



การควบคุมการเคลื่อนที่ของแขนกลโดยใช้คอมพิวเตอร์ในการควบคุม
THE ROBOT ARM'S MOTION CONTROL BY COMPUTER

โดย

นาย ภาคภูมิ เพียรมี
นาย วีรศักดิ์ เลิศนิพนธ์พันธุ์
นาย ศุภกิจ วงศ์สว่างธรรม

อาจารย์ที่ปรึกษา
อาจารย์ชั้นภัทร นันทจิวารชย์

วัน เดือน ปี 1 ค.ศ. ๒๕๔๐
เลขทะเบียน ๐๓๗๑๑๘
เลขเรียกหนังสือ T ๓๘๒๑๑ ๑๕๕๑ ก

ปริญญาานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา ๒๕๓๘

ปริญญาโท ประจำปีการศึกษา 2538

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่องการควบคุมการเคลื่อนที่ของแขนกลโดยใช้คอมพิวเตอร์ในการควบคุม

ผู้จัดทำ

1. นาย ภาคภูมิ เพียรมี
2. นาย วีรศักดิ์ เลิศนิพนธ์พันธุ์
3. นาย สุภกิจ วงศ์สว่างธรรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการเคลื่อนที่ของแขนกลโดยใช้คอมพิวเตอร์ในการควบคุม

THE ROBOT ARM'S MOTION CONTROL BY COMPUTER

นาย ภาคภูมิ เพ็ชรมี รหัส 35104314

นาย วีรศักดิ์ เลิศนิพนธ์พันธุ์ รหัส 35104404

นาย ศุภกิจ วงศ์สว่างธรรม รหัส 35104421

โครงการได้รับการตรวจสอบแล้ว พร้อมทั้งจะทำการสอบได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมการเคลื่อนที่ของแขนกลโดยใช้คอมพิวเตอร์ในการควบคุม

ภาคภูมิ เพียรมี

วีรศักดิ์ เลิศนิพนธ์พันธุ์

ศุภกิจ วงศ์สว่างธรรม

อ.ชินภัทร นันทจิวงกรชัย อาจารย์ที่ปรึกษา

ปีการศึกษา 2538

บทคัดย่อ

โครงการนี้เป็นชุดทดลองการควบคุมตำแหน่งของแขนกลโดยใช้คอมพิวเตอร์และอุปกรณ์ป้อนสัญญาณกลับเพื่อทำการทดลองหาความแม่นยำของระยะการเคลื่อนที่ ตลอดจนศึกษาตัวแปรต่างๆที่มีผลกระทบต่อการควบคุมระยะการเคลื่อนที่ของกระบอกลูกสูบและมอเตอร์ไฟฟ้ากระแสตรงเพื่อเป็นข้อมูลสำหรับนำไปประยุกต์ใช้ในงานต่อไป

THE ROBOT ARM'S MOTION CONTROL BY COMPUTER

Pakpom Preamme

Weerasak Lertniphonphun

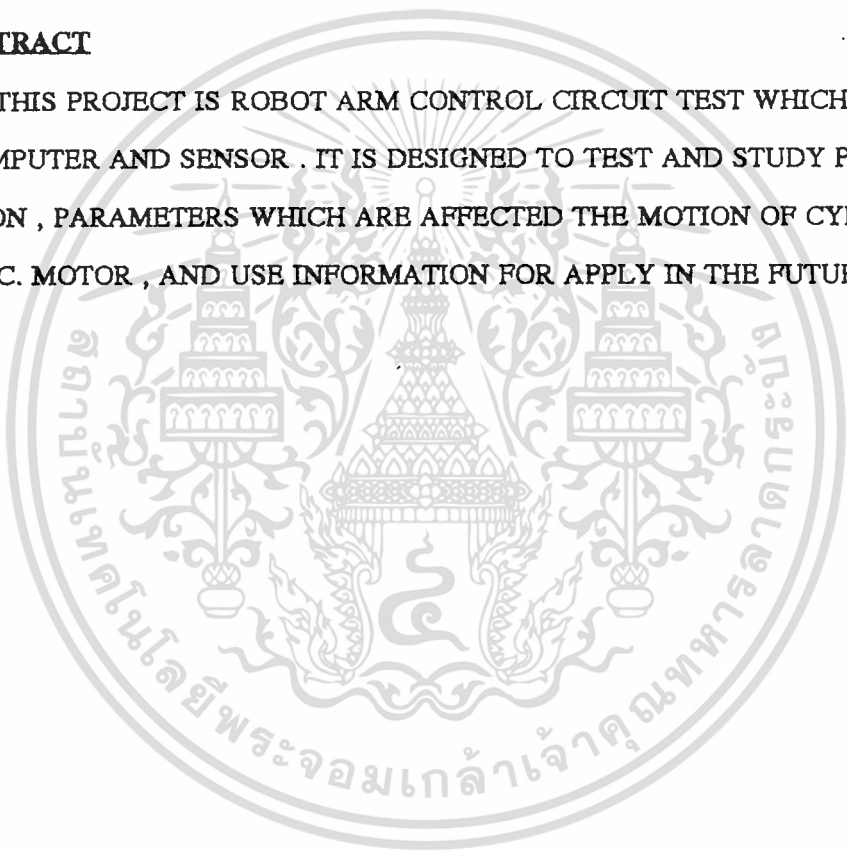
Supakit Wongswangtum

Chinapat Nantagiwakronchai Advisor

1995

ABSTRACT

THIS PROJECT IS ROBOT ARM CONTROL CIRCUIT TEST WHICH CONTROL BY COMPUTER AND SENSOR . IT IS DESIGNED TO TEST AND STUDY PRICISE POSITION , PARAMETERS WHICH ARE AFFECTED THE MOTION OF CYLINDER AND D.C. MOTOR , AND USE INFORMATION FOR APPLY IN THE FUTURE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

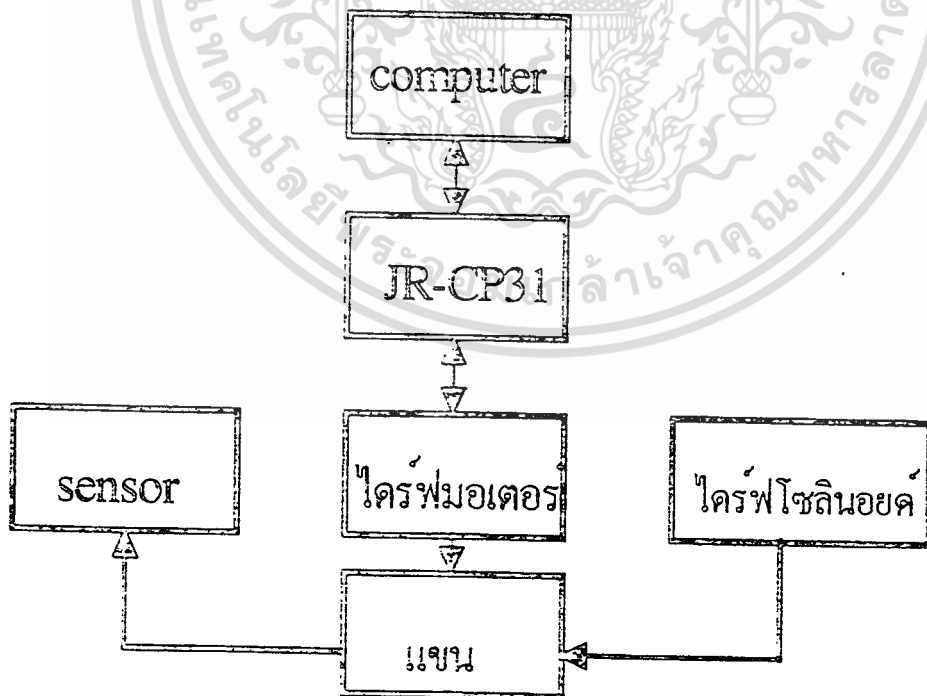
บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีที่เกี่ยวข้องกับแขนกล	3
2.1	ลักษณะโครงสร้างและการคำนวณหาตำแหน่งของแขนกล	3
2.2	อุปกรณ์นิวแมติกที่ใช้ขับเคลื่อนและควบคุมแขนกล	7
2.3	อุปกรณ์ทางด้านอิเล็กทรอนิกส์	9
2.3.1	ชนิดของ Sensor	9
2.3.2	บอร์ด CP-JR 31	12
2.3.3	มอเตอร์กระแสตรง	13
2.3.4	โซลินอยด์	18
2.3.5	การอินเตอร์เฟซ	22
2.3.6	วงจร ADC	27
บทที่ 3	การประกอบแขนกลเพื่อใช้ในการทดลอง	37
3.1	หลักการทํางานของแขนกล	37
3.2	ขนาดและโครงสร้างของแขนกลนิวแมติก	40
3.3	การคำนวณตำแหน่งของแขนกลนิวแมติก	45
3.4	วงจรมิวแมติกที่ใช้ควบคุมแขนกล	48
3.5	วงจรไคร์ฟและวงจรมีเปลี่ยนจากอนาลอกเป็นดิจิตอล	50
บทที่ 4	หลักการออกแบบและโปรแกรมของการทำงาน	54
4.1	แนวคิดของการทำงานของแขนกล	54
4.2	แนวคิดของการทำงานของโปรแกรมบนคอมพิวเตอร์	56
4.3	แนวคิดของการทำงานโปรแกรมของ 8031	62
4.4	โปรแกรมควบคุม	65
บทที่ 5	การทดลองควบคุมแขนกลด้วยคอมพิวเตอร์	66
บทที่ 6	สรุปผลการทดลอง	73
ภาคผนวก		75
เอกสารอ้างอิง		123
กิตติกรรมประกาศ		124

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1 บทนำ

แขนกลในปัจจุบันนิยมใช้ในงานอุตสาหกรรมเป็นจำนวนมาก แขนกลประเภทหนึ่งที่นิยมใช้กันอย่างแพร่หลาย เนื่องจากประหยัด ทำงานในลักษณะที่ซ้ำแบบเดิมได้อย่างรวดเร็ว แทนการใช้คนซึ่งอาจก่อให้เกิดความเบื่อหน่าย แต่ข้อเสียคือการทำงานที่ไม่เที่ยงตรง เนื่องจากลมเป็นสารที่สามารถอัดและขยายได้ง่าย จึงต้องมีส่วนควบคุมเพื่อให้ทำงานได้อย่างเที่ยงตรงและแม่นยำ

รูปที่ 1.1 แสดงถึงบล็อกไดอะแกรมของการทำงานคร่าว ๆ ของแขนกล โดยการควบคุมทั้งหมดจะส่งมาจากคอมพิวเตอร์ที่เขียนโปรแกรมไว้แล้วผ่านพอร์ตอนุกรมมายังบอร์ด JR-CP31 ซึ่งโปรเซสเซอร์หลักของบอร์ดนี้คือ 8031 โดยหน้าที่หลักคือ การไครฟ์มอเตอร์ การไครฟ์โซลินอยด์วาล์ว และ รับข้อมูลจากวงจรตรวจจับตำแหน่ง (sensor) การไครฟ์มอเตอร์เพื่อให้แขนเคลื่อนที่ไปตามแนวระนาบ ส่วนการไครฟ์โซลินอยด์วาล์วเพื่อให้แขนเคลื่อนที่ไปตามแนวระดับ เมื่อได้มีการสั่งให้เคลื่อนไปตำแหน่งที่ต้องการแล้วก็ต้องมีการตรวจจับตำแหน่งเพื่อให้ได้ตำแหน่งที่แน่นอน วงจรตรวจจับตำแหน่งก็จะส่งตำแหน่งของแขน ณ ตอนนั้นไปคำนวณว่าได้ตำแหน่งตามต้องการหรือไม่เพื่อความแม่นยำที่สุด



รูปที่ 1.1 แสดงบล็อกไดอะแกรมของแขนอย่าง ๆ คร่าว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาแบ่งออกเป็น 2 ส่วนหลัก ๆ คือ

1. ด้านอุปกรณ์ (HARDWARE)

1.1 ส่วนของโครงสร้างทางกลได้แก่ การคำนวณ , การออกแบบโครงสร้างทางแขน , การติดตั้งอุปกรณ์ต่าง ๆ

1.2 การควบคุมการทำงานของแขนกล

- ในส่วนของไฮดรอนิวแมติก ใช้วาล์วควบคุมการทำงานของวงจร โดยการควบคุมผ่านเครื่องคอมพิวเตอร์

- อุปกรณ์ตรวจจับต่าง ๆ (SENSOR หรือ TRANSDUCER) เพื่อใช้ควบคุมตำแหน่งที่แน่นอน

2. ด้านโปรแกรม (SOFTWARE)

2.1 ตัวโปรแกรม

- ควบคุมการทำงานของแขนกลผ่านวงจรถูกควบคุม
- รับผลจากตัวอุปกรณ์ตรวจจับต่าง ๆ เพื่อนำมาใช้เป็นมาใช้เป็นข้อมูลในการควบคุม
- แสดงผลการทำงานบนหน้าจอกอมพิวเตอร์

2.2 รูปแบบของการทำงาน

- การควบคุมแขนจากคีย์บอร์ด
- การควบคุมแขนจากการป้อนตำแหน่ง
- การควบคุมแขนจากอ่านตำแหน่งที่ป้อนไว้ในไฟล์ข้อมูล

บทที่ 2 ทฤษฎีและทฤษฎีที่เกี่ยวข้องกับแขนกล

2.1 ลักษณะโครงสร้างและการคำนวณตำแหน่งของแขนกล

โครงสร้างของแขนกล

แขนกลจะประกอบไปด้วย

- ส่วนฐาน

ซึ่งจะทำหน้าที่สำคัญ คือ เป็นที่ติดตั้งโครงแขนกล และอุปกรณ์ควบคุมต่างๆของแขนกล และยังมีหน้าที่ในการหมุนเพื่อให้แขนกลหมุนไปเป็นมุมต่างๆตามที่ต้องการ เนื่องจากการติดตั้งอุปกรณ์มอเตอร์กระแสตรง (D.C. motor) ไว้ที่ฐานของแขนกลด้วย

- โครงของแขนกล

- 1 . โครงแขนชั้นที่1 ซึ่งต่ออยู่กับฐาน จะมีกระบอกสูบ (cylinder) ติดอยู่เพื่อทำการเปลี่ยนมุม θ_1
- 2 . โครงแขนชั้นที่ 2 จะทำมุมกับแนวระดับคงที่ตลอดเวลาไม่ว่าแขนกลจะขยับไปอยู่ในตำแหน่งใดก็ตาม เนื่องจากมีโครงแขนชั้นที่4 เป็นตัวช่วยดึงให้อยู่ในลักษณะที่ทำมุมกับแนวระดับคงที่ตลอดเวลา
- 3 . โครงแขนชั้นที่3 ต่ออยู่บนโครงแขนชั้นที่2 และมีกระบอกสูบบังคับการเปลี่ยนค่ามุม θ_2 ที่เป็นมุมระหว่างโครงแขนชั้นที่2 กับโครงแขนชั้นที่3
- 4 . โครงแขนชั้นที่ 4 ต่ออยู่บนฐาน กับโครงแขนชั้นที่ 2 ทำหน้าที่ดึงให้โครงแขนชั้นที่ 2 ทำมุมคงที่กับแนวระดับตลอดเวลา
- 5 . โครงแขนชั้นที่ 5 ต่ออยู่บนโครงแขนชั้นที่ 2 ทำหน้าที่คอยดึงให้มือจับอยู่ในลักษณะที่อยู่ในแนวระดับตลอดเวลา เพื่อให้มือจับสิ่งของได้โดยไม่ทำให้สิ่งของเอียงไปมา ขณะที่แขนกลเคลื่อนที่เปลี่ยนตำแหน่งไป

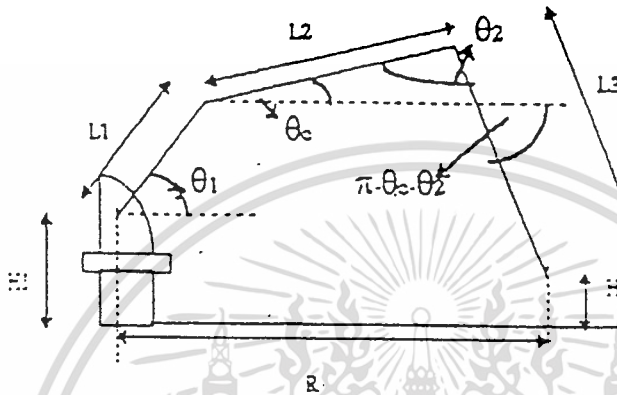
* การติดระบบตรวจจับ (SENSOR) นั้นจะคิดไว้ 3 ตำแหน่ง คือ ที่ฐานหมุนเพื่อไว้ตรวจสอบการหมุนของมอเตอร์ ที่จุดต่อระหว่างฐานกับโครงแขนชั้นที่ 1 และที่จุดต่อระหว่างโครงแขนชั้นที่ 2 กับ 3 เพื่อไว้ตรวจสอบมุมที่เปลี่ยนแปลงไป *

- ส่วนของมือจับ

จะต่ออยู่บนโครงแขนชั้นที่3 จะทำหน้าที่หลักคือ จับและวางสิ่งของที่ต้องการเคลื่อนย้ายไปที่ตำแหน่งต่างๆตามที่ต้องการ ซึ่งมือจับนี้จะมีส่วนประกอบหลักคือ เป็น compact cylinder ที่มีระยะ stroke 3 c.m.

สมการที่ใช้ในการคำนวณเพื่อให้แขนกลไปในตำแหน่งที่ต้องการ

โดยทั่วไปในการควบคุมแขนกลให้เคลื่อนไปในตำแหน่งที่ต้องการนั้น จะต้องมีตำแหน่งอ้างอิงของมุมต่างๆของข้อต่อ เพื่อให้แขนกลไปได้ในตำแหน่งที่ต้องการภายในพื้นที่ทำการ ซึ่งในการทดลองครั้งนี้ลักษณะ โครงร่างของแขนกลจะเป็นดังนี้



- R เป็นระยะระหว่างจุดหมุนที่ฐานกับมือจับ
- H เป็นระยะความสูงของมือจับจากพื้นโต๊ะ
- θ_1 เป็นมุมที่โครงเหล็กท่อนที่ 1 ทำมุมกับแนวระดับ
- θ_2 เป็นมุมที่โครงเหล็กท่อนที่ 3 ทำมุมกับโครงเหล็กท่อนที่ 2
- θ_c เป็นมุมที่โครงเหล็กท่อนที่ 2 ทำมุมกับแนวระดับมีค่าคงที่
- H เป็นค่าความสูงของจุดหมุนที่ฐานจากพื้นโต๊ะมีค่าคงที่

ในการควบคุมตำแหน่งมุมของข้อต่อต่างๆให้เหมาะสมกับตำแหน่ง R กับ H นั้น ถ้าใช้สมการทางคณิตศาสตร์จะมีอยู่ 2 ลักษณะคือ สมการ direct kinematic ซึ่งจะเป็นการป้อนค่าของมุมของแต่ละข้อต่อเข้าไปแล้วทำการคำนวณหาระยะ R กับ H ออกมา กับสมการ inverse kinematic ซึ่งจะเป็นการป้อนค่า R กับ H เข้าไปแล้วทำการคำนวณหาค่ามุม θ_1 กับ θ_2 ออกมา

ในการทดลองครั้งนี้จะใช้สมการแบบ inverse kinematic ในการทดลองเนื่องจาก สิ่งที่ต้องการคือตำแหน่งมุมของข้อต่อที่สอดคล้องกับระยะตำแหน่งของแขนกลที่ต้องการเคลื่อนที่ไป

สมการที่ใช้คำนวณ

จากรูปโครงร่างแขนกลจะได้

$$L1 \cos\theta_1 + L2 \cos\theta_c + L3 \cos(\pi - \theta_c - \theta_2) = R \quad \text{-----①}$$

$$L3 \sin(\pi - \theta_c - \theta_2) - L2 \sin\theta_c - L1 \sin\theta_1 = H - H \quad \text{-----②}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{จาก } \textcircled{1} ; R = L1 \cos\theta_1 + L2 \cos\theta_c + L3 \cos(\pi - \theta_c) \cos\theta_2 \\ + L3 \sin(\pi - \theta_c) \sin\theta_2$$

$$\text{จาก } \textcircled{2} ; H1 - H = L3 \sin(\pi - \theta_c) \cos\theta_2 - L3 \cos(\pi - \theta_c) \sin\theta_2 - L2 \sin\theta_c \\ - L1 \sin\theta_1$$

$$\text{ให้ } L3 \sin(\pi - \theta_c) = C1$$

$$L3 \cos(\pi - \theta_c) = C2$$

$$R - L2 \cos\theta_c = A$$

$$H1 - H + L2 \sin\theta_c = B$$

จะได้ว่า

$$R = L1 \cos\theta_1 + L2 \cos\theta_c + C2 \cos\theta_2 + C1 \sin\theta_2$$

$$L1 \cos\theta_1 = (R - L2 \cos\theta_c) - (C2 \cos\theta_2 + C1 \sin\theta_2)$$

$$= A - (C2 \cos\theta_2 + C1 \sin\theta_2) \quad \text{-----} \textcircled{3}$$

$$H1 - H = C1 \cos\theta_2 - C2 \sin\theta_2 - L2 \sin\theta_c - L1 \sin\theta_1$$

$$L1 \sin\theta_1 = (C1 \cos\theta_2 - C2 \sin\theta_2) - (H1 - H + L2 \sin\theta_c)$$

$$= (C1 \cos\theta_2 - C2 \sin\theta_2) - B \quad \text{-----} \textcircled{4}$$

ยกกำลังสองสมการ $\textcircled{3}$ และ $\textcircled{4}$ จะได้

$$(L1 \cos\theta_1)^2 = A^2 - 2A(C2 \cos\theta_2 + C1 \sin\theta_2) + (C2 \cos\theta_2)^2 \\ + 2C1C2 \sin\theta_2 \cos\theta_2 + (C1 \sin\theta_2)^2 \quad \text{-----} \textcircled{5}$$

$$(L1 \sin\theta_1)^2 = (C1 \cos\theta_2)^2 - 2C1C2 \sin\theta_2 \cos\theta_2 + (C2 \sin\theta_2)^2 \\ - 2B(C1 \cos\theta_2 - C2 \sin\theta_2) + B^2 \quad \text{-----} \textcircled{6}$$

นำ $\textcircled{5}$ + $\textcircled{6}$;

$$L1^2 = (C1^2 + C2^2) \cos^2\theta_2 + (C1^2 + C2^2) \sin^2\theta_2 - 2AC2 \cos\theta_2 - 2AC1 \sin\theta_2 \\ - 2BC1 \cos\theta_2 + 2BC2 \sin\theta_2 + A^2 + B^2 \\ = (C1^2 + C2^2) - (2AC2 + 2BC1) \cos\theta_2 + (2BC2 - 2AC1) \sin\theta_2 + A^2 + B^2$$

$$\text{ให้ } L1^2 - A^2 - B^2 - C1^2 - C2^2 = Z1$$

เอกสารนี้เป็น $2AC2 + 2BC1 = Z2$ ทรัพยากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$2BC2 - 2AC1 = Z3$$

$$Z1 = Z3 \sin\theta_2 - Z2 \cos\theta_2$$

$$Z1 = Z3 (1 - \cos^2\theta_2)^{1/2} - Z2 \cos\theta_2$$

$$Z1^2 + 2Z1Z2 \cos\theta_2 + (Z2 \cos\theta_2)^2 = Z3^2 - (Z3 \cos\theta_2)^2$$

$$(Z2^2 + Z3^2)\cos^2\theta_2 + 2Z1Z2 \cos\theta_2 + (Z1^2 - Z3^2) = 0$$

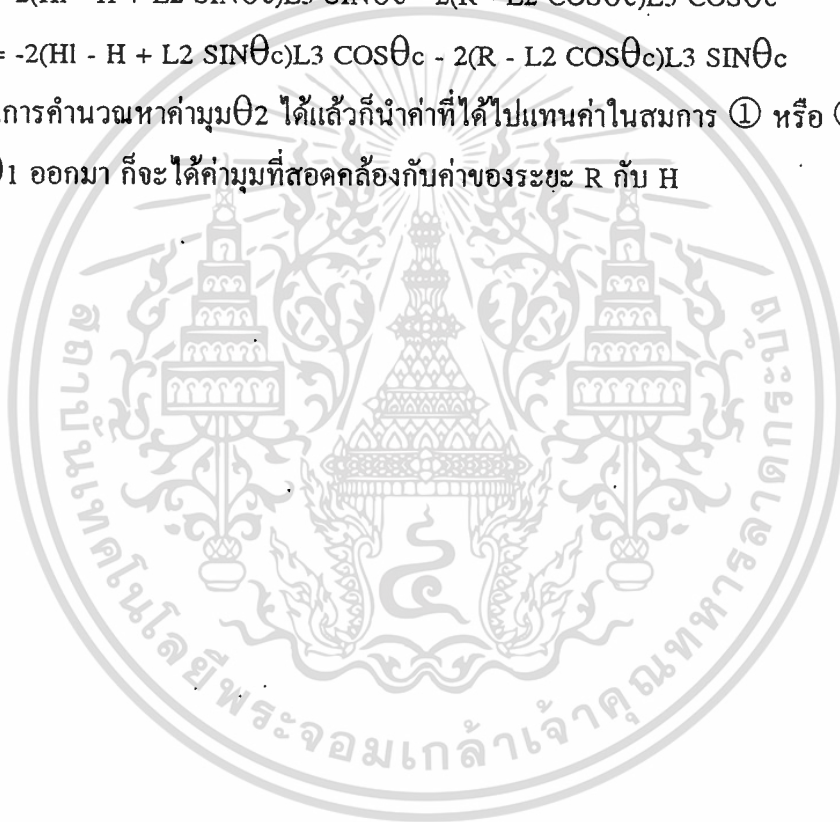
$$\cos\theta_2 = \{-2Z1Z2 \pm [(4Z1^2Z2^2) - 4(Z1^2 - Z3^2)(Z2^2 + Z3^2)]^{1/2}\} / 2(Z2^2 + Z3^2)$$

$$Z1 = L1^2 - (R - L2 \cos\theta_c)^2 - (H1 - H + L2 \sin\theta_c)^2 - L3^2$$

$$Z2 = 2(H1 - H + L2 \sin\theta_c)L3 \sin\theta_c - 2(R - L2 \cos\theta_c)L3 \cos\theta_c$$

$$Z3 = -2(H1 - H + L2 \sin\theta_c)L3 \cos\theta_c - 2(R - L2 \cos\theta_c)L3 \sin\theta_c$$

เมื่อทำการคำนวณหาค่ามุม θ_2 ได้แล้วก็นำค่าที่ได้ไปแทนค่าในสมการ ① หรือ ② เพื่อหาค่าของมุม θ_1 ออกมา ก็จะได้ค่ามุมที่สอดคล้องกับค่าของระยะ R กับ H



2.2 อุปกรณ์นิวแมติกที่ใช้ขับเคลื่อนและควบคุมแขนกล

กระบอกลูกสูบ

กระบอกลูกสูบจะทำหน้าที่เปลี่ยนรูปพลังงานลมอัดหรือน้ำมันไฮดรอลิกที่ถูกอัดให้เปลี่ยนเป็นพลังงานกล ลักษณะการเคลื่อนที่เป็นแบบแนวเส้นตรง ในสมัยก่อนที่ลูกสูบจะเข้ามามีบทบาทในงานอุตสาหกรรมยังใช้กลไกทางกลและทางไฟฟ้าซึ่งมีความยุ่งยากในการควบคุมและปัญหาของช่วงชักจำกัด ดังนั้นจึงมีการพัฒนาลูกสูบเข้ามาใช้ในงานจนถึงปัจจุบัน

กระบอกลูกสูบแบ่งตามลักษณะการทำงานได้ดังนี้

- 1.กระบอกลูกสูบทำงานทางเดียว
- 2.กระบอกลูกสูบชนิดทำงานสองทาง
- 3.กระบอกลูกสูบชนิดทำงานสองทางแบบมีก้านสูบสองข้าง
- 4.กระบอกลูกสูบชนิดทำงานสองทางแบบสองตอน
- 5.กระบอกลูกสูบชนิดทำงานสองทางแบบมีเบรคก้านสูบ
- 6.กระบอกลูกสูบชนิดช่วงชักหลายตำแหน่ง
- 7.กระบอกลูกสูบแบบกระแทก
- 8.กระบอกลูกสูบแบบก้านสูบอยู่กับที่ลูกสูบเคลื่อนที่
- 9.กระบอกลูกสูบชนิดพิเศษที่ใช้กับลักษณะเฉพาะ

เนื่องจากในโครงการครั้งนี้จะใช้กระบอกลูกสูบชนิดทำงานสองทางเท่านั้น ดังนั้นจึงขออธิบายถึงหลักการทำงานของกระบอกลูกสูบชนิดนี้เท่านั้น

กระบอกลูกสูบชนิดทำงานสองทาง ในการทดลองนี้จะใช้ลมอัดดันหัวลูกสูบทั้งสองตอนเคลื่อนที่ออกและตอนเคลื่อนที่กลับ ทำให้ได้แรงทั้งสองทิศทาง เหมาะกับงานที่จะต้องการใช้แรงในตอนลูกสูบเลื่อนออก และ เลื่อนเข้ารวมทั้งลักษณะงานที่ต้องการช่วงชักยาว ปัญหาที่เกิดขึ้นในกรณีช่วงชักยาวเกินไปจะทำให้ก้านสูบโก่งงอได้ ดังนั้นช่วงชักของกระบอกลูกสูบจึงต้องมีการคำนวณหาระยะช่วงชักที่อนุญาตให้ใช้งานได้ และนอกจากนี้การที่กระบอกลูกสูบมีเส้นผ่านศูนย์กลางที่โตเกินไปก็จะทำให้เกิดการสิ้นเปลืองปริมาณลมอัดมาก ดังนั้นในการใช้งานจึงจะต้องเลือกให้เหมาะสม

วาล์วควบคุมทิศทางและวาล์วควบคุมอัตราการไหล

ในระบบนิวแมติก และระบบไฮดรอลิกนั้นการที่จะควบคุมการเคลื่อนที่ของอุปกรณ์ทำงานให้เคลื่อนที่ไปในทิศทางที่ต้องการได้นั้น จำเป็นต้องอาศัยวาล์วต่างๆ ในการควบคุม ซึ่งชนิดของวาล์วควบคุมสามารถแบ่งได้ดังนี้

วาล์วในระบบนิวแมติก แบ่งตามหน้าที่การทำงานได้ 6 ประเภท คือ

1. วาล์วควบคุมทิศทางการไหลของลมอัด
2. วาล์วควบคุมอัตราการไหลของลมอัด
3. วาล์วควบคุมความดันลมอัด
4. วาล์วบังคับลมอัดไหลทางเดียว
5. วาล์วเปิดปิดลมอัด
6. วาล์วแบบผสม

วาล์วในระบบไฮดรอลิก แบ่งตามหน้าที่การทำงานได้ 3 ประเภท คือ

1. วาล์วควบคุมทิศทาง
2. วาล์วควบคุมความดัน
3. วาล์วควบคุมอัตราการไหล

ในโครงการทดลองครั้งนี้ จะใช้วาล์วเพียง 2 ประเภทเท่านั้น คือ วาล์วควบคุมทิศทาง กับ วาล์วควบคุมอัตราการไหล ดังนั้นจึงขอกล่าวถึงหน้าที่ของวาล์วเพียง 2 ชนิดนี้เท่านั้น

-วาล์วควบคุมทิศทางการไหล มีหน้าที่เลือกทิศทางการไหลของลมอัดหรือน้ำมันไฮดรอลิกให้ไปตามทิศทางที่ต้องการ ทั้งนี้เพื่อให้อุปกรณ์ทำงาน เช่น กระบอกสูบ สามารถทำงานได้และเคลื่อนที่ไปในทิศทางที่ต้องการได้โดยใช้หลักการเปิด-ปิด ลมอัด หรือน้ำมันไฮดรอลิกจากรูหนึ่งไปอีกรูหนึ่ง จำนวนรูของวาล์วที่ใช้ควบคุมทิศทางการไหลมีอยู่หลายแบบ เช่น 2,3,4,5 รู ซึ่งจะประกอบด้วยรูสำหรับท่อจ่ายที่ต่อไปยังกับอุปกรณ์ทำงานหรือนำไปใช้งาน และรูสำหรับระบายทิ้ง โดยทั่วไป วาล์วชนิดนี้นิยมที่จะเรียกว่า DIRECTIONAL CONTROL VALVES (D.C.V) ลักษณะโครงสร้างมีอยู่ 3 แบบ คือ แบบแผ่นเปิด-เปิด แบบลูกสูบเลื่อน และแบบแผ่นหมุน

-วาล์วควบคุมอัตราการไหล วาล์วชนิดนี้ทำหน้าที่ควบคุมปริมาณการไหลของลมอัด หรือน้ำมันไฮดรอลิกที่จะส่งไปยังอุปกรณ์ทำงานของระบบ เช่น กระบอกสูบ ทำให้สามารถควบคุมความเร็วของก้านสูบขณะทำงานได้ โดยการติดตั้งวาล์วควบคุมอัตราการไหลในท่อทางลมหรือน้ำมันไฮดรอลิกที่ต่อเข้าระหว่างกระบอกสูบกับวาล์วควบคุมทิศทางการไหล ระยะระหว่างวาล์วควบคุมทิศทางไปถึงวาล์วควบคุมอัตราการไหล และจากวาล์วควบคุมการไหลถึงอุปกรณ์ทำงาน ไม่ควรเกิน 1 เมตร

วาล์วควบคุมอัตราการไหลแบ่งออกได้ 3 แบบ คือ แบบปรับโดยมือหมุน แบบปรับโดยกลไก และแบบวาล์วลดการไหล

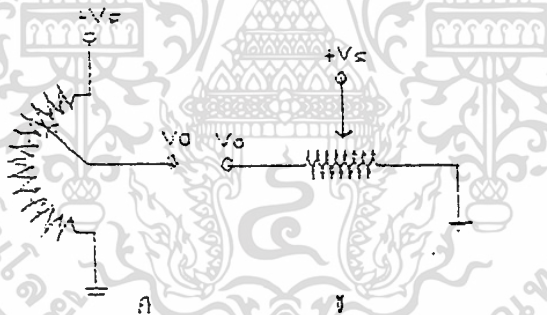
2.3 อุปกรณ์ทางด้านอิเล็กทรอนิกส์

2.3.1 ชนิดของ Sensor

ในการควบคุมหุ่นยนต์สิ่งหนึ่งที่ต้องทราบก็คือตำแหน่งและทิศทางของแต่ละข้อต่อ เพื่อที่จะได้นำไปคำนวณหาตำแหน่งของส่วนปลายแขน เช่น มือจับ, ตัวพ่น ,หัวเชื่อม เป็นต้น ซึ่งการเคลื่อนที่ของข้อต่อต่าง ๆ เป็นไปได้ทั้งการหมุนเพื่อเปลี่ยนมุมมองหรือการเคลื่อนที่เป็นเส้นตรง ทั้งนี้ขึ้นอยู่กับชนิดของหุ่นยนต์ที่ใช้ฐานด้วย ซึ่งแต่ละชนิดก็มีสูตรการคำนวณแตกต่างกัน สิ่งหนึ่งที่ขาดไม่ได้สำหรับการหาตำแหน่งของหุ่นยนต์ก็คือทรานสดิวเซอร์บอกตำแหน่ง (position transducer) ซึ่งสามารถแบ่งได้เป็น 2 ชนิดคือ แบบสมบูรณ์ในตัวเอง (absolute) และแบบต้องต่อร่วมกับวงจรอื่นเพิ่มเติม (incremental)

โพเทนชิโอมิเตอร์ (potentiometer)

เป็นอุปกรณ์ที่แปลตำแหน่งของแกนเลื่อนที่เป็นผลมาจากการเปลี่ยนตำแหน่งของหุ่นยนต์ไปเป็นสัญญาณไฟฟ้าโดยใช้หลักการแบ่งแรงดัน ซึ่งแบ่งออกได้ 2 แบบคือ แบบเชิงมุม (angular) และแบบเชิงเส้น (linear) ดังแสดงในรูปที่ 2.1 ค่าของแรงดันเอาต์พุต (V_o) จะขึ้นอยู่กับตำแหน่งของแกนเลื่อน สามารถคำนวณหาได้ตามสมการ



รูปที่ 2.1 (ก) โพเทนชิโอมิเตอร์แบบเชิงมุม

(ข) โพเทนชิโอมิเตอร์แบบเชิงเส้น

จากรูปที่ 2.1(ก)

$$V_o = V_s R_\theta / R_{tot} = V_s \theta / \theta_{tot} = K\theta$$

และจากรูปที่ 2.1(ข)

$$V_o = V_s R_x / R_{tot} = V_s X / X_{tot} = KX$$

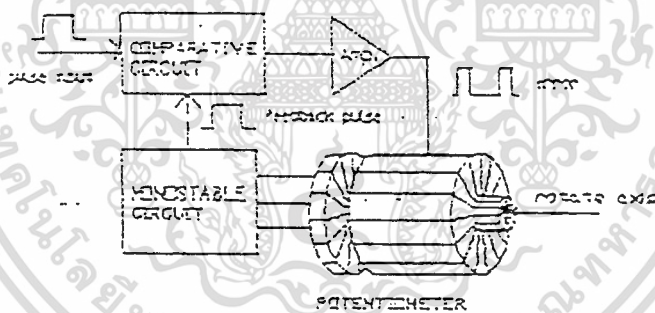
เมื่อ R_θ คือ ค่าความต้านทาน ณ ตำแหน่งที่หมุนไป

R_x คือ ค่าความต้านทาน ณ ตำแหน่งที่เลื่อนไป

K คือ ค่าคงที่

ชนิดของแถบความต้านทานที่ใช้ทำโพเทนชิโอมิเตอร์มีด้วยกันหลายชนิดแตกต่างกันไป เช่น ชนิดฟิล์มคาร์บอน (carbon film) ก็จะมีราคาถูก แต่ถ้าเป็นชนิดฟิล์มเซอรัม (cermet film) ก็จะแพงขึ้นนิดหน่อยแต่จะลดสัญญาณรบกวนบางอย่างได้ ส่วนชนิดไวร์วาวด์ (wire wound) นั้นสามารถทนกำลังไฟฟ้าได้สูง

ข้อดีของการใช้โพเทนชิโอมิเตอร์ก็คือมีขนาดเล็กและราคาถูก นิยมนำมาใช้งานโดยเฉพาะกับหุ่นยนต์ขนาดเล็ก แต่ข้อเสียก็คือหน้าสัมผัสของแถบความต้านทานเกิดการสึกหรองได้ง่าย ทำให้อายุการใช้งานสั้นและอีกประการหนึ่ง โพเทนชิโอมิเตอร์เป็นอุปกรณ์อะนาลอกทำให้การติดต่อกับคอมพิวเตอร์ต้องมีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิทัล (ADC) ในบล็อกไดอะแกรมรูปที่ 2.2 แสดงการควบคุมตำแหน่งโดยการใช้โพเทนชิโอมิเตอร์ ค่าความต้านทานจะขึ้นอยู่กับตำแหน่งของหุ่นยนต์และจะถูกนำมาใช้กำหนดความกว้างของพัลส์ในวงจรโมโนสเตเบิล ผลิตพัลส์ป้อนกลับมาเปรียบเทียบกับพัลส์อินพุตให้เกิดเป็นสัญญาณผิดพลาดออกทางเอาต์พุต โดยความกว้างของสัญญาณผิดพลาดนี้จะเป็นตัวกำหนดคอนโทรลเลอร์และจะหมุนไปในทิศทางที่ถูกบังคับไว้จนกว่าพัลส์อินพุตและพัลส์ป้อนกลับจะมีความกว้างของพัลส์เท่ากัน



รูปที่ 2.2 บล็อกไดอะแกรมแสดงการใช้โพเทนชิโอมิเตอร์ในการควบคุมตำแหน่ง

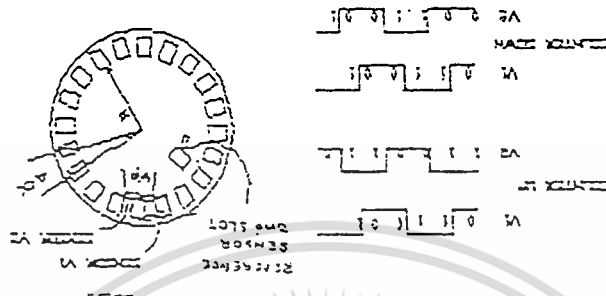
ออปติคัลเอนโค้ดเดอร์ (optical encoders)

เหตุผลหนึ่งที่มีการใช้ออปติคัลเอนโค้ดเดอร์แทนโพเทนชิโอมิเตอร์ก็คือ อายุการใช้งานที่ยาวนานกว่า เพราะไม่มีส่วนที่จะสึกหรองง่าย ออปติคัลเอนโค้ดเดอร์แบ่งออกได้เป็น 2 ลักษณะคือ

- 1 แบบสมบูรณ์ในตัวเอง (absolute) ซึ่งแบ่งได้เป็น 2 แบบคือ แบบไบนารี (binary) และแบบรหัสเกรย์ (gray code)
- 2 แบบไม่สมบูรณ์ในตัวเอง (incremental) ซึ่งในที่นี้จะกล่าวเฉพาะตัวนี้เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออปติคัลเอนโค้ดเดอร์แบบไม่สมบูรณ์ในตัวเอง



รูปที่ 2.3 ลักษณะของออปติคัลเอนโค้ดเดอร์แบบไม่สมบูรณ์ในตัวเอง และลักษณะการตรวจจับจากตัวเซนเซอร์ V_1 , V_2 , V_3

ออปติคัลเอนโค้ดเดอร์แบบนี้จะมีรูปร่างดังในรูปที่ 2.3 ความละเอียดของตำแหน่งมุม ($\Delta\alpha$) จะขึ้นอยู่กับจำนวนของช่อง (n) บนแผ่นจาน สำหรับความกว้างของโฟโต้ไดโอด (w_p) ที่ใช้ก็จะขึ้นอยู่กับค่าความละเอียดของตำแหน่งมุมระยะห่างจากจุดศูนย์กลางถึงโฟโต้ไดโอด (r) ซึ่งเป็นไปดังสมการ

$$\Delta\alpha = 360^\circ / n$$

และ

$$w_p = r \sin(\Delta\alpha/2) = r \sin(360^\circ/2n)$$

ยกตัวอย่างเช่น จานเอนโค้ดเดอร์ขนาด 120 ช่องและมี $\Delta\alpha = 3$ องศา ดังนั้นความกว้างของโฟโต้ไดโอดที่ใช้จะต้องไม่เกิน 0.26 มิลลิเมตร ที่ระยะ = 10 มิลลิเมตร จำนวน LED และโฟโต้ไดโอดที่ใช้จะมีทั้งหมด 3 คู่เสมอ ไม่ว่าจำนวนช่องจะมีมากหรือน้อยก็ตาม โดย 2 คู่แรกจะวางที่ตำแหน่งของช่องและวางห่างกันเท่ากับครึ่งหนึ่งของความกว้างของช่องดังรูปที่ 2.3 เมื่อแผ่นจานหมุนสัญญาณจาก V_1 และ V_2 จะต่างเฟสกัน 90 องศาเสมอ ถ้าเป็นการหมุนตามเข็มนาฬิกาสัญญาณจาก V_1 นำหน้าสัญญาณ V_2 อยู่ 90 องศา เมื่อหมุนแบบทวนเข็มนาฬิกาสัญญาณจาก V_1 ก็จะเปลี่ยนกลับมาเป็นตามหลังสัญญาณ V_2 อยู่ 90 องศา ความแตกต่างนี้จะเป็นตัวชี้ถึงทิศทางการหมุนได้ สำหรับ LED โฟโต้ไดโอดคู่ที่เหลือก็จะถูกจัดวางให้ตรงตำแหน่งของสล็อตอ้างอิง (reference slot) เพื่อกำหนดตำแหน่งมุมโดยการนับพัลส์ที่เกิดขึ้นเริ่มจากสล็อตอ้างอิง

2.3.2 บอร์ด CP-JR31

1. CPU 8031

2. Memory

- มี Socket สำหรับหน่วยความจำ 1 ตัว โดยมีขนาดตั้งแต่ 8KB ถึง 32 KB
- หน่วยความจำประเภท RAM อีก 256 ไบท์ อยู่ในพอร์ต 8155

3. I/O Port

- 8 Bit Bidirection 2 Port (8155 A,B)
- 6 Bit Bidirection 1 Port (8155 C)
- 8 Bit Bidirection 1 Port (P1 of CPU)
- Time/counter 2 channel (3 channel for 8032,8052AH)
- Time/counter 1 channel (8155)

4. Serial Port

- RS232C (Max 232) ระยะการทำงานหวังผล 50 ฟุต
- RS422 (SN75176) ระยะการทำงานหวังผล 1200 ฟุต

ซึ่งจะใช้ CP-JR31 เป็นตัวเชื่อมระหว่างคอมพิวเตอร์กับวงจร ข้อมูลที่ออกจาก CP-JR31 นั้นจะออกมาเป็นข้อมูลแบบ Parallel ส่งไปยังวงจรโคร์ฟซึ่งจะเปลี่ยนแรงดันไฟฟ้าจาก 5V เป็น 24V โดยส่งไปยังโซลินอยด์ซึ่งควบคุมกระสวยลมนิวแมติกซึ่งจะใช้สาย 2 เส้นจาก CP-JR31 มาควบคุม แต่โซลินอยด์ซึ่งควบคุมส่วนมือจับนั้นจะใช้สายเพียงเส้นเดียวเนื่องจากต้องการแคบิบหรือปล่อยเท่านั้น ส่วนของมอเตอร์ซึ่งใช้ควบคุมการหมุนฐานนั้น จะใช้สาย 2 เส้นควบคุมว่าให้หมุนซ้ายหรือขวา

ในกรณี การควบคุมโดยตรงหรือแบบเปิด (open loop) นั้นเท่านั้นก็เพียงพอแล้ว แต่การควบคุมแบบปิด (close loop) จะต้องมี transducer หรือ sensor มาควบคุมระยะ โดยในที่นี้จะคิดไว้ที่กระบอกสูบของแขนกลและที่มอเตอร์ โดยที่กระบอกสูบจะเป็น Potentiometer ทำหน้าที่เปลี่ยนจากค่าระยะยัดหดมาเป็นแรงดันไฟฟ้า แล้วจึงเปลี่ยนค่าแรงดันไฟฟ้ามาเป็นค่าข้อมูลทางดิจิทัลด้วยวงจร A/D ส่งกลับมาเข้า CP-JR31 เพื่อส่งต่อไปยังคอมพิวเตอร์นำค่าที่ได้มาคำนวณเพื่อให้ได้ตำแหน่งที่ถูกต้องต่อไป ด้วยข้อมูล 8 บิตจากวงจร A/D สามารถอ้างอิงตำแหน่งได้ 256 ตำแหน่ง โดยระยะที่สามารถควบคุมได้คือ 7/256 ส่วนค่าที่มอเตอร์จะใช้ Sensor แบบออปติคัลเอนโค้ดเดอร์แบบไม่สมบูรณ์ในตัวเองซึ่งแสดงไว้ในบทของ Sensor แล้ว ซึ่งสำหรับตัวมอเตอร์ที่ใช้ในโครงการนี้จะสร้างสัญญาณออกมากประมาณ 680 pulse เมื่อหมุนไป 1 รอบ

2.3.3 มอเตอร์กระแสตรง

หลักการของมอเตอร์กระแสตรง

มอเตอร์คือ อุปกรณ์ซึ่งทำหน้าที่เปลี่ยนพลังงานกล และมอเตอร์ซึ่งขับเคลื่อนด้วยกระแสไฟฟ้าชนิดนี้ เรียกว่า มอเตอร์ดีซี และที่ขับเคลื่อนด้วยกระแสไฟเอซี เรียกว่า มอเตอร์เอซี นอกจากนี้ยังมีมอเตอร์ขนาดเล็กซึ่งอาจขับเคลื่อนได้ทั้งกระแสไฟดีซีหรือเอซี ทิศทางการหมุนของมอเตอร์ดีซีจะตรงข้ามกับกรณีเครื่องกำเนิดไฟฟ้าดีซีโดยสิ้นเชิง แต่สำหรับโครงสร้างแล้วจะเหมือนกันทุกประการจึงสามารถนำเครื่องกำเนิดไฟฟ้าดีซีมาใช้ทำหน้าที่ของมอเตอร์ดีซีได้

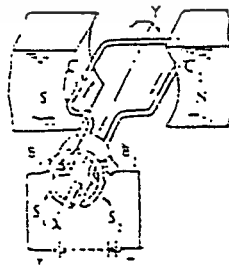
เมื่อป้อนกระแสให้กับมอเตอร์ดีซี ทิศทางการหมุนของมอเตอร์จะเป็นไปตามกฎมือซ้าย ดังรูปที่ 2.4



รูปที่ 2.4 กฎมือซ้าย

การเปลี่ยนกลับไปมาได้ระหว่างเครื่องกำเนิดไฟฟ้าและมอเตอร์

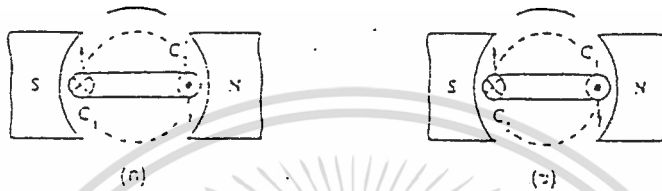
ดังที่ได้กล่าวไว้แล้วว่าเครื่องกำเนิดไฟฟ้ากระแสตรง สามารถทำหน้าที่ของมอเตอร์กระแสตรงได้นั้นสามารถอธิบายได้ดังนี้



รูปที่ 2.5 หลักการของมอเตอร์ดีซี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.5 เมื่อทำการปลดโหลดที่ขั้วของเครื่องกำเนิดไฟฟ้า ออกแล้วต่อแหล่งจ่ายไฟดีซีเข้าแทนที่ โดยให้ขั้วบวกของแหล่งจ่ายต่อกับแปรงถ่าน (+) B_1 และขั้วลบต่อกับแปรงถ่าน (-) B_2 ขณะนี้เครื่องกำเนิดไฟฟ้าจะทำหน้าที่เป็นมอเตอร์ กระแสจะไหลจากแหล่งจ่ายผ่านแปรงถ่าน (+) B_1 เข้าขดอาร์มาเจอร์และมาออกที่แปรงถ่าน (-) B_2 เข้าแหล่งจ่ายเป็นการครบวงจร และมอเตอร์จะหมุนไปในทิศทางซึ่งเป็นไปตามกฎมือซ้าย โดยพิจารณาจากรูปที่ 2.9



รูปที่ 2.6 หลักการของมอเตอร์กระแสตรง

จากรูป 2.6 (ก) เมื่อใช้กฎมือซ้ายจะได้ทิศทางการหมุนของมอเตอร์ในทิศตามเข็มนาฬิกา เมื่อขดลวดหมุนจากตำแหน่งเดิมไปเป็นมุม 90 องศา ลวดตัวนำ C_1 จะอยู่ด้านบนสุดโดยที่ซี่คอมมิวเตเตอร์ S_2 ของลวดตัวนำ C_2 จะเริ่มไม่แตะกับแปรงถ่าน B_2 และจากนั้นจะเริ่มแตะกับแปรงถ่าน B_1 ทำให้กระแสเริ่มไหลกลับทิศโดยไหลจากแปรงถ่าน B_1 ผ่านซี่คอมมิวเตเตอร์ S_2 เข้าลวดตัวนำ C_2 และ C_1 มาออกที่ซี่คอมมิวเตเตอร์ S_1 และแปรงถ่าน B_2 กระแสในลวดตัวนำ C_1 จะมีทิศทางเช่นนี้ไปเรื่อย ๆ ดังรูป 2.6 (ข) จนกระทั่ง C_1 อยู่ในตำแหน่งที่ต่ำสุด เมื่อเลขตำแหน่งนี้ไปกระแสในลวดตัวนำ C_1 จะไหลกลับทิศอีกโดยมีทิศทางดังรูป 2.6 (ก) และจะคงทิศทางนี้ไปเรื่อย ๆ จนถึงตำแหน่งที่ C_1 อยู่สูงสุด จากรูป 2.6 (ก) และ 2.6 (ข) จะเห็นว่าลวดตัวนำซึ่งอยู่ได้ขั้ว N จะมีกระแสไหลในทิศพุ่งออก ขณะที่ลวดตัวนำซึ่งอยู่ได้ขั้ว S มีทิศพุ่งเข้า นั่นคือขณะที่ลวดตัวนำหมุนเคลื่อนที่ไป แม้จะอยู่ที่ตำแหน่งใดก็ตามเนื่องจากทิศทางสัมพัทธ์ระหว่างกระแสและเส้นแรงแม่เหล็กไม่เปลี่ยนแปลง แรงที่กระทำต่อขดลวดจะมีทิศทางที่เสริมไปในทิศทางเดียวกันเสมอ

จากที่กล่าวมา แม้ในกรณีที่มีลวดตัวนำมากกว่านี้ ลวดตัวนำเส้นใดก็ตามซึ่งอยู่ได้ขั้วแม่เหล็กเดียวกันจะมีกระแสไหลในทิศทางเดียวกันทั้งหมด จึงทำให้แรงที่กระทำบนลวดตัวนำแต่ละเส้นมีทิศทางที่เสริมไปในแนวเดียวกันให้มีเตอร์หมุนไปได้ ในกรณีที่มีขดลวดจำนวนมากพอสมควร ผลรวมของแรงที่กระทำบนขดลวดแต่ละขดที่เวลาใด ๆ ให้ถือว่ามีความคงที่ และสำหรับมอเตอร์ในทางปฏิบัติจะเป็นไปตามนี้เช่นกัน จึงสรุปได้ว่าโครงสร้างของมอเตอร์ดีซีจะเหมือนกับของเครื่องกำเนิดไฟฟ้ากระแสตรงทุกประการ ตัวอย่างอื่นในทำนองเดียวกันนี้ เช่น พัดลมเมื่อหมุนใบพัดจะให้กระแสลม ตรงกันข้ามถ้าให้ลมปะทะตัวใบพัด พัดลมก็จะหมุน ในกรณีที่สามารถเปลี่ยนรูปที่ 2.6 (ก) ให้กลับเป็นรูป (ข) และรูป (ข) ให้กลับเป็นรูป (ก) ได้ใน

เมื่อเปรียบเทียบมอเตอร์กับเครื่องกำเนิดไฟฟ้า จะเห็นว่าลวดตัวนำของขดลวดซึ่งหมุนเคลื่อนไปในทิศทางเดียวกันและอยู่ภายใต้ขั้วแม่เหล็กเดียวกัน จะมีทิศทางกระแสที่กลับกัน ตัวอย่างเช่น เมื่อต่อโหลดที่ขั้วของเครื่องกำเนิดไฟฟ้าจะมีกระแสไหลผ่านลวดตัวนำของขดลวด กระแสที่เกิดขึ้นพร้อมกับเส้นแรงแม่เหล็กที่มีอยู่จะทำให้เกิดแรงกระทำบนลวดตัวนำแต่ละเส้นของขดลวดในทิศทางที่ต้านกับทิศทางที่ขดลวดกำลังหมุนอยู่ เมื่อกระแสโหลดหรือกระแสกำเนิดไฟฟ้าอยู่นั้นต้องสูญเสียแรงขับเคลื่อนไปส่วนหนึ่ง ซึ่งถูกนำไปใช้หักล้างกับแรงต้านดังกล่าวเนื่องจากเครื่องกำเนิดไฟฟ้าสามารถทำหน้าที่มอเตอร์ได้ ดังนั้นไม่ว่าจะเป็นโครงสร้างและชื่อเรียกของแต่ละชิ้นส่วน ตลอดจนวิธีการพันขดลวดของเครื่องกำเนิดไฟฟ้าดีซีสามารถนำมาใช้ได้กับกรณีของมอเตอร์

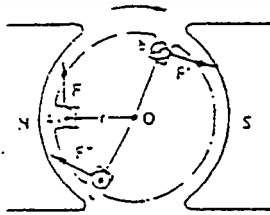
ทอร์ก

รูปที่ 2.7 แสดงแรงที่กระทำบนลวดตัวนำ (a) ซึ่งห่างจากจุดศูนย์กลาง (o) เป็นระยะ r (หน่วยเป็นเมตร) ในทิศทางสัมผัสกับเส้นรอบวงของแกนเหล็กอาร์มาเจอร์ ที่ตำแหน่งนี้กำหนดให้มีค่า F นิวตัน ดังนั้นแรงที่กระทำบนลวดตัวนำนี้จะมีโมเมนต์เท่ากับ $F \cdot r$ ซึ่งจะพยายามขับเคลื่อนแกนเหล็กอาร์มาเจอร์ให้หมุนเคลื่อนที่ไป แต่เนื่องจากในแกนเหล็กอาร์มาเจอร์มีสล๊อตเป็นจำนวนมาก และแต่ละสล๊อตก็มีลวดตัวนำจำนวนมากเช่นกัน ดังนั้นแรงที่กระทำบนลวดตัวนำทั้งหมด จะมีโมเมนต์ซึ่งพยายามขับเคลื่อนแกนเหล็กให้หมุนไปในทิศทางเดียวกัน เมื่อรวมโมเมนต์ย่อยทั้งหมดนี้เข้าด้วยกัน จะได้ผลรวมของโมเมนต์ ดังนี้

$$T_a = (F_1 + F_2 + F_3 + \dots)$$

โดยที่ T_a คือ ทอร์กที่เกิดขึ้นมีหน่วยเป็น mN

ดังนั้นทอร์กคือ ผลรวมของโมเมนต์ของแรงที่กระทำต่อลวดตัวนำรอบจุดศูนย์กลาง



รูปที่ 2.7 โมเมนต์ซึ่งลวดตัวนำอาร์มาเจอร์กระทำรอบจุดศูนย์กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถในการปรับกระแสได้เองตามขนาดของโหลด

เมื่อป้อนแรงดัน V คงที่ให้กับมอเตอร์แบบซิริย และให้ทำงานที่โหลดค่า ๆ มอเตอร์จะหมุนด้วยความเร็วรอบ n ค่าหนึ่ง และจะดึงกระแสจากแหล่งจ่ายไฟดีซีด้วยค่า I_a ค่าหนึ่ง เนื่องจากกำลังงานที่ด้าน Output ของมอเตอร์ในขณะนี้มีน้อย V_L จึงมีค่าน้อยด้วย นั่นคือกระแส I_a ที่แหล่งจ่ายไฟดีซีจ่ายออกจะต้องมีค่าต่ำด้วย เมื่อให้มอเตอร์ทำงานที่โหลดมาก ๆ V_L จะต้องมีค่ามากขึ้น สำหรับกรณีที่แรงดันของแหล่งจ่ายไฟมีค่าคงที่ I_a จำเป็นต้องมีค่ามากขึ้นด้วย นั่นคือเมื่อโหลดน้อยมอเตอร์จะดึงกระแสจากแหล่งจ่ายไฟดีซีน้อย และเมื่อโหลดมากจะดึงกระแสมากขึ้นโดยอัตโนมัติ

ความสามารถในการเร่งหรือหน่วงความเร็วรอบ

ลองพิจารณาความสัมพันธ์ระหว่างโหลดทอร์กซึ่งเป็นทอร์กที่โหลดต้องการกับมอเตอร์ทอร์กซึ่งเป็นทอร์กที่มอเตอร์จ่ายให้กับโหลด เพื่อให้เข้าใจได้ง่ายขึ้นจะยกตัวอย่างมอเตอร์ดีซีแบบขั้วในกรณีการอธิบายการหมุนของมอเตอร์ในขณะที่มีโหลดทางกลนั้น จะเท่ากับการใส่แรงเบรกกระทำต่อแกนของอาร์มาเจอร์เมื่อโหลดทอร์กสูงขึ้น ก็เท่ากับแรงเบรกที่กระทำต่อแกนอาร์มาเจอร์มีค่ามากขึ้น นั่นคือขณะที่มอเตอร์กำลังหมุนภายใต้โหลดค่าหนึ่งด้วยเสถียรภาพที่ดี ถ้าให้โหลดทอร์กมีค่าสูงขึ้น หรือแรงเบรกที่กระทำมีค่ามากขึ้นจะทำให้เกิดความหน่วงขึ้น ความเร็วรอบ n จะมีค่าไม่เปลี่ยนแปลง (แต่ในทางปฏิบัติเนื่องจากอาร์มาเจอร์รีแอคชันทำให้ ϕ มีค่าเปลี่ยนแปลงไปบ้าง) ดังนั้นถ้าความเร็วรอบน้อยลงแรงเคลื่อนไฟฟ้ากลับ (E_b) จะมีค่าต่ำลง แต่เนื่องจาก $I_a = (V - E_b) / r_a$ และทอร์ก $T = K\phi I_a$ มอเตอร์ทอร์กจะมีค่าเพิ่มขึ้น ดังนั้นความเร็วรอบจะลดลงเรื่อย ๆ จนกระทั่งทำให้ทอร์ก T เท่ากับ โหลดพอดี จากนั้นมอเตอร์จะเริ่มทำงานที่ตำแหน่งใหม่นี้

ในทางตรงกันข้ามถ้าโหลดทอร์กมีค่าน้อยกว่ามอเตอร์ทอร์ก แรงที่หมุนแกนอาร์มาเจอร์จะมากกว่าแรงเบรก ทำให้เกิดความเร่งขึ้นและความเร็วรอบจะเพิ่มขึ้น เมื่อความเร็วรอบเพิ่มขึ้นแรงเคลื่อนไฟฟ้ากลับจะมีค่ามากขึ้น กระแส $I_a = (V - E_b) / r_a$ จะมีค่าน้อยลง ทอร์ก $T = K\phi I_a$ จะมีค่าลดลง ความเร่งจะน้อยลงจนเป็น 0 เมื่อทอร์กจากมอเตอร์มีค่าเท่ากับโหลดทอร์กพอดี จากนั้นมอเตอร์จะเริ่มทำงานที่ตำแหน่งใหม่นี้ ในการเริ่มเดินมอเตอร์จากสภาพที่หยุดนิ่ง จนกระทั่งหมุนนั้น แรงเคลื่อนไฟฟ้ากลับ E_b จะค่อย ๆ เพิ่มขึ้นเรื่อย ๆ จาก $E_b = 0$ ในช่วงแรกสุดนั้นเนื่องจากมีกระแสสูงมาก ความเร็วรอบจะสูงขึ้นเรื่อย ๆ จนกระทั่งโหลดทอร์กเท่ากับมอเตอร์ทอร์กและความเร็วรอบจะคงที่ โหลดทอร์กจะเปลี่ยนแปลงตามความเร็วรอบ หรือมีค่าคงที่เมื่อความเร็วรอบเปลี่ยนไป หรือไม่นั้นย่อมขึ้นกับคุณสมบัติของโหลด ตัวอย่างเช่น พัดลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่ขึ้นต้นการคัด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นยิ่งหมุนที่ความเร็วรอบสูงขึ้น โหลดทอร์กหรือทอร์กที่โหลดต้องการจะยังมีค่ามากขึ้นในขณะที่เครื่องกลึงไม่ว่าความเร็วรอบจะเปลี่ยนไปอย่างไรก็ตาม โหลดจะต้องการทอร์กเกือบคงที่ขณะที่เครื่องกลึงไม่ว่าความเร็วรอบจะเปลี่ยนไปอย่างไรก็ตาม โหลดจะต้องการทอร์กเกือบคงที่โดย

อด

การเปลี่ยนทิศทางการหมุน

การทำให้เครื่องจักรกลเดินหน้าหรือถอยหลังได้นั้น จำเป็นที่จะต้องสลับเปลี่ยนทิศทางการหมุนของมอเตอร์ สำหรับเครื่องจักรกลในโรงงาน สามารถควบคุมการเดินหน้าถอยหลังได้ โดยให้มอเตอร์มีทิศทางการหมุนเดียวกันตลอด แต่ต้องอาศัยเกียร์ช่วยในการเปลี่ยนทิศทางการหมุน ซึ่งวิธีนี้เป็นวิธีที่ใช้กันโดยทั่วไป แต่ในบางโอกาสการเปลี่ยนทิศทางการหมุนของเครื่องจักรกลอาจใช้วิธีการเปลี่ยนทิศทางการหมุนของมอเตอร์โดยตรง

ทิศทางการหมุนของมอเตอร์สามารถหาได้โดยอาศัยกฎมือซ้าย ดังนั้นไม่ว่าจะเปลี่ยนทิศทางเส้นแรงแม่เหล็กใหญ่ (นั่นคือทิศทางของกระแสสนาม) หรือทิศทางการไหลของกระแสอาร์มาเจอร์ก็ตามอย่างใดอย่างหนึ่งแล้ว ก็สามารถกลับทิศทางการหมุนของมอเตอร์ได้ สำหรับมอเตอร์ดีซีแบบซีรีย์หรือซันด์ ถ้ากลับขั้วที่ต่อกับแหล่งจ่ายไฟดีซีแล้ว เนื่องจากทั้งทิศทางของกระแสอาร์มาเจอร์และทิศทางของกระแสสนามจะกลับพร้อมกัน ทิศทางการหมุนจะเหมือนเดิมโดยไม่กลับทิศ ในกรณีที่มีอินเตอร์โพลติดตั้งอยู่นั้น ถ้าต้องการกลับทิศกระแสอาร์มาเจอร์ ก็จำเป็นที่จะต้องกลับทิศกระแสของอินเตอร์โพลไปพร้อมกัน นั่นคือในการกลับทิศนั้นให้ถือว่าขดอินเตอร์โพลและขดอาร์มาเจอร์เป็นขดชุดเดียวกัน จะแยกกลับเพียงขดหนึ่งไม่ได้โดยเด็ดขาด

2.3.4 โซลินอยด์

รากศัพท์ของคำว่าโซลินอยด์น่าจะมาจากคำว่า “โซเลน(solen)” ซึ่งมีความหมายทางแพทย์คล้ายกับ เฟือกหุ้มอวัยวะที่ได้รับบาดเจ็บ เมื่อมีประจุกรรมตัวนี้เกิดขึ้น โครงสร้างของมันก็คือ ขดลวดพันรอบๆแกนสารแม่เหล็ก ดังรูปที่ 2.8 นั่นเอง ลักษณะก็คล้ายๆกับทรงกระบอก

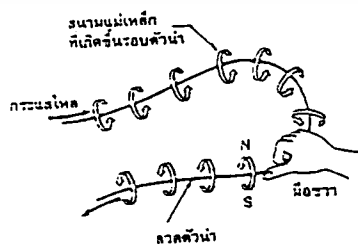
เราใช้โซลินอยด์มาประยุกต์ใช้กับงานที่ต้องการเชื่อมโยงพลังงานไฟฟ้ามาเป็นพลังงานกลโดยตรง โดยสัญญาณไฟฟ้าที่ป้อนเข้ามาทางขดลวด จะทำให้แกนสารแม่เหล็กของโซลินอยด์เกิดการเคลื่อนที่ การเคลื่อนที่นี้เองที่เราสามารถนำไปใช้ประโยชน์ เช่น ชักกลอนประตูเอาไว้ไปลิบ กระเบื้องทำให้กลไกทำงานหรือหยุดทำงาน ฯลฯ โซลินอยด์ที่ใช้กันมีทั้งใช้กับไฟฟ้ากระแสสลับและกระแสตรง



รูปที่ 2.8 โครงสร้างพื้นฐานของโซลินอยด์

หลักการการทำงานของโซลินอยด์

เมื่อมีกระแสไฟฟ้าไหลในลวดตัวนำใดก็ตาม จะเกิดสนามแม่เหล็กขึ้นรอบๆตัวนำนั้น โดยทิศทางของเส้นแรงแม่เหล็กให้ใช้กฎมือขวา(ให้เอามือขวากำรอบเส้นลวด โดยให้นิ้วหัวแม่มือแทนทิศของกระแสไหล นิ้วที่เหลือทั้งหมดจะแสดงทิศทางเส้นแรงแม่เหล็กจากขั้วใต้ไปขั้วเหนือ)



รูปที่ 2.9 แสดงทิศทางของสนามแม่เหล็กที่เกิดขึ้นเมื่อมีกระแสไหลผ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

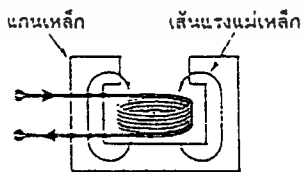


เมื่อเราเอาเส้นลวดลักษณะเดิมแต่ยาวน้อยมาขดเป็นวงๆหลายๆวง ก็จะเกิดลักษณะของขดลวดขึ้นดังรูปที่ 2.10 สนามแม่เหล็กที่เกิดจากขดลวดแต่ละขดจะอยู่ในทิศทางเสริมกัน และก่อให้เกิดเป็นเส้นแรงของสนามแม่เหล็กรวม มีทิศเหนือ-ใต้ ดังรูป ซึ่งก็แสดงว่าในขณะที่ขดลวดนี้ได้ทำหน้าที่เป็นเหมือนแม่เหล็กถาวรแท่งหนึ่ง ซึ่งพร้อมที่จะดูดสารแม่เหล็กทันที แต่เนื่องจากสภาพรอบๆขดลวดเป็นอากาศเส้นแรงแม่เหล็กจึงไม่เข้มข้นมากนัก

เพื่อที่จะไม่ให้สนามแม่เหล็กที่เกิดขึ้นกระจัดกระจายจึงต้องใส่แกนเหล็กอ่อนรูปตัว C เข้ามารอบๆขดลวดดังรูปที่ 2.11 และถ้าเอาแกนกระทู้่ง(plunger) มาใส่เข้าตรงกลางขดลวดในตำแหน่งที่ 1 แกนกระทู้่งจะถูกดูดให้ลึกลงเข้ามาจนสนิทในตำแหน่งที่ 2 ยิ่งระยะทางไกลสักเท่าไร แรงดึงดูดก็จะยิ่งมากขึ้นเท่านั้น

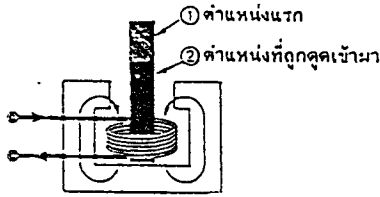


รูปที่ 2.10 แสดงทิศทางของสนามแม่เหล็กที่เกิดขึ้นในขดลวดที่มีกระแสไหล



รูปที่ 2.11 แสดงการเพิ่มเหล็กอ่อนเข้ามาเพื่อเพิ่มความเข้มของสนามแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำ 037118 งานการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 แสดงการเคลื่อนที่ของแกนกระทุ้ง

มีข้อแตกต่างระหว่างโซลินอยด์กระแสตรงและกระแสสลับ คือ ในโซลินอยด์กระแสตรง กระแสที่ไหลในขดลวดจะค่อนข้างคงที่ไม่เปลี่ยนแปลง ไม่ว่าแกนกระทุ้งจะอยู่ในตำแหน่งใดก็ตาม แต่ในโซลินอยด์ไฟฟ้ากระแสสลับ ในขณะที่แกนกระทุ้งอยู่นอกขดลวดจะมีค่าสูงและเมื่อแกนกระทุ้งถูกดูดเข้ามาจนสุดขดลวดกระแสจะลดต่ำลง ลักษณะนี้เองทำให้เราต้องระวังอย่าให้เกิดการติดขัดของแกนกระทุ้งในโซลินอยด์ไฟสลับ เพราะจะทำให้เกิดกระแสหลายๆไหลค้างอยู่ ทำให้ขดลวดร้อนขึ้นและอาจจะไหม้เสียหายได้

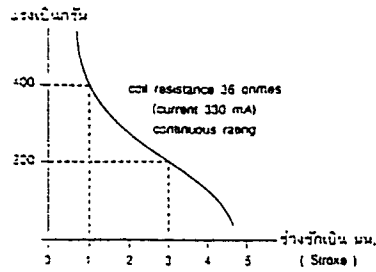
ในโครงสร้างของโซลินอยด์แบบไฟสลับนั้นจะต้องพันขดลวด shade coil หรือแหวน (ring) ซึ่งเป็นขดลวดพันรอบแกนเหล็กเพียงรอบเดียว หรือไม่กี่รอบลัดวงจรเอาไว้เลย จุดประสงค์ที่พันไว้เพราะในไฟกระแสสลับ กระแสจะลดลงมาเป็นศูนย์สองครั้งทุกๆ ไซเคิลช่วงที่กระแสเป็นศูนย์นี้เองทำให้แรงดูดแม่เหล็กลดลงและทำให้เกิดเสียงหึ่งๆขึ้นและการดูดก็ไม่แน่นอน ขดลวดที่เดิมเข้าไปจะทำให้วงจรแม่เหล็กเกิดสภาพเป็น 2 เฟส คือแม้ในขณะที่กระแสเป็น 0 ก็ตามขดลวดแหวนซึ่งมีกระแสที่เกิดจากการเหนี่ยวนำกับสนามแม่เหล็ก จะยังคงมีแรงแม่เหล็กมาเสริมการดูดช่วงนี้ได้แต่ก็จะทำให้เกิดการสูญเสียความร้อนในขดลวดบ้างเล็กน้อย

ขั้นตอนในการเลือกใช้โซลินอยด์

คำนึงถึงหลักใหญ่ๆดังนี้

1. แรงดันใช้งาน ไม่ว่าจะเป็นแบบไฟตรงหรือไฟสลับ ถ้าเป็นไฟสลับให้ดูที่ความถี่ที่ใช้งานด้วย
2. ช่วงชักในการใช้งานของโซลินอยด์จะต้องเคลื่อนที่เป็นระยะทางเท่าไร (เป็นมิลลิเมตร)
3. ขนาดของโหลดว่าจะต้องใช้แรงขนาดเท่าไร (กรัม)
4. ใช้งานต่อเนื่องหรือไม่ หมายถึงเราต้องใส่แรงดันไฟเข้าขดลวดค้างไว้เลยโดยที่ขดลวดไม่ไหม้

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงความสัมพันธ์ระหว่างแรงกับระยะช่วงชักของโซลินอยด์ไฟตรง 12 V. ยี่ห้อโคอิเกะ รุ่น SB-102

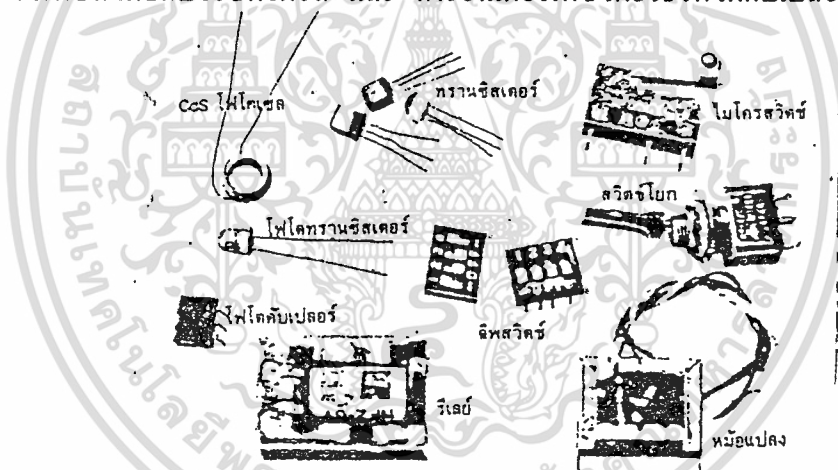
ในรูปที่ 2.13 เป็นตัวอย่างกราฟแสดงความสัมพันธ์ของแรงกับระยะช่วงชักของโซลินอยด์ จะเห็นว่าช่วงชักไกลๆจะมีแรงน้อยมาก และระยะใกล้เข้ามาก็จะมีแรงมากขึ้นเป็นทวีคูณ ในกรณีนี้ โซลินอยด์จะให้แรงดูด 200 กรัมที่ช่วงชัก 3 มม. และจะให้ถึง 400 กรัมที่ช่วงชัก 1 มม.

2.3.5 การอินเตอร์เฟซ

คำว่า อินเตอร์เฟซ (interface) หมายถึง การต่อเชื่อมระหว่างสองสิ่งที่มีคุณสมบัติแตกต่างกัน เราใช้คำนี้ในวงจรดิจิทัลเมื่อเราหมายถึงวงจรหรือวิธีการในการต่อเชื่อมวงจรสองชนิดที่มีคุณสมบัติแตกต่างกันเข้าด้วยกัน การอินเตอร์เฟซยังมีความหมายครอบคลุมไปถึงการอินเตอร์เฟซระหว่างคนกับเครื่อง การอินเตอร์เฟซระหว่างคนกับเครื่องจักรนี้จำเป็นต้องมีวิธีการและอุปกรณ์ที่เหมาะสม เพื่อให้ทั้งคนและเครื่องจักรสามารถส่งข้อมูลถึงกันและทำงานประสานงานกันได้

ยกตัวอย่างเช่น การต่อไอซี TTL เข้ากับไอซี CMOS การต่อเอาต์พุตของไอซีไปควบคุมมอเตอร์ การอ่านสัญญาณอะนาลอก เช่น สัญญาณขนาดความถี่ของเครื่องขยายเสียงเป็นสัญญาณดิจิทัลเพื่อป้อนเข้าวงจรดิจิทัล เป็นต้น

ในบทนี้ จะกล่าวถึงการอินเตอร์เฟซระหว่างอุปกรณ์ในวงจรที่ใช้ในโครงการนี้ คือ การอินเตอร์เฟซสวิทช์กับไอซีดิจิทัล และ การอินเตอร์เฟซโดยใช้ไฟโตคัปเปลอร์



รูปที่ 2.14 ตัวอย่างอุปกรณ์การอินเตอร์เฟซ

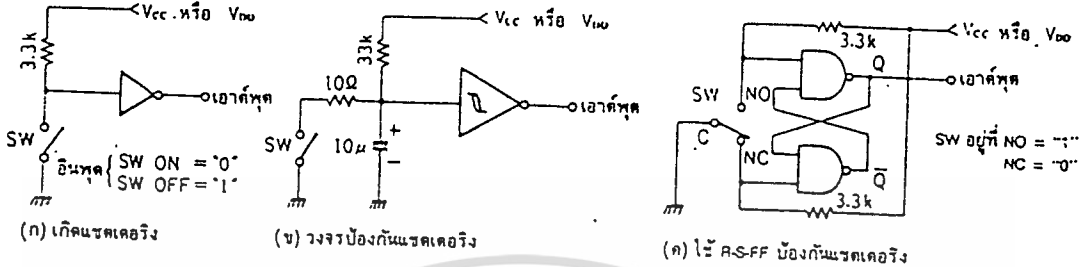
การอินเตอร์เฟซสวิทช์กับไอซีดิจิทัล

เมื่อเราต่อสวิทช์เข้ากับวงจรดิจิทัลเพื่ออาศัยการเปิดปิดสวิทช์เป็นสัญญาณให้กับวงจร เราจะต่อหน้าสัมผัสกับอินพุตของเกต โดยมีตัวต้านทานดึงขึ้น ดังแสดงในรูปที่ 2.15 (ก) เมื่อสวิทช์เปิดจะได้สัญญาณ "1" เป็นอินพุตของเกต ถ้าสวิทช์ปิดจะเกิดลัทธิวงจรลกราวนด์ได้ เป็นสัญญาณ "0" ป้อนเข้าเกตเป็นการเปลี่ยนการ "เปิด" และ "ปิด" เป็นสัญญาณ "1" กับ "0" ได้ง่าย ๆ

แต่ในบางครั้งการต่อสวิทช์เข้ากับ ไอซีแบบวิธีง่าย ๆ แบบนี้จะไม่ไ้ผลและจะเกิดปัญหาขึ้นมากมาย เพราะในระหว่างที่เรากดหรือ โยกสวิทช์นั้นหน้าสัมผัสของสวิทช์จะไม่แตะหรือแยกจากกันอย่างเด็ดขาด จะมีการเค้งและสั้นของหน้าสัมผัสก่อนการแตะหรือการแยกจากกันเสมอ

ไม่ว่าการณ์ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราเรียกปรากฏการณ์นี้ว่า แชตเตอริง (chattering) การสั่นของหน้าสัมผัสนี้ ไอซีดิจิตอลสามารถรับรู้ได้เพราะ ไอซีทำงานเร็วกว่าตาของคนเรามาก ดังนั้นบางครั้งเราคปุ่มเพียงครั้งเดียว แต่เหมือนมีสัญญาณพัลส์เข้าไปในวงจรมากมาย



รูปที่ 2.15 การอินเตอร์เฟซสวิทช์กับไอซี

แต่ในบางครั้งการต่อสวิทช์เข้ากับ ไอซีแบบวิธีง่าย ๆ แบบนี้จะไม่ไ้ผลและจะเกิดปัญหาขึ้นมากมาย เพราะในระหว่างที่เรากดหรือโยกสวิทช์นั้นหน้าสัมผัสของสวิทช์จะไม่แตะหรือแยกจากกันอย่างเด็ดขาด จะมีการเค็งและสั่นของหน้าสัมผัสก่อนการแตะหรือการแยกจากกันเสมอ เราเรียกปรากฏการณ์นี้ว่า แชตเตอริง (chattering) การสั่นของหน้าสัมผัสนี้ ไอซีดิจิตอลสามารถรับรู้ได้เพราะ ไอซีทำงานเร็วกว่าตาของคนเรามาก ดังนั้นบางครั้งเราคปุ่มเพียงครั้งเดียว แต่เหมือนมีสัญญาณพัลส์เข้าไปในวงจรมากมาย

การแก้ปัญหาแชตเตอริงของหน้าสัมผัสนั้นใช้วิธีการตามในรูปที่ 2.15 (ข) และรูปที่ 2.15 (ค) ได้ ในรูปที่ 2.15 (ข) เป็นการต่อตัวเก็บประจุคร่อมหน้าสัมผัส(อาจมีตัวต้านทานเล็ก ๆ อนุกรมกับหน้าสัมผัสไว้ก็ได้)เพื่อให้แรงดันคร่อมหน้าสัมผัสเปลี่ยนแปลงทันทีไม่ได้ เมื่อเปิดสวิทช์ตัวเก็บประจุจะชาร์จผ่าน 33 กิโลโอห์ม ทำให้แรงดันที่อินพุตของเกตค่อย ๆ เพิ่มขึ้น แม้จะมีการสั่นของหน้าสัมผัสก็ไม่มีผลต่อแรงดันมากนัก เมื่อปิดสวิทช์ตัวเก็บประจุจะคายประจุผ่าน 10 โอห์มไปที่สวิทช์ทันที แม้จะมีการสั่นของหน้าสัมผัสอีกก็ไม่มีผลทำให้แรงดันเปลี่ยนแปลงอีกเช่นเดียวกัน เราสามารถเลือกค่า R (ตัว 33 กิโลโอห์ม) และ C ให้เหมาะสมกับระยะเวลาการสั่นของหน้าสัมผัสได้ R หรือ C ใหญ่จะทำให้ระยะเวลาการแก้แชตเตอริงทำได้นานขึ้น

อย่างไรก็ตาม แรงดันที่อินพุตของเกตจะเปลี่ยนแปลงค่อนข้างช้าเพราะผลของตัวเก็บประจุ เราจึงนิยมใช้ไอซีที่เป็นชนิดทรานซิสเตอร์หรือรีเลย์ที่รับสัญญาณเข้าจากสวิทช์ ชนิดทรานซิสเตอร์จะให้เอาต์พุตที่เปลี่ยนจาก “1” เป็น “0” หรือ “0” เป็น “1” ที่ไวขึ้นมาก เมื่อส่งต่อให้ตัวอื่นจะได้ไม่มีปัญหา

การอินเตอร์เฟซสวิทช์ในรูปที่ 2.15 (ค) เป็นวงจรที่นิยมใช้กับสวิทช์ 3 ขั้ว คือมีขั้วร่วม เอกสาขาขั้ว NO และขั้ว NC ใ้รูปร่างคล้ายขั้วและใช้เกิดมากหน่อย แต่เป็นวิธีการอินเตอร์เฟซที่ไ้ผลไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดีที่สุด เกต NAND 2 ตัวในวงจรต่อเป็น RS-FF ขั้ว NO ต่อเข้ากับขั้วเซตและขั้ว NC ต่อเข้ากับขั้วรีเซตของฟลิปฟลอป

หลักการการทำงานที่วงจรสามารถแก้ปัญหาแชนเตอร์ริงมีง่าย ๆ คือ ไม่ว่า RS-FF จะถูกเซตกี่ครั้ง เอาต์พุตจะได้ “1” เหมือนกัน และไม่ว่าจะถูกรีเซตกี่ครั้ง เอาต์พุตก็เป็น “0” เหมือนเดิมเช่นกัน ดังนั้นเมื่อเรากดสวิตช์ หน้าสัมผัสแยกจากขั้ว NC มาขั้ว NO ตรงขั้ว NO จะมีการสั้นของหน้าสัมผัสก็ครั้ง เอาต์พุต Q จะเป็น “1” คงที่เสมอ เมื่อเราปล่อยมือจากการกดสวิตช์ หน้าสัมผัสจะดึงกลับมาแตะขั้ว NC เช่นเดิม ก่อนแตะแม้จะมีการสั้นของหน้าสัมผัสบ้าง แต่เอาต์พุต Q จะเป็น “0” ทันทีและคงค่าไว้เช่นนั้น

การต่อเอาต์พุตของไอซีไปที่สวิตช์ก็มีปัญหาอยู่บ้างเหมือนกัน เช่น การต่อเอาต์พุตของไอซีหลาย ๆ ตัวไปที่สวิตช์เลือกตั้งในรูปที่ 2.16 ข้อควรระวัง คือ สวิตช์เลือกที่ใช้กันมีอยู่ 2 แบบ คือ แบบแยกก่อนแตะ (break before make) และ แบบแตะก่อนแยก (make before break)



รูปที่ 2.16 สวิตช์เลือกที่ใช้กับเอาต์พุตของไอซี

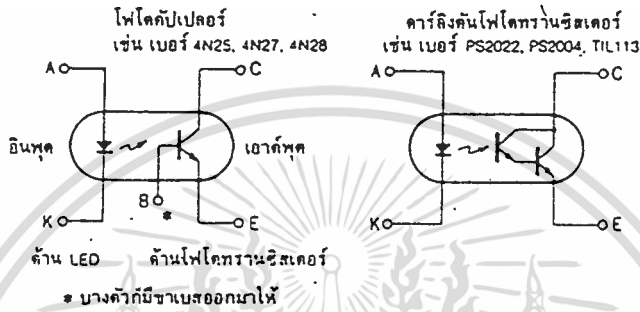
สวิตช์เลือกที่เป็นแบบแยกก่อนแตะจะมีใช้กันมาก เมื่อปิดสวิตช์หน้าสัมผัสเคลื่อนที่ที่จะเคลื่อนจากหน้าสัมผัสตรึงที่แตะอยู่ไปสู่หน้าสัมผัสตรึงขั้วถัดไป ในระหว่างเคลื่อนหน้าสัมผัสจะแยกออกก่อนแล้วค่อยไปแตะหน้าสัมผัสขั้วใหม่

สวิตช์เลือกอีกแบบก็มีใช้กันเหมือนกัน แบบนี้ในระหว่างการเคลื่อนจากหน้าสัมผัสตรึงขั้วหนึ่ง ไปอีกขั้วหนึ่งนั้น หน้าสัมผัสเคลื่อนที่ที่ยังแตะกับขั้วใดขั้วหนึ่งอยู่ตลอดเวลา ในบางขณะจะแตะกับหน้าสัมผัสตรึงทั้งสองขั้วก็มีจึงเรียกว่าแตะก่อนแยก แบบนี้ไม่ควรใช้กับเอาต์พุตของไอซีดิจิทัล เพราะในขณะที่ปิดสวิตช์เลือกจะมีช่วงเวลาหนึ่งที่เอาต์พุตของไอซีลัดวงจรกันเอง แม้จะเป็นช่วงเวลาสั้น ๆ แต่ถ้าปิดสวิตช์บ่อย ๆ จะทำให้ไอซีเสียหายได้

การอินเตอร์เฟซโดยใช้โฟโตคัปเปลอร์

โฟโตคัปเปลอร์เป็นอุปกรณ์ที่ใช้ในการอินเตอร์เฟซกันมาก ภายในโฟโตคัปเปลอร์จะมี LED กับ โฟโตทรานซิสเตอร์วางคู่กันอยู่ LED จะอยู่ทางด้านอินพุต ส่วนโฟโตทรานซิสเตอร์ ไม่ว่ารณใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

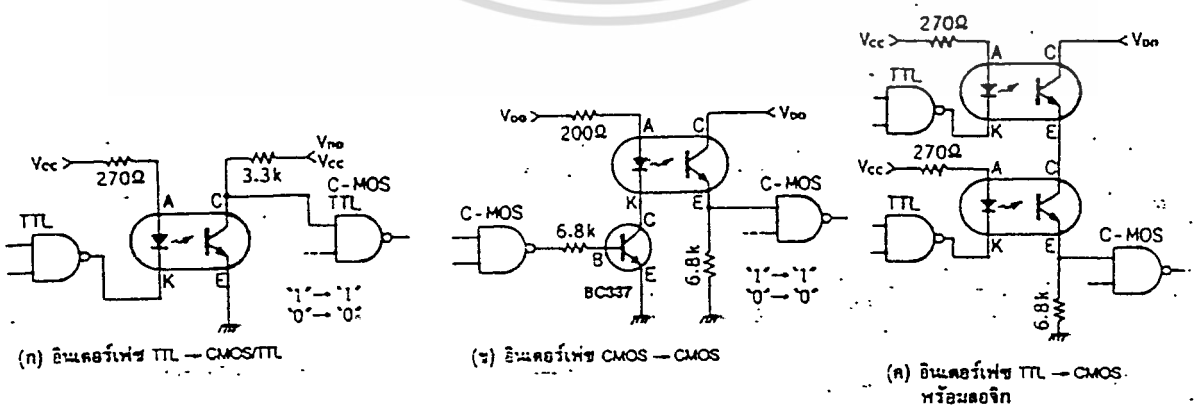
จะอยู่ทางด้านเอาต์พุต เมื่อเราจ่ายกระแสเข้าไปที่ LED LED จะเปล่งแสงออกมา แสงนี้เรามองไม่เห็นเพราะอยู่ภายในโฟโตทรานซิสเตอร์เมื่อได้รับแสงนี้จะอยู่ในสถานะอิ่มตัวหรือ ON โฟโตคัปเปิลเลอร์ที่มีขายกันในปัจจุบัน มีรูปร่างเหมือนไอซีทั่วไป อยู่ในตัวถังแบบคิพมีทั้งแบบพลาสติกและเซรามิก โฟโตทรานซิสเตอร์ภายในบางครั้งเป็นแบบคาร์ลิงตัน มีอัตราขยายสูงมาก อัตราขยายกระแสของโฟโตคัปเปิลเลอร์คิดจากอัตราส่วนของกระแสอินพุตที่ป้อนให้ LED กับกระแสขาออกที่ทรานซิสเตอร์จับได้ บางครั้งเรานำโฟโตคัปเปิลเลอร์มาใช้เป็นวงจรรขยายในวงจรอะนาล็อกก็ได้



รูปที่ 2.17 โครงสร้างของโฟโตคัปเปิลเลอร์

คุณสมบัติที่เด่นที่สุดของโฟโตคัปเปิลเลอร์คือ การที่อินพุตและเอาต์พุตแยกกันทางไฟฟ้าอย่างเด็ดขาดนั่นเอง คุณสมบัติอันนี้ทำให้วงจรที่ต่อเชื่อมกันด้วยโฟโตคัปเปิลเลอร์ไม่ต่อถึงกันทางไฟฟ้า ใช้แหล่งจ่ายไฟแยกกันและมีกราวด์แยกกันได้ ในวงจรที่เกี่ยวข้องกับความปลอดภัยและวงจรที่ต้องการกำจัดสัญญาณรบกวนมักจะใช้โฟโตคัปเปิลเลอร์นี้

รูปที่ 2.18 แสดงวิธีการใช้โฟโตคัปเปิลเลอร์เป็นวงจรรินเตอร์เฟชระหว่างสองวงจร อาจใช้เชื่อมต่อระหว่าง TTL กับ CMOS ก็ได้ หรือระหว่าง CMOS หรือ TTL กันเองก็ได้ สังเกตว่าการต่อวงจรทั้งแบบ (ก) และ (ข) ในรูปที่ 2.17 นั้น โฟโตคัปเปิลเลอร์ทำหน้าที่คล้ายบัฟเฟอร์คือสัญญาณ "1" และ "0" จะถูกส่งถ่ายกันโดยไม่มีกรกลับสัญญาณ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.18 ตัวอย่างการอินเตอร์เฟชโดยใช้โฟโตคัปเปิลเลอร์ โยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18(ค) เป็นการใช้งานที่พลิกแพลงมาก ใช้โฟโตคัปเปิลอร์สองตัว มาต่อเป็นเงื่อนไข AND กัน จะเห็นว่าถ้า LED ทั้งสองสว่างจึงจะทำให้มีกระแสไหลผ่านโฟโตทรานซิสเตอร์ทั้งสองตัวไปที่ตัวต้านทาน 6.8 กิโลโอห์ม ทำให้สัญญาณ “1” เป็นเอาต์พุต ถ้ากรณีอื่นจะได้สัญญาณ “0” เป็นเอาต์พุต เราอาจใช้โฟโตคัปเปิลอร์มาประกอบเป็นวงจร AND ,OR, NAND และ NOR ได้โดยใช้วิธีการเช่นนี้ ทำให้หุ่นเกิดเหล่านี้ลง



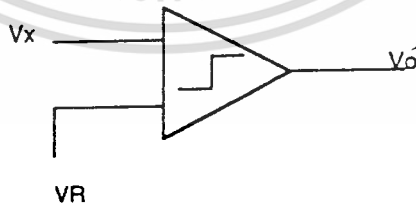
2.8.6 วงจรเปลี่ยนจากอนาลอกเป็นดิจิตอล(ANALOG TO DIGITAL CONVERTER ,ADC)
ลักษณะการจับวงจร ADC มีหลายแบบ แต่ที่นิยมใช้มีเพียงไม่กี่แบบและส่วนใหญ่จะอยู่ในรูปของวงจรรวม

Basic conversion method

วิธีการแปลงสัญญาณอนาลอกเป็นดิจิตอลแบบง่ายๆแสดงในรูปที่ 2.19 แรงดันอินพุตที่ไม่ทราบค่า V_x จะต่อเข้ากับขาอินพุตขาหนึ่งของขานาลอกคอมพารเตอ์ และแรงดันอ้างอิงที่ขนาดแปรตามเวลา V_r ต่อเข้ากับอินพุตอีกขาหนึ่งของคอมพารเตอ์ ลักษณะของทรานเฟอ์ฟังก์ชันของคอมพารเตอ์แสดงในรูปที่ 2.19 ถ้าแรงดันอินพุต V_1 มากกว่าแรงดันอินพุต V_2 แล้วแรงดันเอาต์พุตจะมีลอจิกเป็น 1 ถ้าแรงดันอินพุต V_1 น้อยกว่าแรงดันอินพุต V_2 แล้วแรงดันเอาต์พุตจะเป็น 0

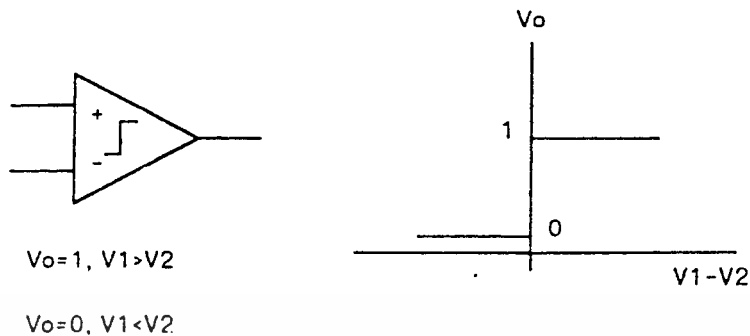
วิธีในการแปลงข้อมูลคือ แรงดันอ้างอิงจะถูกแปรค่าจนกระทั่งรู้ค่าแรงดันอินพุตที่ผิดพลาดไม่เกิน Quantization error ของคอนเวอร์เตอร์ ในแนวความคิดแล้วตรรกะของ ADC คือพยายามเลือกกลุ่มของ ส.ป.ส. ไบนารี A_i เพื่อให้ผลต่างระหว่างแรงดันอินพุต V_x และค่าที่ Quantize ได้ครั้งสุดท้าย น้อยกว่า 0.5 LSB ซึ่งเขียนเป็นสมการได้

$$V_x - V_{fsr} \sum_{i=1}^n A_i 2^i < 0.5 \text{LSB}$$



รูปที่ 2.19 แสดงวิธีการพื้นฐานของ ADC

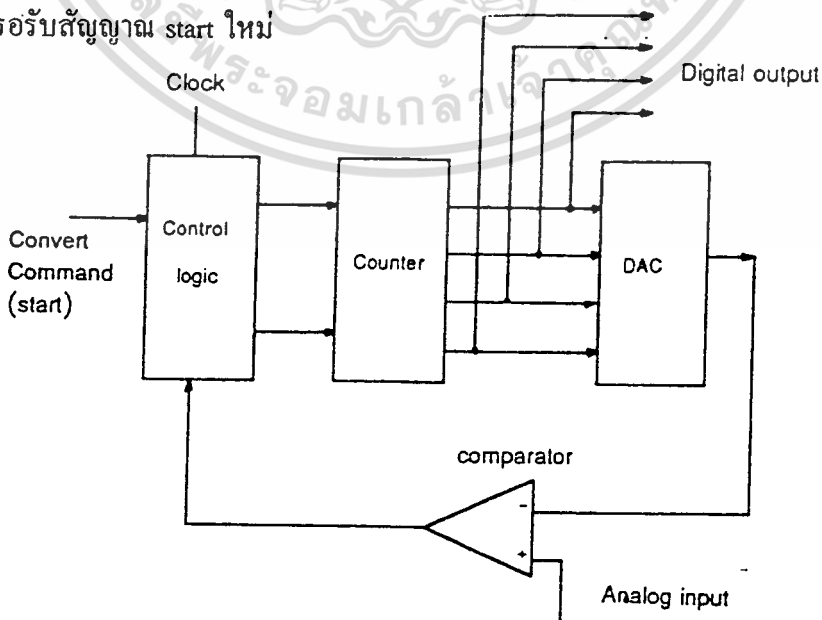
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



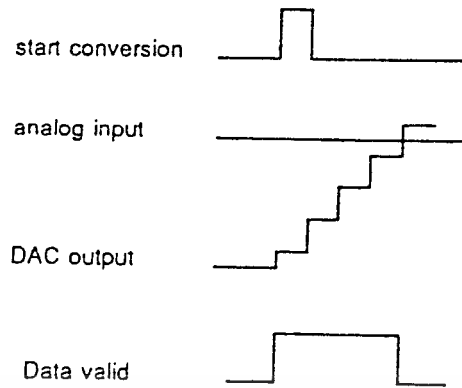
รูปที่ 2.20 แสดงทรานเฟอร์ฟังก์ชันของคอมพารเตอร์

Counter type ADC

การจับวงจร ADC ลักษณะนี้เป็นแบบที่ง่ายที่สุด หลักการทำงานของวงจรคือ การเปรียบเทียบขนาดของแรงดันที่เอาต์พุตของ DAC กับสัญญาณอนาล็อกที่ไม่ทราบค่า V_{in} การทำงานจะเริ่มโดยสัญญาณ start conversion ล็อกจิกคอนโทรลจะรีเซ็ตเคาท์เตอร์ให้เป็นศูนย์ แล้วเริ่มนับ ขึ้นจากศูนย์ เอาต์พุตของเคาท์เตอร์จะป้อนให้ DAC เพื่อแปลงเป็นสัญญาณอนาล็อกลักษณะเป็นขั้นบันได นำมาเปรียบเทียบกับสัญญาณอนาล็อกอินพุตที่คอมพารเตอร์ โดยเคาท์เตอร์จะนับจนกระทั่งเอาต์พุตเท่ากับสัญญาณอนาล็อกอินพุตหรือต่างกันไม่เกิน 1 LSB แล้วคอมพารเตอร์จะเปลี่ยนสถานะไปหยุดการนับของเคาท์เตอร์และ latch ค่าจากเคาท์เตอร์เพื่อรอการประมวลผลต่อไป และรอรับสัญญาณ start ใหม่



เอกสารนี้เป็นเอกสารที่สงวนรูปที่ 2.21 ก) บล็อกไดอะแกรมของ counter type DAC ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 ข) timing diagram counter type DAC

วงจรนี้มีข้อเสียที่ทำงานได้ช้าเพราะการ conversion แต่ละครั้งเคาท์เตอร์จะต้องถูกรีเซ็ตและเริ่มนับจากศูนย์ทุกครั้ง ดังนั้นในการ conversion เป็นดิจิทัล n bit จะต้องใช้จำนวน clock ถึง 2^n เพื่อเปลี่ยนให้ได้ค่าสูงสุดเต็มสเกล ส่วนข้อดีคือ สร้างได้ง่ายรวดเร็วราคาถูกแต่ความแม่นยำขึ้นอยู่กับ DAC ที่ใช้

Tracking ADC

Tracking ADC จะปรับปรุงวงจรแบบ counter type ทางด้านความเร็ว โดยใช้เคาท์เตอร์แบบนับขึ้นลงได้ไม่จำเป็นต้องเริ่มนับจากศูนย์ทุกครั้งแต่จะเริ่มนับจากค่าที่ได้ latch ไว้จากการเปลี่ยนสัญญาณครั้งล่าสุด ดังนั้นส่วนควบคุมทางดิจิทัลจึงซับซ้อนกว่ามาก โดยการทำงานจะเป็นดังนี้ เอาท์พุทจาก DAC จะถูกเปรียบเทียบกับสัญญาณอินพุท V_{in} หาก V_{in} มากกว่าลักษณะ logic ของคอมพารเตอ์จะควบคุมให้เคาท์เตอร์นับขึ้น ถ้า V_{in} น้อยกว่าลักษณะ logic ของคอมพารเตอ์ก็จะควบคุมให้เคาท์เตอร์นับลง จนกว่าค่าที่สุดของเคาท์เตอร์จะต่างจากสัญญาณอนาล็อกอินพุทไม่เกิน 1 LSB และค่าของเคาท์เตอร์จะถูก latch ไว้จากนั้นเคาท์เตอร์จะทำงานแบบ

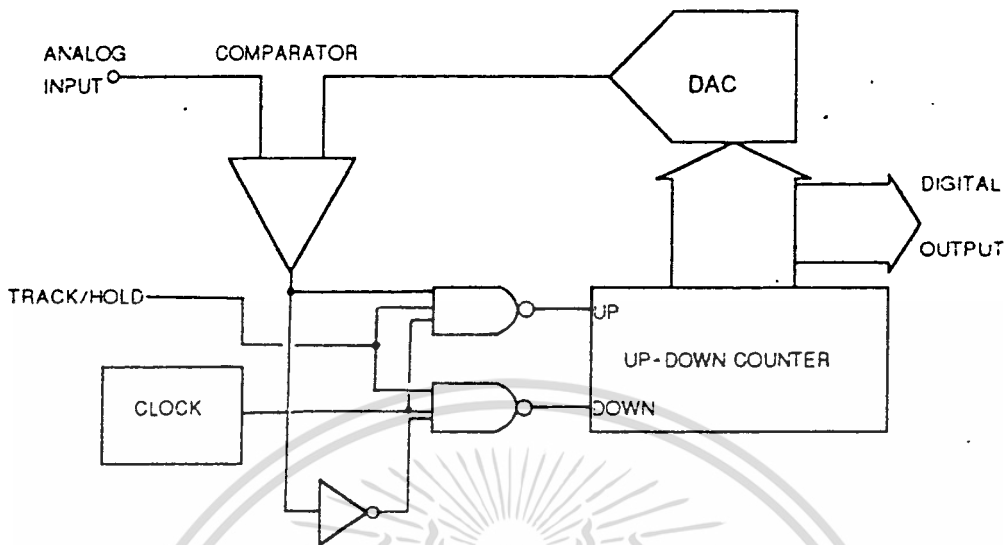
ติดตาม(track)สัญญาณอินพุทจนได้ค่าเท่ากันอีกก็จะ latch ค่าใหม่ไว้

จากลักษณะการทำงานดังกล่าว V_{in} จะต้องไม่เปลี่ยนแปลงเร็วกว่าการทำงานของเคาท์เตอร์ มิฉะนั้นค่าเอาท์พุทที่ได้จะไม่สอดคล้องกับสัญญาณอินพุท ตัวอย่างสัญญาณรูป sine ซึ่งจะเปลี่ยนแปลงขนาดได้มากที่สุดเท่ากับเต็มสเกล อัตราการเปลี่ยนแปลงจะเท่ากับอัตราการเปลี่ยนแปลงของเอาท์พุทของเคาท์เตอร์คือ $1 \text{ LSB/clock period}$ ดังนั้นถ้าต้องการให้ ADC ตามอินพุทได้จะต้องให้

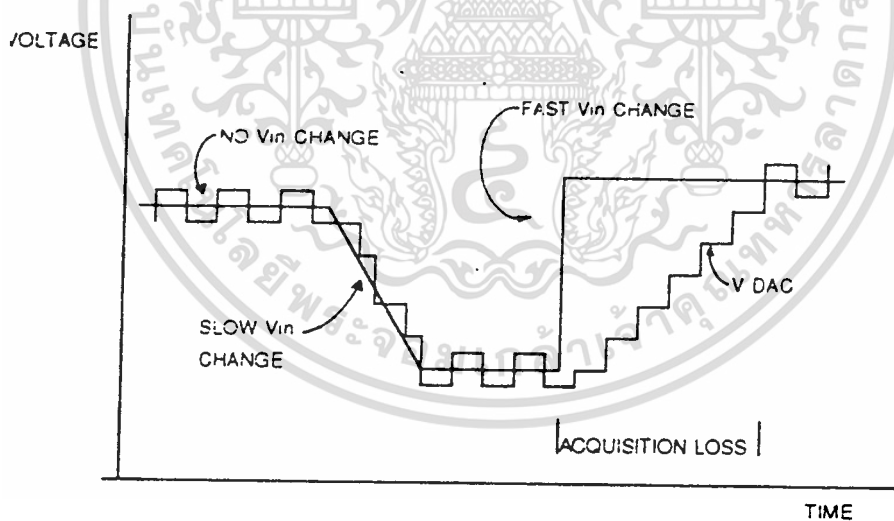
$$V_{FS} \omega_0 / 2 < V_{FS} f_c / 2^n$$

$$f_0 < f_c / \pi 2^n$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 ก) บล็อกไดอะแกรมของวงจร tracking converter



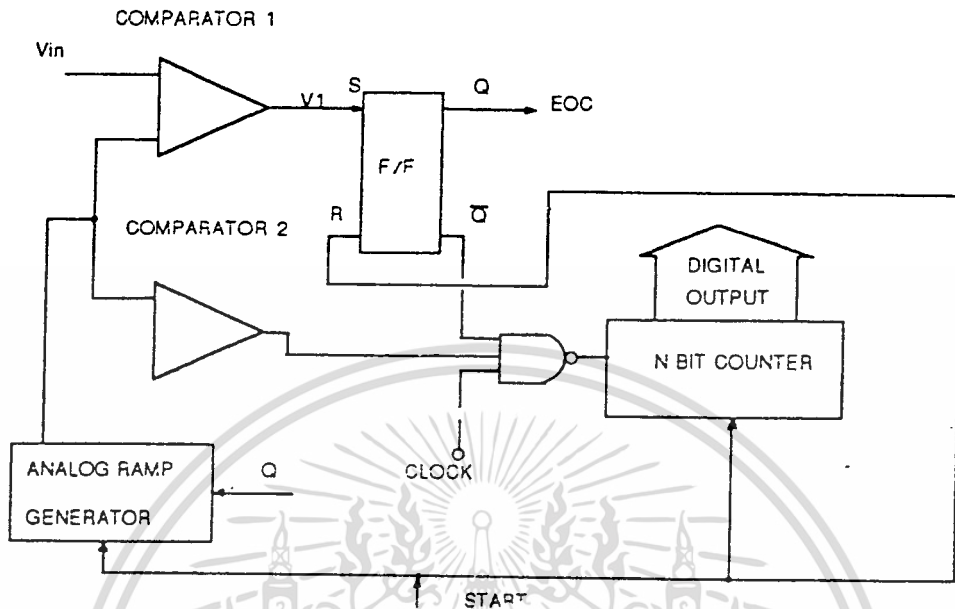
รูปที่ 2.22 ข) Timing Diagram

Integrating ADC

หัวใจสำคัญของวงจร ADC ชนิดนี้คือวงจร integrator เทคนิคของ ADC แบบ Integration ก็จะใช้สัญญาณ ramp ต่อเนื่องแทนสัญญาณขั้นบันไดจาก DAC ซึ่งแบ่งการทำงานได้ 2 แบบ คือ single slope converter และ dual slope converter

1) Single Slope Converter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.23 Single Slope Converter

สัญญาณอนาลอกแบบ ramp จะใช้เป็นแรงดันอ้างอิงที่เพิ่มขึ้นอย่างคงที่จากค่าต่ำกว่าศูนย์เล็กน้อยจนถึงค่าสูงสุดค่าเต็มสเกลเล็กน้อย ซึ่งเวลาที่จะใช้จากการสแกนของสัญญาณ ramp จากศูนย์ถึงค่าแรงดันอินพุทจะเป็นสัดส่วนกับแรงดันอินพุท

การ conversion จะเริ่มด้วยสัญญาณ start conversion ทำการรีเซ็ตไบนารีเคาท์เตอร์ และเริ่มสร้างสัญญาณ ramp จากแรงดันที่ต่ำกว่าศูนย์โวลต์ เมื่อสัญญาณ ramp ผ่านศูนย์โวลต์เอาท์พุทจากคอมพารเตอร์ 2 จะ high และเปิดเกตปล่อยพัลส์เข้าสู่เคาท์เตอร์

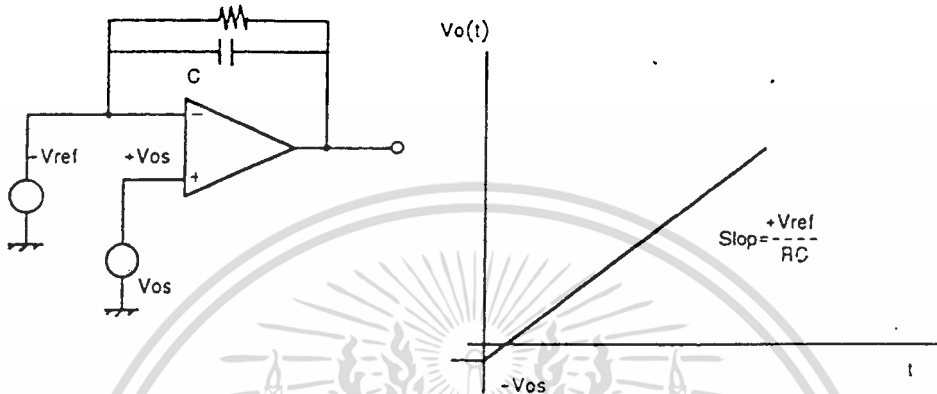
เคาท์เตอร์จะเริ่มนับจนกระทั่งสัญญาณ ramp มีขนาดเท่าแรงดันอนาลอกอินพุท V_{in} ในเวลานี้เอาท์พุทจากคอมพารเตอร์ 1 จะ high และปิดเกตไม่ให้ clock เข้าสู่เคาท์เตอร์ จำนวนพัลส์จากเคาท์เตอร์จะเป็นสัดส่วนแรงดันกับอินพุท เนื่องจาก $V_R = KT$ โดย R เป็น slope ของ ramp (ซึ่งคงที่) ในหน่วย โวลต์/วินาที และ T เป็นจำนวนในการเคาท์เตอร์หารด้วย f_c ซึ่งเป็นความถี่สัญญาณ clock ถ้าให้เลือกสโลปของ ramp เป็น $V_{FSR} f_c / 2^n$ จำนวนที่เคาท์เตอร์นับได้จะเท่ากับอัตราส่วนทางไบนารีหรือเท่ากับ V_{in}/V_{FSR}

เวลาในการเปลี่ยน T_c ของ ADC แบบนี้จะแปรเป็นสัดส่วนโดยตรงกับแรงดันที่อินพุท V_{in} เวลาที่ใช้ในการเปลี่ยนมากที่สุดเมื่อ $V_{in} = V_{FSR}$ คือ $T_{max} = 2^n / f_c$ และเช่นเดียวกับใน ADC แบบเคาท์เตอร์ ramp ค่าของรหัสเอาท์พุทสุดท้ายจะต่างจากค่าของ V_{in} ไม่เกิน 0.5 LSB

ในรูปที่ 2.24 แสดงวงจรกำเนิดแรงดัน ramp อย่างง่าย โดยการต่อแรงดันอ้างอิงกับ integrator เมื่อสวิตช์เปิด C จะทำการประจุและเพิ่มขนาดแรงดันเอาท์พุท ข้อเสียประการหนึ่งคือ

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนไว้เพื่อการใช้งานเท่านั้น ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อใช้งานไปนานๆการเปลี่ยนแปลงค่า RC ตามอุณหภูมิจะทำให้สโลปคลาดเคลื่อนด้วยเหตุนี้ ADC ชนิดนี้จึงไม่เป็นที่นิยมใช้ในปัจจุบัน



รูปที่ 2.24 วงจร ramp voltage generator อย่างง่าย และลักษณะของเอาต์พุต

2) Dual Slope Converter

ADC แบบ dual slope ได้รับการพัฒนาขึ้นมาเพื่อแก้ไขจุดบกพร่องของ single slope ADC การจัดวงจรแสดงในรูป 3.20 ในแต่ละวัฏจักรการทำงานของวงจร จะมี 2 ช่วงคือ T1 และ T2 ในเวลา T1 จะเป็นช่วงเวลาที่ได้รับการออกแบบให้มีค่าแน่นอนคงที่ ในช่วงเวลานี้สัญญาณอินพุตจะต่อเข้ากับ integrator ผ่านสวิตช์ S ซึ่งทำให้เอาต์พุตที่ถูกอินทิเกรต V_{int} เป็นรูปสัญญาณ ramp ที่ขนาดเพิ่มขึ้นทางบวกและสโลปขึ้นอยู่กับขนาดของ V_{in} จนกระทั่ง V_{int} ถึงค่าหนึ่งเมื่อสิ้นสุด T1

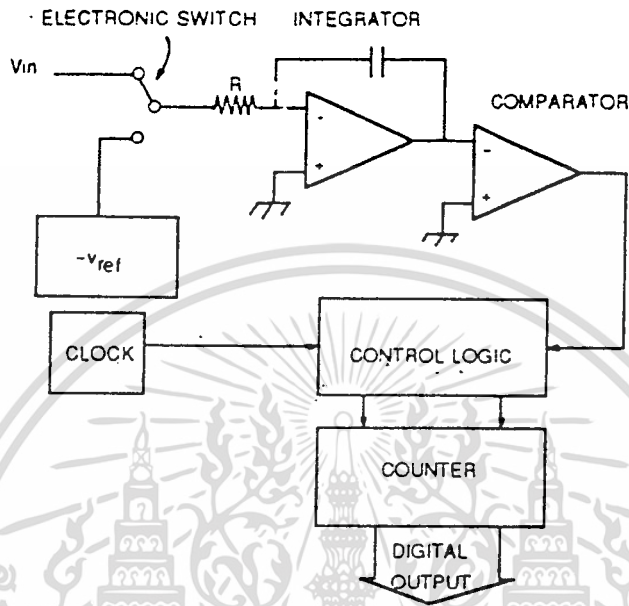
ในช่วงเวลา T2 อินพุตจะถูกตัดออกจาก integrator และต่อกับแรงดันอ้างอิงซึ่งมีค่าเป็นลบเข้ากับอินพุตของ integrator โดยการควบคุมทางลอจิก ในลักษณะเช่นนี้จะทำให้ V_{int} ลดลงด้วยสโลปคงที่จากการคายประจุผ่านลง V_{ref} เมื่อเริ่มต้นเวลา T2 เคาท์เตอร์จะรีเซ็ตและเริ่มนับ จนเมื่อ V_{int} มีค่าลดลงถึงศูนย์ คอมพารเตอ์จะเปลี่ยนสถานะไปบอกส่วนควบคุมลอจิกให้หยุดนับและเอาต์พุตของเคาท์เตอร์จะถูกแปลงเป็นรหัสดิจิทัล ความสัมพันธ์ระหว่างแรงดันอินพุตกับช่วงเวลาจะเป็นไปตามสมการ $T2 = T1 V_{in}/V_{ref}$

คุณลักษณะที่สำคัญของ dual slope มีหลายประการคือ ประการแรกความแม่นยำจะไม่ขึ้นอยู่กับเสถียรภาพของสัญญาณ clock และตัวเก็บประจุ แต่จะขึ้นอยู่กับแรงดันอ้างอิงและความเป็นเชิงเส้นของ integrator ประการที่สองการจำกัดสัญญาณรบกวนด้วยตัวเองของวงจรสามารถทำ

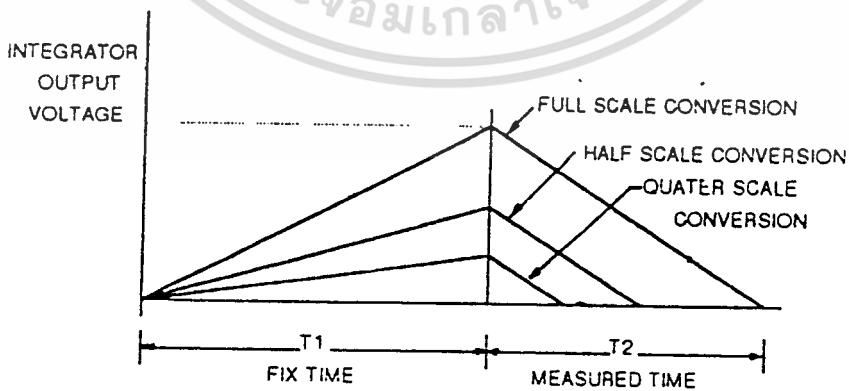
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ ถ้าเซตให้ T_1 มีค่าเท่าคาบเวลาของสัญญาณรบกวน

ส่วนข้อเสียที่สำคัญของ ADC นี้คือความเร็วในการ conversion ค่อนข้างต่ำจึงมักนิยมใช้กับเครื่องมือวัดที่ไม่ต้องการความเร็ว เช่น ดิจิตอลมิเตอร์



รูปที่ 2.25 ก) บล็อกไดอะแกรมของ dual slope ADC

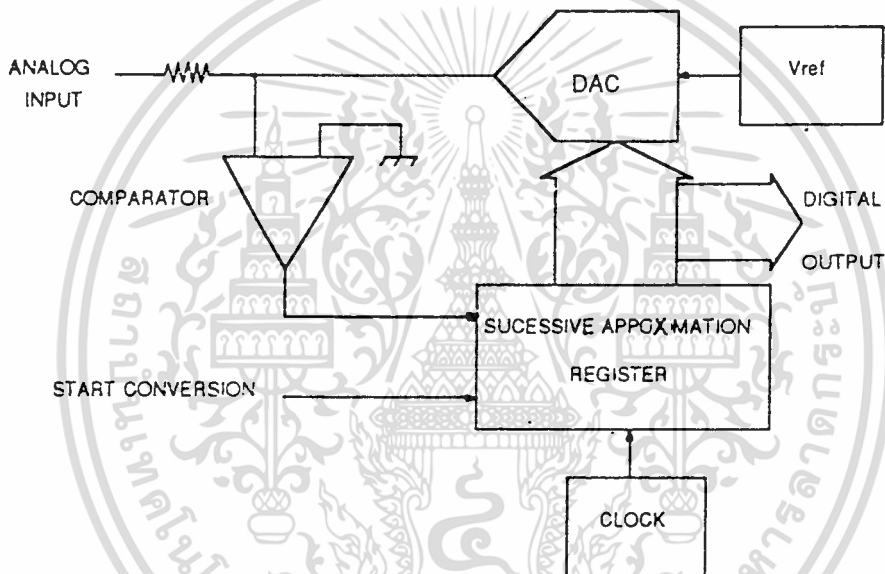


รูปที่ 2.25 ข) การทำงานของ dual slope ADC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Successive Approximation ADC

วงจร ADC ชนิดนี้ได้รับความนิยมในงานประยุกต์ที่ต้องการความเร็วปานกลางและค่อนข้างสูง การจัดวงจรจะคล้ายกับแบบเคาท์เตอร์ ที่ทำงานในลักษณะการป้อนกลับซึ่งบล็อกลอยเคาะแกรมในรูปที่ 3.21 แสดงฟังก์ชันต่างๆใน ADC ชนิดนี้ คอมพารเตอ์จะคอยเปรียบเทียบเอาต์พุตจาก DAC กับอนาลอกอินพุต V_{in} เอาต์พุตจะไปควบคุม Successive Approximation register (SAR) ซึ่งเป็นไอซี MSI(Medium Scale Integrated circuit) ที่ได้รับการออกแบบเป็นพิเศษเพื่อทำหน้าที่นี้โดยเฉพาะ



รูปที่ 2.26 บล็อกลอยเคาะแกรมของ Successive Approximation ADC

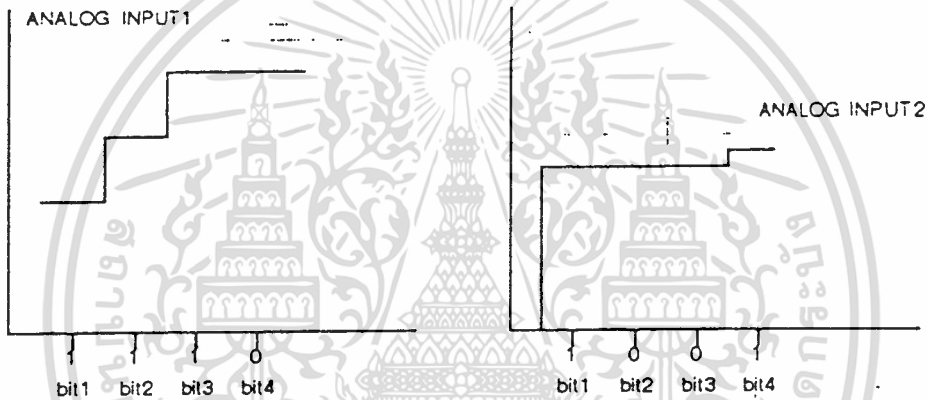
ในรูปที่ 2.26 แสดง timing diagram ของ ADC ที่มีระดับอนาลอก 1 และ 2 ที่ระดับ 1 เมื่อมี clock เข้าไป 1 ลูก จะทำให้ บิตที่ 4 (MSB) เป็น 1 บิตอื่นยังคงเป็น 0 DAC จะเปลี่ยนเอาต์พุตของ SAR เป็นอนาลอกเปรียบเทียบเทียบกับสัญญาณอนาลอกอินพุต ถ้าผลการเปรียบเทียบบอกว่าน้อยกว่าอินพุตก็ให้คงบิตนั้นเป็น 1 ไว้ แต่ถ้ามากกว่าจะให้บิตนั้นเป็น 0 จากนั้นทำการทดสอบบิตถัดไปโดยทำให้เป็น 1 หากผลรวมของสองบิตหรือบิตหลังมากกว่าก็ทำให้บิตนั้นเป็น 0 แต่ถ้าน้อยกว่าให้คง 1 ไว้ แล้วทดสอบบิตถัดไปตามกรรมวิธีดังกล่าวจนครบทุกบิตหรือจนกว่าเอาต์พุตจะต่างจาก V_{in} ไม่เกิน 1 LSB ในตัวอย่างแสดงการทำงานเมื่อ V_{in} ลดต่ำลงมาอีกระดับหนึ่งด้วยเช่นกัน

มีข้อจำกัดประการหนึ่งสำหรับการ conversion คือสัญญาณอนาลอกอินพุตจะต้องคงที่ใน

ช่วงเวลาที่ทำการเปลี่ยนแปลงสัญญาณ โดยเปลี่ยนได้ไม่เกิน 1/2 LBS ในช่วงสุดท้ายของการแปลงค่า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนสัญญาณดิจิทัลเอาต์พุตจะออกมาขนานกันทุกบิตแต่บางแบบให้เอาต์พุตออกมาในลักษณะอนุกรม

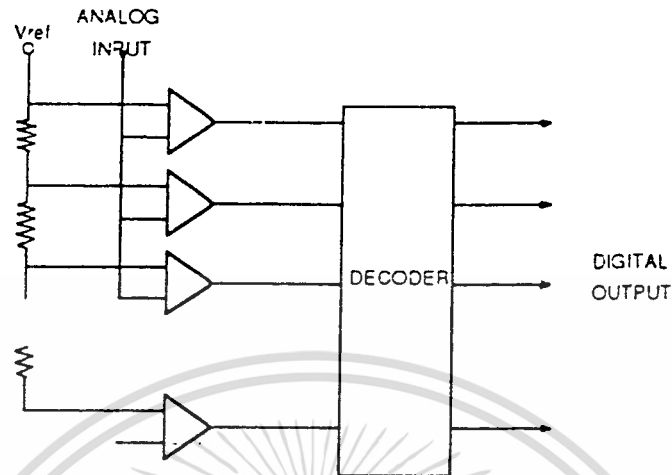
วงจร ADC แบบนี้สามารถทำงานได้สองโหมด คือ โหมดที่ทำงานอิสระและโหมดที่รอคำสั่ง start conversion จากภายนอก เวลาที่ใช้ในการเปลี่ยนสัญญาณใช้ $(n+1)$ ลูกของพัลส์ clock โดย clock ลูกแรกจะใช้ในการรีเซ็ตรีจิสเตอร์ภายใน คุณภาพของระบบจะขึ้นอยู่กับ DAC ที่ใช้ในระบบเป็นอย่างยิ่ง



รูปที่ 2.27 timing diagram ของ SAR

Parallel(Flash) ADC

สำหรับการแปลงสัญญาณที่ต้องการความเร็วสูงมากๆ เช่น ในระบบเรดาร์ จำเป็นต้องใช้ ADC แบบพิเศษ ที่เรียกว่า parallel ADC ซึ่งแสดงบล็อกไดอะแกรมดังรูปที่ 2.28 หลักการทำงานคือ จะใช้คอมพารเตอร์ ทำการเปรียบเทียบสัญญาณอนาล็อกอินพุต กับแรงดันอ้างอิงที่แบ่งแรงดันให้สอดคล้องกับรหัสดิจิทัล ซึ่งจะเห็นว่าอุปสรรคทางด้านความเร็วจะถูกจำกัดเพียง Propagation time ของคอมพารเตอร์เท่านั้น แต่อุปสรรคที่สำคัญของการพัฒนาวงจรชนิดนี้บนชิปไอซี ก็คือวงจรนี้ต้องการคอมพารเตอร์ถึง $2^n - 1$ ตัว แต่ก็ได้ ADC ชนิดที่ทำงานได้รวดเร็วที่สุดเช่นกัน



รูปที่ 2.28 บล็อกไดอะแกรมของ Flash ADC

ADC Chip

จากความก้าวหน้าทางเทคโนโลยีและการต้องการระบบ Data acquisition ที่มีขนาดเล็ก และกินกำลังงานต่ำจึงมีการนำเอาองค์ประกอบของระบบหลายส่วนมาลงไว้บนชิปตัวเดียวกัน การเลือกใช้งานชิป ADC สำหรับงานประยุกต์ต่างๆ ไปจึงมักพิจารณาถึง ว่าต้องการกี่บิต ความเร็วในการแปลง ขนาดของแรงดันอินพุตและไฟเลี้ยง ในการออกแบบระบบที่ใช้ ADC Chip จะขึ้นอยู่กับ การเลือกโหมดและใช้ Timing diagram ที่ผู้ผลิตเสนอมาให้ในคู่มือใช้งาน

โดย ADC Chip ที่ใช้ในโครงการนี้ คือ ADC0809 ซึ่งไม่มีวงจรสร้าง clock จึงจำเป็นที่จะต้องสร้างวงจรภายนอกเพื่อทำการสร้างสัญญาณ clock ให้แก่ Chip ส่วนรายละเอียดอื่นๆ ให้ดูได้ในภาคผนวก

บทที่ 8 การประกอบแขนกลเพื่อใช้ในการทดลอง

8.1 หลักการทำงานของแขนกล

โครงการนี้จะศึกษาการทำงานของแขนกลซึ่งจะควบคุมการทำงานโดยการควบคุมจะทำเป็น 3 แบบใหญ่คือ

1. การควบคุมโดยตรง (Manual control) จะบังคับแขนกลปรับเปลี่ยนระยะมุมและตำแหน่งโดยบังคับจากคีย์บอร์ดโดยตรง ซึ่งระยะตำแหน่งจะกะได้ด้วยสายตาและจะแสดงกราฟฟีกการบังคับ เมื่อเรากดคีย์บอร์ดตามที่กำหนดไว้ แขนก็จะขยับและจะอ่านค่าตำแหน่งนั้นไปแสดงที่กราฟฟีก การสั่งแขนให้จับหรือปล่อยโดยการกด H สำหรับการจับและ R สำหรับการปล่อย และยกเลิกการทำงานโดยการกด S

2. การควบคุมแบบปิด (Close loop control) เมื่อการส่งสัญญาณ input ตั้งให้อุปกรณ์ทำงาน จะมีตัวคอยตรวจจับสัญญาณ output ที่ออกไป แล้วส่งสัญญาณป้อนกลับ (feedback) กลับมายังทาง input เปรียบเทียบกับค่าที่ตั้งเอาไว้แล้วส่งสัญญาณ input ไปใหม่ ซึ่ง input ใหม่ก็จะแก้ค่าความผิดพลาดที่เกิดขึ้นจนกว่าค่าสัญญาณ feedback ที่ส่งมาเปรียบเทียบกับค่า input แล้วเท่ากันหรือเท่ากับค่า error ที่ยอมรับได้

ตัวจับสัญญาณนี้ จะเป็นตัวแปลงสัญญาณทางกายภาพ เช่น ความร้อน , ระยะทาง , ความดัน , ความเร็ว , ความแรง ฯลฯ ให้เป็นสัญญาณทางไฟฟ้า โดยอาจจะเป็นสัญญาณอะนาลอกหรือดิจิตอลก็ได้ ตัวแปลงสัญญาณนี้เรียกอีกอย่างหนึ่งว่า Transducer และ Sensor ซึ่งในโครงการนี้จะใช้ตัวแปลงสัญญาณแบบระยะทางอย่างเดียว

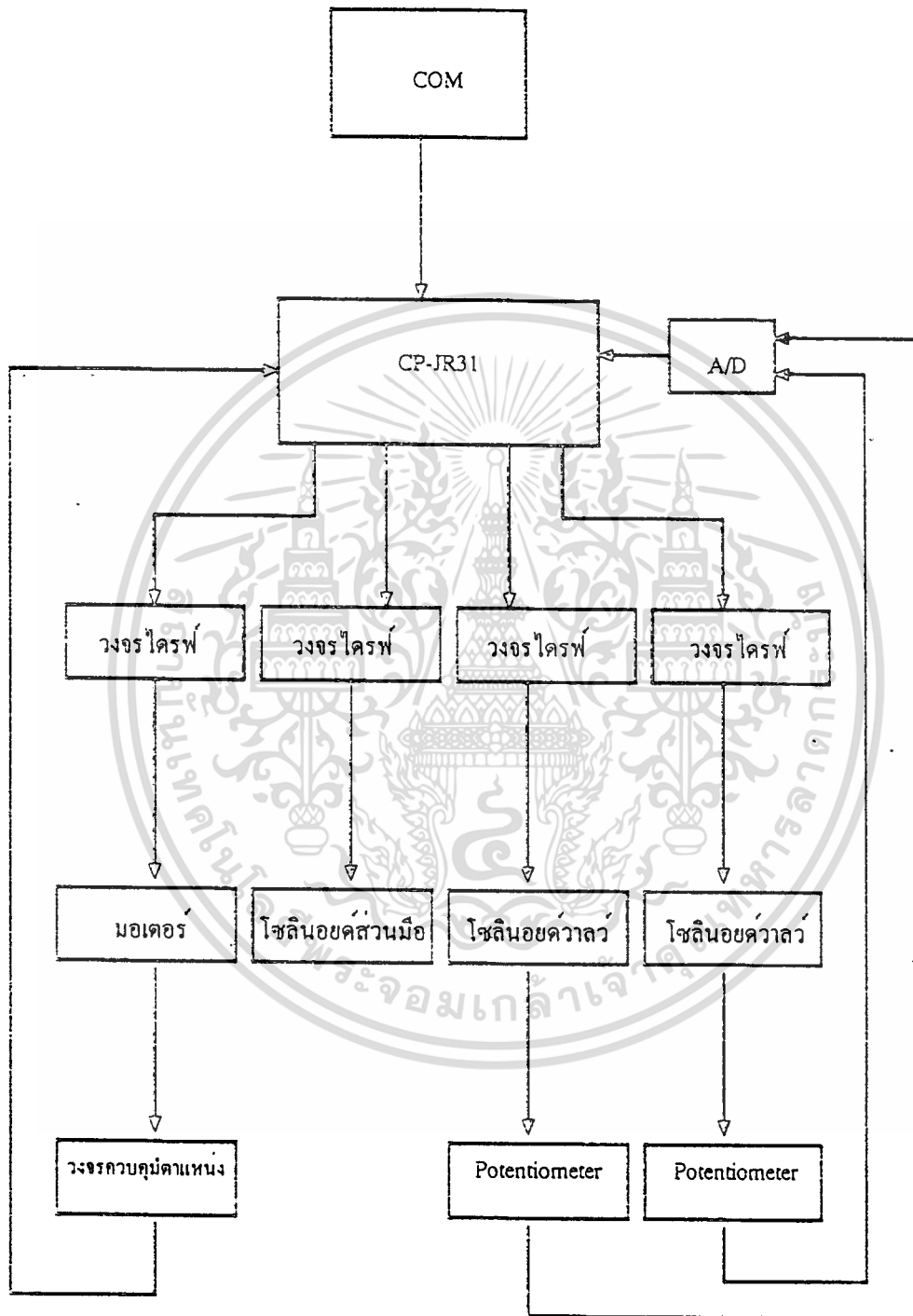
การควบคุมแบบปิดในโครงการนี้จะแบ่งเป็น 2 ชนิดด้วยกันคือ

2.1 การควบคุมแบบอัตโนมัติ (Auto control) การควบคุมที่เราใส่ตำแหน่งที่เราต้องการลงไป แล้วแขนกลก็จะขยับไปยังตำแหน่งนั้นโดยอัตโนมัติ

2.2 การควบคุมแบบอ่านจากไฟล์ (Read file to control) คือการควบคุมแขนกลโดยการอ่านข้อมูลของตำแหน่งจากไฟล์ข้อมูลที่บันทึกไว้ ซึ่งข้อดีของการควบคุมแบบอ่านจากไฟล์คือสามารถกำหนดตำแหน่งให้มันเคลื่อนที่ได้ที่หลายตำแหน่ง ส่วนการทำงานของ การควบคุมแบบอ่านจากไฟล์ที่จะทำงานให้แขนกลขยับนั้นจะเหมือนกับการควบคุมแบบอัตโนมัติ

รูปที่ 3.1 แสดงหลักการทำงานอย่างคร่าว ๆ ของการทำของแขนกลโดยใช้คอมพิวเตอร์เป็นตัวควบคุม หลักการทั้งหมด ซึ่งโปรแกรมควบคุมนั้นจะเขียนด้วยภาษา C แล้วส่งข้อมูลผ่านทาง Serial Port ของคอมพิวเตอร์เข้ามาทางบอร์ด CP-JR 31

ซึ่งบอร์ดประกอบด้วยส่วนต่าง ๆ ดังนี้



รูปที่ 3.1 แสดงหลักการคร่าว ๆ ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.2 ขนาดและโครงสร้างของแขนกลนิวแมติก

ในการทดลองแขนกลจะประกอบไปด้วยส่วนหลักๆ ดังนี้

1. โครงแขนกล ซึ่งจะกล่าว ถึงลักษณะและขนาดในช่วงต่อไป

- ส่วนฐาน เป็นจานวงกลมขนาดเส้นผ่านศูนย์กลาง 25 cm. หนา 12 cm. พร้อมแกนเพลลา
ขนาด 1 นิ้ว ยาวประมาณ 30 cm.

- แผ่นเหล็กหนา 2/8 นิ้ว ลักษณะต่างๆจำนวน 10 แผ่นที่ประกอบกันเป็นแขนกล

- แผ่นเหล็กที่นำมาพับและตัดให้เป็นส่วนของมือจับ

2. วงจรนิวแมติก ประกอบด้วย

- วาล์ว 5/2 D.C.V. set by solenoid(24 V.) and reset by spring ขนาดเกลิยว M1 3ตัว

- วาล์ว 2/2 D.C.V. set by solenoid(24 V.) and reset by spring ขนาดเกลิยว M2 4ตัว

- speed control valve 4 ตัว

- ระบายสูบ ขนาดเส้นผ่านศูนย์กลาง 20 m.m. ระยะ stroke 10 c.m. 2 ระบาย

- ระบายสูบ ขนาดเส้นผ่านศูนย์กลาง 20 m.m. ระยะ stroke 3 c.m. 1 ระบาย

(compact cylinder)

- สายลม ขนาดเส้นผ่านศูนย์กลางภายใน 4 m.m. เส้นผ่านศูนย์กลางภายนอก 6 m.m.
ยาว 20 เมตร

- ข้อต่อ 3ทาง ขนาด 6 m.m. 2 ตัว

- ข้อต่ออง 90° ขนาด 6 m.m. เกลิยว M1 6 ตัว

- ข้อต่ออง 90° ขนาด 6 m.m. เกลิยว M2 8 ตัว

3. อุปกรณ์ควบคุม

- เครื่อง computer 1 เครื่อง

- DC motor 1 ตัว

- บอร์ด CP-JR31

- วงจร ไคร์ฟมอเตอร์ และ โซลินอยด์วาล์ว

- sensor ประกอบด้วย potentiometer

- limit switch

- วงจรแปลงไฟ 5 โวลต์ และ 24 โวลต์

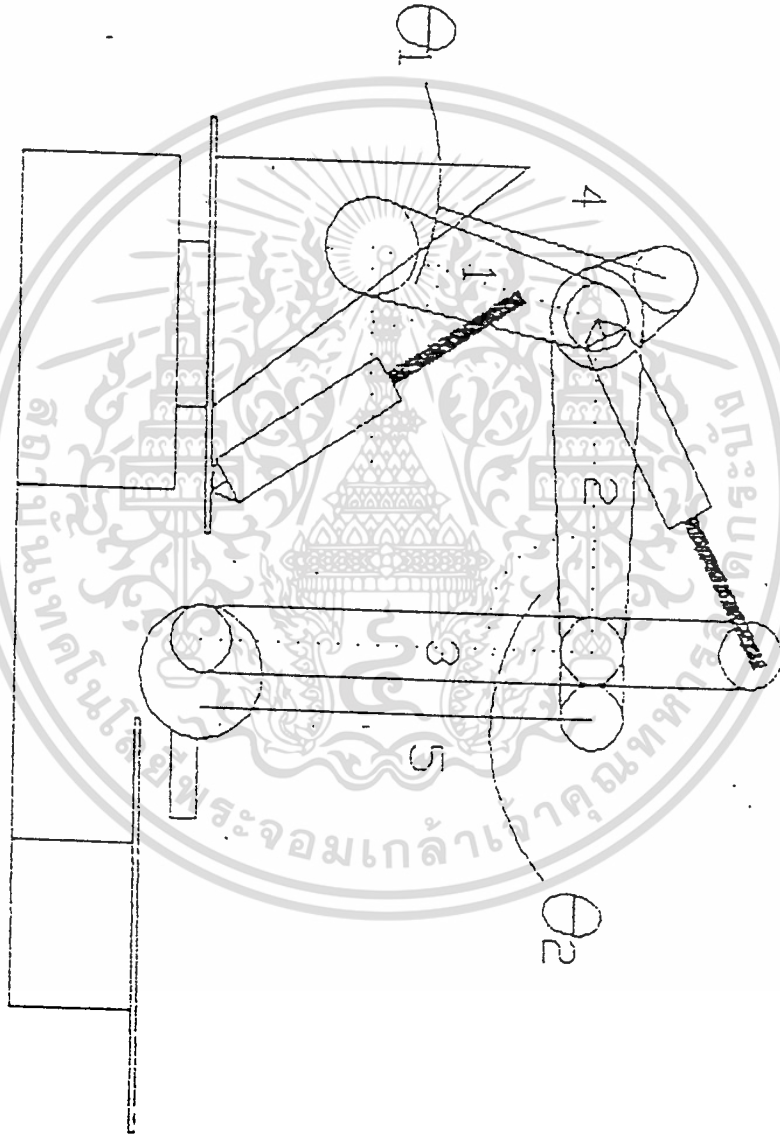
- วงจรเปลี่ยนจากอนาลอกเป็นดิจิทัล

4. อุปกรณ์อื่นๆ

- โต๊ะขนาด 120*150 cm. สูง 120 cm.

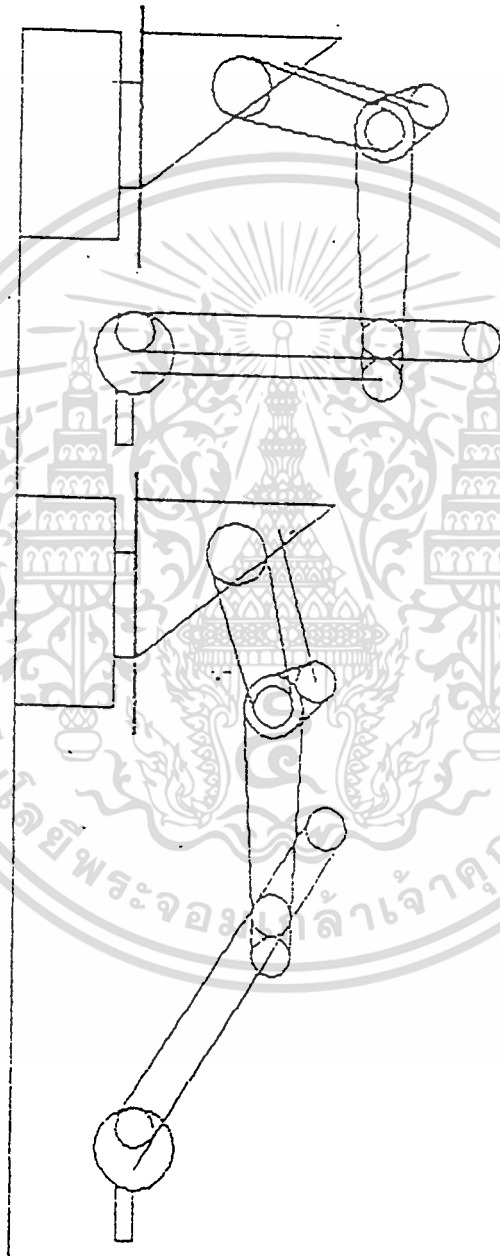
- เฟืองตรง 1 คู่ อัตราทด 1:5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



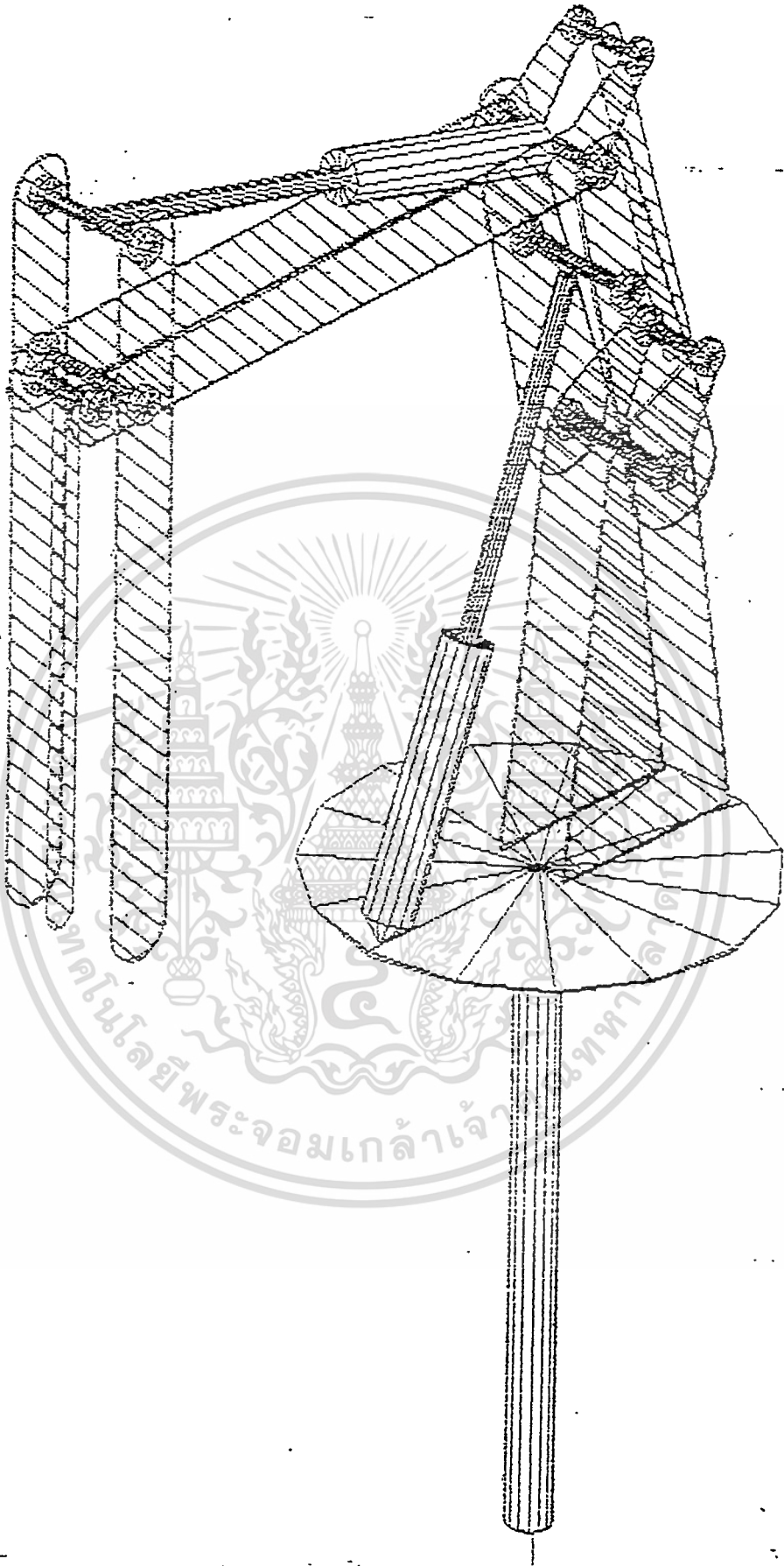
รูปที่ 3.3 แสดงโครงสร้างของแขนกล 2 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



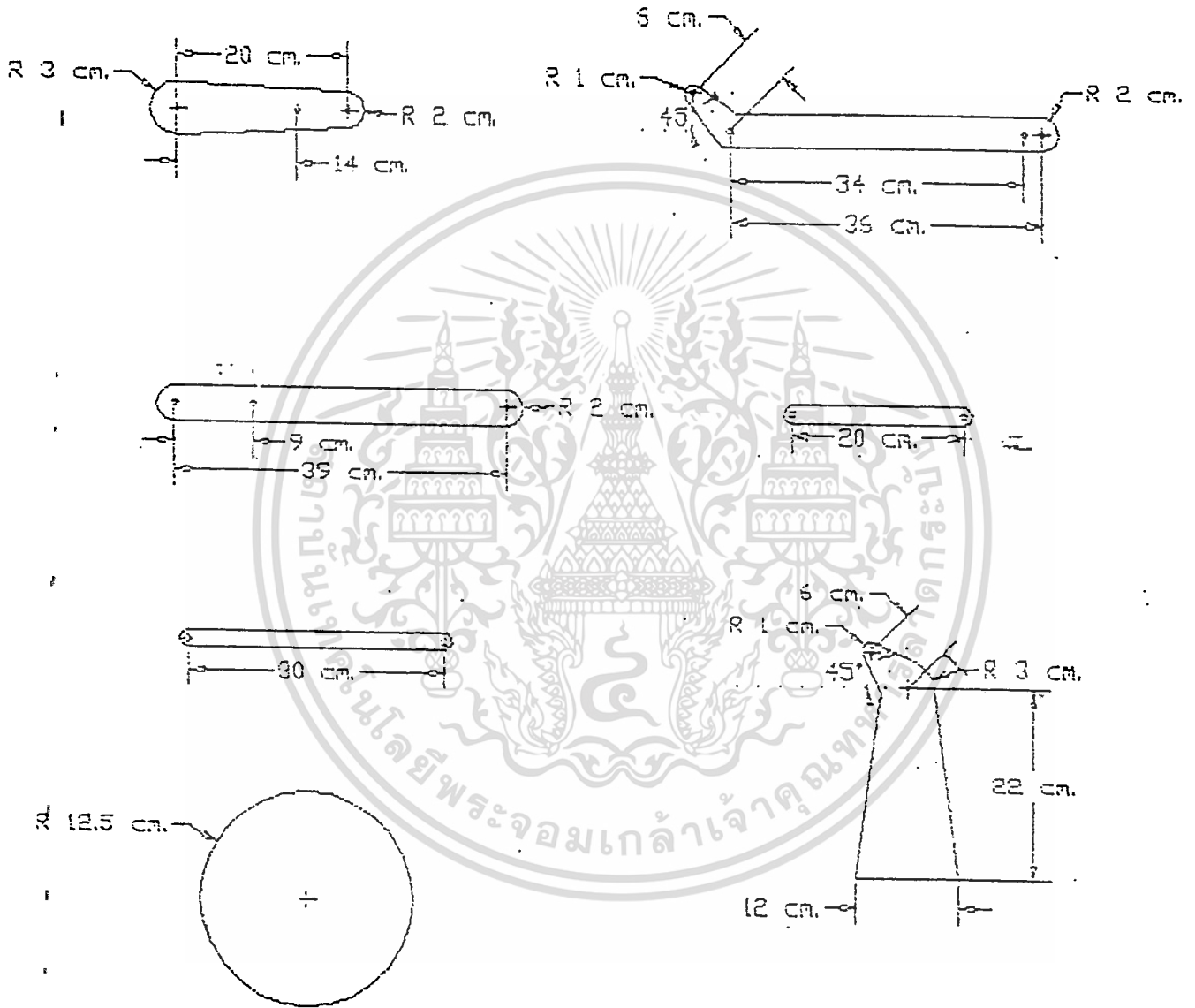
รูปที่ 3.4 แสดงโครงสร้างของแขนกล 2 มิติ เมื่อยึดจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้เพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงโครงสร้างของแขนกล 3 มิติ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การเขียนเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดงขนาดชิ้นส่วนของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8.3 การคำนวณตำแหน่งของแขนกลนิวแมติก

จากหัวข้อ 2.1 เรื่องการคำนวณหาตำแหน่งของแขนกล เมื่อนำเอาวิธีคิดและการคำนวณตำแหน่งมาประยุกต์ใช้กับลักษณะของแขนกลที่ใช้ในการทดลองแล้วจะสามารถหาสมการต่างๆ ได้ดังนี้

$$L1 \cos \theta_1 - L3 \cos \theta_2 + L2 = R \text{ -----①}$$

$$L1 \sin \theta_1 + H1 - L3 = H \text{ -----②}$$

จากสมการ① และ② จะสามารถแก้สมการออกมาได้ดังนี้

$$A \sin^2 \theta_1 + B \sin \theta_1 + D = 0$$

โดย

$$A = [2 L1(H1 - H)]^2 + [2 L1(L2 - R)]^2$$

$$B = 4 L1(H1 - H) C$$

$$C = (L2 - R)^2 + (H1 - H)^2 - (L3^2 - L1^2)$$

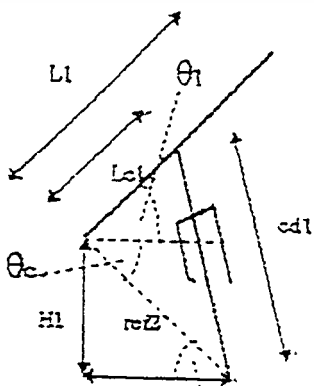
$$D = C^2 - [2 L1(L2 - R)]^2$$

$$\sin \theta_1 = \{-B + \sqrt{(B^2 - 4AD)}\} / 2A$$

$$\cos \theta_2 = (L1 \cos \theta_1 + L2 - R) / L3$$

เมื่อได้มุม θ_1 และ θ_2 ออกมาแล้วก็จะสามารถนำค่าที่ได้ไปคำนวณหาระยะยึดหรือหัดของกระบอสูบแต่ละอันได้โดยใช้หลักการคำนวณดังนี้

กระบอสูบตัวที่ 1



$$\theta_c = \arctan (H1/ref1)$$

$$ref2 = \sqrt{(H1^2 + ref1^2)}$$

$$cd1^2 = Lc1^2 + ref2^2 - 2 Lc1 * ref2 * \cos(\theta_1 + \theta_c)$$

โดย $H1 = 18 \text{ cm.}$

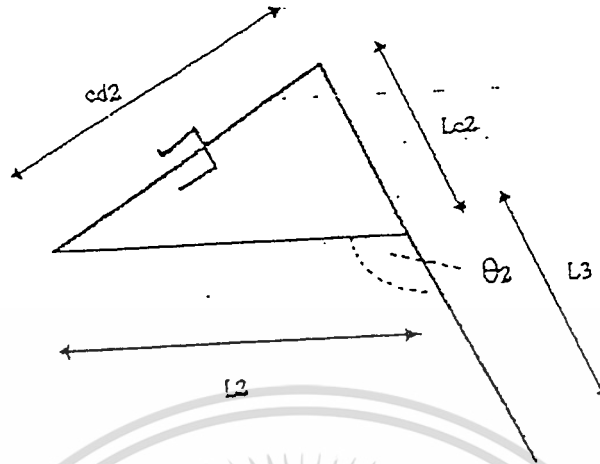
$$Lc1 = 14 \text{ cm.}$$

$$ref1 = 13 \text{ cm.}$$

$$L1 = 20 \text{ cm.}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบอกลูกสูบตัวที่ 2



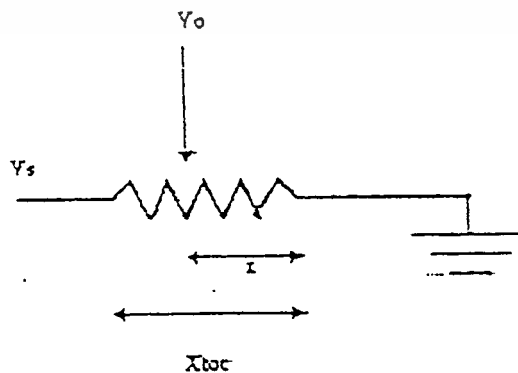
$$cd2 = Lc2^2 + L2^2 - 2 Lc2 * L2 * \cos(\pi - \theta_2)$$

โดย $L2 = 34 \text{ cm.}$

$Lc2 = 9 \text{ cm.}$

เมื่อคำนวณค่า $cd1$ และ $cd2$ ออกมาได้แล้ว ค่อยจากนั้นก็ยังสามารถนำค่าที่ได้ไปทำการคำนวณว่า potentiometer มีค่าความต้านทานเปลี่ยนแปลงไปเท่าไรและจะมีความต่างศักย์ที่ส่งออกมาเป็น output ก็โวลต์ เพื่อที่จะสามารถนำผล output ที่ได้ ออกมาขึ้นไปผ่านวงจร A/D เพื่อแปลงสัญญาณให้เป็นสัญญาณ digital ผ่านบอร์ด CP-JR31 เข้าสู่ computer เพื่อใช้ในการวิเคราะห์ตำแหน่งต่อไป

การคำนวณสัญญาณ output ที่ส่งออกมาจาก potentiometer สามารถคำนวณได้ดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_x = X * R_{tot} / X_{tot}$$

$$V_o = V_s * R_x / R_{tot} = V_s * X / X_{tot}$$

โดย $V_s = 5 \text{ V}$.

$$X_{tot} = 7 \text{ cm.}$$

$$R_{tot} = 10 \text{ K}\Omega$$

ค่า X ที่จะใช้ในการคำนวณหาสัญญาณ output ของ potentiometer นั้นสามารถหาได้จากผลต่างของ $cd1$ หรือ $cd2$ กับค่าความยาวปกติของกระบอกลูกขี้ผึ้ง (เนื่องจากการบิด potentiometer เข้ากับกระบอกลูกขี้ผึ้งนั้น เมื่อกระบอกลูกขี้ผึ้งหัก ค่าความต้านทานของตัว potentiometer จะมีค่าเท่ากับ 0 โอห์ม)

กระบอกลูกขี้ผึ้งตัวที่ 1 ; $X1 = cd1 - \text{ความยาวของกระบอกลูกขี้ผึ้งหัก}$

กระบอกลูกขี้ผึ้งตัวที่ 2 ; $X2 = cd2 - \text{ความยาวของกระบอกลูกขี้ผึ้งหัก(ใช้งาน)}$

ความยาวของกระบอกลูกขี้ผึ้งหัก(ใช้งาน) = 28.5 cm.

ความยาวของกระบอกลูกขี้ผึ้งหัก = 25.5 cm.

ความยาวของกระบอกลูกขี้ผึ้งหัก = 35.5 cm.

ความละเอียดของสัญญาณ output ที่ได้ขึ้นอยู่กับ วงจร A/D ที่ใช้ เนื่องจากในการทดลองใช้วงจร A/D ขนาด 8 บิต ดังนั้นสัญญาณ digital ที่ให้ได้จะมีค่าตั้งแต่ 0 - 255 นั่นก็คือ มันสามารถให้ค่าได้ 256 ค่า

ดังนั้นความละเอียดที่ได้ = $V_s / 256$

$$= 5 / 256 = 0.01953 \text{ โวลต์}$$

นั่นก็คือเมื่อค่า V_o เปลี่ยนแปลงไป 0.01953 โวลต์ ระดับสัญญาณที่ได้จากวงจร A/D ถึงจะเปลี่ยนไป 1 ระดับ

3.4 วงจรนิวแมติกที่ใช้ควบคุมแขนกล

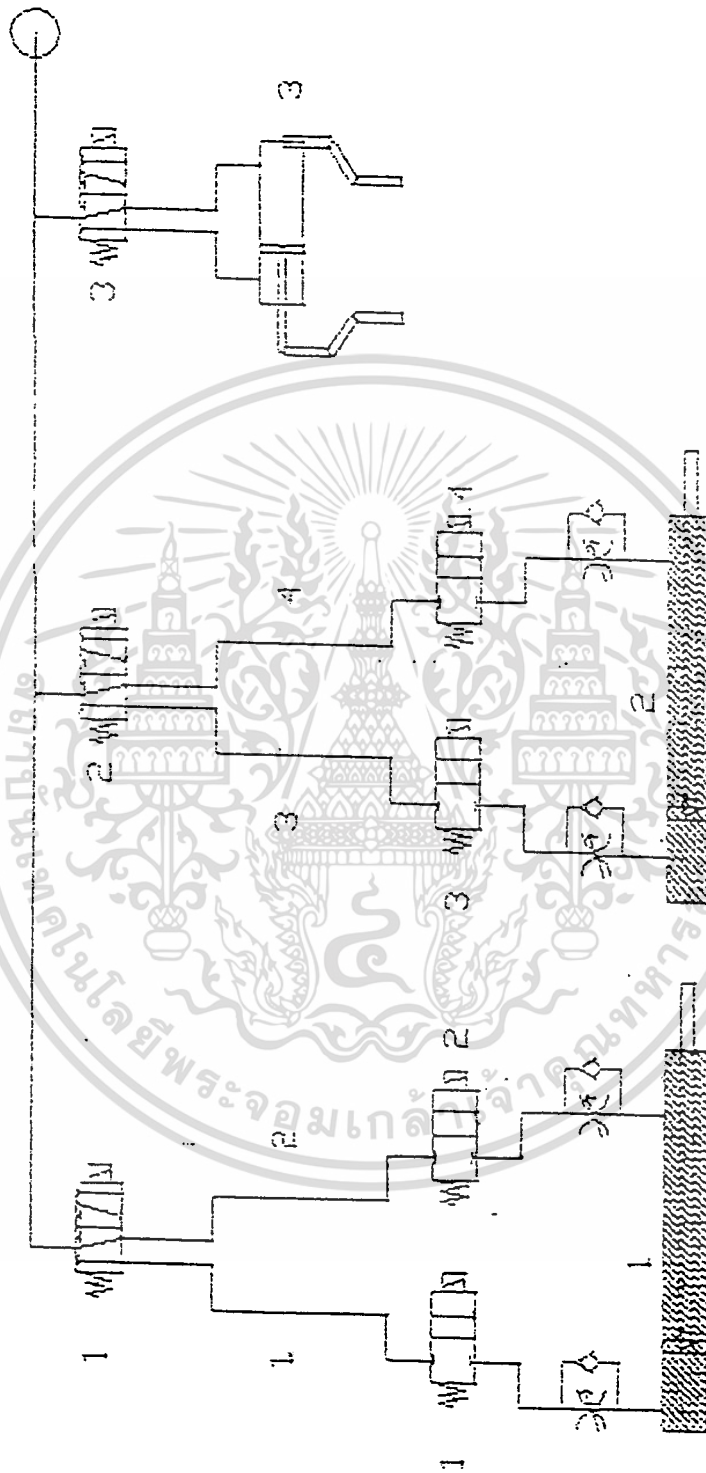
1. วาล์ว 5/2 D.C.V. set by solenoid(24 V.) and reset by spring ขนาดเกลียว M1 3 ตัว
2. วาล์ว 2/2 D.C.V. set by solenoid(24 V.) and reset by spring ขนาดเกลียว M2 4 ตัว
3. speed control valve 4 ตัว
4. กระบอกลูกสูบ ขนาดเส้นผ่านศูนย์กลาง 20 m.m. ระยะ stroke 10 c.m. 2 กระบอก
5. กระบอกลูกสูบ ขนาดเส้นผ่านศูนย์กลาง 20 m.m. ระยะ stroke 3 c.m. 1 กระบอก
(compact cylinder)
6. สายลม ขนาดเส้นผ่านศูนย์กลางภายใน 4 m.m. เส้นผ่านศูนย์กลางภายนอก 6 m.m. ยาว 20 เมตร
7. ข้อต่อ 3 ทาง ขนาด 6 m.m. 2 ตัว
8. ข้อต่ออง 90° ขนาด 6 m.m. เกลียว M1 6 ตัว
9. ข้อต่ออง 90° ขนาด 6 m.m. เกลียว M2 8 ตัว

หลักการทํางาน

เมื่อต้องการให้กระบอกลูกสูบ 1 เคลื่อนที่ออก จะต้องมีการป้อนสัญญาณไฟ 24V. เข้าที่ 5/2 D.C.V. ตัวที่ 1 และ 2/2 D.C.V. ตัวที่ 1 และ 2 จะเป็นผลทำให้ลมสามารถผ่านวาล์ว 5/2 เบอร์ 1 และผ่านวาล์ว 2/2 เบอร์ 1 เข้าไปยังกระบอกลูกสูบเบอร์ 1 ในด้านที่ทำให้กระบอกลูกสูบเบอร์ 1 เคลื่อนที่ออกไปได้ การที่จะทำให้กระบอกลูกสูบเบอร์ 1 หยุดค้างที่ตำแหน่งใดก็ให้หยุดจ่ายไฟเข้าที่วาล์วทั้ง 3 ตัว ถ้าต้องการให้กระบอกลูกสูบเบอร์ 1 เคลื่อนที่กลับ ก็ให้ทำการจ่ายไฟเข้าที่วาล์ว 2/2 เบอร์ 1 และ 2 ก็จะทำให้ลมสามารถผ่านวาล์ว 5/2 เบอร์ 1 และ ผ่านวาล์ว 2/2 เบอร์ 2 ผ่านเข้าไปในกระบอกลูกสูบเบอร์ 1 ในด้านที่มีผลให้กระบอกลูกสูบเบอร์ 1 เคลื่อนที่เข้า เมื่อต้องการให้กระบอกลูกสูบเบอร์ 1 หยุดค้างตำแหน่ง ก็ให้หยุดจ่ายไฟเข้าที่วาล์ว 2/2 เบอร์ 1 และ 2

การควบคุมกระบอกลูกสูบเบอร์ 2 ให้เคลื่อนที่เข้าและออกหรือหยุดค้างตำแหน่งก็สามารถควบคุมได้ เหมือนกับกระบอกลูกสูบเบอร์ 1 แต่วาล์วที่จะใช้ในการควบคุมการทํางาน ก็คือ วาล์ว 5/2 เบอร์ 2 และ วาล์ว 2/2 เบอร์ 3 และ 4

การควบคุมการเคลื่อนที่ของกระบอกลูกสูบเบอร์ 3 ซึ่งทำหน้าที่เป็นมือจับ เมื่อต้องการให้มือจับสิ่งของก็ให้จ่ายไฟ 24 V. เข้าที่วาล์ว 5/2 เบอร์ 3 ซึ่งมีผลให้ลมผ่านเข้าไปในกระบอกลูกสูบเบอร์ 3 ในด้านที่ทำให้กระบอกลูกสูบเบอร์ 3 เคลื่อนที่เข้านั้นจะต้องทำการจ่ายไฟเข้าที่วาล์ว 5/2 เบอร์ 3 ตลอดเวลาที่ทำการจับสิ่งของนั้นอยู่ เมื่อต้องการจะวางสิ่งของก็ให้หยุดจ่ายไฟเข้าที่วาล์ว 5/2 เบอร์ 3 เท่านั้น



รูปที่ 3.7 แสดงวงจรนิวแมติกที่ใช้ควบคุมแขนกล

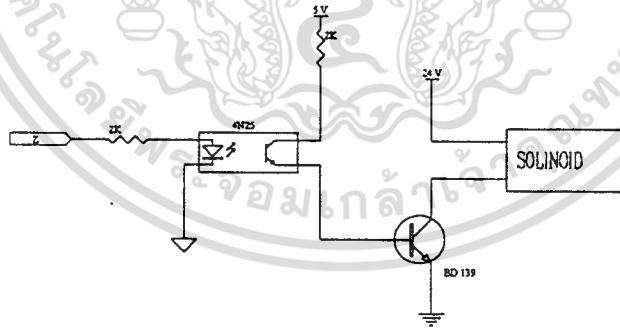
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 วงจรไคร์ฟและวงจรเปลี่ยนจากอนาล็อกเป็นดิจิตอล

วงจรไคร์ฟจะใช้ในการควบคุมมอเตอร์และโซลินอยด์วาล์ว ให้การทำงานเป็นไปได้ตามต้องการโดยวงจรไคร์ฟที่จะใช้นั้นจะรับสัญญาณจากบอร์ด CP-JR 31 ซึ่งจะเป็นสัญญาณไฟฟ้าไปขับมอเตอร์หรือโซลินอยด์วาล์ว โดยมีโฟโตคัปเลอร์เป็นตัวแยก 2 ส่วนออกจากกันเพื่อป้องกันการเสียหายที่คาดไม่ถึงได้ ส่วนวงจรเปลี่ยนจากอนาล็อกเป็นดิจิตอลเพื่อแปลงสัญญาณไฟฟ้าที่ได้จากโพเทนชิโอมิเตอร์ซึ่งเป็นอนาล็อกไปเป็นดิจิตอล

รูปที่ 3.8 เป็นวงจรขับโซลินอยด์วาล์ว เมื่อพอร์ต Z ได้รับสัญญาณจากบอร์ด JR-CP 31 จะเกิดกระแสไหลผ่านโฟโตคัปเลอร์ ทำให้ทรานซิสเตอร์ที่ต่อกับโซลินอยด์วาล์วทำงาน จะเกิดกระแสไหลผ่านโซลินอยด์ทำให้โซลินอยด์เปิดวาล์วสำหรับควบคุมลม เพื่อยกแขนให้ไปตามระดับที่ต้องการ

โซลินอยด์วาล์วดึงกระแส 80 mA และ β ของ Q_2 มีค่าประมาณ 40 จะได้ I_B ของ Q_2 80mAหาร 40 เท่ากับ 2mA โวลต์ที่ตกคร่อม Q_1 เมื่อ Q_1 ทำงานมีค่า 0.2 โวลต์ ตกคร่อม V_{BE} ของ Q_2 0.6 โวลต์ จะได้ ค่า $R_2 = (5-0.2-0.6)/2mA = 2.1 k\Omega$



รูปที่ 3.8 แสดงวงจรขับโซลินอยด์วาล์ว

ส่วนรูปที่ 3.9 แสดงถึงวงจรขับมอเตอร์การทำงานคือ เมื่อมีสัญญาณเข้ามาทางพอร์ต X และ Y เป็น

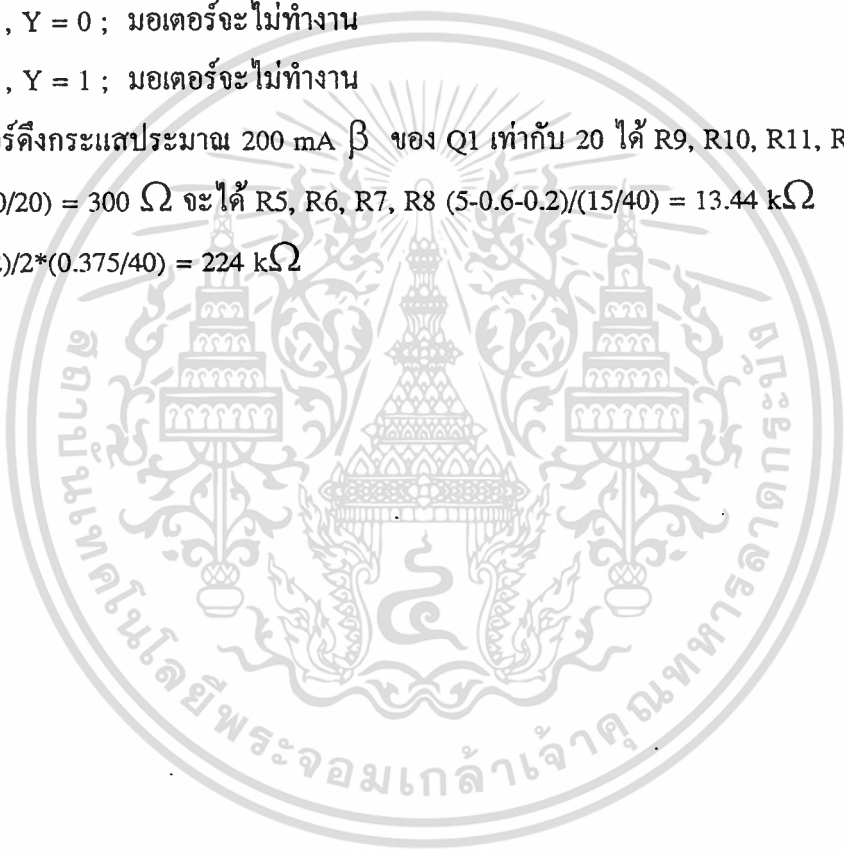
$X = 1, Y = 0$; จะเป็นการหมุนขวา เมื่อ Y ผ่าน G1 จะได้เอาต์พุต 1 และเอาต์พุตของ G3 จะได้ 1 เมื่อ X ผ่าน G2 จะได้เอาต์พุต 0 และเอาต์พุตของ G4 จะได้ 0 Q5 On , Q6 Off จะได้ Q7, Q9, Q12, Q1, Q4 On, และ Q8, Q10, Q11, Q2, Q3 Off

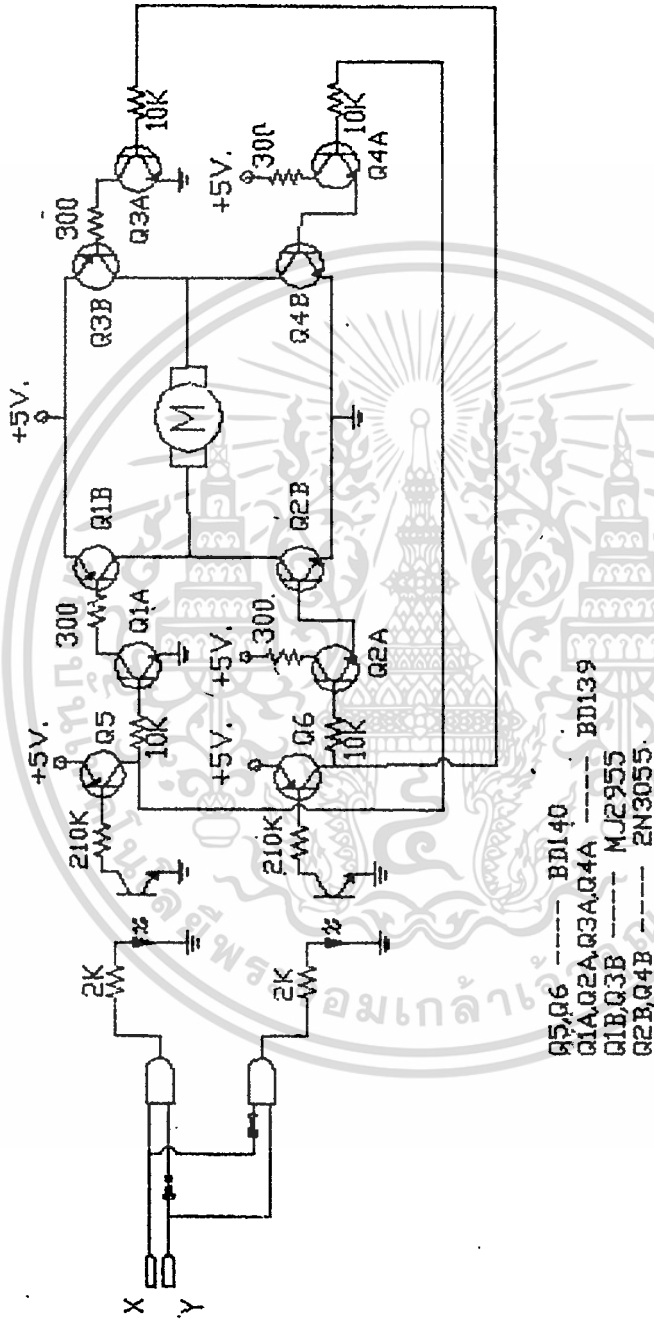
$X = 0, Y = 1$; จะเป็นการหมุนซ้าย เมื่อ X ผ่าน G2 จะได้เอาต์พุต 1 และเอาต์พุตของ G4 จะได้ 1 เมื่อ Y ผ่าน G1 จะได้เอาต์พุต 0 และเอาต์พุตของ G3 จะได้ 0 Q5, Q7, Q9, Q12, Q1, Q4 On และ Q6, Q8, Q10, Q11, Q2, Q3 Off

$X = 0, Y = 0$; มอเตอร์จะไม่ทำงาน

$X = 1, Y = 1$; มอเตอร์จะไม่ทำงาน

มอเตอร์ดึงกระแสประมาณ 200 mA β ของ Q1 เท่ากับ 20 ได้ R9, R10, R11, R12 $(5-0.2-0.60)/(300/20) = 300 \Omega$ จะได้ R5, R6, R7, R8 $(5-0.6-0.2)/(15/40) = 13.44 k\Omega$ ได้ R3, R4 $(5-0.6-0.2)/2*(0.375/40) = 224 k\Omega$

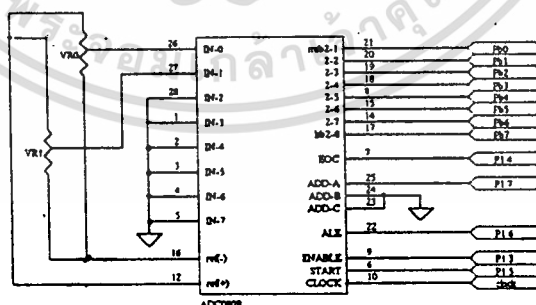




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้ใช้งานเพื่อการศึกษานำไปใช้ประโยชน์ด้านการค้า
รูปที่ 3.9 แสดงวงจรไดรฟ์มอเตอร์
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.10 เป็นรูปแสดงวงจรอนาล็อกเป็นดิจิทัลใช้ เป็น แปลงสัญญาณอนาล็อกเป็นดิจิทัล มีสัญญาณอินพุต 8 ช่อง และมีเอาต์พุต 8 บิต ใช้ขา ADD A,B,C เป็นตัวเลือกช่องสัญญาณ โดย ช่อง 0 (IN0) เลือกโดยกำหนด ADD A,B,C เป็น 000 และช่อง 1(IN1) เลือกโดยกำหนด ADD A,B,C เป็น 100 ต่อขา Vcc(+5v) เข้ากับ Ref (+) และขา GND เข้ากับ Ref (-) เพื่อให้การแปลงสัญญาณได้ในช่วง 0-5 V ขา CLOCK ต่อมาจากสัญญาณสร้าง CLOCK 550 kHz การแปลงสัญญาณควบคุมโดยลำดับขั้นตอนเป็นดังนี้

1. เริ่มด้วยการเลือกขา ADD A เพื่อเลือกช่องสัญญาณที่จะแปลงโดย 0 แปลงสัญญาณขา IN0 และ 1 แปลงสัญญาณขา IN1
2. Set ขา ALE (Address Latch enable) ให้แอดแตรสที่เลือกเข้าสู่การแปลงสัญญาณ
3. Set ขา start เพื่อสั่งให้เริ่มแปลงสัญญาณ
4. ตรวจสอบขา EOC (End of converter) เมื่อเป็น 0 แสดงว่าการแปลงสัญญาณยังไม่เสร็จ ถ้าหากเป็น 1 แสดงว่าการแปลงสัญญาณเสร็จสิ้นลงแล้ว รอจนกว่าสัญญาณจากขานี้เป็น 1
5. Set ขา OE (Output Enable) เพื่อให้สัญญาณเอาต์พุตที่ได้ส่งเข้ามายังวงจรรายนอกให้ port B ของ 8031 ต่อไป



รูปที่ 3.10 แสดงวงจรแปลงข้อมูลจากอนาล็อกเป็นดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

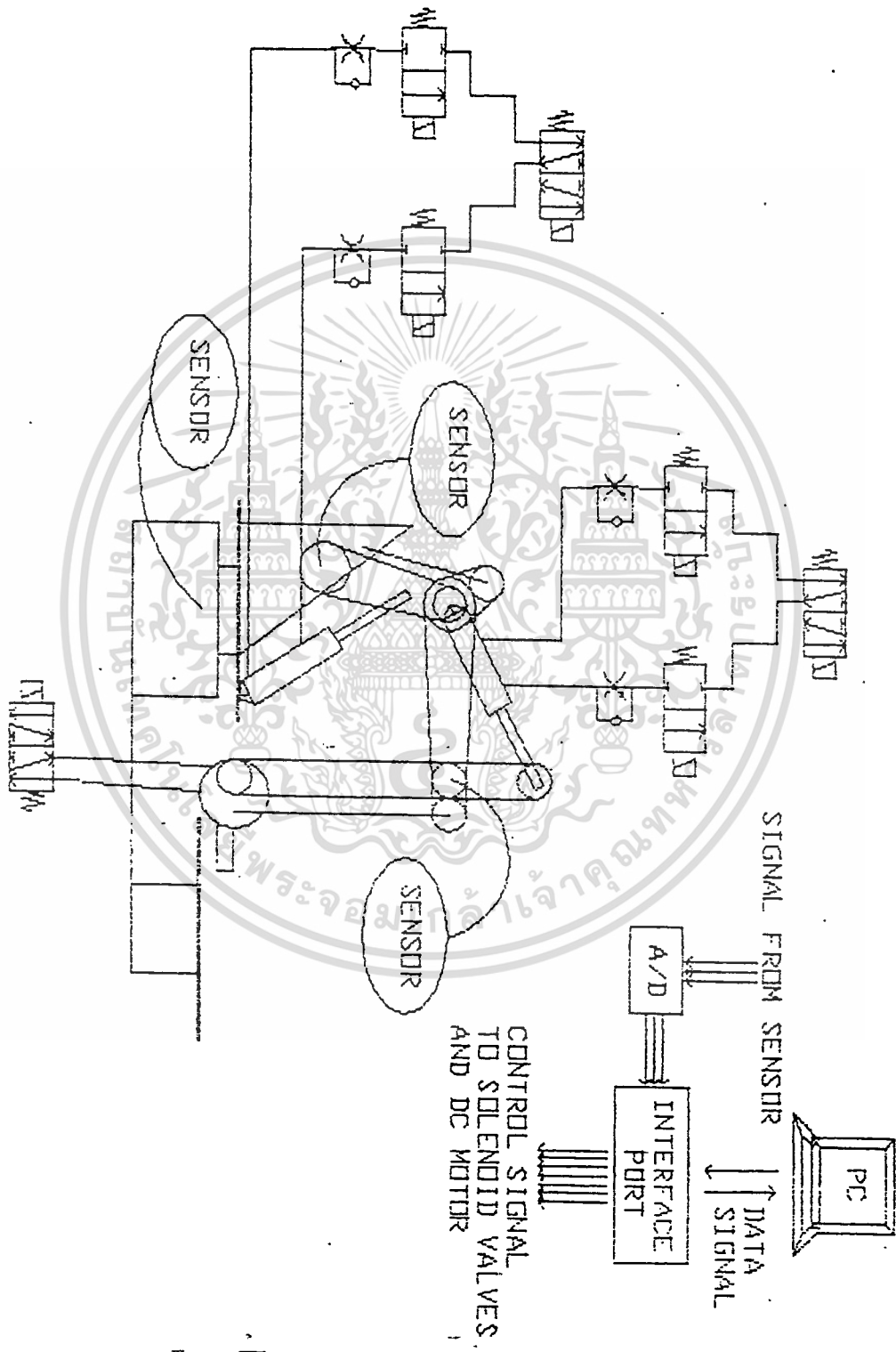
บทที่ 4 หลักการออกแบบและโปรแกรมของการทำงาน

4.1 แนวคิดของการทำงานของแขนกล

หลักการทำงานของแขนกลโดยใช้คอมพิวเตอร์เป็นตัวควบคุมการทำงาน เมื่อต้องการบังคับแขนก็บังคับคำสั่งที่หน้าจอตามที่โปรแกรมไว้แล้ว คอมพิวเตอร์ก็จะทำงานตามคำสั่งส่งข้อมูลผ่านพอร์ตอนุกรมส่งเข้าบอร์ด CP-JR 31 ซึ่งมีโพสเซลเซอร์หลักคือ 8031 เมื่อข้อมูลส่งสู่บอร์ด CP-JR 31 ก็จะถูกทำงานเปลี่ยนข้อมูลที่ได้จากพอร์ตอนุกรมของคอมพิวเตอร์ไปเป็นสัญญาณไฟฟ้าส่งไปยังวงจรขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงและโซลินอยด์

เมื่อขยับไปยังตำแหน่งที่ต้องการแล้วจะต้องมีการตรวจว่า ถูกต้องตามที่ต้องการหรือไม่ โดยมีอุปกรณ์ตรวจจับ คือ ส่วนของกระบอกลูกสูบจะใช้โพเทนชิโอมิเตอร์หรือความต้านทานปรับค่าได้อยู่ ส่วนมอเตอร์จะใช้เอนโค้ดเดอร์ที่อยู่ในตัวมอเตอร์ผู้พัลส์มา ซึ่งส่วนของโพเทนชิโอมิเตอร์จะมี 2 ตัวคือที่กระบอกลูกสูบที่ 1 และ 2 ส่วนของมือจับไม่มี ค่าแรงดันไฟฟ้าของโพเทนชิโอมิเตอร์จะเปลี่ยนตามระยะเคลื่อนของกระบอกลูกสูบแล้วจะส่งข้อมูลมายังวงจรที่ทำหน้าที่เปลี่ยนจากอนาลอกเป็นดิจิตอล ข้อมูลดิจิตอลที่ได้จะส่งเข้าสู่บอร์ด JR-CP 31 พร้อมกับสัญญาณพัลส์ที่ได้จากมอเตอร์ แล้วบอร์ด JR-CP 31 ก็จะทำงานเปลี่ยนค่าต่างเป็นข้อมูลทางดิจิตอลส่งไปยังคอมพิวเตอร์เพื่อประมวลผลให้ได้ตำแหน่งที่แม่นยำที่สุด ซึ่งทั้งหมดนี้สามารถดูได้ตามรูปที่

4.1



รูปที่ 4.1 แสดงการทำงานของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 แนวคิดของการทำงานของโปรแกรมบนคอมพิวเตอร์

โครงการนี้จะศึกษาการทำงานของแขนกลซึ่งจะควบคุมการทำงานโดยการควบคุมจะทำเป็น 3 แบบใหญ่คือ

1. การควบคุมโดยตรง (Manual control) จะบังคับแขนกลปรับเปลี่ยนระยะมุมและตำแหน่งโดยบังคับจากคีย์บอร์ดโดยตรง ซึ่งระยะตำแหน่งจะกะได้ด้วยสายตาและจะแสดงกราฟฟิการบังคับ เมื่อเรากดคีย์บอร์ดตามที่กำหนดไว้ แขนก็จะขยับและจะอ่านค่าตำแหน่งนั้นไปแสดงที่กราฟฟิการ การสั่งแขนให้จับหรือปล่อยโดยการกด H สำหรับการจับและ R สำหรับการปล่อย และยกเลิกการทำงานโดยการกด S ซึ่งการทำงานอย่างคร่าว ๆ แสดงโดย Flow chart รูปที่ 4.3

2. การควบคุมแบบปิด (Close loop control) เมื่อการส่งสัญญาณ input สั่งให้อุปกรณ์ทำงาน จะมีตัวคอยตรวจจับสัญญาณ output ที่ออกไป แล้วส่งสัญญาณป้อนกลับ (feedback) กลับมายังทาง input เปรียบเทียบกับค่าที่ตั้งเอาไว้แล้วส่งสัญญาณ input ไปใหม่ ซึ่ง input ใหม่นี้จะแก้ค่าความผิดพลาดที่เกิดขึ้นจนกว่าค่าสัญญาณ feedback ที่ส่งมาเปรียบเทียบกับค่า input แล้วเท่ากันหรือเท่ากับค่า error ที่ยอมรับได้

ตัวจับสัญญาณนี้ จะเป็นตัวแปลงสัญญาณทางกายภาพ เช่น ความร้อน , ระยะทาง , ความดัน , ความเร็ว , ความแรง ฯลฯ ให้เป็นสัญญาณทางไฟฟ้า โดยอาจจะเป็นสัญญาณอะนาลอกหรือดิจิตอลก็ได้ ตัวแปลงสัญญาณนี้เรียกอีกอย่างหนึ่งว่า Transducer และ Sensor ซึ่งในโครงการนี้จะใช้ตัวแปลงสัญญาณแบบระยะทางอย่างเดียว

การควบคุมแบบปิดในโครงการนี้จะแบ่งเป็น 2 ชนิดด้วยกันคือ

2.1 การควบคุมแบบอัตโนมัติ (Auto control) การควบคุมที่เราใส่ตำแหน่งที่เราต้องการลงไป แล้วแขนกลก็จะขยับไปยังตำแหน่งนั้นโดยอัตโนมัติ รูปที่ 4.4 เป็น Flow chart แสดงการทำงาน ซึ่งอธิบายขั้นตอนการทำงานคร่าว ๆ คือ

1. อ่านตำแหน่งปัจจุบัน
2. อ่านตำแหน่งที่ต้องการ
3. นำผลต่างของทั้ง 2 ตำแหน่งมาคำนวณ
4. เมื่อได้ผลต่างแล้วถ้า

ผลต่างมากกว่า 50 หน่วย ขยับไป 50 จุด

ผลต่างมากกว่า 20 หน่วย ขยับไป 20 จุด

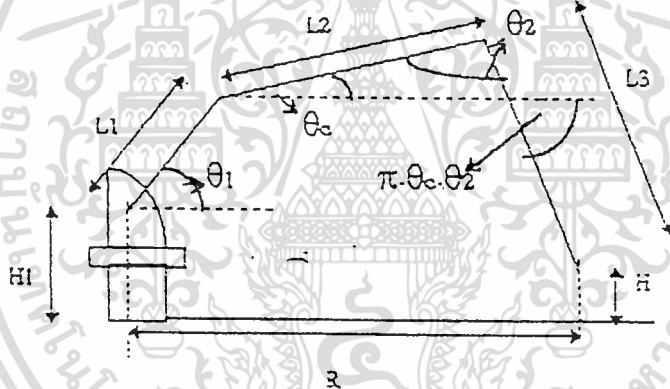
ผลต่างมากกว่า 10 หน่วย ขยับไป 10 จุด

เอกสารนี้เป็นเอกสารที่ส่งผลต่างมากกว่า 5 หน่วย ขยับไป 5 จุด ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ จึงอ่านตำแหน่งปัจจุบัน แล้วมีมาหาผลต่างกับตำแหน่งที่ต้องการอีกครั้งที่มีกรนำไปใช้

6. ถ้าผลต่าง $\neq 0$ ก็จะทำงานในข้อที่ 4 และ 5 ใหม่
- ถ้าผลต่าง = 0 ก็จะเสร็จการทำงาน
7. ต้องการใส่ค่าตำแหน่งใหม่หรือไม่ ถ้าไม่ก็จบ

2.2 การควบคุมแบบอ่านจากไฟล์ (Read file to control) ก็คือการควบคุมแขนกลโดยใช้อ่านข้อมูลของตำแหน่งจากไฟล์ข้อมูลที่บ้านทึกไว้ ซึ่งข้อดีของการควบคุมแบบอ่านจากไฟล์ คือสามารถกำหนดตำแหน่งใหม่มันเคลื่อนที่ได้ทีละหลายตำแหน่ง ส่วนการทำงานของการทำงานแบบอ่านจากไฟล์ที่จะทำงานให้แขนกลขยับนั้นจะเหมือนกับการควบคุมแบบอัตโนมัติ ดูการทำงานได้จาก Flow chart รูปที่ 4.5

รูป Flow chart รูปที่ 4.2 เป็นลักษณะการทำงานของเมนูไว้สำหรับเลือกว่าจะให้ทำงานแบบใด ส่วนสมการในการคำนวณต่าง ๆ ในโปรแกรมจะอธิบายดังนี้

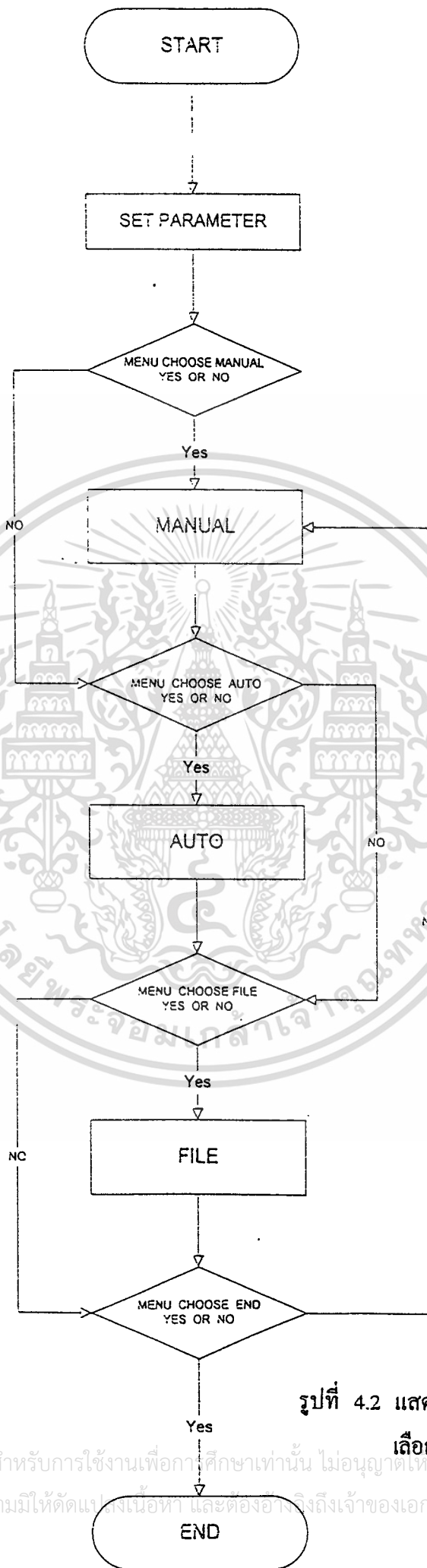


CAL คือสมการเปลี่ยนค่าจาก รัศมี กับ ความสูง เป็น มุมของข้อต่อที่ 1 ทำกับแนวระดับ (θ_1) และ มุมของข้อต่อที่ 2 กับ ข้อต่อที่ 3 (θ_2) ซึ่งวิธีการคำนวณได้จากบทที่ 2

CAL2 คือสมการเปลี่ยนค่าจาก θ_1 , θ_2 และ a_n (มุมของฐาน) เป็นข้อมูลทางดิจิทัล ซึ่งวิธีการคำนวณได้จากบทที่ 3.3

INV_CAL1 คือสมการเปลี่ยนค่าจาก θ_1 และ θ_2 กลับเป็นค่า รัศมี และ ค่าความสูง โดย แลกค่าความยาวไปเป็นแนวระดับและแนวตั้ง

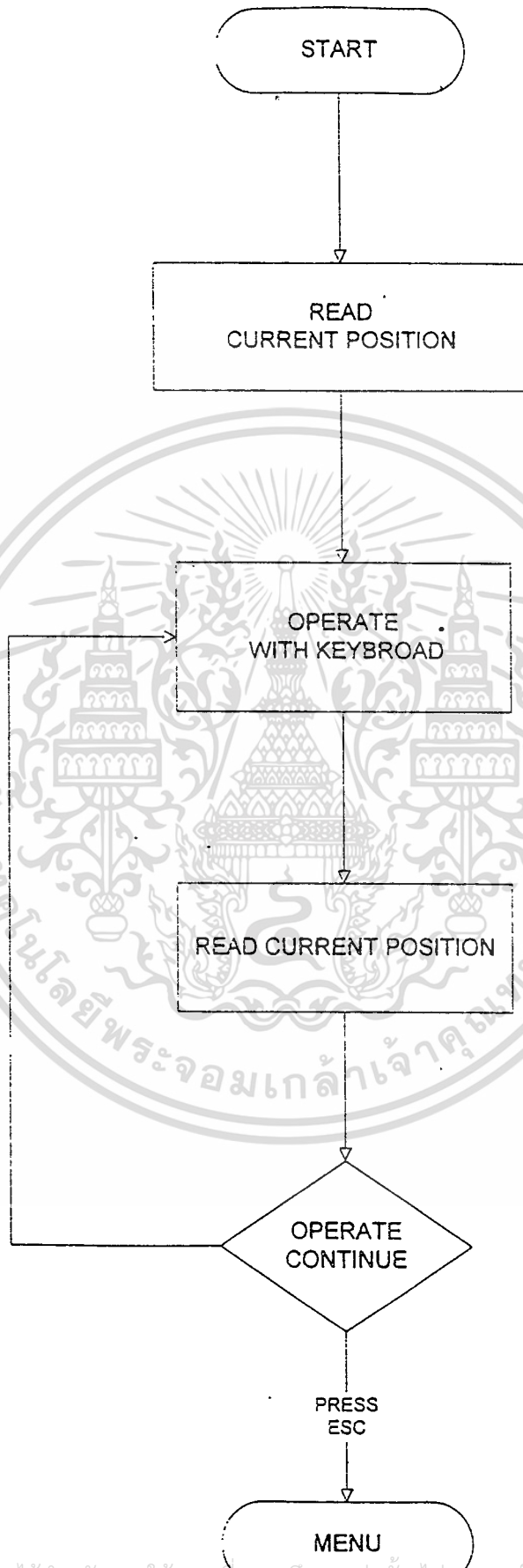
INV_CAL2 คือสมการเปลี่ยนค่าจากข้อมูลทางดิจิทัลไปเป็นข้อมูลของ θ_1 , θ_2 และ a_n



รูปที่ 4.2 แสดงการทำงานของกร

เลือกลักษณะการควบคุม

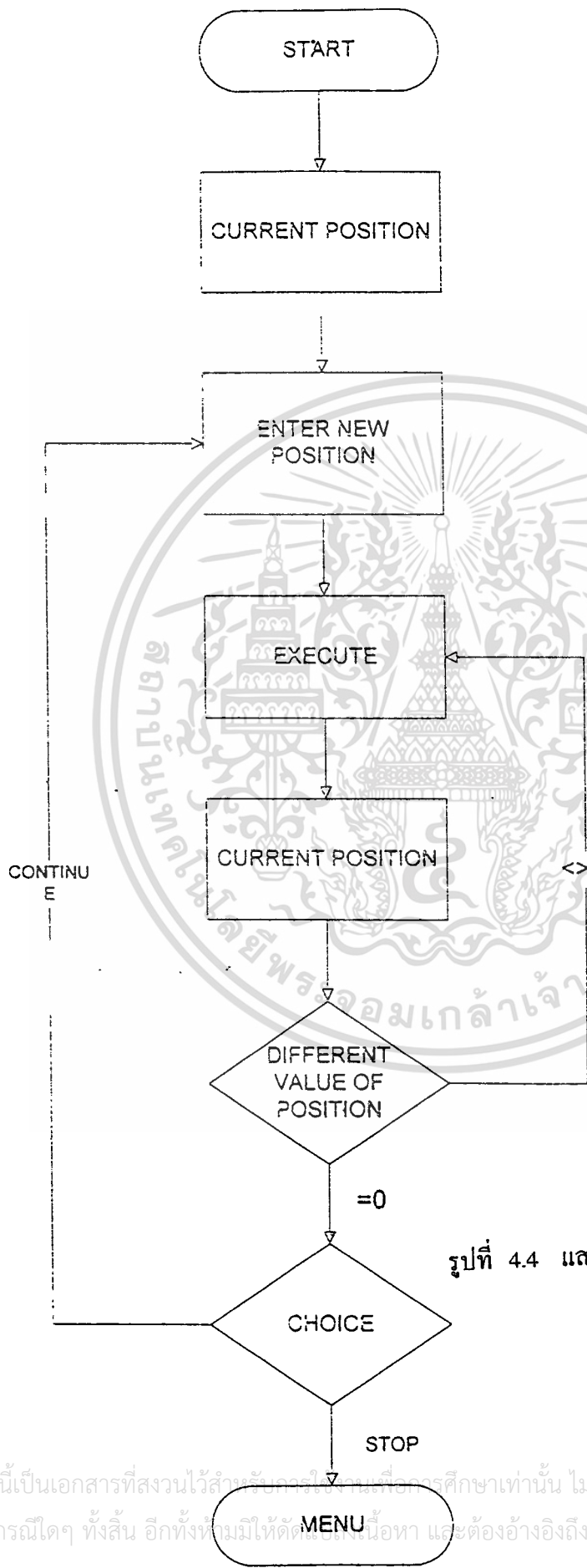
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

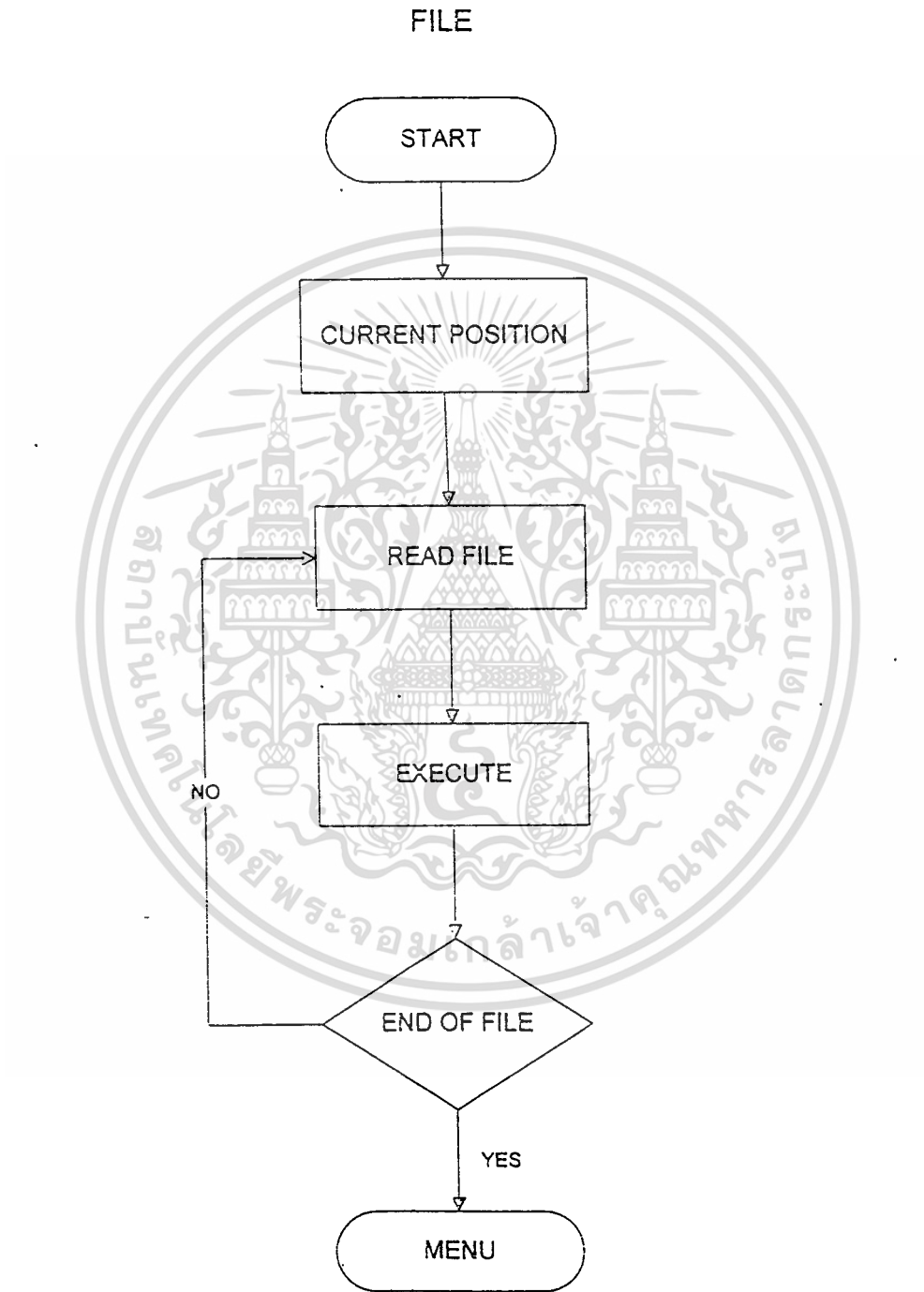
รูปที่ 4.3 แสดงการทำงานของการทำงานควบคุมโดย Manual

AUTO



รูปที่ 4.4 แสดงการทำงานของ การควบคุมโดย Auto

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดต่อหรือแก้ไขเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 แสดงการทำงานของการทำงานควบคุมโดยเก็บค่าไว้ใน File

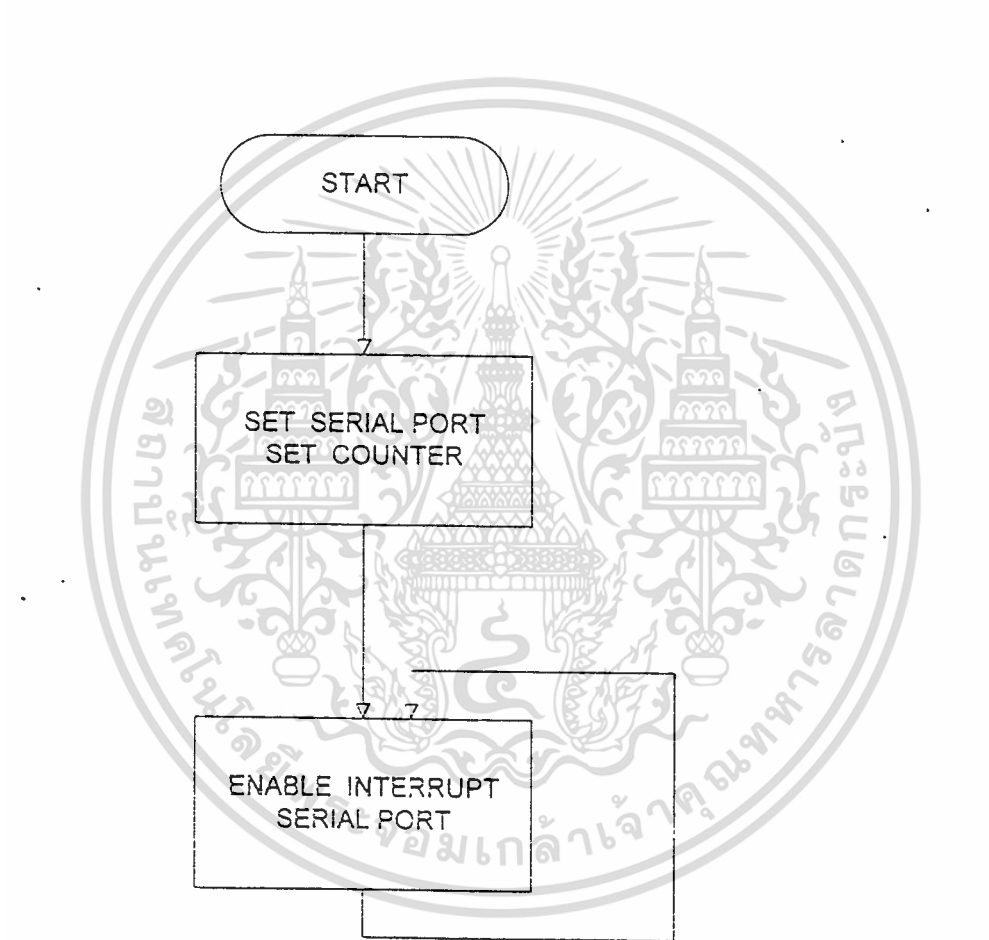
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 แนวคิดของการทำงานโปรแกรมของ 8031

กำหนดการทำงานของ 8031 ให้รับส่งข้อมูลอนุกรมได้ กำหนดโหมดโหมดเมอร์ 1 ให้ใช้กำหนดความเร็วในการรับส่งข้อมูลอนุกรม ให้เคาท์เตอร์ 0 เป็นตัวนับสัญญาณพัลส์จากมอเตอร์ กำหนดโหมดการทำงานของพอร์ต 8155 ให้พอร์ต A ใช้ส่งข้อมูลออกไปควบคุมมอเตอร์และโซลินอยด์วาล์ว และ พอร์ต B ใช้รับข้อมูลจากวงจรแปลงข้อมูลจากดิจิตอลเป็นอนาลอก เคลียร์ค่าในเคาท์เตอร์และเคลียร์ค่าพอร์ต A เซ็ตให้ 8031 สามารถอินเตอร์รัปต์ เมื่อมีการรับส่งข้อมูลอนุกรมได้ วนลูปที่การเซตอินเตอร์รัปต์เพื่อรอให้มีการอินเตอร์รัปต์เกิดขึ้น

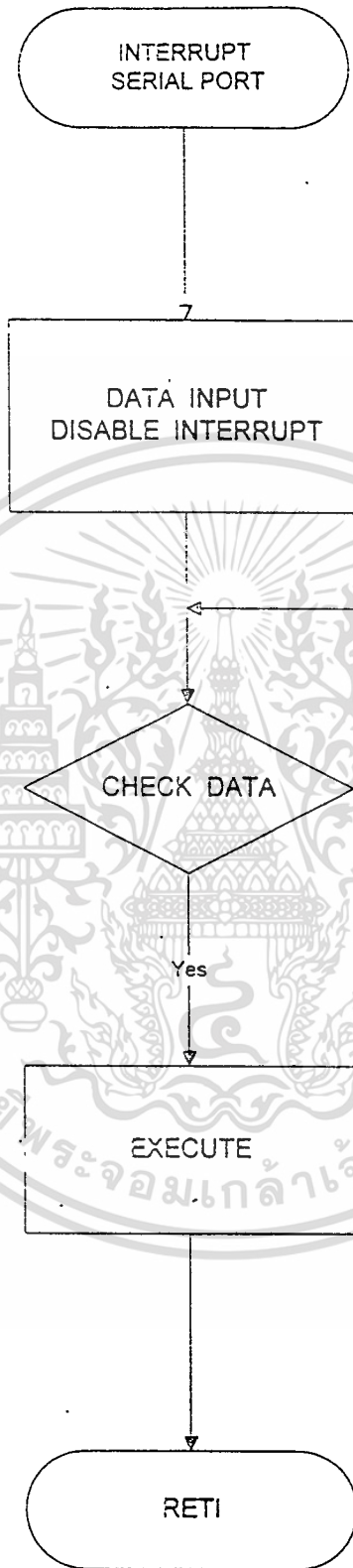
เมื่อมีการอินเตอร์รัปต์จากการรับข้อมูลเข้าสู่โปรแกรมส่วนอินเตอร์รัปต์ กำหนดให้เกิดการอินเตอร์รัปต์ไม่ได้ตรวจสอบข้อมูลที่รับเข้ามา ถ้าตรงรหัสก็ทำงานตามรหัสที่ตั้งไว้ ถ้าไม่ถูกต้องก็ตรวจสอบรหัสถัดไป เมื่อทำงานตามคำสั่งแล้ว ก็ส่งรหัส 26 h ไปที่พอร์ตอนุกรมเพื่อบอกถึงการสิ้นสุดการส่งข้อมูลและกลับไปโปรแกรมหลัก ซึ่งสามารถดูได้จากรูปที่ 1 และ 2 รหัสและการทำงานของ 8031

รหัสที่ส่งมาจากคอมพิวเตอร์	การทำงานของ 8031
61 h	ส่งข้อมูลจากเคาท์เตอร์กลับสู่คอมพิวเตอร์
62 h	ส่งข้อมูลที่ได้จากการแปลงสัญญาณจากอนาลอกเป็นดิจิตอลของกระบอกสูบตัวที่ 1
63 h	ส่งข้อมูลที่ได้จากการแปลงสัญญาณจากอนาลอกเป็นดิจิตอลของกระบอกสูบตัวที่ 2
64 h	ส่งค่าจากลิมิตสวิทช์ของมอเตอร์
65 h	เคลียร์ค่าในเคาท์เตอร์
66 h	หยุดการส่งสัญญาณไปมอเตอร์
67 h	หยุดส่งสัญญาณไปกระบอกสูบตัวที่ 1
68 h	หยุดส่งสัญญาณไปกระบอกสูบตัวที่ 2
69 h	หยุดส่งสัญญาณไปแขนจับ
6A h	มอเตอร์หมุนไปทางขวา
6B h	มอเตอร์หมุนไปทางซ้าย
6C h	กระบอกสูบตัวที่ 1 เคลื่อนที่ออก
6D h	กระบอกสูบตัวที่ 1 เคลื่อนที่เข้า
6E h	กระบอกสูบตัวที่ 2 เคลื่อนที่ออก
6F h	กระบอกสูบตัวที่ 2 เคลื่อนที่เข้า
70 h	แขนจับขยับเข้า



รูปที่ 4.6 แสดงการทำงานของการทำงานของการควบคุมการส่งค่าเข้าออกของบอร์ด CP - JR 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แสดงรูทีนของการรับค่าข้อมูลโดยการอินเทอร์รัปต์

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น เมื่อผู้ใช้งานใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 โปรแกรมควบคุม

โปรแกรมมีอยู่ 2 โปรแกรมซึ่งสามารถดูได้ในภาคผนวก คือ

1. โปรแกรมที่เขียนด้วยภาษา C เพื่อใช้ในการควบคุมการทำงานของแขนกลผ่านเครื่องคอมพิวเตอร์
2. โปรแกรมที่เขียนไว้ใน EPROM ที่ใช้ในการอินเทอร์รัปต์ เพื่อให้ บอร์ด CP-JR 31 สามารถรับส่งข้อมูลผ่าน serial port (com 2) ของคอมพิวเตอร์ได้



บทที่ 5

การทดลองควบคุมแขนกลด้วยคอมพิวเตอร์

การประกอบแขนกลและหน่วยควบคุมต่างๆเพื่อใช้ในการทดลอง

1. นำเอาโครงแขนกลที่ประกอบเสร็จแล้วติดเข้ากับโต๊ะที่จะใช้ในการทดลอง
2. นำเอาคู่เฟืองตรงขนาดอัตราทด 1:5 มาประกอบกับตัวแขนกล
 - เฟืองตัวเล็กจะทำหน้าที่เป็นตัวขับซึ่งจะติดไว้ที่ตัวแกนของ DC motor
 - เฟืองตัวใหญ่จะทำหน้าที่เป็นตัวตามซึ่งจะติดไว้ที่แกนเพลาขนาด 1 นิ้วของโครงแขนกล
3. ติด DC motor พร้อมเฟืองเข้ากับฐานยึดให้เรียบร้อยในชั้นกลางของโต๊ะทดลอง
4. นำเอาระบบนิวแมติกที่ใช้ในการควบคุมมาประกอบเข้ากับ บั๊มลมพร้อมด้วยชุดปรับความดันให้เรียบร้อย แล้วนำไปประกอบเข้ากับแขนกลให้เรียบร้อย โดยจะวางไว้ในชั้นกลางของโต๊ะทดลอง
5. นำเอาวงจรควบคุม คือ บอร์ด CP-JR 31 , วงจรขับโซลินอยด์วาล์ว , วงจรขับมอเตอร์ , วงจรสร้างไฟเลี้ยงระบบ , วงจร ADC และอุปกรณ์ sensor ต่างๆ มาติดตั้งให้เรียบร้อย
 - โทเทินชิโอมิเตอร์จะติดอยู่ที่กระบอกลูกสูบกระบอกละ 1 ตัว
 - Limit switch จะติดอยู่ที่ฐานหมุน 2 ตัว
6. นำคอมพิวเตอร์พร้อมด้วยแผ่นโปรแกรมควบคุม มาเชื่อมต่อเข้ากับ บอร์ด CP-JR 31

การทดลอง**จุดประสงค์**

เพื่อศึกษาผลต่างๆที่เกิดขึ้น ในการควบคุมแขนกลนิวแมติกด้วยคอมพิวเตอร์ในโหมดต่างๆ ของโปรแกรมควบคุมที่เขียนขึ้น

วิธีทดลอง**ตอนที่ 1 การทดลองการหมุนของมอเตอร์**

1. ทำการทดลองสั่งคำสั่งให้มอเตอร์หมุนไปแบบต่อเนื่อง โดยทำการเพิ่มค่ามุมไปครั้งละ 15 องศา แล้วทำการบันทึกผลไว้ เพื่อเปรียบเทียบผลกับมุมที่สั่งเข้าไป
2. ทำการทดลองสั่งคำสั่งให้มอเตอร์หมุนไปแบบไม่ต่อเนื่อง โดยให้กลับมาที่ตำแหน่งศูนย์ องศาแล้วจึงค่อยสั่งให้มอเตอร์หมุนไปตามมุมที่ต้องการ โดยทำการเพิ่มค่ามุมไปครั้งละ 15 องศา แล้วทำการบันทึกผลไว้ เพื่อเปรียบเทียบผลกับมุมที่สั่งเข้าไป

ตอนที่ 2 ทำการทดลองสั่งให้แขนกลเคลื่อนที่ไปด้วยโปรแกรมในโหมดต่างๆ

1. ทำการใช้โปรแกรมในโหมด MANUAL เพื่อควบคุมการทำงานของแขนกล
 - กค A เพื่อให้กระบอกสูบตัวที่ 1 เคลื่อนที่ขึ้น
 - กค B เพื่อให้กระบอกสูบตัวที่ 1 เคลื่อนที่ลง
 - กค D เพื่อให้กระบอกสูบตัวที่ 2 เคลื่อนที่ขึ้น
 - กค C เพื่อให้กระบอกสูบตัวที่ 2 เคลื่อนที่ลง
 - กค O เพื่อให้มอเตอร์หมุนไปทางซ้าย(ทวนเข็มนาฬิกา)
 - กค P เพื่อให้มอเตอร์หมุนไปทางขวา(ตามเข็มนาฬิกา)
 - กค R เพื่อให้กระบอกสูบตัวที่ 3 (มือจับ) ยืดออก
 - กค H เพื่อให้กระบอกสูบตัวที่ 3 (มือจับ) หดเข้า
 ทำการบันทึกผลการทดลองไว้
2. ทำการทดลองในโหมด AUTO บันทึกผลการทดลอง
3. ทำการทดลองในโหมด FILE บันทึกผลการทดลอง

ตารางบันทึกผล

ตอนที่ 1

ตารางสั่งมอเตอร์ให้หมุนแบบต่อเนื่องโดยเพิ่มค่ามุมทีละ 15 องศา

มุมที่ให้มอเตอร์หมุน (องศา)	ค่าที่อ่านจากcomputer (องศา)	ค่าที่วัดได้จริง (องศา)	ERROR (องศา)
15	14.793103	16	+1
30	29.896553	32	+2
45	44.689659	48	+3
60	59.793106	65	+5
75	74.482765	82	+7
90	89.586220	98	+8
105	114.482773	115	+10
120	119.482765	134	+11
135	133.482773	147	+12
150	149.689682	163	+13
165	164.689697	178	+13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางสั่งให้มอเตอร์หมุนโดยให้หมุนเริ่มจากจุดเริ่มต้น

มุมที่ให้มอเตอร์หมุน (องศา)	ค่าที่อ่านจากcomputer (องศา)	ค่าที่วัดได้จริง (องศา)	ERROR (องศา)
15	14.689654	16	+1
30	29.689655	32	+2
45	45.000000	47	+2
60	59.896549	62	+2
75	75.000000	75	0
90	89.793098	91	+1
105	104.586205	104	- 1
120	120.103455	119	- 1
135	134.689667	134	- 1
150	150.206894	147	- 3
165	165.137939	163	- 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตอนที่ 2

ผลการทดลองด้วยโปรแกรม MANUAL (ความดันที่ใช้ในการอัดลม 4.5 bar)

กระบอกสูบตัวที่ 1

ครั้งที่ของการกดปุ่ม	ผลต่างของระยะที่กระบอกสูบตัวที่ 1 เลื่อนออก (cm.)
1	0.6
2	1.1
3	1.5
4	1.7
5	2.1

กระบอกสูบตัวที่ 1

ครั้งที่ของการกดปุ่ม	ผลต่างของระยะที่กระบอกสูบตัวที่ 1 หดเข้า (cm.)
1	1.5
2	2.0
3	2.5
4	1.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบอกสูบตัวที่ 2

ครั้งที่ของการกดปุ่ม	ผลต่างของระยะที่กระบอกสูบตัวที่ 2 เลื่อนออก (cm.)
1	1.0
2	0.8
3	0.6
4	0.5
5	0.4
6	0.3
7	0.3
8	0.3
9	0.3
10	0.3
11	0.3
12	0.3
13	0.2
14	0.2
15	0.2
16	0.2
17	0.1
18	0.1
19	0.1
20	0.1
21	0.1
22	0.1
23	0.1
24	0.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบอกสูบตัวที่ 2

ครั้งที่ของการกดปุ่ม	ผลต่างของระยะที่กระบอกสูบตัวที่ 2 หดเข้า (cm.)
1	2.0
2	1.5
3	1.2
4	0.8
5	0.7
6	0.4
7	0.4

กระบอกสูบตัวที่ 3 เมื่อสั่งให้ยืดยึดออกก็สามารถยืดยึดออกได้และเมื่อสั่งให้หดเข้าก็สามารถหด
กลับที่เดิมได้

มอเตอร์ที่ฐาน เมื่อสั่งให้มอเตอร์หมุนซ้ายและหมุนขวา มอเตอร์จะหมุนไปครั้งละ 15 องศา

ผลการทดลองในโหมด AUTO และ การทดลองในโหมด FILE

ยังไม่สามารถทำการทดลองได้เนื่องจากในชุดทดลองนั้นเกิดเหตุขัดข้องในส่วนของอุปกรณ์
ที่ไขแปลงสัญญาณอนาล็อกเป็นดิจิตอล ซึ่งเป็นส่วนที่สำคัญมากในการควบคุมในโหมดทั้ง 2 นี้

บทที่ 6

สรุปผลการทดลอง

สรุปผลการทดลองและวิจารณ์

จากผลการทดลองควบคุมมอเตอร์ในตอนที่ 1 นั้นจะเห็นว่าเมื่อทำการควบคุมแบบต่อเนื่องจะมีค่าผิดพลาดสะสมเพิ่มขึ้นเรื่อยๆ เนื่องจากว่าในการหมุนของมอเตอร์แต่ละครั้งนั้นจะมีค่าผิดพลาดเกิดขึ้นอยู่แล้ว ดังนั้นเมื่อทำการสั่งให้มอเตอร์หมุนต่อไปในแต่ละครั้งก็จะยิ่งทำให้ค่าผิดพลาดที่เกิดขึ้นจะเริ่มสะสมเพิ่มมากขึ้นเรื่อยๆนั่นเอง ส่วนการควบคุมแบบให้กลับมาเริ่มหมุนจากจุดเริ่มต้นนั้นจะไม่ทำให้เกิดค่าผิดพลาดสะสมขึ้นดังนั้นค่าผิดพลาดที่เกิดขึ้นจึงเกิดขึ้นจากแต่ละขั้นตนเอง

จากผลการทดลองควบคุมแขนกลนิวเมติกในโหมด MANUAL ในตอนที่ 2 นั้นส่วนที่สำคัญคือการบังคับกระบอกลูกสูบตัวที่ 1 และ 2 ซึ่งถ้าพิจารณาจากการยืดออกของกระบอกลูกสูบตัวที่ 1 จะเห็นว่าในช่วงแรกผลต่างของระยะที่กระบอกลูกสูบยืดออก จะน้อยกว่าผลต่างของระยะที่กระบอกลูกสูบยืดออกในช่วงหลังๆ เพราะว่าในช่วงแรกนั้นจะพบวาระยะตามแนวราบจากจุดหมุนที่ฐานถึงปลายมือจับจะมีค่ามากจะทำให้เกิดแรงกดที่กระบอกลูกสูบตัวที่ 1 มากทำให้ระยะยืดออกน้อย ต่อจากนั้นในช่วงหลังระยะตามแนวราบจากจุดหมุนที่ฐานถึงปลายมือจับจะมีค่าน้อยลงทำให้มีแรงกดที่กระบอกลูกสูบตัวที่ 1 ลดลงทำให้ระยะยืดออกมากขึ้นเรื่อยๆ (เมื่อมีแรงกดน้อยลงอากาศก็จะสามารถขยายตัวได้มากขึ้นนั่นเอง) และถ้าพิจารณาจากการหดเข้าของกระบอกลูกสูบตัวที่ 1 ก็จะเห็นว่าในช่วงแรกผลต่างของระยะที่กระบอกลูกสูบหดเข้า จะน้อยกว่าผลต่างของระยะที่กระบอกลูกสูบหดเข้าในช่วงหลังๆ ก็ด้วยเหตุผลเดียวกันกับการยืดออกนั่นเอง ส่วนผลที่เกิดขึ้นในกระบอกลูกสูบตัวที่ 2 นั้นจะเห็นว่าในการยืดออกนั้นในช่วงแรกผลต่างของการยืดออกจะมากกว่าในช่วงหลังก็ด้วยเหตุผลคล้ายๆของเดิมคือระยะตามแนวราบจากจุดหมุนของชิ้นส่วนแขนกลนี้ถึงปลายมือจับจะมีค่ามากในช่วงแรก แต่ในกรณีนี้จะทำให้เกิดแรงดึงที่กระบอกลูกสูบตัวที่ 2 ซึ่งในช่วงหลังๆระยะตามแนวราบจากจุดหมุนของชิ้นส่วนของแขนกลนี้ถึงปลายมือจับจะมีค่าน้อยลงทำให้เกิดแรงดึงน้อยลงด้วยเป็นผลทำให้ระยะยืดออกลดลงนั่นเอง ถ้าพิจารณาจากการหดเข้าของกระบอกลูกสูบตัวที่ 2 ก็จะเห็นว่าในช่วงแรกผลต่างของระยะที่กระบอกลูกสูบหดเข้า จะมากกว่าผลต่างของระยะที่กระบอกลูกสูบหดเข้าในช่วงหลังๆ ก็ด้วยเหตุผลเดียวกันกับการยืดออกนั่นเอง

ในการทดลองในโหมด AUTO และ FILE นั้นเนื่องจากเกิดปัญหาต่างๆที่ตัวแปลงสัญญาณอนาล็อกเป็นดิจิตอลจึงทำให้ไม่สามารถทำการทดลองได้ดังที่กล่าวมาแล้วในส่วนการบันทึกผล

ในส่วนที่ทำการทดลองไปแล้วนั้นค่าผิดพลาดต่างๆที่เกิดขึ้นนั้นจากการพิจารณาแล้วก็จะเห็นว่าส่วนใหญ่เกิดจากความผิดพลาดทางด้านอุปกรณ์ต่างๆ แรงเฉื่อยที่เกิดขึ้นขณะเคลื่อนที่ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่ขึ้นด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นที่ฐานหมุน ,การเขียน โปรแกรมที่ไม่รัดกุมพอที่จะจับสัญญาณป้อนกลับ ,การที่ลมอัดมีการอัด และขยายตัวได้ และในด้านอุปกรณ์การวัดผลการทดลองด้วยนั่นเอง

การปรับปรุงแก้ไขนั้น สิ่งแรกคือควรเปลี่ยนตัว Sensor จากโพเทนชิโอมิเตอร์ที่เป็น ความต้านทานปรับค่าได้เป็นตัว Sensor แบบออปติคัลเอน โค้ดเดอร์ เนื่องจากโพเทนชิโอมิเตอร์ ที่เป็นความต้านทานปรับค่านั้นหาซื้อระยะเวลา 10 cm เท่ากับ ระยะยี่ดของกระบอกสูบไม่ได้ ซึ่งมีปัญหามาถ้ามาใช้กับการกระบอกนิวแมติกที่โซลุ่มที่มีอัดตัวไม่แน่นอน จะทำงานโพเทนชิ โอมิเตอร์ที่เป็นความต้านทานปรับค่าเกิดการเสียหายได้เพราะมีขนาด 7 cm เท่านั้น Sensor แบบออปติคัลเอน โค้ดเดอร์นั้นสามารถทำได้ระยะไม่จำกัดโดยใช้โฟโตทรานซิสเตอร์ แต่จะต้อง มีวงจรนับพัลส์อาจยุ่งยากแต่ได้ผลดีกว่ามาก

สิ่งที่สองคือการทดลอง โปรแกรมที่เขียนกับแขนจริงๆ เพื่อปรับค่าให้เหมาะสมและพัฒนา โปรแกรมให้ดีขึ้น และควรทำในส่วนกราฟฟิก , Auto และ File ให้สมบูรณ์



ภาคผนวก

โปรแกรมภาษา C



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <stdio.h>
#include <conio.h>
#include <graphics.h>
#include <math.h>
#include <stdlib.h>
#include <string.h>
#include <bios.h>
#include <dos.h>

#define PORT_NO 1
#define id_1 1
#define id_2 2
#define id_an 3
#define timeS1 10
#define timeS2 10
#define timeM 20

float l1 = 20,l2 = 34,l3 = 30,hl = 18,lc1 = 14,ref1 = 13,lc2 = 9;
float rf[10],hf[10],anf[10];
float o1,o2,x1,x2;
float r,h,an,rt,ht,at,ri,hi,ani;
int choice,choice2,Nmax,start,stop;
int d_o1,d_o2,d_an,ipdataH,ipdataL,directM;
char destination[10],rt_str[10],ht_str[10],at_str[10];

float cal(float r,float h);          /** r,h => o1,o2  ***/
float cal2(float o1,float o2);       /** o1,o2 => x1,x2  ***/
int cal3(float x1,float x2,float an); /** x1,x2,an => d_o1,d_o2,d_an **/
float inv_cal1(float o1,float o2);   /** o1,o2 => r,h  ***/
float inv_cal2(int dt,int id);       /** dt,id => x1,x2  ***/
float inv_o1(float x1);              /** x1 => o1  ***/

```

```

float inv_o2(float x2);           /*** x2 => o2    ***/
float inv_an(int ipdataL,int ipdataH); /** ipdataL,ipdataH => angle **/
void f_str(float flt);           /*** float => string ***/
void f_strha(float r,float h,float an); /** float r,h,an => string **/
void read();                     /*** read auto's data ***/
void read_exe(float x1,float x2,float an); /*** graphic ***/
int menu();                      /*** menu1 ***/
void text1();                    /*** menu1's text ***/
int menu2();                     /*** menu file ***/
void text2();                    /*** menu's text ***/
void read_file();               /** read data from file **/
void show();                    /*** graphic ***/
void show2();                   /*** auto graphic ***/
void pic_arrow();              /*** graphic ***/
void pic_exe(float x1,float x2,float o1,float o2,float r,float an);
                                /*** graphic ***/
void pic(float o1,float o2,float r,float an);
                                /*** show arm's pic ***/
int dexe(int diff,int id);      /*** before execute ***/
void S1exe(int diff,int start,int stop); /** solidnoid 1's execute **/
void S2exe(int diff,int start,int stop); /** solidnoid 2's execute **/
void Mexe(int diff,int start,int stop); /*** motor's execute ***/
int port(int code);            /** send data to serial port **/
float indata(int ani,int directM); /** receive all data from sensor **/
void start();                  /*** set home ***/
void end_auto(int ani);       /*** graphic ***/

main()
{
    int driver,mode,i,sw,errorcode;
    int out = 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int  data1,data2,data11,data22,d_an1,d_an2;
int  d_anL,d_anH,diff1,diff2,diff3;
char  chk,ckk,ch;
FILE  *file_ptr;

clrscr();
driver = DETECT;

do{
    menu();
    switch(choice)
    { case 0 : show();show2();pic_arrow();start();indata(0,1);
        f_str(ri); strcpy(rt_str,destination);
        f_str(hi); strcpy(ht_str,destination);
        f_str(at); strcpy(at_str,destination);
        setcolor(4);
        outtextxy(170,21,rt_str);outtextxy(170,37,ht_str);outtextxy(170,53,at_str);
        outtextxy(120,420,rt_str);outtextxy(120,436,ht_str);outtextxy(510,446,at_str);
        cal(ri,hi);cal2(o1,o2);cal3(x1,x2,ani);
        pic_exe(x1,x2,o1,o2,ri,ani);
        bar(270,460,370,470);outtextxy(295,462,"PROCESS");
        do{
            port(101);ch = getch();
            switch(ch)
            {
                /*****cylinder1 out*****/
                case 'a' : port(108);delay(50);port(103);directM=1;break;
                /*****cylinder1 in*****/
                case 'z' : port(109);delay(15);port(108);delay(25);port(103);
                    directM=1;break;
                /*****cylinder2 out*****/
                case 'd' : port(110);delay(50);port(111);delay(25);port(104);
                    directM=1;break;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้.

```

        /*****cylinder2 in*****/
        case 'c' : port(111);delay(50);port(110);delay(25);port(104);
                directM=1;break;

        /*****motor right*****/
        case 'p' : port(106);delay(200);port(102);
                directM=0;break;

        /*****motor left*****/
        case 'o' : port(107);delay(200);port(102);
                directM=1;break;

        case 's' : chk ='1';break;

        case 'h' : port(112);directM=1;
                setfillstyle(0,11);bar(242,460,258,470);
                setcolor(8);outtextxy(246,462,"H");break;
        case 'r' : port(69);directM=1;
                setfillstyle(0,11);bar(242,460,258,470);
                setcolor(8);outtextxy(246,462,"R");break;
    }
    indata(ani,directM);
    bar(170,17,227,60);bar(118,400,168,460);bar(508,426,565,466);
    cal(ri,hi);cal2(o1,o2);f_strha(ri,hi,ani);setcolor(4);
    outtextxy(170,21,rt_str);outtextxy(120,420,rt_str);
    outtextxy(170,37,ht_str);outtextxy(120,436,ht_str);
    outtextxy(170,53,at_str); outtextxy(510,446,at_str);
    pic_exe(x1,x2,o1,o2,ri,ani);
    bar(270,460,370,470);outtextxy(295,462,"PROCESS");
    }while(chk != '1');
    chk = '0';cleardevice();break;

case 1 : show();start();indata(0,1);
        f_str(ri); strcpy(rt_str,destination);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

f_str(hi); strcpy(ht_str,destination);
f_str(at); strcpy(at_str,destination);
setcolor(4);
outtextxy(170,21,rt_str);outtextxy(170,37,ht_str);outtextxy(170,53,at_str);
outtextxy(120,420,rt_str);outtextxy(120,436,ht_str);outtextxy(510,446,at_str);
cal(ri,hi);cal2(o1,o2);pic_exe(x1,x2,o1,o2,ri,ani);
bar(270,460,370,470);setcolor(5);outtextxy(279,462,"PRESS KEY");
cal3(x1,x2,ani);
data1 = d_o1;data2 = d_o2;d_an1 = d_an;
do{
do{
ch = getch();
switch (ch)
{ case 's':chk='1'; ckk='1'; break;
case 'c':
setfillstyle(0,11);bar(270,460,370,470);
setcolor(5);outtextxy(275,462,"NEW POSITION");
do{
read();cal(rt,ht);
cal2(o1,o2);read_exe(x1,x2,at);
} while(!((x1>=0)&&(x1<=7)&&(x2>=0)&&(x2<=7)
&&(at>=0)&&(at<=180)));
bar(270,460,370,470);outtextxy(295,462,"PROCESS");
cal3(x1,x2,at);
data11 = d_o1;data22 = d_o2;d_an2 = d_an;
diff1 = data11-data1;
diff2 = data22-data2;
diff3 = d_an2-d_an1;
do{
dexe(diff1,id_1);dexe(diff2,id_2);dexe(diff3,id_an);
port(98);port(98);port(98);port(98);data1 = ipdataL;

```

```

port(99);port(99);port(99);port(99);data2 = ipdataL;
port(97);d_anL = ipdataL;d_anH = ipdataH;port(101);
if (diff3 > 0)
    ani = ani+inv_an(d_anL,d_anH);
else ani = ani-inv_an(d_anL,d_anH);
inv_cal2(data11,id_1);inv_cal2(data22,id_2);
inv_o1(x1);inv_o2(x2);inv_cal1(o1,o2);ri = r;hi = h;
f_strha(ri,hi,ani);pic_exe(x1,x2,o1,o2,ri,ani);
setcolor(5);outtextxy(295,462,"PROCESS");
cal3(x1,x2,ani);d_anl = d_an;
diff1 = data11-data1;delay(500);
diff2 = data22-data2;delay(500);
diff3 = d_an2-d_anl ;delay(500);
port(100);sw = ipdataL;
}while(((abs(diff1) > 5 )||(abs(diff2) > 5 )
||abs(diff3) > 5 ))&&!(sw == 3));
end_auto(ani);port(101);cck='1';break;
case 'h':port(112);
setfillstyle(0,11);bar(270,460,370,470);bar(242,460,258,470);
setcolor(5);outtextxy(275,462,"NEW POSITION");
setcolor(8);outtextxy(246,462,"H");cck='1';break;
case 'r':port(69);
setfillstyle(0,11);bar(270,460,370,470);bar(242,460,258,470);
setcolor(5);outtextxy(275,462,"NEW POSITION");
setcolor(8);outtextxy(246,462,"R");cck='1';break;
}
}while(cck != '1');
}while(chk != '1');
chk = '0'; cck = '0';cleardevice();break;
case 2 : if ((file_ptr = fopen("arobot.dat", "rt")) != NULL)
    {fscanf(file_ptr,"%d \n",&Nmax);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(i=0;i<Nmax;i++)
fscanf(file_ptr,"%f %f %f \n",&rf[i],&hf[i],&anf[i]);
fclose(file_ptr);}
else {settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
outtextxy(150,200,"Read error");
delay(1500);exit('1');closegraph();}
menu2();
switch(choice2)
{ case 0 : read_file();break;
case 1 : break; }
break;
case 3 : out = 0;break;
}
}while(out);
closegraph();
return 0 ;
}

float cal(float r,float h)
{ float s1,c2;
float a,b,c,z1,z2,z3;

c = pow(l2-r,2) + pow(h1-h,2) - (pow(l3,2)-pow(l1,2));
z1 = pow(2*l1*(h1-h),2) + pow(2*l1*(l2-r),2);
z2 = 4*l1*(h1-h)*c;
z3 = pow(c,2) - pow(2*l1*(l2-r),2);
a = -z2/(2*z1);
b = (pow(z2,2) - 4*z1*z3);b = sqrt(b)/(2*z1);
s1 = a+b;
o1 = atan(s1/sqrt(1-pow(s1,2)));
c2 = ( (l1*cos(o1))+l2-r )/l3;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

o2 = atan(sqrt(1-pow(c2,2))/c2);
if(o2<0) o2 = M_PI + o2;
return 0;
}

```

```
float cal2(float o1,float o2)
```

```
{ float oc,ref2,cd1,cd2,a,b;
```

```

oc = atan(hl/ref1);
ref2 = sqrt( pow(hl,2)+pow(ref1,2) );
a = pow(lc1,2) + pow(ref2,2);
b = 2*lc1*ref2*cos(o1+oc);
cd1 = sqrt(a-b);
a = pow(lc2,2) + pow(l2,2);
b = 2*lc2*l2*cos(M_PI-o2);
cd2 = sqrt(a-b);
x1 = cd1-25.5 ; x2 = cd2-28.5;
return 0;
}

```

```
int cal3(float x1,float x2,float an)
```

```

{
setcolor(4); settextstyle(0,0,1);
if((x1>=0)&&(x1<=7))
d_o1 = ceil(x1*256/7);
else {bar(167,13,225,29);setcolor(5);outtextxy(180,21,"error");
delay(1500);exit('1');closegraph();}
if((x2>=0)&&(x2<=7))
d_o2 = ceil(x2*256/7);
else {bar(167,29,225,45);setcolor(5);outtextxy(170,37,"error");
delay(1500);exit('1');closegraph();}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((an>=0)&&(an<=180))
    d_an = an*340/36;
else { bar(167,45,225,63);setcolor(5);outtextxy(180,53,"error");
    delay(1500);exit('1');closegraph();}
return 0;
}

```

```
float inv_call(float o1,float o2)
```

```

{
    r = l1*cos(o1)+l2+l3*cos(M_PI-o2);
    h = h1+l1*sin(o1)-l3*sin(o2);
    return 0 ;
}

```

```
float inv_cal2(int dt,int id)
```

```

{
    switch (id){
        case 1 : x1 = dt*7.0/256;
                break;
        case 2 : x2 = dt*7.0/256;
                break;
    }
}

```

```
return 0;
```

```
}
```

```
float inv_o1(float x1)
```

```
{ float oc,ref2,a,cd1;
```

```
    cd1 = x1+25.5;
```

```
    oc = atan(h1/ref1);
```

```
    ref2 = sqrt( pow(h1,2)+pow(ref1,2) );
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    a = pow(lc1,2) + pow(ref2,2) - pow(cd1,2);
    a = a/(2*lc1*ref2);
    o1 = acos(a) - oc;
    return o1;
}

```

```

float inv_o2(float x2)
{ float oc,ref2,b,cd2;

```

```

    cd2 = x2+28.5;
    b = pow(lc2,2) + pow(l2,2) - pow(cd2,2);
    b = b/(2*lc2*l2);
    o2 = M_PI - acos(b);
    return o2;
}

```

```

float inv_an(int ipdataL,int ipdataH)

```

```

{ float kit,andif;

    kit = (ipdataH*256.0)+ipdataL;
    andif = kit*36/340;
    return andif;
}

```

```

void f_str(float flt)

```

```

{ float b;
  int a,c;
  char astr[10],cstr[10],point[] = ".",zero[] = "000";

```

```

    a = flt/1; b = flt-a; c = b*1000;

```

```

    itoa(a,astr,10); itoa(c,cstr,10);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

strcpy(destination, astr);
strcat(destination, point);
if ( c == 0 ) strcat(destination, zero);
else      strcat(destination, cstr);
}

```

```

void f_strha(float r, float h, float an)

```

```

{
    f_str(r); strcpy(rt_str, destination);
    f_str(h); strcpy(ht_str, destination);
    f_str(an); strcpy(at_str, destination);
}

```

```

void read()

```

```

{
    gotoxy(59,2); scanf("%f",&rt);
    gotoxy(59,3); scanf("%f",&ht);
    gotoxy(59,4); scanf("%f",&at);
    f_str(rt); strcpy(rt_str, destination);
    f_str(ht); strcpy(ht_str, destination);
    f_str(at); strcpy(at_str, destination);
}

```

```

void read_exe(float x1, float x2, float an)

```

```

{
    f_str(x1); outtextxy(20,90, destination);
    f_str(x2); outtextxy(20,100, destination);
    f_str(an); outtextxy(20,110, destination);
    if ((x1>=0)&&(x1<=7)&&(x2>=0)&&(x2<=7)
        &&(an>=0)&&(an<=180))
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    setfillstyle(0,11);bar(460,17,520,60);
    setcolor(4);
    outtextxy(465,21,rt_str);outtextxy(465,37,ht_str);outtextxy(465,53,at_str);
}
else
{
    setcolor(5);
    outtextxy(250,100,"CAN'T GO THIS POSITION PRESS ENTER KEY.");
    getch();
    setfillstyle(0,11);
    bar(460,17,520,60); bar(248,97,630,108); bar(270,460,370,470);
    setcolor(5);outtextxy(279,462,"NEW POSITION");
}
}

int menu()
{ int check = 1;
  int v_pos = 0;
  int y1,y2,key;

  setcolor(GREEN);
  setbkcolor(11);
  settextstyle(SANS_SERIF_FONT,HORIZ_DIR,4);
  line(100,134,450,134);line(100,135,450,135);line(100,136,450,136);
  line(100,140,450,140);line(100,141,450,141);line(100,142,450,142);
  line(0,0,640,0);line(0,0,0,480);line(0,480,640,480);line(640,480,640,0);
  line(1,1,639,1);line(1,1,1,479);line(1,479,639,479);line(639,479,639,1);
  line(2,2,638,2);line(2,2,2,478);line(2,478,638,478);line(638,478,638,2);
  line(3,3,637,3);line(3,3,3,477);line(3,477,637,477);line(637,477,637,3);
  line(4,4,636,4);line(4,4,4,476);line(4,476,636,476);line(636,476,636,4);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setbkcolor(11);setcolor(BLUE);
setfillstyle(1,LIGHTGREEN);bar(130,190,335,230);
text1();
while(check)
{
    key = g etch();
    if(key == 0)
    {
        switch(getch())
        {
            case 72 : v_pos = (v_pos > 0) ? --v_pos : 3;
                    break;
            case 80 : v_pos = (v_pos < 3) ? ++v_pos : 0;
                    break;
        }
    }
    else if (key == 13)
    {
        switch(v_pos){
            case 0 : choice = 0;check = 0;continue;
            case 1 : choice = 1;check = 0;continue;
            case 2 : choice = 2;check = 0;continue;
            case 3 : choice = 3;check = 0;exit('0');continue;}
    }

    setfillstyle(1,11);
    bar(20,145,620,460);
    y1 = 190 + 50*(v_pos); y2 = 230 + 50*(v_pos);
    setfillstyle(1,LIGHTGREEN);
    bar(130,y1,335,y2);
    text1();
}

setfillstyle(1,11);
cleardevice();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return(choice);
}

void text1()
{
    settextstyle(SANS_SERIF_FONT,HORIZ_DIR,4);
    outtextxy(100,100,"Choose method control");
    settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
    outtextxy(150,200,"1. Manual Control");
    outtextxy(150,250,"2. Auto Control");
    outtextxy(150,300,"3. File");
    outtextxy(150,350,"4. Quit");
}

int menu2()
{ int check = 1;
  int v_pos = 0;
  int y1,y2,key;

  setcolor(GREEN);
  line(0,0,640,0);line(0,0,0,480);line(0,480,640,480);line(640,480,640,0);
  line(1,1,639,1);line(1,1,1,479);line(1,479,639,479);line(639,479,639,1);
  line(2,2,638,2);line(2,2,2,478);line(2,478,638,478);line(638,478,638,2);
  line(3,3,637,3);line(3,3,3,477);line(3,477,637,477);line(637,477,637,3);
  line(4,4,636,4);line(4,4,4,476);line(4,476,636,476);line(636,476,636,4);
  setbkcolor(11);setcolor(BLUE);
  setfillstyle(1,LIGHTGREEN);bar(130,190,490,230);
  text2();
  while(check)
  {
      key = getch();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

,
,
if(key == 0)
,
{
    switch(getch())
        { case 72 : v_pos = (v_pos > 0) ? --v_pos : 1;
            break;
          case 80 : v_pos = (v_pos < 1) ? ++v_pos : 0;
            break;
        }
    }
else if (key == 13)
    { switch(v_pos){
        case 0 : choice2 = 0;check = 0;continue;
        case 1 : choice2 = 1;check = 0;continue;}
    }
setfillstyle(1,11);
bar(20,20,620,460);
y1 = 190 + 50*(v_pos); y2 = 230 + 50*(v_pos);
setfillstyle(1,LIGHTGREEN);
bar(130,y1,490,y2);
text2();
}
setfillstyle(1,11);
cleardevice();
return(choice2);
}

void text2()
{
    settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
    outtextxy(100,100,"Choose ");
    settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(150,200,"1. Position read from robot.dat");
outtextxy(150,250,"2. Enter position from keybroad");
}

void read_file()
{ int i;

  settxtstyle(SANS_SERIF_FONT,HORIZ_DIR,1);
  outtextxy(50,25,"Number of position : ");
  f_str(Nmax);outtextxy(250,25,destination);

  for(i=0;i<Nmax;i++)
  {settxtstyle(SANS_SERIF_FONT,HORIZ_DIR,1);
  outtextxy(50,50+25*i,"p . r =");
  outtextxy(200,50+25*i,"; h =");
  outtextxy(315,50+25*i,"; an =");
  itoa(i+1,destination,10);outtextxy(62,50+25*i,destination);
  settxtstyle(0,0,1);
  f_str(rf[i]);outtextxy(140,62+25*i,destination);
  f_str(hf[i]);outtextxy(250,62+25*i,destination);
  f_str(anf[i]);outtextxy(380,62+25*i,destination);
  }

  getch();
  cleardevice();
}

void show(void)
{
  setbkcolor(11);
  setcolor(2);

```

```

line(0,0,640,0);line(0,0,0,480);line(0,480,640,480);line(640,480,640,0);
line(1,1,639,1);line(1,1,1,479);line(1,479,639,479);line(639,479,639,1);
line(2,2,638,2);line(2,2,2,478);line(2,478,638,478);line(638,478,638,2);
line(3,3,637,3);line(3,3,3,477);line(3,477,637,477);line(637,477,637,3);
line(4,4,636,4);line(4,4,4,476);line(4,476,636,476);line(636,476,636,4);
line(320,0,320,80);line(0,80,640,80);line(0,95,640,95);

line(0,110,640,110);line(0,111,640,111);line(0,112,640,112);

line(320,110,320,410);line(260,410,260,480);line(380,410,380,480);
line(260,410,380,410);line(240,455,380,455);line(240,455,240,480);

setcolor(4);
settextstyle(0,0,1);
outtextxy(7,21,"CURRENT POSITION R=      cm.");
outtextxy(7,37,"      H=      cm.");
outtextxy(7,53,"      BASE ANGLE=      degree");
outtextxy(330,21,"GOTO POSITION R=      cm.");
outtextxy(330,37,"      H=      cm.");
outtextxy(330,53,"      BASE ANGLE=      degree");
outtextxy(10,85,"PRESS 'C' TO CONTINUE or PRESS 'S' TO");
outtextxy(315,85,"STOP or PRESS 'H'or'R' FOR HOLD/RELEASE");
outtextxy(120,100,"MESSAGE!!!!");
outtextxy(100,420,"R=      cm.");
outtextxy(100,436,"H=      cm.");
outtextxy(420,446,"BASE ANGLE=      degree");
outtextxy(275,413,"C : CONTINUE");outtextxy(275,423,"S : STOP");
outtextxy(275,433,"H : HOLD");outtextxy(275,443,"R : RELEASE");
setcolor(5);outtextxy(275,462,"NEW POSITION");
setcolor(8);outtextxy(246,462,"R");
}

```

```
void show2()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setfillstyle(0,11);
bar(330,21,600,70);bar(265,413,375,420);bar(9,85,630,92);
setcolor(4);settextstyle(0,0,1);
outtextxy(80,85," PRESS 'S' TO STOP or PRESS 'H'or'R' FOR HOLD/RELEASE");
setcolor(5);outtextxy(300,413,"MANUAL");
bar(270,460,370,470);outtextxy(295,462,"READY");
}

```

```

void pic_arrow()

```

```

{
setcolor(1);
setlinestyle(SOLID_LINE,0,1);
settextstyle(0,0,1);
outtextxy(330,15,"keyboard A move cylender1 up");
outtextxy(330,25,"      Z move cylender1 down");
outtextxy(330,35,"      D move cylender2 up");
outtextxy(330,45,"      C move cylender2 down");
outtextxy(330,55,"      O turn left");
outtextxy(330,65,"      P turn right");
}

```

```

void pic_exe(float x1,float x2,float o1,float o2,float r,float an)

```

```

{
if ((x1>=0)&&(x1<=7)&&(x2>=0)&&(x2<=7)&&(an>=0)&&(an<=180))
{
setfillstyle(0,11);
bar(170,17,227,60);
bar(7,113,310,400);bar(118,400,168,460);
bar(327,113,630,400);bar(508,426,565,466);
bar(270,460,370,470);
pic(o1,o2,r,an);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setcolor(4);
setcolor(5);
}
else
{
setcolor(5);
outtextxy(250,100,"CAN'T GO THIS POSITION PRESS ENTER KEY.");
delay(1000);setfillstyle(0,11);
bar(248,97,630,108);setcolor(5);
}
}

void pic(float o1,float o2,float r,float an)
{ float x1,y1,x2,y2,x3,y3,x4,y4,x5,y5,l1,l2,l3;
int triangle[] = {70,300,40,366,100,366,70,300};

setlinestyle(SOLID_LINE,0,1);
l1=60; l2=102; l3=90;
x1=70; y1=300;
x2=x1+l1*cos(o1); y2=y1-l1*sin(o1);
x3=x2+l2; y3=y2;
x4=x3+l3*cos(M_PI-o2); y4=y3+l3*sin(M_PI-o2);
x5=390+2*r*sin(an*M_PI/180); y5=295+2*r*cos(an*M_PI/180);

setcolor(6);
fillpoly(4,triangle);line(100,366,300,366);
arc(390,295,270,90,140);

setcolor(1);
circle(ceil(x1),ceil(y1),9);floodfill(ceil(x1),ceil(y1),1);
circle(ceil(x2),ceil(y2),6);floodfill(ceil(x2),ceil(y2),1);
circle(ceil(x3),ceil(y3),6);floodfill(ceil(x3),ceil(y3),1);

circle(ceil(x4),ceil(y4),6);floodfill(ceil(x4),ceil(y4),1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

setlinestyle(0,0,3); setcolor(5);
line(ceil(x1),ceil(y1),ceil(x2),ceil(y2));
line(ceil(x2),ceil(y2),ceil(x3),ceil(y3));
line(ceil(x3),ceil(y3),ceil(x4),ceil(y4));
line(ceil(x4),ceil(y4),ceil(x4)+15,ceil(y4));

setcolor(1);
circle(390,295,50);floodfill(390,295,1);
setcolor(5); line(390,295,ceil(x5),ceil(y5));
setcolor(4);
outtextxy(170,21,rt_str); outtextxy(170,37,ht_str); outtextxy(170,53,at_str);
outtextxy(120,420,rt_str); outtextxy(120,436,ht_str); outtextxy(510,446,at_str);
}

int dexe(int diff,int id)
{
if ( diff < 0 )
switch(id){
case 1 : start = 109;stop = 103;S1exe(diff,start,stop);break;
case 2 : start = 111;stop = 104;S2exe(diff,start,stop);break;
case 3 : start = 107;stop = 102;Mexe(diff,start,stop);break;}
else if ( diff > 0 )
switch(id){
case 1 : start = 108;stop = 103;S1exe(diff,start,stop);break;
case 2 : start = 110;stop = 104;S2exe(diff,start,stop);break;
case 3 : start = 106;stop = 102;Mexe(diff,start,stop);break;}
return 0 ;
}

void S1exe(int diff,int start,int stop)
{ int i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

diff = abs(diff);
if ( diff > 50 )
    for (i=0;i<50;i++) {port(start);delay(timeS1);port(stop);}
else if ( diff > 20 )
    for (i=0;i<20;i++) {port(start);delay(timeS1);port(stop);}
else if ( diff > 10 )
    for (i=0;i<10;i++) {port(start);delay(timeS1);port(stop);}
else if ( diff > 5 )
    for (i=0;i< 5;i++) {port(start);delay(timeS1);port(stop);}
else if ( diff > 1 ) {port(start);delay(timeS1);port(stop);}
}

```

```

void S2exe(int diff,int start,int stop)
{ int i;

diff = abs(diff);
if ( diff > 50 )
    for (i=0;i<50;i++) {port(start);delay(timeS2);port(stop);}
else if ( diff > 20 )
    for (i=0;i<20;i++) {port(start);delay(timeS2);port(stop);}
else if ( diff > 10 )
    for (i=0;i<10;i++) {port(start);delay(timeS2);port(stop);}
else if ( diff > 5 )
    for (i=0;i< 5;i++) {port(start);delay(timeS2);port(stop);}
else if ( diff > 1 ) {port(start);delay(timeS2);port(stop);}
}

```

```

void Mexe(int diff,int start,int stop)

```

```

{ int i;

```

```

diff = abs(diff);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if ( diff > 400 )
    for (i=0;i< 40;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 200 )
    for (i=0;i< 20;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 100 )
    for (i=0;i< 10;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 50 )
    for (i=0;i< 6;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 20 )
    for (i=0;i< 4;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 10 )
    for (i=0;i< 3;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 5 )
    for (i=0;i< 2;i++) {port(start);delay(timeM);port(stop);}
else if ( diff > 1 ) {port(start);delay(timeM);port(stop);delay(100);}
}

int port(int code)
{
    unsigned char chr;
    int result,status,x[10];
    int n = 0;

    _bios_serialcom(_COM_INIT,PORT_NO,_COM_9600 | _COM_NOPARITY |
        _COM_CHR8);

    _bios_serialcom(_COM_SEND,PORT_NO,code);

    for(;;){
        status = _bios_serialcom(_COM_STATUS,PORT_NO,0);
        if(!( status & 256 ))
            continue;

        chr = (unsigned char)_bios_serialcom(_COM_RECEIVE,PORT_NO,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x[n] = chr;
if( chr == 255 )
break;
n++;
}
ipdataL = x[0]; ipdataH = x[1];
return 0;
}

```

```

float indata(int anx,int directM)
{
int data_x1,data_x2,data_anL,data_anH,sw;

port(98);port(98);port(98);port(98);
data_x1 = ipdataL;inv_cal2(data_x1,id_1);
port(99);port(99);port(99);port(99);
data_x2 = ipdataL;inv_cal2(data_x2,id_2);
port(97);data_anL = ipdataL;data_anH = ipdataH,port(101);
if (directM) ani = anx + inv_an(data_anL,data_anH);
else { if (ani < 0) ani = 0;
else ani = anx - inv_an(data_anL,data_anH); }
inv_o1(x1);inv_o2(x2);inv_cal1(o1,o2);ri = r;hi = h;
return 0;
}

```

```

void start()
{ int sw;

port(102); port(100);sw = ipdataL;
while(!(sw == 3))
{ port(106);delay(30);port(102);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

port(100);sw = ipdataL;delay(100);}

port(101);

}

void end_auto(int ani)
{
indata(ani,1);cal(r,h);cal2(o1,o2);

setcolor(5);

setfillstyle(0,11);bar(248,97,630,108);

setfillstyle(0,11);bar(460,17,520,60);setcolor(4);

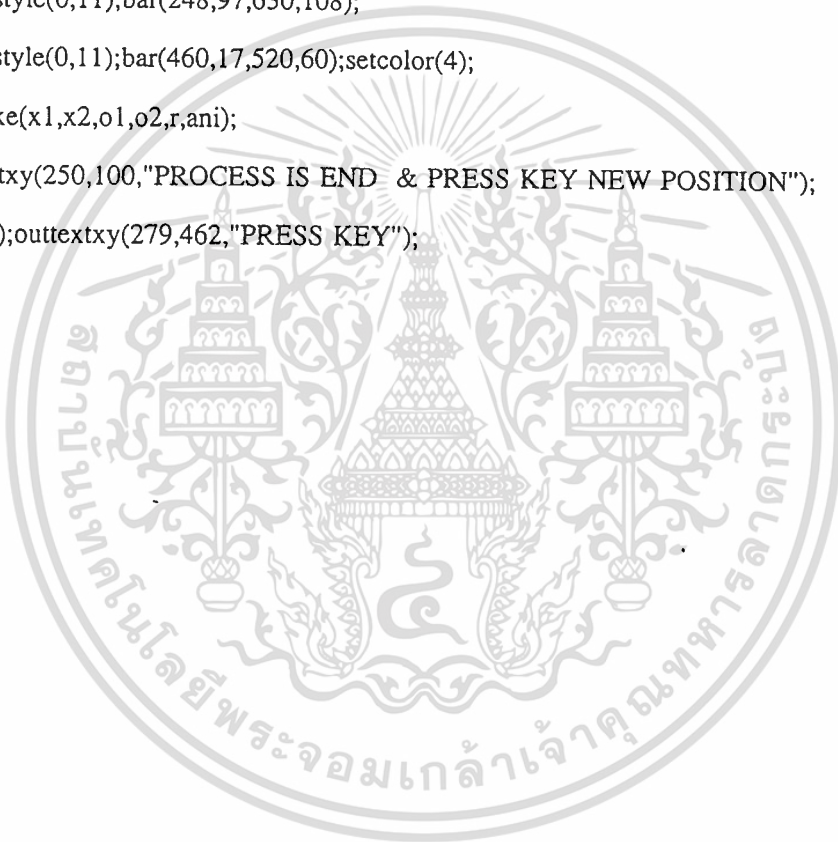
pic_exe(x1,x2,o1,o2,r,ani);

outtextxy(250,100,"PROCESS IS END & PRESS KEY NEW POSITION");

getch();outtextxy(279,462,"PRESS KEY");

}

```



ภาคผนวก
โปรแกรม 8031



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

org 0000h
contrl equ 0e100h
porta equ 0e101h
portb equ 0e102h
portc equ 0e103h

```

```

jmp main

```

```

org 0023h

```

```

jmp recv

```

```

;-----

```

```

main : mov scon,#50h
      mov tmod,#2eh
      mov th1,#fdh
      setb tr1
      clr it0
      setb tr0
      mov dptr,#contrl
      mov a,#0dh
      movx @dptr,a
      mov a,#00h
      mov th0,a
      mov tl0,a
      mov dptr,#porta
      movx @dptr,a
      setb ea
loop : setb es
      jmp loop

```

```

;-----

```

```

recv : clr es

```

```

      clr ri

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mov a,sbuf
```

```
-----
```

```
cjne a,#01h,send1 ;sendpulse
```

```
clr ti
```

```
mov a,ti0
```

```
mov sbuf,a
```

```
jnb ti,$
```

```
clr ti
```

```
mov a,th0
```

```
mov sbuf,a
```

```
jnb ti,$
```

```
clr ti
```

```
reti
```

```
-----
```

```
send1 : cjne a,#02h,send2 ;sendadc1
```

```
orl portc,#01h
```

```
mov a,portb
```

```
clr ti
```

```
mov a,portb
```

```
clr ti
```

```
mov sbuf,a
```

```
jnb ti,$
```

```
clr ti
```

```
reti
```

```
-----
```

```
send2 : cjne a,#03h,send3 ;sendadc2
```

```
orl portc,#02h
```

```
mov a,portb
```

```
clr ti
```

```
mov sbuf,a
```

```
jnb ti,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
clr ti
```

```
reti
```

```
-----  
send3: cjne a,#64h,clrp1 ;sendswitch
```

```
    jb p1.2,a1
```

```
    jb p1.1,a2
```

```
    mov a,#00h
```

```
    jmp snd
```

```
a1:  jb p1.1,a3
```

```
    mov a,#01h
```

```
    jmp snd
```

```
a3:  mov a,#03h
```

```
    jmp snd
```

```
a2:  mov a,#02h
```

```
snd: clr ti
```

```
    mov sbuf,a
```

```
    jnb ti,$
```

```
    clr ti
```

```
    mov a,#0ffh
```

```
    mov sbuf,a
```

```
    jnb ti,$
```

```
    clr ti
```

```
    reti
```

```
-----  
clrp1: cjne a,#65h,stop1 ;clrpulse
```

```
    mov tl0,#00h
```

```
    mov th0,#00h
```

```
    clr ti
```

```
    mov a,#0ffh
```

```
    mov sbuf,a
```

```
    jnb ti,$
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
reti
```

```
-----
stop1: cjne a,#66h,stop2 ;stop1
```

```
mov dptr,#porta
```

```
movx a,@dptr
```

```
anl a,#0fch
```

```
movx @dptr,a
```

```
clr ti
```

```
mov a,#0ffh
```

```
mov sbuf,a
```

```
jnb ti,$
```

```
clr ti
```

```
reti
```

```
-----
stop2: cjne a,#67h,stop3
```

```
mov dptr,#porta
```

```
movx a,@dptr
```

```
anl a,#0f3h
```

```
movx @dptr,a
```

```
clr ti
```

```
mov a,#0ffh
```

```
mov sbuf,a
```

```
jnb ti,$
```

```
clr ti
```

```
reti
```

```
-----
stop3: cjne a,#68h,stop4
```

```
mov dptr,#porta
```

```
movx a,@dptr
```

```
anl a,#0cfh
```

```
movx @dptr,a
```

```
clr ti
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

```

;-----
stop4: cjne a,#69h,right

```

```

mov dptr,#porta
movx a,@dptr
anl a,#03fh
movx @dptr,a
clr ti
mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

```

;-----
right: cjne a,#6ah,left

```

```

mov dptr,#porta
movx a,@dptr
anl a,#0fch
orl a,#01h
movx @dptr,a
clr ti
mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

เอกสารนี้เป็น left: cjne a,#6bh,up1 บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov dptr,#porta
movx a,@dptr
anl a,#0fch
orl a,#02h
movx @dptr,a
clr ti
mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

;------;

```

up1:  cjne a,#6ch,down1
      mov dptr,#porta
      movx a,@dptr
      anl a,#0f3h
      orl a,#0ch
      movx @dptr,a
      clr ti
      mov a,#0ffh
      mov sbuf,a
      jnb ti,$
      clr ti
      reti

```

;------;

```

down1: cjne a,#6dh,up2
      mov dptr,#porta
      movx a,@dptr
      anl a,#0f3h
      orl a,#08h
      movx @dptr,a
      clr ti

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

;------;

```
up2:  cjne a,#6eh,down2
```

```

mov dptr,#porta
movx a,@dptr
anl a,#0cfh
orl a,#30h
movx @dptr,a
clr ti
mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

;------;

```
down2: cjne a,#6fh,keep
```

```

mov dptr,#porta
movx a,@dptr
anl a,#0cfh
orl a,#20h
movx @dptr,a
clr ti
mov a,#0ffh
mov sbuf,a
jnb ti,$
clr ti
reti

```

```

keep: cjne a,#70h,out
      mov dptr,#porta
      movx a,@dptr
      anl a,#03fh
      orl a,#40h
      movx @dptr,a
      clr ti
      mov a,#0ffh
      mov sbuf,a
      jnb ti,$
      clr ti
      reti
;
out:  clr ti
      mov a,#0ffh
      mov sbuf,a
      jnb ti,$
      clr ti
      reti
end

```



6-Pin DIP Optoisolators Transistor Output

These devices consist of a gallium arsenide infrared emitting diode optically coupled to a monolithic silicon phototransistor detector.

- Convenient Plastic Dual-In-Line Package
- Most Economical Optoisolator
- High Input-Output Isolation Guaranteed — 7500 Volts Peak
- Meets or Exceeds All JEDEC Registered Specifications
- UL Recognized. File Number E54915
- VDE approved per standard 0833 6.83 (Certificate number 41853), with additional approval to DIN IEC380 VDE0805, IEC335 VDE0805, IEC65 VDE0860, VDE 110b, covering all other standards with equal or less stringent requirements, including IEC204: 057 aas VDE0113, VDE0160, VDE0832, VDE0833, etc.
- Special lead form available (add suffix "T" to part number) which satisfies VDE0883 6.80 requirement for 8 mm minimum creepage distance between input and output solder pads.
- Various lead form options available. Consult "Optoisolator Lead Form Options" data sheet for details.

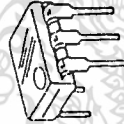
MAXIMUM RATINGS (T_A = 25°C unless otherwise noted)

Rating	Symbol	Value	Unit
INPUT LED			
Reverse Voltage	V _R	3	Volts
Forward Current — Continuous	I _F	60	mA
LED Power Dissipation (T _A = 25°C with Negligible Power in Output Detector Derate above 25°C)	P _D	120	mW
		1.41	mW/°C
OUTPUT TRANSISTOR			
Collector-Emitter Voltage	V _{CEO}	30	Volts
Emitter-Collector Voltage	V _{ECQ}	7	Volts
Collector-Base Voltage	V _{CBQ}	70	Volts
Collector Current — Continuous	I _C	150	mA
Detector Power Dissipation (T _A = 25°C with Negligible Power in Input LED Derate above 25°C)	P _D	150	mW
		1.76	mW/°C
TOTAL DEVICE			
Isolation Surge Voltage (1) (Peak ac Voltage, 60 Hz, 1 sec Duration)	V _{ISO}	7500	V _{ac}
Total Device Power Dissipation (T _A = 25°C Derate above 25°C)	P _D	250	mW
		2.94	mW/°C
Ambient Operating Temperature Range	T _A	-55 to +100	°C
Storage Temperature Range	T _{stg}	-55 to +150	°C
Soldering Temperature (10 sec, 1 16" from case)	T _{sol}	260	°C

(1) Isolation surge voltage is an internal device dielectric breakdown rating. For this test, Pins 1 and 2 are common, and Pins 4, 5 and 6 are common.

4N25
4N25A
4N26
4N27
4N28

6-PIN DIP
OPTOISOLATORS
TRANSISTOR OUTPUT



CASE 730A-02
PLASTIC

SCHEMATIC



1: LED ANODE
2: LED CATHODE
3: N.C.
4: EMITTER
5: COLLECTOR
6: BASE

ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

Characteristic	Symbol	Min	Typ	Max	Unit
INPUT LED					
Forward Voltage (I _F = 10 mA)	V _F	—	1.15 1.3 1.05	1.5	Volts
Reverse Leakage Current (V _R = 3 V)	I _R	—	—	100	μA
Capacitance (V = 0 V, f = 1 MHz)	C _J	—	18	—	pF
OUTPUT TRANSISTOR					
Collector-Emitter Dark Current (I _{VCE} = 10 V, T _A = 25°C)	I _{CEO}	—	1	50	nA
		—	1	100	nA
I _{VCE} = 10 V, T _A = 100°C	I _{CEO}	—	1	—	μA
Collector-Base Dark Current (V _{CB} = 10 V)	I _{CBQ}	—	9.2	—	nA
Collector-Emitter Breakdown Voltage (I _C = 1 mA)	V _{BR(CEO)}	30	45	—	Volts
Collector-Base Breakdown Voltage (I _C = 100 μA)	V _{BR(CBO)}	70	100	—	Volts
Emitter-Collector Breakdown Voltage (I _E = 100 μA)	V _{BR(ECO)}	7	7.8	—	Volts
DC Current Gain (I _C = 2 mA, V _{CE} = 5 V)	h _{FE}	—	500	—	—
Collector-Emitter Capacitance (f = 1 MHz, V _{CE} = 0)	C _{CE}	—	7	—	pF
Collector-Base Capacitance (f = 1 MHz, V _{CB} = 0)	C _{CB}	—	19	—	pF
Emitter-Base Capacitance (f = 1 MHz, V _{EB} = 0)	C _{EB}	—	9	—	pF
COUPLED					
Output Collector Current (I _F = 10 mA, V _{CE} = 10 V)	I _C	2	7	—	mA
		1	5	—	—
Collector-Emitter Saturation Voltage (I _C = 2 mA, I _F = 50 mA)	V _{CE(sat)}	—	0.15	0.5	Volts
Turn-On Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)	t _{on}	—	2.8	—	μs
Turn-Off Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)	t _{off}	—	4.5	—	μs
Rise Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)	t _r	—	1.2	—	μs
Fall Time (I _F = 10 mA, V _{CC} = 10 V, R _L = 100 Ω)	t _f	—	1.3	—	μs
Isolation Voltage (f = 60 Hz, t = 1 sec)	V _{ISO}	7500	—	—	Vac(pk)
Isolation Resistance (V = 500 V)	R _{ISO}	10 ¹¹	—	—	Ω
Isolation Capacitance (V = 0 V, f = 1 MHz)	C _{ISO}	—	0.2	—	pF

TYPICAL CHARACTERISTICS

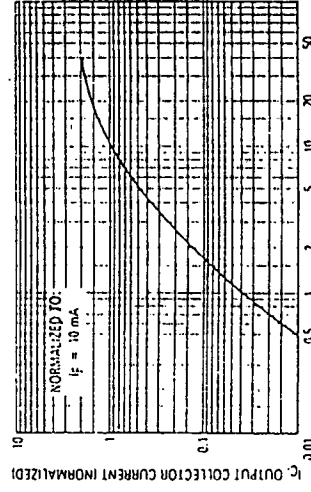
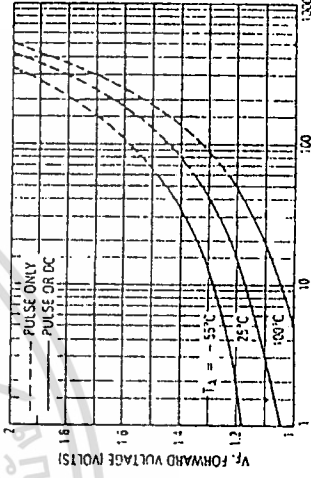


Figure 1. LED Forward Voltage versus Forward Current

Figure 2. Output Current versus Input Current

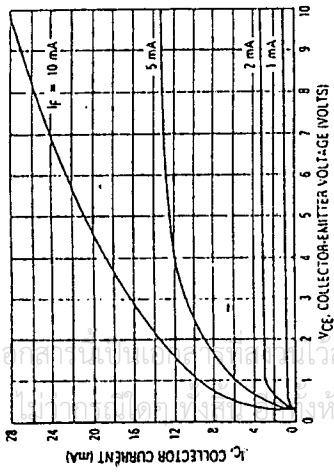


Figure 3. Collector Current versus Collector-Emitter Voltage

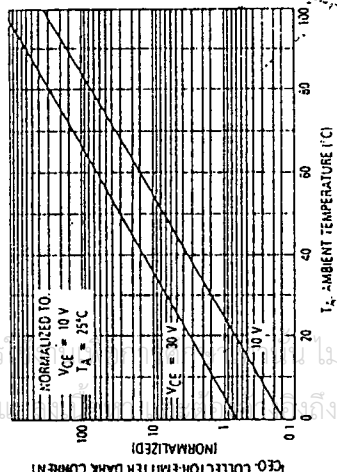


Figure 5. Dark Current versus Ambient Temperature

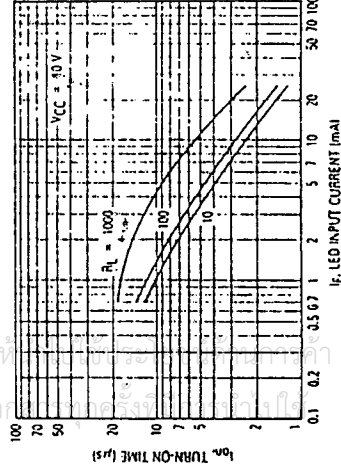


Figure 7. Turn-On Switching Times

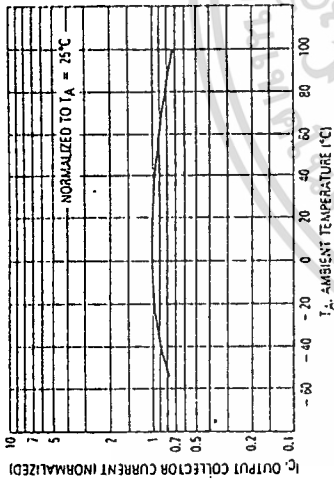


Figure 4. Output Current versus Ambient Temperature

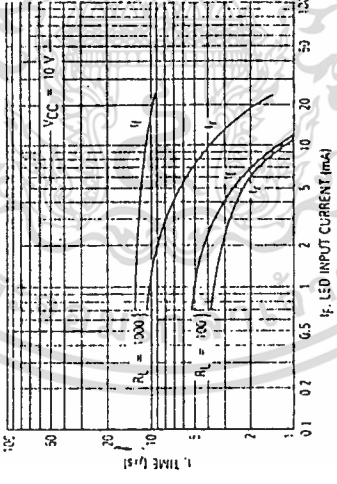


Figure 6. Rise and Fall Times

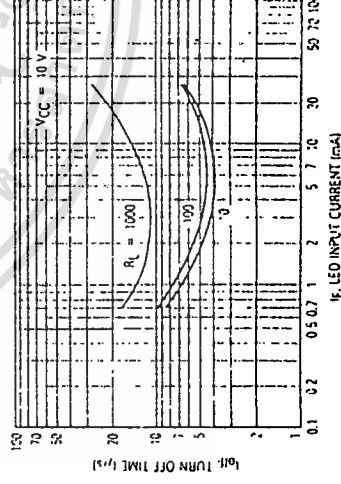


Figure 8. Turn-Off Switching Times

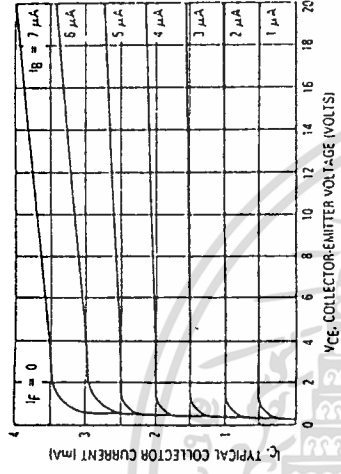


Figure 9. DC Current Gain (Detector Only)

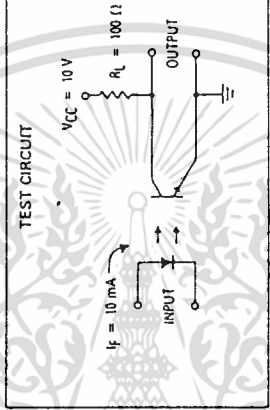


Figure 11. Switching Times

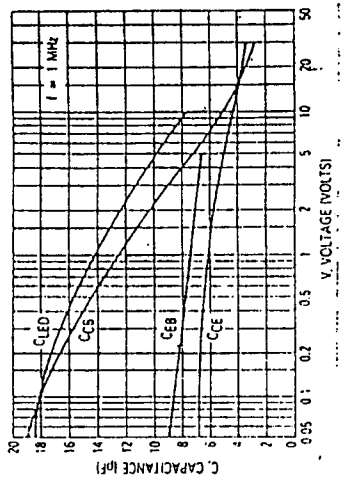


Figure 10. Capacitances versus Voltage

OUTLINE DIMENSIONS

Figure 12 shows the outline dimensions of the device. The dimensions are given in millimeters and inches. The device is a plastic package (CASE 730A-02) with a lead spacing of 0.100 inches. The dimensions are as follows:

MILLIMETERS		INCHES		
DAH	MIN	MAX	MIN	MAX
A	1.13	1.29	0.070	0.500
B	5.10	5.60	0.240	0.750
C	1.93	2.08	0.115	0.750
D	0.11	0.50	0.016	0.900
E	1.02	1.17	0.040	0.900
G	1.54	BSC	0.100	BSC
H	0.71	0.30	0.028	0.017
I	0.38	1.51	0.015	0.100
L	1.67	BSC	0.100	BSC
M	0.7	1.5	0.028	0.15
N	1.54	1.81	0.100	0.150

NOTES:
 1. DIMENSIONING AND TOLERANCING PER ANSI Y14.5M, 1982.
 2. CONTROLLING DIMENSION INCH.
 3. DIM L TO CENTER OF LEAD WHEN FORMED PARALLEL.

STYLE 1:
 PIN 1 ANODE
 1. CATHODE
 2. ANODE
 3. EMITTER
 4. COLLECTOR
 5. BASE

OPTIONAL LEAD CONFIGURATION

PLASTIC CASE 730A-02

8155H/8156H/8155H-2/8156H-2 2048-BIT STATIC HMOS RAM WITH I/O PORTS AND TIMER

- Single +5V Power Supply with 10% Voltage Margins
- 30% Lower Power Consumption than the 8155 and 8156.
- 1256 Word x 8 Bits
- Completely Static Operation
- Internal Address Latch
- 2 Programmable 8-Bit I/O Ports
- 1 Programmable 6-Bit I/O Port
- Programmable 14-Bit Binary Counter/Timer
- Compatible with 8085AH and 8088 CPU
- Multiplexed Address and Data Bus
- Available In EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

The Intel® 8155H and 8156H are RAM and I/O chips implemented in N-Channel, depletion load, silicon gate technology (HMOS), to be used in the 8085AH and 8088 microprocessor systems. The RAM portion is designed with 2048 static cells organized as 256 x 8. They have a maximum access time of 400 ns to permit use with no wait states in 8085AH CPU. The 8155H-2 and 8156H-2 have maximum access times of 330 ns for use with the 8085H-2 and the 5 MHz 8088 CPU.

The I/O portion consists of three general purpose I/O ports. One of the three ports can be programmed to be status pins, thus allowing the other two ports to operate in handshake mode.

A 14-bit programmable counter/timer is also included on chip to provide either a square wave or terminal count pulse for the CPU system depending on timer mode.

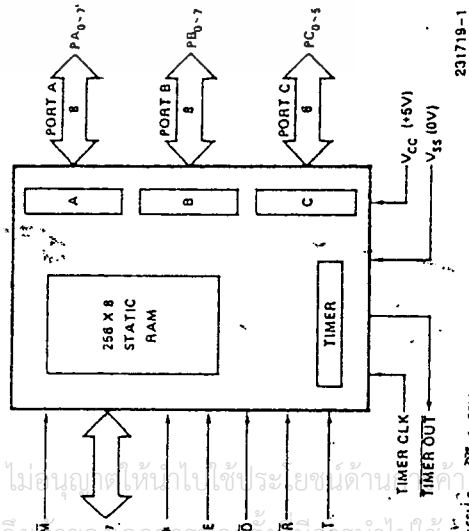


Figure 1. Block Diagram

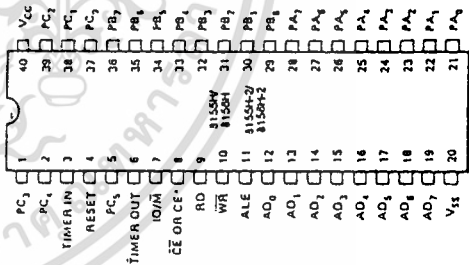


Figure 2. Pin Configuration

Table 1. Pin Description

Symbol	Type	Name and Function
RESET	I	RESET: Pulse provided by the 8085AH to initialize the system (connect to 8085AH RESET OUT). Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be two 8085AH clock cycle times.
AD ₀₋₇	I/O	ADDRESS/DATA: 3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch inside the 8155H/56H on the falling edge of ALE. The address can be either for the memory section or the I/O section depending on the IO/M input. The 8-bit data is either written into the chip or read from the chip, depending on the WR or RD input signal.
CE or \overline{CE}	I	CHIP ENABLE: On the 8155H, this pin is \overline{CE} and is ACTIVE LOW. On the 8156H, this pin is CE and is ACTIVE HIGH.
RD	I	READ CONTROL: Input low on this line with the Chip Enable active enables and AD ₀₋₇ buffers. If IO/M pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port or command/status registers will be read to the AD bus.
WR	I	WRITE CONTROL: Input low on this line with the Chip Enable active causes the data on the Address/Data bus to be written to the RAM or I/O ports and command/status register, depending on IO/M.
ALE	I	ADDRESS LATCH ENABLE: This control signal latches both the address on the AD ₀₋₇ lines and the state of the Chip Enable and IO/M into the chip at the falling edge of ALE.
IO/M	I	I/O MEMORY: Selects memory if low and I/O and command/status registers if high.
PA ₀₋₇ (8)	I/O	PORT A: These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
PB ₀₋₇ (8)	I/O	PORT B: These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the command register.
PC ₀₋₅ (6)	I/O	PORT C: These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the command register. When PC ₀₋₅ are used as control signals, they will provide the following: PC ₀ —A INTR (Port A Interrupt) PC ₁ —ABF (Port A Buffer Full) PC ₂ —A STB (Port A Strobe) PC ₃ —B INTR (Port B Interrupt) PC ₄ —B BF (Port B Buffer Full) PC ₅ —B STB (Port B Strobe)
TIMER IN	I	TIMER INPUT: Input to the timer-counter.
TIMER OUT	O	TIMER OUTPUT: This output can be either a square wave or a pulse, depending on the timer mode.
V _{CC}		VOLTAGE: +5V supply.
V _{SS}		GROUND: Ground reference.

PROGRAMMING OF THE COMMAND REGISTER

The command register consists of eight latches. Four bits (0-3) define the mode of the ports, two bits when it acts as control port, and the last two bits (6-7) are for the timer.

The command register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation with the Chip Enable active and IO/M = 1. The meaning of each bit of the command byte is defined in Figure 5. The contents of the command register may never be read.

READING THE STATUS REGISTER

The status register consists of seven latches, one for each bit; six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the Status Register (Address XXXXX000). Status word format is shown in Figure 6. Note that you may never write to the status register since the command register shares the same I/O address and the command register is selected when a write to that address is issued.

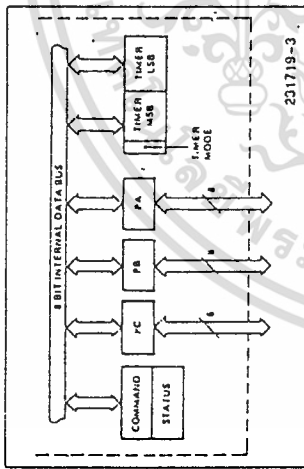


Figure 3. 8155H/8156H Internal Registers

FUNCTIONAL DESCRIPTION

- The 8155H/8156H contains the following:
 - 2K Bit Static RAM organized as 256 x 8
 - Two 8-bit I/O ports (PA & PB) and one 6-bit I/O port (PC)
 - 14-bit timer-counter
- The IO/M (IO/ Memory Select) pin selects either the I/O registers (Command, Status, PA₀₋₇, PB₀₋₇, PC₀₋₅) or the memory (RAM) portion.
- The 8-bit address on the Address/Data lines, Chip Enable input CE or \overline{CE} , and IO/M are all latched on-chip at the falling edge of ALE.

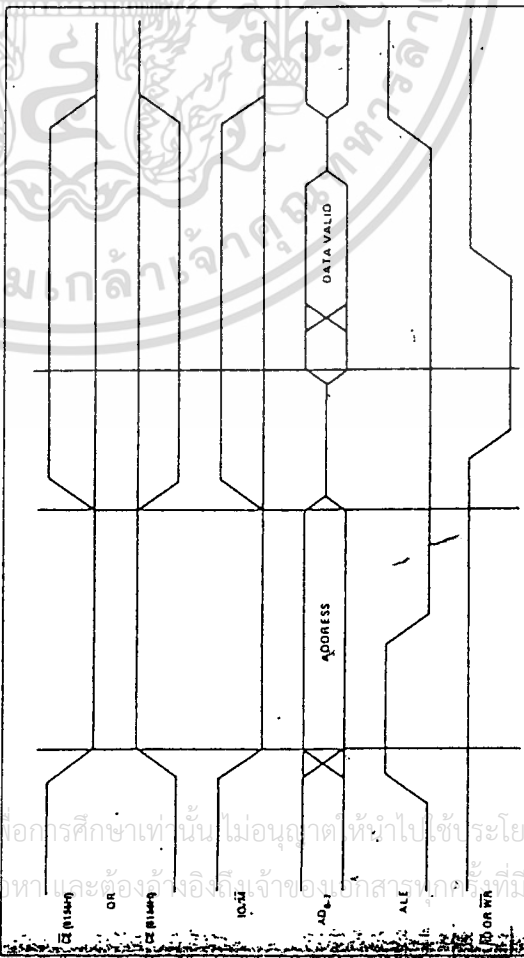


Figure 4. 8155H/8156H On-Board Memory Read/Write Cycle

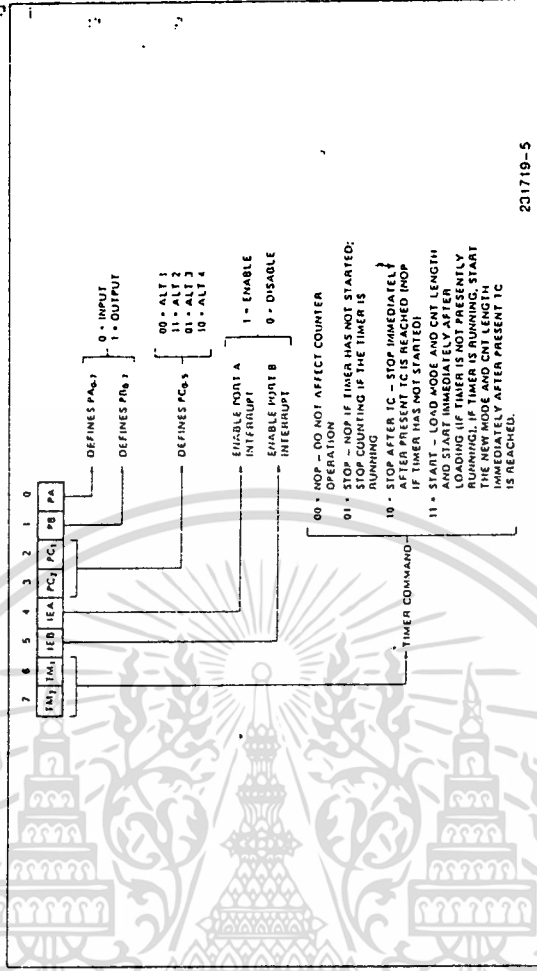


Figure 5. Command Register Bit Assignment

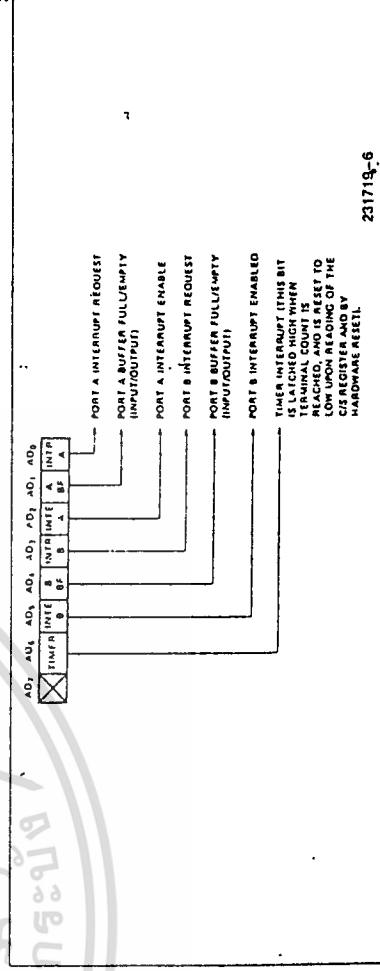


Figure 6. Status Register Bit Assignment

The counter in the 8155H is not initialized to any particular mode or count when hardware RESET occurs, but RESET does stop the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

Please note that the timer circuit on the 8155H/8156H chip is designed to be a square-wave timer, not an event counter. To achieve this, it counts down by twos twice in completing one cycle. Thus, its registers do not contain values directly representing the number of TIMER IN pulses received. You cannot load an initial value of 1 into the count register and cause the timer to operate, as its terminal count value is 10 (binary) or 2 (decimal). (For the detection of single pulses, it is suggested that one of the hardware interrupt pins on the 8085AH be used.) After the timer has started counting down, the values residing in the count registers can be used to calculate the actual number of TIMER IN pulses required to complete the timer cycle if desired. To obtain the remaining count, perform the following operations in order:

1. Stop the count
2. Read in the 16-bit value from the count length registers
3. Reset the upper two mode bits
4. Reset the carry and rotate right one position all 16 bits through carry
5. If carry is set, add 1/2 of the full original count (1/2 full count—1 if full count is odd).

NOTE:

If you started with an odd count and you read the count length register before the third count pulse occurs, you will not be able to discern whether one or two counts has occurred. Regardless of this, the 8155H/56H always counts out the right number of pulses in generating the TIMER OUT waveforms.

8-7 (TM₂ and TM₁) of command register controls are used to start and stop the counter. There are four commands to choose from:

- TM₁**
- 0 STOP—Do not affect counter operation.
 - 1 STOP—NOP if timer has not started; stop counting if the timer is running.
 - 0 STOP AFTER TC—Stop immediately after present TC is reached (NOP if timer has not started)
 - 1 START—Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

While the counter is counting, you may load a new count and mode into the count length registers. Before the new count and mode will be loaded by the counter, you must issue a START command to the counter. This applies even though you only want to change the count and use the previous mode.

In case of an odd-numbered count, the first half-cycle of the squarewave output, which is high, is one count longer than the second (low) half-cycle, as shown in Figure 12.

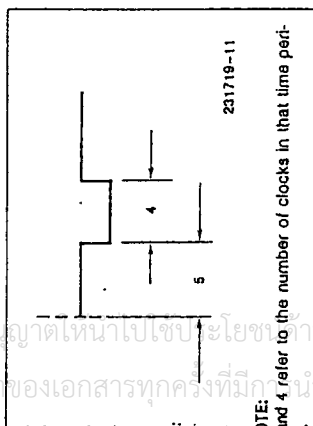


Figure 12. Asymmetrical Square-Wave Output Resulting from Count of 9

8085AH: MINIMUM SYSTEM CONFIGURATION

Figure 13a shows a minimum system using three chips, containing:

- 2K Bytes EPROM
- 38 I/O Pins
- 1 Interval Timer
- 4 Interrupt Levels
- 256 Bytes RAM

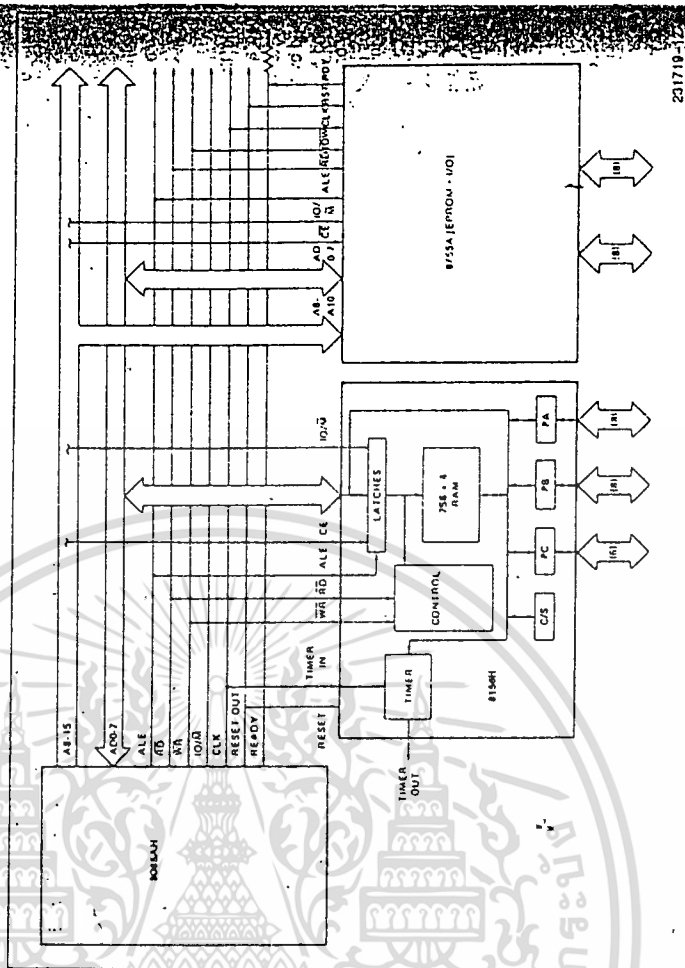


Figure 13a. 8085AH Minimum System Configuration (Memory Mapped I/O)

MAXIMUM RATINGS*

- Under Bias 0°C to + 70°C
- Temperature -65°C to + 150°C
- Power on Any Pin
- In Respect to Ground -0.5V to + 7V
- Power Dissipation 1.5W

NOTICE: This is a production data sheet. The specifications are subject to change without notice.
 *WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and intended exposure beyond the "Operating Conditions" may affect device reliability.

D.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5V \pm 10\%$

Symbol	Parameter	Min	Max	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	$V_{CC} + 0.5$	V	
V_{OL}	Output Low Voltage		0.45	V	$I_{OL} = 2\text{ mA}$
V_{OH}	Output High Voltage	2.4		V	$I_{OH} = -400\ \mu\text{A}$
I_{IL}	Input Leakage		± 10	μA	$0V \leq V_{IN} \leq V_{CC}$
I_{LO}	Output Leakage Current		± 10	μA	$0.45V \leq V_{OUT} \leq V_{CC}$
I_{CC}	V_{CC} Supply Current		125	mA	
$I_{IL}(\text{CE})$	Chip Enable Leakage		+ 100 - 100	μA μA	$0V \leq V_{IN} \leq V_{CC}$

A.C. CHARACTERISTICS $T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5V \pm 10\%$

Symbol	Parameter	8155H/8156H		8155H-2/8156H-2	
		Min	Max	Min	Max
t_{AL}	Address to Latch Setup Time	50		30	
t_{LA}	Address Hold Time after Latch	80		30	
t_{LC}	Latch to READ/WRITE Control	100		40	
t_{RD}	Valid Data Out Delay from READ Control		170		140
t_{LD}	Latch to Data Out Valid		350		270
t_{AD}	Address Stable to Data Out Valid		400		330
t_{LL}	Latch Enable Width		100		70
t_{RDF}	Data Bus Float after READ		0		80
t_{CL}	READ/WRITE Control to Latch Enable		20		10
t_{CLL}	WRITE Control to Latch Enable for C/S Register		125		125
t_{CC}	READ/WRITE Control Width		250		200
t_{DW}	Data In to WRITE Setup Time		150		100
t_{WD}	Data In Hold Time after WRITE		25		25
t_{RV}	Recovery Time between Controls		300		200
t_{WP}	WRITE to Port Output		400		300

- 38 I/O Pins
- 1 Interval Timer
- 2 Interrupt Levels

8088 FIVE CHIP SYSTEM

Figure 13b shows a five chip system containing:

- 125K Bytes RAM
- 12K Bytes EPROM

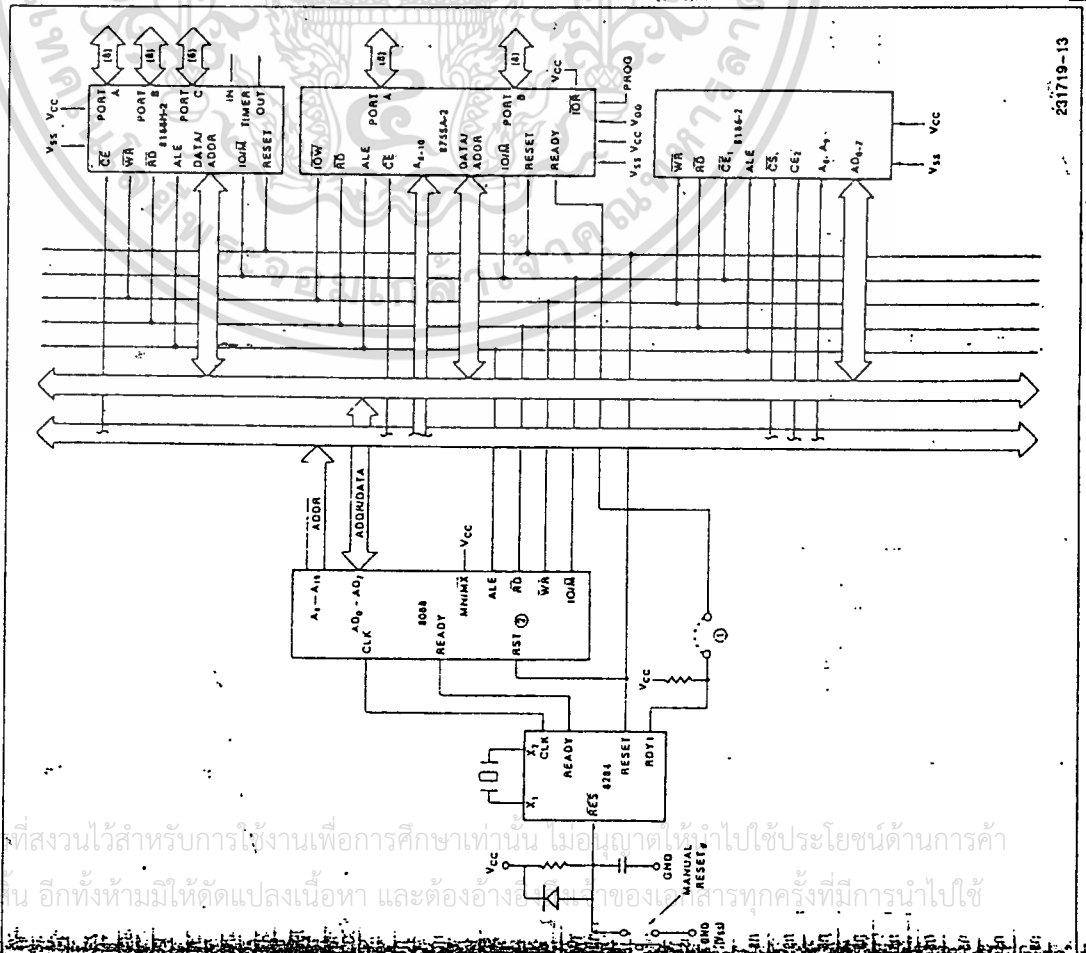
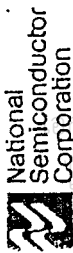


Figure 13b. 8088 Five Chip System Configuration.

เอกสารนี้เป็นทรัพย์สินส่วนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าในกรณีใดๆ หากท่านมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้



ADC0808, ADC0809 8-Bit μ P Compatible A/D Converters with 8-Channel Multiplexer

General Description

The ADC0808, ADC0809 data acquisition component is a nonvolatile CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 156R voltage divider with analog switch tree and a successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TTL TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized by incorporating the most desirable aspects of several A/D conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability, and consumes minimal power. These features make this device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

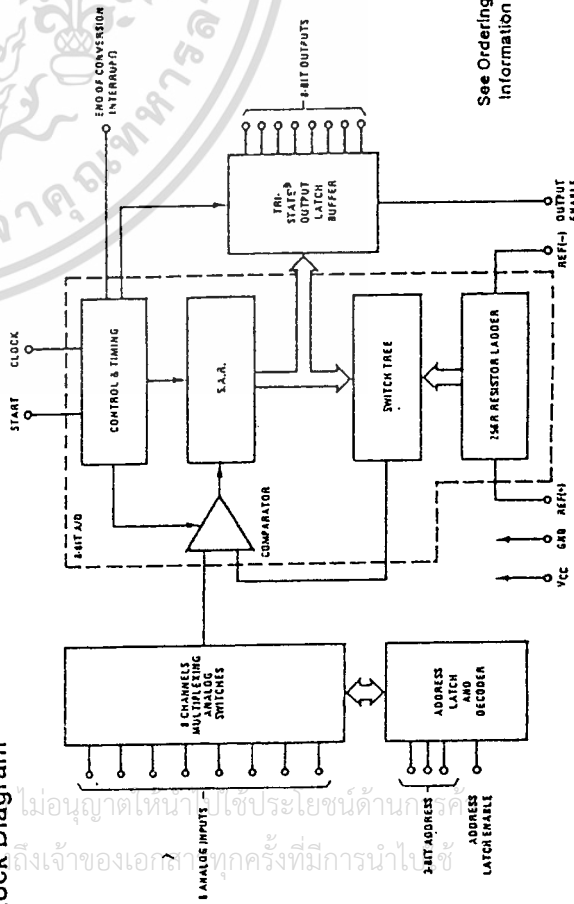
Features

- Easy interface to all microprocessors
- Operates ratiometrically or with 5 V_{CC} or analog span adjusted voltage reference
- No zero or full-scale adjust required
- 8-channel multiplexer with address logic
- 0V to 5V input range with single 5V power supply
- Outputs meet TTL voltage level specifications
- Standard hermetic or molded 28-pin DIP package
- 28-pin molded chip carrier package

Key Specifications

- Resolution 8 bits
- Total Unadjusted Error ± 1 LSB
- Single Supply 5 V_{CC}
- Low Power 15 mW
- Conversion Time 100 μ s

Block Diagram



See Ordering Information

Absolute Maximum Ratings (Notes 1 & 2)

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

Supply Voltage (V_{CC}) (Note 3) 6.5V
Voltage at Any Pin -0.3V to (V_{CC} + 0.3V)
Except Control Inputs

Voltage at Control Inputs -0.3V to +15V
(START, OE, CLOCK, ALE, ADD A, ADD B, ADD C)
Storage Temperature Range -55°C to +150°C
Package Dissipation at T_A = 25°C 375 mW

Lead Temp. (Soldering, 10 seconds) 260°C
Dual-In-Line Package (plastic) 300°C
Molded Chip Carrier Package 215°C
Vapor Phase (60 seconds) 220°C
Infrared (15 seconds) 400V
ESD Susceptibility (Note 11)

Operating Conditions (Notes 1 & 2)

Temperature Range (Note 1)
ADC0808CJ
ADC0808CCJ, ADC0808CCN,
ADC0809CCN
ADC0808CCV, ADC0809CCV
Range of V_{CC} (Note 1)

T_{MIN} \leq T_A \leq T_{MAX}
-55°C \leq T_A \leq +125°C
-40°C \leq T_A \leq +85°C
-40°C \leq T_A \leq +85°C
4.5 V_{CC} to 6.0 V_{CC}

Electrical Characteristics

Converter Specifications: V_{CC} = 5 V_{CC} = V_{REF} - V_{REF}(-), V_{REF}(-) = GND, T_{MIN} \leq T_A \leq T_{MAX} and f_{CLK} = 640 kHz unless otherwise stated.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
ADC0808	Total Unadjusted Error (Note 5)	25°C T _{MIN} to T _{MAX}			$\pm 1/2$ $\pm 1/4$	LSB LSB
ADC0809	Total Unadjusted Error (Note 5)	0°C to 70°C T _{MIN} to T _{MAX}			± 1 $\pm 1/4$	LSB LSB
	Input Resistance	From Ref(+) to Ref(-)	1.0	2.5		k Ω
	Analog Input Voltage Range	(Note 4) V(+) or V(-)	GND - 0.10		V _{CC} + 0.10	V _{CC}
V _{REF} (+)	Voltage, Top of Ladder	Measured at Ref(+)			V _{CC} + 0.1	V
V _{REF} (+), V _{REF} (-)	Voltage, Center of Ladder		V _{CC} /2 - 0.1		V _{CC} /2 + 0.1	V
V _{REF} (-)	Voltage, Bottom of Ladder	Measured at Ref(-)	-0.1	0		V
I _{IN}	Comparator Input Current	f _C = 640 kHz, (Note 6)	-2	± 0.5	2	μ A

Electrical Characteristics

Digital Levels and DC Specifications: ADC0808CJ 4.5V \leq V_{CC} \leq 5.5V, -55°C \leq T_A \leq +125°C unless otherwise noted
ADC0808CCJ, ADC0808CCN, ADC0809CCN and ADC0809CCV, 4.75V \leq V_{CC} \leq 5.25V, -40°C \leq T_A \leq +85°C unless otherwise noted

ANALOG MULTIPLEXER

Symbol	Parameter	Conditions	Min	Typ	Max	Units
I _{OFF} (+)	OFF Channel Leakage Current	V _{CC} = 5V, V _{IN} = 5V, T _A = 25°C T _{MIN} to T _{MAX}		10	200 1.0	nA μ A
I _{OFF} (-)	OFF Channel Leakage Current	V _{CC} = 5V, V _{IN} = 0, T _A = 25°C T _{MIN} to T _{MAX}	-200	-10		nA μ A

Electrical Characteristics (Continued)

Levels and DC Specifications: ADC0808CJ 4.5V ≤ V_{CC} ≤ 5.5V, -55°C ≤ T_A ≤ +125°C unless otherwise noted
 ADC0808CCJ, ADC0808CCN, ADC0808CCV, ADC0809CCN and ADC0809CCV, 4.75V ≤ V_{CC} ≤ 5.25V, -40°C ≤ T_A ≤ +85°C unless otherwise noted

Symbol	Parameter	Conditions	Min	Typ	Max	Units
CONTROL INPUTS						
V _{IN(1)}	Logical "1" Input Voltage		V _{CC} - 1.5			V
V _{IN(0)}	Logical "0" Input Voltage			1.5		V
I _{IN(1)}	Logical "1" Input Current (The Control Inputs)	V _{IN} = 15V		1.0		μA
I _{IN(0)}	Logical "0" Input Current (The Control Inputs)	V _{IN} = 0	-1.0			μA
I _{CC}	Supply Current	f _{CLK} = 640 kHz		3.0		mA
DATA OUTPUTS AND EOC (INTERRUPT)						
V _{OUT(1)}	Logical "1" Output Voltage	I _O = -360 μA	V _{CC} - 0.4			V
V _{OUT(0)}	Logical "0" Output Voltage	I _O = 1.8 mA		0.45		V
V _{OUT(0)}	Logical "0" Output Voltage EOC	I _O = 1.2 mA		0.45		V
I _{OUT}	TRI-STATE Output Current	V _O = 5V V _O = 0		3		μA μA

Electrical Characteristics

Timing Specifications V_{CC} = V_{REF(+)} = 5V, V_{REF(-)} = GND, t_r = t_f = 20 ns and T_A = 25°C unless otherwise noted.

Symbol	Parameter	Conditions	Min	Typ	Max	Units
t _{WS}	Minimum Start Pulse Width	(Figure 5)		100	200	ns
t _{WALE}	Minimum ALE Pulse Width	(Figure 5)		100	200	ns
t _S	Minimum Address Set-Up Time	(Figure 5)		25	50	ns
t _H	Minimum Address Hold Time	(Figure 5)		25	50	ns
t _D	Analog MUX Delay Time From ALE	R _S = 0 Ω (Figure 5)		1	2.5	μs
t _{H1} , t _{L0}	OE Control to 0 Logic State	C _L = 50 pF, R _L = 10k (Figure 6)		125	250	ns
t _{H1} , t _{L0}	OE Control to Hi-Z	C _L = 10 pF, R _L = 10k (Figure 6)		125	250	ns
t _C	Conversion Time	t _C = 640 kHz (Figure 5) (Note 7)	90	100	116	μs
f _C	Clock Frequency	(Figure 5)	10	640	1280	kHz
t _{EOC}	EOC Delay Time		0	8 ± 2	μs	Clock Periods
C _{IN}	Input Capacitance	At Control Inputs		10	15	pF
C _{OUT}	TRI-STATE Output Capacitance	At TRI-STATE Outputs, (Note 12)		10	15	pF

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.
 Note 2: All voltages are measured with respect to GND, unless otherwise specified.
 Note 3: A 10kΩ resistor is connected between V_{CC} to GND and has a typical breakdown voltage of 7 V_{CC}.
 Note 4: Two on-chip diodes are tied to each analog input which will forward conduct for analog input voltages one diode drop below ground or one diode drop greater than the V_{CC} supply. The spec allows 100 mV forward bias of either diodes. This means that as long as the analog V_{IN} does not exceed the supply voltage by more than 100 mV, the output code will be correct. To achieve an absolute 0V_{CC} to 5V_{CC} input voltage range will therefore require a minimum supply voltage of 4.900 V_{CC} over temperature variations, initial tolerance and loading.
 Note 5: Total uncalibrated error includes offset, full-scale, linearity, and multiplier errors. See Figure 7. None of these A/Ds requires a zero or full-scale adjust. However, if an all zero code is desired for an analog input other than 0.0V, or if a narrow full-scale span exists (for example: 0.5V to 4.5V full-scale) the reference voltage can be adjusted to achieve this. See Figure 17.
 Note 6: Comparator input current is a bias current into or out of the comparator stabilized comparator. The bias current varies directly with clock frequency and has little temperature dependence (Figure 6). See paragraph 4.0.
 Note 7: The outputs of the data register are updated one clock cycle before the rising edge of EOC.

Functional Description

Multiplexer. The device contains an 8-channel single-ended analog signal multiplexer. A particular input channel is selected by using the address decoder. Table 1 shows the input states for the address lines to select any channel. The address is latched into the decoder on the low-to-high transition of the address latch enable signal.

TABLE 1

SELECTED ANALOG CHANNEL	ADDRESS LINE			
	C	B	A	A
IN0	L	L	L	L
IN1	L	L	L	H
IN2	L	L	H	L
IN3	L	L	H	H
IN4	L	H	L	L
IN5	L	H	L	H
IN6	L	H	H	L
IN7	L	H	H	H

CONVERTER CHARACTERISTICS

The Converter

The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed

to give fast, accurate, and repeatable conversions over a wide range of temperatures. The converter is partitioned into 3 major sections: the 256R ladder network, the successive approximation register, and the comparator. The converter's digital outputs are positive true.

The 256R ladder network approach (Figure 1) was chosen over the conventional R/2R ladder because of its inherent monotonicity, which guarantees no missing digital codes. Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in Figure 1 are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached - 1/2 LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

The successive approximation register (SAR) performs 9 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. Figure 2 shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

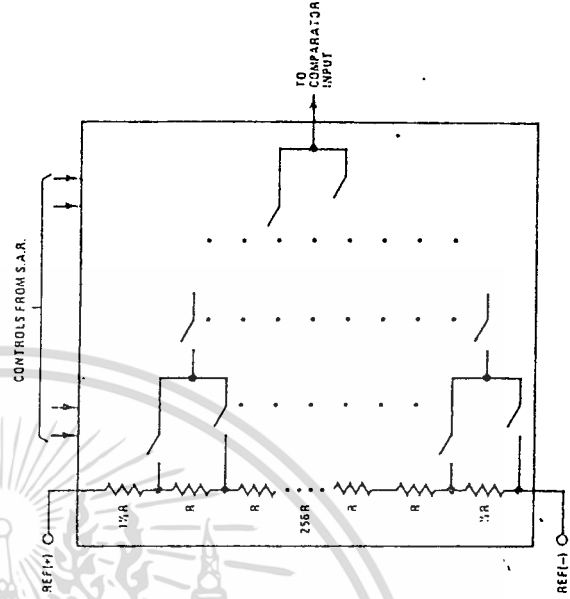


FIGURE 1. Resistor Ladder and Switch Tree

Functional Description (Continued)

The A/D converter's successive approximation register (SAR) is reset on the positive edge of the start conversion (SC) pulse. The conversion is begun on the falling edge of the start conversion pulse. A conversion in process will be interrupted by receipt of a new start conversion pulse. Continuous conversion may be accomplished by tying the end-of-conversion (EOC) output to the SC input. If used in this mode, an external start conversion pulse should be applied after power up. End-of-conversion will go low between 0 and 1 clock pulses after the rising edge of start conversion. The most important section of the A/D converter is the comparator. It is this section which is responsible for the ultimate accuracy of the entire converter. It is also the

comparator drift which has the greatest influence on the repeatability of the device. A chopper-stabilized comparator provides the most effective method of satisfying all the converter requirements.

The chopper-stabilized comparator converts the DC input signal into an AC signal. This signal is then fed through a high gain AC amplifier and has the DC level restored. This technique limits the drift component of the amplifier since the drift is a DC component which is not passed by the AC amplifier. This makes the entire A/D converter extremely insensitive to temperature, long term drift and input offset errors.

Figure 4 shows a typical error curve for the ADC0808 as measured using the procedures outlined in AN-179.

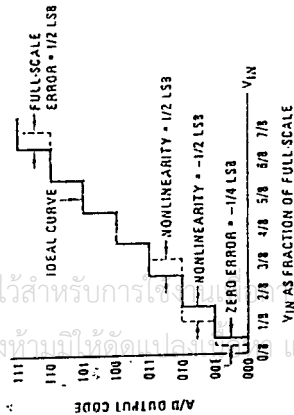


FIGURE 2. 3-Bit A/D Transfer Curve

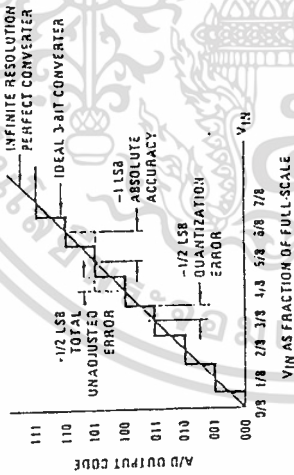


FIGURE 3. 3-Bit A/D Absolute Accuracy Curve

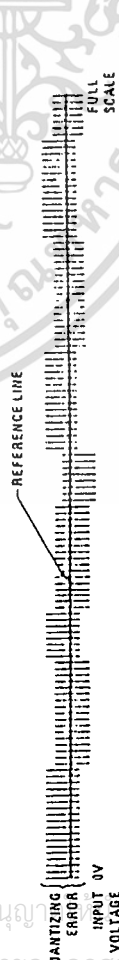
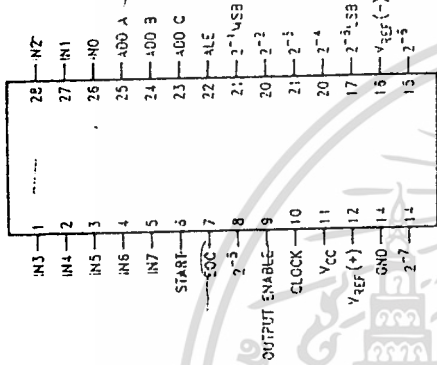


FIGURE 4. Typical Error Curve

Connection Diagrams

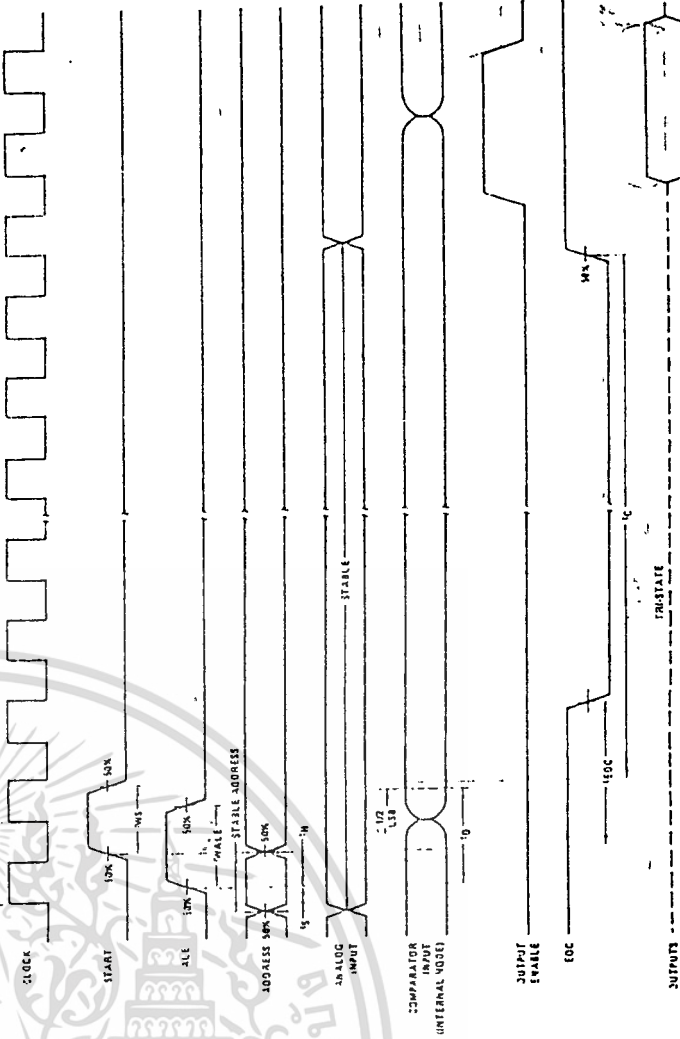
Dual-In-Line Package



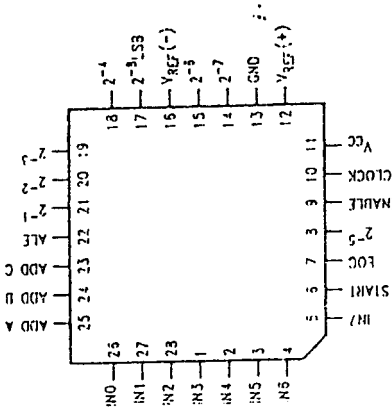
LM5672-11

Order Number ADC0808CCN, ADC0809CCN,
ADC0808CJ or ADC0808CJ
See NS Package J28A or N28A

Timing Diagram



Molded Chip Carrier Package



LM5672-12
Order Number ADC0808CCV or ADC0809CCV
See NS Package V28A

Applications Information (Continued)

The ADC0808 needs less than a milliamp of supply current to develop the supply from the reference is readily accomplished. In Figure 11 a ground referenced system is shown which generates the supply from the reference. The buffer shown can be an op amp of sufficient drive to supply the milliamp of supply current and the desired bus drive, or if a capacitive bus is driven by the outputs a large capacitor will supply the transient supply current as seen in Figure 12. The LM301 is overcompensated to insure stability when loaded by the 10 μ F output capacitor.

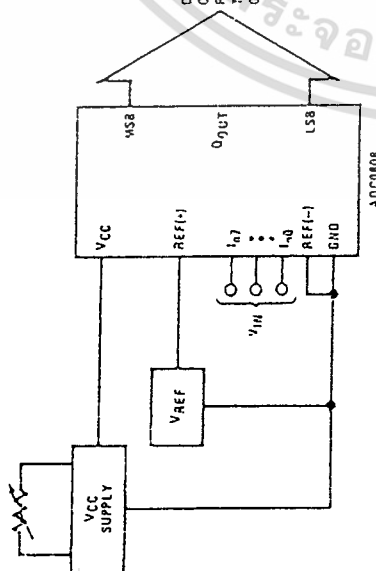


FIGURE 10. Ground Referenced Conversion System Using Trimmed Supply

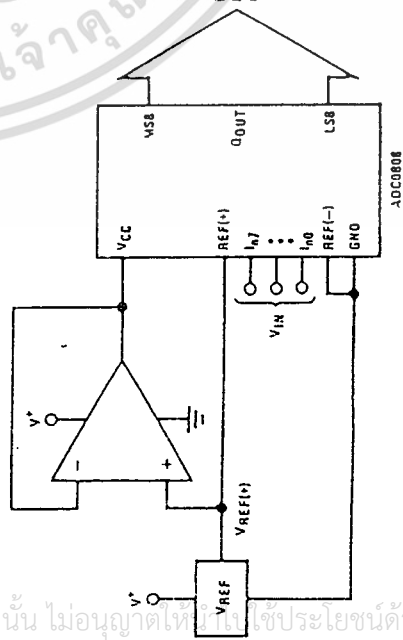


FIGURE 11. Ground Referenced Conversion System with Reference Generating VCC Supply

The top and bottom ladder voltages cannot exceed VCC and ground, respectively, but they can be symmetrically less than VCC and greater than ground. The center of the ladder voltage should always be near the center of the supply. The sensitivity of the converter can be increased, (i.e., size of the LSB steps decreased) by using a symmetrical reference system. In Figure 13, a 2.5V reference is symmetrically centered about VCC/2 since the same current flows in identical resistors. This system with a 2.5V reference allows the LSB bit to be half the size of a 5V reference system.



FIGURE 12. Typical Reference and Supply Circuit

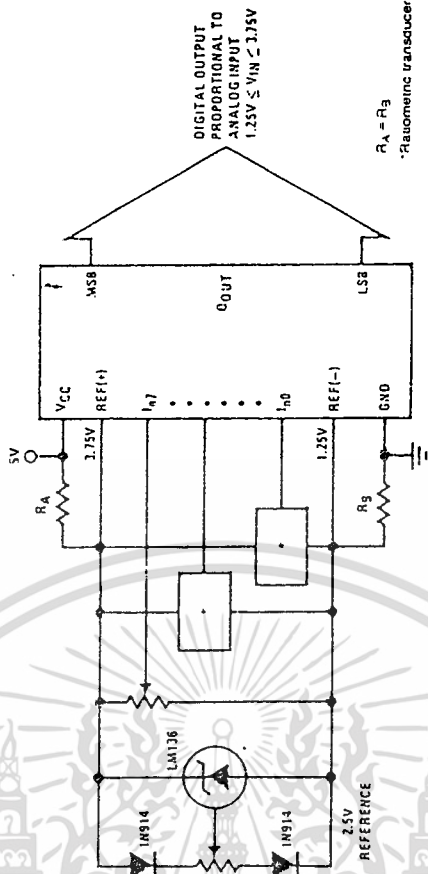


FIGURE 13. Symmetrically Centered Reference

3.0 CONVERTER EQUATIONS

The transition between adjacent codes N and N + 1 is given by:

$$V_{IN} = \left\{ \frac{V_{REF(-)} - V_{REF(+)}}{256 - 312} \right\} \left[\frac{N}{256 - 312} \right] + V_{REF(-)} \quad (2)$$

The center of an output code N is given by:

$$V_{IN} = \left\{ \frac{V_{REF(-)} - V_{REF(+)}}{256} \right\} \left[\frac{N}{256} \right] + V_{REF(-)} \quad (3)$$

The output code N for an arbitrary input are the integers within the range:

$$N = \frac{V_{IN} - V_{REF(-)}}{V_{REF(+)} - V_{REF(-)}} \cdot 256 \pm \text{Absolute Accuracy} \quad (4)$$

where: V_{IN} = Voltage at comparator input

$V_{REF(+)}$ = Voltage at Ref(+)

$V_{REF(-)}$ = Voltage at Ref(-)

V_{TUE} = Total unadjusted error voltage (typically $\pm V_{REF(+)} \cdot 2^{-12}$)

4.0 ANALOG COMPARATOR INPUTS

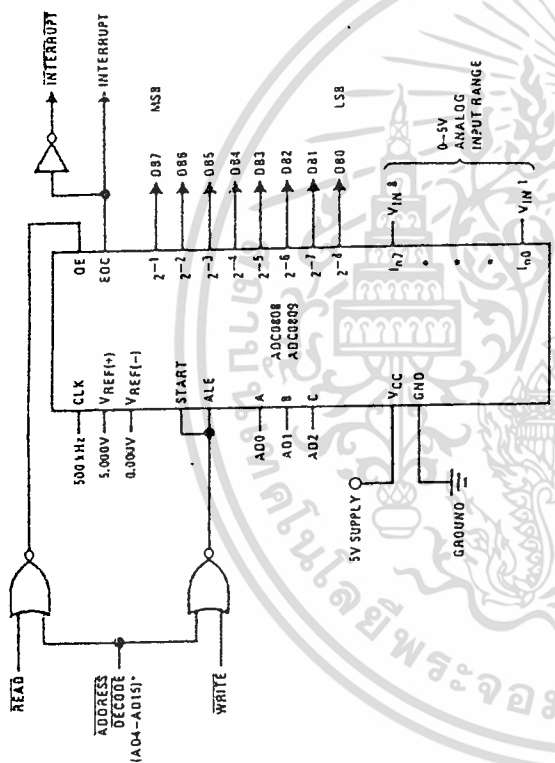
The dynamic comparator input current is caused by the periodic switching of on-chip stray capacitances. These are connected alternately to the output of the resistor ladder/switch tree network and to the comparator input as part of the operation of the chopper stabilized comparator.

The average value of the comparator input current varies directly with clock frequency and with V_{IN} as shown in Figure 6.

If no filter capacitors are used at the analog inputs and the signal source impedances are low, the comparator input current should not introduce converter errors, as the transient created by the capacitance discharge will die out before the comparator output is strobed.

If input filter capacitors are desired for noise reduction and signal conditioning they will tend to average out the dynamic comparator input current; it will then take on the characteristics of a DC bias current whose effect can be predicted conventionally.

Typical Application



TU/M/5672-10

*Address latches needed for 8085 and SC/MP interfacing the ADC0808 to a microprocessor

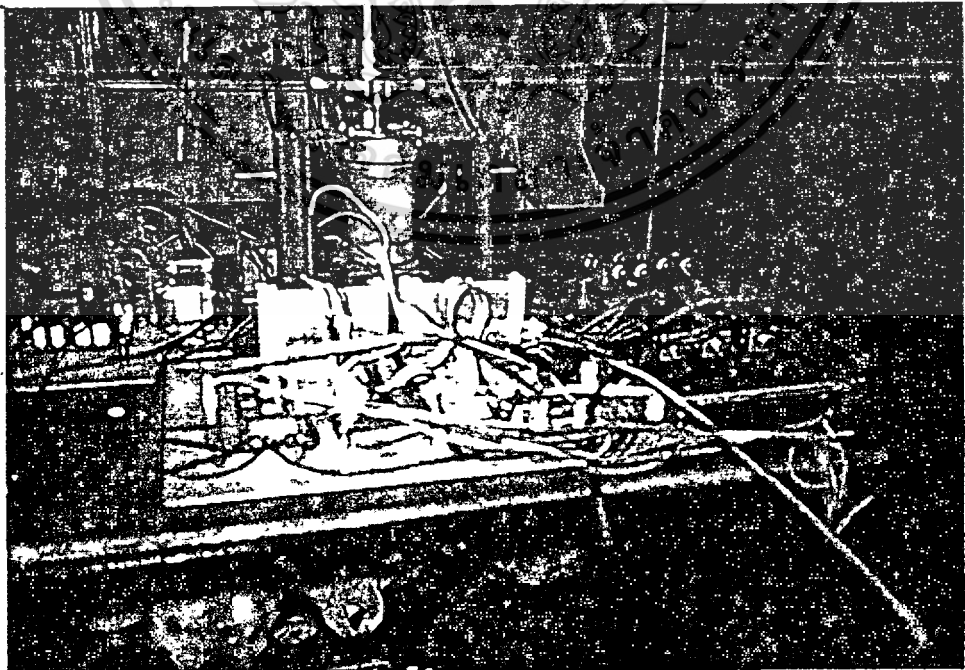
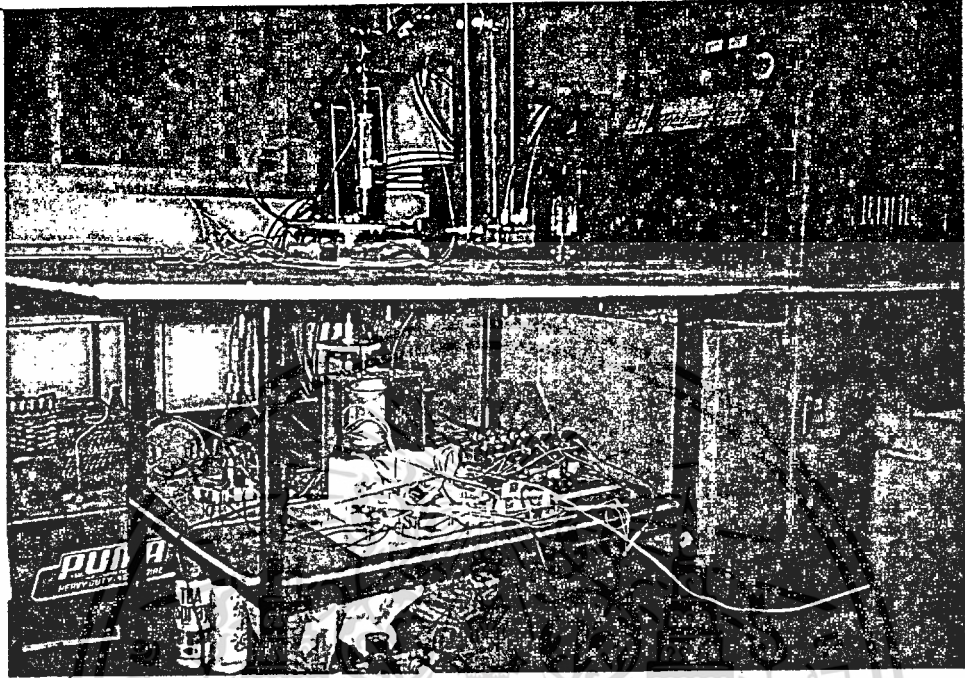
MICROPROCESSOR INTERFACE TABLE

PROCESSOR	READ	WRITE	INTERRUPT (COMMENT)
8080	MEMR	MEMW	INTR (Thru RST Circuit)
8085	RD	WR	INTR (Thru RST Circuit)
Z-80	RD	WR	INT (Thru RST Circuit, Mode 0)
SC/MP	NRDS	NWDS	SA (Thru Sense A)
6800	VMA*φ2-R/W	VMA*φ-R/W	IROA or IROB (Thru P/A)

Ordering Information

TEMPERATURE RANGE	-40°C to +85°C		-55°C to +125°C
Error	± 1/2 LSB Unadjusted	ADC0808CIN	ADC0808CJ
	± 1 LSB Unadjusted	ADC0809CCIN	
Package Outline	N28A Molded Dip	V28A Molded Chip Carrier	J28A Ceramic Dip

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีรูปแสดงแขนกลและวงจรการควบคุมของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. PRINCIPLE OF MEASUREMENT SYSTEMS; JOHN P. BENTLEY
2. นิตยสารเซมิคอนดักเตอร์ ฉบับที่ 102 - 104
3. FUNDAMENTALS OF ROBOTICS ANALYSIS AND CONTROL;
ROBERT J. SCHILLING, PRENTICE HALL INTERNATIONAL, Inc.
4. หนังสือ นวัตกรรมอุตสาหกรรม
ปานเพชร ชินินทร , ภาควิชาเครื่องกล คณะวิศวกรรมเทคโนโลยี
สถาบันเทคโนโลยีราชมงคล วิทยาเขตเทเวศร์
ขวัญชัย สินทร์พิสัยสมบูรณ์ , แผนกวิชาช่างยนต์ คณะวิชาช่างยนต์
สถาบันเทคโนโลยีราชมงคล
วิทยาเขตเทคนิคกรุงเทพฯ
บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
5. รวมบทความ ทฤษฎีและการประยุกต์ใช้งาน อุปกรณ์อิเล็กทรอนิกส์
บริษัท ซีเอ็ดดูเคชั่น จำกัด (มหาชน)
6. การใช้ TURBO C++ เขียนโปรแกรมภาษา C , พ.อ. เจนวิทย์ เหลืองอร่าม, สำนักพิมพ์
สุภาพใจ
7. CP-JR 31 USER'S MANUAL, บริษัท อีทีที จำกัด

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี ก็เพราะว่าได้รับการช่วยเหลือ, ให้การสนับสนุน และให้กำลังใจจากบุคคลต่างๆหลายท่านดังรายนามดังต่อไปนี้

อาจารย์ ชินภัทร นันทจิวารักษ์ และ อาจารย์ จำลอง ปรามแก้ว ซึ่งเป็นผู้ให้คำปรึกษาต่างๆทั้งในด้านทฤษฎีและปฏิบัติ ตลอดจนค่าใช้จ่ายที่จำเป็น พร้อมทั้งยังเปิดโอกาสให้ได้ใช้ความคิดสร้างสรรค์ความรู้ความสามารถได้อย่างเต็มที่

ในการทำโครงการครั้งนี้ นายนพพร นายสกันธ์ และ นายสมบูรณ์ กลุ่มบุคคลซึ่งเป็นผู้ร่วมงานในโครงการครั้งนี้ พร้อมทั้งยังช่วยเหลือและสร้างโครงสร้างทางด้านเครื่องกล เพื่อนำมาใช้ในโครงการครั้งนี้ นายสมปอง นายชัยรัตน์ และนาย ชัยวัฒน์ เพื่อนผู้ซึ่งให้ความช่วยเหลือ และให้คำปรึกษาทางด้านอิเล็กทรอนิกส์

ทางคณะผู้จัดทำขอขอบพระคุณบุคคลต่างๆที่ได้กล่าวมาแล้วในข้างต้นเป็นอย่างยิ่ง ที่ให้การสนับสนุนและช่วยเหลือในด้านต่างๆ และคอยให้กำลังใจในการทำงานตลอดเวลาที่ได้ทำโครงการนี้

ขอขอบคุณเป็นอย่างยิ่ง

คณะผู้จัดทำ