

ระบบดัดลวดอลูมิเนียม
ALUMINIUM WIRE BENDING SYSTEM



กฤษฎา พิมพ์มดี
ชวลิต เกษกิง
ณัฐวุฒิ น้อยบุตรดี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

ระบบดัดลวดอลูมิเนียม
ALUMINIUM WIRE BENDING SYSTEM



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ คณะวิศวกรรมศาสตร์นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2556

ALUMINIUM WIRE BENDING SYSTEM



KRITSADA PIMMADEE
CHAWWALIT KATKING
NUTTHAVUD NOIBUDDEE

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN AUTOMATION ENGINEERING
FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานานาชาติเท่านั้น ไม่ควรนำออกนอกระบบโดยไม่ได้รับอนุญาตจากฝ่ายบริหาร
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงหรือเผยแพร่ข้อมูลใดๆของเอกสารทุกครั้งที่มีการนำไปใช้

สาขาวิชาวิศวกรรมอัตโนมัติ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

หัวข้อปริญญาานิพนธ์ ระบบตัดลวดอลูมิเนียม
ALUMINIUM WIRE BENDING SYSTEM
นักศึกษาผู้จัดทำ นายกฤษฎา พิมพ์ดี รหัสนักศึกษา 53010052
นายชวลิต เกษกิ่ง รหัสนักศึกษา 53010329
นายณัฐภูมิ น้อยบุตดี รหัสนักศึกษา 53010515
ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมอัตโนมัติ
ปีการศึกษา 2556

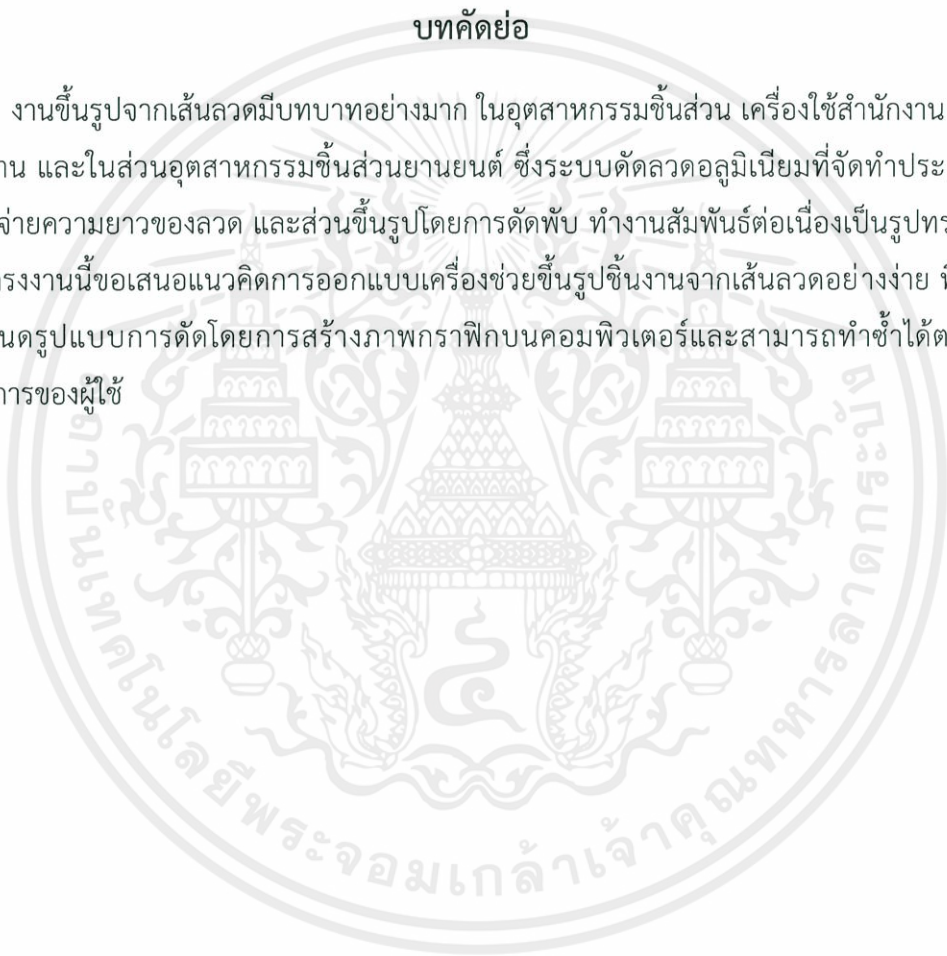
อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ประภาช อุคคกิมพานธุ์	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	ระบบดัดลวดอลูมิเนียม ALUMINIUM WIRE BENDING SYSTEM
นักศึกษาผู้จัดทำ	นายกฤษฎา พิมพม์ดี รหัสนักศึกษา 53010052 นายชวลิต เกษกิจ รหัสนักศึกษา 53010329 นายณัฐวุฒิ น้อยบุตดี รหัสนักศึกษา 53010515
อาจารย์ที่ปรึกษา	รศ.ประภาช อุดคคกิมพันธ์
ปีการศึกษา	2556

บทคัดย่อ

งานขึ้นรูปจากเส้นลวดมีบทบาทอย่างมาก ในอุตสาหกรรมชิ้นส่วน เครื่องใช้สำนักงาน เครื่องใช้ในบ้าน และในส่วนของอุตสาหกรรมชิ้นส่วนยานยนต์ ซึ่งระบบดัดลวดอลูมิเนียมที่จัดทำประกอบด้วย ส่วนจ่ายความยาวของลวด และส่วนขึ้นรูปโดยการดัดพับ ทำงานสัมพันธ์ต่อเนื่องเป็นรูปทรงสามมิติ ในโครงการนี้ขอเสนอแนวคิดการออกแบบเครื่องช่วยขึ้นรูปชิ้นงานจากเส้นลวดอย่างง่าย ที่สามารถกำหนดรูปแบบการดัดโดยการสร้างภาพกราฟิกบนคอมพิวเตอร์และสามารถทำซ้ำได้ตามความต้องการของผู้ใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	ALUMINIUM WIRE BENDING SYSTEM
Authors	MR. Kritsada Pimmadee MR. Chawwalit Katking MR. Nutthavud Noibuddee
Thesis Advisor	Assoc. Prof. Prapart Ukakimaparn
Year	2013

ABSTRACT

Wire Forming plays the important roles in industrial and automobile industrial parts, official and home equipment. The Aluminum wire bending system consists of wire feeding part and wire bending part which work relatedly to three dimension bending. This project is presented the idea of the simply wire forming machine designing which can make pattern by draw graphic on the computer repeatedly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

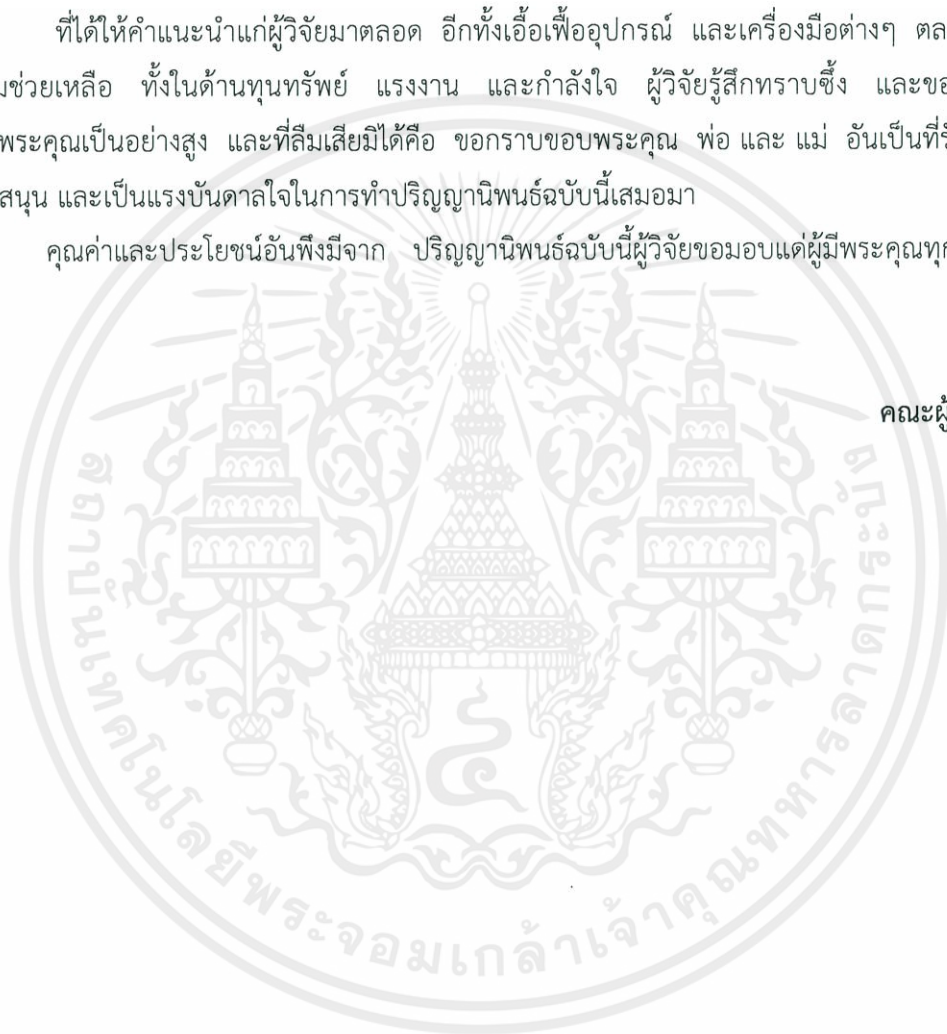
ปริญญาานิพนธ์และโครงการนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับความเมตตาจากหลายๆท่าน
ดังนี้

1. รองศาสตราจารย์ประภาส อุคคกิม่าพันธุ์ (อาจารย์ผู้ควบคุมปริญญาานิพนธ์)
2. นายเจตติยะ ศรีพิทักษ์

ที่ได้ให้คำแนะนำแก่ผู้วิจัยมาตลอด อีกทั้งเอื้อเพื่ออุปกรณ์ และเครื่องมือต่างๆ ตลอดจน
ความช่วยเหลือ ทั้งในด้านทุนทรัพย์ แรงงาน และกำลังใจ ผู้วิจัยรู้สึกทราบบ้าง และขอกราบ
ขอบพระคุณเป็นอย่างสูง และที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณ พ่อ และ แม่ อันเป็นที่รักยิ่งที่
สนับสนุน และเป็นแรงบันดาลใจในการทำปริญญาานิพนธ์ฉบับนี้เสมอมา

คุณค่าและประโยชน์อันพึงมีจาก ปริญญาานิพนธ์ฉบับนี้ผู้วิจัยขอบแต่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	2
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 ทฤษฎี.....	3
2.1 ไมโครคอนโทรลเลอร์.....	3
2.1.1 Chip และ IC ภายใน Arduino Board ที่สำคัญ.....	5
2.1.1.1 ATmega328P-PU	5
2.1.1.2 MAX3421E	9
2.1.1.3 FT232RL.....	11
2.1.2 Software Arduino Environment.....	13
2.2 สเต็ปมอเตอร์	14
2.2.1 แบบ Bipolar.....	16
2.2.2 แบบ Unipolar.....	16
2.3 การสั่งงานควบคุมการหมุนของสเต็ปมอเตอร์	17
2.3.1 การควบคุมการหมุนแบบเฟสเดียว (Single Phase) หรือ Wave Drive....	17
2.3.2 การควบคุมการหมุนแบบสองเฟส (Two Phase) หรือ Full Step Drive..	17
2.3.3 การควบคุมการหมุนแบบครึ่งเฟส (Half Step).....	18

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับอาจารย์และบุคลากรในมหาวิทยาลัยเท่านั้น ไม่สามารถนำเอกสารนี้ไปใช้เพื่อประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังขอสงวนสิทธิ์ในเอกสารทุกครั้งที่มีการปรับปรุงแก้ไข

สารบัญ (ต่อ)

	หน้า
2.3.4 ข้อดีของสแต็ปมอเตอร์เมื่อเทียบกับมอเตอร์กระแสตรง.....	18
2.4 Servo motor.....	19
2.4.1 โครงสร้างระบบควบคุมเซอร์โว.....	19
2.4.2 ระบบเซอร์โว.....	20
2.4.3 ประเภทของเซอร์โว.....	20
2.5 ทฤษฎีการงอ.....	21
2.5.1 เส้นแกนกลาง.....	23
2.5.2 การไหลตัวของโลหะในการขึ้นรูป.....	24
2.5.2.1 ลักษณะของการไหลตัวของโลหะระหว่างการขึ้นรูป.....	25
2.5.2.2 ขนาดของการไหลตัวของโลหะซึ่งมีความสัมพันธ์กับสิ่งต่างๆ.....	25
2.5.3 การกระเด็งตัวกลับของชิ้นงาน.....	25
2.3.5.1 วิธีป้องกันการเกิดการกระเด็งตัวกลับ.....	27
บทที่ 3 วิธีการดำเนินงาน.....	28
3.1 เครื่องตัดลวด.....	28
3.1.1 ชนิดของลวดที่ใช้ตัด.....	28
3.1.2 รัศมีที่ใช้ตัด.....	29
3.1.3 มุมเพื่อสำหรับการสปริงตัวกลับของลวด.....	29
3.1.4 ขนาดของ Pin ตัด.....	30
3.1.5 แรงที่ใช้ในการตัดลวด.....	30
3.1.6 แรงจากความเสียดทาน.....	31
3.1.7 แรงที่กระทำกับ Pin.....	31
3.1.8 ความเค้นดัดที่เกิดขึ้นกับ Pin.....	32
3.1.9 ขนาดของ Pin ที่เหมาะสม.....	33
3.1.10 จำนวน Roller ที่ต้องใช้.....	34
3.1.11 ขนาดของมอเตอร์ส่วนต่างๆ.....	34
3.1.11.1 ขนาดของมอเตอร์ที่จะทำการ Feed.....	34
3.1.11.2 มอเตอร์ที่ต้องใช้ในการหมุนแกนดัด.....	35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มิใช่ผู้ผลิตห้ามใช้หรือปรับหรือแก้ไขการตั้งค่า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีกรณาไปใช้

สารบัญ (ต่อ)

	หน้า
3.1.12 เลือกขนาดมอเตอร์.....	36
3.1.13 การเขียนแบบ.....	37
3.1.14 การทำแบบขึ้นเฉพาะ.....	37
3.1.15 ขั้นตอนการสร้างเครื่อง.....	38
3.1.16 ส่วนประกอบหลักของเครื่อง.....	40
3.2 ชุดควบคุม.....	41
3.2.1 การต่อระบบการจ่ายไฟ.....	41
3.2.2 ลำดับการสั่งส่วนควบคุม.....	43
3.2.2 ปุ่มควบคุม.....	44
3.3 ชุดกำหนดรูปแบบการตัด.....	45
บทที่ 4 การทดลอง และผลการทดลอง.....	46
4.1 ระยะเวลา Feed ลวด.....	46
4.2 มุมองศาการตัด.....	46
บทที่ 5 บทวิจารณ์และสรุป.....	48
5.1 บทวิจารณ์.....	48
5.2 สรุป.....	48
5.3 ข้อเสนอแนะ.....	48
เอกสารอ้างอิง.....	49
ภาคผนวก.....	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่	หน้า
2.1 ลำดับการป้อนกระแสแบบเฟสเดียว.....	17
2.2 ลำดับการป้อนกระแสแบบสองเฟส.....	17
2.3 ลำดับการป้อนกระแสแบบครึ่งเฟส.....	18
3.1 แสดงค่าเฉพาะของลวด Aluminum 1350-O.....	28
3.2 แสดงจำนวนของ rolls ที่ใช้ตัด.....	34
4.1 แสดงระยะที่ Feed ลวดจริง.....	45
4.2 แสดงมุมตัดของลวด.....	46



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่	หน้า
2.1 Arduino Board.....	3
2.2 ซีพียู ATMEGA328P ขนาด 28 ขา.....	5
2.3 เปรียบเทียบการจัดการหน่วยความจำของสถาปัตยกรรม.....	6
2.4 แสดง pin ของ Arduino Board.....	7
2.5 Block Diagram	8
2.6 แสดงการเชื่อมต่อ MAX3421E กับ Microprocessor ด้วย pin 3, 4	9
2.7 Pin Configurations.....	10
2.8 IC FT232RL.....	11
2.9 แสดง Block Diagram ของ FT232RL.....	11
2.10 แสดง Device Pin Out	12
2.11 แสดง Schematics ของ FT232RL ต่อกับ USB.....	12
2.12 แสดงหน้าตาโปรแกรมพัฒนา Arduino Board.....	13
2.13 สเต็ปมอเตอร์แบบ 5 เส้น และสเต็ปมอเตอร์แบบสาย 6 เส้น.....	14
2.14 แสดงรูปโครงสร้างภายในของสเต็ปมอเตอร์.....	14
2.15 การควบคุมระบบสเต็ปมอเตอร์.....	15
2.16 แสดงรูปโครงสร้างระบบควบคุมเซอร์โว.....	19
2.17 เซอร์โวไดรฟ์และมอเตอร์เซอร์โว.....	20
2.18 การแสดงถึงการเปลี่ยนแปลงของความเค้นดึง และความเค้นอัด.....	21
2.19 การเกิดการติดตัวกลับของชิ้นงานหลังการขึ้นรูป.....	22
2.20 ความเค้นที่เกิดขึ้นในบริเวณพื้นที่ของการงอ.....	22
2.21 ตำแหน่งของเส้นแกนกลาง.....	23
2.22 การไหลของโลหะและการบิดงอ.....	24
2.23 การกระด้างตัวกลับของโลหะ SAE 1008 Aluminums-Killed Cold.....	26
3.1 แสดงรัศมีดัดที่ยอม สำหรับวัสดุชิ้นงานที่ไม่ใช่เหล็ก สำหรับโลหะแผ่นหนาต่างกัน.....	29
3.2 แสดงแกนดัด และตัวหัวตัดรวมถึง Pin ดัด.....	33
3.3 แสดงสเต็ปมอเตอร์ Vexta.....	36
3.4 แสดงเซอร์โวมอเตอร์ Yokogawa.....	36
3.5 แสดงสเต็ปมอเตอร์ mycom.....	36
3.6 แสดงโปรแกรมที่ใช้เขียนแบบ.....	37

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆก็ตาม หากมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อผู้จัดทำเอกสารทุกครั้งที่มาไปใช้

สารบัญญภาพ (ต่อ)

ภาพที่	หน้า
3.7 แสดงแบบการต่อไฟฟ้า.....	38
3.8 แสดงแบบที่กึ่ง.....	39
3.9 แสดงการทำโครงประกอบ.....	39
3.10 แสดงส่วนประกอบหลักของเครื่อง.....	40
3.11 แสดงแบบการต่อไฟฟ้า.....	41
3.12 แสดงลำดับการสั่งงานส่วนควบคุม.....	43
3.13 แสดงปุ่มควบคุมแต่ละส่วน.....	44
3.14 โปรแกรม Inkscape และ text file.....	45
3.15 รูปแบบของสัญญาณที่ส่งไปยัง driver ของมอเตอร์.....	45
4.1 แสดงมมดัดที่ทดลอง 10-160 องศา.....	47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของปริญญานิพนธ์

เครื่องตัดสามารถเปลี่ยนวัสดุให้มีรูปร่างแตกต่างกันตามแบบที่กำหนด สิ่งที่ถูกตัดมีความสำคัญต่อการนำไปใช้ในงานโครงสร้าง หรือสิ่งประดิษฐ์ แต่โดยมากเครื่องตัดในปัจจุบันสามารถทำให้วัสดุโค้งงอได้เพื่อทนแรงและให้ความแม่นยำขึ้นเท่านั้น ดังตัวอย่างโครงการเกี่ยวกับการตัดโลหะเรื่อง “เครื่องตัดท่อ, 2547” ใช้การโปรแกรมผ่าน PLC ในการควบคุมและสั่งงาน ใช้กลไกทางด้าน pneumatic ในการออกแรงเพื่อตัดท่อให้ได้มุมตามต้องการ สามารถวัดความยาวและหมุนท่อได้ และโครงการ “การตัดโลหะแผ่นโดยการควบคุมจากไมโครคอนโทรลเลอร์, 2546” สามารถตัดโลหะแผ่นให้มีความโค้งงอตามแบบได้ แต่แบบที่ใช้ตัดต้องมีการเปลี่ยนทุกครั้งเมื่อต้องการเปลี่ยนรูปแบบของโลหะที่ต้องการตัด การทำงานของทั้งสองโครงการนี้ยังไม่สามารถทำงานเป็นขั้นตอนผ่านการสั่งงานจากคอมพิวเตอร์ได้ ต้องอาศัยการป้อนโลหะให้ได้ระยะตามที่ต้องการทุกครั้งเมื่อต้องการเปลี่ยนลักษณะการตัดทำให้สิ้นเปลืองเวลาในการทำงานอีกทั้งเป็นการทำงานโดยใช้คน มีข้อจำกัดหลายด้าน โดยเฉพาะอย่างยิ่งหากเป็นการผลิตจำนวนมาก เมื่อมีการเปลี่ยนตัวคนที่มาทำงาน ผลงานที่ได้ก็อาจเปลี่ยนไป แม้ในต่างประเทศจะมีการผลิตเครื่องตัดลวดที่สามารถทำงานได้โดยอัตโนมัติตัดได้หลายรูปแบบโดยใช้ CNC ควบคุม สามารถทำงานได้เป็นกระบวนการ ทำให้ประหยัดเวลา คุณภาพของงานที่ได้มีความสม่ำเสมอ แต่การใช้เทคโนโลยี CNC (Computer Numerical Control) ต้องใช้องค์ความรู้เฉพาะในการสร้างเครื่องจักร ทำให้ยากที่จะดัดแปลงเพื่อการนำไปใช้ทำประโยชน์เพิ่มเติมจากเครื่องที่ได้ถูกสร้างขึ้นมาเพื่อจำหน่ายและใช้งานเฉพาะด้านตามที่ผู้สร้างได้กำหนดไว้ อีกทั้งต้องใช้เงินทุนสูงในการซื้อเครื่องจักรและต้องติดตั้งและบำรุงรักษา ระบบตัดลวดออลูมิเนียมสร้างเป็นต้นแบบเพื่อให้เกิดการพัฒนาให้การตัดสามารถทำงานได้เป็นระบบ และสามารถพัฒนาประยุกต์ใช้ได้

ระบบตัดลวดออลูมิเนียมสั่งงานผ่านโปรแกรมทางคอมพิวเตอร์ โดยนำ open source software (Processing, Arduino, Inkscape) มาประยุกต์ใช้ เครื่องตัดลวดออลูมิเนียมออกแบบสร้างจากหลักการตัดทั่วไป ความรู้พื้นฐานทางฟิสิกส์กลศาสตร์ และวัสดุศาสตร์ สามารถป้อนและตัดลวดตามการสั่งงาน ลดความยุ่งยากในการเปลี่ยนลักษณะการตัดแต่ละครั้ง และจะเกิดประโยชน์ต่อการพัฒนาประยุกต์ใช้ในงานตัดเหล็กหรือท่อได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของปริญญาโท

1. เพื่อสร้างเครื่องตัดลวดที่สามารถตัดตามการสั่งงานผ่านคอมพิวเตอร์
2. เพื่อสร้างชุดโปรแกรมควบคุมสั่งงานเครื่องตัดลวดช่วยลดความยุ่งยากในการตัด

1.3 ขอบเขตของปริญญาโท

1. ออกแบบและสร้างเครื่องตัดลวดอลูมิเนียม ขนาดเส้นผ่านศูนย์กลาง 2.6 มิลลิเมตร
2. เชื่อมต่อระหว่างกลไกเครื่องตัดลวดกับคอมพิวเตอร์ให้ทำงานร่วมกันผ่านการโปรแกรม
3. สร้างชุดโปรแกรมคอมพิวเตอร์เพื่อสั่งงานเครื่องตัดลวด

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาหลักการและวิธีการตัดลวดโดยใช้เครื่องมือพื้นฐานและการตัดโดยใช้เครื่อง
2. ศึกษาคุณสมบัติของลวด และเลือกลวดที่เหมาะสมกับงาน
3. ศึกษาแนวทางการจัดท่าระบบสั่งการเครื่องตัดลวดผ่านคอมพิวเตอร์
4. ออกแบบชิ้นส่วน และอุปกรณ์ที่เป็นส่วนประกอบของเครื่องและวิธีการประกอบรวมเครื่อง
5. สร้างเครื่องตัดลวดและระบบสั่งการ
6. สร้างโปรแกรมควบคุม
7. ทดลองเชื่อมต่อและสั่งเครื่องทำงานผ่านคอมพิวเตอร์
8. ปรับปรุงให้ระบบสามารถตัดลวดได้ตามที่กำหนด
9. สรุป และวิเคราะห์ผลการดำเนินงาน

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เกิดการเข้าใจในงานโลหะมากขึ้น และการนำเอาความรู้ไปประยุกต์ใช้
2. ได้รับประสบการณ์ในการออกแบบ
3. สามารถนำโครงการนี้ไปพัฒนาให้ดีขึ้น และนำไปใช้ประโยชน์ได้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Arduino Board



รูปที่ 2.1 Arduino Board

Arduino เป็น platform ของ Input/Output Board ที่มี Input/Output ชั้นพื้นฐานที่พอเพียงกับการใช้งานและการเรียนรู้โดย Board มีกับชุดคำสั่งที่ใช้ควบคุม port Input/Output ได้แก่ port digital, port analog, PWM และ Serial port ซึ่ง Arduino เป็นเครื่องมือที่จะทำให้คอมพิวเตอร์สามารถรับสัญญาณจากภายนอกและส่งสัญญาณไปควบคุมอุปกรณ์ภายนอกได้อย่างมีประสิทธิภาพมากกว่าใช้คอมพิวเตอร์ส่วนบุคคล ตัวบอร์ดออกแบบจากไมโครคอมพิวเตอร์ชิพเดี่ยว และมีโปรแกรมพัฒนาสำหรับเขียนโปรแกรมให้ Arduino Board สามารถประยุกต์ทำเครื่องใช้อัจฉริยะ รับสัญญาณจากสวิทช์ หรือ เซนเซอร์ และ ควบคุม หลอดไฟ, มอเตอร์, หรืออุปกรณ์อื่นๆ โครงการงาน Arduino เป็นได้ทั้งการทำงานอิสระ หรือทำงานติดต่อกับโปรแกรมที่ทำงานบนคอมพิวเตอร์ส่วนบุคคล ตัวบอร์ดสามารถประกอบขึ้นใช้เอง หรือซื้อสำเร็จ

Arduino เป็นภาษาอิตาลี ซึ่งใช้เป็นชื่อของโครงการพัฒนาไมโครคอนโทรลเลอร์ตระกูล AVR แบบ Open Source ที่ได้รับการปรับปรุงมาจากโครงการพัฒนา Open Source ของ AVR อีกโครงการหนึ่งซึ่งชื่อว่า “Wiring” แต่เนื่องจากโครงการของ “Wiring” เลือกใช้ AVR หมายเลข ATmega128 ซึ่งเป็นไมโครคอนโทรลเลอร์ที่มีจำนวนของหน่วยความจำ และ Input/Output ค่อนข้างมากและที่สำคัญ ATmega128 เป็นชิพที่มีรูปร่างแบบ SMD จึงทำให้เป็นอุปสรรคสำหรับผู้เริ่มต้นในการสร้างบอร์ดและต่อวงจรขึ้นมาใช้งานกันเอง และบอร์ดจะมีขนาดค่อนข้างใหญ่ ซึ่งอาจดูว่าเกินความจำเป็นสำหรับผู้เริ่มต้นจึงไม่ค่อยได้รับความนิยมเท่าที่ควร แต่หลังจากที่ทีมงาน Arduino

นำ Source Code ของ “Wiring” มาพัฒนาปรับปรุงใหม่โดยให้สามารถใช้งานกับไมโครคอนโทรลเลอร์ขนาดเล็ก Mega8 , Mega168, Mega328 ได้จึงทำให้ระบบวงจรของบอร์ดมีขนาดเล็กกว่า “Wiring” มากและยังใช้อุปกรณ์น้อยชิ้นทำให้ง่ายต่อการต่อวงจรใช้งานกันเอง และยังประหยัดต้นทุน ในการสร้างบอร์ด ไปได้มาก ด้วยเหตุนี้เองที่ทำให้ “Arduino” ได้รับความนิยมจากผู้ใช้งานทั่วโลก เป็นอย่างมากในระยะเวลาอันรวดเร็ว Arduino มีจุดเด่นในเรื่องของความง่ายต่อการเรียนรู้และใช้งาน เนื่องจากมีการออกแบบคำสั่งต่างๆขึ้นมาสนับสนุนการใช้งาน ด้วยรูปแบบที่ง่ายไม่ซับซ้อน ในตลาดไมโครคอนโทรลเลอร์มีตัวเลือกมากมาย เช่น Parallax Basic Stamp, Netmedia's BX-24, Pidgets, MIT's Handyboard, และอีกหลายผู้ผลิตที่มีคุณสมบัติใกล้เคียงกัน คือทำให้โครงการใช้งานง่าย และเน้นการโปรแกรมไมโครคอนโทรลเลอร์เป็นหลัก Arduino ก็เช่นเดียวกันแต่มีข้อแตกต่างที่เห็นได้ชัดคือ

- มีราคาไม่แพง เนื่องจากมี Source Code และวงจรแจก สามารถต่อวงจรขึ้นมาใช้งานได้เอง
- โปรแกรมพัฒนา Arduino ทำงานได้กับ Windows, MacintoshOSX และ Linux ในขณะที่บอร์ดอื่นทำงานได้เฉพาะกับ Windows
- ใช้งานง่าย มีโปรแกรมพัฒนาที่ไม่ซับซ้อนและมีโปรแกรมพัฒนา Arduino ใช้งานง่ายสำหรับผู้เริ่มต้น และมีความสามารถครบความต้องการของนักพัฒนา
- เปิดเผยแพร่ และนำไปพัฒนาขาย Hardware ได้ Arduino ใช้ไมโครคอนโทรลเลอร์ของ Atmel วงจรของบอร์ดตีพิมพ์แบบเปิดเผยวงจรภายใต้ Creative Commons License สามารถนำไปดัดแปลงต่อขยายและเพิ่มประสิทธิภาพ เพื่อศึกษาการทำงานของได้ฟรี
- เปิดเผย Source Code และ นำไปพัฒนาต่อยอดได้ โปรแกรม Arduino ตีพิมพ์แบบเปิดเผย Source Code และสามารถเพิ่มเติมความสามารถผ่าน C++ library, ถ้าต้องการศึกษาให้ลึกซึ่งสามารถเข้าไปเล่น AVR- C ซึ่งเป็นต้นแบบของ Arduino, และสามารถเพิ่มเติม AVR -C โค้ดได้โดยตรง

Arduino เป็นบอร์ดไมโครคอนโทรลเลอร์โดยใช้ AVR ขนาดเล็ก เป็นตัวประมวลผลและสั่งงานเหมาะสำหรับนำไปใช้ในการศึกษาเรียนรู้ระบบไมโครคอนโทรลเลอร์ และนำไปประยุกต์ใช้งานเกี่ยวกับการควบคุมอุปกรณ์ Input/Output ต่างๆได้มากมาย ทั้งในแบบที่เป็นการทำงานตัวเดียวอิสระ หรือเชื่อมต่อสั่งงานร่วมกับอุปกรณ์อื่นๆ เช่น คอมพิวเตอร์ส่วนบุคคล ทั้งนี้ก็เนื่องมาจากว่า Arduino สนับสนุนการเชื่อมต่อกับอุปกรณ์ Input / Output ต่างๆได้มากมาย ทั้ง แบบ Digital และ Analog เช่น การรับค่าจากสวิตช์ หรืออุปกรณ์ตรวจจับ (Sensor) แบบต่างๆ รวมไปถึง การควบคุม

อุปกรณ์ Output ต่างๆ ตั้งแต่ LED, หลอดไฟ, มอเตอร์, รีเลย์ ฯลฯ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยระบบฮาร์ดแวร์ของ Arduino สามารถสร้างและประกอบขึ้นใช้งานได้เองในกรณีที่ผู้ใช้พ่มีความรู้ด้านอิเล็กทรอนิกส์ หรือสามารถซื้อแผงวงจรสำเร็จรูปที่มีการผลิตออกจำหน่ายกัน ในราคาที่ไม่แพง อีกทั้งยังเผยแพร่ Source Code และตัวอย่างต่างๆ ให้ผู้ใช้งานนำไปใช้งาน หรือพัฒนาดัดแปลงต่อยอดได้โดยไม่เสียค่าใช้จ่ายใดๆ

ส่วนภาษาในการเขียนโปรแกรมลงบน Arduino นั้น จะใช้ภาษา C++ ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) อื่นๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจาก ANSI-C เล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐานของ ANSI-C โดยตรง

ซึ่งหากที่ได้ทำการศึกษาค้นคว้าทดลองการใช้งานภาษาซีของ Arduino มาในระยะเวลาหนึ่ง จะพบว่าในความเป็นจริง Arduino นั้นไม่ใช่ C-Compiler โดยตรง แต่ Arduino จะมีลักษณะการทำงานเช่นเดียวกับ Text Editor คือเป็นฉากหน้าในการติดต่อสื่อสารกับผู้ใช้เท่านั้น ส่วนเบื้องหลัง Arduino จะไปเรียกใช้ตัวแปลภาษาซีและ Utility อื่นๆ ที่ใช้เป็นเครื่องมือพัฒนาโปรแกรมของไมโครคอนโทรลเลอร์ตระกูล AVR อีกทีหนึ่ง โดย Arduino จะเลือกใช้ C-Compiler ของ “GNU AVR GCC Toolchain” ร่วมกับ Library Function ของ “avr-libc” ส่วน Utility ที่ใช้ในการ Upload Code ให้กับ AVR นั้น ก็จะใช้ของ “AVRDude” ดังนั้น ผู้ที่เขียนภาษาซีของ AVR เป็นอยู่แล้ว และต้องการประยุกต์ใช้งาน Arduino ให้ได้ประสิทธิภาพการทำงานมากยิ่งขึ้นไปอีก ก็สามารถศึกษาข้อกำหนด และหน้าที่ในการใช้งาน Library และคำสั่งอื่นๆ ที่บรรจุไว้ใน Library ต่างๆ ทั้งจากของ “GNU AVR-GCC Toolchain” และ “avr-libc” เพิ่มเติมอีกเพิ่มใช้เป็นแนวทางในการปรับปรุงและประยุกต์ใช้งาน Arduino ในรูปแบบที่สลับซับซ้อนมากๆ ขึ้นไปได้อีก

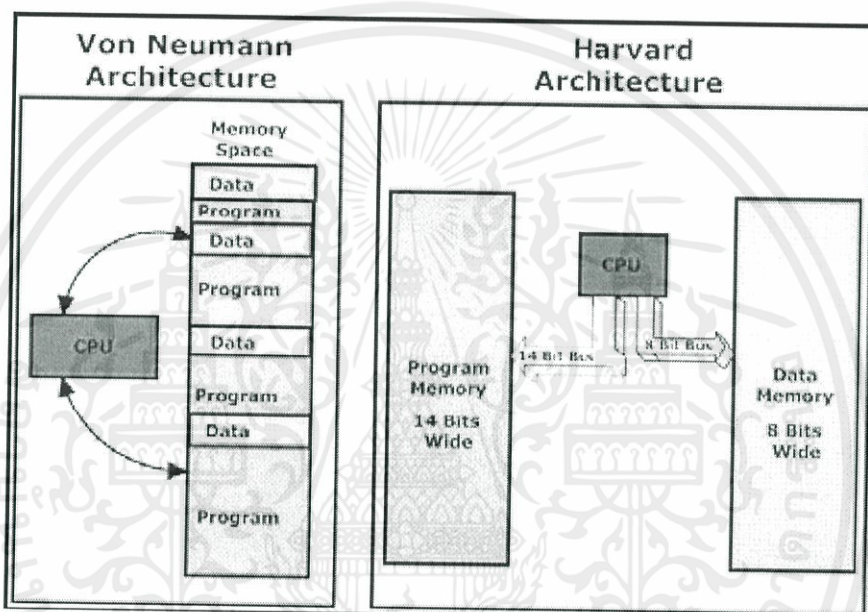
2.1.1 Chip และ IC ภายใน Arduino Board ที่สำคัญ

2.1.1.1 ATmega328P-PU



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้รูปที่ 2.2 ซีพียู ATMEGA328P ขนาด 28 ขา กสารทุกครั้งที่มีการนำไปใช้

ตัวบอร์ด Arduino ที่ใช้ในโครงการนี้จะกล่าวถึงสถาปัตยกรรมของ AVR ขนาด 8 บิต โดยในสถาปัตยกรรม AVR ซึ่งออกแบบโดย ATMEL เมื่อปี 1996 เป็นซีพียูแบบ RISC (Reduced Instruction Set Computer) มีสถาปัตยกรรมการต่อหน่วยความจำแบบ Harvard ซึ่งแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกันโดยเด็ดขาด ดังแสดงในรูปที่ 2.2 โดยใช้หน่วยความจำแบบ Flash สำหรับเป็นหน่วยความจำโปรแกรม และใช้หน่วยความจำแบบ SRAM สำหรับหน่วยความจำข้อมูล และนอกจากนี้ยังมีหน่วยความจำแบบ EEPROM ซึ่งสามารถเก็บข้อมูลเอาไว้ได้โดยไม่ต้องมีไฟเลี้ยง



รูปที่ 2.3 เปรียบเทียบการจัดการหน่วยความจำของสถาปัตยกรรม

จากรูปที่ 2.3 จะเห็นว่าโปรเซสเซอร์ที่ใช้สถาปัตยกรรมแบบ Harvard จะแยกหน่วยความจำ สำหรับเก็บข้อมูลออกจากโปรแกรมอย่างชัดเจน สถาปัตยกรรม AVR และ MCS-51 จะใช้รูปแบบนี้ในการจัดการหน่วยความจำ ส่วนสถาปัตยกรรมแบบ Von-Neumann การตัดสินใจว่าจะเก็บโปรแกรมหรือข้อมูลจะแบ่งเก็บอย่างไรจะทำได้อย่างอิสระ โดยขึ้น อยู่กับโปรแกรมเมอร์ หรือ อาจจะเป็นระบบปฏิบัติการเป็นผู้ดำเนินการให้ลักษณะเด่นของสถาปัตยกรรม AVR คือ คำสั่งส่วนใหญ่สามารถทำงานได้เสร็จภายใน 1 clock cycle ตัว ซีพียู AVR ขนาด 8 บิตจะแบ่งออกเป็นประเภทการใช้งานได้ 5 กลุ่ม ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- tinyAVR เป็นซีพียูในรุ่นเล็ก ซึ่งต้องการความถี่ที่ต่ำกว่าของวงจร โดยเหมาะกับระบบควบคุมขนาดเล็กๆ ที่ต้องการหน่วยความจำและวงจรสนับสนุนไม่มากนัก ซีพียูในรุ่นนี้จะมีราคาถูกลงกว่ากลุ่มอื่น
- megaAVR จะมีชื่ออีกอย่างว่า ATmega โดยมีวงจรสนับสนุนภายในเพิ่มเติมตลอดจนเพิ่มขนาดของหน่วยความจำให้ใช้งานมากกว่าตระกูล Tiny เหมาะกับงานควบคุมทั่วไป
- XMEGA เพิ่มความละเอียดของวงจร A/D จากปกติมีความละเอียด 10 บิตในรุ่นเล็กกว่าเป็น 12 บิต และวงจร DMA controller ซึ่งช่วยลดภาระของซีพียูในการควบคุมการรับส่งข้อมูลระหว่างอุปกรณ์ Input/Output กับหน่วยความจำ
- FPSLIC (AVR core with FPGA) สำหรับงานที่ต้องการควบคุมที่ต้องการความยืดหยุ่นขั้นตอนการออกแบบและพัฒนา โดยผู้ออกแบบสามารถออกแบบวงจรในระดับฮาร์ดแวร์เพิ่มเติมด้วยภาษาบรรยายฮาร์ดแวร์ (HDL: Hardware Description Language) เช่น ภาษา VHDL หรือ ภาษา Verilog และให้วงจรที่ออกแบบทำงานร่วมกับซีพียู AVR core
- Application Specific AVR เป็นซีพียูที่ออกแบบมาโดยเพิ่มวงจรควบคุมเฉพาะด้านเข้าไปซึ่งไม่พบในซีพียูกลุ่ม อื่นๆ เช่นวงจร USB controller หรือวงจร CAN bus เป็นต้น

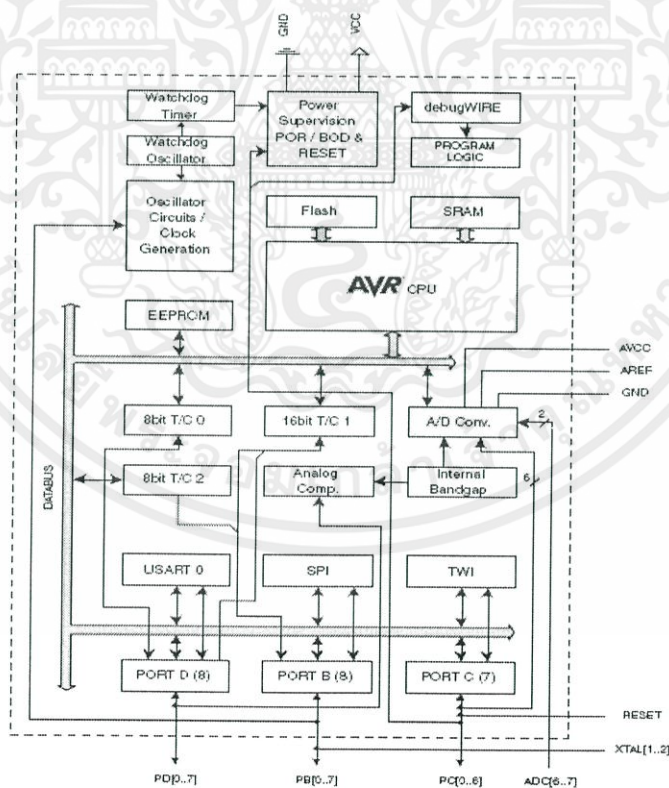
(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AINO) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

รูปที่ 2.4 แสดง pin ของ Arduino Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชิพ AVR มีให้เลือกใช้งานหลายเบอร์ แต่ละเบอร์จะมีขนาดราคา ความสามารถ และขนาดหน่วยความจำตลอดจนถึงวงจรสนับสนุนภายในที่แตกต่างกันออกไป ในโครงการนี้จะเลือกใช้ชิพรุ่น ATmega328P ซึ่งมีคุณสมบัติดังนี้

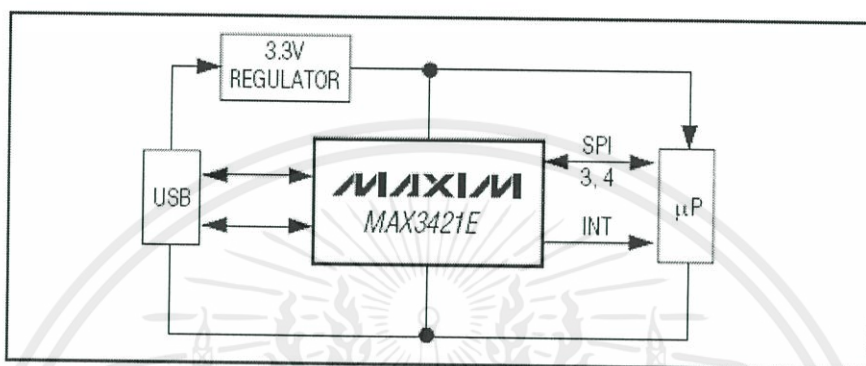
- หน่วยความจำโปรแกรมแบบ FLASH ขนาด 32 กิโลไบต์
- หน่วยความจำข้อมูลแบบ SRAM ขนาด 2 กิโลไบต์
- หน่วยความจำข้อมูลแบบ EEPROM ขนาด 1 กิโลไบต์
- สนับสนุนการเชื่อมต่อแบบ I2C bus
- พอร์ตอินพุตเอาต์พุตจำนวน 23 บิต
- วงจรสื่อสารอนุกรม
- วงจรนับ/จับเวลาขนาด 8 บิต จำนวน 2 ตัว และ u3586 ขนาด 16 บิตจำนวน 1 ตัว
- สนับสนุนช่องสัญญาณสำหรับสร้าง Pulse Width Modulation (PWM) จำนวน 6 ช่อง
- วงจรแปลงอนาลอกเป็นดิจิตอลขนาด 10 บิตในตัวจำนวน 8 ช่อง
- ทำงานได้ตั้งแต่ย่านแรงดัน 1.8-5.5 Volts
- ความถี่ใช้งานสูงสุด 20 MHz



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงรูปที่ 2.5 Block Diagram เข้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.2 MAX3421E

MAX3421E เป็น USB peripheral/host controller ที่มีดีจิจิตอลลอจิกและ วงจรอนาล็อกที่จำเป็นต่อ USB แบบ full-speed หรือ USB REV 2.0 ในตัว MAX3421E นี้จะมี transceiver 15 kV ESD อยู่ภายในเพื่อป้องกันและการ connect / disconnect ซึ่ง MAX3421E จะมีการใช้การเข้าถึงโดยมีอินเทอร์เฟซ SPI ที่ทำงานได้ถึง 26 Mhz



รูปที่ 2.6 แสดงการเชื่อมต่อ MAX3421E กับ Microprocessor ด้วย pin 3, 4

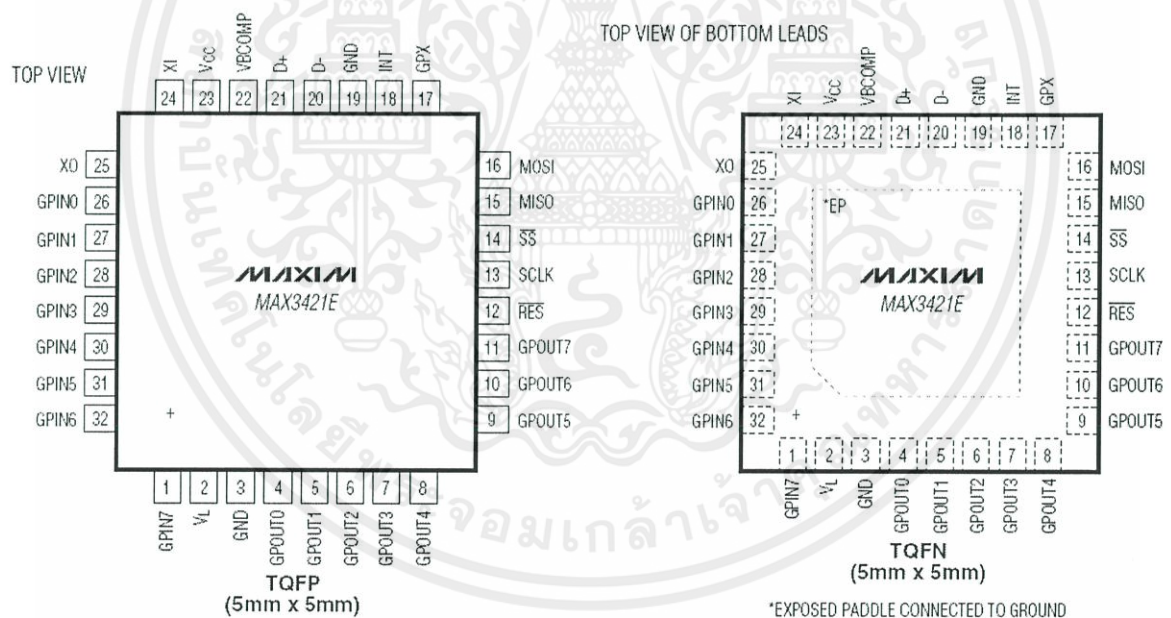
MAX3421E ทำให้การเชื่อมต่อ USB Peripheral นั้นกว้างขึ้น นั่นคือมันสามารถทำให้ใช้ได้กับหลายไมโครโพรเซสเซอร์เช่น ASIC, DSP เป็นต้น เมื่อมันทำงานเป็น USB Host เช่น ทำงานเป็นเมาส์หรือคีย์บอร์ดที่เชื่อมต่อกับระบบ Embedded Firmware จะสามารถทำได้ง่าย เพราะ Devices ได้รับการสนับสนุน และในตัว IC MAX3421E สามารถทำงานได้ถึงอุณหภูมิ -10°C ถึง $+85^{\circ}\text{C}$ ซึ่ง MAX3421E นั้น มีคุณสมบัติดังนี้

- Microprocessor-Independent USB Solution
- Software Compatible with the MAX3420E USB Peripheral Controller with SPI Interface
- Complies with USB Specification Revision 2.0 (Full-Speed 12Mbps Peripheral, Full-/Low-Speed 12Mbps/1.5Mbps Host)
- Integrated USB Transceiver
- Firmware/Hardware Control of an Internal D+ Pullup Resistor

(Peripheral Mode) and D+/D- Pulldown Resistors (Host Mode)

เอกสารนี้เป็นเอกสารของโปรแกรมที่พัฒนาขึ้นโดยบริษัท ไมโครอิเล็กทรอนิกส์ จำกัด ไม่สามารถนำข้อมูลไปใช้ประโยชน์ด้านการค้า
 Programable 3- or 4-Wire, 26MHz SPI Interface ไม่นานญาติให้นำไปใช้ประโยชน์ด้านการค้า
 Level Translators and VL Input Allow Independent System Interface Voltage ไปใช้

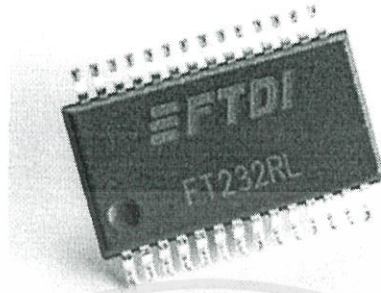
- Internal Comparator Detects VBUS for Self-Powered Peripheral Applications
- ESD Protection on D+, D-, and VBCOMP
- Interrupt Output Pin (Level- or Programmable-Edge) Allows Polled or Interrupt-Driven SPI Interface
- Eight General-Purpose Inputs and Eight General-Purpose Outputs
- Interrupt Signal for General-Purpose Input Pins, Programmable Edge Polarity
- Intelligent USB SIE
- Automatically Handles USB Flow Control and Double Buffering
- Handles Low-Level USB Signaling Details
- Contains Timers for USB Time-Sensitive Operations so SPI Master Does Not Need to Time Events
- Space-Saving Lead-Free TQFP and TQFN Packages (5mm x 5mm)



รูปที่ 2.7 Pin Configurations

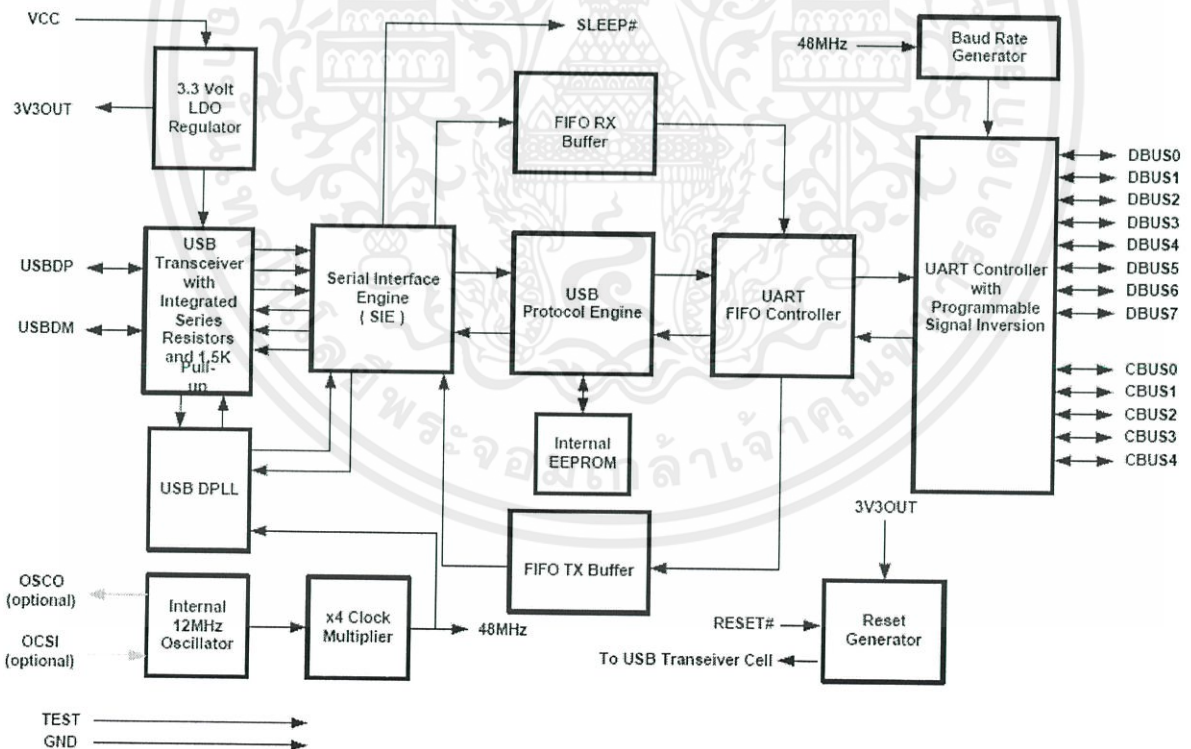
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.3 FT232RL

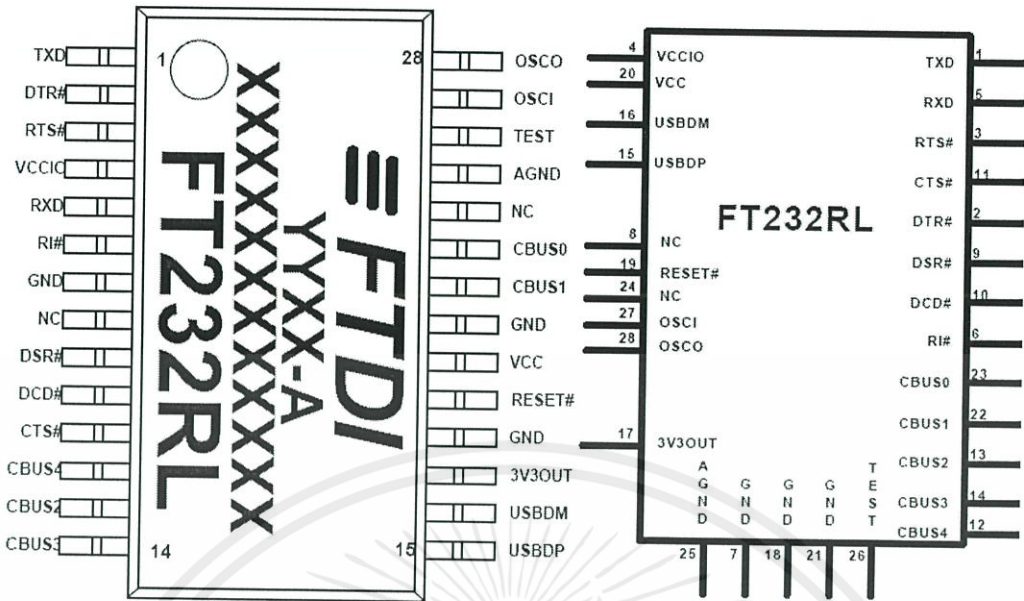


รูปที่ 2.8 IC FT232RL

ใน Arduino Board จะมี IC FT232RL ที่มีไว้สำหรับทำหน้าที่แปลงจาก USB เป็น Serial เพื่อนำไปติดต่อกับไมโครคอนโทรลเลอร์ โดยมี Block Diagram ดังนี้

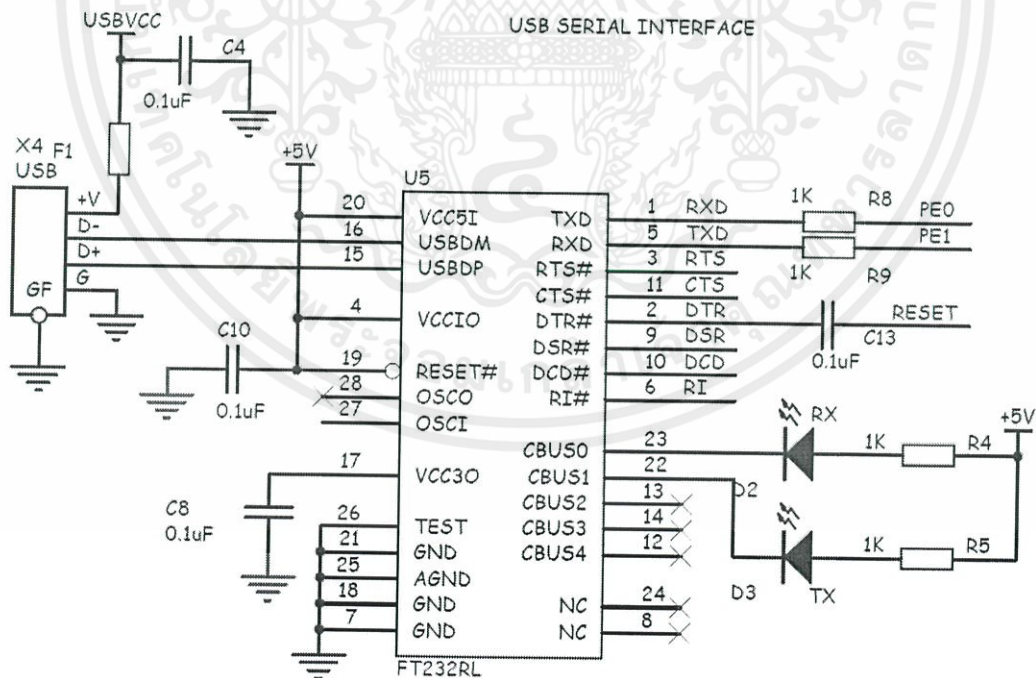


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 2.9 แสดง Block Diagram ของ FT232RL
 ไม่ว่าจะพิมพ์ที่ไหน ก็ต้องมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.10 แสดง Device Pin Out

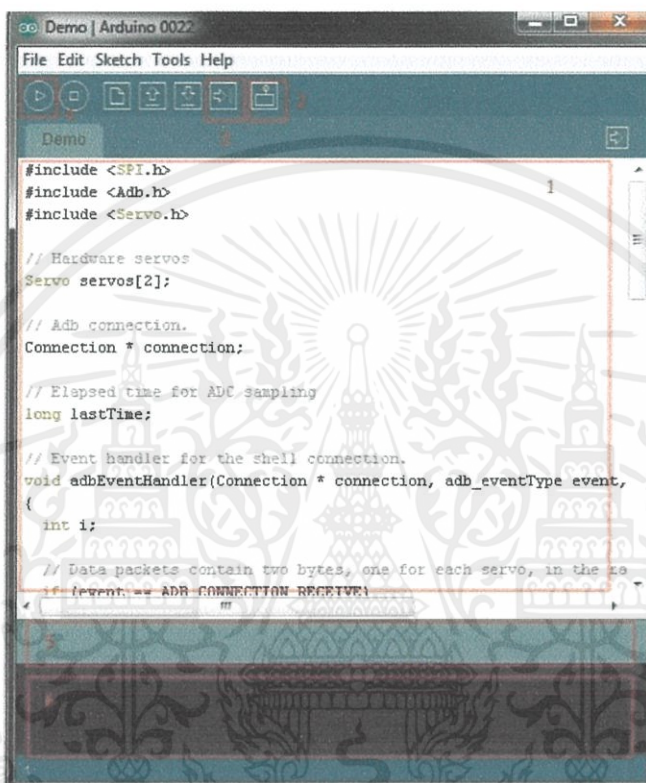
ตัวอย่างในการต่อกับ USB ในบอร์ด Arduino



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งรูปที่ 2.11 แสดง Schematics ของ FT232RL ต่อกับ USB ทุกครั้งที่มีการนำไปใช้

2.1.1 Software Arduino Environment

โปรแกรม Arduino Environment เป็นซอฟต์แวร์ที่พัฒนาขึ้นมาเพื่อการเขียนโปรแกรมลงบนบอร์ด Arduino สามารถรองรับได้ทั้ง ระบบปฏิบัติการ Windows, Linux และ Mac OS X



รูปที่ 2.12 แสดงหน้าต่างโปรแกรมพัฒนา Arduino Board

หน้าต่างของโปรแกรมจะมีส่วนประกอบดังนี้

หมายเลข 1 : ช่องแสดง Code Program

หมายเลข 2 : Serial Monitor

หมายเลข 3 : ปุ่ม Upload to Board

หมายเลข 4 : ปุ่ม Compiler

หมายเลข 5 : ช่องแสดง Status Program

หมายเลข 6 : ช่องแสดง Error

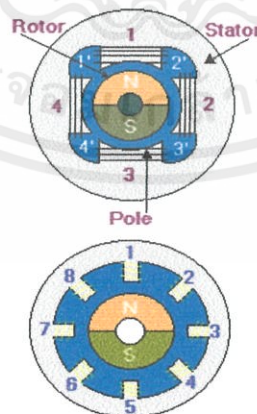
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 สเต็ปมอเตอร์ (STEP MOTOR)

สเต็ปมอเตอร์ เป็นมอเตอร์ที่ขับเคลื่อนด้วยพัลส์ ลักษณะขับเคลื่อนจะหมุนรอบแกนได้ 360 องศา มีลักษณะไม่ต่อเนื่องมีลักษณะเป็นสเต็ป โดยแต่ละสเต็ปจะขับเคลื่อนได้ 1, 1.5, 1.8, หรือ 2 องศา แล้วแต่ละโครงสร้างของมอเตอร์ เมื่อป้อนไฟฟ้าให้กับมอเตอร์ทำให้หมุนเพียงเล็กน้อย ตามเส้นรอบวงและหยุด ซึ่งต่างจากมอเตอร์ทั่วไปที่จะหมุนทันทีและตลอดเวลาเมื่อมีแรงดันไฟฟ้า ข้อดีของสเต็ปมอเตอร์สามารถกำหนดตำแหน่งการหมุนได้ด้วยตัวเลข (องศาหรือระยะทาง) ได้อย่างละเอียด โดยใช้คอมพิวเตอร์หรือไมโครคอนโทรลเลอร์เป็นเครื่องกำหนดและจัดเก็บตัวเลข ลักษณะการนำมอเตอร์ไปใช้จะเป็นงานที่ต้องการตำแหน่ง เช่น ระบบขับเคลื่อนหัวแม่พิมพ์ในเครื่องพิมพ์ หรือระบบขับเคลื่อนหัวอ่านบันทึกเทป ระบบขับเคลื่อนตำแหน่งปากกาใน X-Y PLOTTER



รูปที่ 2.13 สเต็ปมอเตอร์แบบ 5 เส้น และสเต็ปมอเตอร์แบบสาย 6 เส้น

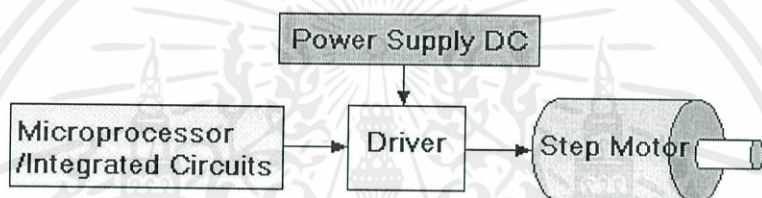


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามรูปที่ 2.14 แสดงรูปโครงสร้างภายในของสเต็ปมอเตอร์ทุกครั้งที่มีการนำไปใช้

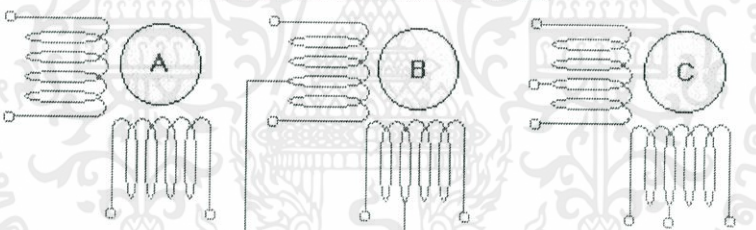
ในที่นี้ถ้าเพิ่มจำนวนของขั้วแม่เหล็กมากขึ้นจะเพิ่มจำนวนของสเต็ปต่อวงจรรอบมากขึ้นตามด้วย

ลักษณะการนำไปใช้งาน สเต็ปมอเตอร์ ใช้งานลักษณะระบบเปิด (Open Loop System) คือ สเต็ปมอเตอร์สามารถทำงานได้โดยไม่ต้องมีการ บ้อนค่าพารามิเตอร์กลับมา (Feed back) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งที่แน่นอนนั้น จะต้องการบ้อนกลับไปยังระบบและตัวบอก ตำแหน่งว่าถูกต้องหรือผิดพลาดให้รับทราบ

วิธีที่ใช้กับสเต็ปมอเตอร์ คือการนำลิมิตสวิทช์ ติดตามตำแหน่งที่จะตรวจจับ เมื่อสเต็ป มอเตอร์เริ่มหมุนแล้วหมุนไปจนถึงตำแหน่งของสวิทช์ตรวจจับสัญญาณ สวิทช์ทำงานก็จะบ้อนกลับไปสู่ระบบ ซึ่งก็จะทำให้รู้การทำงานของสเต็ปมอเตอร์ตลอด ตัววงจรไมโครคอนโทรลเลอร์เองจะมีจุดอ้างอิง ไว้ให้เริ่มต้นการทำงานและอ้างอิงตำแหน่งได้ถูกต้อง

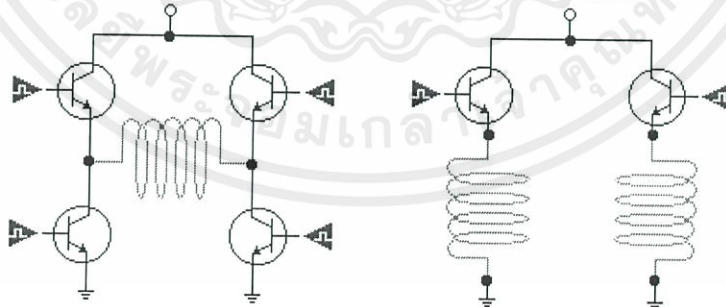


การควบคุมระบบสเต็ปมอเตอร์



A) แบบไบโพลาร์ B)แบบยูนิโพลาร์ 5 สาย C)แบบยูนิโพลาร์ชนิด 6 สาย

การพันขดลวดบนสเตเตอร์ของสเต็ปมอเตอร์



A) แบบไบโพลาร์ B) แบบยูนิโพลาร์

▶ คือ ต่อเข้ากับแหล่งจ่ายไฟหรือจากพอร์ตพีซีเพื่อทรานซิสเตอร์ให้ทำงาน

วงจรการจ่ายไฟให้กับสเต็ปมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับรูปที่ 2.15 การควบคุมระบบสเต็ปมอเตอร์ ไปว่าญาติให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเต็มมอเตอร์เป็นอุปกรณ์จำพวกเชิงกลทางไฟฟ้า โดยมีกลุ่มของไบนารีโวลต์เตทเป็นอินพุตและการเคลื่อนที่แบบเชิงมุมเป็นเอาต์พุต หรือว่าหมุนที่ละสเต็มซึ่งอยู่ระหว่าง 0.1 - 30 องศา ขึ้นอยู่กับโครงสร้างของสเต็มมอเตอร์ โดยตามสัญญาณพัลส์ที่จ่ายให้กับขดสเตเตอร์ทำให้เกิดแรงผลักแกโรเตอร์หมุน สเต็มมอเตอร์มีขดลวดหลายชุดเรียกว่า Phase (เฟส) ดังนั้นสัญญาณที่ต่อเนื่องเป็น Sequence (ซีควีน) ลักษณะของ Binary (ไบนารี) ซึ่งจะต้องไปผ่านวงจร Driver (ไดรเวอร์) ก็จะทำให้โรเตอร์หมุนไปอย่างต่อเนื่อง ตามรูปของการควบคุมสเต็มมอเตอร์

การพันขดลวดบนสเตเตอร์ของสเต็มมอเตอร์มีด้วยกัน 2 วิธี คือ แบบ Bipolar (ไบโพลาร์) กับแบบ Unipolar (ยูนิโพลาร์)

2.2.1 แบบ Bipolar

จะมีการพันขดลวดหนึ่งขด ในแต่ละขั้วแม่เหล็กของสเตเตอร์ โดยขั้วแม่เหล็กที่เกิดขึ้นที่สเตเตอร์จะถูกกำหนดโดยทิศทางของการไหลของกระแสไฟฟ้า ซึ่งสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงกันข้ามได้ เพียงการกลับทิศทางของการไหลในกระแสไฟฟ้า โดยมาจากการควบคุมของวงจรสวิตชิงให้กลับขั้วไฟฟ้า

2.2.2 แบบ Unipolar

แบบนี้มี 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ ทำให้แต่ละขดลวดเกิดขั้วแม่เหล็กในทิศทางตรงกันข้าม เช่นกันการกลับทิศทางขั้วแม่เหล็กทำได้โดยใช้วงจรสวิตชิงให้สลับหนึ่งไปยังอีกขั้วหนึ่งแทนกัน

พื้นฐานการสวิตชิง วงจรการจ่ายไฟให้กับสเต็มมอเตอร์ การพันต่างกันจะให้ประโยชน์คือ แบบยูนิโพลาร์จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพลาร์ โดยสายไฟที่ต่อมาจากตัวสเต็มมอเตอร์ซึ่งแบบไบโพลาร์จะมี 4 สาย ส่วนเป็นแบบยูนิโพลาร์จะมี 5 สายหรือ 6 สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 การสั่งงานควบคุมการหมุนของสเต็ปมอเตอร์

การควบคุมการหมุนของสเต็ปมอเตอร์แบบ Unipolar จะทำการควบคุมกระแสไฟที่จ่ายให้กับขดลวดในแต่ละเฟส อย่างเป็นลำดับที่แน่นอน หากให้กระแสไหลในเฟสใดๆ ก็จะทำให้สถานะของเฟสนั้นๆ เป็นสถานะลอจิก “1” การควบคุมการหมุนของสเต็ปมอเตอร์สามารถทำได้ 3 วิธีดังนี้

2.3.1 การควบคุมการหมุนแบบเฟสเดียว (Single Phase) หรือ Wave Drive

การป้อนกระแสไฟให้กับขดลวดของสเต็ปมอเตอร์ทีละขด โดยจะป้อนกระแสเรียงตามลำดับกันไปตามตาราง 2.1 (ตัวเลข 1 หมายถึง มีการจ่ายแรงดันให้กับ ขดลวด 0 คือไม่จ่ายแรงดันให้กับขดลวด) กระแสที่ไหลในขดลวดจะทำการไหลในทิศทางเดียวกันทุกขด ทำให้แรงขับของสเต็ปมอเตอร์มีน้อย ไม่เป็นที่นิยมใช้

ตาราง 2.1 ลำดับการป้อนกระแสแบบเฟสเดียว

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1
5	Back to Step 1			

2.3.2 การควบคุมการหมุนแบบสองเฟส (Two Phase) หรือ Full Step Drive

ป้อนกระแสไฟให้กับขดลวดของสเต็ปมอเตอร์ทั้ง 2 ขด พร้อมๆ กันไป และป้อนกระแสเรียงตามลำดับกันไป ตามตาราง 2.2 มีกระแสไหลในขดลวดของมอเตอร์มากขึ้น และทำให้มอเตอร์มีแรงบิดมากขึ้น

ตาราง 2.2 ลำดับการป้อนกระแสแบบสองเฟส

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	1	0	0
2	0	1	1	0
3	0	0	1	1
4	1	0	0	1
5	Back to Step 1			

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ซึ่งหากท่านมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายบริการลูกค้า หรือฝ่ายขายของเราได้ที่เบอร์โทรศัพท์ 02-111-1111 หรืออีเมล service@... ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การควบคุมการหมุนแบบครึ่งเฟส (Half Step)

การป้อนกระแสแบบ Two Phase และ Single Phase สลับการไป ตามตาราง 2.3 ซึ่งเป็นการเพิ่มความละเอียดของตำแหน่งในการหมุน

ตาราง 2.3 ลำดับการป้อนกระแสแบบครึ่งเฟส

Step	Phase A	Phase B	Phase A'	Phase B'
1	1	0	0	0
2	1	1	0	0
3	0	1	0	0
4	0	1	1	0
5	0	0	1	0
6	0	0	1	1
7	0	0	0	1
8	1	0	0	1
9	Back to Step 1			

ความเร็วการหมุนของแกนสเต็ปมอเตอร์ขึ้นอยู่กับระยะเวลา (Time Delay) ใน แต่ละลำดับ ถ้า Time Delay มีค่าน้อย แกนจะหมุนเร็วขึ้น ถ้า Time Delay มีค่ามาก แกนจะหมุนช้า และหมุนไม่สม่ำเสมอ

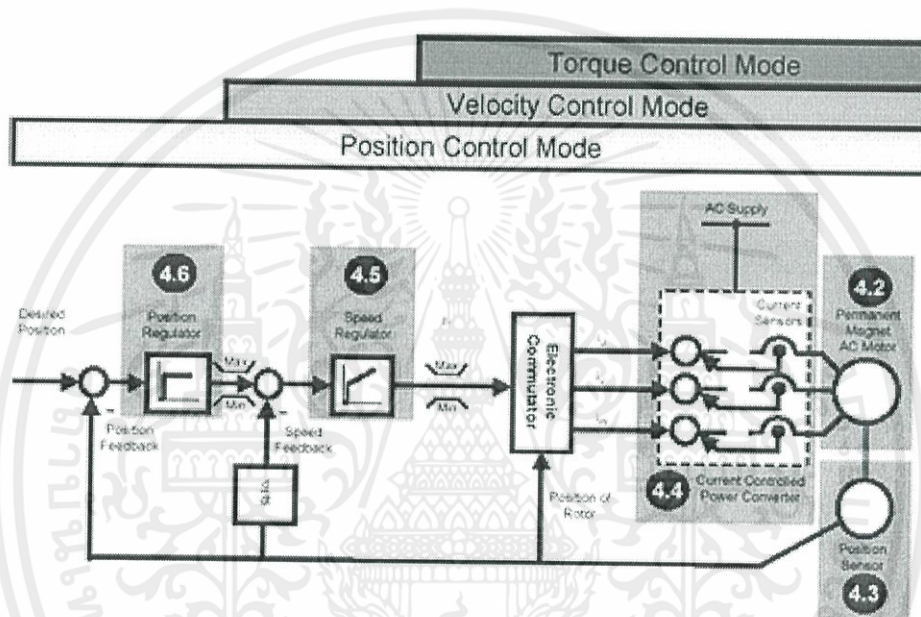
2.3.4 ข้อดีของสเต็ปมอเตอร์เมื่อเทียบกับมอเตอร์กระแสตรง

1. การควบคุมไม่ต้องอาศัยตัวตรวจจบการหมุน
2. ไม่ต้องใช้แปลงถ่าน ดังนั้นจึงไม่มีส่วนที่จะต้องสึกหรอและปัญหาของการสปาร์ค (ที่เกิดจากหน้าสัมผัสของแปลงถ่านแหวนตัวนำในโรเตอร์) ที่ทำให้เกิดสัญญาณรบกวน
3. การควบคุมโดยทางวงจรถิจริตอลหรือไมโครคอนโทรลเลอร์ทำงานได้ง่ายและสะดวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Servo motor

Servo motor คือ มอเตอร์ที่นำมาใช้ในระบบการควบคุมทางความเร็วและตำแหน่งร่วมกัน โดยระบบ Servo จะต้องตอบสนองต่อความเร็วและการเข้าถึงตำแหน่งหรือการเคลื่อนที่ไปที่ระยะเป้าหมายอย่างเหมาะสม ระบบ Servo จะมีการป้อนกลับของความเร็วและตำแหน่งกลับมาที่ส่วนควบคุมหรือเซอร์โวไดรฟ์แล้วแต่การออกแบบ การป้อนกลับความเร็วอาจใช้ Taco meter และการป้อนกลับตำแหน่งอาจใช้ encoder หรือ pulse generator แล้วแต่จะเรียกหรือบางที่จะใช้ resolver ก็ได้ซึ่ง encoder และ resolver ต่างก็สามารถนำมาใช้ป้อนกลับความเร็วได้



รูปที่ 2.16 แสดงรูปโครงสร้างระบบควบคุมเซอร์โว

2.4.1 โครงสร้างระบบควบคุมเซอร์โว

ลักษณะของระบบควบคุมเซอร์โวมอเตอร์จะเป็นระบบควบคุมแบบลูปปิด(Closed loop control) ซึ่งประกอบด้วย 3 โหมดการควบคุมคือ โหมดการควบคุมแรงบิด(Torque Control Mode) ซึ่งอยู่วงรอบหรือลูบในสุด โหมดการควบคุมอัตราเร่ง (Velocity Control Mode)

1. เซอร์โวมอเตอร์ (Servo Motor) (4.2)
2. ชุดควบคุมการขับเคลื่อนเซอร์โว

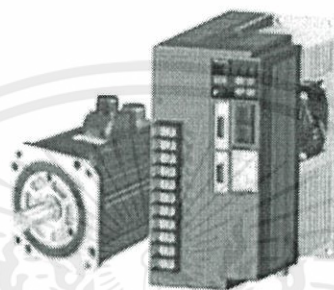
(Servo Drive, Servo Amplifier หรือบ้างก็เรียกว่า servo controller) (4.4, 4.5, 4.6)

3. อุปกรณ์ป้อนกลับ (Feedback Device เช่น Speed encoder และ Position Sensor) (4.3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอ้างอิงเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ระบบเซอร์โว

ระบบเซอร์โว คือ การควบคุมเครื่องจักรกลในที่นี้คือมอเตอร์ให้ทำงานตอบสนองด้านไดนามิกส์ เช่น ความเร็ว อัตราเร่ง แรงบิด และตำแหน่ง ให้ได้ดีที่สุด (Optimum Solution) และใช้เวลาน้อยที่สุด (Time Optimum) ดังนั้น ระบบเซอร์โวมอเตอร์จะประกอบด้วย ระบบไฟฟ้า คอนโทรลและเครื่องจักรกลที่ต้องทำงานสัมพันธ์กัน



รูปที่ 2.17 เซอร์โวไดรฟ์และมอเตอร์เซอร์โว

2.4.3 ประเภทของเซอร์โวมอเตอร์

โดยทั่วไปมีทั้งดีซีเซอร์โวและเอซีเซอร์โว ในเครื่องจักรรุ่นเก่าๆ พบว่า ดีซีเซอร์โวมอเตอร์ มีการใช้เครื่องจักรอุตสาหกรรมมากกว่า เอซีเซอร์โวมอเตอร์ เนื่องจากการควบคุมกระแสสูงๆ ต้องใช้ SCR ปัจจุบันทรานซิสเตอร์ได้พัฒนาขีดความสามารถให้ตัดต่อกระแสสูงและใช้งานที่ความถี่สูงขึ้นจึงทำให้ระบบควบคุมทางเอซีและระบบเซอร์โวได้ถูกนำมาใช้งานมากขึ้น ซึ่งสามารถแยกประเภทของเซอร์โวได้ดังนี้

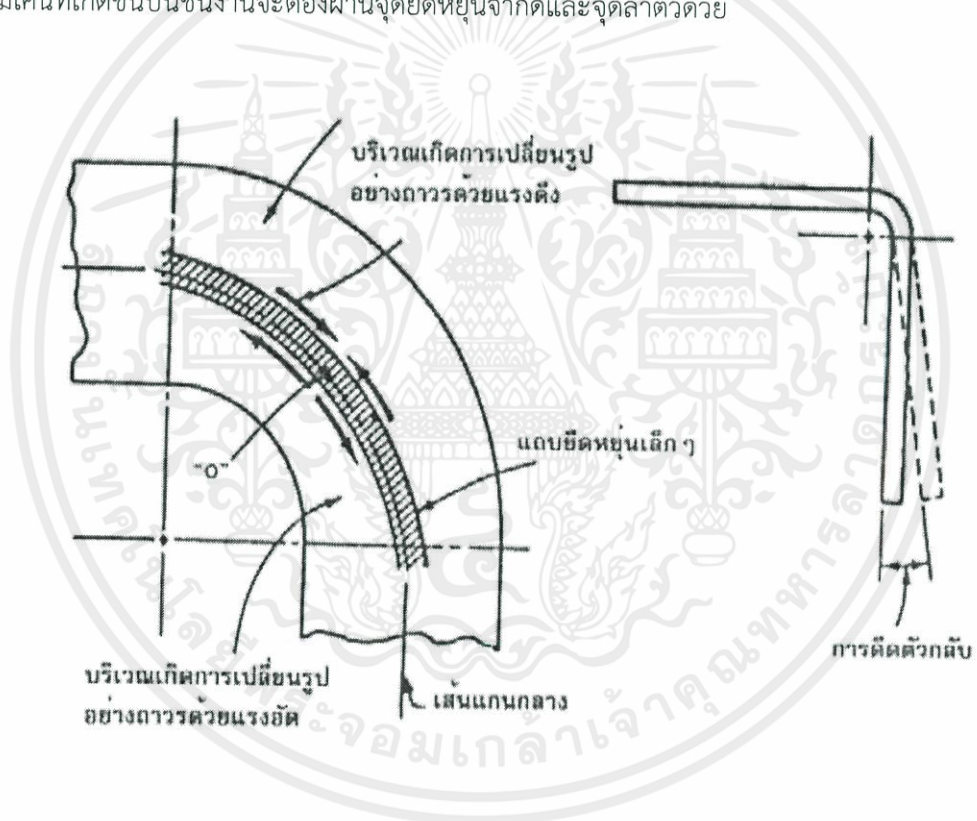
- มอเตอร์ชนิดที่มีแปรงถ่าน (Brush Type = mit Bürsten)
เซอร์โวชนิดนี้ สเตเตอร์จะเป็นแม่เหล็กถาวร ส่วนโรเตอร์ยังใช้แปรงถ่านและคอมมิวเตเตอร์เรียงกระแสเข้าสู่ขดลวดอาเมเจอร์ เหมือนกับดีซีมอเตอร์ทั่วไป
- เซอร์โวมอเตอร์ชนิดไม่มีแปรงถ่าน (Brushless Type = Ohne Bürsten)
 1. เซอร์โวมอเตอร์ในกลุ่มนี้ประกอบด้วยดีซีมอเตอร์
 2. เอซีมอเตอร์ ซึ่งมีทั้งแบบซิงโครนัสเซอร์โว
 3. อะซิงโครนัสเซอร์โว (การนำอินดักชั่นมอเตอร์มาใช้ทำเป็นระบบขับเคลื่อน)
 4. สเต็ปป์เซอร์โวมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ทฤษฎีการงอ (Bending Theory)

ในการขึ้นรูปโลหะโดยการงอขึ้นรูปนั้น ต้องให้แรงแก่ชิ้นงานทั้งนี้ เพื่อให้ชิ้นงานนั้นเปลี่ยนรูปร่างอย่างถาวร ซึ่งแรงที่ให้แก่ชิ้นงานนั้นจะต้องไม่ทำให้ความเค้นที่เกิดขึ้นบนชิ้นงานมากกว่าความเค้นดึงสูงสุดของชิ้นงานนั้น และจะต้องไม่น้อยกว่าจุดยืดหยุ่นจำกัดของชิ้นงานนั้น

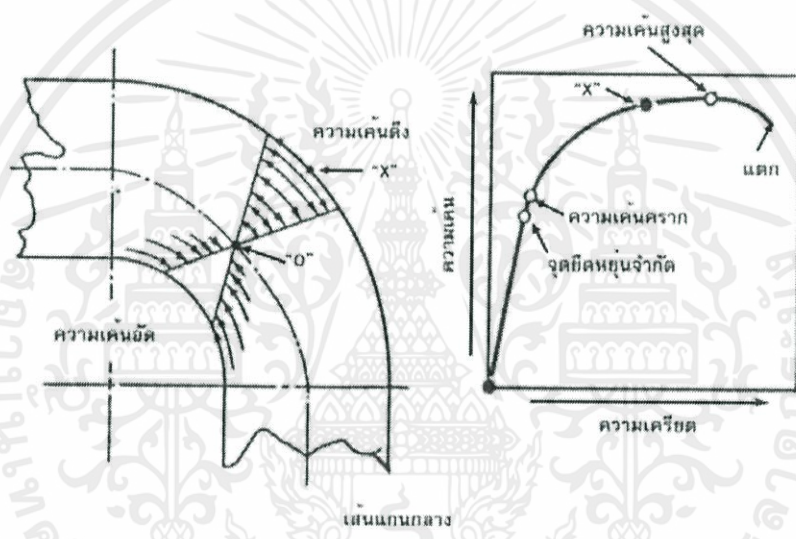
เมื่อให้แรงแก่ชิ้นงานเพื่อทำการงอจะปรากฏว่าความเค้นของชิ้นงานที่เกิดขึ้นจะเริ่มจากบางจุดที่ต่ำกว่าความแข็งแรงสูงสุดของโลหะนั้น และความเค้นที่เกิดขึ้นนั้นจะแผ่กระจายไปยังส่วนต่างๆของชิ้นงาน ซึ่งขณะที่แผ่กระจายนั้น ความเค้นที่เกิดขึ้นบนชิ้นงานก็จะค่อยๆ ลดลงด้วยจนถึงบริเวณที่ความเค้นที่เกิดขึ้นบนชิ้นงานเป็นศูนย์ ซึ่งลักษณะการเกิดความเค้นบนชิ้นงานแบบนี้ จะทำให้เกิดการบิดงอของชิ้นงานมากกว่าจะเกิดการฉีกขาด สำหรับการเปลี่ยนแปลงรูปร่างอย่างถาวรของโลหะนั้น ความเค้นที่เกิดขึ้นบนชิ้นงานจะต้องผ่านจุดยืดหยุ่นจำกัดและจุดล้าตัวด้วย



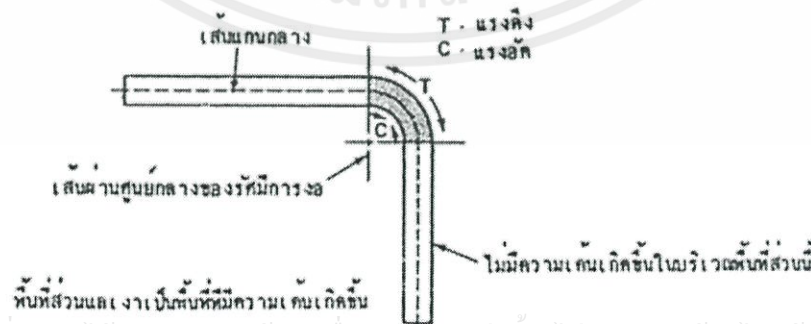
รูปที่ 2.18 การแสดงถึงการเปลี่ยนแปลงของความเค้นดึง และความเค้นอัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.18 เป็นการแสดงถึงการเปลี่ยนแปลงของความเค้นดึง และความเค้นอัดในบริเวณที่มีการดัด จะเห็นได้ว่าที่จุด "0" นั้นความเค้นดึงมีค่าเป็นศูนย์ ซึ่งจุดนี้จะอยู่บนเส้นแกนกลางของวัสดุ (Neutral Axis) บริเวณส่วนที่ใกล้กับเส้นแกนกลาง จะมีความเค้นเกิดขึ้นต่ำกว่าจุดยึดหยุ่นจำกัด (Elastic Limit) โลหะที่อยู่ห่างออกไปจากเส้นแกนกลางของชิ้นงานจะมีความเค้นเกิดขึ้นมากกว่าจุดความแข็งแรงครากตัว (Yield Point) ของชิ้นงาน ซึ่งบริเวณดังกล่าวจะเกิดการเปลี่ยนรูปอย่างถาวร (Plastic Deformation) เมื่อทำการดัดของชิ้นงานแล้ว ขณะดึงพ้นข้อออกส่วนที่อยู่ใกล้กับเส้นแกนกลางจะพยายามดึงตัวกลับเข้าสู่ตำแหน่งเดิม แต่ไม่สามารถดึงตัวกลับคืนได้หมด เพราะมีความต้านทานใน ส่วนที่มีการเปลี่ยนรูปอย่างถาวร ดังนั้นจึงเกิดการติดตัวกลับของชิ้นงานหลังการขึ้นรูป ดังแสดงในรูปที่ 2.19



รูปที่ 2.19 การเกิดการติดตัวกลับของชิ้นงานหลังการขึ้นรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและข้อมูลอ้างอิงใดๆของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.20 ความเค้นที่เกิดขึ้นในบริเวณพื้นที่ของการงอ

2.5.2 การไหลตัวของโลหะในการขึ้นรูป (Metal Flow)

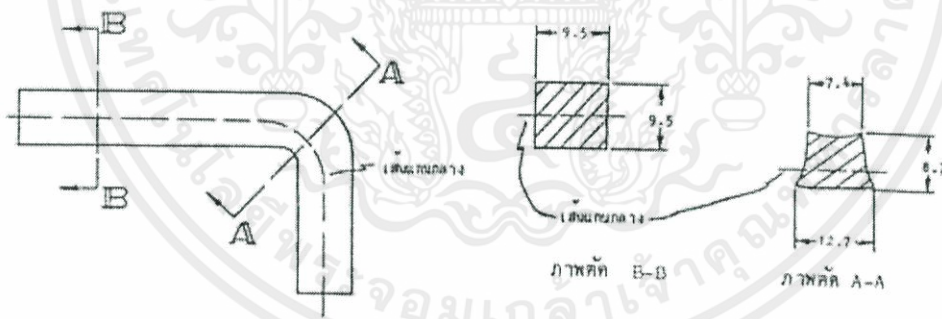
เพราะว่าโลหะบริเวณที่เรานั้นได้มีความเค้นเกิดขึ้นซึ่งมันจะทำให้เกิดการไหลตัวของโลหะ ความเค้นดึงที่ทำให้เกิดการไหลตัวของโลหะจะไปทำให้ความหนาของชิ้นงานในบริเวณที่มีขนาดลดลง ส่วนความเค้นอัดที่ทำให้เกิดการไหลตัวของโลหะจะไปเพิ่มขนาดความกว้างของชิ้นงานให้เพิ่มขึ้น ซึ่งได้แสดงไว้ในรูปที่ 2.21 จากรูปเป็นการแสดงถึงการบิดงอของแท่งโลหะเมื่อได้รับการงอ

2.5.2.1 ลักษณะของการไหลตัวของโลหะระหว่างการขึ้นรูปโดยการดัดงอ

1. ถ้าแท่งโลหะได้ถูกนำมางอจะทำให้เกิดการบิดงอทางพื้นที่หน้าตัดอย่างมากและจะมีการไหลตัวของโลหะทำให้ความหนาของแท่งโลหะลดลงเล็กน้อย แต่จะไปเพิ่มทางด้านกว้างมากขึ้น

2. ถ้าโลหะแผ่นได้ถูกนำมางอจะมีการบิดงอทางพื้นที่หน้าตัดเล็กน้อย แต่ความหนาจะลดลงอย่างมากและไม่มีการเพิ่มตัวขึ้นทางความกว้างเลย

เมื่อเรารองโลหะโอกาสที่จะเกิดการฉีกขาดมีมาก ทั้งนี้เพราะว่าการมีอัตราความกว้างมากต่อความหนาน้อย ส่วนรอยย่นจะปรากฏขึ้นเสมอเมื่อเรารองโลหะแผ่นหรือแท่งโลหะอย่างแรง ทั้งนี้เพราะว่าอัตราความกว้างน้อยต่อความหนามากจะเป็นสาเหตุทำให้เกิดการเพิ่มทางด้านความอึดตัวมากขึ้น



รูปที่ 2.22 การไหลของโลหะและการบิดงอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.2.2 ขนาดของการไหลตัวของโลหะซึ่งมีความสัมพันธ์กับสิ่งต่างๆ คือ

1. เมื่อรัศมีของการงอ (bend of radius) ยาวขึ้นจะทำให้เกิดการไหลตัวของโลหะน้อย
2. โลหะที่บางกว่าจะเกิดการไหลตัวของโลหะน้อย
3. มุมของการงอ (degree of bend) แคบจะทำให้เกิดการไหลตัวของโลหะน้อย
4. ถ้าเป็นการงอช่วงแคบ (narrow bend) จะทำให้มีการไหลตัวของโลหะเนื่องจากแรงดึงมีน้อย แต่การไหลตัวของโลหะเนื่องจากแรงอัดมีมาก
5. ถ้าเป็นการงอช่วงกว้าง (wide bend) จะทำให้มีการไหลตัวของโลหะเนื่องจากแรงดึงมีมาก แต่การไหลตัวของโลหะเนื่องจากแรงอัดมีน้อย
6. ถ้าเป็นโลหะแข็งจะมีการไหลตัวของโลหะทางด้านการดึงตัวมาก และทางด้านนี้ก็มีโอกาสที่จะฉีกขาดได้ง่าย
7. ถ้าเป็นโลหะที่อ่อนจะทำให้มีการไหลตัวของโลหะทางด้านการอัดตัวมาก และทางด้านนี้ก็มีโอกาสที่จะเกิดรอยยุบได้ง่าย

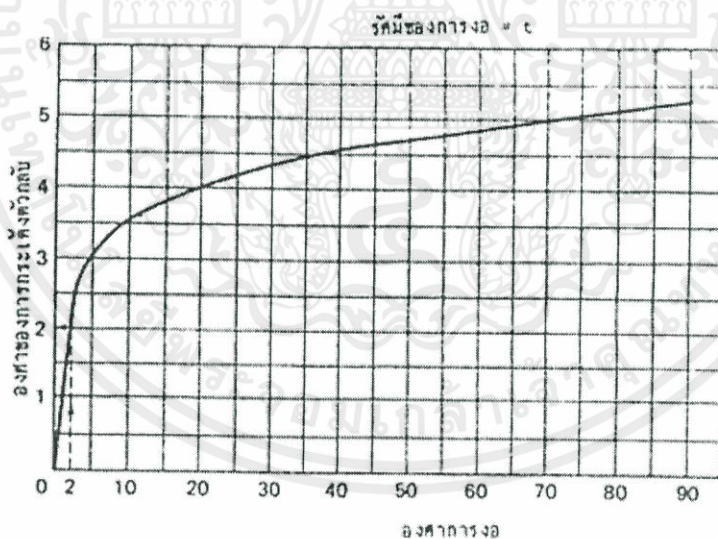
2.5.3 การกระด้างตัวกลับของชิ้นงาน (Spring back)

เมื่อได้ทำการงอชิ้นงานแล้วจะมีความเค้นเกิดขึ้นในบริเวณที่ทำการงอ และความเค้นที่เกิดขึ้นนี้จะมีความแตกต่างกันคือไม่เหมือนกันทุกจุดซึ่งจะเป็นผลทำให้เกิดการกระด้างตัวกลับของชิ้นงาน ความเค้นดึงจะเกิดขึ้นมากที่สุดที่ผิวหน้าด้านนอกของการงอและจะค่อยๆ ลดลงเรื่อยๆ เมื่อระยะของชิ้นงานเข้าใกล้เส้นศูนย์กลางของความหนา ซึ่งความเค้นที่เกิดขึ้นนี้จะกลายเป็นศูนย์ที่เส้นแกนกลาง จากรูปที่ 2.17 เป็นการแสดงถึงการเปลี่ยนแปลงของความเค้นดึงและความเค้นอัดในบริเวณที่มีการงอจากรูปจะเห็นได้ว่าที่จุด 0 นั้น ความเค้นดึงจะมีค่าเป็นศูนย์ซึ่งจุดนี้จะอยู่บนเส้นแกนกลาง และค่าความเค้นจะเพิ่มขึ้นเรื่อยๆ จนกระทั่งถึงจุด x ที่ผิวด้านนอกของชิ้นงานที่จุด x นี้จะมีค่าความเค้นดึงเกิดมากที่สุด

จากรูปที่ 2.17 เราจะเห็นได้ว่าบริเวณส่วนที่ใกล้กับเส้นแกนกลางนั้นจะมีความเค้นที่เกิดขึ้นต่ำกว่าจุดยึดหยุ่นจำกัด สำหรับโลหะส่วนนี้นั้นได้ถูกแสดงด้วยแถบเล็กๆ ที่เรียกว่าแถบยึดหยุ่น (elastic band) แถบเล็กๆ นี้จะอยู่ทั้งสองข้างของเส้นแกนกลาง โลหะที่อยู่ห่างไปจากแถบกลางของชิ้นงานจะมีความเค้นเกิดขึ้นมากกว่าจุดความแข็งแรงล้าตัวของชิ้นงานนั้น และที่บริเวณนี้จะมีการเปลี่ยนแปลงของโลหะอย่างถาวร เมื่อได้ทำการงอชิ้นงานแล้วและขณะที่ดึงเอาฟันช้อออกไปนั้น

ส่วนที่เป็นแถบยืดหยุ่นเล็กๆ ก็พยายามดึงตัวกลับเข้าอยู่ในตำแหน่งเดิม แต่ก็ไม่สามารถจะดึงตัวกลับคืนได้หมด เพราะมีความต้านทานของส่วนที่มีการเปลี่ยนแปลงของโลหะอย่างถาวร ดังนั้นจึงมีการกระด้างกลับเพียงเล็กน้อยเท่านั้น การที่โลหะกระด้างตัวกลับนี้เรียกว่า “ Spring back ” สาเหตุที่โลหะต้องกระด้างกลับก็เพราะว่าต้องการให้อยู่ในสภาวะสมดุลนั่นเอง สำหรับส่วนที่เป็นแถบยืดหยุ่นเล็กๆ นั้นจะเป็นตัวทำให้เกิดแรงของการกระด้างกลับรอบจุด 0 ดังที่แสดงไว้ในรูป ความจริงแล้วในช่วงของการเปลี่ยนแปลงอย่างถาวรของโลหะ (plastic deformation) ก็ยังคงมีส่วนที่เป็นความยืดหยุ่นปนอยู่เล็กน้อยซึ่งส่วนนี้จะไปเพิ่มแรงในการกระด้างตัวกลับให้มากขึ้น

สำหรับรูปที่ 2.22 เราใช้เหล็กกล้าผสมคาร์บอนต่ำเป็นตัวอย่างในการทดลอง จากรูปเราจะเห็นได้ว่าเส้นกราฟที่เกิดขึ้นนั้นคล้ายกับเส้นกราฟแสดงการทดสอบความเค้น-ความเครียดทางดึง เมื่อเราทำการงอชิ้นงานด้วยมุมของการงอ 2 องศา จะปรากฏว่าชิ้นงานจะกระด้างตัวกลับเป็นมุม 2 องศาเช่นเดียวกัน คือมันจะเป็นแผ่นโลหะแบนๆ เช่นเดิม แต่เมื่อเราทำการงอชิ้นงานเป็นมุม 20 องศา การเกิดการกระด้างตัวกลับจะเป็นมุมมากกว่า 4 องศา ทั้งนี้เพราะว่าในพื้นที่ที่ทำการงอนั้นได้เกิดช่วงของการอยู่ตัวเกิดขึ้น



รูปที่ 2.23 การกระด้างตัวกลับของโลหะ SAE 1008 Aluminium-Killed Cold

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3.1 วิธีป้องกันการเกิดการกระเด็นตัวกลับ (Overcoming springback)

1. การงอให้มากกว่าความต้องการ (Overbending)
2. การกดกระแทกชิ้นงานตรงบริเวณที่งอให้เป็นรอย (Bottoming or setting)
3. การยืดชิ้นงานก่อนงอ (Stretch bending)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการสร้าง

3.1 เครื่องตัดลวด

การออกแบบสร้างเครื่องตัดลวด จะต้องทำการคำนวณ หาค่าที่เหมาะสมในการสร้าง เช่น หาขนาดแรงบิดของมอเตอร์ ขนาดสัดส่วนที่จะใช้สร้างจริง ว่ามีความเหมาะสมตรงกับ สิ่งที่จะทำการตัดหรือไม่ ซึ่งจะต้องวิเคราะห์ เพื่อให้สอดคล้องกับขนาดที่จะทำการตัด ตามสัดส่วนรูปร่างต่างๆ

3.1.1 ชนิดของลวดที่ใช้ตัด

ลวดที่ใช้ตัด ขนาด 2.60 มม. ชนิดลวด Aluminum 1350-O เนื่องจากเป็นลวดที่ค่า Ultimate Tensile Strength ต่ำ สามารถทำให้เปลี่ยนรูปได้โดยใช้แรงน้อยที่สุด

ตารางที่ 3.1 แสดงค่าเฉพาะของลวด Aluminum 1350-O

Mechanical Properties	Metric	Comments
Hardness, Brinell	23	500 kg load with 10 mm ball. Calculated value.
Tensile Strength, Ultimate	55.0 - 95.0 MPa	Sheet and plate
	58.0 - 95.0 MPa	Redraw rod
	82.7 MPa	AA; Typical
	58.0 - 95.0 MPa	Wire
	@Thickness <=9.50 mm	
Tensile Strength, Yield	27.6 MPa	AA; Typical

เอกสารนี้เป็นที่ปรึกษาการใช้งานที่ปรึกษาเท่านั้น ไม่อนุญาตให้ Sheet and plate ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อ >= 23 % อย่างอิงถึงเจ้าของ AA; Wire in 10 inches. ไปใช้

ตารางที่ 3.1 (ต่อ) แสดงค่าเฉพาะของลวด Aluminum 1350-O

Modulus of Elasticity 68.9 GPa AA; Typical; Average of tension and compression. Compression modulus is about 2% greater than tensile modulus.

Poisson's Ratio 0.33
 Shear Modulus 26.0 GPa Calculated
 Shear Strength 55.2 MPa AA; Typical

3.1.2 รัศมีที่ใช้ตัด

วัสดุ	ขนาดลวด (mm)	รัศมีตัดลวด (mm)					
		0.8	1	1.5	2	3	4
EN AW-1090	0.8-5	1.2	1.4	1.6	2.5	4	5
EN AW-1050	0.8-5	1	1.2	1.5	2	3	4
EN AW-1050A	0.8-5	1	1.2	1.5	2	3	4
EN AW-1050A-O	0.8-5	1	1.2	1.5	2	3	4
EN AW-1050A-S	0.8-5	1	1.2	1.5	2	3	4
EN AW-1050A-T5	0.8-5	1	1.2	1.5	2	3	4
EN AW-1050A-SMg1-T5	0.8-5	1	1.2	1.5	2	3	4

รูปที่ 3.1 แสดงรัศมีตัดที่ยอมรับ สำหรับวัสดุชิ้นงานที่ไม่ใช่เหล็ก สำหรับโลหะแผ่นหนาต่างกัน

รัศมีตัดเล็กสุดที่สามารถตัดได้ของโลหะแผ่น อลูมิเนียมที่ใช้ (AW – Al99.5) คือ 2.5 mm แต่จะเลือกใช้ที่ 3 mm เพื่อเป็นระยะเผื่อและป้องกันการแตกของเนื้อวัสดุ

3.1.3 ค่ามุมเผื่อสำหรับการสปริงตัวกลับของลวด

เอกสารนี้เป็นเอกสารที่การหามุมเผื่อสำหรับการสปริงตัวกลับของลวด จากหนังสือตารางคู่มือโลหะ เรื่องการดำเนินการค้า ไม่ว่าจะตัดขึ้นรูป เพื่อชดเชยในการตัดให้ได้มุมตามที่ต้องการอย่างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสัมพันธ์

$$\text{รัศมีที่เครื่องมือ} \quad r_1 = k_R (r_2 + 0.5 \times s) - 0.5 \times s \quad (3.1)$$

$$\text{มุมที่เครื่องมือ} \quad \alpha_1 = \frac{\alpha_2}{k_R} \quad (3.2)$$

เมื่อ $\alpha_1 =$ มุมที่เครื่องมือก่อนสปริงกลับ

$\alpha_2 =$ มุมตัดที่ชิ้นงาน (180°)

$r_1 =$ รัศมีที่เครื่องมือ

$r_2 =$ รัศมีตัดที่ชิ้นงาน (2.5 mm)

$k_R =$ แฟคเตอร์สปริงกลับ (EN AW-Al99,0 (at $r_2 : s = 1$) = 0.99)

$s =$ ความหนาของแผ่นโลหะ (2.6 mm)

แทนค่า (3.1) และ (3.2)

$$\begin{aligned} \text{รัศมีที่เครื่องมือ} \quad r_1 &= 0.99(2.5 \text{ mm} + 0.5 \times 2.6 \text{ mm}) - 0.5 \times 2.6 \text{ mm} \\ &= 2.461 \text{ mm} \end{aligned}$$

$$\text{มุมที่เครื่องมือ} \quad \alpha_1 = 180^\circ / .99$$

$$\alpha_1 = 1.82^\circ$$

ได้ว่าหากต้องการรัศมีชิ้นงาน 2.5 mm รัศมีเครื่องมือตัดต้องเป็น 2.461 mm ต้องการมุมตัด 180° ต้องตัดเป็นมุม 181.82° จึงต้องออกแบบให้เครื่องสามารถตัดได้ 185° เพื่อเป็นระยะเผื่อในการสปริงตัวกลับของลวด

3.1.4 ขนาดของ Pin ตัด

หาพื้นที่หน้าตัดของลวด ความยาวของเส้นผ่านศูนย์กลางลวด 2.6 mm

Cross Section Area ของลวด

$$A = \pi \times r^2 \quad (3.3)$$

$$\text{แทนค่า (3.3)} \quad = \pi \times 2.6^2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงหรือทำซ้ำโดยไม่ได้รับอนุญาตของเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 แรงที่ใช้ในการตัดลวด

แรงที่สามารถทำให้ลวดเสียรูปถาวรได้ คือแรงที่มากกว่า Tensile Strength, Yield ของลวด ที่ขนาดพื้นที่หน้าตัดหนึ่ง

ตามความสัมพันธ์

$$F = P \times A \quad (3.4)$$

เมื่อ $P =$ Tensile Strength, Yield ของลวดอะลูมิเนียมที่ใช้

$A =$ cross section area ของลวด

$$\begin{aligned} \text{แทนค่า (3.4)} \quad F &= 27.6 \text{ MPa} \times 5.31 \text{ mm}^2 \\ &= 146.5 \text{ N} \end{aligned}$$

ดังนั้น ใช้แรงในการตัดลวดที่ 146.5 N

3.1.6 แรงจากความเสียดทาน

Pin ต้องรับแรงเสียดทานของ Pin กับลวด ซึ่งประสิทธิ์ความเสียดทาน (μ) ระหว่าง Mild Steel กับ Aluminum ซึ่งมีค่า $\mu = 0.61$ แรงในแนวตั้งฉากกับผิวสัมผัส (normal force, N) หาได้จากแรงกิริยาที่ไปกระทำกับลวด (146.5)

จากแรงจากความเสียดทาน

$$F = \mu \times N \quad (3.5)$$

$$\begin{aligned} \text{แทนค่า (3.5)} \quad &= 0.61 \times (146.5) \\ &= 89.4 \text{ N} \end{aligned}$$

3.1.7 แรงที่กระทำกับ Pin

Pin ต้องสามารถรับแรงกด และ แรงจากความเสียดทาน คือ $146.5 + 89.4 = 236 \text{ N}$

ดังนั้นในการตัดมีแรงมากกระทำกับ Pin ทั้งหมด 236 N

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.8 ความเค้นดัดที่เกิดขึ้นกับ Pin

ความเค้นดัด σ_b ที่เกิดขึ้นกับ Pin เมื่อมีการดัด สามารถคำนวณได้จาก

$$\sigma_b = \frac{M_b}{W} \quad (3.6)$$

เมื่อ M_b = moment ดัด

W = section moment area

ซึ่งสำหรับหน้าตัดวงกลม $W = \frac{\pi d^3}{32}$ (3.7)

แทนค่า (3.7)

$$= \frac{\pi 2.6^3}{32}$$

$$= 1.73 \text{ mm}^2$$

$$M_b = F \times l \quad (3.8)$$

เมื่อ l = เป็นความสูงของจุดรับแรงจากฐาน

F = เป็นแรงที่กระทำกับ Pin (236 N)

แทนค่า (3.8)

$$M_b = 235.9 \times 3$$

$$= 708 \text{ N.mm}$$

ดังนั้น

$$\sigma_b = \frac{M_b}{W}$$

$$= \frac{708}{1.73}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกนัยหนึ่งเป็นการเผยแพร่ข้อมูลของทางบริษัทเข้าสู่ของเอกสารทุกครั้งที่มีการนำไปใช้
ดังนั้น Pin ต้องสามารถรับแรงดัดได้ 818.5 MPa

3.1.9 ขนาดของ Pin ที่เหมาะสม

เพลาลูกโดยทั่วไป ทำมาจาก Mild Steel หรือเหล็กกล้าละมุน ซึ่ง Mild steel 1090 มีค่า Yield Strength 248 MPa เหล็กที่นำมาทำเป็นตัว Pin จึงต้องมีพื้นที่หน้าตัดอย่างน้อยให้มากพอที่จะสามารถทำให้ลวดถูกตัดด้วย Pin ตัดไม่เสียหาย เมื่อเทียบค่า strength ของแรงที่ต้องการกับ Yield Strength ของ Pin ที่ต้องการ

$$\text{จะได้ว่า} \quad 818.5 / 248 = 3.3$$

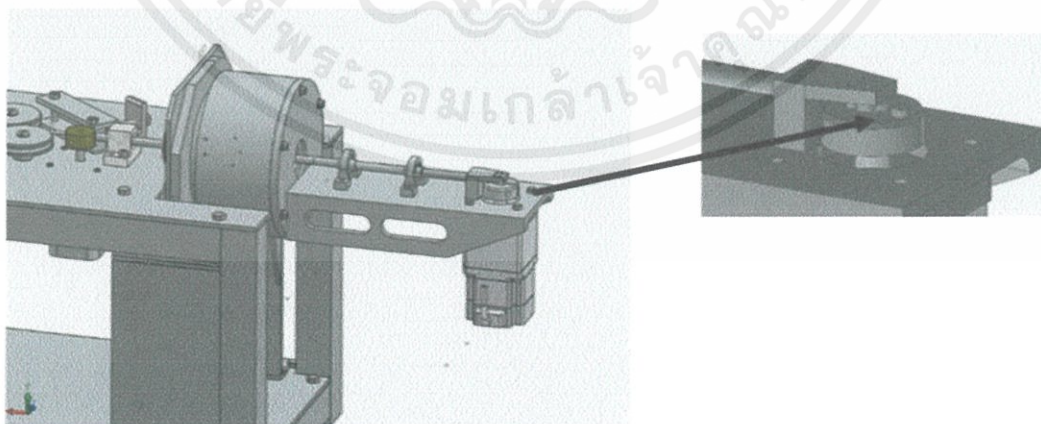
ซึ่งหมายความว่า Pin ตัด ต้องมีพื้นที่เป็น 3.3 เท่าของลวด ($5.31 \times 3.3 = 17.523 \text{ mm}^2$) เพื่อจะสามารถตัดลวด

$$\text{จากสมการการหาพื้นที่วงกลม} \quad A = \pi r^2 \quad (3.10)$$

$$r = \sqrt{\frac{A}{\pi}} \quad (3.11)$$

$$\begin{aligned} \text{แทนค่า (3.11)} \quad &= \sqrt{\frac{17.523}{\pi}} \\ &= 2.362 \text{ mm} \end{aligned}$$

ซึ่งหมายความว่า pin ต้องมีรัศมีอย่างน้อย 2.362 จึงจะเพียงพอต่อการตัด ตามมาตรฐานของเพลที่สามารถหาได้ จึงเลือกเพลให้มีขนาด $d = 6 \text{ mm}$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ถูกแปลงเนื้อหา และต้องแจ้งถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
รูปที่ 3.2 แสดงแกนตัด และตัวหัวตัดรวมถึง Pin ตัด

3.1.10 จำนวน Roller ที่ต้องใช้

ตารางที่ 3.2 แสดงจำนวนของ rolls ที่ใช้ตัด

Range of initial curvature radius Δr [mm]	Yield point R_p [MPa]	Number of straightening rolls n [-]
0	1000	5
60	1000	7
100	1000	9
100	2000	11
160	2000	13

จากตารางจึงได้ว่า ใช้ roller 5 ตัว

3.1.11 ขนาดของมอเตอร์ส่วนต่างๆ

3.1.11.1 ขนาดของมอเตอร์ที่จะทำการ feed

ตามความสัมพันธ์

$$F = C_{rr} \times N \quad (3.12)$$

เมื่อ C_{rr} = คัดของ Ball Bearing ซึ่ง มีค่า 0.001 - 0.0015 จึงนำค่า 0.0015 มาคิด

N = ค่า yield กับพื้นที่หน้าตัดของลวด ซึ่งเป็นแรงที่ roller กระทำกับลวดเพื่อให้

เกิดการตัด

แทนค่า (3.12)

$$N = 27.6 \times \pi (1.3)^2$$

$$= 146.54 \text{ N}$$

ได้ว่า

$$F = C_{rr} \times N$$

$$= 0.0015 \times 146.54 = 0.22 \text{ N}$$

เนื่องจากต้องใช้ roller ทั้งหมด 5 ตัวในการตัดตรง Roller 1 ตัว ก่อให้เกิดแรงเสียดทาน 0.22 N แรงเสียดทานที่เกิดจากการตัดตรงทั้งหมดคือ $0.22 \times 5 = 1.1 \text{ N}$

3.1.10.1 มอเตอร์สำหรับการตัด

เมื่อแรงที่ต้องการสำหรับการตัด คือ 235.9 N ระยะ r คัดจาก รัศมีตัดของลวด + รัศมีของ Pin + 1.5 ของ รัศมีของลวด

$$r = 6 + 3 + 4.2 = 13.2 \quad \text{mm}$$

ตามความสัมพันธ์

$$\text{Torque} = F \times r \quad (3.13)$$

$$\text{แทนค่า (3.1)} \quad = 235.9 \times 13.2$$

$$= 3.114 \text{ Nm}$$

$$\text{Safety factor 200\%} = 3.114 \times 2$$

$$= 6.228 \text{ Nm}$$

ต้องใช้มอเตอร์ที่สามารถให้ torque ขนาด 6.228 Nm หรือมากกว่า เพื่อใช้ในการ

ตัด

3.1.10.2 มอเตอร์ที่ใช้ในการหมุนแขนตัด

คิดที่ $r = 80 \text{ mm}$ เมื่อ $F = mg$ เนื่องจากเป็นแรงที่มากที่สุดที่เกิดขึ้นเมื่อมีการหมุน โดยแรงนี้เกิดเมื่อมุมของแขนตัด 90°

$$\text{แทนค่า (3.1)} \quad \text{Torque} = F \times r$$

$$= 60 \times 0.08$$

$$= 4.8 \text{ Nm}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.12 เลือกขนาดมอเตอร์

ตัวหัวตัด ใช้สแต็ปมอเตอร์รุ่น Vexta Pk566AW-N7.2 ขนาดแรงบิด 1.6 Nm



รูปที่ 3.3 แสดงสแต็ปมอเตอร์ Vexta

ตัวหมุนแกนตัด ใช้เซอร์โวมอเตอร์ รุ่น Yokogawa DM1015B แรงบิด 15 Nm



รูปที่ 3.4 แสดงเซอร์โวมอเตอร์ Yokogawa

ตัวพีดลวด ใช้สแต็ปมอเตอร์ รุ่น mycom PF569-AC แรงบิด 2.0 Nm



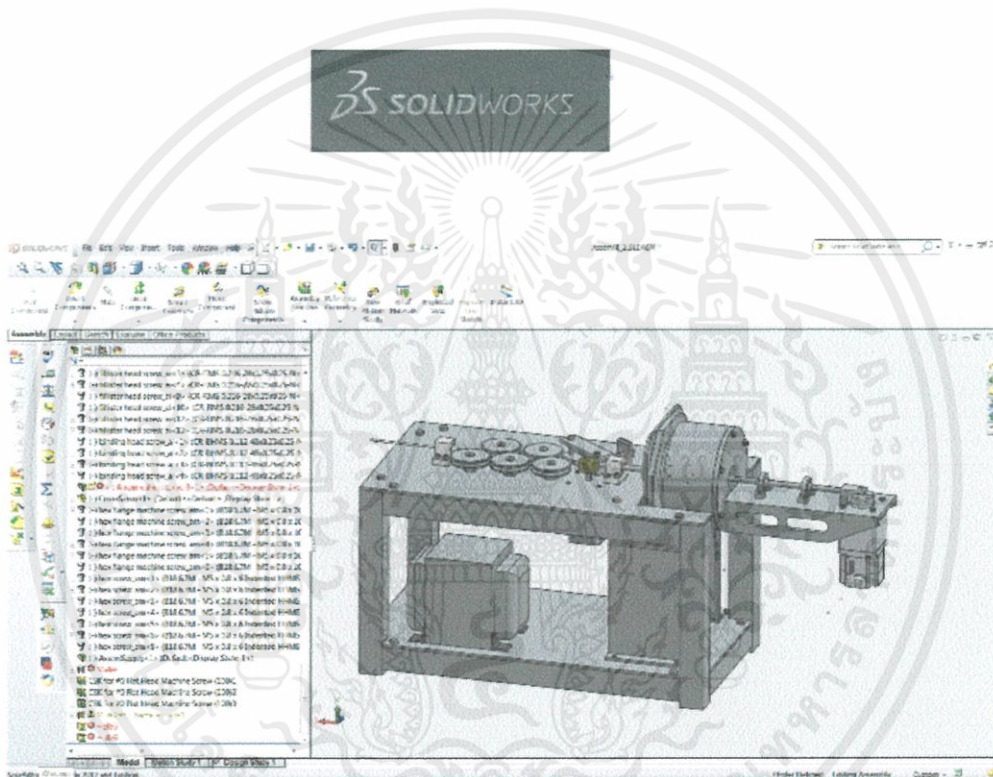
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5 แสดงสแต็ปมอเตอร์ mycom

3.1.13 การเขียนแบบ

ในการเขียนแบบจะใช้โปรแกรม SolidWorks ออกแบบทั้งหมด สามารถสร้างชิ้นงานจำลอง ได้อย่างสมบูรณ์แบบ สามารถนำไปใช้ในการคำนวณทางวิศวกรรม และการตรวจสอบความผิดพลาดได้ และเพื่อเป็นแบบในการสื่อสารกับผู้อื่นให้สามารถเข้าใจตรงกัน และง่ายในการอธิบาย

ซึ่งทั้งหมดที่ออกแบบ จะต้องดูของว่า มีตามต้องการหรือไม่ ถ้ามีก็สามารถวัดขนาด สัดส่วนทั้งหมด แล้วนำส่วนทุกส่วนมาประกอบ เพื่อสร้างแบบจริงต่อไป

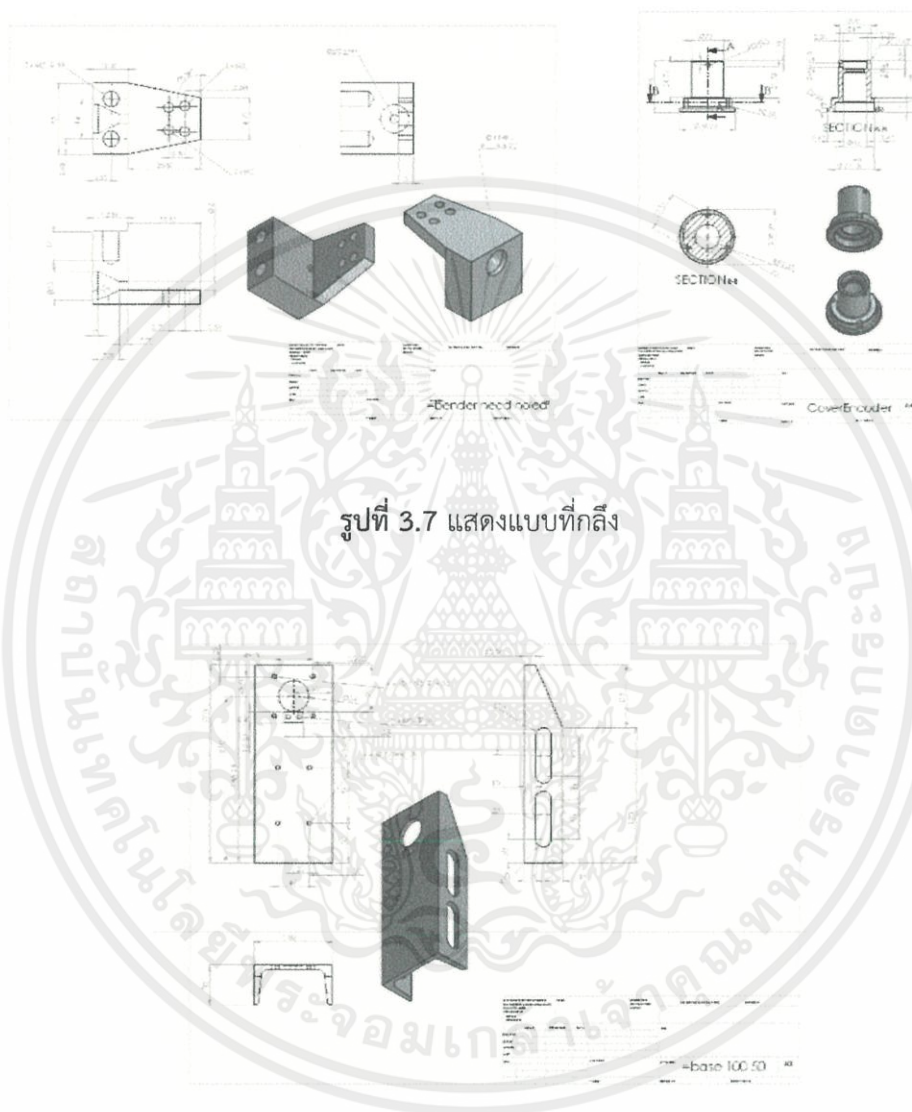


รูปที่ 3.6 แสดงโปรแกรมที่ใช้เขียนแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.14 การทำแบบขึ้นส่วนเฉพาะ

ในส่วนแบบขึ้นส่วนเฉพาะ ที่มีการตัดเพื่อลดน้ำหนัก หรือสั่งกลึง จะต้องสร้างแบบอย่างละเอียด และทำไฟล์แบบดรออิง สำหรับส่งแบบการตัด เจาะรู ปาดและกลึง

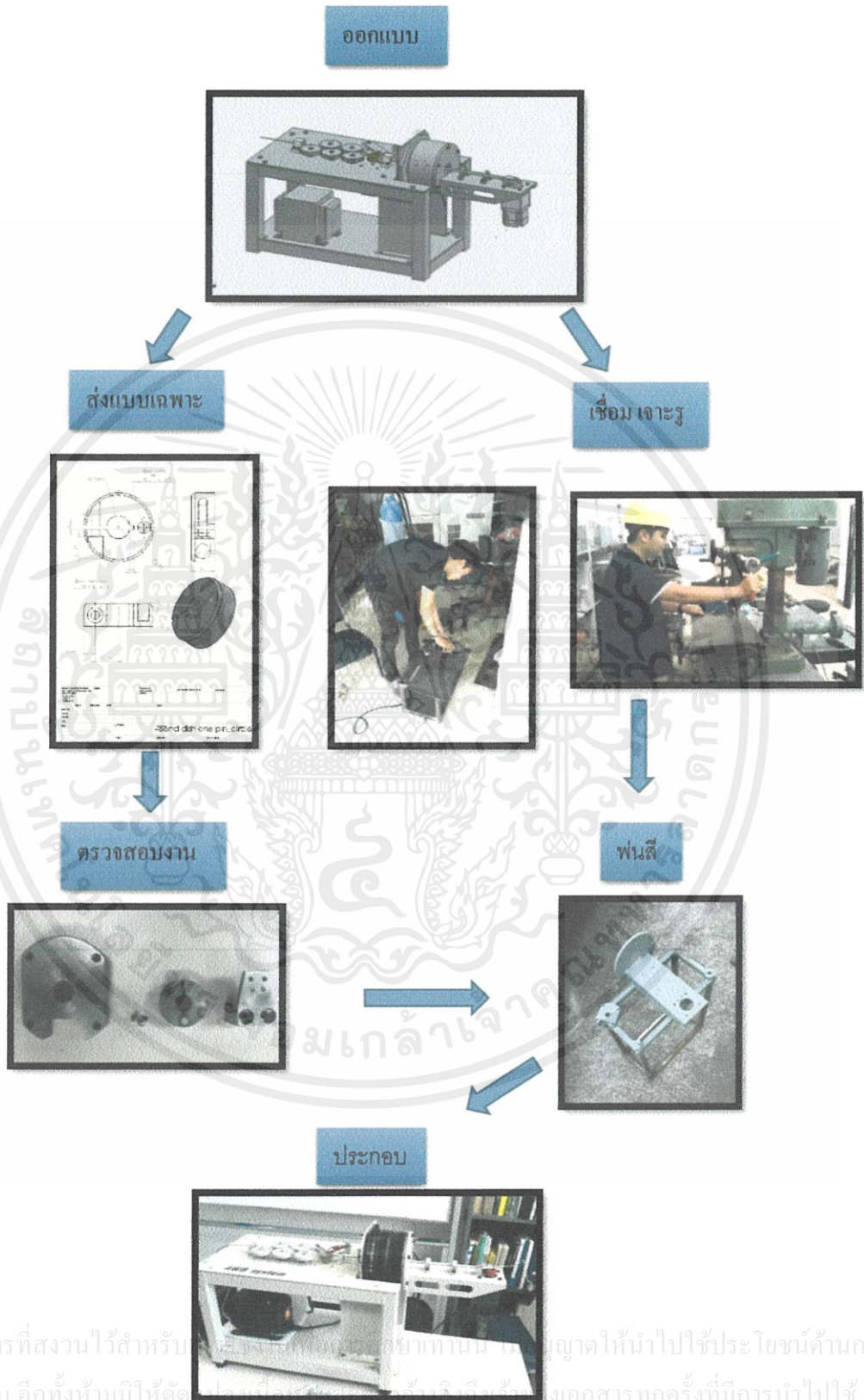


รูปที่ 3.7 แสดงแบบที่กลึง

รูปที่ 3.8 แสดงแบบที่ต้องตัด เจาะรู และปาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.15 ขั้นตอนการสร้างเครื่อง

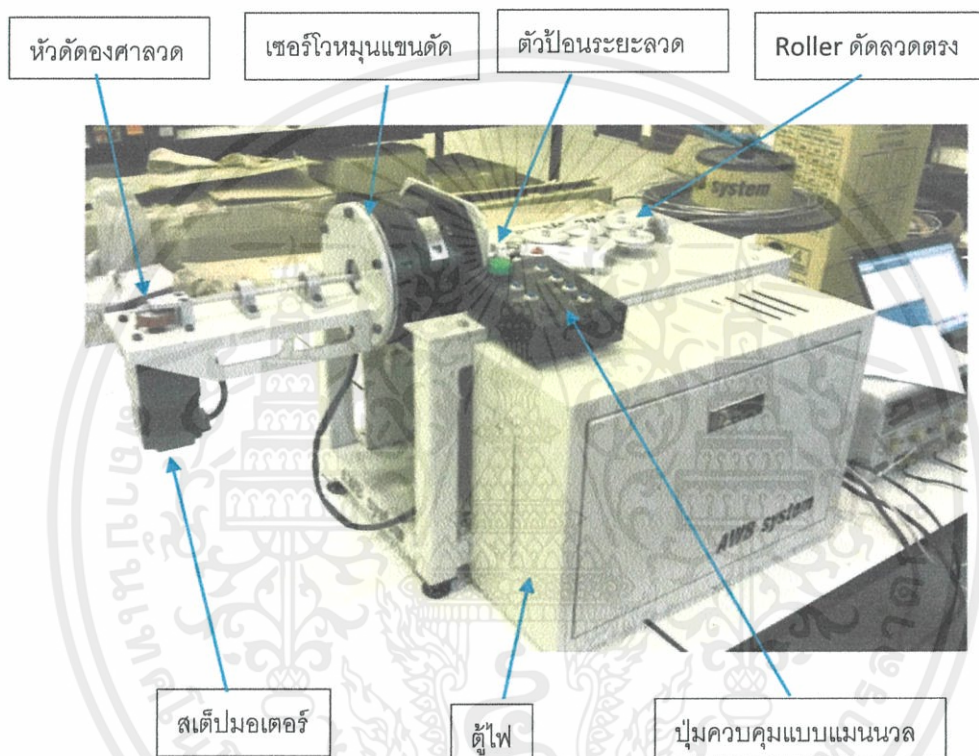


รูปที่ 3.9 แสดงการทำโครงประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอก... เอกสารทุกครั้งที่มีการนำไปใช้

3.1.16 ส่วนประกอบหลักของเครื่อง

เป็นส่วนประกอบด้วยหลักของเครื่องที่ประกอบด้วย ตัวพีตระยะลาด Roller ดัดลดตรง เซอร์โวหมุนแกนตัด หัวตัดลดท หม้อแปลง ปุ่มควบคุมแบบแมนนวล และสตีปมอเตอร์ โดยแต่ละส่วนทำหน้าที่ต่างกันไป



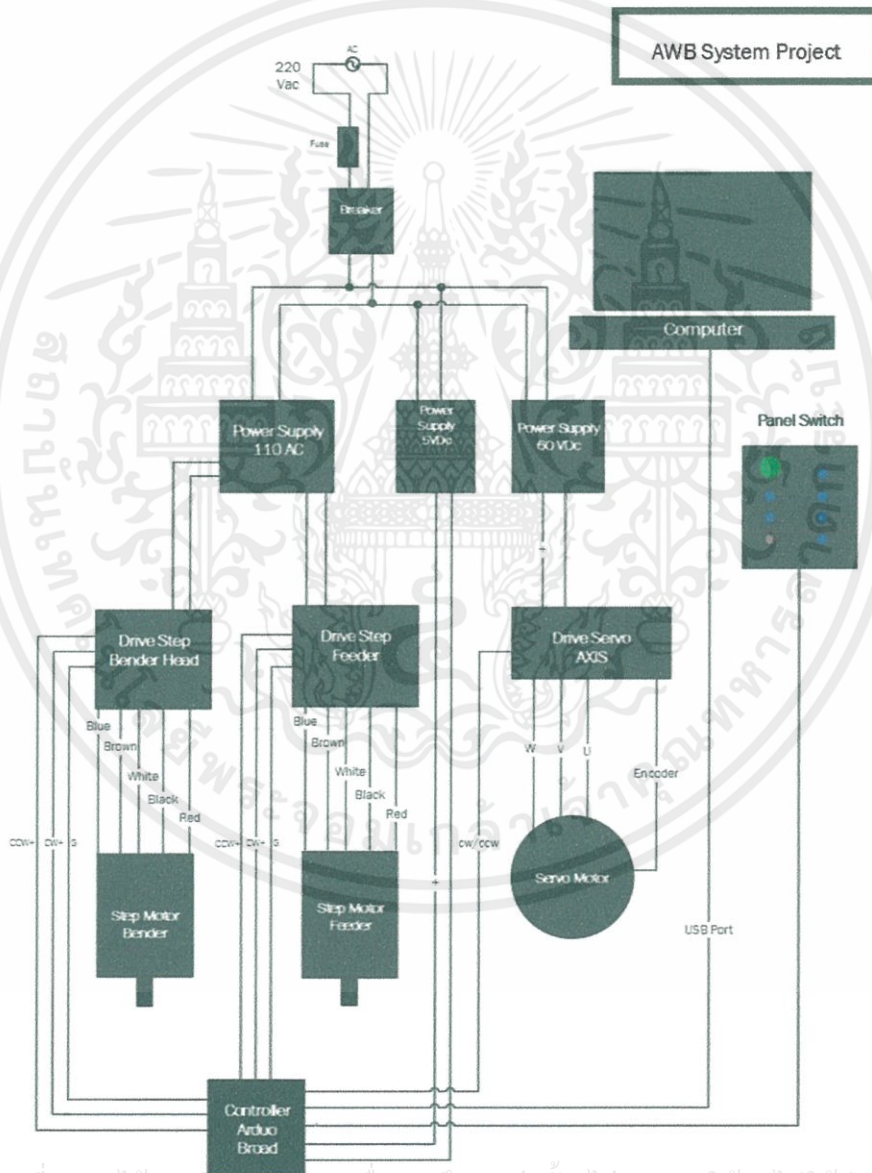
รูปที่ 3.10 แสดงส่วนประกอบหลักของเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ชุดควบคุม

3.2.1 การต่อระบบการจ่ายไฟ

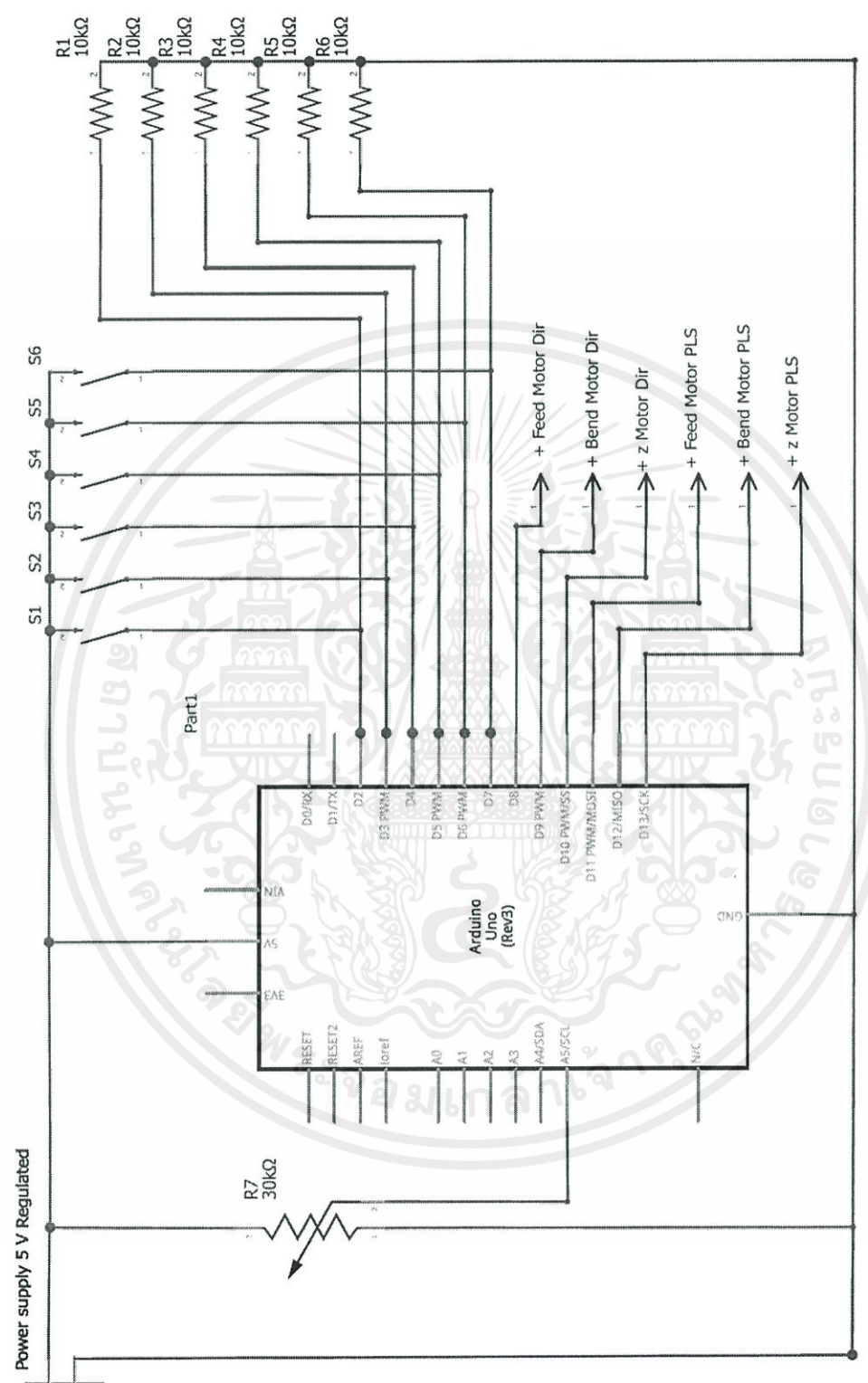
เป็นแบบการต่อสายไฟไปยังส่วนต่างๆ ประกอบด้วย เบรกเกอร์ เพาเวอร์ซัพพลาย สเต็ปมอเตอร์ เซอร์โวมอเตอร์ ไมโครคอนโทรลเลอร์ ไดรฟ์มอเตอร์ ปุ่มควบคุม และคอมพิวเตอร์ ของตัวเครื่องให้สามารถทำงานได้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และแจ้งอ้างอิงเป็นเนื้อหาของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.11 แสดงแบบการต่อไฟฟ้า

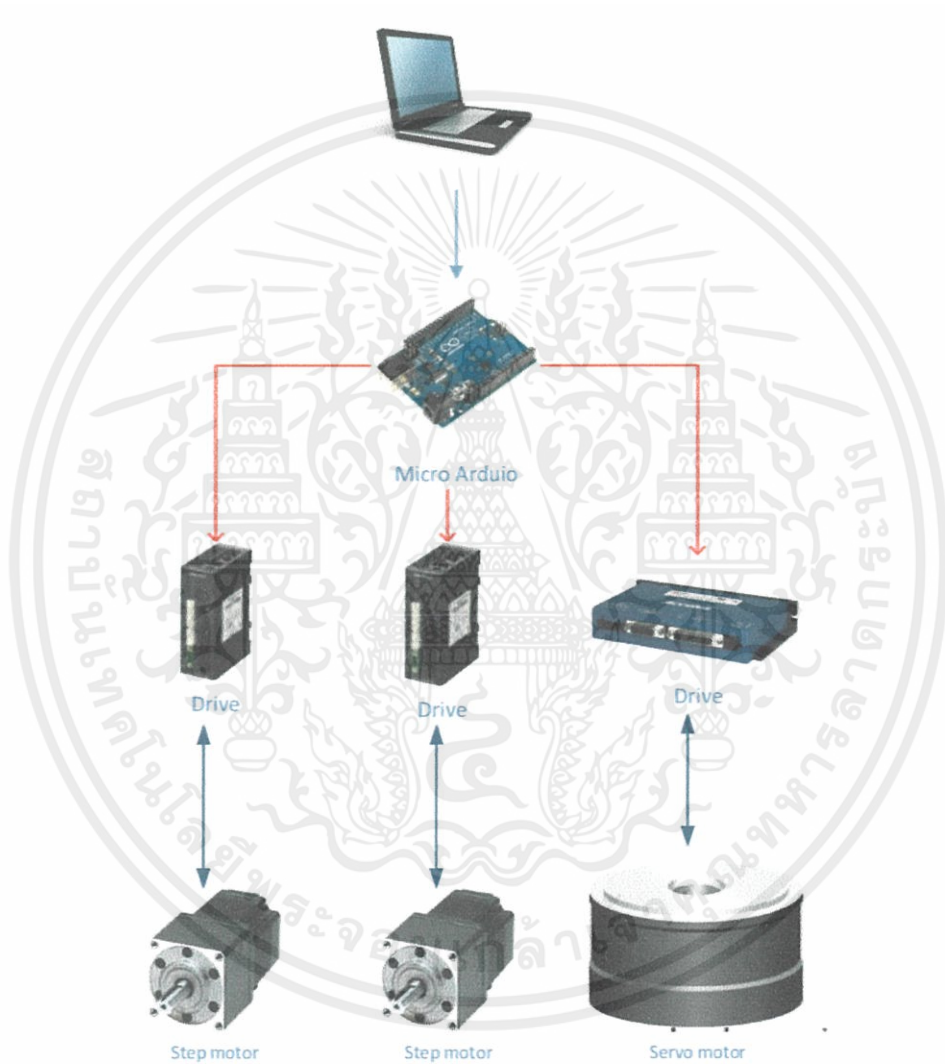
Schematic Diagram ของส่วนควบคุม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 ลำดับการสั่งส่วนควบคุม

เป็นขั้นตอนการสั่งงานตั้งแต่การกำหนดชุดรูปร่างที่จะตัดผ่านทางโปรแกรมคอมพิวเตอร์ไปยังส่วนไมโครคอนโทรลเลอร์ ードรฟ์ จนถึงสแต็ปมอเตอร์ และเซอร์โวมอเตอร์ ซึ่งทั้งหมดจะทำงานร่วมกัน โดยมีลำดับการทำงานตามคำสั่งที่ได้รับ และรูปร่างต่างๆที่ต่างกัน



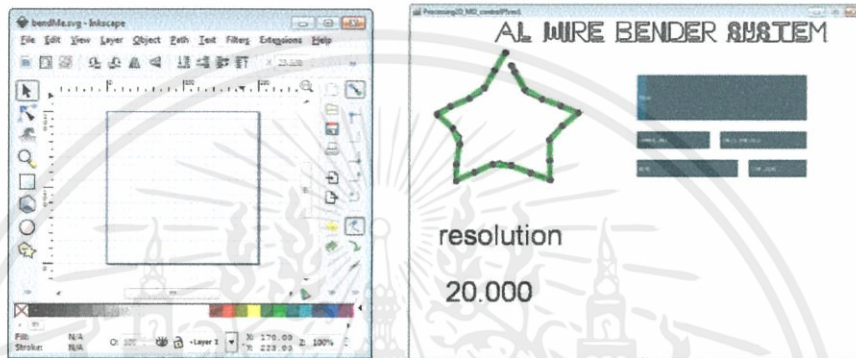
รูปที่ 3.12 แสดงลำดับการสั่งงานส่วนควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ชุดกำหนดรูปแบบการตัด

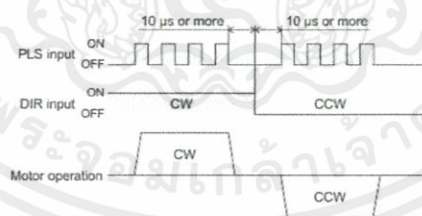
ผู้ใช้งานทำงานผ่านชุดกำหนดรูปแบบของการตัด แล้วชุดนี้จะส่งงานไปยังเครื่องตัดลวด

1. ใช้ระยะ และมุมทำให้เกิดรูปร่าง เป็นหลักในการสร้างชุดควบคุม โดยกำหนดผ่านโปรแกรม Inkscape ซึ่งจะบันทึกรูปที่วาดเป็นไฟล์ vector ที่มีทั้งขนาดและทิศทาง แล้วจึงนำค่าไปยังโปรแกรม Processing เพื่อกำหนดเป็นมุม และระยะป้อน ผ่าน User Interface ไปยังเครื่องตัดลวดอัตโนมัติ



รูปที่ 3.14 โปรแกรม Inkscape และ User Interface

2. เครื่องตัดลวดรับค่ามาในส่วนของตัวควบคุม Arduino UNO microcontroller board โปรแกรมที่ถูกสร้างขึ้นจะประมวลผลระยะป้อน และมุมต้นเป็น pulse และ direction ไปยัง driver ของ step motor และ servo motor แต่ละตัว



รูปที่ 3.15 รูปแบบของสัญญาณที่ส่งไปยัง driver ของมอเตอร์

3. มอเตอร์แต่ละตัวทำงานประสานกันตามกระบวนการตัดลวดที่ตั้งค่าไว้จนลวดเกิดเป็นรูปร่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลอง และผลการทดลอง

4.1 ระยะเวลา Feed ลวด

เป็นการทดสอบวัดค่าความยาวของลวดที่ถูกป้อนออกมา เทียบกับระยะ Feed ที่กำหนดด้วยโปรแกรม เพื่อทดสอบความแม่นยำของชุดป้อนลวด

ตารางที่ 4.1 แสดงระยะที่ Feed ลวดจริง

ระยะ Feed (mm)	ครั้งที่ 1 (mm)	ครั้งที่ 2 (mm)	ครั้งที่ 3 (mm)	ค่าเฉลี่ย (mm)	ความผิดพลาด (%)
10	11.26	9.59	9.19	10.01	0
20	20.01	20.00	20.30	20.10	0.5
40	40.77	40.41	39.46	40.21	0.5
80	81.16	83.34	82.45	82.31	2.8
100	103.44	103.36	103.97	103.59	3.5
120	118.47	122.33	121.45	120.75	0.6
140	139.00	140.00	139.50	139.50	-0.3
160	161.90	163.50	162.03	162.47	1.5
200	202.50	201.49	199.00	200.99	0.4
250	249.00	249.50	249.90	249.17	0
300	303.82	298.86	303.60	302.09	0.6

4.2 มุมองศาการตัด

เป็นการทดสอบวัดค่ามุมองศาที่ได้จริง เพื่อเปรียบเทียบค่าที่สั่งผ่านโปรแกรม ว่าค่าที่ได้มีความใกล้เคียงมากน้อยเพียงใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 แสดงมุมดัดของลวด

มุมดัด (degree)	ครั้งที่ 1 (degree)	ครั้งที่ 2 (degree)	ครั้งที่ 3 (degree)	ค่าเฉลี่ย (degree)	ความผิดพลาด (%)
10	10.00	11.00	11.00	10.67	6.7
20	19.00	17.50	18.70	18.40	8
30	29.00	28.00	29.50	28.83	-3.9
40	39.50	36.50	38.90	38.30	-4.2
50	50.00	49.00	49.50	49.50	-1
60	60.50	56.00	58.50	58.33	-2.7
70	71.00	65.20	67.00	67.73	-3.2
80	79.00	77.50	78.00	78.16	-2.3
90	90.00	86.50	88.50	88.33	-1.8
100	101.00	100.20	103.00	101.10	1.1
110	111.00	109.00	113.00	111.00	0
120	123.00	122.50	123.00	122.83	2.3
130	133.50	132.50	131.00	132.33	1.7
140	145.50	140.50	143.50	143.17	2.3
150	156.00	152.50	152.00	153.50	2.3
160	161.50	160.00	161.00	160.83	0.5



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปที่ 4.1 แสดงมุมดัดที่ทดลอง 10-160 องศา
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดเบสลงเนื้อหา และต้องอ้างอิงถึงชื่อของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทวิจารณ์และสรุป

5.1 บทวิจารณ์

ผลจากการวัดค่าระยะที่ลวดถูกป้อน ขนาดของที่ได้จริง กับ ระยะป้อน และมุมที่สั่ง มีความคลาดเคลื่อน จึงสร้างสมการเพื่ออธิบายรูปแบบการตัด เพื่อปรับปรุงให้ค่าที่ได้มีใกล้เคียงกันมากที่สุด การทำรูปแบบการตัดตามการวัดและกำหนดของผู้ใช้ เมื่อทำงานตัดจริง งานที่ตัดออกมาเป็นไปตามรูปร่างที่วาดไว้ตามโปรแกรม เว้นแต่ในรูปร่างที่มีการโค้งงอมากหรือซับซ้อน ผลการตัดที่ได้จะเกิดข้อผิดพลาดตามข้อจำกัดของเครื่อง

5.2 สรุป

ระบบตัดลวดอคูมิเนียม แบ่งการควบคุมหลักได้ 3 แบบคือ การป้อนลวด การตัดองศาลวด และส่วนหมุนแกนแกนตัด ซึ่งในการควบคุมสั่งการจะใช้ไมโครคอนโทรลเลอร์ Arduino Uno โดยจะถูกสั่งงานการประมวลผลของโปรแกรม Processing และสร้างรูปแบบในการตัดด้วย โปรแกรม Inkscape ซึ่งชุดโปรแกรมทั้งหมดเป็น Open Source แปรส่งข้อมูลจากรูปแบบที่ต้องการเป็นคำสั่งให้กับเครื่องตัดลวด แล้วเครื่องตัดลวดจึงทำงานตัดตามคำสั่งที่ได้รับ ลวดออกมาเป็นรูปร่างตามกำหนด

5.3 ข้อเสนอแนะ

ระบบตัดลวดอคูมิเนียมนี้ จะมีประโยชน์อย่างยิ่งหากได้รับการพัฒนาเป็นให้สามารถตัดโลหะที่มีเส้นผ่านศูนย์กลางมากขึ้นได้ และใช้ในงานภาคอุตสาหกรรมขนาดเล็ก อีกทั้งสามารถพัฒนาเป็นเครื่องตัดท่อซึ่งสามารถใช้ประโยชน์จากการตัดได้มากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] บรรเลง ศรนิล, สมนึก วัฒนศรียกุล. ตารางคู่มืองานโลหะ. กรุงเทพมหานคร : มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ : 2554
- [2] Engineer-abc. “Coefficient of friction, Rolling resistance and Aerodynamics”. [Online]. Available: <http://www.tribology-abc.com/abc/cof.htm>
- [3] Wikipedia. “Rolling resistance”. [Online]. Available: http://en.wikipedia.org/wiki/Rolling_resistance
- [4] Copley Controls. “Setup & Tuning”. [Online]. Available: <http://www.copleycontrols.com/motion/Products/Software/driveSoft.html>
- [5] Oriental motor. “Stepper Motor”. [Online]. Available: <http://orientalmotor.com/technology/index.html#step>
- [6] Arduino. “Arduino Uno”. [Online]. Available: <http://arduino.cc/en/Main/ArduinoBoardUno>
- [7] สุชิน เชื้อน้อย. “Servo motor”. [Online]. Available: http://www.9engineer.com/index.php?m=article&a=show&article_id=2175
- [8] Aimagin. “Stepper motor”. [Online]. Available: <http://aimagin.com/blog/driving-step-motor-and-rc-servo-motor/?lang=th>
- [9] Kmutt. “ทฤษฎีการขึ้นรูปงานโลหะแผ่นโดยการดัดงอ”. [Online]. Available: <http://eu.lib.kmutt.ac.th/elearning/Courseware/TEN437/main/e-learning/lessen/04/main.htm>

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Processing

```

//import these libraries from libraries folder
import processing.serial.*;
import geomerative.*;
Serial arduino; //object "arduino" of the datatype "Serial" is
created ***29012014
//Declaring the objects being used to print
RShape shp;
RPoint[] points;
int maxArray = 255;
float[] angles = new float [maxArray]; //a new array of floats called "angles"
that have the length = maxArray is created***29012014
float[] feeds = new float [maxArray];
float res = 20; //default resolution
int interimSteps = 1; //a counter for the preview
int steps =1;
int lastpty = 0;
boolean chngRes = false;
boolean preview = false;
void setup() {
  size (400, 400); //creates window
  background (255);
  println(Serial.list());
  arduino = new Serial(this, Serial.list() [0], 9600); //sets arduino usb port, a new
object "arduino" of the datatype "Serial" is created ***29012014
// VERY IMPORTANT: Allways initialize the library in the setup
  RG.init(this);
  shp = RG.loadShape("bendMe.svg"); //loads file from folder
  steps = int(shp.getCurveLength()/res);
  smooth(); //make the edge smooth
  frameRate(24); // set frame rate that refresh in fps
***29012014
  println ();
  println ("SET THE SYSTEM TO HOME POSITION");
  println ("PRESS MOUSE BUTTON TO SET RESOLUTION");
  println ("PRESS:");
  println ("1. PROCESS SHAPE");

```

```

println ("2. PREVIEW SHAPE");
println ("3. SEND SHAPE TO ARDUINO");
println ();
println ("RESOLUTION IS CURRENTLY = " + res);
println ("NUMBER OF STEPS = " + steps);
}
void draw() {
  translate(width/4, height/4);          // Sets draw location and scale
  noFill();
  if (!preview) {                        //draws exact shape from file bendMe.svg
    RG.setPolygonizer(RG.ADAPTATIVE);
    stroke(0, 0, 200, 150);             // set the color of point
    shp.draw();
    RG.setPolygonizer(RG.UNIFORMLENGTH);
    if (chnRes) {                        //sets resolution if mouse is pressed
      background(255);
      float minDist = shp.getCurveLength()/maxArray;           // minimum line
distance(resolution) based on array size
      res = map(float(mouseY), 0.0, float(height), minDist, 128); //resolution based
on mouse location
      shp.draw();
    }
    RG.setPolygonizerLength(res);
    points = shp.getPoints();           //point resolution based on mouse
location
    // draw a shape of lines connecting the points
    if (points != null) {
      noFill();
      stroke(0, 200, 0, 50);
      strokeWeight (5);
      beginShape();
      for (int i=0; i<points.length; i++) {
        vertex(points[i].x, points[i].y);
      }
      endShape();
      strokeWeight (1);
      fill(0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

stroke(0);
for (int i=0; i<points.length; i++) {
  ellipse(points[i].x, points[i].y, 5, 5);
}
}
}
else { //draws preview shape based on set
resolution
  background (255);
  stroke (random (255), random (255), random (255));
  strokeWeight (5);
  pushMatrix();
  noFill();
  beginShape();
  for (int i=1; i<=steps+2; i++) { //preview bend by relative angles
    line (0, 0, 0, feeds[i-1]);
    translate (0, feeds [i-1]);
    rotate (-radians(angles [i+1])); //bend4
  }
  preview = !preview;
  popMatrix ();
}
}

void mousePressed() { //mouse press sets resolution based
on mouse location
  println ("RESOLUTION SET TO = " + res + " mm");
  steps = int(shp.getCurveLength()/res);
  println ("NUMBER OF STEPS = " + steps);
  chngRes = !chngRes;
}

void keyPressed() { //If keys are pressed run corrsponding
subroutine
switch (key) {
case '1': //calculates bend angles and feed lengths
  calcs(); // call function calcs ***29012014
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และ

```

break;

case '2': //shows a shape preview based on set
resolution
    preview = !preview; // to do preview in draw fn
***29012014
    break;

case '3': //sends bend angles and feed lengths to
the arduino for printing
    sendArd (); // to do fn sendArd
    break;

case '4': //sends bend angles and feed lengths to
the arduino for printing
    stopWork (); // to do fn sendArd
    break;
}
}

//this subroutine accepts feedback from the arduino. It is not necessary but
sometimes needed to establish initial serial communication
void serialEvent (Serial p) {
    String inString = arduino.readStringUntil ('\n');
    if (inString !=null) {
        println(inString);
    }
}

void calcs () {
    float lastAng=0;
    float lastlastAng = 0;
    println ( key );
    println ();
    println ( "Processing shape" );

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ผู้ใช้ที่ฝ่าฝืนให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

println ();
println (points.length);
//calculate relative angles between points and assign to array angles[]
angles [0] = 0; // first angle is zero
for (int i = 1; i < points.length; i++) {
    feeds[i-1] = dist (points[i-1].x, points[i-1].y, points[i].x, points[i].y); //feed
lengths are the distances between points ***29012014 Got feed length
    float deltaX = points[i].x - points[i-1].x; //change in x axis
between points
    float deltaY = points[i].y - points[i-1].y; //change in y axis
between points

//the following equations calculate the bend angles between each point
//there are different equations for each coordinate quadrant
if ((deltaX >=0 && deltaY >=0)||((deltaX <=0 && deltaY >=0)) {
    angles[i] =(degrees(atan(deltaX/deltaY)) - (lastAng + lastlastAng));
}
if (deltaX <= 0 && deltaY < 0) {
    angles[i] = (180 + abs(degrees(atan(deltaX/deltaY)))) - (lastAng + lastlastAng);
}
if (deltaX >0 && deltaY <0) {
    angles[i] = (180 - abs(degrees(atan(deltaX/deltaY)))) - (lastAng + lastlastAng);
}
if (angles[i]>720) {
    angles[i] = angles[i]-360;
}
if (angles[i]<-180) {
    angles[i] = 360+angles[i];
}
if (angles[i]>180) {
    angles[i] = angles[i]-360;
}

//shows matrix of feeds and bend angles
println (round(1 * ( feeds[i-1 ] )) + " \t"+round(angles[i])+"\t"+"0 " );
//change the 1 before feeds to change scale. z bend remains 0 because this is a 2D
shape

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับเอาไว้ใช้งานเพื่อการศึกษานานาชาติ โดยเอกสารนี้จะไม่ไปใช้ประโยชน์ด้านอื่นๆ
 ไม่ว่ากรณีใดๆก็ตาม อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (abs(angles [i]) > 160) {                                     //because we are sending
bytes to the arduino board no angle or feed length can exceed 125 *****set max
bending angle here
        println ("WARNING - ANGLE TOO BIG");
    }

    lastlastAng = lastAng+lastlastAng;                             //sets variables to know
last 2 bend angles needed to calculate bends
    lastAng = angles[i];
}
}

void sendArd () {                                                 //sends bend angles and feed lengths to
the arduino board for printing
    byte feedmotor = 126;                                         //these byte values are markers so the
arduino knows if a value is a bend, feed, or end it's the end of the array
    byte bendmotor = 125;
    byte end = 127;

    println ( key );
    println ();
    println ("Sending to Arduino" );
    println ("Printing" );
    println ();

//send commands
for (int i = 2; i < steps +10; i++) {                             // **** 31012014 original steps +233

//sends arduino feed marker followed by feed length
    delay (100);
    arduino.write (feedmotor);
    delay (100);
    arduino.write (byte(round (feeds[i-2])));
    delay (100);
    println (feeds[i-2]);
//sends arduino bend marker followed by bend angle
    arduino.write (bendmotor);
    delay (100);

```

```

    arduino.write (byte(round (-angles[i])));
    println (angles[i];
}
println (end+128);
println("commands sent to printer"); //tells arduino the array of values in complete
****added 31012014
    arduino.write(end);
}

```

```

void stopWork(){
    byte end = 127;
    println ("clear value");
    arduino.write (end);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Arduino

```

// control manual by switches
const int swFeed = 2;
const int swFeedBack = 3;
const int swBend = 4;
const int swBendBack = 5;
const int swZRotate = 6;
const int swZRotateBack = 7;

// motor control
const int feedMotorDir = 8;
const int bendMotorDir = 9;
const int zMotorDir = 10;
const int feedMotorPls = 11;
const int bendMotorPls = 12;
const int zMotorPls = 13;

const int pulseWidth = 250;
const int pulseDelay = 200;

int byteRead;

float dist = 0; // Record the number of steps we've taken
float lastDist = 0;
int flipArmDeg = 2048 ; // flip 180 deg to bend negative angle
float angle = 0;
float angleC = 0;
float rotations;
float lastAngP = 0;

// Drive directions
boolean ccw = true; // counter-clockwise drive direction
boolean cw = false; // clockwise drive direction
boolean curDir = ccw;

// declare manual control

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int swFeedState = 0 ;
int swFeedBackState = 0;
int swBendState = 0 ;
int swBendBackState = 0 ;
int swZRotateState = 0;
int swZRotateBackState = 0;
int sensorPin = A5;           // select the input pin for the
potentiometer
int sensorValue = 0;         // variable to store the value coming from
the sensor
int pulseWidthM = 0 ;
int pulseDelayM = 0 ;

int lastbend = 0;
int fieldindex=0;
int values[300];             //creates array

boolean dodge = false;
float lastAng = 0;
float lastFeed = 0;
int feedCount = 0;

void setup() {
  pinMode(swFeed, INPUT);
  pinMode(swFeedBack, INPUT);
  pinMode(swBend, INPUT);
  pinMode(swBendBack, INPUT);
  pinMode(swZRotate, INPUT);
  pinMode(swZRotateBack, INPUT);
  pinMode(feedMotorDir, OUTPUT);
  pinMode(bendMotorDir, OUTPUT);
  pinMode(zMotorDir, OUTPUT);
  pinMode(feedMotorPls, OUTPUT);
  pinMode(bendMotorPls, OUTPUT);
  pinMode(zMotorPls, OUTPUT);
  // digitalWrite(feedMotorDir, HIGH);
  digitalWrite(bendMotorDir, LOW);
  digitalWrite(zMotorDir, HIGH);

```

```

Serial.begin(9600);
}

void feed(float dist){
  if (dist != 0){
    Serial.println("feeding");
    Serial.println(dist);
    lastFeed = dist;
    float rotations = 0;
    float feedCalib = 85; //feed mm per revolution of motor_ Edited
    **28012014 original is 25.4*PI
    lastDist = dist;
    dist = (dist/feedCalib)*1000; //dist converted from px in processing to
steps_ Edited ( 1 mm = 3.779528)
    rotations = dist;
    delay(1000);
    digitalWrite(feedMotorDir, HIGH);
    for(int i=0 ; i<=rotations ; i++){
      digitalWrite(feedMotorPls, HIGH);
      delayMicroseconds(pulseWidth);
      digitalWrite(feedMotorPls, LOW);
      delayMicroseconds(pulseDelay);
    }

    feedCount ++;
  }
}

void bend (int angle) { //bender pin movement
  /* if(dodge == true){
    unDodgePin();
  }
  */
  if (angle!=0){ //sets direction of bend based on +
    or - angle
    Serial.println("bending");
    Serial.println(angle);
    /* boolean dir=cw;

```

```

boolean back=ccw;
if (angle<0){
  dir = ccw;
  back = cw;
}
*/
angle = abs(angle);

```

```

if(feedCount == 1 && lastFeed < 10){
  float plusAng = (lastAng -plusAng)/lastFeed;
  angle = angle + plusAng;
}

lastAng = angle;
delay(1000);
float rotations = 0;
float angleCalib = 20; //bending angle calibrate
/* if(angle <= 10 ){
  angle = -.00347*angle*angle + 1.8577*angle - .0677;
}
if(10 < angle <= 25 ){
  angle = .0099*angle*angle + 0.2679*angle + 11.329;
}
if(25 < angle <= 50 ){
  angle = -.0099*angle*angle + 1.5807*angle - 9.3919;
}
if(50 < angle <= 100 ){
  angle = -.0021*angle*angle + 0.953*angle + 2.7475;
}
if(100 < angle <= 125 ){
  angle = 0.5094*angle + 25.707;
}
if(125 < angle <= 160 ){
  angle = .0109*angle*angle - 2.5875*angle + 242.4;
}
*/
// angle = -.0016*angle*angle + .8984*angle + 3.5414;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ * ลื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//angle = 0.6434*angle + 9.238;
/*
if(angle <= 55 ){
angle = - .0028*angle*angle + 0.9975*angle + 2.3959;
}
if(angle > 55){
angle = - .0006*angle*angle + 0.6795*angle + 14.379;
}
*/

```

```

// angle = .00000009*angle*angle*angle*angle - .00002*angle*angle*angle -
.0012*angle*angle + .9375*angle + 3.358;
// angle = .0000001*angle*angle*angle*angle - .00002*angle*angle*angle -
.0011*angle*angle + .9503*angle + 2.566;

```

```

// 5 order is too much degree
// angle = .000000008*angle*angle*angle*angle*angle -
.000003*angle*angle*angle*angle + .0005*angle*angle*angle - .0314*angle*angle +
1.7598*angle + 2.7969;

```

// quite good

```

// angle = .00000002*angle*angle*angle*angle + .00002*angle*angle*angle -
.0069*angle*angle + 1.2687*angle + 4.669;

```

```

angle = .00000007*angle*angle*angle*angle + .0000007*angle*angle*angle -
.0053*angle*angle + 1.2151*angle + 4.8845;

```

```

//angle = - .0000001*angle*angle*angle*angle + .00006*angle*angle*angle -
.0092*angle*angle + 1.3031*angle + 4.8304;

```

```

//angle = - .0000003*angle*angle*angle*angle + .0001*angle*angle*angle -
.0134*angle*angle + 1.4179*angle + 6.0626;

```

```

//angle = -.0000000002*angle*angle*angle*angle*angle*angle +
.00000002*angle*angle*angle*angle*angle - .000005*angle*angle*angle*angle +
.0006*angle*angle*angle - .0365*angle*angle + 1.8371*angle + 2.558;

```

//lastest

```

//angle = .000000003*angle*angle*angle*angle*angle -
.000001*angle*angle*angle*angle + .0002*angle*angle*angle - .0156*angle*angle +
1.353*angle + 4.5092;

```

```

//angle = - .00000008*angle*angle*angle*angle + .00003*angle*angle*angle -
.0062*angle*angle + 1.161*angle + 5.2779;
//      angle      =   -.000000000005*angle*angle*angle*angle*angle*angle      +
.000000005*angle*angle*angle*angle*angle      - .000002*angle*angle*angle*angle      +
.0002*angle*angle*angle - .0169*angle*angle + 1.3719*angle + 4.4573;
//avg
// angle = - .00000009*angle*angle*angle*angle + .00004*angle*angle*angle -
.0082*angle*angle + 1.2841*angle + 3.3614;
//      angle      =           .000000005*angle*angle*angle*angle*angle      -
.000002*angle*angle*angle*angle + .0003*angle*angle*angle - .0236*angle*angle +
1.5869*angle + 2.4151;
//      angle      =   -.00000000003*angle*angle*angle*angle*angle*angle      +
.00000002*angle*angle*angle*angle*angle      - .000004*angle*angle*angle*angle      +
.0005*angle*angle*angle - .0303*angle*angle + 1.6779*angle + 2.2264;

/*
  if(angle <= 29){
    angle = -.0002*angle*angle*angle*angle + .0135*angle*angle*angle -
.3037*angle*angle + 3.5782*angle + .2844;
  }
  if(29 < angle <= 64){
    angle = -.00002*angle*angle*angle*angle + .0032*angle*angle*angle -
.2065*angle*angle + 6.5735*angle - 46.595;
  }
  if(angle > 64){
    angle = .0000003*angle*angle*angle*angle - .00009*angle*angle*angle +
.008*angle*angle + .5961*angle + 9.0398;
  }
*/

```

```

Serial.println(angle);
rotations = angle*angleCalib;
digitalWrite(bendMotorDir, LOW);
for(int i=1 ; i<rotations ; i++){
  digitalWrite(bendMotorPls, HIGH);
  delayMicroseconds(pulseWidth);
  digitalWrite(bendMotorPls, LOW);
  delayMicroseconds(pulseDelay);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆก็ตาม กรุณาแจ้งผู้ดูแลระบบและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    delay (100);
    digitalWrite(bendMotorDir, HIGH);
    for(int i=1 ; i<rotations ; i++){
        digitalWrite(bendMotorPls, HIGH);
        delayMicroseconds(pulseWidth);
        digitalWrite(bendMotorPls, LOW);
        delayMicroseconds(pulseDelay);
    }
    feedCount = 0;
}
}

void zbind (float angle) { //rotates z motor
    if (angle!=0){
        Serial.println("Z bending");
        Serial.println(angle);
        boolean dir=cw;
        if (angle<0){ //+ angle is clockwise - angle is counter
clockwise
            dir = ccw;
        }
        float rotations = 0;
        angle = abs(angle);
        angle = (angle/360)*4096; //converts angle to motor steps
        rotations = angle;
        digitalWrite (zMotorDir, dir); //sets motor direction
        for (int i=0; i <=rotations ; i++){ //rotates z motor appropriate number of
steps
            digitalWrite(zMotorPls, HIGH);
            delayMicroseconds (pulseWidth);
            digitalWrite(zMotorPls, LOW);
            delayMicroseconds (2000);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void pushWire(){
  Serial.println("Pushing");
  int push = 160;
  float rotations = 0;
  float feedCalib = 85;           //feed mm per revolution of motor_ Edited
  **28012014 original is 25.4*PI
  push = (push/feedCalib)*1000;   //dist converted from px in processing to
  steps_Edited ( 1 mm = 3.779528)
  rotations = push;
  delay(1000);
  digitalWrite(feedMotorDir, HIGH);
  for(int i=0 ; i<=rotations ; i++){
    digitalWrite(feedMotorPls, HIGH);
    delayMicroseconds(pulseWidth);
    digitalWrite(feedMotorPls, LOW);
    delayMicroseconds(pulseDelay);
  }
}

void pullWire(){
  Serial.println("pulling");
  int pull = 160;
  float rotations = 0;
  float feedCalib = 85;           //feed mm per revolution of motor_ Edited
  **28012014 original is 25.4*PI
  pull = (pull/feedCalib)*1000;   //dist converted from px in processing to
  steps_Edited ( 1 mm = 3.779528)
  rotations = pull;
  delay(1000);
  digitalWrite(feedMotorDir, LOW);
  for(int i=0 ; i<=rotations ; i++){
    digitalWrite(feedMotorPls, HIGH);
    delayMicroseconds(pulseWidth);
    digitalWrite(feedMotorPls, LOW);
    delayMicroseconds(pulseDelay);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void dodgePin (){

    delay(100);
    int dodgeAng = 120;
    float rotations = 0;
    float angleCalib = 20;           //bending angle calibrate

    Serial.println(angle);
    rotations = dodgeAng*angleCalib;
    digitalWrite(bendMotorDir, HIGH);
    for(int i=1 ; i<rotations ; i++ ){
        digitalWrite(bendMotorPls, HIGH);
        delayMicroseconds(pulseWidth);
        digitalWrite(bendMotorPls, LOW);
        delayMicroseconds(pulseDelay);
    }
    dodge = true;
}

void unDodgePin(){
    if(dodge == true){
        delay(100);
        int dodgeAng = 120;
        float rotations = 0;
        float angleCalib = 20;           //bending angle calibrate

        Serial.println(angle);
        rotations = dodgeAng*angleCalib;
        digitalWrite(bendMotorDir, LOW);
        for(int i=1 ; i<rotations ; i++){
            digitalWrite(bendMotorPls, HIGH);
            delayMicroseconds(pulseWidth);
            digitalWrite(bendMotorPls, LOW);
            delayMicroseconds(pulseDelay);
        }
        dodge = false;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

```

```

void flipArm(){
  pushWire();

```

```

  Serial.println("flipArm ");

```

```

  float rotations = 0;

```

```

  delay(100);

```

```

  rotations = flipArmDeg;

```

```

  digitalWrite(zMotorDir, HIGH);

```

```

  for(int i=1 ; i<rotations ; i++ ){

```

```

    digitalWrite(zMotorPls, HIGH);

```

```

    delayMicroseconds(pulseWidth);

```

```

    digitalWrite(zMotorPls, LOW);

```

```

    delayMicroseconds(pulseDelay);

```

```

  }

```

```

  curDir = !curDir; //direction reversed for next duck

```

```

  rotations = 0;

```

```

  pullWire();

```

```

}

```

```

void flipBack(){

```

```

  pushWire();

```

```

  // if (lastAng > 30){

```

```

  //   dodgePin();

```

```

  // }

```

```

  float rotations = flipArmDeg;

```

```

  digitalWrite(zMotorDir, LOW);

```

```

  for(int i=1 ; i<rotations ; i++ ){

```

```

    digitalWrite(zMotorPls, HIGH);

```

```

    delayMicroseconds(pulseWidth);

```

```

    digitalWrite(zMotorPls, LOW);

```

```

    delayMicroseconds(pulseDelay);

```

```

  }

```

```

curDir = !curDir; //direction reversed for next duck
rotations = 0;
pullWire();
}

void loop() {

int copies = 0;
while(Serial.available())
{
int in = Serial.read();
byte incoming = in+128; //converts bytes from processing
if (incoming != 255){ //255 signifies end of incoming array
Serial.println(Serial.read());
values[fieldindex] = values[fieldindex]*10+incoming; //fills array fieldindex
from incoming processing
fieldindex++; // count up field index
}
else{
Serial.println("END"); //if array end marker inputs from processing
end
for (int i=0; i<=fieldindex;i++){
Serial.println(values[i]-128); //***30012014 print values got from
processing
}
copies=copies+1;

}

}

if (copies==1){ //once bend is complete eject shape
delay (1000);
motorrn();
if(curDir == cw){
flipBack();
}
feed(30); //eject

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

copies=copies+1;
Serial.println("bending END");
Serial.flush();

}

```

```
// manual control
```

```

swFeedState = digitalRead(swFeed);
if(swFeedState == HIGH){
  pulseWidthM = pulseWidth + analogRead(sensorPin);
  pulseDelayM = pulseDelay + analogRead(sensorPin);
  digitalWrite(feedMotorDir, HIGH);
  digitalWrite(feedMotorPls, HIGH);
  delayMicroseconds(pulseWidthM);
  digitalWrite(feedMotorPls, LOW);
  delayMicroseconds(pulseDelayM);
}
swFeedBackState = digitalRead(swFeedBack) ;
if( swFeedBackState == HIGH){
  pulseWidthM = pulseWidth + analogRead(sensorPin);
  pulseDelayM = pulseDelay + analogRead(sensorPin);
  digitalWrite(feedMotorDir, LOW);
  digitalWrite(feedMotorPls, HIGH);
  delayMicroseconds(pulseWidthM);
  digitalWrite(feedMotorPls, LOW);
  delayMicroseconds(pulseDelayM);
}

```

```

swBendState = digitalRead(swBend);
if( swBendState == HIGH){
  pulseWidthM = pulseWidth + analogRead(sensorPin);
  pulseDelayM = pulseDelay + analogRead(sensorPin);
  digitalWrite(bendMotorDir, HIGH);
  digitalWrite(bendMotorPls, HIGH);
  delayMicroseconds(pulseWidthM);
  digitalWrite(bendMotorPls, LOW);
}

```

เอกสารนี้เป็นเอกสารผลงานวิจัยหรือบทความวิชาการที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ หากมีข้อผิดพลาดประการใดขออภัยและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delayMicroseconds(pulseDelayM);
}

swBendBackState = digitalRead(swBendBack);
if(swBendBackState == HIGH ){
    pulseWidthM = pulseWidth + analogRead(sensorPin);
    pulseDelayM = pulseDelay + analogRead(sensorPin);
    digitalWrite(bendMotorDir, LOW);
    digitalWrite(bendMotorPls, HIGH);
    delayMicroseconds(pulseWidthM);
    digitalWrite(bendMotorPls, LOW);
    delayMicroseconds(pulseDelayM);
}

swZRotateState = digitalRead(swZRotate);
if(swZRotateState == HIGH){
    pulseWidthM = pulseWidth + analogRead(sensorPin);
    pulseDelayM = pulseDelay + analogRead(sensorPin);
    digitalWrite(zMotorDir, LOW);
    digitalWrite(zMotorPls, HIGH);
    delayMicroseconds(pulseWidthM);
    digitalWrite(zMotorPls, LOW);
    delayMicroseconds(pulseDelayM);
}

swZRotateBackState = digitalRead(swZRotateBack);
if(swZRotateBackState == HIGH){
    pulseWidthM = pulseWidth + analogRead(sensorPin);
    pulseDelayM = pulseDelay + analogRead(sensorPin);
    digitalWrite(zMotorDir, HIGH);
    digitalWrite(zMotorPls, HIGH);
    delayMicroseconds(pulseWidthM);
    digitalWrite(zMotorPls, LOW);
    delayMicroseconds(pulseDelayM);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void motorrun(){
  int lastbend=0;
  for (int i=0; i<= fieldindex;i++){
    delay (100);
    if ((values[i]-128)==126){                                     //convert bytes from processing and
    look for feed motor marker
      feed (values[i+1]-128);                                     //if feed motor marker detected next
    value in array is a feed length
    }
    else if ((values[i]-128)==125){                               //convert bytes from processing and
    look for bend motor marker
      int bendAng = (values[i+1]-128);                           //if bend motor marker detected
    next value in array is a bend angle

    if (bendAng<0&&curDir==ccw) {                                //if incoming bend angle is
    opposite direction from previous angle duck pin****
      flipArm ();
      delay (100);
    }
    else if (bendAng>0 && curDir ==cw){
      flipBack();
      delay(100);
    }

    bend (bendAng);
    lastbend = bendAng;
  }
  else if ((values[i]-128)==124){                               //convert bytes from processing and
  look for z motor marker
    zbind (values[i+1]-128);                                     //if z motor marker detected next
  value in array is z bend angle
  }

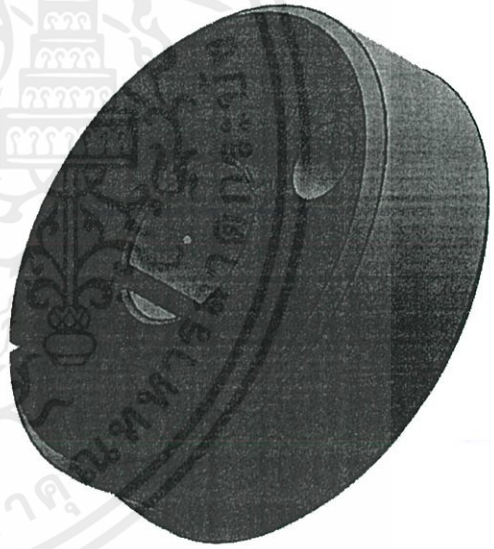
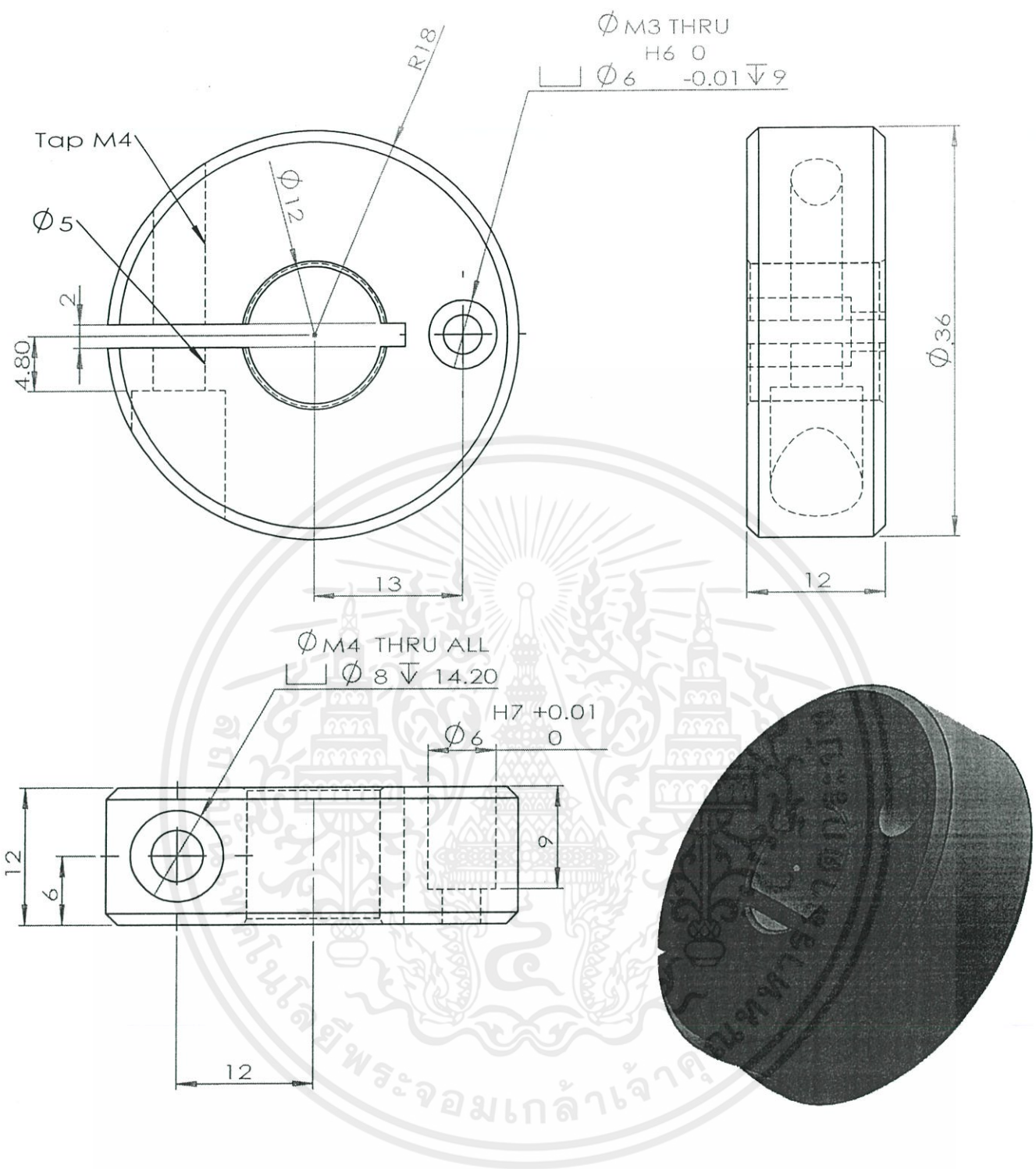
  else if (Serial.available()){
    int in = Serial.read();
    byte incoming = in+128;                                     //converts bytes from processing
    if (incoming == 255){

```

```
fieldindex =0 ;  
Serial.println("Stopping");  
Serial.flush();  
  
}  
  
}  
}  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



UNLESS OTHERWISE SPECIFIED:
 DIMENSIONS ARE IN MILLIMETERS
 SURFACE FINISH:
 TOLERANCES:
 LINEAR:
 ANGULAR:

FINISH:

DEBUR AND
 BREAK SHARP
 EDGES

DO NOT SCALE DRAWING

REVISION

NAME	SIGNATURE	DATE
DRAWN		
CHK'D		
APP'VD		
MFG		
G.A		

TITLE: แทนัน ไมออนูชาติ หนา ไปไซประ โยชนคานการค่า

งานวิชาการนี้... ทั้งหมดนี้... ให้แก่... และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MATERIAL:

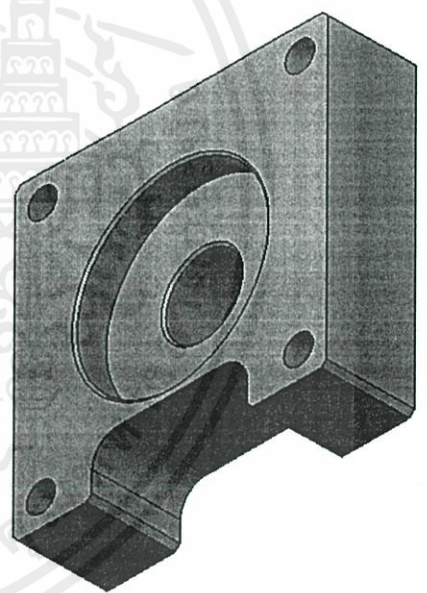
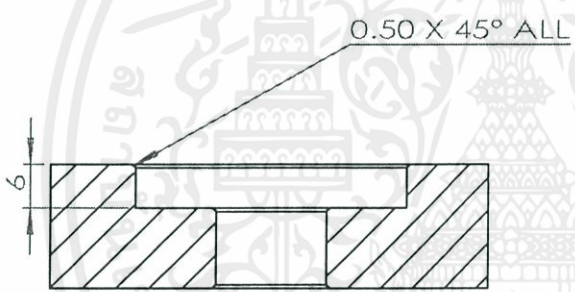
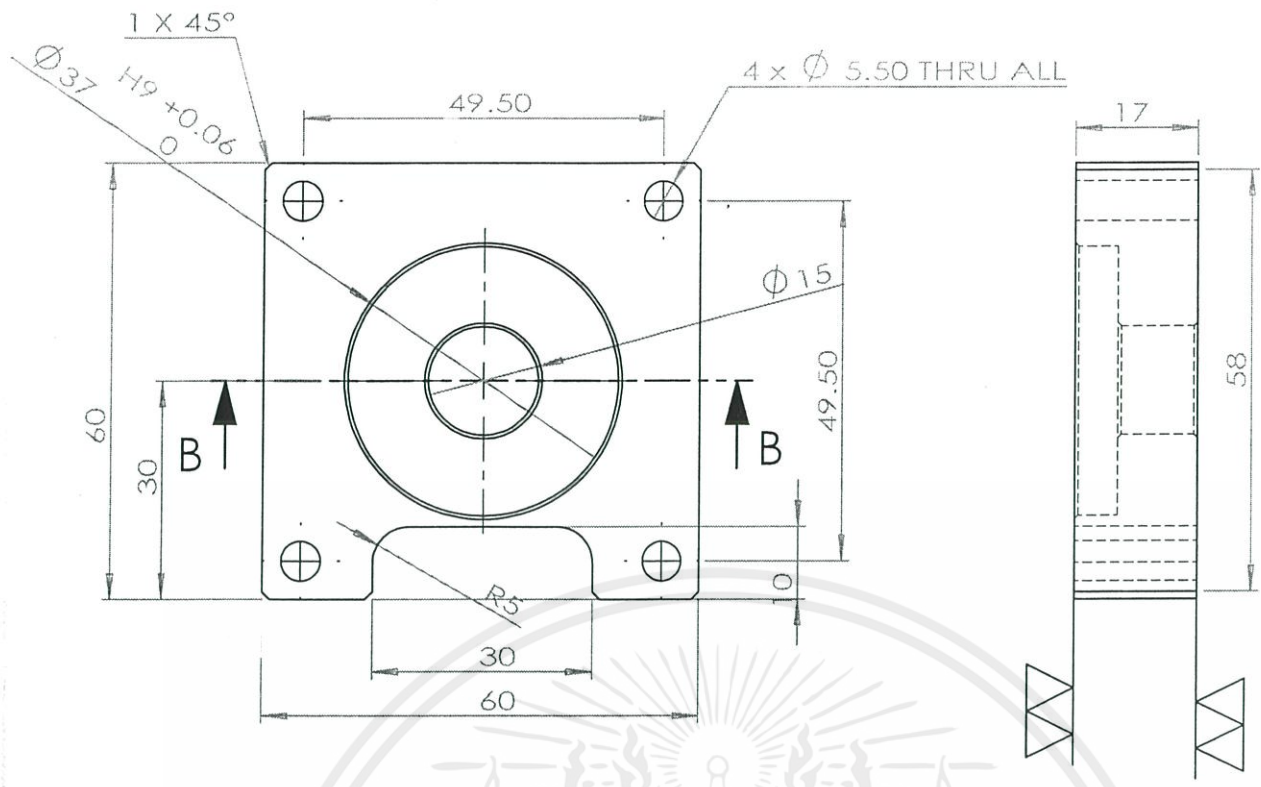
DWG NO.

=Bend dish one pin_circled A4

WEIGHT:

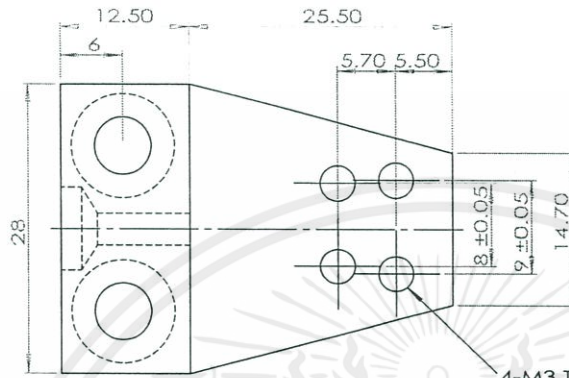
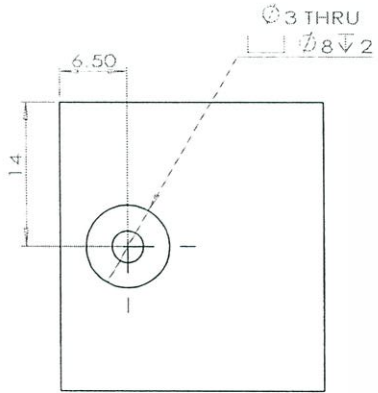
SCALE:2:1

SHEET 1 OF 1

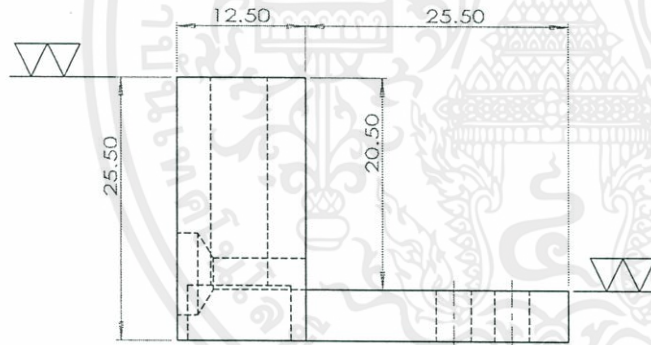


SECTION B-B
SCALE 1 : 1

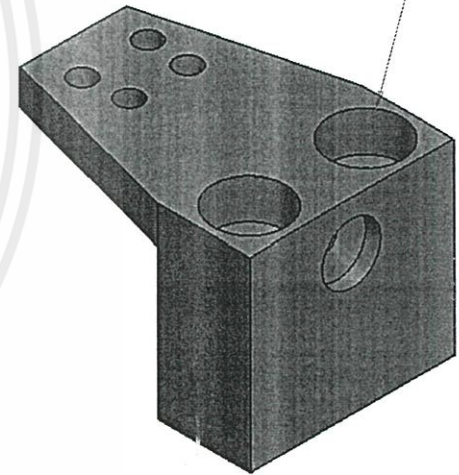
UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS		FINISH:	DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
TOLERANCES: LINEAR: ANGULAR:					
NAME	SIGNATURE	DATE	TITLE:	ไม่อนุญาตให้ใช้แบบหรือรูปถ่ายแบบใด ๆ ใด ๆ ในการผลิตโดยไม่ได้รับอนุญาตจากวิศวกรผู้ออกแบบ	
DRAWN				ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้	
CHK'D				DWG NO.	=support stepper1 A4
APPV'D				WEIGHT:	SCALE:1:2
MFG					SHEET 1 OF 1
Q.A					



4-M3 Thruall



$\varnothing 5.50$ THRU
 $R5 \nabla 5.40$



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS		FINISH:	DEBUR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING	REVISION
SURFACE FINISH:					
TOLERANCES:					
LINEAR:					
ANGULAR:					
DRAWN	NAME	SIGNATURE	DATE	TITLE:	
CHEKED					
APPVD					
MFG					
Q.A				MATERIAL:	
				DWG NO.	
				=bender head holed	
				SCALE:2:1	
				WEIGHT:	
				SHEET 1 OF 1	

5 Phase Stepper Motor

- Five leads for easy wiring configurations
- Wide selection range to suit various applications
- 0.3~63 KgcM Max. Holding Torque
- Compact size for high Torque performance
- Highly cost effective



5 Phase Stepper Motor

Type	Motor Size (mm)	Motor Model	Max. Holding Torque (kgcm)	Rotor Inertia (gcm ²)	Basic Step Angle***	Phase Current (Amps)	Motor Weight (kg)
HI-TORQUE (PF Series)	28	PEE 533 A (B)	0.33	9	0.72°	0.75	0.1
		PEE 535 A (B)	0.6	18	0.72°	0.75	0.17
		PF 543 AC (BC)	1.3	35	0.72°	0.75	0.25
	42	PF 544 AC (BC)	1.8	54	0.72°	0.75	0.3
		PF 545 AC (BC)	2.4	68	0.72°	0.75	0.4
		PF 564 AC (BC)	4.2	175	0.72°	1.4	0.6
		PF 566 AC (BC)	8.3	280	0.72°	1.4	0.8
	60	PF 569 AC (BC)	16.6	560	0.72°	1.4	1.3
		PF 596 AC (BC)	20.9	1400	0.72°	1.4	1.7
		PF 599 AC (BC)	40.9	2700	0.72°	1.4	2.8
85	PF 5913 AC (BC)	62.9	4000	0.72°	1.4	3.8	
HI-TORQUE HI-SPEED (PF Series)	60	PF 569 HAC (BC)	16.6	560	0.72°	2.8	1.3
		PF 596 HAC (BC)	20.9	1400	0.72°	2.8	1.7
	85	PF 599 HAC (BC)	40.9	2700	0.72°	2.8	2.8
		PF 5913 HAC (BC)	62.9	4000	0.72°	2.8	3.8

** Brake Type Stepper Motor **

HI-TORQUE (PF Series) *Brake Type	60	PF 564 ACM	4.2	320	0.72°	1.4	0.9
		PF 566 ACM	8.3	425	0.72°	1.4	1.1
		PF 569 ACM	16.6	705	0.72°	1.4	1.6
	85	PF 596 ACM	20.9	2200	0.72°	1.4	2.4
		PF 599 ACM	40.9	3500	0.72°	1.4	3.5
		PF 5913 ACM	62.9	4800	0.72°	1.4	4.5
HI-TORQUE HI-SPEED (PF Series) *Brake Type	60	PF 569 HACM	16.6	705	0.72°	2.8	1.6
		PF 596 HACM	20.9	2200	0.72°	2.8	2.4
	85	PF 599 HACM	40.9	3500	0.72°	2.8	3.5
		PF 5913 HACM	62.9	4800	0.72°	2.8	4.5

(A) after the unit # denotes single shaft configuration. (B) after the unit # denotes thru-shaft configuration.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Other Specifications

Motor Electrical Specifications

Dielectric Strength	No abnormality detected after the application of 0.5KV at 50 Hz between motor windings and frame for duration of one minute		
Insulation Resistance	100 Mohms or better with 500V potential applied between motor windings and frame at normal ambient temperature and humidity		
Insulation Class	Class B	Operating Environment Temperature	0°C ~ + 50°C

Motor Mechanical Specifications

Shaft Radial Play	0.025 mm (max) at load 0.5 Kg
Shaft Axial Play	0.075mm (max) at load 1 Kg
Step Angle Accuracy	± 3 min (PF Series)

Applicable Driver Range

The above motors can be used on the following MYCOM stepping drivers. Selection is generally based on applicable speed range and supply input as indicated below :-

Low Speed Range (24Vdc supply input)

UPS503 series
IMS500 series
INS500 series (NanoDrive)
INS501 series (NanoDrive)

Standard Speed Range (110Vac supply input)

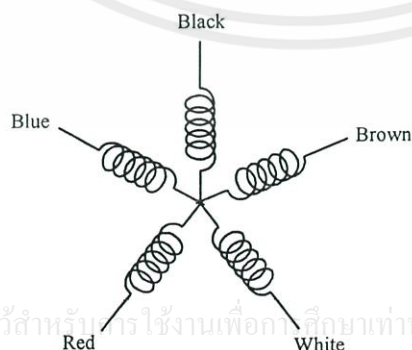
PSU50 series
UPS52 series
UPS53 series
IMS50-x1x series
INS50-x1x series (NanoDrive)

Standard Speed Range (220Vac supply input)

IMS50-x2x series
INS50-x2x series (NanoDrive)

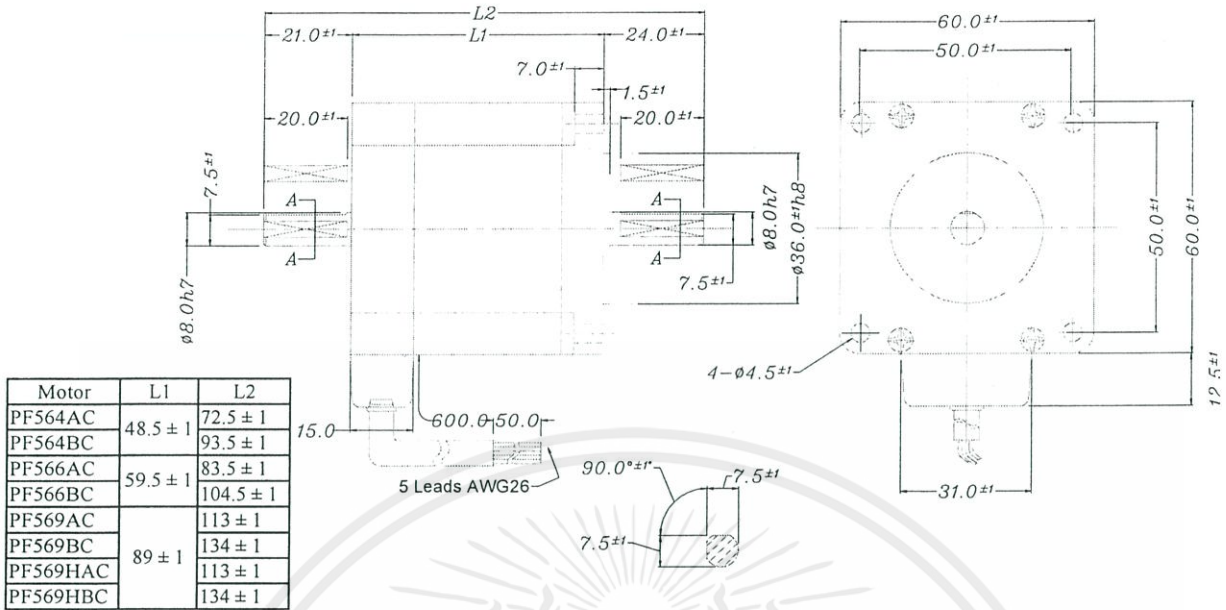
** kindly refer to our sales staff for other series of drivers

5 Phase Motor Connection Diagram

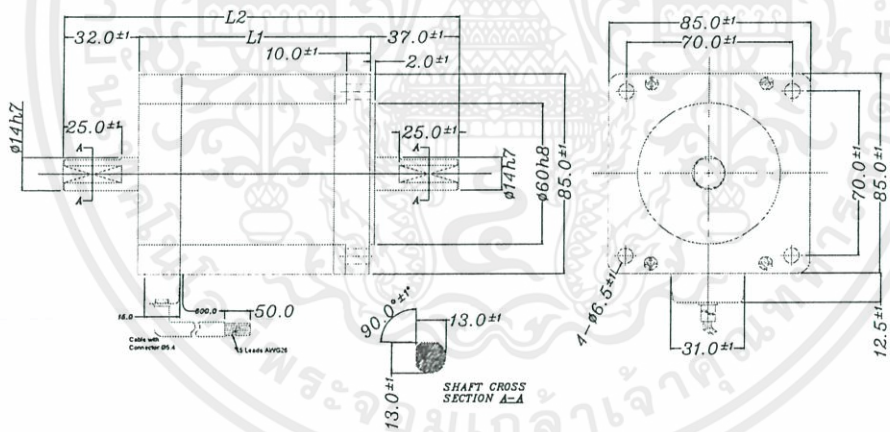


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 Phase Stepper Motor Dimensions – PF 564 / PF 566 / PF 569



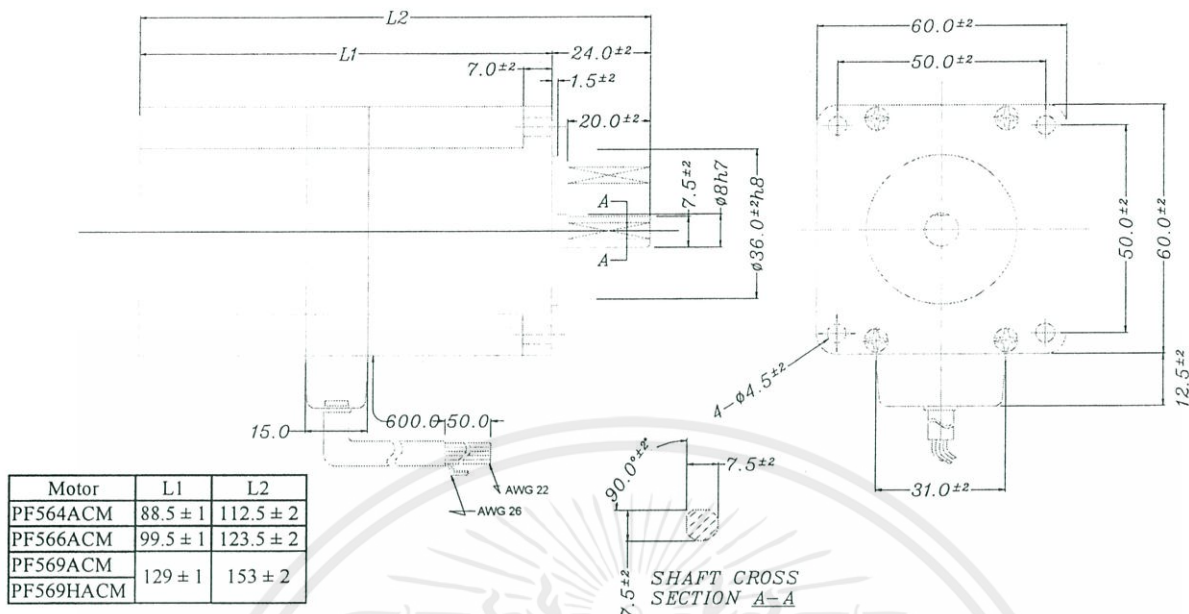
5 Phase Stepper Motor Dimensions – PF 596 / PF 599 / PF 5913



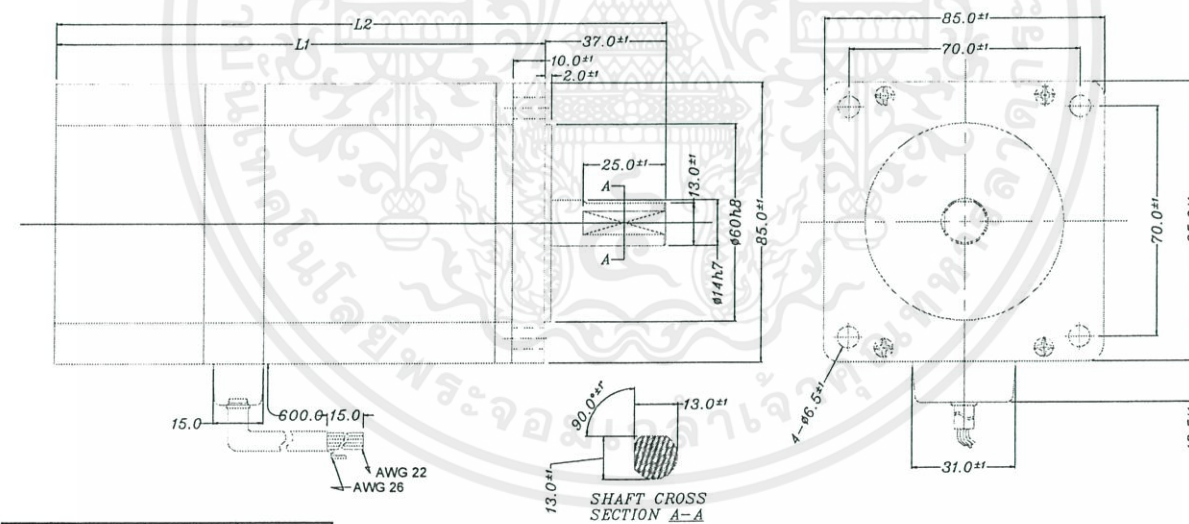
Motor	L1	L2
PF596AC	68 ± 1	105 ± 1
PF596BC		137 ± 1
PF596HAC		105 ± 1
PF596HBC		137 ± 1
PF599AC	98 ± 1	135 ± 1
PF599BC		167 ± 1
PF599HAC		135 ± 1
PF599HBC		167 ± 1
PF5913AC	128 ± 1	165 ± 1
PF5913BC		197 ± 1
PF5913HAC		165 ± 1
PF5913HBC		197 ± 1

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Brake Type Stepper Motor Dimensions – PF 56xACM / PF 569 HACM



Brake Type Stepper Motor Dimensions – PF 59xACM / PF 59x HACM



MYCOM, Inc.

12, S. Shimobano, Saga Hirosawa, Ukyo,
Kyoto 616, Japan

Tel : (81) 75-882-3601 Fax : (81) 75-880-6531

Email : support@mycom-japan.co.jp

MYCOM Technology (S) Pte Ltd

No 1 Sims Lane, #05-05 One Sims Lane
Singapore 387355

Tel : (065) 6743-4476 Fax : (065) 6743-4576

Email : sales@mycommts.com.sg

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและ
ข้อมูลใดๆที่มีมีการนำไปใช้

Control Modes

- Gearing, Position, Velocity, Torque

Command Interface

- Stepper commands
- ±10V velocity/torque command
- PWM velocity/torque command
- Master encoder [Gearing]

Communications

- RS232

Feedback

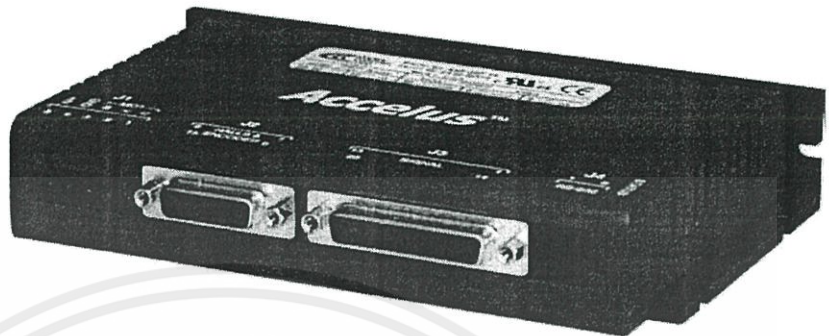
- Digital Quad A/B encoder
- Digital Halls

I/O - Digital

- 6 inputs, 2 outputs

Dimensions: mm [in]

- 168 x 99 x 31 [6.6 x 3.9 x 1.2]



MODEL	IC	IP	VDC
ASP-055-18	6	18	55
ASP-090-09	3	9	90
ASP-090-18	6	18	90
ASP-090-36	12	36	90
ASP-180-09	3	9	180
ASP-180-18	6	18	180

DESCRIPTION

Accelus is a 100% digital servoamplifier in a panel-mount package with a family of power options to 12A continuous and 36A peak. This new series offers sinusoidal commutation of brushless motors in torque, velocity, or position modes and fast, easy set up with CME 2™ software.

CME 2™ software communicates with Accelus through an RS-232 link for complete amplifier setup. Auto-phasing and auto-tuning algorithms in CME 2™ slash set up times for fast system commissioning and eliminate "re-wire and try" so common in brushless motor installations. CME 2™ automates current loop tuning, as well as motor, Hall, and encoder phasing. A powerful oscilloscope and signal generator display amplifier performance for fine tuning thereafter. Amplifier control parameters are saved in non-volatile flash memory. OEM's can inventory one part, and configure amplifiers on-site to each axis in a machine.

Advanced field-oriented-control ensures the highest motor torque over a wide speed range, minimizing motor heating and maximizing efficiency. Digital control algorithms transform stator currents into direct and quadrature components. The torque-producing quadrature current is controlled by the current loop, and the direct component is driven to zero eliminating losses from current that doesn't produce torque. Space-vector modulation produces higher speeds than sine-pwm modulation from the same power supply.

Accelus works with motion controllers that close position-loops using incremental encoder feedback and process the position error in a PID filter to produce an amplifier command for torque, force, or velocity. Only one ±10V analog, or a one or two-wire digital PWM/DIR control signal is required. All commutation is done in the amplifier.

In position-mode, Accelus accepts two-wire digital step-motor control signals (CW/CCW, or Count/Direction), or operates as a slave from a master encoder. The ratio between input position pulses and motor position is programmable.

Velocity control is derived from motor encoder signals. Velocity mode is useful not only for speed-setpoint applications, but enables operation with PLC's or controllers that output position-error signals with no PID filtering.

All amplifier circuits are DC coupled and operate from unregulated transformer-isolated DC power supplies, or regulated switching power supplies.

The panel-mount package is compatible with the mounting footprint of Copley's 5xx1 series trapezoidal brushless amplifiers, offering an easy upgrade to sinusoidal commutation with digital control.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GENERAL SPECIFICATIONS

Test conditions: Load = 3-phase Wye connected load, 2 mH line-line. Ambient temperature = 25 °C. +HV = HVmax

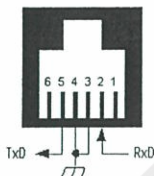
MODEL	ASP-055-18	ASP-090-09	ASP-090-18	ASP-090-36	ASP-180-09	ASP-180-18	
OUTPUT POWER							
Peak Current	18 (12.7)	9 (6.4)	18 (12.7)	36 (25.5)	9 (6.4)	18 (12.7)	Adc (Arms)
Peak time	1	1	1	1	1	1	1
Continuous current	6 (4.2)	3 (2.1)	6 (4.2)	12 (8.5)	3 (2.1)	6 (4.2)	Adc (Arms) per phase
Peak Output Power	0.92	0.79	1.55	2.95	1.59	3.15	kW
Continuous Output Power	0.32	0.27	0.53	1.0	0.53	1.06	kW
INPUT POWER							
HV _{min} to HV _{max}	+20 to +55	+20 to +90	+20 to +90	+20 to +90	+20 to +180	+20 to +180	Vdc, Transformer-isolated
Peak current	20	10	20	40	10	20	Adc (1 sec) peak
Continuous current	6.7	3.3	6.7	13.3	3.3	6.7	Adc continuous
PWM OUTPUTS							
Type	3-phase MOSFET inverter, 20 kHz center-weighted PWM, space-vector modulation						
PWM ripple frequency	40 kHz						
COMMUTATION & CONTROL							
Current loop	20 kHz (50 μs period) update rate						
Velocity & position loop	4 kHz (250 μs period) update rate						
Commutation	Sinusoidal, field-oriented control of DC brushless motor						
Phase Initialization	Amplifier initializes in trapezoidal commutation until a Hall transition occurs, then switches to sinusoidal commutation with phase-correction at each Hall signal transition						
BANDWIDTH							
Current loop, small signal	3 kHz typical, bandwidth will vary with tuning & load inductance						
HV Compensation	Changes in HV do not affect bandwidth						
REFERENCE INPUTS (NOTE 1)							
Analog torque/velocity	±10Vdc, 12 bit resolution	Differential (J3-2,14)					
Input impedance	66 kΩ	Between Ref(+), Ref(-)					
Digital position reference	Pulse/Direction, CW/CCW	Stepper commands (2 MHz maximum rate)					
	Quad A/B Encoder	20 Mcount/sec (after quadrature), 5 Mline/sec					
Digital torque & velocity reference	PWM, Polarity	PWM = 0~100%, Polarity = 1/0					
	PWM	PWM = 50% +/-50%, no polarity signal required					
	PWM frequency range	1 kHz minimum, 100 kHz maximum					
	PWM minimum pulse width	220 ns					
DIGITAL INPUTS (NOTE 1)							
All inputs	74HC14 Schmitt trigger operating from 5.0 Vdc with RC filter on Input						
Logic levels	Vin-LO < +1.35 Vdc, Vin-HI > +3.65 Vdc, Input voltage range 0 to +24 Vdc						
Pull-up, pull-down control	[IN2,3,4] have group selectable connection of 10 kΩ input pull-up/down resistor to +5 Vdc or ground						
Enable [IN2]	Dedicated input with 330us RC filter for amplifier enable, active level programmable						
GP [IN1,2,3,4]	General Purpose inputs with 330us RC filter and active level select, [IN1,3,4] have programmable functions						
HS [IN5,6]	High-Speed Inputs inputs with 100ns RC filter, active level select, and programmable functions						
SERIAL DATA INPUT							
RS-232	RxD, TxD, Gnd in 6-position, 4-contact RJ-11 type modular connector, and on J3-24, 25, & 13 Full-duplex, serial communication port for amplifier setup and control, 9600 to 115200 baud Data protocol: binary						
MOTOR CONNECTIONS							
Phase U, V, W	Amplifier outputs to 3-phase Wye or delta connected brushless motors with floating neutral						
Hall U, V, W	Digital Hall signals						
Encoder A, /A, B, /B, (X, /X)	Quadrature encoder signals, single-ended or differential (X or Index signal not required)						
	5 MHz maximum line frequency (20 Mcount/s) when driven from active differential outputs						
[IN1] Motemp	See Digital Inputs (above) for details (Note 1)						
STATUS INDICATORS							
Amp Status	Bicolor LED. Amplifier status indicated by color, and blinking or non-blinking condition						
DIGITAL OUTPUTS (NOTE 1)							
Type	Current-sinking MOSFET open-drain output with 1kΩ pullup to +5 V through diode, 1 Adc sink max, +30 Vdc max.						
[OUT1,2]	Programmable External flyback diode required with inductive loads						
PROTECTIONS							
HV Overvoltage	+HV > HV _{max}	Amplifier outputs turn off until +HV < HV _{max} (See Input Power for HV _{max})					
HV Undervoltage	+HV < 20 ^{min} Vdc	Amplifier outputs turn off until +HV > 20 ^{min} Vdc					
Amplifier over temperature	PC Board > 70 °C.						
Short circuits	Output to output, output to ground, internal PWM bridge faults						
I ² T Current limiting	Programmable: continuous current, peak current, peak time						
Motor over temperature	Digital inputs programmed for overtemp function will disable amplifier						
MECHANICAL & ENVIRONMENTAL							
Size	6.58 in (167,1 mm) X 3.89 in (98,81 mm) X 1.17 in (29,72 mm)						
Weight	0.94 lb (0.43 kg) for amplifier without heatsink						
Ambient temperature	0 to +45 °C operating, -40 to +85 °C storage						
Humidity	0% to 95%, non-condensing						
Contaminants	Pollution degree 2						
Environment	IEC68-2: 1990						
Cooling	Heat sink and/or forced air cooling may be required for continuous power output (see pg. 8 & 9)						

Notes
1. Digital input & output functions are programmable. Default functions are shown here.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COMMUNICATION

Accelus is configured via a three-wire, full-duplex RS-232 port that operates from 9600 to 115,200 Baud. CME 2™ provides a graphic user interface (GUI) to set up all of Accelus features via a computer serial port. Connections to the Accelus RS-232 port P1 are via an RJ-11 style connector, and through the signal connector J3 (J3-24 & 25). Rx/D, Tx/D, and Gnd signals comprise the signals supported. The Accelus Serial Cable Kit contains a modular cable, and an adapter that connects to a 9-pin, Sub-D serial port connector (COM1, COM2, etc.) on PC's and compatibles.



RS-232 Port

STATUS LED

A single bi-color LED gives the state of the amplifier by changing color, and either blinking or remaining solid. The possible color and blink combinations are:

- **Green/Solid:** Amplifier OK and enabled. Will run in response to reference input.
- **Green/Slow-Blinking:** Amplifier OK but not-enabled. Will run when enabled.
- **Green/Fast-Blinking:** Positive or Negative limit switch active. Amplifier will only move in direction not inhibited by limit switch.
- **Red/Solid:** Temporary fault condition. Amplifier will resume operation when fault is removed.
- **Red/Blinking:** Latching fault. Operation will not resume until amp is Reset

Fault conditions: Short-circuits from output to output, output to ground, and internal shorts or over current conditions, amplifier or motor over-temperature, over or under-voltage, encoder power loss, motor phasing error (current position > 60° electrical from Hall angle), or position-mode following error. Faults are programmable as latching or non-latching.

DIGITAL INPUTS

There are six digital inputs [IN1-6] five of which are programmable functions. Inputs [IN1, 5, & 6] have 10 kΩ pull-up resistors that connect to +5 Vdc to work with grounded switches, NPN open-collector, CMOS, or TTL outputs.

[IN2] always functions as the Enable input, and controls the ON/OFF state of the amplifier outputs. [IN2] can function simply as the amp-enable or as the amp-enable with reset. With the reset options selected, the amplifier will reset when [IN2] goes from the active to the inactive level. The default selection is active-LO with no reset. This setting is the fail-safe condition. In order to make the amplifier operate, the enable input must be connected and must be grounded to operate the amplifier. If a wire were to break, or the controller malfunction, the input would not be grounded and the amplifier would not operate. If the input is set to Active-HI, it is not in a fail-safe mode, and will be enabled with no connection to the [IN2] input. This setting is therefore not

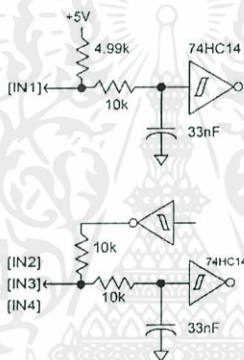
recommended for general operation.

The other digital enable inputs, [IN1], [IN3-6] have alternate functions that are settable via CME 2™:

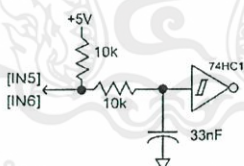
- Positive Limit Switch
- Negative Limit Switch
- Amplifier Reset
- Motor temp sensor

In addition to the alternate functions, the active level for each input is individually programmable.

Amplifier reset takes place on transitions of the input and is programmable to 1/0 or 0/1. The motor temp sensor function will disable the amplifier if a switch in the motor opens or closes when the motor overheats. The motor temperature switch or sensor should be grounded. The active-level setting is then set depending on the type of switch: normally open, or normally closed.



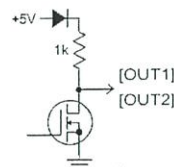
Input resistors are programmable to pull-up to +5 Vdc or pull-down to signal ground as a group for [IN2,3,4].



DIGITAL OUTPUTS

Digital outputs are open-drain MOSFETs with 1 kΩ pull-up resistors to +5 Vdc. These can sink up to 1 Adc from external loads operating from power supplies to +30 Vdc. When driving inductive loads such as a motor brake, an external fly-back diode is required. A diode in the output is for driving PLC inputs that are opto-isolated and connected to +24 Vdc. The diode prevents conduction from +24 Vdc through the 1 kΩ resistor to +5 Vdc in the amplifier. This could turn the input on, giving a false indication of the amplifier output state. These outputs are level-selectable like the inputs, and each output is programmable for various functions.

Digital outputs



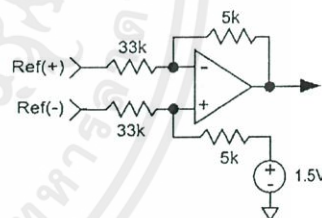
REFERENCE INPUTS

The Reference inputs command the amplifier to produce an output. Accelus has analog and digital reference inputs. Only one type can be active at a time.

ANALOG REFERENCE INPUT

The analog ±10 Vdc signal is an industry standard for torque or velocity control. The analog reference input is a differential amplifier which is to be connected to the motion controller ground and DAC output. Using a differential amplifier is important because there may be potential differences between the amplifier and controller grounds. The differential amplifier construction rejects these differences and measures the controller output relative to ground at the controller. Note that the voltage between Ref(+) and Ref(-) inputs must be zero to produce a "zero" amplifier command. Grounding Ref(-), and allowing Ref(+) to be open will produce a large command, as will grounding Ref(+) and letting Ref(-) be open.

When wiring the controller DAC output to the reference inputs, be sure to use both reference inputs, and connect Ref(-) to ground at the controller, and not at the amplifier.



IMPORTANT!

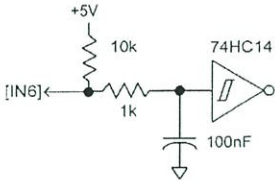
ALWAYS CONNECT BOTH ANALOG REF INPUTS. THERE MUST BE ZERO VOLTS BETWEEN REF(+) & REF(-) FOR ZERO OUTPUT FROM THE AMPLIFIER!

DIGITAL REFERENCE INPUTS

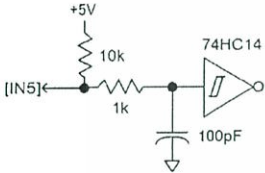
There are two logic inputs [IN5] and [IN6] for digital reference signals that are programmable for controlling torque, velocity, or position. If not used as reference inputs, they are programmable as logic inputs. When used as reference inputs these should be driven by active-output devices (i.e. CMOS, TTL). The input resistors define the default polarities when inputs are open, or for use with open-collector devices as digital logic inputs.

DIGITAL REFERENCE INPUTS (CONT'D)

PWM Torque/velocity input
or CW/Pulse/Encoder-A position input



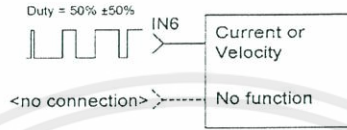
Torque/Velocity polarity
or CCW/Direction/Encoder-B position input



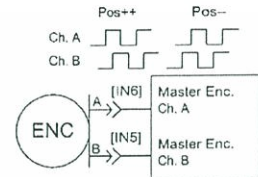
Duty-cycles of 0%, and 100% would result in negative full-scale, or positive full-scale outputs. So, the duty-cycle controls not only the magnitude, but also the polarity of the amplifier outputs.

The scale-factor for amplifier-output vs. PWM inputs is settable via CME 2™ software in both cases.

50% PWM signals for
torque/velocity reference



Master encoder signals as
position reference inputs



IMPORTANT!

AMPLIFIERS WITH SERIAL NUMBERS EQUAL TO OR LESS THAN THE NUMBERS SHOWN HERE HAVE THE INPUT CONFIGURATION SHOWN BELOW FOR [IN6]

ASP-090-09, ASP-180-09: s/n 5102xxxx
ASP-055-18, ASP-090-18, ASP-180-18: 3102xxxx
ASP-090-36: 3502xxxx

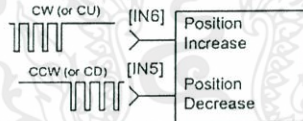
NOTE: THE LAST FOUR DIGITS ARE NOT SIGNIFICANT

When operating Accelus in position mode, the digital reference inputs accept step-motor pulses in two formats, or quadrature-encoder signals. In either case, the ratio between input pulses, and motor encoder counts is programmable.

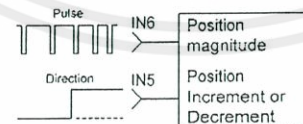
The first format of stepper-command signals is the CW/CCW (clockwise/counter-clockwise) format, which could also be called CU/CD (count-up/count-down). Pulses at [IN6] will increase the position-command to the amplifier, and pulses at [IN5] will decrease it. The other stepper-command format is the Pulse & Direction one where pulses at [IN6] will increment or decrement the position-command depending on the DC level at the Direction input, [IN5].

The active-edge is programmable for 0-1 or 1-0 transitions when in Pulse/Dir or CW/CCW input modes.

Digital reference inputs configured as CW/CCW inputs in position mode with 0-1 active-edges:



Digital reference inputs configured as Pulse-Direction inputs in position mode with 0-1 active-edges:



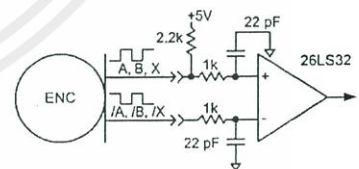
Digital reference inputs can also connect to a quadrature encoder that outputs two pulse trains corresponding to angle of rotation, or linear travel. The pulse trains are phase-shifted 90° which gives them the name "quadrature", and each time one changes, the amplifier can interpret the change as an incremental position command. The amplifier decodes the A and B channel signals to determine if it is an increase, or decrease in position. If the encoder is mounted on a motor that is controlled by

MOTOR CONNECTIONS

Motor connections are of three types: phase, Halls, and encoder. The phase connections carry the amplifier output currents that drive the motor to produce motion. The Hall signals are three digital signals that give absolute position feedback within an electrical commutation cycle. The encoder signals give incremental position feedback and are used for velocity and position modes, as well as sinusoidal commutation.

MOTOR ENCODER

The input circuit for the motor encoder signals is a differential line-receiver with R-C filtering on the inputs. The circuit is shown below. Encoders with differential outputs are preferred because they are less susceptible to noise that can be picked on single-ended outputs. For best results, encoder cabling should use twisted pair cable with one pair for each of the encoder outputs: A & /A, B & /B, and X & /X. Shielded twisted-pair is best for noise rejection. If single-ended encoders are used they should connect to J2-6, 7, 8. Encoders that require 200 mA or less can operate from amplifier +5 Vdc on J2-11. If more current is required then +5 Vdc from the user system can connect to J3-22 and will be available on J2-10. The encoder input circuit is shown below.



MOTOR HALL SIGNALS

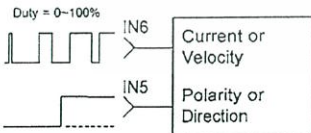
Hall signals are single-ended signals that provide absolute feedback within one electrical cycle of the motor. There are three of them (U, V, & W) and they may be sourced by magnetic sensors in the motor, or by encoders that have Hall tracks. They typically operate at much lower frequencies than the motor encoder signals, and in Accelus they are used for commutation-initialization after power-on, and for checking the motor phasing after the amplifier has switched to sinusoidal commutation.

DIGITAL REFERENCE INPUTS

For torque or velocity control, the inputs may be configured in two formats:

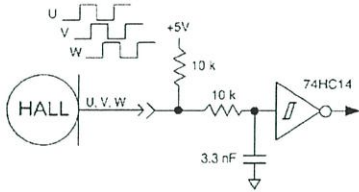
1. PWM (0~100%) & Polarity
2. PWM (50%)

In the first case, the PWM signal can vary from 0% to 100%, and the Polarity signal is a DC level that controls the direction of the motor. The PWM duty-cycle controls the amplifier output current, or motor velocity. In current mode, 100% corresponds to the maximum output current. In velocity mode, it commands the maximum velocity that is configured.



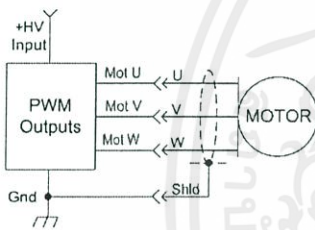
Another type of PWM input is the "50%" type. There is only one PWM signal that connects to [IN6]. The other digital input [IN5] is not used in this mode. A 50% duty-cycle corresponds to a zero-current command in torque mode, or a zero-speed output in velocity mode.

MOTOR HALL SIGNALS (CONT'D)



MOTOR PHASE CONNECTIONS

The amplifier output is a three-phase PWM inverter that converts the DC buss voltage (+HV) into three sinusoidal voltage waveforms that drive the motor phase-coils. The three phase currents I_u , I_v , and I_w sum to zero so individual conductors in the motor cable should be sized for the continuous current rating of the amplifier. Motor cabling should use twisted, shielded conductors for CE compliance, and to minimize PWM noise coupling into other circuits. The motor cable shield should connect to motor frame and the amplifier HV ground terminal (J1-2) for best results.



GROUNDING CONSIDERATIONS

All circuits in Accelus share a common circuit ground (Gnd on J1-4, and Signal Ground on J2-5,12, and J3-10,12,13,15,16). Input logic circuits are referenced to Signal Ground, as are analog Reference inputs, digital outputs, encoder and Hall signals. For this reason, amplifier Gnd terminals should connect to the users' common ground system so that signals between amplifier and controller are at the same common potential, and to minimize noise. The system ground should, in turn, connect to an earthing conductor at some point so that the whole system is referenced to "earth".

Because current flow through conductors produces voltage-drops across them, it is best to connect the amplifier HV return (Gnd) to system earth, or circuit-common through the shortest path, and to leave the power-supply floating. In this way, the power supply (-) terminal connects to ground at the amplifier Gnd terminals, but the voltage drops across the cables will not appear at the amplifier ground, but at the power supply negative terminal where they will have less effect.

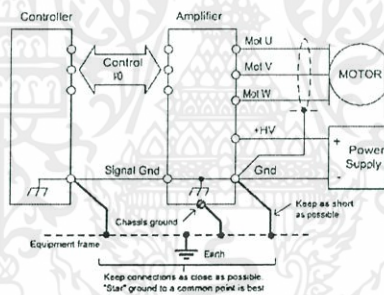
Motor phase currents are balanced, but currents can flow between the PWM outputs, and the motor cable shield. To minimize the effects of these currents on nearby circuits, the cable shield should connect to Gnd (J1-2).

The amplifier case does not connect to any amplifier circuits. Connections to the case are provided on connectors J2-1, and J3-1. Cables to these connectors should be shielded for CE compliance, and the shields should connect to these terminals. When installed, the amplifier case should connect to the system chassis. This maximizes the shielding effect of the case, and provides a path to ground for noise currents that may occur in the cable shields.

Signals from controller to amplifier are referenced to +5Vdc, and other power supplies in user equipment. These power supplies should also connect to system ground and earth at some point so that they are at same potential as the amplifier circuits.

The final configuration should embody three current-carrying loops. First, the power supply currents flowing into and out of the amplifier at the +HV and Gnd pins on J1. Second the amplifier outputs driving currents into and out of the motor phases, and motor shield currents circulating between the U, V, and W outputs and Gnd. And, lastly, logic and signal currents connected to the amplifier control inputs and outputs.

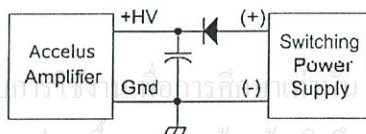
For CE compliance and operator safety, the amplifier should be earthed by using external tooth lockwashers under the mounting screws. These will make contact with the aluminum chassis through the anodized finish to connect the chassis to the equipment frame ground.



POWER SUPPLIES

Accelus operates typically from transformer-isolated, unregulated DC power supplies. These should be sized such that the maximum output voltage under high-line and no-load conditions does not exceed the amplifiers maximum voltage rating. Power supply rating depends on the power delivered to the load by the amplifier. In many cases, the continuous power output of the amplifier is considerably higher than the actual power required by an incremental motion application.

Operation from regulated switching power supplies is possible if a diode is placed between the power supply and amplifier to prevent regenerative energy from reaching the output of the supply. If this is done, there must be external capacitance between the diode and amplifier.



MOUNTING & COOLING

Accelus has slots for mounting to panels at 0° or 90°. Cooling is by conduction from amplifier heatplate to mounting surface, or by convection to ambient.

FUSING & PROTECTIONS

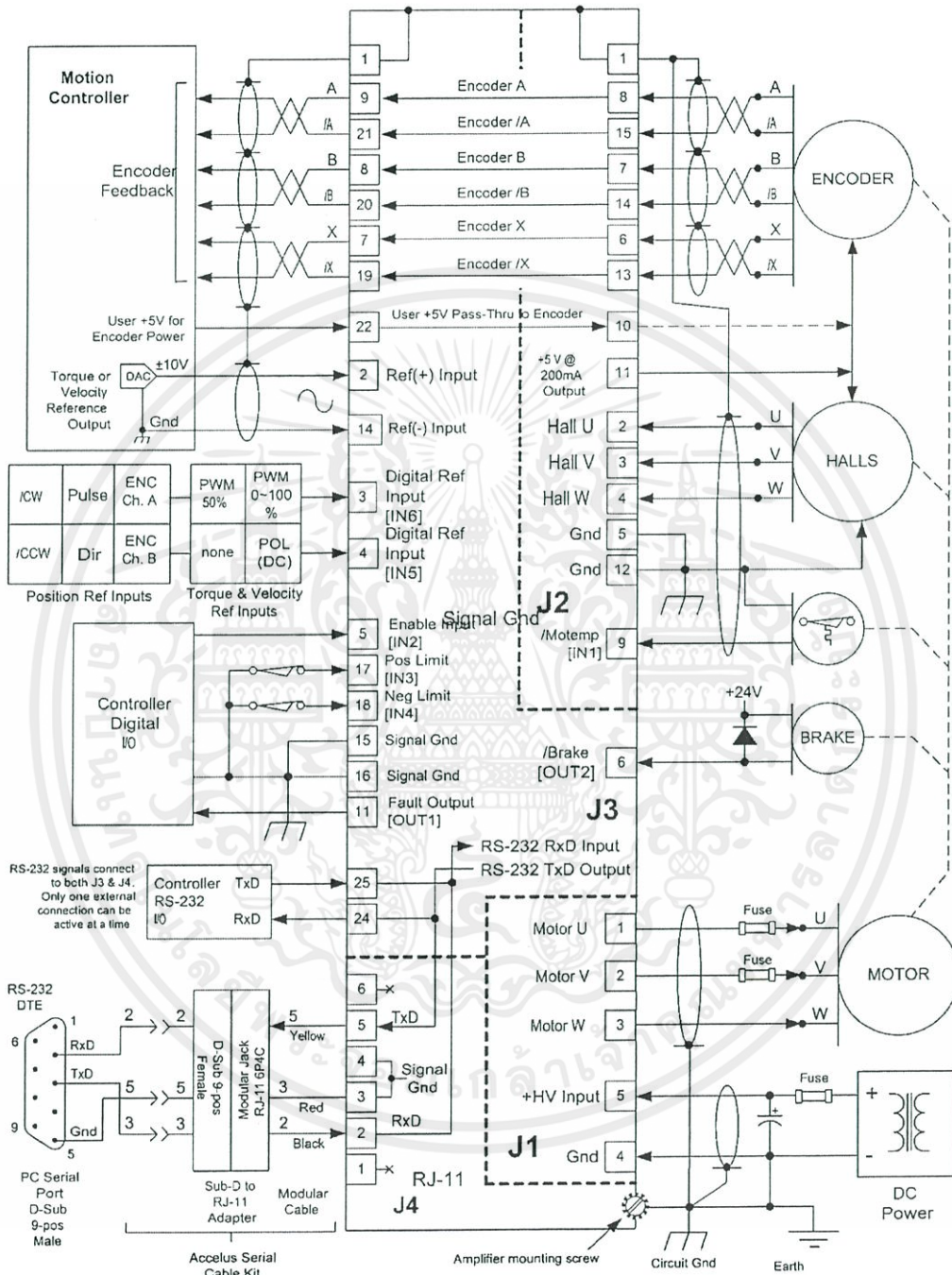
Fusing of input power connections to Accelus protects external circuits from an uncontrolled failure of the amplifier. Fusing of motor phase connections provides protection of the motor from overcurrents due to either mis-configuration of the amplifier, or uncontrolled failure of the amplifier. Motor phase fusing is recommended for linear motor applications as this type of motor has a lower thermal capacity, in general, than rotary motors due to the mounting of the phase coils in epoxy "fins". This factor, plus the higher cost factors associated with linear motor installations (mounting and accessibility, time to repair, initial cost) make the use of phase-fuses advisable as the ultimate protection for such motors. In many applications, Accelus will have peak and continuous ratings greater than the motors' ratings, so operator errors during setup or adjustments can easily damage the motor with little stress to the amplifier.

Depending on the application, fast-acting fuses may be the fuse of choice for motor phase protection. Typically, two fuses provide sufficient protection for motor currents as the phase currents sum to zero. If grounding of the motor winding is possible, then three fuses would provide complete protection for any overcurrents flowing out of the amplifier and into the motor.

Sizing of motor phase fuses should take into account the peak and rms current over the anticipated duty cycle of the motor, and motor ratings. The final value selected should be tested in the equipment to prove that no false tripping occurs under worst-case temperature and operating-current conditions.

The maximum input current to the amplifier should not exceed its rated output currents under normal operating conditions. Time-delay fuses typically carry 75% of their rating continuously, so choosing a fuse that can carry 33% more current than the continuous current rating of the amplifier should be prevent false tripping. The DC voltage rating should be sufficient to carry the amplifier +HV operating voltage.

AMPLIFIER CONNECTIONS



NOTES

1. The function of input signals on J2-9, and J3-3,4,17,18 and output signals on J3-6, 11 are programmable. Default functions are shown.

2. The function of [IN2] on J3-5 is always amplifier Enable and is not programmable.

3. The active level of [IN2,3, and 4] is programmable as a group to pull-up or pull-down.

เอกสารนี้เพื่อใช้ประกอบการศึกษาเท่านั้น ไม่ควรนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AMPLIFIER CONNECTORS

- J1: +HV, Gnd, & Motor U,V,W Outputs
Molex/Beau 5,08mm, Eurostyle®, 5-position receptacle.
- J2: Motor Encoder & Halls
Sub-D, 15-position, female. #4-40 standoffs for cable connector shell screws.
- J3: Amplifier Signals
Sub-D, 25-position, female. #4-40 standoffs for cable connector shell screws.
- J4: Modular receptacle. RJ-11 style. 6-position, 4-contact.

MATING (CABLE) CONNECTORS

- P1: Molex/Beau 5,08mm, Eurostyle®, 5-position terminal block 860505-00, or equivalent.
 - P2: Sub-D, 15-position, male, #4-40 locking screws.
 - P3: Sub-D, 25-position, male, #4-40 locking screws.
 - P4: Modular plug. RJ-11 style. 6-position, 4-contact.
- Note: Connector Kit ASP-CK contains one each of P1, P2, and P3.
P4 is part of the cable assembly in Serial Cable Kit SER-CK

J4 SIGNAL	PIN
No Connection	1
RxD Input	2
Ground	3
Ground	4
TxD Output	5
No Connection	6

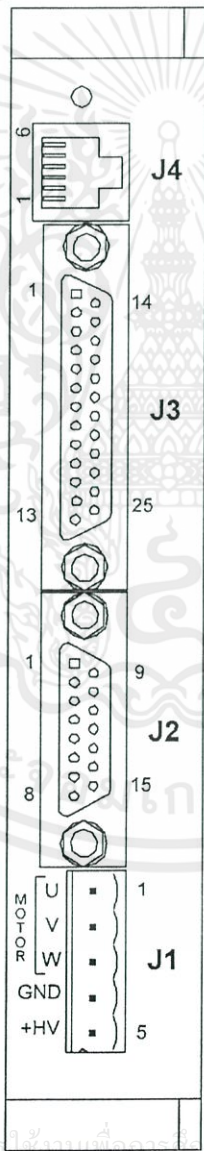
J3 SIGNAL	PIN
Chassis Ground	1
Ref(+) Input	2
PWM Ref Input [IN6]	3
Direction Input [IN5]	4
Enable Input [IN2]	5
Motor Brake Output [OUT2]	6
Encoder X Output	7
Encoder B Output	8
Encoder A Output	9
Signal Ground	10
Fault Output [OUT1]	11
Signal Ground	12
Signal Ground	13

J2 SIGNAL	PIN
Chassis Ground	1
Hall U Input	2
Hall V Input	3
Hall W Input	4
Signal Ground	5
Encoder X Input	6
Encoder B Input	7
Encoder A Input	8

PIN	J3 SIGNAL
14	Ref(-) Input
15	Signal Ground
16	Signal Ground
17	[IN3] Forward Enable Input
18	[IN4] Reverse Enable Input
19	Encoder /X Output
20	Encoder /B Output
21	Encoder /A Output
22	User +5V Pass-Thru
23	+5V @ 200 mA
24	RS-232 TxD Output
25	RS-232 RxD Input

PIN	J2 SIGNAL
9	[IN1] Motor Temp Sensor
10	User +5V Pass-Thru
11	+5V @ 200 mA
12	Signal Ground
13	Encoder /X Input
14	Encoder /B Input
15	Encoder /A Input

PIN	J1 SIGNAL
1	Motor U Output
2	Motor V Output
3	Motor W Output
4	GND
5	+HV Input



NOTES

1. The function of signals on J2-9, and J3-3, 4, 6, 11, 17,18 are programmable. Default functions are shown.
2. The function of [IN2] on J2-5 is always Amplifier Enable and is not programmable. The active level of [IN2] is programmable, and amplifier Reset with enable is programmable.

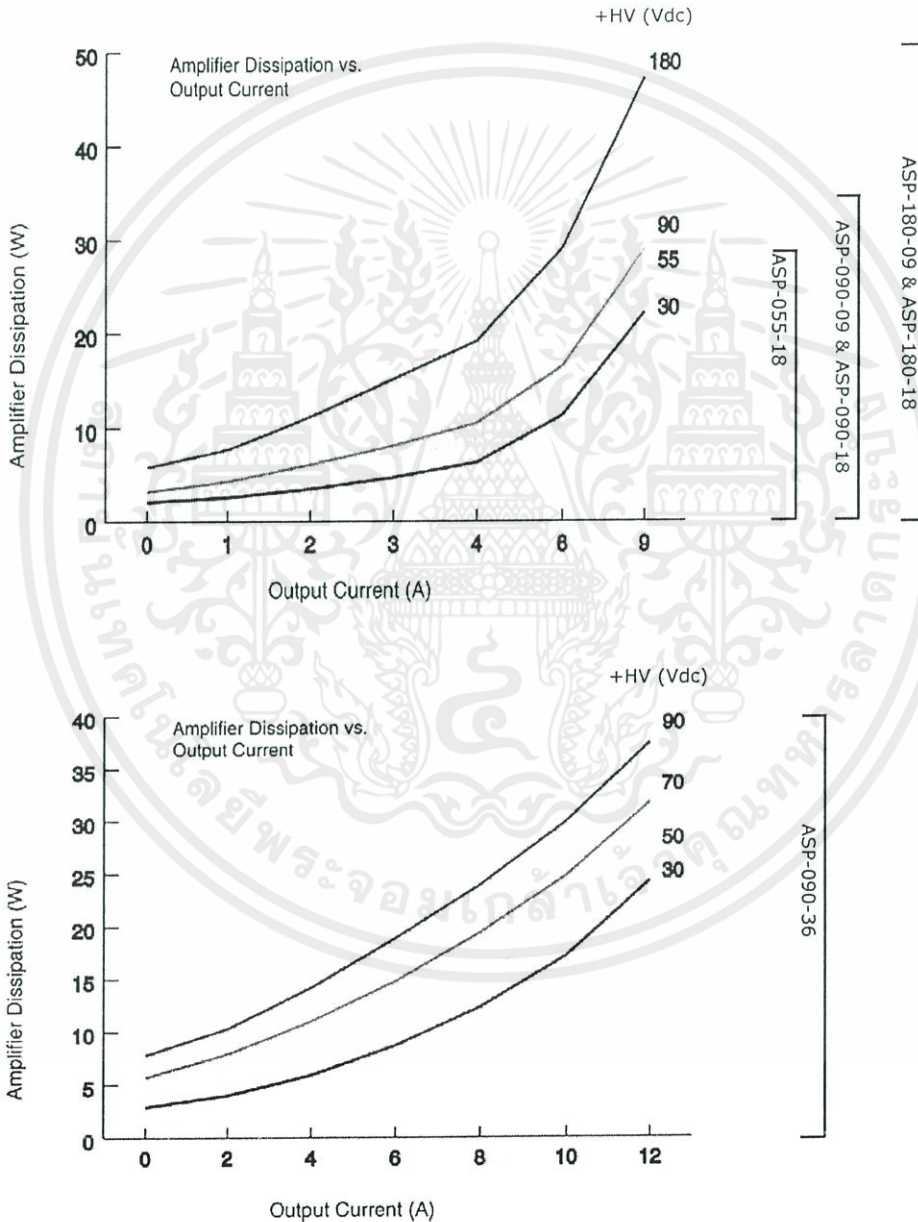
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการที่บริษัทมีไว้ใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POWER DISSIPATION

The charts on this page show the amplifier internal power dissipation for the *Accelus* models under differing power supply and output current conditions. Amplifier output current is calculated from the motion profile, motor, and load conditions. The values on the chart represent the rms (root-mean-square) current that the amplifier would provide during operation. The +HV values are for the average DC voltage of the amplifier power supply.

When +HV and amplifier output current are known, the amplifier power dissipation can be found from the chart. Once this is done use the data on the facing page to find amplifier thermal resistance. From this calculate the maximum ambient operating temperature. If this result is lower than the known maximum ambient temperature then a mounting with a lower thermal resistance must be used.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

MOUNTING

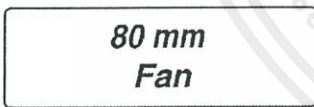
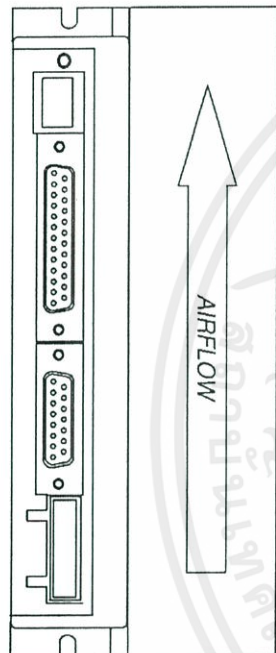
Thermal data for convection-cooling with a heatsink assumes a vertical mounting of the amplifier on a thermally conducting surface. Heatsink fins run parallel to the long axis of the amplifier. When fan-cooling is used vertical mounting is not necessary to guarantee thermal performance of the heatsink.

THERMAL RESISTANCE

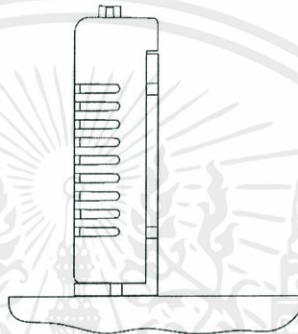
Thermal resistance is a measure of the temperature rise of the amplifier heatplate due to power dissipation in the amplifier. It is expressed in units of °C/W where the degrees are the temperature rise above ambient.

E.g., an amplifier dissipating 20 W mounted with no heatsink or fan would see a temperature rise of 28 °C above ambient based on the thermal resistance of 1.4 °C/W. Using the amplifier maximum heatplate temperature of 70 °C and subtracting 28 °C from that would give 42 °C as the maximum ambient temperature the amplifier in which the amplifier could operate before going into thermal shutdown.

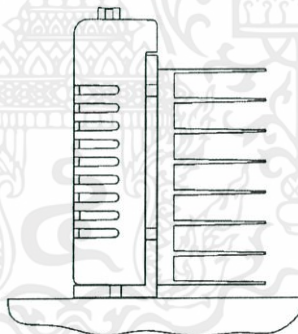
TOP VIEW, VERTICAL MOUNTING



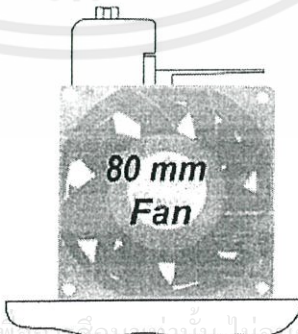
END VIEWS
VERTICAL MOUNTING



NO HEATSINK, NO FAN	°C/W
CONVECTION	2.9



HEATSINK, NO FAN	°C/W
CONVECTION	1.7



HEATSINK + FAN	°C/W
FORCED-AIR, 300 LFM	0.6

Note: Fans are customer-supplied

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

E40 Series

Diameter ϕ 40mm Shaft type/Hollow type/Built-in type Incremental Rotary encoder

Line-up

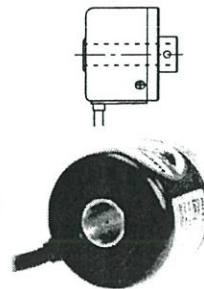
Features

- 12-24VDC power supply of line driver output(Line-up)
- Easy installation at narrow space
- Small moment of inertia
- Power supply : 5VDC, 12-24VDC \pm 5%
- Various output types

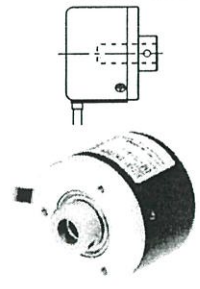
⚠ Please read "Caution for your safety" in operation manual before using.



E40S SERIES



E40H SERIES



E40HB SERIES

Ordering information

E40	H	8	5000	3	N	24	
Series	Shaft type	Hollow type	Pulse/1Revolution	Output phase	Output	Power supply	Cable
S: Shaft type	External	Inner	Refer to resolution	2:A, B	T: Totem pole output	5 : 5VDC \pm 5% 24: 12-24VDC \pm 5%	Blank: Normal type (*) C: Cable outgoing connector type
H: Hollow type		6: ϕ 6mm		3:A, B, Z	N: NPN open collector output		
HB: Hollow built-in type	6: ϕ 6mm 8: ϕ 8mm	8: ϕ 8mm 10: ϕ 10mm 12: ϕ 12mm		4:A, \bar{A} , B, \bar{B}	V: Voltage output		
				6:A, \bar{A} , B, \bar{B} , Z, \bar{Z}	L: Line driver output		

※ Standard : E40S6-PULSE-3-N-24
E40H8-PULSE-3-N-24
E40HB8-PULSE-3-N-24

※ Standard: A, B, Z

※ Cable length : 250mm

Specifications

Item	Diameter ϕ 40mm shaft/hollow shaft/hollow built-in type of incremental rotary encoder		
Resolution(P/R)	(Note1) *1. *2, *5, 10, *12, 15, 20, 23, 25, 30, 35, 40, 45, 50, 60, 75, 100, 120, 150, 192, 200, 240, 250, 256, 300, 360, 400, 500, 512, 600, 800, 1000, 1024, 1200, 1500, 1800, 2000, 2048, 2500, 3000, 3600, 5000 (Not indicated resolution is customizable.)		
Electrical specification	Output phase	A, B, Z phase (Line driver : \bar{A} , \bar{A} , B, \bar{B} , Z, \bar{Z} phase)	
	Phase difference of output	Phase difference between A and B : $\frac{T}{4} \pm \frac{T}{8}$ (T=1 cycle of A phase)	
	Control output	Totem pole output	• Low \Rightarrow Load current: Max. 30mA, Residual voltage : Max. 0.4VDC • High \Rightarrow Load current: Max. 10mA, Output voltage (Power supply 5VDC): Min. (Power supply-2.0)VDC, Output voltage (Power supply 12-24VDC): Min. (Power supply-3.0)VDC
		NPN open collector output	Load current : Max. 30mA, Residual voltage : Max. 0.4VDC
		Voltage output	Load current : Max. 10mA, Residual voltage : Max. 0.4VDC
		Line driver output	Low \Rightarrow Load current : Max. 20mA, Residual voltage : Max. 0.5VDC High \Rightarrow Load current : Max. -20mA, Output voltage : Min. 2.5VDC
	Response time (Rise/Fall)	Totem pole output	Max. 1 μ s
		NPN open collector output	Max. 1 μ s
		Voltage output	Max. 1 μ s
		Line driver output	Max. 0.5 μ s
	Max. Response frequency	300kHz	
	Power supply	• 5VDC \pm 5% (Ripple P-P : Max. 5%) • 12-24VDC \pm 5% (Ripple P-P : Max. 5%)	
	Current consumption	Max. 80mA (disconnection of the load), Line driver output : Max. 50mA (disconnection of the load)	
Insulation resistance	Min. 100M Ω (at 500VDC megger between all terminals and case)		
Dielectric strength	750VAC 50/60Hz for 1 minute (Between all terminals and case)		
Connection	Cable outgoing type, 250mm cable outgoing connector type		
Mechanical specification	Starting torque	Shaft Type : Max. 40gf \cdot cm (0.004N \cdot m), Hole Type : Max. 50gf \cdot cm (0.005N \cdot m)	
	Moment of inertia	Max. 40g \cdot cm ² (4×10^{-6} kg \cdot m ²)	
	Shaft loading	Radial : Max. 2kgf, Thrust : Max. 1kgf	
	Max. allowable revolution	(Note2) 5000rpm	
Vibration	1.5mm amplitude at frequency of 10 to 55Hz in each of X, Y, Z directions for 2 hours		
Shock	Max. 50G		
Ambient temperature	-10 to 70 $^{\circ}$ C (at non-freezing status), Storage : -25 to 85 $^{\circ}$ C		
Ambient humidity	35 to 85%RH, Storage : 35 to 90%RH		
Protection	IP50 (IEC standard)		
Cable	ϕ 5mm, 5P, Length : 2m, Shield cable (Line driver output : ϕ 5mm, 8P)		
Accessory	• Shaft type : ϕ 6mm coupling standard, ϕ 8mm coupling (Sold separately) • Hole type : Bracket		
Approval	CE (Except for line driver output)		
Unit weight	Approx. 160g		

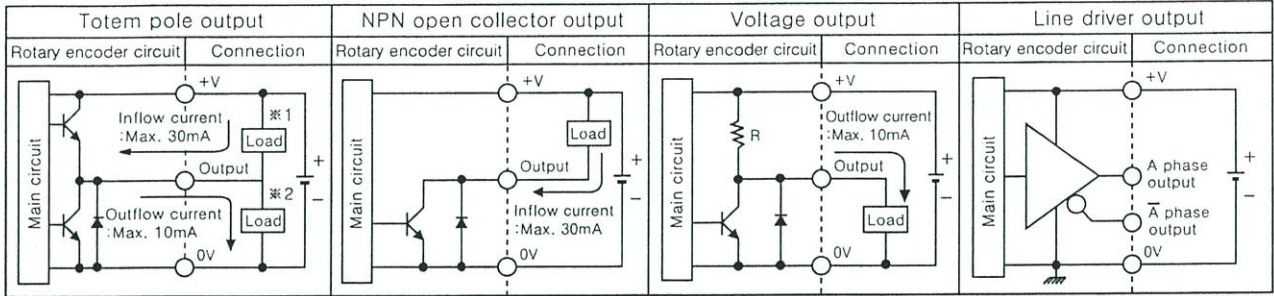
※ (Note1) '*' pulse is only for A, B phase (Line Driver output is for A, \bar{A} , B, \bar{B} phase)

※ (Note2) Max. allowable revolution \geq Max. response revolution [Max. response revolution (rpm) = $\frac{\text{Max. response frequency}}{\text{Resolution}} \times 60 \text{ sec.}$]

Make sure that max. response revolution should be lower than max. allowable revolution when selecting the resolution.

Incremental ϕ 40mm Shaft/Hollow Shaft/Built-in Type

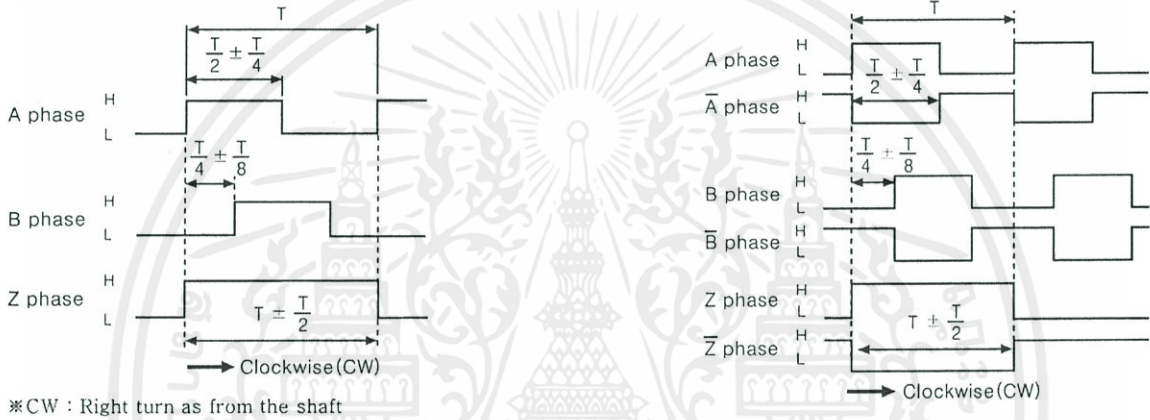
Control output diagram



- Totem pole output type can be used for NPN open collector output type (※1) or Voltage output type (※2).
- All output circuits of A, B, Z phase are the same. (Line driver output is A, \bar{A} , B, \bar{B} , Z, \bar{Z})

Output waveform

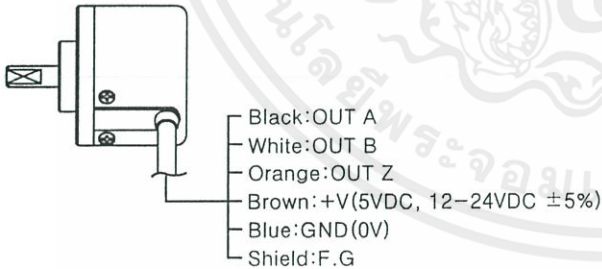
- Totem pole output / NPN open collector output / Voltage output
- Line driver output



Connections

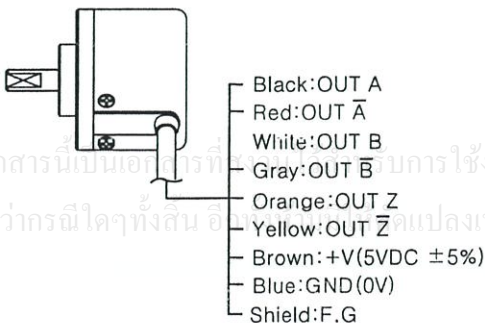
Normal type

- Totem pole output / NPN open collector output / Voltage output



- ※ Unused wires must be insulated.
- ※ The metal case and shield wire of encoder should be grounded (F.G).

Line driver output



Cable outgoing connector type

- Totem pole output
NPN open collector output
Voltage output
- Line driver output



Totem pole output NPN open collector output Voltage output			Line driver output		
Pin No	Function	Cable color	Pin No	Function	Cable color
①	OUT A	Black	①	OUT A	Black
②	OUT B	White	②	OUT \bar{A}	Red
③	OUT Z	Orange	③	+V	Brown
④	+V	Brown	④	GND	Blue
⑤	GND	Blue	⑤	OUT B	White
⑥	F.G	Shield	⑥	OUT \bar{B}	Gray
			⑦	OUT Z	Orange
			⑧	OUT \bar{Z}	Yellow
			⑨	F.G	Shield

※ F.G (Field Ground): It should be grounded separately.

- (I) Photo eye/IR sensor
- (J) Fiber optic sensor
- (K) Back Area sensor
- (L) Proximity sensor
- (M) Pressure sensor
- (N) Rotary encoder
- (O) Connector/Socket
- (P) Temp. controller
- (Q) SER/Power controller
- (R) Counter
- (S) Timer
- (T) Panel meter
- (U) Tacho/Speed/Pulse meter
- (V) Display unit
- (W) Sensor controller
- (X) Switching power supply
- (Y) Stepping motor & Driver & Controller
- (Z) Graphic/Logic panel
- (AA) Field network device
- (AB) Production stoppage models & replacement

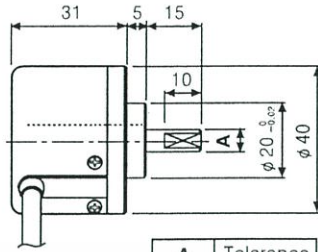
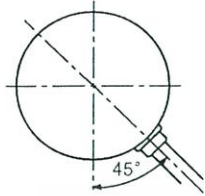
E40 Series

Dimensions

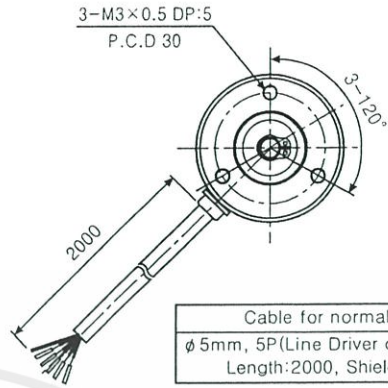
Normal type

(Unit:mm)

Shaft type

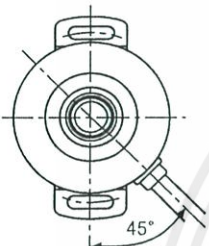


A	Tolerance
φ 6	-0.01 -0.015
φ 8	-0.01 -0.02

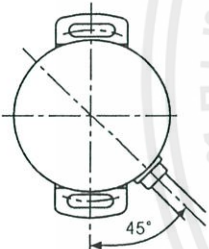


Cable for normal type
φ 5mm, 5P(Line Driver output : 8P),
Length:2000, Shield cable

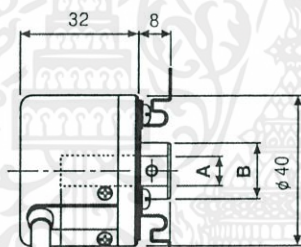
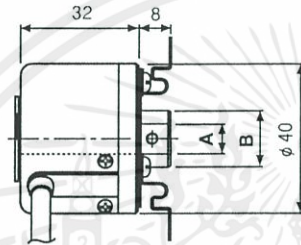
Hollow shaft / Hollow shaft built-in type



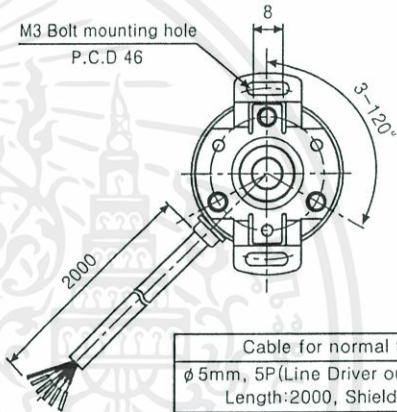
E40H



E40HB



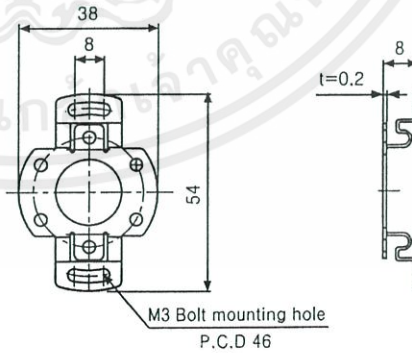
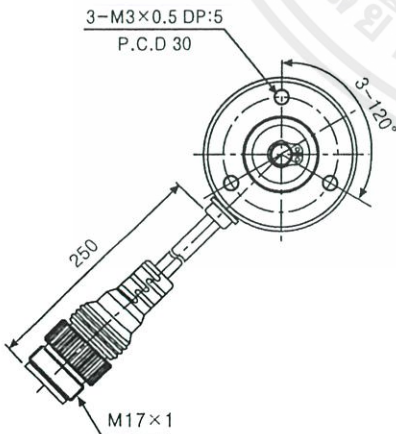
A	φ 6	φ 8	φ 10	φ 12
Tolerance			+0.015 0	
B	φ 15		φ 17	



Cable for normal type
φ 5mm, 5P(Line Driver output : 8P),
Length:2000, Shield cable

Cable outgoing connector type

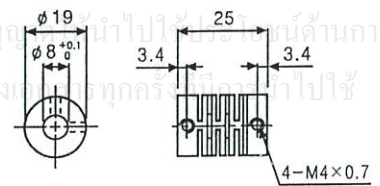
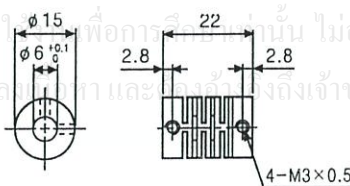
Bracket(E40H, E40HB)



Coupling(E40S)

φ 6 Coupling

φ 8 Coupling



※Connector cable is customizable and see G-6 for specifications.

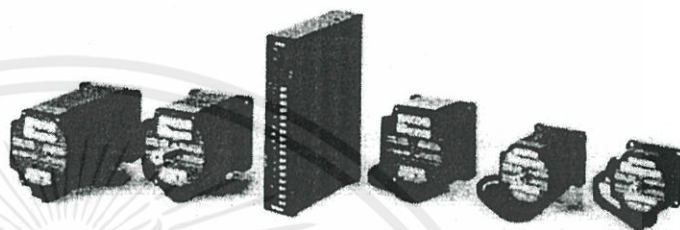
UPS 52 PULSTAR V BRAKE TYPE SERIES 5-Phase Stepping Motors and Drivers*

with Integral Fail-Safe Electro Magnetic Brake

High Torque Version

A New Generation of 5-Phase Units Employ Rare Earth Magnet Technology to Offer the Highest Performance Stepping Motor/Driver Package Available.

- Very High Torque in a Compact Package.
- 500 (Full) / 1000 (Half) Mechanical Steps per Revolution.
- Virtually No Low Speed Resonance.
- Fast brake activation time (20ms).
- The brake can be used to gain additional static holding torque.



* Protected by US/Intl. Patents

UPS 52 Series (A) after the unit # denotes single shaft configuration. (B) after the unit # denotes thru-shaft configuration.

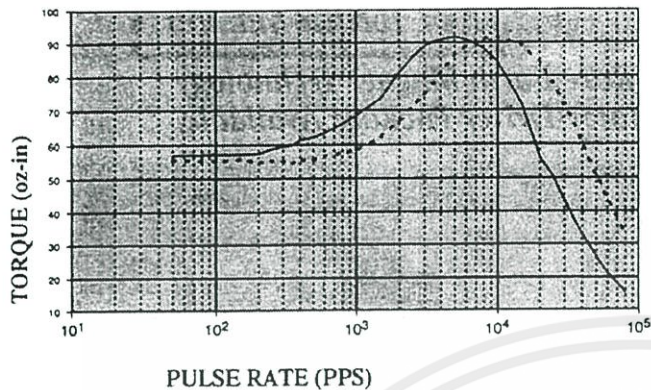
MOTOR SIZE (mm)	MOTOR AND DRIVER COMBINATION UNIT #	MAX. HOLDING TORQUE oz-in (kg-cm)	ROTOR INERTIA oz-in ² (g cm ²)	STEP ANGLE FULL/HALF	PHASE CURRENT (A)	MOTOR WEIGHT oz (kg)	DRIVER TYPE	MOTOR TYPE
23 (60)	UPS52-564 ACM	58.3 (4.2)	0.87 (175)	0.72°/0.36°	1.4	17.6 (0.5)	UPS52-130	PF 564 ACM
	UPS52-566 ACM	115.3 (8.3)	1.4 (280)	0.72°/0.36°	1.4	26.4 (0.75)	UPS52-130	PF 566 ACM
	UPS52-569 ACM	230.5 (16.6)	2.8 (560)	0.72°/0.36°	1.4	45.8 (1.3)	UPS52-130	PF 569 ACM
34 (85)	UPS52-596 ACM	291.6 (20.9)	7.0 (1400)	0.72°/0.36°	1.4	52.9 (1.5)	UPS52-130	PF 596 ACM
	UPS52-599 ACM	568 (40.9)	13.5 (2700)	0.72°/0.36°	1.4	88.2 (2.5)	UPS52-130	PF 599 ACM
	UPS52-5913 ACM	873.5 (62.9)	20 (4000)	0.72°/0.36°	1.4	123.4 (3.5)	UPS52-130	PF 5913 ACM

Other Specifications

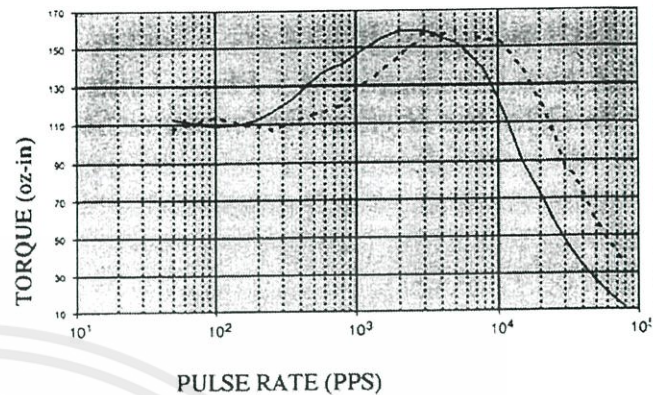
Motor		Driver	
Step Angle Accuracy	± 3 min.	UPS 52 - 130	
Shaft Radial Play	0.025 mm max. / load 0.5 Kg	Drive Type	Star Bi-Polar. Constant Current Chopper
Shaft Axial Play	0.075 mm max. / load 1 Kg	Power Requirements	115 ~ 120V AC. 50/60 Hz
Dielectric Strength	No abnormality detected after the application of 0.5KV at 50Hz between motor windings and frame for duration of one minute.	Power Consumption	4 Amp Max.
Insulation Class	Class B	Built In Functions	Auto current reduction at stand still. Current off. Excitation timing signal output. Driver overheat warning signal output.
Insulation Resistance	100M ohms or better with 500V potential applied between motor windings and frame at normal ambient temperature and humidity.	Input Pulse Types	STEP and DIRECTION or CW/CCW signal (selectable). All opto isolated. Voltage H = 4 ~ 5V. L = 0 ~ 0.5V. Resistance 220 ohms.
Operating Environment Temperature	0°C ~ +50°C	Dielectric Strength	No abnormality detected after the application of AC 1KV between case and power input terminals. case and signal I/O terminals. signal I/O terminals and power input terminals for duration of one minute.
		Insulation Resistance	100M ohms or better with 500V potential applied between case and power input terminals and signal I/O terminals.
		Operating Environment Temperature	0°C ~ + 40°C

Performance Characteristics

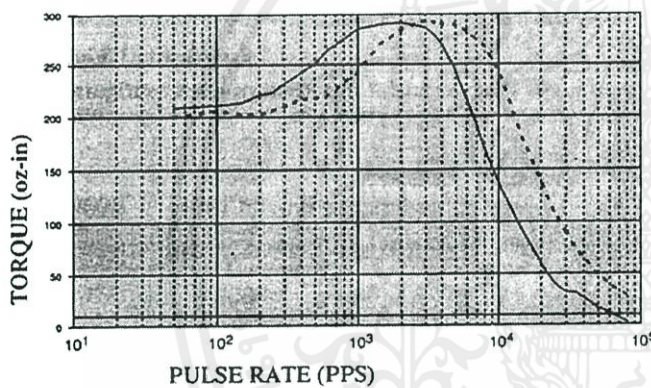
UPS52-564



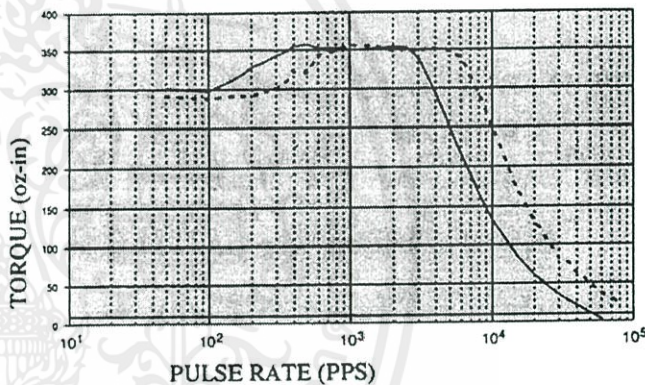
UPS52-566



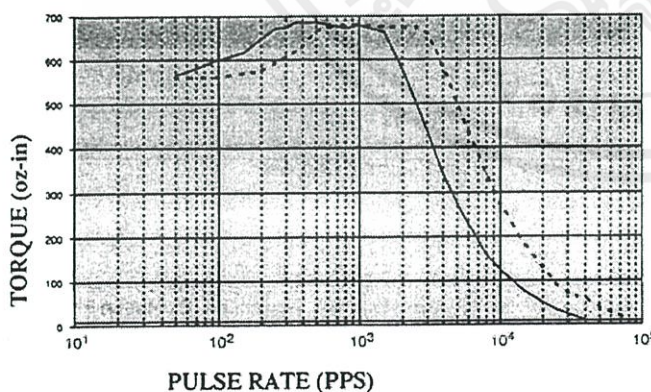
UPS52-569



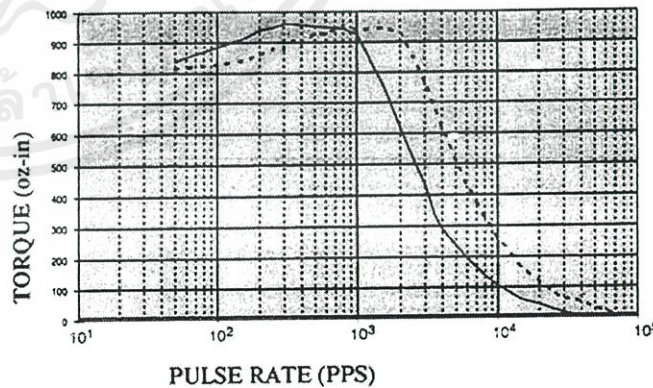
UPS52-596



UPS52-599



UPS52-5913

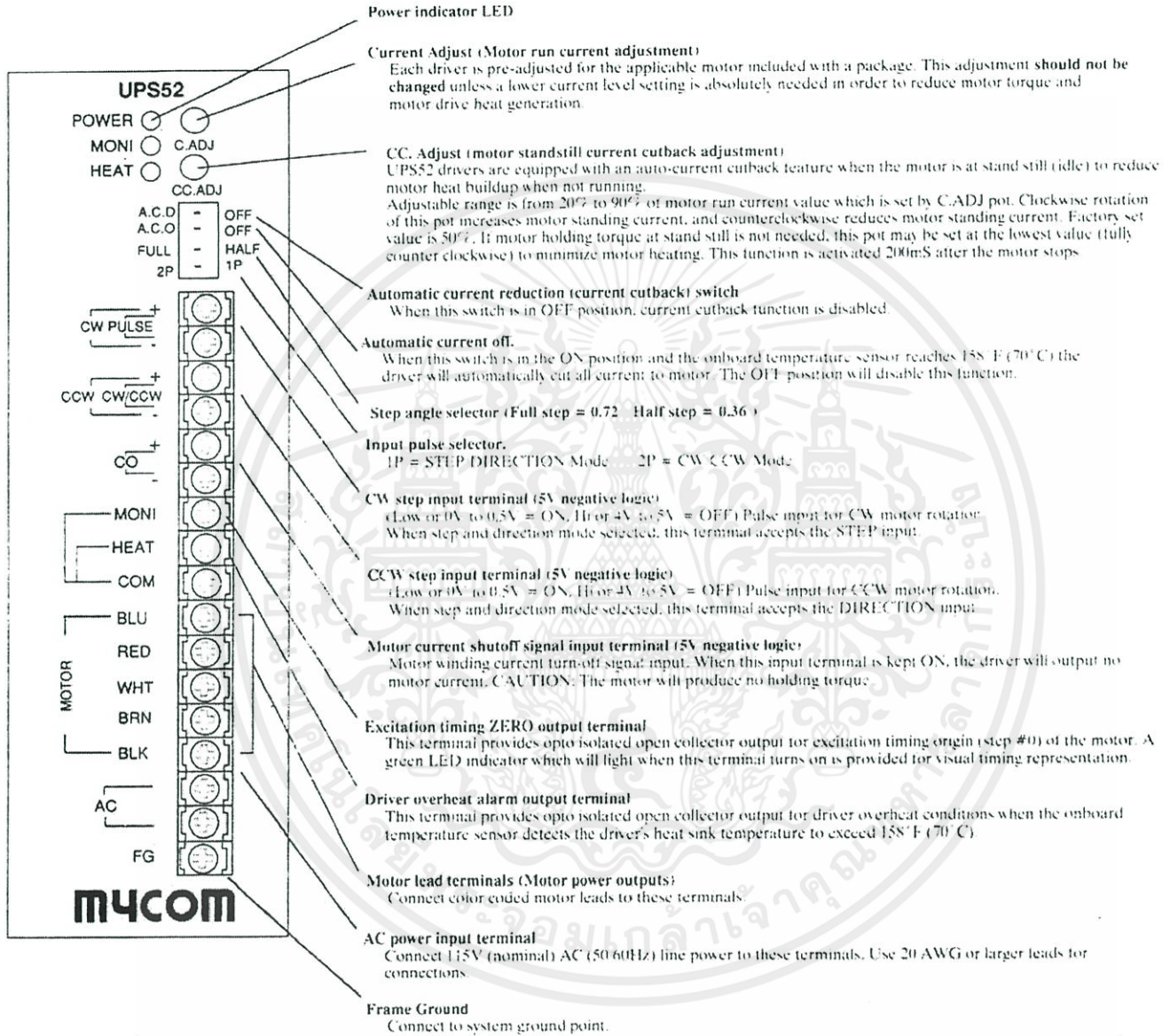


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึง

————— FULL STEP
..... HALF STEP

Driver Data

UPS52 Driver Functions



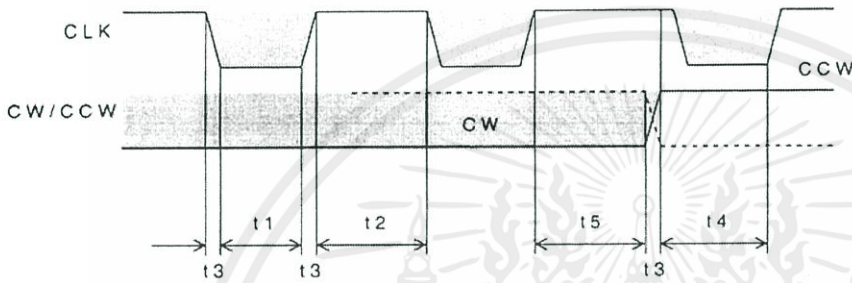
CAUTION: TO AVOID INJURY BY ELECTRICAL SHOCK & DAMAGE, DO NOT TOUCH MOTOR LEAD TERMINALS WHEN THE POWER IS ON! MAXIMUM MOTOR CHOPPING VOLTAGE WILL APPROACH 140VDC. NO CONNECTION CHANGES MAY BE MADE WHILE THE DRIVE POWER IS ON. HAZARDOUS VOLTAGES MAY BE PRESENT AT THESE TERMINALS. IF THE MOTOR OUTPUTS ARE SHORT CIRCUITED, SERIOUS DAMAGE TO THE DRIVER WILL RESULT. THIS WILL NOT BE COVERED UNDER WARRANTY, AND WILL VOID REMAINDER OF THE ORIGINAL PRODUCT WARRANTY. IF MOTOR LEAD TO TERMINAL CONNECTION IS TO BE MADE VIA A CONNECTOR SET, DO NOT USE MALE CONNECTORS OF ANY TYPE TO TERMINATE DRIVER OUTPUTS TO THE MOTOR. ALWAYS USE FEMALE CONNECTOR ON DRIVER OUTPUT SIDE TO AVOID POSSIBLE SHORT CIRCUIT ACCIDENTS.

Electromagnetic Brake Specifications:

	PF56XACM	PF59XXACM
Brake Coil Voltage:	DC 24V ± 5%	
Current Consumption	0.25A	0.42A
Brake Holding Torque	8 kgcm	40 kgcm
Brake Activation Time	20 msec max	20 msec max
Brake Release Time	30 msec max	50 msec max
Duty cycle	Continuous	

* This is not a speed control brake
 * Motor must NOT be started before the brake is released.

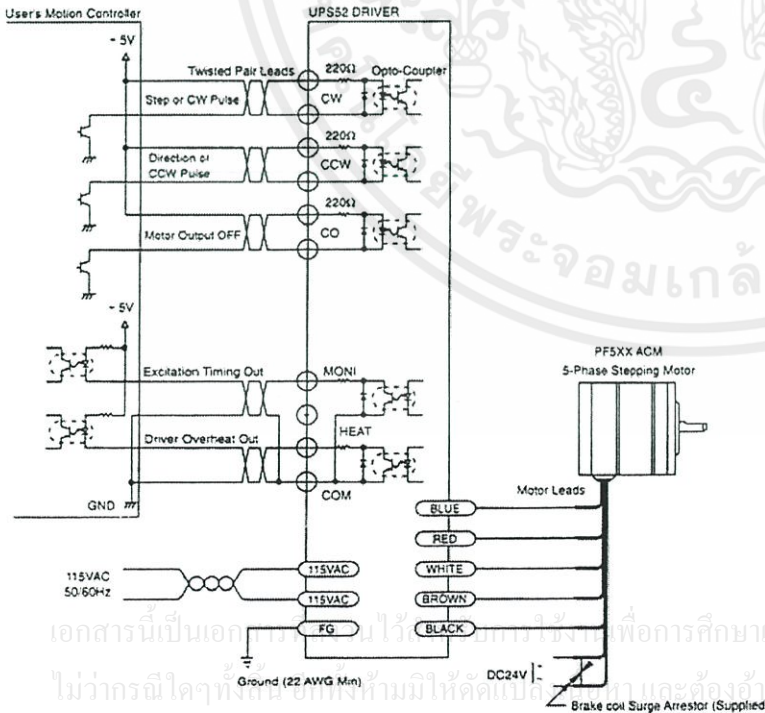
Input Pulse Waveform



- t1 (ON) ≥ 6μ s
- t2 (OFF) ≥ 10μ s
- t3 ≤ 2μ s
- t4 ≥ 20μ s
- t5 ≥ 24μ s

1. Input voltage: High = 4V to 5V, Low = 0V to 0.5V
2. Required input pulse characteristics
 Pulse duty: 50% or less, Width: 6 micro sec. or more, Rise/Fall time: 2 micro sec. or less.
3. If the input voltage exceeds 5V, use a series resistor of an appropriate value to limit the current flow through the opto-coupler LED to 15mA.

UPS52 Series Brake Type Connection



Driver Installation Notes

- * No connections or physical installation should be performed while the driver power is ON.
- * This driver is primarily designed for indoor use.
- * Avoid contact with corrosive materials, water, oils and excessive dust.
- * It may be necessary to install DC line noise filter(s) where strong noise sources exist near the driver. The line noise induction problems may be improved or avoided by a careful wire routing plan.
- * Be certain that conductive particles do not fall into the driver's internal structure.

* Supplied surge arrester (no polarity) must be connected between the + and - terminals of the 24VDC supply.

