สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบหุ่นยนต์สองแขน

TWO ARMS ROBOT SYSTEM

โดย

นายวุฒิชัย เจริญสุขนิรันดร 41013314

นายสมศักดิ์ บุญกลอย 41013316

เลขหมู่.....
เลขทะเบียน 42373
วัน, เดือน, ปี 20 พ.ค. 2545

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TWO ARMS ROBOT SYSTEM

Mr. Wuttichai	Charoensuknirundon	41013314
Mr. Somsak	Boongloy	41013316

Project Report Submitted in Partial Fulfillment of the Requirement

for the Bachelor 's Degree

Department of Industrial Technology

Faculty of Engineering

King Mongkut ' s Institute of Technology Ladkrabang

2000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบหุ่นยนต์สองแขน
ชื่อนักศึกษา	นายวุฒิชัย เจริญสุขนิรันดร 41013314
	นายสมศักดิ์ บุญกลอย 41013316
อาจารย์ที่ปรึกษา	ผศ.ดร. ปีติเขต สู้รักษา
ภาควิชา	เทคนิคอุตสาหกรรม
ปีการศึกษา	2543

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

.....หัวหน้าภาควิชาเทคนิคอุตสาหกรรม

( )

คณะกรรมการสอบ

.....ประธานคณะกรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

ลิขสิทธิ์ของคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Report	Two Arms Robotic System	
By	Mr. Wuttichai Charoensuknirundon	41013314
	Mr. Somsak Boongloy	41013316
Project Report Advisor	Assist Dr. Pitikhate Sooraksa	
Department	Industrial Technology	
Academic Year	2000	

Accepted by the Faculty of Engineering , King Mongkut ' s Institute of technology , Ladkrabang in partial fulfillment of the requirements for the bachelor ' s degree

Project Report Committee

.....Chairman

( )

.....Member

( )

..... Member

( )

..... Member

( )

..... Member

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ ระบบหุ่นยนต์สองแขน
ชื่อนักศึกษา นายวุฒิชัย เจริญสุขนิรันดร 41013314
นายสมศักดิ์ บุญกลอย 41013316
อาจารย์ที่ปรึกษา ผศ.ดร. ปิติเขต สู้รักษา
ภาควิชา เทคนิคอุตสาหกรรม
ปีการศึกษา 2543

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ ได้อธิบายถึงหลักการทำงานของแขนกลสองแขน โดยใช้การควบคุมแบบระบบ PID ขนาน ซึ่งสามารถควบคุมแขนกลได้อย่างมีประสิทธิภาพ ลักษณะของระบบหุ่นยนต์ประกอบไปด้วย ภาคแหล่งจ่ายไฟ ไมโครคอนโทรลเลอร์ PID คอนโทรลเลอร์ วงจรขับมอเตอร์ และวงจรเซ็นเซอร์ตำแหน่ง

ผลการทดลองและประยุกต์ใช้งานแสดงให้เห็นว่าต้นแบบของแขนกลสองแขนทำงานได้ดีด้วยวิธีการควบคุมที่ได้ออกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Project Report	Two Arms Robotic System
By	Mr. Wuttichai Charoensuknirundon 41013314
	Mr. Somsak Boongloy 41013316
Project Report Advisor	Asst.Prof.Dr. Pitikhate Sooraksa
Department	Industrial Technology
Academic Year	2000

Abstract

This thesis describes design principles of the two arms robot system . The robot arms are controlled by using PID algorithms working in parallel processing . The system composes of power sources , microcontrllers PID software , drivers , and position sensor .

The experimental results and the applications show that the prototype works effectively by merely using the proposed algorithms

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

เนื่องจากระบบหุ่นยนต์สองแขนประกอบไปด้วย 2 ส่วน คือ ฮาร์ดแวร์ และซอฟแวร์ จึงจำเป็นต้องใช้ความรู้และการค้นคว้าอย่างมาก ซึ่งความซับซ้อนทางกลวิธีทางการโปรแกรมนั้นมากเป็นพิเศษ ซึ่งคณะผู้จัดทำมีความถนัดทางด้านนี้น้อยมาก แต่หากได้รับความอนุเคราะห์ด้านคำปรึกษาชี้แนะในแนวทางที่ถูกต้องของอาจารย์ผู้มีพระคุณ ซึ่งนอกจากจะได้รับการสั่งสอน ถ่ายทอดความรู้แล้ว ยังได้รับความห่วงใยต่อคณะผู้จัดทำฉันท์ศิษย์และอาจารย์อีกด้วย

ฉะนั้นโอกาสอันเหมาะสมนี้ คณะผู้จัดทำขอได้กล่าวคำขอบพระคุณในความรู้ ความห่วงใยและปรารถนาดี ที่คณะผู้จัดทำได้รับ จากท่านอาจารย์ ผศ. ดร. ปิติเขต สู้รักษา และรุ่นพี่ที่ให้ความช่วยเหลือ คณะผู้จัดทำ (ณัฐวุฒิ เพ็งจันทอง) มา ณ. ที่นี้ด้วยความสำนึกในพระคุณ อย่างสูง ส่วนข้อผิดพลาดหรือแนวทางที่ผิดนั้น คณะผู้จัดทำขอน้อมรับไว้เพื่อแก้ไขต่อไปในอนาคต

คณะผู้จัดทำ

16 ตุลาคม 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทคัดย่อ		ก
กิตติกรรมประกาศ		ข
สารบัญ		ค
สารบัญรูป		จ
บทที่ 1	บทนำ	1
	1.1 ความเป็นและความสำคัญของปัญหา	1
	1.2 ขีดความสามารถของโครงงาน	1
	1.3 เนื้อหาโดยสังเขป	1
บทที่ 2	ทฤษฎีและหลักการ	2
	2.1 ระบบควบคุม	2
	2.2 ดีซีเซอร์โวมอเตอร์	4
	2.3 ระบบเซอร์โว	5
	2.4 การสื่อสารระหว่างซีพียูหลายตัว	10
	2.5 P-คอนโทรลเลอร์	13
	2.6 PI-คอนโทรลเลอร์	14
	2.7 PID-คอนโทรลเลอร์	15
บทที่ 3	การสร้างและการออกแบบ	18
	3.1 หลักการทำงานของแขนกล	19
	3.2 ส่วนประกอบทางโครงสร้าง	26
	3.3 ส่วนประกอบทางโครงสร้าง	44
บทที่ 4	การทดลองและผลการทดลอง	48
	4.1 ขั้นตอนการทดลอง	48
	4.2 ผลการทดลอง	48
บทที่ 5	ปัญหาและข้อเสนอแนะ	51
	5.1 สรุปโครงงาน	51
	5.2 ปัญหาในการทำโครงงาน	51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

บทที่ 2
รูปที่ 2.1 กระบวนการที่เราจะควบคุม 2
รูปที่ 2.2 ตัวอย่างระบบควบคุมแบบลูปเปิด 2
รูปที่ 2.3 ระบบควบคุมป้อนกลับแบบลูปปิด 3
รูปที่ 2.4 ระบบควบคุมแบบมีหลายตัวแปร 4
รูปที่ 2.5(a) แสดงระบบเซอร์โว 6
รูปที่ 2.5(b) แสดงบล็อกไดอะแกรมของระบบเซอร์โว 8
รูปที่ 2.5(c) แสดงบล็อกไดอะแกรมของ ระบบเซอร์โวแบบย่อ 9
รูปที่ 2.6(a) การต่อ CPU แบบจุดต่อจุด 10
รูปที่ 2.6(b) การต่อ CPU แบบหลายจุด 11
รูปที่ 2.7 ข้อมูลกำหนดตำแหน่ง 11
รูปที่ 2.8 ข้อมูลข่าวสาร 12
รูปที่ 2.9 แสดงตัวอย่าง P-คอนโทรลเลอร์ 13
รูปที่ 2.10 แสดงบล็อกไดอะแกรม P-คอนโทรลเลอร์ 13
รูปที่ 2.11 แสดงผลตอบสนองของสัญญาณ 14
รูปที่ 2.12 แสดงตัวอย่าง PI-คอนโทรลเลอร์ 14
รูปที่ 2.13 แสดงบล็อกไดอะแกรม PI-คอนโทรลเลอร์ 15
รูปที่ 2.14 แสดงผลตอบสนองของสัญญาณ 15
รูปที่ 2.15 แสดงตัวอย่าง PID-คอนโทรลเลอร์ 16
รูปที่ 2.16 แสดงบล็อกไดอะแกรม PID-คอนโทรลเลอร์ 16
รูปที่ 2.17 แสดงผลตอบสนองของสัญญาณ 17

บทที่ 3
รูปที่ 3.1 แสดงบล็อกไดอะแกรมของวงจรทั้งหมด 18
รูปที่ 3.2 แสดงโครงสร้างของ Encoder ของมอเตอร์ 20
รูปที่ 3.3 แสดงรูปคลื่นของ A และ B 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.4 แสดงบล็อกไดอะแกรม PID-คอนโทรลเลอร์	22
	หน้า
รูปที่ 3.5 แสดงวงจรตรวจจับทิศทางการหมุนของมอเตอร์	23
รูปที่ 3.6 แสดงวงจรสมบูรณ์ของชุดควบคุมมอเตอร์แบบ PID-คอนโทรลเลอร์	24
รูปที่ 3.7 แสดงบล็อกไดอะแกรมการทำงานของโปรแกรมควบคุมมอเตอร์	25
รูปที่ 3.8 แสดงการสร้างสัญญาณ PWM	26
รูปที่ 3.9 แสดงตัวอย่างการสุ่มสัญญาณ error	28
รูปที่ 3.10 แสดงการหาค่าอินทิกรัลด้วยพื้นที่ใต้กราฟ	29
รูปที่ 3.11 แสดงการต่อโพเทนชิโอมิเตอร์เข้ากับเพลามอเตอร์	30
รูปที่ 3.12 แสดงบล็อกไดอะแกรม	32
รูปที่ 3.13 แสดงวงจรสมบูรณ์ของ PID-คอนโทรลเลอร์	32
รูปที่ 3.14 แสดงวงจรสมบูรณ์ของ PID คอนโทรล	33
รูปที่ 3.15 แสดงวงจร Switch & Display	34
รูปที่ 3.16 แสดงปุ่มควบคุมและแสดงผลที่ด้านหลังของแขนกล	35
รูปที่ 3.17 แสดงวงจรภาคจ่ายไฟ	37
รูปที่ 3.18 แสดงวงจรขับมอเตอร์	38
รูปที่ 3.18 แสดงโฟลวชาร์ตการทำงานของ CPU ตัวหลัก	39
รูปที่ 3.20 แสดงวงจรเมนบอร์ด	40
รูปที่ 3.21 แสดงโครงสร้างของข้อมูลใน 1 บล็อก	41
รูปที่ 3.22 แสดงโปรแกรมที่รับข้อมูลจากผู้ใช้	42
รูปที่ 3.23 แสดงส่วนประกอบฐาน	44
รูปที่ 3.24 แสดงส่วนประกอบแขน	44
รูปที่ 3.25 แสดงส่วนประกอบฐานรอง	45
รูปที่ 3.26 แสดงขนาดของส่วนประกอบฐาน	45
รูปที่ 3.27 แสดงขนาดของแขนท่อนที่1	46
รูปที่ 3.28 แสดงขนาดของแขนท่อนที่2	46
รูปที่ 3.29 แสดงขนาดของแขนท่อนที่3	47
รูปที่ 3.30 แสดงขนาดของแขนท่อนที่4	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

รูปที่ 4.1 แสดงผลตอบสนองของระบบควบคุมแบบ PID ขณะที่ไม่มีโหลด 57
รูปที่ 4.2 แสดงผลตอบสนองของระบบควบคุมแบบ PID ขณะที่มีโหลด 58
รูปที่ 4.3 แสดงผลตอบสนองของระบบควบคุมแบบ PID ขณะที่มีโหลด(ต่างกัน) 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันแขนกลหรือแขนหุ่นยนต์ก็เป็นเครื่องจักรกลอีกประเภทหนึ่ง จะมีบทบาทมากในการผลิตหรือดำเนินกิจกรรมต่างๆ ซึ่งนิยมใช้กันอย่างแพร่หลายในงานอุตสาหกรรมโดยทั่วไป ซึ่งประโยชน์หลักๆของแขนกลคือจะใช้กับงานที่เป็นอันตรายต่อมนุษย์และงานที่ต้องการความแม่นยำ เช่น

1. งานที่ทำแล้วเกิดอันตรายได้ง่าย เช่น งานประเภทเชื่อมในที่สูงเหนือศีรษะ งานใต้ดิน หรือใต้ทะเลลึก ๆ
3. งานที่ใช้ความละเอียดค่อนข้างสูง
4. งานที่สิ่งแวดล้อมไม่เอื้ออำนวย เช่น งานในอวกาศ
5. งานที่ทำเป็นระยะเวลานานๆ

1.2 ขีดความสามารถของโครงงาน

1. สามารถโปรแกรมการเคลื่อนไหวในลักษณะต่างๆของแขนกลได้
2. สามารถนำแขนกลชุดนี้ไปใช้กับงานต่างๆได้ เช่น ยกวัตถุ
3. สามารถใช้ชุดแขนกลนี้เพื่อเป็นการศึกษาถึงระบบควบคุมได้

1.3 เนื้อหาโดยสังเขป

บทที่ 2 จะเป็นทฤษฎีและหลักการที่เกี่ยวข้องที่สำคัญจะกล่าวถึงในเรื่องของ ระบบดีซีเซอร์โวมอเตอร์ การควบคุมระบบคอนโทรล แบบ PID คอนโทรลเลอร์

บทที่ 3 เป็นการออกแบบและการสร้าง กล่าวถึงการออกแบบการสร้างวงจรตัวควบคุมแบบพีไอดี วงจรขับมอเตอร์กระแสตรง วงจรที่ใช้เป็นตัวควบคุม และโครงสร้างของโครงงาน รวมไปถึงการเขียนโปรแกรมที่ใช้ในการควบคุมส่วนต่างๆด้วยโปรแกรมภาษาแอสแซมบลี้ของไมโครคอนโทรลเลอร์ตระกูล MCS - 51

บทที่ 4 การทดลองและผลการทดลอง มีเนื้อหาเกี่ยวกับการทดลองของตัวควบคุมแบบพีไอดีโดยการควบคุมตำแหน่งการเคลื่อนที่ของแขนกล

บทที่ 5 ปัญหา แนวทางแก้ไขและพัฒนา กล่าวถึงข้อสรุปข้อบกพร่องต่าง ๆ ของชุดแขนกล ปัญหาที่เกิดในการจัดสร้างและกล่าวถึงแนวทางในการปรับปรุง การแก้ไขและการพัฒนาชุดแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ระบบควบคุม

ระบบควบคุม คือ ระบบที่อาศัยพื้นฐานการป้อนกลับเพื่อควบคุมให้ได้ผลตอบสนองของระบบเป็นไปตามที่เราต้องการ

พื้นฐานในการวิเคราะห์ระบบควบคุมกระทำได้โดยอาศัยหลักทฤษฎีระบบลิเนียร์ โดยกำหนดถึงความสัมพันธ์ของอินพุท-เอาท์พุทของส่วนประกอบของระบบ ดังนั้นส่วนประกอบของระบบหรือกระบวนการที่เราจะทำการควบคุมสามารถแสดงได้ด้วยบล็อคไดอะแกรม ดังแสดงในรูปที่ 2.1

รูปที่ 2.1 กระบวนการที่เราจะควบคุม

ระบบควบคุมแบ่งออกได้เป็น 2 แบบ คือ

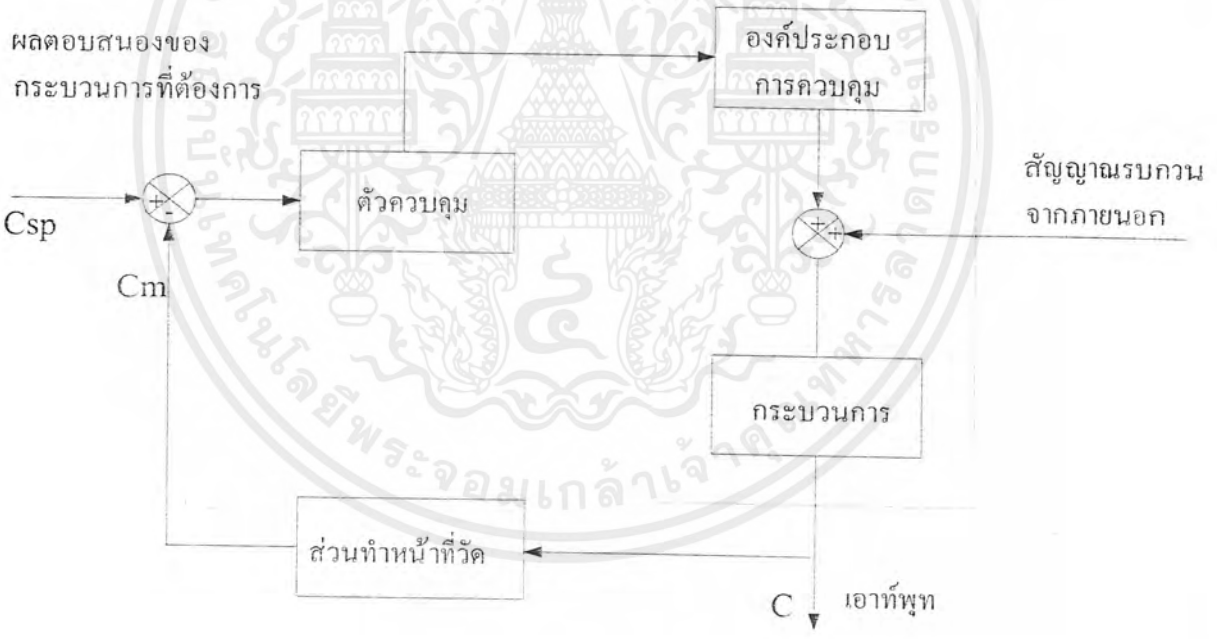
ก. ระบบควบคุมแบบลูปเปิด คือ ระบบที่มีตัวควบคุมเป็นตัวควบคุมเพื่อให้ได้ผลตอบสนองตามที่เราต้องการ ระบบแบบนี้ไม่มีการป้อนกลับแสดงดังรูปที่ 2.2

รูปที่ 2.2 ตัวอย่างระบบควบคุมแบบลูปเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

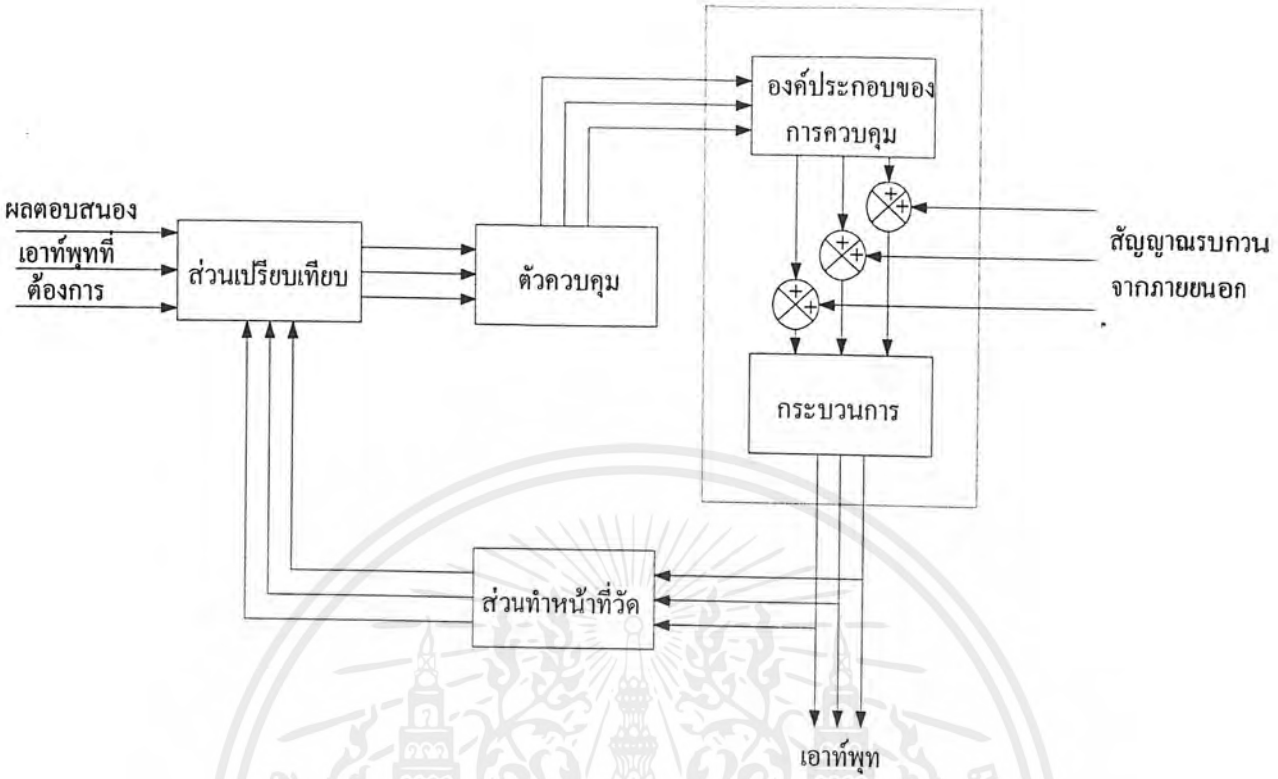
ข. ระบบควบคุมแบบลูปปิดหรือแบบมีการป้อนกลับ คือ ระบบที่มีการป้อนกลับเอาเอาท์พุท ของระบบกลับไปเปรียบเทียบกับอินพุทของระบบจะได้เป็นความแตกต่างหรือสัญญาณผิดพลาด ป้อนไปให้ตัวควบคุมเพื่อไปควบคุมระบบหรือกระบวนการให้ได้ผลตอบสนองตามที่เราต้องการ ระบบควบคุมป้อนกลับแบบลูปปิดแสดงได้ดังรูปที่ 2.3

การวิเคราะห์และออกแบบระบบควบคุมจะอยู่บนพื้นฐานในแนวความคิดของการป้อนกลับ นอกจากนั้นแล้วระบบที่มีความยุ่งยากซับซ้อนมากขึ้น ความสัมพันธ์ของตัวแปรที่ถูกคอนโทรลต่าง ๆ จะต้องพิจารณาในแบบแผนการควบคุมที่อยู่ในลักษณะบล็อคไดอะแกรมของระบบควบคุมแบบมี ตัวแปรหลายตัวแสดงดังในรูปที่ 2.4



รูปที่ 2.3 ระบบควบคุมป้อนกลับแบบลูปปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ระควบคุมแบบมีตัวแปรหลายตัว

2.2 ดีซีเซอร์โวมอเตอร์

ดีซีเซอร์โวมอเตอร์มีหลายชนิดที่ใช้ในงานอุตสาหกรรม ดีซีมอเตอร์ที่ใช้ในระบบเซอร์โว เรียกว่า ดีซีเซอร์โวมอเตอร์ เมื่อมีการหมุนจะทำให้เกิดความเฉื่อยเกิดขึ้นเล็กน้อย มอเตอร์แบบที่มีแรงบิดสูงมาก มีขายเพื่อให้เลือกใช้ให้เหมาะสมกับงาน ดีซีเซอร์โวมอเตอร์แบบใช้พลังงานค่อนข้างต่ำจะใช้เป็นอุปกรณ์เครื่องมือ และใช้เกี่ยวกับคอมพิวเตอร์ เช่น ทำให้แผ่น DISK หมุน, ในเทปเดค และในเครื่อง PRINTER

ในดีซีเซอร์โวมอเตอร์สนามแม่เหล็กที่พันรอบแกนอาจจะเชื่อมโยงไปถึง หรือแยกออกจากอาร์เมเจอร์ เมื่อสนามแม่เหล็กไม่ขึ้นอยู่กับกระแสอาร์เมเจอร์ในดีซีเซอร์โวมอเตอร์ ด้วยเหตุนี้สนามแม่เหล็กจะคงที่ได้ ดังนั้นดีซีเซอร์โวมอเตอร์จะมีแม่เหล็กภายในตัวมัน ดีซีเซอร์โวมอเตอร์จะมีการแพร่ของสนามแม่เหล็ก เปรียบมีสนามแม่เหล็กภายในตัวมันซึ่งมันถูกควบคุมโดยกระแสอาร์เมเจอร์ เรียกว่า อาร์เมเจอร์ควบคุมดีซีเซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

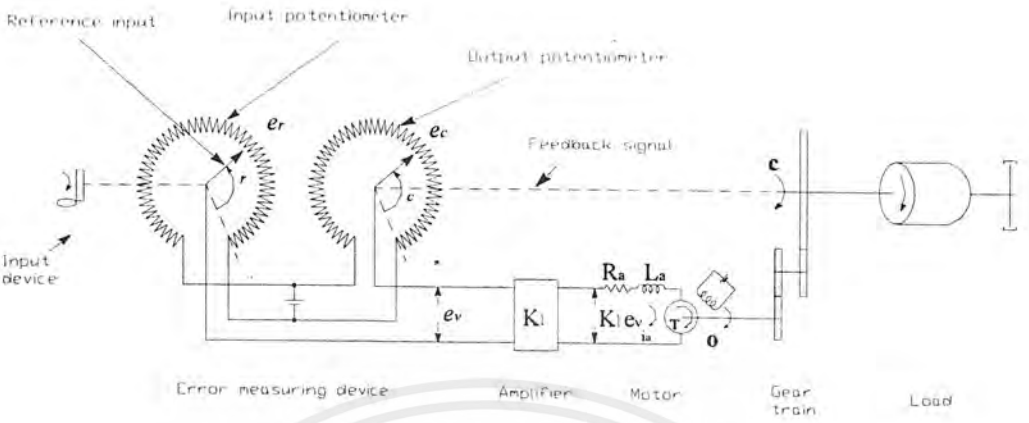
ในหัวข้อต่อมา เมื่อกระแสอาร์เมเจอร์มีค่าคงที่ ซึ่งถูกควบคุมโดยแรงดันสนามแม่เหล็ก ดีซีมอเตอร์ถูกเรียกว่า สนามแม่เหล็กควบคุมดีซีมอเตอร์ (การควบคุมความเร็วของระบบจะใช้สนามแม่เหล็กควบคุมดีซีมอเตอร์) ซึ่งกระแสอาร์เมเจอร์จะต้องคงที่ อย่างไรก็ตามนั่นคือจะเกิดความเสียเปรียบ ความเสียหายที่จะมี คือจะต้องมีแหล่งจ่ายกระแสคงที่ ซึ่งทำได้ยากมาก นั่นคือจะต้องมีแหล่งจ่ายแรงดันคงที่ด้วย เวลาคงที่ของสนามแม่เหล็กควบคุมดีซีมอเตอร์ โดยทั่วไปมีการเปรียบเทียบมากกับเวลาคงที่ของอาร์เมเจอร์ควบคุมอาร์ดีซีมอเตอร์

ดีซีเซอร์โวมอเตอร์ อาจจะถูกขับโดยใช้อิเล็กทรอนิกส์ควบคุม , ความถี่หรือเรียกว่า เซอร์โวไดร์ฟเวอร์เพื่อขับมอเตอร์ เซอร์โวไดร์ฟเวอร์จะควบคุมดีซีเซอร์โวมอเตอร์ให้ทำงานได้หลายอย่าง เช่น จากตำแหน่งหนึ่งเคลื่อนที่ไปอีกตำแหน่งหนึ่ง , ใช้ความเร็วในการเคลื่อนที่ชิ้นงาน และการโปรแกรมใช้เพื่อการเพิ่มความเร็ว การควบคุมทางอิเล็กทรอนิกส์ใช้ pulse-width-modulated ในการขับเซอร์โวมอเตอร์ ซึ่งจะใช้ควบคุมในระบบควบคุมหุ่นยนต์ , ระบบควบคุมเชิงตัวเลข และการควบคุมตำแหน่งต่างๆ และหรือ ระบบควบคุมความเร็ว สิ่งที่กล่าวอาจหมายถึงอาร์เมเจอร์ควบคุมเซอร์โวมอเตอร์

2.3 ระบบเซอร์โว

เมื่อพิจารณาระบบเซอร์โว ในรูปที่ 2.5 (a) จุดประสงค์ของระบบนี้คือ การควบคุมตำแหน่งเครื่องกลให้สอดคล้องกับตำแหน่งอ้างอิง หลักการกล่าวคือ อินพุทโพเทนทิโอมิเตอร์ และเอ้าท์พุทโพเทนทิโอมิเตอร์ เป็นตัวเปรียบเทียบความคลาดเคลื่อน ตำแหน่งทางด้านอินพุทและเอ้าท์พุทจะถูกแปลงไปอยู่ในรูปสัญญาณไฟฟ้า ทางด้านอินพุทสัญญาณไฟฟ้าจะทำให้เกิดมุม (r) โดยการกวาดของแกนภายในอินพุทโพเทนทิโอมิเตอร์ ทางด้านเอ้าท์ก็หลักการเช่นเดียวกันกับทางด้านอินพุท แต่จะเกิดมุม (c) โดยการกวาดของแกนภายในเอ้าท์พุทโพเทนทิโอมิเตอร์ ความแตกต่างระหว่างมุม (r) และมุม (c) คือสัญญาณผิดพลาด (e) หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 (a)

$$e = r - c$$

เป็นไปได้ที่ผลต่างระหว่าง $e_r - e_c = e_v$ คือแรงดันผิดพลาด เมื่อ e_r คือ สัดส่วนของ r และ e_c คือ สัดส่วนของ c นั่นคือ $e_r = K_0 r$ และ $e_c = K_0 c$ เมื่อ K_0 คือ สัมประสิทธิ์คงที่ แรงดันผิดพลาดจะตกคร่อมโพเทนทิโอมิเตอร์ และจะถูกขยายโดยภาคขยาย เกณฑ์การขยายคือ K_1 แรงดันเอ้าท์พุทที่ถูกขยาย แล้วจะไปป้อนให้วงจรอาร์เมเจอร์ของมอเตอร์ (การขยายอาจมีอินพุทอิมพีแดนซ์สูงมากที่โพเทนทิโอมิเตอร์ และมีวงจรอิมพีแดนซ์สูงมาก ซึ่งไม่ก่อให้เกิดการทนต่อกระแสเดรน) เวลาเดียวกันการขยายอาจมีเอ้าท์พุทอิมพีแดนซ์ต่ำ ซึ่งมันจะป้อนกระแสเข้าไปในวงจรอาร์เมเจอร์ของมอเตอร์ แรงดันคงที่นี้จะทำให้เกิดสนามแม่เหล็กพันรอบแกน ถ้าเกิดยังมีความผิดพลาด มอเตอรีจะเพิ่มแรงบิดเพื่อไปขับโหลด ดังนั้นความผิดพลาดจะน้อยลงจนเท่ากับศูนย์ สำหรับกระแสคงที่หาได้จากแรงบิดที่ถูกเพิ่มโดยมอเตอร์คือ

$$T = K_2 i_a$$

เมื่อ K_2 คือ แรงบิดของมอเตอร์คงที่ และ I_a คือ กระแสอาร์เมเจอร์

หมายเหตุ ถ้ากระแส i_a มีการป้อนกลับโดยแรงบิด T ซึ่งมีผลให้การหมุนของส่วนที่หมุนหมุนกลับทิศทาง

เมื่ออาร์เมเจอร์เกิดการหมุนจะเกิดแรงดันขึ้นโดยฟลั๊กซ์ และการเหนี่ยวนำในอาร์เมเจอร์จะเกิดความเร็วเชิงมุมขึ้น แรงดันเหนี่ยวนำ e_b คืออัตราส่วนความเร็วเชิงมุม de/dt หรือ

$$e_b = K_3 \frac{d\theta}{dt} \qquad (1\text{-}1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ e_b คือ แรงเคลื่อนที่เนื่องจากสนามแม่เหล็กย้อนกลับ K_3 คือ แรงเคลื่อนที่เนื่องจากสนามแม่เหล็กคงที่ของมอเตอร์ และ θ คือมุมที่เกิดจากการเคลื่อนที่ของเพลามอเตอร์

ความเร็วในการควบคุมอาร์เมเจอร์ดีซีเซอร์โวมอเตอร์ คือการควบคุมโดยแรงดันอาร์เมเจอร์ e_a (แรงดันอาร์เมเจอร์ $e_a = K_1 e_v$ หรือคือเอ้าท์พุทที่ถูกขยายแล้ว) เมื่อเราดิฟเฟอเรนเชียลสมการของวงจรอาร์เมเจอร์ คือ

$$La \frac{di_a}{dt} + R_a I_a + e_b = e_a$$

$$La \frac{di_a}{dt} + R_a I_a + K_3 \frac{d\theta}{dt} = K_1 e_v \qquad (1\text{-}2)$$

สมการที่แรงบิดสมดุลย์ คือ

$$J_0 \frac{d^2\theta}{dt^2} + b_0 \frac{d\theta}{dt} = T = K_2 i_a \qquad (1\text{-}3)$$

เมื่อ J_0 คือความเฉื่อยรวมของมอเตอร์ , โหลด และเฟืองที่ต่อกับเพลามอเตอร์ และ b_0 คือสัมประสิทธิ์ความเสียดทานจากความหนืดรวมของมอเตอร์ , โหลด และเฟืองที่ต่อกับเพลา เมื่อย้ายฟังก์ชั่นเพลามอเตอร์ที่เคลื่อนที่ และแรงดันผิดพลาด จากสมการที่ (1-2) และ (1-3) ผลที่ได้คือ

$$\frac{\theta(s)}{E_v(s)} = \frac{K_1 K_2}{s(L_a s + R_a)(J_0 s + b_0) + K_2 K_3 s} \qquad (1\text{-}4)$$

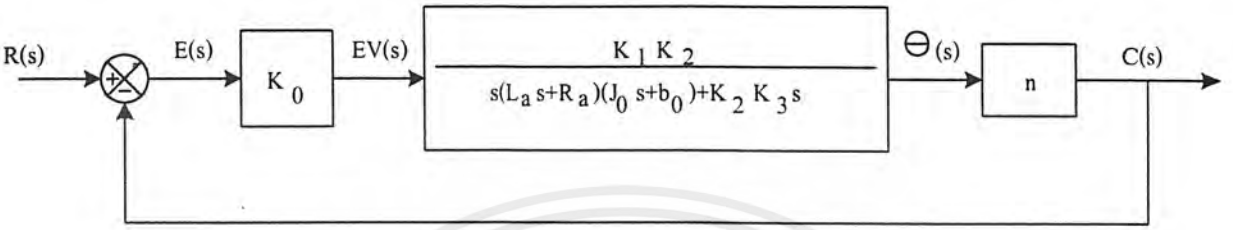
เมื่อ $\theta(s) = L\{\theta(t)\}$ และ $E_v(s) = L\{e_v(t)\}$ เราสมมติว่าอัตราเฟืองของเฟืองทั้งหมดเป็นเอาท์พุทการหมุนของเพลาที่ n เวลาของการหมุนของเพลามอเตอร์ ด้วยเหตุนี้

$$C(s) = N\theta(s) \qquad (1\text{-}5)$$

เมื่อ $C(s) = L\{c(t)\}$ และ $C(t)$ คือมุมการเคลื่อนที่ของเพลาที่หมุน ความสัมพันธ์ระหว่าง $E_v(s)$, $R(s)$ และ $C(s)$ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$E_v(s) = K_0\{R(s) - C(s)\} = K_0 E(s) \qquad (1\text{-}6)$$



รูปที่ 2.5 (b)

เมื่อ R(s) = L{r(t)} บล็อคไดอะแกรมของระบบนี้อาจจะสร้างมาจากสมการที่ (1-4), (1-5) และ (1-6) เหมือนรูปที่ 1(b) การ Transfer function ในทาง Feed Forward ของระบบนี้ คือ

$$G(s) = \frac{C(s)}{\theta(s)}\ \frac{\theta(s)}{E_v(s)}\ \frac{E_v(s)}{E(s)} = \frac{K_0K_1K_2N}{s\{(L_as+R_a)(J_0s+b_0)+K_2K_3}$$

เนื่องจาก L_a มีค่าน้อยมาก อาจจะไม่นำมาคิด และ G(s) ที่ได้จะเปลี่ยนเป็น

$$G(s) = \frac{K_0K_1K_2N}{s\{R_a(J_0s+b_0)+K_2K_3\}}$$

$$= \frac{K_0K_1K_2N/R_a}{J_0s^2+(b_0+\frac{K_2K_3}{R_a})s} \qquad (1\text{-}7)$$

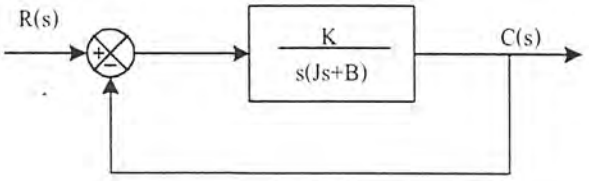
ในเทอม $\{b_0+(K_2K_3/R_a)\}s$ แสดงถึงแรงเคลื่อนที่เนื่องจากสนามแม่เหล็กย้อนกลับของมอเตอร์มีผลทำให้เกิดความหนืดเพิ่มขึ้นของระบบ ความเฉื่อย(J_0) และความหนืด $\{b_0+(K_3K_2/R_a)\}$ จะเกี่ยวโยงไปถึงเพลามอเตอร์ เมื่อ J_0 และ $b_0+(K_2K_3/R_a)$ ถูกคูณด้วย $1/n^2$ ผลที่ได้ของความเฉื่อย และความหนืดในเทอมนี้ของเพลาที่หมุน ทำให้พารามิเตอร์เปลี่ยนเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$J = J_0 / n^2$ เท่ากับขณะที่ความเฉื่อยมีผลต่อการหมุนของเพลา

$B = \{b_0 + (K_2 K_3 / R_a)\} / n^2$

$K = K_0 K_1 K_2 / n\ R_a$



รูปที่ 2.5 (c)

เมื่อ Transfer function G(s) ในสมการที่ (1-7) สามารถทำให้ง่ายขึ้น ผลที่ได้

$$G(s) = \frac{K}{Js^2 + Bs}$$

หรือ

$$G(s) = \frac{K_m}{s(T_m s + 1)} \qquad (1\text{-}8)$$

เมื่อ $K_m = \frac{K}{B}$, $T_m = \frac{J}{B} = \frac{R_a J_0}{R_a b_0 + K_2 K_3}$

บล็อกไดอาแกรมของระบบนี้ ในรูปที่ 2.5 (b) สามารถทำให้ง่ายขึ้นดังในรูปที่ 2.5 (c) ในสิ่งที่ตามมาทำให้ผลตอบสนองเกี่ยวกับพลังงานของระบบนี้ถึงอินพุท แบบ unit-step , unit-ramp และunit-impulse

จากสมการที่ (1-7) และ (1-8) การTransfer function มีระยะเวลา 1/s ในสมการที่(1-8) สังเกตได้ว่าเมื่อเวลาคงที่ R_a และ J_0 จะมีค่าน้อยและจะมีค่าลดลงจนเข้าใกล้ศูนย์

ผลของโหลดตามพลังงานที่ได้จากเซอร์โวมอเตอร์ ส่วนใหญ่ความสำคัญระหว่างคุณสมบัติของเซอร์โวมอเตอร์ คือการที่มีความเร็วสูงในเรื่องแรงบิด เมื่อขณะหมุนความเฉื่อยอาจจะมีค่าสูงมาก เนื่องจากเซอร์โวมอเตอร์กระทำต่อเนื่องหรือเปลี่ยนแปลงตามที่กำหนด เช่น เร่งความเร็ว หรือ ช้าลงของการหมุน เมื่อเวลาหนึ่งไปอีกเวลาหนึ่ง เซอร์โวมอเตอร์อาจจะสามารถดูดกลืนพลังงานกลได้ การกระทำของเซอร์โวมอเตอร์เมื่อใช้เบรคควรจะทำเมื่อต้องการเบรค

J_m และb_m ตามลำดับคือ ขณะที่มีความเฉื่อยและความหนืดเมื่อเกิดการหมุน และ J_L , b_L ตามลำดับ คือขณะที่มีความเฉื่อยและความหนืด เมื่อโหลดได้รับแรงขับจากเพลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

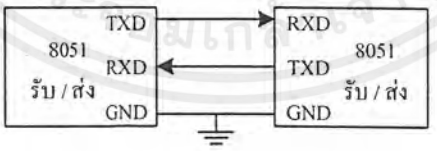
สมมติว่าขณะที่มีความเฉื่อย และความหนืดของเฟืองแต่ละฟัน หรือมี J_L และ b_L ตามลำดับ ดังนั้นขณะที่ความเท่ากันของความเฉื่อย J_{eq} เกี่ยวโยงไปถึงเพลามอเตอร์ และความเท่ากันของความหนืด b_{eq} เกี่ยวโยงไปถึงเพลามอเตอร์ สามารถเขียนได้เป็น

$$J_{eq} = J_m + n^2 J_L$$
$$B_{eq} = b_m + n^2 f_L$$

เมื่อ n(n<1) คือ อัตราเฟืองระหว่างมอเตอร์และโหลด ถ้าอัตราเฟือง n น้อย และ $J_m >> n^2 J_L$ ดังนั้นขณะที่ความเฉื่อยของโหลด เกี่ยวโยงไปถึงเพลามอเตอร์มีค่าน้อยจะมีผลเกี่ยวกับ ขณะที่มีการหมุนจะมีความหนืดเกิดขึ้น ซึ่งก็เหมือนกับการเสียดสีโดยทั่วไป เมื่ออัตราเฟือง n มีค่าน้อย การ transfer function ของ electric servomotor อาจจะเกิดการเสียดสีและความหนืดกับโหลด ถ้า J_m ไม่เท่ากับ $n^2 J_L$ อย่างไรก็ตาม ขณะที่ความเท่ากันของความเฉื่อย J_{eq} อาจจะใช้ประเมินค่า การ transfer function ของมอเตอร์และโหลดโดยรวม

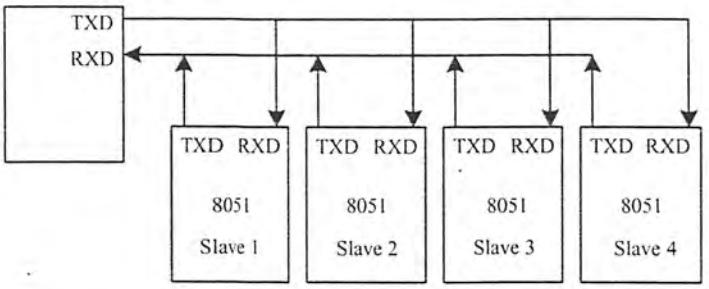
2.4 การสื่อสารระหว่างซีพียูหลายตัว (Multi Processor Communication)

ในการติดต่อสื่อสารทั่วไปบางครั้งต้องการใช้ซีพียูหลายตัว เนื่องจากงานที่ใช้ซีพียูอาจไม่เพียงพอ ซึ่งการสื่อสารระหว่างซีพียูทั่วไปจะใช้การติดต่อสื่อสารแบบจุดซึ่งมีตัวส่ง 1 ตัว และตัวรับอีก 1ตัว หรือใช้ซีพียูมากกว่า 2ตัว เพื่อที่จะทำให้เกิดการควบคุมที่ประสิทธิภาพมากขึ้นดังรูปที่ 2.6



รูปที่ 2.6 (a) การต่อแบบจุดต่อจุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



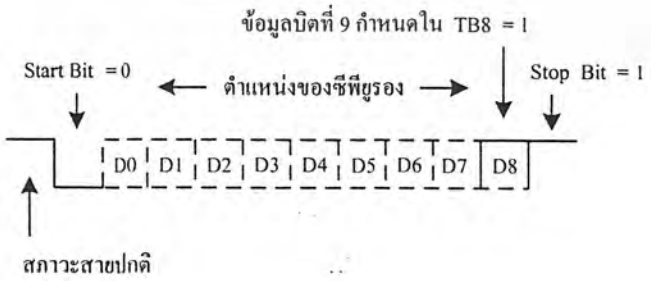
รูปที่ 2.6 (b) การต่อแบบหลายจุด

รูปที่ 2.6 การเชื่อมต่อไมโครคอนโทรลเลอร์จุดต่อจุดและหลายจุด

การทำงานของพอร์ตอนุกรมของ ไมโครคอนโทรลเลอร์ MCS-51 ในโมด 2และโมด 3 มีการทำงานพิเศษ ที่ใช้สำหรับการเชื่อมต่อสื่อสารระหว่างซีพียูหลายตัวได้ การทำงานทั้ง 2 โมดนี้ข้อมูลบิตที่ 9 จะถูกนำไปเก็บในบิต RB8 เราสามารถโปรแกรมเพื่อกำหนดให้พอร์ตอนุกรมส่งสัญญาณร้องขออินเตอร์รัพต์ได้เมื่อได้รับข้อมูลเข้ามาแล้วได้ค่าใน RB8 = 1 ซึ่งการควบคุมนี้เกิดจากการเซตบิต SM2 ที่อยู่ในรีจิสเตอร์ SCON การทำงานในการเชื่อมต่อสื่อสารกับซีพียูหลายตัวจะมีการทำงานลักษณะดังนี้

1. ข้อมูลที่เป็นตัวกำหนดตำแหน่ง (Address byte)

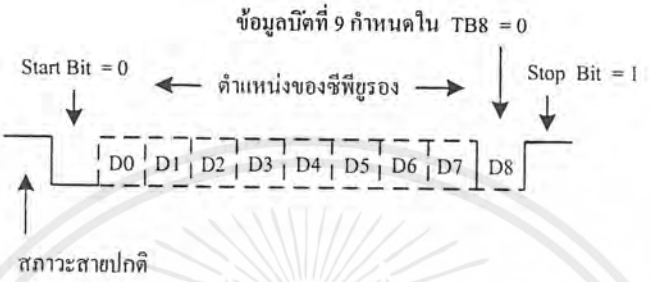
ข้อมูลที่เป็นตัวกำหนดตำแหน่งจะมีค่าบิตข้อมูลบิตที่ 9 (DB) มีค่าเป็น 1 ใช้สำหรับกำหนดตำแหน่งของซีพียูรอง ที่ซีพียูหลักต้องการติดต่อด้วย ลักษณะของข้อมูลที่จะกำหนดตำแหน่งแดงดังรูปที่ 2.7



รูปที่ 2.7 ข้อมูลกำหนดตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ข้อมูลที่ใช้เป็นข้อมูลข่าวสาร (Data Byte)

เป็นไบต์ที่ใช้เป็นข้อมูลสำหรับการรับและส่งข้อมูล และนำไปใช้งานโดยข้อมูลไบต์นี้จะกำหนดให้บิตที่ 9 เป็น 0 ซึ่งรูปแบบของไบต์ข้อมูลจะเป็นดังรูปที่ 2.8

รูปที่ 2.8 ข้อมูลข่าวสาร (Data byte)

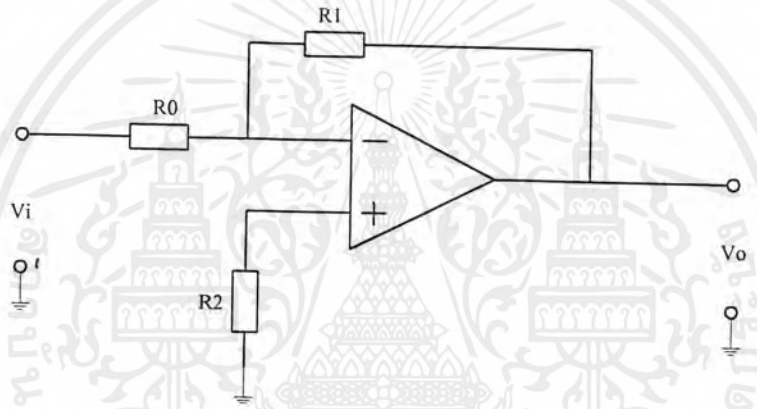
เมื่อซีพียูหลักต้องการส่งข้อมูล 1 บล็อกไปให้กับซีพียูรองตัวใดตัวหนึ่งในหลายๆตัว ขั้นแรกซีพียูหลักต้องส่งค่า 1 ไบต์ ซึ่งถือเป็นตัวบอกตำแหน่งของซีพียูรอง (Address byte) ออกไปก่อนเพื่อเป็นการเลือกซีพียูรองที่ต้องการให้รับข้อมูล โดยข้อมูล 1 ไบต์นี้จะแตกต่างกับข้อมูลอื่นๆ คือ บิตที่ 9 มีค่าเป็น 1 ข้อมูลไบต์นี้ถูกส่งไปยังซีพียูรองทุกตัว หากเรากำหนดให้ซีพียูรองทุกตัวมีค่าบิต SM2 = 1 เมื่อซีพียูรองทุกตัวได้รับข้อมูลไบต์ตำแหน่งแล้วจะเกิดการอินเตอร์รัพต์ขึ้นภายในตัวซีพียูรอง ซีพียูรองจะทำการตรวจสอบว่าตำแหน่งที่รับมานั้นมีค่าตรงกับตัวเองหรือไม่ ซีพียูรองที่มีตำแหน่งตรงกับตำแหน่งที่รับได้จะทำการเคลียบิต SM2 และเตรียมรับข้อมูลที่เป็นข่าวสารต่อไป สำหรับซีพียูรองที่มีตำแหน่งไม่ตรงกับตำแหน่งที่รับมาก็จะยังคงค่า SM2 = 1 อยู่ต่อไป และจบโปรแกรมตอบสนองการอินเตอร์รัพต์แล้วกลับไปทำโปรแกรมที่ค้างอยู่ และรอการอินเตอร์รัพต์ที่จะเกิดขึ้นจากการรับไบต์ตำแหน่งอีก หลังจากซีพียูหลักส่งข้อมูลไบต์แรกซึ่งเป็นตำแหน่งออกไปแล้วจะเริ่มส่งข้อมูลซึ่งถือเป็นไบต์ข้อมูลมีบิตที่ 9 เป็น 0 ตามออกไป ข้อมูลนี้จะมีเพียงซีพียูที่มีตำแหน่งตรงกันเท่านั้นที่รับข้อมูลไว้ การส่งก็จะเป็นระหว่างซีพียูหลักกับซีพียูรองที่มีตำแหน่งตรงกันเท่านั้น เมื่อการส่งข้อมูลสิ้นสุดลงซีพียูหลักจะต้องส่งข้อมูลไบต์สุดท้ายที่ใช้เป็นตัวบอกให้ด้านรับทราบว่าสิ้นสุดข้อมูล ซึ่งอาจใช้รหัส ASCII ค่า 03 (ETX End of text) หรือ 04 (EOT End of transmission) ก็ได้ เมื่อซีพียูรองได้รับรหัสนี้แล้วก็จะทราบว่าเป็นข้อมูลตัวสุดท้ายและจะหยุดการรับข้อมูล แล้วเซตค่าบิต SM2 = 1 เพื่อเริ่มต้นรอรับการส่งข้อมูลที่เป็นตำแหน่งรอบในต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

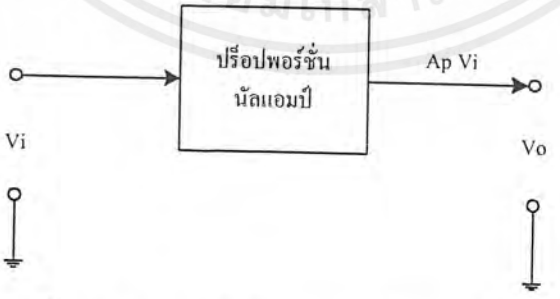
บิต SM2 จะไม่มีผลในโมด 0 สำหรับการใช้งานในโมด 1 หาก SM2 = 1 การอินเตอร์รัพต์ของการ รับข้อมูลจะไม่เกิดขึ้นหากไม่มีการ Stop bit เข้ามา

2.5 P – คอนโทรลเลอร์

P - คอนโทรลเลอร์ได้แก่ลิเนียร์แอมปลิไฟเออร์ซึ่งเป็นพวกออฟแอมป์ที่มีการป้อนกลับด้วยความต้านทาน ดังนั้นค่าเฟสชิฟท์ของมันจึงมีค่าน้อยมาก และอัตราขยายของลูป ของระบบคอนโทรลจะมีค่ามากกว่าหนึ่ง ตัวอย่างของ P- คอนโทรลเลอร์แสดงได้ในรูปที่ 2.9

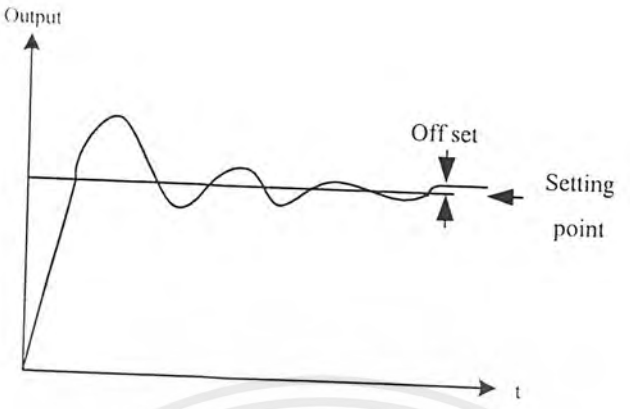


รูปที่ 2.9 แสดงตัวอย่างวงจร P – คอนโทรลเลอร์



รูปที่ 2.10 แสดงบล็อคไดอะแกรม P - คอนโทรลเลอร์

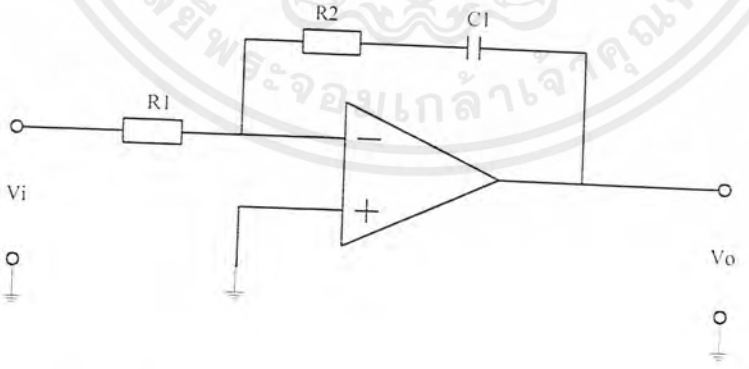
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงผลตอบสนองของสัญญาณ

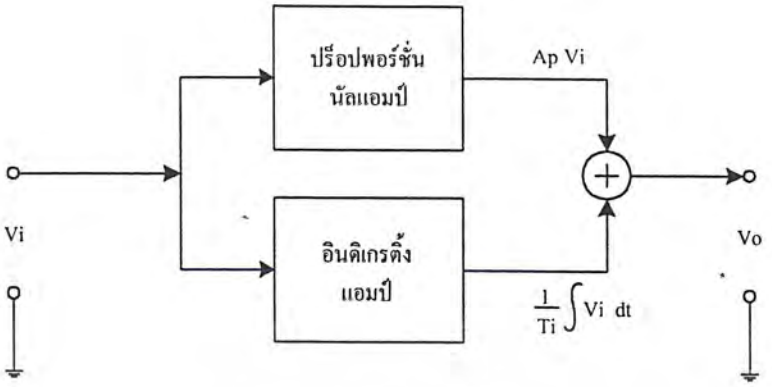
2.6 PI – คอนโทรลเลอร์

PI - คอนโทรลเลอร์สามารถสร้างขึ้นได้ด้วยออฟแอมป์เพียงตัวเดียว อัตราขยายของ P - คอนโทรลเลอร์จะต้องไม่มีค่าสูงเกินไปไม่ยังงั้นระบบจะเกิดการออสซิเลท ตัวอย่างของ PI – คอนโทรลเลอร์แสดงได้ในรูปที่ 2.12

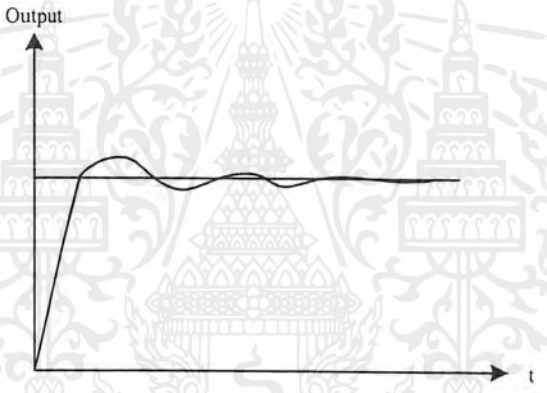


รูปที่ 2.12 แสดงวงจรตัวอย่างของ PI - คอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 แสดงบล็อคไดอะแกรม PI - คอนโทรลเลอร์

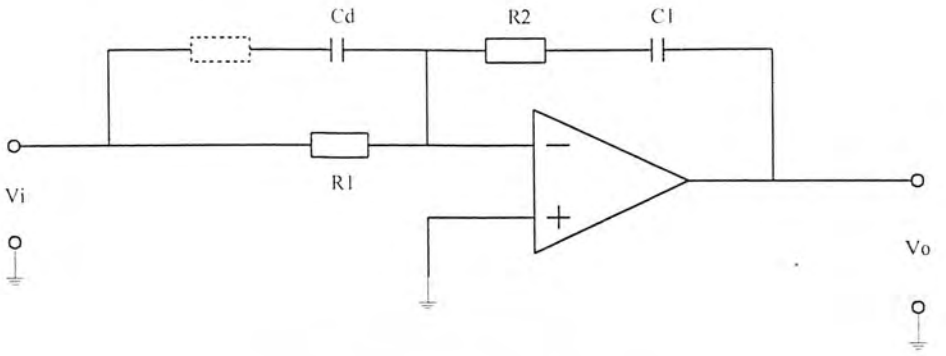


รูปที่ 2.14 แสดงผลตอบสนองของสัญญาณ

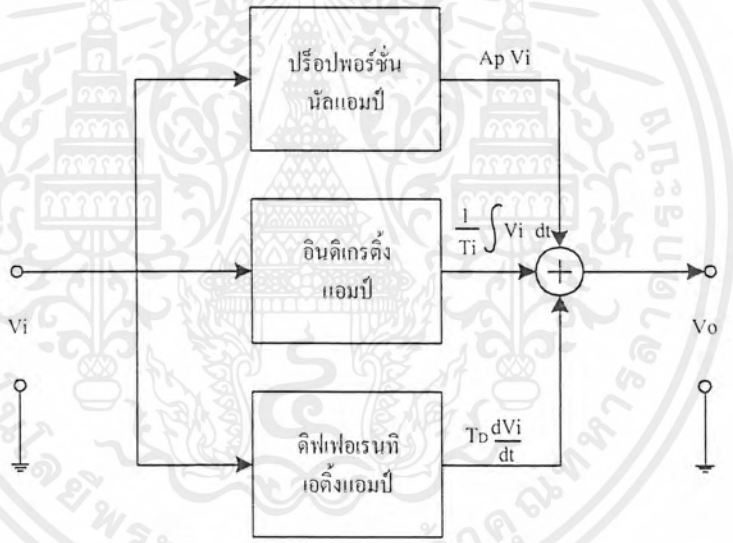
2.7 PID– คอนโทรลเลอร์

เป็นการแก้ไขผลเสียของการควบคุมแต่ละแบบ ซึ่งจะทำให้ Overshoot มีค่าน้อยลงและช่วงเวลาการเกิด Overshoot สั้นลงทำให้มีประสิทธิภาพในการควบคุมมากขึ้น ตัวอย่างของ PID – คอนโทรลเลอร์แสดงได้ในรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



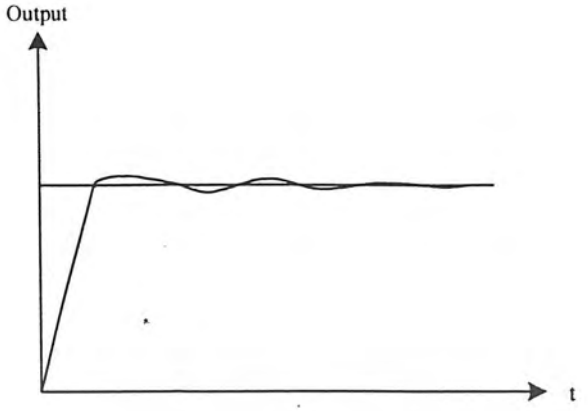
รูปที่ 2.15 แสดงตัวอย่างวงจร PID – คอนโทรลเลอร์



รูปที่ 2.16 แสดงบล็อคไดอะแกรม PID – คอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.17 แสดงผลตอบสนองของสัญญาณ

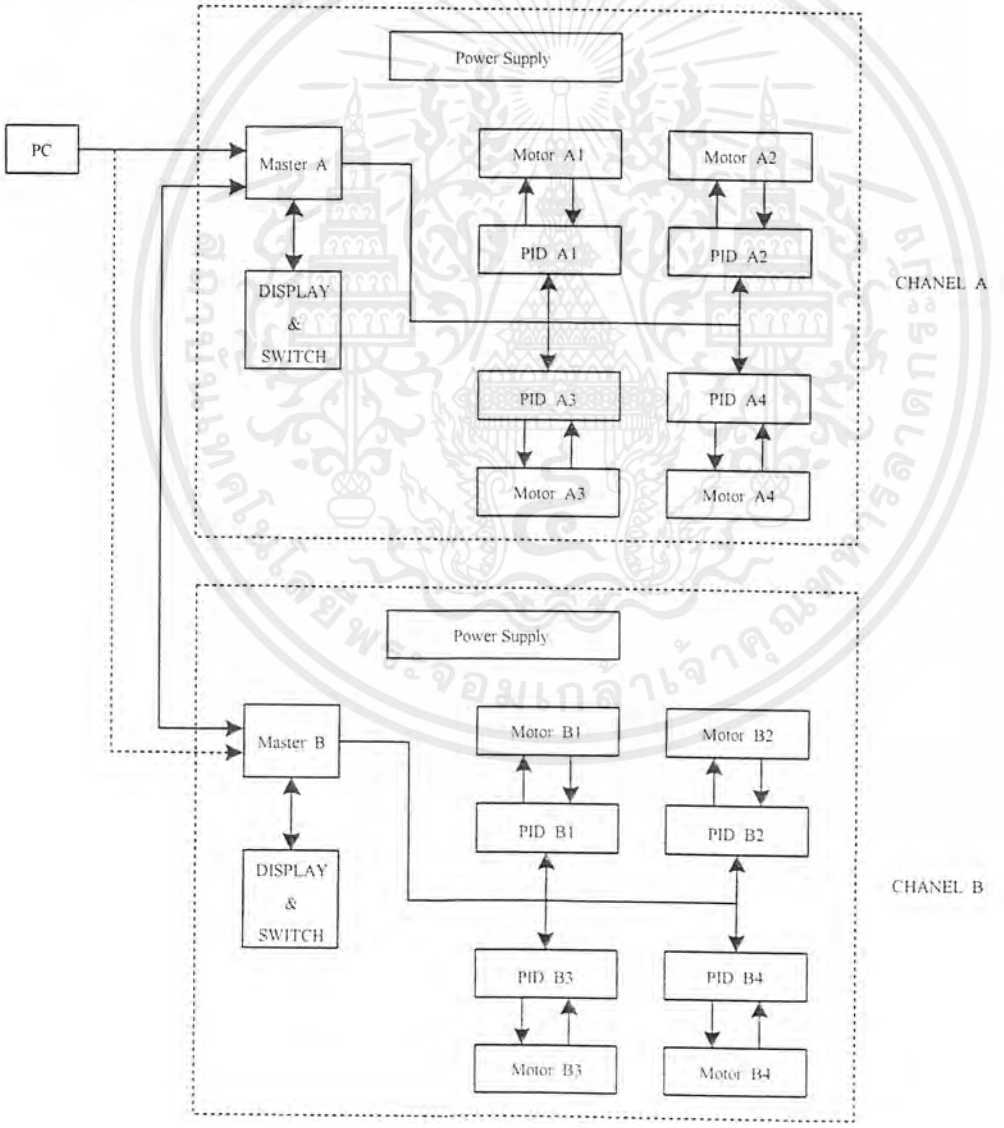
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

42373

บทที่3

การสร้างและการออกแบบ

ในการสร้างโครงงานชุดนี้ จะใช้ไมโครคอนโทรลเลอร์ เบอร์ AT82C52 เป็นหัวใจสำคัญ โดยจะใช้ไมโครคอนโทรลเลอร์ประมวลผลแบบ Multiprocessing ซึ่งในแขนกลแต่ละข้างจะใช้ไมโครคอนโทรลเลอร์5 ตัว ในการประมวลผล และในการควบคุมการเคลื่อนที่ของแขนกลจะใช้ไมโครคอนโทรลเลอร์ประมวลผลตามหลักการของ PID CONTROL ซึ่งแขนกลแต่ละข้างจะมีองค์ประกอบทางฮาร์ดแวร์เหมือนกันทุกประการดังนั้นจึงขอนำมากล่าวเพียงข้างเดียว



รูปที่ 3.1 แสดงบล็อกไดอะแกรมของวงจรทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของแขนกลจะทำงานได้ 2 ลักษณะคือ การทำงานแบบแขนข้างเดียว (Single Arm) โดยแขนทั้งสองข้างทำงานโดยอิสระต่อกัน และการทำงานอีกลักษณะหนึ่งก็คือ การทำงานแบบสองแขนทำงานร่วมกัน (Two Arms) ซึ่งผู้ใช้สามารถเลือกได้ที่โปรแกรม จากบล็อกไดอะแกรมในรูปที่ 3.1 สามารถอธิบายได้ดังนี้

3.1หลักการทำงานของแขนกล

จากบล็อกไดอะแกรมในรูปที่ 3.1 การทำงานจะเริ่มจากผู้ใช้ (User) ป้อนโปรแกรมการทำงานบนเครื่อง PC จากนั้นส่งค่าโปรแกรมดังกล่าวไปที่ CPU ตัวหลัก (ในกรณีที่ใช้งานแบบ Single Arm CPU MASTER (A,B) จะเป็นตัวรับข้อมูลจาก PC แต่ถ้าใช้งานแบบ Two Arms CPU MASTER A จะเป็นตัวรับข้อมูล) จากนั้น CPU ตัวหลักก็จ่ายข้อมูลให้กับ CPU รองที่ทำหน้าที่เป็น PID CONTROL ตามช่วงจังหวะเวลาที่โปรแกรมไว้ และ CPU รอง ก็จะทำการควบคุมให้มอเตอร์หมุนไปตามตำแหน่งที่รับค่ามาจาก CPU ตัวหลัก

การทำงานของวงจรโดยละเอียด

PID CONTROL จะมีทั้งหมด 4 ชุด ต่อแขนหนึ่งข้าง โดยที่ PID1 และ PID2 จะมีการทำงานเหมือนกัน และใช้ DC - SERVO MOTOR ของบริษัท SANYO DENKI รุ่น Supper – U ส่วน PID3 และ PID4 ทำงานเหมือนกัน ซึ่งใช้ DC MOTOR แบบไม่มี Encoder ซึ่งในการอธิบายจะแยกอธิบายเป็น 2 ชุด คือ ชุด PID ที่ใช้กับ PID1 , PID2 และ PID3 , PID4

ชุดควบคุม PID1,2

หลักการทำงาน

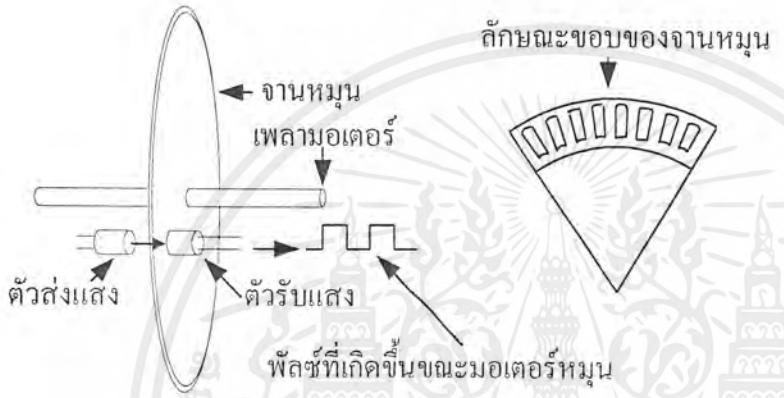
การใช้ไมโครคอนโทรลเลอร์ทำหน้าที่เป็นตัวควบคุมตำแหน่งการหมุนของมอเตอร์โดยใช้หลักการของ PID Control จะต้องมีส่วนประกอบที่สำคัญดังนี้คือ

- ชุดไมโครคอนโทรลเลอร์ ใช้ในการประมวลผลและรับข้อมูลจาก Main controller
- ชุดตรวจจับตำแหน่งของมอเตอร์ เป็นชุดสร้างสัญญาณเพื่อแจ้งตำแหน่งการหมุนของมอเตอร์
- ชุดตรวจจับทิศทางการหมุนของมอเตอร์ จะทำหน้าที่บอกทิศทางการหมุนของมอเตอร์
- ชุดขับมอเตอร์แบบ PWM
- ชุดสร้างพัลซ์เพื่อกำหนดความถี่ของ PWM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสร้างสัญญาณ Error ให้กับ PID Controller

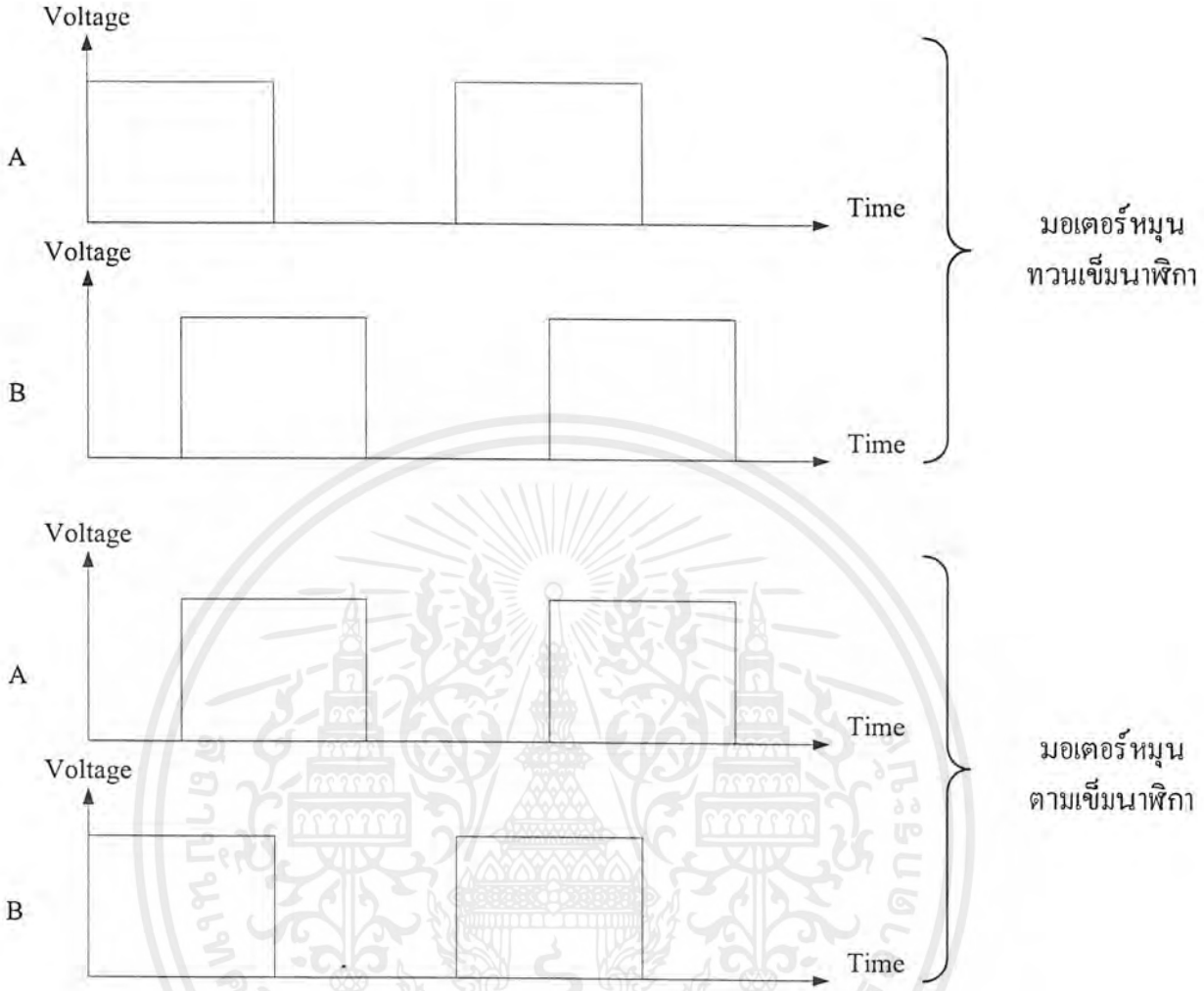
สัญญาณ error ที่จะถูกส่งไปให้ไมโครคอนโทรลเลอร์เพื่อประมวลผลตามหลักการของ PID Control ในการควบคุมให้มอเตอร์หยุดที่ตำแหน่งที่ต้องการจะได้มาจากชุด Encoder ของมอเตอร์ ซึ่งชุด Encoder จะสร้างพัลซ์ออกมาในขณะที่มอเตอร์เกิดการหมุน ดังรูปที่ 3.2



รูปที่ 3.2 แสดงโครงสร้างของ Encoder ของมอเตอร์

ชุด Encoder ที่นำมาใช้ในโครงงานนี้เป็นชุดที่ติดมากับ DC servo motor ของบริษัท Sanyo Denki รุ่น Super U สัญญาณที่ออกมาจาก Encoder มี 6 เส้นคือ $A, B, C, \overline{A}, \overline{B}, \overline{C}$ โดยที่ C จะผลิตพัลซ์ออกมา 1 ลูกเมื่อมอเตอร์หมุนครบ 1 รอบ ส่วนA และ B จะผลิตพัลซ์ออกมา 500 ลูกเมื่อมอเตอร์หมุนครบ 1 รอบ (จากการทดลอง) โดยที่ A และ B จะมีเฟสเหลื่อมล้ำกันอยู่ 90^0 ดังรูปที่ 3.10 ซึ่งระหว่าง A และ B เส้นใดจะมีเฟสนำหน้าหรือล้าหลังขึ้นอยู่กับทิศทางการหมุนของมอเตอร์ คือถ้ามอเตอร์หมุนทวนเข็มนาฬิกา เฟสของ A จะนำหน้า B อยู่ 90^0 และถ้ามอเตอร์หมุนตามเข็มนาฬิกาเฟสของ A จะล้าหลัง B อยู่ 90^0 ซึ่งจากหลักการนี้จะนำมาใช้ในการตรวจจับทิศทางการหมุนของมอเตอร์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



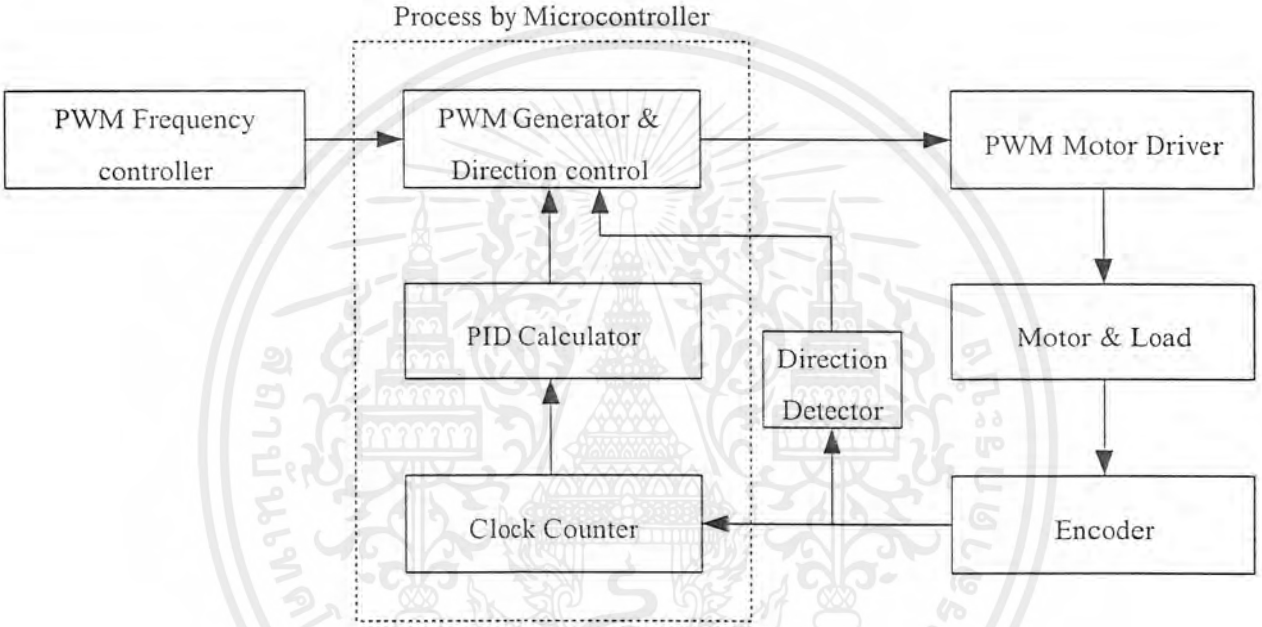
รูปที่ 3.3 แสดงรูปคลื่นสัญญาณของ A และ B เมื่อมอเตอร์หมุนในทิศทางทวนและตามเข็มนาฬิกา

ในขณะที่มอเตอร์หยุดอยู่ที่ตำแหน่งเริ่มต้น Encoder จะยังไม่สร้างพัลซ์ออกมาแต่เมื่อมอเตอร์เริ่มเคลื่อนที่ Encoder จะเริ่มสร้างพัลซ์ออกมาดังที่กล่าวมาแล้วข้างต้น ดังนั้นในการควบคุมให้มอเตอร์หมุนไปที่ตำแหน่งต่างๆจึงสามารถทำได้โดยใช้จำนวนพัลซ์เป็นตัวบอกตำแหน่ง เช่น ต้องการให้มอเตอร์ไปหยุดที่ตำแหน่ง 180^0 จะต้องควบคุมให้มอเตอร์หมุนและ Encoder จะต้องสร้างพัลซ์ออกมา 250 ลูก (ถ้ามอเตอร์หมุนไป 360^0 จะได้พัลซ์ออกมา 500 ลูก) จากนั้นจึงควบคุมให้มอเตอร์หยุดหมุน ซึ่งในการควบคุมตำแหน่งการหมุนของมอเตอร์แบบ PID นี้ จะใช้จำนวนพัลซ์ที่ใช้บอกระยะห่างของตำแหน่งการหมุนของมอเตอร์ มาเป็นสัญญาณ Error ของระบบ เช่น ถ้าต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมให้มอเตอร์หมุนไป 180^0 ซึ่งมีระยะห่างเป็นจำนวนพัลซ์ 250 ลูก ซึ่งจะได้สัญญาณ Error เท่ากับ 250 และเมื่อมอเตอร์เริ่มหมุนไปในตำแหน่งที่ต้องการสัญญาณ Error จะลดลงเรื่อยๆจนกระทั่งถึงตำแหน่งที่ต้องการนั้น จะได้สัญญาณ Error เท่ากับศูนย์ ซึ่งค่า Error ที่ได้นี้จะถูกส่งไปประมวลผลตามหลักการของ PID Control ต่อไป

<u>บล๊อกไดอะแกรมของ PID Controller</u>

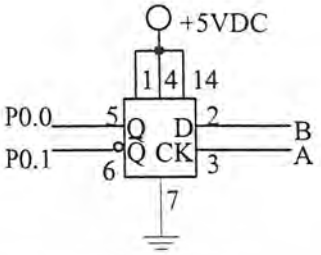


รูปที่ 3.4 แสดงบล๊อกไดอะแกรมของชุด PID Controller

Encoder ทำหน้าที่ตรวจจับตำแหน่งการหมุนของมอเตอร์โดยจะสร้างพัลซ์ออกมาในขณะที่มอเตอร์หมุนดังที่ได้กล่าวมาแล้วข้างต้น

Direction detector ทำหน้าที่ตรวจจับทิศทางการหมุนของมอเตอร์แล้วส่งผลลัพธ์ที่ได้ไปให้กับไมโครคอนโทรลเลอร์ อุปกรณ์ที่ใช้ในการสร้างตัวตรวจจับทิศทางคือ IC 74LS74 ซึ่งเป็น D – Flip-flop TTL ต่อเป็นวงจรดังรูปที่ 3.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 แสดงวงจรตรวจจับทิศทางการหมุนของมอเตอร์

จากรูปที่ 3.5 เป็นวงตรวจจับทิศทางที่ใช้ D-Flip-flop ซึ่งทำงานที่ขอบขาขึ้น เมื่อพิจารณารูปที่ 3.3 ประกอบ จะเห็นว่าเมื่อมอเตอร์หมุนในทิศทางทวนเข็มนาฬิกาจะได้ลอจิก '0' ที่ขา Q และถ้ามอเตอร์หมุนในทิศทางตามเข็มนาฬิกาจะได้ลอจิก '1' ที่ขา Q สัญญาณที่ได้จากขา 5 และขา 6 จะถูกส่งไปประมวลผลที่ไมโครคอนโทรลเลอร์ต่อไป

Clock counter ทำหน้าที่นับจำนวนพัลซ์ที่มาจาก Encoder เพื่อสร้างเป็นสัญญาณ Error ซึ่ง Clock counter จะใช้ Timer 0 ในไมโครคอนโทรลเลอร์ (ตระกูล MCS-51) เป็นตัวนับ

PID Calculator เป็นซอฟต์แวร์ที่รับค่า Error มาจาก Clock counter เพื่อคำนวณตามหลักการของ PID control และส่งค่าผลลัพธ์ที่ได้ไปให้กับ PWM Generator เพื่อสร้างสัญญาณ PWM ไปให้กับชุดขับมอเตอร์แบบ PWM (PWM Motor Driver)

PWM Generator & Direction detector ทำหน้าที่สร้างสัญญาณ PWM ไปให้กับชุดขับมอเตอร์โดยรับค่าผลลัพธ์จาก PID Calculator เพื่อมาเป็นตัวกำหนดความกว้างของพัลซ์ของสัญญาณ PWM และความถี่ PWM จะถูกกำหนดจากชุด PWM Frequency controller โดยสัญญาณ PWM จะถูกส่งออกไปที่ขา P0.7 ของ ไมโครคอนโทรลเลอร์ และอีกหน้าที่หนึ่งคือเป็นตัวสร้างสัญญาณควบคุมทิศทางการหมุนของมอเตอร์ซึ่งสัญญาณนี้จะถูกส่งออกไปที่ขา P2.0 และ P2.1 และส่งออกไปให้ชุดขับมอเตอร์

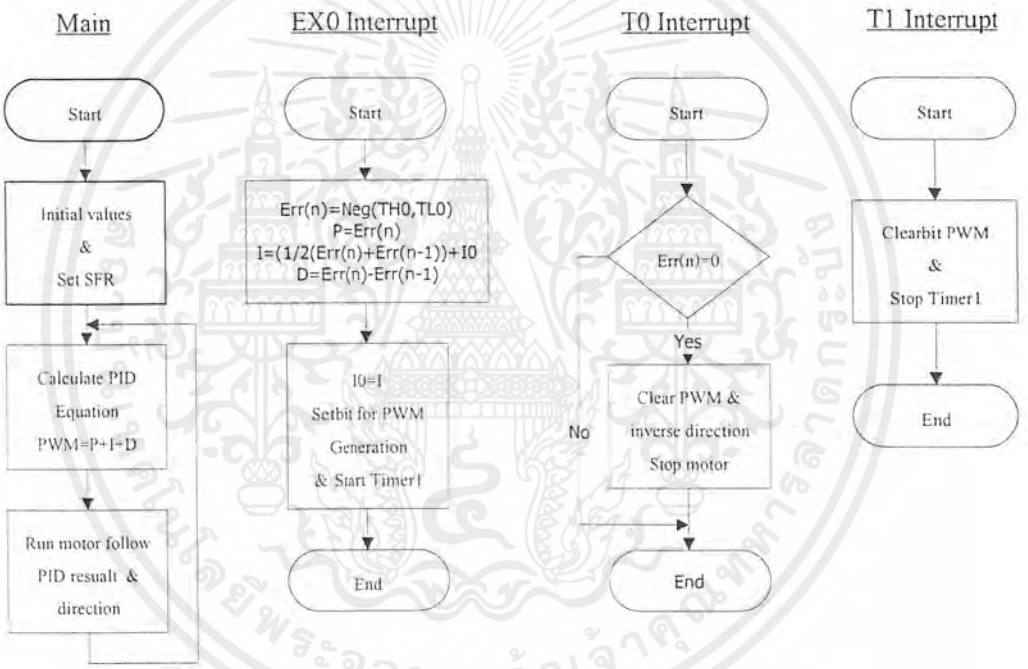
PWM Frequency controller ทำหน้าที่สร้างพัลซ์ความถี่คงที่ให้กับ PWM Generator เพื่อเป็นตัวกำหนดความถี่ของสัญญาณ PWM และทำหน้าที่เป็นตัวกำหนดค่า Sampling time ของการหาค่า Derivative และค่า Integral ในการคำนวณโดยซอฟต์แวร์ อุปกรณ์ที่ใช้ในการสร้างพัลซ์ความถี่คงที่คือ Astable multivibrator โดยใช้ IC 555

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PWM Motor driver ทำหน้าที่ขับมอเตอร์โดยใช้ไอซีสำเร็จรูปเบอร์ L298 ของบริษัท ST Microelectronics โดยจะรับสัญญาณมาจากไมโครคอนโทรลเลอร์สองชุดคือ สัญญาณ PWM และสัญญาณควบคุมทิศทางการหมุนของมอเตอร์

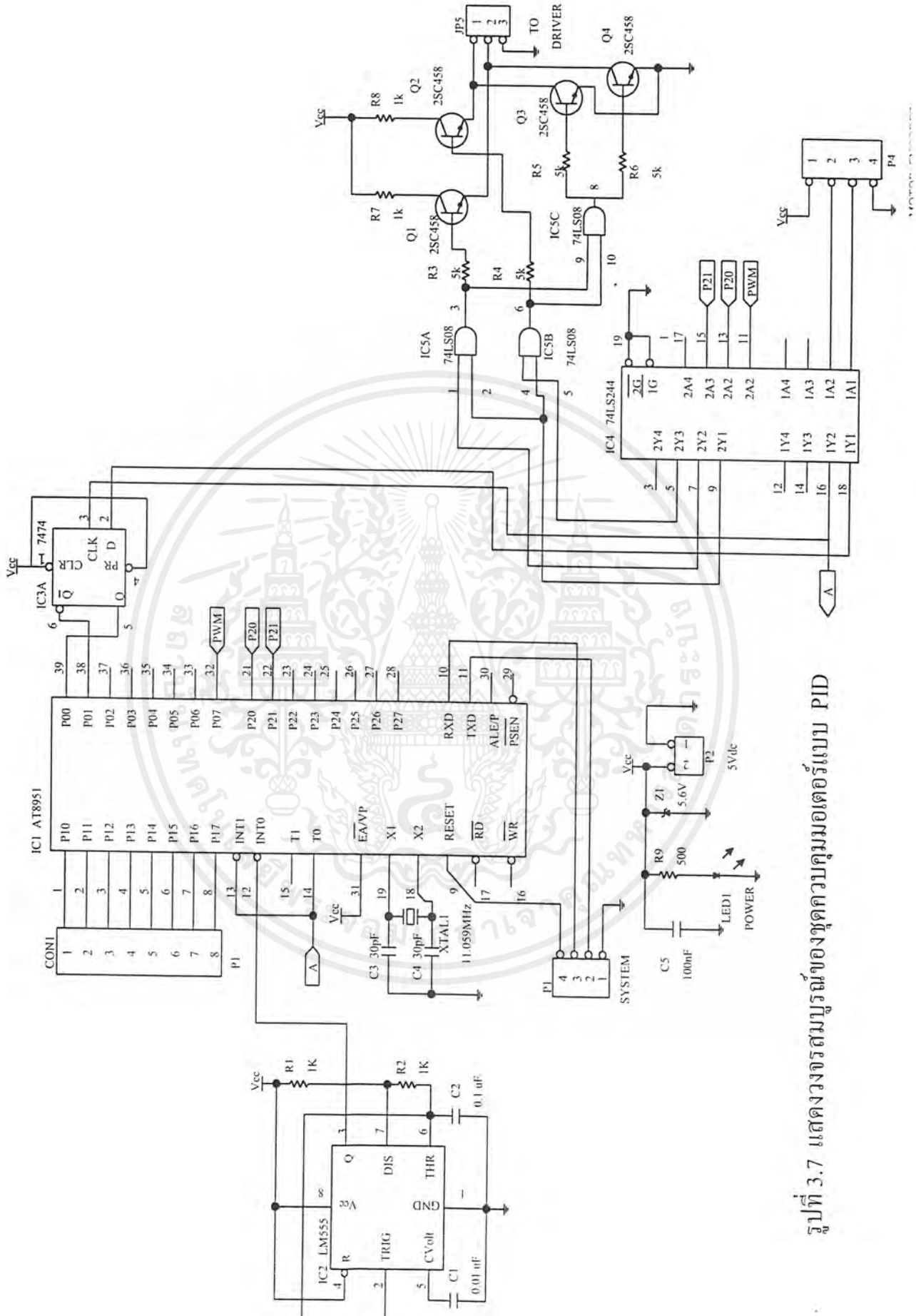
Motor & Load ทำหน้าที่หมุนตัวแขนกล (Load) ให้ไปตำแหน่งที่ต้องการ โดยในโครงงานนี้ใช้ DC Servo motor ของบริษัท Sanyo Denki รุ่น Super U

รูปที่ 3.6 จะแสดงโฟลวชาร์ตการทำงานของโปรแกรม PID1,2 และรูปที่ 3.7 แสดงวงจรสมบูรณ์ของชุด PID1,2



รูปที่ 3.6 แสดงโฟลวชาร์ตการทำงานของโปรแกรม PID Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

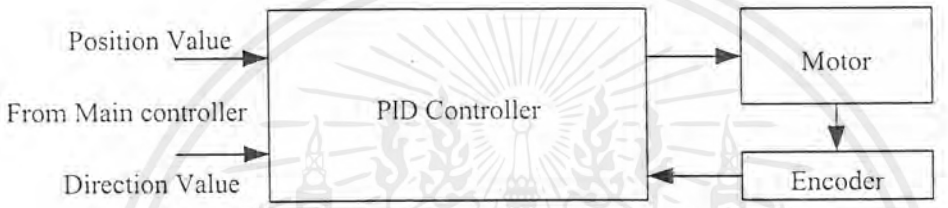


รูปที่ 3.7 แสดงวงจรสมบูรณ์ของชุดควบคุมมอเตอร์แบบ PID

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โปรแกรมควบคุมมอเตอร์แบบ PID

โปรแกรมควบคุมมอเตอร์แบบ PID เป็นโปรแกรมภาษาแอสแซมบลี้ที่ใช้ชุดคำสั่งของไมโครคอนโทรลเลอร์ตระกูล MCS-51 (MCS-51 Standard instruction set) ซึ่งจุดมุ่งหมายของโปรแกรมนี้คือจะรับค่าตำแหน่งการหมุนและทิศทางการหมุนของมอเตอร์มาจากชุดควบคุมหลัก (Main control) และจะควบคุมให้มอเตอร์หมุนไปที่ตำแหน่งและทิศทางนั้นๆ และรักษาตำแหน่งของมอเตอร์ไว้จนกว่าจะได้รับคำสั่งชุดใหม่จากชุดควบคุมหลักดังบล็อกไดอะแกรมต่อไปนี้



รูปที่ 3.8 แสดงบล็อกไดอะแกรมการทำงานของโปรแกรมควบคุมมอเตอร์แบบ PID

<u>การทำงานของโปรแกรมควบคุมตำแหน่งมอเตอร์แบบ PID</u>

โปรแกรมควบคุมตำแหน่งการหมุนของมอเตอร์นี้จะใช้ไมโครคอนโทรลเลอร์เบอร์ AT89C51 เป็นตัวประมวลผล ซึ่งโฟลวชาร์ตได้แสดงดังรูปที่ 3.6 โปรแกรมมีส่วนประกอบต่างๆที่สัมพันธ์กันดังนี้

การตรวจจับตำแหน่งการหมุนของมอเตอร์การทำงานจะเริ่มจากการกำหนดค่าตำแหน่งการหมุนขนาด 16 บิตให้กับตัวแปร POSH และ POSL (Position high byte & Position low byte) และกำหนดค่าทิศทางการหมุนขนาด 8 บิตให้กับตัวแปร DS (Direction set) ซึ่งค่าที่ใช้กำหนดให้กับตัวแปรเหล่านี้จะถูกส่งมาจากชุดควบคุมหลัก (Main controller) ทางพอร์ตอนุกรรมโดยการสื่อสารจะเป็นแบบมัลติโปรเซสเซอร์ ตัวอย่างเช่นต้องการให้มอเตอร์หมุนไปที่ตำแหน่ง 255 ของพัลซ์จากเอนโคดเดอร์ และหมุนในทิศทางตามเข็มนาฬิกา จะต้องกำหนดค่าให้กับตัวแปรดังนี้

POSH = 00H

POSL = 0FFH

DS = 01H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทำ 2'S Complement ค่าของตัวแปร POSH และ POSL และนำค่าที่ได้ไปเก็บที่รีจิสเตอร์ TH0 และ TL0 ของไทม์เมอร์ 0 ตามลำดับ โดยที่ ไทม์เมอร์ 0 (Timer 0 : T0) จะถูกกำหนดการทำงานเป็นตัวนับขนาด 16 บิต (16 Bits Counter : Mode 1) และเมื่อมอเตอร์เริ่มหมุนและ Encoder เริ่มจ่ายพัลซ์ออกมา T0 จะทำการนับไปเรื่อยๆจนกระทั่งเกิดโอเวอร์โฟลวซึ่งจะเป็นจุดที่มอเตอร์จะต้องหยุดหมุน

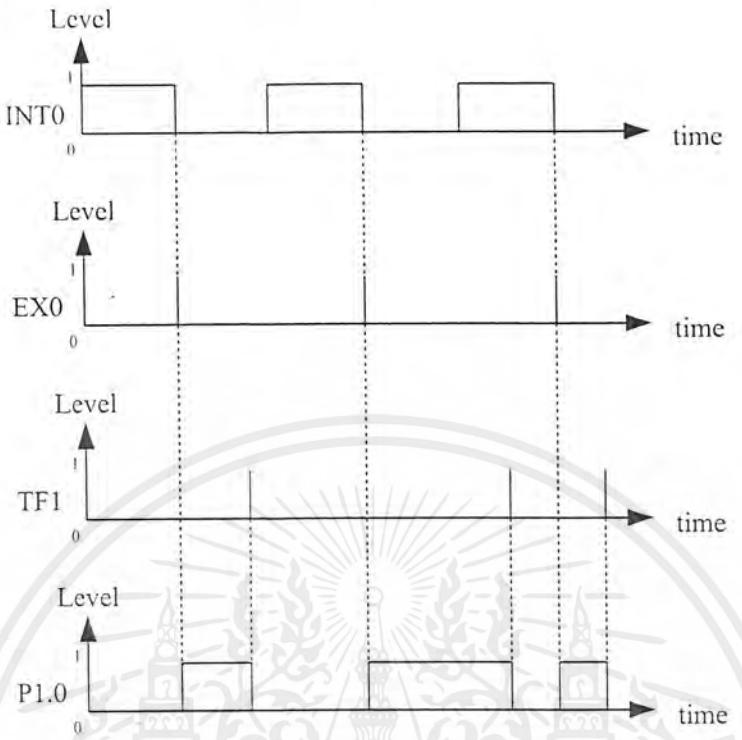
สรุปในการตรวจจับตำแหน่งของมอเตอร์จะใช้ T0 เป็นตัวตรวจจับโดยใช้จุดโอเวอร์โฟลวเป็นจุดที่ต้องการให้มอเตอร์หยุดหมุน

<u>การสร้างสัญญาณ PWM</u>

การสร้างสัญญาณ PWM (Pulse Width Modulation) จะเริ่มจากการสร้างพัลซ์จากภายนอกป้อนเข้ามาที่ขา INT0 ซึ่งที่ External interrupt 0 จะเซ็ทให้มีการอินเตอร์รัพต์ที่สัญญาณขอบขาลงที่ขา INT0 และที่โปรแกรมบริการอินเตอร์รัพต์ของ EX0 (External interrupt 0 Service routien 0 : EX0ISR) จะทำการ Set บิต P0.7 และจะเริ่มสตาร์ทการทำงานของ Timer 1 ซึ่งทำงานในโหมดตั้งเวลา 16 บิต จนกระทั่ง Timer 1 เกิดโอเวอร์โฟลว ที่โปรแกรมบริการอินเตอร์รัพต์ของ Timer 1 จะทำการ Clear บิต P0.7 ซึ่งจะทำให้เกิดสัญญาณ PWM ออกมาที่ขา P0.7

สรุปคือความถี่ของพัลซ์ที่ป้อนเข้ามาจากภายนอกจะเป็นความถี่ของสัญญาณ PWM และจะใช้ Timer 1 (T1) เป็นตัวกำหนดความกว้างของพัลซ์ ซึ่งการกำหนดความกว้างของพัลซ์จะเริ่มจากการคำนวณหาค่า PID และนำค่าผลลัพธ์ของ PID มากำหนดให้กับรีจิสเตอร์ TH1 และ TL1 ของ T1 ซึ่งทำงานในโหมดตัวจับเวลาโหมด 1 (16 Bits Timer : Mode 1) ดังนั้นค่าความกว้างของพัลซ์ของสัญญาณ PWM จึงขึ้นอยู่กับค่าผลลัพธ์ของ PID ที่คำนวณได้ ดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



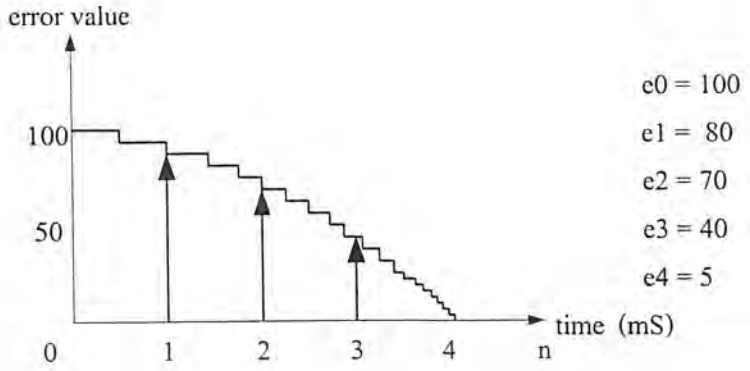
รูปที่ 3.9 แสดงการสร้างสัญญาณ PWM

การสร้างสัญญาณ PID

การสร้างสัญญาณ PID เริ่มจากการสุ่มค่าของสัญญาณ error เป็นช่วงๆตามวิธีการต่อไปนี้

- การสุ่มสัญญาณ เป็นการสุ่มเอาค่า error (สัญญาณ error ที่กล่าวมาแล้วข้างต้น)ที่ช่วงเวลาต่างๆเป็นช่วงๆ ค่าที่ได้จะนำไปใช้ในการคำนวณหาค่า P,I และ D โดยความถี่ของการสุ่มสัญญาณสำหรับในโครงงานนี้ใช้ความถี่เดียวกับความถี่ของสัญญาณ PWM คือ 1 kHz ดังนั้นช่วงเวลาของการสุ่มสัญญาณคือ 1 mS ตัวอย่างของการสุ่มสัญญาณสามารถแสดงได้ดังรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 แสดงตัวอย่างการสุ่มสัญญาณ error

หมายเหตุ จากการทดลองพบว่าเมื่อมอเตอร์หมุนที่ความเร็วสูงสุด Encoder ของมอเตอร์จะผลิตความถี่ (ความถี่สูงสุด) ออกมา 14 kHz ดังนั้นถ้าช่วงของการสุ่มสัญญาณเป็น 1 mS สัญญาณ error จะเปลี่ยนแปลงได้สูงสุดเป็น 14 ซึ่งหลักการนี้จะนำไปพิจารณาในการหาค่า Differential (D)ด้วย

- การหาค่า P (Proportional value) ค่า P จะได้จากค่า error ที่สุ่มมาโดยตรงดังนี้

$$P = en \qquad (3.1)$$

P = Proportional value

en = ค่า error จากการสุ่มลำดับที่ n

ในโปรแกรมภาษาแอสแซมบลี้จะเป็นส่วนหนึ่งของโปรแกรมบริการอินเตอร์รัพต์ของ External interrupt 0 ดังนี้

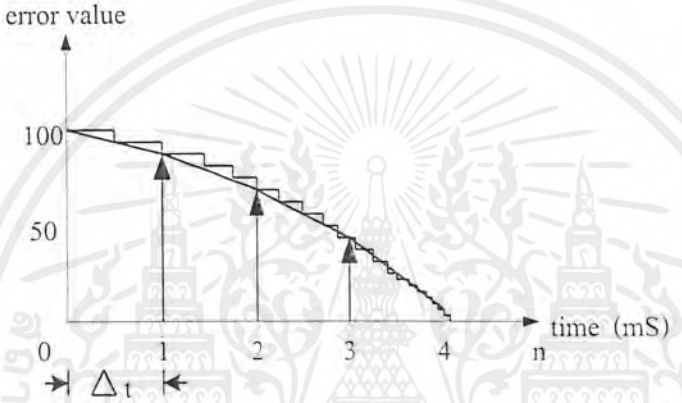
```
        MOV   A,POSH      ; โหลดค่า error 8 บิตด้านสูงมาที่ Acc
        JNZ   ADJ0        ; ตรวจสอบค่า 8 บิตด้านสูงว่ามากกว่า 0 หรือไม่
        MOV   PP,POSL     ; เก็บค่า error ที่ได้ใน PP กรณีที่ error มีขนาดไม่เกิน 8 บิต
END0:   ----------
ADJ0:   MOV   PP,#0FFH    ; กรณีนี้ error เกิน 8 บิตต้องปรับให้เป็นค่าสูงสุดขนาด 8 บิต
        MOV   A,#0FFH
        MOV   ERR0,0FFH
        SJMP  END0
```

* ค่าของ POSH และ POSL ที่นำมาคิดจะเป็นค่าที่ได้จากการทำ 2'S Complement ของ Timer 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมจะเห็นว่าค่าของ error ที่มีขนาด 16 บิตเมื่อนำมาใช้เป็นค่า Proportional (P:PP) จะลดขนาดลงเหลือ 8 บิตเนื่องจากการเขียนโปรแกรมคำนวณทางคณิตศาสตร์ที่ใช้ตัวแปรขนาด 8 บิตจะทำได้ง่ายกว่าและใช้เวลาในการประมวลผลน้อยกว่าการเขียนแบบใช้ตัวแปรขนาด 16 บิต ซึ่งเวลาที่ใช้ในการประมวลผลของโปรแกรมทั้งหมดจะต้องน้อยกว่าเวลาของการสุ่มสัญญาณคือ 1 mS

- <u>การหาค่า I (Integral value)</u> การหาค่าอินทิกรัลสามารถหาได้จากผลรวมของพื้นที่ใต้กราฟดังนี้



รูปที่ 3.11 แสดงการหาค่าอินทิกรัลด้วยพื้นที่ใต้กราฟ

จากรูปจะได้พื้นที่ใต้กราฟในช่วงหนึ่งๆดังนี้

$$\text{Area} = \frac{1}{2}(\text{en} + \text{en-1}) * \Delta t$$

และสมการของพื้นที่ใต้กราฟทั้งหมดซึ่งเป็นค่าของอินทิกรัลคือ

$$I = \sum_{n=0}^{n=\infty} \frac{1}{2}(en + en-1) * \Delta t$$

โปรแกรมที่ใช้คำนวณคือ

```
        MOV   B,#02H
        MOV   A,POSL
        ADD   A,ERR0
        JC    ADJ0
        MOV   ERR0,A
END0:   DIV   AB
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD   A,I0
        JC    ADJ1
        MOV   I,A
        MOV   I0,A
ADJ0:   ---------- (เหมือนกันกับการหาค่า P)
```

จากโปรแกรมจะเห็นว่าไม่ได้คูณค่า Δt เข้าไปในสมการด้วย เนื่องจากค่า Δt มีค่าน้อยมาก คือ 0.001 Sec เมื่อคูณเข้าไปแล้วจะทำให้ค่าของ I ลดลงและในการเขียนโปรแกรมคำนวณแบบเลขทศนิยมขนาด 16 บิต จะทำให้โปรแกรมใช้เวลาในการประมวลผลเพิ่มขึ้นมากซึ่งอาจมีผลทำให้โปรแกรมทั้งหมดทำงานได้ไม่ทันในเวลาหนึ่งช่วงการสุ่มสัญญาณ ดังนั้นจะได้เป็นสมการดังนี้

จาก

$$I = \sum_{n=0}^{n=\infty} \frac{1}{2}(en + en-1) * \Delta t$$

$$I = 0.001 * \sum_{n=0}^{n=\infty} \frac{1}{2}(en + en-1)$$

ถ้าเปลี่ยนเป็น

$$I = \sum_{n=0}^{n=\infty} \frac{1}{2}(en + en-1)$$

แสดงว่ามีการคูณสมการเดิมด้วย 1000 นั่นคือโปรแกรมจะมีเกนเริ่มต้นที่ 1000 ซึ่งจากการทดลองปรากฏว่าใช้งานได้แต่จะเกิดแดมพ์ขึ้น

<u>การหาค่า D (Differential value)</u>

การหาค่าดิฟเฟอเรนเชียลจะทำได้โดยการหาผลต่างของสัญญาณ error ระหว่างช่วงของการสุ่มสัญญาณ ซึ่งเขียนเป็นสมการได้ดังนี้

$$D = \left| e_n - e_{n-1} \right|$$

จากโปรแกรมจะใช้โปแกรมบริการอินเตอร์รัพต์ของ EX1 เป็นตัวหาค่าผลต่างของสัญญาณ error ซึ่งสัญญาณอินเตอร์รัพต์จะใช้สัญญาณเดียวกันกับสัญญาณอินเตอร์รัพต์ของ T0 โดยที่โปรแกรมจะเริ่มนับจำนวนพัลซ์เมื่อเกิดอินเตอร์รัพต์ที่ INT0 และจะหยุดนับเมื่อเกิดการอินเตอร์รัพต์ที่ INT0 ครั้งต่อไปอีกเช่นกันซึ่งจะทำให้ได้ค่าผลต่าง

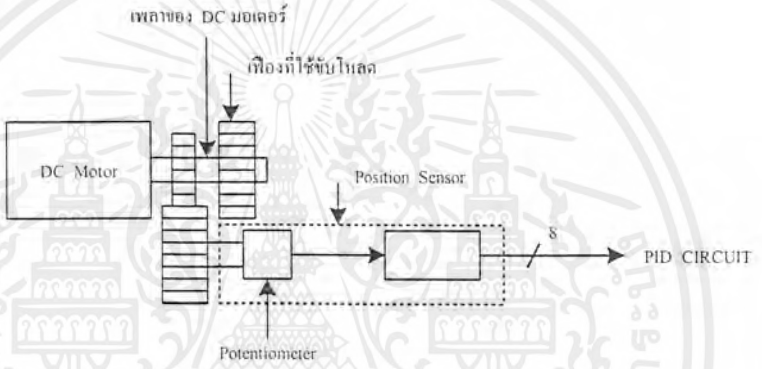
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า PID ที่หามาได้ดังกล่าวข้างต้นเมื่อนำมาใช้งานจริงต้องมีการปรับแต่งด้วยการคูณด้วยค่าคงที่เพื่อให้เกิดความเหมาะสมกับมอเตอร์และโหลดซึ่งค่าคงที่นี้คือค่าเกนของ PID ดังนั้นสมการของ PID คือ

PID = KP*P + KI*I + KD* D

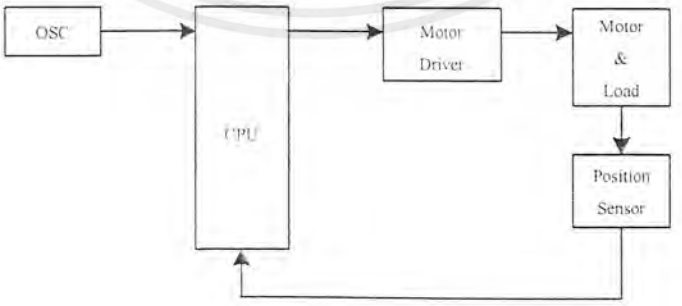
<u>ชุดควบคุม PID3,4</u>

เนื่องจาก PID CONTROLLER ชุดนี้จะใช้กับ DC MOTOR ที่ไม่มี Encoder ดังนั้นจึงต้องเพิ่มชุดตรวจจับตำแหน่งของมอเตอร์ (Position Sensor) เข้าไปโดยการต่อโพเทนทิโอมิเตอร์กับเพลามอเตอร์ ดังรูปที่ 3.12



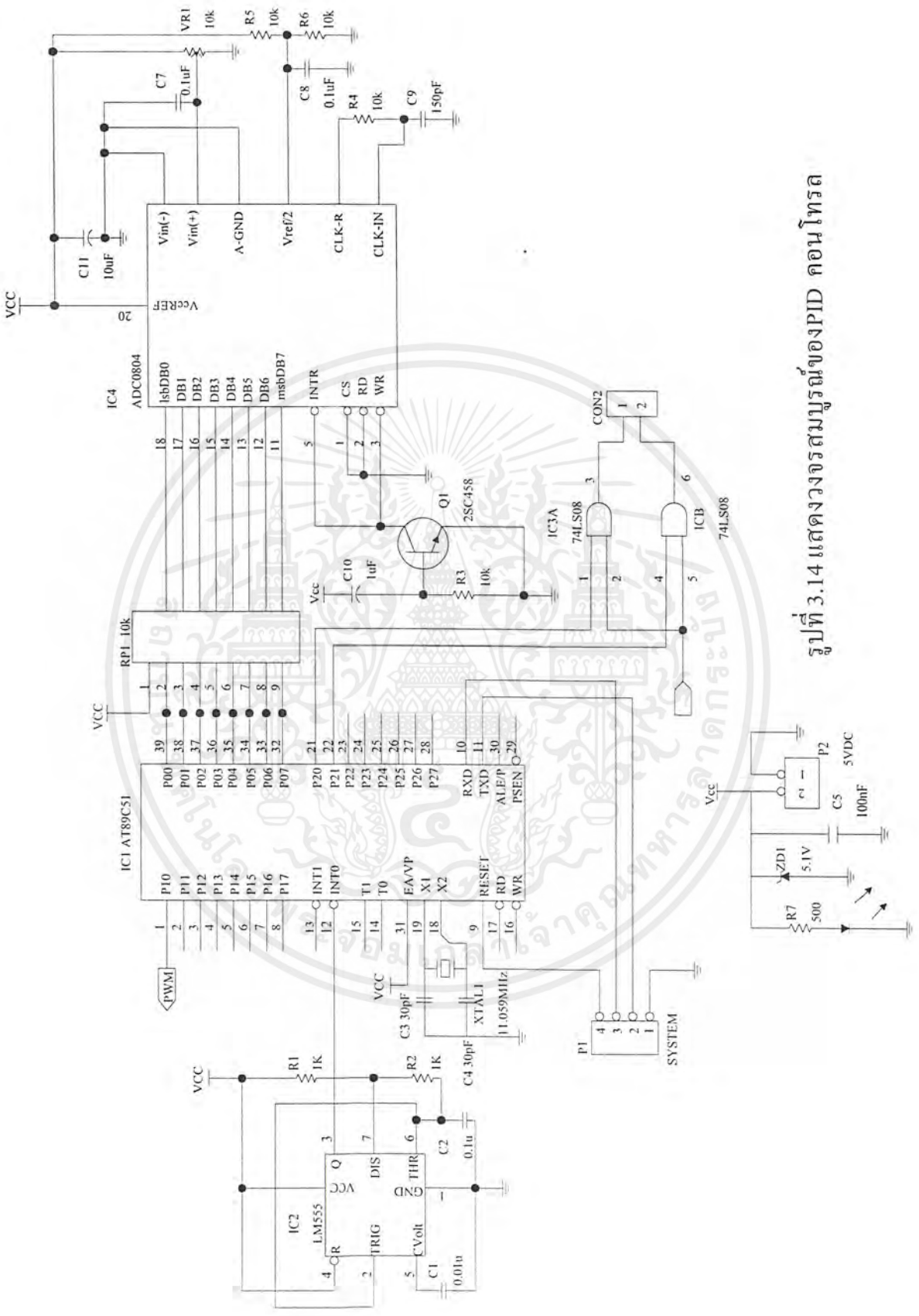
รูปที่ 3.12 แสดงการต่อโพเทนทิโอมิเตอร์เข้ากับเพลามอเตอร์

<u>บล็อกไดอะแกรมของชุด PID3,4</u>



รูปที่ 3.13 แสดงบล็อกไดอะแกรมของชุด PID3,4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 แสดงวงจรสมบูรณ์ของPID คอนโทรล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

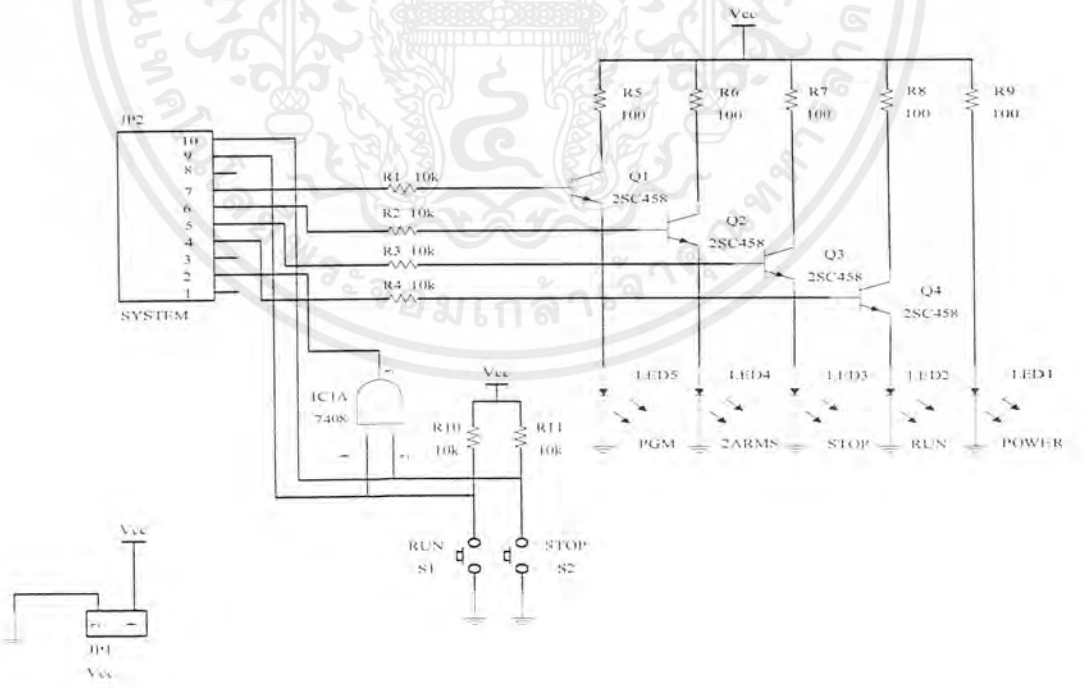
การทำงาน

หลักการทำงานส่วนใหญ่เหมือนกับ PID CONTROLLER ชุดที่แล้ว คือ จะใช้สมการในการคำนวณเหมือนกัน และใช้สัญญาณจากภายนอกที่มาจากวงจรอะสเตเบิ้ลมัลติไวเบรเตอร์ (Astable multivibrestor) เป็นตัวกำหนดความถี่ของสัญญาณ PWM และเป็นตัวกำหนดอัตราการสุ่มตัวอย่างสัญญาณ (Sampling rate) เหมือนกัน ส่วนที่ต่างกันคือ ชุด Position Sensor ที่ใช้โพทนทิโอมิเตอร์ทำงานร่วมกับ ADC (Analog to Digital Convertor) เป็นตัวบอกตำแหน่งของมอเตอร์ ให้กับชุดคำนวณ PID ซึ่งเป็นซอฟแวร์ ซึ่งวงจรแสดงได้ดังรูปที่ 3.14

การทำงานจะเริ่มจากการที่โปรแกรมจะทำการสุ่มค่าสัญญาณ error ที่ส่งมาจาก ADC แล้วนำมาเปรียบเทียบกับค่าเซทพอยท์ (Set point) ซึ่งเป็นค่า I / P ที่รับมาจาก CPU หลัก (Master CPU) จากนั้นผลที่ได้มาคำนวณตามหลักการของ PID CONTROL ซึ่งจะเหมือนกับหลักการของ PID ชุดที่แล้วมา

DISPLAY & SWITCH

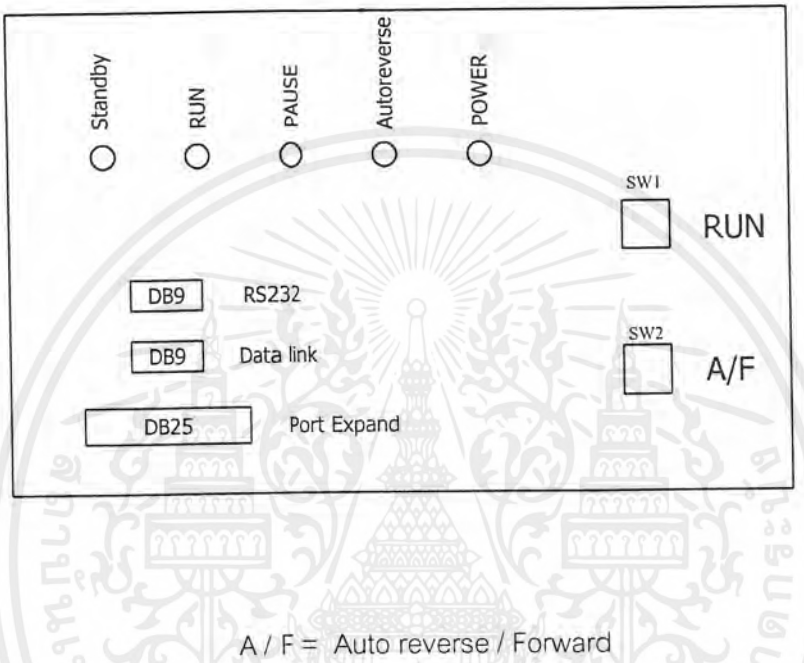
จะทำหน้าที่แสดงผล และควบคุมการทำงานของแขนกล ซึ่งการควบคุมแขนกลจะควบคุมได้สองทางคือ ควบคุมจากสวิตซ์ที่อยู่บนตัวแขนกล และควบคุมจากโปรแกรม INTERFACE จาก PC ซึ่งวงจรแสดงได้ดังรูปที่ 3.15



รูปที่ 3.15 แสดงวงจร Switch & Display

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของ DISPLAY & SWITCH จะประกอบไปด้วย LED ที่ทำหน้าที่เป็นตัวแสดงสถานะการทำงานของแขนกล 5 หลอด และสวิตซ์ ควบคุม 2 ตัวดังรูปที่ 3.16



รูปที่ 3.16 ปุ่มควบคุมและแสดงผลที่ด้านหลังของแขนกล

LED Indicator

ประกอบด้วย LED 3 หลอด ซึ่งจะใช้แจ้งสถานะการทำงานของแขนกลดังนี้

POWER ใช้แสดงสถานะของไฟเลี้ยงวงจรทั้งหมด ถ้า LED สว่างแสดงว่ามีไฟเลี้ยงวงจร

STANDBY ถ้า LED นี้กระพริบด้วยความถี่ประมาณ 0.5 Hz แสดงว่าแขนกลกำลังอยู่ในสภาวะการหาตำแหน่งเริ่มต้น ซึ่งสภาวะนี้แขนกลยังไม่พร้อมที่จะรับข้อมูลหรือรับคำสั่งจาก User แต่ถ้า LED stand by ติดสว่างตลอด แสดงว่า แขนกลพร้อมที่จะรับคำสั่งจาก User แล้ว

RUN เป็น LED ที่ใช้แสดงสภาวะขณะกำลังทำงานตาโปรแกรมของเครื่อง คือ ถ้า LED หลอดนี้สว่างแขนกลกำลังอยู่ในช่วงของการทำงานตามโปรแกรมที่ตั้งไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAUSE เป็น LED ที่ใช้แสดงสถานะของการหยุดการทำงานชั่วคราวของแขนกล ซึ่งถ้า LED หลอดนี้สว่างแสดงว่าจะหยุดทำงานชั่วคราว จนกว่าจะได้รับคำสั่ง RUN มาจาก PC หรือสวิตช์ควบคุม

A / F (Auto reverse / Forward) เป็น LED แสดงสถานะของการวนรอบโดยอัตโนมัติ คือ ถ้า LED สว่างแสดงว่า ถ้าแขนกลทำงานตามโปรแกรมไปจนถึงข้อมูลบล็อกสุดท้ายของโปรแกรมมันจะกลับไปเริ่มทำงานตามโปรแกรมของข้อมูลบล็อกแรกโดยอัตโนมัติแต่ถ้า LED หลอดนี้ไม่ติดแสดงว่าถ้าโปรแกรมทำงานไปจนถึงบล็อกสุดท้ายแล้ว โปรแกรมจะกลับไปสู่สถานะ Stand by ถึงหยุดทำงานตามโปรแกรมและแขนกลจะกลับไปอยู่ที่ตำแหน่งเริ่มต้นใหม่อีกครั้ง

Control Switch มี 2 ปุ่ม คือ

RUN เป็นสวิตช์ที่ทำงานได้สองหน้าที่คือ กด RUN และกดให้ PAUSE ซึ่งจะสังเกตได้จาก LED Indicator ถ้าต้องการหยุดให้กดสองปุ่มพร้อมกัน

A/F เป็นสวิตช์ควบคุมให้เครื่องเข้าสู่สภาวะของโหมด Autoreverse โดยสังเกตได้จาก LED Autoreverse จะสว่างเมื่อแขนกลเข้าสู่โหมด Autoreverse

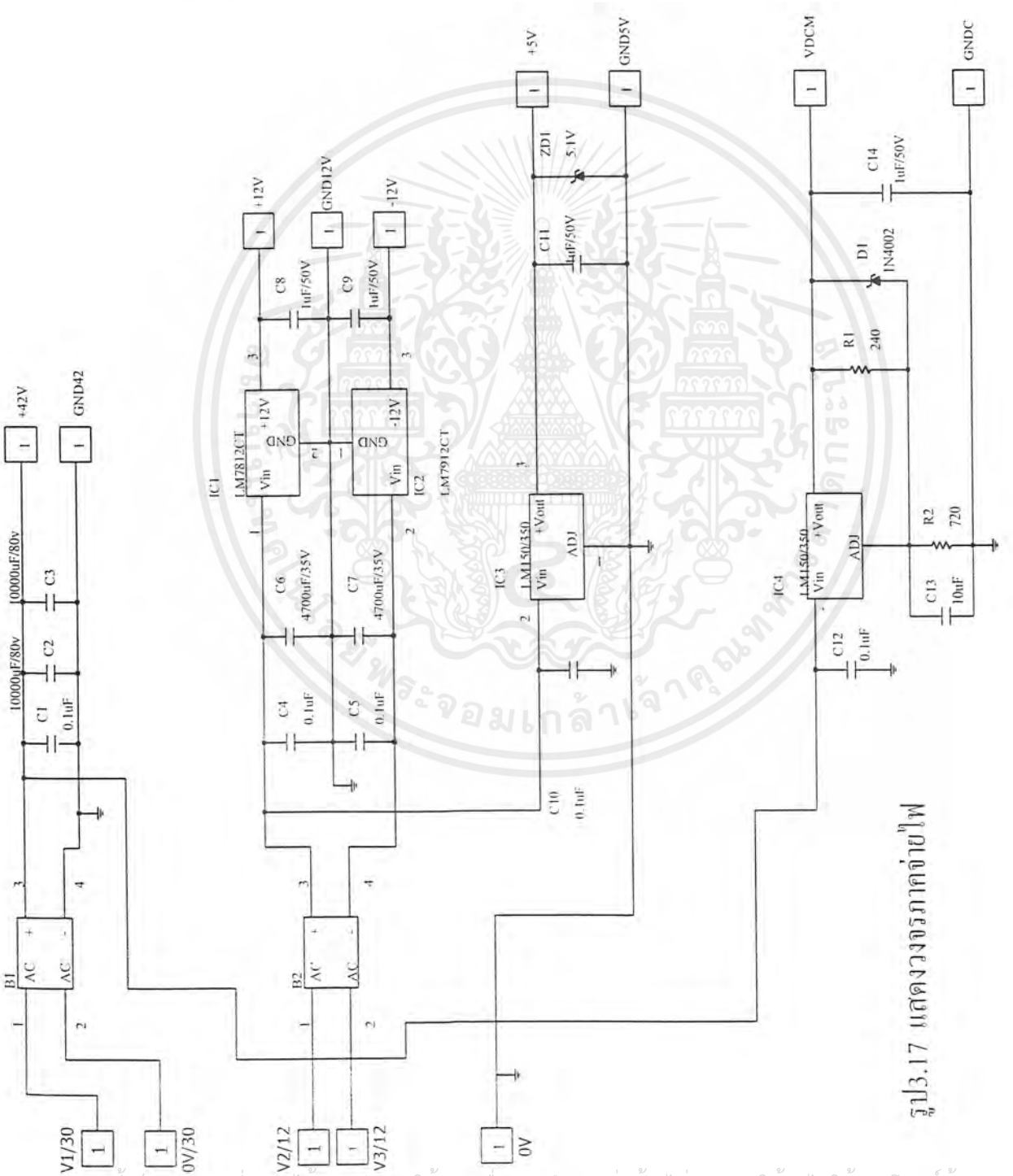
ส่วนจุดต่อที่เป็น Conector แบบ DB9 และ DB25 นั้นจะมีหน้าที่ดังนี้

- RS232 จะใช้สำหรับต่อเข้ากับพอร์ต RS232 ของ PC เพื่อที่จะรับข้อมูลโปรแกรมจาก PC
- Data link จะใช้สำหรับกรณีใช้งานในโหมดสองแขน (Two Arms) คือใช้สำหรับส่งข้อมูลการทำงานของแขนข้างซ้าย ซึ่งข้อมูลนี้จะรับมาจาก PC โดยแขนข้างขวาอีกทอดหนึ่ง ซึ่งโปรแกรมควบคุมการส่งข้อมูลแบบสองแขนจะอยู่ใน CPU หลักของแขนข้างขวา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งจ่ายไฟฟ้ากระแสตรง(DC Power supply)

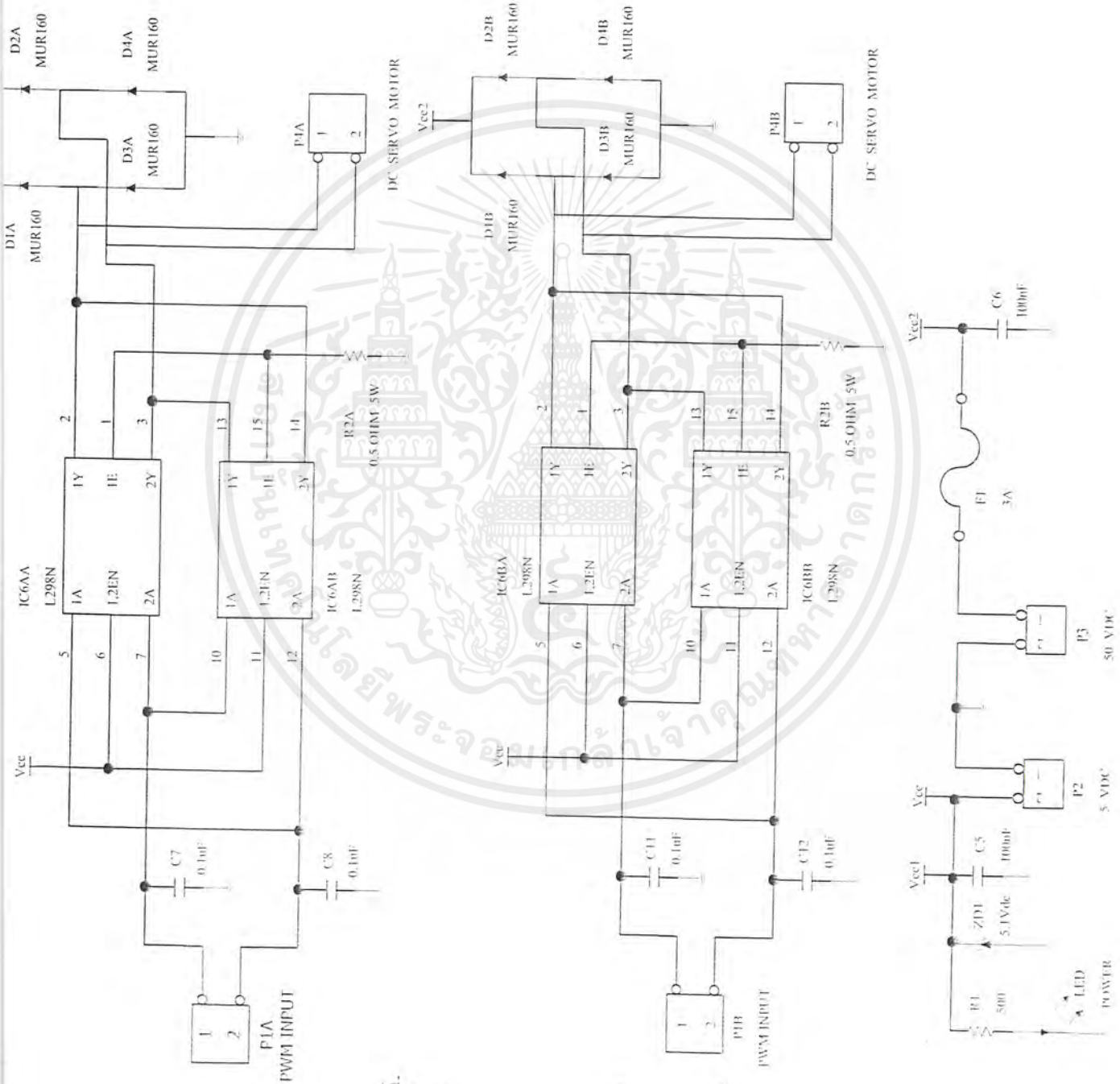
แหล่งจ่ายไฟฟ้ากระแสตรงที่จ่ายแรงดันไฟเลี้ยงให้กับวงจรทั้งหมด มีหลายระดับคือ +35 โวลท์ , +18 โวลท์ , +12 โวลท์ , -12 โวลท์ และ +5 โวลท์ ซึ่งวงจรของภาคจ่ายไฟแสดงดังรูปที่ 3.17



รูป3.17 แสดงวงจรภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรขับมอเตอร์(DC DRIVER MOTOR)

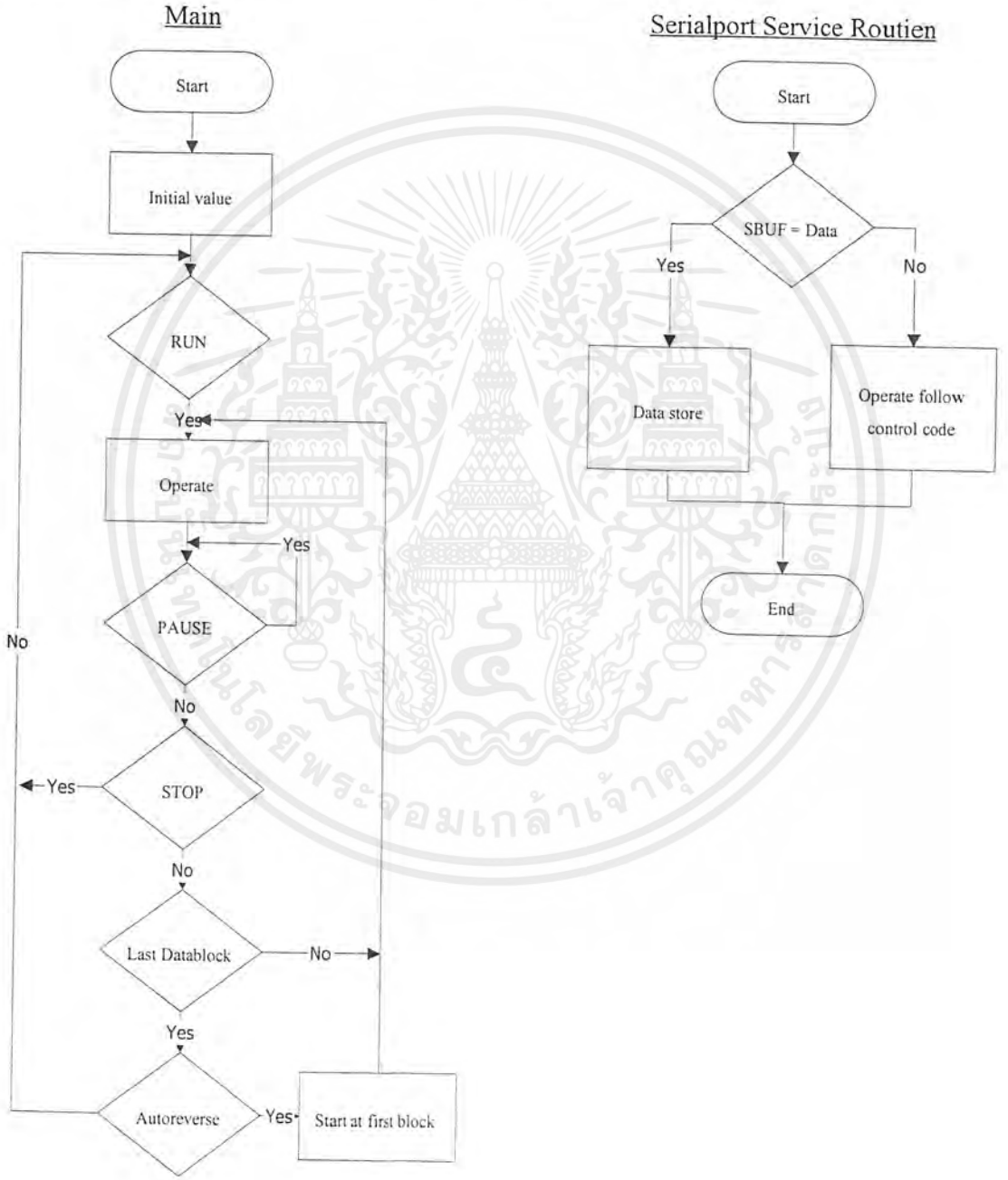


รูปที่ 3.18 แสดงวงจรภาคขับมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

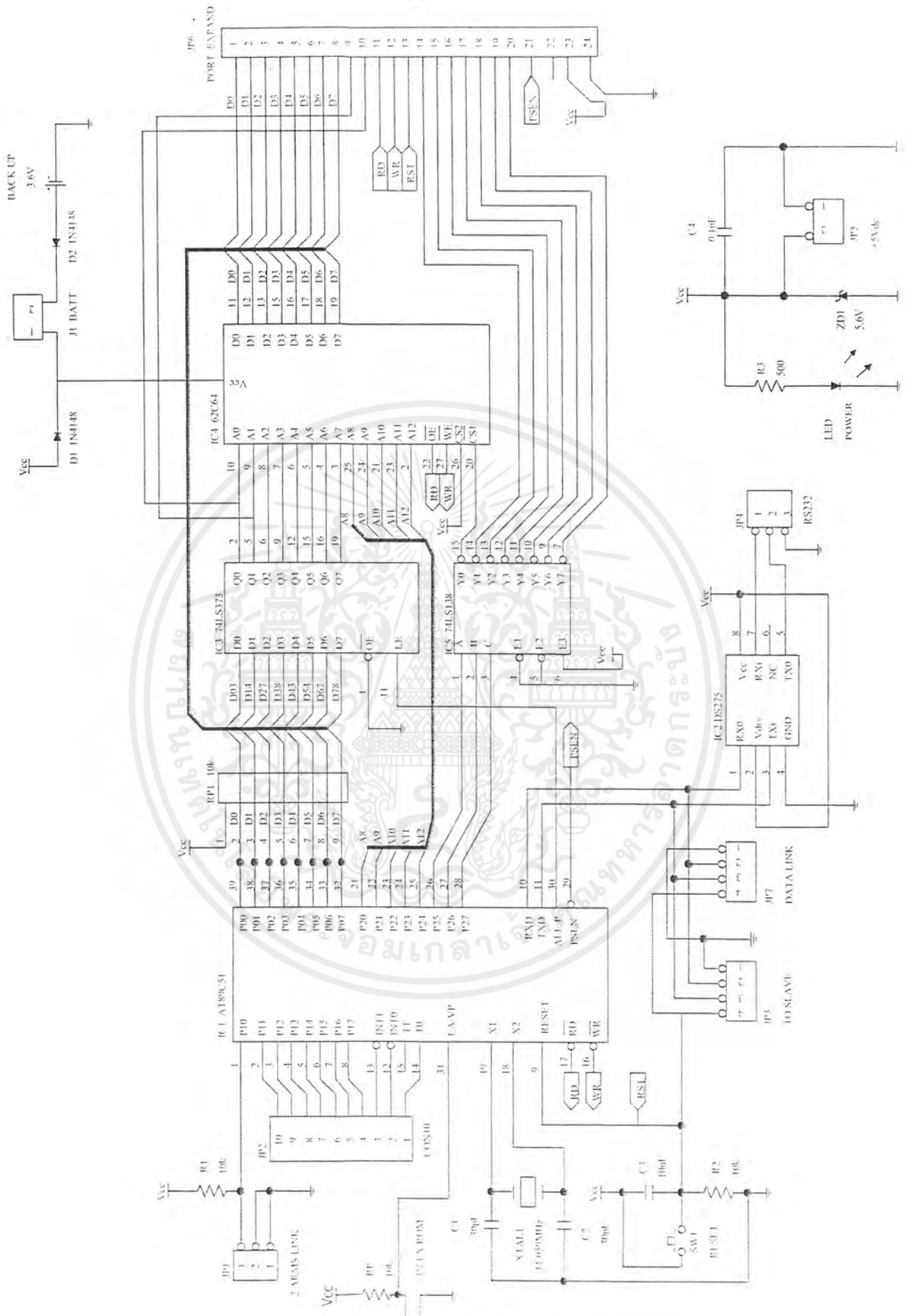
เมนบอร์ด(Main Controller)

เมนบอร์ดจะเป็นส่วนที่ใช้ควบคุมระบบทั้งหมดประกอบไปด้วย IC เบอร์ AT89C52 ซึ่งเป็นไมโครคอนโทรลเลอร์ เป็น CPU ประจำบอร์ด เมนบอร์ดจะทำหน้าที่เก็บข้อมูลของการเคลื่อนไหวของแขนกล และจะสั่งให้ชุด PID Motor Control ทำการควบคุมให้มอเตอร์หมุนไปที่ตำแหน่งใดๆ โฟลวชาร์ตและวงจรของเมนบอร์ดแสดงดังรูปที่ 3.19 และรูปที่ 3.20 ตามลำดับ



รูปที่ 3.19 แสดงโฟลวชาร์ตการทำงานของโปรแกรม CPU ตัวหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

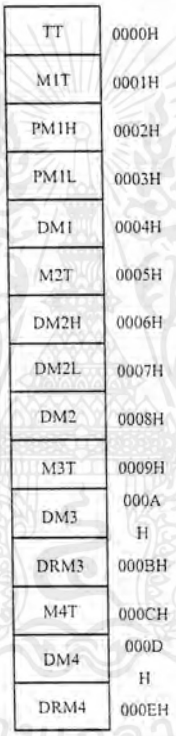


รูปที่ 3.2 แสดงวงจรแม่บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MASTER (Main Controller) จะทำหน้าที่จ่ายข้อมูลที่รับเข้ามาจาก PC แล้วทำการจ่ายข้อมูลไปให้แก่ CPU ตัวรองทุกตัวตามจังหวะเวลาที่ทำการโปรแกรมไว้ ซึ่งการส่งข้อมูลให้แก่ CPU ตัวรอง จะส่งผ่าน Serial Port และส่งในโหมด Multi Processor สำหรับ CPU ตัวหลักของ Channel A จะทำหน้าที่อีกอย่างหนึ่งคือทำหน้าที่ส่งข้อมูลให้กับ CPU ตัวหลัก ของ Channel B ในกรณีที่ทำงานในโหมด 2 แขน (two arms) ด้วย

การรับข้อมูลของ CPU ตัวหลักจาก PC จะรับเข้ามาในรูปบล็อกข้อมูลซึ่งในข้อมูลแต่ละบล็อกจะมีส่วนประกอบที่เหมือนกันดังนี้



รูปที่ 3.21 แสดงโครงสร้างของข้อมูลใน 1 บล็อค

CPU ตัวหลัก จะรับข้อมูลจาก PC เข้ามาเป็นบล็อกดังรูปที่ 3.21 ซึ่งจะถูกนำมาเก็บไว้ในหน่วยความจำ (RAM) ของระบบซึ่งหน่วยความจำดังกล่าวจะมี Battery Backup อยู่ตลอดเวลา (เราสามารถเลือกใช้ Battery Backup ได้โดยการ Set Jumper บน Board) ซึ่งข้อมูลแต่ละ Byte จะมีความหมายดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TT = เวลาที่จะใช้ทั้งหมดในการที่แขนกลจะทำงานตามข้อมูลในบล็อกนั้นๆ ถ้า TT = 1 หมายถึง 0.1 Sec

M1T = เวลาเริ่มต้นที่จะทำให้ M1 เริ่มทำงาน ซึ่งเวลานี้จะดูจาก TT โดยที่ TT จะเริ่มนับจาก 0 จนถึงค่าที่กำหนดให้กับ TT โดย TT จะเพิ่มค่าทุกๆ 0.1 Sec และ M1T จะเป็นตัวบอกว่าให้ M1 เริ่มทำงาน เมื่อ TT นับถึงค่าใด

PM1H = คือค่าของจำนวน Pulse ที่จะกำหนดให้มอเตอร์หมุนไปในตำแหน่งที่กำหนดซึ่งรายละเอียดได้กล่าวมาแล้วในหัวข้อของ PID Control ในบทที่ 3 PM1H จะเป็นค่า 8 Bit ด้านสูงของค่าทั้งหมดซึ่งจะนำไปรวมกับค่าของ PM1L เป็น 16 Bit

PM1L = เป็นค่าบอกจำนวน Pulse ที่จะให้ M1 หมุนไป 8 Bit ด้านต่ำ

DM1 = จะใช้กำหนดทิศทางการหมุนให้กับ M1 คือค่า DM1 = 1 หมุนทวนเข็มนาฬิกา
ค่า DM1 = 2 หมุนตามเข็มนาฬิกา

M2P = เวลาเริ่มต้นทำงานของ M2

PM2H = ค่า 8 Bit ด้านสูงของจำนวน Pulse ที่จะควบคุมให้ M2 หมุน

PM2L = ค่า 8 Bit ด้านต่ำของจำนวน Pulse ที่จะควบคุมให้ M2 หมุน

DM2 = ค่าที่ใช้กำหนดทิศทางการหมุนของ M2 ซึ่งหลักการเหมือนกับ DM1

M2T = ค่าที่ใช้กำหนดเวลาเริ่มต้นของ M3

TM3 = ค่าที่ใช้กำหนดตำแหน่งการหมุนของ M3

DM3 = ค่าที่ใช้กำหนดทิศทางการหมุนของ M3

M4T = ค่าที่ใช้กำหนดเวลาเริ่มต้นของ M4

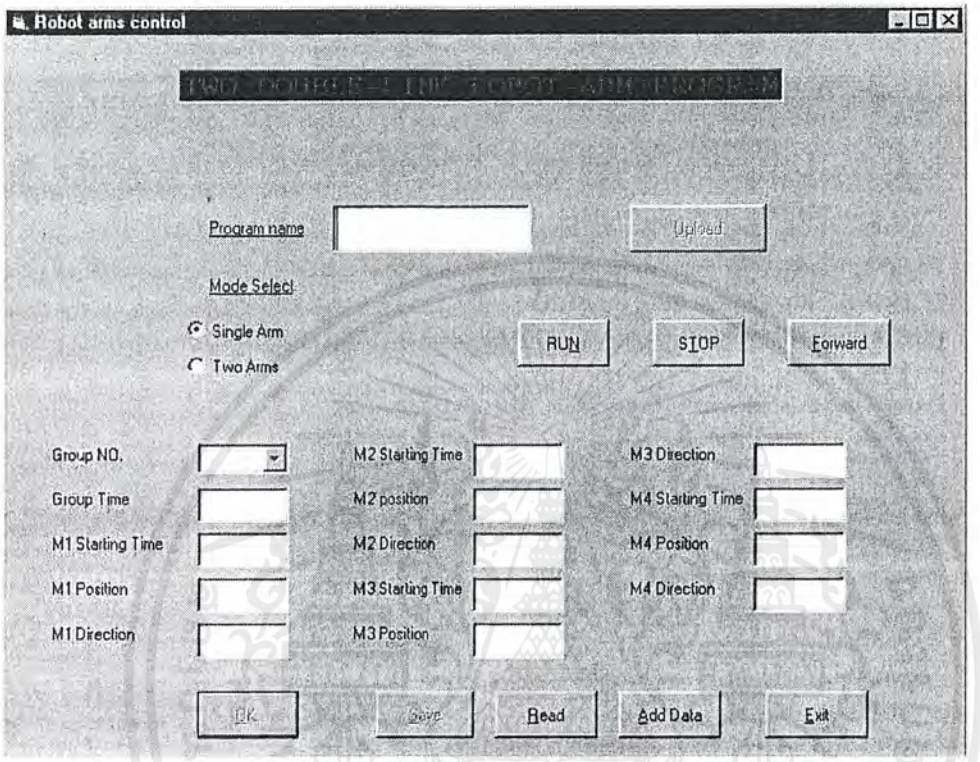
PM4 = ค่าที่ใช้กำหนดตำแหน่งการหมุนของ M4

DM4 = ค่าที่ใช้กำหนดทิศทางการหมุนของ M4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การป้อนข้อมูลการทำงานให้กับแขนกล

ในการป้อนข้อมูลเพื่อควบคุมการทำงานของแขนกลสามารถทำได้โดยการป้อนผ่านโปรแกรมที่เขียนขึ้นดังรูป



รูปที่ 3.22 แสดงโปรแกรมที่รับข้อมูลจากผู้ใช้

ในการป้อนข้อมูลสามารถทำได้โดยการป้อนข้อมูลเข้าไปตามช่องต่างๆดังนี้

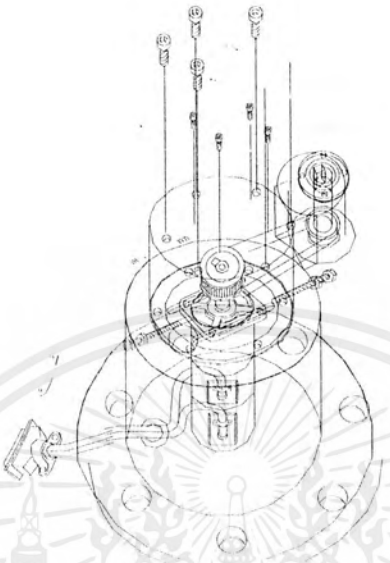
- Program name ใช้ป้อนชื่อของโปรแกรมซึ่งจะต้องป้อนชื่อก่อนเป็นอันดับแรกถ้าหากไม่ป้อนชื่อโปรแกรมก่อนแล้ว ช่องอื่นๆจะไม่รับข้อมูล
- Group No. ใช้สำหรับใส่หมายเลขของบล๊อคข้อมูล (บล๊อคข้อมูลที่ได้กล่าวถึงมาแล้วในหัวข้อที่ผ่านมา)
- M1 Starting Time ใช้กำหนดค่าเวลาเริ่มต้นทำงานให้กับมอเตอร์ตัวที่1
- M1 Position ใช้กำหนดตำแหน่งการหมุนให้กับมอเตอร์ตัวที่ 1 ซึ่งจะใส่ค่าเป็นจำนวนพัลซ์(รายละเอียดกล่าวไว้แล้วในเรื่อง PID Control)
- M1 Direction ใช้กำหนดทิศทางการหมุนของมอเตอร์โดยที่จะรับค่าตัวเลขได้สองค่าคือ 1 สำหรับการหมุนทวนเข็มนาฬิกา และ 2 สำหรับหมุนตามเข็มนาฬิกา

สำหรับช่องที่เหลือจะเป็นช่องรับข้อมูลสำหรับมอเตอร์อีก 3 ตัวที่เหลือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

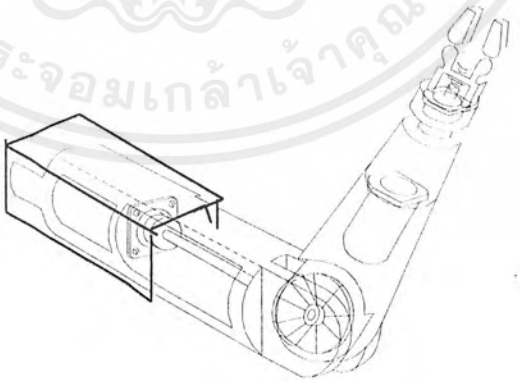
3.3 ส่วนประกอบทางโครงสร้าง

3.3.1 ฐาน



รูปที่ 3.23 แสดงส่วนประกอบฐาน

3.3.2 แขน

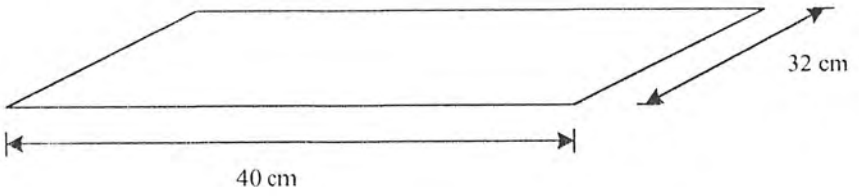


รูปที่ 3.24 แสดงส่วนประกอบแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

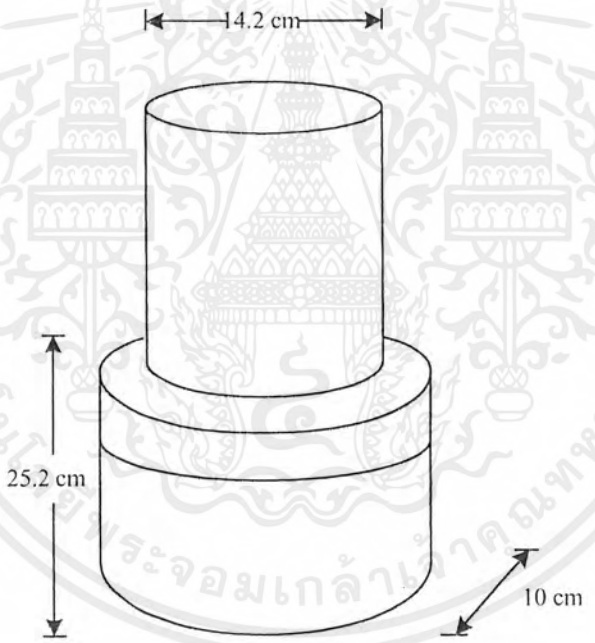
3.3.3 ขนาดของโครงสร้าง

ฐานรอง



รูปที่ 3.25 แสดงส่วนประกอบฐานรอง

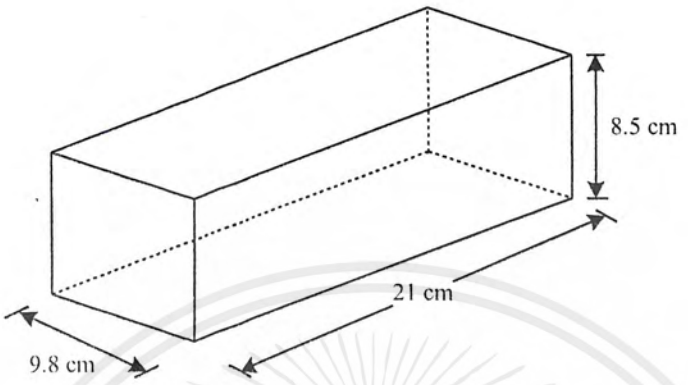
ฐาน



รูปที่ 3.26 แสดงขนาดของส่วนประกอบฐาน

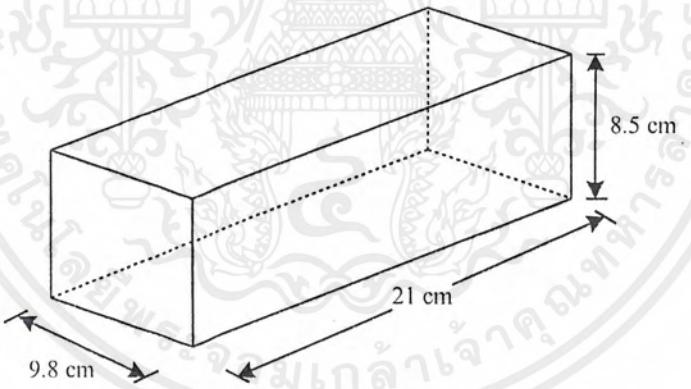
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนท่อนที่ 1



รูปที่ 3.27 แสดงขนาดของแขนท่อนที่1

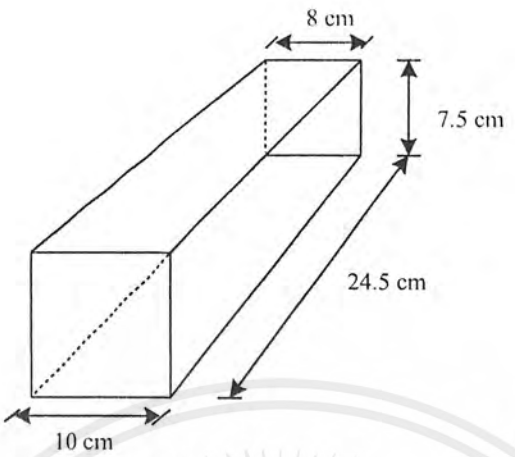
แขนท่อนที่ 2



รูปที่ 3.28 แสดงขนาดของแขนท่อนที่2

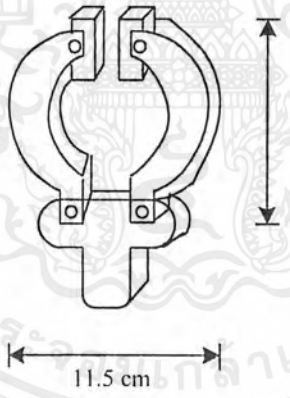
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แขนท่อนที่ 3



รูปที่ 3.29 แสดงขนาดของแขนท่อนที่3

แขนท่อนที่ 4



รูปที่ 3.30 แสดงขนาดของแขนท่อนที่4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

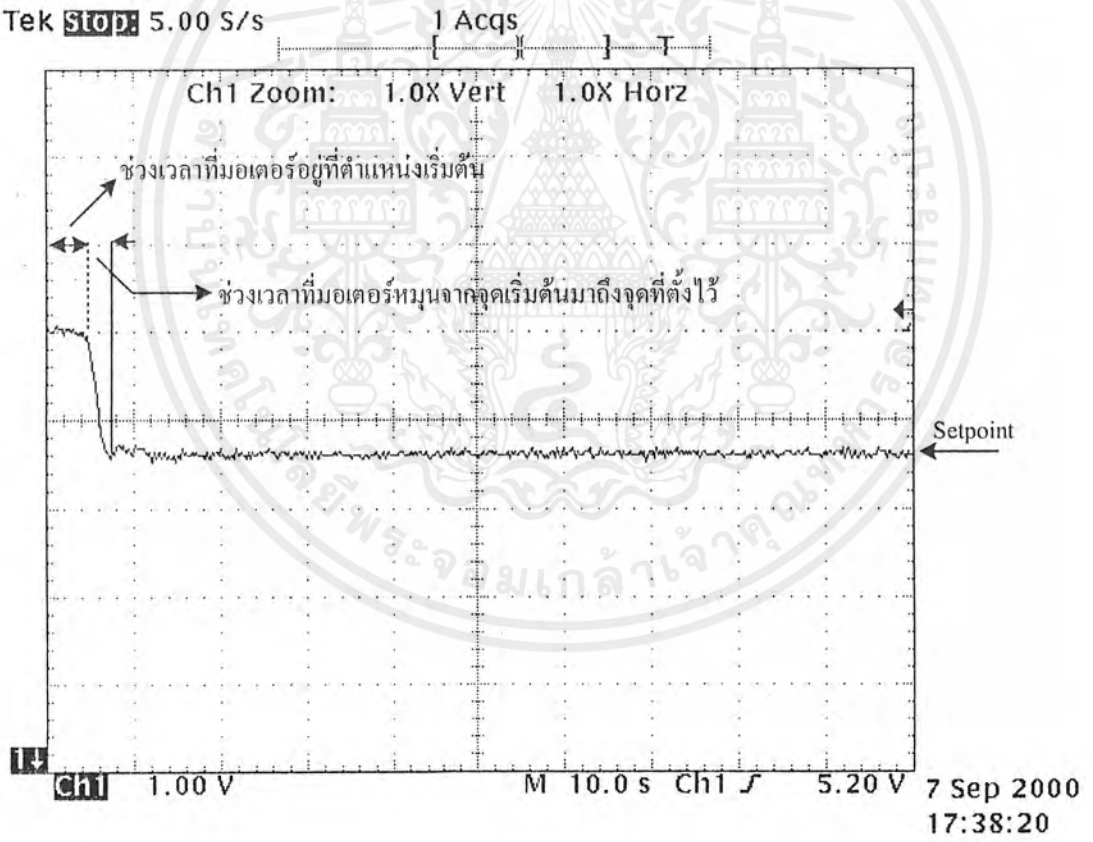
การทดลองและผลการทดลอง

4.1 ขั้นตอนการทดลอง

ขั้นตอนการทดลองต่างๆมีดังนี้

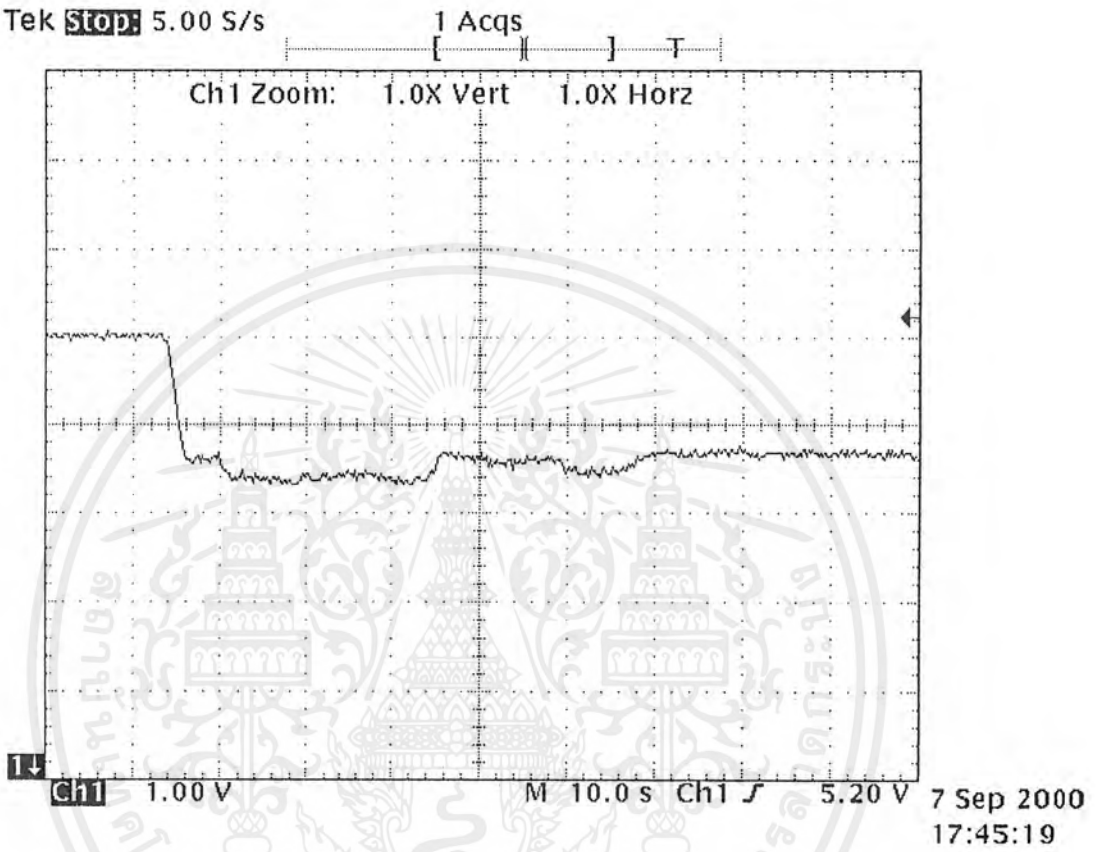
1. เสียบปลั๊กแหล่งจ่ายไฟ
2. เปิดสวิตซ์แหล่งจ่ายไฟ
3. กดสวิตซ์ RUN ในการทำงาน

4.2 ผลการทดลอง



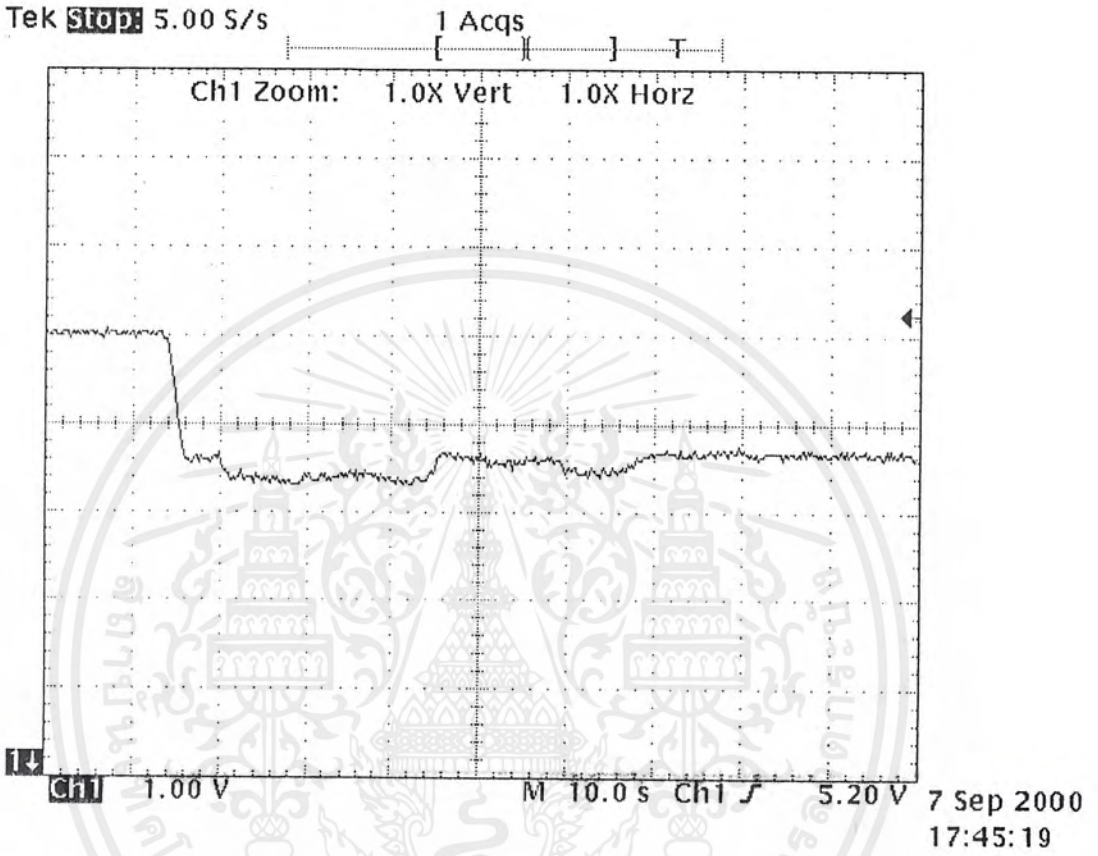
รูปที่ 4.1 แสดงผลตอบสนองของระบบควบคุมแบบ PID ขณะที่ไม่มีโหลด หมุน45องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดงผลตอบสนองของระบบควบคุมแบบ PID ขณะที่มีโหลด หมุน 45 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงผลตอบสนองของระบบควบคุมแบบ PID ขณะที่มีโหลด (ต่างกัน) หมุน 45 องศา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
ปัญหาและข้อเนอแนะ

5.1 สรุปโครงงาน

แขนกลสองแขนที่สร้างขึ้นมีลักษณะ 3 ข้อต่อ ในการหมุนของข้อต่อแต่ละข้อจะใช้มอเตอร์เป็นตัวหมุน ซึ่งแขนกลหนึ่งข้างจะใช้มอเตอร์ทั้งหมด 4 ตัว การทำงานของมอเตอร์นั้นจะถูกควบคุมโดยไมโครคอนโทรลเลอร์ เบอร์ AT 89C52 แขนกลหนึ่งข้อจะใช้ไมโครคอนโทรลเลอร์ทั้งหมด 5 ตัว และระบบที่ใช้ควบคุมจะเป็นแบบ PID คอนโทรลเลอร์ ซึ่งการออกแบบได้กล่าวมาแล้วในบทที่ 3 การทำงานของแขนกลจะมีอิสระต่อกัน ในการใช้งานแขนกลสามารถสั่งงานแบบอัตโนมัติหรือใช้คนควบคุมได้ ซึ่งเลือกได้จากการกดสวิตซ์

5.2 ปัญหาในการทำโครงงาน

5.2.1 ปัญหาในการสร้างแขนกล

- ผู้จัดทำขาดความรู้ทางด้านเครื่องกลจึงไม่สามารถแก้ปัญหาที่เกิดขึ้นได้โดยง่าย
- ขบวนการเลือกและจัดหาวัสดุที่นำมาสร้างหุ่นยนต์ใช้เวลานานเกินควร
- วัสดุอุปกรณ์บางอย่างที่ตรงกับความต้องการหายาก เช่น มอเตอร์

5.2.2 ปัญหาในการควบคุม

- ในการควบคุมดีซีเซอร์โวมอเตอร์ให้มีเสถียรภาพเป็นไปได้ค่อนข้างยาก เช่นความร้อนที่เกิดขึ้นกับมอเตอร์จะทำให้เกิดผลกับเสถียรภาพโดยตรง
- ในการควบคุมตำแหน่งของแขนกลอาจมีข้อผิดพลาดบ้างเพราะ Sensor อาจทำงานผิดพลาดซึ่งอาจเกิดขึ้นได้กับมอเตอร์ที่มีสภาพเก่ามากแล้ว

5.3 ข้อเนอแนะ

ข้อเสนอแนะนี้จัดทำขึ้นเพื่อเป็นแนวทางในการพัฒนา แขนกลให้มีความสามารถในการทำงานได้ดีขึ้น โดยจะนำเสนอเป็นข้อๆ

- ติดตั้ง sensor ที่มีคุณภาพที่ดีกว่าเดิม เพื่อให้ควบคุมตำแหน่งได้แม่นยำ
- เพิ่มชุดล้อเคลื่อนที่โดยอัตโนมัติ
- พัฒนาโปรแกรมให้ใช้งานได้หลายอย่าง เพื่อให้เกิดประสิทธิภาพสูงสุด

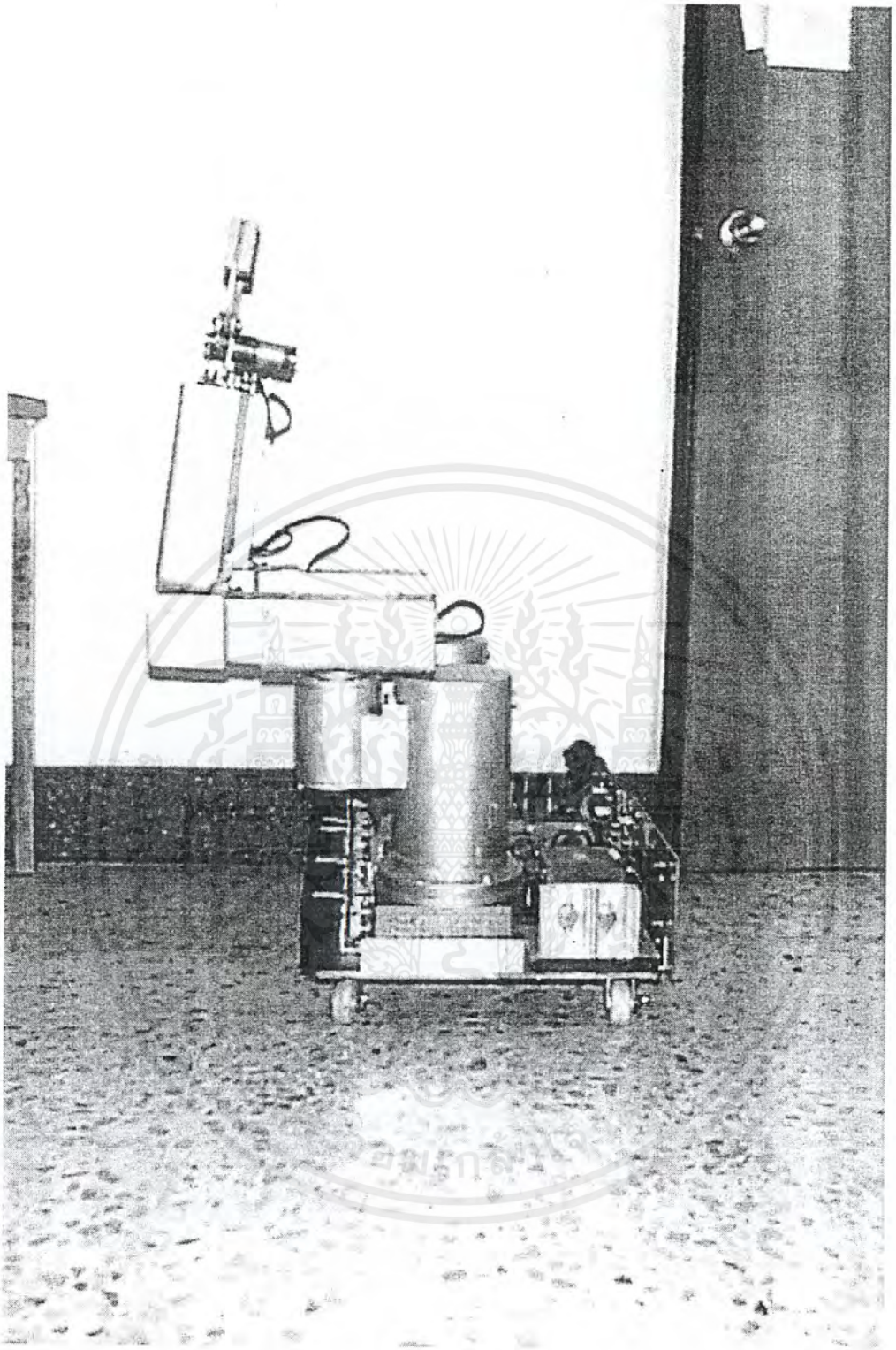
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาผนวก

ก

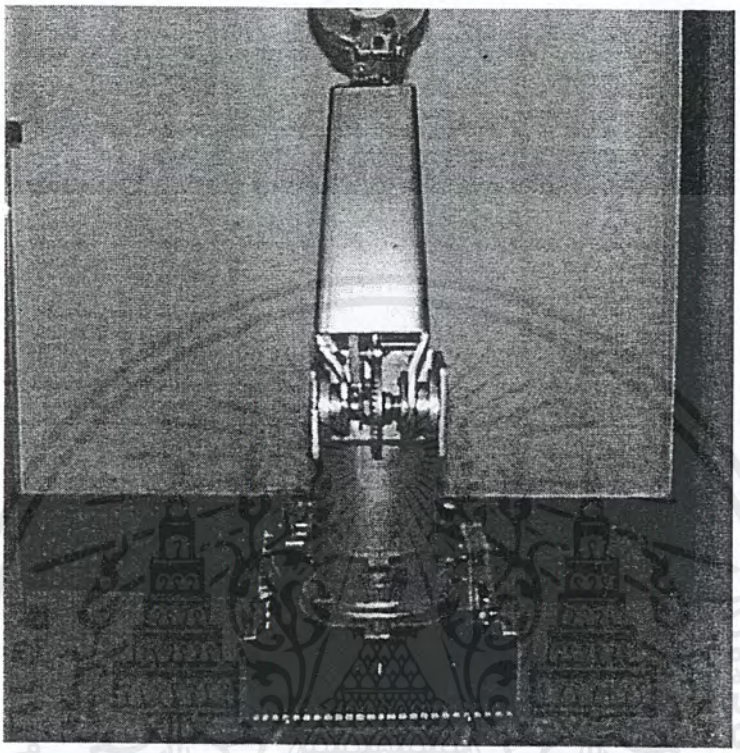
รูปถ่ายจริงของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



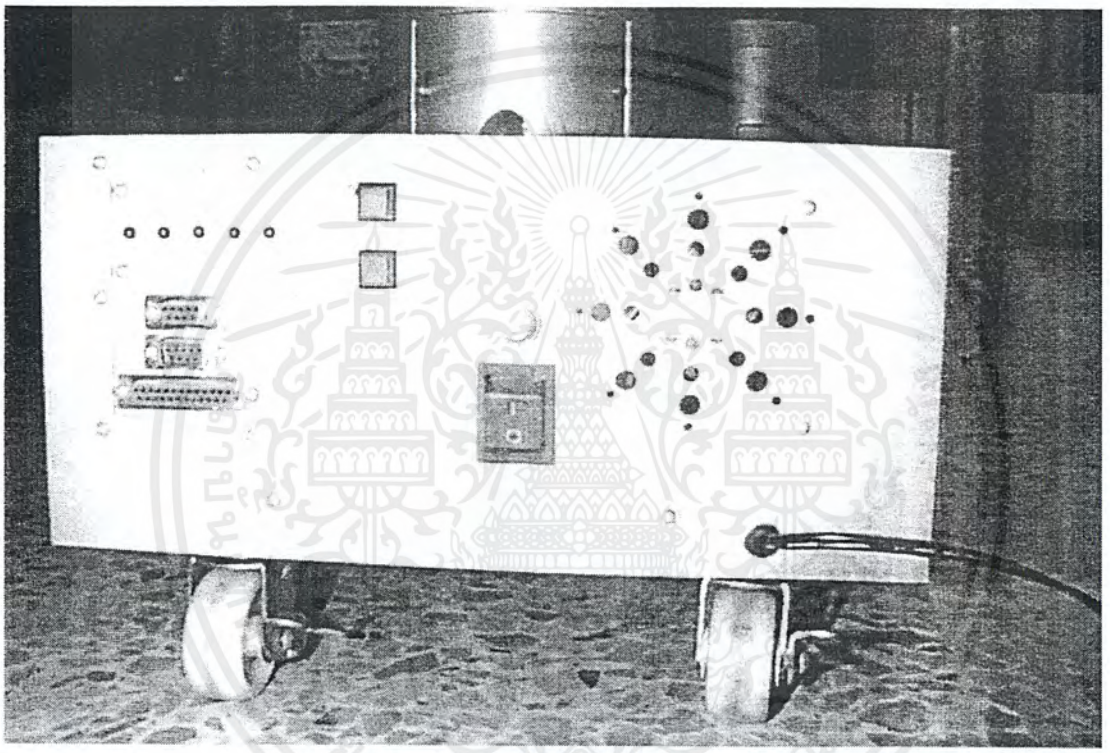
ภาพแสดงหุ่นยนต์สองแขน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพแสดงด้านหน้าของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพแสดงชุดควบคุมและแสดงผลด้านหลังของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ข

แสดง Source Code ของโปรแกรมที่ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม PID1.2

```
        POSL    EQU     40H     ;SETTING POINT
        POSH    EQU     41H     ;     "
        DST     EQU     42H     ;DIRECTION SET
        POSL0   EQU     43H     ;PASS POSL VALUE
        PP      EQU     44H     ;PROPERTIONAL VALUE
        I       EQU     45H     ;INTIGRAL VALUE
        I0      EQU     46H     ;PASS SUMMATION
        D       EQU     47H     ;DERIVERTIVE VALUE
        D0      EQU     48H     ;DERIVERTIVE CALCULATE
        KP      EQU     49H     ;PROPERTIONAL GAIN
        KI      EQU     4AH     ;INTIGRAL GAIN
        KD      EQU     4BH     ;DERIVERTIVE GAIN
        INAL    EQU     4CH     ;LOW BYTE I/P FOR ADDER
        INAH    EQU     4DH     ;HIGH BYTE I/P FOR ADDER
        ADL     EQU     4EH     ;LOW BYTE I/P ADDER
        ADH     EQU     4FH     ;HIGH BYTE I/P ADDER
        ADRL     EQU      50H      ;LOW BYTE OF ADDER
RESUALT
        ADRH     EQU      51H      ;HIGH BYTE OF ADDER
RESUALT
        PWL     EQU     52H     ;PULSE WIDTH BYTE LOW
        PWH     EQU     53H     ;PULSE WIDTH BYTE HIGH
        PSL     EQU     54H     ;BYTE LOW OF POSITION SET
        PSH      EQU      55H      ;BYTE HIGH OF POSITION
SET
        DSTT    EQU     56H     ;DIRECTION VALUE
        RST      EQU      57H      ;DATA CHANGE STATUS(RESET
PROGRAM)
        ADDRS    EQU      01H      ;ADRESS OF SLAVE
PROCESSOR
        ORG     0000H           ;START MONITOR PROGRAM
        LJMP    MAIN0
;---------------------------------------
        ORG     000BH
        LJMP    T0ISR
;---------------------------------------
        ORG     0013H
        INC     D0
        RETI
;---------------------------------------
        ORG     001BH
        LJMP    T1ISR
;---------------------------------------
        ORG     0023H
        LJMP    SISR
;---------------------------------------
        ORG     002BH
        LJMP    T2ISR
;---------------------------------------
;---------------------------------------
;       TIMMER2 ISR                     ;
;---------------------------------------
T2ISR:  CJNE    R0,#00H,END3
        PUSH    ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PUSH    PSW
        CPL     P1.0
        CLR     EX1             ;DISABLE INT1
        CLR      TF1               ;CANCEL INTERRUPT FROM
TIMMER1
        CLR     TR1             ;STOP TIMMER1
        MOV     TL1,PWL         ;SET PULSE WIDTH
        MOV     TH1,PWH         ;       "
        MOV     D,D0            ;STORE DERIVERTIVE VALUE
        LCALL   T0COM
        MOV     A,POSH
        JNZ         ADJ0                        ;ADJUST  P,I
VALUE(SATULATE)
        MOV     PP,POSL         ;STORE PROPERTIONAL VALUE
        MOV     A,POSL          ;CALCULATE INTIGRAL VALUE
END0:   CLR     C
        ADD     A,POSL0         ;         "
        RRC     A
        CLR     C
        ADD     A,I0
        JC      ADJ1            ;         "
        MOV     I,A             ;         "
END1:   MOV     I0,I            ;         "
        MOV     POSL0,POSL
        MOV     A,PWL
        ORL     A,PWH
        JZ      CLRPWM
        SETB    P0.7
END2:   SETB    TR1             ;ENABLE TIMMER1
        SETB    EX1             ;ENABLE INT1
        CLR     T2CON.7
        POP     PSW
        POP     ACC
END3:   RETI
ADJ0:   MOV     PP,#0FFH
        MOV     A,#0FFH
        SJMP    END0
ADJ1:   MOV     I,#0FFH
        SJMP    END1
CLRPWM: CLR     P0.7
        SJMP    END2
;---------------------------------------
;        INTERRUPT FOR TIMER0          ;
;---------------------------------------
T0ISR:  PUSH    ACC
        PUSH    PSW
        PUSH    DPL
        PUSH    DPH
        CLR     P0.7
        CLR     POSL
        CLR     PP
        CLR     I
        CLR     I0
        CLR     D
        CLR     D0
WAIT:   MOV     A,RST
        JNZ     CON1
        MOV     A,TL0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JZ      WAIT
CON1:   MOV     A,P0
        CPL     A
        ANL     A,#03H
        MOV     DST,A
        MOV     R0,#0FFH
        POP     DPH
        POP     DPL
        POP     PSW
        POP     ACC
        RETI
;---------------------------------------
;       INTERRUPT FOR TIMMER1           ;
;---------------------------------------
T1ISR:  CLR     P0.7
        CLR     TR1
        RETI
;---------------------------------------
;       INTERRUPT FOR SERIAL PORT       ;
;---------------------------------------
SISR:   PUSH    ACC
        PUSH    PSW
        JB      RI,RCV0
        CLR     TI
        SJMP    ENDSL
RCV0:   JNB     SM2,GETDAT
        MOV     A,SBUF
        CJNE    A,#ADDRS,ENDSL
        CLR     SM2
        MOV     R3,#03H         ;RECIEVE BYTE COUNTER
        MOV     R1,#PSL         ;INITIAL ADDRESS AT PSL
        SJMP    ENDSL
GETDAT: MOV     A,SBUF
        MOV     @R1,A
        MOV     P1,A
        INC     R1
        DJNZ    R3,ENDSL
        SETB    SM2
        MOV     RST,#0FFH
ENDSL:  CLR     RI
        POP     PSW
        POP     ACC
        RETI
;---------------------------------------
;       MAIN PROGRAM                    ;
;---------------------------------------
MAIN0:  MOV     R6,#0FH
DL0:    MOV     R7,#0FFH
        DJNZ    R7,$
        DJNZ    R6,DL0
;---------------------------------------
;       SYSTEM  CONFIGURATION           ;
;---------------------------------------
        SETB    PS              ;SET SERIAL PORT FOR HIGH
PRIORITY
        MOV     T2CON,#00H      ;TIMER2 = AUTORELOAD MODE
        SETB      EA                    ;ENABLE ACCESS FOR
INTERRUPT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,#15H          ;SET MODE C/T 0= COUNTER
16 BITS
        MOV     TMOD,A          ;SET MODE C/T 1= TIMMER 16
BITS
        SETB    IT1             ;SELECT EDGE TRIG FOR INT1
        SETB     ES                     ;ENABLE SERIAL PORT
INTERRUPT
        SETB    ET0             ;ENABLE C/T 0
        SETB    ET1             ;ENABLE C/T 1
        SETB    ET2             ;ENABLE C/T 2
        MOV     SCON,#0B0H      ;SET MODE 2 MULTIPROCESSOR
COMMUNICATION
        ORL     PCON,#80H       ;SET BOUAD RATE = 1/32 OSC
        MOV      RCAP2L,#03H      ;SET TIME BASE FOR PWM
FREQUENCY
        MOV     TL2,#03H        ;               "
        MOV     RCAP2H,#0FDH    ;               "
        MOV     TH2,#0FDH       ;               "

        MOV     PSL,#00H
        MOV     PSH,#00H
        MOV     DSTT,#00H
;-------------------------------------
;           SET PID GAIN              ;
;-------------------------------------
        MOV        KP,#03H                  ;SET GAIN OF
PROPERTIONAL
        MOV     KI,#01H          ;SET GAIN OF INTIGRAL
        MOV     KD,#02H          ;SET GAIN OF DERIVERTIVE

;------------------------------
;       INITAIL VALUE          ;
;------------------------------
REST:   MOV     R0,#00H
        MOV     R4,#00H
        MOV     R2,#00H
        MOV     D0,#00H
        MOV     I0,#00H
        MOV     PP,#00H
        MOV     I,#00H
        MOV     D,#00H
        MOV     POSL0,#00H
        MOV     RST,#00H
        MOV     POSL,#00H
        MOV     POSH,#00H
;-----------------------------------
;       SET POSITION               ;
;-----------------------------------
        MOV     POSL,PSL
        MOV     POSH,PSH

;-----------------------------------
;       SET DIRECTION              ;
;-----------------------------------
        MOV     DST,DSTT
;-----------------------------------
;-------------------------------------
;       2'S COMPLEMENT OF POSITION     ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;----------------------------------------
        MOV     TL0,POSL
        MOV     TH0,POSH
        LCALL   T0COM
        MOV     TL0,POSL
        MOV     TH0,POSH

        SETB    TR0             ;RUN C/T 0
        SETB    T2CON.2         ;RUN TIMMER 2
;----------------------------------------
;       RUN MOTOR                       ;
;----------------------------------------
        LCALL    OUTD                   ;RUN  FOLLOW DIRECTION
SET(DST)
;----------------------------------------
MAIN:   LCALL   PIDCAL
        MOV     A,RST
        JNZ     REST
        CJNE     R0,#0FFH,MAIN    ;R0 = T0 OVERFLOW STATUS
,00=OVERFLOW
        MOV     R0,#00H
        LCALL   T0COMCT          ;2'S COMPLEMENT OF TIMMER
0
        LCALL   OUTD
;MAIN1:  LCALL    CHECKD
;        MOV     A,RST
;        JNZ     REST
;           CJNE      R4,#00H,MAIN1     ;DIRECTION STATUS,
DST=DR:R1=00
        LJMP    MAIN
;----------------------------------------
;       SUBROUTIEN                      ;
;----------------------------------------
PIDCAL: PUSH    ACC
        MOV     A,PP
        MOV     B,KP
        MUL     AB
        MOV     INAL,A
        MOV     INAH,B
        MOV     A,I
        MOV     B,KI
        MUL     AB
        MOV     ADL,A
        MOV     ADH,B
        LCALL   ADDER
        CJNE    R2,#00H,ADJ3
        MOV     INAL,ADRL
        MOV     INAH,ADRH
        MOV     A,D
        MOV     B,KD
        MUL     AB
        MOV     ADL,A
        MOV     ADH,B
        LCALL   ADDER
        CJNE    R2,#00H,ADJ3     ;CHECK CARRY OF ADDER
        MOV     A,ADRL
        CPL     A
        CLR     C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD     A,#01H
        MOV     PWL,A
        MOV     A,ADRH
        CPL     A
        ADDC    A,#00H
        MOV     PWH,A
CON:    POP     ACC
        RET
ADJ3:   MOV     PWL,#00H
        MOV     PWH,#00H
        SJMP    CON
;-------------------------------------
;       2'S COMPLEMENT OF TIMMER0      ;
;-------------------------------------
T0COM:  PUSH    ACC
        PUSH    PSW
        MOV     A,TL0
        CPL     A
        CLR     C
        ADD     A,#01H
        MOV     POSL,A
        MOV     A,TH0
        CPL     A
        ADDC    A,#00H
        MOV     POSH,A
        POP     PSW
        POP     ACC
        RET
T0COMCT:PUSH    ACC
        MOV     A,TL0
        CPL     A
        CLR     C
        ADD     A,#01H
        MOV     TL0,A
        MOV     A,TH0
        CPL     A
        ADDC    A,#00H
        MOV     TH0,A
        POP     ACC
        RET
OUTD:   PUSH    ACC
        MOV     A,DST
        MOV     P2,A
        POP     ACC
        RET

CHECKD: PUSH    ACC
        MOV     A,P0
        ANL     A,#03H
        SUBB    A,DST
        MOV     R4,A
        POP     ACC
        RET
ADDER:  PUSH    ACC
        PUSH    PSW
        MOV     A,INAL
        CLR     C
        ADD     A,ADL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     ADRL,A
        MOV     A,INAH
        ADDC    A,ADH
        MOV     ADRH,A
        MOV     A,#00H
        ADDC    A,#00H
        MOV     R2,A
        POP     PSW
        POP     ACC
        RET
        END
```

โปรแกรมหลักสำหรับ CPU หลัก

```
        CLK     EQU     40H     ;CLOCK NUMBER (1 = 100mS)
        ADDR    EQU     41H     ;ADDRESS OF SLAVE PROCESSOR
        STS1    EQU     42H
        STS2    EQU     43H
        STS3    EQU     44H
        STS4    EQU     45H
        KEY     EQU     46H
        RUN     EQU     47H
        STOP    EQU     48H
        A_F     EQU     49H
        MODE    EQU     4AH
        DATA    EQU     4BH
        FLAG    EQU     4CH
        DAT     EQU     4DH
        M       EQU     4EH
        BSIZE   EQU     4FH
        ADRL    EQU     50H
        ADRH    EQU     51H
        BSIZE0  EQU     52H
;---------------------------------------
        ORG     0000H
        LJMP    MAIN0
;---------------------------------------
        ORG     0003H
        LJMP    EX0ISR
;---------------------------------------
;        ORG     0023H
;        LJMP    SISR
;---------------------------------------
        ORG     002BH
        LJMP    T2ISR
;---------------------------------------
EX0ISR: PUSH    ACC
        PUSH    PSW
        CLR     EX0
        LCALL   DELAY_S
        MOV     A,P1
        ANL     A,#06H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     KEY,A
SLOOP:  MOV     A,P1
        ANL     A,#06H
        JNZ     SCON2
        MOV     KEY,#00H
SCON2:  CJNE    A,#06H,SLOOP
        MOV     A,KEY
        CJNE    A,#00H,SCON0
        MOV     STOP,#0FFH
        MOV     RUN,#00H
        MOV     P1,#1FH
        SJMP    S_END
SCON0:  CJNE    A,#04H,SCON1
        MOV     STOP,#00H
        MOV     A,RUN
        CPL     A
        MOV     RUN,A
        CLR     P1.4
        CPL     P1.5
        MOV     C,P1.5
        CPL     C
        MOV     P1.6,C
        SJMP    S_END
SCON1:  CJNE    A,#02H,S_END
        MOV     A,A_F
        CPL     A
        MOV     A_F,A
        CPL     P1.7
S_END:  SETB    EX0
        POP     PSW
        POP     ACC
        RETI

T2ISR:  PUSH    ACC
        PUSH    PSW
        MOV     A,R1
        CPL     A
        MOV     R1,A
        JNZ     ENDT2
        MOV     A,CLK
        INC     A
        MOV     CLK,A
ENDT2:  CLR     T2CON.7                 ;CLEAR TF2
        POP     PSW
        POP     ACC
        RETI

;---------------------------------------
;       MAIN PROGRAM                    ;
;---------------------------------------
MAIN0:  MOV     R6,#0FFH                ;DELAY
 DL1:   MOV     R7,#0FFH
        DJNZ    R7,$
        DJNZ    R6,DL1

;---------------------------------------
        SETB    ET2
        SETB    IT0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SETB    EX0
            MOV     SP,#0FH
            MOV     TMOD,#22H
            MOV     T2CON,#00H          ;SET T2 = AUTORELOAD MODE
            MOV     SCON,#0E0H          ;SET SERIAL PORT = MOD3
            MOV      R1,#00H                ;SET F0 FLAG FOR TIME
 BASE GENERATION
            MOV     CLK,#00H            ;INITIAL CLOCK COUNT ,CLK
 =1 ,TIME =100mS
            MOV     TL2,#01H            ;SET TIME FOR T2 = 50 mS
            MOV     RCAP2L,#01H         ;       "
            MOV     TH2,#4CH            ;       "
            MOV     RCAP2H,#4CH         ;       "
            MOV     TL1,#0FDH
            MOV     TH1,#0FDH
            MOV     P1,#0FH
            SETB    EA
            SETB    REN
            SETB    TR1
            SETB    T2CON.2             ;START TIMER 2
            MOV     P3,#0FFH
            MOV     P1,#0FH
            MOV     FLAG,#00H
            MOV     M,#00H
            MOV     DAT,#00H
LP0:        MOV     DPTR,#0000H
            MOV     STS1,#00H
            MOV     STS2,#00H
            MOV     STS3,#00H
            MOV     STS4,#00H
            MOV     RUN,#00H
            MOV     STOP,#00H
            MOV     A_F,#00H
            SETB    T2CON.2
;-----------------------------------
;       MAIN    LOOP                ;
;-----------------------------------
            MOV     R7,#02H
LP:         MOV     CLK,#00H
            LCALL   DELAY
            SETB    TB8
            MOV     A,#0FFH
            MOV     SBUF,A
            JNB     TI,$
            CLR     TI
            CPL     P1.4
            DJNZ    R7,LP
            LCALL   DELAY
            MOV     CLK,#00H
            MOV     P1,#1FH
            MOV     CLK,#00H
            MOV     RUN,#00H
            MOV     P1,#1FH
MAIN:       JB      RI,SISR
MAIN1:      MOV     A,RUN
            JNZ     MCON0
            CLR     T2CON.2
            MOV     A,STOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNZ     LP0
        SJMP    MAIN
MCON0:  SETB    T2CON.2
        MOV     A,STOP
        JNZ     LP0
        LCALL   COMP
        MOVX    A,@DPTR
        JZ      END1
        SUBB    A,CLK
        JNZ     MAIN
        LCALL   INCD15
        MOV     CLK,#00H
        MOV     STS1,#00H
        MOV     STS2,#00H
        MOV     STS3,#00H
        MOV     STS4,#00H
        SJMP    MAIN
END1:   MOV     DPTR,#0000H
        MOV     CLK,#00H
        MOV     STS1,#00H
        MOV     STS2,#00H
        MOV     STS3,#00H
        MOV     STS4,#00H
        MOV     A,A_F
        JNZ     END2
        LJMP    LP0
END2:   LJMP    MAIN
;---------------------------------------
;       SUB ROUTEEN                     ;
;---------------------------------------
SISR:   PUSH    ACC
        PUSH    PSW
        PUSH    DPH
        PUSH    DPL
        SETB    RS0
        JB      RI,RECIEV
        CLR     TI
        LJMP    S_END0
RECIEV: MOV     A,FLAG
        JNZ     MSTS1
        MOV     A,SBUF
        JNZ     MSTS0
        MOV     FLAG,#01H
        MOV     DAT,#00H
        MOV     M,#0FFH
        LJMP    S_END0
MSTS0:  MOV     FLAG,#02H
        LJMP    S_END0
MSTS1:  MOV     A,M
        JZ      MSTS2
        MOV     A,SBUF
        MOV     MODE,A
        MOV     M,#00H
        LJMP    S_END0
MSTS2:  MOV     A,FLAG
        CJNE    A,#01H,MSTS3
        MOV     A,DAT
        JNZ     DAT0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,SBUF
        MOV     BSIZE,A
        MOV     BSIZE0,#0FH
        MOV     ADRL,#00H
        MOV     ADRH,#00H
        MOV     DAT,#0FFH
        LJMP    S_END0
DAT0:   MOV     R0,BSIZE
        MOV     R1,BSIZE0
        MOV     DPH,ADRH
        MOV     DPL,ADRL
        MOV     A,SBUF
        MOVX    @DPTR,A
        INC     DPTR
        MOV     ADRL,DPL
        MOV     ADRH,DPH
        DEC     R1
        MOV     BSIZE0,R1
        CJNE    R1,#00,DAT1
        MOV     R1,#0FH
        MOV     BSIZE0,R1
        DEC     R0
        MOV     BSIZE,R0
        CJNE    R0,#00H,DAT1
        MOV     FLAG,#00H
        MOV     DAT,#00H
DAT1:   SJMP    S_END0
MSTS3:  CJNE    A,#02H,S_END0
        MOV     A,SBUF
        CJNE    A,#04H,MSTS4
        MOV     A_F,#00H
        CLR     P1.7
        MOV     FLAG,#00H
        SJMP    S_END0
MSTS4:  CJNE    A,#05H,MSTS5
        MOV     A_F,#0FFH
        SETB    P1.7
        MOV     FLAG,#00H
        SJMP    S_END0
MSTS5:  CJNE    A,#01H,MSTS6
        MOV     RUN,#00H
        MOV     STOP,#00H
        SETB    P1.6
        CLR     P1.5
        CLR     P1.4
        MOV     FLAG,#00H
        SJMP    S_END0
MSTS6:  CJNE    A,#02H,MSTS7
        MOV     RUN,#0FFH
        MOV     STOP,#00H
        SETB    P1.5
        CLR     P1.4
        CLR     P1.6
        MOV     FLAG,#00H
        SJMP    S_END0
MSTS7:  CJNE    A,#03H,S_END0
        MOV     STOP,#0FFH
        MOV     RUN,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     P1,#1FH
        MOV     FLAG,#00H
S_END0: CLR     RI
        POP     DPL
        POP     DPH
        POP     PSW
        POP     ACC
        LJMP    MAIN1

COMP:   PUSH    ACC
        PUSH    PSW
        PUSH    DPL
        PUSH    DPH
        INC     DPTR
        MOV     ADDR,#01H
        MOVX    A,@DPTR
        SUBB    A,CLK
        JNZ     NEXT0
        MOV     A,STS1
        JNZ     NEXT0
        MOV     STS1,#0FFH
        LCALL   SEND
NEXT0:  LCALL   INCD4
        MOV     ADDR,#02H
        MOVX    A,@DPTR
        SUBB    A,CLK
        JNZ     NEXT1
        MOV     A,STS2
        JNZ     NEXT1
        MOV     STS2,#0FFH
        LCALL   SEND
NEXT1:  LCALL   INCD4
        MOV     ADDR,#03H
        MOVX    A,@DPTR
        SUBB    A,CLK
        JNZ     NEXT2
        MOV     A,STS3
        JNZ     NEXT2
        MOV     STS3,#0FFH
        LCALL   SEND
NEXT2:  LCALL   INCD3
        MOV     ADDR,#04H
        MOVX    A,@DPTR
        SUBB    A,CLK
        JNZ     NEXT3
        MOV     A,STS4
        JNZ     NEXT3
        MOV     STS4,#0FFH
        LCALL   SEND
NEXT3:  POP     DPH
        POP     DPL
        POP     PSW
        POP     ACC
        RET
SEND:   PUSH    ACC
        PUSH    PSW
        PUSH    DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PUSH    DPH
        MOV     P3,#0FFH
        MOV     R0,#03H
        MOV     A,ADDR
        SETB    TB8
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        MOV     A,ADDR
        SUBB    A,#03H
        JC      CON0
        DEC     R0
CON0:   INC     DPTR
        MOVX    A,@DPTR
        CLR     TB8
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI
        DJNZ    R0,CON0
        POP     DPH
        POP     DPL
        POP     PSW
        POP     ACC
        RET
DELAY:  MOV     A,#0FH
        CJNE    A,CLK,DELAY
        RET
DELAY_S:MOV     R6,#6FH
DL0:    MOV     R5,#0FFH
        DJNZ    R5,$
        DJNZ    R6,DL0
        RET
INCD15: INC     DPTR
INCD14: INC     DPTR
INCD13: INC     DPTR
INCD12: INC     DPTR
INCD11: INC     DPTR
INCD10: INC     DPTR
INCD9:  INC     DPTR
INCD8:  INC     DPTR
INCD7:  INC     DPTR
INCD6:  INC     DPTR
INCD5:  INC     DPTR
INCD4:  INC     DPTR
INCD3:  INC     DPTR
INCD2:  INC     DPTR
        INC     DPTR
        RET
        END
```

โปรแกรม PID3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POS     EQU     40H     ;SETTING POINT
        DST     EQU     42H     ;DIRECTION SET
        POS0    EQU     43H     ;PASS POSL VALUE
        PP      EQU     44H     ;PROPERTIONAL VALUE
        I       EQU     45H     ;INTIGRAL VALUE
        I0      EQU     46H     ;PASS SUMMATION
        KP      EQU     49H     ;PROPERTIONAL GAIN
        KI      EQU     4AH     ;INTIGRAL GAIN
        KD      EQU     4BH     ;DERIVERTIVE GAIN
        INAL    EQU     4CH     ;LOW BYTE I/P FOR ADDER
        INAH    EQU     4DH     ;HIGH BYTE I/P FOR ADDER
        ADL     EQU     4EH     ;LOW BYTE I/P ADDER
        ADH     EQU     4FH     ;HIGH BYTE I/P ADDER
        ADRL   EQU     50H     ;LOW BYTE OF ADDER RESUALT
        ADRH   EQU     51H     ;HIGH BYTE OF ADDER RESUALT
        PWL     EQU     52H     ;PULSE WIDTH BYTE LOW
        PWH     EQU     53H     ;PULSE WIDTH BYTE HIGH
        RST     EQU     57H     ;DATA CHANGE STATUS(RESET
PROGRAM)
        INIF    EQU     59H     ;NITIAL STATE STATUS
        ADDRS     EQU         04H         ;ADRESS OF SLAVE
PROCESSOR
        TF2     BIT     T2CON.7 ;TIMER2 FLAG
        TR2     BIT     T2CON.2 ;BIT START OF TIMER2

        ORG     0000H           ;START MONITOR PROGRAM
        LJMP    MAIN0
;---------------------------------------
        ORG     000BH
        LJMP    T0ISR
;---------------------------------------
        ORG     001BH
        LJMP    T1ISR
;---------------------------------------
        ORG     0023H
        LJMP    SISR
;---------------------------------------
;       EXTERNAL INT0 SERVICE ROUTIEN  ;
;---------------------------------------
T0ISR:  PUSH    ACC
        PUSH    PSW
        CLR     TF1                 ;CANCEL INTERRUPT FROM
TIMMER1
        CLR     TR1                 ;STOP TIMMER1
        MOV     A,P0
        CLR     C
        SUBB    A,POS
        JC      COMP0
        JZ      EEND0
        MOV     DST,#02H
        MOV     PP,A
ECON0:  CLR     C
        ADD     A,POS0
        JNC     ECON1
        MOV     A,#0FFH
ECON1:  CLR     C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD     A,I0
        JNC     ECON2
        MOV     A,#0FFH
ECON2:  MOV     I,A
        MOV     I0,A
        MOV     POS0,PP
        MOV     A,PWL
        JNZ     ECON3
        MOV     A,PWH
        JZ      EEND1
ECON3:  MOV     TL1,PWL
        MOV     TH1,PWH
        SETB    P1.0
        SETB    TR1
        MOV     TL0,#80H
        MOV     TH0,#0FBH
        POP     PSW
        POP     ACC
        RETI
EEND0:  MOV     DST,#00H
        MOV     PP,#00H
        MOV     I,#00H
        MOV     I0,#00H
EEND1:  CLR     P1.0
        MOV     TL0,#80H
        MOV     TH0,#0FBH
        POP     PSW
        POP     ACC
        RETI
COMP0:  MOV     DST,#01H
        CPL     A
        ADD     A,#01H
        MOV     PP,A
        LJMP    ECON0
;---------------------------------------
;       INTERRUPT FOR TIMMER1           ;
;---------------------------------------
T1ISR:  CLR     P1.0
        CLR     TR1
        RETI
;---------------------------------------
;       INTERRUPT FOR SERIAL PORT       ;
;---------------------------------------
SISR:   PUSH    ACC
        PUSH    PSW
        JB      RI,RCV0
        CLR     TI
        SJMP    ENDSL
RCV0:   JNB     SM2,GETDAT
        MOV     A,SBUF
        CJNE    A,#0FFH,RCON
        MOV     INIF,#0FFH
        SJMP    ENDSL
RCON:   CJNE    A,#ADDRS,ENDSL
        CLR     SM2
        MOV     R3,#01H         ;RECIEVE BYTE COUNTER
        MOV     R1,#POS         ;INITIAL ADDRESS AT POS
        SJMP    ENDSL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
GETDAT: MOV     A,SBUF
        MOV     @R1,A
        INC     R1
        DJNZ    R3,ENDSL
        SETB    SM2
ENDSL:  CLR     RI
        POP     PSW
        POP     ACC
        RETI
;---------------------------------------
;       MAIN PROGRAM                    ;
;---------------------------------------
MAIN0:  MOV     R6,#0FH
DL0:    MOV     R7,#0FFH
        DJNZ    R7,$
        DJNZ    R6,DL0
;---------------------------------------
;       SYSTEM  CONFIGURATION           ;
;---------------------------------------
;        SETB    PS              ;SET SERIAL PORT FOR HIGH
PRIORITY
        MOV     A,#11H          ;SET MODE C/T 0= TIMMER 16
BITS
        MOV     TMOD,A          ;SET MODE C/T 1= TIMMER 16
BITS
        MOV     SCON,#0F0H      ;SET MODE 3 MULTIPROCESSOR
COMMUNICATION
        MOV       T2CON,#30H        ;T2 = BAUD RATE GEN
=9.6kb/Sec T&R
        MOV     TL2,#0DCH       ;               "
        MOV     RCAP2L,#0DCH    ;               "
        MOV     TH2,#0FFH       ;               "
        MOV     RCAP2H,#0FFH    ;               "
        MOV     TL0,#80H
        MOV     TH0,#0FBH
 ;        SETB     ES                ;ENABLE SERIAL PORT
INTERRUPT
        SETB    ET0
        SETB    ET1             ;ENABLE C/T 1
        SETB     EA                     ;ENABLE ACCESS FOR
INTERRUPT
        SETB    TR2
        SETB    TR0
;---------------------------------------
;            SET PID GAIN             ;
;---------------------------------------
        MOV       KP,#05H                   ;SET GAIN OF
PROPERTIONAL
        MOV     KI,#02H           ;SET GAIN OF INTIGRAL
;---------------------------------------
REST:   MOV     POS,#30H
        MOV     INIF,#00H
;-----------------------------
;       INITAIL VALUE         ;
;-----------------------------
        MOV     R0,#00H
        MOV     R4,#00H
        MOV     R2,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     I0,#00H
        MOV     PP,#00H
        MOV     I,#00H
        MOV     POS0,#00H
        MOV     POS,#00H
MAIN:   LCALL   PIDCAL
        MOV     A,INIF
        JNZ     REST
        LCALL   OUTD
        SJMP    MAIN
;---------------------------------------
;       SUBROUTIEN                      ;
;---------------------------------------
PIDCAL: PUSH    ACC
        PUSH    PSW
        MOV     A,PP
        MOV     B,KP
        MUL     AB
        MOV     INAL,A
        MOV     INAH,B
        MOV     A,I
        MOV     B,KI
        MUL     AB
        MOV     ADL,A
        MOV     ADH,B
        LCALL   ADDER
        CJNE    R2,#00H,ADJ3    ;CHECK CARRY OF ADDER
        MOV     A,ADRL
        CPL     A
        CLR     C
        ADD     A,#01H
        MOV     PWL,A
        MOV     A,ADRH
        CPL     A
        ADDC    A,#00H
        MOV     PWH,A
CON:    POP     PSW
        POP     ACC
        RET
ADJ3:   MOV     PWL,#01H
        MOV     PWH,#00H
        SJMP    CON
OUTD:   PUSH    ACC
        MOV     A,DST
        MOV     P2,A
        POP     ACC
        RET
ADDER:  PUSH    ACC
        PUSH    PSW
        MOV     A,INAL
        CLR     C
        ADD     A,ADL
        MOV     ADRL,A
        MOV     A,INAH
        ADDC    A,ADH
        MOV     ADRH,A
        MOV     A,#00H
        ADDC    A,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOV     R2,A
POP     PSW
POP     ACC
RET
END
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมรับค่าข้อมูลจากผู้ใช้

Module

```
Public Buffer As Variant
Public BlockNO As Integer
Public DataBlock(0 To 1000, 1 To 13) As String
Public Blocksize As Integer
Public Itemsize As String
Public PName As String
Public Rname As String
Public Ritemsize As String
Public Rdatablock(0 To 1000, 1 To 13) As String
```

Form1

```
Private Sub Form_Load()
  With MSComm1
     .CommPort = 2
     .Settings = "9600,n,8,1"
     .PortOpen = True
  End With
Name1.SetFocus
End Sub
Private Sub Form_Unload(Cancel As Integer)
  MSComm1.PortOpen = False
  Form2.Hide
  Form3.Hide
End Sub

Private Sub GNO_Change()
Gtime.Enabled = True
M1time.Enabled = True
M1pos.Enabled = True
M1dir.Enabled = True
M2time.Enabled = True
M2pos.Enabled = True
M2dir.Enabled = True
M3time.Enabled = True
M3pos.Enabled = True
M3dir.Enabled = True
M4time.Enabled = True
M4pos.Enabled = True
M4dir.Enabled = True
OKButton.Enabled = True
If Val(GNO.Text) > GNO.ListCount + 1 Then
  GNO.Text = GNO.ListCount + 1
  MsgBox "หมายเลขบล๊อก (BlockNumber) จะต้องเรียงตามลำดับเท่านั้น", vbOKOnly
  Else
End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BlockNO = Val(GNO.Text)
Gtime.Text = DataBlock(BlockNO, 1)
M1time.Text = DataBlock(BlockNO, 2)
M1pos.Text = DataBlock(BlockNO, 3)
M1dir.Text = DataBlock(BlockNO, 4)
M2time.Text = DataBlock(BlockNO, 5)
M2pos.Text = DataBlock(BlockNO, 6)
M2dir.Text = DataBlock(BlockNO, 7)
M3time.Text = DataBlock(BlockNO, 8)
M3pos.Text = DataBlock(BlockNO, 9)
M3dir.Text = DataBlock(BlockNO, 10)
M4time.Text = DataBlock(BlockNO, 11)
M4pos.Text = DataBlock(BlockNO, 12)
M4dir.Text = DataBlock(BlockNO, 13)
End Sub

Private Sub GNO_Click()
BlockNO = Val(GNO.Text)
Gtime.Text = DataBlock(BlockNO, 1)
M1time.Text = DataBlock(BlockNO, 2)
M1pos.Text = DataBlock(BlockNO, 3)
M1dir.Text = DataBlock(BlockNO, 4)
M2time.Text = DataBlock(BlockNO, 5)
M2pos.Text = DataBlock(BlockNO, 6)
M2dir.Text = DataBlock(BlockNO, 7)
M3time.Text = DataBlock(BlockNO, 8)
M3pos.Text = DataBlock(BlockNO, 9)
M3dir.Text = DataBlock(BlockNO, 10)
M4time.Text = DataBlock(BlockNO, 11)
M4pos.Text = DataBlock(BlockNO, 12)
M4dir.Text = DataBlock(BlockNO, 13)
End Sub

Private Sub M1dir_Change()
  If Val(M1dir.Text) > 2 Then
     M1dir.Text = 2
     MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
  ElseIf Val(M1dir.Text) < 1 Then
     M1dir.Text = 1
     MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
     Else
  End If
  End Sub

Private Sub M2dir_Change()
 If Val(M2dir.Text) > 2 Then
     M2dir.Text = 2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
  ElseIf Val(M2dir.Text) < 1 Then
    M2dir.Text = 1
    MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
    Else
  End If
End Sub

Private Sub M3dir_Change()
 If Val(M3dir.Text) > 2 Then
    M3dir.Text = 2
    MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
  ElseIf Val(M3dir.Text) < 1 Then
    M3dir.Text = 1
    MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
    Else
  End If
End Sub

Private Sub M4dir_Change()
 If Val(M4dir.Text) > 2 Then
    M4dir.Text = 2
    MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
  ElseIf Val(M4dir.Text) < 1 Then
    M4dir.Text = 1
    MsgBox "ค่าแสดงทิศทางต้องเป็น 1 หรือ 2 เท่านั้น โดยที่ 1 คือ หมุนทวนเข็มนาฬิกา 2 หมุนตามเข็ม
นาฬิกา", vbOKOnly
    Else
  End If
End Sub

Private Sub Name1_Change()
  PName = Name1.Text
  GNO.Enabled = True
End Sub

Private Sub OKButton_Click()
If BlockNO > Val(GNO.ListCount) Then 'BlockNo ถูกควบคุมค่าอยู่ที่ GNO_on change
  GNO.AddItem BlockNO
  Else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End If
BlockNO = GNO.Text
DataBlock(BlockNO, 1) = Gtime.Text
DataBlock(BlockNO, 2) = M1time.Text
DataBlock(BlockNO, 3) = M1pos.Text
DataBlock(BlockNO, 4) = M1dir.Text
DataBlock(BlockNO, 5) = M2time.Text
DataBlock(BlockNO, 6) = M2pos.Text
DataBlock(BlockNO, 7) = M2dir.Text
DataBlock(BlockNO, 8) = M3time.Text
DataBlock(BlockNO, 9) = M3pos.Text
DataBlock(BlockNO, 10) = M3dir.Text
DataBlock(BlockNO, 11) = M4time.Text
DataBlock(BlockNO, 12) = M4pos.Text
DataBlock(BlockNO, 13) = M4dir.Text
If Val(Gtime.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M1time.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M1pos.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M1dir.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M2time.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M2pos.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M2dir.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M3time.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M3pos.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M3dir.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M4time.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M4pos.Text) <> 0 Then
   GoTo pass
   ElseIf Val(M4dir.Text) = 0 Then
      Uploadbutton.Enabled = True
      Save1.Enabled = True
      GoTo Zero
     Else
pass:   GNO.Text = BlockNO + 1
   Uploadbutton.Enabled = False
   Save1.Enabled = False
Zero:       End If
Blocksize = GNO.ListCount
Itemsize = Str$(GNO.ListCount)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub

Private Sub Read1_Click()
   Form3.Show
End Sub

Private Sub Save1_Click()
   Form2.Show
End Sub
```

Form2

```
Private Sub CancelButton_Click()
   Form2.Hide
End Sub

Private Sub Dir1_Change()
   File1.Path = Dir1.Path
   Path.Text = Dir1.Path & "\" & Fname.Text
  End Sub
Private Sub Drive1_Change()
   Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
   Path.Text = Dir1.Path & "\" & File1.FileName
   Fname.Text = File1.FileName
End Sub

Private Sub Fname_Change()
   Path.Text = Dir1.Path & "\" & Fname.Text
End Sub

Private Sub SaveButton_Click()
Dim FileNum As Integer
   FileNum = FreeFile
   Open Path.Text For Output As #FileNum
      Print #FileNum, PName
      Print #FileNum, Itemsize
      b% = 1
    Do
      Print #FileNum, DataBlock(b%, 1)
      Print #FileNum, DataBlock(b%, 2)
      Print #FileNum, DataBlock(b%, 3)
      Print #FileNum, DataBlock(b%, 4)
      Print #FileNum, DataBlock(b%, 5)
      Print #FileNum, DataBlock(b%, 6)
      Print #FileNum, DataBlock(b%, 7)
      Print #FileNum, DataBlock(b%, 8)
      Print #FileNum, DataBlock(b%, 9)
      Print #FileNum, DataBlock(b%, 10)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Print #FileNum, DataBlock(b%, 11)
        Print #FileNum, DataBlock(b%, 12)
        Print #FileNum, DataBlock(b%, 13)
        b% = b% + 1
        Loop Until b% = Blocksize
        Form2.Hide
        Close #FileNum
End Sub
```

Form3

```
Private Sub Command1_Click()
Dim FileNum As Integer
FileNum = FreeFile
    Open Read.Text For Input As FileNum
    Input #FileNum, Rname
    Input #FileNum, Ritemsize
    c% = 1
    Do While Not EOF(FileNum)
       Input #FileNum, Rdatablock(c%, 1)
       Input #FileNum, Rdatablock(c%, 2)
       Input #FileNum, Rdatablock(c%, 3)
       Input #FileNum, Rdatablock(c%, 4)
       Input #FileNum, Rdatablock(c%, 5)
       Input #FileNum, Rdatablock(c%, 6)
       Input #FileNum, Rdatablock(c%, 7)
       Input #FileNum, Rdatablock(c%, 8)
       Input #FileNum, Rdatablock(c%, 9)
       Input #FileNum, Rdatablock(c%, 10)
       Input #FileNum, Rdatablock(c%, 11)
       Input #FileNum, Rdatablock(c%, 12)
       Input #FileNum, Rdatablock(c%, 13)
       c% = c% + 1
    Loop
    Close #FileNum
    Debug.Print Rdatablock(3, 2)
    Debug.Print Rdatablock(3, 3)
    Debug.Print Rdatablock(4, 1)
    Debug.Print Rdatablock(4, 2)
    Debug.Print Rdatablock(4, 3)
    Form3.Hide
    End Sub

Private Sub Command2_Click()
    Form3.Hide
End Sub

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Drive1_Change()
   Dir1.Path = Drive1.Drive
End Sub

Private Sub File1_Click()
   Read.Text = Dir1.Path & "\" & File1.FileName
End Sub
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ค

แสดง L298N Datasheet

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)

Multiwatt15

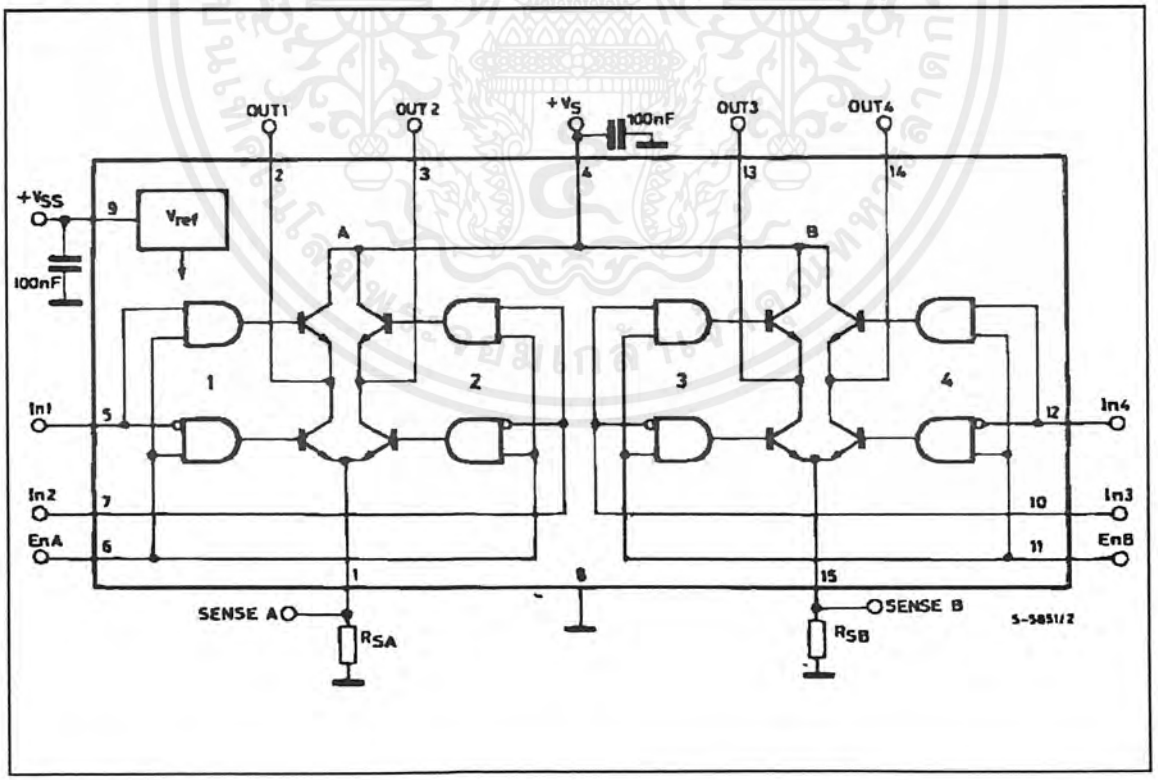
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

BLOCK DIAGRAM

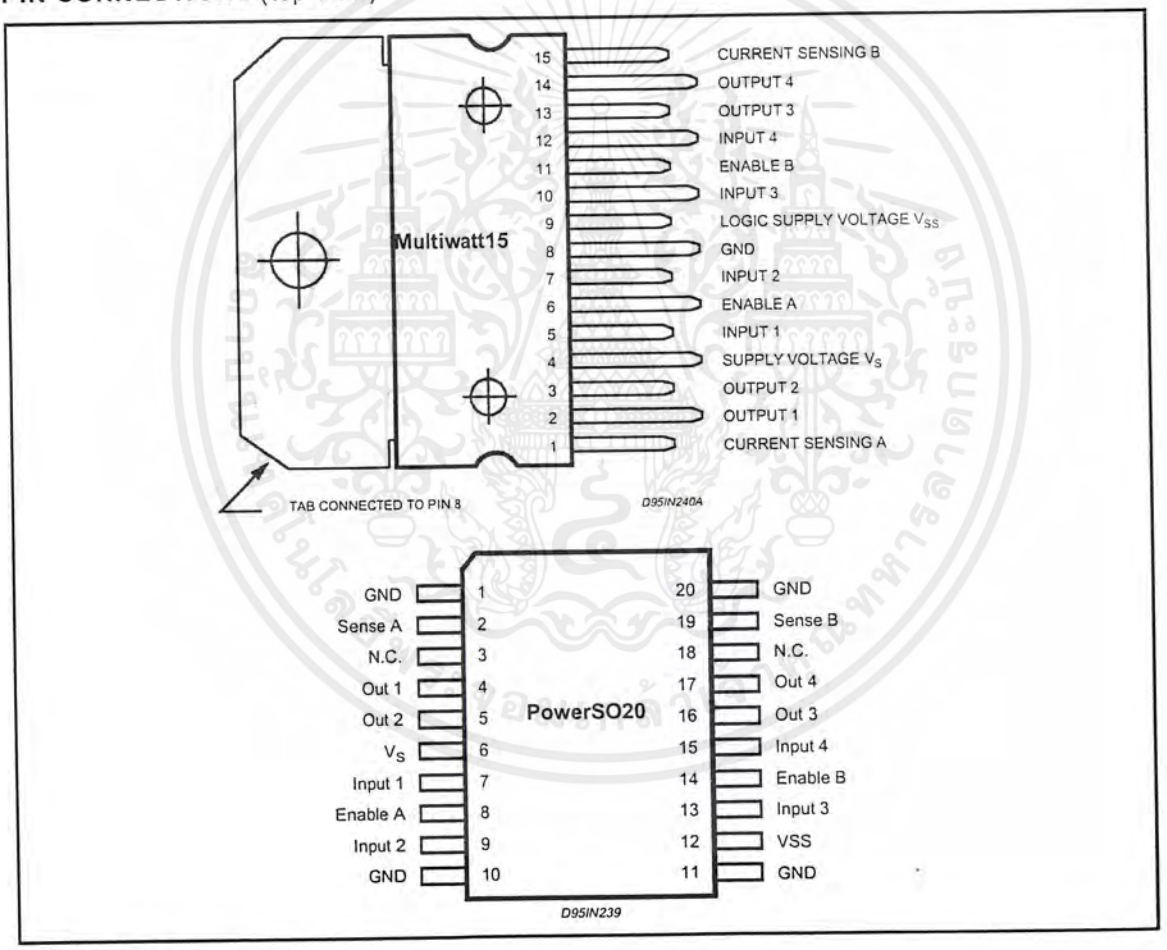


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i,V_{en}	Input and Enable Voltage	–0.3 to 7	V
I_O	Peak Output Current (each Channel) – Non Repetitive (t = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)



THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

(*) Mounted on aluminum substrate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_i = L V_i = H		13 50	22 70	mA mA
		V_{en} = L V_i = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_i = L V_i = H		24 7	36 12	mA mA
		V_{en} = L V_i = X			6	mA
V_{iL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{iH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{iL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_i = L			–10	µA
I_{iH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	µA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	µA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	µA
$V_{CEsat(H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat(L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		µs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		µs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		µs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		µs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		µs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		µs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		µs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		µs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		µs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		µs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		µs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		µs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		µs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		µs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		µs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		µs

1) 1)Sensing voltage can be –1 V for t ≤ 50 µsec; in steady state V_{sens} min ≥ –0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

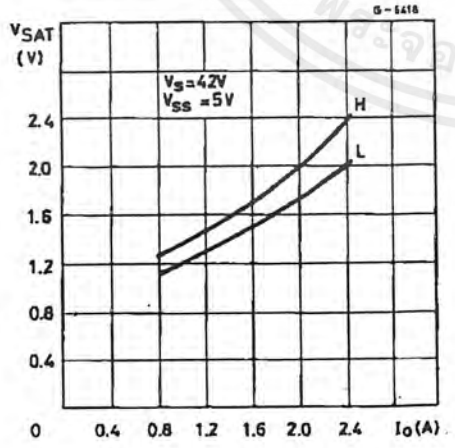
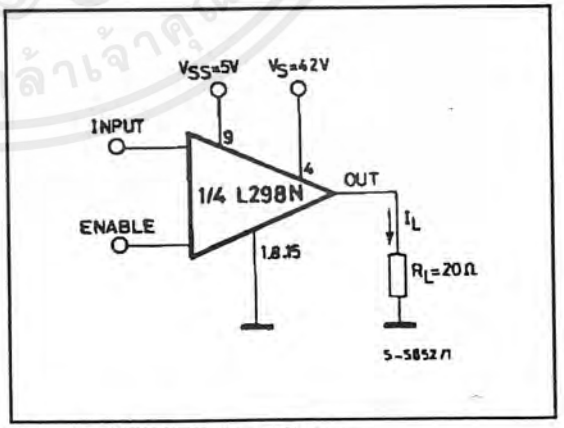


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

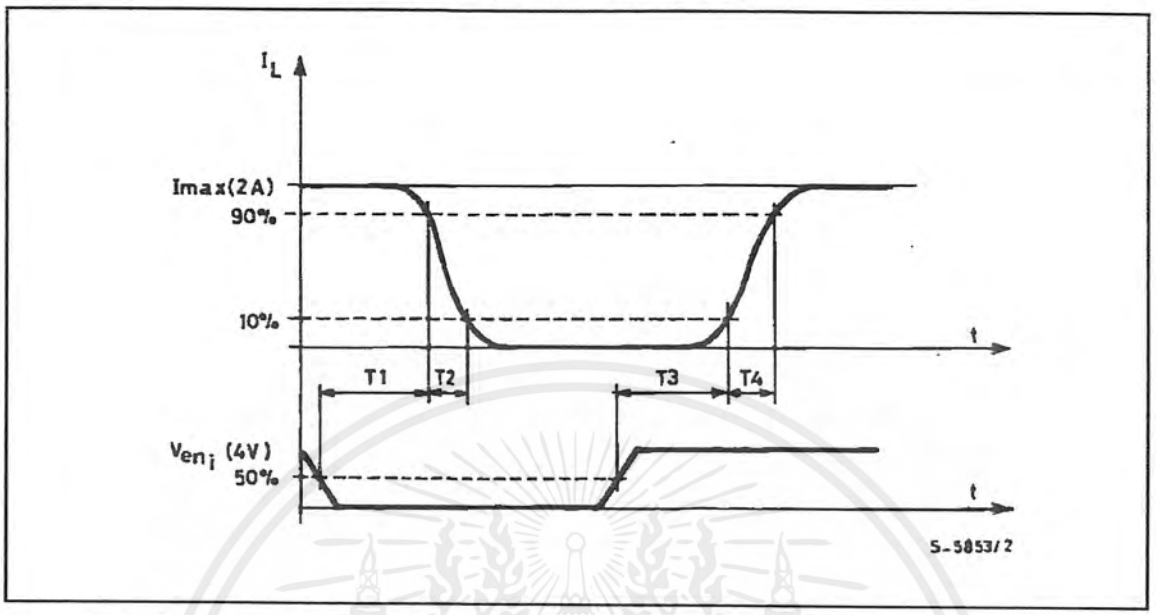
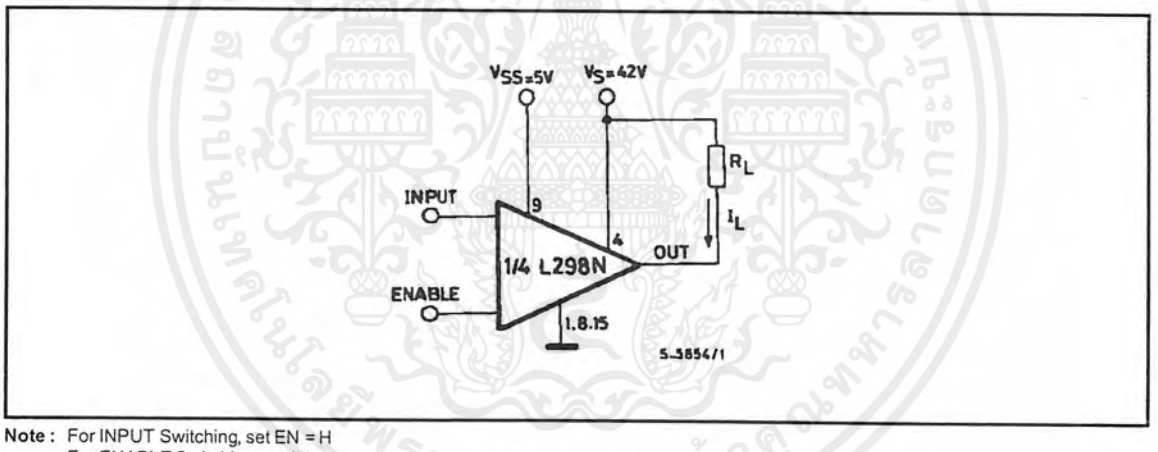


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

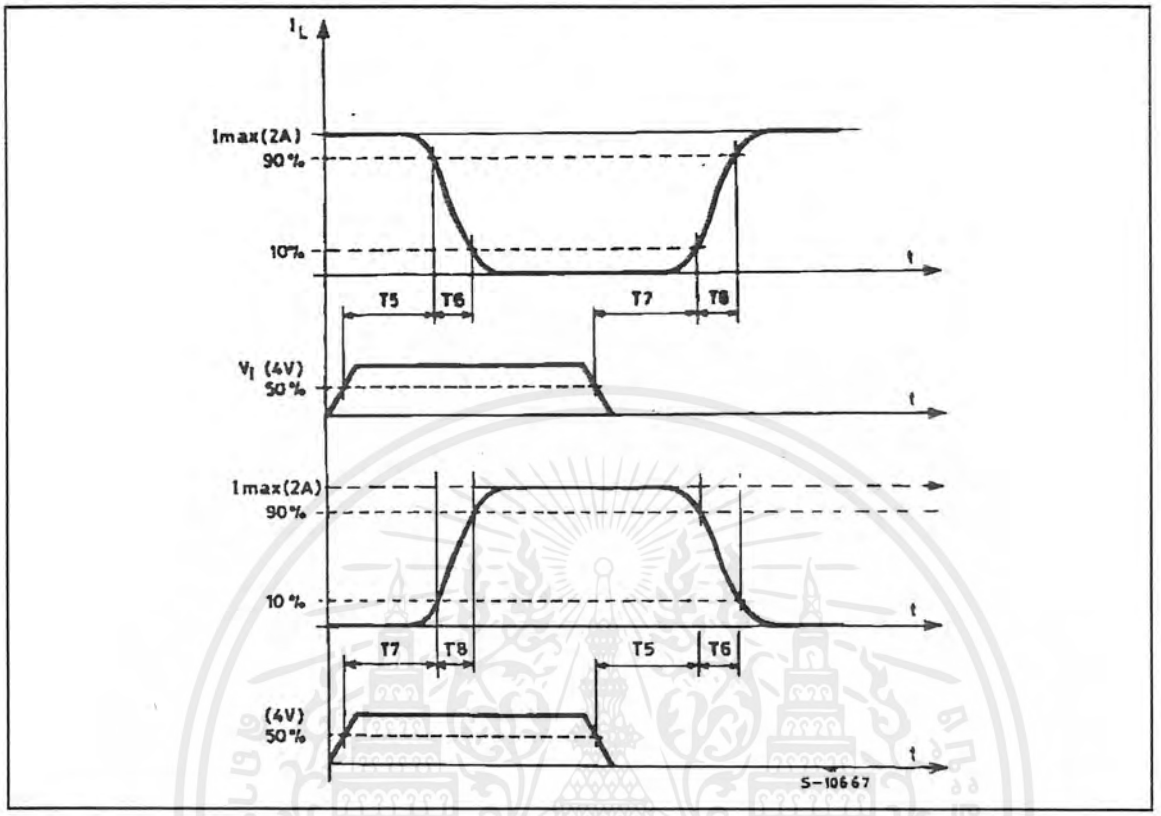
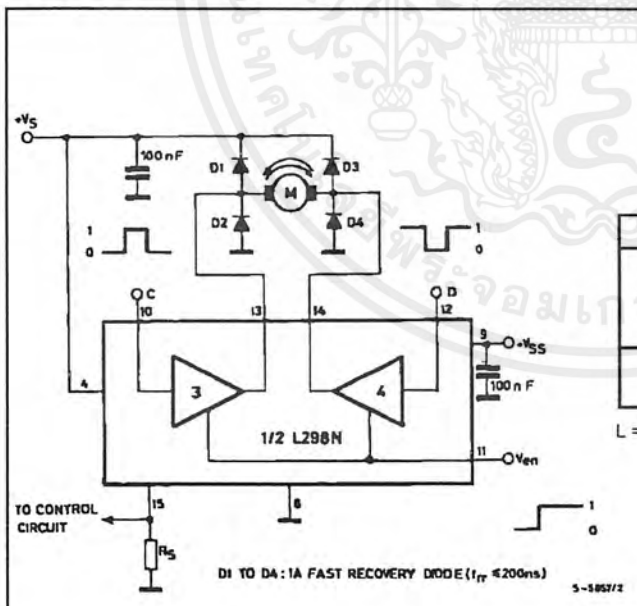


Figure 6 : Bidirectional DC Motor Control.

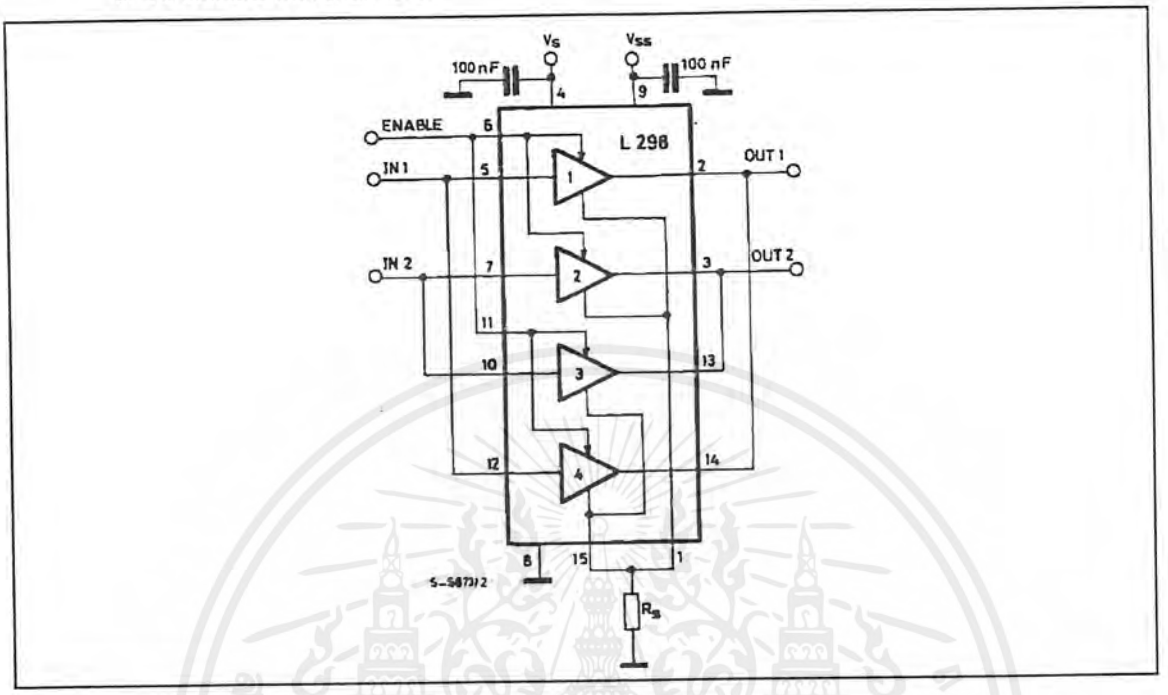


Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

The L298 integrates two power output stages (A ; B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differenzial mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output : an external resistor (R_{SA} ; R_{SB}.) allows to detect the intensity of this current.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a VF as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Shottky diodes would be preferred.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

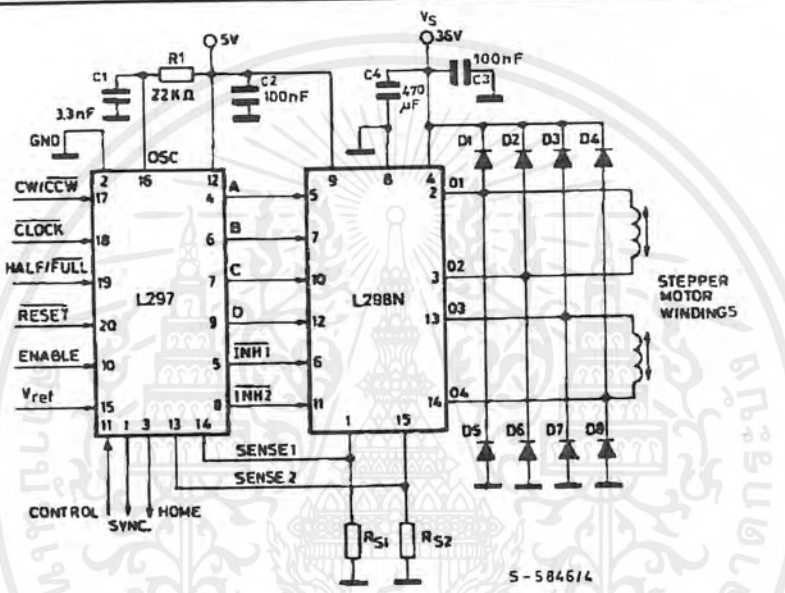
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\,\Omega$

D1 to D8 = 2 A Fast diodes { $V_F \leq 1.2\,V$ @ $I = 2\,A$; $trr \leq 200\,ns$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 9 : Suggested Printed Circuit Board Layout for the Circuit of fig. 8 (1:1 scale).

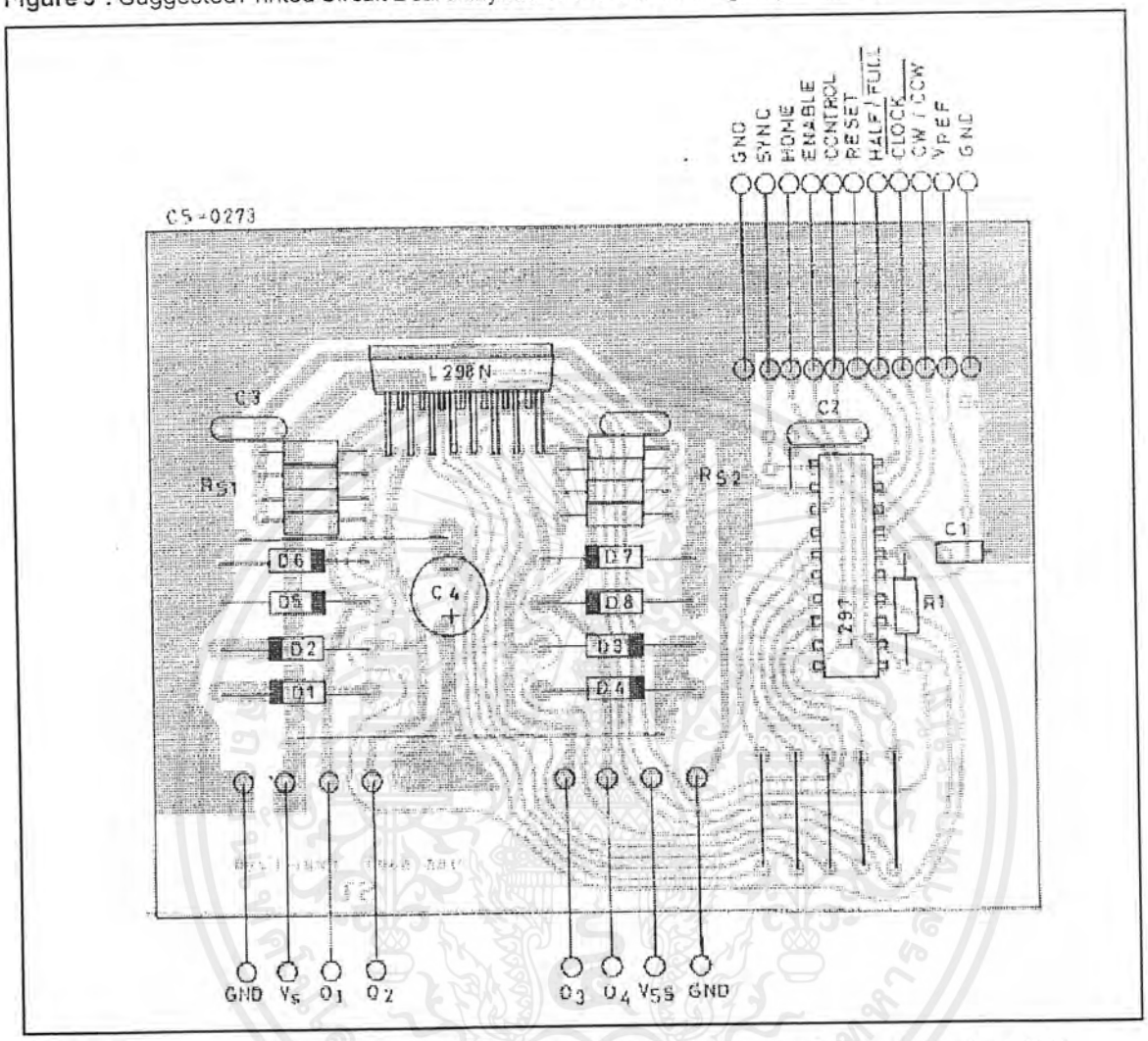
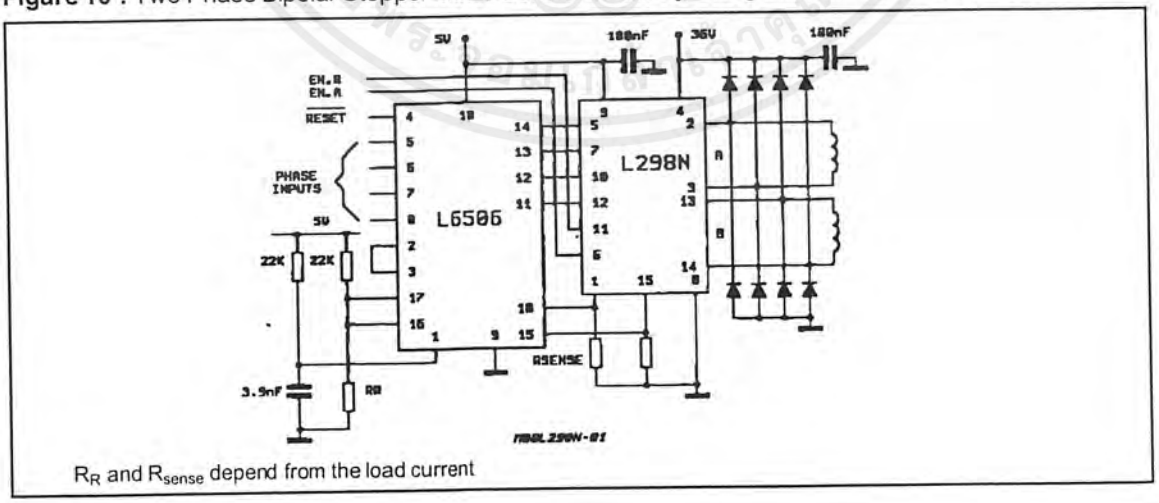


Figure 10 : Two Phase Bipolar Stepper Motor Control Circuit by Using the Current Controller L6506.

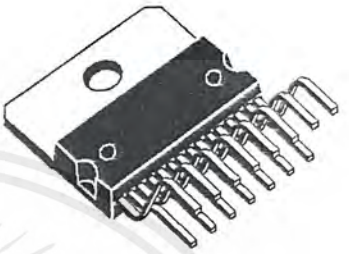


R_R and R_{sense} depend from the load current

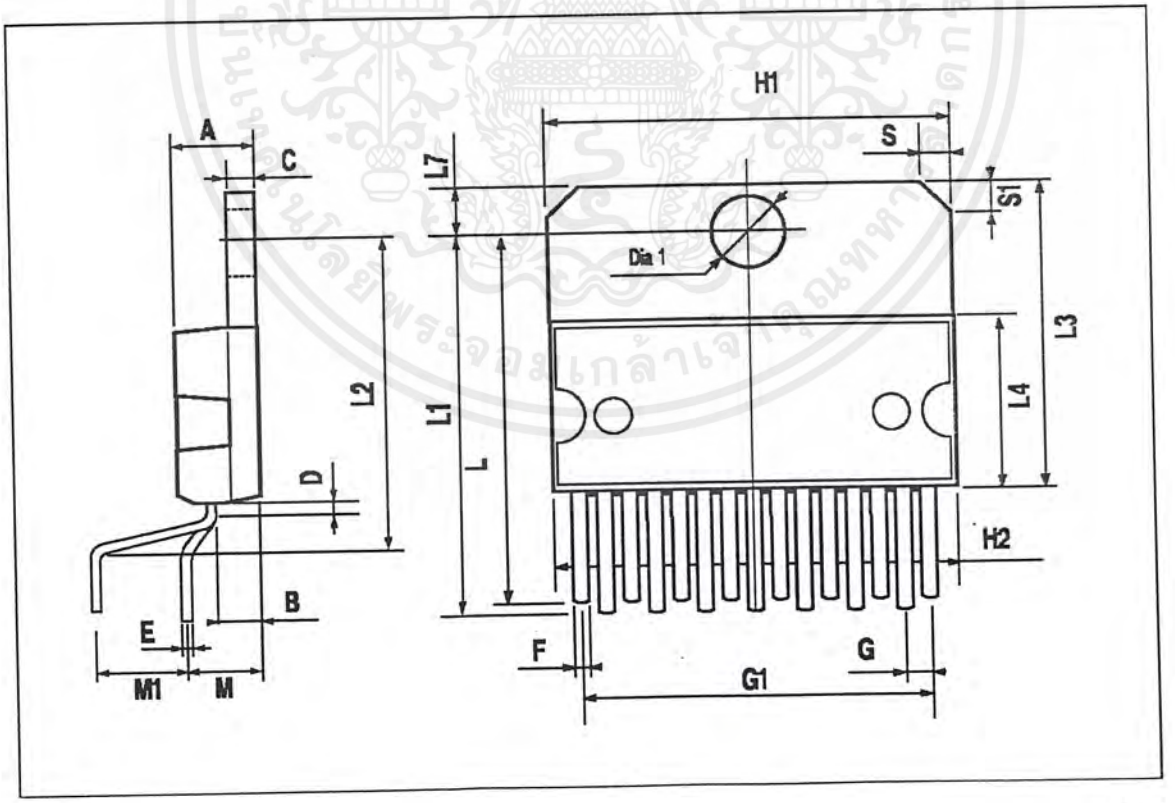
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
D		1			0.039	
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.02	1.27	1.52	0.040	0.050	0.060
G1	17.53	17.78	18.03	0.690	0.700	0.710
H1	19.6			0.772		
H2			20.2			0.795
L	21.9	22.2	22.5	0.862	0.874	0.886
L1	21.7	22.1	22.5	0.854	0.870	0.886
L2	17.65		18.1	0.695		0.713
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L7	2.65		2.9	0.104		0.114
M	4.25	4.55	4.85	0.167	0.179	0.191
M1	4.63	5.08	5.53	0.182	0.200	0.218
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



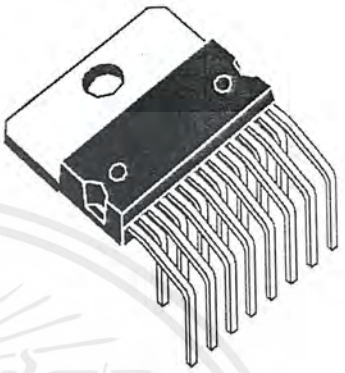
Multiwatt15 V



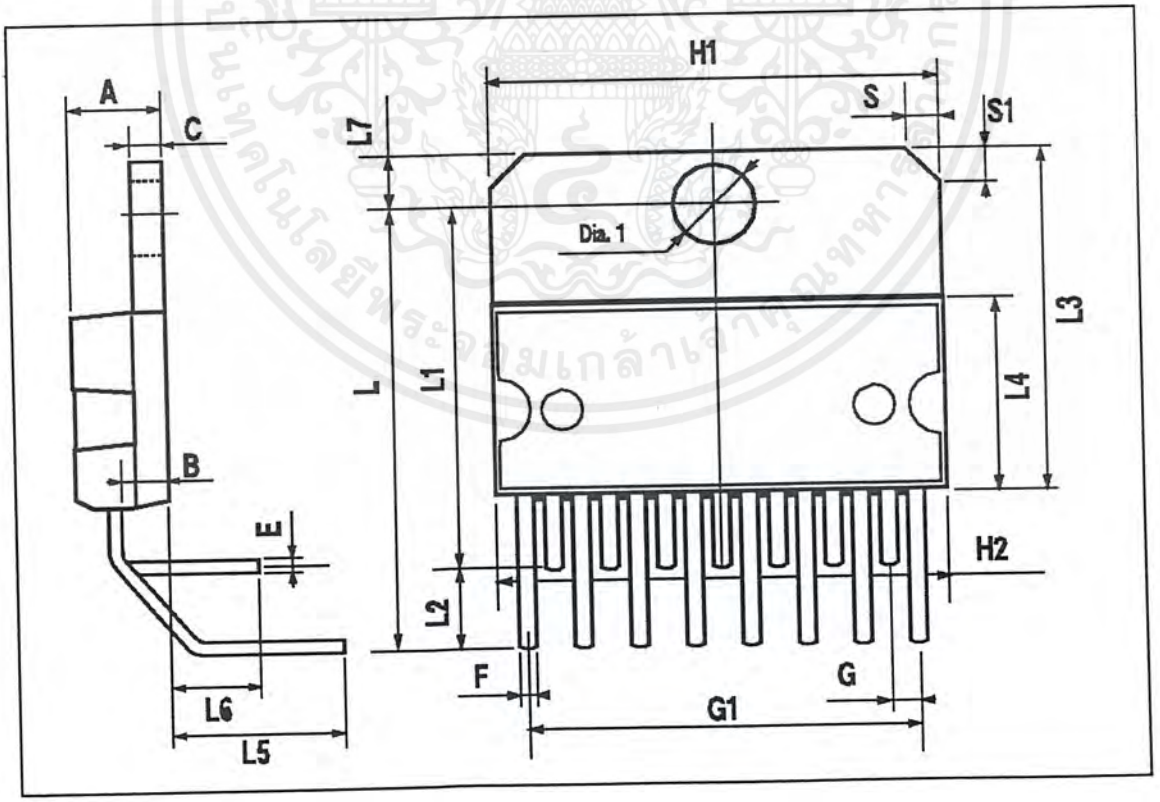
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			5			0.197
B			2.65			0.104
C			1.6			0.063
E	0.49		0.55	0.019		0.022
F	0.66		0.75	0.026		0.030
G	1.14	1.27	1.4	0.045	0.050	0.055
G1	17.57	17.78	17.91	0.692	0.700	0.705
H1	19.6			0.772		
H2			20.2			0.795
L		20.57			0.810	
L1		18.03			0.710	
L2		2.54			0.100	
L3	17.25	17.5	17.75	0.679	0.689	0.699
L4	10.3	10.7	10.9	0.406	0.421	0.429
L5		5.28			0.208	
L6		2.38			0.094	
L7	2.65		2.9	0.104		0.114
S	1.9		2.6	0.075		0.102
S1	1.9		2.6	0.075		0.102
Dia1	3.65		3.85	0.144		0.152

OUTLINE AND MECHANICAL DATA



Multiwatt15 H



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

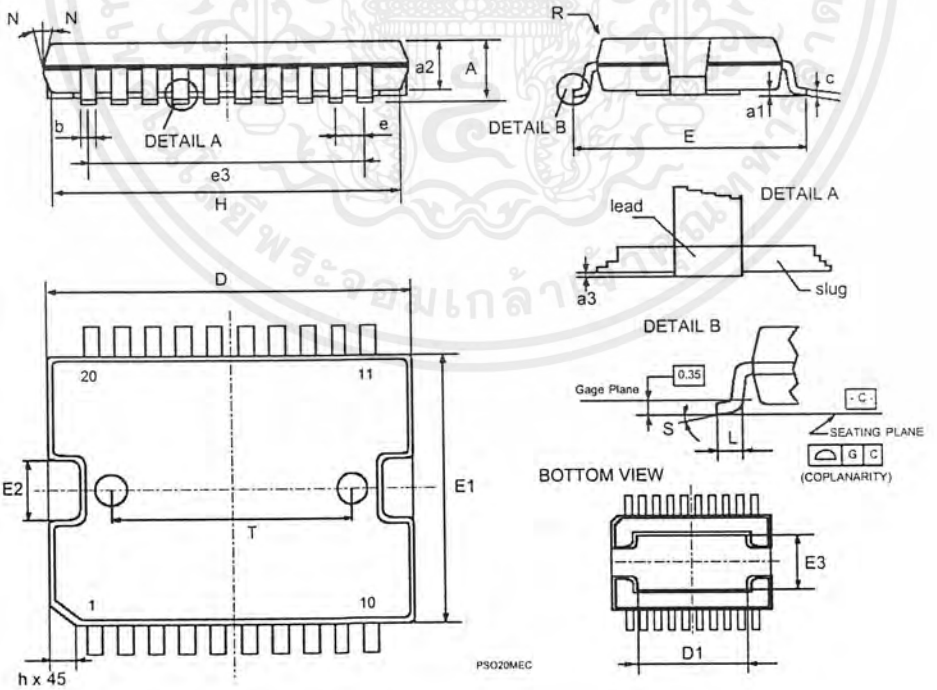
DIM.	mm			inch		
	MIN.	TYP.	MAX.	MIN.	TYP.	MAX.
A			3.6			0.142
a1	0.1		0.3	0.004		0.012
a2			3.3			0.130
a3	0		0.1	0.000		0.004
b	0.4		0.53	0.016		0.021
c	0.23		0.32	0.009		0.013
D (1)	15.8		16	0.622		0.630
D1	9.4		9.8	0.370		0.386
E	13.9		14.5	0.547		0.570
e		1.27			0.050	
e3		11.43			0.450	
E1 (1)	10.9		11.1	0.429		0.437
E2			2.9			0.114
E3	5.8		6.2	0.228		0.244
G	0		0.1	0.000		0.004
H	15.5		15.9	0.610		0.626
h			1.1			0.043
L	0.8		1.1	0.031		0.043
N	10° (max.)					
S	8° (max.)					
T		10			0.394	

(1) "D and F" do not include mold flash or protrusions.
- Mold flash or protrusions shall not exceed 0.15 mm (0.006").
- Critical dimensions: "E", "G" and "a3"

OUTLINE AND MECHANICAL DATA

JEDEC MO-166

PowerSO20



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Information furnished is believed to be accurate and reliable. However, STMicroelectronics assumes no responsibility for the consequences of use of such information nor for any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of STMicroelectronics. Specification mentioned in this publication are subject to change without notice. This publication supersedes and replaces all information previously supplied. STMicroelectronics products are not authorized for use as critical components in life support devices or systems without express written approval of STMicroelectronics.

The ST logo is a registered trademark of STMicroelectronics

© 2000 STMicroelectronics – Printed in Italy – All Rights Reserved

STMicroelectronics GROUP OF COMPANIES

Australia - Brazil - China - Finland - France - Germany - Hong Kong - India - Italy - Japan - Malaysia - Malta - Morocco - Singapore - Spain - Sweden - Switzerland - United Kingdom - U.S.A.

http://www.st.com

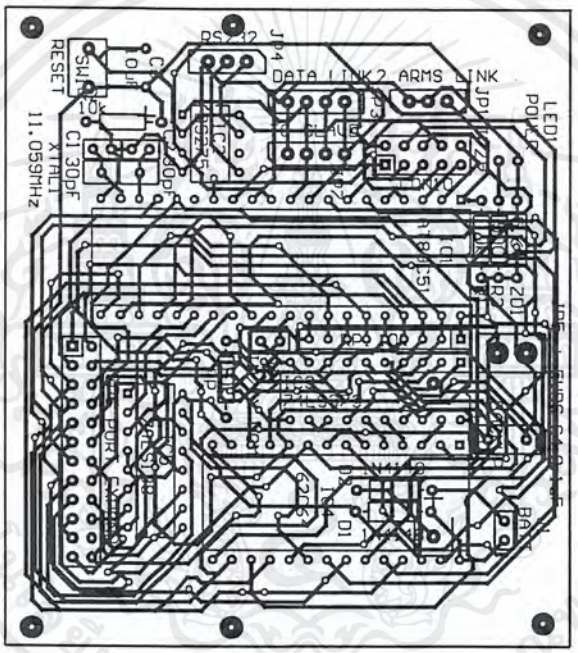
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ง

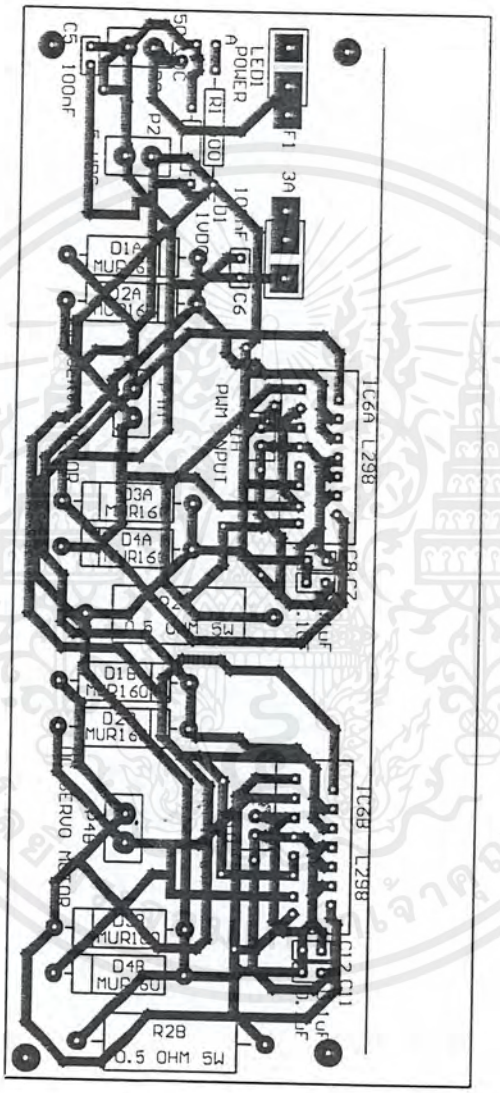
ลายวงจรของวงจรทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



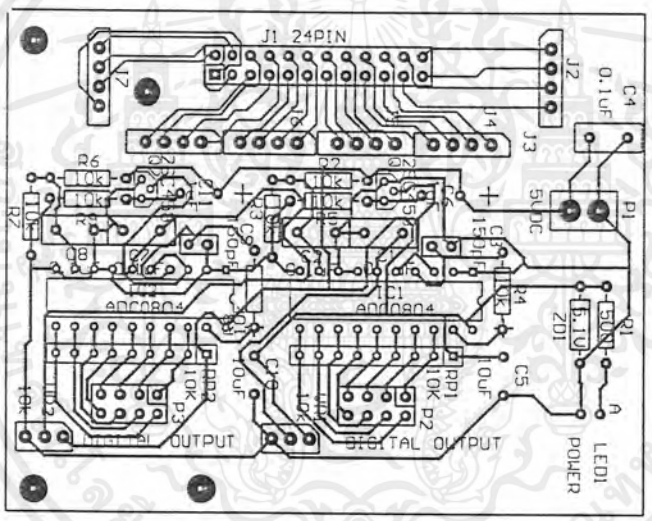
ลายวงจรของภาค Main Control

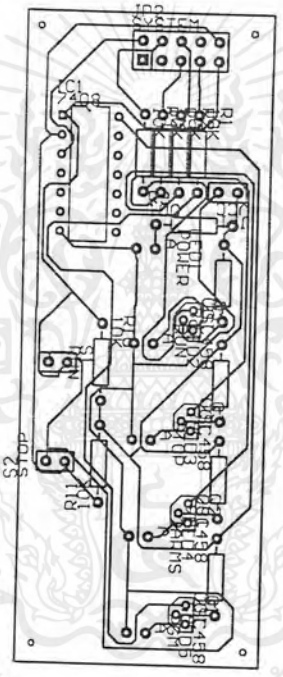
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลายวงจรของภาค Motor Driver

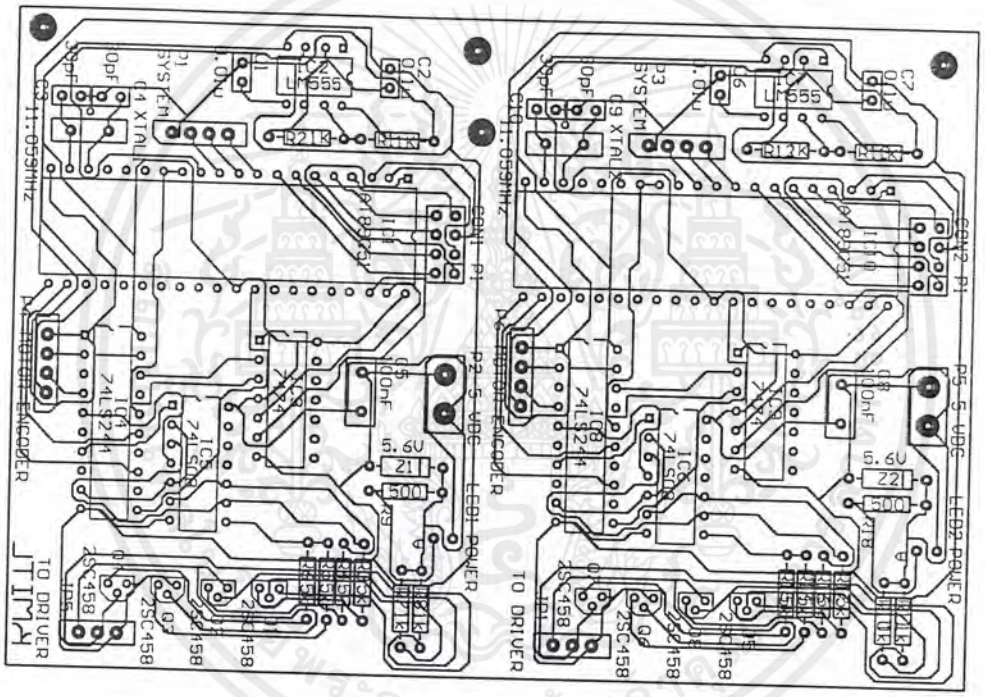
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


J1 24PIN
J2
C4
0.1uF
J3
5VDC
P1
R1
50Ω
ZD1
LED1
POWER
R4
C5
RP1
10K
RP2
10K
C10
C11
C12
C13
C14
C15
C16
C17
C18
C19
C20
C21
C22
C23
C24
C25
C26
C27
C28
C29
C30
C31
C32
C33
C34
C35
C36
C37
C38
C39
C40
C41
C42
C43
C44
C45
C46
C47
C48
C49



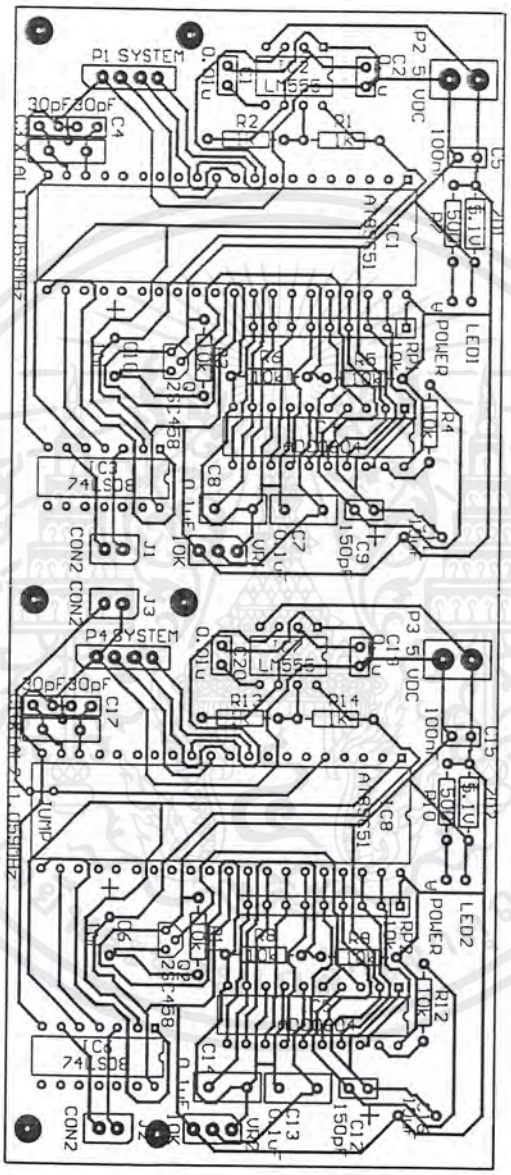
ลายวงจรของภาค DISPLAY & SWITCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



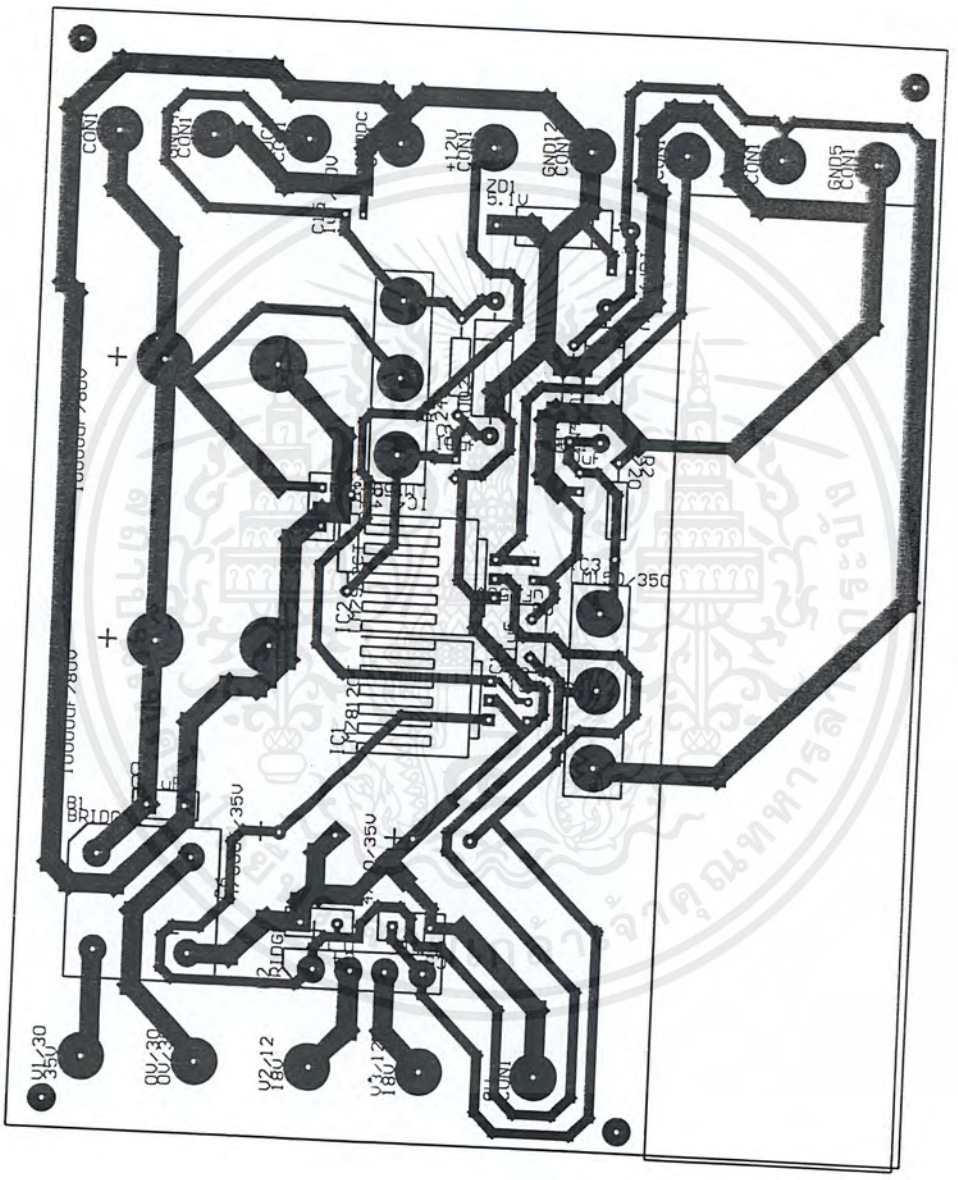
ลายวงจรของภาค PID1,2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลายวงจรของภาค PID3.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลายวงจรของภาคจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. อุดมจีน ประดับ " ไมโครคอนโทรลเลอร์ MCS- 51 " : สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
2. โยธิน เปรมปราณีรัชต์ " ระบบเซอร์โว และ อิเล็กทรอนิกส์คอนโทรลมอเตอร์ " : สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
3. Hutsushiko Ogata " Modern Control Engineering " Prentice – Hall

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้