**ภาควิชาครุศาสตร์วิศวกรรม**

**คณะครุศาสตร์อุตสาหกรรม**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ใบรับรองปริญญานิพนธ์**

**ชื่อหัวข้อ** ชุดทดลองการควบคุมมอเตอร์ดีซีด้วยสัญญาณพัลซ์

Pulse Width Modulation Demonstrator

**ชื่อนักศึกษา**

1. นายเทพบดินทร์ บริรักษ์อราวินท์ รหัสประจำตัว 42035368
2. นายนภดล ไชยบัณฑิต รหัสประจำตัว 42035374
3. นายยุทธนากร บุญศิลป์ รหัสประจำตัว 42035377
4. นายอภิชาติ สุจริต รหัสประจำตัว 42035387

**หลักสูตร** ครุศาสตร์อุตสาหกรรมบัณฑิต **สาขาวิชา** เทคโนโลยีการวัดคุมทางอุตสาหกรรม

**อาจารย์ที่ปรึกษา** อาจารย์วรวิทย์ สมหา

**อาจารย์ที่ปรึกษาร่วม** อาจารย์สุรพงษ์ สิริพงศ์ดี

| คณะกรรมการสอบปริญญานิพนธ์ | ลายมือชื่อ |
|---|---|
| 1. อาจารย์วรวิทย์ สมหา | ............................................ |
| 2. อาจารย์สุรพงษ์ สิริพงศ์ดี | ............................................ |
| 3. อาจารย์โกศล ตราชู | ............................................ |
| 4. อาจารย์ปิยะ ศุภวราสุวัฒน์ | ............................................ |
| 5. อาจารย์อมรชัย ชัยชนะ | ............................................ |

**วัน/เดือน/ปีที่สอบ** วันเสาร์ที่ 9 ธันวาคม พ.ศ. 2543 เวลา 11.00 น.

**สถานที่สอบ** ห้อง ค.301 คณะครุศาสตร์อุตสาหกรรม สจล.

ภาควิชารับรองแล้ว

ลงนาม............................................

(ผศ.วิสุทธิ์ อธิพรธรรม)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

วันที่...10...เดือน...เมษายน...พ.ศ.............

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# ปริญญานิพนธ์

## ชุดทดลองการควบคุมดีซีมอเตอร์ด้วยสัญญาณพัลซ์
## PULSE WIDTH MODULATION DC MOTOR DRIVE DEMONSTRATOR

T040200

นายเทพบดินทร์ บริรักษ์อราวินท์
นายนภดล ไชยบัณฑิต
นายยุทธนากร บุญศิลป์
นายอภิชาติ สุจริต

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตร์อุตสาหกรรมบัณฑิต
สาขาวิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2543

ฝพ.
ท 61A
2543
เลขหมู่............
เลขทะเบียน 40200
วัน, เดือน, ปี 20 ส.ค. 2544
b. 11093043
i.
เอกสารนี้เป็นเอกสารที่จัดเก็บไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ปริญญานิพนธ์

**เรื่อง** ชุดทดลองการควบคุมดีซีมอเตอร์ด้วยสัญญาณพัลซ์

Pulse Width Modulation DC. Motor Drive Demonstrator

## วัตถุประสงค์

1. เพื่อศึกษาหลักการทำงานของ PWM
2. เพื่อนำเอาหลักการ PWM มาควบคุมมอเตอร์ไฟฟ้ากระแสตรง
3. เพื่อศึกษาถึงสถาปัตยกรรมของไมโครคอนโทรลเลอร์
4. เพื่อศึกษาถึงการประยุกต์ใช้งานของไมโครคอนโทรลเลอร์
5. เพื่อศึกษาถึงหลักการเขียนโปรแกรมแอสเซมบลี

## ประโยชน์ที่คาดว่าจะได้รับ

1. มีความเข้าใจหลักการทำงานของ PWM
2. สามารถนำเอาหลักการ PWM มาควบคุมมอเตอร์ไฟฟ้ากระแสตรงได้
3. มีความเข้าใจถึงสถาปัตยกรรมของไมโครคอนโทรลเลอร์
4. สามารถประยุกต์ใช้งานของไมโครคอนโทรลเลอร์ได้
5. สามารถเขียนโปรแกรมแอสเซมบลีได้

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **ชื่อหัวข้อ** | ชุดทดลองการควบคุมดีซีมอเตอร์ด้วยสัญญาณพัลซ์ |
| **นักศึกษา** | นายเทพบดินทร์ บริรักษ์อราวินท์ |
| | นายนภดล ไชยบัณฑิต |
| | นายยุทธนากร บุญศิลป์ |
| | นายอภิชาติ สุจริต |
| **อาจารย์ที่ปรึกษา** | อาจารย์วรวิทย์ สมหา |
| **อาจารย์ที่ปรึกษาร่วม** | อาจารย์สุรพงษ์ สิริพงศ์ดี |
| **หลักสูตร** | ครุศาสตร์อุตสาหกรรมบัณฑิต |
| **สาขาวิชา** | เทคโนโลยีการวัดคุมทางอุตสาหกรรม |
| **ปีการศึกษา** | 2543 |


**บทคัดย่อ**

ปริญญานิพนธ์ฉบับนี้ นำเสนอการออกแบบและสร้างชุดทดลอง การควบคุมดีซีมอเตอร์ด้วยสัญญาณพัลซ์ โดยใช้ตัวประมวลผลด้วยไมโครคอนโทรเลอร์ สามารถควบคุมการหมุนของมอเตอร์ได้ทั้ง 2 ทิศทาง และยังรับสัญญาณจากตัวตรวจจับตำแหน่งมาประมวลผลเป็นระบบควบคุมป้อนกลับ เพื่อให้การทำงานสมบูรณ์ยิ่งขึ้น ส่วนประกอบของชุดทดลองนี้ประกอบไปด้วยมอเตอร์เกียร์ซึ่งมีเอนโคดเดอร์เป็นส่วนประกอบอยู่ นอกจากนั้นยังมีในส่วนของอุปกรณ์ที่ใช้ในการประมวลผลในที่นี้ก็คือไมโครคอนโทรลเลอร์ ซึ่งไมโครคอนโทรลเลอร์จะรับสัญญาณจากเอนโคดเดอร์ ระบบเซนเซอร์ และ อุปกรณ์อินพุทซึ่งได้แก่แมทริกซ์สวิตซ์ จากนั้นจึงนำสัญญาณต่างๆ ที่ได้มาประมวลผลและสั่งงานอุปกรณ์เอาท์พุท ซึ่งได้แก่ มอเตอร์ จอแสดงผล ต่อไป

ชุดทดลองนี้สามารถควบคุมความเร็วและตำแหน่งของมอเตอร์ได้ดังนั้นจึงสามารถนำไปประยุกต์ใช้งานได้หลากหลายเช่น ทำเป็นระบบขับสายพานลำเลียง เป็นตัวต้นกำลังให้กับเครื่องกำเนิดไฟฟ้า ใช้กับงานควบคุมตำแหน่งของหัวเจาะ เป็นต้น


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Thesis Title** | Pulse Width Modulation DC. Motor Drive Demonstrator |
| **Students** | Mr. Thapbodin Borerakarawin |
| | Mr. Noppadon Chaibundid |
| | Mr. Yuttanakorn Boonsin |
| | Mr. Apichat Sutjarit |
| **Advisor** | Mr. Worawit Somha |
| **Co-Advisor** | Mr. Surapong Siripongdee |
| **Education Level** | Bachelor of Science in Industrial Education |
| **Program in** | Industrial Instrument Technology |
| **Acadamic Year** | 2000 |

## ABSTRACT

This thesis presents a desingning and inplementation of Pulse Width Modulation DC. Motor Drive Demonstrator. By processor with microcontroller. It can control rotate of motor two ways and it can get signal from sensor come to process result to feedback control system for perfect working. The component of Demonstrator is gear-motor with Encoder and it have the Processing device is a Microcontroller. The Microcontroller will be receive the signal from the encoder, Sensor system and input device is a matrix switch. From there the system are to take the other signal send to the microprocessor and processing the signal this here. Form there the Microcontroller sent the signal-control to output device that is a gear-motor and Display indicator.

These Demonstrators can control the speed and position of motor thus to the other work such as to drive the conveyer system. The power movement of the electric generator. and control the position of the Drills working etc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ลุล่วงไปด้วยดี เนื่องมาจากความร่วมมือของสมาชิกภายในกลุ่มทุกๆ ท่าน ขอขอบคุณคณาจารย์ภาควิชาครุศาสตร์วิศวกรรมทุกท่านที่ให้ความอนุเคราะห์เครื่องมือ และอุปกรณ์ ตลอดจนคำแนะนำ แนวความคิด ความรู้ต่างๆ แนวทางการแก้ไขปัญหา คำสั่งสอนว่ากล่าวตักเตือน ในการทำโครงงานเสมอมา ขอขอบคุณห้องสมุดคณะครุศาสตร์อุตสาหกรรม ห้องสมุดคณะวิศวกรรมศาสตร์ และหอสมุดกลาง ที่ช่วยอำนวยความสะดวกและเอื้อเฟื้อสถานที่ในการค้นคว้าหาข้อมูล ขอขอบคุณเจ้าหน้าที่ห้องโสตและผู้สนับสนุนท่านอื่นๆ ที่ให้ความอนุเคราะห์ในเรื่องอุปกรณ์และเครื่องมือ และที่สำคัญที่สุด คือ บิดา มารดาที่ได้คอยช่วยเหลือในด้านงบประมาณการศึกษา และเป็นผู้ให้กำลังใจด้วยดีเสมอมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| เรื่อง | หน้า |
|---|---|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ | II |
| กิตติกรรมประกาศ | III |
| สารบัญ | IV |
| สารบัญตาราง | VII |
| สารบัญรูป | VIII |
| บทที่ 1 บทนำ | 1 |
| 1.1 ความเป็นมาและความคัญของปริญญานิพนธ์ | 1 |
| 1.2 ขีดความสามารถของโครงงาน | 1 |
| 1.3 เนื้อหาโดยสังเขป | 2 |
| บทที่ 2 ทฤษฎีและหลักการ | 3 |
| 2.1 แนะนำเซอร์โวแอมปลิไฟเออร์ | 3 |
| 2.1.1 การขยายแบบเชิงเส้น | 3 |
| 2.1.2 SCR แอมปลิไฟเออร์ | 5 |
| 2.1.3 สวิทชิ่งแอมปลิไฟเออร์ | 7 |
| 2.1.4 ลิเนียร์เซอร์โวแอมปลิไฟเออร์ | 8 |
| 2.1.5 พัลซ์วิดท์โมดูเลชั่นแอมปลิไฟเออร์ | 10 |
| 2.1.6 การทำงานของพัลซ์วิดท์โมดูเลชั่นแอมปลิไฟเออร์ | 11 |
| 2.1.7 ระบบควบคุมความเร็วที่ใช้พัลซ์วิดท์โมดูเลชั่นแอมปลิไฟเออร์ | 13 |
| 2.2 องค์ประกอบของระบบเซอร์โว | 14 |
| 2.2.1 หลักการทำงานของดีซีมอเตอร์ | 14 |
| 2.2.2 โมเดลคณิตศาสตร์ของมอเตอร์ไฟฟ้ากระแสตรง | 15 |
| 2.2.3 การหลีกเลี่ยงข้อผิดพลาดในการเลือกขนาดมอเตอร์ | 21 |
| 2.2.4 ความสัมพันธ์ระหว่างการเคลื่อนที่แบบเป็นเชิงเส้นและแบบเป็นเชิงมุม | 23 |
| 2.2.5 ระบบเกียร์ | 23 |
| 2.2.6 ระบบสายพานและคานงัด | 25 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)

| เรื่อง | หน้า |
|---|---|
| 2.3 เซ็นเซอร์และเอนโคดเดอร์ของเพลามอเตอร์ | 25 |
| 2.3.1 โพเทนทิโอมิเตอร์ | 25 |
| 2.3.2 ทาโคมิเตอร์ | 27 |
| 2.3.3 อินครีเมนท์เอนโคดเดอร์ | 28 |
| 2.4 ระบบควบคุมความเร็ว | 31 |
| 2.4.1 ระบบการควบคุมความเร็วของมอเตอร์แบบวงรอบเปิด | 31 |
| 2.4.2 ระบบการควบคุมความเร็วของมอเตอร์แบบวงรอบปิด | 33 |
| 2.4.3 ระบบการควบคุมความเร็วได้เพียงทิศทางเดียวหรือควอดแดรนท์เดียว | 34 |
| 2.4.4 การวิเคราะห์และออกแบบระบบควบคุมความเร็ว | 35 |
| 2.4.5 ระบบควบคุมแรงบิด | 36 |
| 2.4.6 แอมปลิไฟเออร์ที่ประกอบด้วยแหล่งกำเนิดแรงดันและกระแส | 36 |
| 2.4.7 การเปลี่ยนแปลงความเร็วของมอเตอร์ | 37 |
| 2.5 ระบบการควบคุมตำแหน่งการเคลื่อนที่ | 39 |
| 2.5.1 การวิเคราะห์ระบบเซอร์โวที่ใช้งานในการควบคุมตำแหน่งหมุนของเพลามอเตอร์ไฟฟ้ากระแสตรง | 40 |
| 2.5.2 คุณสมบัติไดนามิคของระบบเซอร์โว | 40 |
| 2.5.3 คุณสมบัติสะแตติคของระบบเซอร์โว | 41 |
| 2.6 ไมโครคอนโทรลเลอร์ | 41 |
| 2.6.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51 | 42 |
| 2.6.2 โครงสร้างของไมโครคอนโทรลเลอร์ตระกูล MCS-51 | 43 |
| 2.6.3 โครงสร้างของ 8051 | 44 |
| 2.6.4 สถาปัตยกรรมภายในของ 8051 | 46 |
| 2.6.5 การจัดหน่วยความจำของ 8051 | 48 |
| 2.6.6 ฐานเวลาในการทำงานของไมโครคอนโทรลเลอร์ | 50 |
| 2.6.7 การทำงานของ 8051 | 50 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)


| เรื่อง | หน้า |
|---|---|
| บทที่ 3 การออกแบบ การสร้าง และการทำงาน | 51 |
| 3.1 การออกแบบอุปกรณ์ทางอิเล็กทรอนิกส์ | 51 |
| 3.1.1 ภาคจ่ายไฟ | 51 |
| 3.1.2 ภาคระบบควบคุม | 53 |
| 3.1.3 ภาคขับมอเตอร์ | 53 |
| 3.1.4 ภาคระบบป้อนกลับ | 54 |
| 3.1.5 ภาคอินพุท | 55 |
| 3.1.6 ภาคเอาท์พุท | 56 |
| 3.1.7 ตัวตรวจจับตำแหน่ง | 56 |
| 3.2 การออกแบบอุปกรณ์ทางกล | 57 |
| บทที่ 4 การทดลองและผลการทดลอง | 59 |
| 4.1 การทดลองในส่วนของอิเล็กทรอนิกส์ | 59 |
| 4.2 การทดลองในส่วนของโปรแกรม | 59 |
| บทที่ 5 บทสรุป ปัญหา แนวทางการแก้ไขและพัฒนา | 65 |
| 5.1 บทสรุป | 65 |
| 5.2 ปัญหาและแนวทางการแก้ไข | 65 |
| 5.2.1 ส่วนของอุปกรณ์อิเล็กทรอนิกส์ | 65 |
| 5.2.2 การประกอบวงจรและทดสอบการทำงานจริง | 66 |
| 5.2.3 การออกแบบทางกล | 67 |
| 5.2.4 แรงบิดของมอเตอร์ | 67 |
| 5.3 แนวทางในการพัฒนา | 67 |
| ภาคผนวก ก โปรแกรม Pulse Width Modulation | 68 |
| ภาคผนวก ข วิธีการใช้งาน | 113 |
| ภาคผนวก ค รูปวงจร และภาพโครงงาน | 118 |
| ภาคผนวก ง รายละเอียดคุณสมบัติเฉพาะของอุปกรณ์ | 121 |
| บรรณานุกรม | 160 |
| ประวัติผู้แต่ง | 161 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

| เรื่อง | หน้า |
| --- | --- |
| ตารางที่ 2.1 ไมโครคอนโทรลเลอร์ตระกูล 8051 | 42 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป


| เรื่อง | หน้า |
|---|---|
| รูปที่ 2.1 ภาคเอาท์พุทแบบ "H" (ใช้แหล่งจ่ายไฟตัวเดียว) | 3 |
| รูปที่ 2.2 ภาคเอาท์พุทแบบ "T" (ใช้แหล่งจ่ายไฟสองตัว) | 4 |
| รูปที่ 2.3 ดีซีมอเตอร์และทาโคมิเตอร์สำหรับใช้ในระบบบังคับแบบมีการป้อนกลับ | 19 |
| รูปที่ 2.4 มอเตอร์ไฟฟ้ากระแสตรงที่มีเกียร์บล็อก | 19 |
| รูปที่ 2.5 ระบบมอเตอร์และลีดสกรู | 23 |
| รูปที่ 2.6 ระบบซี่เฟืองตรงและซี่เฟืองจักร | 23 |
| รูปที่ 2.7 ระบบการคัปปลิ้งของเกียร์ | 23 |
| รูปที่ 2.8 ระบบเกียร์ที่สามารถเพิ่มแรงบิดได้หลายเท่า | 24 |
| รูปที่ 2.9 ระบบกว้านและสายพาน | 25 |
| รูปที่ 2.10 โรตารีโพเทนทิโอมิเตอร์ชนิดหมุนได้ 10 รอบ | 26 |
| รูปที่ 2.11 โครงสร้างภายในของโพเทนทิโอมิเตอร์แบบเพลาหมุนได้รอบเดียว | 26 |
| รูปที่ 2.12 บล็อกไดอะแกรมของระบบการบังคับความเร็ว | 27 |
| รูปที่ 2.13 ตัวอย่างกลไกของออปโตอินครีเมนท์ | 28 |
| รูปที่ 2.14 ตัวเซ็นเซอร์แบบมีช่องปิดเปิดให้แสงผ่านได้ช่องเดียวและแบบมีหลายช่อง | 29 |
| รูปที่ 2.15 โรตารีเอนโคดเดอร์ที่มีเอาท์พุทเป็นไบนารี | 30 |
| รูปที่ 2.16 ตัวอย่างส่วนประกอบของเอนโคดเดอร์ | 30 |
| รูปที่ 2.17 โครงสร้างของ 8051 | 43 |
| รูปที่ 2.18 สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์เบอร์ 8051 | 45 |
| รูปที่ 2.19 ขาต่างๆ ของ 8051 | 47 |
| รูปที่ 2.20 การจัดพื้นที่หน่วยความจำสำหรับโปรแกรมของ 8051 | 48 |
| รูปที่ 2.21 การจัดพื้นที่หน่วยความจำข้อมูล | 48 |
| รูปที่ 2.22 ผังเวลาของซีพียู | 49 |
| รูปที่ 3.1 วงจรแสดงภาคจ่ายไฟ (แบบแหล่งจ่ายไฟคู่และแบบแหล่งจ่ายไฟเดี่ยว) | 52 |
| รูปที่ 3.2 วงจรแสดงภาคระบบควบคุม | 53 |
| รูปที่ 3.3 วงจรแสดงภาคขับมอเตอร์ | 54 |
| รูปที่ 3.4 วงจรแสดงภาคระบบป้อนกลับ | 55 |
| รูปที่ 3.5 วงจรแสดงภาคอินพุท | 55 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป(ต่อ)

| เรื่อง | หน้า |
|---|---|
| รูปที่ 3.6 วงจรแสดงภาคเอาท์พุท | 56 |
| รูปที่ 3.7 วงจรแสดงตัวตรวจจับตำแหน่ง | 57 |
| รูปที่ 3.8 แสดงการออกแบบลูกพูลเลย์ | 58 |
| รูปที่ 3.9 ภาพตัดขวางของคอนเวนเยอร์ | 58 |
| รูปที่ 4.1 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 10 เปอร์เซนต์ | 60 |
| รูปที่ 4.2 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 20 เปอร์เซนต์ | 60 |
| รูปที่ 4.3 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 30 เปอร์เซนต์ | 61 |
| รูปที่ 4.4 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 40 เปอร์เซนต์ | 61 |
| รูปที่ 4.5 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 50 เปอร์เซนต์ | 62 |
| รูปที่ 4.6 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 60 เปอร์เซนต์ | 62 |
| รูปที่ 4.7 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 70 เปอร์เซนต์ | 63 |
| รูปที่ 4.8 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 80 เปอร์เซนต์ | 63 |
| รูปที่ 4.9 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 90 เปอร์เซนต์ | 64 |
| รูปที่ 4.10 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 100 เปอร์เซนต์ | 64 |
| รูปที่ ข.1 การแสดงผลของจอ LCD 1 | 114 |
| รูปที่ ข.2 การแสดงผลของจอ LCD 2 | 114 |
| รูปที่ ข.3 การแสดงผลของจอ LCD 3 | 114 |
| รูปที่ ข.4 การแสดงผลของจอ LCD 4 | 115 |
| รูปที่ ข.5 การแสดงผลของจอ LCD 5 | 115 |
| รูปที่ ข.6 การแสดงผลของจอ LCD 6 | 115 |
| รูปที่ ข.7 การแสดงผลของจอ LCD 7 | 116 |
| รูปที่ ข.8 การแสดงผลของจอ LCD 8 | 116 |
| รูปที่ ข.9 การแสดงผลของจอ LCD 9 | 116 |
| รูปที่ ข.10 การแสดงผลของจอ LCD 10 | 116 |
| รูปที่ ข.11 การแสดงผลของจอ LCD 11 | 117 |
| รูปที่ ค.1 วิธีการตรวจจับตำแหน่ง | 119 |
| รูปที่ ค.2 เครื่องต้นแบบ | 119 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญรูป(ต่อ)

| เรื่อง | หน้า |
|---|---|
| รูปที่ ค.3 การเชื่อมต่อไมโครคอนโทรลเลอร์เข้ากับอุปกรณ์ | 120 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปริญญานิพนธ์

เนื่องจากว่าทางกลุ่มได้เล็งเห็นปัญหาเกี่ยวกับการควบคุมตำแหน่งและการควบคุมความเร็วของมอเตอร์นั้น สามารถทำได้ยากและไม่ละเอียดเพียงพอ เนื่องมาจากสัญญาณป้อนกลับส่วนใหญ่ที่ใช้ในปัจจุบัน เป็นสัญญาณแบบเปิด-ปิด จึงไม่สามารถตรวจสอบการทำงานของระบบอย่างต่อเนื่องได้ ในการควบคุมตำแหน่งที่ละเอียดมากนั้น ต้องใช้ตัวตรวจจับแบบเปิด-ปิด หลายจุดเพื่อให้ได้ค่าตำแหน่งที่ละเอียดตามความต้องการของผู้ใช้ จึงเป็นการสิ้นเปลืองตัวตรวจจับแบบเปิด-ปิดเป็นอย่างมาก และในส่วนของการควบคุมความเร็วของมอเตอร์ในงานอุตสาหกรรม จะสามารถเลือกความเร็วรอบในการหมุนได้ไม่มากนัก อีกทั้งอุปกรณ์ที่ใช้สำหรับวัดความเร็วรอบ ก็ยังไม่สามารถนำมาใช้เป็นสัญญาณป้อนกลับ เพื่อมาประมวลผลในการควบคุมความเร็วของมอเตอร์ได้โดยตรง

จากปัญหาดังกล่าว ทางกลุ่มจึงได้จึงได้พยายามศึกษาและค้นคว้าวิธีการควบคุมมอเตอร์ไฟฟ้ากระแสตรง โดยใช้หลักการ PWM (Pulse Width Modulation) เพื่อที่จะสามารถควบคุมความเร็วร่วมกับการควบคุมตำแหน่งได้ โดยการนำเอาไมโครคอนโทรลเลอร์ มาใช้ในการประมวลผลข้อมูล ซึ่งจะทำให้การควบคุมตำแหน่งและความเร็วในการใช้งานมีประสิทธิภาพมากยิ่งขึ้น

ดังนั้นทางกลุ่มของข้าพเจ้า จึงมีความต้องการที่จะนำเอาหลักการควบคุมดังกล่าวมาจัดทำชุดสาธิตการสอน เพื่อให้นักศึกษาได้มีความรู้ ความเข้าใจในหลักการ PWM (Pulse Width Modulation) มากยิ่งขึ้น และเพื่อให้นักศึกษาได้มีการฝึกหัดและทำการทดลองใช้งานจริงจากชุดทดลองนี้ เนื่องจากชุดทดลองนี้ได้นำเอาไมโครคอนโทรลเลอร์มาใช้ในการควบคุมมอเตอร์ไฟฟ้ากระแสตรง ดังนั้นนักศึกษาที่ได้ทำการทดลองจึงมีประสบการณ์จริงในการใช้ซอฟท์แวร์อีกด้วย

## 1.2 ขีดความสามารถของโครงงาน

1. สามารถควบคุมความเร็วของมอเตอร์ไฟฟ้ากระแสตรงด้วยไมโครคอนโทรลเลอร์ได้
2. สามารถควบคุมตำแหน่งของมอเตอร์ไฟฟ้ากระแสตรงด้วยไมโครคอนโทรลเลอร์ได้
3. สามารถเลือกฟังก์ชั่นการทำงานแบบวงรอบปิดหรือวงรอบเปิดจากคีย์บอร์ด
4. สามารถใช้เอ็นโค๊ดเดอร์เป็นตัวตรวจจับการทำงานของมอเตอร์ไฟฟ้ากระแสตรงได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. สามารถตรวจจับความเร็วรอบและแสดงผลไปยัง LCD ได้
6. สามารถใช้เป็นชุดทดลองเพื่อแสดงถึงหลักการทำงานของ PWM ให้กับนักศึกษาได้

## 1.3 เนื้อหาโดยสังเขป

เนื้อหาภายในปริญญานิพนธ์ฉบับนี้ ประกอบไปด้วย

บทที่ 2 กล่าวถึง ทฤษฎีและหลักการของโครงงาน ซึ่งจะประกอบไปด้วยเนื้อหาที่เกี่ยวข้องกับโครงงาน ได้แก่ PWM, เซนเซอร์และเอนโคดเดอร์, ระบบควบคุมความเร็ว, ระบบควบคุมตำแหน่ง, ไมโครคอนโทรลเลอร์

บทที่ 3 กล่าวถึง การออกแบบ การสร้าง และการทำงาน ซึ่งประกอบไปด้วย แนวคิดวิธีการออกแบบวงจร ในภาคต่างๆ

บทที่ 4 กล่าวถึง การทดลองและผลการทดลอง ที่ได้มาจากการออกแบบ การสร้าง และการทำงานจากบทที่ 3 ซึ่งจะบอกผลการทดลองในส่วนต่างๆ

บทที่ 5 กล่าวถึง บทสรุป ปัญหา แนวทางแก้ไขปัญหาที่สำคัญในการทำโครงงาน และแนวทางในการพัฒนาโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# ทฤษฎีและหลักการ

## 2.1 แนะนำเซอร์โวแอมปลิไฟเออร์

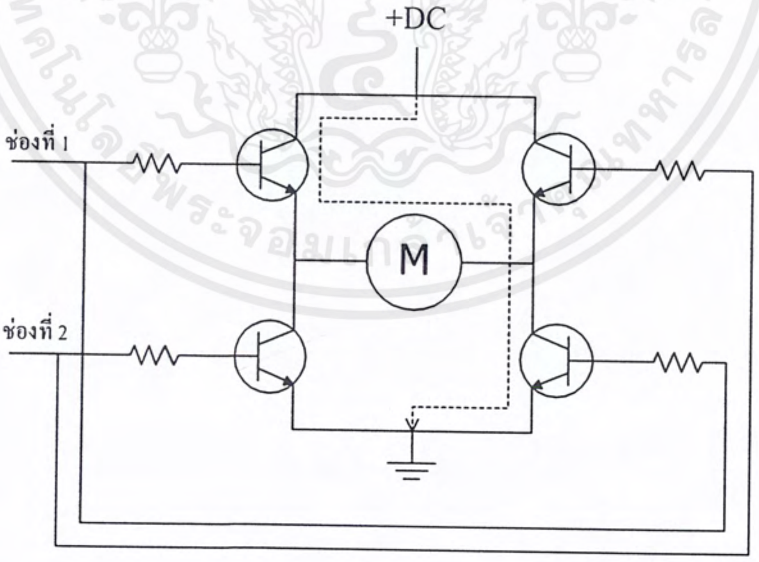
เซอร์โวแอมปลิไฟเออร์ที่แท้จริงสามารถให้เอาท์พุทแรงดันหรือกระแสที่มีค่าเป็นบวกและลบได้ดังนั้นมันจะทำงานได้ใน 4 ควอดแดร้นท์

เราสามารถจัดประเภทของเซอร์โวแอมปลิไฟเออร์ออกได้เป็น 3 ประเภท คือ

1) การขยายแบบเชิงเส้น
2) SCR (Silicon Control Rectifier) แอมปลิไฟเออร์
3) สวิทชิ่งแอมปลิไฟเออร์

### 2.1.1 การขยายแบบเชิงเส้น

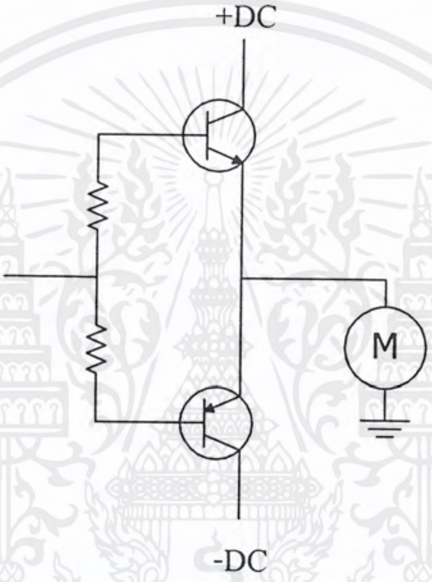
เซอร์โวแอมปลิไฟเออร์แบบทรานซิสเตอร์ จัดออกได้ตามหลักการออกแบบภาคเอาท์พุทออกได้เป็น 2 แบบ คือ แบบ "H" และ "T" มีลักษณะวงจรดังรูปที่ 2.1 และ 2.2




**รูปที่ 2.1** ภาคเอาท์พุทแบบ "H" (ใช้แหล่งจ่ายไฟตัวเดียว)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรภาคเอาท์พุทแบบ "H" หรือแบบบริดจ์ประกอบด้วยทรานซิสเตอร์ 4 ตัว ใช้แหล่งจ่ายไฟฟ้ากระแสตรงเพียงตัวเดียว ภาคเอาท์พุทแบบนี้มีข้อดี คือ ง่าย แหล่งจ่ายไฟใช้เพียงขั้วเดียว (+DC) และการป้องกันแรงดันป้อนกลับจะมีการแบ่งส่วนกันระหว่างทรานซิสเตอร์ อย่างไรก็ตาม เป็นการยากที่จะให้วงจรภาคเอาท์พุทแบบนี้ทำงานเป็นเชิงเส้น และกระทำการป้อนกลับของกระแสและแรงดันจากมอเตอร์กระแสทำได้ยาก เนื่องจากมอเตอร์อยู่ในสภาวะลอยตัว (หรือไม่ได้ต่ออยู่กับกราวนด์)




**รูปที่ 2.2** ภาคเอาท์พุทแบบ "T" (ใช้แหล่งจ่ายไฟสองตัว)

วงจรภาคเอาท์พุทแบบ T จะต้องใช้แหล่งจ่ายสองชุด (+DC และ –DC) และทรานซิสเตอร์สองตัวที่เป็นส่วนกลับกัน แต่สามารถขับให้เป็นเชิงเส้นได้ง่ายและสามารถกระทำป้อนกลับของแรงดันและกระแสได้ง่าย ด้วยเหตุผลดังกล่าวนี้วงจรแบบ T มักได้รับการนิยมใช้เป็นการขยายแบบเชิงเส้น การไบอัสเอาท์พุททรานซิสเตอร์ ต้องให้ความระมัดระวังเป็นพิเศษ เพราะถ้าหากทรานซิสเตอร์ทั้งสองตัวนำกระแสในเวลาเดียวกันจะทำให้เกิดการลัดวงจรระหว่างแหล่งจ่าย 2 ชุด

คุณสมบัติที่แสดงถึงความสัมพันธ์ระหว่างอินพุทกับเอาท์พุท ของภาคเอาท์พุทตามปกติมักจะมีลักษณะของ Dead Zone ซึ่งเป็นลักษณะการขยายแบบไม่เป็นเชิงเส้น ถ้าเราให้การขยายนั้นมีการป้อนกลับแบบลบ ก็สามารถลดความไม่เป็นเชิงเส้นดังกล่าวลงได้ ในกรณีเมื่อการขยายแบบเชิงเส้นใช้งานเป็นแบบ “Bang Bang Control” (ON – OFF Control) ค่าของ Dead Zone ไม่เป็นสิ่งสำคัญ เนื่องจากการขยายจะทำงานเฉพาะในส่วนที่นำกระแสหรือไม่นำกระแสเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุทกำลังของการขยายแบบเชิงเส้นจะถูกกำจัดด้วยปริมาณความร้อนที่เกิดขึ้นที่ภาคเอาท์พุทเอง เนื่องจากกำลังที่สูญเสียในภาคเอาท์พุทจะเท่ากับผลคูณของกระแสและแรงดันตกคร่อมทรานซิสเตอร์ทรานซิสเตอร์และตัวระบายความร้อนที่ต่อร่วมอยู่จะต้องสามารถทนต่อความร้อนที่เกิดจากกำลังที่สูญเสียในตัวทรานซิสเตอร์เอง

ตัวแปรอีกอันหนึ่งที่ต้องพิจารณาในการออกแบบตัวขยาย คือ เซคอนดารี่เบรคดาวน์คาแร็คเตอร์ริสติค (Secondary Brake Down Characteristic) ของทรานซิสเตอร์ซึ่งจะเป็นตัวจำกัดระยะเวลาการไหลของกระแสสูงสุดที่ระดับแรงดันที่กำหนดเพื่อให้แน่ใจว่าไม่เกินจำกัด ดังนั้นลิเนียร์หรือเซอร์โวแอมปลิไฟเออร์ควรจะต้องมีวงจรจำกัดกระแสเพื่อป้องกันไม่ให้แอมปลิไฟเออร์พังหรือเสียหายได้ภายใต้เงื่อนไขที่มอเตอร์ถูกจับให้หยุดหรือมีภาระแรงบิดเกินไป

ในบางกรณีเซอร์แอมปลิไฟเออร์จะมีวงจรจำกัดกระแสถึง 2 ภาคเพื่อใช้จำกัดกระแสสูงสุดสูงๆตอนมอเตอร์เพิ่มอัตราเร่ง ( Acceteration ) โดยแบ่งออกเป็น

1. วงจรจำกัดกระแสอันแรกจะตั้งไว้ที่ระดับกระแสสูงๆ ด้วยการกำจัดเวลาเป็นเศษส่วนของวินาที

2. ส่วนวงจรจำกัดกระแสอันที่สองจะถูกตั้งไว้ที่ระดับกระแสต่ำกว่า โดยไม่คำนึงถึงระยะเวลา นอกจากนั้นวงจรนี้ไม่เพียงใช้ป้องกันภาคเอาท์พุทของการขยายเท่านั้น แต่ยังป้องกันมอเตอร์ให้ปลอดภัยจากอุบัติเหตุที่จะเกิดภาระเกินได้ง่าย

### 2.1.2 SCR แอมปลิไฟเออร์

ระบบควบคุมความเร็วด้วย SCR มักจะใช้สายไฟฟ้ากระแสสลับที่ความถี่ (50-60 เฮิรตซ์) สำหรับระบบเฟสเดียวแบบครึ่งคลื่น ย่านของระบบจะถูกจำกัดอยู่ที่ 2-3 เฮิรตซ์ (หรือมีช่วงพิสัยการควบคุมความเร็วเท่ากับ 5:1) ส่วนระบบเต็มคลื่นแบบสามเฟส จะมีย่านอยู่ที่ 25-30 เฮิรตซ์ (หรือมีช่วงพิสัยการควบคุมความเร็วเท่ากับ 20:1)

ระบบควบคุมความเร็วที่ใช้ SCR จะไม่สามารถควบคุมความเร็วได้ช่วงพิสัยกว้างเหมือนกับการควบคุมด้วยทรานซิสเตอร์ที่มีการขยายแบบเชิงเส้น ปัญหาหนักอีกอันหนึ่งของการควบคุมมอเตอร์ไฟฟ้ากระแสตรงด้วย SCR คือ มันจะให้รูปแบบของตัวประกอบที่มีค่าสูง ซึ่งอาจจะทำให้การทำงานของมอเตอร์เลวลงและมีผลกระทบให้เกิดความร้อนในตัวมอเตอร์

รูปแบบของตัวประกอบของการควบคุมด้วย SCR จะขึ้นอยู่กับชนิดของการควบคุม (ครึ่งคลื่น, เต็มคลื่น, สายกำลังไฟฟ้าสามเฟส) ขนาดของมุมที่นำกระแส และขึ้นอยู่กับความเหนี่ยวนำของมอเตอร์ ดังนั้นจึงเป็นการยากที่จะหาค่ารูปแบบของตัวประกอบโดยที่ไม่รู้รายละเอียดการนำไปประยุกต์ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีการควบคุมเซอร์โวมอเตอร์ที่มีคุณภาพสูงและมีความเหนี่ยวนำต่ำ เราสามารถกำหนดค่ารูปแบบของตัวประกอบ ได้เมื่อภาระเป็นความต้านทาน เราสามารถจะทำให้รูปแบบของตัวประกอบมีค่าน้อยลงได้โดยการเพิ่มตัวเหนี่ยวนำอนุกรมกับวงจรของมอเตอร์ซึ่งจะทำให้กระแส RMS มีค่าต่ำสุด อย่างไรก็ตาม การเพิ่มค่าของความเหนี่ยวนำอาจมีผลทำให้ย่านของระบบลดลง ซึ่งจะมีผลกระทบถึงช่วงพิสัยของความเร็วหรือเสถียรภาพของระบบ

ปัญหาที่สำคัญอีกอันหนึ่งในระบบควบคุมความเร็วด้วย SCR คือ Transportation Lag ในระบบที่เกิดจากการกำหนดมุมให้ SCR นำกระแสแต่ละครึ่งรอบ มีระยะห่างกันมากเกินไป หมายถึงในกรณีการควบคุมที่ความเร็วต่ำ ระบบที่อยู่ในสถานะภาพดังกล่าว จะทำงานในลักษณะที่มีขนาดของมุม การนำกระแสน้อยมาก และถ้าระบบได้รับการเปลี่ยนแปลงจากภาระทันทีทันใด ซึ่งจะยังผลให้ความเร็วเปลี่ยน และเป็นสาเหตุให้สัญญาณผิดพลาดไป ทำให้ภาคเอาท์พุทมีขนาดของมุมการนำกระแสเพิ่มขึ้นอย่างมาก จากนั้นจะเกิดผลการเปลี่ยนแปลงของกำลังอย่างหนัก คือ ส่วนของการนำกระแสในครึ่งรอบเพิ่มขึ้น แม้ว่าในระหว่างนั้นสัญญาณผิดพลาดจะกลับเข้าสู่ค่าปกติแล้วก็ตาม ในตอนนี้มอเตอร์จะมีความเร็วสูงที่ผิดไปเป็นจำนวนมาก ซึ่งมันจะต้องลดความเร็วลงสู่ความเร็วที่กำหนดไว้อีกครั้ง คือ มีรูปแบบของมุมการนำกระแสคล้ายเดิม คุณลักษณะนี้เป็นการเกิดความเร็วติดขัดหรือกระตุก (Cogging Action)

การแก้ไขปัญหาของระบบ กระทำได้ไม่ยาก คือ ทำให้ผลตอบสนองของระบบช้าลงโดยใช้ Lag Network ก็จะทำให้ผลของความไม่เสถียรภาพหายไป แต่ก็จะทำให้ระบบมีผลตอบสนองต่อการเปลี่ยนแปลงภาระ และการเปลี่ยนแปลงคำสั่งของความเร็วเป็นไปอย่างเชื่องช้า การแก้ปัญหาด้วย Lag Network เพื่อให้ระบบมีผลตอบสนองช้าลงนี้ไม่มีความจำเป็นในระบบ การใช้งานที่มีภาระคงที่หรือภาระที่มีค่าการเคลื่อนที่ของความเฉื่อยสูง ระบบการควบคุมความเร็วที่มีช่วงพิสัยการปรับความเร็วกว้าง มีคุณภาพการควบคุมสูง จะไม่ใช้การแก้ปัญหาด้วย Lag Network

การชดเชยระบบควบคุมด้วย SCR ได้ถูกต้องเหมาะสมกับการประยุกต์ไปใช้งาน การควบคุมด้วย SCR ก็จะเป็นระบบควบคุมความเร็วที่ดี เรามีตัวประกอบหลักที่ต้องพิจารณาดังนี้

- คุณภาพการทำงานระบบ
- ราคา
- ความคล่องตัวในการใช้งาน
- อัตถประโยชน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.3 สวิทชิ่งแอมปลิไฟเออร์

การขยายแบบเชิงเส้นที่ใช้ในระบบควบคุมความเร็วจะมีปัญหาเกี่ยวกับความร้อนที่เกิดขึ้นในภาคเอาท์พุทซึ่งจะเป็นต้องใช้พัดลมระบายความร้อนในการขยายแบบเชิงเส้นที่มีขนาดมากกว่า 100 ถึง 200 วัตต์ (ขึ้นอยู่กับอุณหภูมิของอากาศและการออกแบบตัวระบายความร้อน)

สวิทช์ชิ่งแอมปลิไฟเออร์สามารถแก้ไขปัญหาความร้อนที่เกิดในภาคเอาท์พุทได้โดยภาคเอาท์พุทของสวิทช์ชิ่งแอมปลิไฟเออร์จะทำงานเป็นสวิทซ์อย่างรวดเร็วจากสถานะที่ไม่นำกระแสไปสู่สถานะการนำกระแสเต็มที่ ดังนั้นจะทำให้ภาคเอาท์พุททำงานในย่านที่มีการสูญเสียน้อยที่สุด

ปัจจุบันเราสามารถออกแบบกำลังสวิทซ์ซึ่งแอมปลิไฟเออร์ที่สามารถให้เอาท์พุทเป็นรูปคลื่นสี่เหลี่ยมที่มีสวิทช์ชิ่งเรทถึง 50 กิโลเฮิรตซ์ การที่จะได้สวิทช์ชิ่งเรทสูงๆ นั้น เราจะต้องใช้เทคนิคการป้อนกลับด้วยกระแสบวก ( Positive Current Feedback ) ด้วยเทคนิคนี้จะทำให้ระบบเซอร์โวมีย่าน ได้หลาย กิโลเฮิรตซ์

กำลังที่สูญเสียในสวิทช์ชิ่งแอมป์ มีสาเหตุมาจากสองประการ คือ

1) แรงดันตรงตกคร่อมเอาท์พุททรานซิสเตอร์มีค่าประมาณ 1 หรือ 2 โวลท์

2) ทรานซิชั่นไทม์ (transition time) คือ เวลาที่ใช้ในการสวิทช์จากสถานะ ON ไปสู่สถานะ OFF แอมปลิไฟเออร์ที่สวิทช์ชิ่งเรทสูงทรานซิชั่นไทม์จะเป็นเวลาส่วนสำคัญของสวิทช์ชิ่งไทม์ทั้งหมด ดังนั้นการกำหนดสวิทช์ชิ่งเรทให้มีค่าสูงเพื่อหวังให้ได้ย่านกว้างที่สุดนั้นสวิทช์ชิ่งเรทอาจมีผลกระทบถึงการสูญเสียกำลังในภาคเอาท์พุทด้วย

การควบคุมกำลังด้วยสวิทช์ชิ่งแอมปลิไฟเออร์กระทำได้ 3 วิธีที่เป็นพื้นฐานดังนี้

1) พัลซ์ – วิดท์โมดูเลชั่น (Pulse Width Modulation หรือ PWM)
2) พัลซ์ – ฟรีเควนซีโมดูเลชั่น (Pulse Frequency Modulation หรือ PFM)
3) SCR (Silicon Control Rectifier)

**พัลซ์-วิดท์โมดูเลชั่น (Pulse Width Modulation หรือ PWM)**

ระบบ PWM ปกติจะใช้ แหล่งจ่ายไฟฟ้ากระแสตรงและแอมปลิไฟเออร์เป็นตัวสวิทช์ชิ่งแหล่งจ่ายแรงดันให้ ON และ OFF ที่ความถี่คงที่และมีส่วนของมุม ON ที่ปรับค่าได้ (α) ดังนั้นเราสามารถปรับค่าแรงดันเฉลี่ยที่จ่ายให้ภาระจะขึ้นอยู่กับสวิทช์ชิ่งเรทและอินดัคแต้นซ์ของภาระ

เมื่อภาระเพิ่มขึ้นความถี่ของ PWM แอมปลิไฟเออร์จะคงที่ แต่จะเปลี่ยนค่าคงที่ดิวตี้ไซเกิลไปตามภาระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**พัลซ์-ฟรีเควนซีโมดูเลชั่น (Pulse Frequency Modulation หรือ PFM)**

ระบบ PFM จะมีการควบคุมให้ส่วนของมุมเปิดที่คงที่ และส่วนของความถี่ในการสวิทช์ (Repetition Rate) ที่ปรับค่าได้ การควบคุมในแบบที่ปรับค่าความถี่ของพัลซ์ในช่วงที่กว้างมากจะทำให้เกิดปัญหาเกี่ยวกับการสูญเสียกำลัง

การเปลี่ยนสวิทช์ชิ่งเรทหรือความถี่ในการสวิทช์ไปตามภาระ จะทำให้เกิดผลเสีย คือ

- ทำให้เกิดรีโซแนนซ์โหมดในระบบ
- ทำให้เกิดเสียงรบกวน
- ทำให้เกิดการสูญเสียกำลังในภาคเอาท์พุท

**SCR (Silicon Control Rectifier)**

วงจร SCR สำหรับการควบคุมวงจรไฟฟ้ากระแสตรง ตามปกติมักจะใช้ร่วมกับแหล่งจ่ายแรงดันไฟฟ้ากระแสสลับที่เรียงกระแสแล้ว และวงจรเรียงกระแสสามารถจะอยู่ก่อนหน้าหรือหลังส่วนของการควบคุมของการขยาย

## 2.1.4 ลิเนียร์เซอร์โวแอมปลิไฟเออร์

ลิเนียร์เซอร์โวแอมปลิไฟเออร์โดยปกติสร้างขึ้นเพื่อใช้เป็นอุปกรณ์ขยายแบบเชิงเส้น เช่นเดียวกันกับออปแอมป์ หรือดิฟเฟอร์เรนเชียลแอมปลิไฟ (Differential Amplifier) ที่ใช้ไปกระตุ้นภาคกำลังให้ไปขับมอเตอร์ ภาคกำลังจะมีการป้อนกลับแบบวงรอบปิด เพื่อทำให้ได้ค่าตัวแปรที่ต้องการ ได้แก่ แรงดันมอเตอร์ กระแสของมอเตอร์หรือพารามิเตอร์ทั้งสองดังกล่าวมีความเป็นเชิงเส้น ในกรณีนี้ความไม่เป็นเชิงเส้นและดริฟ (Driff) ที่เกิดขึ้นจากอุณหภูมิในภาคกำลังจะลดลงโดยอัตราขยายวงรอบของระบบ นอกจากนั้นออปแอมป์ ยังมักใช้เป็นเสมือนการชดเชยหรือใช้เป็นซัมมิ่งแอมปลิไฟ (Summing Amplifier) ทำหน้าที่เหมือนกับจุดรวมของสัญญาณบังคับ และสัญญาณป้อนกลับของระบบ ซึ่งได้แก่สัญญาณที่ได้จากแอนะลอกทาโคมิเตอร์ (ในระบบการควบคุมความเร็ว) และสร้างสัญญาณผิดพลาด ซึ่งจะถูกขยายโดยภาคกำลังเพื่อไปแก้ค่าตัวแปร ที่ต้องการควบคุมให้มีค่าที่ถูกต้อง

การขยายแบบเชิงเส้นจะเป็นตัวควบคุมแรงดันหรือกระแสของมอเตอร์โดยการควบคุมขนาดของแรงดันที่จ่ายให้กับมอเตอร์ดังนั้นมันจะเป็นตัวที่ลดแรงดันเอาไว้ซึ่งเท่ากับผลต่างของแหล่งจ่ายแรงดันและแรงดันที่มอเตอร์ที่ใช้งานจริงในขณะที่มีกระแสที่เหมาะสมไหลผ่านกำลังงานจะสูญเสียในเอาท์พุททรานซิสเตอร์เป็นจำนวนมากโดยเฉพาะขณะที่มอเตอร์วิ่งที่ความเร็วต่ำแรงบิดสูง เพราะขณะนั้นแรงดันย้อนกลับของมอเตอร์ต่ำกระแสจึงสูง นี้เป็นความแตกต่างที่เห็นได้ชัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อเทียบกับสวิทช์ชิ่งแอมปลิไฟ สวิทช์ชิ่งแอมปลิไฟควบคุมแรงดันของมอเตอร์โดยปรับดิ้วตี้ไซเกิลของแรงดันที่จ่ายให้กับมอเตอร์และให้มันทำงานทุกๆ ภาวะอิ่มตัวหรือที่ภาวะ OFF ดังนั้นการสูญเสียของกำลังงานจะน้อยในแต่ละภาวะ ทำให้ประสิทธิภาพการทำงานสูงขึ้นอย่างไรก็ตามสวิทช์ชิ่งแอมปลิไฟก็สร้างปัญหาใหม่ขึ้นอีก เพื่อหลีกเลี่ยงการสูญเสียการสวิทช์ที่ความถี่สูงค่าของอินดัคแต้นซ์ของมอเตอร์ (หรือภายนอก) ต้องมีค่าน้อยดังนั้นจึงเป็นการเพิ่มค่าคงที่ทางเวลาของระบบ ซึ่งจะทำให้ย่านของระบบลดน้อยลงด้วย นอกจากนั้นยังเกิดปัญหาการรบกวนของสนามแม่เหล็กไฟฟ้า (BACK EMF) ซึ่งมักจะเกิดขึ้นรุนแรงทำให้ระบบเกิดความยุ่งยากและทำงานผิดพลาดขึ้นได้ ดังนั้นจึงเป็นการยากที่จะตัดสินว่าแอมปลิไฟชนิดไหนดีกว่ากัน

การขยายแบบเชิงเส้นตามปกตินิยมใช้ในระบบที่มีย่านกว้างและกำลังทำงานต่ำ (น้อยกว่า 2-3 ร้อยวัตต์) และมักจะใช้สำหรับขับคลื่นขดลวดหมุนของมอเตอร์ที่มีความเฉื่อยต่ำให้เพิ่มกระแสด้วยอัตราเร่าสูงในช่วงเวลาอันสั้น (2-3 มิลลิวินาที) ซึ่งกระแสสูงสุดที่กำหนดไว้ของเอาท์พุททรานซิสเตอร์อาจจะยังใช้งานได้ดี

ในทางตรงกันข้ามสวิทช์ชิ่งแอมปลิไฟโดยทั่วๆ ไปมักจะใช้ในระบบที่ใหญ่ๆ โดยเฉพาะอย่างยิ่งระบบดังกล่าวต้องการขยายช่วงการที่ความเร็วต่ำและมีแรงบิดสูงซึ่งจะทำให้เกิดการสูญเสียกำลังงานอย่างมากในการขยายแบบเชิงเส้น

การขยายแบบเชิงเส้นอาจจำแนกตามหน้าที่จะแบ่งได้เป็น ยูนิไดเร็คชั่นแนลแอมปลิไฟและแบบไบไดเร็คชั่นแนลแอมปลิไฟ และยังสามารถแบ่งตามโครงสร้างได้อีก ดังต่อไปนี้

**ยูนิไดเร็คชั่นแนลเซอร์โวแอมปลิไฟเออร์**

ยูนิไดเร็คชั่นแนลเซอร์โวแอมปลิไฟ เป็นเครื่องขยายแบบง่ายที่สุดของการขยายแบบเชิงเส้น และใช้ในระบบที่มอเตอร์ถูกขับให้หมุนได้ในทิศทางเดียว นั่นคือ ฟังก์ชั่นถ่ายโอนแบบเชิงเส้นของการขยายแบบนี้จะทำงานอยู่ในควอดแดรนท์แรกเท่านั้น และไม่สามารถให้แรงดันหรือกระแสไหลกลับทิศทางได้ ระบบนี้แสดงคุณสมบัติการทำงานโดยสามารถเพิ่มอัตราเร่งได้อย่างรวดเร็ว แต่สามารถลดความเร็วลงได้อย่างเชื่องช้า เนื่องจากมอเตอร์จะลดความเร็วลงได้ด้วยแรงเสียดทานและ Viscous Damping ของระบบเท่านั้น

**ไบไดเร็กชั่นแนลเซอร์โวแอมปลิไฟเออร์**

ไบไดเร็กชั่นแนลเซอร์โวแอมปลิไฟสามารถให้มอเตอร์ทำงานได้ทั้งสองทิศทางและมีฟังก์ชั่นถ่ายโอนเชิงเส้น ตลอดถึงช่วงความเร็วเป็นศูนย์ แอมปลิไฟแบบนี้สามารถบังคับได้เต็มทั้ง 4 ควอดรันท์ และสามารถป้อนแรงดันและกระแสทั้งขั้วบวกและขั้วลบให้กับมอเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไบไดเร็กชั่นแนลแอมปลิไฟแบบพื้นฐาน 2 แบบ ได้แก่ ลิเนียร์ไบโพล่าร์ (Linear Bipolar) และลิเนียร์บริดจ์ (Linear Bridge) ซึ่งแบ่งออกเป็นลักษณะโครงสร้างของภาคเอาท์พุท

ไบโพล่าร์แอมปลิไฟจะใช้แหล่งจ่ายไฟฟ้าถึงสองลักษณะสำหรับการทำงานแบบ Forward และ Reverse แรงดันที่ป้อนให้มอเตอร์ เท่ากับผลต่างระหว่างแหล่งจ่ายแรงดันเนื่องด้วยลิเนียร์แอมปลิไฟโดยทั่วไปมีประสิทธิภาพไม่ค่อยดีนัก โดยเฉพาะอย่างยิ่งที่สภาวะความเร็วต่ำแรงบิดสูง ซึ่งในสภาวะนี้กระแสจะไหลสูงและแรงดันตกคร่อมทรานซิสเตอร์สูงด้วยดังนั้นในระบบใหญ่จำเป็นต้องใช้ตัวระบายความร้อนขนาดใหญ่หรือใช้พัดลมระบายความร้อนด้วย

ลิเนียร์ไบโพล่าร์แอมปลิไฟเป็นแอมปลิไฟที่ออกแบบได้ง่ายกว่าลิเนียร์บริดจ์แต่การออกแบบไบโพล่าร์มีข้อเสียที่สำคัญหลายข้อซึ่งควรประเมินข้อดีข้อเสียของระบบเสียก่อน

1. แหล่งจ่ายไฟฟ้าเป็นแบบไบโพล่าร์ คือมีสองขั้ว บวกและลบ
2. ค่า $v_{ceo}$ ของกำลังทรานซิสเตอร์ ต้องสูงกว่าผลบวกของแหล่งจ่ายแรงดันทั้งสอง ($2v_s$)
3. ลิเนียร์ไบโพล่าร์แอมปลิไฟจะต้องทนต่อ การพังทลายในการไบอัสตรงที่เอาท์พุทของทรานซิสเตอร์กำลัง อาจจะเสียได้ในช่วงการลดความเร็วของภาระที่มีความเฉื่อยสูงลงจากความเร็วสูงๆ

## 2.1.5 พัลซ์วิดท์โมดูเลชั่นแอมปลิไฟเออร์ (PWMA)

ในระบบ "ดีซีเซอร์โว" แอมปลิไฟจะมีหน้าที่ควบคุมกระแสและแรงดันที่ใช้ในการขับเคลื่อนมอเตอร์ เพื่อให้ได้ค่าความเร็วตามที่ต้องการ เรารู้จักลิเนียร์แอมปลิไฟกันมาแล้ว ซึ่งเป็นแอมปลิไฟที่มีความสัมพันธ์อย่างง่ายๆ และมีช่วงย่านกว้างมาก แต่สมบัติแอมปลิไฟมีข้อจำกัดบางประการ เพราะแอมปลิไฟต้องสูญเสียพลังงานไปที่เอาท์พุททรานซิสเตอร์มาก เช่น มอเตอร์ตัวหนึ่งต่ออนุกรมกับแอมปลิไฟโดยใช้แหล่งจ่ายร่วมกัน แหล่งจ่ายจะจ่ายไฟ 50 โวลท์ มอเตอร์ใช้ไฟไป 10 โวลท์ และกระแส 20 แอมแปร์ มอเตอร์จึงกินพลังงานไป 200 วัตต์ แต่จะเห็นว่ามีการสูญเสียที่แอมปลิไฟถึง 800 วัตต์ คือจะมากเป็น 4 เท่าของมอเตอร์

วิธีหนึ่งที่จะแก้ปัญหาการสูญเสียพลังงานคือการใช้แอมปลิไฟควบคุมมอเตอร์โดยการเปลี่ยนดิวตี้ไซเกิล Duty Cycle ของแรงดันที่จ่ายให้กับมอเตอร์ แอมปลิไฟชนิดนี้เรียกว่า "สวิทช์ชิ่งแอมปลิไฟ" ซึ่งสามารถควบคุมความเร็วต่ำๆ โดยมีแรงบิดสูงโดยไม่สิ้นเปลืองพลังงานเหมือนพวกลิเนียร์แอมปลิไฟ

สวิทช์ชิ่งแอมปลิไฟนี้ทรานซิสเตอร์แอมปลิไฟจะทำหน้าที่คล้ายสวิทช์ เมื่อทรานซิสเตอร์นำกระแสแรงดันตกคร่อมตัวมันจะน้อยมากจนตัดทิ้งได้ แต่เมื่อนำกระแสแรงดันที่ตกคร่อมตัวมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีค่ามากเข้าใกล้ $v_{cc}$ และมีกระแสไหลผ่านน้อยมากถือว่าเป็นศูนย์ ด้วยเหตุดังกล่าวการสูญเสียในเอาท์พุททรานซิสเตอร์จึงมีค่าต่ำ ระบบสวิทช์ซึ่งสามารถสร้างได้หลายวิธีมีทฤษฎีง่ายทฤษฎีหนึ่งในการให้แอมปลิไฟสวิทช์ด้วยความถี่ที่คงที่และสามารถแปรค่าของ ON และ OFF ของพัลซ์ได้ตามต้องการแอมปลิไฟชนิดนี้เรียกว่า "พัลซ์วิดท์โมดูเลเตอร์แอมปลิไฟ" ซึ่งเราสามารถออกแบบให้แปรค่าได้ทั้งค่าพัลซ์และความถี่ก่อประโยชน์ในการควบคุมรักษาระดับกระแสสูงๆได้ แต่อาจเกิดออสซิเลท หรือการทำให้เกิดเสียงร้องที่ความถี่รีโซแนนซ์ ดังนั้นในบทนี้จะกล่าวถึงแอมปลิไฟแบบ PWM ที่มีความถี่คงที่เท่านั้น

แอมปลิไฟแบบ PWM สามารถทำงานได้ 3 แบบ คือ ไบโพล่าร์ ยูนิโพล่าร์และลิมิตยูนิโพล่าร์ ซึ่งในบทนี้เราจะได้อธิบายรายละเอียดในแต่ละแบบ เราจะได้ศึกษาปัญหาการสูญเสียในแอมปลิไฟตลอดจนปัญหาการรักษาความถี่ในการสวิทช์ให้ถูกต้อง

จากการศึกษาการใช้งานกับดีซีมอเตอร์นี้เราสามารถนำไปประยุกต์ใช้กับสเตปมอเตอร์และใช้เป็นแอมปลิไฟของลิเนียร์มอเตอร์ได้อีกด้วย

### 2.1.6 การทำงานของพัลซ์วิดท์โมดูเลชั่นแอมปลิไฟเออร์

แอมปลิไฟแบบ PWM สามารถแบ่งได้เป็น 3 ชนิด ตามลักษณะของการทำงานคือไบโพล่าร์ ยูนิโพล่าร์และจำกัดยูนิโพล่าร์ สำหรับไบโพลล่าร์ซึ่งง่ายที่สุดจะดูการทำงานได้ตามรูปที่ที่ 7.38 โดยที่เราจะกำหนดให้มีความถี่กมารสวิทช์เป็น $f_s$ $t_{on}$ ที่เกิดขึ้นในส่วนแรกและ $t_{off}$ เกิดในส่วนหลังโดย

$$t_{on} \quad \text{เมื่อ} \quad 0 < t < t_1$$
$$t_{off} \quad \text{เมื่อ} \quad t_1 < t < t_f$$

ไบโพล่าร์จะมี $T_1$ และ $T_4$ นำกระแสระหว่างเฟส ON ส่วน $T_2$ และ $T_3$ จะนำกระแสขณะเฟส OFF จะไดฟังก์ชั่นตกคร่อมมอเตอร์เป็น

$$V_m = V_{AB} \begin{cases} V_s \quad 0 < t < t_1 \\ -V_s \quad t_1 < t < t_f \end{cases}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบยูนิโพล่าร์ จะลดจำนวนทรานซิสเตอร์ในการสวิทช์ลงการสวิทช์ขึ้นกับ $V_{in}$ เป็นบวกหรือลบ เมื่อ $V_{in}$ เป็นบวก $T_4$ จะนำกระแสตลอดคาบ ในขณะที่ $T_1$ นำกระแสในช่วงเฟส ON และ $T_2$ จะนำกระแสในช่วงเฟส OFF เมื่อ $V_{in}$ เป็นลบ $T_2$ จะนำกระแสตลอดโดยมี $T_3$ และ $T_4$ สลับกันทำงานเมื่อ $V_{in}$ เป็นบวกได้

$$V_{in} \begin{cases} V_s & 0 < t < t_1 \\ 0 & t_1 < t < t_f \end{cases}$$

การแสดงค่า $V_{in}$ ในทางลบจะเหมือนกันเพียงแต่ $V_{in}$ เป็นลบเท่านั้น

จากลักษณะของ 2 แบบดังที่กล่าวมานั้นมีประโยชน์เหมือนกัน ซึ่งในแต่ละกรณีจะมีทรานซิสเตอร์คู่หนึ่ง $(T_1,T_2)$ และ $(T_3,T_4)$ จะหยุดนำกระแสขณะที่อีกคู่นำกระแสซึ่งมีเวลาเก็บสระสมและเวลาที่ปล่อยออกมาของทรานซิสเตอร์ เกิดขึ้นและมันอาจเป็นไปได้ในทรานซิสเตอร์ทั้งหมดนำกระแสในเวลาเดียวกัน ซึ่งจะทำให้เกิดการลัดวงจรของแหล่งจ่าย เราจำเป็นต้องหลีกเลี่ยงถาวะดังกล่าว ซึ่งสามารถสร้างโดยการสร้างช่วง Delay Time ระหว่างการหยุดและการนำกระแสของทรานซิสเตอร์และด้วยเหตุดังกล่าว ความถี่ของการสวิทช์จะถูกจำกัดในวงที่แคบลง

แบบที่ 3 ของการทำงานคือ แบบจำกัดยูนิโพล่าร์จะแสดงให้เห็นคือมีความจำเป็นต้องมีช่วง Delay ซึ่งการสวิทช์ขึ้นอยู่กับค่า $V_{in}$ เมื่อ $V_{in}$ เป็นบวก $T_4$ จะนำกระแสตลอด $T_1$ จะสวิทช์เป็น ON ในช่วงเฟส ดังนั้นในช่วงเฟส ON ทั้ง $T_1$ และ $T_4$ จะ ON ยังผลแก่แรงดันของมอเตอร์ $V_m$ คือ

$$V_m = V_s \quad 0 < t < t_1$$

ระหว่างเฟส OFF จะมี $T_4$ นำกระแสเพียงตัวเดียวเป็นผลให้ $V_m$ ขึ้นกับ $I_{AB}$ ตราบใดที่ $I_{AB} > 0$ ซึ่งเป็นสภาวะปกติเมื่อ $V_{AB} > 0$ กระแส $I_{AB}$ จะไหลผ่าน $D_2$ และ $T_4$ เป็นผลให้ $V_a = 0$ และ

$$V_m = V_{AB} = 0 \text{ เมื่อ } \begin{cases} t_1 < t < t_f \\ I_{AB} > 0 \end{cases}$$

ในกรณีที่ $I_{AB}$ เป็นลบ กระแสจะไหลผ่าน $D_1$ และ $D_4$ เป็นผลให้ $V_A = V_S$ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V_m = V_{AB} = V_S \text{ เมื่อ } \begin{matrix} t_1 < t < t_f \\ I_{AB} > 0 \end{matrix}$$

ซึ่งจะเกิดขึ้นภายหลังเปลี่ยนขั้ว $V_m$

ในที่สุดถ้าเราสามารถทำให้ $I_{AB} = 0$ (เข้าใกล้ศูนย์จนถือเป็นศูนย์) จะทำให้ทั้ง $D_1$ และ $D_4$ ไม่นำกระแสและแรงดัน $V_m$ จะอยู่ระหว่างค่าศูนย์และ $V_s$ ดังต่อไปนี้

$$0 < V_m < V_s \text{ เมื่อ } \begin{matrix} t_1 < t < t_f \\ I_{AB} = 0 \end{matrix}$$

## 2.1.7 ระบบควบคุมความเร็วที่ใช้พัลซ์วิดท์โมดูเลชั่นแอมปลิไฟเออร์

PWM สวิทช์ซิ่งแอมปลิไฟเออร์ที่ทำงานด้วยทรานซิสเตอร์จะใช้ในระบบควบคุมที่มีประสิทธิภาพสูง ระบบควบคุมความเร็วที่มีกำลังสูงๆ และในระบบเซอร์โว

เรามักจะใช้ความถี่ของพัลซ์ในระบบ PWM ที่มีค่าสูงกว่า 1 กิโลเฮิรตซ์ (มักจะมีค่าถึง 10 กิโลเฮิรตซ์) และความถี่นี้จะขึ้นอยู่กับความต้องการของผลตอบสนองของระบบต่อย่าน อินดัคแตนซ์ของมอเตอร์และคุณสมบัติการสูญเสียกำลังในตัวมอเตอร์ที่ความสูงในเวลาเดียวกัน เราจะต้องคำนึงถึง Audio Noise ที่เกิดจากขดลวด, ตัวระบายความร้อน และส่วนของโครงประกอบของตัวมอเตอร์ มันจะดังออกมาเป็นเสียงรบกวนและในการประยุกต์ใช้งานของระบบ PWM นี้เราสามารถเพิ่มความถี่ของพัลซ์ให้สูงขึ้นถึงจุดที่เสียงของ Noise นั้น หูคนไม่ได้ยินเสียงนั้น

ในการพิจารณาคุณสมบัติของแรงดันและกระแสของระบบ PWM ขั้นแรกเราพิจารณาถึงมอเตอร์ในอุดมคติและคุณลักษณะของมอเตอร์ในระบบ PWM วงจรสมมูลย์ของมอเตอร์

ลูกคลื่นของกระแสที่ไหลในมอเตอร์ในระหว่างสวิทช์ซิ่งโหมด ไม่ได้ขึ้นอยู่เพียงสวิทช์ซิ่งเรทเท่านั้น แต่ยังขึ้นอยู่กับความเร็วของมอเตอร์ (W) ผลรวมของอินดัคแตนซ์ (L) ความต้านทานของมอเตอร์ ($R_a$) และระดับของกระแสในไซเกิลสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราจะต้องคำนวณหาการสูญเสียกำลังในตัวมอเตอร์ไฟฟ้ากระแสตรง เนื่องจากแหล่งจ่ายแรงดันไฟฟ้ากระแสตรงจะถูกสวิทช์เปิดและปิดที่ความถี่สูง การสูญเสียกำลังในระบบอาจจะเกิดจากตัวประกอบที่สำคัญดังนี้ (ขึ้นอยู่กับการออกแบบของมอเตอร์)

- Eddy Current Losses
- Hysteresis Losses
- Armature Commutation Losses
- Vicous Friction Losses
- Armture Resistance Losses

ในกรณีที่ PWM แอมปลิไฟเออร์ รูปแบบของตัวประกอบจะขึ้นอยู่กับ ความถี่ของพัลซ์, Electrical Time Constant ของมอเตอร์ และค่าความเหนี่ยวนำที่ต่ออนุกรมกับมอเตอร์

ผลจากการใช้วงจร PWM จะทำให้การสูญเสียกำลังในรูปของความร้อนในตัว แอมปลิไฟเออร์ลดลงอย่างมาก และเมื่อคิดถึงผลรวมของการสูญเสียกำลังทั้งหมดของระบบแล้วดีขึ้น (ผลรวมของการสูญเสียน้อยลง) แต่ในบางกรณีการสูญเสียกำลังในตัวมอเตอร์อาจจะมีค่าสูงกว่าในระบบที่ใช้การขยายแบบเชิงเส้น

มีตัวประกอบอื่นๆ ที่เราจะต้องพิจารณาเมื่อใช้ระบบ PWM คือ การกำเนิด Electrical Noise ซึ่งจะไปรบกวนในวงจรที่มีระดับกำลังต่ำ ถ้าไม่ได้ให้ความระมัดระวังเกี่ยวกับการ Shield ที่เหมาะสมและการ Ground ส่วนที่มีกระแสสูงของระบบ

## 2.2 องค์ประกอบของระบบเซอร์โว (Elements of Servo System)

จุดมุ่งหมายในบทนี้เป็นการอธิบายถึงส่วนประกอบต่างๆ ของระบบเซอร์โว เช่น ดีซี เซอร์โวมอเตอร์ (ดีซีมอเตอร์แบบต่างๆ) เอซีเซอร์โวมอเตอร์ (อินดัดชั่นมอเตอร์แบบ 2 เฟส) สเตปปิ้ง (Stepping) มอเตอร์แบบต่างๆ และพวกส่วนประกอบประเภทเชิงกลที่ใช้กันทั่วๆ ไประบบการคอนโทรลที่มีการป้อนกลับ เช่น ระบบลีดสกรู ระบบเฟือง ระบบสายพาน และระบบเกียร์ เป็นต้น

### 2.2.1 หลักการทำงานของดีซีมอเตอร์

ดีซีมอเตอร์เป็นทรานสดิวเซอร์แรงบิดซึ่งมีการออกแบบให้มีคุณลักษณะพิเศษ คือ แรงบิดของเพลาของดีซีมอเตอร์จะเป็นสัดส่วนโดยตรงกับกระแสอาร์มาเจอร์ แรงบิดของเพลาของดีซีมอเตอร์จะได้จากผลระหว่างสนามแม่เหล็กและขดลวดตัวนำ หลักการนี้แสดงได้ในรูปที่ 2.2 ในที่นี้กระแสที่ไหลในขดลวดตัวนำจะสร้างฟิลด์ที่ประกอบด้วยเส้นแรงแม่เหล็ก $\phi$ และขดลวดตัวนำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหล่านั้นอยู่ห่างจากศูนย์กลางการหมุนเท่ากับ r ความสัมพันธ์ระหว่างแรงปิดของเพลาและกระแสเท่ากับ

$$T = K\phi I$$

เมื่อ T คือ แรงบิดของเพลามีหน่วยเป็นนิวตัน-เมตร (N-M)

$\phi$ คือ เส้นแม่เหล็กมีหน่วยเป็นเวเบอร์ (Wb)

I คือ กระแสเป็นแอมแปร์ (A)

K คือ ตัวคงที่

ดังนั้นแรงปิดของเพลาจะเป็นสัดส่วนโดยตรงกับผลคูณของเส้นแม่เหล็กและกระแสเมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดโวลท์เต็จตกคร่อมตัวมันเองโวลท์เต็จนี้จะเป็นสัดส่วนกับความเร็วของเพลาของมอเตอร์และต้านการไหลของกระแส ความสัมพันธ์ระหว่างโวลท์เต็จย้อนกลับนี้และความเร็วของเพลามอเตอร์คือ

$$E = K\phi\ \omega$$

เมื่อ E คือ โวลท์เต็จย้อนกลับ emf มีหน่วยเป็นโวลท์ (V)

$\theta$ คือ เส้นแรงแม่เหล็กมีหน่วยเป็นเวเบอร์ (Wb)

$\omega$ คือ ความเร็วของมอเตอร์มีหน่วยเป็นเรเดียน/วินาที (rad/s)

### 2.2.2 โมเดลคณิตศาสตร์ของมอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงที่ใช้ร่วมกับดีซีแอมปลิไฟทั้งในระบบการบังคับตำแหน่งและการบังคับความเร็วมักจะได้รับการประยุกต์ใช้เป็นส่วนประกอบสร้างกำลังงานในระบบการนำร่องและระบบการบังคับต่างๆ และเนื่องจากวิทยาการเกี่ยวกับสารแม่เหล็กและการขยายด้วย Solic State ทำให้มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวรได้รับความนิยม ใช้เป็นส่วนประกอบการขับเคลื่อนในระบบการบังคับแบบวงรอบปิดต่างๆ มากขึ้น การออกแบบและการชดเชยระบบดังกล่าวได้อย่างเหมาะสมจะต้องใช้โมเดลคณิตศาสตร์ของส่วนประกอบทั้งหมดในระบบ ในหัวข้อนี้เราจะได้พัฒนาลิเนียร์โมเดลของดีซีมอเตอร์แบบแม่เหล็กถาวรและแบบฟิลด์แยกกระตุ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**โมเดลอิเล็คโทรแมคคานิคอล (Model Electro Machanical)**

ส่วนสำคัญของดีซีมอเตอร์แบบฟิลด์แยกกระตุ้นมีโมเดล

$$V_i(t) = R_a I_a(t) + L_a \frac{d_{i(t)}}{dt} + V_g(t)$$

เมื่อ $R_a$ คือ ความต้านทานของอาร์มาเจอร์

$L_a$ คือ อินดัดแต้นซ์ของอาร์มาเจอร์

$V_g$ คือ โวลท์เต็จกำเนิดในอาร์มาเจอร์ (โวลท์เต็จย้อนกลับ)

$R_f$ คือ ความต้านทานของฟิลด์

$L_f$ คือ อินดัดแตนซ์ของฟิลด์

$\phi$ คือ ช่องว่างอากาศของเส้นแรงสนามแม่เหล็ก

$\omega$ คือ ความเร็วของเพลาอาร์มาเจอร์

$T_g$ คือ แรงบิดที่พัฒนาขึ้นในมอเตอร์

$T_f$ คือ แรงบิดเสียดทานของมอเตอร์

$T_j$ คือ แรงเฉื่อยของมอเตอร์

$T_L$ คือ แรงบิดภาระบนเพลาของมอเตอร์

เทอมโวลท์เต็จ $V_g(t)$ ในสมการคือ โวลทเต็จย้อนกลับของมอเตอร์ซึ่งเกิดขึ้นเมื่อเส้นลวดตัวนำของอาร์มาเจอร์หมุนตัดเส้นแรงแม่เหล็กซึ่งเกิดขึ้น โดยกระแสของฟิลด์ ($I_f$) ตามกฎของฟา - ราเดย์ลูปของเส้นลวดตัวนำหมุนในฟิลด์แม่เหล็กคงที่จะมีการเหนี่ยวนำโวลท์เต็จขึ้นในขดลวดนั้น

$$V(t) = \frac{d\lambda(t)}{dt}$$

เมื่อ $\lambda(t)$ เส้นแรงแม่เหล็กที่รั่วไปยังขดลวดและ t คือเวลาในการหมุนของคอมมิวเทเตอร์ของมอเตอร์ การควบคุมวงจรของแต่ละส่วนของตัวนำในโรเตอร์จะเกิดโวลท์เต็จขึ้นในส่วนของตัวนำนั้นตามสมการเมื่อ $d\lambda(t)/dt$ จะเป็นสัดส่วนต่อเส้นแรงแม่เหล็กในช่องว่างอากาศและความเร็วเชิงมุม $\omega(t)$ หรือเราจะได้ว่า

$$V_g(t) = K\phi(t)\omega(t)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



สมมติให้กระแสของฟิลด์มีค่าคงที่และไม่คิดถึงส่วนการเปลี่ยนแปลงในเส้นแรงฟิลด์เนื่องจากอาร์มาเจอร์รีแอคชั่น (Armature Reaction) เส้นแรงฟิลด์ก็จะมีค่าคงที่ดังนั้นสมการ ก็จะกลายเป็น

$$V_g(t) = K_e\,\omega(t)$$

เมื่อเราสมมติให้เส้นแรงของฟิลด์มีค่าคงที่ แรงบิดของแม่เหล็กไฟฟ้าซึ่งเกิดขึ้นแก่โรเตอร์ของมอเตอร์จะเป็นสัดส่วนกับกระแสอาร์มาเจอร์

$$T_g(t) = K_t\,i_a(t)$$

เมื่อ K คือค่าคงที่ของแรงบิดของมอเตอร์

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์คือผลคูณของแรงบิดที่เกิดขึ้นและความเร็วเชิงมุม

$$P_g(t) = T_g(t)\,\omega(t)$$

กำลังงานเชิงกลที่เกิดขึ้นในโรเตอร์ทั้งหมดนี้จะจ่ายไปยังภาระที่ต่ออยู่กับเพลาของมอเตอร์ แต่กำลังงานนี้บางส่วนจะสูญเสียไปในมอเตอร์ การสูญเสียเนื่องมาจากแรงเสียดทาน หมายถึงความหน่วงเนื่องจากลมที่มีต่อโรเตอร์ แรงเสียดทานตัวรองรับโรเตอร์ กระแสที่ไหลวนในเหล็กของโรเตอร์และอิสเทรีซิส

$$T_g(t) = T_f(t) + T_L(t) + J\frac{d\omega(t)}{dt}$$

เมื่อ $T_g(t)$ คือ แรงบิดของมอเตอร์

$T_f(t)$ คือ แรงบิดที่ต้องชนะการสูญเสียเนื่องจากการเสียดทาน

$T_J(t)$ คือ แรงบิดเพื่อใช้เพิ่มอัตราเร่งแก่ ความเฉื่อยของภาระ

$T_L(t)$ คือ แรงบิดภาระ

ในช่วงเวลาใดๆ ก็ตาม แรงบิดของมอเตอร์จะต้องเท่ากับและมีทิศทางตรงข้ามกับผลรวมของแรงบิด $T_f(t)$ $T_J(t)$ และ $T_L(t)$ ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

40200

เมื่อ J คือ ผลรวมของ โมเม้นท์แรงเฉื่อยของโรเตอร์และภาระที่ต่ออยู่ที่เพลาของมอเตอร์

ผลรวมของแรงบิดเสียดทานที่ประกอบกันชิ้นที่เพลาของมอเตอร์ซึ่งเป็นลิเนียร์ฟังก์ชั่นความเร็วเชิงมุมของโรเตอร์เรียกว่า ส่วนประสอบของวิสคอสฟริคชั่นและมักจะอยู่ในเทอมที่แยกออกจากฟริคชั่นอื่นๆ ซึ่งแสดงได้ด้วยสมการต่อไปนี้

$$T_g(t) = T_f(t) + T_L(t) + J\frac{d\omega(t)}{dt} + B\omega(t)$$

เมื่อ B คือ สัมประสิทธิ์ของวิสคอสฟริคชั่นของมอเตอร์และภาระที่ต่ออยู่กับเพลาของมอเตอร์และภาระที่ต่ออยู่กับเพลาของมอเตอร์

$T_f(t)$ คือ ผลรวมของฟริคชั่นของภาระและของมอเตอร์ทั้งหมด มีแรงต้านของลมและการสูญเสียกำลังในเหล็กของเพลามอเตอร์ยกเว้นวิสคอสฟริคชั่น (Viscos Friction)

สมการดังกล่าวเป็นชุดสมการพื้นฐานของดีซีมอเตอร์โมเดลและจากสมเหล่านี้เราสามารถจะหาทรานสเฟอร์ฟังก์ชั่น (Transfer Function) ของดีซีมอเตอร์ได้ โดยใส่ลาปลาสทรานสฟอร์ม (Laplace Transform) ทั้งสองข้างของชุดสมการพื้นฐานและเขียนได้ใหม่เป็น

$$\begin{aligned} V_1(s) - V_g(s) &= (R_a + sL_a)\, I_a(s) \\ V_g(s) &= K_e\,\omega(s) \\ T_g(s) &= K_t\, I_a(s) \\ T_g(s) - T_f(s) - T_L(s) &= (B + sJ)\,\omega(s) \end{aligned}$$

**ข้อควรสังเกต**

สมมติว่าโวลท์เต็จที่ป้อนให้กับวงจรอาร์มาเจอร์ของมอเตอร์มีค่าคงที่ดังนั้นมอเตอร์จะหมุนด้วยความเร็วคงที่คือ ทำงานอยู่ที่สภาวะสงบนิ่งด้วยภาระที่คงที่ กำลังงานเชิงกลที่เกิดขึ้นโดยโรเตอร์หาได้จากสมการ ก็จะได้

$$P_g = T_g\,\omega = K_t I_a\,\omega$$

เมื่อทุกเทอมในสมการสุดท้ายมีค่าคงที่เนื่องจากมอเตอร์ทำงานอยู่ที่สภาวะสงบนิ่งกำลังไฟฟ้าที่ถูกดูดกลืนโดยอาร์มาเจอร์ต้องเท่ากับผลคูณของโวลท์เต็จคร่อมอาร์มาเจอร์และการะแสที่ไหน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านดังนั้น

$$P = V_g I_a = K_e \omega I_a$$

ดังนั้นเราจะได้ว่า กำลังงานเชิงกลที่เกิดขึ้นต้องเท่ากับกำลังไฟฟ้าที่ถูกดูดกลืนในโรเตอร์ คือ สรุปได้ว่า $K_t = K_e$

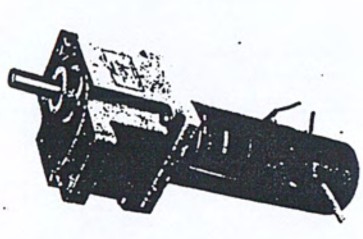
**ฟังก์ชั่นถ่ายโอนของมอเตอร์ไฟฟ้ากระแสตรง**

ในระบบที่มีสองอินพุท และมีเอาท์พุทเป็นทั้งความเร็วเชิงหมุน $\omega$ และการเคลื่อนที่แบบเชิงมุม $\theta$ ความเร็วเอาท์พุทของระบบเขียนได้เป็น

$$\omega(s) = G_1(s)V_1(s) + G_2(s)\,[T_f(s) + T_L(s)]$$

เมื่อ G (s) คือ ฟังก์ชั่นถ่ายโอนระหว่างโวลท์เต็จและความเร็ว

**รูปที่ 2.3** ดีซีมอเตอร์และทาโคมิเตอร์สำหรับใช้ในระบบบังคับแบบมีการป้อนกลับ



**รูปที่ 2.4** มอเตอร์ไฟฟ้ากระแสตรงที่มีเกียร์บล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การสูญเสียกำลังงานของมอเตอร์ไฟฟ้ากระแสตรง**

ในหัวข้อนี้เราจะได้กำหนดให้มอเตอร์ไฟฟ้ากระแสตรงเป็นเสมือนอุปกรณ์แปลงพลังงาน คือ จะแปลงพลังงานอินพุททางไฟฟ้าไปเป็นพลังงานเอาท์พุททางเชิงกล

ผลรวมของกำลังงานไฟฟ้าอินพุทเท่ากับ ผลคูณของอาร์มาเจอร์โวลท์เต็จและกระแสอาร์มาเจอร์ดังนั้นผลรวมของอินพุทเพาเวอร์ได้เท่ากับ

$$P_i(t) = V_i(t) i_a(t)$$

เราจะได้สมการโวลท์เต็จของวงจรอาร์มาเจอร์ซึ่งสัมพันธ์กับโวลท์เต็จอินพุทและการแสอาร์มาเจอร์

$$V_i(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + K_e \omega(t)$$

แทนค่าสมการ จะได้นิพจน์ของกำลังงานอินพุทรวมในเทอมของกระแสอาร์มาเจอร์และความเร็วเชิงมุมของเพลา

$$P_i(t) = R_a i_a^2(t) + L_a i_a \frac{di_a(t)}{dt} + K_a i_a(t)$$

กระแสอาร์มาเจอร์สามารถเขียนให้อยู่ในเทอมของพารามิเตอร์เชิงกลของมอเตอร์และแรงบิดของเพลาด้วยการรวมสมการ

$$i_a(t) = \frac{1}{K_t}\left[ J\frac{d\omega(t)}{dt} + B(t) + T_f(t) + T_L(t) \right]$$

ซึ่งเมื่อเราแทนค่าลงในสมการ จะได้ดังต่อไปนี้

$$P_i(t) = R_a i_a^2(t) + \frac{K_e B\omega^2(t)}{K_t} + \frac{K_e\omega(t)T_f(t)}{K_t} + \frac{K_e\omega(t)T_f(t)}{K_t}$$

$$+ \frac{K_e}{K_t} J\omega(t)\frac{d\omega(t)}{dt} + L_a i_a(t)\frac{di_a(t)}{dt}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาถึงผลรวมของกำลังอินพุทในสมการ

**เทอมแรก** $I^2R$ : การสูญเสียในขดลวดอาร์มาเจอร์เนื่องจากการไหลของกระแสอาร์มาเจอร์ การสูญเสียทางเชิงกลในตัวมอเตอร์ ประกอบด้วยการสูญเสียของกำลังงานในเทอมที่สอบและเทอมที่สาม

**เทอมที่สอง** แสดงถึงการสูญเสียกำลังงานเนื่องจากวิสคอสฟริคชั่น

**เทอมที่สาม** แสดงถึงการสูญเสียกำลังงานเนื่องจากแรงบิดเสียดทานอื่นๆ ทั้งหมด

**เทอมที่สี่** แสดงถึงเอาท์พุทเชิงกลของมอเตอร์

สองเทอมหลังแสดงถึงการไหลเข้าหรือออกของกำลังงานในอุปกรณ์สะสมพลังงานในที่นี้ได้แก่อินดัดแต้นซ์ และแรงเฉื่อยของเพลา ถ้าในช่วงเริ่มต้นจนกระทั่วสิ้นสุดของคาบเวลาที่กำหนดให้ใดๆ พลังงานที่สะสมอยู่ในส่วนของอินดัดแต้นซ์และแรงเฉื่อยของเพลาจะมีค่าคงที่ ดังนั้นจึงไม่มีการเปลี่ยนแปลงของพลังงานที่สะสมอยู่ ซึ่งก็จะไม่เกิดผลของการสูญเสียกำลังงาน ดังนั้นผลของกำลังเฉลี่ยที่สูญเสียให้แก่ แรงเฉื่อยของเพลาและอินดัดแตนซ์มีค่าเป็นศูนย์

ประสิทธิภาพ (Efficiency) ของมอเตอร์ที่ทำหน้าที่เป็นเสมือนตัวแปลงพลังงาน สามารถคำนวณได้จากอัตราส่วนของเพาเวอร์เอาท์พุทเชิงกลต่อ ผลรวมของเพาเวอร์อินพุทเฉลี่ย

$$\text{ประสิทธิภาพ (กำลังงาน)} = \frac{TL}{\dfrac{R_a I_a^2 K_t}{K_e} + B\omega + T_f + T_L} \times 100\%$$

เมื่อพารามิเตอร์ทั้งหมดพิจารณาได้ว่ามีค่าคงที่

ประสิทธิภาพของการแปลงกำลังไฟฟ้าเป็นกำลังงานเชิงกลสามารถหาดได้จากอัตราส่วนของเพาเวอร์เอาท์พุทเชิงกลที่พัฒนาขึ้นโดยโรเตอร์

$$\text{ประสิทธิภาพในการแปลงทางกำลังงานเชิงกล} = \frac{T_L}{B\omega + T_f + T_L} \times 100\%$$

### 2.2.3 การหลีกเลี่ยงข้อผิดพลาดในการเลือกขนาดของมอเตอร์

เรามักจะใช้มอเตอร์ขับภาระโดยผ่านระบบเกียร์ เพื่อที่จะกำหนดแรงบิดที่ใช้เพิ่มอัตราเร่งของภาระในระบบดังกล่าว ค่าอินเนอร์เชีย (แรงเฉื่อย) ของภาระรวมถึงแรงเฉื่อยของแกนมอเตอร์ตามปกติจะมีค่าเป็นสัดส่วนกับค่า $1/N^2$ ดังนั้นสมการทั่วๆ ไปที่ใช้คำนวณหาแรงบิดของมอเตอร์ที่ต้องการจะมีค่าเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$T_s = \frac{T_{Lf}}{Nn} + J_M a_M + \frac{J_L a_M}{N^2}$$

เมื่อ $T_s$ = เป็นแรงบิดที่แกนมอเตอร์ต้องการ

$T_{Lf}$ = ฟริคชั่นของภาระ

$J_M$ = แรงเฉื่อยของมอเตอร์

$J_L$ = แรงเฉื่อยของภาระ

$a_M$ = อัตราเร่งของแกนมอเตอร์

N = อัตราส่วนของเกียร์ (ควาทเร็วมอเตอร์/ความเร็วภาระ)

n = ประสิทภาพของเกียร์บล็อค

สมการจะมีความเที่ยงตรงที่ดีเมื่อ n = 1 เมื่อเทอมที่สามมีค่าน้อยเมื่อเทียบกับอีกสองเทอม อย่างไรก็ตามเมื่อ n เกิดมีค่าน้อยกว่า 1 และเทอมที่สามมีค่าสูงในระบบคอนโทรลการเคลื่อนที่ใช้งานกับส่วนมากมักต้องการแรงบิดของมอเตอร์ที่มีค่าสูงกว่าค่าที่คำนวณได้ในสมการ

เออร์เรอร์เกิดขึ้นเนื่องจากแรงบิดที่ต้องการของแกนมอเตอร์เพื่อที่จะเพิ่มอัตราเร่งให้กับภาระจะถูกกระทบกระเทือนจากประสิทธิภาพของเกียร์บล็อค ดังนั้นจากสมการเขียนใหม่ได้เป็น

$$T_s = \frac{T_{Lf}}{Nn} + J_M a_M + \frac{J_L a_M}{N^2 n}$$

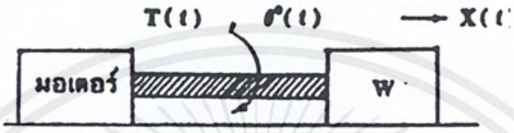
ผลกระทบที่เกิดขึ้นจากประสิทธิภาพของเกียร์บล็อคจะแสดงให้เห็นได้จากตัวอย่างของสเตปมอเตอร์ที่มีแรงเฉื่อยของมอเตอร์เท่ากับ 0.5 หน่วยและแรงเฉื่อยของภาระเท่ากับ 50 หน่วย มีอัตราส่วนของเกียร์เท่ากับ 10:1 ค่า $T_{Lf}$ = 0 และค่า a = 1 แทนค่าเหล่านี้ลงในสมการต่างๆ เราสามารถพล็อตความสัมพันธ์ระหว่างแรงบิดที่ต้องการจริงๆ กับแรงบิดจากสมการได้

ประสิทธิภาพของเกียร์มีค่าต่ำ แรงบิดที่ต้องใช้เพิ่มอัตราเร่งของมอเตอร์และภาระจะมีค่าสูงกว่าที่คำนวนได้ในสมการ ส่วนเพาเวอร์ที่สูญเสียในตัวมอเตอร์จะแปรค่าเป็นกำลังสอบของแรงบิด และเมื่อประสิทธิภาพของเกียร์มีค่าสูงๆ เพาเวอร์สูญเสียก็จะมีอิทธิพลน้อยลง

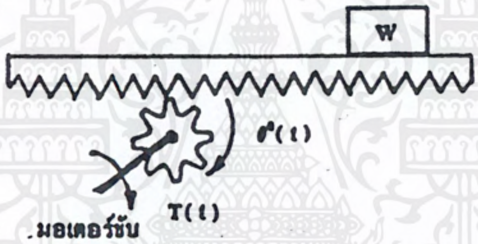
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.4 ความสัมพันธ์ระหว่างการเคลื่อนที่แบบเป็นเชิงเส้นและแบบเป็นเชิงมุม

ปัญหาเกี่ยวกับการบังคับการเคลื่อนที่ซึ่งประสบกันบ่อยๆ คือ การแปลงการเคลื่อนที่แบบเชิงมุมให้ไปเป็นการเคลื่อนที่แบบเชิงเส้น ตัวอย่างการแก้ปัญหาดังกล่าวแสดงได้ในรูปที่ 2.5และ 2.6 ซึ่งเป็นระบบง่ายๆ ที่มีแรงเฉื่อยสมมูลย์ต่ออยู่โดยตรงกับตัวมอเตอร์

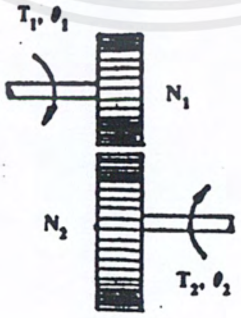



รูปที่ 2.5 ระบบมอเตอร์และลีดสกรู




รูปที่ 2.6 ระบบซี่เฟืองตรงและเฟืองจักร

### 2.2.5 ระบบเกียร์




รูปที่ 2.7 ระบบการคัปปลิ้งของเกียร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเกียร์ คานงัดหรือสายพานในลูกกรอกเป็นอุปกรณ์เครื่องกลซึ่งสามารถส่งพลังงานจากส่วนหนึ่งของระบบไปยังส่วนอื่นๆ ได้ในรูปที่ของแรงงาน แรงบิด ความเร็ว และการเคลื่อนที่ นอกจากนั้นอุปกรณ์เหล่านี้ยังเป็นเสมือนอุปกรณ์สำหรับประสาน (Matching) ที่สามารถใช้ส่งผ่านกำลังงานให้ได้ค่าสูงสุด รูปที่ 2.7 แสดงถึงการคัปปลิ้ง (Cuppling) เกียร์ 2 ตัว เข้าด้วยกัน แรงเฉื่อยและแรงเสียดทานของเกียร์จะไม่มานำมาคิดในเมื่อพิจารณาถึงเกียร์ในอุดมคติ

ความสัมพันธ์ระหว่างแรงบิด $T_1$ และ $T_2$ การเลื่อนที่เชิงมุม $\theta_1$ และ $\theta_2$ และจำนวนซี่ฟัน $N_1$ และ $N_2$ ของระบบเกียร์สามารถหาได้จากหลักเกณฑ์ต่อไปนี้

(1) จำนวนซี่ฟันของเกียร์จะเป็นสัดส่วนกับรัศมี r และ r ของเกียร์นั่น คือ

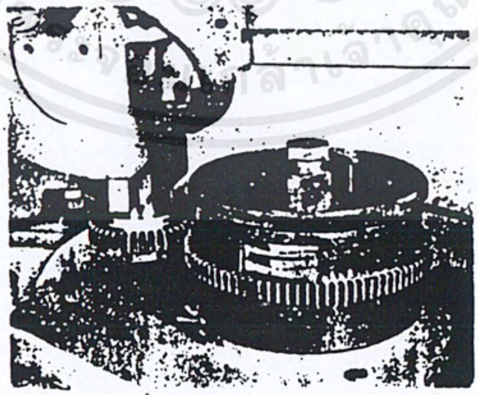
$$r_1 N_2 = r_2 N_1$$

(2) ระยะทางการเคลื่อนที่ไปของเกียร์แต่ละตัวจะมีค่าเท่ากันดังนั้น

$$\theta_1 r_1 = \theta_2 r_2$$

(3) แรงงานที่ได้จากเกียร์ตัวหนึ่งจะเท่ากับแรงงานที่ได้จากเกียร์อีกตัวหนึ่งเนื่องจากสมมติให้ว่าไม่มีการสูญเสียแรงงานดังนั้น

$$_1\theta_1 = T_2\theta_2$$

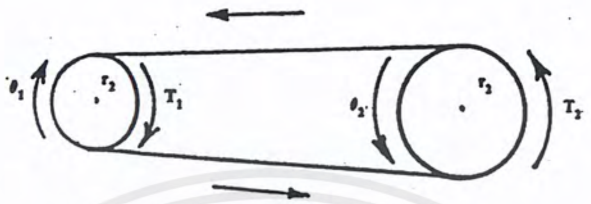


**รูปที่ 2.8** ระบบเกียร์ที่สามารถเพิ่มแรงบิดได้หลายเท่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.6 ระบบสายพานและคานงัด

ระบบการขับด้วยสายพานหรือโซ่ก็มีจุดมุ่งหมายเดียวกับระบบเกียร์ นอกจากว่าระบบสายพานหรือโซ่สามารถส่งผ่านพลังงานไปได้ระยะทางไกลกว่าโดยที่ไม่ต้องใช้เกียร์จำนวนมาก



**รูปที่ 2.9** ระบบกว้านและสายพาน

เราก็ยังสามารถใช้สมการกับกรณีนี้ได้ซึ่งเป็นความจริงว่าการสะท้อนหรือทรานสมิทแต้นซ์ (Transmittance) ของแรงบิด แรงเฉื่อยและแรงเสียดทานเป็นต้นก็คล้ายคลึงกับการสะท้อนกลับของระบบเกียร์

## 2.3 เซ็นเซอร์และเอนโคคเดอร์ของเพลามอเตอร์ (Sensors and Shaft Encoders)

เซ็นเซอร์และเอนโคคเดอร์เป็นส่วนประกอบที่สำคัญในระบบควบคุมที่มีการป้อนกลับในระบบควบคุมแบบวงรอบเปิด เอนโคคเดอร์มักจะใช้สำหรับตรวจสอบคุณสมบัติการทำงานของระบบ ส่วนในระบบควบคุมแบบวงรอบปิด เซ็นเซอร์และเอนโคคเดอร์ใช้เป็นตัวป้อนสัญญาณกลับเพื่อการควบคุม นอกจากนั้นส่วนประกอบเหล่านี้ยังใช้สำหรับพิสูจน์ความเหมือนกันของกระบวนการที่ไม่รู้ค่าหรือกระบวนการที่มีการเปลี่ยนแปลงในบทนี้จะได้ก่าวถึงหลักการทำงานและการนำไปใช้งานของเซ็นเซอร์และเอนโคคเดอร์ที่ใช้กันมากในระบบควบคุม เช่น โพเทนทิโอมิเตอร์, ออฟติกคอลเอนโคดเดอร์, แมกนิติคพิคอัพทรานสดิวเซอร์จำพวกแม่เหล็กไฟฟ้าและรีโซลเวอร์

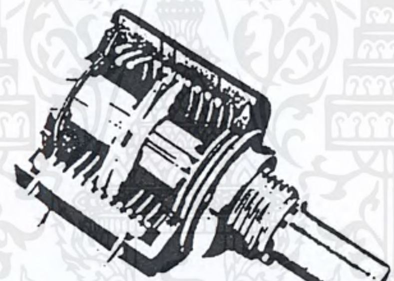
### 2.3.1โพเทนทิโอมิเตอร์

โพเทนทิโอมิเตอร์เป็นเครื่องมือที่เปลี่ยนพลังงานกลไปเป็นพลังงานไฟฟ้าอินพุทที่ป้อนเข้าโพเทนทิโอมิเตอร์จะอยู่ในรูปของการเคลื่อนที่แบบเชิงกล และอาจเป็นการเคลื่อนที่แบบหมุนรอบหรือเคลื่อนที่เป็นเชิงเส้นก็ได้ เมื่อป้อนโวลท์เต็จเข้าที่ปลายข้างที่อยู่กับที่ของโพเทนทิโอมิเตอร์เอาท์พุทโวลท์เต็จจะได้ออกมาที่ปลายของส่วนที่เคลื่อนที่ขอโพเทนทิโอมิเตอร์ซึ่งจะแปรผันโดย

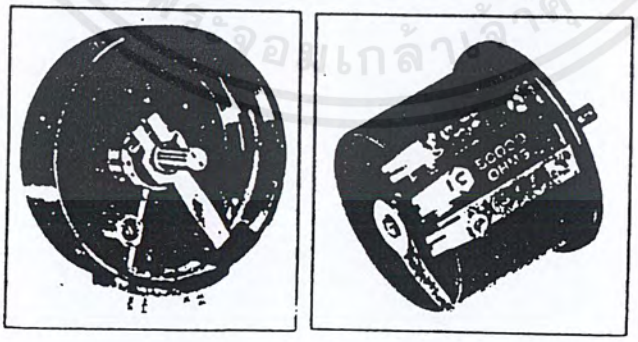
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตรงกับความเร็วของการหมุนหรือระยะของการเคลื่อนที่เป็นเชิงเส้นโพเทนทิโอมิเตอร์แบบหมุนรอบได้นั้นมีแบบหมุนได้รอบเดียวหรือหมุนได้หลายรอบ และส่วนใหญ่จะประกอบขึ้นด้วยขอลวดหรือความต้านทานของพลาสติกที่เป็นตัวนำ ในรูปที่ 2.10 แสดงถึงรูปของโพเทนทิโอมิเตอร์ชนิดหมุนได้ 10 รอบ และรูปที่ 2.11 แสดงถึงโครงสร้างภายในของโพเทนทิโอมิเตอร์แบบหมุนได้รอบเดียวโพเทนทิโอมิเตอร์แบบพลาสติกตัวนำได้รับความนิยมใช้มากในระบบควบคุมที่ต้องการความเที่ยงตรงเนื่องจากโพเทนทิโอมิเตอร์ดังกล่าวมีความละเอียดไม่จำกัด มีอายุการใช้งานได้ยาวนาน ให้เอาท์พุทโวลท์เต็จที่ราบเรียบ และมีสแตติค (Static) น้อยส์ (Noise) ต่ำ

ตัวอย่างความคลาดเคลื่อนของความต้านทานของโพเทนทิโอมิเตอร์ชนิดมาตรฐานจะอยู่ระหว่าง 5 เปอร์เซ็นต์ และ 10 เปอร์เซ็นต์และโพเทนทิโอมิเตอร์ชนิดพิเศษจะมีความคลาดเคลื่อนประมาณ 1 เปอร์เซ็นต์ ความเป็นลิเนียร์ของความต้านทานของโพเทนทิโอมิเตอร์จะอยู่ในพิสัย 0.25 ถึง 0.5 เปอร์เซ็นต์



**รูปที่ 2.10** โรตารีโพเทนทิโอมิเตอร์ชนิดหมุนได้ 10 รอบ



**รูปที่ 2.11** โครงสร้างภายในของโพเทนทิโอมิเตอร์แบบเพลาหมุนได้รอบเดียว

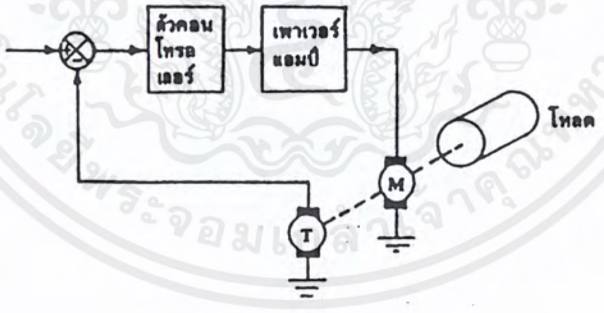
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.3.2 ทาโคมิเตอร์

ทาโคมิเตอร์เป็นเครื่องมือที่สามารถแปลงพลังงานกลไปเป็นพลังงานไฟฟ้าและให้กำเนิดเอาท์พุทโวลท์เต็จที่เป็นสัดส่วนกับแมกนิจูดของความเร็วเชิงมุม ในระบบการบังคับมอเตอร์โดยทั่วไปจะใช้ทาโคมิเตอร์เป็นตัวแสดงความเร็วเพื่ออ่านค่าความเร็วของเพลาของมอเตอร์หรือเพื่อป้อนความเร็วกลับสำหรับการบังคับความเร็วหรือสำหรับทำให้เสถียรภาพของระบบดีขึ้น

ลักษณะการทำงานของดีซีทาโคมิเตอร์กับดีซีมอเตอร์จะต้องตรงกันข้าม ส่วนดีซีทาโคมิเตอร์ (D.C. Tacho Meter) กับดีซีเยนเนอเรเตอร์ (D.C. Generator) จะเหมือนกันเพียงแต่ต่างกันที่การใช้งาน กล่าวคือ ดีซีทาโคมิเตอร์มีขอบเขตการใช้งานเพียงเป็นตัววัดความเร็วของเพลาแล้วส่งผลออกมาเป็นสัญญาที่อ่านค่าได้ง่ายเท่านั้น แต่ถ้านำทาโคมิเตอร์ไปใช้ในโรงงานอุตสาหกรรมซึ่งมีสัญญารบกวนมาก และบางครั้งต้องส่งผลที่วัดได้ไปเป็นระยะทางไกลพอประมาณ ทาโคมิเตอร์จึงต้องมีเอาท์พุทอิมพีแด้นซ์ต่ำและผลิตกำลังงานออกมาสูงพอประมาณ ในกรณีนี้เราอาจจะเรียกทาโคมิเตอร์นี้ว่าเป็นดีซีเยนเนอเรเตอร์

ดีซีโคมิเตอร์สามารถใช้เป็นตัวป้อนความเร็วของเพลาของมอเตอร์กลับเพื่อเปรียบเทียบกับสัญญาณเปรียบเทียบแล้วขยายผลต่างที่ได้ให้ไปขับมอเตอร์ในระบบการบังคับความเร็วของมอเตอร์โดยการต่อทาโคมิเตอร์เข้ากับเพลาของมอเตอร์โดยตรงดังแสดงในรูปที่ 2.12




**รูปที่ 2.12** บล็อคไดอะแกรมของระบบการบังคับความเร็ว

ส่วนการใช้งานอย่างที่สามของทาโคมอเตอร์ คือ ใช้เป็นตัววัดความเร็วของเพลา ของมอเตอร์แล้วส่งผลออกมาให้เห็นโดยการต่อเอาท์พุทโวลท์เต็จของทาโคมิเตอร์เข้ากับโวลท์มิเตอร์แล้วอ่านค่าความเร็วการหมุนของมอเตอร์ในรูปของโวลท์เต็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ชนิดของดีซีทาโคมิเตอร์**

ดีซีทาโคมิเตอร์กับดีซีมอเตอร์มีโครงสร้างเหมือนกันแต่การทำงานตรงกันข้ามซึ่งมีทั้งชนิดมีแปรงถ่านและชนิดไม่มีแปรงถ่าน สำหรับชนิดมีแปรงถ่านยังแบ่งออกได้เป็นชนิดแกนเหล็ก (Iron Core) และชนิดขดลวดหมุน (Moving Coil)

**โมเดลคณิตศาสตร์ของทาโคมิเตอร์**

คุณลักษณะพื้นฐานของทาโคมิเตอร์ คือ จะให้เอาท์พุทโวลท์เต็จเป็นสัดส่วนกับความเร็วของโรเตอร์ ดังนั้นคุณลักษณะไดนามิคของทาโคมิเตอร์สามารถแสดงได้โดยสมการ

$$e_g(t) = K_g \frac{d\theta(t)}{dt} = Kg\,\omega(t)$$

เมื่อ $e_g(t)$ = เอาท์พุทโวลท์เต็จ (โวลท์)

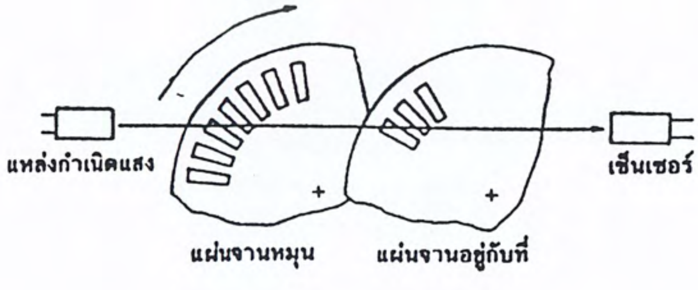
$\theta(t)$ = ระยะการเคลื่อนที่ของโรเตอร์ (เรเดียน)

$\omega(t)$ = ความเร็วของโรเตอร์ (เรเดียน /วินาที)

$K_g$ = ค่าคงที่ของทาโคมิเตอร์ (โวลท์/รอบต่อนาที x $10^3$)

## 2.3.3 อินคริเมนท์เอนโคดเดอร์

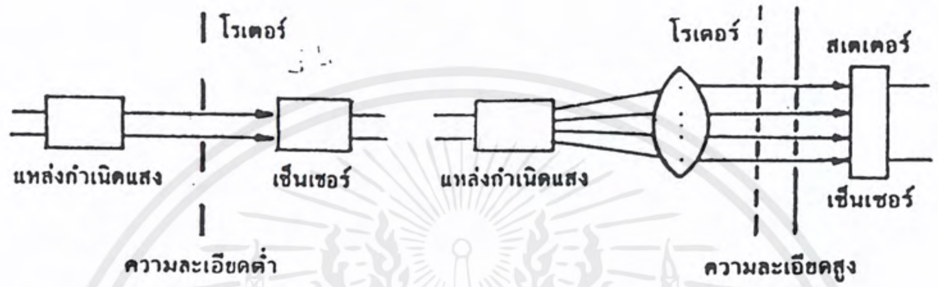
ในระบบการบังคับตำแหน่งหรือความเร็วของมอเตอร์ ต้องใช้อินคริเม้นท์เอนโคดเดอร์หรือเอนโคดเดอร์สำหรับรักษาตำแหน่งและสำหรับสร้างสัญญาณป้อนกลับ โดยที่ตัวเอนโคดเดอร์จะสร้างสัญญาพัลซ์ที่แปรผันตรงกับการหมุนของเพลา ซึ่งสามารถนำไปใช้ในการรับรู้ความเร็วของเพลามอเตอร์ในรูปของอัตราจำนวนพัลซ์ได้




**รูปที่ 2.13** ตัวอย่างกลไกของออปโตอินคริเมนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินคริเม้นท์เอนโคดเดอร์ประกอบด้วยส่วนที่สำคัญ คือ ตัวกำเนิดแสง จานหมุน (Rotary Disk) จานอยู่กับที่และตัวเซนเซอร์ บนแผ่นจานหมุนทำเป็นช่องโดยรอบดังแสดงในรูปที่ 2.13 และบนแผ่นอยู่กับที่จะมีช่องสำหรับให้แสดงผ่านตรงข้ามไปยังตัวเซนเซอร์ ถ้าเป็นเอนโคดเดอร์ที่ใช้วัดความเร็วต่ำไม่ต้องมีแผ่นอยู่กับที่ก็ได้ ส่วนตัวกำเนิดแสงอาจจะเป็นหลอดไฟหรือ LED ก็ได้ ดังแสดงในรูปที่ 2.14




**รูปที่ 2.14** ตัวเซ็นเซอร์แบบมีช่องปิดเปิดให้แสงผ่านได้ช่องเดียวและแบบมีหลายช่อง

**ความละเอียดของอินคริเม้นท์เอนโคดเดอร์**

ความละเอียดของเอนโคดเดอร์ คือ จำนวนคาบเวลาของสัญญาณเอาท์พุทต่อการหมุนของเพลา 1 รอบ ซึ่งบวกเป็นจำนวนพัลซ์ต่อรอบหรือจำนวนไซเคิลต่อ 360 องศา มุมทางเชิงกลหรือไซเคิลต่อองศา เอนโคดเดอร์ที่ใช้กันทั่ว ๆ ไปมีค่าความละเอียดตั้งแต่ 15 ถึง 10,000 พัลซ์ต่อรอบ

ในทางปฏิบัติเนื่องจากแสงที่ออกจากแหล่งกำเนิดเป็นลำแสงเดียว ถ้าเราต้องการให้แสงที่ผ่านช่องไปยังเซนเซอร์เป็นเส้นตรงพร้อม ๆ กัน (Collimation) ก็ทำได้โดยใช้เลนซ์หรือ พาราโบลิครีแฟลคเตอร์ (Parabolic Reflagter)

จำนวนพัลซ์ต่อ 1 รอบของสัญญาณที่เอนโคดเดอร์สร้างออกมาจะเท่ากับจำนวนช่องว่างบนแผ่นจานหมุน และความกว้างของช่องว่างกับความกว้างของแถบทึบระหว่างช่องว่างจะเท่ากัน เพราะฉะนั้นเราสามารถคำนวนความกว้างของช่องว่าง ($\omega$) ได้จาก

$$\omega = \frac{\pi D}{2N}$$

เมื่อ D = เส้นผ่าศูนย์กลางเฉลี่ยของแผ่นจาน

N = จำนวนของความละเอียดเป็นพัลซ์ต่อรอบ

$\omega$ = ความกว้างของช่องว่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เอาท์พุทของเอนโคดเดอร์**

โดยทั่วๆ ไปแล้วสัญญาณเอาท์พุทที่ออกจากเอนโคดเดอร์โดยตรงจะมีระดับไม่เพียงพอในการควบคุมหรือสำหรับการประมวลสัญญาณ ดังนั้น จึงต้องมีวงจรขยายและแปลงรูปร่างลูกคลื่นสัญญาณต่อไว้ในตัวเอนโคดเดอร์ด้วยเสมอ สัญญาณลูกคลื่นที่ได้จากตัวเซนเซอร์ปกติแล้ว จะเป็นรูปสัญญาณสามเหลี่ยมหรือรูปสัญญาณซายน์ ขึ้นอยู่กับความละเอียดที่ต้องการ รูปสัญญาณเหล่านี้สามารถทำให้เป็นสัญญาณรูปสี่เหลี่ยมได้โดยการต่อตัวคอมพาเรเตอร์ เข้ากับลิเนียร์แอมปลิไฟของเอนโคดเดอร์ก็จะได้เอาท์พุทเป็นลูกคลื่นสี่เหลี่ยมตามต้องการ

**รูปที่ 2.15** โรตารี่เอนโคดเดอร์ที่มีเอาท์พุทเป็นไบนารี่
(a) ลักษณะตัวอย่างแบบหนึ่ง
(b) แผ่นจากภายในที่ประกอบด้วยข้อมูลไบนารี่

ในอินครีเม้นท์เอนโคดเดอร์บางชนิดจะมีพัลซ์ที่แสดงถึงจำนวนรอบของการหมุนสำหรับใช้เป็นศูนย์ในการอ้างอิงพัลซ์ที่ใช้แสดงจำนวนรอบนี้จะเกิดขึ้นท 1 พัลซ์ต่อ 1 รอบ โดยทั่วไปใช้บอกถึงตำแหน่งเชิงกลหรือใช้เป็นสัญญาณเคลียร์จำนวนที่นับไว้ในหน่วยเก็บข้อมูล

**รูปที่ 2.16** ตัวอย่างส่วนประกอบของเอนโคดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.4 ระบบควบคุมความเร็ว

## (Velocity Control System)

ระบบการควบคุมความเร็วรอบ หรือความเร็ว คือ ระบบที่สามารถกำหนดความเร็วให้คงที่หรือแปรค่าได้ ตามที่เราต้องการด้วยการตั้งค่าสัญญาณ อินพุทของระบบ การแปรค่าไปของความเร็วจะต้องอยู่ภายในย่านของระบบ ระบบการควบคุมความเร็วมีทั้งแบบการควบคุมแบบวงรอบเปิด ซึ่งเป็นระบบการควบคุมอย่างง่ายๆ และการควบคุมแบบวงรอบปิดเป็นการควบคุมที่ให้ความเที่ยงตรงของความเร็วสูง

ในหน่วยนี้จะได้อธิบายเชิงเปรียบเทียบถึงคุณสมบัติการควบคุมความเร็วแบบเปิดลูปของมอเตอร์ด้วยตัวควบคุมเลอร์แบบต่างๆ จากนั้นก็จะได้อธิบายถึงการควบคุมความเร็วแบบปิดลูปซึ่งยังสามารถแบ่งออกได้เป็น 2 แบบคือ การควบคุมความเร็วแบบเร็คกูเลเตอร์ คือ การควบคุมความเร็วได้เพียงทิศทางเดียว และการควบคุมความเร็วแบบเซอร์โว คือ การควบคุมความเร็วได้สองทิศทาง ส่วนการวิเคราะห์ให้เห็นความแตกต่างระหว่างระบบการควบคุมความเร็ว จะอธิบายด้วยบล็อคไดอะแกรม โดยการแยกวิเคราะห์ให้เห็นความแตกต่างระหว่างระบบการควบคุมความเร็วที่ให้ขยายแบบขยายกระแสและขยายแรงดัน การวิเคราะห์จากฟังชั่นถ่ายโอนสองแบบ คือ แบบแรกเป็น ฟังชั่นถ่ายโอนระหว่างความเร็วเอาท์พุทและอินพุท แบบที่สองเป็นฟังก์ชั่นถ่ายโอนระหว่างความเร็วเอาท์พุทและภาระ นอกจากนั้นก็จะเป็นตัวอย่างการออกแบบระบบควบคุมความเร็วของอินดัคชั่นมอเตอร์แบบสองเฟส

### 2.4.1 ระบบการควบคุมความเร็วของมอเตอร์แบบวงรอบเปิด

ตัวคอนโทรลเลอร์ที่ใช้ควบคุมความเร็วของมอเตอร์แบบเปิดลูปแบ่งออกได้เป็น 4 แบบ ดังต่อไปนี้

**ตัวคอนโทรลเลอร์แบบความต้านทานที่ปรับค่าได้**

การควบคุมแบบนี้เป็นวิธีการควบคุมที่ง่ายที่สุดและใช้กันในยุคแรกๆ มีประสิทธิภาพการควบคุมความเร็วอยู่ในพิสัย 4 :1 และให้การเร็คกูเลทความเร็วของมอเตอร์ไม่ดีต่อการเปลี่ยนแปลงโหลดและโวลท์เตจของสายส่ง การควบคุมแบบนี้ไม่มีประสิทธิภาพเพราะว่ากำลังไฟสูญเสียไปในตัวความต้านทาน

การควบคุมแบบนี้ให้คุณสมบัติการสตาร์ทดี (ให้แรงบิดสูงที่ความเร็วต่ำ) แต่จะให้ความเร็วสูงจัดมากเมื่อมอเตอร์อยู่ในภาวะที่ภาระน้อยๆ ดังนั้นการควบคุมแบบนี้จะเป็นประโยชน์เฉพาะภาวะที่แรงต้าน (Friction) คงที่ยกตัวอย่างเช่น การควบคุมความเร็วของเครื่องจักรเย็บผ้าเครื่องผสม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อาหาร และการบังคับแบบนี้จะให้การเร็คกูเลทความเร็วเลวลงเมื่อความเร็วลดลงทำให้เป็นการยากที่จะควบคุมความเร็วให้ได้พิสัยกว้าง

ส่วนการควบคุมความเร็วของมอเตอร์ขนานด้วยตัวคอนโทรลเลอร์แบบรีโอสตาร์ท ตัวคอนโทรลเลอร์แบบอนุกรมอยู่กับวงจรอาร์มาเจอร์และขดลวดของฟิลด์ถูกกระตุ้นด้วยแรงดันคงที่ (ถ้าเป็นมอเตอร์แบบฟิลด์ที่ได้จากแม่เหล็กถาวรก็ใช้ได้)

ผลของการควบคุมแสดงในรูปซึ่งแสดงถึงความสัมพันธ์ระหว่างความเร็วกับแรงบิดการควบคุมแบบนี้ถ้าเพิ่มความต้านทานในวงจรอาร์มาเจอร์มากขึ้น ก็ทำให้การเร็คกูเลทความเร็วลดลง ด้วยการควบคุมแบบนี้จึงเหมาะสมสำหรับงานที่แรงบิดของโหลดคงที่ดีกว่างานที่มีภาวะของโหลดเปลี่ยนแปลงในช่วงกว้าง

**ตัวคอนโทรลเลอร์แบบหม้อแปลงที่ปรับค่าได้ (Variac) กับตัวเรียงกระแส**

การควบคุมแบบนี้สามารถขับดีซีมอเตอร์ได้พิสัยความเร็ว 10 : 1 และการให้การเร็คกูเลทและประสิทธิภาพที่ดีกว่าการควบคุมด้วยรีโอสตาร์ท

การควบคุมโดยการปรับค่าบังคับแรงดันของมอเตอร์ขนานด้วยการปล่อยกระแสไฟฟ้าให้มีสนามแม่เหล็กไฟฟ้าคงที่ ผลของคุณสมบัติความเร็ว แรงบิด ได้รับการปรับปรุงดีขึ้นกว่าการควบคุมด้วยความต้านทานที่ปรับค่า และให้การเรียงกระแสความเร็วคงที่ได้ดีขึ้นตลอดพิสัยความเร็วที่กว้างกว่า

**ตัวคอนโทรลเลอร์แบบ SCR (ไทริสเตอร์)**

ด้วยการทำงานเป็นครึ่งคลื่นจะให้คุณสมบัติคล้ายกับการควบคุมด้วยการปรับค่า อย่างไรก็ตามระบบของ SCR ที่ทำงานแบบเต็มคลื่นสามารถให้พิสัยการควบคุมความเร็วได้ถึง 20 : 1 เมื่อใช้เทคนิคการชดเชย “IR” (หมายถึง เทคนิคการคับรู้กระแสและป้อนกลับเป็นระบบวงรอบปิด) ด้วยวิธีการชดเชยการควบคุมความเร็วนี้สามารถให้การเร็คกูเลทความเร็วได้ถึง 3% จากไม่มีภาระถึงภาวะที่มีภาระเต็มที่ เช่น มอเตอร์มีความเร็วที่กำหนดให้ 1800 รอบต่อนาที การควบคุมให้เร็คกูเลทความเร็วเท่ากับ 3% ของความเร็วที่ตั้งไว้ (Set Speed)

**ตัวคอนโทรลเลอร์แบบควบคุมแรงดันอาร์มาเจอร์ของมอเตอร์ด้วยเยนเนอเรเตอร์**

เป็นวิธีการควบคุมความเร็วแบบวงรอบเปิดที่ค่อนข้างยาก มอเตอร์ที่มีความเร็วคงที่จะใช้เป็นตัวขับคลื่นเยนเนอเรเตอร์ที่มีแรงดันของฟิลด์ถูกควบคุมให้แปรค่าได้ ดังนั้นเยนเนอเรเตอร์จะเป็นผลิตโวลท์เตจที่ปรับค่าได้จ่ายให้กับอาร์มาเจอร์ของมอเตอร์และผลที่ได้ของคุณสมบัติ แรงบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วจะดีกว่า เนื่องจากการควบคุมแบบนี้ให้การเร็คกูเลทที่เป็นอิสระต่อความเร็วที่ได้ตั้งไว้

ผลลัพธ์ที่ได้ของการควบคุมแบบนี้ให้คุณสมบัติที่เหนือกว่าการควบคุมความเร็วของมอเตอร์วิธีต่างๆ ที่ได้กล่าวมาแล้ว อย่างไรก็ตามเนื่องจากราคาแพงของชุดมอเตอร์เยนเนอเรเตอร์และส่วนการควบคุมฟิลด์ของเยนเนอเรเตอร์ทำให้การควบคุมความเร็ววิธีนี้ใช้ไม่ได้ผลดีกับควบคุมความเร็วของมอเตอร์ตัวเล็กๆ การประยุกต์ใช้งานส่วนใหญ่จะใช้ในงานอุตสาหกรรมที่ต้องการควบคุมความเร็วของมอเตอร์ตัวใหญ่ๆ ตั้งแต่ 1 แรงม้าขึ้นไป เพราะว่าคุณสมบัติการขยายกำลังไฟฟ้าของชุดมอเตอร์เยนเนอเรเตอร์ เหมาะสมใช้ในระบบการควบคุมวงรอบปิดในยุคแรกๆ เมื่อหลอดสูญญากาศกำลังสูงและเครื่องขยายแบบแม่เหล็กเริ่มนำมาใช้สำหรับการควบคุม

วิธีการควบคุมดังที่กล่าวมาแล้วข้างต้นใช่ว่าจะได้รับการพัฒนาเพื่อใช้งานทั้งหมดทุกวิธี วิธีการควบคุมเหล่านั้นถือได้ว่าเป็นตัวอย่างของปัญหาที่จะต้องได้พบ ในการควบคุมแบบวงรอบเปิดของระบบการทำงานที่ควบคุมด้วยมือในอดีตเราสามารถเห็นได้ว่า ในแต่ละกรณีก็มีข้อเสียที่ไม่ต้องการเกิดขึ้น ซึ่งเป็นข้อจำกัดให้แต่ละวิธีใช้งานได้เฉพาะอย่างเท่านั้น

### 2.4.2 การควบคุมความเร็วของมอเตอร์แบบวงรอบปิด

ระบบการควบคุมความเร็วของดีซีมอเตอร์แบบวงรอบปิด แบ่งออกได้เป็น 2 แบบคือ

1. การควบคุมความเร็วแบบเร็คกูเลเตอร์
2. การควบคุมความเร็วแบบเซอร์โว

**ระบบการควบคุมความเร็วแบบเร็คกูเลเตอร์**

จะเป็นการบังคับความเร็วเพียงทิศทางเดียวโดยทำงานในควอดแดรนท์แรกเท่านั้น ระบบไม่สามารถจะให้แรงบิดของดีซีมอเตอร์ที่มีค่าเป็นลบ และไม่สามารถกลับทิศทางของความเร็วได้

ระบบการควบคุมความเร็วแบบทิศทางเดียวนี้เราสามารถจะเพิ่มไดนามิคเบรคเข้าไปในระบบได้ทำให้เราสามารถจำกัดแรงบิดลบของมอเตอร์ในควอดแดรนท์ที่สองได้ แต่เนื่องจากการบังคับในควอดแดรนท์ที่สองนี้อยู่ในพื้นที่การควบคุมให้มอเตอร์หยุดหมุน ดังนั้นเราถือได้ว่า การควบคุมความเร็วแบบเร็คกูเลเตอร์เป็นการควบคุมแบบควอดแดรนท์เดียว

**ระบบการควบคุมความเร็วแบบเซอร์โว**

จะเป็นการบังคับความเร็วได้สองทิศทาง นอกจากนั้นระบบแบบนี้ยังสามารถให้ความเร็วและแรงบิดของมอเตอร์ได้ทั้งในทิศทางลบและบวก ซึ่งแสดงถึงพื้นที่การควบคุมแบบเซอร์โวคือสามารถทำงานได้แบบสองทิศทางหรือเป็นการควบคุมด้ 4 ควอดแดรนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.3 ระบบการควบคุมความเร็วได้เพียงทิศทางเดียวหรือควอดแดรนท์เดียว

ระบบการควบคุมแบบควบคุมความเร็วได้เพียงควอดแดรนท์เดียว จะใช้เพาเวอร์แอมปลิไฟเออร์แบบขยายสัญญาณได้ทิศทางเดียว สามารถจ่ายกำลังงานให้มอเตอร์เฉพาะกระแสและแรงดันที่เป็นบวก (หรือลบเท่านั้น) ดังนั้นการควบคุมความเร็วของมอเตอร์

เพาเวอร์แอมป์ที่ใช้ในระบบควบคุม ที่มีการทำงานในคว็อดแดร้นท์เดียว มีคุณลักษณะ คือ เอาท์พุทอิมพีแด้นซ์ของมันจะต่ำตราบเท่าที่ $V > E_g$ และกระแส $I_a$ มีค่าเป็นบวกเมื่อภาวะดังกล่าวแปรไปเอาท์พุทอิมพีแด้นซ์ก็จะกลับมีค่าสูงมากและไม่ยอมให้กระแสลบไหลผ่าน

เราสามารถปรับปรุงเพาเวอร์แอมป์ให้ดีขึ้นได้ โดยใช้วงจรที่สามารถรับรู้กระแสตราบเท่าที่กระแสเป็นบวกวงจรนี้จะไม่ทำงาน แต่ถ้ากระแส $I_a$ เป็นศูนย์ วงจรจะลัดระหว่างขั้วของมอเตอร์ ซึ่งทางผ่านให้กระแสลงไหลวนอยู่ในอาร์มาเจอร์ ดังนั้นจะเป็นการหยุดมอเตอร์ วิธีนี้เรียกว่า ไดนามิคเบรค

วงจรไดนามิคเบรคจะเป็นตัวทำให้เกิดการลัดวงจรคร่อมมอเตอร์ เมื่อระบบการควบคุมไม่ยอมปล่อยให้กระแสไหลผ่านมอเตอร์ในภาวะซึ่งอยู่ในช่วงเวลาที่อยู่นอกเหนือการควบคุม ไดนามิคเบรคจะขึ้นอยู่กับแรงดันย้อนกลับ (Counter EMF) ของอาร์มาเจอร์ที่จะทำให้เกิดกระแสเบรคแรงบิดห้ามการหมุนของเพลาจะมีค่าสูงสุดที่ความเร็วสูงและมีประสิทธิภาพต่ำมีความเร็วต่ำ

**ระบบการควบคุมความเร็วแบบทิศทางเดียวด้วยทรานซิสเตอร์เพียงตัวเดียว**

ระบบควบคุมความเร็วของดีซีมอเตอร์อย่างง่าย ๆ ที่ประกอบด้วยทรานซิสเตอร์เพียงตัวเดียวและใช้กันอย่างกว้างขวางในกรณี เมื่อโหลดคงที่และความเร็วถูกจำกัดอยู่ในช่วงหนึ่งข้อดี ก็คือ เป็นระบบที่ง่ายมาก ๆ และราคาถูก

ระบบประกอบด้วย โปเทนทิโอมิเตอร์ขนาด 5 วัตต์ (W) และทรานซิสเตอร์กำลังติดตั้งบนแผ่นระบายความร้อน

**ขีดจำกัดแรงบิดของมอเตอร์**

การป้องกันไม่ให้เกิดภาระเกินเกิดขึ้นภายในระบบเราจะต้องมีตัวรับรู้กระแสซึ่งจะทำให้วงจรขยายหยุดทำงานที่ระดับกระแสที่ต้องการเนื่องจากแรงบิดของมอเตอร์เป็นสัดส่วนกับกระแสในดีซีมอเตอร์แบบแม่เหล็กถาวร ตัวจำกัดกระแสจะเป็นผลถึงการจำกัดแรงบิดนั่นเอง คุณสมบัติระหว่างแรงบิดและความเร็วของมอเตอร์ แสดงให้เห็นถึงความแตกต่างของขอบเขตการควบคุมสองของเขต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบางกรณีมีความจำเป็นอย่างยิ่งที่จะต้องให้การเพิ่มแรงบิดสูงกว่าขีดจำกัดแรงบิดเป็นเวลาบ่อยครั้งจึงจำเป็นต้องป้องกันระบบไม่ให้เกิดภาระเกิน ระบบการป้องกันภาระเกินที่กระทำได้โดยกำหนดหน่วงเวลา ที่เป็นฟังก์ชั่นตัวจำกัดแรงบิดของมอเตอร์ ค่าหน่วงเวลาโดยทั่วไปมีค่าประมาณ 150 มิลลิวินาที สำหรับมอเตอร์ที่มีอาร์มาเจอร์เป็นแกนเหล็ก และมีค่าลดลง สำหรับมอเตอร์ที่มีความเฉื่อยต่ำ ในช่วงเวลาหน่วงเวลาดังกล่าว มอเตอร์จะเพิ่มอัตราเร่งอย่างรวดเร็วด้วยกระแสที่เพิ่มขึ้นอย่างไม่จำกัด และต่อจากนั้นกระแสก็จะถูกจำกัดหลังจากมอเตอร์ได้เพิ่มอัตราเร่งสุดสิ้นไปแล้ว ผู้ที่ใช้ระบบการควบคุมความเร็วอาจจะพบว่าการจำกัดกระแสมีประโยชน์ เมื่อต้องการเพิ่มอัตราเร่งให้กับภาระที่มีความเฉื่อยสูง วงจรจำกัดกระแสในกรณีจะควบคุมการเพิ่มอัตราเร่งจนถึงความเร็วที่ต้องการ โดยที่วงจรควบคุมหรือมอเตอร์ไม่ได้รับอันตราย

**การควบคุมความเร็วที่สามารถกลับทิศทางได้**

วงจรกลับทิศทางของมอเตอร์ อาจทำงานได้ด้วยรีเลย์แทนที่จะใช้สวิทซ์ที่ทำงานด้วยมือทำให้ได้ระบบการควบคุมความเร็วที่มีคุณสมบัติแบบควบคุมได้สองทิศทางสำหรับการบังคับในระยะทางไกล อย่างไรก็ตามวิธีการควบคุมแบบนี้ตามปกติทั่วไปใช้งานไม่ได้ผลดีในระบบการควบคุมตำแหน่งแบบปิดลูฟหรือในระบบเซอร์โวควบคุมความเร็ว ระบบเซอร์โวตามปกติจะมีการเปลี่ยนกลับทิศทางอยู่ตลอดเวลารอบๆ จุดสมดุลย์ ดังนั้นหน้าสัมผัสของรีเลย์จะสึกกร่อนอย่างรวดเร็วนอกจากนั้นแล้วระบบดังกล่าวยังขึ้นอยู่กับไทม์ดีเลย์และความไม่เป็นเชิงเส้นเนื่องจากการสวิทซ์ของรีเลย์

## 2.4.4 การวิเคราะห์และออกแบบระบบควบคุมความเร็ว

ระบบควบคุมความเร็วโดยทั่วไป จะต้องออกแบบระบบให้มีความสอดคล้องกับภาวะทั้ง 2 สภาวะต่อไปนี้

**ความเร็วของผลตอบสนอง**

ระบบการควบคุมความเร็วโดยทั่วไป ควรจะสามารถตอบสนองตามสัญญาณคำสั่งควบคุมได้อย่างรวดเร็ว ข้อกำหนดนี้อาจแสดงได้ทั้งในเทอมของผลตอบสนองต่อสเตปของระบบหรือในเทอมแบนด์วิดท์ของระบบ

**การเปลี่ยนแปลงของความเร็ว**

การเปลี่ยนแปลงของความเร็วเนื่องจากการเปลี่ยนแปลงของโหลดและเนื่องจากสิ่งรบกวนอื่น ๆ ต้องมีค่าน้อยที่สุดเท่าที่จะทำได้ นอกจากนั้นคุณลักษณะชั่วขณะที่เกิดขึ้นต่อการเปลี่ยนแปลงของโหลดความจะลดลงเร็วที่สุดเท่าที่จะทำได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การวิเคราะห์ระบบการควบคุมความเร็วที่ใช้การขยายกระแส**

เซอร์โวแอมปลิไฟเออร์สามารถแบ่งออกได้เป็นสองกลุ่มหลัก ตามชนิดของการป้อนกลับที่มีใช้กัน มีการขยายแรงดันและการขยายกระแส แอมปลิไฟเออร์ทั้งสองชนิดจะต้องศึกษาแยกกัน และฟังก์ชั่นถ่ายโอนโมเดลสำหรับแอมปลิไฟเออร์แต่ละชนิดจะมีลักษณะต่างกัน

**การวิเคราะห์ระบบควบคุมความเร็วที่ใช้การขยายแรงดัน**

แอมปลิไฟเออร์ในอุดมคติควรจะมีอัตราขยายคงที่สำหรับทุกความถี่ อย่างไรก็ตามจะเป็นไปไม่ได้ในทางปฏิบัติ และแอมปลิไฟเออร์ทั้งหลายจะมีย่านที่ถูกจำกัด ถ้าฟังก์ชั่นถ่ายโอนของแอมปลิไฟเออร์มีขั้วเดียวฟังก์ชั่นถ่ายโอนระหว่างคำสั่งอินพุท $V_{in}(s)$ และเอาท์พุท $V_{out}(s)$

## 2.4.5 ระบบควบคุมแรงบิด

ในบางกรณีเราต้องการรักษาให้แรงบิดของมอเตอร์คงที่ เนื่องจากแรงบิดจะเป็นสัดส่วนกับกระแสของมอเตอร์ นั่นหมายความว่าเราจะต้องให้กระแสของมอเตอร์ภาระคงที่

แรงดันอินพุท $V_i = -I_d R$ จะถูกเปรียบเทียบกับแรงดันป้อนกลับ $V_f = I_o R$ และเป็นอินพุทให้กับแอมปลิไฟเออร์ที่มีอัตราขยายสูงถ้ากระแสเอาท์พุท $I_o$ มีค่าต่างไปจากกระแสที่ต้องการ $I_d$ ความแตกต่างดังกล่าวจะถูกขยายและใช้สำหรับไปแก้ไขสภาวะให้ถูกต้อง

## 2.4.6 แอมปลิไฟเออร์ที่ประกอบด้วยแหล่งกำเนิดแรงดันและกระแส

**การขยายแรงดัน**

แหล่งกำเนิดแรงดันจะให้เอาท์พุทอิมพิแดนซ์ (Output Impedance) ต่ำต่อมอเตอร์ คือ จะไม่ทำให้แรงดันเอาท์พุทเปลี่ยนแปลงและสามารถลดการเปลี่ยนแปลงความเร็วของมอเตอร์เนื่องจากการขยายแรงดันสามารถรับ (Sink) กระแสจาก Back EMF Generator ดังนั้น การขยายแรงดันจะให้ปริมาณการแด้มป์ (Damp) จำนวนหนึ่ง ซึ่งสามารถทำให้เสถียรภาพ (Stability) ดีขึ้น

**การขยายกระแส**

แหล่งกำเนิดกระแสจะให้เอาท์พุทอิมพิแดนซ์สูงต่อมอเตอร์ ดังนั้นแอมปลิไฟเออร์กระแสเกือบจะไม่มีการแด้มป์ แอมปลิไฟเออร์แบบนี้จะเป็นประโยชน์สำหรับการขับ Multi Phase หรือ Brushless มอเตอร์ไฟฟ้ากระแสตรงซึ่ง Mutual Coupling ระหว่างขดลวด อาจเป็นสาเหตุ ให้เกิดแด้มป์มากเกินไปถ้าหากใช้การขยายแรงดัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การขยายกระแส สามารถกำจัดระบบที่ขึ้นอยู่กับอินดัคแตนซ์ของมอเตอร์ คือ สามารถหักล้าง Electrical Pole ของมอเตอร์ได้อย่างแท้จริงเมื่อสัญญาณมีการเปลี่ยนแปลงเพียงเล็กน้อย คือการยอมรับของแหล่งกำเนิดกระแสจะมีค่าไม่มากเกินไป การอิ่มตัวของแรงดัน

นอกจากนั้นฟังก์ชั่นถ่ายโอนจะประกอบด้วย ผลของ Integration ระหว่างกระแสและอัตราเร่ง ซึ่งมีผลเสียต่อเสถียรภาพของระบบ

## 2.4.7 การเปลี่ยนแปลงความเร็วของมอเตอร์

การเปลี่ยนแปลงความเร็วของมอเตอร์หักเหไปจากความเร็วที่ต้องการมีสาเหตุมาจากหลายแฟคเตอร์ การแปรไปของความเร็วของมอเตอร์แบ่งออกได้ 2 พวกคือ การแปรความเร็วไปอย่างช้า ๆ เรียก Long Term Speed Variations (LTSV) และการเปลี่ยนแปลงความเร็วของมอเตอร์เป็นไปอย่างรวดเร็วเราเรียกว่า Instantaneous Speed Variation (ISV) เทร็ชโฮลด์ (Trace Hold) ระหว่างลักษณะการเปลี่ยนแปลงทั้ง 2 แบบขึ้นอยู่กับข้อกำหนดในการนำไปใช้งาน ตัวอย่างของระยะเวลาจำกัดสำหรับการเปลี่ยนแปลงของความเร็วคือ 1 นาที ดังนั้น LTSV คือการเปลี่ยนแปลงของความเร็วที่ใช้เวลายาวนานกว่า 1 นาที ส่วน ISV คือการเปลี่ยนแปลงของความเร็วที่สั้นกว่า 1 นาที

LTSV เกิดขึ้นเนื่องจากผลการเปลี่ยนแปลงของค่าพารามิเตอร์ซึ่งเป็นสาเหตุมาจากอุณหภูมิเปลี่ยนแปลงโวลท์เตจขับเคลื่อนและส่วนประกอบชำรุด ในการประยุกต์ใช้งานที่ต้องการเสถียรภาพ เป็นระยะเวลายาวนาน (LTSV) เราจำเป็นต้องใช้ระบบที่มีการเปลี่ยนแปลงของพารามิเตอร์อยู่ภายในค่าที่สามารถยอมรับได้ ถ้าเราไม่สามารถทำให้ค่าการเปลี่ยนแปลงของพารามิเตอร์อยู่ภายในขีดจำกัดได้เราต้องดัดแปลงระบบให้มีความไวต่อการเปลี่ยนแปลงของพารามิเตอร์น้อยที่สุด ระบบดังกล่าวได้แก่ระบบการบังคับความเร็วแบบเฟสล็อคลูป (Phase Lock Loop)

ส่วนในกรณีของ ISV นั้นค่อนข้างยุ่งยากซับซ้านและขึ้นอยู่กับข้อกำหนดของการใช้งานเป็นส่วนใหญ่ ตามปกติเหตุผลหลักของ ISV อธิบายได้ดังต่อไปนี้

**เวลาของผลตอบสนอง**

เมื่อสัญญาณคำสั่งเปลี่ยนแปลง ระบบก็จะเข้าสู่โหลดทรานเชียนท์ ก่อนที่จะเข้าสู่ความเร็วที่ต้องการ ผลตอบสนองทรานเซียนท์นี้สามรถจะคาดหมายได้โดยขึ้นอยู่กับแบนด์วิดท์ ของลูปและเสถียรภาพของระบบ

**การเปลี่ยนแปลงของแรงบิดภาระ**

เมื่อแรงบิดโหลดเปลี่ยนแปลงจะเป็นสาเหตุให้ความเร็วของมอเตอร์เปลี่ยนไป ผลที่เกิดขึ้นนี้ไม่เป็นที่ต้องการ เนื่องจากเราต้องการให้ความเร็วของมอเตอร์เป็นอิสระต่อโหลดความสัมพันธ์ระหว่างการเปลี่ยนแปลงโหลดที่สอดคล้องการเปลี่ยนแปลงของความเร็วมอเตอร์ไม่ค่อยมีใครจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สนใจมากนัก เพราะว่าการวัดผลของการเปลี่ยนแปลงโหลดกระทำได้ยากลำบาก ถ้าเราต้องการให้ผลตอบสนองต่อการเปลี่ยนแปลงของโหลดของระบบเป็นไปอย่างรวดเร็วเราควรจะหลีกเลี่ยงการออกแบบชดเชยระบบด้วยวงจร Lag – Lead ที่มีซีโร (Zero) อยู่ที่ความถี่ต่ำ ๆ เพราะว่าวงจร Lag – Lead จะทำให้ผลตอบสนองของระบบต่อการเปลี่ยนแปลงโหลดเป็นไปอย่างเชื่องช้า

**การกระเพื่อมของแรงบิด**

แรงบิดที่เกิดขึ้นของมอเตอร์ไม่ได้เป็นสัดส่วนที่เที่ยงตรงกับกระแสโดยตรงมันจะค่อย ๆ แปรไปเป็นฟังก์ชั่นกับตำแหน่งเชิงมุมของเพลามอเตอร์ การเปลี่ยนแปลงแรงบิดเหล่านี้เราเรียกว่าการกระเพื่อมของแรงบิด (Torque Ripples) และยังเป็นสาเหตุให้เกิด ISV ผลของการกระเพื่อมของแรงบิดมีความสำคัญอย่างยิ่งที่ความเร็วต่ำ ๆ เพราะว่าความถี่ของการกระเพื่อมต่ำ ถ้าการกระเพื่อมของแรงบิดและความถี่ของมันสามารถวัดได้ เราก็สามารถจะคาดหมายถึงผลการเปลี่ยนแปลงของความเร็วของมอเตอร์ได้

**แรงเสียดทานภายใน**

แรงเสียดทานที่สามารถแปรค่าได้ในตัวมอเตอร์จะทำให้ความเร็วของมอเตอร์เปลี่ยนแปลงไปได้เช่นเดียวกับสาเหตุจากการเปลี่ยนแปลงของโหลด แรงเสียดทานภายในตัวมอเตอร์สามารถจะวัดค่าได้และกำหนดไว้โดยโรงงานผู้ผลิตมอเตอร์นั้น

**สัญญาณรบกวน (Noise)**

การเปลี่ยนแปลงของความเร็วมอเตอร์อาจจะเป็นผลตอบสนองที่มีต่อสัญญาณรบกวนหรือสัญญาณแปลกปลอมจากภายนอกและสัญญาณที่ไม่ต้องการอื่น ๆ ผลของสัญญาณน้อยส์นี้มีความสำคัญอย่างยิ่งในระบบที่ใช้แอมปลิไฟแบบพัลซ์วิดท์โมดูเลชั่นหรือเฟสดีเท็คเตอร์ เราสามารถวัดสัญญาณที่ไม่ต้องการเหล่านี้ได้ที่หลายจุดในระบบ ถ้าเราไม่สามารถจะลดขนาดของสัญญาณเหล่านี้ลงได้เราก็ควรจะเพิ่มความถี่ของสัญญาณเหล่านี้เพื่อลดผลที่จะกระทบกระเทือนต่อความเร็วของมอเตอร์

**สัญญาณรบกวนที่เกิดในทาโคมิเตอร์**

ลักษณะที่เลวร้ายที่สุดของการเปลี่ยนแปลงความเร็วของมอเตอร์เป็นสาเหตุมาจากสัญญาณรบกวนที่เกิดขึ้นในทาโคมิเตอร์เพราะว่าระบบไม่มีการป้องกันการรบกวนของสัญญาณที่เกิดขึ้นดังกล่าวสัญญาณรบกวนเหล่านี้อธิบายได้ในรูปของเออร์เรอร์ความเร็วของระบบซึ่งลูปพยามยามจะแก้ไขเออร์เรอร์ความเร็วดังกล่าว ดังนั้นในการประยุกต์ใช้งานต่าง ๆ ทาโคมิเตอร์ จึงมีความสำคัญอย่างยิ่งที่จะต้องเป็นแบบกำเนิดสัญญาณรบกวนต่ำเมื่อระดับความเที่ยงตรงของความเร็วของระบบขึ้นอยู่กับการป้อนกลับความเร็วนี้ได้จากทาโคมิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.5 ระบบควบคุมตำแหน่งการเคลื่อนที่ (Position Control System)

ระบบควบคุมการเคลื่อนที่ส่วนมากมักต้องการหมุนโหลดให้ไปหยุดในตำแหน่งที่กำหนด และยึดตำแหน่งนั้นไว้จนกว่าจะได้รับคำสั่งควบคุมการหมุนอันใหม่ การเคลื่อนที่อีกแบบหนึ่งของระบบนี้ คือ ในบางครั้งต้องการให้โหลดเคลื่อนที่ไปตามระยะตำแหน่งที่กำหนดให้ข้อกำหนดการเคลื่อนที่ทั้งสองแบบดังกล่าวสามารถกระทำได้โดยอาศัยระบบการควบคุมตำแหน่ง

ระบบการควบคุมตำแหน่ง ตำแหน่งเชิงมุมของมอเตอร์ที่ต่ออยู่กับโหลดจะสอดคล้องกับคำสั่งอินพุท เมื่อสัญญาณควบคุมคงที่เพลาของมอเตอร์ก็จะล็อคกับตำแหน่งที่ต้องการเมื่อสัญญาณควบคุมแปร ไปอย่างต่อเนื่องตำแหน่งของมอเตอร์ก็จะเคลื่อนตามการควบคุมตราบเท่าที่การแปรไปของสัญญาณควบคุมอยู่ภายในย่านของวงรอบ

ตัวเซ็นเซอร์ตำแหน่งอาจเป็นพวกที่ให้สัญญาณเอาท์พุทได้ต่อเนื่อง เช่น โพเทนทิโอมิเตอร์ ซึ่งระบบการควบคุมแบบนี้จะเป็นไปอย่างต่อเนื่องหรือเป็นระบบการควบคุมแบบแอนะลอก ระบบการควบคุมตำแหน่งแบบแอนะลอกมีความบะเอียดในการควบคุมตำแหน่งได้ดีเยี่ยมซึ่งเป็นคุณสมบัติที่เหมาะสมในการควบคุมตำแหน่ง อย่างไรก็ตาม สัญญาณของตำแหน่งที่มีการแปรอย่างต่อเนื่องจะถูกลิมิตอยู่ในช่วงตำแหน่งช่วงหนึ่งเท่านั้น และการจำกัดนี้ จะเป็นตัวจำกัดการเคลื่อนที่

ตัวเซ็นเซอร์ตำแหน่งแบบอื่นได้แก่เซ็นเซอร์แบบดิจิตอล คือ พวกเอนโคดเดอร์สำหรับตำแหน่งของเพลา พวกเซ็นเซอร์แบบนี้มีช่วงการทำงานของตำแหน่งไม่มีขีดจำกัด อย่างไรก็ตาม เซ็นเซอร์พวกนี้มีข้อเสียในเรื่องความละเอียดของการควบคุมตำแหน่งอยู่ภายในค่าที่จำกัดนอกจากนั้นสัญญาณของตำแหน่งจะอยู่ลักษณะของสัญญาณดิจิตอลซึ่งสัญญาณนี้จะต้องป้อนเข้า D/A คอนเวอร์เตอร์ก่อนเพื่อแปลงไปเป็นสัญญาณของตำแหน่งแบบแอนะลอก

มีหลักความจริงอยู่ว่า เราไม่สามารถจะควบคุมตำแหน่งเอาท์พุทของระบบให้เที่ยงตรงได้มากไปกว่าความเที่ยงตรงที่วัดได้จากตัวเซ็นเซอร์ ปัญหาหลักในการออกแบบระบบการควบคุมตำแหน่ง คือ การทำให้ระบบมีเสถียรภาพภายใต้อัตราขยายที่เหมาะสมและมีแบนด์วิดท์ของลูปอย่างพอเพียง ดังนั้นเราจะได้เน้นถึงปัญหาของเสถียรภาพและหลักการออกแบบต่อไประบบการควบคุมแบบป้อนกลับ โดยทั่วไปจุดมุ่งหมายหลัก คือ ต้องการให้ระบบมีความเที่ยงตรงที่สภาวะคงที่ (Steady State Accuracy) เป็นที่น่าพอใจ ดังนั้นระบบจึงไม่นิยมที่จะให้มีการลดอัตราขยายล่งเพื่อทำให้ระบบมีเสถียรภาพ ถ้าอัตราขยายที่ต้องการเพื่อกำหนดความเที่ยงตรงทำให้ระบบไม่มีเสถียรภาพ หรือ ผลตอบสนองมีการแด้มป์ที่เลว เมื่อนั้นระบบจะต้องได้รับการเปลี่ยนแปลง (มีการชดเชย) ในวิถีทางที่จะให้ได้ผลตอบสนองตามที่ต้องการ โดยไม่ทำให้ตามเที่ยงตรงด้อยลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.5.1 การวิเคราะห์ระบบเซอร์โวที่ใช้งานในการควบคุมตำแหน่งหมุนของเพลามอเตอร์ไฟฟ้ากระแสตรง

ระบบเซอร์โวที่ใช้งานสำหรับควบคุมตำแหน่งหมุนของเพลาดีซีมอเตอร์ ซึ่งอาจนำไปใช้งานเป็นสตริปชาร์ทเร็คกอร์ดเดอร์ (Strip Chart Recorder) หรือนำไปใช้สำหรับควบคุมตำแหน่งแขนของหุ่นยนต์ (Robot Arm) ก็ได้ ระบบเซอร์โวที่ประกอบด้วย วงรอบการควบคุมความเร็วรอบและวงรอบการควบคุมแรงบิด ก่อนที่จะได้อธิบายถึงการวิเคราะห์ระบบเซอร์โวต่อไปขอทบทวนถึงความหมายของระบบเซอร์โวกันก่อน ระบบเซอร์โวคือระการควบคุมเครื่องจักรกลที่ไปขับโหลดให้มีการเคลื่อนไหวในลักษณะเป็นไดนามิคของตำแหน่งความเร็วรอบ และแรงบิดเป็นต้น ระบบเซอร์โวจะต้องตอบสนองต่อการเปลี่ยนแปลงของตัวแปรอินพุท r(t) หรือต่อการเปลี่ยนแปลงของภาระ ดังนั้นการวิเคราะห์ระบบเซอร์โวโดยทั่วๆ ไป เราจะต้องพิจารณาถึงคุณสมบัติการทำงานของระบบอยู่ 2 ประการ คือคุณสมบัติไดนามิค และคุณสมบัติสแตติค (Static) หรือ คงที่ของระบบ

### 2.5.2 คุณสมบัติไดนามิคของระบบเซอร์โว

หมายถึง ผลตอบสนองของระบบต่อการเปลี่ยนแปลงของตัวแปรอินพุท (Set Point) หรือผลตอบสนองของระบบต่อการเปลี่ยนแปลงภาระ (External Disturbance)

ผลตอบสนองของระบบเซอร์โวต่อการเปลี่ยนแปลงของสเตปอินพุทมีลักษณะเป็นไดนามิคแบ่งออกได้เป็น 3 ลักษณะ

1. ผลตอบสนองของระบบจะมีการเปลี่ยนแปลงลดลงเป็นไซเกิล เมื่อแด้มป์ปิ้งแฟคเตอร์ของระบบมีค่าน้อยกว่าหนึ่ง

2. ผลตอบสนองของระบบจะมีการเปลี่ยนแปลงลดลงเป็นวิกฤต เมื่อแด้มป์ปิ้งแฟคเตอร์ของระบบมีค่าเท่ากับหนึ่ง

3. ผลตอบสนองของระบบจะมีการเปลี่ยนแปลงลดลงอย่างช้าๆ เมื่อแด้มป์ปิ้งแฟคเตอร์ของระบบมีค่ามากกว่าหนึ่ง

**ตัวแปรที่เป็นตัวกำหนดคุณสมบัติไดนามิคของระบบเซอร์โว** ได้แก่

**อัตราขยายวงรอบ (loop gain)**
- ค่าสูงจะเป็นผลให้ตอบสนองของระบบลดลงเป็นไซเกิล
- ค่าปานกลางจะเป็นผลให้ผลตอบสนองของระบบลดลงเป็นวิกฤต
- ค่าต่ำจะเป็นผลให้ผลตอบสนองของระบบลดลงอย่างช้า ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**อัตราขยายของวงรอบป้อนกลับ** - ค่าสูงจะเป็นผลให้ตอบสนองของระบบลดลงอย่างช้า ๆ

**ความเร็ว**
- ค่าปานกลางจะเป็นผลให้ผลตอบสนองของระบบลดลงเป็นวิกฤต
- ค่าต่ำจะเป็นผลให้ผลตอบสนองของระบบลดลงเป็นไซเกิล

**ฟริคชั่นและแรงเฉื่อย**
- ค่าสูงจะเป็นผลให้ผลตอบสนองของระบบลดลงอย่างช้า ๆ
- ค่าปานกลางจะเป็นผลให้ผลตอบสนองของระบบลดลงเป็นวิกฤต
- ค่าต่ำจะเป็นผลให้ผลตอบสนองของระบบลดลงเป็นไซเกิล

**ผลการป้อนกลับของแรงบิด** - จะทำให้ผลตอบสนองของระบบเป็นอิสระต่อภาระแรงบิด

### 2.5.3 คุณสมบัติสแตติคของระบบเซอร์โว

คือคุณสมบัติของระบบเมื่อผลตอบสนองของระบบเข้าสู่สถานะคงที่ (Steady State) แล้ว ซึ่งแสดงอยู่ในรูปของเออร์เรอร์ที่สถานะคงที่หรือความแตกต่างระหว่างตัวแปรอินพุท r(t) และตัวแปรเอาท์พุท c(t) เมื่อเวลาเข้าใกล้อินฟินิตี้ (Infinity) หมายถึง ความเที่ยงตรงของระบบ

**ตัวแปรที่เป็นตัวกำหนดความผิดพลาดที่สถานะคงที่** ได้แก่

**อัตราขยายของวงรอบ (Loop Gain)**
- เมื่อมีค่าสูง Error มีค่าน้อย (หมายความว่าระบบมีความเที่ยงตรงดี)
- ค่าปานกลาง Error มีค่าปานกลาง (หมายความว่าระบบมีความเที่ยงตรงพอใช้ได้)
- ค่าต่ำ Error มีค่ามาก (หมายความว่าระบบมีความเที่ยงตรงไม่ดี)

## 2.6 ไมโครคอนโทรลเลอร์ (Micro Controller)

ไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีขนาด 8 บิต ประกอบด้วยไมโครคอนโทรลเลอร์เบอร์ต่างๆ ทุกๆ เบอร์จะมีสถาปัตยกรรมพื้นฐานที่เหมือนกัน เพียงแต่มีขนาดหน่วยความจำภายในและภายนอกที่แตกต่างกัน เพื่อความเหมาะสมในการนำเอาไปใช้งานตามความต้องการต่างๆ แต่เดิม 8051 ถูกสร้างด้วยวิธี HMOS I แต่ในปัจจุบันได้สร้างด้วยวิธี HMOS II จึงมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อเป็น 8051AH ไมโครคอนโทรลเลอร์ในตระกูล 51 นั้น ถึงแม้ว่าจะมีหลายเบอร์แต่เราก็จะเรียกว่าเป็น "8051" ซึ่งหมายถึงไมโครคอนโทรลเลอร์ตระกูล51 นั้น ส่วนเบอร์ 8032 และ 8052 มีหน่วยความจำภายในเพิ่มขึ้นและมีวงจรนับ/จับเวลา ขนาด 16 บิตเพิ่มขึ้นมาดังตารางที่ 2.1

**ตารางที่ 2.1** ไมโครคอนโทรลเลอร์ตระกูล 8051

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

### 2.6.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

1. ต้องการแหล่งจ่ายไฟ + 5 V. ชุดเดียว
2. มีหน่วยความจำโปรแกรม ( Program Memory ) ขนาด 4 กิโลไบต์ สำหรับเบอร์ 8051 และ 8031 สำหรับเบอร์ 8052 มีหน่วยความจำถึง 8 กิโลไบต์
3. มีหน่วยความจำสำหรับเก็บข้อมูล ( Data Memory ) ขนาด 128 ไบต์ สำหรับเบอร์ 8052 ขึ้นไปมีถึง 256 ไบต์
4. มีหน่วยความจำสำหรับเก็บโปรแกรมและข้อมูลแยกจากกันอย่างละ 64 กิโลไบต์
5. มีไทม์เมอร์เคาน์เตอร์ (Timer Counter) ขนาด 16 บิต 2 ชุด ( สำหรับเบอร์ 8052 มี 3 ชุด ) ทำงานได้ 4 โหมด
6. รับอินเตอร์รัพท์ (Interrupt) ได้ 6 แหล่ง 5 เวคเตอร์ สำหรับเบอร์ 8052 ขึ้นไปมี 8 แหล่ง 6 เวคเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

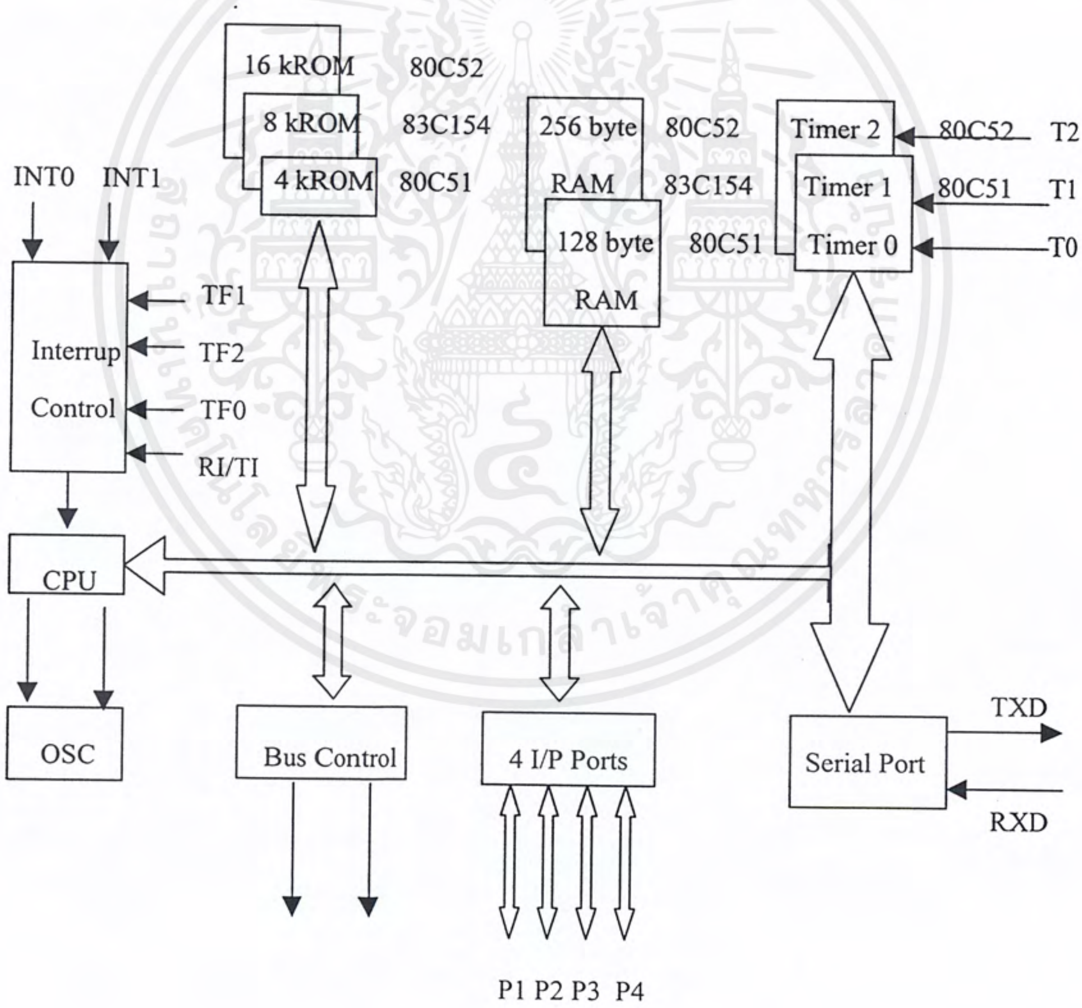
7. มีพอร์ท (Port) รับส่งข้อมูลอนุกรม ( UART ) 2 พอร์ทแบบ Full Duplex เลือกรับได้ 4โหมด

8. มีคำสั่งในการทำ AND, OR หรือ Complement ได้ทั้งแบบ 8 บิตและ 1 บิต

9. มีวงจรออสซิเลเตอร์ภายใน

### 2.6.2 โครงสร้างของไมโครคอนโทรเลอร์ตระกูล MCS-51

ภายใน 8051 จะประกอบด้วยเกทชนิดต่างๆ เช่น AND, OR, NOT ซึ่งแต่ละเกทเหล่านี้จะนำมาออกแบบให้มีหน้าที่การทำงานต่างๆ เช่น วงจรถอดรหัสคำสั่ง, วงจรสัญญาณนาฬิกา เป็นต้น โครงสร้างภายในของ 8051 จะประกอบด้วยส่วนย่อยๆ ดังรูป 2.17




**รูปที่ 2.17** โครงสร้างของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.3 โครงสร้างของ 8051

จะประกอบด้วย 3 ส่วน หลักๆ ดังนี้

**1. ซีพียู ( Central Processing Unit )**

ซีพียู จะทำหน้าที่ สร้างสัญญาณควบคุมการติดต่อกับส่วนอื่นๆ เรียกว่า วงจรควบคุม (Control Unit) สัญญาณที่สร้างจากวงจรควบคุม ได้แก่ สัญญาณสำหรับการติดต่อกับหน่วยความจำ , อุปกรณ์รับข้อมูลเข้าหรือส่งข้อมูลออก ซึ่งส่วนควบคุมการขัดจังหวะ และส่วนควบคุมบัสก็เป็นส่วนหนึ่งของวงจรควบคุมด้วย การสร้างสัญญาณวงจรควบคุมจากซีพียูนี้จะทำการสร้างสัญญาณโดยการถอดรหัสจากคำสั่งที่มีการกำหนดไว้และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างขึ้นจากวงจรออสซิเลเตอร์เพื่อให้ทุกๆส่วนทำงานประสานกันอย่างถูกต้อง

ในซีพียูยังประกอบด้วยส่วนประมวลผล ( Arithmetic Logic Unit ) ทำหน้าที่ประมวลผลข้อมูล เช่น การบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในหน่วยความจำที่ต้องการ

**2. หน่วยความจำ ( Memory )**

หน่วยความจำ มีไว้สำหรับจดจำข้อมูล ซึ่งในการนำข้อมูลเข้าและออกจากหน่วยความจำ เราจำเป็นต้องรู้ตำแหน่งของหน่วยความจำ ( Address ) ในการนำข้อมูลเข้าไปเก็บในหน่วยความจำ เรียกว่า การเขียนข้อมูล และการนำข้อมูลออกจากหน่วยความจำ เรียกว่า การอ่านข้อมูล ในไมโครคอนโทรลเลอร์ 8051 ข้อมูลในแต่ละตำแหน่งจะมีขนาด 8 บิต ดังนั้นแต่ละตำแหน่งของหน่วยความจำสามารถเก็บความจำข้อมูลมีค่าได้ระหว่าง $00000000_2$ ถึง $11111111_2$ หรือ 00H ถึง 0FFH ในการติดต่อหน่วยความจำจะต้องมีสัญญาณสามกลุ่มคือ

1. ตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ซึ่งในไมโครคอนโทรลเลอร์ 8051 จะมีหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลที่มีขนาดสูงสุดชนิดละ 65536 ตำแหน่ง (64 กิโลไบต์ ) ดังนั้นการอ้างตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งของหน่วยความจำจะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2 ทั้งหมด 16 เส้น ( $2^{16}$ เท่ากับ 65536 )

2. ข้อมูลที่อ่านหรือเขียนกับหน่วยความจำในตำแหน่งที่เราต้องการ

3. สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำเพื่อจะบอกกับหน่วยความจำว่าต้องการอ่านหรือเขียนข้อมูล โดยวงจรถอดรหัสคำสั่งจะทำการสร้างสัญญาณควบคุมจากคำสั่งที่อ่านเข้ามาจากหน่วยความจำโปรแกรม

**3. อุปกรณ์อินพุท/เอาท์พุท ( Input / Output Device )**

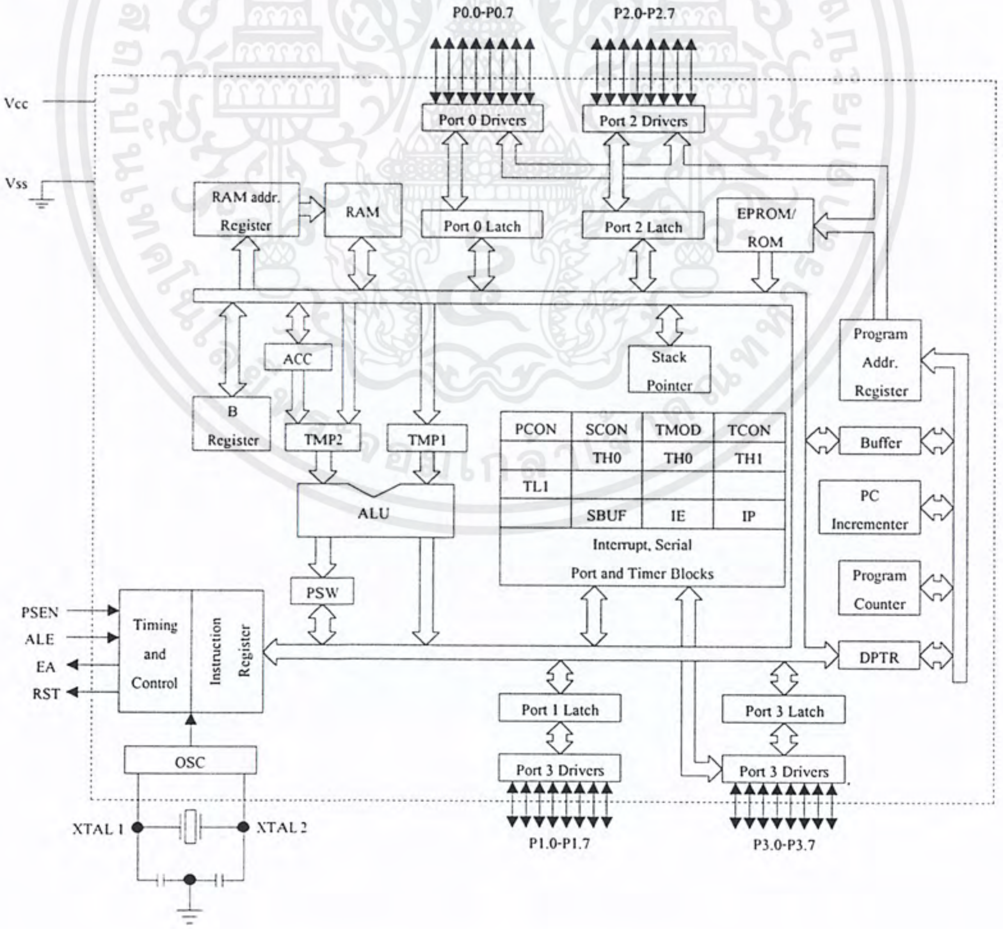
อุปกรณ์อินพุท/เอาท์พุท จะเป็นส่วนใช้ส่งข้อมูลเข้าหรือนำข้อมูลออกจากไมโครคอนโทรลเลอร์ 8051 ทำให้สามารถติดต่อกับอุปกรณ์ภายใน อุปกรณ์อินพุท/เอาท์พุท ได้แก่ อินพุท/เอาท์พุทพอร์ตแบบขนาน วงจรนับ/จับเวลา 0 วงจรนับ/จับเวลา 1 พอร์ตสื่อสารอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. พอร์ตแบบขนาน เป็นที่สำหรับใช้รับส่งข้อมูล ซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัวไมโครคอนโทรลเลอร์เบอร์ 8051 มีทั้งหมด 4 พอร์ต โดยแต่ละพอร์ตจะรับส่งข้อมูลได้ 8 บิต มีพอร์ต P0, P1, P2 และ P3 บางพอร์ตใช้งานได้มากกว่า 1 อย่าง

2. วงจรนับเวลา/จับเวลา 0 และวงจรนับ/จับเวลา 1 เป็นวงจรที่สามารถทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอกของไมโครคอนโทรลเลอร์เบอร์ 8051 หรือจำนวนของสัญญาณนาฬิกาภายในไมโครคอนโทรลเลอร์เบอร์ 8051 ก็ได้ สามารถตั้งค่าเริ่มต้นของการนับและอ่านค่าการนับได้โดยซีพียู

3. พอร์ตอนุกรม ซีพียูจะอ่านและเขียนข้อมูล พอร์ตอนุกรมเป็นแบบ 8 บิต แต่ละข้อมูลจะถูกส่งออกมาจากไมโครคอนโทรลเลอร์เบอร์ 8051 เรียงไปทีละบิตออกจากขา TXD และในการรับข้อมูลก็จะรับเข้ามาทีละบิตทางขา RXD และจัดเรียงใหม่เป็น 8 บิต เพื่อให้ซีพียูอ่านไปใช้งานต่อไป ในไมโครคอนโทรลเลอร์เบอร์ 8051 มีพอร์ตใช้งานได้หลายแบบทำให้สะดวกแก่การนำไปใช้งานต่างๆ ได้มากมาย การนำพอร์ตไปใช้งานจะต้องเขียนโปรแกรมขึ้นมาควบคุม




**รูปที่ 2.18** สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์เบอร์ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.4 สถาปัตยกรรมภายในของ 8051

MCS-51 ใช้เทคโนโลยีในการผลิตแบบ NMOS และ CMOS เบอร์8032 และ 8052 จะมี ROM BASIC อยู่ภายในจึงสะดวกสำหรับโปรแกรมเมอร์ที่จะเขียนโปรแกรมด้วยภาษาเบสิก โครงสร้างภายในสำหรับเบอร์ 8051 ดังแสดงดังรูปที่ 2.18 ซึ่งจะอธิบายถึงส่วนประกอบย่อยๆ ภายในตัวของไมโครคอนโทรลเลอร์เบอร์ 8051 และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขาของไมโครคอนโทรลเลอร์ 8051

ไมโครคอนโทรลเลอร์เบอร์ 8051 บรรจุอยู่ภายในวงจรรวมแบบ DIP หรือ Dual Inline Package แบบ 40 ขาดังนี้

Vcc ( ขา 40 ) ต่อกับไฟ +5V.

Vss ( ขา 20 ) เป็นขา GND

Port 0 ( ขา 32-39 ) มี 8 บิต คือ P0.0 – P0.7 มีโครงสร้างแบบ Open-Drain Bi-Directional โดยสามารถใช้งานได้ 2 หน้าที่คือ Address Bus และ Data Bus นอกจากนี้ยังใช้งานเป็นอินพุตเอาท์พุตได้

Port 1 (ขา 1-8 ) มี 8 บิต คือ P1.0 – P1.8 ใช้งานเป็นอินพุตและเอาท์พุตพอร์ตทั่วไป

Port 2 ( ขา 21-28 ) มี 8 บิต คือ P2.0 – P2.7 ใช้งานเป็นอินพุต/เอาท์พุตพอร์ตทั่วไปแล้วยังใช้เป็นตัวส่ง Address ไบต์สูง เพื่อติดต่อกับหน่วยความจำภายนอก

Port 3 ( ขา 10-17 ) มี 8 บิต คือ P3.0 – P3.7 ใช้งานเป็นอินพุต/เอาท์พอร์ตทั่วไปและใช้งานในหน้าที่พิเศษดังนี้

3.0/RXD ( Serial Input Port ) ใช้รับข้อมูลแบบอนุกรม

3.1/TXD ( Serial Output Port ) ใช้ส่งข้อมูลแบบอนุกรม

3.2/INT0 ( External Interrupt 0 ) ใช้รับสัญญาณขัดจังหวะจากภายนอกเบอร์0

3.3/INT1 ( External Interrupt 1 ) ใช้รับสัญญาณขัดจังหวะจากภายนอกเบอร์1

3.4/T0 ( Counter 0 External Input ) ใช้เป็นอินพุตให้วงจรนับ/จับเวลา ชุดที่ 0

3.5/T1 ( Counter 1 External Input ) ใช้เป็นอินพุตให้วงจรนับ/จับเวลา ชุดที่ 1

3.6/WR ( External Data Memory Write Strobe ) ขาควบคุมการเขียนข้อมูลจากหน่วยความจำภายนอก

3.7/RD ( External Data Memory Read Strobe ) ขาควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอก

RST ( ขา 9 ) Reset ใช้สำหรับรีเซ็ตวงจรภายในชิพ เพื่อเริ่มต้นการทำงานใหม่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การรีเซ็ตต้องป้อนลอจิก “1” นานอย่างน้อย 2 แมชชีนไซเคิล

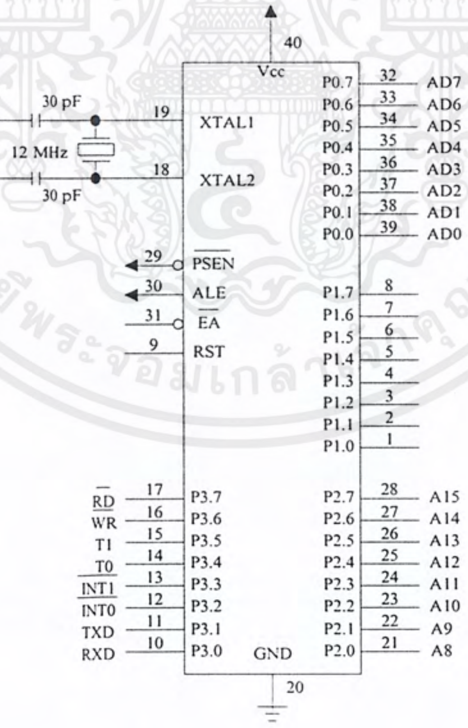
ALE ( ขา 30 ) Address Latch Enable เป็นขาส่งสัญญาณออกไปภายนอก เพื่อควบคุมการLatch ค่า Address ไบต์ต่ำจากพอร์ต 0

PSEN ( ขา 29 ) Program Strobe Enable เป็นสัญญาณเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อขานี้ Active มีลอจิกเป็น “0” จะอ่านโปรแกรมจากหน่วยความจำภายนอกและถ้าเป็นการอ่านโปรแกรมภายในขานี้จะไม่มีการ Active

EA ( ขา 31 ) External Access เป็นขาที่ใช้สำหรับเลือกว่าให้ทำงานจากหน่วยความจำโปรแกรมภายในหรือหน่วยความจำภายนอกชิพ เมื่อขานี้ Active จะมีลอจิกเป็น “0” จะเป็นการทำงานตามคำสั่งในหน่วยความจำโปรแกรมภายนอก

XTAL1 ( ขา 19 ) ใช้ต่อคริสตัลภายนอก โดยอินพุตเข้าสู่วงจรออสซิเลเตอร์

XTAL2 ( ขา 18 ) ใช้ต่อคริสตัลภายนอก โดยเอาท์พุตออกจากวงจรออสซิเลเตอร์




**รูปที่ 2.19** แสดงขาต่างๆ ของ 8051

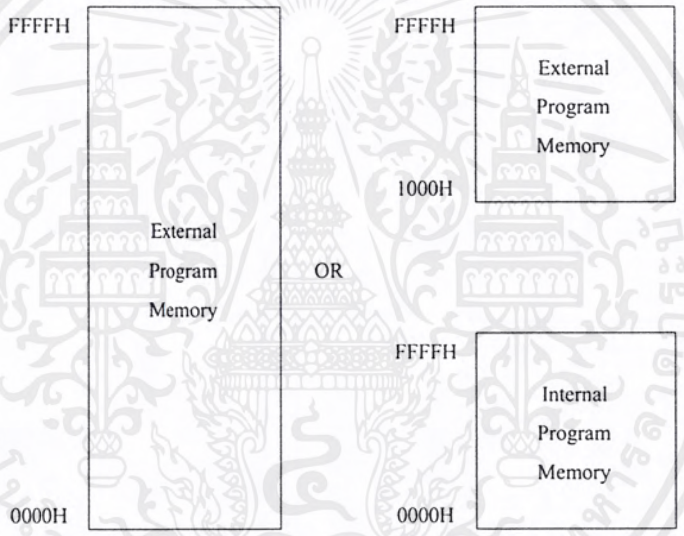
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2.6.5 การจัดการหน่วยความจำของ 8051

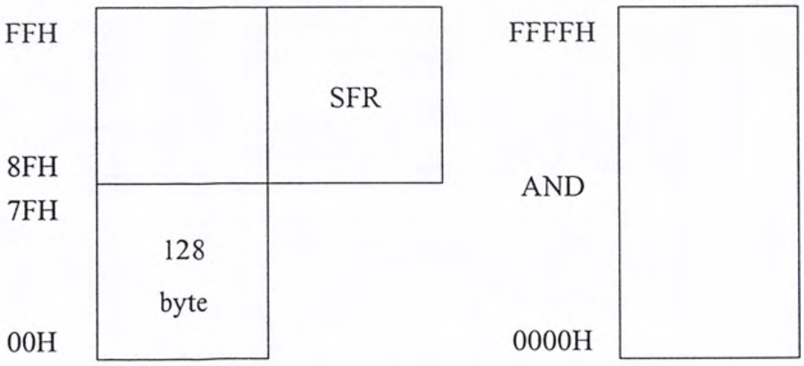
หน่วยความจำของ 8051 แบ่งออกเป็น 2 แบบตามลักษณะการใช้งาน ดังนี้

**1. หน่วยความจำโปรแกรม ( Program Memory )**

เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปของภาษาเครื่องซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำโปรแกรมไปทำการถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่นๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำนี้เป็นแบบรอมและผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ ส่วนที่เป็นหน่วยความจำโปรแกรมก็คือรอมขนาด 4 กิโลไบต์ ดังรูปที่ 2.20




**รูปที่ 2.20** การจัดพื้นที่หน่วยความจำสำหรับโปรแกรมของ 8051

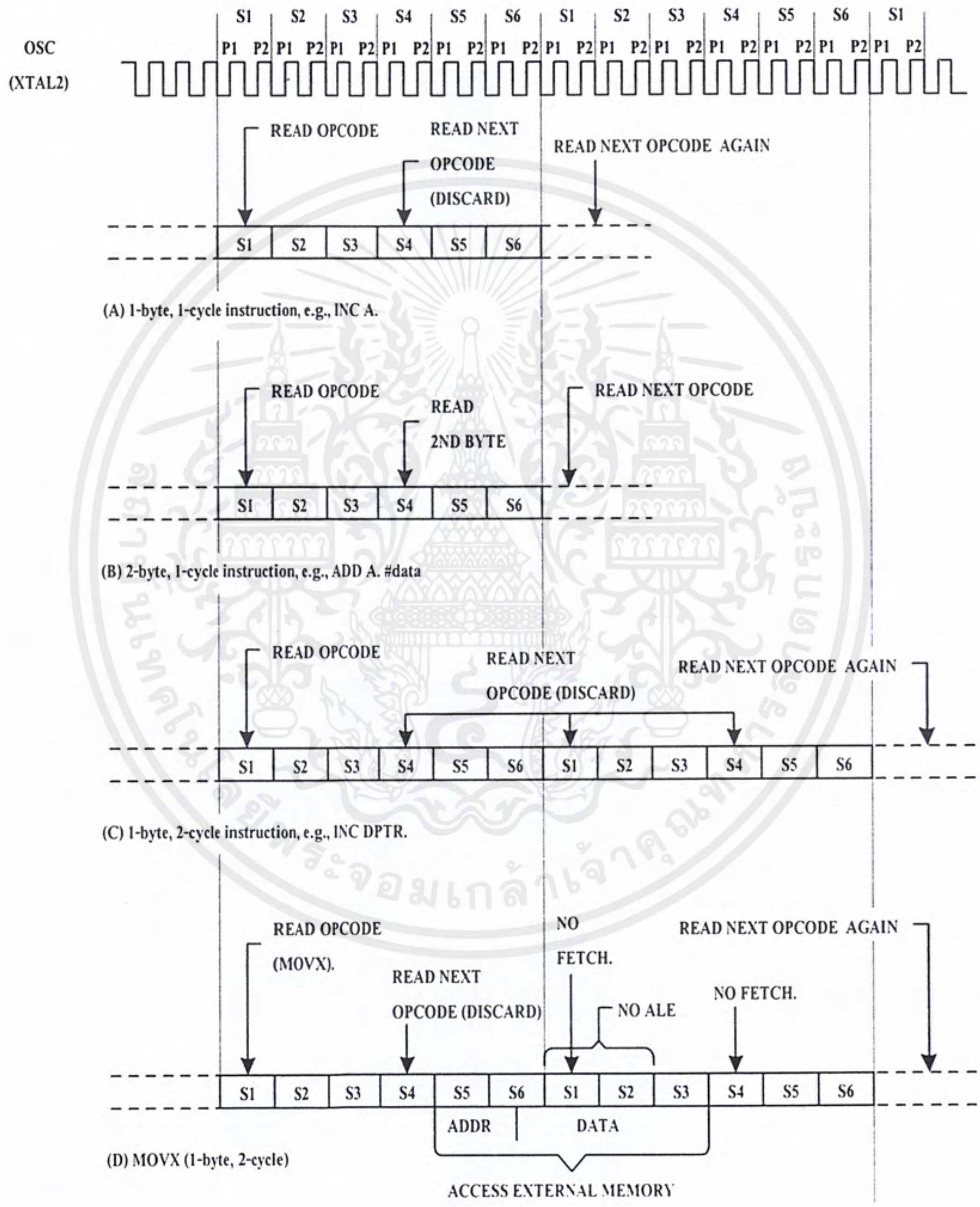



**รูปที่ 2.21** การจัดพื้นที่หน่วยความจำข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 2. หน่วยความจำข้อมูล (Data Memory)

เป็นหน่วยความจำที่ใช้เก็บข้อมูล สามารถอ่านและเขียนข้อมูลได้ ซึ่งหน่วยความจำภายในมีขนาดเพียง 128 ไบต์ ส่วนหน่วยความจำภายนอกไอซีมีขนาด 64 กิโลไบต์ ดังรูปที่ 2.21




**รูปที่ 2.22** ผังเวลาของซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.6.6 ฐานเวลาในการทำงานของไมโครคอนโทรลเลอร์

แมชชีนไซเคิล ( Machine Cycle ) คือรอบการทำงานของคำสั่ง เป็นค่าที่น้อยที่สุดในการทำคำสั่งใดคำสั่งหนึ่ง ถ้าเป็นคำสั่งที่ซับซ้อนมากก็ต้องใช้เวลานาน 2-3 แมชชีนไซเคิล

1 แมชชีนไซเคิล จะประกอบด้วยสัญญาณนาฬิกาจำนวน 12 ลูก โดยสัญญาณนาฬิกาแต่ละลูกเรียกว่า " เฟส " ( Phase ) สัญญาณนาฬิกา 2 เฟส รวมกันเป็น 1 สเตท ( State ) เพราะฉะนั้นใน 1 แมชชีนไซเคิลจึงมี 6 สเตท

### 2.6.7 การทำงานของ 8051

เมื่อป้อนไฟเลี้ยงให้กับไมโครคอนโทรลเลอร์เบอร์ 8051 ซึ่งมีวงจรรีเซ็ตเมื่อปิดเครื่องจะเกิดการรีเซ็ตการทำงานภายในไมโครคอนโทรลเลอร์เบอร์ 8051 เริ่มจากภาคโปรแกรมเคาน์เตอร์ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนเส้นทางหมายเลข 1 เส้นทางนี้มีขนาด 16 บิต ค่าตำแหน่งหน่วยความจำนี้ถูกส่งไปเก็บไว้ที่ Program ADDR Register ค่าตำแหน่งหน่วยความจำจะปรากฏลงบนบัส 16 บิต หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากการรีเซ็ต ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมการเลือกได้ว่าเป็นรอมภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วนของวงจรเวลาและควบคุม ถ้าป้อนสัญญาณลอจิก 0 เข้าที่ขา EA เป็นการเลือกใช้รอมภายใน 8051 โดยที่วงจรเวลาและควบคุมจะสร้างสัญญาณไปยังรอมภายใน ให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมายังเส้นทางหมายเลข 2 ข้อมูลจากรอมถูกส่งไปยังเส้นทางหมายเลข 3 ที่เรียกว่าเส้นทางข้อมูลภายใน แล้วนำไปเก็บไว้ที่รีจิสเตอร์ IR ( Instruction Register ) เพื่อส่งไปให้กับวงจรเวลาและควบคุมทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆต่อไป ในกรณีที่เลือกรอมภายนอก โดยป้อนลอจิก 1 เข้าที่ขา EA จะทำให้วงจรเวลาและควบคุมส่งสัญญาณไปยังพอร์ต 0 และพอร์ต 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนเส้นทางหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ต 0 ไปยังเส้นทางข้อมูลภายในแล้วไปเก็บไว้ที่รีจิสเตอร์ IR เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจากรอมภายใน การทำงานในช่วงค่าตำแหน่งในหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ในรีจิสเตอร์ IR เรียกว่า " ช่วงของการเฟตช์ ( Fetch )" ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า "Execute Cycle"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# การออกแบบ การสร้าง และการทำงาน

ขั้นตอนของการออกแบบ การสร้างและการทำงานของชุดทดลองการควบคุมมอเตอร์ ดีซีด้วยวงจรพัลซ์สามารถแบ่งออกได้เป็นส่วน ๆ คือ ส่วนของการออกแบบทางอิเล็กทรอนิกส์ และส่วนของการออกแบบอุปกรณ์ทางแมคคานิกส์ ซึ่งจะกล่าวถึงส่วนต่าง ๆ ต่อไปนี้

## 3.1 การออกแบบอุปกรณ์ทางอิเล็กทรอนิกส์

ในการออกแบบอุปกรณ์ทางอิเล็กทรอนิกส์นี้ ทางผู้จัดทำได้ค้นคว้ามาจากหนังสือต่างๆ ด้วยกัน เพื่อนำมาประกอบเป็นความรู้ในการจัดทำส่วนของอุปกรณ์ทางอิเล็กทรอนิกส์ ทั้งนี้ทางผู้จัดทำได้พยายามทำส่วนของอุปกรณ์อิเล็กทรอนิกส์ โดยวิธีการลองผิดลองถูก ซึ่งก็ได้ผลมาเป็นที่น่าพอใจอย่างมาก

โครงสร้างและส่วนประกอบของอุปกรณ์ทางอิเล็กทรอนิกส์ ได้แก่

1. ภาคจ่ายไฟ
2. ภาคระบบควบคุม
3. ภาคขับมอเตอร์
4. ภาคระบบป้อนกลับ
5. ภาคอินพุท
6. ภาคเอาท์พุท
7. ตัวตรวจจับตำแหน่ง

### 3.1.1 ภาคจ่ายไฟ

ภาคจ่ายไฟในโครงงานนี้ จะใช้แหล่งจ่ายไฟแบบแหล่งจ่ายไฟคู่ ขนาดแรงดัน 15 V ซึ่งจะใช้เป็นแหล่งจ่ายในการขับมอเตอร์ และแหล่งจ่ายไฟแบบแหล่งจ่ายไฟเดี่ยวขนาดแรงดัน 5 V ซึ่งจะใช้เป็นแหล่งจ่ายให้กับภาคระบบควบคุมต่างๆ

ภาคจ่ายไฟนี้มีอุปกรณ์ที่สำคัญ ดังนี้

1. หม้อแปลงแบบแหล่งจ่ายไฟคู่ ขนาด 220V / 15V 0V 15V และ หม้อแปลงแบบแหล่งจ่ายไฟเดี่ยว ขนาด 220V / 6V 0V

2. Bridge Rectifier

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. IC. Regulator เบอร์ 7815

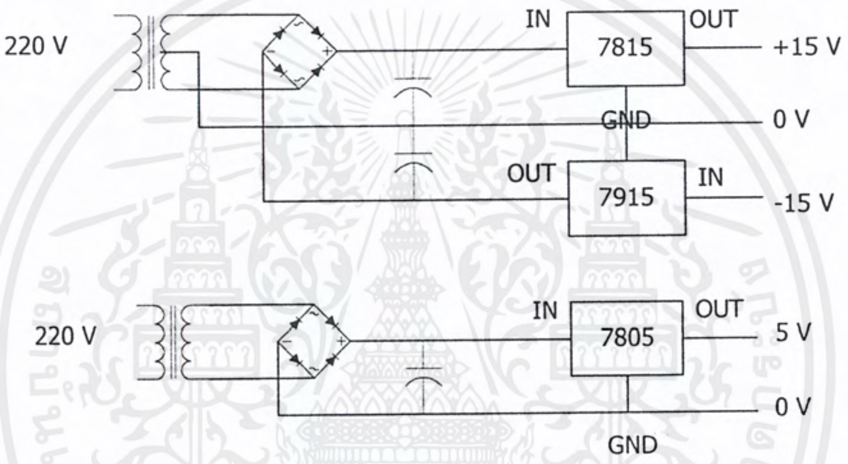
เบอร์ 7915

เบอร์ 7805

4. Capacitor ขนาด 2200 μF 25 V

ขนาด 2200 μF 50 V

**วงจร**




**รูปที่ 3.1** วงจรแสดงภาคจ่ายไฟ (แบบแหล่งจ่ายไฟคู่ และแบบแหล่งจ่ายไฟเดี่ยว)

**การทำงาน**

เมื่อป้อนไฟเข้าแหล่งจ่ายที่หม้อแปลงด้านขดปฐมภูมิแล้ว ในด้านทุติยภูมินั้นจะต่ออยู่กับวงจรเรียงกระแสซึ่งเป็นตัวแปลงไฟฟ้ากระแสสลับให้เป็นไฟฟ้ากระแสตรง จากนั้นจะผ่านตัวเก็บประจุเพื่อทำให้สัญญาณที่ได้เรียบขึ้น และตัวเก็บประจุจะต่ออยู่กับ IC. Regulate เพื่อนำค่าที่เราต้องการไปใช้งาน

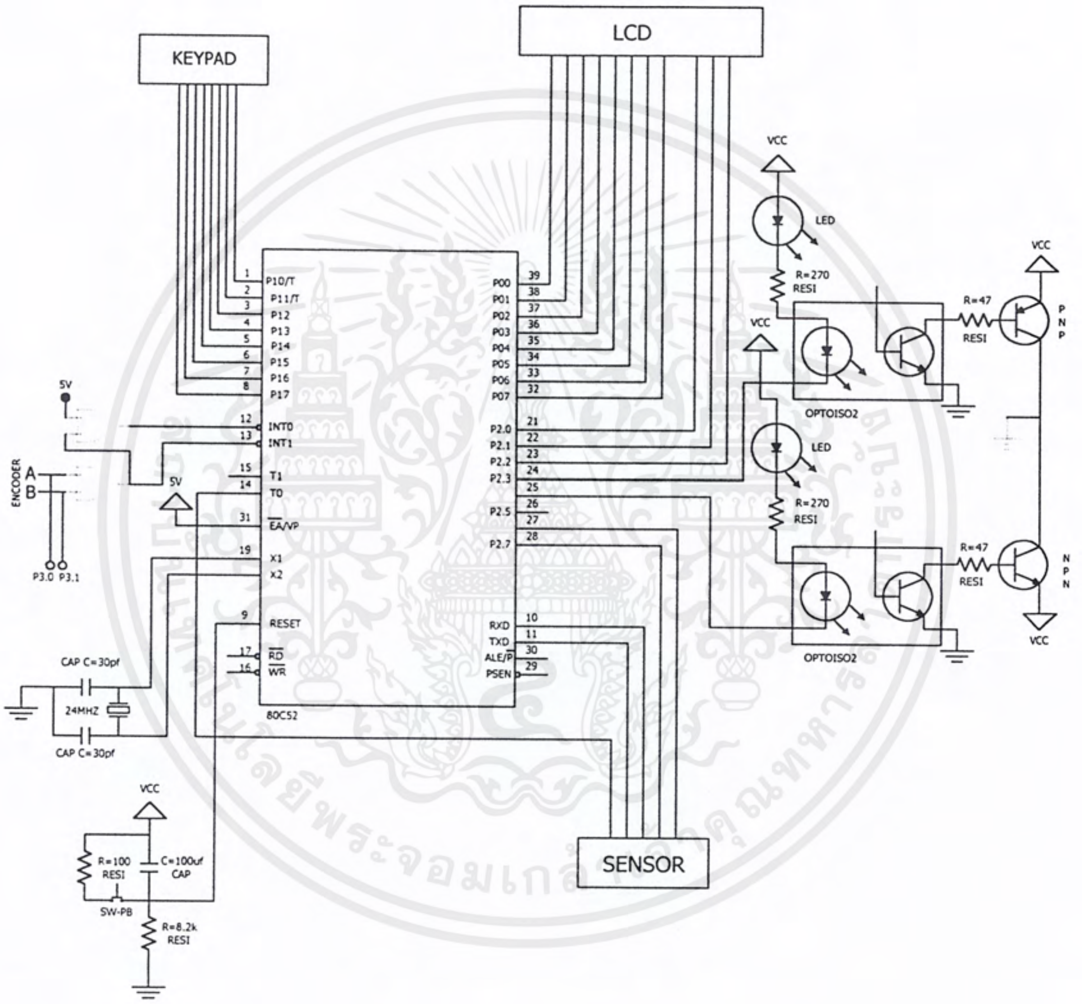
จะเห็นได้ว่ามีภาคจ่ายไฟอยู่ 2 ชุด ชุดแรกคือ ชุดที่เป็นแหล่งจ่ายไฟคู่นั้น ใช้เป็นแหล่งจ่ายในการขับมอเตอร์ ดังที่ได้กล่าวมาแล้ว และชุดที่สองคือ ชุดที่เป็นแหล่งจ่ายไฟเดี่ยว ซึ่งใช้เป็นแหล่งจ่ายให้กับภาคระบบควบคุมต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.2 ภาคระบบควบคุม

ในภาคนี้เป็นหัวใจสำคัญต่อระบบมาก เนื่องจากว่ามีไมโครคอนโทรลเลอร์อยู่ด้วย ซึ่งเป็นตัวควบคุมระบบการทำงานในภาคนี้ทั้งหมด

**วงจร**




**รูปที่ 3.2** วงจรแสดงภาคระบบควบคุม

### 3.1.3 ภาคขับมอเตอร์

ในภาคนี้เป็นการควบคุมมอเตอร์เป็นสำคัญ ซึ่งจะต้องมีการออกแบบอุปกรณ์ทางอิเล็กทรอนิกส์ให้ได้เหมาะสมกับตัวมอเตอร์เอง

ภาคขับมอเตอร์มีอุปกรณ์ที่สำคัญ ดังนี้

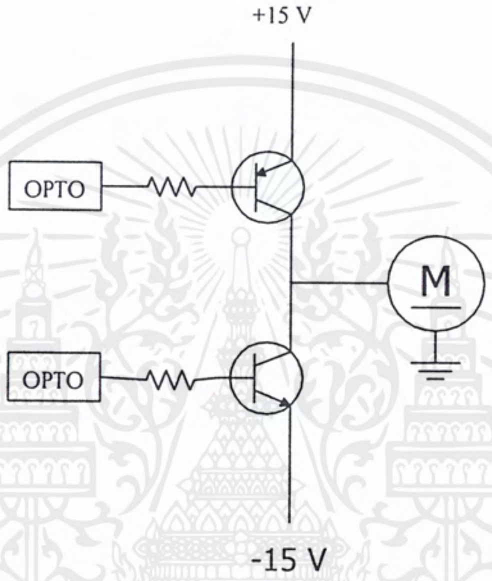
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ทรานซิสเตอร์กำลัง (Power Transistor) เบอร์ MJ2955

เบอร์ 2N3055

2. OPTO Isolate Transistor

**วงจร**




**รูปที่ 3.3** วงจรแสดงภาคขับมอเตอร์

**การทำงาน**

ในการขับมอเตอร์ จะใช้แหล่งจ่ายไฟคู่ เพื่อเป็นประโยชน์ต่อการกลับทางหมุนของมอเตอร์ โดยจะใช้ OPTO Isolate Transistor เป็นตัวควบคุมมอเตอร์ คือ ถ้า Drive Low คือ ลอจิก (Logic) เป็น "0" จะเป็นการสั่งให้ OPTO Isolate Transistor ทำงาน แต่ถ้า Drive High คือ ลอจิกเป็น "1" ก็จะเป็นการสั่งให้ OPTO Isolate Transistor ไม่ทำงาน

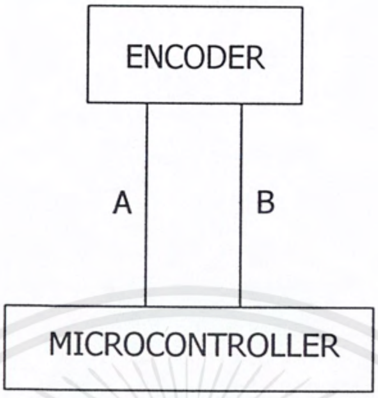
ในการกลับทางหมุนของมอเตอร์นั้น OPTO Isolate Transistor จะทำงานสลับกัน จึงทำให้กลับทางหมุนมอเตอร์ได้

### 3.1.4 ภาคระบบป้อนกลับ

ภาคระบบป้อนกลับ เป็นภาคที่มีการตรวจสอบผลที่ได้จาก Output มาทำการวิเคราะห์ดูว่าระบบปลายทางสามารถทำงานได้ตรงตามที่ต้องหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร

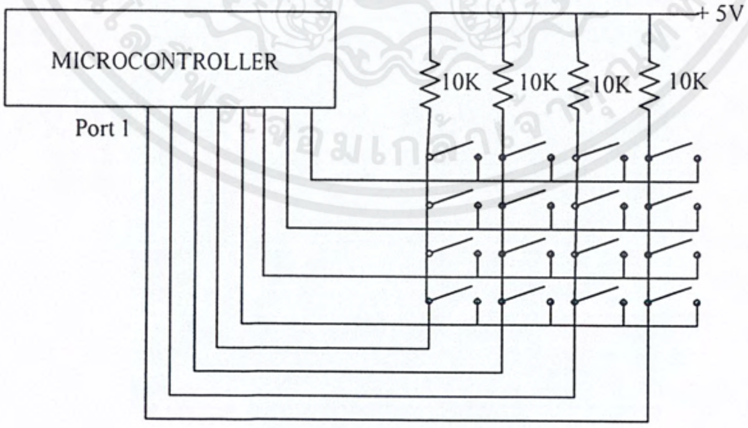



**รูปที่ 3.4** วงจรแสดงภาคระบบป้อนกลับ

### 3.1.5 ภาคอินพุท

ภาค Input เป็นภาคที่มีการป้อนข้อมูลใดๆ เข้ามาเพื่อต้องการให้ตอบสนองตามคำสั่ง Input ที่ใช้ในการโครงงานนี้ ก็คือ คีย์แพ็ด (Key Pad) ซึ่งเป็นตัวป้อนโปรแกรมให้ทำงานตามที่ต้องการ

วงจร



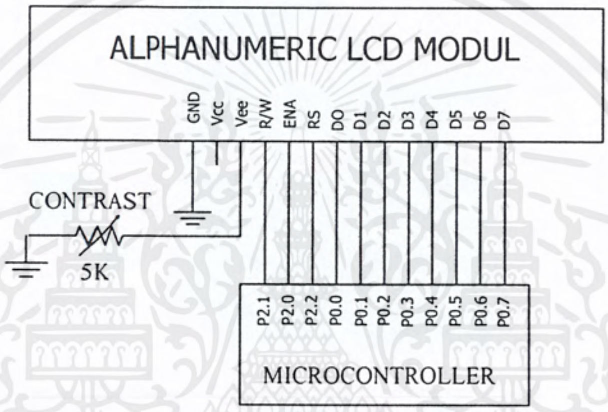

**รูปที่ 3.5** วงจรแสดงภาคอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.1.6 ภาคเอาท์พุท

ในภาคนี้ เป็นผลที่ได้มาจากภาค Input เปรียบเสมือนว่า ให้ภาค Output แสดงผลเพื่อต้องการให้ผู้ใช้งานได้ทราบถึงค่าต่างๆ ได้อย่างเข้าใจ และถูกต้อง ต่อการใช้งาน ในโครงงานนี้ตัวที่ใช้เป็นภาคแสดงผล คือ LCD (Liquid Crystal Display)

**วงจร**




**รูปที่ 3.6** วงจรแสดงภาคเอาท์พุท

### 3.1.7 ตัวตรวจจับตำแหน่ง

ตัวตรวจจับตำแหน่งเป็นอุปกรณ์ที่สามารถบอกได้ว่า วัตถุอยู่ที่ตำแหน่งใดของสายพาน ซึ่งตัวตรวจจับตำแหน่งนี้มีใช้อยู่ 2 ชนิด คือ อินฟาเรดเซนเซอร์ และตัวเอนโคดเดอร์

ตัวตรวจจับตำแหน่งมีอุปกรณ์ที่สำคัญดังนี้

1. อินฟาเรดเซนเซอร์ทั้งตัวรับและตัวส่ง
2. ค่าความต้านทาน 200 $\Omega$, 500 K$\Omega$ และ 10 K$\Omega$
3. ทรานซิสเตอร์ชนิด NPN

**การทำงาน**

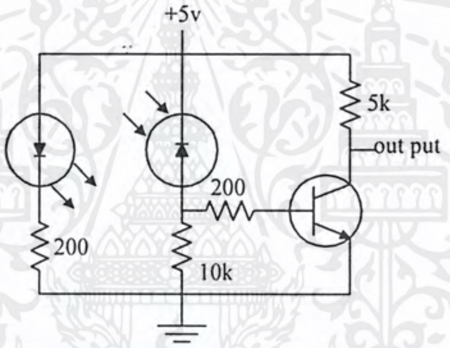
ลักษณะการทำงานของตัวตรวจจับตำแหน่งที่เป็นตัวอินฟาเรดเซนเซอร์ คือ ในขณะที่ไม่มีวัตถุเคลื่อนที่ผ่านลำแสงอินฟาเรด ซึ่งไม่สามารถมองเห็นได้ด้วยตาเปล่า จะไม่มีอะไรเกิดขึ้น กล่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ ลอจิกที่ตัวไมโครคอนโทรลเลอร์ได้รับก็ยังเป็นลอจิก "0" อยู่ แต่เมื่อมีวัตถุผ่านเข้ามากั้นลำแสงอินฟาเรด ก็จะทำให้ไมโครคอนโทรลเลอร์ได้รับลอจิก "1" และสั่งให้มอเตอร์หยุดหมุน ซึ่งจะเป็นผลให้ของที่อยู่บนสายพานลำเลียงไม่ตกหล่นลงไป

เหตุผลที่มีการนำทรานซิสเตอร์มาต่อร่วมในวงจรก็เนื่องมาจากว่า ในขณะที่ไม่ได้ต่อทรานซิสเตอร์ในวงจร จะทำให้แรงดันระหว่างลอจิก "0" และลอจิก "1" มีความแตกต่างกันน้อยมาก เป็นผลทำให้ตัวไมโครคอนโทรลเลอร์ไม่สามารถแยกถึงความแตกต่างของลอจิกนั้นๆ ได้ จึงให้ทรานซิสเตอร์ทำหน้าที่เป็นตัวขยายแรงดันเพื่อให้ตัวไมโครคอนโทรลเลอร์สามารถแยกแยะความแตกต่างระหว่างลอจิก "0" และลอจิก "1" ได้

**วงจร**



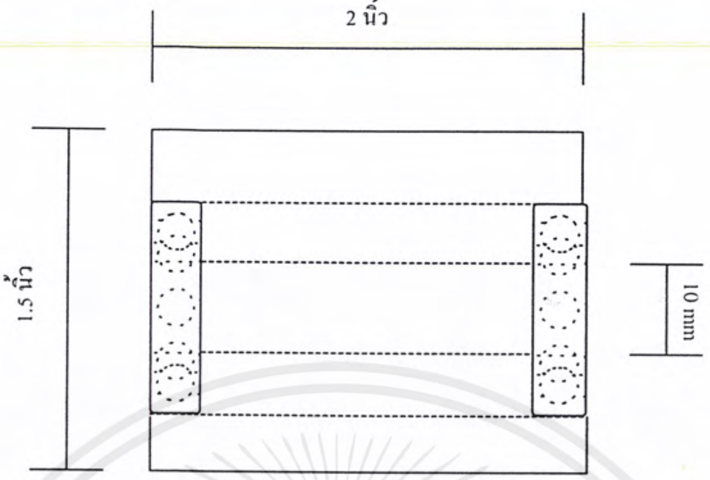

**รูปที่ 3.7** วงจรแสดงตัวตรวจจับตำแหน่ง

## 3.2 การออกแบบอุปกรณ์ทางกล

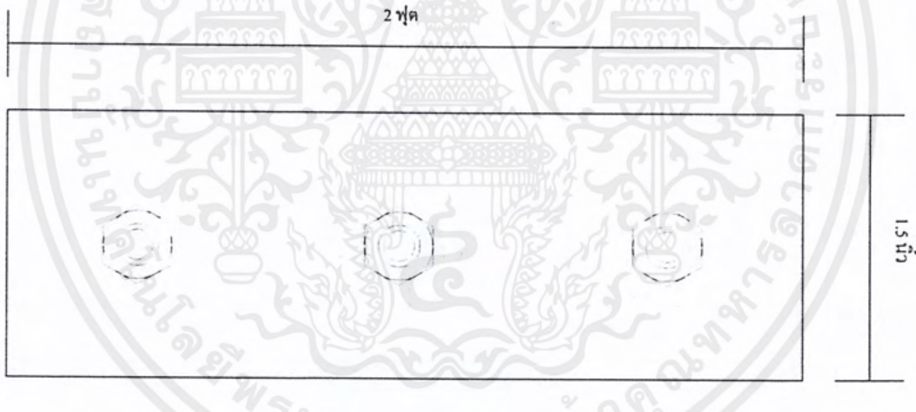
ในการออกแบบโครงงานในส่วนของทางกลนี้ ทางผู้จัดทำได้ศึกษาตัวอย่างโครงงานชิ้นอื่นๆ แล้วนำมาดัดแปลงแก้ไขให้เหมาะสมกับโครงงานของตัวเอง ประกอบกับทางผู้จัดได้ศึกษาจากบุคคลต่างๆ ไม่ว่าจะเป็น ช่างกลึง, ช่างยนต์, ตลอดจนไปถึงอาจารย์ภาควิชาครุศาสตร์วิศวกรรม ซึ่งได้ให้คำแนะนำที่เป็นประโยชน์ต่อโครงงาน

ในการออกแบบคอนเวนเยอร์นั้น ผู้จัดทำได้นำเหล็กพืด (เหล็กแบน) ขนาดความกว้าง 2 นิ้ว และความยาว 2 ฟุต มาทำเป็นขอบำว้สำหรับยึดพูลเลย์และขาตั้ง ส่วนลูกพูลเลย์ได้นำอะลูมิเนียมมากลึงซึ่งมีขนาดตามรูปที่ 3.8 ในส่วนของสายพานได้นำเศษเบาะรถยนต์มาตัดให้ได้ขนาดตามความยาวและความกว้างของพูลเลย์ ในรูปที่ 3.9 จะแสดงถึงการประกอบส่วนต่างๆ เข้าด้วยกันเป็นคอนเวนเยอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.8** แสดงการออกแบบลูกพูลเลย์




**รูปที่ 3.9** แสดงภาพตัดข้างของคอนเวนเยอร์

และส่วนที่สำคัญอีกอย่างในการออกแบบทางกล คือ การเลือกใช้มอเตอร์ ซึ่งมอเตอร์ก็มีอยู่ด้วยกันมากมายหลายชนิด ดังนั้นเราต้องเลือกให้เหมาะสมกับงาน

สำหรับโครงงานนี้ มอเตอร์ที่ใช้ต้องมีตัวเอนโคดเดอร์อยู่ด้วย และเนื่องจากว่าต้องนำมอเตอร์มาใช้ขับสายพานจำเป็นต้องหามอเตอร์ที่มีกำลังแรงขับที่สูง หรืออาจจะหามอเตอร์ที่มีเกียร์บล็อกเพื่อเพิ่มแรงบิดของมอเตอร์

ระบบเกียร์นี้เป็นอุปกรณ์เครื่องกลซึ่งสามารถส่งพลังงานจากส่วนหนึ่งของระบบไปยังส่วนอื่นๆ ได้ในรูปของแรงงาน แรงบิด ความเร็ว และการเคลื่อนที่ นอกจากนั้นอุปกรณ์เหล่านี้ยังเป็นเสมือนอุปกรณ์สำหรับประสานที่สามารถใช้ส่งผ่านกำลังงานให้ได้ค่าสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4

# การทดลองและผลการทดลอง

## 4.1 การทดลองในส่วนของอิเล็กทรอนิกส์

การทดลองในส่วนของอิเล็กทรอนิกส์จะได้มาจากการออกแบบอุปกรณ์ทางอิเล็กทรอนิกส์ในภาคต่างๆ ดังนี้

1. ภาคจ่ายไฟ จากการออกแบบ จะมีการทดลองในภาคนี้ คือ ได้นำแรงดันจ่ายเข้าที่ด้านอินพุท และทำการวัดแรงดันที่ด้านเอาท์พุท ให้ได้ระดับแรงดันตามที่ต้องการ ได้แก่ 5 V.และ 15 V.

2. ภาคระบบควบคุม จากการออกแบบ จะมีการทดลองในภาคนี้ คือ จะทำการป้อนโปรแกรมให้กับไมโครคอนโทรลเลอร์เพื่อทำการทดสอบวงจรอิเล็กทรอนิกส์ในแต่ละส่วนว่าใช้ได้หรือไม่

3. ภาคขับมอเตอร์ จากการออกแบบ จะมีการทดลองในภาคนี้ คือ ทดลองจ่ายไฟจากแหล่งจ่ายเพื่อไปควบคุมการทำงานของอุปกรณ์อิเล็กทรอนิกส์ในภาคขับมอเตอร์ แล้วทำการสลับขั้วเพื่อตรวจสอบดูว่ามอเตอร์สามารถหมุนกลับทิศทางได้หรือไม่

4. ภาคระบบป้อนกลับจากการออกแบบจะมีการทดลองในภาคนี้ คือ ใช้ออสซิลโลสโคปวัดสัญญาณเอาท์พุทของเอนโคดเดอร์เพื่อตรวจสอบดูว่า ตัวเอนโคดเดอร์นั้น สามารถส่งสัญญาณเอาท์พุทออกมาได้จริงหรือไม่ จากนั้นจึงทดลองใช้ไมโครคอนโทรลเลอร์รับค่าสัญญาณดังกล่าวเข้าประมวลผลและส่งสัญญาณไปยังหน่วยแสดงผลต่อไป

5. ภาคอินพุท จากการออกแบบ จะมีการทดลองในภาคนี้ คือ จะใช้ไมโครคอนโทรลเลอร์รับสัญญาณจากแมททริกสวิทช์ เข้ามาทำการประมวลผลและแสดงผลค่าที่ได้ออกทางจอ LCD

6. ภาคเอาท์พุท จากการออกแบบ จะมีการทดลองในภาคนี้ คือ ใช้ไมโครคอนโทรลเลอร์ส่งสัญญาณเพื่อทดสอบการแสดงผลของจอ LCD และวงจรขับมอเตอร์ว่าทำงานถูกต้องหรือไม่

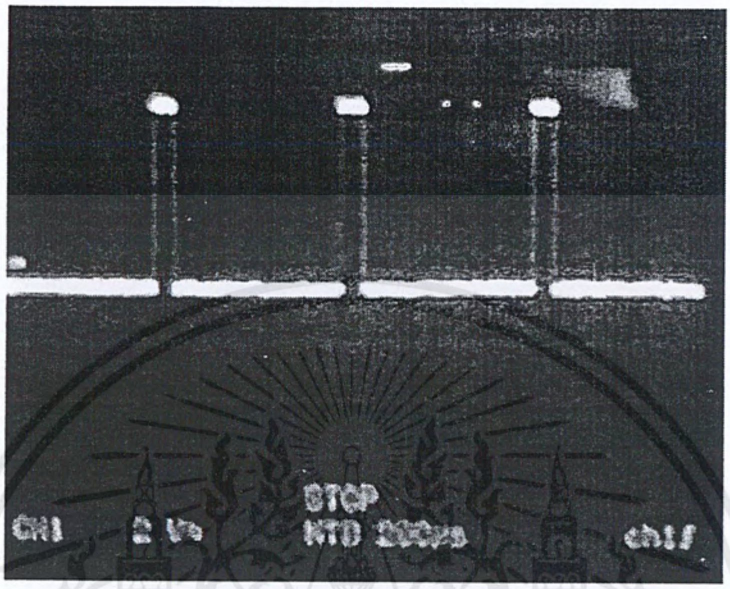
7. ตัวตรวจจับตำแหน่ง จากการออกแบบ จะมีการทดลองในภาคนี้ คือ ออกแบบวงจรจัดไบอัสให้กับอินฟาเรดเซนเซอร์ให้เหมาะสมกับวงจร เพื่อให้ได้เป็นลอจิก 0 และลอจิก 1 ที่สมบูรณ์

## 4.2 การทดลองในส่วนของโปรแกรม

ใช้โปรแกรมควบคุมให้ไมโครคอนโทรลเลอร์สร้างสัญญาณรูปคลื่นสี่เหลี่ยม เพื่อที่จะใช้ทำการปรับความเร็วของมอเตอร์ซึ่งมีเปอร์เซนต์ดิวตี้ไซเคิลตั้งแต่ 10 – 100 เปอร์เซนต์

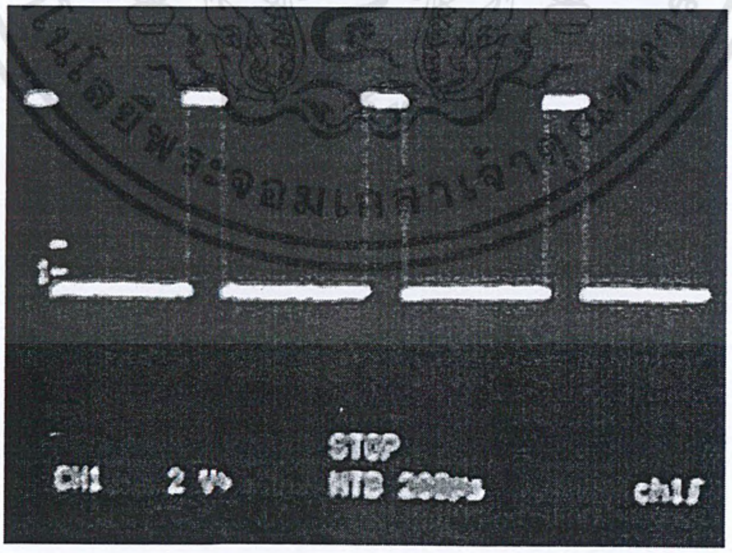
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ที่ 10 เปอร์เซนต์**




**รูปที่ 4.1** แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 10 เปอร์เซนต์

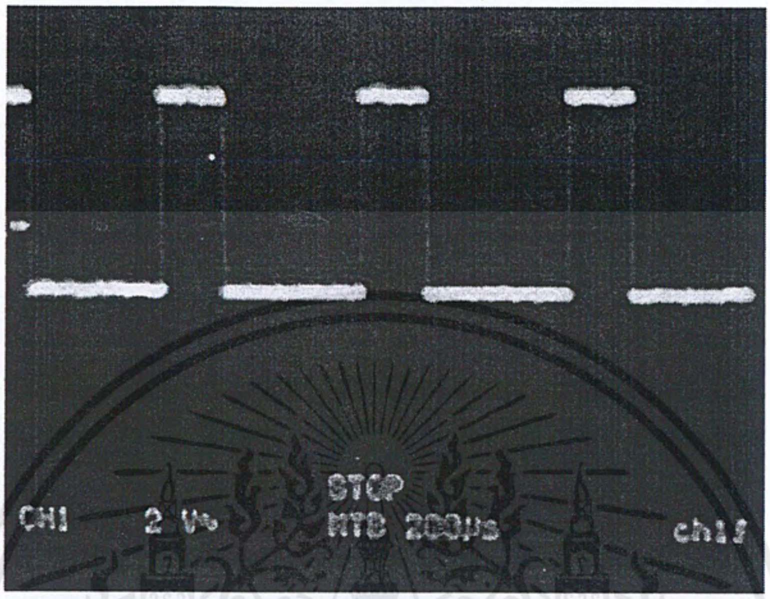
**ที่ 20 เปอร์เซนต์**




**รูปที่ 4.2** แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 20 เปอร์เซนต์

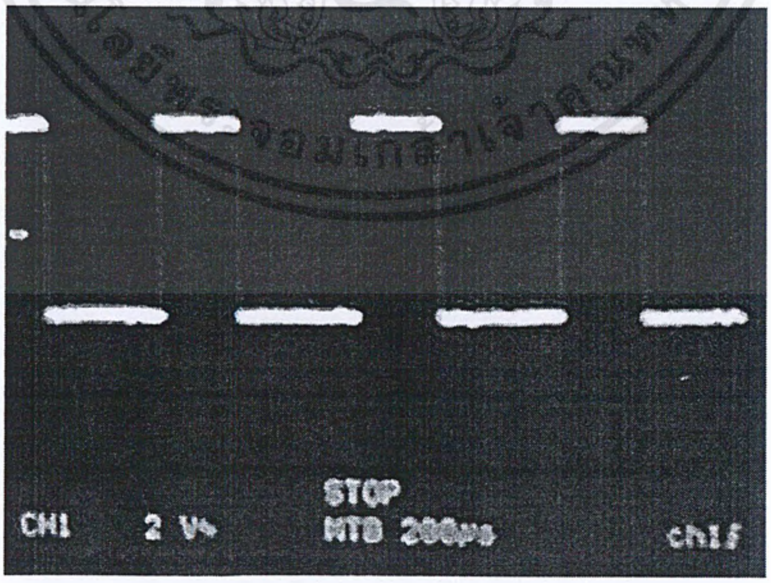
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 30 เปอร์เซนต์




รูปที่ 4.3 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 30 เปอร์เซนต์

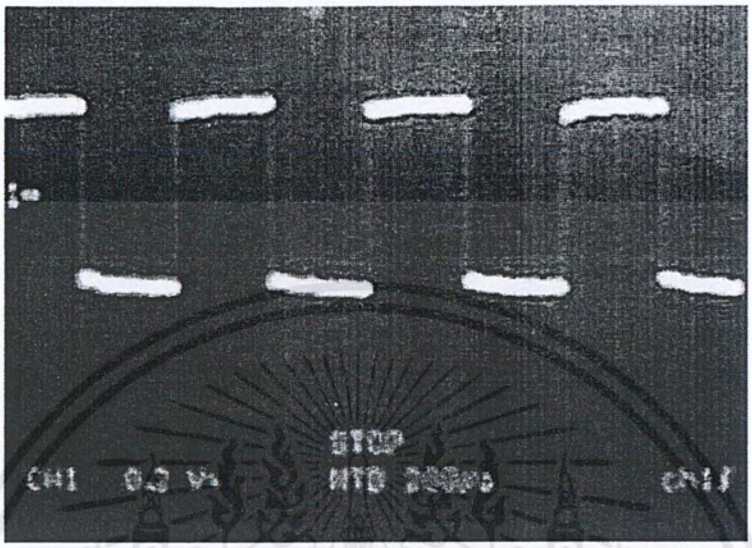
ที่ 40 เปอร์เซนต์




รูปที่ 4.4 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 40 เปอร์เซนต์

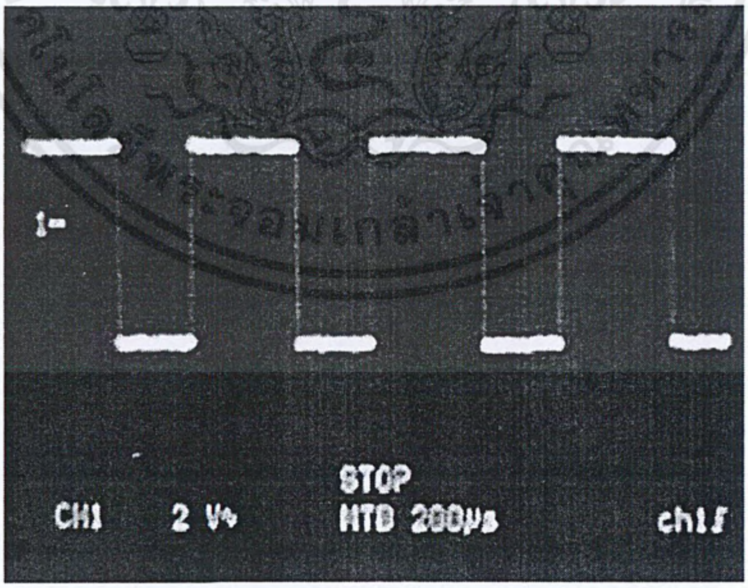
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 50 เปอร์เซนต์




รูปที่ 4.5 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 50 เปอร์เซนต์

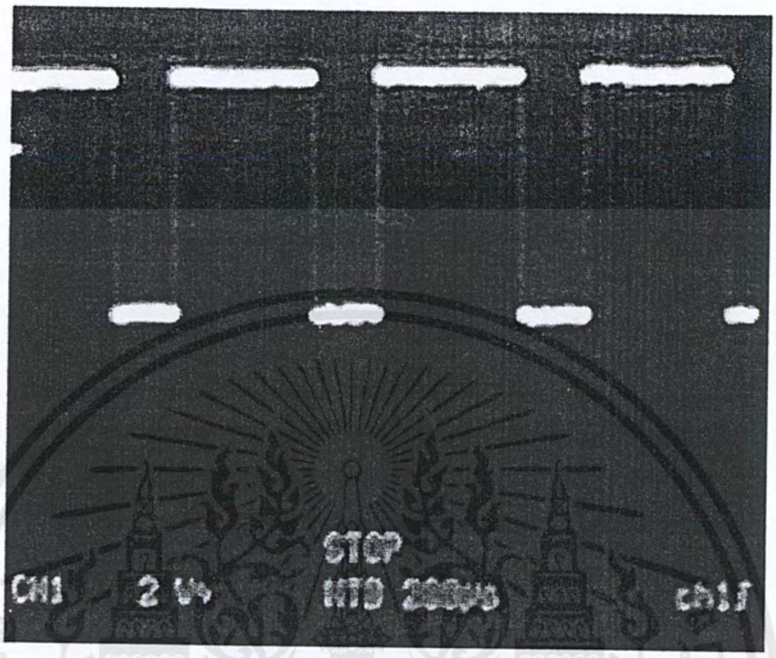
ที่ 60 เปอร์เซนต์




รูปที่ 4.6 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 60 เปอร์เซนต์

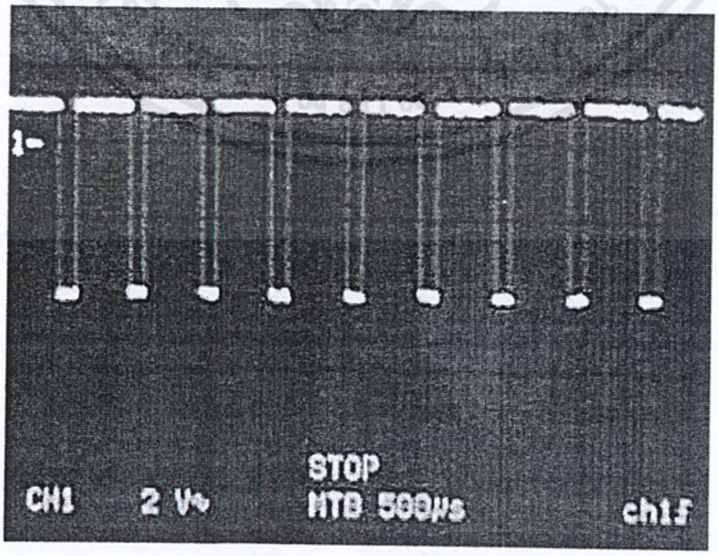
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 70 เปอร์เซนต์




รูปที่ 4.7 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 70 เปอร์เซนต์

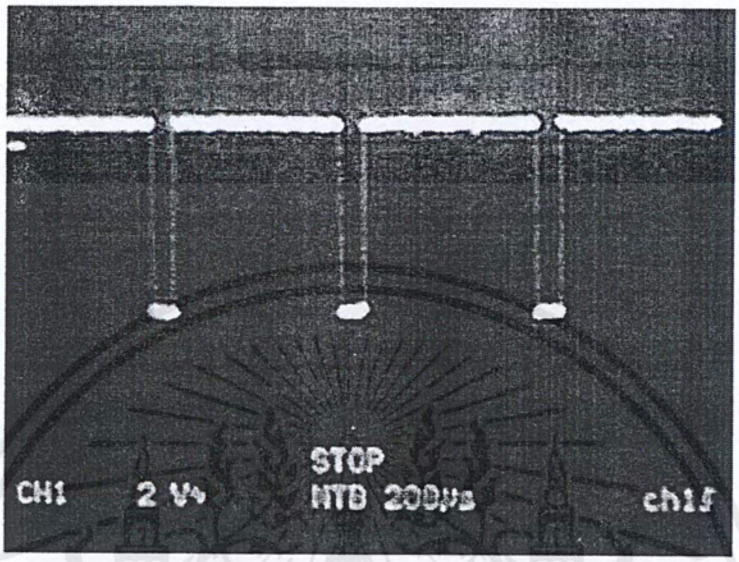
ที่ 80 เปอร์เซนต์




รูปที่ 4.8 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 80 เปอร์เซนต์

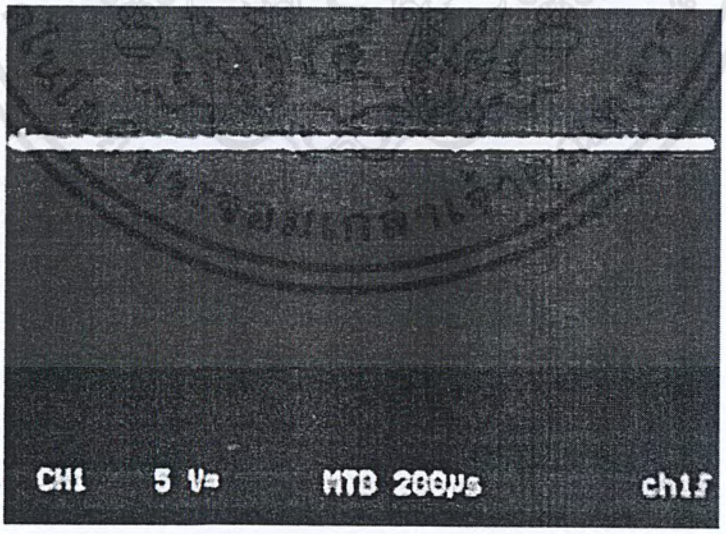
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 90 เปอร์เซนต์




รูปที่ 4.9 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 90 เปอร์เซนต์

ที่ 100 เปอร์เซนต์




รูปที่ 4.10 แสดงเปอร์เซนต์ดิวตี้ไซเคิลที่ 100 เปอร์เซนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่5

# บทสรุป ปัญหา แนวทางแก้ไขและพัฒนา

## 5.1 บทสรุป

โครงงานเรื่องนี้เป็นเรื่องเกี่ยวกับการควบคุมมอเตอร์ไฟฟ้ากระแสตรงโดยใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 โดยจุดประสงค์หลัก คือ การควบคุมควบเร็วและตำแหน่งของมอเตอร์โดย แสดงการทำงานให้เห็นในรูปของสายพานลำเลียงซึ่งสามารถที่จะปรับความเร็วของในการเคลื่อนที่และหยุดในตำแหน่งต่างๆ ของสายพานได้ทุกจุด

การควบคุมความเร็วจะอาศัยหลักการของความถี่เข้ามาช่วย โดยการที่จะให้มอเตอร์หมุนเร็วหรือช้านั้นจะขึ้นอยู่กับการปรับความกว้างของสัญญาณรูปคลื่นสี่เหลี่ยมที่ป้อนเข้ามา ถ้ารูปคลื่นสี่เหลี่ยมที่ป้อนเข้ามานั้นมีความกว้างในช่วงเวลาเปิด (T-on) กว้างกว่าเวลาปิด (T-off) ใน1รอบการทำงาน (1 Cycle) จะทำให้แรงดันตกคร่อมมอเตอร์มีค่ามาทำให้มอเตอร์หมุนได้เร็วแต่ถ้าหากช่วงเวลาในการเปิดน้อยกว่าช่วงเวลาในการปิดในหนึ่งรอบการทำงาน จะมีผลให้แรงดันที่ตกคร่อมมอเตอร์มีค่าน้อยส่งผลให้มอเตอร์หมุนช้า ด้วยหลักการข้างต้นจึงสามารถที่จะปรับควบคุมความเร็วของมอเตอร์ได้

ส่วนของการควบคุมตำแหน่งการเคลื่อนที่ของสายพานนั้นจะอาศัยอุปกรณ์ตรวจจับที่เรียกว่าตัวเข้ารหัสเป็นอุปกรณ์หลัก โดยตัวเข้ารหัสนี้จะให้สัญญาณรูปคลื่นสี่เหลี่ยมออกมาเมื่อเกิดการหมุนขึ้นโดยการหมุนครบหนึ่งรอบจะมีสัญญาณรุปคลื่นสี่เหลี่ยมเกิดขึ้นเป็นจำนวนที่แน่นอน ดังนั้นเมื่อมอเตอร์หมุนทำหน้าที่ลากสายพานไปแล้วนั้นจะทำให้สามารถคำนวนค่าของรูปคลื่นสี่เหลี่ยมที่จะออกมาได้ว่าตั้งแต่ตำแหน่งเริ่มต้นไปจะสิ้นสุดตำแหน่งสุดท้ายของสายพานนั้น มีสัญญาณออกจากตัวเข้ารหัสจำนวนเท่าใด เมื่อเป็นเช่นนั้นก็จะสามารถทราบตำแหน่งของวัตถุที่วางอยู่บนสายพานได้ โดยการเทียบอัตราส่วนของการหมุนและสัญญาณรูปคลื่นสี่เหลี่ยมที่ตัวเข้ารหัสส่งออกมานั่นเอง

## 5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่เกิดขึ้นระหว่างการทำโครงงานนี้อาจแบ่งเป็นส่วนๆ ได้ดังนี้คือ

### 5.2.1ส่วนของอุปกรณ์อิเล็คทรอนิกส์

1. การออกแบบในส่วนของวงจรอิเล็กทรอนิกส์กำลัง ซึ่งจะเป็นวงจรที่ใช้ในการขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ไฟฟ้ากระแสตรงเนื่องจากวิธีการที่จะใช้ควบคุมมีหลายรูปแบบ อีกทั้งกลุ่มของข้าพเจ้าไม่มีความรู้เบื้องต้นเพียงพอในการออกแบบจึงเกิดปัญหาดังกล่าวขึ้น

วิธีการแก้ไขปัญหาในข้อ 1 คือ จะต้องมีการค้นคว้าหาข้อมูลเพิ่มเติมจากตำราทางวิชาการต่างๆ เพื่อให้ได้มาซึ่งวิธีการที่เหมาะสม จากนั้นจึงนำข้อมูลที่ได้นั้นไปปรึกษาอาจารย์ที่ปรึกษาโครงงานจึงได้ข้อสรุปออกมาว่าควรใช้การควบคุมแบบ Half Bridge Transistor

2. การออกแบบในส่วนของการเชื่อมต่อระหว่างวงจรภาคอิเล็กทรอนิกส์กำลังที่ใช้ในการควบคุมมอเตอร์เข้ากับอุปกรณ์ไมโครคอนโทรลเลอร์เนื่องจากกระแสที่ต้องใช้ควบคุมนั้น ต้องใช้กระแสที่สูงมากในการกระตุ้นให้ทรานซิสเตอร์กำลังทำงาน ดังนั้นจึงไม่สามารถนำเอาไมโครคอนโทรลเลอร์เข้ามาควบคุมได้โดยตรง และการเชื่อมต่อนั้นจะต้องไม่ให้กระแสส่วนใหญ่ซึ่งมีขนาดมากนั้นไหลเข้ามายังไมโครคอนโทรลเลอร์ได้ เนื่องจากจะทำให้เกิดความเสียหายกับอุปกรณ์ควบคุม

วิธีการแก้ไขปัญหาในข้อ 2 จึงต้องมีการนำเอาสัญญาณที่ไมโครคอนโทรลเลอร์ ส่งมานั้นไปทำการขยายกระแสให้สูงขึ้นก่อนที่จะนำไปใช้ควบคุมการกระตุ้นของตัวอุปกรณ์อิเล็กทรอนิกส์กำลังอีกทีหนึ่งและวิธีการที่จะไม่ให้กระแสส่วนใหญ่ไหลเข้ามายังตัวของไมโครคอนโทรลเลอร์ด้วยนั้นก็สามารถกระทำได้โดยใช้อุปกรณ์เชื่อมต่อทางแสงหรือที่เรียกว่าทรานซิสเตอร์ที่ทำงานด้วยแสง (Opto-Transistor) เข้ามาเป็นตัวขยายกระแสและให้แสงเป็นตัวขวางกันไม่ให้กระแสส่วนใหญ่ไหลเข้ามาในส่วนของตัวควบคุมได้

3. ปัญหาในการออกแบบลายวงจร เนื่องจากเริ่มแรกทีเดียวนั้นทางกลุ่มของข้าพเจ้าได้ใช้แผ่นวงจรอเนกประสงค์ จึงทำให้เกิดปัญหาเล็กน้อยตามมาเป็นอันมากเนื่องจากสายที่ใช้ต่อวงจรนั้นไม่แข็งแรงและมักจะทำให้เกิดการลัดวงจรอยู่เสมอ

วิธีการแก้ไขปัญหาในข้อ 3 คือ เปลี่ยนเป็นการใช้แผ่นวงจรพิมพ์ที่ทำขึ้นเอง ซึ่งสามารถแก้ไขความไม่แข็งแรงของการต่อสายในวงจรได้ดีมาก

### 5.2.2 การประกอบวงจรและทดสอบการทำงานจริง

ในส่วนนี้มีปัญหาสำคัญที่เกิดขึ้นคือในส่วนของทรานซิสเตอร์กำลังนั้นไม่สามารถใช้ทรานซิสเตอร์เบอร์ที่ออกแบบไว้ในตอนแรกคือเบอร์ H1061 และ เบอร์ B507 ได้เนื่องมาจากไม่สามารถทนกระแสในสภาวะไบอัสกลับได้ และอีกทั้งยังมีกระแสรั่วไหล ที่ขาคอลเล็กเตอร์และขาอิมิตเตอร์อีกด้วย ทำให้มอเตอร์ เริ่มหมุนในขณะที่ ไมโครคอนโทรเลอร์ยังไม่สั่งงาน

วิธีการแก้ไขปัญหาในส่วนนี้คือ เปลี่ยนเบอร์ของตัวทรานซิสเตอร์กำลังไปเป็นเบอร์ 2N3055 และ เบอร์ 2N2955 แทนเนื่องจาก เป็นเบอร์ที่มีพิกัดการทนกระแสและอัตราทนแรงดันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มากกว่า อีกทั้งยังทีกระแสรั่วไหลที่ขาคอลเล็กเตอร์และขาอิมิตเตอร์น้อยมาก ทำให้สามารถควบคุมการทำงานได้ดีขึ้นมาก

### 5.2.3 การออกแบบทางกล

ในส่วนนี้มีปัญหาเกิดขึ้นในส่วนของระบบสายพานลำเลียงอันเนื่องมาจากสายพานลำเลียงไม่สามารถหมุนได้อย่างอิสระเพราะว่าเกิดความฝืดขึ้นระหว่าง แกนพูลเลย์กับรางเหล็กที่ใช้สำหรับจับยึด สายพานไว้ดังนั้นจึงส่งผลให้สายพานไม่สามารถหมุนได้ อีกทั้งยังส่งผลให้มอเตอร์ไม่สามารถขับสายพานได้ด้วยเนื่องจากแรงบิดไม่เพียงพอที่จะหมุนให้สายพานเคลื่อนที่ไปได้

วิธีการแก้ปัญหาคือ ใช้ตลับลูกปืนเพื่อลดความฝืดระหว่างรอยต่อของ พูลเลย์กับขอบรางโดยการฝังตลับลูกปืนเข้าไปในแกนพูลเลย์ และ ฝังไว้ที่ของรางเพื่อลงความเสียดทานระหว่างแกนหมุนกับขอบรางของสายพานด้วย ส่งผลให้สายพานสามารถหมุนได้อย่าอิสระ

### 5.2.4 แรงบิดของมอเตอร์

เนื่องจากมอเตอร์ไฟฟ้ากระแสตรงมีแรงบิดที่ต่ำจึงไม่สามารถที่จะฉุดลากสายพานให้หมุนได้ ถึงแม้ว่าจะสามารถลดความฝืดได้อย่างมากแล้วก็ตามก็ยังไม่สามารถที่จะขับให้สายพานเคลื่อนที่ไปได้อีกทั้งยังส่งผลไปยังทรานซิสเตอร์กำลัง เพราะว่าเมื่อมอเตอร์ไม่หมุนจะทำให้เกิดการดึงการแสสูงมากทำให้ทรานซิสเตอร์กำลังเสียหายอีกด้วย

วิธีการแก้ปัญหาคือ จะต้องใช้มอเตอร์ที่มีเกียร์ ซึ่งมีแรงบิดสูงแต่ความเร็วรอบต่ำเข้ามาช่วยซึ่งผลปรากฏว่ามอเตอร์สามารถขับไปได้ทั้งในขณะที่ไม่มีโหลดและมีโหลด

## 5.3 แนวทางในการพัฒนา

ในส่วนของการพัฒนาเพิ่มเติมนั้น จะเป็นไปได้ว่า การโปรแกรมการทำงานเข้าไปยังตัวของไมโครคอนโทรเลอร์นั้นจะต้องใช้เครื่องโปรแกรม EPROM อาจไม่สะดวกนักถ้าหากมีการแก้-ไขโปรแกรมบ่อยๆ การพัฒนาตรงจุดนี้คือจะต้องเพิ่มส่วนของ Hardware ให้สามารถโปรแกรมผ่านทาง Serial Port หรือ Parallel Port ได้เพื่อความสะดวกในการโปรแกรมถ้าหากมีความต้องการตั้งโปรแกรมการทำงานใหม่อยู่บ่อยๆ

อีกส่วนหนึ่งที่มีความเป็นไปได้ก็คือการนำเอาอุปกรณ์อื่นๆ เข้ามาใช้งานร่วมด้วยเช่นระบบนิวแมติกส์ มาจับชิ้นงานที่เลื่อนมาบนสายพานหรือระบบตรวจจับข้อผิดพลาดของชิ้นงานที่วิ่งมาบนสายพานลำเลียงเช่นไม่ได้ปิดฝาหรือไม่ได้ปิดฉลากข้างขวดเป็นต้น ก็จะใช้ในการศึกษาการทำงานของระบบโรงงานอัตโนมัติหรือที่เรียนกันว่าระบบโรงงานอัตโนมัติได้ในระดับหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก

# โปรแกรม Pulse Width Modulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# โปรแกรม Pulse Width Modulation

```
;********************** LCD define **********************
LCDRS       BIT     P2.2    ;ON BOARD LCD (CONTROL BIT)
LCDRW       BIT     P2.1
LCDEN       BIT     P2.0
F_LED       BIT     p2.3
R_LED       BIT     p2.4

;********************** infarade define*********************
infa1       BIT     p2.6    ; start point
infa2       BIT     p2.7    ; end point

;**********************keypad define*********************
NKEY        EQU     7FH
memkey      EQU     7bh         ;ค่าการหมุน (ซ้าย หรือ ขวา)
memkey1     EQU     7ah         ;ค่าของduty

;**********************delay define*********************
dtime0      EQU     7eh
dtime1      EQU     7dh
dtime2      EQU     7ch

;********************Encoder Define*********************
ENCDATA     EQU     6AH         ; 6AH->6DH
ENCPOINT    EQU     69H         ; POINTER ENCODER DATA
ENCNEW      EQU     68H         ; INPUT NEW ENCODER DATA
ABSPOST     EQU     45H         ; ABSOLUTE POSITION (3 BYTE)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            org     0000h
            ljmp    boot
            ORG     0013H
            LJMP    ENCODER             ; READ ENCODER ADDRESS


boot:       mov     r0,#0ffh
b1:         mov     r1,#0ffh
            djnz    r1,$
            djnz    r0,b1
            ljmp    main

main:
            clr     LCDRS               ; init LCD
            clr     LCDRW
            clr     LCDEN
            lcall   init
start:      mov     a,#80h                      ;* Position
start1:     lcall   msg1                        ;* Outtext แสดงผล press 2nd to select
            lcall   delay
start2:     lcall   main_menu
            sjmp    start

main_menu:  lcall   scankey
            lcall   delay
            cjne    a,#0bh,main_menu

speedmenu:  mov     a,#0c0h             ;speed menu
            lcall   msg2                        ;text speed press enter
            lcall   delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
speedmenu1:    lcall   scankey
               lcall   delay
               cjne    a,#0fh,chkrept          ;ไปเช็คว่ามีการกด enterหรือไม่
               ljmp    turn                    ;เรียกโปรแกรมย่อยเลือกเลือกทิศทาง

chkrept:       lcall   scankey                 ;เช็คว่ามีการกด 2nd หรือไม่
               lcall   delay
               cjne    a,#0bh,speedmenu1

positionmenu:  mov     a,#0c0h                 ;position of text
               lcall   msg3                    ;text position press enter
               lcall   delay
post:          lcall   scankey
               lcall   delay
               cjne    a,#0fh,chk2nd           ;โดดเช็คว่ามีการกด 2nd ซ้ำหรือไม่
               ljmp    selposition             ;กระโดดไปโปรแกรมเลือกตำแหน่ง

chk2nd:        lcall   scankey
               lcall   delay
               cjne    a,#0bh,post             ;กด 2nd ซ้ำหรือไม่
               ljmp    start2

;************* โปรแกรมเลือกduty cycle *******************
turn:
               mov     a,#0c0h
               lcall   msg4                    ;text 'press->,<-to select
               lcall   delay
;****************************************************
way1:          lcall   scankey                 ;ถ้าไม่กด ent ก็จะโดดมาที่นี่
               lcall   delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
up:     cjne    a,#03h,down         ;key -> check
        mov     memkey,a            ;จำค่าไว้
        mov     a,#0c0h
        lcall   msg5                ;'forward press enter '
        lcall   delay
        lcall   scankey
        cjne    a,#0fh,clear        ;ถ้าไม่กดenter
        lcall   selduty             ;เลือก duty cycle
        ljmp    load                ;กระโดดไปเพื่อทำงาน

down:   lcall   scankey
        lcall   delay
        cjne    a,#07h,ent
        mov     memkey,a            ;จำค่าไว้
        mov     a,#0c0h
        lcall   msg6                ;'reverse press enter '
ent:    lcall   scankey
        cjne    a,#0fh,clear        ;ถ้าไม่กดenter
        lcall   selduty             ;เลือก duty cycle
        ljmp    load                ;กระโดดไปเพื่อทำงาน
clear:
        lcall   scankey
        lcall   delay
        cjne    a,#0ch,way1
        mov     a,#0c0h
        lcall   msg0
        ljmp    start               ;กลับไป main menu
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
load:
        mov     a,memkey            ;load มาดู
        cjne    a,#03h,rig
        ljmp    run                 ;กระโดดไปทำงานการหมุนซ้าย
rig:
        cjne    a,#07h,turn
        ljmp    run                 ;กระโดดไปทำงานด้านขวา

run:
        mov     a,#0c0h
        lcall   msg7                ;'press CLEAR to STOP'
        mov     a,memkey
left:   cjne    a,#03h,right
        clr     f_led               ;หมุนซ้าย 100%
        lcall   delay               ;go to long delay
        lcall   delay
        ljmp    chk_du

right:  clr     r_led               ;หมุนซ้าย100%
        lcall   delay
        lcall   delay
        ljmp    chk_du


;******************* load dutycycle value from memory *******************
chk_du:
        mov     a,memkey1           ;เช็ค duty cycle ที่ถูกเลือก
du0:    cjne    a,#0dh,du1          ;เปรียนเทียบค่าที่ถูกเลือก
        ljmp    duty100             ;กระโดดไปทำงานที่100%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
du1:
        cjne    a,#00h,du2
        ljmp    duty10              ;ไปทำงานที่10%

du2:
        cjne    a,#01h,du3
        ljmp    duty20              ;ไปทำงานที่20%

du3:
        cjne    a,#02h,du4
        ljmp    duty30              ;ไปทำงานที่30%

du4:
        cjne    a,#04h,du5
        ljmp    duty40              ;ไปทำงานที่40%

du5:
        cjne    a,#05h,du6
        ljmp    duty50              ;ไปทำงานที่50%

du6:
        cjne    a,#06h,du7
        ljmp    duty60              ;ไปทำงานที่60%

du7:
        cjne    a,#08h,du8
        ljmp    duty70              ;ไปทำงานที่70%
du8:
        cjne    a,#09h,du9
        ljmp    duty80              ;ไปทำงานที่80%
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
du9:
                cjne    a,#0ah,tu
                ljmp    duty90              ;ไปทำงานที่90%
tu:             ljmp    turn
selposition:
                lcall   scankey
                lcall   delay
                mov     a,#0c0h
                lcall   msg7
selposit:       lcall   scankey
                lcall   delay
                cjne    a,#0ch,selposit
                mov     a,#0c0h
                lcall   msg0
                ljmp    start

;* LCD Initial

init:           mov     a,#00111000b        ;* Function set
                lcall   lcdwi
                mov     a,#00111000b        ;* Function set
                lcall   lcdwi
                mov     a,#00111000b        ;* Function set
                lcall   lcdwi
                mov     a,#00111000b        ;* Function set
                lcall   lcdwi
                mov     a,#00001100b        ;* Disolay on/off
                lcall   lcdwi
                mov     a,#1                ;* Clear
                lcall   lcdwi
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     a,#080h         ; Position
        lcall   lcdwi
        ret

;* LCD write data
;* in : a

lcdwd:  mov     p0,a
        setb    LCDRS
        clr     LCDRW           ;R/W=0
        setb    LCDEN           ;ENABLE=0 (INVERT)
        nop
        clr     LCDEN           ;ENABLE=1
        nop
        setb    LCDEN           ;ENABLE=0
        nop
        mov     a,#0
lcdwd1: dec     a
        jnz     lcdwd1
        ret

;* LCD write command
;* in : a

lcdwi:  mov     p0,a
        clr     LCDRS
        clr     LCDRW           ;R/W=0
        setb    LCDEN           ;ENABLE=0 (INVERT)
        nop
        clr     LCDEN           ;ENABLE=1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                nop
                setb    LCDEN               ;ENABLE=0
                nop
                mov     a,#0
lcdwi1:         dec     a
                jnz     lcdwi1
                ret

;* Outtext function

text:           clr     a
                movc    a,@a+dptr
                jz      text1
                lcall   lcdwd
                inc     dptr
                sjmp    text
text1:          ret

;* Delay function

delay:          push    7
                push    6
                mov     r7,#150
d1:             mov     r6,#0
d2:             djnz    r6,d2
                djnz    r7,d1
                pop     6
                pop     7
                ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;***************************************************
;                         %duty cycle              *
;***************************************************

;******* duty cycle = 10% *************************
;******* t - on=10 %  *****************************

duty10:
        mov     a,memkey
mk10:   cjne    a,#03h,mk11         ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
        ljmp    fo

mk11:   clr     r_led               ;ให้หมุนขวา
        ljmp    l10

fo:     clr     f_led               ;ให้หมุนซ้าย
        ljmp    l10

l10:    mov     tmod,#01h           ;timer 0 mode1
loop10: mov     th0,#0ffh           ;counting value
        mov     tl0,#0ceh
        setb    tr0                 ;starting timer

wait10: jnb     tf0,wait10          ;wait for overflow
        clr     tr0                 ;stop timer 0
        clr     tf0                 ;clear overflow flag

        mov     a,memkey
mkc10:  cjne    a,#03h,mkc11
        cpl     f_led
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ljmp    duty10f

mkc11:
            cpl     r_led
            ljmp    duty10f
;*********** t - off=90 %  ******************************
duty10f:
            mov     tmod,#01h           ;timer 0 mode1
loop10f:    mov     th0,#0feh           ;counting value
            mov     tl0,#3eh
            setb    tr0                 ;starting timer

            mov     a,#0c0h
            lcall   msg19

wait10f:    jnb     tf0,wait10f         ;wait for overflow
            clr     tr0                 ;stop timer 0
            clr     tf0                 ;clear overflow flag

            lcall   updatedisp          ;เรียกโปรแกรม update display
            jb      infa1,st1           ;stop by infarate
            jb      infa2,st1
            lcall   scankey             ;checkey
            lcall   delay
            cjne    a,#0ch,loop10

st1:        ljmp    stop

;********** end of duty 10 % ***************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********** duty cycle = 20 % *********************** **
;******* t - on=20 %  *********************************

duty20:
          mov    a,memkey
mk20:     cjne   a,#03h,mk22          ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
          ljmp   fp

mk22:     clr    r_led                ;ให้หมุนขวา
          ljmp   l20

fp:       clr    f_led                ;ให้หมุนซ้าย
          ljmp   l20

l20:
          mov    tmod,#01h            ;timer 0 mode1
loop20:   mov    th0,#0ffh            ;counting value
          mov    tl0,#9ch
          setb   tr0                  ;starting timer

wait20:   jnb    tf0,wait20           ;wait for overflow
          clr    tr0                  ;stop timer 0
          clr    tf0                  ;clear overflow flag

          mov    a,memkey
mkc20:    cjne   a,#03h,mkc22
          cpl    f_led
          ljmp   duty20f
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mkc22:
                cpl     r_led
                ljmp    duty20f

;*********** t - off=80 %  ******************************
duty20f:
                mov     tmod,#01h               ;timer 0 mode1
loop20f:        mov     th0,#0feh               ;counting value
                mov     tl0,#70h
                setb    tr0                     ;starting timer
                mov     a,#0c0h
                lcall   msg19

wait20f:        jnb     tf0,wait20f             ;wait for overflow
                clr     tr0                     ;stop timer 0
                clr     tf0                     ;clear overflow flag
                lcall   updatedisp              ;เรียกโปรแกรม update display
                jb      infa1,st2               ;stop by infarate
                jb      infa2,st2
                lcall   scankey                 ;checkey
                lcall   delay
                cjne    a,#0ch,loop20

st2:            ljmp    stop

;*********** end of duty 20 % ******************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********** duty cycle = 30 % **************************
;******* t - on=30 %  ***********************************

duty30:

            mov     a,memkey
mk30:       cjne    a,#03h,mk33         ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
            ljmp    f1

mk33:       clr     r_led               ;ให้หมุนขวา
            ljmp    l30

f1:         clr     f_led               ;ให้หมุนซ้าย
            ljmp    l30

l30:
            mov     tmod,#01h           ;timer 0 mode1
loop30:     mov     th0,#0ffh           ;counting value
            mov     tl0,#6ah
            setb    tr0                 ;starting timer
wait30:     jnb     tf0,wait30          ;wait for overflow
            clr     tr0                 ;stop timer 0
            clr     tf0                 ;clear overflow flag

            mov     a,memkey
mkc30:      cjne    a,#03h,mkc33
            cpl     f_led
            ljmp    duty30f
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mkc33:
                cpl     r_led
                ljmp    duty30f



;*********** t - off=70 %  *****************************
duty30f:
                mov     tmod,#01h           ;timer 0 mode1
loop30f:        mov     th0,#0feh           ;counting value
                mov     tl0,#0a2h
                setb    tr0                 ;starting timer
                mov     a,#0c0h
                lcall   msg19

wait30f:        jnb     tf0,wait30f         ;wait for overflow
                clr     tr0                 ;stop timer 0
                clr     tf0                 ;clear overflow flag
                lcall   updatedisp          ;เรียกโปรแกรม update display
                jb      infa1,st3           ;stop by infarate
                jb      infa2,st3
                lcall   scankey             ;checkey
                lcall   delay
                cjne    a,#0ch,loop30

st3:            ljmp    stop

;********** end of duty 30 % **************************



;********** duty cycle = 40 % *************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******* t - on=40 %  **************

duty40:

            mov     a,memkey
mk40:       cjne    a,#03h,mk44          ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
            ljmp    f2

mk44:       clr     r_led                ;ให้หมุนขวา
            ljmp    l40

f2:         clr     f_led                ;ให้หมุนซ้าย
            ljmp    l40

l40:

            mov     tmod,#01h            ;timer 0 mode1
loop40:     mov     th0,#0ffh            ;counting value
            mov     tl0,#38h
            setb    tr0                  ;starting timer

wait40:     jnb     tf0,wait40           ;wait for overflow
            clr     tr0                  ;stop timer 0
            clr     tf0                  ;clear overflow flag

            mov     a,memkey
mkc40:      cjne    a,#03h,mkc44
            cpl     f_led
            ljmp    duty40f
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mkc44:
                cpl     r_led
                ljmp    duty40f


;*********** t - off=60 %  **********
duty40f:
                mov     tmod,#01h           ;timer 0 mode1
loop40f:        mov     th0,#0feh           ;counting value
                mov     tl0,#0d4h
                setb    tr0                 ;starting timer

                mov     a,#0c0h
                lcall   msg19

wait40f:        jnb     tf0,wait40f         ;wait for overflow
                clr     tr0                 ;stop timer 0
                clr     tf0                 ;clear overflow flag
                lcall   updatedisp          ;เรียกโปรแกรม update display
                jb      infa1,st4           ;stop by infarate
                jb      infa2,st4
                lcall   scankey             ;checkey
                lcall   delay
                cjne    a,#0ch,loop40

st4:            ljmp    stop

;********** end of duty 40 % **************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********** dyty cycle = 50 % **************************
;******* t - on=50 %  *************

duty50:

        mov     a,memkey
mk50:   cjne    a,#03h,mk55         ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
        ljmp    f3

mk55:   clr     r_led               ;ให้หมุนขวา
        ljmp    l50

f3:     clr     f_led               ;ให้หมุนซ้าย
        ljmp    l50

l50:

        mov     tmod,#01h           ;timer 0 mode1
loop50: mov     th0,#0ffh           ;counting value
        mov     tl0,#06h
        setb    tr0                 ;starting timer

wait50: jnb     tf0,wait50          ;wait for overflow
        clr     tr0                 ;stop timer 0
        clr     tf0                 ;clear overflow flag

        mov     a,#0c0h
        lcall   msg19

        mov     a,memkey
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mkc50:      cjne   a,#03h,mkc55
            cpl    f_led
            ljmp   duty50f


mkc55:
            cpl    r_led
            ljmp   duty50f



;************** stop & check infarate *********************
duty50f:    lcall  updatedisp        ;เรียกโปรแกรม update display
            jb     infa1,st5         ;stop by infarate
            jb     infa2,st5
            lcall  scankey           ;checkey
            lcall  delay
            cjne   a,#0ch,loop50

st5:        ljmp   stop
;*********** end of duty 50 % ***************************


;*********** duty cycle = 60 % *************************
;******* t - on=60 %  **************


duty60:


            mov    a,memkey
mk60:       cjne   a,#03h,mk66       ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
            ljmp   f4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mk66:       clr     r_led           ;ให้หมุนขวา
            ljmp    l60

f4:         clr     f_led           ;ให้หมุนซ้าย
            ljmp    l60

l60:
            mov     tmod,#01h       ;timer 0 mode1
loop60:     mov     th0,#0feh       ;counting value
            mov     tl0,#0d4h
            setb    tr0             ;starting timer

wait60:     jnb     tf0,wait60      ;wait for overflow
            clr     tr0             ;stop timer 0
            clr     tf0             ;clear overflow flag
            mov     a,memkey
mkc60:      cjne    a,#03h,mkc66
            cpl     f_led
            ljmp    duty60f

mkc66:
            cpl     r_led
            ljmp    duty60f

;*********** t - off=40 %  *************
duty60f:
            mov     tmod,#01h       ;timer 0 mode1
loop60f:    mov     th0,#0ffh       ;counting value
            mov     tl0,#38h
            setb    tr0             ;starting timer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov     a,#0c0h
            lcall   msg19

wait60f:    jnb     tf0,wait60f             ;wait for overflow
            clr     tr0                     ;stop timer 0
            clr     tf0                     ;clear overflow flag


            lcall   updatedisp              ;เรียกโปรแกรม update display
            jb      infa1,st6               ;stop by infarate
            jb      infa2,st6
            lcall   scankey                 ;checkey
            lcall   delay
            cjne    a,#0ch,loop60

st6:        ljmp    stop

;*********** end of duty 60 % ************************************************

;*********** duty cycle = 70 % ***********************************************
;******* t - on=70 %  **************

duty70:

            mov     a,memkey
mk70:       cjne    a,#03h,mk77         ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
            ljmp    f5

mk77:       clr     r_led               ;ให้หมุนขวา
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          ljmp    l70

f5:       clr     f_led            ;ให้หมุนซ้าย
          ljmp    l70

l70:

          mov     tmod,#01h        ;timer 0 mode1
loop70:   mov     th0,#0feh        ;counting value
          mov     tl0,#0a2h
          setb    tr0              ;starting timer

wait70:   jnb     tf0,wait70       ;wait for overflow
          clr     tr0              ;stop timer 0
          clr     tf0              ;clear overflow flag
          mov     a,memkey
mkc70:    cjne    a,#03h,mkc77
          cpl     f_led
          ljmp    duty70f
mkc77:
          cpl     r_led
          ljmp    duty70f

;*********** t - off=30 %  *************
duty70f:
          mov     tmod,#01h        ;timer 0 mode1
loop70f:  mov     th0,#0ffh        ;counting value
          mov     tl0,#6ah
          setb    tr0              ;starting timer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     a,#0c0h
                lcall   msg19

wait70f:        jnb     tf0,wait70f          ;wait for overflow
                clr     tr0                  ;stop timer 0
                clr     tf0                  ;clear overflow flag
                lcall   updatedisp           ;เรียกโปรแกรม update display
                jb      infa1,st7            ;stop by infarate
                jb      infa2,st7
                lcall   scankey              ;checkey
                lcall   delay
                cjne    a,#0ch,loop70
st7:            ljmp    stop

;*********** end of duty 70 %
**************************************************

;*********** duty cycle = 80 %
**************************************************
;******* t - on=80 %  **************

duty80:
                mov     a,memkey
mk80:           cjne    a,#03h,mk88          ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
                ljmp    f6

mk88:           clr     r_led                ;ให้หมุนขวา
                ljmp    l80

f6:             clr     f_led                ;ให้หมุนซ้าย
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ljmp    l80

l80:
            mov     tmod,#01h           ;timer 0 mode1
loop80:     mov     th0,#0feh           ;counting value
            mov     tl0,#70h
            setb    tr0                 ;starting timer

wait80:     jnb     tf0,wait80          ;wait for overflow
            clr     tr0                 ;stop timer 0
            clr     tf0                 ;clear overflow flag
            mov     a,memkey
mkc80:      cjne    a,#03h,mkc88
            cpl     f_led
            ljmp    duty80f

mkc88:
            cpl     r_led
            ljmp    duty80f
;*********** t - off=20 %  *************
duty80f:
            mov     tmod,#01h           ;timer 0 mode1
loop80f:    mov     th0,#0ffh           ;counting value
            mov     tl0,#9ch
            setb    tr0                 ;starting timer
            mov     a,#0c0h
            lcall   msg19

wait80f:    jnb     tf0,wait80f         ;wait for overflow
            clr     tr0                 ;stop timer 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            clr     tf0                 ;clear overflow flag
            lcall   updatedisp          ;เรียกโปรแกรม update display
            jb      infa1,st8           ;stop by infarate
            jb      infa2,st8
            lcall   scankey             ;checkey
            lcall   delay
            cjne    a,#0ch,loop80
st8:        ljmp    stop

;*********** end of duty 80 % ***************************



;*********** duty cycle = 90 % **************************
;******* t - on=90 %  **************

duty90:
            mov     a,memkey
mk90:       cjne    a,#03h,mk99         ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
            ljmp    f7
mk99:       clr     r_led               ;ให้หมุนขวา
            ljmp    l90

f7:         clr     f_led               ;ให้หมุนซ้าย
            ljmp    l90

l90:

            mov     tmod,#01h           ;timer 0 mode1
loop90:     mov     th0,#0feh           ;counting value
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov     tl0,#03eh
            setb    tr0                 ;starting timer

wait90:     jnb     tf0,wait90          ;wait for overflow
            clr     tr0                 ;stop timer 0
            clr     tf0                 ;clear overflow flag
            mov     a,memkey
mkc90:      cjne    a,#03h,mkc99
            cpl     f_led
            ljmp    duty90f
mkc99:
            cpl     r_led
            ljmp    duty90f
;*********** t - off=10 %  *************
duty90f:
            mov     tmod,#01h           ;timer 0 mode1
loop90f:    mov     th0,#0ffh           ;counting value
            mov     tl0,#0ceh
            setb    tr0                 ;starting timer
            mov     a,#0c0h
            lcall   msg19

wait90f:    jnb     tf0,wait90f         ;wait for overflow
            clr     tr0                 ;stop timer 0
            clr     tf0                 ;clear overflow flag
            lcall   updatedisp          ;เรียกโปรแกรม update display
            jb      infa1,st9           ;stop by infarate
            jb      infa2,st9
            lcall   scankey             ;checkey
            lcall   delay
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                cjne    a,#0ch,loop90
st9:            ljmp    stop

;*********** end of duty 90 % *********************************************

;*********** duty cycle 100 % *********************************************

duty100:
                mov     a,memkey
mk100:          cjne    a,#03h,mk101        ;เปรียบเทียบค่าในหน่วยความจำ(การหมุน)
                ljmp    full

mk101:          clr     r_led               ;ให้หมุนขวา
                ljmp    l100

full:           clr     f_led               ;ให้หมุนซ้าย
                ljmp    l100
                mov     a,#0c0h
                lcall   msg19

l100:
                lcall   updatedisp          ;เรียกโปรแกรม update display
                jb      infa1,st10          ;stop by infarate
                jb      infa2,st10
                lcall   scankey             ;checkey
                lcall   delay
                cjne    a,#0ch,duty100
st10:           ljmp    stop
;*********** end of duty 100 % *********************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**************************************************
;*          scankey                                *
;**************************************************
scankey:
KEYIN:
            MOV   NKEY,#00H

CHK_C0:     MOV   A,#0EFH
            MOV   P1,A
            MOV   A,P1
            ANL   A,#00FH
            CJNE  A,#00FH,COL0_IN
            AJMP  CHK_C1

COL0_IN:
            MOV   NKEY,#00H
            AJMP  CHKROW

CHK_C1:     MOV   A,#0DFH
            MOV   P1,A
            MOV   A,P1
            ANL   A,#00FH
            CJNE  A,#00FH,COL1_IN
            AJMP  CHK_C2

COL1_IN:    MOV   NKEY,#01H
            AJMP  CHKROW

CHK_C2:     MOV   A,#0BFH
            MOV   P1,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV    A,P1
            ANL    A,#00FH
            CJNE   A,#00FH,COL2_IN
            AJMP   CHK_C3

COL2_IN:    MOV    NKEY,#02H
            AJMP   CHKROW

CHK_C3:     MOV    A,#07FH
            MOV    P1,A
            MOV    A,P1
;           MOVX   A,@DPTR
            ANL    A,#00FH
            CJNE   A,#00FH,COL3_IN
            AJMP   nokey

COL3_IN:    MOV    NKEY,#03H

CHKROW:     MOV    A,P1
            ANL    A,#0FH
            CJNE   A,#0EH,CHK_R1
            MOV    A,NKEY
            ADD    A,#00H
            MOV    NKEY,A
            RET

CHK_R1:     MOV    A,P1
            ANL    A,#0FH
            CJNE   A,#0DH,CHK_R2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV    A,NKEY
            ADD    A,#04H
            MOV    NKEY,A
            RET

CHK_R2:     MOV    A,P1
            ANL    A,#0FH
            CJNE   A,#0BH,CHK_R3
            MOV    A,NKEY
            ADD    A,#08H
            MOV    NKEY,A
            RET

CHK_R3:     MOV    A,P1
            ANL    A,#0FH
            CJNE   A,#07H,NOKEY
            MOV    A,NKEY
            ADD    A,#0CH
            MOV    NKEY,A
            RET

NOKEY:
            mov    a,#0f0h
            ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*********************** long   delay *******************

dtime:      mov    dtime0,#0ffh
dt1:        mov    dtime1,#0ffh
dt0:        mov    dtime2,#0ffh
            djnz   dtime2,$
            djnz   dtime1,dt0
            djnz   dtime0,dt1
            ret
;********************** end of long delay ***************

;*************************************************
;         stop motor                                     *
;*************************************************

stop:       setb   f_led
            setb   r_led
            lcall  delay
            clr    f_led
            lcall  delay
            lcall  delay
            setb   f_led
            lcall  delay
            lcall  delay
            clr    r_led
            lcall  delay
            lcall  delay
            setb   r_led
            setb   f_led                 ;protection การไม่ให้transister ทำงานพร้อมกัน
            mov    a,#80h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   msg0
                mov     a,#0c0h
                lcall   msg0
                ljmp    start

;**************************************************
;           select duty cycle                     *
;**************************************************

selduty:
                mov     a,#80h
                lcall   msg0
                lcall   delay
                mov     a,#0c0h
                lcall   msg0
                lcall   delay
                mov     a,#80h
                lcall   msg8                    ;'press 0-9 to select dutycycle
                lcall   delay

selmain:        lcall   scankey
                lcall   delay
                cjne    a,#0fh,clear1
                ret

clear1:         lcall   scankey
                cjne    a,#0ch,sel0
                mov     a,#0c0h
                lcall   msg0
                ljmp    start
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sel0:   cjne    a,#0dh,sel1        ;ถ้าไม่ใช่1กระโดดไปเช็คkey2
        mov     memkey1,a          ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg9               ;dutycycle 100% enter
        lcall   scankey
        cjne    a,#0fh,selmain
        ret

sel1:   cjne    a,#00h,sel2        ;ถ้าไม่ใช่1กระโดดไปเช็คkey3
        mov     memkey1,a          ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg10              ;dutycycle 10% enter
        lcall   scankey
        cjne    a,#0fh,selmain
        ret

sel2:   cjne    a,#01h,sel3        ;ถ้าไม่ใช่1กระโดดไปเช็คkey4
        mov     memkey1,a          ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg11              ;dutycycle 20% enter
        lcall   scankey
        cjne    a,#0fh,selmain
        ret
sel3:   cjne    a,#02h,sel4        ;ถ้าไม่ใช่1กระโดดไปเช็คkey5
        mov     memkey1,a          ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg12              ;dutycycle 30% enter
        lcall   scankey
        cjne    a,#0fh,selmain
        ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sel4:   cjne    a,#04h,sel5         ;ถ้าไม่ใช่1กระโดดไปเช็คkey6
        mov     memkey1,a           ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg13               ;dutycycle 40% enter
        lcall   scankey
        cjne    a,#0fh,selmain
        ret

sel5:   cjne    a,#05h,sel6         ;ถ้าไม่ใช่1กระโดดไปเช็คkey7
        mov     memkey1,a           ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg14               ;dutycycle 50% enter
        lcall   scankey
        cjne    a,#0fh,selx1
        ret
selx1:  ljmp    selmain

sel6:   cjne    a,#06h,sel7         ;ถ้าไม่ใช่1กระโดดไปเช็คkey8
        mov     memkey1,a           ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg15               ;dutycycle 60% enter
        lcall   scankey
        cjne    a,#0fh,selx2
        ret
selx2:  ljmp    selmain

sel7:   cjne    a,#08h,sel8         ;ถ้าไม่ใช่1กระโดดไปเช็คkey9
        mov     memkey1,a           ;เก็บค่าที่กด
        mov     a,#0c0h
        lcall   msg16               ;dutycycle 70% enter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   scankey
                cjne    a,#0fh,selx3
                ret
selx3:          ljmp    selmain

sel8:           cjne    a,#09h,sel9           ;ถ้าไม่ใช่1กระโดดไปเช็คkey0
                mov     memkey1,a             ;เก็บค่าที่กด
                mov     a,#0c0h
                lcall   msg17                 ;dutycycle 80% enter
                lcall   scankey
                cjne    a,#0fh,selx4
                ret
selx4:          ljmp    selmain

sel9:           cjne    a,#0ah,selx4          ;ถ้าไม่ใช่โดดไปselmain
                mov     memkey1,a             ;เก็บค่าที่กด
                mov     a,#0c0h
                lcall   msg18                 ;dutycycle 90% enter
                lcall   scankey
                cjne    a,#0fh,selx4
                ret
;****************************************************
updatedisp:     ret
msg0:           lcall   lcdwi
                mov     dptr,#str0
                lcall   text
                ret


msg1:           lcall   lcdwi
                mov     dptr,#str1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   text
                ret
msg2:           lcall   lcdwi
                mov     dptr,#str2
                lcall   text
                ret
msg3:           lcall   lcdwi
                mov     dptr,#str3
                lcall   text
                ret
msg4:           lcall   lcdwi
                mov     dptr,#str4
                lcall   text
                ret
msg5:           lcall   lcdwi
                mov     dptr,#str5
                lcall   text
                ret
msg6:           lcall   lcdwi
                mov     dptr,#str6
                lcall   text
                ret
msg7:           lcall   lcdwi
                mov     dptr,#str7
                lcall   text
                ret
msg8:           lcall   lcdwi
                mov     dptr,#str8
                lcall   text
                ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
msg9:       lcall   lcdwi
            mov     dptr,#str9
            lcall   text
            ret
msg10:      lcall   lcdwi
            mov     dptr,#str10
            lcall   text
            ret
msg11:      lcall   lcdwi
            mov     dptr,#str11
            lcall   text
            ret
msg12:      lcall   lcdwi
            mov     dptr,#str12
            lcall   text
            ret
msg13:      lcall   lcdwi
            mov     dptr,#str13
            lcall   text
            ret
msg14:      lcall   lcdwi
            mov     dptr,#str14
            lcall   text
            ret
msg15:      lcall   lcdwi
            mov     dptr,#str15
            lcall   text
            ret
msg16:      lcall   lcdwi
            mov     dptr,#str16
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   text
                ret
msg17:          lcall   lcdwi
                mov     dptr,#str17
                lcall   text
                ret
msg18:          lcall   lcdwi
                mov     dptr,#str18
                lcall   text
                ret
msg19:          lcall   lcdwi
                mov     dptr,#str19
                lcall   text
                ret
;******************* H/D , D/H , Encoder ****************
;**********************
;* HEX TO DECIMAL     *
;* input : dptr       *
;* output : r1:r2:r3: *
;**********************

htod:           clr     a
                mov     r1,a
                mov     r2,a
                mov     r3,a
                mov     r4,#16
htod1:          mov     a,dpl
                rlc     a
                mov     dpl,a
                mov     a,dph
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                rlc     a
                mov     dph,a
                mov     r5,#3
                mov     r0,#3
htod2:          mov     a,@R0
                addc    a,acc
                da      a
                mov     @r0,a
                dec     r0
                djnz    r5,htod2
                djnz    r4,htod1
                ret
;***************************************************
;* DECIMAL TO HEX   *
;* INPUT : R1:R2:R3 *
;* OUTPUT : DPTR    *
;***************************************************

dtoh:           mov     r4,#16                  ; CONVERT DECIMAL -> HEX
dtoh1:          mov     r5,#3
                mov     r0,#1
                clr     c
dtoh2:          mov     a,@R0
                rrc     a
                push    psw
                jnb     acc.7,dtoh3
                clr     c
                subb    a,#30h
dtoh3:          jnb     acc.3,dtoh4
                clr     c
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                subb    a,#3
dtoh4:          mov     @r0,a
                inc     r0
                pop     psw
                djnz    r5, dtoh2
                mov     a,dph
                rrc     a
                mov     dph,a
                mov     a,dpl
                rrc     a
                mov     dpl,a
                djnz    r4,dtoh1
                ret

;* Out digit to lcd
;* input : dptr
;* output : display at current position of LCD

digit:          lcall   htod
                mov     a,r1        ;r1
                rr      a
                rr      a
                rr      a
                rr      a
                anl     a,#0fh
                add     a,#30h
                lcall   lcdwd
                mov     a,r1
                anl     a,#0fh
                add     a,#30h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
lcall   lcdwd

mov     a,r2        ; r2
rr      a
rr      a
rr      a
rr      a
anl     a,#0fh
add     a,#30h
lcall   lcdwd
mov     a,r2
anl     a,#0fh
add     a,#30h
lcall   lcdwd
mov     a,r3        ; r3
rr      a
rr      a
rr      a
rr      a
anl     a,#0fh
add     a,#30h
lcall   lcdwd
mov     a,r3
anl     a,#0fh
add     a,#30h
lcall   lcdwd
ret
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************
; READ ENCODER                                   *
; CALL FROM INTERRUPT                            *
;*************************************************

ENCODER:    PUSH  PSW
            PUSH  ACC              ; SAVE REGISTER & FLAG
            PUSH  0
            PUSH  1
            PUSH  2
            MOV   R0,ENCPOINT      ; CHECK NEW POSITION
            MOV   A,P3
            ANL   A,#3
            CJNE  A,ENCNEW,ENCODER3
            SJMP  ENCODERO
ENCODER3:   MOV   ENCNEW,A         ; CHECK BUTTOM POINT
            MOV   A,R0
            MOV   R1,A
            CJNE  A,#ENCDATA,ENCODER5
            MOV   R0,#ENCDATA+3
            SJMP  ENCODER6
ENCODER5:   DEC   R0
ENCODER6:   MOV   A,@R0
            CJNE  A,ENCNEW,ENCODER4
            MOV   ENCPOINT,R0      ; ADD TO POSITION
            MOV   A,ABSPOST+1
            ADD   A,#1
            MOV   ABSPOST+1,A
            MOV   A,ABSPOST
            ADDC  A,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV   ABSPOST,A
            SJMP  ENCODERO
ENCODER4:   PUSH  1                    ; CHECK TOP POINT
            POP   0
            MOV   A,R0
            CJNE  A,#ENCDATA+3,ENCODER7
            MOV   R0,#ENCDATA
            SJMP  ENCODER8
ENCODER7:   INC   R0
ENCODER8:   MOV   A,@R0
            CJNE  A,ENCNEW,ENCODERO
            MOV   ENCPOINT,R0          ; SUB TO POSITION

            CLR   C
            MOV   A,ABSPOST+1
            SUBB  A,#1
            MOV   ABSPOST+1,A
            MOV   A,ABSPOST
            SUBB  A,#0
            MOV   ABSPOST,A
            SJMP  ENCODERO
ENCODERO:   POP   2
            POP   1
            POP   0
            POP   ACC
            POP   PSW
            RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;******************* String Measage **********************
str0:   db   '                    ',0
str1:   db   'press 2ND to select ',0
str2:   db   'speed press enter   ',0
str3:   db   'position press enter',0
str4:   db   'press->,<-to select ',0
str5:   db   'forward press enter ',0
str6:   db   'reverse press enter ',0
str7:   db   'press CLEAR to STOP ',0
str8:   db   'press 0-9 to select ',0
str9:   db   'dutycycle 100% enter',0
str10:  db   'dutycycle 10% enter ',0
str11:  db   'dutycycle 20% enter ',0
str12:  db   'dutycycle 30% enter ',0
str13:  db   'dutycycle 40% enter ',0
str14:  db   'dutycycle 50% enter ',0
str15:  db   'dutycycle 60% enter ',0
str16:  db   'dutycycle 70% enter ',0
str17:  db   'dutycycle 80% enter ',0
str18:  db   'dutycycle 90% enter ',0
str20:  db   'Encoder :           ',0
             end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข

# วิธีการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# วิธีการใช้งาน

วิธีการใช้งานของชุดทดลองนี้มีวิธีการดังต่อไปนี้

**ขั้นตอนที่1.** จะเห็นข้อความที่แสดงอยู่บนจอภาพว่า press 2nd to select

| |
|---|
| press 2nd to select |

**รูปที่ ข.1** การแสดงผลของจอ LCD 1

**ขั้นตอนที่2.** ถ้ากด 2 nd หนึ่งครั้งจะแสดงผลเป็น speed press enter

| |
|---|
| speed press enter |

**รูปที่ ข.2** การแสดงผลของจอ LCD 2

ถ้ากด 2 nd อีกครั้งจะแสดงผล position press enter

| |
|---|
| position press enter |

**รูปที่ ข.3** การแสดงผลของจอ LCD 3

**ขั้นตอนที่3.** กด enter ใน function ที่ต้องการเลือกเช่นเลือกโหมด speed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ขั้นตอนที่4.** จะแสดงผล press ←,→ to select

press <--,--> to select

**รูปที่ ข.4** การแสดงผลของจอ LCD 4

**ขั้นตอนที่5.** ให้ทำการกดปุ่ม ← จะมีข้อความแสดงบนจอภาพว่า forward press enter

forward press enter

**รูปที่ ข.5** การแสดงผลของจอ LCD 5

ถ้าหากกด → ซ้ำอีกครั้งหนึ่งจอภาพจะแสดงผลว่า reverse press enter

reverse press enter

**รูปที่ ข.6** การแสดงผลของจอ LCD 6

ถ้าหากมีการกด 2nd ซ้ำหลายๆครั้งก็จะแสดงผล speed และ position สลับกันไป ถ้าต้องการใช้งานFunctionใดให้กดปุ่ม enter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ขั้นตอนที่6.** เมื่อกดenterแล้วจะปรากฏข้อความว่า press 0-9 to select duty cycle

press 0-9 to select duty

**รูปที่ ข.7** การแสดงผลของจอ LCD 7

ให้กดปุ่ม 0ไปจนถึง 9 เพื่อเลือก เปอร์เซ็นต์ ดิวตี้ไซเคิลในการทำงานถ้ากด 1 จะแสดงผลดังนี้10 % duty cycle press enter

10% duty cycle press enter

**รูปที่ ข.8** การแสดงผลของจอ LCD 8

และถ้ากดไปเรื่อยๆ เรียงลำดับไปจนถึง 9 คือ 90 % duty cycle press enter

90% duty cycle press enter

**รูปที่ ข.9** การแสดงผลของจอ LCD 9

100% duty cycle press enter

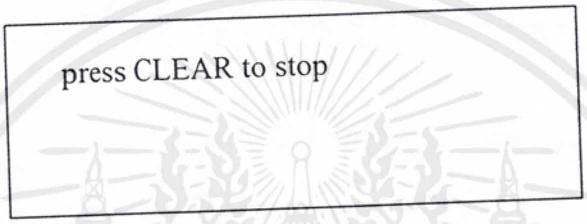
**รูปที่ ข.10** การแสดงผลของจอ LCD 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ถ้าหากกดปุ่ม 0 จะเป็น 100 % duty cycle ซึ่งเป็นการสั่งให้มอเตอร์หมุนที่ความเร็วสูงสุด

**ขั้นตอนที่7.** เมื่อกดปุ่ม enter แล้วสายพานจะเริ่มหมุดด้วยทิศทางและความเร็วตามเปอร์เซ็นต์ดิวตี้ไซเคิลที่กำหนด

**ขั้นตอนที่8.** จากนั้นจอภาพจะแสดงผล press CLEAR  to stop ถ้าหากกด enter สายพานจะหยุด




**รูปที่ ข.11** การแสดงผลของจอ LCD 11

หมุนสายพานจะทำการติดตั้งตัวตรวจจับอินฟาเรดไว้ที่หัวและท้ายของสายพานด้วยดังนั้นถ้าวัตถุเคลื่อนที่มาจนสุดสายพานแล้วจะทำให้หยุดสายพานได้เองโดยอัตโนมัติหรือสามารถนำไปใช้ในการรับค่าก็ได้

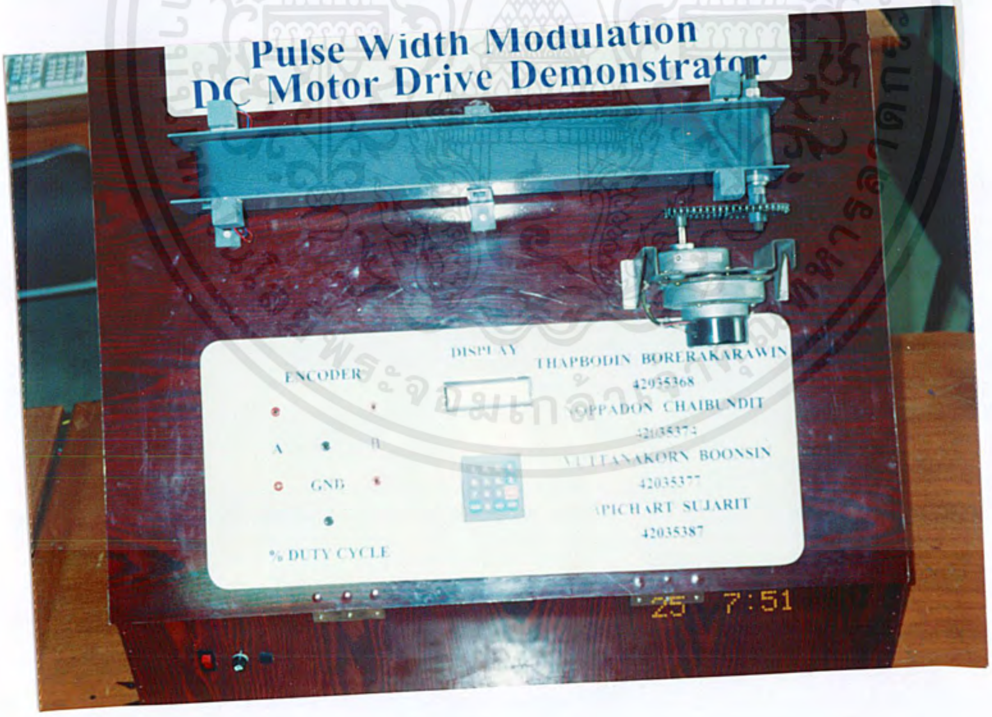
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ค

# รูปวงจร และภาพโครงงาน

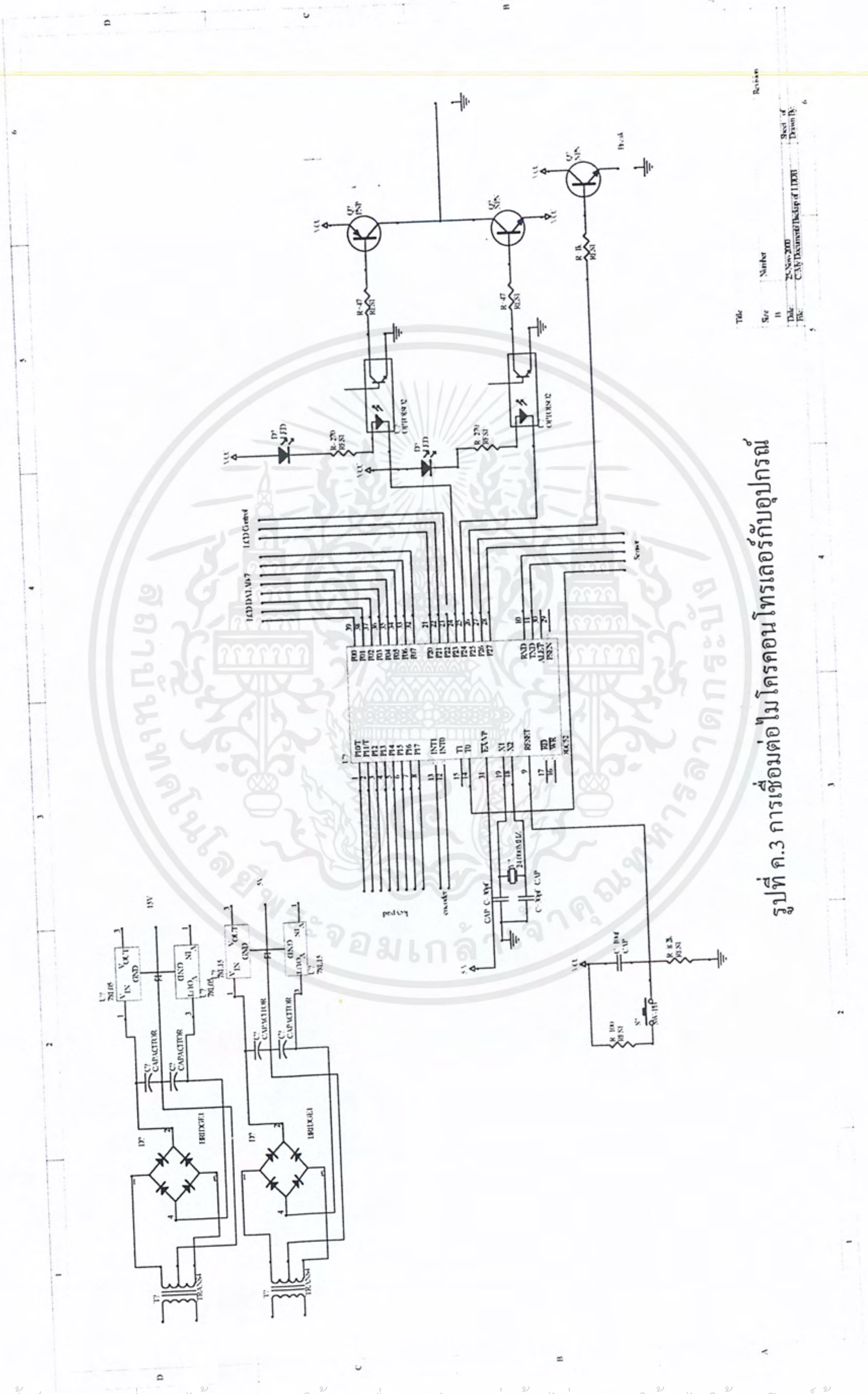
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ค.1 วิธีการตรวจจับตำแหน่ง




รูปที่ ค.2 เครื่องต้นแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ค.3 การเชื่อมต่อไมโครคอนโทรเลอร์กับอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ง
# รายละเอียดคุณสมบัติเฉพาะของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ.ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# 8-Bit Microcontroller

# with 8K Bytes Flash AT89S8252

## Features

- Compatible with MCS-51™ Products
- 8K Bytes of In-System Re programmable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2K Bytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- 4.0V to 6V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer
- Power Off Flag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Description**

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8K bytes of Downloadable Flash programmable and erasable read only memory and 2K bytes of EEPROM. The device is manufactured using Atmel's high density nonvol-atile memory technology and is compatible with the industry standard 80C51 instruction set and pin out. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional non-volatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications. The AT89S8252 provides the following standard features: 8K bytes of Downloadable Flash, 2K bytes of EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watch-dog timer, two Data Pointers, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt sys-tem to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hard-ware reset. The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a serial programming interface and allows the program memory to be written to or read from unless Lock Bit 2 has been activated.

**Pin Description**

**V**cc

Supply voltage.

**GND**

Ground.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Port 0**

Port 0 is an 8-bit open drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs. Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to x-ternal pro-gram and data memory. In this mode, P0 has internal pull ups. Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pull ups are required during program verification.

**Port 1**

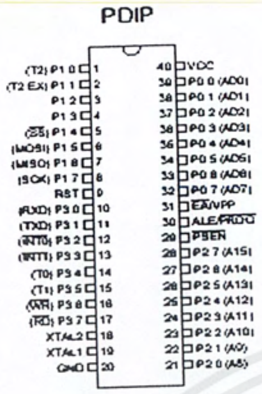
Port 1 is an 8-bit bi-directional I/O port with internal pull ups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pull ups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current ($I_{IL}$) because of the internal pull ups. Some Port 1 pins provide additional functions. P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively. Port 1 also receives the low-order address bytes during Flash programming and verification.

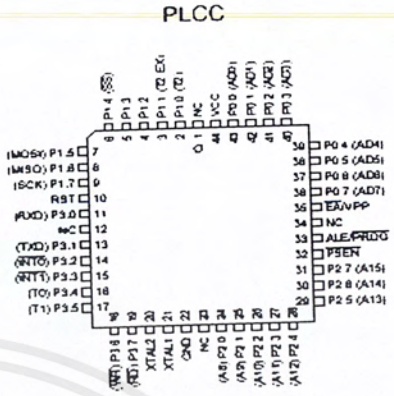
**Port 2**

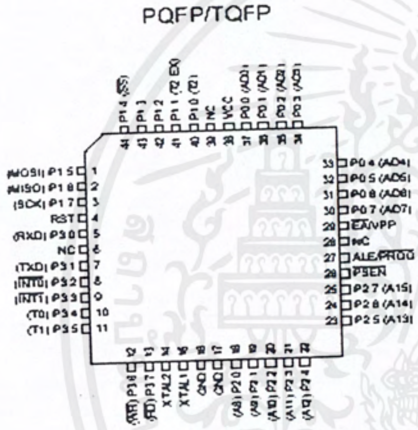
Port 2 is an 8-bit bi-directional I/O port with internal pull ups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pull ups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current ($I_{IL}$) because of the internal pull ups. Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pull ups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register. Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Pin Configurations**

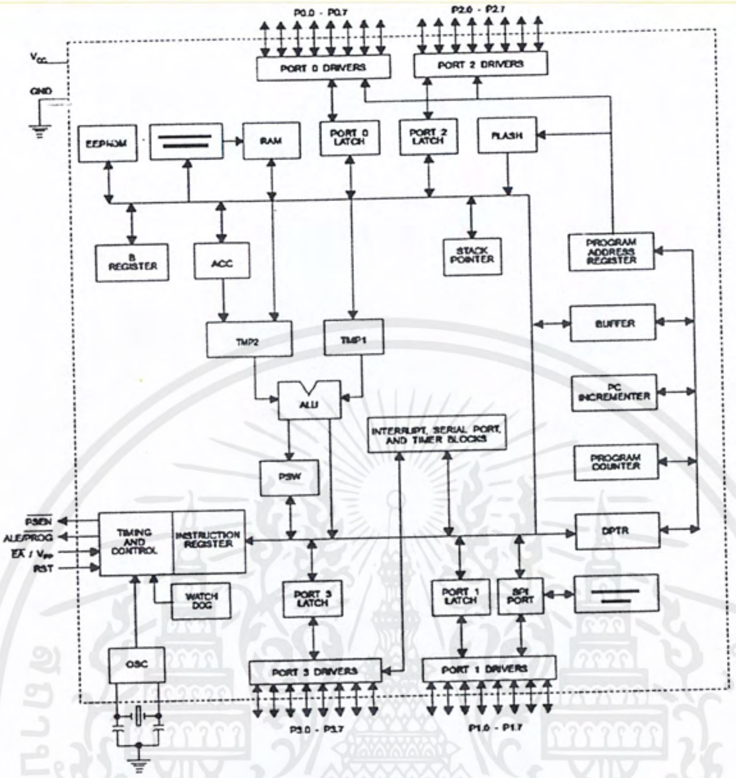






เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Block Diagram**




| Port Pin | Alternate Functions |
|---|---|
| P1.0 | T2 (external count input to Timer/Counter 2), clock-out |
| P1.1 | T2EX (Timer/Counter 2 capture/reload trigger and direction control) |
| P1.4 | $\overline{SS}$ (Slave port select input) |
| P1.5 | MOSI (Master data output, slave data input pin for SPI channel) |
| P1.6 | MISO (Master data input, slave data output pin for SPI channel) |
| P1.7 | SCK (Master clock output, slave clock input pin for SPI channel) |

Furthermore, P1.4, P1.5, P1.6, and P1.7 can be configured as the SPI slave port select, data input/output and shift clock input/output pins as shown in the following table.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**Port 3**

Port 3 is an 8 bit bi-directional I/O port with internal pull ups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pull ups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current ($I_{IL}$) because of the pull ups. Port 3 also serves the functions of various special features of the AT89S8252, as shown in the following table. Port 3 also receives some control signals for Flash programming and verification.

| Port Pin | Alternate Functions |
|---|---|
| P3.0 | RXD (serial input port) |
| P3.1 | TXD (serial output port) |
| P3.2 | $\overline{INT0}$ (external interrupt 0) |
| P3.3 | $\overline{INT1}$ (external interrupt 1) |
| P3.4 | T0 (timer 0 external input) |
| P3.5 | T1 (timer 1 external input) |
| P3.6 | $\overline{WR}$ (external data memory write strobe) |
| P3.7 | $\overline{RD}$ (external data memory read strobe) |

**RST**

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

**ALE/PROG**

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input (PROG) during Flash programming. In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory. If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the micro controller is in external execution mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**PSEN**

Program Store Enable is the read strobe to external pro-gram memory. When the AT89S8252 is executing code from external program memory, PSEN is activated twice each machine cycle, except that two PSEN activations are skipped during each access to external data memory.

**EA/V PP**

External Access Enable. EA must be strapped to GND in order to enable the device to fetch code from external pro-gram memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, EA will be internally latched on reset. EA should be strapped to $V_{cc}$ for internal program executions. This pin also receives the 12-volt programming enable voltage ($V_{PP}$) during Flash programming when 12-volt programming is selected.

**XTAL1**

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

**XTAL2**

Output from the inverting oscillator amplifier.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. AT89S8252 SFR Map and Reset Values

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0F8H | | | | | | | | | 0FFH |
| 0F0H | B 00000000 | | | | | | | | 0F7H |
| 0E8H | | | | | | | | | 0EFH |
| 0E0H | ACC 00000000 | | | | | | | | 0E7H |
| 0D8H | | | | | | | | | 0DFH |
| 0D0H | PSW 00000000 | | | | | SPCR 000001XX | | | 0D7H |
| 0C8H | T2CON 00000000 | T2MOD XXXXXX00 | RCAP2L 00000000 | RCAP2H 00000000 | TL2 00000000 | TH2 00000000 | | | 0CFH |
| 0C0H | | | | | | | | | 0C7H |
| 0B8H | IP XX000000 | | | | | | | | 0BFH |
| 0B0H | P3 11111111 | | | | | | | | 0B7H |
| 0A8H | IE 0X000000 | | SPSR 00XXXXXX | | | | | | 0AFH |
| 0A0H | P2 11111111 | | | | | | | | 0A7H |
| 98H | SCON 00000000 | SBUF XXXXXXXX | | | | | | | 9FH |
| 90H | P1 11111111 | | | | | | WMCON 00000010 | | 97H |
| 88H | TCON 00000000 | TMOD 00000000 | TL0 00000000 | TL1 00000000 | TH0 00000000 | TH1 00000000 | | | 8FH |
| 80H | P0 11111111 | SP 00000111 | DP0L 00000000 | DP0H 00000000 | DP1L 00000000 | DP1H 00000000 | SPDR XXXXXXXX | PCON 0XXX0000 | 87H |

## Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

**Note:** that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect. User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Timer 2 Registers

Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 9) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16 bit capture mode or 16-bit auto-reload mode.

## Watchdog and Memory Control Register

The WMCON register contains control bits for the Watchdog Timer (shown in Table 3).

**Table 2.** T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | TF2 | EXF2 | RCLK | TCLK | EXEN2 | TR2 | $C/\overline{T2}$ | $CP/\overline{RL2}$ |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| TF2 | Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1. |
| EXF2 | Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1). |
| RCLK | Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflows to be used for the receive clock. |
| TCLK | Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock. |
| EXEN2 | Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX. |
| TR2 | Start/Stop control for Timer 2. TR2 = 1 starts the timer. |
| $C/\overline{T2}$ | Timer or counter select for Timer 2. $C/\overline{T2}$ = 0 for timer function. $C/\overline{T2}$ = 1 for external event counter (falling edge triggered). |
| $CP/\overline{RL2}$ | Capture/Reload select. $CP/\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. $CP/\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow. |

The EEMEN and EEMWE bits are used to select the 2K bytes on-chip EEPROM, and to enable byte-write. The DPS bit selects one of two DPTR registers available.

## SPI Registers

Control and status bits for the Serial Peripheral Interface are contained in registers SPCR (shown in Table 4) and SPSR (shown in Table 5). The SPI data bits are contained in the SPDR register. Writing the SPI data register during serial data transfer sets the Write Collision bit, WCOL, in the SPSR register. The SPDR is double buffered for writing and the values in SPDR are not changed by Reset.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Interrupt Registers

The global interrupt enable bit and the individual interrupt enable bits are in the IE register. In addition, the individual interrupt enable bit for the SPI is in the SPCR register. Two priorities can be set for each of the six interrupt sources in the IP register.

## Dual Data Pointer Registers

To facilitate accessing both internal EEPROM and external data memory, two banks of 16 bit Data Pointer Registers are provided: DP0 at SFR address locations 82H-83H and DP1 at

**Table 3.** WMCON—Watchdog and Memory Control Register

WMCON Address = 96H    Reset Value = 0000 0010B

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | PS2 | PS1 | PS0 | EEMWE | EEMEN | DPS | WDTRST | WDTEN |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| PS2<br>PS1<br>PS0 | Prescaler Bits for the Watchdog Timer. When all three bits are set to "0", the watchdog timer has a nominal period of 16 ms. When all three bits are set to "1", the nominal period is 2048 ms. |
| EEMWE | EEPROM Data Memory Write Enable Bit. Set this bit to "1" before initiating byte write to on-chip EEPROM with the MOVX instruction. User software should set this bit to "0" after EEPROM write is completed. |
| EEMEN | Internal EEPROM Access Enable. When EEMEN = 1, the MOVX instruction with DPTR will access on-chip EEPROM instead of external data memory. When EEMEN = 0, MOVX with DPTR accesses external data memory. |
| DPS | Data Pointer Register Select. DPS = 0 selects the first bank of Data Pointer Register, DP0, and DPS = 1 selects the second bank, DP1 |
| WDTRST<br>RDY/$\overline{BSY}$ | Watchdog Timer Reset and EEPROM Ready/$\overline{Busy}$ Flag. Each time this bit is set to "1" by user software, a pulse is generated to reset the watchdog timer. The WDTRST bit is then automatically reset to "0" in the next instruction cycle. The WDTRST bit is Write-Only. This bit also serves as the RDY/$\overline{BSY}$ flag in a Read-Only mode during EEPROM write. RDY/$\overline{BSY}$ = 1 means that the EEPROM is ready to be programmed. While programming operations are being executed, the RDY/$\overline{BSY}$ bit equals "0" and is automatically reset to "1" when programming is completed. |
| WDTEN | Watchdog Timer Enable Bit. WDTEN = 1 enables the watchdog timer and WDTEN = 0 disables the watchdog timer. |

84H-85H. Bit DPS = 0 in SFR WMCON selects DP0 and DPS = 1 selects DP1. The user should always initialize the DPS bit to the appropriate value before accessing the respective Data Pointer Register.

## Power Off Flag

The Power Off Flag (POF) is located at bit_4 (PCON.4) in the PCON SFR. POF is set to "1" during power up. It can be set and reset under software control and is not affected by RESET.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 4. SPCR—SPI Control Register

SPCR Address = D5H — Reset Value = 0000 01XXB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPIE | SPE | DORD | MSTR | CPOL | CPHA | SPR1 | SPR0 |

| Symbol | Function |
|---|---|
| SPIE | SPI Interrupt Enable. This bit, in conjunction with the ES bit in the IE register, enables SPI interrupts: SPIE = 1 and ES = 1 enable SPI interrupts. SPIE = 0 disables SPI interrupts. |
| SPE | SPI Enable. SPI = 1 enables the SPI channel and connects $\overline{SS}$, MOSI, MISO and SCK to pins P1.4, P1.5, P1.6, and P1.7. SPI = 0 disables the SPI channel. |
| DORD | Data Order. DORD = 1 selects LSB first data transmission. DORD = 0 selects MSB first data transmission. |
| MSTR | Master/Slave Select. MSTR = 1 selects Master SPI mode. MSTR = 0 selects Slave SPI mode. |
| CPOL | Clock Polarity. When CPOL = 1, SCK is high when idle. When CPOL = 0, SCK of the master device is low when not transmitting. Please refer to figure on SPI Clock Phase and Polarity Control. |
| CPHA | Clock Phase. The CPHA bit together with the CPOL bit controls the clock and data relationship between master and slave. Please refer to figure on SPI Clock Phase and Polarity Control. |
| SPR0<br>SPR1 | SPI Clock Rate Select. These two bits control the SCK rate of the device configured as master. SPR1 and SPR0 have no effect on the slave. The relationship between SCK and the oscillator frequency, $F_{OSC}$, is as follows:<br>SPR1 SPR0 SCK = $F_{OSC}$ divided by<br>0 0 4<br>0 1 16<br>1 0 64<br>1 1 128 |

## Data Memory—EEPROM and RAM

The AT89S8252 implements 2K bytes of on-chip EEPROM for data storage and 256 bytes of RAM. The upper 128 bytes of RAM occupy a parallel space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space. When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space. For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2). MOV 0A0H, #data Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

MOV @R0, #data

**Note** that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are avail-able as stack space. The on-chip EEPROM data memory is selected by setting

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 5. SPSR—SPI Status Register

SPSR Address = AAH — Reset Value = 00XX XXXXB

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPIF | WCOL | — | — | — | — | — | — |

| Symbol | Function |
|---|---|
| SPIF | SPI interrupt Flag. When a serial transfer is complete, the SPIF bit is set and an interrupt is generated if SPIE = 1 and ES = 1. The SPIF bit is cleared by reading the SPI status register with SPIF and WCOL bits set, and then accessing the SPI data register. |
| WCOL | Write Collision Flag. The WCOL bit is set if the SPI data register is written during a data transfer. During data transfer, the result of reading the SPDR register may be incorrect, and writing to it has no effect. The WCOL bit (and the SPIF bit) are cleared by reading the SPI status register with SPIF and WCOL set, and then accessing the SPI data register. |

the EEMEN bit in the WMCON register at SFR address location 96H. The EEPROM address range is from 000H to 7FFH. The MOVX instructions are used to access the EEPROM. To access off-chip data memory with the MOVX instructions, the EEMEN bit needs to be set to "0". The EEMWE bit in the WMCON register needs to be set to "1" before any byte location in the EEPROM can be written. User software should reset EEMWE bit to "0" if no further

Table 6. SPDR—SPI Data Register

SPDR Address = 86H — Reset Value = unchanged

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 |

EEPROM write is required. EEPROM write cycles in the serial programming mode are self-timed and typically take 2.5 ms. The progress of EEPROM write can be monitored by reading the RDY/BSY bit (read-only) in SFR WMCON. RDY/BSY = 0 means programming is still in progress and RDY/BSY = 1 means EEPROM write cycle is completed and another write cycle can be initiated. In addition, during EEPROM programming, an attempted read from the EEPROM will fetch the byte being written with the MSB complemented. Once the write cycle is completed, true data are valid at all bit locations.

## Programmable Watchdog Timer

The programmable Watchdog Timer (WDT) operates from an independent oscillator. The prescaler bits, PS0, PS1 and PS2 in SFR WMCON are used to set the period of the Watchdog Timer from 16 ms to 2048 ms. The available timer periods are shown in the following table and the actual timer periods (at $V_{cc}$ = 5V) are within ฑ30% of the nominal. The WDT is disabled by

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power-on Reset and during Power Down. It is enabled by setting the WDTEN bit in SFR WMCON (address = 96H). The WDT is reset by setting the WDTRST bit in WMCON. When the WDT times out without being reset or disabled, an internal RST pulse is generated to reset the CPU.

**Table 7. Watchdog Timer Period Selection**

| WDT Prescaler Bits | | | Period (nominal) |
|---|---|---|---|
| PS2 | PS1 | PS0 | |
| 0 | 0 | 0 | 16 ms |
| 0 | 0 | 1 | 32 ms |
| 0 | 1 | 0 | 64 ms |
| 0 | 1 | 1 | 128 ms |
| 1 | 0 | 0 | 256 ms |
| 1 | 0 | 1 | 512 ms |
| 1 | 1 | 0 | 1024 ms |
| 1 | 1 | 1 | 2048 ms |

## Timer 0 and 1

Timer 0 and Timer 1 in the AT89S8252 operate the same way as Timer 0 and Timer 1 in the AT89C51, AT89C52 and AT89C55. For further information, see the October 1995 Micro controller Data Book, page 2-45, section titled, "Timer/Counters."

## Timer 2

Timer 2 is a 16 bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/T2 in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 8. Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency. In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

**Figure 1.** Timer 2 in Capture Mode

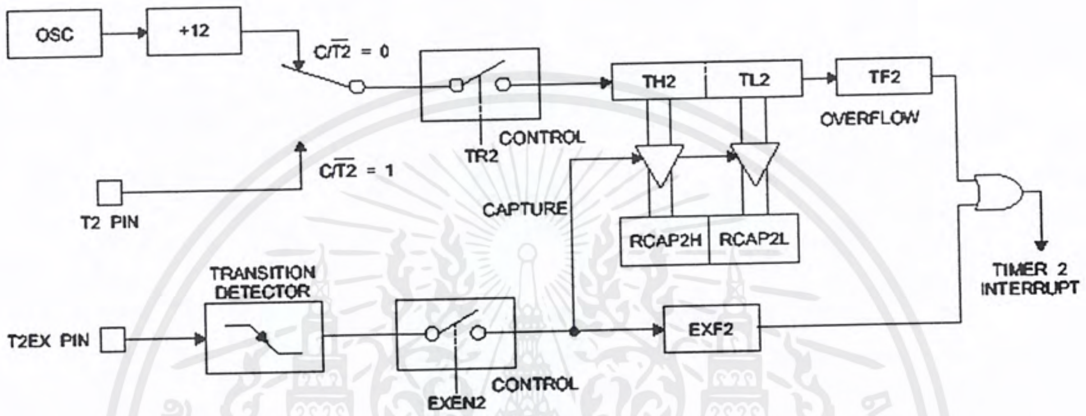



**Table 8.** Timer 2 Operating Modes

| RCLK + TCLK | CP/RL2 | TR2 | MODE |
|---|---|---|---|
| 0 | 0 | 1 | 16-bit Auto-Reload |
| 0 | 1 | 1 | 16-bit Capture |
| 1 | X | 1 | Baud Rate Generator |
| X | X | 0 | (Off) |

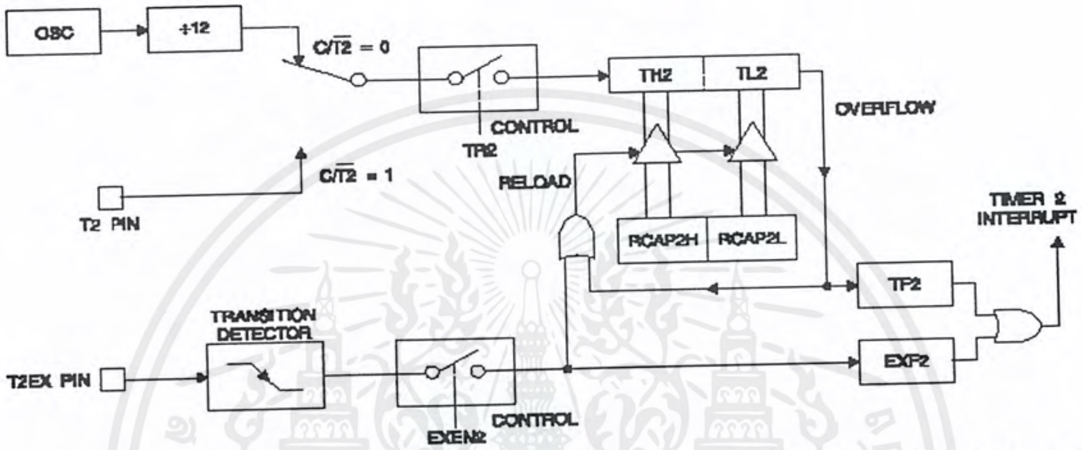
## Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16 bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a l-to- 0 transition at external input T2EX also causes the cur-rent value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1. **Auto-Reload (Up or Down Counter)** Timer 2 can be programmed to count up or down when configured in its 16 bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 9). Upon reset, the DCEN bit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin. Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The over-flow also

**Figure 2.** Timer 2 in Auto Reload Mode (DCEN = 0)




causes the timer registers to be reloaded with the 16 bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16 bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transitionalso sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled. Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16 bit value inRCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively. A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers. The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

**Table 9.** T2MOD – Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | T2OE | DCEN |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |

| Symbol | Function |
|---|---|
| — | Not implemented, reserved for future use. |
| T2OE | Timer 2 Output Enable bit. |
| DCEN | When set, this bit allows Timer 2 to be configured as an up/down counter. |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

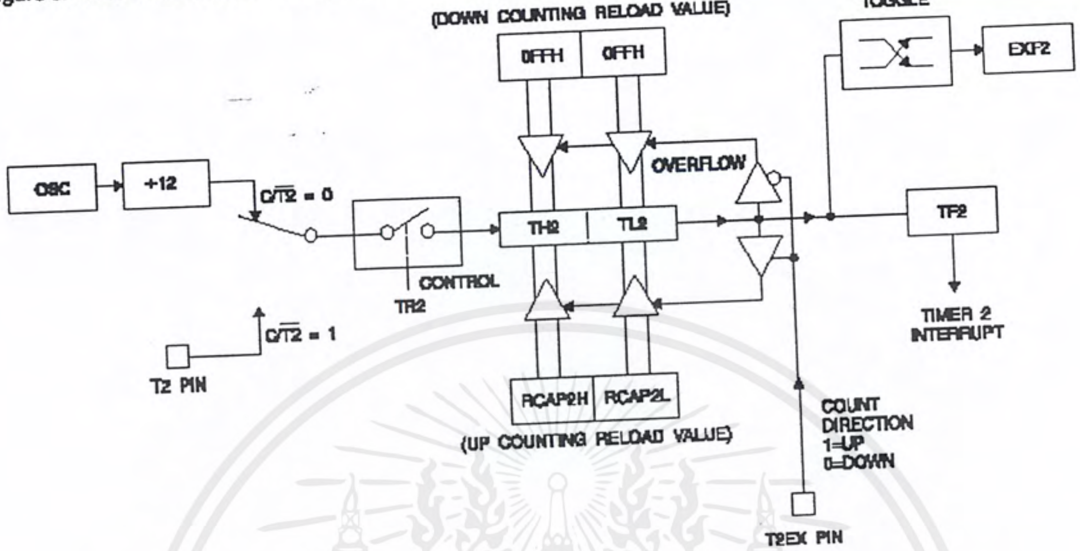
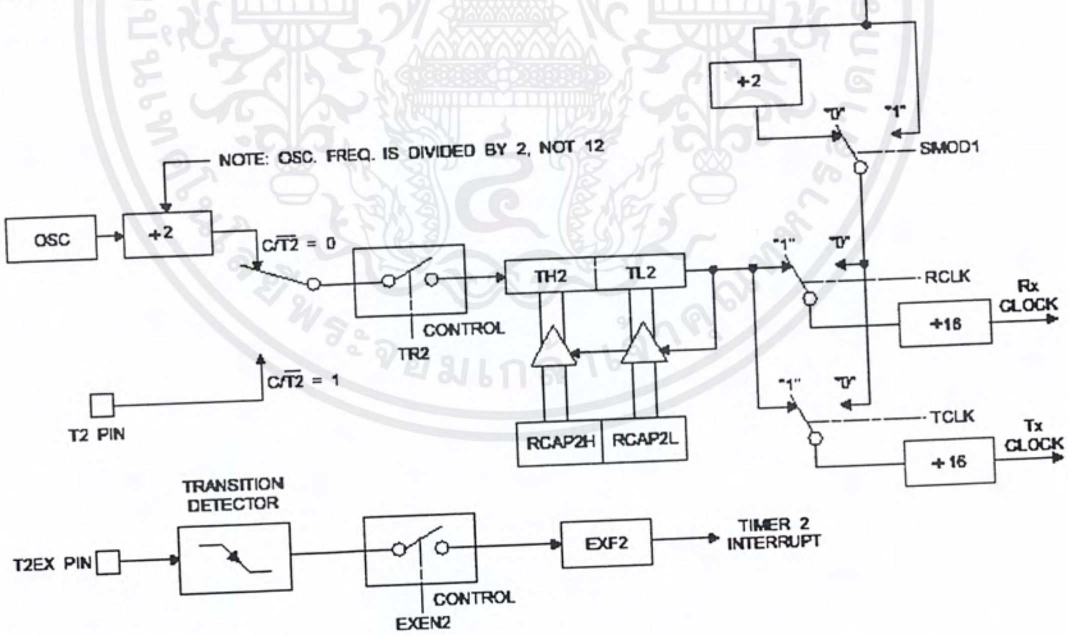



Figure 4. Timer 2 in Baud Rate Generator Mode



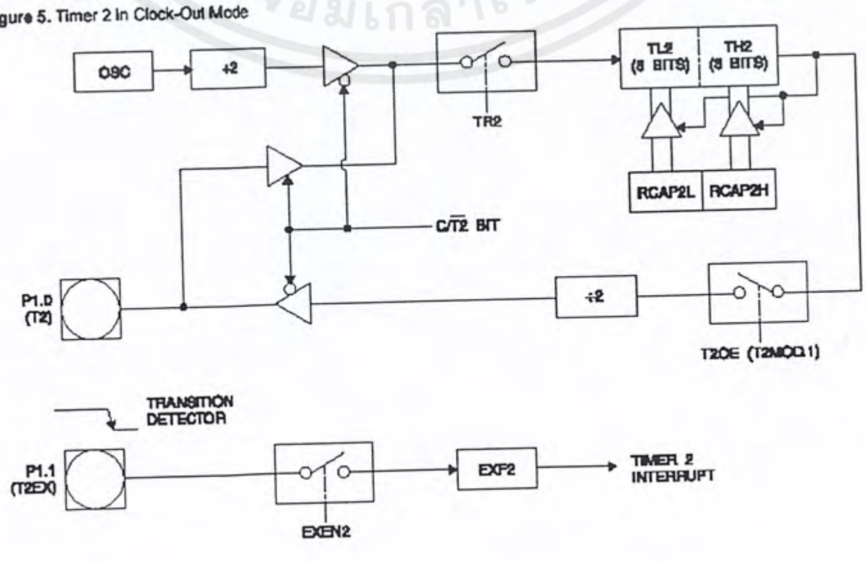

## Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4. The baud rate generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16 bit value in registers RCAP2H and RCAP2L, which are preset by software. The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation. The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation (CP/T2 = 0). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below. Modes 1 and 3 Baud Rates= Timer 2 Overflow Rate/16 where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16 bit unsigned integer. Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if XEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt. Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers. Modes 1 and 3/Baud Rate =Oscillator

Figure 5. Timer 2 in Clock-Out Mode



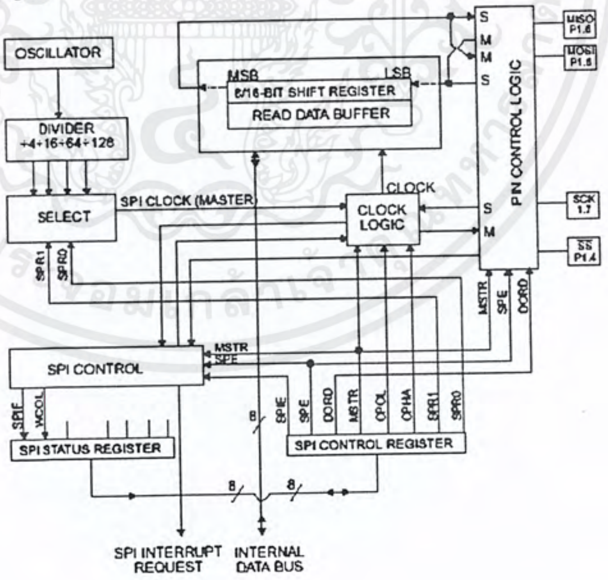

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/0 pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency. To configure the Timer/Counter 2 as a clock generator, bit C/T2 (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer. The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation. Clock Out Frequency Oscillator  In the clock-out mode, Timer 2 rollovers will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously.

**Note**, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

**Figure 6.** SPI Block Diagram




## UART

The UART in the AT89S8252 operates the same way as the UART in the AT89C51, T89C52 and AT89C55. For further information, see the October 1995 Micro controller Data Book, page 2-49, section titled, "Serial Interface."

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ.ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Serial Peripheral Interface

The serial peripheral interface (SPI) allows high-speed synchronous data transfer between the AT89S8252 and peripheral devices or between several AT89S8252 devices. The AT89S8252 SPI features include the following:

- Full-Duplex, 3-Wire Synchronous Data Transfer
- Master or Slave Operation
- 1.5-MHz Bit Frequency (max.)
- LSB First or MSB First Data Transfer
- Four Programmable Bit Rates
- End of Transmission Interrupt Flag
- Write Collision Flag Protection
- Wakeup from Idle Mode (Slave Mode Only)

The interconnection between master and slave CPUs with SPI is shown in the following figure. The SCK pin is the clock output in the master mode but is the clock input in the slave mode. Writing to the SPI data register of the master CPU starts the SPI clock generator, and the data written shifts out of the MOSI pin and into the MOSI pin of the slave CPU. After shifting one byte, the SPI clock generator stops, setting the end of transmission flag (SPIF). If both the SPI interrupt enable bit (SPIE) and the serial port interrupt enable bit (ES) are set, an interrupt is requested. The Slave Select input, SS/P1.4, is set low to select an individual SPI device as a slave. When SS/P1.4 is set high, the SPI port is deactivated and the MOSI/P1.5 pin can be used as an input. There are four combinations of SCK phase and polarity with respect to serial data, which are determined by control bits CPHA and CPOL. The SPI data transfer formats are shown in Figures 8 and 9.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 7. SPI Master-Slave Interconnection

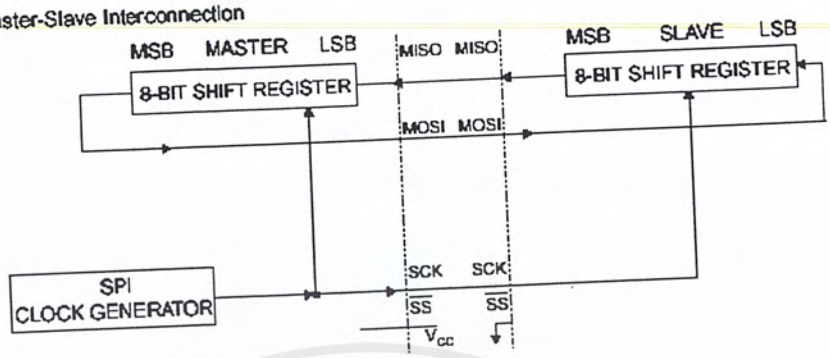
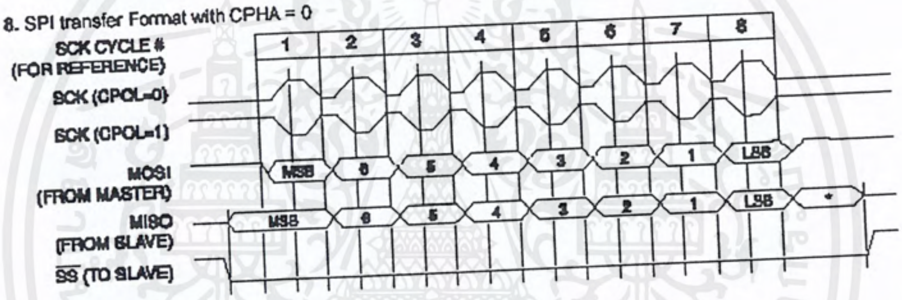
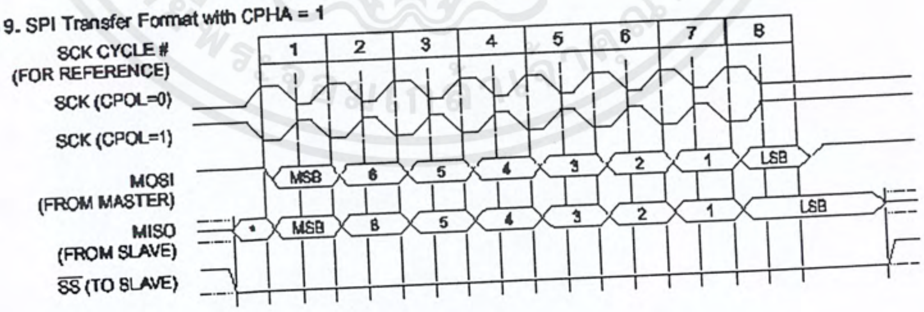



Figure 8. SPI transfer Format with CPHA = 0




*Not defined but normally MSB of character just received

Figure 9. SPI Transfer Format with CPHA = 1




*Not defined but normally LSB of previously transmitted character

## Interrupts

The AT89S8252 has a total of six interrupt vectors: two external interrupts (INT0 and INT1), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 10. Each of these interrupt sources can be individually enabled or disabled

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 10. Interrupt Enable (IE) Register

(MSB) (LSB)

| EA | — | ET2 | ES | ET1 | EX1 | ET0 | EX0 |
|---|---|---|---|---|---|---|---|

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

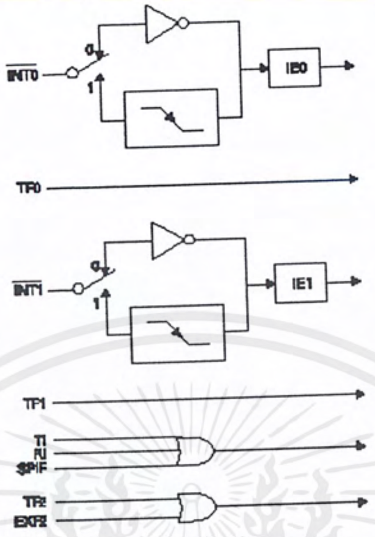
| Symbol | Position | Function |
|---|---|---|
| EA | IE.7 | Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit. |
| — | IE.6 | Reserved. |
| ET2 | IE.5 | Timer 2 interrupt enable bit. |
| ES | IE.4 | SPI and UART interrupt enable bit. |
| ET1 | IE.3 | Timer 1 interrupt enable bit. |
| EX1 | IE.2 | External interrupt 1 enable bit. |
| ET0 | IE.1 | Timer 0 interrupt enable bit. |
| EX0 | IE.0 | External interrupt 0 enable bit. |

User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.

by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once. Note that Table 10 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products. Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software. The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 10. Interrupt Sources




## Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 11. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 12. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maxi-mum voltage high and low time specifications must be observed. **Idle**

Figure 12. External Clock Drive Configuration

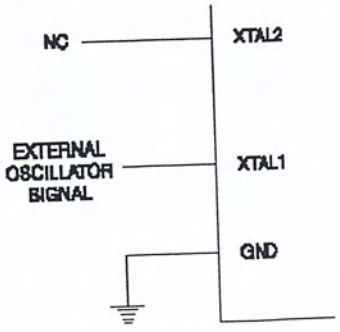
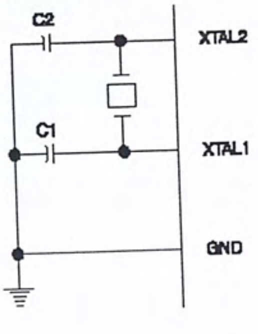



Figure 11. Oscillator Connections




Note: Note: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Mode

In idle mode, the CPU puts itself to sleep while all the onchipperipherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset. Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

## Power Down Mode

In the power down mode, the oscillator is stopped and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. Exit from power down can be initiated either by a hardware reset or by an enabled external interrupt. Reset redefines the SFR but does not change the on-chip RAM. The reset should not be activated before vcc is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

**Status of External Pins During Idle and Power Down Modes**

| Mode | Program Memory | ALE | PSEN | PORT0 | PORT1 | PORT2 | PORT3 |
|---|---|---|---|---|---|---|---|
| Idle | Internal | 1 | 1 | Data | Data | Data | Data |
| Idle | External | 1 | 1 | Float | Data | Address | Data |
| Power Down | Internal | 0 | 0 | Data | Data | Data | Data |
| Power Down | External | 0 | 0 | Float | Data | Data | Data |

To exit power down via an interrupt, the external interrupt must be enabled as level sensitive before entering power down. The interrupt service routine starts at 16 ms (nominal) after the enabled interrupt pin is activated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Program Memory Lock Bits

The AT89S8252 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table. When lock bit 1 is programmed, the logic level at the EA pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of EA must agree with the current logic level at that pin in order for the device to function properly. Once programmed, the lock bits can only be unprogrammed with the Chip Erase operations in either the parallel or serial modes.

**Lock Bit Protection Modes(1)(2)**

| Program Lock Bits | | | | Protection Type |
|---|---|---|---|---|
| | LB1 | LB2 | LB3 | |
| 1 | U | U | U | No internal memory lock feature. |
| 2 | P | U | U | MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory. $\overline{EA}$ is sampled and latched on reset and further programming of the Flash memory (parallel or serial mode) is disabled. |
| 3 | P | P | U | Same as Mode 2, but parallel or serial verify are also disabled. |
| 4 | P | P | P | Same as Mode 3, but external execution is also disabled. |

Notes: 1. U = Unprogrammed
2. P = Programmed

## Programming the Flash and EEPROM

Atmel's AT89S8252 Flash Microcontroller offers 8K bytes of in-system reprogrammable Flash Code memory and 2K bytes of EEPROM Data memory. The AT89S8252 is normally shipped with the on-chip Flash Code and EEPROM Data memory arrays in the erased state (i.e. contents = FFH) and ready to be programmed. This device supports a High-Voltage (12V) Parallel programming mode and a Low-Voltage (5V) Serial programming mode. The serial programming mode provides a convenient way to download the AT89S8252 inside the user's system. The parallel programming mode is compatible with conventional third party Flash or EPROM programmers. The Code and Data memory arrays are mapped via separate address spaces in the serial programming mode. In the parallel programming mode, the two arrays occupy one contiguous address space: 0000H to 1FFFH for the Code array and 2000H to 27FFH for the Data array. The Code and Data memory arrays on the AT89S8252 are programmed byte-by-byte in either programming mode. An auto-erase cycle is provided with the self-timed programming

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

operation in the serial programming mode. There is no need to perform the Chip Erase operation to reprogram any memory location in the serial programming mode unless any of the lock bits have been programmed. In the parallel programming mode, there is no auto-erase cycle. To reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays.

## Parallel Programming Algorithm

To program and verify the AT89S8252 in the parallel programming mode, the following sequence is recommended:

1. Power-up sequence: Apply power between $V_{cc}$ and GND pins. Set RST pin to "H". Apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Set PSEN pin to "L" ALE pin to "H" EA pin to "H" and all other pins to "H".

3. Apply the appropriate combination of "H" or "L" logic levels to pins P2.6, P2.7, P3.6, P3.7 to select one of the programming operations shown in the Flash Programming Modes table.

4. Apply the desired byte address to pins P1.0 to P1.7 and P2.0 to P2.5. Apply data to pins P0.0 to P0.7 for Write Code operation.

5. Raise EA/$_{VPP}$ to 12V to enable Flash programming, erase or verification.

6. Pulse ALE/PROG once to program a byte in the Code memory array, the Data memory array or the lock bits. The byte-write cycle is self-timed and typically takes 1.5 ms.

7. To verify the byte just programmed, bring pin P2.7 to "L" and read the programmed data at pins P0.0 to P0.7.

8. Repeat steps 3 through 7 changing the address and data for the entire 2K or 8K bytes array or until the end of the object file is reached.

9. Power-off sequence: Set XTAL1 to "L". Set RST and EA pins to "L". Tu r n $V_{cc}$ power off. In the parallel programming mode, there is no auto-erase cycle and to reprogram any non-blank byte, the user needs to use the Chip Erase operation first to erase both arrays. **DATA Polling**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89S8252 features DATA Polling to indicate the end of a write cycle. During a write cycle in the parallel or serial programming mode, an attempted read of the last byte writ-ten will result in the complement of the written datum on P0.7 (parallel mode), and on the MSB of the serial output byte on MISO (serial mode). Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. DATA Polling may begin any time after a write cycle has been initiated. **Ready/Busy** The progress of byte programming in the parallel programming mode can also be monitored by the RDY/BSY output signal. Pin P3.4 is pulled Low after ALE goes High during programming to indicate BUSY. P3.4 is pulled High again when programming is done to indicate READY.

## Program Verify

If lock bits LB1 and LB2 have not been programmed, the programmed Code or Data byte can be read back via the address and data lines for verification. The state of the lock bits can also be verified directly in the parallel programming mode. In the serial programming mode, the state of the lock bits can only be verified indirectly by observing that the lock bit features are enabled.

## Chip Erase

Both Flash and EEPROM arrays are erased electrically at the same time. In the parallel programming mode, chip erase is initiated by using the proper combination of control signals and by holding ALE/PROG low for 10 ms. The Code and Data arrays are written with all "1"s in the Chip Erase operation. In the serial programming mode, a chip erase operation is initiated by issuing the Chip Erase instruction. In this mode, chip erase is self-timed and takes about 16 ms. During chip erase, a serial read from any address location will return 00H at the data outputs.

## Serial Programming Fuse

A programmable fuse is available to disable Serial Programming if the user needs maximum system security. The Serial Programming Fuse can only be programmed or erased in the Parallel Programming Mode. The AT89S8252 is shipped with the Serial Programming Mode enabled. **Reading the Signature Bytes:** The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 must be pulled to a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

logic low. The values returned are as follows: (030H) = 1EH indicates manufactured by Atmel (031H) = 72H indicates 89S8252 **Programming Interface** Every code byte in the Flash and EEPROM arrays can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion. All major programming vendors offer worldwide support for the Atmel micro controller series. Please contact your local programming vendor for the appropriate software revision.

## Serial Downloading

Both the Code and Data memory arrays can be programmed using the serial SPI bus while RST is pulled to VCC. The serial interface consists of pins SCK, MOSI (input) and MISO (output). After RST is set high, the Programming Enable instruction needs to be executed first before pro-gram/ erase operations can be executed. An auto-erase cycle is built into the self-timed programming operation (in the serial mode ONLY) and there is no need to first execute the Chip Erase instruction unless any of the lock bits have been programmed. The Chip Erase operation turns the content of every memory location in both the Code and Data arrays into FFH. The Code and Data memory arrays have separate address spaces: 0000H to 1FFFH for Code memory and 000H to 7FFH for Data memory. Either an external system clock is supplied at pin XTAL1 or a crystal needs to be connected across pins XTAL1 and XTAL2. The maximum serial clock (SCK) frequency should be less than 1/40 of the crystal frequency. With a 24 MHz oscillator clock, the maximum SCK frequency is 600 kHz.

## Serial Programming Algorithm

To program and verify the AT89S8252 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence: Apply power between $V_{cc}$ and GND pins. Set RST pin to "H". If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 24 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 40.

3. The Code or Data array is programmed one byte at a time by supplying the address and data together with the appropriate Write instruction. The selected memory location is first automatically erased before new data is written. The write cycle is self-timed and typically takes less than 2.5 ms at 5V.

4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.

5. At the end of a programming session, RST can be set low to commence normal operation. Power-off sequence (if needed): Set XTAL1 to "L" (if a crystal is not used). Set RST to "L". Tu r n V cc power off. **Serial Programming Instruction** The Instruction Set for Serial Programming follows a 3-byte protocol and is shown in the following table:

**Instruction Set**

| Instruction | Input Format | | | Operation |
|---|---|---|---|---|
| | Byte 1 | Byte 2 | Byte 3 | |
| Programming Enable | 1010 1100 | 0101 0011 | xxxx xxxx | Enable serial programming interface after RST goes high. |
| Chip Erase | 1010 1100 | xxxx x100 | xxxx xxxx | Chip erase both 8K & 2K memory arrays. |
| Read Code Memory | aaaa a001 | low addr | xxxx xxxx | Read data from Code memory array at the selected address. The 5 MSBs of the first byte are the high order address bits. The low order address bits are in the second byte. Data are available at pin MISO during the third byte. |
| Write Code Memory | aaaa a010 | low addr | data in | Write data to Code memory location at selected address. The address bits are the 5 MSBs of the first byte together with the second byte. |
| Read Data Memory | 00aa a101 | low addr | xxxx xxxx | Read data from Data memory array at selected address. Data are available at pin MISO during the third byte. |
| Write Data Memory | 00aa a110 | low addr | data in | Write data to Data memory location at selected address. |
| Write Lock Bits | 1010 1100 | LB1 LB2 LB3 x x111 | xxxx xxxx | Write lock bits. Set LB1, LB2 or LB3 = "0" to program lock bits. |

Notes: 1. $\overline{\text{DATA}}$ polling is used to indicate the end of a write cycle which typically takes less than 2.5 ms at 5V.
2. "aaaaa" = high order address.
3. "x" = don't care.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flash and EEPROM Parallel Programming Modes

| Mode | RST | $\overline{PSEN}$ | ALE/$\overline{PROG}$ | $\overline{EA}/V_{PP}$ | P2.6 | P2.7 | P3.6 | P3.7 | Data I/O P0.7:0 | Address P2.5:0 P1.7:0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Serial Prog. Modes | H | h[1] | h[1] | x | | | | | | |
| Chip Erase | H | L | ‾_/‾ (2) | 12V | H | L | L | L | X | X |
| Write (10K bytes) Memory | H | L | ‾_/‾ | 12V | L | H | H | H | DIN | ADDR |
| Read (10K bytes) Memory | H | L | H | 12V | L | L | H | H | DOUT | ADDR |
| Write Lock Bits: | H | L | ‾_/‾ | 12V | H | L | H | L | DIN | X |
| Bit - 1 | | | | | | | | | P0.7 = 0 | X |
| Bit - 2 | | | | | | | | | P0.6 = 0 | X |
| Bit - 3 | | | | | | | | | P0.5 = 0 | X |
| Read Lock Bits: | H | L | H | 12V | H | H | L | L | DOUT | X |
| Bit - 1 | | | | | | | | | @P0.2 | X |
| Bit - 2 | | | | | | | | | @P0.1 | X |
| Bit - 3 | | | | | | | | | @P0.0 | X |
| Read Atmel Code | H | L | H | 12V | L | L | L | L | DOUT | 30H |
| Read Device Code | H | L | H | 12V | L | L | L | L | DOUT | 31H |
| Serial Prog. Enable | H | L | ‾_/‾ (2) | 12V | L | H | L | H | P0.0 = 0 | X |
| Serial Prog. Disable | H | L | ‾_/‾ (2) | 12V | L | H | L | H | P0.0 = 1 | X |
| Read Serial Prog. Fuse | H | L | H | 12V | H | H | L | H | @P0.0 | X |

Notes:
1. "h" = weakly pulled "High" internally.
2. Chip Erase and Serial Programming Fuse require a 10-ms $\overline{PROG}$ pulse. Chip Erase needs to be performed first before reprogramming any byte with a content other than FFH.
3. P3.4 is pulled Low during programming to indicate RDY/$\overline{BSY}$.
4. "X" = don't care

**Figure 14.** Programming the Flash/EEPROM Memory

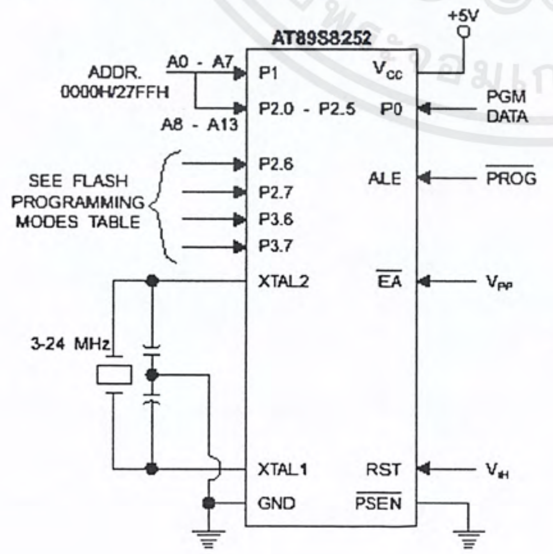
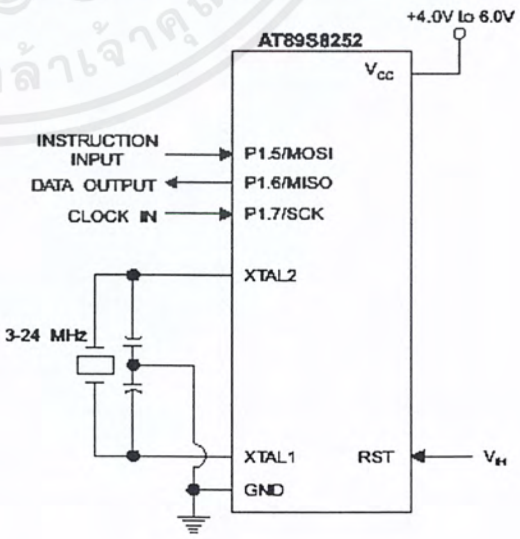
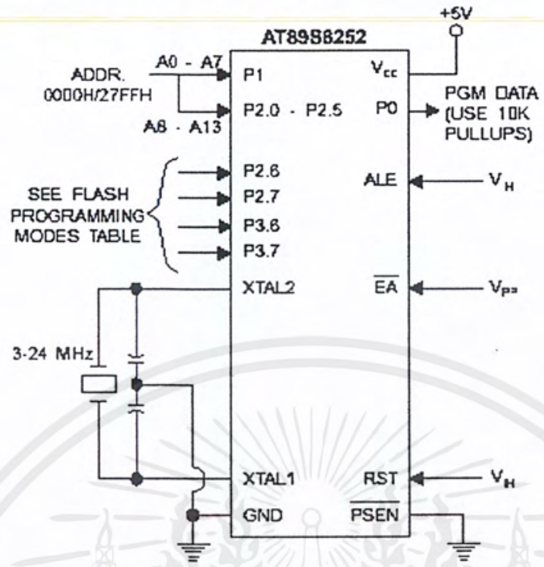



**Figure 15.** Flash/EEPROM Serial Downloading




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 16. Verifying the Flash/EEPROM Memory



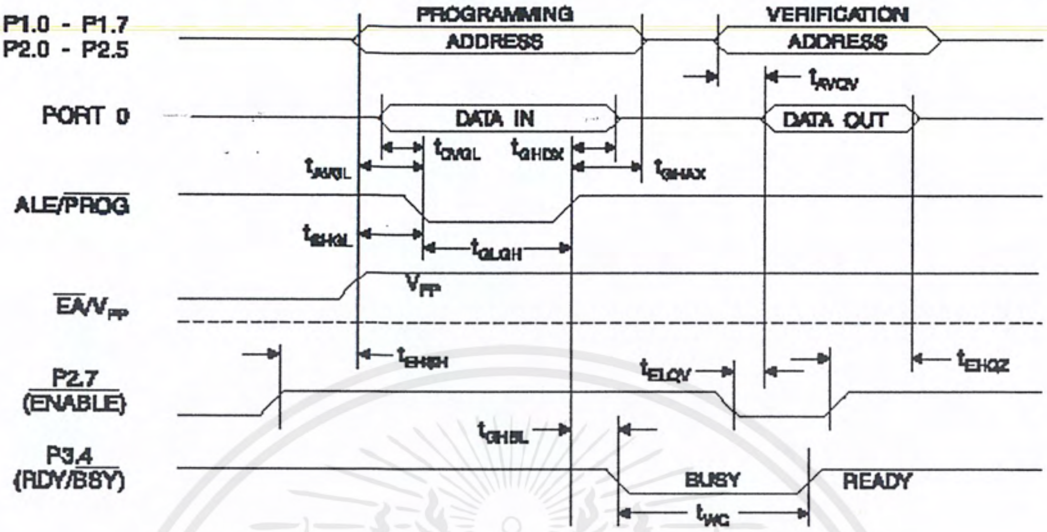

## Flash Programming and Verification Characteristics-Parallel Mode

$T_A$ = 0°C to 70°C, $V_{CC}$ = 5.0V ± 10%

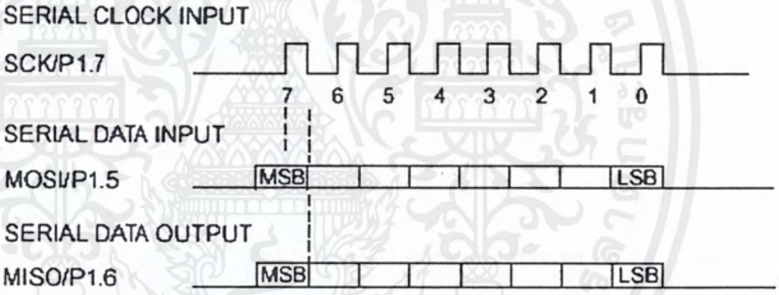
| Symbol | Parameter | Min | Max | Units |
|---|---|---|---|---|
| $V_{PP}$ | Programming Enable Voltage | 11.5 | 12.5 | V |
| $I_{PP}$ | Programming Enable Current | | 1.0 | mA |
| $1/t_{CLCL}$ | Oscillator Frequency | 3 | 24 | MHz |
| $t_{AVGL}$ | Address Setup to $\overline{PROG}$ Low | $48t_{CLCL}$ | | |
| $t_{GHAX}$ | Address Hold After $\overline{PROG}$ | $48t_{CLCL}$ | | |
| $t_{DVGL}$ | Data Setup to $\overline{PROG}$ Low | $48t_{CLCL}$ | | |
| $t_{GHDX}$ | Data Hold After $\overline{PROG}$ | $48t_{CLCL}$ | | |
| $t_{EHSH}$ | P2.7 ($\overline{ENABLE}$) High to $V_{PP}$ | $48t_{CLCL}$ | | |
| $t_{SHGL}$ | $V_{PP}$ Setup to $\overline{PROG}$ Low | 10 | | µs |
| $t_{GLGH}$ | $\overline{PROG}$ Width | 1 | 110 | µs |
| $t_{AVQV}$ | Address to Data Valid | | $48t_{CLCL}$ | |
| $t_{ELQV}$ | $\overline{ENABLE}$ Low to Data Valid | | $48t_{CLCL}$ | |
| $t_{EHQZ}$ | Data Float After $\overline{ENABLE}$ | 0 | $48t_{CLCL}$ | |
| $t_{GHBL}$ | $\overline{PROG}$ High to $\overline{BUSY}$ Low | | 1.0 | µs |
| $t_{WC}$ | Byte Write Cycle Time | | 2.0 | ms |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Flash/EEPROM Programming and Verification Waveforms - Parallel Mode




## Serial Downloading Waveforms




### Absolute Maximum Ratings*

Operating Temperature................................ -55°C to +125°C

Storage Temperature .................................... -65°C to +150°C

### Voltage on Any Pin

with Respect to Ground ....................................-1.0V to +7.0V

Maximum Operating Voltage............................................ 6.6V

DC Output Current...................................................... 15.0 mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause ermanent dam-age to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

## DC Characteristics

The values shown in this table are valid for $T_A$ = -40°C to 85°C and $V_{CC}$ = 5.0V ± 20%, unless otherwise noted.

| Symbol | Parameter | Condition | Min | Max | Units |
|---|---|---|---|---|---|
| $V_{IL}$ | Input Low Voltage | (Except $\overline{EA}$) | -0.5 | 0.2 $V_{CC}$ - 0.1 | V |
| $V_{IL1}$ | Input Low Voltage ($\overline{EA}$) | | -0.5 | 0.2 $V_{CC}$ - 0.3 | V |
| $V_{IH}$ | Input High Voltage | (Except XTAL1, RST) | 0.2 $V_{CC}$ + 0.9 | $V_{CC}$ + 0.5 | V |
| $V_{IH1}$ | Input High Voltage | (XTAL1, RST) | 0.7 $V_{CC}$ | $V_{CC}$ + 0.5 | V |
| $V_{OL}$ | Output Low Voltage (1) (Ports 1,2,3) | $I_{OL}$ = 1.6 mA | | 0.5 | V |
| $V_{OL1}$ | Output Low Voltage (1) (Port 0, ALE, $\overline{PSEN}$) | $I_{OL}$ = 3.2 mA | | 0.5 | V |
| $V_{OH}$ | Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$) | $I_{OH}$ = -60 µA, $V_{CC}$ = 5V ± 10% | 2.4 | | V |
| | | $I_{OH}$ = -25 µA | 0.75 $V_{CC}$ | | V |
| | | $I_{OH}$ = -10 µA | 0.9 $V_{CC}$ | | V |
| $V_{OH1}$ | Output High Voltage (Port 0 in External Bus Mode) | $I_{OH}$ = -800 µA, $V_{CC}$ = 5V ± 10% | 2.4 | | V |
| | | $I_{OH}$ = -300 µA | 0.75 $V_{CC}$ | | V |
| | | $I_{OH}$ = -80 µA | 0.9 $V_{CC}$ | | V |
| $I_{IL}$ | Logical 0 Input Current (Ports 1,2,3) | $V_{IN}$ = 0.45V | | -50 | µA |
| $I_{TL}$ | Logical 1 to 0 Transition Current (Ports 1,2,3) | $V_{IN}$ = 2V, $V_{CC}$ = 5V ± 10% | | -650 | µA |
| $I_{LI}$ | Input Leakage Current (Port 0, EA) | 0.45 < $V_{IN}$ < $V_{CC}$ | | ±10 | µA |
| RRST | Reset Pulldown Resistor | | 50 | 300 | KΩ |
| $C_{IO}$ | Pin Capacitance | Test Freq. = 1 MHz, $T_A$ = 25°C | | 10 | pF |
| $I_{CC}$ | Power Supply Current | Active Mode, 12 MHz | | 25 | mA |
| | | Idle Mode, 12 MHz | | 6.5 | mA |
| | Power Down Mode (2) | $V_{CC}$ = 6V | | 100 | µA |
| | | $V_{CC}$ = 3V | | 40 | µA |

Notes: 1. Under steady state (non-transient) conditions, I OL must be externally limited as follows: Maximum I OL per port pin: 10 mA Maximum I OL per 8-bit port: Port 0: 26 mA Ports 1, 2, 3: 15 mA Maximum total I OL for all output pins: 71 mA If I OL exceeds the test condition, V OL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions. 2. Minimum V CC for Power Down is 2V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AC Characteristics

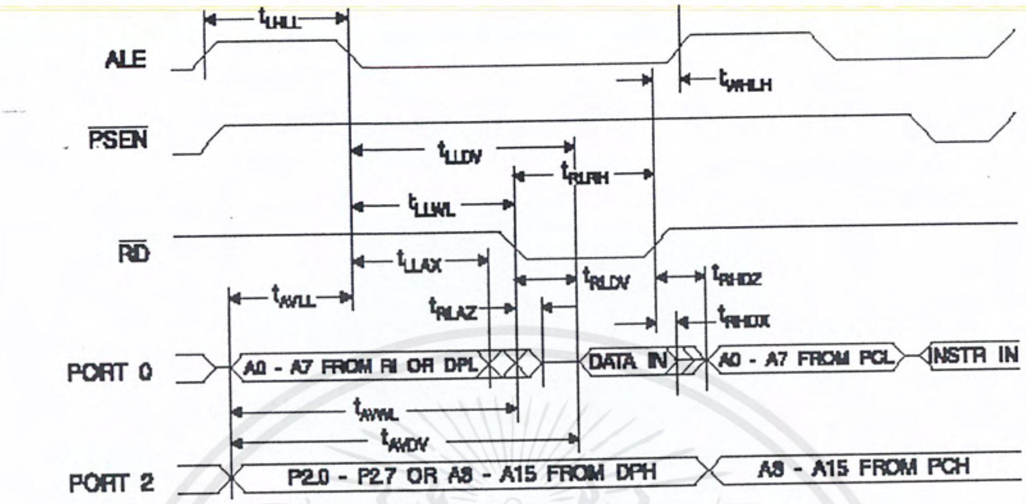
Under operating conditions, load capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; load capacitance for all other outputs = 80 pF.

### External Program and Data Memory Characteristics

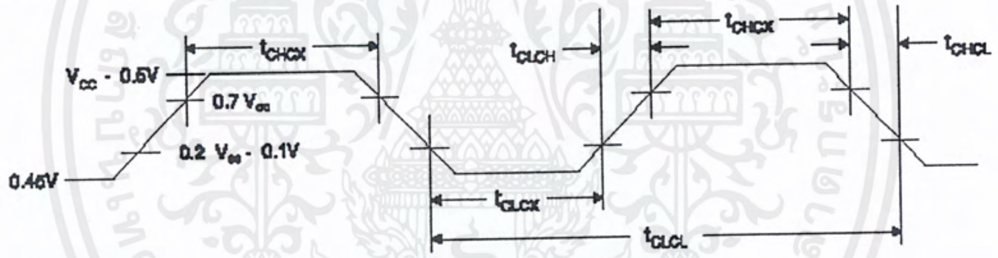
| Symbol | Parameter | Variable Oscillator | | Units |
|---|---|---|---|---|
| | | Min | Max | |
| $1/t_{CLCL}$ | Oscillator Frequency | 0 | 24 | MHz |
| $t_{LHLL}$ | ALE Pulse Width | $2t_{CLCL}$ - 40 | | ns |
| $t_{AVLL}$ | Address Valid to ALE Low | $t_{CLCL}$ - 13 | | ns |
| $t_{LLAX}$ | Address Hold After ALE Low | $t_{CLCL}$ - 20 | | ns |
| $t_{LLIV}$ | ALE Low to Valid Instruction In | | $4t_{CLCL}$ - 65 | ns |
| $t_{LLPL}$ | ALE Low to $\overline{PSEN}$ Low | $t_{CLCL}$ - 13 | | ns |
| $t_{PLPH}$ | $\overline{PSEN}$ Pulse Width | $3t_{CLCL}$ - 20 | | ns |
| $t_{PLIV}$ | $\overline{PSEN}$ Low to Valid Instruction In | | $3t_{CLCL}$ - 45 | ns |
| $t_{PXIX}$ | Input Instruction Hold After $\overline{PSEN}$ | 0 | | ns |
| $t_{PXIZ}$ | Input Instruction Float After $\overline{PSEN}$ | | $t_{CLCL}$ - 10 | ns |
| $t_{PXAV}$ | $\overline{PSEN}$ to Address Valid | $t_{CLCL}$ - 8 | | ns |
| $t_{AVIV}$ | Address to Valid Instruction In | | $5t_{CLCL}$ - 55 | ns |
| $t_{PLAZ}$ | $\overline{PSEN}$ Low to Address Float | | 10 | ns |
| $t_{RLRH}$ | $\overline{RD}$ Pulse Width | $6t_{CLCL}$ - 100 | | ns |
| $t_{WLWH}$ | $\overline{WR}$ Pulse Width | $6t_{CLCL}$ - 100 | | ns |
| $t_{RLDV}$ | $\overline{RD}$ Low to Valid Data In | | $5t_{CLCL}$ - 90 | ns |
| $t_{RHDX}$ | Data Hold After $\overline{RD}$ | 0 | | ns |
| $t_{RHDZ}$ | Data Float After $\overline{RD}$ | | $2t_{CLCL}$ - 28 | ns |
| $t_{LLDV}$ | ALE Low to Valid Data In | | $8t_{CLCL}$ - 150 | ns |
| $t_{AVDV}$ | Address to Valid Data In | | $9t_{CLCL}$ - 165 | ns |
| $t_{LLWL}$ | ALE Low to $\overline{RD}$ or $\overline{WR}$ Low | $3t_{CLCL}$ - 50 | $3t_{CLCL}$ + 50 | ns |
| $t_{AVWL}$ | Address to $\overline{RD}$ or $\overline{WR}$ Low | $4t_{CLCL}$ - 75 | | ns |
| $t_{QVWX}$ | Data Valid to $\overline{WR}$ Transition | $t_{CLCL}$ - 20 | | ns |
| $t_{QVWH}$ | Data Valid to $\overline{WR}$ High | $7t_{CLCL}$ - 120 | | ns |
| $t_{WHQX}$ | Data Hold After $\overline{WR}$ | $t_{CLCL}$ - 20 | | ns |
| $t_{RLAZ}$ | $\overline{RD}$ Low to Address Float | | 0 | ns |
| $t_{WHLH}$ | $\overline{RD}$ or $\overline{WR}$ High to ALE High | $t_{CLCL}$ - 20 | $t_{CLCL}$ + 25 | ns |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## External Data Memory Read Cycle




## External Clock Drive Waveforms

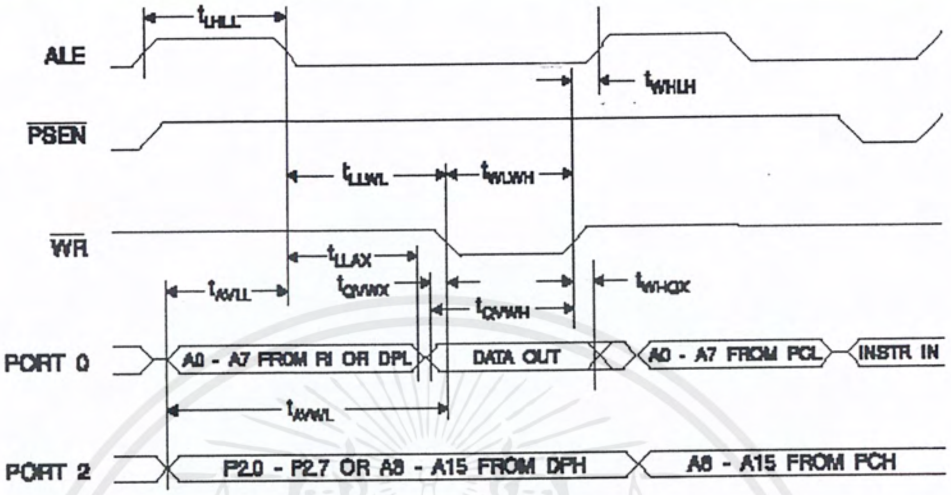



## External Clock Drive

| Symbol | Parameter | $V_{CC}$ = 4.0V to 6.0V | | Units |
|---|---|---|---|---|
| | | Min | Max | |
| $1/t_{CLCL}$ | Oscillator Frequency | 0 | 24 | MHz |
| $t_{CLCL}$ | Clock Period | 41.6 | | ns |
| $t_{CHCX}$ | High Time | 15 | | ns |
| $t_{CLCX}$ | Low Time | 15 | | ns |
| $t_{CLCH}$ | Rise Time | | 20 | ns |
| $t_{CHCL}$ | Fall Time | | 20 | ns |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## External Data Memory Write Cycle

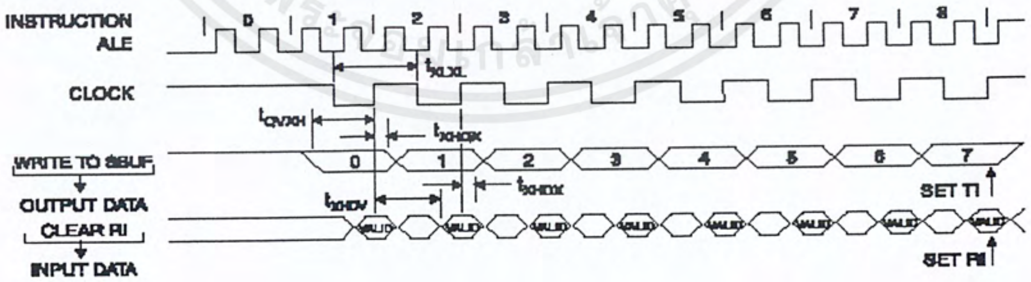



## Serial Port Timing: Shift Register Mode Test Conditions

The values in this table are valid for $V_{CC}$ = 4.0V to 6V and Load Capacitance = 80 pF.

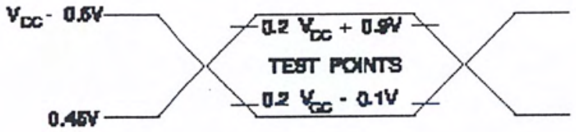
| Symbol | Parameter | Variable Oscillator | | Units |
|---|---|---|---|---|
| | | Min | Max | |
| $t_{XLXL}$ | Serial Port Clock Cycle Time | $12t_{CLCL}$ | | µs |
| $t_{QVXH}$ | Output Data Setup to Clock Rising Edge | $10t_{CLCL}$ - 133 | | ns |
| $t_{XHQX}$ | Output Data Hold After Clock Rising Edge | $2t_{CLCL}$ - 117 | | ns |
| $t_{XHDX}$ | Input Data Hold After Clock Rising Edge | 0 | | ns |
| $t_{XHDV}$ | Clock Rising Edge to Input Data Valid | | $10t_{CLCL}$ - 133 | ns |

## Shift Register Mode Timing Waveforms




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

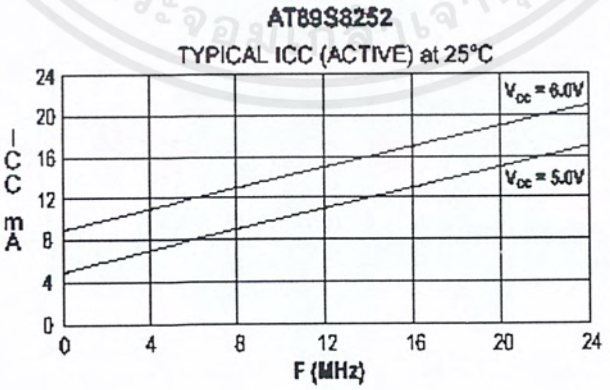
## AC Testing Input/Output Waveforms[1]




**Notes:** 1. AC Inputs during testing are driven at V CC - 0.5V for a logic 1 and 0.45V for a logic 0. Timing mea-surements are made at V IH min. for a logic 1 and V IL max. for a logic 0.

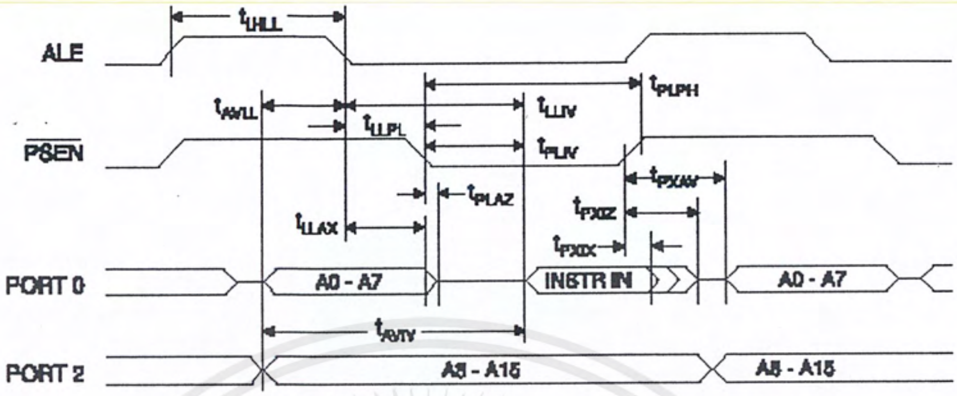
## Float Waveforms[1]

Notes: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V OH /V OL level occurs.



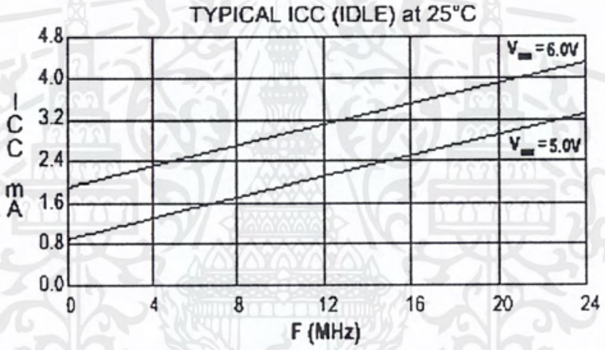

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## External Program Memory Read Cycle



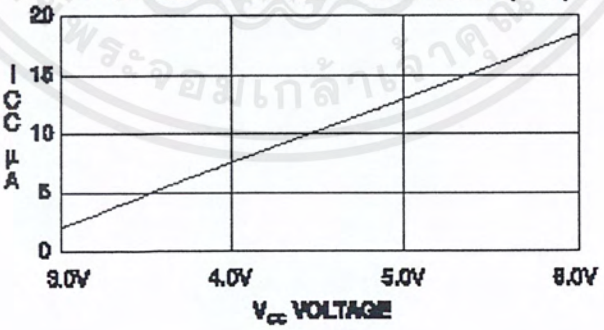

**AT89S8252**

TYPICAL ICC (IDLE) at 25°C




**AT89S8252**

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)




Notes: 1. XTAL1 tied to GND for Icc (power down)

2. Lock bits programmed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## Ordering Information

| Speed (MHz) | Power Supply | Ordering Code | Package | Operation Range |
|---|---|---|---|---|
| 16 | 4.0V to 6.0V | AT89S8252-16AA | 44A | Automotive (-40°C to 105°C) |
| | | AT89S8252-16JA | 44J | |
| | | AT89S8252-16PA | 40P6 | |
| | | AT89S8252-16QA | 44Q | |
| 24 | 4.0V to 6.0V | AT89S8252-24AC | 44A | Commercial (0°C to 70°C) |
| | | AT89S8252-24JC | 44J | |
| | | AT89S8252-24PC | 40P6 | |
| | | AT89S8252-24QC | 44Q | |
| | 4.0V to 6.0V | AT89S8252-24AI | 44A | Industrial (-40°C to 85°C) |
| | | AT89S8252-24JI | 44J | |
| | | AT89S8252-24PI | 40P6 | |
| | | AT89S8252-24QI | 44Q | |
| 33 | 4.5V to 5.5V | AT89S8252-33AC | 44A | Commercial (0°C to 70°C) |
| | | AT89S8252-33JC | 44J | |
| | | AT89S8252-33PC | 40P6 | |
| | | AT89S8252-33QC | 44Q | |

☐ = Preliminary Information

| Package Type | |
|---|---|
| 44A | 44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP) |
| 44J | 44 Lead, Plastic J-Leaded Chip Carrier (PLCC) |
| 40P6 | 40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP) |
| 44Q | 44 Lead, Plastic Gull Wing Quad Flatpack (PQFP) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บรรณานุกรม

1. โยธิน เปรมปราณีรัชต์. **ระบบเซอร์โวและอิเล็กทรอนิกส์คอนโทรลเลอร์**. ภาควิชาวิศวกรรมระบบควบคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

2. Alexander Kusko, **Solid-State DC. Motor Drives**, The MIT Press, 1969.

3. รศ.สมยศ จุณณะปียะ. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-8051**. กรุงเทพฯ. บริษัทอินโวตีฟ เอ็กเพอริเมนต์ จำกัด.กรุงเทพฯ. 2533

4. ธีรวัฒน์ ประกอบผล. **การประยุกต์ใช้งานไมโครคอนโทรลเลอร์**. กรุงเทพฯ. บริษัทประชาชน จำกัด. กรุงเทพฯ. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง

| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายเทพบดินทร์ บริรักษ์อราวินท์ |
| วันเดือนปีเกิด | วันที่ 24 เมษายน พ.ศ. 2522 |
| สถานที่เกิด | โรงพยาบาลนครคริสเตียน |
| ภูมิลำเนาเดิม | บ้านเลขที่ 11/1 หมู่ 3 ตำบลโพธิ์เสด็จ อำเภอ เมือง จังหวัดนครศรีธรรมราช |
| ที่อยู่ปัจจุบัน | บ้านเลขที่ 11/1 หมู่ 3 ตำบลโพธิ์เสด็จ อำเภอ เมือง จังหวัดนครศรีธรรมราช |
| โทรศัพท์ | 075-347196 |

## ประวัติการศึกษา

| | |
|---|---|
| ประถมศึกษา | โรงเรียนศรีธรรมราชศึกษา |
| มัธยมศึกษาตอนต้น | โรงเรียนเบญจมราชูทิศ |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | วิทยาลัยเทคนิคนครศรีธรรมราช |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคนครศรีธรรมราช |
| ปริญญาตรี | สาขาวิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชา ครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |
| ทุนการศึกษา | 1. ได้รับทุนยกเว้นค่าหน่วยกิจ ปีพ.ศ. 2542<br>2. ได้รับทุนงบประมาณ ปีพ.ศ. 2543 |
| คติพจน์ | ศักดิ์ศรีสำคัญยิ่งกว่าสิ่งอื่นใดและจงใช้สติปัญญานำทางเพื่อรักษาศักดิ์ศรีนั้นไว้ให้จงได้ |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง

| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายนภดล ไชยบัณฑิต |
| วันเดือนปีเกิด | วันที่ 2 มกราคม พ.ศ. 2521 |
| สถานที่เกิด | โรงพยาบาลพัทลุง |
| ภูมิลำเนาเดิม | บ้านเลขที่ 49 หมู่ 6 ตำบลเขาเจียก อำเภอเมือง จังหวัดพัทลุง |
| ที่อยู่ปัจจุบัน | บ้านเลขที่ 49 หมู่ 6 ตำบลเขาเจียก อำเภอ เมือง จังหวัดพัทลุง |
| โทรศัพท์ | 074-620071 |

## ประวัติการศึกษา

| | |
|---|---|
| ประถมศึกษา | โรงเรียนเทศบาลบ้านคูหาสวรรค์ |
| มัธยมศึกษาตอนต้น | โรงเรียนพัทลุง |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | วิทยาลัยเทคนิคพัทลุง |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | วิทยาลัยเทคนิคสัตหีบ |
| ปริญญาตรี | สาขาวิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม ภาควิชา ครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |
| ทุนการศึกษา | - |
| คติพจน์ | เวลาและกระแสน้ำไม่เคยคอยใคร |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง

| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายยุทธนากร บุญศิลป์ |
| วันเดือนปีเกิด | วันที่ 18 สิงหาคม พ.ศ. 2522 |
| สถานที่เกิด | จังหวัดลพบุรี |
| ภูมิลำเนาเดิม | บ้านเลขที่ 176/2 หมู่ 12 ตำบลธารเกษม<br>อำเภอ พระพุทธบาท จังหวัดสระบุรี |
| ที่อยู่ปัจจุบัน | บ้านเลขที่ 124 หมู่ 8 ตำบลห้วยเหนือ<br>อำเภอ ขุขันธ์ จังหวัดศรีสะเกษ |
| โทรศัพท์ | 045-671375 |

## ประวัติการศึกษา

| | |
|---|---|
| ประถมศึกษา | โรงเรียนเมืองใหม่ (ชลอราษฎ์รังสฤษ) |
| มัธยมศึกษาตอนต้น | โรงเรียนพระนารายณ์ |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | วิทยาลัยเทคนิคศรีสะเกษ |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | สถาบันเทคโนโลยีราชมงคล<br>วิทยาเขตพระนครเหนือ |
| ปริญญาตรี | สาขาวิชา เทคโนโลยีการวัดคุม<br>ทางอุตสาหกรรม<br>ภาควิชา ครุศาสตร์วิศวกรรม<br>คณะครุศาสตร์อุตสาหกรรม |
| ทุนการศึกษา | ได้รับทุนงบประมาณ ปีพ.ศ. 2543 |
| คติพจน์ | ไม่นอนตื่นสาย ไม่อายทำกิน<br>ไม่หมิ่นเงินน้อย ไม่คอยวาสนา |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้แต่ง

| | |
|---|---|
| ชื่อผู้ทำปริญญานิพนธ์ | นายอภิชาติ สุจริต |
| วันเดือนปีเกิด | วันที่ 11 ตุลาคม พ.ศ. 2521 |
| สถานที่เกิด | โรงพยาบาลจังหวัดชลบุรี |
| ภูมิลำเนาเดิม | บ้านเลขที่ 58-60 ถนนเกาะแก้ว อำเภอ พนัสนิคม จังหวัดชลบุรี |
| ที่อยู่ปัจจุบัน | บ้านเลขที่ 58-60 ถนนเกาะแก้ว อำเภอ พนัสนิคม จังหวัดชลบุรี |
| โทรศัพท์ | - |

## ประวัติการศึกษา

| | |
|---|---|
| ประถมศึกษา | โรงเรียนวัฒนานุศาสตร์ |
| มัธยมศึกษาตอนต้น | โรงเรียนพนัสพิทยาคาร |
| ประกาศนียบัตรวิชาชีพ (ปวช.) | เทคโนโลยีภาคตะวันออก (อีเทค) |
| ประกาศนียบัตรวิชาชีพชั้นสูง (ปวส.) | เทคโนโลยีภาคตะวันออก (อีเทค) |
| ปริญญาตรี | สาขาวิชา เทคโนโลยีการวัดคุม ทางอุตสาหกรรม ภาควิชา ครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม |
| ทุนการศึกษา | - |
| คติพจน์ | ทำวันนี้ให้ดีที่สุด |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้