ปีการศึกษา 2533

ระบบควบคุมตำแหน่งและความเร็วของมอเตอร์กระแสตรง

(Control Speed and Position of DC-Servo Motor)

โดย

บุญยง	มานิตย์โชติพิสิฐ	30.1130
ศิษเฎศ	ทองสิมา	30.1278
สถิตย์พร	พงศ์วิโรจน์	30.1286

อาจารย์ที่ปรึกษา

วิทยา ทิพย์สุวรรณพร

027931

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมตำแหน่งและความเร็วของมอเตอร์กระแสตรง

(Control Speed and Position of DC-Servo Motor)

ผู้จัดทำ

1. บุญยง มานิตย์โชติพิสิฐ 30.1130
2. ศิษเฎศ ทองสิมา 30.1278
3. สถิตย์พร พงศ์วิโรจน์ 30.1286

..... อาจารย์ที่ปรึกษา

(อาจารย์ วิทยา ทิพย์สุวรรณเพร)

เลขหมู่ T 33098 ป 2

เลขทะเบียน 027937

วัน, เดือน, ปี 9 ส.ค. 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

027931

ระบบควบคุมตำแหน่งและความเร็วของมอเตอร์กระแสตรง

(Control Speed and Position of DC-Servo Motor)

บุญยง มานิตย์โชติพิสิฐ รหัส 30.1130

ศิษเฎศ ทองสิมา รหัส 30.1278

สถิตย์พร พงศ์วิโรจน์ รหัส 30.1286

อาจารย์ที่ปรึกษา

วิทยา ทิพย์สุวรรณพร

ปีการศึกษา 2533

บทคัดย่อ

การควบคุมความเร็วและตำแหน่งของ ดีซีมอเตอร์ให้ทำงานได้ตามต้องการมีทั้งการใช้การควบคุมแบบอนาลอก และแบบดิจิตอล ในที่นี้เราได้ใช้การควบคุมทั้งแบบอนาลอกและแบบดิจิตอล โดยระบบควบคุมที่เป็นดิจิตอลนั้นจะใช้ควบคุมตำแหน่ง และระบบควบคุมที่เป็นอนาลอกจะใช้ในการควบคุมความเร็วของดีซีมอเตอร์

ไมโครคอนโทรลเลอร์ 8031 ได้ถูกนำมาใช้ในการควบคุมความเร็วและตำแหน่ง หรืออีกนัยหนึ่งไมโครคอนโทรลเลอร์ 8031 จะทำการแปลงข้อมูลที่ใช้ในการกำหนดตำแหน่งและความเร็วไปเป็นสัญญาณควบคุมดีซีมอเตอร์

นอกเหนือจากนี้ ในการควบคุมดีซีมอเตอร์มากกว่าหนึ่งตัวเราจะใช้คอมพิวเตอร์ควบคุมและสั่งงานระบบควบคุมดีซีมอเตอร์ทั้งหมด(ควบคุมไมโครคอนโทรลเลอร์ 8031 แต่ละตัว) เทคนิคนี้เราเรียกว่า มัลติโปรเซสเซอร์(multiprocessor technique)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control Speed and Position of DC-Servo Motor

Boonyong Manitchotipisit No. 30.1130

Sissades Tongsima No. 30.1278

Satitporn Pongviroj No. 30.1286

Vittaya Tipsuwunaporn advisor

Academic Year 1990

Abstract

Nowadays,speed and position controls have two types of controller,analog controller and digital controller. In this report we use both of analog controller to control speed and digital controller to control position.

Single chip 8031 microcontroller used in speed and position controls on the other hand 8031 microcontroller will transfer speed and position code in term of control signals to control DC-Servo motor.

Furthermore, application of DC-Servo motor control more than one unit can use the technique that called multiprocessor technique.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	...1
บทที่ 2 ทฤษฎี	...4
2.1 คุณสมบัติของมอเตอร์แบบแม่เหล็กถาวร	...4
2.2 สมการของมอเตอร์	...8
2.3 ทรานสเฟอร์ฟังก์ชั่น	..11
2.4 ส่วนไดร์ฟเวอร์	..13
2.5 การแก้ปัญหาเดดแบนด์	..15
2.6 ระบบควบคุมแบบดิจิตอล	..18
2.7 การควบคุมความเร็วโดยใช้ระบบควบคุมแบบดิจิตอล	..18
2.8 ส่วนของตัวควบคุม	..20
2.9 หลักการทำงานของตัวควบคุม	..24
2.10 การเปิดหาค่าตารางจากข้อมูล	..26
2.11 การสื่อสารมัลติโปรเซสเซอร์	..27
บทที่ 3 หลักการทำงาน	..32
3.1 หลักการทำงานของส่วนฮาร์ดแวร์	..32
3.2 หลักการทำงานของส่วนซอฟท์แวร์	..34
บทที่ 4 การทดลองและผลการทดลอง	..41
4.1 การทดลอง	..41
4.2 ผลการท์ดลอง	..45
บทที่ 5 บทสรุปและวิจารณ์	..47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

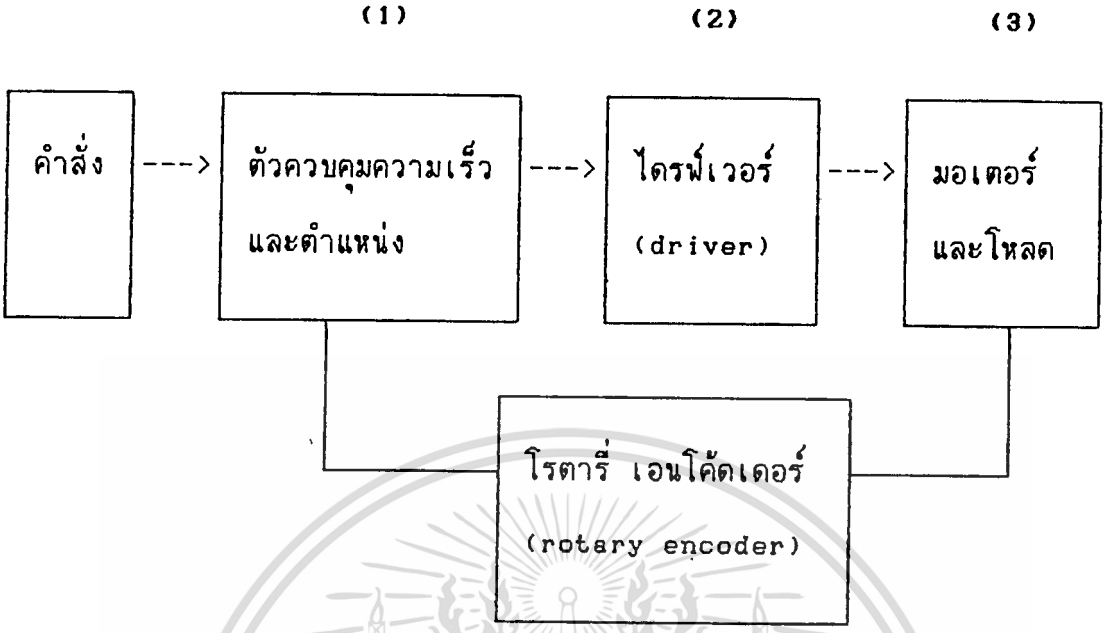
บทนำ

ในปัจจุบันแม้ว่าจะมีการใช้สเตปมอเตอร์ ในงานอุตสาหกรรมกันอย่างแพร่หลาย แต่อย่างไร ก็ดีสเตปมอเตอร์เหล่านั้นก็มีขีดจำกัดทางด้านการใช้งานในกรณีที่ต้องรับโหลดมาก ซึ่งในลักษณะนี้จึงมีผู้หันมาสนใจในดีซีมอเตอร์

การควบคุมดีซีมอเตอร์ในปัจจุบันได้ใช้ความก้าวหน้าทางเทคโนโลยีทางไมโคร-อีเลคทรอนิคส์และไมโครโปรเซสเซอร์ เข้ามาช่วยทำให้การควบคุมมีความแน่นอนและง่ายต่อการใช้งานมากกว่าระบบควบคุมแบบดั้งเดิมที่ใช้พลังจักรกล ซึ่งได้แก่ มอเตอร์ไฟฟ้า ไฮโดรลิกแอกทูเอเตอร์ และพวกเบรคคลัทช์ แต่ด้วยความเจริญก้าวหน้าของเทคโนโลยีการสร้างแม่เหล็กถาวรให้มีคุณภาพสูง ทำให้ดีซีมอเตอร์กลายเป็นพลังจักรกลที่สำคัญในระบบควบคุม

ลักษณะของดีซีมอเตอร์ที่แตกต่างจากสเตปมอเตอร์อย่างเห็นได้ชัด คือดีซีมอเตอร์เป็นอนาลอกแอคทูเอเตอร์ไม่มีตำแหน่งหยุดแน่นอนเหมือนสเตปมอเตอร์ ดังนั้นการควบคุมจึงต้องเป็นการควบคุมแบบลูปปิด(Close loop control). คือตำแหน่งหรือความเร็วทางเอาท์พุท จะถูกป้อนกลับไปเปรียบเทียบกับตำแหน่งอินพุทอ้างอิง(Setpoint) เพื่อให้ได้คุณสมบัติในการทำงาน ซึ่งเป็นลักษณะของระบบเซอร์โวนั่นเอง ซึ่งระบบเซอร์โวโดยทั่วๆไป จะประกอบไปด้วยการควบคุมหลายๆอย่างในระบบเดียวกัน ซึ่งในที่นี่จะเป็นระบบเซอร์โวที่มีการควบคุมตำแหน่ง($\theta(t)$) และความเร็ว (W) ไปพร้อมๆ กัน โดยมีพื้นฐานการควบคุมแสดงดังบล็อคไดอะแกรมต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. ตัวควบคุมความเร็วและตำแหน่ง เป็นส่วนของระบบที่ทำให้เกิดสัญญาณควบคุมไปควบคุมมอเตอร์ให้เกิดการรักษาความเร็วและหยุดได้ตามตำหน่งที่ผู้ใช้ต้องการ

2. ส่วนวงจรไดรฟ์เวอร์ เป็นส่วนประกอบของระบบที่อยู่ระหว่างตัวควบคุม(1) กับ ดีซีมอเตอร์ (3) โดยมีหน้าที่ในการปรับรูปสัญญาณ และขยายสัญญาณให้เหมาะสมก่อนที่จะป้อนไปขับดีซีมอเตอร์และโหลด (3)

3. ดีซีมอเตอร์และโหลด คือระบบที่ถูกควบคุมหรือส่วนที่ออกแรงทำงาน โดยดีซีมอเตอร์ในที่นี้จะเป็นแบบแม่เหล็กถาวร ที่มีคุณสมบัติในการทำงานที่ดี โดยมีอามาร์เจอร์อินดักแตนซ์ (Amature Inductance) และแรงเฉื่อยของตัวโรเตอร์ (Rotor) ต่ำ

4. เอนโค้ดเดอร์ เป็นสิ่งประดิษฐ์ที่ใช้รับรู้หรือตรวจรับสัญญาณเอาท์พุทที่ต้องการโดยจะไม่มีผลของการโหลดดิ้ง(loading) สัญญาณที่ตรวจจับได้นี้จะป้อนกลับไปเปรียบเทียบกับทำให้ได้สัญญาณความผิดพลาด(Error Signal) โดยในที่นี้เราได้ใช้อินครีเมนทอล เอนโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมดีซีมอเตอร์สามารถนำไปประยุกต์ใช้งานในด้านต่างๆ ดังเช่น

- เครื่องจักรกลสำหรับพันคอยล์
- เครื่องจักรกลสำหรับการต่อสาย
- X-Y พล็อทเตอร์
- ระบบควบคุมการตามร่องแสงอาทิตย์ของโซล่าร์เซล
- เครื่องมือแพทย์
- เครื่องจักรกลทางเกษตร
- เครื่องจักรกลทอผ้า
- ระบบควบคุมในเครื่องบิน
- ระบบขับเคลื่อนของกล้องฉายหนัง
- เครื่องมือผลิตสารกึ่งตัวนำ
- เครื่องมือจักรกลที่สามารถตัดเชื่อมเครื่องจักรอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

ทฤษฎีที่จะนำมาใช้ในการอธิบายการทำงานของโครงงานนี้ สามารถแยกการอธิบายออกเป็นเรื่องต่างๆ ซึ่งจะอธิบายเฉพาะเรื่องที่จำเป็นได้ดังต่อไปนี้

2.1 คุณสมบัติของมอเตอร์แบบแม่เหล็กถาวร(PM MOTOR)

2.2 สมการของมอเตอร์

2.3 ทรานส์เฟอร์ฟังก์ชั่น

2.4 ส่วนไดร์ฟเวอร์

2.5 การแก้ปัญหาเดดแบนด์(Dead Band)

2.6 ระบบควบคุมแบบดิจิตอล

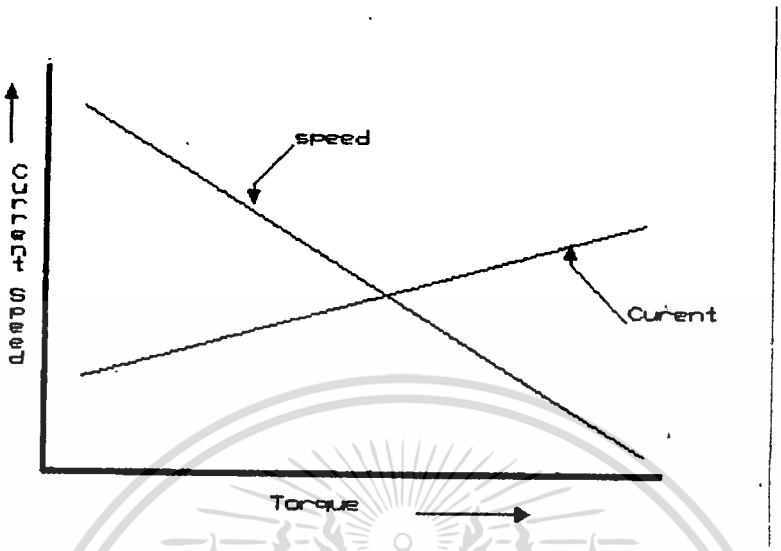
2.7 การควบคุมความเร็วโดยใช้ระบบควบคุมแบบดิจิตอล

2.8 ส่วนของคอนโทรลเลอร์(Controller)

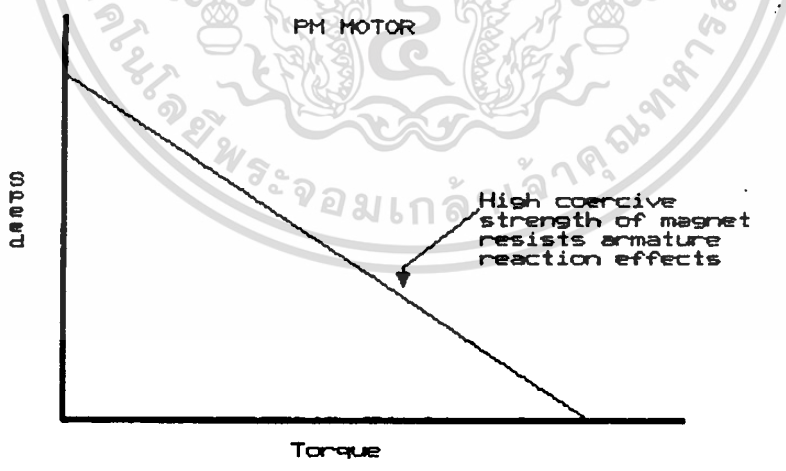
2.1 คุณสมบัติของมอเตอร์แบบแม่เหล็กถาวร(PM MOTOR)

โดยทั่วไปแล้วมอเตอร์แบบแม่เหล็กถาวร(Permanent Magnet Motors) จะมีสนามแม่เหล็กที่เกิดจากแม่เหล็กถาวร ดังนั้นจึงไม่ต้องใช้พลังงานในการสร้างสนามแม่เหล็กที่ฟิลด์คอลย์(Field Coil) และค่าสนามแม่เหล็กที่สเตเตอร์(Stator) จะมีค่าคงที่ตลอด จึงทำให้ความสัมพันธ์ระหว่างความเร็ว กับ แรงบิด มีความสัมพันธ์แบบเชิงเส้นคงที่ตลอดทุกช่วงการทำงาน ดังในรูปที่ 2.1.1 และความเป็นเชิงเส้นที่แรงบิดสูงๆของมอเตอร์แม่เหล็กถาวรจะดีกว่ามอเตอร์ขนานดังในรูปที่ 2.1.2 และ 2.1.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



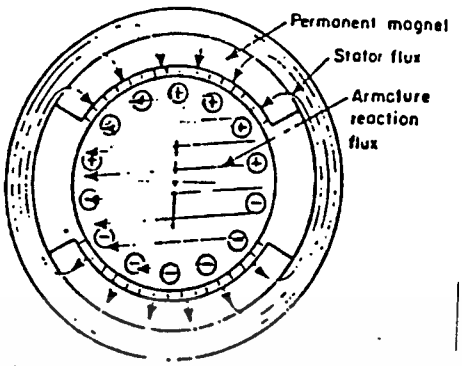
รูปที่ 2.1.1 *กราฟแสดงความสัมพันธ์ระหว่างกระแส-ความเร็วและแรงบิด*



รูปที่ 2.1.2 *กราฟแสดงความสัมพันธ์ระหว่างความเร็วและแรงบิดของมอเตอร์แบบแม่เหล็กถาวร*

จากกราฟคุณสมบัติของมอเตอร์ขนาน จะมีลักษณะมีความสัมพันธ์แบบไม่เชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Due to the coercive strength of the permanent magnet motor, it is virtually insensitive to demagnetization effects.

ข) *มอเตอร์แบบแม่เหล็กถาวร*

รูปที่ 2.1.4 *การเลื่อนของตำแหน่งที่ขั้วแม่เหล็ก*

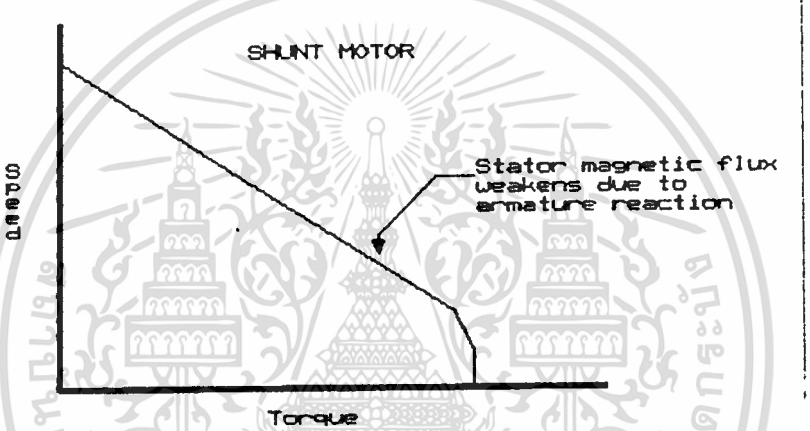
แต่ถ้าเปรียบเทียบกับมอเตอร์แบบแม่เหล็กถาวร ตามรูปที่ 2.1.4 เพราะค่าเปอร์มิอะบิลิตี้(Permeability(μ)) ของวัสดุที่เป็นแม่เหล็กถาวร มีค่าต่ำมาก(เกือบเท่ากับอากาศ) ซึ่งในกรณีนี้จะเกิดแรงโคเออซีฟ(Coercive) สูงๆ มาต้านการเปลี่ยนแปลงของฟลักซ์แม่เหล็ก เมื่อเกิดสนามแม่เหล็กอาร์เมเจอร์รีแอ็คชั่น ซึ่งเป็นผลให้คุณสมบัติของ แรงบิด กับ ความเร็ว มีความสัมพันธ์เชิงเส้น

ข้อดีของมอเตอร์แบบแม่เหล็กถาวรพอสรุปได้ดังนี้

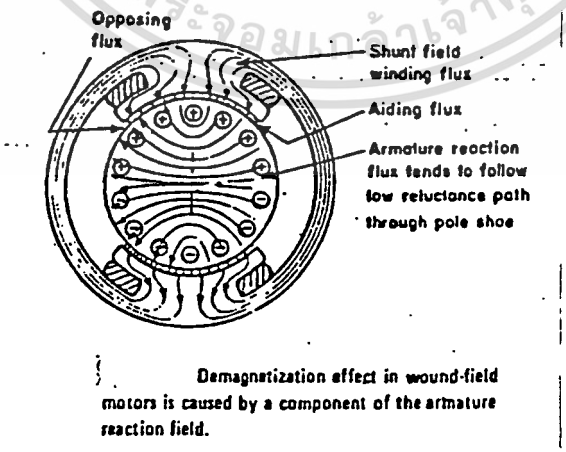
1. คุณสมบัติของ แรงบิด กับ ความเร็ว จะเป็น มีความสัมพันธ์แบบเชิงเส้น
2. การประยุกต์ใช้งานเหมาะกับระบบที่ต้องการแรงบิดของโหลดสูง (เพราะให้ค่าของกระแสอาร์มาเจอร์ ที่สูงกว่าดีซีมอเตอร์ แบบฟิลด์เป็นขดลวด)
3. ไม่มีกำลังสูญเสียในฟิลด์ (ทำให้เกิดความร้อนเนื่องจากกระแสฟิลด์คอลย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นที่แรงบิดที่มีค่าสูงๆ สาเหตุที่เป็นเช่นนั้นเนื่องจากฟลักซ์แม่เหล็กของอาร์เมเจอร์รี-แอ็คชั่น(Amature Reaction) ซึ่งจะวิ่งไปตาม ทิศทางที่มีค่ารีลักแตนซ์ต่ำ ไปยังฐานของขั้วแม่เหล็กและที่ระดับกระแสสูงๆ จะเป็นสาเหตุให้เกิดการเลื่อนของมุมของตำแหน่งที่ขั้วแม่เหล็กติดตั้งอยู่ และทำให้ระดับของฟลักซ์แม่เหล็ก มีค่าลดลง ตามรูปที่ 2.1.4



รูปที่ 2.1.3 *กราฟแสดงความสัมพันธ์ระหว่างความเร็วและแรงบิดของมอเตอร์แบบขนาน*



ก) *มอเตอร์แบบขนาน*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่าเมื่อเปรียบเทียบกับมอเตอร์ ที่มีขนาดแรงม้าเท่ากัน

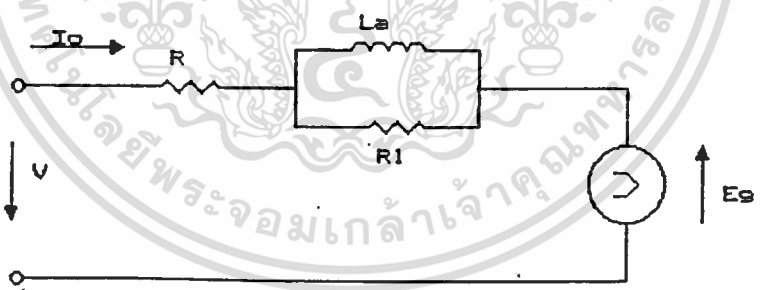
2.2 สมการของมอเตอร์

สมการของมอเตอร์สามารถจำแนกออกได้เป็น 2 ประเภทใหญ่ได้ดังนี้

2.2.1 สมการอีเล็กตริกคอล(Electrical Equations)

2.2.2 สมการไดนามิก(Dynamic Equations)

2.2.1 สมการอีเล็กตริกคอล



รูปที่ 2.2.1 *วงจรแบบจำลองมอเตอร์*

แสดงถึงความสัมพันธ์ของตัวแปรทางอีเล็กตริกคอล คุณสมบัติแรกที่จะพิจารณา คือ อิมพีแดนซ์ของมอเตอร์ซึ่งในการหาจะทำโดยการล็อคมอเตอร์ ให้อยู่ที่ตำแหน่งคงที่ ที่หนึ่งแล้วป้อนค่าความต่างศักย์ ที่เป็น sine wave เข้าไปโดยเปลี่ยนแปลงค่าความถี่ ซึ่งจะทำให้เราสามารถวัดค่ากระแส แล้วนำมาหาค่าของอิมพีแดนซ์ โดยเป็นอัตราส่วนของค่าความต่างศักย์ กับกระแส จะได้ค่าอิมพีดนซ์ อีกส่วนหนึ่งซึ่งเกิดจากการขนานกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของอิมพีเดนซ์(L_a) และความต้านทาน(R_L) อินดิวซ์อีเอ็มเอฟ(Induced emf;E_g) ซึ่งเกิดจากการหมุนของมอเตอร์ ดังนั้นจึงสามารถเขียนวงจรแบบจำลองของมอเตอร์ ได้เป็นดังรูปที่ 2.2.1

ห้องสมุดกลาง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

R_L : คือค่าความสูญเสียในวงจรแม่เหล็กซึ่งปกติจะมีค่ามากกว่า R (ประมาณ 5-10 เท่า) ดังนั้นผลของ R_L บนมอเตอร์จะไม่สำคัญมาก ในทางปฏิบัติเราจะละ R_L ได้ ดังนั้นวงจรแบบจำลองของมอเตอร์ จะเหลือเพียง R, L_a, E_g จากรูปที่ 2.2.1 จะได้ความสัมพันธ์ดังนี้คือ

$$V = L_a \frac{dI_a}{dt} + R \times I_a + E_g \qquad \text{----------(1)}$$

V : ค่าศักดาที่ป้อนให้กับมอเตอร์

I_a : กระแสที่ไหลเข้ามอเตอร์

โดย E_g จะมีค่าแปรผันตรงกับค่า ความเร็วเชิงมุม(w)

$$E_g = K_E \times w \qquad \text{----------(2)}$$

ดังนั้นจากสมการ (1) และ (2) จะได้ว่า

$$V = L_a \frac{dI_a}{dt} + R \times I_a + K_E \times w \qquad \text{----------(3)}$$

ซึ่งสมการนี้เรียกว่า สมการอีเล็กตริกคอลของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

027931

2.2.2 สมการไดนามิค

เนื่องจากค่าสนามแม่เหล็กฟิลด์ในมอเตอร์มีค่าคงที่ ดังนั้น กระแสจึงเป็นสัดส่วนโดยตรงกับ แรงบิด เพราะฉนั้นจะได้ว่า

$$T_m = K_t \times I_a \qquad \text{----------(4)}$$

ดังนั้นจะได้ว่า

$$T_m = T_f + B \times w \qquad \text{----------(5)}$$

T_m : ค่าแรงบิดในมอเตอร์

T_f : ค่าคงที่ของแรงเสียดทานในมอเตอร์

Bxw : ค่าแรงบิดทุกอย่างที่เป็นสัดส่วนโดยตรงกับความเร็วเชิงมุม

ดังนั้นจากสมการ (4) และ (5) จะได้ความสัมพันธ์ระหว่างแรงบิด และความเร็ว คือ

$$T_a = (J_m + J_L)\frac{dw}{dt} + Bxw + T_f + T_L \qquad \text{--------(6)}$$

J_m : ค่าโมเมนท์อินเนอร์เชีย(moment of inertia) ของมอเตอร์

J_L : โมเมนท์อินเนอร์เชียของโหลด

T_L : ค่าแรงบิดของโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งสมการนี้เรียกว่า สมการไดนามิค(Dynamic Equation) ทั้งนี้สมการนี้ตั้งอยู่บนพื้นฐานว่า ความเร็วของมอเตอร์คือความเร็วเดียวกับโหลด

2.3 ทรานเฟอร์ฟังก์ชั่นของมอเตอร์

เราจะหาทรานเฟอร์ฟังกชั่นระหว่างค่าความต่างศักย์ของมอเตอร์และความเร็ว ซึ่งเราจะให้ $T_L=0$, $T_f=0$ เนื่องจากทั้ง 2 อย่างไม่มีผลต่อทรานเฟอร์ฟังก์ชั่น จากสมการ (3) เมื่อแปลงเป็นรูปแบบลาปลาซ จะได้ว่า

$$V(s) = (sL_a + R)\, I_a(s) + K_E x W(s) \quad \text{--------(7)}$$

จากสมการ (4) จะได้

$$T_g(s) = K_T \times I_a(s) \quad \text{----------(8)}$$

จากสมการ (6) จะได้

$$T_g(s) = s(J_m+J_L) \times W(s) \quad \text{----------(9)}$$

เราจะให้โมเมนท์อินเนอร์เชียทั้งหมดเป็น J

$$J = J_m + J_L \quad \text{----------(10)}$$

จากสมการ (8) และ (9) จะได้

$$I_a(s) = ((sJ+B)xW(s))/K_t \quad \text{----------(11)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการ (11) และ (7) จะได้

$$V(s) = \frac{((sL_a+R)(sJ+B)W(s)+K_E xW(s))}{K_T} \quad ----(12)$$

ดังนั้นเราจะได้ว่าทรานเฟอร์ฟังก์ชั่น $G_m(s)$

$$G_m(s) = W(s)/V(s)$$

$$G_m(s) = \frac{K_T}{(SL_a+R)(SJ+B)+K_E K_T} \quad ----------(13)$$

ดังนั้นเราจะจัดฟอร์มจากสมการ (13) ให้อยู่ในรูป

$$G_m(s) = \frac{1/K_E}{(s_1+1)(s_2+1)}$$

$$= \frac{K_T}{JL_a} \frac{1}{(s-p_1)(s-p_2)}$$

t_1 และ t_2 คือ ค่าไทม์คอนสแตนส์(Time Constants)

p_2 และ p_2 คือ โพล(Poles) ของทรานเฟอร์ฟังก์ชั่น

ซึ่ง Poles คือ ค่ารากของสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$s^2 L_a J + s(L_a B + BJ) + RB + K_E K_T = 0$$

เรียกว่า สมการคุณลักษณะ(Characteristic Equation)

และความสัมพันธ์ของโพล กับค่าไทม์คอนสแตนส์ คือ

$$t_1 = -\frac{1}{p_1} \qquad t_2 = -\frac{1}{p_2}$$

2.4 ส่วนไดร์ฟเวอร์

การควบคุมความเร็วของ ดีซีมอเตอร์โดยใช้ ไอซี L292

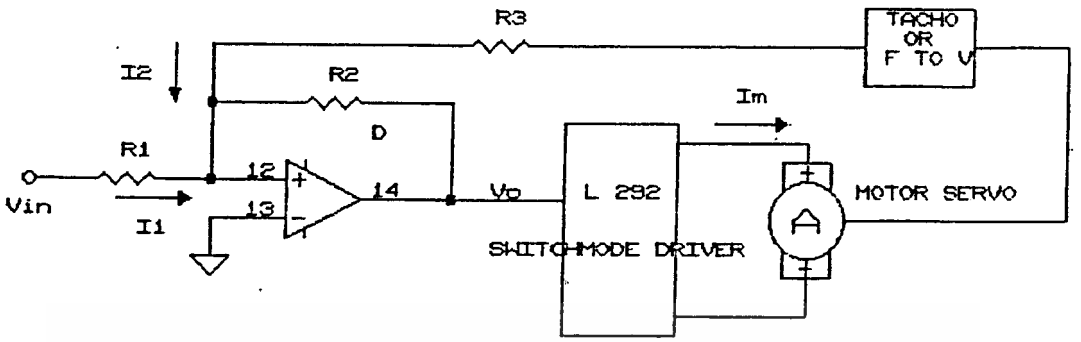
ไอซี L292 เป็นเพาว์เวอร์แอมปลิไฟ แบบพัลซ์วิดทโมดูเลชั่น ใช้ดีซีซัพพลายและแอมปลิไฟเออร์เป็นตัวสวิทชิ่งซัพพลายโวลเตจให้ ON และ OFF ปริมาณของเพาเวอร์ที่ส่งผ่านให้โหลดจะขึ้นอยู่กับสวิทชิ่งเรท(Switching Rate) และอินดักเตนซ์ของโหลดโดยจะทำการเปลี่ยนแปลงดิวตี้ไซเคิลของโวลเตจ เมื่อเปรียบเทียบกับลิเนียร์แอมปลิไฟเออร์ที่ใช้ในระบบควบคุม จะมีปัญหาเกี่ยวกับความร้อนที่เกิดขึ้นในภาคเอาท์พุท ซึ่งสวิทชิ่งแอมปลิไฟเออร์สามารถแก้ปัญหาความร้อนนี้ได้

ในที่นี้ความถี่ของการสวิทชิ่งจะเกิดจาก R และ C ที่ขา 10,11 ของไอซี L292 โดย $f_{osc} = 1/2RC$ (R > 8.2 KOhm) ซึ่งความถี่นี้จะมีค่าในช่วง 1-30 KHz

ในกรณีที่ระบบควบคุมของเราเป็นการควบคุมความเร็วเราสามารถแสดงให้เห็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นไดอะแกรมของวงจรดังรูปที่ 2.4.1



รูปที่ 2.4.1 *ไดอะแกรมการควบคุมความเร็วของไอซี L 292*

โดยกระแส I_1 จะเป็นสัดส่วนกับการตั้งค่า V_{in} (sp) และ I_2 นั้นจะเป็นสัดส่วนความสัมพันธ์ของความเร็วมอเตอร์ ซึ่งกระแสทั้งสองส่วนนี้จะเป็นอินพุทของวงจรซัมมิ่งแอมป์ในที่นี้เราจะเรียกมันว่า เออเรอร์แอมปลิไฟเออร์(Error Amplifier) ที่สภาวะ steady state เราจะได้ว่า $I_1 = -I_2$ กล่าวคือ $V_o = 0$ นั่นเอง และในสภาวะที่ไม่คงที่(transient) เราจะได้ว่า

$$V_o = -R_3*(I_1+I_2)$$

และเนื่องจาก ชิพ ไอซี L292 ทำงานเป็นทรานคอนดักเตนซ์(G_m) ดังนั้น

$$I_m = G_m*V_o$$

ซึ่งกระแส I_m ส่วนนี้จะเป็นตัวขับมอเตอร์ให้เกิดการหมุนที่ความเร็วตามที่ผู้ใช้ต้องการ

<u>การคำนวณค่า R1,R2,R3</u>

กำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

V_{im} : โวลเตจอินพุทที่ใช้ควบคุมความเร็วสูงสุด

n_M : ความเร็วสูงสุดของมอเตอร์

K_g : ค่าคงที่ของทาโคมิเตอร์

ที่ $I_1 = -I_2$; $V_{iM}/R1 = -K_g * n_M/R2$

และค่า R2 นี้เป็นค่าอิมพีเดนซ์ของทาโคมิเตอร์ ซึ่งเราสามารถหาได้จากคุณสมบัติของทาโคมิเตอร์ และจะเห็นได้ว่าเราก็จะสามารถหาค่า R1 ได้อีกเช่นกัน

ส่วนค่า R3 จะเป็นค่าอัตราการขยายของระบบควบคุมแบบ propotional control ซึ่งจะทำให้ระบบตอบสนองไวถ้ากำหนดค่ามากๆ แต่อย่างไรก็ดีการกำหนดอัตราการขยายให้สูงเกินไปจะเป็นผลทำให้เกิดโอเวอร์ชูต ที่จุดสมดุลย์ของระบบ (steady state) ซึ่งในการหาค่าที่ดีที่สุดจำเป็นต้องพิจารณาจากการศึกษาจากทรานเฟอร์พังก์ชั่นของระบบ แล้วทำการหาจุด critical damp

ในกรณีที่เราใช้วงจรแปลงความถี่จาก incremental encoderไปเป็นโวลเตจป้อนกลับมาที่ เออเรอร์แอมปลิไฟเออร์ เราจะต้องใส่ค่า R2 เพื่อจะจำกัดกระแสไว้ค่าๆหนึ่ง ซึ่งเราต้องทดลองหาค่า R2 ที่เหมาะสมพอที่จะตรวจสอบความผิดพลาด I_1 และ I_2 ได้

2.5 <u>การแก้ปัญหาเดดแบนด์</u>

เนื่องจากมอเตอร์จะมีกระแสค้างอยู่ระดับหนึ่งซึ่งเราเรียกว่า current drain กระแสระดับนี้จะเป็นระดับขีดเริ่ม(treshold level) ของการเริ่มหมุนของมอเตอร์ เมื่อมีการป้อนอินพุทเป็นระดับโวลเตจ (V_i) เข้าไป กล่าวคือโวลเตจอินพุทที่เราป้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าไปนั้นจะต้องมีระดับสูงกว่าระดับขีดเริ่มจึงจะทำให้มอเตอร์เริ่มหมุนได้

เมื่อ G_m : ค่าทรานคอนดักเตนซ์ของไอซี L292

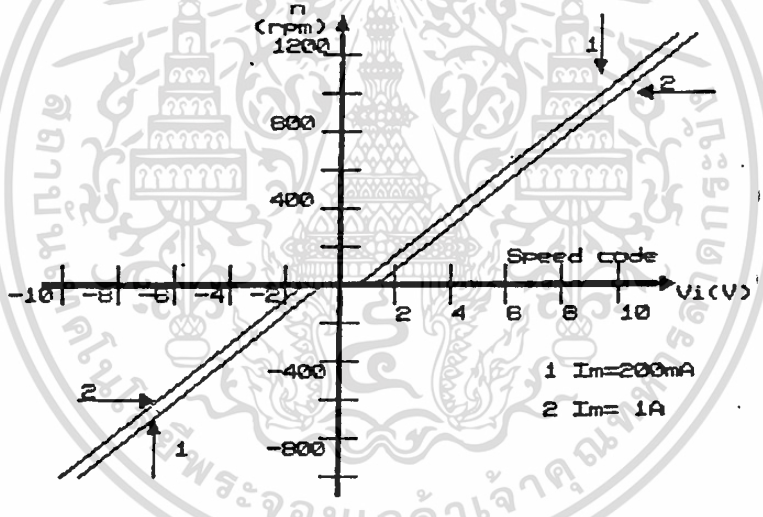
I : คือค่ากระแสในตอนเริ่มต้น

$V_o = I/G_m$ เมื่อ $G_m = 220$ mA/V

เราจะได้ว่าระดับของโวลเตจอินพุทจะหาได้จาก

$$V_i = V_o * R_i / R_s = I * R_i / G_m * R_s$$

จากผลเช่นนี้เมื่อเราพล็อตกราฟจะได้ดังรูปที่ 2.4.2



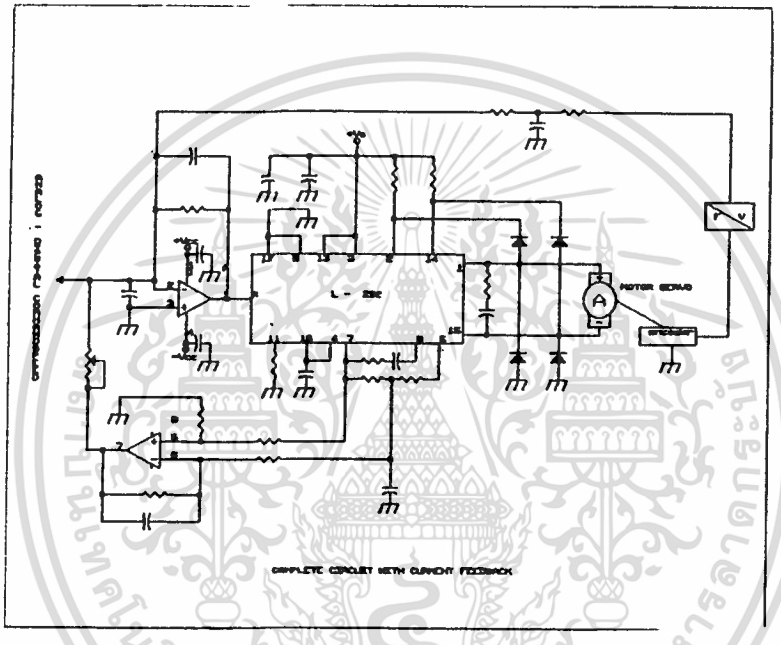
รูปที่ 2.4.2 *กราฟแสดงความสัมพันธ์ระหว่างโวลเตจอินพุทและความเร็วรอบของมอเตอร์*

ลักษณะเช่นนี้จะมีผลดีในกรณีที่เราต้องการให้มอเตอร์หยุดหมุนที่ระดับโวลเตจอินพุทได้อย่างแน่นอน สำหรับการป้อนโวลเตจอินพุทโดยใช้โพเทนชิโอมิเตอร์ ซึ่งนอกจากการใช้ประโยชน์จากเดดแบนด์ นี้แล้วเราสามารถหยุดมอเตอร์โดยการให้ลอจิกศูนย์เข้าที่ขา

13 (inhibit)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

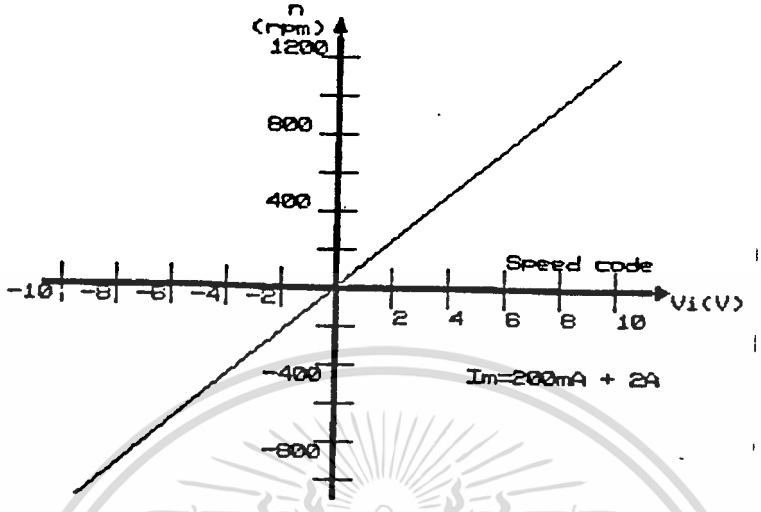
เมื่อเราต้องการแก้ปัญหาที่เกิดขึ้นนี้ แนวความคิดก็คือเอากระแสเดรนจากมอเตอร์ป้อนกลับเข้ามาที่จุดซัมมิ่งของเออเรอร์แอมปลิไฟเออร์ เพื่อจะใช้กระแสส่วนนี้ชดเชยกับกระแสเดรนของมอเตอร์ ซึ่งถ้าทำให้การชดเชยนี้เท่ากัน ก็จะเสมือนกับว่าเราป้อนโวลเตจอินพุทโดยที่ไม่มีระดับเทรชโฮลเกิดขึ้นในระบบ ซึ่งวงจรที่สมบูรณ์เมื่อมีการป้อนกลับเพื่อชดเชย (feedback compensate) เป็นดังรูปที่ 2.4.3



รูปที่ 2.4.3 วงจรสมบูรณ์ที่มีเดอเรนท์ฟีดแบค

และเมื่อเราปรับค่าโพเทนซิโอมิเตอร์ P_2 ในวงจรการป้อนกลับของกระแสเดรนของมอเตอร์ที่ไหลอยู่ระหว่างขา 5 และ 7 ของไอซี L292 ให้มีค่าของการป้อนกลับที่เหมาะสมจะทำให้ความไม่ต่อเนื่องที่เกิดขึ้นรอบจุดออริจิ้นของกราฟระหว่างโวลเตจอินพุทและความเร็วรอบของมอเตอร์ จะลดน้อยลงจนแทบจะไม่มี ดังรูปที่ 2.4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.4 *กราฟแสดงความสัมพันธ์ระหว่างโวลเตจอินพุทและความเร็วรอบของมอเตอร์*

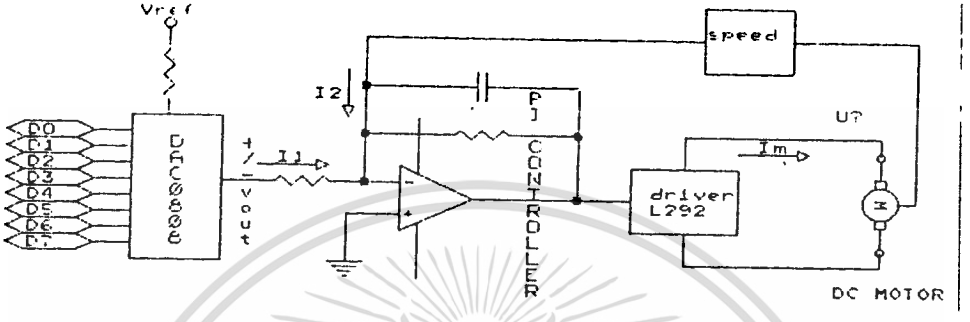
2.6 ระบบการควบคุมแบบดิจิตอล (System with Digital Control)

เนื่องจากความก้าวหน้าทางเทคโนโลยีทางไมโครโปรเซสเซอร์ ทำให้มีการผลิตชิพไอซีที่เป็นไมโครโปรเซสเซอร์ที่รวมเอาระบบของไมโครคอนโทรลเล่อเข้ามาไว้ในชิพไอซีเพียงตัวเดียว และการใช้งานโดยการอินเตอร์เฟสกับภายนอกก็มีเพียงชิพซัพพอร์ทในการแปลงสัญญาณในทางไมโครโปรเซสเซอร์หรือสัญญาณดิจิตอล ให้เป็นสัญญาณอนาลอกนั่นคือระดับของกระแสหรือโวลเตจ ซึ่งมีความละเอียดและรวดเร็วมาก นอกจากนี้ข้อดีของการใช้ระบบควบคุมแบบดิจิตอลก็คือสามารถทำการควบคุมขบวนการที่ซับซ้อน เช่น การใช้มัลติโพรเซสซิ่ง และการควบคุมตัวแปรหลายๆตัวในเวลาเดียวกัน ซึ่งจะประหยัดเวลาใช้งานและแก้ไขงานได้ง่ายกว่าการควบคุมแบบอนาลอก

2.7 การควบคุมความเร็วโดยใช้ระบบควบคุมแบบดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาไดอะแกรมการควบคุมแบบดิจิตอล



รูปที่ 2.7.1 วงจรการควบคุมแบบดิจิตอล

ค่าของกระแส I_1 จะขึ้นอยู่กับค่าของกระแสอ้างอิง I_{ref} และสัญญาณดิจิตอล D_1-D_8

$$I_1 = -I_{ref}*((D_1/2+D_2/4+...+D_7/256)-1/2)$$

และ D_N = "1" ถ้าเป็นระดับสูง

D_N = "0" ถ้าเป็นระดับต่ำ

ในทำนองกันกับระบบควบคุมแบบอนาลอกนั่นก็คือในกรณีที่ไม่มีกระแสเดรน ให้ $I_1 = -I_2$ จะทำให้มอเตอร์หมุนอยู่ในสภาวะคงที่ โดยเป็นสภาวะที่เราต้องให้มีความเร็วของมอเตอร์ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และถ้าต้องการให้ไม่เกิดระดับเทรชโฮลเราก็ทำการป้อนกลับของกระแสเพื่อชดเชยค่ากระแสเดรน

2.8 ส่วนของตัวควบคุม (Controller)

เนื่องจากระบบที่ออกแบบ จะมีการเชื่อมโยงการติดต่อสั่งงานจากคอมพิวเตอร์ (IBM PC/XT) และมีส่วนอินพุทรับค่าของตัวแปรโปรเซส(PROCESS VARIABLE)เข้ามาเพื่อที่จะส่งค่าสัญญาณควบคุมไปยังส่วนควบคุมความเร็วซึ่งจัดรวมกับส่วนไดรฟ์เวอร์ จากลักษณะความต้องการดังกล่าวจึงพิจารณาเลือกใช้ไมโครคอนโทรลเลอร์ 8031 มาใช้งาน

คุณลักษณะเด่นของไมโครคอนโทรลเลอร์ 8031

1. มีวงจรออสซิลเลเตอร์ (Oscillator) และวงจรนาฬิกา (Clock) บนชิพ
2. มีตัวจับเวลา/ตัวนับ (TIMER/COUNTER)
3. มีพอร์ตแบบขนานที่สามารถตั้งค่าแต่ละบิทได้
4. มีพอร์ตอนุกรมที่โปรแกรมอัตราการรับส่งข้อมูล และรูปแบบการรับส่งได้
5. มีอินพุทรับสัญญาณแบบอินเตอร์รัพจากภายนอกได้
6. กำหนดการใช้งานการอินเตอร์รัพของพอร์ตอนุกรม ตัวจับเวลา และตัวนับได้
7. CPU มีความเร็วการทำงานสูง (หนึ่งวัฏจักรคำสั่งใช้เวลา 1 ไมโครวินาที ด้วยการใช้คริสตัล 12 เม็กกะเฮิร์ต)
8. มีหน่วยจำภายใน (internal RAM) ให้ใช้งาน

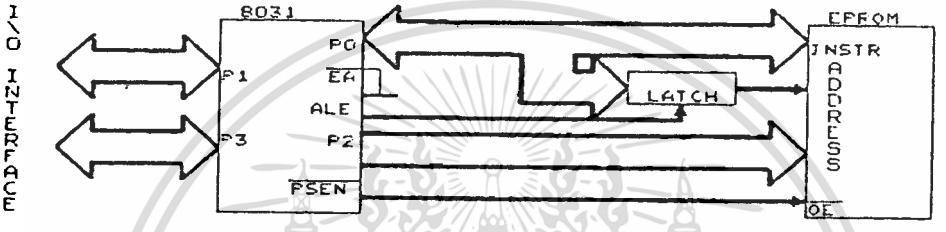
จากคุณสมบัติที่กล่าวมาทำให้ตัวควบคุม(Controller) มีขนาดกระทัดรัดและมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพสูง เพราะได้รวมชิพซัพพอร์ท (Chip support)ไว้ภายในตัวมันเอง การสร้างก็เพียงแค่ต่อ หน่วยความจำโปรแกรมภายนอก (EPROM)ให้มัน

ส่วนประกอบหลักของตัวควบคุม

1. ไมโครคอนโทรลเลอร์ 8031 รวมทั้งหน่วยความจำโปรแกรมภายนอก

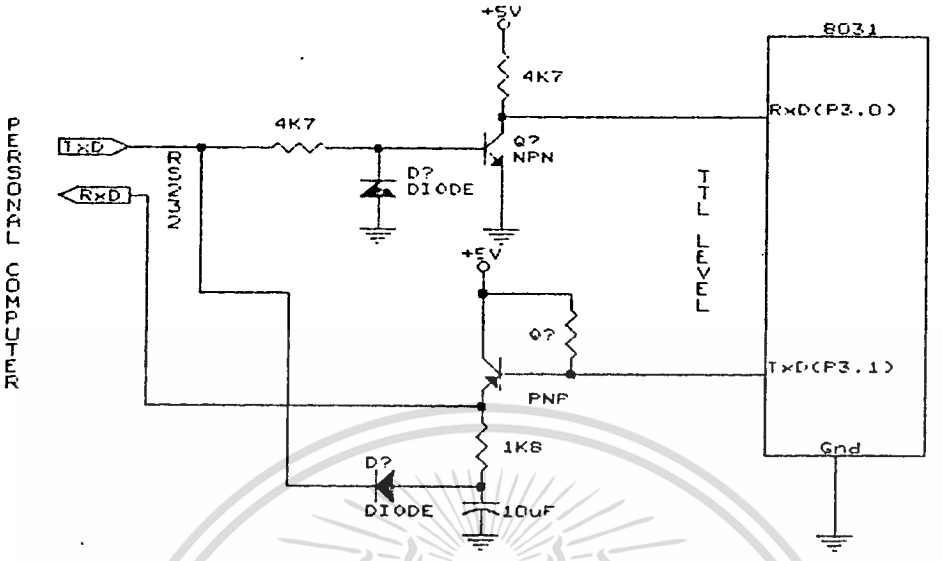


รูปที่ 2.8.1 *รูปแบบการเชื่อมต่อหน่วยความจำโปรแกรมกับ 8031*

ศึกษารายละเอียด และวงจรการใช้งาน 8031 ได้จากหนังสือคู่มือไอซีไมโครโปรเซสเซอร์ MCS-51

2. วงจรเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ 8031 กับคอมพิวเตอร์จะประกอบด้วยส่วนแปลงระดับสัญญาณ TTL (0V,5V) ไปเป็นระดับสัญญาณที่ใช้กับ RS-232 (-12V,+12V) และส่วนแปลงระดับสัญญาณจาก RS-232 เป็นระดับสัญญาณ TTL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8.2 วงจรเชื่อมต่อพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ 8031

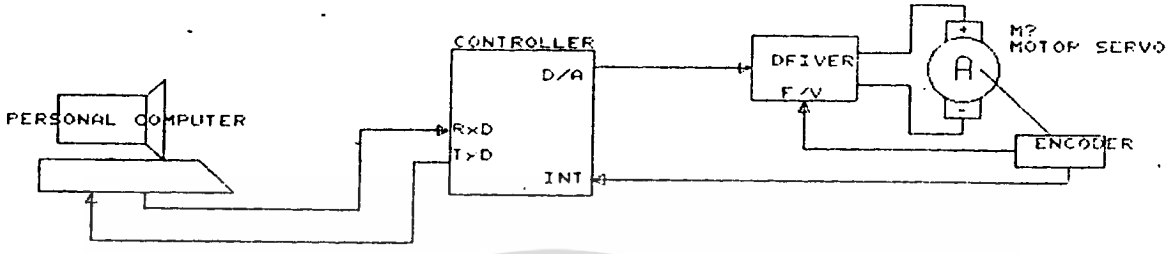
หมายเหตุ เพื่อความสะดวกและลดขั้นตอนในการพัฒนา โครงงานนี้จึงได้ใช้บอร์ดควบคุม CP-32 ของบริษัท ETT ซึ่งเป็นชุดฮาร์ดแวร์ไมโครคอนโทรลเลอร์ 8031 ที่มีวงจรส่วนติดต่อสื่อสารกับคอมพิวเตอร์ทาง RS-232 พอร์ต อยู่พร้อมในบอร์ดเดียวกันแล้ว

3. วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาลอก(D/A converter)

ซึ่งแสดงไว้ในภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณา อินพุท และ เอาท์พุท ของไมโครคอนโทรลเลอร์ 8031 ดังในรูปที่ 2.8.4



รูปที่ 2.8.4 *บล็อกไดอะแกรม อินพุท เอาท์พุท ของไมโครคอนโทรลเลอร์ 8031*

อินพุท

1. อินเตอร์รัพท์ภายนอก (INT0) ใช้ในการรับค่าตัวแปรโปรเซส ในที่นี้ก็คือ พัลส์จากเอนโค้ดเดอร์
2. Receive data (RxD) รับข้อมูลอนุกรมจากคอมพิวเตอร์ซึ่งเป็นระดับสัญญาณที่ใช้กับ RS-232(-12V,+12V) เราจึงต้องผ่านวงจรแปลระดับสัญญาณให้เป็นระดับสัญญาณทีทีแอลเสียก่อน

เอาท์พุท

1. สัญญาณควบคุมที่เป็นดิจิตอลขนาด 8บิต (Speed code) เพื่อนำไปแปลงเป็นสัญญาณอนาลอกสั่งงานชุดควบคุมความเร็ว
2. สัญญาณควบคุมทิศทางการหมุน 1บิต จะเอาไปต่อกับขาควบคุมของอนาลอกสวิทช์เพื่อเลือกทิศทางการหมุนของมอเตอร์
3. Transmitt data (TxD) ส่งข้อมูลอนุกรม ซึ่งเป็นระดับสัญญาณ TTL จึงต้องผ่านวงจรแปลงให้เป็นระดับสัญญาณของ RS-232 คือ +12 V,-12V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 หลักการทำงานของตัวควบคุม

จะเริ่มจากการรับข้อมูลที่คอมพิวเตอร์ส่งมาให้ ซึ่งจะเป็นค่าของตำแหน่ง (ระยะทางที่ต้องการให้เคลื่อนไป) และค่าความเร็วของการเคลื่อนที่นำมาเก็บเข้าหน่วยความจำเพื่อทำการประมวลผลให้มีการส่งค่าทิศทางพร้อมกับ คำสั่งความเร็ว(Command Speed)(ระดับค่าความต่างศักย์ควบคุม) ไปยังส่วน ขับความเร็ว(Speed driver) ค่าคำสั่งความเร็ว จะเพิ่มเป็นขั้นบันไดโดยจะมีการหน่วงเวลาเล็กน้อยก่อนที่จะเปลี่ยนค่าในขั้นต่อไป เพื่อไม่ให้มีโอเวอร์ชูตเกิดขึ้น ขณะเข้าหาค่าความเร็วที่ต้องการในขณะที่เริ่มมีการเคลื่อนที่ ก็จะมีสัญญาณของตัวแปรโปรเซส(จากเอ็นโคดเดอร์) เข้ามาให้ตัวควบคุมตรวจสอบตำแหน่งปัจจุบัน กับค่าตำแหน่งที่ต้องการ (Set point) ได้เป็นค่าค่าความผิดพลาดของตำแหน่ง ขณะที่ค่าความผิดพลาดมีค่ามากตัวควบคุมก็จะส่งคำสั่งความเร็วให้เคลื่อนที่ด้วยความเร็วที่กำหนดไปเรื่อยจนพบว่าค่าความผิดพลาดมีค่าเข้าช่วงที่จะต้องเริ่มมีการควบคุมการเคลื่อนที่ให้หยุดณ.ตำแหน่งที่ต้องการ โดยตัวควบคุมจะลดค่าความเร็วลงด้วยการส่งค่าคำสั่งความเร็วที่เหมาะสมสำหรับที่ค่าความผิดพลาดค่าต่างๆ ซึ่งค่านี้เราเอา (Lookup table method)
โดยเก็บค่าอยู่ในหน่วยความจำเรียงตำแหน่งตามลำดับของค่าความผิดพลาด

กรณีต่างๆ

a : เริ่มจากความเร็วสูงสุดแล้วลดลงมา $V_{(a)} = VT_{(a)}$

b : เริ่มจากความเร็วต่ำกว่าความเร็วสูงสุด $V_{(a)} < VT_{(a)}$

c : ตำแหน่งเป้าหมาย(set point)มีค่าอยู่ในช่วงที่ต้องเริ่มทำการควบคุมตำแหน่ง

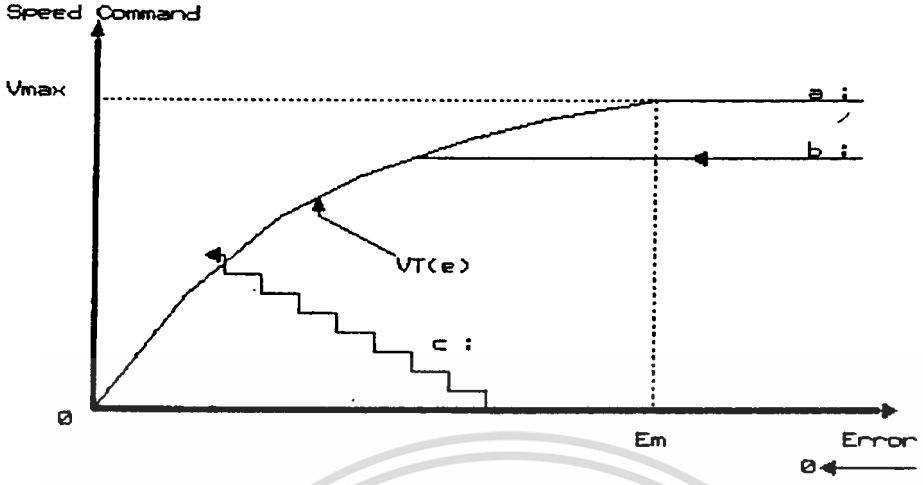
$S_p < E_m$

a : Sp > Em และ $V_{(a)} = VT_{(a)}$

b : Sp > Em และ $V_{(a)} < VT_{(a)}$

c : $S_p < E_m$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 *กราฟแสดงความสัมพันธ์ระหว่างคำสั่งความเร็วและความผิดพลาด*

กำหนด

$VT_{(e)}$: ค่าคำสั่งความเร็ว ในตาราง ณ ค่าความผิดพลาดใดๆ

V_m : ค่าคำสั่งความเร็วสูงสุด(maximum speed command)

E_m : ค่าความผิดพลาดสูงสุดที่เราจะเริ่มทำการควบคุมโดยวิธีการหาค่าจากตาราง(look up table)

$v_{(e)}$: ค่าคำสั่งความเร็ว ณ เวลาปัจจุบัน

S_p : ตำแหน่งเป้าหมายคิดเป็นจำนวนของพัลส์(pulse)ที่จะให้มอเตอร์เคลื่อนที่ไปหา

POS : ตำแหน่งปัจจุบันหรือจำนวนพัลส์ที่นับได้ ณ เวลาปัจจุบัน

err : error = S_p - POS

v : ค่าของความเร็วการเคลื่อนที่ที่ต้องการ(setpoint speed)

เราสามารถสรุปลำดับการทำงานของตัวควบคุมได้ดังโฟล์วชาร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 การเปิดหาค่าจากตารางข้อมูล(look up table)

สามารถกระทำได้โดยนำเอาค่าตำแหน่งของหน่วยความจำที่เริ่มต้นของตารางบวกกับค่าความผิดผลาดได้เป็นค่าของตำแหน่ง ที่จะนำไปชี้เอาข้อมูลที่ต้องการออกมา

ตัวอย่าง

กำหนดตารางเริ่มที่ตำแหน่ง 4000H - 40FFH

โปรแกรมย่อยเรียกตารางเป็นดังนี้

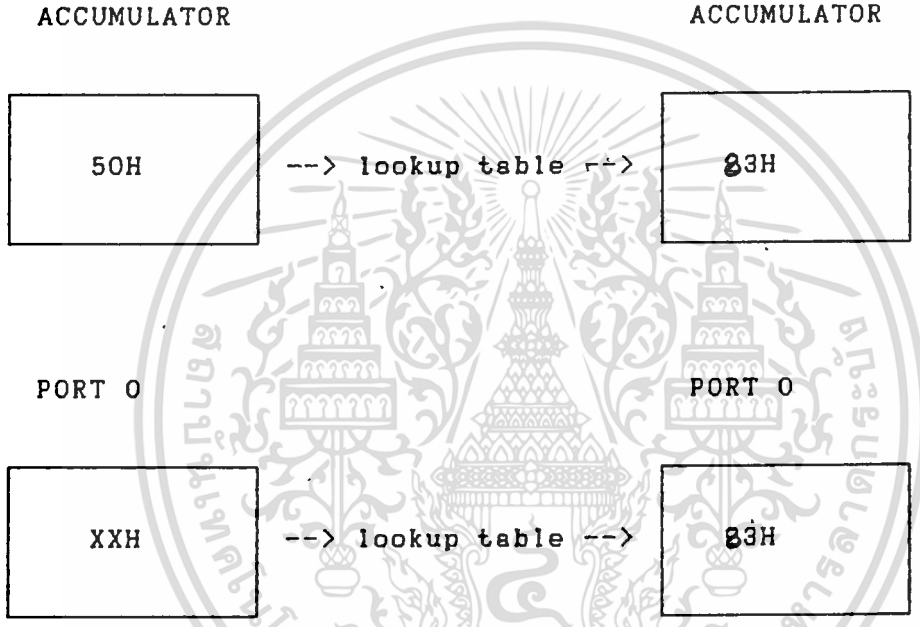
```
TABLE:    MOV  DPTR,#4000H
          XCH  A,DPL
          RET
          TABLE
```

4000H	00H
4001H	01H
.	.
.	.
.	.
.	.
4050H	83H
.	
.	.
.	.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าขณะนั้นค่าความผิดผลาดมีค่าเป็น 50H โปรแกรมการดึงค่าจากตารางจะได้เป็น

```
MOV     A,error         ;error = #50H
LCALL   TABLE
MOV     P0,A            ;P0 ต่อกับ D/A
```

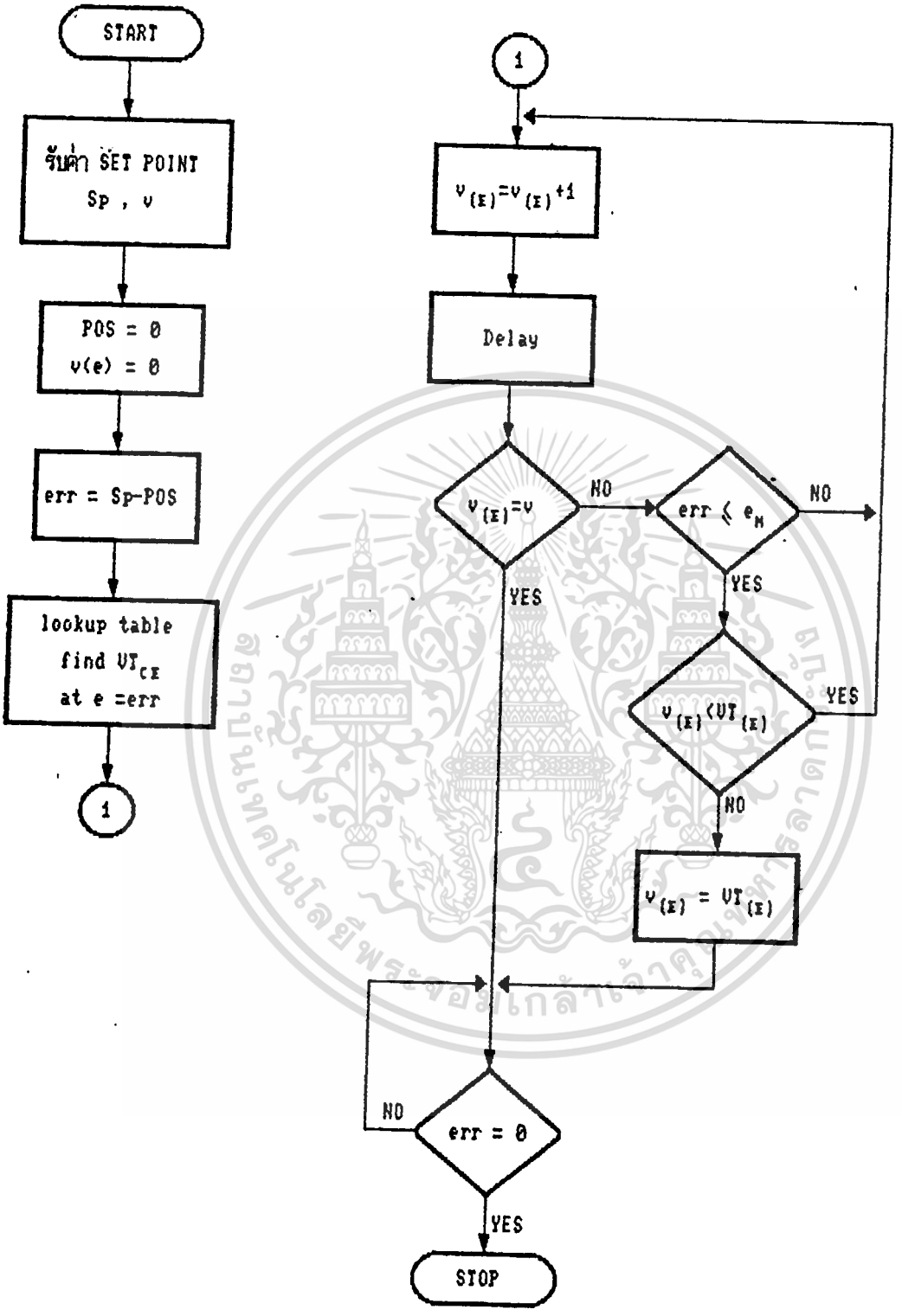


2.11 การสื่อสารมัลติโปรเซสเซอร์ (multiprocessor communication)

การใช้งานพอร์ทอนุกรมของ 8031 ในโหมด 2 และ 3 มีการใช้งานพิเศษสำหรับการสื่อสารทางมัลติโปรเซสเซอร์ ทั้ง 2 โหมดนี้จะมีการรับข้อมูล 9 บิทโดยข้อมูลในบิทที่ 9 จะถูกนำไปเก็บที่ บิท RB8 ใน SCON รีจิสเตอร์ และการสื่อสารจะเป็นลักษณะอินเทอร์รัพมาบอกเพื่อให้สามารถทำงานอื่นได้

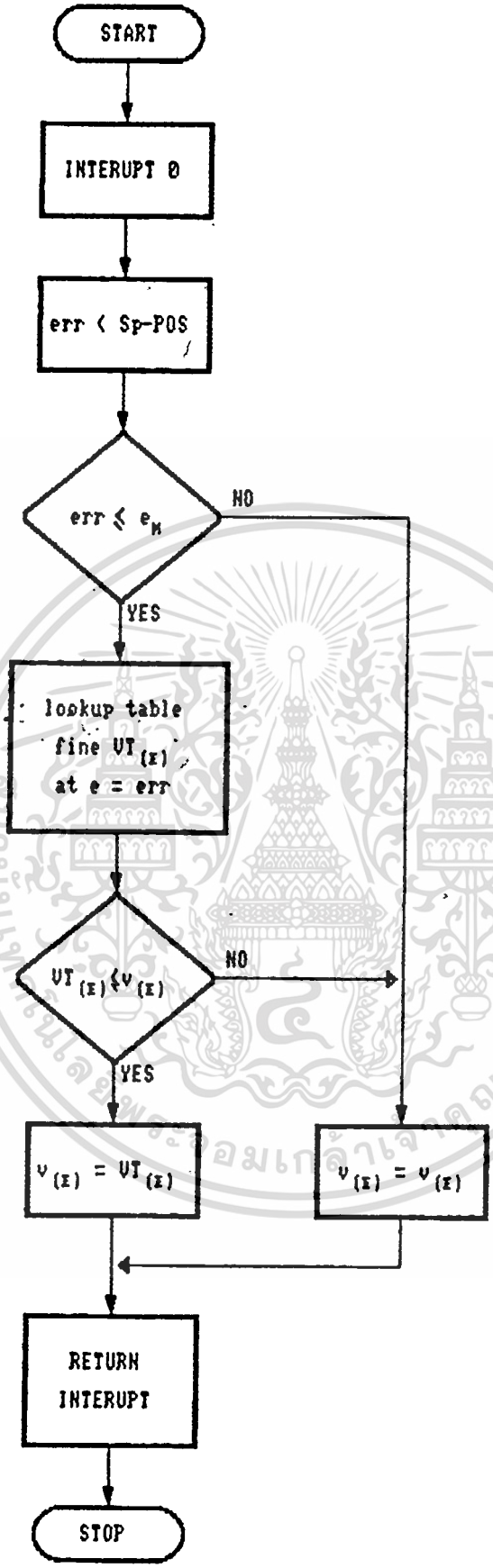
เมื่อตัวโปรเซสเซอร์หลัก(ในงานนี้ใช้ไมโครคอมพิวเตอร์) ต้องการส่งข้อมูลจำนวนหนึ่งไปยังสลาฟโปรเซสเซอร์(slave processor)หรือ ตัวควบคุม โดยทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ส่วนของโปรแกรมการทำงานหลัก(Main Program)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



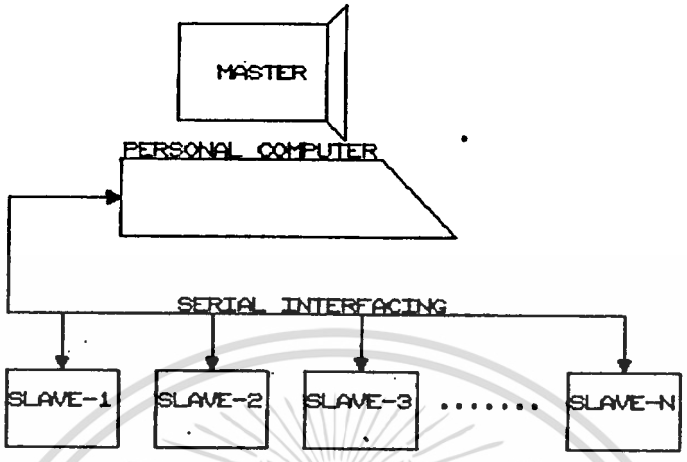
รูปแสดงการทำงานเมื่อเกิดการอินเทอรัพท์เข้ามาโดยพัลส์จากเอนโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มันจะต้องส่งข้อมูลเพื่อเลือกสลาฟที่ต้องการ(address byte) ก่อน เราจะแยกความแตกต่างของข้อมูลที่ใช้เลือกสลาฟ และข้อมูลที่ใช้ในการควบคุมตำแหน่งและความเร็ว (data byte) โดยการกำหนดบิทที่ 9 ของข้อมูลที่ส่งมาถ้าบิทที่ 9 เป็น "1" จะหมายถึงข้อมูลที่ใช้เลือกสลาฟ และถ้าเป็น "0" จะหมายถึง ข้อมูลที่ใช้ในการควบคุมตำแหน่งและความเร็ว

ในตอนเริ่มต้นเราจะต้องกำหนดบิท SM2 ใน SCON รีจิสเตอร์(serial control register) ให้มีค่าเป็น "1" และอนุญาตให้มีการอินเทอรัพท์จากพอร์ตสื่อสารอนุกรมเข้ามาได้ โดยปกติการอินเทอรัพท์จะเกิดขึ้นเมื่อมีการรับ สตอปบิท(stop bit) เข้ามาแล้ว แต่ในการใช้งานแบบมัลติโปรเซสเซอร์ จะมีการตรวจสอบบิท RB8 ว่ามีค่าเป็น "1" หรือไม่ โดยถ้าเป็นจริงจะทำให้เกิดการอินเทอรัพท์ นั่นก็คือสลาฟทุกตัวจะสามารถรับข้อมูลที่ใช้เลือกสลาฟที่ส่งมาได้ ถ้ามันเป็นสลาฟที่ต้องการ (target slave processor)ที่ถูกกำหนดด้วยข้อมูลที่ใช้เลือกสลาฟ มันจะทำการกำหนดค่าในบิท SM2 ให้เป็น "0" เพื่อเตรียมรับข้อมูลที่ใช้ในการควบคุมตำแหน่งและความเร็วที่จะเข้ามาต่อไป สลาฟตัวอื่นที่ไม่ได้ถูกข้อมูลที่ใช้เลือกสลาฟกำหนดจะมีค่าในบิท SM2 เป็น "1" อยู่เหมือนเดิม และยังคงทำงานในส่วนของตัวเองต่อไปเนื่องจากจะไม่เกิดการอินเทอรัพท์ของข้อมูลที่กำหนดตำแหน่งและความเร็วที่ส่งเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 รูปแบบการสื่อสารมัลติโปรเซสเซอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

หลักการทำงาน

หลักการทำงานของโครงงานนี้ แบ่งได้เป็น 2 ส่วนใหญ่ๆ คือ

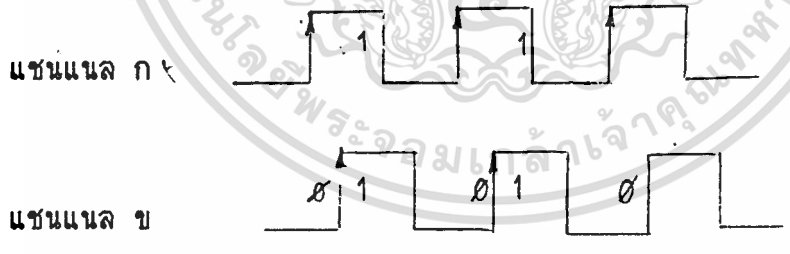
3.1 หลักการทำงานของส่วนฮาร์ดแวร์

3.2 หลักการทำงานของส่วนซอฟแวร์

3.1 หลักการทำงานของส่วนฮาร์ดแวร์

3.1.1 การทำงานของส่วนแปลงสัญญาณจากเอนโคดเดอร์เป็นโวลเตจฟีดแบค

จากเอนโคดเดอร์ของมอเตอร์จะให้สัญญาณพัลซ์สี่เหลี่ยมออกมา 2 แชนแนล ซึ่งมีความต่างเฟสกัน 90 องศา ดังรูป 3.1



รูปที่ 3.1 สัญญาณเอนโคดเดอร์ของมอเตอร์

ซึ่งเราสามารถใช้สัญญาณทั้ง 2 แชนแนบเพื่อบอกการเปลี่ยนแปลงทิศทางของการหมุนของมอเตอร์ได้โดยพิจรณาจากซีเควนเชียลของไดอะแกรม จะเห็นได้ว่าที่ขอบขาขึ้นของแชนแนล ข จะตรงกับลอจิก "1" ของแชนแนล ก และในทำนองเดียวกันที่ของขาขึ้นของแชนแนบ ก ก็จะตรงกับลอจิก "0" ของแชนแนล ข ซึ่งในการ

เอกสารนี้สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ในทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมุนในทิศหนึ่งนั้นจะเป็นไปตามลักษณะนี้ แต่ถ้าหมุนกลับทิศกันจะทำให้สัญญาณของแชนแนล ก จะสลับกับสัญญาณในแชนแนล ข จากหลักการนี้เองเมื่อเราเอาสัญญาณทั้งสองแชนแนลไปผ่าน D Flip-Flop โดยให้แชนแนล ก เป็นขา D และ แชนแนล ข เป็นขาคล็อก (Clock) ในขณะที่หมุนไปทิศทางหนึ่ง ค่าที่ขา Q ของ D Flip-Flop จะเป็นลอจิก "1" ค้างตลอด และเมื่อกลับทิศการหมุนของมอเตอร์ก็จะทำให้ขา Q เป็น "0" ไปในทันที วงจรการตรวจสอบทิศทางการหมุน จะแสดงไว้ในภาคผนวก

ส่วนการแปลงความถี่ของเอนโคดเดอร์ไปเป็นระดับโวลเตจค่าต่างๆนั้นเราได้ใช้ไอซีที่ทำหน้าที่เปลี่ยนความถี่ไปเป็นแรงดัน(frequency to voltage convertor : FVC) ซึ่งสามารถกำหนดการทำงานโดยอุปกรณ์ค่าความต้านทานและค่าเก็บประจุ ซึ่งต่ออยู่ถายนอกชิ้นไอซีที่กล่าวในที่นี้เราใช้ LM 331 แปลงความถี่จากพัลซ์สี่เหลี่ยมของเอนโคดเดอร์(0-5 KHz) ซึ่งการต่อค่า R,C ต่างๆ ดังแสดงวงจรไว้ในภาคผนวก โดยความถี่ทางอินพุทจะไปทำการเปิดปิดสวิทช์ เมื่อเก็บประจุเข้าไปในตัวเก็บประจุ ซึ่งแรงดันทางออกก็คือแรงดันที่ตกคร่อมตัวเก็บประจุ ดังนั้น FVC ก็จะให้ความสัมพันธ์ของแรงดันออกกับความถี่ของสัญญาณทางเข้า

และเนื่องจากแรงดันที่ป้อนกลับไปเปรียบเทียบนั้นต้องมีทิงซีกบวก และซีกลบตามทิศทางการหมุนของมอเตอร์ กล่าวคือเมื่อมอเตอร์หมุนไปในทิศทางหนึ่งจะต้องให้สัญญาณแรงดันที่มีค่าตรงข้ามกับการหมุนในอีกทิศทางหนึ่ง

จากการที่เราได้สัญญาณบอกทิศทางจาก D Flip-Flop และ การเปลี่ยนความถี่จากเอนโคดเดอร์ไปเป็นแรงดันขนาดหนึ่ง 2 ปัจจัยที่มีอยู่นี้เราจิงใช้ไอซีชิพ MC14053 เป็น CMOS ที่ทำหน้าที่เป็นตัวมัลติเพลกเซอร์สัญญาณอนาลอก (Analog Multiplexer) และเป็นตัวดีมัลติเพลกเซอร์(Analof Demultiplexer) ในกรณีหลังนั้นเราได้ใช้คุณสมบัตรของ DMUX ทำให้ได้แรงดันที่ออกมาที่ขาเอาท์พุทของไอซี 2 ขาไม่พร้อมกันโดยถ้าหมุนในทิศทางหนึ่งจะมีแรงดันออกในขาหนึ่งและอีกขาหนึ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหมุนทิศทางตรงกันข้าม ซึ่งเราก็เอาสัญญาณทั้งสองมาผ่าน ออปแอมป์ ซึ่งต่อในลักษณะของ ดิฟแอมป์ ทำให้ได้สัญญาณที่เอาท์พุทมีทั้งซีกบวก และ ซีกลบ รูปของการแปลงสัญญาณดังแสดงในไว้ในภาคผนวก

3.1.2 การทำงานของส่วนอนาลอกคอนโทรลและไดร์ฟเวอร์

3.1.2.1 ส่วนอนาลอกคอนโทรล

เมื่อเราได้ค่าโวลเตจฟิดแบคจากวงจรป้อนกลับเราก็จะนำเอาค่าโวลเตจซึ่งจะเปลี่ยนแปลงไปตามความเร็วของตัวมอเตอร์นั้นป้อนเข้าไปในส่วนของอนาลอกคอนโทรล ซึ่งจะเป็น PI คอนโทรลเลอร์ ในส่วนนี้นั้นออกแบบมาในลักษณะของวงจรซัมมิ่งแอมป์ ซึ่งจะทำการเปรียบเทียบค่าสัญญาณเซ็ทพอยท์กับ ส่วนฟิดแบค เพื่อให้ได้ค่าเออเรอร์ซิกแนลออกมาควบคุมวงจรไดรฟ์เวอร์ให้ขับมอเตอร์ไปที่ความเร็วที่เราต้องการ และเนื่องจากการควบคุมแบบพรอพพอชั่นนอล(propotional) นี้จะทำให้เกิดค่าออฟเซ็ตของความเร็วค่าค่าหนึ่ง นั่นคือความเร็วจากที่ผู้ใช้สั่งไปบอกให้มอเตอร์หมุนอยู่นั้นไม่ใช่ความเร็วที่แท้จริงจึงต้องมีส่วนคอนโทรลโดยใช้ตัวอินทิเกรเตอร์สัญญาณเออเรอร์ และจะมีส่วนของซีโร่ 1/s ที่จะไปคูณกับค่าออฟเซ็ตเมื่อเราเทคลิมิตเข้าใกล้อินฟินิตี้ 1/s = 0 ทำให้พจน์ของออฟเซ็ตที่เราเอาอินทิเกรตไปต่อจะถูกคูณด้วย "0"

อย่างไรก็ดีค่าที่เราใช้เป็นพรอพโพชั่นนอลและอินทิเกรเตอร์ ก็ยังไม่ใช่ค่าที่ดีที่สุด จึงจำเป็นที่จะต้องปรับค่าต่างๆ โดยในที่นี้เราจะใช้ Trimpot 30 KOhm เป็นตัวปรับเกน(Gain) ของระบบคอนโทรลและ Timpot 50 KOhm อีกตัวหนึ่งเป็นตัวปรับค่าของสัญญาณที่ฟิดแบ็คกลับมา

3.1.2.2 ส่วนไดร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังที่กล่าวมาในทฤษฎีแล้วนั่นก็คือส่วนนี้จะเป็นแอมป์ที่จะขยายสัญญาณเออเรอร์ เพื่อไปควบคุมการขับมอเตอร์ โดยตัวขยายนี้จะเป็นในลักษณะของ P.W.M ซึ่งจะทำการเปลี่ยนแปลงค่าดิวตี้ไซเคิลของไฟขนาด 30 โวลท์ ที่จ่ายให้กับตัวไอซี และมีความถี่ของการสวิทชิ่ง 16.7 KHz (ค่าตัวเก็บประจุที่ใช้ในการกำหนดความถี่ของสวทชิ่งเรทคือ 1.8 nF) ทางด้านเอาท์พุทของไอซี L292 เราได้ทำการต่อทรานซิสเตอร์ BDW23B และ BDW24B เพื่อช่วยทำให้วงจรสามารถจ่ายกระแสให้กับมอเตอร์มาถึง 4 แอมป์ได้ ทั้งนี้นั้นจะช่วยทำให้มอเตอร์ที่แรงบิดที่โหลดสูงๆได้ดีขึ้น วงจรการทำงานของไอซี L292 และ ส่วนของอนาลอกคอนโทรลแสดงไว้ในภาคผนวก

3.2 หลักการทำงานส่วนซอฟแวร์

ซอฟแวร์ที่ใช้ในโครงงานนี้ สามารถแยกออกได้เป็น 2 ส่วน ได้แก่

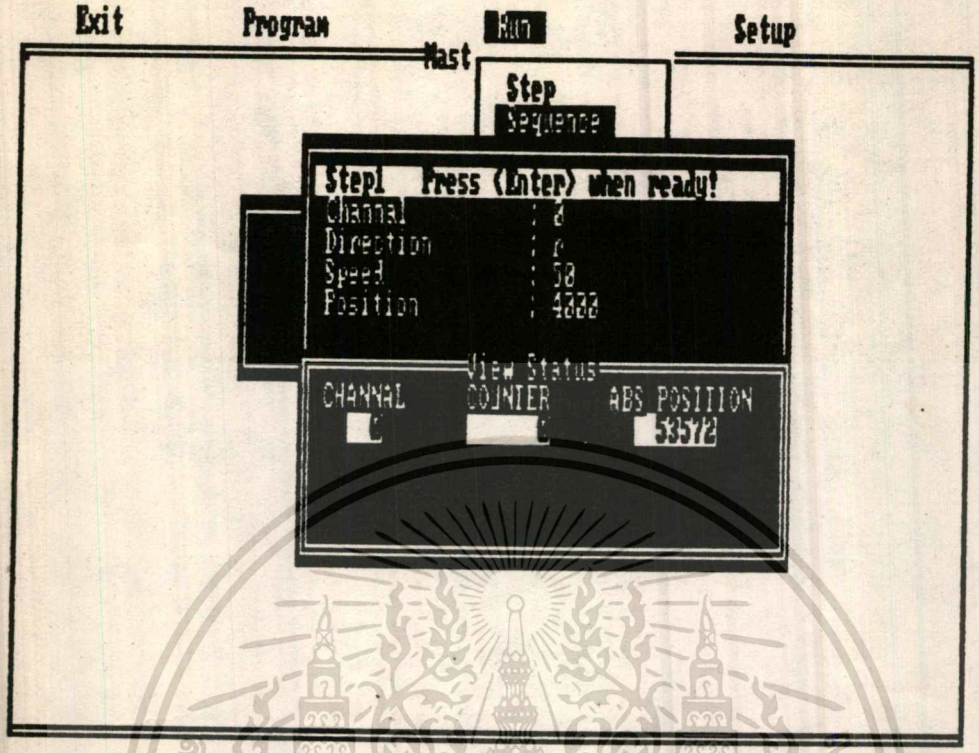
3.2.1 มาสเตอร์(Master) เป็นซอฟแวร์ทำงานบนเครื่องไมโครคอมพิเตอร์

3.2.2 สลาฟ(Slave) เป็นซอฟแวร์ทำงานบน CP-32 board ซึ่งใช้ไมโครโปรเซสเซอร์เบอร์ 8031 เป็นตัวควบคุม

3.2.1 มาสเตอร์ เป็นส่วนที่จะควบคุมงานของระบบทั้งหมดโดยจะส่งค่าเป้าหมาย (Set Point) ผ่านทางพอร์ตสื่อสารอนุกรม RS-232 ให้กับสลาฟรับไปทำการควบคุม

ซอฟแวร์ส่วนนี้เขียนโดยใช้ภาษาปาสคาล ซึ่งสะดวกในการทำงานแสดงผลออกทางจอภาพ และการป้อนเก็บข้อมูลที่ใช้ในการควบคุมต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 *ภาพจากหน้าจอคอมพิวเตอร์ขณะทำงาน*

ลักษณะซอฟแวร์ต้นแบบในโครงงานนี้ประกอบด้วย

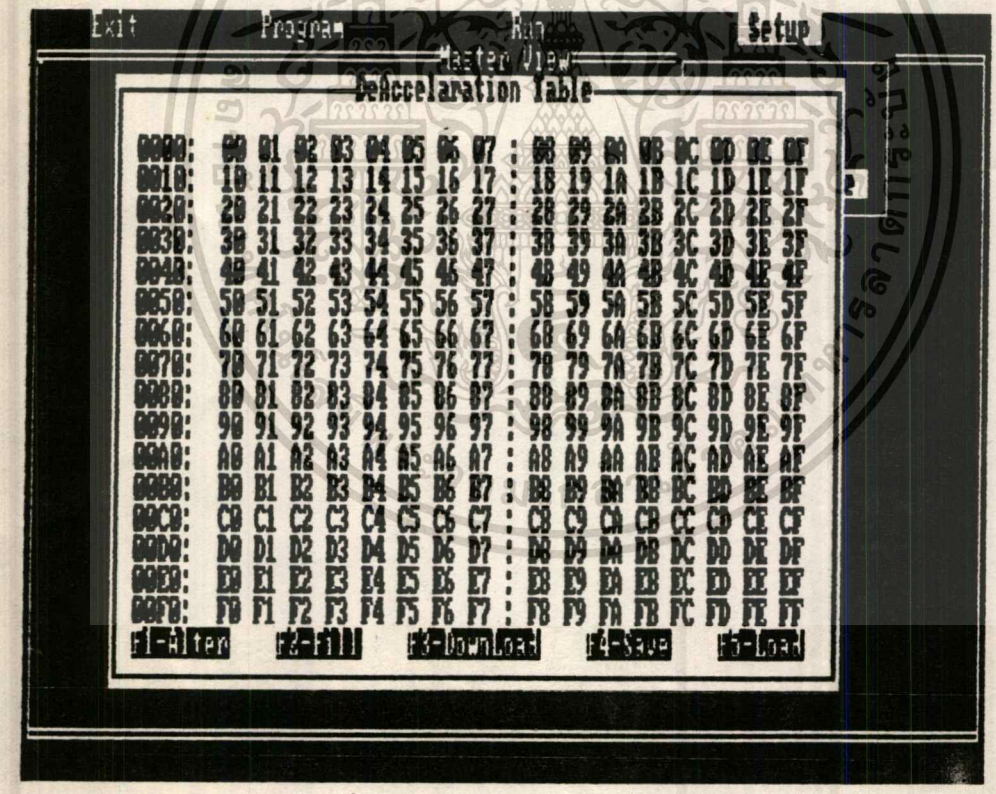
- ระบบพูลดาวน์เมนู เพื่อให้ง่ายในการใช้งาน
- การกำหนดอัตราสื่อสารข้อมูลทางพอร์ตอนุกรมให้เท่ากับ 4800 baud ซึ่งพบว่าค่าที่ใช้งานได้โดยไม่ผิดพลาด
- โปรแกรมลำดับการทำงานของสลาฟได้โดยแต่ละขั้น จะให้ผู้ใช้กำหนดข้อมูลของหมายเลขสลาฟ(Channal) ค่าความเร็ว(Speed) ทิศทาง(Direction) และตำแหน่ง(Position) ที่ต้องการ
- สามารถทำงานได้เป็นแบบลำดับทีละขั้น หรือแบบลำดับต่อเนื่องได้

  <u>กรณีแบบลำดับขั้น</u> จะให้มีการหยุดรอรับคีย์ <RETURN> ก่อนที่จะทำงานในลำดับต่อไปทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>กรณีแบบลำดับต่อเนื่อง</u> การทำงานจะต่อเนื่องไม่หยุดเลยจนถึงลำดับสุดท้าย

- มีการแสดงค่าของตำแหน่งปัจจุบันของสลาฟตัวใดๆตลอดเวลาออกทางจอภาพ โดยวิธีการขออ่านค่า Counter ของสลาฟขึ้นมาโดยมีช่วงการสุ่ม(Sampling) คงที่ไปจนกว่าสลาฟจะทำงานเสร็จคือมอเตอร์หยุดหมุน
- สามารถกำหนดตำแหล่งอ้างอิง(Index Position) นำมาใช้อ้างอิงระยะทางของตำแหน่งที่อยู่ปัจจุบัน(ABSPosition) กับตำแหน่งอ้างอิง ซึ่งจะได้ค่ามีเครื่องหมายเป็นได้ทั้ง บวก และ ลบ
- มีการสร้างเปลี่ยนแปลงตารางข้อมูล(Table) ให้กับสลาฟตัวใดก็ได้ ตารางข้อมูลนี้จะนำไปใช้ในการลดค่าความเร็ว เมื่อเข้าถึงช่วงระยะเบรคก่อนที่จะถึงตำแหน่งเป้าหมาย



รูปที่ 3.7 *ภาพจากหน้าจอขณะแสดงการสร้างตาราง*

- สามารถเปลี่ยนช่วงระยะเบรค(Break Rang) มาก น้อยได้ เพื่อให้มอเตอร์หยุดอย่างนิ่มนวล และถึงตำแหน่งเป้าหมายเร็วที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ ซอฟแวร์ส่วนมาสเตอร์ในโครงงานนี้ สร้างเพื่อใช้ในการศึกษาพัฒนาระบบการควบคุมตำแหน่งตัวซอฟแวร์สามารถเขียนขึ้นโดยไม่ยาก จึงไม่ขอกล่าวรายละเอียดในที่นี้

3.2.2 สลาฟ

รูปแบบของการทำงานที่ได้ออกแบบไว้เป็นดังนี้ เริ่มจากเปิดไฟให้ระบบ จะมีการกำหนดโหมดการทำงานของ 8031 ให้สามารถตอบสนองการอินเตอร์รัพท์จากภายนอก(EXO) และอินเตอร์รัพท์จากพอร์ตสื่อสารอนุกรม(RxD) ได้ และปรับอัตราการสื่อสารเป็น 4800 บอร์ด ให้เท่ากับอัตราสื่อสารของมาสเตอร์ จากนั้นจะอ่านค่าหมายเลขตำแหน่ง(Channal) ของตัวมันเองจากดิฟสวิทช์ที่ตั้งไว้ แล้วส่งค่า 80H ออก D/A เพื่อให้ได้ระดับศักดาที่ใช้ควบคุมเป็นศูนย์โวลท์คือให้มอเตอร์มีความเร็วเท่ากับศูนย์ พร้อมทั้งส่งบิตควบคุมชุดเบรคแม่เหล็กให้จับยึดโรเตอร์ไม่ให้สามารถขยับตัวได้

สลาฟจะคอยตรวจสอบว่ามีการส่งคำสั่งมาจากมาสเตอร์ หรือไม่ ไบท์คำสั่งที่จะส่งมาจะประกอบด้วย 4 บิตล่างเป็นหมายเลขสลาฟ และ 4 บิตบนเป็นหมายเลขคำสั่ง (Command)

COMMAND CHANNAL

กำหนดให้

คำสั่งหมายเลข 1xH หมายถึง การสั่งให้มอเตอร์หมุนได้

4xH หมายถึง จะมีการส่งตารางข้อมูลลงมา ให้สลาฟเตรียมรับทั้งหมด 256ไบท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5xH หมายถึง ให้เตรียมรับข้อมูลค่าเป้าหมาย 5 ไบท์ ได้แก่

ค่าความเร็ว(CMSPD) 1 ไบท์

ทิศทาง(CMDRT) 1 ไบท์

ตำแหน่งเป้าหมาย(CMPOS1,CMPOS2) 2 ไบท์

ช่วงระยะเบรค(EM) 1 ไบท์

6xH หมายถึง ให้สลาฟส่งค่า(Counter) ปัจจุบันไปให้มาสเตอร์เพื่อทำการแสดงผลออกหน้าจอภาพ

มาสเตอร์จะส่งชุดคำสั่งเหล่านี้พร้อมหมายเลขสลาฟเป้าหมาย เรียงลำดับขั้นการทำงานตามที่โปรแกรมไว้ สลาฟเป้าหมายที่ต้องการเท่านั้นที่จะรับคำสั่งไปทำคำสั่งหมายเลข 5xH จะต้องส่งมาก่อน แล้วตามด้วย 1xH เสมอ เพื่อให้มีการปลดเบรคออกแล้วหมุนด้วยความเร็ว และระยะทางตามที่กำหนดหลังจากเสร็จการทำงานสลาฟก็จะรอรับคำสั่งต่อไป

ในโปรแกรมมีการออกแบบให้สามารถปรับความชันของความเร็วที่ลดลงขณะใกล้หยุดให้มากน้อยได้ ขึ้นกับค่าช่วงระยะเบรคที่กำหนด เมื่อเข้าระยะเบรค ความเร็วจะเริ่มลดลงตามค่าจากตารางข้อมูลที่สร้างได้ 256 ไบท์

จะกำหนดว่าระยะจาก err=0 ถึง $err=2^{EM}$ คือ 100%(FFH) และ กรณีกำหนดช่วงระยะเบรคเปลี่ยนไป แต่ใช้ค่าจากตารางเดิม จะแสดงไว้ในภาคผนวก

การ lookup table กระทำโดยการใช้ค่า offset ซึ่งมีค่าเท่ากับ

$$(err/2^{EM})^E \text{ FFH}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ err : Setpoint-Counter

2^{EM} : ช่วงระยะเบรค

ช่วงระยะเบรคจะเป็นลักษณะของ 2 ยกกำลัง n(n=8,9,...) เพื่อความสะดวกและรวดเร็วในการทำการหาร ให้สามารถใช้การเลื่อนบิตไปทางขวา(Shift Right) แทนการเขียนชุดคำสั่งหารข้อมูลที่มีตัวตั้ง 3 ไบท์ ซึ่งจะต้องใช้ชุดคำสั่งยาวและเสียเวลาการคำนวณมากกว่าวิธีการเลื่อนบิตหลายเท่า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลอง

ในการทดลองการควบคุมความเร็วและตำแหน่ง แบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

4.1.1 การควบคุมความเร็วโดยการใช้อนาลอกคอนโทรล

4.1.2 การควบคุมความเร็วและตำแหน่งโดยการนำเอาไมโครคอมพิเตอร์เข้ามาควบคุมด้วย

4.1.1 การควบคุมความเร็วโดยการใช้อนาลอกคอนโทรล

ส่วนของการควบคุมส่วนนี้นั้นจะใช้ภาคไดรฟเวอร์หรือส่วนของ L292 เป็นตัวควบคุมความเร็ว โดยจะมีการส่งคำสั่งควบคุมความเร็วให้กับบอร์ดไดรฟเวอร์ และมีการห้อนกลับดังกล่าวมาในบทที่ 3 แล้ว การควบคุมเราจะตรวจสอบการหมุนทางด้านซ้ายและด้านขวาของมอเตอร์โดยจะจ่ายแรงดันที่เป็นซีกบวกและซีกลบ จากการแบ่งแรงดัน(Voltage Devider) ออกมา เป็นอินพุทของตัวอนาลอกคอนโทรลเลอร์ เพื่อนำเอาค่าสัญญาณที่ได้มาเปรียบเทียบกับภาคฟีดแบคความเร็วของมอเตอร์

4.1.2 การควบคุมความเร็วและตำแหน่งโดยการนำเอาไมโครคอมพิเตอร์เข้ามาควบคุมด้วย

ดังที่กล่าวไปแล้วในบทที่ 3 ว่าการควบคุมความเร็วและตำแหน่งจะต้องมีการตั้งค่าเซ็ตพอยท์ ให้กับระบบที่เราจะทำการควบคุม โดยในที่นี้ผู้ใช้จะทำการป้อนค่าความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เร็วและตำแหน่งที่ต้องการให้หยุดโดยพิจารณาเป็นจำนวนพัลส์ของเอนโคดเดอร์ที่อินเทอรัพท์เข้ามาจำนวน 0-65535 พัลส์

ขั้นตอนและวิธีการใช้งานโปรแกรมมาสเตอร์ซึ่งเป็นซอฟแวร์ใช้งานบนเครื่องไมโครคอมพิเตอร์เพื่อสนับสนุนการทำงานของฮาร์ดแวร์ที่สร้างขึ้น โดยโปรแกรมส่วนนี้จะทำงานร่วมกับโปรแกรมที่ทำงานบน MCS 51 คอนโทรลบอร์ดในลักษณะของการส่งข้อมูลแบบอนุกรมไปให้คอนโทรลบอร์ดแต่ละตัว

สำหรับโปรแกรมนี้ ไฟล์ที่ต้องนำไปใช้งานได้แก่

1. MASTER.EXE เป็นไฟล์โปรแกรมหลัก
2. K.TAB เป็นตารางการชลอความเร็วของการหยุดมอเตอร์

อนึ่ง โปรแกรมจะต้องถูกใช้งานบนดอส(MS-DOS or PC-DOS) ตั้งแต่เวอร์ชั่น 2.0 ขึ้นไป

<u>ขั้นตอนการทดลอง</u>

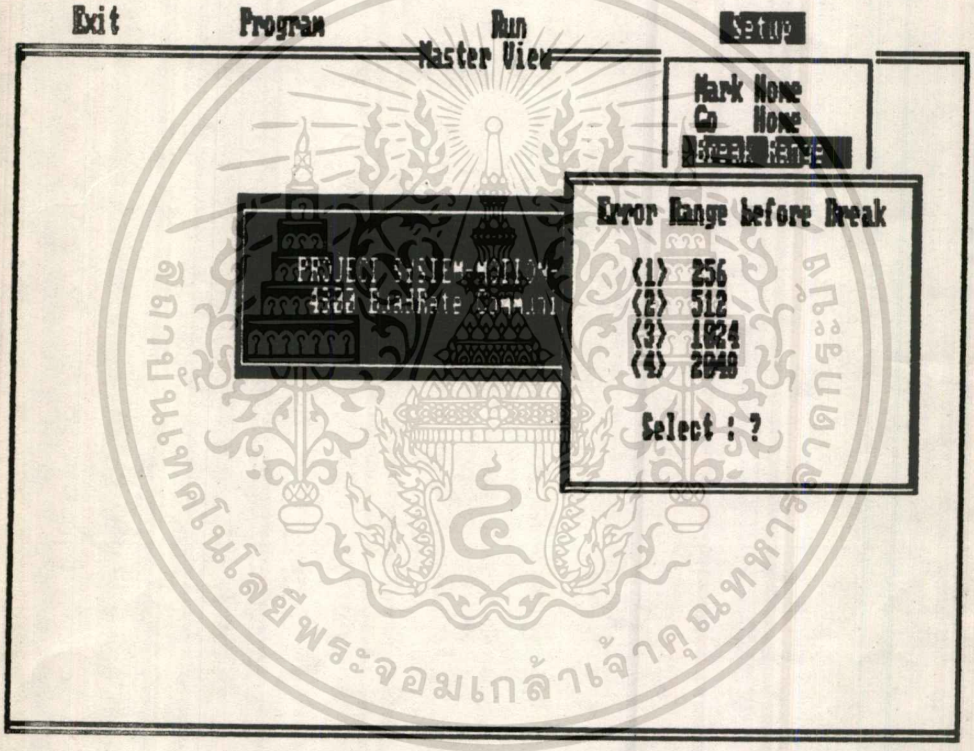
เริ่มแรกต้องทำการเปิดสวิทช์เครื่องภายนอกก่อน แล้วทำการเรียกโปรแกรม MASTER.EXE ซึ่งจะแบ่งการทำงานออกเป็น 4 เมนูหลัก ในรูปที่ 4.1 และมีรายละเอียดดังนี้

1. Exit ใช้เมื่อต้องการออกจากโปรแกรม MASTER.EXE
2. Program เป็นการตั้งค่าคำสั่งความเร็วและตำแหน่งทั้งยังเลือกสลาฟที่จะควบคุม
3. Run ทำการรันโปรแกรมที่เราตั้งค่าสั่งเอาไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Setup แบ่งออกเป็น

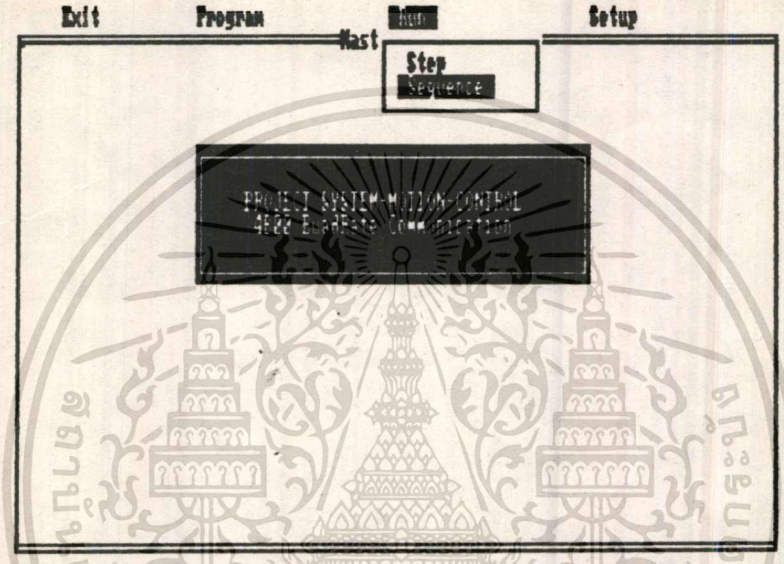
- MARK HOME ตั้งค่าตำแหน่งอ้างอิงขณะนั้น
- BREAK RANGE ตั้งจำนวนตำแหน่งที่จะทำให้เกิดการชลอความเร็ว
- MOTION TABLE ตั้งค่าตารางของการหยุดของมอเตอร์ใช้ร่วมกับ BREAK RANGE



รูปที่ 4.1 *รูปจากหน้าจอคอมพิวเตอร์ของเมนูหลัก*

หลังจากทำการตั้งค่าในเมนู Program และ Setup เรียบร้อยแล้วจะทำการส่งค่าคอมมานด์ไปยังคอนโทรลบอร์ด MCS 51 โดยการเลือกเมนู Run ดังแสดงในรูปที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 รูปจากหน้าจอคอมพิวเตอร์ขณะเลือกเมนู Run

หลังจากทำการตั้งค่าในเมนู Program และ Setup เรียบร้อยแล้วจะทำการส่งค่า คอมมนด์ ไปยังคอนโทรลบอร์ด MCS 51 โดยการเลือกเมนู Run ในเมนูนี้จะส่งข้อมูลแบบอนุกรมไปยังสลาฟเพื่อใช้ในการควบคุมความเร็วและตำแหน่งของมอเตอร์อีกที

หมายเหตุ การส่งค่าต่างๆของคอนโทรลบอร์ดนนั้น ในการทดลองเราได้ใช้มอเตอร์ Super 5 ซึ่งมีระบบเบรคในตัวมันเอง ดังนั้นการส่งคำสั่งต่างๆจึงต้องมีคำสั่งการปลดเบรค และคำสั่งการเอนเนเบิล(Enable) ภาคไดรฟเวอร์ด้วย นอกจากนี้ยังต้องมีวงจรส่วนการลปลดแบรกและทำการอินเทอร์ลอก(Interlock) เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้องกันการจ่ายไฟให้กับมอเตอร์เกิน กรณีที่เกิดการเบรคกระทันหัน โดยยังไม่ได้หยุดจ่ายไฟให้กับมอเตอร์ ซึ่งในวงจรเราได้ใช้วงจรรีเลย์(Relay) และไทม์เมอร์ 555 โมโนสเตเบิลมัลติไวเบรเตอร์ (Monostable Multivibrator) ที่ใช้ในการสั่งการจ่ายไฟ เพื่อทำการปลดเพรคและจำค่านั้นเอาไว้ เทคนิคนี้ทำโดยการใช้ตัวเก็บประจุขนาด 100 uF ให้เกิดการชาร์จ และดีสชาร์จ(Charge & Discharge) เป็นพีค (peak) เพื่อไปปลดเบรค ประมาณ 70 โวลท์ ซึ่งวงจรการปลดเบรคแสดงในไว้ภาคผนวก

4.2 ผลการทดลอง

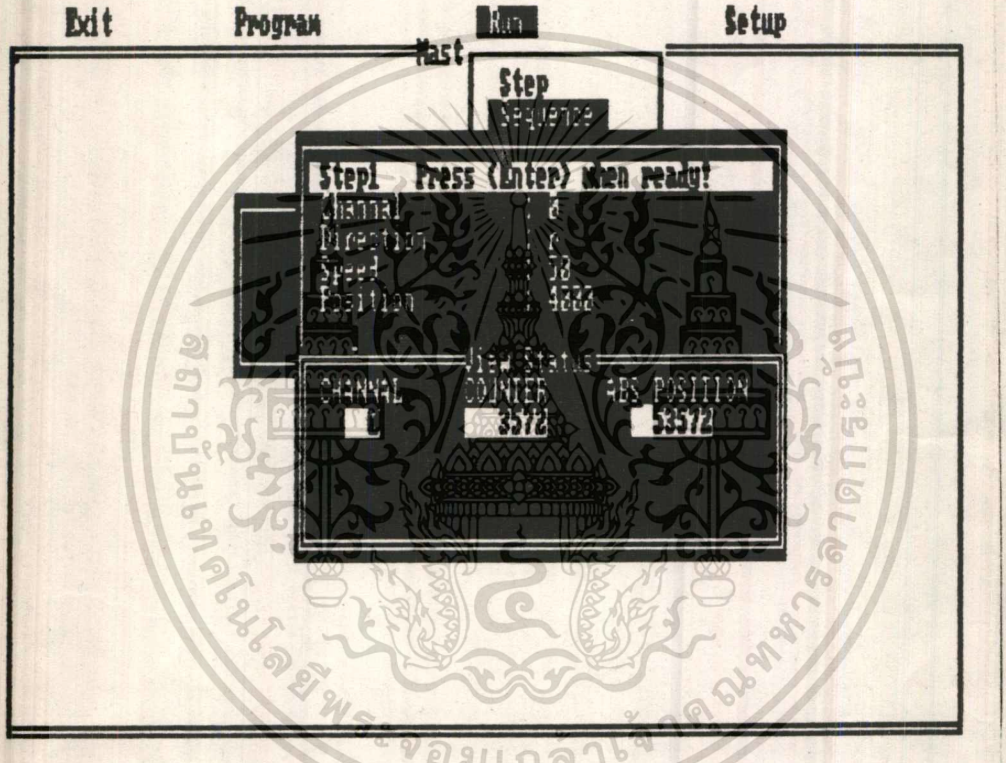
4.2.1 ผลการทดลองของระบบควบคุมความเร็วโดยใช้อนาลอกคอนโทรลนั้น จำเป็นที่จะต้องปรับค่า K_p เป็นเกนนของพรอพโพชั่นนอล คอนโทรลเลอร์(Propotional Controller) ให้สัมพันธ์กับค่า I เป็นอินทิกับไทม์ ซึ่งเราใช้ค่าตัวเก็บประจุ ขนาด 1 nF เป็นตัวอินทิเกรต การปรับใช้วิธีลองผิดลองถูกจนได้ค่าที่เหมาะสม คือมอเตอร์สามารถควบคุมความเร็วได้โดยไม่เกิดการออสซิลเลท(Oscillate) แต่อย่างไรก็ตามการหมุนทางด้านซ้ายและขวาให้ลักษณะการเคลื่อนที่ที่ไม่เท่ากัน คือการเคลื่อนที่ไม่เรียบเท่าที่ควร ทั้งนี้เนนื่องจากแปรงถ่านของมอเตอร์ไม่ดีมีรอยสึกหรอ

4.2.2 ผลการทดลองการควบคุมความเร็วและตำแหน่ง โดยการนำเอาไมโครคอมพิวเตอร์เข้ามาช่วยควบคุมทางด้านตำแหน่งของการหยุดมีการแสดงผลบนจอคอมพิวเตอร์ได้มาจากการอินเทอรัพท์ค่าความถี่ของเอนโคดเดอร์เข้ามา ตำแหน่งที่ตั้งไว้กับตำแหน่งที่หยุดจะมีค่าใกล้เคียงกันผิดพลาด 1 2 พัลส์ อาจเนื่องมาจากโมเมนต์อินเนอเชียของมอเตอร์ แต่เมื่อมีการปป้อนสัญญาณการเบรคเมื่อถึงตำแหน่ง จะทำให้ไม่เกิดความผิดพลาดเลย โดยจะใช้ร่วมกับการชลอความเร็วของตารางการเคลื่อนที่เมื่อใกล้จะถึงตำแหน่งหยุด โดยการเคลื่อนที่จะช้าลงเรื่อยๆ และเมื่อถึงตำแหน่งที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องการให้หยุดก็จะมีคำสั่งความเร็วเป็น "0" พร้อมกับสัญญาณการปล่อยเบรค (Signal Break Active) การแสดงผลบนจอจะไม่มีความผิดพลาด เนื่องจากตำแหน่งเลย

การแสดงผลบนหน้าจอแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 *รูปจากหน้าจอคอมพิวเตอร์ขณะแสดงการควบคุมความเร็วและตำแหน่ง*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

จากการพัฒนาโครงงานนนนี้มาทั้งหมด ทำให้สามารถสรุปคุณสมบัติหลักในการทำงานของชุดควบคุมความเร็วและตำแหน่งของมอเตอร์ได้ดังนี้

1. มีระบบควบคุมหลัก ทำการส่งข้อมูลจากผู้ใช้ไปยังระบบควบคุมย่อย โดยทำการส่งแบบอนุกรมผ่านโปรแกรมที่ทำงานนบนบอร์ดไมโครคอมพิวเตอร์
2. มีระบบควบคุมย่อย คือทำการรับข้อมูลอนุกรมจากโปรแกรมระบบควบคุมหลัก และส่งคำสั่งในรูปของสัญญาณดิจิตอลจากคอนโทรลบอร์ด MCS 51 ผ่านชุดแปลงสัญญาณดิจอตอลเป็นอนาลอก(D/A Converter) ไปควบคุมมอเตอร์ โดยโปรแกรมที่ทำการควบคุมมอเตอร์เก็บอยู่ใน EPROM
3. ระบบการควบคุมความเร็วของมอเตอร์ซึ่งเป็นอนาลอกคอนโทรล โดยจะรับค่าเซ็ตพ้อยท์(Setpoint) จากระบบควบคุมหลักผ่านระบบควบคุมย่อย มายังภาคไดรฟเวอร์

นอกจากนี้ยังมีระบบควบคุมการปลดเบรคของมอเตอร์ในกรณีที่ใช้มอเตอร์ซึ่งมีเบรคอยู่ด้วย ซึ่งจะทำการป้องกันการจ่ายไม่ให้มอเตอร์ขณะที่ยังเบรคอยู่ด้วยรีเลย์(Relay)

สำหรับปัญหาและข้อจำกัดบางอย่างสำหรับโครงงานนี้ รวมทั้งช่องทางการแก้ไขนั้นสามารถอธิบายเฉพาะที่สำคัญได้ดังนี้

1. ปัญหาที่เกิดขึ้นเนื่องจากการหมุนที่ไม่เรียบของมอเตอร์ สามารถแก้ไขโดยการหามอเตอร์ที่มีแปรงถ่านที่ดี หรือทำการเปลี่ยนแปรงถ่านของมอเตอร์ใหม่
2. ข้อจำกัดในการทำงานของโปรแกรมหลักในเรื่องการตั้งค่าจำนวนพัลส์ก่อนเกิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหยุดหมุนและการหน่วงความเร็วก่อนหยุดหมุน เราสามารถทำการทดลองต่อไปเรื่อยๆโดยลองผิดลองถูกหาค่าตารางการหยุด และจำนวนพัลส์ก่อนการหยุดที่เหมาะสมนุ่มนวนมากกว่านี้

3. การแก้ไขโครงสร้างของฮาร์ดแวร์ส่วนป้อนกลับ จะช่วยทำให้ไม่เกิดสัญญาณรบกวนเนื่องมาจากการหมุนของมอเตอร์สร้างพัลส์เทรน (Pulse Train) จากเอนโคดเดอร์ ในส่วนของอนาลอกคอนโทรล ซึ่งจะทำให้การควบคุมมอเตอร์ที่ความเร็วต่ำๆได้ดีขึ้นนมากกว่านี้
4. การเพิ่มขนาดแอมป์ของแหล่งจ่ายไฟตรงจะช่วยทำให้มอเตอร์สามารถขับโหลดได้สูง ซึ่งจะเป็นผลดีในการนำไปประยุกต์ใช้งานต่อไป

แต่ถึงอย่างไรก็ตามการทำงานทั้งหมดก็ถือว่าบรรลุวัตถุประสงค์ ขั้นต้นแล้ว ตามที่ได้สรุปคุณสมบัติไว้ในตอนแรก ซึ่งโครงงานสามารถนำมาพัฒนาให้มีคุณสมบัติการทำงานและการนำไปใช้งานนได้ดีกว่าเดิม โดยแนวทางในการพัฒนาคือ

1. พัฒนาให้สามารถควบคุมสลาฟ หรือมอเตอร์ได้หลายตัวมากขึ้นกว่านี้
2. พัฒนาการใช้งานมอเตอร์ให้มีความสัมพันธ์ของการเคลื่อนที่ของมอเตอร์ เช่น การเคลื่อนที่ของพล็อตเตอร์ตั้งแต่ 2 แกนขึ้นไป
3. พัฒนาโดยการนำไปประยุกต์ใช้งานเฉพาะอย่างดังกล่าวในบทนำเป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานระบบควบคุมความเร็วและตำแหน่งของดีซีมอเตอร์ และปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงได้ด้วยดีได้นั้น เนื่องจากการได้รับการสนับสนุนช่วยเหลือจากหลายฝ่ายด้วยกัน โดยการให้ข้อมูล อุปกรณ์ ซอฟแวร์สนับสนุน คำแนะนำและกำลังใจ ซึ่งเป็นประโยชน์ยิ่งต่อการดำเนินงาน

ขอขอบคุณท่านอาจารย์ที่ปรึกษา อาจารย์ วิทยา ทิพย์สุวรรณพร ที่ได้กรุณาช่วยเหลือ และสนับสนุนทางด้านอุปกรณ์เครื่องมือและข้อมูลค้นคว้าหลายๆอย่างเป็นอย่างดี

นอกจากนั้นขอขอบคุณ

* โครงงาน Automatic Telephone Information ที่ได้สนับสนุนให้คำแนะนำเกี่ยวกับซอฟแวร์ของโครงงานในระบบหลัก

* คุณ ณัฐพล มงคลประดิษฐ์ ที่ได้ให้คำแนะนำเกี่ยวกับเทคนิคการเขียนโปรแกรมการทำงานของระบบ และยังให้ความช่วยเหลือเกี่ยวกับปริญญานิพนธ์ฉบับนี้จนลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณ อาจารย์และเพื่อนๆ ทุกท่านที่ได้ให้การสนับสนุนทางด้านอุปกรณ์บางอย่างและให้ใช้สถานที่ในการทำโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. Takashi Kenjo, Power Electronics for the Microprocessor Age, Oxforde University Press., 1990

2. Adel S. Sedra Kenneth C. Smith, Microoelectronic Circuits, Holt-Saunders International Editions., 1982

3. Electro-Craft Co., DC Motor Speed Control and Servo System, Engineering Handbook, 3rd edition., 1975

4. Intel, Microprocessor Hand Book MCS-51 Microcontrollers

5. Intel, MCS Basic-52 User's Manual

6. โยธิน เปรมปราณีรัชต์, ระบบเซอร์โว และอิเล็คโทรนิคส์คอนโทรลมอเตอร์, ตำราชุดวศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, พิมพ์ครั้งที่ 1, 2533

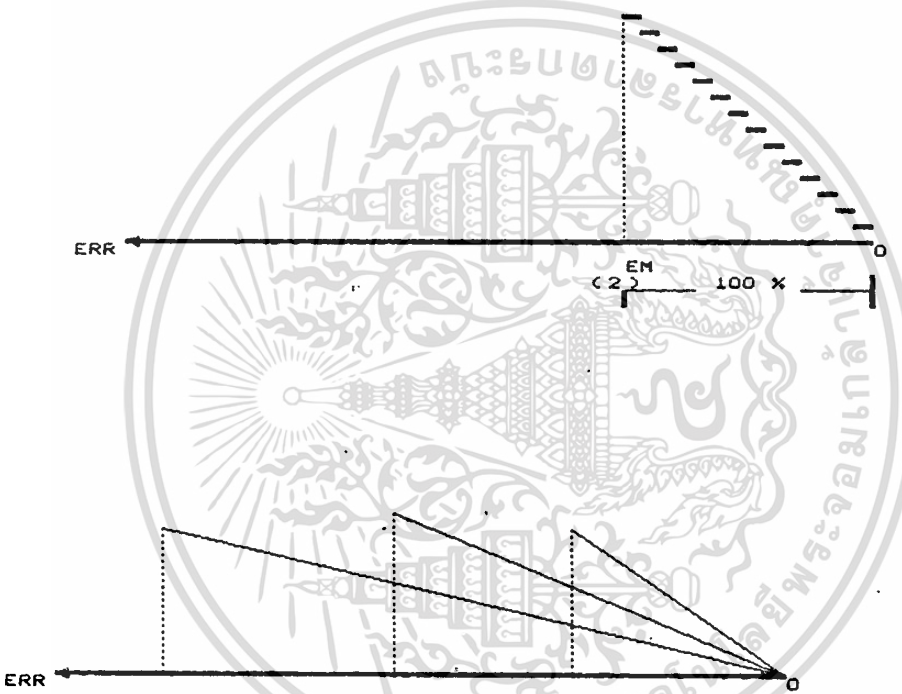
7. บุญเลิศ เอี่ยมทัศนา, คู่มือเทอร์โบปาสคาล รุ่น 4.0-5.0, พิมพ์ครั้งที่ 2, พ.ศ. 2532

8. สุรศักดิ์ สงวนพงษ์, แอดวานซ์เทอร์โบปาสคาล version 4.0, ซีเอ็ดยูเคชั่น, พิมพ์ครั้งที่ 1, 2531

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

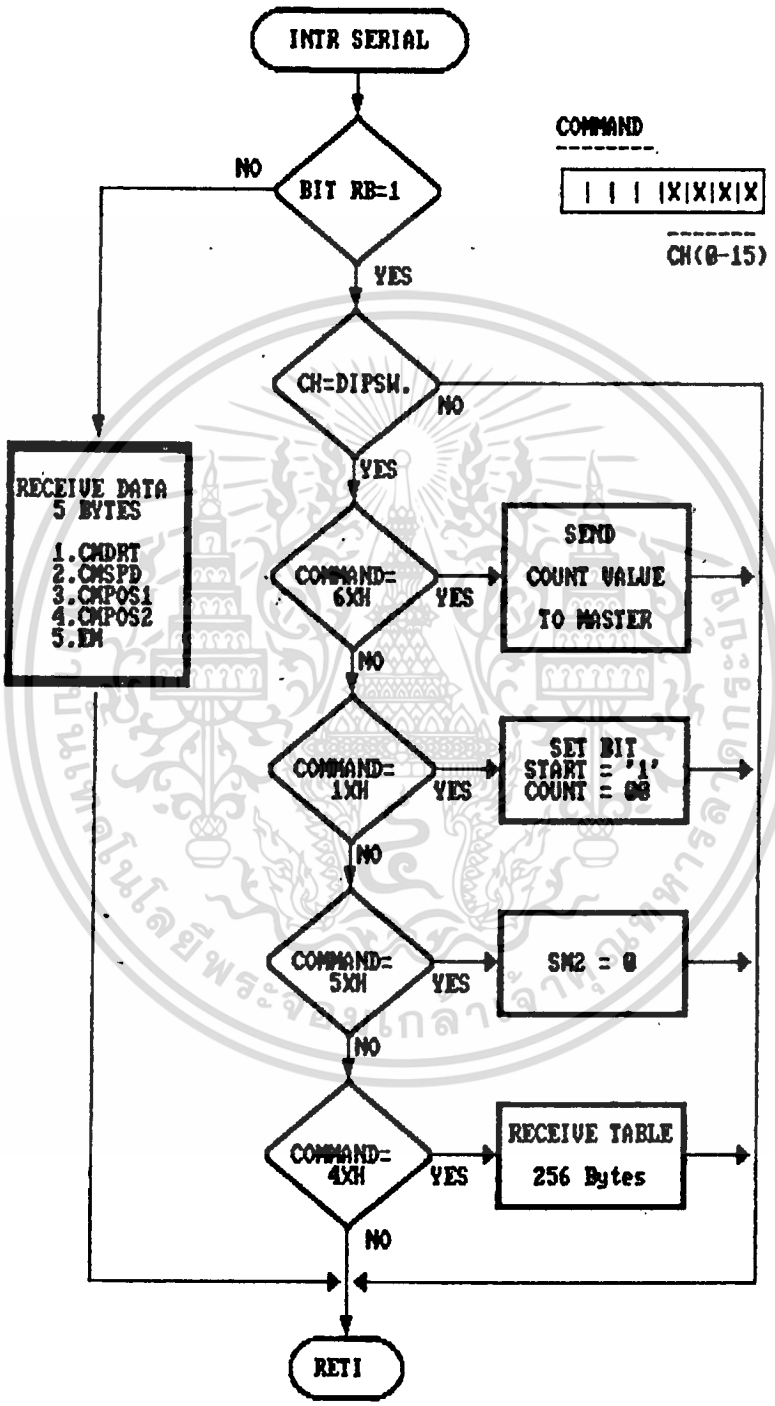
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



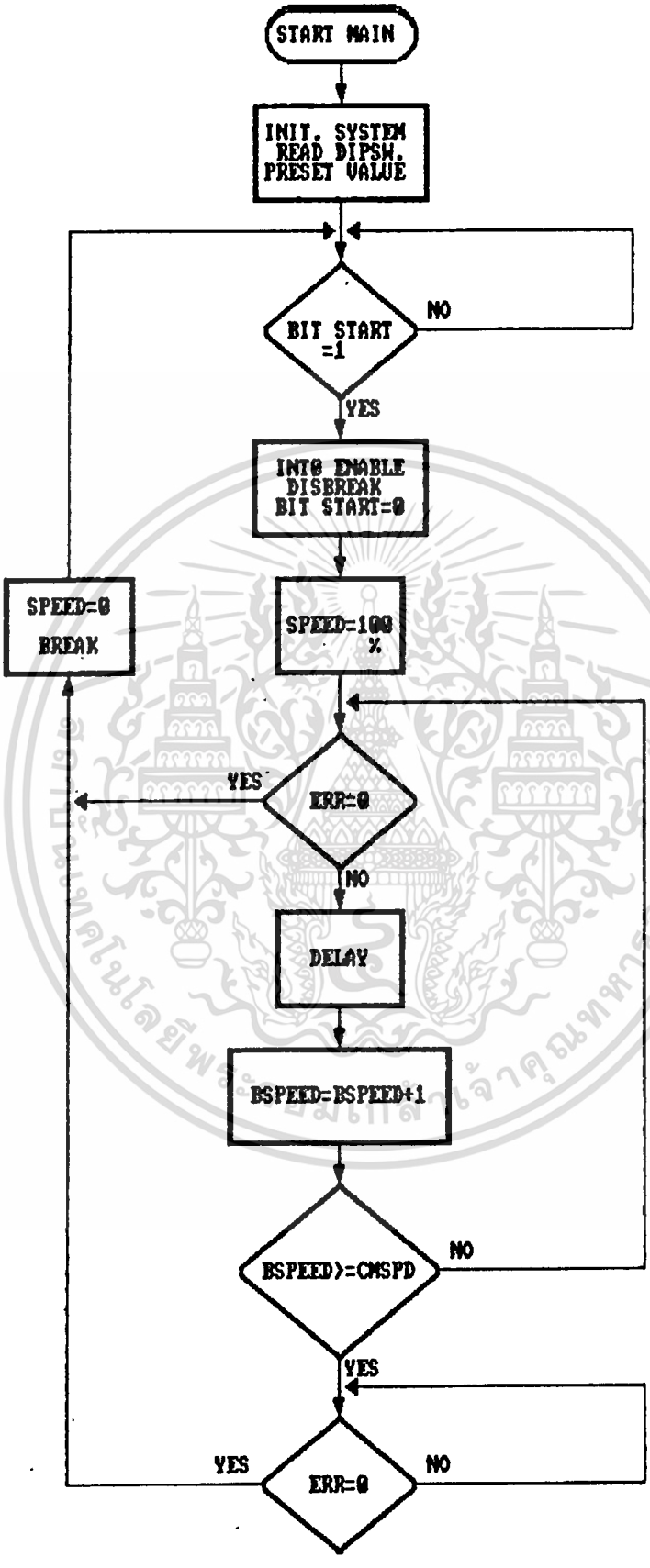
Size	Document Number	REV
A		
Date: March 26, 1991	Sheet of	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า บริษัทฯ ขอสงวนสิทธิ์ในการดำเนินคดีตามกฎหมายกับผู้ที่นำเอกสารนี้ไปใช้ในทางที่ผิด

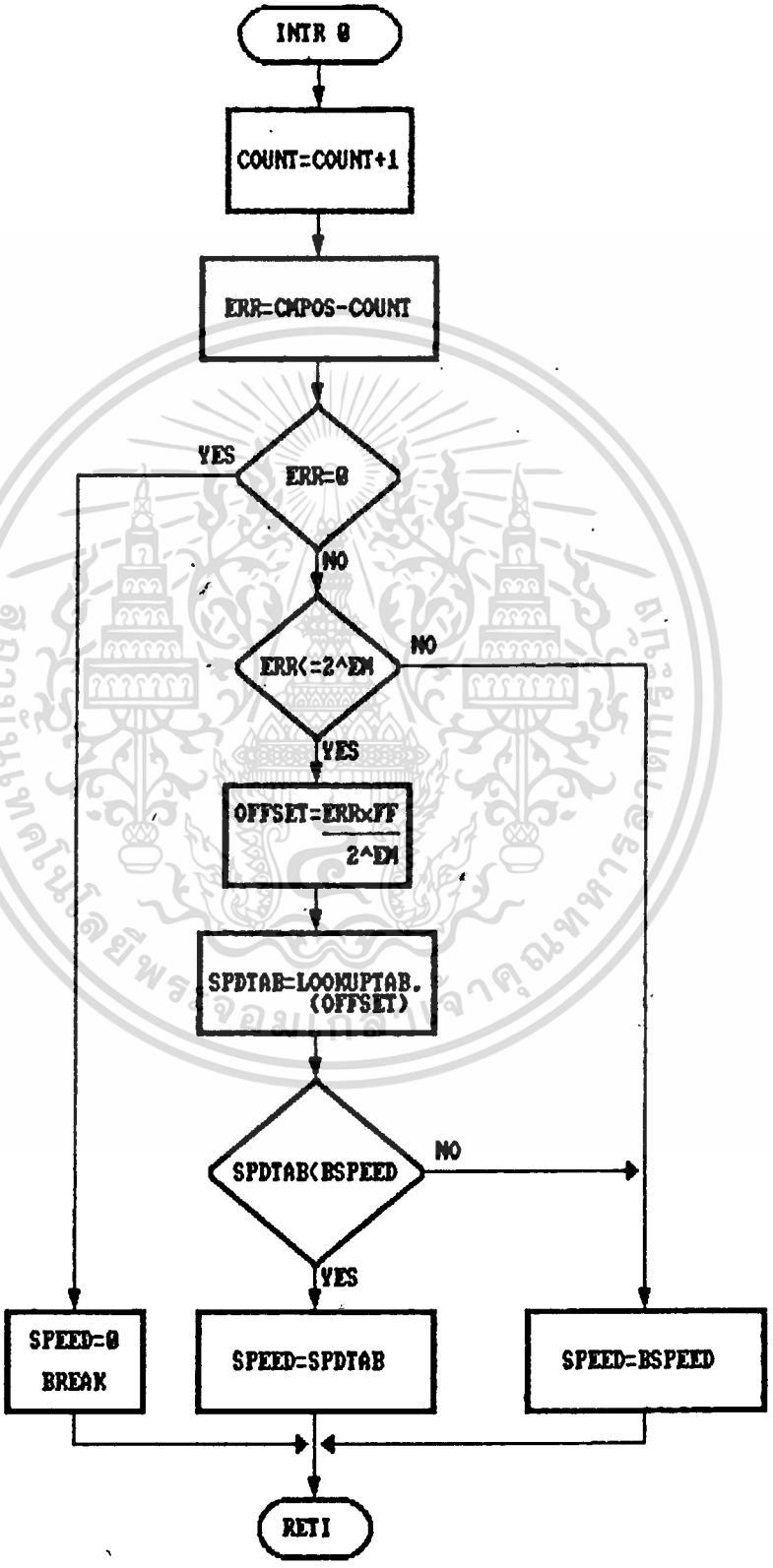


T33098

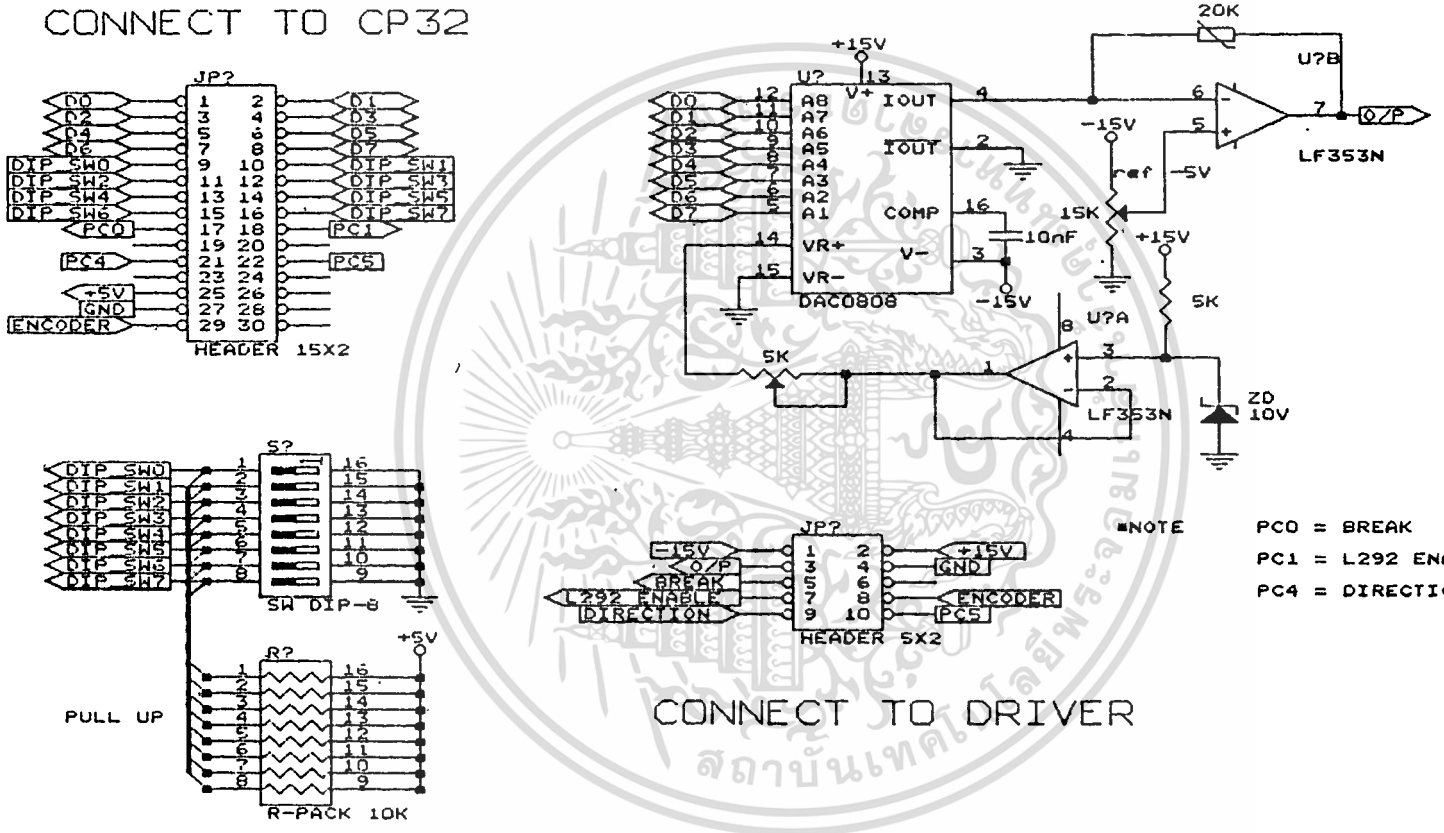
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



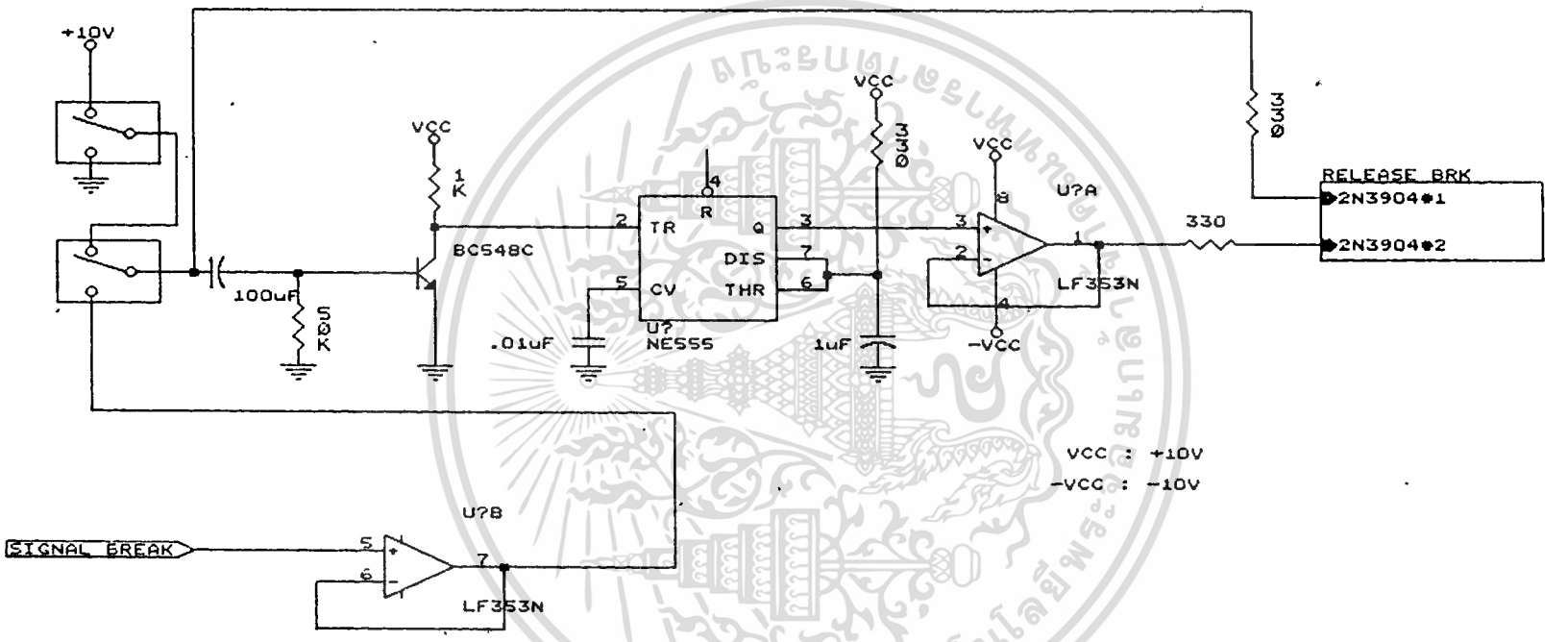
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

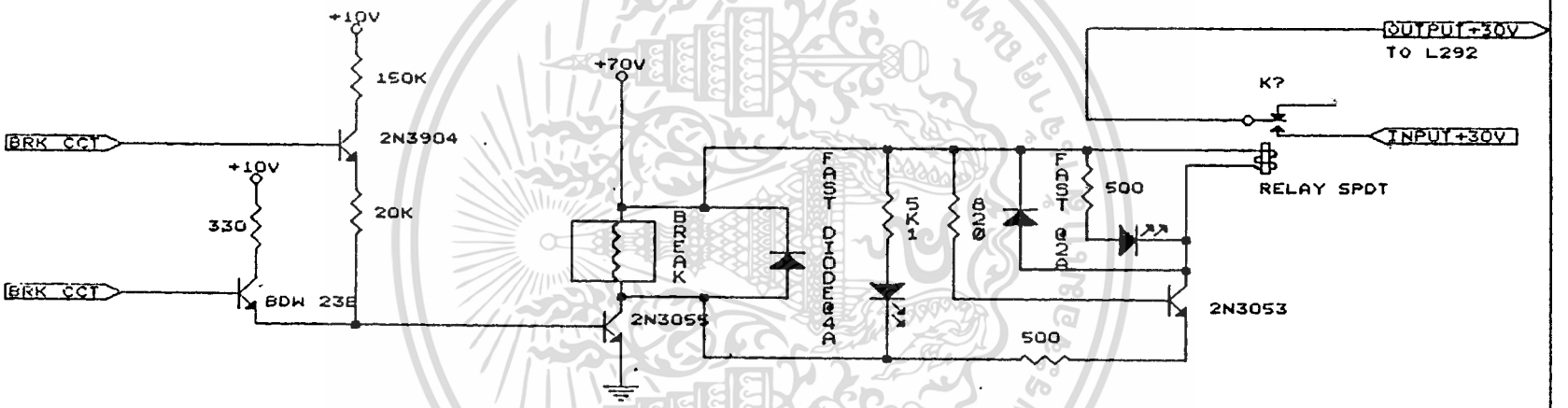


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

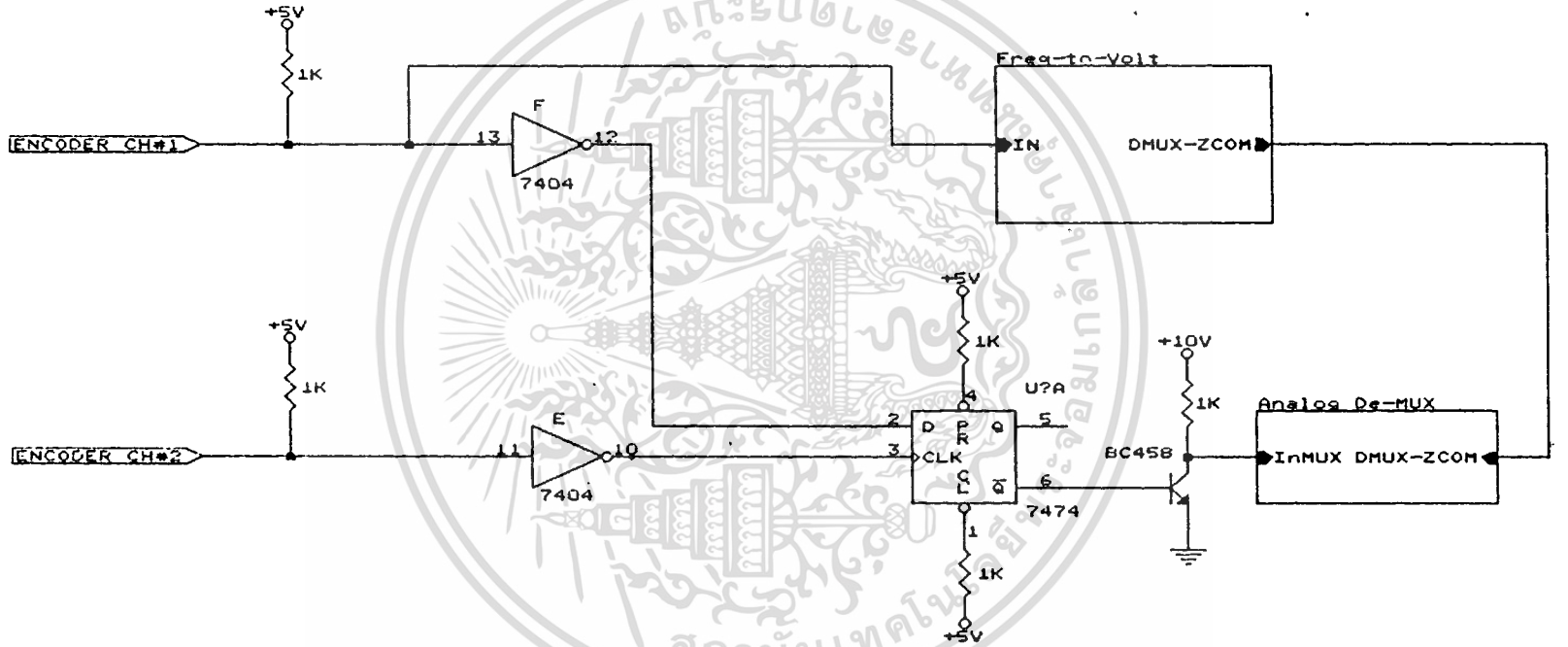


Title	BREAK CIRCUIT	
Size A	Document Number	REV
Date: March 21, 1991	Sheet of	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

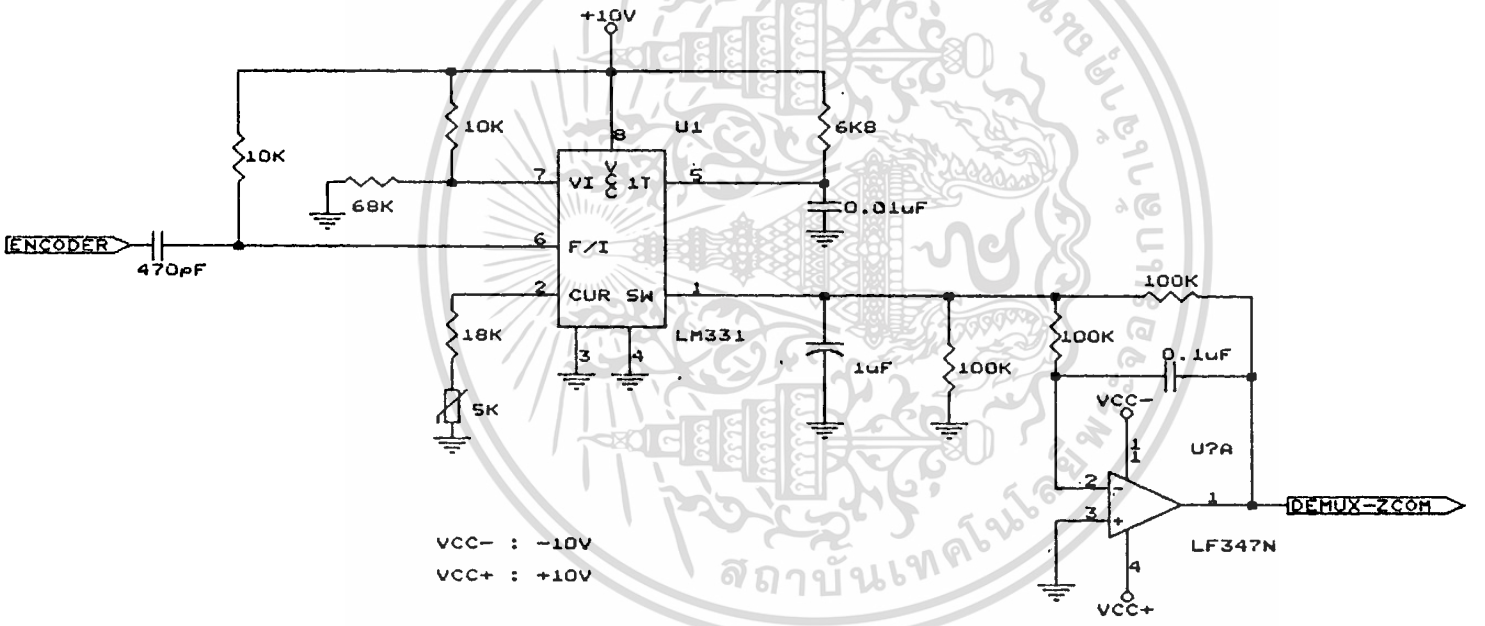


Title	BREAK CIRCUIT CONTINUE	
Size A	Document Number	REV
Date: March 22, 1991	Sheet of	



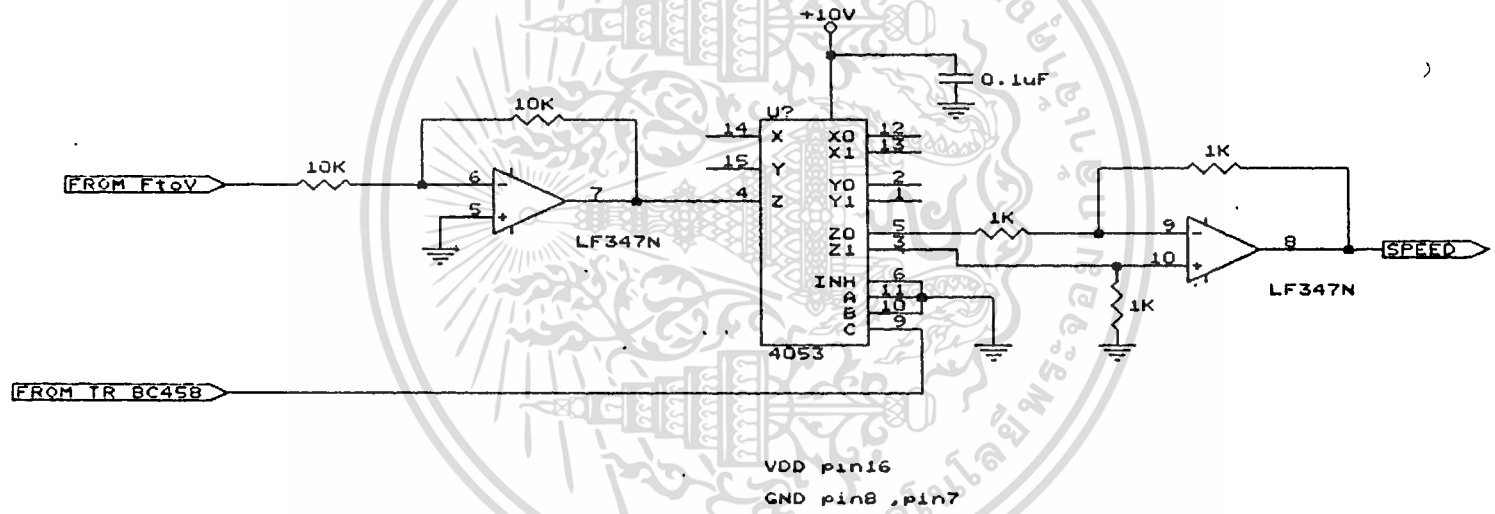
SPEED & POSITION CONTROLLER OF DC MOTOR		
Title: FEEDBACK FROM ENCODER PART		
Size: A	Document Number: 01	REV
Date: March 21, 1991	Sheet 1 of	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



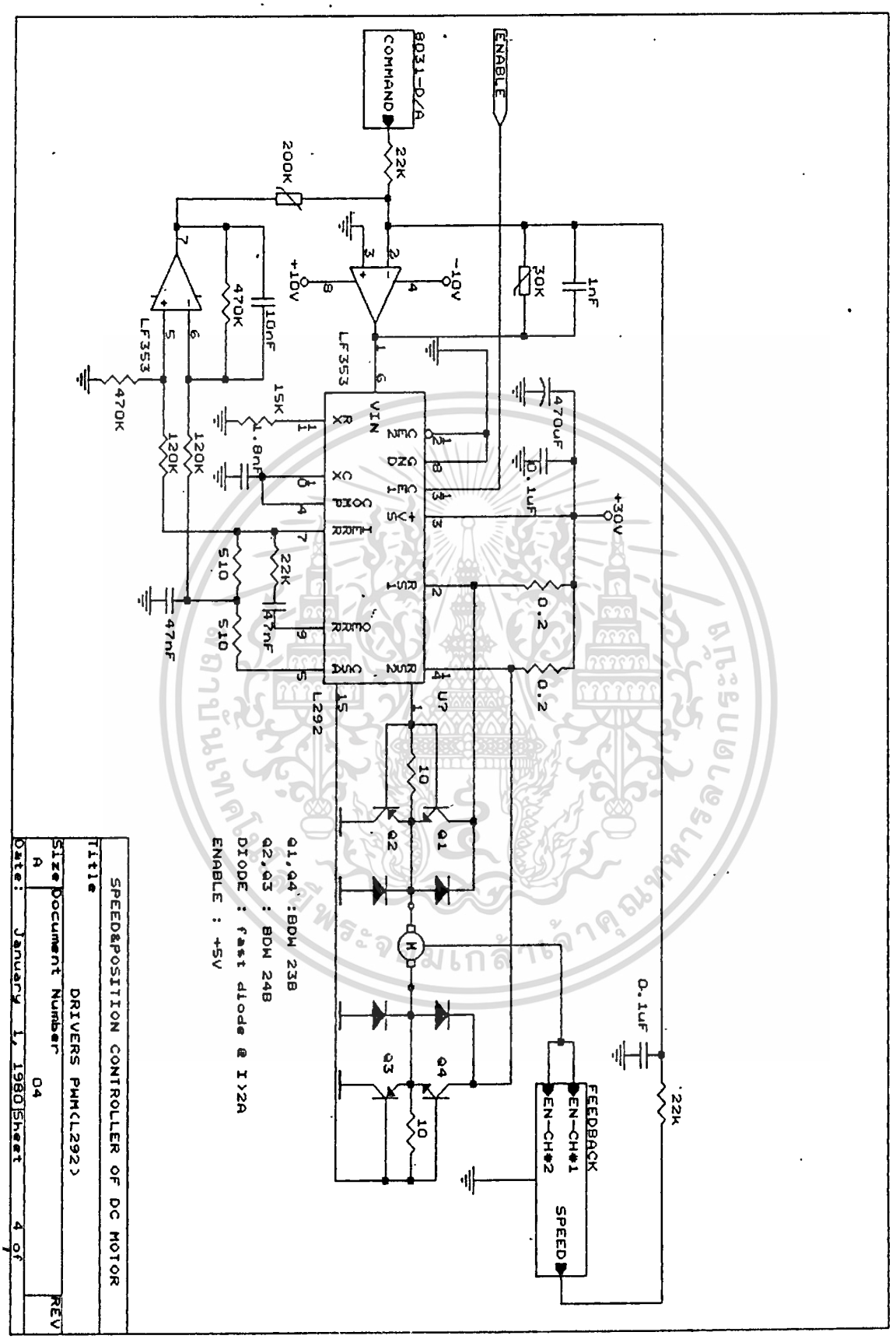
SPEED & POSITION CONTROLLER OF DC MOTOR		
Title: FREQUENCY TO VOLTAGE PART		
Size: A	Document Number: 02	REV:
Date: January 1, 1980	Sheet 2 of	

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



SPEED & POSITION CONTROLLER OF DC MOTOR		
Title: ANALOG MULTIPLEXER DEMULTIPLEXER		
Size: A	Document Number: 03	REV:
Date: January 1, 1980	Sheet 3 of	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

```
;*********************************
;      DC SERVO CONTROLLER
;*********************************
 CHANNAL EQU    40H                ;CHANNAL OF CONTROLLER
 TIME    EQU    41H                ;DELAY TIME
 BUFFER  EQU    49H                ;RECEIVE COMMAND BUFFER
 CMDRT   EQU    50H                ;COMMAND DIRECTION
 CMSPD   EQU    51H                ;COMMAND SPEED
 CMPOS2  EQU    52H                ;COMMAND POSITION  HIGHBYTE
 CMPOS1  EQU    53H                ;                  LOWBYTE
 EM      EQU    54H                ;RANG OF CHECKING ERROR
                                   ;    = 2**EM  { IF EM=0AH THEN
                                   ;         RANG = 1024 }
 EMH     EQU    55H                ;2**EM HIGHBYTE
 SPEED   EQU    56H                ;SPEED VALUE FOR CONTROL O/P
 SPDIV2  EQU    57H                ;SPEED/2
 COUNTH  EQU    60H                ;COUNTER HIGHBYTE
 COUNTL  EQU    61H                ;        LOWBYTE
 ERRH    EQU    62H                ;ERROR HIGHBYTE
 ERRL    EQU    63H                ;ERROR LOWBYTE
 RESULTX EQU    64H                ;       HIGH
 RESULTH EQU    65H                ;       MID
 RESULTL EQU    66H                ;RESULT LOWBYTE OF EQUATION
 RES     EQU    67H                ;CARRY  BYTE
 ERRHI   EQU    69H                ;ERRH BUFFER
 ERRLO   EQU    6AH                ;ERRL
 OFFSET  EQU    6BH                ;OFFSET FOR LOOKUP TABLE
 BSPEED  EQU    6CH                ;SPEED BUFFER FOR CALCULATING
 SPDTAB  EQU    6DH                ;SPEED VALUE FROM TABLE
 START   BIT    78H                ;BIT START ENABLE
 INRANG  BIT    79H                ;BIT CHECK ERROE<=2**EM
 ERRO    BIT    7AH                ;BIT CHECK ERROR<=0
 PORTCON EQU    0E0E3H             ;ADDRESS CONTROL 8255
 PORTA   EQU    0E0E0H             ;ADDRESS 8255 PORTA
 PORTB   EQU    0E0E1H             ;             PORTB
 PORTC   EQU    0E0E2H             ;             PORTC
 TABLE   EQU    0000H              ;ADDRESS TABLE

         ORG    0000H
         LJMP   MAIN
;--------------------------------
;       INTERRUPT VECTOR
;--------------------------------
         ORG    0003H              ;VECTOR INTO
         LJMP   JINTO
 RETO:   RETI

         ORG    001BH              ;VECTOR TF1
 RETF1:  RETI                      ;INTERRUPT BY AUTORELOAD TIMER1

         ORG    0023H              ;VECTOR RI , TI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
         CLR   ES                ;DISABLE SERIAL INTERRUPT
         LJMP  JRI
 RETR:   CLR   RI
         SETB  ES                ;ENABLE SERIAL INTERRUPT
         RET

;------------INTERRUPT 0 ROUTINE
;
 JINTO:  PUSH  ACC
         PUSH  PSW
         CLR   EA
         MOV   A,COUNTL          ;INCREASE COUNTER VALUE
         INC   A
         CJNE  A,#00H,BYTE1
 BYTE2:  INC   COUNTH            ;HIGHTBYTE=61H
 BYTE1:  MOV   COUNTL,A          ;LOWBYTE=60H
         CLR   C                 ;CLEAR CARRY FLAG
         MOV   A,CMPOS1
         SUBB  A,COUNTL          ;ERROR=SETPOINT-COUNTER
         MOV   ERRL,A            ;ERROR LOWBYTE
         MOV   A,CMPOS2
         SUBB  A,COUNTH
         MOV   ERRH,A            ;ERROR HIGHBYTE
         ORL   A,ERRL            ;ACC=0 WHEN ERROR=00H
         JZ    BEO
         JNB   CY,NOTO           ;CY=1 WHEN COMPOS<COUNT
 BEO:    SETB  ERRO              ;  THEN ERRO=1
         CLR   EXO               ;STOP COUNTING BEFORE (TRICK)
         LCALL BREAK             ; STOP MOTION
         LJMP  QUIT
 NOTO:   MOV   A,ERRH
         SUBB  A,EMH             ;CHECK ERRH>=EMH ORNOT!
         JNB   CY,OVRANG         ;IF OUT OF RANG2**EM RETI
         SETB  INRANG            ;ERROR<=2**EM

 LOOKTAB:LCALL EQUAT             ;PROCESS LOOKUP TABLE
         MOV   A,OFFSET
         MOV   DPTR,#TABLE       ;TABLE BEGIN AT 0000H
         XCH   A,DPL             ;DPH:DPL=00:OFFSET
         MOVX  A,@DPTR
         MOV   SPDTAB,A
         CLR   CY
         SUBB  A,BSPEED          ;CHECK SPEEDTAB<=BSPEED
         JNB   CY,OVRANG         ;IF SPDTAB>BSPEED THEN CY=0
         MOV   SPEED,SPDTAB      ;
         LCALL OUT_SP
         LJMP  QUIT

 OVRANG: MOV   SPEED,BSPEED
         LCALL OUT_SP
 QUIT:   SETB  EA
         POP   PSW
         POP   ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LJMP   RETO

;------RECEIVE INTERRUPT ROUTINE
;
JRI:    PUSH   ACC
        PUSH   PSW
        JNB    RB8,DATB          ;DATABYTE Y/N ? (RB8=1>ADDRESSBYTE)
        MOV    A,SBUF            ;                (RB8=0>DATABYTE   )
        MOV    BUFFER,A          ;RECEIVE -->BUFFER(70H)
        ANL    A,#0FH            ;CHECK CHANNAL
CHKCH:  CJNE   A,CHANNAL,IGNOR   ;DEFINE ADDRESSCHANNEL = DIP 4SW.
        MOV    A,BUFFER
        ANL    A,#0F0H           ;CHECK COMMAND

        CJNE   A,#60H,CHKRUN     ;COMMAND READ REAL POSITIONS ?
        MOV    SBUF,COUNTH       ;SEND COUNTS HIGHBYTE TO MASTER2
CHKTXD: JNB    TI,CHKTXD         ;TRANSMIT FINISH YET!
        CLR    TI
        MOV    SBUF,COUNTL       ;SEND COUNT LOWBYTE TO MASTER2
CHKTXD2:JNB    TI,CHKTXD2        ;TRANSMIT FINISH YET!
        CLR    TI
        LJMP   IGNOR

CHKRUN: CJNE   A,#10H,CHKLOAD    ;COMMAND START MOTION
        MOV    COUNTL,#0H        ;PRESET COUNT VALUE
        MOV    COUNTH,#0H        ;                  (EMH)
        LCALL  FEMH              ;CALCULATE 2**EM HIGHBYTE
        SETB   START             ;DEFINE BIT 78H=START ENABLE
        CLR    ERRO              ;CLEAR BIT CHECK ERROR<=0
        CLR    INRANG            ;                ERROR<=2**EM
        LJMP   IGNOR

CHKLOAD:CJNE   A,#50H,CHKTAB
        CLR    SM2               ;PREPARE FOR RECEIVE DATABYTES(BIT 9th
        MOV    R1,#50H           ;MEMORYADDRESS FOR KEEP DATA
        LJMP   IGNOR

DATB:   MOV    R2,#05H           ;LOOP KEEP DATA 5 BYTES
CHK2:   JNB    RI,CHK2           ;RECEIVE A DATA READY!
        CLR    RI
        MOV    A,SBUF            ;KEEP A DATA
        MOV    @R1,A
        INC    R1
        DJNZ   R2,CHK2           ;NEXT BYTE
        SETB   SM2               ;RECEIVE ADDRESSBYTE(BIT 9th=1) ONLY
        LJMP   IGNOR

CHKTAB: CJNE   A,#40H,IGNOR      ;DLOAD TABLE TO EXTERNAL RAM
        CLR    RI
        CLR    SM2               ;PREPARE FOR RECEIVE DATABYTES(BIT 9th
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV   DPTR,#TABLE       ;ADDRESS_TABLE = 8000H
          MOV   R2,#00H           ;LOOP KEEP TABLE 256BYTES
 CHK3:    JNB   RI,CHK3
          CLR   RI
          MOV   A,SBUF            ;KEEP DATA
          MOVX  @DPTR,A
          INC   DPTR
          DJNZ  R2,CHK3           ;NEXT BYTE
          SETB  SM2               ;RECEIVE ADDRESSBYTE(BIT 9th=1) ONL

 IGNOR:   POP   PSW               ;
          POP   ACC
          LJMP  RETR

;--------------------------------
;         DELAY TIME
 DELAY:   NOP
          MOV   R2,TIME
 DLY1:    MOV   R3,#0FFH
 DLY2:    DJNZ  R3,DLY2
          DJNZ  R2,DLY1
          RET

;--------------------------------
;SET I/O PORT  ( INITIAL 8255 ON CP32 )
 PORTSET:MOV   TIME,#0F5H         ;DELAY AFTER RESET HARDWARE
          LCALL DELAY             ;DEFINE
          MOV   DPTR,#PORTCON     ;8255 CONTROL WORD
          MOV   A,#82H            ;PORTA,PORTC = O/P
          MOVX  @DPTR,A           ;PORTB       I/P
          RET

;--------------------------------
;    INITIAL PROGRAM+MODESET
;--------------------------------
                                  ;      TIMER1     |      TIMER0
 INITSYS:MOV   TMOD,#00101001B    ;GATE,C/T,M1,M0 | GATE,C/T,M1,M0
                                  ;T1=TIMER MODE2(AUTO RELOAD)
                                  ;T0=TIMER MODE1(16BIT TIMER)
                                  ;
          ORL   TCON,#00000101B   ;TF1,TR1,TF0,TR0,IE1,IT1,IE0,IT0
                                  ;EXTERNAL INT_FALLING EDGE
                                  ;
          MOV   IE,#00010001B     ;EA,-,-,ES,ET1,EX1,ET0,EX0
                                  ;1=ENABLE   0=DISABLE
                                  ;
          MOV   IP,#00000001B     ;-,-,-,PS,PT1,PX1,PT0,PX0
                                  ;PRIORITY LEVEL
                                  ;1=HIGH 0=LOW
;--------------------------------
;SET TIMER1 TO GENERATE BAUD RATES
;
;      BAUD RATES = (K*OscFreq) / (32*12*[256-TH1])
;      OSC=11.059MHz,TH1=FAH THEN RATE=4800 BAUD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV   PCON,#00H          ;SMOD=0 THEN K=1
          MOV   TH1,#0FAH          ;DEFINE VALUE FOR TH1
          SETB  TR1                ;RUN TIMER1
;---------------------------------
;         SET SERIAL
          ORL   SCON,#11100000B    ;SM0,SM1,SM2,REN,TB8,RB8,TI,RI
                                   ;MODE3:9-BIT UART
                                   ;SM2=USED FOR MULTIPROCESSOR COMMUN
          SETB  REN                ;RECEIVE ENABLE
          RET

;---------------------------------
;         BREAK MOTION
 BREAK:   PUSH  ACC
          MOV   A,#80H             ;80H D TO A = 0 VOLT.
          MOV   P1,A
          MOV   DPTR,#PORTA        ;DIGITAL TO ANALOG CONTROL OUTPUT
          MOVX  @DPTR,A
          MOV   DPTR,#PORTC        ;SELECT PORTC OF 8255
          MOV   A,#00H             ; CONNECT TO MEGNETIC BREAK
          MOVX  @DPTR,A            ;  AND L292.CHIP ENABLE
          POP   ACC
          RET

;---------------------------------
;         DISBREAK
 DISBRK:  PUSH  ACC
          MOV   A,#80H             ;BEGINING SPEED=0 VOLT.
          MOV   P1,A
          MOV   DPTR,#PORTA        ;DIGITAL TO ANALOG CONTROL OUTPUT
          MOVX  @DPTR,A
          MOV   DPTR,#PORTC        ;PORTC0 = UNLOCKEDBREAK
          MOV   A,#0FFH            ;PORTC1 = ENABLE L292
          MOVX  @DPTR,A
          POP   ACC
          RET

;---------------------------------
; FIND VALUE HIGHBYTE OF 2**EM
 FEMH:    MOV   R2,EM              ;RESULT=2**EM
          MOV   RESULTL,#01H
          MOV   RESULTH,#00H
          CLR   C
 MUL_2:   MOV   A,RESULTL
          RLC   A
          MOV   RESULTL,A
          MOV   A,RESULTH
          RLC   A
          MOV   RESULTH,A
          CLR   C
          DJNZ  R2,MUL_2           ;DEFINE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV    EMH,A              ;EM=8,9,10,11,12,13,14,15
          RET

;-------------------------------
;         PRESET VALUE
 PRESET:  MOV    DPTR,#TABLE        ;ADDRESS_TABLE = 0000H
          MOV    R2,#00H            ;PRESET TABLE 256BYTES
          MOV    A,#00
 CHKLP:   MOVX   @DPTR,A
          INC    DPTR
          INC    ACC
          DJNZ   R2,CHKLP           ;NEXT BYTE
          MOV    DPTR,#TABLE
          MOV    R2,#20H
          MOV    A,#20H             ;FILL ADDRESS 00-20H=#20H
 CHKLP1:  MOVX   @DPTR,A            ;
          INC    DPTR               ;
          DJNZ   R2,CHKLP1          ;

          MOV    DPTR,#PORTB        ;READ CHANNAL FROM DIP SWITCH
          MOVX   A,@DPTR            ;    CONNECT AT PORTB
          MOV    CHANNAL,A          ;DEFINE CHANNAL OF CONTROLLER

          MOV    CMDRT,#0H          ;DIRECTION NORMAL TURN RIGHT
          MOV    CMSPD,#0H          ;SPEED NORMAL 0%
          MOV    CMPOS1,#0H         ;COMMAND POSITION LOWBYTE
          MOV    CMPOS2,#0H         ;                 HIGHBYTE
          MOV    COUNTL,#0H         ;PRESET COUNT VALUE
          MOV    COUNTH,#0H         ;
          MOV    EM,#0AH            ;EM=10  { RANG=1024 }
          CLR    START              ;NORMAL STOP
          LCALL  BREAK
          RET

;-------------------------------
;         EQUATION
;         (ERROR * 100%)/(2**EM)        {100%=#0FFH}
 EQUAT:   MOV    ERRHI,ERRH
          MOV    ERRLO,ERRL
          MOV    B,#0FFH
          MOV    A,ERRLO
          MUL    AB
          MOV    RESULTL,A          ;RESULT LOWBYTE
          MOV    RES,B              ;CARRYBYTE
          MOV    B,#0FFH
          MOV    A,ERRHI
          MUL    AB
          ADD    A,RES
          MOV    RESULTH,A          ;RESULT MIDBYTE
          MOV    A,B
          ADDC   A,#00H             ;ADD WITH CARRY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          MOV    RESULTX,A          ;RESULT HIGHBYTE
 DIV_EM:  MOV    R2,EM              ;RESULT DIV 2**EM
 DIV_2:   CLR    C
          MOV    A,RESULTX
          RRC    A
          MOV    RESULTX,A
          MOV    A,RESULTH
          RRC    A
          MOV    RESULTH,A
          MOV    A,RESULTL
          RRC    A
          MOV    RESULTL,A
          DJNZ   R2,DIV_2
          MOV    OFFSET,RESULTL     ;USE RESULTL FOR LOOK UP TABLE
          RET

;-------------------------------------
;SEND COMMAND SPEED TO ANALOG OUTPUT
 OUT_SP:  PUSH   ACC
          PUSH   PSW
          MOV    A,SPEED            ;LOAD SPEED VALUE(%)
          MOV    B,#2H
          DIV    AB                 ;SPEED/2
          MOV    SPDIV2,A           ;
          MOV    A,#80H             ;80H D_A = 0 VOLT
          MOV    R5,CMDRT
          CJNE   R5,#00H,RIGHT      ; 1 = RIGHT , 0 = LEFT
          SUBB   A,SPDIV2           ;VELOCITY=80H-SPEED/2
          SJMP   D_A
 RIGHT:   ADD    A,SPDIV2           ;VELOCITY=80H+SPEED/2
 D_A:     MOV    P1,A
          MOV    DPTR,#PORTA        ;DIGITAL TO ANALOG CONTROL OUTPUT
          MOVX   @DPTR,A
          POP    PSW
          POP    ACC
          RET

;*********************************
;         MAIN PROGRAM
;*********************************
 MAIN:    LCALL  PORTSET            ;SET MODE 8255
          LCALL  INITSYS            ;SET SYSTEM OF CP32
          LCALL  PRESET             ;SET VALUE ALL VARIABLE
          SETB   EA                 ;ENABLE ALL INTERRUPT
          CLR    EX0
 LOOP:    NOP
          JNB    START,LOOP         ;START ENABLE (Y/N)?
          SETB   EX0
          LCALL  DISBRK             ;DISBREAK ROTER
          MOV    SPEED,#0FFH
          LCALL  OUT_SP
          MOV    BSPEED,#19H        ;INITIAL SPEED
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHKPOS: JB     ERRO,STOP          ;ERROR<=0 ORNOT!
        MOV    TIME,#05H          ;DELAY TIME
        LCALL  DELAY
        INC    BSPEED             ;BSPEED=BSPEED+1
        CLR    CY                        !
        MOV    A,BSPEED
        SUBB   A,CMSPD            ;CHECK BSPEED>=CMSPD ORNOT
        JB     CY,CHKPOS
        MOV    BSPEED,CMSPD
POSCHK: JNB    ERRO,POSCHK        ;ERROR=0 ORNOT
STOP:   CLR    EXO                ;STOP COUNTING BEFORE (TRICK)
        LCALL  BREAK              ; STOP MOTION
        CLR    START              ;CLEAR BIT CHECK
        LJMP   LOOP               ;DO NEXT COMMAND

        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้