สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบควบคุมอัตโนมัติขนาดเล็ก 2

MICRO CIM

(Micro Computer Integrated Manufacturing)

โดย

นายปฐมพงษ์ ชัยชนะ

นายสราวุฒิ เพิ่มอยู่เย็น

เลขหมู่.....
เลขทะเบียน 62906
วัน,เดือน,ปี 23 ส.ค. 2549

b 11b 33905
.....
.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมแมคคาทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2548

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมอัตโนมัติขนาดเล็ก
MICRO CIM

ผู้จัดทำ นายปฐมพงษ์ ชัยชนะ 45010432
นายศราวุฒิ เพิ่มอยู่เย็น 45010749

..... อาจารย์ที่ปรึกษา
(อาจารย์เทพจิตร์ เชยโภคา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอัตโนมัติขนาดเล็ก 2

MICRO CIM

(Micro Computer Integrated Manufacturing)

โดย

นายปฐมพงษ์ ชัยชนะ 45010432

นายศราวุฒิ เพิ่มอยู่เย็น 45010749

อาจารย์ที่ปรึกษา

อาจารย์เทพจิตร์ เชยโภคา

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการสร้างและออกแบบ วงจรขับ(Driver Circuit) และส่วนควบคุม การทำงานของแขนกลรุ่น SCORBOT-ER Vplus โดยใช้ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 รุ่น AT89C51RD2 ภาษาที่ใช้ในการเขียนโปรแกรมคือภาษาซี ผู้ใช้สามารถสั่งให้แขนกลเคลื่อนที่ไปยังตำแหน่งต่างๆ ที่ต้องการได้ โดยการกดปุ่มที่แสดงบนหน้าจอคอมพิวเตอร์สามารถทำงานได้ 2 แบบ คือ แบบแมนนวล(manual) และแบบอัตโนมัติ(automatic) การทำงานแบบอัตโนมัตินั้น จะทำงานโดยการกำหนดตำแหน่งเพื่อให้แขนกลทำงาน โดยการกำหนดตำแหน่งต่างๆ โดยแมนนวลแล้วเก็บค่าตำแหน่งที่ได้ไว้ เมื่อต้องการให้แขนกลทำงานแบบอัตโนมัติก็เพียงแค่กดปุ่ม auto สั่งให้แขนกลทำงานตามที่ได้เก็บค่าไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมอัตโนมัติขนาดเล็ก 2

MICRO CIM

(Micro Computer Integrated Manufacturing)

โดย

นายปฐมพงษ์ ชัยชนะ 45010432

นายศราวุฒิ เพิ่มอยู่เย็น 45010749

อาจารย์ที่ปรึกษา

อาจารย์เทพจิตร์ เชยโภคา

ABSTRACT

This project is design to create driver and controller for robot SCORBOT-ER Vplus by Microcontroller MCS-51 No. AT89C51RD2. The language which is used in programming for controlling the robotic arm is C computer language. User can interface robot with computer. It can be divided into 2 mode. In automatic mode the robotic arm will move to final position by the value. That see by manual and keep that value in computer and then press auto mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความเมตตาเป็นอย่างสูง จากอาจารย์เทพจิตร์ เชยโภคา อาจารย์ที่ปรึกษาที่ได้ให้คำแนะนำ ข้อคิดเห็น ให้ความช่วยเหลือแก่คณะผู้จัดทำ

ขอขอบพระคุณคณาจารย์ทุกท่านที่ได้ให้คำแนะนำ และได้ประสิทธิประสาทวิชาความรู้แก่คณะผู้จัดทำ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณ บิดา มารดา ของคณะผู้จัดทำ ที่ได้ให้ความสนับสนุนให้โอกาสผู้จัดทำได้เล่าเรียนจนถึงทุกวันนี้ และเป็นกำลังใจแก่ผู้จัดทำตลอดมา อีกทั้งขอขอบคุณเพื่อนๆ ที่คอยให้ความช่วยเหลือและเป็นกำลังใจแก่ผู้จัดทำ จนทำให้ปริญญานิพนธ์ฉบับนี้เสร็จสิ้นสมบูรณ์

คณะผู้จัดทำ
นายปฐมพงษ์ ชัยชนะ
นายศราวุฒิ เพิ่มอยู่เย็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญภาพ	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 แขนกล	2
2.2 มอเตอร์ไฟฟ้ากระแสตรง	10
2.3 เอนโคดเดอร์	17
2.4 ไมโครสวิตซ์	20
2.5 MCS-51	21
2.6 พอร์ตขนาน	32
บทที่ 3 โครงสร้างทางฮาร์ดแวร์	36
3.1 วงจร MCS-51	36
3.2 วงจรขับมอเตอร์	37
3.3 วงจรเอนโคดเดอร์	38
3.4 วงจรปรับความเร็ว	39
3.5 วงจรปรับแต่งสัญญาณ	40
บทที่ 4 โครงสร้างการทำงาน	41
4.1 โครงสร้างของแขนกล	41
4.2 Flow chart แสดงการทำงานของโปรแกรม	47
บทที่ 5 สรุปและข้อเสนอแนะ	49
5.1 สรุป	49
5.2 ข้อเสนอแนะ	50
5.3 ปัญหาที่พบ	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

หน้า

ภาคผนวก A

ภาคผนวก B

เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

รูปที่	หน้า
2.1 ส่วนต่าง ๆ ของ หุ่นยนต์เปรียบเทียบกับสรีระของมนุษย์	3
2.2 Gantry Robot	5
2.3 Work envelope of Gantry Robot	5
2.4 Cylindrical Robot	6
2.5 Work envelope of Cylindrical Robot	6
2.6 Spherical Robot	7
2.7 Work envelope of Spherical Robot	7
2.8 SCARA Robot	8
2.9 Work envelope of SCARA Robot	8
2.10 Articulated Robot	9
2.11 Work envelope of Articulated Robot	9
2.12 มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร	10
2.13 มอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนาม	10
2.14 วงจรสมมูลมอเตอร์ไฟฟ้ากระแสตรง	12
2.15 การทำงานของมอเตอร์ไฟฟ้ากระแสตรงทั้ง 4 ย่านการทำงาน	13
2.16 ดีซีมอเตอร์แบบอาร์เมเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก	14
2.17 คุณสมบัติระหว่างแรงบิดกับความเร็วของดีซีมอเตอร์อนุกรมภายใต้ภาวะโวลท์เตจคงที่	14
2.18 ดีซีมอเตอร์แบบแยกปรับสนามแม่เหล็กได้	15
2.19 คุณสมบัติระหว่างแรงบิดกับความเร็วของชันท์มอเตอร์อนุกรมภายใต้ภาวการณ์กระตุ้นสนามแม่เหล็กคงที่และอาร์เมเจอร์โวลต์เตจคงที่	15
2.20 ดีซีมอเตอร์แบบฟิลด์เป็นแบบแม่เหล็กถาวร	15
2.21 ระบบเอนโคดเดอร์แบบโรตารี่	18
2.22 เอนโคดเดอร์ที่นับการเพิ่มค่าแบบเส้นตรง	19
2.23 ตัวอย่างกลไกของเอนโคดเดอร์แบบออพติคอลอินครีเมนท์	19
2.24 ตัวอย่างตัวจับสัญญาณแบบมีช่องปิดเปิดให้แสงผ่านได้ช่องเดียวและแบบมีหลายช่อง	19
2.25(a) ตัวอย่างลูกคลื่นเอาท์พุตสี่เหลี่ยมของอุปกรณ์เอนโคดเดอร์ช่องเดียว	20
2.25(b) สัญญาณเอนโคดเดอร์ 2 ช่อง ที่มีมุมเฟสต่างกัน 90 องศา	20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ(ต่อ)

รูปที่	หน้า
2.26 ไมโครสวิตซ์	21
2.27 บล็อกไดอะแกรมของ MCS-51	22
2.28 การจัดขาของ MCS-51	23
2.29 พื้นที่บริเวณหน่วยความจำข้อมูลในตำแหน่งที่ 30H-7FH	27
2.30 การจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ (Special Function Register)	28
2.31 พอร์ตขนาน	32
2.32 การส่งข้อมูลของพอร์ตขนาน	32
2.33 Lay out	35
3.1 วงจร MCS-51	36
3.2 โครงสร้างภายใน TA7257P	37
3.3 การเชื่อมต่อ IC เบอร์ TA7257P กับอุปกรณ์พื้นฐานภายนอก	38
3.4 เอนโคดเดอร์	38
3.5 วงจรเอนโคดเดอร์	39
3.6 วงจรปรับความเร็ว	39
3.7 วงจรปรับแต่งสัญญาณ	40
4.1 SCORBOT-ER V plus	41
4.2 ชิ้นส่วนแขนกล	41
4.3 ทิศทางการเคลื่อนที่ของโครงสร้างแขนกล	42
4.4 ขอบเขตพื้นที่การทำงานของแขนกล	43
4.5 เอนโคดเดอร์และสายไฟเชื่อมต่อ	43
4.6 flow chart การทำงานของ MCS1	47
4.7 flow chart การทำงานของ MCS2	48
5.1 ปุ่มควบคุมบนหน้าจอคอมพิวเตอร์	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 การทำงานของจุดต่อ (Joint) ของหุ่นยนต์ อุตสาหกรรมในขั้นพื้นฐาน	4
2.2 จุดต่อของหุ่นยนต์แต่ละชนิด	4
2.3 หน่วยความจำสำหรับเก็บโปรแกรมของไอซีไมโครคอนโทรลเลอร์	24
2.4 ตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน(Internal data memory)	25
2.5 แฟลกต่าง ๆ ของรีจิสเตอร์ PSW	29
2.6 Carry Flag	29
2.7 ขาของพอร์ตขนาน	33
2.8 Register Address	34
3.1 แสดงอินพุตและเอาท์พุตของ IC เบอร์ TA7257P	37
4.1 ทิศทางการเคลื่อนที่ของโครงสร้างแขนกลแต่ละข้อต่อและตำแหน่งมอเตอร์	42
4.2 การเชื่อมต่อสายไฟจากมอเตอร์, เอนโคดเตอร์และไมโครสวิตช์ออกภายนอก	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

โครงงานนี้ได้แสดงการทำงานร่วมกันของอุปกรณ์ต่างๆ คือ แขนกล วงจรควบคุมการทำงาน โปรแกรมการควบคุมการทำงานด้วยภาษาซี และส่วนติดต่อกับผู้ใช้ ซึ่งสามารถสั่งงานผ่านทางคอมพิวเตอร์ได้ โดยการทำงานทั้งหมดเริ่มจาก กำหนดค่าตำแหน่งเริ่มต้นและตำแหน่งสุดท้าย ซึ่งก็คือ ตำแหน่งหยิบจับชิ้นงานและวางชิ้นงาน ตามลำดับ ซึ่งแขนกลจะสามารถทำงานได้อัตโนมัติ โดยจะเคลื่อนที่จากตำแหน่งอ้างอิงไปยังตำแหน่งเริ่มต้นที่กำหนด และกลับไปยังจุดอ้างอิง จากนั้นเคลื่อนไปตำแหน่งสุดท้ายและกลับไปที่จุดอ้างอิงอีกครั้ง เป็นอันสิ้นสุด 1 รอบการทำงาน

วัตถุประสงค์

เพื่อควบคุมกระบวนการทำงานของแขนกลโดยจะทำงานร่วมกันเป็นขั้นตอน ผ่านทางคอมพิวเตอร์เพื่อให้เป็นการทำงานอัตโนมัติขนาดเล็ก

ขอบเขต

ศึกษาทฤษฎีและหลักการทำงานโดยทั่วไปของแขนกล วงจรควบคุม โปรแกรมที่ใช้ควบคุมการทำงานของแขนกล ให้ทำงานร่วมกันได้อย่างเป็นระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 แขนกล

ในยุคเริ่มต้นของการส่งเสริมอุตสาหกรรมในประเทศไทยจะเห็นได้ว่ามีโรงงานต่าง ๆ เข้ามาตั้งฐานผลิตในเมืองไทยจำนวนมากทำให้เกิดนิคมอุตสาหกรรมขึ้นหลายแห่ง ทั้งนี้เนื่องจากรัฐบาลมีนโยบายส่งเสริมอุตสาหกรรมที่ชัดเจน ค่าแรงงานถูก ลดรายจ่ายเนื่องจากภาษีการนำเข้าของสินค้า และวัตถุดิบบางตัว แต่ ณ ปัจจุบันนี้ค่าแรงบ้านเราสูงขึ้นและสูงกว่าประเทศเพื่อนบ้าน เช่น จีน เวียดนาม อินโดนีเซีย ฯลฯ ในขณะเดียวกันคุณภาพแรงงานไม่ได้มาตรฐาน ขาดความรู้และทักษะจึงทำให้หลายบริษัทได้ย้ายฐานการผลิตไปยังประเทศที่มีค่าแรงงานถูกกว่า และอีกหลายบริษัทที่พยายามปรับตัวเอง โดยมีการนำเทคโนโลยีระบบอัตโนมัติ (Automation Technology) เข้ามาใช้งานเพื่อให้สินค้าสามารถแข่งขันในตลาดโลกได้ทั้งในเรื่องราคา และคุณภาพ โดยเฉพาะในเรื่องคุณภาพ ปัจจุบันโรงงานที่ผลิตสินค้าส่งออกหรือส่งให้กับลูกค้าที่เป็นบริษัทของต่างประเทศมักจะประสบปัญหาในเรื่องคุณภาพ มีทั้ง ผลิตสินค้าไม่ได้มาตรฐานตามที่ลูกค้ากำหนด หรือ ผลิตสินค้าไม่ทันตามกำหนดเวลา อาจเนื่องจากมีการเปลี่ยนรุ่นผลิตภัณฑ์อยู่บ่อยๆ ต้องใช้เวลาในการ Set up ปัจจุบันจึงมีการนำเทคโนโลยีต่าง ๆ เข้ามาใช้ หนึ่งในเทคโนโลยีที่มีความยืดหยุ่นสูง ได้แก่ หุ่นยนต์อุตสาหกรรม เนื่องจากการเปลี่ยนการทำงานสามารถทำได้โดยการเปลี่ยนโปรแกรม นอกจากนี้คุณภาพของผลิตภัณฑ์ที่ได้มีความสม่ำเสมอเป็นมาตรฐานเดียวกัน

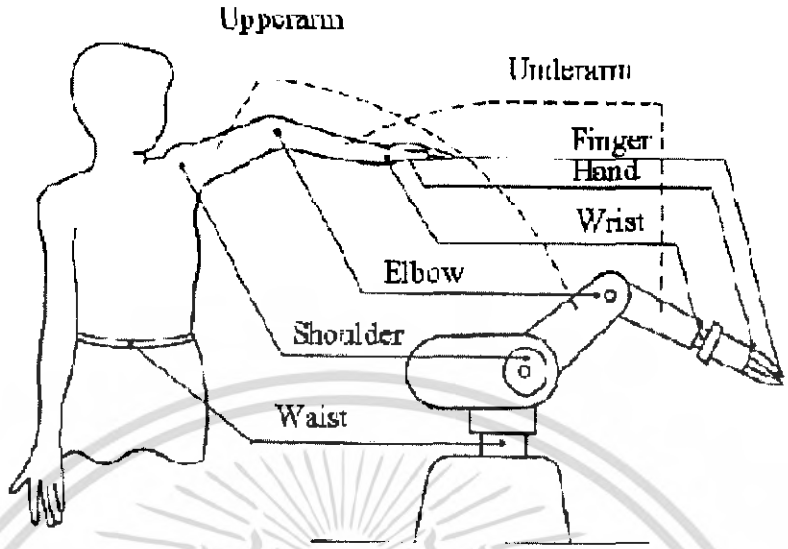
หุ่นยนต์ (Robotics) มีมากมายหลายชนิดขึ้นอยู่กับจุดประสงค์ของการใช้งาน เช่น หุ่นยนต์ที่เป็นของเล่น หุ่นยนต์ใช้ในงานการแพทย์ หุ่นยนต์เพื่องานวิจัย หุ่นยนต์สำรวจ และหุ่นยนต์อุตสาหกรรม ฯลฯ

ในที่นี้จะกล่าวถึงหุ่นยนต์อุตสาหกรรม (Industrial Robot)

การทำงานของหุ่นยนต์อุตสาหกรรมจะเลียนแบบร่างกายของมนุษย์ โดยจะเลียนแบบเฉพาะส่วนของร่างกายที่จะนำไปใช้ประโยชน์ ในอุตสาหกรรมเท่านั้น นั่นคือช่วงแขนของมนุษย์ ดังนั้น บางคนอาจจะได้ยินคำว่าแขนกล ซึ่งก็หมายถึงหุ่นยนต์อุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของหุ่นยนต์อุตสาหกรรมเปรียบเทียบกับแขนมนุษย์ แสดงดังรูปที่ 2.1



รูปที่ 2.1 ส่วนต่าง ๆ ของ หุ่นยนต์เปรียบเทียบกับสรีระของมนุษย์

ปัจจุบันและในอนาคตหุ่นยนต์อุตสาหกรรมจะเข้ามามีบทบาทในอุตสาหกรรมมากขึ้น โดยจะทำงานแทนมนุษย์ในงานต่าง ๆ เหล่านี้งานที่อันตราย เช่น งานยกเหล็กเข้าเตาหลอม งานที่เกี่ยวข้องกับสารเคมีงานซ้ำซากน่าเบื่อ เช่น งานยกสินค้าจากสายงานการผลิต งานประกอบ งานบรรจุผลิตภัณฑ์งานที่ต้องการคุณภาพมาตรฐานเดียวกัน เช่น งานเชื่อม งานตัด งานที่ต้องใช้ทักษะความชำนาญสูง เช่น งานเชื่อมแนว เชื่อมเลเซอร์งานที่ต้องใช้ความละเอียดประณีต เช่น งานประกอบชิ้นส่วนอิเล็กทรอนิกส์ งานตรวจสอบ (Inspection) ฯลฯ

นิยามของหุ่นยนต์ (Robotics definition)

หุ่นยนต์ หรือ ภาษาอังกฤษเขียนว่า Robot มาจากบทละครของ นายคาเรล คาเปก (Karel Kapek) นักแต่งนิยายชาว เช็ก เรื่อง R.U.R (Rossum's Universal Robots) ซึ่งหมายถึงคนงานคำจำกัดความของ หุ่นยนต์ตามมาตรฐาน ISO 8373 An automatically controlled, reprogrammable, multipurpose, manipulator programmable in three or more axes which may be either fixed in place or mobile for use in industrial automation application หุ่นยนต์ คือ เครื่องจักรที่ถูกควบคุมอัตโนมัติ สามารถเขียนโปรแกรมใหม่ได้ ใช้งานเอนกประสงค์ โปรแกรมการเคลื่อนที่จะต้องสามารถโปรแกรมให้เคลื่อนที่ได้อย่างน้อย 3 แกนหรือมากกว่า หุ่นยนต์อาจจะยึดอยู่กับที่หรือย้ายตำแหน่ง (Mobile) เพื่อใช้ในงานอุตสาหกรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแบ่งชนิดของหุ่นยนต์

โดยทั่วไปการแบ่งชนิดของหุ่นยนต์ จะแบ่งตามลักษณะรูปทรงของพื้นที่ทำงาน (Envelope Geometric) แต่ก่อนจะอธิบายชนิดของหุ่นยนต์ขออธิบายการทำงานของจุดต่อ (Joint) ของหุ่นยนต์ อุตสาหกรรมในขั้นพื้นฐานมี 2 ชนิดด้วยกัน ดังนี้

ตารางที่ 2.1 การทำงานของจุดต่อ (Joint) ของหุ่นยนต์ อุตสาหกรรมในขั้นพื้นฐาน

ชนิด	สัญลักษณ์	หมายเหตุ
Revolute (R)		เป็นการหมุนรอบแกน (Rotary)
Prismatic (P)		การเคลื่อนที่เชิงเส้น (Linear motion)

จุดต่อ (Joint) ทั้งสองแบบเมื่อนำมาต่อเข้าด้วยกันอย่างน้อย 3 แกนหลักจะได้พื้นที่ทำงาน (Work envelope) ที่มีลักษณะแตกต่างกันไป ซึ่งสามารถนำมาแบ่งชนิดของหุ่นยนต์ได้ดังต่อไปนี้

ตารางที่ 2.2 จุดต่อของหุ่นยนต์แต่ละชนิด

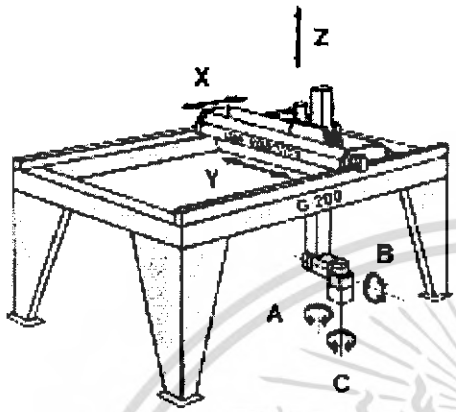
ชนิดของหุ่นยนต์	แกนที่ 1 (เอว)	แกนที่ 1 (เอว)	แกนที่ 3 (ข้อศอก)
Cartesian (gantry)	P	P	P
Cylindrical	R	P	P
Spherical (Polar)	R	R	P
SCARA	R	P	R
Articulated	R	R	R

R = Revolute, P = Prismatic

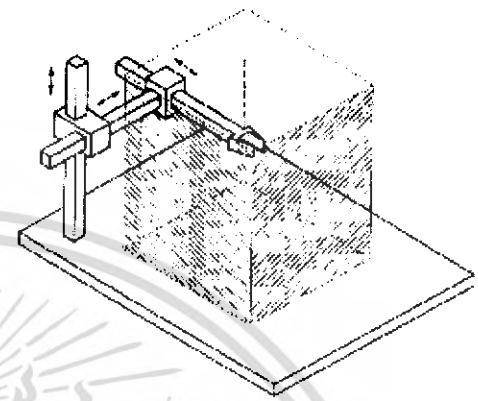
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Cartesian (gantry) Robot

แกนทั้ง 3 ของหุ่นยนต์จะเคลื่อนที่เป็นแบบเชิงเส้น (Prismatic) ถ้าโครงสร้างมีลักษณะคล้าย Overhead Crane จะเรียกว่าเป็นหุ่นยนต์ชนิด gantry แต่ถ้าหุ่นยนต์ไม่มีขาตั้งหรือขาเป็นแบบอื่น เรียกว่า ชนิด Cartesian



รูปที่ 2.2 Gantry Robot



รูปที่ 2.3 Work envelope of Gantry Robot

ข้อดี

- เคลื่อนที่เป็นแนวเส้นตรงทั้ง 3 มิติ
- การเคลื่อนที่สามารถทำความเข้าใจง่าย
- มีส่วนประกอบง่าย ๆ
- โครงสร้างแข็งแรงตลอดการเคลื่อนที่

ข้อเสีย

- ต้องการพื้นที่ติดตั้งมาก
- บริเวณที่หุ่นยนต์เข้าไปทำงานได้ จะเล็กกว่าขนาดของตัวหุ่นยนต์
- ไม่สามารถเข้าถึงวัตถุจากทิศทางข้างใต้ได้
- แกนแบบเชิงเส้นจะ Seal เพื่อป้องกันฝุ่นและของเหลวได้ยาก

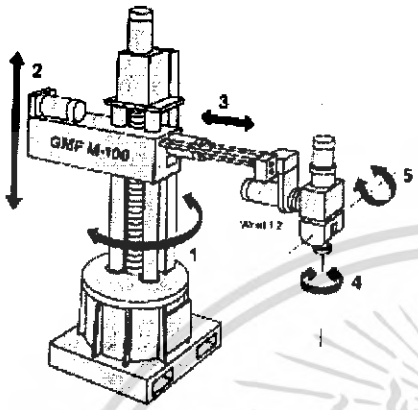
- การประยุกต์ใช้งาน

เนื่องจากโครงสร้างมีความแข็งแรงตลอดแนวการเคลื่อนที่ ดังนั้นจึงเหมาะกับงานเคลื่อนย้ายของหนัก ๆ หรือเรียกว่างาน Pick-and-Place เช่นใช้โหลดชิ้นงานเข้าเครื่องจักร (Machine loading) ,ใช้จัดเก็บชิ้นงาน (Stacking)นอกจากนี้ยังสามารถใช้ในงานประกอบ (Assembly) ที่ไม่ต้องการเข้าถึงในลักษณะที่มีมุมหมุน เช่น ประกอบอุปกรณ์อิเล็กทรอนิกส์ และงานตรวจสอบต่าง ๆ

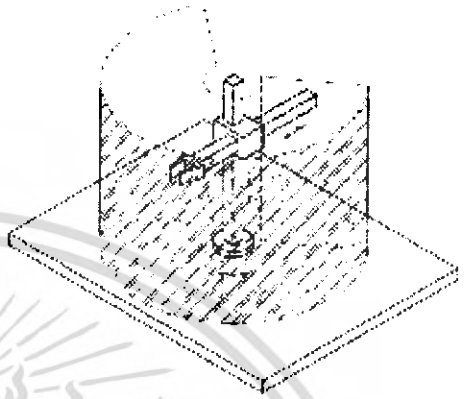
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. Cylindrical

หุ่นยนต์ประเภทนี้จะมีแกนที่ 2 (ไหล่) และแกนที่ 3 (ข้อศอก) เป็นแบบ prismatic ส่วนแกนที่ 1 (เอว)จะเป็นแบบหมุน(revolute) ทำให้การเคลื่อนที่ได้พื้นที่การทำงานเป็นรูปทรงกระบอก ดังรูป



รูปที่ 2.4 Cylindrical Robot



รูปที่ 2.5 Work envelope of Cylindrical Robot

ข้อดี

- มีส่วนประกอบไม่ซับซ้อน
- การเคลื่อนที่สามารถเข้าใจได้ง่าย
- สามารถเข้าถึงเครื่องจักรที่มีการเปิด-ปิด หรือเข้าไปในบริเวณที่เป็นช่องหรือโพรง ได้ง่าย (Loading) เช่น การโหลดชิ้นงาน เข้าเครื่อง CNC

ข้อเสีย

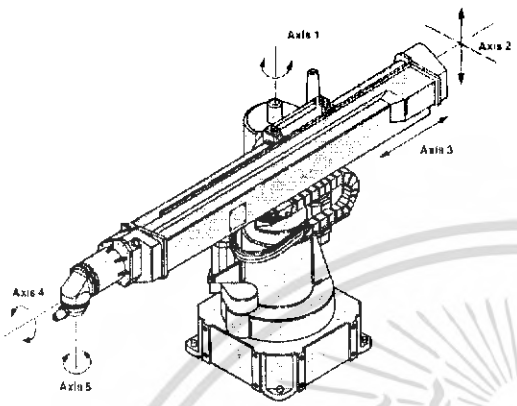
- มีพื้นที่ทำงานจำกัด
- แกนที่เป็นเชิงเส้นมีความยุ่งยากในการ seal เพื่อป้องกันฝุ่นและของเหลว

- **การประยุกต์ใช้งาน** โดยทั่วไปจะใช้ในการหยิบยกชิ้นงาน (Pick-and-Place) หรือป้อนชิ้นงานเข้าเครื่องจักร เพราะสามารถเคลื่อนที่เข้าออกบริเวณที่เป็นช่องโพรงเล็ก ๆ ได้สะดวก

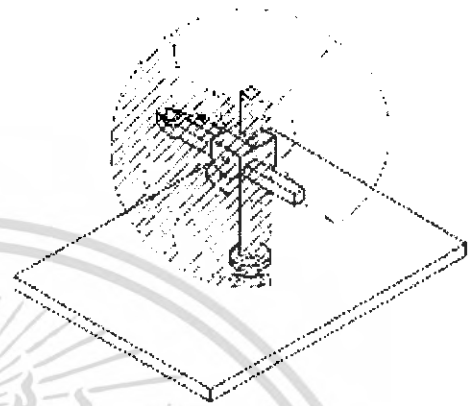
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Spherical Robot (Polar)

มีสองแกนที่เคลื่อนในลักษณะการหมุน (Revolute Joint) คือแกนที่ 1 (เอว) และแกนที่ 2 (ไหล่) ส่วนแกนที่ 3 (ข้อศอก) จะเป็นลักษณะของการเคลื่อนที่แนวเส้นตรง ดังรูปที่ 2.6 ซึ่งทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.7



รูปที่ 2.6 Spherical Robot



รูปที่ 2.7 Work envelope of Spherical Robot

ข้อดี

- มีปริมาตรการทำงานมากขึ้นเนื่องจากการหมุนของแกนที่ 2 (ไหล่)
- สามารถที่จะก้มลงมาจับชิ้นงานบนพื้นได้สะดวก

ข้อเสีย

- มีระบบพิกัด (Coordinate) และส่วนประกอบ ที่ซับซ้อน
- การเคลื่อนที่และระบบควบคุมมีความซับซ้อนขึ้น

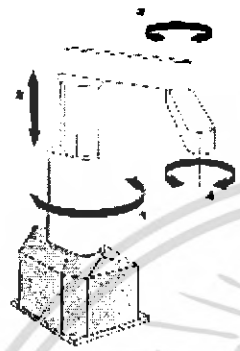
- **การประยุกต์ใช้งาน**

ใช้ในงานที่มีการเคลื่อนที่ในแนวตั้ง (Vertical) เพียงเล็กน้อย เช่น การโหลดชิ้นงานเข้าออกจากเครื่องปั้ม (Press) หรืออาจจะใช้งานเชื่อมจุด (Spot Welding)

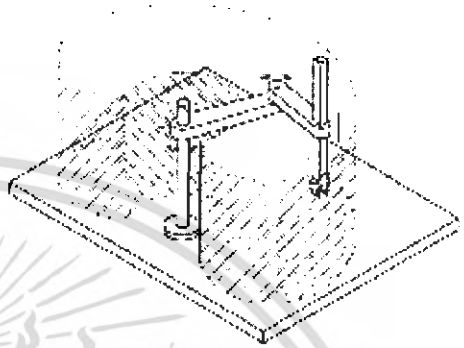
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. SCARA Robot

หุ่นยนต์ SCARA (Selective Compliance Assembly Robot Arm) จะมีลักษณะแกนที่ 1 (เอว) และแกนที่ 3 (ข้อศอก) หมุนรอบแกนแนวตั้ง และแกนที่ 2 จะเป็นลักษณะการเคลื่อนที่ขึ้นลง (Prismatic) ดังรูปที่ 2.8 ทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.9 หุ่นยนต์ SCARA จะเคลื่อนที่ได้รวดเร็วในแนวระนาบ และมีความแม่นยำสูง



รูปที่ 2.8 SCARA Robot



รูปที่ 2.9 Work envelope of SCARA Robot

ข้อดี

- สามารถเคลื่อนที่ในแนวระนาบ และขึ้นลงได้รวดเร็ว
- มีความแม่นยำสูง

ข้อเสีย

- มีพื้นที่ทำงานจำกัด
- ไม่สามารถหมุน (rotation)ในลักษณะมุมต่างๆได้
- สามารถยกน้ำหนัก (Payload) ได้ไม่มากนัก

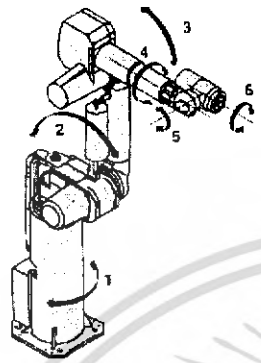
การประยุกต์ใช้งาน

เนื่องจากการเคลื่อนที่ในแนวระนาบและขึ้นลงได้รวดเร็วจึงเหมาะกับ งานประกอบชิ้นส่วนทางอิเล็กทรอนิกส์ซึ่ง ต้องการความรวดเร็วและการเคลื่อนที่ที่ไม่ต้องการ การหมุนมากนัก แต่จะไม่เหมาะกับงานประกอบชิ้นส่วนทางกล (Mechanical part)ซึ่งส่วนใหญ่การประกอบจะอาศัยการหมุน(rotation)ในลักษณะมุมต่างๆ นอกจากนี้ SCARA robot ยังเหมาะกับงานตรวจสอบ (Inspection) งานบรรจุภัณฑ์ (Packaging)

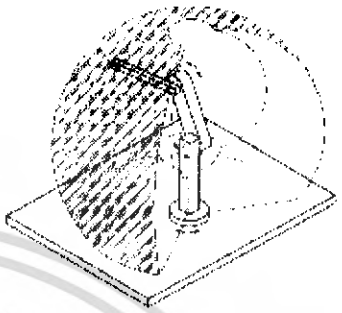
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Articulated Arm (Revolute)

ทุกแกนการเคลื่อนที่จะเป็นแบบหมุน (Revolute) รูปแบบการเคลื่อนที่จะคล้ายกับแขนคน ซึ่งจะประกอบด้วยช่วงเอว ท่อนแขนบน ท่อนแขนล่าง ข้อมือ การเคลื่อนที่ทำให้ได้พื้นที่การทำงาน ดังรูปที่ 2.11



รูปที่ 2.10 Articulated Robot



รูปที่ 2.11 Work envelope of Articulated Robot

ข้อดี

- เนื่องจากทุกแกนจะเคลื่อนที่ในลักษณะของการหมุนทำให้มีความยืดหยุ่นสูงในการเข้าไปยังจุด ต่าง ๆ
- บริเวณข้อต่อ (Joint) สามารถ Seal เพื่อป้องกันฝุ่น ความชื้น หรือน้ำ ได้ง่าย
- มีพื้นที่การทำงานมาก
- สามารถเข้าถึงชิ้นงานทั้งจากด้านบนด้านล่าง
- เหมาะกับการใช้มอเตอร์ไฟฟ้า เป็นชุดขับเคลื่อน

ข้อเสีย

- มีระบบพิกัด (Coordinate) ที่ซับซ้อน
- การเคลื่อนที่และระบบควบคุมทำความ เข้าใจได้ยากขึ้น
- ควบคุมให้เคลื่อนที่ในแนวเส้นตรง (Linear) ได้ยาก
- โครงสร้างไม่มั่นคงตลอดช่วงการเคลื่อนที่ เพราะบริเวณขอบ Work envelope ปลายแขนจะมีการสั่น ทำให้ความแม่ยำลดลง

การประยุกต์ใช้งาน

หุ่นยนต์ชนิดนี้สามารถใช้งานได้กว้างขวางเพราะสามารถเข้าถึงตำแหน่งต่าง ๆ ได้ดี เช่นงานเชื่อม Spot Welding, Path Welding ,งานยกของ , งานตัด ,งานทากาว ,งานที่มีการเคลื่อนที่ยาก ๆ เช่น งานพ่นสี งาน sealing ฯลฯ

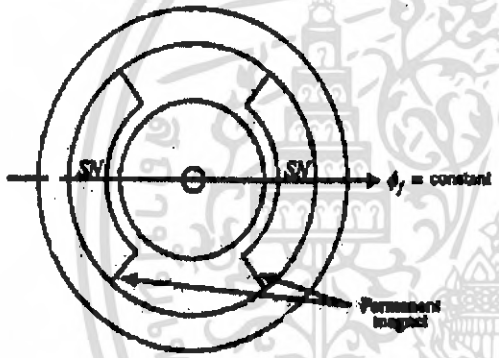
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 มอเตอร์ไฟฟ้ากระแสตรง (DC Motor)

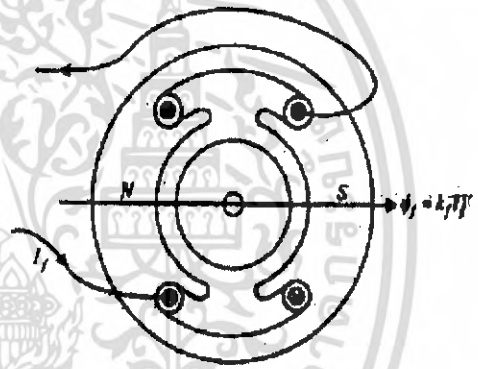
DC Motor โดยพื้นฐานจะประกอบด้วยองค์ประกอบหลัก 2 ส่วนด้วยกัน ได้แก่ ชุดขดลวด (Filed Winding) ซึ่งโดยทั่วไปมักจะพันอยู่บนโครงหรือสเตเดอร์ของมอเตอร์ (Frame or Stator) และชุดขดลวดอาร์เมเจอร์ (Armature Winding) ซึ่งโดยทั่วไปมักจะพันอยู่บนโรเตอร์ (Rotor) โดยที่ขดลวดทั้งสองชุดจะถูกป้อนด้วยไฟฟ้ากระแสตรง ชุดขดลวดอาร์เมเจอร์จะต่ออยู่กับตัวคอมมิวเตเตอร์ (Commutator) ซึ่งติดอยู่ที่ตัวโรเตอร์ หน้าที่ของตัวคอมมิวเตเตอร์จะเหมือนกับวงจรเรียงกระแส เพื่อให้กระแสอาเมเจอร์ไหลผ่านแปลงถ่าน (Brush) เพียงทิศทางเดียวไม่ว่าตัวโรเตอร์จะหมุนที่ความเร็วเท่าใดก็ตาม

วงจรสมมูลของมอเตอร์ไฟฟ้ากระแสตรง

สำหรับวงจรมอเตอร์ไฟฟ้ากระแสตรงสเตเตอร์ จะสร้างเส้นแรงแม่เหล็ก (Filed Flux) ไม่ว่าจะเป็นกรณีใช้แม่เหล็กถาวร (Permanent Magnet) ดังรูป



รูปที่ 2.12 มอเตอร์ไฟฟ้ากระแสตรงแบบแม่เหล็กถาวร



รูปที่ 2.13 มอเตอร์ไฟฟ้ากระแสตรงแบบขดลวดสนาม

ซึ่งเส้นแรงแม่เหล็กที่เกิดจากแม่เหล็กถาวรจะคงที่สม่ำเสมอ หรือเกิดจากขดลวดสนาม (Filed Winding) โดยที่กระแสสนาม (Filed Current) จะเป็นตัวควบคุมเส้นแรงแม่เหล็ก ถ้ากำหนดให้สเตเตอร์ที่ทำหน้าที่ในการสร้างเส้นแรงแม่เหล็กไม่เกิดการอิ่มตัวจะได้

$$\phi_f = K_f I_f \qquad (2.1)$$

เมื่อ ϕ_f คือ เส้นแรงแม่เหล็ก

K_f คือ ค่าคงที่เส้นแรงแม่เหล็ก

I_f คือ กระแสสนาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับโรเตอร์ จะมีขดลวดพันรอบตัวโรเตอร์ดังกล่าวซึ่งจะเรียกขดลวดดังกล่าวว่า ขดลวดอาร์เมเจอร์ ซึ่งจะทำหน้าที่ในการรับพลังงานไฟฟ้าเข้าสู่มอเตอร์ไฟฟ้ากระแสตรงโดยผ่านทางขดลวดอาร์เมเจอร์ ในขณะที่มอเตอร์ไฟฟ้ากระแสสลับนั้น ส่วนที่ทำหน้าที่ในการรับพลังงานไฟฟ้าจะกลายเป็นสเตเตอร์แทน และด้วยความแตกต่างดังกล่าว เป็นผลทำให้มอเตอร์ไฟฟ้ากระแสสลับสามารถใช้กับงานที่ต้องการกำลังสูงๆ ได้ แรงบิดของมอเตอร์ไฟฟ้ากระแสตรงจะเกิดจากความสัมพันธ์ คือ

$$T_{cm} = K_t \phi_f i_a \qquad (2.2)$$

เมื่อ T_{cm} คือ แรงบิดของมอเตอร์ไฟฟ้า

K_t คือ ค่าคงที่แรงบิดของมอเตอร์ไฟฟ้ากระแสตรง

ϕ_f คือ เส้นแรงแม่เหล็ก

i_a คือ กระแสอาร์เมเจอร์

ค่าแรงเคลื่อนเหนี่ยวนำย้อนกลับ (Back-emf) ในส่วนอาร์เมเจอร์จะเกิดจากการหมุนตัวนำของอาร์เมเจอร์ด้วยความเร็ว ω_m ตัดผ่านเส้นแรงแม่เหล็กดังสมการ

$$e_a = K_c \phi_f \omega_m \qquad (2.3)$$

เมื่อ e_a คือ ค่าแรงเคลื่อนเหนี่ยวนำย้อนกลับ

K_c คือ ค่าคงที่แรงดันไฟฟ้าของมอเตอร์ไฟฟ้ากระแสตรง

ϕ_f คือ เส้นแรงแม่เหล็ก

ω_m คือ ความเร็วตัวนำของอาร์เมเจอร์

ในทางปฏิบัติจะมีการควบคุมแหล่งจ่ายแรงดันไฟฟ้าป้อนที่ขั้วของอาร์เมเจอร์ โดยค่ากระแสอาร์เมเจอร์สามารถหาได้จากค่าของแหล่งจ่ายแรงดันไฟฟ้า ค่าแรงเคลื่อนเหนี่ยวนำย้อนกลับ ค่าความต้านทานของขดลวดอาร์เมเจอร์ และค่าความเหนี่ยวนำของขดลวดอาร์เมเจอร์ ดังสมการ และรูปที่ 2.14

$$v_t = e_a + R_a i_a + L_a (dI_a/dt) \qquad (2.4)$$

เมื่อ v_t คือ แรงดันไฟฟ้าป้อนที่ขั้วของอาร์เมเจอร์

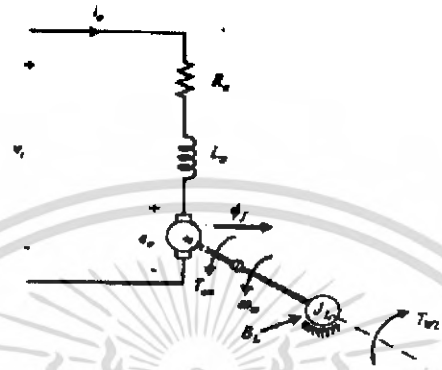
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

e_a คือ ค่าแรงเคลื่อนเหนี่ยวนำย้อนกลับ

R_a คือ ค่าความต้านทานของขดลวดอาร์เมเจอร์

i_a คือ กระแสอาร์เมเจอร์

L_a คือ ค่าความเหนี่ยวนำของขดลวดอาร์เมเจอร์



รูปที่ 2.14 วงจรสมมูลมอเตอร์ไฟฟ้ากระแสตรง

สำหรับความสัมพันธ์ระหว่างแรงบิดของมอเตอร์ กับแรงบิดที่โหลดต้องการ จะเท่ากับ

$$T_{em} = J(d\omega_m/dt) + B\omega_m + T_{WL}(t) \qquad (2.5)$$

เมื่อ T_{em} คือ แรงบิดของมอเตอร์ไฟฟ้า

J คือ ผลรวมวงจรสมมูลของแรงเฉื่อย (Total Equivalent Inertia) ของโหลด

B คือ ผลรวมวงจรสมมูลของการแกว่ง (Total Equivalent Damping)

ω_m คือ ความเร็วตัวนำของอาร์เมเจอร์

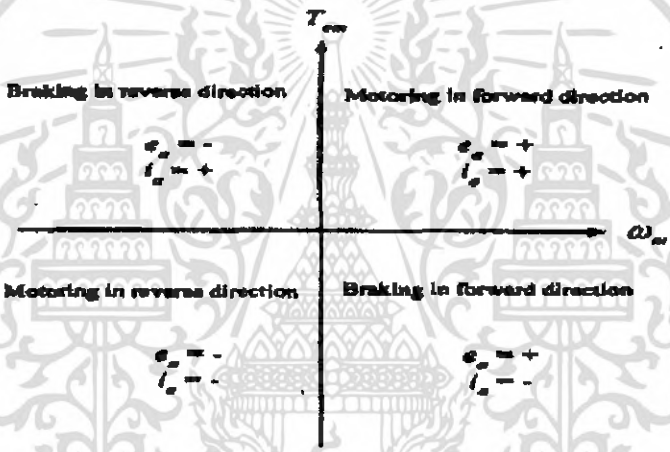
T_{WL} คือ วงจรสมมูลของแรงบิดที่โหลดต้องการในการทำงาน

เครื่องจักรกลไฟฟ้ากระแสตรงนั้น สามารถทำงานเป็นมอเตอร์หรือเป็นเครื่องกำเนิดไฟฟ้ากระแสตรงก็ได้ แต่โดยทั่วไปไม่นิยมนำมาใช้เป็นเครื่องกำเนิดไฟฟ้ากระแสตรง แต่อย่างไรก็ดีในขณะที่ทำการเบรกมอเตอร์ไฟฟ้ากระแสตรงนั้น จะเกิดสภาวะเสมือนเป็นเครื่องกำเนิดไฟฟ้ากระแสตรงในขณะที่ความเร็วเริ่มลดลง ดังนั้นจึงจำเป็นที่จะต้องพิจารณากรณีที่มอเตอร์ไฟฟ้ากระแสตรงทำตัวเป็นเครื่องกำเนิดไฟฟ้ากระแสตรงขณะเบรก โดยกำหนดให้เส้นแรงแม่เหล็กมีค่าคงที่และมอเตอร์กำลังขับโหลดที่ความเร็วเท่ากับ ω_m และเพื่อที่จะลดความเร็วของมอเตอร์ไฟฟ้าลงโดยทำให้ค่าแรงดันไฟฟ้า v_t มีค่าต่ำกว่าค่าแรงเคลื่อนเหนี่ยวนำย้อนกลับ e_a จะเป็นผลทำให้กระแสอาร์เมเจอร์ i_a ไหลกลับทิศทาง ค่าแรงบิด T_{em} ก็กลับทิศทางเช่นกัน นอกจากนั้นพลังงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จลน์ที่เกิดจากแรงเฉื่อยของโหลดก็จะแปลงเป็นพลังงานไฟฟ้า ซึ่งพฤติกรรมดังกล่าวหมายถึงมอเตอร์ไฟฟ้าได้ทำตัวเป็นเครื่องกำเนิดไฟฟ้า โดยพลังงานจะถูกส่งผ่านไปยังแหล่งจ่ายแรงดันไฟฟ้า v_t หรือกระจายให้อยู่ในรูปความร้อนด้วยการใช้ตัวต้านทาน

ขณะที่มอเตอร์ไฟฟ้ากระแสตรงทำการเบรก ขั้วของแรงเคลื่อนเหนี่ยวนำย้อนกลับe_a จะไม่เปลี่ยนแปลง เพราะทิศทางในการหมุนยังคงไม่เปลี่ยนแปลงและเมื่อความเร็วของโรเตอร์ลดลงเป็นผลทำให้แรงเคลื่อนเหนี่ยวนำย้อนกลับ e_a มีค่าลดลง (สมมติให้เส้นแรงแม่เหล็ก ϕ_f มีค่าคงที่) จะกระทั่งโรเตอร์ของมอเตอร์ไฟฟ้ากระแสตรงหยุดหมุน และขั้วของแหล่งจ่ายแรงดันไฟฟ้ามีการเปลี่ยนแปลง ก็จะมีผลทำให้ทิศทางการหมุนของตัวโรเตอร์เปลี่ยนแปลงเช่นกันดังนั้นมอเตอร์ไฟฟ้ากระแสตรงจะสามารถทำงานได้ในทิศทางใดทิศทางหนึ่ง และเมื่อทำการเบรก แรงบิดของตัวมอเตอร์ไฟฟ้ากระแสตรงก็สามารถกลับทิศทางได้ดังรูปที่ 2.15



รูปที่ 2.15 การทำงานของมอเตอร์ไฟฟ้ากระแสตรงทั้ง 4 ย่านการทำงาน

ซึ่งความสัมพันธ์ระหว่างแรงบิดกับความเร็วรอบสำหรับย่านการทำงานของมอเตอร์ไฟฟ้ากระแสตรงทั้ง 4 ย่านการทำงาน

ประเภทของมอเตอร์ไฟฟ้ากระแสตรง

มอเดอร์ไฟฟ้ากระแสตรงสามารถแบ่งออกได้เป็นหลายประเภท ขึ้นอยู่กับลักษณะการสร้างสนามแม่เหล็กของตัวมอเตอร์ และขึ้นอยู่กับพื้นฐานการออกแบบโครงสร้างของอาร์เมเจอร์ การแบ่งประเภทตามลักษณะการจ่ายสนามแม่เหล็กแยกได้เป็น 2 แบบ คือ

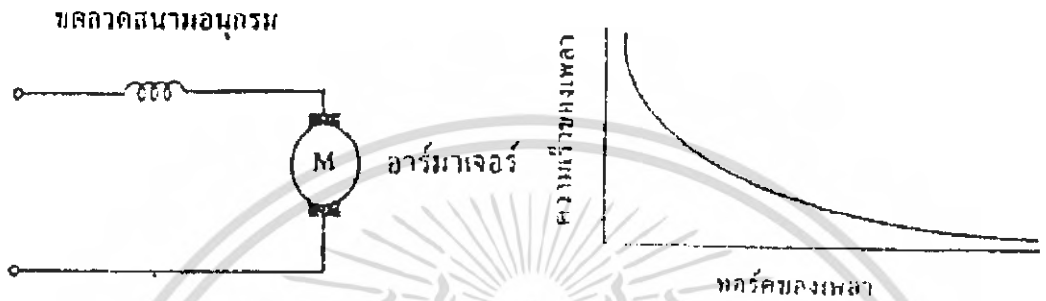
1. แบบปรับเส้นแรงแม่เหล็กได้
2. แบบเส้นแรงแม่เหล็กมีค่าคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. มอเตอร์ไฟฟ้ากระแสตรงแบบปรับเส้นแรงแม่เหล็กได้ แบ่งออกได้เป็น 2 แบบ คือ

1.1 แบบขดลวดสนามแม่เหล็กต่ออนุกรมอยู่กับขดลวดอาร์เมเจอร์

มอเตอร์แบบนี้มีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแส ดังนั้น เส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ และเราจะได้ความสัมพันธ์ระหว่างความเร็วและแรงบิดเป็น nonlinear ดังรูป



รูปที่ 2.16 ดีซีมอเตอร์แบบอาร์เมเจอร์ต่ออนุกรมกับขดลวดสนามแม่เหล็ก

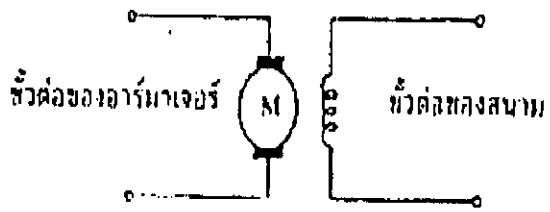
รูปที่ 2.17 คุณสมบัติระหว่างแรงบิดกับความเร็วของดีซีมอเตอร์อนุกรมภายใต้ภาวะโวลท์เตจคงที่

มอเตอร์ต่อดังกล่าวจะใช้ในงานสภาวะเฉพาะเมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดต่ำที่ความเร็วสูง เช่น ระบบขับเคลื่อนของรถลาก

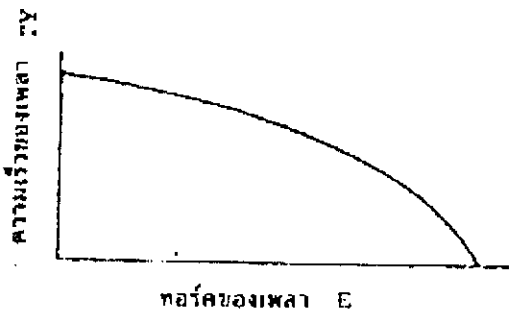
1.2 แบบขดลวดสนามแม่เหล็กแยกกระตุ้น

มอเตอร์แบบนี้มักนิยมเรียกกันว่ามอเตอร์ขนาน (shunt motor) มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์เมเจอร์ ยังผลให้สามารถควบคุมพารามิเตอร์ของมอเตอร์ให้มีค่าคงที่ได้ตลอดช่วงพิสัยที่กว้าง มอเตอร์แบบนี้มักจะใช้งานในกรณีระบบบังคับการเคลื่อนที่ต้องการแรงบิดสูง ในรูปที่ 2.18 แสดงคุณสมบัติระหว่างแรงบิดกับความเร็วมอเตอร์ขนานภายใต้สภาวการณ์การกระตุ้นสนามแม่เหล็กคงที่ และอาร์เมเจอร์โวลท์เตจคงที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 ดีซีมอเตอร์แบบแยกปรับสนามแม่เหล็กได้



รูปที่ 2.19 คุณสมบัติระหว่างแรงบิดกับความเร็วของมอเตอร์ขนานอนุกรมภายใต้ภาวการณ์กระตุ้นสนามแม่เหล็กคงที่และอาร์เมเจอร์โวลต์เตจคงที่

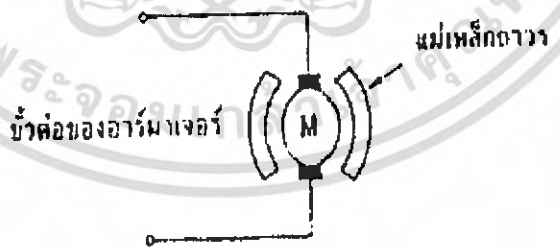
2. มอเตอร์ไฟฟ้ากระแสตรงแบบเส้นแรงแม่เหล็กคงที่

ระบบการกระตุ้นสนามแม่เหล็กของมอเตอร์ โดยทั่วไปในปัจจุบันมักใช้เป็นแบบแม่เหล็กถาวร ดังแสดงในรูปที่ 2.20 ในระบบนี้เส้นแรงของสนามแม่เหล็กมีค่าคงที่ ดังนั้น อัตราส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดจะมีค่าคงที่ ดังนั้น จะได้สมการ

$$T = K_I I \tag{2.6}$$

และสมการ

$$E = K_w w \tag{2.7}$$



รูปที่ 2.20 ดีซีมอเตอร์แบบฟิลด์เป็นแบบแม่เหล็กถาวร

ระบบนี้จะให้ความสัมพันธ์ระหว่างกระแสอาร์เมเจอร์ แรงบิดและความเร็วอยู่ในลักษณะเป็นเส้นตรง (linear) สมการทางไฟฟ้าของมอเตอร์ไฟฟ้ากระแสตรงแบบนี้เขียนได้เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$V = K_E w + Ldi/dt + Ri \quad (2.8)$$

เมื่อ V คือ แรงดันที่ป้อนให้กับมอเตอร์

K_E คือ ค่าคงที่ของแรงดันย้อนกลับ

L คือ ค่าความเหนี่ยวนำของอาร์เมเจอร์

R คือ ความต้านทานที่ขั้วของมอเตอร์

สมการไดนามิคของมอเตอร์ คือ

$$T_G = Jdw/dt + Bw + T_F + T_L \quad (2.9)$$

เมื่อ T_G คือ แรงบิดที่กำเนิดโดยมอเตอร์

J คือ ผลรวมของโมเมนต์ของแรงเฉื่อยของมอเตอร์และโหลด

B คือ สัมประสิทธิ์ของ viscos damping

T_F คือ แรงบิดเสียดทานภายใน

T_L คือ แรงบิดโหลด

สมการต่างๆ ของมอเตอร์แบบแยกกระตุ้นสนามแม่เหล็กจะเหมือนกับการกระตุ้นสนามแม่เหล็กคงที่ อย่างไรก็ตามข้อดีของมอเตอร์แบบสนามแม่เหล็กแม่เหล็กถาวรซึ่งเหนือกว่ามอเตอร์แบบมีโครงสร้างของสนามแม่เหล็กด้วยการพันของขดลวด คือ ไม่มีกำลังสูญเสียในสนามแม่เหล็กมีประสิทธิภาพสูงกว่า และมีขนาดเล็กกว่าเมื่อเทียบกับมอเตอร์ที่มีขนาดของกำลังม้าเท่ากันนอกจากนั้นความสัมพันธ์เชิงเส้นยังให้ค่าของกระแสอาร์เมเจอร์ที่สูงกว่ามอเตอร์กระแสตรงแบบ field เป็นขดลวด การประยุกต์ใช้งานเหมาะกับระบบที่ต้องการแรงบิดของโหลดสูง

พื้นฐานของระบบควบคุมมอเตอร์ไฟฟ้ากระแสตรง

ส่วนประกอบพื้นฐานของระบบคอนโทรลมอเตอร์ไฟฟ้ากระแสตรงประกอบด้วยบล็อกที่สำคัญ 4 บล็อก คือ

1. ตัวคอนโทรลเลอร์ เป็นส่วนประกอบของระบบที่ทำให้เกิดสัญญาณคอนโทรลไปบังคับมอเตอร์ไฟฟ้ากระแสตรงและโหลดคอนโทรลเลอร์ที่ให้สัญญาณคอนโทรลเป็นสัญญาณอนาลอก เรียกว่า อนาลอกคอนโทรลเลอร์ ส่วนคอนโทรลเลอร์ที่ให้สัญญาณคอนโทรลเป็นสัญญาณดิจิดอล เรียกว่า ดิจิตอลคอนโทรลเลอร์

2. วงจรไดรเวอร์หรือพาวเวอร์แอมปลิไฟเออร์ เป็นส่วนประกอบของระบบที่อยู่ระหว่างตัวคอนโทรลเลอร์กับมอเตอร์ไฟฟ้ากระแสตรงและโหลดมีหน้าที่ปรับรูปแบบและขยาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าธนบุรี

สัญญาณให้เหมาะสมก่อนที่จะป้อนเข้าไปขับมอเตอร์ไฟฟ้ากระแสตรงและโหลด วงจรไดรเวอร์ส่วนใหญ่ ได้แก่ พาวเวอร์แอมปลิไฟเออร์ ซึ่งอาจแบ่งย่อยออกเป็นลิเนียร์พาวเวอร์แอมปลิไฟเออร์และพัลซ์วิดโมดูเลชันแอมปลิไฟเออร์

3. ฟีดแบ็คทรานสดิวเซอร์หรือเอนโค้ดเดอร์ เป็นสิ่งประดิษฐ์ที่ใช้ตัวรับรู้หรือดีเทคสัญญาณเอาต์พุตที่ต้องการโดยไม่มีผลของการ loading สัญญาณที่ดีเทคนี้จะป้อนกลับไปเปรียบเทียบกับสัญญาณอ้างอิงทำให้ได้สัญญาณ error ฟีดแบ็คทรานสดิวเซอร์ แบ่งออกได้เป็น 2 แบบ คือ

3.1 อนาลอกทรานสดิวเซอร์ คือ สิ่งประดิษฐ์เปลี่ยนพลังงานรูปแบบหนึ่งให้เป็นสัญญาณอนาลอก ได้แก่ พวกทาโคเจนเนอเรเตอร์ โพเทนทิโอมิเตอร์และซิงโคร เป็นต้น

3.2 ดิจิตอลทรานสดิวเซอร์ เป็นสิ่งประดิษฐ์เปลี่ยนพลังงานรูปแบบหนึ่งให้เป็นสัญญาณดิจิตอล ได้แก่ พวกอินคริเมนท์เอนโค้ดเดอร์รีโซลเวอร์ แมกนิติคพิคอัพ

4. มอร์เตอร์ไฟฟ้ากระแสตรงและโหลด คือ ระบบที่ถูกคอนโทรลหรือส่วนที่ออกแรงทำงานซึ่งจะเป็นเครื่องจักรกล มอเตอร์ไฟฟ้ากระแสตรงหรืออะไรก็ตามที่ให้ตัวแปร มอเตอร์ไฟฟ้ากระแสตรงในที่นี้เป็นแบบแม่เหล็กถาวรที่มีคุณสมบัติการทำงานสูง มีอาร์เมเจอร์อินดัคแต้นซ์และแรงเฉื่อยของโรเตอร์ต่ำ

ลักษณะการควบคุมของระบบมอเตอร์ไฟฟ้ากระแสตรง

ระบบการคอนโทรลมอเตอร์ไฟฟ้ากระแสตรงสามารถจำแนกลักษณะการคอนโทรลออกได้เป็น 2 แบบ คือ ระบบอนาลอกคอนโทรล และระบบดิจิตอลคอนโทรล ในโครงงานนี้ใช้ลักษณะการควบคุมแบบดิจิตอล จึงขออธิบายแต่ระบบดิจิตอลคอนโทรล

ระบบดิจิตอลคอนโทรล คือ ระบบที่คอนโทรลลูฟฟีดแบ็คทรานดิวเซอร์เป็นดิจิตอลทรานดิวเซอร์ซึ่งสามารถวัดค่าไดนามิกวาริเอเบิลออกเป็นสัญญาณดิจิตอลหรือในรูปของสัญญาณเอนโค้ดดิ้งของไบนารี คือ สัญญาณเอาท์พุตของดิจิตอลทรานดิวเซอร์จะเป็นโค้ดไบนารี่ป้อนกลับไปยังตัวดิจิตอลคอนโทรลเลอร์เพื่อควบคุมให้คุณสมบัติการทำงานของระบบเป็นไปตามที่ต้องการ

2.3 ตัวจับสัญญาณ (sensor) และเอนโคดเดอร์ (encoder) ในระบบการควบคุมมอเตอร์

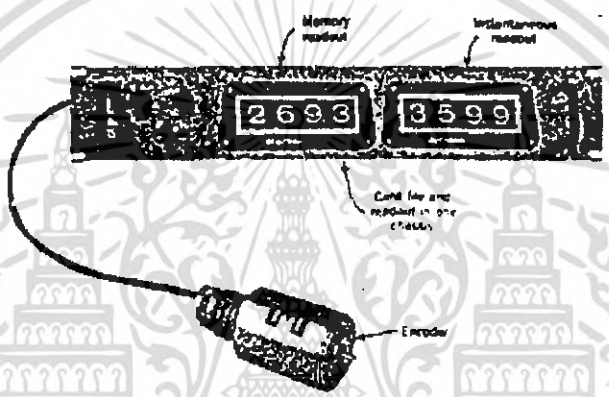
ตัวจับสัญญาณและเอ็นโคดเดอร์ เป็นส่วนประกอบที่สำคัญในระบบคอนโทรลที่มีการป้อนกลับในระบบคอนโทรลแบบเปิดลูป เอ็นโคดเดอร์มักใช้สำหรับตรวจสอบคุณสมบัติการทำงานของระบบ ส่วนในระบบคอนโทรลแบบปิดลูป ตัวจับสัญญาณและเอ็นโคดเดอร์ใช้เป็นตัวป้อนสัญญาณกลับเพื่อการคอนโทรล นอกจากนั้นส่วนประกอบเหล่านี้ ยังใช้สำหรับพิสูจน์ความเหมือนกันของกระบวนการที่ไม่รู้ค่าหรือกระบวนการที่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
62906

ตัวจับสัญญาณและเอ็นโคดเดอร์ที่ใช้กันมากในระบบคอนโทรล เช่น โพเทนทิโอมิเตอร์ (Potentiometer), ออพติคอลเอนโคดเดอร์ (optical encoder), แมกเนติกปิคทรานสดิวเซอร์ (magnetic picks transducer) จำพวกแม่เหล็กไฟฟ้า และรีโซลเวอร์ (resolver)

ตัวจับสัญญาณและเอนโคดเดอร์ที่ใช้กับแขนกลนี้เป็นเอนโคดเดอร์แบบที่นับการเพิ่มค่า อินครีเมนทอลเอนโคดเดอร์ (increment encoder)

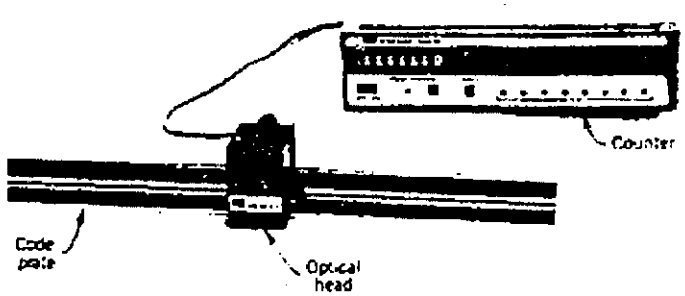
เอนโคดเดอร์ที่นับการเพิ่มค่าใช้ในระบบการบังคับตำแหน่งหรือความเร็วของมอเตอร์ และสร้างสัญญาณป้อนกลับ โดยที่ตัวเอนโคดเดอร์จะสร้างสัญญาณลูกคลื่น (pulse) ที่แปรผันตรงกับการหมุนของเพลาซึ่งสามารถนำไปใช้กับการรับรู้ความเร็วของเพลามอเตอร์ในรูปของอัตราจำนวน



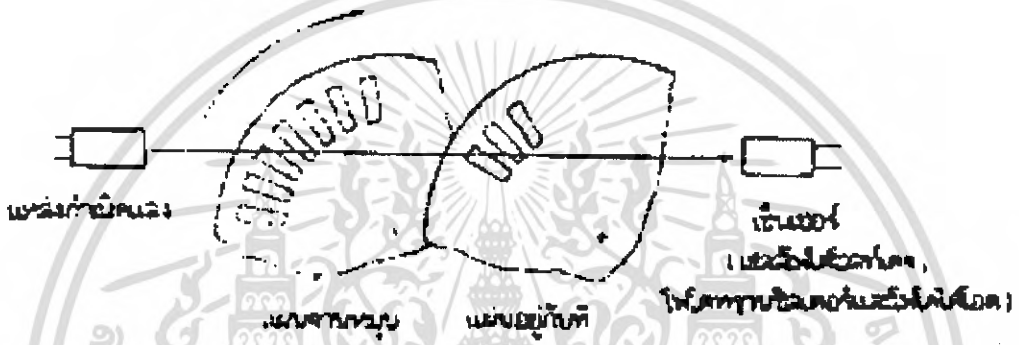
รูปที่ 2.21 ระบบเอนโคดเดอร์แบบโรตารี่

สัญญาณลูกคลื่นได้ ในรูปที่ 2.21 และรูปที่ 2.22 แสดงรูปลักษณะของเอ็นโคดเดอร์ที่นับการเพิ่มค่าแบบโรตารี่และแบบ linear เอ็นโคดเดอร์ที่นับการเพิ่มค่าประกอบด้วยส่วนที่สำคัญคือตัวกำเนิดแสงจานหมุน (rotary disk) จานอยู่กับที่ และตัวจับสัญญาณบนแผงวงจานทำเป็นช่องโดยรอบดังแสดงในรูปที่ 2.23 และแผ่นอยู่กับที่จะมีช่องสำหรับให้แสงผ่านตรงเข้าไปยังตัวจับสัญญาณ ถ้าเป็นเอ็นโคดเดอร์ที่ใช้วัดความเร็วต่ำไม่ต้องมีแผ่นอยู่กับที่ก็ได้ ส่วนตัวกำเนิดแสงอาจจะเป็นหลอดไฟหรือหลอด LED ก็ได้ ดังแสดงในรูปที่ 2.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 เอนโคดเดอร์ที่นับการเพิ่มค่าแบบเส้นตรง



รูปที่ 2.23 ตัวอย่างกลไกของเอนโคดเดอร์แบบออพติคอลอินครีเมนท์

รูปที่ 2.24 ตัวอย่างตัวจับสัญญาณแบบมีช่องปิดเปิดให้แสงผ่านได้ช่องเดียวและแบบมีหลายช่อง

ความละเอียดของเอ็นโคดเดอร์ที่นับการเพิ่มค่า

ความละเอียดของเอ็นโคดเดอร์คือจำนวนคาบเวลาของสัญญาณเอาท์พุตต่อการหมุนของเพลา 1 รอบ ซึ่งเป็นจำนวนสัญญาณลูกคลื่นต่อรอบหรือจำนวนรอบต่อ 360 องศามุมทางเชิงกล หรือรอบต่อองศา เอ็นโคดเดอร์ที่ใช้กันทั่วๆ มีค่าความละเอียดตั้งแต่ 15-10000 สัญญาณลูกคลื่นต่อรอบ

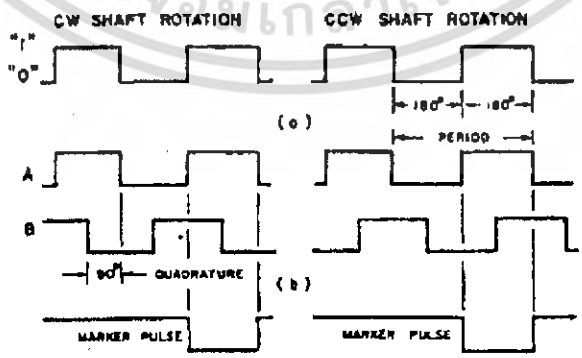
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากแสงที่ออกมาจากแหล่งกำเนิดเป็นลำแสงเดียว ถ้าเราต้องการให้แสงที่ผ่านช่องไปยังตัวจับสัญญาณเป็นเส้นตรงพร้อมๆ กัน ก็ทำได้โดยใช้เลนส์หรือพาราโบลิกรีเฟลกเตอร์ (parabolic reflector) จำนวนสัญญาณรูปคลื่นต่อ 1 รอบของสัญญาณที่เอ็นโคดเดอร์สร้างออกมาจะเท่ากับจำนวนช่องว่างบนแผ่นจานหมุนและความกว้างของช่องว่างเท่ากับความกว้างของแถบทึบระหว่างช่องว่างจะเท่ากัน

เอาท์พุตของเอ็นโคดเดอร์

โดยทั่วๆ ไปแล้วสัญญาณเอาท์พุตที่ออกจากเอ็นโคดเดอร์โดยตรงจะมีระดับไม่เพียงพอในการควบคุมหรือสำหรับการประมวลสัญญาณ ดังนั้นจึงต้องมีวงจรขยายและแปลงรูปร่างลูกคลื่นสัญญาณต่อไว้ในเอ็นโคดเดอร์ด้วยเสมอ สัญญาณรูปคลื่นที่ได้จากตัวรับสัญญาณปกติจะเป็นรูปสัญญาณสามเหลี่ยมหรือรูปสัญญาณซายน์ขึ้นอยู่กับความละเอียดที่ต้องการ รูปสัญญาณเหล่านี้สามารถทำให้เป็นสัญญาณรูปสี่เหลี่ยมได้ โดยการต่อตัวคอมพาราเดอร์เข้ากับ linear amplifier ของเอ็นโคดเดอร์ก็จะได้เอาท์พุตเป็นลูกคลื่นสี่เหลี่ยมตามต้องการ รูปที่ 2.25a แสดงถึงรูปคลื่นเอาท์พุตสี่เหลี่ยมของเอ็นโคดเดอร์ชนิด 1 ช่อง ไม่ว่าเพลาจะหมุนในทิศทางใดก็ได้ สัญญาณออกมาเหมือนกันจึงเหมาะที่ใช้กับงานที่ไม่กำหนดทิศทางเท่านั้น ส่วนในรูปที่ 2.25b แสดงสัญญาณ 2 ชุดที่ได้จากเอ็นโคดเดอร์ชนิด 2 ช่องนี้จะต่าง 90 องศาทางไฟฟ้า เรียกสัญญาณ 2 ช่องนี้ว่าเป็นควอดราเจอร์ (Quadrature) ซึ่งเหมาะที่จะใช้ในการรับรู้ทิศทางการหมุนของเพลาหรือใช้ควบคุมระยะที่ซับซ้อนอื่นๆ จากสัญญาณในรูปที่ 2.25b จะเห็นได้ว่าสัญญาณทั้ง 2 ช่องจะเริ่มจาก 0 ถึง 1 และ 1 ถึง 0 ขึ้นอยู่กับทิศทางการหมุนของเอ็นโคดเดอร์

ในเอ็นโคดเดอร์ที่นับการเพิ่มค่าบางชนิดจะมีสัญญาณลูกคลื่นแสดงถึงจำนวนรอบของการหมุน สำหรับใช้เป็นศูนย์ในการอ้างอิงสัญญาณลูกคลื่นที่ใช้แสดงจำนวนรอบนี้ จะเกิดขึ้น 1 สัญญาณลูกคลื่นต่อ 1 รอบ โดยทั่วไปแล้วใช้บอกถึงตำแหน่งเชิงกลหรือใช้เป็นสัญญาณเคลียร์จำนวนที่นับไว้ในหน่วยเก็บข้อมูล



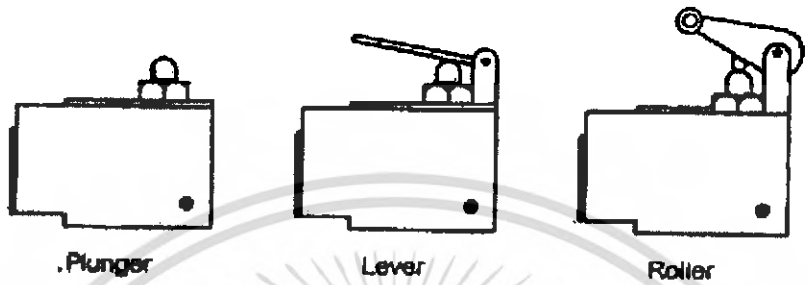
รูปที่ 2.25a ตัวอย่างลูกคลื่นเอาท์พุตสี่เหลี่ยมของอุปกรณ์เอนโคดเดอร์ช่องเดียว

รูปที่ 2.25b สัญญาณเอนโคดเดอร์ 2 ช่อง ที่มีมุมเฟสต่างกัน 90 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ไมโครสวิตช์ (Microswitch)

คือ สวิตช์ขนาดเล็ก การทำงานอาศัยการเคลื่อนที่ขนาดเล็กมากๆ ของหน้าสัมผัสทำให้สวิตช์ประเภทนี้มีความไวในการทำงานมาก และเหมาะกับงานที่มีการสัมผัสงานโดยตรง รูปแบบต่างๆ ของไมโครสวิตช์แสดงได้ดังรูปที่ 2.26



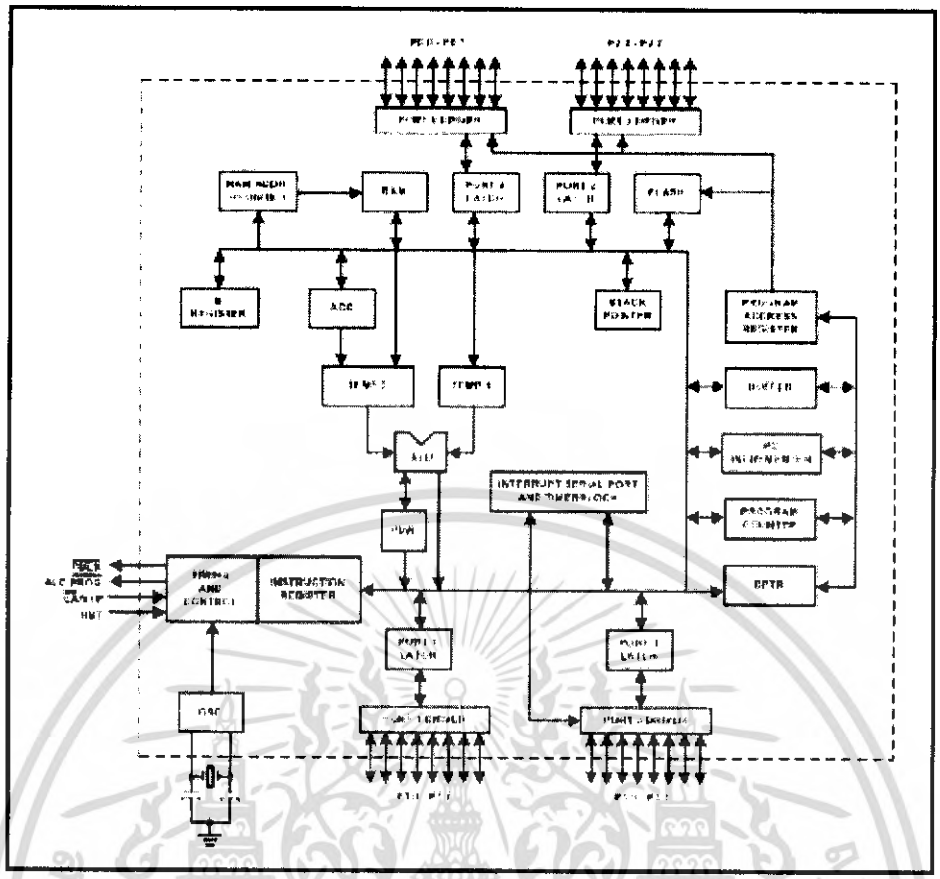
รูปที่ 2.26 ไมโครสวิตช์

ซึ่งมีรูปแบบของหน้าสัมผัสแตกต่างกันไปตามลักษณะของการนำไปประยุกต์ใช้งาน ปกติหน้าสัมผัสของไมโครสวิตช์เป็นไปทั้งหน้าสัมผัสแบบปกติปิด (normally close) หรือแบบปกติเปิด (normally open) ขึ้นกับจุดประสงค์ของการใช้งานว่าต้องการให้สวิตช์ทำหน้าที่ในการตัดหรือต่อวงจรทำงาน เช่น สวิตช์หยุดการทำงานของเครื่องจักร หรือ แจ้งเตือนด้วยเสียง ทดสอบขนาด และน้ำหนักของผลิตภัณฑ์ว่าเป็นไปตามคุณสมบัติข้อกำหนดหรือไม่

2.5 MCS-51

ไมโครคอนโทรลเลอร์หรืออาจเรียกว่าชิงเกิลชิพไมโครคอนโทรลเลอร์ (Single Chip Microcontroller) เป็นอุปกรณ์ที่นำเอาไมโครโพรเซสเซอร์มารวมกับหน่วยความจำและอุปกรณ์ไอโอต่าง ๆ เอาไว้ในตัวเดียวกันสามารถทำงานได้ทันทีเมื่อป้อนไฟเลี้ยงและสัญญาณนาฬิกาให้ตัวไมโครคอนโทรลเลอร์ จึงทำให้การออกแบบวงจรง่ายขึ้น และมีขนาดเล็กลงเป็นอย่างมาก บล็อกไดอะแกรมของไมโครคอนโทรลเลอร์แสดงในรูปที่ 2.27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

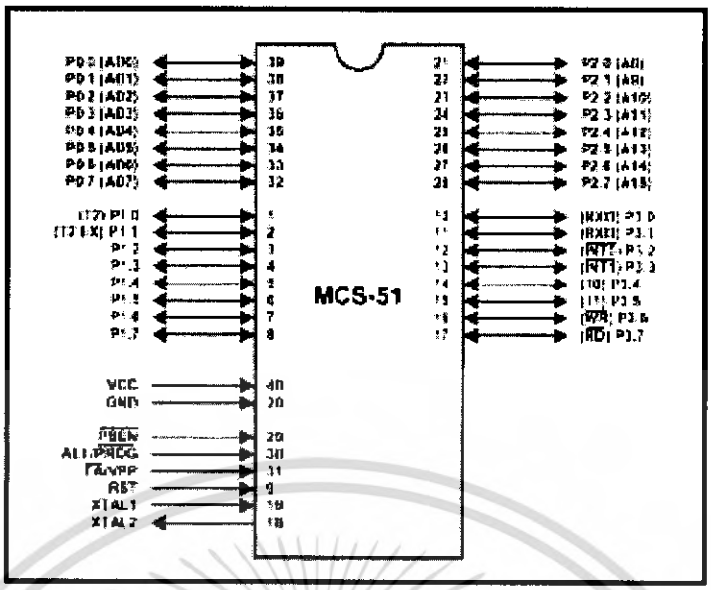


รูปที่ 2.27 บล็อกไดอะแกรมของ MCS-51

การพัฒนาขีดความสามารถของไมโครคอนโทรลเลอร์ได้เป็นมาอย่างต่อเนื่อง ซึ่งมีการพัฒนาทั้งในด้านฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) ในการพัฒนาด้านฮาร์ดแวร์นั้น สิ่งที่เห็นได้ชัดเจนคือ การพัฒนาเกี่ยวกับหน่วยความจำ ซึ่งได้เริ่มมีการพัฒนามาจาก PROM EPROM EEPROM และ ในปัจจุบัน ได้พัฒนามาจนถึงหน่วยความจำที่เรียกว่า FLASH MEMORY

ในส่วนของซอฟต์แวร์นั้นได้พัฒนาการเขียนโปรแกรมมาจาก การเขียนโปรแกรมในระดับบิต (เลขฐาน 2), ระดับไบต์ (เลขฐาน 16), แอสแซมบลี, และภาษาในระดับสูง เช่น ภาษาเบสิกและภาษาซี เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 การจัดขาของ MCS-51

การจัดหน่วยความจำของไอซีไมโครคอนโทรลเลอร์

หน้าที่การทำงานของหน่วยความจำจะทำหน้าที่เก็บโปรแกรมคำสั่ง และข้อมูลที่จะใช้ในการกำหนดค่าต่างๆให้กับไมโครคอนโทรลเลอร์ หรือใช้เก็บค่าต่างๆที่ไมโครคอนโทรลเลอร์ได้กระทำตามคำสั่งการจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 จะแบ่งหน่วยความจำออก เป็น 3 กลุ่มคือ

1. หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory) หรือ (Code Memory)
2. หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)
3. รีจิสเตอร์ที่ทำหน้าที่เฉพาะ (Special Function)

การจัดหน่วยความจำของไอซีไมโครคอนโทรลเลอร์

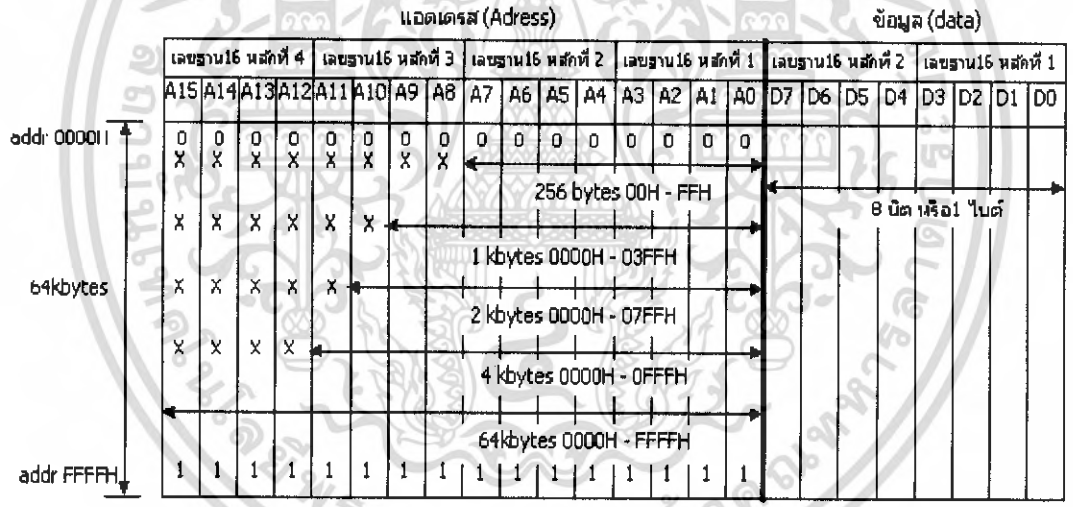
1. หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory) หรือหน่วยความจำรหัสคำสั่ง (Code Memory)

หน่วยความจำสำหรับเก็บโปรแกรม (ทำหน้าที่เช่นเดียวกับรอม) หรือหน่วยความจำรหัสคำสั่ง (Code Memory) จะทำหน้าที่เก็บชุดคำสั่งเพื่อให้ไมโครคอนโทรลเลอร์ปฏิบัติตามคำสั่งนั้น ๆ ยกตัวอย่าง เช่นในขณะที่เราเปิดเครื่องไมโครเวฟ จะมีการแสดงผลรายการหลักที่หน้าจอ LCD เพื่อคอยให้เราป้อนค่าเวลาที่ต้องการจะอุ่นอาหาร คำสั่งที่จอ LCD เพื่อให้เราป้อนข้อมูลนั้นจะเขียนคำสั่งอยู่ในส่วนของหน่วยความจำโปรแกรมนั้นเอง ถึงเราจะเปิดเครื่องไมโครเวฟกี่ครั้ง ก็จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแสดงผลที่ LCD ให้เราป้อนค่าเวลาที่ต้องการเหมือนเดิม ภายในตัวไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C1051AT89C2051 และ AT89C4051 จะมีหน่วยความจำที่เก็บโปรแกรมได้ 1, 2 และ 4 Kbytes ตามลำดับ หน่วยความจำจะเป็นลักษณะแบบแฟลช ที่มีคุณสมบัติในการใช้งานโดยสามารถจะทำการลบข้อมูลด้วยไฟฟ้า และเก็บข้อมูลเข้าเก็บไว้ในตัวไมโครคอนโทรลเลอร์ ได้กว่า 1000 ครั้ง โดยใช้เครื่องโปรแกรมที่ไม่ยุ่งยากและราคาไม่แพง (สามารถรักษาข้อมูลไว้ได้นานหลายปี) ข้อสังเกต ส่วนของแอดเดรส(ADRRES)ไม่สามารถที่จะใช้ตำแหน่งเดียวกันได้ แต่ข้อมูล (DATA)สามารถที่จะมีข้อมูลเหมือนกันได้ จากตารางอุปมาเหมือนกับมีกระดาษจำนวนเท่ากับ 2n บรรทัดมาให้ ดังนั้นหากต้องการเขียนข้อมูลใดๆลงในแต่ละบรรทัด จะต้องมีตำแหน่งของบรรทัดที่ไม่ซ้ำกัน และการที่จะเลือกจำนวนของบรรทัด ขึ้นอยู่กับปริมาณของข้อมูลที่ต้องการจะเขียน ยกตัวอย่างเช่นหากต้องการขนาดของข้อมูลในการเขียนโปรแกรมเพียง 2000 บรรทัด อาจจะเลือกใช้หน่วยความจำขนาด 2 Kbytes โดยมีแอดเดรสตั้งแต่ 0000H - 07FFH

ตารางที่ 2.3 หน่วยความจำสำหรับเก็บโปรแกรมของไอซีไมโครคอนโทรลเลอร์



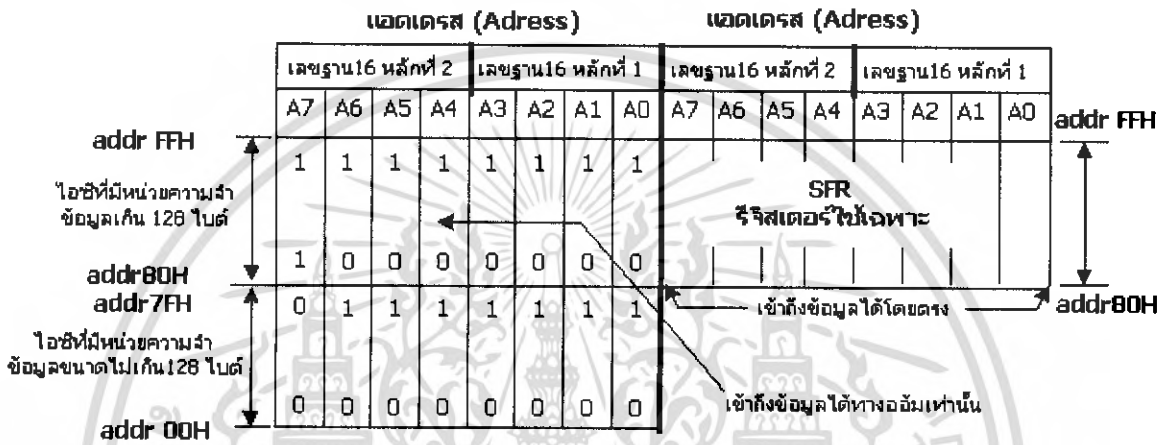
2. หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

หน่วยความจำข้อมูล(RAM) จะทำหน้าที่เก็บรักษาข้อมูล โดยข้อมูลอาจจะเป็นค่าหลังจากไมโครคอนโทรลเลอร์ ทำการการประมวลผล หรือเก็บค่าข้อมูลที่จะให้กับไมโครคอนโทรลเลอร์ประมวลผลในขณะนั้น และจะทำหน้าที่เป็น สแตก (Stack)บางส่วน (ส่วนของสแตกจะอธิบายในลำดับต่อไป) ยกตัวอย่าง เช่น ถ้าเป็นเครื่องไมโครเวฟที่ใช้สำหรับอุ่นอาหาร ก็คือส่วนที่เราป้อนข้อมูลเช่นเวลา หรืออุณหภูมิที่เป็นปัจจุบัน หลังจากหน่วยความจำโปรแกรมแสดงรายการ หลักที่ LCD นั้นเอง สังเกตว่าหากเราปิดเครื่อง แล้วเปิดเครื่องใหม่อีกครั้งหนึ่ง ค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่เป็นเวลา และอุณหภูมิเดิมที่เรากำหนดไว้ในครั้งแรกก็จะหายไป และจะให้เราป้อนค่าข้อมูลใหม่อีกครั้ง ดังนั้นการที่จะรักษาข้อมูลเดิมไว้ได้ จะต้องมีแหล่งจ่ายไฟสำรองไว้สำหรับเพื่อเลี้ยงให้กับตัวไอซีตลอดเวลา หรือที่ เรียกว่า Battery backup * สำหรับไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C1051 จะมีหน่วยความจำที่เก็บข้อมูลได้ 64 bytes ส่วนAT89C2051 และ AT89C4051 จะมีหน่วยความจำที่เก็บข้อมูลได้ 128 bytes

ตารางที่ 2.4 ตำแหน่งแอดเดรสของหน่วยความจำข้อมูลภายใน(Internal data memory)



หน่วยความจำข้อมูลภายในยังแบ่งส่วนของการใช้งานได้อีกเป็นสองส่วนคือ หน่วยความจำข้อมูลภายใน 128 ไบต์จะเป็นหน่วยความจำที่ใช้งานทั่วไปอยู่ที่ตำแหน่งแอดเดรส 00H-7FH และหน่วยความจำในตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งจะเป็นส่วนของรีจิสเตอร์เฉพาะ (Special Function Register) ในส่วนของหน่วยความจำที่ใช้งานทั่วไป จะแสดงได้ดังรูป2.10 โดยพื้นที่ของหน่วยความจำข้อมูลภายในที่ตำแหน่งแอดเดรส 00H-7FH ก็ยังสามารถที่จะแบ่งออกเป็นส่วนย่อยได้ดังนี้

1. พื้นที่ในหน่วยความจำข้อมูล (แรม)ตำแหน่งที่ 00H-1FH จำนวน 32 ไบต์ จะถูกแบ่งออกเป็น กลุ่ม เรียกว่า แบงค์ (Bank) และในแต่ละแบงค์ จะมี 8 ไบต์ ดังแสดงในรูป10 พื้นที่ในแต่ละแบงค์จะถูกใช้งานเป็นรีจิสเตอร์ที่ใช้งานทั่วไป (รีจิสเตอร์ R0-R7 เป็นรีจิสเตอร์ที่มีขนาด 8 บิต หรือ 1 ไบต์) โดยที่รีจิสเตอร์ R0 จะอยู่ในตำแหน่งแรกของแต่ละแบงค์ และ รีจิสเตอร์ R7 จะอยู่ในตำแหน่งสุดท้ายของแต่ละแบงค์ ในการนำไปใช้งาน จะเลือกใช้รีจิสเตอร์ R0-R7 ได้เพียงแบงค์เดียว และเลือกใช้พื้นที่ของรีจิสเตอร์ R0-R7 ในแบงค์ใดๆก็ได้ โดยการกำหนดค่าข้อมูลที่รีจิสเตอร์ PSW ในส่วนของรีจิสเตอร์เฉพาะ (Special Function Register) หากไม่ได้กำหนดค่าใดๆเลย เมื่อทำการรีเซตให้กับ ไอซีไมโครคอนโทรลเลอร์ จะถูกกำหนด ให้เริ่มต้นใช้งานที่รีจิสเดอร์ R0-R7 ในหน่วยความจำตำแหน่งแบงค์ 0 ให้เอง ดังนั้นในการทดลองเริ่มต้นในส่วนแรกๆ เราจะยังไม่

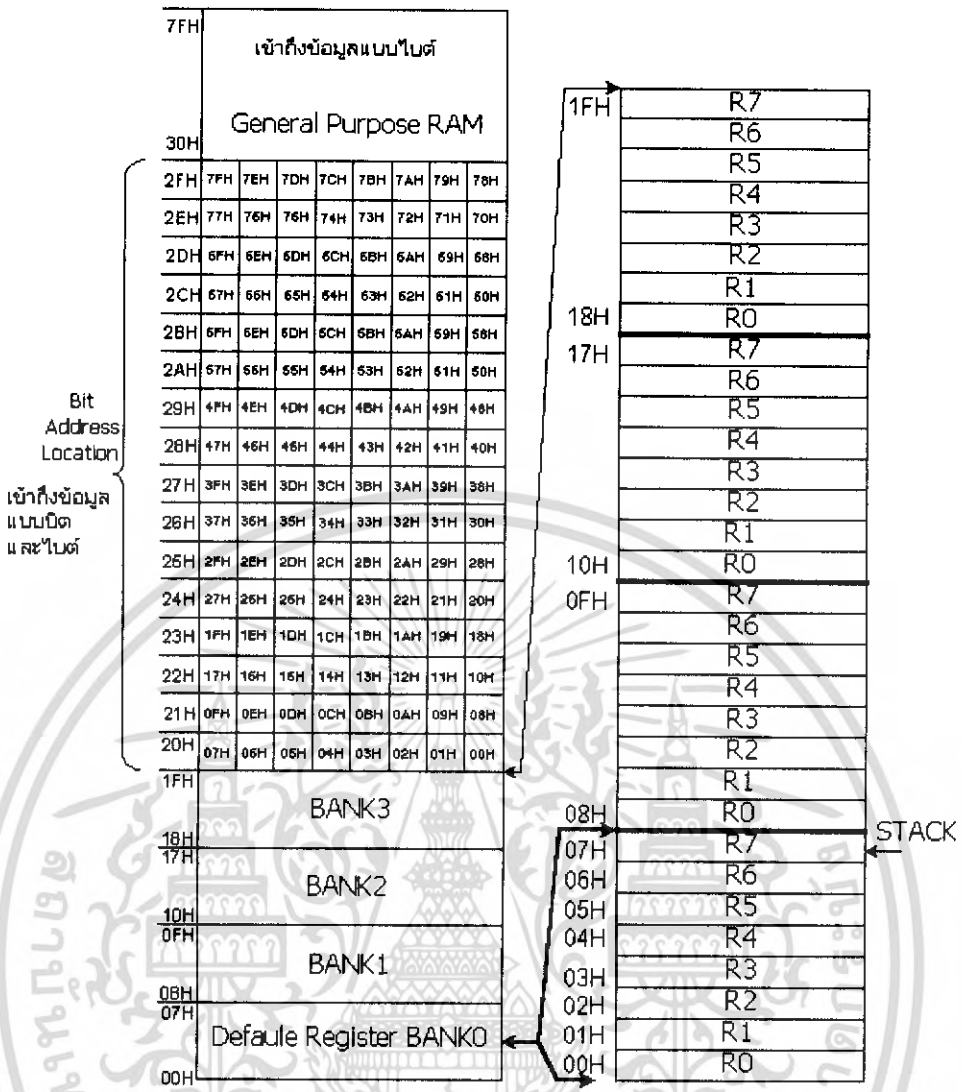
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กำหนดค่าใดๆ ในการเลือกใช้งานรีจิสเตอร์แบงค์อื่นๆ (จะใช้เพียงรีจิสเตอร์ R0-R7ใน หน่วยความจำตำแหน่งแบงค์ 0 ที่ถูกกำหนดมาให้เท่านั้นก่อน)

2. พื้นที่ในหน่วยความจำข้อมูลภายใน (แรม) ตำแหน่งแอดเดรสที่ 20H-2FH จำนวน 16 ไบต์ เป็นส่วนที่สามารถใช้งานในลักษณะการเข้าข้อมูลแบบ ไบต์หรือแบบบิตได้ และสามารถอ้างตำแหน่งแบบบิตได้โดยตรง เพียงแต่ระบุตำแหน่งหรือชื่อของบิตนั้นๆได้ ซึ่งจะมีด้วยกันอยู่จำนวนทั้งหมด 128 บิต แต่ละบิตจะมีหมายเลขตำแหน่งของบิตคือ 00H-7FH โดยตำแหน่งบิตที่ 00H ก็คือข้อมูลของบิตต่ำสุดในตำแหน่งแอดเดรสที่ 20H หรือ เราอาจเรียกว่า(20H.1) และตำแหน่งของบิตที่ 7FH คือข้อมูลบิตสูงสุดในตำแหน่งแอดเดรสที่ 2FH หรือเราอาจเรียกว่า (20H.7) การอ้าง ตำแหน่งแบบบิตจะทำให้โปรแกรมทำงานได้รวดเร็วขึ้น

3. พื้นที่บริเวณหน่วยความจำข้อมูลในตำแหน่งที่ 30H-7FH จะเป็นพื้นที่ของหน่วยความจำใช้งานทั่วไป และการติดต่อกับข้อมูลในตำแหน่งต่างๆ ของหน่วยความจำส่วนนี้จะอ้างตำแหน่งข้อมูลได้ในลักษณะของแบบไบต์เท่านั้น และพื้นที่ส่วนนี้เราอาจจะใช้เป็นสแตกได้ (รายละเอียดของสแตก อยู่ในเรื่องหน่วยความจำแบบ SFR)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

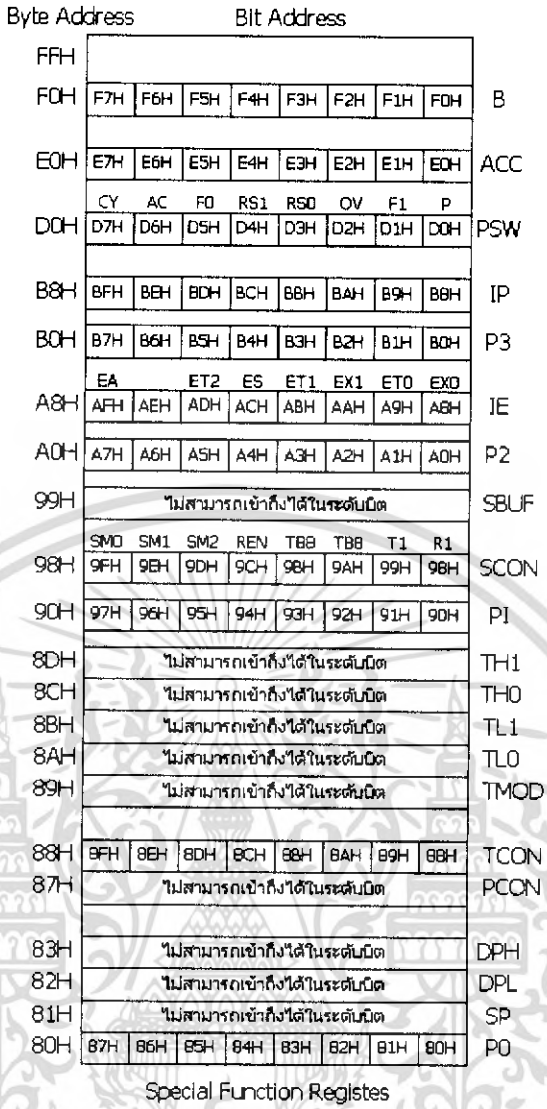


รูปที่ 2.29 พื้นที่บริเวณหน่วยความจำข้อมูลในตำแหน่งที่ 30H-7FH

หน่วยความจำสำหรับเก็บค่ารีจิสเตอร์เฉพาะ (Special Function Register)

รีจิสเตอร์เฉพาะหรือ รีจิสเตอร์พิเศษ (Special Function Register) ที่ใช้ในการควบคุมการทำงานของไมโครคอนโทรลเลอร์ จะอยู่ในหน่วยความจำตำแหน่งแอดเดรสที่ 80H-FFH ซึ่งสามารถจะเรียกใช้ชื่อของรีจิสเตอร์ได้โดยตรง หรืออาจจะเรียกชื่อตามตำแหน่งแอดเดรสก็ได้ รีจิสเตอร์เฉพาะจะประกอบด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.30 การจัดหน่วยความจำและตำแหน่งของรีจิสเตอร์เฉพาะ (Special Function Register)

ACC (Accumulator) หรือ รีจิสเตอร์ A เป็นรีจิสเตอร์ขนาด 8 บิต และมีแอดเดรสอยู่ที่ตำแหน่ง E0H สามารถที่จะเข้าถึงข้อมูล ในระดับบิตได้ ใช้งานเป็นตัวกระทำร่วมทางคณิตศาสตร์ เช่น การบวก ลบ คูณ หาร เป็นต้น และทำหน้าที่เป็นตัวเก็บผลลัพธ์ที่ได้จากการคำนวณ ทางคณิตศาสตร์ ซึ่งรีจิสเตอร์นี้จะถูกใช้งานบ่อยมากในการเขียนโปรแกรม

รีจิสเตอร์ B เป็นรีจิสเตอร์ขนาด 8 บิตที่มีแอดเดรสอยู่ที่ตำแหน่ง F0H ใช้ในการกระทำในคำสั่งคูณหรือหารข้อมูลโดยใช้ร่วมกับรีจิสเตอร์ A จะทำหน้าที่เก็บค่าผลลัพธ์ที่เป็นเศษของการหาร และเก็บผลลัพธ์ของค่าผลคูณไบต์บน และยังใช้เก็บข้อมูลทั่วไปได้ในกรณีไม่ได้ทำคำสั่งในการคูณหรือหาร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PSW: (Program Status Word) เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่แสดงสถานะภาพการทำงานของโปรแกรม ซึ่งจะมีการเปลี่ยนแปลง หลังจากมีการทำงานในคำสั่งต่างๆ และยังใช้เป็นตัวเลือกตำแหน่งแบงก์ของรีจิสเตอร์ (Register Bank) R0-R7 อีกด้วย ผลของบิตต่างๆ สามารถนำไปเป็นเงื่อนไขในการกระโดด (Jump) และค่าของบิตต่างๆ ใน PSW สามารถที่จะเซตหรือเคลียร์บิต ด้วยคำสั่งทางซอฟต์แวร์ได้

แฟลกต่าง ๆ ของรีจิสเตอร์ PSW จะอยู่ในตำแหน่งของบิตต่างๆ ดังนี้

ตารางที่ 2.5 แฟลกต่าง ๆ ของรีจิสเตอร์ PSW

PSW.7	PSW.6	PSW.5	PSW.4	PSW.3	PSW.2	PSW.1	PSW.0
CY	AC	F0	RS1	RS0	OV	-	P

Register	RS0	RS1	ตำแหน่งหน่วยความจำ
BANK 0	0	0	00H - 07H
BANK 1	0	1	07H – 0FH
BANK 2	1	0	10H – 17H
BANK 3	1	1	18H – 1FH

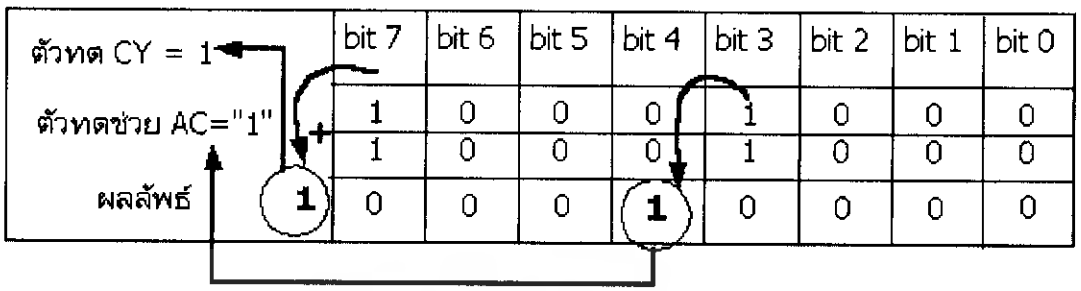
CY: (Carry Flag) เป็นบิตที่ทำหน้าที่แสดงสถานะของตัวทด เช่นในกรณีของการบวกหากนำเลข 8 บิต 2 จำนวนมาบวกกัน แล้วปรากฏว่า ผลบวกที่ได้มีค่ามากเกิน 8 บิต ก็จะทำให้สถานะของบิต CY ถูกเซตเป็น 1 แต่หากผลบวกไม่เกิน 8 บิตป้ายบอกสถานะที่บิต CY จะยังเป็น 0 และในทำนองเดียวกัน จะทำหน้าที่เป็นตัวยืมในกรณีของการลบ ใช้เป็นตัวร่วมกับแอคคิวมูเลเตอร์ (Register A)ในการหมุนบิต และเราสามารถใช้ค่าของ CY เป็นเงื่อนไขในการเขียนโปรแกรมในการกระโดด (Jump) ได้

ตารางที่ 2.6 Carry Flag

ตัวทด CY = 1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	1	0	0	0	0	0	0	0
+	1	0	0	0	1	0	0	0
ผลลัพธ์ 1	0	0	0	0	1	0	0	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC: (Auxiliary Carry Flag) เป็นแฟลกตัวทดช่วยในกรณีที่มีการบวกเลขสองจำนวน แล้วมีการทดระหว่างบิตที่ 3 ไปบิตที่ 4 ทำให้มีการเซตค่าที่บิต AC เป็น "1" ดังตัวอย่าง



F0 : (Flag 0) เป็นแฟลกที่ใช้งานทั่วไปซึ่งเราสามารถใช้เป็นแฟลกสภาวะ (Status flag) ของโปรแกรมโดยการเซต หรือรีเซตด้วยคำสั่ง ทางซอฟต์แวร์(กันไว้สำหรับผู้ใช้) RS1-RS0 : (Register Bank Select)เป็นตัวกำหนดการเลือกพื้นที่ใช้งานของกลุ่มรีจิสเตอร์ R0 - R7 (รูปที่1.10) ในแบงค์ต่างๆโดยการกำหนดสถานะที่บิต RS0 และ RS1 ตามตารางRegister RS0 RS1 ตำแหน่งหน่วยความจำ

OV : (Overflow Flag) เป็นบิตที่แสดงสถานะโอเวอร์โฟลว์ ซึ่งจะถูกเซตหรือเคลียร์จากการทำงานของคำสั่งทางคณิตศาสตร์ แล้วเกิดการทดข้ามจากบิตที่ 6 มายังบิตที่ 7 เช่นในการนำเลขสองจำนวนมารวมกัน แล้วได้ผลลัพธ์มากกว่า +127(ฐานสิบ) หรือต่ำกว่า -128(ฐานสิบ) ในบิตที่ 7 (ซ้ายมือสุด) จะแสดงเป็นบิตสถานะของค่าบวก หรือลบโดยถ้าสถานะเป็น 1 จะเป็นค่าบวก ถ้าสถานะเป็น 0 จะเป็นค่าลบ ดังนั้นเมื่อมีการเกิดโอเวอร์โฟลว์ขึ้น จะทำให้แฟลก OV ถูกเซตเป็น "1"

P : (Parity Flag) เป็นบิตแสดงสถานะที่ใช้ตรวจสอบจำนวนบิตที่เป็น "1" ในข้อมูลของแอคคิวมูเลเตอร์ (Register A) โดยบิต P จะถูกเซตเป็น "1" เมื่อสถานะทั้ง 8 บิตมีเลข 1 เป็นจำนวนคี่ (odd) และบิต P จะถูกเซตเป็น "0" เมื่อสถานะของทั้ง 8 บิตในแอคคิวมูเลเตอร์ (Register A) มีจำนวนเลข 1 เป็นจำนวนคู่ (even) หรือ นับจำนวนเลข 1 ของข้อมูลในรีจิสเตอร์ A ทั้ง 8 บิตนั้นเอง

ตัวชี้สแตก SP (Stack Pointer) เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง 81H ใช้ในการเก็บค่าของตัวชี้บอก ตำแหน่งแอดเดรส เมื่อรีเซตระบบของไมโครคอนโทรลเลอร์ ค่าของตัวชี้สแตกจะถูกกำหนดให้เริ่มต้นชี้ที่ตำแหน่งแอดเดรส 07H (ข้อมูลที่รีจิสเตอร์SP จะมีค่าเท่ากับ 07H) ซึ่งจะเป็นตำแหน่งแอดเดรสของรีจิสเตอร์ R7 ในแบงค์ 0 สแตก(Stack) คือการจองพื้นที่หน่วยความจำข้อมูล บริเวณหนึ่งขึ้นมา เป็นตำแหน่งที่ใช้ในการเก็บข้อมูล โดยให้ รีจิสเตอร์ SP (Stack Pointer) เป็นตัวชี้บอกตำแหน่งแอดเดรสว่าข้อมูลนั้นเก็บไว้ที่ตำแหน่งใดของ หน่วยความจำข้อมูล และหลังจากที่นำข้อมูลไปเก็บไว้ในหน่วยความจำพื้นที่ๆ จองไว้แล้ว ค่าข้อมูลในตัวชี้สแตก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(รีจิสเตอร์ SP) ก็จะชี้ค่าไปยังตำแหน่ งแอดเดรสใหม่ต่อไป (ขึ้นอยู่กับจำนวนของไบต์ข้อมูลที่จะนำไปเก็บ) ดังนั้นถ้าหากจะนำข้อมูลค่าต่อไปจัดเก็บอีก ก็จะเป็นตำแหน่งแอดเดรสที่อยู่ถัดไป

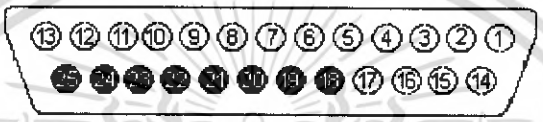
สแตกจะถูกใช้งานในขณะที่มีการเรียกใช้โปรแกรมย่อยโดยคำสั่ง CALL หลังจากนั้นก็จะนำแอดเดรสที่ตำแหน่งถัดจากคำสั่ง CALL (โปรแกรมเคาน์เตอร์) ไปเก็บไว้ที่หน่วยความจำในพื้นที่ๆจองไว้ และหลังจากที่ไมโครคอนโทรลเลอร์ทำงานตามคำสั่งในโปรแกรมย่อยจนเสร็จสิ้นแล้ว โปรแกรมย่อยก็จะปิดท้ายด้วยคำสั่ง RET ดังนั้นไมโครคอนโทรลเลอร์จะกลับไปที่ตำแหน่งเดิมได้ ก็จะต้องไปดูข้อมูลที่ตัวชี้สแตก (Stack Pointer) ว่าชี้อยู่ที่ตำแหน่งแอดเดรสใด ซึ่งค่าข้อมูลของสแตกในตำแหน่งนั้นนั้น ก็คือแอดเดรสที่โปรแกรมจะต้องกลับไปทำงานต่อ ดังนั้นการเรียกโปรแกรมย่อย ซ้อนกันหลายๆครั้ง ค่าในสแตกก็จะซ้อนทับกัน การที่จะกลับค่าไปที่ตำแหน่งแอดเดรสเดิมได้ ก็ต้องออกจากสแตกที่ละชั้นไป เราจึงเรียกวิธีการนี้ว่า FILO (First In Last Out) หรือเข้าก่อนออกที่หลัง ในส่วนของสแตกยังสามารถเก็บข้อมูลของรีจิสเตอร์ต่างๆได้ด้วย เช่นกรณีไมโครคอนโทรลเลอร์ กระโดดไปทำที่โปรแกรมย่อย และในส่วนของโปรแกรมย่อยมีคำสั่งโดยเรียกใช้รีจิสเตอร์ที่เราใช้ในโปรแกรมหลักด้วย เช่น รีจิสเตอร์ PSW รีจิสเตอร์ A หรือ รีจิสเตอร์ R0 ฯลฯ ซึ่งถ้าหากเราไม่เก็บค่าข้อมูลเดิมของรีจิสเตอร์ไว้ก่อน ค่าข้อมูลในรีจิสเตอร์ดังกล่าวอาจจะเปลี่ยนแปลงข้อมูลไป ดังนั้นจึงต้องนำข้อมูลเดิมไปเก็บไว้ที่สแตกก่อนด้วยคำสั่ง PUSH หลังจากนั้นเมื่อออกจากโปรแกรมย่อยจึงจะทำการคืนค่าของรีจิสเตอร์ในสแตกกลับมา ให้เป็นข้อมูลเดิมโดยคำสั่ง POP (รายละเอียดอยู่ในบทที่ 4 เรื่องโปรแกรมย่อยและการกระโดด)ทุกครั้งที่เริ่มรีเซตระบบ ไอซีไมโครคอนโทรลเลอร์ ก็จะเริ่มทำงานที่ตำแหน่งแอดเดรส 0000H ของหน่วยความจำโปรแกรมแฟลช (Flash memory) และจะเริ่มปฏิบัติตามคำสั่งที่ถูกเขียน เป็นข้อมูลขนาด 8 บิต (1ไบต์)ในแต่ละแอดเดรสของหน่วยความจำ โปรแกรม ซึ่งบางคำสั่งอาจจะใช้เนื้อที่ในการเก็บข้อมูลเพียง 1 แอดเดรส และบางคำสั่งอาจจะต้องใช้เนื้อที่ 2 -3 แอดเดรส จำนวนของแอดเดรสที่จะเก็บข้อมูลในแต่ละคำสั่ง สามารถเปิดดูได้จากตารางคำสั่ง ของไมโครคอนโทรลเลอร์ (บทที่ 4 หัวข้อที่ 4.2.5) ดังนั้นไอซีไมโครคอนโทรลเลอร์ที่มีหน่วยความจำโปรแกรมขนาด 1Kbytes ก็อุปมาเหมือนกับมีกระดาษไว้เขียนคำสั่ง 1,024 บรรทัด หากมีการเขียนคำสั่งที่ยาวกว่า 1024 บรรทัด แต่ละบรรทัดมีขนาด 8 บิด (1ไบต์) ก็จะต้องเลือกไอซีที่มีหน่วยความจำโปรแกรมให้มากกว่านี้

DPTR: (Data Pointer) เป็นรีจิสเตอร์ขนาด 16 บิตที่ประกอบด้วยรีจิสเตอร์ขนาด 8 บิต 2 ตัวคือรีจิสเตอร์ DPL และ DPH ซึ่งเราสามารถเลือกการใช้งานในลักษณะ 8 บิด 2 ตัวหรือ 16 บิต 1 ตัวก็ได้ จะมีแอดเดรสอยู่ที่ตำแหน่ง 82H,83H ตามลำดับ ใช้สำหรับเป็นตัวชี้ ตำแหน่งของหน่วยความจำหรือตำแหน่งของอุปกรณ์อินพุตเอาต์พุตที่ไมโครคอนโทรลเลอร์ต้องการติดต่อด้วย และใช้เป็นตัวกำหนดตำแหน่งเริ่มต้น (Base) ของตารางในการทำงานเกี่ยวกับ Look up table (รายละเอียดและตัวอย่างการใช้งานอยู่ในบทที่ 5 การทดลองเรื่อง Look up table)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

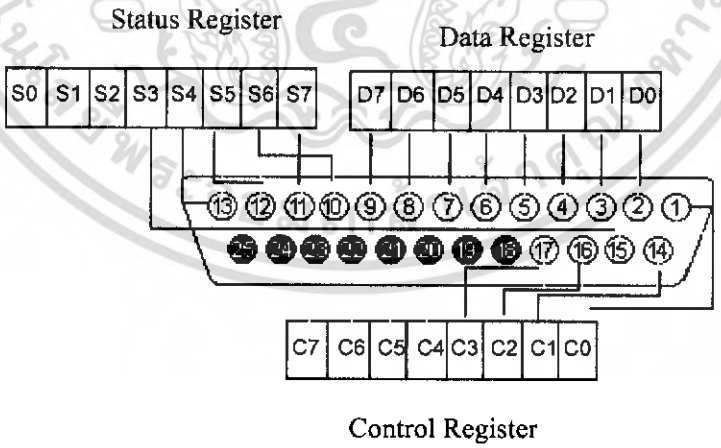
รีจิสเตอร์พอร์ต P0-P3 พอร์ต P1และพอร์ต P3 เป็นรีจิสเตอร์ขนาด 8 บิต ซึ่งพอร์ต P0 มีแอดเดรสอยู่ที่ตำแหน่ง 80H พอร์ต P1 มีแอดเดรสอยู่ที่ตำแหน่ง 90H พอร์ต P2มีแอดเดรสอยู่ที่ตำแหน่ง A0H และพอร์ต P3มีแอดเดรสอยู่ที่ตำแหน่ง B0H ข้อมูลที่อยู่ในรีจิสเตอร์ เหล่านี้จะเป็นค่าเดียวกับค่าของสัญญาณที่ขาต่างๆของพอร์ต เป็นรีจีสเตอร์ที่สามารถเข้าถึงข้อมูลแบบบิตได้ เป็นพอร์ตแบบสองทิศทางคือเป็นได้ทั้ง พอร์ตอินพุต และพอร์ตเอาต์พุต สามารถคงสภาวะ(Latch) ขณะเป็นเอาต์พุต สำหรับเบอร์ AT89CX051 พอร์ต P3 จะมีขาภายนอกเพียง 7 ขา ส่วนขาที่ P3.6 จะเป็นขาที่อยู่ภายใน ไอซีเป็นขาเอาต์พุตของวงจรเปรียบเทียบระหว่างขา P1.0 และ P1.1

2.6 พอร์ตขนาน (Parallel port)



รูปที่ 2.31 พอร์ตขนาน

พอร์ตขนาน เดิมเรียกว่า Printer Port เนื่องจากจะมีการนำพอร์ตขนาน มาใช้งานติดต่อกับเครื่องพรินเตอร์เป็นหลักโดยที่พอร์ตขนานนั้น สามารถให้ความเร็วในการส่งผ่านข้อมูลได้รวดเร็วกว่าพอร์ตอนุกรม และยังสามารถส่งข้อมูลขนาน 8 บิตออกไปได้โดยตรง



รูปที่ 2.32 การส่งข้อมูลของพอร์ตขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พอร์ตขนานของเครื่องคอมพิวเตอร์ประกอบด้วยสัญญาณทั้งหมด 25 เส้นสัญญาณ (DB25) โดยสัญญาณจะแบ่งออกได้เป็น 3 กลุ่มใหญ่ๆ ตามลักษณะหน้าที่ของสัญญาณ ประกอบด้วย

1) Data Register (Data bus) จำนวน 8 เส้น

2) Control Register จำนวน 4 เส้น

3) Status Register จำนวน 5 เส้น

ตารางที่ 2.7 ขาของพอร์ตขนาน

Pin No (DB25)	Signal name	Direction	Register - bit	Inverted
1	nStrobe	Out	Control-0	Yes
2	Data0	In/Out	Data-0	No
3	Data1	In/Out	Data-1	No
4	Data2	In/Out	Data-2	No
5	Data3	In/Out	Data-3	No
6	Data4	In/Out	Data-4	No
7	Data5	In/Out	Data-5	No
8	Data6	In/Out	Data-6	No
9	Data7	In/Out	Data-7	No
10	nAck	In	Status-6	No
11	Busy	In	Status-7	Yes
12	Paper-Out	In	Status-5	No
13	Select	In	Status-4	No
14	Linefeed	Out	Control-1	Yes
15	nError	In	Status-3	No
16	nInitialize	Out	Control-2	No
17	nSelect-Printer	Out	Control-3	Yes
18-25	Ground	-	-	-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Register (Data bus) จะมีอยู่ 8 ขา ตั้งแต่ขาที่ 2 ถึงขาที่ 9 บางทีมักถูกเรียกว่า Data Register ซึ่ง Register ตัวนี้จะส่งค่าได้อย่างเดียว ไม่สามารถรับค่าได้

Status Register เป็นพอร์ตที่อ่านได้อย่างเดียวไม่สามารถเขียนข้อมูลได้ พอร์ตนี้จะมีสัญญาณเข้าอยู่ 5 สัญญาณ และสัญญาณ IRQ กับสัญญาณสงวนไว้อีกสองบิต โดยสัญญาณ Busy จะ Active Low

ลักษณะการทำงานของแต่ละบิตใน Status Register มีดังนี้

- Bit7 Busy เมื่อ Active หมายถึงพรินเตอร์จะไม่รับข้อมูล
- Bit6 nAck เมื่อ Active หมายถึงพรินเตอร์พร้อมที่จะทำงาน (Active Low)
- Bit5 Paper End เมื่อ Active หมายถึงพรินเตอร์ไม่มีกระดาษ
- Bit4 Select เมื่อ Active หมายถึงเลือกพรินเตอร์
- Bit3 nError เมื่อ Active หมายถึงพรินเตอร์เกิดข้อผิดพลาด (Active Low)
- Bit2, Bit1, Bit0 ไม่ใช้

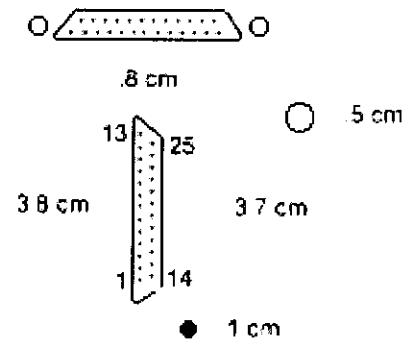
Control Register เป็นพอร์ตที่ใช้ในการควบคุมพรินเตอร์ สัญญาณในกลุ่มนี้จะ Active Low ยกเว้นสัญญาณ Initialize เท่านั้นที่ไม่ถูก Invert ลักษณะการทำงานของแต่ละบิตใน Control Register มีดังนี้

- Bit3 nSelect Printer เมื่อ Active หมายถึงเลือกพรินเตอร์
- Bit2 nInitialize เมื่อ Active หมายถึงรีเซตพรินเตอร์
- Bit1 nAuto Feed เมื่อ Active หมายถึงพรินเตอร์กระทำ Line Feed
- Bit0 nStrobe เมื่อ Active หมายถึงการบอกให้พรินเตอร์ทราบว่ามีข้อมูลเข้ามาแล้ว

ตาราง 2.8 Register Address

Register Address		
Register	LPT1	LPT2
Data registar(baseaddress + 0)	0x378	0x278
Dtatus register (baseaddress + 1)	0x379	0x279
Control register (baseaddress + 2)	0x37a	0x27a

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

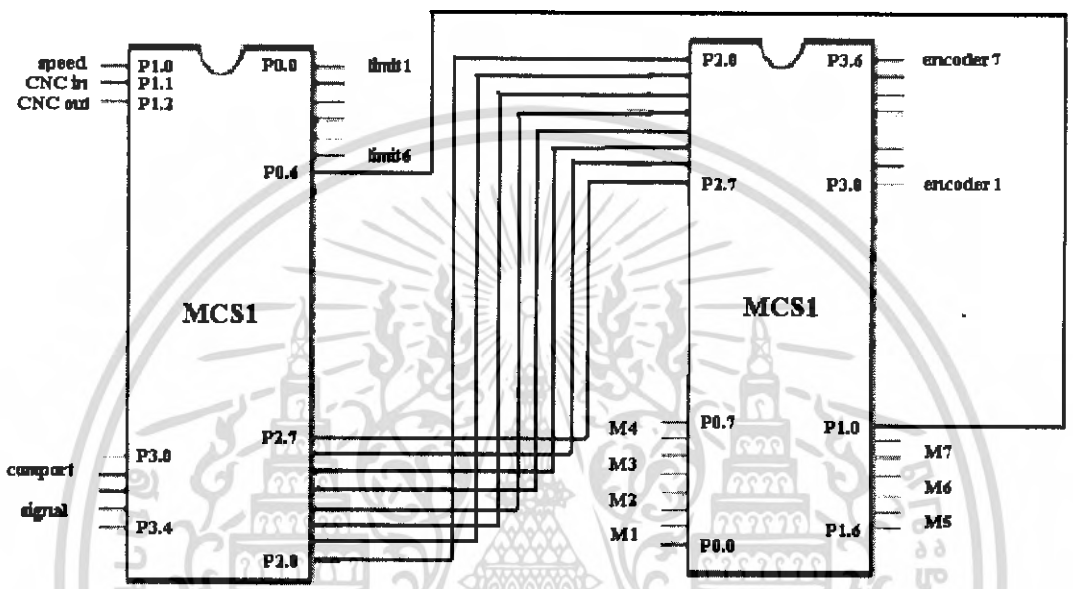


รูปที่ 2.33 Lay out

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
โครงสร้างทางฮาร์ดแวร์

3.1 วงจร MCS-51



รูปที่ 3.1 วงจร MCS-51

วงจร MCS-51 มีหน้าที่ติดต่อกับคอมพิวเตอร์ เพื่อรับคำสั่งมาทำงาน รับสัญญาณจาก Limit switches และ เครื่อง CNC แล้วนำมาประมวลผลเพื่อสร้างสัญญาณควบคุมหุ่นยนต์ เนื่องจากสัญญาณที่ติดต่อกับ MCS-51 มีหลายสัญญาณ ทำให้ต้องใช้ MCS-51 อีกตัวมาเพื่อการเขียนโปรแกรมทำได้ง่ายและไม่ซับซ้อน

MCS 1

P0.0 – P0.5	ติดต่อกับ Limit switch 6 ตัว
P0.6	ติดต่อกับ P1.6 ของ MCS 2
P1.0	รับสัญญาณควบคุมความเร็ว
P1.1 – P1.2	ติดต่อกับ CNC
P2.0 – P2.7	ติดต่อกับ MCS 2
P3.0 – P3.4	ติดต่อกับ computer

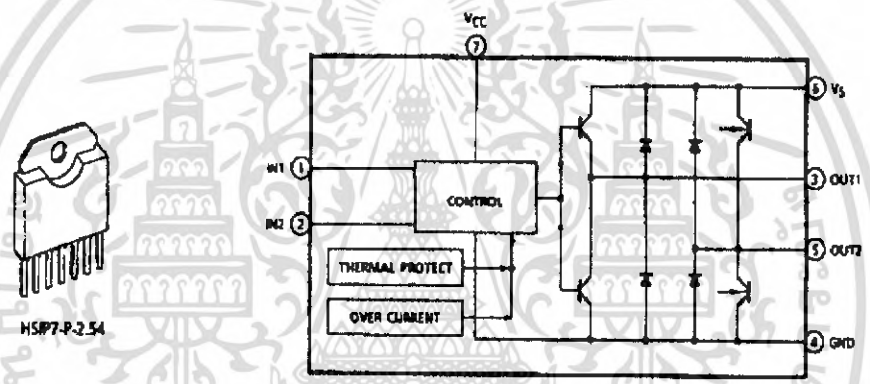
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS 2

P0.0 – P0.7	ติดต่อกับ Motor 1 – 4
P1.0 – P1.5	ติดต่อกับ Motor 5 – 7
P1.6	ติดต่อกับ P0.6 ของ MCS 1
P2.0 – P2.7	ติดต่อกับ MCS 1
P3.0 – P3.6	ติดต่อกับ Encoder 7 ตัว

3.2 วงจรขับมอเตอร์ (Drive Motor Circuit)

เป็นส่วนทำหน้าที่ในการส่งสัญญาณเอาท์พุตออกไปควบคุมการเคลื่อนที่ของแขนกล โดยในที่นี้เลือกใช้ IC เบอร์ TA7257P ซึ่งมีโครงสร้างภายในดังรูปที่ 3.2



รูปที่ 3.2 โครงสร้างภายใน TA7257P

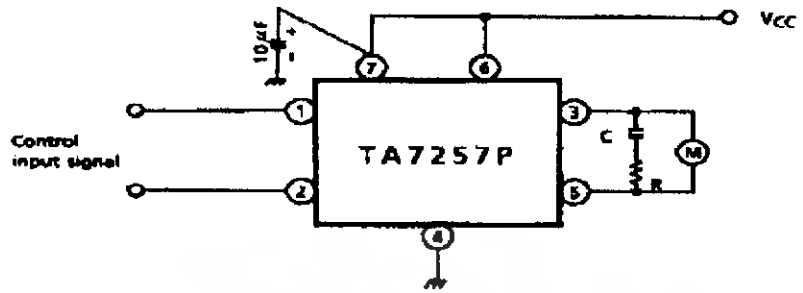
IC เบอร์ TA7257P สั่งงานด้วยลอจิก "0" และ "1" (5V และ 0V) จากขาเอาท์พุตของไมโครคอนโทรลเลอร์เข้าขา IN1, 2 ซึ่ง TA7257P 1 ตัว ใช้ 2 บิตในการสั่งงานมอเตอร์ 1 ตัว ทำงานได้ 4 โหมด ดังตารางที่ 3.1

ตารางที่ 3.1 แสดงอินพุตและเอาท์พุตของ IC เบอร์ TA7257P

IN1	IN2	OUT1	OUT2	MODE
1	1	L	L	BRAKE
0	1	L	H	CW/CCW
1	0	H	L	CCW/CW
0	0	HIGH IMPEDANCE		STOP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนการต่อวงจร แสดงได้ดังรูปที่ 3.3

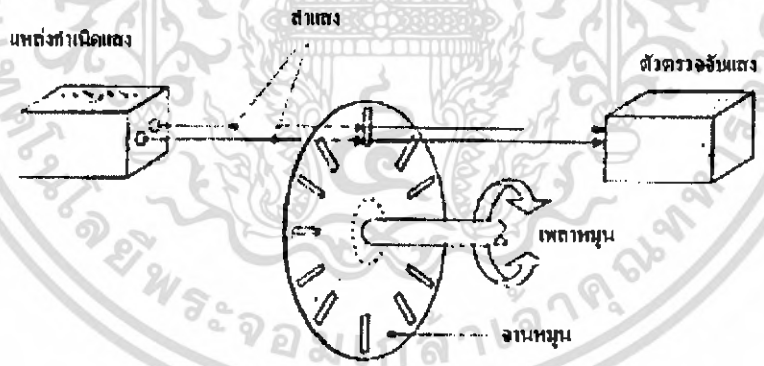


รูปที่ 3.3 การเชื่อมต่อ IC เบอร์ TA7257P กับอุปกรณ์พื้นฐานภายนอก

3.3 วงจรเอนโคดเดอร์ (Encoder Circuit)

เป็นส่วนที่ทำหน้าที่ในการตรวจสอบสัญญาณที่ออกมาจากเอนโคดเดอร์ของมอเตอร์แต่ละตัว เพื่อนำสัญญาณที่ได้ไปเป็นอินพุตในการตรวจสอบค่าตำแหน่งการเคลื่อนที่ของแขนกล

วงจรเอนโคดเดอร์ที่ใช้ร่วมกันกับมอเตอร์ของแขนกลนี้ เป็นแบบออพติคอลเอนโคดเดอร์มีลักษณะดังรูปที่ 3.4



รูปที่ 3.4 เอนโคดเดอร์

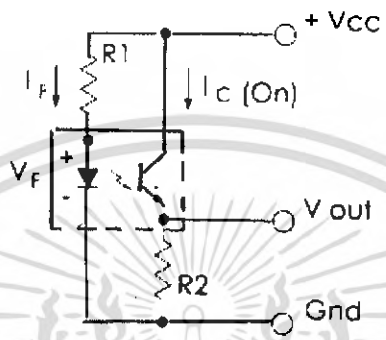
ส่วนประกอบที่สำคัญของออพติคอลเอนโคดเดอร์

1. แหล่งกำเนิดแสง (Source) ในที่นี้ใช้หลอดอินฟราเรด
2. ตัวตรวจจับแสง (Phototransistor)
3. จานหมุน มอเตอร์ที่ตัวมือจับใช้ชนิด 3 ช่อง ส่วนที่เหลืออีก 5 ตัวใช้ชนิด 6 ช่อง
4. วงจรตรวจจับสัญญาณ (Phototransistor circuit) ใช้วงจร common collector ซึ่งมีโหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานอยู่ 2 ชนิด คือ

1. สวิตซ์โหมด (Switch mode) จะให้เอาท์พุตเป็นค่า on หรือ off ออกมาเมื่อมีการตรวจจับสัญญาณ
2. แอกทีฟโหมด (Active mode) จะให้ค่าเอาท์พุตที่ตอบสนองขึ้นอยู่กับแสงหรือระดับการส่องสว่าง โดยที่ I_C จะเป็นสัดส่วนโดยตรงกับความเข้มแสง



รูปที่ 3.5 วงจรเอนโคดเดอร์

$$R_1 = (V_{CC}-V_F)/I_F \quad (3.1)$$

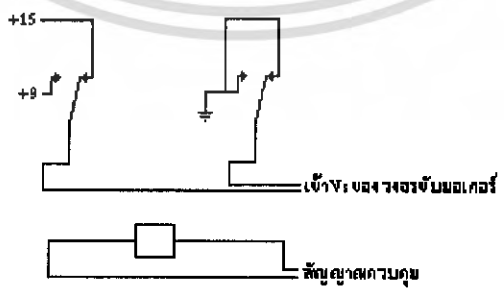
$$R_2 = (V_{CC}-V_{CE(sat)})/I_{C(on)} \quad (3.2)$$

ดังนั้น จะใช้

$$R_1 = 220\Omega$$

$$R_2 = 500k\Omega$$

3.4 วงจรปรับความเร็ว

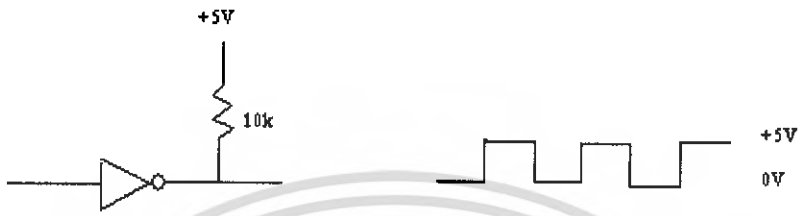


รูปที่ 3.6 วงจรปรับความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วงจรปรับแต่งสัญญาณ

ใช้ IC เบอร์ 7404 เป็นตัวปรับแต่งสัญญาณ เป็นอินเวอเตอร์ (inverter) ขยายสัญญาณอนาลอก (analog) แบบโวลต์ต่ำ และรูปแบบสัญญาณไม่เสถียรให้ได้ 5V เป็น ลอจิก high "1" และ 0V เป็น ลอจิก low "0" เพื่อให้ตัวคอนโทรลเลอร์รับรู้สัญญาณ



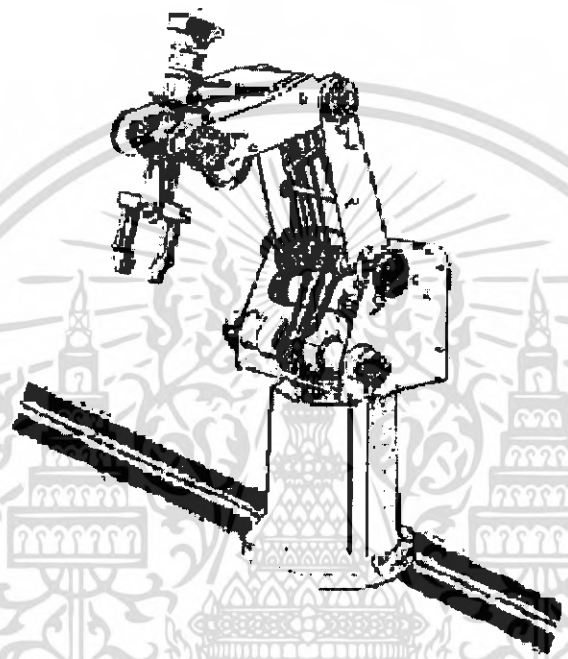
รูปที่ 3.7 วงจรปรับแต่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
โครงสร้างการทำงาน

4.1 โครงสร้างของแขนกล

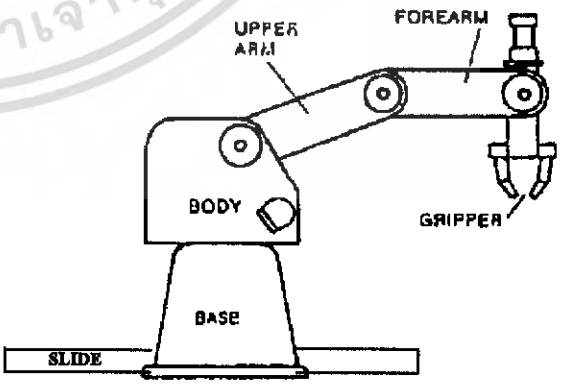
แขนกลที่ใช้เป็นของบริษัท ESHED ROBOTEC Ltd. รุ่น SCORBOT-ER V plus.



รูปที่ 4.1 SCORBOT-ER V plus

ประกอบด้วยชิ้นส่วนหลัก

1. Base
2. Body
3. Upper arm
4. Forearm
5. Gripper
6. Slide
7. มอเตอร์ 12VDC 7 ตัว
8. เอนโคดเดอร์ 7 ตัว
9. ไมโครสวิตช์ 5 ตัว
10. ส่วนส่งผ่านการเคลื่อนที่ เช่น เฟือง สายพานต่างๆ



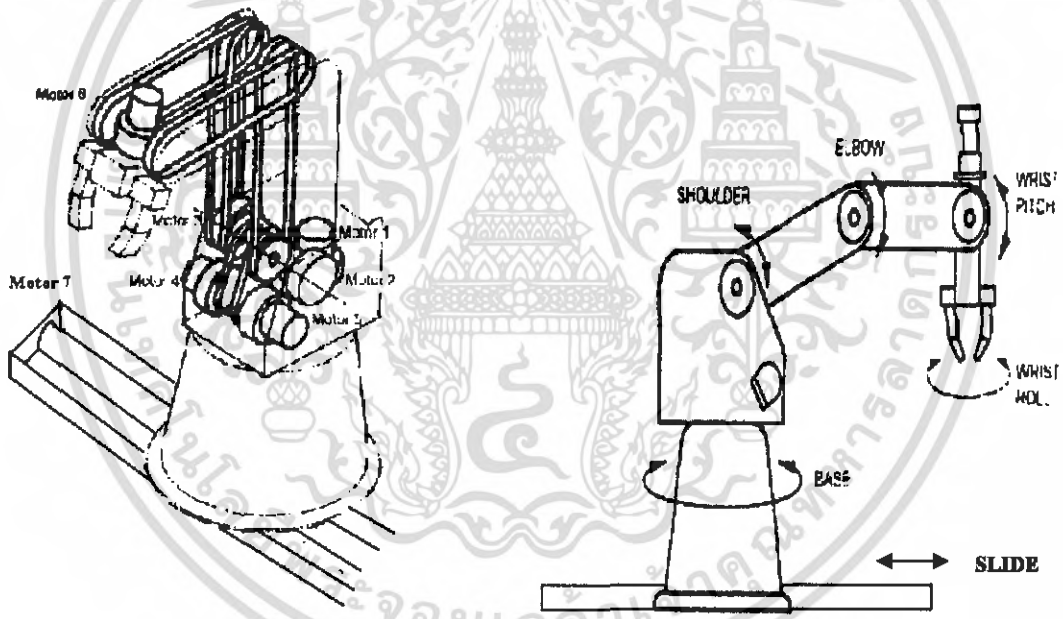
รูปที่ 4.2 ชิ้นส่วนแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ทิศทางการเคลื่อนที่ของโครงสร้างแขนกลแต่ละข้อต่อและตำแหน่งมอเตอร์

Axis No.	Joint Name	Motion	Motor No.
1	Base	Rotates the body.	1
2	Shoulder	Raises and lowers the uppers arm.	2
3	Elbow	Raises and lowers the forearm.	3
4	Wrist Pitch	Raises and lowers the end effector (gripper).	4+5
5	Wrist Roll	Rotates the end effector (gripper).	4+5

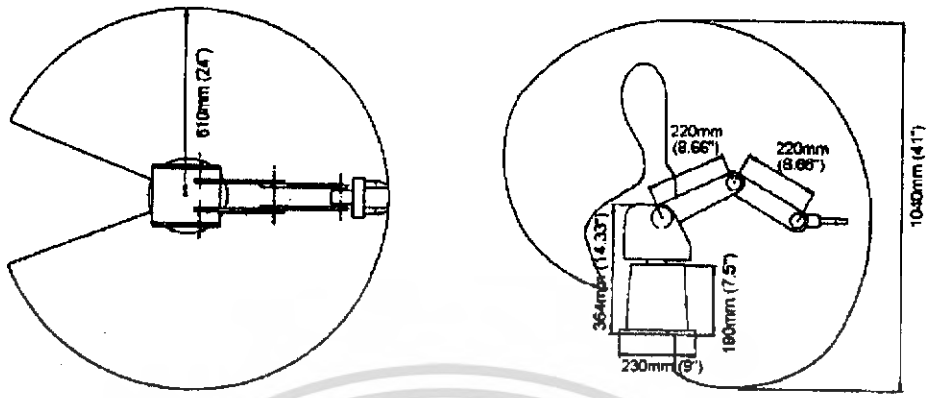
นอกจากนี้ยังมีมอเตอร์ No.6 ใช้ในการหนีบจับวัตถุของ Gripper และมอเตอร์ No.7 ใช้เป็น Slid เลื่อนหุ่นยนต์



รูปที่ 4.3 ทิศทางการเคลื่อนที่ของโครงสร้างแขนกล

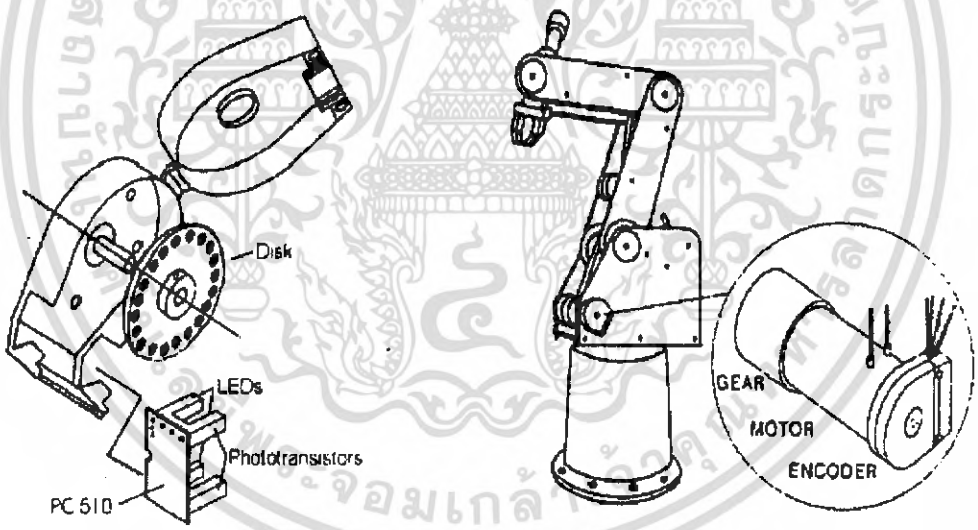
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตพื้นที่การทำงานของแขนกล



รูปที่ 4.4 ขอบเขตพื้นที่การทำงานของแขนกล

เอนโคดเดอร์และสายไฟเชื่อมต่อ



รูปที่ 4.5 เอนโคดเดอร์และสายไฟเชื่อมต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 การเชื่อมต่อสายไฟจากมอเตอร์, เอนโคดเดอร์และไมโครสวิตช์ออกภายนอก

SCORBOT-ER Vplus Wiring							
	Robot Arm Signal			Lead to Molex 12-pin Connector		Lead to D50 Connector	
Axis	Motor	Encoder	Microswitch	Color	Pin#	Color	Pin#
1	+					White	50
	-					Gray/Green	17
2	+					White	49
	-					White/Green	16
3	+					White	48
	-					Orange/Brown	15
4	+					White	47
	-					Orange/Green	14
5	+					White	46
	-					Orange/Grey	13
Gripper	+			Gray	8	White	45
	-			Yellow	7	Orange/Blue	12
Slide	+					Green	
	-					Red	
1		GND 1				White	33*
		P_I 3				White/Grey	5
		V_{LED} 2				Yellow	11
		P_0 4				Brown	2
2		GND 1				White	32*
		P_I 3				White/Orange	21
		V_{LED} 2				Yellow	27
		P_0 4				Grey	1
3		GND 1				White	31*
		P_I 3				Brown/Blue	4
		V_{LED} 2				Yellow	10
		P_0 4				Green	36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

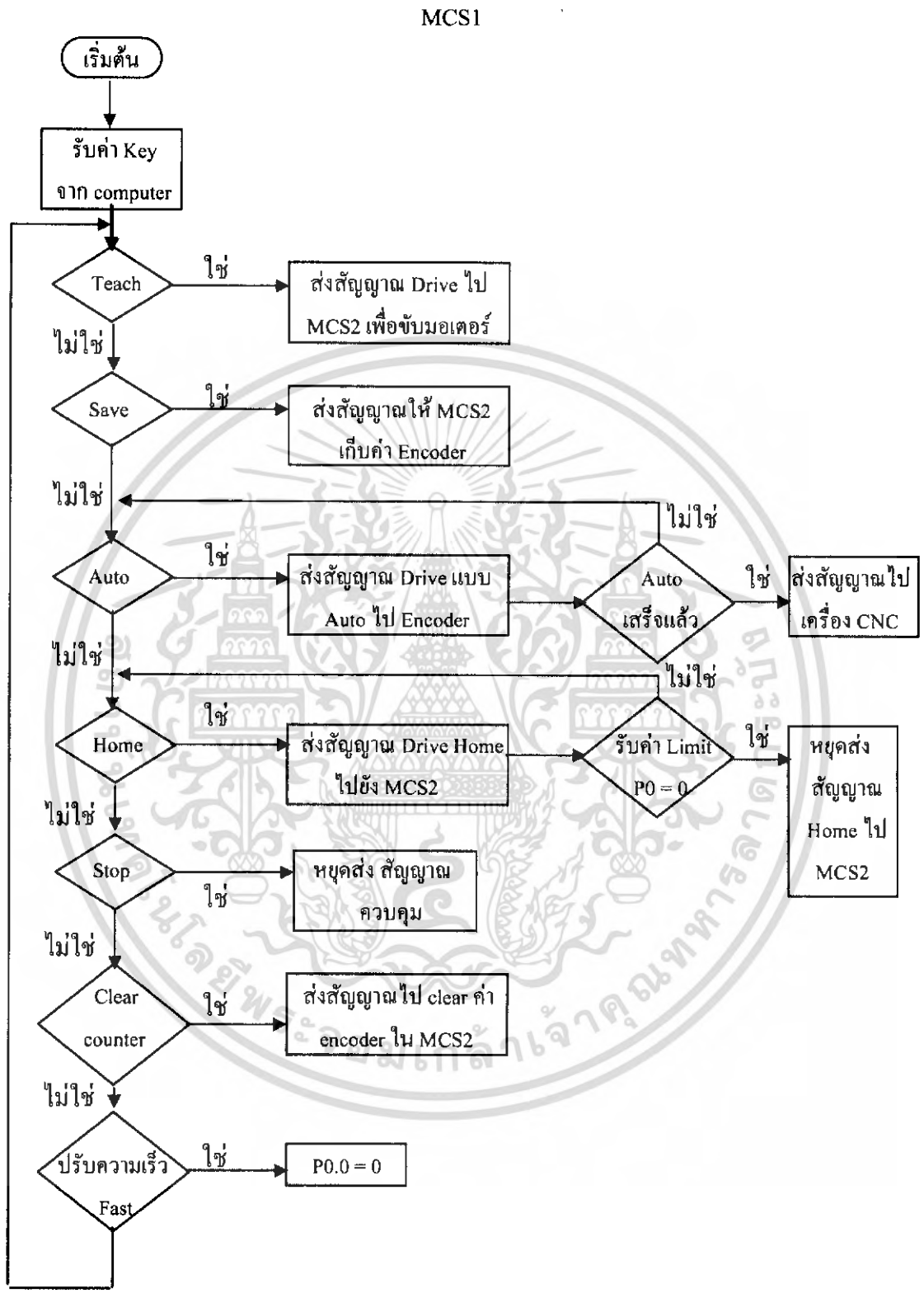
SCORBOT-ER Vplus Wiring							
	Robot Arm Signal			Lead to Molex 12-pin Connector		Lead to D50 Connector	
Axis	Motor	Encoder	Microswitch	Color	Pin#	Color	Pin#
4		GND 1				White	30*
		P1 3				Green/Brown	20
		VLED 2				Yellow	26
		P0 4				Orange	35
5		GND 1				White	29*
		P1 3				Green/Blue	3
		VLED 2				Yellow	9
		P0 4				Blue	18
Gripper		GND 1		Black	12	White	28*
		P1 3		Green	11	Grey/Blue	19
		VLED 2		Yellow	10	White	25
		P0 4		Brown	9	White/Blue	34
Slide		GND 1				Black	
		P1 3				White	
		VLED 2				Brown	
		P0 4					
1			GND			White	33*
			MS			Brown	23
2			GND			White	32*
			MS			Grey	7
3			GND	White	1	White	31*
			MS	White	2	Orange	24
4			GND	Blue	3	White	30*
			MS	Blue	4	green	8
5			GND	Orange	5	White	29*
			MS	Orange	6	Blue	6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SCORBOT-ER Vplus Wiring							
	Robot Arm Signal			Lead to Molex 12-pin Connector		Lead to D50 Connector	
Axis	Motor	Encoder	Microswitch	Color	Pin#	Color	Pin#
Gripper			No connection			White Brown/Grey	28* 22
Slide			GND MS			Black Red	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

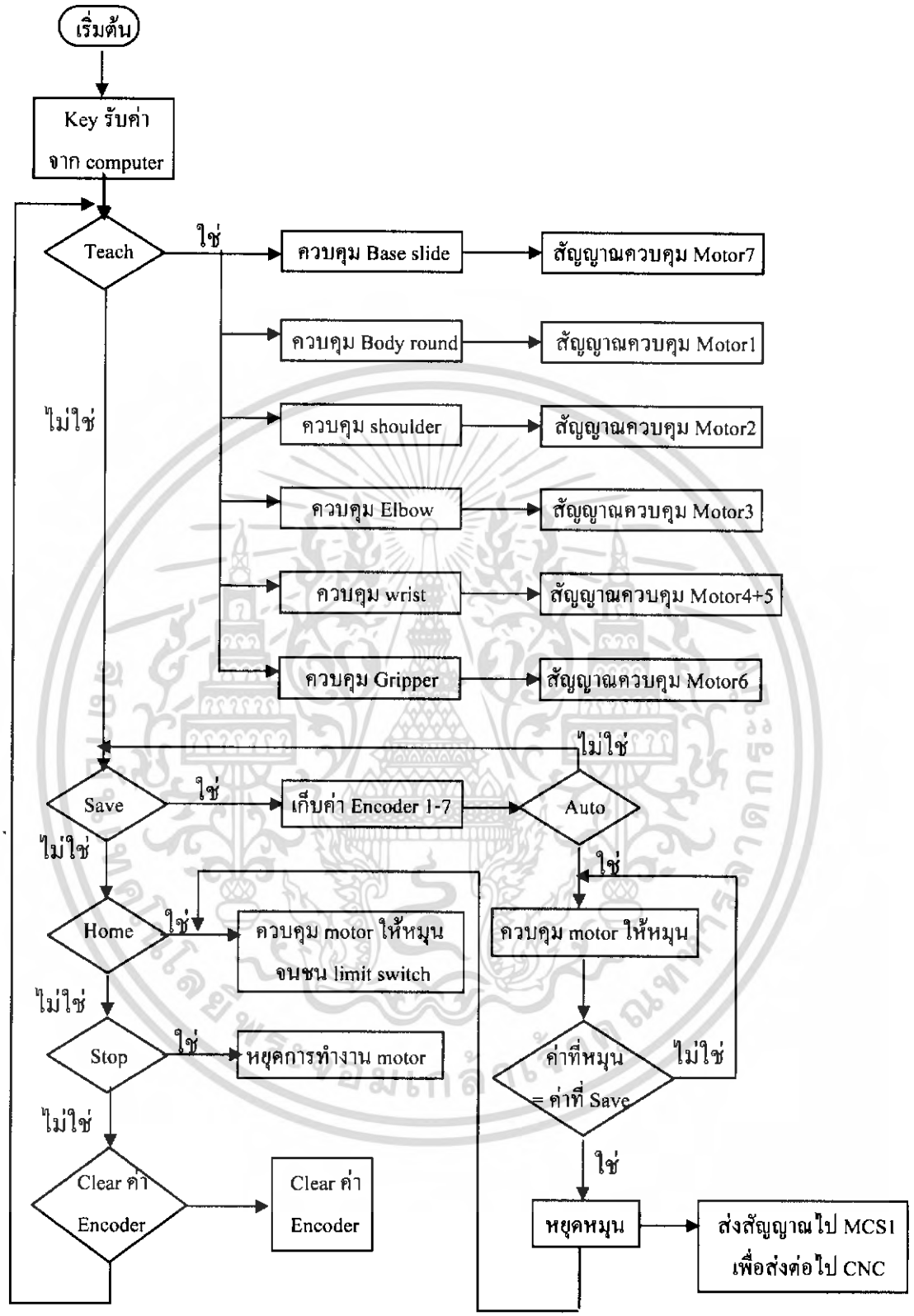
4.2 Flow chart แสดงการทำงานของโปรแกรม



รูปที่ 4.6 flow chart การทำงานของ MCS1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS2

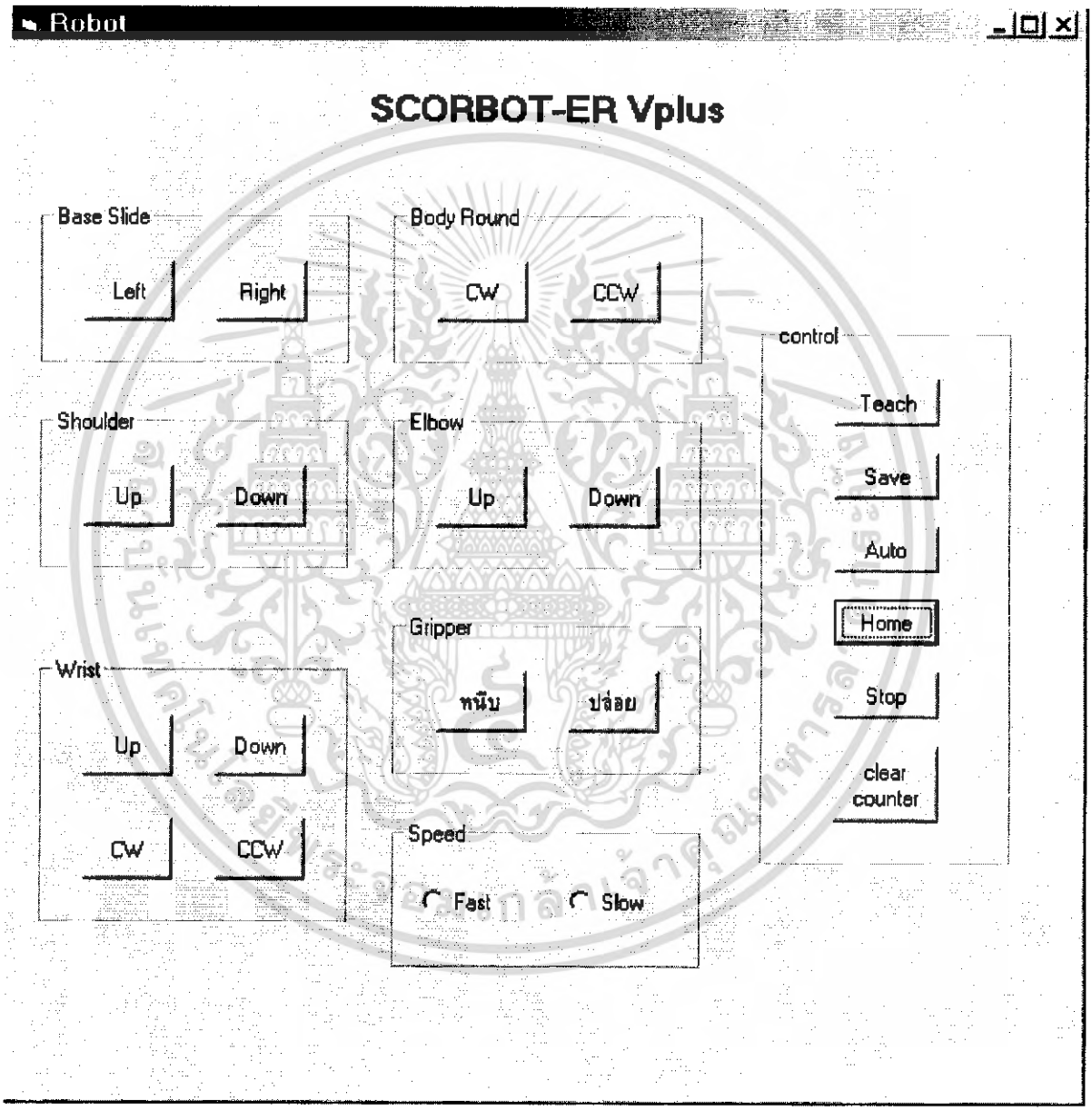


รูปที่ 4.7 flow chart การทำงานของ MCS2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ข้อสรุปและข้อเสนอแนะ

5.1 ข้อสรุป



รูปที่ 5.1 ปุ่มควบคุมบนหน้าจอคอมพิวเตอร์

1. เมื่อต้องการขับเคลื่อนแขนกลให้กด Teach แล้วเลือก action ตามต้องการ
2. เมื่อกด Save จะเก็บค่าการเคลื่อนที่แต่ละครั้งของการขับเคลื่อนทีละแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เมื่อกด Auto จะทำการ Run Auto จนค่าการเคลื่อนที่เท่ากับค่า Encoder ที่เก็บไว้แล้วจึงหยุด
4. เมื่อกด Stop จะทำการหยุดการขับเคลื่อนของแขนกล ทุก action ที่กระทำอยู่
5. เมื่อกด Clear counter จะลบค่า Encoder ที่เก็บไว้ทั้งหมด ให้เริ่มค่าใหม่ที่ 0
6. เมื่อกด Fast หุ่นจะเคลื่อนที่เร็วขึ้น
7. เมื่อกด Slow หุ่นจะเคลื่อนที่ช้าลง

* ก่อนการทำ Home ทุกครั้ง ต้องให้หุ่นทุกข้อไม่ได้อยู่ที่ค่า Home มิฉะนั้นจะเกิดการ error *

5.2 ข้อเสนอแนะ

1. มอเตอร์ที่ควบคุมแขนส่วนที่รับน้ำหนักไม่สามารถ break ตัวเอง เนื่องจากแรงโน้มถ่วงได้ อาจสามารถปรับปรุงวงจรมอเตอร์ให้มีความเหนี่ยวนำในตัว และ break ตัวเอง เนื่องจากแรงโน้มถ่วงขณะหยุดแขนได้

2. วิธีปรับปรุงความเร็วของมอเตอร์สามารถพัฒนาได้ดีกว่านี้ โดยวิธีอื่น เช่น PWM (Pulse Width Modulation)

3. ในงานที่ต้องการความแม่นยำสูง สามารถควบคุมตำแหน่งได้ละเอียดกว่านี้โดยใช้เซอร์โวมอเตอร์

4. สามารถปรับปรุงพัฒนาให้เครื่อง CNC และหุ่นยนต์อุตสาหกรรมนี้ มี program ใน application เดียวกันได้ โดยศึกษาวิธีใช้โปรแกรมการเขียน application นี้ ให้มีการแยกเอาท์พุตออกที่ port out

5. สามารถพัฒนาให้แขนกลกลับ Home ได้ แม้มีแขนข้อใดข้อหนึ่งชนลิมิตสวิตซ์ (limit switch) อยู่แล้ว

6. สามารถเปลี่ยนคอนโทรลเลอร์ให้มีหน่วยความจำเพิ่มมากขึ้น ทำให้ Save ค่า action ได้หลายค่า และลด error

5.3 ปัญหาที่พบ

1. มอเตอร์ตัวที่ 4 และตัวที่ 5 ซึ่งต้องทำงานร่วมกันโดยขับเคลื่อนการหมุนและขึ้นลงของมือจับ มีทอล์กไม่เท่ากัน เมื่อมีการทำงานร่วมกันทำให้ค่าตำแหน่งมีความผิดพลาดได้

2. การ Save ค่าเอนโคดเดอร์ ถ้าทำผิดขั้นตอน เมื่อ run auto จะเกิด error ได้

3. ถ้าเก็บค่าเอนโคดเดอร์หลายๆ action เมื่อ run auto อาจเกิด error ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

David G.Alciatore, Michael B. Histand. Introduction to Mechatronics and Measurement systems. Department of Mechanical Engineering Colorado State University : 2003.

ชัยวัฒน์ ลิ้มพรจิตรวิไล, นคร ภักดีชาติ, ธีรบูลย์ หล่อวิเชียรรุ่ง. ปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษาซี. กรุงเทพมหานคร.

ผศ.ดร. วรพงศ์ ตั้งศรีรัตน์. Sensor and Transducers in Manufacturing. คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

อภิชาต ภู่พลับ. เริ่มต้นเขียนโปรแกรมติดต่อและควบคุมฮาร์ดแวร์ด้วย Visual Basic. นนทบุรี : อินโฟเพรส. 2546.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE MCS-51 (1)

```
#include<reg51.h>
sbit oc4=P2^7;
sbit oc3=P2^6;
sbit oc2=P2^5;
sbit oc1=P2^4;
sbit TEACH=P2^0;
sbit SAVE=P2^1;
sbit AUTO=P2^2;
sbit HOME=P2^3;
sbit t1=P3^0;
sbit t2=P3^1;
sbit t3=P3^2;
sbit t4=P3^3;
sbit t5=P3^4;
sbit t6=P3^5;
sbit limm1=P0^0;
sbit limm2=P0^1;
sbit limm3=P0^2;
sbit limm4=P0^3;
sbit limm5=P0^4;
sbit limm7=P0^5;
sbit limout=P0^6;
sbit cnc=P1^1;
sbit cncout=P1^2;
sbit v=P1^0;
void delay(int time)
{
  int i,j;
  for(i=0;i<time;i++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  for(j=0;j<250;j++);
}
main()
{
  P0=0xff;
  P1=0xff;
  P2=0xff;
  P3=0xff;
  limout=1;
  while(1)
  {  P3=0xff;
    if(t1==0&&t2==0&&t3==0&&t4==0&&t5==0&&t6==0)
     {TEACH=0;SAVE=1;AUTO=1;HOME=1;
       while(TEACH==0)
       {
       if(t1==0&&t2==0&&t3==1&&t4==0&&t5==0&&t6==0){oc1=0;oc2=0;oc3=0;oc4=0;}
       //m1     4
       if(t1==1&&t2==0&&t3==1&&t4==0&&t5==0&&t6==0){oc1=0;oc2=0;oc3=0;oc4=1;}
       //m1     5
       if(t1==0&&t2==1&&t3==1&&t4==0&&t5==0&&t6==0){oc1=0;oc2=0;oc3=1;oc4=0;}
       //m2     6
       if(t1==1&&t2==1&&t3==1&&t4==0&&t5==0&&t6==0){oc1=0;oc2=0;oc3=1;oc4=1;}
       //m2     7
       if(t1==0&&t2==0&&t3==0&&t4==1&&t5==0&&t6==0){oc1=0;oc2=1;oc3=0;oc4=0;}
       //m3     8
       if(t1==1&&t2==0&&t3==0&&t4==1&&t5==0&&t6==0){oc1=0;oc2=1;oc3=0;oc4=1;}
       //m3     9
       if(t1==0&&t2==1&&t3==0&&t4==1&&t5==0&&t6==0){oc1=0;oc2=1;oc3=1;oc4=0;}
       //m4     10
       if(t1==1&&t2==1&&t3==0&&t4==1&&t5==0&&t6==0){oc1=0;oc2=1;oc3=1;oc4=1;}
       //m4     11
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(t1==0&&t2==0&&t3==1&&t4==1&&t5==0&&t6==0){oc1=1;oc2=0;oc3=0;oc4=0;}
//m5     12
if(t1==1&&t2==0&&t3==1&&t4==1&&t5==0&&t6==0){oc1=1;oc2=0;oc3=0;oc4=1;}
//m5     13
if(t1==0&&t2==1&&t3==1&&t4==1&&t5==0&&t6==0){oc1=1;oc2=0;oc3=1;oc4=0;}
//m6     14
if(t1==1&&t2==1&&t3==1&&t4==1&&t5==0&&t6==0){oc1=1;oc2=0;oc3=1;oc4=1;}
//m6     15
if(t1==0&&t2==0&&t3==0&&t4==0&&t5==1&&t6==0){oc1=1;oc2=1;oc3=0;oc4=0;}
//m7     16
if(t1==1&&t2==0&&t3==0&&t4==0&&t5==1&&t6==0){oc1=1;oc2=1;oc3=0;oc4=1;}
//m7     17
if(t1==0&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0){oc1=1;oc2=1;oc3=1;oc4=0;}
//stop   18
if(t1==1&&t2==0&&t3==0&&t4==0&&t5==0&&t6==0) {TEACH=1;}    //1
if(t1==0&&t2==1&&t3==0&&t4==0&&t5==0&&t6==0) {TEACH=1;}    //2
if(t1==1&&t2==1&&t3==0&&t4==0&&t5==0&&t6==0) {TEACH=1;}    //3
if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}   //19
if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}   //20
 }
}
else  {oc1=1;oc2=1;oc3=1;oc4=1;TEACH=1;SAVE=1;AUTO=1;HOME=1;}
if(t1==1&&t2==0&&t3==0&&t4==0&&t5==0&&t6==0)
{
 TEACH=1;SAVE=0;AUTO=1;HOME=1;
 while(t1==1&&t2==0&&t3==0&&t4==0&&t5==0&&t6==0);
 }//save
 if(t1==0&&t2==1&&t3==0&&t4==0&&t5==0&&cnc==0&&t6==0)
 {
    TEACH=1;SAVE=1;AUTO=0;HOME=1;oc1=1;oc2=1;oc3=1;oc4=1;
     limout=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   while(limout==1)
    { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
      if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
     }delay(2020);
if(limm2==1)
{
 while(limm2==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
   }
   limout=0;delay(20);limout=1;
}
else {limout=0;delay(20);limout=1;}
delay(3000);
if(limm3==1)
{
 while(limm3==1)
 { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
   if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
  }
  limout=0;delay(20);limout=1;
}
 else {limout=0;delay(20);limout=1;}
 if(limm7==1)
 {
   while(limm7==1)
   { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
     if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
   }
   limout=0;delay(20);limout=1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 else {limout=0;delay(20);limout=1;}
 if(limm5==1)
 {
  while(limm5==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
  }
  limout=0;delay(20);limout=1;
}
else {limout=0;delay(20);limout=1;}
delay(1000);
if(limm4==1)
{
  while(limm4==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
  }
  limout=0;delay(20);limout=1;
}
else {limout=0;delay(20);limout=1;}
if(limm1==1)
{
  while(limm1==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0) {v=1;}
  }
  limout=0;delay(20);limout=1;
}
else {limout=0;delay(20);limout=1;}
cncout=0;delay(1000);cncout=1;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else {oc1=1;oc2=1;oc3=1;oc4=1;TEACH=1;SAVE=1;AUTO=1;HOME=1;}
if(t1==1&&t2==1&&t3==0&&t4==0&&t5==0&&t6==0)
{
 TEACH=1;SAVE=1;AUTO=1;HOME=0;delay(2000);
 if(limm2==1)
 {
  while(limm2==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}  //19
   if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0){v=1;}
  }
  limout=0;delay(20);limout=1;
 }
 else {limout=0;delay(20);limout=1;}
 if(limm3==1)
 {
  while(limm3==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}  //19
   if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0){v=1;}
  }
  limout=0;delay(20);limout=1;
 }
 else {limout=0;delay(20);limout=1;}
 if(limm7==1)
 {
  while(limm7==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}  //19
   if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0){v=1;}
  }
  limout=0;delay(20);limout=1;
}
else {limout=0;delay(20);limout=1;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 if(limm5==1)
 {
  while(limm5==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}   //19
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0){v=1;}
  }
  limout=0;delay(20);limout=1;
 }
 else {limout=0;delay(20);limout=1;}
 if(limm4==1)
 {
  while(limm4==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}   //19
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0){v=1;}
  }
  limout=0;delay(20);limout=1;
 }
 else {limout=0;delay(20);limout=1;}
 if(limm1==1)
 {
  while(limm1==1)
  { if(t1==1&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0) {v=0;}   //19
    if(t1==0&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0){v=1;}
  }
  limout=0;delay(20);limout=1;
 }
 else {limout=0;delay(20);limout=1;}
 cncout=0;delay(1000);cncout=1;
 while(t1==1&&t2==1&&t3==0&&t4==0&&t5==0&&t6==0);
}//home
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if(t1==0&&t2==1&&t3==0&&t4==0&&t5==1&&t6==0)
    {
     oc1=1;oc2=1;oc3=1;oc4=0;TEACH=1;SAVE=1;AUTO=1;HOME=1;
    } //stop
    if(t1==1&&t2==0&&t3==1&&t4==0&&t5==1&&t6==0)
    {
      TEACH=1;SAVE=1;AUTO=0;HOME=0;
    } //21
  }
}
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE MCS-51 (2)

```
#include<reg51.h>
sbit oc1=P2^4;
sbit oc2=P2^5;
sbit oc3=P2^6;
sbit oc4=P2^7;
sbit TEACH=P2^0;
sbit SAVE=P2^1;
sbit AUTO=P2^2;
sbit HOME=P2^3;
sbit enc1=P3^0;
sbit enc2=P3^1;
sbit enc3=P3^2;
sbit enc4=P3^3;
sbit enc5=P3^4;
sbit enc6=P3^5;
sbit enc7=P3^6;
sbit m11=P0^0;
sbit m12=P0^1;
sbit m21=P0^2;
sbit m22=P0^3;
sbit m31=P0^4;
sbit m32=P0^5;
sbit m41=P0^6;
sbit m42=P0^7;
sbit m51=P1^0;
sbit m52=P1^1;
sbit m61=P1^2;
sbit m62=P1^3;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
sbit m71=P1^4;
sbit m72=P1^5;
sbit lim=P3^7;
int num[30];
char side[30];
char motor,true,mot;
int sum,ss;
int c,i,b=0;
void delay(int time)
{
 int i,j;
 for(i=0;i<time;i++)
 for(j=0;j<250;j++);
}
void   encoder(unsigned char y)
{
 switch(y)
 {
  case 1:
    {
     motor=1;
     if(b!=enc1) {sum++;b=enc1;}
    } break;
  case 2:
    {
     motor=1;
     if(b!=enc1) {sum--;b=enc1;}
    }   break;
  case 3:
    {
     motor=2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   if(b!=enc2) {sum++;b=enc2;}
  } break;
case 4:
  {
   motor=2;
   if(b!=enc2) {sum--;b=enc2;}
  }   break;
case 5:
  {
   motor=3;
   if(b!=enc3) {sum++;b=enc3;}
  } break;
case 6:
  {
   motor=3;
   if(b!=enc3) {sum--;b=enc3;}
  }   break;
case 7:
  {
   motor=4;
   if(b!=enc4) {sum++;b=enc4;}
  } break;
case 8:
  {
   motor=4;
   if(b!=enc4) {sum--;b=enc4;}
  }   break;
case 9:
  {
   motor=5;
   if(b!=enc5) {sum++;b=enc5;}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    } break;
  case 10:
    {
     motor=5;
     if(b!=enc5) {sum--;b=enc5;}
    }   break;
  case 11:
    {
     motor=6;
     if(b!=enc6) {sum++;b=enc6;}
    } break;
  case 12:
    {
     motor=6;
     if(b!=enc6) {sum--;b=enc6;}
    }   break;
  case 13:
    {
     motor=7;
     if(b!=enc7) {sum++;b=enc7;}
    } break;
  case 14:
    {
     motor=7;
     if(b!=enc7) {sum--;b=enc7;}
    }   break;
  }
}
void receive()
{
  if(oc1==0&&oc2==0&&oc3==0&&oc4==0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ m11=1;m12=0;m21=1;m22=1;m41=1;m42=1;m51=1;m52=1;m61=1;m62=1;m71=1;
 m72=1;m31=1;m32=1;encoder(1);
}//m1
if(oc1==0&&oc2==0&&oc3==0&&oc4==1)
{ m11=0;m12=1;m21=1;m22=1;m41=1;m42=1;m51=1;m52=1;m61=1;m62=1;m71=1;
 m72=1;m31=1;m32=1;encoder(2);}//m1
if(oc1==0&&oc2==0&&oc3==1&&oc4==0)
{ m21=1;m22=0;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m61=1;m62=1;m71=1;
 m72=1;m31=1;m32=1;encoder(3);}//m2
if(oc1==0&&oc2==0&&oc3==1&&oc4==1)
{ m21=0;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m61=1;m62=1;m71=1;
 m72=1;m31=1;m32=1;encoder(4);
}//m2
if(oc1==0&&oc2==1&&oc3==0&&oc4==0)
{ m31=1;m32=0;m21=1;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m61=1;
 m62=1;m71=1;m72=1;encoder(5);
}//m3
if(oc1==0&&oc2==1&&oc3==0&&oc4==1)
{ m31=0;m32=1;m21=1;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m61=1;
 m62=1;m71=1;m72=1;encoder(6);
}//m3
if(oc1==0&&oc2==1&&oc3==1&&oc4==0)
{ m41=0;m42=1;m51=1;m52=0;m21=1;m22=1;m11=1;m12=1;m61=1;m62=1;m71=1;
 m72=1;m31=1;m32=1;encoder(7);
}//m4
if(oc1==0&&oc2==1&&oc3==1&&oc4==1)
{ m41=1;m42=0;m51=0;m52=1;m21=1;m22=1;m11=1;m12=1;m61=1;m62=1;m71=1;
 m72=1;m31=1;m32=1;encoder(8);
}//m4
if(oc1==1&&oc2==0&&oc3==0&&oc4==0)
{ m41=1;m42=0;m51=1;m52=0;m21=1;m22=1;m11=1;m12=1;m61=1;m62=1;m71=1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  m72=1;m31=1;m32=1;encoder(9);
 }//m5
 if(oc1==1&&oc2==0&&oc3==0&&oc4==1)
 { m41=0;m42=1;m51=0;m52=1;m21=1;m22=1;m11=1;m12=1;m61=1;m62=1;m71=1;
  m72=1;m31=1;m32=1;encoder(10);
 }//m5
 if(oc1==1&&oc2==0&&oc3==1&&oc4==0)
 { m61=1;m62=0;m21=1;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m71=1;
  m72=1;m31=1;m32=1;encoder(11);
 }//m6
 if(oc1==1&&oc2==0&&oc3==1&&oc4==1)
 { m61=0;m62=1;m21=1;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m71=1;
  m72=1;m31=1;m32=1;encoder(12);
 }//m6
 if(oc1==1&&oc2==1&&oc3==0&&oc4==0)
 { m71=1;m72=0;m21=1;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m61=1;
  m62=1;m31=1;m32=1;encoder(13);
 }//m7
 if(oc1==1&&oc2==1&&oc3==0&&oc4==1)
 { m71=0;m72=1;m21=1;m22=1;m11=1;m12=1;m41=1;m42=1;m51=1;m52=1;m61=1;
  m62=1;m31=1;m32=1;encoder(14);
 }//m7
 if(oc1==1&&oc2==1&&oc3==1&&oc4==0) {P1=0xff;P0=0xff;}
}
void homework()
{  m61=0;m62=1;delay(2000);P1=0xff;P0=0xff;
  m21=1;m22=0;if(lim==1) {while(lim==1);}P1=0xff;P0=0xff;delay(20);
  m31=1;m32=0;delay(3000);P1=0xff;P0=0xff;
  m31=0;m32=1;if(lim==1) {while(lim==1);}P1=0xff;P0=0xff;delay(20);
  m71=0;m72=1; if(lim==1) {while(lim==1);}P1=0xff;P0=0xff;delay(20);
  m41=0;m42=1;m51=0;m52=1;if(lim==1){while(lim==1);}P1=0xff;P0=0xff;delay(20);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    //clrm5
    m41=0;m42=1;m51=1;m52=0;delay(1000);P1=0xff;P0=0xff;
   m41=1;m42=0;m51=0;m52=1;if(lim==1){while(lim==1);}P1=0xff;P0=0xff;delay(20);
   //downm4
   m11=1;m12=0;if(lim==1) {while(lim==1);}P1=0xff;P0=0xff;delay(20);
}
void encoderd(unsigned char g)
{
 switch(g)
  {
   case 1: if(b!=enc1) {ss--;b=enc1;}break;
   case 2: if(b!=enc1) {ss++;b=enc1;}break;
   case 3: if(b!=enc2) {ss--;b=enc2;}break;
   case 4: if(b!=enc2) {ss++;b=enc2;}break;
   case 5: if(b!=enc3) {ss--;b=enc3;}break;
   case 6: if(b!=enc3) {ss++;b=enc3;}break;
   case 7: if(b!=enc4) {ss--;b=enc4;}break;
   case 8: if(b!=enc4) {ss++;b=enc4;}break;
   case 9: if(b!=enc5) {ss--;b=enc5;}break;
   case 10: if(b!=enc5) {ss++;b=enc5;}break;
   case 11: if(b!=enc6) {ss--;b=enc6;}break;
   case 12: if(b!=enc6) {ss++;b=enc6;}break;
   case 13: if(b!=enc7) {ss--;b=enc7;}break;
   case 14: if(b!=enc7) {ss++;b=enc7;}break;
  }
}
void driver()
{
   if(mot==1)
     { if(ss>0)
              { m11=1;m12=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                while(true==1)
                 {
                  encoderd(1);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)  {ss=0;}  if(ss==0)
true=0;
                 }
                }
       if(ss<0) {m11=0;m12=1;
                while(true==1)
                {
                 encoderd(2);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
                }
               }
      }
    if(mot==2)
      {
       if(ss>0)
          { m21=1;m22=0;
           while(true==1)
           {
            encoderd(3);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
           }
          }
        if(ss<0)
          {
           m21=0;m22=1;
           while(true==1)
           {
            encoderd(4);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            }
           }
        }
      if(mot==3)
      { if(ss>0)
            { m31=1;m32=0;
             while(true==1)
              {
                encoderd(5);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
              }
            }
       if(ss<0) {m31=0;m32=1;
            while(true==1)
             {
              encoderd(6);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
              }
            }
      }
     if(mot==4)
       {
        if(ss>0)
            { m41=0;m42=1;m51=1;m52=0;
             while(true==1)
             {
              encoderd(7);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
             }
            }
         if(ss<0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            {
             m41=1;m42=0;m51=0;m52=1;
              while(true==1)
               {
                 encoderd(8);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)        {ss=0;}if(ss==0)
true=0;
               }
             }
        }
    if(mot==5)
     {  if(ss>0)
             {m41=1;m42=0;m51=1;m52=0;
              while(true==1)
               {
                 encoderd(9);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)       {ss=0;}if(ss==0)
true=0;
               }
              }
      if(ss<0) {m41=0;m42=1;m51=0;m52=1;
             while(true==1)
              {
                encoderd(10);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)       {ss=0;}if(ss==0)
true=0;
               }
             }
      }
    if(mot==6)
      {
       if(ss>0)
           { m61=1;m62=0;
            while(true==1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                {
                 encoderd(11);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
                }
               }
          if(ss<0)
               {
                m61=0;m62=1;
                while(true==1)
                 {
                  encoderd(12);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
                 }
                }
         }
    if(mot==7)
     { if(ss>0)
               { m71=1;m72=0;
                while(true==1)
                 {
                  encoderd(13);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)    {ss=0;}if(ss==0)
true=0;
                 }
                }
       if(ss<0) {m71=0;m72=1;
               while(true==1)
                {
                 encoderd(14);if(oc1==1&&oc2==1&&oc3==1&&oc4==0)      {ss=0;}if(ss==0)
true=0;
                }
               }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
}
main()
{  delay(1000);
 P0=0xff;
 P1=0xff;
 P2=0xff;
 P3=0xff;
 b=0;true=1;c=0;motor=0;
  sum=0;ss=0;
  for(i=0;i<30;i++)
  {
   num[i]=0;side[i]=0;
  }
  while(1)
 {
   TEACH=1;
    if(TEACH==0)
      {
          receive();
      }
   else
       {P0=0xff;P1=0xff;}
   AUTO=1;
    if(AUTO==0&&HOME==1)
      { for(i=0;i<30;i++)
       { ss=num[i];
         mot=side[i];
         true=1;
         driver();
         P0=0xff;P1=0xff;delay(100);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        }
        lim=0;delay(20);lim=1;
        homework();
        delay(2000);
       }
     SAVE=1;
     if(SAVE==0)
       {num[c]=sum;
       side[c]=motor;
       c++;
        sum=0;motor=0;P0=0xff;P1=0xff;
         while(SAVE==0);
       }
     HOME=1;
     if(HOME==0&&AUTO==1)
      {
       homework();
       while(HOME==0&&AUTO==1);
      }
      if(AUTO==0&&HOME==0)
      {
      for(i=0;i<30;i++)
      {
        num[i]=0;side[i]=0;
      }
      sum=0;motor=0;P0=0xff;P1=0xff;while(HOME==0&&AUTO==0);
      }
  }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CODE VISUAL BASIC

```
Public pwrite As Integer
Private Sub Command1_Click(Index As Integer)
   Out pwrite, &H10
End Sub
Private Sub Command10_Click()
   Out pwrite, &HB
End Sub
Private Sub Command11_Click(Index As Integer)
   Out pwrite, &HC
End Sub
Private Sub Command12_Click(Index As Integer)
   Out pwrite, &HD
End Sub
Private Sub Command13_Click(Index As Integer)
      Out pwrite, &HE
End Sub
Private Sub Command14_Click(Index As Integer)
   Out pwrite, &HF
End Sub
Private Sub Command15_Click(Index As Integer)
   Out pwrite, &H0
End Sub
Private Sub Command16_Click(Index As Integer)
   Out pwrite, &H1
End Sub
Private Sub Command17_Click(Index As Integer)
   Out pwrite, &H2
End Sub
Private Sub Command18_Click(Index As Integer)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Out pwrite, &H3
End Sub
Private Sub Command19_Click()
    Out pwrite, &H12
End Sub
Private Sub Command2_Click()
    Out pwrite, &H11
End Sub
Private Sub Command20_Click()
    Out pwrite, &H20
End Sub
Private Sub Command3_Click()
    Out pwrite, &H4
End Sub
Private Sub Command4_Click()
    Out pwrite, &H5
End Sub
Private Sub Command5_Click()
    Out pwrite, &H6
End Sub
Private Sub Command6_Click(Index As Integer)
    Out pwrite, &H7
End Sub
Private Sub Command7_Click(Index As Integer)
    Out pwrite, &H8
End Sub
Private Sub Command8_Click(Index As Integer)
    Out pwrite, &H9
End Sub
Private Sub Command9_Click()
    Out pwrite, &HA
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form_Load()
   pwrite = &H378
End Sub
Private Sub Option1_Click()
   Out pwrite, &H13
End Sub
Private Sub Option2_Click()
   Out pwrite, &H14
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 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
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

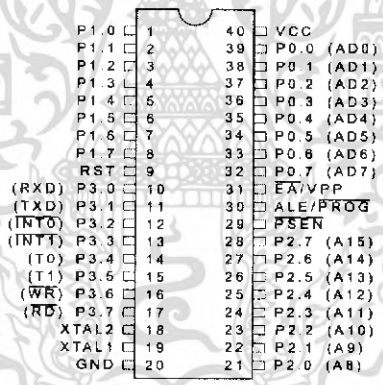
Description

The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes 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.

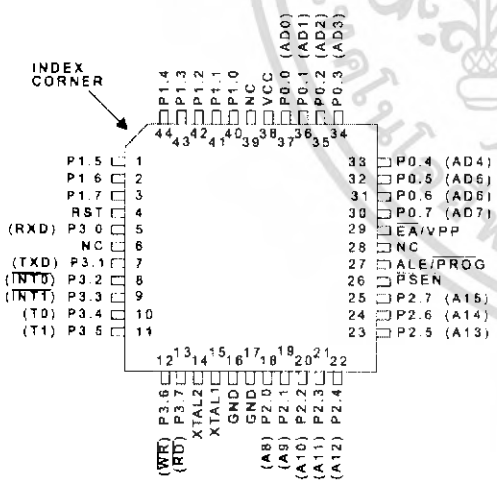
(continued)

Pin Configurations

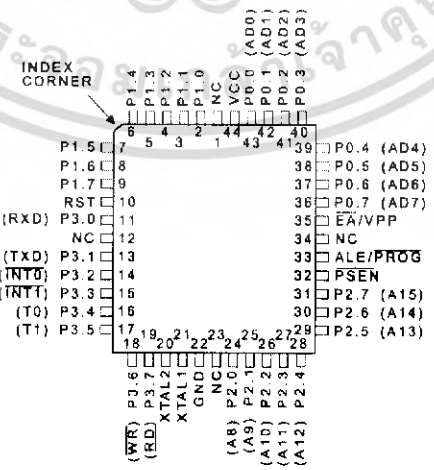
PDIP



PQFP/TQFP



PLCC



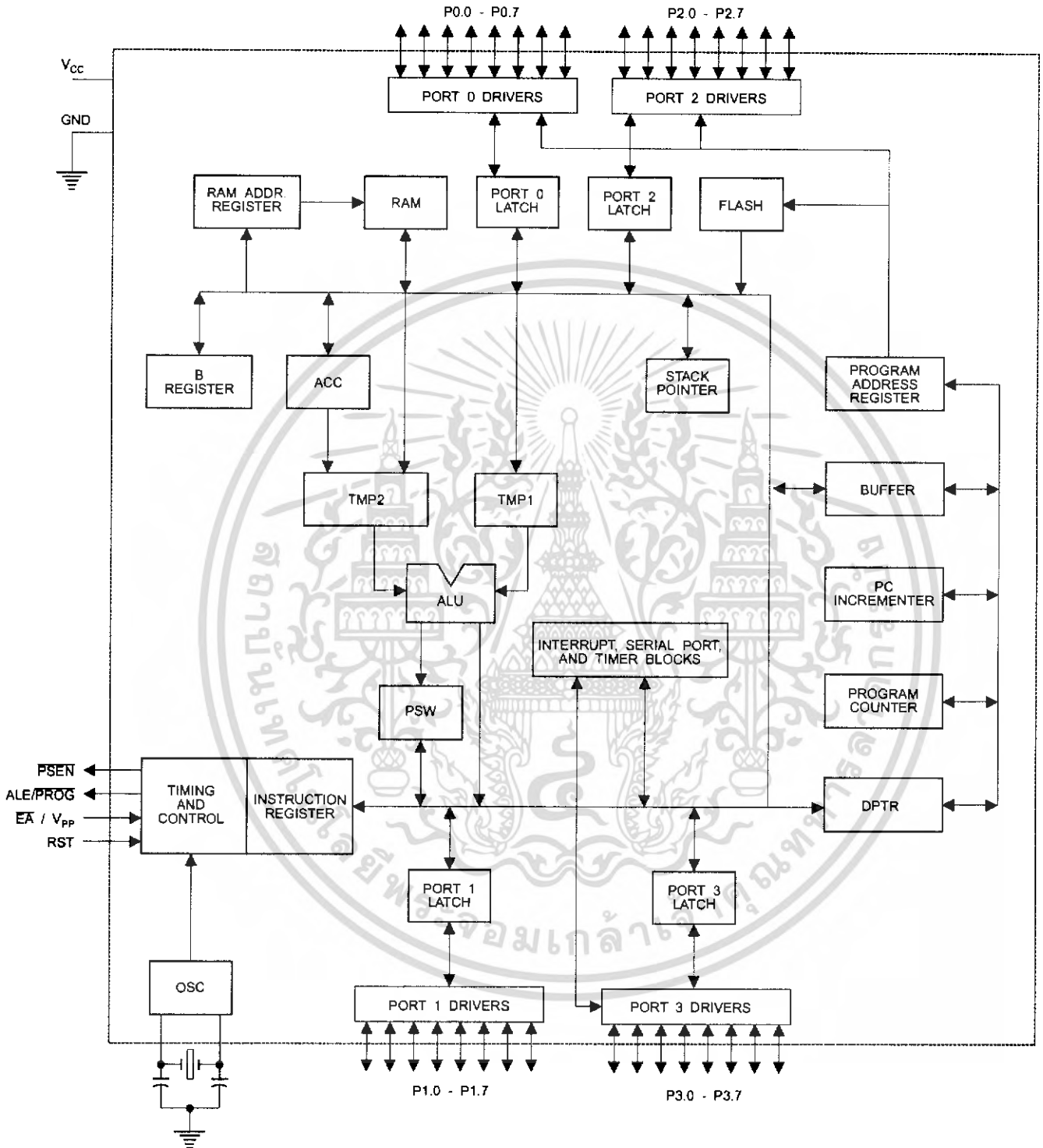
8-Bit Microcontroller with 4K Bytes Flash

AT89C51

0265F-A–12/97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

lock Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt 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 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	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

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

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

ALE/$\overline{PROG}$
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 ($\overline{PROG}$) 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.

$\overline{PSEN}$
Program Store Enable is the read strobe to external program memory.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

When the AT89C51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/V_{PP}
External Access Enable. $\overline{EA}$ 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, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ 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}.

XTAL1
Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2
Output from the inverting oscillator amplifier.

Oscillator Characteristics

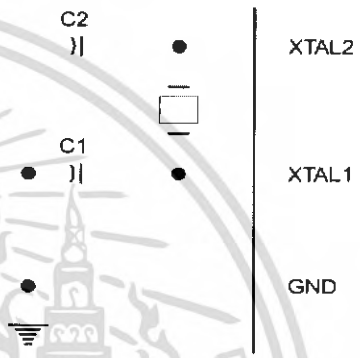
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected 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.

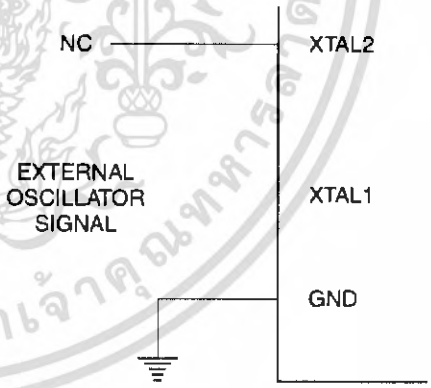
It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port 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.

Figure 1. Oscillator Connections



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

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down Modes

Mode	Program Memory	ALE	$\overline{PSEN}$	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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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.

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	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash 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} = 12V	V_{PP} = 5V
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 12V for the high-voltage programming mode.
5. Pulse 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.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ 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. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 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 12V programming
(032H) = 05H indicates 5V 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	$\overline{PSEN}$	ALE/$\overline{PROG}$	$\overline{EA}$/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	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	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10-ms PROG pulse.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3. Programming the Flash

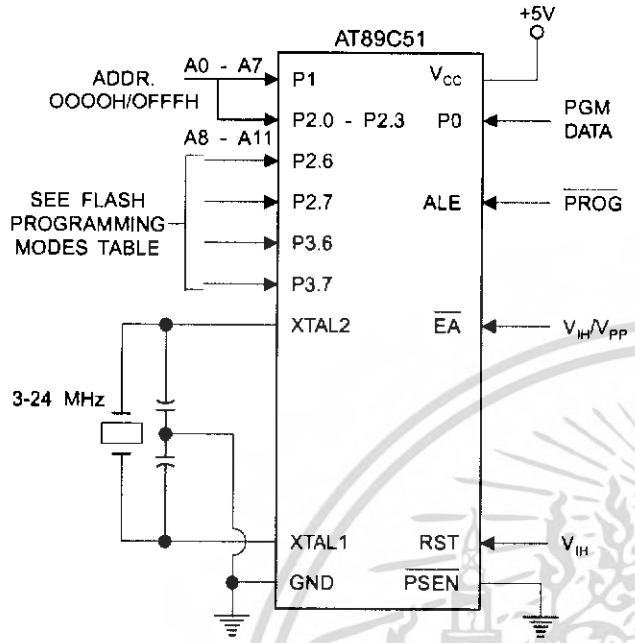
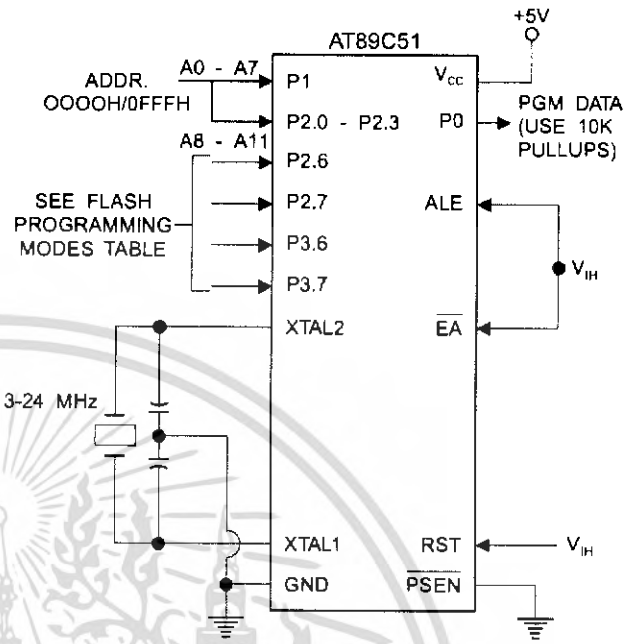


Figure 4. Verifying the Flash



Flash Programming and Verification Characteristics

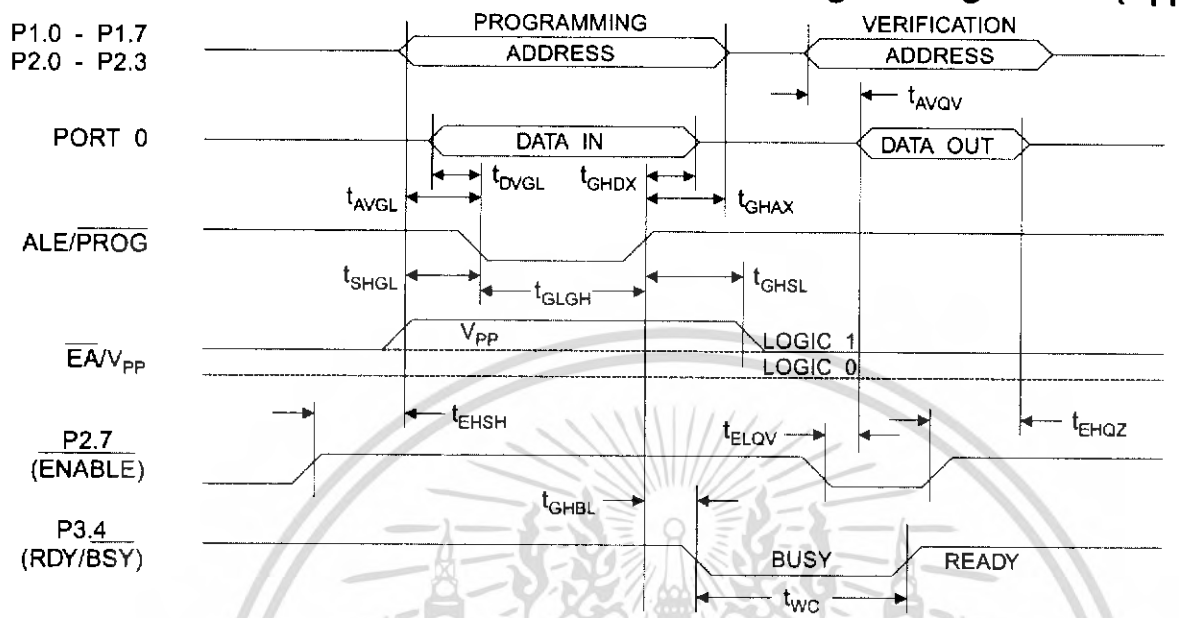
T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

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
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{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_{EHQZ}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

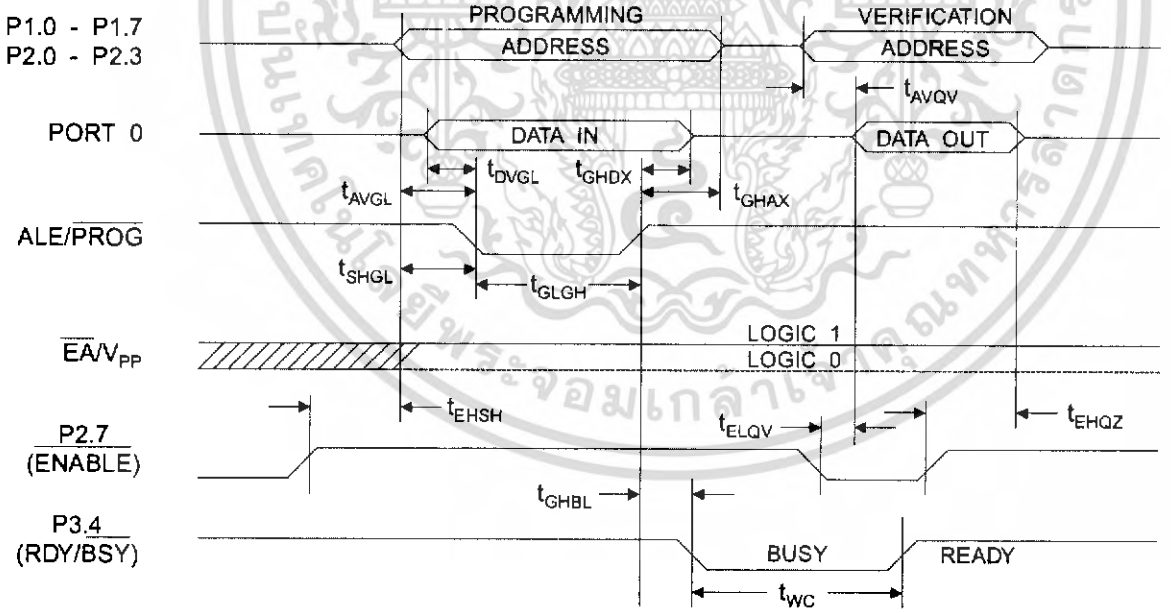
Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High Voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low Voltage Mode (V_{PP} = 5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*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.

DC Characteristics

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

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ± 10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode(2)	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

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

2. Minimum V_{CC} for Power Down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

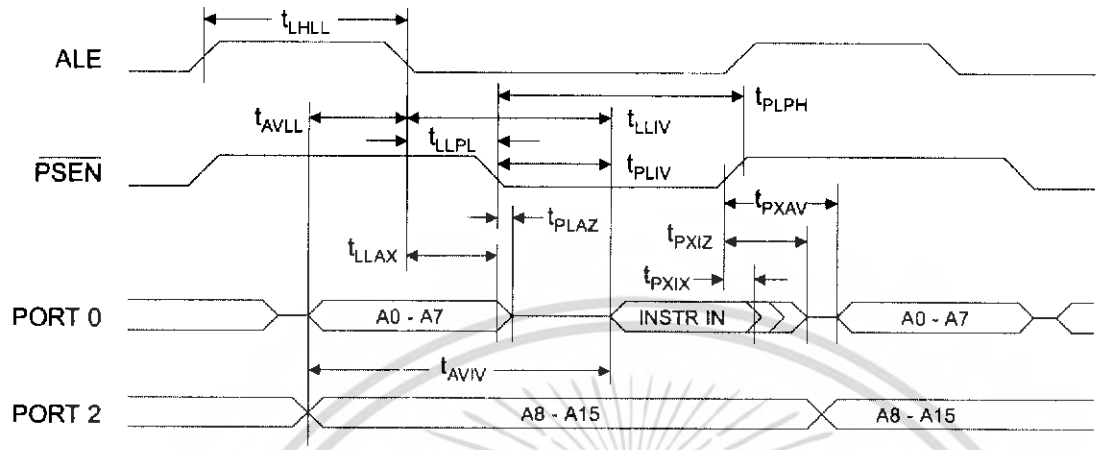
(Under Operating Conditions; Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

External Program and Data Memory Characteristics

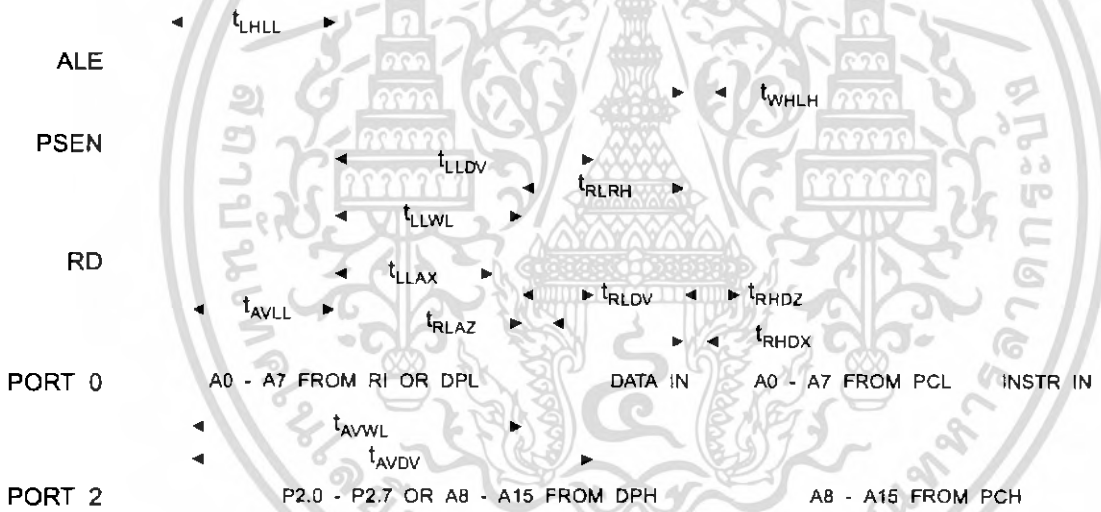
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

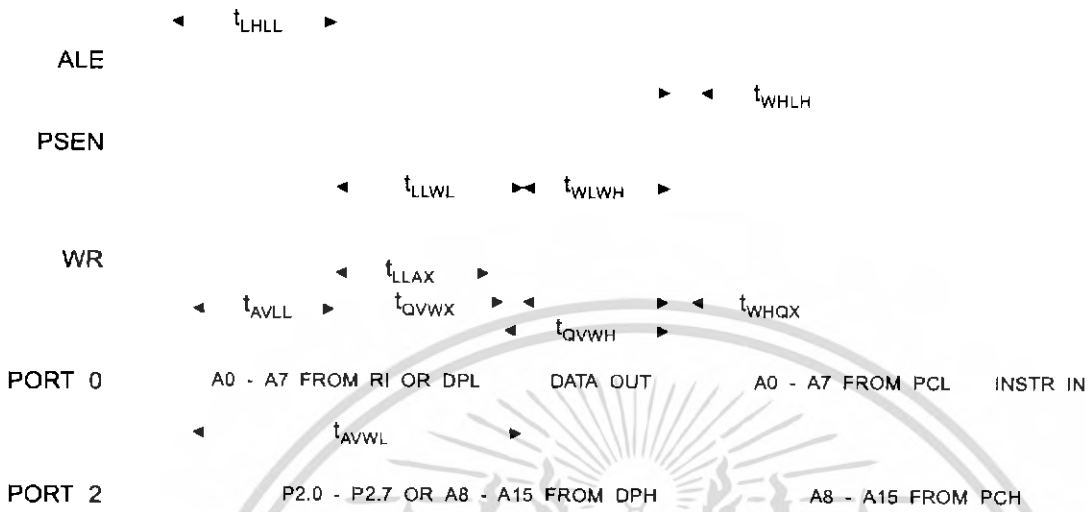


External Data Memory Read Cycle

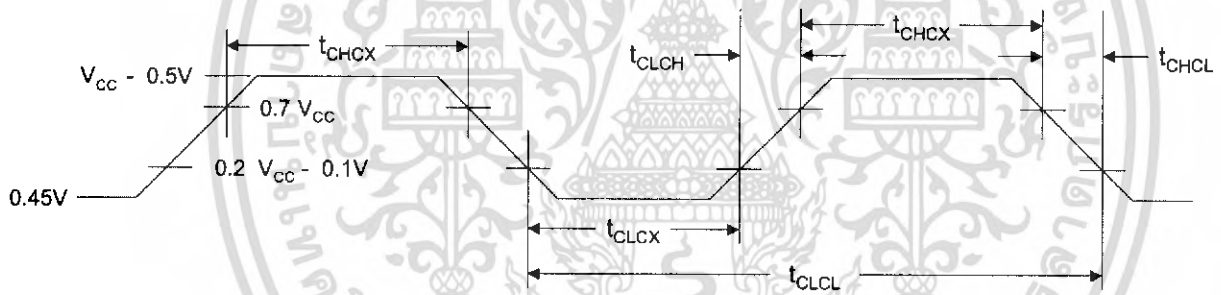


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

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

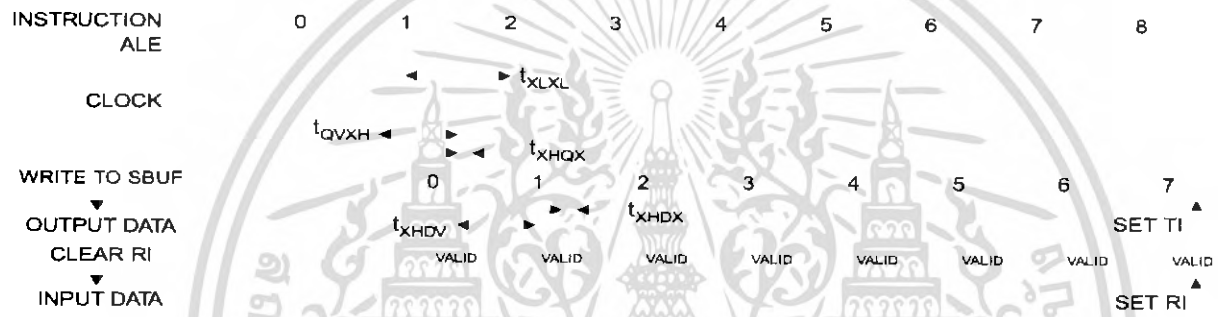
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

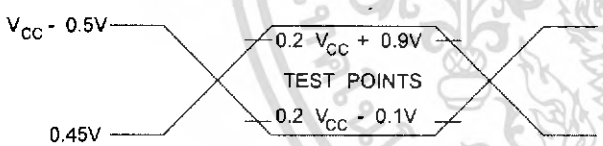
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		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

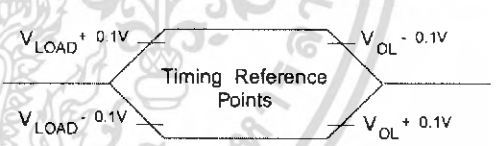


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V 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 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ± 20%	AT89C51-12AC	44A	Commercial
		AT89C51-12JC	44J	(0°C to 70°C)
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial
		AT89C51-12JI	44J	(-40°C to 85°C)
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
		AT89C51-12AA	44A	Automotive
		AT89C51-12JA	44J	(-40°C to 105°C)
		AT89C51-12PA	40P6	
		AT89C51-12QA	44Q	
16	5V ± 20%	AT89C51-16AC	44A	Commercial
		AT89C51-16JC	44J	(0°C to 70°C)
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial
		AT89C51-16JI	44J	(-40°C to 85°C)
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
		AT89C51-16AA	44A	Automotive
		AT89C51-16JA	44J	(-40°C to 105°C)
		AT89C51-16PA	40P6	
		AT89C51-16QA	44Q	
20	5V ± 20%	AT89C51-20AC	44A	Commercial
		AT89C51-20JC	44J	(0°C to 70°C)
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial
		AT89C51-20JI	44J	(-40°C to 85°C)
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
24	5V ± 20%	AT89C51-24AC	44A	Commercial
		AT89C51-24JC	44J	(0°C to 70°C)
		AT89C51-24PC	44P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial
		AT89C51-24JI	44J	(-40°C to 85°C)
		AT89C51-24PI	44P6	
		AT89C51-24QI	44Q	

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

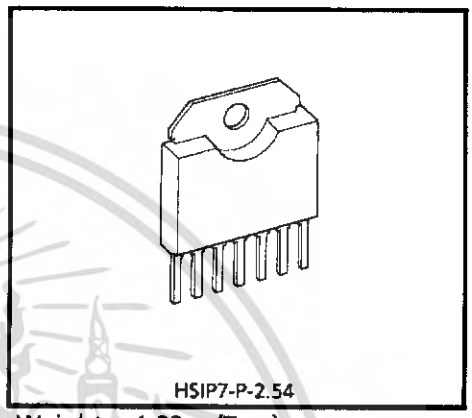
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TOSHIBA BIPOLAR LINEAR INTEGRATED CIRCUIT SILICON MONOLITHIC

TA7257P

BRIDGE DRIVER

The TA7257P is a Full Bridge Driver for blashed DC Motor Rotation control.
Forward Rotation, Reverse Rotation, Stop and Braking operations are available.
It's designed for Loading and Reel Motor driver for VTR and Tape Deck, and any other consumer and industrial applications.
TA7257P have Operation Supply Voltage terminal and Motor Driving Supply Voltage terminal independently therefore Servo control operation is applicable.



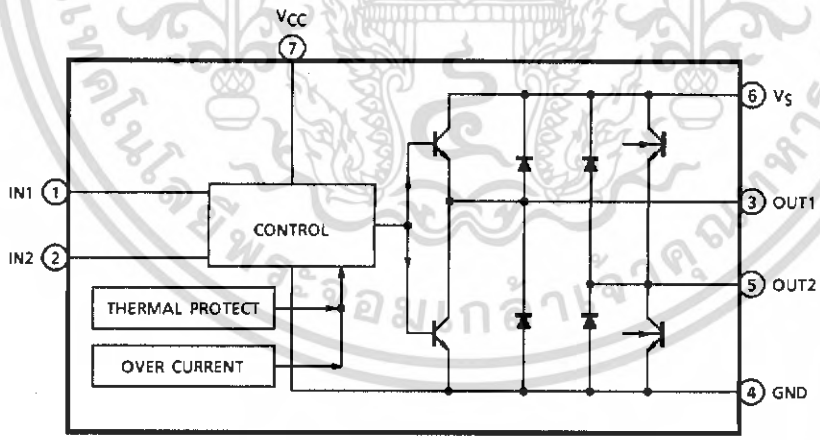
HSIP7-P-2.54

Weight : 1.88 g (Typ.)

FEATURES

- Output Current Up to 1.5 A (AVE.), and 4.5 A (PEAK).
- 4 Function Modes (CW, CCW, STOP and Brake) are Controlled by 2 Logic Signals Fed Into 2 Input Terminals.
- Build in Over Current Protector and Thermal Shut Down Circuit.
- Operating Voltage Range : $V_{CC\,(opr.)}$ = 6~18 V, $V_{S\,(opr.)}$ = 0~18 V

BLOCK DIAGRAM



980910EBA1

- TOSHIBA is continually working to improve the quality and the reliability of its products. Nevertheless, semiconductor devices in general can malfunction or fail due to their inherent electrical sensitivity and vulnerability to physical stress. It is the responsibility of the buyer, when utilizing TOSHIBA products, to observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- The products described in this document are subject to the foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTION

PIN No.	SYMBOL	FUNCTIONAL DESCRIPTION
1	IN1	Input terminal
2	IN2	Input terminal
3	OUT1	Output terminal
4	GND	GND terminal
5	OUT2	Output terminal
6	V_S	Supply voltage terminal for Motor drive
7	V_{CC}	Supply voltage terminal for Logic

FUNCTION

IN1	IN2	OUT1	OUT2	MODE
1	1	L	L	Brake
0	1	L	H	CW / CCW
1	0	H	L	CCW / CW
0	0			Stop

MAXIMUM RATINGS (Ta = 25°C)

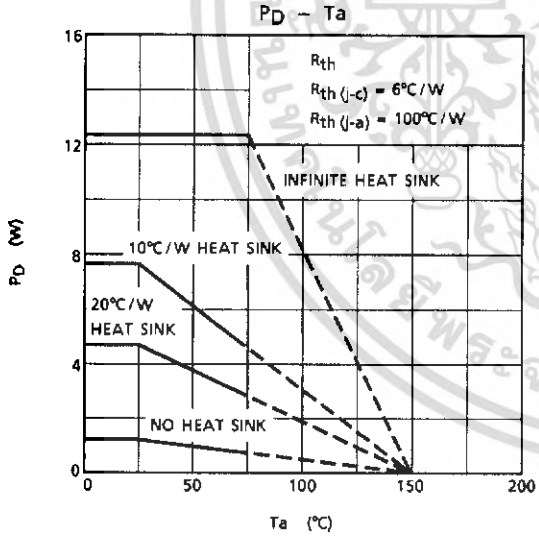
CHARACTERISTIC		SYMBOL	RATING	UNIT
Peak Supply Voltage	Peak	V_{CC} (MAX.)	25	V
	Operate	V_{CC} (opr.)	18	
Output Current	PEAK	I_O (PEAK)	4.5	A
	AVE.	I_O (AVE.)	1.5	
Power Dissipation		P_D	12.5 (Note)	W
Operating Temperature		T_{opr}	−30~75	°C
Storage Temperature		T_{stg}	−55~150	°C

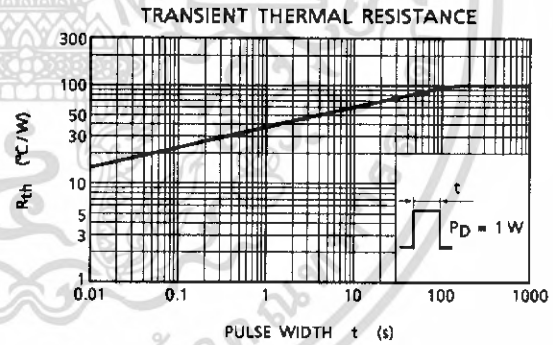
(Note) Tc = 75°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (Ta = 25°C)

CHARACTERISTIC		SYMBOL	TEST CIR-CUIT	TEST CONDITION	MIN.	TYP.	MAX.	UNIT
Supply Current		I_{CC1}	—	V_{CC} = 18 V Output OFF stop mode	—	6.5	13	mA
		I_{CC2}		V_{CC} = 18 V Output OFF CW / CCW mode	—	10	20	
Saturation Voltage	Upper	$V_{S1}U$	—	V_{CC} = 18 V, I_O = 0.1 A	—	0.7	1.0	V
	Lower	$V_{S1}L$			—	0.6	0.9	
	Upper	$V_{S2}U$		V_{CC} = 18 V, I_O = 1.1 A	—	1.0	1.4	
	Lower	$V_{S2}L$			—	0.9	1.3	
Output Transistor Leakage Current	Upper	I_L U	—	V_S = 18 V	—	—	100	μA
	Lower	I_L L			—	—	100	
Input Voltage 1, 2		V_{IN} (H)	—	T_j = 25°C, pin ① and pin ②	3.0	—	—	V
		V_{IN} (L)			—	—	0.8	
Diode Forward Voltage		V_F U	—	I_F = 1.0 A	—	2.0	—	V
		V_F L			—	1.25	—	
Limiting Current		I_{SC}	—	—	—	3.5	—	A
Input Current		I_{IN}	—	—	—	1	10	μA





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION NOTE

(1) Input circuit

Input circuit is shown in Fig.1. It's a "Low active" type voltage comparator that's one input connect to Input terminal (pin ①, or ②) and the other to built-in temperature compensated voltage reference (V_{TH} = 1.4 V Typ.)
If a voltage above $V_{IN\,(H)}$ fed into the Input Terminal that means "Logic 1" and less than $V_{IN\,(L)}$ or connect to GND means "Logic 0".

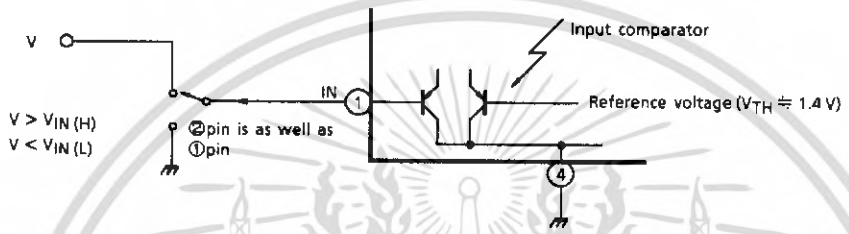


Fig.1

(2) Basic application circuit

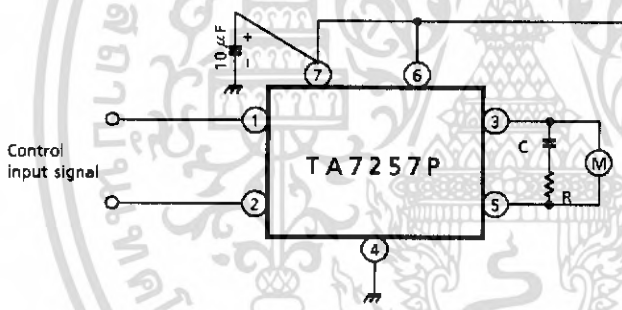


Fig.2

(Note 1) Fig.2 shows the basic application circuit.
Optimum values of the C, R depend on the inherent constant of a motor and parasitic C, R values around the circuit.
Normally, recommended to use 0.1 μF and 33 Ω.

(Note 2) Utmost care is necessary in the design of the output line, V_S, V_{CC} and GND line since IC may be destroyed due to short-circuit between outputs, air contamination fault, or fault by improper grounding.

(Note 3) Be careful when switching the input because rush current may occur.
When switching, stop mode should be entered or current limitation resister R should be inserted.

(Note 4) The IC functions cannot be guaranteed when turning power on of off.
Before using the IC for application, check that there are no problems.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(3) Additional diode

i) If the braking operation is so loose, connect a additional diode between each output to GND, (See Fig.3)

ii) If the back electromotive pulse generated in output coil is so strong.
Internally connected back electromotive suppression diode may be damaged by this pulse.
In such a case connect a additional diode between each output to V_{CC}. (See Fig.4)

iii) In case of mounted on radiators, do not use silicon rubber.

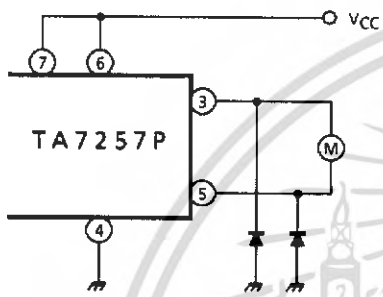


Fig.3

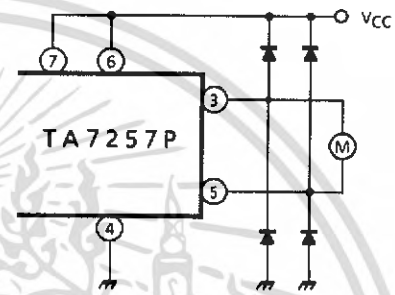
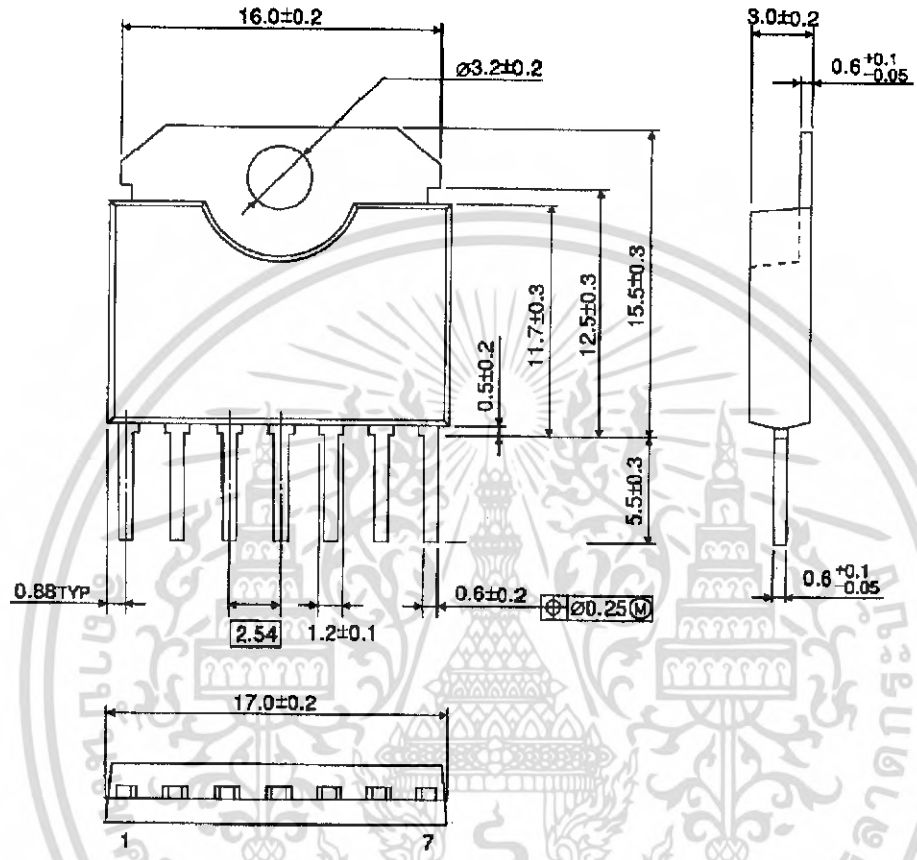


Fig.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OUTLINE DRAWING
HSIP7-P-2.54

Unit : mm



Weight : 1.88 g (Typ.)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS04
Hex Inverting Gates

General Description

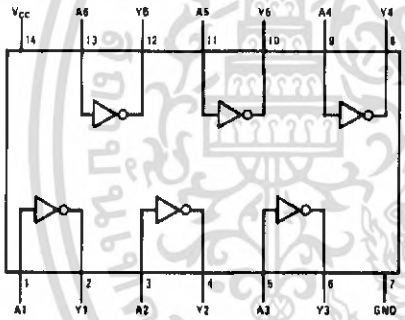
This device contains six independent gates each of which performs the logic INVERT function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS04M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS04SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS04N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code

Connection Diagram



Function Table

$Y = \overline{A}$

Input	Output
A	Y
L	H
H	L

H = HIGH Logic Level
L = LOW Logic Level

© 2000 Fairchild Semiconductor Corporation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings(Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	−65°C to +150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V_{CC}	Supply Voltage	4.75	5	5.25	V
V_{IH}	HIGH Level Input Voltage	2			V
V_{IL}	LOW Level Input Voltage			0.8	V
I_{OH}	HIGH Level Output Current			−0.4	mA
I_{OL}	LOW Level Output Current			8	mA
T_A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V_I	Input Clamp Voltage	V_{CC} = Min, I_I = −18 mA			−1.5	V
V_{OH}	HIGH Level Output Voltage	V_{CC} = Min, I_{OH} = Max, V_{IL} = Max	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	V_{CC} = Min, I_{OL} = Max, V_{IH} = Min		0.35	0.5	V
		I_{OL} = 4 mA, V_{CC} = Min		0.25	0.4	
I_I	Input Current @ Max Input Voltage	V_{CC} = Max, V_I = 7V			0.1	mA
I_{IH}	HIGH Level Input Current	V_{CC} = Max, V_I = 2.7V			20	µA
I_{IL}	LOW Level Input Current	V_{CC} = Max, V_I = 0.4V			−0.36	mA
I_{OS}	Short Circuit Output Current	V_{CC} = Max (Note 3)	−20		−100	mA
I_{CCH}	Supply Current with Outputs HIGH	V_{CC} = Max		1.2	2.4	mA
I_{CCL}	Supply Current with Outputs LOW	V_{CC} = Max		3.6	6.6	mA

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.

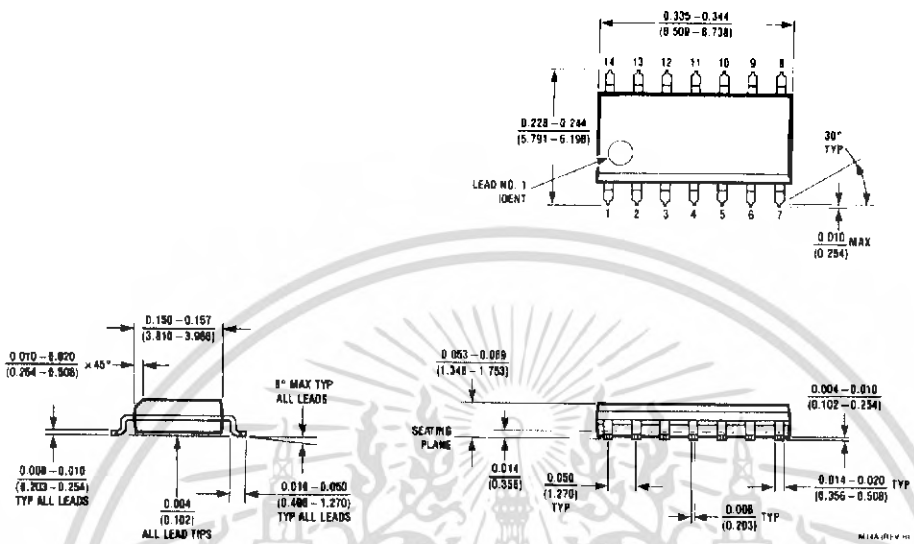
Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	R_L = 2 kΩ				Units
		C_L = 15 pF		C_L = 50 pF		
		Min	Max	Min	Max	
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	3	10	4	15	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	3	10	4	15	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

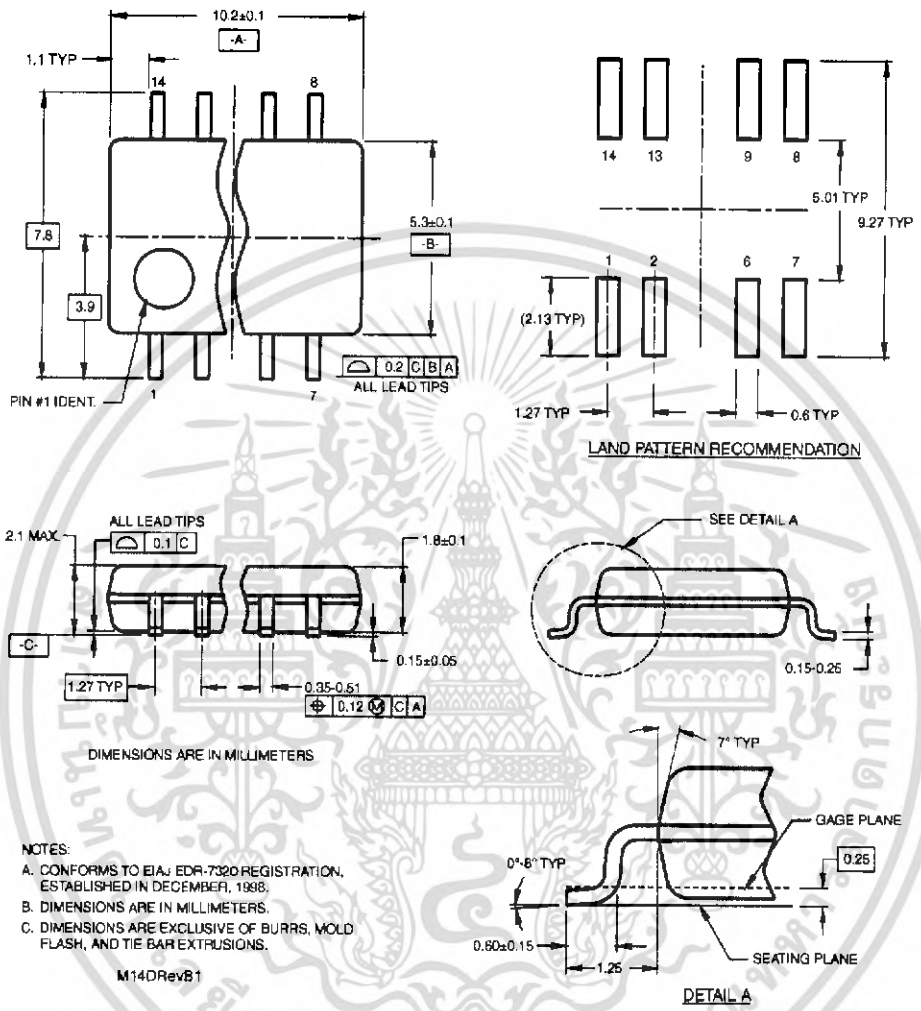
Physical Dimensions inches (millimeters) unless otherwise noted



14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
Package Number M14A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



NOTES:

A. CONFORMS TO EIAJ EDR-7320 REGISTRATION, ESTABLISHED IN DECEMBER, 1998.

B. DIMENSIONS ARE IN MILLIMETERS.

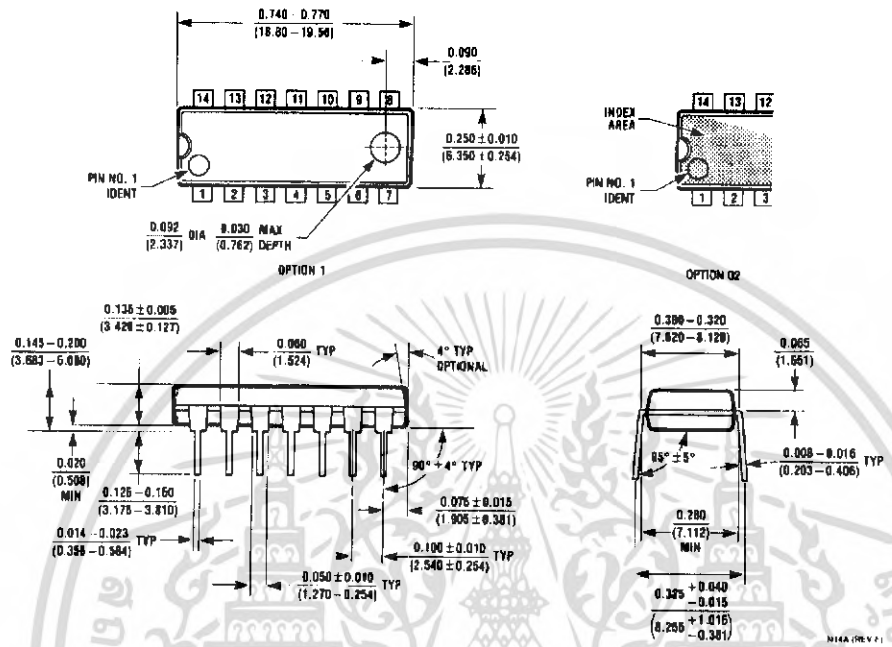
C. DIMENSIONS ARE EXCLUSIVE OF BURRS, MOLD FLASH, AND TIE BAR EXTRUSIONS.

M14DRevB1

14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
Package Number M14D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Physical Dimensions inches (millimeters) unless otherwise noted (Continued)



14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide
Package Number N14A

Fairchild does not assume any responsibility for use of any circuitry described, no circuit patent licenses are implied and Fairchild reserves the right at any time without notice to change said circuitry and specifications.

LIFE SUPPORT POLICY

FAIRCHILD'S PRODUCTS ARE NOT AUTHORIZED FOR USE AS CRITICAL COMPONENTS IN LIFE SUPPORT DEVICES OR SYSTEMS WITHOUT THE EXPRESS WRITTEN APPROVAL OF THE PRESIDENT OF FAIRCHILD SEMICONDUCTOR CORPORATION. As used herein:

1. Life support devices or systems are devices or systems which, (a) are intended for surgical implant into the body, or (b) support or sustain life, and (c) whose failure to perform when properly used in accordance with instructions for use provided in the labeling, can be reasonably expected to result in e significant injury to the user.
2. A critical component in any component of a life support device or system whose failure to perform can be reasonably expected to cause the failure of the life support device or system, or to affect its safety or effectiveness.

www.fairchildsemi.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้