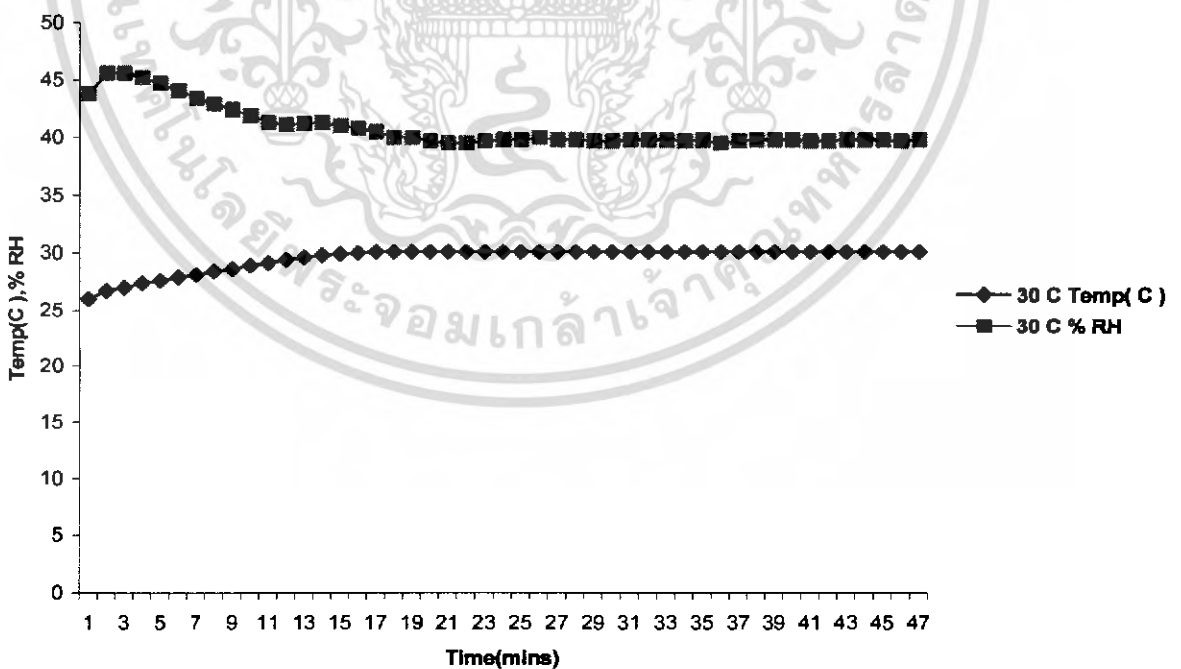


รูปที่ 4.15 กราฟแสดงการเปรียบเทียบการควบคุมแบบต่างๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

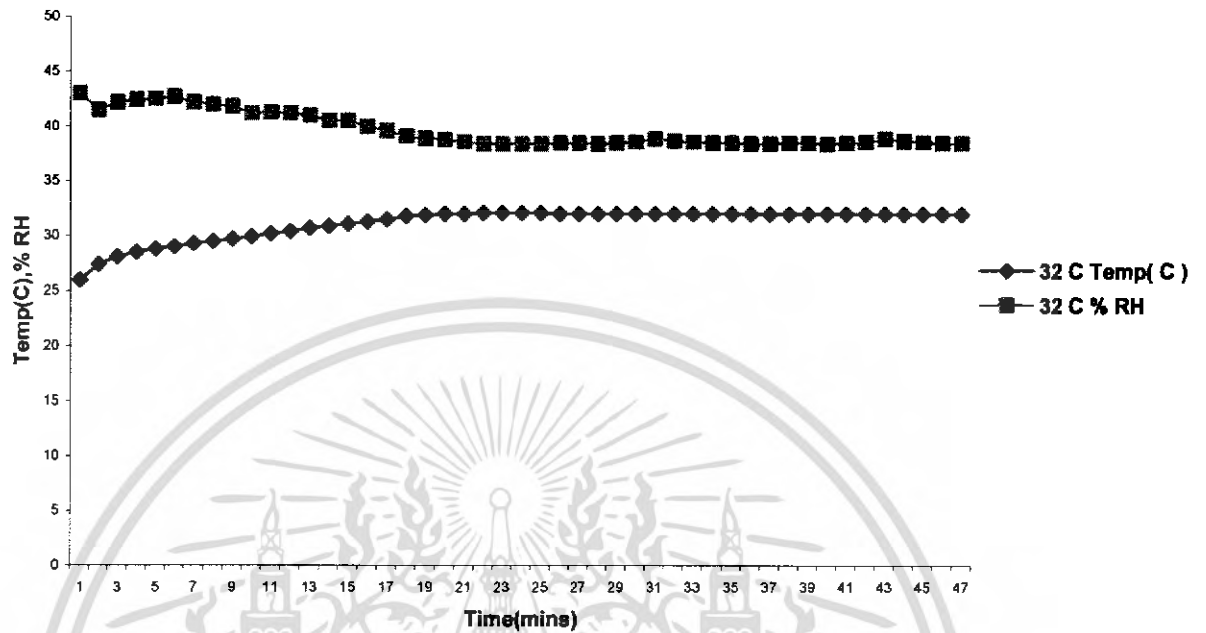


รูปที่ 4.16 กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้เท่ากับ 28C

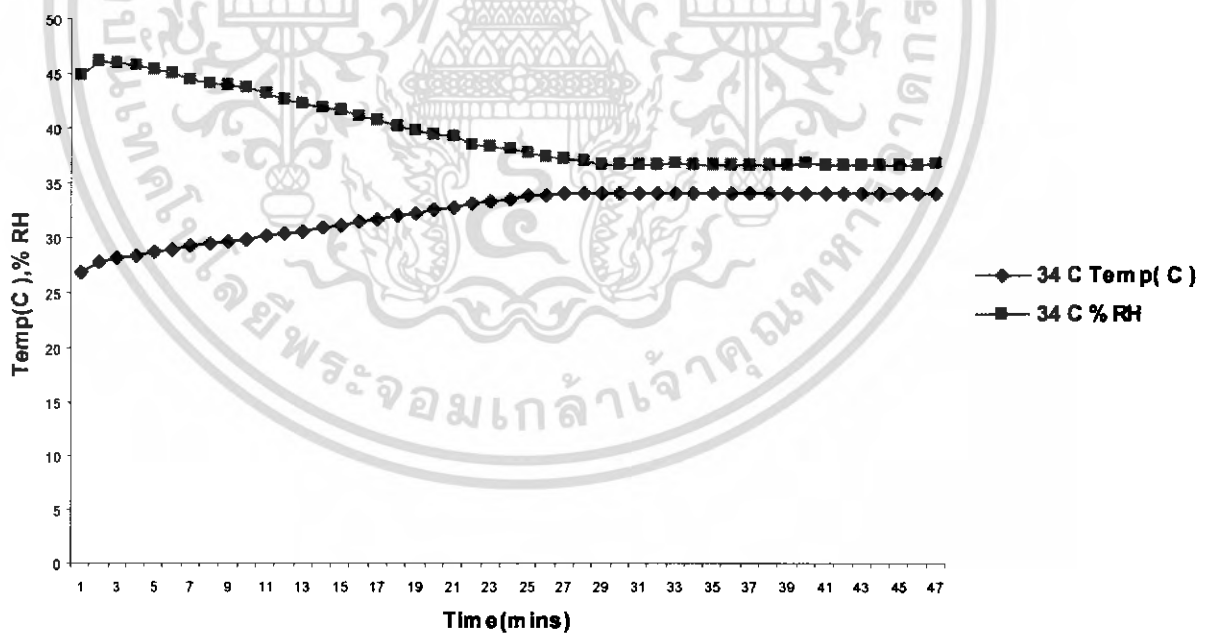


รูปที่ 4.17 กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้เท่ากับ 30C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

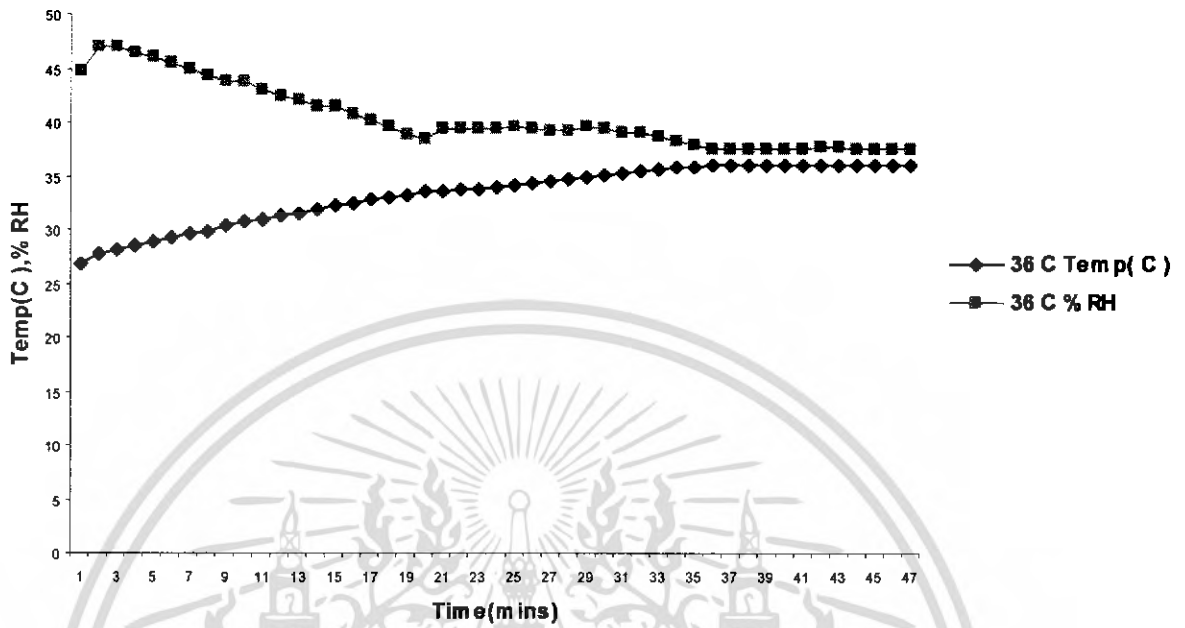


รูปที่ 4.18 กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้เท่ากับ 32C



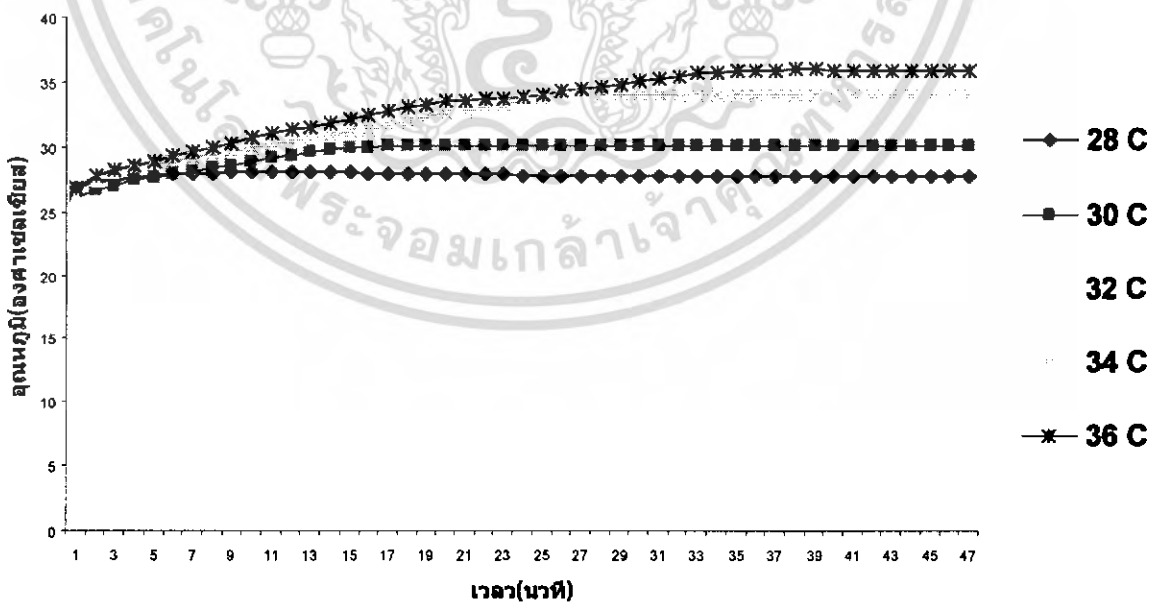
รูปที่ 4.19 กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้เท่ากับ 34C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 กราฟแสดงค่าอุณหภูมิ,เปอร์เซ็นต์ความชื้น ที่เวลาต่างๆ ขณะตั้งอุณหภูมิภายในตู้เท่ากับ 36C

จากรูปที่ 4.16 ถึง รูปที่ 4.20 จะเห็นว่าค่าอุณหภูมิจะแปรผกผันกับค่าเปอร์เซ็นต์ความชื้น



รูปที่ 4.21 กราฟแสดงอุณหภูมิต่างๆ ภายในตู้ ที่สามารถควบคุมได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### สรุป วิจัยการผลิตการทดลอง และข้อเสนอแนะ

#### สรุป และวิจัยการผลิตการทดลอง

โครงการนี้เป็นารออกแบบตู้อบเค้กทารกแรกเกิดในระบบควบคุมแบบดิจิทัล โดยใช้ไมโครคอนโทรลเลอร์ตระกูลMCS-51 เป็นตัวประมวลผลเพื่อควบคุมการทำงานทั้งหมดของระบบ และนำอัลกอริทึมของระบบควบคุมแบบพีไอดี มาใช้ในการเขียน โปรแกรมประมวลผล

ในภาคการศึกษาที่ 2 นี้ ได้ทำการทดลองหาค่าคงที่ของสมการ การควบคุมแบบพีไอดี โดยใช้ขดลวดความร้อนเป็นโหลด ทำหน้าที่ให้ความร้อน จากผลการทดลองที่ได้สามารถสรุปได้ดังนี้

1. สามารถควบคุมอุณหภูมิในช่วง  $25^{\circ}\text{C}$ - $38^{\circ}\text{C}$  โดยมีความผิดพลาดที่ระดับ 0.3-0.5 องศาเซลเซียส เมื่อใช้ SHT15
2. ค่า  $K_p$ ,  $K_i$  และ  $K_d$  มีผลต่อความเสถียรภาพของระบบ โดยที่  $K_p$  จะมีผลเมื่ออุณหภูมิเกิดการเปลี่ยนแปลงโดยดูได้จากสมการในระบบ PID และค่า  $K_i$  จะมีผลต่อการแกว่งของระบบ ส่วนค่า  $K_d$  มีผลต่อการเข้าสู่จุดกำหนด(set point) ของอุณหภูมิที่ใกล้เคียงและมีเสถียรภาพของระบบ

โดยค่าที่เลือกใช้ เป็นดังนี้  $K_p = 110$

$K_i = 0.5$

$K_d = 0.1$

3. อัลกอริทึมแบบ PID เหมาะสมกับระบบการควบคุมอุณหภูมิ

#### ข้อเสนอแนะ

ในการทดลองเพื่อหาค่าคงที่ที่เหมาะสมนั้นจะต้องทำในระบบปิด และต้องเป็นระบบที่เหมือนกันทุกประการ ทั้งอากาศภายนอก และตำแหน่งของอุปกรณ์ต่างๆ ที่ใช้ทำการทดลอง เพื่อจะสามารถเปรียบเทียบผลการทดลองได้อย่างถูกต้องแม่นยำมากที่สุด

## เอกสารอ้างอิง

1. เกรียงศักดิ์ จีระแพทย์, “แนวทางการพัฒนาระบบเพื่อลดอัตราการตายของทารกแรกเกิด”, หนังสือวันอนามัยโลก, 2541 , หน้า 69-74
2. เกรียงศักดิ์ จีระแพทย์, “ทารกกลุ่มเสี่ยง”, กุมารเวชศาสตร์, เล่ม 1, 2540, หน้า 228-236
3. เกรียงศักดิ์ จีระแพทย์, “เกณฑ์การประเมินการดูแลทารกแรกเกิด โครงการลูกเกิดรอดแม่ปลอดภัย”, กรมอนามัย กระทรวงสาธารณสุข, 2541
4. เกรียงศักดิ์ จีระแพทย์, “การดูแลระบบการหายใจในทารกแรกเกิด”, 2536
5. ประพุทธ ศิริบุญย และอรุพล บุญประกอบ, “ทารกแรกเกิด”, โครงการตำรา-ศิริราช คณะแพทยศาสตร์ศิริราชพยาบาล มหาวิทยาลัยมหิดล, 2533
6. ชัยวัฒน์ ถิรมตระกูลจิตร วิไล และวรพจน์ กรามแก้ววัฒนกุล, “เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51”, โนวาทีฟเอ็กพอร์ทิเมนต์ กรุงเทพ
7. สมยศ จุณณะปิยะ, “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์”, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 2543
8. คู่มือการใช้งาน V-85 atom infant incubator
9. สำเนาเอกสาร “ระบบการควบคุมป้อนกลับ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ก

## โปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

*****lcd.h*****

```

```

sbit e = P3^6;

```

```

sbit rs = P3^7;

```

```

-----
void delay(int tick)

```

```

-----
{
unsigned int i,j;
for(i=0;i<tick;i++)
for(j=0;j<250;j++);
}

```

```

-----
void lcd_command(unsigned char com)

```

```

-----
{
rs = 0;
e = 1;
P0 = com;
delay(5);
e = 0;
delay(5);
}

```

```

-----
void lcd_text(unsigned char text)

```

```

-----
{
rs = 1;
e = 1;
P0 = text;
delay(5);
e = 0;
delay(5);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
-----
```

```
void lcd_clear()
-----
```

```
{
lcd_command(0x01);
}
-----
```

```
void lcd_jumporigin()
-----
```

```
{
lcd_command(0x02);
}
-----
```

```
void lcd_init()
-----
```

```
{
delay(250);
delay(250);
lcd_command(0x38);
lcd_command(0x0c);
lcd_command(0x01);
}
-----
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****scankey4x3.h*****/
sbit c1 = P2^6;      // Bit Column
sbit c2 = P2^5;      // Bit Column2
sbit c3 = P2^4;      // Bit Column3
sbit r1 = P2^0;      // Bit Row1
sbit r2 = P2^1;      // Bit Row2
sbit r3 = P2^2;      // Bit Row3
sbit r4 = P2^3;      // Bit Row4

-----

void delay_db(int time)
-----
{
    do          // do-while loop for delay
    {
        time--; // Decrease counter
    }while(time>0); // If time>0 work in block
}

-----

unsigned char scankey(void)
-----
{

    unsigned char ret = 0xFF; // Initial value ret = 0xFF
    do{
        c1 = 0;                // Scan column1
        if(r1==0)              // Check push key 1
        {
            delay_db(30000);    // Delay debounce
            ret = 0x01;         // Return value = 0x01
        }
        if(r2==0)              // Check push key 4

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay_db(30000);    // Delay debounce
    ret = 0x04;        // Return value = 0x04
}
if(r3==0)             // Check push key 7
{
    delay_db(30000);    // Delay debounce
    ret = 0x07;        // Return value = 0x07
}
if(r4==0)
{
    delay_db(30000);
    ret = 0x0A;
}
c1 = 1;              // Stop check Column1
c2 = 0;              // Scan Column2
if(r1==0)            // Check push key 2
{
    delay_db(30000);    // Delay debounce
    ret = 0x02;        // Return value = 0x02
}
if(r2==0)            // Check push key 5
{
    delay_db(30000);    // Delay debounce
    ret = 0x05;        // Return value = 0x05
}
if(r3==0)            // Check push key 8
{
    delay_db(30000);    // Delay debounce
    ret = 0x08;        // Return value = 0x08
}
if(r4==0)            // Check push key 0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    delay_db(30000);    // Delay debounce
    ret = 0x00;        // Return value = 0x00
}
c2 = 1;              // Stop check Column2
c3 = 0;              // Scan Column3
if(r1==0)            // Check push key 3
{
    delay_db(30000);    // Delay debounce
    ret = 0x03;        // Return value = 0x03
}
if(r2==0)            // Check push key 6
{
    delay_db(30000);    // Delay debounce
    ret = 0x06;        // Return value = 0x06
}
if(r3==0)            // Check push key 9
{
    delay_db(30000);    // Delay debounce
    ret = 0x09;        // Return value = 0x09
}
if(r4==0)
{
    delay_db(30000);
    ret = 0x0B;
}
c3 = 1;              // Stop check Column3

}while(ret==0xFF);
return(ret);        // Return key value
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\*\*\*\*\*

TX

\*\*\*\*\*

```

#include<AT89x52.h>

#include <intrins.h> //Keil library (is used for _nop()_ operation)
#include <math.h> //Keil library
#include <stdio.h> //Keil library

sbit control_1 = P2^0;
sbit control_4 = P2^1;

typedef union
{
    unsigned int i;
    float f;
} value;
value humi_val,temp_val;
float Humi,Temp;
unsigned char round,count;

//-----
// modul-var
//-----

enum {TEMP,HUMI};

#define DATA P1_6
#define SCK P1_7

#define noACK 0
#define ACK 1

//adr command r/w

#define STATUS_REG_W 0x06 //000 0011 0
#define STATUS_REG_R 0x07 //000 0011 1
#define MEASURE_TEMP 0x03 //000 0001 1
#define MEASURE_HUMI 0x05 //000 0010 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define RESET    0x1e //000 1111 0
```

```
//-----
char s_write_byte(unsigned char value)
//-----
{
    unsigned char i,error=0;
    for (i=0x80;i>0;i/=2)    //shift bit for masking
    { if (i & value) DATA=1;    //masking value with i , write to SENSI-BUS
      else DATA=0;
      SCK=1;    //clk for SENSI-BUS
      _nop_();_nop_();_nop_();    //puls with approx. 5 us
      SCK=0;
    }
    DATA=1;    //release DATA-line
    SCK=1;    //clk #9 for ack
    error=DATA;    //check ack (DATA will be pulled down by SHT11)
    SCK=0;
    return error;    //error=1 in case of no acknowledge
}

```

```
//-----
char s_read_byte(unsigned char ack)
//-----
{
    unsigned char i,val=0;
    DATA=1;    //release DATA-line
    for (i=0x80;i>0;i/=2)    //shift bit for masking
    { SCK=1;    //clk for SENSI-BUS
      if (DATA) val=(val | i);    //read bit
      SCK=0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA=!ack;           //in case of "ack==1" pull down DATA-Line
SCK=1;               //clk #9 for ack
_nop_();_nop_();_nop_(); //pulswith approx. 5 us
SCK=0;
DATA=1;              //release DATA-line
return val;
}

```

```

//-----
void s_transstart(void)
//-----
{
DATA=1; SCK=0;      //Initial state
_nop_();
SCK=1;
_nop_();
DATA=0;
_nop_();
SCK=0;
_nop_();_nop_();_nop_();
SCK=1;
_nop_();
DATA=1;
_nop_();
SCK=0;
}

```

```

//-----
void s_connectionreset(void)
//-----
{

```

```

    unsigned char i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

DATA=1; SCK=0;          //Initial state
for(i=0;i<9;i++)        //9 SCK cycles
{ SCK=1;
  SCK=0;
}
s_transstart();        //transmission start
}

//-----
char s_measure(unsigned char *p_value, unsigned char *p_checksum, unsigned char mode)
//-----
// makes a measurement (humidity/temperature) with checksum
{
  unsigned error=0;
  unsigned int i;

  s_transstart();      //transmission start
  switch(mode){        //send command to sensor
    case TEMP  : error+=s_write_byte(MEASURE_TEMP); break;
    case HUMI  : error+=s_write_byte(MEASURE_HUMI); break;
    default   : break;
  }
  for (i=0;i<65535;i++) if(DATA==0) break; //wait until sensor has finished the measurement
  if(DATA) error+=1;    // or timeout (~2 sec.) is reached
  *(p_value) =s_read_byte(ACK); //read the first byte (MSB)
  *(p_value+1)=s_read_byte(ACK); //read the second byte (LSB)
  *p_checksum =s_read_byte(noACK); //read checksum
  return error;
}

//-----
void calc_sth15(float *p_humidity ,float *p_temperature)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----
// calculates temperature [?C] and humidity [%RH]
// input : humi [Ticks] (12 bit)
//      temp [Ticks] (14 bit)
// output: humi [%RH]
//      temp [?C]
{ const float C1=-4.0;          // for 12 Bit
  const float C2=+0.0405;      // for 12 Bit
  const float C3=-0.0000028;   // for 12 Bit
  const float T1=+0.01;       // for 14 Bit @ 5V
  const float T2=+0.00008;    // for 14 Bit @ 5V

  float rh=*p_humidity;       // rh: Humidity [Ticks] 12 Bit
  float t=*p_temperature;     // t: Temperature [Ticks] 14 Bit
  float rh_lin;               // rh_lin: Humidity linear
  float rh_true;              // rh_true: Temperature compensated humidity
  float t_C;                  // t_C : Temperature [?C]

  t_C=t*0.01 - 40;           //calc. temperature from ticks to [?C]
  rh_lin=C3*rh*rh + C2*rh + C1; //calc. humidity from ticks to [%RH]
  rh_true=(t_C-25)*(T1+T2*rh)+rh_lin; //calc. temperature compensated humidity [%RH]
  if(rh_true>100)rh_true=100; //cut if the value is outside of
  if(rh_true<0.1)rh_true=0.1; //the physical possible range

  *p_temperature=t_C;        //return temperature [?C]
  *p_humidity=rh_true;       //return humidity[%RH]
}

```

```

//-----
void service_timer0(void) interrupt 1
//-----

```

```

{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char buf[7];
unsigned int i,a,b;
    TH0 = 0X4B;
    TL0 =0xFD;
    round++;
if (round >= 40)
{
    control_1 = 1;
round = 0;
    buf[1] = (((int)Humi)/10);
    buf[2] = (((int)Humi)%10);
    a = Humi*10;
    buf[3] = a%10;
    buf[4]= (((int)Temp)/10);
    buf[5] = (((int)Temp)%10);
    b = Temp*10;
    buf[6] = b%10;
    for(i=1;i<7;i++)
    {
        SBUF = buf[i];
        while(~TI);
        TI = 0;
    }

    control_1 = 0;
}
}

```

```
//-----
```

```
void main(void)
```

```
//-----
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    unsigned char error,checksum;
    round=0;
    count = 0;
    control_1 = 0;
    control_4 = 0;
    s_connectionreset();
    SCON = 0x40;                // Mode serial port
    TMOD = 0x21;
    TH1 = 0xFD;
    TL1 = 0xFD;
    TF1 = 0;
    T1 =0;
    TR1 = 1;
    TH0 = 0x4B;
    TL0 = 0xFD;
    EA = 1;                    // Enable Interrupt
    ET0 = 1;                  // Enable Interrupt timer0
    TR0 = 1;
    while(1)
    {
        error=0;
        error+=s_measure((unsigned char*) &humi_val.i,&checksum,HUMI); //measure
humidity
        error+=s_measure((unsigned char*) &temp_val.i,&checksum,TEMP); //measure
temperature
        if(error!=0) s_connectionreset();        //in case of an error: connection reset
        else
        {
            humi_val.f=(float)humi_val.i;        //converts integer to float

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp_val.f=(float)temp_val.i;           //converts integer to float
    calc_sth15(&humi_val.f,&temp_val.f); //calculate humidity,
temperature
    }

    Temp = temp_val.f;
    Humi = humi_val.f;
    }
}

*****
RX
*****

#include<AT89x52.h>
#include <intrins.h> //Keil library (is used for _nop()_ operation)
#include <math.h> //Keil library
#include <stdio.h> //Keil library
#include <led.h>
#include <i2c.h>
#include <scankey4x3.h>

#define AT24C16_ID 0xA0
sbit IPH = 0xB7;
sbit alarm1 = P1^2;
sbit alarm2 = P1^3;
sbit alarm3 = P1^4;
sbit sensor = P1^0;

code unsigned char ptr1[13] = "HUMI: %RH"; // Store in program memory
code unsigned char ptr2[11] = "TEMP: C"; // Store in program memory
code unsigned char *settemp = "Setting Temp"; // Pointer for screen text "Setting
Temp"
code unsigned char *temp = "_. "; // Pointer for screen
text "_. "

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

unsigned char temp_buf[4];           // For keep Date value from push key
unsigned char serial[7];
unsigned char eep_value[3];
float pid_control,setTm,error_0,T_value,Inte_0,Inte;
unsigned int piddelay,piddelayT,round;
unsigned char page=0 ;
unsigned char eep_value1,eep_value2,eep_value3;
int counter;

```

```
#define OnloadT P3_4
```

```
#define AT24C16_ID 0xA0
```

```
//-----
void EEPROM_WRITE(unsigned char dat1,unsigned char dat2,unsigned char dat3)

```

```
//-----
{
EA = 0;
i2c_start();
i2c_wrddata(AT24C16_ID);
i2c_wrddata(0x00);
i2c_wrddata(dat1);
i2c_wrddata(dat2);
i2c_wrddata(dat3);
i2c_stop();
EA = 1;
}

```

```
//-----
```

```
char EEPROM_READ(unsigned char addr)
```

```
//-----
{

```

```
unsigned char ret;
```

```
EA = 0;
```

```
i2c_start();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
i2c_wrddata(AT24C16_ID);
```

```
i2c_wrddata(addr);
```

```
i2c_start();
```

```
i2c_wrddata(AT24C16_ID+1);
```

```
ret = i2c_rddata();
```

```
i2c_stop();
```

```
EA = 1;
```

```
return(ret);
```

```
}
```

```
//-----
```

```
float pid(float setTm)
```

```
//-----
```

```
{
```

```
    const float kp = +110;
```

```
    const float ki = 0.5;
```

```
    const float kd = 0.1;
```

```
    float error, Inte;
```

```
    error = (setTm - T_value);
```

```
    Inte = ki*((0.1*error) + Inte_0);
```

```
    pid_control = (kp*error) + (ki*((0.1*error) + Inte)) + (kd*(error - error_0)/0.1);
```

```
    if(pid_control > 95)
```

```
        pid_control = 95;
```

```
    if(pid_control < 5)
```

```
        pid_control = 5;
```

```
    error_0 = error;
```

```
    Inte_0 = Inte;
```

```
    return pid_control;
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----
void display_data(void)
//-----
{
unsigned char buffer[15];
unsigned char i,j;
    for(i=0;i<13;i++)
        buffer[i] = *(ptr1+i);
    buffer[5] = serial[1]+'0';
    buffer[6] = serial[2]+'0';
    buffer[7] = '.';
    buffer[8] = serial[3]+'0';
    lcd_jumporigin();
    lcd_command(0x80);
    for(i=0;i<13;i++)
        lcd_text(buffer[i]);

    for(j=0;j<11;j++)
        buffer[j] = *(ptr2+j);
    buffer[5] = serial[4]+'0';
    buffer[6] = serial[5]+'0';
    buffer[7] = '.';
    buffer[8] = serial[6]+'0';
    lcd_jumporigin();
    lcd_command(0xC0);
    for(j=0;j<11;j++)
        lcd_text(buffer[j]);
}

//-----
void display_settemp(void)
//-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    // Keep for couter loop
    //-----Show text screen on LCD("Setting Temp")-----
    unsigned char i;
    lcd_clear(); // Clear LCD display
    lcd_jumporigin(); // Go to origin address
    lcd_command(0x82); // Set LCD address 02H
    for(i=0;i<12;i++) // Loop for show screen "Setting
Time" by pointer settime
    lcd_text(*(settemp+i));
    lcd_jumporigin();
    lcd_command(0xC7);
    lcd_text(0xDF);
    lcd_text('C');
    //-----
    //-----Show text screen on LCD-----
    lcd_jumporigin(); // Go to origin address
    lcd_command(0xC3);
    for(i=0;i<4;i++)
    lcd_text(*(temp+i));
    //-----
    //----- Return to set curcer bring for insert value -----
    lcd_jumporigin(); // Go to origin address
    lcd_command(0x0C); // Display ON and OFF curcer bring
    lcd_command(0xC3); // Set LCD address 43H
    //-----
}

```

```

//-----

```

```

void pushkey()

```

```

//-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
unsigned char x_key,push_key ;
    push_key=0;
    x_key = 0;

    while(x_key < 4)                // Check for end insert setting Temp
    {
        push_key=scankey();
        if(push_key == 0x0A)        // Check cancel
        {
            lcd_clear();
            page = 0;
            break;
        }
        else if(x_key == 3 && push_key==0x0B) // Check enter for keep value
        {
            lcd_clear();
            EEPROM_WRITE(temp_buf[1],temp_buf[2],temp_buf[3]);
            page = 0;
            break;
        }
        else if( x_key < 3 && push_key != 0x0A && push_key != 0x0B ) // Check for
insert setting temp value
        {
            x_key++;                // increase counter 1 time
            temp_buf[x_key] = push_key;        // Keep temp setting for write
to EEPROM

            push_key = push_key | 0x30;        // convert to ascii code
            lcd_text(push_key);

            if(x_key==2)                // IF push key 2 time shift curcer 1 time
            {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_jumporigin();
        lcd_command(0xC6);
    }
}

}

}

//-----
void chk_setting()
//-----
{
unsigned char key=0;
while(1)
{
    key = scankey();
    if((key == 0x0A)||(key == 0x0B))
        break;
}

if(key == 0x0A) //-----1st condition-----
{
    // Check cancel
    lcd_clear();
    display_settemp();
    pushkey();
}

else if(key == 0x0B) //-----2nd condition-----
{
    // Check enter for keep value
    lcd_clear();
    page = 0;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

//-----
void display_setting()
//-----
{
// Show temp value
from EEPROM
unsigned char i,eep_value;
    lcd_jumporigin();
    lcd_command(0xC3);

for(i=0;i<3;i++)
{
    eep_value=EEPROM_READ(i);
    eep_value=eep_value|0x30;
    if(i==2)
    {
        lcd_jumporigin();
        lcd_command(0xC6);
    }
    lcd_text(eep_value);
}

    chk_setting();
}

```

```

//-----
void sevice_int0(void) interrupt 0
//-----

```

{

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

page = 1;                                // Interrupt service of switch checking
}

//-----
void service_timer0(void) interrupt 1
//-----
{

    counter++;
    if (counter == piddelayT)
    {
        TR0 = 0;
        OnloadT = 0;
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        OnloadT = 1;
        counter = -1;
    }
}

//-----
void service_int1(void) interrupt 2
//-----
{
    TR0 = 1;
    piddelayT = piddelay;
}

//-----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void service_serial(void) interrupt 4
```

```
//-----
```

```
{
```

```
EA=0;
```

```
while(~RI);
```

```
RI = 0;
```

```
serial[1] = SBUF ;
```

```
while(~RI);
```

```
RI = 0;
```

```
serial[2] = SBUF ;
```

```
while(~RI);
```

```
RI = 0;
```

```
serial[3] = SBUF ;
```

```
while(~RI);
```

```
RI = 0;
```

```
serial[4] = SBUF ;
```

```
while(~RI);
```

```
RI = 0;
```

```
serial[5] = SBUF ;
```

```
while(~RI);
```

```
RI = 0;
```

```
serial[6] = SBUF ;
```

```
EA=1;
```

```
}
```

```
//-----
```

```
void service_timer2(void) interrupt 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//-----
{
    EA=0;
    TF2 = 0;
    round++;

    if(round >= 1000) //0.1 s
    {
        round = 0;
        T_value = (serial[4]*10) + serial[5] + (serial[6]*0.1);
        eep_value1= EEPROM_READ(0x00);
        eep_value2= EEPROM_READ(0x01);
        eep_value3= EEPROM_READ(0x02);
        setTm = (eep_value1*10) + eep_value2 + (eep_value3*0.1);
        pid_control = pid(setTm);
        piddelay = 100 -(int)pid_control;
    }
    EA=1;
}
//-----
//                               Main program
//-----
void main(void)

```

```

{
    Inte_0 = 0 ;
    error_0 = 0;
    pid_control = 0;
    piddelay = 0;
    counter = -1;
    round = 0;
    setTm = 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

alarm1 = 0;
alarm2 = 0;
alarm3 = 0;
sensor = 1;
lcd_init();
SCON = 0x50;           // Mode serial port
TMOD = 0x22;          // Timer0 16 bit Timer1 8 bit auto reload
TH1 = 0xFD;
TL1 = 0xFD;
TI = 0;                // Clear bit over flag TX
RI = 0;
TF1 = 0;
TR1 = 1;
TH0 = 0xA4;
TF0 = 0;
ET0 = 1;              // Enable Interrupt timer0
T2MOD &= 0xFC;
T2MOD |= 0x01;
T2CON = 0x00;
TH2 = 0x94;
TL2 = 0x00;
RCAP2H = 0x94;
RCAP2L = 0x00;
TF2 = 0;
ET2 = 1;              // Enable Interrupt timer0
TR2 = 1;
PS = 1;
IPH = 0x10;
IT0 = 1;
IT1 = 1;
EX0 = 1;
EXI = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EA = 1;                                // Enable Interupt
ES = 1;

while(1)                                // Infinite loop
{
//-----
//-----Check condition display page0,page1-----
switch(page)                            // Check display page
{
case 0 : display_data();
        if (T_value >= setTm+2)
            alarm1 = 1;
        else alarm1 = 0;
        if (T_value <= setTm-2)
            alarm2 = 1;
        else alarm2 = 0;
        if (sensor == 0)
            alarm3 = 1;
        else alarm3 = 0;
        break;

case 1 : display_settemp();              // Show text "setting Temp"
        display_setting();                // Setting Temp

        break;

} //end switch
} //end while
} //end main program

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

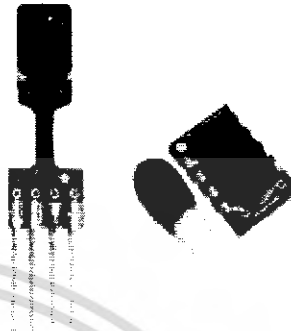


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 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.

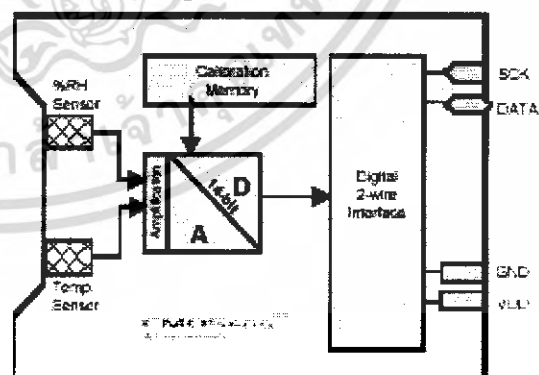
### Applications

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

### 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

### Schematic Diagram



### 1 Sensor Performance Specifications<sup>1)</sup>

Parameter	Conditions	Min.	Typ.	Max.	Units
<b>Humidity</b>					
Resolution		0.5	0.03	0.03	% RH
		8	12	12	bit
Repeatability			±0.1		% RH
Accuracy <sup>2)</sup> & Interchangeability		see figure 1			
Nonlinearity	10 - 90 %RH	<1 <sup>3)</sup>		±3 <sup>4)</sup>	% RH
Range		0		100	% RH
Response time	1τe (63%) steady moisture air		4		s
Hysteresis			±1		% RH
Long term stability	Typical		< 1		% RH/yr
<b>Temperature</b>					
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	1τe (63%)	5		30	s

Table 1 Sensor Performance Specifications

#### 1.1 Converting the digital output to physical values

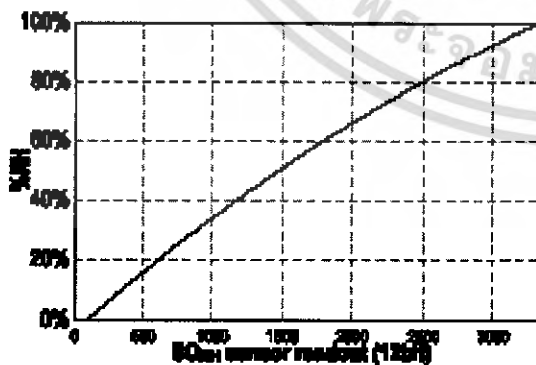
##### 1.1.1 Humidity

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:

$$RH_{linear} = d_1 + d_2 \cdot SO_{RH} + d_3 \cdot (SO_{RH})^2$$

$d_1 = -4$     $d_2 = 0.0405$     $d_3 = -2.8 \cdot 10^{-6}$    for 12bit  $SO_{RH}$   
 $d_1 = -4$     $d_2 = 0.648$     $d_3 = -7.2 \cdot 10^{-4}$    for 8bit  $SO_{RH}$

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



<sup>1)</sup> For operation within normal operation range as described in Chapter 3. RH accuracy at 25°C

<sup>2)</sup> Not including non-linearity

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

<sup>4)</sup> Where  $SO_{RH}$  is the sensor output for relative humidity

<sup>5)</sup> Min. value after compensation with formula in chapter 1.1.1. Max. value without any compensation

Relative Humidity absolute accuracy

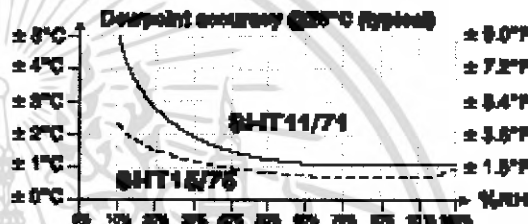
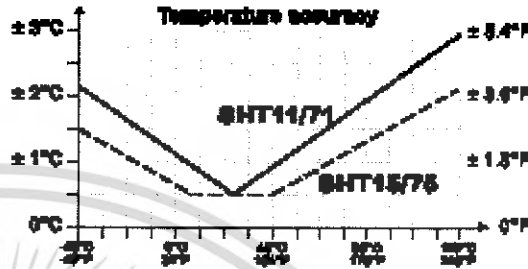
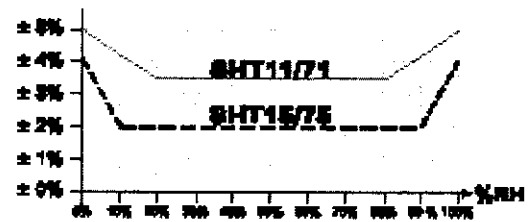


Figure 1 Rel Humidity, Temperature and Dewpoint accuracies

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

$$RH_{temp} = (T_{RH} - 25) \cdot (t_1 + t_2 \cdot SO_{RH}) + RH_{linear}$$

with  $t_1 = 0.01$ ;  $t_2 = 0.00008$  for 12bit  $SO_{RH}$ ;  $t_2 = 0.00128$  for 8bit  $SO_{RH}$   
 This equals  $-0.12\%RH / ^\circ C @ 50\%RH$

##### 1.1.2 Temperature

The temperature sensor is very linear by design. Use the following formula to convert from digital readout to temperature:  $Temperature = d_1 + d_2 \cdot SO_T$   
 Use the appropriate table entries for 5V or 3V.

$SO_T$	Celsius		Fahrenheit	
	$d_1$	$d_2$	$d_1$	$d_2$
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  $-0.2^{\circ}C/V @ 25^{\circ}C$

##### 1.1.3 Dewpoint

See application note "Dewpoint calculation" for more information.

## 2 Serial Interface

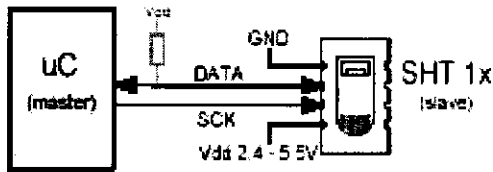


Figure 2 Typical application circuit

### 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 content.

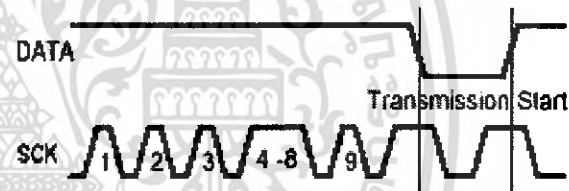
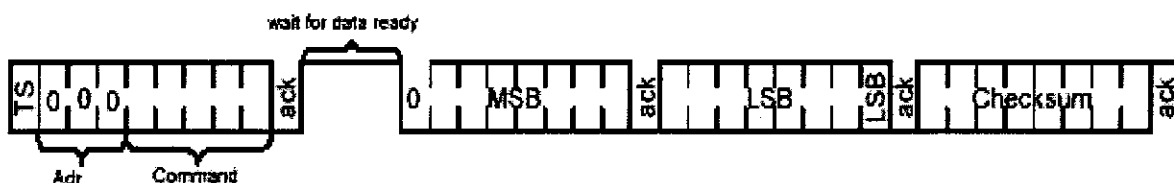


Figure 4 Connection reset sequence

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 ไม่ว่าจะกรณีใดๆ 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  $\pm 15\%$  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 5<sup>th</sup> 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.

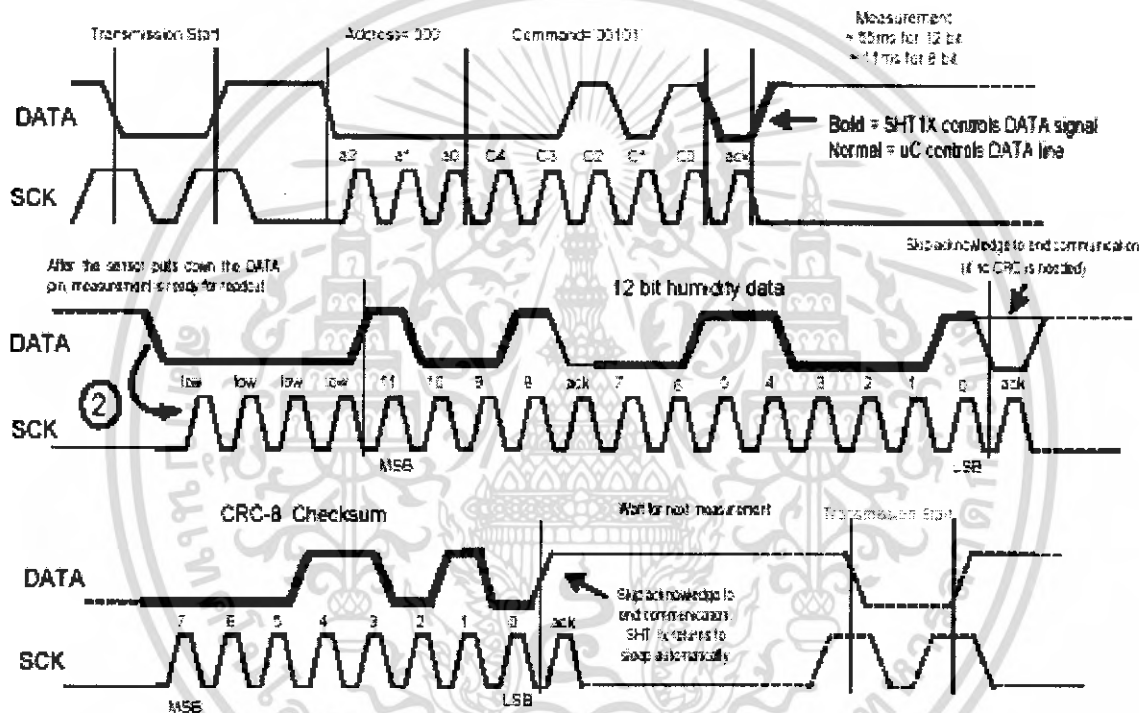


Figure 5 Example RH measurement sequence for value '0000'1001' 0011'0001' = 2353 = 75.75%RH

## 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  $\pm 0.05V$

### 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 sensimeter is heated!

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า 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

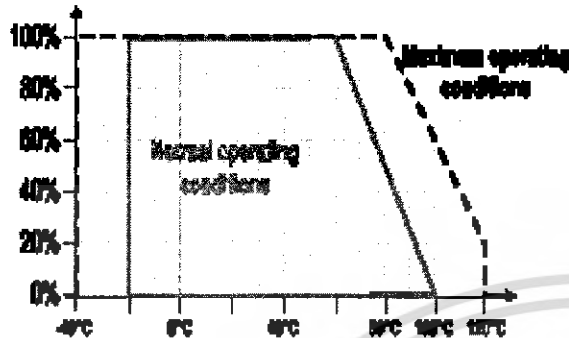


Figure 6 Recommended operating conditions

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 <sup>(1)</sup>
Temperature Cycles	JESD22-A104-A -40 +125°C, 1000cy	Within Specifications
Pressure Cooker	JESD22-A110-B 2.3bar 125°C 95%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<sup>(1)</sup>

#### 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<sub>amb</sub>=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	270	280		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 <sub>SCK</sub>	VDD > 4.5 V			10	MHz
	VDD < 4.5 V			1	MHz
T <sub>FHD</sub>	Output load 5 pF	3.5	10	20	ns
	Output load 100 pF	30	40	200	ns
T <sub>CLH</sub>	SCK high time		100		ns
T <sub>CLL</sub>	SCK low time		100		ns
T <sub>V</sub>	DATA valid time		250		ns
T <sub>SU</sub>	DATA set up time		100		ns
T <sub>HD</sub>	DATA hold time	0	10		ns
T <sub>R/TF</sub>	SCK rise/fall time			200	ns

Table 5 SHT1x/SHT7x I/O Signals Characteristics

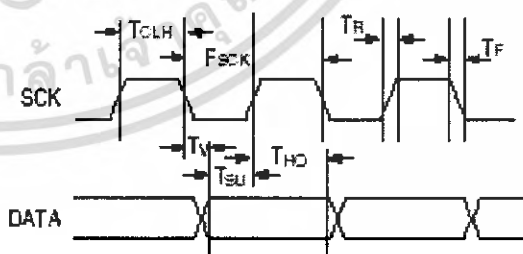


Figure 7 Timing Diagram

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

## 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 0.6mm 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

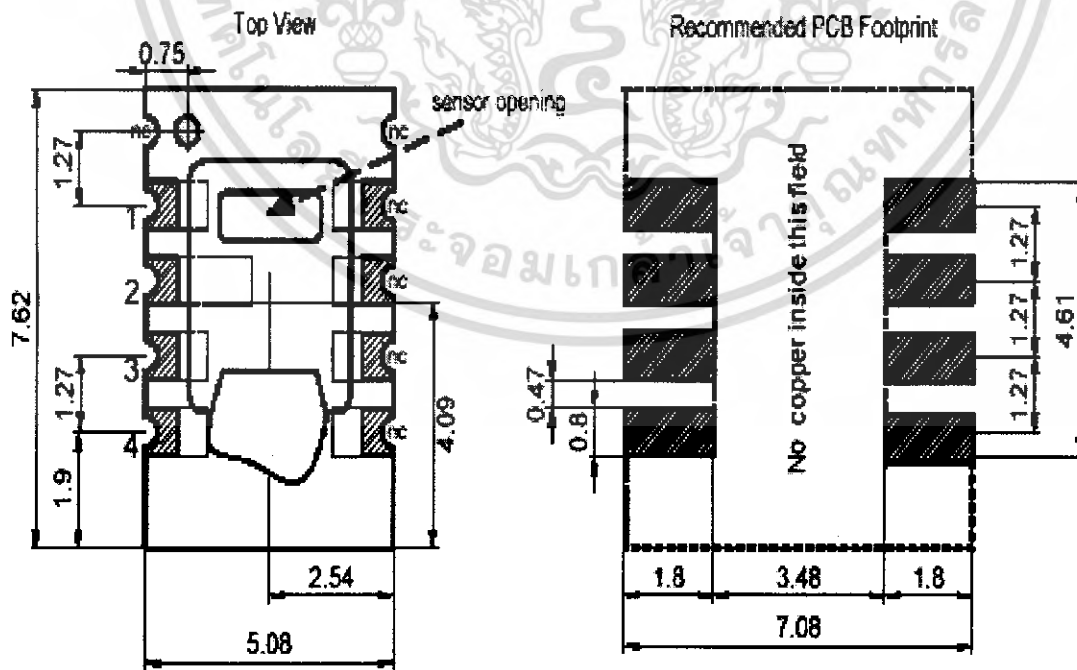
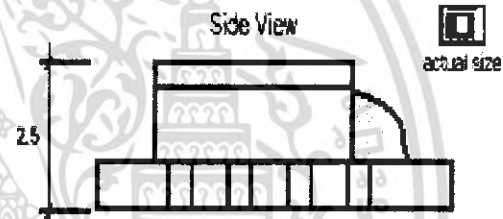


Figure 9 SHT1x drawing and footprint dimensions in mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ 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.

## 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

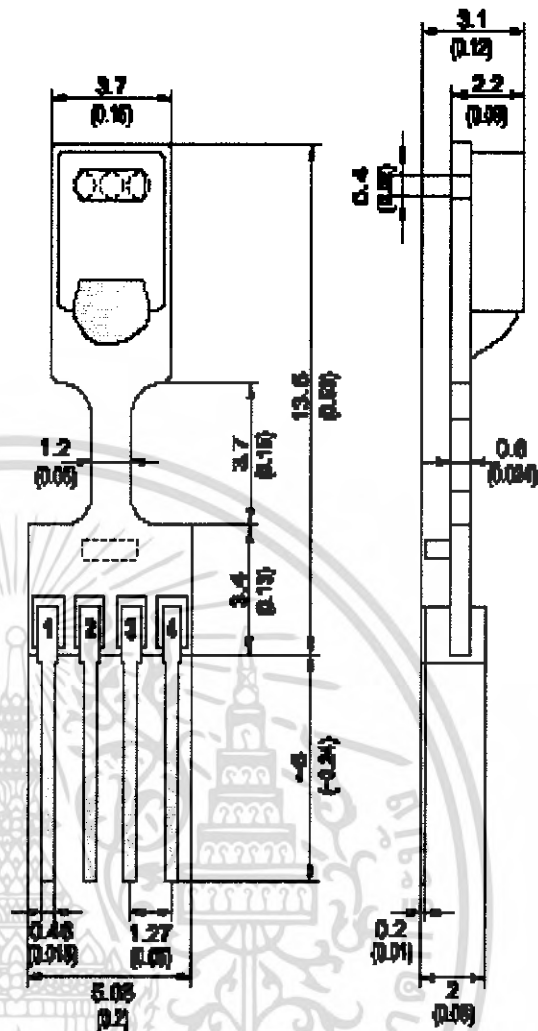


Figure 10 SHT7x dimensions in mm (inch), preliminary

heat conduction copper layers between the SHT7x and the rest of the PCB should be minimized and a slit may be milled in between.

Prolonged direct exposure of the SHT7x/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.

## 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%
	6	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](http://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.**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้