สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ไมโครเมาส์

# MICROMOUSE

โดย


| | | |
|---|---|---|
| นาย ฐิติกร | ลัคนาวัฒน์ | รหัส 40013404 |
| นาย นิเวศน์ | สุภาภา | รหัส 40013413 |
| นาย ประพนธ์ | เลิศเบญจมงคล | รหัส 40013415 |


ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เลขหมู่.....
เลขทะเบียน 36820
วัน, เดือน, ปี 29 ส.ค. 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ไมโครเมาส์

**Micromouse**


ผู้จัดทำ

1. นาย ฐิติกร ลัดนาวัฒน์
2. นาย นิเวศน์ สุภาภา
3. นาย ประพนธ์ เลิศเบญจมงคล


..... อาจารย์ที่ปรึกษา

( รศ. ดร. ฟูศักดิ์ ชีวสุวิทย์ )


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ไมโครเมาส์

ฐิติกร ลัคนาวัฒน์
นิเวศน์ สุภาภา
ประพนธ์ เลิศเบญจมงคล

อาจารณ์ที่ปรึกษา
รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์

## บทคัดย่อ

โครงงานไมโครเมาส์ได้สร้างขึ้นเพื่อใช้เป็นแนวทางการศึกษาหุ่นยนต์ที่มีความสามารถในการตัดสินใจเองได้ ซึ่งในโครงงานนี้ได้ใช้ไมโครคอนโทรลเลอร์ตระกูล 8051 เป็นตัวประมวลผลกลาง (ซีพียู) โดยตัวไมโครเมาส์นี้จะรับข้อมูล และโปรแกรมผ่านทางช่องสื่อสารอนุกรมของเครื่องคอมพิวเตอร์ที่อยู่ในรูปเลขฐานสิบหก การทดสอบไมโครเมาส์ จะทำทดสอบโดยการเปลี่ยนแปลงรูปแบบของเขาวงกตไปมา ไมโครเมาส์จะเคลื่อนที่ค้นหาเส้นทางจากจุดเริ่มต้นไปยังจุดมุ่งหมายที่ต้องการได้ โดยการตัดสินใจด้วยตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MicroMouse

Thitikorn Luckanawat

Niwate Supapa

Prapon Lerdbenjamongkol

Advisor

Assoc.Prof. Dr. Fusak Cheevasuvit

## Abstract

This project presents a method for implementing a micromouse which can be self making decision. The microcontroller 8051 is used as the CPU of the micromouse. The control program is designed on the personal computer and sent to the micromouse via the serial port. By assigning the destination point, the micromouse can be automatically searching the route.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

กว่าจะได้โครงสร้างทางกลของไมโครเมาส์ ให้พวกเราสานงานต่อได้นั้น ดูเหมือนเป็นอุปสรรคแรกที่สำคัญของเรา ซึ่งต้องขอบใจน้องช่างกลคนหนึ่งที่เป็นธุระให้ จากนั้นได้มาดัดแปลงชิ้นส่วนทางกลต่างๆที่โรงฝึกงานพื้นฐาน ของสถาบันเรา ต้องขอบคุณ คุณลุงผู้ดูแลโรงฝึกที่ได้หยิบยื่นเครื่องมือให้ใช้ ขอบคุณเจ้าหน้าที่พัสดุ (Store Keeper) ที่ให้ความช่วยเหลือด้านการยืมสนามเขาวงกต เพื่อนๆที่ได้ช่วยในเรื่องจิปาถะนานา ส่วนโปรแกรมรับข้อมูลจากคอมพิวเตอร์ที่อยู่ใน 89C52 เขียนโดยเครื่องเขียนไมโครคอนโทรลเลอร์ Atmel ของ อาจารย์ ทรงชัย วีระทวีมาศ

ขอบพระคุณสำหรับความสะดวกในทุกเรื่องที่ รศ. ดร. ฟูศักดิ์ ชีวสุวิทย์ อาจารย์ที่ปรึกษาของเราได้อำนวยให้

และบุคคลที่เราไม่เคยลืมตลอดเวลา พระคุณพ่อ และแม่ที่ส่งเสียให้เราได้ร่ำเรียน...

คณะผู้จัดทำ

our_mouse@usa.net

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| หัวเรื่อง | หน้า |
|---|---|
| **บทคัดย่อ ภาษาไทย** | **I** |
| **บทคัดย่อ ภาษาอังกฤษ** | **II** |
| **กิตติกรรมประกาศ** | **III** |
| **สารบัญ** | **IV** |
| **สารบัญรูปภาพ** | **VII** |
| **สารบัญตาราง** | **IX** |
| **บทที่ 1 บทนำ** | |
| 1.1 บทนำ | 1 |
| 1.2 วัตถุประสงค์ | 1 |
| **บทที่ 2 ทฤษฎี และหลักการ** | |
| 2.1 ลักษณะโดยรวมของไมโครเมาส์ | 2 |
| 2.1.1 มาตรฐานของเขาวงกต | 2 |
| 2.1.2 มาตรฐานของไมโครเมาส์ | 3 |
| 2.2 โครงสร้างสถาปัตยกรรม MCS-51 | 4 |
| 2.2.1 การจัดการทางสถาปัตยกรรม | 7 |
| 2.2.2 การเข้าถึงหน่วยความจำภายนอก | 7 |
| 2.2.3 สัญญาณ PSEN | 9 |
| 2.2.4 สัญญาณ ALE | 9 |
| 2.2.5 ตัวจับเวลา/ตัวนับ (Timer/Counter) | 10 |
| 2.2.5.1 ตัวจับเวลา/ตัวนับ0 และตัวจับเวลา/ตัวนับ1 | 10 |
| 2.2.5.2 ตัวจับเวลา/ตัวนับ 2 | 13 |
| 2.2.6 รีเซ็ต (RESET) | 16 |
| 2.2.7 Power-on reset | 18 |
| 2.2.8 การจัดการหน่วยความจำ | 19 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)


| หัวเรื่อง | หน้า |
|---|---|
| 2.3 8255 | 20 |
| 2.3.1 ลักษณะพื้นฐานของ 8255 | 20 |
| 2.3.2 การจำแนกกลุ่มของพอร์ต 8255 | 21 |
| 2.3.3 รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255 | 22 |
| 2.3.4 การเชื่อมต่อ 8255 กับ 8051 | 23 |
| 2.3.5 การเชื่อมต่อ MCS-51 กับหน่วยความจำ | 26 |
| 2.3.6 การต่อ MCS-51 เข้ากับ EPROM และ RAM | 29 |
| 2.4 สเต็ปปิ้งมอเตอร์ | 29 |
| 2.4.1 ชนิดวาริเอเบิลรีลักแตนซ์ (Variable reluctance : VR) | 29 |
| 2.4.2 ชนิดเพอร์มาเนนต์แมกเนต (Permanent magnet : PM) | 30 |
| 2.4.3 ชนิดไฮบริด (Hybrid) | 30 |
| 2.4.4 การพันขดลวด หรือคอยล์บนสเต็ปปิ้ง | 30 |
| 2.4.4.1 สเต็ปปิ้งมอเตอร์แบบไบโพล่าร์ (Bipolar) | 30 |
| 2.4.4.2 สเต็ปปิ้งมอเตอร์แบบยูนิโพล่าร์ (Unipolar) | 31 |
| 2.4.5 การกระตุ้น และการควบคุมการหมุนของสเต็ปปิ้งมอเตอร์ | 31 |
| 2.4.6 ซัพเพรสเซอร์ | 33 |
| **บทที่ 3 การออกแบบ** | |
| 3.1 ภาคเซ็นเซอร์ | 36 |
| 3.2 ไมโครคอนโทรลเลอร์ | 37 |
| 3.3 ภาคชุดขับ และสเต็ปปิ้งมอเตอร์ | 37 |
| 3.4 ด้าน Mechanic | 37 |
| 3.5 ด้านโปรแกรม | 37 |
| 3.5.1 ส่วนของโปรแกรมรับ HEX ไฟล์ | 37 |
| 3.5.2 ส่วนของโปรแกรมการปรับล้อ | 38 |
| 3.5.3 ส่วนโปรแกรมเพื่อจดจำเส้นทาง | 41 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)


| หัวเรื่อง | หน้า |
|---|---|
| **บทที่ 4 การทดลอง** | |
| 4.1 ภาคเซ็นเซอร์ | 54 |
| 4.2 ด้านโปรแกรม | 54 |
| 4.2.1 การทดลองโปรแกรมรับ HEX ไฟล์ | 54 |
| 4.2.2 การทดลองโปรแกรมการปรับล้อ | 55 |
| 4.2.3 การทดลองโปรแกรมโดยรวมของไมโครเมาส์ | 55 |
| **บทที่ 5 วิจารณ์ และสรุป** | |
| 5.1 ปัญหา และอุปสรรคที่พบ | 59 |
| 5.2 แนวทางในการพัฒนา | 59 |
| **บรรณานุกรม** | 60 |
| **ภาคผนวก** | 61 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ


| หัวเรื่อง | หน้า |
|---|---|
| รูปที่ 2.1 แสดงไมโครเมาส์ขณะเดินในเขาวงกต | 2 |
| รูปที่ 2.2 แสดงลักษณะรูปแบบของไมโครเมาส์ | 3 |
| รูปที่ 2.3 การจัดขาลักษณะภายนอกของ MCS-51 | 5 |
| รูปที่ 2.4 แสดงโครงสร้างสถาปัตยกรรมภายในของ MCS-51 | 8 |
| รูปที่ 2.5 แสดงจังหวะการเข้าถึงหน่วยความจำภายนอก | 10 |
| รูปที่ 2.6 แสดงการทำงานในโหมด 1 ของตัวจับเวลา และตัวนับ 1 | 11 |
| รูปที่ 2.7 แสดงการทำงานในโหมด 2 ของตัวจับเวลา และตัวนับ 1 | 12 |
| รูปที่ 2.8 แสดงการทำงานในโหมด 3 ของตัวจับเวลา และตัวนับ 1 | 13 |
| รูปที่ 2.9 แสดงการทำงานในโหมดของตัวจับเวลา/ตัวนับ 2 | 14 |
| รูปที่ 2.10 แสดงการทำงานในโหมดแบบบรรจุอัตโนมัติของตัวจับเวลา/ตัวนับ 2 | 15 |
| รูปที่ 2.11 แสดงแผนภูมิการรีเซ็ต | 16 |
| รูปที่ 2.12 แสดงวงจรการจัดการรีเซ็ตของ HMOS | 17 |
| รูปที่ 2.13 วงจรการจัดการรีเซ็ตของ CHMOS | 17 |
| รูปที่ 2.14 วงจรPower-on reset | 18 |
| รูปที่ 2.15 แสดงการส่งค่าของข้อมูลควบคุมการทำงานภายใน 8255 | 23 |
| รูปที่ 2.16 แสดงการทำงานของ 8255 ใน MODE 0 | 25 |
| รูปที่ 2.17 แสดงการทำงานของ 8255 ใน MODE 0 (ต่อ) | 26 |
| รูปที่ 2.18 แสดงการใช้ไดโอดซัพเพรสเซอร์ | 33 |
| รูปที่ 2.19 แสดงการใช้ไดโอด และรีซิสเตอร์ซัพเพรสเซอร์ | 34 |
| รูปที่ 2.20 แสดงการใช้ซีเนอร์ไดโอดซัพเพรสเซอร์ | 34 |
| รูปที่ 2.21 แสดงการใช้คอนเดนเซอร์ซัพเพรสเซอร์ | 35 |
| รูปที่ 3.1 แสดงการวางตำแหน่งของเซ็นเซอร์ | 36 |
| รูปที่ 3.2 แสดงเมื่อไมโครเมาส์อยู่ในตำแหน่งวิ่งตรง | 39 |
| รูปที่ 3.3 แสดงเมื่อไมโครเมาส์อยู่ในตำแหน่งเอียงไปทางขวา | 39 |
| รูปที่ 3.4 แสดงเมื่อไมโครเมาส์อยู่ในตำแหน่งเอียงไปทางซ้าย | 39 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)

หัวเรื่อง หน้า


**รูปที่ 3.5** แสดงการป้อนสัญญาณ Clock ให้ล้อขวาหมุนเร็วกว่าล้อซ้าย 40

**รูปที่ 3.6** แสดงการตรวจพบกำแพงด้านหน้า และเดินตรงต่อเพื่อหาจุดหมุน 42

**รูปที่ 3.7** แสดงการตรวจจับกำแพงด้านข้างเกิดการผิดพลาด 42

**รูปที่ 3.8** แสดงช่วงการตรวจจับกำแพงด้านข้าง 43

**รูปที่ 3.9** แสดงไมโครเมาส์เกิดการเข้าใจผิดว่ามีกำแพงด้านข้างจึงหมุนตัว 180° 43

**รูปที่ 3.10** แสดงไมโครเมาส์เมื่อเดินไม่ครบบล็อก 44

**รูปที่ 3.11** แสดงไมโครเมาส์เมื่อเดินเกิน 1 บล็อก 44

**รูปที่ 3.12** แสดง Flow Chart การทำงาน MAIN หลัก 1 45

**รูปที่ 3.13** แสดง Flow Chart การทำงาน MAIN หลัก 2 46

**รูปที่ 3.14** แสดง Flow Chart การตรวจสอบตำแหน่งกำแพงด้านข้าง 46

**รูปที่ 3.15** แสดง Flow Chart การทำงาน Block Pattern 47

**รูปที่ 3.16** แสดง Flow Chart การทำงาน Block Pattern (ต่อ) 48

**รูปที่ 3.17** แสดง Flow Chart การทำงาน T2 INT 49

**รูปที่ 3.18** แสดง Flow Chart การเดินตรง 50

**รูปที่ 3.19** แสดง Flow Chart การตั้งค่าพิกัดของไมโครเมาส์ 51

**รูปที่ 4.1** แสดงไมโครเมาส์วิ่งในเขาวงกตแบบที่ 1 56

**รูปที่ 4.2** แสดงไมโครเมาส์วิ่งในเขาวงกตแบบที่ 2 57



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

หัวเรื่อง หน้า


**ตารางที่ 2.1** แสดงค่าต่างๆในเรจิสเตอร์หลังการรีเซ็ต 19

**ตารางที่ 2.2** แสดงการกำหนดรูปแบบการทำงานของพอร์ต 21

**ตารางที่ 2.3** แสดงความหมายของเส้นสัญญาณต่างๆ 22

**ตารางที่ 2.4** แสดงการระบุเรจิสเตอร์ หรือพอร์ตภายใน 8255 23

**ตารางที่ 2.5** แสดงการอ่าน หรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับเรจิสเตอร์ 24

**ตารางที่ 2.6** แสดงรูปแบบการกำหนดบิต 25

**ตารางที่ 2.7** แสดงลำดับการกระตุ้นเพื่อให้สเต็ปปิ้งมอเตอร์ทำงานด้วย
การกระตุ้นแบบเวฟ 31

**ตารางที่ 2.8** แสดงลำดับการกระตุ้นเพื่อให้สเต็ปปิ้งมอเตอร์ทำงานด้วย
การกระตุ้นแบบ 2 เฟส 32

**ตารางที่ 2.9** แสดงลำดับการกระตุ้นเพื่อให้สเต็ปปิ้งมอเตอร์ทำงานด้วย
การกระตุ้น แบบครึ่งสเต็ป 33



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 บทนำ

เนื่องจากปัจจุบันนี้ ได้มีการนำหุ่นยนต์ในลักษณะต่างๆมาช่วยในการเพิ่มผลผลิตในงานอุตสาหกรรม ซึ่งการออกแบบ และการทำงานของหุ่นยนต์จะมีความซับซ้อนมาก ซึ่งจำเป็นต้องมีหน่วยที่ควบคุมในส่วนนี้ ส่วนมากแล้วจะใช้ไมโครโปรเซสเซอร์มาทำหน้าที่ตรงส่วนนี้ โดยไมโครโปรเซสเซอร์ จะทำหน้าที่ประมวลผลเพื่อไปควบคุมหุ่นยนต์ให้ทำงานในลักษณะต่างๆ ในกระบวนการผลิต

จากข้างต้นจึงได้เกิดแนวคิดที่จะออกแบบจำลองการทำงานของหุ่นยนต์เบื้องต้นซึ่งก็ คือ ไมโครเมาส์ โดยไมโครเมาส์จะเป็นรูปแบบการจำลองการทำงานคล้ายหุ่นยนต์ที่มีลักษณะ เป็นรถจำลองเคลื่อนที่อัตโนมัติภายในเส้นทางที่มีลักษณะเป็นเขาวงกต และสามารถที่จะหาทางออก หรือจุดที่กำหนดให้เป็นจุดสิ้นสุดได้เอง

จากโครงงานนี้ จะช่วยให้ผู้ที่สนใจทางด้านหุ่นยนต์อาจเกิดแนวคิด และแนวทางที่จะพัฒนาและสร้างสรรค์งานที่จำเป็นที่ต้องมีการใช้ไมโครคอนโทรลเลอร์ มาควบคุมด้านงานกล และพัฒนาเป็นหุ่นยนต์ที่มีความซับซ้อนต่อไป

## 1.2 วัตถุประสงค์

1. เพื่อเป็นแนวคิดเบื้องต้นในการที่จะพัฒนา หรือนำไปควบคุมหุ่นยนต์ขนาดเล็กต่อไป
2. เพื่อศึกษา และนำไมโครคอนโทรลเลอร์ ตระกูล MCS-51 มาประยุกต์ใช้งานในการควบคุม
3. เพื่อศึกษาการทำงานของ สเต็ปปิ้งมอเตอร์ และนำมาประยุกต์ใช้งานได้
4. เพื่อศึกษา การทำงานของ เซ็นเซอร์ และนำมาประยุกต์ใช้งานได้
5. เพื่อศึกษา และเขียนโปรแกรมให้สามารถทำงานได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

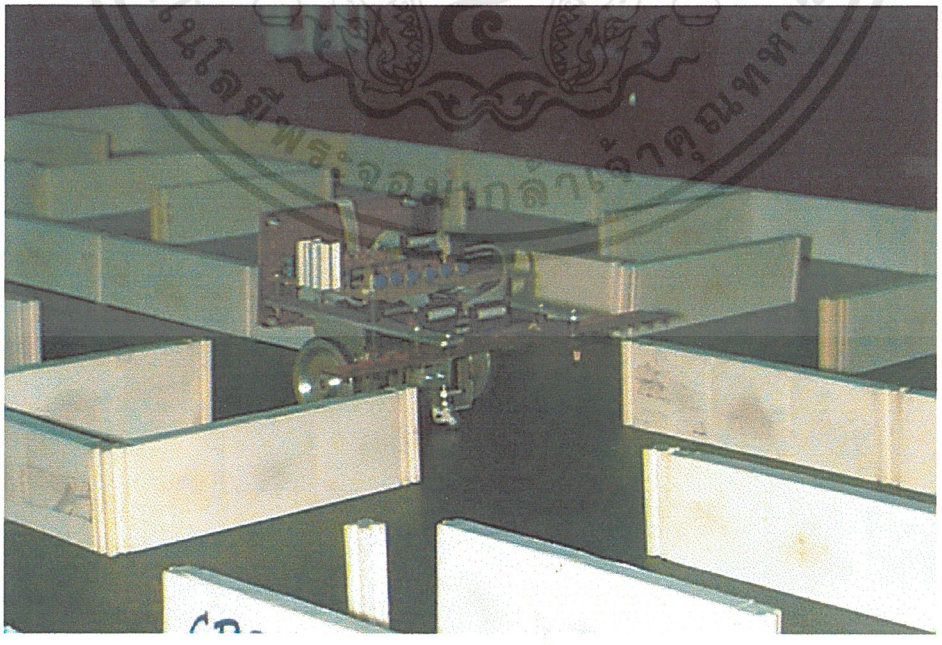
# บทที่ 2
# ทฤษฎี และหลักการ

## 2.1 ลักษณะโดยรวมของไมโครเมาส์

ไมโครเมาส์ คือ หุ่นยนต์ (ROBOT) ที่มีลักษณะเป็นรถขนาดเล็กที่เคลื่อนที่เองได้ โดยสามารถจะเรียนรู้หาจุดปลายทางของเขาวงกตที่ไม่ทราบรูปแบบล่วงหน้าได้ ซึ่งการที่จะทำเช่นนี้ได้นั้น ไมโครเมาส์จะต้องมีความคิดในการตัดสินใจ วิธีการตรวจจับกำแพงที่ไว้ใจได้ การควบคุมสเต็ปปิ้งมอเตอร์ที่มีประสิทธิภาพ และชิ้นส่วนทางด้านกลที่เที่ยงตรง ดังนั้นผู้สร้างจะต้องมีความรู้ และทักษะด้านโปรแกรม ไมโครคอนโทรลเลอร์ การออกแบบวงจรอิเลคทรอนิคส์ และด้านกล จะเห็นได้ว่าการได้ทำไมโครเมาส์ขึ้นมาสักตัว เปรียบเสมือนเป็นการสร้างสื่อที่เป็นประโยชน์ต่อการฝึก และการสอนให้กับนักศึกษา และผู้สนใจทางด้านวิศวกรรม และเทคโนโลยี

### 2.1.1 มาตรฐานของเขาวงกต

เขาวงกตจะประกอบไปด้วยเส้นทางที่สลับซับซ้อนขนาด 16 x 16 บล็อก แต่ละบล็อกมีขนาด 18 x 18 เซนติเมตร กำแพงที่นำมาประกอบเป็นเขาวงกต จะต้องมีสีขาวแต่ละชิ้นมีความสูง 5 เซนติเมตร หนา 1.2 เซนติเมตร ความกว้างของช่องทางเดิน 16.8 เซนติเมตร รอบนอกของเขาวงกตจะต้องมีกำแพงปิดล้อม ส่วนพื้นควรทำด้วยไม้ที่ได้ทาสีดำชนิดไม่ขึ้นเงา และทำให้ลื่น



รูปที่ 2.1 แสดงไมโครเมาส์ขณะเดินในเขาวงกต

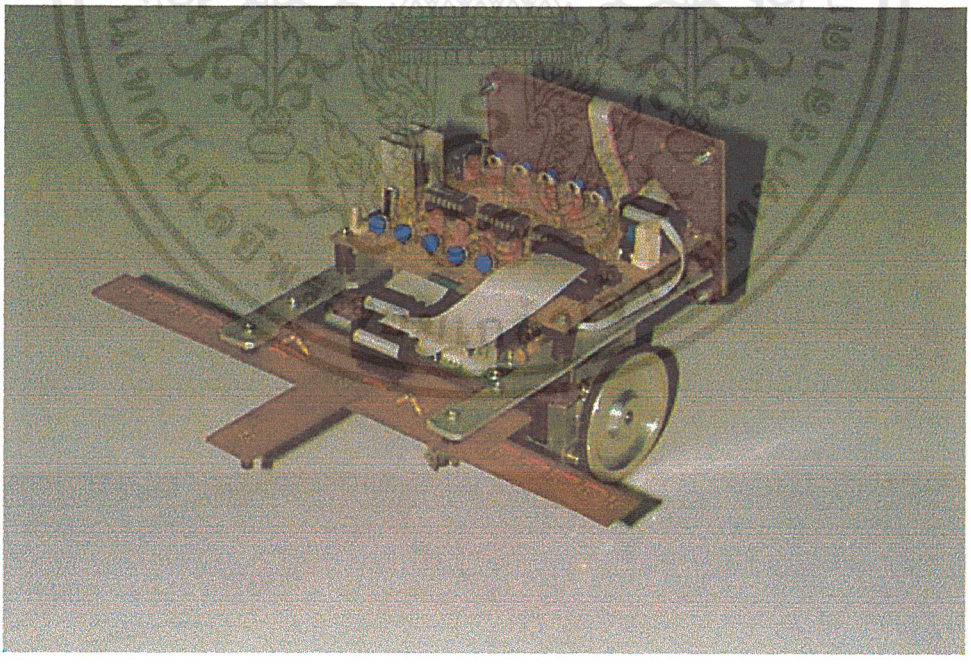
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดเริ่มต้นของการเคลื่อนที่ควรจะเป็นมุมใดมุมหนึ่งของเขาวงกต บล็อกที่เป็นจุดเริ่มต้นมีกำแพงล้อมไว้ทั้งสามด้าน ด้านที่ว่างคือ ทิศเหนือ ส่วนด้านข้างทั้งสองคือ ทิศตะวันตก และทิศใต้ช่วงกลางของเขาวงกตไม่ควรมีการกั้นกำแพง เพื่อจะได้มีทางเลือกที่หลากหลาย ส่วนปลายทางของการเคลื่อนที่ควรเป็นบล็อกที่อยู่ตรงกลางของเขาวงกต

เสาช่วยประกอบกำแพง มีขนาด 1.2 x 1.2 x ความสูง 5 เซนติเมตร การที่จะตั้งเสาได้ควรจะมีกำแพงประกอบด้วยอย่างน้อยหนึ่งด้าน ยกเว้นบล็อกปลายทาง

**2.1.2 มาตรฐานไมโครเมาส์**

ตัวไมโครเมาส์ จะต้องมีขนาดไม่เกินไปกว่า 25 x 25 เซนติเมตร ส่วนความสูงนั้นไม่จำกัด ไมโครเมาส์ต้องทำทุกอย่างด้วยตัวเอง จะต้องไม่มีการช่วยเหลือจากภายนอก โครงสร้างของไมโครเมาส์ต้องไม่ออกแบบให้มีชิ้นส่วนที่จะทำลายเขาวงกต และจะต้องไม่มีส่วนใดส่วนหนึ่งของไมโครเมาส์วางลงบนตัวเขาวงกตขณะที่กำลังเคลื่อนที่ สำหรับการหาบล็อกปลายทางของไมโครเมาส์จะต้องไม่ใช่เกิดจากการเคลื่อนที่ ที่เป็นลักษณะการเดินข้ามกำแพง



รูปที่ 2.2 แสดงลักษณะรูปแบบของไมโครเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2 โครงสร้างสถาปัตยกรรม MCS-51**

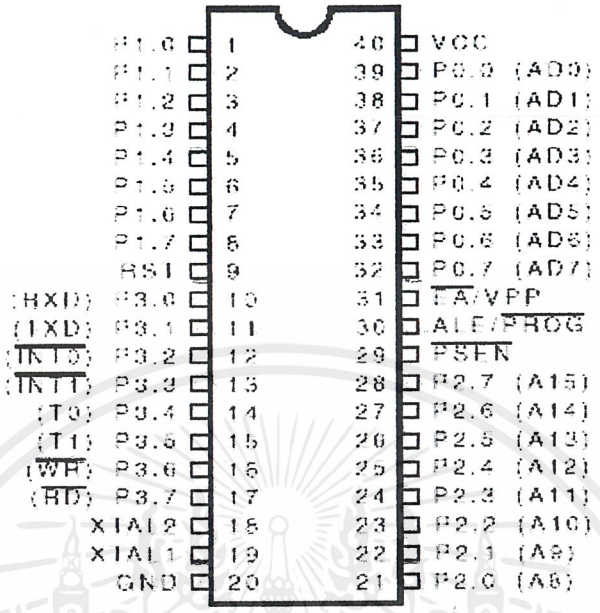
ตระกูล MCS-51 ได้ออกแบบมาเพื่อใช้ในอุตสาหกรรมมาตรฐานขนาด 8 บิต ไมโครคอนโทรลเลอร์ และมีความสามารถในการควบคุม ประยุกต์ใช้งานในเรื่อง Sequential Real Time Control, Close Loop Control และ Data Control

คุณสมบัติของ ไมโครคอนโทรลเลอร์ ตระกูล MCS-51

1. ใช้ HMOS และ CHMOS เทคโนโลยีในการสร้าง และทำงานด้วยแหล่งจ่ายไฟขนาด 5 โวลต์ เพียงแหล่งเดียว
2. CPU มีขนาด 8 บิต
3. มีวงจรออสซิลเลเตอร์ และวงจรนาฬิกาบนชิป
4. ชุดแบงค์ (BANK) เรจิสเตอร์มี 4 ชุด แต่ละชุดมี 8 ตัว
5. มีตัวจับเวลา/ตัวนับ ขนาด 16 บิต 2 ชุด
6. มีพอร์ตไอโอแบบขนานสองทิศทางจำนวน 4 พอร์ต พอร์ตละ 8 บิต
7. พอร์ตแบบอนุกรมสามารถโปรแกรมการรับส่งแบบ Full Duplex ที่ความเร็วสูง
8. หนึ่งวัฏจักรคำสั่งจะกินเวลา 1 ไมโครวินาที ด้วยการใช้คริสทัล 12 เม็กกะเฮิรตซ์
9. แอดเดรสข้อมูลภายนอกได้ 64 กิโลไบต์
10. แอดเดรสโปรแกรมภายนอกได้ 64 กิโลไบต์
11. สามารถกำหนดเลขที่อยู่ข้อมูลขนาดไบต์ หรือ บิต ได้โดยตรง
12. มีซอฟต์แวร์บิตแฟลกสำหรับผู้ใช้ที่กำหนดเองได้ 128 บิต
13. โครงสร้างอินเตอร์รัพต์จะติดตั้งได้ถึง 5 แหล่ง
14. ตัวโพรเซสเซอร์สามารถใช้งานแบบบูลีน (Boolean)ได้สำหรับกระบวนการงานควบคุม
15. มีคำสั่งคูณ และหารทางฮาร์ดแวร์ที่ทำงานได้ภายใน 4 ไมโครวินาที
16. ตัวเลขทางคณิตศาสตร์ ใช้ได้ทั้งระบบไบนารี และเดซิมัล

ตระกูล MCS-51 จะมีทั้งแบบมี ROM ภายในตัว หรือไม่มี ROM หรือมี EPROM บนชิปเดียวกัน และจะมีตำแหน่งขาที่เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.3 การจัดขาลักษณะภายนอกของ MCS-51**

การแบ่งกลุ่มขาตามสถาปัตยกรรมของ MCS-51 สามารถแบ่งได้ 4 กลุ่ม คือ

1. กลุ่มขารับแหล่งจ่ายไฟฟ้า และระบบสัญญาณนาฬิกา
2. กลุ่มขาแอดเดรส และข้อมูล
3. กลุ่มขาควบคุม
4. กลุ่มขาพอร์ตแบบขนาน และอนุกรม

รายละเอียดหน้าที่ของขาแต่ละขา

Vss เป็นขาสำหรับต่อลงดิน

Vcc เป็นขาที่ต่อไฟกระแสตรง

Port 0 เป็นพอร์ตไอโอขนาด 8 บิตแบบ Open Drain Bidirectional สามารถรับโหลด TTL ได้ 8 ตัว การเขียนค่า "1" ไปที่พอร์ตนี้ จะเป็นการปล่อยลอย (Float) ขาของพอร์ตนี้ ทำให้ทำงานเป็นอินพุต มีสถานะอิมพีแดนซ์สูงในการให้พอร์ตนี้ทำงานแบบไอโอ ถ้าทำงานเป็นแบบมัลติเพลกซ์ด้วยสัญญาณแอดเดรสไบต์ต่ำกับบัสข้อมูล สำหรับใช้งานหน่วยความจำภายนอก และยังใช้งานเป็นตัวส่งข้อมูลออกทางพอร์ตนี้เมื่อใช้ตรวจสอบโปรแกรม ROM ภายใน และการโปรแกรมตัว EPROM ภายในแต่ต้องต่อความต้านทานพูลอัพจากภายนอก 10 กิโลโอห์ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Port 1 เป็นพอร์ตไอโอ 8 บิตแบบ Open Drain Bidirectional พร้อมด้วยการพูลอัพจากภายใน ถ้าเป็นพอร์ตเอาต์พุตสามารถรับโหลด TTL ได้ 4 ตัว เมื่อถูกเขียนค่า "1" จะมีสถานะสูงด้วยการพูลอัพจากภายใน จะเป็นการ Initial ใช้งานพอร์ตนี้เป็นอินพุต ขณะที่เป็นอินพุตการให้สัญญาณลงต่ำจะเป็นการจ่ายกระแสออก

Port 2 เป็นพอร์ตไอโอ 8 บิตแบบ Open Drain Bidirectional ด้วยการพูลอัพจากภายใน ทำหน้าที่เป็นบัฟเฟอร์เอาต์พุตสามารถจ่ายโหลด TTL ได้ 4 ตัว และใช้งานเป็นตัวส่งแอดเดรสไบต์สูง การใช้งานแบบนี้จะมีการพูลอัพจากภายในช่วยให้การส่งค่า "1" ได้ระดับที่แน่นอน

Port 3 เป็นพอร์ตไอโอ 8 บิตแบบพูลอัพจากภายในสามารถรับโหลด TTLตระกูล LS ได้ 4 ตัว และมีการทำงานตามฟังก์ชั่นพิเศษ คือ

P3.0 RxD พอร์ตอนุกรมอินพุต

P3.1 TxD พอร์ตอนุกรมเอาต์พุต

P3.2 INT0\ อินเตอร์รัพต์ภายนอกตัวที่1

P3.3 INT1\ อินเตอร์รัพต์ภายนอกตัวที่2

P3.4 T0 สัญญาณกระตุ้นที่ตัวจับเวลา/ตัวนับ0

P3.5 T1 สัญญาณกระตุ้นที่ตัวจับเวลา/ตัวนับ1

P3.6 WR\ สัญญาณควบคุมการเขียน

P3.7 RD\ สัญญาณควบคุมการอ่าน

RST เป็นขารีเซ็ต ต้องคงสถานะค่าสูงเป็นเวลาอย่างน้อยประมาณ 2 วัฏจักรระหว่างที่ออสซิลเลเตอร์ทำงานขณะที่ต้องการรีเซ็ต โดยต่อความต้านทานพูลดาวน์จากขา RST ลงดิน และเพื่อให้ตัวชิปรีเซ็ตได้โดยอัตโนมัติ ขณะเปิดไฟจะใช้คาปาซิเตอร์ต่อคร่อมระหว่างขา RST กับขา Vcc

ALE ,PROG\ เป็นขาแอดเดรสแลทช์อีนาเบิ้ลด้วยการส่งพัลซ์ออกไปสำหรับแลทช์ค่าแอดเดรสไบต์ต่ำจากพอร์ต 0 ในระหว่างการเข้าถึงข้อมูลจากหน่วยความจำภายใน ALE จะส่งสัญญาณนาฬิกาออกมาในอัตราความเร็วคงที่ ที่ 1/8 ของความถี่ออสซิลเลเตอร์ตลอดเวลา จึงสามารถใช้สัญญาณจากขานี้เป็นตัวจับเวลาภายนอกหรือเป็นความถี่สัญญาณนาฬิกา และยังใช้เป็นสัญญาณพัลซ์เข้าสำหรับการควบคุมการโปรแกรม EPROM ภายในชิป

PSEN\ Program Storage Enable เป็นสโตรปอ่านข้อมูลจากโปรแกรมหน่วยความจำภายนอก เมื่อชิปทำงานด้วยโปรแกรมภายนอกขา PSEN จะสร้างสโตรปต่ำสองครั้ง ภายในแต่ละวัฏจักรแมชชีนสัญญาณจะมีสถานะสูง หรือพัลซ์ต่ำทั้งสองลูกจะหายไป เมื่อทำงานในช่วงการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือเขียนข้อมูลจากหน่วยความจำข้อมูลภายนอก และPSEN จะไม่มีพัลซ์ส่งออกถ้าชิปทำงานด้วยโปรแกรมหน่วยความจำภายใน

EA\ ,Vpp ถ้ามีสถานะสูงตัว CPU จะทำงานตามโปรแกรมที่อยู่ในหน่วยความจำภายใน ถ้า EA มีสถานะต่ำจะเป็นการควบคุมให้ CPU ทำงานตามโปรแกรมหน่วยความจำภายนอก

XTAL1 ใช้เป็นตัวอินพุตเข้าสู่ตัวออสซิลเลเตอร์ขยายแบบ Invert

XTAL2 ใช้เป็นตัวเอาต์พุตจากตัวออสซิลเลเตอร์ขยายแบบ Invert

**2.2.1 การจัดการทางสถาปัตยกรรม**

รูปที่ 2.2 เป็นบล็อกไดอะแกรมแบ่งตามลักษณะงานทางสถาปัตยกรรมภายในของ MCS-51 โดยซิงเกิลชิปแต่ละตัวจะประกอบไปด้วยหน่วยศูนย์กลางประมวลผลหน่วยความจำสองชนิด คือ แบบ ROM กับ RAM หรือ EPROM พอร์ตเอาต์พุต อินพุต โหมดเรจิสเตอร์สถานะ และข้อมูล ส่วนวงจรตรรกในการ RANDOM ที่จำเป็นสำหรับตัวแปรของฟังก์ชั่นการต่อพ่วงต่างๆ จะติดต่อกันด้วนบัสข้อมูลขนาด 8 บิต และมีบัฟเฟอร์สำหรับการติดต่อข้อมูลกับภายนอกผ่านพอร์ตไอโอเมื่อต้องการขยายหน่วยความจำ หรือพอร์ตไอโอ

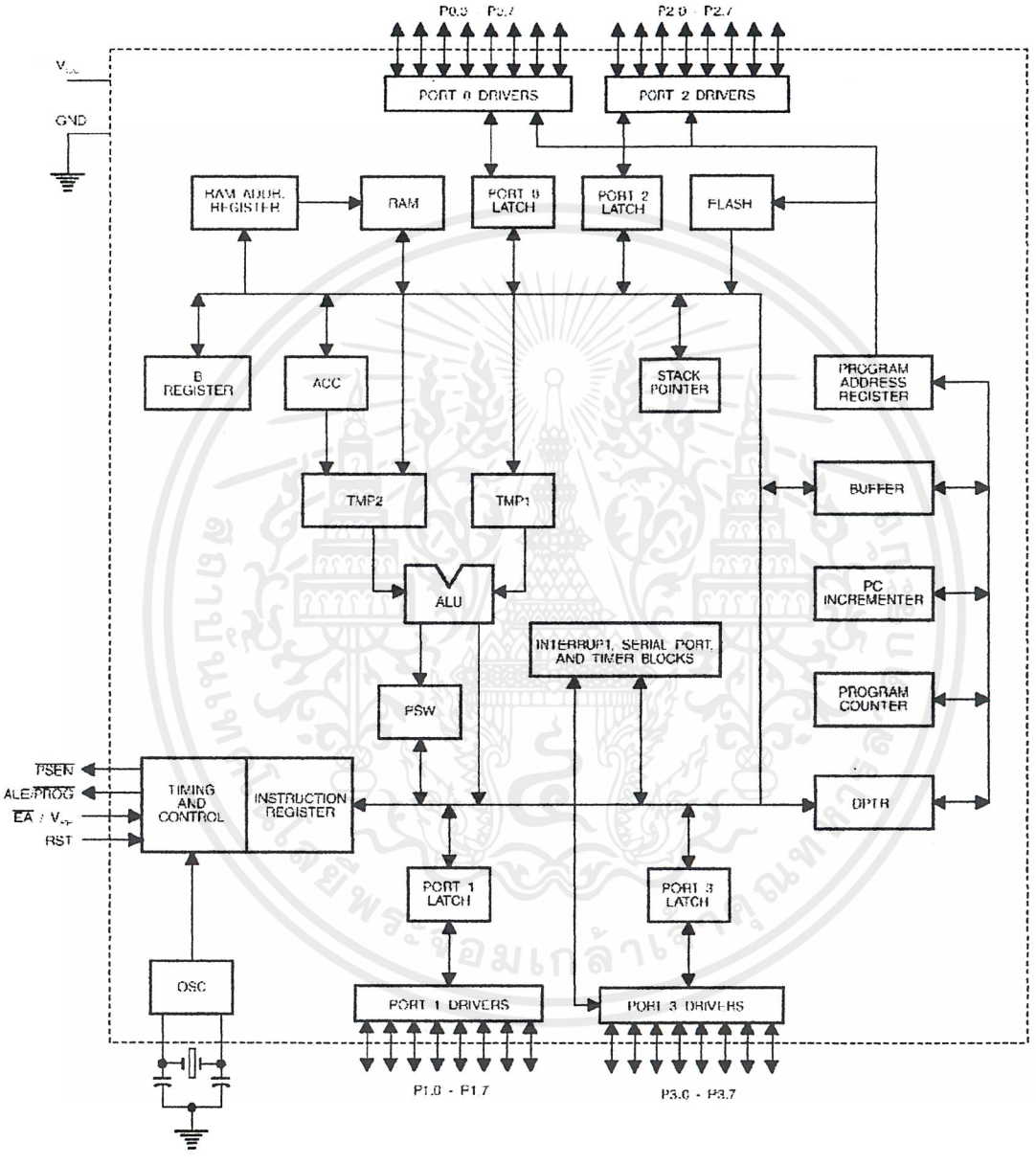
**2.2.2 การเข้าถึงหน่วยความจำภายนอก**

ลักษณะการเข้าถึงหน่วยความจำภายนอกมี 2 แบบ คือการเข้าถึงของหน่วยความจำโปรแกรมภายนอก กับของหน่วยความจำข้อมูลภายนอก การเข้าถึงของหน่วยความจำโปรแกรมภายนอกจะใช้ขาสัญญาณ PSEN (Program Store Enable) แอกทีฟต่ำเป็นสโตปควบคุมการอ่าน และการเข้าถึงของหน่วยความจำข้อมูลภายนอกจะใช้ขา RD หรือ WR แอกทีฟต่ำเป็นสัญญาณสโตปควบคุมหน่วยความจำ

การแฟทช์โปรแกรมภายนอกจะใช้ขาแอดเดรส 16 บิตเสมอส่วนการเข้าถึงของหน่วยความจำข้อมูลสามารถกำหนดเลขที่อยู่ได้ทั้ง 16 บิตแอดเดรส

เมื่อใช้ 16 บิตแอดเดรสไบต์สูงของค่าแอดเดรสจะส่งออกที่พอร์ต 2 และจะคงสถานะค่านั้นตลอดในช่วงวัฏจักรการอ่าน และการเขียน ระหว่างช่วงเวลานี้ตัวแลทช์ของพอร์ต 2 ใน SFR จะไม่ต้องประกอบด้วยค่า “1” และค่าข้อมูลใน SFR จะไม่มีการเซ็ตถ้าช่วงวัฏจักรการใช้หน่วยความจำภายนอกไม่มีการเข้าถึงหน่วยข้อมูลในวัฏจักรต่อมาค่าใน SFR ของพอร์ต 2 จะปรากฏค่าเดิมกลับมาใหม่ในวัฏจักรตัวต่อมาถ้าใช้เป็น 8 บิตแอดเดรสค่าใน SFR ของพอร์ตจะยังคงค่าเดิมที่ขาพอร์ต 2 ตลอดช่วงวัฏจักรการใช้ความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.4 แสดงโครงสร้างสถาปัตยกรรมภายในของ MCS-51**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีใช้แอดเดรสไบต์ต่ำเป็นช่วงเวลามัลติเพลกซ์ กับข้อมูลของพอร์ต 0 ขาสัญญาณแอดเดรสข้อมูล จะขับ FET ทั้ง 2 ตัวในพอร์ต 0 เป็นบัฟเฟอร์ส่งข้อมูลออก ดังนั้นการใช้งานพอร์ต 0 จะไม่มีการรับกระแสเข้าจึงไม่จำเป็นต้องมีการพูลอัพจากภายนอก

สัญญาณ ALE (Address Latch Enable) ก็จะใช้เป็นขาควบคุมรับไบต์แอดเดรสเก็บไว้ภายนอก ซึ่งค่าแอดเดรสจะคงที่ที่ช่วงขอบขาลงของ ALE ดังนั้นในวัฏจักรการเขียนข้อมูลจะถูกเขียนออกไปที่พอร์ต 0 ก่อนที่ WR จะแอกทีฟต่ำส่วนวัฏจักรการอ่านข้อมูลจะรับเข้ามาที่พอร์ต 0 ก่อนสโตปการอ่านจะปรากฏเล็กน้อย และระหว่างการเข้าถึงของหน่วยความจำภายนอก ตัวซีพียูจะส่งค่า 0FFH มาเก็บไว้ที่พอร์ต 0 ของ SFR

การใช้หน่วยความจำโปรแกรมภายนอก จะขึ้นอยู่กับ

1. เมื่อ EA\ แอกทีฟ
2. เมื่อตัวนับโปรแกรม PC ประกอบด้วยตัวเลขที่มีค่ามากกว่า 0FFFH

**2.2.3 สัญญาณ PSEN**

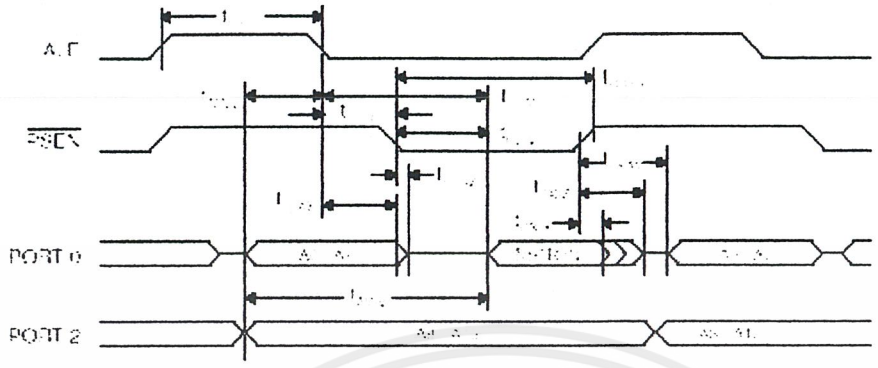
ใช้เป็นตัวควบคุมแฟทช์การอ่านโปรแกรมจากภายนอก PSEN\ จะไม่แอกทีฟถ้ามีการแฟทช์โปรแกรมจากภายในเมื่อซีพียูเข้าถึงการใช้โปรแกรมภายนอก PSEN\ จะแอกทีฟ 2 ครั้งในแต่ละช่วงวัฏจักรการแฟทช์ ยกเว้นคำสั่ง MOVX ช่วงเวลาของการ PSEN\ เกิดแอกทีฟจะไม่เหมือนกับช่วง RD\ แอกทีฟ ช่วงวัฏจักรการอ่านที่สมบูรณ์จะรวมเอาช่วงที่ ALE แอกทีฟ และแอกทีฟซ้ำลูกที่สองกับสัญญาณการควบคุม RD\ ที่เกิดพัลซ์ต่ำ ประกอบเข้าด้วยกัน จะใช้เวลา 12 สถานะคาบเวลา ส่วนช่วงเวลาของ PSEN\ ที่สมบูรณ์จะรวมเอาช่วงที่ ALE แอกทีฟ และแอกทีฟซ้ำลูกที่สองกับสัญญาณควบคุม PSEN\ เข้าด้วยกันจะใช้เวลา 6 สถานะคาบเวลา

**2.2.4 สัญญาณ ALE**

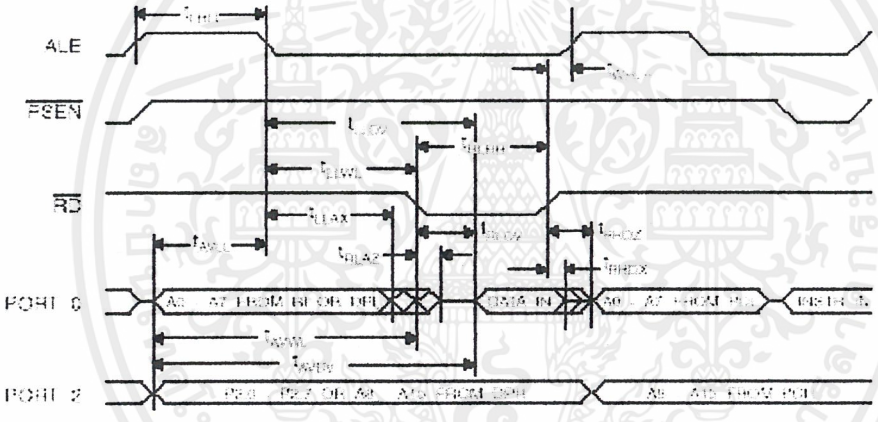
ฟังก์ชั่นหลักของ ALE คือ การใช้งานในการให้จังหวะในการแลทช์เอาไบต์ต่ำ ของแอดเดรสจาก P0 ไปเก็บไว้ภายนอกเพื่อใช้ในการถอดรหัสแอดเดรสโปรแกรมภายนอก โดย ALE จะทำงานแอกทีฟสองครั้งในทุกๆวัฏจักรแมชชีน สัญญาณนี้จะเกิดขึ้นตลอดเวลาแม้จะไม่ได้แฟทช์จากภายนอก มีช่วงเวลาเดียวที่ ALE ไม่เกิดพัลซ์ คือระหว่างการเข้าถึงหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**External Program Memory Read Cycle**




**External Data Memory Read Cycle**




รูปที่ 2.5 แสดงจังหวะการเข้าถึงหน่วยความจำภายนอก

**2.2.5 ตัวจับเวลา/ตัวนับ (Timer/Counter)**

**2.2.5.1 ตัวจับเวลา/ตัวนับ0 และตัวจับเวลา/ตัวนับ1**

แต่ละตัวสามารถทำงานเป็นตัวจับเวลา หรือตัวนับโดยการเซ็ตหรือเคลียร์บิตในเรจิสเตอร์ TMOD ในกลุ่ม SFR ในฟังก์ชั่นตัวจับเวลาตัวเรจิสเตอร์จะเพิ่มค่าทุกๆวัฏจักรแมชชีน ดังนั้นตัวเลขในเรจิสเตอร์จะเป็นจำนวนของวัฏจักรแมชชีน เนื่องจากแต่ละวัฏจักรแมชชีน ประกอบด้วย 12 คาบของออสซิลเลเตอร์ อัตราการนับแต่ละครั้งจะกินเวลา 1/12 ของความถี่ออสซิลเลเตอร์

ในฟังก์ชั่นตัวนับเรจิสเตอร์จะเพิ่มค่าทุกครั้งที่มีการเปลี่ยนแปลงสถานะจาก “1” เป็น “0” ที่เข้ามาที่ขา T0 หรือ T1 ในฟังก์ชั่นนี้สัญญาณภายนอกที่เข้ามาจะถูกแซมปลิ้ง ระหว่างช่วง S5P2 ของทุกวัฏจักรแมชชีน ถ้าแซมปลิ้งสัญญาณเข้ามาเป็นระดับสูงในวัฏจักรหนึ่ง ถ้าในวัฏจักรต่อมาของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเป็นระดับต่ำเรจิสเตอร์จะนับเพิ่มหนึ่งค่า โดยค่าใหม่ของตัวนับ จะปรากฏที่เรจิสเตอร์ช่วง S3P1 ของวัฏจักร ค่าหนึ่งที่รับเข้าไปจะใช้ช่วง 2 วัฏจักรแมชชีนในการรับช่วงการเปลี่ยน “1”เป็น “0” ดังนั้นค่าสูงสุดในการนับจะมีอัตรา 1/24 ของความถี่ออสซิลเลเตอร์ และสัญญาณอินพุตที่นับนั้นจะไม่มีช่วงระยะห่าวที่แน่นอนของ Duty Cycle แต่จะถูกนับ เมื่อระดับแรงดันที่ถูกแซมปลิ้งในแต่ละครั้งต้องมีช่วงคงที่อย่างน้อย 1 วัฏจักรแมชชีนก่อนที่จะเปลี่ยนค่าระดับแรงดันใหม่ในการเลือกทำงานระหว่างตัวนับกับตัวจับเวลา

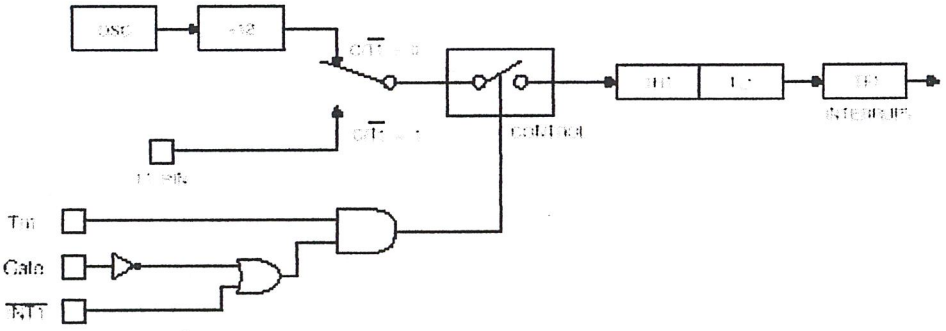
โหมด 0

เรจิสเตอร์ตัวจับเวลาถูกกำหนดให้มี 13 บิตด้วยการนับขึ้นเมื่อเป็น “1” หมดทุกบิตจะกลับมาที่ “0” ทุกบิตใหม่ เมื่อกลับมาเป็น “0” ทุกบิต จะเกิด Overflow ไปทดให้แฟลกอินเตอร์รัฟต์ TF1 ปรับเป็น “1” การควบคุมให้เริ่มนับตัวอินพุตจะควบคุมด้วยการอีนาเบิล TR1=1, GATE=0, INT1\=1 การปรับ GATE=1 เป็นการติดตั้งตัวนับด้วยสัญญาณจากภายนอกที่เข้ามาที่ขา INT1\ ,TR1จะเป็นบิตควบคุมในเรจิสเตอร์ TMOD ของ SFR

เรจิสเตอร์ตัวมี 13 บิต ประกอบด้วย TH1 8 บิต และ TL 5 บิตของอันดับต่ำ การเซ็ตแฟลก TR1.ให้ทำงานจะไม่ได้เคลียร์ค่าในเรจิสเตอร์ของ TH1 และTL1 การทำงานในโหมด 0 ในตัวจับเวลา/ตัวนับ0 จะทำงานเหมือนกับตัวจับเวลา/ตัวนับ 1 โดยใช้ TR0 และINT0\ รวมกันควบคุมแทนสัญญาณต่างๆ

โหมด 1

จะมีการทำงานเหมือนโหมด 0 ต่างกันเฉพาะการใช้เรจิสเตอร์ตัวจับเวลา/ตัวนับมีขนาด 16 บิต โดยไม่มี Prescaler คือ ความถี่ 1/12 ของออสซิลเลเตอร์เป็นความถี่ที่เข้ามาถูกหารด้วย 16 บิตในเรจิสเตอร์ตัวนับ

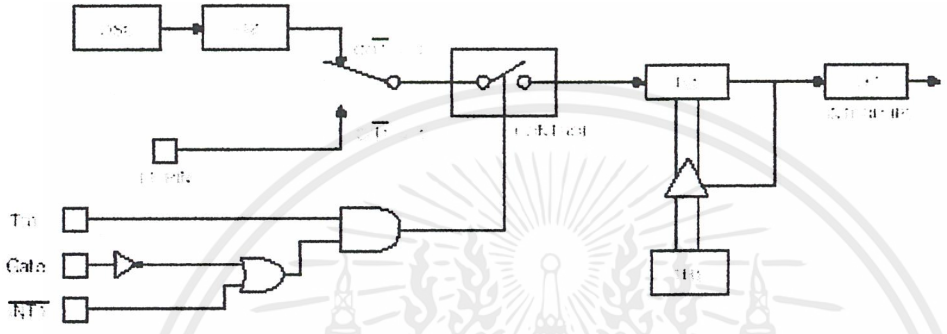


รูปที่ 2.6 แสดงการทำงานในโหมด 1 ของตัวจับเวลาและตัวนับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 2

จะมีการทำงานโดยการกำหนดให้ตัวนับ 8 บิตของ TL1 และจะโหลดใหม่โดยอัตโนมัติทุกครั้งเมื่อมีการ Overflow จาก TL1 ไม่เพียงแต่ TF1 จะปรับเป็น "1" แต่ TL1 จะถูกโหลดโดยอัตโนมัติจากค่าที่ตั้งไว้ใน TH1 ซึ่งค่าใน TH1 สามารถตั้งได้ด้วยซอฟต์แวร์

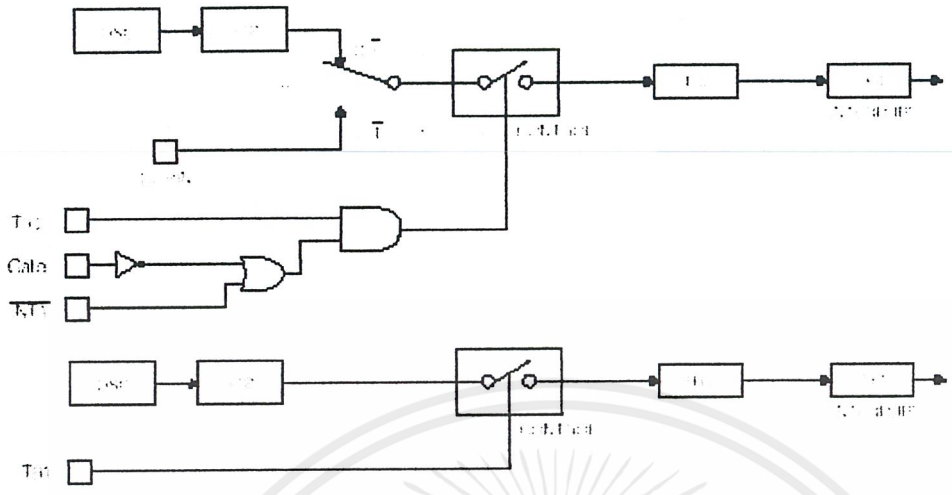



รูปที่ 2.7 แสดงการทำงานในโหมด 2 ของตัวจับเวลา และตัวนับ 1

โหมด 3

ถ้าใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3มีการทำงานเป็นตัวนับ มีผลเช่นเดียวกับการตั้ง TR1=0 และใช้ตัวจับเวลา/ตัวนับ 0 ในโหมด 3 จัดการให้ TL0 และTH0เป็นตัวนับสองตัวแรกที่แยกออกจากกันวงจรตรรก ควบคุมสำหรับโหมด 3ที่ใช้ตัวจับเวลา/ตัวนับ 0 ร่วมกับบิตควบคุมของ C/T, GATE, TR0, INT0\ และ TF0 ตัว TH0 จะถูกล็อกค์ให้ทำงานในฟังก์ชั่นตัวจับเวลา และใช้บิตแฟลก TR1 และ TF1 เข้าร่วมทำงานในโหมด 3 ดังนั้นตัว TH0 จะควบคุมการอินเตอร์รัพต์ของตัวจับเวลา 1 เป็นกลุ่มจับเวลา และขนาด 8 บิต 2 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงการทำงานในโหมด 3 ของตัวจับเวลา และตัวนับ 1

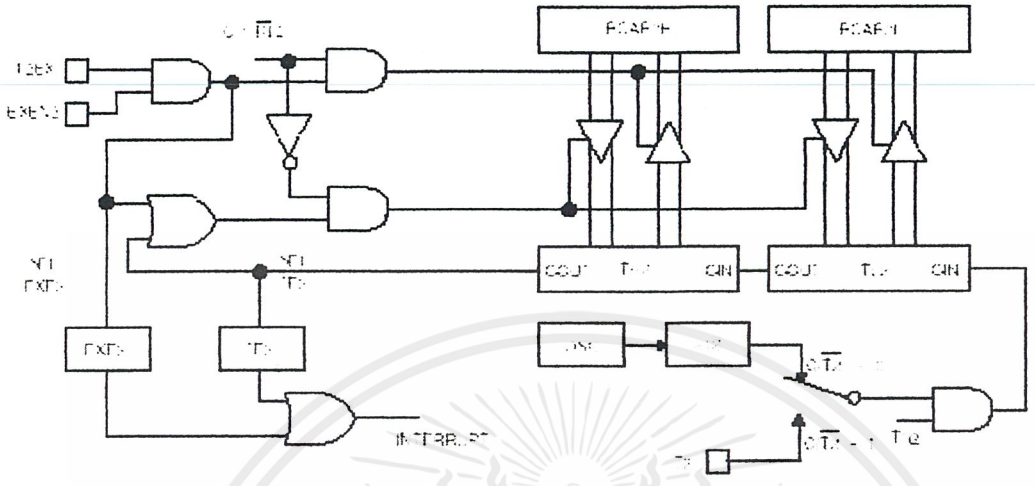
**2.2.5.2 ตัวจับเวลา/ตัวนับ 2**

ตัวจับเวลา/ตัวนับ 2 เป็นตัวจับเวลา และตัวนับขนาด 16 บิตแบบบรรจุเข้าโดยอัตโนมัติ และใช้เรจิสเตอร์ควบคุมของ SFR เป็น T2CON ตัวจับเวลา/ตัวนับ 2 ทำงานเป็นตัวจับเวลาเรจิสเตอร์ตัวจับเวลา/ตัวนับ 2 จะเพิ่มค่าทุกครั้งในแต่ละวัฏจักรแมชชีน ขณะที่เป็นตัวนับ ตัวจับเวลา/ตัวนับ 2 จะเพิ่มค่าขึ้นอยู่กับการเปลี่ยนสถานะจาก "1"ไปเป็น "0" เข้ามาที่ T2 สัญญาณอินพุตถูกแซมปลิ้งที่ S3P2 ของทุกวัฏจักรแมชชีนในการใช้ฟังก์ชั่นให้แอกทีฟ เพราะฉะนั้นเมื่อสัญญาณตัวอย่างแสดงสถานะสูงในช่วงวัฏจักรแมชชีนหนึ่ง การนับจะเพิ่มขึ้นหนึ่ง ค่าตัวใหม่จะปรากฏที่เรจิสเตอร์ตัวจับเวลา/ตัวนับ 2 ช่วง S3P1 ของแต่ละวัฏจักร เป็นการนับหนึ่งที่สัญญาณการเปลี่ยนแปลงถูกกระตุ้น อัตราการนับสูงสุดจะเป็น 1/24 ของออสซิลเลเตอร์

โหมดการทำงานชองตัวจับเวลา/ตัวนับ 2 มี 3 โหมดด้วยกันคือ แบบ CAPTURE แบบธรรมดา แบบบรรจุอัตโนมัติใหม่ และเป็นตัวสร้างอัตราบ๊อด สามารถเลือกการทำงานโดยเซ็ตค่าต่างๆใน T2CON

การใช้โหมด CAPTURE ด้วยการติดตั้ง CP/RL2\=1 จะมีโอกาสเลือกการทำงาน 2 อย่างด้วยการเลือกตั้งค่าที่บิต EXEN2 ของ T2CON โดยตั้ง EXEN2=0 ก็จะเป็นการเลือกตัวจับเวลา/ตัวนับ 2 เป็นตัวจับเวลา หรือตัวนับขนาด 16 บิต ถ้า EXEN2=1 ตัวจับเวลา/ตัวนับจะยังคงทำงานเช่นเดิมเพียงแต่เพิ่มการทำงานการตรวจสอบสัญญาณจากภายนอกที่เข้ามาที่ขา T2EX ในสภาวะที่เกิดการเปลี่ยนแปลงจาก "1" เป็น "0" ทำให้มีการย้ายค่าข้อมูลใน TL2 และ TH2 ในชณะนั้นเข้าสู่เรจิสเตอร์ RCAP2L และ RCAP2H โดยที่เรจิสเตอร์ทั้งสองจัดอยู่ในกลุ่ม SFR การเปลี่ยนแปลงที่สภาวะT2EX จะทำให้บิต T2EX ในเรจิสเตอร์ T2CON เซ็ตค่าเป็น "1" และ EXF2 ก็จะเป็นเช่นเดียวกับบิตTF2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



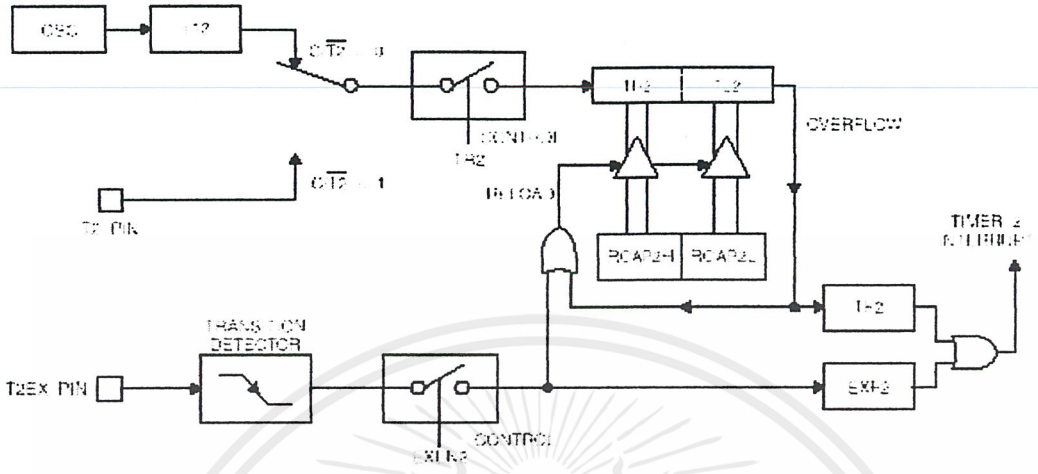
รูปที่ 2.9 แสดงการทำงานในโหมดของตัวจับเวลา/ตัวนับ 2

การใช้โหมดแบบบรรจุอัตโนมัติใหม่ด้วยการติดตั้งให้ CP/RL2\=0 สามารถเลือกการทำงานได้ 2 วิธี ด้วยการติดตั้งค่าในบิต EXEN2 ของ T2CON

1.ถ้าปรับ EXEN2=0 เมื่อตัวจับเวลา 2 เพิ่มค่าถึง 0 ไม่เพียงแต่จะเซ็ตบิต TF2 เป็น 1 เท่านั้น แต่ตัวจับเวลาตัวนับ 2 จะถูกบรรจุค่าที่ตั้งไว้ใน RCAP2L และ RCAP2H ด้วยซอฟต์แวร์แต่แรกกลับเข้าไปที่ TL2 และ TH2 ตามลำดับ

2.ถ้าปรับ EXEN2=1 จะทำงานเช่นเดียวกับวิธีที่ 1 และถ้าเกิดการเปลี่ยนแปลงจาก “1”เป็น “0” ที่ T2EX จากภายนอกจะทำให้มีการบรรจุค่าจาก RCAP2L และ RCAP2Hกลับเข้าไปใหม่โดยอัตโนมัติ พร้อมทั้งเซ็ตค่าบิตใน EXF2 ของเรจิสเตอร์ T2CONด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



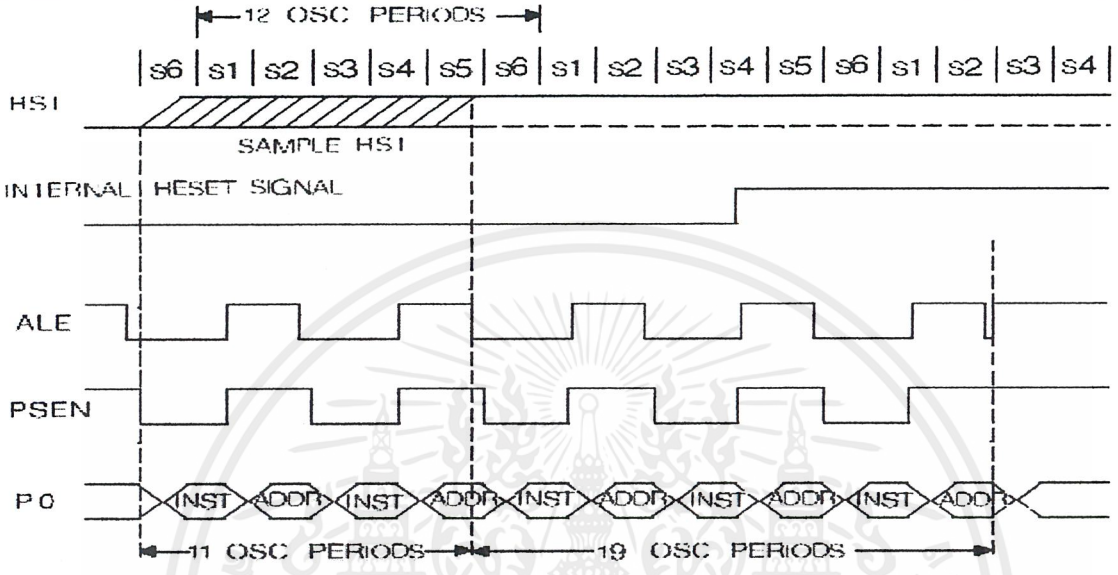

รูปที่ 2.10 แสดงการทำงานในโหมดแบบบรรจุอัตโนมัติของตัวจับเวลา/ตัวนับ 2

ตัวจับเวลา/ตัวนับ 2 สามารถใช้เป็นตัวกำเนิดสัญญาณนาฬิกาให้อัตราบ๊อดได้ในการใช้โหมดรับส่งอนุกรม 1 และ 3 ด้วยการเซ็ต RCLK หรือ TCLK ถ้า RCLK =TCLK=0 เป็นการเลือกตัวจับเวลา/ตัวนับ 1 เป็นตัวกำเนิดอัตราบ๊อด ในโหมดนี้ตัวจับเวลา/ตัวนับ 2 จะให้พัลซ์ของ Overflow มากกว่าที่จะใช้ตัวจับเวลา และ ตัวนับเป็นสัญญาณนาฬิกาจ่ายความถี่ออสซิลเลเตอร์แก่อนุกรมพอร์ต ถ้าแฟลก C/T2\=0 ตัวนับจะเพิ่มหนึ่งด้วยช่วง 1/2 ของความถี่ออสซิลเลเตอร์ ถ้าแฟลก C/T2\=1 ตัวนับจะเพิ่มหนึ่งด้วยการตอบสนองตามสัญญาณเข้าที่เปลี่ยนแปลงจาก “1”เป็น “0” ที่ขา T2EX ความถี่สูงสุดมีค่าเท่ากับ 1/24 * Fosc ถ้า EXEN2 เซ็ตเป็น “1” ขณะที่ตัวจับเวลา/ตัวนับ 2 ถูกใช้เป็นสัญญาณนาฬิกาพอร์ตอนุกรมขาลงที่ T2EX จะเซ็ต EXF2 ถ้าไม่มีการทำงานแบบบรรจุใหม่อัตโนมัติหรือแบบ CAPTURE เกิดขึ้น ปกติที่ตัวจับเวลา/ตัวนับ 2ใช้เป็นโหมดสร้างอัตราบ๊อด T2EX อาจใช้เป็นตัวอินเตอร์รัพต์จากภายนอก TF2 จะไม่มีผลตามตัวนับเวลาเมื่อเกิด Overflow ขณะเดียวกันการบรรจุใหม่โดยอัตโนมัติจาก RCAP2H และ RCAP2Lจะเกิดขึ้นโดยไม่คำนึงถึงสถานะของ CP/RL2\

TH2 และ TL2 ไม่สามารถเขียนหรืออ่านโดยปราศจากข้อผิดพลาด ในขณะที่ตัวจับเวลา และตัวนับ 2 ใช้เป็นโหมดตัวสร้างอัตราบ๊อด และตัว RCAP2H และ RCAP2L จะต้องไม่ถูกโหลดค่าเข้าไประหว่างการใช้โหมดนี้ด้วย ระหว่างการใช้ตัวจับเวลา และตัวนับ 2 เป็นโหมดตัวสร้างอัตราบ๊อด จะไม่สามารถเขียน หรืออ่านข้อมูลใน TH2 และ TL2 เรจิสเตอร์ได้ เช่นเดียวกับตัว RCAP2H และ RCAP2Lก็ต้องไม่ถูกเขียนข้อมูลเข้าไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.6 รีเซ็ต (RESET)




รูปที่ 2.11 แสดงแผนภูมิการรีเซ็ต

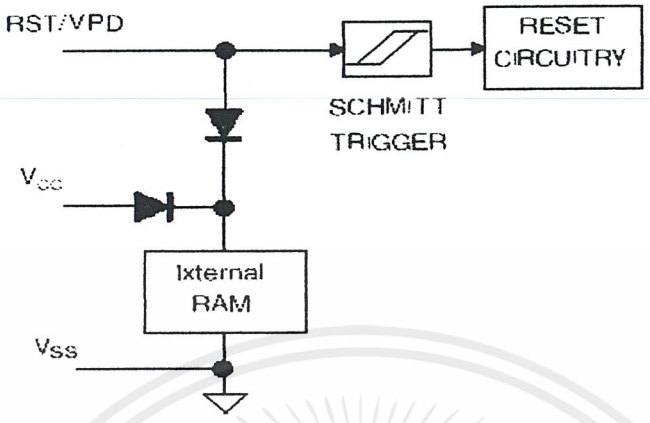
การรีเซ็ตนับเป็นส่วนสำคัญของการที่จะให้ไมโครคอนโทรลเลอร์ทำงานได้อย่างน่าเชื่อถือ เพราะถ้าการให้จังหวะแก่ขารีเซ็ตไม่ถูกต้องจะทำให้ตัวไมโครคอนโทรลเลอร์มีปัญหาดังนั้นจึงควรที่จะทำความเข้าใจกับความต้องการจังหวะเวลาของไมโครคอนโทรลเลอร์แต่ละชนิดอย่าถูกต้องเสียก่อนที่จะออกแบบสร้างวงจรรีเซ็ต

วงจรรีเซ็ตสำหรับรุ่น 8051 ที่เป็น HMOS จะต่อที่ขารีเซ็ตคือขา RST/VPD วงจร Schmitt trigger ถูกใช้เป็นตัว Input สำหรับขจัดสัญญาณรบกวน Noise และที่ Output ของ Schmitt trigger จะถูกแซมปลิ้งเก็บเข้าไปด้วย

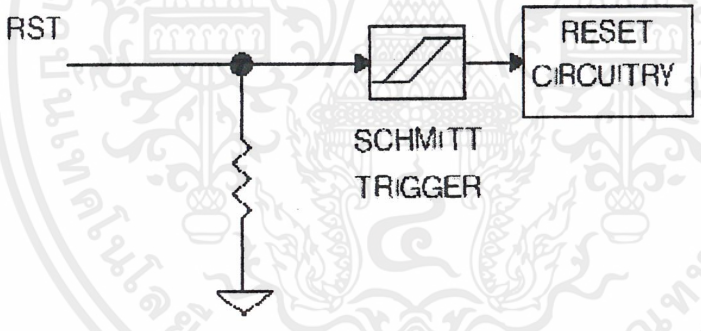
ทั้ง HMOS และCHMOS จะทำงานได้ด้วยการรีเซ็ตสถานะสูงที่ขา RST/VPD เป็นเวลาอย่างน้อย 2 วัฏจักรแมชชีน นับจากที่ออสซิลเลเตอร์เริ่มทำงาน ตัวซีพียูจะตอบสนองด้วยการสร้างสัญญาณรีเซ็ตภายใน สัญญาณรีเซ็ตภายนอกจะเป็นแบบ Asynchronous ต่อสัญญาณนาฬิกาภายในโดยที่สัญญาณรีเซ็ตจะถูกแซมปลิ้งสำหรับทุกวัฏจักรแมชชีน ส่วนขาพอร์ตจะยังคงรักษาสภาวะตามสภาพขณะนั้นเป็นเวลา 19 คาบเวลาของออสซิเลเตอร์ หลังจากสภาวะสูงทางแซมปลิ้ง และใช้เวลาประมาณ 19-31 คาบเวลาออสซิเลเตอร์ หลังจากสัญญาณรีเซ็ตจากภายนอกเข้าขา RST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง


รูปที่ 2.12 แสดงวงจรการจัดการรีเซ็ตของ HMOS



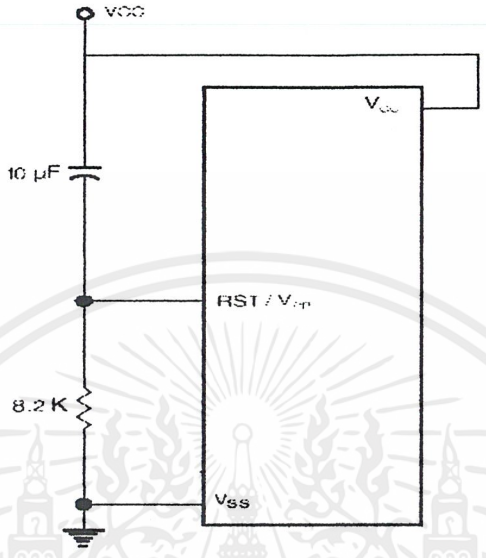

รูปที่ 2.13 วงจรการจัดการรีเซ็ตของ CHMOS

ขณะที่ RST มีสภาวะสูง ALE และ PSEN\ จะถูกดึงให้สถานะสูงแบบอ่อนหลังจาก RST ถูกดึงลงสถานะต่ำมันจะใช้เวลาประมาณ 1 ถึง 2 วัฏจักรแมชชีน สำหรับการเริ่มสัญญาณของ ALE และ PSEN\ ด้วยเหตุผลนี้อุปกรณ์อื่นๆไม่สามารถที่ Synchronous กับช่วงจังหวะภายในของ 8051

ลำดับการทำงานแอลกอริทึมของการรีเซ็ตภายในก็โดยการเขียนค่า "0" ใส่ลงในบิตทุกตัวของ SFR ยกเว้นพอร์ตแลทช์ตัวชี้สแตก และ SBUF ไม่ทราบค่า หลังการรีเซ็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
36820

**2.2.7 Power-on reset**




รูปที่ 2.14 วงจรPower-on reset

การรีเซ็ตอัตโนมัติสามารถจะทำได้เมื่อมีแรงดันไฟที่ Vcc ด้วยการต่อกับขา RST ผ่านตัวคาปาซิเตอร์ขนาด 10 ไมโครฟารัด และต่อตัวต้านทานขนาด 8.2 กิโลโอห์ม ต่อคร่อมระหว่าง Vss กับ RST เพื่อที่จะแบ่งให้ช่วงเวลาขอบขาขึ้นมีไม่เกินมิลลิวินาที และการรีเซ็ตจะเกิดจากช่วงเวลาที่เกิดออสซิลเลตเริ่มต้นไปไม่เกิน 10 มิลลิวินาที โดยวงจรการรีเซ็ตแบบ Power-on เมื่อเปิดไฟกระแสจะไหลผ่านเข้า RST ความต่างศักย์ระหว่าง Vcc กับแรงดันที่ตกคร่อมตัวคาปาซิเตอร์ และจะลดลงจากค่า Vcc เป็นลักษณะชาร์ตเข้า ยิ่งถ้าคาปาซิเตอร์มีขนาดใหญ่ขึ้นแรงดัน Vrst ที่คร่อม RST ก็จะลดช้าลงค่า Vrst จะต้องรักษาระดับให้นานมากกว่าระดับเทสโฮล ของตัวสมิททริกเกอร์ เพื่อที่จะให้มีเวลาพอ ให้การรีเซ็ตเกิดขึ้นอย่างสมบูรณ์เวลาที่ต้องการคือช่วงการเกิดออสซิเลตเริ่มต้นบวกกับอีก 2 วัฏจักรแมชชีนหรืออย่างน้อย 2 sec 24 คาบวงจรออสซิเลเตอร์ซึ่งมีค่าอย่างน้อย 2 ไมโครวินาทีที่ค่าความถี่ 12 MHz ออสซิเลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| เรจิสเตอร์ | มีค่าข้อมูลเป็น | เรจิสเตอร์ | มีค่าข้อมูลเป็น |
|---|---|---|---|
| PC | 0000H | T2CON | 00H |
| ACC | 00H | TH0 | 00H |
| B | 00H | TL0 | 00H |
| PSW | 00H | TH1 | 00H |
| SP | 07H | TL1 | 00H |
| DPTR | 0000H | TH2 | 00H |
| P0-P3 | 0FFH | TL2 | 00H |
| IP[8051] | XXX00000B | RLDH | 00H |
| IP[8052] | XX000000B | RLDH | 00H |
| IE[8051] | 0XX00000B | SCAN | 00H |
| IE[8052] | 0X000000B | SBUF | Indeterminate |
| TMOD | 00H | PCON[HMOS] | 0XXXXXXXB |
| TCON | 00H | PCON[CMOS] | 0XXX0000B |

ตารางที่ 2.1 แสดงค่าต่างๆในเรจิสเตอร์หลังการรีเซ็ต

**2.2.8 การจัดการหน่วยความจำ**

MCS-51 แบ่งตามพื้นฐานหน่วยความจำของการกำหนดเลขที่อยู่แอดเดรสได้ 3 ส่วนประกอบด้วยเนื้อที่

1. 64 กิโลไบต์หน่วยความจำโปรแกรม
2. 64 กิโลไบต์หน่วยความจำข้อมูลภายนอก
3. 256 ไบต์เป็นหน่วยความจำข้อมูลภายใน

เนื้อที่หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมประกอบด้วย ส่วนภายใน และภายนอกชิป ถ้าขา EA มีสถานะสูง MCS-51 จะทำโปรแกรมภายใน โดยโปรแกรมต้องมีความยาวไม่เกิน 0FFF(4K) และจะแฟลทช์โปรแกรมภายนอกตำแหน่ง 1000H ถึง 0FFFH และถ้าขา EA มีสถานะต่ำจะแฟลทช์ข้อมูลโปรแกรมภายนอก ตัวนับโปรแกรมขนาด 16 บิต จะเป็นตัวกำหนดเลขที่อยู่โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื้อที่หน่วยความจำข้อมูล

หน่วยความจำข้อมูลประกอบด้วยหน่วยความจำข้อมูลภายใน และหน่วยความจำข้อมูลภายนอก หน่วยความจำข้อมูลภายนอกสามารถเข้าถึงด้วยคำสั่ง MOVX

หน่วยความจำข้อมูลภายในสามารถแบ่งตามลักษณะงาน คือ จำนวน 128 ไบต์ของบริเวณตำแหน่งล่างในเนื้อที่แรมภายใน และส่วนของ 128 ไบต์อีกบริเวณหนึ่งใช้เป็นเรจิสเตอร์ฟังก์ชั่นพิเศษ ขณะที่ใช้ส่วนบนของแรมภายใน และบริเวณของ SFR ทั้งสองส่วนนี้จะถูกปันส่วนให้ใช้ค่าแอดเดรสภายใน แต่จะเข้าถึงในแต่ละบริเวณของทั้งสองนี้ได้ ด้วยการกำหนดโหมดเลขที่อยู่ต่างกัน

หน่วยความจำข้อมูล แบ่งเป็น 4 แบงค์ในแต่ละแบงค์มีเรจิสเตอร์ 8 ตัวมีตำแหน่งตั้งแต่ 0 ถึง 31 ในบริเวณล่างของแรม แบงค์เหล่านี้สามารถใช้ได้คราวละ 1 แบงค์เท่านั้น ด้วยการกำหนดเริ่มแรกภายใน 2 บิตของเรจิสเตอร์ PSW ว่าจะเลือกใช้แบงค์ใด และบริเวณตำแหน่งตั้งแต่ 20H ถึง2FH จำนวน 16 ตำแหน่งๆ ละ 1ไบต์สามารถที่จะกำหนดเลขที่อยู่ของแต่ละบิตได้

การเข้าถึงของข้อมูลในรูปของคำสั่งไมโครคอนโทรลเลอร์ แบ่งเป็นลักษณะงานได้ดังนี้

1. การกำหนดตำแหน่งที่อยู่โดยเรจิสเตอร์
2. การกำหนดตำแหน่งที่อยู่โดยตรง
3. การกำหนดตำแหน่งที่อยู่โดยอ้อมด้วยเรจิสเตอร์
4. การกำหนดตำแหน่งที่อยู่โดยทันที
5. การกำหนดตำแหน่งที่อยู่โดยอ้อมด้วยเรจิสเตอร์เป็นฐานร่วมกับค่าเรจิสเตอร์ดรรชนี

## 2.3 8255

การขยายเพิ่มเติมพอร์ตอินพุต และเอาต์พุตของ 8051 นอกจากการใช้ไอซีประเภทแลตช์ และ บัฟเฟอร์ประกอบกันเข้ากับบัสของระบบแล้ว ยังสามารถใช้ไอซีความเร็วสูง (LSI) เบอร์ 8255 ซึ่งสามารถทำหน้าที่ได้ทั้งพอร์ตอินพุต หรือเอาต์พุตตามการโปรแกรมด้วยซอฟแวร์ ทำให้มีความอ่อนตัวในการนำไปใช้งานตามความประสงค์มากขึ้น ภายในบทนี้จะได้กล่าวแนะนำถึงลักษณะพื้นฐานของไอซีนี้ รวมถึงการโปรแกรมใช้งาน และการเชื่อมต่อเข้ากับบัสของ 8051 ซึ่งจะทำให้สามารถนำไปประยุกต์ใช้ได้อย่างเหมาะสมต่อไป

### 2.3.1 ลักษณะพื้นฐานของ 8255

ไอซีเบอร์ 8255 ได้รับการออกแบบเพื่อทำหน้าที่เป็นพอร์ต สำหรับการรับส่งข้อมูลแบบขนานระหว่างอุปกรณ์ภายนอกกับไมโครคอนโทรลเลอร์ ความอ่อนตัวในการนำไปใช้งานของ 8255 ได้แก่การที่สามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ต ให้เป็นการเอาต์พุท หรืออินพุทได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สะดวก เพียงแค่การส่งข้อมูลควบคุมจากไมโครคอนโทรลเลอร์ก่อนที่จะเริ่มต้นใช้งานเท่านั้นความสามารถเช่นนี้ เรียกว่า Programmable คือ สามารถโปรแกรมการทำงานได้ ทำให้ได้รับความนิยมนำไปใช้งานอย่างกว้างขวาง

### 2.3.2 การจำแนกกลุ่มของพอร์ต 8255

ในบรรดาพอร์ตทั้งสามของ 8255 คือ พอร์ต A พอร์ตB และพอร์ต C โดยพื้นฐานนั้น ล้วนเป็นพอร์ตขนานที่ประกอบด้วยสัญญาณ 8 เส้น ซึ่งแต่ละเส้นจะแทนบิตของข้อมูลพอร์ต ซึ่งอาจจะกล่าวในอีกลักษณะหนึ่งว่าเป็นพอร์ตแบบ 8 บิต นอกจากนี้ยังสามารถอ้างถึงแต่ละบิตของเส้นสัญญาณพอร์ตนี้ได้โดยอิสระ อย่างไรก็ตาม 8255 ได้จัดกลุ่มของพอร์ตเหล่านี้ออกเป็นสองกลุ่ม (Group) คือ Group A และ Group B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ต

| ชื่อกลุ่ม | ลักษณะ |
|---|---|
| Group A | พอร์ต A จำนวน 8 บิต (ทุกบิตของพอร์ต)<br>พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตบนของพอร์ต) |
| Group B | พอร์ต B จำนวน 8 บิต (ทุกบิตของพอร์ต)<br>พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตล่างของพอร์ต) |

ตารางที่ 2.2 แสดงการกำหนดรูปแบบการทำงานของพอร์ต

จากตารางการทำงานข้างต้นจะเห็นว่า จำนวนเส้นสัญญาณทั้งหมดของพอร์ต C (PC0-PC7) ได้ถูกแยกออกเป็นกลุ่ม คือ กลุ่มของ 4 บิตล่าง (Lower nibble) จาก PC0-PC3 และกลุ่มของ 4 บิต บน (Upper nibble) จาก PC4-PC7 ดังนั้น Group A และGroup B ของ 8255 จึงมีจำนวนบิตในแต่ละกลุ่มเป็นจำนวนถึง 12 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สัญญาณ | ความหมาย |
|---|---|
| D0-D7 | กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียน หรือ อ่าน |
| CS\ | สัญญาณเลือกอุปกรณ์ เมื่อขาสัญญาณนี้เป็นระดับลอจิกต่ำ ซีพียูก็สามารถเขียน หรือ อ่านข้อมูลจาก 8255 ได้ |
| RD\ | สัญญาณบอกสถานะต้องการอ่านข้อมูลจากเรจิสเตอร์ของ 8255 |
| WR\ | สัญญาณบอกสถานะต้องการเขียนข้อมูลจากเรจิสเตอร์ของ 8255 |
| A0-A1 | สัญญาณระบุตำแหน่งเรจิสเตอร์ภายใน 8255 ที่ต้องการ |
| RESET | สัญญาณการรีเซ็ตวงจรทำงานภายในเรจิสเตอร์ของ 8255 |
| PA0-PA7 | กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต A ของ 8255 |
| PB0-PB7 | กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต B ของ 8255 |
| PC0-PC7 | กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต C ของ 8255 |

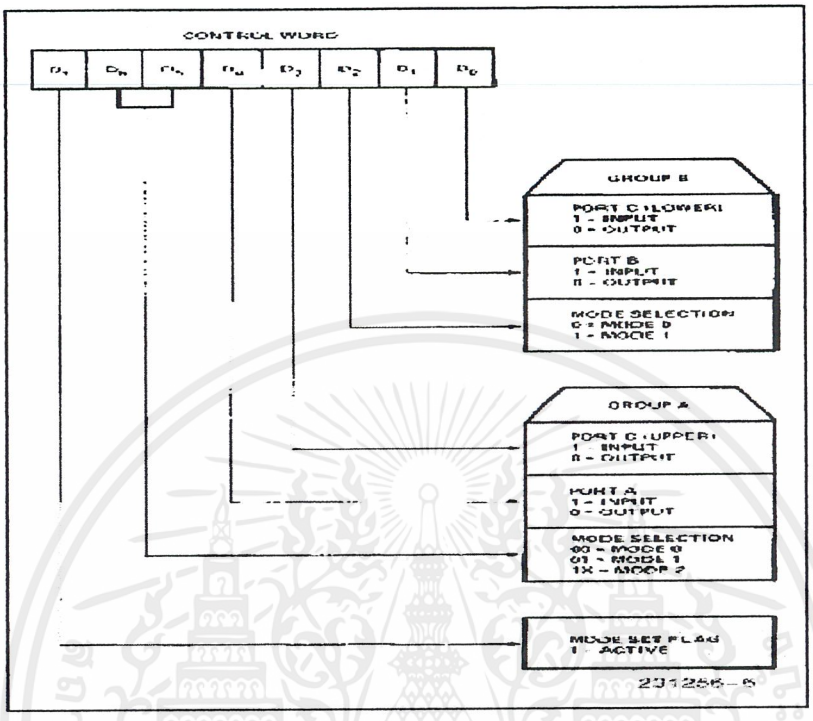
ตารางที่ 2.3 แสดงความหมายของเส้นสัญญาณต่างๆ

**2.3.3 รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255**

การกำหนดให้พอร์ตทั้งสามของ 8255 ทำงานในลักษณะต่างๆ กัน หรือที่เรียกว่า โหมดการทำงาน (Mode) จะเริ่มด้วยการส่งค่าของข้อมูลไบต์หนึ่งให้กับเรจิสเตอร์ควบคุมการทำงานภายใน 8255 ข้อมูลนี้จะเรียกว่า ไบต์ข้อมูลควบคุม (Control word) โดยแต่ละบิตของข้อมูลนี้จะมีความหมายที่ระบุถึงความต้องการต่างๆ ไปแสดงในรูปที่ 2.15 การส่งข้อมูลไบต์นี้จะต้องเริ่มต้นเป็นลำดับแรกก่อนที่จะได้มีการดำเนินการใดๆ กับ 8255 ทั้งสิ้น

ตามความหมายของบิตภายในตารางที่ 2.6 จะเห็นว่าการเลือกให้พอร์ตใดพอร์ตหนึ่งทำหน้าที่เป็นอินพุตก็เพียงแต่กำหนดค่าข้อมูล 1 ให้กับบิตที่เกี่ยวข้องกับพอร์ตนั้น หรือกรณีตรงข้ามสำหรับการเอาต์พุตก็เพียงกำหนดค่าข้อมูล 0 เท่านั้น อย่างไรก็ตามการกำหนดให้ไบต์ข้อมูลควบคุมนี้มีผลอย่างถูกต้อง ก็จะต้องทำการกำหนดให้บิต D7 มีค่าเป็น 1 เสมอ สำหรับบิตที่บอกถึงโหมดการทำงาน (บิต D6 D5 และD2) นั้นจะกล่าวถึงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.15 แสดงการส่งค่าของข้อมูลควบคุมการทำงานภายใน 8255

**2.3.4 การเชื่อมต่อ 8255 กับ 8051**

เมื่อพิจารณาภาพของ 8255 จะเห็นว่ามีขาสัญญาณแอดเดรสจำนวน 2 เส้น คือ A0 และ A1 ทำให้ตำแหน่งของแอดเดรสที่จะอ้างได้มีค่าเป็น 4 ตำแหน่ง ซึ่งแต่ละตำแหน่งจะมีความหมายถึงการระบุเรจิสเตอร์ หรือพอร์ตภายใน 8255

| A1 | A0 | ชื่อของเรจิสเตอร์ |
|---|---|---|
| 0 | 0 | พอร์ต A |
| 0 | 1 | พอร์ต B |
| 1 | 0 | พอร์ต C |
| 1 | 1 | เรจิสเตอร์ควบคุม |

ตารางที่ 2.4 แสดงการระบุเรจิสเตอร์ หรือพอร์ตภายใน 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อพิจารณาค่าของแอดเดรสเหล่านี้ร่วมกับระดับลอจิก ของขาสัญญาณ RD\ และ WR\ จะเป็นการอ่านหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับเรจิสเตอร์นั้นตามลำดับ

| RD\ | WR\ | A1 | A0 | ความหมาย |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | เขียนข้อมูลให้กับพอร์ต A |
| 1 | 0 | 0 | 0 | อ่านข้อมูลจากกับพอร์ต A |
| 0 | 1 | 0 | 1 | เขียนข้อมูลให้กับพอร์ต B |
| 1 | 0 | 0 | 1 | อ่านข้อมูลจากกับพอร์ต B |
| 0 | 1 | 1 | 0 | เขียนข้อมูลให้กับพอร์ต C |
| 1 | 0 | 1 | 0 | อ่านข้อมูลจากกับพอร์ต C |
| 0 | 1 | 1 | 1 | เขียนข้อมูลให้กับเรจิสเตอร์ควบคุม |
| 1 | 0 | 1 | 1 | เป็นสภาวะที่ไม่ถูกต้อง |

ตารางที่ 2.5 แสดงการอ่านหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับเรจิสเตอร์

ดังนั้นโดยทั่วไปจึงมักจะกำหนดให้แอดเดรสของ 8255 ทั้งสี่ตำแหน่งนี้ อยู่ในแอดเดรสช่วงใดช่วงหนึ่งของระบบ เช่น 10H, 11H, 12H, 13H โดยขาสัญญาณแอดเดรส ที่นอกเหนือไปจาก A0 และ A1 นำมาเข้ายังตัวถอดรหัสแอดเดรส เพื่อสร้างสัญญาณเลือกอุปกรณ์ (CS\) ในช่วงที่ต้องการ

ขาสัญญาณควบคุมอื่นๆ คือ RD\ และ WR\ มักจะเชื่อมต่อเข้ากับขาสัญญาณชื่อเดียวกันของ 8051 ได้โดยตรง ทำให้แอดเดรสของ 8255 อยู่ในพื้นที่ของหน่วยความจำข้อมูลของ 8051 ดังที่ได้อธิบายผ่านมาแล้ว สำหรับขาสัญญาณ RESET ของ 8255 ซึ่งจะมีผลทำให้เกิดการรีเซ็ต หรือเริ่มสภาวะการทำงานใหม่เมื่อระดับของขาสัญญาณเป็นลอจิกสูง ดังนั้นหากว่าจะใช้สัญญาณการรีเซ็ตเดียวกับของ 8051 เพื่อที่จะรีเซ็ต 8255 ด้วยก็สามารถต่อได้โดยตรง

ส่วนขาสัญญาณ D0-D7 ก็สามารถนำไปเชื่อมต่อโดยตรงเข้ากับบัสของ 8051 ได้เช่นกันโดยสมมติว่าไม่จำเป็นต้องมีการใช้วงจร หรือไอซีบัฟเฟอร์ขับสัญญาณบัสข้อมูล

การทำงานโหมด 0 ของ 8255

เมื่อ 8255 ได้รับการกำหนดให้ทำงานในโหมดนี้ จะทำให้พอร์ตต่างๆ มีหน้าที่เป็นพอร์ตอินพุต หรือเอาต์พุต ได้เพียงลักษณะเดียวเท่านั้น การเริ่มต้นจะทำโดยการส่งไบต์ข้อมูลควบคุมให้

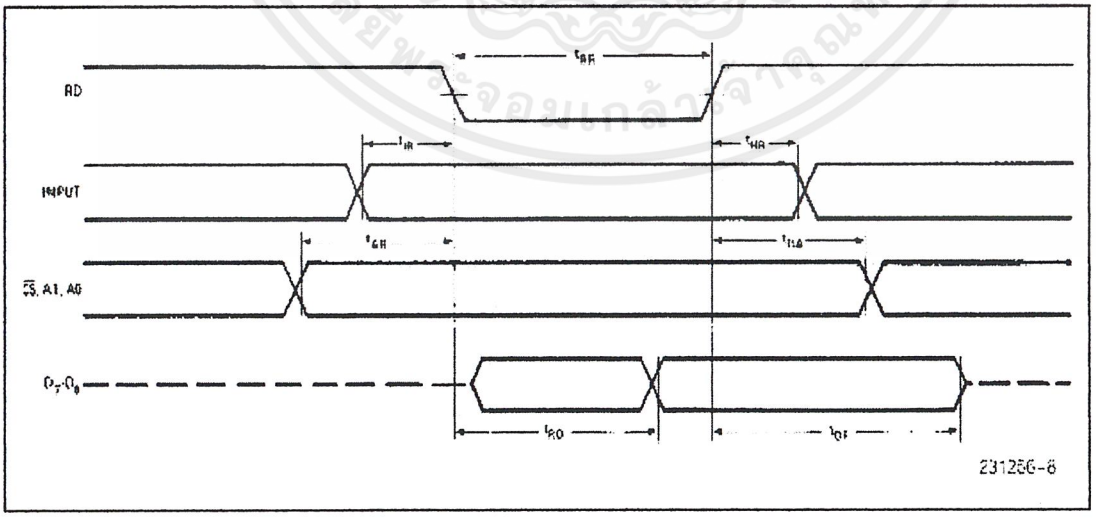
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับเรจิสเตอร์ควบคุม ต่อไปจะแสดงให้เห็นถึงรูปแบบการกำหนดบิต เมื่อต้องการให้พอร์ต A, B และ C ทำหน้าที่เป็นพอร์ตเอาต์พุตทั้งหมด

| ตำแหน่งบิต | ค่าข้อมูล | ความหมาย |
|---|---|---|
| D7 | 1 | ระบุให้ทราบว่าเป็นไบต์ข้อมูลควบคุม |
| D6 และD5 | 00 | กำหนดโหมดการทำงานให้กับพอร์ต A เป็นโหมด 0 |
| D4 | 0 | ระบุพอร์ต A เป็นการเอาต์พุตข้อมูล |
| D3 | 0 | กำหนดให้เส้นสัญญาณสี่บิตบนของพอร์ต C เป็นการเอาต์พุตข้อมูล |
| D0 | 0 | กำหนดโหมดการทำงานให้กับพอร์ต B เป็นโหมด 0 |
| D1 | 0 | ระบุพอร์ต B เป็นการเอาต์พุตข้อมูล |
| D2 | 0 | กำหนดให้เส้นสัญญาณสี่บิตล่างของพอร์ต C เป็นการเอาต์พุตข้อมูล |

ตารางที่ 2.6 แสดงรูปแบบการกำหนดบิต

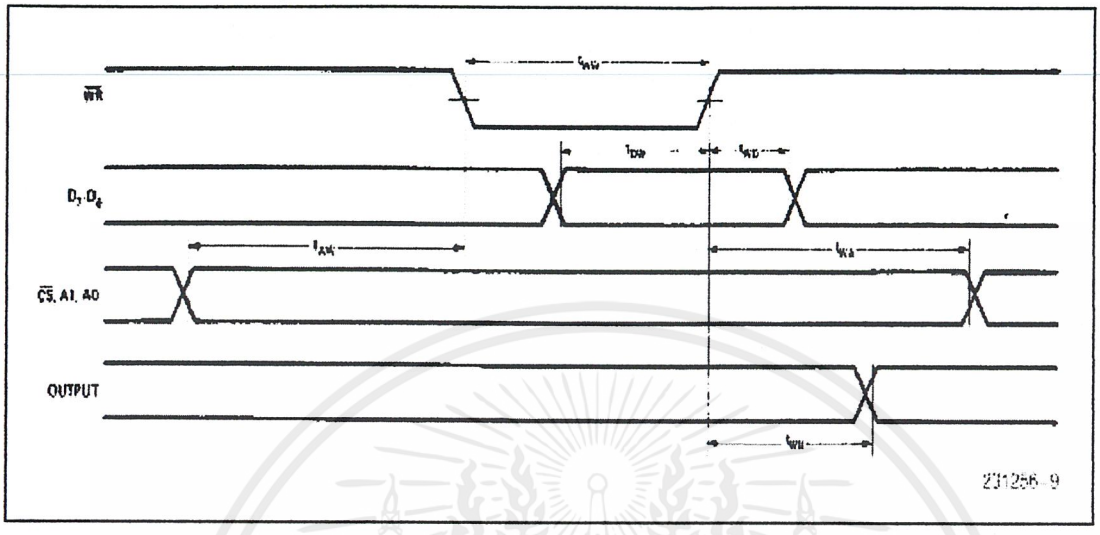
MODE 0 (BASIC INPUT)




รูปที่ 2.16 แสดงการทำงานของ 8255 ใน MODE 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 (BASIC OUTPUT)




รูปที่ 2.17 แสดงการทำงานของ 8255 ใน MODE 0 (ต่อ)

**2.3.5 การเชื่อมต่อ MCS-51 กับหน่วยความจำ**

แบ่งได้เป็น 2 ส่วนใหญ่ๆ คือ

- Data Memory
- Program Memory

DATA MEMORY

เป็นหน่วยความจำที่ใช้เก็บข้อมูล มีความจุได้ถึง 64 Kb เป็นหน่วยความจำประเภทอ่าน และเขียนได้ (RAM) สัญญาณจาก 8051 ที่ใช้ในการอ่านข้อมูลจาก RAM คือ RD\ และสัญญาณที่ใช้ในการเขียนข้อมูลลง RAM คือ WR\

RAM (Random Access Memory)

เป็นหน่วยความจำที่สามารถอ่าน และเขียนข้อมูลได้ ข้อมูลจะสูญหายทันทีที่ขาดไฟเลี้ยง แบ่งเป็น 2 ชนิด คือ Static RAM และ Dynamic RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Static Ram

เป็นหน่วยความจำชนิดหนึ่งที่ง่ายในการนำมาใช้งาน แต่ราคาจะสูงเมื่อเทียบกับ Dynamic Ram ซึ่งราคาถูกกว่า แต่จะยุ่งยากขึ้นมาก็ คือ ต้องทำการ Refresh หน่วยความจำตลอดเวลาส่วน Static Ram นั้นจะใช้ลักษณะของวงจร ฟลิปฟลอป เป็นพื้นฐาน ดังนั้นเมื่อไม่มีการเขียนข้อมูลใหม่ข้อมูลนั้นๆ จะคงที่ตลอดไปเท่าที่มีไฟเลี้ยงอยู่

Dynamic Ram

เราจะเรียกสั้นๆว่า DRAM ลักษณะของ RAM ชนิดนี้ ใช้การเก็บประจุที่ขา เกตของ MOSFET เพื่อเป็นการเก็บข้อมูล และเมื่อมีการอ่านข้อมูลออกมา ประจุที่เก็บไว้จะถูกคายออกมา และหมดไป ซึ่งถ้าเป็นเช่นนี้ หน่วยความจำจะเป็นแบบ Destructive ดังนั้น DRAM จึงต้องมีการรีเฟรช เพื่อทำให้ข้อมูลยังคงอยู่ไม่สูญหายไปไหน ข้อดีของ DRAM ที่เหนือกว่า SRAM คือ มีความหนาแน่นของเซลล์หน่วยความจำสูงกว่า มีความสิ้นเปลืองกำลังต่ำกว่า และใช้เวลาในการเข้าถึงข้อมูลเร็วกว่า แต่มีข้อเสียในเรื่องการที่จะต้องทำการรีเฟรช ซึ่งจะยุ่งยากในการใช้มากกว่า SRAM

ศัพท์ที่ควรทราบเกี่ยวกับหน่วยความจำ

| | |
|---|---|
| Access time | ระยะเวลาจากที่กำหนดสัญญาณระบุตำแหน่งให้กับหน่วยความจำ ไปจนถึงเวลาที่ข้อมูลปรากฏออกมาที่เอาต์พุต ของไอซีหน่วยความจำ ระยะเวลานี้รวมถึงเวลาที่เสียไปตั้งแต่ส่งสัญญาณเลือกตัวไอซี (chip enable) การถอดรหัสของตำแหน่ง และการทำงานของวงจร output enable |
| Address hold time | คือ เวลาที่ต้องการนับตั้งแต่จุดสิ้นสุดของสัญญาณ write จนถึงเวลาที่เอาสัญญาณในการกำหนดตำแหน่งออกไปได้โดยข้อมูลไม่สูญหาย |
| Address setup time | คือ ระยะเวลาที่ต้องการเพื่อให้สัญญาณในการกำหนดตำแหน่งให้อยู่ในสภาวะคงที่ก่อนที่จะมีการเขียนให้แก่หน่วยความจำ เพื่อการเขียนข้อมูล |
| Chip enable (CE\) | คือ สัญญาณที่ใช้กำหนดให้ไอซีหน่วยความจำนี้ทำงานหรือไม่ซึ่งเมื่อสัญญาณนี้แอคตีฟ (โดยปกติแอคตีฟที่ระดับ “0”) ไอซีหน่วยความจำนี้จึงจะอยู่ในสภาวะที่ทำงานได้ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output enable (OE\) คือ สัญญาณที่ป้อนให้แก่ไอซี ซึ่งเมื่ออยู่ในสภาวะแอคตีฟ จะทำให้เอาต์พุตของไอซีหน่วยความจำนี้ติดต่อกับบัสข้อมูลของระบบ แต่เมื่อสัญญาณนี้ไม่แอคตีฟ เอาต์พุตของไอซีจะอยู่ในสภาวะอิมพีแดนซ์สูง

PROGRAM MEMORY

เป็นหน่วยความจำที่ใช้เก็บโปรแกรมการทำงาน เป็นหน่วยความจำที่อ่านข้อมูลได้อย่างเดียว (ROM) สัญญาณจาก 8051 ที่ใช้ในการอ่าน ROM คือ PSEN\

ROM (Read Only Memory)

เป็นหน่วยความจำแบบถาวร เมื่อขาดไฟเลี้ยงข้อมูล จะไม่สูญหาย เป็นหน่วยความจำที่อ่านข้อมูลได้อย่างเดียว คือข้อมูลจะต้องถูกเขียนลงไปในหน่วยความจำตั้งแต่ต้น หลังจากนั้นก็เป็นการเรียกใช้เพียงอย่างเดียว ไม่สามารถเปลี่ยนแปลงแก้ไขได้อีก ดังนั้น เราจึงเรียกหน่วยความจำประเภทนี้ว่า Read Only หน่วยความจำแบบรอม เป็นหน่วยความจำแบบ Non volatile คือข้อมูลในรอมจะไม่สูญหายเมื่อไม่มีไฟเลี้ยง หน่วยความจำแบบนี้ จะใช้ในระบบไมโครคอมพิวเตอร์ทุกเครื่อง เพื่อเก็บโปรแกรมสำหรับให้ระบบเริ่มต้นทำงานได้ หรือเรียกว่า โปรแกรมมอนิเตอร์ หรือ bootstrap ROM สามารถจำแนกตามลักษณะการเขียนข้อมูลได้ดังนี้

Mask Programmed ROM

เป็นรอมชนิดที่ได้ทำการเขียนข้อมูลเข้าไปในหน่วยความจำตั้งแต่ตอนสร้างชิพ และไม่สามารถแก้ไขข้อมูลภายในได้อีก รอมแบบนี้จะต้องสร้างคราวละมากๆ เนื่องจากต้นทุนในการผลิตสูง ในตระกูล MCS-51 ก็จะเป็นเบอร์ 8051 และ 8052

Programmable Read Only Memory หรือ PROM

หน่วยความจำแบบนี้ ผู้ใช้สามารถโปรแกรมเองได้ โดยกรรมวิธีจ่ายพัลซ์แรงดันสูง เข้าไปทำลายฟิวส์ ภายในตัวไอซีหน่วยความจำ เพื่อทำให้เป็นลอจิก 0 หรือ 1 ณ ตำแหน่งที่กำหนด และเมื่อโปรแกรมไปแล้ว ก็ไม่สามารถทำการแก้ไขได้อีกเลย

Erasable Programmable Read Only Memory หรือ EPROM

หน่วยความจำชนิดนี้ ผู้ใช้สามารถโปรแกรมข้อมูลลงไปได้ และสามารถลบได้โดยใช้วิธี Ultra violet ฉายผ่านช่องกระจกบนตัวไอซี และหลังจากที่ลบข้อมูล ก็สามารถนำไปโปรแกรมใหม่ได้อีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrically Alterable Read Only Memory หรือ EAROM

เป็นหน่วยความจำชนิดนี้ ผู้ใช้สามารถโปรแกรมข้อมูลเข้าไปได้ และสามารถลบออกได้โดยใช้สัญญาณไฟฟ้า ทำให้การโปรแกรม และการลบทำได้โดยสะดวก และอาจทำได้โดยไม่ต้องถอดออกจากวงจร ซึ่งคล้ายกับหน่วยความจำในตระกูล MCS-51 ของ บริษัท ATMEL มาก เช่นเบอร์ 89C51 และ 89C52 ที่ใช้ในการทำโปรเจคนี้ โดยใช้เป็นโปรแกรมมอนิเตอร์ไว้รับข้อมูลผ่านเครื่อง PC

**2.3.6 การต่อ MCS-51 เข้ากับ EPROM และ RAM**

การต่อใช้งานนี้เราอาจจะไม่ใช้ Decoder มาถอดรหัสตำแหน่งของหน่วยความจำก็ได้ ถ้าเราติดตั้งเพียงอย่างละ 1 ตัวเท่านั้น เพราะแอดเดรสของ Program Memory และ Data Memory ใช้แอดเดรสซ้อนกันอยู่ สัญญาณที่ใช้ในการอ่านข้อมูลจาก Program Memory (EPROM) ก็คือ PSEN\ และสัญญาณที่ใช้อ่านข้อมูลจาก Data Memory (RAM) ก็คือ RD\ ส่วนการเขียนข้อมูลลง RAM จะใช้ WR\ แต่เนื่องจากในการทำโปรเจคครั้งนี้จะต้องมีการ Download ข้อมูลโปรแกรมจาก PC ลงสู่ตัว RAM ซึ่ง RAM นี้จะต้องมีการเขียน อ่าน และรันโปรแกรม เราจึงนำขา PSEN\ กับ RD\ มาทำการ AND กัน แล้วนำเอาต์พุตที่ได้มาต่อกับขา RD\ ของ RAM

**2.4 สเต็ปปิ้งมอเตอร์**

สเต็ปปิ้งมอเตอร์แบ่งตามพื้นฐานได้เป็น 3 ชนิด

1.วาริเอเบิลรีลักแตนซ์ (Variable reluctance : VR)

2. เพอร์มาเนนต์แมกเนต (Permanent magnet : PM)

3.ไฮบริด (Hybrid)

**2.4.1 ชนิดวาริเอเบิลรีลักแตนซ์ (Variable reluctance : VR)**

มีโครงสร้างของโรเตอร์แบบมัลติทูธ (Multi tooth) ทำจากเหล็กอ่อนมอเตอร์ชนิดนี้เมื่อใช้มือหมุนเพลาของมอเตอร์ที่โรเตอร์จะไม่ปรากฏการณ์ทางแม่เหล็ก (Magnetism) จึงสามารถหมุนได้ตลอดโดยไม่ติดขัดแตกต่างจากชนิด PM และชนิด Hybrid ซึ่งมีสนามแม่เหล็กที่โรเตอร์ เมื่อหมุนจะรู้สึกติดขัด เหมือนเป็นเฟือง สเต็ปปิ้งมอเตอร์ชนิดนี้มีจุดด้อยในเรื่องความถูกต้องของตำแหน่งและทำงานได้ไม่ดีนักเมื่อมีสเต็ปในการหมุนสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4.2 ชนิดเพอร์มาเนนต์แมกเนต (Permanent magnet : PM)**

มีโครงสร้างของโรเตอร์แบบเรียบไม่มีซี่ขั้วแม่เหล็ก บนโรเตอร์จะเป็นแบบแม่เหล็กถาวรการควบคุมทำได้โดยป้อนกระแสกระตุ้นที่ขดลวดบนสเตเตอร์ เช่น ถ้าเป็นสเตเตอร์แบบ 4 เฟส จะมีขั้วแม่เหล็กอยู่ 4 ขั้ว ซึ่งมีคอยล์พันอยู่แยกจากกันขั้วแม่เหล็กถาวรบนโรเตอร์ จะถูกแรงดึงดูดจากขั้วแม่เหล็กบนสเตเตอร์ ถึงแม้จะไม่ป้อนกระแสไฟฟ้าอีกต่อไป ทำให้เกิดเป็นแรงยึดหน่วงขึ้น สเต็ปปิ้งมอเตอร์ชนิดนี้มีข้อดีในเรื่องความถูกต้องของตำแหน่งเมื่อเปรียบเทียบกับสเต็ปปิ้งมอเตอร์ชนิดอื่น

**2.4.3 ชนิดไฮบริด (Hybrid)**

เป็นชนิดที่นิยมใช้มากที่สุด โดยเฉพาะเมื่อนำมาใช้ในอุปกรณ์ที่ใช้ในเครื่องคอมพิวเตอร์ สเต็ปปิ้งมอเตอร์ชนิดนี้มีโครงสร้างภายในคือ สเตเตอร์เป็นชนิดวาริเอเบิลรีลักแตนซ์ ส่วนโรเตอร์จะเป็น ชนิดเพอร์มาเนนต์แมกเนตนำมาประกอบเข้าด้วยกัน จึงทำให้เป็นมอเตอร์ชนิดที่มีแรงยึดหน่วงสูง มีแรงบิด แรงดึง และแรงผลักดี และยังคงทำงานได้ดีแม้จะมีจำนวนของสเต็ปต่อรอบในการหมุนสูงก็ตาม

ยังมี สเต็ปปิ้งมอเตอร์อีกชนิดหนึ่งที่ปรับปรุงมาจาก ชนิดเพอร์มาเนนต์แมกเนต คือ ชนิดแรเอิร์ธเพอร์มาเนนต์แมกเนต (Rare earth permanent magnet ) หรือเรียกว่า ชนิด ดิสก์แมกเนตสเต็ปปิ้งมอเตอร์ (Disk magnet steppers) โครงสร้างของโรเตอร์ของมอเตอร์ชนิดนี้มีลักษณะเป็นแผ่นซึ่งยึดติดกับเพลาของมอเตอร์ โครงสร้างแบบนี้ทำให้เกิดโมเมนต์ของความเฉื่อยต่ำมาก มีอัตราเร่งสูง มอเตอร์ชนิดนี้จึงจัดเป็นมอเตอร์ที่มีประสิทธิภาพสูงทั้งในด้านแรงบิดดึง กำลังทางกลที่ได้ของมอเตอร์ ความถูกต้องของตำแหน่งสูงมาก และความเร็วในการเริ่มหมุน และหยุดสูง มีความสูญเสียของกำลังงานต่ำ

**2.4.4 การพันขดลวดหรือคอยล์บนสเต็ปปิ้ง**

1.แบบไบโพล่าร์ (Bipolar)

2.แบบยูนิโพล่าร์ (Unipolar)

**2.4.4.1 สเต็ปปิ้งมอเตอร์แบบไบโพล่าร์ (Bipolar)**

มีการพันขดลวด 1 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ ขั้วแม่เหล็กที่เกิดขึ้นบนสเตเตอร์ถูกกำหนดโดยทิศทางของกระแสไฟฟ้า และสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงข้าม โดยการกลับทิศทางการไหลของ กระแสไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.4.4.2 สเต็ปปิ้งมอเตอร์แบบยูนิโพล่าร์ (Unipolar)**

จะมีการพันขดลวด 2 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ ซึ่งแต่ละขดจะทำให้เกิดขั้วแม่เหล็กเปลี่ยนไปมาโดยการสวิตชิ่งกระแสไฟฟ้าจากขดลวดหนึ่งไปยังอีกขดลวดหนึ่ง โดยปกติขดลวดทั้งสองจะมีการเชื่อมต่อกัน หรือจุดร่วมเพื่อลดจำนวนของสายไฟที่ต่อจากมอเตอร์ แต่การพันขดลวดแบบยูนิโพล่าร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพล่าร์ เพราะจะมีเพียงครึ่งหนึ่งของขดลวดที่ถูกกระตุ้นทำให้ทำงาน

**2.4.5 การกระตุ้น และการควบคุมการหมุนของสเต็ปปิ้งมอเตอร์**

การกระตุ้น และการควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีเควนเชียลในรูปแบบที่ถูกต้อง สามารถแบ่งได้เป็น 3 ลักษณะ คือ

1. แบบเวฟ (Wave)
2. แบบ 2 เฟส (Two phase)
3. แบบครึ่งสเต็ป (Half step)

แบบเวฟ (Wave)

เป็นการกระตุ้นแบบที่ง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่ง และเรียงถัดกันไป เช่น ขดที่ 1, 2, 3, 4, 1 หรือ 1, 4, 3, 2, 1 ขึ้นอยู่กับทิศทางที่ต้องการให้หมุน จึงทำให้มีขดลวดเพียงขดเดียวเท่านั้นที่ถูกกระตุ้นในระยะเวลาหนึ่ง

| สเต็ปที่ | เฟสที่ 1 | เฟสที่ 2 | เฟสที่ 3 | เฟสที่ 4 |
|---|---|---|---|---|
| 1 | ทำงาน | --- | --- | --- |
| 2 | --- | ทำงาน | --- | --- |
| 3 | --- | --- | ทำงาน | --- |
| 4 | --- | --- | --- | ทำงาน |

ตารางที่ 2.7 แสดงลำดับการกระตุ้นเพื่อให้สเต็ปปิ้งมอเตอร์ทำงานด้วยการกระตุ้นแบบเวฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ 2 เฟส (Two phase)

มีลักษณะคล้ายกับแบบเวฟแต่การกระตุ้นแบบนี้จะทำการกระตุ้นโดยการจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกัน และเรียงถัดกันไปเช่นเดียวกับแบบเวฟคือเริ่มจากขดลวด 1-2, 2-3, 3-4, 4-1, 1-2 หรือ 1-4, 4-3, 3-2, 2-1, 1-4 ขึ้นอยู่กับทิศทางการหมุน การเพิ่มจำนวนของขดลวดที่ถูกกระตุ้นทำให้เพิ่มแรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ด้วยแรงดึงอย่างเต็มแรงจาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกัน และต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป แต่การกระตุ้นในลักษณะนี้ต้องใช้แหล่งจ่ายกำลังไฟฟ้ามากขึ้น

| สเต็ปที่ | เฟสที่ 1 | เฟสที่ 2 | เฟสที่ 3 | เฟสที่ 4 |
|---|---|---|---|---|
| 1 | ทำงาน | ทำงาน | --- | --- |
| 2 | --- | ทำงาน | ทำงาน | --- |
| 3 | --- | --- | ทำงาน | ทำงาน |
| 4 | ทำงาน | --- | --- | ทำงาน |

ตารางที่ 2.8 แสดงลำดับการกระตุ้นเพื่อให้สเต็ปปิ้งมอเตอร์ทำงานด้วยการกระตุ้นแบบ 2 เฟส

แบบครึ่งสเต็ป (Half step)

มีรูปแบบที่เกิดจากการผสมผสานกันระหว่างการกระตุ้นแบบเวฟ และแบบ 2 เฟสเพื่อเพิ่มจำนวนของสเต็ปต่อรอบอีกเท่าตัว ในการกระตุ้นจะกระตุ้นขดลวดเรียงกันไปเป็นลำดับ คือ เริ่มจากขดลวด 1, 1-2, 2, 2-3, 3, 3-4, 4, 4-1, 1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะมีแรงบิดมากขึ้นอีกเพราะช่วงสเต็ปมีระยะสั้นลง และแต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขด ที่ถูกกระตุ้นพร้อมกันความถูกต้องของตำแหน่งจะมีความแม่นยำมากขึ้น แต่การกระตุ้นในลักษณะนี้จะต้องทำการหมุนถึง 2 สเต็ปจึงจะได้การหมุนเท่ากับ 1 สเต็ปเหมือนกับการควบคุมใน 2 แบบแรก และแหล่งจ่ายไฟฟ้าจะเท่ากับแบบ 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| สเต็ปที่ | เฟสที่ 1 | เฟสที่ 2 | เฟสที่ 3 | เฟสที่ 4 |
|---|---|---|---|---|
| 1 | ทำงาน | --- | --- | --- |
| 2 | ทำงาน | ทำงาน | --- | --- |
| 3 | --- | ทำงาน | --- | --- |
| 4 | --- | ทำงาน | ทำงาน | --- |
| 5 | --- | --- | ทำงาน | --- |
| 6 | --- | --- | ทำงาน | ทำงาน |
| 7 | --- | --- | --- | ทำงาน |
| 8 | ทำงาน | --- | --- | ทำงาน |

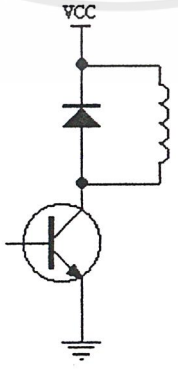
ตารางที่ 2.9 แสดงลำดับการกระตุ้นเพื่อให้สเต็ปปิ้งมอเตอร์ทำงานด้วยการกระตุ้น แบบครึ่งสเต็ป

**2.4.6 ซัพเพรสเซอร์**

เมื่อทรานซิสเตอร์หยุดนำกระแส จะทำให้เกิดแรงเคลื่อน E.M.F. ซึ่งแรงเคลื่อนชนิดนี้อาจเป็นอันตรายแก่วงจร และเป็นสัญญาณรบกวนได้ เรามีวิธีป้องกันคือ

1. ไดโอดซัพเพรสเซอร์

กระแสหมุนเวียน CIRCULATING CURRENT จะเริ่มไหลจากทรานซิสเตอร์ หยุดนำกระแส และศักย์ดาของขา C จะเท่ากับศักย์ดาของแหล่งจ่าย ข้อเสีย คือ กระแสจะหยุดหมุนเวียนอยู่นาน และจะทำให้เกิดแรงบิดห้ามล้อมาก พลังงานส่วนใหญ่จะสูญเสียในความต้านทานของขดลวด มีปัญหาเรื่องทำความเย็น

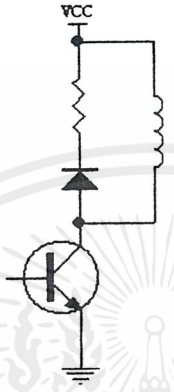



รูปที่ 2.18 แสดงการใช้ไดโอดซัพเพรสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ไดโอด และรีซิสเตอร์ซัพเพรสเซอร์

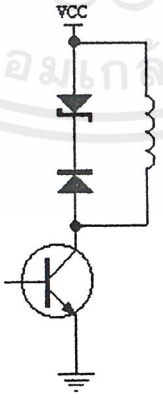
ถ้า Rs ยิ่งมาก กระแสหมุนเวียนก็จะลดลงเร็วขึ้น แต่ศักย์ดาของขา C จะมีค่าสูงขึ้น พลังงานส่วนใหญ่จะสูญเสียใน Rs




รูปที่ 2.19 แสดงการใช้ไดโอด และรีซิสเตอร์ซัพเพรสเซอร์

3. ซีเนอร์ไดโอดซัพเพรสเซอร์

เมื่อทรานซิสเตอร์ CUT OFF กระแสจะลดลงได้เร็วกว่า 2 แบบแรก และศักย์ดาที่ C จะเท่ากับศักย์ดาของซีเนอร์บวกกับศักย์ดาของแหล่งจ่ายซึ่งเป็นอิสระต่อกระแส พลังงานส่วนใหญ่สูญเสียใน ซีเนอร์

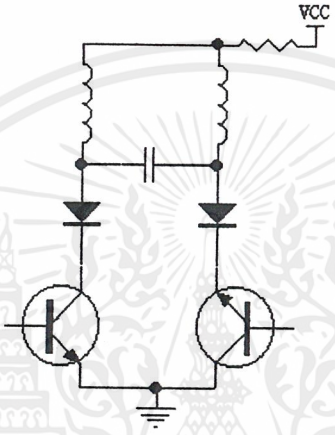



รูปที่ 2.20 แสดงการใช้ซีเนอร์ไดโอดซัพเพรสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.คอนเดนเซอร์ซัพเพรสเซอร์

จะใส่ตัว C ให้ 01 กับ 03 และ 02 กับ 04 เมื่อทรานซิสเตอร์หยุดนำกระแส กระแสตัว C จะไหลผ่านทางไดโอด และจะดูดกลืนกระแสที่ค่อยๆ ลดลงจากขดลวดของมอเตอร์ เพื่อป้องกันทรานซิสเตอร์ และยังช่วยแดมป์ไฟ คือช่วยลดความร้อนที่เกิดขึ้นในขดลวด STATOR เนื่องจากการ OSCILLATE ของ ROTOR




รูปที่ 2.21 แสดงการใช้คอนเดนเซอร์ซัพเพรสเซอร์

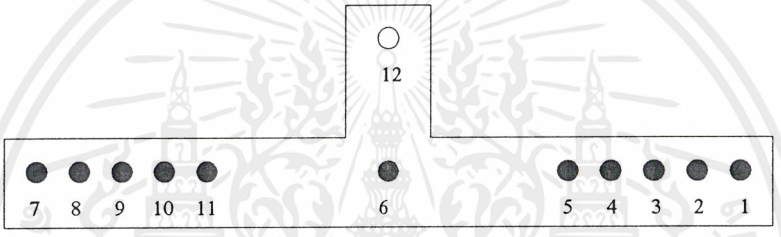
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การออกแบบ

**3.1 ภาคเซ็นเซอร์**

ในที่นี้เราได้ใช้ลักษณะการส่ง และรับของแสงอินฟราเรด โดยอาศัยหลักการสะท้อนของแสงเป็นตัวตรวจจับ เราได้เลือกใช้ตัวส่ง และตัวรับที่อยู่ในตัวโมดุลเดียวกัน เพื่อเป็นการง่ายที่จะจัดเซ็นเซอร์ทั้ง 12 ตัวให้อยู่ในระนาบเดียวกันมากที่สุด ละอีกทั้งจะทำให้มีขนาดเล็กลงอีกด้วย

จากนั้นเราก็มากำหนดตำแหน่งของตัวตรวจจับ เพื่อให้สอดคล้องกับลักษณะของกำแพง โดยเราจะวางตำแหน่งต่างๆ ไว้ดังรูป 3.1




รูปที่ 3.1แสดงการวางตำแหน่งของเซ็นเซอร์

จะเห็นว่าเราใช้ตัวตรวจจับกำแพงด้านหน้าอยู่ 2 ตัว โดยมีตัวหนึ่งสามารถที่จะรับรู้กำแพงด้านหน้าได้ไวกว่า ที่ทำเช่นนี้เพื่อจะได้เป็นการส่งสัญญาณให้แก่ไมโครคอนโทรลเลอร์ จะได้สั่งให้สเต็ปปิ้งมอเตอร์ทำการชะลอตัวไมโครเมาส์ ก่อนที่จะเกิดการชนกำแพงเสียก่อน

ส่วนการตรวจจับกำแพงด้านข้าง เราจะใช้ตัวตรวจจับข้างละ 5 ตัว โดยในการควบคุมในการเคลื่อนที่ของตัวไมโครเมาส์ เราจะพยามชดเชยให้อยู่แนวกลางเสมอ ซึ่งมีผลอย่างมากในการจดจำเส้นทาง

จากนั้นเราได้มาออกแบบลักษณะการจ่ายพลังงานให้แก่ โฟโตไดโอด ซึ่งมี 2 แนวทาง คือ การส่งเป็นพัลซ์ ซึ่งการส่งอย่างนี้จะทำให้ส่งได้ไกลขึ้น ซึ่งจะทำให้ตัวตรวจจับอาจต่อได้โดยตรงกับไมโครคอนโทรลเลอร์ได้ แต่หากเราจะเลือกวิธีนี้ เราจำเป็นต้องหาวงจรกำเนิดสัญญาณมาเพิ่มเติม อีกทั้งหากการบัดกรีตัวส่ง และตัวรับไม่ได้อยู่ในระนาบเดียวกันจริงๆ ก็จะเกิดข้อผิดพลาดขึ้นได้ ดังนั้นเราจึงเลือกใช้แนวทางที่ 2 คือการให้พลังงานที่จ่ายเป็นแบบไฟตรงคงที่ การให้แหล่งจ่ายแบบนี้แก่ตัวโฟโตไดโอด จะทำให้สัญญาณที่โฟโตทรานซิสเตอร์ได้มีกำลังอ่อนลง จึงต้องมีการขยายสัญญาณ ในที่นี้เราเลือกใช้ออปแอมป์เบอร์ LM339 ซึ่งมีออปแอมป์ 4 ตัวในรูปแบบไอซีตัวเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กัน โดยจะมีความต้านทานปรับค่าได้ ต่อทางด้านอินพุต เพื่อเป็นปรับแต่งการขยายสัญญาณได้อีกด้วย อีกทั้งจะเป็นการชดเชยเมื่อมีการบัดกรีตัวส่งและตัวรับที่ไม่ได้อยู่ในระนาบเดียวกันด้วย

เมื่อรับได้ตำแหน่งการวางของตัวอุปกรณ์ และลักษณะการจ่ายพลังงานแล้วเราก็มาออกแบบแผ่นวงจรพิมพ์ ซึ่งรวมถึงวงจรขยายสัญญาณออปแอมป์รวมด้วยกัน

### 3.2 ไมโครคอนโทรลเลอร์

เราได้ใช้บอร์ดคอนโทรลสำเร็จรูปของ SILA รุ่น V-3155 เพื่อเป็นการสะดวก และประหยัดเวลา แต่ทั้งนี้ก็ไม่มีบอร์ดใดที่จะเหมาะสมกับงานของเราโดยตรง เราจึงต้องมีการดัดแปลง และปรับปรุงเพิ่มเติม โดยเราจะทำให้บอร์ดนี้สามารถรันโปรแกรม อ่าน และเขียนบนหน่วยความจำแบบ RAM ได้

### 3.3 ภาคขุดขับ และสเต็ปปิ้งมอเตอร์

เราได้ใช้สเต็ปปิ้งมอเตอร์ที่มีอยู่ใน Disk Drive ขนาด 5.25 นิ้วในคอมพิวเตอร์รุ่นเก่าๆ ซึ่งเราก็ไม่รู้ค่ารายละเอียดมันนัก เนื่องจากไม่มี Specification บอก ทำให้เราต้องออกแบบวงจรขับให้สามารถขับกระแสสูงๆได้ไว้ก่อน โดยเราจะใช้ทรานซิสเตอร์ 2 ตัวต่อแบบดาร์ลิงตันกันต่อขดลวด 1 เฟส

### 3.4 ด้าน Mechanic

เมื่อเราได้ขนาดแผ่นวงจร อุปกรณ์ต่างๆ และมอเตอร์ เราจึงมาออกแบบการจัดวางของสิ่งเหล่านี้ เราได้เลือกใช้อลูมิเนียมเป็นโครงสร้างหลักที่สำคัญทั้งหมดในตัวไมโครเมาส์ เนื่องจากมีน้ำหนักเบา รองลงมาก็มีใช้แผ่นพลาสติกบ้างในการเชื่อมต่อของโครงสร้าง จากนั้นเราก็ได้กำหนดขนาดของโครง เพื่อให้เข้ากับสิ่งที่เราต้องการให้รวมเข้าด้วยกัน

### 3.5 ด้านโปรแกรม

แบ่งออกเป็น 3 ส่วน คือ

#### 3.5.1 ส่วนของโปรแกรมรับ HEX ไฟล์

ซึ่งเป็นโปรแกรมที่จะทำหน้าที่รับข้อมูลที่เป็น HEX ไฟล์จากคอมพิวเตอร์ มาใส่เก็บไว้ใน RAM ภายในบอร์ดควบคุมของไมโครเมาส์ซึ่งข้อมูลที่เป็น HEX ไฟล์นี้ก็ คือโปรแกรมในส่วนของการควบคุมการเคลื่อนที่ และการตัดใจของไมโครเมาส์ที่เขียน และพัฒนาได้ โดยการเขียนโปรแกรมเป็นภาษาแอสแซมบี้ แล้วใช้โปรแกรมในการแปลงโปรแกรมที่เขียนขึ้นนี้เป็น HEX ไฟล์ และทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหลดโปรแกรมดังกล่าวเข้าไปไว้ใน RAM เพื่อให้ไมโครเมาส์ทำงานได้ตามต้องการ ดังนั้นจึงจำเป็นที่จะต้องมีโปรแกรมรับ HEX ไฟล์เพื่อทำหน้าที่นำข้อมูลที่เป็น HEX ไฟล์ของโปรแกรมควบคุมมาจัดเก็บไว้ใน RAM เพื่อให้ไมโครเมาส์ทำงาน หรือทดสอบการทำงานของไมโครเมาส์ได้

เนื่องจากโปรแกรมที่แปลงเป็น HEX ไฟล์แล้วจะมีรูปแบบที่แน่นอนดังนี้

:BCAAAATTHH.....HHCC

: คือ START CHARACTER

BC คือ จำนวน BYTE ของข้อมูลในบรรทัด มีค่าเป็นเลขฐาน 16

AAAA คือ ADDRESS ของข้อมูลใน BYTE แรก

TT คือ ชนิดของข้อมูลในบรรทัดนั้นๆ

ถ้า TT = 0 เป็น DATA RECORD

ถ้า TT = 1 เป็น END-OF-FILE RECORD

HH คือ ข้อมูลแต่ละ BYTE

CC คือ ค่า CHECKSUM ของบรรทัดนั้นๆ โดยจะเป็นค่า 2's COMPLEMENT

ในการส่งข้อมูลจากคอมพิวเตอร์จะเป็นรหัส ASCII ฉะนั้นโปรแกรมรับข้อมูลจะต้องนำรหัส ASCII ที่ได้มาแปลงเป็นข้อมูลแล้วนำเก็บไว้ใน RAM ส่วนอัตราความเร็วในการส่งก็จะต้องมีการตั้งอัตราในการรับส่งให้เท่ากันระหว่างคอมพิวเตอร์ กับตัวไมโครคอนโทรลเลอร์ ซึ่งเลือกใช้ที่อัตรา BAUD RATE 9600 ฉะนั้นภายในโปรแกรมรับ HEX ไฟล์ จะต้องมีการตั้งค่า BAUD RATE ไว้ที่ 9600 ด้วยซึ่งจะทำให้การรับส่งข้อมูลไม่ผิดพลาด

โปรแกรมรับ HEX ไฟล์นี้เป็นโปรแกรมที่มีความสำคัญโปรแกรมหนึ่ง ถ้าโปรแกรมในส่วนนี้มีความผิดพลาด ก็จะทำให้ข้อมูลที่โหลดเข้าไปใน RAM เกิดผิดพลาด ซึ่งก็จะเป็นผลให้ไมโครเมาส์ไม่ทำงานตามต้องการ จึงต้องเขียนโปรแกรมในส่วนนี้ให้มีประสิทธิภาพ

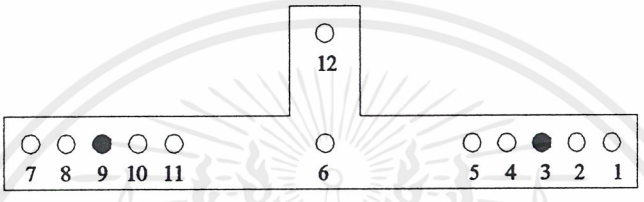
### 3.5.2 ส่วนของโปรแกรมการปรับล้อ

ในการเคลื่อนที่ของไมโครเมาส์ไปในช่องทางที่เป็นเขาวงกตนั้น ไมโครเมาส์อาจเคลื่อนที่ไปไม่ตรงช่องทาง ยิ่งถ้ามีการเลี้ยวเกิดขึ้น ก็อาจจะทำให้ไมโครเมาส์อยู่ไม่ตรงช่องทางวิ่ง ถ้าปล่อยให้ไมโครเมาส์เคลื่อนที่ไปก็อาจจะทำให้ไมโครเมาส์ชนกับช่องทางวิ่งที่เป็นกำแพงได้ และก็ไม่สามารถเคลื่อนที่ต่อไปได้จึงจำเป็นที่จะต้องมีโปรแกรมช่วยในการปรับล้อ หรือปรับให้ไมโครเมาส์เคลื่อนที่ให้ตรงช่องทางมากที่สุด

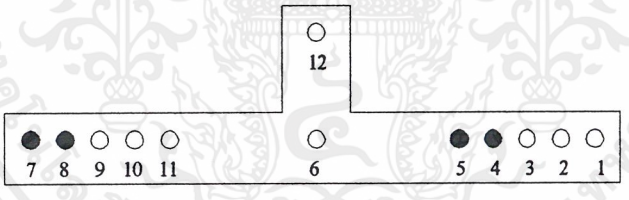
การที่เราจะทราบได้ว่าไมโครเมาส์เคลื่อนที่ไม่ตรงช่องทางนั้นจะใช้เซ็นเซอร์ของปีกทั้งสองข้าง ซึ่งใช้เซ็นเซอร์ 10 ตัวในการตรวจจับ ซึ่งใช้หลักการดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

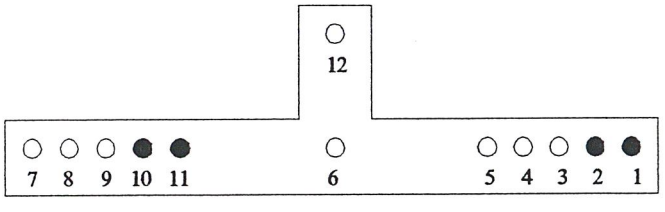
ถ้าเซ็นเซอร์ตัวที่ 9 และ 3 ตรวจจับได้ ก็จะให้ไมโครเมาส์วิ่งตรงไปโดยไม่มีการปรับล้อทั้งสองข้าง แต่ถ้าเซ็นเซอร์ตัวที่ 1, 2, 10 และ 11 ตรวจจับได้ ซึ่งก็แสดงว่าตัวไมโครเมาส์เคลื่อนที่ไม่ตรงช่องทางวิ่งแล้ว มีการเอียงไปทางซ้าย ก็จะต้องมีการปรับล้อ โดยที่จะให้ล้อทางด้านซ้ายของตัวไมโครเมาส์เคลื่อนที่เร็วกว่าล้อทางด้านขวา และถ้าเซ็นเซอร์ตัวที่ 4, 5, 7 และ 8 ตรวจจับได้ ก็แสดงว่ามีการเอียงไปทางขวา จึงต้องมีการปรับล้อโดยให้ล้อทางด้านขวาของตัวไมโครเมาส์เคลื่อนที่เร็วกว่าล้อทางด้านซ้าย




รูปที่ 3.2 แสดงเมื่อไมโครเมาส์อยู่ในตำแหน่งวิ่งตรง




รูปที่ 3.3 แสดงเมื่อไมโครเมาส์อยู่ในตำแหน่งเอียงไปทางขวา




รูปที่ 3.4 แสดงเมื่อไมโครเมาส์อยู่ในตำแหน่งเอียงไปทางซ้าย

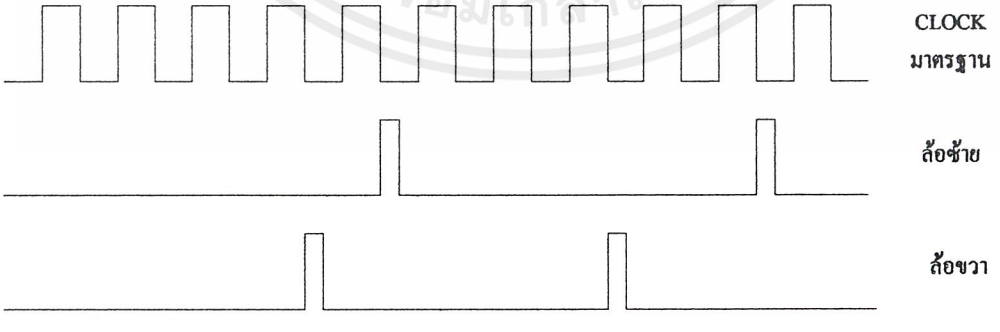
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าไมโครเมาส์เคลื่อนที่ไปในช่องทางที่มีกำแพงเพียงด้านเดียว ก็จะใช้เซ็นเซอร์ของปีกทางด้านที่สามารถจับได้เพียงด้านเดียวในการปรับล้อให้ไมโครเมาส์วิ่งตรง ตามที่กล่าวมาแล้วข้างต้นว่าถ้าเซ็นเซอร์ตัวใดตรวจจับได้ก็ให้ปรับล้อใด และถ้าช่องทางที่ไมโครเมาส์เคลื่อนที่ไปไม่มีกำแพงทั้งสองด้าน ก็จะให้ไมโครเมาส์เคลื่อนที่ตรงไปโดยไม่มีการปรับล้อ

หลักในการปรับความเร็วของล้อทำได้ดังนี้ คือ

ต้องสร้างสัญญาณ Clock มาตรฐานขึ้นมา แล้วนำสัญญาณ Clock ที่สร้างนี้มาใช้เป็นสัญญาณอ้างอิงในการที่จะกระตุ้นให้สเต็ปปิ้งมอเตอร์ทำงาน โดยการนับจำนวนสัญญาณ Clock ให้ได้ตามต้องการแล้วก็ทำการกระตุ้นสเต็ปปิ้งมอเตอร์ เช่นเมื่อนับสัญญาณ Clock ครบ 5 ลูกแล้วก็ให้กระตุ้นสเต็ปปิ้งมอเตอร์ 1 ครั้ง ซึ่งถ้าต้องการให้ไมโครเมาส์เคลื่อนที่ตรงไปโดยไม่มีการปรับล้อ ก็จะต้องกระตุ้นสเต็ปปิ้งมอเตอร์พร้อมกันทั้งล้อซ้าย และขวา ส่วนในการปรับความเร็วล้อก็ใช้การนับสัญญาณ Clock เช่นกัน คือถ้าต้องการให้ล้อใดหมุนเร็วก็กระตุ้นสเต็ปปิ้งมอเตอร์ล้อนั้นโดยการนับสัญญาณ Clock ให้น้อยลงเหลือ 4 ลูกแล้วจึงกระตุ้นสเต็ปปิ้งมอเตอร์ที่ล้อนั้นเช่น ถ้าต้องการให้ล้อขวาหมุนเร็วกว่าล้อซ้าย ก็ให้นับสัญญาณ Clock ครบ 4 ลูกแล้วจึงกระตุ้นสเต็ปปิ้งมอเตอร์ที่ล้อขวา และนับสัญญาณ Clock ต่อจนครบ 5 ลูกแล้วจึงกระตุ้นสเต็ปปิ้งมอเตอร์ที่ล้อซ้าย และจะทำการปรับความเร็วของล้อต่อไปจนกว่าไมโครเมาส์จะวิ่งตรงแล้วจึงหยุดปรับ

ส่วนการนับสัญญาณ Clock จำนวนเท่าไรแล้วกระตุ้นสเต็ปปิ้งมอเตอร์นั้น จะมีค่าเท่าไรก็ได้ตามความเหมาะสมซึ่งเกิดจากการทดลอง ซึ่งในที่นี้ได้ทำการทดลองแล้วได้ค่า 5 ลูกกับ 4 ลูกซึ่งเป็นค่าที่เหมาะสมแล้ว




รูปที่ 3.5 แสดงการป้อนสัญญาณ Clock ให้ล้อขวาหมุนเร็วกว่าล้อซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.5.3 ส่วนโปรแกรมเพื่อจดจำเส้นทาง**

ก่อนอื่นเราต้องทราบพิกัดบล็อกของตำแหน่งเริ่มต้น และปลายทางก่อน โดยปกตินั้นจะให้บล็อกเริ่มต้นอยู่ที่พิกัด 0, 0 ส่วนจุดปลายทางนั้นจะเป็นบล็อกใดๆ ก็ได้ภายในสนามเขาวงกต

หลักสำคัญในการจำเส้นทางคือ การทำให้ทราบถึงจำนวนความถี่ของแต่ละพิกัดที่ไมโครเมาส์ได้เคลื่อนที่ผ่านมา โดยถ้าหากทางใดได้เดินผ่านไปมากครั้งแล้ว ไมโครเมาส์จะพยามไม่ผ่านไปพิกัดนั้นอีก แต่ในกรณีที่มีทางแยก และแต่ละทางต่างก็มีจำนวนความถี่ในการผ่านเท่ากัน หรือว่าในทางแยกนั้นต่างก็ไม่เคยผ่านทั้งหมด เรามีเกณฑ์ในการตัดสินใจโดยเลือกให้ความสำคัญในทางตรงเป็นอันดับแรก เนื่องด้วยทุกครั้งที่เกิดการหมุนตัวเพื่อทำการเลี้ยว หรือกลับหลังหัน ตัวไมโครเมาส์จะมีโอกาสที่อยู่ในตำแหน่งที่ไม่ถูกต้องได้ง่ายกว่าการเดินตรงไปเรื่อยๆ ส่วนการให้ความสำคัญในการเลี้ยวซ้ายหรือขวานั้น เราจะให้ทางใดก่อนก็ได้ ในที่นี้ได้ให้ความสำคัญทางด้านซ้ายสำคัญกว่าทางด้านขวา

การเพิ่มค่าความถี่ในแต่ละพิกัด

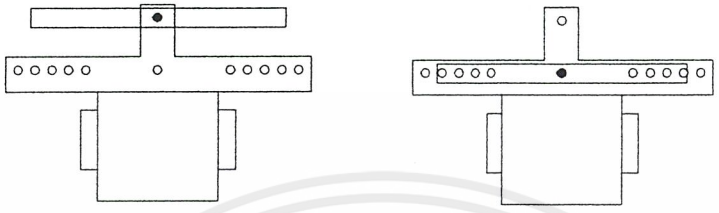
ปัญหาของการเพิ่มค่าความถี่มีอยู่ว่า เราจะทราบได้อย่างไรว่าขณะนี้ตัวไมโครเมาส์ได้กำลังอยู่ในพิกัดใด ซึ่งเมื่อทราบก็จะมีผลทำให้การเพิ่มค่าความถี่ได้อย่างถูกต้อง

การนับจำนวนสเต็ปที่ป้อนให้กับสเต็ปปิ้งมอเตอร์เป็นทางแก้ไขที่เราได้เลือกใช้ จากการทดลองพบว่า เมื่อป้อนสัญญาณแก่สเต็ปปิ้งมอเตอร์จำนวน 215 พัลซ์ ไมโครเมาส์จะเคลื่อนที่ได้ใกล้เคียงกับระยะทางของบล็อกหนึ่งที่สุด ซึ่งทำให้ทราบว่าไมโครเมาส์ได้เคลื่อนที่ผ่านพิกัดใดแล้วบ้างได้อย่างแม่นยำเกือบครบ 100 เปอร์เซ็นต์ บางคนอาจยังสงสัยว่า เหตุใดจึงไม่สามารถทำให้ความแม่นยำได้เต็มร้อย หากเราลองพิจารณาถึงหลักความจริง จะพบว่าไมโครเมาส์คงจะไม่สามารถเดินได้ตรงตลอดการเคลื่อนที่แน่ๆ ซึ่งอาจเกิดจากการผิดพลาดของตัวไมโครเมาส์เองเป็นต้นว่าล้อทั้งสองข้างมีขนาดต่างกันเพียงนิด หรือเกิดการผิดพลาดในสนามเขาวงกตที่ถูกสร้างขึ้นโดยที่พื้นของสนามไม่ได้ระนาบเดียวกันทั่วทั้งสนาม ถ้าไม่มีการชดเชยข้อผิดพลาดที่ค่อยๆสะสมขึ้นมาเหล่านี้ จะส่งผลให้โปรแกรมส่วนของการเพิ่มค่าความถี่คำนวณผิดได้ เนื่องจากข้อผิดพลาดนี้ไม่ได้ส่งผลในระยะทางอันสั้น ดังนั้นเราจึงชดเชยข้อผิดพลาดเมื่อไมโครเมาส์เคลื่อนที่เข้าสู่ทางที่มีกำแพงด้านหน้าขวางทางอยู่

การโปรแกรมเพื่อชดเชยข้อผิดพลาดนี้ กำหนดให้ทุกครั้งที่เซ็นเซอร์ด้านหน้าตรวจพบว่ามีกำแพงอยู่ข้างหน้าอยู่ จะสั่งให้ไมโครเมาส์เคลื่อนที่ไปข้างหน้าเรื่อยๆ จนกระทั่งที่เซ็นเซอร์ส่วนกลางได้ตรวจพบกำแพง โดยไม่สนใจการนับจำนวนสเต็ปที่ป้อนให้กับสเต็ปปิ้งมอเตอร์ว่าจะขาด หรือเกิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

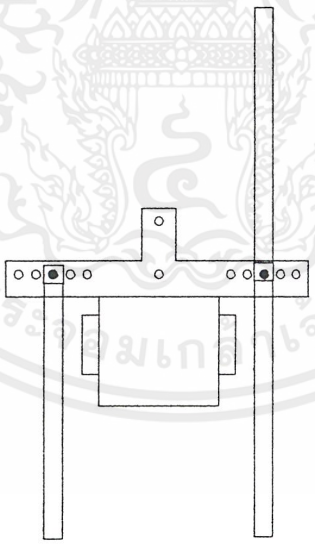
ไปจากจำนวน 215 พัลซ์ หรือไม่ จากนั้นไมโครเมาส์จะทำการหมุนตัว 90° ไปในทางใด หรือกลับหลังหมุนตัว 180° ตามขั้นตอนของโปรแกรมต่อไป



รูปที่ 3.6 แสดงการตรวจพบกำแพงด้านหน้า และเดินตรงต่อเพื่อหาจุดหมุน

วิธีการตรวจจับกำแพงด้านข้าง

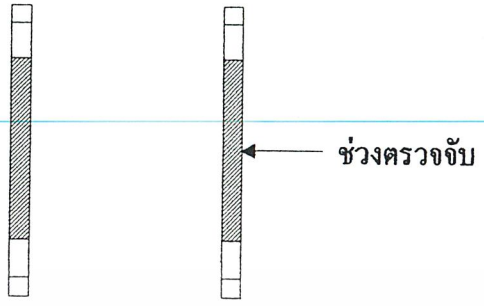
เหตุจำเป็นที่ต้องมีวิธีการตรวจจับกำแพงด้านข้าง เพราะลักษณะกำแพงจะทำให้การตรวจจับของไมโครเมาส์พลาดได้ดังรูป



รูปที่ 3.7 แสดงการตรวจจับกำแพงด้านข้างเกิดการผิดพลาด

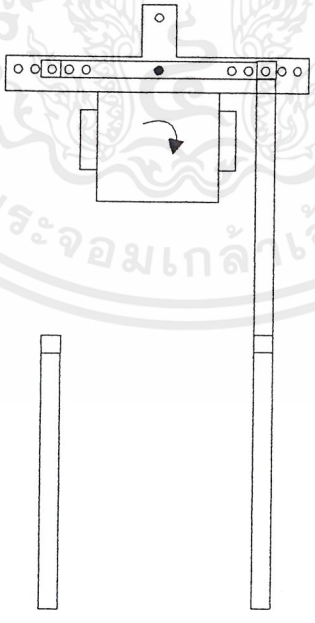
ส่วนวิธีแก้ไข ทำได้โดยจะเลือกช่วงการตรวจจับ แทนที่จะตรวจจับกำแพงด้านข้างทั้งหมด ก็จะเลือกช่วงกลาง (ส่วนแรเงา) ดังรูปที่ 3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.8 แสดงช่วงการตรวจจับกำแพงด้านข้าง

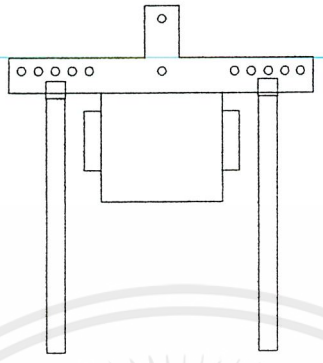
การตรวจจับนี้ดูเหมือนไม่น่ามีปัญหาอันใด แต่แท้จริงต้องมีการใช้โปรแกรมเข้ามาช่วย ปัญหาก็ คือ ไมโครเมาส์ มีโอกาสที่เคลื่อนที่ครบ 215 สเต็ป แต่จริงๆยังเคลื่อนที่ไม่ครบหนึ่งบล็อก ซึ่งเกิดจากการเดินเบี้ยว หลังจากที่ไมโครเมาส์เคลื่อนที่ครบ 215 สเต็ป ไมโครเมาส์ก็จะทำการตรวจสอบกำแพงด้านข้างทันทีผลก็คือ ไมโครเมาส์เกิดการเข้าใจผิดว่ามีกำแพงด้านข้าง ทำให้มีการตัดสินใจในการเลือกเส้นทางผิดพลาด จากรูปที่ 3.9 แทนที่ไมโครเมาส์จะทำการหมุนตัวไปทางซ้ายไมโครเมาส์กลับทำการการหมุนตัว 180° โดยเข้าใจผิดคิดว่าเป็นทางตัน



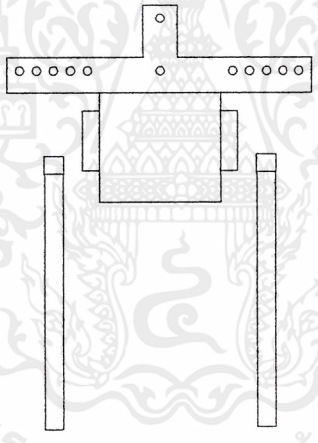
รูปที่ 3.9 แสดงไมโครเมาส์เกิดการเข้าใจผิดว่ามีกำแพงด้านข้างจึงหมุนตัว 180°

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับกรณีที่ไมโครเมาส์ได้เดินเกินหนึ่งบล็อก ก็จะทำให้เกิดการผิดพลาดได้เช่นกัน

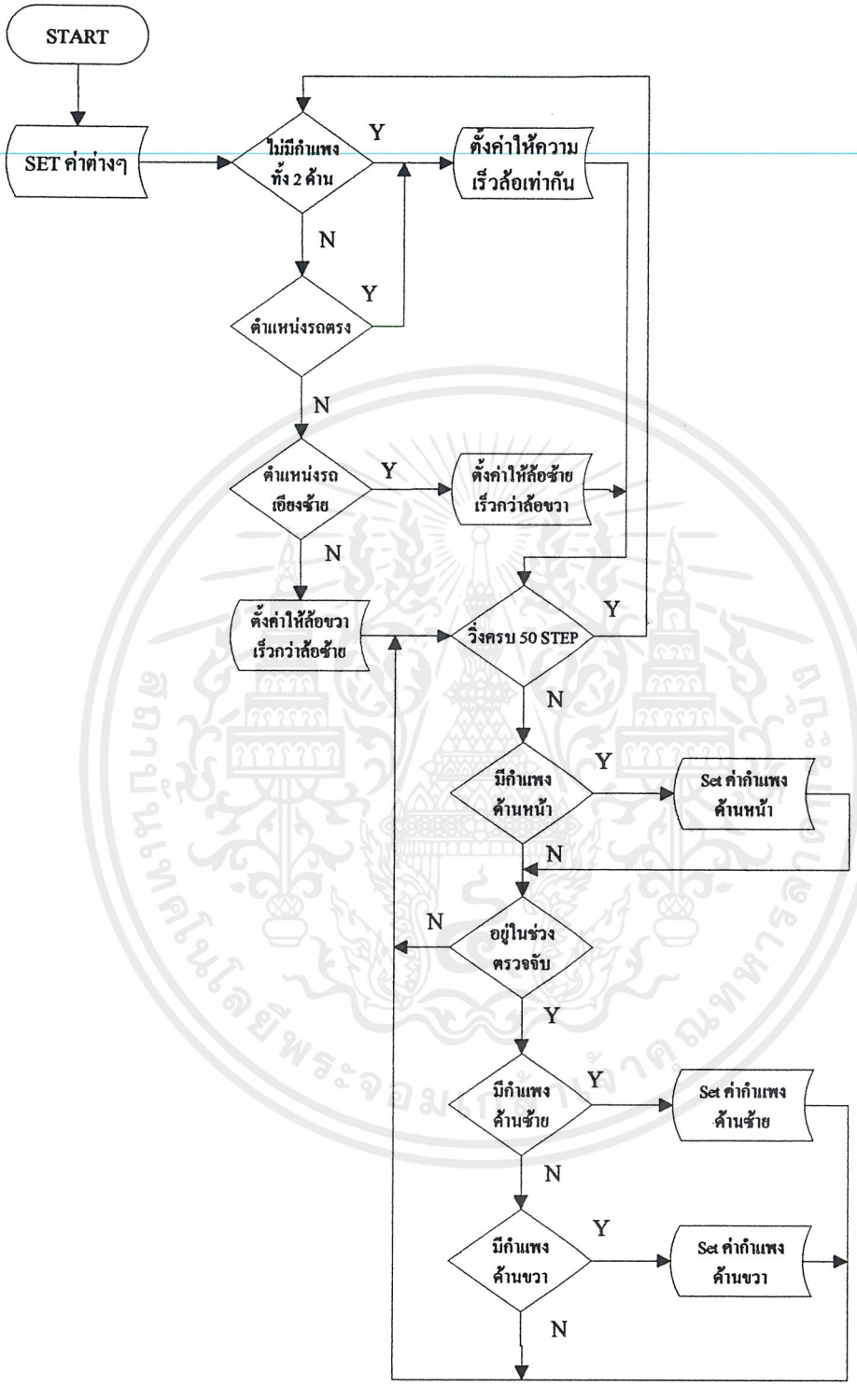


รูปที่ 3.10 แสดงไมโครเมาส์เมื่อเดินไม่ครบบล็อก



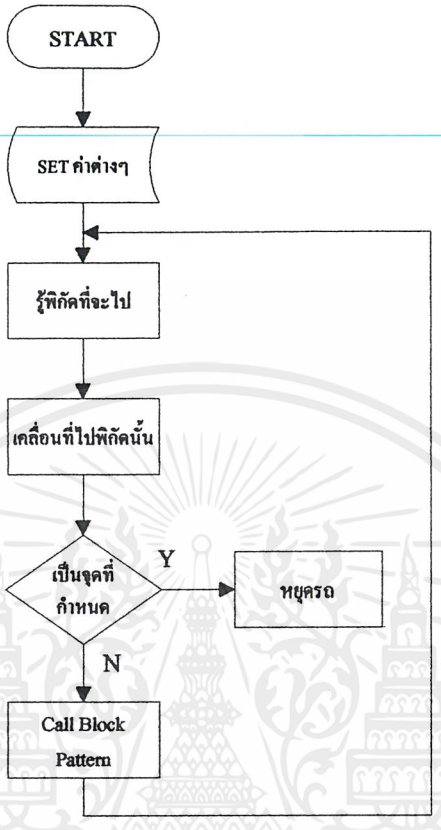
รูปที่ 3.11 แสดงไมโครเมาส์เมื่อเดินเกิน 1 บล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

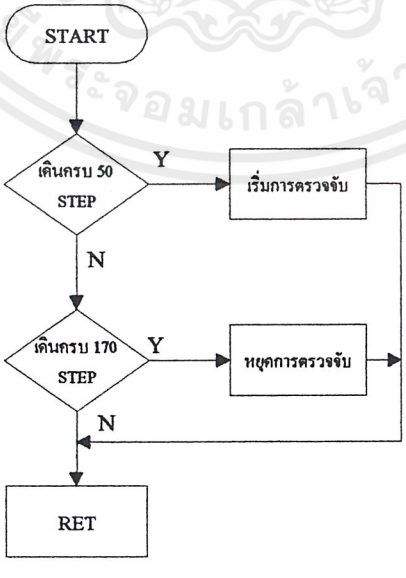



รูปที่ 3.12 แสดง Flow Chart การทำงาน MAIN หลัก 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

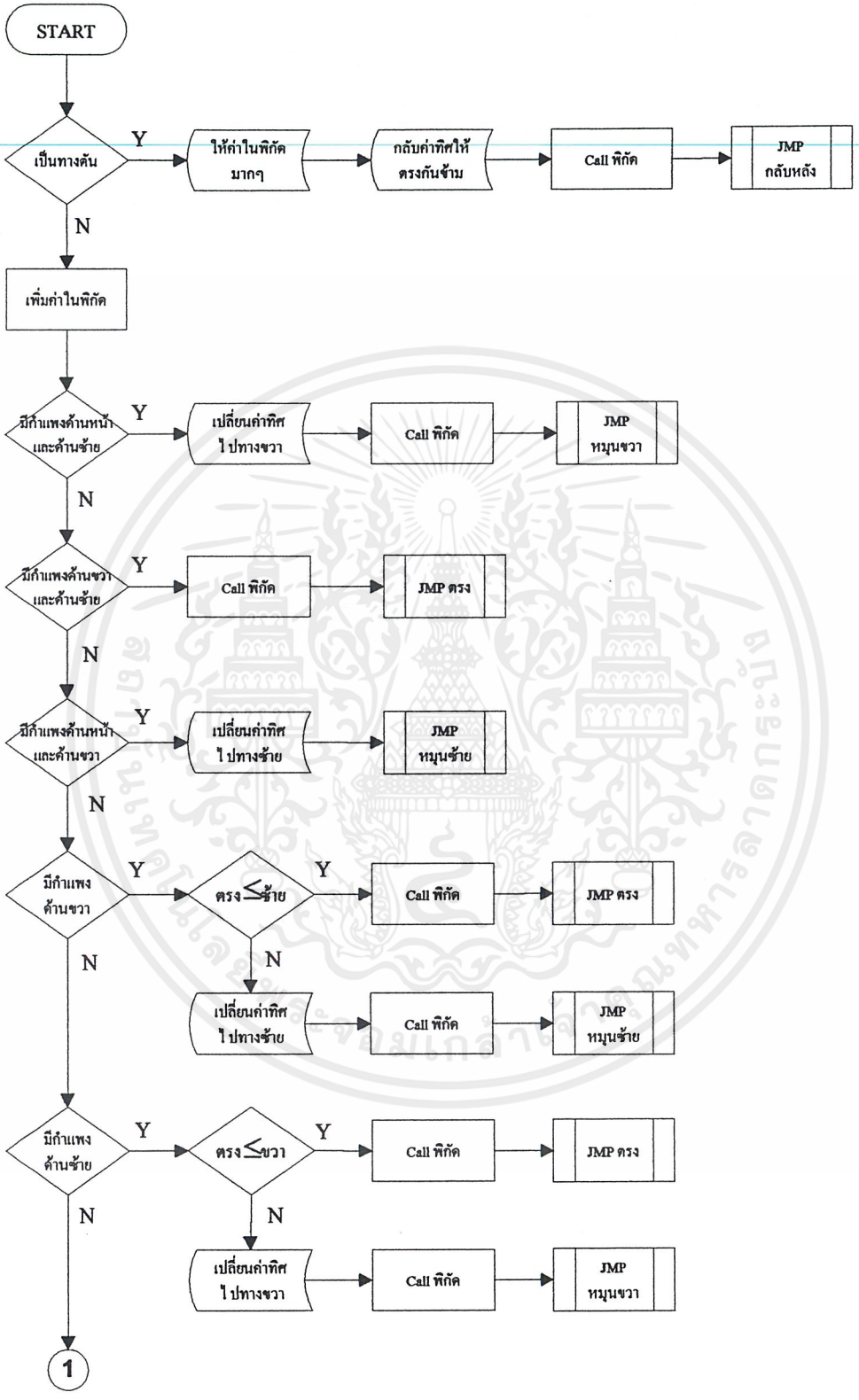



รูปที่ 3.13 แสดง Flow Chart การทำงาน MAIN หลัก 2



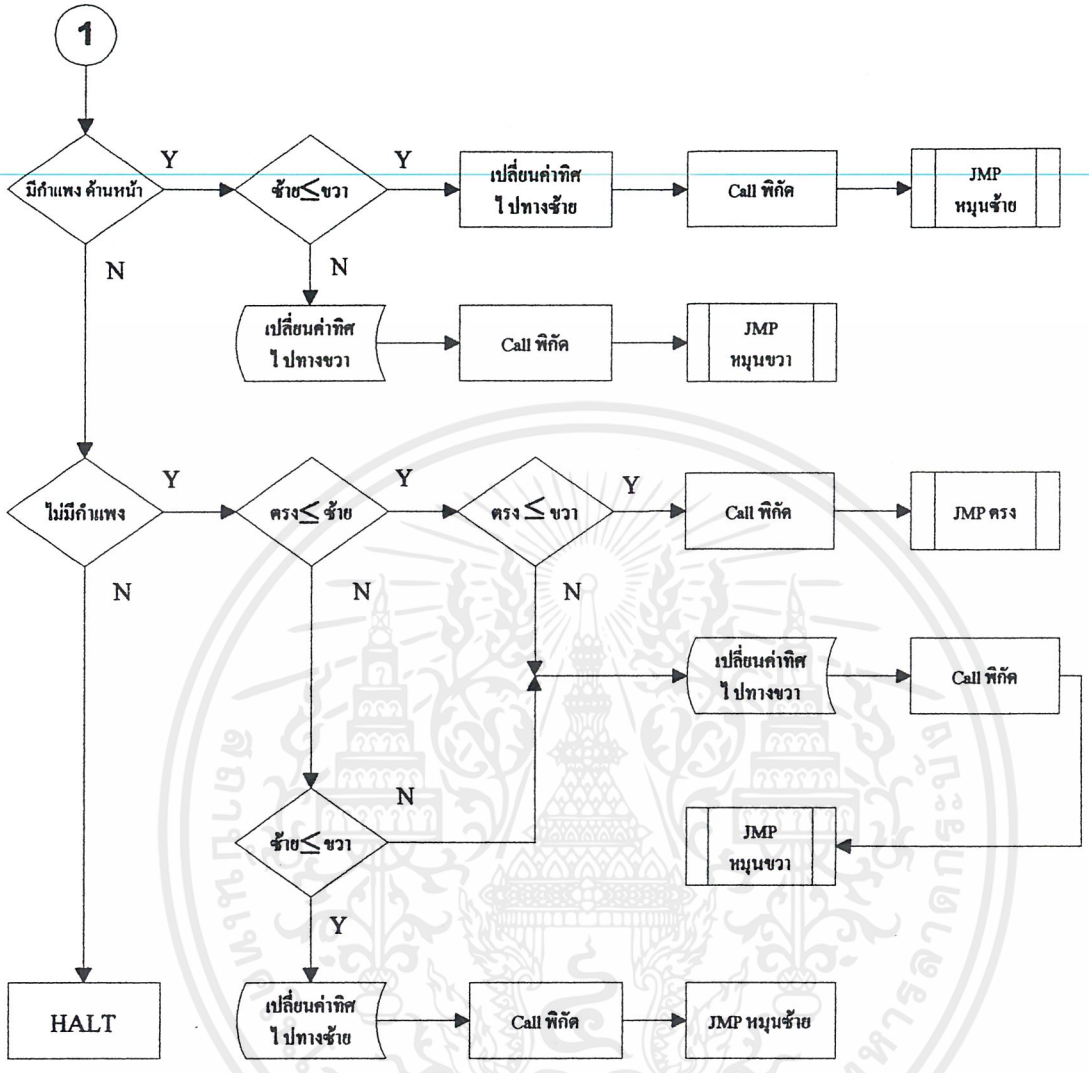

รูปที่ 3.14 แสดง Flow Chart การตรวจสอบตำแหน่งกำแพงด้านข้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



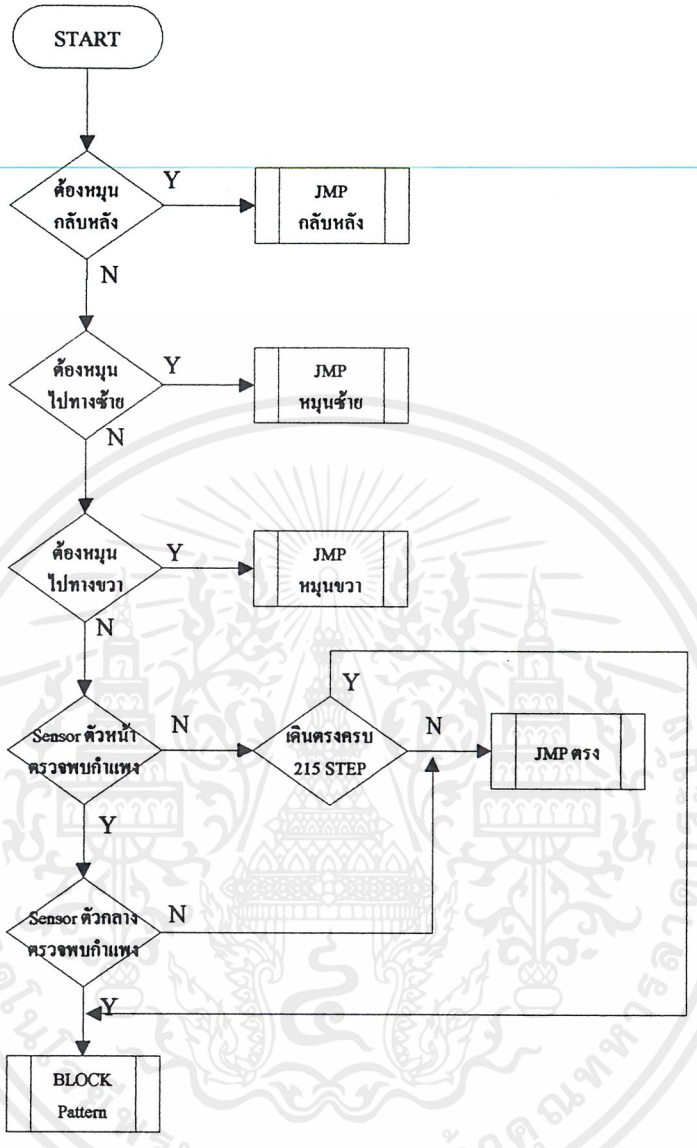

**รูปที่ 3.15 แสดง Flow Chart การทำงาน Block Pattern**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



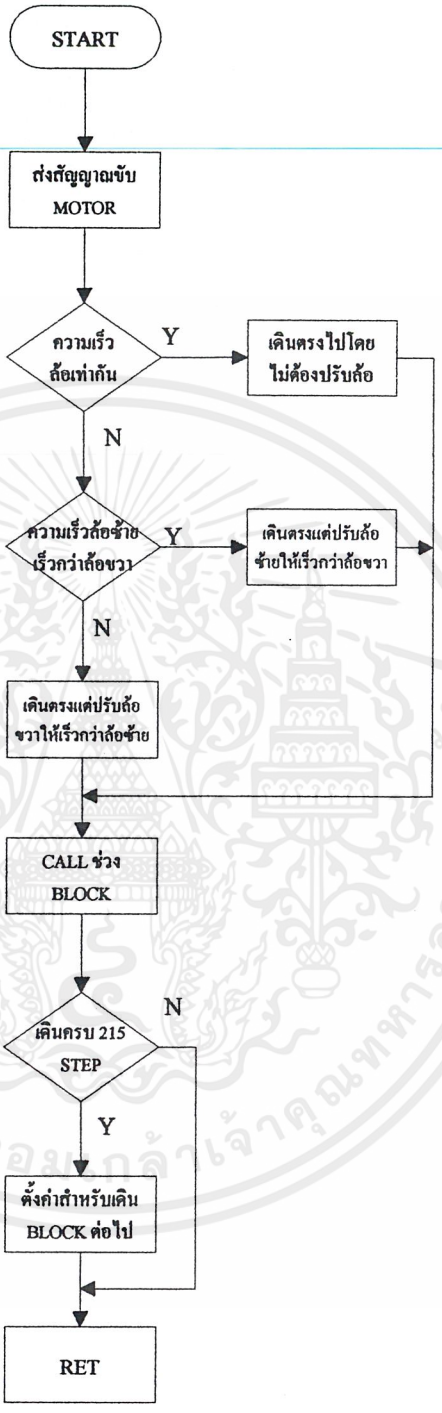

รูปที่ 3.16 แสดง Flow Chart การทำงาน Block Pattern (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



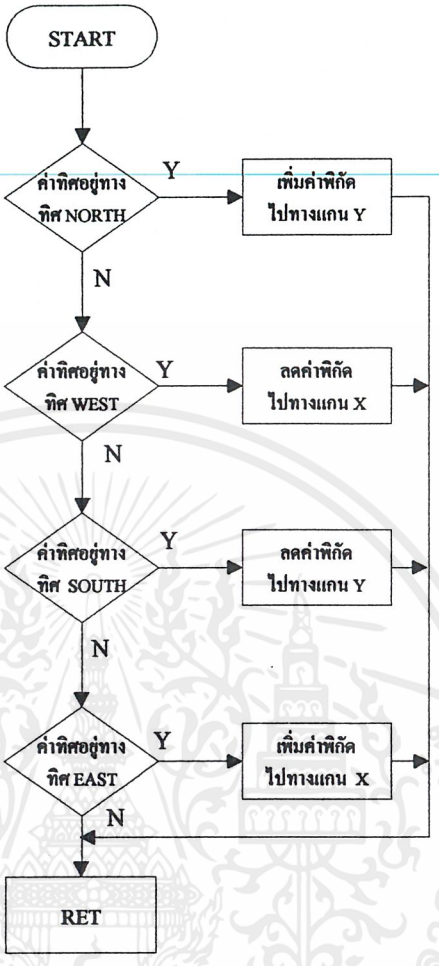

รูปที่ 3.17 แสดง Flow Chart การทำงาน T2 INT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.18 แสดง Flow Chart การเดินตรง**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3.19 แสดง Flow Chart การตั้งค่าพิกัดของไมโครเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12 จะกล่าวถึงโปรแกรมหลัก หลังจากที่ได้ตั้งค่าพารามิเตอร์ของไมโครเมาส์ที่จำเป็นต่างๆแล้ว ก็จะทำการตรวจสอบตำแหน่งไมโครเมาส์ ซึ่งการตรวจสอบตำแหน่งไมโครเมาส์ว่าเอียงหรือตรงนั้น จำเป็นต้องอาศัยกำแพงด้านข้างเป็นตัวอ้างอิง หากไม่มีกำแพงด้านข้างไว้อ้างอิงก็จำต้องยอมสั่งไมโครเมาส์ให้เคลื่อนที่ด้วยความเร็วล้อที่เท่ากันทั้งสองข้าง แต่ในกรณีที่พบกำแพงด้านข้างไมโครเมาส์จะทำการตรวจสอบตำแหน่งไมโครเมาส์ว่าอยู่ในตำแหน่งที่ตรงหรือไม่ ถ้าเกิดพบว่าไม่ตรงไมโครเมาส์ก็จะทำการชดเชยเพิ่มความเร็วล้อด้านที่เอียงให้เร็วกว่าอีกล้อ เพื่อบังคับให้ไมโครเมาส์เคลื่อนที่ด้วยตำแหน่งที่ตรงเสมอ การตรวจสอบสาวะตำแหน่งไมโครเมาส์จะทำทุกครั้งที่ไมโครเมาส์เคลื่อนที่ครบ 50 สเต็ป

เมื่อได้ทำการตรวจสอบสภาวะตำแหน่งไมโครเมาส์ไปแล้วก็จะทำการตรวจกำแพง ซึ่งเมื่อพบกำแพงด้านใดก็จะทำการรับรู้ แต่สำหรับการตรวจกำแพงด้านข้างจะตรวจกันเป็นระยะๆ โดยตรวจเฉพาะช่วงกลางบล็อกเท่านั้น

รูปที่ 3.13 เป็นการอธิบายถึงหลักการของโปรแกรมอย่างกว้างๆทั้งหมด หลังจากที่ได้เซตค่าต่างๆแล้ว ในขั้นต้นเราจะให้ไมโครเมาส์เคลื่อนที่ไปข้างหน้า ซึ่งก็เป็นการทราบพิกัดที่จะไป เมื่อได้เคลื่อนที่ไปยังพิกัดนั้นแล้ว ก็จะมีการตรวจสอบว่าพิกัดที่ได้เคลื่อนที่มาถึงนั้น คือ จุดปลายทางหรือไม่ ถ้าเป็นจุดปลายทางก็จะทำการหยุด เพื่อแสดงการรับรู้ว่าได้เคลื่อนที่มาถึงปลายทางแล้ว หากไม่ใช่จุดปลายทาง ก็จะทำการเรียกโปรแกรมย่อย Block Pattern เพื่อทำการตัดสินใจเลือกหาเส้นทางที่เหมาะสมในการเคลื่อนที่จุดครั้งต่อไป

รูปที่ 3.14 เป็นโปรแกรมย่อยที่ไว้กำหนดจุดที่อนุญาตให้เริ่มและจุดให้หยุดการตรวจสอบกำแพงด้านข้าง จาก Flow Chart จะเห็นว่าระยะตรวจสอบกำแพงด้านข้างคือ เมื่อไมโครเมาส์ได้เคลื่อนที่ได้ตั้งแต่ 50 สเต็ป จนเคลื่อนที่ครบ 170 สเต็ป

รูปที่ 3.15 แสดงถึงส่วนความคิดการตัดสินใจในการเลือกเส้นทางที่จะเคลื่อนที่ของไมโครเมาส์ จากแนวคิดที่ว่า ไมโครเมาส์จะมีโอกาสที่จะเจอกำแพงตามแยกต่างๆ ได้ 8 รูปแบบ จึงได้ทำการเขียนโปรแกรมรองรับรูปแบบเหล่านั้นไว้ทั้งหมด จากรูปจะเห็นว่า หากที่บล็อกนั้นเป็นรูปแบบของทางตัน โปรแกรมจะทำการให้ค่าความถี่ในการผ่านบล็อกนั้นมากๆ เพื่อจะได้ไม่ต้องเคลื่อนที่มายังบล็อกที่ตันอีกโดยไม่จำเป็น ส่วนรูปแบบอื่นๆ จะเพิ่มค่าความถี่ในการผ่านเพียงครั้งหนึ่งเท่านั้น

จะเห็นว่า 4 รูปแบบที่ไม่จำเป็นต้องมีการตัดสินใจว่าควรไปยังทิศทางใดดี เนื่องจาก 4 รูปแบบนี้ไม่มีเส้นทางให้เลือกได้แก่ รูปแบบทางตัน ทางทีมีแต่เลี้ยวซ้าย ทางทีมีแต่เลี้ยวขวา และทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรง ส่วนรูปแบบที่เหลือจะต้องมีการตัดสินใจ เนื่องจากมีทางให้เลือกตั้งแต่สองทางขึ้นไป โดยเกณฑ์ในการตัดสินใจจะทำโดยการเปรียบเทียบ คือจะเลือกเคลื่อนที่ไปยังเส้นทางที่ความถี่ในการผ่านน้อยที่สุด แต่ถ้าหากจำนวนความถี่ในการผ่านในแต่ทางที่ให้เลือกมีค่าเท่ากัน จะมีการให้ความสำคัญจากมากไปน้อยดังนี้

1. ทางตรง
2. ทางซ้าย
3. ทางขวา

แต่ถ้าหากกำแพงที่ไมโครเมาส์ตรวจสอบได้ ไม่ได้อยู่ในแปดรูปแบบนี้ แสดงว่าเกิดการผิดพลาดทางโปรแกรมขึ้น จะแสดงข้อผิดพลาดนี้โดย Halt

ทุกครั้งที่ Timer2 เกิดการอินเตอร์รัพต์ โปรแกรมจะทำการกระโดดมาทำงานตาม Flow Chart รูปที่ 3.17 ซึ่งมีการทำงานดังนี้

ขั้นแรกของการอินเตอร์รัพต์ โปรแกรมจะทำการตรวจสอบว่า ขณะนี้ไมโครเมาส์ได้อยู่ในช่วงหมุนตัวหรือไม่ ทั้งการหมุนตัวกลับหลังหัน หมุนตัวไปทางซ้าย และทางขวา ถ้าหากไมโครเมาส์ยังอยู่ในขั้นตอนการหมุนอยู่ ก็จะทำการหมุนต่อจนเสร็จ หลังจากตรวจสอบการหมุนแล้ว ก็จะทำการเคลื่อนที่ตรงจนครบ 215 สเต็ป ในกรณีที่ไม่มีกำแพงด้าน แต่ถ้าหากมีกำแพงด้านหน้า ก็จะทำการเคลื่อนที่ตรงจนกว่ากว่าเซ็นเซอร์ตัวกลางจะตรวจสอบได้ ซึ่งเป็นการชดเชยข้อผิดพลาดดังที่ได้กล่าวมาแล้ว

รูปที่ 3.18 เป็นการอธิบายโปรแกรมย่อยการเคลื่อนที่ตรง เริ่มแรกจะส่งสัญญาณขับมอเตอร์ทั้งสองข้าง จากนั้นจึงทำการตรวจสอบความเร็วของมอเตอร์ว่ามีลักษณะเช่นไร หากมีความเร็วเท่ากันก็ไม่มีความจำเป็นที่จะต้องปรับล้อ แต่ถ้าหากไม่เท่ากัน ก็จะปรับแต่งเพื่อชดเชยล้อข้างที่ผิดไป จากรูปจะเห็นว่าหลังการตรวจสอบความเร็ว จะมีการเรียกโปรแกรมย่อยช่วงบล็อก และเมื่อทุกๆ ครั้งที่เคลื่อนที่ครบหนึ่งบล็อก จะมีการเซตค่าพารามิเตอร์ขึ้นมาใหม่ เพื่อเตรียมการเคลื่อนที่ไปยังบล็อกครั้งต่อไป

เมื่อคนเราต้องเดินที่ทางไปยังสถานที่สลับซับซ้อน ก็จำเป็นต้องมีเครื่องมือช่วยในการบอกทิศและ ตำแหน่งที่อยู่ในปัจจุบัน ไมโครเมาส์ก็เช่นกันจากการทำงานในรูปที่ 3.19จะเป็นการโปรแกรมให้มีลักษณะคล้ายเข็มทิศคือ เมื่อไมโครเมาส์กำลังเคลื่อนที่อยู่โดยหันหน้าไปยังทิศเหนือ เราจะให้เพิ่มค่าพิกัดไปทางแกน Y หรือหันหน้าไปยังทิศตะวันตก ก็จะลดค่าพิกัดไปทางแกน X หรือ หันหน้าไปยังทิศใต้ ก็จะลดค่าไปทางแกน Y หรือ หันหน้าไปยังทิศตะวันออก ก็จะเพิ่มค่าพิกัดไปทางแกนเพิ่มค่าพิกัดไปทางแกน X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# การทดลอง

## 4.1 ภาคเซ็นเซอร์

ได้ทดลองหาค่ากระแสที่เหมาะสมแก่โฟโตไดโอด โดยการทดลองเปลี่ยนค่าความต้านทาน จนได้ค่าความต้านทานที่เหมาะสม พร้อมทั้งการปรับแต่ง $V_R$ ที่อยู่ในภาคขยายสัญญาณในออปแอมป์ ซึ่งถือเป็นการชดเชยจากการที่ตัวส่ง และตัวรับไม่ได้อยู่ในระนาบเดียวกันจริงๆ

## 4.2 ด้านโปรแกรม

### 4.2.1 การทดลองโปรแกรมรับ HEX ไฟล์

ส่วนของโปรแกรมรับ HEX ไฟล์นี้ ได้ใช้เครื่องเขียน EEPROM ในการเขียนโปรแกรมส่วนนี้ลงใน EEPROM ภายในตัวไมโครคอนโทรลเลอร์เลย เนื่องจากเราได้เลือกใช้ตัวไมโครคอนโทรลเลอร์เบอร์ 89C52 ซึ่งมี EEPROM ในตัว การเขียนโปรแกรมลงใน EEPROM นั้นก็ให้ปฏิบัติตามคู่มือของเครื่องเขียนแต่ละบริษัท เมื่อเขียนลงใน EEPROM เรียบร้อยแล้วก็นำตัวไมโครคอนโทรลเลอร์ใส่ลงในบอร์ดควบคุม แล้วทำการทดสอบโปรแกรมที่เขียนขึ้นว่าใช้การได้ หรือไม่

ในการทดลองก็ต้องเขียนโปรแกรมขึ้นมาใหม่เพื่อใช้ในการทดสอบ ในที่นี้ได้เขียนโปรแกรมไฟกระพริบ ที่ได้ต่อไว้แล้วทาง Port 1 โดยเขียนใน Editor บนคอมพิวเตอร์แล้วทำการแปลงเป็น HEX ไฟล์โดยใช้โปรแกรมอีกตัวหนึ่งเป็นตัวแปลง คือ SXA 51 ซึ่งก็จะได้โปรแกรมไฟกระพริบที่เป็น HEX ไฟล์แล้ว

ส่วนในการโหลดโปรแกรม HEX ไฟล์ที่ได้ลง RAM ของบอร์ดควบคุมนั้นจะต้องทำการเลือกจั๊มเปอร์ที่ขา EXT/IN ให้เป็น IN ก่อน เพื่อที่จะให้ไมโครคอนโทรลเลอร์ทำการ Run โปรแกรมที่อยู่ภายในตัวเอง และต่อสาย RS 232 จาก Computer มาเข้าที่ Port RS232 ของบอร์ดควบคุม จากนั้นทำการกดสวิทช์รีเซ็ตบนบอร์ดควบคุม และทำการ คัดลอกไฟล์ที่เขียนขึ้นเข้าไปไว้ใน RAM ของบอร์ดควบคุมโดยอยู่ในการทำงานบน DOS ตั้ง Mode และทำการคัดลอกดังนี้

C:\> mode com1 96,n,8,1,p ↵

C:\> copy ชื่อไฟล์.HEX com1 ↵

โปรแกรมที่เป็น HEX ไฟล์ก็จะถูกโหลดเข้าไปไว้ใน RAM ภายในเรียบร้อยจากนั้นก็ทำการทดลองบนบอร์ดควบคุม โปรแกรมไฟกระพริบก็ทำงานได้ถูกต้อง ซึ่งก็แสดงว่า โปรแกรมรับ HEX ไฟล์ที่เขียนขึ้นนี้ใช้ได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**4.2.2 การทดลองโปรแกรมการปรับล้อ**

การทดลองโปรแกรมในส่วนนี้นั้น ก็จะต้องอาศัยโปรแกรมรับ HEX ไฟล์ ในการที่จะนำข้อมูล หรือโปรแกรมในส่วนนี้เข้าไปเก็บไว้ใน RAM และการทดลองก็จะทดลองโดยการ RUN โปรแกรมที่อยู่ใน RAM ซึ่งขั้นแรกจะเขียนโปรแกรมในการปรับล้อโดยใช้เซ็นเซอร์เพียงด้านเดียวก่อนโดย เซ็นเซอร์อีกด้านหนึ่งจะยังไม่สนใจ ในการทดลองก็จะใช้กำแพงเพียงด้านเดียวในการทดลอง โดยจะตั้งให้กำแพงเรียงตรงกันก่อน แล้วให้ไมโครเมาส์ทดลองวิ่ง ดูว่าสามารถวิ่งตรงโดยไม่ชนกำแพงหรือไม่ออกนอกทิศทางหรือไม่ ทดลอง และแก้ไขโปรแกรมจนได้ตามต้องการ

ทดลองขั้นต่อไป จะตั้งกำแพงโดยให้กำแพงมีความโค้งเล็กน้อย แล้วให้ไมโครเมาส์ทดลองวิ่งแล้วดูว่าไมโครเมาส์สามารถวิ่งเกาะกำแพงไปได้โดยไม่ชนกำแพง และไม่ออกนอกทิศทาง หรือไม่ ทดลองจนได้โปรแกรมตามต้องการ

และทำการเขียนโปรแกรมใหม่โดยใช้เซ็นเซอร์อีกด้านหนึ่งในการตรวจจับแทน และทำการทดลองตามขั้นตอนข้างต้นจนได้โปรแกรมตามต้องการ จากนั้นก็จะเขียนโปรแกรมโดยใช้เซ็นเซอร์ทั้งสองด้านในการตรวจจับ และการทดลองก็จะตั้งกำแพงให้มีทั้งสองด้าน โดยจะให้วิ่งตรงอย่างเดียวก่อน แล้วทดลองว่าไมโครเมาส์วิ่งได้โดยไม่ชนกำแพงหรือไม่ ทำการแก้ไขโปรแกรมจนได้ตามต้องการก็เสร็จสิ้นในส่วนของโปรแกรมปรับล้อ

**4.2.3 การทดลองโปรแกรมโดยรวมของไมโครเมาส์**

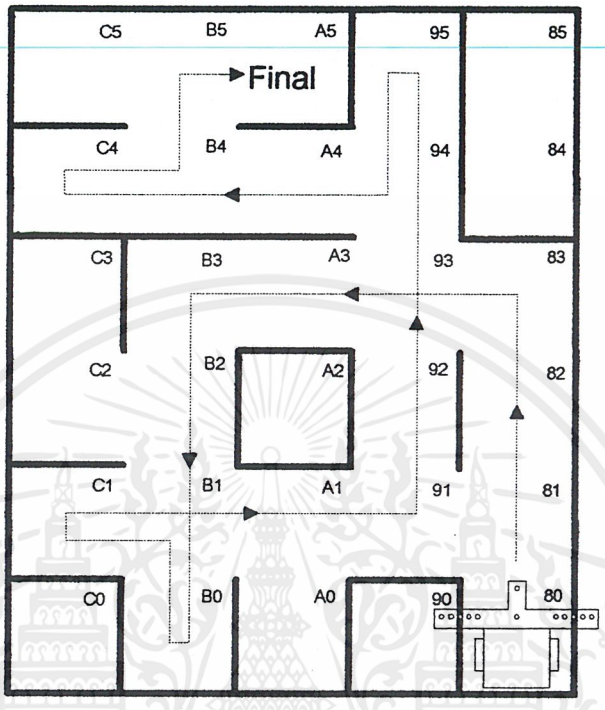
ซึ่งจะเป็นการนำโปรแกรมการปรับล้อ และการตัดสินใจของไมโครเมาส์เข้าไว้ด้วยกันซึ่งในการทดลองขั้นต้นก็จะสร้างสนามที่ง่ายๆก่อน แล้วเขียนโปรแกรมให้ไมโครเมาส์ทดลองวิ่ง ว่าสามารถวิ่งได้ทั่วทุกช่องของสนามหรือไม่ ซึ่งในการเขียนโปรแกรมขั้นต้นจะต้องให้ไมโครเมาส์สามารถวิ่งได้ทั่วสนามก่อน ซึ่งถ้าไมโครเมาส์สามารถวิ่งได้ทั่วสนามนั้นก็แสดงว่าไมโครเมาส์ก็สามารถที่จะหาทางออกหรือจุดสิ้นสุดได้

จากนั้นก็จะเพิ่มความยากของสนามขึ้นเรื่อยๆ และพัฒนาโปรแกรมจนไมโครเมาส์สามารถที่จะเดินได้ทั่วสนาม โดยไม่ว่าสนามจะมีลักษณะอย่างไร จนได้โปรแกรมที่ดีที่สุด

ขั้นสุดท้ายก็จะกำหนดทางออกหรือจุดสิ้นสุด แล้วให้ไมโครเมาส์ทดลองวิ่งหาทางออก หรือจุดสิ้นสุดที่กำหนดให้ได้ แล้วพัฒนาโปรแกรมโดยให้ไมโครเมาส์ วิ่งหาทางออกหรือจุดสิ้นสุดให้เร็วที่สุดตามต้องการ ซึ่งในการทดลองโปรแกรมที่เขียนขึ้นก็ได้ผลการทดลองตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองโปรแกรมของไมโครเมาส์โดยวิ่งในเขาวงกต



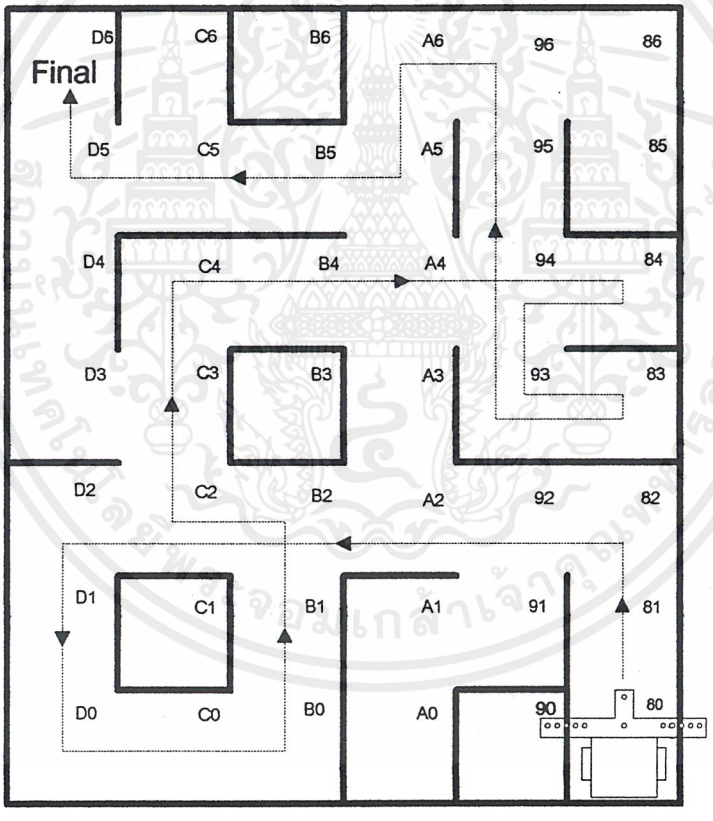

รูปที่ 4.1 แสดงไมโครเมาส์วิ่งในเขาวงกตแบบที่ 1

จากการทดลองให้ไมโครเมาส์วิ่งในเขาวงกตลักษณะดังรูปที่ 4.1 โดยมีจุดสิ้นสุดที่ตำแหน่งพิกัดหน่วยความจำ A5 ก็จะป้อนตำแหน่งพิกัด ให้โปรแกรมเมื่อไมโครเมาส์วิ่งไปในเขาวงกตก็จะทำการเพิ่มพิกัดดังกล่าวเอง เพื่อให้ทราบตำแหน่งที่ตัวไมโครเมาส์อยู่ในขณะนั้น และจะทำการเก็บค่าพิกัดว่าเคยมาตำแหน่งนั้นแล้ว เพื่อจะได้วิ่งซ้ำมาที่เดิมอีกถ้าไม่จำเป็น ส่วนจุดเริ่มต้นจะต้องอยู่มุมใด มุมหนึ่งของเขาวงกต โดยต้องชิดทางด้านขวาของเขาวงกต และหันหน้าไปทางทิศเหนือดังรูปที่ 4.1

จากการทดลองรูปแบบเขาวงกตตามรูปที่ 4.1 ไมโครเมาส์จะวิ่งตรงไปที่พิกัด 83 ซึ่งเราจะให้ความสำคัญทางด้านตรงสำคัญที่สุด จากนั้นจะเลี้ยวซ้ายแล้ววิ่งตรงไปที่พิกัด B3 แล้วเลี้ยวซ้ายวิ่งตรงไปที่พิกัด B0 เนื่องจากเป็นทางตันไมโครเมาส์จะหมุนกลับหลัง แล้ววิ่งมาที่พิกัด B1 ซึ่งเป็นทางแยกแต่เนื่องจาก B2 ซึ่งเป็นพิกัดที่วิ่งตรงเราได้ให้ความสำคัญมากกว่าแต่ว่าพิกัด B2 ไมโครเมาส์ได้เคยวิ่งผ่านมาแล้ว 1 ครั้งเมื่อเปรียบเทียบกับพิกัด C1 และ A1ซึ่งไมโครเมาส์ยังไม่เคยวิ่งไป ไมโครเมาส์ก็จะไม่วิ่งตรงไปที่พิกัด B2 แต่จะเลี้ยวซ้ายไปที่พิกัด C1 แทน เพราะว่าเราได้ให้ความสำคัญทางซ้ายมากกว่า เมื่อมาถึงพิกัด C1 ซึ่งเป็นทางตันไมโครเมาส์จะหมุนกลับหลัง แล้ววิ่งตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไปที่พิกัด 91 เป็นทางแยกแล้วทำการเปรียบเทียบความสำคัญ และทางที่เคยมาแล้ว ซึ่งพิกัด 81 สำคัญกว่าแต่เคยมาแล้ว ไมโครเมาส์ก็จะเลี้ยวซ้าย แล้ววิ่งตรงไปที่พิกัด 95 แต่เป็นทางตันก็จะหมุนกลับหลัง แล้ววิ่งมาที่พิกัด 94 ทำการเปรียบเทียบความสำคัญและจำนวนครั้งที่ได้เคยผ่านมาแล้ว ไมโครเมาส์ก็จะเลี้ยวขวา และวิ่งตรงไปที่พิกัด C4 ซึ่งเป็นทางตันก็จะหมุนกลับหลังวิ่งมาที่พิกัด B4 ทำการเปรียบเทียบความสำคัญ และจำนวนครั้งที่ได้เคยผ่านมาแล้ว ไมโครเมาส์ก็จะเลี้ยวซ้ายวิ่งไปที่พิกัด B5 เป็นทางแยกทำการเปรียบเทียบความสำคัญและจำนวนครั้งที่ได้เคยผ่านมาแล้ว ซึ่งพิกัด C5 เป็นพิกัดที่สำคัญกว่า และยังไม่เคยมาแต่ไมโครเมาส์จะไม่เลี้ยวซ้าย จะเลี้ยวขวาแทน เนื่องจากพิกัด A5 เป็นพิกัดสิ้นสุด เนื่องจากไมโครเมาส์จะทำการเปรียบเทียบหาจุดสิ้นสุดทุกครั้งที่เดินไปเจอทางแยก ถ้าพบจุดสิ้นสุดก็จะวิ่งไปทันที เพราะฉะนั้นไมโครเมาส์จึงเลี้ยวขวาไปยังพิกัด A5 แล้วหยุด




รูปที่ 4.2 แสดงไมโครเมาส์วิ่งในเขาวงกตแบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองให้ไมโครเมาส์วิ่งในเขาวงกตที่มีขนาดเพิ่มขึ้นดังรูปที่ 4.2 ซึ่งมีจุดสิ้นสุดที่พิกัด D6

เมื่อทำการทดลองวิ่งไมโครเมาส์ ไมโครเมาส์จะวิ่งไปที่พิกัด 82 แล้วเลี้ยวซ้าย วิ่งตรงไปที่พิกัด D2 แล้วเลี้ยวซ้ายวิ่งตรงไปที่พิกัด D0 เลี้ยวซ้าย วิ่งตรงไปที่พิกัด B0 เลี้ยวซ้ายวิ่งตรงไปที่พิกัด B2 ทำการเปรียบเทียบความสำคัญ และจำนวนครั้งที่ได้เคยผ่านมาแล้ว เป็นจุดสิ้นสุด หรือไม่ ไมโครเมาส์จะเลี้ยวซ้ายซึ่งมีความสำคัญมากกว่า แล้ววิ่งตรงไปที่พิกัด C2 ทำการเปรียบเทียบความสำคัญ และจำนวนครั้งที่ได้เคยผ่านมาแล้ว เป็นจุดสิ้นสุดหรือไม่ แล้วเลี้ยวขวาวิ่งไปที่พิกัด C4 เลี้ยวขวาวิ่งตรงไปที่พิกัด 84 เป็นทางตันจึงหมุนกลับหลังวิ่งไปที่พิกัด 94 ทำการเปรียบเทียบความสำคัญ เลี้ยวซ้ายวิ่งไปที่พิกัด 93 เลี้ยวซ้ายวิ่งไปที่พิกัด 83 เป็นทางตันจึงหมุนกลับหลังวิ่งไปที่พิกัด 93 เลี้ยวขวาวิ่งไปที่พิกัด 96 ทำการเปรียบเทียบความสำคัญ แล้วเลี้ยวซ้ายวิ่งไปที่พิกัด A6 เลี้ยวซ้ายวิ่งไปที่พิกัด A5 ทำการเปรียบเทียบความสำคัญเลี้ยวขวาวิ่งตรงไปที่พิกัด D5 ทำการเปรียบเทียบความสำคัญ ซึ่งเจอจุดสิ้นสุดแล้วก็จะเลี้ยวขวาวิ่งไปที่พิกัด D6 ซึ่งเป็นจุดสิ้นสุด ไมโครเมาส์จึงหยุดเดิน

จากการทดลองได้ยกตัวอย่างเขาวงกตที่มีลักษณะแตกต่างกันเพียง 2 ลักษณะเท่านั้นแต่ในการทดลองไมโครเมาส์วิ่งในเขาวงกตที่มีลักษณะอื่นๆอีก ไมโครเมาส์ก็สามารถหาจุดสิ้นสุดได้และสามารถทำงานตามโปรแกรมที่ได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# วิจารณ์ และสรุป

ไมโครเมาส์นั้นจะประกอบด้วย ส่วนของตัวรถไมโครเมาส์ และโปรแกรมควบคุม ซึ่งในส่วนของตัวรถไมโครเมาส์ จะต้องให้มีขนาดเหมาะสมกับขนาดของเส้นทางที่จะให้ไมโครเมาส์เคลื่อนที่ไป เนื่องจากไมโครเมาส์จะต้องทำการตรวจสอบเส้นทางที่จะเคลื่อนที่ไปว่ามีลักษณะอย่างไร แล้วนำข้อมูลที่ตรวจสอบได้มาทำการวิเคราะห์ และตัดสินใจ ในการที่จะเคลื่อนที่ไปทิศทางใดเพื่อที่จะได้หาจุดสิ้นสุดที่ต้องการได้ ถ้าตัวรถมีขนาดที่ไม่เหมาะสมก็อาจจะทำให้การเคลื่อนที่ไปในเส้นทางเกิดการผิดพลาดได้

ส่วนของโปรแกรมควบคุมก็มีความสำคัญเช่นเดียวกัน ซึ่งจะต้องเขียนโปรแกรมให้ไมโครเมาส์สามารถตัดสินใจ ในการที่จะเลือกเคลื่อนที่ไปในทิศทางใด เพื่อที่จะหาจุดสิ้นสุดให้ได้ และโปรแกรมยังต้องทำการสั่งงานให้ไมโครเมาส์เคลื่อนที่ให้ตรงทิศทางตลอดเวลา ไม่เช่นนั้นไมโครเมาส์ก็ไม่สามารถที่ไปได้

ดังนั้นในการสร้างไมโครเมาส์จะต้องให้ทั้งสองส่วนคือ ส่วนของตัวรถไมโครเมาส์ และโปรแกรมควบคุมเหมาะสมกัน ถ้าส่วนหนึ่งส่วนใดมีปัญหา ก็จะทำให้ไมโครเมาส์ไม่สามารถที่จะเคลื่อนที่ไปในเส้นทาง และหาจุดสิ้นสุดที่ต้องการได้

## 5.1 ปัญหา และอุปสรรคที่พบ

1. เซ็นเซอร์ เกิดจากแสงจากภายนอกมารบกวนทำให้การตรวจจับเกิดการผิดเพี้ยนได้
2. การปรับความไวของเซ็นเซอร์ให้มีค่าที่เหมาะสมทำได้ยาก
3. การปรับแต่งล้อพยุงด้านหน้าควรปรับแต่งให้ล้อมีความสูงจากพื้นเล็กน้อย เพื่อทำให้การเลี้ยว และ การหมุนกลับของรถมีความเที่ยงตรงขึ้น
4. ล้อที่ใช้ในการขับเคลื่อน ต้องใส่ยางที่ล้อ ให้ล้อมีความฝืด

## 5.2 แนวทางในการพัฒนา

1. ให้พยามหาสเต็ปปิ้งมอเตอร์ที่มีแรงบิดมาก ขณะที่ขนาดพอเหมาะกับการเคลื่อนที่ของไมโครเมาส์
2. เมื่อได้สเต็ปปิ้งมอเตอร์ที่ดี และรับน้ำหนักได้มากขึ้นแล้ว ควรเปลี่ยนแหล่งจ่ายไฟจากภายนอก เป็นแบตเตอรี่ที่ไมโครเมาส์บรรทุกได้
3. ให้ออกแบบวงจรขับเซ็นเซอร์ที่มีเสถียรภาพดีกว่านี้
4. พัฒนาโปรแกรมไมโครเมาส์ให้ทำงานได้มีประสิทธิภาพได้ดี และฉลาดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม


1. สุนทร วิทูสุรพจน์, "การใช้งานไมโครคอนโทรลเลอร์ตระกูล 8051",บริษัท ซีเอ็ดยูเคชั่น จำกัด (มหาชน)

2. พิพัฒน์ เลาหสงคราม, "ไมโครคอนโทรลเลอร์ MCS-48, MCS-51",สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3. ชัยวัฒน์ ลิ้มพรจิตรวิไล, "MC-2 บอร์ดควบคุมสเต็ปปิ้งมอเตอร์ผ่านคอมพิวเตอร์",วารสารเซมิคอนดักเตอร์อิเลคทรอนิกส์,ฉบับที่ 175,กันยายน 2540,หน้า 201-211

4. สมยศ จุณณะปิยะ, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ ตระกูล MCS-51",สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2541

5. NP Micromouse Kit Technical Manual ( Revision 2, Jan'96) Alpha Centre, Ngee, Ann Polytecnic, Singapore



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**MicroMouse Contest Rules**

I. Maze Specifications

The maze shall comprise 16 x 16 multiples of an 18 cm x 18 cm unit square. The walls constituting the maze shall be 5 cm high and 1.2 cm thick. Passageways between the walls shall be 16.8 cm wide. The outside wall shall enclose the entire maze.

The sides of the maze shall be white, and the top of the walls shall be red. The floor of the maze shall be made of wood and finished with a non-gloss black paint. The coating on the top and sides of the walls shall be selected to reflect infred-red light and the coating on the floor shall absorb it.

The start of the maze shall be located at one of the four corners. The starting square shall have walls on three sides. The starting square orientation shall be such that when the open wall is to the "north", outside maze walls are on the "west" and "south". At the center of the maze shall be a large opening which is composed of 4 unit squares. This central square shall be the destination. A red post, 20 cm high, and 2.5 cm on each side, may be placed at the center of the large destination square if requested by the handler.

Small square posts, each 1.2 cm x 1.2 cm x 5 cm high, at the four corners of each unit are called latice points. The maze shall be constituted such that there is at least one wall touching each lattice point, except for the destination square.

The dimensions of the maze shall be accurate to within 5% or 2 cm, whichever is less. Assembly joints on the maze floor shall not incolve steps greater than 0.5 mm. The change of slope at an assembly joint shall not be greater than 4. Gaps between the walls of adjacent squares shal not be greater than 1 mm.

II. MicroMouse Specifications

A Micro Mouse shall be self contained. It shall not use an energy source employing a combustion process.

The length and width of a Micro Mouse shall be restricted to a square region of 25 cm x 25 cm. The dimensions of a Micro Mouse which changes its geometry during a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

run shall never be greater than 25 cm x 25 cm. The height of a Micro Mouse is unrestricted.

A Micro Mouse shall not leave anything behind while negotiating the maze.

A Micro Mouse shall not jump over, climb, scratch, damage, or destroy the walls of the maze.

III. Contest Rules

The basic function of a Micro Mouse is to travel from the start square to the destination square. This is called a run. The time it takes is called the run time. Traveling from the destination square back to the start square is not considered a run. The total time from the first activation of the Micro Mouse until the start of each run is also mesured. This is called the maze time. If a mouse requires manual assistance at any time during the contest it is considered touched. By using these three parameters the scoring of the contest is designed to reward speed, efficiency of maze solving, and self-reliance of the Micro Mouse.

The scoring of a Micro Mouse shall be done by computing a handicapped time for each run. This shall be calculated by adding the time for each run to 30 of the maze time associated with that run and subtracting a 10 second bonus if the Micro Mouse has not been touched yet1. 1 For example assume a Micro Mouse, after being on the maze for 4 minutes without being touched, starts a run which takes 20 seconds; the run will have a handicapped time of: 20+4300-10=18seconds The run with the fastest handicapped time for each Micro Mouse shall be the offical time of that Micro Mouse.

Each contesting Micro Mouse shall be subject to a time limit of 15 minutes on the maze. Within this time limit, the Micro Mouse may make as many runs as possible. When the Micro Mouse reaches the maze center it may be manually lifted out and restarted or it may make its own way back to the start square. Manually lifting it out shall be considered touching the Micro Mouse and will cause it to loose the 10 second bonus on all further runs.

The time for each run shall be measured from the moment the Micro Mouse leaves the start square until it enters the finish square. The total time on the maze shall be

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

measured from the time the Micro Mouse is first activated. The mouse does not have to move when it is first activated but it must be positioned in the start square ready to run.

The time taken to negotiate the maze shall be measured either manually by the contest officials or by infra-red sensors set at the start and destination. If infra-red sensors are used, the start sensor shall be positioned at the boundary between the start square and the next unit square. The destination sensor shall be placed at the entrance to the destination square. The infra-red beam of each sensor shall be horizontal and positioned approximately 1 cm above the floor.

The starting procedure of the Micro Mouse shall not offer a choice of strategies to the handler.

Once the maze configuration for the contest is disclosed, the operator shall not feed the Micro Mouse with any maze information.

The illumination, temperature, and humidity of the room in which the maze is located shall be those of an ambient enviornment. Requests to adjust the illumination may be accepted at the discretion of the contest officials.

If a Micro Mouse appears to be malfunctioning, the handlers may ask the judges for permission to abandon the run and restart the Micro Mouse at the beginning. A Micro Mouse shall not be re-started merely because it has taken a wrong turn.

If a Micro Mouse team elects to stop because of technical problems, the judges may, at their discretion, permit the team to run again later in the contest with a 3 minute maze time penalty1. 1 For example, assume a Micro Mouse is stopped after 4 minutes; it must be restarted as if it had already run for 7 minutes, and will have only 8 more minutes to run.

If any part of a Micro Mouse is replaced during its performance, such as batteries or EPROMS, or if any significant adjustment is made, the memory of the maze within the Micro Mouse shall be erased before restarting. Slight adjustments, such as to the sensors may be allowed at the discretion of the judges, but operation of speed or strategy controls is expressly forbidden without a memory erasure.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

No part of the Micro Mouse (with the possible excpetion of batteries) shall be transferred to another Micro Mouse. For example if one chassis is used with two alternative controllers, then they are the same Micro Mouse and must perform within a single 15 minute allocation. The memory must be cleared with the change of a controller.

The contest officials shall reserve the right to stop a run, or disqualify a Micro Mouse, if they believe its continued operation is endangering the condition of the maze. Thanks to the University of Rochester for these rules adapted from the IEEE.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## โปรแกรมรับข้อมูล

```
                ORG 000H
                LJMP START
                ORG 100H

START:  MOV P1,#00H
                MOV R2,#0FFH
LOOP:           MOV R3,#0FFH
                DJNZ R3,$
                DJNZ R2,LOOP
                MOV P1,#0FFH
                MOV A,#0FDH
                MOV TH1,A
                MOV TL1,A
                MOV TMOD,#00100001B
                SETB TR1
                MOV SCON,#01010000B
RE_RX:  CLR TI
                CLR RI
REC:            JNB RI,$
                MOV A,SBUF
                CLR RI
                CJNE A,#3AH,REC
                CALL DECODE_CH
                MOV R6,A
                CALL DECODE_CH
                MOV DPH,A
                CALL DECODE_CH
                MOV R7,DPL
                MOV DPL,A
                CALL DECODE_CH
                CJNE A,#1H,READ_DATA
                AJMP RECEIVE
```

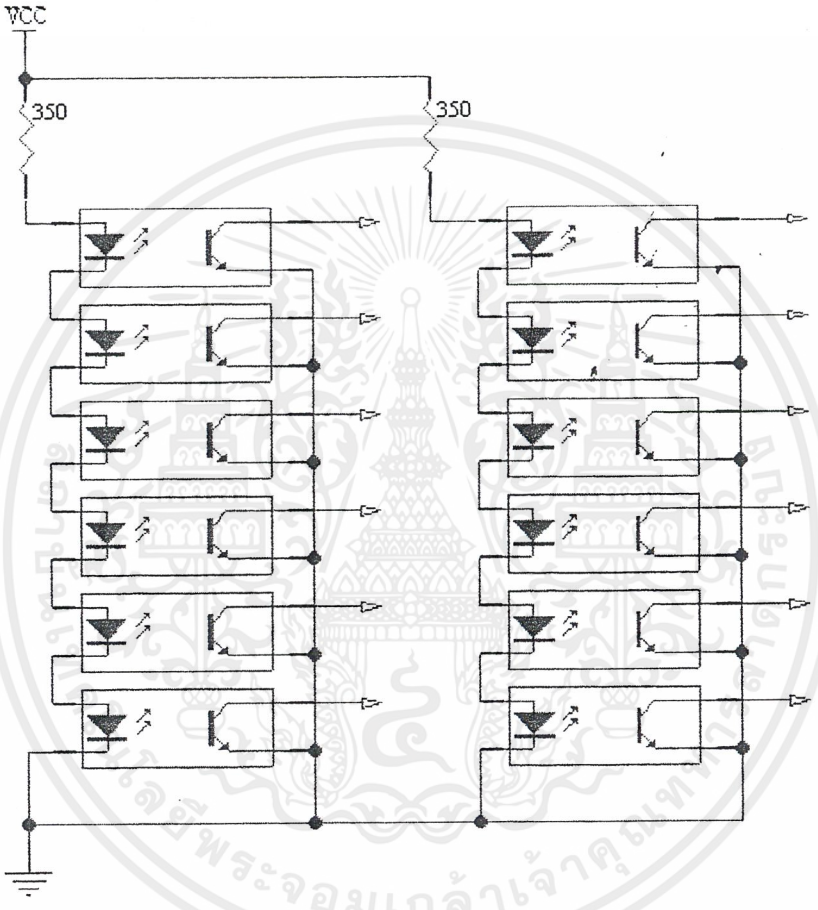
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
READ_DATA:  INC R6
READ_T:     DJNZ R6,READ
            AJMP CHECK_SUM
READ:       CALL DECODE_CH
            MOVX @DPTR,A
            MOV P1,A
            INC DPL
            MOV R5,DPL
            CJNE R5,#0H,INC_DPH
            INC DPH
INC_DPH:    AJMP READ_T
CHECK_SUM:  CALL DECODE_CH
            AJMP REC

            ;=============================================
            ;******* DECODE CHARECTER TO HEXIMAL ******
            ;=============================================

                        ;translate ascii is hex

DECODE_CH:  CLR RI
            JNB RI,$
            MOV A,SBUF
            CLR RI
            CALL DEC_CHAR
            MOV R4,A
            JNB RI,$
            MOV A,SBUF
            CLR RI
            CALL DEC_CHAR
            MOV R3,A
            MOV A,R4
            SWAP A
            ORL A,R3
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

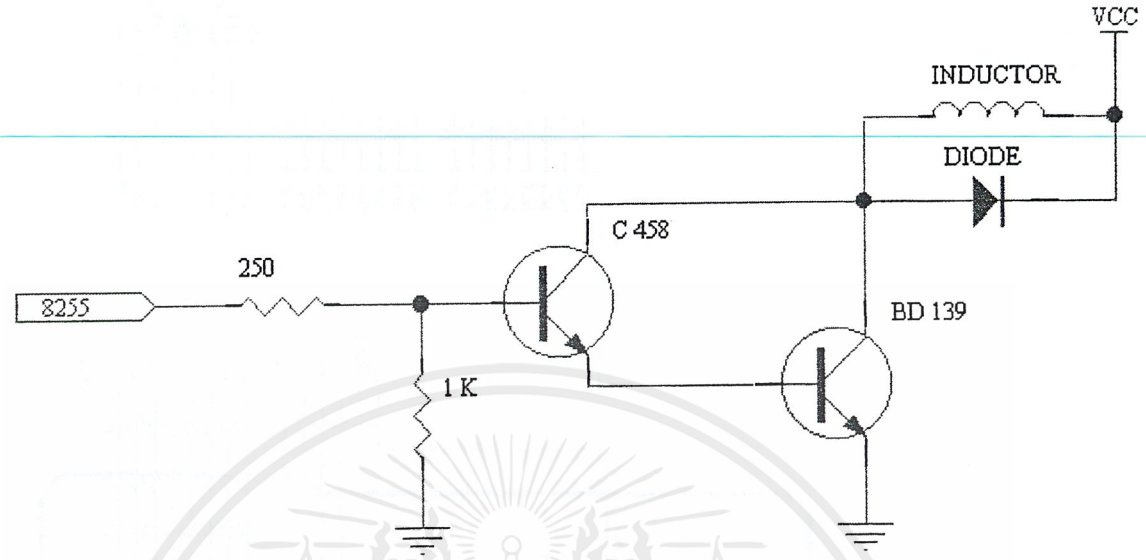
```
DEC_CHAR:   CLR C
            SUBB A,#30H
            MOV R2,A
            CLR C
            SUBB A,#0AH
            JC EXIT_DEC
            MOV A,R2
            SUBB A,#7H
            RET
EXIT_DEC:   MOV A,R2
            RET
RECEIVE:    MOV A,R7
            CPL A
SHOW:       MOV P1,A
            AJMP SHOW
            END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

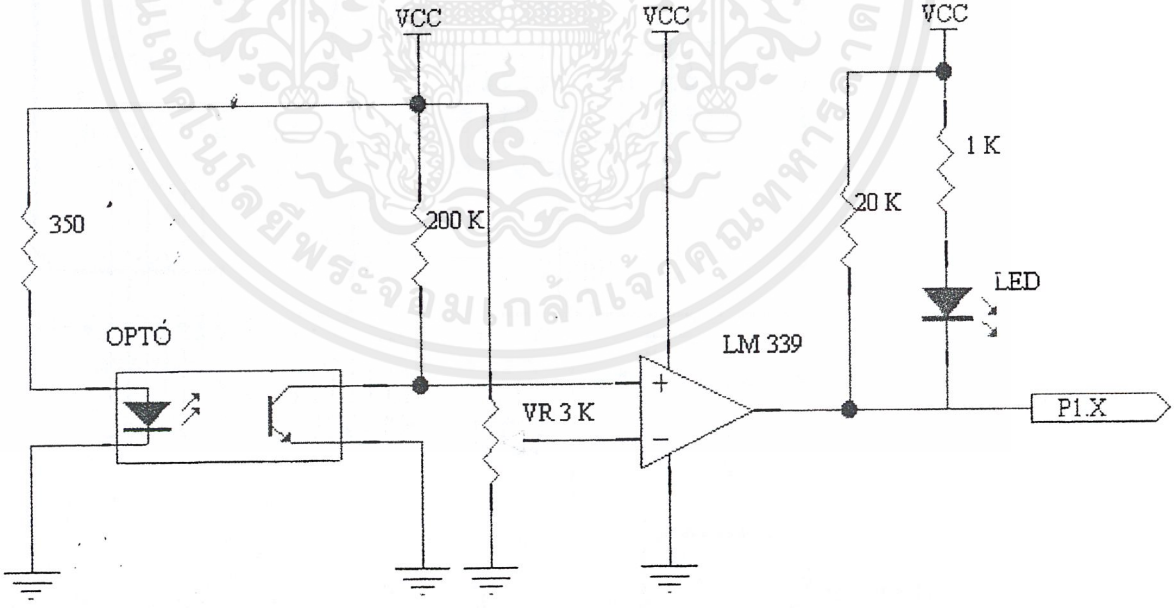



**วงจรแสดงการต่อวงจรโฟโตไดโอด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

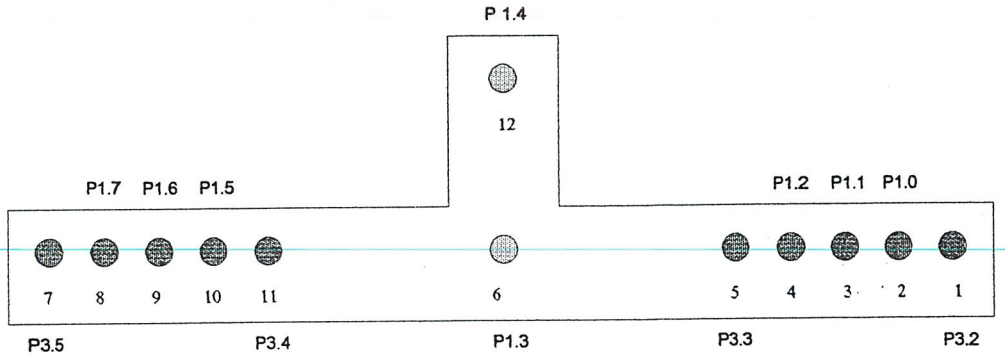



**รูปวงจรแสดงการต่อทรานซิสเตอร์ขับมอเตอร์**

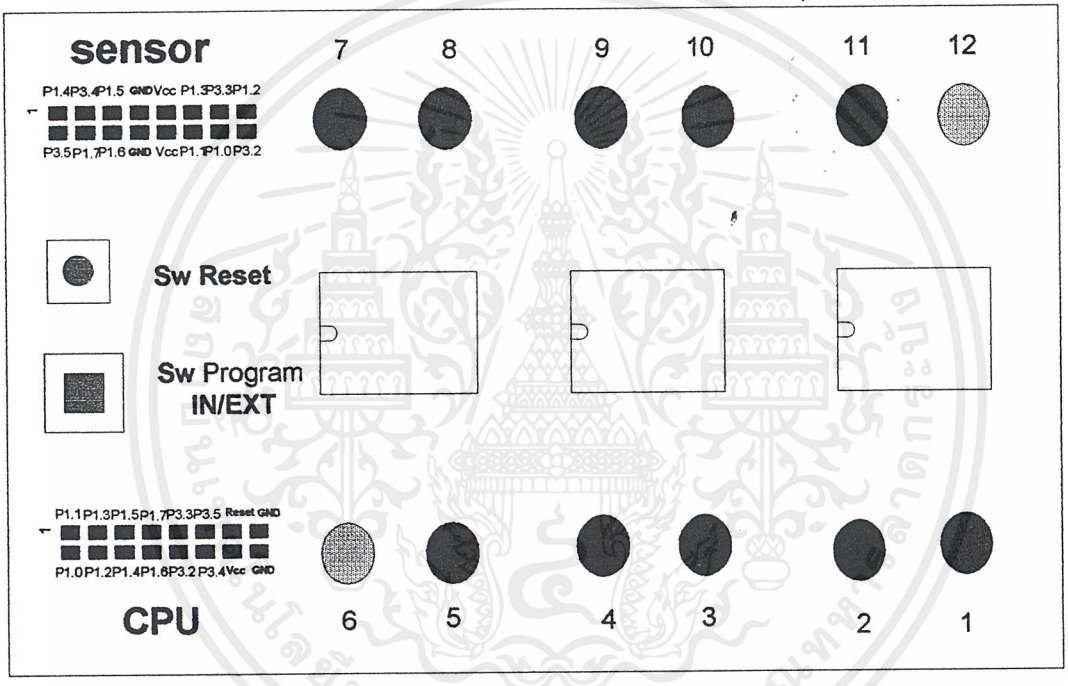



**รูปวงจรแสดงการต่อออฟแอมป์ขับโฟโต้ไดโอด**

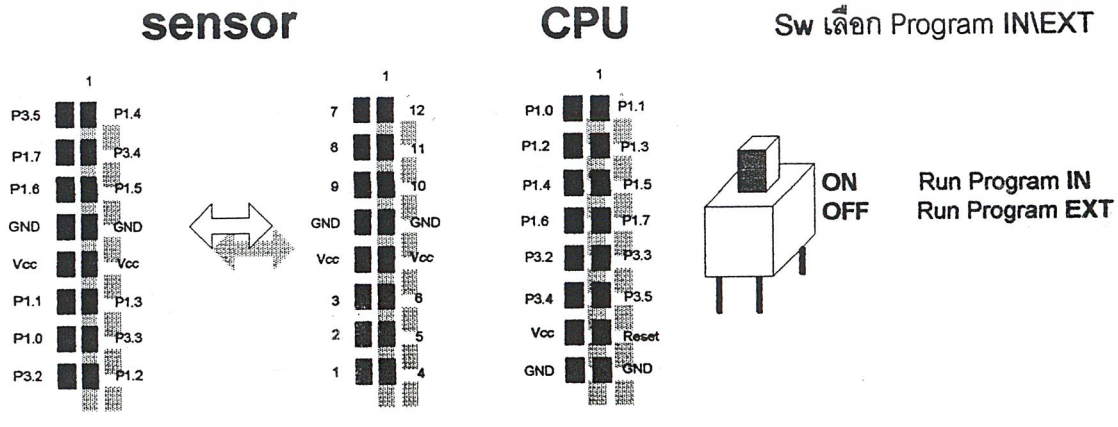
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



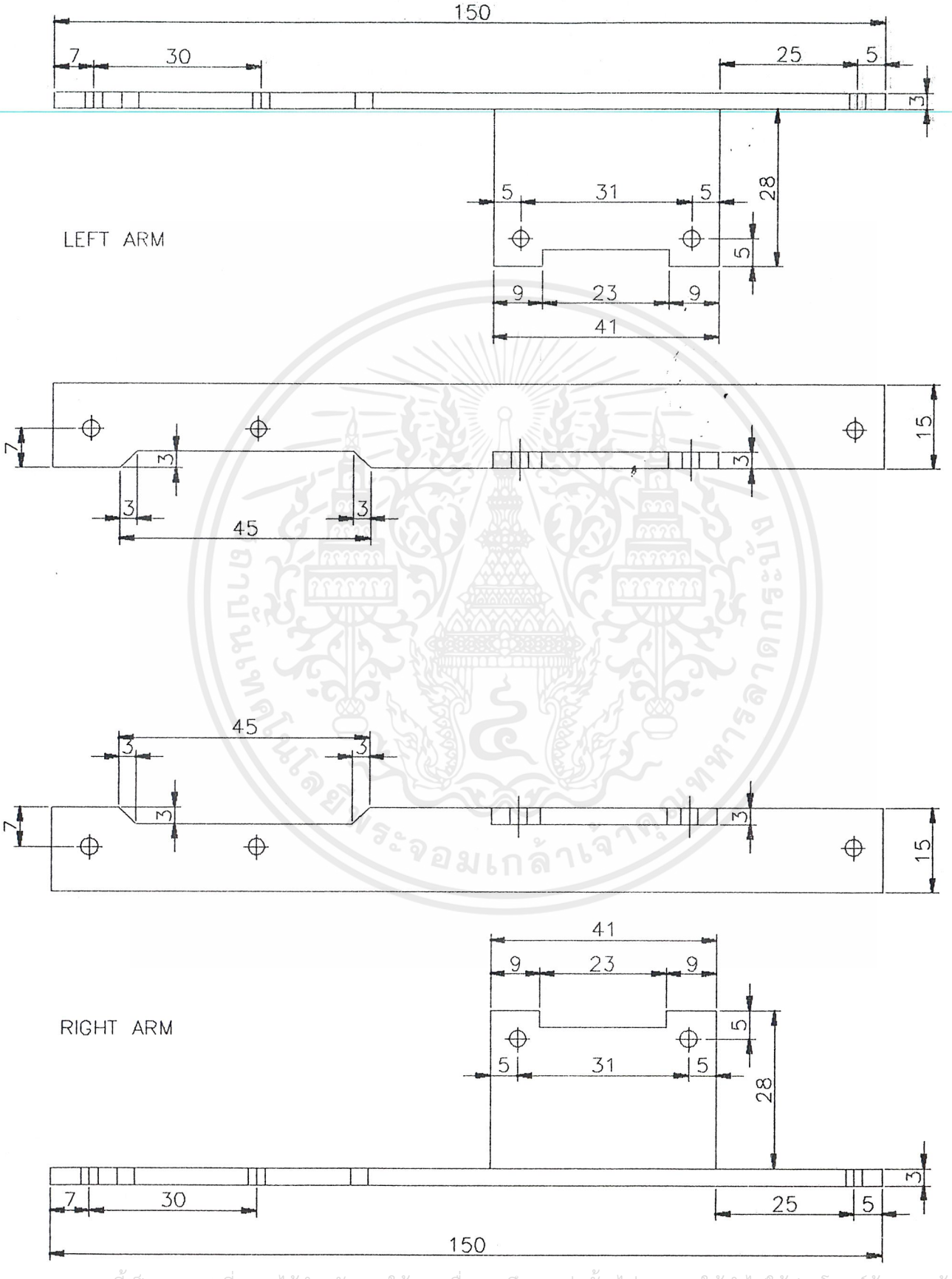

แสดงตำแหน่งการต่อ Port CPU กับ Sensor




แสดงตำแหน่ง LED บนแผ่นวงจรสัมพันธ์ กับ ตำแหน่ง Sensor

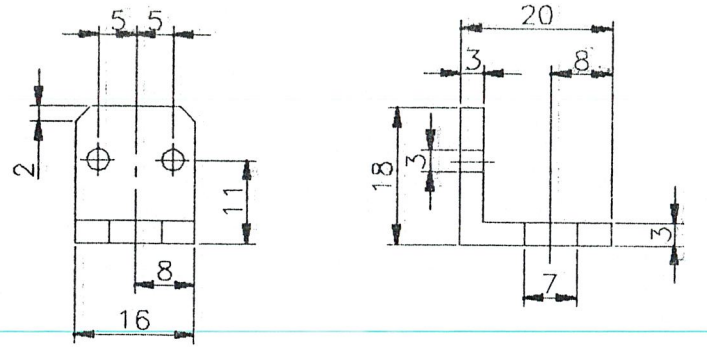



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

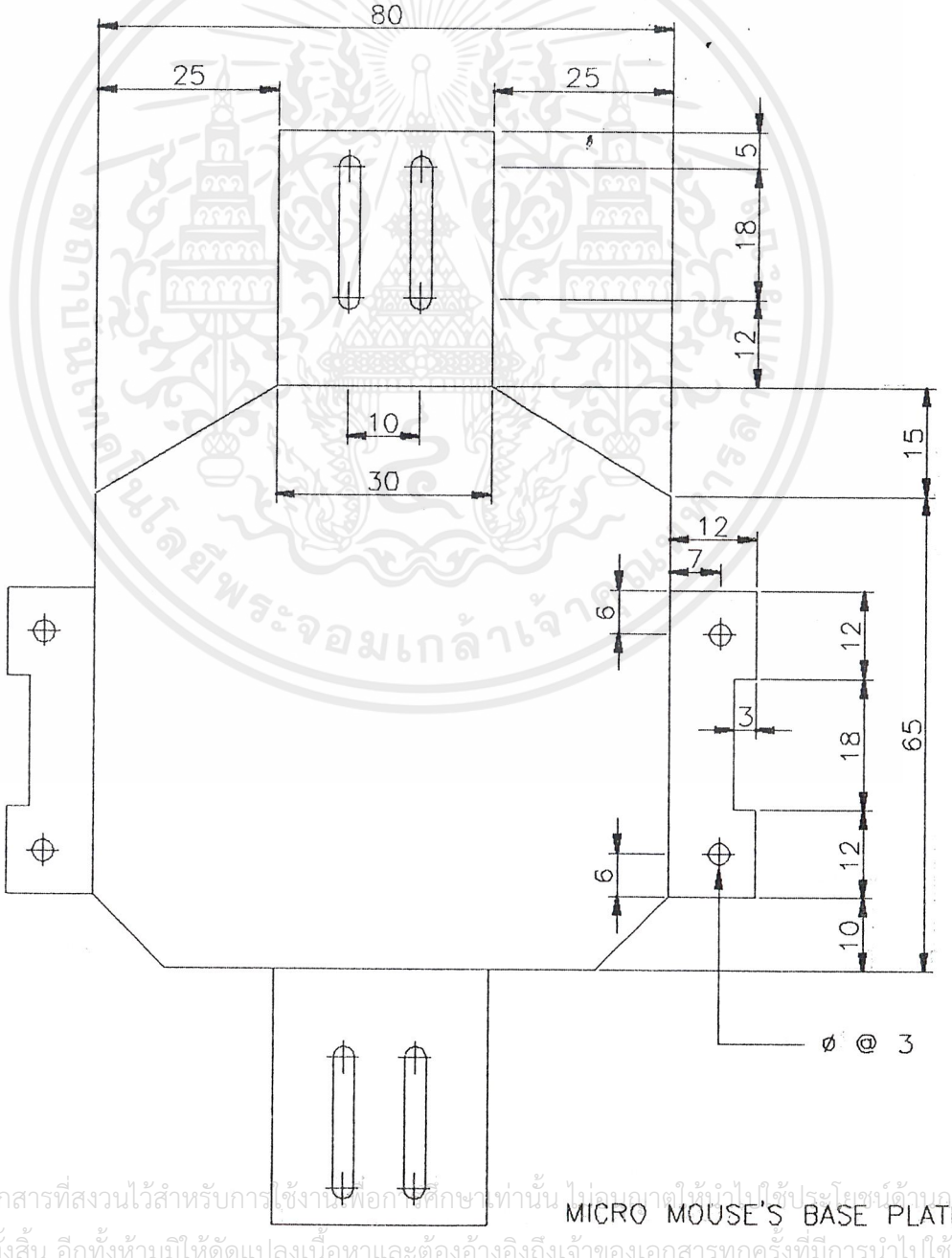



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICRO MOUSE'S ARMS

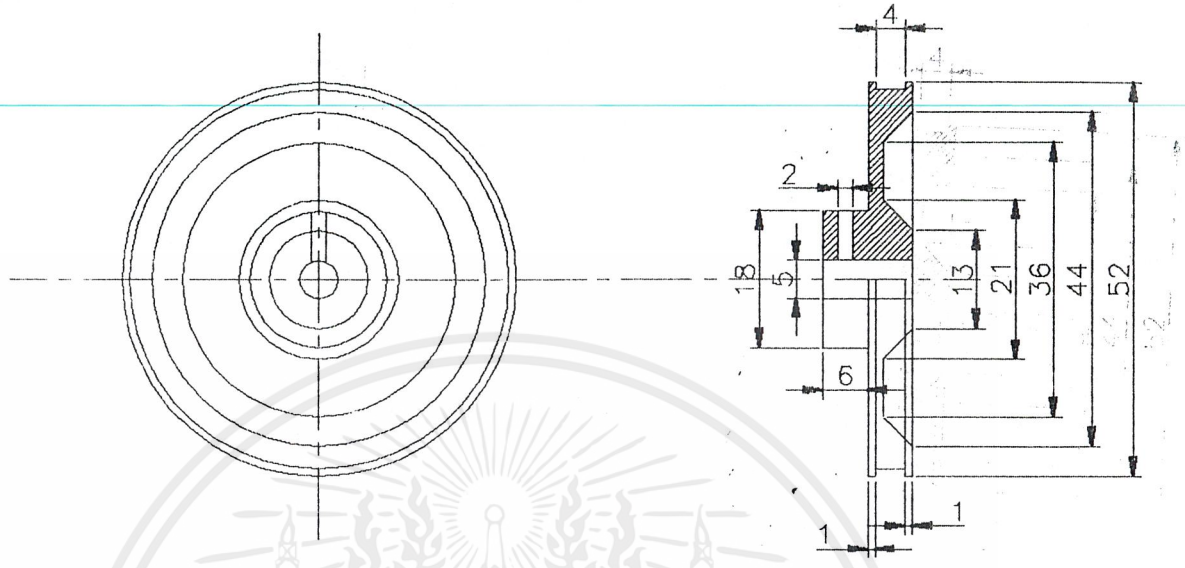



ROLLER'S SUPPORT

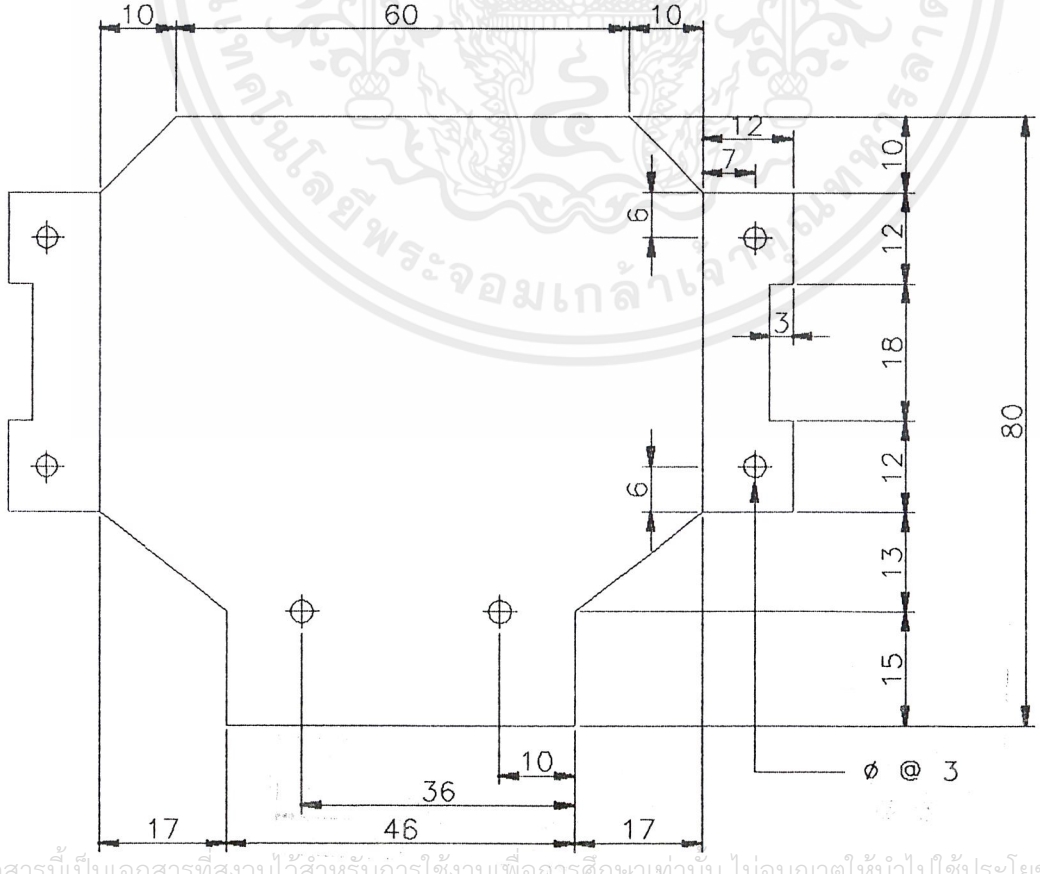



MICRO MOUSE'S BASE PLATE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




MICRO MOUSE'S WHEEL




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MICRO MOUSE'S UPPER PLATE