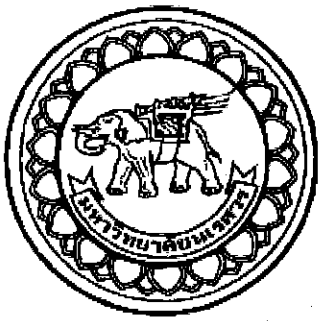


# เครื่องฟักไข่อัตโนมัติ

# AUTOMATIC INCUBATOR

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 29 S.A. 2547
เลขทะเบียน 4700185
เลขเรียกหนังสือ
มหาวิทยาลัยนเรศวร
TJ220.ก
2546

นายภัคพงษ์ หาญใจ รหัส 43360528
นายโชติพงษ์ เรืองฤทธิ์ รหัส 43360759

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2546

# ใบรับรองโครงงานวิศวกรรม

**หัวข้อโครงงาน** เครื่องฟักไข่อัตโนมัติ

**ผู้ดำเนินโครงงาน** นายภัคพงษ์ หาญใจ รหัส 43360528

นายโชติพงษ์ เรืองฤทธิ์ รหัส 43360759

**อาจารย์ที่ปรึกษา** อาจารย์ศิริพร เดชะศิลารักษ์

**สาขาวิชา** วิศวกรรมคอมพิวเตอร์

**ภาควิชา** วิศวกรรมไฟฟ้าและคอมพิวเตอร์

**ปีการศึกษา** 2546

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

( อาจารย์ศิริพร เดชะศิลารักษ์ )

.....กรรมการ

( ดร. ยงยุทธ ชนบดีเฉลิมรุ่ง )

.....กรรมการ

( อาจารย์ศุภวรรณ คำคงศักดิ์ )

| | | | | |
|---|---|---|---|---|
| **หัวข้อโครงงาน** | เครื่องฟักไข่อัตโนมัติ | | | |
| **ผู้ดำเนินโครงงาน** | นายภัคพงษ์ | หาญใจ | รหัส | 43360528 |
| | นายโชติพงษ์ | เรืองฤทธิ์ | รหัส | 43360759 |
| **อาจารย์ที่ปรึกษา** | อาจารย์ศิริพร | เดชะศิลารักษ์ | | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | | | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | | | |
| **ปีการศึกษา** | 2546 | | | |

.....

## บทคัดย่อ


โครงงานวิศวกรรมนี้ เป็นโครงงานที่ศึกษาและออกแบบต้นแบบของเครื่องฟักไข่ ที่สามารถควบคุมอุณหภูมิ โดยใช้เซนเซอร์วัดอุณหภูมิ DS1621 และใช้มอเตอร์ในการควบคุมการหมุนแผงไข่ ผ่านทางไมโครคอนโทรลเลอร์และไมโครคอมพิวเตอร์ ทั้งยังเก็บข้อมูลการทำงานของระบบไว้ในระบบฐานข้อมูล เพื่อการแสดงผลและนำไปใช้ในการศึกษาและพัฒนาระบบต่อไปได้

โครงงานนี้ได้ทำการทดลองการฟักไข่จำนวน 10 ฟองภายในเวลา 21 วัน โดยตั้งแต่วันที่ 1- 19 กำหนดอุณหภูมิไว้ที่ 37.5 องศาเซลเซียส และวันที่ 19-21 กำหนดอุณหภูมิ 36.5 องศาเซลเซียส และมีการพลิกแผงไข่ 8 ครั้งต่อวัน ซึ่งผลที่ได้ใกล้เคียงกับเครื่องที่เกษตรกรใช้ทั่วไปในปัจจุบัน เมื่ออยู่ในสภาวะแวดล้อมเดียวกัน

| | | |
|---|---|---|
| **Project Title** | Automatic Incubator | |
| **Name** | Mr.Pakapong Harnjai | ID.43360528 |
| | Mr.Chotepong Ruangrit | ID.43360759 |
| **Project Advisor** | Miss Siriporn Dachasilaruk | |
| **Major** | Computer Engineering | |
| **Department** | Electrical and Computer Engineering | |
| **Academic Year** | 2003 | |

.....


## ABSTRACT

This project is study and design automatic incubator that can control temperature by using a DS1621 IC as a temperature sensor and to control a motor by the microcontroller and the microcomputer. The microcomputer stores data on a database system for display and further study.

This project has been test incubator within 21 days, 1-19 has control temperature at 37.5 Celsius then 19-21 has control temperature at 36.5 Celsius and turn motor 8 times per days. Finally the result is similar to the incubator that farmer used.

# กิตติกรรมประกาศ

โครงงานวิศวกรรมคอมพิวเตอร์ สำเร็จได้ด้วยดีก็เนื่องด้วยความอนุเคราะห์จากท่านอาจารย์ที่ปรึกษาโครงงานคือ อ.ศิริพร เดชะศิลารักษ์ ซึ่งได้ให้ทั้งคำแนะนำในด้านต่างๆ ตั้งแต่ทางด้านทฤษฎี ด้านโปรแกรมมิ่ง ด้านการเขียนโครงงานวิจัย และอื่นๆ รวมทั้งขอขอบคุณพี่ๆที่คอยให้ความช่วยเหลือในทุกๆด้าน อีกทั้งกำลังใจที่ได้รับจากทางบ้านและเพื่อนๆ

ในโอกาสนี้ทางคณะผู้จัดทำโครงงานจึงขอขอบคุณทุกๆท่านที่มีส่วนช่วยทำให้โครงงานนี้ประสบความสำเร็จได้ด้วยดี

นายภัคพงษ์ หาญใจ
นายโชติพงษ์ เรืองฤทธิ์

# สารบัญ

หน้า


บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ซ

บทที่ 1 บทนำ ..... 1

1.1 หลักการและเหตุผล ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 1

1.3 ขอบข่ายของงาน ..... 2

1.4 กิจกรรมการดำเนินการ ..... 2

1.5 ผลที่คาดว่าจะได้รับ ..... 3

1.6 งบประมาณ ..... 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง ..... 4

2.1 การฟักไข่และตู้ฟักไข่ ..... 4

2.2 ไมโครคอนโทรลเลอร์ตระกูล 8051 ..... 12

2.3 เซนเซอร์วัดอุณหภูมิ DS18S20 ..... 25

2.4 มอเตอร์กระแสตรงและการควบคุม ..... 27

บทที่ 3 การออกแบบเครื่องฟักไข่ ..... 31

3.1 เครื่องฟักไข่ที่ใช้กันอยู่ในปัจจุบัน ..... 31

3.2 เครื่องฟักไข่ที่ทำการออกแบบ ..... 34

3.3 แนวคิดการเขียนโปรแกรม ..... 44

บทที่ 4 การทดลอง ..... 48

4.1 การทำงานในส่วนโปรแกรมติดต่อกับผู้ใช้ ..... 48

# สารบัญ (ต่อ)

หน้า

4.2 การทดลองเครื่องฟักไข่ ..... 51

บทที่ 5 บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการทดลอง..... 57

5.2 ข้อเสนอแนะและแนวทางแก้ไข..... 57

5.3 แนวทางการพัฒนาต่อไป.....58

เอกสารอ้างอิง .....59

ภาคผนวก ก .....61

ภาคผนวก ข.....63

ภาคผนวก ค..... 64

ภาคผนวก ง..... 67

ภาคผนวก จ.....69

ประวัติผู้เขียนโครงงาน ..... 79

# สารบัญตาราง


ตารางที่ หน้า

1.1 ขั้นตอนการดำเนินงานโครงงานเครื่องฟักไข่อัตโนมัติ ..... 2

2.1 แสดงผลของจำนวนครั้งในการกลับไข่ต่อวันในช่วงการฟัก ..... 10

2.2 แสดงผลของระยะเวลาที่ใช้ในการกลับไข่ในแต่ละช่วงของการฟัก ..... 11

2.3 ตารางแสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัทอินเทล .....13

2.4 โหมดการทำงานของพอร์ตอนุกรม .....17

2.5 รีจีสเตอร์ควบคุมการทำงานและบอกสถานะการสื่อสารข้อมูลอนุกรม SCON .....18

2.6 แฟล็กสถานะของรีจีสเตอร์ในการรับข้อมูล .....22

2.7 ตารางแสดงอัตราบอดมาตรฐานที่นิยมใช้กันในการสื่อสารอนุกรม โดยการกำหนดค่า .....22
ตัวแปรต่างๆสำหรับการเกิดอัตราบอดโดยใช้ Timer 1

4.1 การทดลองการพลิกไข่ด้วยมอเตอร์ตามองศาที่กำหนด..... 51

4.2 เปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและเทอร์โมมิเตอร์ชุดที่1..... 52

4.3 เปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและเทอร์โมมิเตอร์ชุดที่2.....53

4.4 การทดลองเพื่อเปรียบเทียบเครื่องฟักไข่ออกแบบใหม่กับเครื่องฟักไข่ที่มีอยู่ในปัจจุบัน.....56

# สารบัญรูป


รูปที่ หน้า

2.1 ตำแหน่งการวางไข่ฟัก ต้องทำมุม 45 องศา และกลับไข่หมุนสลับไปมา..... 9

2.2 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของ MCS-51..... 14

2.3 การกำหนดหน้าที่ขาสัญญาณของไอซี 8051..... 15

2.4(a) ข้อมูลสื่อสารแบบอนุกรม ข้อมูลหนึ่งไบต์จะถูกส่งออกคราวละบิตเป็นลำดับจนครบทั้งแปด..... 16

2.4(b) ข้อมูลสื่อสารแบบขนาน ข้อมูลแต่ละบิตภายในหนึ่งไบต์จะถูกส่งออกมาพร้อมกัน ..... 16

2.5 แผนภาพแสดงการทำงานของวงจรส่วนการรับและส่งข้อมูลอนุกรมของ 8051..... 17

2.6 แผนภาพเวลาของสัญญาณอนุกรมโหมดศูนย์..... 20

2.7 รูปแบบของสัญญาณอนุกรมในโหมด 1 ซึ่งมีลักษณะเดียวกับรูปแบบของการสื่อสารแบบอะซิงโครนัสที่ใช้ข้อมูล 8 บิต บิตเริ่มต้นและบิตสุดท้ายอย่างละ 1 บิต..... 21

2.8 รูปแบบของสัญญาณข้อมูลอนุกรมในโหมด 2 ซึ่งมีลักษณะรูปแบบการสื่อสารที่ใช้บิตข้อมูลจำนวน 9 บิต บิตเริ่มต้นและบิตสุดท้ายอย่างละ 1 บิต..... 23

2.9 แสดงรายละเอียดของไอซีวัดอุณหภูมิ DS1621..... 26

2.10 ไดอะแกรมแสดงระบบควบคุมมอเตอร์พื้นฐาน ..... 27

2.11 สัญญาณควบคุมมอเตอร์ด้วยวิธี PFM..... 28

2.12 สัญญาณควบคุมมอเตอร์แบบ PWM ..... 28

3.1 รูระบายอากาศในเครื่องฟักไข่ในปัจจุบัน..... 32

3.2 คันโยกปรับระดับในเครื่องฟักไข่ในปัจจุบัน ..... 32

3.3 ลักษณะภายในเครื่องฟักไข่ในปัจจุบัน ..... 33

3.4 เทอร์โมมิเตอร์ในเครื่องฟักไข่ในปัจจุบัน ..... 33

3.5 แผนผังระบบควบคุมภายในเครื่องฟักไข่ ..... 34

3.6 ด้านหน้าของเครื่องฟักไข่ที่ทำการออกแบบ ..... 34

3.7 ด้านข้องของเครื่องฟักไข่ที่ทำการออกแบบ..... 35

3.8 ลักษณะภายในของเครื่องฟักไข่ที่ทำการออกแบบ ..... 35

3.9(a) โครงสร้างของฐานข้อมูล..... 36

3.9(b) โครงสร้างของฐานข้อมูล..... 36

3.10 หน้าจอหลักของโปรแกรม ..... 37

3.11 หน้าจอเซ็ทพอร์ต ..... 37

# สารบัญรูป(ต่อ)


รูปที่ หน้า

3.12 หน้าจอแสดงการเตือน(Alarm) .....38

3.13 หน้าจอกำหนดสูตรสำหรับการฟักไข่ ..... 39

3.14 หน้าจอกำหนดสูตรสำหรับการฟักไข่ ..... 39

3.15 หน้าจอกำหนดสูตรสำหรับการฟักไข่ ..... 40

3.16 หน้าจอกำหนดสูตรสำหรับการฟักไข่ ..... 40

3.17 แผนผังระบบควบคุมมอเตอร์ภายในเครื่องฟักไข่ .....41

3.18 มอเตอร์และคอนแท็คสวิตซ์.....41

3.19 แผนผังระบบควบคุมอุณหภูมิภายในเครื่องฟักไข่ .....42

4.1 หน้าต่างหลักของโปรแกรมในส่วนติดต่อกับผู้ใช้.....48

4.2 หน้าต่างแสดงตารางสูตรต่างๆ.....48

4.3 แสดงหน้าต่างเพิ่มสูตร.....49

4.4 แสดงหน้าต่างแก้ไขสูตร..... 49

4.5 แสดงหน้าต่างแสดงตารางสัญญาณเตือน.....50

4.6 แสดงหน้าต่างกำหนดค่า Com Port.....50

4.7 กราฟแสดงการเปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์ ชุดที่1.....54

4.8 กราฟแสดงการเปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์ ชุดที่2.....54

ก-1 วงจรไดรเวอร์มอเตอร์และอินเตอร์เฟส..... 61

ก-2 วงจรไดรเวอร์มอเตอร์และอินเตอร์เฟส..... 62

ข-1 วงจร 3A Solid State Relay..... 63

ค-1 วงจรของบอร์ดวัดอุณหภูมิผ่านระบบบัส $I^2C$.....64

ค-2 ขาของ DS1621 ไอซีวัดอุณหภูมิแบบดิจิตอล.....65

ง-1 วงจรบอร์ด ET-RTC V4.0.....67

# บทที่ 1
# บทนำ

## 1.1 หลักการและเหตุผล

ในปัจจุบันอุตสาหกรรมทางด้านการเกษตรได้เจริญเติบโตขึ้นอย่างรวดเร็วเป็นผลให้ระบบการผลิตแบบดั้งเดิมนั้นเกิดความล่าช้า และด้อยประสิทธิภาพในการผลิต การแข่งขันทางด้านเทคโนโลยีเพื่อการผลิตได้เจริญเติบโตอย่างรวดเร็ว การนำเข้าเทคโนโลยีจากต่างประเทศจึงมีมากขึ้นอันเป็นผลทำให้เกิดการเสียดุลทางการค้า เพราะฉะนั้นเราควรที่จะพัฒนาเทคโนโลยีทางการเกษตรของเราเอง เพื่อส่งเสริมการผลิตและการส่งออกด้วย

ในด้านการผลิตทางการเกษตรนั้น อุตสาหกรรมการเลี้ยงไก่นับเป็นอุตสาหกรรมที่ทำรายได้ให้กับเกษตรกรเป็นอย่างมาก การมีเทคโนโลยีที่ดีจะช่วยให้เกษตรกรลดภาระต้นทุนการผลิต ต้นทุนทางด้านแรงงานและสามารถเพิ่มจำนวนการผลิตได้อย่างรวดเร็ว ซึ่งเป็นผลทำให้กิจการมีผลตอบแทนเพิ่มขึ้น

การฟักไข่นับเป็นกระบวนการอย่างหนึ่งซึ่งก่อให้เกิดความความยุ่งยากกับเกษตรกรเป็นอย่างยิ่งในการผลิตซึ่งจะต้องมีการควบคุมปัจจัยหลายหลายด้านเพื่อให้การฟักไข่สำเร็จลุล่วงวัตถุประสงค์จึงมีการนำเครื่องฟักไข่เข้ามาช่วยควบคุมการฟักไข่โดยการฟักไข่โดยเครื่องฟักไข่เป็นการพัฒนาวิธีการฟักไข่ จากวิธีการธรรมชาติสัตว์ปีกหลายชนิดมีอายุการฟักไข่ที่แตกต่างกันออกไป รวมถึงอุณหภูมิ และปัจจัยอื่นๆ ในการฟัก แม้แต่สัตว์ปีกชนิดเดียวกัน แตกต่างกันที่ พันธุ์เบา พันธ์หนักอายุการฟักก็แตกต่างออกไปด้วยเช่นกัน การเจริญเติบโตของตัวอ่อนในฟองไข่ในขณะฟักจนกระทั่งฟักออกมาเป็นตัวนั้นจำเป็นต้องอาศัยปัจจัยต่าง ๆ ที่เกี่ยวข้องและเหมาะสมจึงมีผลทำให้ประสิทธิภาพและอัตราการฟักออกได้ดี ตลอดจนคุณภาพลูกไก่ด้วย

## 1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาการประยุกต์ใช้ระบบวิศวกรรมควบคุมไมโครคอนโทรลเลอร์
2. เพื่อพัฒนาซอฟต์แวร์เพื่อการควบคุมการทำงานของไมโครคอนโทรลเลอร์และอุปกรณ์อิเล็กทรอนิกส์โดยใช้โปรแกรมภาษาซี
3. เพื่อพัฒนาซอฟต์แวร์เพื่อการติดต่อระหว่างชุดไมโครคอนโทลเลอร์กับฐานข้อมูล (Database)โดยใช้โปรแกรม Visual Basic 6.0
4. เพื่ออำนวยความสะดวกแก่เกษตรกรในการฟักไข่

## 1.3 ขอบข่ายของงาน

1. พัฒนาซอฟต์แวร์เพื่อการควบคุมสำหรับการฟักไข่ซึ่ง โปรแกรมจะทำงานบนไมโครคอนโทรลเลอร์ที่จะควบคุมการทำงานของอุปกรณ์สำหรับการฟักไข่

2. โปรแกรมที่ใช้พัฒนาคือ Turbo C , Visual Basic

3. ในการวิจัยครั้งนี้จะถือเอาความสะดวกสบายของเกษตรกรและผลของการเก็บข้อมูลเกี่ยวกับการฟักไข่เป็นส่วนสำคัญมากกว่าประสิทธิภาพในการฟักไข่

4. ในการวิจัยครั้งนี้จะทำการวิจัยในส่วนของการฟักไข่ด้วยตู้ฟักไข่เท่านั้น ไม่รวมไปถึงการคัดไข่ การดูแลไข่ก่อนฟักและหลังฟัก

## 1.4 กิจกรรมการดำเนินการ

ตารางที่ 1.1 ขั้นตอนการดำเนินงานโครงงานเครื่องฟักไข่อัตโนมัติ

| กิจกรรม | ปี 2545 | | ปี 2546 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี. ค. | เม. ย. | พ.ค. | มิ. ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. |
| 1. ศึกษาและรวบรวมข้อมูลที่เกี่ยวข้องกับการฟักไข่ รวมทั้งปัญหาของตู้ฟักไข่ที่มีอยู่ในปัจจุบัน | | | | | | | | | | | | |
| 2. ออกแบบสร้างตู้ฟักไข่ และระบบฐานข้อมูลที่เกี่ยวข้อง | | | | | | | | | | | | |
| 3. สร้างตู้ฟักไข่และระบบฐานข้อมูล | | | | | | | | | | | | |
| 4. ทดลองการฟักไข่ และบันทึกผล | | | | | | | | | | | | |

**ตารางที่ 1.1(ต่อ)** ขั้นตอนการดำเนินงานโครงงานเครื่องฟักไข่อัตโนมัติ

| กิจกรรม | ปี 2545 | | ปี 2546 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี. ค. | เม. ย. | พ.ค. | มิ. ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. |
| 5. ทำการแก้ไขและปรับปรุง | | | | | | | | | | | | |
| 6. จัดทำรายงาน | | | | | | | | | | | | |

## 1.5 ผลที่คาดว่าจะได้รับ

1. ความรู้ทางด้านการออกแบบและพัฒนาระบบควบคุมอัตโนมัติ
2. ลดค่าใช้จ่ายและเพิ่มความสะดวกสบายของเกษตรกรในการเลี้ยงไก่
3. สามารถผลิตตู้ฟักไข่อัตโนมัติได้

## 1.6 งบประมาณ

| | | |
|---|---|---|
| 1. ตู้ฟักไข่ | 1000 | บาท |
| 2. ชุด Microcontroller | 500 | บาท |
| 3. วงจรชุดเซนเซอร์ DS1621 | 500 | บาท |
| รวมเป็นเงินทั้งสิ้น | 2000 | บาท |

# บทที่ 2
# หลักการและทฤษฎีที่เกี่ยวข้อง

## 2.1 การฟักไข่และตู้ฟักไข่ [1]

### 2.1.1 ประสิทธิภาพในการฟักไข่

ในการฟักไข่จะให้ได้ผลนั้นมีปัจจัยที่จะต้องนำมาพิจารณาร่วมกันอยู่หลายปัจจัยด้วยกัน การประเมินผลของการฟักไข่ หรือประสิทธิภาพของการฟักไข่จะต้องพิจารณาสิ่งต่าง ๆ ต่อไปนี้ประกอบคือ

1. อายุของพ่อแม่ไก่ พ่อแม่ที่มีอายุมากจะทำให้อัตราการผสมติดและการฟักออกต่ำกว่าไก่ที่มีอายุน้อยอัตราการผสมติดและการฟักออกจะเพิ่มขึ้นเรื่อย ๆ นับแต่ไข่ไก่ฟองแรกไปสูงสุดเมื่อไก่ไข่ไปได้ 14-16 สัปดาห์ จากนั้นจะค่อย ๆ ลดลงในทางปฏิบัติไข่ไก่ระยะ 1-3 สัปดาห์แรก จะไม่นำไปฟักเพราะไข่ยังฟองเล็กเกินไป แต่จะเก็บไข่ฟักหลังจาก 3 สัปดาห์ ไปแล้ว และเช่นกันไก่ที่ไข่ครบปีแล้วจะไม่เก็บไข่เข้าฟัก เพราะอัตราการผสมติดจะฟักออกต่ำ ดังนั้นจะเก็บเฉพาะช่วงสัปดาห์ที่ 4 ถึง สัปดาห์ที่ 52

2. อาหารที่ใช้เลี้ยงไก่ การใช้อาหารไก่ไข่ที่ใช้ในการผลิตไข่บริโภคมาเลี้ยงไก่แม่พันธุ์ แล้วจะทำให้การฟักออกต่ำ จึงจำเป็นที่จะต้องเพิ่มไวตามิน เอ, ดี, อี และบีให้มากขึ้นและอาหารจะต้องใหม่สดอยู่เสมอเพราะอาหารเก่าเก็บไว้นานไวตามินจะเสื่อมสลายทำให้ไข่ขาดไวตามินที่เกี่ยวข้องกับการผสมติดและฟักออกสำหรับอาหารไก่พ่อพันธุ์จะแตกต่างกับอาหารแม่พันธุ์ตรงที่มีธาตุแคลเซี่ยมและฟอสฟอรัสที่ต่ำกว่า

3. การเก็บและคัดไข่ ควรเก็บเฉพาะไข่ขนาด 50-65 กรัม/ฟอง ใหญ่หรือเล็กกว่านี้คัดออกพร้อมนี้ได้คัดไข่บุบ ร้าวผิวเปลือกบาง ขรุขระ และรูปร่างผิดปกติออกหลังจากคัดไข่แล้วจะต้องรมควันฆ่าเชื้อโรค ก่อนที่จะนำไปเข้าฟักจะต้องนำไข่ออกผึ่งไว้ในอุณหภูมิห้องอย่างน้อย 12 ชั่วโมง หรือผึ่งอากาศนอกห้องเย็นไว้หนึ่งคืนก่อนจึงนำเข้าตู้ฟัก

4. ตู้ฟักไข่และอุปกรณ์ที่เกี่ยวข้อง โดยหลักการและวิธีการแล้วทุกตู้จะต้องมีพัดลม ทำหน้าที่กระจายความร้อนในตู้ให้สม่ำเสมอพร้อมกันนี้จะทำหน้าที่ดูดอากาศดีเข้าไปในตู้ และ อีกด้านหนึ่งจะเป่าอากาศเสียออกจากตู้การตั้งพัดลมจะต้องอยู่ในตำแหน่งที่ไม่เป่าลมไปกระทบไข่โดยตรง ด้านหน้าพัดลมจะมีลวดร้อนไฟฟ้าให้ความร้อนแก่ตู้ฟัก พัดลมจะพัดผ่านความร้อนแล้วนำความร้อนไปกระทบผนังตู้ก่อนแล้วจึงกระจายไปบนไข่ไก่ด้วยแรงสะท้อนจุดตรงที่ลมกระทบผนังนี้จะเจาะรูไว้สำหรับให้อากาศออกส่วนด้านตรงข้ามของรูออกจะเป็นตำแหน่งเจาะรูสำหรับอากาศดีเข้าตู้ในกรณีที่ไฟฟ้าดับและพัดลมไหม้จะทำให้อากาศภายในตู้ร้อนจัด และไม่มีอากาศหมุนเวียน ลูกไก่จะตายหมดดังนั้นในด้านปฏิบัติจึงต้องเปิดฝาตู้ฟักไข่ไว้จนกว่าไฟฟ้าจะมาหรือซ่อมพัดลมเสร็จ การเปิดฝา

ตู้ฟักขึ้นอยู่กับอายุของไข่ในตู้ถ้าหากไข่อายุน้อยเปิดเพียงแง้มตู้ไว้เป็นพอแต่ถ้าไข่อายุมาก จะต้องเปิดกว้างขึ้นเพื่อป้องกันไม่ให้อากาศภายในร้อนจัด ซึ่งเป็นความร้อนที่เกิดจากการหายใจของลูกไก่ จุดวิกฤตที่จะต้องเอาใจใส่อย่างยิ่งคือช่วงสุดท้ายของการฟักไข่คือระหว่าง 18-21 วัน ถ้าหากไฟฟ้าดับพัดลมไม่เดินลูกไก่จะตายภายใน 10-20 นาที เพราะลูกไก่ขาดอากาศ จึงต้องคอยระวังอย่างใกล้ชิดและเปิดฝาตู้ทันทีที่ไฟดับ ลวดร้อนไฟฟ้าเป็นแหล่งให้ความร้อนแก่ตู้ฟักจะวางอยู่หน้าพัดลมหรือใกล้ ๆ พัดลม ลวดร้อนมีขนาดตั้งแต่ 100 วัตต์ ถึง 1,500 วัตต์ ขึ้นอยู่กับขนาดของตู้ ตู้บรรจุ 10,000 ฟอง ใช้ลวดร้อนประมาณ 750-1,000 วัตต์ ในตู้หนึ่ง ๆ อาจจะวางลวดร้อนไฟฟ้าไว้หลายแห่งตามจำนวนพัดลมที่ใช้

การควบคุมอุณหภูมิของตู้ฟักไข่ให้อยู่ระดับที่ต้องการและรักษาระดับให้สม่ำเสมอเหมาะกับความต้องการฟักไข่นั้น ในปัจจุบันมีอยู่ คือ ระบบที่หนึ่งเป็นระบบที่ควบคุมด้วยเทอร์โมมิเตอร์ที่มีคุณสมบัติวัดอุณหภูมิภายในตู้ และทำหน้าที่เป็นตัวตัดไฟเข้าลวดร้อนไฟฟ้าผ่านการทำงานของระบบควบคุมด้วยเทอร์โมมิเตอร์ มีอุปกรณ์ที่เกี่ยวข้อง 4 ชิ้น คือ เทอร์โมมิเตอร์แผงอิเล็กทรอนิคส์ Selenoi และลวดร้อนไฟฟ้าเทอร์โมมิเตอร์จะถูกประดิษฐ์ขึ้นเพื่อควบคุมอุณหภูมิ 99 องศาฟาเรนไฮต์ หรือ 100 องศาฟาเรนไฮต์ หรือ 98.8 องศาฟาเรนไฮต์ หรือ 86 องศาฟาเรนไฮต์ เลือกได้ตามต้องการ เทอร์โมมิเตอร์นี้ราคาแพงประมาณอันละ 2,400-3,200 บาท และเป็นวัตถุทำด้วยแก้วบางแตกได้ง่ายถ้าไม่ระวังเทอร์โมมิเตอร์มีปลอกตะกั่วอยู่ 2 แห่งภายในปลอกตะกั่วจะมีลวดแพลตินัมแข็งเชื่อมระหว่างปรอทภายในเทอร์โมิเตอร์กับตะกั่วรอบนอก ทำหน้าที่เป็นสะพานไฟฟ้า Selenoi (ซีลีนอย)ทำหน้าที่เป็น สวิตซ์ปิดเปิดไฟแรงสูงไปยังลวดร้อนไฟฟ้าสวิตซ์มีความทนทานต่อความร้อนที่เกิดจากการไฟฟ้ากระโดดจากขั้วหนึ่งไปขั้วตรงกันข้ามอันเนื่องมาจากลวดร้อนกินไฟมาก การปิดหรือเปิดสวิชของ Selenoi จะสั่งการโดยเทอร์โมมิเตอร์อีกขึ้นหนึ่งตัว Selenoi ส่วนมากจะใช้ไฟกระแสตรง DC 24 V แผงอิเล็กทรอนิคส์ เป็นส่วนประกอบที่ช่วยลดกระแสไฟฟ้าที่ไหลผ่านเทอร์โมมิเตอร์ไม่ให้กระโดด อันเป็นสาเหตุที่ทำให้แถบปรอทบาง ๆ ขาดหรือไหม้ใช้การไม่ได้ การที่เครื่องฟักไข่ไม่ทำงานตามปกติเกือบทั้งหมดไม่ว่าจะเป็นความร้อนและความชื้น สาเหตุเนื่องมาจากแผงอิเล็กทรอนิคส์ไหม้การจากเทอร์โมมิเตอร์ไปยัง Selenoi ถูกตัดขาดด้วยแผงอิเล็คทรอนิคส์ จึงทำให้เครื่องฟักไข่ไม่ทำงานแผงอิเล็กทรอนิคส์นี้อยู่ใกล้ ๆ กับ Selenoi เป็นแผ่นบาง ๆ รูปสี่เหลี่ยมขนาดประมาณ 4 x 5 ที่มี Transistor และ Resister เป็นส่วนประกอบ ถ้าหากแผงนี้ไม่ทำงานและไหม้จำเป็นจะต้องเปลี่ยนใหม่ โดยปกติแล้วแผงนี้มีอายุการใช้งานได้นานหลายปีถ้าหากไม่มีเหตุไฟฟ้าลัดวงจรเกิดขึ้น บางครั้งแผงไม่ทำงานอาจเนื่องมาจากขั้วเสียบไฟฟ้าของแผงหลวมไฟฟ้าเดินไม่สะดวกก็เป็นได้ ดังนั้นก็ทดลองขยับและเสียบให้แน่นก็ช่วยแก้ปัญหาได้ ลวดร้อนไฟฟ้าเป็นแหล่งกำเนิดความร้อนในตู้ฟักไข่ใช้ไฟ 220 V กินไฟฟ้าตั้งแต่ 100-1500 วัตต์ มีรูปร่างต่างกันตามแต่ผู้ผลิตตู้ฟักกำหนดส่วนใหญ่แล้วมี 2 แบบคือแบบเส้นลวดขดเป็นวงกลม เช่น ลวดร้อนของเตาไฟฟ้าที่ใช้หุงต้มกันในบ้าน ลวดร้อนนี้จะถูกยืดออกให้ยาวเพื่อไม่ให้เกิดความร้อนจัดจน

ออกสีแดง ชนิดนี้มีจุดอ่อนคือถ้าหากมือของเราจับพลาดไปถูกเส้นลวดเข้าจะถูกไฟช๊อต และบางครั้งลวดร้อนจะขาด เนื่องจากใช้งานนานหรือไฟฟ้าลัดวงจร ลวดร้อนแบบที่สองเป็นขดลวดรูปตัวยู ภายในไส้กลางจะบรรจุลวดร้อนไฟฟ้ารอบ ๆ ลวดร้อนอัดด้วยสารประเภทซิลิกอนไม่เป็นฉนวนไฟฟ้า จึงทำให้ไฟฟ้าไม่ช๊อตเมื่อมือของเราสัมผัสลวดร้อนชนิดนี้ใช้งานได้ทนทานและไม่ค่อยจะขาดเนื่องมาจากไฟไม่ลัดวงจรลวดร้อนทั้งสองแบบทำงานโดยการควบคุมของ Selenoi จะเป็นตัวปิดหรือเปิดกระแสไฟฟ้าให้ผ่านลวดร้อน

การควบคุมอุณหภูมิและความชื้นของตู้ฟักไข่ด้วยเทอร์โมมิเตอร์มีจุดที่จะต้องตรวจสอบและวิเคราะห์ เพื่อแก้ไขในกรณีที่เกิดมีปัญหาการทำงานของตู้ผิดปกติคือ

1. ตรวจสอบตัวเทอร์โมมิเตอร์ อาจจะเสียบไม่สนิทกับขาเสียบทำให้ไฟเดินไม่สะดวก หรือขาเสียบเป็นสนิม หรือไม่ก็ปรอทขาดเป็นท่อน ๆ ทำให้ไฟฟ้าผ่านไม่ได้ ปรอทมีคุณสมบัติเป็นโลหะที่มีสภาพเป็นของเหลว ดังนั้นถ้าหากมันขาดไฟก็เดินผ่านไม่ต่อเนื่องวิธีการแก้ไขคือ นำเทอร์โมมิเตอร์ไปแช่ในตู้เย็นช่องน้ำแข็ง ให้ปรอทหดตัวและต่อกันใหม่อีกอย่างปรอทอาจจะไหม้ทำให้ปรอทขาดและสั้นกว่าเก่า จะทำให้ อุณหภูมิของตู้ฟักสูงกว่าที่กำหนดสาเหตุเนื่องมาจาก แผงอิเล็กทรอนิคส์เสียไม่ทำงานหรือการต่อกระแสไฟฟ้าผ่านเทอร์โมมิเตอร์โดยตรงสำหรับเทอร์โมมิเตอร์ที่ใช้ควบคุมความชื้นจะต้องทำความสะอาดผ้าฝ้ายหุ้มกระเปาะทุก ๆ สัปดาห์ซักฟอกเอาหินปูนออก ทำให้การดูดซึมและระเหยของน้ำถูกต้องยิ่งขึ้น

2. แผงอิเล็กทรอนิคส์โดยปกติไม่มีปัญหาอาจจะเสียบไม่แน่นหรือแผงเคลื่อนที่ในกรณีแผงไหม้ก็ต้องให้ช่างวิทยุเป็นคนตรวจสอบให้หรือส่งไปที่กลุ่มงานสัตว์ปีก ในขณะที่ส่งไปนั้นให้ใช้แผงสำรองทำงานแทนถ้าแผงผิดปกติส่วนใหญ่แล้วอุปกรณ์ที่เกี่ยวกับความร้อนและความชื้นจะไม่ทำงานเลย

**2.1.2 ปัจจัยที่สำคัญในการฟักไข่**

อุณหภูมิในระหว่างการฟัก ตัวอ่อนที่มีชีวิตจะสามารถเจริญเติบโตได้ดีที่สุดในอุณหภูมิที่เหมาะสม ตัวอ่อนจะเจริญเติบโตได้ไม่ดีในอุณหภูมิที่สูงหรือต่ำกว่าอุณหภูมิที่เหมาะสม หรืออัตราการเจริญเติบโตจะลดลง และตัวอ่อนจะอ่อนแอมาก การทำให้อุณหภูมิต่ำจนตัวอ่อนหยุดการเจริญเติบโตเกินกว่าที่จะสามารถพัฒนาตัวเองได้ เป็นการยากมากที่จะสามารถกำหนดตัวเลขอุณหภูมิที่ใช้ให้ถูกต้องแน่นอนได้ เพราะมีปัจจัยหลายอย่างเข้ามาเกี่ยวข้องนอกจากนี้ความแปรปรวนยังขึ้นกับสายพันธุ์ของแม่ไก่ด้วยแต่โดยทั่วไปถ้าอุณหภูมิไม่เกิน 65 องศาฟาเรนไฮต์ ตัวอ่อนในฟองไข่ก็จะไม่เจริญเติบโต

1. อุณหภูมิที่เหมาะสมในการฟัก การฟักไข่ที่แท้จริงจะเกิดขึ้นครั้งแรกในตัวแม่ไก่เอง แม่ไก่บางตัวจะฟักไข่ที่อุณหภูมิประมาณ 95-105 องศาฟาเรนไฮต์ (35-40.5 องศาเซลเซียส) อย่างไรก็ตามอุณหภูมิทีเหมาะสมก็จะอยู่ในช่วงระหว่างตัวเลขทั้งสองนี้ ซึ่งถือว่าเป็นช่วงที่ตัวอ่อนสามารถเจริญเติบโตได้ดีที่สุด จากการศึกษาทำให้เรารู้ว่าตัวอ่อนต้องการอุณหภูมิช่วง 19 วันแรกของการฟัก

ค่อนข้างสูงกว่าช่วง 2 วันสุดท้ายของการฟัก ความถูกต้องของอุณหภูมิที่เหมาะสมจะเปลี่ยนแปลงไปตามการจัดการตู้ฟักและผู้ผลิต แต่ละรายจะกำหนดอุณหภูมิที่เหมาะสมต่อการฟักออกและคุณภาพของลูกไก่ที่ดีที่สุดมาให้แล้ว เมื่ออุณหภูมิในการฟักเปลี่ยนแปลงไปจากจุดที่เหมาะสม จะทำให้การฟักออกลดลง และ ลูกไก่ที่มีลักษณะผิดปกติจะมีเพิ่มมากขึ้นด้วย อุณหภูมิของอากาศที่อยู่ในตู้ฟักจะมีผลต่อระยะเวลาที่ใช้ในการออกฟัก อุณหภูมิที่สูงกว่าระดับที่เหมาะสมจะทำให้การฟักออกเร็วขึ้นและหากอุณหภูมิต่ำกว่าระดับที่เหมาะสม การฟักออกก็จะช้าลง ดังนั้นจึงไม่แนะนำให้เพิ่มหรือลดอุณหภูมิที่แตกต่างไปจากอุณหภูมิที่เหมาะสม เพราะจะทำให้ลูกไก่ที่ได้อ่อนแอและคุณภาพของลูกไก่เลวลง อุณหภูมิที่เหมาะสมในการฟักจะไม่เหมือนกันในไข่ทุกฟอง จึงจำเป็นต้องกำหนดอุณหภูมิที่ใช้โดยอาศัยค่าเฉลี่ย ซึ่งขึ้นอยู่กับปัจจัยหลายอย่าง เช่น ขนาดของฟองไข่ คุณภาพของเปลือกไข่ พันธุกรรม(รวมทั้งพันธุ์และสายพันธุ์ของไก่)อายุไข่เมื่อนำเข้าฟัก และความชื้นในอากาศในระหว่างการฟัก

2. อุณหภูมิที่เหมาะสมทั้ง 3 ระยะ การเจริญเติบโตของตัวอ่อนอาจแบ่งออกได้เป็น 3 ระยะ แต่ละระยะจะต้องการอุณหภูมิที่แตกต่างกัน ดังนี้

2.1 ก่อนการวางไข่ อุณหภูมิของเลือดในตัวแม่ไก่อาจผันแปรอยู่ระหว่าง 105 - 107 องศาฟาเรนไฮต์ (40.6 - 41.7 องศาเซลเซียส) ตัวอ่อนในระยะแรกจะแบ่งเซลล์ได้เป็นหลายเซลล์ในช่วง 22 ชั่วโมงแรก ในระหว่างที่มีการรวมตัวกันของเซลล์ไข่และเซลล์อสุจิและในช่วงที่กำลังวางไข่ อุณหภูมิที่เหมาะสมในการพัฒนาของตัวอ่อนในช่วงนี้ จะเท่ากับอุณหภูมิของตัวแม่ไก่

2.2 ช่วง 19 วันแรกของการฟัก แม้ว่าการฟักในช่วงแรกๆจะมีการเปลี่ยนแปลงอย่างช้าๆก็ตาม แต่อุณหภูมิที่เหมาะสมก็จะอยู่ประมาณ 99.5 องศาฟาเรนไฮต์ (37.5 องศาเซลเซียส)

2.3 ช่วงวันที่20 และ21ของการฟักการฟักออกที่ดีที่สุดในระยะนี้จะเกิดขึ้นในสภาพที่อุณหภูมิต่ำกว่าในช่วง 19 วันแรกของการฟัก คือประมาณ 98 - 99 องศาฟาเรนไฮต์ (36.4 - 37.2 องศาเซลเซียส) ความแปรปรวนของอุณหภูมิที่ใช้ในการฟักทั้งสามระยะนี้ จะเป็นตัวชี้ให้เห็นการพัฒนาของตัวอ่อนที่ขึ้นอยู่กับสภาพแวดล้อมที่เหมาะสมเป็นสำคัญ และเนื่องจากอุณหภูมิที่เหมาะสมในการฟักอยู่ในช่วงแคบ ๆ ดังนั้น ตู้ฟักไข่ที่ใช้จะต้องสามารถควบคุมอุณหภูมิให้อุณหภูมิภายในตู้เกิดมีความแปรปรวนให้น้อยที่สุด

3. การปรับอุณหภูมิในช่วงการฟัก อุณหภูมิที่เหมาะสมในการฟักของตัวอ่อน จะใช้ค่าเฉลี่ยของไข่ที่นำเข้าฟักทั้งหมดมาเป็นตัวกำหนด การฟักออกจะเพิ่มขึ้นอย่างเห็นได้ชัด หากเรารู้ว่าที่อุณหภูมิที่ระดับไหนเหมาะสมกับการฟักไข่ในช่วงใดของการฟักของไข่แต่ละประเภท ไม่ว่าจะเป็นขนาดของไข่ที่ฟักที่จะเปลี่ยนแปลงไปในแต่ละฤดูกาล และสภาพแวดล้อมอื่น ๆ ที่เข้ามาเกี่ยวข้อง ควรมีการปรับอุณหภูมิที่เหมาะสมให้กับไข่ฟักทั้งสามระยะ ควรมีการทดลองหาอุณหภูมิที่เหมาะสมในการฟักในสภาพพื้นที่ต่าง ๆ ตลอดจนประเภทของไข่ฟักที่เกี่ยวข้อง บางครั้งอาจทำให้การฟักออกและคุณภาพของลูกไก่ที่ดีขึ้น

4. ผลของอุณหภูมิสูงที่เกิดกับตัวอ่อน ในระยะเวลา 24 ชั่วโมง หากตัวอ่อนได้รับอุณหภูมิสูงถึง 104 องศาฟาเรนไฮต์ (40.0 องศาเซลเซียส) เป็นเวลา 16 ชั่วโมง จะไม่ใช่สาเหตุหลักที่ทำให้เกิดความเสียหายต่อการฟักออก แต่หากตัวอ่อนได้รับอุณหภูมิสูงถึง 110 องศาฟาเรนไฮต์ (43.3 องศาเซลเซียส) เป็นเวลานาน 6 ชั่วโมง จะเป็นสาเหตุที่ทำให้การฟักออกลดลงและหากได้รับนานถึง 9 ชั่วโมง อาจทำให้การฟักออกลดลงมากหากตัวอ่อนได้รับอุณหภูมิสูงถึง 115 องศาฟาเรนไฮต์ (46.1 องศาฟาเรนไฮต์) เป็นเวลา 3 ชั่วโมงหรือ120 องศาฟาเรนไฮต์ (48.6 องศาเซลเซียส) เป็นเวลา 1 ชั่วโมง สามารถทำให้ตัวอ่อนทั้งหมดตายได้ ลูกไก่ที่ได้รับความเครียดที่เกิดจากอุณหภูมิสูงจะมีลักษณะเป็นขนยุ่งเป็นกลุ่มๆ ขนไหม้แดงสั้นและแข็ง และลักษณะการเดินจะมีลักษณะผิดปกติไม่มั่นคง

5. ผลของอุณหภูมิต่ำที่เกิดกับตัวอ่อน ในธรรมชาติแม่ไก่จะออกจากรังไข่วันละหลาย ๆ ครั้ง ในระหว่างที่กำลังฟักไข่ การเย็นลงของตัวไข่ในระหว่างที่แม่ไก่ไม่อยู่ในช่วงสั้นๆนี้ ไม่ได้เป็นตัวกำหนดการฟักออกที่แน่ชัดมากนัก ในระหว่างการฟักไข่โดยใช้ตู้ฟักก็มีหลายๆครั้งเช่นกันที่เกิดไฟฟ้าดับ ซึ่งทำให้อุณหภูมิภายในตู้ฟักลดลงเป็นช่วงเวลาสั้น ๆ หรืออาจหลาย ๆ ชั่วโมง ในระยะ 3 วันแรกของการฟักตัวอ่อนจะมีความทนทานต่อการที่ได้รับอุณหภูมิต่ำ มากกว่าในช่วงหลังของการฟัก หลังจาก 3 วันไปแล้วความทนทานจะค่อย ๆ ลดลงไปตามลำดับจนถึงวันที่ 19 ของการฟัก ลูกไก่ที่ฟักออกแล้วในตู้เกิดจะทนต่อสภาพความเย็นได้มากกว่าสภาพอากาศร้อน

6. อุณหภูมิต่ำจะทำให้ต้องใช้เวลาในการฟักออกนานขึ้น เนื่องจากการฟักออกเป็นตัวของตัวอ่อนเป็นขบวนการที่มีความซับซ้อนมาก การลดลงของอุณหภูมิในช่วงของการฟักทำให้การฟักออกต้องใช้เวลามากขึ้น อย่างไรก็ตามเวลาในการฟักจะเพิ่มขึ้น เพียงแค่ครึ่งเดียวของเวลาที่ไข่ได้รับความเย็นเท่านั้น เนื่องจากอุณหภูมิในตู้ฟักยากที่จะลดลงมาถึงอุณหภูมิห้องได้หรือประมาณ 65-70 องศาฟาเรนไฮต์ (18-21 องศาเซลเซียส) สิ่งที่อยู่ภายในฟองไข่จะสามารถรักษาอุณหภูมิในฟองไข่ให้สูงกว่าอุณหภูมิห้องอยู่ได้นานหลายชั่วโมง

7. อุณหภูมิต่ำเป็นสาเหตุทำให้ลูกไก่อยู่ผิดตำแหน่งมากขึ้น ไข่ที่ได้รับอุณหภูมิที่ต่ำ ในช่วง 19 วันแรกของการฟัก จะเป็นสาเหตุทำให้ตัวอ่อนมีลักษณะอยู่ผิดท่าและผิดตำแหน่งเพิ่มมากขึ้น

8. อุณหภูมิต่ำในช่วงวันที่ 20 และ 21 ของการฟัก ถึงแม้ว่าตัวอ่อนจะสามารถทนอุณหภูมิต่ำในช่วง 19 วันแรกของการฟักในระยะสั้นๆได้พอสมควร แต่ในช่วง 2 วันสุดท้ายของการฟักหากเกิดอุณหภูมิที่ตัวอ่อนได้รับต่ำลงมากกว่า 2 องศาฟาเรนไฮต์ ก็อาจทำให้เกิดความเสียหายอย่างมากมายได้ การขัดข้องของไฟฟ้าคือจุดสำคัญที่มีผลต่อการฟักออกไม่สม่ำเสมอ

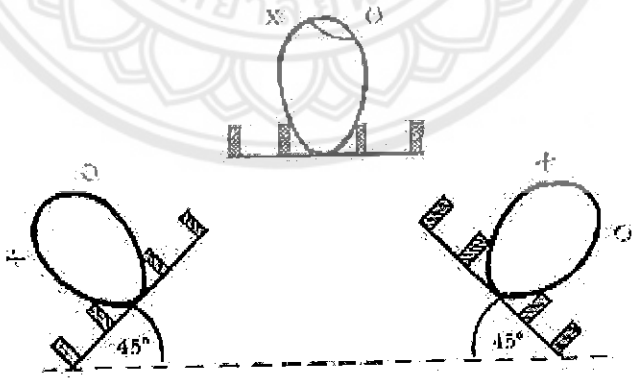
ในกรณีที่ไฟฟ้าดับ พัดลมที่ใช้ในการกระจายความร้อนในตู้ให้สม่ำเสมอจะหยุดทำงาน อุณหภูมิในตู้จะเพิ่มขึ้นสูงสุดไข่ที่อยู่ในห้องนั้นจะได้รับความร้อนสูงมาก ในขณะที่ส่วนที่อยู่ด้านล่างจะได้รับความเย็น เมื่อลูกไก่ฟักออกความร้อนส่วนเกินนี้จะทำให้ลูกไก่ตายและหายใจไม่ออก

9. การติดตั้งเครื่องกำเนิดไฟฟ้า กระแสไฟฟ้าเป็นสิ่งที่จำเป็นอย่างยิ่งในช่วงกำลังที่มีการฟักออก (20 - 21 วัน) การติดตั้งเครื่องกำเนิดไฟฟ้าสำรองจึงมีความจำเป็นอย่างมาก การติดตั้งเครื่องกำเนิดไฟฟ้าที่สามารถให้กระแสไฟฟ้าได้เกินกว่าความต้องการ กระแสในช่วงปกติ และไฟฟ้าที่ใช้ไม่ควรติด ๆ ดับ ๆ

10. การตรวจสอบเทอร์โมมิเตอร์ ควรมีโปรแกรมการตรวจสอบความถูกต้องแน่นอนของเครื่องมือวัดอุณหภูมิเป็นครั้งคราวอยู่เป็นประจำ ซึ่งจะทำให้เกิดความมั่นใจในระดับหนึ่ง

ตำแหน่งการวางไข่ในช่วงการพัก ไข่ฟักจำเป็นต้องวางให้อยู่ในตำแหน่งที่เหมาะสม และต้องมีการกลับไข่ทุกวัน ในช่วง 1-18 วันของการฟัก ไข่ที่นำเข้าฟักควรวางเอาด้านป้านขึ้น ซึ่งมันจะเป็นด้านป้านขึ้น ซึ่งมันจะเป็นธรรมชาติ เพราะศีรษะของลูกไก่ปกติจะอยู่ใกล้ช่องอากาศ และการเจริญเติบโตของตัวอ่อนส่วนมากมักจะหันเอาศีรษะขึ้นด้านบน การหมุนตัวของลูกไก่ โดยทั่วไปจะเกิดขึ้นในช่วง 2 สัปดาห์แรกของการฟัก และจะเสร็จสมบูรณ์ได้ง่ายกว่าการวางไข่ด้านบนซึ่งเป็นด้านแหลมของฟองไข่ ดังนั้น เมื่อลูกไก่พร้อมที่จะฟักออกจงอยปากของลูกไก่ก็จะไม่สามารถเจาะผ่านเข้าไปในช่องอากาศเพื่อเริ่มหายใจด้วยปอดได้

การกลับไข่ในระหว่างการฟัก ไข่แดงในไข่ที่ออกมาจากแม่ไก่ใหม่ๆ จะมีความถ่วงจำเพาะที่พอเหมาะที่สามารถทรงตัวอยู่ในไข่ขาวได้ ไข่ที่อยู่ในช่วงของการฟัก หากไม่มีการกลับไข่จะทำให้ความถ่วงจำเพาะของไข่ลดลง ทำให้ไข่แดงขึ้นมาอยู่ที่ไข่ขาวเหลวติดกับด้านนอกของไข่ขาวนั้น ไข่ขาวข้นจะถูกแยกออกเป็นสองชั้น โดยไข่ขาวเหลวมาติดอยู่ข้างใน และทำให้ตัวอ่อนตายในที่สุด




**รูปที่2.1** ตำแหน่งการวางไข่ฟัก ต้องทำมุม 45 องศา และกลับไข่หมุนสลับไปมา

การฟักไข่ตามธรรมชาติแม่ไก่จะกลับไข่วันละหลายๆครั้ง ส่วนการฟักไข่ที่ใช้ตู้ฟักจะวางไข่โดยเอาด้านป้านขึ้นข้างบน และมีการกลับไข่โดยการหมุนกลับไปกลับมาตามแนวนอน ไม่ควรกลับไข่เป็นวงกลมโดยรอบ เพราะจะทำให้ถุงน้ำคล่ำแตกและเป็นผลให้ตัวอ่อนตาย การกลับไข่ส่วนมากจะทำให้อยู่ในตำแหน่งประมาณ 45 องศาจากด้านที่ตั้งตรงและเมื่อกลับอีกครั้ง ก็จะไปอยู่ในตำแหน่งที่ตรงกันข้ามกับตำแหน่งเดิม

1. ระยะห่างในการกลับไข่ ในระหว่าง 18 วันแรกของการฟัก ต้องมีการกลับไข่เป็นประจำและบ่อยๆ การฟักออกของไข่ที่ใช้วลาในการกลับไข่แตกต่างกันจะทำให้เปอร์เซนต์การฟักไข่แตกต่างกันตั้งแต่ 2-10 ครั้งต่อวันแสดงไว้ในตารางที่ 2.1

ตารางที่ 2.1 แสดงผลของจำนวนครั้งในการกลับไข่ต่อวันในช่วงการฟัก

| จำนวนครั้งในการกลับไข่/วัน | การฟักออกของไข่มีเชื้อ(%) |
|---|---|
| 2 | 78.1 |
| 4 | 85.3 |
| 6 | 92.0 |
| 8 | 92.1 |
| 10 | 92.2 |

ถึงแม้จะมีการกลับไข่ทุก 15 นาทีก็ไม่ได้ทำให้การฟักเสียหายการกลับไข่มากกว่า 6 ครั้งต่อวันไม่ได้ทำให้การฟักออกดีขึ้นกว่าเดิม

2. ระยะเวลาในการกลับไข่ ตารางที่ 2.2 แสดงผลของระยะเวลาที่ใช้ในการกลับไข่แต่ละระยะของการฟัก ผลการทดลองนี้จะเป็นตัวชี้ให้เห็นว่าการกลับไข่ในสองสัปดาห์แรก จะมีความสำคัญมาก การกลับไข่ในช่วงสัปดาห์สุดท้ายของการฟัก ในทางปฏิบัติไข่ที่นำเข้าฟักจะมีหลายอายุอยู่ในตู้เดียวกันดังนั้น การกลับไข่ทั้งหมดจึงต้องทำร่วมกัน และสิ่งที่สำคัญคือ การกลับไข่ทุกครั้งจะต้องทำให้ขบวนการนี้สมบูรณ์ที่สุดโดยเร็ว ไข่จะหยุดนิ่งอยู่กับที่จนกระทั่งมีการกลับไข่ครั้งต่อไป การฟักออกจะลดลงเมื่อไข่อยู่นิ่งกับที่โดยไม่มีการกลับไข่

**ตารางที่2.2** แสดงผลของระยะเวลาที่ใช้ในการกลับไข่ในแต่ละช่วงของการฟัก

| จำนวนครั้งในการกลับไข่ | การฟักออกของไข่มีเชื้อ (%) |
|---|---|
| ไม่มีการกลับไข่ | 28 |
| 1-7 | 78 |
| 1-14 | 95 |
| 1-18 | 92 |

**2.1.3 ตู้ฟักไข่**

ตู้ฟักไข่ในปัจจุบันมีอุปกรณ์ที่สำคัญต่างๆที่ช่วยควบคุมระบบการทำงานของตู้ฟักไข่มากมายขึ้นอยู่กับชนิดและขนาดของตู้ฟักไข่ อุปกรณ์ของตู้ฟักไข่มีดังต่อไปนี้

อุปกรณ์ให้ความร้อนและควบคุมอุณหภูมิในตู้ฟักไข่ ซึ่งพอจะจำแนกรายละเอียดดังนี้

1. อุปกรณ์ให้ความร้อนในตู้ฟักไข่ ความร้อนของอากาศในตู้ฟักไข่ได้จากพลังงานไฟฟ้าซึ่งสามารถให้ความร้อนได้โดยผ่านขดลวดหรือหลอดไฟ

2. อุปกรณ์ควบคุมอุณหภูมิในตู้ฟักไข่ ซึ่งประกอบไปด้วยตัวรับความรู้สึกจากความร้อน (Sensor) และตัวสวิตซ์ มีหน้าที่ควบคุมการทำงานของอุปกรณ์ไฟฟ้าที่ให้ความร้อน

อุปกรณ์ทำความชื้นและควบคุมความชื้นในตู้ฟักไข่

โดยมากแล้วสำหรับตู้ฟักไข่ที่ไม่ได้เป็นอุตสาหกรรมขนาดใหญ่ ก็จะใช้เพียงถาดน้ำ เพื่อใช้วิธีการระเหยของน้ำ เพื่อเพิ่มหรือลดความชื้นของตู้ฟักไข่โดยการเพิ่มพื้นที่ผิวของการระเหยน้ำ ทำได้โดยการเพิ่มขนาดของภาชนะบรรจุน้ำ หรือใช้ผ้าหรือวัสดุอื่นๆที่สามารถดูดซับน้ำได้ดีแขวนไว้ในตู้ฟักไข่

3. รูระบายอากาศและอุปกรณ์ควบคุมการระบายอากาศในตู้ฟักไข่

4. อุปกรณ์กลับไข่ในตู้ฟักไข่

5. แผงไฟและอุปกรณ์ตัดตอนทางไฟฟ้า

6. แบตเตอรี่

## 2.2 ไมโครคอนโทรลเลอร์ตระกูล 8051 [2]

บรรดาไมโครคอนโทรลเลอร์ที่มีการผลิตจากบริษัทต่าง ๆ จำนวนมากขึ้น ไมโครคอนโทรลเลอร์จากบริษัท อินเทล (Intel Corporation) ในตระกูล MCS-51 ได้มีการนำไปใช้งานกันแพร่หลายนับตั้งแต่ปี ค.ศ.1980 เป็นต้นมา ในระยะที่ผ่านมาก็ได้มีอีกหลายบริษัท เช่น Phillips และ Siemens เป็นต้น ได้รับสิทธิในการผลิตจำหน่ายและได้มีการเพิ่มประสิทธิภาพและหน่วยการทำงานต่าง ๆ มากขึ้น ทำให้ปัจจุบันมีไมโครคอนโทรลเลอร์จากผู้ผลิตต่าง ๆ ที่มีที่มีพื้นฐานมาจากไมโครคอนโทรลเลอร์ MCS-51 ของบริษัทอินเทลอยู่เป็นจำนวนมาก

ไมโครคอนโทรลเลอร์ตระกูลMCS-51ประกอบด้วยไมโครคอนโทรลเลอร์หลายรุ่น (version) ซึ่งมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหรือจำนวนของหน่วยทำงานภายในที่แตกต่างกันออกไป เพื่อความเหมาะสมในงานประยุกต์ต่าง ๆ ตามความต้องการ ดังแสดงให้เห็นดังตารางของรูปที่ 2.3 โดยมีทั้งลักษณะที่ใช้เทคโนโลยีชิปความเร็วสูงมาก (LSI) แบบ HMOS หรือ CHMOS ซึ่งมีคุณลักษณะที่สูงมากขึ้น และสิ้นเปลืองกำลังไฟฟ้าน้อยกว่ามาก อย่างไรก็ตามการอ้างถึงไมโครคอนโทรลเลอร์ MCS-51 ในที่นี้จะเรียกรวมกันว่า 8051 แทน

### 2.2.1 คุณลักษณะพื้นฐานของ 8051

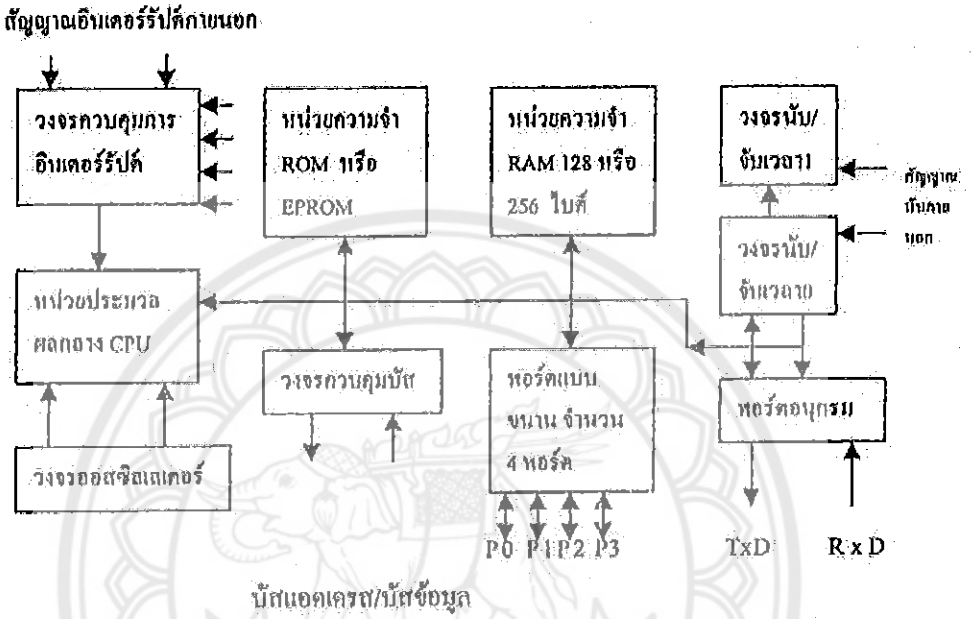
จากแผนภาพในรูปที่ 2.2 แสดงให้เห็นถึงหน่วยการทำงานพื้นฐานของไมโครคอนโทรลเลอร์เบอร์ต่าง ๆ ที่จัดอยู่ภายในตระกูล MCS-51 นี้ประกอบด้วย

- หน่วยประมวลผลกลางขนาด 8 บิต
- หน่วยประมวลผลสำหรับข้อมูลแบบบิต ( Boolean Processor )
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำโปรแกรม 64 กิโลไบต์
- ความสามารถในการอ้างตำแหน่งของหน่วยความจำข้อมูล 64 กิโลไบต์
- หน่วยความจำโปรแกรมภายในขนาด 4 กิโลไบต์ แบบอีพรอม ( EPROM ) เบอร์ 8751 หรือ แบบรอม ( ROM ) เบอร์ 8051
- พอร์ตอินพุตเอาต์พุตแบบขนานจำนวน 32 เส้น ซึ่งสามารถทำงานได้อย่างอิสระ
- วงจรนับ/จับเวลาขนาด 16 บิต จำนวนสองวงจร
- วงจรสื่อสารแบบอนุกรมแบบฟูลดูเพล็กซ์ ( Full Duplex )

**ตารางที่ 2.3** ตารางแสดงไมโครคอนโทรลเลอร์ตระกูล MCS-51 ของบริษัทอินเทล

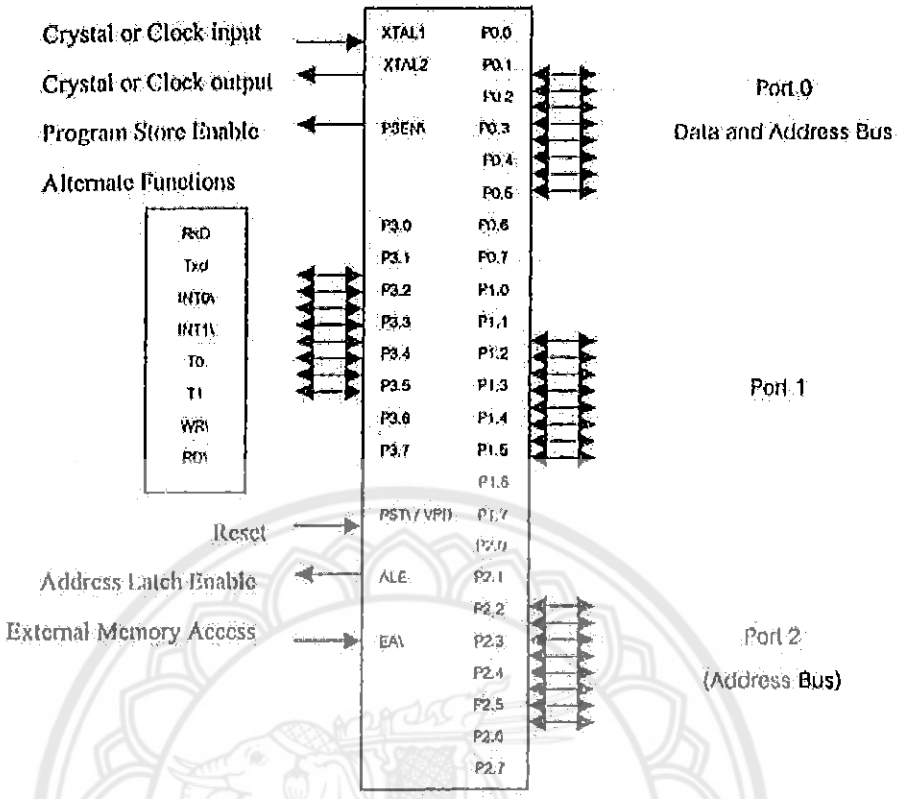
| EMBEDDED CONTROLLERS | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Feature | 8051AH | 8031AH | 8751H | 80C51BH | 80C31BH | 87C51 | 8052AH | 8032AH | 8752 | 8044H |
| Program memory (Bytes) | 4k | - | 4k | 4k | - | 4k | 8k | - | 8k | 4k |
| RAM Memory (Bytes) | 128 | 128 | 128 | 128 | 1218 | 128 | 256 | 256 | 256 | 194 |
| Program Memory Expansion | 64K | 64K | 64K | 64K | 64K | 64K | 64K | 64K | 64K | 64K |
| Data Memory Expansion | 64K | 64K | 64K | 64K | 64K | 64K | 64K | 64K | 64K | 64K |
| MaxClock Frequency (Mhz) | 12 | 12 | 12 | 16 | 16 | 16 | 16 | 12 | 12 | 12 |
| Typical Instruction Time(uS) | 1 | 1 | 1 | 0.75 | 0.75 | 0.75 | 1 | 1 | 1 | 1 |
| 16-bit Timer/counter | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 |
| Serail Comunications | Synchronous Mode, Asynchronous Mode, 9 or 10 – Bit Programable | | | | | | | | | HDLC / SDLC |
| No. of I/O Lines Interrupt Sources | 32 5 | 16 5 | 32 5 | 32 5 | 16 5 | 32 5 | 32 6 | 16 6 | 32 6 | 32 5 |
| (Two Priority Level) Power Requirements 125 | 125 | 250 | 24 | 24 | 29 | 175 | 175 | 175 | 200 | |
| (ICC Max mA) Programmable PowerModes idle Power down | - | - | - | 4.0 mA 50uA | 4.0 mA 51uA | 4.0 mA 52uA | - | - | - | - 30mA |

- วงจรควบคุมการอินเตอร์รัปต์จากแหล่งกำเนิดสัญญาณ 6 ประเภทพร้อมการกำหนดลำดับความสำคัญได้สองระดับ
- วงจรออสซิลเลเตอร์ภายใน




รูปที่ 2.2 แผนภาพบล็อกแสดงหน่วยทำงานพื้นฐานของ MCS-51

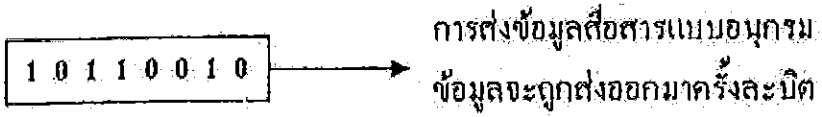
โดยมากแล้วไมโครคอนโทรลเลอร์ตระกูลนี้ มักจะมีรูปร่างของไอซีเป็นแบบ DIP ขนาด 40 ขา ดังแผนภาพในรูปที่ 2.3 ซึ่งแต่ละขาสัญญาณจะมีหน้าที่ที่ระบุชัดเจนตามสัญลักษณ์ชื่อย่อที่กำกับในแต่ละขา อย่างไรก็ตามจะมีบางขาสัญญาณที่อาจจะมีหน้าที่ได้มากกว่าหนึ่งอย่าง (ซึ่งเขียนกำกับไว้ว่า Alternate Function ในรูปที่ 2.3) ซึ่งจะไม่สามารถใช้งานในเวลาเดียวกันได้ ตัวอย่างเช่น ขาสัญญาณบิต 0 ของพอร์ต 3 (ใช้ตัวย่อ P3.0) อาจจะใช้เป็นขาสัญญาณเอาต์พุต/อินพุตตามปกติ หรืออาจทำหน้าที่เป็นขาสัญญาณอินพุตของข้อมูลสื่อสารแบบอนุกรม (R&D) ให้กับวงจรสื่อสารแบบอนุกรมของ8051 ได้ ซึ่งการกำหนดว่าจะทำงานในลักษณะใดก็ขึ้นอยู่กับการเชื่อมต่อวงจรเข้ากับขาสัญญาณและโปรแกรมควบคุมของระบบนั้น



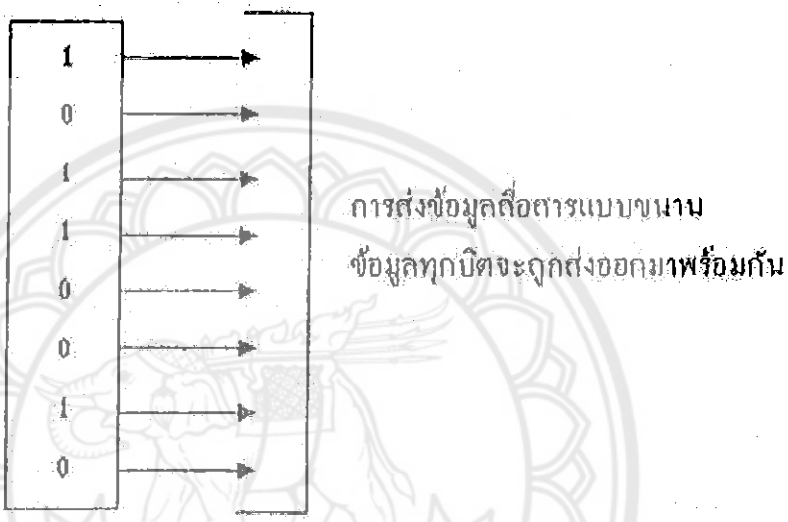

รูปที่2.3 การกำหนดหน้าที่ขาสัญญาณของไอซี 8051

**2.2.2 การสื่อสารข้อมูลอนุกรม**

การสื่อสารข้อมูลอนุกรมเป็นการรับส่งข้อมูลในลักษณะของบิตหรือกลุ่มของบิต คราวละหนึ่งบิตเป็นลำดับเรื่อยไปจนสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมากเนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกันจึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วยในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆเพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก ลองพิจารณาเปรียบเทียบการสื่อสารทั้งสองประเภทได้จากรูป 2.4 (a) และ 2.4 (b)



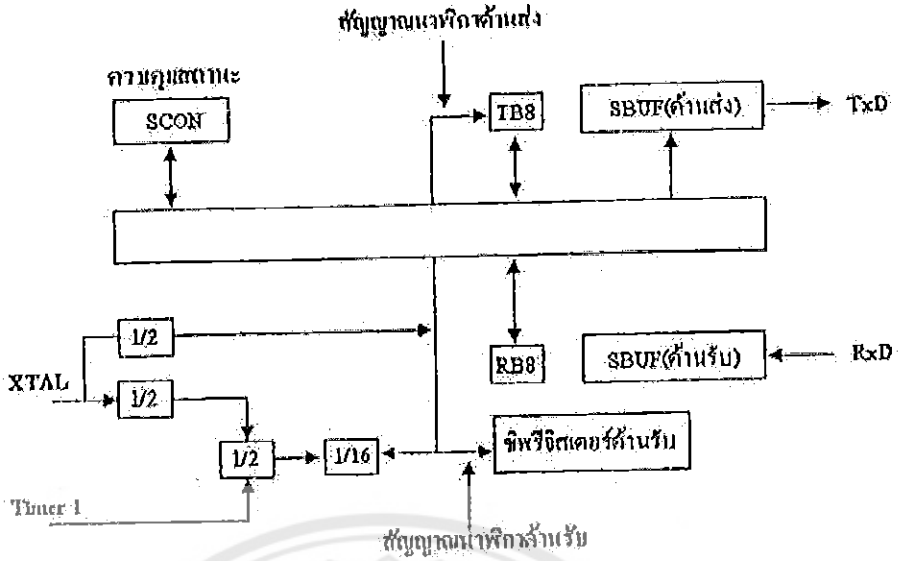

**รูปที่ 2.4 (a)** ข้อมูลสื่อสารแบบอนุกรม ข้อมูลหนึ่งไบต์จะถูกส่งออกคราวละบิตเป็นลำดับจนครบทั้งแปด




**รูปที่ 2.4(b)** ข้อมูลสื่อสารแบบขนาน ข้อมูลแต่ละบิตภายในหนึ่งไบต์จะถูกส่งออกมาพร้อมกัน

### 2.2.3 การจัดการข้อมูลอนุกรมของ 8051

พอร์ตอนุกรมของ 8051 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ซึ่งหมายถึงความสามารถในการรับและส่งข้อมูลอนุกรมในเวลาเดียวกัน จากรูปที่2.5 แสดงให้เห็นถึงแผนภาพการทำงานอย่างง่ายของวงจรส่วนจัดการข้อมูลอนุกรมของ 8051 โดยทางด้านวงจรตัวส่ง ประกอบด้วยรีจีสเตอร์ SBUF ทำหน้าที่เก็บข้อมูลที่จะส่งออก การใช้คำสั่งเขียนหรือโอนย้ายข้อมูลมายังรีจีสเตอร์นี้จะเป็นการส่งข้อมูลนั้นออกไปยังพอร์ตอนุกรมทางขาสัญญาณ TxD โดยอัตโนมัติ ส่วนวงจรด้านตัวรับ ประกอบด้วยรีจีสเตอร์ SBUF เช่นเดียวกัน แต่ทำหน้าที่เก็บข้อมูลที่นำมาจากส่วนของวงจรเลื่อนบิตหรือชิพรีจีสเตอร์ของวงจรจัดการข้อมูลอนุกรมภายในสัญญาณข้อมูลอนุกรมที่รับเข้าจะผ่านมาทางขาสัญญาณ RxD




**รูปที่ 2.5** แผนภาพแสดงการทำงานของวงจรส่วนการรับและส่งข้อมูลอนุกรมของ 8051

พอร์ตอนุกรมของ 8051 สามารถโปรแกรมให้ทำหน้าที่ในรูปแบบต่างๆกัน 4 แบบ โดยการกำหนดค่าบิต SM0 และ SM1 ซึ่งอยู่ภายในรีจิสเตอร์ควบคุมและบอกสถานะ SCON ดังตารางที่ 2.4 โหมดการทำงานทั้ง 4 แบบของพอร์ตอนุกรมมีดังนี้

**ตารางที่ 2.4** โหมดการทำงานของพอร์ตอนุกรม

| โหมดการทำงาน | คำอธิบาย |
|---|---|
| โหมด0 | เป็นการขยายพอร์ตอินพุตเอาต์พุต โดยร่วมกับไอซีชิพรีจีสเตอร์ภายนอก ประเภททีทีแอลหรือซีมอส |
| โหมด1 | ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal asychronous reciver / transmitter) โดยการใช้กลุ่มข้อมูลแบบ 10 บิต และสามารถเปลี่ยนแปลงอัตราเร็วในการส่งข้อมูลได้ |
| โหมด2 | ใช้สำหรับเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิตและกำหนดอัตราเร็วในการส่งข้อมูลคงที่ |
| โหมด3 | ใช้สำหรับเชื่อต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิตและสามารถเปลี่ยนแปลงอัตราเร็วในการส่งข้อมูลได้ |

นอกจากนี้โหมด 2 และ 3 ยังมีการดำเนินการแบบพิเศษออกไป โดยสามารถนำมาใช้ประโยชน์ในการสื่อสารข้อมูลแบบที่มีไมโครโปรเซสเซอร์หลายตัวทำงานร่วมกันได้ ซึ่งจะได้อธิบายรายละเอียดเป็นลำดับไป

จากแผนภาพตารางที่ 2.5 ชิพรีจีสเตอร์ภายในตัวส่งจะทำหน้าที่เลื่อนบิตข้อมูลออกไปภายนอกโดยไม่มีการบัพเฟอร์ และเมื่อใดที่มีการเขียนข้อมูลให้กับรีจีสเตอร์ SBUF แสดงว่ามีความต้องการที่จะส่งข้อมูลนี้ออกไปแบบอนุกรม สำหรับชิพรีจีสเตอร์ทางด้านรับจะทำการเลื่อนบิตข้อมูลที่รับเข้ามาเก็บไว้ เมื่อบิตของข้อมูลที่รับเข้ามาครบถ้วนตามจำนวนที่กำหนดไว้ตามลักษณะโหมดการทำงานต่างๆแล้ว จะถูกย้ายไปเก็บยังรีจิสเตอร์ SBUF ต่อไป อย่างไรก็ตามการย้ายข้อมูลนี้จะเกิดขึ้นก็ต่อเมื่อรีจีสเตอร์ SBUF นั้นไม่มีข้อมูลที่จะทำการส่งหรือได้ส่งข้อมูลออกไปเสร็จสิ้นแล้ว

**ตาราง**ที่ 2.5 รีจีสเตอร์ควบคุมการทำงานและบอกสถานะการสื่อสารข้อมูล**อนุกรม** SCON

| ชื่อบิต | ตำแหน่ง | ความหมาย |
|---|---|---|
| SM0 | SCON.7 | บิตเลือกโหมดการทำงาน |
| SM1 | SCON.6 | บิตเลือกโหมดการทำงาน |
| SM2 | SCON.5 | แฟล็กกำหนดการทำงานแบบมัลติโปรเซสเซอร์ |
| REN | SCON.4 | แฟล็กยอมให้มีการรับข้อมูล |
| TB8 | SCON.3 | ค่าของบิตที่9สำหรับการส่งข้อมูลออก |
| RB8 | SCON.2 | ค่าของบิตที่9ของข้อมูลที่รับเข้า |
| TI | SCON.1 | แฟล็กแสดงการอินเตอร์รัปต์ภายหลังการส่งข้อมูล |
| RI | SCON.0 | แฟล็กแสดงการอินเตอร์รัปต์เมื่อมีข้อมูลรับเข้า |

### 2.2.4 การอินเตอร์รัปต์ของการสื่อสารอนุกรม

เนื่องจากการส่งหรือรับข้อมูลอนุกรมในการส่งข้อมูลไบต์หนึ่งๆ ค่อนข้างจะใช้เวลานานหลายมิลลิวินาที ดังนั้นเพื่อการจัดการเกี่ยวกับการสื่อสารแบบนี้เป็นไปอย่างมีประสิทธิภาพ 8051 จึงได้กำหนดให้บิตหรือแฟล็กสถานะที่เกี่ยวข้องทั้งหมด จัดรวมอยู่ภายในรีจีสเตอร์ SCON เท่านั้นเช่น แฟล็ก TI ซึ่งมีค่าเป็นหนึ่ง เพื่อแจ้งให้ทราบว่าได้รับข้อมูลผ่านเข้าทางพอร์ตอนุกรม เมื่อแฟล็กตัวใดตัวหนึ่งนี้มีค่าเป็นหนึ่ง จะมีผลทำให้เกิดการอินเตอร์รัปต์ขึ้น ดังนั้นภายในโปรแกรมจะต้องทำการตรวจสอบจากสถาวะของแฟล็กเหล่านี้เองว่ามีการอินเตอร์รัปต์ขึ้นด้วยสาเหตุใด จากนั้นจึงค่อยทำการกำหนดค่าศูนย์ให้กับแฟล็กนั้น ลักษณะดังกล่าวนี้จะมีความแตกต่างไปจากการอินเตอร์รัปต์จากสัญญาณอื่นๆเช่น วงจรนับ วงจรจับเวลา เป็นต้น ซึ่งจะมีการกำหนดค่าศูนย์ ให้กับแฟล็กสถานะที่เกี่ยวข้องโดยอัตโนมัติ ภายหลังจากที่ได้เข้าไปทำงานยังส่วนของโปรแกรมย่อยบริการอินเตอร์รัปต์ ดังนั้นจึงขอให้สังเกตความแตกต่างในส่วนนี้ด้วย

**2.2.5 กระบวนการรับและส่งข้อมูลอนุกรมของ 8051**

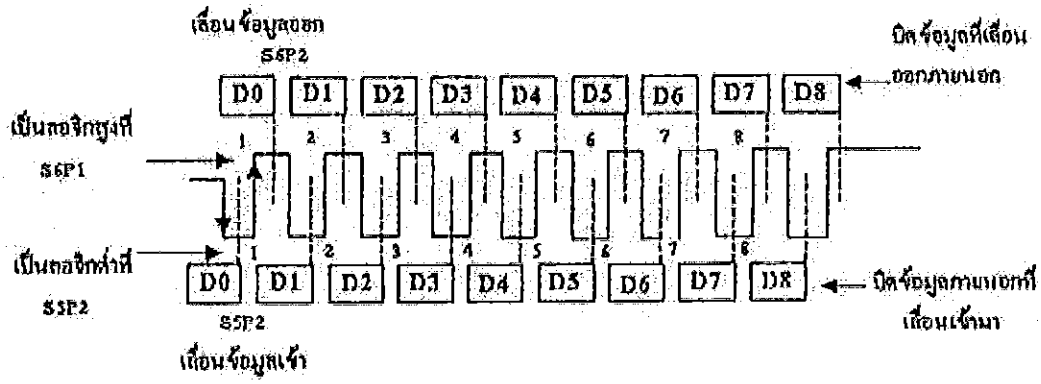
การส่งข้อมูลออกทางพอร์ตอนุกรมของ8051 จะเริ่มต้นขึ้น ภายหลังเมื่อมีการเขียนข้อมูลลงในรีจิสเตอร์ SBUF ข้อมูลนี้จะถูกจัดการด้วยวิธีการทางด้านฮาร์ดแวร์ในการเลื่อนบิตและส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกครบถ้วนแล้วจึงทำการกำหนดค่าของแฟล็ก TI ให้เป็นหนึ่งเพื่อแจ้งให้ทราบว่า ขณะนี้รีจิสเตอร์ SBUF ว่าง และพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้ว ในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่ต้องรอให้แฟล็ก TI มีค่าเป็นหนึ่งก่อนจะมีผลทำให้ข้อมูลที่ส่งออกไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่าบิต REN (Reciever Enable) ให้มีค่าเป็นหนึ่งก่อน หลังจากนั้นเมื่อมีบิตของข้อมูลถูกส่งเข้ามาจากภายนอกระบบฮาร์ดแวร์ของ8051 จึงจะทำการเลื่อนบิตเหล่านี้เข้ามาโดยอัตโนมัติ และเมื่อบิตสุดท้ายถูกเลื่อนเข้าเรียบร้อยแล้ว ข้อมูลเหล่านั้นจะถูกย้ายมาเก็บยังรีจิสเตอร์ SBUF และทำการกำหนดให้แฟล็ก RI ให้มีค่าเป็นหนึ่ง ซึ่งมีผลทำให้เกิดการอินเตอร์รัปต์โปรแกรมขึ้น

**2.2.6 พอร์ตอนุกรมโหมด 0**

การทำงานของพอร์ตอนุกรมในโหมดศูนย์ เป็นการขยายพอร์ตอินพุตหรือพอร์ตเอาต์พุตให้มีจำนวนมากขึ้น โดยจะทำการสร้างสัญญาณนาฬิกาขึ้นเพื่อให้จังหวะของการทำงานที่พร้อมกันสำหรับการเลื่อนบิตเข้าหรือออกจากไอซีรีจิสเตอร์ภายนอก เมื่อมีการโอนย้ายข้อมูลมายังรีจิสเตอร์ในแต่ละครั้งจะมีผลทำให้เกิดการส่งบิตข้อมูลทั้ง 8 บิตออกมา แม้ว่าแฟล็ก สถานะ TI ยังคงมีค่าเป็นหนึ่งอยู่ก็ตาม นอกจากนี้แล้วเมื่อใดก็ตามค่าของแฟล็กสถานะ RI เป็นค่าหนึ่งก็ควรจะย้ายข้อมูลที่รับเข้ามานั้นออกไปจากรีจิสเตอร์ SBUF เสียก่อนที่จะได้มีการกำหนดค่าแฟล็ก RI ให้เป็นศูนย์เพื่อรับข้อมูลต่อไป

การทำงานของพอร์ตอนุกรมในโหมดศูนย์ เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิตโดยเพียงขาสัญญาณ RxD เท่านั้น ส่วนขาสัญญาณ TxD จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะการเลื่อนข้อมูลกับวงจรเลื่อนบิตภายนอก สำหรับอัตราการเลื่อนบิต จะถูกกำหนดไว้ที่ค่าคงที่ที่ค่า1/2 ของความถี่ออสซิลเลเตอร์ จากรูปที่ 2.6 แสดงให้เห็นถึงแผนภาพเวลาสัญญาณต่างๆในโหมดศูนย์ เมื่อมีการรับและส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกา โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดขึ้นในตัวของ 8051 เอง และมีจุดประสงค์เพื่อนำไปใช้สำหรับวงจรชิพรีจิสเตอร์ภายนอกเท่านั้น



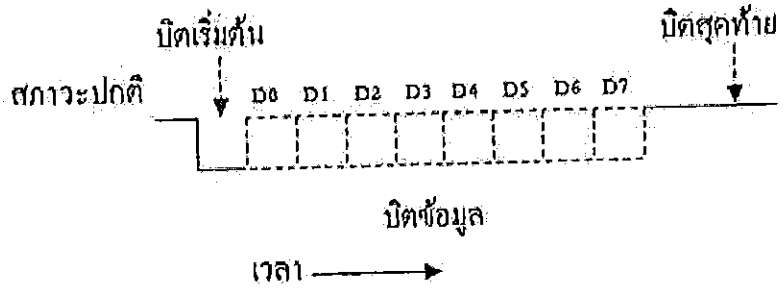

รูปที่ 2.6 แผนภาพเวลาของสัญญาณอนุกรมโหมดศูนย์

สัญญาณนาฬิกาที่สร้างขึ้นทางขาสัญญาณ TxD นี้จะสลับค่าไปมาจากลอจิกสูงไปต่ำในราวช่วงใกล้เคียงกับเวลาขอบขาลงของสัญญาณ ALE ซึ่งอยู่ในคาบเวลาออสซิลเลเตอร์ที่ 15 ภายหลังจากที่ได้ทำการโอนสั่งย้ายข้อมูลมยังรีจีสเตอร์ SBUF หรือคำสั่งที่ทำให้ค่าแฟล็กสถานะ RI เป็นศูนย์ หลังจากนั้นสัญญาณนาฬิกาก็จะเปลี่ยนแปลงอีกครั้งราวช่วงใกล้เคียงกับเวลาขอบขาลงของสัญญาณ ALE ในคาบเวลาออสซิลเลเตอร์หลังจากนั้นอีก 6 คาบ แล้วจะดำเนินไปในลักษณะเช่นนี้จนกระทั่งข้อมูลทั้ง 8 บิต ได้ส่งออกไปเรียบร้อยแล้ว เมื่อสัญญาณขอบขาขึ้นของสัญญาณนาฬิกานี้เกิดขึ้นครบจำนวน 8 ครั้งแล้ว จึงจะมีผลทำให้แฟล็กสถานะ TI หรือ RI มีค่าเป็น 1 ขึ้น และสถานะของสัญญาณ TXD ก็จะเป็นระดับลอจิกสูงไปตลอด

ข้อมูลที่จะถูกส่งออกไปภายนอกจะถูกเลื่อนบิตนัยสำคัญต่ำออกไปก่อนเป็นลำดับแรก โดยจะเริ่มขึ้นในเวลาเริ่มต้นของคาบเวลาออกซิลเลเตอร์ ภายหลังจากที่ได้ทำคำสั่งการโอนย้ายข้อมูลมายังรีจิสเตอร์ SBUF สำหรับบิตแรกของข้อมูลที่รับมานั้นจะถูกแลตซ์ไว้ด้วยขอบขาขึ้นของสัญญาณนาฬิกาในคาบเวลาออสซิลเลเตอร์ที่ 24 ภายหลังจากที่ได้มีการกำหนดให้แฟล็กสถานะ RI เป็นค่าศูนย์ หลังจากนั้นในคาบเวลาออสซิลเลเตอร์อีก 12 คาบ ถัดมาก็จะได้รับบิตต่อไป ซึ่งจะดำเนินการในลักษณะเช่นนี้จนกระทั่ง ได้จำนวนบิตข้อมูลครบทั้ง 8 บิต

**2.2.7 พอร์ตอนุกรมโหมด 1**

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้นจำนวน 1 บิต บิตข้อมูลจำนวน 8 บิต และบิตสุดท้ายอีก 1 บิต ดังแสดงในรูป 2.7




**รูปที่ 2.7** รูปแบบของสัญญาณอนุกรมในโหมด 1 ซึ่งมีลักษณะเดียวกับรูปแบบของการสื่อสารแบบอะซิงโครนัสที่ใช้ข้อมูล 8 บิต บิตเริ่มต้นและบิตสุดท้ายอย่างละ 1 บิต

โดยข้อมูลจะถูกส่งออกไปทางขาสัญญาณ TXD และรับเข้ามาทางขาสัญญาณ RXD ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออกจะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรก และบิตสุดท้ายของข้อมูลที่รับเข้ามาจัดเก็บไว้ในบิต RBB ภายในรีจิสเตอร์ SCON สำหรับอัตราความเร็วในการส่งข้อมูลของโหมด 1 นั้น สามารถกำหนดเลือกได้

โหมดการทำงานนี้สามารถใช้ในการติดต่อกับพอร์ตสื่อสารอนุกรมแบบ RS-232C ของเครื่องคอมพิวเตอร์ทั่วไปได้ ดังนั้นในกรณีที่ใช้เป็นข้อมูล 7 บิตและไม่ใช้บิตที่ 8 เป็นบิตพาริตี้ก็อาจจะกำหนดค่าบิตนี้ให้เป็น 1 ซึ่งจะทำให้ด้านรับมองบิตนี้เป็นบิตสุดท้ายไป สำหรับกรณีที่ 8051 เป็นฝ่ายรับข้อมูลของระบบนี้ซึ่งมีเพียง 7 บิต ก็จะมองค่าของบิตสุดท้ายของข้อมูลที่รับมานี้เป็นค่าของข้อมูลบิตที่ 8 แทนและยังคงรอรับบิตสุดท้ายต่อไป อย่างไรก็ตามเนื่องว่าระดับสัญญาณ ของบิตสุดท้ายนี้จะเป็นระดับลอจิกสูงเช่นเดียวกับสภาวะของสายสื่อสารเมื่อไม่มีการส่งข้อมูล ดังนั้นระบบก็จะอ่านค่านี้เข้า ซึ่งก็ยังคงถือว่าถูกต้องตามหลักการโดยปริยาย

การส่งข้อมูลจะเกิดขึ้นภายหลังจากที่ได้มีการเขียนหรือโอนย้ายข้อมูลเข้าไปยังรีจีสเตอร์ SBUF โดยผู้เขียนโปรแกรมจะต้องทำการตรวจสอบค่าของแฟล็กสถานะ TI ภายในรีจีสเตอร์ SCON ซึ่งจะมีค่าเป็น 1 ภายหลังที่ข้อมูลได้เลื่อนบิตออกไปภายนอกแล้ว สำหรับการรับข้อมูลจะเริ่มต้นขึ้นเมื่อได้มีการกำหนดค่า 1 ให้กับบิต REN และมีการเปลี่ยนแปลงระดับสัญญาณที่ขาสัญญาณ RxD เกิดขึ้นการสุ่มอ่านค่าบิตข้อมูลเข้ามาจะใช้อัตราเดียวกับอัตราบอดที่ได้กำหนดไว้ในราวช่วงกลางคาบเวลาของบิตหลังจากที่ได้รับข้อมูลครบจำนวน 10 บิตแล้ว และหากมีสภาวะดังในตารางต่อไปนี้เกิดขึ้นก็จะมีผลให้เกิดการย้ายข้อมูลไปเก็บยังรีจิสเตอร์ SBUF

**ตารางที่ 2.6** แฟล็กสถานะของรีจีสเตอร์ในการรับข้อมูล

| | |
|---|---|
| 1 | แฟล็ก RI มีค่าเป็น 0 (แสดงว่าได้มีการอ่านไบต์ของข้อมูลเข้ามาแล้ว และพร้อมที่จะรับข้อมูลถัดไป) |
| 2 | ค่าของบิตสุดท้ายเป็น 1 (แสดงว่าข้อมูลที่ได้รับข้อมูลนั้นถูกต้อง จึงได้โอนย้ายไปเก็บยังรีจีสเตอร์ SBUF โดยไม่สนใจค่าของบิต SM2) |

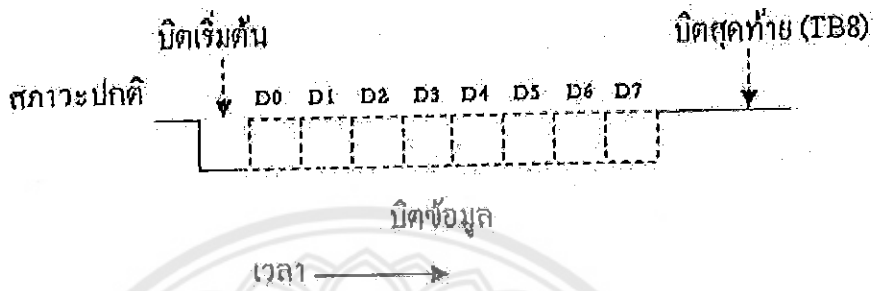
ดังนั้นสรุปได้ว่าข้อมูลที่รับเข้ามาจำนวน 10 บิตนั้น ส่วนของบิตเริ่มต้นไม่มีการนำไปใช้งานอีกต่อไป บิตข้อมูลจำนวน 8 บิตนั้นจะถูกย้ายไปเก็บยังรีจีสเตอร์ SBUF และส่วนของบิตสุดท้ายจะถูกนำไปเก็บในตำแหน่งของบิต RB8 ภายในรีจีสเตอร์ SCON นอกจากนี้ยังมีแฟล็กสถานะ RI ซึ่งจะเป็นค่าหนึ่ง เพื่อบอกสภาวะว่าได้มีการรับข้อมูลใหม่เข้ามาแล้ว ในกรณีที่โปรแกรมได้อ่านข้อมูลจากรีจีสเตอร์ SBUF แล้วแต่ไม่ได้กำหนดบิต RI ให้เป็นค่าศูนย์อีกครั้ง ข้อมูลที่รับเข้ามาใหม่หลังจากนั้นสูญหายไป

**ตารางที่**2.7 ตารางแสดงอัตราบอดมาตรฐานที่นิยมใช้กันในการสื่อสารอนุกรม โดยการกำหนดค่าตัวแปรต่างๆสำหรับการเกิดอัตราบอดโดยใช้ Timer 1

| อัตราบอด (โหมด1 และ 3 ) | ความถี่ออสซิลเลเตอร์(MHz) | บิต SMOD | C/T\ | MODE | Reload Value |
|---|---|---|---|---|---|
| 65.5K | 12 | 1 | 0 | 2 | FDh |
| 19.2K | 11.059 | 1 | 0 | 2 | FDh |
| 9600 | 11.059 | 0 | 0 | 2 | FDh |
| 4800 | 11.059 | 0 | 0 | 2 | Fah |
| 2400 | 11.059 | 0 | 0 | 2 | F4h |
| 1200 | 11.059 | 0 | 0 | 2 | E8h |
| 1375 | 11.059 | 0 | 0 | 2 | 1Dh |
| 110 | 6 | 0 | 0 | 2 | 72h |
| 110 | 12 | 0 | 0 | 1 | FEEBh |

**2.2.8 พอร์ตอนุกรมโหมด 2 และ 3**

การทำงานโหมด 2 หรือโหมด3 ของพอร์ตอนุกรมจะทำการรับส่งข้อมูล 11 บิต เช่นเดียวกัน ซึ่งประกอบด้วย บิตเริ่มต้น บิตข้อมูลจำนวน 8 บิต บิตข้อมูลบิตที่9 และบิตสุดท้าย ดังแสดงในรูปที่ 2.8 แต่สำหรับโหมด 3 จะสามารถเปลี่ยนแปลงอัตราการส่งข้อมูลได้ไม่ได้ถูกกำหนดไว้คงที่เช่นในโหมด 2




**รูปที่2.8** รูปแบบของสัญญาณข้อมูลอนุกรมในโหมด 2 ซึ่งมีลักษณะรูปแบบการสื่อสารที่ใช้บิตข้อมูลจำนวน 9 บิต บิตเริ่มต้นและบิตสุดท้ายอย่างละ 1 บิต

อัตราบอดโหมด 2

ความถี่อัตราบอด = 2 SMOD / 32X ความถี่ออสซิลเลเตอร์

อัตราบอดโหมด 3

ใช้อัตราการโอเวอร์โฟลว์ของ Timer1 หรือ Timer2 โดยการคำนวณอัตราบอดเช่นเดียวกับของพอร์ตอนุกรมที่ทำงานในโหมด 1

การส่งข้อมูลอนุกรมในโหมด 2 และ 3 จะต้องนำค่าข้อมูลนั้นไปเก็บยังรีจิสเตอร์ SBUF สำหรับค่าของบิตที่9 ที่เพิ่มขึ้นนั้นนำมาจากค่าของบิต TB8 ภายในรีจีสเตอร์ SCON ซึ่งจะต้องได้รับการกำหนดค่าจากผู้ใช้งาน เมื่อข้อมูลถูกเลื่อนบิตออกไปภายนอกเรียบร้อยแล้ว แฟล็กสถานะ TI จะมีค่าเป็น 1 เช่นเดียวกับโหมดอื่น ๆที่ผ่านมา และผู้ใช้จะต้องทำการเปลี่ยนกลับให้เป็นค่า 0 ตามเดิม สำหรับการรับข้อมูลจะถูกนำมาเก็บไว้ภายในรีจีสเตอร์ SBUF เช่นเดียวกันโดยค่าของบิตที่ 9 จะนำไปเก็บไว้ยังบิต RB8 ภายในรีจีสเตอร์ SCON

**2.2.9 เรียลไทม์ชิพ ET-RTC V4.0**

การนำไมโครโปรเซสเซอร์ไปใช้ในงานที่มีความเกี่ยวข้องกับเรื่องของเวลานั้น RTC เหมาะที่จะใช้งานลักษณะนี้เพราะสามารถบอกได้ทั้ง วัน เดือน ปี วันในรอบสัปดาห์ ชั่วโมง นาที วินาที ในการติดต่อกับ RTC นั้นเปรียบเสมือนกับการติดต่อ พอร์ตอินพุตเอาท์พุต คือ เราสามารถเขียนข้อมูลเกี่ยวกับเวลาไปที่เบอร์พอร์ตของ RTC ตัวนับเวลาภายในก็จะเดินตามเวลาที่เราตั้งให้ และเราก็สามารถอ่านข้อมูลจาก RTC ได้เช่นกัน

MSM 6242B เป็นไอซี REALTIME CLOCK /CALENDAR ชนิด CMOS ใช้ต่อกับบัสของ MICROPROCESSOR/MICROCOMPTER ได้โดยตรง มี ADDRESSBUS และ DATA BUS ขนาด 4 บิต มี CONTROL REGISTER ขนาด 4 บิต 3 ตัว คือ CD,CE,CF MSM6242B โดยปกติจะทำงานที่ 5V +10% ที่-30 ถึง 25 C มีแพ็คเก็ต 3แบบ คือ 18 PIN PLASTIC DIP, 24 PIN PLASTIV FLAT PACKAGE และแบบ PIN PLCC PACKAGE

<u>การจัดขาและหน้าที่ของขาต่าง ๆ</u>

MSM 6242B ได้ถูกออกแบบมาให้อินเตอร์เฟสเข้ากับ CPU ตระกูล 8085 ,MCS 48 และ Z80 ได้ด้วย สำหรับหน้าที่ของขาต่าง ๆ มีดังนี้

D0-D3 (DATA BUS) เป็นบัสข้อมูลอินพุท/เอาท์พุท สามารถต่อเข้ากับบัสของ MICROCON TROLLER ได้โดยตรงใช้ในการอ่านและ เขียนข้อมูลของรีจีสเตอร์ภายในที่เป็น นาฬิกา/ปฏิทิน และ รีจีสเตอร์ควบคุมโดย D0 = LSB และ D3 = MSB

A0-A3 (ADDRESS BUS) เป็นบัสแอดเดรสสำหรับติดต่อกับรีจีสเตอร์ภายในของ RTC เพื่อที่จะเขียนหรืออ่านข้อมูลในตำแหน่งนั้น A0-A3 จะใช้ร่วมกับ ALE สำหรับการอ้างตำแหน่งรีจีสเตอร์

ALE (ADDRESS LATCH ENABLE) เมื่อ CSO = 0 และ ALE เปลี่ยนจากลอจิก "1" ไปเป็นลอจิก "0" แอดเดรสจะถูกแลตซ์เอาไว้ภายในตัวของ RTC MICROCONTROLER /MICROPROCESSORS ที่มีขา ALE เป็นเอาท์พุตควรจะต่อเข้ากับขา ALE ของ RTC ด้วยแต่ถ้าไม่มีขา ALE ให้ต่อขา ALE ของ MSM 6242B เข้ากับ VDD

WR (WRITE ENABLE) ใช้เขียนข้อมูลเข้าไปในรีจีสเตอร์ของ RTC แอคตีฟที่ลอจิก "0" โดยที่ CS1 = 1 และ CS0 = 0

RD (READ ENABLE) ใช้อ่านข้อมูลจากรีจีสเตอร์ของ RTC แอคตีฟที่ลอจิก "0" โดยที่ CS1 = 1 และ CS0 = 1 และ RD กับ WR จะต้องไม่แอคตีฟพร้อมกัน

CS0,CS1 (CHIP SELECTS 0,1) เป็น CHIP SELECT ทำหน้าที่ ENABLE/ DISABLE การทำงานของ ALE,RD และ WR โดยที่ CSO และ ALE จะทำงานร่วมกันส่วน CS1 กับ ALE จะทำงานแยกกัน

STD.P (STAND PULSE OUTPUT) เป็นขาเอาซ์พุทชนิด N-CH OPEN DRAIN ใช้ต่อเข้ากับขา INTERRUPT ของ CPU

XT (XTAL OSILLATOR INPUT /OUTPUT) ต่อเข้ากับตัวคริสตอล 32.768 KHz ถ้าต้องการป้อนความถี่จากภายนอก 32.768 KHz ทำได้โดยป้อนความถี่เข้าที่ขา XT ถ้าความถี่มาจากเอาท์พุตของไอซี TTL ควรต่อ R PULL~UP ไว้ด้วย ส่วนขา XT ควรปล่อยลอยไว้

VDD เป็นขา POWER SUPPLY +2~+6V

GND ขา GROUND

ต 9 ส.ค. 2547 ห้องสมุดคณะวิศวกรรมศาสตร์ 4700185

ว 50667 ศ.2 ส372ค 2546

ลักษณะของ RTC BOARD

RTC V4.0 เป็น BOARD ขนาด 4.0 cm X 5.5 cm มีCONNECTOR สำหรับ เสียบบน ET-BOARD ทางด้าน 40 PIN CONNECTOR ของ Z80 และยังมี CONNECTOR อีกชุดเพื่อขยายระบบออกไปใช้งานร่วมกับระบบอื่นได้อีกบนบอร์ดจะมี BATTERY ขนาด 3.6 VDC ทำการ BACK UP ข้อมูลในกรณีที่ไฟดับและยังทำให้นาฬิกาเดินตามปกติไม่มีผลกระทบเมื่อไฟดับ และถ้าต้องการเปลี่ยนแบตเตอรี่เป็นชนิดนิเกิลแคดเมี่ยมก็สามารถทำได้โดยใส่ R330 ลงไปตำแหน่งที่พิมพ์ว่า NIC ซึ่งจะอยู่ใต้ BATTERY

การ DECODE PORT เราใช้ ไอซีเบอร์ 74HCT138 ในการ DECODE PORT โดยมีเบอร์พอร์ตให้เลือก 8 เบอร์คือ 00H,20H,40H,60H,80H,A0H,C0H,E0H สามารถเลือกเบอร์ได้โดยการใช้ JUMPER

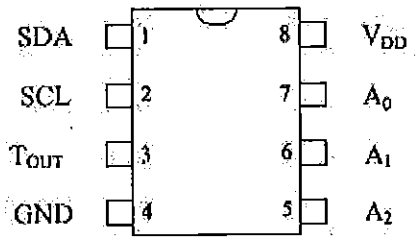
การทำงานร่วมกับ ET-BOARD

นำ RTC บอร์ดมาเสียบลงทางด้าน 40 PIN CONNECTOR ของ Z80 บน ET-BOARD ซึ่งเราสามารถเขียนโปรแกรมให้ RTC แสดงเวลาทำเป็นนาฬิกาหรือแสดง วัน เดือน ปี วันในรอบสัปดาห์ก็ได้ หรือให้มีการตั้งปลุกโดยนำไปเทียบกับเวลาที่ตั้งไว้ซึ่งเก็บไว้ในหน่วยความจำ แล้วให้เอาท์พุตออกไปที่เบอร์พอร์ตของ 8255 ที่ขยายออกมาใน ET-BOARD ก็ได้แล้วแต่ผู้ใช้จะเขียนโปรแกรม

## 2.3 ไอซีวัดอุณหภูมิแบบดิจิตอล DS1621

เป็นไอซีวัดอุณหภูมิที่ใช้ในการติดต่อผ่านระบบบัส $I^2C$ สำหรับคุณสมบัติที่สำคัญของ DS1621 มีดังนี้

- สามารถวัดอุณหภูมิได้ตั้งแต่ -55°C ถึง +125°C โดยมีความละเอียดในการวัด 0.5°C เทียบเป็นหน่วยของฟาเรนไฮต์ได้เท่ากับ -67°F ถึง +257°F ความละเอียด 0.9°F โดยไม่ต้องใช้อุปกรณ์ต่อร่วมภายนอก
- ความละเอียดของข้อมูลอุณหภูมิดิจิตอล 9 บิต
- ใช้ไฟเลี้ยงได้ตั้งแต่ 2.7 - 5.5 V
- ใช้เวลาในการวัดอุณหภูมิแล้วแปลงเป็นข้อมูลดิจิตอล 1 วินาที
- สามารถทำงานเป็นเทอร์โมสตัต (thermostat) ได้ พร้อมขาเอาต์พุต 1 ขา
- สามารถตั้งค่าอุณหภูมิที่ต้องการเมื่อทำงานเป็นเทอร์โมสตัตได้ โดยติดต่อผ่านระบบบัส $I^2C$ และค่าที่กำหนดนี้จะคงอยู่ตลอดไป แม้ปลดไฟเลี้ยงแล้วก็ตาม สามารถเปลี่ยนแปลงได้โดยการกำหนดทางซอฟต์แวร์เท่านั้น
- สามารถต่อพ่วงกันได้สูงสุด 8 ตัว




**รูปที่ 2.9** แสดงรายละเอียดของไอซีวัดอุณหภูมิ DS1621

**2.3.1 การทำงานของ DS1621**

เมื่อ DS1621 เริ่มทำงาน จะทำการวัดอุณหภูมิ โดยภายในไอซีจะมีตัวตรวจจับอุณหภูมิบรรจุอยู่ ผลการวัดจะส่งต่อไปยังส่วนจัดการแอดเดรสและควบคุมอินพุตเอาท์พุต ในส่วนนี้นำข้อมูลที่วัดได้ส่งต่อไปยังบัสผ่านขา SDA โดยได้รับการกำหนดจังหวะการทำงานจากส่วนรีจิสเตอร์ควบคุมและแสดงสถานะ เนื่องจาก DS1621 มีขากำหนดแอดเดรส 3 ขา คือ A2-A0 จึงทำให้สามารถต่อพ่วง DS1621 ได้สูงสุด 8 ตัว โดยแต่ละตัวจะต้องกำหนดสถานะลอจิกที่ขาแอดเดรสทั้ง 3 ให้แตกต่างกัน ส่วนจัดการแอดเดรสและควบคุมอินพุตเอาท์พุตของ DS1621 แต่ละตัวเป็นตัวจัดการส่งข้อมูลตามที่อุปกรณ์มาสเตอร์ร้องขอ

ค่าของอุณหภูมิที่วัดได้ถูกเก็บอยู่ในรูปของข้อมูลเลขฐานสองขนาด 9 บิต ข้อมูลดิจิตอลเอาท์พุต จะแบ่งเป็น 2 ส่วน 8 บิตบนหรือไบต์สูงจะเป็นค่าอุณหภูมิจำนวนเต็ม โดยบิต MSB จะเป็นตัวแสดงเครื่องหมายว่า อุณหภูมิเป็นบวกหรือลบ หากบิต MSB เป็น "1" แสดงว่าอุณหภูมิที่วัดได้มีค่าเป็นลบ ในขณะที่ 8 บิตล่างหรือไบต์ล่างจะมีใช้งานเพียงบิตเดียวจะเป็นบิตแสดงค่าความละเอียด ซึ่งเท่ากับ $0.5^{\circ}C$

เมื่อสั่งให้ DS1621 ส่งข้อมูลอุณหภูมิออกมา DS1621 จะใช้เวลาไม่เกิน 1 วินาที จึงจะส่งข้อมูลออกมา ดังนั้นในการควบคุมการทำงานจึงต้องมีการหน่วงเวลา เพื่อรอให้ DS1621 วัดและแปลงค่าออกมาเป็นข้อมูลให้เรียบร้อยเสียก่อน

**2.3.2 การทำงานในโหมดเทอร์โมสตัต**

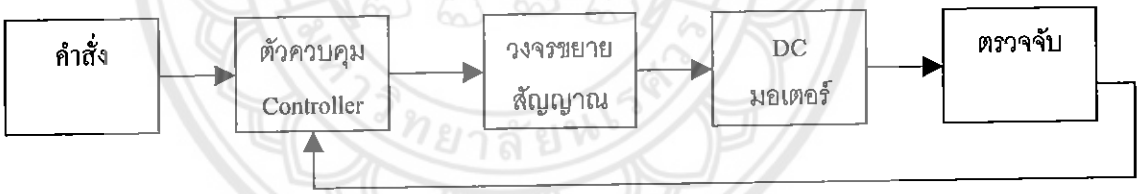
นอกจากวัดอุณหภูมิแล้ว DS1621 ยังสามารถทำงานในแบบเทอร์โมสตัตหรือให้เอาต์พุตเมื่ออุณหภูมิต่ำหรือสูงถึงจุดที่กำหนดไว้โดยผู้ใช้สามารถโปรแกรมได้ตามต้องการและค่าที่โปรแกรมนี้จะอยู่ตลอดไปจนกว่าจะมีการแก้ไขแม้ว่าจะตัดไฟเลี้ยงไปแล้วก็ตาม ผู้ใช้งานต้องกำหนดค่าของจุดทริกอุณหภูมิสูง (high temperature trigger : TH) และ จุดทริกอุณหภูมิต่ำ (low temperature trigger : TL) และเลือกสัญญาณแอกตีฟว่า เมื่ออุณหภูมิสูงมาถึงจุดที่กำหนดจะให้ขาเอาต์พุต Tout ส่งสัญญาณลอจิกแบบใดออกไป สามารถเลือกได้ 2 แบบคือ ลอจิก "0" หรือ "1"

เมื่อกำหนดค่าเรียบร้อยแล้ว DS1621 จะวัดอุณหภูมิไปเรื่อย ๆ จนกระทั่งเมื่ออุณหภูมิสูงจนถึงจุด TH ขา Tout ก็จะแอกตีฟทันที และแอกตีฟค้างอยู่เช่นนั้นจนกว่าอุณหภูมิที่วัดได้ลดต่ำลงกว่าจุด

TL ดังนั้นการกำหนดค่าของ TH และ TL จึงสำคัญมาก หากกำหนดไส้แคบหรือมีความแตกต่างกันน้อยเกินไป และอุณหภูมิที่วัดจริงนั้นมีการเปลี่ยนแปลงในช่วงแคบ ๆ หรือจริง ๆ แล้วค่อนข้างคงที่ จะทำให้เอาพุตที่ขา Tout เกิดสัญญาณลอจิกที่ไม่คงที่ จนคล้ายกับเป็นสัญญาณรบกวนส่งผลให้การนำการนำสัญญาณไปใช้งานทำได้ไม่ดี แต่ถ้าหากกำหนดไว้ห่างหรือมีความแตกต่างมากเกินไปอาจทำให้มองได้ว่า วงจรเทอร์โมสตัตนี้มีความไวในการทำงานต่ำ เพราะว่าจะตัดเอาท์พุตต้องรอให้อุณหภูมิลดค่าลงต่ำกว่าจุด TH มาก ๆ ซึ่งอาจส่งผลเสียหากการลดลงของอุณหภูมิที่เกิดขึ้นจริงลงไม่ถึงจุด TL แต่อุปกรณ์ที่ต้องการนำสัญญาณนี้ไปใช้มีความสามารถในการทนอุณหภูมิในช่วงระหว่าง TH และ TL ได้ไม่นาน ก็จะส่งผลให้เกิดความเสียหายต่ออุปกรณ์นั้น ๆ ได้

## 2.4 มอเตอร์กระแสตรงและการควบคุม [3]

มอเตอร์แบบกระแสตรง(DC Motor) ในปัจจุบันสามารถจำแนกได้หลายประเภทขึ้นอยู่แบวิธีการสร้างการควบคุม สำหรับการหมุนของมอเตอร์กระแสตรง (Direct Current) จ่ายให้กับขดลวดสนามไฟฟ้า หรือขดลวดผลิตแรงเคลื่อนไฟฟ้า (Motor armature) ภายนอก สำหรับแรงบิตของมอเตอร์จะขึ้นอยู่กับเส้นแรงแม่เหล็ก ถ้าหากเส้นแรงแม่เหล็กมากจะทำให้มีแรงบิตสูง มอเตอร์กระแสตรงส่วนใหญ่ถ้าหมุนด้วยความเร็วสูงจะมีแรงบิดต่ำ ถ้าหมุนด้วยความเร็วต่ำจะมีแรงบิดสูง ระบบควบคุมมอเตอร์พื้นฐานสามารถเขียนได้ตามไดอะแกรมดังรูป




รูปที่2.10 ไดอะแกรมแสดงระบบควบคุมมอเตอร์พื้นฐาน

*คำสั่ง* จะเป็นการสั่งว่าจะให้มอเตอร์ทำงานอย่างไร หมุนอย่างไร ความเร็วอย่างไร

*ตัวควบคุม (Controller)* เป็นตัวกำเนิดสัญญาณควบคุมที่จ่ายให้กับมอเตอร์

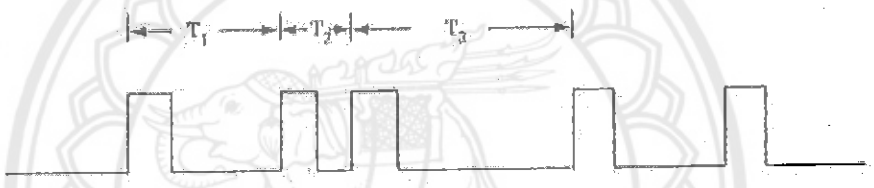
*วงจรขยายกำลัง(Power Amplifier)*จะนำสัญญาณจากตัวควบคุมมาขยายให้แรงขึ้น เพื่อให้ได้ระดับสัญญาณที่พอเหมาะสำหรับมอเตอร์แต่ละรุ่น

*ตัวตรวจจับ(Encoder)* จะประกอบด้วยทรานสดิวเซอร์ เพื่อตรวจดูการหมุนของมอเตอร์ว่าเป็นอย่างไร เป็นไปตามที่ต้องการหรือไม่ โดยจะเปรียบเทียบกับสัญญาณอ้างอิงระดับหนึ่ง และส่งสัญญาณนี้กลับมาให้ตัวควบคุมอีกทีหนึ่ง

### 2.4.1 การควบคุมความเร็ว

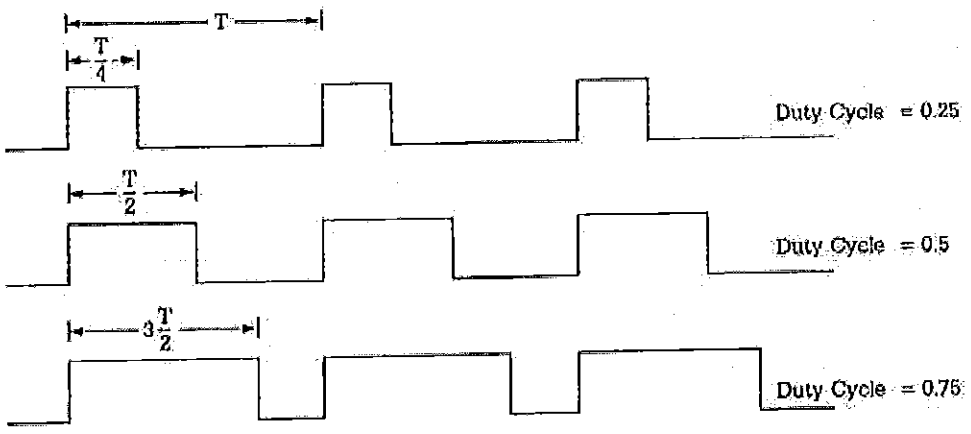
โดยทั่วไปแล้วเมื่อเริ่มจ่ายกระแสไฟฟ้าให้กับตัวมอเตอร์กระแสตรง ตัวมันจะหมุนด้วยความเร็วสูงสุดที่สามารถหมุนได้ด้วยแรงดันไฟฟ้าค่านั้น การควบคุมความเร็วจะทำโดยควบคุมแรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์ แต่การทำงานแบบนี้จะต้องจ่ายแรงดันไฟฟ้าให้กับมอเตอร์ตลอดเวลา ซึ่งจะทำให้เกิดการสูญเสียกำลังไฟฟ้า(power consumptions)และจะมีความร้อนเกิดขึ้นตลอดเวลา ดังนั้นจึงมีการใช้วิธีการจ่ายแรงดันเป็นช่วงๆให้กับมอเตอร์แทน ถ้าหากช่วงการจ่ายแรงดันและหยุดจ่ายแรงดันแตกต่างกันไปก็จะทำให้ความเร็วของการหมุนมอเตอร์ต่างกันไปด้วย การควบคุมความเร็วด้วยวิธีนี้มีการใช้งานอยู่ 2 ลักษณะคือ แบบ pulse-frequency modulation(PFM) และแบบ pulse-width modulation

1. แบบ pulse-frequency modulation หรือ PFM การทำงานแบบนี้จะควบคุมการจ่ายแรงดันและหยุดการจ่ายแรงดันเป็นช่วง ดังรูป 2.11 การทำงานแบบนี้จะทำให้ระดับแรงดันไฟฟ้าที่มอเตอร์ได้รับมีค่าแตกต่างกันไป ความเร็วของการหมุนมอเตอร์จะขึ้นกับอัตราการเปิดปิดมอเตอร์นี้




**รูปที่ 2.11** สัญญาณควบคุมมอเตอร์ด้วยวิธี PFM

2. แบบ pulse-width modulation หรือ PWM การควบคุมมอเตอร์ในแบบ PWM ค่าความถี่ของสัญญาณควบคุมจะไม่แน่นอน แต่การควบคุมแบบ PWM นี้จะใช้ความถี่ในการควบคุมที่คงที่ แต่จะปรับแรงดันที่จ่ายให้กับมอเตอร์ โดยวิธีการปรับค่าดิวตี้ไซเคิล (Duty cycle) ดังแสดงในรูป 2.12 แต่วิธีนี้จะใช้กับความถี่สูงมากๆไม่ดี




**รูปที่ 2.12** สัญญาณควบคุมมอเตอร์แบบ PWM

จากวิธีการควบคุมมอเตอร์ข้างต้นจะเห็นว่าเราสามารถสร้างสัญญาณพัลส์ลักษณะนี้จากไมโครคอนโทรลเลอร์ได้ โดยใช้สัญญาณเอาท์พุตเพียงบิตเดียว

**2.4.2 หลักการทำงานของไดรเวอร์มอเตอร์(Driver Motor)**

ไดรเวอร์มอเตอร์ (Driver Motor) มีหน้าที่รับอินพุตจาก คอนโทรลเลอร์ซึ่งประกอบด้วยสัญญาณพัลส์ (Pulse Width Modulation, PWM) ทิศทาง (Direction , Dir) และ สัญญาณปิด – เปิด (Enable,EN) เมื่อได้รับสัญญาณอินพุตดังกล่าวแล้ว ก็จะควบคุมกระแสไฟฟ้าที่จะนำไปขับมอเตอร์ตามคำสั่ง

จากวงจรไดรเวอร์มอเตอร์จะมีการไอโซเลต (Isolate) ด้วยออปโต้ไอโซเลต (Opto isolate) ทั้ง 3 อินพุต เพื่อทำการแยกผ่านสัญญาณจากบอร์ดคอนโทรลเลอร์กับบอร์ดไดรเวอร์มอเตอร์ ซึ่งสามารถป้องกันสัญญาณรบกวนจากมอเตอร์ไปรบกวนบอร์ดคอนโทรลเลอร์และบอร์ดอื่นๆได้

กระแสไฟฟ้าอินพุต จะไหลผ่านไดโอดเปล่งแสงภายในออปโต้ไอโซเลต เมื่อไดโอดเปล่งแสงสว่างจะทำให้เกิดกระแสปริมาณเล็กน้อย ไหลออกจากออปโต้ทรานซิสเตอร์ผ่านความต้านทานที่มีค่าสูง จะทำให้เกิดความต่างศักย์ขึ้น แล้วนำความต่างศักย์ดังกล่าวไปเปรียบเทียบด้วย ออปแอมป์ โดยจะต้องปรับค่าความต้านทานออปเซต (R-Opset) ให้มีแรงดันประมาณครึ่งหนึ่งของแรงดันที่คร่อมตัวต้านทาน สัญญาณที่ได้จากออปแอมป์จะเป็นสัญญาณลอจิก โดนสัญญาณทั้ง 3 จะไปควบคุมการทำงานของไอซีถอดรหัส (74LS138) ซึ่งสัญญาณทิศทางจะเข้าที่ขา 1(A) ของไอซี สัญญาณพัลส์เข้าที่ขา 6 (GI) และสัญญาณเปิด-เปิดเข้าที่ขา 4 (G2A)

ผลของสัญญาณทิศทางจะสามารถทำให้ขา 15(Y0) และขา 14(Y1) ทำงานสับกัน ขึ้นอยู่กับลอจิกของสัญญาณ สำหรับสัญญาณพัลส์จะทำให้เอาท์พุตของไอซีทำงาน และไม่ทำงานเป็นจังหวะตามลักษณะพัลส์ ซึ่งจะมีผลต่อการควบคุมกระแสเฉลี่ยที่นำไปขับมอเตอร์ ส่วนสัญญาณปิด-เปิดนั้น หากมีลอจิก '1' จะทำให้ไดรเวอร์ไม่ทำงาน นั่นคือ ไม่มีกระแสไปขับมอเตอร์ แต่ถ้ามีลอจิก '0'ก็จะทำงาน

สำหรับวงจรที่ใช้ในการจ่ายกระแสให้มอเตอร์จะใช้มอตเฟต (MOSFET) ต่อกันแบบ เอช-บริดช์ (H-bridge) ขณะทำงานจะสั่งให้มอตเฟตทำงาน (ON) ทีละ 2 ตัวทะแยงมุมกันเพื่อให้กระแสไหลผ่านมอเตอร์ ซึ่งวงจรได้ออกแบบไม่ให้มอตเฟตข้างเดียวกันทำงานพร้อมกัน การควบคุมการทำงานของมอสเฟตทั้ง 4 ตัวจะรับสัญญาณมาจากไอซรถอดรหัส (74LS138) ขา15(Y0) และขา 14(Y1) ซึ่งปกติแล้วจะทำงานไม่พร้อมกัน โดยจะควบคุมมอสเฟตไขว้กัน

สัญญาณที่ควบคุมมอสเฟตชนิดพี(P-Channel) จะผ่านบัฟเฟอร์(7407) นั่นคือหากอินพุตเป็นลอจิก '0' เอาท์พุตของบัฟเฟอร์จะต่อลงกราวด์ ทำให้มีกระแสไหลผ่านลงกราวด์ ทำให้มีกระแสไหลผ่านตัวต้านทานเกิดแรงดันตกคร่อม $V_{GS}$ ประมาณ –5V ซึ่งจะทำให้มอสเฟตทำงาน หากเป็นลอจิก '1' ก็จะไม่ทำงานเนื่องจาก $V_{GS}$ เท่ากับ 0 V

สัญญาณที่ควบคุมมอสเฟตชนิดเอ็น (N-Channel) จะผ่านนอตเกต(74LS04) และเข้าบัฟเฟอร์ (7407) โดยหากอินพุตของนอตเกตเป็นลอจิก '0' ก็จะทำให้เอาท์พุตของบัฟเฟอร์มีความต้านทานสูงมาก (High- Z) ทำให้เกิดกระแสไหลผ่านตัวต้านทาน เกิดแรงดันตกคร่อม $V_{GS}$ ประมาณ 5 V ทำให้มอสเฟตทำงาน หากเป็นลอจิก '1' ก็จะไม่ทำงาน

ก่อนที่กระแสไฟฟ้าที่ขับมอเตอร์จะต่อลงกราวด์ จะมีการต่อตัวต้านทานที่มีค่าน้อยมากและมีวัตต์สูงๆมาคั่นก่อน เพื่อนำโวลต์ที่เกิดขึ้นไปตรวจสอบปริมาณกระแสไฟฟ้าที่ไหลผ่านมอเตอร์เพื่อป้องกันอันตรายจากกระแสโอเวอร์โหลด(Overload) โดยการใช้ตัวต้านทานและตัวเก็บประจุค่าน้อยๆ ทำการลดสัญญาณรบกวน และนำไปเปรียบเทียบกับโวลต์จากตัวต้านทานปรับค่าได้ด้วยออปแอมป์ แล้วนำสัญญาณที่ได้ไปเข้าไอซีถอดรหัส (74LS138) ขา 2(B) ซึ่งหากออปแอมป์ให้ลอจิก '1' จะทำให้เอาท์พุตของไอซีถอดรหัส (74LS138) ทำงานที่ขา 13(Y2) หรือ ขา 14(Y3) แทน ซึ่งเป็นผลให้มอสเฟตไม่จ่ายกระแสให้มอเตอร์ เป็นการรักษาความปลอดภัยขั้นหนึ่ง นอกจากนั้นสัญญาณที่ออกจากออปแอมป์ดังกล่าว ยังถูกขยายความกว้างของพัลส์ที่เกิดจากกระแสเกินเพื่อขับไดโอดเปล่งแสงแสดงให้ผู้ใช้เห็นชัดเจนขึ้น โดยไอซีโมโนสเตเบิล(74LS123) ซึ่งจะทำให้ไดโอดเปล่งแสงสว่างนานขึ้น

# บทที่ 3
# การออกแบบเครื่องฟักไข่

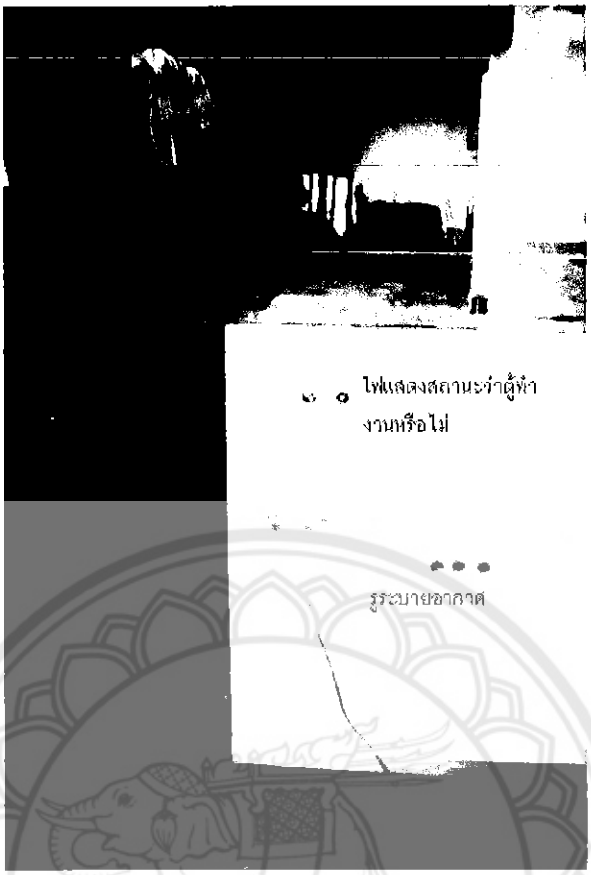
## 3.1 เครื่องฟักไข่ที่ใช้กันอยู่ในปัจจุบัน

### 3.1.1 ขนาดและโครงสร้างของตู้ฟักไข่

- ตู้ฟักไข่เป็นตู้ขนาดเล็ก เป็นโครงสี่เหลี่ยม ขนาดสูง 50 เซนติเมตร กว้าง 50 เซนติเมตร และยาว 100 เซนติเมตร
- โครงสร้างของตู้ประกอบโครงขึ้นมาด้วยโครงเหล็กเชื่อมติดกัน
- ผนังของตู้ทำด้วยไม้อัดขนาดหนาประมาณ 0.5 เซนติเมตร ที่ผนัง 1 ด้านมีรูระบายอากาศ
- มีแผงไข่เป็นชั้นๆเพื่อวางไข่จำนวน 2 ชั้น 1 ถาดวางได้ประมาณ 25-30 ฟองสามารถบรรจุไข่ได้มากสุดประมาณ 50 ฟอง

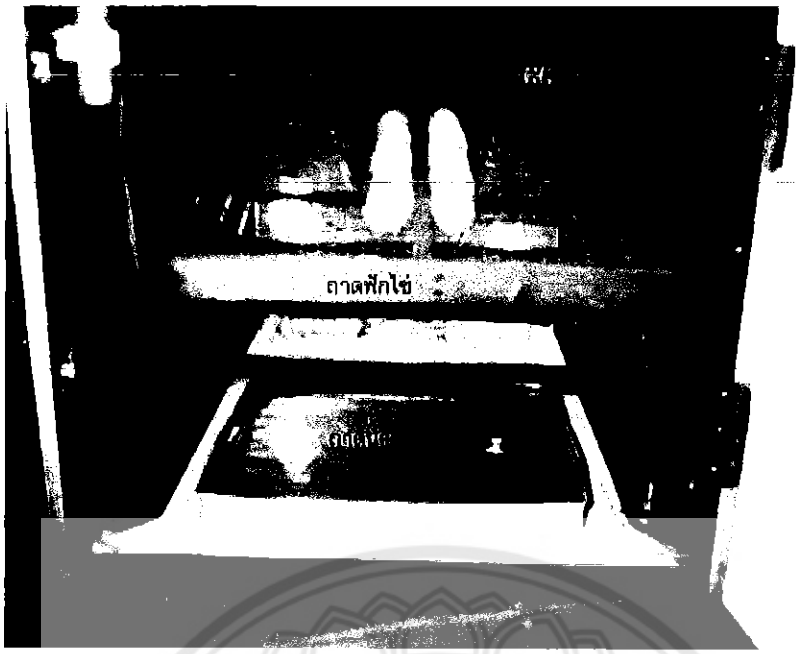
### 3.1.2 หลักการทำงานของตู้ฟักไข่

- ระบบให้ความร้อนของตู้ทำโดยการใช้หลอดไฟเพื่อให้ความร้อน
- ควบคุมอุณหภูมิโดยใช้หลักการของเทอร์โมสตัสเหมือนในเตารีดโดยตั้งอุณหภูมิที่ต้องการไว้ซึ่งจะต้องเทียบกับเทอร์โมมิเตอร์ด้วยตัวผู้ใช้เอง
- มีพัดลมเป็นตัวกระจายความร้อนโดยการพัดลมร้อนจากหลอดไฟเข้ามาหาแผงไข่โดยตรง
- การกลับไข่จะใช้จะใช้คันโยกซึ่งเชื่อมติดกับแผงวางไข่ เมื่อดึงคันโยก ก็จะโยกให้แผงไข่พลิกกลับมาเป็นแนว 45 องศา
- การกลับไข่ต้องให้ผู้ใช้มากลับแผงไข่เอง
- การให้ความชื้นกับไข่ ใช้การระเหยของน้ำในถาดน้ำที่จัดไว้




**รูปที่ 3.1** รูระบายอากาศในเครื่องฟักไข่ในปัจจุบัน

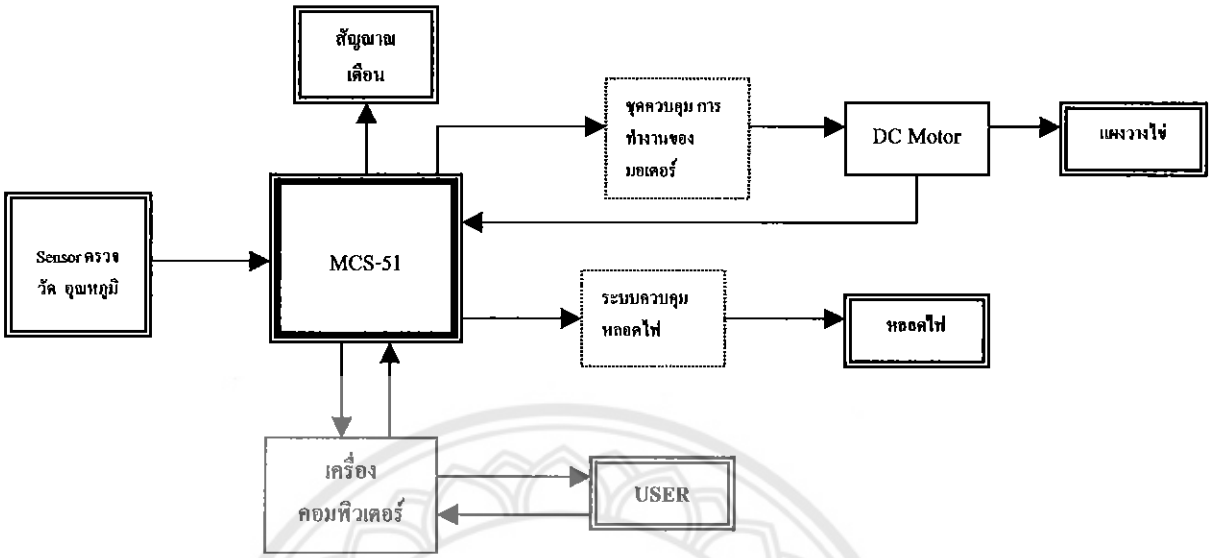
**รูปที่ 3.2** คันโยกปรับระดับในเครื่องฟักไข่ในปัจจุบัน




**รูปที่ 3.3** ลักษณะภายในเครื่องฟักไข่ในปัจจุบัน

**รูปที่ 3.4** เทอร์โมมิเตอร์ด้านหน้าของเครื่องฟักไข่ในปัจจุบัน

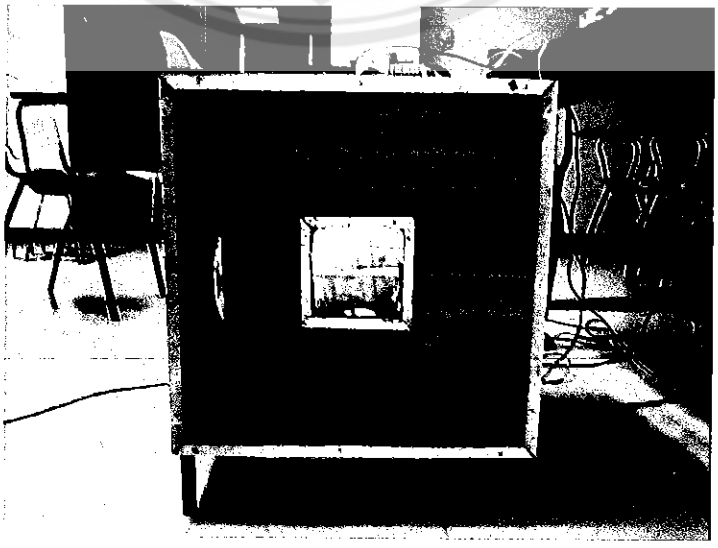
## 3.2 เครื่องฟักไข่ที่ทำการออกแบบ



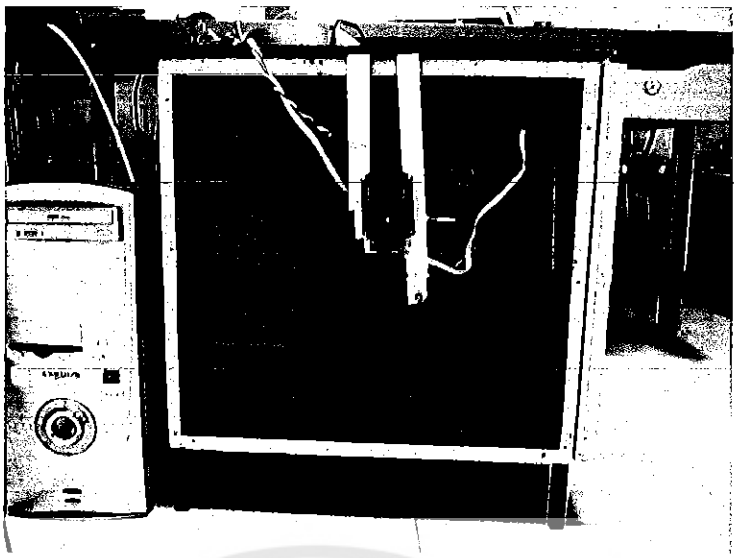

รูปที่ 3.5 แผนผังระบบควบคุมภายในเครื่องฟักไข่ที่ทำการออกแบบ

### 3.2.1 ขนาดและโครงสร้างของตู้ฟักไข่

- ใช้โครงสร้างและขนาดเดียวกับตู้ฟักไข่ที่มีอยู่ในปัจจุบัน
- ผนังของตู้ฟักไข่เพิ่มผนังไม้อัดอีก 1 ชั้นเพื่อการควบคุมอุณหภูมิที่ดีขึ้น
- ลดจำนวนชั้นของแผงฟักไข่เหลือ1ชั้น เพราะทดลองกับไข่จำนวนไม่มาก
- วางตำแหน่งของพัดลมที่ใช้กระจายความร้อนให้หันหน้าพัดลมพัดความร้อนเข้าสู่ผนัง เพื่อให้สะท้อนความร้อนแทนการพัดเข้าสู่ไข่โดยตรงทั้งยังช่วยในการกระจายความร้อน



รูปที่ 3.6 ด้านหน้าของเครื่องฟักไข่ที่ทำการออกแบบ

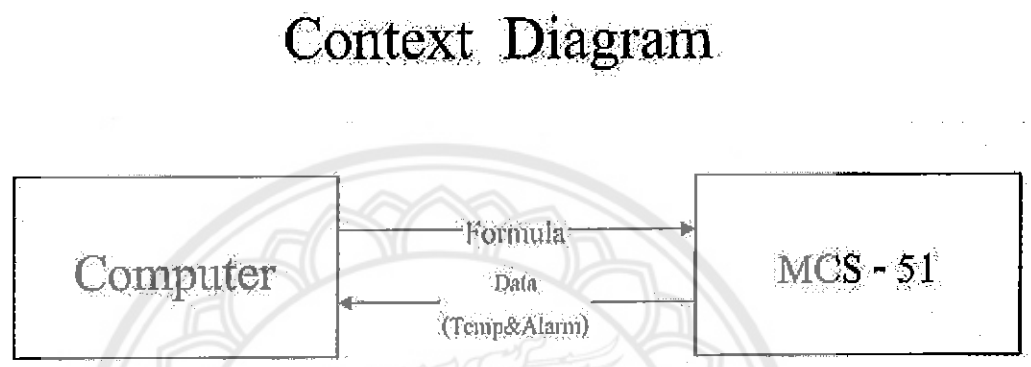


รูปที่ 3.7 ด้านข้างของเครื่องฟักไข่ที่ทำการออกแบบ

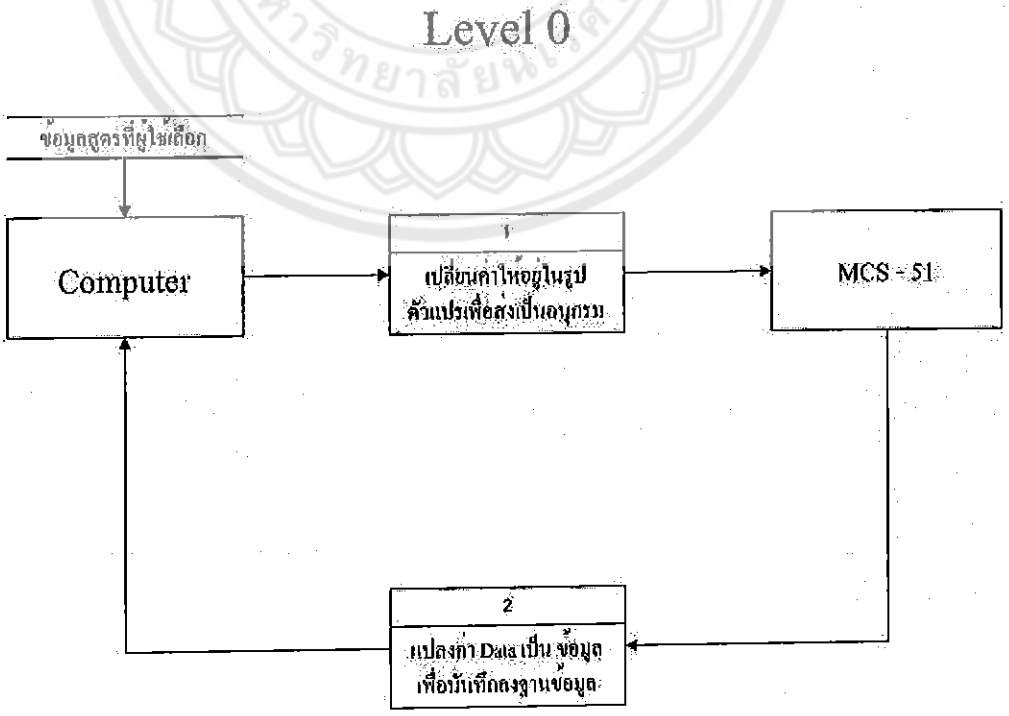
รูปที่ 3.8 ลักษณะภายในของเครื่องฟักไข่ที่ทำการออกแบบ

### 3.2.2 ระบบฐานข้อมูล

ระบบฐานข้อมูลจะมีการบันทึกทั้งสองฝั่งทั้งทางคอมพิวเตอร์และ MCS-51โดยทางฝั่งคอมพิวเตอร์นั้นจะมีฐานข้อมูลสูตรเพื่อส่งสูตรไปให้MCS-51 ทางพอร์ท RS-232 และทางฝั่ง MCS-51 นั้นจะมีการบันทึกฐานข้อมูล Alarm และ อุณหภูมิ เพื่อเก็บค่าไว้และส่งไปให้คอมพิวเตอร์เมื่อนำมาเชื่อมต่อ



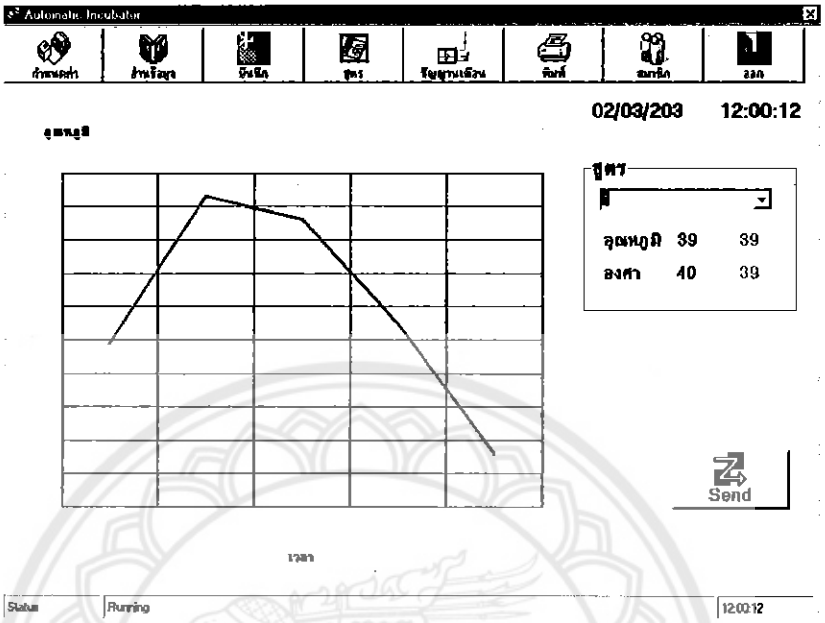

รูปที่ 3.9(a) โครงสร้างของฐานข้อมูล




รูปที่ 3.9(b) โครงสร้างของฐานข้อมูล

ในส่วนของลักษณะของโปรแกรมที่ใช้ติดต่อระหว่างระบบควบคุมและผู้ใช้งานนั้นได้ทำการออกแบบไว้ดังนี้

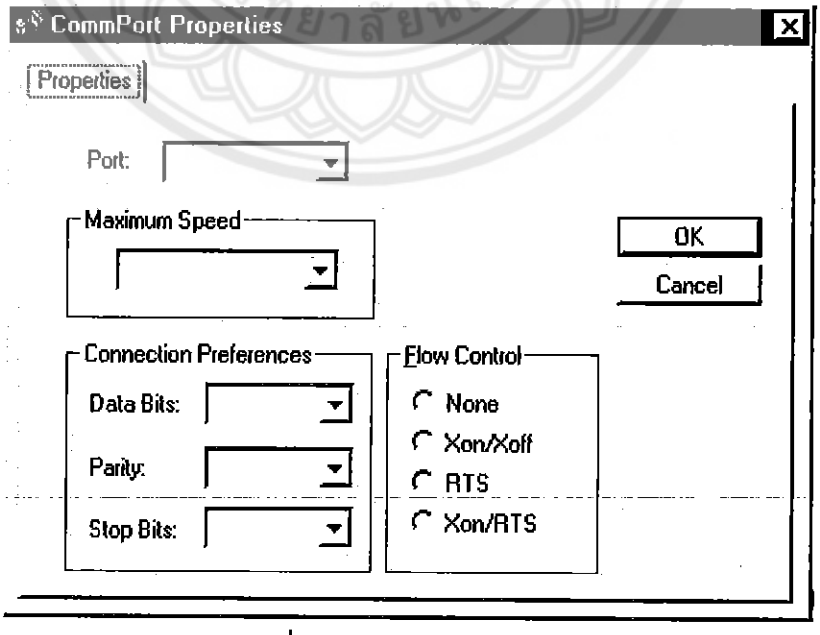
1. ส่วนของหน้าจอหลัก




**รูปที่ 3.10** หน้าจอหลักของโปรแกรม

จะประกอบไปด้วย ส่วนของการแสดงผลเป็นกราฟแสดงอุณหภูมิ การกำหนดสูตรของการฟักไข่ บาร์แสดงเวลาปัจจุบันและเวลาที่ผ่านมาของการฟัก และลิงค์ไปยังหน้าจอส่วนควบคุมอื่นๆ

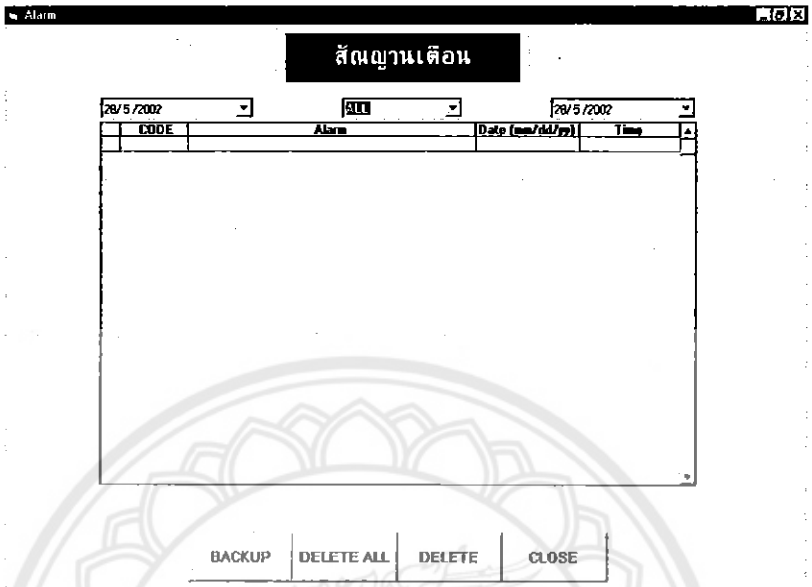
2. ส่วนของหน้าจอเซ็ทพอร์ต




**รูปที่ 3.11** หน้าจอเซ็ทพอร์ต

จะเป็นหน้าจอที่ใช้กรอกค่าเพื่อการติดต่อกันระหว่างไมโครคอมพิวเตอร์และไมโครคอนโทรลเลอร์ รวมทั้งแสดงผลการยืนยันค่าเมื่อทำการติดต่อสำเร็จ

3. ส่วนของหน้าจอแสดงการเตือน(Alarm)

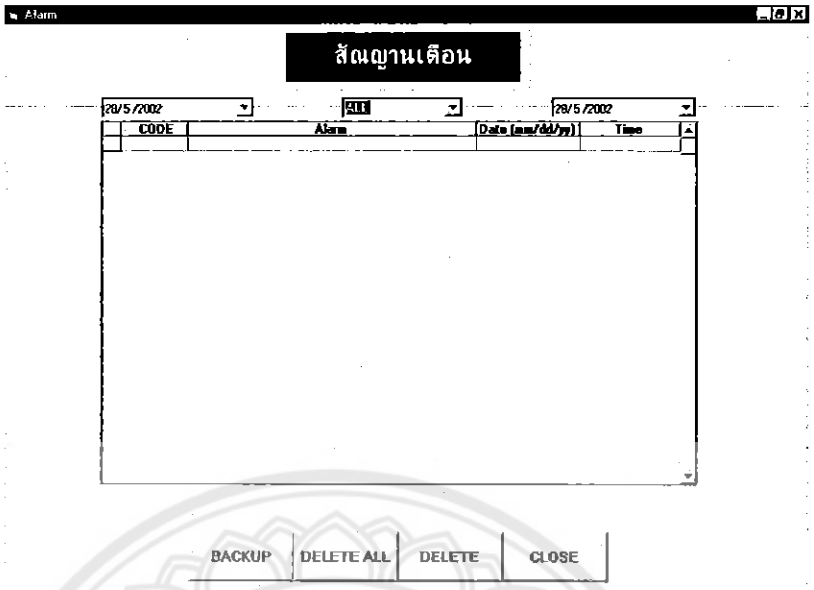



รูปที่ 3.12 หน้าจอแสดงการเตือน(Alarm)

จะแสดงข้อมูลของการเกิดข้อผิดพลาดของระบบ โดยจะเก็บข้อมูลไว้ในฐานข้อมูลด้วย ซึ่งจะแสดงผล ของครั้งที่เกิด เกิดความผิดพลาดอะไร วันไหน เมื่อไหร่ ซึ่งข้อผิดพลาดหลักที่จะบันทึกไว้มีดังนี้

- ข้อผิดพลาดเกี่ยวกับอุณหภูมิ
- ข้อผิดพลาดของการกลับไข่(Motor)

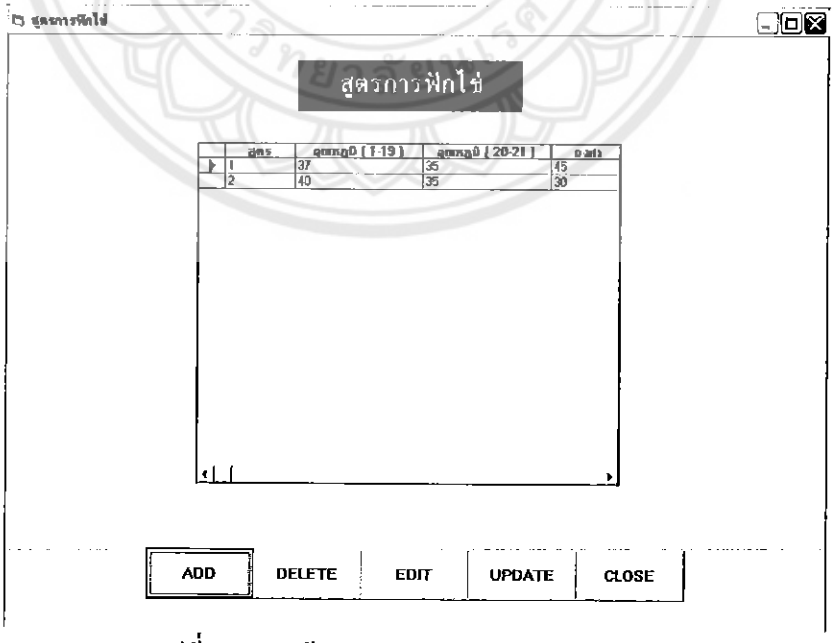
4. ส่วนของหน้าจอกำหนดสูตรสำหรับการฟักไข่




**รูปที่ 3.13** หน้าจอกำหนดสูตรสำหรับการฟักไข่

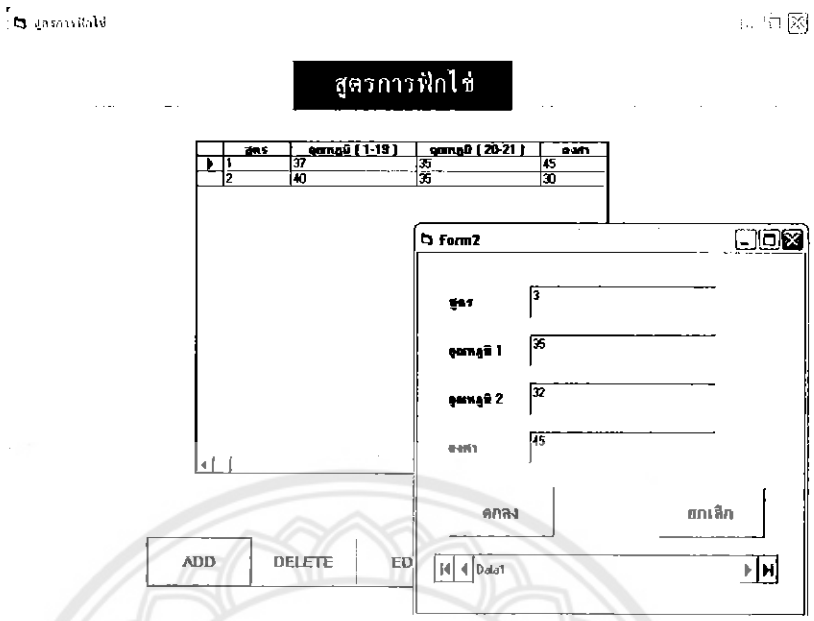
เป็นส่วนที่ให้ผู้ใช้เป็นผู้กำหนดสูตรสำหรับการฟักไข่ทั้งส่วนของอุณหภูมิและองศาของการกลับไข่ซึ่งสามารถเพิ่มเติมสูตรหรือลบสูตรได้และเมื่อเลือกสูตรแล้วก็จะไปแสดงผลการเลือกที่หน้าจอหลัก

ตัวอย่างการเพิ่มสูตร



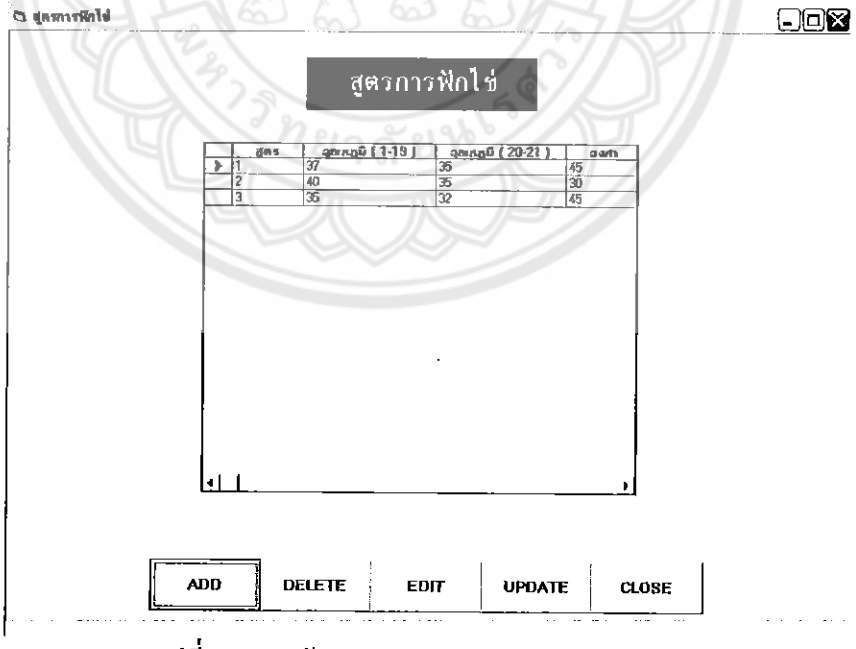

**รูปที่ 3.14** หน้าจอกำหนดสูตรสำหรับการฟักไข่

- ขั้นแรก กด ADD เพื่อเพิ่มสูตรดังในรูป 3.11



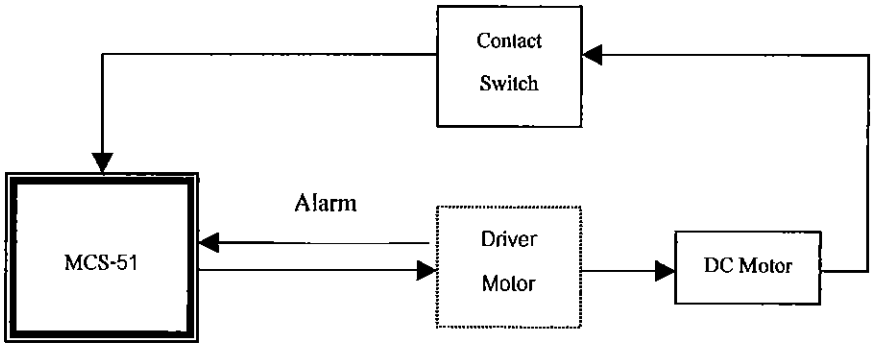

**รูปที่ 3.15** หน้าจอกำหนดสูตรสำหรับการฟักไข่

- ขั้นที่สองทำการกรอกสูตรดังในรูป 3.12
- กดตกลงเพื่อบันทึกสูตรลงใน Database ดังในรูป 3.13




**รูปที่ 3.16** หน้าจอกำหนดสูตรสำหรับการฟักไข่

### 3.2.3 การพลิกไข่




รูปที่ 3.17 แผนผังระบบควบคุมมอเตอร์ภายในเครื่องฟักไข่

<u>รูปแบบ</u>

ในส่วนประกอบของอุปกรณ์ในการพลิกไข่นั้น จะประกอบไปด้วย

1. แผงไข่ซึ่งยึดกับแกนเหล็ก โดยใช้แกนเหล็กเป็นจุดหมุน
2. มอเตอร์กระแสตรง(DC Motor) เพื่อทำหน้าที่หมุนแผงไข่
3. ไดรเวอร์มอเตอร์(Driver Motor) เป็นตัวขับสัญญาณที่ได้จากคอนโทรลเลอร์ให้เป็นแรงดันขับมอเตอร์
4. คอนแท็ค สวิตช์(Contact Switch) ทำหน้าที่เป็นตัวควบคุมองศาของแผงโดยจะเป็นตัวเช็คจุดการทำงานของมอเตอร์โดยตั้งไว้ที่องศาต่างๆที่เราใช้ในการศึกษา

รูปที่ 3.18 มอเตอร์และคอนแท็คสวิตช์

การทำงาน

ในการควบคุมการพลิกไข่ โดยรวมทั้งหมดแล้วการพลิกไข่จะขึ้นอยู่กับการควบคุมมอเตอร์ ซึ่งในส่วนของการควบคุมมอเตอร์นั้น ได้แบ่งการควบคุมไว้เป็น 2 ส่วน โดยแยกตามหน้าที่ในการพลิกไข่ ซึ่งจะแบ่งได้ดังนี้

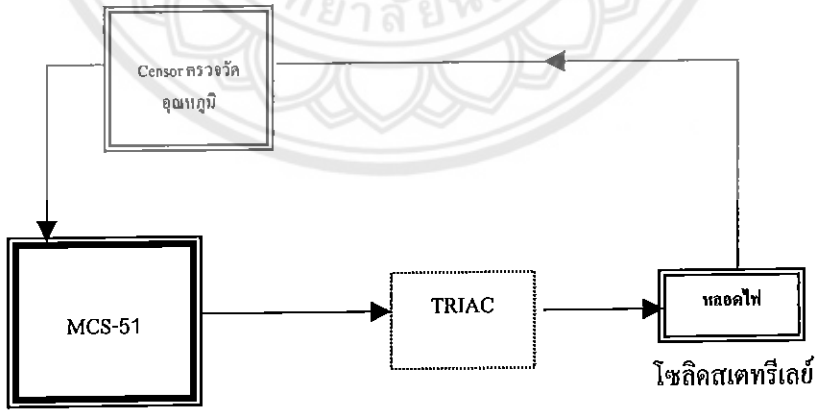
1. ส่วนควบคุมการหมุนของมอเตอร์
2. ส่วนควบคุมองศาของการกลับไข่

โดยมีรายละเอียดทั้งหมดดังนี้

1. ส่วนของการควบคุมการหมุนของมอเตอร์นั้น เราจะกำหนดไว้ว่า เมื่อได้รับสัญญาณมาจากไมโครคอนโทรลเลอร์ให้หมุนมอเตอร์ สัญญาณจะผ่านเข้าไปยังบอร์ดของไดรเวอร์มอเตอร์ ซึ่งจะทำหน้าที่เพิ่มแรงดันให้กับสัญญาณที่ได้รับเข้ามา โดยเมื่อได้รับสัญญาณมาแล้วก็จะทำหน้าที่เป็นตัวขับให้มอเตอร์ทำงาน

2. ส่วนควบคุมองศาของการพลิกไข่ เราจะใช้คอนแท็คสวิตช์(Contact Switch) เป็นตัวควบคุมองศา โดยกำหนดให้ติดตั้ง คอนแท็คสวิตช์ในมุมที่เราต้องการให้หยุดการทำงานของมอเตอร์ ซึ่งเราจะกำหนดไว้ที่ 30 องศา และ 45 องศา เมื่อมอเตอร์หมุนจนหน้าสัมผัสของแผงไข่มาสัมผัสกับ คอนแท็คสวิตช์ จะทำให้มอเตอร์หยุดทำงานและรอจนกว่าวัฏจักรของการทำงานใหม่จะมาก็จะดำเนินการหมุนมอเตอร์กลับโดย สัญญาณที่ได้จากคอนแท็คสวิตช์จะเป็นสัญญาณในการกำหนด Direction ในไดรเวอร์มอเตอร์

**3.2.4. ส่วนควบคุมอุณหภูมิ**




**รูปที่ 3.19** แผนผังระบบควบคุมอุณหภูมิภายในเครื่องฟักไข่

รูปแบบ

ในส่วนของอุปกรณ์ควบคุมอุณหภูมินั้นจะประกอบไปด้วย

1. หลอดไฟ เป็นตัวกำเนิดความร้อนให้กับตู้ฟักไข่
2. ไอซี ตรวจวัดอุณหภูมิ DS1621
3. โซลิดสเตทรีเลย์(Solid State Relay) MM-SSR3A

การทำงาน

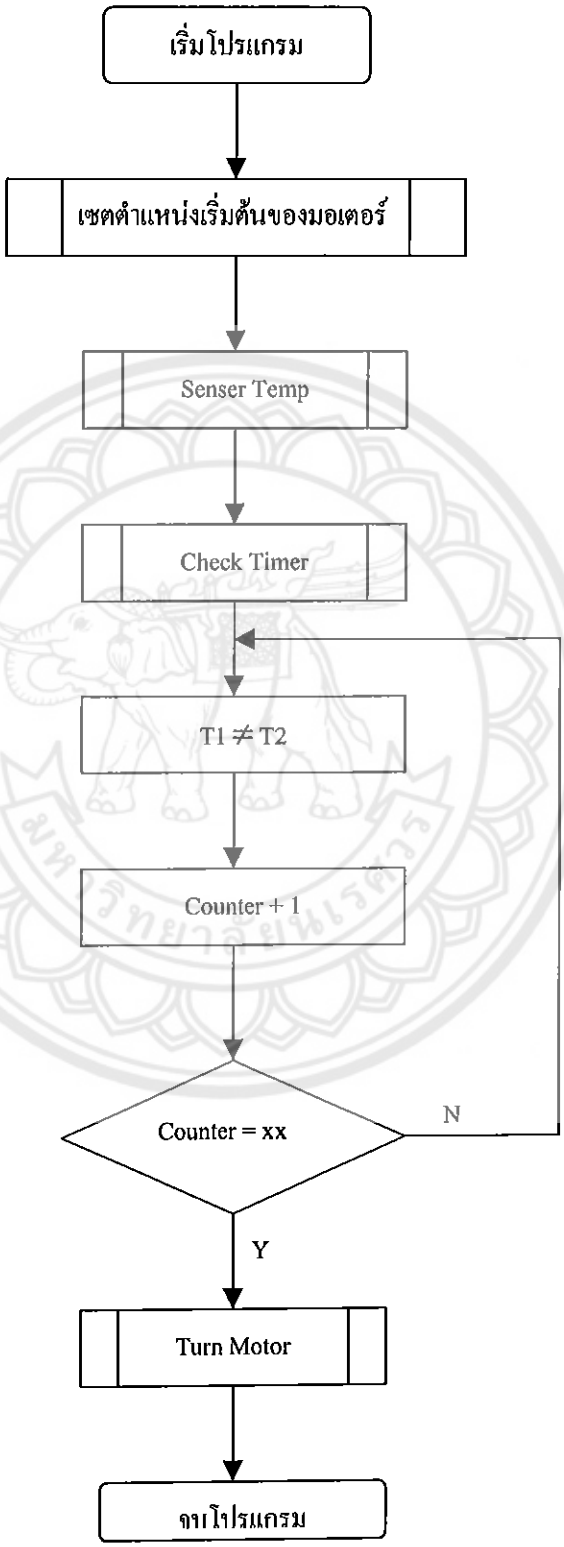
ส่วนของการควบคุมอุณหภูมิโดยใช้หลอดไฟนั้น เราจะควบคุมโดยการควบคุมปริมาณแรงดันไฟฟ้าที่จ่ายให้กับหลอดไฟ โดยสัญญาณที่ให้มาจากไมโครคอนโทรลเลอร์นั้น เราจะควบคุมสัญญาณที่ให้มาเป็นระดับ Logic TTL โดยตรง(0-5VDC)โดยจะให้สัญญาณเข้าไปยังบอร์ดของโซลิดสเตทรีเลย์(Solid State Relay)MM-SSR3A ซึ่งจะทำหน้าที่ตัดต่อแรงดันไฟ AC ในลักษณะเดียวกับ Relay โดยใช้ TRIAC เป็นตัวควบคุม ซึ่งจะไม่มีการตัดต่อแบบหน้าสัมผัสทำให้ตัดปัญหาเรื่องไฟสปาร์ค(SPARK)ไปได้ และ ไม่เกิดสัญญาณรบกวนต่อระบบไมโครคอนโทรลเลอร์ซึ่งแรงดันไฟฟ้าที่ออกมาจากบอร์ดโซลิดสเตทรีเลย์ ก็จะเป็นแรงดันไฟฟ้าให้กับหลอดไฟต่อไป ซึ่งอุณหภูมิที่ใช้นั้นมาจากไอซีตรวจวัดอุณหภูมิเมื่อ DS1621 เริ่มทำงาน จะทำการวัดอุณหภูมิ โดยภายในไอซีจะมีตัวตรวจจับอุณหภูมิบรรจุอยู่ ผลการวัดจะส่งต่อไปยังส่วนจัดการแอดเดรสและควบคุมอินพุตเอาท์พุต ในส่วนนี้นำข้อมูลที่วัดได้ส่งต่อไปยังบัสผ่านขา SDA โดยได้รับการกำหนดจังหวะการทำงานจากส่วนรีจิสเตอร์ควบคุมและแสดงสถานะ เนื่องจาก DS1621 มีขากำหนดแอดเดรส 3 ขา คือ A2-A0 จึงทำให้สามารถต่อพ่วง DS1621 ได้สูงสุด 8 ตัว โดยแต่ละตัวจะต้องกำหนดสถานะลอจิกที่ขาแอดเดรสทั้ง 3 ให้แตกต่างกัน ส่วนจัดการแอดเดรสและควบคุมอินพุตเอาท์พุตของ DS1621 แต่ละตัวเป็นตัวจัดการส่งข้อมูลตามที่อุปกรณ์มาสเตอร์ร้องขอ

ค่าของอุณหภูมิที่วัดได้ถูกเก็บอยู่ในรูปของข้อมูลเลขฐานสองขนาด 9 บิต ข้อมูลดิจิตอลเอาท์พุตจะแบ่งเป็น 2 ส่วน 8 บิตบนหรือไบต์สูงจะเป็นค่าอุณหภูมิจำนวนเต็ม โดยบิต MSB จะเป็นตัวแสดงเครื่องหมายว่า อุณหภูมิเป็นบวกหรือลบ หากบิต MSB เป็น “1” แสดงว่าอุณหภูมิที่วัดได้มีค่าเป็นลบ ในขณะที่ 8 บิตล่างหรือไบต์ล่างจะมีใช้งานเพียงบิตเดียวจะเป็นบิตแสดงค่าความละเอียดซึ่งเท่ากับ 0.5 C

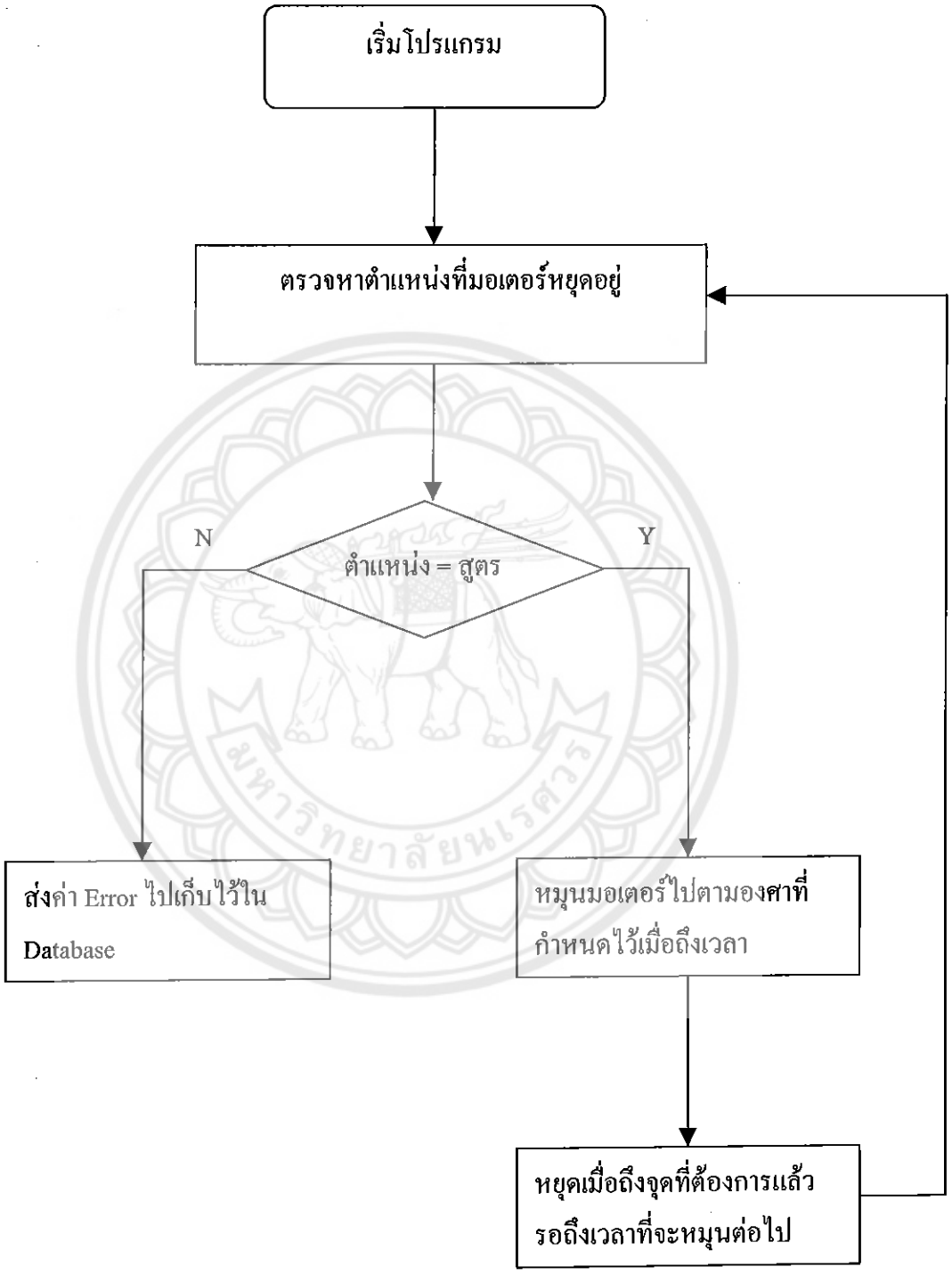
เมื่อสั่งให้ DS1621 ส่งข้อมูลอุณหภูมิออกมา DS1621 จะใช้เวลาไม่เกิน 1 วินาที จึงจะส่งข้อมูลออกมา ดังนั้นในการควบคุมการทำงานจึงต้องมีการหน่วงเวลา เพื่อรอให้ DS1621 วัดและแปลงค่าออกมาเป็นข้อมูลให้เรียบร้อยเสียก่อน

## 3.3 แนวคิดการเขียนโปรแกรม

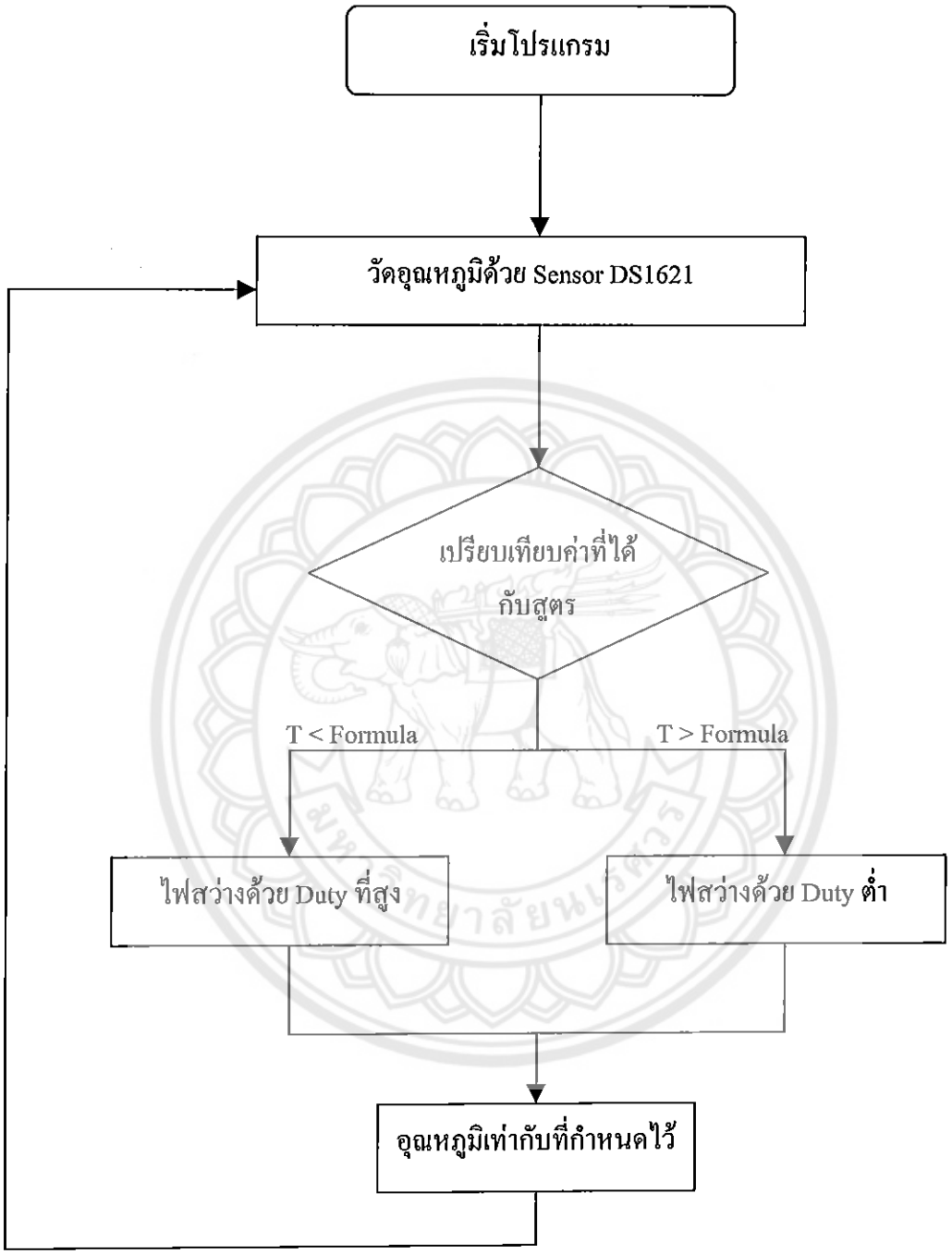
### 3.3.1 โปรแกรมเซตค่าหน่วงเวลาของมอเตอร์



### 3.3.2 โปรแกรม การหมุนมอเตอร์



### 3.3.3 โปรแกรมตรวจวัดและควบคุมอุณหภูมิ



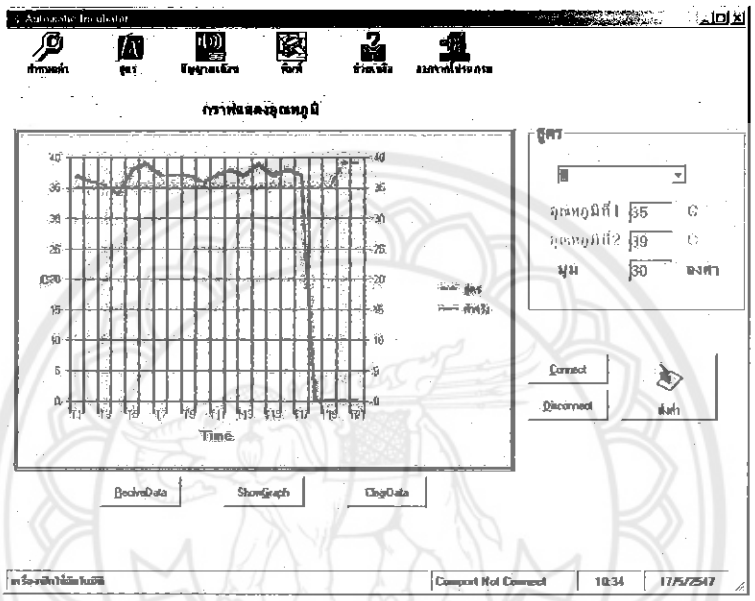
### 3.3.4 โปรแกรมเมื่อทำการส่งผ่านข้อมูลระหว่างคอมพิวเตอร์และ MCS-51ทางพอร์ท RS-232

# บทที่ 4

# การทดลอง

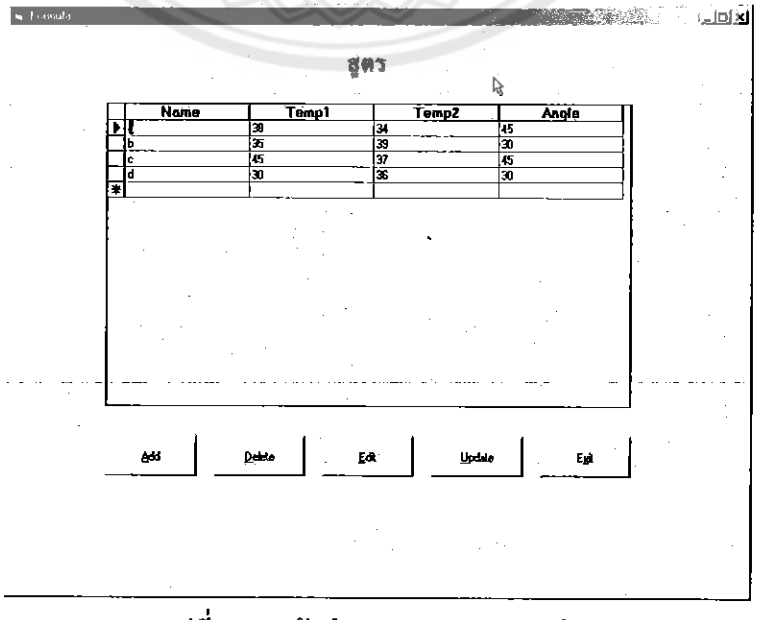
## 4.1 การทำงานในส่วนโปรแกรมติดต่อกับผู้ใช้

1. เมื่อเข้ามาในโปรแกรมจะมีลักษณะดังรูปข้างล่างนี้



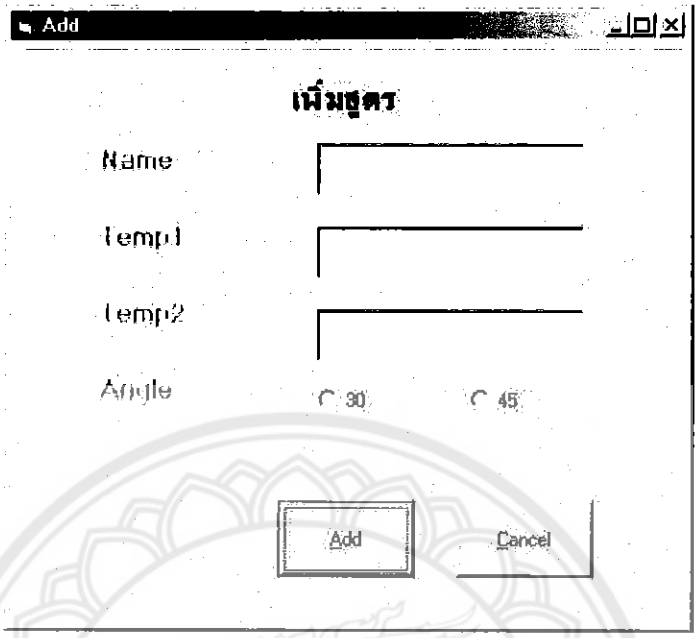

**รูปที่ 4.1 หน้าต่างหลักของโปรแกรมในส่วนติดต่อกับผู้ใช้**

2. รูปแสดงหน้าต่างแสดงตารางสูตรต่างๆ



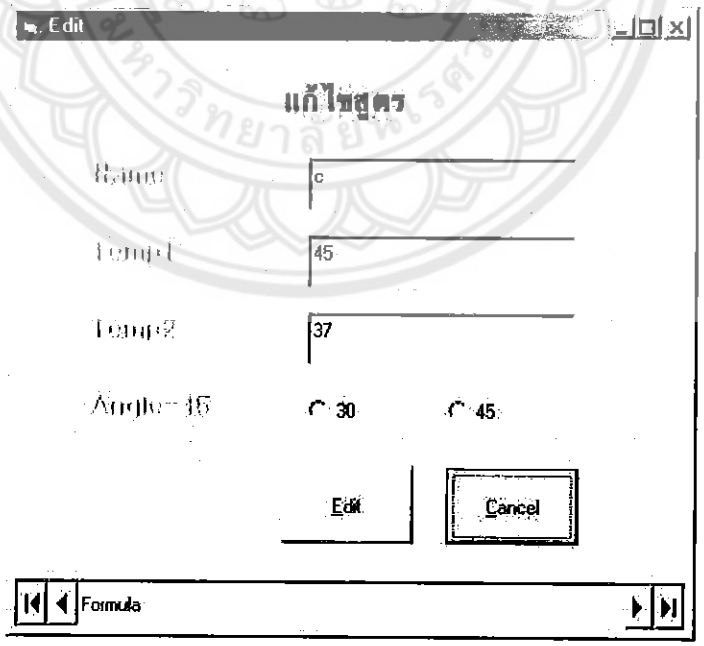

**รูปที่ 4.2 หน้าต่างแสดงตารางสูตรต่างๆ**

3. รูปแสดงหน้าต่างเพิ่มสูตร



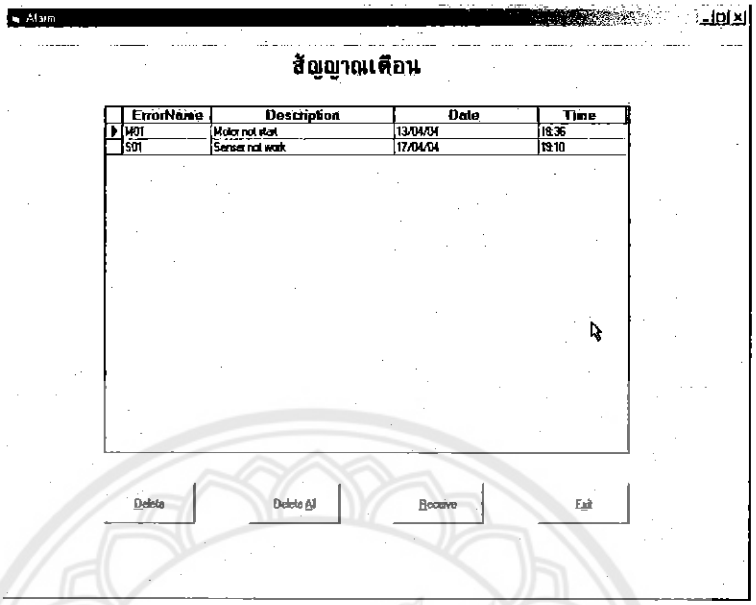

**รูปที่ 4.3 แสดงหน้าต่างเพิ่มสูตร**

4. รูปแสดงหน้าต่างแก้ไขสูตร



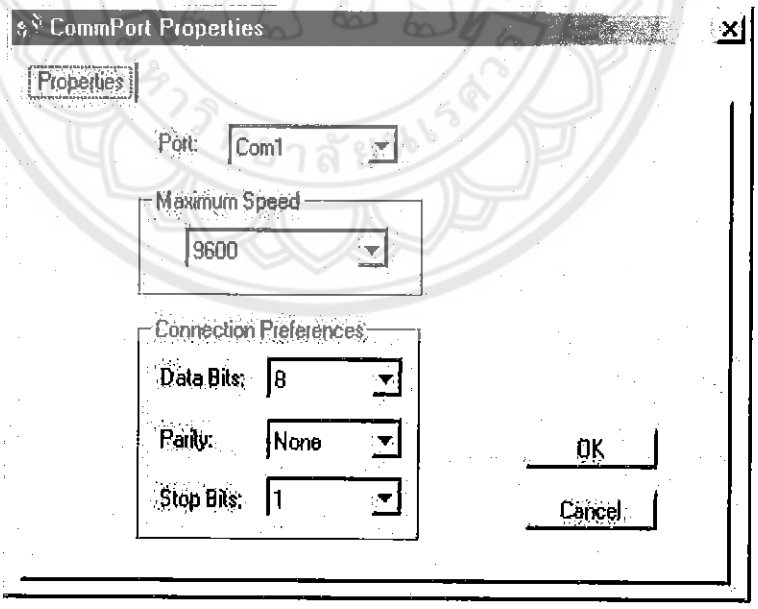

**รูปที่ 4.4 แสดงหน้าต่างแก้ไขสูตร**

5. รูปแสดงหน้าต่างแสดงตารางสัญญาณเตือน




**รูป 4.5 แสดงหน้าต่างแสดงตารางสัญญาณเตือน**

6. รูปแสดงหน้าต่างกำหนดค่า Com Port




**รูป 4.6 แสดงหน้าต่างกำหนดค่า Com Port**

## 4.2 การทดลองเครื่องฟักไข่

ในการทดลองในโครงงานนี้จะแบ่งออกเป็น 3 ส่วนคือ

### 4.2.1 การทดสอบการพลิกไข่ด้วยDC มอเตอร์

จากส่วนประกอบทั้งหมดของอุปกรณ์ที่ใช้สำหรับพลิกไข่ เมื่อประกอบครบหมดแล้วจึงลงมือทำการทดสอบการทำงานของชุดพลิกไข่โดยกำหนดสถานการณ์เพื่อการทดสอบดังนี้

1. การพลิกไข่ที่มุม 30 องศา
2. การพลิกไข่ที่มุม 45 องศา

จากการทดสอบแล้วบันทึกผลการทำงานของชุดพลิกไข่แล้วสรุปผลการทำงาน

ผลการทดลอง

เมื่อทำการรันโปรแกรมที่ควบคุมการเคลื่อนที่ของมอเตอร์แล้ว มอเตอร์จะทำการเซทตำแหน่งไปที่จุดเริ่มต้น(SW1) หลังจากนั้นจะทำการหยุดรอเวลาตามที่กำหนดจนถึงเวลาแล้ว ก็จะทำการเคลื่อนที่กลับมาหยุดที่ตำแหน่งองศาที่ตั้งไว้

ตารางที่ 4.1 การทดลองการพลิกไข่ด้วยมอเตอร์ตามองศาที่กำหนด

| องศาที่กำหนด | มุมที่มอเตอร์พลิกได้ |
|---|---|
| 30 | 30 |
| 45 | 45 |

สรุปผลการทดลอง

ส่วนควบคุมการทำงานของมอเตอร์สามารถควบคุมการพลิกไข่ได้ตามองศาและเวลาที่กำหนดไว้

### 4.2.2 การทดลองเพื่อเปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์

ทำการทดลองวัดอุณหภูมิภายในเครื่องฟักไข่เมื่อมีการเพิ่มอุณหภูมิให้ โดยวัดอุณหภูมิทุกๆ 2 นาที เป็นเวลา 1 ชั่วโมง โดยทำการทดลองเหมือนกัน 2 ครั้งรวมเป็นระยะเวลา 2 ชั่วโมง ได้ผลการทดลองดังนี้

ผลการทดลอง

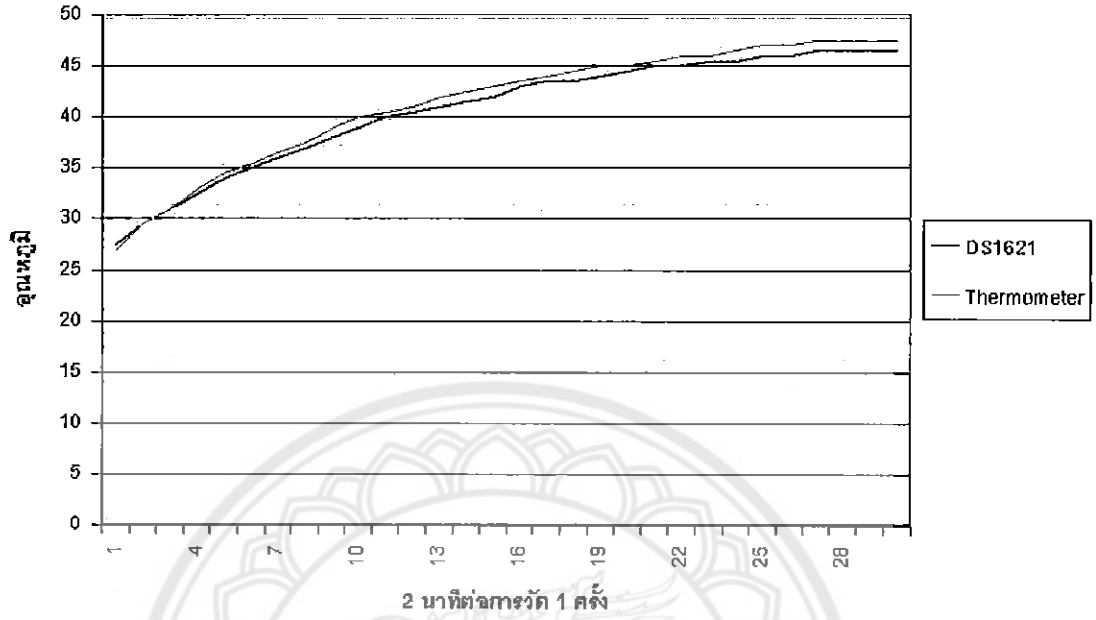
**ตารางที่ 4.2** เปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์ชุดที่1

| ไอซี DS1621 | เทอร์โมมิเตอร์ | ผลต่าง |
|---|---|---|
| 27.5 | 27 | 0.5 |
| 29.5 | 29.5 | 0 |
| 31 | 31 | 0 |
| 32.5 | 33 | -0.5 |
| 34 | 34.5 | -0.5 |
| 35 | 35.5 | -0.5 |
| 36 | 36.5 | -0.5 |
| 37 | 37.5 | -0.5 |
| 38 | 39 | -1 |
| 39 | 40 | -1 |
| 40 | 40.5 | -0.5 |
| 40.5 | 41 | -0.5 |
| 41 | 42 | -1 |
| 41.5 | 42.5 | -1 |
| 42 | 43 | -1 |
| 43 | 43.5 | -0.5 |
| 43.5 | 44 | -0.5 |
| 43.5 | 44.5 | -1 |
| 44 | 45 | -1 |
| 44.5 | 45 | -0.5 |
| 45 | 45.5 | -0.5 |
| 45 | 46 | -1 |
| 45.5 | 46 | -0.5 |
| 45.5 | 46.5 | -1 |
| 46 | 47 | -1 |
| 46 | 47 | -1 |
| 46.5 | 47.5 | -1 |
| 46.5 | 47.5 | -1 |
| 46.5 | 47.5 | -1 |
| 46.5 | 47.5 | -1 |

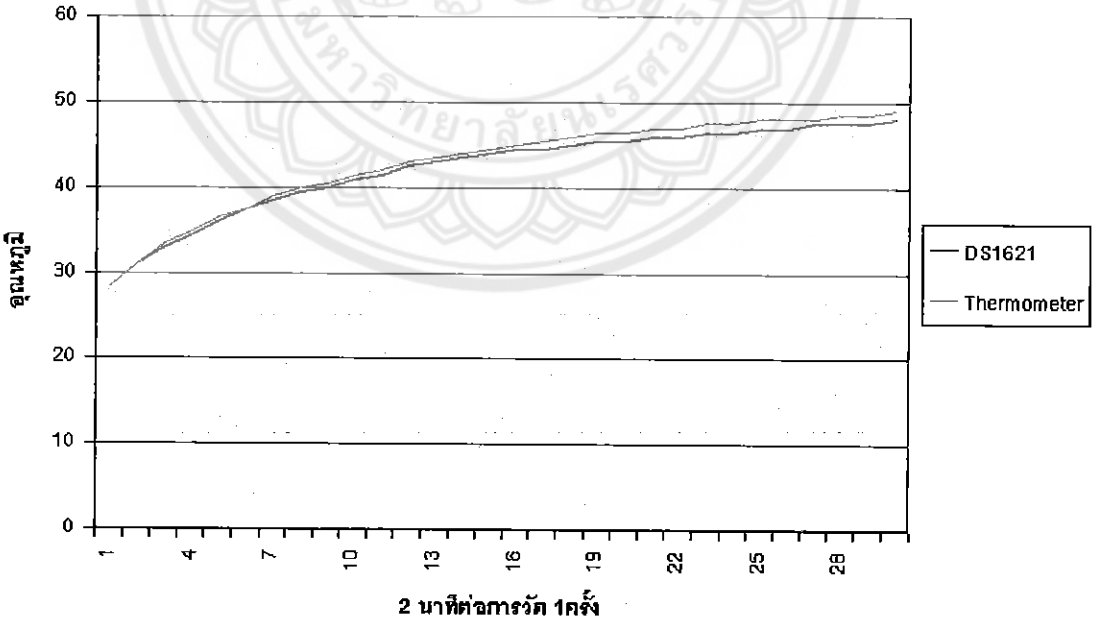
**ตารางที่ 4.3** เปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์ ชุดที่2

| ไอซี DS1621 | เทอร์โมมิเตอร์ | ผลต่าง |
|---|---|---|
| 28.5 | 28.5 | 0 |
| 31 | 31 | 0 |
| 33 | 33.5 | -0.5 |
| 34.5 | 35 | -0.5 |
| 36 | 36.5 | -0.5 |
| 37.5 | 37.5 | 0 |
| 38.5 | 39 | -0.5 |
| 39.5 | 40 | -0.5 |
| 40 | 40.5 | -0.5 |
| 41 | 41.5 | -0.5 |
| 41.5 | 42 | -0.5 |
| 42.5 | 43 | -0.5 |
| 43 | 43.5 | -0.5 |
| 43.5 | 44 | -0.5 |
| 44 | 44.5 | -0.5 |
| 44.5 | 45 | -0.5 |
| 44.5 | 45.5 | -1 |
| 45 | 46 | -1 |
| 45.5 | 46.5 | -1 |
| 45.5 | 46.5 | -1 |
| 46 | 47 | -1 |
| 46 | 47 | -1 |
| 46.5 | 47.5 | -1 |
| 46.5 | 47.5 | -1 |
| 47 | 48 | -1 |
| 47 | 48 | -1 |
| 47.5 | 48 | -0.5 |
| 47.5 | 48.5 | -1 |
| 47.5 | 48.5 | -1 |
| 48 | 49 | -1 |

จากข้อมูลที่ได้นำมาเขียนกราฟได้ดังนี้




**รูปที่ 4.7** กราฟแสดงการเปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์ ชุดที่1




**รูปที่ 4.8** กราฟแสดงการเปรียบเทียบค่าอุณหภูมิที่วัดได้ระหว่างไอซีวัดอุณหภูมิและ เทอร์โมมิเตอร์ชุดที่2

จากข้อมูลที่ได้มาทั้งหมดเมื่อนำมาหาผลต่างของข้อมูลแล้วได้ผลดังนี้

- อุณหภูมิที่ได้จากการวัดด้วยเซนเซอร์วัดอุณหภูมิและเทอร์โมมิเตอร์มีผลต่าง 0 องศาเซลเซียสจำนวน 5 ค่า
- อุณหภูมิที่ได้จากการวัดด้วยเซนเซอร์วัดอุณหภูมิและเทอร์โมมิเตอร์มีผลต่าง 0.5 องศาเซลเซียสจำนวน 27 ค่า
- อุณหภูมิที่ได้จากการวัดด้วยเซนเซอร์วัดอุณหภูมิและเทอร์โมมิเตอร์มีผลต่าง 1 องศาเซลเซียสจำนวน 28 ค่า

จากการทดลองทั้ง 2 ชุดได้ค่าเฉลี่ยของผลต่างคือ 0.68333 และ 0.66667 องศาเซลเซียส

<u>สรุปผลการทดลอง</u>

อุณหภูมิที่ได้จากการวัดด้วยเซนเซอร์วัดอุณหภูมิและเทอร์โมมิเตอร์มีผลต่างประมาณ 0.67 องศาถือว่ามีค่าใกล้เคียงกัน

**4.2.3 การทดลองเพื่อทดสอบการฟักไข่ที่ได้รับการออกแบบใหม่ กับเครื่องฟักไข่ที่มีอยู่ในปัจจุบัน**

การจะทดลองโดยให้ได้ผลการทดลองที่ได้ผลดีนั้น ควรจะมีการกำหนดตัวแปรที่ใช้ในการทดลองให้ครอบคลุมและเหมือนกันให้มากที่สุด

แต่จากการออกแบบเครื่องฟักไข่ขึ้นมาใหม่นี้ มีจุดประสงค์เพื่ออำนวยความสะดวกสบายให้แก่ผู้ใช้ เพราะฉะนั้น การทดลองในเรื่องความสะดวกสบายจึงไม่คำนึงถึงมากนักในการทดลองนี้ แต่จะไปคำนึงถึงการฟักไข่ที่ต้องมีคุณภาพอย่างน้อยต้องทัดเทียมเครื่องฟักไข่ที่มีอยู่ในปัจจุบันให้มากที่สุด

การทดลองเพื่อเปรียบเทียบเครื่องฟักไข่ออกแบบใหม่กับเครื่องฟักไข่ที่มีอยู่ในปัจจุบัน

- กำหนดระยะเวลา 1 รอบของการฟักไข่(1รอบคือ 21 วัน)
- ทดลองภายในสภาวะแวดล้อมเดียวกัน ใช้ไข่ไก่ในการทดลอง
- กำหนดให้ปัจจัยที่ใช้ในการฟักไข่ตามลักษณะของเครื่องฟักไข่แต่ละชนิด

**ตารางที่ 4.4** การทดลองเพื่อเปรียบเทียบเครื่องฟักไข่ออกแบบใหม่กับเครื่องฟักไข่ที่มีอยู่ในปัจจุบัน

| เครื่องฟักไข่ที่มีอยู่ในปัจจุบัน | เครื่องฟักไข่ที่ได้รับการออกแบบใหม่ |
|---|---|
| ไข่ชนิดเดียวกันขนาดเท่ากัน 10 ฟอง | ไข่ชนิดเดียวกันขนาดเท่ากัน 10 ฟอง |
| ควบคุมอุณหภูมิโดยเทอร์โมสตัส | ควบคุมอุณหภูมิโดยโซลิดสเตทรีเลย์ |
| พลิกไข่โดยใช้แรงคน | พลิกไข่ด้วยแผงไข่มอเตอร์ |
| ใช้ถาดน้ำในการให้ความชื้น | ใช้ถาดน้ำในการให้ความชื้น |
| ใช้หลอดไฟในการให้ความร้อน | ใช้หลอดไฟในการให้ความร้อน |
| ต้นทุนประมาณ 5000 บาท | ต้นทุนประมาณ 3500 บาท |

เมื่อเริ่มทำการทดลองคอยบันทึกผลของการทดลอง

- จำนวนไข่ที่ฟักนับตั้งแต่เริ่มฟักจนวันที่ 21 โดยการทดลองได้เริ่มทำตั้งแต่วันที่ 5 กุมภาพันธ์ 2547 และสิ้นสุดเมื่อวันที่ 26 กุมภาพันธ์ 2547 โดยทำการทดลองจำนวน 1 รอบ

<u>ผลการทดลอง</u>

เมื่อทำการฟักไข่ของทั้ง 2 เครื่องพร้อมกันจนถึงวันที่ 21 ปรากฏว่า

1.เครื่องฟักไข่ที่มีอยู่ในปัจจุบันมีการฟักจำนวน 6 ฟอง

2.เครื่องฟักไข่ที่ได้รับการออกแบบใหม่มีการฟักจำนวน 5 ฟอง

<u>สรุปผลการทดลอง</u>

เครื่องฟักไข่ที่ได้รับการออกแบบใหม่มีประสิทธิภาพทัดเทียมกับเครื่องฟักไข่ที่มีอยู่ในปัจจุบัน

# บทที่ 5
# บทสรุปและข้อเสนอแนะ

## 5.1 สรุปผลการทดลอง

จากการทดลองฟักไข่ด้วยเครื่องฟักไข่ที่ออกแบบและสร้างต้นแบบขึ้นมานั้น ได้ผลของการฟักไข่ได้ใกล้เคียงกับเครื่องฟักไข่ที่ใช้กันอยู่ทั่วไปในปัจจุบัน

การพลิกแผงไข่ด้วยมอเตอร์นั้นสามารถพลิกไข่ได้ตามองศาที่กำหนดไว้และยังสามารถพลิกแผงไข่ตามเวลาที่เหมาะสมตามข้อมูลที่กำหนดไว้ให้

ส่วนด้านการควบคุมอุณหภูมินั้นสามารถควบคุมอุณหภูมิให้คงที่ตามอุณหภูมิที่กำหนดมีความแตกต่างของอุณหภูมิที่วัดได้กับอุณหภูมิจริงอยู่ที่ประมาณ 0.6 องศาเซลเซียส และเมื่อถึงเวลาที่ต้องเปลี่ยนอุณหภูมิก็สามารถเปลี่ยนได้ตามที่กำหนดไว้ด้วย

จากการทดลองในโครงงานนี้ นอกจากจะต้องมีความเข้าใจในการใช้ไมโครคอนโทรลเลอร์ การออกแบบวงจรและการพัฒนาโปรแกรมบนพีซีแล้ว ยังต้องมีความรู้ในเรื่องการติดต่อกับอุปกรณ์อิเล็กทรอนิกส์ที่ใช้การติดต่อแบบ $I^2C$ อีกด้วยอีกทั้งยังต้องมีความเข้าใจในการออกแบบโครงสร้างกลไกต่างๆให้ดี

## 5.2 ข้อเสนอแนะและแนวทางแก้ไข

1. เนื่องจากความไม่สะดวกในการทำแผ่นปริ้นท์วงจรจึงได้ประกอบบอร์ดอินเตอร์เฟส โดยใช้สายไวร์-แรบปิ้ง ซึ่งอาจจะทำให้เกิดสัญญาณรบกวนได้ง่าย

2. ในการควบคุมตำแหน่งของ มอเตอร์ซึ่งใช้ลิมิตสวิตช์นั้น มีความยุ่งยากในการติดตั้งเพราะต้องออกแบบโครงสร้างให้ดีก่อนจะติดตั้งเพื่อป้องกันการทำงานที่ผิดพลาด

3. มอเตอร์ที่ใช้เป็น DC มอเตอร์ซึ่งทำให้ไม่สามารถกำหนดตำแหน่งการหมุนได้เหมือนสเต็ปมอเตอร์ ดังนั้นจึงใช้สวิตช์เพื่อการควบคุมแทน

4. มอเตอร์เมื่อมีการหมุนแล้วเมื่อจะสั่งให้หยุดจะมีแรงเฉื่อยที่เกิดขึ้น จึงจำเป็นต้องมีการแก้ไขโดยการเขียนโปรแกรมเพิ่มเติม เพื่อแก้ปัญหาตรงจุดนี้

5. ในส่วนของบอร์ดวัดอุณหภูมิ เป็นบอร์ดที่มีการติดต่อผ่านทาง $I^2C$ Bus ซึ่งทำให้ต้องมีการปรับเปลี่ยนการติดต่อกับระบบใหม่เพื่อรองรับ ทำให้ต้องมีการศึกษาเพิ่มเติมอีก

6. เซ็นเซอร์ที่ใช้ เป็นเซนเซอร์แบบดิจิตอล ทำให้การติดต่อระหว่างมาสเตอร์และ เซ็นเซอร์มีความยุ่งยาก ทั้งในเรื่องของคำสั่งและ Timing Diagram ซึ่งทำให้เสียเวลาในการศึกษาและทำความเข้าใจ

7. ในโครงงานนี้มีความจำเป็นที่จะต้องใช้เนื้อที่ในไมโครคอนโทรลเลอร์ เป็นจำนวนมาก เพื่อทำการเก็บข้อมูลเพื่อนำไปจัดเก็บใน database ทำให้มีความยุ่งยากในการเขียนโปรแกรม เพื่อให้กระทัดรัดและเปลืองหน่วยความจำน้อยที่สุด

8. ปัญหาแรมเสีย เป็นปัญหาที่ตรวจพบได้ยากและเสียเวลา ดังนั้นก่อนที่จะประกอบลงบอร์ดควรตรวจสอบก่อนทุกตัว

9. เนื่องจาก ลักษณะของโปรแกรมที่เขียนเป็นการติดต่อกับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 และไมโครคอมพิวเตอร์ ผ่านทาง RS232 ดังนั้นผู้วิจัยจะต้องมีความเข้าใจในลักษณะของภาษาและโครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51 ดีเป็นอย่างมาก เพื่อไม่ให้เกิดปัญหาและข้อผิดพลาดในการเขียนโปรแกรม

10. เนื่องจาก อุปกรณ์บางชิ้นไม่มีคู่มือสำหรับอุปกรณ์ ชิ้นนั้น จึงทำให้เสียเวลาในการค้นคว้าและทำความเข้าใจ เพื่อใช้งานอุปกรณ์นั้น แม้แต่การแจ้งแอดเดรสเพื่อให้ไมโครคอนโทรลเลอร์รู้ตำแหน่งของอุปกรณ์นั้นต้องทำการทดลองค่าหลายค่า ตามการค้นคว้าจากทางอินเตอร์เน็ท ซึ่งทำให้เสียเวลามาก

11. การทดลองเกี่ยวกับอุณหภูมิเป็นเรื่องที่กำหนดตัวแปรควบคุมในการทดลองยากมาก เนื่องจากสภาวะแวดล้อมต่างๆ รวมไปถึงการขาดอุปกรณ์เครื่องมือวัดที่ใช้ในการทดลอง จึงใช้การทดลองสุ่มตามการเขียนโปรแกรมจนได้ค่าตามต้องการ

12. จากการทดลอง พบว่าความรู้ที่ผู้วิจัยได้ศึกษามาขาดความรู้ทางทักษะในการปฏิบัติการกับไมโครคอนโทรลเลอร์ที่มากพอ โดยมีความรู้เพียงพื้นฐานเท่านั้น แต่ในการทดลองนี้ต้องการทักษะในการทดลองมากพอสมควร ทำให้เมื่อประสบปัญหาในการทดลองแล้ว ไม่สามารถวิเคราะห์ปัญหาว่าเกิดจากสาเหตุใดได้ ควรที่จะมีการทดลองในการเรียนให้ละเอียดมากกว่านี้ เพื่อสร้างทักษะให้เพียงพอ

## 5.3 แนวทางการพัฒนาต่อไป

1. ขยายขนาดของตู้ฟักให้มีความจุมากขึ้น
2. เพิ่มการควบคุมความชื้นให้กับระบบ
3. มีค่าสูตรในการควบคุมให้มากขึ้น
4. เพิ่มบอร์ดวัดอุณหภูมิให้มากขึ้นเพื่อความละเอียดในการวัดอุณหภูมิ

# เอกสารอ้างอิง


[1] มานิตย์ เทวรักษ์พิทักษ์. **การฟักไข่และการจัดการโรงฟัก.** เชียงใหม่ : ภาควิชาเทคโนโลยีทางสัตว์ สถาบันเทคโนโลยีการเกษตรแม่โจ้. 2538.

[2] สุนทร วิทูรสุรพจน์. **การใช้งานไมโครคอนโทรลเลอร์ ตระกูล 8051.** กรุงเทพมหานคร : ซีเอ็ดยูเคชัน. 2538.

[3] ธีรวัฒน์ ประกอบผล. **การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี.** กรุงเทพมหานคร : สำนักพิมพ์ ส.ส.ท. 2547.

[4] ชวลิต ขุนราม. **เซมิคอนดักเตอร์ อิเล็กทรอนิกส์.** ฉบับที่ 207. กรุงเทพมหานคร: ซีเอ็ดยูเคชั่น. 2543.

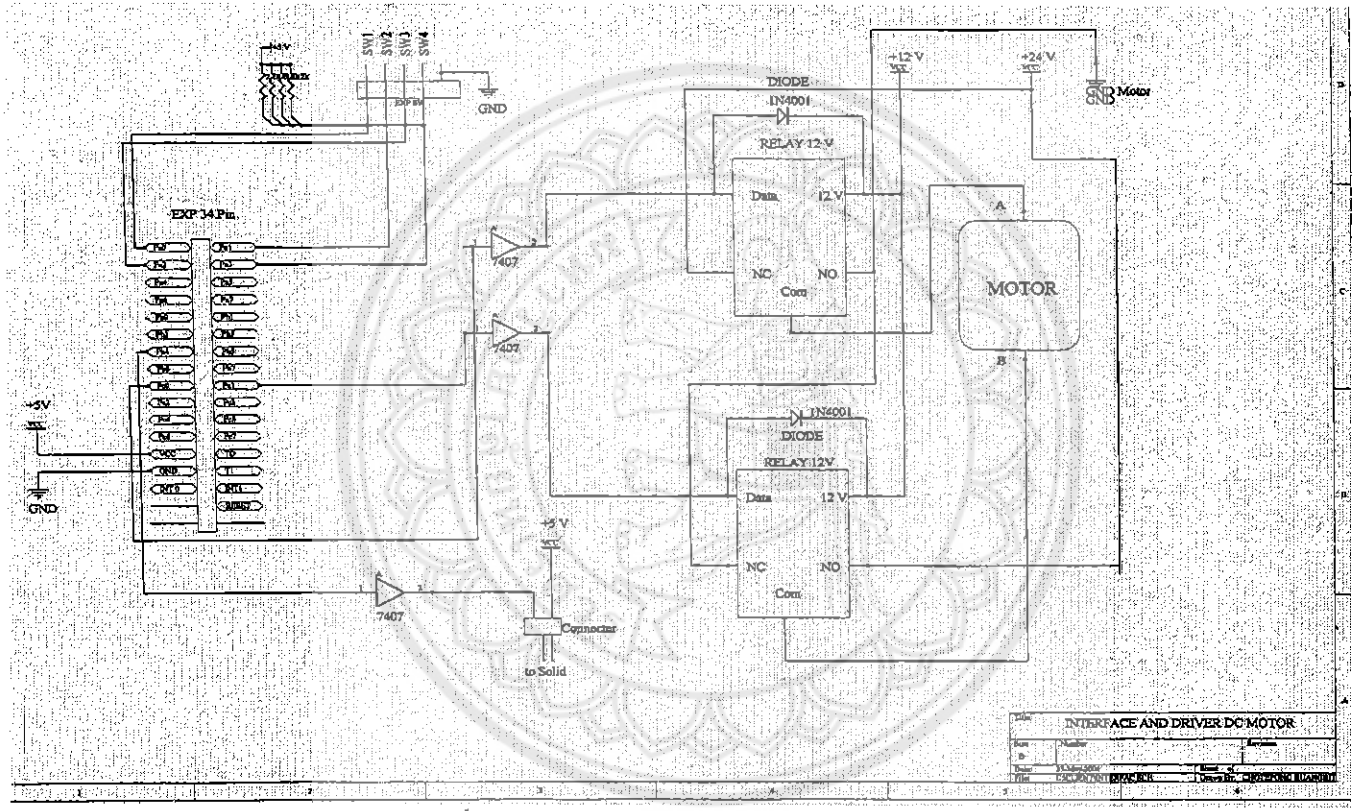
[5] KATSUHIKO OGATA. **MODERN CONTROL ENGINEER.** 3 rd EDITION. UNITED STATES OF AMERICA : PRETICE-HALL, INC. 1997.

[6] JOHN G. BOLLINGER, NEIL A. DUFFIE. **COMPUTER CONTROL OF MACHINES AND PROCESSES.** UNITED STATES OF AMERICA :ADDISON-WESLEY PUBLISHING COMPANY. 1988.

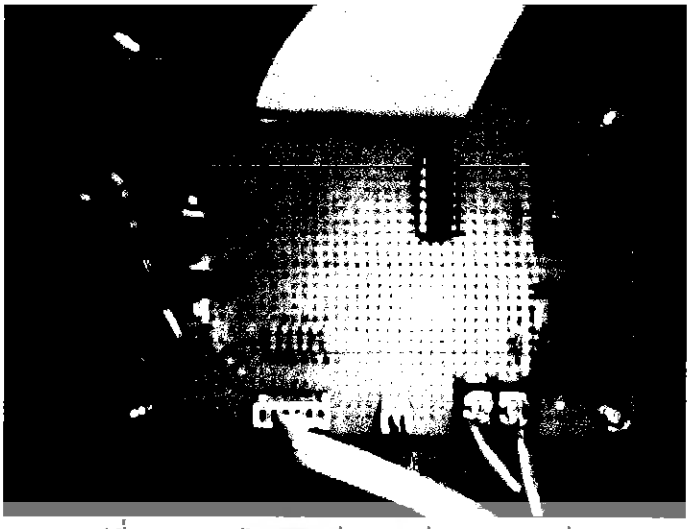
# ภาคผนวก

# ภาคผนวก ก

# วงจรไดรเวอร์มอเตอร์และอินเตอร์เฟส




**รูปที่ ก-1** วงจรไดรเวอร์มอเตอร์และอินเตอร์เฟส



รูปที่ ก-2 วงจรไดรเวอร์มอเตอร์และอินเตอร์เฟส

**การทำงานของบอร์ดอินเตอร์เฟสและไดร์เวอร์มอเตอร์**

ในส่วนของบอร์ดอินเตอร์เฟสได้แบ่งเป็น 3 ส่วนด้วยกันคือ

1. ส่วนรับค่าจากcontact switch
2. ส่วนส่งบิตควบคุมให้กับSolid State Relay
3. ส่วนควบคุม Driver motor

1. ในส่วนของการรับค่าจาก Contact switch นั้น จะมีการต่อ R pull up ไว้ที่ขารับ เพื่อ pull up ค่าจากสวิตซ์แล้วจึงรับค่าเข้าไปที่ขาของ Port Pa8255 ที่ขา Pa0-Pa3

2. ในส่วนของการส่งบิตควบคุมให้กับ Solid State Reley นั้นจะส่งค่าผ่าน Port Pb4 แล้วจึงผ่านบัฟเฟอร์ เพื่อแปลงสัญญาณให้เป็น บิต 1-0 แล้ว จึงส่งเข้าคอนเน็คเตอร์ของ Solid State Relay

3. ส่วนควบคุม Driver Motor นั้นจะส่งบิตควบคุมมาทาง Port Pc0-Pc3 แต่จะเช็คที่บิต 0 และ 2 เท่านั้น จึงต่อบิตนี้เข้าสู่บัฟเฟอร์ก่อนที่จะเข้าสู่ขา Com ของ Reley ซึ่งต่อ ไฟเลี้ยง 12 V และ ไฟมอเตอร์ 24 V ไว้แล้ว เมื่อปล่อยบิต 00 เข้ามาทำให้ Relay เกิดหน้าคอนแท็คที่สลับกันของ relay ทำให้กระแสสามารถไหลผ่านมอเตอร์ได้ทำให้มอเตอร์ทำงาน แต่เมื่อ ป้อนบิต 01 หรือว่า 02 ทำให้หน้าคอนแท็คไปในทางเดียวกันทำให้ไม่เกิดกระแสไหลผ่านมอเตอร์ จึงทำให้มอเตอร์หยุดทำงาน

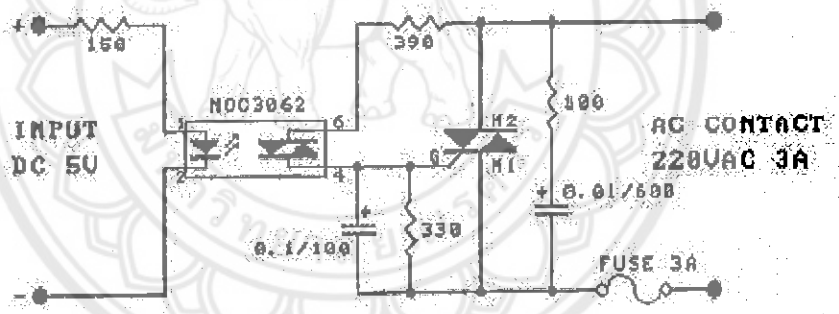
# ภาคผนวก ข

# ชุดโมดูล 3A Solid State Relay

MM-SSR3A คือ โมดูล Solid State Relay ซึ่งทำหน้าที่เป็นตัวตัดต่อแรงดันไฟ AC ในลักษณะเช่นเดียวกับ RELAY โดยใช้ TRIAC เป็นตัวควบคุม ทั้งนี้จะไม่มีการตัดต่อแบบหน้าสัมผัส จึงตัดปัญหาเรื่องไฟสปาร์ค (SPARK) ทำให้มีอายุการใช้งานได้นานกว่าและไม่เกิดสัญญาณรบกวนต่อระบบไมโคร MM-SSR3A สามารถควบคุมด้วยระดับสัญญาณระดับ LOGIC TTL ได้โดยตรง (0-5VDC) จึงเหมาะอย่างยิ่งสำหรับงานควบคุมที่ใช้ระบบดิจิตอล หรือไมโครคอนโทรลเลอร์

คุณสมบัติ

1. สามารถใช้กับไฟ 220 VAC กระแส 3 A (ตัว TRIAC ใช้รุ่นกระแส 6 A)
2. การตัดต่อจะกระทำที่แรงดัน INPUT ระดับ 0 (Zero Voltage Crossing)
3. แรงดัน INPUT DC 5 V กระแส 10 mA
4. ขนาดบอร์ด กว้าง 3.1 cm ยาว 7.7 cm สูง 2.0 cm




รูปที่ ข-1 วงจร 3A Solid State Relay

<u>หมายเหตุ</u>

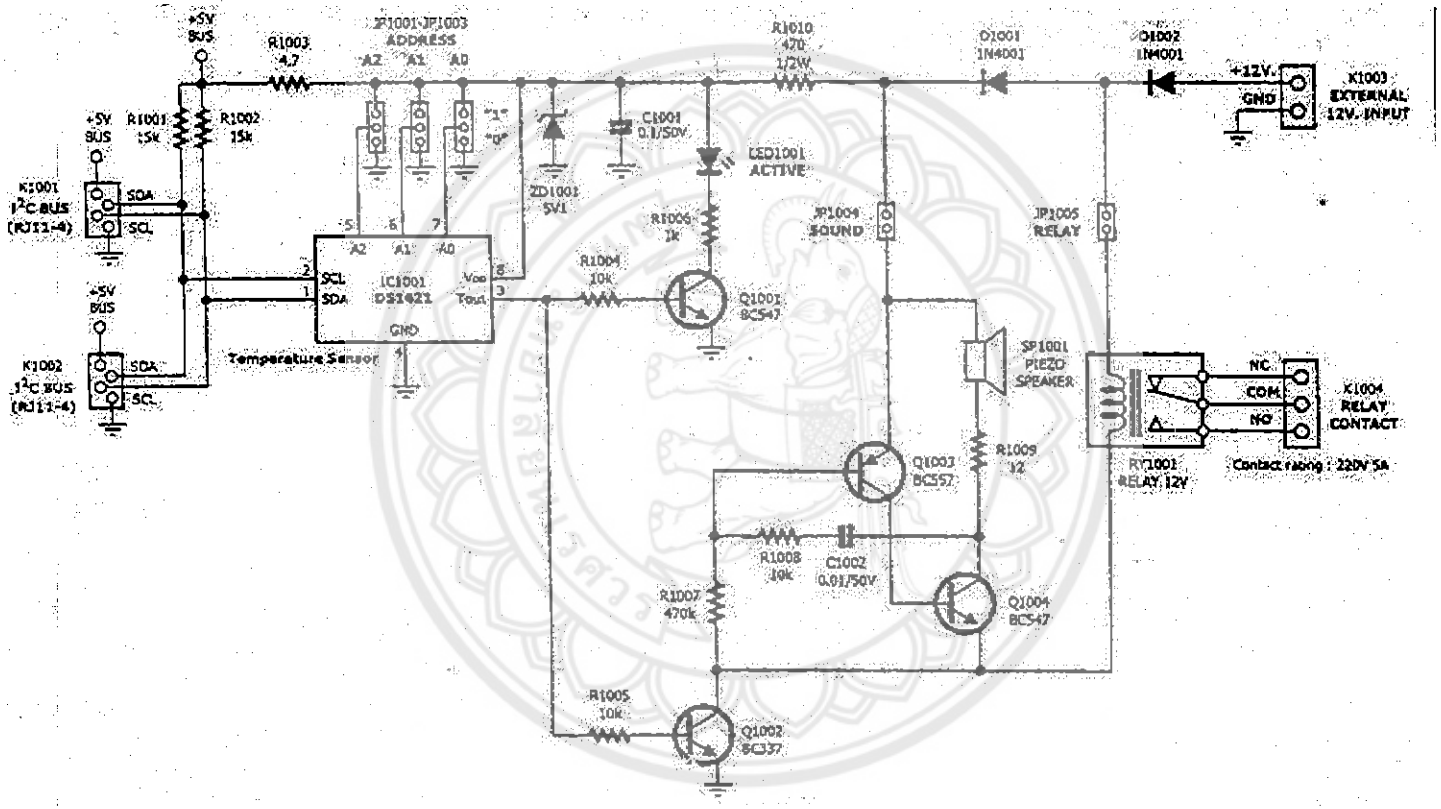
1. กรณีใช้กับ 8255 PORT สามารถต่อแบบ ACTIVE HIGH ได้ และการต่อกับ PORT ของ MCS-51 ให้ใช้แบบ ACTIVE LOW

2. ในกรณีต่อควบคุมโหลดกระดิ่งไฟฟ้าให้ต่อหลอดไฟขนาดเล็กขนานกับกระดิ่งไว้ด้วย จึงจะทำให้กระดิ่งทำงานได้

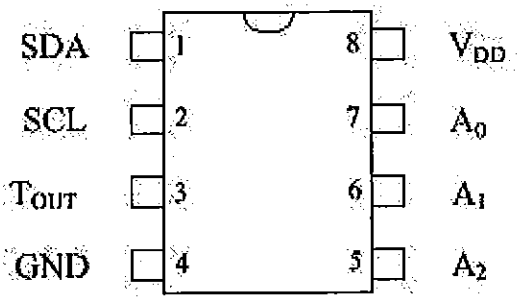
3. ส่วนในกรณีการต่อเมื่อนำไปใช้กับไฟหรือกระแสที่สูงกว่านี้ สามารถต่อได้โดยใช้ Magnetic Relay เข้าช่วย

## ภาคผนวก ค

# วงจรของบอร์ดวัดอุณหภูมิผ่านระบบบัส $I^2C$




**รูปที่ ค-1** วงจรของบอร์ดวัดอุณหภูมิผ่านระบบบัส $I^2C$




**รูปที่ ค-2** ขาของ DS1621 ไอซีวัดอุณหภูมิแบบดิจิตอล

**ตารางที่ ค-1** การจัดขาและการทำงานของ DS1621 ไอซีวัดอุณหภูมิแบบดิจิตอล

| ขาที่ | ชื่อขา | หน้าที่การทำงาน |
|---|---|---|
| 1 | SDA | ขาข้อมูลอนุกรมสำหรับเชื่อมต่อกับระบบบัส I2C |
| 2 | SCL | ขาสัญญาณนาฬิกาสำหรับเชื่อมต่อกับระบบบัส I2C |
| 3 | $T_{OUT}$ | "แอกทีฟ" เมื่ออุณหภูมิสูงถึงจุดกระตุ้นสูง (TH)<br>"รีเซต" เมื่ออุณหภูมิลดลงต่ำกว่าจุดกระตุ้นต่ำ (TL) |
| 4 | GND | ขาต่อกราวด์ของวงจร |
| 5-7 | A2-A0 | ขา Address |
| 8 | +Vcc | ขาต่อไฟเลี้ยง |

**คำสั่งในการใช้งาน DS1621**

การอ่านและการเขียนข้อมูลกับ DS1621 มีรูปแบบต่างๆกันไปตามลักษณะข้อมูลและคำสั่งในการเขียนข้อมูลหรือคำสั่งไปยัง DS1621 อุปกรณ์มาสเตอร์จะกำหนดค่าแอดเดรสของ DS 1621 ตัวที่ต้องการติดต่อ แล้วตามด้วยการเซทบิต R/W จากนั้นจะรอการตอบรับจากอุปกรณ์สเลฟ ซึ่งก็คือ DS1621 เมื่อได้รับการตอบรับแล้ว อุปกรณ์มาสเตอร์จะจัดเตรียมคำสั่งเพื่อใช้งาน DS1621 ต่อไป โดยทุกครั้งที่มีการเขียนคำสั่งไปยัง DS1621 ต้องเคลียร์บิต R/W ด้วยเสมอ เมื่อ DS1621 ได้รับคำสั่งจากอุปกรณ์มาสเตอร์ก็จะดำเนินการจามที่กำหนดมาจากอุปกรณ์มาสเตอร์ และทุกครั้งจะต้องส่งสัญญษณรับรู้หรือตอบรับกลับไปยังอุปกรณ์มาสเตอร์ สำหรับคำสั่งทั้งหมดของ DS 1621 มีดังนี้

**คำสั่งอ่านอุณหภูมิ (Read Temperature : ข้อมูลคำสั่งเป็น AAH)**

เป็นคำสั่งอ่านค่าอุณหภูมิล่าสุดที่ DS 1621 แปลงเสร็จสมบูรณ์ โดยข้อมูลมีขนาด 2 ไบต์

**คำสั่งเข้าถึงรีจิสเตอร์กำหนดการทำงานและแสดงสถานะ(Access Config : ข้อมูลคำสั่งเป็น ACH)**

เป็นคำสั่งสำหรับเข้าถึงหรือติดต่อกับรีจิสเตอร์กำหนดการทำงานและแสดงสถานะ (config) ถ้าหากบิต R/W ถูกกำหนดให้เป็น "0" เป็นการระบุว่าต้องการเขียนข้อมูลไปยังรีจิสเตอร์เพื่อ

กำหนดลักษณะการทำงานของ DS1621 โดยข้อมูลที่เขียนนั้นมีขนาด 1 ไบต์ แต่ถ้าหากบิต R/W เป็น"1" จะเป็นการเข้าถึงเพื่ออ่านค่าของรีจิสเตอร์เพื่อตรวจสอบสถานะการทำงานของ DS1621

**คำสั่งอ่านตัวนับ(Read Counter : ข้อมูลคำสั่งเป็น A8H)**

เป็นคำสั่งอ่านค่าตัวนับที่ใช้ในการแปลงค่าอุณหภูมิเมื่อใช้คำสั่งนี้บิต R/W เป็น "1" เท่านั้น

**คำสั่งอ่านค่าตัวนับละเอียด(Read Sloe : ข้อมูลคำสั่งเป็น A9H)**

เป็นคำสั่งสำหรับอ่านค่าตัวนับละเอียด ในกรณีที่ต้องการให้แสดงรายละเอียดค่าของอุณหภูมิ ซึ่งเก็บอยู่ในตัวนับละเอียด (Slope counter) ในการใช้งานคำสั่งนี้ต้องกำหนดให้บิต R/W เป็น "1"

**คำสั่งเริ่มต้นแปลงค่าอุณหภูมิ(Start Convert T : ข้อมูลคำสั่งเป็น EEH)**

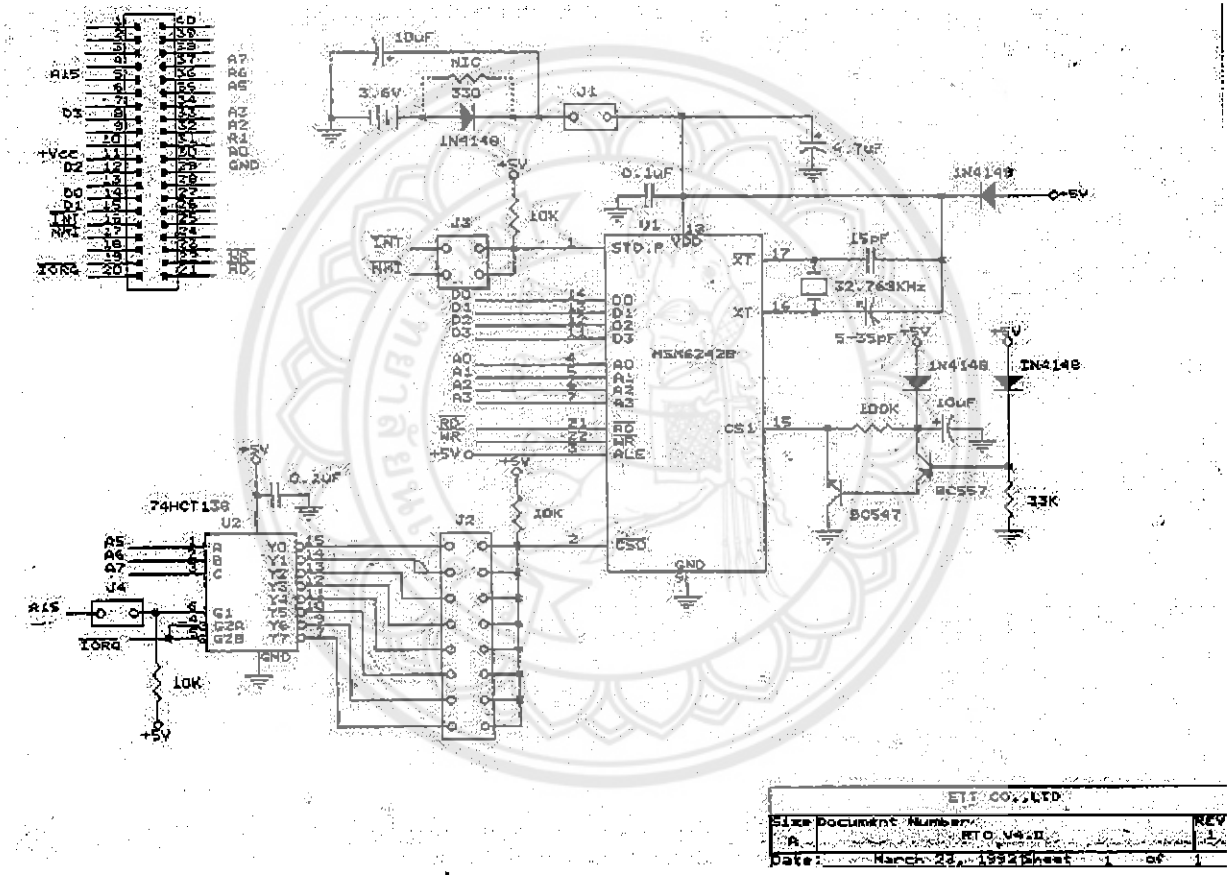
เป็นคำสั่งที่กำหนดให้ DS1621 เริ่มต้นการแปลงค่าอุณหภูมิ

**คำสั่งหยุดการแปลงค่าอุณหภูมิ(Stop Convert T : ข้อมูลคำสั่งเป็น 22H)**

เป็นคำสั่งที่กำหนดให้ DS1621 หยุดการแปลงค่าอุณหภูมิ มักจะใช้เมื่อต้องการให้ DS1621 หยุดทำการวัดและแปลงค่าอุณหภูมิอย่างต่อเนื่อง หลังจากที่ DS1621 กระทำคำสั่งนี้ ค่าอุณหภูมิที่แปลงได้ล่าสุดจะยังคงอยู่ จนกว่าจะที่การกำหนดให้ DS1621 กลับมาเริ่มต้นแปลงค่าอุณหภูมิใหม่

# ภาคผนวก ง

# วงจรบอร์ด ET-RTC V4.0




**รูปที่ ง-1** วงจรบอร์ด ET-RTC V4.0

**REGISTER ต่างๆ ของ RTC V4.0**

S1,S10,MI1,MI10,H1,H10,D1,D10,MO1,MO10,Y1,Y10,W

กลุ่มอักษรเหล่านี้เป็นชื่อย่อของรีจิสเตอร์ตามลำดับ คือ

SECOND1,SECONE10,MINUTE1,MINUTE10,HOUR1.HOUR10,DAY1,DAY10,MONTH1, MONTH10,YEAR1,YEAR10,และ WEEK ในการกำหนดค่าให้รีจิสเตอร์เหล่านี้จะต้องให้เป็นรหัส BCD ตัวอย่างเช่น รีจิสเตอร์ S1(S8,S4,S2,S1) = 1001 ซึ่งจะหมายถึง 9 วินาที

**CD REGISTER (CONTROL D REGISTER)**

HOLD (D0) เมื่อเซตบิตนี้เป็น "1" สัญญาณ CLOCK 1 Hz ที่จะเข้ามาที่ S1 จะถูกหยุดไว้ในเวลานี้เองบิต D1 (Busy) ซึ่งเป็นบิตสถานะจะสามารถอ่านได้ เมื่อBUSY เท่ากับ "0" รีจิสเตอร์ S1~W สามารถอ่านเขียนได้ ถ้าในช่วงเวลานี้มีตัวทดเกิดขึ้นที่วงจรนับของ S1 จะมีผลทำให้หลักหน่วยของวินาทีมีค่าเพิ่มขึ้นอีก 1 วินาที หลังจากที่ HOLD = 0 เมื่อ CS1 = 0 จะทำให้ HOLD = 0 โดยไม่สนสภาวะการณ์อื่นๆ

BUSY (D1) เป็นบิตที่แสดงสถานะของการอินเตอร์เฟสกับ ไมโครคอนโทรลเลอร์ บิตนี้สามารถอ่านได้อย่างเดียวเท่านั้น เมื่อบิต BUSY = 0 หมายถึงพร้อมที่จะให้อ่านหรือเขียนกับรีจิสเตอร์ S1 W(ADDRESS 0 )

IRQ FLAG (D2 ) สถานะนี้จะสัมพันธ์กับระดับสัญญาณของขา STD.P คือเมื่อ STD.P = 0 แล้ว IRQ FLAG จะเท่ากับ "1" และถ้า STD.P เป็น "1" บิต IRQ FLAG จะเป็น "0" IRQ FLAG จะเป็นตัวบอกไมโครคอนโทรลเลอร์ว่าการ INTERRUPT เกิดจาก MSM6242B (เมื่อ IRQ FLAG = 1) บิตสถานะ IRQ FLAG จะทำงานร่วมกับรีจิสเตอร์ตัวอื่นอีก

30 ADJ (D3) ถ้าเซตบิตนี้ให้เป็น "1" ในขณะที่เราเซตเวลานั้น ถ้าหลักวินาทีนับไปได้น้อยกว่า 30 วินาที จะมีผลทำให้หลักวินาทีถูกเซตเป็น "00" แต่ถ้าเกิดหลักวินาทีนับไปได้มากกว่าหรือเท่ากับ 30 วินาที จะมีผลทำให้หลักวินาทีเพิ่มค่าขึ้นอีก 1 นาที แล้วหลักวินาทีก็จะถูกเซตให้เป็น "00" ในขณะที่เซตบิตนี้เป็น "1" นั้นไม่ควรที่จะอ่านหรือเขียนในเวลา 125 us หลังจากนั้นบิตนี้จะถูกเซตเป็น "1" มันจะเปลี่ยนกลับเป็น "0" อย่างอัตโนมัติหลังจากนั้นก็สามารถอ่านหรือเขียนข้อมูลในรีจิสเตอร์ได้

**CE REGISTER (CONTROL E REGISTER)**

MASK (D0) เป็นบิตที่ใช้ควบคุมเอาท์พุต ของ STD.P เมื่อ MASK = 1 จะมีผลทำให้ STD.P = 1(OPEN) คือไม่สามารถใช้บิตอื่นมาเปลี่ยนสภาวะของ STD.P ได้ และเมื่อ MASK = 0 ก็จะทำให้ STD.P = OUTPUT MODE นั่นก็คือ บิตอื่นๆสามารถควบคุมเอาท์พุตของ STD.P ได้ตามต้องการ

INTERPT/STND (D1) ใช้เป็นตัวเลือกสัญญาณเอาท์พุตของ STD.P ได้ 2 โหมด คือ INTERRUPT กับSTANDARD TIMING WAVEFORMS

T0 (D2),T1 (D3) บิตทั้งสองนี้เป็นตัวกำหนดคาบเวลาของสัญญาณเอาท์พุตของ STD.P ทั้ง 2 โหมด

**CF REGISTER (CONTROL F REGISTER)**

REST (D0) บิตนี้จะใช้ในการ CLEAR CLOCK ภายในที่ใช้ในการนับ/หารของวินาที เมื่อ REST = 1 จะทำให้ STD.P = 1 และวงจรนับภายในจะถูก RESET และเมื่อต้องการให้วงจรนับภายในทำงานต่อ (ออกจากการ RESET) จำเป็นจะต้องให้ REST = 0 ถ้า CS1 = 1 ดังนั้น REST =1 อย่างอัตโนมัติ

STOP (D1) จะใช้ในการหยุดตัวทดที่จะเข้าไปในวงจรหารความถี่ 8192 Hz และมีการหน่วงเวลาไป 122 us ก่อนที่เวลาจะทำการเดินหรือหยุดเดิน หลังจากที่มีการเปลี่ยนสภาวะของ FLAG นี้ เป็น " 1 " = STOP / เป็น " 0 " = RUN ในขณะที่จะเซตเวลาให้ RTC นั้นควรให้บิตนี้เป็น " 1 " เพื่อไม่ให้ตัวทดเข้ามาที่หลักวินาที หลังจากเซตเวลาให้ RTC เสร็จแล้วจึงให้บิตนี้เป็น " 0 " เพราะว่าในขณะที่เราเซตเวลานั้นเกิดมีตัวทดเข้ามาจะทำให้หลักวินาทีเพิ่มค่าขึ้นอีก 1 วินาที หลังจากที่เราเซตเวลาแล้ว

24/12 (D2) บิตนี้เป็นการเลือกว่าจะใช้เวลาเดินแบบ 24 ชม.หรือ 12 ชม. มี (AM/PM) ถ้าเลือกโหมด 1- 24 ชม. บิต PM/AM จะไม่ถูกนำใช้ (มีค่าเป็น " 0 ") แต่ถ้าเลือกโหมด 0-12 ชม. บิต PM/AM จะมีการเปลี่ยนสภาวะไปด้วยสำหรับการเซตบิตมีขั้นตอนดังนี้

1. ต้องให้ REST BIT = 1
2. 24/12 HOUR BIT = 0 หรือ 1 ถ้าเป็น " 0 " หมายถึงโหมด 24 ชม. ถ้าเป็น " 1 " หมายถึงโหมด 24 ชม.
3. REST BIT = 0

หมายเหตุ REST จะต้องเป็น " 1 " ถึงจะเขียนบิต 24/12 ได้

TEST (D3) เมื่อบิตนี้เป็น " 1 " อินพุตของวงจรนับในหลักวินาทีจะมาจากวงจรนับ/หารแทนที่จะมาจากภาคหาร 15 ดังนั้นวงจรนับหลักวินาทีจะนับความถี่ที่ 5.4163 Hz แทน (ปกติจะนับที่ความถี่ 1 Hz) เมื่อ TEST = 1(TEST MODE) บิต STOP และบิต REST จะไม่ต้องถูกเซต ในขณะที่อยู่ใน TEST MODE (TEST = 1) ถ้า HOLD = 1 วงจรนับภายในจะถูกหยุดไว้ แต่เมื่อ HOLD กลับมาเป็น " 0 " จะไม่รับรองว่าที่ได้จะถูกต้อง

# ภาคผนวก จ

# Source Code

## Function read temperature

```
void i2c_high(void)  //tricker SCL high
{
  SCL=1;
  delay(j);
}
void i2c_low(void) //tricker SCL low
                    {
  SCL=0;
  delay(j);
}
void i2c_NACK()//send master not acknowledge
                          {
  SDA=1;
  delay(j);
  i2c_high();
  i2c_low();
  SCL=1;
}
void i2c_start(void) //Start the i2c bus
                          {
  SDA=1;
  i2c_high();
  SDA=0;
  delay(j);
  i2c_low();
  SDA=1;
```

```
}
void i2c_stop(void) //stop the i2c bus
                    {
  SDA=0;
  i2c_high();
  SDA=1;
}
bit i2c_wrbyte(unsigned dat) //write data to slave
{
  xdata unsigned char i;
  bit outbit;
  for (i=1;i<=8;i++)  //for loop 8 bit
        {
    outbit=dat&0x80;                    //check bit 1  or 0
    SDA=outbit;
    dat=dat<<1;
    i2c_high();              //pull for send data
    i2c_low();
    }
  SDA=1;
  i2c_high();
  outbit=SDA;
  i2c_low();
  return(outbit);
}
unsigned char i2c_rdbyte(void) //read 1 byte from sensor
        {
  xdata unsigned char i,dat;
  bit inbit;
  dat=0;
  for (i=1;i<=8;i++)
    {
```

```
    i2c_high();
    inbit=SDA;
    dat=dat<<1;
    dat=dat|inbit;
    i2c_low();
    }
  SDA=0;
  delay(j);
  SCL=1;
  delay(j);
  SCL=0;
  delay(j);
  SDA=1;

// delay(j);
// i2c_high();
// inbit=SDA;
// i2c_low();
// SCL=1;
// if (~inbit) dat=0xff;
  return(dat);
}
unsigned char i2c_rdbyteL(void)                    //read 1 byte temperature from sensor
{
  xdata unsigned char i,dat;
  bit inbit;
  dat=0;
  for (i=1;i<=8;i++)
    {
    i2c_high();
    inbit=SDA;
    dat=dat<<1;
```

```
    dat=dat|inbit;
    i2c_low();
    }
// SDA=1;
// delay(j);
  SCL=1;
  delay(j);
  SCL=0;
  delay(j);
// SDA=0;
// SDA=1;
// delay(j);
// i2c_high();
// i2c_low();
// SCL=1;
// delay(50);
// SCL=0;
// i2c_NACK();
  return(dat);
}
void StartConv(void)                    //convert temperature to sensor
{
  i2c_start();
  i2c_wrbyte(0x90);
  i2c_wrbyte(0xee);
  i2c_stop();
}
float ReadTemp(void)                    //read temperature from sensor
{
  xdata int datL;
  xdata float datH;
```

```
  i2c_start();                          //start i2c
  i2c_wrbyte(0x90);                     //send address to sensor
  i2c_wrbyte(0xaa);                     //send command to read register
  i2c_start();                          //start i2c
  i2c_wrbyte(0x91);                     //send command to read register
  datH=i2c_rdbyte();                    //read 1 byte MSB and store to datH
  datL=i2c_rdbyteL();                   //read 1 byte LSB and store to datL
  SDA=0;
  i2c_high();
  SDA=1;
//  datL=0x80;
//  i2c_NACK();
//  i2c_stop();
//  ftoa(datL,StrTmp);
//  string_to_lcd(2,1,StrTmp);          //show temp on LCD

   if ((datL&0x80)==0x80)               //check LSB to add 0.5
    {
      datH = datH+0.5;                 //+0.5 to temp
   }

  return(datH);                         //return datH tofunction

}
```

**Check temperature to control temp**

```
if (TMP>=temp_date)                     //check temp from ds1621 and temp_date
    {
     // string_to_lcd(2,10,"CLOSE");     //send state for close solid state
      Pb8255=0xff;
    }
```

```
 else
    {
     // string_to_lcd(1,10,"OPEN");
       Pb8255=0x00;                          //send state for  open solid state
    }
```

**Control Motor**

```
void turn_motor(void)                        //function turn motor up&down
{
   sw1=Pa8255;                               //receive bit switch from Pa
     if ((sw1&0x01)==0x01)                   //motor touch switch 1
       {
       Pc8255=0xff;                          //send state turn motor down
       delay(10000);
       }
   if ((sw1&0x08)==0x08)                     //motor touch switch 4

       {
       Pc8255=0x00;                          //send state turn motor up
       delay(10000);
        }
}
void Stop_Motor(void)                        //stop motor
{
  sw1=Pa8255;                                //receive bit from Pa
   if ((sw1&0x01)==0x01)                     //Stop  when touch switch 1
       {
       Pc8255=0x02;                          //send state stop motor
       }
   if ((sw1&0x08)==0x08)                     //Stop  when touch switch 4
       {
         Pc8255=0x02;                        //send state stop motor
```

```
        }
    }
void Set_Home(void)                    //set motor to start position
{
        Pc8255=0x02;                   //send state turn motor stop before check
        Pc8255=0x00;                   //send state turn motor up to set home
       if ((sw1&0x01)==0x01)           //check switch for stop
         {
           Pc8255=0x02;                //send state for stop
         }
```

**Show time and store data of time**

```
if ( ( (CD&0x04)!=0) )                 //check CD register
     {
//            delay(100);

              DataS1=CD;
              CD=DataS1&0xfb;
              delay(100);
             //  Pa=S1;

              DataS2=S10;
              DataS1=S1;
              DataS1=DataS1&0x3f;
              DataS2=DataS2&0x3f;

              DataM2=Mi10;
              DataM1=Mi1;
              DataM1=DataM1&0x3f;
              DataM2=DataM2&0x3f;
```

```
DataH2=H10;
DataH1=H1;
DataH1=DataH1&0x3f;
DataH2=DataH2&0x3f;


WCharxy(1,9,DataH2);
WCharxy(1,10,DataH1);

 WCharxy(1,11,':');

WCharxy(1,12,DataM2);
WCharxy(1,13,DataM1);

WCharxy(1,14,':');

WCharxy(1,15,DataS2);
WCharxy(1,16,DataS1);

//date

DataD2=D10;
DataD1=D1;


DataD1=DataD1&0x3f;
DataD2=DataD2&0x3f;

DataMo2=Mo10;
DataMo1=Mo1;
DataMo1=DataMo1&0x3f;
DataMo2=DataMo2&0x3f;
```

```
            DataY2=Y10;
            DataY1=Y1;
            DataY1=DataY1&0x3f;
            DataY2=DataY2&0x3f;
            // WCharxy(2,1,DataD2);
           //  WCharxy(2,2,DataD1);
           //   WCharxy(2,3,'/');
           //  WCharxy(2,4,DataMo2);
           //  WCharxy(2,5,DataMo1);
      }//if cd04
```

**Check serial port**

```
RI = 0;                                  //check bit RI from rs232
        mBUF =SBUF;                      //send SBUF register to mBUF
        mess[0] =mBUF;                   //send mBUF to array
if (mess[0] == 'F')                      //check  SBUF  for  receive
       {
        //        recv_message(DATA_IN);  //use function to read SBUF
       // for (f=0;f<=8;f++)
       //  {
       //     if ()

         // mBUF = SBUF;
         //  DATA_IN[f]=mBUF;
        // }
       //DATA_IN[9]='\0';
          //     string_to_lcd(2,8,DATA_IN);

       }//end if mess F
```

# ประวัติผู้เขียนโครงงาน

ชื่อ นายภัคพงษ์ หาญใจ

เกิดวันที่ 10 กันยายน 2524

ภูมิลำเนา 37 หมู่ 5 ต.แม่แฝกใหม่ อ.สันทราย จ.เชียงใหม่

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนปรินส์รอยแยลส์วิทยาลัย
- จบระดับมัธยมศึกษาจากโรงเรียนปรินส์รอยแยลส์วิทยาลัย
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-Mail : **Best_cmu@hotmail.com**

ชื่อ นายโชติพงษ์ เรืองฤทธิ์

เกิดวันที่ 13 กุมภาพันธ์ 2524

ภูมิลำเนา 203 ม.2 ต.บ้านตึก อ.ศรีสัชนาลัย จ.สุโขทัย

ประวัติการศึกษา

- จบระดับประถมศึกษาจากโรงเรียนเทศบาลท่าอิฐ
- จบระดับมัธยมศึกษาจากโรงเรียนอุตรดิตถ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

e-Mail : **Chotepong@hotmail.com**