



ระบบจำลองการเคลื่อนที่ของลิฟต์
(SIMULATED MOTION OF LIFT)



โดย

นางสาวธิภาพรรณ แซ่ลิ

นางสาวภรณ์ เพชรจอม

นางสาวโยทกา ชมภูศรี

| |
|------------------------------------|
| วัน เดือน ปี..... 11. ๑๒. 2541 |
| เลขทะเบียน..... 038874 |
| เลขเรียกหนังสือ... J ๒๐118 น ๒๗๓ ๖ |

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ **038874** คำ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจำลองการเคลื่อนที่ของลิฟต์
(SIMULATED MOTION OF LIFT)

โดย

นางสาวธิภาพรรณ แซ่ลี รหัสประจำตัว 37014174
นางสาวภรณ์ เพชรจอม รหัสประจำตัว 37014312
นางสาวโยทกา ชมภูศรี รหัสประจำตัว 37014347

อาจารย์ที่ปรึกษา

ผศ.ดร. เกียรติศักดิ์ คมวัชระ

อาจารย์ สว่าง เลิศถิรสุนทร

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจำลองการเคลื่อนที่ของลิฟต์
(SIMULATED MOTION OF LIFT)

โดย

นางสาวธิภาพรรณ แซ่ลี

นางสาวภรณ์ เพชรจอม

นางสาวโยทกา ชมภูศรี

อาจารย์ที่ปรึกษา

ผศ.ดร.เกียรติศักดิ์ คมวัชระ

อาจารย์ส่วาง เลิศดิทรสุนทร

บทคัดย่อ

ปฏิญานิพนธ์นี้ได้นำเสนอการพัฒนาการประยุกต์ใช้งานไมโครคอนโทรลเลอร์ 89C51 ในการควบคุมระบบจำลองการเคลื่อนที่ของลิฟต์ ให้สามารถเคลื่อนที่ขึ้นลงตามคำสั่งที่ได้รับจากภายนอก โดยไมโครคอนโทรลเลอร์ทำหน้าที่เป็นตัวประมวลผลการเคลื่อนที่ขึ้นลงตามชั้นของลิฟต์ซึ่งนำสัญญาณจากเอนโคเดอร์ของมอเตอร์ที่ทำหน้าที่ขับเคลื่อนตัวลิฟต์ มาประมวลผลระยะทางการเคลื่อนที่ให้ความถูกต้อง นอกจากนี้ไมโครคอนโทรลเลอร์ยังทำหน้าที่ควบคุมระบบจำลองให้สามารถเปิด-ปิด ประตูลิฟต์รวมทั้งแสดงผลตำแหน่งเมื่อลิฟต์เคลื่อนที่ไปถึงชั้นใดๆได้

ABSTRACT

According to this project, the 89C51 microcontroller is applied to control the simulation of the lift motion system in order to make it have the capabilities to present upward or downward movements that belong to the input from the outer part. The 89C51 can process these movements by receiving the code signals from the lift drive motor. Furthermore, the microcontroller also controls the system to open the lift door and display lift position, once it reaches any floor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบพระคุณ ผศ.ดร.เกียรติศักดิ์ คมวัชระ อาจารย์สว่าง เลิศภิรตสุนทร
อาจารย์สุมิตร พนาอุดมทรัพย์ และอาจารย์ทุกๆท่านที่กรุณาให้คำปรึกษา แนวคิด
ตลอดจนแนวทางการแก้ไขปัญหาที่เกิดขึ้น

กราบขอบพระคุณบิดามารดาที่ให้โอกาสในการศึกษาเล่าเรียน และคอยเป็น
กำลังใจสนับสนุนเรื่อยมา

ขอบคุณเพื่อนๆทุกคนที่ให้ความช่วยเหลือเพื่ออุปกรณ์และข้อมูลต่างๆ
ตลอดจนการจัดทำปริญญาบัตร

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อ | ก |
| กิตติกรรมประกาศ | ข |
| สารบัญ | ค |
| สารบัญรูป | ง |
| บทที่ 1 บทนำ | 1 |
| บทที่ 2 ทฤษฎีและหลักการ | 2 |
| 2.1 ส่วนประกอบและการทำงานโดยทั่วไปของลิฟต์ | 2 |
| 2.2 หลักการทำงานของระบบ | 16 |
| 2.3 หลักการทำงานของภาคตรวจจับ | 16 |
| 2.4 หลักการทำงานของภาคขับเคลื่อน | 17 |
| บทที่ 3 การออกแบบและจัดสร้าง | 18 |
| 3.1 คุณสมบัติของระบบจำลองการเคลื่อนที่ของลิฟต์ | 18 |
| 3.2 ระบบโดยรวม | 19 |
| 3.3 ส่วนการควบคุมและประมวลผล | 20 |
| 3.4 ส่วนการขับเคลื่อน | 24 |
| 3.5 ส่วนการรับและแสดงผล | 25 |
| 3.6 ส่วนการจ่ายพลังงาน | 25 |
| 3.7 ระบบใช้งานจริง | 26 |
| บทที่ 4 การทดลองและผลการทดลอง | 29 |
| บทที่ 5 บทสรุป | 33 |
| เอกสารอ้างอิง | |
| ภาคผนวก ก FLOW CHART และ PROGRAM | |
| ภาคผนวก ข DATA SHEET | |

สารบัญรูป

| | หน้า |
|---|------|
| 1. รูปที่ 2.1 แสดงการติดตั้ง Steel Tape | 7 |
| 2. รูปที่ 2.2 แสดงลักษณะการแขวนของสลิง | 8 |
| 3. รูปที่ 2.3 แสดงประตูที่ช่องลิฟต์และประตูที่ตัวลิฟต์ | 10 |
| 4. รูปที่ 2.4 แสดงอุปกรณ์ Door Lock | 14 |
| 5. รูปที่ 2.5 แสดงอุปกรณ์ Door Close | 15 |
| 6. รูปที่ 2.6 แสดงสัญญาณที่ได้จาก Encoder | 17 |
| 7. รูปที่ 3.1 แสดงโครงสร้างของระบบจำลองการเคลื่อนที่ของลิฟต์ | 18 |
| 8. รูปที่ 3.2 แสดงลักษณะของตัวลิฟต์ | 19 |
| 9. รูปที่ 3.3 แสดงบล็อกไดอะแกรมโดยรวมของระบบควบคุมการเคลื่อนที่ของลิฟต์ | 19 |
| 10. รูปที่ 3.4 แสดงบล็อกไดอะแกรมของชุดควบคุมการทำงานของระบบลิฟต์ | 20 |
| 11. รูปที่ 3.5 แสดงบล็อกไดอะแกรมการทำงานของชิพควบคุมความเร็ว LM629N | 20 |
| 12. รูปที่ 3.6 แสดงวงจรควบคุมและประมวลผล | 23 |
| 13. รูปที่ 3.7 แสดงวงจรขับมอเตอร์ขับเคลื่อนตัวลิฟต์ | 24 |
| 14. รูปที่ 3.8 แสดงวงจรขับมอเตอร์เปิด-ปิดประตูลิฟต์ | 24 |
| 15. รูปที่ 3.9 แสดงวงจรรับและแสดงผล | 25 |
| 16. รูปที่ 3.10 แสดงวงจรจ่ายพลังงานส่วนที่ 1 | 25 |
| 17. รูปที่ 3.11 แสดงวงจรจ่ายพลังงานส่วนที่ 2 | 26 |
| 18. รูปที่ 3.12 แสดงโครงสร้างของลิฟต์ | 26 |
| 19. รูปที่ 3.13 แสดงลักษณะตัวลิฟต์ | 27 |
| 20. รูปที่ 3.14 แสดงกลไกการเปิด-ปิดประตู | 27 |
| 21. รูปที่ 3.15 แสดงวงจรควบคุมและประมวลผล | 28 |
| 22. รูปที่ 3.16 แสดงวงจรรับและแสดงผล | 28 |
| 23. รูปที่ 4.1 แสดงสัญญาณเอาต์พุตที่ได้จากการทดลองที่ 1 | 30 |
| 24. รูปที่ 4.2 แสดงสัญญาณเอาต์พุตที่ได้จากการทดลองที่ 2 | 30 |
| 25. รูปที่ 4.3 แสดงสัญญาณเอาต์พุตที่ได้จากการทดลองที่ 3 | 31 |
| 26. รูปที่ 4.4 แสดงสัญญาณ PHASE A | 31 |
| 27. รูปที่ 4.5 แสดงสัญญาณ PHASE B | 32 |

บทที่ 1

บทนำ

ปัจจุบันระบบลิฟต์มีบทบาทต่อชีวิตประจำวันของประชาชนโดยทั่วไปมากขึ้น ดังจะเห็นได้ว่าในอาคารสมัยใหม่โดยส่วนใหญ่แล้วจะมีระบบลิฟต์เพื่ออำนวยความสะดวก ในการขนส่งเคลื่อนย้ายทั้งคนและสิ่งของ จึงสามารถจะกล่าวได้ว่าระบบลิฟต์เป็นระบบที่จำเป็นและมีความสำคัญมากที่สุดระบบหนึ่ง

โครงการนี้ได้ทำ การศึกษาระบบการเคลื่อนที่ของลิฟต์ และได้ทำ การประยุกต์ใช้ไมโครคอนโทรลเลอร์ 89C51 มาใช้ในการควบคุมระบบซึ่งสามารถแบ่งออกเป็นส่วนที่สำคัญดังนี้คือ

- ส่วนวงจรจ่ายพลังงาน (Power supply circuit part)
- ส่วนวงจรควบคุมและประมวลผล (Data control and processing part)
- ส่วนวงจรขับเคลื่อนมอเตอร์กระแสตรง (DC motor driver part)
- ส่วนรับและแสดงผล (Input and output part)

รายละเอียดของส่วนประกอบต่างๆ รวมทั้งวงจรและโปรแกรมการทำงานจะได้กล่าวโดยละเอียดในบทต่อไป

บทที่ 2

ทฤษฎี และหลักการ

2.1 ส่วนประกอบและการทำงานโดยทั่วไปของลิฟต์

ลิฟต์โดยทั่วไปจะมีส่วนประกอบที่มีลักษณะใกล้เคียงกัน กล่าวคือ ประกอบไปด้วยตู้ลิฟต์ และน้ำหนักถ่วงแขวนติดเข้าด้วยกัน โดยลวดสลิงที่คล้องผ่านรอกซึ่งขับเคลื่อนรอกนี้ด้วยมอเตอร์ ที่ตัวตู้ลิฟต์สามารถเคลื่อนขึ้น-ลงได้ ซึ่งลิฟต์ทั่วไปจะประกอบไปด้วยส่วนประกอบใหญ่ ๆ ดังนี้คือ

1. ส่วนประกอบที่เป็นส่วนของห้องเครื่องลิฟต์ (MACHINE ROOM)

1.1 TRACTION MACHINE & MOTOR

1.2 MAGNET BRAKE

1.3 SPEED GOVERNOR

1.4 FLOOR CONTROLLER

1.5 BEAM PULLEY

1.6 PANALY

2 ส่วนประกอบที่อยู่ภายในช่องลิฟต์ (HOISTWAY)

2.1 GUIDE RAIL

2.2 ROPE

- MAIN ROPE

- GOVERNOR ROPE

2.3 COMPENSATING CHAIN

2.4 COMPENSATING ROPE

2.5 TENSION PULLEY

- COMPENSATING ROPE TENSION PULLEY

- GOVERNOR ROPE TENSION PULLEY

- FLOOR CONTROLLER TENSION PULLEY

2.6 STEEL TAPE

2.7 TAIL CORD (TRAVELLING CABLE)

2.8 LANDING DEVICES

- SHEILDING DEVECES

- SLOW DOWN CAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 SWITCH

- LIMIT SWITCH
- FINAL LIMIT SWITCH
- PIT SWITCH

2.10 COUNTER WEIGHT

2.11 BUFFER

- SPRING BUFFER
- OIL BUFFER

2.12 WIRING & PIPING

2.13 CAR CAGE ASSEMBLY

3. ส่วนประกอบของตู้ลิฟต์ (CAGE)

3.1 OPERATION BOARD

- MAIN OPB
- SUB OPB

3.2 CARPOSITION INDICATOR

3.3 ILLUMINATION INSIDE CAGE

- FLUORESCENT OR INCANDESENT LAMP
- EMERGENCY LAMP

3.4 INTERCOMMUNICATION EQUIPMENT

3.5 CAGE

3.6 GUIDE SHOE, GUIDE ROLLER & OILER

3.7 WEIGHTING DEVICE

3.8 SLOW DOWN SWITCH

3.9 LANDING DEVICE

- R.M.POSITECTOR

3.10 CAGE SLOW DOWN CAM

3.11 SAFETY DEVICE

4. ส่วนประกอบของประตูลิฟต์

4.1 DOOR MACHINE

- CAGE DOOR
- HATCH DOOR

- GATE SWITCH
- TYPE DOOR MACHINE
- 4.2 LINK MACHANISM
- 4.3 DOOR LOCK SWITCH
- 4.4 DOOR SAFETY
 - SHOE TYPE
 - PHOTO TYPE
 - INDUCTION TYPE
- 4.5 JAMB & SILL
- 4.6 HALL BUTTON
- 4.7 HALL LANTERN
- 4.8 HALL INDICATOR

หน้าที่ของส่วนประกอบที่สำคัญของลิฟต์
ส่วนประกอบของลิฟต์มีหน้าที่ดังนี้คือ

1. ELEVATOR MOTOR

แบ่งเป็นชนิดใหญ่ ๆ ได้ 2 ชนิด คือ

1.1 ALTERNATING CURRENT MOTOR หรือมอเตอร์กระแสสลับ แบ่งออกเป็น

1.1.1 ONE SPEED MOTOR เป็นมอเตอร์ชนิด 3-PHASE INDUCTION MOTOR ที่ถูกออกแบบมาใช้กับแรงบิด (TORQUE) สูง และมีแรงเฉื่อย (INERTIA) น้อยภายในจะประกอบด้วย STATOR ที่มี 6 ขั้ว และออกแบบเป็น INDUCTION MOTOR ทั่ว ๆ ไป แต่ในบางกรณีที่มีมอเตอร์ขนาดแรงขับมาก ๆ หรือลิฟต์ขนาดใหญ่จะใช้ มอเตอร์ชนิดที่ ROTOR เป็นขดลวด (WIRE WOUND ROTOR)

1.1.2 TWO SPEED MOTOR เป็น AC MOTOR ที่ขดลวด STATOR มี 2 กลุ่มพันอยู่รวมกัน คือ กลุ่มหนึ่งมี 6 POLE และอีกกลุ่มหนึ่งมี 24 POLE สำหรับความเร็วสูงและความเร็วต่ำตามลำดับ

- DIRECT CURRENT MOTOR หรือมอเตอร์ไฟฟ้ากระแสตรง

สำหรับลิฟต์ที่มีความเร็วสูงส่วนใหญ่จะใช้ MOTOR ชนิดนี้ เนื่องจากสามารถที่จะควบคุมความเร็วของลิฟต์ให้เพิ่มขึ้นและลดลงได้ดีกว่าชนิด AC MOTOR ในรุ่นแรกของระบบลิฟต์ชนิดนี้จะใช้ร่วมกับ MOTOR - GENERATOR SET ซึ่ง MOTOR - GENERATOR SET จะ

เป็นตัวจ่ายพลังงานไฟฟ้าให้ DC MOTOR เพื่อควบคุมความเร็วโดยใช้อุปกรณ์ประเภท THYRISTOR เข้ามาแทน

2. TRACTION MACHINE ELEVATOR

TRACTION MACHINE คือระบบถ่ายกำลังงานจากมอเตอร์ไปยังตัวลิฟต์ โดยใช้หลักการของแรงเสียดทานระหว่างตัวรอกกับสลิงที่คล้องผ่านรอก traction machine นี้แบ่งออกเป็น 2 ประเภทคือ

- GEARLESS TRACTION MACHINE เป็นระบบถ่ายทอดกำลังที่ใช้ระบบขับเคลื่อนโดยตรงโดยไม่ผ่านระบบเกียร์ที่รอกของรอกที่ใช้แขวนสลิง ตัวลิฟต์จะต่อโดยตรงกับเพลลาของมอเตอร์ รวมทั้งระบบหยุดลิฟต์ (DRUM BREAK) ด้วย ระบบนี้ส่วนใหญ่จะใช้กับลิฟต์ที่มีความเร็วสูง (ประมาณไม่ต่ำกว่า 120 เมตร/นาที)

- GEARED TRACTION MACHINE สำหรับลิฟต์ที่มีความเร็วไม่สูงมาก (ปกติความเร็วต่ำกว่า 105 เมตร/นาที) จะใช้ระบบ GEARED TRACTION MACHINE ซึ่งเป็นระบบถ่ายทอดกำลังจาก MOTOR ไปยังรอกขับเคลื่อนตัวลิฟต์โดยผ่านระบบเกียร์ที่รอกลงมา โดย GEARED TRACTION MACHINE นี้จะมีระบบหยุดลิฟต์ (DRUM BREAK) อยู่ระหว่างเพลลาของมอเตอร์ของระบบเกียร์

3. SPEED GOVERNOR

SPEED GOVERNOR เป็นอุปกรณ์ที่ติดตั้งอยู่ในห้องเครื่องลิฟต์มีหน้าที่คอยตรวจจับความเร็วในการเคลื่อนที่ของลิฟต์ไม่ให้เคลื่อนที่เกินกว่าความเร็วที่กำหนดไว้ โดยใช้ทฤษฎีของแรงเหวี่ยงหนีศูนย์กลาง GOVERNOR หมุนได้โดยอาศัยสลิง GOVERNOR ROPE ที่ยึดกับตัวลิฟต์โดยตรงและมีลูกตุ้มอยู่ภายในป้อลิฟต์คอยถ่วงให้สลิงตั้งอยู่เสมอ

GOVERNOR นี้จะประกอบไปด้วยสวิทช์ไฟฟ้าและน้ำหนักกดสลิง (CATCH WEIGHT) สวิทช์ไฟฟ้ามีหน้าที่ตัดกระแสไฟฟ้าที่จะป้อนเข้ามอเตอร์โดยผ่านระบบควบคุมลิฟต์ ส่วนน้ำหนักกดสลิงนั้นจะทำหน้าที่กดสลิงเพื่อให้สลิงไปกระชาก SAFETY CATCH ให้ SAFETY CATCH ทำงานบังคับมิให้ลิฟต์เคลื่อนที่ต่อไปได้ ซึ่งเป็นการป้องกันลิฟต์ตกลงไปกระแทกกับบ่อ

การทำงานของ GOVERNOR นี้จะทำงาน 2 ขั้นตอน คือ

1. เมื่อความเร็วของมอเตอร์เพิ่มขึ้นถึง 125 % ของความเร็วสูงสุดของลิฟต์นั้น ๆ

GOVERNOR ก็จะไปชนกระเดื่องสวิทช์ไฟฟ้า ทำให้งจรของงานสวิทช์ไฟฟ้าเปิดออก ทำให้ระบบคอนโทรลลิฟต์ตัดกระแสไฟฟ้าที่จะป้อนเข้ามอเตอร์ และตัดกระแสไฟฟ้าที่ป้อนเข้า MAGNETIC BRAKE ทำให้ลิฟต์หยุดการเคลื่อนที่

2. เมื่อความเร็วของลิฟต์เพิ่มขึ้นถึง 135 % ของความเร็วสูงสุดของลิฟต์นั้น GOVERNOR จะบังคับน้ำหนักกดลงสลิงให้ตกลงจากที่ยึดลงมาจับสลิง สลึงก็จะไปกระชาก SAFETY CATCH ให้จับยึดกับราง ทำให้ลิฟต์หยุดการเคลื่อนที่ทันที

ดังนั้นจะเห็นว่า GOVERNOR จะมีหน้าที่บังคับให้ลิฟต์หยุดโดยทางไฟฟ้าก่อนแต่ถ้าไม่สามารถบังคับให้ลิฟต์หยุดทางไฟฟ้าได้แล้ว จึงจะบังคับให้ลิฟต์หยุดการทำงานโดยทางกล

4. SAFETY DEVICE (SAFETY CATCH)

SAFETY CATCH เป็นอุปกรณ์ที่ติดตั้งอยู่ด้านล่างของผู้ลิฟต์ เป็นอุปกรณ์ที่ทำงานด้วยระบบทางกลเพื่อบังคับให้ลิฟต์หยุดการเคลื่อนที่ ในกรณีที่ลิฟต์เคลื่อนที่ด้วยความเร็วเกินกว่า 135 % ของความเร็วสูงสุด

การทำงานของ SAFETY CATCH นี้จะทำงานเมื่อสลิงของ GOVERNOR (GOVERNOR ROPE) ถูกจับยึดโดย GOVERNOR WEIGHT ทำให้ SLING หยุดการเคลื่อนที่ทันใดในขณะที่ตัวลิฟต์ยังคงเคลื่อนที่อยู่ทำให้เกิดลักษณะการกระชากขึ้น ซึ่งการกระชากนี้จะทำให้ดึงก้านเหล็ก (PUSH ROD) ทำให้แท่งลิ้มเคลื่อนที่บีบกดเข้ากับรางลิฟต์ทำให้ลิฟต์หยุดได้

SAFETY DEVICE แบ่งออกเป็น 2 ชนิด คือ

1. R TYPE SAFETY DEVICE ใช้สำหรับลิฟต์ที่มีความเร็วไม่เกิน 45 เมตร/นาที ใช้ลูกล้อ (CATCH ROLLER) ในการกดราง

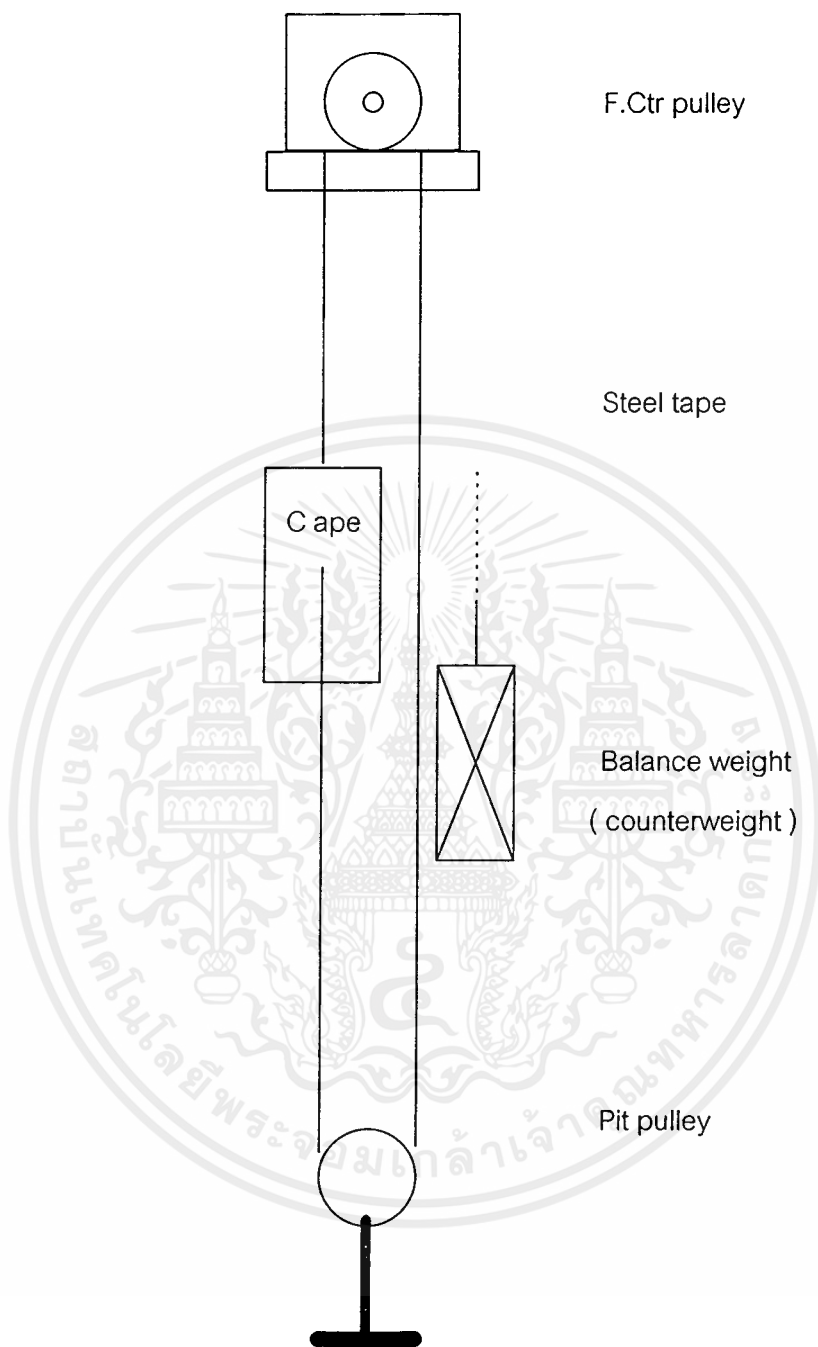
2. W TYPE SAFETY DEVICE ใช้สำหรับลิฟต์ที่มีความเร็วเกิน 45 เมตร/นาที ใช้ลักษณะลิ้มสามเหลี่ยมในการกดกับราง

5. FLOOR CONTROLLER

FLOOR CONTROLLER หรือ FLOOR SELECTOR นี้มีลักษณะเป็นตู้ CONTROL ภายในประกอบด้วย CONTROL ไฟฟ้าที่มีการเคลื่อนที่สัมพันธ์กับการเคลื่อนที่ของตัวลิฟต์ โดยอาศัยเฟืองทดรอบจากฟันเฟืองที่มีการหมุนรอบแกนแผ่นเหล็กที่เรียกว่า STEEL TAPE ที่ติดยึดกับตัวลิฟต์

FLOOR CONTROLLER มีหน้าที่ตรวจเช็ค ตำแหน่งของตัวลิฟต์ว่าอยู่ที่ชั้นใด ควบคุมไฟบอกชั้นทิศทางการเคลื่อนที่ และการเข้าออกของลิฟต์ให้ถูกต้อง และตรงตามความต้องการของผู้ใช้ลิฟต์

PERFORATED TAPE (.VS - DB)



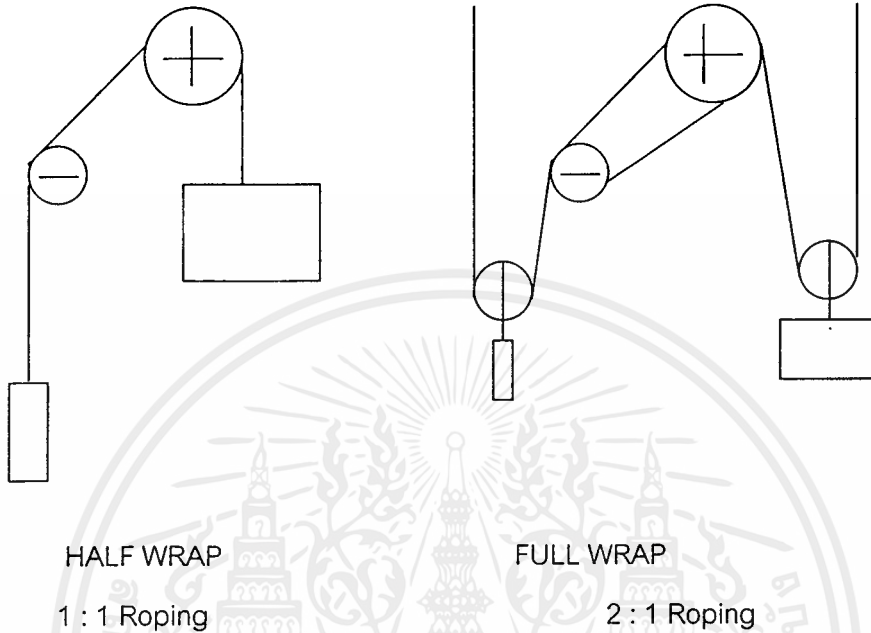
BRIEF ILLUSTRATION OF STEELTAPE
INSTALLATION

รูปที่ 2.1 แสดงการติดตั้ง Steel Tape

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. ROPES AND THE ROPING

ROPES หรือส่วนใหญ่เรียกกันว่าสลิงนั้น มีหน้าที่หิ้วหรือดึงตัวตู้ลิฟต์ และน้ำหนักถ่วง (COUNTER WEIGHT) ให้ขึ้นลง โดยอาศัยแรงเสียดทานระหว่าง ROPES กับ SHEAVE ROPING หรือการแขวนของสลิงนั้นมี 2 ลักษณะ



รูปที่ 2.2 แสดงลักษณะการแขวนของสลิง

7. SHEAVE

SHEAVE หรือมู่เลย์นี้ใช้สำหรับพาดสลิง และทำให้สลิงเคลื่อนที่เพื่อที่จะดึงให้ลิฟต์ขึ้นหรือลงตามความต้องการ จำนวนช่อง (GROOVE) ของ SHEAVE จะขึ้นอยู่กับขนาดน้ำหนักบรรทุกของลิฟต์ ช่องของ SHEAVE นี้จะมีลักษณะต่าง ๆ กัน ตามความต้องการของแรงเสียดทาน ที่จะต้องไปให้เกิดการลื่นไหลของสลิงกับช่อง SHEAVE

SHEAVE มี 2 ชนิด

1. MAIN SHEAVE เป็นมู่เลย์ที่เชื่อมต่อโยงโดยตรงกับ TRACTION MACHINE เป็นตัวดึงในการให้สลิงเคลื่อนที่

2. DEFLECTION SHEAVE หรือ BREAM PULLEY มีหน้าที่กางสลิงให้มีระยะห่างจากตัวตู้ลิฟต์ตามระยะที่เหมาะสม (ขึ้นอยู่กับขนาดของตู้ลิฟต์)

8. COUNTER WEIGHT

COUNTER WEIGHT มีหน้าที่ เป็นน้ำหนักถ่วงของตู้ลิฟต์ และดึง ROPE ให้มีความตึงเพียงพอที่จะทำให้เกิดความเสียดทานระหว่าง มู่เลย์ และ ROPE ที่ต้องการถ่วงตัวลิฟต์นั้นให้เกิดความสมดุลย์ระหว่างตัวตู้ลิฟต์และ COUNTER WEIGHT จะทำให้เกิดประโยชน์ คือ ลดภาระ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของมอเตอร์ที่จะจุดลิฟต์ให้น้อยลง และเมื่อมีการเคลื่อนที่ของน้ำหนักตัวลิฟต์จริงรวมกับครึ่งหนึ่งของน้ำหนักลิฟต์ COUNTER WEIGHT นี้จะประกอบด้วย โครงลูกถ่วง (COUNTER WEIGHT FRAME) ลูกถ่วง(SUB WEIGHT)

9. COMPENSATING CHAIN AND ROPE

เป็นอุปกรณ์ที่ใช้ถ่วงตัวลิฟต์ ให้เกิดความสมดุลย์ของน้ำหนักของตัวลิฟต์ชนิดหนึ่งเช่นกัน ในกรณีที่ลิฟต์มีระยะการเคลื่อนที่ขึ้นลงมากกว่า 30 เมตรขึ้นไป เมื่อระยะการเคลื่อนที่ของลิฟต์มากกว่า 30 เมตรนี้ น้ำหนักของ ROPE CABLE ก็จะมีผลต่อน้ำหนักของลิฟต์ด้วย ในกรณีที่ลิฟต์อยู่ชั้นบนและชั้นล่าง จึงต้องมีการชดเชยน้ำหนักด้วย COMPENSATING CHAIN AND ROPE ซึ่ง COMPENSATING CHAIN AND ROPE นี้จะยึดติดอยู่ระหว่างด้านล่างของ ตัวตัวลิฟต์ และด้านล่างของ COUNTER WEIGHT RAME การที่จะเลือกใช้ว่าจะใช้ CHAIN(โซ่) หรือ ROPE(สลิง) นั้นจะขึ้นอยู่กับขนาดบรรทุก ระยะทางเคลื่อนที่ของตัวลิฟต์และความเร็ว

10. GUIDE RAILS

GUIDE RAILS หรือรางของระบบลิฟต์ โดยปกติจะมีรูปตัดเป็นรูปตัว T ทำจากเหล็กหนา มีหน้าที่เป็นตัวบังคับให้ตัวลิฟต์ และ COUNTER WEIGHT เคลื่อนที่ขึ้นและลงโดยไม่ส่ายไปมา การเลือกใช้กับลิฟต์นั้นจะขึ้นอยู่กับขนาดและน้ำหนักของตัวลิฟต์ ลักษณะการติดตั้งรางเข้ากับโครงสร้างและระยะจับยึดราง

การแบ่งประเภทของราง จะแบ่งตามน้ำหนักเป็นกิโลกรัมต่อความยาว 1 เมตร และมาตรฐานความยาวของราง ก็คือ 5 เมตร

11. BUFFER

BUFFER เป็นอุปกรณ์ที่ติดตั้งอยู่ภายในบ่อลิฟต์ มีหน้าที่ป้องกันมิให้ตัวลิฟต์วิ่งเลยระหว่างชั้นล่างสุดลงมากกระทบกับกันบ่อลิฟต์ อันเนื่องมาจากเกิดอุบัติเหตุ หรือระบบควบคุมลิฟต์เกิดการขัดข้อง ซึ่งไม่สามารถบังคับให้ลิฟต์หยุดลงได้

BUFFER มี 2 ชนิด ตามโครงสร้าง คือ

1. SPRING BUFFER ส่วนใหญ่ใช้สำหรับลิฟต์ที่มีความเร็วไม่เกิน 60 เมตรต่อนาที
2. OIL BUFFER ส่วนใหญ่ใช้สำหรับลิฟต์ที่มีความเร็วตั้งแต่ 90 เมตรต่อนาที

12. CAR (CAGE)

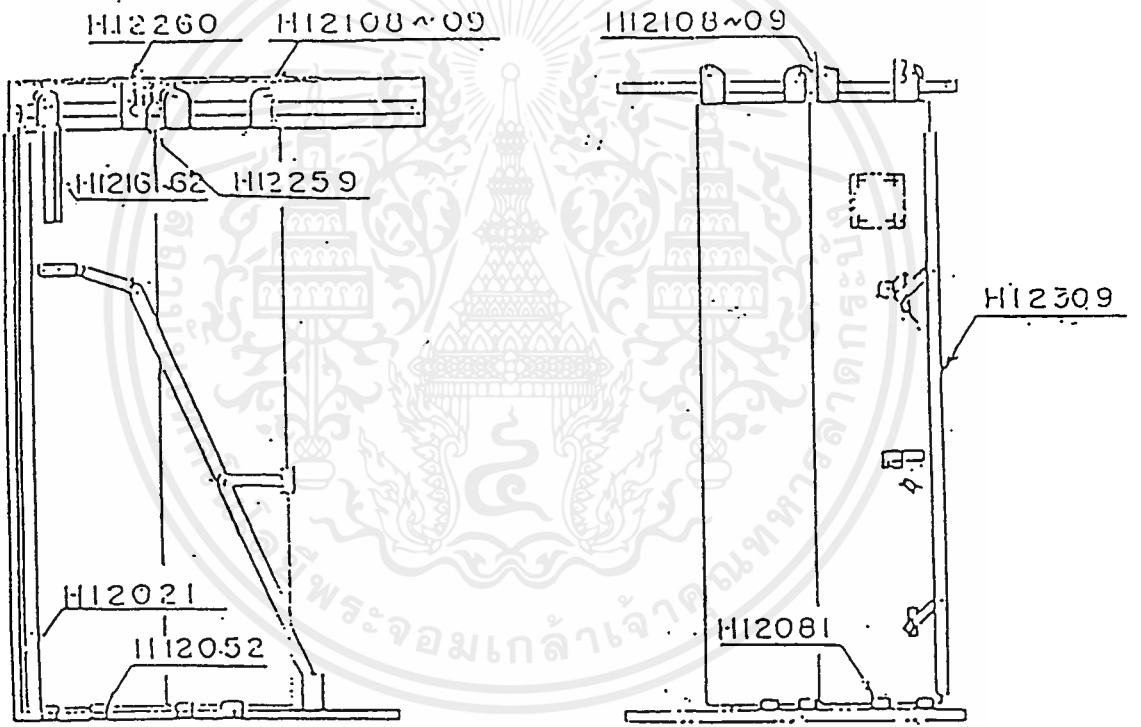
CAR หรือตู้ลิฟต์ หรือห้องโดยสารลิฟต์ มีลักษณะเป็นกล่องสี่เหลี่ยม ประกอบด้วย ประตูลิฟต์ ผนังข้าง เพดาน แผงควบคุมลิฟต์ ไฟแสดงตำแหน่งลิฟต์ พัดลมระบายอากาศ ฝ้าลิฟต์ และโครงสร้างเกือบทั้งหมดทำจากเหล็ก มีหน้าที่บรรทุกผู้โดยสารและสิ่งของขึ้นลง

13. DOOR OPERATION MACHANISM

DOOR OPERATION MACHANISM แบ่งออกเป็น 2 ชนิดใหญ่ ๆ คือ

1. SIDE SLIDING คือการทำงานของประตูแบบเปิดออกทางด้านแนวนอน

2. UP SLIDING คือการทำงานของประตูลิฟต์แบบเปิดออกทางแนวตั้ง นอกจากนี้ยังมีประเภทของการทำงานของประตูแบบอื่น อีก คือ

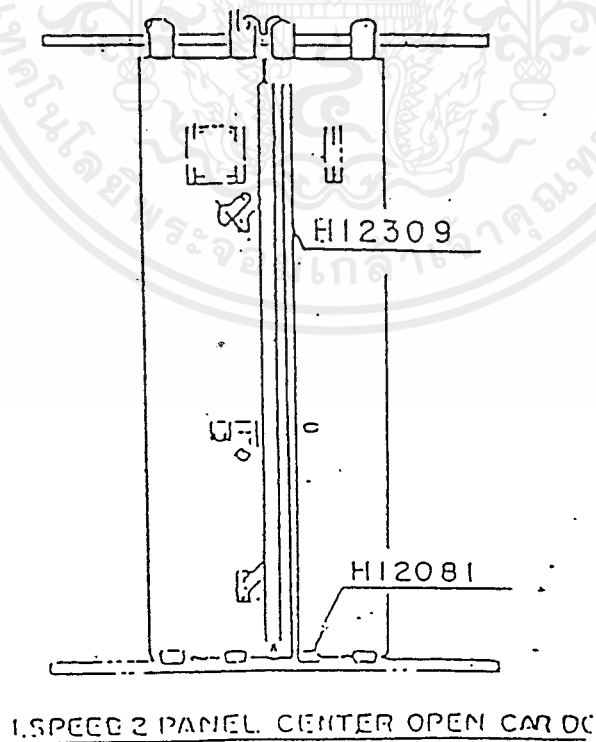
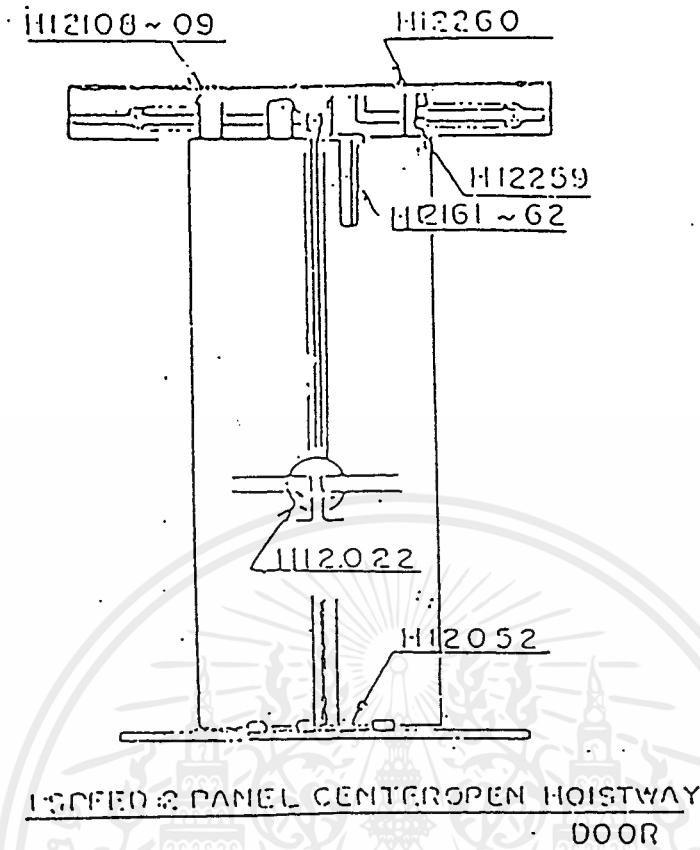


2 SPEED 2 PANEL HOISTWAY DOOR

2 SPEED 2 PANEL CAR DOOR

รูปที่ 2.3 แสดงประตูที่ช่องลิฟต์และประตูที่ตัวลิฟต์

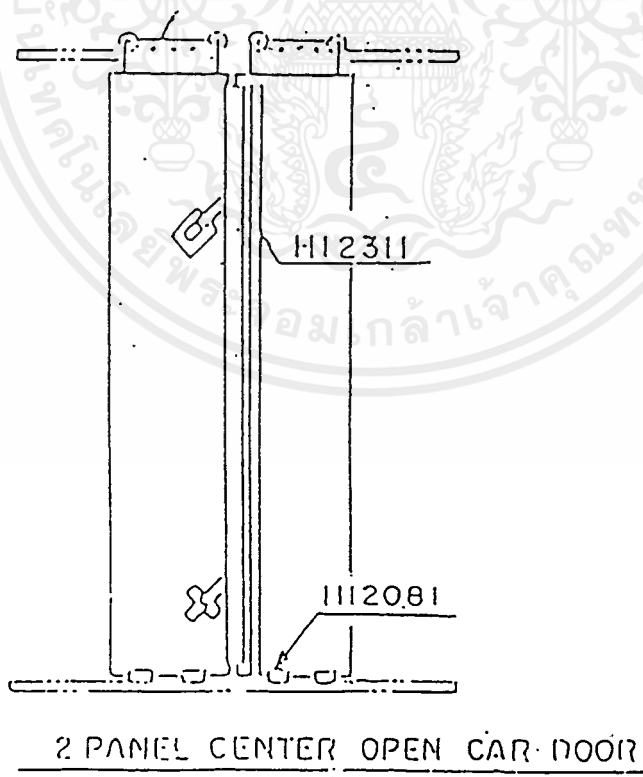
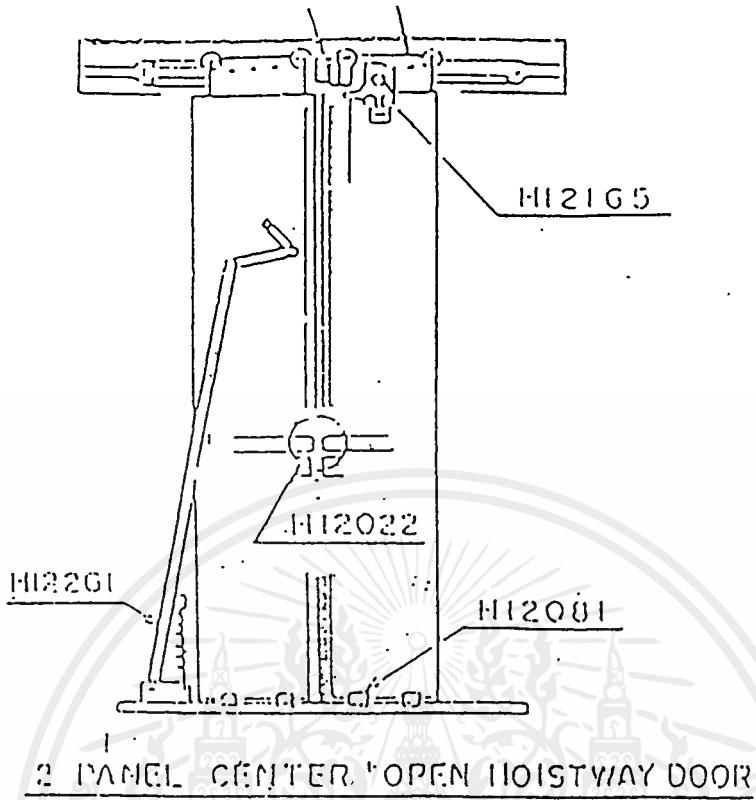
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3(ต่อ) แสดงประตูที่ช่องลิฟต์และประตูที่ตัวลิฟต์

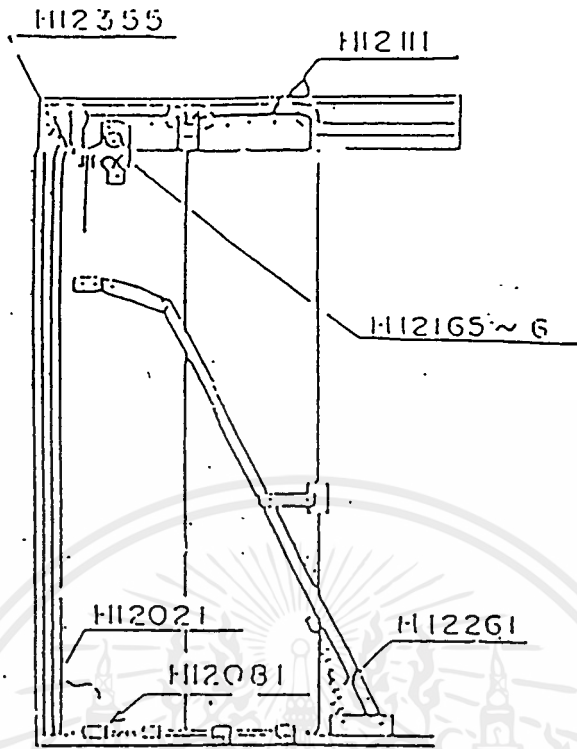
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

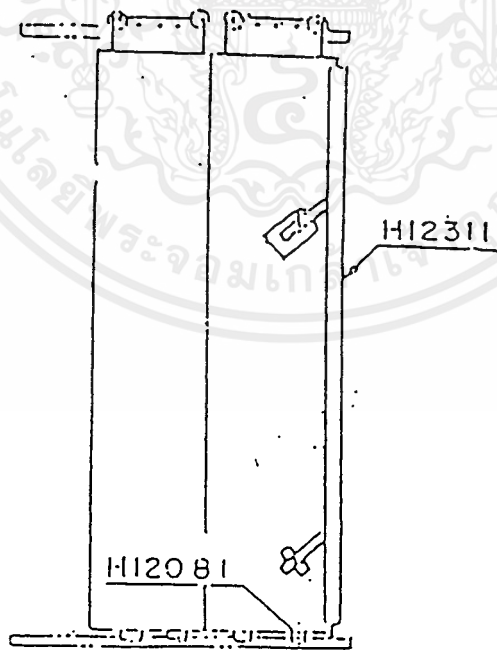


รูปที่ 2.3(ต่อ) แสดงประตูที่ช่องลิฟต์และประตูที่ตัวลิฟต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2 SPEED 2 PANEL HOISTWAY DOOR



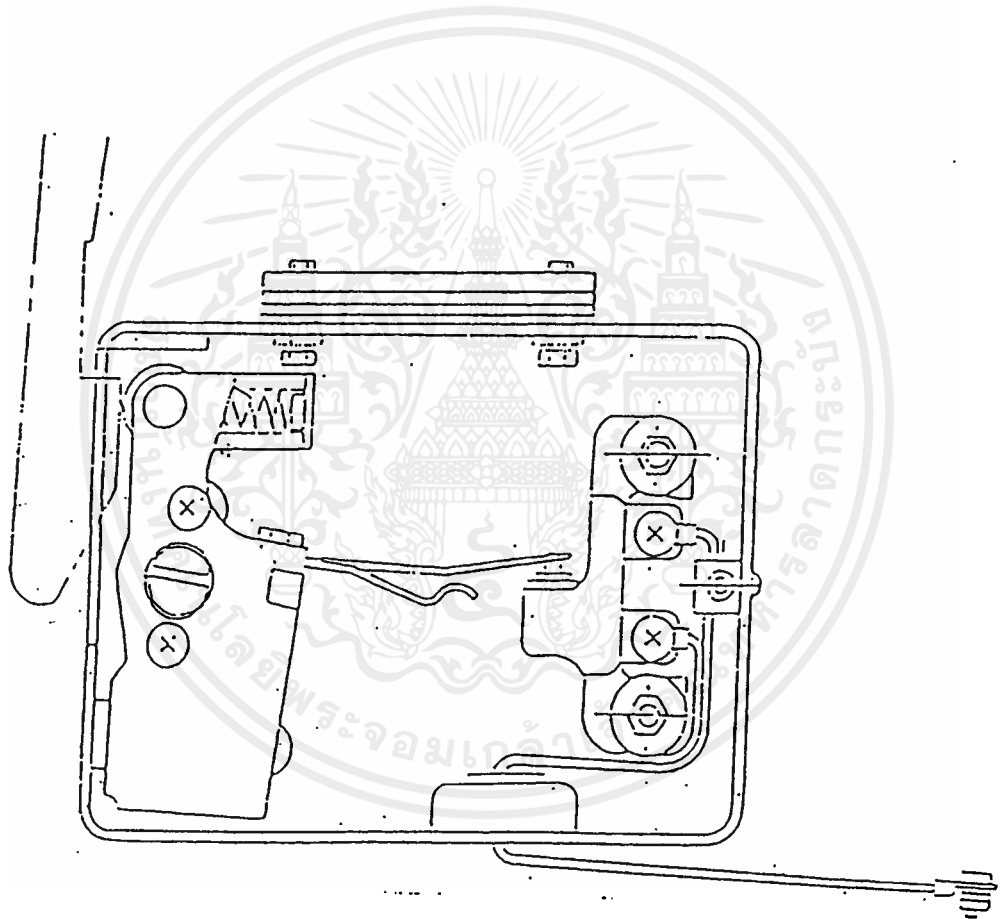
2 SPEED 2 PANEL CAR DOOR

รูปที่ 2.3(ต่อ) แสดงประตูที่ห้องลิฟต์และประตูที่ตัวลิฟต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

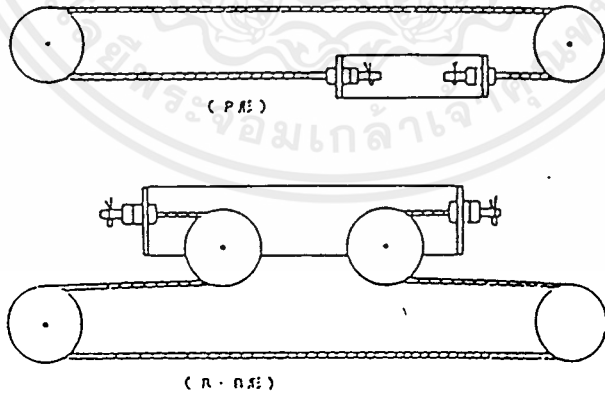
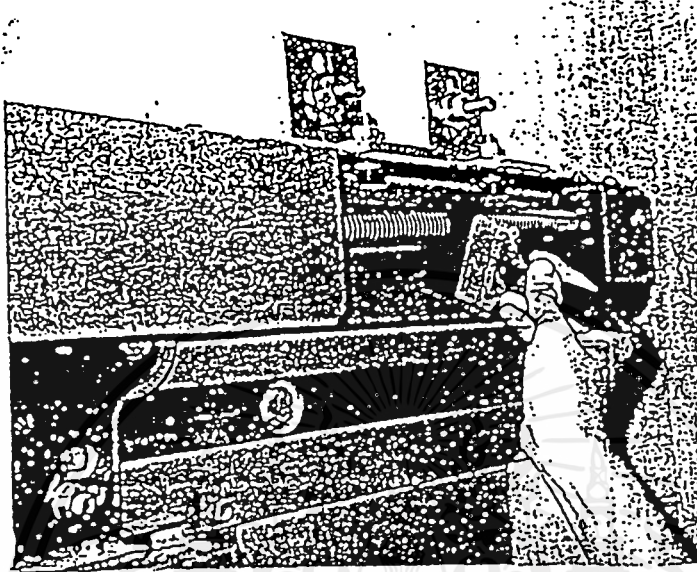
การทำงานของระบบเปิด-ปิดประตุนั้นจะทำงานโดยอาศัยบานประตูของตัวลิฟต์เป็นตัวพาพาให้ประตูนอก (HATCHDOOR) เปิดออก ดังนั้นการที่ประตูนอก(HATCHDOOR) จะเปิดออกได้นั้นจะอยู่ในช่วงระยะเวลาหนึ่งเท่านั้น ซึ่งเราเรียกว่า DOOR ZONE ต้นกำเนิดของการเปิด-ปิดประตูนี้จะมาจาก DOOR MOTOR ที่ติดอยู่บนหลังคาลิฟต์นั่นเอง

สำหรับประตูนอก (HATCHDOOR) นี้จะไม่สามารถเปิดออกโดยบุคคลที่อยู่ภายนอก โดยจะมีกลไกทางกลเป็นตัวกีดขวางติดตั้งไว้ ซึ่งอุปกรณ์นี้เรียกว่า DOOR LOCK แต่ยังสามารถเปิดประตูนี้ได้โดยใช้กุญแจสำหรับเปิดประตูลิฟต์ ซึ่งออกแบบโดยเฉพาะตามชนิดของลิฟต์



รูปที่ 2.4 แสดงอุปกรณ์ Door Lock

นอกจากนี้ ประตูนอกรังยังมีอุปกรณ์ที่สามารถปิดตัวเองได้อีกด้วย ซึ่งอุปกรณ์ที่บังคับให้ปิดตัวเองได้นี้เรียกว่า DOOR CLOSE ปัจจุบันส่วนใหญ่ประตูลิฟต์จะใช้ติดตั้งอยู่ที่รางแขวนประตู หรือ HEADER เป็นตัวปิดประตู



รูปที่ 2.5 แสดงอุปกรณ์ Door Close

13. INDICATOR

INDICATOR คืออุปกรณ์ที่แสดงหรือแจ้งให้ผู้โดยสาร และผู้ที่ต้องการจะใช้ลิฟต์ที่ตำแหน่งของตัวลิฟต์ทราบทิศทางการเคลื่อนที่ของตัวลิฟต์ รูปแบบและลักษณะของ INDICATOR มีลักษณะรูปแบบขึ้นอยู่กับความต้องการของผู้ใช้และบริษัทผู้ผลิต

14. LANDING DEVICE

LANDING DEVICE คืออุปกรณ์ที่ใช้ในการตรวจสอบระดับชั้นของลิฟต์ เพื่อให้ลิฟต์หยุดให้ตรงตามระดับชั้นจริงของอาคาร ภายหลังจากที่ลิฟต์ได้คำสั่งให้ลดความเร็วลง เพื่อเข้าจอดในชั้นที่ได้รับคำสั่งแล้วอุปกรณ์ที่ใช้ในการตรวจสอบนี้จะใช้ทฤษฎีของ MAGNETIC FIELD

LANDING DEVICE นี้ประกอบไปด้วย 2 ส่วนคือ

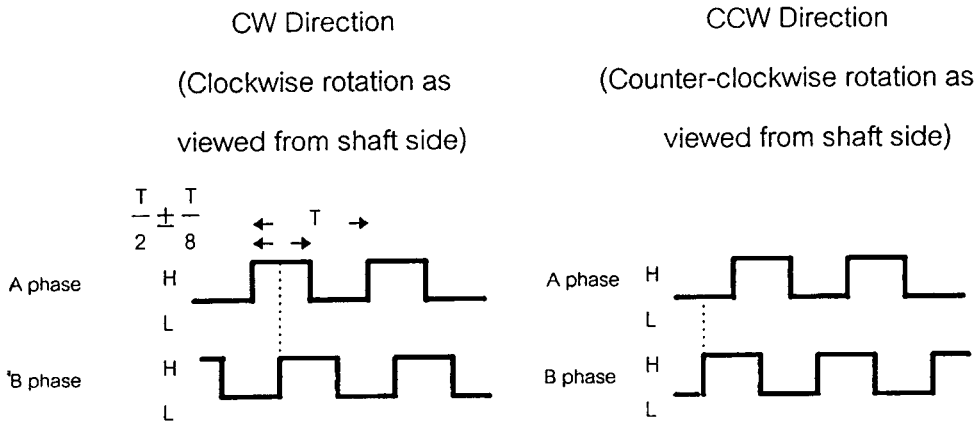
1. POSITION DETECTOR (POSITECTOR) ติดตั้งอยู่บนหลังคาลิฟต์
2. SHEILDING PLATE เป็นแผ่นเหล็กติดตั้งอยู่ภายในช่องลิฟต์

2.2 หลักการทำงานของระบบ

หลักการทำงานของระบบการเคลื่อนที่ของลิฟต์ก็คือลิฟต์สามารถเคลื่อนที่ได้โดยอาศัยการควบคุมจากหน่วยประมวลผลกลาง โดยหน่วยประมวลผลกลางจะทำ การรับอินพุตจากภายนอกซึ่งก็คือการระบุการเคลื่อนที่ของลิฟต์ของผู้ใช้ในแต่ละชั้นเข้ามาเพื่อใช้ในการประมวลผลส่งการส่งารส่วนวงจรขับเคลื่อนมอเตอร์กระแสตรงเพื่อให้ลิฟต์เคลื่อนที่ไปตามต้องการ

2.3 หลักการทำงานของภาคตรวจจับ

วิธีการตรวจจับที่เลือกใช้ คือ วิธีการตรวจวัดระยะทางโดยรับสัญญาณจากเอนโคเดอร์หรืออุปกรณ์เข้ารหัส ซึ่งเป็นอุปกรณ์ที่เปลี่ยนลักษณะการหมุนเชิงกล (ในที่นี้คือการหมุนของแกนมอเตอร์) มาเข้าเป็นสัญญาณดิจิทัลนั่นคือเมื่อเกิดการหมุนของแกนมอเตอร์ โดยเอนโคเดอร์ จะสร้างจำนวน pulse ออกมาขึ้นอยู่กับระยะทางที่หมุนไป ซึ่งในที่นี้เอนโคเดอร์จะสร้างสัญญาณ pulse ออกมา 500 pulse ต่อการหมุน 1 รอบ หรือมีความละเอียด 1.39 องศาต่อสแต็ปสัญญาณ output จะแสดงดังรูป ในที่นี้จะใช้ output 2 เส้น ซึ่งมีความถี่เท่ากันแต่มีเฟสต่างกัน 90 องศา ถ้า Encoder หมุนในทิศทางตามเข็มนาฬิกา A lead B แต่ถ้าหมุนในทิศทางทวนเข็มนาฬิกา B lead A



รูปที่ 2.6 แสดงสัญญาณที่ได้จาก Encoder

2.4 หลักการทำงานของภาคขับเคลื่อน

ภาคขับเคลื่อนของระบบจำลองจะถูกแบ่งออกเป็น 2 ส่วน คือ

1. มอเตอร์ขับเคลื่อนตัวลิฟต์
2. มอเตอร์ที่ทำหน้าที่เปิด-ปิดประตูลิฟต์

แรงดันไฟฟ้าที่จ่ายให้กับมอเตอร์จะถูกควบคุมด้วยวิธีการ Pulse Width Modulation การบังคับทิศทาง

ทำได้โดยส่งสัญญาณบอกทิศทางให้วงจรขับเคลื่อนมอเตอร์ซึ่ง

1. ในกรณีของมอเตอร์ขับเคลื่อนลิฟต์ สัญญาณบอกทิศทางจะประมวลมาจากอินพุตเลือกชั้นจากภายนอก โดยไมโครคอนโทรลเลอร์จะตัดสินใจจากอินพุตที่ได้รับนี้ว่าลิฟต์จะเคลื่อนที่ขึ้นหรือลง

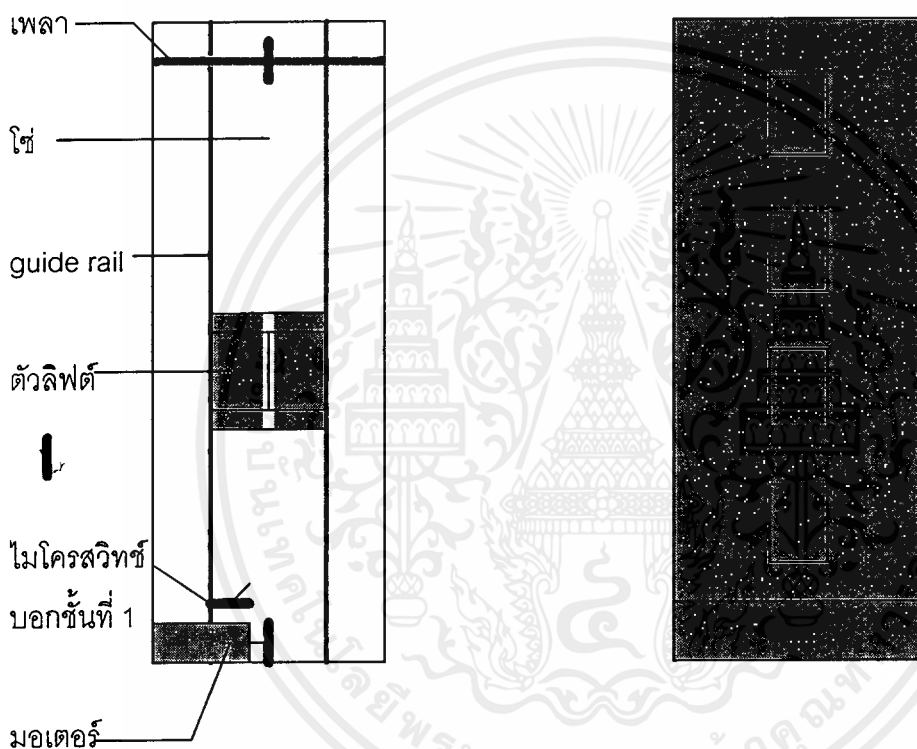
2. ส่วนมอเตอร์ที่ทำหน้าที่เปิด-ปิดประตูลิฟต์ ไมโครคอนโทรลเลอร์จะสั่งให้มอเตอร์ทำการเปิดประตูเมื่อลิฟต์เคลื่อนที่ถึงชั้นที่ต้องการ หลังจากนั้นทำการหน่วงเวลาพอสมควร แล้วจึงปิดประตู

บทที่ 3

การออกแบบและจัดสร้าง

3.1 คุณสมบัติของระบบจำลองการเคลื่อนที่ของลิฟต์

3.1.1 ระบบจำลองมีจำนวน 4 ชั้น การจัดวางโครงสร้างภายในเป็นดังรูป คือมอเตอร์ขับเคลื่อนลิฟต์วางอยู่ด้านล่าง ตัวลิฟต์ยึดติดกับโซ่ และเคลื่อนที่ขึ้นลงโดยมี guide rail ช่วยประคองให้ตัวลิฟต์เคลื่อนที่ได้โดยไม่เอียง

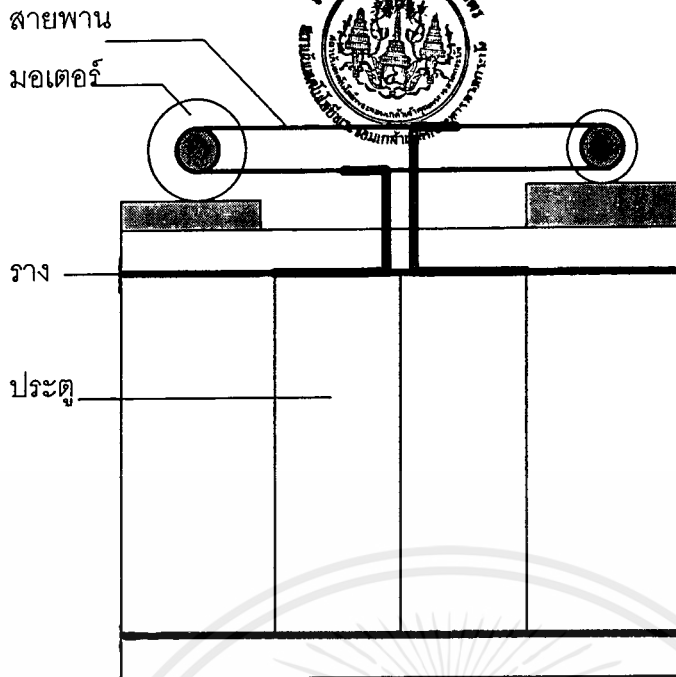


รูปที่ 3.1 แสดงโครงสร้างของระบบจำลองการเคลื่อนที่ของลิฟต์

3.1.2 การขับเคลื่อน

- การขับเคลื่อนลิฟต์ใช้มอเตอร์กระแสตรงขนาดแรงดันสูงสุด 24 โวลต์ กระแสสูงสุด 1.8 แอมแปร์ และเอนโคเดอร์ของมอเตอร์สร้างสัญญาณ pulse จำนวน 500 pulse ต่อหนึ่งรอบการหมุน

- การขับเคลื่อนประตูลิฟต์ใช้มอเตอร์กระแสตรงขนาดแรงดันสูงสุด 12 โวลต์ กระแสสูงสุด 1 แอมแปร์ ควบคุมการเปิดและปิดประตูโดยใช้สายพานดังแสดงในรูปที่ 3.2 และสามารถตรวจสอบการปิดของประตูด้วยสัญญาณจากไมโครสวิตช์ที่ติดตั้งไว้ที่ประตูลิฟต์

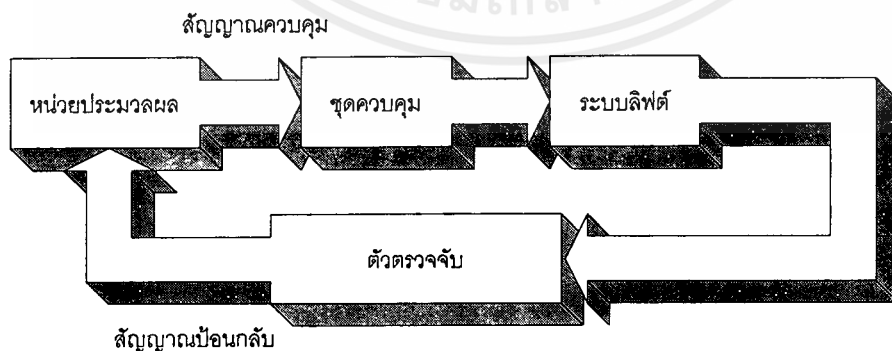


รูปที่ 3.2 แสดงลักษณะของตัวลิฟต์

3.1.3 ที่ระยะต่ำสุดของการเคลื่อนที่มีการติดตั้งไมโครสวิทช์เพื่อทำหน้าที่แจ้งหน่วยประมวลผลกลางว่าลิฟต์เคลื่อนที่ถึงจุดต่ำสุด ซึ่งเป็นจุดเริ่มต้นของชั้นที่ 1

3.2 ระบบโดยรวม (Overall system)

ระบบจำลองการเคลื่อนที่ของลิฟต์ประกอบไปด้วยส่วนสำคัญหลายส่วนซึ่งสามารถแสดงระบบโดยรวมอย่างคร่าวๆด้วยบล็อกไดอะแกรมดังนี้

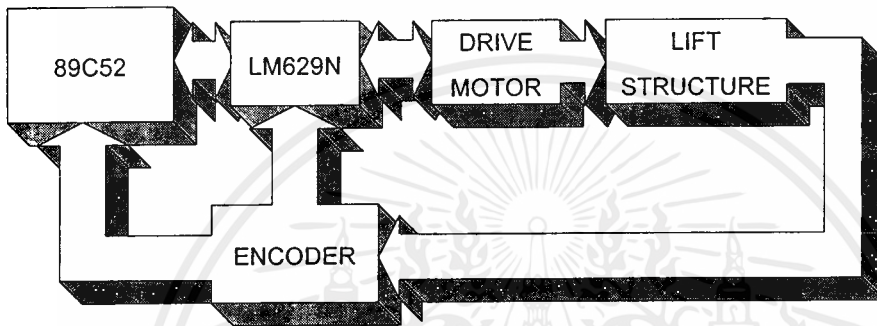


รูปที่ 3.3 แสดงบล็อกไดอะแกรมโดยรวมของระบบควบคุมการเคลื่อนที่ของลิฟต์

3.3 ส่วนการควบคุมและประมวลผล (Data Control and Processing Part)

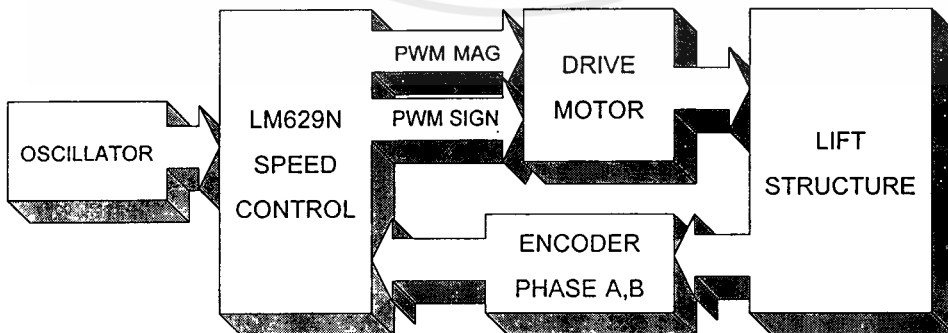
ในส่วนควบคุมและประมวลผลนี้จะรับอินพุตมาจากภายนอกซึ่งอินพุตในที่นี้ก็คือการเลือกการเคลื่อนที่ของลิฟต์ เพื่อทำการประมวลผล และส่งคำสั่งออกไปควบคุมระบบการทำงานอันได้แก่ ส่วนวงจรการขับเคลื่อน วงจรของส่วนควบคุมและประมวลผลนี้สามารถแสดงได้ดังรูป

- ชุดควบคุมการเคลื่อนที่ขึ้น/ลงของตัวลิฟต์
- ชุดควบคุมการเคลื่อนที่ขึ้น/ลงของตัวลิฟต์ประกอบด้วย



รูปที่ 3.4 แสดงบล็อกไดอะแกรมของชุดควบคุมการทำงานของระบบลิฟต์

- ชิปควบคุมความเร็ว LM629N



รูปที่ 3.5 แสดงบล็อกไดอะแกรมการทำงานของชิปควบคุมความเร็ว LM629N

Power ICs LM629N เป็นชิพควบคุมการเคลื่อนที่ (Motion Control IC) ทำงานร่วมกับหน่วยประมวลผลหลัก ซึ่งออกแบบมาให้ใช้ได้ทั้งดีซีเซอร์โวมอเตอร์และระบบเซอร์โวแมคคานิคซึมอื่นๆ มีการป้อนกลับของเฟส A , B ตัว Encoder จะนำค่าเฟสที่ได้มาคำนวณ เพื่อที่จะได้สัญญาณนำไปควบคุมการเคลื่อนที่ของลิฟต์

คุณสมบัติของ LM629N

- มีรีจิสเตอร์ตำแหน่งความเร็ว ความเร่งขนาด 32 บิต
- เป็นฟิลเตอร์ดิจิตอลแบบ PID
- สามารถกำหนดเส้นทางของความเร็ว (trajectory)
- สามารถกำหนดค่าคาบการซีกตัวอย่างได้
- ให้เอาท์พุตเป็น PWM ซึ่งมีทั้งขนาดและทิศทาง
- ค่าความเร็ว ค่าตำแหน่งที่ต้องการและค่าพารามิเตอร์ของฟิลเตอร์สามารถเปลี่ยนแปลงได้ขณะที่ทำการควบคุมจริง
- สามารถทำงานได้ทั้งโหมดการควบคุมตำแหน่งและโหมดควบคุมความเร็ว
- สามารถโปรแกรมการทำงานขัดจังหวะ (interrupt) กับหน่วยประมวลผลหลักได้
- ติดต่อข้อมูลกับหน่วยประมวลผลหลักแบบขนานขนาด 8 บิตได้
- มีหน่วยรับสัญญาณการป้อนกลับจาก Encoder

สามารถดูรายละเอียดได้จากคู่มือการควบคุมและประมวลผล

ตัวอย่างตาราง CONTROL WORD ใน LM629N ที่ถูกเลือกใช้ในโปรแกรม

Filter Control word Bit Allocation

| Bit Position | Function |
|--------------|------------------------------------|
| Bit 15 | Derivative Sampling Interval Bit 7 |
| Bit 14 | Derivative Sampling Interval Bit 6 |
| Bit 13 | Derivative Sampling Interval Bit 5 |
| Bit 12 | Derivative Sampling Interval Bit 4 |
| Bit 11 | Derivative Sampling Interval Bit 3 |
| Bit 10 | Derivative Sampling Interval Bit 2 |
| Bit 9 | Derivative Sampling Interval Bit 1 |
| Bit 8 | Derivative Sampling Interval Bit 0 |
| Bit 7 | Not used |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

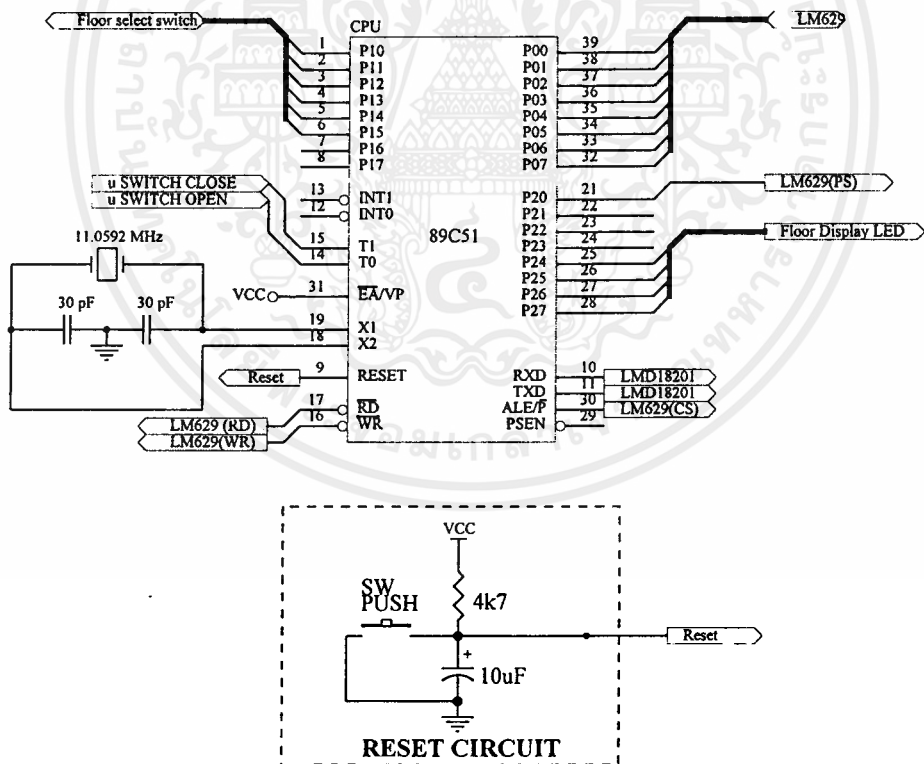
| | |
|-------|-----------------|
| Bit 6 | Not used |
| Bit 5 | Not used |
| Bit 4 | Not used |
| Bit 3 | Loading kp Data |
| Bit 2 | Loading ki Data |
| Bit 1 | Loading kd Data |
| Bit 0 | Loading il Data |

Trajectory Control word Bit Allocation

| Bit Position | Function |
|--------------|--|
| Bit 15 | Not used |
| Bit 14 | Not used |
| Bit 13 | Not used |
| Bit 12 | Forward Direction (velocity Mode Only) |
| Bit 11 | Velocity Mode |
| Bit 10 | Stop Smoothly (decelerate as Programmed) |
| Bit 9 | Stop Abruptly (Maximum deceleration) |
| Bit 8 | Turn off Motor (Output Zero Drive) |
| Bit 7 | Not used |
| Bit 6 | Not used |
| Bit 5 | Acceleration will be Loaded |
| Bit 4 | Acceleration will be Relative |
| Bit 3 | Velocity will be Loaded |
| Bit 2 | Velocity will be Relative |
| Bit 1 | Position will be Loaded |
| Bit 0 | Position will be Relative |

Status Byte Bit Allocation

| Bit Position | Function |
|--------------|-------------------------------------|
| Bit 7 | Motor Off |
| Bit 6 | Breakpoint Reached (Interrupt) |
| Bit 5 | Excessive Position Error(Interrupt) |
| Bit 4 | Wraparound Occurred(Interrupt) |
| Bit 3 | Index Pulse Observed(Interrupt) |
| Bit 2 | Trajectory Complete(Interrupt) |
| Bit 1 | Command Error(Interrupt) |
| Bit 0 | Busy Bit |



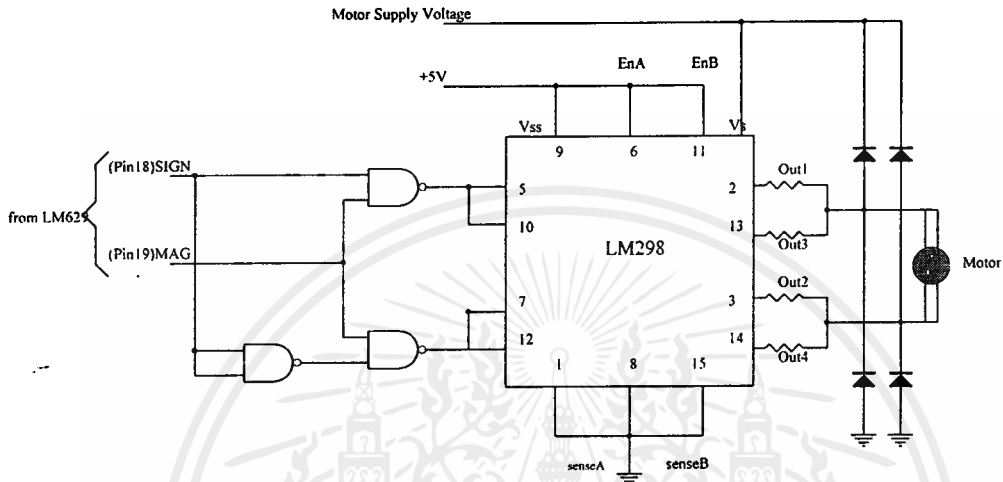
รูปที่ 3.6 แสดงวงจรควบคุมและประมวลผล

3.4 ส่วนการขับเคลื่อน (DC Motor Driver Part)

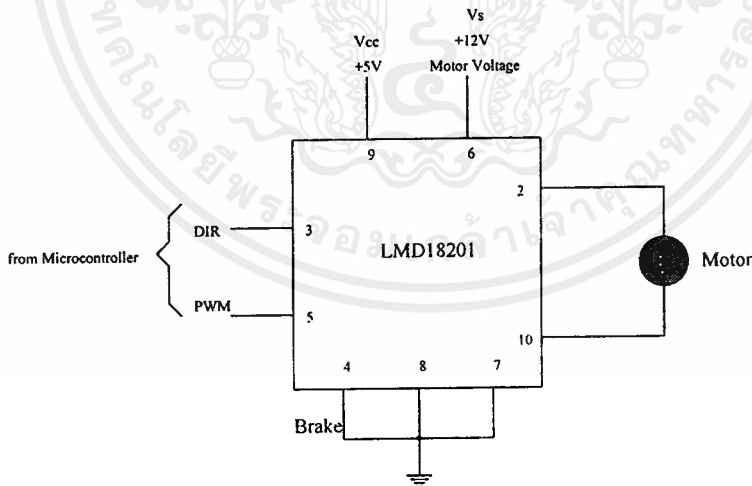
แบ่งเป็น 2 ส่วนคือ

1. มอเตอร์ขับเคลื่อนตัวลัพท์
2. มอเตอร์เปิด-ปิดประตูลัพท์

มีรายละเอียดของวงจรดังรูป



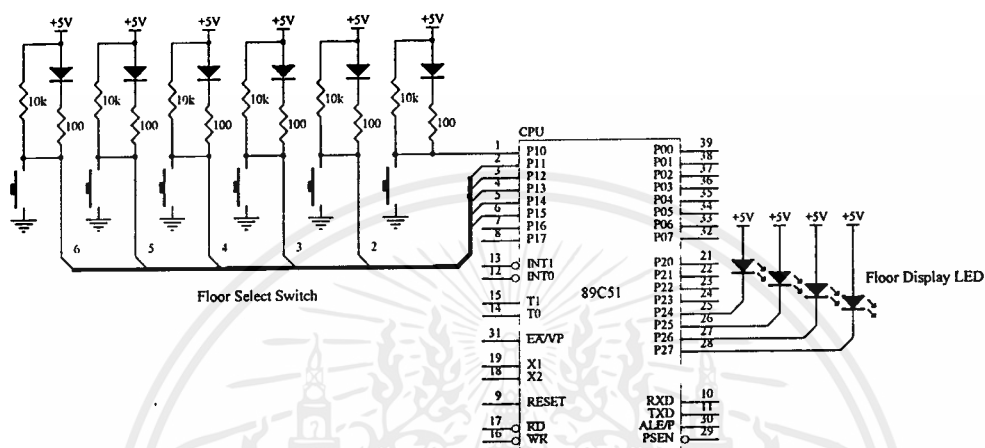
รูปที่ 3.7 แสดงวงจรขับเคลื่อนมอเตอร์ขับเคลื่อนตัวลัพท์



รูปที่ 3.8 แสดงวงจรขับเคลื่อนมอเตอร์เปิด-ปิดประตูลัพท์

3.5 ส่วนรับและแสดงผล (Input and Output Part)

1. ส่วนรับผล คือ ส่วนของสวิตช์ที่ใช้เลือกชั้น
2. ส่วนแสดงผล จะแสดงผลชั้นที่ลิฟต์เคลื่อนที่ไปถึง
ดูรายละเอียดได้จากรูปวงจร

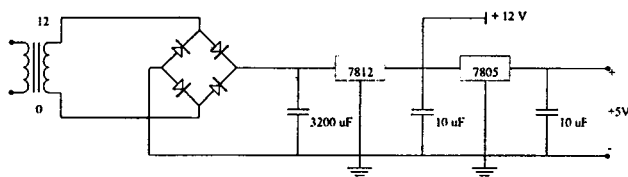


รูปที่ 3.9 แสดงวงจรรับและแสดงผล

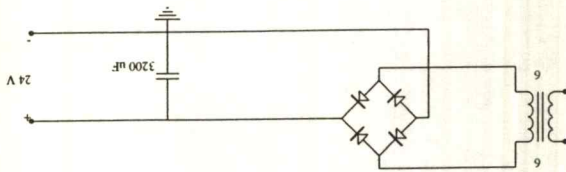
3.6 ส่วนวงจรจ่ายพลังงาน (Power supply circuit part)

แหล่งจ่ายไฟหลักของระบบแบ่งเป็น 2 ส่วน

1. ส่วนแรกให้แรงดันเอาต์พุตขนาด 5 โวลต์ป้อนเป็นไฟเลี้ยงวงจร และแรงดันเอาต์พุตขนาด 12 โวลต์จ่ายให้กับมอเตอร์ที่ทำหน้าที่เปิด-ปิดประตูลิฟต์
2. ส่วนที่ 2 ให้แรงดันเอาต์พุตขนาด 24 โวลต์จ่ายให้กับมอเตอร์ขับเคลื่อนลิฟต์
สามารถดูรายละเอียดได้จากรูปวงจร

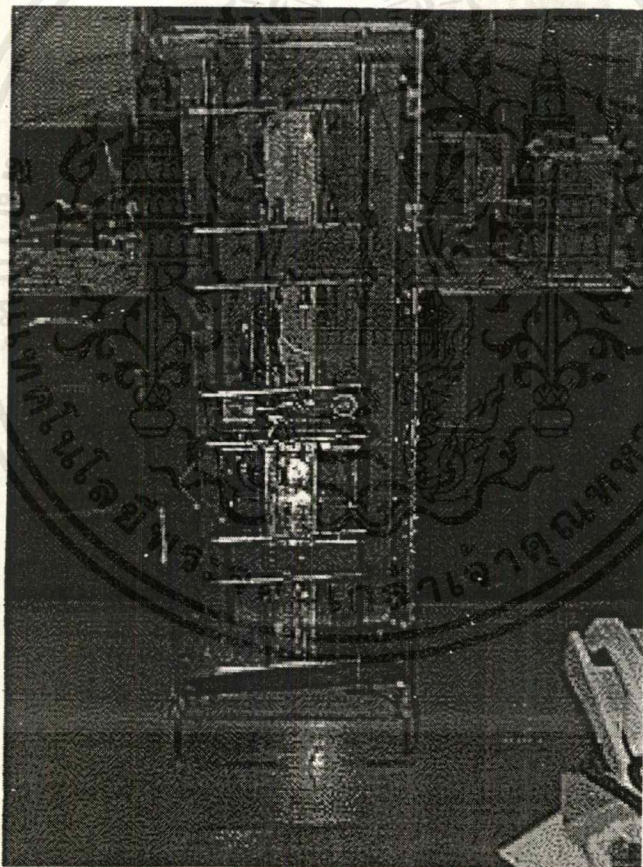


รูปที่ 3.10 แสดงวงจรจ่ายพลังงานส่วนที่ 1

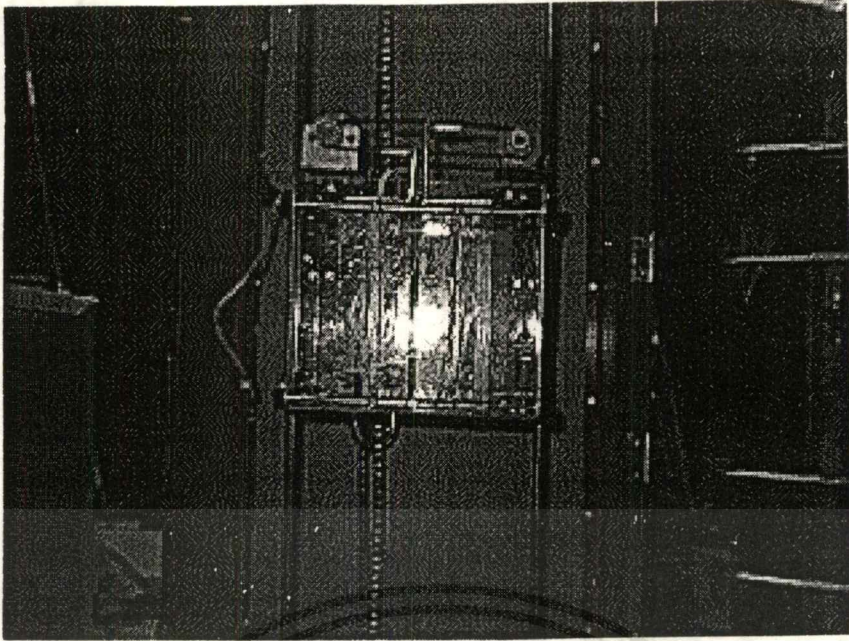


รูปที่ 3.11 แสดงวงจรจ่ายพลังงานส่วนที่ 2

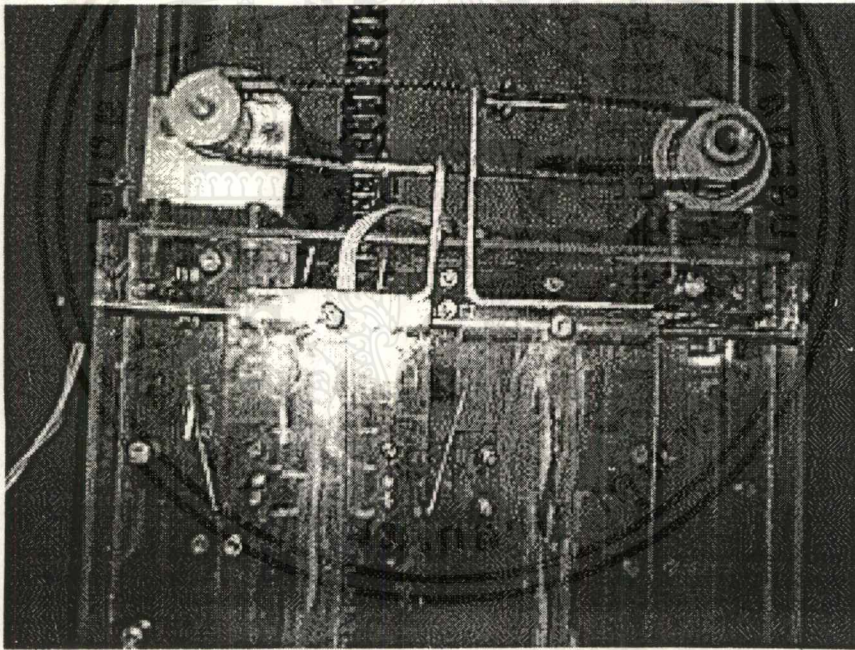
3.7 ระบบใช้งานจริง (Real System)



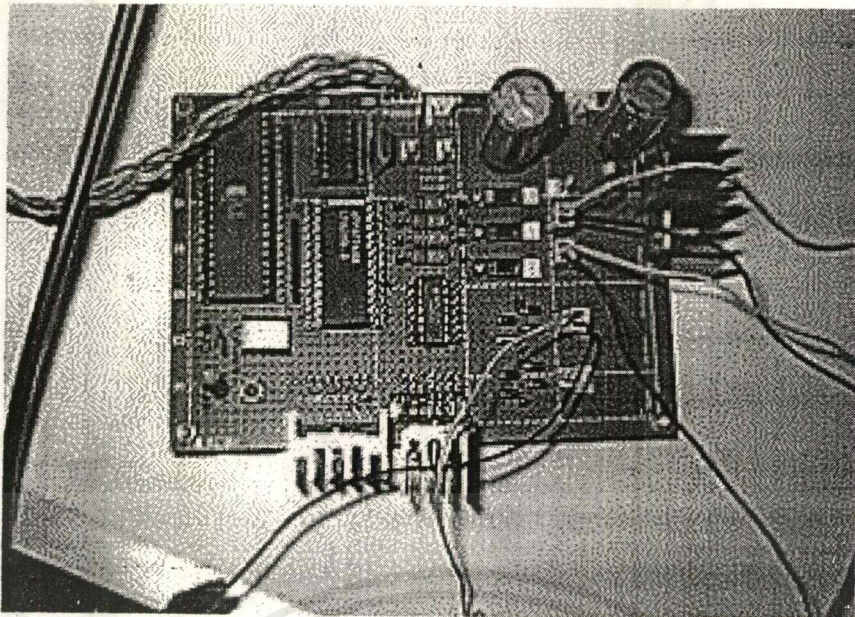
รูปที่ 3.12 แสดงโครงสร้างของลิฟต์



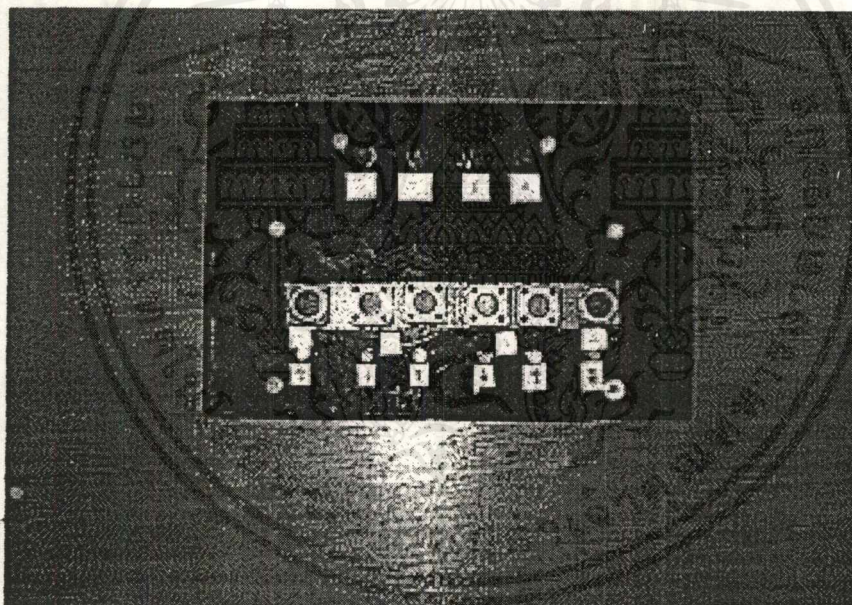
รูปที่ 3.13 แสดงลักษณะตัวลิฟต์



รูปที่ 3.14 แสดงกลไกการเปิด-ปิดประตู



รูปที่ 3.15 แสดงวงจรควบคุมและประมวลผล



รูปที่ 3.16 แสดงวงจรับและแสดงผล

บทที่ 4

การทดลองและผลการทดลอง

การที่ระบบลิฟต์จะมีการทำงาน ที่มีประสิทธิภาพและความแม่นยำสูงได้นั้นขึ้นอยู่กับระบบทั้ง 3 ส่วน ดังนี้

1. โครงสร้างของระบบลิฟต์
2. โปรแกรมที่เขียนขึ้นเพื่อใช้ควบคุมระบบ
3. ชุดควบคุมการขับเคลื่อนมอเตอร์

โดยชุดควบคุมการขับเคลื่อนมอเตอร์ในระบบนี้ใช้ LM629N เป็นตัวควบคุมซึ่ง LM629N จะต้องมีการตั้งค่าพารามิเตอร์ต่างๆเพื่อสามารถทำการควบคุมการขับเคลื่อนมอเตอร์ให้ทำงานได้อย่างถูกต้องและแม่นยำ

4.1 การทดลอง

เนื่องจากการใช้งาน LM629N จำเป็นต้องตั้งค่าพารามิเตอร์ K_p , K_i , K_d เพื่อทำการควบคุมชุดขับเคลื่อนมอเตอร์ ดังนั้นจึงกำหนดพารามิเตอร์ของระบบที่ต้องการควบคุมและทำการปรับเปลี่ยนค่าพารามิเตอร์ K_p , K_i , K_d เพื่อดูผลการตอบสนองของระบบว่าเป็นอย่างไร โดยใช้วิธี Trial and Error ในการปรับเปลี่ยนค่าพารามิเตอร์

กำหนดให้ระบบที่เราต้องการควบคุมมีรายละเอียดดังนี้

ค่าความเร่งสูงสุด (A) : 0FFFFH

ค่าความเร็วสูงสุด (V) : 0FFFFH

จำนวนรอบที่หมุน (P) : 0FFFFH

K_p : 01FFH

K_i : 00H

K_d : 00H

Integration Limit (il) : 0FFH

4.2 ผลการทดลอง

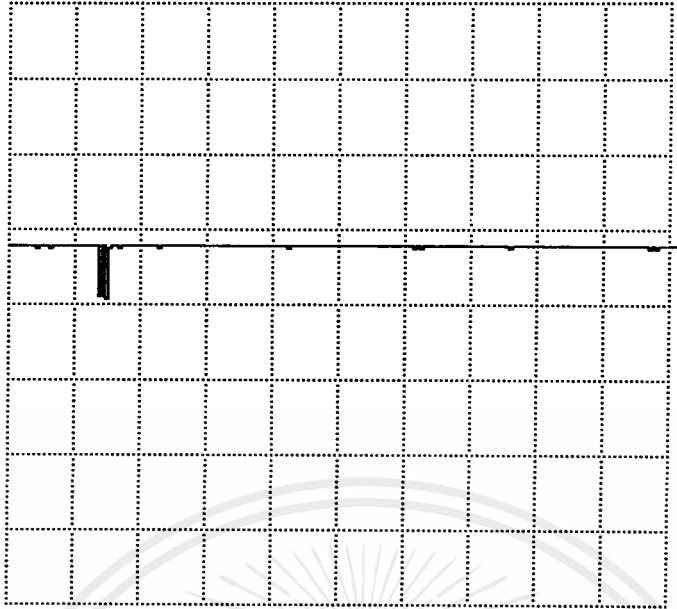
ได้ทำการทดลองทั้งหมด 3 การทดลองด้วยกัน คือ

การทดลองที่ 1 กำหนดให้พารามิเตอร์ของ LM629N เป็นดังนี้

K_p : 01FFH , K_i : 00H , K_d : 00H

ซึ่งได้ผลการทดลองของสัญญาณเอาต์พุตดังแสดงในรูปที่ 4.1

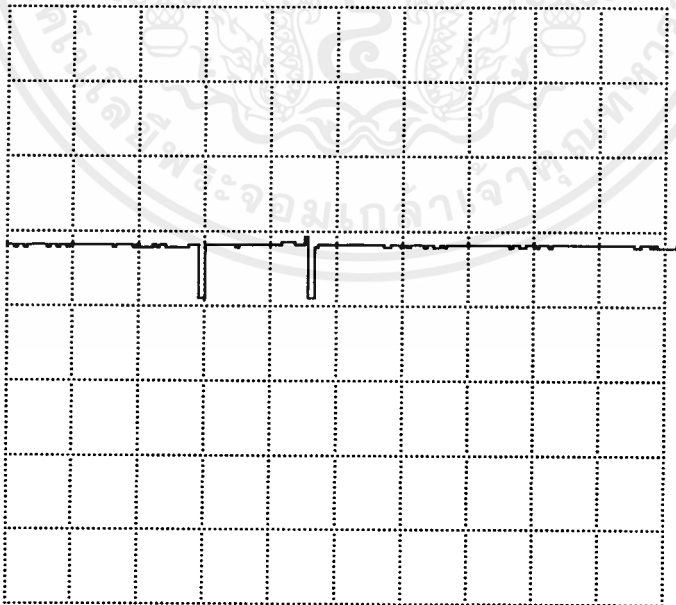
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 สัญญาณ เอาท์พุทที่ได้จากการทดลองที่ 1
การทดลองที่ 2 กำหนดให้พารามิเตอร์ของ LM629N เป็นดังนี้

$K_p : 01FFH$, $K_i : 0FFH$, $K_d : 00H$

ซึ่งได้ผลการทดลองของสัญญาณเอาท์พุทดังแสดงในรูปที่ 4.2



รูปที่ 4.2 สัญญาณ เอาท์พุทที่ได้จากการทดลองที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 3 กำหนดค่าพารามิเตอร์ของ LM629N เป็นดังนี้

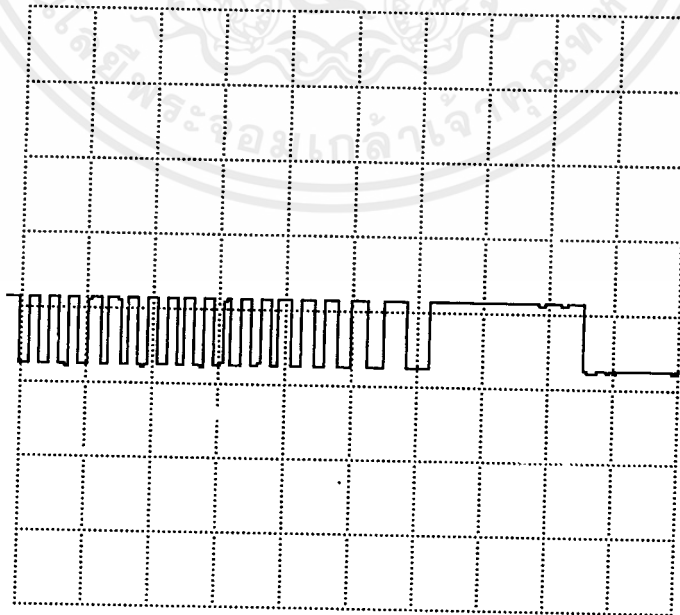
Kp : 01FFH , Ki : 30H , Kd : 05H

ซึ่งได้ผลการทดลองของสัญญาณเอาต์พุตดังแสดงในรูปที่ 4.3



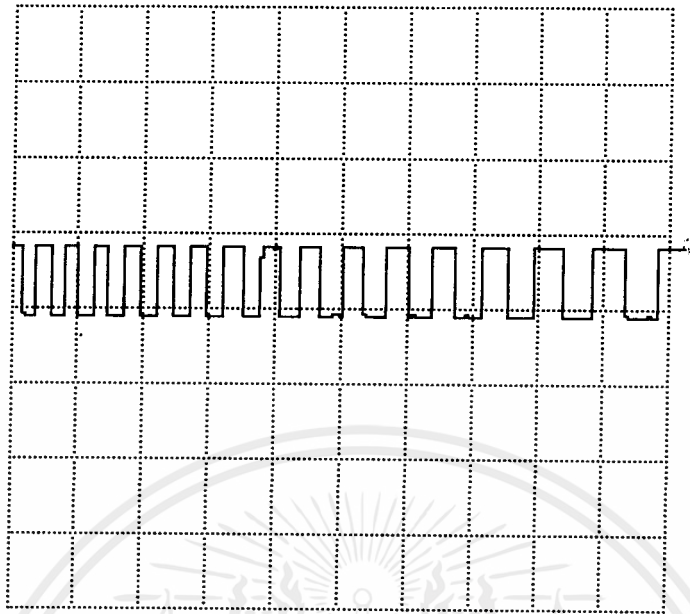
รูปที่ 4.3 สัญญาณ เอาต์พุตที่ได้จากการทดลองที่ 3

นอกจากนี้ได้ทำการวัดสัญญาณ Phase A และ Phase B จากเอนโคเดอร์ซึ่งได้แสดงในรูปที่ 4.4 และ 4.5 ตามลำดับ



รูปที่ 4.4 แสดงสัญญาณ PHASE A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงสัญญาณ PHASE B

บทที่ 5

บทสรุป

5.1 ปัญหา

ปัญหาที่พบในการทำงานของโครงการชิ้นนี้ สามารถแบ่งได้เป็น

1. ปัญหาในการจัดทำโครงสร้างของระบบจำลอง เนื่องจากเป็นงานที่ต้องการความละเอียดและความแม่นยำในการประกอบสูง แต่อุปกรณ์ในการทำที่มีไม่สามารถให้ความเที่ยงตรงตามที่ต้องการได้จึงทำให้เกิดข้อผิดพลาด ทำให้ระบบจำลองทั้งในส่วนของเคลื่อนที่ขึ้น-ลง และส่วนกลไกการเปิด-ปิดประตูทำงานได้ไม่ดีเท่าที่ควร
2. ปัญหาในการเลือกใช้อุปกรณ์ในการทำวงจร เนื่องจากอุปกรณ์ที่เลือกใช้ไม่เป็นที่แพร่หลาย บางครั้งจึงทำให้ยากต่อการหาข้อมูล และรายละเอียดที่ต้องการ
3. ปัญหาเรื่องการติดต่อกันระหว่างไมโครคอนโทรลเลอร์ 89C51 และชิพควบคุมความเร็ว LM629N เนื่องจากชิพทั้ง 2 ตัวนี้ไม่สามารถติดต่อกันโดยตรงได้ (DIRECT ADDRESSING MODE) จึงต้องทำการติดต่อกันโดยอ้อม (INDIRECT ADDRESSING MODE) การสร้างสัญญาณ CHIP SELECT ,READ ,WRITE ให้กับ LM629N จึงไม่จำเป็นต้องสร้างจากตัวโปรแกรมแต่สามารถสร้างได้จากวงจรภายในของไมโครคอนโทรลเลอร์เลย สัญญาณที่ได้นั้นจะมี TIMING DIAGRAM ที่สอดคล้องกับ LM629N มากกว่าที่สร้างจากตัวโปรแกรม

5.2 แนวทางในการพัฒนาและปรับปรุง

1. ปรับปรุงโครงสร้างของระบบจำลองให้มีความสามารถในการทำงานเพิ่มมากขึ้น เช่น โดยการติดตั้งอุปกรณ์เพื่อความปลอดภัยเช่นเดียวกับในระบบลิฟต์จริง รวมทั้งปรับเปลี่ยนอุปกรณ์ที่ใช้ในการทำงานให้ดียิ่งขึ้น
2. ปรับปรุงส่วนของวงจร เช่น เพิ่มวงจรที่จะทำให้ระบบจำลองสามารถควบคุมได้โดย PLC หรือทำการติดต่อกับคอมพิวเตอร์ เพราะใช้งานได้กว้างกว่า สะดวกกว่า เป็นต้น
3. ปรับปรุงส่วนของโปรแกรมให้เหมือนลิฟต์จริง มีการคำนวณอินพุตที่รับกับการจอดลิฟต์ในแต่ละชั้นเพื่อให้เหมือนจริงยิ่งขึ้น

เอกสารอ้างอิง

- 1.Siam Elevator, (HITACHI)," HITACHI Elevator",Siam Elevator co.,ltd,1990
- 2.National Semiconductor,"National Power Ics",National Semiconductor,1995
- 3.Atmel."Atmel Flash Microcontrollers",Atmel
- 4.ETT ,"Microprocessor Data Book MCS-51 Microcontrollers",ETT co.,ltd

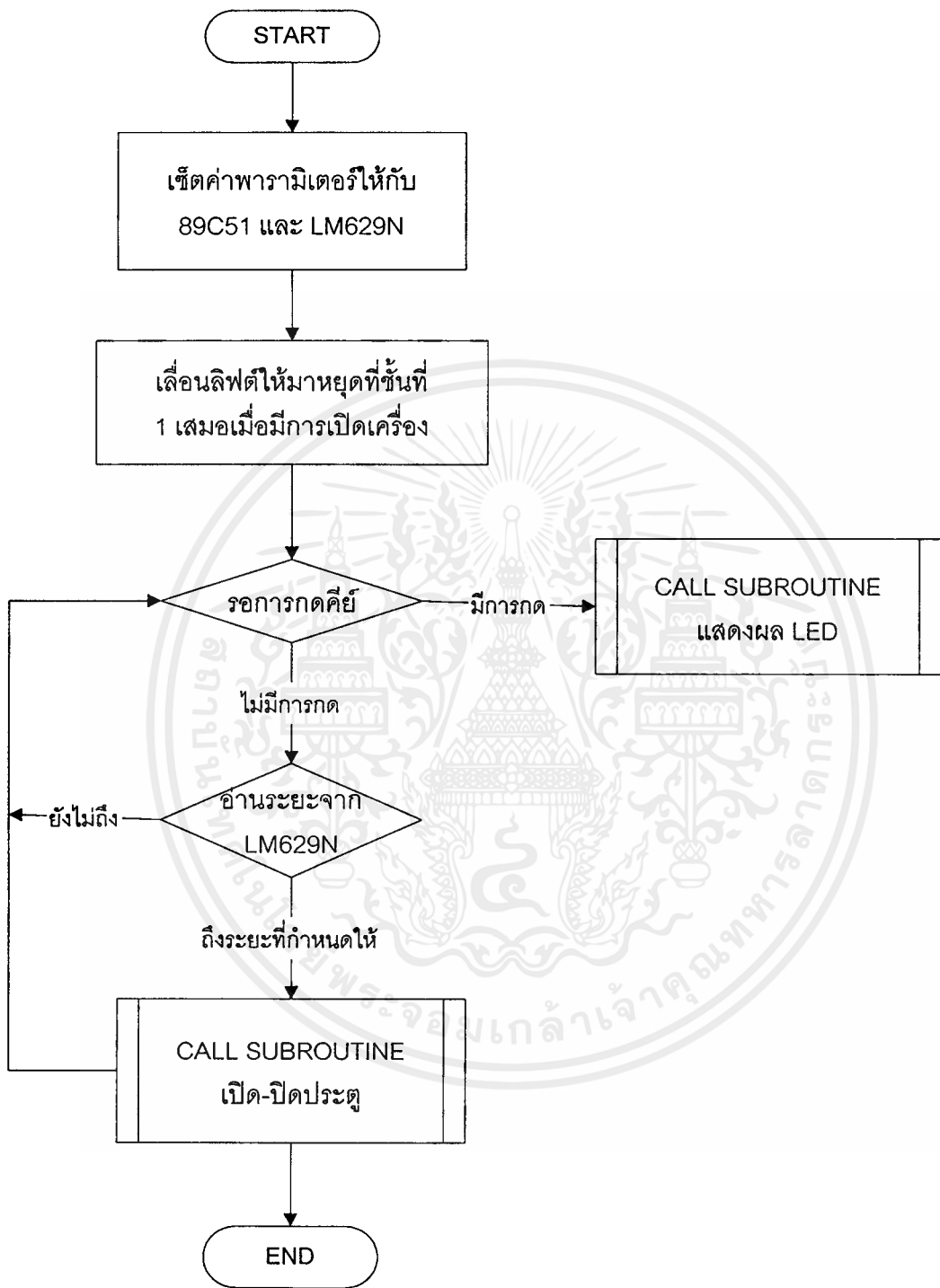


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

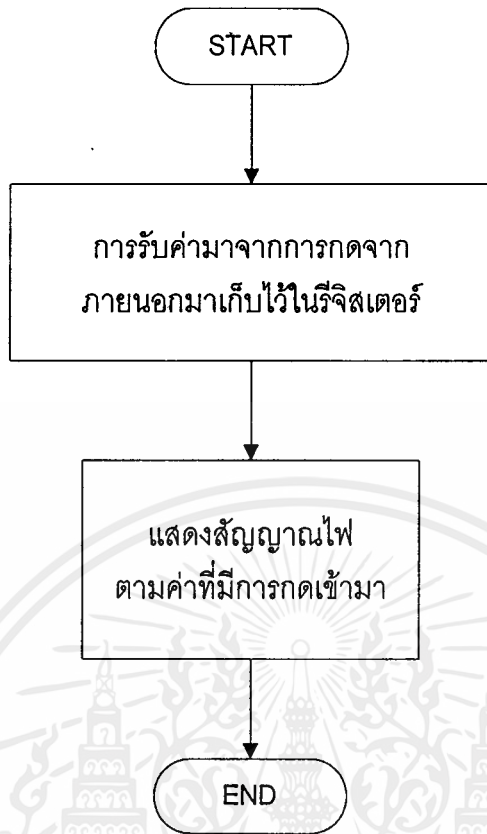
ภาคผนวก ก



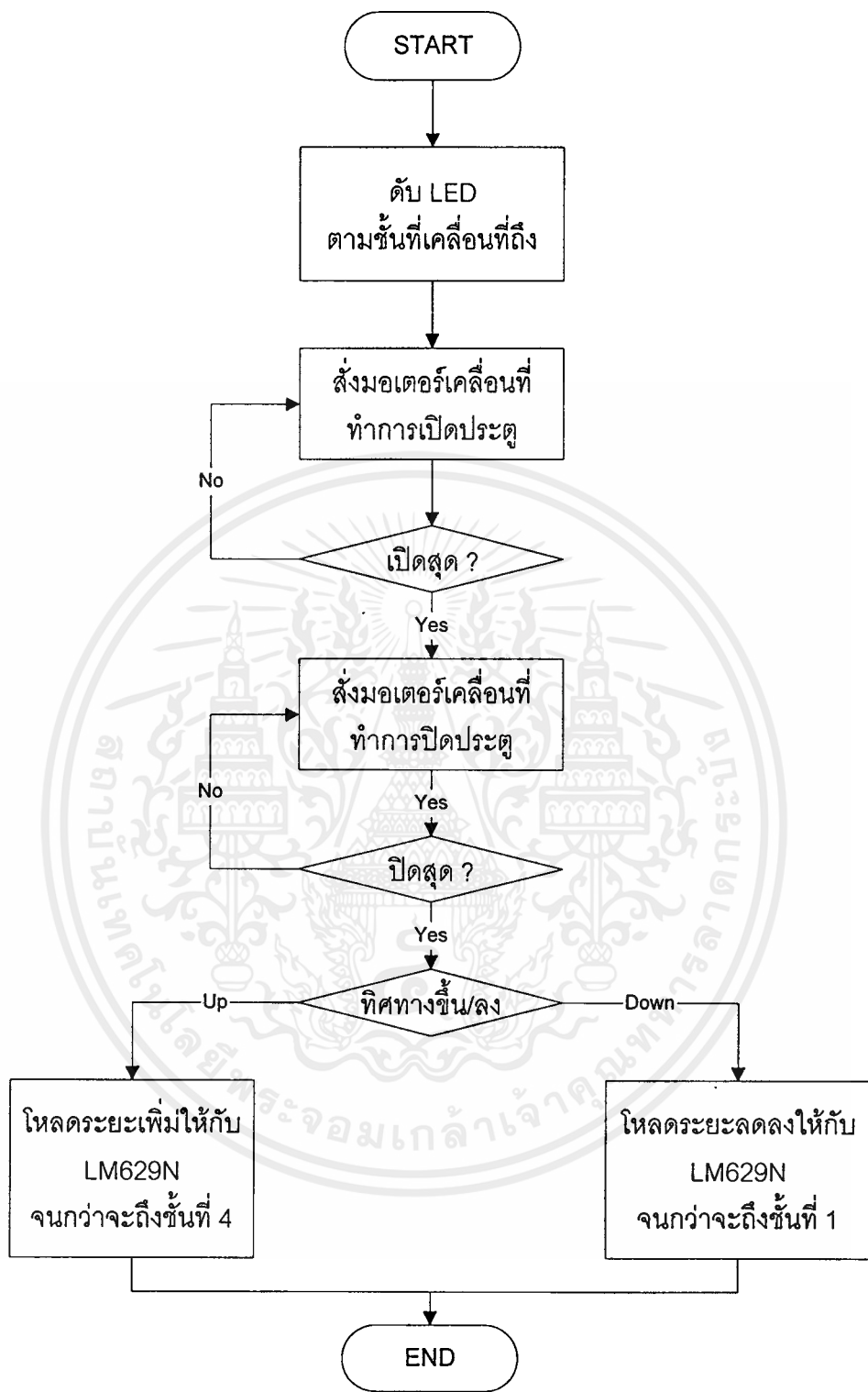
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART แสดงการทำงานของโปรแกรมทั้งหมด



FLOW CHART แสดง SUBROUTINE แสดงผล LED



FLOW CHART แสดง SUBROUTINE เปิด-ปิดประตู

```

;FILE NAME      A.ASM
;DESCRIPTION    PROGRAM TEST
;HARDWARE      89C51
;DD-MM-YY      07-04-98

                ORG 0000H

BANK:          DS 32
DIR:           DS 1
STORY:         DS 1
SWITCH:        DS 1
FIL:           DS 10
TRJ:           DS 14
SYSSTACK:      DS 32

RESETCODE     EQU 00H
LFIL          EQU 1EH
UDF           EQU 04H
DFH           EQU 02H
LTRJ          EQU 1FH
STT           EQU 01H
RDRP          EQU 0AH

R_COMMAND     EQU 0000H
R_DATA        EQU 0100H

                ORG 0000H

                AJMP R_RESET

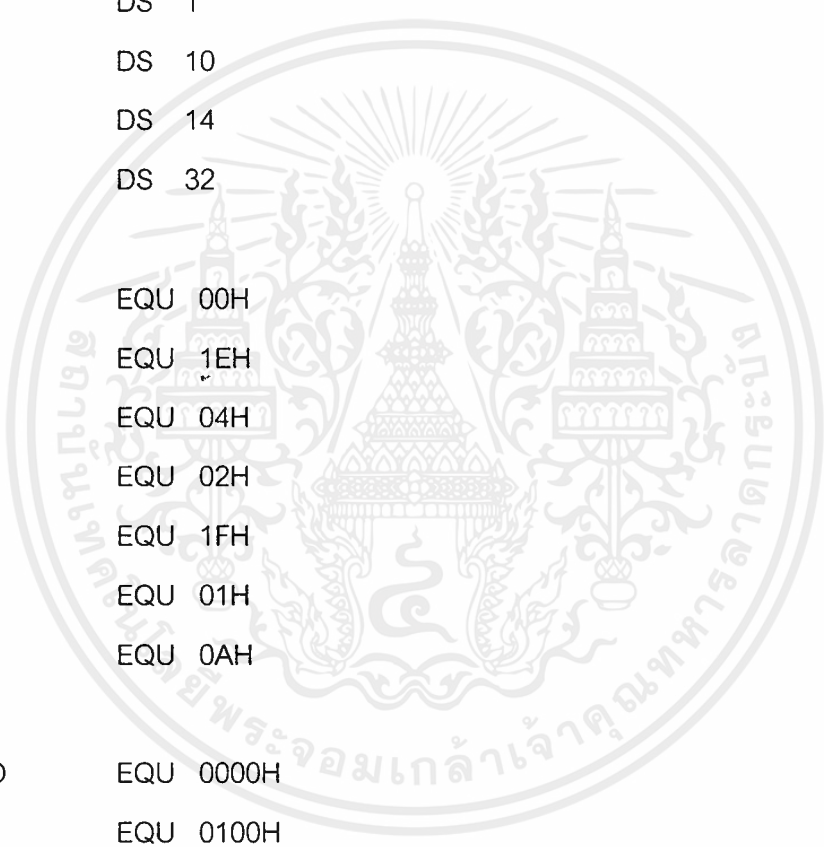
R_RESET:      MOV R0,#0FFH

LOOP1:        MOV R1,#0FFH

                DJNZ R1,$

                DJNZ R0,LOOP1

```



```
MOV IE,#00H
MOV SWITCH,#0FFH
MOV DIR,#00H
MOV FIL+0,#00H ;FILTER CONTROL WORD
MOV FIL+1,#00001111B
MOV FIL+2,#01H ;KP
MOV FIL+3,#0FFH
MOV FIL+4,#00H ;KI
MOV FIL+5,#00H
MOV FIL+6,#00H ;KD
MOV FIL+7,#00H
MOV FIL+9,#00H ;il
MOV FIL+10,#0FFH
MOV TRJ+0,#00H ;TRAJECTORY CONTROL WORD
MOV TRJ+1,#00101010B
MOV TRJ+2,#00H ;A
MOV TRJ+3,#00H
MOV TRJ+4,#0FFH
MOV TRJ+5,#0FFH
MOV TRJ+6,#00H ;V
MOV TRJ+7,#00H
MOV TRJ+8,#0FFH
MOV TRJ+9,#0FFH
MOV TRJ+10,#0C0H ;P
MOV TRJ+11,#00H
MOV TRJ+12,#00H
MOV TRJ+13,#00H
LCALL LM629
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SET_DFH:      SETB P3.2
              JB  P3.2,SET_DFH
              MOV TRJ+0,#00011001B
              MOV TRJ+1,#00101010B
              MOV TRJ+10,#00H
              MOV TRJ+11,#00H
              MOV TRJ+12,#00H
              MOV TRJ+13,#00H
              LCALL R_LTRJ
              LCALL R_STT
              LCALL R_DFH
              SETB DIR.7
              MOV STORY,#00H
GET_KEY3:     MOV P1,#0FFH
              MOV A,P1
              ANL A,#11011111B
              CJNE A,#11011111B,TEST
              LJMP GET_KEY3
TEST:        SETB ACC.5
              MOV SWITCH,A
              MOV P1,SWITCH
              LJMP OPN_DOOR

```

;..... TURN OFF LED

```

R_LED:       MOV A,STORY
              CJNE A,#01H,CHK2
              SETB SWITCH.5
              SJMP LED_END

```

```

CHK2:      MOV  A,STORY
           CJNE A,#02H,CHK3
           ORL  SWITCH,#00011000B
           SJMP LED_END

CHK3:      MOV  A,STORY
           CJNE A,#03H,CHK4
           ORL  SWITCH,#00000110B
           SJMP LED_END

CHK4:      MOV  A,STORY
           CJNE A,#04H,LED_END
           SETB SWITCH.0

LED_END:   MOV  P1,SWITCH
           RET

;..... ROUTINE OF DOOR .....
RUP:      LJMP R_UP
OPN_DOOR: CLR  P3.0
           SETB P3.1
           JB   P3.4,OPN_DOOR
           MOV  R0,#0FFH

LOOP3:    MOV  R1,#0FFH
           DJNZ R1,$
           DJNZ R0,LOOP3

CLS_DOOR: SETB P3.0
           SETB P3.1
           JB   P3.5,CLS_DOOR
           JB   DIR.7,RUP

R_DW:     DEC  STORY
           LCALL R_LED
           MOV  A,STORY
           CJNE A,#02H,LD_DW
           MOV  TRJ+0,#00H

```

```

MOV TRJ+1,#00101010B
MOV TRJ+10,#0C0H
MOV TRJ+11,#00H
MOV TRJ+12,#00H
MOV TRJ+13,#00H

LCALL R_LTRJ
LCALL R_STT
R_SDFH: SETB P3.2
JB P3.2,GET_KEY2
MOV TRJ+0,#00011001B
MOV TRJ+1,#00101010B
MOV TRJ+10,#00H
MOV TRJ+11,#00H
MOV TRJ+12,#00H
MOV TRJ+13,#00H
LCALL R_LTRJ
LCALL R_STT
LCALL R_DFH
SETB DIR.7
MOV STORY,#00H
CLR SWITCH.5
MOV A,SWITCH
CJNE A,#11011111B,OPN_DOOR
GET_KEY2: MOV P1,#0FFH
MOV A,P1
ANL SWITCH,A
MOV P1,SWITCH
SJMP R_SDFH

```

```

LD_DW:      CLR DIR.7
            MOV TRJ+0,#00H
            MOV TRJ+1,#00101010B
            MOV A,STORY
            CJNE A,#04H,DW2
            MOV TRJ+10,#00H
            MOV TRJ+11,#00H
            MOV TRJ+12,#11H
            MOV TRJ+13,#30H
            LCALL N_DW
DW2:        CJNE A,#03H,N_DW
            MOV TRJ+10,#00H
            MOV TRJ+11,#00H
            MOV TRJ+12,#08H
            MOV TRJ+13,#98H
N_DW:       LCALL R_LTRJ
            LCALL R_STT
            AJMP GET_KEY
R_UP:       INC STORY
            LCALL R_LED
            MOV A,STORY
            CJNE A,#04H,LD_UP
            AJMP LD_DW
LD_UP:      SETB DIR.7
            MOV TRJ+0,#00H
            MOV TRJ+1,#00101010B
            MOV A,STORY
            CJNE A,#01H,UP2
            MOV TRJ+10,#00H
            MOV TRJ+11,#00H
            MOV TRJ+12,#08H

```

```
MOV TRJ+13,#98H
LJMP N_UP
```

```
UP2:    CJNE A,#02H,UP3
        MOV TRJ+10,#00H
        MOV TRJ+11,#00H
        MOV TRJ+12,#11H
        MOV TRJ+13,#30H
        LJMP N_UP
```

```
UP3:    CJNE A,#03H,LD_DW
        MOV TRJ+10,#00H
        MOV TRJ+11,#00H
        MOV TRJ+12,#19H
        MOV TRJ+13,#0C8H
```

```
N_UP:   LCALL R_LTRJ
        LCALL R_STT
        AJMP GET_KEY
```

```
;..... MAIN PROGRAM .....
```

```
GET_KEY: MOV P1,#0FFH
        MOV A,P1
        ANL SWITCH,A
        MOV P1,SWITCH
        MOV DPTR,#R_COMMAND
        MOV A,#RDRP
        MOVX @DPTR,A
        LCALL R_BUSY
        MOV DPTR,#R_DATA
        MOVX A,@DPTR
        MOV R0,A
```

```

MOV DPTR,#R_DATA
MOVX A,@DPTR
MOV R1,A
LCALL R_BUSY
MOV DPTR,#R_DATA
MOVX A,@DPTR
MOV R2,A
MOV DPTR,#R_DATA
MOVX A,@DPTR
MOV R3,A
LCALL R_BUSY
MOV A,R3
CJNE A,TRJ+13,GET_KEY
MOV A,R2
CJNE A,TRJ+12,GET_KEY
MOV A,R1
CJNE A,TRJ+11,GET_KEY
MOV A,R0
CJNE A,TRJ+10,GET_KEY
LJMP OPN_DOOR

```

;..... SUBROUTINE

```

R_BUSY:      MOV DPTR,#R_COMMAND
             MOVX A,@DPTR
             JB  ACC.0,R_BUSY
             RET

```

```

R_DFH:      MOV DPTR,#R_COMMAND
             MOV A,#DFH
             MOVX @DPTR,A
             LCALL R_BUSY
             RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

R_LFIL:    MOV  DPTR,#R_COMMAND

           MOV  A,#LFIL

           MOVX @DPTR,A

           LCALL R_BUSY

           MOV  DPTR,#R_DATA    ;FILTER CONTROL WORD

           MOV  A,FIL+0

           MOVX @DPTR,A

           MOV  DPTR,#R_DATA

           MOV  A,FIL+1

           MOVX @DPTR,A

           LCALL R_BUSY

           MOV  DPTR,#R_DATA    ;KP

           MOV  A,FIL+2

           MOVX @DPTR,A

           MOV  DPTR,#R_DATA

           MOV  A,FIL+3

           MOVX @DPTR,A

           LCALL R_BUSY

           MOV  DPTR,#R_DATA    ;KI

           MOV  A,FIL+4

           MOVX @DPTR,A

           MOV  DPTR,#R_DATA

           MOV  A,FIL+5

           MOVX @DPTR,A

           LCALL R_BUSY

           MOV  DPTR,#R_DATA    ;KD

           MOV  A,FIL+6

           MOVX @DPTR,A

           MOV  DPTR,#R_DATA

           MOV  A,FIL+7

           MOVX @DPTR,A

```

```
LCALL R_BUSY
MOV DPTR,#R_DATA ;il
MOV A,FIL+8
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,FIL+9
MOVX @DPTR,A
LCALL R_BUSY
RET
```

```
R_UDF: MOV DPTR,#R_COMMAND
MOV A,#UDF
MOVX @DPTR,A
LCALL R_BUSY
RET
```

```
R_LTRJ: MOV DPTR,#R_COMMAND
MOV A,#LTRJ
MOVX @DPTR,A
LCALL R_BUSY
MOV DPTR,#R_DATA ;TRJ CONTROL WORD
MOV A,TRJ+0
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,TRJ+1
MOVX @DPTR,A
LCALL R_BUSY
MOV DPTR,#R_DATA ;A
MOV A,TRJ+2
MOVX @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV DPTR,#R_DATA
MOV A,TRJ+3
MOVX @DPTR,A
LCALL R_BUSY
MOV DPTR,#R_DATA
MOV A,TRJ+4
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,TRJ+5
MOVX @DPTR,A
LCALL R_BUSY
MOV DPTR,#R_DATA ;V
MOV A,TRJ+6
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,TRJ+7
MOVX @DPTR,A
LCALL R_BUSY
MOV DPTR,#R_DATA
MOV A,TRJ+8
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,TRJ+9
MOVX @DPTR,A
LCALL R_BUSY
MOV DPTR,#R_DATA ;P
MOV A,TRJ+10
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,TRJ+11
MOVX @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCALL R_BUSY
MOV DPTR,#R_DATA
MOV A,TRJ+12
MOVX @DPTR,A
MOV DPTR,#R_DATA
MOV A,TRJ+13
MOVX @DPTR,A
LCALL R_BUSY
RET
R_STT: MOV DPTR,#R_COMMAND
MOV A,#STT
MOVX @DPTR,A
LCALL R_BUSY
RET
LM629: MOV DPTR,#R_COMMAND
MOV A,#RESETCODE
MOVX @DPTR,A
LCALL R_BUSY
MOV R0,#0FFH
LOOP2: MOV R1,#0FFH
DJNZ R1,$
DJNZ R0,LOOP2
LCALL R_LFIL
LCALL R_UDF
LCALL R_LTRJ
LCALL R_STT
RET
END

```

ภาคผนวก ข



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
Endurance: 1,000 Write/Erase Cycles
Data Retention: 10 Years
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Five Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five source two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is

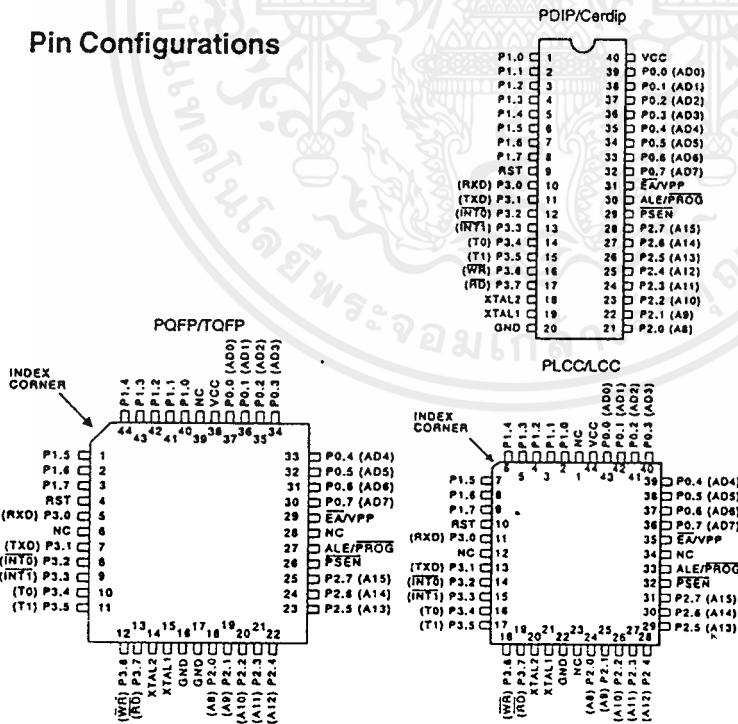


8-Bit Microcontroller with 4 Kbytes Flash

AT89C51

continued

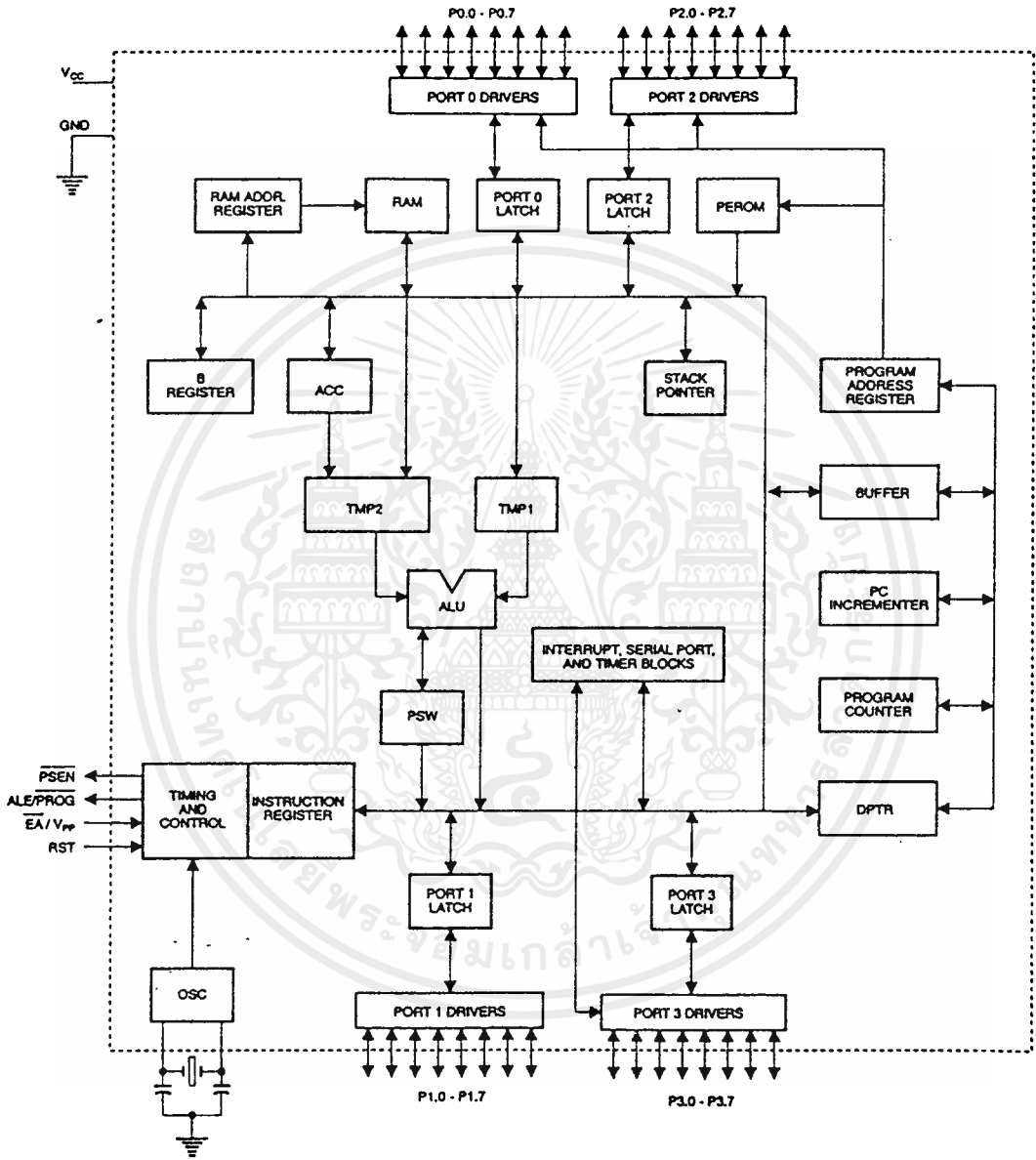
Pin Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AT89C51

Description (Continued)

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 system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional 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 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. 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 pullups 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 pullups.

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. 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 pullups 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 pullups.

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 it uses strong internal pullups 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.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. 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 pullups 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 pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

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

Port 3 also receives some control signals for Flash programming and programming verification.

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 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 microcontroller is in external execution mode.

PSEN

Program Store Enable is the read strobe to external program memory.

When the AT89C51 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 program 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, for parts that require 12-volt V_{PP}.

continued





Pin Description (Continued)

XTAL1

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

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. 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 2. 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 maximum voltage high and low time specifications must be observed.

Idle Mode

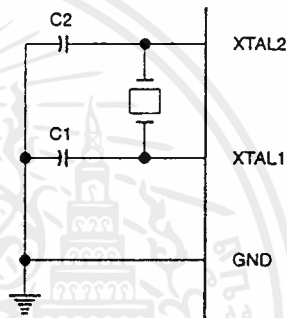
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle 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 is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

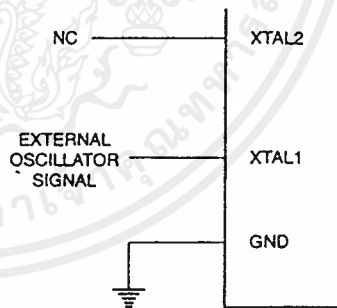
In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF \pm 10 pF for Crystals
= 40 pF \pm 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

| 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 |

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the \overline{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. It is necessary that the latched value of \overline{EA} be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

| Program Lock Bits | | | | Protection Type |
|-------------------|-----|-----|---|--|
| LB1 | LB2 | LB3 | | |
| 1 | U | U | U | No program lock features. |
| 2 | P | U | U | MOV _C 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 is disabled. |
| 3 | P | P | U | Same as mode 2, also verify is disabled. |
| 4 | P | P | P | Same as mode 3, also external execution is disabled. |

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

| | $V_{pp} = 12\text{ V}$ | $V_{pp} = 5\text{ V}$ |
|---------------|--|--|
| Top-Side Mark | AT89C51 xxxx yyww | AT89C51 xxxx-5 yyww |
| Signature | (030H)=1EH (031H)=51H (032H)=FFH | (030H)=1EH (031H)=51H (032H)=05H |

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.

4. Raise \overline{EA}/V_{pp} to 12 V for the high-voltage programming mode.
5. Pulse $\overline{ALE}/\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. 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 can also be monitored by the RDY/BSY output signal. P3.4 is pulled low after \overline{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 data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding $\overline{ALE}/\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,





031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

- (030H) = 1EH indicates manufactured by Atmel
- (031H) = 51H indicates 89C51
- (032H) = FFH indicates 12 V programming
- (032H) = 05H indicates 5 V programming

Programming Interface

Every code byte in the Flash array 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 microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

| Mode | RST | PSEN | ALE/ PROG | EA/ V _{PP} | P2.6 | P2.7 | P3.6 | P3.7 |
|---------------------|-----|------|----------------|------------------------|------|------|------|------|
| Write Code Data | H | L | | H/12V ⁽¹⁾ | L | H | H | H |
| Read Code Data | H | L | H | H | L | L | H | H |
| Write Lock Bit - 1 | H | L | | H/12V | H | H | H | H |
| Bit - 2 | H | L | | H/12V | H | H | L | L |
| Bit - 3 | H | L | | H/12V | H | L | H | L |
| Chip Erase | H | L | ⁽²⁾ | H/12V | H | L | L | L |
| Read Signature Byte | H | L | H | H | L | L | L | L |

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

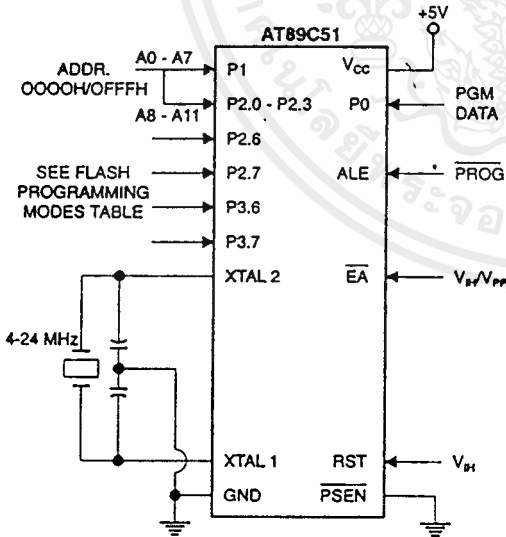
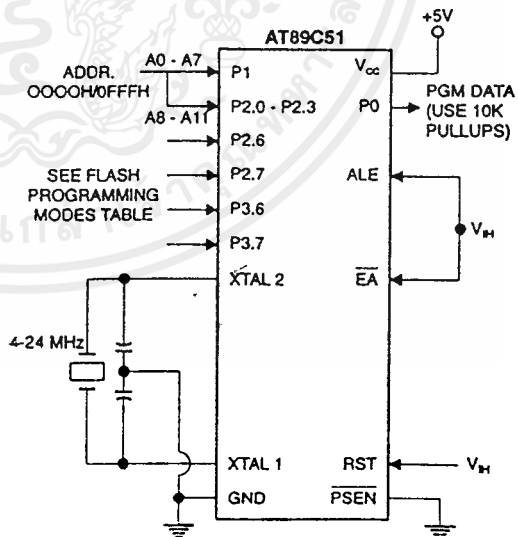


Figure 4. Verifying the Flash



AT89C51

Flash Programming and Verification Characteristics

$T_A = 21^\circ\text{C}$ to 27°C , $V_{CC} = 5.0 \pm 10\%$

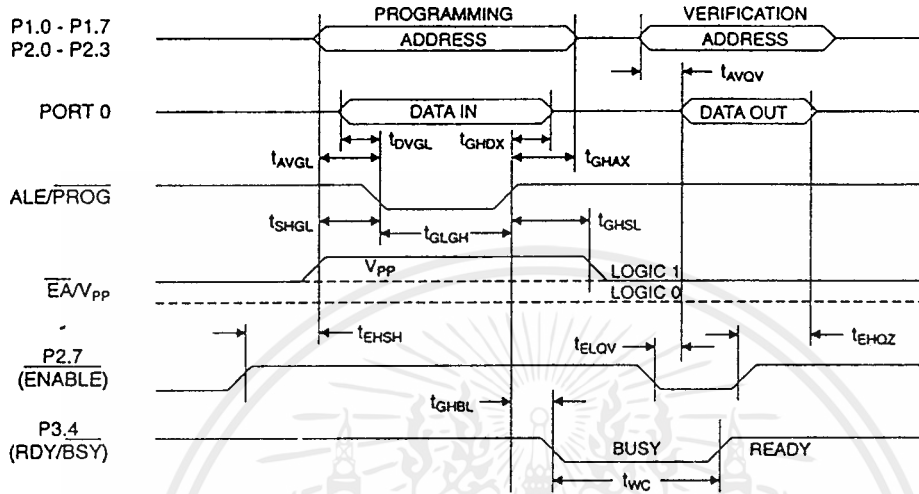
| Symbol | Parameter | Min | Max | Units |
|------------------|---|---------------------|---------------------|---------------|
| $V_{PP}^{(1)}$ | Programming Enable Voltage | 11.5 | 12.5 | V |
| $I_{PP}^{(1)}$ | Programming Enable Current | | 1.0 | mA |
| $1/f_{CLCL}$ | Oscillator Frequency | 4 | 24 | 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_{GDHX} | Data Hold After \overline{PROG} | 48t _{CLCL} | | |
| t_{ESHX} | P2.7 (ENABLE) High to V_{PP} | 48t _{CLCL} | | |
| t_{SHGL} | V_{PP} Setup to \overline{PROG} Low | 10 | | μs |
| $t_{GHSL}^{(1)}$ | V_{PP} Hold After \overline{PROG} | 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_{EHQV} | 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 |

Note: 1. Only used in 12-volt programming mode.

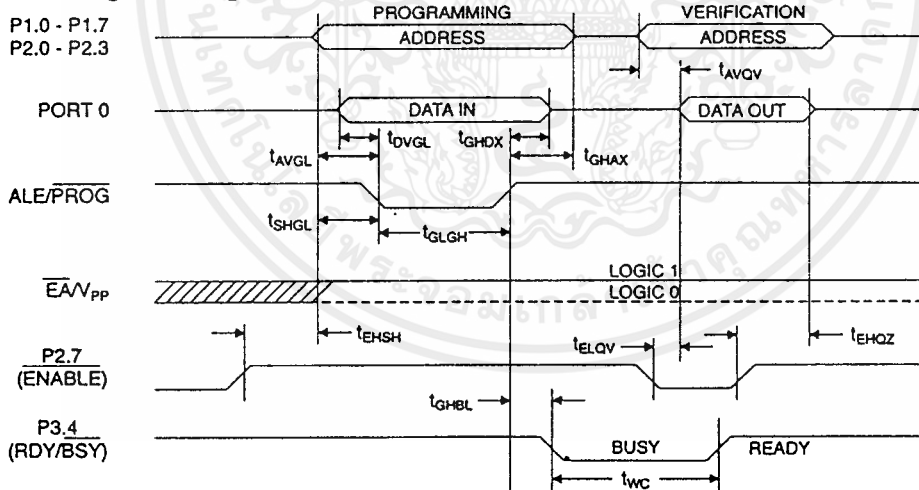




Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



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 | -1.0 V to +7.0 V |
| Maximum Operating Voltage | 6.6 V |
| DC Output Current | 15.0 mA |

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage 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.

D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0 V ± 20% (unless otherwise noted)

| Symbol | Parameter | Condition | Min | Max | Units |
|------------------|--|--|--------------------------|--------------------------|-------|
| V _{IL} | Input Low Voltage | (Except $\bar{E}A$) | -0.5 | 0.2 V _{CC} -0.1 | V |
| V _{IL1} | Input Low Voltage ($\bar{E}A$) | | -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 [±] | | 0.45 | V |
| V _{OL1} | Output Low Voltage ⁽¹⁾ (Port 0, ALE, PSEN) | I _{OL} = 3.2 mA | | 0.45 | V |
| V _{OH} | Output High Voltage (Ports 1,2,3, ALE, PSEN) | I _{OH} = -60 μA; V _{CC} = 5 V ± 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} = 5 V ± 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} = 0.45 V | | -50 | μA |
| I _{TL} | Logical 1 to 0 Transition Current (Ports 1,2,3) | V _{IN} = 2 V | | -650 | μA |
| I _{LI} | Input Leakage Current (Port 0, $\bar{E}A$) | 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 | | 20 | mA |
| | | Idle Mode, 12 MHz | | 5 | mA |
| | Power Down Mode ⁽²⁾ | V _{CC} = 6 V | | 100 | μA |
| | | V _{CC} = 3 V | | 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 2 V.





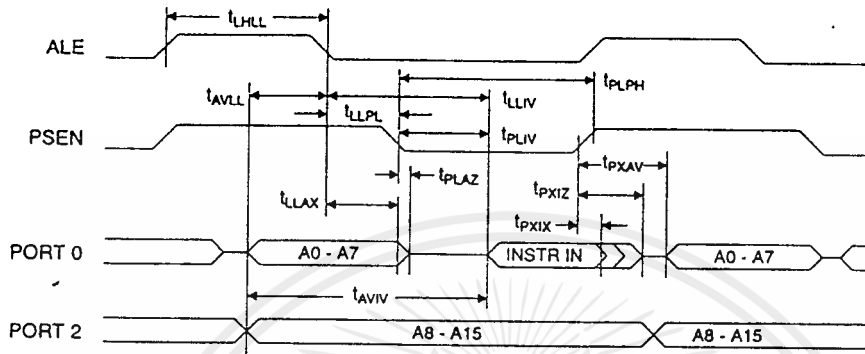
A.C. Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/PROG, and PSEN = 100 pF; Load Capacitance for all other outputs = 80 pF)

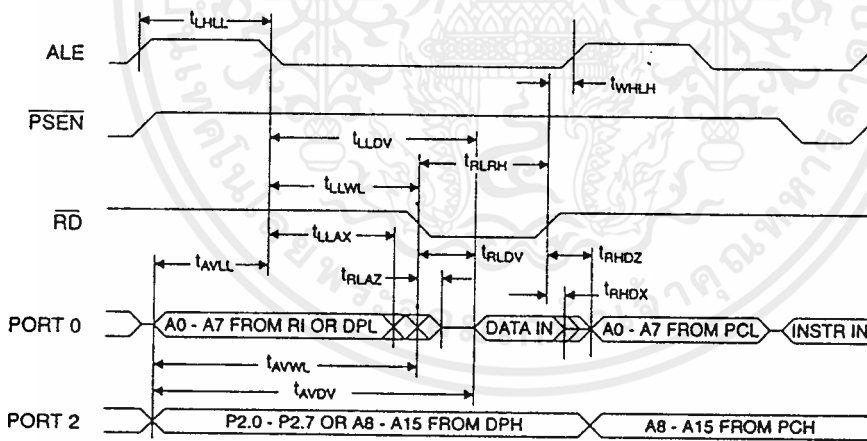
External Program and Data Memory Characteristics

| Symbol | Parameter | 12 MHz Oscillator | | 16 to 24 MHz Oscillator | | Units |
|---------------------|------------------------------------|-------------------|-----|-------------------------|-------------------------|-------|
| | | Min | Max | Min | Max | |
| 1/t _{CLCL} | Oscillator Frequency | | | 0 | 24 | MHz |
| t _{LHLL} | ALE Pulse Width | 127 | | 2t _{CLCL} -40 | | ns |
| t _{AVLL} | Address Valid to ALE Low | 28 | | t _{CLCL} -13 | | ns |
| t _{LLAX} | Address Hold After ALE Low | 48 | | t _{CLCL} -20 | | ns |
| t _{LLIV} | ALE Low to Valid Instruction In | | 233 | | 4t _{CLCL} -65 | ns |
| t _{LLPL} | ALE Low to PSEN Low | 43 | | t _{CLCL} -13 | | ns |
| t _{PLPH} | PSEN Pulse Width | 205 | | 3t _{CLCL} -20 | | ns |
| t _{PLIV} | PSEN Low to Valid Instruction In | | 145 | | 3t _{CLCL} -45 | ns |
| t _{PXIX} | Input Instruction Hold After PSEN | 0 | | 0 | | ns |
| t _{PXIZ} | Input Instruction Float After PSEN | | 59 | | t _{CLCL} -10 | ns |
| t _{PXAV} | PSEN to Address Valid | 75 | | t _{CLCL} -8 | | ns |
| t _{AVIV} | Address to Valid Instruction In | | 312 | | 5t _{CLCL} -55 | ns |
| t _{PLAZ} | PSEN Low to Address Float | | 10 | | 10 | ns |
| t _{RLRH} | RD Pulse Width | 400 | | 6t _{CLCL} -100 | | ns |
| t _{WLWH} | WR Pulse Width | 400 | | 6t _{CLCL} -100 | | ns |
| t _{RLDV} | RD Low to Valid Data In | | 252 | | 5t _{CLCL} -90 | ns |
| t _{RHDZ} | Data Hold After RD | 0 | | 0 | | ns |
| t _{RHDZ} | Data Float After RD | | 97 | | 2t _{CLCL} -28 | ns |
| t _{LDV} | ALE Low to Valid Data In | | 517 | | 8t _{CLCL} -150 | ns |
| t _{AVDV} | Address to Valid Data In | | 585 | | 9t _{CLCL} -165 | ns |
| t _{LLWL} | ALE Low to RD or WR Low | 200 | 300 | 3t _{CLCL} -50 | 3t _{CLCL} +50 | ns |
| t _{AVWL} | Address to RD or WR Low | 203 | | 4t _{CLCL} -75 | | ns |
| t _{QVWX} | Data Valid to WR Transition | 23 | | t _{CLCL} -20 | | ns |
| t _{QVWH} | Data Valid to WR High | 433 | | 7t _{CLCL} -120 | | ns |
| t _{WHQX} | Data Hold After WR | 33 | | t _{CLCL} -20 | | ns |
| t _{RLAZ} | RD Low to Address Float | | 0 | | 0 | ns |
| t _{WLHL} | RD or WR High to ALE High | 43 | 123 | t _{CLCL} -20 | t _{CLCL} +25 | ns |

External Program Memory Read Cycle

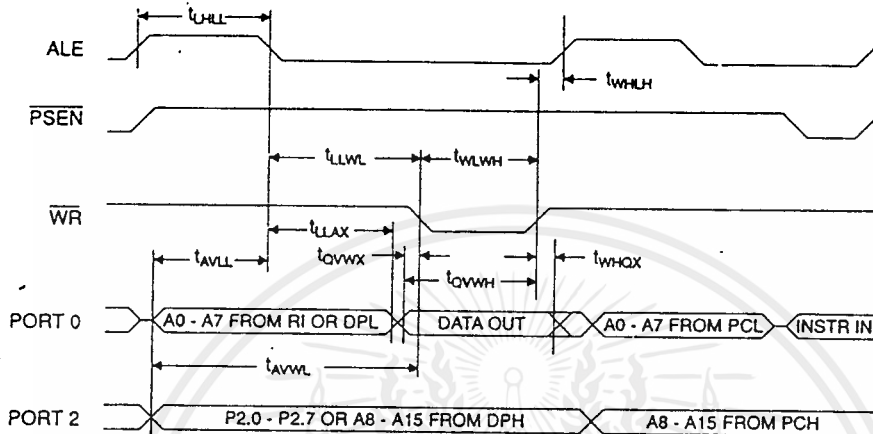


External Data Memory Read Cycle

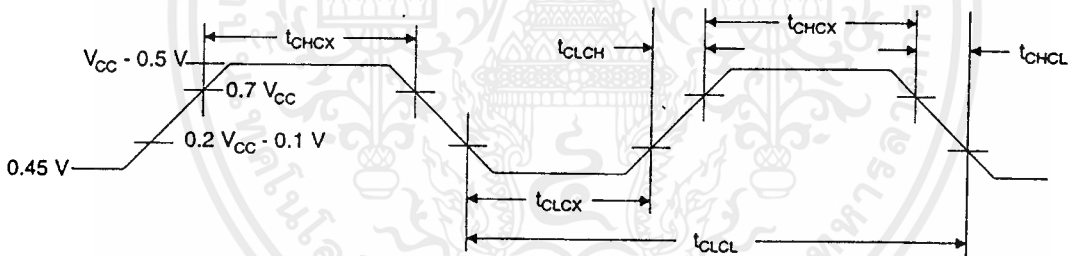




External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

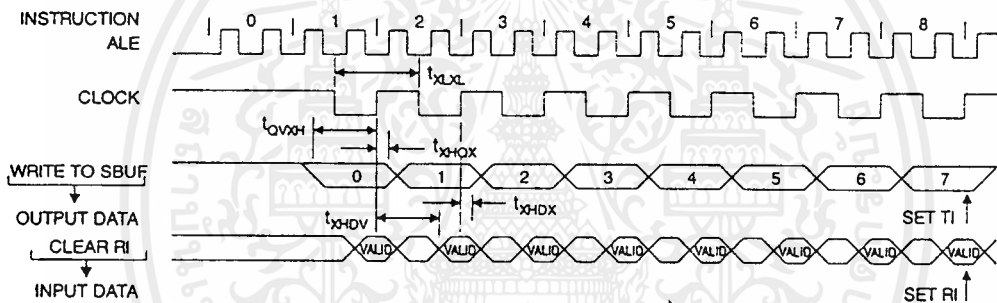
| Symbol | Parameter | Min | Max | Units |
|-----------------|----------------------|------|-----|-------|
| $1/\text{CLCL}$ | Oscillator Frequency | 0 | 24 | MHz |
| t_{CLCL} | Clock Period | 41.6 | | ns |
| t_{CHCX} | High Time | 15 | | ns |
| t_{CLCX} | Low Time | 15 | | ns |
| t_{CLCH} | Rise Time | | 20 | ns |
| t_{CHCL} | Fall Time | | 20 | ns |

Serial Port Timing: Shift Register Mode Test Conditions

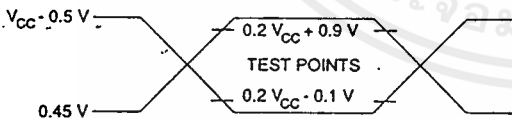
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

| Symbol | Parameter | 12 MHz Osc | | Variable Oscillator | | Units |
|-------------------|--|------------|-----|--------------------------|--------------------------|-------|
| | | Min | Max | Min | Max | |
| txLXL | Serial Port Clock Cycle Time | 1.0 | | 12t _{CLCL} | | μs |
| t _{OVXH} | Output Data Setup to Clock Rising Edge | 700 | | 10t _{CLCL} -133 | | ns |
| t _{XHOX} | Output Data Hold After Clock Rising Edge | 50 | | 2t _{CLCL} -33 | | ns |
| t _{XHDX} | Input Data Hold After Clock Rising Edge | 0 | | 0 | | ns |
| t _{XHDV} | Clock Rising Edge to Input Data Valid | | 700 | | 10t _{CLCL} -133 | ns |

Shift Register Mode Timing Waveforms

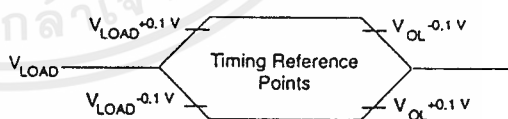


AC Testing Input/Output Waveforms ⁽¹⁾



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms ⁽¹⁾



Note: 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.





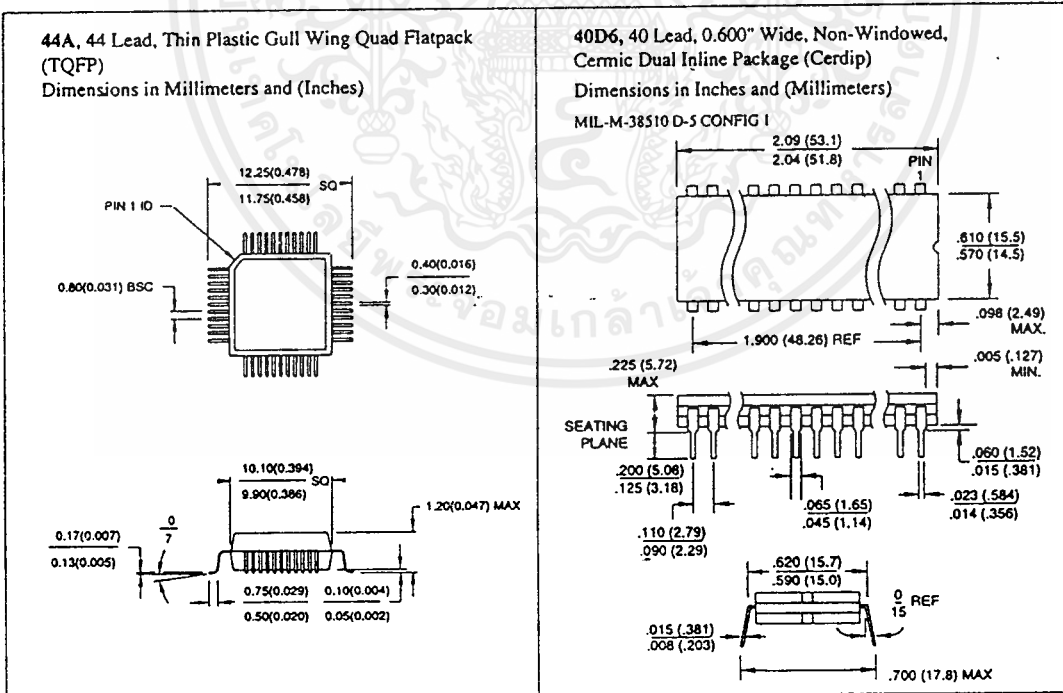
Ordering Information

| Speed (MHz) | Power Supply | Ordering Code | Package | Operation Range | |
|--------------|------------------|---------------|---|--------------------------------|-------------------------------|
| 12 | 5 V \pm 20% | AT89C51-12AC | 44A | Commercial (0°C to 70°C) | |
| | | AT89C51-12JC | 44J | | |
| | | AT89C51-12PC | 40P6 | | |
| | | AT89C51-12QC | 44Q | | |
| | | AT89C51-12AI | 44A | | Industrial (-40°C to 85°C) |
| | | AT89C51-12JI | 44J | | |
| | AT89C51-12PI | 40P6 | | | |
| | AT89C51-12QI | 44Q | | | |
| | 5 V \pm 10% | AT89C51-12AA | 44A | Automotive (-40°C to 125°C) | |
| | | AT89C51-12JA | 44J | | |
| AT89C51-12PA | | 40P6 | | | |
| AT89C51-12QA | | 44Q | | | |
| | AT89C51-12DM | 40D6 | Military (-55°C to 125°C) | | |
| | AT89C51-12LM | 44L | | | |
| | AT89C51-12DM/883 | 40D6 | Military/883C Class B, Fully Compliant (-55°C to 125°C) | | |
| | AT89C51-12LM/883 | 44L | | | |
| 16 | 5 V \pm 20% | AT89C51-16AC | 44A | Commercial (0°C to 70°C) | |
| | | AT89C51-16JC | 44J | | |
| | | AT89C51-16PC | 40P6 | | |
| | | AT89C51-16QC | 44Q | | |
| | | AT89C51-16AI | 44A | | Industrial (-40°C to 85°C) |
| | | AT89C51-16JI | 44J | | |
| | AT89C51-16PI | 40P6 | | | |
| | AT89C51-16QI | 44Q | | | |
| | | AT89C51-16AA | 44A | Automotive (-40°C to 125°C) | |
| | | AT89C51-16JA | 44J | | |
| AT89C51-16PA | | 40P6 | | | |
| AT89C51-16QA | | 44Q | | | |
| 20 | 5 V \pm 20% | AT89C51-20AC | 44A | Commercial (0°C to 70°C) | |
| | | AT89C51-20JC | 44J | | |
| | | AT89C51-20PC | 40P6 | | |
| | | AT89C51-20QC | 44Q | | |
| | | AT89C51-20AI | 44A | | Industrial (-40°C to 85°C) |
| | | AT89C51-20JI | 44J | | |
| AT89C51-20PI | 40P6 | | | | |
| AT89C51-20QI | 44Q | | | | |
| 24 | 5 V \pm 20% | AT89C51-24AC | 44A | Commercial (0°C to 70°C) | |
| | | AT89C51-24JC | 44J | | |
| | | AT89C51-24PC | 44P6 | | |
| | | AT89C51-24QC | 44Q | | |
| | | AT89C51-24AI | 44A | | Industrial (-40°C to 85°C) |
| | | AT89C51-24JI | 44J | | |
| AT89C51-24PI | 44P6 | | | | |
| AT89C51-24QI | 44Q | | | | |

Ordering Information

| Package Type | |
|--------------|--|
| 44A | 44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP) |
| 40D6 | 40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip) |
| 44J | 44 Lead, Plastic J-Leaded Chip Carrier (PLCC) |
| 44L | 44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC) |
| 40P6 | 40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP) |
| 44Q | 44 Lead, Plastic Gull Wing Quad Flatpack (PQFP) |

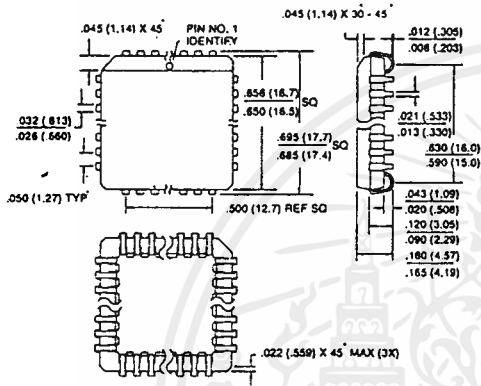
Packaging Information



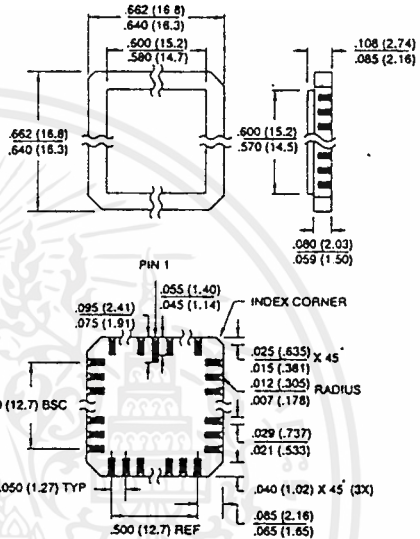


Packaging Information

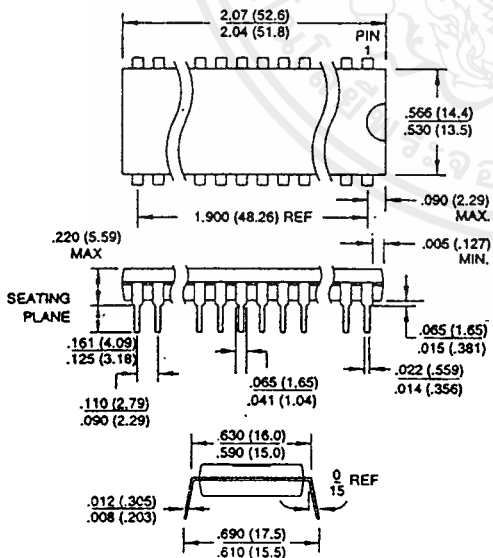
44J, 44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
 Dimensions in Inches and (Millimeters)
 JEDEC OUTLINE MO-047 AC



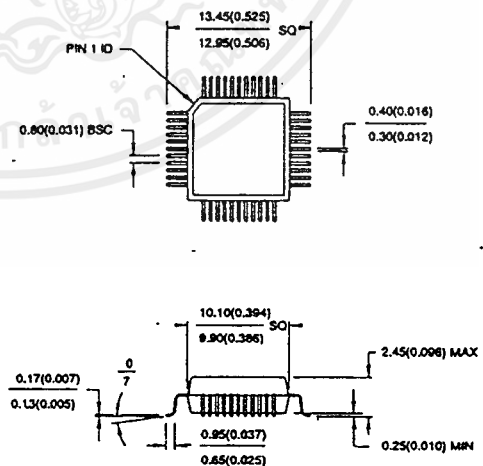
44L, 44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
 Dimensions in Inches and (Millimeters)



40P6, 40 Lead, 0.600" Wide, Plastic Dual In-Line Package (PDIP)
 Dimensions in Inches and (Millimeters)



44Q, 44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)
 Dimensions in Millimeters and (Inches)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM628/LM629 Precision Motion Controller

General Description

The LM628/LM629 are dedicated motion-control processors designed for use with a variety of DC and brushless DC servo motors, and other servomechanisms which provide a quadrature incremental position feedback signal. The parts perform the intensive, real-time computational tasks required for high performance digital motion control. The host control software interface is facilitated by a high-level command set. The LM628 has an 8-bit output which can drive either an 8-bit or a 12-bit DAC. The components required to build a servo system are reduced to the DC motor/actuator, an incremental encoder, a DAC, a power amplifier, and the LM628. An LM629-based system is similar, except that it provides an 8-bit PWM output for directly driving H-switches. The parts are fabricated in NMOS and packaged in a 28-pin dual in-line package or a 24-pin surface mount package (LM629 only). Both 6 MHz and 8 MHz maximum frequency versions are available with the suffixes -6 and -8, respectively, used to designate the versions. They incorporate an SDA core processor and cells designed by SDA.

Features

- 32-bit position, velocity, and acceleration registers
- Programmable digital PID filter with 16-bit coefficients
- Programmable derivative sampling interval
- 8- or 12-bit DAC output data (LM628)
- 8-bit sign-magnitude PWM output data (LM629)
- Internal trapezoidal velocity profile generator
- Velocity, target position, and filter parameters may be changed during motion
- Position and velocity modes of operation
- Real-time programmable host interrupts
- 8-bit parallel asynchronous host interface
- Quadrature incremental encoder interface with index pulse input
- Available in a 28-pin dual in-line package or a 24-pin surface mount package (LM629 only)

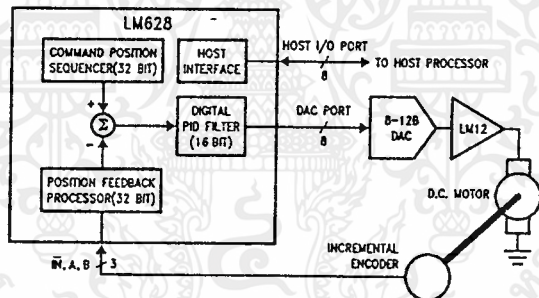
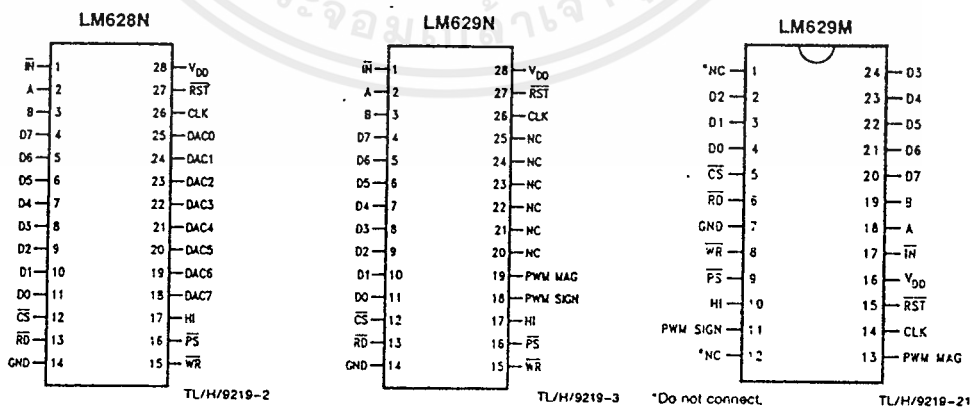


FIGURE 1. Typical System Block Diagram.

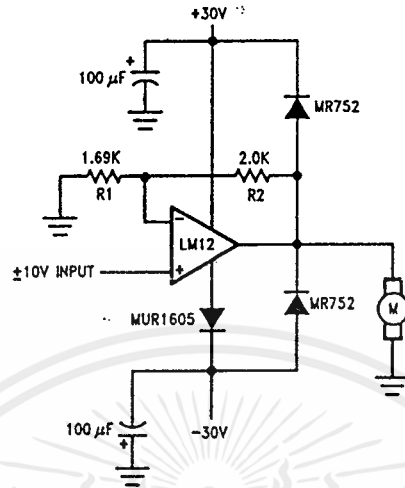
TL/H/9219-1

Connection Diagrams



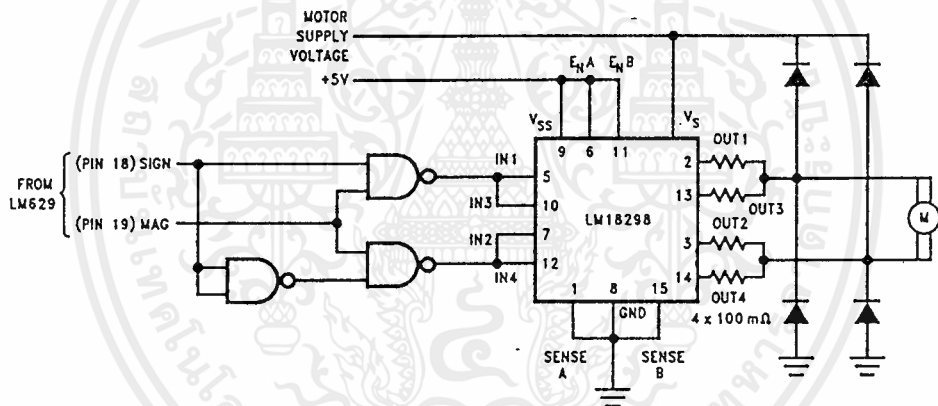
Order Number LM629M-6, LM629M-8, LM628N-6, LM628N-8, LM629N-6 or LM629N-8
See NS Package Number M24B or N28B

Typical Applications (Continued)



TL/H/9219-17

FIGURE 15. Driving a Motor with the LM12 Power Op Amp



TL/H/9219-18

FIGURE 16. PWM Drive for Brush/Commutator Motors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Theory of Operation (Continued)

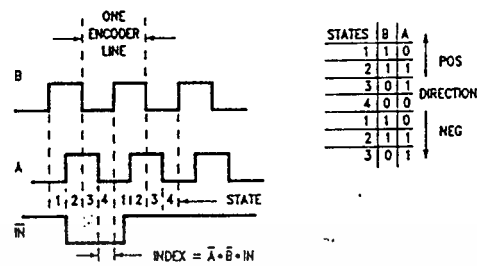


FIGURE 9. Quadrature Encoder Signals

TL/H/9219-11

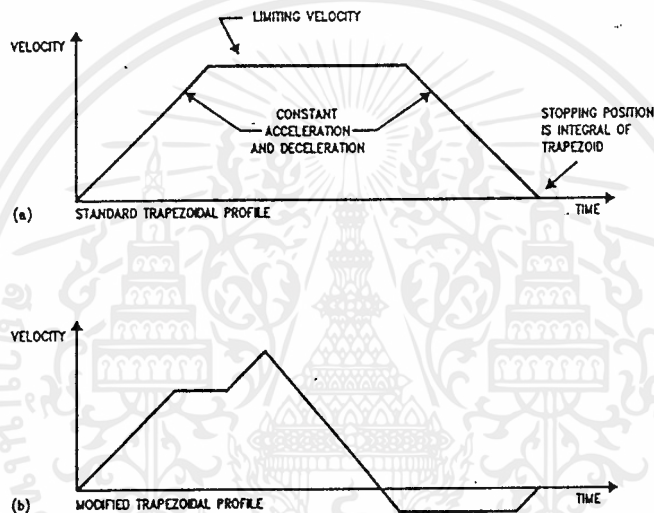


FIGURE 10. Typical Velocity Profiles

TL/H/9219-12

VELOCITY PROFILE (TRAJECTORY) GENERATION

The trapezoidal velocity profile generator computes the desired position of the motor versus time. In the position mode of operation, the host processor specifies acceleration, maximum velocity, and final position. The LM628 uses this information to affect the move by accelerating as specified until the maximum velocity is reached or until deceleration must begin to stop at the specified final position. The deceleration rate is equal to the acceleration rate. At any time during the move the maximum velocity and/or the target position may be changed, and the motor will accelerate or decelerate accordingly. Figure 10 illustrates two typical trapezoidal velocity profiles. Figure 10 (a) shows a simple trapezoid, while Figure 10 (b) is an example of what the trajectory looks like when velocity and position are changed at different times during the move.

When operating in the velocity mode, the motor accelerates to the specified velocity at the specified acceleration rate and maintains the specified velocity until commanded to stop. The velocity is maintained by advancing the desired position at a constant rate. If there are disturbances to the motion during velocity mode operation, the long-time average velocity remains constant. If the motor is unable to maintain the specified velocity (which could be caused by a locked rotor, for example), the desired position will continue to be increased, resulting in a very large position error. If this

condition goes undetected, and the impeding force on the motor is subsequently released, the motor could reach a very high velocity in order to catch up to the desired position (which is still advancing as specified). This condition is easily detected; see commands LPEI and LPES.

All trajectory parameters are 32-bit values. Position is a signed quantity. Acceleration and velocity are specified as 16-bit, positive-only integers having 16-bit fractions. The integer portion of velocity specifies how many counts per sampling interval the motor will traverse. The fractional portion designates an additional fractional count per sampling interval. Although the position resolution of the LM628 is limited to integer counts, the fractional counts provide increased average velocity resolution. Acceleration is treated in the same manner. Each sampling interval the commanded acceleration value is added to the current desired velocity to generate a new desired velocity (unless the command velocity has been reached).

One determines the trajectory parameters for a desired move as follows. If, for example, one has a 500-line shaft encoder, desires that the motor accelerate at one revolution per second per second until it is moving at 600 rpm, and then decelerate to a stop at a position exactly 100 revolutions from the start, one would calculate the trajectory parameters as follows:

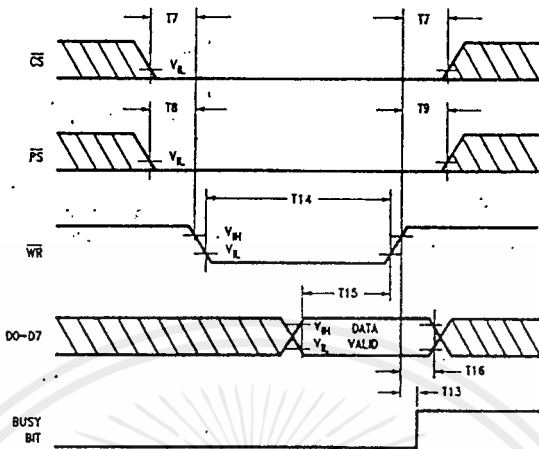


FIGURE 5. Command Byte Write Timing

TL/H/9219-7

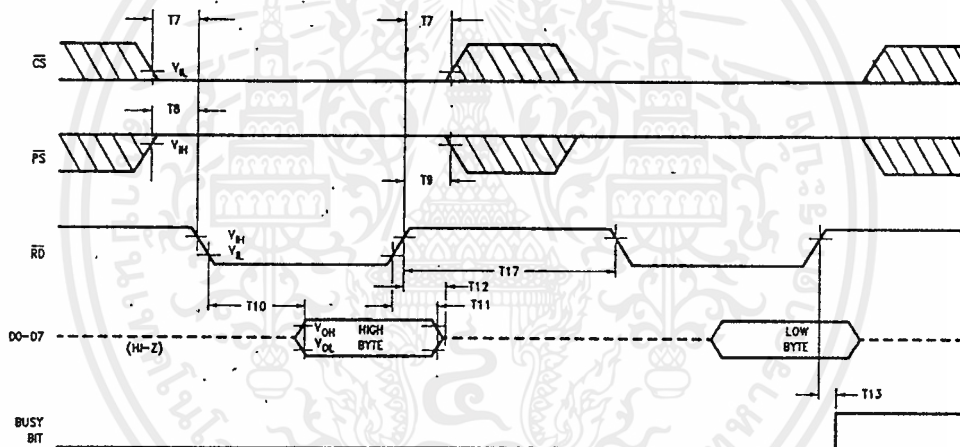


FIGURE 6. Data Word Read Timing

TL/H/9219-8

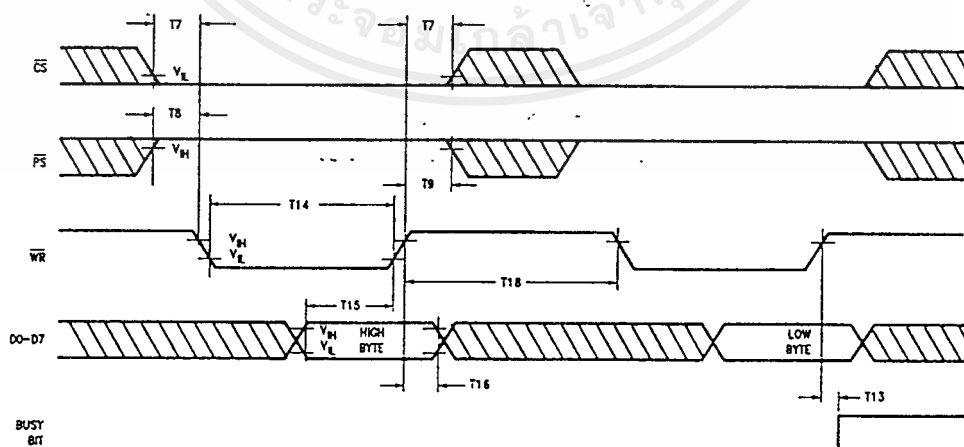


FIGURE 7. Data Word Write Timing

TL/H/9219-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics (Continued)

(V_{DD} and T_A per Operating Ratings; f_{CLK} = 6 MHz; C_{LOAD} = 50 pF; Input Test Signal t_r = t_f = 10 ns)

| Timing Interval | T# | Tested Limits | | Units |
|---|-----|---------------|----------|-------|
| | | Min | Max | |
| STATUS BYTE READ TIMING (See Figure 4) | | | | |
| Chip-Select Setup/Hold Time | T7 | 0 | | ns |
| Port-Select Setup Time | T8 | 30 | | ns |
| Port-Select Hold Time | T9 | 30 | | ns |
| Read Data Access Time | T10 | | 180 | ns |
| Read Data Hold Time | T11 | 0 | | ns |
| \overline{RD} High to Hi-Z Time | T12 | | 180 | ns |
| COMMAND BYTE WRITE TIMING (See Figure 5) | | | | |
| Chip-Select Setup/Hold Time | T7 | 0 | | ns |
| Port-Select Setup Time | T8 | 30 | | ns |
| Port-Select Hold Time | T9 | 30 | | ns |
| Busy Bit Delay | T13 | | (Note 3) | ns |
| \overline{WR} Pulse Width | T14 | 100 | | ns |
| Write Data Setup Time | T15 | 50 | | ns |
| Write Data Hold Time | T16 | 120 | | ns |
| DATA WORD READ TIMING (See Figure 6) | | | | |
| Chip-Select Setup/Hold Time | T7 | 0 | | ns |
| Port-Select Setup Time | T8 | 30 | | ns |
| Port-Select Hold Time | T9 | 30 | | ns |
| Read Data Access Time | T10 | | 180 | ns |
| Read Data Hold Time | T11 | 0 | | ns |
| \overline{RD} High to Hi-Z Time | T12 | | 180 | ns |
| Busy Bit Delay | T13 | | (Note 3) | ns |
| Read Recovery Time | T17 | 120 | | ns |
| DATA WORD WRITE TIMING (See Figure 7) | | | | |
| Chip-Select Setup/Hold Time | T7 | 0 | | ns |
| Port-Select Setup Time | T8 | 30 | | ns |
| Port-Select Hold Time | T9 | 30 | | ns |
| Busy Bit Delay | T13 | | (Note 3) | ns |
| \overline{WR} Pulse Width | T14 | 100 | | ns |
| Write Data Setup Time | T15 | 50 | | ns |
| Write Data Hold Time | T16 | 120 | | ns |
| Write Recovery Time | T18 | 120 | | ns |

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond the above Operating Ratings.

Note 2: When operating at ambient temperatures above 70°C, the device must be protected against excessive junction temperatures. Mounting the package on a printed circuit board having an area greater than three square inches and surrounding the leads and body with wide copper traces and large, uninterrupted areas of copper, such as a ground plane, suffices. The 28-pin DIP (N) and the 24-pin surface mount package (M) are molded plastic packages with solid copper lead frames. Most of the heat generated at the die flows from the die, through the copper lead frame, and into copper traces on the printed circuit board. The copper traces act as a heat sink. Double-sided or multi-layer boards provide heat transfer characteristics superior to those of single-sided boards.

Note 3: In order to read the busy bit, the status byte must first be read. The time required to read the busy bit far exceeds the time the chip requires to set the busy bit. It is, therefore, impossible to test actual busy bit delay. The busy bit is guaranteed to be valid as soon as the user is able to read it.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|--|-----------------|
| Voltage at Any Pin with Respect to GND | -0.3V to +7.0V |
| Ambient Storage Temperature | -65°C to +150°C |
| Lead Temperature | |
| 28-pin Dual In-Line Package (Soldering, 4 sec.) | 260°C |
| 24-pin Surface Mount Package (Soldering, 10 sec.) | 300°C |
| Maximum Power Dissipation ($T_A \leq 85^\circ\text{C}$, Note 2) | 605 mW |
| ESD Tolerance ($C_{ZAP} = 120 \text{ pF}$, $R_{ZAP} = 1.5\text{k}$) | 2000V |

Operating Ratings

| | |
|------------------------------|--|
| Temperature Range | $-40^\circ\text{C} < T_A < +85^\circ\text{C}$ |
| Clock Frequency: | |
| LM628N-6, LM629N-6, LM629M-6 | $1.0 \text{ MHz} < f_{\text{CLK}} < 6.0 \text{ MHz}$ |
| LM628N-8, LM629N-8, LM629M-8 | $1.0 \text{ MHz} < f_{\text{CLK}} < 8.0 \text{ MHz}$ |
| V_{DD} Range | $4.5\text{V} < V_{\text{DD}} < 5.5\text{V}$ |

DC Electrical Characteristics (V_{DD} and T_A per Operating Ratings; $f_{\text{CLK}} = 6 \text{ MHz}$)

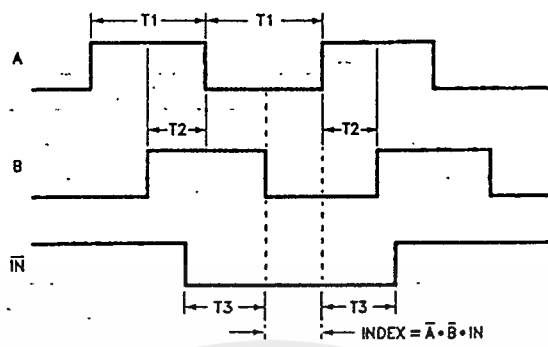
| Symbol | Parameter | Conditions | Tested Limits | | Units |
|------------------------|-----------------------------------|--|---------------|-----|---------------|
| | | | Min | Max | |
| I_{DD} | Supply Current | Outputs Open | | 110 | mA |
| INPUT VOLTAGES | | | | | |
| V_{IH} | Logic 1 Input Voltage | | 2.0 | | V |
| V_{IL} | Logic 0 Input Voltage | | | 0.8 | V |
| I_{IN} | Input Currents | $0 \leq V_{\text{IN}} \leq V_{\text{DD}}$ | -10 | 10 | μA |
| OUTPUT VOLTAGES | | | | | |
| V_{OH} | Logic 1 | $I_{\text{OH}} = -1.6 \text{ mA}$ | 2.4 | | V |
| V_{OL} | Logic 0 | $I_{\text{OL}} = 1.6 \text{ mA}$ | | 0.4 | V |
| I_{OUT} | TRI-STATE* Output Leakage Current | $0 \leq V_{\text{OUT}} \leq V_{\text{DD}}$ | -10 | 10 | μA |

AC Electrical Characteristics

(V_{DD} and T_A per Operating Ratings; $f_{\text{CLK}} = 6 \text{ MHz}$; $C_{\text{LOAD}} = 50 \text{ pF}$; Input Test Signal $t_r = t_f = 10 \text{ ns}$)

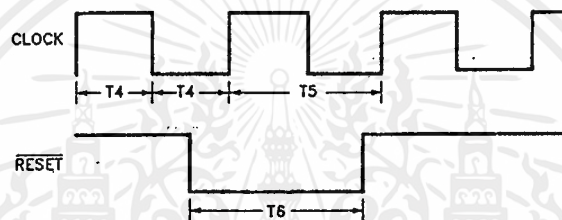
| Timing Interval | T# | Tested Limits | | Units |
|--|------------------------------|-----------------------------|-----|---------------|
| | | Min | Max | |
| ENCODER AND INDEX TIMING (See Figure 2) | | | | |
| Motor-Phase Pulse Width | T1 | $\frac{16}{f_{\text{CLK}}}$ | | μs |
| Dwell-Time per State | T2 | $\frac{8}{f_{\text{CLK}}}$ | | μs |
| Index Pulse Setup and Hold (Relative to A and B Low) | T3 | 0 | | μs |
| CLOCK AND RESET TIMING (See Figure 3) | | | | |
| Clock Pulse Width | LM628N-6, LM629N-6, LM629M-6 | T4 | 78 | ns |
| | LM628N-8, LM629N-8, LM629M-8 | T4 | 57 | ns |
| Clock Period | LM628N-6, LM629N-6, LM629M-6 | T5 | 166 | ns |
| | LM628N-8, LM629N-8, LM629M-8 | T5 | 125 | ns |
| Reset Pulse Width | T6 | $\frac{8}{f_{\text{CLK}}}$ | | μs |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



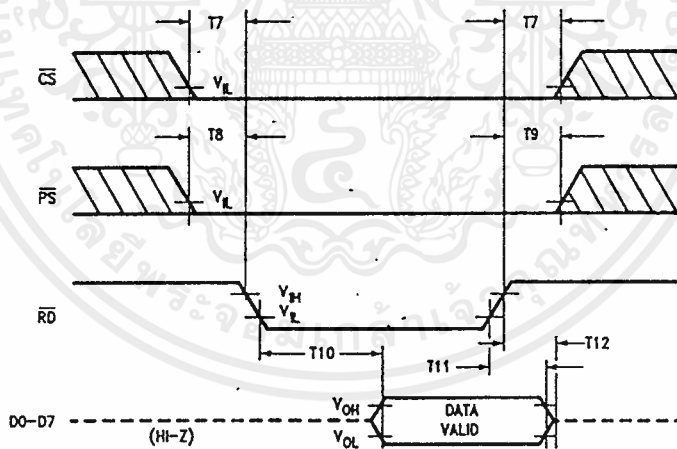
TL/H/9219-4

FIGURE 2. Quadrature Encoder Input Timing



TL/H/9219-5

FIGURE 3. Clock and Reset Timing



TL/H/9219-6

FIGURE 4. Status Byte Read Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pinout Description

(See Connection Diagrams) Pin numbers for the 24-pin surface mount package are indicated in parentheses.

Pin 1 (17), Index (\overline{IN}) Input: Receives optional index pulse from the encoder. Must be tied high if not used. The index position is read when Pins 1, 2, and 3 are low.

Pins 2 and 3 (18 and 19), Encoder Signal (A, B) Inputs: Receive the two-phase quadrature signals provided by the incremental encoder. When the motor is rotating in the positive ("forward") direction, the signal at Pin 2 leads the signal at Pin 3 by 90 degrees. Note that the signals at Pins 2 and 3 must remain at each encoder state (See Figure 9) for a minimum of 8 clock periods in order to be recognized. Because of a four-to-one resolution advantage gained by the method of decoding the quadrature encoder signals, this corresponds to a maximum encoder-state capture rate of 1.0 MHz ($f_{CLK} = 8.0$ MHz) or 750 kHz ($f_{CLK} = 6.0$ MHz). For other clock frequencies the encoder signals must also remain at each state a minimum of 8 clock periods.

Pins 4 to 11 (20 to 24 and 2 to 4), Host I/O Port (D0 to D7): Bi-directional data port which connects to host computer/processor. Used for writing commands and data to the LM628, and for reading the status byte and data from the LM628, as controlled by \overline{CS} (Pin 12), \overline{PS} (Pin 16), \overline{RD} (Pin 13), and \overline{WR} (Pin 15).

Pin 12 (5), Chip Select (\overline{CS}) Input: Used to select the LM628, and for writing and reading operations.

Pin 13 (6), Read (\overline{RD}) Input: Used to read status and data.

Pin 14 (7), Ground (GND): Power-supply return pin.

Pin 15 (8), Write (\overline{WR}) Input: Used to write commands and data.

Pin 16 (9), Port Select (\overline{PS}) Input: Used to select command or data port. Selects command port when low, data port when high. The following modes are controlled by Pin 16:

1. Commands are written to the command port (Pin 16 low),
2. Status byte is read from command port (Pin 16 low), and
3. Data is written and read via the data port (Pin 16 high).

Pin 17 (10), Host Interrupt (HI) Output: This active-high signal alerts the host (via a host interrupt service routine) that an interrupt condition has occurred.

Pins 18 to 25, DAC Port (DAC0 to DAC7): Output port which is used in three different modes:

1. LM628 (8-bit output mode): Outputs latched data to the DAC. The MSB is Pin 18 and the LSB is Pin 25.

2. LM628 (12-bit output mode): Outputs two, multiplexed 6-bit words. The less-significant word is output first. The MSB is on Pin 18 and the LSB is on Pin 23. Pin 24 is used to demultiplex the words; Pin 24 is low for the less-significant word. The positive-going edge of the signal on Pin 25 is used to strobe the output data. Figure 8 shows the timing of the multiplexed signals.

3. LM629 (sign/magnitude outputs): Outputs a PWM sign signal on Pin 18 (11 for surface mount), and a PWM magnitude signal on Pin 19 (13 for surface mount). Pins 20 to 25 are not used in the LM629. Figure 11 shows the PWM output signal format.

Pin 26 (14), Clock (CLK) Input: Receives system clock.

Pin 27 (15), Reset (\overline{RST}) Input: Active-low, positive-edge triggered, resets the LM628 to the internal conditions shown below. Note that the reset pulse must be logic low for a minimum of 8 clock periods. Reset does the following:

1. Filter coefficient and trajectory parameters are zeroed.
2. Sets position error threshold to maximum value (7FFF hex), and effectively executes command LPEI.
3. The SBPA/SBPR interrupt is masked (disabled).
4. The five other interrupts are unmasked (enabled).
5. Initializes current position to zero, or "home" position.
6. Sets derivative sampling interval to $2048/f_{CLK}$ or 256 μ s for an 8.0 MHz clock.
7. DAC port outputs 800 hex to "zero" a 12-bit DAC and then reverts to 80 hex to "zero" an 8-bit DAC.

Immediately after releasing the reset pin from the LM628, the status port should read '00'. If the reset is successfully completed, the status word will change to hex '84' or 'C4' within 1.5 ms. If the status word has not changed from hex '00' to '84' or 'C4' within 1.5 ms, perform another reset and repeat the above steps. To be certain that the reset was properly performed, execute a RSTI command. If the chip has reset properly, the status byte will change from hex '84' or 'C4' to hex '80' or 'C0'. If this does not occur, perform another reset and repeat the above steps.

Pin 28 (16), Supply Voltage (V_{DD}): Power supply voltage (+5V).

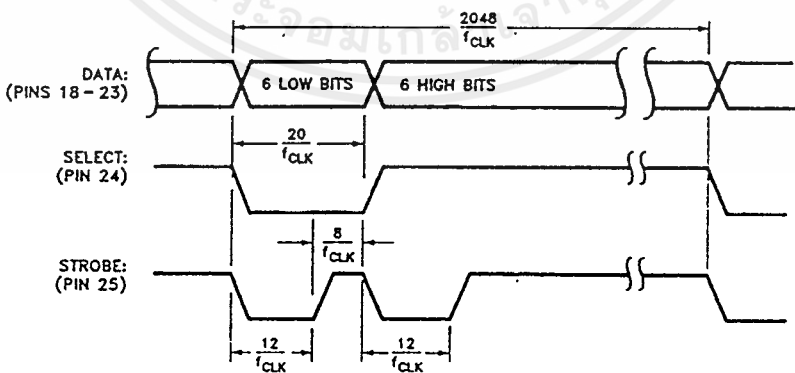


FIGURE 8. 12-Bit Multiplexed Output Timing

TJ/H/9219-10

Theory of Operation

INTRODUCTION

The typical system block diagram (See *Figure 1*) illustrates a servo system built using the LM628. The host processor communicates with the LM628 through an I/O port to facilitate programming a trapezoidal velocity profile and a digital compensation filter. The DAC output interfaces to an external digital-to-analog converter to produce the signal that is power amplified and applied to the motor. An incremental encoder provides feedback for closing the position servo loop. The trapezoidal velocity profile generator calculates the required trajectory for either position or velocity mode of operation. In operation, the LM628 subtracts the actual position (feedback position) from the desired position (profile generator position), and the resulting position error is processed by the digital filter to drive the motor to the desired position. Table I provides a brief summary of specifications offered by the LM628/LM629:

POSITION FEEDBACK INTERFACE

The LM628 interfaces to a motor via an incremental encoder. Three inputs are provided: two quadrature signal inputs,

and an index pulse input. The quadrature signals are used to keep track of the absolute position of the motor. Each time a logic transition occurs at one of the quadrature inputs, the LM628 internal position register is incremented or decremented accordingly. This provides four times the resolution over the number of lines provided by the encoder. See *Figure 9*. Each of the encoder signal inputs is synchronized with the LM628 clock.

The optional index pulse output provided by some encoders assumes the logic-low state once per revolution. If the LM628 is so programmed by the user, it will record the absolute motor position in a dedicated register (the index register) at the time when all three encoder inputs are logic low.

If the encoder does not provide an index output, the LM628 index input can also be used to record the home position of the motor. In this case, typically, the motor will close a switch which is arranged to cause a logic-low level at the index input, and the LM628 will record motor position in the index register and alert (interrupt) the host processor. Permanently grounding the index input will cause the LM628 to malfunction.

TABLE I. System Specifications Summary

| | |
|--------------------|--|
| Position Range | -1,073,741,824 to 1,073,741,823 counts |
| Velocity Range | 0 to 1,073,741,823/2 ¹⁶ counts/sample; ie, 0 to 16,383 counts/sample, with a resolution of 1/2 ¹⁶ counts/sample |
| Acceleration Range | 0 to 1,073,741,823/2 ¹⁶ counts/sample/sample; ie, 0 to 16,383 counts/sample/sample, with a resolution of 1/2 ¹⁶ counts/sample/sample |
| Motor Drive Output | LM628: 8-bit parallel output to DAC, or 12-bit multiplexed output to DAC LM629: 8-bit PWM sign/magnitude signals |
| Operating Modes | Position and Velocity |
| Feedback Device | Incremental Encoder (quadrature signals; support for index pulse) |
| Control Algorithm | Proportional Integral Derivative (PID) (plus programmable integration limit) |
| Sample Intervals | Derivative Term: Programmable from 2048/t _{CLK} to (2048 * 256)/t _{CLK} in steps of 2048/t _{CLK} (256 to 65,536 μs for an 8.0 MHz clock). Proportional and Integral: 2048/t _{CLK} |

พระจอมเกล้าเจ้าคุณ

Theory of Operation (Continued)

let $P = \text{target position (units = encoder counts)}$
 let $R = \text{encoder lines} \cdot 4 \text{ (system resolution)}$
 then $R = 500 \cdot 4 = 2000$
 and $P = 2000 \cdot \text{desired number of revolutions}$
 $P = 2000 \cdot 100 \text{ revs} = 200,000 \text{ counts (value to load)}$
 $P \text{ (coding)} = 00030D40 \text{ (hex code written to LM628)}$

let $V = \text{velocity (units = counts/sample)}$
 let $T = \text{sample time (seconds)} = 341 \mu\text{s (with 6 MHz clock)}$

let $C = \text{conversion factor} = 1 \text{ minute}/60 \text{ seconds}$
 then $V = R \cdot T \cdot C \cdot \text{desired rpm}$
 and $V = 2000 \cdot 341E-6 \cdot 1/60 \cdot 600 \text{ rpm}$
 $V = 6.82 \text{ counts/sample}$
 $V \text{ (scaled)} = 6.82 \cdot 65,536 = 446,955.52$
 $V \text{ (rounded)} = 446,956 \text{ (value to load)}$
 $V \text{ (coding)} = 0006D1EC \text{ (hex code written to LM628)}$

let $A = \text{acceleration (units = counts/sample/sample)}$
 $A = R \cdot T \cdot T \cdot \text{desired acceleration (rev/sec/sec)}$
 then $A = 2000 \cdot 341E-6 \cdot 341E-6 \cdot 1 \text{ rev/sec/sec}$
 and $A = 2.33E-4 \text{ counts/sample/sample}$
 $A \text{ (scaled)} = 2.33E-4 \cdot 65,536 = 15.24$
 $A \text{ (rounded)} = 15 \text{ (value to load)}$
 $A \text{ (coding)} = 0000000F \text{ (hex code written to LM628)}$

The above position, velocity, and acceleration values must be converted to binary codes to be loaded into the LM628. The values shown for velocity and acceleration must be multiplied by 65,536 (as shown) to adjust for the required integer/fraction format of the input data. Note that after scaling the velocity and acceleration values, literal fractional data cannot be loaded; the data must be rounded and converted to binary. The factor of four increase in system resolution is due to the method used to decode the quadrature encoder signals, see *Figure 9*.

PID COMPENSATION FILTER

The LM628 uses a digital Proportional Integral Derivative (PID) filter to compensate the control loop. The motor is held at the desired position by applying a restoring force to the motor that is proportional to the position error, plus the integral of the error, plus the derivative of the error. The following discrete-time equation illustrates the control performed by the LM628:

$$u(n) = k_p \cdot e(n) + k_i \sum_{N=0}^n e(n) + k_d [e(n') - e(n' - 1)] \quad (\text{Eq.1})$$

where $u(n)$ is the motor control signal output at sample time n , $e(n)$ is the position error at sample time n , n' indicates sampling at the derivative sampling rate, and k_p , k_i , and k_d are the discrete-time filter parameters loaded by the users.

The first term, the proportional term, provides a restoring force proportional to the position error, just as does a spring obeying Hooke's law. The second term, the integration term, provides a restoring force that grows with time, and thus ensures that the static position error is zero. If there is

a constant torque loading, the motor will still be able to achieve zero position error.

The third term, the derivative term, provides a force proportional to the rate of change of position error. It acts just like viscous damping in a damped spring and mass system (like a shock absorber in an automobile). The sampling interval associated with the derivative term is user-selectable; this capability enables the LM628 to control a wider range of inertial loads (system mechanical time constants) by providing a better approximation of the continuous derivative. In general, longer sampling intervals are useful for low-velocity operations.

In operation, the filter algorithm receives a 16-bit error signal from the loop summing-junction. The error signal is saturated at 16 bits to ensure predictable behavior. In addition to being multiplied by filter coefficient k_p , the error signal is added to an accumulation of previous errors (to form the integral signal) and, at a rate determined by the chosen *derivative* sampling interval, the previous error is subtracted from it (to form the derivative signal). All filter multiplications are 16-bit operations; only the bottom 16 bits of the product are used.

The integral signal is maintained to 24 bits, but only the top 16 bits are used. This scaling technique results in a more usable (less sensitive) range of coefficient k_i values. The 16 bits are right-shifted eight positions and multiplied by filter coefficient k_i to form the term which contributes to the motor control output. The absolute magnitude of this product is compared to coefficient k_i , and the lesser, appropriately signed magnitude then contributes to the motor control signal.

The derivative signal is multiplied by coefficient k_d each *derivative* sampling interval. This product contributes to the motor control output *every* sample interval, independent of the user-chosen *derivative* sampling interval.

The k_p , limited k_i , and k_d product terms are summed to form a 16-bit quantity. Depending on the output mode (wordsize), either the top 8 or top 12 bits become the motor control output signal.

LM628 READING AND WRITING OPERATIONS

The host processor writes commands to the LM628 via the host I/O port when Port Select (\overline{PS}) input (Pin 16) is logic low. The desired command code is applied to the parallel port line and the Write (\overline{WR}) input (Pin 15) is strobed. The command byte is latched into the LM628 on the rising edge of the \overline{WR} input. When writing command bytes it is necessary to first read the status byte and check the state of a flag called the "busy bit" (Bit 0). If the busy bit is logic high, no command write may take place. The busy bit is never high longer than 100 μs , and typically falls within 15 μs to 25 μs .

The host processor reads the LM628 status byte in a similar manner: by strobing the Read (\overline{RD}) input (Pin 13) when \overline{PS} (Pin 16) is low; status information remains valid as long as \overline{RD} is low.

Writing and reading data to/from the LM628 (as opposed to writing commands and reading status) are done with \overline{PS} (Pin 16) logic high. These writes and reads are always an integral number (from one to seven) of two-byte words, with the first byte of each word being the more significant. Each byte requires a write (\overline{WR}) or read (\overline{RD}) strobe. When transferring data words (byte-pairs), it is necessary to first read the status byte and check the state of the busy bit. When the

Theory of Operation (Continued)

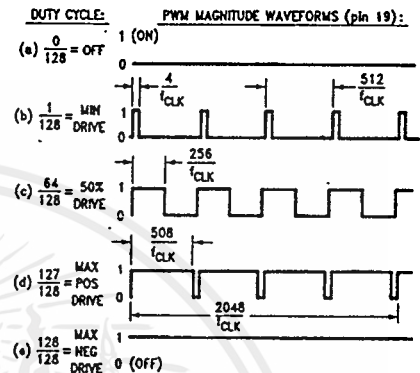
busy bit is logic low, the user may then sequentially transfer both bytes comprising a data word, but the busy bit must again be checked and found to be low before attempting to transfer the next byte pair (when transferring multiple words). Data transfers are accomplished via LM628-internal interrupts (which are not nested); the busy bit informs the host processor when the LM628 may not be interrupted for data transfer (or a command byte). If a command is written when the busy bit is high, the command will be ignored.

The busy bit goes high immediately after writing a command byte, or reading or writing a second byte of data (See Figures 5 thru 7).

MOTOR OUTPUTS

The LM628 DAC output port can be configured to provide either a latched eight-bit parallel output or a multiplexed 12-bit output. The 8-bit output can be directly connected to a flow-through (non-input-latching) D/A converter; the 12-bit output can be easily demultiplexed using an external 6-bit latch and an input-latching 12-bit D/A converter. The DAC output data is offset-binary coded; the 8-bit code for zero is 80 hex and the 12-bit code for zero is 800 hex. Values less than these cause a negative torque to be applied to the motor and, conversely, larger values cause positive motor torque. The LM628, when configured for 12-bit output, provides signals which control the demultiplexing process. See Figure 8 for details.

The LM629 provides 8-bit, sign and magnitude PWM output signals for directly driving switch-mode motor-drive amplifiers. Figure 11 shows the format of the PWM magnitude output signal.



TL/H/9219-13

Note: Sign output (pin 18) not shown

FIGURE 11. PWM Output Signal Format

TABLE II. LM628 User Command Set

| Command | Type | Description | Hex | Data Bytes | Note |
|---------|------------|--------------------------|------|------------|------|
| RESET | Initialize | Reset LM628 | 00 | 0 | 1 |
| PORT8 | Initialize | Select 8-Bit Output | 05 | 0 | 2 |
| PORT12 | Initialize | Select 12-Bit Output | 06 | 0 | 2 |
| DFH | Initialize | Define Home | 02 | 0 | 1 |
| SIP | Interrupt | Set Index Position | 03 | 0 | 1 |
| LPEI | Interrupt | Interrupt on Error | 1B | 2 | 1 |
| LPES | Interrupt | Stop on Error | 1A | 2 | 1 |
| SBPA | Interrupt | Set Breakpoint, Absolute | 20 | 4 | 1 |
| SBPR | Interrupt | Set Breakpoint, Relative | 21 | 4 | 1 |
| MSKI | Interrupt | Mask Interrupts | 1C | 2 | 1 |
| RSTI | Interrupt | Reset Interrupts | 1D | 2 | 1 |
| LFIL | Filter | Load Filter Parameters | 1E | 2 to 10 | 1 |
| UDF | Filter | Update Filter | 04 | 0 | 1 |
| LTRJ | Trajectory | Load Trajectory | 1F | 2 to 14 | 1 |
| STT | Trajectory | Start Motion | 01 | 0 | 3 |
| RDSTAT | Report | Read Status Byte | None | 1 | 1, 4 |
| RDSIGS | Report | Read Signals Register | 0C | 2 | 1 |
| RDIP | Report | Read Index Position | 09 | 4 | 1 |
| RDDP | Report | Read Desired Position | 08 | 4 | 1 |
| RDRP | Report | Read Real Position | 0A | 4 | 1 |
| RDDV | Report | Read Desired Velocity | 07 | 4 | 1 |
| RDRV | Report | Read Real Velocity | 0B | 2 | 1 |
| RDSUM | Report | Read Integration Sum | 0D | 2 | 1 |

Note 1: Commands may be executed "On the Fly" during motion.

Note 2: Commands not applicable to execution during motion.

Note 3: Command may be executed during motion if acceleration parameter was not changed.

Note 4: Command needs no code because the command port status-byte read is totally supported by hardware.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

User Command Set

GENERAL

The following paragraphs describe the user command set of the LM628. Some of the commands can be issued alone and some require a supporting data structure. As examples, the command STT (STarT motion) does not require additional data; command LFIL (Load FILter parameters) requires additional data (derivative-term sampling interval and/or filter parameters).

Commands are categorized by function: initialization, interrupt control, filter control, trajectory control, and data reporting. The commands are listed in Table II and described in the following paragraphs. Along with each command name is its command-byte code, the number of accompanying data bytes that are to be written (or read), and a comment as to whether the command is executable during motion.

Initialization Commands

The following four LM628 user commands are used primarily to initialize the system for use.

RESET COMMAND: RESET the LM628

Command Code: 00 Hex
Data Bytes: None
Executable During Motion: Yes

This command (and the hardware reset input, Pin 27) results in setting the following data items to zero: filter coefficients and their input buffers, trajectory parameters and their input buffers, and the motor control output. A zero motor control output is a half-scale, offset-binary code: (80 hex for the 8-bit output mode; 800 hex for 12-bit mode). During reset, the DAC port outputs 800 hex to "zero" a 12-bit DAC and reverts to 80 hex to "zero" an 8-bit DAC. The command also clears five of the six interrupt masks (only the SBPA/SBPR interrupt is masked), sets the output port size to 8 bits, and defines the current absolute position as home. Reset, which may be executed at any time, will be completed in less than 1.5 ms. Also see commands PORT8 and PORT12.

PORT8 COMMAND: Set Output PORT Size to 8 Bits

Command Code: 05 Hex
Data Bytes: None
Executable During Motion: Not Applicable

The default output port size of the LM628 is 8 bits; so the PORT8 command need not be executed when using an 8-bit DAC. This command must not be executed when using a 12-bit converter; it will result in erratic, unpredictable motor behavior. The 8-bit output port size is the required selection when using the LM629, the PWM-output version of the LM628.

PORT12 COMMAND: Set Output PORT Size to 12 Bits

Command Code: 06 Hex
Data Bytes: None
Executable During Motion: Not Applicable

When a 12-bit DAC is used, command PORT12 should be issued very early in the initialization process. Because use of this command is determined by system hardware, there is only one foreseen reason to execute it later: if the RESET command is issued (because an 8-bit output would then be selected as the default) command PORT12 should be im-

mediately executed. This command must not be issued when using an 8-bit converter or the LM629, the PWM-output version of the LM628.

DFH COMMAND: DeFINE Home

Command Code: 02 Hex
Data Bytes: None
Executable During Motion: Yes

This command declares the current position as "home", or absolute position 0 (Zero). If DFH is executed during motion it will not affect the stopping position of the on-going move unless command STT is also executed.

Interrupt Control Commands

The following seven LM628 user commands are associated with conditions which can be used to interrupt the host computer. In order for any of the potential interrupt conditions to actually interrupt the host via Pin 17, the corresponding bit in the interrupt mask data associated with command MSKI must have been set to logic high (the non-masked state).

The identity of all interrupts is made known to the host via reading and parsing the status byte. Even if all interrupts are masked off via command MSKI, the state of each condition is still reflected in the status byte. This feature facilitates polling the LM628 for status information, as opposed to interrupt driven operation.

SIP COMMAND: Set Index Position

Command Code: 03 Hex
Data Bytes: None
Executable During Motion: Yes

After this command is executed, the absolute position which corresponds to the occurrence of the next index pulse input will be recorded in the index register, and bit 3 of the status byte will be set to logic high. The position is recorded when both encoder-phase inputs and the index pulse input are logic low. This register can then be read by the user (see description for command RDIP) to facilitate aligning the definition of home position (see description of command DFH) with an index pulse. The user can also arrange to have the LM628 interrupt the host to signify that an index pulse has occurred. See the descriptions for commands MSKI and RSTI.

LPEI COMMAND: Load Position Error for Interrupt

Command Code: 1B Hex
Data Bytes: Two
Data Range: 0000 to 7FFF Hex
Executable During Motion: Yes

An excessive position error (the output of the loop summing junction) can indicate a serious system problem; e.g., a stalled rotor. Instruction LPEI allows the user to input a threshold for position error detection. Error detection occurs when the absolute magnitude of the position error exceeds the threshold, which results in bit 5 of the status byte being set to logic high. If it is desired to also stop (turn off) the motor upon detecting excessive position error, see command LPES, below. The first byte of threshold data written with command LPEI is the more significant. The user can have the LM628 interrupt the host to signify that an excessive position error has occurred. See the descriptions for commands MSKI and RSTI.

Interrupt Control Commands (Continued)

LPES COMMAND: Load Position Error for Stopping

Command Code: 1A Hex
 Data Bytes: Two
 Data Range: 0000 to 7FFF Hex
 Executable During Motion: Yes

Instruction LPES is essentially the same as command LPEI above, but adds the feature of turning off the motor upon detecting excessive position error. The motor drive is not actually switched off, it is set to half-scale, the offset-binary code for zero. As with command LPEI, bit 5 of the status byte is also set to logic high. The first byte of threshold data written with command LPES is the more significant. The user can have the LM628 interrupt the host to signify that an excessive position error has occurred. See the descriptions for commands MSKI and RSTI.

SBPA COMMAND:

Command Code: 20 Hex
 Data Bytes: Four
 Data Range: C0000000 to 3FFFFFFF Hex
 Executable During Motion: Yes

This command enables the user to set a breakpoint in terms of absolute position. Bit 6 of the status byte is set to logic high when the breakpoint position is reached. This condition is useful for signaling trajectory and/or filter parameter updates. The user can also arrange to have the LM628 interrupt the host to signify that a breakpoint position has been reached. See the descriptions for commands MSKI and RSTI.

SBPR COMMAND:

Command Code: 21 Hex
 Data Bytes: Four
 Data Range: See Text
 Executable During Motion: Yes

This command enables the user to set a breakpoint in terms of relative position. As with command SBPA, bit 6 of the status byte is set to logic high when the breakpoint position (relative to the current commanded target position) is reached. The relative breakpoint input value must be such that when this value is added to the target position the result remains within the absolute position range of the system (C0000000 to 3FFFFFFF hex). This condition is useful for signaling trajectory and/or filter parameter updates. The user can also arrange to have the LM628 interrupt the host to signify that a breakpoint position has been reached. See the descriptions for commands MSKI and RSTI.

MSKI COMMAND: MaSK Interrupts

Command Code: 1C Hex
 Data Bytes: Two
 Data Range: See Text
 Executable During Motion: Yes

The MSKI command lets the user determine which potential interrupt condition(s) will interrupt the host. Bits 1 through 6 of the status byte are indicators of the six conditions which are candidates for host interrupt(s). When interrupted, the host then reads the status byte to learn which condition(s) occurred. Note that the MSKI command is immediately followed by two data bytes. Bits 1 through 6 of the second (less significant) byte written determine the masked/un-masked status of each potential interrupt. Any zero(s) in this

6-bit field will mask the corresponding interrupt(s); any one(s) enable the interrupt(s). Other bits comprising the two bytes have no effect. The mask controls only the host interrupt process; reading the status byte will still reflect the actual conditions independent of the mask byte. See Table III.

TABLE III. Mask and Reset Bit Allocations for Interrupts

| Bit Position | Function |
|----------------|-------------------------------|
| Bits 15 thru 7 | Not Used |
| Bit 6 | Breakpoint Interrupt |
| Bit 5 | Position-Error Interrupt |
| Bit 4 | Wrap-Around Interrupt |
| Bit 3 | Index-Pulse Interrupt |
| Bit 2 | Trajectory-Complete Interrupt |
| Bit 1 | Command-Error Interrupt |
| Bit 0 | Not Used |

RSTI COMMAND: ReSeT Interrupts

Command Code: 1D Hex
 Data Bytes: Two
 Data Range: See Text
 Executable During Motion: Yes

When one of the potential interrupt conditions of Table III occurs, command RSTI is used to reset the corresponding interrupt flag bit in the status byte. The host may reset one or all flag bits. Resetting them one at a time allows the host to service them one at a time according to a priority programmed by the user. As in the MSKI command, bits 1 through 6 of the second (less significant) byte correspond to the potential interrupt conditions shown in Table III. Also see description of RSTAT command. Any zero(s) in this 6-bit field reset the corresponding interrupt(s). The remaining bits have no effect.

Filter Control Commands

The following two LM628 user commands are used for setting the derivative-term sampling interval, for adjusting the filter parameters as required to tune the system, and to control the timing of these system changes.

LFIL COMMAND: Load FILter Parameters

Command Code: 1E Hex
 Data Bytes: Two to Ten
 Data Ranges ...
 Filter Control Word: See Text
 Filter Coefficients: 0000 to 7FFF Hex (Pos Only)
 Integration Limit: 0000 to 7FFF Hex (Pos Only)
 Executable During Motion: Yes

The filter parameters (coefficients) which are written to the LM628 to control loop compensation are: k_p , k_i , k_d , and il (integration limit). The integration limit (il) constrains the contribution of the integration term

$$\left[k_i \cdot \sum_{N=0}^n e(n) \right]$$

(see Eq. 1) to values equal to or less than a user-defined maximum value; this capability minimizes integral or reset "wind-up" (an overshooting effect of the integral action). The positive-only input value is compared to the absolute

Filter Control Commands (Continued)

magnitude of the integration term; when the magnitude of integration term value exceeds *il*, the *il* value (with appropriate sign) is substituted for the integration term value.

The derivative-term sampling interval is also programmable via this command. After writing the command code, the first two data bytes that are written specify the derivative-term sampling interval and which of the four filter parameters is/are to be written via any forthcoming data bytes. The first byte written is the more significant. Thus the two data bytes constitute a filter control word that informs the LM628 as to the nature and number of any following data bytes. See Table IV.

TABLE IV. Filter Control word Bit Allocation

| Bit Position | Function |
|--------------|------------------------------------|
| Bit 15 | Derivative Sampling Interval Bit 7 |
| Bit 14 | Derivative Sampling Interval Bit 6 |
| Bit 13 | Derivative Sampling Interval Bit 5 |
| Bit 12 | Derivative Sampling Interval Bit 4 |
| Bit 11 | Derivative Sampling Interval Bit 3 |
| Bit 10 | Derivative Sampling Interval Bit 2 |
| Bit 9 | Derivative Sampling Interval Bit 1 |
| Bit 8 | Derivative Sampling Interval Bit 0 |
| Bit 7 | Not Used |
| Bit 6 | Not Used |
| Bit 5 | Not Used |
| Bit 4 | Not Used |
| Bit 3 | Loading <i>kp</i> Data |
| Bit 2 | Loading <i>ki</i> Data |
| Bit 1 | Loading <i>kd</i> Data |
| Bit 0 | Loading <i>il</i> Data |

Bits 8 through 15 select the derivative-term sampling interval. See Table V. The user must locally save and restore these bits during successive writes of the filter control word.

Bits 4 through 7 of the filter control word are not used.

Bits 0 to 3 inform the LM628 as to whether any or all of the filter parameters are about to be written. The user may choose to update any or all (or none) of the filter parameters. Those chosen for updating are so indicated by logic one(s) in the corresponding bit position(s) of the filter control word.

The data bytes specified by and immediately following the filter control word are written in pairs to comprise 16-bit words. The order of sending the data words to the LM628 corresponds to the descending order shown in the above description of the filter control word; i.e., beginning with *kp*, then *ki*, *kd* and *il*. The first byte of each word is the more-significant byte. Prior to writing a word (byte pair) it is necessary to check the busy bit in the status byte for readiness. The required data is written to the primary buffers of a double-buffered scheme by the above described operations; it is not transferred to the secondary (working) registers until the UDF command is executed. This fact can be used advantageously; the user can input numerous data ahead of their actual use. This simple pipeline effect can relieve potential host computer data communications bottlenecks, and facilitates easier synchronization of multiple-axis controls.

UDF COMMAND: UpDate Filter

Command Code: 04 Hex
Data Bytes: None
Executable During Motion: Yes

The UDF command is used to update the filter parameters, the specifics of which have been programmed via the LFIL command. Any or all parameters (derivative-term sampling interval, *kp*, *ki*, *kd*, and/or *il*) may be changed by the appropriate command(s), but command UDF must be executed to affect the change in filter tuning. Filter updating is synchronized with the calculations to eliminate erratic or spurious behavior.

Trajectory Control Commands

The following two LM628 user commands are used for setting the trajectory control parameters (position, velocity, acceleration), mode of operation (position or velocity), and direction (velocity mode only) as required to describe a desired motion or to select the mode of a manually directed stop, and to control the timing of these system changes.

LTRJ COMMAND: Load TRAJectory Parameters

Command Code: 1F Hex
Data Bytes: Two to Fourteen
Data Ranges . . .
Trajectory Control Word: See Text
Position: C0000000 to 3FFFFFFF Hex
Velocity: 00000000 to 3FFFFFFF Hex
(Pos Only)
Acceleration: 00000000 to 3FFFFFFF Hex
(Pos Only)
Executable During Motion: Conditionally, See Text

TABLE V. Derivative-Term Sampling Interval Selection Codes

| | Bit Position | | | | | | | | Selected Derivative Sampling Interval |
|------|--------------|----|----|----|----|----|---|---|---------------------------------------|
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 256 μ s |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 512 μ s |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 768 μ s |
| | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1024 μ s, etc . . . |
| thru | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 65,536 μ s |

Note: Sampling intervals shown are when using an 8.0 MHz clock. The 256 corresponds to 2048/8 MHz; sample intervals must be scaled for other clock frequencies.

Trajectory Control Commands (Continued)

The trajectory control parameters which are written to the LM628 to control motion are: acceleration, velocity, and position. In addition, indications as to whether these three parameters are to be considered as absolute or relative inputs, selection of velocity mode and direction, and manual stopping mode selection and execution are programmable via this command. After writing the command code, the first two data bytes that are written specify which parameter(s) is/are being changed. The first byte written is the more significant. Thus the two data bytes constitute a trajectory control word that informs the LM628 as to the nature and number of any following data bytes. See Table VI.

TABLE VI. Trajectory Control Word Bit Allocation

| Bit Position | Function |
|--------------|--|
| Bit 15 | Not Used |
| Bit 14 | Not Used |
| Bit 13 | Not Used |
| Bit 12 | Forward Direction (Velocity Mode Only) |
| Bit 11 | Velocity Mode |
| Bit 10 | Stop Smoothly (Decelerate as Programmed) |
| Bit 9 | Stop Abruptly (Maximum Deceleration) |
| Bit 8 | Turn Off Motor (Output Zero Drive) |
| Bit 7 | Not Used |
| Bit 6 | Not Used |
| Bit 5 | Acceleration Will Be Loaded |
| Bit 4 | Acceleration Data Is Relative |
| Bit 3 | Velocity Will Be Loaded |
| Bit 2 | Velocity Data Is Relative |
| Bit 1 | Position Will Be Loaded |
| Bit 0 | Position Data Is Relative |

Bit 12 determines the motor direction when in the velocity mode. A logic one indicates forward direction. This bit has no effect when in position mode.

Bit 11 determines whether the LM628 operates in velocity mode (Bit 11 logic one) or position mode (Bit 11 logic zero).

Bits 8 through 10 are used to select the method of *manually stopping* the motor. These bits are *not* provided for one to merely specify the desired *mode* of stopping, in position mode operations, normal stopping is always smooth and occurs automatically at the end of the specified trajectory. Under exceptional circumstances it may be desired to manually intervene with the trajectory generation process to affect a premature stop. In velocity mode operations, however, the normal means of stopping *is* via bits 8 through 10 (usually bit 10). Bit 8 is set to logic one to stop the motor by turning off motor drive output (outputting the appropriate off-set-binary code to apply zero drive to the motor); bit 9 is set to one to stop the motor abruptly (at maximum available acceleration, by setting the target position equal to the current position); and bit 10 is set to one to stop the motor smoothly by using the current user-programmed acceleration value. Bits 8 through 10 are to be used *exclusively*; only one bit should be a logic one at any time.

Bits 0 through 5 inform the LM628 as to whether any or all of the trajectory controlling parameters are about to be written, and whether the data should be interpreted as absolute or relative. The user may choose to update any or all (or

none) of the trajectory parameters. Those chosen for updating are so indicated by logic one(s) in the corresponding bit position(s). Any parameter may be changed while the motor is in motion; however, if acceleration is changed then the next STT command must not be issued until the LM628 has completed the current move or has been manually stopped.

The data bytes specified by and immediately following the trajectory control word are written in pairs which comprise 16-bit words. Each data item (parameter) requires two 16-bit words; the word and byte order is most-to-least significant. The order of sending the parameters to the LM628 corresponds to the descending order shown in the above description of the trajectory control word; i.e., beginning with acceleration, then velocity, and finally position.

Acceleration and velocity are 32 bits, positive only, but range only from 0 (00000000 hex) to $[2^{30}] - 1$ (3FFFFFFF hex). The bottom 16 bits of both acceleration and velocity are scaled as fractional data; therefore, the least-significant integer data bit for these parameters is bit 16 (where the bits are numbered 0 through 31). To determine the coding for a given velocity, for example, one multiplies the desired velocity (in counts per sample interval) times 65,536 and converts the result to binary. The units of acceleration are counts per sample per sample. The value loaded for acceleration must not exceed the value loaded for velocity. Position is a signed, 32-bit integer, but ranges only from $-[2^{30}]$ (C0000000 hex) to $[2^{30}] - 1$ (3FFFFFFF Hex).

The required data is written to the primary buffers of a double-buffered scheme by the above described operations; it is not transferred to the secondary (working) registers until the STT command is executed. This fact can be used advantageously; the user can input numerous data ahead of their actual use. This simple pipeline effect can relieve potential host computer data communications bottlenecks, and facilitates easier synchronization of multiple-axis controls.

STT COMMAND: STArT Motion Control

Command Code: 01 Hex
 Data Bytes: None
 Executable During Motion: Yes, if acceleration has not been changed

The STT command is used to execute the desired trajectory, the specifics of which have been programmed via the LTRJ command. Synchronization of multi-axis control (to within one sample interval) can be arranged by loading the required trajectory parameters for each (and every) axis and then simultaneously issuing a single STT command to all axes. This command may be executed at any time, unless the acceleration value has been changed and a trajectory has not been completed or the motor has not been manually stopped. If STT is issued during motion and acceleration has been changed, a command error interrupt will be generated and the command will be ignored.

Data Reporting Commands

The following seven LM628 user commands are used to obtain data from various registers in the LM628. Status, position, and velocity information are reported. With the exception of RDSTAT, the data is read from the LM628 data port after first writing the corresponding command to the command port.

Data Reporting Commands (Continued)

RDSTAT COMMAND: Read STATus Byte

Command Code: None
 Byte Read: One
 Data Range: See Text
 Executable During Motion: Yes

The RDSTAT command is really not a command, but is listed with the other commands because it is used very frequently to control communications with the host computer. There is no identification code; it is directly supported by the hardware and may be executed at any time. The single-byte status read is selected by placing CS, PS and RD at logic zero. See Table VII.

TABLE VII. Status Byte Bit Allocation

| Bit Position | Function |
|--------------|--------------------------------------|
| Bit 7 | Motor Off |
| Bit 6 | Breakpoint Reached [Interrupt] |
| Bit 5 | Excessive Position Error [Interrupt] |
| Bit 4 | Wraparound Occurred [Interrupt] |
| Bit 3 | Index Pulse Observed [Interrupt] |
| Bit 2 | Trajectory Complete [Interrupt] |
| Bit 1 | Command Error [Interrupt] |
| Bit 0 | Busy Bit |

Bit 7, the motor-off flag, is set to logic one when the motor drive output is off (at the half-scale, offset-binary code for zero). The motor is turned off by any of the following conditions: power-up reset, command RESET, excessive position error (if command LPES had been executed), or when command LTRJ is used to manually stop the motor via turning the motor off. Note that when bit 7 is set in conjunction with command LTRJ for producing a manual, motor-off stop, the actual setting of bit 7 does not occur until command STT is issued to affect the stop. Bit 7 is cleared by command STT, except as described in the previous sentence.

Bit 6, the breakpoint-reached interrupt flag, is set to logic one when the position breakpoint loaded via command SBPA or SBPR has been exceeded. The flag is functional independent of the host interrupt mask status. Bit 6 is cleared via command RSTI.

Bit 5, the excessive-position-error interrupt flag, is set to logic one when a position-error interrupt condition exists. This occurs when the error threshold loaded via command LPEI or LPES has been exceeded. The flag is functional independent of the host interrupt mask status. Bit 5 is cleared via command RSTI.

Bit 4, the wraparound interrupt flag, is set to logic one when a numerical "wraparound" has occurred. To "wraparound" means to exceed the position address space of the LM628, which could occur during velocity mode operation. If a wraparound has occurred, then position information will be in error and this interrupt helps the user to ensure position data integrity. The flag is functional independent of the host interrupt mask status. Bit 4 is cleared via command RSTI.

Bit 3, the index-pulse acquired interrupt flag, is set to logic one when an index pulse has occurred (if command SIP had been executed) and indicates that the index position register has been updated. The flag is functional independent of the host interrupt mask status. Bit 3 is cleared by command STI.

Bit 2, the trajectory complete interrupt flag, is set to logic one when the trajectory programmed by the LTRJ command and initiated by the STT command has been completed. Because of overshoot or a limiting condition (such as commanding the velocity to be higher than the motor can achieve), the motor may not yet be at the final commanded position. This bit is the logical OR of bits 7 and 10 of the Signals Register, see command RDSIGS below. The flag functions independently of the host interrupt mask status. Bit 2 is cleared via command RSTI.

Bit 1, the command-error interrupt flag, is set to logic one when the user attempts to read data when a write was appropriate (or vice versa). The flag is functional independent of the host interrupt mask status. Bit 1 is cleared via command RSTI.

Bit 0, the busy flag, is frequently tested by the user (via the host computer program) to determine the busy/ready status prior to writing and reading any data. Such writes and reads may be executed only when bit 0 is logic zero (not busy). Any command or data writes when the busy bit is high will be ignored. Any data reads when the busy bit is high will read the current contents of the I/O port buffers, not the data expected by the host. Such reads or writes (with the busy bit high) will not generate a command-error interrupt.

RDSIGS COMMAND: Read SIGNALS Register

Command Code: 0C Hex
 Bytes Read: Two
 Data Range: See Text
 Executable During Motion: Yes

The LM628 internal "signals" register may be read using this command. The first byte read is the more significant. The less significant byte of this register (with the exception of bit 0) duplicates the status byte. See Table VIII.

TABLE VIII. Signals Register Bit Allocation

| Bit Position | Function |
|--------------|---|
| Bit 15 | Host Interrupt |
| Bit 14 | Acceleration Loaded (But Not Updated) |
| Bit 13 | UDF Executed (But Filter Not yet Updated) |
| Bit 12 | Forward Direction |
| Bit 11 | Velocity Mode |
| Bit 10 | On Target |
| Bit 9 | Turn Off upon Excessive Position Error |
| Bit 8 | Eight-Bit Output Mode |
| Bit 7 | Motor Off |
| Bit 6 | Breakpoint Reached [Interrupt] |
| Bit 5 | Excessive Position Error [Interrupt] |
| Bit 4 | Wraparound Occurred [Interrupt] |
| Bit 3 | Index Pulse Acquired [Interrupt] |
| Bit 2 | Trajectory Complete [Interrupt] |
| Bit 1 | Command Error [Interrupt] |
| Bit 0 | Acquire Next Index (SIP Executed) |

Bit 15, the host interrupt flag, is set to logic one when the host interrupt output (Pin 17) is logic one. Pin 17 is set to logic one when any of the six host interrupt conditions occur (if the corresponding interrupt has not been masked). Bit 15 (and Pin 17) are cleared via command RSTI.

Bit 14, the acceleration-loaded flag, is set to logic one when acceleration data is written to the LM628. Bit 14 is cleared by the STT command.

Data Reporting Commands (Continued)

Bit 13, the UDF-executed flag, is set to logic one when the UDF command is executed. Because bit 13 is cleared at the end of the sampling interval in which it has been set, this signal is very short-lived and probably not very profitable for monitoring.

Bit 12, the forward direction flag, is meaningful only when the LM628 is in velocity mode. The bit is set to logic one to indicate that the desired direction of motion is "forward"; zero indicates "reverse" direction. Bit 12 is set and cleared via command LTRJ. The actual setting and clearing of bit 12 does not occur until command STT is executed.

Bit 11, the velocity mode flag, is set to logic one to indicate that the user has selected (via command LTRJ) velocity mode. Bit 11 is cleared when position mode is selected (via command LTRJ). The actual setting and clearing of bit 11 does not occur until command STT is executed.

Bit 10, the on-target flag, is set to logic one when the trajectory generator has completed its functions for the last-issued STT command. Bit 10 is cleared by the next STT command.

Bit 9, the turn-off on-error flag, is set to logic one when command LPES is executed. Bit 9 is cleared by command LPEI.

Bit 8, the 8-bit output flag, is set to logic one when the LM628 is reset, or when command PORT8 is executed. Bit 8 is cleared by command PORT12.

Bits 0 through 7 replicate the status byte (see Table VII), with the exception of bit 0. Bit 0, the acquire next index flag, is set to logic one when command SIP is executed; it then remains set until the next index pulse occurs.

RDIP COMMAND: Read Index Position

Command Code: 09 Hex
Bytes Read: Four
Data Range: C0000000 to 3FFFFFFF Hex
Executable During Motion: Yes

This command reads the position recorded in the index register. Reading the index register can be part of a system error checking scheme. Whenever the SIP command is executed, the new index position minus the old index position, divided by the incremental encoder resolution (encoder lines times four), should always be an integral number. The RDIP command facilitates acquiring these data for host-based calculations. The command can also be used to identify/verify home or some other special position. The bytes are read in most-to-least significant order.

RDDP COMMAND: Read Desired Position

Command Code: 08 Hex
Bytes Read: Four
Data Range: C0000000 to 3FFFFFFF Hex
Executable During Motion: Yes

This command reads the instantaneous desired (current *temporal*) position output of the profile generator. This is the "setpoint" input to the position-loop summing junction. The bytes are read in most-to-least significant order.

RDRP COMMAND: Read Real Position

Command Code: 0A Hex
Bytes Read: Four
Data Range: C0000000 to 3FFFFFFF Hex
Executable During Motion: Yes

This command reads the current actual position of the motor. This is the feedback input to the loop summing junction. The bytes are read in most-to-least significant order.

RDDV COMMAND: Read Desired Velocity

Command Code: 07 Hex
Bytes Read: Four
Data Range: C0000001 to 3FFFFFFF
Executable During Motion: Yes

This command reads the integer and fractional portions of the instantaneous desired (current *temporal*) velocity, as used to generate the desired position profile. The bytes are read in most-to-least significant order. The value read is properly scaled for numerical comparison with the user-supplied (commanded) velocity; however, because the two least-significant bytes represent *fractional* velocity, only the two most-significant bytes are appropriate for comparison with the data obtained via command RDRV (see below). Also note that, although the velocity input data is constrained to positive numbers (see command LTRJ), the data returned by command RDDV represents a *signed* quantity where negative numbers represent operation in the reverse direction.

RDRV COMMAND: Read Real Velocity

Command Code: 0B Hex
Bytes Read: Two
Data Range: C000 to 3FFF Hex, See Text
Executable During Motion: Yes

This command reads the *integer* portion of the instantaneous actual velocity of the motor. The internally maintained fractional portion of velocity is not reported because the reported data is derived by reading the incremental encoder, which produces only integer data. For comparison with the result obtained by executing command RDDV (or the user-supplied input value), the value returned by command RDRV must be multiplied by 216 (shifted left 16 bit positions). Also, as with command RDDV above, data returned by command RDRV is a *signed* quantity, with negative values representing reverse-direction motion.

RDSUM COMMAND: Read Integration-Term SUMmation Value

Command Code: 0D Hex
Bytes Read: Two
Data Range: 00000 Hex to \pm the Current Value of the Integration Limit
Executable During Motion: Yes

This command reads the value to which the integration term has accumulated. The ability to read this value may be helpful in initially or adaptively tuning the system.

Typical Applications

Programming LM628 Host Handshaking (Interrupts)

A few words regarding the LM628 host handshaking will be helpful to the system programmer. As indicated in various portions of the above text, the LM628 handshakes with the host computer in two ways: via the host interrupt output (Pin 17), or via polling the status byte for "interrupt" conditions. When the hardwired interrupt is used, the status byte is also read and parsed to determine which of six possible conditions caused the interrupt.

Typical Applications (Continued)

When using the hardwired interrupt it is very important that the host interrupt service routine does not interfere with a command sequence which might have been in progress when the interrupt occurred. If the host interrupt service routine were to issue a command to the LM628 while it is in the middle of an ongoing command sequence, the ongoing command will be aborted (which could be detrimental to the application).

Two approaches exist for avoiding this problem. If one is using hardwired interrupts, they should be disabled at the host prior to issuing any LM628 command sequence, and re-enabled after each command sequence. The second approach is to avoid hardwired interrupts and poll the LM628 status byte for "interrupt" status. The status byte always reflects the interrupt-condition status, independent of whether or not the interrupts have been masked.

Typical Host Computer/Processor Interface

The LM628 is interfaced with the host computer/processor via an 8-bit parallel bus. *Figure 12* shows such an interface and a minimum system configuration.

As shown in *Figure 12*, the LM628 interfaces with the host data, address and control lines. The address lines are decoded to generate the LM628 \overline{CS} input; the host address LSB directly drives the LM628 \overline{PS} input. *Figure 12* also shows an 8-bit DAC and an LM12 Power Op Amp interfaced to the LM628.

LM628 and High Performance Controller (HPC) Interface

Figure 13 shows the LM628 interfaced to a National HPC High Performance Controller. The delay and logic associated with the \overline{WR} line is used to effectively increase the write-data hold time of the HPC (as seen at the LM628) by causing the \overline{WR} pulse to rise early. Note that the HPC CK2 output provides the clock for the LM628. The 74LS245 is used to decrease the read-data hold time, which is necessary when interfacing to fast host busses.

Interfacing a 12-Bit DAC

Figure 14 illustrates use of a 12-bit DAC with the LM628. The 74LS378 hex gated-D flip-flop and an inverter demultiplex the 12-bit output. DAC offset must be adjusted to minimize DAC linearity and monotonicity errors. Two methods exist for making this adjustment. If the DAC1210 has been socketed, remove it and temporarily connect a 15 k Ω resistor between Pins 11 and 13 of the DAC socket (Pins 2 and 6 of the LF356) and adjust the 25 k Ω potentiometer for 0V at Pin 6 of the LF356.

If the DAC is not removable, the second method of adjustment requires that the DAC1210 inputs be presented an all-zeros code. This can be arranged by commanding the appropriate move via the LM628, but with no feedback from the system encoder. When the all-zeros code is present, adjust the pot for 0V at Pin 6 of the LF356.

A Monolithic Linear Drive Using LM12 Power Op Amp

Figure 15 shows a motor-drive amplifier built using the LM12 Power Operational Amplifier. This circuit is very simple and can deliver up to 8A at 30V (using the LM12L/LM12CL). Resistors R1 and R2 should be chosen to set the gain to provide maximum output voltage consistent with maximum input voltage. This example provides a gain of 2.2, which allows for amplifier output saturation at $\pm 22V$ with a $\pm 10V$ input, assuming power supply voltages of $\pm 30V$. The amplifier gain should not be higher than necessary because the system is non-linear when saturated, and because gain should be controlled by the LM628. The LM12 can also be configured as a current driver, see 1987 Linear Databook, Vol. 1, p. 2-280.

Typical PWM Motor Drive Interfaces

Figure 16 shows an LM18298 dual full-bridge driver interfaced to the LM629 PWM outputs to provide a switch-mode power amplifier for driving small brush/commutator motors. *Figure 17* shows an LM621 brushless motor commutator interfaced to the LM629 PWM outputs and a discrete device switch-mode power amplifier for driving brushless DC motors.

Incremental Encoder Interface

The incremental (position feedback) encoder interface consists of three lines: Phase A (Pin 2), Phase B (Pin 3), and Index (Pin 1). The index pulse output is not available on some encoders. The LM628 will work with both encoder types, but commands SIP and RDIP will not be meaningful without an index pulse (or alternative input for this input ... be sure to tie Pin 1 high if not used).

Some consideration is merited relative to use in high Gaussian-noise environments. If noise is added to the encoder inputs (either or both inputs) and is such that it is not sustained until the next encoder transition, the LM628 decoder logic will reject it. Noise that mimics quadrature counts or persists through encoder transitions must be eliminated by appropriate EMI design.

Simple digital "filtering" schemes merely reduce susceptibility to noise (there will always be noise pulses longer than the filter can eliminate). Further, any noise filtering scheme reduces decoder bandwidth. In the LM628 it was decided (since simple filtering does not eliminate the noise problem) to not include a noise filter in favor of offering maximum possible decoder bandwidth. Attempting to drive encoder signals too long a distance with simple TTL lines can also be a source of "noise" in the form of signal degradation (poor risetime and/or ringing). This can also cause a system to lose positional integrity. Probably the most effective countermeasure to noise induction can be had by using balanced-line drivers and receivers on the encoder inputs. *Figure 18* shows circuitry using the DS26LS31 and DS26LS32.

Typical Applications (Continued)

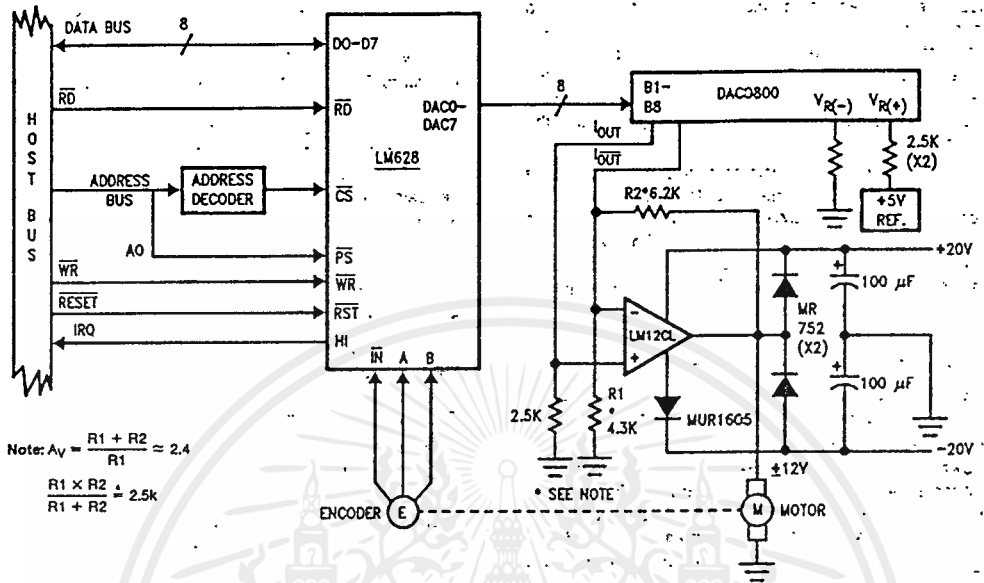


FIGURE 12. Host Interface and Minimum System Configuration

TL/H/9219-14

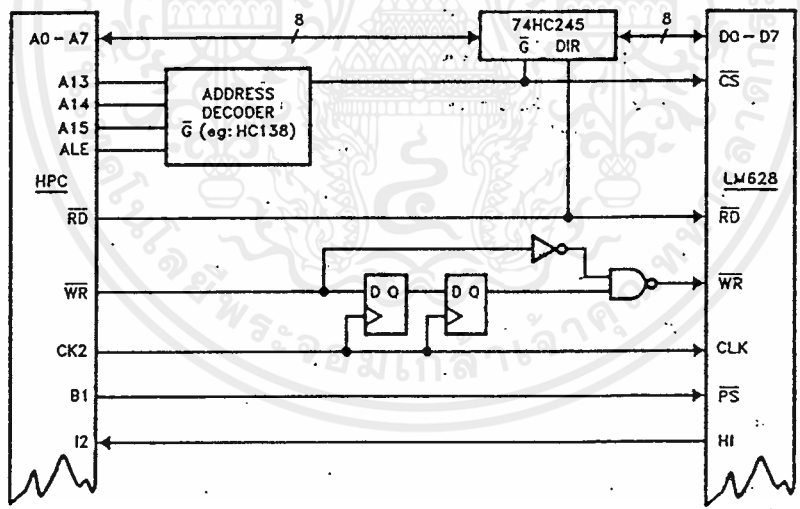


FIGURE 13. LM628 and HPC Interface

TL/H/9219-15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Typical Applications (Continued)

TL/H/9210-16

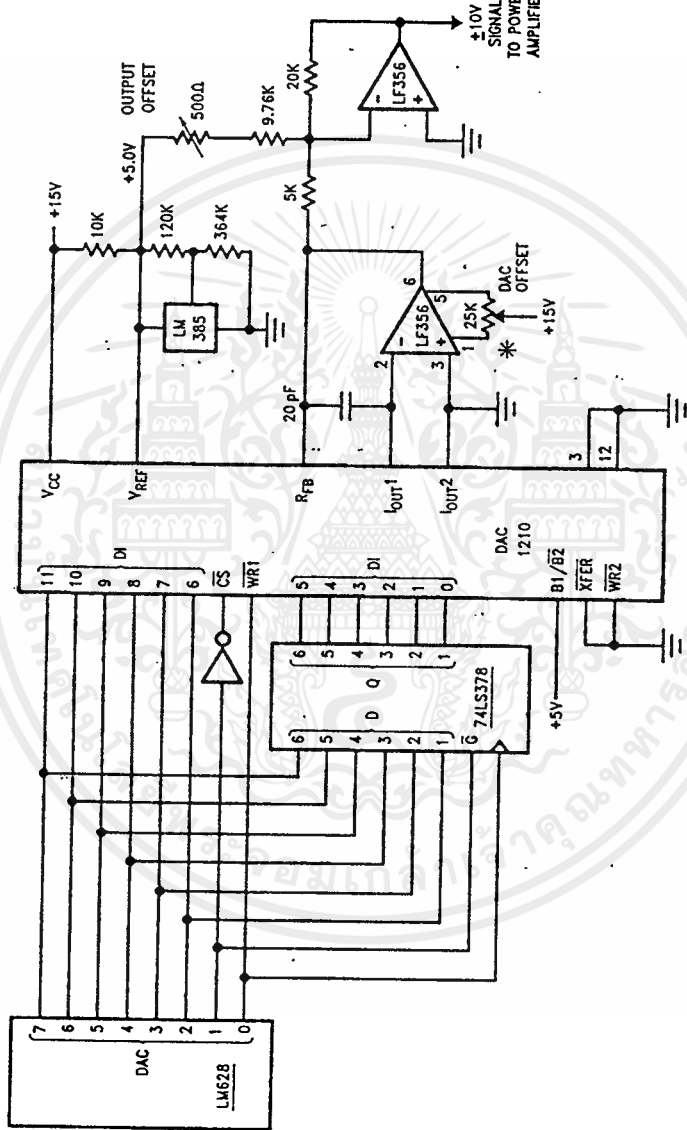


FIGURE 14. Interfacing a 12-Bit DAC and LM628
 *DAC offset must be adjusted to minimize DAC linearity and monotonicity errors. See text.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

typical Applications (Continued)

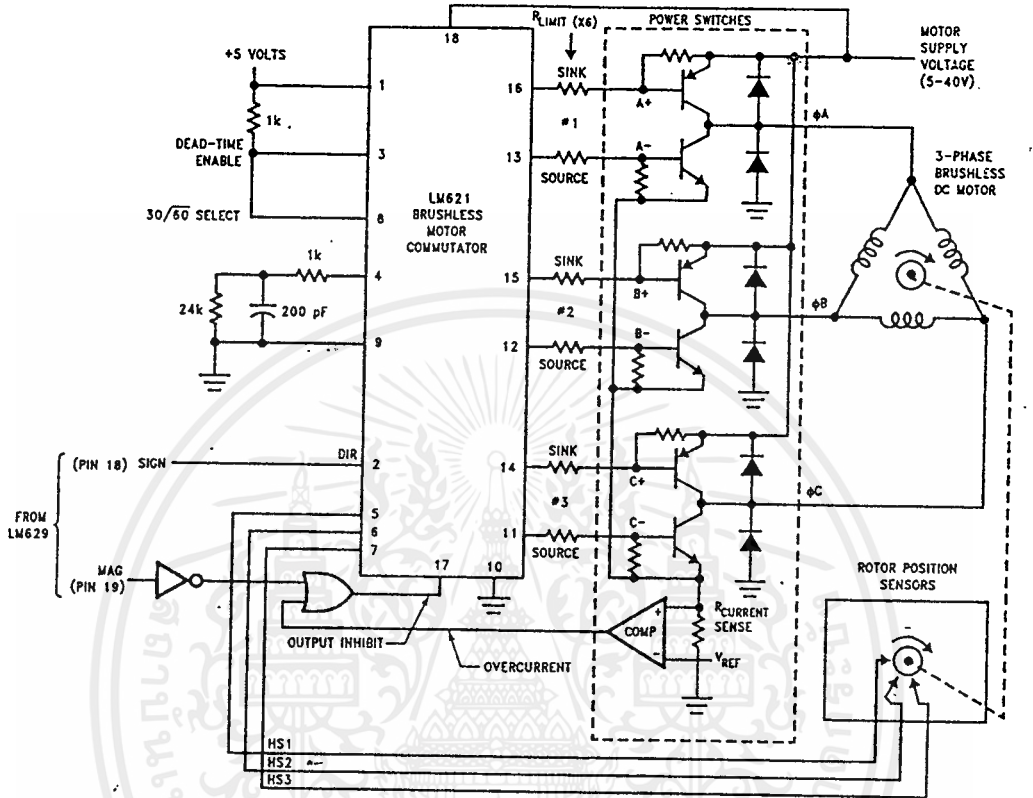


FIGURE 17. PWM Drive for Brushless Motors

TL/H/9219-19

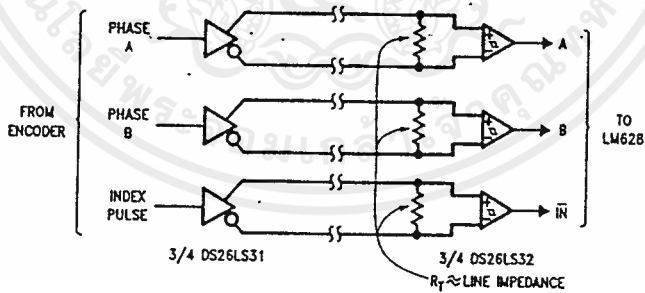
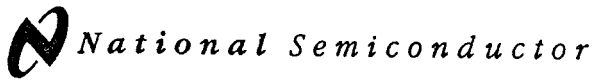


FIGURE 18. Typical Balanced-Line Encoder Input Circuit

TL/H/9219-20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



LMD18201 3A, 55V H-Bridge

General Description

The LMD18201 is a 3A H-Bridge designed for motion control applications. The device is built using a multi-technology process which combines bipolar and CMOS control circuitry with DMOS power devices on the same monolithic structure. The H-Bridge configuration is ideal for driving DC and stepper motors. The LMD18201 accommodates peak output currents up to 6A. Current sensing can be achieved via a small sense resistor connected in series with the power ground lead. For current sensing without disturbing the path of current to the load, the LMD18200 is recommended.

Features

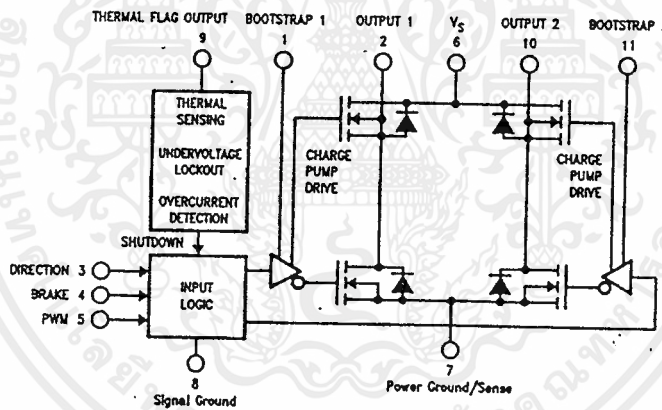
- Delivers up to 3A continuous output
- Operates at supply voltages up to 55V
- Low $R_{DS(ON)}$ typically 0.33 Ω per switch

- TTL and CMOS compatible inputs
- No "shoot-through" current
- Thermal warning flag output at 145°C
- Thermal shutdown (outputs off) at 170°C
- Internal clamp diodes
- Shorted load protection
- Internal charge pump with external bootstrap capability

Applications

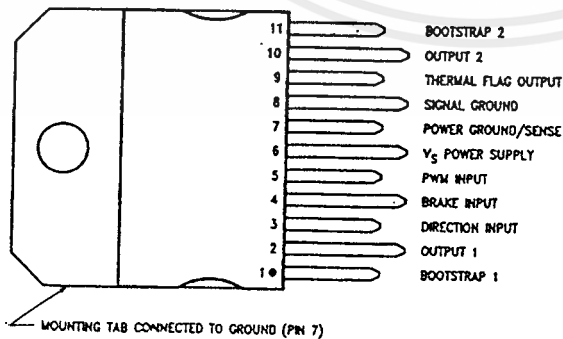
- DC and stepper motor drives
- Position and velocity servomechanisms
- Factory automation robots
- Numerically controlled machinery
- Computer printers and plotters

Functional Diagram



TL/H/10793-1

Connection Diagram and Ordering Information



Top View

TL/H/10793-2

Order Number LMD18201T
See NS Package Number TA11B

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

| | |
|---|-------------------|
| Total Supply Voltage (V_S , Pin 6) | 60V |
| Voltage at Pins 3, 4, 5 and 9 | 12V |
| Voltage at Bootstrap Pins (Pins 1 and 11) | $V_{OUT} \pm 16V$ |
| Peak Output Current (200 ms) | 6A |
| Continuous Output Current (Note 2) | 3A |
| Power Dissipation (Note 3) | 25W |
| Sense Voltage (Pin 7 to Pin 8) | +0.5V to -1.0V |

| | |
|--|-----------------|
| Power Dissipation ($T_A = 25^\circ\text{C}$, Free Air) | 3W |
| Junction Temperature, $T_{J(max)}$ | 150°C |
| ESD Susceptibility (Note 4) | 1500V |
| Storage Temperature, T_{STG} | -65°C to +150°C |
| Lead Temperature (Soldering, 10 sec.) | 300°C |

Operating Ratings (Note 1)

| | |
|-----------------------------|-----------------|
| Junction Temperature, T_J | -40°C to +125°C |
| V_S Supply Voltage | +12V to +55V |

Electrical Characteristics

The following specifications apply for $V_S = 42V$, unless otherwise specified. Boldface limits apply over the entire operating temperature range, $-40^\circ\text{C} \leq T_J \leq +125^\circ\text{C}$, all other limits are for $T_A = T_J = 25^\circ\text{C}$. (Note 5)

| Symbol | Parameter | Conditions | Typ | Limit | Units |
|--------------|---------------------------------|--|------------|----------------|---------------------|
| $R_{DS(ON)}$ | Switch ON Resistance | Output Current = 3A (Note 6) | 0.33 | 0.4/0.6 | Ω (max) |
| $R_{DS(ON)}$ | Switch ON Resistance | Output Current = 6A (Note 6) | 0.33 | 0.4/0.6 | Ω (max) |
| V_{CLAMP} | Clamp Diode Forward Drop | Clamp Current = 3A (Note 6) | 1.2 | 1.5 | V (max) |
| V_{IL} | Logic Low Input Voltage | Pins 3, 4, 5 | | -0.1 0.8 | V (min) V (max) |
| I_{IL} | Logic Low Input Current | $V_{IN} = -0.1V$, Pins = 3, 4, 5 | | -10 | μA (max) |
| V_{IH} | Logic High Input Voltage | Pins 3, 4, 5 | | 2 12 | V (min) V (max) |
| I_{IH} | Logic High Input Current | $V_{IN} = 12V$, Pins = 3, 4, 5 | | 10 | μA (max) |
| | Undervoltage Lockout | Outputs Turn OFF | | 9 11 | V (min) V (max) |
| T_{JW} | Warning Flag Temperature | Pin 9 $\leq 0.8V$, $I_L = 2\text{ mA}$ | 145 | | °C |
| $V_{F(ON)}$ | Flag Output Saturation Voltage | $T_J = T_{JW}$, $I_L = 2\text{ mA}$ | 0.15 | | V |
| $I_{F(OFF)}$ | Flag Output Leakage | $V_F = 12V$ | 0.2 | 10 | μA (max) |
| T_{JSD} | Shutdown Temperature | Outputs Turn OFF | 170 | | °C |
| I_S | Quiescent Supply Current | All Logic Inputs Low | 13 | 25 | mA (max) |
| $t_{D(ON)}$ | Output Turn-On Delay Time | Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$ | 300 300 | | ns ns |
| t_{ON} | Output Turn-On Switching Time | Bootstrap Capacitor = 10 nF Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$ | 100 80 | | ns ns |
| $t_{D(OFF)}$ | Output Turn-Off Delay Times | Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$ | 200 200 | | ns ns |
| t_{OFF} | Output Turn-Off Switching Times | Bootstrap Capacitor = 10 nF Sourcing Outputs, $I_{OUT} = 3A$ Sinking Outputs, $I_{OUT} = 3A$ | 75 70 | | ns ns |
| t_{PW} | Minimum Input Pulse Width | Pins 3, 4 and 5 | 1 | | μs |
| t_{CPR} | Charge Pump Rise Time | No Bootstrap Capacitor | 20 | | μs |

Electrical Characteristics (Continued)

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its rated operating conditions.

Note 2: See Application Information for details regarding current limiting.

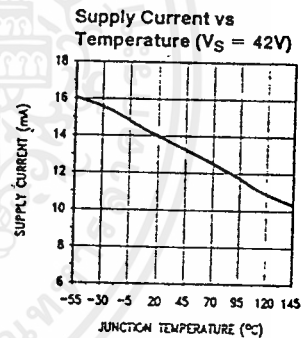
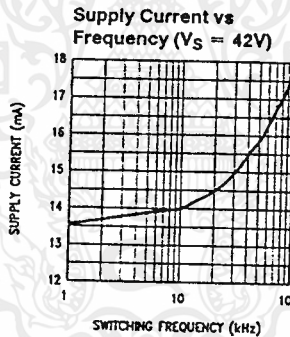
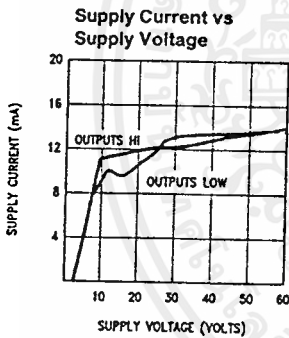
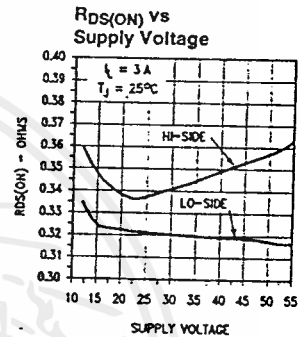
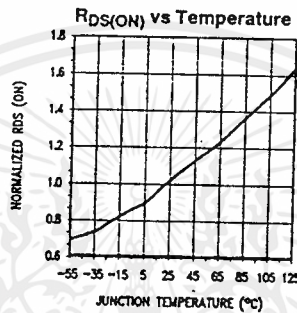
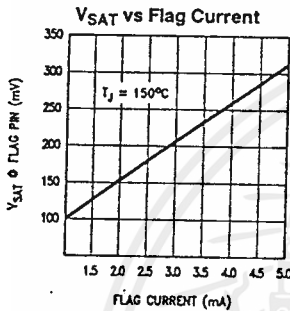
Note 3: The maximum power dissipation must be derated at elevated temperatures and is a function of $T_{J(max)}$, θ_{JA} , and T_A . The maximum allowable power dissipation at any temperature is $P_{D(max)} = (T_{J(max)} - T_A)/\theta_{JA}$, or the number given in the Absolute Ratings, whichever is lower. The typical thermal resistance from junction to case (θ_{JC}) is 1.0°C/W and from junction to ambient (θ_{JA}) is 30°C/W. For guaranteed operation $T_{J(max)} = 125^\circ\text{C}$.

Note 4: Human-body model, 100 pF discharged through a 1.5 kΩ resistor. Except Bootstrap pins (pins 1 and 11) which are protected to 1000V of ESD.

Note 5: All limits are 100% production tested at 25°C. Temperature extreme limits are guaranteed via correlation using accepted SOC (Statistical Quality Control) methods. All limits are used to calculate AOQL (Average Outgoing Quality Level).

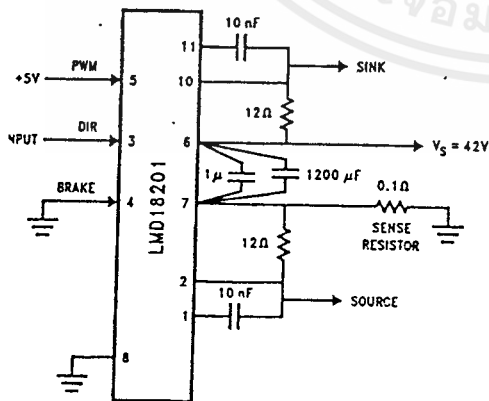
Note 6: Output currents are pulsed ($t_W < 2$ ms, Duty Cycle $< 5\%$).

Typical Performance Characteristics



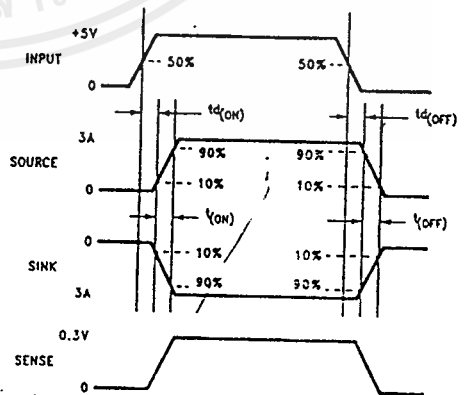
TL/H/10793-3

Test Circuit



TL/H/10793-8

Switching Time Definitions



TL/H/10793-9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pinout Description (See Connection Diagram)

Pin 1, BOOTSTRAP 1 Input: Bootstrap capacitor pin for half H-Bridge number 1. The recommended capacitor (10 nF) is connected between pins 1 and 2.

Pin 2, OUTPUT 1: Half H-Bridge number 1 output.

Pin 3, DIRECTION Input: See Table I. This input controls the direction of current flow between OUTPUT 1 and OUTPUT 2 (pins 2 and 10) and, therefore, the direction of rotation of a motor load.

Pin 4, BRAKE Input: See Table I. This input is used to brake a motor by effectively shorting its terminals. When braking is desired, this input is taken to a logic high level and it is also necessary to apply logic high to PWM input, pin 5. The drivers that short the motor are determined by the logic level at the DIRECTION input (Pin 3): with Pin 3 logic high, both current sourcing output transistors are ON; with Pin 3 logic low, both current sinking output transistors are ON. All output transistors can be turned OFF by applying a logic high to Pin 4 and a logic low to PWM input Pin 5; in this case only a small bias current (approximately -1.5 mA) exists at each output pin.

Pin 5, PWM Input: See Table I. How this input (and DIRECTION input, Pin 3) is used is determined by the format of the PWM Signal.

Pin 6, V_S Power Supply

Pin 7, POWER GROUND/SENSE Connection: This pin is the ground return for the power DMOS transistors of the H-Bridge. The current through the H-Bridge can be sensed by adding a small, 0.1 Ω , sense resistor from this pin to the power supply ground.

Pin 8, SIGNAL GROUND: This is the ground return for the internal logic circuitry used to control the PWM switching of the H-Bridge.

Pin 9, THERMAL FLAG Output: This pin provides the thermal warning flag output signal. Pin 9 becomes active-low at 145°C (junction temperature). However the chip will not shut itself down until 170°C is reached at the junction.

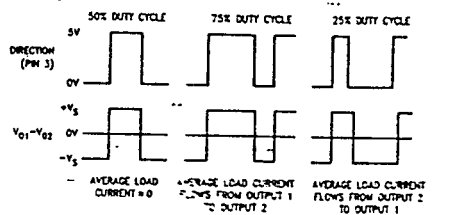
Pin 10, OUTPUT 2: Half H-Bridge number 2 output.

Pin 11, BOOTSTRAP 2 Input: Bootstrap capacitor pin for half H-Bridge number 2. The recommended capacitor (10 nF) is connected between pins 10 and 11.

TABLE I. Logic Truth Table

| PWM | Dir | Brake | Active Output Drivers |
|-----|-----|-------|-----------------------|
| H | H | L | Source 1, Sink 2 |
| H | L | L | Sink 1, Source 2 |
| L | X | L | Source 1, Source 2 |
| H | H | H | Source 1, Source 2 |
| H | L | H | Sink 1, Sink 2 |
| L | X | H | NONE |

Locked Anti-Phase PWM Control



TL/H/10²³-4

Application Information

TYPES OF PWM SIGNALS

The LMD18201 readily interfaces with different forms of PWM signals. Use of the part with two of the more popular forms of PWM is described in the following paragraphs.

Simple, locked anti-phase PWM consists of a single, variable duty-cycle signal in which is encoded both direction and amplitude information. A 50% duty-cycle PWM signal represents zero drive, since the net value of voltage (integrated over one period) delivered to the load is zero. For the LMD18201, the PWM signal drives the direction input (pin 3) and the PWM input (pin 5) is tied to logic high.

Sign/magnitude PWM consists of separate direction (sign) and amplitude (magnitude) signals. The (absolute) magnitude signal is duty-cycle modulated, and the absence of a pulse signal (a continuous logic low level) represents zero drive. Current delivered to the load is proportional to pulse width. For the LMD18201, the DIRECTION input (pin 3) is driven by the sign signal and the PWM input (pin 5) is driven by the magnitude signal.

USING THE THERMAL WARNING FLAG

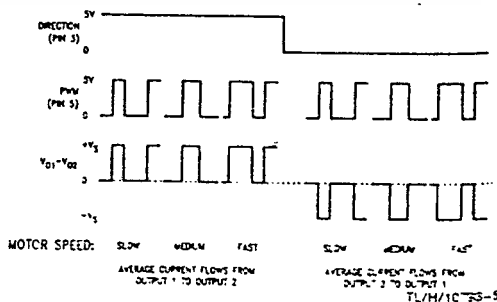
The THERMAL FLAG output (pin 9) is an open collector transistor. This permits a wired OR connection of thermal warning flag outputs from multiple LMD18201's, and allows the user to set the logic high level of the output signal swing to match system requirements. This output typically drives the interrupt input of a system controller. The interrupt service routine would then be designed to take appropriate steps, such as reducing load currents or initiating an orderly system shutdown. The maximum voltage compliance on the flag pin is 12V.

SUPPLY BYPASSING

During switching transitions the levels of fast current changes experienced may cause troublesome voltage transients across system stray inductances.

It is normally necessary to bypass the supply rail with a high quality capacitor(s) connected as close as possible to the V_S Power Supply (Pin 6) and POWER GROUND (Pin 7). A 1 μ F high-frequency ceramic capacitor is recommended. Care should be taken to limit the transients on the supply pin below the Absolute Maximum Rating of the device. When operating the chip at supply voltages above 40V a voltage suppressor (transorb) such as P6KE62A is recommended from supply to ground. Typically the ceramic capacitor can be eliminated in the presence of the voltage suppressor. Note that when driving high load currents a greater amount of supply bypass capacitance (in general at least 100 μ F per Amp of load current) is required to absorb the recirculating currents of the inductive loads.

Sign/Magnitude PWM Control



TL/H/10²³-5

Application Information (Continued)

CURRENT LIMITING

Current limiting protection circuitry has been incorporated into the design of the LMD18201. With any power device it is important to consider the effects of the substantial surge currents through the device that may occur as a result of shorted loads. The protection circuitry monitors the current through the upper transistors and shuts off the power device as quickly as possible in the event of an overload condition (the threshold is set to approximately 10A). In a typical motor driving application the most common overload faults are caused by shorted motor windings and locked rotors. Under these conditions the inductance of the motor (as well as any series inductance in the V_{CC} supply line) serves to reduce the magnitude of a current surge to a safe level for the LMD18201. Once the device is shut down, the control circuitry will periodically try to turn the power device back on. This feature allows the immediate return to normal operation once the fault condition has been removed. While the fault remains however, the device will cycle in and out of thermal shutdown. This can create voltage transients on the V_{CC} supply line and therefore proper supply bypassing techniques are required.

The most severe condition for any power device is a direct, hard-wired ("screwdriver") long term short from an output to ground. This condition can generate a surge of current through the power device on the order of 15 Amps and require the die and package to dissipate up to 500W of power for the short time required for the protection circuitry to shut off the power device. This energy can be destructive, particularly at higher operating voltages (>30V) so some precautions are in order. Proper heat sink design is essential and it is normally necessary to heat sink the V_{CC} supply pin (pin 6) with 1 square inch of copper on the PC board.

INTERNAL CHARGE PUMP AND USE OF BOOTSTRAP CAPACITORS

To turn on the high-side (sourcing) DMOS power devices, the gate of each device must be driven approximately 8V more positive than the supply voltage. To achieve this an internal charge pump is used to provide the gate drive voltage. As shown in *Figure 1*, an internal capacitor is alternately switched to ground and charged to about 14V, then switched to V_S thereby providing a gate drive voltage greater than V_S . This switching action is controlled by a continuously running internal 300 kHz oscillator. The rise time of this drive voltage is typically 20 μ s which is suitable for operating frequencies up to 1 kHz.

For higher switching frequencies, the LMD18201 provides for the use of external bootstrap capacitors. The bootstrap principle is in essence a second charge pump whereby a large value capacitor is used which has enough energy to quickly charge the parasitic gate input capacitance of the power device resulting in much faster rise times. The switching action is accomplished by the power switches themselves (*Figure 2*). External 10 nF capacitors, connected from the outputs to the bootstrap pins of each high-side switch provide typically less than 100 ns rise times allowing switching frequencies up to 500 kHz.

INTERNAL PROTECTION DIODES

A major consideration when switching current through inductive loads is protection of the switching power devices from the large voltage transients that occur. Each of the four switches in the LMD18201 have a built-in protection diode to clamp transient voltages exceeding the positive supply or ground to a safe diode voltage drop across the switch.

The reverse recovery characteristics of these diodes, once the transient has subsided, is important. These diodes must come out of conduction quickly and the power switches must be able to conduct the additional reverse recovery current of the diodes. The reverse recovery time of the diodes protecting the sourcing power devices is typically only 70 ns with a reverse recovery current of 1A when tested with a full 3A of forward current through the diode. For the sinking devices the recovery time is typically 100 ns with 4A of reverse current under the same conditions.

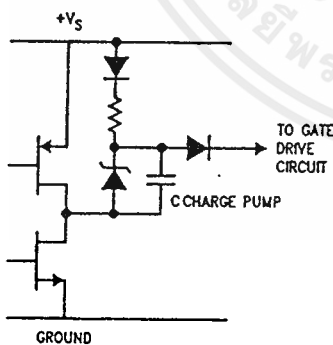


FIGURE 1. Internal Charge Pump Circuitry

TL/H/10793-6

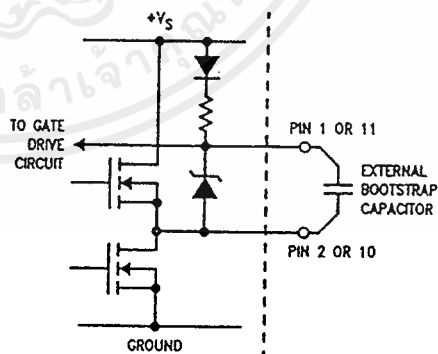


FIGURE 2. Bootstrap Circuitry

TL/H/10793-7

Typical Applications

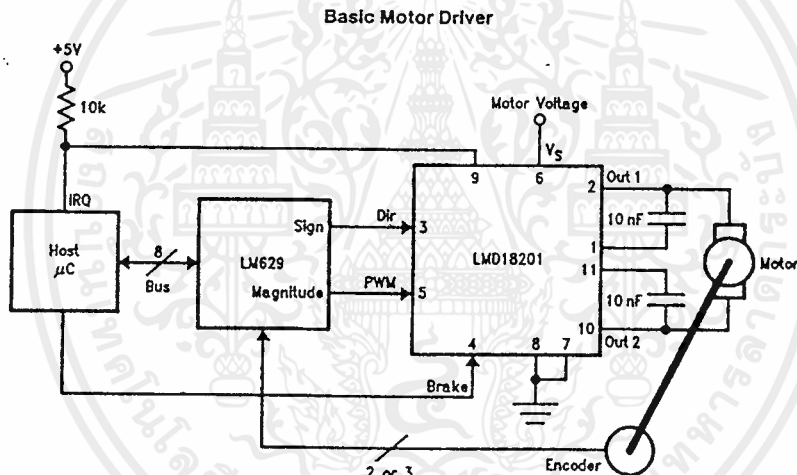
BASIC MOTOR DRIVER

The LMD18201 can directly interface to any Sign/Magnitude PWM controller. The LM629 is a motion control processor that outputs a Sign/Magnitude PWM signal to coordinate either positional or velocity control of DC motors. The LMD18201 provides fully protected motor driver stage.

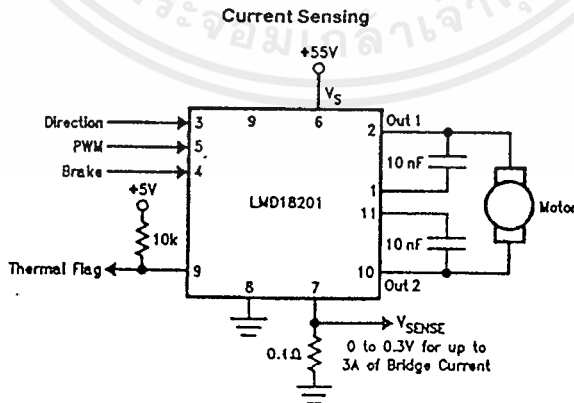
CURRENT SENSING

In many motor control applications it is desirable to sense and control the current through the motor. For these types of applications a companion product, the LMD18200, is also available. The LMD18200 is identical to the LMD18201 but has current sensing transistors that output a current directly proportional to the current conducted by the two upper DMOS power devices to a separate current sense pin. This technique does not require a low valued, power sense resistor and does not subtract from the available voltage drive to the motor.

To sense the bridge current through the LMD18201 requires the addition of a small sense resistor between the power ground/sense pin (Pin 7) and the actual circuit ground. This resistor should have a value of 0.1Ω or less to stay within the allowable voltage compliance of the sense pin, particularly at higher operating current levels. The voltage between power ground/sense (Pin 7) and the signal ground (Pin 8) must stay within the range of $-1V$ to $+0.5V$. Internally there is approximately 25Ω between pins 7 and 8 and this resistance will slightly reduce the value of the external sense resistor. Approximately 70% of the quiescent supply current (10 mA) flows out of pin 7. This will cause a slight offset to the voltage across the sense resistor when the bridge is not conducting. During reverse recovery of the internal protection diodes the voltage compliance between pins 7 and 8 may be exceeded. The duration of these spikes however are only approximately 100 ns and do not have enough time or energy to disrupt the operation of the LMD18201.



TL/H/10793-10



TL/H/10793-11