



การสร้างชุดควบคุมความชื้น โดยการปรับอัตราการไหลของอากาศที่  
อีวาโปเรเตอร์

HUMUDITY CONTROL BY ADJUST  
AIR FLOW RATE AT EVAPORATOR

นาย สาคร บิดา  
นาย พยุง ยอดจันทร์  
นาย สิทธิศักดิ์ สวัสดิ์ผล

|                           |
|---------------------------|
| ห้องสมุดคณะวิศวกรรมศาสตร์ |
| วันที่รับ.....            |
| เลขทะเบียน..... 5200077   |
| เลขเรียกหนังสือ.....      |
| มหาวิทยาลัยธนบุรี         |

15014239 c.2  
ป.ร.  
สขก  
2551

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยธนบุรี

ปีการศึกษา 2551



### ใบรับรองโครงการวิศวกรรมเครื่องกล

หัวข้อโครงการ : การสร้างชุดควบคุมความชื้นโดยการปรับอัตราการไหล  
ของอากาศที่ี้อวโปเรเตอร์

HUDUMIDY CONTROL BY ADJUST

AIR FLOW RATE AT EVAPORATOR

ผู้ดำเนินโครงการ : นาย ศาคร บิดา รหัส 48380115

นาย พยุ่ง ยอดจันทร์ รหัส 48380222

นาย สิทธิศักดิ์ สวัสดิ์ผล รหัส 48380264

อาจารย์ที่ปรึกษาโครงการ : อาจารย์ นินนาท ราชประดิษฐ์

อาจารย์ผู้ช่วยที่ปรึกษาโครงการ : อาจารย์ ปองพันธ์ โอทกานนท์

สาขาวิชา : วิศวกรรมเครื่องกล

ภาควิชา : วิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติโครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตาม  
หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมเครื่องกล

คณะกรรมการสอบโครงการ

.....ประธานกรรมการ

(อาจารย์ นินนาท ราชประดิษฐ์)

๑๒๐๗

๑๖

.....กรรมการ

(อาจารย์ ปองพันธ์ โอทกานนท์)

.....กรรมการ

(อาจารย์ สิทธิโชค ผูกพันธ์)

๑๖

๑๖

.....กรรมการ

(อาจารย์ สุรัตน์ ปัญญาแก้ว)

## ใบรับรองโครงการ

|                                |                                                                          |           |               |
|--------------------------------|--------------------------------------------------------------------------|-----------|---------------|
| หัวข้อโครงการ                  | : การสร้างชุดควบคุมความชื้นโดยการปรับอัตราการใช้ของอากาศที่ีวาลไปเรเตอร์ |           |               |
| ผู้ดำเนินโครงการ               | นาย ศาคร                                                                 | บิดา      | รหัส 48380115 |
|                                | นาย พยุง                                                                 | ยอดจันทร์ | รหัส 48380222 |
|                                | นาย สิทธิศักดิ์                                                          | สวัสดิ์ผล | รหัส 48380264 |
| อาจารย์ที่ปรึกษาโครงการ        | : อาจารย์ นินนาท ราชประดิษฐ์                                             |           |               |
| อาจารย์ผู้ช่วยที่ปรึกษาโครงการ | : อาจารย์ ปองพันธ์ โอทกานนท์                                             |           |               |
| สาขาวิชา                       | : วิศวกรรมเครื่องกล                                                      |           |               |
| ภาควิชา                        | : วิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์มหาวิทยาลัยนเรศวร                   |           |               |

### บทคัดย่อ

โครงการนี้มีวัตถุประสงค์เพื่อเป็นแนวทางในการควบคุมความชื้น โดยการสร้างชุดควบคุมเพื่อปรับความเร็วของพัดลมที่ีวาลไปเรเตอร์ให้เหมาะสมกับความชื้นของห้องโดยอัตโนมัติ และใช้โปรแกรม CCS C compiler 3.202, MPLAB และ Proteus 7 Professionals ช่วยในการออกแบบ

จากการทดลองพบว่า การปรับอากาศโดยใช้ชุดควบคุมความชื้นอากาศให้มีปริมาณความชื้นสัมพัทธ์ต่ำสุดที่ 44 %RH และสูงสุดที่ 53 %RH ซึ่งมีช่วงการแกว่งของความชื้นที่ประมาณ 44 – 53 %RH และในการปรับอากาศโดยไม่ใช้ชุดควบคุมความชื้นนั้น อากาศจะมีปริมาณความชื้นสัมพัทธ์ต่ำสุดประมาณ 37 %RH และสูงสุดที่ 44 %RH จะมีช่วงการแกว่งของความชื้นสัมพัทธ์อยู่ที่ประมาณ 37 – 44 %RH

การใช้ชุดควบคุมความชื้นจะส่งผลให้ความชื้นของห้องที่ปรับอากาศไม่ลดต่ำจนเกินไป ซึ่งจะช่วยในการควบคุมสภาวะความสบายของมนุษย์ และเพื่อเป็นแนวทางในการนำไปประยุกต์ใช้กับการควบคุมความเร็วของพัดลมที่ีวาลไปเรเตอร์ของเครื่องปรับอากาศต่อไป

**Project Title** : HUDUMIDY CONTROL BY ADJUST  
AIR FLOW RATE AT EVAPORATOR

**Name** : Mr. Sacon Bida Code 48380115  
Mr. Payoong Yodjun Code 48380222  
Mr. Sittisak Sawatdeepon Code 48380264

**Project Advisor** : Mr. Ninnart Ratchapradit  
Mr. Pongpun Otakanon

**Department** : Mechanical Engineering

**Academic** : 2008

---

### Abstract

This project has two primary purposes are to create a course of humidity control, by making instrument for tune a velocity of fan to proper with a room humidity is automatically, and use a several programs consist of CCS C compiler 3.202, MPLAB, and Proteus 7 Professionals for use to creating.

When we are looking for the experiment we will find out that the air-conditioning by using an instrument control a humidity of air which makes result to a lowest relative humidity at 44 %RH and a highest at 53 %RH, and a variable range of humidity is about 44 – 53 %RH. In the other ways, air-conditionings that are without an instrument control a humidity of air will make result to a lowest relative humidity at 37 %RH and highest at 44 %RH and a variable range of humidity is about 37 – 44 %RH.

Hence, find that this using an instrument control a humidity of air will make result to the humidity of room is proper and not too lower for a comfortableness of human and it is an interest for applied with a fan velocity control that is the Evaporator fan motor of air-condition in the further.

design

## กิตติกรรมประกาศ

คณะผู้ดำเนินโครงการขอขอบพระคุณคณะบุคคล และหน่วยงานที่เกี่ยวข้องที่ได้ให้ความอนุเคราะห์ ให้คำปรึกษาให้โครงการประสบความสำเร็จลุล่วงไปได้ด้วยดี ซึ่งได้แก่

- 1.อาจารย์นันทา ราชประดิษฐ์ อาจารย์ที่ปรึกษาโครงการ
- 2.อาจารย์ปองพันธ์ โอทกานนท์ อาจารย์ผู้ช่วยที่ปรึกษาโครงการ
- 3.คณาจารย์ทุกท่านที่ได้ให้ความรู้ตลอดจนคำแนะนำ
- 4.มหาวิทยาลัยนเรศวร
- 5.สมาชิกในกลุ่มและเพื่อนทุกคน

รวมทั้งขอกราบขอบพระคุณบิดามารดา รวมทั้งผู้มีพระคุณทุกท่านที่ได้กล่าวนามไว้ ณ ที่นี้ ที่ให้การสนับสนุนและเป็นกำลังใจให้เสมอมา



ศาสตราจารย์

พญ.

สิทธิศักดิ์

บิดา

ยอดจันทร์

สวัสดิ์ผล

## สารบัญ

|                                                | หน้า |
|------------------------------------------------|------|
| ใบรับรองโครงงาน                                | ก    |
| บทคัดย่อ                                       | ข    |
| Abstract                                       | ค    |
| กิตติกรรมประกาศ                                | ง    |
| สารบัญ                                         | จ    |
| สารบัญตาราง                                    | ช    |
| สารบัญรูปภาพ                                   | ซ    |
| สารบัญกราฟ                                     | ฅ    |
| ลำดับสัญลักษณ์                                 | ฉ    |
| บทที่ 1 บทนำ                                   | 1    |
| 1.1 ที่มาและความสำคัญของโครงงาน                | 1    |
| 1.2 วัตถุประสงค์ของโครงงาน                     | 1    |
| 1.3 ขอบเขตของโครงงาน                           | 2    |
| 1.4 ขั้นตอนการดำเนินงาน                        | 2    |
| 1.5 ผลที่คาดว่าจะได้รับ                        | 2    |
| บทที่ 2 หลักการและทฤษฎี                        | 3    |
| 2.1 หลักการของระบบปรับอากาศ                    | 3    |
| 2.2 การทำความเย็นและลดความชื้นที่อิวาโปเรเตอร์ | 4    |
| 2.3 ความชื้นสัมพัทธ์ (Relative Humidity)       | 6    |
| 2.4 กฎของฟัดลม                                 | 7    |
| 2.5 ความสบายเชิงความร้อน (Thermal Comfort)     | 7    |
| 2.6 ไมโครคอนโทรลเลอร์                          | 8    |
| 2.7 ทฤษฎี Pulse Width Modulation (PWM)         | 10   |
| 2.8 Humidity and Temperature Sensor (SHT15)    | 14   |
| บทที่ 3 วิธีการดำเนินงาน                       | 17   |
| 3.1 ขั้นตอนและวิธีการดำเนินงาน                 | 17   |

## สารบัญ ( ต่อ )

|                                                                   | หน้า |
|-------------------------------------------------------------------|------|
| 3.2 อุปกรณ์และเครื่องมือวัด                                       | 18   |
| 3.3 ขั้นตอนการทดลอง                                               | 19   |
| 3.4 การสร้างชุดควบคุมความเร็วพัดลม                                | 20   |
| 3.4.1 การใช้โปรแกรม Compiler CCS Version 3.202                    | 20   |
| 3.4.2 ใช้โปรแกรม MPLAB IDE v7.01                                  | 25   |
| 3.4.3 ใช้โปรแกรม Proteus 7 Professional                           | 31   |
| 3.4.4 ขั้นตอนการสร้างแผงวงจร                                      | 41   |
| บทที่ 4 ผลการดำเนินงานและการวิเคราะห์                             | 53   |
| 4.1 กราฟเปรียบเทียบ Duty Cycle กับความเร็วรอบ                     | 53   |
| 4.2 กราฟเปรียบเทียบ Duty Cycle กับอัตราไหลของอากาศ                | 54   |
| 4.3 อัตราไหลของอากาศที่ได้จากการทดลองเปรียบเทียบกับทฤษฎี          | 55   |
| 4.4 กราฟเปรียบเทียบความเร็วรอบของมอเตอร์และอัตราไหลอากาศ          | 56   |
| 4.5 ความชื้นสัมพัทธ์ ณ เวลาต่างๆของห้องปรับอากาศที่อุณหภูมิ 25 °C | 57   |
| บทที่ 5 บทสรุป                                                    | 59   |
| 5.1 สรุป                                                          | 59   |
| 5.2 ข้อเสนอแนะ                                                    | 60   |
| บรรณานุกรมและหนังสืออ้างอิง                                       | 61   |
| ภาคผนวก                                                           | 62   |
| ภาคผนวก ก อุปกรณ์ที่ใช้สร้างชุดควบคุมความเร็วพัดลม                | 63   |
| ภาคผนวก ข ข้อเสนอแนะในการเขียนโปรแกรม, การออกแบบโครงงาน           | 77   |
| ภาคผนวก ค Code ของโปรแกรม PWC C Compiler IDE                      | 81   |
| ภาคผนวก ง การตั้งค่าอุปกรณ์                                       | 94   |
| ประวัติผู้ทำโครงงาน                                               | 96   |

## สารบัญตาราง

|                                                                      | หน้า |
|----------------------------------------------------------------------|------|
| ตารางที่ 2.1 แสดงค่าคงที่ของ d1                                      | 15   |
| ตารางที่ 2.2 แสดงค่าคงที่ของ d2                                      | 15   |
| ตารางที่ 2.3 แสดงค่าคงที่ C1 และ C2                                  | 16   |
| ตารางที่ 2.4 แสดงค่าคงที่ t1 และ t2                                  | 16   |
| ตารางที่ 4.3 ค่าของอัตราการใช้ของอากาศจากการทดลองเปรียบเทียบกับทฤษฎี | 55   |



## สารบัญรูปภาพ

|                                                                     | หน้า |
|---------------------------------------------------------------------|------|
| รูปที่ 2.1 วัฏจักรการทำความเย็นแบบอัดไอ                             | 4    |
| รูปที่ 2.2 กระบวนการทำความเย็นและลดความชื้น                         | 5    |
| รูปที่ 2.3 แสดง PWM Duty Cycle                                      | 10   |
| รูปที่ 2.4 การใช้โมดูล CCP1 และ TIMER2 สร้างสัญญาณ PWM              | 11   |
| รูปที่ 2.5 แสดงการกำหนด Duty Cycle                                  | 11   |
| รูปที่ 2.6 แสดงการต่อ Sensor                                        | 14   |
| รูปที่ 3.1 ห้องทดลองและตำแหน่งเครื่องมือวัด                         | 19   |
| รูปที่ 3.4.1 แสดงหน้าต่างเริ่มต้นโปรแกรม                            | 21   |
| รูปที่ 3.4.2 แสดง Manual Project Create                             | 22   |
| รูปที่ 3.4.3 แสดงการตั้งชื่อไฟล์                                    | 22   |
| รูปที่ 3.4.4 แสดงการเลือก Target Device                             | 23   |
| รูปที่ 3.4.5 แสดงหน้าต่างสำหรับเขียน Source Code                    | 23   |
| รูปที่ 3.4.6 แสดง Flow Chart แสดงการทำงานของชุดควบคุมความเร็วพัดลม  | 24   |
| รูปที่ 3.4.7 แสดงการเปิดโปรแกรม MPLAB IDE V7.01                     | 25   |
| รูปที่ 3.4.8-3.4.9 แสดงหน้าต่างเมื่อเปิดโปรแกรม MPLAB IDE V7.01     | 26   |
| รูปที่ 3.4.10 แสดงการกำหนดค่าความถี่ของ CRYSTAL                     | 26   |
| รูปที่ 3.4.11 แสดงการเลือกอุปกรณ์ PIC18F452                         | 27   |
| รูปที่ 3.4.12 แสดงการ Set Language Tool Location                    | 27   |
| รูปที่ 3.4.13 แสดงการกำหนดชื่อและตำแหน่งของไฟล์                     | 28   |
| รูปที่ 3.4.14 แสดงหน้าต่างทำงานหลังการกำหนดชื่อและที่อยู่ไฟล์       | 28   |
| รูปที่ 3.4.15 แสดงการเปิดไฟล์ขึ้นมาใช้งาน                           | 29   |
| รูปที่ 3.4.16 แสดงการเลือก Build All                                | 29   |
| รูปที่ 3.4.17 แสดงการ Select Programmer                             | 30   |
| รูปที่ 3.4.18 แสดงการต่อวงจรควบคุมความเร็วพัดลม                     | 30   |
| รูปที่ 3.4.19 แสดงหน้าต่างเริ่มต้นของโปรแกรม Proteus 7.0 Profession | 31   |
| รูปที่ 3.4.20 แสดงการเลือกอุปกรณ์ในโปรแกรม Proteus 7.0 Profession   | 31   |

## สารบัญรูปภาพ ( ต่อ )

|                                                                                   | หน้า |
|-----------------------------------------------------------------------------------|------|
| รูปที่ 3.4.21 แสดงการใส่ Key word และเลือกอุปกรณ์ในโปรแกรม Proteus 7.0 Profession | 32   |
| รูปที่ 3.4.22 แสดงรายชื่ออุปกรณ์ที่ใช้ต่อวงจร                                     | 32   |
| รูปที่ 3.4.23 แสดงการวางตำแหน่งของ PIC18F425 ในวงจร                               | 33   |
| รูปที่ 3.4.24 แสดงการวางจอแสดงผลชนิด LCD ของวงจร                                  | 33   |
| รูปที่ 3.4.25 แสดงการต่อวงจรของ LCD กับ PIC18F425                                 | 34   |
| รูปที่ 3.4.26 แสดงการต่อวงจรของ LCD กับ PIC18F425                                 | 34   |
| รูปที่ 3.4.27 แสดงการต่อสายดินและ Power ให้กับจอแสดงผล LCD                        | 35   |
| รูปที่ 3.4.28 แสดงตำแหน่งการวางของ SHT15 ของวงจร                                  | 36   |
| รูปที่ 3.4.29 แสดงการต่อวงจรของ SHT15 กับ PIC18F452 ของวงจร                       | 36   |
| รูปที่ 3.4.30 แสดงการต่อตัวต้านทานและ Power ให้กับ SHT15                          | 37   |
| รูปที่ 3.4.31 แสดงการต่อตัวต้านทานและ Power ให้กับ PIC18F452                      | 37   |
| รูปที่ 3.4.32 แสดงการจัดตำแหน่งและต่อวงจรของ CAP และ CRYSTAL                      | 38   |
| รูปที่ 3.4.33 แสดงการจัดตำแหน่งและต่อวงจรของ CAP และ CRYSTAL                      | 38   |
| รูปที่ 3.4.34 แสดงการจัดตำแหน่งของตัวต้านทาน (R3), TIP122 และ Motor               | 39   |
| รูปที่ 3.4.35 แสดงวงจรที่ต่อเสร็จสมบูรณ์                                          | 40   |
| รูปที่ 3.4.36 แสดงตำแหน่งขาของ PIC18F452                                          | 41   |
| รูปที่ 3.4.37 แสดงการบัดกรีขาของ PIC18F452 เข้ากับบอร์ดเนกประสงค์                 | 42   |
| รูปที่ 3.4.38 แสดงการบัดกรี Terminal, L7805C, TIP122 และ LM317T                   | 42   |
| รูปที่ 3.4.39 แสดงการตัดเส้นทองแดง T2, T3                                         | 43   |
| รูปที่ 3.4.40 แสดงการบัดกรีตัวเก็บประจุเข้ากับบอร์ดเนกประสงค์                     | 43   |
| รูปที่ 3.4.41 แสดงการใช้สายไฟอ่อนบัดกรีจาก L7805C เพื่อส่งกระแสไฟฟ้าให้กับ IC     | 44   |
| รูปที่ 3.4.42 แสดงการบัดกรีขา 30 เข้ากับขา 11 และ ขา 31 เข้ากับ ขา 12 ของ IC      | 44   |
| รูปที่ 3.4.43 แสดงการบัดกรี CRYSTAL ต่อกับขา 13 และ 14 ของ IC                     | 45   |
| รูปที่ 3.4.44 แสดงการต่อตัวต้านทาน 290R, 390R และสายไฟ                            | 45   |
| รูปที่ 3.4.45 แสดงการต่อสายไฟ 12 V เข้ากับ LM317T และ ขั้ว adj เข้ากับ 220R       | 46   |
| รูปที่ 3.4.46 แสดงการต่อ Terminal 4 PIN สำหรับต่อ Sensor SHT15                    | 46   |

## สารบัญรูปภาพ ( ต่อ )

|                                                                                                      | หน้า |
|------------------------------------------------------------------------------------------------------|------|
| รูปที่ 3.4.47 แสดงการต่อสายไฟขั้วบวกจาก LM317 เข้ากับ PIN บวก                                        | 47   |
| รูปที่ 3.48 แสดงการต่อตัวต้านทาน 10K เข้ากับ สาย DATA                                                | 47   |
| รูปที่ 3.49 แสดงการตัดเส้นทองแดง ระหว่าง ขา 1 กับขั้ว ลบ และ ขา 4 กับขั้วบวก                         | 48   |
| รูปที่ 3.50 แสดงการต่อตัวต้านทาน 10k ระหว่างขา 1 กับขา 11                                            | 48   |
| รูปที่ 3.51 แสดงการต่อตัวต้านทาน 10k ระหว่างขา 17 เข้ากับขา Base ของ TIP 122                         | 49   |
| รูปที่ 3.52 แสดงการต่อสายไฟ gnd จาก L7805C เข้ากับ ขา emitter<br>และ collector เข้ากับ terminal (T3) | 49   |
| รูปที่ 3.53 แสดงการวาง socket สำหรับ LCD พร้อมกับตัดเส้นทองแดง                                       | 50   |
| รูปที่ 3.54 แสดงการต่อสายไฟสำหรับส่งข้อมูลให้ LCD                                                    | 50   |
| รูปที่ 3.55 แสดงแผงวงจรที่ต่อเสร็จสมบูรณ์                                                            | 51   |
| รูปที่ 3.56 แสดงภาพด้านหลังของแผงวงจรที่ต่อเสร็จสมบูรณ์                                              | 52   |
| รูปที่ ก 1 Peripheral Interface Controller (PIC)                                                     | 64   |
| รูปที่ ก 2 Features Include                                                                          | 64   |
| รูปที่ ก 3 Pin Outs                                                                                  | 65   |
| รูปที่ ก 4 จอแสดงผล (LCD)                                                                            | 65   |
| รูปที่ ก 5 Monolithic Regulator                                                                      | 66   |
| รูปที่ ก 6 การต่อวงจร โดยใช้ Monolithic Regulator                                                    | 66   |
| รูปที่ ก 7 Monolithic Regulator ในรูปของไอซี (MOTOROLA MC7800C SERIES)                               | 67   |
| รูปที่ ก 8 โครงสร้างของทรานซิสเตอร์ชนิด NPN และ PNP                                                  | 68   |
| รูปที่ ก 9 สัญลักษณ์ที่ใช้แทนทรานซิสเตอร์ในวงจรอิเล็กทรอนิกส์                                        | 68   |
| รูปที่ ก 10 ตัวต้านทานชนิดค่าคงที่ (Fixed Resistor)                                                  | 72   |
| รูปที่ ก 11 โครงสร้างของตัวต้านทานชนิดค่าคงที่                                                       | 72   |
| รูปที่ ก 12 ตัวต้านทานปรับค่า (Adjustable Resistor)                                                  | 73   |
| รูปที่ ก 13 ตัวต้านทานชนิดแปลงค่า (Variable Resistor)                                                | 74   |
| รูปที่ ก 14 ชุดความคุมความเร็วพัดลม                                                                  | 75   |
| รูปที่ ก 15 อุปกรณ์เพิ่มความชื้น                                                                     | 75   |

## สารบัญรูปภาพ (ต่อ)

|                                      | หน้า |
|--------------------------------------|------|
| รูปที่ ก 16 ภาพอุปกรณ์ด้านหน้า       | 76   |
| รูปที่ ก 17 ภาพอุปกรณ์ด้านหลัง       | 76   |
| รูปที่ ง 1 เครื่องโปรแกรมข้อมูลลง IC | 95   |



## สารบัญกราฟ

|                                                                  | หน้า |
|------------------------------------------------------------------|------|
| กราฟที่ 4.1 กราฟเปรียบเทียบ Duty Cycle กับความเร็วรอบ            | 52   |
| กราฟที่ 4.2 กราฟเปรียบเทียบ Duty Cycle กับอัตราไหลของอากาศ       | 53   |
| กราฟที่ 4.3 อัตราไหลของอากาศที่ได้จากการทดลองเปรียบเทียบกับทฤษฎี | 54   |
| กราฟที่ 4.4 กราฟเปรียบเทียบความเร็วรอบของมอเตอร์และอัตราไหลอากาศ | 55   |
| กราฟที่ 4.5 ความชื้นสัมพัทธ์ที่เวลาต่างๆของห้องปรับอากาศ         | 56   |



## ลำดับสัญลักษณ์

| สัญลักษณ์ | ความหมาย                                                | หน่วย                   |
|-----------|---------------------------------------------------------|-------------------------|
| BHP       | Break Horse Power                                       | hp                      |
| Cfm       | อัตราการไหลเชิงปริมาตรของอากาศ                          | ft <sup>3</sup> /min    |
| E         | แรงดันที่ตกคร่อมตัวต้านทาน                              | W                       |
| f         | ความถี่                                                 | Hz                      |
| fosc      | ความถี่ XTAL                                            | Hz                      |
| $h_a$     | ค่าของเอนทัลปีของอากาศชื้น                              | kJ/kg <sub>dryair</sub> |
| $h_{ai}$  | เอนทัลปีของอากาศชื้นที่ทางเข้าอุปกรณ์แลกเปลี่ยนความร้อน | kJ/kg <sub>dryair</sub> |
| $h_{ao}$  | เอนทัลปีของอากาศชื้นที่ทางออกอุปกรณ์แลกเปลี่ยนความร้อน  | kJ/kg <sub>dryair</sub> |
| I         | ค่ากระแสไฟฟ้า                                           | A                       |
| $m_a$     | อัตราการไหลเชิงมวลของอากาศ                              | kg/s                    |
| Q         | อัตราการไหลเชิงปริมาตรของอากาศ                          | m <sup>3</sup> /s       |
| $Q_l$     | อัตราการถ่ายเทความร้อนแฝง                               | kW                      |
| $Q_s$     | อัตราการถ่ายเทความร้อนสัมผัส                            | kW                      |
| R         | ความต้านทาน                                             | Ohm                     |
| RPM       | ความเร็วรอบของพัดลม                                     | m/s                     |
| T         | คาบเวลา                                                 | S                       |
| $t_a$     | อุณหภูมิของอากาศชื้น                                    | °C                      |
| $t_{ai}$  | อุณหภูมิของอากาศชื้นที่ทางเข้าอีวาโปเรเตอร์             | °C                      |
| $t_{ao}$  | อุณหภูมิของอากาศชื้นที่ทางออกอีวาโปเรเตอร์              | °C                      |
| Tosc      | คาบเวลาของ PWM                                          | S                       |
| V         | ศักย์ไฟฟ้า                                              | W                       |
| W         | Humidity ratio ของอากาศชื้น                             | kg/kg <sub>dryair</sub> |
| $W_i$     | Humidity ratio ของอากาศชื้นที่ทางเข้าอีวาโปเรเตอร์      | kg/kg <sub>dryair</sub> |
| $W_o$     | Humidity ratio ของอากาศชื้นที่ทางออกอีวาโปเรเตอร์       | kg/kg <sub>dryair</sub> |

## บทที่ 1

### บทนำ

#### 1.1 ที่มาและความสำคัญของโครงการ

เนื่องจากประเทศไทยตั้งอยู่ในเขตร้อนชื้นจึงทำให้อากาศมีอุณหภูมิสูงตลอดทั้งปี เครื่องปรับอากาศจึงเป็นอุปกรณ์ที่ช่วยอำนวยความสะดวกในการปรับอากาศให้เหมาะสมกับสภาวะความสบายของผู้อยู่อาศัย ซึ่งเครื่องปรับอากาศในปัจจุบันสามารถปรับสภาวะอากาศได้โดยการควบคุมอุณหภูมิตามที่กำหนด แต่ยังไม่สามารถควบคุมความชื้นสัมพัทธ์ของอากาศให้คงที่หรือให้เหมาะสมกับสภาวะความสบายของมนุษย์ได้

โครงการนี้จึงทำการศึกษาเกี่ยวกับการหาแนวทางควบคุมความชื้น โดยการสร้างชุดทดลองเพื่อควบคุมความเร็วพัลลมด้วยวิธีพัลส์วidthมอดูเลชั่น (Pulse Width Modulation) ซึ่งจะนำไปใช้ปรับอัตราการไหลของอากาศ ในส่วนของชุดควบคุมมีเซนเซอร์ (Sensor) วัดค่าความชื้น ซึ่งสัญญาณที่ได้จะถูกส่งไปที่ชุดควบคุมเพื่อปรับความเร็วของพัลลมให้เหมาะสมกับความชื้นสัมพัทธ์ของอากาศโดยอัตโนมัติ ซึ่งจะนำไปสู่การปรับอากาศให้มีอุณหภูมิและความชื้นที่เหมาะสมกับความสบายของผู้อาศัยต่อไป

#### 1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อศึกษาแนวทางการควบคุมความชื้นของอากาศ โดยการสร้างชุดควบคุมความเร็วพัลลมเพื่อนำไปใช้ปรับอัตราการไหลของอากาศที่อีวาโปเรเตอร์

1.2.2 เพื่อนำไปประยุกต์ใช้งานในระบบปรับอากาศ

### 1.3 ขอบเขตของโครงการ

- 1.3.1 ใช้การควบคุมแบบพัลส์วิดมอดูเลชั่น (Pulse Width Modulation) ในการปรับอัตราการไหลของอากาศ
- 1.3.2 ใช้พัลลัม DC 12 V รุ่น C 8025 12 V และ 0.25 A
- 1.3.3 ห้องที่ใช้ทำการทดลองมีขนาดด้านกว้าง 3 เมตร ยาว 4 เมตร สูง 2.2 เมตร ผนังทั้ง 4 ด้านเป็นคอนกรีตหนา 10 ซม. และเพดานเป็นฝ้าแบบแขวน (T-bar) ใช้แผ่นยิปซัมเป็นวัสดุ
- 1.3.4 อุณหภูมิที่ทำการทดลอง 25-26 °C
- 1.3.5 เครื่องปรับอากาศที่ใช้คือ TOSHIBA ขนาด 12000 BTU/hr (เบอร์ 5)

### 1.4 ขั้นตอนการดำเนินงาน

| การดำเนินงาน         | 2551         |             |            | 2552  |       |
|----------------------|--------------|-------------|------------|-------|-------|
|                      | มิ.ย. – ก.ค. | ส.ค. – ต.ค. | พ.ย – ก.พ. | มี.ค. | เม.ย. |
| 1. ศึกษาข้อมูล       | ↔            |             |            |       |       |
| 2. สร้างชุดควบคุม    |              | ↔           |            |       |       |
| 3. ทดลองและบันทึกผล  |              |             | ↔          |       |       |
| 4. วิเคราะห์การทดลอง |              |             |            | ↔     |       |
| 5. จัดทำรูปเล่ม      |              |             |            |       | ↔     |

ตารางที่ 1.1 แสดงขั้นตอนการดำเนินงาน

### 1.5 ผลที่คาดว่าจะได้รับ

- 1.5.1 ชุดควบคุมอัตราการไหลของอากาศ
- 1.5.2 ได้แนวทางในการควบคุมปริมาณความชื้นสัมพัทธ์ของอากาศ

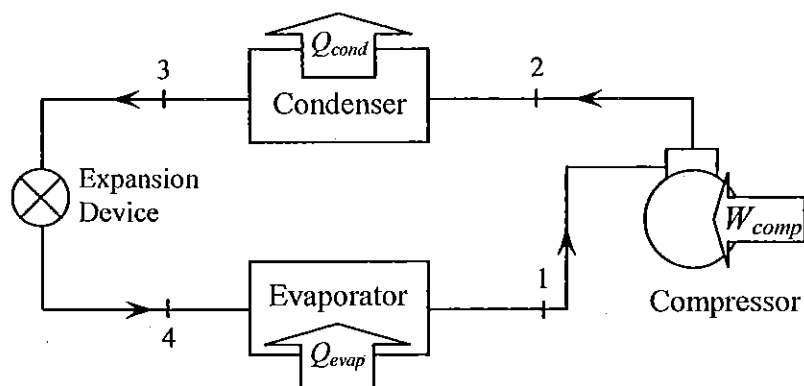
## บทที่ 2

### หลักการและทฤษฎี

การทำงานของเครื่องปรับอากาศ มีลักษณะการทำงานเป็นวัฏจักร การปรับอากาศ มีความสัมพันธ์กับสภาวะความสบายเชิงความร้อนของมนุษย์ ซึ่งมีความเกี่ยวเนื่องกับความชื้นสัมพัทธ์ของอากาศ ดังนั้นการควบคุมปริมาณความชื้นของอากาศ ส่งผลต่อสภาวะความสบายของมนุษย์ ซึ่งในบทนี้อธิบายเกี่ยวกับ หลักการทำงานของเครื่องปรับอากาศ, ความสบายเชิงความร้อน, ความชื้นสัมพัทธ์ของอากาศ และการเขียนวงจรควบคุมความชื้นของอากาศดังต่อไปนี้

#### 2.1 หลักการของระบบปรับอากาศ

หลักการทำงานของเครื่องปรับอากาศ พิจารณาจากตำแหน่งที่ 1 คอมเพรสเซอร์ดูดสารทำความเย็นที่อยู่ในสถานะ ไออิ่มตัวเข้าไป และบีบอัดสารทำความเย็นแบบไอเซนโทรปิก (Isentropic Process) ซึ่งจะส่งผลให้สารทำความเย็นมีอุณหภูมิและความดันสูงขึ้น จากนั้นสารทำความเย็นที่อยู่ในสถานะก๊าซ ไหลผ่านตำแหน่งที่ 2 เข้าสู่คอนเดนเซอร์ และขณะที่สารทำความเย็นไหลผ่านคอนเดนเซอร์จะเกิดการแลกเปลี่ยนความร้อนกับอากาศ ซึ่งส่งผลให้สารทำความเย็นมีอุณหภูมิลดลงจนกระทั่งเกิดการกลั่นตัวและเปลี่ยนสถานะเป็นของเหลวอิ่มตัว และเมื่อออกจากคอนเดนเซอร์ ณ ตำแหน่งที่ 3 ซึ่งในช่วงนี้สารทำความเย็นมีอุณหภูมิและความดันสูง จึงจำเป็นต้องถูกบีบให้ไหลผ่านอุปกรณ์ throttling (เช่น วาล์วขยายตัว หรือท่อคาปิลารี) จนกระทั่งความดันและอุณหภูมิลดต่ำลง ในตำแหน่งที่ 4 นั้นสารทำความเย็นที่อยู่ในสถานะของผสมอิ่มตัวที่มีคุณภาพไอลำไหลผ่านอีวาโปเรเตอร์ และเกิดการแลกเปลี่ยนความร้อนกับอากาศ ซึ่งส่งผลให้สารทำความเย็นเปลี่ยนสถานะเป็นไออิ่มตัว หลังจากนั้นสารทำความเย็นจะถูกดูดเข้าคอมเพรสเซอร์และทำงานตามวัฏจักรต่อไป



รูปที่ 2.1 วงจรการทำความเย็นแบบอัดไอ

## 2.2 การทำความเย็นและลดความชื้นที่อีวาโปเรเตอร์

เนื่องจากในอากาศโดยปกติมีลักษณะเป็นอากาศชื้นที่มีไอน้ำผสมอยู่ ในการพิจารณาคุณสมบัติจำเป็นต้องพิจารณารวมถึงไอน้ำที่ประกอบอยู่ด้วย ซึ่งค่าของเอนทัลปีของอากาศ หาได้จากสมการ [1] ต่อไปนี้

$$h_a = t_a + W(2501 + 1.805t_a) \quad (2.1)$$

โดยที่

- $h_a$  คือ ค่าของเอนทัลปีของอากาศชื้น,  $\text{kJ/kg}_{\text{dryair}}$
- $t_a$  คือ อุณหภูมิของอากาศชื้น,  $^{\circ}\text{C}$
- $W$  คือ Humidity ratio ของอากาศชื้น,  $\text{kg/kg}_{\text{dryair}}$

เมื่ออากาศไหลผ่านอีวาโปเรเตอร์จะเกิดกระบวนการทำความเย็นและลดความชื้นในอากาศโดยที่สามารถหาอัตราการถ่ายเทความร้อน เมื่ออากาศไหลผ่านอุปกรณ์แลกเปลี่ยนความร้อนได้จากสมการ

$$Q = m_a(h_{ai} - h_{ao}) \quad (2.2)$$

โดยที่

- $m_a$  คือ อัตราการไหลเชิงมวลของอากาศ,  $\text{kg/s}$
- $h_{ai}$  คือ เอนทัลปีของอากาศชื้นที่ทางเข้าอุปกรณ์แลกเปลี่ยนความร้อน,  $\text{kJ/kg}_{\text{dryair}}$
- $h_{ao}$  คือ เอนทัลปีของอากาศชื้นที่ทางออกอุปกรณ์แลกเปลี่ยนความร้อน,  $\text{kJ/kg}_{\text{dryair}}$

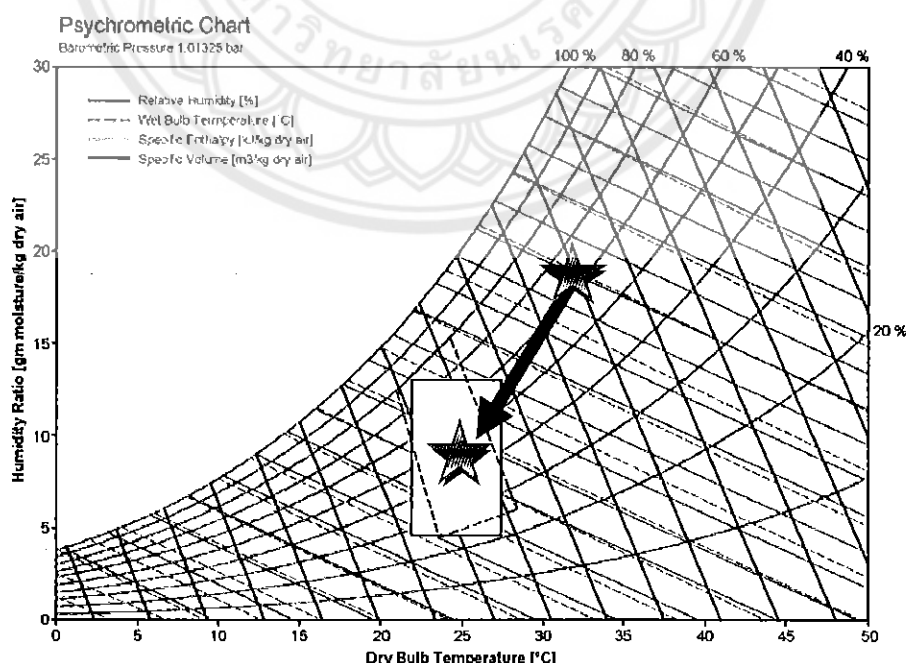
ในระบบปรับอากาศเพื่อที่จะรักษาสภาวะของพื้นที่ให้มีอุณหภูมิและความชื้นตามที่ต้องการ ปริมาณความร้อนสัมผัส (sensible heat) และความร้อนแฝง (latent heat) ต้องกำจัดออกจากอากาศที่จ่ายให้กับห้องที่ปรับอากาศ ในการกำจัดความร้อนสัมผัสนั้น อากาศที่จ่ายเข้าไปต้องเย็นกว่าอุณหภูมิห้องที่ได้ออกแบบ และในการกำจัดความร้อนแฝง อากาศที่จ่ายเข้าไปต้องมีจุดน้ำค้างต่ำกว่าจุดน้ำค้างของห้องที่ได้ออกแบบไว้เช่นกัน ความร้อนภายในห้องที่ถูกกำจัดออกมาโดยอากาศที่ทำการปรับสภาวะแล้ว ประกอบไปด้วยความร้อนสัมผัสของห้อง (room sensible heat) และความร้อนแฝงของห้อง (room latent heat) ในการรักษาสภาวะของห้องที่ออกแบบไว้ ไม่เพียงแต่ต้องกำจัดปริมาณความร้อนทั้งหมดที่ห้องได้รับในเวลานั้น แต่จำเป็นต้องกำจัดปริมาณความร้อนสัมผัสและความร้อนแฝงในอัตราส่วนที่พอเหมาะ ซึ่งอัตราส่วนความร้อนสัมผัสต่อความร้อนทั้งหมด เรียกว่า ค่าความร้อนสัมผัส (Sensible-Heat Factor, SHF) ซึ่งแสดงได้ดังสมการต่อไปนี้

$$SHF = \frac{Q_s}{Q_s + Q_l} \quad (2.3)$$

โดยที่

$Q_s$  คือ อัตราการถ่ายเทความร้อนสัมผัส, kW

$Q_l$  คือ อัตราการถ่ายเทความร้อนแฝง, kW



รูปที่ 2.2 กระบวนการทำความเย็นและลดความชื้น

หากพิจารณาขบวนการทำความเย็นและลดความชื้นที่อีวาโปเรเตอร์ ปริมาณความร้อนสัมผัส และปริมาณความร้อนแฝงที่ถูกกำจัดที่อีวาโปเรเตอร์ สามารถหาค่าได้จากสมการ [1]

$$Q_s = m_a C_p (t_{ai} - t_{ao}) \quad (2.4)$$

และ

$$Q_l = m_a h_{fg} (W_i - W_o) \quad (2.5)$$

โดยที่

|          |                                                                                  |
|----------|----------------------------------------------------------------------------------|
| $t_{ai}$ | คือ อุณหภูมิของอากาศชั้นที่ทางเข้าอีวาโปเรเตอร์, °C                              |
| $t_{ao}$ | คือ อุณหภูมิของอากาศชั้นที่ทางออกอีวาโปเรเตอร์, °C                               |
| $W_i$    | คือ Humidity ratio ของอากาศชั้นที่ทางเข้าอีวาโปเรเตอร์, kg/ kg <sub>dryair</sub> |
| $W_o$    | คือ Humidity ratio ของอากาศชั้นที่ทางออกอีวาโปเรเตอร์, kg/ kg <sub>dryair</sub>  |
| $C_p$    | คือ ค่าสัมประสิทธิ์การถ่ายเทความร้อน, kJ/kg.K                                    |
| $h_{fg}$ | คือ ค่าเอนทัลปีของอากาศในสถานะของผสม (2501 kJ/kg)                                |

## 2.3 ความชื้นสัมพัทธ์ (Relative Humidity) [2]

ความชื้นสัมพัทธ์คือ อัตราส่วนโดยมวลของไอน้ำในอากาศในขณะหนึ่งต่อไอน้ำสูงสุดที่อากาศสามารถรับไว้ได้ อากาศที่มีปริมาณความชื้นสัมบูรณ์เป็น 100% หมายถึงอากาศ "อิ่มตัว" (Saturated Air) ซึ่งไม่สามารถรับปริมาณไอน้ำเพิ่มได้ และการมีไอน้ำปริมาณมากเกินไปในอากาศไอน้ำส่วนที่เกินนั้นจะ "กลั่นตัว" (Condense) ออกมาเป็นหยดน้ำเหลือไว้แต่เพียงปริมาณไอน้ำที่ยังคงอิ่มตัวและมีความชื้นสัมพัทธ์เป็น 100%

## 2.4 กฎของพัดลม (Fan Law) [2]

กฎของพัดลมเป็นการแสดงความสัมพันธ์ระหว่างประสิทธิภาพของพัดลมกับความเร็วรอบเมื่อต้องทำงานที่สภาวะที่แตกต่างกัน หรือพัดลมมีขนาดที่แตกต่างกัน ซึ่งความสัมพันธ์สามารถพิจารณาได้จากสมการดังต่อไปนี้

$$2.4.1. \text{CFM แปรผันตาม RPM จะได้} \quad \frac{CFM_2}{CFM_1} = \frac{RPM_2}{RPM_1} \quad (2.6)$$

$$2.4.2. \text{SP แปรผันตาม RPM}^2 \text{ จะได้} \quad \frac{SP_2}{SP_1} = \left( \frac{RPM_2}{RPM_1} \right)^2 \quad (2.7)$$

$$2.4.3. \text{BHP แปรผันตาม RPM}^3 \text{ จะได้} \quad \frac{BHP_2}{BHP_1} = \left( \frac{RPM_2}{RPM_1} \right)^3 \quad (2.8)$$

## 2.5 ความสบายเชิงความร้อน (Thermal Comfort)

คำว่า Thermal Comfort มีความหมายที่ถูกกำหนดโดย ASHRAE Standard 51-1981 [1] ว่า “The condition of mind that expresses satisfaction with the thermal environment” ซึ่งมีนัยว่าสภาวะทางจิตใจที่คนส่วนใหญ่มีความพึงพอใจกับสภาพอากาศนั้นๆ ความพึงพอใจในสภาพอากาศใดๆ นั้น ถือเป็นเรื่องของแต่ละบุคคล [2] ซึ่งมีความแตกต่างกัน ในส่วนของปัจจัยที่มีผลต่อสภาวะความสบายในทางกายภาพนั้นมีองค์ประกอบหลัก [3] ดังนี้

- อุณหภูมิอากาศ (Air Temperature)
- อุณหภูมิจากรังสีความร้อน (Mean Radiant Temperature)
- ความชื้นสัมพัทธ์ (Relative Humidity)
- ความเร็วลม (Air Velocity)
- กิจกรรมของบุคคล (Activity or Metabolic Rate)
- เสื้อผ้าของบุคคล (Clothing Thermal Resistance)

## 2.6 ไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) มาจากคำ 2 คำ คือ ไมโคร (Micro) หมายถึงขนาดเล็ก และคำว่า คอนโทรลเลอร์ (controller) หมายถึงตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้น ไมโครคอนโทรลเลอร์ จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ กล่าวคือภายใน ไมโครคอนโทรลเลอร์ ได้รวมอุปกรณ์ต่างๆ อาทิเช่น ซีพียู, หน่วยความจำ และพอร์ต ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยทำการบรรจุไว้ในไมโครคอนโทรลเลอร์ (Microcontroller)

### 2.6.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

โครงสร้างโดยทั่วไป ของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ๆ ดังต่อไปนี้

2.6.1.1 หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit)

2.6.1.2 หน่วยความจำ (Memory) สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ของเครื่องคอมพิวเตอร์ คือข้อมูลใดๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไป อีกส่วนหนึ่งคือหน่วยความจำข้อมูล (Data Memory) เปรียบเหมือนกระดานทดในการคำนวณของซีพียู และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน หากไม่มีกระแสไฟฟ้า ข้อมูลก็จะหายไปคล้ายกับหน่วยความจำ (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีกระแสไฟฟ้า และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีกระแสไฟฟ้า

2.6.1.3 ส่วนเชื่อมต่อกับอุปกรณ์ภายนอก หรือพอร์ต (Port) มี 2 ลักษณะคือ พอร์ตอินพุต (Input Port) และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต (Output Port) ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ นำไปประมวลผลและส่งไปพอร์ตเอาต์พุต เพื่อแสดงผลเช่น การคิดสว่างของหลอดไฟ เป็นต้น

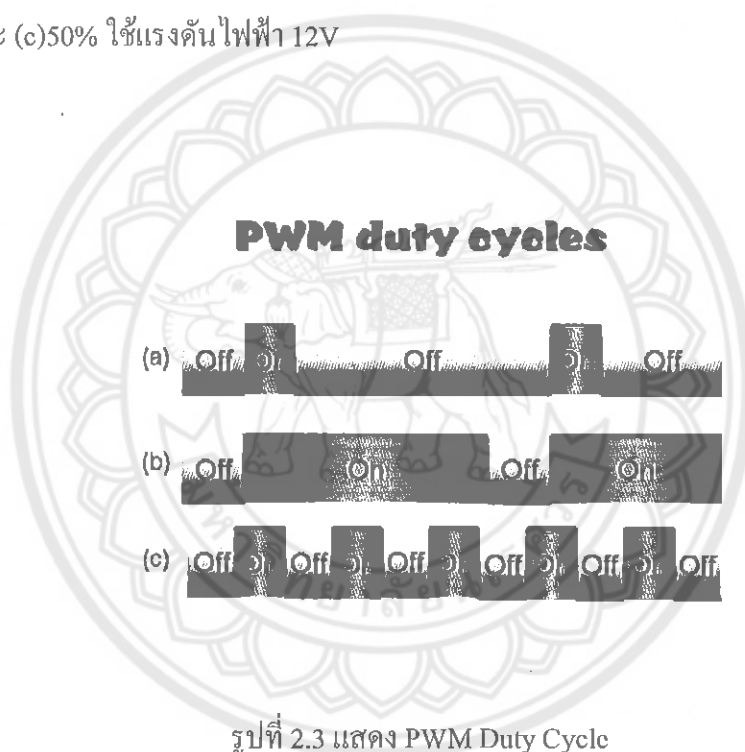
2.6.1.4 ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูล

ระหว่าง ซีพียู หน่วยความจำและพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัว ไมโครคอนโทรลเลอร์ โดยแบ่งเป็นบัสข้อมูล ( Data Bus ) , บัสแอดเดรส ( Address Bus ) และ บัสควบคุม Control Bus ) “บัสข้อมูล” เป็นสายสัญญาณที่บรรจุข้อมูล เพื่อการประมวลผล ทั้งหมด ขนาดของบัสขึ้นอยู่กับความสามารถการประมวลผลของซีพียู สำหรับในงานทั่วไป ขนาดของบัสข้อมูลเป็น 8 บิต และในปัจจุบันได้มีการพัฒนาขึ้นมาจนถึง 16,32 และ 64 บิต “บัสแอดเดรส” เป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับ หน่วยความจำนั้น ซีพียู ต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ดังนั้นจำนวน สายสัญญาณของแอดเดรสจึงต้องมีจำนวนมาก ยิ่งมากเท่าไร ก็จะเป็นการแสดง ขนาดของ หน่วยความจำที่ไมโครคอนโทรลเลอร์สามารถติดต่อได้ โดยสามารถคำนวณได้จาก จำนวน แอดเดรสของหน่วยความจำ = 2 ยกกำลัง  $n$  (  $n$  คือจำนวนของเส้นทาง ) “ บัสควบคุม” เป็นกลุ่ม ของสายสัญญาณควบคุมการติดต่อทั้งหมดของซีพียูกับหน่วยความจำและพอร์ต สำหรับ สายสัญญาณเลือกควบคุมหลักได้แก่ สายสัญญาณเลือก-อ่าน-เขียนหน่วยความจำ สายสัญญาณ เลือกเลือก อ่าน-เขียน ข้อมูลกับพอร์ต

2.6.1.5 วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์ จะขึ้นอยู่กับข้อกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็สามารถทำได้ถี่ขึ้น ส่งผลให้ไมโครคอนโทรลเลอร์นั้น มีความเร็วในการประมวลผลสูง

## 2.7 ทฤษฎี Pulse Width Modulation (PWM)

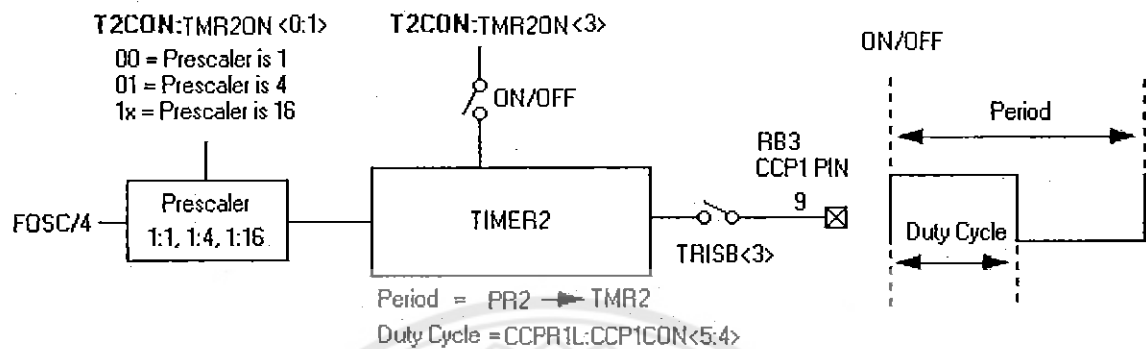
PWM เป็นอีกวิธีหนึ่งที่นิยมใช้ในการควบคุมความเร็วมอเตอร์กระแสตรง (DC) หลักการทำงานก็คือกระแสไฟฟ้าที่ป้อนเข้ามอเตอร์จะเป็นพัลส์ โดยจะมีความถี่คงที่ แต่ความกว้างของพัลส์นั้นสามารถเปลี่ยนแปลงได้ ถ้าพัลส์มีความกว้างมากความเร็วมอเตอร์จะมีค่าสูง (แรงดันไฟฟ้าเฉลี่ยมาก) แต่ถ้าพัลส์แคบความเร็วมอเตอร์จะต่ำ (แรงดันไฟฟ้าเฉลี่ยน้อย) ซึ่งความกว้างของพัลส์จะแสดงในรูปของเปอร์เซ็นต์ที่เรียกว่า Duty Cycle ตัวอย่างในรูปจะแสดง Duty Cycle ที่ (a) 20% (b) 80% และ (c) 50% ใช้แรงดันไฟฟ้า 12V



รูปที่ 2.3 แสดง PWM Duty Cycle

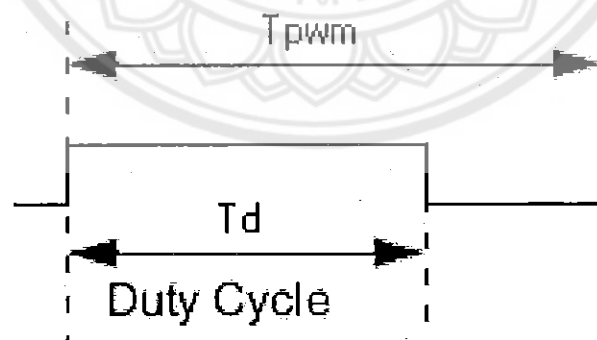
สัญญาณ PWM ที่สร้างขึ้นในการทดลองครั้งนี้จะสร้างจาก MCU (Micro Control Unit) ซึ่งได้เลือกใช้ PIC18F452 เนื่องจากมีโมดูล CCP (CCP ประกอบด้วย 3 โหมดการทำงานคือ Capture/Compare/PWM) ในการสร้างสัญญาณ PWM ได้ 2 ช่องทาง และโมดูล CCP มี Registry ขนาด 16 Bit เพื่อใช้ในการทำงานโหมด PWM และจะทำงานร่วมกับ Timer2 โดยในการกำหนดให้ CCP ทำงานในโหมด PWM นั้นสามารถที่จะสร้าง Duty Cycle ได้

### 2.7.1 การใช้โมดูล CCP1 และ TIMER2 สร้างสัญญาณ PWM



รูปที่ 2.4 การใช้โมดูล CCP1 และ TIMER2 สร้างสัญญาณ PWM

จากรูปสังเกตได้ว่า Timer2 ได้รับสัญญาณนาฬิกา FOSC/4 จากระบบ โดยสามารถกำหนด Prescaler ได้เอง ส่วน PR2 จะเป็นตัวกำหนดคาบเวลา และ CCPR1 มีขนาด 16 Bit ใช้ในการควบคุม duty cycle ประกอบด้วย CCPR1H และ CCPR1L จากนั้นจึงทำการสร้างสัญญาณ PWM ซึ่งในวิธีนี้จะเป็นการกำหนด Duty Cycle แบบคงที่แต่ในการทดลองนี้จะแบบไม่กำหนดค่าคงที่ให้กับ Duty Cycle



รูปที่ 2.5 แสดงการกำหนด Duty Cycle

## 2.7.2 ขั้นตอนการใช้งาน CCP1 กำเนิดสัญญาณ

2.7.2.1 การหาคาบเวลาของ PWM (PWM PERIOD) เพื่อกำหนดค่าให้กับ PR2  
กำหนดความถี่ที่ต้องการจะได้คาบเวลาของพัลส์ แปรผกผันกับความถี่ และ Duty Cycle  
สามารถหาช่วงความถี่ในการใช้งานซึ่งถูกกำหนดโดย XTAL

$$T = \frac{1}{f} \quad (2.9)$$

$$T = T_{osc} \times 4 \times P \times (\text{period} + 1) \quad (2.10)$$

T คือ คาบเวลา, S

f คือ ความถี่, Hz

P คือ Prescaler ของ timer 2

Period คือ คาบเวลาเพื่อ Reset สัญญาณนาฬิกา Timer2 มีค่า 0-255

Tosc คือ คาบเวลาของ PWM, S

$$T_{osc} = \frac{1}{f_{osc}} \quad (2.11)$$

fosc คือ ความถี่ XTAL, Hz

2.7.2.2 การหาความถี่ต่ำสุดที่สามารถใช้งานได้

จากสมการที่ (2.10) กำหนด P=16 period = 255

$$T = T_{osc} \times 4 \times 16 \times (255 + 1) = 16384 T_{osc} \quad (2.12)$$

จากสมการที่ (2.11) ใช้ค่า XTAL=20 MHz สามารถสร้างสัญญาณที่กว้างที่สุด

$$T_{osc} = \left(\frac{1}{20}\right)^6 = 5 \times 10^{-8} \text{ S} \quad (2.13)$$

$$T = 16384 \times 5 \times 10^{-8} = 8.192 \times 10^{-4} \text{ หรือ } 0.8 \text{ ms} \quad (2.14)$$

$$\text{จากสมการที่ (2.14) ค่าความถี่ต่ำสุด } f = \frac{1}{0.0008192} = 1220.7 \text{ Hz} \quad (2.15)$$

2.7.2.3 การหาความถี่สูงสุดที่สามารถใช้งานได้

จากสมการที่ (2.10) กำหนด P=1, period = 0

$$T = T_{osc} \times 4 \times 1 \times (0 + 1) = 4 T_{osc} \quad (2.16)$$

จากสมการที่ (2.11) หากใช้ XTAL=20 MHz สามารถสร้างสัญญาณที่กว้างที่สุด

$$T_{osc} = \left(\frac{1}{20}\right)^6 = 5 \times 10^{-8} \text{ s} \quad (2.17)$$

$$T = 4 \times 5 \times 10^{-8} = 2 \times 10^{-7} \text{ หรือ } 0.2 \text{ us (หรือความถี่ต่ำที่สุด } = 1/T = 5 \text{ MHz)}$$

จากสมการที่ (2.9) ความถี่ต่ำสุด  $f = \frac{1}{0.2 \times 10^{-6}} = 5 \text{ MHz}$  (2.18)

2.7.2.4) การหาค่ารีจิสเตอร์ Period (PR2) เพื่อกำหนดคาบเวลาของ TIMER2 เพื่อใช้ความถี่ หาได้จากสูตร

$$\text{Period} = \left( \frac{T}{T_{osc} \times 4 \times P} \right) - 1 \quad (2.19)$$

2.7.2.5) การคำนวณหา Value เพื่อกำหนดค่าการทำงานในโหมด PWM

จากฟังก์ชัน Setup\_pwm1\_duty (value)

$$\text{Value} = \frac{T_{dx} f_{osc}}{P} \quad (2.20)$$

$$T_d = \frac{T_x \text{DutyCycle}}{100} \quad (2.21)$$

การใช้งาน PWM สร้างความถี่ 5 KHz duty cycle 50% เมื่อใช้ XTAL มีความถี่ 20MHz

จากสมการที่ (2.19)  $\text{Period} = \left( \frac{1/5000}{(1/20 \times 10^6) \times 4 \times 4} \right) - 1 = 249$  (2.22)

จากสมการที่ (2.21)  $T_d = \frac{2 \times 10^{-4} \times 50}{100} = 0.1 \text{ ms}$  (2.23)

จากสมการที่ (2.20)  $\text{value} = \frac{0.0001 \times 20 \times 10^6}{4} = 500$  (2.24)

ผลลัพธ์ที่ได้

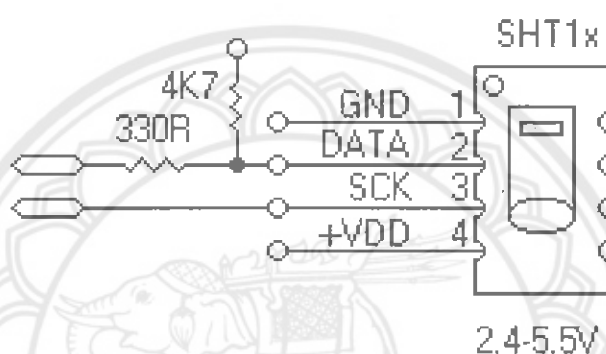
```
setup_ccp1 (CCP_PWM);
```

```
setup_timer_2 (T2_DIV_BY_4, 249, 1);
```

```
setup_pwm1_duty (value);
```

## 2.8 Humidity and Temperature Sensor (SHT15)

การใช้งาน Humidity and Temperature Sensor SHT15 เป็นอุปกรณ์วัดอุณหภูมิและความชื้นสัมพัทธ์ โดยรับสัญญาณแบบดิจิทัลแล้วนำมาแปลงเป็นค่าอุณหภูมิและความชื้นสัมพัทธ์ เพื่อแสดงผลไปยังหน้าจอ LCD



รูปที่ 2.6 แสดงการต่อ Sensor

### 2.8.1 PIN Assignment

2.8.1.1 Ground (GND) เป็นส่วนที่ต่อลงสายกราวด์

2.8.1.2 Serial data (DATA) เป็นสายข้อมูล โดยสายนี้จะต้องต่อตัวต้านทานไฟฟ้าขนาด 4.7k-10k เข้ากับ VDD

2.8.1.3 Serial Clock Input (SCK) เป็นสัญญาณนาฬิกา synchronize ระหว่างไมโครคอนโทรลเลอร์กับ Sensor

2.8.1.4 VDD เป็นสายไฟฟ้าขั้วบวก

## 2.8.2 การเปลี่ยนข้อมูลดิจิทัล (digital) มาเป็นข้อมูลของอุณหภูมิ

ค่าของข้อมูล Temperature ที่ได้จากตัวเซนเซอร์ มีลักษณะเป็นเชิงเส้นสามารถคำนวณได้โดยใช้สูตร

$$T = d1 + d2 \times SO_T \quad (2.25)$$

โดยที่

T คือ อุณหภูมิ, °C หรือ °F

d1, d2 คือ ค่าคงที่

SO<sub>T</sub> คือ Serial Output Temperature ค่าดิจิทัลของอุณหภูมิ

| VDD   | d1 (°C) | d1 (°F) |
|-------|---------|---------|
| 5 V   | -40.1   | -40.2   |
| 4 V   | -39.8   | -39.6   |
| 3.5 V | -39.7   | -39.5   |
| 3 V   | -39.6   | -39.3   |
| 2 V   | -39.4   | -38.9   |

ตารางที่ 2.1 แสดงค่าคงที่ของ d1

| SOT   | d2 (°C) | d2 (°F) |
|-------|---------|---------|
| 14bit | 0.01    | 0.018   |
| 12bit | 0.04    | 0.072   |

ตารางที่ 2.2 แสดงค่าคงที่ของ d2

### 2.8.3 การเปลี่ยนข้อมูล digital มาเป็นข้อมูลของความชื้น

ค่าของข้อมูล Humidity มีลักษณะเป็นสมการพหุคูณ สามารถคำนวณได้จากสมการ

$$RH_{linear} = C1 + C2 \cdot SO_{RH} + C3 \cdot SO_{RH}^2 \quad (2.26)$$

$$RH_{true} = (T_{oc} - 25) \times (t1 + t2 \times SO_{RH}) + RH_{linear} \quad (2.27)$$

โดยที่

$RH_{linear}$  คือ ค่าความชื้นสัมพัทธ์

$SO_{RH}$  คือ Serial Output humidity ค่าดิจิทัลของความชื้น

$C1, C2, C3, t1, t2$  คือ ค่าคงที่

| SORH   | C1      | C2     |
|--------|---------|--------|
| 12 bit | -2.0468 | 0.0367 |
| 8 bit  | -2.0468 | 0.5872 |

ตารางที่ 2.3 แสดงค่าคงที่ C1 และ C2

| SORH   | t1   | t2      |
|--------|------|---------|
| 12 bit | 0.01 | 0.00008 |
| 8 bit  | 0.01 | 0.00128 |

ตารางที่ 2.4 แสดงค่าคงที่ t1 และ t2

## บทที่ 3

### การดำเนินงานทดลอง

เนื่องจากในเบื้องต้นทราบว่าความเร็วของพัดลมที่อิวาโปรเตอร์ มีผลต่ออัตราการไหลของอากาศ และปริมาณความชื้นสัมพัทธ์ของอากาศ [8] รวมทั้งปริมาณความชื้นสัมพัทธ์ของอากาศเป็นหนึ่งในปัจจัยของความสบายเชิงความร้อนของมนุษย์ [3] ดังนั้นโครงการนี้จึงได้ทำการศึกษาเกี่ยวกับการหาแนวทางในการออกแบบโดยการสร้างชุดควบคุมความเร็วพัดลมขึ้นมาเพื่อนำไปประยุกต์ใช้ในการปรับอัตราการไหลของอากาศที่อิวาโปรเตอร์ และเป็นแนวทางในการควบคุมปริมาณความชื้นของอากาศให้คงที่และเป็นไปตามความต้องการของผู้อยู่อาศัยมากที่สุด ซึ่งจะพิจารณาในขอบเขตของสถานะความสบายของมนุษย์เป็นหลัก

#### 3.1 ขั้นตอนและวิธีการดำเนินงาน

ขั้นตอนการดำเนินงานของโครงการจะพิจารณาตั้งแต่เริ่มดำเนินโครงการ การศึกษาค้นคว้า การออกแบบ การสร้างชุดควบคุม การทดลอง ตลอดจน การสรุปและวิเคราะห์ผลการทดลอง ซึ่งได้จัดเรียงลำดับการดำเนินงานดังนี้

- 3.1.1 ศึกษาค้นคว้าเกี่ยวกับแนวทางในการควบคุมความชื้นสัมพัทธ์ของห้องปรับอากาศ
- 3.1.2 ศึกษาค้นคว้าเกี่ยวกับแนวทางในการควบคุมความเร็วพัดลม
- 3.1.3 พิจารณาทางเลือกและเตรียมแผนงานในการดำเนินการ
- 3.1.4 เขียนโปรแกรมควบคุมความเร็วพัดลม
- 3.1.5 จัดเตรียมอุปกรณ์และสร้างชุดควบคุมความเร็วพัดลม
- 3.1.6 ทดสอบการทำงานโปรแกรมและแผงวงจรโดยการสุ่มป้อนค่าความชื้นต่าง ๆ กัน
- 3.1.7 ทำการปรับปรุงทั้งโปรแกรมและแผงวงจร
- 3.1.8 ต่อแผงวงจรควบคุมเข้ากับพัดลมและเครื่องเพิ่มความชื้นสัมพัทธ์
- 3.1.9 นำชุดควบคุมไปทดลองใช้กับห้องปรับอากาศและบันทึกผล
- 3.1.10 บันทึกค่าความชื้นสัมพัทธ์ของห้องปรับอากาศที่ใช้ชุดควบคุมกับไม่ใช้ชุดควบคุม
- 3.1.11 นำผลการทดลองที่ได้มาเปรียบเทียบและหาความสัมพันธ์
- 3.1.12 สรุปและวิเคราะห์ผลการทดลอง

### 3.2 อุปกรณ์และเครื่องมือวัด

อุปกรณ์และเครื่องมือที่ใช้ทำโครงการเป็นสิ่งที่สำคัญมาก เนื่องจากการใช้อุปกรณ์ที่มีความทันสมัยและมีประสิทธิภาพจะส่งผลให้การดำเนินงานทำได้ง่ายขึ้น รวมทั้งช่วยให้ผลการทดลองมีความผิดพลาดลดลง ซึ่งอุปกรณ์ที่ใช้ในการทดลองนี้มีจำนวนมาก แต่ในที่นี้จะกล่าวถึงเฉพาะอุปกรณ์หลักที่จำเป็นต้องใช้ในการดำเนินงาน ซึ่งมีดังนี้

- 3.2.1 ไมโครคอนโทรลเลอร์รุ่น PIC18F452
- 3.2.2 บอร์ดทดลอง
- 3.2.3 จอแสดงผล ชนิด LCD  $16 \times 2$
- 3.2.4 อุปกรณ์เพิ่มความชื้น
- 3.2.5 Regulator ชนิด Monolithic เบอร์ L7805C และ LM317T
- 3.2.6 ทรานซิสเตอร์ (Transistor) เบอร์ TIP122
- 3.2.7 คาปาซิเตอร์ (Capacitor) ขนาด 0.47  $\mu\text{F}$  22  $\mu\text{F}$  22pF
- 3.2.8 ตัวต้านทานไฟฟ้าชนิดค่าคงที่ (Fixed Resistor) ขนาด 10 K $\Omega$  220 $\Omega$  390 $\Omega$
- 3.2.9 ตัวต้านทานชนิดเปลี่ยนแปลงค่าได้ (Variable Resistor)
- 3.2.10 เครื่องปรับอากาศ TOSHIBA เบอร์ 5 ขนาด 1 ตันความเย็น (12000Btu/hr)
- 3.2.11 ฮอตไวร์ แอนนิโมมิเตอร์ (Hot wire anemometer) วัดความเร็วลม
- 3.2.12 แทคโอมิเตอร์ Tachometer เครื่องวัดความเร็วรอบ (RPM)
- 3.2.13 เครื่องมือวัดความชื้นและอุณหภูมิ (SILA)
- 3.2.14 เครื่องโปรแกรมไมโครคอนโทรลเลอร์ รุ่น PICSTART Plus
- 3.2.15 แผ่นปรินตอนกประสงค์
- 3.2.15 Relay 12V

### 3.3 ขั้นตอนการทดลอง

ในการดำเนินการทดลองชุดควบคุมความชื้นจะใช้ความถี่ 20M Hz จะได้ค่า value สูงสุดที่ 1000 และทำการทดสอบโดยการวัดค่าความเร็วรอบกับอัตราไหลของอากาศ ที่ค่า value ต่างๆ และนำชุดควบคุมความชื้นมาทดลองในห้องปรับอากาศโดยเชื่อมต่อให้ถูกต้องดังที่

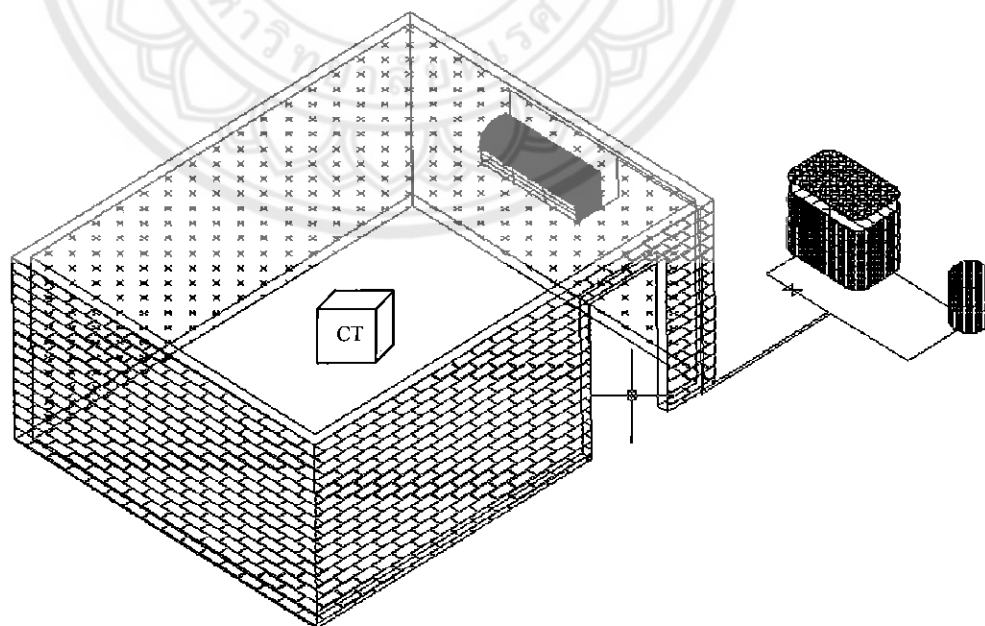
#### การทดลองที่ 3.3.1 ทดสอบชุดควบคุมความชื้น

3.3.1.1 เชื่อมค่า value ของ Duty Cycle ที่ 1000 คือ Duty Cycle = 100% ทำการวัดความเร็วรอบและอัตราไหลของอากาศ (อ่านเพิ่มเติมใน หัวข้อ 2.7.1)

3.3.1.2 เชื่อมค่า value ของ Duty Cycle ที่ 900 คือ Duty cycle = 90% ทำการวัดความเร็วรอบและอัตราไหลของอากาศ

3.3.1.3 เชื่อมค่า value ของ Duty cycle ลดลงมาเรื่อยๆ ครั้งละ 100 จากนั้นจึงทำการวัดความเร็วรอบและอัตราไหลของอากาศ

ชุดทดลองจะประกอบด้วยห้องทดลองซึ่งใช้เครื่องปรับอากาศ TOSHIBA เบอร์ 5 ขนาด 1 ตันความเย็น (1200 Btu/hr) และชุดควบคุมความชื้น



รูปที่ 3.1 ห้องทดลองและตำแหน่งเครื่องมือวัด

**การทดลองที่ 3.3.2 ทดสอบพัฒมล่าของอัตราการไหลของอากาศ**

3.3.2.1 วัดค่าอัตราไหลของอากาศที่ความเร็วรอบสูงสุด

3.3.2.2 เช็ทความเร็วรอบ 100%-0% ทำการวัดอัตราไหลของอากาศ

**การทดลองที่ 3.3.3 ไม่ใช้อุปกรณ์ที่ช่วยควบคุมความชื้น**

3.3.3.1 เปิดเครื่องปรับอากาศเซ็ทที่อุณหภูมิ 24-26°C

3.3.3.2 บันทึกค่าอุณหภูมิและความชื้นภายในห้องทุกๆ 1 นาที

**การทดลองที่ 3.3.4 ใช้อุปกรณ์ที่ช่วยควบคุมความชื้น**

3.3.3.1 เปิดเครื่องปรับอากาศเซ็ทที่อุณหภูมิ 24-26°C

3.3.3.2 บันทึกค่าอุณหภูมิและความชื้นภายในห้อง

3.3.3.3 บันทึกอัตราไหลของอากาศและความเร็วของพัดลม

จากนั้นนำผลการทดลองที่ได้ทำการวิเคราะห์หาความสัมพันธ์ เปอร์เซนต์ความคลาดเคลื่อน และสรุปผลการทดลอง เพื่อเป็นแนวทางในการควบคุมความชื้นให้อยู่ภายใต้ขอบเขตของสถานะความสบายเชิงความร้อนของมนุษย์พัฒนาต่อไป

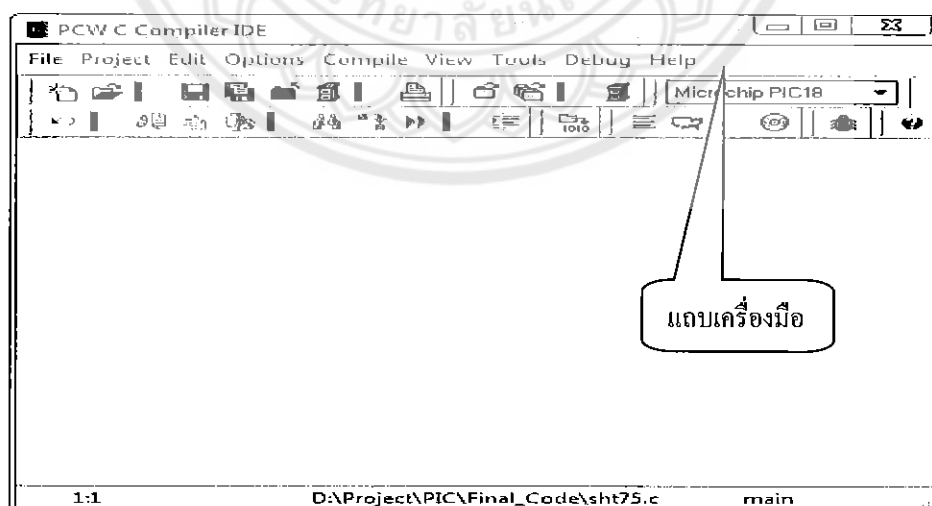
### 3.4 การสร้างชุดควบคุมความเร็วพัฒนา

ในการสร้างชุดควบคุมได้แบ่งการทำงานออกเป็น 3 ส่วน ซึ่งประกอบด้วยการใช้โปรแกรม CCS C compiler 3.202, MPLAB และ Proteus 7 Professionals รวมทั้งมีการแสดง Flow Chart การทำงานของชุดควบคุมความเร็วพัฒนา ซึ่งมีขั้นตอนการทำงานดังนี้

#### 3.4.1 การใช้โปรแกรม Compiler CCS Version 3.202 ในการเขียน code คำสั่งพัฒนาทำงาน

CCS C compiler เป็นซอฟต์แวร์สำหรับแปลโปรแกรมภาษา C ของงานไมโครคอนโทรลเลอร์ PIC โดยในโครงการนี้ได้เลือกใช้ CCS C compiler 3.202 รุ่น PCW ซึ่งใช้งานได้กับไมโครคอนโทรลเลอร์ PIC18F452 และภาษาที่ใช้เขียนกับ CCS C compiler จะใช้ภาษา C โดยมีการใช้งานของโปรแกรมสำหรับเขียน code ดังนี้

- หลังจากติดตั้งโปรแกรมแล้วก็ทำการเปิด PCW ขึ้นมาใช้งาน โดยไปที่ start → All Program → PIC- C → PIC C compiler หรือคลิกที่ไอคอน PIC C compiler ดังรูปต่อไปนี
- เมื่อเปิด PIC C compiler จะปรากฏหน้าต่างโปรแกรมดังรูปต่อไปนี

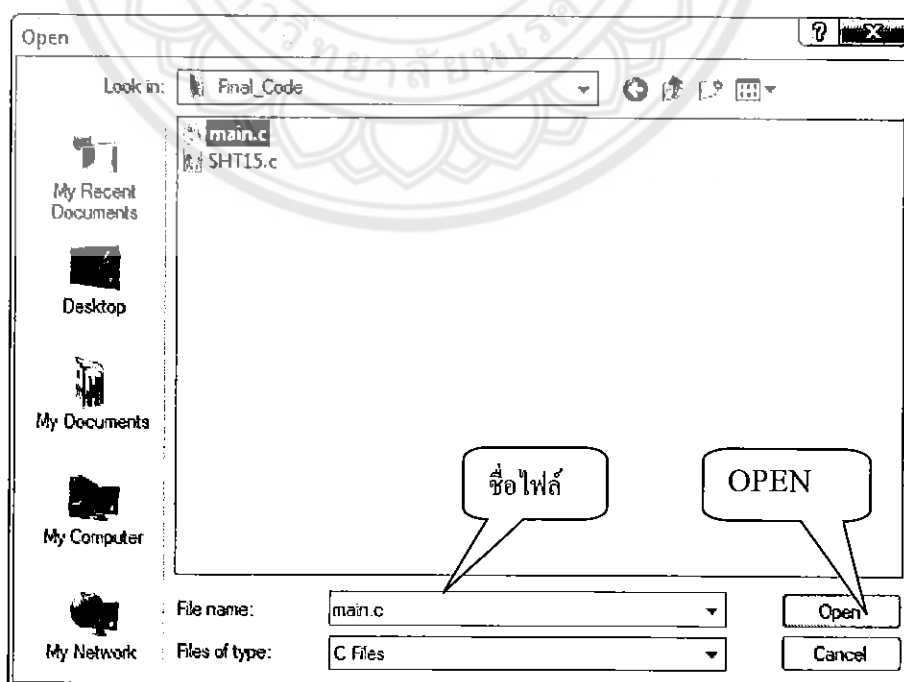


รูปที่ 3.4.1 แสดงหน้าต่างเริ่มต้นโปรแกรม

- ไปที่เมนู Project → New → Manual Create คลิกเมาส์ที่ปุ่ม OPEN (ตรงที่มีจุด 3 จุด) จะได้หน้าต่าง Open กำหนดชื่องานที่ต้องการในช่อง File name พร้อมทั้งเลือกที่สำหรับเก็บไฟล์งาน (โครงการนี้เก็บไฟล์ไว้ที่ D:\Project\PIC\Final\_Code และตั้งชื่อไฟล์ว่า main.c) จากนั้นก็กดปุ่ม OPEN

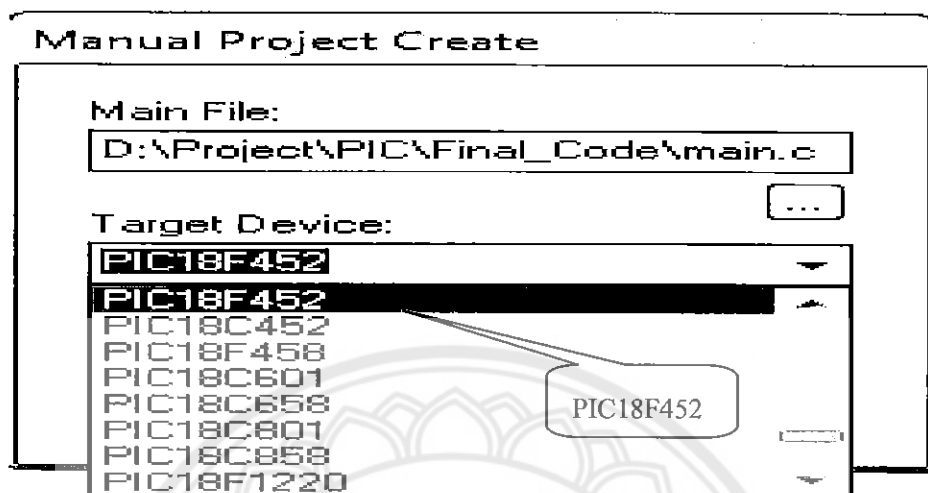


รูปที่ 3.4.2 แสดง Manual Project Create



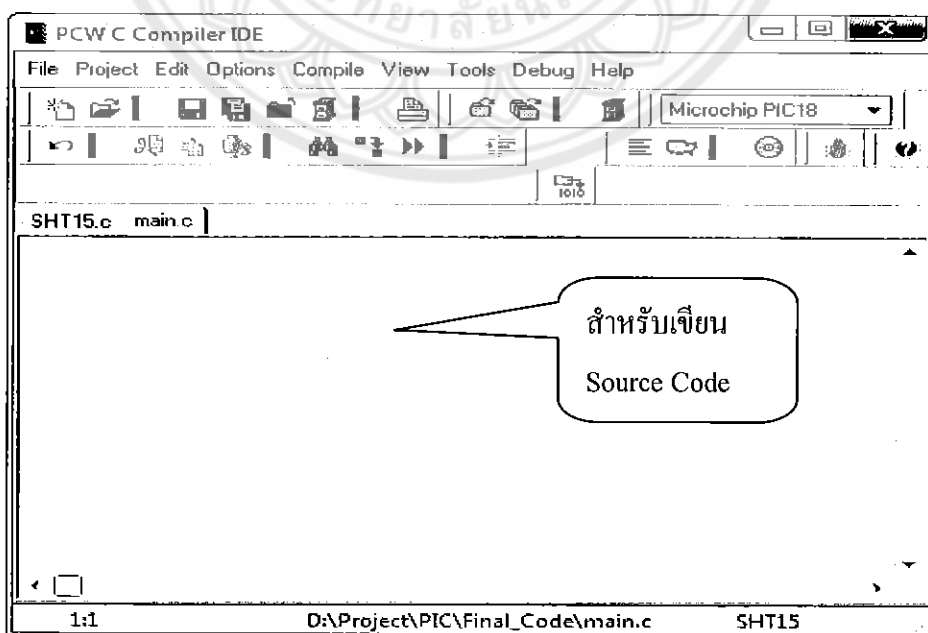
รูปที่ 3.4.3 แสดงการตั้งชื่อไฟล์

- จากรูปที่ 3.3 เลือก Target Device ไปที่ PIC18F452



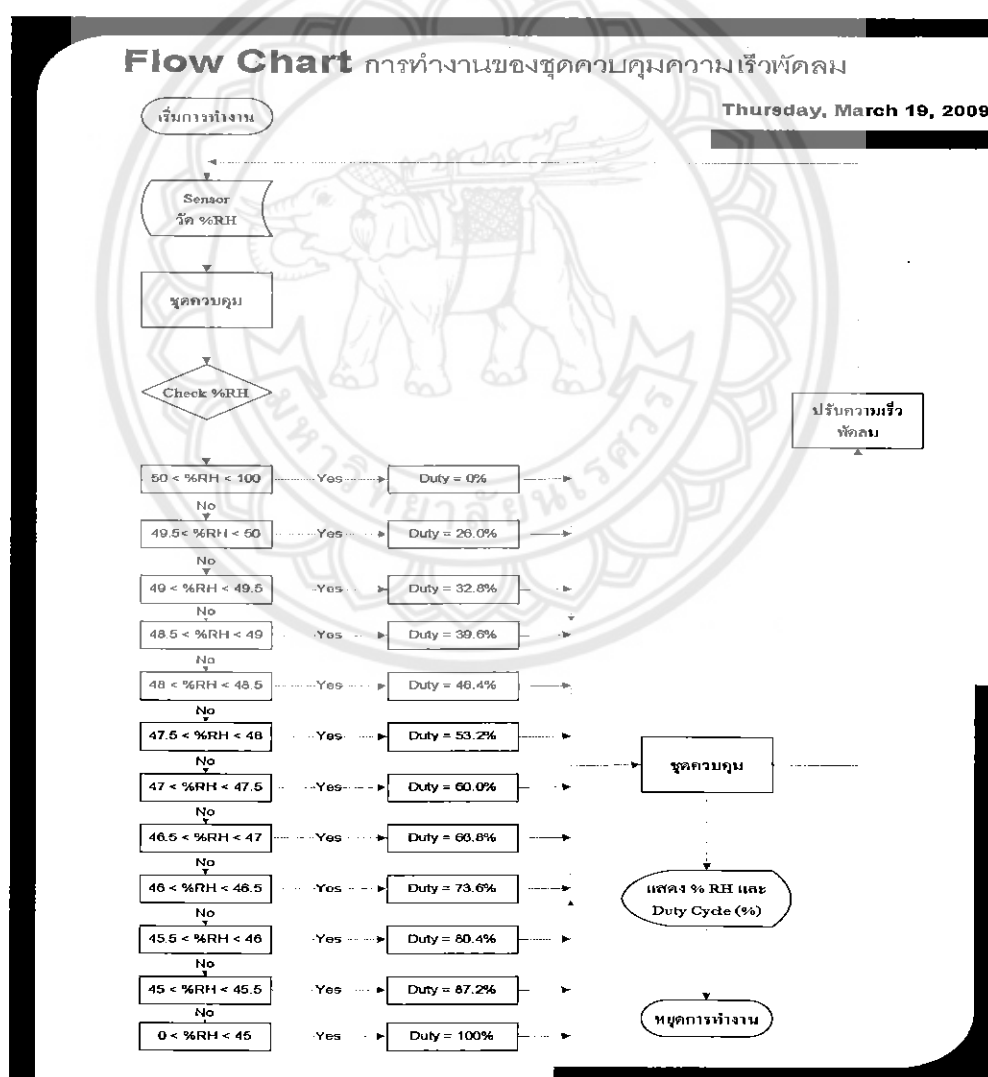
รูปที่ 3.4.4 แสดงการเลือก Target Device

- คลิกปุ่ม Create จะปรากฏหน้าต่างสำหรับเขียน Source Code ดังรูปด้านล่างนี้ (จากรูปได้สร้าง SHT15.c เพิ่มขึ้นมาเพื่อเขียน Driver ของ SHT15)



รูปที่ 3.4.5 แสดงหน้าต่างสำหรับเขียน Source Code

Flow Chart ของการทำงานของชุดควบคุมความเร็วพัดลมโดยเซ็นเซอร์อุณหภูมิที่ตั้ง โดยใช้ Sensor(SHT15) เป็นตัววัดความชื้น และชุดควบคุมความชื้นจะ Check ความชื้นที่วัดได้ เพื่อจะไปถึงงานให้พัดลมหมุนที่ค่า Duty Cycle ต่างๆ โดยที่ Duty Cycle= 0% คือพัดลมจะไม่ทำงาน Duty Cycle 100% คือพัดลมจะทำงานที่ความเร็วรอบสูงสุด ณ ที่นี้ เนื่องจากสภาวะความสบาย จึงควบคุมความชื้นให้ได้ 45-50 %RH และจากการทดลองในห้องปรับอากาศจะมีความชื้นที่ต่ำเกินไปจึงกำหนดให้ ถ้าความชื้นที่วัดได้อยู่ในช่วง 100-50 %RH พัดลมจะหยุดหมุน Duty Cycle= 0% ถ้าความชื้นที่วัดได้อยู่ในช่วง 45-0 %RH พัดลมจะทำงานที่ Duty Cycle = 100% จากนั้นใช้จอ LCD16x2 ในการแสดงผลข้อมูลของความชื้น และ ค่าของ Duty Cycle



รูปที่ 3.4.6 แสดง Flow Chart แสดงการทำงานของชุดควบคุมความเร็วพัดลม

## 3.4.2 ใช้โปรแกรม MPLAB IDE v7.01 ในการนำ Code ที่เขียนลงใน PIC18F452

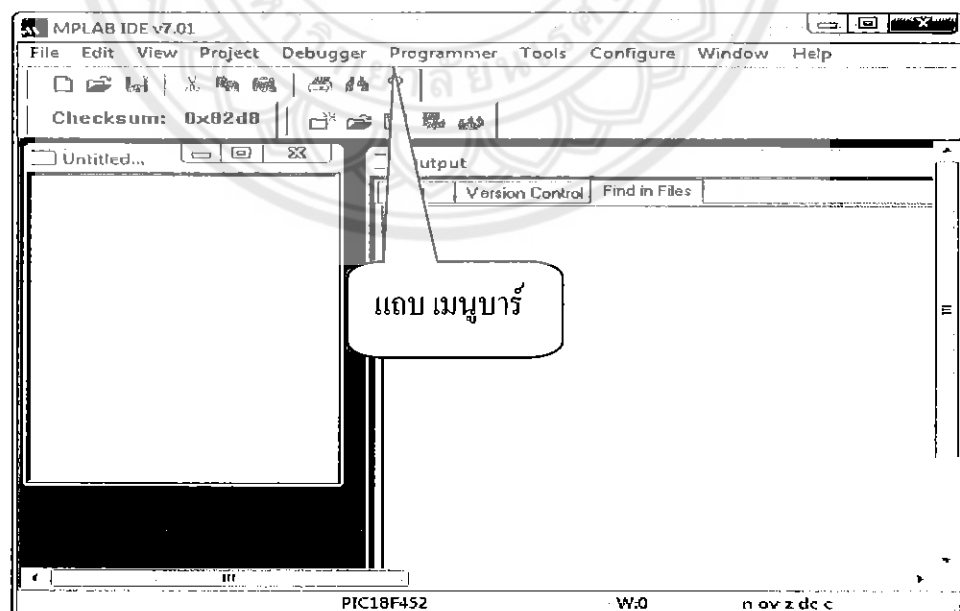
๑๖๒๘-ก  
๒๕๕๑.

MPLAB เป็นซอฟต์แวร์ที่สร้างขึ้นมาเพื่อใช้สำหรับตีพิมพ์ Source code ซึ่งรองรับได้หลายภาษารวมทั้งสามารถทำการนำโปรแกรมลงในไมโครคอนโทรลเลอร์ PIC ได้ โดย MPLAB นั้นได้สนับสนุนการใช้งาน CCS Compiler และลิงก์โค้ดโปรแกรมผ่าน Command Line ซึ่งในโครงการนี้จะใช้ MPLAB V.7 สำหรับโปรแกรม Source Code ลงในไมโครคอนโทรลเลอร์ PIC โดยแสดงการใช้งานได้ดังนี้

- หลังจากติดตั้งโปรแกรมแล้วก็ทำการเปิด MPLAB ขึ้นมาใช้งาน ไปที่ ที่ start → All Program → Microchip → MPLAB IDE v7.01 → คลิกที่ไอคอน MPLAB IDE

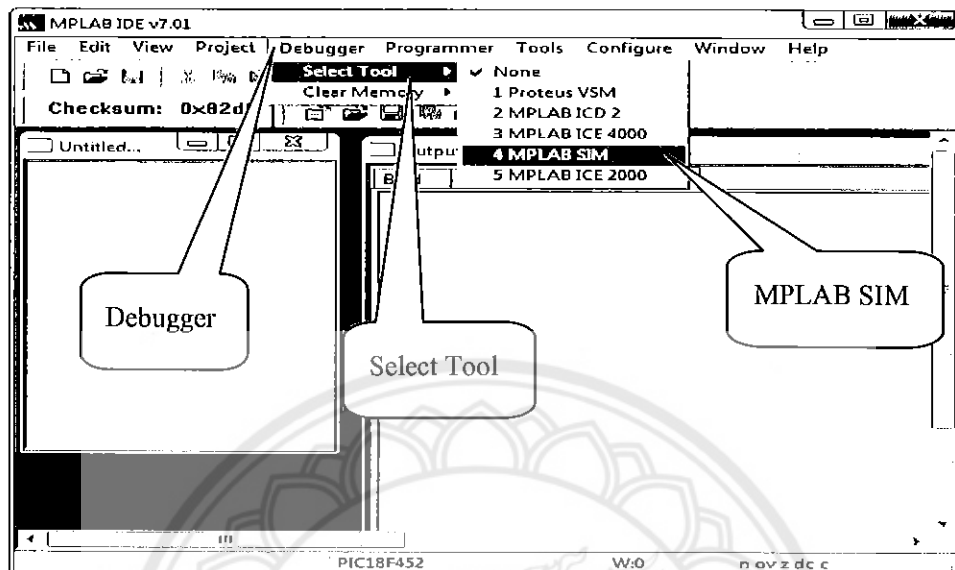


รูปที่ 3.4.7 แสดงการเปิดโปรแกรม MPLAB IDE V7.01



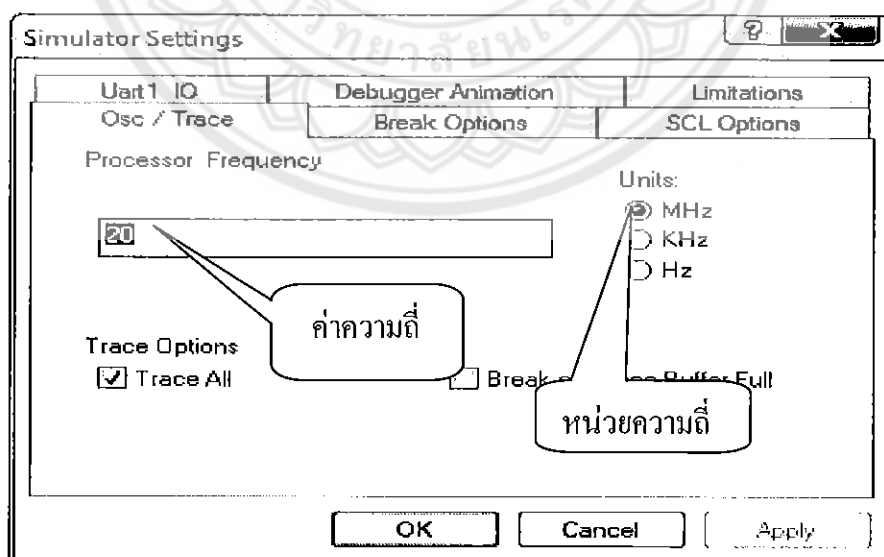
รูปที่ 3.4.8 แสดงหน้าต่างเมื่อเปิดโปรแกรม MPLAB IDE V7.01

- ไปที่เมนูบาร์เลือก Debugger → Select Tool → เลือก MPLAB SIM



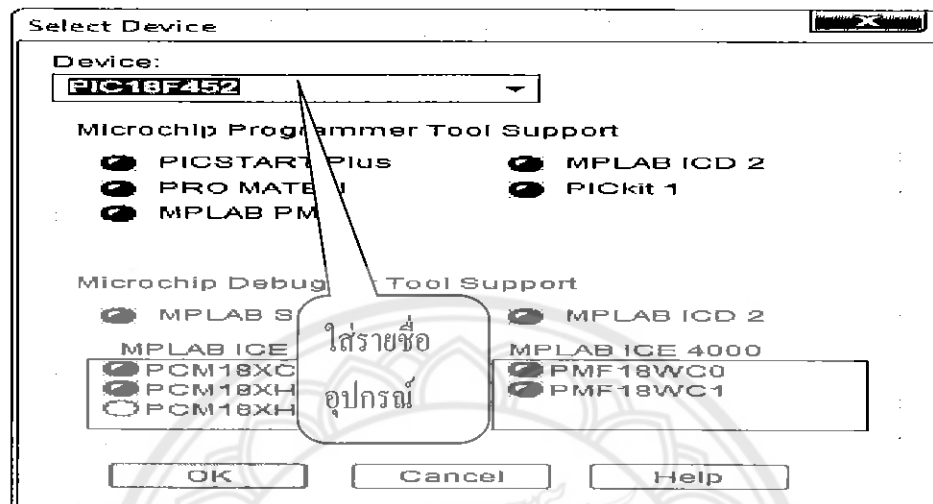
รูปที่ 3.4.9 แสดงหน้าต่างเมื่อเปิดโปรแกรม MPLAB IDE V7.01

- ไปที่เมนูบาร์เลือก Debugger → Settings → ในช่อง Processor / frequency ให้ใส่ความถี่ของ CRYSTAL ที่เลือกใช้งานซึ่งในโครงการนี้ใช้ที่ความถี่ 20 MHz หลังจากนั้นกด OK



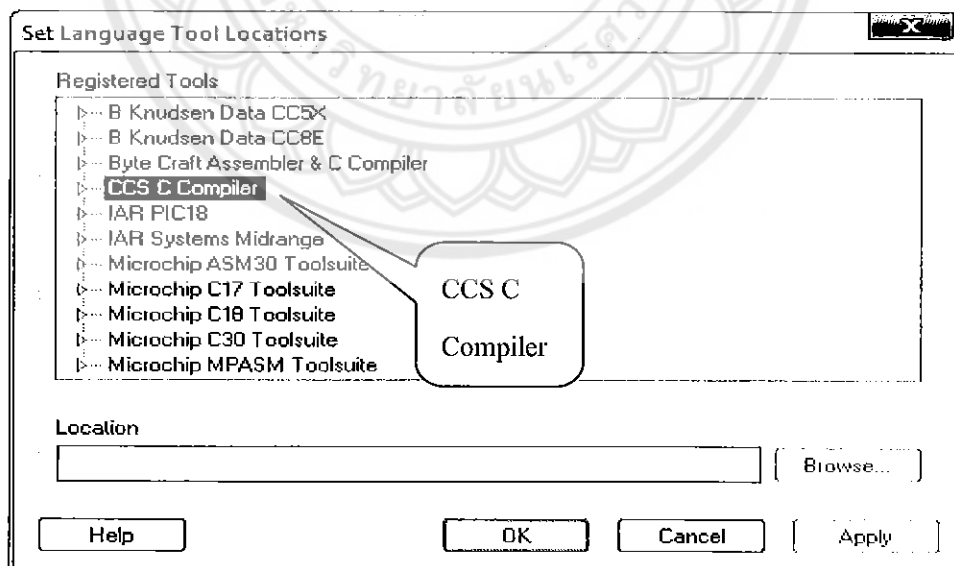
รูปที่ 3.4.10 แสดงการกำหนดค่าความถี่ของ CRYSTAL

- เลื่อนเมาส์ไปที่ Configure → Select Device → ในช่อง Device เลือก PIC18F452 แล้วกด OK



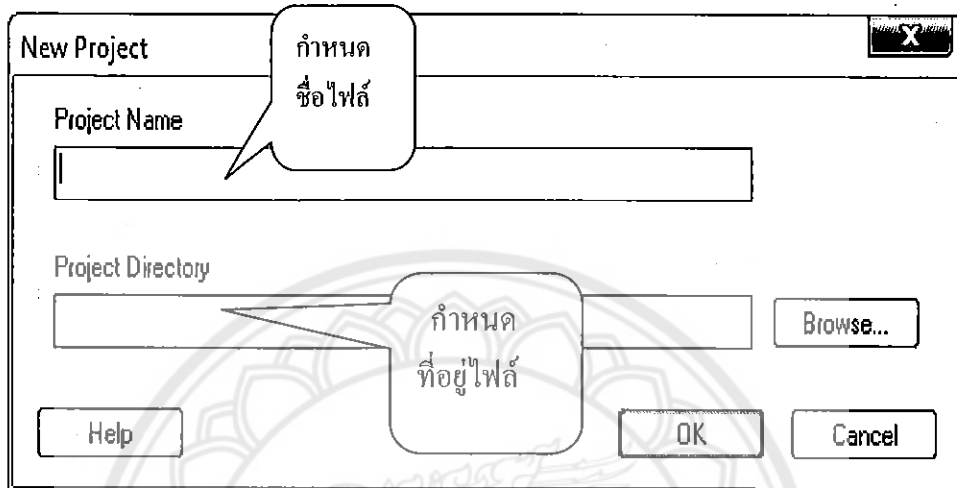
รูปที่ 3.4.11 แสดงการเลือกอุปกรณ์ PIC18F452

- ไปที่ Project → Set Language Tool Locations → คลิกเลือกที่ CCS C Compiler กด OK



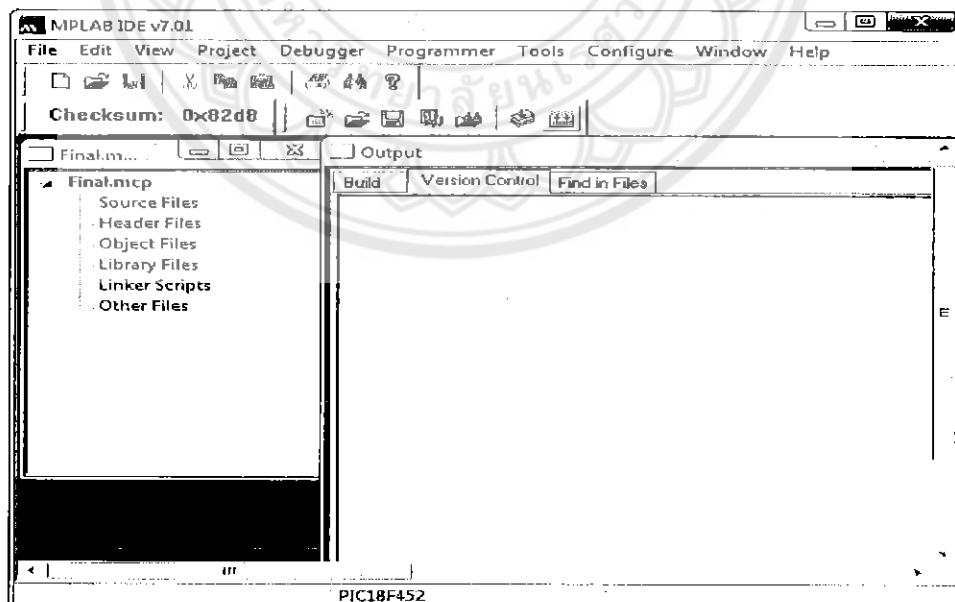
รูปที่ 3.4.12 แสดงการ Set Language Tool Location

- ไปที่ Project → New จะปรากฏดังรูปด้านล่างนี้
  - Project name ให้ใส่ชื่อที่ต้องการบันทึก
  - Project Directory เลือกที่อยู่ไฟล์ที่ต้องการ จากนั้นกด OK



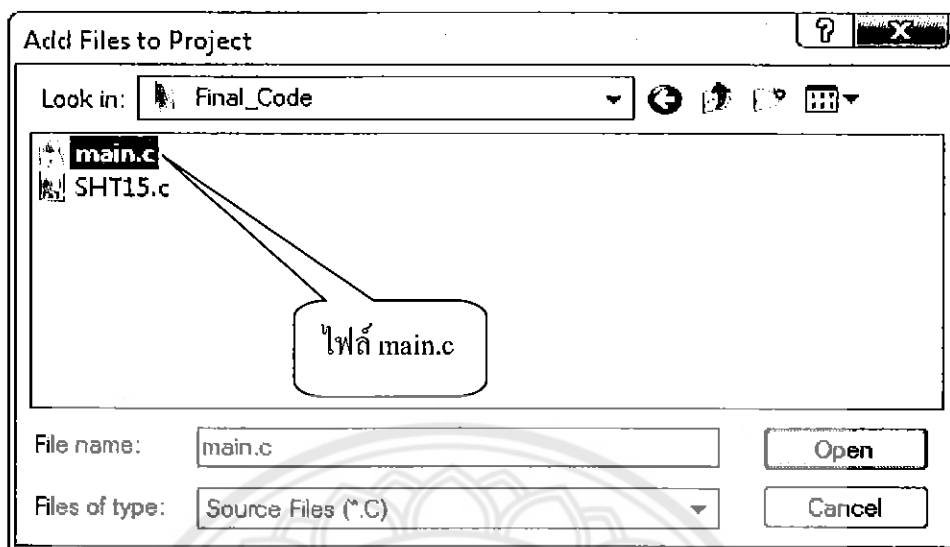
รูปที่ 3.4.13 แสดงการกำหนดชื่อและตำแหน่งของไฟล์

- จากนั้นกด OK จะปรากฏหน้าต่างสำหรับการทำงานดังรูป



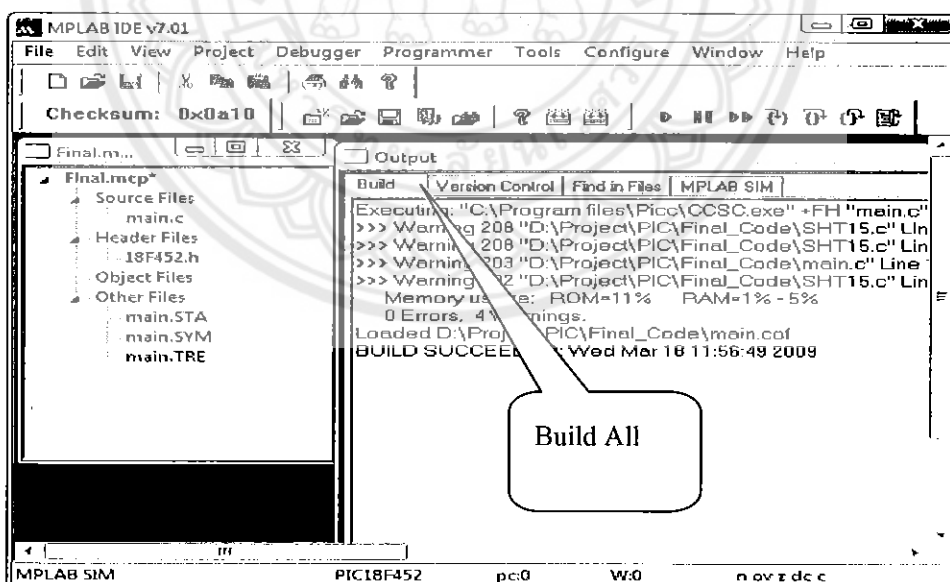
รูปที่ 3.4.14 แสดงหน้าต่างทำงานหลังการกำหนดชื่อและที่อยู่ไฟล์

- ไปที่ Project → Add file To Project ให้เลือกไฟล์ main.c ที่สร้างจาก CCS C แล้วกด OK



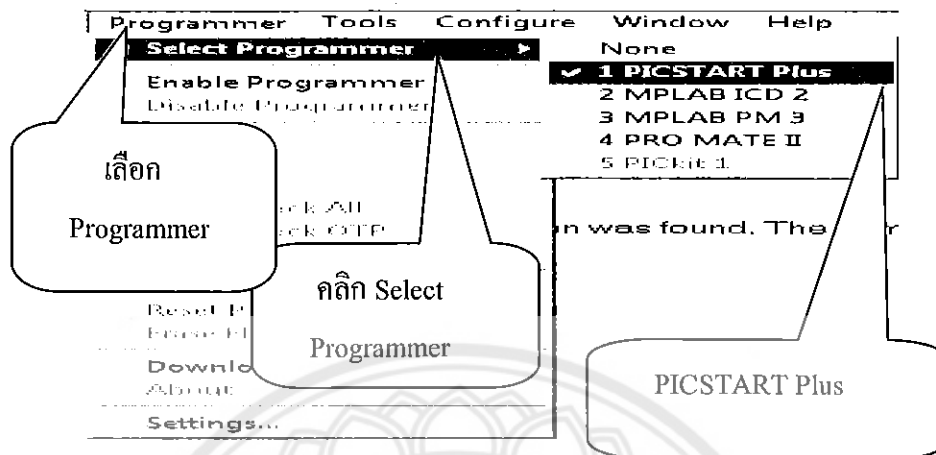
รูปที่ 3.4.15 แสดงการเปิดไฟล์ขึ้นมาใช้งาน

- ไปที่ Project → Build All จะปรากฏหน้าต่างดังรูป



รูปที่ 3.4.16 แสดงการเลือก Build All

- ไปที่ Programmer → Select Programmer แล้วเลือก PICSTART Plus



รูปที่ 3.4.17 แสดงการ Select Programmer

- ไปที่ Programmer แล้วทำตามลำดับดังนี้
  - กด Enable Programmer เพื่อเชื่อมต่ออุปกรณ์
  - กด reset เพื่อ reset ข้อมูลใน PIC18F452
  - รอจนกว่าจะ reset เสร็จจากนั้นจึงกด Erase Flash Device 1 ครั้งแล้วตามด้วย Program รอจนกว่าจะมีคำว่า Successfully แสดงขึ้นมา แล้วจึงกด Disable Program
  - นำไมโครคอนโทรลเลอร์ไปใช้งานกับแผงวงจร

หลังจากนั้นก็นำไมโครคอนโทรลเลอร์ PIC ไปทดลองโดยการต่อวงจรตามที่จำลองใน Proteus 7 Professionals ลงในบอร์ดทดลอง เพื่อทดสอบการทำงาน



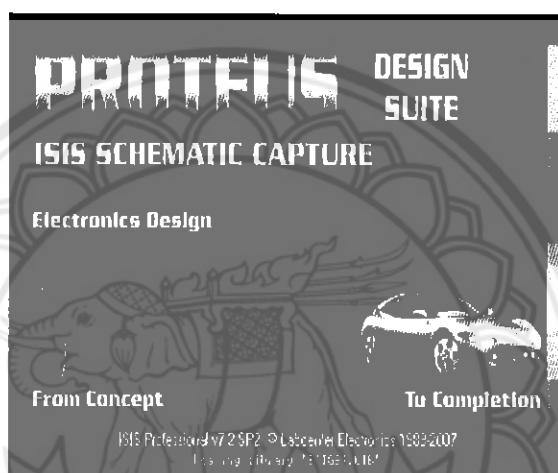
รูปที่ 3.4.18 แสดงการต่อวงจรควบคุมความเร็วพัดลม

เมื่อทดสอบกับบอร์ดทดลองจนแน่ใจแล้วว่าสามารถทำงานได้จริง จึงลงมือสร้างแผงวงจรขึ้นมา ซึ่งวิธีการสร้างจะกล่าวในส่วนถัดไป

### 3.4.3 ใช้โปรแกรม Proteus 7 Professional เป็นโปรแกรม Simulate

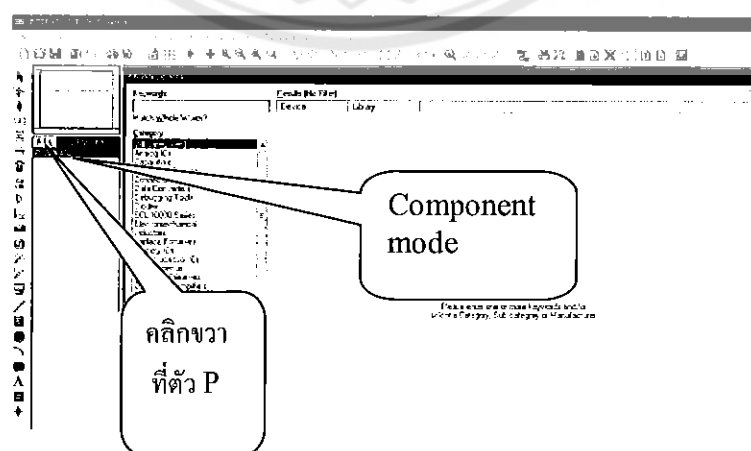
ในส่วนนี้เป็นการออกแบบชุดควบคุมความเร็วพัดลมโดยใช้โปรแกรม Proteus 7.0 Professional ในการออกแบบวงจรควบคุม ซึ่งโปรแกรมมีอุปกรณ์อิเล็กทรอนิกส์ ที่ใช้ในการต่อวงจร ทำให้สะดวกในการต่อวงจร, กำหนดตำแหน่ง, ชนิดของอุปกรณ์และสามารถตรวจสอบการทำงานวงจรที่สร้างได้ ซึ่งในส่วนของการขั้นตอนและวิธีการสร้างชุดควบคุมแสดงมีดังนี้

- เปิดโปรแกรม Proteus 7.0 Professional



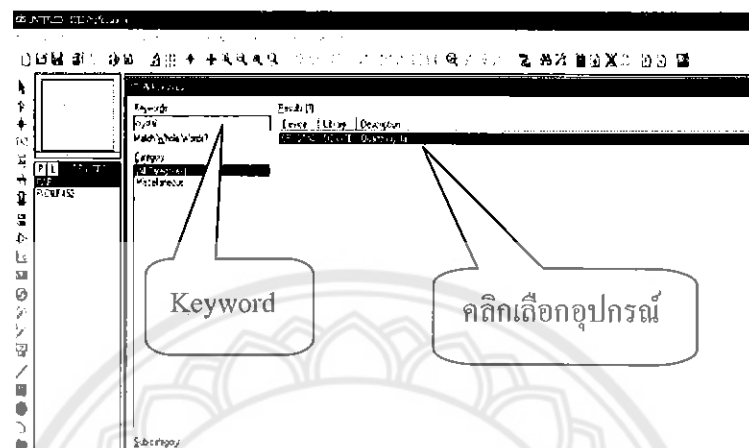
รูปที่ 3.4.19 แสดงหน้าเริ่มต้นของ โปรแกรม Proteus 7.0 Profession

- เลือก Component Mode ที่เมนูบาร์ด้านซ้ายมือ หลังจากนั้นคลิกขวาที่ตัว P จะปรากฏหน้าต่างดังรูปด้านล่าง



รูปที่ 3.4.20 แสดงการเลือกอุปกรณ์ใน โปรแกรม Proteus 7.0 Profession

- พิมพ์ชื่อและรุ่นของอุปกรณ์ที่ต้องการใส่ช่อง Keyword เพื่อทำการค้นหา หลังจากนั้นจะปรากฏรายชื่ออุปกรณ์ขึ้นที่หน้าต่างด้านขวามือ ให้ทำการคลิกเลือกอุปกรณ์

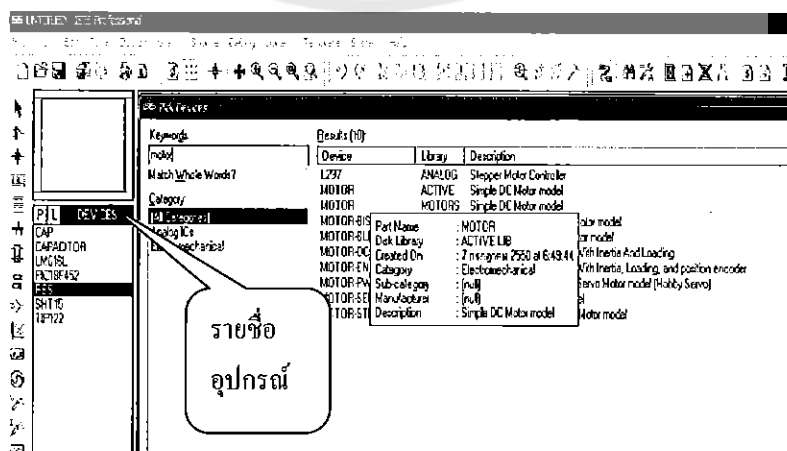


รูปที่ 3.4.21 แสดงการใส่ Key word และเลือกอุปกรณ์ในโปรแกรม Proteus 7.0 Profession

- พิมพ์รายชื่อที่ใช้สำหรับต่อวงจรอุปกรณ์ใส่ช่อง Keyword ตามรายการดังต่อไปนี้

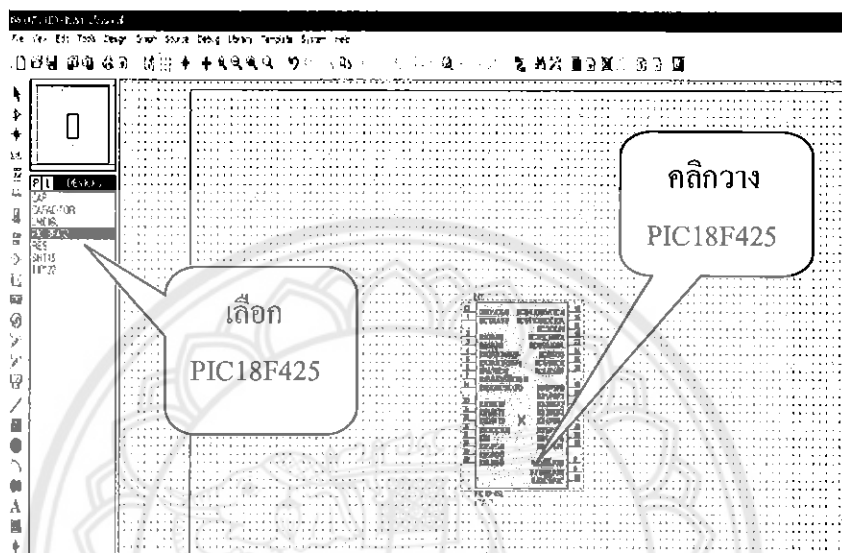
|       |           |           |
|-------|-----------|-----------|
| CAP   | Capacitor | LM016L    |
| RES   | SHT15     | PIC18F452 |
| Motor | TIP122    | CRYSTAL   |

เมื่อเลือกอุปกรณ์ครบตามที่กำหนด ให้ปิดหน้าต่าง Pick Device เพื่อกลับสู่หน้าต่างหลักของโปรแกรม

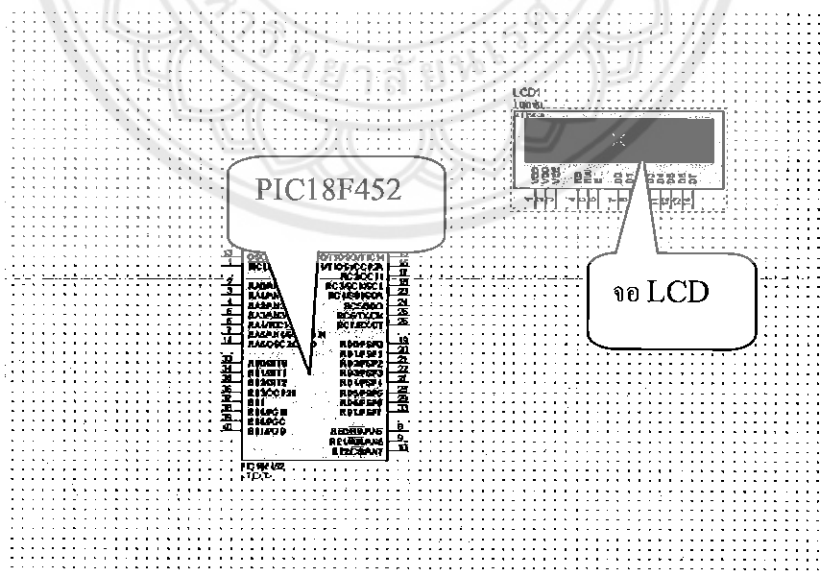


รูปที่ 3.4.22 แสดงรายชื่ออุปกรณ์ที่ใช้ต่อวงจร

- คลิกเลือก PIC18F425 ในช่อง Device หลังจากนั้นคลิกขวาหนึ่งครั้ง จะปรากฏรูปร่างของอุปกรณ์ หลังจากนั้นคลิกขวาอีกครั้งเพื่อวางอุปกรณ์ในตำแหน่งที่ต้องการ

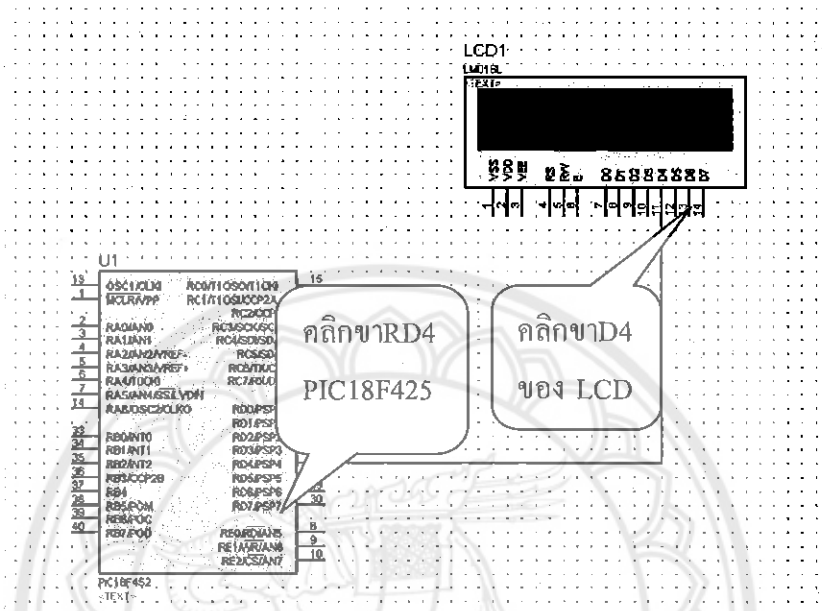


รูปที่ 3.4.23 แสดงการวางตำแหน่งของ PIC18F425 ในวงจร

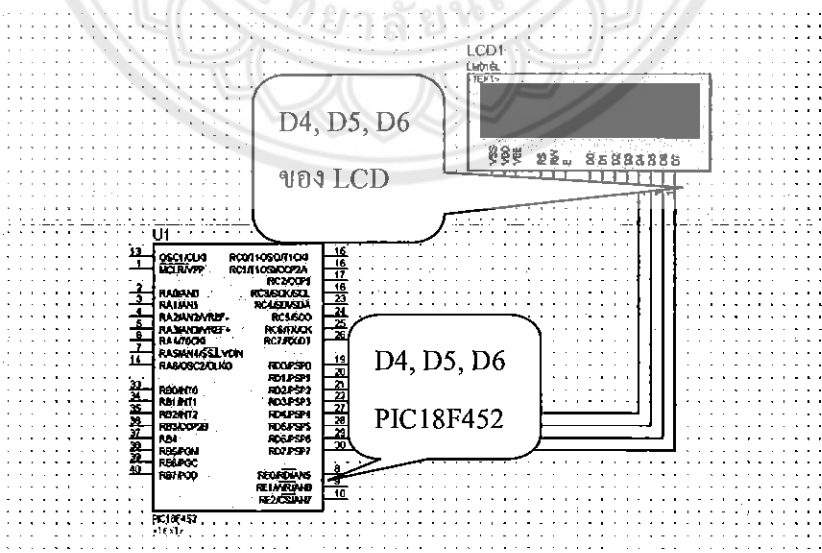


รูปที่ 3.4.24 แสดงการวางจอแสดงผลชนิด LCD ของวงจร

- ต่อวงจรระหว่าง จอแสดงผลชนิด LCD กับ PIC18F425 โดยคลิกขาที่ขา D4 ของ LCD หลังจากนั้นเลื่อนเมาส์ไปคลิกที่ขา RD4 ของ PIC18F425 ซึ่งจะได้ดังรูปด้านล่างนี้

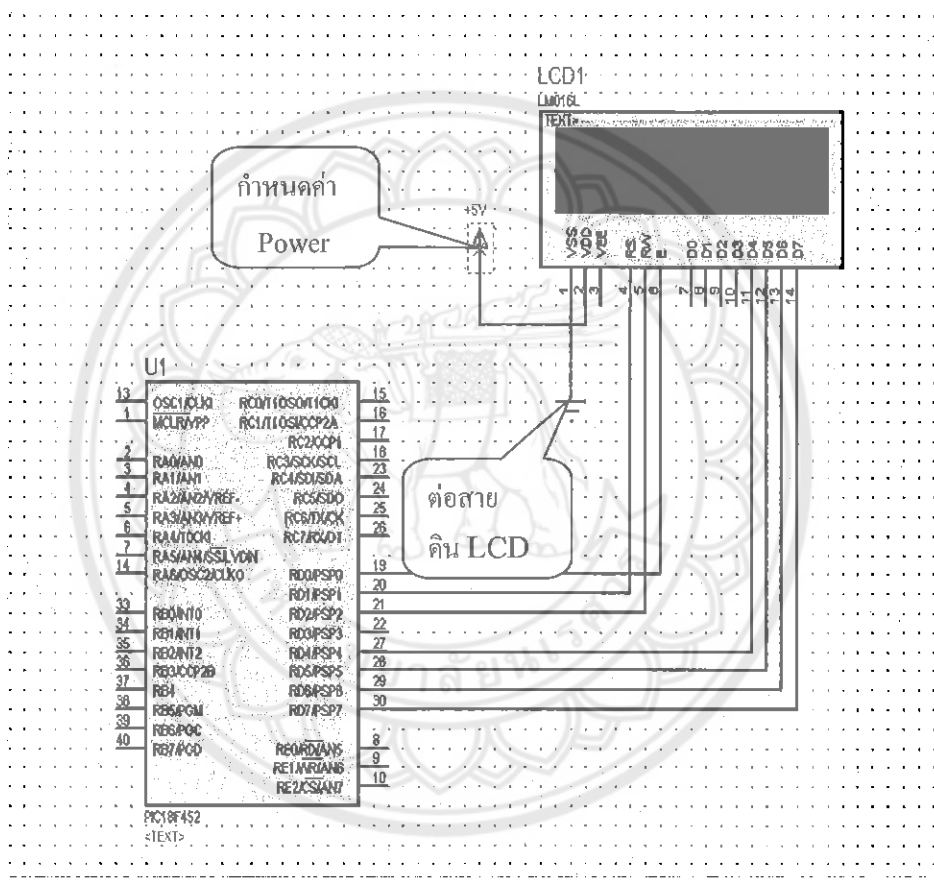


รูปที่ 3.4.25 แสดงการต่อวงจรของ LCD กับ PIC18F425

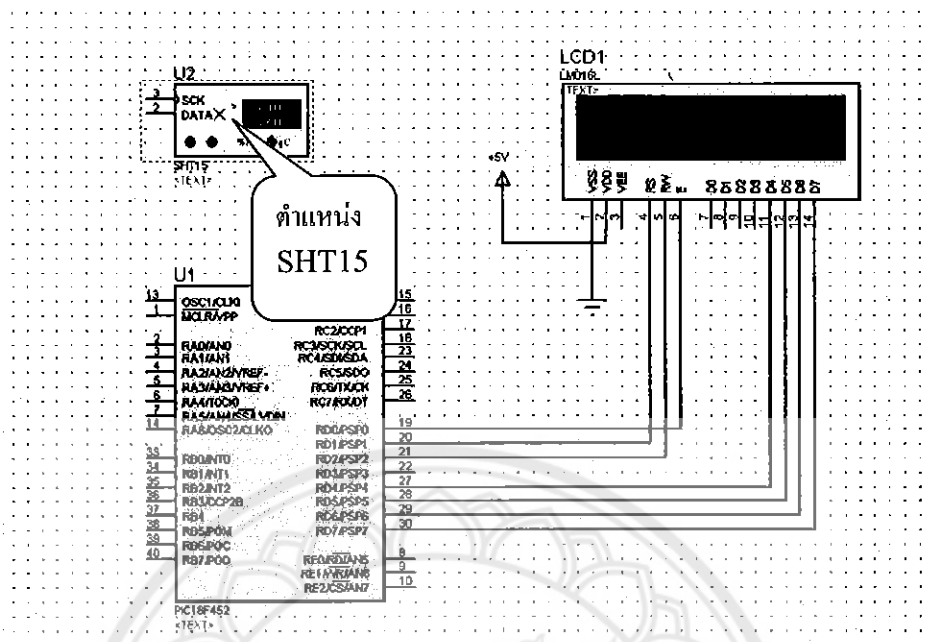


รูปที่ 3.4.26 แสดงการต่อวงจรของ LCD กับ PIC18F425

- ต่อขา RS กับ RD1, RW กับ RD2 และ E กับ RD0 ของ LCD กับ PIC18F452
- คลิกเลือก Terminals Mode หลังจากนั้นเลือก Ground และ Power
- ต่อสายดินและ Power ให้กับจอ LCD หลังจากนั้นดับเบิลคลิกที่หัวลูกศรเพื่อกำหนดค่า Power ให้เป็น +5V (ใส่เครื่องหมาย)

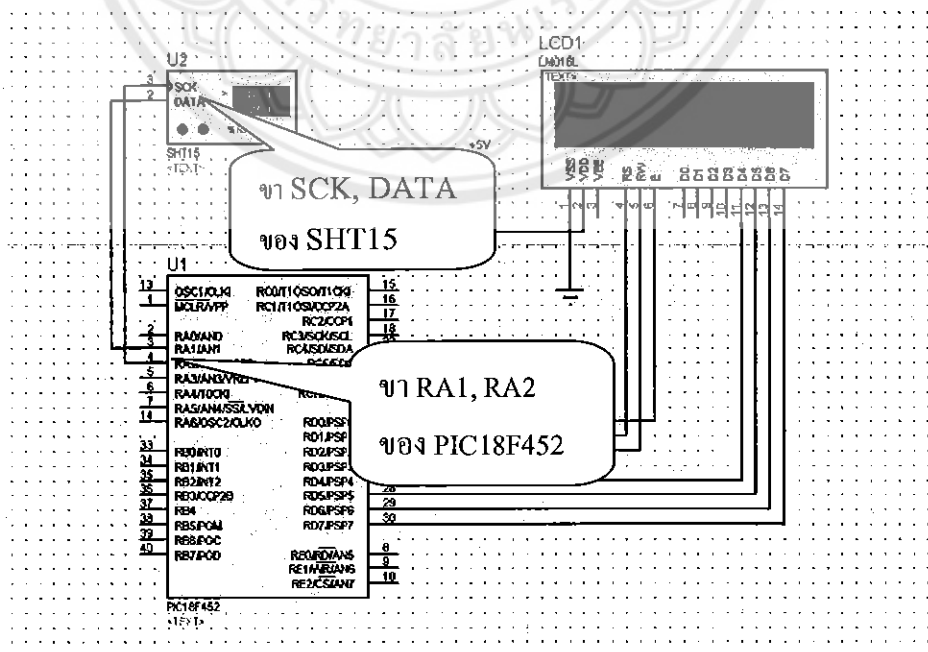


รูปที่ 3.4.27 แสดงการต่อสายดินและ Power ให้กับจอแสดงผล LCD



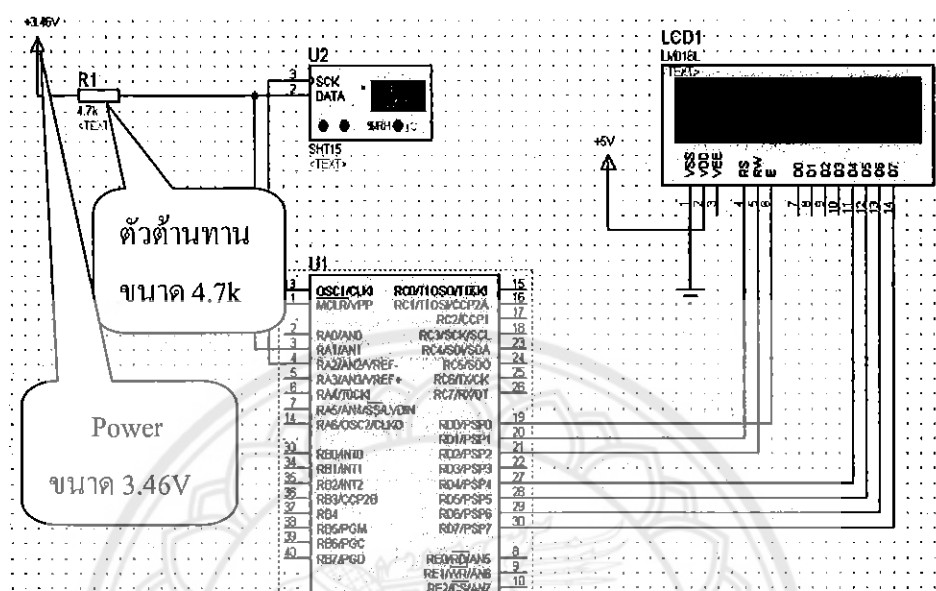
รูปที่ 3.4.28 แสดงตำแหน่งการวางของ SHT15 ของวงจร

- ต่อสาย SCK กับ ขา RA2 และต่อขาDATA กับ ขาRA1ของ SHT15 กับ PIC18F452 ตามลำดับ



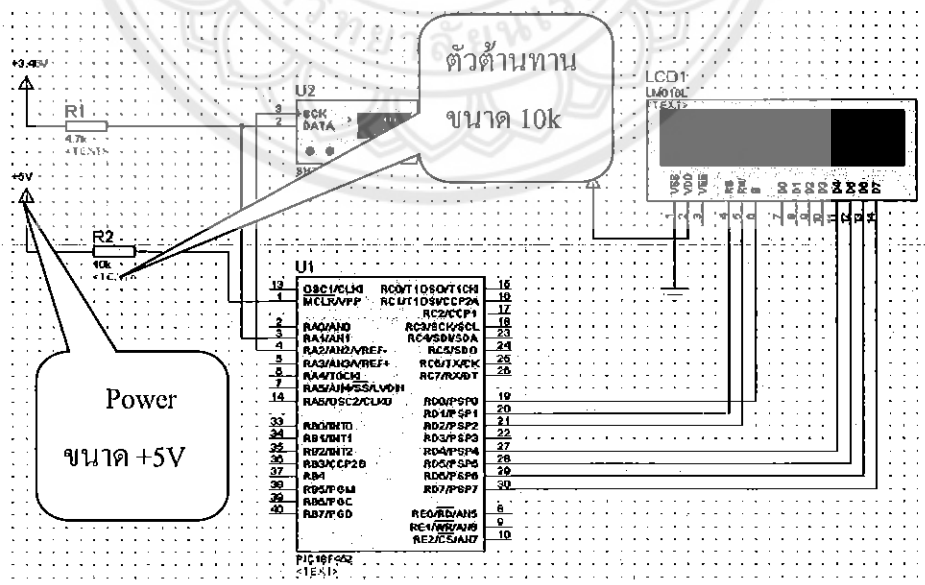
รูปที่ 3.4.29 แสดงการต่อวงจรของ SHT15 กับ PIC18F452 ของวงจร

- ต่อตัวต้านทานขนาด 4.7k และ Power ขนาด 3.46V เข้ากับสาย DATA ของ SHT15 ดังรูป



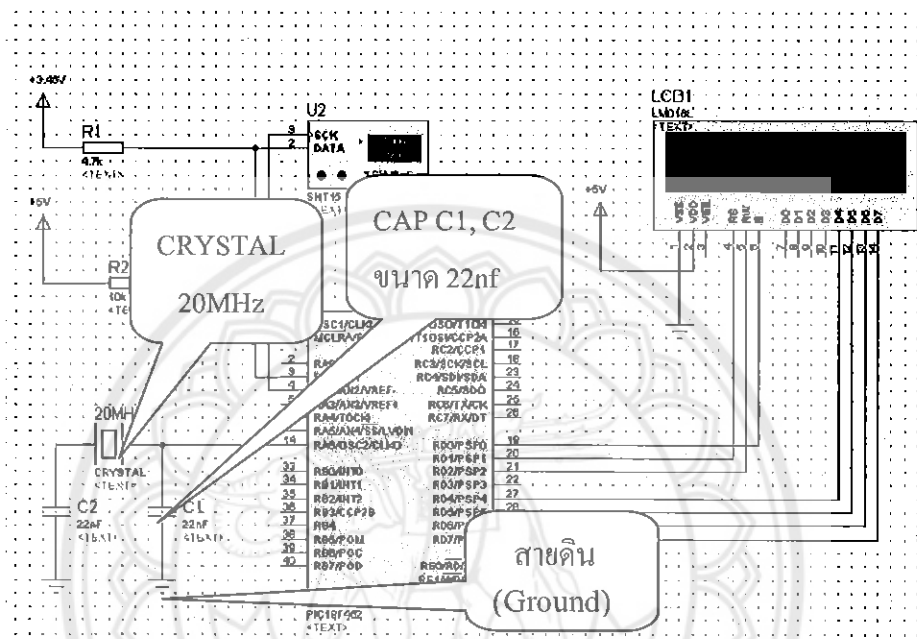
รูปที่ 3.4.30 แสดงการต่อตัวต้านทานและ Power ให้กับ SHT15

- ต่อตัวต้านทานขนาด 10k และ Power ขนาด +5V เข้ากับขาที่1ของ PIC18F452

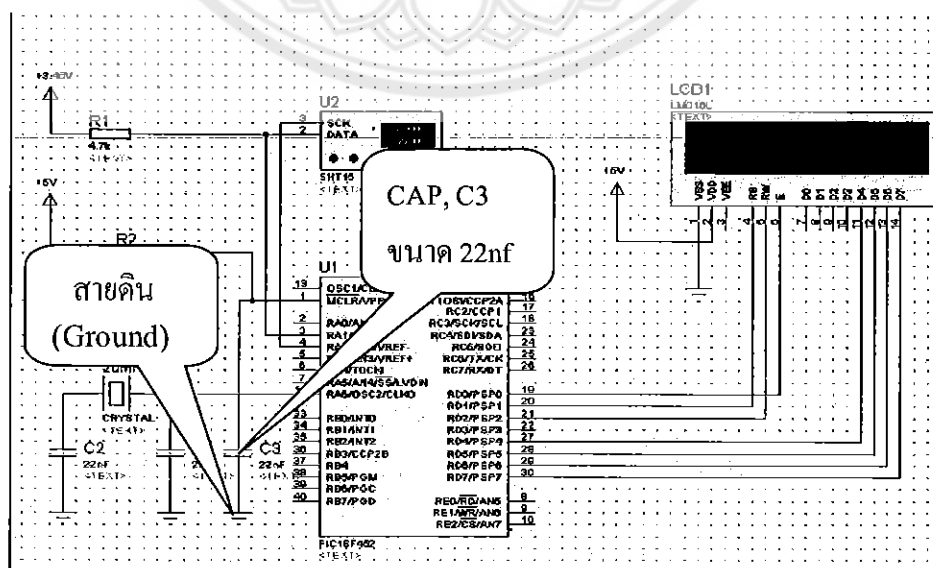


รูปที่ 3.4.31 แสดงการต่อตัวต้านทานและ Power ให้กับ PIC18F452

- จัดตำแหน่งและต่อวงจร CAP ขนาด 22nf จำนวน 2 ตัว พร้อมทั้งต่อสายดินและ ต่อวงจรของCRYSTAL ขนาด 20MHz เข้ากับขา RA6 ของ PIC18F452 รวมทั้งต่อ CAPและสายดินเข้ากับขาที่ 1 ของ PIC18F452 ดังรูป 2.13และ 2.14 ด้านล่างนี้

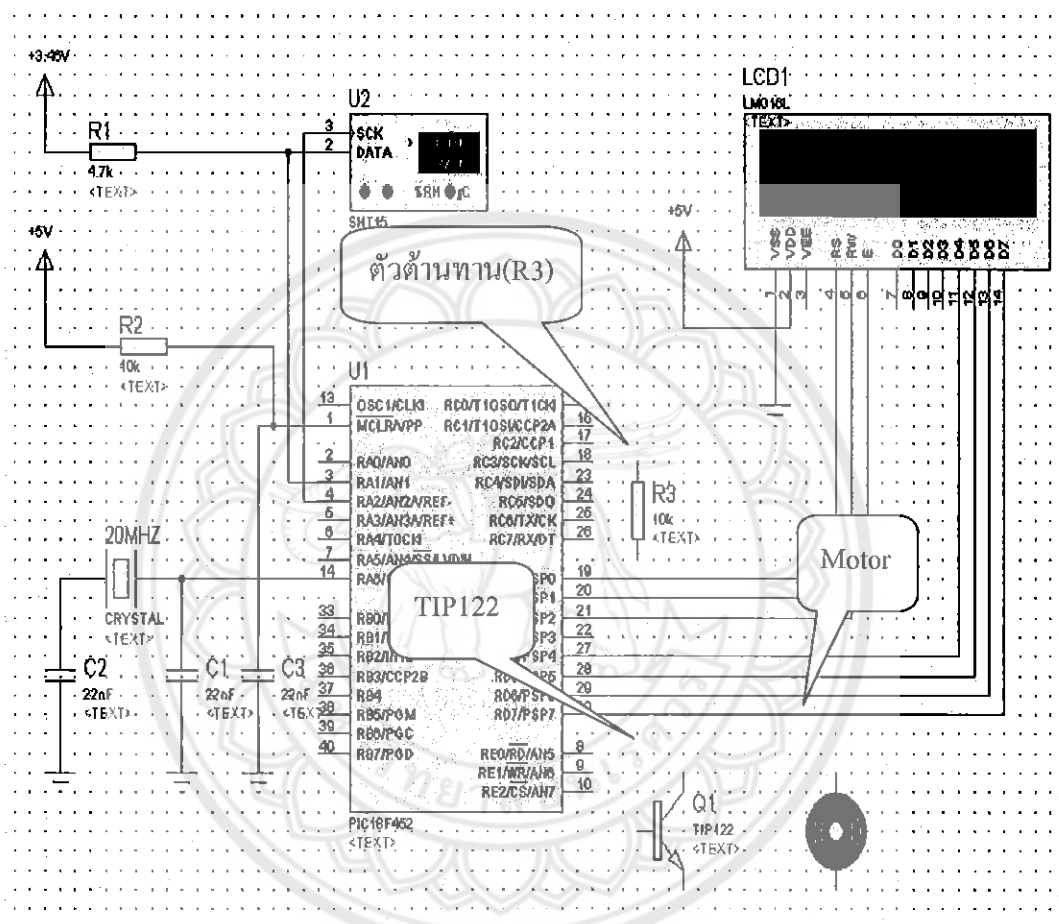


รูปที่ 3.4.32 แสดงการจัดตำแหน่งและต่อวงจรของ CAP และ CRYSTAL



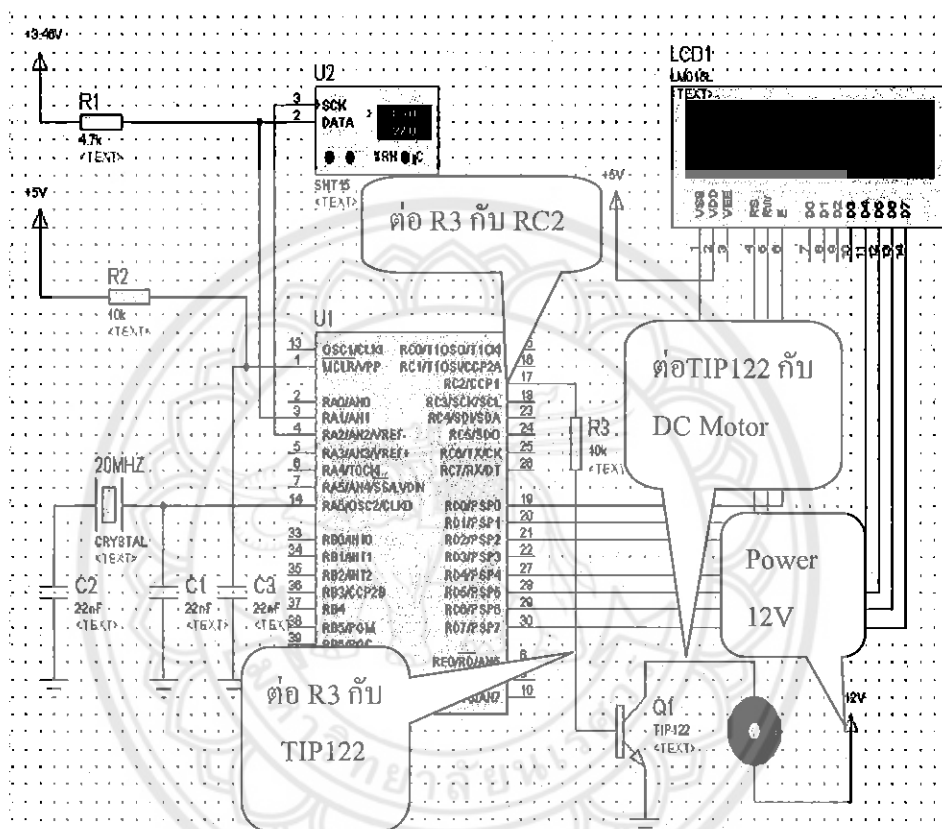
รูปที่ 3.4.33 แสดงการจัดตำแหน่งและต่อวงจรของ CAP และ CRYSTAL

- วางตำแหน่งของตัวต้านทาน(R3)ขนาด 10K, TIP122 และ DC Motor ตามรูป3.34



รูปที่ 3.4.34 แสดงการจัดตำแหน่งของตัวต้านทาน (R3), TIP122 และ Motor

- คอตัวต้านทาน(R3)ขนาด 10k เข้ากับขา RC2 ของ PIC18F452
- คอตัวต้านทาน(R3)ขนาด 10k พร้อมทั้งคอสายดินเข้ากับ TIP122
- คอTIP122 พร้อมทั้งคอ Power ขนาด 12V ให้กับ DC Motor

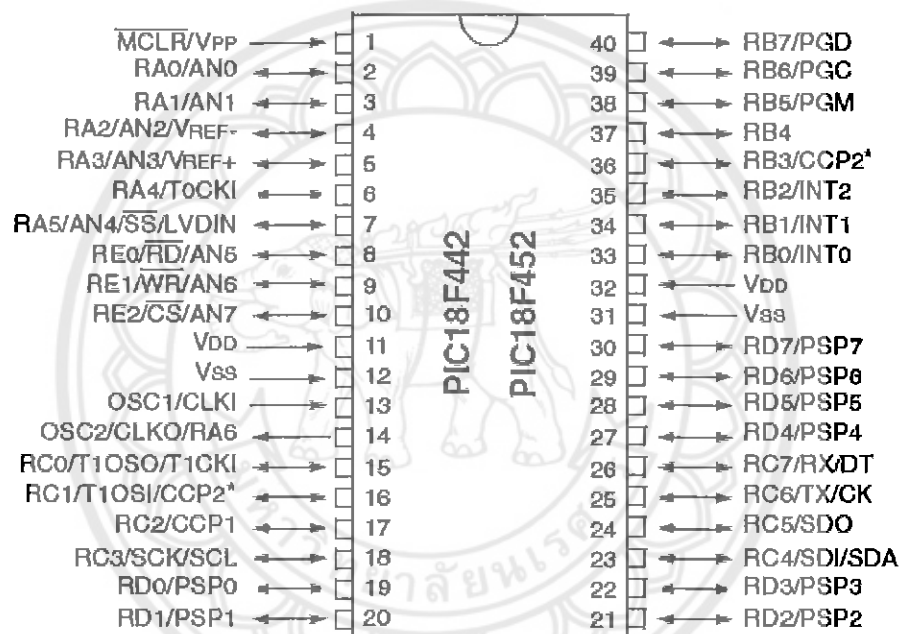


รูปที่ 3.4.35 แสดงวงจรที่ต่อเสร็จสมบูรณ์

เมื่อดาวจรเสร็จสมบูรณ์ให้ตรวจสอบการทำงานของวงจรควบคุมโดยการSimulate วงจร เพื่อเป็นการตรวจสอบความถูกต้องของวงจรควบคุมก่อนนำไปสร้างชุดควบคุมจริง

### 3.4.4 ขั้นตอนการสร้างแผงวงจร

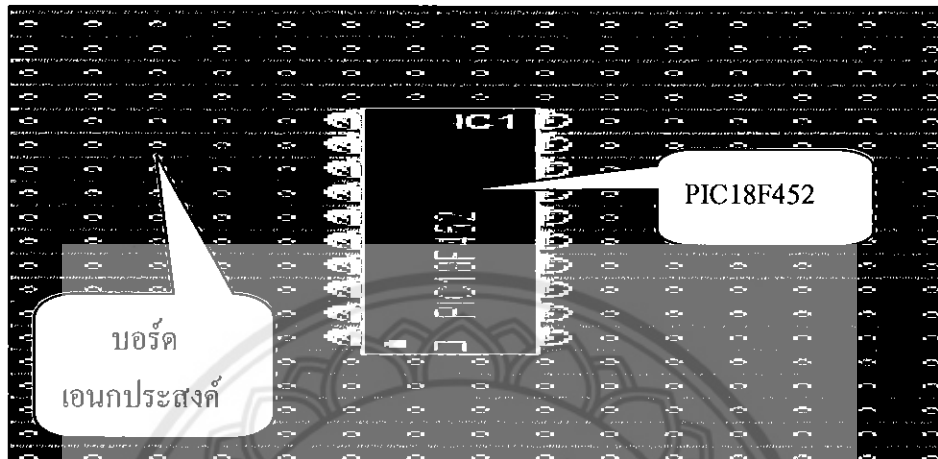
จาก Datasheet ของ PIC18F452 จะทำให้เราทราบว่าแต่ละขาของ IC (ในส่วนนี้จะใช้คำว่า IC เรียกแทน PIC18F452) นั้นมีหน้าที่อย่างไรและตำแหน่งการวาง ซึ่งมีความสำคัญในการสร้าง เนื่องจากต้องมีการบัดกรีแต่ละขาของ IC เข้ากับอุปกรณ์อื่นๆ ซึ่งตำแหน่งขาของ IC แสดงได้ดังนี้



รูปที่ 3.4.36 แสดงตำแหน่งขาของ PIC18F452

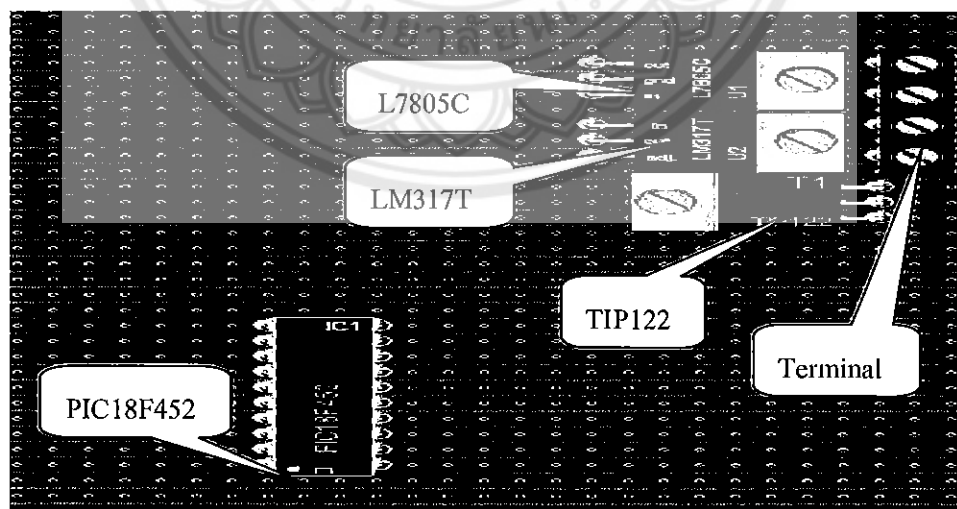
ในการสร้างแผงวงจรนี้จะใช้รูปแบบวงจรจากโปรแกรม Proteus และหน้าที่ของแต่ละอุปกรณ์ได้แสดงไว้แล้วในภาคผนวก

- ขั้นตอนแรกจะวาง IC เข้ากับบอร์ดคอนเนกประสงค์โดยให้ด้านหัว (ด้านขาหมายเลข 1) อยู่ทางด้านล่าง พร้อมทั้งบัดกรีขาทั้ง 40 ขาเข้ากับบอร์ดคอนเนกประสงค์ดังรูป



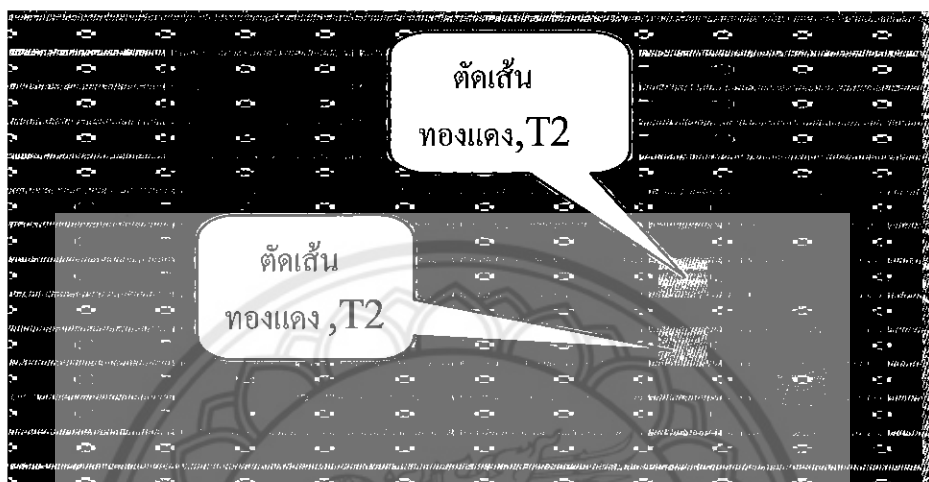
รูปที่ 3.4.37 แสดงการบัดกรีขาของ PIC18F452 เข้ากับบอร์ดคอนเนกประสงค์

- บัดกรี Terminal, L7805C, TIP122 และ LM317T ลงบนบอร์ดคอนเนกประสงค์ ดังรูป



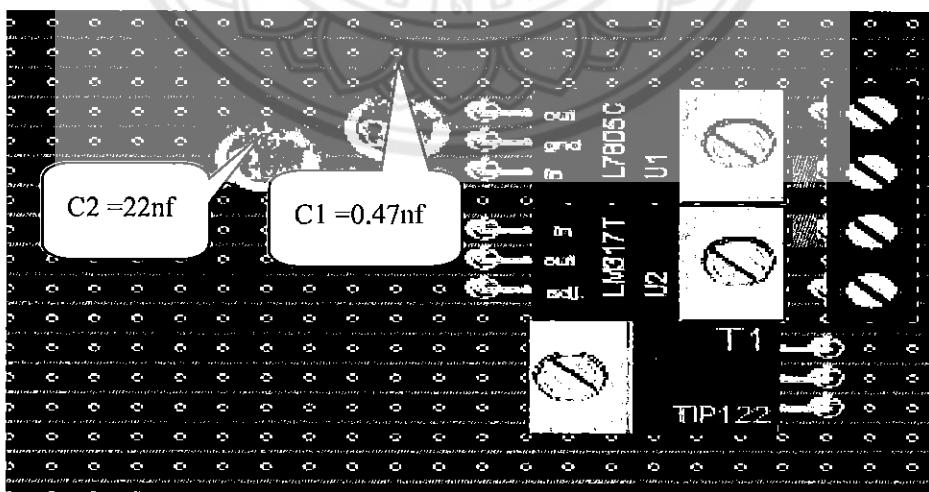
รูปที่ 3.4.38 แสดงการบัดกรี Terminal, L7805C, TIP122 และ LM317T

- พลิกบอร์ดนอนกประสงค์กลับไปด้านหลังเพื่อทำการตัดเส้นทองแดง T2 , T3 ดังรูป



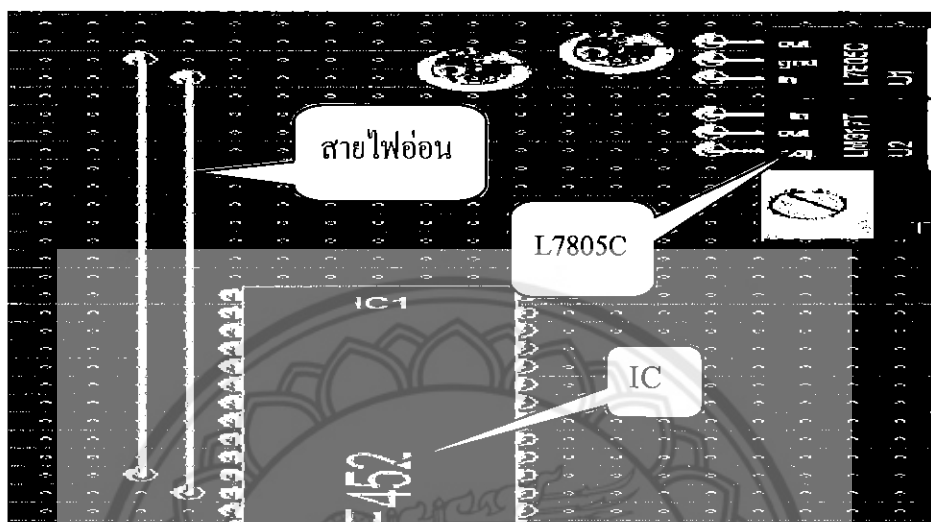
รูปที่ 3.4.39 แสดงการตัดเส้นทองแดง T2, T3

- บัดกรีตัวเก็บประจุเข้ากับบอร์ดนอนกประสงค์ โดย C1 มีค่า 0.47nf และ C2 มีค่า 22nf ดังรูป



รูปที่ 3.4.40 แสดงการบัดกรีตัวเก็บประจุเข้ากับบอร์ดนอนกประสงค์

- ด้านหน้าของบอร์ดคอนเนกประสงค์ ให้ใช้สายไฟอ่อนบัดกรีจาก L7805C เพื่อส่งกระแสไฟฟ้าให้กับ IC โดยขั้วบวกต่อเข้ากับขา 30 และขั้วลบต่อกับขา 31 ดังรูป



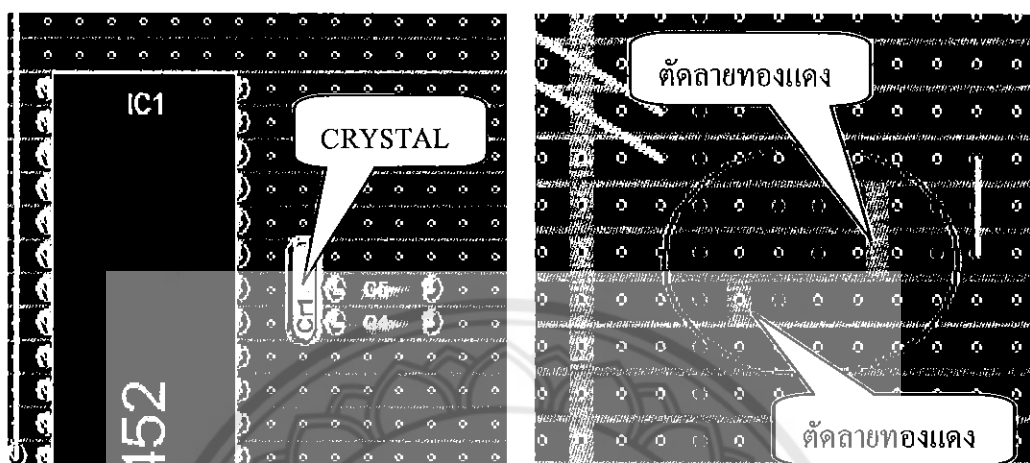
รูปที่ 3.4.41 แสดงการใช้สายไฟอ่อนบัดกรีจาก L7805C เพื่อส่งกระแสไฟฟ้าให้กับ IC

- จากนั้นก็พลิกกลับไปด้านหลังบอร์ดขา 30 เข้ากับขา 11 และ ขา 31 เข้ากับ ขา 12 ดังรูป



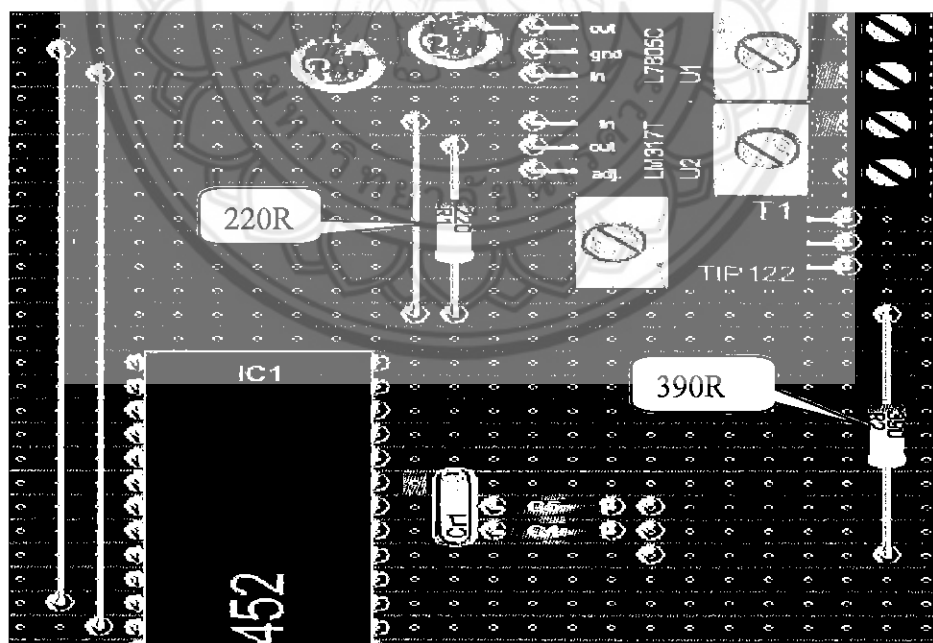
รูปที่ 3.4.42 แสดงการบัดกรีขา 30 เข้ากับขา 11 และ ขา 31 เข้ากับ ขา 12 ของ IC

- บัดกรี CRYSTAL (ต่อกับขา 13 และ 14 ของ IC) และ ตัวเก็บประจุ 22nf เข้ากับบอร์ดอเนกประสงค์ พร้อมตัดลายทองแดง(ด้านหลัง)ดังรูป



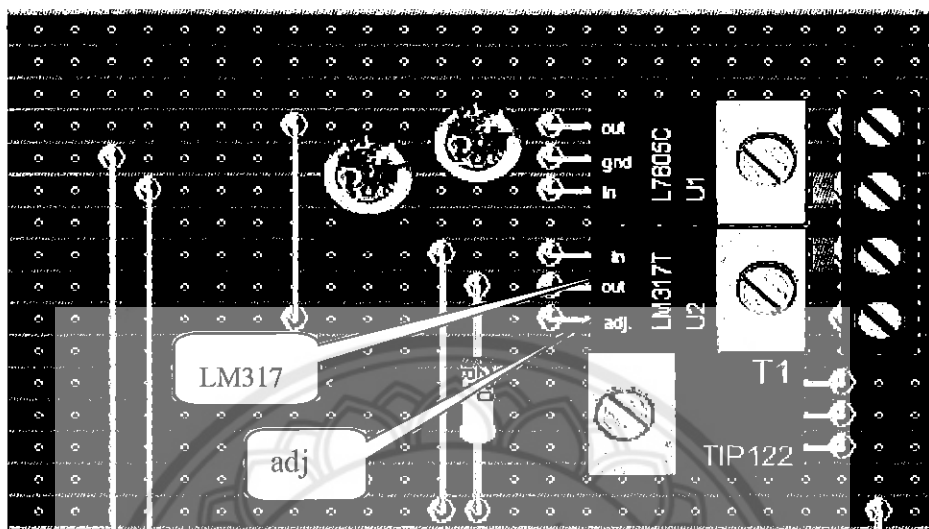
รูปที่ 3.4.43 แสดงการบัดกรี CRYSTAL ต่อกับขา 13 และ 14 ของ IC

- ต่อตัวต้านทาน 290R 390R และสายไฟ ดังรูป



รูปที่ 3.4.44 แสดงการต่อตัวต้านทาน 290R, 390R และสายไฟ

- ต่อสายไฟ 12 V เข้ากับ LM317T และ ขั้ว adj เข้ากับ 220R ดังรูป



รูปที่ 3.4.45 แสดงการต่อสายไฟ 12 V เข้ากับ LM317T และ ขั้ว adj เข้ากับ 220R

- ต่อ Terminal 4 PIN สำหรับต่อ Sensor SHT15 พร้อมทั้งต่อสายไฟขั้วลบเข้ากับ Terminal



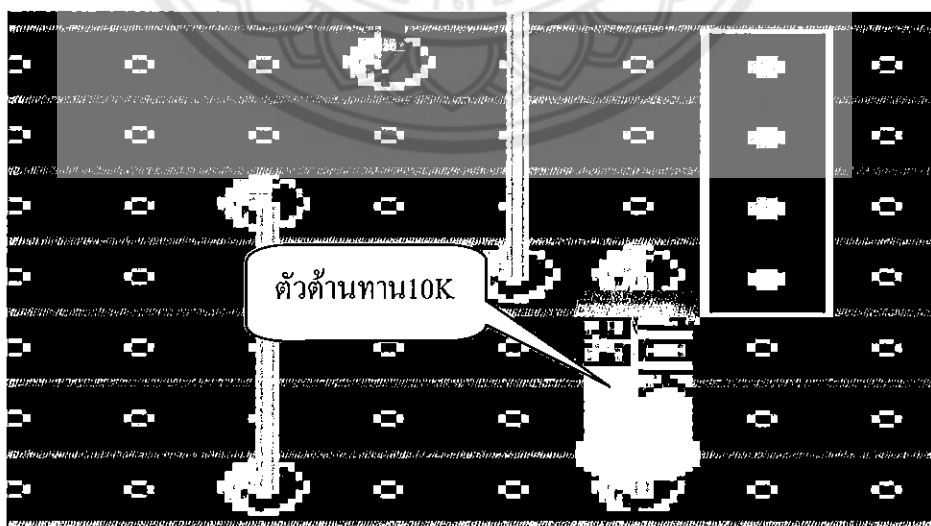
รูปที่ 3.4.46 แสดงการต่อ Terminal 4 PIN สำหรับต่อ Sensor SHT15

- พลิกกลับไปด้านหลังของบอร์ดเนกประสงค์ พร้อมต่อสายไฟจั่วบวกจาก LM317 เข้ากับ PIN บวก



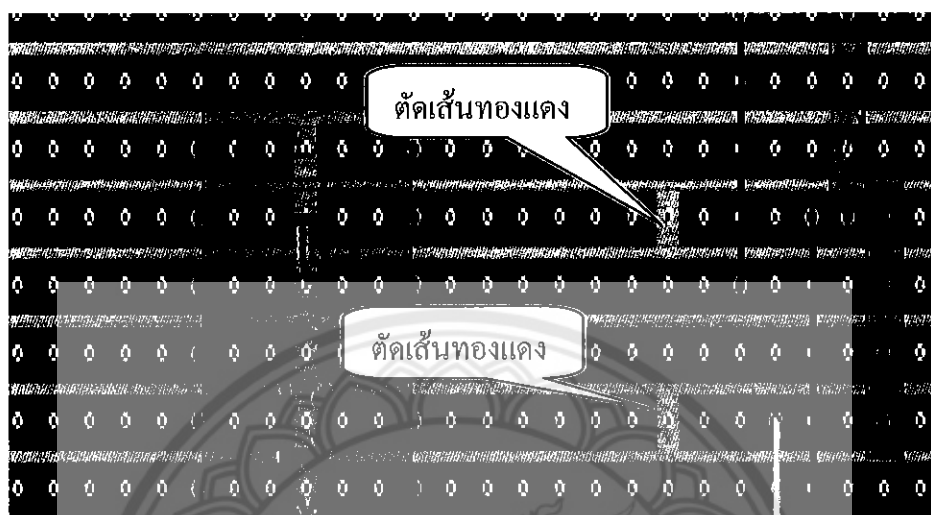
รูปที่ 3.4.47 แสดงการต่อสายไฟจั่วบวกจาก LM317 เข้ากับ PIN บวก

- ที่ด้านหน้าของบอร์ดเนกประสงค์ ต่อตัวต้านทาน 10K เข้ากับ สาย DATA



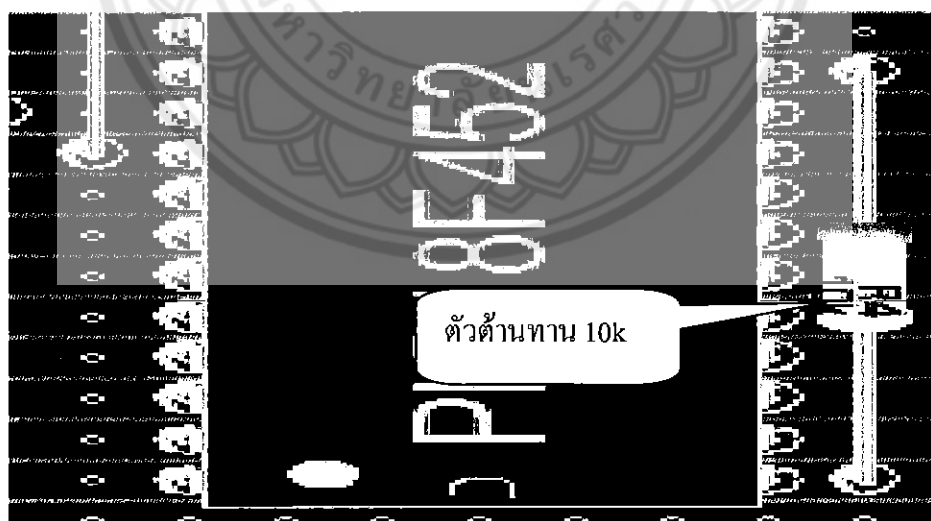
รูปที่ 3.4.48 แสดงการต่อตัวต้านทาน 10K เข้ากับ สาย DATA

- ตัดเส้นทองแดง ระหว่าง ขา 1 กับขั้ว ลบ และ ขา 4 กับขั้วบวก



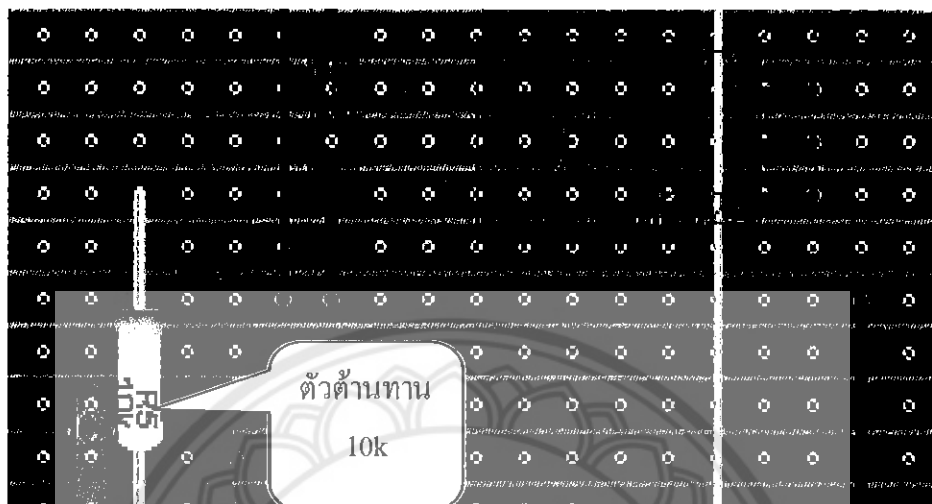
รูปที่ 3.4.49 แสดงการตัดเส้นทองแดง ระหว่าง ขา 1 กับขั้ว ลบ และ ขา 4 กับขั้วบวก

- ต่อตัวต้านทาน 10k ระหว่างขา 1 กับขา 11



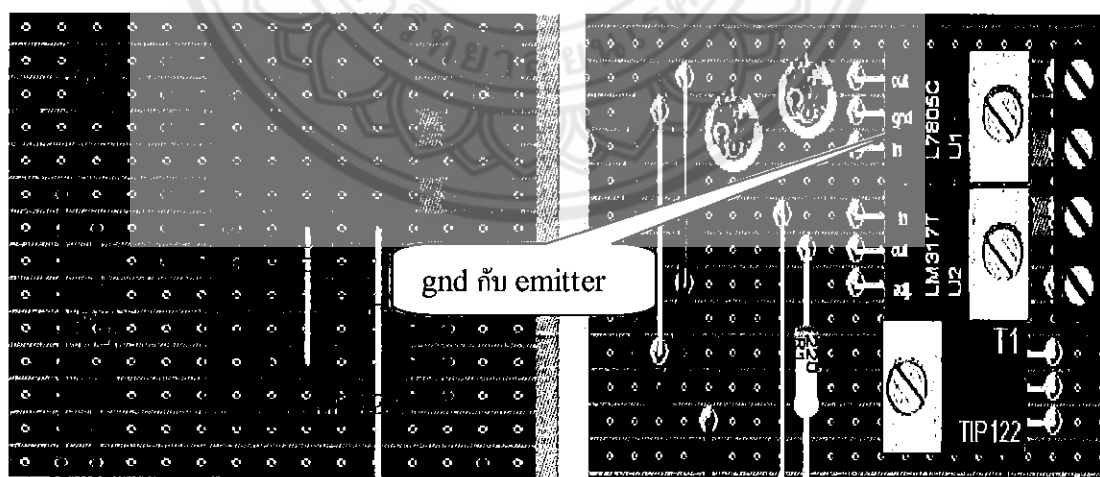
รูปที่ 3.4.50 แสดงการต่อตัวต้านทาน 10k ระหว่างขา 1 กับขา 11

- ต่อตัวต้านทาน 10k ระหว่าง ขา 17 เข้ากับขา Base ของ TIP 122



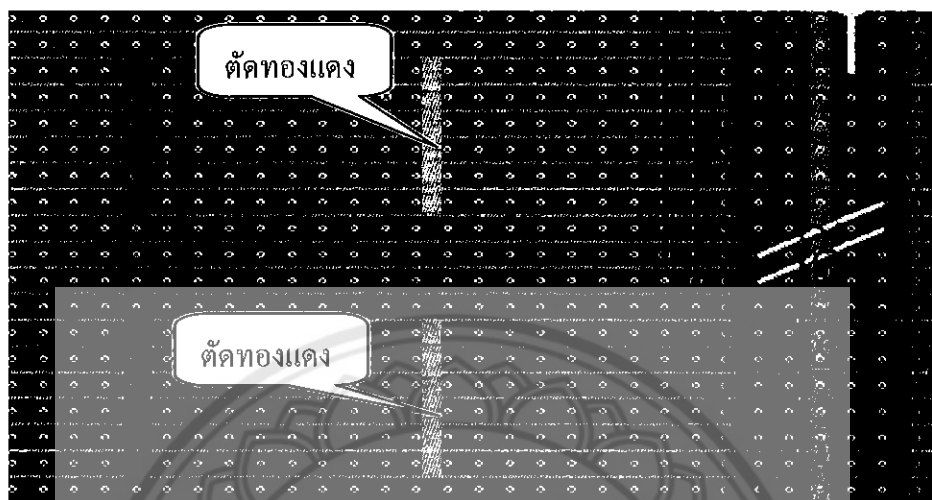
รูปที่ 3.4.51 แสดงการต่อตัวต้านทาน 10k ระหว่างขา 17 เข้ากับขา Base ของ TIP 122

- ต่อสายไฟ gnd จาก L7805C เข้ากับ ขา emitter และ collector เข้ากับ terminal (T3)



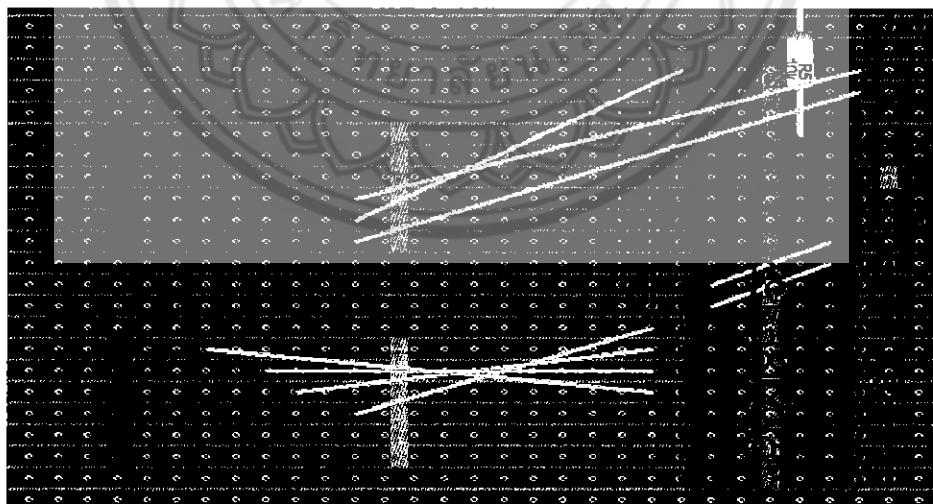
รูปที่ 3.4.52 แสดงการต่อสายไฟ gnd จาก L7805C เข้ากับ ขา emitter และ collector เข้ากับ terminal (T3)

- วาง socket สำหรับ LCD พร้อมกับตัดเส้นทองแดง



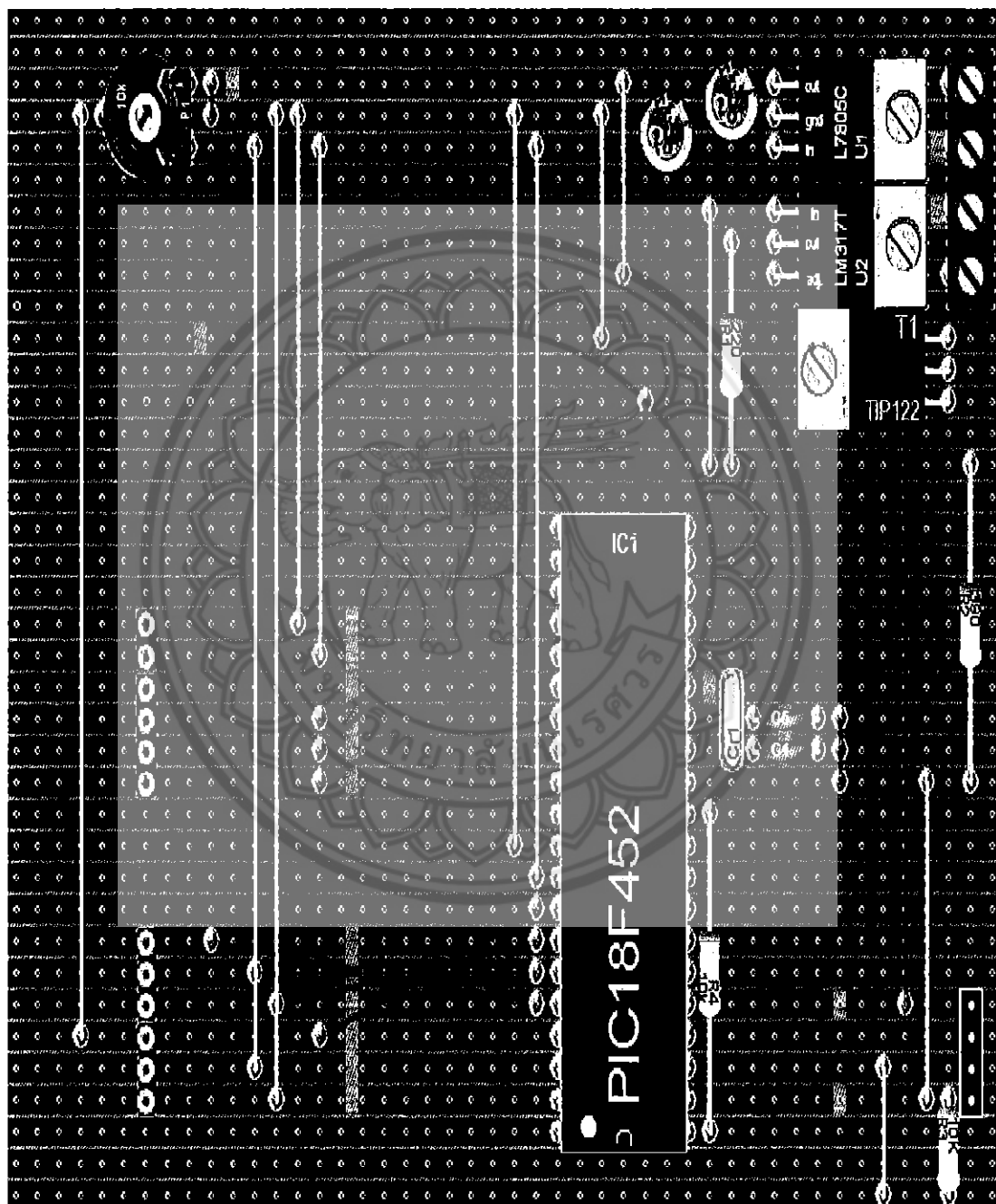
รูปที่ 3.4.53 แสดงการวาง socket สำหรับ LCD พร้อมกับตัดเส้นทองแดง

- ต่อสายไฟสำหรับส่งข้อมูลให้ LCD โดยดูการต่อสายไฟได้จาก วงจรใน Proteus



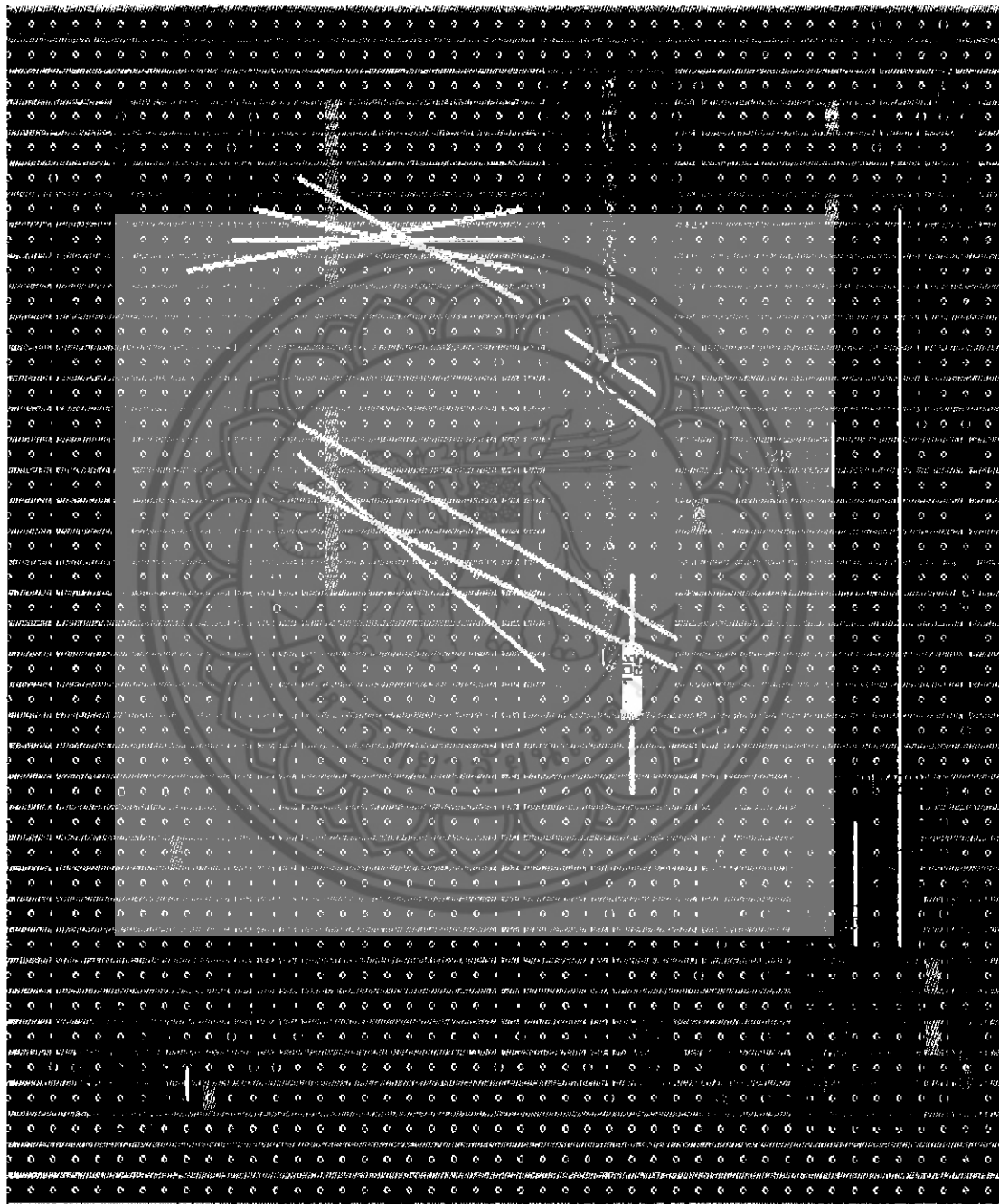
รูปที่ 3.4.54 แสดงการต่อสายไฟสำหรับส่งข้อมูลให้ LCD

- แผงวงจรที่ต่อเสร็จพร้อมใช้งานดังรูปต่อไปนี้



รูปที่ 3.4.55 แสดงแผงวงจรที่ต่อเสร็จสมบูรณ์

● ด้านหลังของแผงวงจรที่ต่อเสร็จสมบูรณ์



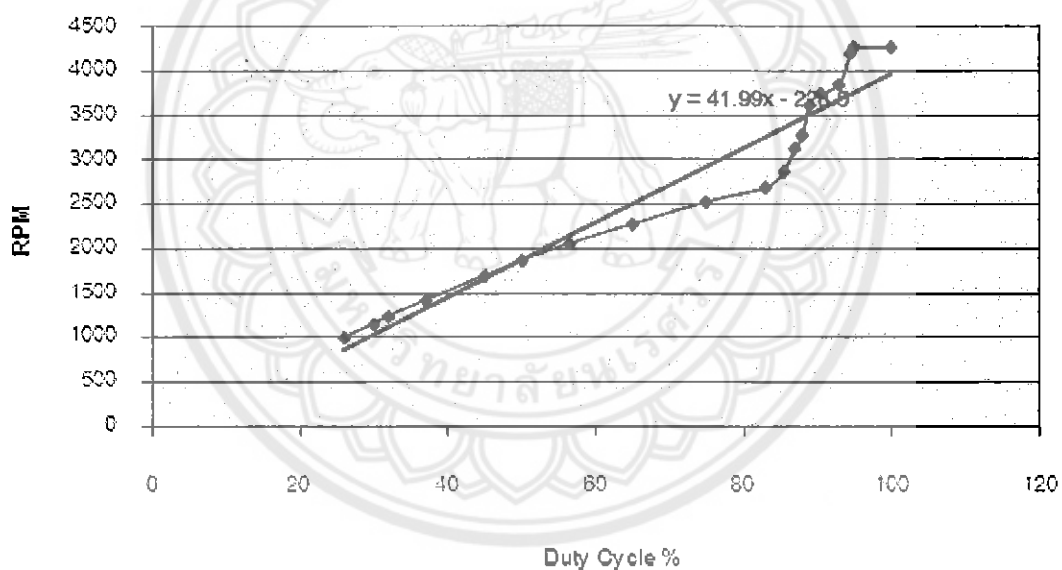
รูปที่ 3.4.56 แสดงภาพด้านหลังของแผงวงจรที่ต่อเสร็จสมบูรณ์

## บทที่ 4

### ผลการดำเนินงานและการวิเคราะห์

ผลของการดำเนินงานที่ได้จากการทดลอง จะนำมาแสดงในรูปของกราฟแสดงความสัมพันธ์ของตัวแปรต่างๆ รวมทั้งนำผลที่ได้มาวิเคราะห์และเปรียบเทียบเพื่อหาผลสรุปต่อไป

#### 4.1 ความสัมพันธ์ของ Duty Cycle กับความเร็วรอบ



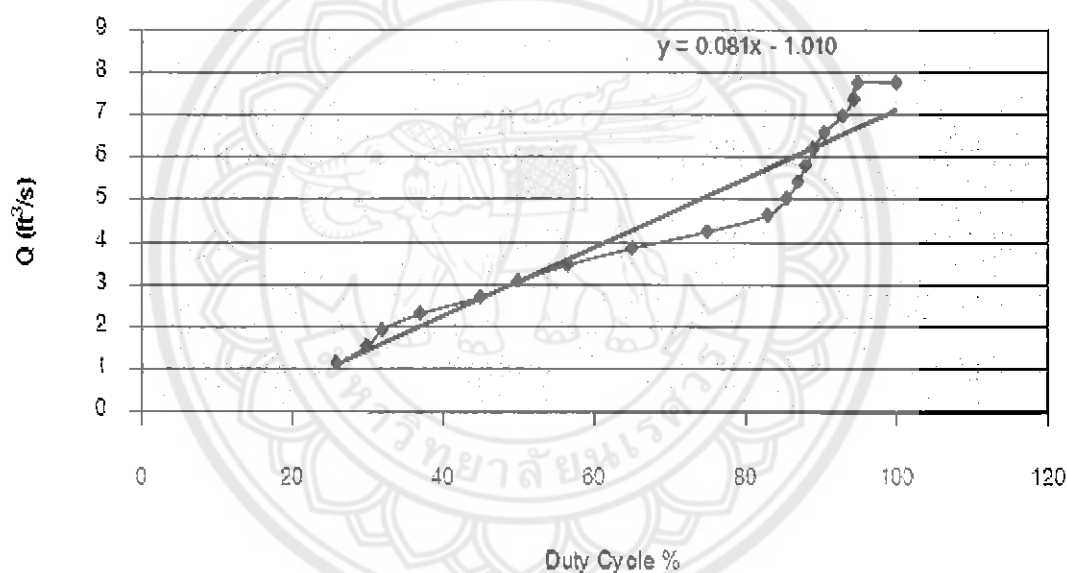
4.1 กราฟเปรียบเทียบ Duty Cycle กับความเร็วรอบ

การเปรียบเทียบ Duty Cycle กับความเร็วรอบ พบว่าในช่วง Duty Cycle น้อยกว่า 26% ความเร็วรอบของมอเตอร์ของพัดลมมีค่าต่ำมาก เนื่องจากแรงขับเคลื่อนไม่สามารถที่จะเอาชนะแรงเสียดทานของพัดลมได้ เมื่อปรับ Duty Cycle อยู่ในช่วง 26 ถึง 80 % ความเร็วรอบของมอเตอร์ของพัดลม (RPM) มีค่ามากขึ้น ซึ่งการเพิ่มขึ้นในช่วงนี้มีลักษณะค่อนข้างเป็นเส้นตรง แต่ยังคงมีความคลาดเคลื่อนเกิดขึ้นจึงทำให้กราฟที่ได้คลาดเคลื่อนตาม และเมื่อปรับ Duty Cycle อยู่

ในช่วงตั้งแต่ 80% ขึ้นไปความเร็วรอบของมอเตอร์ของพัดลมมีค่าเพิ่มขึ้นอย่างรวดเร็ว โดยสังเกตได้จากกราฟที่มีความชันสูงขึ้น และที่ Duty Cycle มีค่ามากกว่า 95% จะได้ความเร็วพัดลมมีค่าสูงสุดที่ 4270 RPM

ดังนั้นสามารถกล่าวได้ว่า Duty Cycle แปรผันโดยตรง กับความเร็วรอบโดย RPM เป็นฟังก์ชันของ Duty Cycle ซึ่งสมการแสดงความสัมพันธ์ คือ  $RPM = 41.99 \text{Duty Cycle} - 228.53$  โดยสมการแสดงความสัมพันธ์ดังกล่าวสามารถนำไปใช้เพื่อหาความเร็วรอบ(RPM)

#### 4.2 ความสัมพันธ์ของ Duty Cycle กับอัตราการไหลของอากาศ



#### 4.2 กราฟเปรียบเทียบ Duty Cycle กับอัตราการไหลของอากาศ

การเปรียบเทียบ Duty Cycle กับอัตราการไหลของอากาศ ในช่วงที่มีค่า Duty Cycle น้อยกว่า 260 อัตราการไหลของอากาศมีค่าน้อยมาก เนื่องจากแรงขับเคลื่อนไม่สามารถที่จะเอาชนะแรงเฉื่อยของพัดลมได้ เมื่อทำการปรับค่า Duty Cycle ให้อยู่ในช่วง 260 ถึง 800 ส่งผลให้อัตราการไหลของอากาศมีค่ามากขึ้นลักษณะค่อนข้างเป็นเส้นตรง แต่ยังคงมีการแกว่งตัวเกิดขึ้น และเมื่อเราปรับค่า Duty Cycle ให้อยู่ในช่วงตั้งแต่ 800 ขึ้นไปพบว่า อัตราการไหลของอากาศมีค่าเพิ่มขึ้นอย่างรวดเร็ว โดยสังเกตได้จากกราฟที่มีความชันสูงขึ้น และ Duty Cycle ประมาณ 950 จะให้ค่าสูงสุดของอัตราการไหลของอากาศที่ประมาณ 2 ft³/s

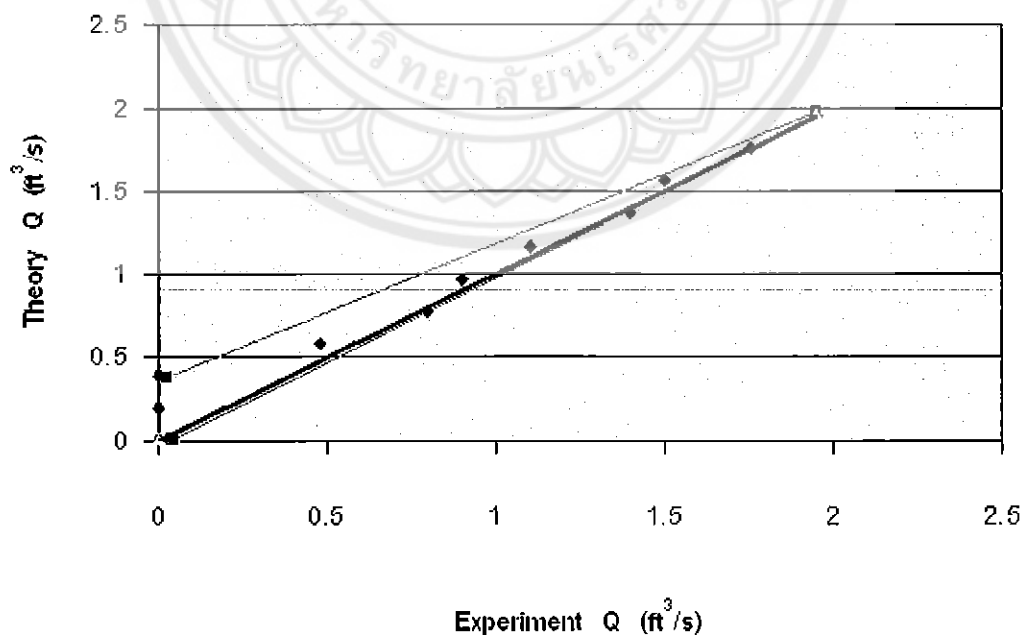
ดังนั้น Duty Cycle แปรผันโดยตรงกับอัตราไหลของอากาศ ในช่วงที่ค่า Duty Cycle อยู่ระหว่าง 260 – 800 โดย  $Q$  เป็นฟังก์ชันของ Duty Cycle ซึ่งมีสมการแสดงความสัมพันธ์เป็น  $Q = 0.002 \text{ Duty cycle} - 0.2531$  โดยสมการแสดงความสัมพันธ์ดังกล่าวสามารถนำไปใช้เพื่อหาอัตราไหลของอากาศ ( $\text{ft}^3/\text{s}$ ) โดยใช้ได้เฉพาะกับพัดลม DC 12 V รุ่น C 8025 12V และ 0.25 A

#### 4.3 อัตราการไหลของอากาศที่ได้จากการทดลองเปรียบเทียบกับทฤษฎี

อัตราการไหลของอากาศเราสามารถคำนวณจาก  $Q = VA$  เนื่องจากเครื่องมือวัดได้เป็นความเร็ว ( $\text{ft}/\text{s}$ ) และพื้นที่ ( $\text{ft}^2$ ) จะได้ค่าอัตราการไหลของอากาศได้จากตารางที่ 4.3

| Theory                     | 0 | 10    | 20   | 30    | 40   | 50   | 60   | 70   | 80   | 90   | 100  |
|----------------------------|---|-------|------|-------|------|------|------|------|------|------|------|
| RPM                        | 0 | 384.3 | 854  | 1281  | 1708 | 2135 | 2562 | 2989 | 3416 | 3843 | 4270 |
| $Q (\text{ft}^3/\text{s})$ | 0 | 0.195 | 0.39 | 0.585 | 0.78 | 0.98 | 1.17 | 1.37 | 1.56 | 1.76 | 1.95 |
| Experiment                 |   |       |      |       |      |      |      |      |      |      |      |
| RPM                        | 0 | 0     | 0    | 1281  | 1708 | 2135 | 2562 | 2989 | 3416 | 3843 | 4270 |
| $Q (\text{ft}^3/\text{s})$ | 0 | 0     | 0    | 0.48  | 0.8  | 0.9  | 1.1  | 1.4  | 1.5  | 1.75 | 1.95 |

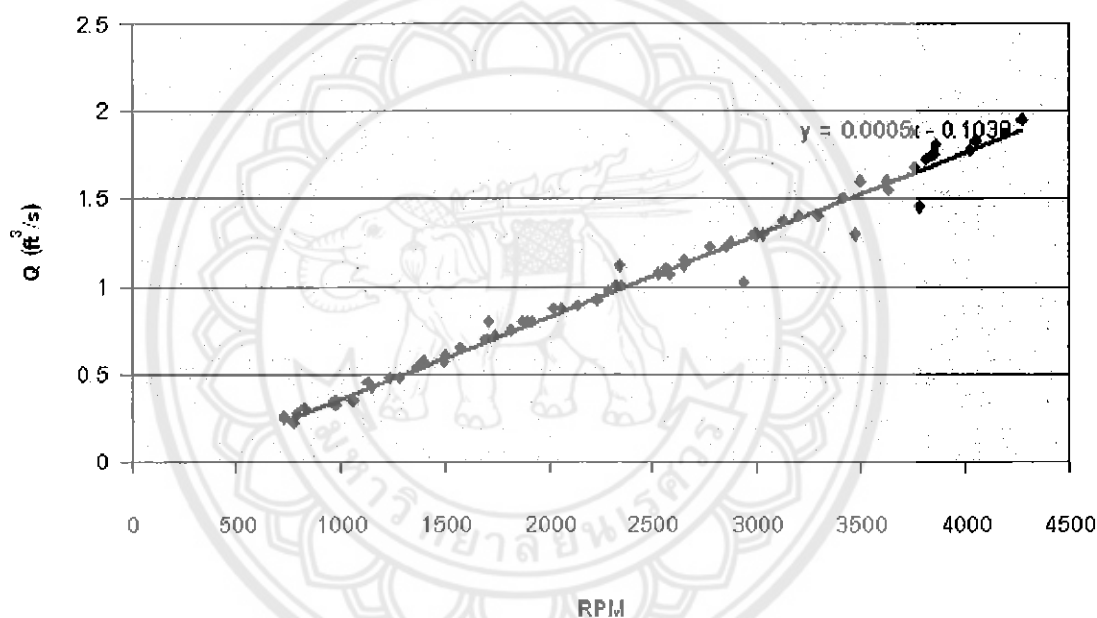
ตาราง 4.3 ค่าของอัตราการไหลของอากาศจากการทดลองเปรียบเทียบกับทฤษฎี



#### 4.3 อัตราการไหลของอากาศที่ได้จากการทดลองเปรียบเทียบกับทฤษฎี

จากกราฟพบว่าอัตราไหลของอากาศจะมีค่าความผิดพลาดค่อนข้างสูงในช่วงเริ่มต้นซึ่งมีอัตราไหลของอากาศต่ำๆ และเมื่ออัตราไหลของอากาศมีค่าเพิ่มสูงขึ้น ค่าของความผิดพลาดมีแนวโน้มที่จะลดลง ซึ่งค่าความผิดพลาดอาจมาจากหลายสาเหตุ ทั้งค่าความผิดพลาดที่เกิดจากเครื่องมือที่ใช้ทำการทดลอง, ตำแหน่งที่ใช้วัดค่า ตลอดจนผู้ปฏิบัติการทดลอง ดังนั้นการนำผลลัพท์ไปใช้งานค่าความผิดพลาดที่เกิดขึ้นประมาณ +4% และ -1%

#### 4.4 ความสัมพันธ์ของความเร็วรอบของมอเตอร์และอัตราการไหลของอากาศ

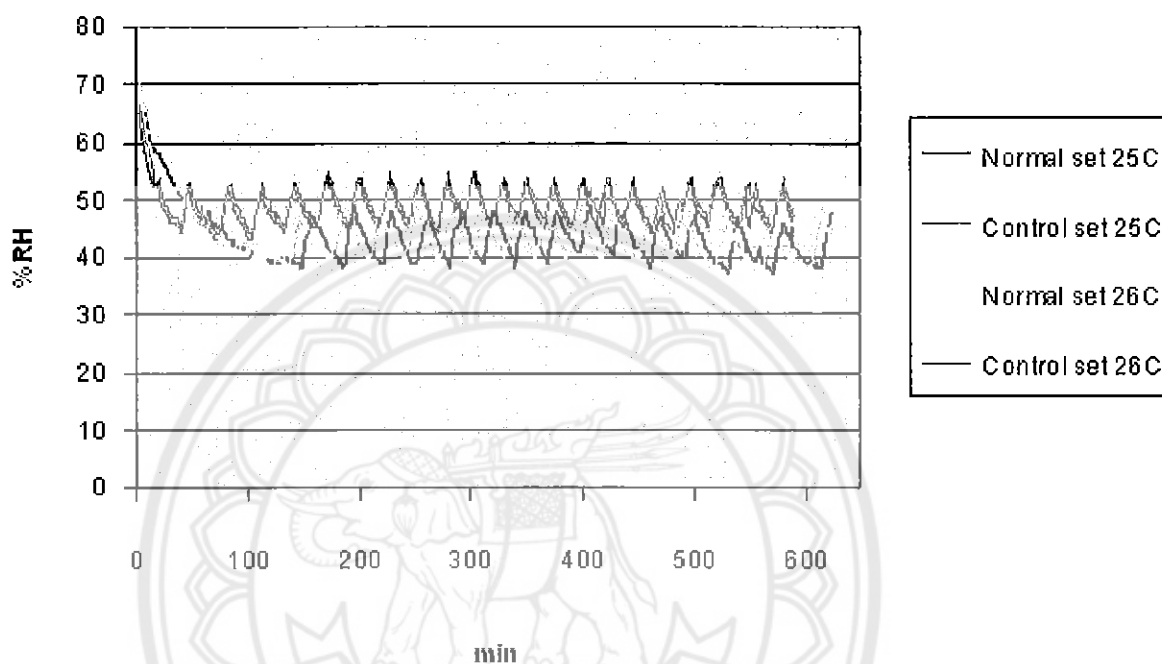


4.4 กราฟเปรียบเทียบความเร็วรอบของมอเตอร์และอัตราการไหลของอากาศ

จากการทดลองเปรียบเทียบความเร็วรอบของมอเตอร์พัดลมและอัตราการไหลของอากาศ พบว่าชุดข้อมูลที่ได้มีแนวโน้มเป็นเส้นตรง ซึ่งหมายถึงเมื่อความเร็วรอบของมอเตอร์พัดลม เพิ่มขึ้น จะส่งผลให้มีอัตราการไหลของอากาศ มีค่าเพิ่มขึ้น และมีความคลาดเคลื่อนของข้อมูลบางค่าที่ได้จากการทดลอง ซึ่งอาจเกิดจากความคลาดเคลื่อนของเครื่องมือที่ใช้ทำการทดลอง

ดังนั้นสามารถกล่าวได้ว่าความเร็วรอบแปรผันโดยตรงกับอัตราการไหลของอากาศ โดยอัตราการไหลของอากาศเป็นฟังก์ชันของความเร็วรอบของมอเตอร์พัดลมซึ่งสมการแสดงความสัมพันธ์คือ  $Q = 0.0005RPM - 0.1038$

#### 4.5 ความชื้นสัมพัทธ์ ณ เวลาต่างๆของห้องปรับอากาศที่อุณหภูมิ 25 °Cและ26°C



4.5 กราฟความชื้นสัมพัทธ์ ณ เวลาต่างๆของห้องปรับอากาศ

กราฟที่ 4.5 แสดงถึงความชื้นสัมพัทธ์ ที่เวลาต่างๆของห้องปรับอากาศ ซึ่งกราฟเส้นสีน้ำเงินเป็นระบบที่ไม่ได้ควบคุมความชื้นตั้งอุณหภูมิ 25 °C กราฟเส้นสีชมพูเป็นระบบที่ใช้อุปกรณ์ควบคุมความชื้นตั้งอุณหภูมิ 25 °C เส้นสีเหลืองเป็นระบบที่ไม่ได้ควบคุมความชื้นตั้งอุณหภูมิ 26 °C และกราฟเส้นสีฟ้าเป็นระบบที่ใช้อุปกรณ์ควบคุมความชื้นตั้งอุณหภูมิ 26 °C แกนในแนวนอน (X) เป็นเวลา(min) และแกนในแนวตั้ง (Y) เป็นความชื้นสัมพัทธ์ พบว่า ณ เวลาเริ่มต้นอากาศมีปริมาณความชื้นสูงประมาณ 69 %RH และเมื่อทำการปรับอากาศจะส่งผลให้ ความชื้นของอากาศลดลงเรื่อยๆ จนถึงค่าต่ำสุดซึ่งมีความชื้นสัมพัทธ์ประมาณ 37 %RH และสูงสุดที่ 44 %RH ในการปรับอากาศโดยไม่ใช้ชุดควบคุมความชื้นจะมีช่วงการแกว่งของความชื้นสัมพัทธ์อยู่ที่ประมาณ 37 – 44 %RH และในส่วนของการปรับอากาศโดยใช้ชุดควบคุมความชื้น พบว่าค่าต่ำสุดของความชื้นสัมพัทธ์ประมาณ 44 %RH และสูงสุดที่ประมาณ 53 %RH และมีช่วงการแกว่งของความชื้นสัมพัทธ์อยู่ที่ประมาณ 44 – 53 %RH และการตั้งค่าอุณหภูมิมีผลต่อความชื้นสัมพัทธ์ ถ้าตั้งอุณหภูมิ

ที่สูงความชื้นก็จะสูง ในทางกลับกันถ้าตั้งค่าอุณหภูมิที่ต่ำความชื้นก็จะต่ำ ซึ่งการใช้ชุดควบคุมความชื้นจะส่งผลให้ความชื้นของห้องที่ปรับอากาศไม่ลดต่ำจนเกินไป ซึ่งจะช่วยในการควบคุมสภาวะความสบายของมนุษย์ เหตุที่ความชื้นสัมพัทธ์ของอากาศมีช่วงการแกว่งเนื่องจากเครื่องปรับอากาศที่ใช้ไม่ใช่ระบบอินเวอร์เตอร์ จึงทำให้ความชื้นแกว่งตามการทำงานของเครื่องปรับอากาศ ซึ่งความชื้นสัมพัทธ์จะมีค่าเพิ่มขึ้นในช่วงที่เครื่องปรับอากาศหยุดทำงาน และมีค่าลดต่ำลงในช่วงที่เครื่องปรับอากาศทำงาน และการทำงานของเครื่องปรับอากาศจะเป็นการทำงานช่วงสั้นๆ ซึ่งขึ้นกับอุณหภูมิที่ผู้ใช้งานกำหนด

การใช้ชุดควบคุมความชื้นร่วมกับการปรับอากาศเป็นอีกแนวทางหนึ่งในควบคุมความชื้นของห้องปรับอากาศ ซึ่งสามารถควบคุมได้ในระดับหนึ่ง แต่ยังไม่สามารถจะควบคุมได้ให้ความชื้นคงที่ได้ เนื่องจากห้องที่นำมาทดลองนั้นมีขนาดใหญ่ และเครื่องทำความชื้นมีขนาดเล็ก แต่อย่างไรก็ตามชุดควบคุมความชื้นเร็วพัฒนา ก็สามารถเป็นแนวทางในการนำไปประยุกต์ใช้กับการควบคุมความเร็วของพัดลมที่อิวาปไปเรเตอร์ของเครื่องปรับอากาศต่อไป



## บทที่ 5

### สรุปและข้อเสนอแนะ

#### 5.1 สรุป

การควบคุมความเร็วรอบของมอเตอร์กระแสตรง (DC) ซึ่งความกว้างของ Duty Cycle มีผลต่อความเร็วรอบ ซึ่งช่วงของ Duty Cycle ที่กว้างจะส่งผลให้ความเร็วรอบของมอเตอร์สูง และช่วงของ Duty Cycle ที่แคบจะส่งผลให้ความเร็วรอบของมอเตอร์ต่ำ ดังนั้นสามารถกล่าวได้ว่า Duty Cycle แปรผันโดยตรงกับความเร็วรอบของมอเตอร์ โดยที่ RPM เป็นฟังก์ชันของ Duty Cycle ซึ่งสมการแสดงความสัมพันธ์ คือ  $RPM = 4.2 \text{Duty Cycle} - 228.53$

การเปรียบเทียบ Duty Cycle กับอัตราการไหลของอากาศ ในช่วงที่มีค่า Duty Cycle ต่ำๆ อัตราการไหลของอากาศมีค่าต่ำ เมื่อทำการปรับค่า Duty Cycle ให้อยู่ในช่วง 260 ถึง 800 ส่งผลให้อัตราการไหลของอากาศมีค่ามากขึ้นและมีลักษณะเป็นเส้นตรง แต่ยังคงมีการแกว่งตัวเกิดขึ้น และเมื่อเราปรับค่า Duty Cycle ให้อยู่ในช่วงตั้งแต่ 800 ขึ้นไปพบว่า อัตราไหลของอากาศมีค่าเพิ่มขึ้นอย่างรวดเร็ว โดยสังเกตได้จากกราฟที่มีความชันสูงขึ้น ดังนั้นสามารถกล่าวได้ว่า Duty Cycle แปรผันโดยตรงกับอัตราการไหลของอากาศ ในช่วงที่ค่า Duty Cycle อยู่ระหว่าง 260 – 800 โดย Q เป็นฟังก์ชันของ Duty Cycle ซึ่งสมการแสดงความสัมพันธ์คือ  $Q = 0.002 \text{Duty cycle} - 0.2531$

การเปรียบเทียบอัตราการไหลของอากาศที่ได้จากการทดลองกับทฤษฎี พบว่าอัตราการไหลของอากาศจะมีค่าความผิดพลาดค่อนข้างสูงในช่วงเริ่มต้น ที่มีอัตราการไหลของอากาศต่ำๆ และเมื่ออัตราการไหลของอากาศมีค่าเพิ่มสูงขึ้น ค่าของความผิดพลาดมีแนวโน้มที่จะลดลง ดังนั้นการนำพัฒนาไปใช้งานค่าความผิดพลาดที่อาจจะเกิดขึ้นประมาณ +4% และ -1%

การทดลองเปรียบเทียบความเร็วรอบของมอเตอร์พัฒนาและอัตราการไหลของอากาศ พบว่าชุดข้อมูลที่ได้มีแนวโน้มเป็นเส้นตรง ซึ่งหมายถึงเมื่อความเร็วรอบของมอเตอร์พัฒนาที่เพิ่มขึ้นจะส่งผลให้อัตราการไหลของอากาศมีค่าสูงขึ้น ดังนั้นสามารถกล่าวได้ว่าความเร็วรอบแปรผันโดยตรงกับอัตราการไหลของอากาศ โดยอัตราการไหลของอากาศ เป็นฟังก์ชันของความเร็วรอบของมอเตอร์พัฒนาซึ่งสมการแสดงความสัมพันธ์คือ  $Q = 0.0005 \text{RPM} - 0.1038$

จากการทดลองพบว่า ณ เวลาเริ่มต้นอากาศมีปริมาณความชื้นสัมพัทธ์ประมาณ 68 %RH และเมื่อทำการปรับอากาศจะส่งผลให้ ความชื้นสัมพัทธ์ของอากาศลดลงเรื่อยๆ จนถึงค่าต่ำสุดซึ่งมีค่าความชื้นสัมพัทธ์ประมาณ 37 %RH และสูงสุดที่ 44 %RH และในการปรับอากาศโดยไม่ใช้ชุดควบคุมความชื้นจะมีช่วงการแกว่งของความชื้นสัมพัทธ์อยู่ที่ประมาณ 37 – 44 %RH และในส่วนของการปรับอากาศโดยใช้ชุดควบคุมความชื้น พบว่าค่าต่ำสุดของความชื้นสัมพัทธ์ประมาณ 44 %RH และสูงสุดที่ประมาณ 53 %RH และมีช่วงการแกว่งของความชื้นสัมพัทธ์อยู่ที่ประมาณ 44 – 53 %RH ซึ่งการใช้ชุดควบคุมความชื้นร่วมกับการปรับอากาศจะส่งผลให้ความชื้นของห้องที่ปรับอากาศไม่ลดต่ำจนเกินไป ซึ่งจะช่วยในการควบคุมสภาวะความสบายของมนุษย์ และเพื่อเป็นแนวทางในการนำไปประยุกต์ใช้ในการควบคุมความเร็วพัดลมที่อวาล์วโปรเตอร์ของเครื่องปรับอากาศต่อไป

## 5.2 ปัญหาและข้อเสนอแนะ

5.2.1. ค่าของอุณหภูมิและความชื้นสัมพัทธ์ที่วัดได้จาก SHT15 เมื่อนำมาแสดงผลที่ LCD มีปัญหาค่าที่แสดงผลเกิดการค้าง ซึ่งอาจมีสาเหตุจากโค้ดในส่วนของการสั่งงานและอ่านค่าของเซนเซอร์ (SHT 15)

5.2.32. กระแสไฟฟ้าที่เข้า SHT 15 อยู่ในช่วง 2.4-5.5V ดังนั้นควรใช้ Regulator ในการแปลงกระแสไฟฟ้าเพื่อป้องกัน SHT 15 เสียหาย

5.2.3. อุปกรณ์โปรแกรมข้อมูล ใช้ได้กับเฉพาะ Port R232 ซึ่งต้องใช้กับคอมพิวเตอร์ในห้อง lab ไม่สามารถใช้ post USB 2.0 ในการลงข้อมูลลงไมโครชิพ PIC18f452

5.2.4. เกิดปัญหาพัดลมไม่ทำงานที่ค่า Duty Cycle ต่ำๆ เพราะพัดลมมีแรงเสียดทาน ดังนั้นในการเริ่มต้นใช้งานควรปรับให้พัดลมทำงานที่ Duty Cycle สูงๆ หลังจากนั้นจึงปรับให้พัดลมทำงานที่ความเร็วรอบปกติ

5.2.5. การต่อแผงวงจรควรทำการตรวจสอบอย่างละเอียด เนื่องจากการต่อวงจรในโปรแกรมอาจจะถูกต้องและใช้งานได้ แต่ในการสร้างแผงวงจรจริงอาจจะใช้งานไม่ได้

5.2.6. เกิดปัญหาในการจัดหาอุปกรณ์ที่ใช้สร้างชุดควบคุมจึงส่งผลให้แผนการดำเนินเกิดความคลาดเคลื่อนเนื่องจากต้องรออุปกรณ์ ยกตัวอย่างเช่น PIC18f452 และ LCD16x2 ซึ่งสามารถสั่งซื้อจากทางเว็บไซต์ [www.happyMCU.com](http://www.happyMCU.com)

### เอกสารอ้างอิงและบรรณานุกรม

1. N.Rachapradit, "Effect of refrigerant flow and air flow on performance of Inverter split type air-conditioner," King Mongkut's University of Technology Thonburi, Cdastr 2007 international conferent.
2. Edward G.PITA(2002), "Air Conditioning Principles and System 4 th".  
PrenticeHall, New Jersey, Ohio.
3. ASHRAE 51-1981, (1981) "ASHRAE Standard Thermal Environment Condition For Human Occupancy". American Society of Heating, Refrigerating and Air Conditioning Engineer (ASHRAE), Atlanta GA USA.
4. Evans, M. (1980) Housing, "Climate and Comfort". The Architectural Press, London.
5. Fanger, P. O. (1970) "Thermal Comfort". Danish Technical Press, Copenhagen.
6. Oseland, N. (1992) "Thermal Comfort or Complaisance?" Building Service, July 1992, pp.51.
7. Raw, G.J. and Oseland, N. (1984) "Why another thermal comfort conference?", "Thermal Comfort Past, Present and Future". Proceeding of a conference held at Building Research Establishment, Garton 9-10 June 1993, pp.1-10.
8. เรวัตร ขาโต, โอฟาร สืบพันธุ์, "การศึกษาความเร็วของอากาศที่อ้าวไปโปรเตอร์ที่มีผลต่อความสบายเชิงความร้อนสำหรับเครื่องปรับอากาศระบบอินเวอร์เตอร์จากข้อมูลการทดลอง" .ปริญญานิพนธ์ สาขาวิศวกรรมเครื่องกล, มหาวิทยาลัยนเรศวร. 2550.





## อุปกรณ์ที่ใช้สร้างชุดควบคุมความเร็วพัดลม

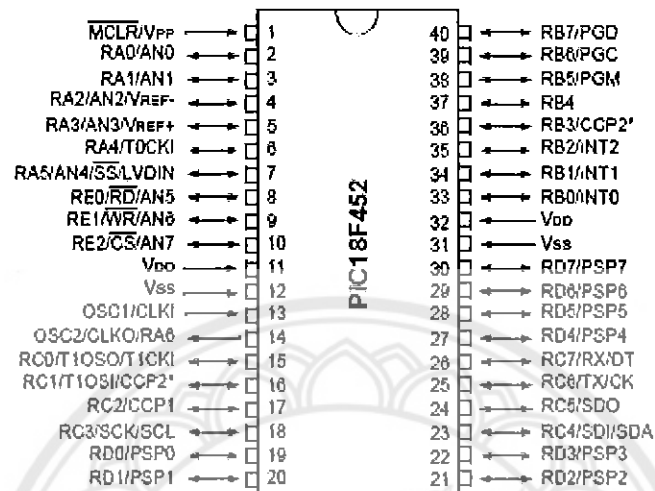


รูปที่ ก 1 PIC18F425

PIC ย่อมาจากคำว่า Peripheral Interface Controller ซึ่ง microcontroller ประกอบด้วย PROGRAM MEMORY, RAM, EEPROM, SERIAL, I2C, PWM, A/D ฯลฯ รวมทั้งมีฟังก์ชันประมวลผล หน่วยความจำ การทำงานคล้ายกับ CPU ของ computer สามารถทำงานได้โดยไม่ต้องอุปกรณ์เสริมจากภายนอก

|                     | PIC18F452      |
|---------------------|----------------|
| Program Memory      | 16384x16       |
| RAM (Bytes)         | 1536           |
| EEPROM Data (Bytes) | 256            |
| Max Speed (MHz)     | 40             |
| I/O                 | 34             |
| Package             | 40-pin DIP     |
| A/D Channels        | 8 (10 Bit)     |
| PWM Channels        | 2              |
| Brown Out Reset     | Yes            |
| Serial I/O          | AUSART/I2C/SPI |

รูปที่ ก 2 แสดง Features Include



รูปที่ ก 3 แสดง Pin Outs



รูปที่ ก 4 จอแสดงผลแบบ LCD

LCD เป็นตัวแสดงผลที่ดีกว่าตัวแสดงผลแบบ 7 segment เนื่องจากสามารถแสดงตัวอักษร ข้อความได้ชัดเจนและยังสามารถรับภาระ ด้านการเลื่อน แทรก คงค้างข้อความและข้อมูลไว้ใน หน่วยความจำเฉพาะ รวมถึงสามารถอ่านกลับมาที่ CPU ได้ ขั้นตอนการทำงานอย่าง คือ เริ่มจาก การป้อนไฟแล้วรอการ reset อย่างน้อย 5 Msec จากนั้นเป็นการส่งสัญญาณควบคุมโดยให้ ขา สัญญาณ 0,0 และสุดท้ายเป็นการ ส่งค่าตำแหน่งและรหัสตัวอักษร ปัญหาเท่าที่พบส่วนใหญ่เกิดจาก

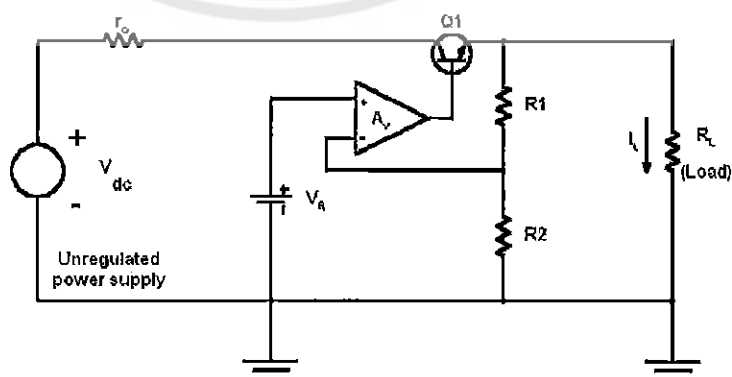
ขั้นตอนการ Initialize ไม่สมบูรณ์ ซึ่งอาจเกิดจากการ หน่วงเวลาแต่ละคำสั่ง การควบคุมขา E ซึ่งต้องเหมาะสมกับกลาง Plus ของขาข้อมูล (Data) อื่นๆ ส่วนการ Control LCD นั้น

สามารถควบคุมแบบปกติโดยใช้ขา Data 8 bit และขา Control อีก 4 bit หรือยังสามารถควบคุม data โดยใช้แบบ 4 bit แต่ต้องส่งคำสั่ง 2 ครั้ง นอกจากนี้ยังสามารถควบคุม Data โดยใช้ 1 bit ก็ได้แต่ต้องใช้ IC FlipFlop มาควบคุม



รูปที่ ก 5 Monolithic Regulator

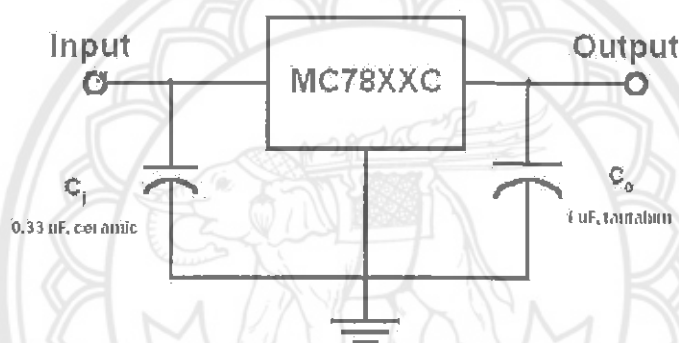
Regulated คือ วงจรอิเล็กทรอนิกส์ที่สร้างแรงดันกระแสตรงที่ โดยไม่ขึ้นอยู่กับกระแสที่จ่ายให้กับโหลด, ไม่ขึ้นกับอุณหภูมิ และไม่ขึ้นกับความแปรผันต่างๆในสายส่งกระแสกลับ



รูปที่ ก 6 แสดงรูปแบบการต่อวงจร

วงจรตามรูปที่ 6 สามารถใช้อุปกรณ์ต่างๆตามท้องตลาดได้ โดยอาจใช้ออปแอมป์ 741 หรือ LM301A เป็น DIFFERENTIAL AMPLIFIER และใช้ LM103, LM199 หรือ ZENER DIODE เป็น REFERENCE DIODE แทน BATTERY VR ด้วยเทคโนโลยีของ MICROELECTRONICS ซึ่งสามารถรวมทุกอย่างให้อยู่ใน IC ตัวเดียวได้ซึ่งจะให้ประสิทธิภาพที่ดีกว่า ในราคาที่ต่ำกว่า และมีความน่าเชื่อถือด้านคุณภาพ ซึ่งเรียกไอซี(IC) ที่ทำหน้าที่เป็นเรกูเลเตอร์ว่าเรกูเลเตอร์สำเร็จรูป (MONOLITHIC REGULATOR)

MONOLITHIC REGULATOR ซึ่งอยู่ในรูปของไอซี คือ MOTOROLA MC7800C SERIES ซึ่งมี 3 ขา และให้แรงดันบวกขนาดคงที่ โดยสามารถดูตัวอย่างการใช้งานได้จากด้านล่างนี้



รูปที่ 6 แสดง MOTOROLA MC7800C SERIES

## Transistor

ทรานซิสเตอร์ (TRANSISTOR) คือ สิ่งประดิษฐ์ทำจากสารกึ่งตัวนำมีสามขา (TREE LEADS) กระแสหรือแรงเคลื่อน เพียงเล็กน้อยที่ขาหนึ่งจะควบคุมกระแสที่มีปริมาณมากที่ไหลผ่านขาทั้งสองข้างได้ หมายความว่าทรานซิสเตอร์เป็นทั้งเครื่องขยาย (AMPLIFIER) และสวิตช์ ทรานซิสเตอร์ ทรานซิสเตอร์ชนิดสองรอยต่อเรียกด้วยตัวย่อว่า BJT (BIPOlar JUNCTION TRANSISTOR) ทรานซิสเตอร์ (BJT) ถูกนำไปใช้งานอย่างแพร่หลาย เช่น วงจรขยายในเครื่องรับวิทยุและเครื่องรับโทรทัศน์หรือนำไปใช้ในวงจรอิเล็กทรอนิกส์ที่ทำหน้าที่เป็นสวิตช์ (Switching) เช่น เปิด-ปิด รีเลย์ (Relay) เพื่อควบคุมอุปกรณ์ไฟฟ้าอื่น ๆ เป็นต้น

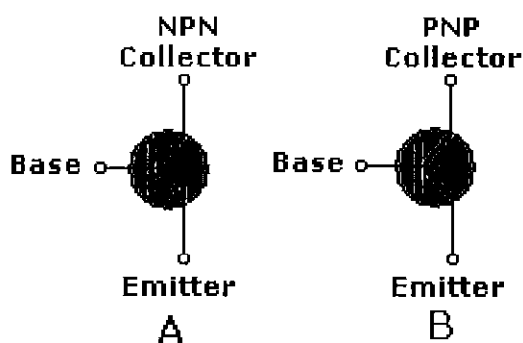
### โครงสร้างของทรานซิสเตอร์

ทรานซิสเตอร์ชนิดสองรอยต่อหรือ BJT นี้ ประกอบด้วยสารกึ่งตัวนำชนิดพีและเอ็นต่อกัน โดยการเติมสารเจือปน (Doping) จำนวน 3 ชั้นทำให้เกิดรอยต่อ (Junction) ขึ้นจำนวน 2 รอยต่อ การสร้างทรานซิสเตอร์จึงสร้างได้ 2 ชนิด คือ ชนิดที่มีสารชนิด N 2 ชั้น เรียกว่าชนิด NPN และชนิดที่มีสารชนิด P 2 ชั้น เรียกว่าชนิด PNP โครงสร้างของทรานซิสต์ชนิด NPN และชนิด PNP แสดงดังรูปด้านล่างนี้



รูปที่ 8 โครงสร้างของทรานซิสเตอร์ชนิด NPN และชนิด PNP

เมื่อพิจารณาจากรูปจะเห็นว่าโครงสร้างของทรานซิสเตอร์จะมีสารกึ่งตัวนำ 3 ชั้น แต่ละชั้นจะต่อลวดตัวนำจากเนื้อสารกึ่งตัวนำไปใช้งาน ชั้นที่เล็กที่สุด (บางที่สุด) เรียกว่า เบส (Base) ตัวอักษรย่อ B สำหรับสารกึ่งตัวนำชั้นที่เหลือคือ คอลเลกเตอร์ (collector หรือ c) และอิมิตเตอร์ (Emitter หรือ E) นั่นคือทรานซิสเตอร์ทั้งชนิด NPN จะมี 3 ขา คือ ขาเบส ขาคอลเลกเตอร์ ขาวงจรอิเล็กทรอนิกส์นิยมเขียนทรานซิสเตอร์แทนด้วยสัญลักษณ์ดังรูป



รูปที่ 9 แสดงสัญลักษณ์ที่ใช้แทนทรานซิสเตอร์

## Capacitor

ตัวเก็บประจุ หรือ คาปาซิเตอร์ (capacitor) เป็นอุปกรณ์อิเล็กทรอนิกส์อย่างหนึ่ง ทำหน้าที่เก็บพลังงานในสนามไฟฟ้า ที่สร้างขึ้นระหว่างคู่ขนาน โดยมีความจุไฟฟ้าเท่ากัน แต่มีชนิดของประจุตรงข้ามกัน บางครั้งเรียกตัวเก็บประจุชนิดนี้ว่า คอนเดนเซอร์ (condenser) เป็นอุปกรณ์พื้นฐานสำคัญในงานอิเล็กทรอนิกส์

### การเก็บประจุ

การเก็บประจุ คือ การเก็บอิเล็กตรอนไว้ที่แผ่นเพลตของตัวเก็บประจุ เมื่อนำแบตเตอรี่ต่อกับตัวเก็บประจุ อิเล็กตรอนจากขั้วลบของแบตเตอรี่ จะเข้าไปรวมกันที่แผ่นเพลต ทำให้เกิดประจุลบขึ้นและยังส่งสนามไฟฟ้าไป ผลักอิเล็กตรอนของแผ่นเพลตตรงข้าม ซึ่งโดยปกติในแผ่นเพลตจะมี ประจุเป็นบวกและลบ เมื่ออิเล็กตรอนจากแผ่นเพลต ถูกผลักให้หลุดออกจึงเหลือประจุบวกมากกว่าประจุลบ ยิ่งอิเล็กตรอนถูกผลักออกไปมากเท่าไร แผ่นเพลตนั้นก็จะเป็นบวกมากขึ้นเท่านั้น

### การคายประจุ

ตัวเก็บประจุที่ถูกประจุแล้ว ถ้าเรายังไม่นำขั้วตัวเก็บประจุมาต่อกัน อิเล็กตรอนก็ยังคงอยู่ที่แผ่นเพลต แต่ถ้ามีการครบวงจร ระหว่างแผ่นเพลตทั้งสองเมื่อไร อิเล็กตรอนก็จะวิ่งจากแผ่นเพลตทางด้านลบ ไปครบวงจรที่แผ่นเพลตบวกทันที เราเรียกว่า "การคายประจุ"

### ชนิดของตัวเก็บประจุ

ชนิดของตัวเก็บประจุแบ่งตามวัสดุการใช้งานแบ่งออกได้ 2 ชนิด คือ

1. ตัวเก็บประจุแบบปรับค่าได้ (Variable capacitor)
2. ตัวเก็บประจุชนิดคงที่ (Fixed capacitor)

### ตัวเก็บประจุชนิดคงที่ (Fixed capacitor)

Capacitor ชนิดนี้มีขั้วบวกและขั้วลบกำหนดไว้ ส่วนใหญ่จะเป็นแบบกลม ดังนั้น การนำไปใช้งานจะต้องคำนึงถึงการต่อขั้วให้กับ Capacitor ซึ่งสังเกตที่ขั้วของตัวเก็บประจุว่า ขั้วไหนที่เป็นขั้วลบจะมีลูกศรชี้ไปที่ขั้วนั้น และในลูกศรจะมีเครื่องหมายลบกำกับ

- ตัวเก็บประจุแบบกระดาษ (Paper capacitor) ตัวเก็บประจุแบบเปเปอร์ นำไปใช้งานซึ่งต้องการค่าความต้านทานของฉนวนที่มีค่าสูง และมีเสถียรภาพต่ออุณหภูมิสูงได้ดี มีค่าความจุที่ดีในช่วงอุณหภูมิที่กว้าง

- ตัวเก็บประจุแบบไมก้า (Mica capacitor)

ตัวเก็บประจุแบบไมก้าน มีเสถียรภาพต่ออุณหภูมิ และ ความถี่ดี มีค่าตัวประกอบการสูญเสียต่ำ และสามารถทำงาน ได้ดีที่ความถี่สูง นำมาใช้ในงานหลากหลาย เช่น ในวงจรออสซิลเลเตอร์ วงจรกรองสัญญาณ และวงจรขยายความถี่วิทยุกำลังสูง ซึ่งไม่มีการผลิตตัวเก็บประจุแบบไมก้าค่าความจุสูงๆ เนื่องจากเก็บประจุแบบไมก้ามีราคาแพง จะทำให้ค่าใช้จ่ายในการผลิตสูง

- ตัวเก็บประจุแบบเซรามิก (Ceramic capacitor)

ตัวเก็บประจุชนิดเซรามิก โดยทั่วไปตัวเก็บประจุชนิดนี้มีลักษณะกลมๆ แบนๆ บางครั้งอาจพบแบบสี่เหลี่ยมแบนๆ ส่วนใหญ่ตัวเก็บประจุชนิดนี้ มีค่าน้อยกว่า 1 ไมโครฟารัด และเป็นตัวเก็บประจุชนิดที่ไม่มีขั้ว และสามารถทนแรงดันได้ประมาณ 50-100 โวลต์ค่าความจุของตัวเก็บประจุชนิดเซรามิกที่มีใช้กันในปัจจุบันอยู่ในช่วง 1 พิโกฟารัด ถึง 0.1 ไมโครฟารัด

- ตัวเก็บประจุแบบอิเล็กโทรไลติก (Electrolytic capacitor)

ตัวเก็บประจุชนิดอิเล็กโทรไลติก ตัวเก็บประจุชนิดนี้ ต้องระวังในการนำไปใช้งาน เพราะมีขั้วที่แน่นอนพิมพ์ติดไว้ด้าน ข้างตัวถังอยู่แล้ว กรณีที่กำหนดค่าแรงดันให้กับตัวเก็บประจุผิดขั้ว อาจเกิดความเสียหายกับตัวเก็บประจุ และอุปกรณ์ที่ประกอบร่วมกับตัวเก็บประจุ ขั้วของตัวเก็บประจุชนิดนี้สังเกตได้จาก ขาที่ยาวจะเป็นขั้วบวก และขาที่สั้นจะเป็นขั้วลบ

- ตัวเก็บประจุแบบน้ำมัน (Oil capacitor)
- ตัวเก็บประจุแบบโพลีเอทิลีน (Polyethylene capacitor)
- ตัวเก็บประจุ แทนทาลัม (Tantalum capacitor)

ตัวเก็บประจุแบบแทนทาลัม จะให้ค่าความจุสูงในขณะที่ตัวถังที่บรรจุมีขนาดเล็ก และมีอายุในการเก็บรักษาค่อนข้างมาก ตัวเก็บประจุแบบแทนทาลัมนี้ มีหลายชนิดให้เลือกใช้ เช่น ชนิด โซลิด (solid type) ชนิด ซินเทอร์สแลก (sintered slug) ชนิดฟอยล์ธรรมดา (plain foil) ชนิดเอ็ชฟอยล์ (etched foil) ชนิดเว็ทสแลก (wet slug) และ ชนิดชิป (chip) การนำไปใช้งานต่างๆ ประกอบด้วยวงจรกรองความถี่ต่ำ วงจรส่งผ่านสัญญาณ ชนิด โซลิดนั้นไม่ไวต่ออุณหภูมิ และมีค่าคุณสมบัติระหว่างค่าความจุอุณหภูมิต่ำกว่า ตัวเก็บประจุ แบบอิเล็กทรอนิกส์ชนิดใด ๆ สำหรับงานที่ตัวเก็บประจุแบบแทนทาลัมไม่เหมาะสมกับ วงจรตั้งเวลาที่ใช้ RC ระบบกระตุ้น (triggering system) หรือ วงจรเลื่อนเฟส (phase - shift net work) เนื่องจากตัวเก็บประจุแบบนี้ มีค่าคุณสมบัติของการดูดกลืนของไดอิเล็กตริก สูง ซึ่งหมายถึงเมื่อตัวเก็บประจุถูกคายประจุ สารไดอิเล็กตริกยังคงมีประจุหลงเหลืออยู่ ดังนั้นแม้ว่าตัวเก็บประจุที่มีคุณสมบัติของ การดูดกลืนของสารไดอิเล็กตริกสูงจะถูกคายประจุ ประจุจนเป็นศูนย์แล้วก็ตาม จะยังคงมีประจุเหลืออยู่เป็นจำนวนมาก ซึ่งทำให้เกิดปัญหาในวงจรตั้งเวลา และ วงจรอื่นที่คล้ายกัน

- ตัวเก็บประจุแบบไมลา (Milar capacitor)
- ตัวเก็บประจุแบบไบ โพลาร์ (Bipolar capacitor)
- ตัวเก็บประจุแบบโพลีโพรไพลีน (Polypropylene)

#### ตัวเก็บประจุแบบปรับค่าได้ Variable capacitor

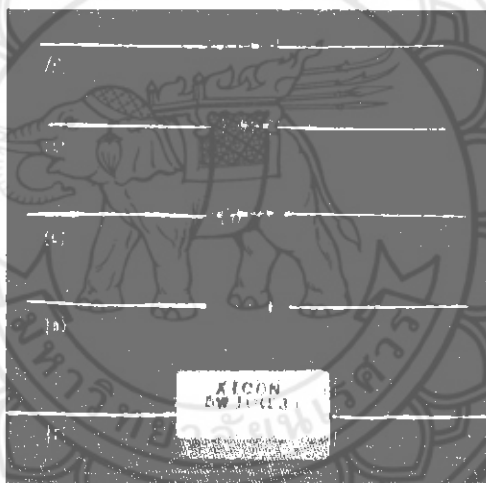
Capacitor ชนิดที่ไม่มีค่าคงที่ ซึ่งจะมีการนำวัสดุต่างๆ มาสร้างขึ้นเป็น Capacitor โดยทั่วไปจะมีค่าความจุไม่มากนัก โดยประมาณไม่เกิน 1 ไมโครฟารัด (m F) Capacitor ชนิดนี้เปลี่ยนค่าความจุได้ จึงพบเห็นอยู่ในเครื่องรับวิทยุต่าง ๆ ซึ่งเป็นตัวเลือกหาสถานีวิทยุโดยมีแกนหมุน Trimmer หรือ Padder เป็น Capacitor ชนิดปรับค่าได้ ซึ่งคล้าย ๆ กับ Variable Capacitor แต่จะมีขนาดเล็กกว่า การใช้ Capacitor แบบนี้ถ้าต่อในวงจรแบบอนุกรมกับวงจรเรียกว่า Padder Capacitor ถ้านำมาต่อขนานกับวงจร เรียกว่า Trimmer Resistor

#### ตัวต้านทานไฟฟ้า

คือ อุปกรณ์ที่นักวิทยาศาสตร์ประดิษฐ์ขึ้นมา เพื่อต่อร่วมกับวงจรเพื่อบังคับให้กระแสไฟฟ้าในวงจร เปลี่ยนแปลงตามต้องการ ทำจากวัสดุที่ปล่อยให้อิเล็กตรอนหลุดจาก ตัวต้านทานได้น้อย ตัวต้านทานไฟฟ้ามีหน่วยเป็น โอห์ม ( Ohm ) ซึ่งเป็นนามของ George Simon

Ohm ตัวต้านทานหรือ รีซิสเตอร์ มีการผลิตออกมาหลายแบบ หลักการต่อไปนี้จะได้นำเอาค่า ความต้านทานเป็นหลักสามารถ แบ่งออกได้ 3 แบบ คือ

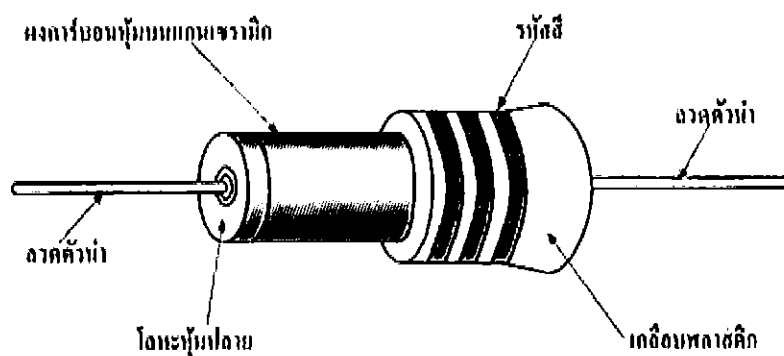
1. ตัวต้านทานชนิดค่าคงที่ ( Fixed Resistor ) เป็นตัวต้านทานที่มีความต้านทานคงที่ โดยจะกำหนดค่าความต้านทานเป็นรหัส เช่น ตัวเลข โค้ดสี จะพบเห็นได้ในวงจรทั่วไป



รูปที่ ก 10 ตัวต้านทานชนิดค่าคงที่ ( Fixed Resistor )

โครงสร้างของตัวต้านทานค่าคงที่

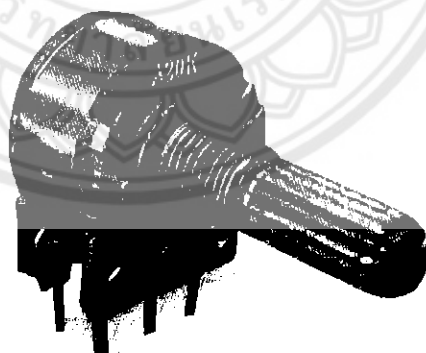
กรรมวิธีในการผลิตตัวต้านทานมีด้วยกันหลายวิธี ตามแต่ชนิดของตัวต้านทาน เช่นการใช้ลวดพันรอบโครงสร้าง ของตัวต้านทาน แล้วก็ต่อขาออกมาใช้งาน



รูปที่ ก 11 โครงสร้างของตัวต้านทานค่าคงที่

## 2. ตัวต้านทานปรับค่าได้ (Adjustable Resistor)

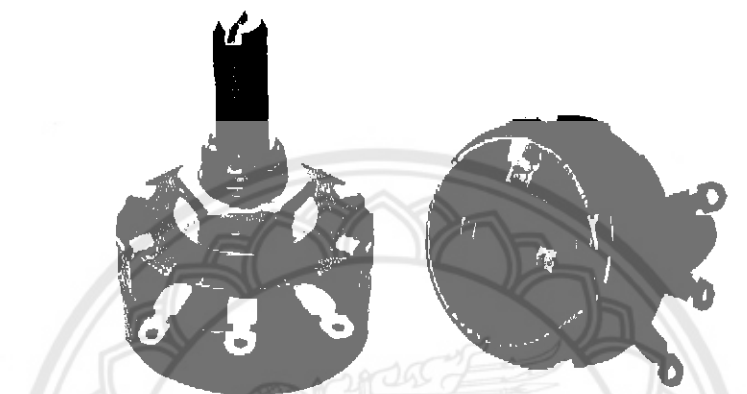
รีซิสเตอร์แบบ (Tap Resistor) เป็นตัวต้านทานที่ใช้กับงานที่มีกำลัง วัตต์สูงๆ และงานที่ต้องการเปลี่ยนแปลงค่าความต้านทานอยู่บ่อยๆ สามารถเลือกค่าได้ค่าหนึ่ง โดยปกติจะมี 1 ขั้ว หรือมากกว่านั้นแยกออกมาเพื่อเลือกนำไปใช้งาน เพื่อให้การทำงานเป็นไปตามวัตถุประสงค์



รูปที่ ก 12 ตัวต้านทานปรับค่าได้ (Adjustable Resistor)

ตัวต้านทานปรับค่าได้จะมีอยู่หลายประเภท เช่น แบบหมุน ซึ่งมีทั้งแบบธรรมดาที่หมุนได้รอบเดียวที่ใช้กันทั่วไป และแบบ พิเศษที่หมุนได้หลายรอบ ( Trimmer Potentiometer) และแบบ สไลด์

3. ตัวต้านทานชนิดเปลี่ยนแปลงค่าได้ (Variable Resistor) เป็นตัวต้านทานที่สามารถปรับค่าความต้านทานได้อย่างต่อเนื่อง ในช่วงค่าความต้านทานที่กำหนดไว้ จะใช้ในงานที่ต้องการปรับค่าความต้านทานบ่อยๆ ตัวต้านทานชนิดนี้ มีหน้าคอนแทกสำหรับการหมุนเลื่อนหน้าคอนแทก



รูปที่ ก 13 ตัวต้านทานชนิดเปลี่ยนแปลงค่าได้ (Variable Resistor)

การปรับค่าความต้านทานโดยมีแกนยื่นออกมาเพื่อใช้สำหรับหมุนปรับค่า อย่างเช่น วอลุ่มเรจเสียงแบบธรรมดา หรือ แบบสไลด์ อีกแบบหนึ่งไม่มีแกนหมุนเราเรียกกันว่า วอลุ่มเกือกม้า หรือ Trim pot วารีเอเบิลรีซิสเตอร์ อาจจะทำมาจาก คาร์บอน เซอร์เมท (เซรามิกผสมเงิน) หรือพลาสติก ดังนั้น ตัวต้านทานชนิดนี้จะแบ่งเป็น 3 ชนิด คือ

1. โปเทนชิโอมิเตอร์ (Potentiometer) ตัวต้านทานชนิดนี้ส่วนใหญ่ จะมีหน้าสัมผัสเป็นแบบถ่าน (carbon) ดังนั้นจะ ใช้กับงานที่มีกระแสเล็กน้อย เช่น เป็นตัวควบคุมระดับความดังของเสียง
2. รีโอสตัส (Rheostat) ตัวต้านทานชนิดนี้จะมีหน้าสัมผัส เป็นแบบลวดพัน ดังนั้น จึงใช้กับงานที่มีกระแสมากๆ เช่น ใช้ปรับกระแส หรือแรงดันในเครื่องแหล่งจ่ายไฟ
3. ชนิดพิเศษ (Special Resistor) เป็นตัวต้านทานที่มีลักษณะพิเศษสามารถแบ่งได้เป็น 2 ชนิด คือ เปลี่ยนตาม อุณหภูมิ (Thermistor) ตัวต้านทานชนิดนี้ค่าความต้านทานจะเปลี่ยนแปลงไป ส่วนใหญ่จะใช้สำหรับชดเชยการ เปลี่ยนแปลงต่ออุณหภูมิ ของวงจร หรือ ใช้เป็นตัวเปลี่ยนแปลงอุณหภูมิให้เป็นสัญญาณทางไฟฟ้า เปลี่ยนตามความสว่าง (Light Dependent Resistor –LDR) ตัวต้านทานชนิด

นี่เป็นตัวต้านทานที่ไวต่อแสง โดยค่าความต้านทานจะลดลง เมื่อความเข้มของแสงที่ตกกระทบบมีค่ามาก ส่วนใหญ่จะใช้เป็นตัวเปลี่ยนแปลงความเข้มของแสง ให้เป็นสัญญาณทางไฟฟ้า



รูปที่ ก 14 ชุดควบคุมความเร็วพัดลม

ชุดควบคุมความเร็วพัดลม โดยการสร้างระบบไว้ในบทที่ 3 ใช้Senser(SHT 15)ในการวัดอุณหภูมิและความชื้น นำข้อมูลไปประมวลผล แล้วไมโครชิพจะสั่งงานให้พัดลมหมุนตามที่ต้องการ โดยสามารถนำไปประยุกต์ใช้เพื่อควบคุมความชื้นโดยการปรับอัตราไหลของอากาศที่อีวาโปเรเตอร์



รูปที่ ก 15 อุปกรณ์เพิ่มความชื้น

อุปกรณ์เพิ่มความชื้นโดยใช้พัดลมในการควบคุมความชื้นที่ปล่อยออกมาเป็นอุปกรณ์ที่ประดิษฐ์ขึ้นมาเพื่อทดลองในห้องปรับอากาศ



รูปที่ ก 16 ภาพอุปกรณ์ด้านหน้า



รูปที่ ก 17 ภาพอุปกรณ์ด้านหลัง



ข้อแนะนำในการเขียนโปรแกรมและการออกแบบโครงงาน

### ข้อแนะนำในการเขียนโปรแกรม

- ผังงาน (Flow chart) เป็นสิ่งที่จำเป็นอย่างยิ่ง ในการเขียนโปรแกรมที่มีเงื่อนไข มากๆ ซึ่งจะช่วยให้ลำดับขั้นตอนได้สะดวกมากขึ้น
- การกำหนดชื่อของเลเบล (Label) ควรตั้งชื่อให้สื่อกับความหมายใกล้เคียงกับหน้าที่ ของตำแหน่งนั้นๆ เช่น เลเบลแสดงผลอาจจะใช้ชื่อ DISPLAY: เพิ่มค่าข้อมูล ใช้INC\_DATA:
- โปรแกรมย่อย (Subroutine) ควรจะระบุชื่อให้สื่อกับความหมาย โดยทำเป็นคอมเมนต์บน ส่วนของโปรแกรมย่อย เพื่อความสะดวกในการตรวจสอบภายหลัง เช่น

```
*****
*** โปรแกรมย่อยรับค่าคีย์จากสัญญาณ DTMF ***
*****
```



## การออกแบบโครงการ

การออกแบบหรือสร้างโครงการ เพื่อให้ได้ตามวัตถุประสงค์ และเงื่อนไข ที่กำหนดไว้ ซึ่งในขั้นตอนการออกแบบหรือสร้างโครงการนั้นสามารถที่จะสรุปเป็นลำดับขั้นตอนได้ดังนี้

- การกำหนดจุดประสงค์ของงานเป็นการกำหนดเป้าหมายของโครงการ เช่นทำเพื่อนำไปใช้งานกับอะไร
- การกำหนดขอบเขต และเงื่อนไขของโครงการ
- การคำนึงถึงอุปกรณ์ต่อพ่วง เช่น เครื่องคอมพิวเตอร์ PC พรินเตอร์ หรือการเชื่อมต่อกับระบบเครือข่ายข้อมูล
- การเลือกใช้วัสดุอุปกรณ์ คุณสมบัติของไอซีไมโครคอนโทรลเลอร์ที่เลือกใช้ จำนวนของพอร์ตอินพุต-เอาต์พุตที่ใช้งาน ความสะดวกในการออกแบบลายวงจรพิมพ์ อุปกรณ์มีสัดส่วนและขนาดที่มาตรฐาน
- การออกแบบวงจร และลายวงจรพิมพ์ อุปกรณ์ป้องกันที่เพิ่มในวงจร เช่นการเพิ่มตัวเก็บประจุที่ใช้สำหรับกรองความถี่สูงทั้งการพูลอัพให้มีสถานะที่แน่นอน การใช้ไอซีบัฟเฟอร์ ฯลฯ การออกแบบวงจรพิมพ์ให้รองรับกับอุปกรณ์ ที่ต่อพ่วง รวมทั้งความยืดหยุ่นกับขนาดของอุปกรณ์ประเภทเดียวกันที่นำมาใช้งาน เช่น อุปกรณ์ประเภท SMD, สวิตช์, ตัวเก็บประจุ ฯลฯ
- การประกอบวงจร และการตรวจสอบ ลำดับการประกอบวงจรไม่มีข้อกำหนดที่แน่นอน แต่ขึ้นอยู่กับความสะดวกในการบัดกรี และการตรวจสอบเมื่อมีปัญหา การออกแบบโครงการที่สร้างเป็นเครื่องต้นแบบ ควรประกอบตามลำดับวงจร เช่นภาคจ่ายไฟเป็นอันดับแรก และต้องทำงานได้ ก่อนประกอบวงจร ในส่วนถัดไปของลำดับงาน การประกอบโครงการ สามารถเริ่มวางอุปกรณ์ตัวที่มีความสูงน้อยที่สุดก่อน และตามด้วยตัวที่มีความสูงถัดไปเป็นลำดับ การตรวจสอบโครงการให้ทดสอบทีละลำดับโดยไล่จากวงจร และการทำงานของอุปกรณ์นั้นๆ เช่น รีเลย์, แอลซีดี ฯลฯ
- การนำไปใช้งาน (ประเมิน)ต้องนำมาทดสอบกับสถานะต่างๆ เช่นสัญญาณรบกวนจากการกระชากของมอเตอร์ขณะออกตัว หรือเป็นเหตุการณ์อื่นๆ ที่อยู่ในบริเวณที่จะต้องนำโครงการนั้นๆ ไปใช้งาน

- การแก้ไขข้อผิดพลาด (ปรับปรุง) โครงการที่ออกแบบไว้เป็นเครื่องต้นแบบนั้น ข้อผิดพลาดหรือการปรับปรุงเพิ่มเติม สามารถเกิดขึ้นได้ทุกครั้ง ดังนั้นการแก้ไขหรือปรับปรุงต้องทำด้วยความรอบคอบ เช่นการบัดกรีอุปกรณ์เพิ่มเติม การตัดลาย PCB การเดินสายไวร์เฟด ต้องมั่นคงแข็งแรงเพื่อลดปัญหา เมื่อนำไปทดลอง





\*\*\*\*\* Main Code\*\*\*\*\*

```
#include <18F452.h>
```

```
#fuses HS,NOWDT,PUT,NOPROTECT,NOLVP
```

```
#use delay(clock=20000000)
```

```
#include<lcd.c>
```

```
#include<SHT15.c>
```

```
void main()
```

```
{
```

```
float restemp, truehumid;
```

```
lcd_init();
```

```
sht_init();
```

```
while(TRUE)
```

```
{
```

```
sht_rd (restemp, truehumid);
```

```
lcd_gotoxy(1,1);
```

```
// setup frequency 5KHz
```

```
setup_ccp1(CCP_PWM);
```

```
setup_timer_2(T2_DIV_BY_4, 249, 1);
```

```
printf(lcd_putc, "RH : %3.2f%% ", truehumid);
```

```
if(truehumid>=50) // 50
```

```
{
```

```
output_low(PIN_B7);
```

```
output_low(PIN_C3);
```

```
}
```

```

else
{
    output_high(PIN_B7);
    output_high(PIN_C3);
}

if(truehumid>=50)
{
    set_pwm1_duty(0);
    printf(lcd_putc, "Duty Cycle : 0 %%");
}

else if(truehumid>=49.8)
{
    set_pwm1_duty(700); // startup fan 0.5S
    delay_ms(500);
    set_pwm1_duty(260);
    printf(lcd_putc, "Duty Cy : 26 %%");
}

else if(truehumid>=49.5)
{
    set_pwm1_duty(260);
    printf(lcd_putc, "Duty Cy : 26 %%");
}

else if(truehumid>=49)
{
    set_pwm1_duty(328);
    printf(lcd_putc, "Duty Cy : 32.8 %%");
}

else if(truehumid>=48.5)

```

```

{

    set_pwm1_duty(396);
    printf(lcd_putc, "Duty Cy : 39.6 %%");
}

else if(truehumid>=48)
{

    set_pwm1_duty(464);
    printf(lcd_putc, "Duty Cy : 46.4 %%");
}

else if(truehumid>=47.5)
{

    set_pwm1_duty(532);
    printf(lcd_putc, "Duty Cy : 53.2 %%");
}

else if(truehumid>=47)
{

    set_pwm1_duty(600);
    printf(lcd_putc, "Duty Cy : 60.0 %%");
}

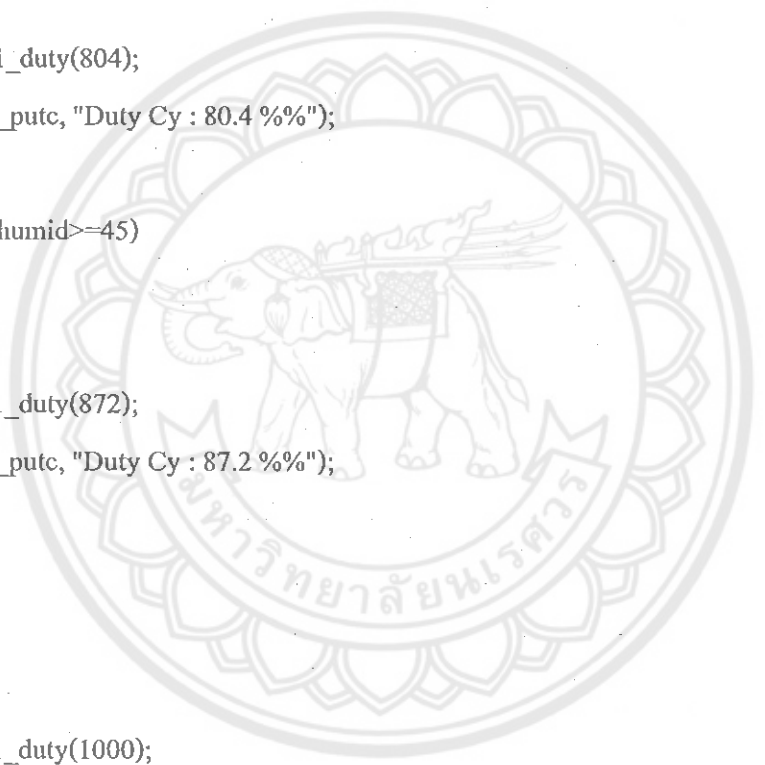
else if(truehumid>=46.5)
{

    set_pwm1_duty(668);
    printf(lcd_putc, "Duty Cy : 66.8 %%");
}

else if(truehumid>=46)

```

```
{  
  
    set_pwm1_duty(736);  
    printf(lcd_putc, "Duty Cy : 73.6 %%");  
}  
  
else if(truehumid>=45.5)  
{  
  
    set_pwm1_duty(804);  
    printf(lcd_putc, "Duty Cy : 80.4 %%");  
}  
  
else if(truehumid>=45)  
{  
  
    set_pwm1_duty(872);  
    printf(lcd_putc, "Duty Cy : 87.2 %%");  
}  
  
else  
{  
  
    set_pwm1_duty(1000);  
    printf(lcd_putc, "Duty Cy : 100 %%");  
}  
  
    delay_ms(500);  
}  
}
```



### Driver file for SHT15 Temperature & Humidity Sensor with PIC18F452

```
#define sht_data_pin  PIN_E0
```

```
#define sht_clk_pin  PIN_E2
```

```
***** Function to alert SHT15 *****
```

```
void comstart (void)
```

```
{
```

```
    output_float(sht_data_pin); //data high
```

```
    output_bit(sht_clk_pin, 0); //clk low
```

```
    delay_us(1);
```

```
    output_bit(sht_clk_pin, 1); //clk high
```

```
    delay_us(1);
```

```
    output_bit(sht_data_pin, 0); //data low
```

```
    delay_us(1);
```

```
    output_bit(sht_clk_pin, 0); //clk low
```

```
    delay_us(2);
```

```
    output_bit(sht_clk_pin, 1); //clk high
```

```
    delay_us(1);
```

```
    output_float(sht_data_pin); //data high
```

```
    delay_us(1);
```

```
    output_bit(sht_clk_pin, 0); //clk low
```

```
}
```

```
***** Function to reset SHT15 communication *****
```

```
void comreset (void)
```

```
{
```

```

int8 i;

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

//***** Function to write data to SHT15 *****
int1 comwrite (int8 iobyte)
{
    int8 i, mask = 0x80;
    int1 ack;

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

```

```

    output_bit(sht_clk_pin, 1);          //clk high
    delay_us(1);
    mask = mask >> 1;                    //shift to next bit
}

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

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

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

    //shift in MSB data
    for(i=0; i<8; i++)
    {
        iobyte = iobyte << 1;
        output_bit(sht_clk_pin, 1);          //clk high
        delay_us(1);

```

```

    if (input(sht_data_pin)) iobyte |= mask1; //shift in data bit
    else iobyte |= mask0;

    output_bit(sht_clk_pin, 0);          //clk low
    delay_us(1);
}

```

//send ack 0 bit

```

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

```

//shift in LSB data

```

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

    output_bit(sht_clk_pin, 0);          //clk low
    delay_us(1);
}

```

//send ack 1 bit

```

output_float(sht_data_pin); //data high
delay_us(1);

```

```

output_bit(sht_clk_pin, 1); //clk high
delay_us(2);
output_bit(sht_clk_pin, 0); //clk low

return(iobyte);
}

```

\*\*\*\*\* Function to wait for SHT15 reading \*\*\*\*\*

```

void comwait (void)
{
    int16 sht_delay;

    output_float(sht_data_pin); //data high
    output_bit(sht_clk_pin, 0); //clk low
    delay_us(1);
    for(sht_delay=0; sht_delay<30000; sht_delay++) // wait for max 300ms
    {
        if(!input(sht_data_pin)) break; //if sht_data_pin low, SHT15 ready
        delay_us(10);
    }
}

```

\*\*\*\*\* Function to soft reset SHT15 \*\*\*\*\*

```

void sht_soft_reset (void)
{
    comreset(); //SHT15 communication reset
}

```

```
comwrite(0x1e);    //send SHT75 reset command
delay_ms(15);      //pause 15 ms
}
```

\*\*\*\*\* Function to measure SHT15 temperature \*\*\*\*\*

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

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

\*\*\*\*\* Function to measure SHT15 RH \*\*\*\*\*

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

    comstart();        //alert SHT15
    ack = comwrite(0x05); //send measure RH command and read ack status
```

```

if(ack == 1) return;
comwait();      //wait for SHT15 measurement to complete
iobyte = comread(); //read SHT75 temp data
return(iobyte);
}

```

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

```

void calculate_data (int16 temp, int16 humid, float & tc, float & rhlin, float & rhtrue)
{
    float truchumid1, rh;

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

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

    //calculate True RH reading
    rhtrue = ((tc - 25.0) * (0.01 + (0.00008 * rh))) + rhlin-2.8;
}

```

\*\*\*\*\* Function to measure & calculate SHT15 temp & RH \*\*\*\*\*

```

void sht_rd (float & temp, float & truchumid)
{
    int16 restemp, reshucid;

    float realhumid;

    restemp = 0; truchumid = 0;
}

```

```
restemp = measuretemp(); //measure temp  
reshumid = measurehumid(); //measure RH  
calculate_data (restemp, reshumid, temp, realhumid, truehumid); //calculate temp & RH  
}
```

```
//***** Function to initialise SHT15 on power-up *****/
```

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



ภาคผนวก ง  
การตั้งค่าอุปกรณ์



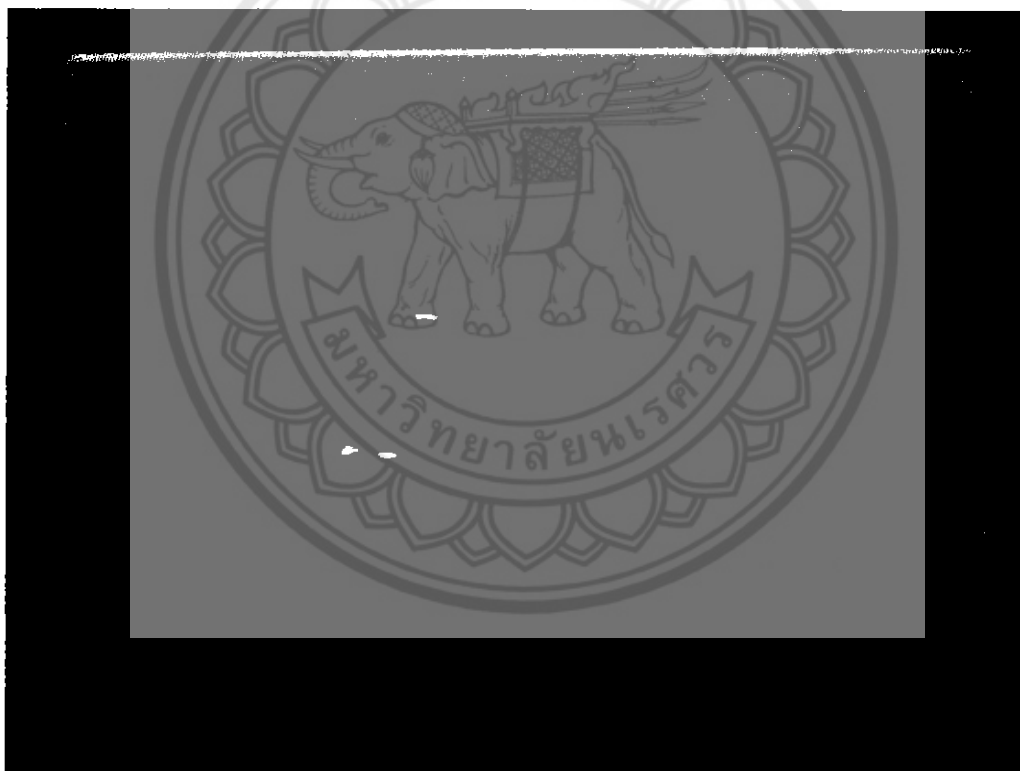
ในกรณีที่ต้องการจะเปลี่ยนแปลงเงื่อนไขการทำงานของอุปกรณ์ ซึ่งค่า default จะตั้งไว้ที่ 50 %RH สามารถทำได้ดังนี้

1. เปิดไฟล์ main.c ด้วยโปรแกรม CCS C compiler ( อ่านการใช้งานเพิ่มเติมที่หัวข้อ 3.4.1)
2. สังเกต code บรรทัดที่ 17 และ 25 ซึ่งจะเป็น

`if(truehumid>=50)//50 และ if(truehumid>=50)`

ให้เปลี่ยนตัวเลข 50 เป็นค่าที่ต้องการ จากนั้นก็ compile (ปุ่ม F9)และ save

3. ถอด IC ออกจากอุปกรณ์เพื่อนำไปทำการ flash ข้อมูลลงด้วยเครื่องโปรแกรมผ่าน MPLAB IDE v7.01 (อ่านการใช้งานได้จากหัวข้อ 3.4.2)
4. ใส่ IC เข้ากับอุปกรณ์ก็จะสามารถใช้งานได้ตามปกติ



รูปที่ 1 เครื่องโปรแกรมข้อมูลลง IC

## ประวัติผู้ทำโครงการ

ชื่อ - นามสกุล นาย ศาคร บิดา  
วัน - เดือน - ปีเกิด วันที่ 22 พฤศจิกายน พ.ศ.2529  
ประวัติการศึกษา พ.ศ.2541 จบการศึกษาระดับประถมศึกษา ณ ร.ร.ราชประชานุเคราะห์ 15  
พ.ศ.2544 จบการศึกษาระดับมัธยมศึกษาตอนต้น ณ ร.ร.ราชประชานุ  
เคราะห์ 15  
พ.ศ.2547 จบการศึกษาระดับมัธยมศึกษาตอนต้น ณ ร.ร.เชียงใหม่  
วิทยาคม

ชื่อ - นามสกุล นาย พยุง ยอดจันทร์  
วัน - เดือน - ปีเกิด วันที่ 26 มีนาคม พ.ศ.2530  
ประวัติการศึกษา พ.ศ.2541 จบการศึกษาระดับประถมศึกษา ณ ร.ร. บ้านด้าคะนอง  
พ.ศ.2544 จบการศึกษาระดับมัธยมศึกษาตอนต้น ณ ร.ร. สากเหล็กวิทยา  
พ.ศ.2547 จบการศึกษาระดับมัธยมศึกษาตอนปลาย ณ ร.ร. สากเหล็กวิทยา

ชื่อ - นามสกุล นายสิทธิศักดิ์ สวัสดิ์ผล  
วัน - เดือน - ปีเกิด วันที่ 5 กันยายน พ.ศ.2530  
ประวัติการศึกษา พ.ศ.2541 จบการศึกษาระดับประถมศึกษา ณ ร.ร.พัฒนาเวช  
พ.ศ.2544 จบการศึกษาระดับประถมศึกษา ณ ร.ร.บ้านฉางกาญจนกุล  
วิทยา  
พ.ศ.2547 จบการศึกษาระดับมัธยมศึกษาตอนปลาย ณ ร.ร.บ้านฉางกาญ  
จนกุลวิทยา