สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# เครื่องให้อาหารปลาอัตโนมัติ

## Automatic Fish Feeding Machine

T055679

โดย

นายปราโมทย์ แพทย์พันธ์

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลขหมู่..........
เลขทะเบียน 55679
วัน,เดือน,ปี 24 พ.ค. 2548

# เครื่องให้อาหารปลาอัตโนมัติ

Automatic Fish Feeding Machine


โดย

นายปราโมทย์ แพทย์พันธ์

อาจารย์ที่ปรึกษา

อาจารย์ถาวร เบญจนราสุทธิ์


ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2546


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2546

สาขาวิชาระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องให้อาหารปลาอัตโนมัติ

Automatic Fish Feeding Machine

ผู้จัดทำ

นายปราโมทย์ แพทย์พันธ์ 44015290

........................................ อาจารย์ที่ปรึกษา

(อาจารย์ถาวร เบญจนราสุทธิ์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เครื่องให้อาหารปลาตู้อัตโนมัติ

โดย
นายปราโมทย์ แพทย์พันธ์

อาจารย์ที่ปรึกษา
อาจารย์ถาวร เบญจนราสุทธ์

ปีการศึกษา 2546

**บทคัดย่อ**

ในปัจจุบันเทคโนโลยีเข้ามามีบทบาทอย่างมาก เพื่อตอบสนองความต้องการทางด้านอุปโภคของมนุษย์ จึงมีการนำเทคโนโลยีเข้ามาใช้เพื่อให้ความสะดวกแก่ผู้คน เครื่องให้อาหารปลาอัตโนมัติเช่นกันที่ช่วยอำนวยความสะดวก แก่ผู้ที่เลี้ยงปลาตู้

ส่วนประกอบของเครื่องให้อาหารปลาประกอบด้วย ส่วนควบคุม และส่วนกลไก โดยในการควบคุมจะใช้ MCS-51 เป็นตัวควบคุมการทำงาน ซึ่งมีหลักการทำงานดังนี้ เริ่มต้นลิ้นตัวที่ 1 จะเปิดออกเพื่อปล่อยอาหารเข้าช่องวัดปริมาณอาหารจนถึงระดับที่อุปกรณ์ตรวจวัดระดับติดตั้งไว้ หลังจากนั้นลิ้นตัวที่ 1 ก็จะปิด และเมื่อถึงเวลาที่กำหนดลิ้นตัวที่ 2 ก็จะค่อยๆ เปิดปล่อยอาหารออกจนหมด และอาหารถูกจ่ายจนหมด ลิ้นตัวที่ 2 ก็จะปิด แล้วจะเริ่มทำงานใหม่เมื่อถึงเวลาที่กำหนด

เครื่องให้อาหารปลาสามารถให้อาหารปลาได้ตามต้องการทั้งปริมาณและเวลาที่กำหนด อย่างไรก็ตาม ยังมีความผิดพลาดในการทำงานในส่วนปริมาณอาหาร กล่าวคือ ปริมาณอาหารที่จ่าย มีความผิดพลาดเฉลี่ยประมาณ 4.22%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# Automatic Fish Feeding Machine

By

Mr. Pramote Pedpun

Advisor

Taworn Benjanarasuth

Year 2003

ABTRACT

Currently, technology plays an important role in response to consumer's needs. Also, advance technology is made use for more convenience among people. The automatic fish feeding mahine is developed for those in fish feeding.

The automatic fish feeding machine consists of control part and mechanism part. Microcontroller MCS-51 is used in this project. The working process of the automatic fish feeding machine starts with the opening of the first tongue in order to let the food feeding into slot. As the food is reached the required level detected by the sensor, the first slot will be closed. When reaching the time fixed, the second tongue then gradually lets the food out. After all the food in the slot is emptied, the second tongue is closed. The working process will start again at the specified time.

The machine can work efficiently; however, there is an error related to amount of the food approximately 4.22 % of the average level.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

**ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จได้ด้วยดี ผู้เขียนขอกราบขอบพระคุณ อาจารย์ถาวร เบญจนราสุทธิ์ อาจารย์ที่ปรึกษาเป็นอย่างสูง ที่ได้ให้คำปรึกษาแนวทางในการแก้ปัญหา ตลอดจนการฝึกฝนให้ผู้เขียนมีความสามารถในการทำวิจัย ได้อย่างมีประสิทธิภาพ รวมถึงถามความก้าวหน้าอย่างสม่ำเสมอ ทำให้ผู้เขียนทำงานอย่างมีระบบ และใช้เวลาอย่างมีประสิทธิภาพ**

ขอกกราบขอบพระคุณอาจารย์ทุกๆ ท่านในภาควิชาวิศวกรรมระบบควบคุม ที่กรุณาให้คำปรึกษาและให้ปัญญาในการแก้ปัญหาต่างๆ รวมถึงแง่คิดที่เป็นประโยชน์ต่อตัวผู้เขียนมาก

และท้ายที่สุดขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ที่รักเคารพ ที่สนับสนุนให้โอกาสแก่ผู้เขียนได้เล่า เรียนจนถึงวันนี้ ตลอดจนเพื่อนๆ พี่ๆ น้องๆ ที่ได้ให้คำปรึกษา แนะนำ และเป็นกำลังใจให้ด้วยดีเสมอมา จนสามารถนำสิ่งต่างๆ ที่ได้รับนั้นมาจัดทำปริญญานิพนธ์ฉบับนี้ให้เสร็จสมบูรณ์

ด้วยคำแนะนำและคำปรึกษาจากทุกคนทำให้ปริญญานิพนธ์ฉบับนี้ให้เสร็จสมบูรณ์ลุล่วง คุณค่าและคุณประโยชน์ อันพึงมีจากปริญญานิพนธ์ฉบับนี้ ผู้เขียนขอมอบแด่ผู้มีพระคุณทุกท่าน

ปราโมทย์ แพทย์พันธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

| | หน้า |
|---|---|
| **บทคัดย่อภาษาไทย** | I |
| **บทคัดย่อภาษาอังกฤษ** | II |
| **กิตติกรรมประกาศ** | III |
| **สารบัญ** | IV |
| สารบัญรูปประกอบ | VI |
| สารบัญตารางประกอบ | VII |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความเป็นมาและความสำคัญของโครงงานปริญญานิพนธ์ | 1 |
| 1.2 วัตถุประสงค์ของโครงงานปริญญานิพนธ์ | 1 |
| 1.3 ขอบเขตของโครงงานปริญญานิพนธ์ | 2 |
| 1.4 รายละเอียดของโครงงานปริญญานิพนธ์ | 2 |
| บทที่ 2 ทฤษฎีและหลักการ | 3 |
| 2.1 ไมโครคอนโทรลเลอร์ ( MCS-51) | 3 |
| 2.2 ไอซีสร้างฐานเวลาจริง (Real time clock) | 7 |
| 2.3 จอแสดงผลโมดูล (LCD) | 12 |
| 2.4 สวิทช์แบบเมตริกซ์หรือคีย์แพด(Keypad) | 18 |
| 2.5 ตัวส่งแสงอินฟาเรดและตัวรับแสงอินฟาเรด | 20 |
| 2.6 มอเตอร์กระแสตรง | 22 |
| บทที่ 3 ขั้นตอนและวิธีดำเนินการ | 23 |
| 3.1 การทำงานเครื่องให้อาหารปลา | 23 |
| 3.2 ลำดับการทำงานของเครื่องให้อาหารปลา | 24 |
| 3.3 การออกแบบเครื่องให้อาหารปลา | 25 |
| 3.4 วิธีใช้เครื่องให้อาหารปลา | 32 |
| **บทที่ 4 การทดลอง** | 35 |
| **4.1 การทดลองเพื่อทดสอบเรื่องปริมาณอาหาร** | 35 |
| **4.2 การทดลองเพื่อทดสอบเรื่องเวลา** | 38 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)


| | หน้า |
|---|---|
| บทที่ 5 สรุปผลการทดลองและวิจารณ์ | 39 |
| 5.1 สรุปผล | 39 |
| 5.2 ปัญหาและวิธีแก้ไข | 39 |
| 5.3 ข้อเสนอแนะ | 40 |
| บรรณานุกรม | 41 |
| ภาคผนวก | |
| ภาคผนวก ก. โปรแกรมการทำงานของไมโครคอนโทรลเลอร์(MCS- 51) | 42 |
| ภาคผนวก ข. คู่มือของโฟโต้เซนเซอร์ | 51 |


สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูปประกอบ

หน้า

**รูปที่** 2.1 แสดงการจัดตำแหน่งขาต่างๆ ของไมโครคอนโทรลเลอร์(MCS-51) 4
**รูปที่ 2.2 แสดงการจัดขาของไอซีสร้างฐานเวลาจริง (DS1307)** 8
รูปที่ 2.3 แสดงโครงสร้างภายในของไอซีสร้างฐานเวลาจริง (DS1307) 9
รูปที่ 2.4 การจัดพื้นที่ของหน่วยความจำภายในไอซีสร้างฐานเวลาจริง (DS1307) 10
รูปที่ 2.5 แสดงรูปของการเขียนข้อมูลติดต่อกับไอซีสร้างฐานเวลาจริง (DS1307) 11
รูปที่ 2.6 แสดงรูปของการอ่านข้อมูลติดต่อกับไอซีสร้างฐานเวลาจริง (DS1307) 12
รูปที่ 2.7 แสดงขาต่าง ๆ ของจอแสดงผล LCD แบบ 16 ตัวอักษร 2 บรรทัด 13
รูปที่ 2.8 แสดงวงจรของสวิทซ์แบบเมตริกซ์หรือคีย์แพด 18
รูปที่ 2.9 แสดงการเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์(MCS-51) 19
รูปที่ 2.10 การไบอัสตรงของไดโอด 21
รูปที่ 3.1 บล็อกไดอะแกรมเครื่องให้อาหารปลา 23
รูปที่ 3.2 แผนผังการทำงานของเครื่องให้อาหารปลา 24
รูปที่ 3.3 รูปโครงสร้างหัวจ่ายอาหาร 26
รูปที่ 3.4 แสดงการต่อคีย์สวิตซ์แบบเมทริกซ์กับไมโครคอนโทรลเลอร์ 27
รูปที่ 3.5 การเชื่อมต่อจอแสดงผล LCD กับไมโครคอนโทรลเลอร์ 28
รูปที่ 3.6 แสดงการเชื่อมต่อ DS 1307 กับไมโครคอนโทรลเลอร์ 29
รูปที่ 3.7 วงจรตรวจจับโดยใช้แสงอินฟาเรด 30
รูปที่ 3.8 ชุดขับมอเตอร์กระแสตรง 31
รูปที่ 3.9 วงจรไฟเลี้ยง 5 โวลท์ 32
รูปที่ 3.10 รูปเครื่องให้อาหารปลา 32
รูปที่ 3.11 แสดงวิธีใช้เครื่องให้อาหารปลา 34
**รูปที่ 4.1 แสดงโครงสร้างหัวจ่ายอาหาร** 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

หน้า

ตารางที่ 2.1 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์ 4

ตารางที่ 2.2 แสดงหน้าที่พิเศษของแต่ละขาพอร์ต 3 6

ตารางที่ 2.3 แสดงการเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจากขา SWQ 10

ตารางที่ 4.1 ผลการทดลองวัดปริมาณอาหารและความเที่ยงตรงในการจ่ายอาหาร 36

ตารางที่ 4.2 ตารางคำนวณหาค่าผิดพลาด 37

ตารางที่ 4.3 ผลการทดลองของการตั้งเวลาให้อาหารปลา 38

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของโครงงานปริญญานิพนธ์

ปัจจุบันการเลี้ยงปลาตู้สวยงามได้วิวัฒนาการทั้งด้านอุปกรณ์การเลี้ยง ตลอดจนสิ่งอำนวยความสะดวกแก่นักเลี้ยงปลาได้เป็นอันมาก สะดวกสบายอีกทั้งประหยัดเวลาในการดูแลบำรุงรักษาอีกด้วย เพียงแค่ขอให้นักเลี้ยงปลาได้ศึกษาหลักการบางสิ่งบางอย่างที่จำเป็นเท่านั้น คุณก็สามารถเป็นนักเลี้ยงปลาที่ดีได้ อีกทั้งยังเป็นผู้สร้างความสุขและความเพลิดเพลินแก่ครอบครัว

เนื่องจากบางครั้งผู้เลี้ยงมักประสบปัญหาเกี่ยวกับการให้อาหาร ถ้าให้อาหารมากเกินไปปลากินไม่หมด ทำให้อาหารตกค้างในตู้ เกิดผลเรื่องน้ำเน่าเสียตามมา และถ้าให้อาหารน้อยเกินไปปลาไม่พอกิน มีผลต่อการเจริญเติบโตของปลา ดังนั้นผู้เลี้ยงควรอาหารในปริมาณที่เหมาะสม และเวลาที่แน่นอน เราจึงนำแนวคิดนี้ มาประยุกติโดยใช้การควบคุมอัตโนมัติเข้ามาควบคุมการให้อาหารปลาอัตโนมัติ

ดังนั้นโครงงานนี้จึงนำแนวคิดความต้องการอาหารของปลา ทำเครื่องให้อาหาปลาอัตโนมัติ โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุม การให้อาหารแบบอัตโนมัติที่มีประสิทธิภาพ

## 1.2 วัตถุประสงค์ของโครงงานปริญญานิพนธ์

1. ศึกษาการใช้งานอุปกรณ์เซนเซอร์ ตัวส่งแสงอินฟาเรด และตัวรับแสงอินฟาเรด
2. ศึกษาการประมวลผลโดยใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51
3. ศึกษาการควบคุมเวลาด้วย ไอซีสร้างฐานเวลาจริง(Real time clock)
4. ศึกษาการใช้คีย์แพด(Keypad) และการแสดงผลโดยใช้ตัวแสดงผลแบบผลึก เหลว (Liquid Crystal Display)
5. ศึกษาการใช้งานที่ได้จากการทดลอง เพื่อเปรียบเทียบกับการใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 1.3 ขอบเขตของโครงงานปริญญานิพนธ์

1. ใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวประมวลผล
2. เครื่องให้อาหารปลาอัตโนมัติสามารถปรับปริมาณอาหารที่เหมาะสมได้
3. เครื่องให้อาหารปลาอัตโนมัติสามารถตั้งเวลาให้อาหารได้ โดยสามารถกำหนดจำนวนมื้อ 1 – 8 มื้อ

### 1.4 รายละเอียดของปริญญานิพนธ์

ในปริญญานิพนธ์ฉบับนี้ได้แบ่งเนื้อหาออกเป็น 5 บท และภาคผนวกโดยในแต่ละบทมีรายละเอียดดังต่อไปนี้

บทที่ 1 กล่าวโดยรวมในการทำโครงงานปริญญานิพนธ์ถึง บทนำ ความเป็นมา วัตถุประสงค์ ขอบเขต

บทที่ 2 เป็นเนื้อหาเกี่ยวกับทฤษฎีและหลักการ

บทที่ 3 ขั้นตอนและวิธีดำเนินงาน

บทที่ 4 เป็นการทดลองและผลการทดลอง

บทที่ 5 เป็นการสรุปผลการทดลองและวิจารณ์เกี่ยวกับตัวโครงงานส่วนที่ต้องเพิ่ม และแนวทางการพัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2
# ทฤษฎีและหลักการ

## 2.1 ไมโครคอนโทรลเลอร์ MCS-51

### 2.1.1 คุณสมบัติทั่วไปของ ไมโครคอนโทรเลอร์ MCS-51

- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีวงจรออสซิลเลเตอร์และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
- มีขาสัญญาณอินพุตเอาต์พุตจำนวน 32 บิต
- สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก (External data memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- มีหน่วยความจำโปรแกรมภายในตัว (On-chip program memory) ขนาด 4 K โดยเฉพาะเบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K สำหรับเบอร์ 8031 และ M8032 จะไม่มีหน่วยความจำในส่วนนี้
- หน่วยความจำภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่ายส่งผลให้การเขียนโปรแกรมทำได้ง่ายขึ้น
- มีไทม์เมอร์ / เคาน์เตอร์ (Timer/Counters) ขนาด 16 บิต จำนวน 2 ตัวโดยเฉพาะเบอร์ 8052 และ 8032 จะมีไทม์เมอร์ / เคาน์เตอร์จำนวน 3 ตัว
- การอินเตอร์รัปต์สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8052 และ 8032 จะทำการอินเตอร์รัปต์ได้จาก 6 แหล่งกำเนิด
- มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบฟูลดูเพล็กซ์ (Full duplex)
- มีคำสั่งการคำนวณคณิตศาสตร์และทางตรรกศาสตร์
- คำสั่งส่วนใหญ่ใช้เวลาเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิรตซ์
- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว

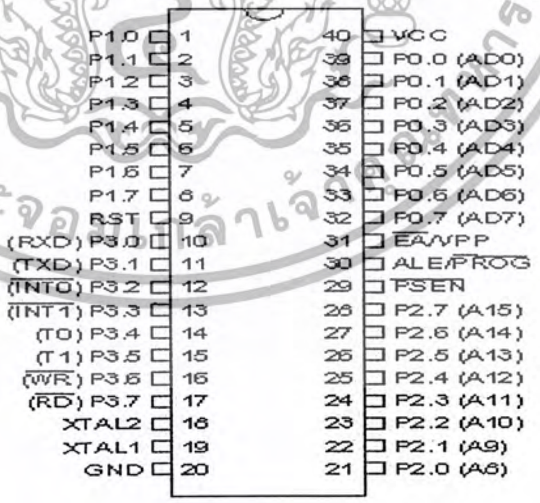
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เบอร์ | หน่วยความจำภายใน | | จำนวนไทเมอร์/เคาน์เตอร์ | อินเตอร์รัพต์หมายเลข |
|---|---|---|---|---|
| | เก็บโปรแกรม | เก็บข้อมูล | | |
| 8052H | 8K x 8 ROM | 256 x 8 ROM | 3 x 16 Bit | 6 |
| 8051H | 4K x 8 ROM | 128 x 8 ROM | 2 x 16 Bit | 5 |
| 8051 | 4K x 8 ROM | 128 x 8 ROM | 2 x 16 Bit | 5 |
| 8032AH | ไม่มี | 256 x 8 ROM | 3 x 16 Bit | 6 |
| 8031AH | ไม่มี | 128 x 8 ROM | 2 x 16 Bit | 5 |
| 8031 | ไม่มี | 128 x 8 ROM | 2 x 16 Bit | 5 |
| 8751H | 4K x 8 EPROM | 128 x 8 ROM | 2 x 16 Bit | 5 |
| 80751H-12 | 4K x 8 EPROM | 128 x 8 ROM | 2 x 16 Bit | 5 |

ตารางที่ 2.1 แสดงคุณสมบัติของไมโครคอนโทรลเลอร์แต่ละเบอร์ในตระกูล MCS-51

### 2.1.2 **โครงสร้างภายนอก** MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นฐานที่เหมือนกันดังแสดงใน รูปที่ 1 สำหรับหน้าที่การใช้งานของแต่ละขามีดังนี้




รูปที่ 2.1 แสดงการจัดตำแหน่งขาต่างๆของไมโครคอนโทรลเลอร์ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา Vcc เป็นขาป้อนแรงไฟเลี้ยง +5 โวลต์
- ขา Vss เป็นกราวด์
- ขารีเซต (RST) ใช้สำหรับการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซตต้องคงสถานะเป็น 1 อย่างน้อย 2 แมชีนไซเกิล ในขณะที่ออสซิลเลเตอร์ทำงานอยู่
- ขา ALE / PROG เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลตช์ (Latch) ค่าตำแหน่งแอด เดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอกและขานี้ยังทำหน้าที่เป็นอินพุตรับพัลส์ในการโปรแกรม (Program pulse input) ในส่วนของความจำ EPROM
- ขา PSEN (Program store enable) ทำหน้าที่เป็นสัญญาณสโตรปเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอกเมื่อไมโครคอนโทรลเลอร์ประมวลผลจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรปจำนวน 2 ครั้ง ในแต่ละแมชีนไซเกิล แต่ในขณะที่ติดต่อกับหน่วยความจำภายนอกจะไม่มีการส่งสัญญาณสโตรปแต่อย่างใด
- ขา EA / VPP (External enable/VPP) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายใน หรือจากภายนอกโดยมีสถานะเป็น 0 จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-0FFFH แต่ถ้าบิตป้องกัน (Security bit) ในหน่วยความจำ EPROM ถูกโปรแกรมไว้ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอก นอกจากนี้ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (Vpp) ขนาด 21 โวลต์ เพื่อใช้ในการระหว่างโปรแกรม EPROM
- ขา XTAL1 และขา XTAL2เป็นขาอินพุตและเอาต์พุตของวงจรอินเวอร์ติ้งออสซิลเลเตอร์แอมปลิไฟ
- ขาพอร์ต 0 (Port 0) มี 8 ขา ได้แก่ ขา P0.0 - P0.7 เป็นขาอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับงาน ใช้งานทั่วไปโดยใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้ขาพอร์ตอยู่ในสภาวะปล่อยลอย ซึ่งในลักษณะนี้สามารถนำมาใช้กำหนดอินพุตอิมพีแดนซ์สูงได้ นอกจากพอร์ตนี้จะทำงานเป็นอินพุตเอาต์พุตก็ยังถูกใช้งานในการติดต่อกับหน่วย ความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์ต่ำ (A0-A7) ซึ่งจะใช้งานเป็นแบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 บิต (D0-D7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขาพอร์ต 1 (Port 1)มีขา 8 ขา ได้แก่ขา P1.0-P1.7 เป็นขาอินพุตเอาต์พุตแบบ 2 ทิศทาง โดยการใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อเป็นพอร์ตอินพุต
- ขาพอร์ต 2 (Port 2)มีขา 8 ขา ได้แก่ขา P2.0-P2.7 เป็นขาอินพุตเอาต์พุตแบบ 2 ทิศทาง โดยการใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานกับหน่วยความจำภายนอกโดยทำหน้าที่กำหนดแอดเดรสไบต์สูง (A8-A15)
- ขาพอร์ต 3 (Port 3)มีขา 8 ขา ได้แก่ขา P3.0-P3.7 เป็นขาอินพุตเอาต์พุตแบบ 2 ทิศทาง โดยการใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ตเพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งาน ดังตารางที่ 2.2

| ขาพอร์ต | หน้าที่พิเศษ |
|---|---|
| P3.0 | RXD (Serial input port) |
| P3.1 | TXD (Serial output port) |
| P3.2 | INT0 (External interrupt 0) |
| P3.3 | INT1 (External interrupt 1) |
| P3.4 | T0 (Timer 0 external input) |
| P3.5 | T1 (Timer 1 external input) |
| P3.6 | WR (External data memory write strobe) |
| P3.7 | RD (External data memory write strobe) |

ตารางที่ 2.2 แสดงหน้าที่พิเศษของแต่ละขาพอร์ต 3 (Port 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.2 ไอซีสร้างฐานเวลาจริง ( Real time clock )

ในโครงการชิ้นนี้ เลือกใช้ DS1307 โดยผู้ผลิตคือ ดัลลัสเซมิคอนดักเตอร์ (Dallas semiconductor) มีหน้าที่สร้างฐานเวลาจริงให้แก่ระบบไมโครคอนโทรลเลอร์ โดย DS1307 จะให้ข้อมูลเกี่ยวกับฐานเวลาจริงทั้งหมด ไม่ว่าจะเป็นค่าของเวลาที่ละเอียดถึงหลักวินาที,นาที,ชั่วโมง, วันที่, วันในสัปดาห์,เดือนและปี โดยสามารถปรับวันเดือนปี ให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปี อธิกสุรทินด้วย

2.2.1 คุณสมบัติทางเทคนิค

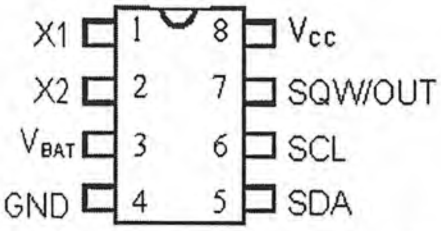
- เป็นไอซีรีลไทม์คล็อกให้ข้อมูลตั้งแต่วินาทีจนถึงปีรวมถึงการกำหนดวันในปีอธิกสุรทินด้วยสามารถให้ข้อมูลเวลาได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100
- มีหน่วยความจำนอนโวลาไทล์แรม 56 ไบต์อยู่ภายใน สามารถใช้เก็บข้อมูลทั่วไปได้
- ใช้การเชื่อมต่อแบบระบบบัส $I^2C$
- มีวงจรตรวจจับไฟเลี้ยงต่ำหรือหายไปอย่างอัตโนมัติ และสามารถรักษาข้อมูลเวลาได้แม้ไม่มีไฟเลี้ยงไอซี

2.2.2 รายละเอียดขาต่อใช้งานของ DS 1307

ในรูปที่ 2.2 แสดงการจัดขาของ DS1307 แต่ละขามีหน้าที่และการใช้งานดังนี้

- $V_{cc}$, GND (ขา 8,4) ต่อกับไฟเลี้ยง +5V
- $V_{BAT}$ (ขา 3) ใช้ต่อกับแบตเตอรี่ 3 V เพื่อรักษาการทำงานของวงจรสร้างฐานเวลาของ DS1307 ให้คงอยู่ต่อไป แม้ว่าไม่มีไฟเลี้ยงจ่ายให้แก่ DS1307 ชนิดของแบตเตอรี่ที่เหมาะสม คือ แบตเตอรี่แบบลิเธียม ซึ่งมีความจุ 40 mAhr หรือมากกว่า จะสามารถรักษาข้อมูลได้นาน 10 ปีที่อุณหภูมิ 25 องศาเซลเซียส
- SDA, SCL (ขา 5 และ 6) เป็นขาสำหรับเชื่อมต่อกับไมโครคอนโทรลเลอร์ บนระบบบัส $I^2C$
- SQW/OUT (ขา 7) ที่ขานี้จะมีสัญญาณรูปสี่เหลี่ยมส่งออกมา โดยสามารถเลือกความถี่ได้ 1 Hz, 4.096kHz, 8.192kHz , 32kHz ในการใช้งานต้องมีการต่อตัวต้านทาน 1k พูลอัปที่ขานี้ด้วย
- X1, X2 (ขา 1และ2) ใช่ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอล เข้ากับขาทั้งสองนี้และที่แต่ละขาต้องต่อตัวเก็บประจุค่าต่ำๆ ประมาณ 15 pF คร่อมกับขากราวด์ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.2 แสดงการจัดขาของไอซีสร้างฐานเวลาจริง (DS1307)

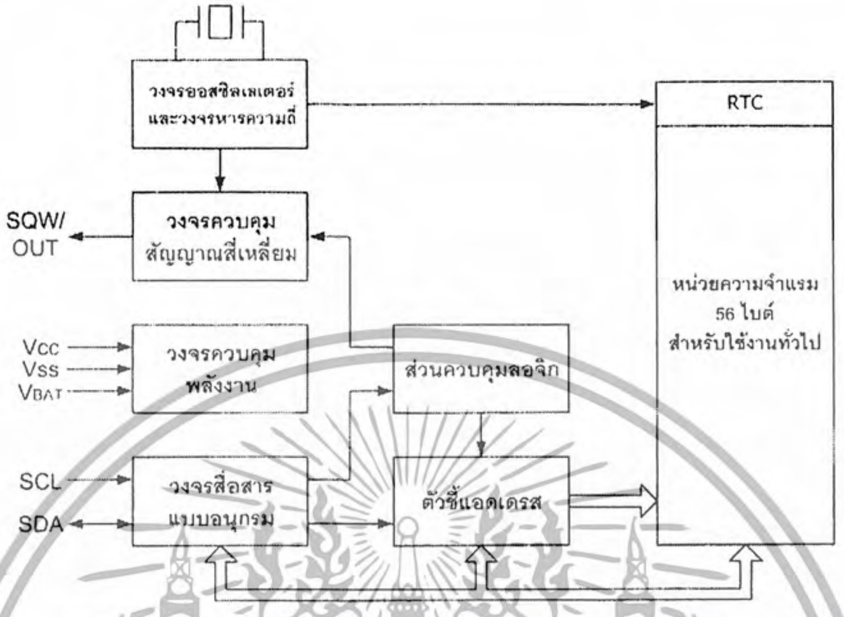
**2.2.3 การทำงานของ DS1307**

ไอซี DS1307 จัดการเชื่อมต่อระบบบัส $I^2C$ โดยจะทำงานเป็นอุปกรณ์สเลฟเสมอ ดังนั้นการติดต่อเพื่อการใช้งานจึงต้องกำหนดรูปแบบตามที่กำหนดไว้ในการติดต่อแบบ $I^2C$ ในรูปที่แสดงส่วนประกอบหลักที่สำคัญและไดอะแกรมการทำงานของ DS1307 วงจรออสซิลเลเตอร์ถือเป็นหัวใจหลักของไอซี เนื่องจากเป็นจุดเริ่มต้นของการสร้างเวลาจริง ในขณะที่ DS1307 ทำงานที่ขา SQW/OUT จะมีสัญญาณพัลส์สี่เหลี่ยมส่งออกมาตลอดเวลาในกรณีที่มีการ อีนาเบิลวงจรกำเนิดสัญญาณพัลส์ที่รีจิสเตอร์ควบคุม ค่าความถี่ของสัญญาณนี้สามารถเลือกได้ 4 ค่าคือ 1Hz, 4.096kHz, 8.192kHz และ 32 kHz พร้อมกันนั้นก็จะมีการเก็บค่าของเวลาในหน่วยความจำนอนโวลาไทล์แรม ซึ่งมีขนาดความจำรวม 64 ไบต์ แต่จัดสรรให้ใช้เก็บข้อมูลเวลา 8 ไบต์ และเป็นหน่วยความจำสำหรับเก็บข้อมูลทั่วไปสำหรับผู้ใช้งานอีก 56 ไบต์

วงจรควบคุมพลังงานไฟฟ้าจะคอยตรวจสอบสภาวะของไฟเลี้ยงไอซี หากไฟเลี้ยงต่ำกว่า 1.25 x $V_{BAT}$ ก็จะควบคุมให้ DS1307 หยุดการทำงานรีเซตค่าตัวนับแอดเดรสภายในทำให้ไม่สามารถติดต่อกับ DS1307 ได้ ดังนั้นการใช้งาน DS1307 ต้องระมัดระวังอย่าให้ไฟเลี้ยงต่ำกว่า 1.25 x $V_{BAT}$ หรือประมาณ 3.75 โวลท์ ในกรณีที่ใช้ $V_{BAT}$ เท่ากับ 3 โวลท์ ถ้าหากไฟเลี้ยงมีค่าต่ำกว่า $V_{BAT}$ ไอซี DS1307 จะเข้าสู่โหมดสำรองข้อมูลกระแสต่ำทันที จะไม่มีการส่งสัญญาณพัลส์ออกมาที่ขา SQW/OUT แต่วงจรสร้างฐานเวลายังคงทำงานเพื่อให้ค่าของข้อมูลเวลาเดินไปอย่างไม่ผิดพลาด เมื่อมีไฟเลี้ยงปรากฏขึ้นอีกครั้ง DS1307 ก็จะสามารถให้ค่าของเวลาที่เป็นจริงแก่ผู้ใช้งานได้ต่อไป

วงจรสื่อสารอนุกรมภายใน DS1307 ได้รับการกำหนดให้ทำงานตามรูปแบบของบัสเป็นช่องทางการสื่อสารระหว่าง DS1307 กับอุปกรณ์มาสเตอร์ ผู้ใช้งานสามารถเข้าถึงหน่วยความจำที่ใช้เก็บค่าเวลาและหน่วยความจำใช้งานทั่วไปได้โดยการเขียนข้อมูลตามรูปแบบที่กำหนดในระบบ $I^2C$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.3 โครงสร้างภายในของไอซีสร้างฐานเวลาจริง (Real time clock DS1307)

### 2.2.4 การจัดสรรหน่วยความจำใน DS1307

ในรูปที่ 2.4 แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน DS1307 พื้นที่ 7 ไบต์ตั้งแต่แอดเดรส 00H-06H เป็นพื้นที่ของรีจิสเตอร์ค่าเวลาที่ใช้เก็บข้อมูลเกี่ยวกับเวลา ไบต์ต่อมาที่แอดเดรส 07H เป็นพื้นที่ของรีจิสเตอร์ควบคุมการทำงานของ DS1307 ในรูปที่ 2.4 แสดงรายละเอียดของรีจิสเตอร์ค่าเวลาและรีจิสเตอร์ควบคุม DS1307

ด้วยการจัดสรรพื้นที่แบบนี้ทำให้ผู้ใช้งานสามารถเรียกข้อมูลออกมาได้ตามเราต้องการโดยไม่จำเป็นต้องอ่านออกมาทั้งหมดก็ได้ ค่าของเวลาทั้งหมดจะอยู่ในรูปของเลขฐานสิบ สำหรับการแสดงเวลาในรูปของชั่วโมง สามารถเลือกได้ว่าต้องการแบบ 12 ชั่วโมง หรือ 24 ชั่วโมง โดยกำหนดบิตที่ 6 ของแอดเดรส 02H และเมื่อเลือกแบบ 12 ชั่งโมง ที่ บิตที่ 5 ในแอดเดรสเดียวกันจะใช้ในการแสดงค่า AM/PM โดยถ้าบิตนี้เป็น “1” หมายถึง ค่าชั่วโมงในขณะนี้เป็นช่วงหลังเที่ยงวัน ในกรณีที่เป็นแบบ 24 ชั่วโมง บิตนี้จะใช้ในการแสดงค่า 2 ของหลักสิบในหน่วยชั่วโมง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | บิต 7 | | | | | | บิต 0 | | |
|---|---|---|---|---|---|---|---|---|---|
| 00H | CH | วินาที หลักสิบ | | | วินาที | | | | 00-59 |
| 01H | X | นาที หลักสิบ | | | นาที | | | | 00-59 |
| 02H | X | 12, 24 | A/P | 10 HR | ชั่วโมง | | | | 01-12, 00-23 |
| 03H | X | X | X | X | X | วัน | | | 1-7 |
| 04H | X | X | วันที่ หลักสิบ | | วันที่ | | | | 01-31 |
| 05H | X | | เดือน หลักสิบ | | เดือน | | | | 01-12 |
| 06H | ปี หลักสิบ | | | | ปี | | | | 00-99 |
| 07H | OUT | X | X | SQWE | X | X | RS1 | RS2 | |

รูปที่ 2.4 แสดงการจัดสรรพื้นที่ของหน่วยความจำภายใน ไอซีสร้างฐานเวลาจริง (DS1307)

**2.2.5 รีจิสเตอร์ควบคุม**

มีแอดเดรสที่ 07H มีรายละเอียดของแต่ละบิตดังนี้

- OUT (Outputcontrol) : ใช้ในการควบคุมระบบลอจิกที่ขา SWQ/OUT ในกรณีที่ดิสเอเบิลการกำเนิดสัญญาณสี่เหลี่ยม โดยถ้าบิตนี้เป็น " 1 " ที่ขา SWQ/OUT ก็จะเป็น " 1 " ถ้าบิตนี้เป็น " 0"ที่ขา SWQ/OUT ก็จะเป็น "0"
- SQWE (Square Wave Enable) : ใช้ในการอินาเบิลวงจรกำเนิดสัญญาณสี่เหลี่ยมที่ขา SWQ/OUT ถ้าต้องการให้มีสัญญาณให้มีสัญญาณสี่เหลี่ยมออกมาให้กำหนดบิตนี้เป็น "1"
- RS0 , RS1 (Rate Select) : ใช้ในการเลือกความถี่ของสัญญาณสี่เหลี่ยม ที่ออกจากของ SWQ/OUT มีรายละเอียดดังตารางที่ 2.3

| RS1 | RS0 | ค่าความถี่ของสัญญาณสี่เหลี่ยม |
|---|---|---|
| 0 | 0 | 1Hz |
| 0 | 1 | 4.096kHz |
| 1 | 0 | 8.192kHz |
| 1 | 1 | 32.768khz |

ตารางที่ 2.3 แสดงการเลือกความถี่ของสัญญาณสี่เหลี่ยมที่ออกจาก SWQ

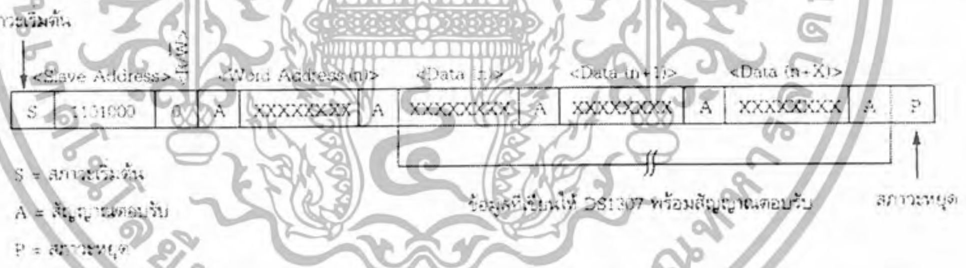
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.6 โหมดการทำงานของ DS1307**

มีด้วยกัน 2 โหมดคือโหมดเขียนและอ่านข้อมูล ในการใช้งาน DS1307 ตามปกติ จะใช้งานเฉพาะโหมดอ่านข้อมูลเท่านั้น เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1307 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่และต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1307 จำเป็นอย่างยิ่งจะต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นการอ่านข้อมูลต่อไป

**2.2.7 โหมดการเขียนข้อมูล**

มีรูปแบบดังรูปที่ 2.5 เริ่มต้นเมื่อไมโครคอนโทรลเลอร์ ทำการกำหนดสภาวะเริ่มต้น (START:S) จากนั้นส่งข้อมูลกำหนดแอดเดรส 1101000 ตามด้วยข้อมูลเลือกการเขียนนั่นคือค่า 0 จากนั้นจะรอการตอบรับจาก DS1307 ขั้นตอนต่อมาคือ ส่งข้อมูลเพื่อเลือกแอดเดรสที่ต้องการเขียนจากนั้นจะรอการตอบรับจาก DS1307 เมื่อมีการต้อนรับกลับมาเรียบร้อย ก็เริ่มทยอยเขียนข้อมูลลงไปครั้งละแอดเดรส หลังจากเขียนข้อมูลในแต่ละแอดเดรส จะต้องหยุดการตอบรับจาก DS1307 ทุกครั้งจึงสามารถเขียนข้อมูลต่อไปได้ เมื่อเขียนเรียบร้อยแล้วก็ให้ส่งสภาวะหยุด (STOP:P) เป็นอันสิ้นสุดกระบวนการเขียนข้อมูล



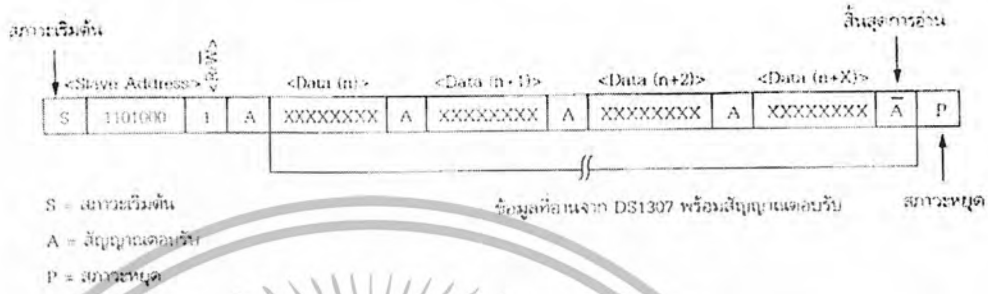

รูปที่ 2.5 แสดงรูปของข้อมูลสำหรับติดต่อกับ DS1307 ในโหมดการเขียนข้อมูล

**2.2.8 โหมดการอ่านข้อมูล**

มีรูปแบบดังรูปที่ · 2.6 เริ่มต้นการทำงานเหมือนกับโหมดการเขียนข้อมูลคือไมโครคอนโทรลเลอร์กำหนดสภาวะเริ่มต้นแล้วส่งข้อมูลกำหนดแอดเดรสตามด้วยข้อมูลเลือกการอ่านซึ่งเท่ากับ "1" จากนั้นรอคอยการตอบรับจาก DS1307 เมื่อตอบรับเรียบร้อย DS1307 จะทยอยส่งข้อมูลออกมาให้ไมโครคอนโทรลเลอร์คราวละ 1 แอดเดรสหรือ 1 ไบต์ โดยแอดเดรสที่เลือกอ่านข้อมูลจะต้องมีการกำหนดล่วงหน้าด้วยโหมดการเขียนข้อมูล วิธีการง่ายๆ คือ เข้าสู่โหมดการเขียนข้อมูล ให้ทำการสร้างสภาวะเริ่มต้นและส่งข้อมูลกำหนดแอดเดรสใหม่อีกครั้ง ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเลือกโหมดการอ่านข้อมูล ข้อมูลที่ออกมาจาก DS1307 ก็จะเป็นข้อมูลจากแอดเดรสที่กำหนดไว้ก่อนหน้านี้




รูปที่2.6 แสดงรูปแบบของข้อมูลสำหรับการติดต่อกับ DS1307 ในโหมดการอ่านข้อมูล

## 2.3 จอแสดงผลโมดูล LCD

### 2.3.1 รายละเอียดเกี่ยวกับจอแสดงผล

ในโมดูล LCD ประกอบไปด้วย 3 ส่วนหลักๆ คือ

- ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็น โดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนจอ LCD

- ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของโมดูล LCD เช่น ลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมชนิดนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม LCD แบบอักขระ ส่วน HD61830 ใช้ควบคุม LCD แบบกราฟฟิก

- ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่เป็นตัวขับนี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น

### 2.3.2 โครงสร้างภายในของตัวควบคุม โมดูล LCD

ในการใช้งานโมดูลของ LCD จำเป็นต้องทำความเข้าใจเกี่ยวกับโครงสร้างและคำสั่งที่ใช้ในการควบคุมให้ดีเสียก่อน

-บัฟเฟอร์อินพุตเอาท์พุตเป็นส่วนที่ใช้ในการติดต่อรับส่งขอมูลกับอุปกรณ์ภายนอกเพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

- รีจิสเตอร์คำสั่ง (Imstruction register :IR) เป็นรีจิสเตอร์ใช้รับส่งข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจิสเตอร์ข้อมูล (Data register : DR) เป็นรีจิสเตอร์ใช้รับส่งข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

- แรมเก็บข้อมูลแสดงผล (Display data RAM : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตารางของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและแรมเก็บตัวอักษร เพื่อนำไปแสดงทีตัวแสดงผล

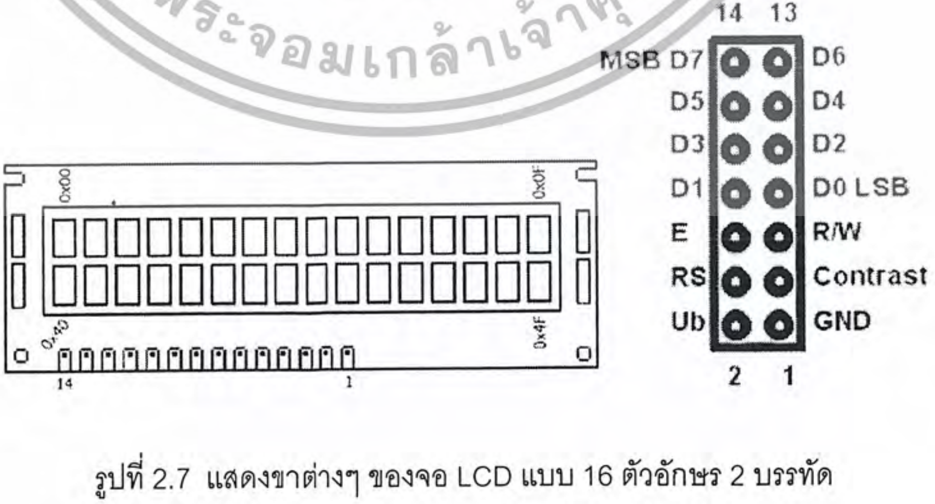
- รอมเก็บตัวอักษร (Character generator ROM : CGROM ) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านข้อมูลแล้วไปแสดงที่ตัวแสดงผลได้ ซึ่งมีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

- แรมเก็บตัวอักษร (Character generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGRAM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGRAM คือเขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

- แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

**2.3.3 โมดูล LCD ขนาด 16 ตัวอักษร 2 บรรทัด ( LCD 16 x 2 )**

โมดูล LCD ขนาด 16 x 2 มีขาต่อใช้งานทั้งสิ้น 14 ขา รายละเอียดการทำงานของแต่ละขามีดังนี้




รูปที่ 2.7 แสดงขาต่างๆ ของจอ LCD แบบ 16 ตัวอักษร 2 บรรทัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GND(ขา 1) ใช้สำหรับต่อกราวด์

Ub (ขา 2) ต่อไฟเลี้ยง +5 โวลท์

Contrast (ขา 3) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS(ขา 4) เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น " 0" ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น " 1 " ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

R/W (ขา 5) เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น "0" เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น " 1 " จะเป็นการอ่านข้อมูล

E (ขา 6) เป็นขาอีนาเบิล LCD ให้ทำงาน

D0-D7 (ขา 7 – 14) เป็นขาที่ใช้ทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต

**2.3.4 คำสั่งควบคุมโมดูล** LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ R/W เป็น "0" แล้วเขียนคำสั่งควบคุม คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมีดังนี้

-คำสั่งเคลียร์ตัวแสดงผล (Clear display)

มีคำสั่งข้อมูลเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างเข้าไปใน DDRAM เมื่อตัวควบคุมเอ็กซ์คิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น "0" เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D (ซึ่งจะกล่าวทีหลัง) ให้เป็น " 1 "

- คำสั่ง Return home

ต้องกำหนดให้บิต 1 ของข้อมลเป็น "1" เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

- คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น "1" เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น " 0 " เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น "0 " แอดเดรสจะลดลง

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H – 07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึงกำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือและแอดเดรสของ DDRAM เพิ่มขึ้น

- คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

| บิต7 | บิต6 | บิต5 | บิต4 | บิต3 | บิต2 | บิต1 | บิต0 |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 | D | C | B |

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น " 1 " จะเป็นการเปิดจอแสดงผล ถ้าเป็น " 0 " จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น " 1 " ถ้ากำหนดเป็น " 0 " จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกะพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น " 1 " เคอร์เซอร์จะกะพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H – 0FH ( 8 รูปแบบคำสั่ง ) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กะพริบ

- คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

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

- คำสั่งกำหนดฟังก์ชั่นการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น " 0 " จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น " 1 " จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น " 0 " จะแสดงผล 1 บรรทัด ถ้าเป็น " 1 " จะแสดง 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น " 1 "

บิต F ใช้เลือกความละเอียดของตัวอักษรให้การแสดงผล ถ้าบิตนี้เป็น "0" จะเป็นการแสดงผลแบบ 5 x 7 จุด และถ้าเป็น " 1 " จะแสดงผลเป็นแบบ 5 x 10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนด โมดูล LCD ให้ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5 x 7 จุด

- คำสั่งเลือกแอดเดรสของ CGRAM

เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น " 0 " บิต 6 เป็น 1 ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H – 3FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


- คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูลนั้น ที่บิต 7 ต้องเป็น " 1 " และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N เป็น " 0 " แอดเดรสของ DDRAM จะอยู่ระหว่าง80H–0CFH และถ้าบิต N เป็น"1"แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-0C7H

- คำสั่งอ่านแฟลก BUSY และแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

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

แอดเดรสไบต์สูง แอดเดรสไบต์ต่ำ

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY ( ในตารางจะกำหนดเป็น BF ) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น " 0 " แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น " 1 " แสดงว่าขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/W เป็น " 1 " ด้วย แต่สัญญาณที่ RS ยังต้องเป็น " 0 " อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่งนอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0-6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.3.5 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมให้โมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องป้อนคำสั่ง แล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูลที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้น คือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก " 0 " หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่ง ในทางตรงกันข้าม หากขา RS ได้รับลอจิก " 1 " ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ใน

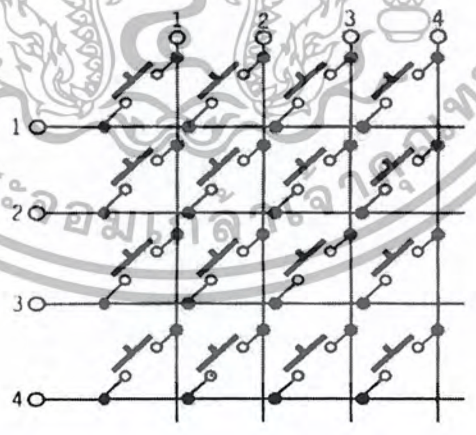
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

55679

การแสดงผลเมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้อนกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น " 1 " เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช้คำสั่ง ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/W เป็น " 1 " ข้อมูลในขนาด 8 บิต ( หรือ 4บิต ) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอกจิก " 1 " ให้ขา RS แล้ว ต้องกำหนดให้ขา R/W เป็น " 0 " ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

### 2.4 สวิทซ์เมตริกซ์หรือคีย์แพด

การอ่านหรือรับค่าการกดสวิทซ์ซึ่งก็เป็นอีกลักษณะหนึ่งของการใช้งานของตัวไมโครโทรลเลอร์ต้องสามารถรองรับและเชื่อมต่อใช้งานร่วมกับวงจรของสวิทซ์มี 2 ลักษณะใหญ่ๆ คือ ต่อเข้ากับไฟเลี้ยงหรือกราวด์โดยตรง เมื่อสวิทซ์ตัวใดต่อวงจรสามารถอ่านค่าโดยตรงวงจรในลักษณะนี้ไม่มีความซับซ้อน สามารถอ่านค่าของสวิทซ์ได้ง่ายและรวดเร็ว มีข้อเสียคือ ถ้าจำนวนของสวิทซ์มีมากๆ จำนวนของสายข้อมูลก็จะมีมากตาม ทำให้ระบบหรือวงจรโดยรวมมีขนาดใหญ่และสิ้นเปลือง

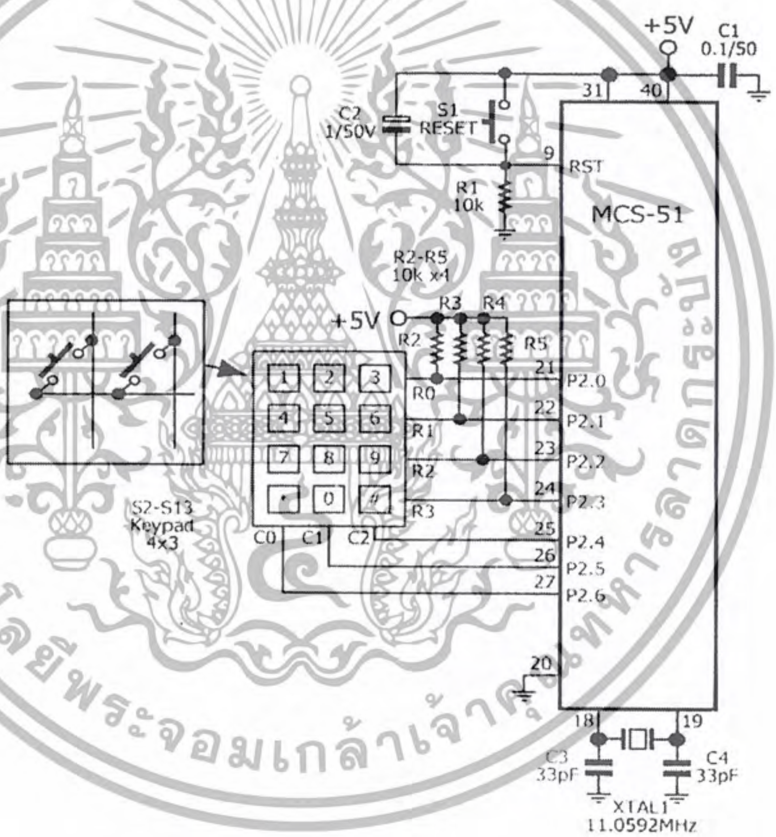



รูปที่ 2.8 วงจรสวิทซ์เมตริกซ์หรือคีย์แพด

การต่อวงจรของสวิทซ์อีกลักษณะหนึ่งคือ การต่อวงจรแบบเมตริกซ์ (Matrix switch)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังในรูปที่ 2.9 สวิทช์จะถูกต่อกันในแนวแกนตั้งและแกนนอน จะเรียกแนวตั้งว่า หลักหรือคอลัมน์ (Column) ในขณะที่แนวนอนจะเรียกว่า แถวหรือโรว์ (Row) ดังนั้นค่าของสวิทช์ประกอบด้วยตำแหน่งในแนวหลักและแถว กระบวนการที่จะทำให้ได้มาซึ่งค่าของสวิทช์มีขั้นตอนซับซ้อนพอสมควรแต่ วงจรของสวิทช์แบบนี้มีข้อดีคือสามารถรองรับการเพิ่มเติมของสวิทช์ได้อย่างสะดวก เพียงเพิ่มจำนวนสวิทช์และแก้ไขซอฟต์แวร์อีกเล็กน้อยเท่านั้น ทำให้วงจรสวิทช์แบบเมตริกซ์เป็นที่นิยมใช้มากในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติที่จำนวนสวิทช์มากกว่า 8 ตัว ในการใช้งานทั่วไปจะเรียกสวิทช์แบบเมตริกซ์นี้ว่า คีย์แพด (Keypad)




รูปที่ 2.9 วงจรเชื่อมต่อคีย์แพดเข้ากับไมโครคอนโทรลเลอร์ MSC-51

การเชื่อมต่อคีย์แพดเข้าไปกับไมโครโทรลเลอร์ MSC-51

ตัวอย่างวงจรในรูปที่ 2.9 จะใช้พอร์ต 2 ไมโครคอนโทรลเลอร์เชื่อมต่อเข้ากับคีย์แพดทั้ง 7 เส้นคือ สายคอลัมน์ 3 เส้น คือ C0-C2 และสายทางโรว์หรือแถวอีก 4 เส้นคือ R0-R3 โดน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฉพาะที่ขาพอร์ต P2.0-P2.3 จะต้องต่อตัวต้านทานพูลอัปไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ โดยที่ไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล " 0 " ไปยัง P2.6,P2.5 และ P2.4 ตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายคอลัมน์ของคีย์แพดไมโครคอนโทรลเลอร์จะทำการอ่านค่าที่ P2.0-P2.3 จะไม่เป็น 111 อีกต่อไปเป็นการแจ้งให้ทราบว่ามีการกดคีย์แพดขึ้น จากนั้นไมโครคอนโทรลเลอร์ก็จะทำหน้าที่ค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งสิ่งที่จะได้มาอย่างแรกคือ ค่าตำแหน่งของคีย์นั้น จากนั้นก็จะนำค่าตำแหน่งนั้นไปเปิดตารางข้อมูล เพื่อจะได้หมายเลขของคีย์ที่กดอย่างแท้จริง ยกตัวอย่างจากในรูปที่ 2.9 หากคีย์ 0 ถูกกดไมโครคอนโทรลเลอร์จะได้ตำแหน่งของคีย์ 0 เป็น 0BH จากนั้นนำค่า 0BH นี้ไปเปิดตารางก็จะได้ข้อมูลเป็น 0 หมายถึงคีย์ 0 ซึ่งกระบวนการหลังจากนี้จะเป็นหน้าที่ของซอฟแวร์

## 2.5 ตัวส่งแสงอินฟาเรดและตัวรับแสงอินฟาเรด

จะทำหน้าที่ ตรวจจับปริมาณของอาหารปลาโดยอาศัย LED อินฟาเรดเป็นตัวรับตัวส่งข้อมูลและส่งไปยังหน่วยประมวลผล ทฤษฎีและความรู้ทั่วไปเกี่ยวกับตัวรับและตัวส่งอินฟาเรด

### 2.5.1 รังสีอินฟาเรด

มีลักษณะเช่นเดียวกับคลื่นวิทยุ และรังสีเอ็กซ์คือมีการแผ่พลังงานออกมาในรูปของคลื่นแม่เหล็กไฟฟ้า คุณสมบัติต่างๆ ของแสงอินฟาเรดมีดังนี้

1. เป็นแสงที่ไม่สามารถมองเห็นด้วยตาเปล่า
2. เดินทางในอากาศด้วยความเร็ว $3*10^8$ เมตรต่อวินาที
3. มีความยาวคลื่นที่ใช้ประมาณ 720 – 1200 นาโนเมตร

แสงอินฟาเรดนี้ยังมีแหล่งกำเนิดของแสงอินฟาเรดนี้อยู่หลายแหล่งด้วยกันเช่น

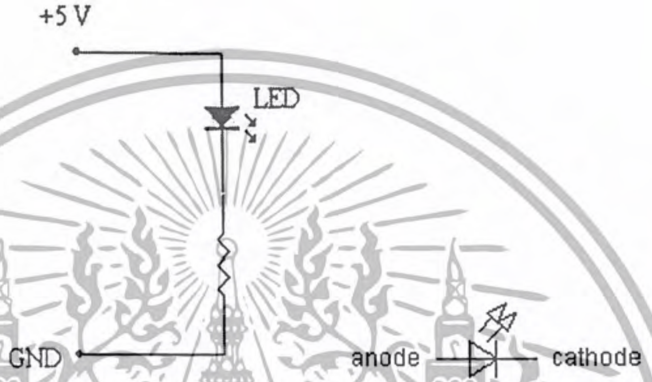
1. แสงอินฟาเรดที่เกิดจากปฏิกิริยาเคมี
2. แสงอินฟาเรดที่เกิดจากการเคลื่อนที่ของอิเล็กตรอน
3. แสงอินฟาเรดเกิดขึ้นในทางไฟฟ้า

### 2.5.2 ตัวส่งแสงอินฟาเรด

สำหรับคุณสมบัติของแสงอินฟาเรดที่เรานำมาใช้ มีแหล่งกำเนิดแสงอินฟาเรดมาจากแหล่งกำเนิดทางไฟฟ้าคือ เราใช้ LED อินฟาเรด รุ่น TLL 38 ของบริษัท TEXAS INSTRUMENTS ซึ่งมีคุณสมบัติให้ลำแสงอินฟาเรดออกมา เมื่อตัวมันนำกระแสและมีความยาวคลื่นประมาณ 940 นาโนเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการของ LED อินฟาเรดก็คือ เมื่อให้แรงดันไบแอสตรงกับมัน ด้วยแรงดันไฟตรงต่ำๆ กระแสไบแอสตรงคือ อิเล็กตรอนจะไหลผ่านรอยต่อจาก N ไปยังส่วน P และโฮลไปยังส่วน N และจะรวมกับพาหนะข้างน้อย เป็นผลทำให้เกิดการีคอมไบเนชั่น และปล่อยพลังงานออกมาเป็นแสงและความร้อน




รูปที่ 2.10 การไบแอสตรงของไดโอด

2.5.3 **ตัวรับแสงอินฟาเรด**

ตัวรับแสงหมายถึง อุปกรณ์ที่สามารถเปลี่ยนพลังงานให้แปรค่ากับค่าของพลังงานทางไฟฟ้าได้โดยตัวอุปกรณ์จะต้องประกอบด้วยสารกึ่งตัวนำซึ่งอาจจะนำมาต่อเชื่อมให้เกิดรอยต่อหรือเป็นเนื้อสารกึ่งตัวนำอย่างเดียวก็ได้ โฟโต้ไดโอด เป็นอุปกรณ์ไวแสงชนิดหนึ่งที่ประกอบด้วยรอยต่อ PN การใช้งานตัวโฟโต้ไดโอดจะให้กระแสไหลผ่านตัวมันที่ขึ้นอยู่กับแสง เมื่อโฟโต้ไดโอดชนิดซิลิก่อนถูกไบแอสด้วยแรงดันค่าหนึ่ง และมีแสงส่องไปที่บริเวณรอยต่อถ้าแสงที่ส่องมีความยาวคลื่นที่พอเหมาะจะมีกระแสไหลในวงจร โดยเป็นสัดส่วนกับความเข้มของแสงที่ส่องผ่านอุปกรณ์นั้นลักษณะทั่วไปขณะไบแอสตรงจะยังคงเหมือนกับไดโอดธรรมดาคือให้กระแสผ่านตัวมันเองได้การทำงานของโฟโต้ไดโอดพอจะกล่าวได้อย่างสังเขปดังนี้ เมื่อแสงตกกระทบบริเวณรอยต่อจำทำให้เกิดการไหลข้ามรอยต่อเป็นกระแส เนื่องจากแสงได้โดยปกติค่ากระแสเนื่องจากแสงมีค่าค่อนข้างต่ำ ดังนั้นการใช้งานของโฟโต้ไดโอด มีสัมฤทธิ์ของการเปลี่ยนแปลงตามอุณหภูมิต่ำ และผลตอบสนองต่อแสงไว นอกจากนี้ยังสามารถควบคุมผลตอบสนองต่อความถี่ของแสงและความเร็วได้โดยรูปร่างลักษณะและการโด๊ปของรอยต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.6 มอเตอร์ไฟฟ้ากระแสตรง**

มอเตอร์ไฟฟ้ากระแสตรง ( DC Motor ) เป็นทรานดิวเซอร์ (Transducecer) แรงบิดซึ่งมีการออกแบบให้มีคุณสมบัติพิเศษ คือแรงบิดของเพลา (Shaft) ของมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์เมเจอร์ (Armature Current) แรงบิดของเพลามอเตอร์จะเป็นผลระหว่างแรงบิดของเพลาและกระแสเท่ากับ

$$T = KI\Phi$$

เมื่อ

| | | |
|---|---|---|
| $T$ | คือ | แรงบิดของเพลา (N.m) |
| $\Phi$ | คือ | เส้นแรงแม่เหล็ก (wb) |
| $I$ | คือ | กระแส (A) |
| $K$ | คือ | ค่าคงที่ |

ดังนั้นแรงบิดของเพลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแรงแม่เหล็กและกระแส

เมื่อ

ขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กจะทำให้เกิดโวลท์เตทตกคร่อมตัวมันเอง และโวลท์เตทนี้จะเป็นสัดส่วนกับความเร็วของเพลามอเตอร์ และต้านการไหลของกระแส ความสัมพันธ์ระหว่างแรงดันไฟฟ้าต้านกลับ และความเร็วของเพลามอเตอร์คือ

$$E = K\Phi w$$

เมื่อ

| | | |
|---|---|---|
| $E$ | คือ | แรงดันไฟฟ้าต้านกลับ emf (V) |
| $w$ | คือ | ความเร็วของมอเตอร์ (rad/s) |

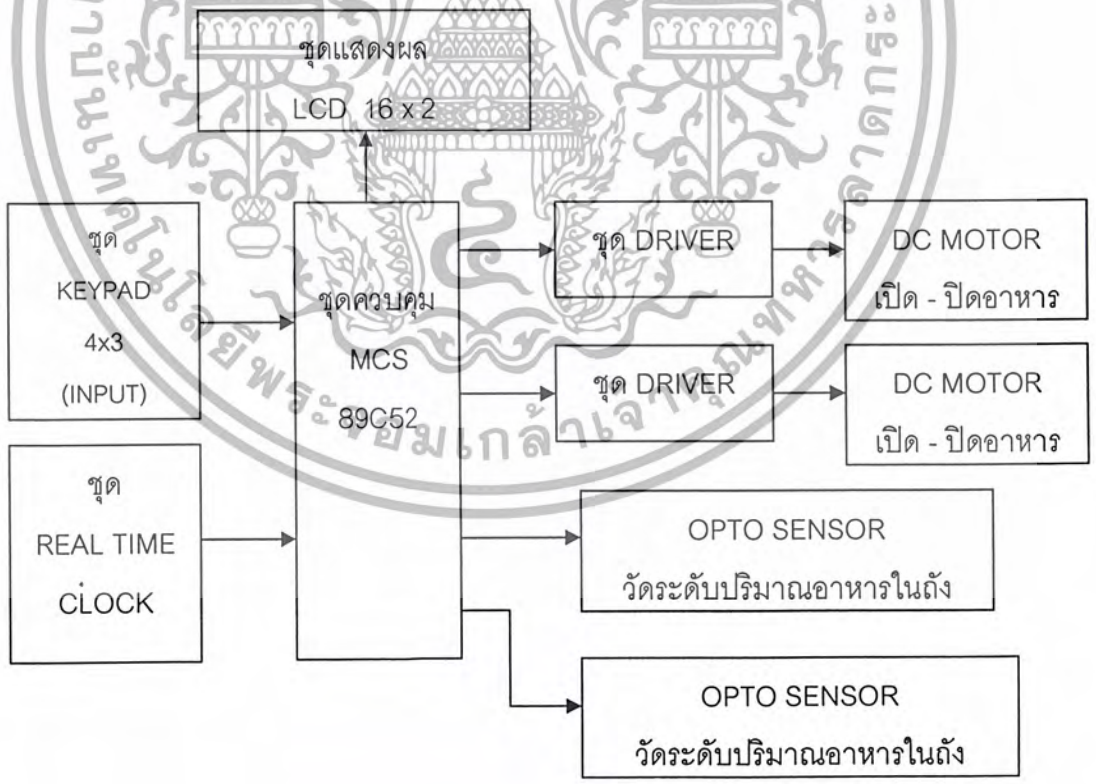
**สมการทั้งสองแสดงถึงหลักการทำงานพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรง**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# ขั้นตอนและวิธีดำเนินการ

## 3.1 การทำงานเครื่องให้อาหารปลา

เครื่องให้อาหารปลาใช้ ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เป็นตัวควบคุม ต่อเชื่อมกับอุปกรณ์รอบข้าง โดยใช้ LCD 16 ขนาด 16 ตัวอักษร 2 บรรทัดเป็นอุปกรณ์แสดงผล ใช้ Keypad ขนาด 4x3 เป็นอินพุต เพื่อป้อนข้อมูลให้ ไมโครคอนโทรลเลอร์ สำหรับ ไอซีสร้างฐานเวลาจริง เลือกใช้ เบอร์ DS1307 เพื่อให้ข้อมูลเวลาแก่ไมโครคอนโทรลเลอร์ ชุดขับมอเตอร์กระแสตรงมี 2 ชุด สำหรับ เปิด- ปิด ลิ้นจ่ายอาหาร และชุดโฟโต้เซนเซอร์ 2 ชุด เพื่อวัดระดับอาหาร ดังแสดงใน รูปที่ 3.1

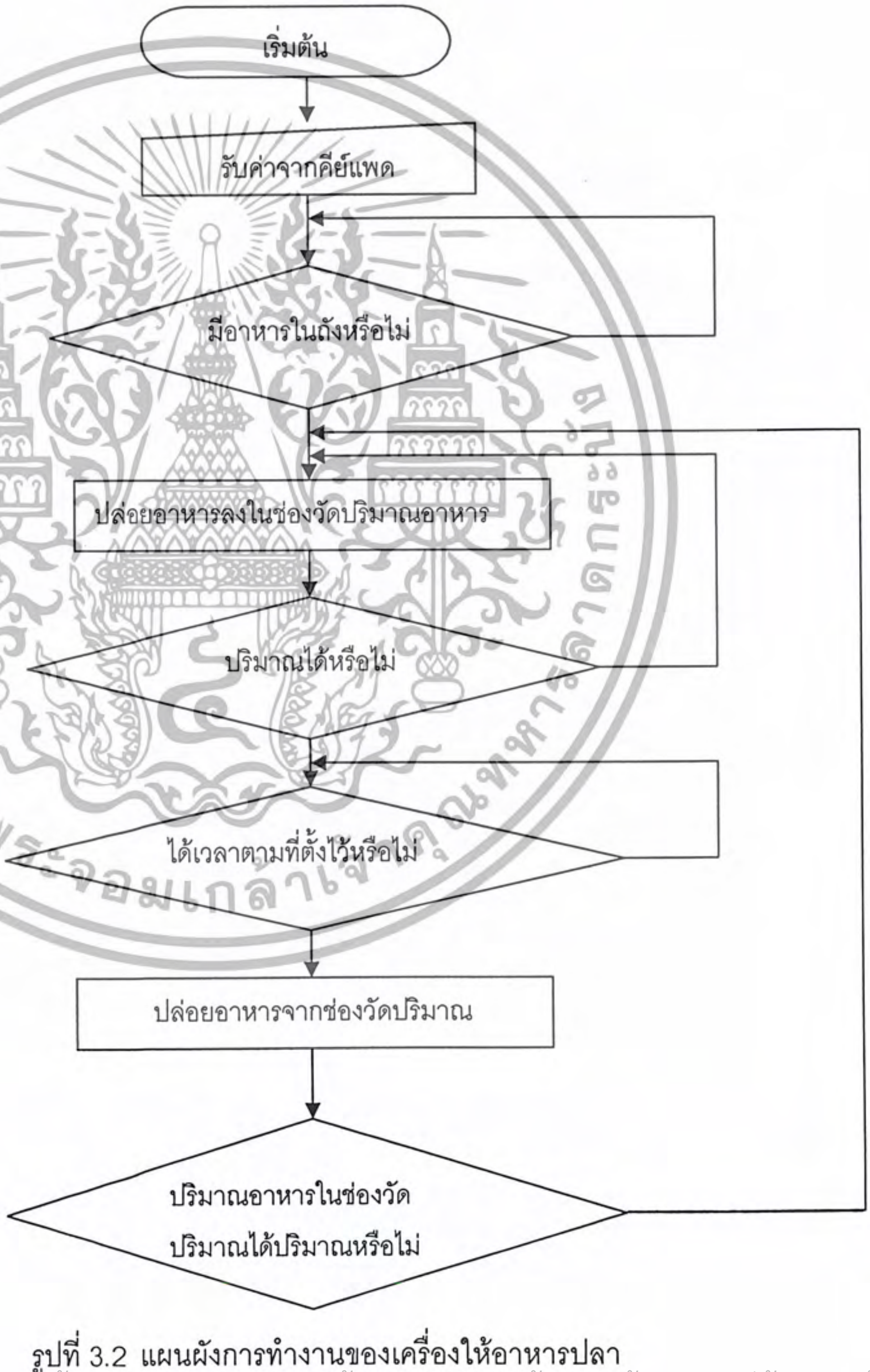



รูปที่ 3.1 บล็อกไดอะแกรมเครื่องให้อาหารปลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2 ลำดับขั้นการทำงานของเครื่องให้อาหารปลา**

การทำงานของเครื่องให้อาหารปลาต้องมีการป้อนเวลาจริงในปัจจุบันให้ระบบเก็บค่า โดยป้อนจากคีย์แพด จากนั้นเซนเซอร์จะตรวจจับว่ามีอาหารในถังหรือไม่ ถ้ามีอาหารลิ้นจะเปิดออกเพื่อจ่ายอาหารให้ช่องวัดปริมาณ ปล่อยอาหารถึงระดับเซนเซอร์ที่ติดตั้งไว้ ลิ้นที่ 1 ก็จะหยุดการจ่ายอาหาร เมือถึงเวลาที่กำหนดลิ้นที่ 2 จะจ่ายอาหารจนหมดช่องวัดปริมาณ ดังรูปที่ 3.2




รูปที่ 3.2 แผนผังการทำงานของเครื่องให้อาหารปลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 การออกแบบเครื่องให้อาหารปลา

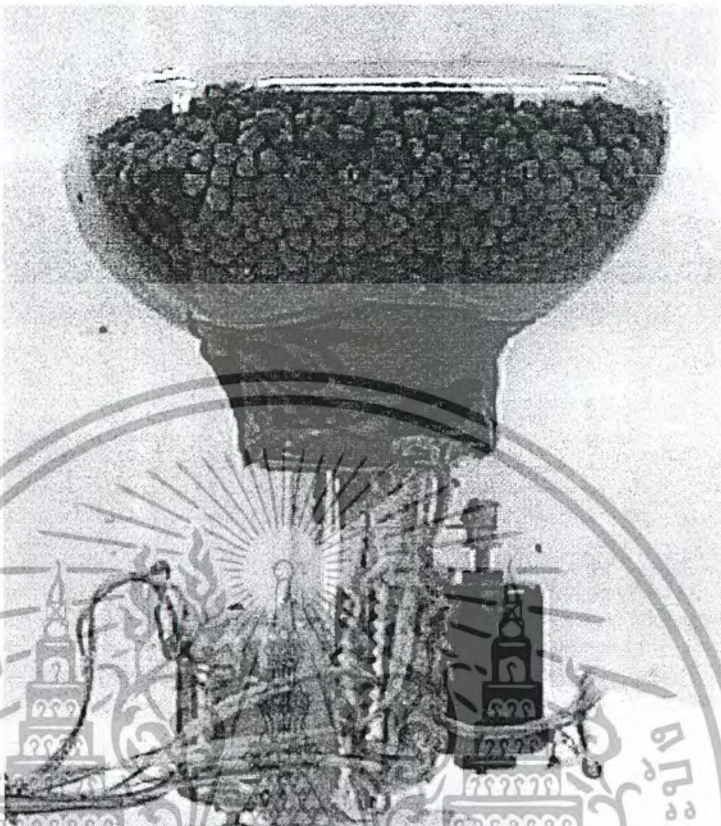
ส่วนประกอบของเครื่องให้อาหารปลาประกอบด้วย ส่วนควบคุมและส่วนกลไก ส่วนควบคุมจะใช้ MCS – 51 โดยมีการทำงานดังนี้ เริ่มต้น เซนเซอร์ที่ติดตั้งไว้ส่วนปลายของถังจะตรวจจับว่ามีอาหารในถังหรือไม่ ถ้าไม่มีอาหารปลาในถัง จอแสดงผล LCD จะแสดงว่า NO FOOD รอจนกว่าในถังจะมีอาหาร เมื่อมีอาหารปลา ลิ้นจะเปิดเพื่อปล่อยอาหารเข้าช่องวัดปริมาณอาหาร จนถึงระดับที่ตัวเซนเซอร์ที่ติดตั้งไว้ หลังจากนั้นลิ้นที่ 1 ก็จะปิด ลิ้นที่ 2 จะรอเวลาที่ตั้งไว้ และเมื่อถึงเวลาก็ค่อย ๆ เปิดปล่อยอาหารออกจนหมด จากนั้นเริ่มทำงานอีกครั้งตามที่ กำหนดทาง INPUT

เครื่องให้อาหารปลาอัตโนมัติมีการออกแบบแบ่งได้เป็น 3 ส่วนใหญ่ คือ 1. โครงสร้าง 2. วงจรและอุปกรณ์ 3. โปรแกรม

3.3.1. **ส่วนโครงสร้างมีส่วนประกอบ 3 ส่วน**

- ถังใส่อาหาร
- ลิ้น เปิด – ปิด จ่ายอาหาร
- ท่อวัดระดับปริมาณอาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 รูปของโครงสร้างหัวจ่ายอาหาร

-ถังใส่อาหาร ทำจากพลาสติกใสที่ส่วนปลายถังติดตั้ง โฟโต้เซนเซอร์ ทั้งตัวรับ และตัวส่งห่างกัน 2.5 เซนติเมตร รอบข้างหุ้มด้วยวัสดุทึบแสง เพื่อป้องกันแสงรบกวนภายนอก

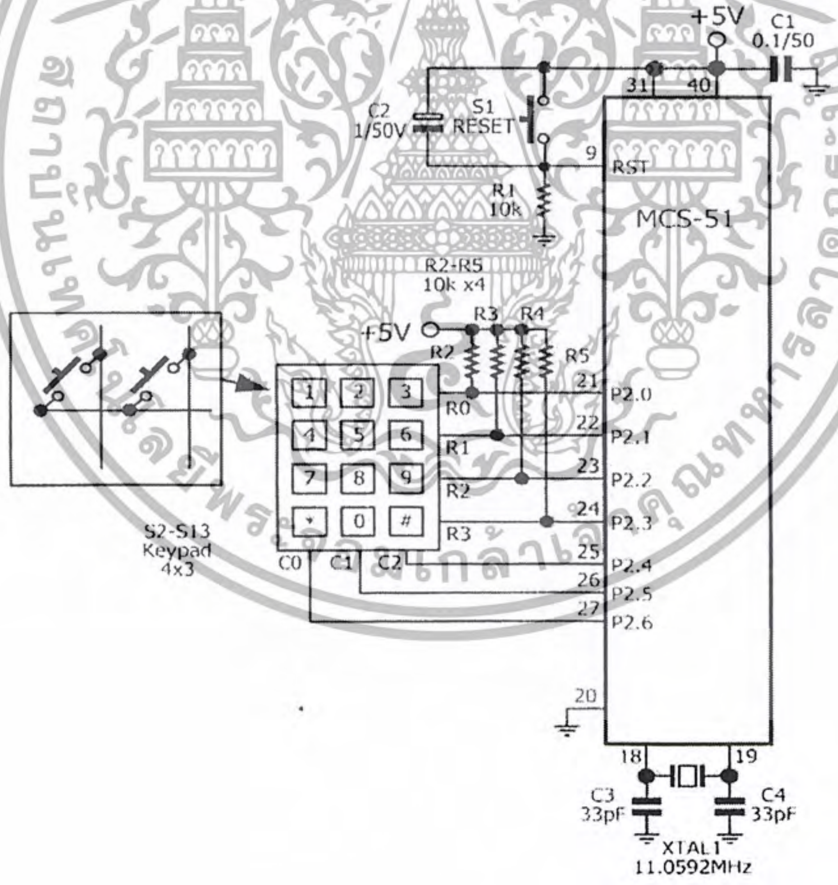
-ลิ้น เปิด – ปิด จ่ายอาหาร ทำจากโลหะแผ่นตัดเป็นวงรี ส่วนปลายด้านหนึ่งยึดกับผนังท่อส่วน ปลายอีกด้านหนึ่ง เจาะรู ร้อยเชือกเพื่อนำมาผูกติดกับเพลาของมอเตอร์

-ท่อวัดระดับ เป็นท่อใส่เสาะร่องตามแนวท่อ เพื่อติดตั้งโฟโต้เซนเซอร์ และสามารถปรับระดับโฟโต้เซนเซอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.3.2 ส่วนของวงจรและอุปกรณ์มีส่วนประกอบ 5 ส่วน**

- ชุดสวิตซ์เมทริกซ์หรือคีย์แพด(Keypad ) 4 X 3 โดยชุดสวิตซ์เมทริกซ์หรือคีย์แพด(Keypad) จะทำการเชื่อมต่อกับไมโครคอนโทรลเลอร์โดยใช้พอร์ต 2 เชื่อมต่อกับคีย์ 7 เส้น การต่อคีย์สวิตซ์แบบแมทริกริกซ์นั้นจะนำคีย์สวิตซ์สวิตซ์แต่ละตัวมาเรียงกันแบบเมทริกซ์ เราสามารถอ้างตำแหน่งของคีย์ในแนวแกน x, y ได้ การสแกนคีย์ในแบบนี้ทำได้รวดเร็วไม่ว่าคีย์ใดถูกกดก็สามารถอ่านได้ทันที โดยที่ขาพอร์ต P2.0 – P2.3 จะต่อตัวต้านทานไว้เพื่อกำหนดสภาวะเริ่มต้นที่ไม่มีการกดคีย์ ไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยัง P2.6,P2.5 และ P2.4 ตามลำดับทุกครั้งที่มีการกดคีย์ ตัวไมโครคอนโทรลเลอร์จะทำการอ่านค่าจาก P2.0 – P2.3 เข้ามาเก็บที่ตัวไมโครคอนโทรลเลอร์ ซึ่งมีค่าข้อมูล"1" ดังในรูปที่ 3.4




รูปที่ 3.4 การต่อคีย์สวิตซ์เมทริกซ์หรือคีย์แพดกับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชุดแสดงผล LCD ชุดแสดงผล LCD ที่นำมาใช้ในโครงงาน จะมีขนาด 16 ตัวอักษร 2 บรรทัด จะมีขาทั้งหมด 14 ขา ซึ่งมีรายละเอียดการทำงานดังนี้

Vss (ขา 1) ต่อกับกราวด์

$V_{DD}$ (ขา 2) ต่อไฟเลี้ยง + 5 โวลต์

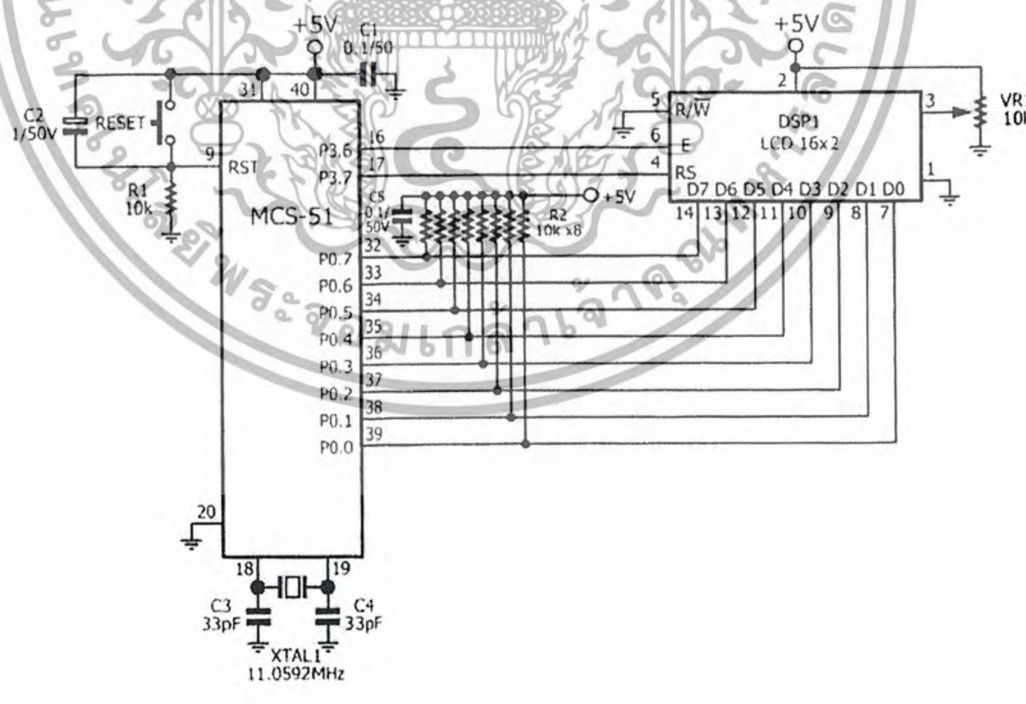
$V_{O}$ (ขา 3 ) เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS ( ขา 4 ) เป็นขาอินพุตที่ใช้แยกชนิดของข้อมูลที่ทำการประมวลผลว่าเป็นคำสั่งของรีจิสเตอร์ IR รีจิสเตอร์ DR โดยที่ค่าที่ส่งมาเป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าข้อมูลที่ส่งมาเป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับแสดงผลรูปที่ 3.5 การเชื่อมต่อ LCD

R/W (ขา 5 ) เป็นขาที่ใช้เลือกการอ่านหรือข้อมูลกับโมดูล LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6 ) เป็นขาสำหรับรับสัญญาณพัลส์เอ็นเนเบิลโมดูล LCD .ให้ทำงาน

D0-D7 (ขา 7-14) เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอกขนาด 8 บิต ซึ่งต่อกับพอร์ต 0 ของไมโครคอนโทรลเลอร์




รูปที่ 3.5 การเชื่อมต่อจอแสดงผล LCD กับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชุดสร้างฐานเวลาจริง(Real time clock) เลือกใช้ไอซีเบอร์ DS1307 ไอซีนี้จะมีวงจรกำเนิดสัญญาณนาฬิกาของมันเอง ทำงานอิสระจากระบบไมโครคอนโทรลเลอร์ รับส่งข้อมูลกับไมโครคอนโทรลเลอร์ได้โดยใช้บัสแบบ $I^2C$ สามารถให้ข้อมูลเกี่ยวกับเวลาไอซี นี้มีทั้งหมด 8 ขา เชื่อมต่อดังนี้

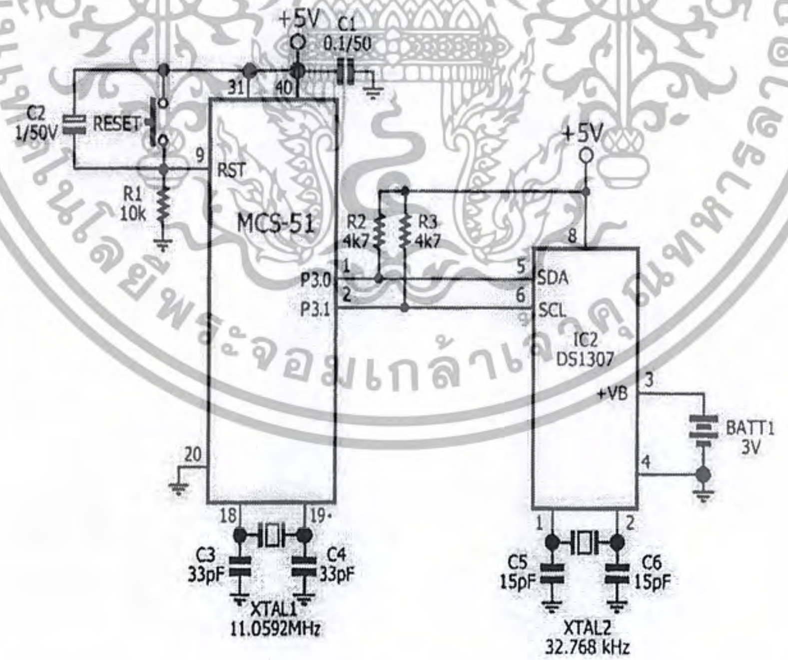
$V_{CC}$, GND (ขา8, ขา4) ใช้ต่อกับแหล่งจ่ายไฟเลี้ยงของระบบ

X1, X2 (ขา1, ขา2) ต่อกับคริสตอลความถี่ 32.768 kHz และตัวเก็บประจุ 12.5 pF

$V_{BAT}$(ขา3) ต่อกับแบตเตอรี่สำรองขนาด 3 โวลต์ เพื่อให้ไอซีทำงานขณะหยุดจ่ายไฟ

SDA, SCL (ขา5, ขา6) เป็นขาที่ใช้สำหรับสื่อสารทางบัสแบบ $I^2C$ ต่อเข้ากับ P3.0 และ P3.1 ตามลำดับ

จากโครงสร้างเราจะพบว่า วงจรออสซิลเลเตอร์จะทำให้เวลาในไอซี เดินไปอย่างต่อเนื่อง ถ้าหากต้องการอ่านค่าวันเวลาสามารถอ่านข้อมูลจาก RAM ในตำแหน่งเวลาที่ต้องการ ถ้าหากต้องการตั้งค่าเวลาใหม่ก็ให้เขียนข้อมูลลงไปใน RAM ตำแหน่งที่สอดคล้องกัน ในตัวไอซีจะมีวงจรควบคุมการจ่ายไฟเลี้ยง ถ้าหากแรงดัน $V_{CC}$ ตกลงมาต่ำกว่า $1.25 \times V_{BAT}$ ไอซีจะเข้าสู่โหมดทำงานแบบกินกระแสต่ำ โดยใช้พลังงานจากแบตเตอรี่สำรอง



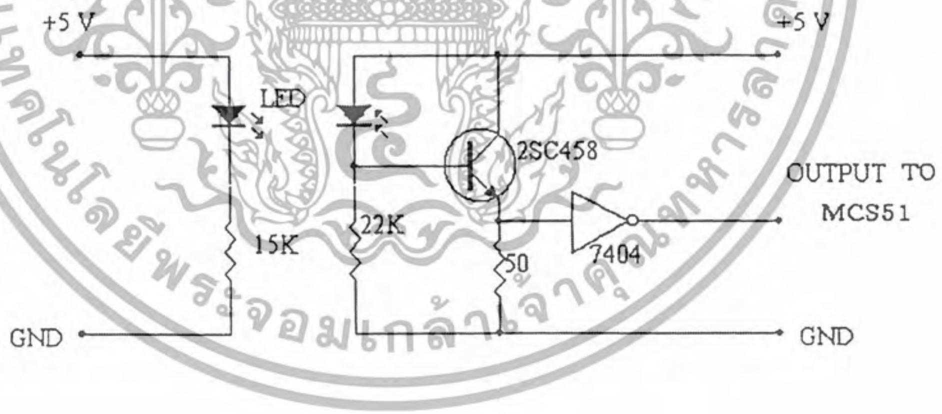

รูปที่ 3.6 แสดงการเชื่อมต่อไอซี DS1307 กับไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ชุดวงจรตรวจจับระดับโฟโต้เซนเซอร์ (Opto sensor)

ก) ชุดวัดปริมาณอาหารได้นำเอาโฟโต้เซนเซอร์(Opto sensor) มาใช้วัดระดับของอาหารปลาว่าได้จำนวนปริมาณที่ต้องการหรือยัง และตรวจสอบว่ามีอาหารปลาในถังหรือเปล่า โดยนำวงจรโฟโต้เซนเซอร์(Opto sensor) มาใช้ และนำสัญญาณของวงจรที่ได้ส่งไปที่ไมโครคอนโทรลเลอร์เป็น "0" และ "1" และเมื่อถึงระดับที่ต้องการไมโครคอนโทรลเลอร์จะทำการส่งสัญญาณมายังมอเตอร์ให้ ปิด – เปิด ลิ้นที่ท่อส่งอาหารปลา และยังสั่งให้แสดงผลว่าไม่มีอาหารเมื่ออาหารหมด

ข) ศึกษาและออกแบบสร้างวงจรตรวจจับโดยให้สามารถใช้ได้ในเครื่องจำลองโดยจะต้องมีระยะห่างของตัวรับและตัวส่งห่างกันประมาณ 2 - 5 เซนติเมตร วงจรตรวจจับด้วยแสงอินฟาเรดจะใช้โฟโต้ไดโอดเป็นตัวตรวจจับ โดยติดตั้งตัวตรวจจับนี้ไว้ตำแหน่งตรงข้ามกัน เมื่อตำแหน่งของอาหารปลาไหลมาปิดแสงอินฟาเรดระหว่างตัวส่งและตัวรับทำให้ได้สัญญาณ ลอจิกมีสถานะเป็น "0" แต่ถ้าไม่มีอาหารปลามาบังลำแสงอินฟาเรด จะได้สัญญาณที่มีสถานะเป็นลอจิก "1"




รูปที่ 3.7 วงจรตรวจจับโดยใช้แสงอินฟาเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจรเมื่อตัวรับแสงอินฟาเรดได้รับแสงจากตัวส่งทำให้กระแสไหลผ่านเข้าไปที่ขาเบสของทรานซิสเตอร์ทำงาน และจะมีแรงดันตกคร่อมที่ R 22 k$\Omega$ ซึ่งแรงดัน ตกคร่อมนี้คือแรงดันที่จะนำไปใช้งาน โดยแรงดันเอาต์พุตของวงจรจะบ่งบอกสถานะของลอจิกให้ชัดเจน คือ ถ้าแรงดันเกินประมาณ 2.3 โวลท์ จะทำให้สถานะลอจิกเป็น High แต่ถ้าแรงดันต่ำกว่า 0.8 โวลท์จะได้สถานะของลอจิกเป็น Low และแรงดันที่ได้นี้ ก็คือเอาต์พุตที่จะนำไปใช้ต่อ

เมื่อได้วงจรตรวจจับแล้วก็จะทำการออกแบบโครงสร้างจำลองจำนวน 2 ชุดเพื่อจะทำการติดตั้งวงจรตรวจจับทั้ง 2 ชุด ดังแสดงในรูปที่ 3.7

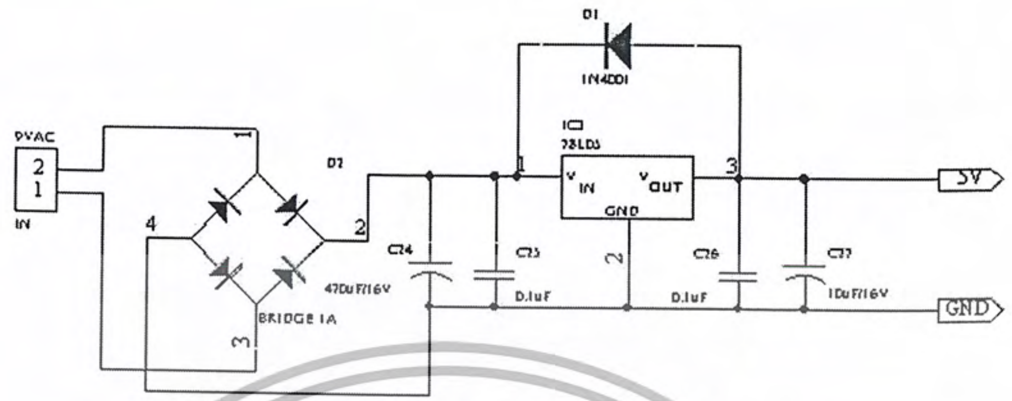
-ชุดขับมอเตอร์กระแสตรง

เนื่องจากไมโคคอนโทรลเลอร์จ่ายกระแสออกจากพอร์ตได้น้อยมาก ประมาณ 20 mA ไม่สามารถขับโหลดได้โดยตรงวงจรดังรูปที่ 3.8 จึงเป็นวงจรที่ต่อจากพอร์ต 1 ที่ใช้ควบมอเตอร์กระแสตรงทั้ง 2 ตัว จากการทดลองพบว่ามอเตอร์กระแสตรงที่ใช้กระแสสตาทร์สูงมาก ถ้าใช้แหล่งจ่ายเดียวกับไมโครคอนโทรลเลอร์จะทำให้ระบบทำงานผิดปรกติ เพื่อป้องกันจึงใช้รีเลย์มาเป็นตัวป้องกันปัญหาดังกล่าว

รูปที่ 3.8 ชุดขับมอเตอร์กระแสตรง

- ชุดวงจรไฟเลี้ยงวงจร (Supply) ส่วนของไฟเลี้ยง 5 V วงจรเรียงกระแส (Rectifier) ได้ใช้ไดโอดแบบบริดจ์ (Bridge rectifier) เพื่อให้ได้สัญญาณเต็มคลื่น (Full wave rectifier) จากนั้นใช้ (C4) ค่า 470 $\mu$F ต่อคร่อมเพื่อช่วยลดการกระเพื่อมของสัญญาณทำให้สัญญาณเรียบขึ้นและทำการควบคุมแรงดันโดยใช้ไอซีเบอร์ 7805 เพื่อรักษาระดับแรงดันให้คงที่ ที่ +5V โดยที่ (C7) ทำหน้าที่ปรับค่าตอบสนองในภาวะชั่วครู่ (Transient response) ของวงจรจะรักษาระดับแรงดันโดยตัวเก็บประจุจะทำหน้าที่เป็นแหล่งสะสมประจุเพื่อที่จะป้อนกระแสเข้าโหลดในช่วงสั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.9 วงจรไฟเลี้ยง 5 โวลท์

**3.4 วิธีการใช้งานเครื่องให้อาหารปลา**

รูปที่ 3.10 รูปเครื่องให้อาหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. จากรูปที่ 3.10 เมื่อกดสวิทช์ที่กล่องจ่ายไฟเพื่อเปิดเครื่อง ส่วนแสดงผลจะแสดงเวลาปัจจุบันในกรณีที่มีอาหารปลาอยู่ถังใส่อาหาร ถ้าไม่มีอาหารจะแสดงว่า NO FOOD รอจนกว่าจะเติมอาหารปลาลงไปในถัง จึงจะแสดงเวลาปัจจุบัน

2. กด 0 ที่คีย์แพดเมื่อต้องการตั้งค่าเวลาใหม่ หรือกำหนดเวลาจ่ายอาหารและจำนวนมื้อ ส่วนแสดงผลจะแสดงเป็นสองตัวเลือก

- ตัวเลือกที่หนึ่งเมือกดหมายเลข 1 จะเข้าสู่ขั้นตอนการตั้ง วัน เดือน ปี และเวลาปัจจุบัน โดยที่คีย์ * คือ การยกเลิก และคีย์ # คือการตกลง ถ้าต้องการตั้งค่าเวลาใหม่ก็ให้กด # ที่คีย์แพด แล้วตามด้วยตัวเลขของเวลา เมื่อใส่ครบหน้าจอจะเข้าสู่ฟังก์ชั่นปกติ

- ตัวเลือกที่สองเมื่อกดหมายเลข 2 จะเข้าสู่ขั้นตอนการตั้งค่าจำนวนมื้ออาหาร และเวลาของมื้อนั้น สามารถกำหนดได้ 1-8 มื้อ เมื่อเลือกมื้อแล้วต้องใส่ค่าเวลาของมื้อนั้นด้วย โดยที่คีย์ # คือการตกลง เมื่อใส่ตัวเลขเวลาครบ ส่วนแสดงผลจะกลับสู่ การแสดงเวลาปัจจุบัน เพื่อรอการทำงานเมื่อถึงเวลาที่ตั้งไว้เพื่อจ่ายอาหาร

3. กด * เมื่อต้องการให้ส่วนแสดงผลแสดงผลแสดงเวลาปัจจุบัน

4. เมื่อมีการหยุดการจ่ายไฟให้กับวงจร ไม่ต้องมีการตั้งค่าเวลาปัจจุบันใหม่เพราะมีแบตเตอรี่สำรองไว้จ่ายไฟแทนไว้ แต่ต้องมีการกำหนดจำนวนมื้อและเวลาในแต่ละมื้อใหม่อีกครั้งหนึ่ง ขั้นตอนการใช้งานแสดงในรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 แสดงวิธีใช้เครื่องให้อาหารปลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

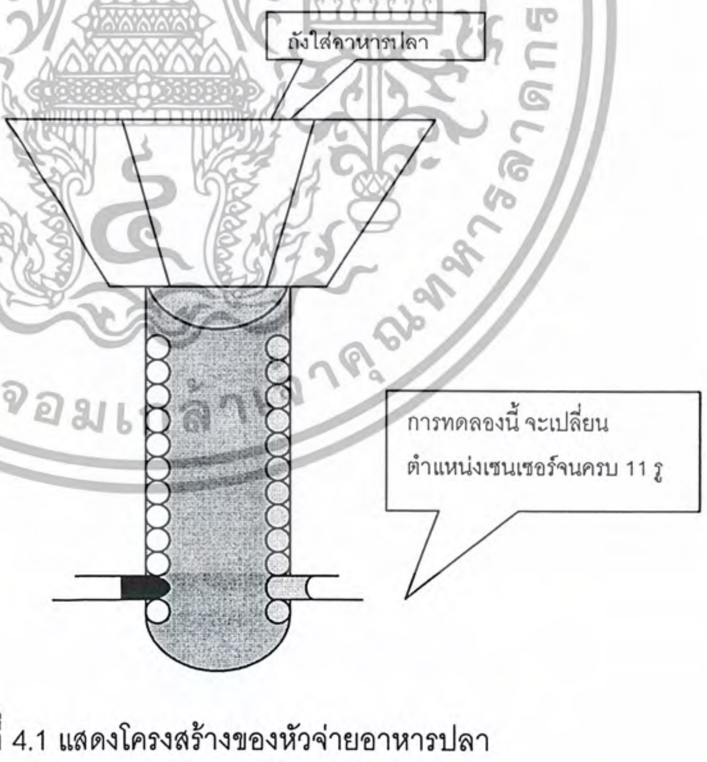
# บทที่ 4
# การทดลอง

**เงื่อนไขความต้องการปลา ที่ต้องการอาหารในปริมาณที่เพียงพอ และเป็นเวลาจึงออกแบบ** การทดลองเป็นสองเงื่อนไข

1. ปริมาณอาหารสามารถปรับความละเอียดได้หรือไม่ มีความเที่ยงตรงหรือไม่
2. เวลาการให้อาหารตรงกับที่ตั้งเวลาไว้หรือไม่

**4.1 การทดลองที่ 1 (ทดสอบเรื่องปริมาณอาหาร)**

เครื่องให้อาหารปลาได้ทำการเจาะรู 11 รู เพื่อติดตั้งเซนเซอร์ในช่องวัดปริมาณอาหาร และเมื่ออาหารมาบังเซนเซอร์ ไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์ตัวที่ 1 หยุดการจ่าย การทดลองนี้จึงออกแบบเพื่อทดสอบการปรับละเอียด และ ความเที่ยงตรงในการจ่ายอาหารในแต่ละครั้ง




รูปที่ 4.1 แสดงโครงสร้างของหัวจ่ายอาหารปลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ลำดับขั้นการทดลอง**

1.ติดตั้งเซนเซอร์ไว้ในตำแหน่งล่างสุดซึ่งเป็นตำแหน่งที่ติดตั้งเมื่อต้องการการจ่ายอาหารน้อยที่สุด จากนั้นเติมอาหารไปในถังจ่ายอาหาร ลิ้นจ่ายอาหารจะเปิดเป็นช่วงทำให้อาหารตกลงมาทีละน้อยประมาณ 0 –5 เม็ด เมื่ออาหารปลามาบังเซนเซอร์ก็จะหยุดการจ่ายอาหาร

2.นำอาหารที่ได้มาเก็บไว้ แล้วลองทดอีกครั้งโดยที่เซนเซอร์ยังอยู่ที่ตำแหน่งเดิมเพื่อทดสอบความเที่ยงตรง ทำแบบเดิม 4 ครั้ง

3.เมื่อครบ 4 ครั้งแล้ว ติดตั้งเซนเซอร์ที่รูที่ 2 เป็นการปรับระดับให้มีการจ่ายอาหารมากขึ้น ทดสอบที่ตำแหน่งเดิมจนครบ 4 ครั้ง แล้วจึงเปลี่ยนรู ทำจนครบ 11 รู

4.นำปริมาณอาหารที่ได้ในแต่ละครั้งมานับเม็ด บันทึกค่าลงตาราง

| รูที่ | ครั้งที่ 1 | ครั้งที่ 2 | ครั้งที่ 3 | ครั้งที่ 4 |
|---|---|---|---|---|
| รูที่ 1 | 34 เม็ด | 31 เม็ด | 35 เม็ด | 38 เม็ด |
| รูที่ 2 | 53 เม็ด | 59 เม็ด | 56 เม็ด | 52 เม็ด |
| รูที่ 3 | 69 เม็ด | 52 เม็ด | 62 เม็ด | 64 เม็ด |
| รูที่ 4 | 74 เม็ด | 73 เม็ด | 76 เม็ด | 80 เม็ด |
| รูที่ 5 | 94 เม็ด | 83 เม็ด | 87 เม็ด | 73 เม็ด |
| รูที่ 6 | 101 เม็ด | 100 เม็ด | 102 เม็ด | 96 เม็ด |
| รูที่ 7 | 109 เม็ด | 107 เม็ด | 106 เม็ด | 103 เม็ด |
| รูที่ 8 | 123 เม็ด | 126 เม็ด | 111 เม็ด | 120 เม็ด |
| รูที่ 9 | 130 เม็ด | 125 เม็ด | 140 เม็ด | 125 เม็ด |
| รูที่ 10 | 156 เม็ด | 153 เม็ด | 143 เม็ด | 138 เม็ด |
| รูที่ 11 | 153 เม็ด | 155 เม็ด | 162 เม็ด | 150 เม็ด |

**ตารางที่ 4.1 ผลการทดลองวัดปริมาณอาหารและความเที่ยงตรงในการจ่ายอาหาร**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| รูที่ | ค่าเฉลี่ยอาหาร (เม็ด) | ค่าผิดพลาดครั้งที่ 1 (%) | ค่าผิดพลาดครั้งที่ 2 (%) | ค่าผิดพลาดครั้งที่ 3 (%) | ค่าผิดพลาดครั้งที่ 4 (%) | ค่าผิดพลาดเฉลี่ย (%) |
|---|---|---|---|---|---|---|
| รูที่ 1 | 34.5 | 1.44 | 10.14 | -1.44 | -10.14 | 5.79 |
| รูที่ 2 | 55 | 3.63 | -7.27 | -1.18 | 5.45 | 4.38 |
| รูที่ 3 | 61.75 | -11.74 | 15.78 | -0.40 | -3.64 | 7.89 |
| รูที่ 4 | 75.75 | 2.31 | 3.63 | -0.33 | -5.61 | 2.97 |
| รูที่ 5 | 84.25 | -11.57 | 1.48 | -3.26 | 13.35 | 7.41 |
| รูที่ 6 | 99.75 | -1.25 | -0.25 | -2.25 | 3.75 | 1.87 |
| รูที่ 7 | 106.25 | -2.58 | -0.70 | 0.23 | 3.05 | 1.64 |
| รูที่ 8 | 120 | -2.50 | -5.0 | 7.5 | 0 | 3.75 |
| รูที่ 9 | 130 | 0 | 3.84 | -7.69 | 3.84 | 3.84 |
| รูที่ 10 | 147.5 | -5.76 | -3.72 | 3.05 | 6.44 | 4.74 |
| รูที่ 11 | 155 | 1.29 | 0 | -4.51 | 3.22 | 2.25 |

ตารางที่ 4.2 ตารางการคำนวณหาค่าผิดพลาดเฉลี่ย

การคำนวณ

ค่าเฉลี่ยอาหาร = {(ปริมาณอาหารครั้งที่ 1 + ปริมาณอาหารครั้งที่ 2 + ปริมาณอาหารครั้งที่ 3 + ปริมาณอาหารครั้งที่ 4 ) / 4 }

ค่าความผิดพลาด = {(ค่าเฉลี่ยอาหาร - ปริมาณอาหาร) / ค่าเฉลี่ยอาหาร } x 100%

ซึ่งจากการวัดเปรียบเทียบอาหาร ผลการทำงานของการวัดระดับอาหารโดยชุดอินฟาเรด พบว่าเกิดการคลาดเคลื่อนของปริมาณอาหารในแต่ละครั้ง ซึ่งค่าความคลาดเคลื่อนโดยรวมมีค่าเฉลี่ยประมาณ 4.22% อย่างไรก็ตามการมีรูปรับระดับชุดอินฟาเรด 11 รู ณ ระดับต่างๆ กันนั้น ช่วยให้สามารถปรับความละเอียดของปริมาณอาหารได้ในปริมาณต่างๆ กัน ได้ดี เป็นไปตามความต้องการของผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.2 การทดลองที่ 2 (ทดสอบเรื่องเวลา)**

การทดลองนี้เพื่อทดสอบการทำงานของเครื่องให้อาหารปลา ในเรื่องของเวลา ซึ่งบนหน้าจอแสดงผล LCD จะแสดงเวลาปัจจุบัน ในขั้นตอนการทดลองเนื่องจากเรากำหนดให้เครื่องให้อาหารปลาสามารถกำหนดมื้อได้ 1 – 8 มื้อ โดยที่ในแต่ละมื้อเราเป็นผู้กำหนดทางคีย์แพดว่าจะจ่ายอาหารเวลาใด ในขั้นตอนการทดลอง เราจะกำหนดให้มีการจ่ายทั้ง 8 มื้อแล้วสังเกตว่าในแต่ละมื้อตรงเวลาหรือไม่

| | ตั้งเวลาที่กำหนด | ทดลองครั้งที่1 | ทดลองครั้งที่ 2 | ทดลองครั้งที่ 3 |
|---|---|---|---|---|
| มื้อที่ 1 | 10.05 | 10.05 | 10.05 | 10.05 |
| มื้อที่ 2 | 11.05 | 11.05 | 11.05 | 11.05 |
| มื้อที่ 3 | 12.05 | 12.05 | 12.05 | 12.05 |
| มื้อที่ 4 | 13.05 | 13.05 | 13.05 | 13.05 |
| มื้อที่ 5 | 14.05 | 14.05 | 14.05 | 14.05 |
| มื้อที่ 6 | 15.05 | 15.05 | 15.05 | 15.05 |
| มื้อที่ 7 | 16.05 | 16.05 | 16.05 | 16.05 |
| มื้อที่ 8 | 17.05 | 17.05 | 17.05 | 17.05 |

ตารางที่ 4.3 ผลการทดลองของการตั้งเวลาให้อาหารปลา

ขั้นตอนการตั้งเวลาให้อาหาร ทำการตั้งเวลาให้อาหารปลาตามเวลาที่กำหนด หลังจากนั้นก็สังเกตการทำงานของเครื่องให้อาหารปลา สรุปว่าเครื่องให้อาหารปลาทำงานตรงเวลาที่กำหนด

พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# สรุปผลการทดลองและวิจารณ์

## 5.1 สรุปผล

ในการทำโครงงานปริญญานิพนธ์เครื่องให้อาหารปลาอัตโนมัติ จากผลการทดลองสามารถสรุปผลของโครงงานปริญญานิพนธ์ได้ดังนี้

1. การทดลองการวัดปริมาณอาหารปลาด้วย OPTO SENSOR พบว่าในการวัดปริมาณของอาหารปลามีความผิดพลาดของการวัดประมาณ 4.22%
2. การทดลองการตั้งเวลาให้อาหารปลาตามเวลาที่กำหนด จากการทดลองสามารถทำงานได้ตามเวลาที่ตั้งไว้

## 5.2 ปัญหาและวิธีแก้ไข

ในขั้นตอนการจัดทำโครงงานปริญญานิพนธ์ มีปัญหาอุปสรรคเกิดขึ้นมากมาย ทำให้ผู้จัดทำต้องเสียเวลาในการแก้ไขปัญหาไปพอสมควร รวมถึงงบค่าใช้จ่ายที่เพิ่มขึ้นจากการแก้ไขปัญหาที่เกิดขึ้น ดังนี้

1. ชุดรับส่งอินฟาเรดบางครั้งไม่สามารถทำงานได้ เนื่องจากการติดตั้งของตัวส่งและตัวรับของอินฟาเรดไม่ตรงกัน ดังนั้นจะต้องติดตั้งตัวรับและตัวส่งของอินฟาเรดให้ตรงกัน เพื่อให้วงจรอินฟาเรดทำงานได้ถูกต้อง

2. การวัดปริมาณอาหารในท่อวัดระดับเกิดการคาดเคลื่อนเนื่องจาก เมื่อทำการปล่อยอาหารจากลิ้นลงในช่องวัดปริมาณอาหารปลาในแต่ละครั้ง การเรียงตัวของอาหารไม่เหมือนกัน เช่น บางเวลาจุดยอดของการเรียงตัวของอาหารอยู่ตรงกลาง บางเวลาเรียงตัวเสมอกัน ทำให้ปริมาณอาหารในการวัดแต่ละครั้งไม่เท่ากัน ดังนั้นควรหาอุปกรณ์การวัดที่สามารถมาแก้ปัญหาดังที่กล่าวมาข้างต้น เช่น LOAD CELL

3.ชุดรับส่งอินฟาเรดมีปัญหาเรื่องแสงรบกวนมาก ในการนำมาใช้งานต้องมีการป้องกันแสงธรรมชาติด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**5.3 ข้อเสนอแนะ**

1. ควรนำ LOAD CELL มาใช้ในการวัดน้ำหนักของอาหารปลาในถัง ทำให้มีความเที่ยงตรงมากขึ้น และสามารถในการวัดละเอียดดีกว่า

2. ลิ้นจ่ายอาหารควรเลือกใช้วัสดุที่มีความแข็งที่เหมาะสมจะทำให้การ เปิด – ปิด ลิ้นโดยใช้มอเตอร์เป็นตัวบังคับไม่เกิดการทำให้มอเตอร์ดึงกระแสมากเพื่อเอาชนะแรงต้าน และถ้ามากเกินไปทำให้มอเตอร์เสียหายได้เสียหาย

3. ในการเลี้ยงปลาตู้ สิ่งที่เราต้องเอาใจใส่นอกจากระบบการให้อาหารแล้ว คือระบบคุณภาพน้ำและระบบความสะอาดของตู้ปลา ถ้าต้องการให้การเลี้ยงปลาตู้ถูกควบคุมอัตโนมัติเพื่อสุขภาพปลาที่แข็งแรง ควรเพิ่มระบบดังกล่าวเพื่อเพิ่มความสะดวกแก่ผู้เลี้ยงปลาตู้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <at89x52.h>
#include <stdio.h>
#include <absacc.h>
#include <ctype.h>
#include <stdlib.h>
#include <intrins.h>
#include <string.h>

#define   BUFSIZE    16     /* ASCII buffer size */
#define   FALSE      0
#define   TRUE       1
#define   LF         10     /* line feed chr */
#define   CR         13     /* return chr */
#define   RTCADD     0xd0   /* ds1307 RTC address */

#define   k_cancel   10
#define   k_ent      11

sbit   SCL   =  P3^1;  /* I2c SCL bit */
sbit   SDA   =  P3^0;  /* I2c SDA bit */

sbit   sensor  =  P2^7;   /*sensor food1*/
sbit   sensor1 =  P3^5;   /*sensor food2*/
sbit   motor   =  P1^1;
sbit   motor1  =  P1^0;

sbit   lcdrs  =  P3^7;
sbit   lcdrw  =  P3^2;
sbit   lcden  =  P3^6;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
char readKbd (void);

/* Gobal variable */
   char   data   a, b, c, d, k_a, k_b, TimeMin, TimeHour;
        char   data   ASCBUF[BUFSIZE];

        char   code   scankey[]={0xbf, 0xdf, 0xef};
        char   code   keycode[]={1, 4, 7, 10, 2, 5, 8, 0, 3, 6, 9, 11};

        bit    data   TimeFlag;

typedef struct CLOCK          /* Structure for RTC */
{
   char sec;
   char min;
   char hour;
   char day;
   char date;
   char month;
   char year;
} CLOCK;
CLOCK RTC;
              /************************/
              /*    Bit delay        */
              /************************/
void Bitdl (void)
{
   _nop_();   /* Delay time for read EEPROM */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   _nop_();
}
              /*****************************/
              /*   Clear(L) SCL line    */
              /*****************************/
void Clrscl (void)
{
   SCL = 0;      /* clear line */
   Bitdl();
}
              /*****************************/
              /*   Set(H) SCL line    */
              /*****************************/
void Setscl (void)
{
   SCL = 1;      /* set line   */
   while(!SCL);
   Bitdl();
}
           /********************************/
           /*   Pulse gen for SCL line    */
           /********************************/
void Plscl (void)
{
   Setscl();
   Clrscl();
}
           /********************************/
           /*   Set stop condition        */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          /*******************************/
void Stop (void)
{
   SDA = 0;Setscl();
   SDA = 1;Bitdl();
}
          /*******************************/
          /*    Set start condition     */
          /*******************************/
void Start (void)
{
   SDA = 1;
   Setscl();
   SDA = 0;
   Bitdl();
   Clrscl();
}
       /********************************************/
       /*    Receive data(byte) from I2C bus      */
       /********************************************/
char Rxbyte (void)
{
   unsigned char tmp, dat;

   dat = 0;
   for(tmp=1; tmp<=8; tmp++)   /* Send 1 byte to EEPROM */
   {
      dat <<= 1;Setscl();
      dat |= SDA;Clrscl();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    return(dat);
}
        /*******************************************/
        /*   Transmit data(byte) to I2C bus    */
        /*******************************************/
bit Txbyte (unsigned com)
{
    bit erflag;

    unsigned char tmp;
    for(tmp=1; tmp<=8; tmp++)
    {
        SDA  = com & 0x80;
        com <<= 1;
        Plscl();
    }

    SDA = 1;
    Setscl();
    if(SDA) erflag = 1;
    else   erflag = 0;
    Clrscl();
    return(erflag);
}
/* Get data from any register */
/* in -->> address(addr)   */
/* in -->> register(reg)   */
/* out <<-- data(*dat)     */
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*  err <<--  0 = no error,1 = error */

bit Getreg (char addr,char reg,char *dat)
{
   EA=0;
   if((SCL==0) || (SDA==0))
   {                    /* check line first      */
      EA=1;
      return(1);
   }
   Start();
   if(Txbyte(addr))
   {                    /* Xmit register address  */
      Stop();
      EA=1;
      return(1);
   }
   if(Txbyte(reg))
   {                    /* Xmit register command  */
      Stop();
      EA=1;
      return(1);
   }
   Start();             /* Set start condition   */
   if(Txbyte(addr|0x01))
   {                 /* Xmit register address + receive command */
      Stop();
      EA=1;
      return(1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }
    *dat = Rxbyte();        /* Recv data byte        */
    SDA = 1;
    Plscl();
    Stop();
    EA=1;
    return(0);          /* return data to caller  */
}
/*  Put data to any register   */
/*  in  -->>  address(addr)   */
/*  in  -->>  register(reg)   */
/*  in  -->>  data(dat)       */
/*  err <<--  0 = no error, 1 = error */

bit Putreg(char addr,char reg,char dat)
{
    EA=0;
    if((SCL==0) || (SDA==0))
    {                   /* Ckeck line first     */
      EA=1;
      return(1);          /* line is busy return 1 */
    }
    Start();              /* Set Start condition   */
    if(Txbyte(addr))
    {                     /* Xmit register address */
      Stop();
      EA=1;
      return(1);
    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if(Txbyte(reg))
  {                     /* Xmit register command */
    Stop();
    EA=1;
    return(1);
  }
  Txbyte(dat);             /* Xmit data           */
  Stop();               /* Set stop condition    */
  EA=1;
  return(0);
}
        /****************************************/
        /*   Get time from RTC(Ds1307)        */
        /****************************************/
bit GetRTC (void)
{
  Getreg(RTCADD,0,&RTC.sec);   /* get second      (00 - 59) */
  Getreg(RTCADD,1,&RTC.min);   /* get minute      (00 - 59) */
  Getreg(RTCADD,2,&RTC.hour);  /* get hour        (00 - 23) */
  Getreg(RTCADD,3,&RTC.day);   /* get day of week (01 - 07) */
  Getreg(RTCADD,4,&RTC.date);  /* get date        (01 - 31) */
  Getreg(RTCADD,5,&RTC.month); /* get month       (01 - 12) */
  Getreg(RTCADD,6,&RTC.year);  /* get year        (00 - 99) */
  return (0);
}
           /****************************/
           /*   Set time to rtc     */
           /****************************/
void SetRTC (void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
   Putreg(RTCADD,0,RTC.sec);  /* Put to RTC */
   Putreg(RTCADD,1,RTC.min);
   Putreg(RTCADD,2,RTC.hour);
   Putreg(RTCADD,4,RTC.date); /* Put to RTC */
   Putreg(RTCADD,5,RTC.month);
   Putreg(RTCADD,6,RTC.year);
}
         /*****************************************/
         /*   Delay function                     */
         /*   input : int loop counter           */
         /*****************************************/
void delay (unsigned int count)
{
   for (; count; count--)    /* Loop Delay */
   {
     if (RI==1)    /* Check Serial port recieve bit */
       break;
   }
}
           /*********************************/
           /*   Check Keyboard hit          */
           /*********************************/
bit kbhit (void)
{
   P2=0x0f;
   if ((P2&0x0f)!=0x0f)  /* Key hit return true */
     return (TRUE);
   else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      return (FALSE);    /* Non return false */
}
            /**********************/
            /*   Read keyboard  */
            /**********************/
char readKbd (void)
{
   while (1)
   {
      for (k_a=0; k_a<3; k_a++)  /* Loop scan 4 bit */
      {
         P2=scankey[k_a];   /* Check key press */
         k_b=P2&0x0f;
         if  (k_b==0x0f)
            continue;
         else
         {
            delay (3000);   /* Key debouce */
            k_b=P2&0x0f;
            if  (k_b==0x0f)
               continue;

            if  (k_b==0x0e)     /* bit 1 */
               k_b=0;
            if  (k_b==0x0d)     /* bit 2 */
               k_b=1;
            if  (k_b==0x0b)     /* bit 3 */
               k_b=2;
            if  (k_b==0x07)     /* bit 4 */
```

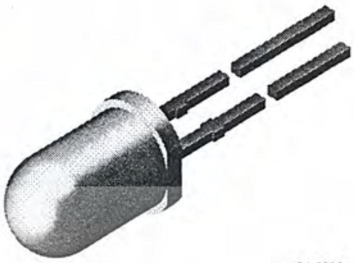
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# High Speed IR Emitting Diode in ∅ 5 mm (T-1¾) Package

## Description

TSHF5400 is a high speed infrared emitting diode in GaAlAs on GaAlAs double hetero (DH) technology, molded on copper frame, in a clear, untinted plastic package.

The new technology combines the high speed of DH-GaAlAs with the efficiency of standard GaAlAs and the low forward voltage of the standard GaAs technology.



94 8390

## Features

- High modulation bandwidth (10 MHz)
- Extra high radiant power and radiant intensity
- Low forward voltage
- Suitable for high pulse current operation
- Standard T-1¾ (∅ 5 mm) package
- Angle of half intensity $\varphi = \pm 22°$
- Peak wavelength $\lambda_p = 870$ nm
- High reliability
- Good spectral matching to Si photodetectors

TSHF5400 is ideal for the design of transmission systems according to IrDA requirements and for carrier frequency based systems (e.g. ASK / FSK - coded, 450 kHz or 1.3 MHz).

## Applications

Infrared high speed remote control and free air data transmission systems with high modulation frequencies or high data transmission rate requirements.

## Parts Table

| Part | Ordering code |
|---|---|
| TSHF5400 | IE31 - T 1 3/4 |

## Absolute Maximum Ratings

$T_{amb}$ = 25 °C, unless otherwise specified

| Parameter | Test condition | Symbol | Value | Unit |
|---|---|---|---|---|
| Reverse Voltage | | $V_R$ | 5 | V |
| Forward Current | | $I_F$ | 100 | mA |
| Peak Forward Current | $t_p/T = 0.5$, $t_p = 100$ µs | $I_{FM}$ | 200 | mA |
| Surge Forward Current | $t_p = 100$ µs | $I_{FSM}$ | 1.5 | A |
| Power Dissipation | | $P_V$ | 160 | mW |
| Junction Temperature | | $T_j$ | 100 | °C |
| Operating Temperature Range | | $T_{amb}$ | - 40 to + 100 | °C |
| Storage Temperature Range | | $T_{stg}$ | - 40 to + 100 | °C |
| Soldering Temperature | t ≤ 5 sec, 2 mm from case | $T_{sd}$ | 260 | °C |
| Thermal Resistance Junction/Ambient | | $R_{thJA}$ | 270 | K/W |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VISHAY

## Basic Characteristics

$T_{amb}$ = 25 °C, unless otherwise specified

| Parameter | Test condition | Symbol | Min | Typ. | Max | Unit |
|---|---|---|---|---|---|---|
| Forward Voltage | $I_F$ = 100 mA, $t_p$ = 20 ms | $V_F$ | | 1.35 | 1.6 | V |
| | $I_F$ = 1 A, $t_p$ = 100 µs | $V_F$ | | 2.4 | 3.0 | V |
| Temp. Coefficient of $V_F$ | $I_F$ = 100 mA | $TK_{VF}$ | | - 1.7 | | mV/K |
| Reverse Current | $V_R$ = 5 V | $I_R$ | | | 10 | µA |
| Junction Capacitance | $V_R$ = 0 V, f = 1 MHz, E = 0 | $C_j$ | | 160 | | pF |
| Radiant Intensity | $I_F$ = 100 mA, $t_p$ = 20 ms | $I_e$ | 25 | 40 | | mW/sr |
| | $I_F$ = 1 A, $t_p$ = 100 µs | $I_e$ | | 400 | | mW/sr |
| Radiant Power | $I_F$ = 100 mA, $t_p$ = 20 ms | $\phi_e$ | | 35 | | mW |
| Temp. Coefficient of $\phi_e$ | $I_F$ = 100 mA | $TK_{\phi e}$ | | - 0.7 | | %/K |
| Angle of Half Intensity | | $\varphi$ | | ± 22 | | deg |
| Peak Wavelength | $I_F$ = 100 mA | $\lambda_p$ | | 870 | | nm |
| Spectral Bandwidth | $I_F$ = 100 mA | $\Delta\lambda$ | | 40 | | nm |
| Temp. Coefficient of $\lambda_p$ | $I_F$ = 100 mA | $TK_{\lambda p}$ | | 0.2 | | nm/K |
| Rise Time | $I_F$ = 100 mA | $t_r$ | | 30 | | ns |
| Fall Time | $I_F$ = 100 mA | $t_f$ | | 30 | | ns |
| Virtual Source Diameter | method: 63% encircled energy | ∅ | | 2.2 | | mm |

## Typical Characteristics ($T_{amb}$ = 25 °C unless otherwise specified)

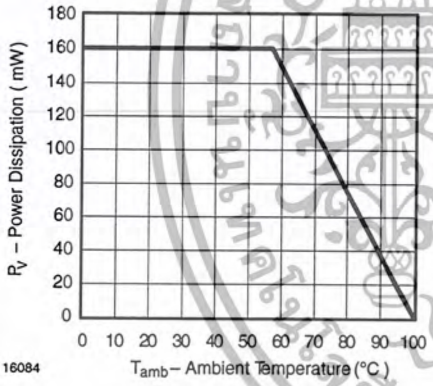



Figure 1. Power Dissipation vs. Ambient Temperature

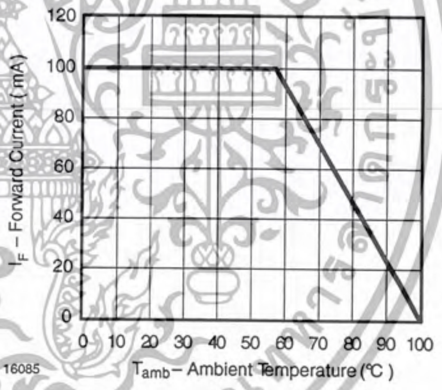



Figure 2. Forward Current vs. Ambient Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

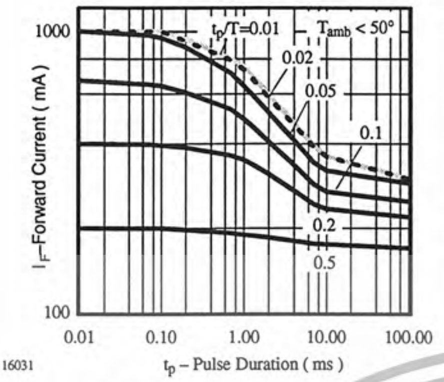



Figure 3. Pulse Forward Current vs. Pulse Duration

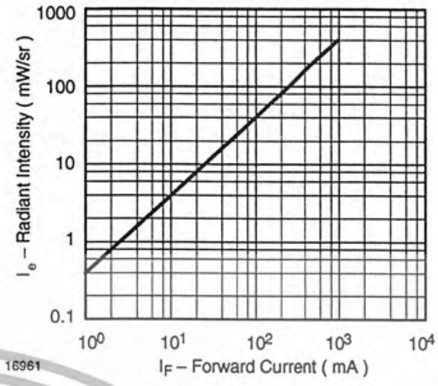



Figure 6. Radiant Intensity vs. Forward Current

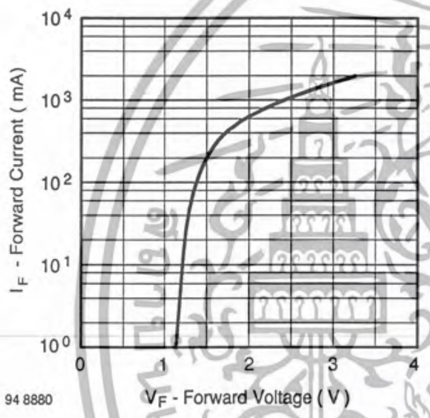



Figure 4. Forward Current vs. Forward Voltage

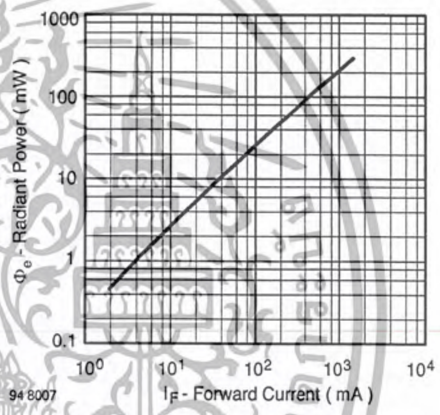



Figure 7. Radiant Power vs. Forward Current

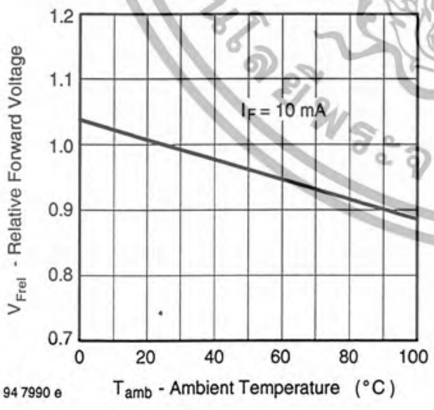



Figure 5. Relative Forward Voltage vs. Ambient Temperature

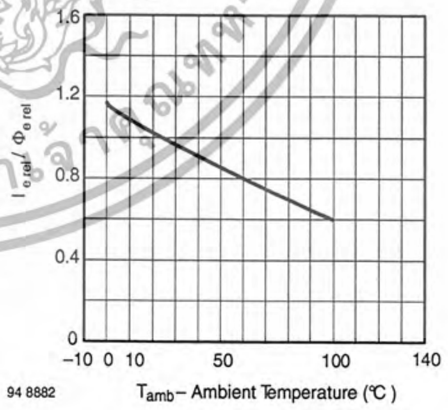



Figure 8. Rel. Radiant Intensity/Power vs. Ambient Temperature



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

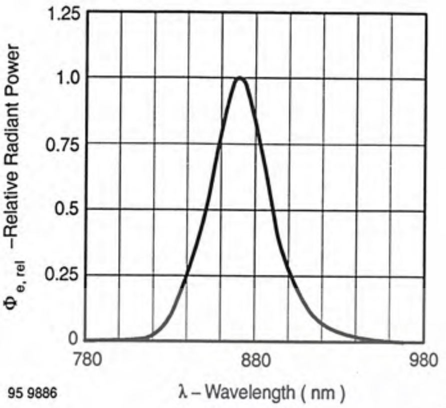



Figure 9. Relative Radiant Power vs. Wavelength

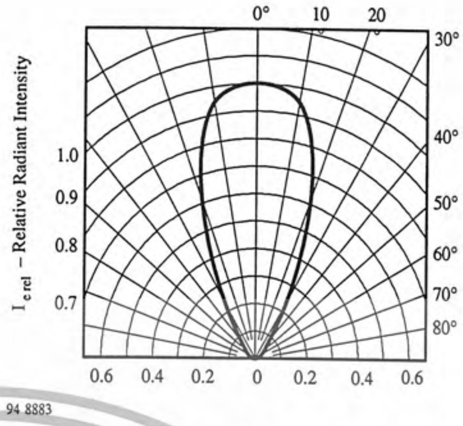
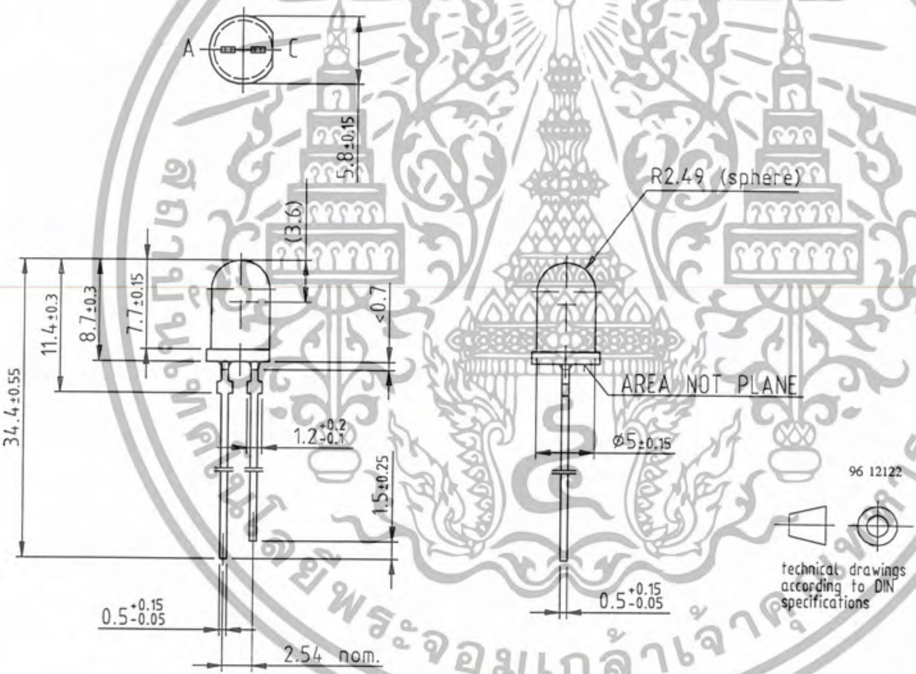



Figure 10. Relative Radiant Intensity vs. Angular Displacement

## Package Dimensions in mm






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Ozone Depleting Substances Policy Statement

It is the policy of **Vishay Semiconductor GmbH** to

1. Meet all present and future national and international statutory requirements.
2. Regularly and continuously improve the performance of our products, processes, distribution and operatingsystems with respect to their impact on the health and safety of our employees and the public, as well as their impact on the environment.

It is particular concern to control or eliminate releases of those substances into the atmosphere which are known as ozone depleting substances (ODSs).

The Montreal Protocol (1987) and its London Amendments (1990) intend to severely restrict the use of ODSs and forbid their use within the next ten years. Various national and international initiatives are pressing for an earlier ban on these substances.

**Vishay Semiconductor GmbH** has been able to use its policy of continuous improvements to eliminate the use of ODSs listed in the following documents.

1. Annex A, B and list of transitional substances of the Montreal Protocol and the London Amendments respectively
2. Class I and II ozone depleting substances in the Clean Air Act Amendments of 1990 by the Environmental Protection Agency (EPA) in the USA
3. Council Decision 88/540/EEC and 91/690/EEC Annex A, B and C (transitional substances) respectively.

**Vishay Semiconductor GmbH** can certify that our semiconductors are not manufactured with ozone depleting substances and do not contain such substances.

**We reserve the right to make changes to improve technical design and may do so without further notice.**

Parameters can vary in different applications. All operating parameters must be validated for each customer application by the customer. Should the buyer use Vishay Semiconductors products for any unintended or unauthorized application, the buyer shall indemnify Vishay Semiconductors against all claims, costs, damages, and expenses, arising out of, directly or indirectly, any claim of personal damage, injury or death associated with such unintended or unauthorized use.

Vishay Semiconductor GmbH, P.O.B. 3535, D-74025 Heilbronn, Germany
Telephone: 49 (0)7131 67 2831, Fax number: 49 (0)7131 67 2423



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้