การพัฒนาเครื่องมือในการถอด HSA/VCM อัตโนมัติ

นางสาวพชิรารัชต์ แก้วคำไสย์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2552

DEVELOPMENT OF HSA/VCM AUTOMATIC REMOVAL MACHINE

Phachirarat Kaewkamsai

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Engineering in Mechanical Engineering

Suranaree University of Technology

Academic Year 2009

การพัฒนาเครื่องมือในการถอด HSA/VCM อัตโนมัติ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

คณะกรรมการสอบวิทยานิพนธ์

_____

(ผศ. ดร.จิระพล ศรีเสริฐผล)
ประธานกรรมการ

_____

(รศ. ร.อ. ดร.กนต์ธร ชำนิประศาสน์)
กรรมการ (อาจารย์ที่ปรึกษาวิทยานิพนธ์)

_____

(อ. ดร.ธีระชาติ พรพิบูลย์)
กรรมการ

_____

(อ. ดร.กีรติ สุลักษณ์)
กรรมการ

_____

(ศ. ดร.ชูกิจ ลิมปีจำนงค์)
รองอธิการบดีฝ่ายวิชาการ

_____

(รศ. น.อ. ดร.วรพจน์ ขำพิศ)
คณบดีสำนักวิชาวิศวกรรมศาสตร์

พชิรารัชต์ แก้วคำไสย์ : การพัฒนาเครื่องมือในการถอด HSA/VCM อัตโนมัติ
(DEVELOPMENT OF HSA/VCM AUTOMATIC REMOVAL MACHINE)
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์, 199 หน้า.

งานวิจัยนี้สร้างเครื่องต้นแบบในการถอดชุดหัวอ่าน (Head Stack Assembly : HSA) และชุดควบคุมการเคลื่อนที่แบบ Voice Coil Motor (VCM) ในกระบวนการผลิตและประกอบฮาร์ดดิสก์ไดรฟ์ ในกระบวนการผลิตเดิมเมื่อมีการถอดชุด HAS และ VCM ออกมาเพื่อทำการปรับแก้โดยใช้แรงงานคนนั้น มักพบปัญหาด้านความสะอาดและความเสียหายของส่วนประกอบอื่น ๆ ขึ้น เนื่องจากกระบวนการต้องการการทำงานที่มีความแม่นยำสูง นอกจากนั้นการใช้แรงงานคนจะเสียเวลาในการทำงานมาก งานวิจัยนี้จึงจัดสร้างเครื่องต้นแบบเพื่อใช้ถอดชุด HSA/VCM ที่ทำงานโดยอัตโนมัติ ควบคุมด้วยไมโครคอนโทรลเลอร์ ซึ่งงานวิจัยเกี่ยวข้องกับการออกแบบระบบถอดสกรู ระบบดูดสกรูและสารปนเปื้อนในกระบวนการถอดแบบอัตโนมัติ พร้อมทั้งออกแบบโปรแกรมกำหนดขั้นตอนการทำงาน เครื่องต้นแบบที่ได้ช่วยลดความผิดพลาด และเวลาเนื่องจากวิธีการทำงานของมนุษย์ ปรับปรุงคุณภาพของผลิตภัณฑ์ ลดการปนเปื้อนและลดแรงกระแทกในระหว่างการถอด อีกทั้งลดต้นทุนในการผลิต อันเนื่องมาจากความเสียหายในกระบวนการปรับแก้โดยใช้มนุษย์แบบเดิม

สาขาวิชา วิศวกรรมเครื่องกล ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2552 ลายมือชื่ออาจารย์ที่ปรึกษา_____

PHACHIRARAT KAEWKAMSAI : DEVELOPMENT OF HSA/VCM AUTOMATIC REMOVAL MACHINE. THESIS ADVISOR : ASSOC. PROF. FLT. LT. KONTORN CHAMNIPRASART, Ph.D., 199 PP.

HEAD STACK ASSEMBLY/VOICE COIL MOTOR/HARD DISK DRIVE

This research represents the prototype machine that uses to produce and assemble Hard Disk Drive in part of the Head Stack Assembly (HSA) removal and control set of the Voice Coil Motor (VCM) movement. Due to original production, we found the problem about cleanliness and damage of other components. This research requires to create the prototype machine for Head Stack Assembly and Voice Coil Motor automatic removal that control by Microcontroller. This research has four parts. The first part is the design of screw removal system. The second part is the design of screw suck system. The third part is the contaminant in production of automatic removal and the last one is the design of work procedure. Researcher expects HSA/VCM automatic removal machine will reduce error and contamination, time-consuming of the human work process, improve quality of the product, avoid G-shock during the removal and reduce the production capital.

School of Mechanical Engineering Student's Signature _____

Academic Year 2009 Advisor's Signature _____

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงตามวัตถุประสงค์ทุกประการ ทั้งนี้ผู้ทำวิจัยขอขอบพระคุณบุคคลต่าง ๆ ที่ให้คำปรึกษา แนะนำ ชี้แนะ ช่วยเหลือ และเป็นกำลังใจมาโดยตลอดอย่างดียิ่งดังนี้

ศูนย์เทคโนโลยีอิเล็กทรอนิกส์ และคอมพิวเตอร์แห่งชาติ (NECTEC) ที่ให้เงินทุนสนับสนุนในการทำวิจัย

รองศาสตราจารย์ เรืออากาศเอก ดร.กนต์ธร ชำนิประศาสน์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ที่ได้ให้ความรู้ คำปรึกษา และความช่วยเหลือสนับสนุนเป็นอย่างดีในการทำวิจัย รวมทั้งสละเวลาตรวจสอบแก้ไขให้วิทยานิพนธ์มีความถูกต้องสมบูรณ์มากยิ่งขึ้น

ผู้ช่วยศาสตราจารย์ ดร.จิระพล ศรีเสริฐผล และอาจารย์ ดร.สมศักดิ์ ศิวดำรงพงศ์ ที่ให้ความรู้ คำแนะนำด้านวิชาการที่เป็นประโยชน์ในการดำเนินงานวิจัย และเป็นแบบอย่างที่ดีในการทำวิจัยแก่ผู้ทำวิจัย

รองศาสตราจารย์ นาวาอากาศเอก ดร.วรพจน์ ขำพิศ รองศาสตราจารย์ ดร.ทวิช จิตรสมบูรณ์ และคณาจารย์ทุกท่าน ที่ประสิทธิ์ประสาทวิชาความรู้ อบรมสั่งสอนทั้งในเนื้อหาอันเป็นทฤษฎี และปฏิบัติด้วยความเมตตากรุณา

คุณจุฑารัช พิไชยแพทย์ คุณอาภรณ์พรรณ ศรีอัครวิทยา และ เจ้าหน้าที่ประจำสำนักวิชาวิศวกรรมศาสตร์ ที่อำนวยความสะดวกในด้านธุรการต่าง ๆ

คุณกฤษณา เนระแก และคุณดนัย อ่ำเจริญ ผู้ร่วมในงานวิจัยนี้ ที่คอยต่อสู้ ฟันฝ่าปัญหาและอุปสรรคต่าง ๆ ให้ความช่วยเหลือ เป็นกำลังใจ ตั้งแต่เริ่มทำงานวิจัยนี้มาด้วยกัน

คุณสกล ประสานเสียง และคุณเสกสรรค์ พัทธามาศ ที่ช่วยให้คำปรึกษา และช่วยลงมือลงแรงในงานวิจัยนี้จนกระทั่งเสร็จสมบูรณ์

บริษัท ฮิตาชิ โกลบอล สตอเรจ เทคโนโลยีส์ (ประเทศไทย) จำกัด และ บุคคลต่าง ๆ ที่คอยสนับสนุนในทุกด้านด้วยดีเสมอมา

สุดท้ายนี้ ขอกราบขอบพระคุณบิดา มารดา ที่ให้ชีวิต เลี้ยงดูอบรม สนับสนุนด้านการศึกษา และคอยเป็นกำลังใจตลอดเวลา จนทำให้ผู้ทำวิจัยประสบความสำเร็จได้ในวันนี้ ขอขอบคุณญาติพี่น้องที่เป็นห่วงและสนับสนุนมาโดยตลอด ขอขอบพระคุณอย่างยิ่งจากใจจริงทุก ๆ ท่าน

พชิรารัชต์ แก้วคำไสย์

สารบัญ

หน้า

บทคัดย่อ (ภาษาไทย) ..... ก
บทคัดย่อ (ภาษาอังกฤษ) ..... ข
กิตติกรรมประกาศ ..... ค
สารบัญ ..... ง
สารบัญตาราง ..... ซ
สารบัญรูป ..... ฌ
คำอธิบายสัญลักษณ์และคำย่อ ..... ฌ

บทที่

1 บทนำ 1
1.1 ความเป็นมาและความสำคัญของปัญหา 1
1.2 วัตถุประสงค์ของงานวิจัย 2
1.3 ขอบเขตของการวิจัย 2
1.4 วิธีดำเนินงานวิจัย 2
1.4.1 ระเบียบวิธีวิจัย 2
1.4.2 สถานที่ทำการวิจัย 3
1.4.3 เครื่องมือที่ใช้ในการวิจัย 3
1.5 ประโยชน์ที่คาดว่าจะได้รับ 3
2 ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง 4
2.1 อุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ 4
2.1.1 กระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ 5
2.1.2 ขั้นตอนในการประกอบฮาร์ดดิสก์ไดรฟ์ 7
2.1.3 กระบวนการถอดเพื่อแก้ไขฮาร์ดดิสก์ไดรฟ์ 8
2.2 ส่วนประกอบฮาร์ดดิสก์ไดรฟ์ 9

สารบัญ (ต่อ)

หน้า

2.2.1 ชุดหัวอ่าน (Head Stack Assembly : HSA) 9
2.2.2 Voice Coil Motor (VCM) 10
2.2.3 Flex cable 12
2.2.4 สกรู (Screw) 14
2.2.5 อุปกรณ์พักหัวอ่าน 14
2.2.6 Head clip 16
2.3 ความเสียหายของส่วนประกอบในฮาร์ดดิสก์ไดรฟ์
ในกระบวนการผลิต 17
2.3.1 แรงกระแทก (Shock) 17
2.3.2 การสั่นสะเทือน (Vibration) 17
2.3.3 การปนเปื้อน (Contamination) 17
2.3.4 ไฟฟ้าสถิต (Electrostatic Discharge : ESD) 18
2.4 แรงกระแทก และการวิเคราะห์สัญญาณการกระแทก 19
2.4.1 แรงกระแทก 19
2.5 เครื่องจักรอัตโนมัติ 22
2.6 การออกแบบเครื่องจักรกล 23
2.6.1 กระบวนการในการออกแบบ 23
2.6.2 ข้อควรพิจารณาในการออกแบบ 24
2.7 ระบบควบคุมอัตโนมัติ 26
2.7.1 องค์ประกอบของระบบควบคุมอัตโนมัติ 27
2.7.2 ไมโครคอนโทรลเลอร์ (Microcontroller) 29
2.7.3 การติดต่อกับไมโครคอนโทรลเลอร์ด้วย
ไมโครซอฟท์วิชวลเบสิก 6.0 33

สารบัญ (ต่อ)

หน้า

3 การดำเนินงานวิจัย 39
3.1 กระบวนการถอด HSA/VCM โดยมนุษย์ 39
3.2 การออกแบบเครื่องจักรและระบบกลไก 41
3.2.1 การออกแบบเครื่องจักรด้วยโปรแกรม SolidWorks 41
3.2.2 ชุดส่วนประกอบที่ใช้ในเครื่องต้นแบบ 44
3.3 การควบคุมระบบการทำงานของเครื่องจักร 51
3.3.1 การพัฒนาโปรแกรมของบอร์ด ET-EASY MEGA1280 (DUINO MEGA) 52
3.3.2 การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ตระกูล AVR ด้วย Arduino 53
3.3.3 แผนผังของโปรแกรมควบคุมการทำงานของ เครื่องต้นแบบที่ควบคุมด้วยไมโครคอนโทรลเลอร์ 53
3.3.4 การติดต่อและควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่านพอร์ตอนุกรม กับ Microsoft Visual Basic 6.0 56
4 ผลการทดลองและวิเคราะห์ผลการทดลอง 61
4.1 ลำดับทำงานของเครื่องต้นแบบ 61
4.2 การปรับตั้งค่าตำแหน่ง และตัวแปรของเครื่องต้นแบบ 66
4.3 การวิเคราะห์แรงกระแทก 70
5 สรุปและข้อเสนอแนะ 73
รายการอ้างอิง 75
ภาคผนวก
ภาคผนวก ก. รายละเอียดขนาดชิ้นส่วนเครื่องต้นแบบ 76
ภาคผนวก ข. รายละเอียดโปรแกรม 112

สารบัญ (ต่อ)

หน้า

ข.1 โปรแกรมควบคุมและสั่งงานการทำงาน
ของเครื่องต้นแบบ 113
ข.2 โปรแกรมติดต่อและควบคุมการทำงานของ
อุปกรณ์ต่าง ๆ ระหว่างคอมพิวเตอร์กับ
ไมโครคอนโทรลเลอร์ 150
ภาคผนวก ค. คู่มือการใช้งานเครื่องต้นแบบ 181
ค.1 การติดตั้งเครื่องต้นแบบก่อนการใช้งาน 182
ค.2 ขั้นตอนการใช้งานเครื่องต้นแบบ 191
ภาคผนวก ง. รายชื่อบทความวิจัยที่ได้รับการตีพิมพ์เผยแพร่ 192
ประวัติผู้เขียน 199

สารบัญตาราง

ตารางที่ **หน้า**

2.1 ตระกูลไมโครคอนโทรลเลอร์ 33
2.2 แสดงรายละเอียดของสายสัญญาณ 36
4.1 ตารางบันทึกเวลาจากการปรับค่าอุปกรณ์จำนวน 10 ครั้ง 67
4.2 แสดงเวลาที่ใช้ในแต่ละขั้นตอนการทำงาน 68

สารบัญรูป

รูปที่ **หน้า**

2.1 ขั้นตอนในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ 5
2.2 การประกอบฮาร์ดดิสก์ไดรฟ์ 8
2.3 องค์ประกอบของฮาร์ดดิสก์ไดรฟ์ 9
2.4 ชุดหัวอ่าน และส่วนประกอบ 10
2.5 Voice Coil Motor (VCM) 11
2.6 ลักษณะของ VCM โดยแม่เหล็ก 1 ชั้นประกอบด้วยขั้วแม่เหล็ก 2 ขั้ว 11
2.7 การประกอบชุดหัวอ่านเข้ากับ VCM 12
2.8 ส่วนประกอบของ flex cable 13
2.9 จุดต่อของชุดขดลวดบน flex cable 13
2.10 ลักษณะตัวอย่างของสกรูภายในฮาร์ดดิสก์ไดรฟ์ 14
2.11 ลักษณะของอุปกรณ์พักหัวอ่านขนาดต่าง ๆ 15
2.12 การพักหัวอ่านบนอุปกรณ์พักหัวอ่านขนาดต่าง ๆ 15
2.13 Head clip 16
2.14 การใส่ head clip บนชุดหัวอ่าน 16
2.15 การหาค่าการดลจากกราฟแรงและเวลา 20
2.16 การวิเคราะห์สัญญาณการกระแทกจากความเร่งกับเวลา 21
2.17 ขั้นตอนในการออกแบบ 24
2.18 ระบบการควบคุมแบบเปิด 28
2.19 ระบบควบคุมแบบปิด หรือระบบการควบคุมแบบป้อนกลับ 28
2.20 โครงสร้างไมโครคอนโทรลเลอร์ 31
2.21 ลักษณะของตัวเชื่อมต่อ 35
3.1 อุปกรณ์ที่ดำเนินการถอดหลังจากกระบวนการขันสกรูออก ที่ตำแหน่ง HSA ยึดกับฐาน 39
3.2 ชุดจับยึดชิ้นงานของกระบวนการถอด HSA/VCM โดยมนุษย์ 40
3.3 ตำแหน่งที่ทำการขันสกรูออก และกด flex 41

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.4 ลักษณะของเครื่องต้นแบบที่ทำการออกแบบ
ด้วยโปรแกรม SolidWorks 42
3.5 ชุดจับยึดชิ้นงานที่ออกแบบ 42
3.6 การหลุดของ HSA ออกจากแผ่นข้อมูล 43
3.7 ชุดขาตั้งยึดกระบอกสูบลมของการกดฮาร์ดดิสก์ไดรฟ์ และ flex 43
3.8 ชุดขาตั้งยึดกระบอกสูบลมของชุดท่อสุญญากาศ 44
3.9 ชุดขาตั้งยึดกระบอกสูบไฟฟ้าของชุดไขควงออกจากแผ่นข้อมูล 44
3.10 ชุดการเคลื่อนที่แบบหมุนของบริษัท NSK รุ่น YS3008FN001 45
3.11 แผนภาพระบบของชุดการเคลื่อนที่แบบหมุน 46
3.12 ชุดกระบอกสูบไฟฟ้าของบริษัท IAI
รุ่น RCP2CR-SA6-I-PM-2-100-P1-S-BL-JP 47
3.13 แผนภาพระบบของชุดกระบอกสูบไฟฟ้า 47
3.14 ชุดกระบอกสูบลมของบริษัท SMC รุ่น MXQ 48
3.15 แผนภาพระบบของชุดกระบอกสูบลม 48
3.16 Filter regulator ของบริษัท SMC รุ่น AW20-02BG 49
3.17 Speed control fitting ชนิด Elbow ของบริษัท SMC 49
3.18 ชุดอุปกรณ์ขันสกรูของบริษัท Technart 50
3.19 แผนภาพระบบของชุดอุปกรณ์ขันสกรู 51
3.20 บอร์ด ET-EASY MEGA1280 (DUINO MEGA)
ใช้ชิพรุ่น ATMEGA 1280 52
3.21 แผนผังของโปรแกรมควบคุมการทำงานของเครื่องต้นแบบ 56
3.22 แผนผังแสดงขั้นตอนการติดต่อสื่อสารไมโครคอนโทรลเลอร์
กับคอมพิวเตอร์ 57
3.23 หน้าต่างการติดต่อ และควบคุมอุปกรณ์ของเครื่องต้นแบบ 58
3.24 หน้าต่างการตั้งค่าของโปรแกรม RCPC 60
4.1 เครื่องต้นแบบ 61

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

4.2 การวางฮาร์ดดิสก์ไดรฟ์บนชุดจับยึดชิ้นงาน....62
4.3 ชิ้นงานเคลื่อนที่เข้ามาที่ตำแหน่งทำงาน....62
4.4 การกดฮาร์ดดิสก์ไดรฟ์ และกด flex....63
4.5 ตำแหน่งปฏิบัติงานของชุดกระบอกสูบลมยึดชุดท่อสุญญากาศ....63
4.6 ตำแหน่งปฏิบัติงานของชุดกระบอกสูบไฟฟ้ายึดไขควง....64
4.7 ตำแหน่งของชุดกระบอกสูบไฟฟ้ายึดไขควง
เมื่อเคลื่อนที่ขึ้นไปตำแหน่งบน....64
4.8 ตำแหน่งของชุดกระบอกสูบลมยึดชุดท่อสุญญากาศ
เมื่อเคลื่อนที่ขึ้น....65
4.9 ตำแหน่งชุดกระบอกสูบลมยึดอุปกรณ์กดฮาร์ดดิสก์ไดรฟ์
และกด flex เมื่อเคลื่อนที่ขึ้น....65
4.10 ลักษณะของ HSA/VCM และFlex อยู่บนที่พักบนตัวจับยึดชิ้นงาน....66
4.11 ลักษณะของชิ้นงานภายหลังจากกระบวนการขันสกรูออก....68
4.12 การปรับค่าที่ให้เวลาในการทำงานดีที่สุดของชุดการเคลื่อนที่
แบบหมุน....69
4.13 ค่าตำแหน่งและตัวแปรที่ทำการปรับที่ให้เวลาในการทำงานดีที่สุด
ของชุดการเคลื่อนที่ในแนวเส้นตรงของชุดไขควง....70
ก.1 องค์ประกอบของเครื่องต้นแบบ....77
ก.2 หมายเลขของชิ้นส่วนต่าง ๆ ของเครื่องต้นแบบ....77
ก.3 หมายเลขของชิ้นส่วนในส่วนชุดขาตั้งยึดกระบอกสูบไฟฟ้า....78
ก.4 หมายเลขของชิ้นส่วนในส่วนชุดขาตั้ง
ยึดกระบอกสูบลมของชุดกดฮาร์ดดิส์ไดรฟ์....78
ก.5 หมายเลขของชิ้นส่วนของชุดกดฮาร์ดดิสก์ไดรฟ์....79
ก.6 หมายเลขชิ้นส่วนของชุดจับยึดชิ้นงาน....79
ก.7 หมายเลขชิ้นส่วนของชุด HOUSING.... 80
ก.8 หมายเลขชิ้นส่วนของ ABSORBER....80

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ก.9 รายละเอียดขนาดของ BASE PLATE 81
ก.10 รายละเอียดขนาดของ ROTATION PLATE 81
ก.11 รายละเอียดขนาดของ OPEN PLATE 82
ก.12 รายละเอียดขนาดของ PRECISION PLATE 82
ก.13 รายละเอียดขนาดของ PIN GUIDE 83
ก.14 รายละเอียดขนาดของ HOUSING GUIDE 83
ก.15 รายละเอียดขนาดของ PLATE GUIDE 84
ก.16 รายละเอียดขนาดของ SLIDE PIN 84
ก.17 รายละเอียดขนาดของ HSA SUPPORT-1 85
ก.18 รายละเอียดขนาดของ SLIDE PLATE-1 85
ก.19 รายละเอียดขนาดของ COLUMN PLATE-1-BACK 86
ก.20 รายละเอียดขนาดของ COLUMN PLATE-1-FRONT 86
ก.21 รายละเอียดขนาดของ FIX PLATE-1 87
ก.22 รายละเอียดขนาดของ SLIDE PLATE-2 87
ก.23 รายละเอียดขนาดของ FRONT PLATE-1 88
ก..24 รายละเอียดขนาดของ ANGLE PLATE-1 88
ก..25 รายละเอียดขนาดของ ADJUSTING BOLTS M8 89
ก.26 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M8-1-BOTTOM 89
ก.27 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M8-1-TOP 90
ก.28 รายละเอียดขนาดของ BLOCK ADJUSTING M8 90
ก.29 รายละเอียดขนาดของBLOCK FOR ADJUSTING BOLTS M8-2 91
ก.30 รายละเอียดขนาดของ HSA SUPPORT 91
ก.31 รายละเอียดขนาดของ ABSORBER-01 92

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ก.32 รายละเอียดขนาดของ HSA SUPPORT-2 92
ก.33 รายละเอียดขนาดของ DATUM PLATE-1 93
ก.34 รายละเอียดขนาดของ VERTICAL PLATE-1 93
ก.35 รายละเอียดขนาดของ ANGLE PLATE-2 94
ก.36 รายละเอียดขนาดของ DATUM PIN-1 94
ก.37 รายละเอียดขนาดของ DATUM PIN-2 95
ก.38 รายละเอียดขนาดของ DATUM PIN-3 95
ก.39 รายละเอียดขนาดของ VERTICAL PLATE-2 96
ก.40 รายละเอียดขนาดของ SLIDE PLATE-3 96
ก.41 รายละเอียดขนาดของ FIX PLATE-2 97
ก.42 รายละเอียดขนาดของ COLUMN PLATE-2 97
ก.43 รายละเอียดขนาดของ SLIDE PLATE-4 98
ก.44 รายละเอียดขนาดของ ANGLE PLATE-3 98
ก.45 รายละเอียดขนาดของ ADJUST BOLTS M6 99
ก.46 รายละเอียดขนาดของ BLOCK ADJUSTING M6 99
ก.47 รายละเอียดขนาดของ BLOCK FOR ADJUSTING
BOLTS M6-BOTTOM 100
ก.48 รายละเอียดขนาดของ BLOCK FOR ADJUSTING
BOLTS M6-TOP 100
ก.49 รายละเอียดขนาดของ ABSORBER-02 101
ก.50 รายละเอียดขนาดของ FRONT PLATE-1 101
ก.51 รายละเอียดขนาดของ ABSORBER-03 102
ก.52 รายละเอียดขนาดของ ABSORBER-04 102
ก.53 รายละเอียดขนาดของ FRONT PLATE-2 103
ก.54 รายละเอียดขนาดของ VACUUM PLATE-1 103
ก.55 รายละเอียดขนาดของ FRONT PLATE-3 104

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ก.56 รายละเอียดขนาดของ VACUUM PLATE-2 104
ก.57 รายละเอียดขนาดของ VACUUM PLATE-3 105
ก.58 รายละเอียดขนาดของ VACUUM PLATE-4 105
ก.59 รายละเอียดขนาดของ VACUUM ANGLE PLATE 106
ก.60 รายละเอียดขนาดของ VACUUM PLUG....106
ก.61 รายละเอียดขนาดของ SIDE PLATE-1 107
ก.62 รายละเอียดขนาดของ ABSORBER-05....107
ก.63 รายละเอียดขนาดของ ANGLE PLATE-4....108
ก.64 รายละเอียดขนาดของ WASHER M4....108
ก.65 รายละเอียดขนาดของ WASHER M6....109
ก.66 รายละเอียดขนาดของ WASHER M8....109
ก.67 รายละเอียดขนาดของ SCREW HOLDER....110
ก.68 รายละเอียดขนาดของ SUPPORT SERVO MOTOR....110
ก.69 รายการหมายเลขของชิ้นส่วน....111
ค.1 หน้าต่างโปรแกรม Setup machine 182
ค.2 หน้าต่างโปรแกรมในส่วนการตั้งค่าตัวแปร
ของชุดการเคลื่อนที่แบบหมุน 183
ค.3 หน้าต่างการเลือกการตั้งค่าตำแหน่งและตัวแปร
ของโปรแกรม RCPC 184
ค.4 หน้าต่างการใส่ค่าตำแหน่ง ตัวแปรและการส่งข้อมูล
ไปชุดกล่องควบคุม 284
ค.5 ไฟสถานะของบอร์ดไมโครคอนโทรลเลอร์....186
ค.6 หน้าต่างแสดงไฟสถานะเมื่อกดปุ่ม HOME MOTOR....188
ค.7 หน้าต่างแสดงไฟสถานะเมื่อกด Clamp down....188
ค.8 หน้าต่างแสดงไฟสถานะเมื่อกด Vacuum down....189
ค.9 หน้าต่างแสดงไฟสถานะเมื่อกดเปิดสกรู 189

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

ค.10 หน้าต่างแสดงไฟสถานะเมื่อกดเปิดปั๊ม.....190
ค.11 หน้าต่างแสดงไฟสถานะเมื่อกดปุ่ม HOME
ของชุดกระบอกสูบไฟฟ้า.....190

คำอธิบายสัญลักษณ์และคำย่อ

HDD	Hard Disk Drive
HSA	Head Stack Assembly
VCM	Voice Coil Motor
G-Shock	Gravity Shock
HDA	Head Disk Assembly
PCBA	Printed Circuit Board Assembly
AE	Arm Electronic
FPC	Flexible Printed Circuit
HGA	Head Gimbals Assembly
CSS	Contact Start-Stop
ANSI	The American National Standards Institute
MCU	Micro Controller Unit
RS-232	Recommended Standard-232
DTE	Data Terminal Equipment
DCE	Data Communication Equipment
ROM	Read Only Memory
RAM	Random Access Memory
EEPROM	Erasable Electrically Read-Only Memory
VB	Visual Basic
GUI	Graphic User Interface
PLC	Programmable Logic Controller

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์มีการขยายตัวสูงขึ้น บริษัทผู้ผลิตฮาร์ดดิสก์ไดรฟ์จึงต้องมีการพัฒนาเทคโนโลยีในการผลิตให้สูงขึ้น เพื่อให้ได้ผลิตภัณฑ์ที่มีประสิทธิภาพและเพื่อแข่งขันกับบริษัทผู้ผลิตอื่น อุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์การผลิตชิ้นส่วนต่าง ๆ มักจะถือว่าเป็นกระบวนการซึ่งต้องอาศัยการผลิตขั้นสูง นั่นคือ ผู้ผลิตต้องการให้กระบวนการผลิตชิ้นงานเป็นกระบวนการผลิตที่มีประสิทธิภาพสูง มีการทำงานที่รวดเร็วและผิดพลาดน้อย เช่น ความต้องการชิ้นงานที่มีความสะอาดสูงมากปราศจากสิ่งปนเปื้อนขนาดเล็ก ความต้องการระบบประกอบและถอดประกอบชิ้นงานอัตโนมัติที่มีความแม่นยำสูง เป็นต้น ดังนั้นกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ส่วนใหญ่จะใช้เครื่องจักรอัตโนมัติในกระบวนการผลิต เพื่อให้ได้ผลิตภัณฑ์มีคุณภาพ และประสิทธิภาพสูง

บริษัท ฮิตาชิ โกลบอล สตอเรจ เทคโนโลยีส์ (ประเทศไทย) จำกัด เป็นศูนย์กลางในการผลิตฮาร์ดดิสก์ไดรฟ์รายใหญ่ของประเทศไทย ผลิตฮาร์ดดิสก์ไดรฟ์ขนาด 1 1.8 และ 2.5 นิ้ว ที่เหมาะสำหรับผู้ใช้คอมพิวเตอร์ทั่วไป องค์กรธุรกิจ และสำหรับผู้บริโภคผลิตภัณฑ์อิเล็กทรอนิกส์ และฮาร์ดดิสก์ไดรฟ์ขนาด 3.5 นิ้ว ซึ่งเป็นฮาร์ดดิสก์ไดรฟ์ที่นิยมใช้กันอย่างแพร่หลายสำหรับ เครื่องคอมพิวเตอร์แบบตั้งโต๊ะ

กระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ เมื่อประกอบฮาร์ดดิสก์ไดรฟ์แล้วจะต้องมีการทดสอบประสิทธิภาพการทำงานจริงของฮาร์ดดิสก์ไดรฟ์ ซึ่งหลังจากที่ทดสอบการทำงานแล้วพบความผิดพลาดในตัวฮาร์ดดิสก์ไดรฟ์ เช่น พบว่ามีความผิดพลาดในการเคลื่อนที่ของชุดหัวอ่าน (Head Stack Assembly : HSA) เป็นต้น จะต้องทำการถอดประกอบฮาร์ดดิสก์ไดรฟ์เพื่อปรับปรุง ปรับเปลี่ยน หรือซ่อมแซมชิ้นส่วนที่บกพร่องให้ได้ผลิตภัณฑ์ที่ได้จากกระบวนการผลิตดีขึ้น ในกระบวนการถอดประกอบฮาร์ดดิสก์ไดรฟ์เพื่อทำการปรับแก้จะต้องมีการถอดชุด HSA และชุดควบคุมการเคลื่อนที่แบบ Voice Coil Motor (VCM) ออกมาก่อน ซึ่งในปัจจุบันกระบวนการถอดชุด HSA/VCM ของทางบริษัท ฮิตาชิ โกลบอล สตอเรจ เทคโนโลยีส์ (ประเทศไทย) จำกัด โดยส่วนใหญ่เป็นการใช้แรงงานคนในการถอดประกอบ และจากผลการทำงานโดยใช้แรงงานคนจะเกิดปัญหาในด้านความเสียหายของชิ้นส่วน ความสะอาดและเวลา ซึ่งทางบริษัท ฮิตาชิ โกลบอล

สตอเรจ เทคโนโลยีส์ (ประเทศไทย) จำกัด มีความเห็นว่าควรสร้างเครื่องต้นแบบควบคุมด้วยระบบอัตโนมัติขึ้นมาใช้ในกระบวนการถอด HSA/VCM โดยพัฒนาจากกระบวนการถอด HSA/VCM โดยแรงงานคนของฮาร์ดดิสก์ไดรฟ์ขนาด 2.5 นิ้ว เพื่อลดความผิดพลาด ลดระยะเวลาในการทำงาน ลดค่าใช้จ่ายในการนำเข้า เพื่อให้เป็นเครื่องต้นแบบของบริษัท และพัฒนาเทคโนโลยีในกระบวนการถอด HSA/VCM และเพิ่มประสิทธิภาพในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ของภาคอุตสาหกรรม

1.2 วัตถุประสงค์ของงานวิจัย

1) ออกแบบและสร้างเครื่องต้นแบบ ในกระบวนการถอดประกอบชุด HSA/VCM โดยอัตโนมัติ
2) ออกแบบระบบควบคุมการทำงานของเครื่องต้นแบบโดยอัตโนมัติ
3) เพิ่มประสิทธิภาพการทำงานของการถอด HSA/VCM ให้มีความเสียหาย และลดระยะเวลาในการทำงานให้น้อยที่สุด

1.3 ขอบเขตของงานวิจัย

1) ออกแบบเครื่องต้นแบบในการถอด HSA/VCM ของฮาร์ดดิสก์ไดรฟ์ ขนาด 2.5 นิ้ว
2) ใช้โปรแกรม SolidWorks 2008 ออกแบบและจำลองระบบการทำงานของเครื่องต้นแบบ
3) หาเวลาที่ดีที่สุดในการทำงานของเครื่องต้นแบบ
4) วิเคราะห์แรงกระแทกที่เกิดต่อผลิตภัณฑ์ ในระหว่างกระบวนการถอด HSA/VCM เพื่อหาค่าที่ผลิตภัณฑ์ยอมรับได้

1.4 วิธีดำเนินการวิจัย

วิธีการดำเนินการวิจัยในส่วนของงานวิจัยนี้จะแบ่งออกเป็น 3 ส่วน คือ ระเบียบวิธีวิจัย สถานที่ทำการวิจัย และเครื่องมือที่ใช้ในงานวิจัย โดยมีรายละเอียดดังนี้

1.4.1 ระเบียบวิธีวิจัย จะแบ่งเป็นลำดับขั้นตอนดังนี้

1) ศึกษากระบวนการถอดประกอบ HSA/VCM โดยแรงงานคนอย่างละเอียด
2) ศึกษาลักษณะของเครื่องจักรอัตโนมัติที่ต้องทำงานประเภทนี้แทนคน
3) ออกแบบเครื่องจักรกลและกลไกต่าง ๆ ตามทฤษฎีการออกแบบกลไก
4) ออกแบบกลไกเพื่อรองรับการทำงานอัตโนมัติ

5) ศึกษาระบบ-ระเบียบวิธีการควบคุมอัตโนมัติที่เหมาะสมในการที่จะควบคุมระบบกลไก

1.4.2 สถานที่ทำการวิจัย

มหาวิทยาลัยเทคโนโลยีสุรนารี 111 ถนนมหาวิทยาลัย ตำบลสุรนารี อำเภอเมือง จังหวัดนครราชสีมา 30000

1.4.3 เครื่องมือที่ใช้ในการวิจัย

เครื่องมือที่ใช้ในการวิจัยประกอบด้วย

1) คอมพิวเตอร์
2) โปรแกรม SolidWork 2008
3) อุปกรณ์ และฮาร์ดแวร์ต่าง ๆ จากบริษัท ฮิตาชิ โกลบอล สตอเรจ เทคโนโลยีส์ (ประเทศไทย) จำกัด

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1) เครื่องต้นแบบควบคุมอัตโนมัติสำหรับกระบวนการถอด HSA/VCM เพื่อลดความผิดพลาดเนื่องจากการทำงานของแรงงานคน
2) ได้เครื่องต้นแบบที่สามารถเพิ่มคุณภาพของผลิตภัณฑ์ ลดการปนเปื้อน และลดแรงกระแทกในระหว่างการถอดซึ่งจะต้องมีความสามารถที่จะทำงานได้ภายในกรอบที่บริษัทยอมรับให้ใช้ในกระบวนการผลิตได้
3) ภาคอุตสาหกรรมสามารถลดระยะเวลา และลดต้นทุนในการผลิต จากความเสียหายในกระบวนการแก้ไขงานที่ผิดพลาด โดยสามารถลดระยะเวลาลงได้ไม่น้อยกว่าร้อยละ 20 และลดยอดการเสียหายลงได้ไม่น้อยกว่าร้อยละ 30

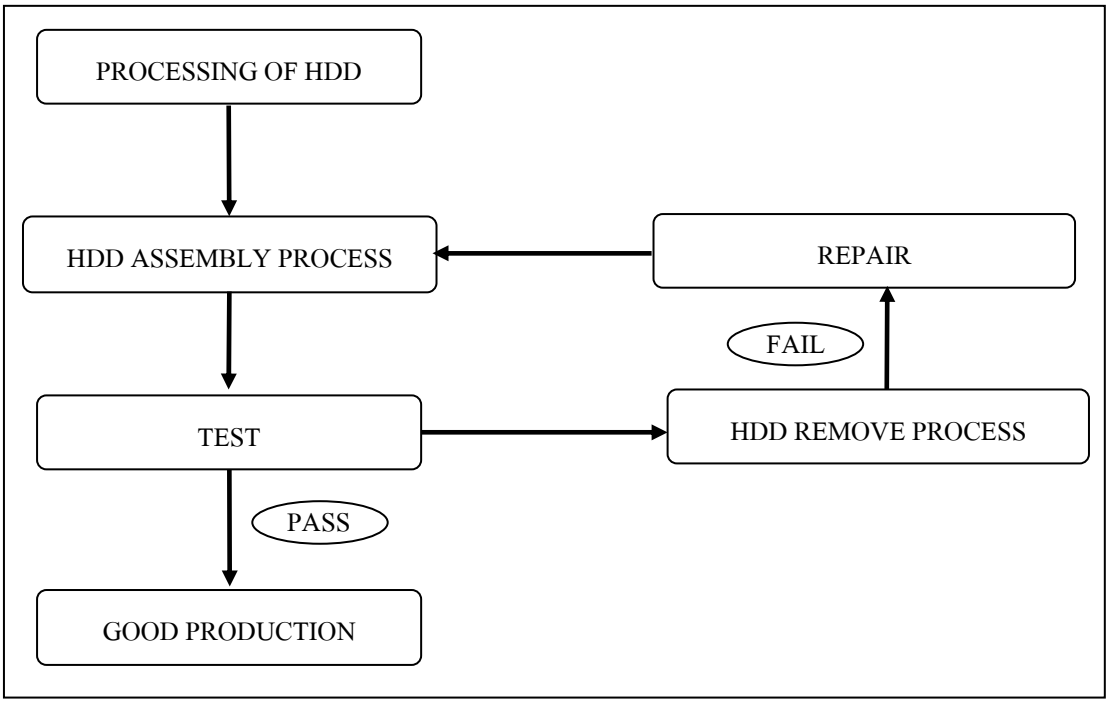
บทที่ 2
ปริทัศน์วรรณกรรมและงานวิจัยที่เกี่ยวข้อง

เทคโนโลยีการผลิตฮาร์ดดิสก์ไดรฟ์ของภาคอุตสาหกรรมในปัจจุบัน กระบวนการผลิตต่าง ๆ จะเป็นการผลิตด้วยเครื่องจักรอัตโนมัติ เนื่องจากการใช้เครื่องจักรอัตโนมัติเข้ามาช่วยในกระบวนการผลิตทำให้กระบวนการผลิตมีประสิทธภาพมากขึ้น ภาคอุตสาหกรรมสามารถลดต้นทุน และเวลาในการผลิตจากการใช้แรงงานคนในการผลิตได้

บทนี้กล่าวถึงอุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์ในส่วนต่าง ๆ เช่น ขั้นตอนการผลิตชิ้นส่วน การประกอบฮาร์ดดิสก์ไดรฟ์ กระบวนการถอดประกอบเพื่อแก้ไขฮาร์ดดิสก์ไดรฟ์ และความเสียหายของชิ้นงานในระหว่างกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ เพื่อให้ทราบถึงหลักสำคัญในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ และนำข้อมูลไปทำการออกแบบเครื่องจักรอัตโนมัติที่ใช้ในกระบวนการถอด HSA/VCM โดยจะกล่าวถึงอุปกรณ์ในฮาร์ดดิสก์ไดรฟ์ ที่เกี่ยวข้องกับกระบวนการถอด HSA/VCM ที่ต้องให้ความสำคัญต่อผลที่จะเกิดขึ้นกับอุปกรณ์ ภายหลังจากกระบวนการถอด HSA/VCM ประเภทเครื่องจักรอัตโนมัติ ทฤษฎีเกี่ยวกับการออกแบบเครื่องจักรกล ระบบควบคุมอัตโนมัติด้วยไมโครตัวควบคุมเลอร์ และโปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 ที่ช่วยในการติดต่อสื่อสารระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

2.1 อุตสาหกรรมการผลิตฮาร์ดดิสก์ไดรฟ์

ในการผลิตฮาร์ดดิสก์ไดรฟ์ เมื่อกระบวนการประกอบเสร็จสิ้นจนกระทั่งได้ฮาร์ดดิสก์ไดรฟ์ที่พร้อมใช้งานนั้น ก่อนที่จะนำส่งไปให้ผู้บริโภค ทางบริษัทผู้ผลิตจะต้องทำการทดสอบคุณสมบัติต่าง ๆ ของฮาร์ดดิสก์ไดรฟ์ ซึ่งถ้าพบว่าคุณสมบัติต่าง ๆ ผ่านตามมาตรฐานที่กำหนดก็จะนำไปบรรจุและส่งต่อไปยังผู้บริโภค แต่ถ้าคุณสมบัติต่าง ๆ ไม่ผ่านตามมาตรฐานที่กำหนดก็จะต้องนำกลับไปถอดประกอบเพื่อทำการตรวจหาสาเหตุ



รูปที่ 2.1 ขั้นตอนในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์

2.1.1 กระบวนการผลิตฮาร์ดดิสก์ไดรฟ์

การผลิตฮาร์ดดิสก์ไดรฟ์มีกระบวนการที่สำคัญ 2 อย่าง คือกระบวนการประกอบหัวอ่านเข้ากับชิ้นส่วนจับยึดหัวอ่านกับแขน (Head Disk Assembly : HDA) และกระบวนการประกอบอุปกรณ์ทางอิเล็กทรอนิกส์บนแผ่นวงจรพิมพ์ (Printed Circuit Board Assembly : PCBA) ซึ่งส่วนแรกจะเป็นส่วนการประกอบชุดหัวอ่านและเขียน จานเก็บข้อมูล และส่วนประกอบที่เป็นชิ้นส่วนทางกล ในขณะที่ส่วนที่สองจะเป็นการประกอบชิ้นส่วนทางด้านวงจรอิเล็กทรอนิกส์ที่ใช้ในการควบคุมการทำงานของฮาร์ดดิสก์ไดรฟ์ และเชื่อมต่อกับอุปกรณ์อื่น ๆ เช่น คอมพิวเตอร์ เป็นต้น โดยสามารถแบ่งขั้นตอนการผลิตฮาร์ดดิสก์ไดรฟ์ออกเป็น 3 ส่วนด้วยกัน คือ การผลิตชิ้นส่วนพื้นฐาน (Component fabrication) การประกอบชิ้นส่วนย่อย (Subassembly) การประกอบสุดท้าย (Final assembly) โดยกระบวนการประกอบหัวอ่านเข้ากับชิ้นส่วนจับยึดหัวอ่านกับแขนมีขั้นตอนดังนี้

1) การผลิตชิ้นส่วนพื้นฐานต่าง ๆ จะเป็นการเริ่มกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ โดยกระบวนการแรกจะอยู่ในส่วนของการผลิตชิ้นส่วนของหัวอ่าน (Head fabrication) จะเริ่มจากการผลิตแผ่นหัวอ่าน (Wafer) เพื่อที่จะทำหัวอ่านและเขียนข้อมูล (Slider) ขั้นตอนต่อมาคือการผลิตแผ่นเก็บข้อมูล (Media) ซึ่งในที่นี้คือแผ่นดิสก์ (Hard Disk Platter or Disk) ซึ่งผลิตจาก

2) แก้วที่ต้องทำเป็นพิเศษเคลือบด้วยสารแม่เหล็ก รวมไปถึงชิ้นส่วนในการประกอบมอเตอร์ที่ใช้ ในการหมุนแผ่นดิสก์ (Spindle motor) นอกจากนี้ยังมีการผลิตฐาน (Base) และชิ้นส่วนอื่น ๆ ในการผลิตชิ้นส่วนจับยึดหัวอ่านกับแขน (Suspension)

3) กระบวนการประกอบชิ้นส่วนย่อยมีหลายกระบวนการ ดังนี้ กระบวนการของการประกอบชุดหัวอ่าน (Head Stack Assembly : HSA) เป็นการเชื่อมหัวอ่านและเขียนข้อมูลและชิ้นส่วนจับยึดหัวอ่านกับแขนเข้าด้วยกัน โดยนำ Head Gimbals Assembly (HGA) หลายอันมาวางซ้อนกันและเชื่อมต่อกับแขน (Arm) และชุดขดลวดมอเตอร์สำหรับการหมุนหัวอ่าน (Coil assembly) และแผ่นวงจรพิมพ์ของ Flex (Flexible Printed Circuit : FPC) เพื่อประกอบเป็นระบบของหัวอ่านที่ซ้อนกันเป็นชั้น ๆ ที่สามารถเคลื่อนที่ไปมาได้ กระบวนการประกอบชุดขดลวดมอเตอร์ของแขนหัวอ่าน (Actuator Coil Assembly) และ FPC ซึ่งกระบวนการนี้บางทีเรียกว่า Voice Coil Motor Assembly (VCMA) และกระบวนการประกอบชิ้นส่วนต่าง ๆ ของชุดแกนหมุนแผ่นข้อมูล

4) การประกอบสุดท้ายภายหลังจากที่ได้ชุดหัวอ่าน ชุดแกนหมุนแผ่นข้อมูล ฐาน ฝาครอบบน (Top cover) และแผ่นดิสก์จะมีการประกอบชิ้นส่วนเหล่านี้เข้าด้วยกันเป็นชุดหัวอ่านเข้ากับชิ้นส่วนจับยึดหัวอ่านกับแขน เพื่อรอการประกอบกับส่วนของชุดอุปกรณ์ทางอิเล็กทรอนิกส์บนแผ่นวงจรพิมพ์

กระบวนการประกอบอุปกรณ์ทางอิเล็กทรอนิกส์บนแผ่นวงจรพิมพ์สามารถแบ่งขั้นตอนในการผลิตได้ดังนี้

1) การผลิตชิ้นส่วนพื้นฐาน โดยเริ่มผลิตแผ่นหัวอ่านเพื่อที่จะทำวงจรรวม (Integrated Circuit : IC) และมีการผลิตแผ่นวงจรพิมพ์ (Printed Circuit Board : PCB)

2) การประกอบชิ้นส่วนย่อย เป็นการนำวงจรรวม และชิ้นส่วนอิเล็กทรอนิกส์อื่น ๆ ไปประกอบเข้ากับแผ่นวงจรพิมพ์ ซึ่งกระบวนการนี้เรียกว่า กระบวนการประกอบอุปกรณ์ทางอิเล็กทรอนิกส์บนแผ่นวงจรพิมพ์

3) เมื่อได้ทั้งชิ้นส่วนจับยึดหัวอ่านกับแขนและชุดอุปกรณ์ทางอิเล็กทรอนิกส์บนแผ่นวงจรพิมพ์แล้ว จะมีการประกอบเข้าด้วยกันเป็นฮาร์ดดิสก์ไดรฟ์ในขั้นตอนของการประกอบสุดท้าย ในการผลิตฮาร์ดดิสก์ไดรฟ์นั้นยังมีกลุ่มวัสดุอื่น ๆ และกลุ่มวัสดุทางอ้อม (Indirect materials) ที่ใช้อยู่ในทุกขั้นตอนของการผลิตฮาร์ดดิสก์ไดรฟ์ ตั้งแต่การผลิตชิ้นส่วนพื้นฐานไปจนถึงการประกอบสุดท้าย กลุ่มวัสดุอื่น ๆ จะเป็นชิ้นส่วนที่ไม่ได้เป็นส่วนประกอบหลัก เช่น สกรู ชิ้นส่วนโลหะอื่น ๆ หรือชิ้นส่วนที่ไม่สามารถบอกได้ว่าถูกใช้เป็นส่วนประกอบในกระบวนการผลิตใดอย่างแน่ชัด ในขณะที่กลุ่มวัสดุทางอ้อมจะเป็นสิ่งที่ใช้ร่วมในการผลิตแต่ไม่ได้เป็นชิ้นส่วนในการประกอบฮาร์ดดิสก์ไดรฟ์โดยตรง เช่น วัสดุที่เกี่ยวข้องกับห้องสะอาด

ต่าง ๆ ได้แก่ ตัวกรองฝุ่น ถุงมือ ชุดทำงาน หน้ากาก รองเท้า และวัสดุที่เกี่ยวข้องกับการลดประจุไฟฟ้า เช่น แผ่นรองพื้นวัสดุที่ใช้ในการบรรจุ และบรรจุภัณฑ์ต่าง ๆ รวมถึงระบบเครื่องจักรกลอัตโนมัติ และการสร้างชุดจับยึดชิ้นงาน (Jig/fixture) ที่จะช่วยสนับสนุนการผลิตฮาร์ดดิสก์ไดรฟ์

2.1.2 ขั้นตอนในการประกอบฮาร์ดดิสก์ไดรฟ์

1) Unit Setting คือ นำฐานของฮาร์ดดิสก์ไดรฟ์วางลงในชุดจับยึดชิ้นงาน เพื่อเป็นฐานในการประกอบชุดแผ่นข้อมูล คือ ประกอบแผ่นเก็บข้อมูล แหวนรองระหว่างแผ่นข้อมูลกับฐาน (Spacer ring) และแผ่นเก็บข้อมูลแผ่นที่สองลงบนฐานของฮาร์ดดิสก์ไดรฟ์

2) Top Clamp Mount & Screw Fastening คือ ประกอบแผ่นครอบแกนหมุน (Top clamp) กันแผ่นข้อมูลหลุด เพื่อยึดแผ่นเก็บข้อมูลและยึดแผ่นครอบแกนหมุนกับฐานของฮาร์ดดิสก์ไดรฟ์

3) Balance Check คือ ตรวจสอบแผ่นเก็บข้อมูลว่ามีการเอียงของแผ่นเก็บข้อมูลเกิดขึ้นหรือไม่

4) Ramp Mount & Screw คือ ประกอบอุปกรณ์พักหัวอ่าน (Ramp) และใช้สกรูยึดติดกับฐานของฮาร์ดดิสก์ไดรฟ์ และนำไปดูดฝุ่นจากการขันสกรู

5) HSA/VCM Mount คือ ประกอบชุดหัวอ่านตามด้วย Voice Coil Motor (VCM) และขันสกรูยึดชุดหัวอ่านกับฐาน และนำไปดูดฝุ่นจากการขันสกรู

6) VCM Screw Fastening คือ ขันสกรูยึด VCM และดูดฝุ่นหลังจากขันสกรู

7) Flex Screw Fastening คือ ขันสกรูยึด Flex Cable ที่ติดมากับชุดหัวอ่านยึดกับฐานของฮาร์ดดิสก์ไดรฟ์ และนำไปดูดฝุ่นจากการขันสกรู

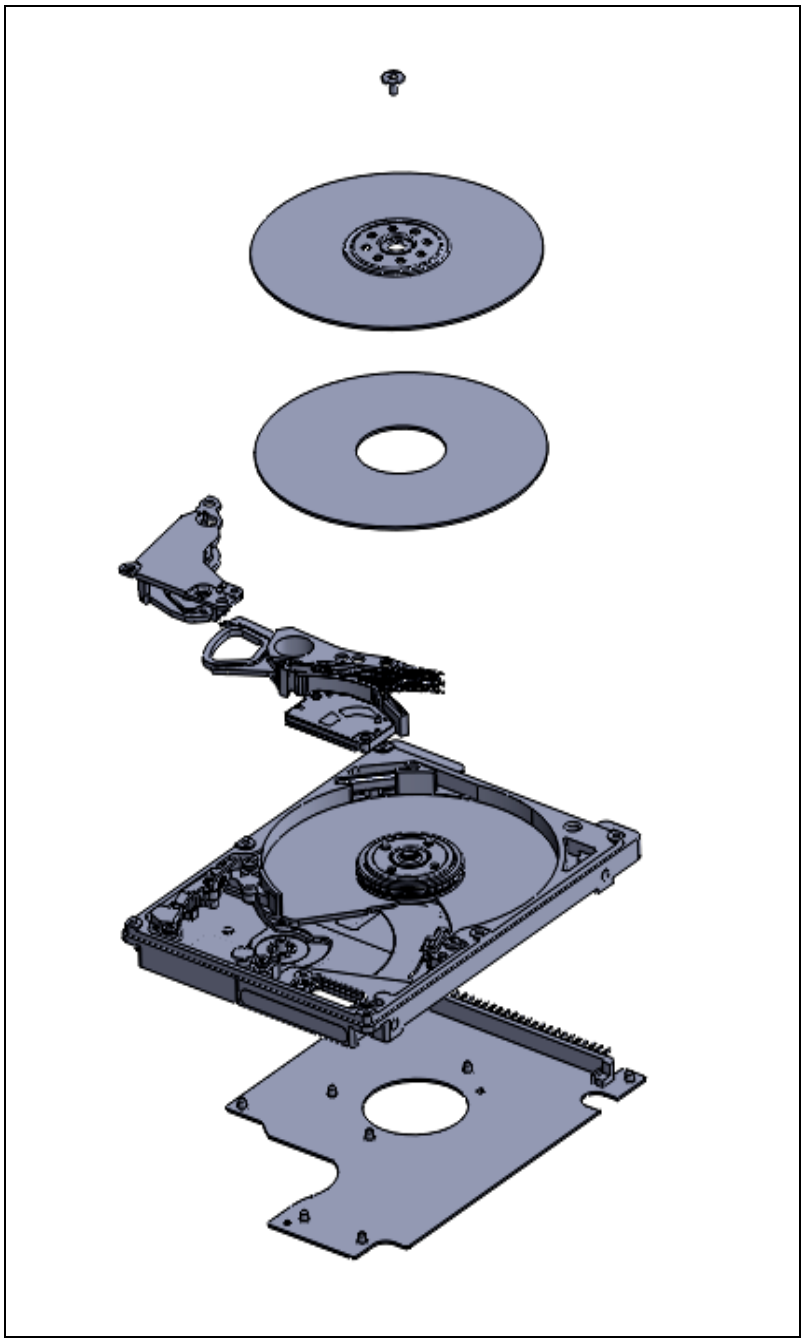
8) Outer Crash Stop Mount คือ ประกอบอุปกรณ์ป้องกันไม่ให้ชุดหัวอ่านตกจากอุปกรณ์พักหัวอ่านทั้งสองด้านในขณะที่แผ่นข้อมูลหยุดหมุน และป้องกันไม่ให้ชุดหัวอ่านเข้าไปชนกับแหวนรองระหว่างแผ่นข้อมูลกับฐาน และแผ่นครอบแกนหมุนในขณะที่แผ่นข้อมูลหมุนที่เรียกว่า หยุดการชน (Clash stop) ลงไปใน VCM และถอด Head clip ออกมา

9) Latch/Long Mount คือ ประกอบสลัก (Short Latch) และคาน (Long Lever) ลงในฐานของฮาร์ดดิสก์ไดรฟ์ เพื่อป้องกันไม่ให้ชุดหัวอ่านตกจากอุปกรณ์พักหัวอ่านมาสัมผัสกับแผ่นข้อมูล ในขณะที่แผ่นข้อมูลหยุดหมุน

10) Auto Gang Vacuum คือ ทำความสะอาดจากการขันสกรูด้วยการดูดอีกครั้ง

11) QC Gate คือ ทำการตรวจสอบอุปกรณ์ภายในฮาร์ดดิสก์ไดรฟ์ว่ามีครบหรือไม่ และนำฝาครอบบนมาปิดฐานของฮาร์ดดิสก์ไดรฟ์

12) Top Cover Screw Fastening คือ ขันสกรูยึดฝาครอบบนกับฐานของฮาร์ดดิสก์



รูปที่ 2.2 การประกอบฮาร์ดดิสก์ไดรฟ์

2.1.3 กระบวนการถอดเพื่อแก้ไขฮาร์ดดิสก์ไดรฟ์

1. Top cover screw remove คือ ทำการขันสกรูที่บริเวณฝาครอบบนออก
2. Latch & Long lever remove คือ การถอดประกอบชุดสลักและคานที่ป้องกันไม่ให้ชุดหัวอ่านตกจากอุปกรณ์พักหัวอ่านมาสัมผัสกับแผ่นข้อมูล ขณะที่แผ่นข้อมูลหยุดหมุน

3. Flex & VCM screw remove คือ ทำการขันสกรูที่ยึด Flex และ VCM ออก

4. HSA/VCM remove คือ การขันสกรูที่ยึดชุดหัวอ่านกับฐานออกเพื่อที่จะทำการแยกชุดหัวอ่าน และ VCM

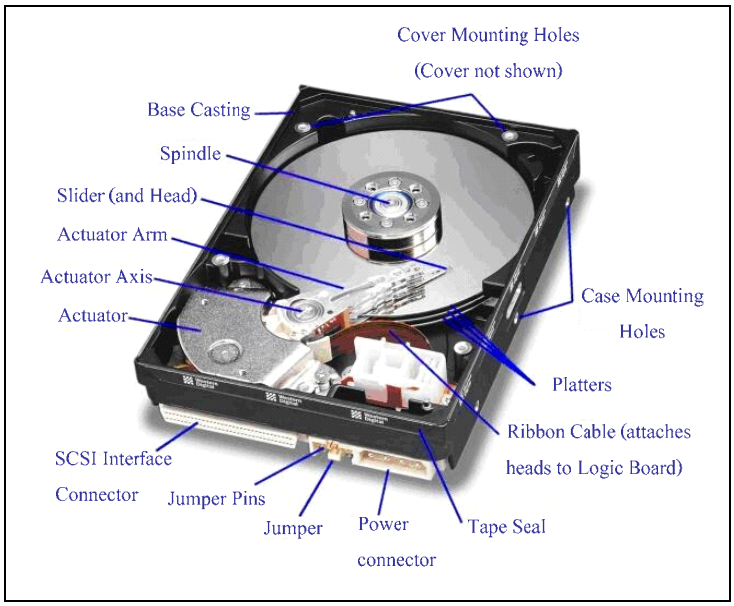
5. Top clamp remove คือ การถอดแผ่นครอบแกนหมุนที่ยึดแผ่นเก็บข้อมูลออก

6. Disk/Spacer ring remove คือ การถอดประกอบส่วนของแผ่นข้อมูลและแหวนรองระหว่างแผ่นข้อมูลกับฐาน

7. Serial number removal คือ การนำป้ายที่บ่งชี้หมายเลขฮาร์ดดิสก์ไดรฟ์ออก

2.2 ส่วนประกอบฮาร์ดดิสไดรฟ์

ส่วนนี้จะอธิบายหน้าที่ และการทำงานของส่วนประกอบในฮาร์ดดิสก์ไดรฟ์เฉพาะส่วนประกอบที่เกี่ยวข้องสำคัญในกระบวนการถอด HSA/VCM



รูปที่ 2.3 องค์ประกอบของฮาร์ดดิสก์ไดรฟ์

2.2.1 ชุดหัวอ่าน (Head Stack Assembly : HSA)

ชิ้นส่วนประกอบของชุดหัวอ่านประกอบไปด้วย หัวอ่านและเขียนข้อมูล แขน แกนชุดหัวอ่าน (Pivot) VCM ชุดขดลวดมอเตอร์ โดยชุดหัวอ่านทำหน้าที่อ่านและเขียนสัญญาณแม่เหล็กลงบนแผ่นข้อมูล โดยอุปกรณ์ที่ทำหน้าที่อ่านและเขียนสัญญาณแม่เหล็ก ภายใน

ตัวชุดหัวอ่านคือ หัวอ่านและเขียนข้อมูล ซึ่งตัวหัวอ่านและเขียนข้อมูลมีโครงสร้างภายในที่สลับซับซ้อนมาก เนื่องจากเป็นการนำชั้นสารหลาย ๆ ชั้นมารวมกัน ซึ่งถือว่าเป็นหัวใจหลักของการอ่านและเขียนข้อมูล โดยเมื่อมีการอ่านสัญญาณหัวอ่านและเขียนข้อมูลที่ทำการอ่าน (Read head) จะนำสัญญาณส่งมาขยายที่ตัวควบคุมการอ่านและเขียนสัญญาณของหัวอ่านและเขียนข้อมูล (Arm Electronic Module : AE Module) เพื่อให้ได้ความแรงของสัญญาณ หลังจากนั้นจึงส่งสัญญาณไปที่ตัวเชื่อมต่อข้อมูล (Pin Connector) เพื่อเชื่อมต่อให้ชุดประมวลผลและควบคุมการทำงาน (Card) ถอดรหัสและทำการประมวลผลต่อไป

ในการเขียนสัญญาณข้อมูลจากชุดประมวลผลและควบคุมการทำงานจะถูกส่งผ่านมาทางตัวเชื่อมต่อข้อมูล โดยมีตัวควบคุมการอ่านและเขียนสัญญาณของหัวอ่านและเขียนข้อมูลทำหน้าที่ในการเลือกหัวที่ใช้ในการเขียนสัญญาณ (Write Head) หลังจากนั้นสัญญาณจะถูกส่งไปยังหัวเขียนที่ได้เลือกไว้เพื่อทำการเขียนสัญญาณลงบนแผ่นข้อมูล

รูปที่ 2.4 ชุดหัวอ่านและส่วนประกอบ

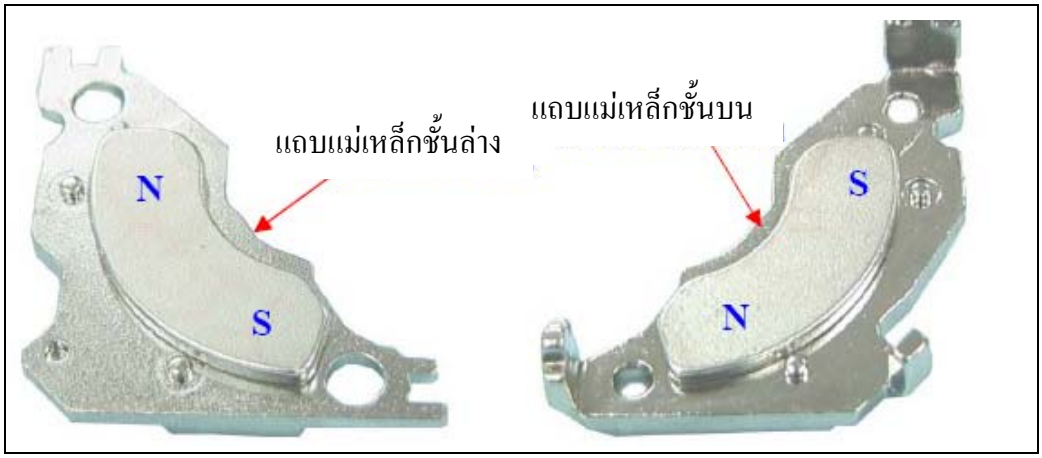
2.2.2 Voice Coil Motor (VCM)

VCM เป็นส่วนประกอบหนึ่งที่ช่วยทำให้ชุดหัวอ่านเกิดการเคลื่อนที่ โดยโครงสร้างของตัว VCM นั้นประกอบด้วยแม่เหล็ก 2 ชั้น แต่ละชั้นจะประกอบด้วยขั้วแม่เหล็ก 2 ขั้ว คือ ขั้วเหนือกับขั้วใต้ เมื่อประกอบเป็นตัว VCM แม่เหล็ก 2 ชั้นที่มีขั้วต่างกันจะเกิดการดูดกัน (จากหลักการของแม่เหล็กขั้วที่เหมือนกันจะผลักกัน ส่วนขั้วที่ต่างกันจะดูดกัน) เมื่อนำมาประกอบกับชุด HSA ที่มีชุดขดลวดเป็นส่วนฐานและทำหน้าที่ในการสร้างสนามแม่เหล็กขึ้นทำให้บริเวณรอบ ๆ VCM มีสนามแม่เหล็กเกิดขึ้น ดังนั้นขั้วแม่เหล็กที่เหมือนกันจะเกิดแรงผลักกัน ส่วนขั้วที่ต่างกันจะเกิดแรงดูดกัน จึงทำให้เกิดแรงดูดและแรงผลักขึ้นรอบ ๆ สนามแม่เหล็ก

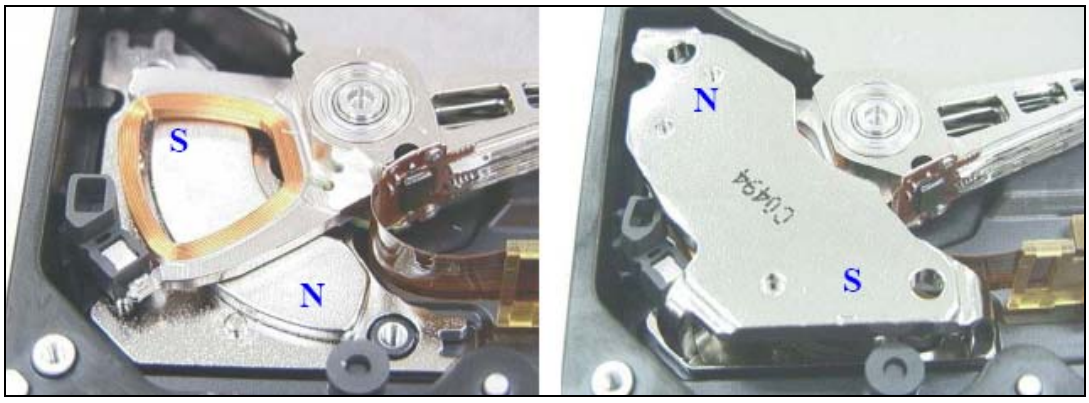
และ VCM ส่งผลให้ชุดหัวอ่านเกิดการเคลื่อนที่ ส่วนการเคลื่อนที่ของชุดหัวอ่านจะช้าหรือเร็วขึ้นอยู่กับปริมาณกระแสไฟฟ้าที่จ่ายให้กับชุดขดลวด

การจัดเรียงขั้วแม่เหล็กระหว่างขั้วเหนือและขั้วใต้นั้นแตกต่างกัน แสดงดังรูปที่ 2.5 เนื่องจากต้องการให้แถบแม่เหล็กชั้นบนและชั้นล่างมีขั้วที่ต่างกัน เพราะเมื่อประกอบเป็นตัว VCM แล้ว ขั้วทั้งสองจะได้ประกบกันแน่นเนื่องจากแรงดึงดูดของแม่เหล็ก สำหรับการประกอบ VCM เข้ากับชุดหัวอ่านนั้นชุดขดลวดจะอยู่ตรงกลางระหว่างแถบแม่เหล็กชั้นบนและชั้นล่าง

รูปที่ 2.5 VCM



รูปที่ 2.6 ลักษณะของ VCM โดยแม่เหล็ก 1 ชั้นประกอบด้วยขั้วแม่เหล็ก 2 ขั้ว



รูปที่ 2.7 การประกอบชุดหัวอ่านเข้ากับ VCM

2.2.3 Flax cable

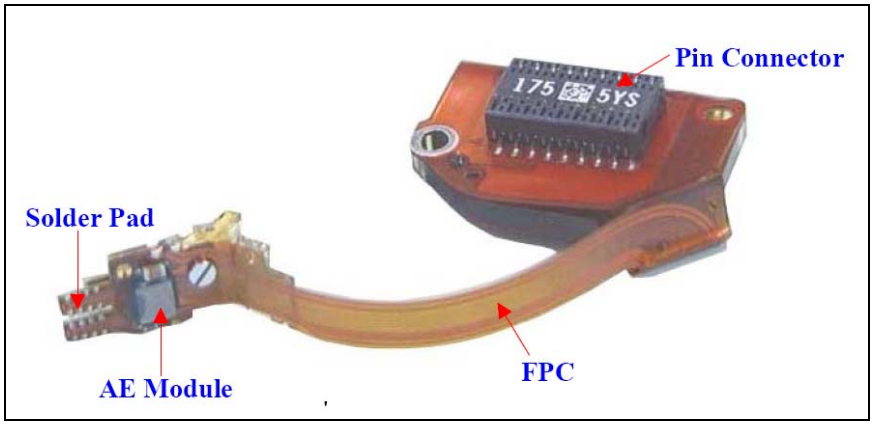
flax cable เป็นอุปกรณ์รับส่งสัญญาณการอ่านเขียนข้อมูลระหว่าง HGA กับชุดประมวลผลและควบคุมการทำงาน โดยมีตัวควบคุมการอ่านและเขียนสัญญาณของหัวอ่านและเขียนข้อมูลทำหน้าที่เลือกหัวอ่านและเขียนข้อมูลในการเขียนสัญญาณลงบนแผ่นข้อมูล และขยายสัญญาณที่ได้จากการอ่านส่งเข้าชุดประมวลผลและควบคุมการทำงานเพื่อทำการประมวลผล โดยส่วนประกอบของ flax cable แบ่งได้เป็น 4 ส่วนคือ ตัวเชื่อมสัญญาณ (Solder pad) : ตัวควบคุมการอ่านและเขียนสัญญาณของหัวอ่านและเขียนข้อมูล ตัวเชื่อมต่อข้อมูลและแผ่นวงจรพิมพ์ของ flax ซึ่งแต่ละส่วนมีหน้าที่ต่าง ๆ ดังนี้

1) ตัวเชื่อมสัญญาณ เป็นจุดเชื่อมสัญญาณระหว่างตัวควบคุมการอ่านและเขียนสัญญาณกับหัวอ่านและเขียนข้อมูล

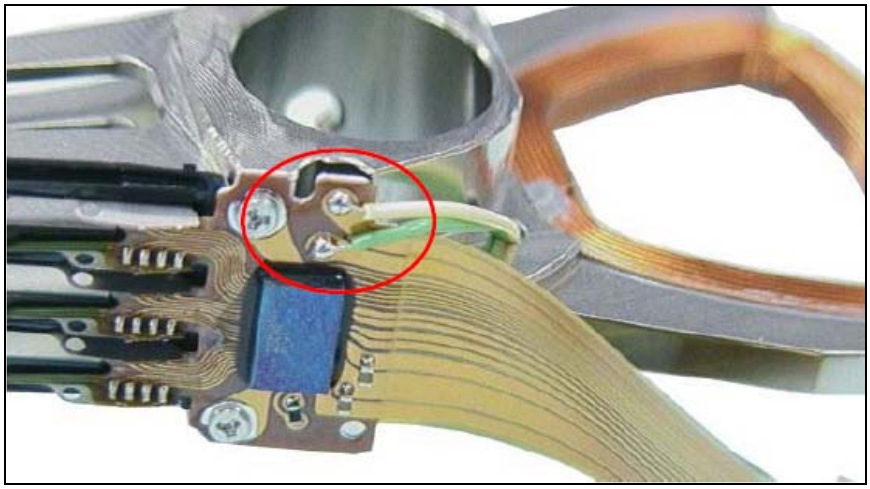
2) ตัวควบคุมการอ่านและเขียนสัญญาณของหัวอ่านและเขียนข้อมูล ทำหน้าที่ควบคุมการอ่านและเขียนสัญญาณของหัวหัวอ่านและเขียนข้อมูล เมื่อมีการอ่านสัญญาณหัวอ่านจะถูกเลือกใช้งาน โดยภาครูปแบบควบคุม (Mode control) ของตัวควบคุมการอ่านและเขียนสัญญาณของหัวอ่านและเขียนข้อมูล ซึ่งสัญญาณที่ได้จะถูกส่งไปยังภาคขยายสัญญาณการอ่าน (Read amplifier) เพื่อขยายสัญญาณให้มีความแรงและลดปัญหาสัญญาณรบกวน หลังจากนั้นจึงส่งสัญญาณให้ชุดประมวลผลและควบคุมการทำงานเพื่อทำการถอดรหัสและประมวลผลต่อไป ในลักษณะเดียวกันเมื่อมีคำสั่งให้เขียนสัญญาณ ภาครูปแบบควบคุมจะทำการเลือกหัวเขียนแต่ละหัวเพื่อทำการเขียนสัญญาณลงบนแผ่นข้อมูล จากหลักการดังกล่าวจึงทำให้ฮาร์ดดิสก์ไดรฟ์สามารถอ่านและเขียนสัญญาณได้ทีละหลายหัว หรือทั้งอ่านและเขียนในเวลาเดียวกัน

3) ตัวเชื่อมต่อข้อมูล ทำหน้าที่เชื่อมต่อข้อมูลระหว่างหัวอ่านและเขียนกับชุดประมวลผลและควบคุมการทำงาน และนำกระแสไฟฟ้าจากชุดประมวลผลและควบคุมการทำงานไปจ่ายให้ กับชุดขดลวดเพื่อควบคุมการเคลื่อนที่ของชุดหัวอ่าน

4) แผ่นวงจรพิมพ์ของ Flex คือ วงจรไฟฟ้าแบบอ่อน แบ่งออกเป็น 3ชนิด คือ Single Sided เป็นแผ่นวงจรที่ใช้งานได้ด้านเดียว Double Side เป็นแผ่นวงจรที่ใช้งานได้สองด้านและ Multi Layered เป็นแผ่นวงจรที่รวมลายวงจรหลาย ๆ ชั้นเข้าด้วยกัน



รูปที่ 2.8 ส่วนประกอบของ flax cable



รูปที่ 2.9 จุดต่อของชุดขดลวดบน flax cable

2.2.4 สกรู (Screw)

ในกระบวนการประกอบฮาร์ดดิสก์ไดรฟ์ การยึดชิ้นส่วนต่าง ๆ ให้ติดกัน ส่วนมากจะใช้สกรูในการยึด เริ่มตั้งแต่ชิ้นส่วนที่อยู่ภายในจนถึงชิ้นส่วนภายนอกฮาร์ดดิสก์ไดรฟ์ เช่น การยึด Flex Cable ให้ติดกับชุดขดลวด หรือการยึดฝาครอบบนให้ติดกับฐาน เป็นต้น ซึ่งขนาดและรูปร่างของสกรูแต่ละชนิดจะมีขนาดและลักษณะที่แตกต่างกันไป ไม่ว่าจะเป็นขนาดเส้นผ่านศูนย์กลาง ความยาว หรือขนาดเกลียว ทั้งนี้ก็ขึ้นอยู่กับลักษณะการยึดชิ้นส่วน โดยการประกอบสกรูเข้ากับชิ้นส่วนต่าง ๆ นั้นจะใช้ไขควง ซึ่งมีไขควงที่ใช้อยู่ 2 แบบคือ ไขควงไฟฟ้าที่ตั้งค่าแรงบิดได้ (Electric Torque Driver) และไขควงมือที่ตั้งค่าแรงบิดได้ (Manual Torque Driver)

สำหรับการขันยึดสกรูในแต่ละชิ้นส่วน จะมีการกำหนดค่าแรงบิดในการขันสกรูเป็นค่ากำหนด เพราะถ้าไม่ได้ค่าตามที่กำหนดไว้อาจทำให้แน่นหรือหลวมจนเกินไป และอาจทำให้ฮาร์ดดิสก์ไดรฟ์เกิดความเสียหายได้ สกรูที่ใช้ในฮาร์ดดิสก์ไดรฟ์มีชื่อเรียกตามตำแหน่งที่ใช้ขัน

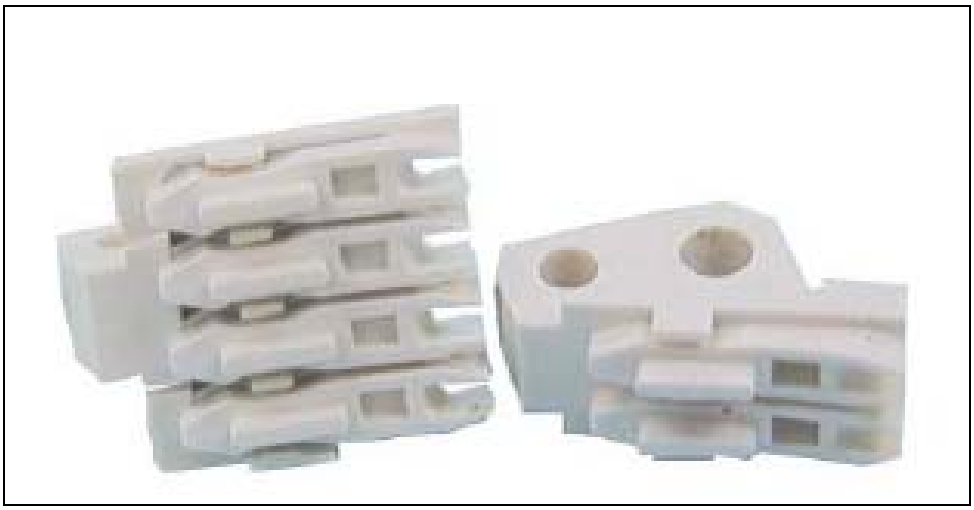
รูปที่ 2.10 ลักษณะตัวอย่างของสกรูภายในฮาร์ดดิสก์ไดรฟ์

2.2.5 อุปกรณ์พักหัวอ่าน

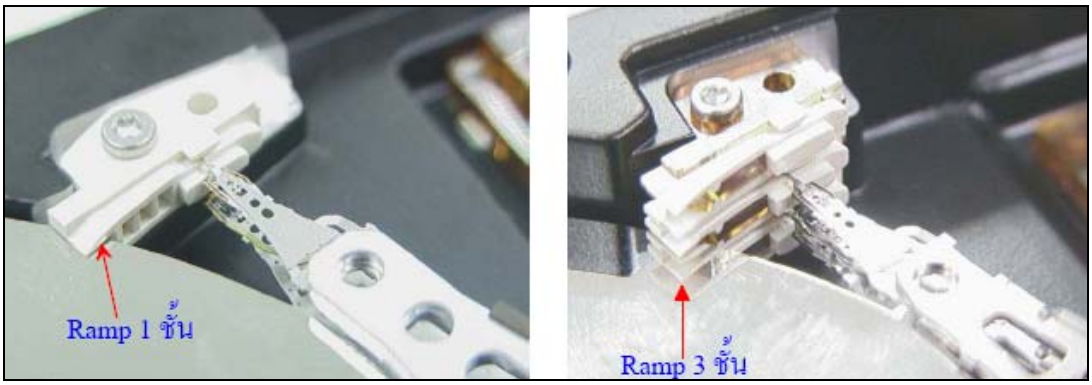
การทำงานของฮาร์ดดิสก์ไดรฟ์ในสภาวะที่ไม่มีการอ่านและเขียนสัญญาณนั้น ตัวชุดหัวอ่านจะถูกสั่งให้เคลื่อนที่ไปอยู่ ณ ตำแหน่งวงในสุดของแผ่นข้อมูล ซึ่งเรียกว่า Inner Diameter (ID) วิธีการพักหัวชุดหัวอ่านแบบนี้เรียกว่า Contact Start-Stop (CSS) วิธีการนี้ชุดแกนหมุนแผ่นข้อมูลต้องขับเคลื่อนให้แผ่นข้อมูลหมุนตลอดเวลาเพื่อสร้างแรงลมในการยกหัวอ่านและเขียนข้อมูลให้ลอยอยู่เหนือแผ่นข้อมูล ซึ่งจะทำให้ฮาร์ดดิสก์ไดรฟ์ต้องทำงานอยู่ตลอดเวลาเมื่อไม่มีการอ่านและเขียนสัญญาณ และทำให้สูญเสียเนื้อที่บางส่วนในการเก็บข้อมูลเพราะต้องเหลือพื้นที่ดังกล่าวไว้สำหรับให้ชุดหัวอ่านเข้าไปพัก ดังนั้นจึงได้นำเทคโนโลยีของอุปกรณ์พักหัวอ่านเข้ามาใช้เป็นที่พักชุดหัวอ่านเมื่อไม่มีการอ่านและเขียนสัญญาณ วิธีการนี้เรียกว่า Load/Unload Technology

โดยเมื่อไม่มีคำสั่งให้อ่านและเขียนสัญญาณขดลวดที่อยู่บริเวณชุดแกนหมุนแผ่นข้อมูลจะกำเนิดพลังงานไฟฟ้าจ่ายให้กับขดลวดของชุดหัวอ่านเพื่อควบคุมให้ชุดหัวอ่านเคลื่อนที่ไปจอดที่ อุปกรณ์พักหัวอ่าน

จำนวนชั้นของอุปกรณ์พักหัวอ่านที่ใช้ในแต่ละรุ่นงานนั้นขึ้นอยู่กับจำนวนหัว HGA ที่ใช้ โดยอุปกรณ์พักหัวอ่าน 1 ชั้นสามารถพักหัว HGA ได้ 2 หัว นั่นหมายความว่าถ้ามี HGA จำนวน 8 หัว อุปกรณ์พักหัวอ่านจะต้องมีความสูง 4 ชั้นนั่นเอง



รูปที่ 2.11 ลักษณะของอุปกรณ์พักหัวอ่านขนาดต่าง ๆ



รูปที่ 2.12 การพักหัวอ่านบนอุปกรณ์พักหัวอ่านขนาดต่าง ๆ

2.2.6 Head clip

การอ่านและเขียนสัญญาณแม่เหล็กบนแผ่นข้อมูล สามารถอ่านและเขียนสัญญาณแม่เหล็กได้ทั้ง 2 ด้าน ดังนั้นตัวชุดหัวอ่านจึงต้องมีทั้ง HGA ตัวบนและ HGA ตัวล่างซึ่งเมื่อประกอบเป็นชุดหัวอ่านเสร็จสมบูรณ์แล้วทิศทางของหัวอ่านและเขียนข้อมูลทั้งสองจะประกบเข้าหากัน และถ้าไม่มีเครื่องมือป้องกันจะทำให้หัวอ่านและเขียนข้อมูลติดกันได้และส่งผลให้เกิดความเสียหายขึ้นกับหัวอ่านและเขียนข้อมูล ดั้งนั้นตัวชุดหัวอ่านจึงต้องมีอุปกรณ์สำหรับกั้นกลางระหว่าง HGA ตัวบนและ HGA ตัวล่างก่อนที่จะทำการเคลื่อนย้ายไปสู่กระบวนการประกอบกับชิ้นส่วนอื่น ๆ ซึ่งหลังจากที่ประกอบแล้วต้องทำการถอด head clip ออก เนื่องจากมีอุปกรณ์พักหัวอ่านซึ่งเป็นที่พักของชุดหัวอ่านป้องกันไม่ให้หัวอ่านและเขียนข้อมูลติดกันอยู่แล้ว

รูปที่ 2.13 Head clip

รูปที่ 2.14 การใส่ head clip บนชุดหัวอ่าน

ความเสียหายของส่วนประกอบในฮาร์ดดิสก์ไดรฟ์ในกระบวนการผลิต

กระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ความเสียหายของส่วนประกอบในฮาร์ดดิสก์ไดรฟ์เกิดมาจากหลายสาเหตุ คือ แรงกระแทก การสั่นสะเทือน ไฟฟ้าสถิต และฝุ่น

2.3.1 แรงกระแทก (Shock)

ขั้นตอนต่าง ๆ ในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ทั้งการประกอบ และถอดประกอบชิ้นส่วนออกมาเพื่อแก้ไขจะต้องมีการกระแทกเกิดขึ้นกับชิ้นงาน ซึ่งค่าแรงกระแทกที่กระทำกับชิ้นงานจะทำให้ชิ้นงานเสียหาย ดังนั้นในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์จึงให้ความสำคัญต่อค่าแรงกระแทกที่เกิดขึ้นในกระบวนการ ซึ่งค่าแรงกระแทกที่ยอมรับได้ของบริษัทผู้ผลิตจะคิดในหน่วยของความเร่งที่กระทำกับชิ้นงาน โดยเปรียบเทียบว่าเป็นกี่เท่ากับความเร่งเนื่องจากแรงดึงดูดของโลก

2.3.2 การสั่นสะเทือน (Vibration)

การขนส่งชิ้นงานในกระบวนการผลิต ชิ้นงานจะสั่นสะเทือนอย่างต่อเนื่องในระหว่างการขนส่ง ซึ่งเป็นสาเหตุให้ส่วนประกอบต่าง ๆ ภายในฮาร์ดดิสก์ไดรฟ์เกิดความเสียหายได้ เช่น ชิ้นส่วนบางชิ้นส่วนเกิดการคลายตัวของเกลียวที่ขันยึด ทำให้ส่วนประกอบบางชิ้นแยกจากกัน และเกิดรอยขีดข่วนจากการเสียดสี

2.3.3 การปนเปื้อน (Contamination)

ในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ การปนเปื้อนจากสิ่งแปลกปลอมที่เข้าไปปะปนกับสายการผลิตและส่งผลกระทบให้เกิดความเสียหายกับตัวงานหรือกระบวนการผลิตโดยสิ่งปนเปื้อนที่เป็นปัจจัยสำคัญที่ส่งผลกระทบต่อฮาร์ดดิสก์ไดรฟ์ คือ ฝุ่นละอองของแข็ง สารประกอบไอออนิก สารประกอบออแกนนิก และสารหรือฝุ่นที่มีคุณสมบัติทางแม่เหล็ก สิ่งปนเปื้อนเหล่านี้มีแหล่งที่มาจากหลายสาเหตุ คือ มนุษย์ วัสดุอุปกรณ์ที่ใช้ในกระบวนการผลิต การทำงาน และจากอากาศภายนอกที่เข้าสู่พื้นที่ควบคุมของกระบวนการผลิต วิธีควบคุมสิ่งปนเปื้อนในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ มีความจำเป็นต้องควบคุมสภาวะความสะอาดต่าง ๆ ให้อยู่ในระดับที่ไม่ก่อให้เกิดความผิดพลาดในการผลิต โดยการควบคุมการพื้นที่ในการทำงานให้อยู่ในสภาวะความสะอาด ที่เรียกว่า ห้องสะอาด (Clean room) ซึ่งบริเวณห้องสะอาดควรที่มีการควบคุมสภาพแวดล้อมของห้องอย่างเหมาะสมในเรื่องต่าง ๆ ดังนี้ รูปแบบการไหลของอากาศ (Air flow pattern) อนุภาคแขวนลอยในอากาศ อุณหภูมิ ความดัน ความชื้น แสง การสั่น และปัจจัยอื่น ๆ ที่ส่งผลกระทบต่อคุณภาพของผลิตภัณฑ์

2.3.4 ไฟฟ้าสถิต (Electrostatic Discharge : ESD)

ไฟฟ้าสถิต คือ ประจุไฟฟ้าที่เกิดขึ้นจากความไม่สมดุลย์ของอิเล็กตรอนบนพื้นผิวของวัสดุ ซึ่งความไม่สมดุลย์ของอิเล็กตรอนทำให้เกิดสนามไฟฟ้าที่สามารถวัดได้ และสนามไฟฟ้านี้ก็จะมีผลหรือมีอิทธิพลต่อวัสดุที่อยู่รอบ ๆ ไฟฟ้าสถิตสามารถเปลี่ยนแปลงคุณลักษณะทางไฟฟ้าของอุปกรณ์ประเภทสารกึ่งตัวนำ ไม่ว่าจะทำให้มันแย่ลงหรือว่าทำลายให้เกิดความเสียหาย ซึ่งไฟฟ้าสถิต อาจทำให้ระบบอิเล็กทรอนิกส์เสียหาย ทำให้อุปกรณ์อิเล็กทรอนิกส์ทำงานผิดปกติหรือล้มเหลว ปัญหาอื่นที่เกิดจากไฟฟ้าสถิตนั้นคือปัญหาในห้องสะอาด ทำให้พื้นผิววัสดุที่มีประจุดึงดูดฝุ่นละอองไว้ ทำให้ยากต่อการทำความสะอาดหรือเอาออก เมื่อฝุ่นเหล่านี้ถูกดูดเข้าไปติดพื้นผิวของเวเฟอร์ที่ทำจากซิลิกอนหรือวงจรทางไฟฟ้า ฝุ่นเหล่านี้ก็จะทำให้เกิดการเสียของเวเฟอร์และทำให้ผลผลิตต่ำ การที่จะควบคุมไฟฟ้าสถิตต้องเริ่มต้นด้วยความเข้าใจว่าไฟฟ้าสถิตเกิดขึ้นอย่างไรก่อน ประจุไฟฟ้าสถิตส่วนใหญ่เกิดจากการสัมผัสแล้วแยกของวัสดุอย่างเดียวกันหรือคนละอย่าง ตัวอย่างเช่นเวลาคนเดินไปตามพื้นก็จะทำให้เกิดไฟฟ้าสถิตเพราะว่าส้นรองเท้าแตะแล้วก็แยกจากพื้น อุปกรณ์อิเล็กทรอนิกส์ที่บรรจุเข้าหรือนำออกจากบรรจุภัณฑ์ก็ทำให้เกิดไฟฟ้าสถิต แม้ว่าขนาดของประจุไฟฟ้าสถิตจะต่างกันในกรณีดังกล่าว แต่แน่นอนที่สุดคือประจุไฟฟ้าสถิตได้เกิดขึ้นแล้วความเสียหายที่เกิดจากไฟฟ้าสถิตกับอุปกรณ์อิเล็กทรอนิกส์สามารถเกิดได้ตั้งแต่ ผู้ผลิตจนถึง ณ จุดการใช้งาน ความเสียหายเป็นผลมาจากการจับฉวยหรือเข้าใกล้อุปกรณ์นั้น ๆ โดยไม่มีการระมัดระวังเรื่องการควบคุมไฟฟ้าสถิตเป็นอย่างดีพอ โดยทั่วไปแล้วความหายเสียถูกแบ่งได้เป็น 2 กลุ่มคือ กลุ่มที่เรียกว่าความเสียหายทันที (catastrophic failure) และกลุ่มที่เรียกว่าความเสียหายแบบแฝง (latent defect)

หลักการพื้นฐานการควบคุมไฟฟ้าสถิต

1) การออกแบบที่สามารถต้านทานการประทุของไฟฟ้าสถิตเป็นอย่างดี

2) กำจัดและลดการเกิดไฟฟ้าสถิต โดยการกำจัดหรือลดวิธีการที่ทำให้เกิดไฟฟ้าสถิต จัดการให้วิธีการและวัสดุที่ใช้ในการผลิตมีศักย์ไฟฟ้าเทียบเท่ากัน และจัดการให้มีการกราวด์ที่เหมาะสมเพื่อลดการเกิดและการสะสมประจุไฟฟ้าสถิต

3) ทำให้การถ่ายเทประจุเป็นไปอย่างช้า ๆ และเป็นกลาง โดยการกราวด์ การไอออนไนเซชั่น และการใช้วัสดุที่เป็นตัวนำ

4) ป้องกันผลิตภัณฑ์จากไฟฟ้าสถิต โดยการกราวด์ที่เหมาะสม และใช้วัสดุการบรรจุที่ป้องกันไฟฟ้าสถิตได้

2.4 แรงกระแทกและการวิเคราะห์สัญญาณการกระแทก

2.4.1 แรงกระแทก

ในขณะที่วัตถุกำลังเคลื่อนที่แล้วมีแรงมาต้านเพื่อให้วัตถุหยุด จะพบว่าวัตถุที่มีความเร็วมาก แรงที่ใช้ต้านต้องมีค่ามาก หรือหยุดวัตถุนั้นได้ยากกว่าเมื่อวัตถุนั้นมีความเร็วน้อย การที่หยุดวัตถุนั้นได้ยากแสดงว่าวัตถุนั้นมีความพยายามที่พุ่งไปข้างหน้ามาก ในขณะเดียวกันความพยายามที่จะพุ่งไปข้างหน้าของวัตถุจะขึ้นกับมวลของวัตถุนั้นด้วย เช่น รถบรรทุกขณะวิ่งด้วยความเร็วเท่ากับรถสามล้อ แสดงว่ารถบรรทุกมีความพยายามพุ่งไปข้างหน้ามากกว่ารถสามล้อ ถึงแม้ความเร็วเท่ากันแต่มวลต่างกัน มวลกับความเร็วจึงเป็นส่วนประกอบของความพยายามที่วัตถุที่จะพุ่งไปข้างหน้าเรียกว่าโมเมนตัม

โมเมนตัม หมายถึง ปริมาณที่บอกสภาพการเคลื่อนที่ของวัตถุ ขึ้นกับมวลและความเร็วของวัตถุ

$$\vec{P} = m\vec{v} \tag{2.1}$$

ปริมาณที่เกิดจากการที่มีแรงกระทำต่อวัตถุในช่วงเวลาหนึ่งแล้วทำให้วัตถุนั้นเกิดการเคลื่อนที่ หรือมีความเร็วเกิดขึ้น หรือไม่เกิดขึ้น และมีผลทำให้วัตถุมีการเปลี่ยนแปลงโมเมนตัมเรียกว่า การดล โดยการดลหาได้จากสมการที่ (2.2) หรือพื้นที่ใต้กราฟของแรงกับเวลา ถ้าแรงกระทำมีค่าไม่คงที่ ดังรูปที่ 2.7

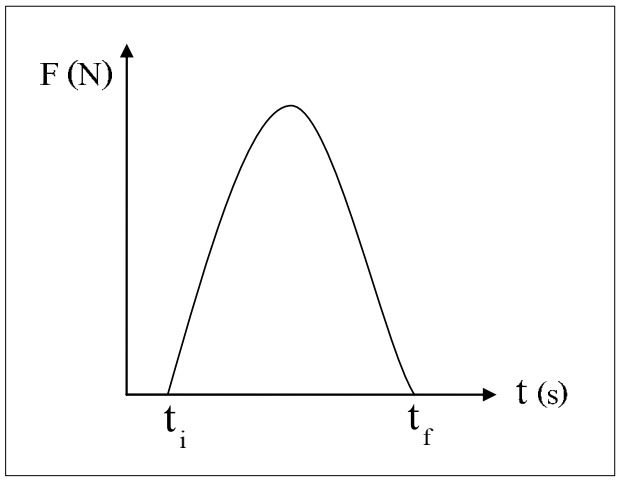
$$\vec{P} = m\vec{v} - m\vec{u} \tag{2.2}$$

โดยที่ P คือ โมเมนตัมเชิงเส้นของวัตถุ

m คือ มวลของวัตถุ

v คือ ความเร็วก่อนการชนของวัตถุ

u คือ ความเร็วหลังการชนของวัตถุ



รูปที่ 2.15 หาค่าการดลจากกราฟแรงและเวลา

ค่าแรงที่เกิดขึ้นจากอัตราการเปลี่ยนแปลงโมเมนตัมนั้นเรียกอีกอย่างหนึ่งว่าแรงดล (Impulse) ซึ่งจากกฎการเคลื่อนที่ข้อที่ 2 ของนิวตัน จะได้สมการแรงดลคือ

$$\sum F = ma \tag{2.3}$$

เมื่อ $a = \frac{(v-u)}{t}$ แทนค่าลงในสมการที่ (2.3) จะได้

$$\sum F = m\left(\frac{v-u}{\Delta t}\right) = \frac{mv-mu}{\Delta t} \tag{2.4}$$

ดังนั้นสรุปได้ว่า แรงลัพธ์ที่กระทำต่อวัตถุจะเท่ากับอัตราการเปลี่ยนโมเมนตัมของวัตถุนั้น หรืออาจให้นิยามของแรงได้ว่า แรงคืออัตราการเปลี่ยนโมเมนตัม เขียนในรูปสมการได้คือ

$$\sum F = \frac{mv-mu}{\Delta t} = ma$$

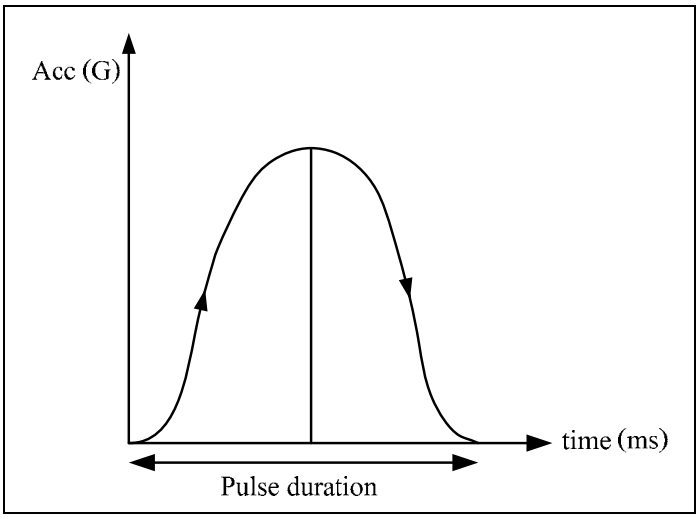
2.4.2 การวิเคราะห์สัญญาณการกระแทก

ในการวัดค่าแรงกระแทกด้วยวิธีหาสัญญาณการกระแทก (Shock pulse) เป็นการบันทึกค่าระหว่างความเร่ง (acceleration) กับเวลา (time) โดยใช้ตัวหยั่งสัญญาณความเร่ง (Accelerometer) ทดสอบ โดยติดตัวหยั่งสัญญาณความเร่งบนชิ้นงานบริเวณตำแหน่งตามแนวแกนที่ต้องการทราบค่า สัญญาณการกระแทกจะเป็นรูป “half sine” ความสูงของสัญญาณ หรือค่าความเร่งสูงสุดเป็นตัวบ่งชี้ความเสียหาย (damage potential) ของผลิตภัณฑ์ เพราะมีความสัมพันธ์กับแรงกระแทกสูงสุด (maximum force impact) จากกฎของนิวตัน (Newton’s laws) ในส่วนของความกว้างของสัญญาณเป็นตัวแสดงช่วงเวลาของการเกิดแรงกระแทก (impact force) ดังนั้นค่าความเร่งสูงสุดที่เกิดขึ้นมาจาก

$$\text{Acceleration} = \frac{\text{velocity change}}{\text{time}} \tag{2.5}$$

กราฟสัญญาณการกระแทกเป็นการบันทึกระหว่างความเร็วฉับพลัน (instantaneous acceleration) กับเวลาระหว่างการกระแทก ดังนั้นค่า acceleration หาได้จาก

$$\text{Acceleration} = \frac{dV}{dt} \tag{2.6}$$



รูปที่ 2.16 การวิเคราะห์สัญญาณการกระแทกจากความเร่งกับเวลา

นอกจากนี้ในการคำนวณหาค่าความเร่งเฉลี่ย (average acceleration) เพื่อเปรียบเทียบว่าเป็นกี่เท่าของความเร่งเนื่องจากแรงดึงดูดของโลก โดยเปลี่ยน acceleration เป็นค่า G หาได้จาก

$$\text{Average G} = \frac{(\text{velocity change})}{\text{shock duration}} \times \frac{1}{\text{g}} \tag{2.7}$$

$$\text{Average G} = \frac{\Delta V}{t} \times \frac{1}{g} \tag{2.8}$$

2.5 เครื่องจักรอัตโนมัติ

อุตสาหกรรมการผลิตในอดีตใช้แรงงานมนุษย์ สัตว์ รวมทั้งพลังงานจากธรรมชาติในกระบวนการผลิต เมื่อเริ่มมีการปฏิวัติอุตสาหกรรม จึงเริ่มมีการเปลี่ยนแปลงวิธีการ และระบบการผลิตมาเป็นการใช้เครื่องมือและเครื่องจักรกลแบบง่าย ๆ จนถึงแบบสลับซับซ้อน ปัจจุบันได้มีการนำระบบการทำงานแบบอัตโนมัติมาใช้ในชีวิตประจำวันมาก และภาคอุตสาหกรรมมากขึ้นทุก ๆ วันนี้ระบบการทำงานที่เป็นระบบอัตโนมัติจะถูกคิดค้นเพิ่มขึ้น เพื่อเพิ่มความสะดวกให้กับมนุษย์ และเมื่อเราพิจารณาถึงสิ่งที่ได้รับจากการนำเครื่องจักรแบบอัตโนมัติมาใช้ในภาคอุตสาหกรรม จะพบว่ามีการทำงานที่สะดวกและง่ายดายขึ้น ทำให้กระบวนการผลิตมีประสิทธิภาพ และรวดเร็วขึ้น อีกทั้งสินค้าก็จะมีมาตรฐานที่ดี การใช้เครื่องจักรอัตโนมัติในการทำงานยังสามารถช่วยลดความเสียหายที่เกิดจากการทำงานของมนุษย์ลงไปได้

เครื่องจักรอัตโนมัติ (Automation) คือ เครื่องจักรที่มีการนำคอมพิวเตอร์เข้ามาช่วยในการทำงาน ทำให้เครื่องจักรสามารถทำงานได้เอง โดยอาจเป็นเพียงส่วนหนึ่งของขั้นตอนการผลิตเท่านั้น หรือเป็นการผลิตแบบอัตโนมัติทั้งระบบเลยก็ได้ มนุษย์จะมีหน้าที่ในการออกคำสั่งและดูแลเครื่องจักรเท่านั้น

ประเภทของเครื่องจักรอัตโนมัติ แบ่งออกได้เป็น 2 ประเภท คือ

1. ระบบเครื่องจักรกึ่งอัตโนมัติ ได้แก่ เครื่องจักรที่นำเอาคอมพิวเตอร์เข้ามาช่วยในการควบคุมหรือการทำงานในบางขั้นตอน ทั้งนี้ เพื่อให้มนุษย์สามารถควบคุมการทำงานบางประเภท เช่น การทำงานที่มีอันตรายสูง การทำงานที่ต้องอาศัยความละเอียดแม่นยำ เป็นต้น โดยขั้นตอนอื่น ๆ ที่เหลือจะยังคงเป็นการทำงานที่อาศัยมนุษย์เป็นหลัก

2. ระบบเครื่องจักรอัตโนมัติ ได้แก่ เครื่องจักรที่นำเอาคอมพิวเตอร์เข้ามาช่วยควบคุมหรือทำงานในทุกขั้นตอน โดยจะให้มนุษย์มีหน้าที่ออกคำสั่ง ดูแล รักษาระบบเครื่องเท่านั้น การทำงานโดยใช้เครื่องจักรอัตโนมัติทุกขั้นตอนจะเหมาะกับงานที่ต้องการการควบคุมอย่างเต็มที่ไม่ว่าจะเป็นด้านคุณภาพหรือความสะอาด อีกทั้งยังใช้ในงานที่ไม่สามารถใช้มนุษย์ทำได้ เช่น งานที่ต้องใช้อุณหภูมิสูง ๆ เกินกว่าที่คนเราจะสามารถทนได้ เช่น เป็นร้อย เป็นพันองศาเซลเซียส เป็นต้น

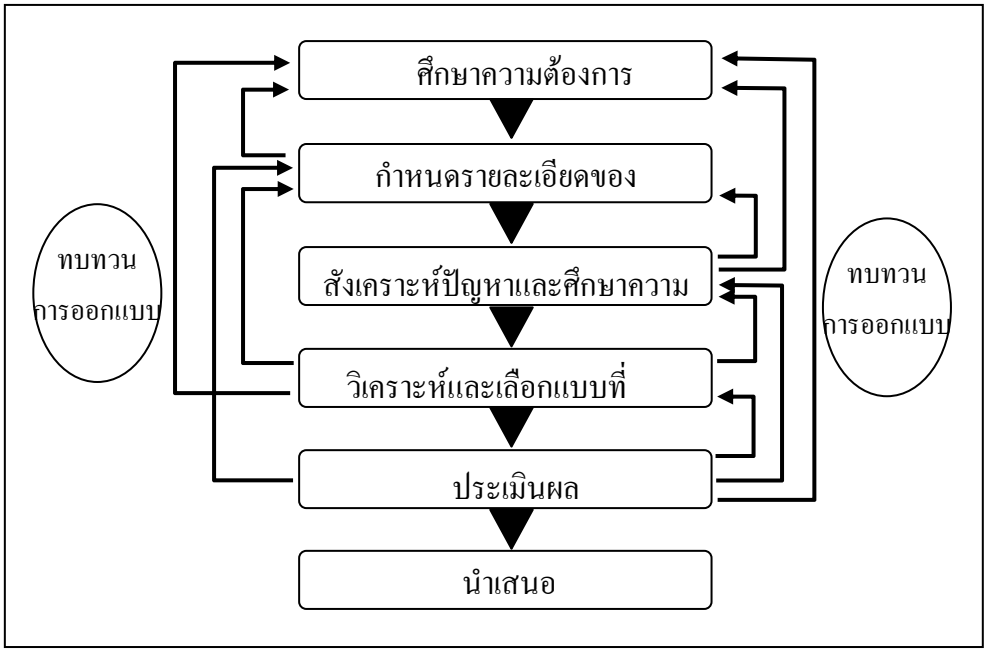
2.6 การออกแบบเครื่องจักรกล

การสร้างเครื่องจักรกล ก่อนสร้างจะต้องผ่านกระบวนการการออกแบบ ซึ่งกระบวนการออกแบบจะต้องคำนึงถึงรายละเอียดต่าง ๆ ของเครื่องจักรกลที่จะสร้างขึ้น โดยจะต้องกำหนดข้อมูลพื้นฐานทั้งหมดของเครื่องจักรกลที่จะสร้างขึ้นมาก่อน เช่น ต้องกำหนดหน้าที่พื้นฐาน ต้องพิจารณาถึงกลไกการทำงานและโครงสร้างที่จำเป็น ต้องกำหนดรูปทรงรายละเอียดพื้นฐาน ขนาด วัสดุ และสิ่งอื่น ๆ อีกมากมาย ผลที่ได้จากการออกแบบ ก็คือ แบบวาดที่พร้อมสำหรับการสร้างเครื่องจักร

2.6.1 กระบวนการในการออกแบบ

กระบวนการในการออกแบบจะมีขั้นตอนดังแสดงตามรูปที่ 2.17 (Joseph E.S., Charles R.M., and Richard G.B., 1989) โดยจะเริ่มตั้งแต่การศึกษาความต้องการ และตัดสินใจว่าจะทำอย่างไร และหลังจากที่ทำการออกแบบซ้ำไปซ้ำมาหลาย ๆ ครั้งกระบวนการจะสิ้นสุดลงที่การมีแผนดำเนินงานซึ่งตรงกับความต้องการ

ขั้นตอนในการออกแบบหลาย ๆ ขั้นตอน อาจจะจำเป็นต้องมีการทบทวน และพิจารณาอย่างละเอียดหลาย ๆ ครั้ง ต้องสร้างแบบวาดที่ไม่มีข้อผิดพลาดหากมีการออกแบบผิดพลาดหรือแนวความคิดพื้นฐานการออกแบบผิดพลาดเครื่องจักรกลที่สร้างขึ้นก็จะเป็นเครื่องจักรกลที่ไม่สามารถทำงานได้ตามต้องการ



รูปที่ 2.17 ขั้นตอนในการออกแบบ

2.6.2 ข้อควรพิจารณาในการออกแบบ

2.6.2.1 ความแข็งแรง

ความแข็งแรงของเครื่องจักรกลเป็นสิ่งสำคัญที่สุดในการสร้างเครื่องจักรกล แรงที่กระจะต้องไม่เกินขอบเขตที่วัสดุที่ใช้ทำโครงสางเครื่องจักรกลจะรับได้ จะต้องเพิ่มความแข็งแกร่งของเครื่องจักร เพื่อไม่ให้เกิดการเสียรูปทรงจนเกินไปเมื่อมีการรับแรง ความแข็งแรงนั้นจะเกี่ยวข้องกับสิ่งต่าง ๆ ดังต่อไปนี้

1) ความแข็งแรง

ความแข็งแรงสถิต แรงที่กระทำจะต้องไม่เกินขอบเขตที่วัสดุที่ประกอบเป็นโครงสร้างเครื่องจักรกลจะรับได้ ความล้า ในกรณีที่รับแรงกระทำซ้ำ ๆ ถึงแม้ว่าความเค้นที่กระทำจะมีขนาดต่ำกว่าขนาดของความเค้นที่ยอมรับได้ แต่หากต้องรับแรงกระทำซ้ำ ๆ ตลอดเวลาวัสดุก็สามารถแตกหักได้

2) ความแกร่ง

การงอ (ความแกร่งเชิงสถิต) หากมีแรงมากระทำบนวัสดุ จะทำให้เกิดการเปลี่ยนรูปทรง ไม่ว่าจะเป็นการยืด การหด การงอ และอื่น ๆ การที่จะรักษาหน้าที่และคุณสมบัติของเครื่องจักรไว้ การเปลี่ยนตำแหน่งต่าง ๆ เมื่อเทียบกับแรงที่กระทำจะต้องมีขนาดไม่มากจนเกินไป ซึ่งมีความจำเป็นในการเพิ่มความแกร่งของเครื่องจักรกล

ความถี่เฉพาะ (ความแกร่งเชิงจลน์) วัตถุทุกชนิดจะมีความถี่เฉพาะของตนเอง การสั่นที่ความถี่เฉพาะจะเกิดขึ้นเมื่อวัตถุถูกกระทบหรือถูกกระตุ้นจากภายนอก หากเครื่องจักรกลรับการสั่นจากภายนอกเข้ามา หรือเครื่องจักรกลรวมแหล่งกำเนิดการสั่น หากความถี่ของการสั่นจากภายนอกตรงกับความถี่เฉพาะของเครื่องจักรกล พลังงานการสั่นอย่างรุนแรงจะถูกสะสมในเครื่องจักรกล การสั่นของเครื่องจักรกลก็จะมากขึ้นเรื่อย ๆ ในที่สุดเครื่องจักรกลอาจจะพังได้ ปรากฏการณ์ที่เกิดขึ้นเรียกว่า เรโซแนนซ์ การป้องกันการเกิดเรโซแนนซ์เครื่องจักรกลจะต้องออกแบบให้มีค่าความถี่เฉพาะสูง โดยมีค่าสูงกว่าความถี่รบกวนที่เข้ามา

2.6.2.2 พื้นที่และน้ำหนัก

พื้นที่ ชิ้นส่วนจะต้องวางไว้ภายในพื้นที่ที่กำหนดภายในเครื่องจักรกล หรือภายในพื้นที่ที่กำหนดของเครื่องจักรกล เป็นข้อกำหนดบังคับที่เกี่ยวกับพื้นที่ เช่นความสูงจะต้องไม่เกินความสูงที่กำหนดไว้

น้ำหนัก น้ำหนักจะเกี่ยวข้องโดยตรงกับข้อกำหนดเกี่ยวกับพื้นที่ด้วย น้ำหนักต้องออกแบบให้มีค่าต่ำกว่าเกณฑ์ของข้อกำหนดบังคับ

2.6.2.3 การสร้าง

ความเป็นไปได้ในการสร้างชิ้นงานให้ได้ตามรูปทรงที่ต้องการนั้น จะต้องออกแบบให้มีความเป็นไปได้ในการสร้าง จะต้องเลือกวิธีในการขึ้นรูปชิ้นงาน ซึ่งอาจจะเป็นวิธีการกลึง การเชื่อม หรือการหล่อ การสร้างชิ้นงานจำนวน 1 ชิ้นหรือทำการทดลองการสร้างนั้น โดยทั่วไปจะใช้การกลึงเอาเนื้อวัสดุออก ผู้ออกแบบต้องออกแบบรูปทรงให้สามารถกลึงได้ และชิ้นงานจะต้องสร้างได้ง่าย โดยเลือกกลไกและโครงสร้างที่ง่ายต่อการสร้าง หากเลือกใช้วิธีที่ยากจะทำให้เสียเวลาและกำลังงาน และยังอาจก่อให้เกิดความผิดพลาดในการสร้าง

2.6.2.4 การประกอบและแยกชิ้นส่วน

เลือกกลไกและรูปทรงที่สามารถประกอบและแยกชิ้นส่วนได้ และจะต้องออกแบบพื้นที่ หรือรูปทรงที่สามารถสอดใส่เครื่องมือที่ใช้ในการดำเนินการประกอบได้ เนื่องจากการประกอบชิ้นส่วนต่าง ๆ เข้าด้วยกันตามลำดับ จะต้องมีการยึดชิ้นส่วนเข้าด้วยกันโดยใช้หมุดเกลียว ก็จะต้องมีการเผื่อพื้นที่ เพื่อให้สามารถสอดใส่เครื่องมือในการขันเกลียวได้ และระหว่างการประกอบจะต้องมีการตรวจสอบสภาพควบคู่ไปด้วย หากมีความผิดพลาดในการประกอบจึงมีความจำเป็นที่จะต้องแยกชิ้นส่วนบางชิ้นออกมาเพื่อตรวจสอบและแก้ไข หรือประกอบเข้าด้วยกันใหม่อีกครั้ง ดังนั้นเครื่องจักรกลที่สร้างขึ้นควรมีความสามารถในการแยกชิ้นส่วนเฉพาะส่วนเมื่อต้องการการแก้ไขเครื่อง

2.6.2.5 การควบคุมการทำงาน

เครื่องจักรกลจะต้องออกแบบให้ใช้งานได้ง่าย ขั้นตอนการควบคุมและรูปทรงจะต้องง่ายต่อการทำงาน เพื่อป้องกันการเกิดข้อผิดพลาดในการควบคุมและการใช้งาน

2.6.2.6 ความคงทนและความปลอดภัย

ความคงทน คือ เลือกวัสดุที่ไม่เป็นสนิม โดยการชุบ เคลือบสี หรือทาสี นอกจากนี้ต้องคำนึงถึงการป้องกันการเกิดฝุ่นละลองด้วย

ความปลอดภัย คือ ชิ้นส่วนที่หมุนหรือเคลื่อนที่ต้องติดตั้งฝาครอบ หรือติดตั้งสัญญาณเตือนเพื่อป้องกันไม่ให้มือแหย่เข้าไปในส่วนที่เคลื่อนที่ของเครื่องจักรกล ป้องกันความผิดพลาดในการควบคุมการทำงาน หรือการทำงานผิดพลาด จะต้องออกแบบเครื่องจักรกลไม่ให้เกิดความผิดพลาดในการควบคุมทำงาน และออกแบบวิธีป้องกันหากเกิดข้อผิดพลาดขึ้น ทำการจัดวางตำแหน่งและกำหนดทิศทางการทำงานให้เหมาะสม มีการติดตั้งสวิตซ์ฉุกเฉิน หากมีการทำงานผิดพลาดขึ้น และทำงานได้อย่างปลอดภัยถึงแม้ว่าเครื่องจะเสียแล้วก็ตามก็ยังจะต้องมีความปลอดภัย เช่น หากใบมีดขึ้นรูปของเครื่องกลึงแตกหัก ควรมีระบบแจ้งเตือนเพื่อหยุดเครื่องฉุกเฉินและความแน่นอนของชิ้นส่วนหรือวัสดุ จะต้องเลือกชิ้นส่วนที่สามารถถูกใช้ซ้ำหลาย ๆ ครั้งได้ มีความแน่นอนสูง และมีความความแข็งแรงอย่างเพียงพอ

2.6.2.7 มาตรฐานและข้อบังคับ

การออกแบบเครื่องจักรกลจะต้องเป็นไปตามมาตรฐาน และข้อบังคับของแต่ละชิ้นส่วน โดยมาตรฐานแล้วควรใช้ชิ้นส่วนและส่วนประกอบที่ใช้ร่วมกันได้ เช่น ขนาดของเกลียวสามารถถูกสร้างให้มีขนาดเท่าใดก็ได้ แต่ผู้ออกแบบควรเลือกใช้ขนาดที่เป็นมาตรฐาน เช่น มาตรฐานเมตริกซ์ที่ใช้ในญี่ปุ่น สำหรับอเมริกาและอังกฤษจะใช้มาตรฐานแบบนิ้ว ดังนั้นจึงไม่สามารถใช้หมุดเกลียวของอเมริกาประกอบชิ้นส่วนที่ผลิตในญี่ปุ่นได้ ส่วนข้องบังคับมีข้อกำหนดที่ถูกบังคับใช้ตามกฎหมาย โดยเฉพาะข้อกำหนดที่เกี่ยวข้องกับความปลอดภัย ซึ่งจะต้องใช้พิจารณาในการออกแบบเครื่องจักรกลในกรณีที่เกิดอุบัติเหตุระหว่างการใช้งาน ข้อบังคับทั้งหมดเกี่ยวกับความปลอดภัยจะมีอยู่ใน “ข้อบังคับความปลอดภัยในการทำงาน”

2.7 ระบบควบคุมอัตโนมัติ

ระบบควบคุมอัตโนมัติหมายถึง การทำงานของระบบหรือเครื่องจักรที่สามารถทำงานได้ด้วยตัวเองอย่างต่อเนื่อง เมื่อมีการให้สัญญาณเริ่มต้นการทำงาน โดยที่ระบบนั้นจะเป็นการทำงานแบบวนรอบวัฏจักรเดิมตลอด หรือสามารถเปลี่ยนแปลงรูปแบบการทำงานโดยใช้วิธีการนำ

สัญญาณอินพุตมาวิเคราะห์ประมวลผล และสั่งงานเอาต์พุตตามเงื่อนไขรูปแบบของโปรแกรมที่ออกแบบกำหนดไว้

วัตถุประสงค์ของการใช้ระบบควบคุมอัตโนมัติโดยทั่วไปคือ เพื่อเพิ่มประสิทธิภาพและคุณภาพในการผลิต เพิ่มผลผลิต และลดต้นทุนในการผลิต เนื่องจากสามารถลดจำนวนการใช้มนุษย์ในระบบการผลิต เพื่อความรวดเร็ว ถูกต้อง และแม่นยำของระบบการผลิตตลอดจนทำให้เครื่องจักรทำงานได้ประสิทธิภาพสูง

2.7.1 องค์ประกอบของระบบควบคุมอัตโนมัติ

องค์ประกอบของระบบควบคุมอัตโนมัติ เป็นการนำองค์ประกอบและหน้าที่ในการทำงานหลาย ๆ ส่วนมาต่อเชื่อมกันขึ้นเป็นระบบ เพื่อให้เกิดการตอบสนองตามที่ต้องการ โดยระบบจะมีส่วนควบคุมเป็นส่วนทำหน้าที่ควบคุมการทำงานของเครื่องจักรให้ได้ตามเป้าหมายที่ต้องการ ซึ่งการควบคุมระบบในงานอุตสาหกรรมจะมีองค์ประกอบหลัก ๆ ที่สำคัญคือ

1) สัญญาณอินพุต (Input Signal) เป็นสัญญาณที่ใช้ในการสั่งงานระบบ และรวมถึงการแสดงสถานการณ์ทำงานของระบบ เพื่อให้ส่วนที่ทำหน้าที่ควบคุมวิเคราะห์และประมวลผลสั่งงาน

2) กระบวนการหรือระบบที่ต้องการควบคุม (Process) เป็นระบบการทำงานหรือเครื่องจักรที่ต้องการควบคุมการทำงานด้วยระบบควบคุมอัตโนมัติ

3) สัญญาณเอาต์พุต (Output Signal) เป็นสัญญาณที่ทำหน้าที่ในการสั่งงานอุปกรณ์ต่าง ๆ ให้ทำงานตามที่ส่วนควบคุมกำหนด

4) สัญญาณป้อนกลับ (Feedback Signal) เป็นสัญญาณที่ทำหน้าที่ในการบอกถึงสถานะ การทำงานของระบบ เพื่อให้ส่วนที่ทำหน้าที่ควบคุมวิเคราะห์และประมวลผลสั่งงานอุปกรณ์ เป็นสัญญาณที่ใช้กับระบบควบคุมแบบปิด หรือระบบการควบคุมแบบป้อนกลับ

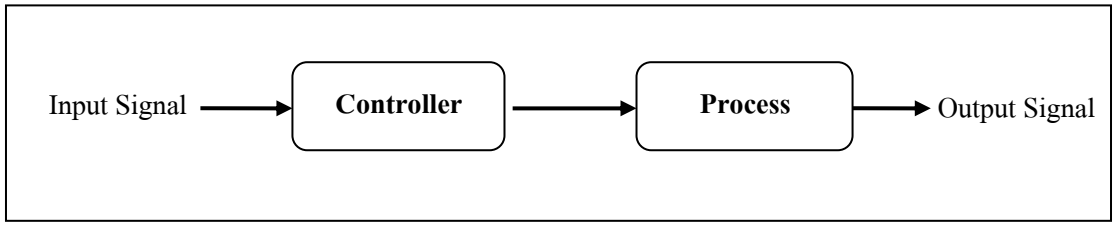
5) อุปกรณ์ควบคุม (Controller) เป็นอุปกรณ์ที่ทำหน้าที่ออกคำสั่ง หรือกำหนดสัญญาณควบคุมตามเงื่อนไขการควบคุมที่กำหนดไว้ล่วงหน้า คำสั่งหรือสัญญาณควบคุมนี้อาจจะเป็นฟังก์ชันการควบคุมเวลา หรือสัญญาณอินพุตที่ได้รับจากอุปกรณ์ตรวจวัด

ระบบการควบคุมอัตโนมัติสามารถแบ่งออกตามลักษณะการทำงาน ได้เป็น 2 แบบคือระบบการควบคุมแบบเปิด (Open loop control system) และระบบควบคุมแบบปิดหรือระบบการควบคุมแบบป้อนกลับ (Closed loop or Feedback control system)

2.7.1.1 ระบบการควบคุมแบบเปิด (Open loop control system)

ระบบการควบคุมแบบเปิดคือ ระบบควบคุมที่มีการควบคุมในลักษณะที่สั่งงานไปยังอุปกรณ์ควบคุมอย่างเดียวโดยไม่มีการอ่านค่าผลลัพธ์ของระบบป้อนกลับ เป็นระบบ

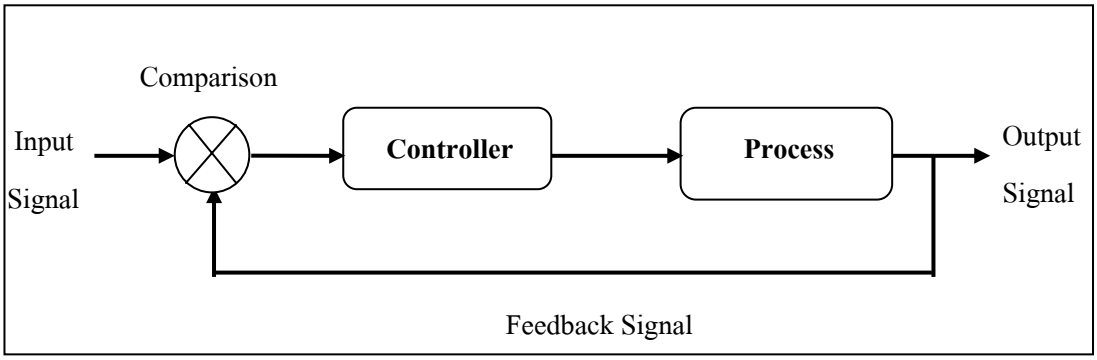
ที่มีการทำงานง่ายที่สุดและมีอุปกรณ์ภายในที่ไม่ยุ่งยาก ค่าเอาต์พุตที่ได้จะไม่มีผลต่อการควบคุมกระบวนการของระบบ คือ จะไม่มีการนำเอาต์พุตที่ได้กลับมาเปรียบเทียบกับค่าอินพุตที่ป้อนให้กับระบบ



รูปที่ 2.18 ระบบการควบคุมแบบเปิด

2.7.1.2 ระบบควบคุมแบบปิดหรือระบบการควบคุมแบบป้อนกลับ (Closed loop or Feedback control system)

ระบบควบคุมแบบปิด หรือระบบการควบคุมแบบป้อนกลับคือ ระบบควบคุมที่มีการควบคุมในลักษณะที่มีการสั่งงานไปยังเครื่องควบคุมแล้วมีการอ่านค่าผลลัพธ์ของระบบที่ป้อนกลับมา เพื่อเปรียบเทียบและประมวลผล แล้วจึงส่งสัญญาณเอาต์พุตสั่งงานควบคุมไปใหม่เพื่อให้ผลลัพธ์ของระบบเป็นตามเป้าหมายที่กำหนดไว้ตามเงื่อนไขการทำงานของระบบหรือเครื่องจักรนั้น ๆ



รูปที่ 2.19 ระบบควบคุมแบบปิดหรือระบบการควบคุมแบบป้อนกลับ

การเลือกใช้ประเภทของการควบคุม การควบคุมแบบเปิดจะเหมาะกับระบบที่ทราบว่าอินพุตของระบบจะมีการเปลี่ยนแปลงอย่างไร และจะต้องไม่มีสัญญาณรบกวนจากภายนอก เช่นระบบควบคุมที่ตัวแปรที่ต้องการควบคุม คือ ตำแหน่ง ความเร็ว และความเร่ง สำหรับการควบคุมแบบปิดเหมาะกับระบบที่ไม่ทราบการเปลี่ยนแปลงของอินพุตตามเวลา หรือกรณีที่มีสัญญาณจากภายนอก หรือการเปลี่ยนแปลงค่าพารามิเตอร์ของอุปกรณ์ในระบบโดยไม่คาดคิดมาก่อน เช่น ระบบควบคุมที่ตัวแปรที่ต้องการควบคุม คือ อุณหภูมิ ระดับของของเหลวหรือของแข็ง อัตราการไหล ความดัน แรง ส่วนผสม ค่าความเป็นกรดเป็นด่าง ความชื้น ความหนืด และความหนาแน่น ในบางกรณีระบบใด ๆ อาจจะมีทั้งระบบการควบคุมแบบเปิดและปิดร่วมกันได้

การออกแบบการใช้งานระบบการควบคุมอัตโนมัติของเครื่องจักรในปัจจุบันสำหรับเครื่องจักรแบบกึ่งอัตโนมัติ และแบบอัตโนมัติ สามารถออกแบบให้เลือกใช้ได้หลายประเภทขึ้นอยู่กับความเหมาะสม และสามารถใช้งานได้อย่างมีประสิทธิภาพ โดยประเภทระบบควบคุมอัตโนมัติที่ใช้ในการควบคุมเครื่องจักรมีหลายประเภท เช่น ไมโครตัวควบคุมเลอร์ PLC (Programmable Logic Controller) Distributed control System (DCS) และ Hybrid System เป็นต้น สำหรับการเลือกใช้ระบบควบคุมอัตโนมัติในการใช้งานจะต้องศึกษาเลือกรุ่น และขนาดให้เหมาะสมกับการควบคุมการทำงานของระบบและเครื่องจักร เนื่องจากอุปกรณ์แต่ละรุ่นจะมีขนาดของจำนวนอินพุตและเอาต์พุตที่แตกต่างกัน และมีข้อจำกัดความสามารถในการใช้งานแตกต่างกัน

2.7.2 ไมโครคอนโทรลเลอร์ (Microcontroller)

ระบบควบคุมการทำงานแบบไมโครคอนโทรลเลอร์ได้ถูกนำมาใช้ในงานอุตสาหกรรมเพื่อควบคุมเครื่องจักรและระบบการผลิตต่าง ๆ เช่น ควบคุมระบบในรถยนต์ ควบคุมหุ่นยนต์อุตสาหกรรม หรือควบคุมระบบขนถ่ายวัสดุ เป็นต้น

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำ คำหนึ่งคือ ไมโคร หมายถึง ขนาดเล็ก และคำว่า คอนโทรลเลอร์ หมายถึง ตัวควบคุมหรืออุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงหมายถึงอุปกรณ์ควบคุมขนาดเล็ก แต่ในตัวอุปกรณ์ควบคุมขนาดเล็กนี้ได้บรรจุความสามารถที่คล้ายคลึงกับระบบคอมพิวเตอร์ที่คนโดยส่วนใหญ่คุ้นเคย กล่าวคือ ภายในไมโครคอนโทรลเลอร์ได้รวมเอาหน่วยประมวลผลกลาง (Central Processing Unit : CPU) หน่วยความจำ (Memory) และพอร์ต (Port) ซึ่งเป็นส่วนประกอบหลักสำคัญของระบบคอมพิวเตอร์เข้าไว้ด้วยกัน โดยสามารถโปรแกรมการทำงานได้ซับซ้อน สามารถรับข้อมูลในรูปสัญญาณดิจิตอลเข้าไปทำการประมวลผลแล้วส่งผลลัพธ์ข้อมูลดิจิตอลออกมาเพื่อนำไปใช้งานตามที่ต้องการได้

ไมโครคอนโทรลเลอร์ภายในชิพจะมีหน่วยความจำ และพอร์ตอยู่ในชิพเพียงตัวเดียวซึ่งอาจจะเรียกได้ว่าเป็นคอมพิวเตอร์ชิพเดี่ยว ไมโครคอนโทรลเลอร์เป็นไมโครโพรเซสเซอร์ชนิดหนึ่ง เช่นเดียวกับหน่วยประมวลผลกลางที่ใช้ในคอมพิวเตอร์ แต่ได้รับการพัฒนาแยกออกมา

ภายหลังเพื่อนำไปใช้ในวงจรทางด้านงานควบคุม คือ แทนที่ในการใช้งานจะต้องต่อวงจรภายนอกต่าง ๆ เพิ่มเติมเช่นเดียวกับไมโครโปรเซสเซอร์ ก็จะทำการรวมวงจรที่จำเป็น เช่น หน่วยความจำ ส่วนอินพุต/เอาต์พุตบางส่วนเข้าไปในตัววงจรรวมเดียวกัน และเพิ่มวงจรบางอย่างเข้าไปด้วยเพื่อให้มีความสามารถเหมาะสมกับการใช้ในงานควบคุม เช่น วงจรตั้งเวลา วงจรการสื่อสารอนุกรม และวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอล เป็นต้น สรุปคือ ไมโครคอนโทรลเลอร์ เท่ากับ ไมโครโพรเซสเซอร์ + หน่วยความจำ + อินพุต/เอาต์พุต

ไมโครคอนโทรลเลอร์สามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวาง โดยมักจะเป็นการนำไปใช้ฝังในระบบของอุปกรณ์อื่น ๆ (Embedded Systems) เพื่อใช้ควบคุมการทำงานบางอย่าง เช่น ใช้ในรถยนต์ เตาอบไมโครเวฟ เครื่องปรับอากาศ หรือเครื่องซักผ้าอัตโนมัติ เป็นต้น เพราะว่าไมโครคอนโทรลเลอร์มีข้อดีเหมาะสมต่อการใช้ในงานควบคุมหลายประการ เช่น ชิพวงจรรวมและระบบที่ได้มีขนาดเล็ก ระบบที่ได้มีราคาถูกกว่าการใช้ชิพไมโครโพรเซสเซอร์ วงจรที่ได้จะมีความซับซ้อนน้อย ช่วยลดข้อผิดพลาดที่อาจจะเกิดขึ้นได้ในการต่อวงจร มีคุณสมบัติเพิ่มเติมสำหรับงานควบคุมโดยเฉพาะซึ่งใช้งานได้ง่าย และช่วยลดระยะเวลาในการพัฒนาระบบได้

2.7.2.1 โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์

โครงสร้างโดยทั่วไปของไมโครคอนโทรลเลอร์นั้น สามารถแบ่งออกมาได้เป็น 5 ส่วนใหญ่ ๆ ดังต่อไปนี้

1) หน่วยประมวลผลกลาง

2) หน่วยความจำ สามารถแบ่งออกเป็น 2 ส่วน คือ หน่วยความจำที่มีไว้สำหรับเก็บโปรแกรมหลัก (Program Memory) เปรียบเสมือนฮาร์ดดิสก์ไดรฟ์ของเครื่องคอมพิวเตอร์ตั้งโต๊ะ คือ ข้อมูลใด ๆ ที่ถูกเก็บไว้ในนี้จะไม่สูญหายไปแม้ไม่มีไฟเลี้ยง อีกส่วนหนึ่ง คือ หน่วยความจำข้อมูล (Data Memory) ใช้เป็นเหมือนกกระดาษทดในการคำนวณของหน่วยประมวลผลกลาง และเป็นที่พักข้อมูลชั่วคราวขณะทำงาน แต่หากไม่มีไฟเลี้ยงข้อมูลก็จะหายไป คล้ายกับหน่วยความจำแรม (RAM) ในเครื่องคอมพิวเตอร์ทั่วไป แต่สำหรับไมโครคอนโทรลเลอร์สมัยใหม่ หน่วยความจำข้อมูลจะมีทั้งที่เป็นหน่วยความจำแรม ซึ่งข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง และเป็นอีอีพรอม (EEPROM : Erasable Electrically Read-Only Memory) ซึ่งสามารถเก็บข้อมูลได้แม้ไม่มีไฟเลี้ยง

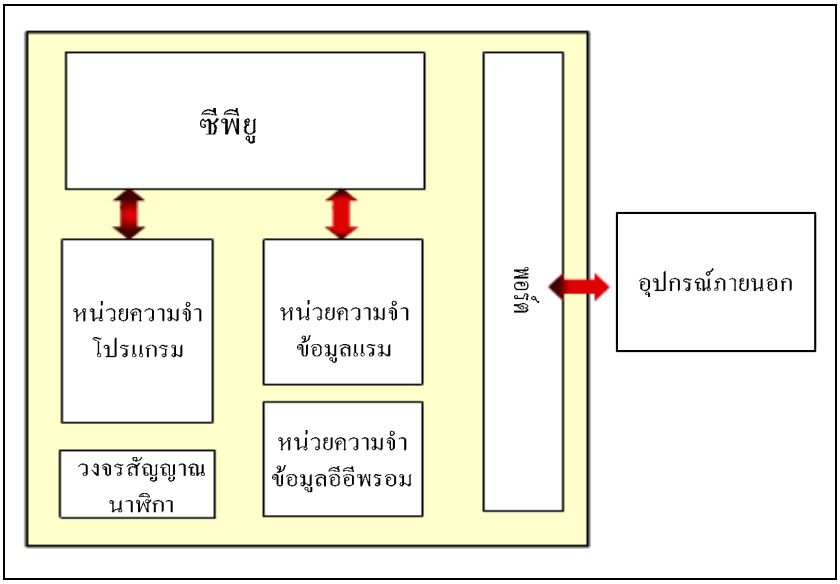
3) ส่วนติดต่อกับอุปกรณ์ภายนอก หรือพอร์ตมี 2 ลักษณะคือ พอร์ตอินพุต และพอร์ตส่งสัญญาณหรือพอร์ตเอาต์พุต ส่วนนี้จะใช้ในการเชื่อมต่อกับอุปกรณ์ภายนอก ถือว่าเป็นส่วนที่สำคัญมาก ใช้ร่วมกันระหว่างพอร์ตอินพุต เพื่อรับสัญญาณ อาจจะด้วยการกดสวิตช์ เพื่อนำไปประมวลผลและส่งไปพอร์ตเอาต์พุตเพื่อแสดงผล เช่น การติดสว่างของหลอดไฟ เป็นต้น

4) ช่องทางเดินของสัญญาณ หรือบัส (BUS) คือเส้นทางการแลกเปลี่ยนสัญญาณข้อมูลระหว่างหน่วยประมวลผลกลาง หน่วยความจำ และพอร์ต เป็นลักษณะของสายสัญญาณ จำนวนมากอยู่ภายในตัวไมโครคอนโทรลเลอร์โดยแบ่งเป็น บัสข้อมูล (Data Bus) บัสแอดเดรส (Address Bus) และบัสควบคุม (Control Bus)

บัสข้อมูลเป็นสายสัญญาณที่บรรจุข้อมูล เพื่อการประมวลผลทั้งหมด ขนาดของบัสจะขึ้นอยู่กับความสามารถของการประมวลผลของหน่วยประมวลผลกลาง สำหรับในงานทั่ว ๆ ไปขนาดของบัสข้อมูลจะเป็น 8 บิต และในปัจจุบันได้มีการพัฒนาขึ้นมาจนถึง 16 32 และ 64 บิต

บัสแอดเดรสเป็นสายสัญญาณที่บรรจุค่าตำแหน่งของหน่วยความจำ โดยการติดต่อกับหน่วยความจำนั้นหน่วยประมวลผลกลาง ต้องกำหนดตำแหน่งที่ต้องการอ่านหรือเขียนก่อน ดังนั้นจำนวนสายสัญญาณของแอดเดรสจึงต้องมีจำนวนมาก ยิ่งมากเท่าไหร่ก็จะเป็นการแสดงขนาดของหน่วยความจำที่ไมโครคอนโทรลเลอร์สามารถติดต่อได้ โดยสามารถคำนวณได้จาก จำนวนแอดเดรสของหน่วยความจำเท่ากับ 2^n เมื่อ n คือจำนวนของเส้นทาง

บัสควบคุมเป็นกลุ่มของสายสัญญาณควบคุมการติดต่อทั้งหมดของหน่วยประมวลผลกลางกับหน่วยความจำและพอร์ต สำหรับสายสัญญาณเลือกควบคุมหลักได้แก่ สายสัญญาณเลือกอ่านและเขียนหน่วยความจำ สายสัญญาณเลือกอ่านและเขียนข้อมูลกับพอร์ต



รูปที่ 2.20 โครงสร้างไมโครคอนโทรลเลอร์

5) วงจรกำเนิดสัญญาณนาฬิกา นับเป็นส่วนประกอบที่สำคัญมากอีกส่วนหนึ่ง เนื่องจากการทำงานที่เกิดขึ้นในตัวไมโครคอนโทรลเลอร์จะขึ้นอยู่กับการกำหนดจังหวะ หากสัญญาณนาฬิกามีความถี่สูง จังหวะการทำงานก็จะสามารถทำได้ถี่ขึ้น ส่งผลให้ไมโครคอนโทรลเลอร์ตัวนั้นมีความเร็วในการประมวลผลสูงตามไปด้วย

2.7.2.2 ภาษาที่ใช้สำหรับการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์

การเพิ่มคำสั่งในการควบคุมไมโครคอนโทรลเลอร์สามารถทำได้โดยการเขียนโปรแกรม โดยภาษาที่ใช้สำหรับการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์แบ่งได้เช่นเดียวกับการเขียนโปรแกรมบนคอมพิวเตอร์ คือ ภาษาระดับสูง และภาษาระดับต่ำ ภาษาระดับสูงเช่น ภาษาซี (C) และภาษาเบสิก (Basic) ข้อดี คือ เขียนง่าย แก้ไขเปลี่ยนแปลง หรือเพิ่มเติมได้ง่าย ส่วนข้อเสียก็คือ การทำงานจะช้า และขนาดโปรแกรมที่เขียนมีขนาดใหญ่ ภาษาระดับต่ำซึ่งก็คือ ภาษาแอสแซมบลี (Assembly) ข้อดีคือ โปรแกรมแปลภาษาที่ทำหน้าที่แปลภาษาระดับสูงมาเป็นภาษาเครื่อง (Compiler) แจกฟรี ขนาดโปรแกรมหลังจากแปลเป็นภาษาเครื่องแล้วมีขนาดเล็กและโปรแกรมมีความเร็ว แต่ข้อเสียก็คือ เขียนยาก เพราะลักษณะภาษาไม่ค่อยสื่อความหมายและแก้ไขเปลี่ยนแปลงยาก

2.7.2.3 ลักษณะงานที่เหมาะสมกับไมโครคอนโทรลเลอร์

1) งานที่เกี่ยวข้องกับการควบคุมหรือจัดการสัญญาณอินพุตและเอาต์พุต
2) เชื่อมต่อกับอุปกรณ์อินพุต/เอาต์พุตไม่เกิน 10 แบบ
3) ใช้ความเร็วในการประมวลผลไม่สูงมาก
4) ใช้หน่วยความจำในการเก็บข้อมูลไม่มาก
5) ต้องการแผงวงจรควบคุมที่มีขนาดเล็ก

2.7.2.4 ตระกูลของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์มีหลายขนาดตั้งแต่ 8 บิต จนในปัจจุบันได้มีการพัฒนาขึ้นมาจนถึง 16 32 และ 64 บิต แต่ขนาด 8 บิต ก็ยังได้รับความนิยมใช้งานกันมาก ไมโครคอนโทรลเลอร์มีหลายยี่ห้อ หลายตระกูล และหลายเบอร์ด้วยกัน ซึ่งแต่ละเบอร์ก็จะมีโครงสร้างภายใน และความสามารถในการทำงานที่แตกต่างกัน ทำให้เลือกใช้กับงานได้อย่างเหมาะสม มีหลายบริษัทได้ผลิตออกมาจำหน่ายให้เลือกใช้งาน ซึ่งในแต่ละตระกูลก็มีให้เลือกใช้หลายเบอร์ ดังแสดงในตารางที่ 2.1

ตารางที่ 2.1 ตระกูลไมโครคอนโทรลเลอร์

ตระกูล	บริษัทที่ผลิต
MCS - 51	Intel : Atmel : Dallas : NXP (Philips)
PIC	Microchip
68HC	Freescale (Motorola)
Z8	Zilog
AVR	Atmel

2.7.3 การติดต่อกับไมโครคอนโทรลเลอร์ด้วยไมโครซอฟท์วิชวลเบสิก 6.0

โปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 เป็นโปรแกรมสำหรับพัฒนาโปรแกรมประยุกต์ที่กำลังเป็นที่นิยมใช้อยู่ในปัจจุบัน ถูกออกแบบมาเพื่อทำงานบนระบบปฏิบัติการวินโดวส์ (Windows Operating System) โดยใช้ภาษาวิชวลเบสิก (Visual Basic) โดยภาษาวิชวลเบสิกพัฒนามาจากภาษาเบสิก (Beginner's All-purpose Symbolic Instruction Code : BASIC) ซึ่งเป็นภาษาคอมพิวเตอร์ระดับสูง ใช้งานง่าย เหมาะสำหรับผู้เริ่มใช้คอมพิวเตอร์ เพราะใช้คำในภาษาอังกฤษที่เข้าใจง่าย และเมื่อเป็นวิชวลเบสิก จึงใช้ในลักษณะของการมองเห็นได้ ซึ่งคำว่า วิชวล หมายถึงวิธีการที่ใช้สร้างติดต่อกับผู้ใช้แบบกราฟิกหรือรูปภาพ โดยโปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 ได้เปลี่ยนรูปแบบการเขียนโปรแกรมใหม่ โดยมีชุดคำสั่งมาสนับสนุนการทำงาน มีเครื่องมือต่าง ๆ ที่เรียกกันว่า ตัวควบคุมไว้สำหรับช่วยในการออกแบบโปรแกรม ที่เน้นการออกแบบหน้าจอแบบกราฟิก หรือที่เรียกว่า การติดต่อกับผู้ใช้แบบกราฟฟิก (Graphic User Interface : GUI) ทำให้การจัดรูปแบบหน้าจอเป็นไปได้ง่าย และในการเขียนโปรแกรมนั้นจะเขียนแบบ Event Driven Programming คือ โปรแกรมจะทำงานก็ต่อเมื่อเหตุการณ์เกิดขึ้น ตัวอย่างของเหตุการณ์ได้แก่ ผู้ใช้เลื่อนเมาส์ ผู้ใช้กดปุ่มบนคีย์บอร์ด และผู้ใช้กดปุ่มเมาส์ เป็นต้น จึงทำให้การพัฒนาโปรแกรมเพื่อนำมาใช้งานทำได้สะดวก รวดเร็ว และมีความสามารถสูง เหมาะสำหรับการพัฒนาโปรแกรมใช้งานหลายด้าน เช่น งานด้านคำนวณทั่วไป งานด้านฐานข้อมูล เกมส์ และการเชื่อมต่อกับอุปกรณ์ภายนอก หรือที่เรียกว่า ฮาร์ดแวร์อินเตอร์เฟส

การใช้งานไมโครคอนโทรลเลอร์เมื่อต้องการที่จะให้ข้อมูลของไมโครคอนโทรลเลอร์ไปแสดงผลที่คอมพิวเตอร์ หรือใช้คอมพิวเตอร์ในการควบคุมไมโครคอนโทรลเลอร์ การเชื่อมต่อระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ ที่เรียกว่าฮาร์ดแวร์อินเตอร์เฟส คือ การทำงานติดต่อสื่อสาร และโอนถ่ายข้อมูลร่วมกันระหว่างหน่วยประมวลผลกลางกับอุปกรณ์ต่าง ๆ นอกเหนือจากจะต้องทำงานติดต่อกับหน่วยความจำแล้วยังต้อง

มีการติดต่อกับอุปกรณ์ภายนอกที่มีการส่งข้อมูลอินพุต เอาต์พุตอีกทางหนึ่ง ซึ่งเป็นการเพิ่มประสิทธิภาพให้ระบบสมบูรณ์

การติดต่อสื่อสาร โอนถ่ายข้อมูลสามารถจำแนกได้ 2 แบบคือ แบบขนาน (Parallel) และแบบอนุกรม (Serial)

2.7.3.1 การสื่อสารข้อมูลแบบขนาน

การส่งข้อมูลแบบขนาน ทำได้โดยการส่งข้อมูลออกมาทีละไบต์ คือ 8 บิต จากอุปกรณ์ส่งไปยังอุปกรณ์รับ ตัวกลางระหว่าง 2 เครื่องจะต้องมีช่องทางให้ข้อมูลเดินทางอย่างน้อย 8 ช่องทาง โดยมากจะเป็นสายขนานให้กระแสไฟฟ้าวิ่งมากกว่าจะเป็นตัวกลางชนิดอื่น เนื่องจากมีสัญญาณสูญหายไปกับความต้านทานของสาย ระยะทางระหว่าง 2 เครื่องไม่ควรจะเกิน 100 ฟุต การส่งแบบขนานส่วนมากจะทำในระยะใกล้ ๆ เนื่องจากจะต้องมีช่องทางเดินของสัญญาณมากกว่า 8 สาย และอุปกรณ์ที่ติดต่อแบบขนานกับคอมพิวเตอร์ที่เห็นโดยทั่วไป คือ เครื่องพิมพ์

2.7.3.2 การสื่อสารข้อมูลแบบอนุกรม

ในการสื่อสารข้อมูลแบบอนุกรม ข้อมูลถูกส่งออกมาทีละบิตระหว่างจุดส่งและจุดรับ จะเห็นว่าการส่งข้อมูลแบบนี้ช้ากว่าแบบขนานตัวกลางการสื่อสารแบบอนุกรมต้องการเพียงช่องเดียวหรือสายเพียงคู่เดียว ค่าใช้จ่ายในสื่อกลางถูกกว่าแบบขนานสำหรับการส่งระยะทางไกล ๆ

การส่งข้อมูลแบบอนุกรม ข้อมูลจากจุดส่งจะถูกเปลี่ยนให้เป็นอนุกรมเสียก่อนแล้วค่อยทยอยส่งออกทีละบิตไปยังจุดรับ ณ ที่จุดรับจะต้องมีกลไกในการเปลี่ยนข้อมูลที่ส่งมาทีละบิตให้เป็นสัญญาณแบบขนานที่ลงตัวพอดี นั่นคือ บิต 1 ลงที่บัสข้อมูลเส้นที่ 1 พอดี การที่จะทำให้ การแปลงสัญญาณจากอนุกรมทีละบิตให้ลงตัวพอดีนั้นจำเป็นจะต้องมีกลไกที่เหมาะสมเพื่อป้องกันการผิดพลาดในการรับ การสื่อสารแบบอนุกรม สามารถแบ่งออกเป็น 3 รูปแบบดังนี้

1) Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว

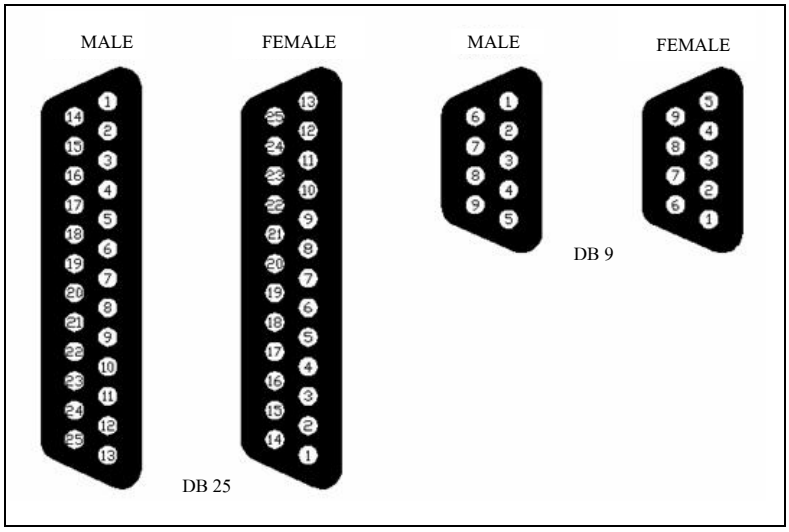
2) Half Duplex สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน

3) Full Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

การส่งข้อมูลแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะใช้มาตรฐานการถ่ายโอนข้อมูล และคำสั่งผ่านพอร์ตอนุกรม (Recommended Standard-232 : RS-232C) โดยมาตรฐานการถ่ายโอนข้อมูล และคำสั่งผ่านพอร์ตอนุกรมเป็นมาตรฐานเชื่อมต่อข้อมูลแบบอนุกรม ที่ได้รับการออกแบบมาเพื่อที่จะทำให้อุปกรณ์จากผู้ผลิตต่างกันสามารถทำงานร่วมกันได้ มาตรฐานหลายชนิดได้รับการออกแบบขึ้นมารองรับเช่นกัน แต่มาตรฐานที่ได้รับความนิยมและใช้

กันกว้างขวางมากที่สุดคือ มาตรฐานการถ่ายโอนข้อมูล และคำสั่งผ่านพอร์ตอนุกรมซึ่งถูกประกาศใช้ในปี 1969 โดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) มาตรฐานการถ่ายโอนข้อมูล และคำสั่งผ่านพอร์ตอนุกรมได้แบ่งอุปกรณ์ออกเป็น 2 ประเภท ทั้งนี้ก็เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลบนสายเส้นเดียวกัน ซึ่งอุปกรณ์ทั้งสองประเภทนี้ก็คือ อุปกรณ์ Data Terminal Equipment (DTE) เป็นอุปกรณ์สำหรับส่งข้อมูลอุปกรณ์และ Data Communication Equipment (DCE) เป็นอุปกรณ์สำหรับรับข้อมูล

ตามมาตรฐานการถ่ายโอนข้อมูล และคำสั่งผ่านพอร์ตอนุกรม ตัวเชื่อมต่อของ DTE จะเป็นตัวผู้ ส่วนตัวเชื่อมต่อของ DCE จะเป็นตัวเมีย ซึ่งตัวเชื่อมต่อที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้น จะมีอยู่ 2 ลักษณะคือ แบบ 9 ขา และแบบ 25 ขา บางครั้งเราจะเรียกว่า DB9 และ DB25 ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน โดยจะติดตั้งอยู่หลังเครื่องคอมพิวเตอร์ ระดับแรงดันจะมีค่าระหว่าง -3 โวลต์ถึง -15 โวลต์ สำหรับลอจิกสูงและลอจิกต่ำจะมีระดับแรงดันระหว่าง +3 โวลต์ถึง +15 โวลต์ สามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุด 50 ฟุต หรือ 150 เมตรแต่ถ้าเราต้องการสื่อสารกับอุปกรณ์อื่นที่อยู่ห่างกันมาก ๆ เราจำเป็นต้องใช้อุปกรณ์อื่น ๆ เข้าช่วย เช่นการใช้อุปกรณ์ที่ทำหน้าที่ในการแปลงสัญญาณอนาล็อกให้เปลี่ยนเป็นสัญญาณดิจิตอล (Modem) เป็นต้น



รูปที่ 2.21 ลักษณะของตัวเชื่อมต่อ

ตารางที่ 2.2 แสดงรายละเอียดของสายสัญญาณ

D-Type 25 Pin	D-Type 9 Pin	สัญลักษณ์	ชื่อสัญญาณ
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

รายละเอียดของสายสัญญาณประกอบไปด้วย

1) Transmit Data (TD) ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์

2) Receive Data (RD) ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์

3) Request To Send (RTS) ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้อุปกรณ์ปลายทางส่งข้อมูลกลับมา

4) Clear To Send (CTS) ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วย พร้อมที่จะรับข้อมูลหรือไม่ โดยจะคอยนับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทางขา TD

5) Data Set Ready (DSR) ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปการณ์ปลายทาง จะใช้คู่กับขา TD

6) Signal Ground (SG) เป็นกราวนด์ของระบบ

7) Carrier Detect (CD) ขานี้ตอบสนองเมื่อมีการส่งสัญญาณจากโมเด็ม

8) Data Terminal Ready (DTR) ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยโดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง

9) Ring Indicator (RI) ขานี้จะตอบสนองเมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

2.7.3.3 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมที่นิยมใช้กับคอมพิวเตอร์นั้น เป็นการสื่อสารข้อมูลแบบอะซิงโครนัส นั่นคือ ต้องใช้สายสัญญาณเส้นเดียวทำหน้าที่ทั้งส่วนที่เป็นข้อมูล และส่วนที่ใช้ควบคุมการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากการส่งแบบอนุกรม จึงต้องถูกแยกว่าใช้สำหรับวัตถุประสงค์ใด โดยสามารถแบ่งได้เป็น 4 ส่วนคือ

1) บิตเริ่มต้น (Start Bit) ขนาด 1 บิต จะใส่ที่จุดเริ่มต้นเสมอ เพื่อเตือนอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง

2) บิตข้อมูล (Data Character) ขนาด 7 หรือ 8 บิต ในการส่งบิตข้อมูลจะส่งเป็นกลุ่ม ๆ โดยทั่วไปจะส่งเป็น 7 หรือ 8 บิต

3) บิตพาริตี้ (Parity Bit) ขนาด 1 บิต ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง โดยจะใส่บิตพาริตี้เข้าไป แต่ทั้งตัวรับและตัวส่งจะต้องรู้กันว่าใช้พาริตี้แบบไหนในการส่งข้อมูล

4) บิตจบ (Stop Bit) ขนาด 1 หรือ 2 บิต เป็นบิตที่ส่งมาปิดท้ายข้อมูล

อัตราเร็วในการรับส่งข้อมูลแบบอนุกรม อุปกรณ์ในการรับ และส่งข้อมูลจะต้องทำงานด้วยอัตราเร็วเท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งค่าอัตราเร็วในการสื่อสารแบบอนุกรมสำหรับมาตรฐานการถ่ายโอนข้อมูล และคำสั่งผ่านพอร์ตอนุกรม นั้นมีใช้ดังนี้ 110 150 300 600 1200 2400 4800 9600 และ 19200 บิตต่อวินาที

2.7.3.4 การเขียนโปรแกรมติดต่อ และควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่านพอร์ตอนุกรมกับไมโครซอฟท์วิชวลเบสิก 6.0

การเขียนโปรแกรมติดต่อ และควบคุมการทำงานของอุปกรณ์ต่าง ๆ ไปยังบอร์ดไมโครตัวควบคุมเลอร์ด้วยโปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 การเขียนรหัสโปรแกรมจะถูกแบ่งออกเป็นส่วน ๆ เรียกว่า โพรซีเดอร์ (Procedure) แต่ละโพรซีเดอร์จะประกอบไปด้วยชุดคำสั่งที่พิมพ์เข้าไปแล้วทำให้ตัวควบคุมหรือเป้าหมายนั้น ๆ ตอบสนองการกระทำของผู้ใช้ ซึ่งเรียกว่าการเขียนโปรแกรมเชิงวัตถุ (Object Oriented Programming : OOP) แต่ตัวภาษาวิชวลเบสิกยังไม่ถือว่าเป็นการเขียนโปรแกรมเชิงวัตถุอย่างแท้จริง เนื่องจากข้อจำกัดหลาย ๆ อย่างที่ไมโครซอฟท์วิชวลเบสิก 6.0 ไม่สามารถทำได้ เนื่องจากไมโครซอฟท์วิชวลเบสิก 6.0 นั้นไม่ได้เน้นการติดต่อกับระบบฮาร์ดแวร์ทำให้มีตัวควบคุมที่เกี่ยวข้องกับการติดต่อ และควบคุมฮาร์ดแวร์มีอยู่ค่อนข้างน้อย สำหรับตัวควบคุมมาตรฐานตัวเดียวที่มีมาให้สำหรับการติดต่อกับฮาร์ดแวร์ในโปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 ก็คือ MSComm ซึ่งเน้นการติดต่อกับพอร์ตแบบอนุกรม

จุดเริ่มต้นของการพัฒนาโปรแกรมประยุกต์ด้วยไมโครซอฟท์วิชวลเบสิก 6.0 ก็คือการนำตัวควบคุมชนิดต่าง ๆ ที่ VB 6.0 จัดเตรียมไว้นำมาสร้างอินเตอร์เฟส การพัฒนาโปรแกรมประยุกต์ที่ดี ทำได้โดยการออกแบบการเชื่อมต่อที่ใช้งานง่าย เป็นมิตรกับผู้ใช้ จะส่งผลให้ระยะเวลาในการพัฒนาโปรแกรมประยุกต์ลดลงไปได้มาก เพราะสิ่งที่เหลืออยู่คือการเขียนรหัสโปรแกรมเพื่อทำให้โครงการนั้นทำงานให้สมบูรณ์มากที่สุด

หลักของการพัฒนาโปรแกรมประยุกต์ด้วยโปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 ก็คือ การสร้างองค์ประกอบต่าง ๆ ของโปรแกรมประยุกต์ด้วยตัวควบคุม โดยมีรูปแบบที่สื่อด้วยภาพ หรือที่เรียกว่า การออกแบบการเชื่อมต่อ ต่อมาก็คือการเขียนชุดคำสั่งเพื่อรองรับเหตุการณ์ต่าง ๆ ที่อาจเกิดขึ้นได้ ซึ่งจะแบ่งออกเป็น 5 ขั้นตอนใหญ่ ๆ คือ

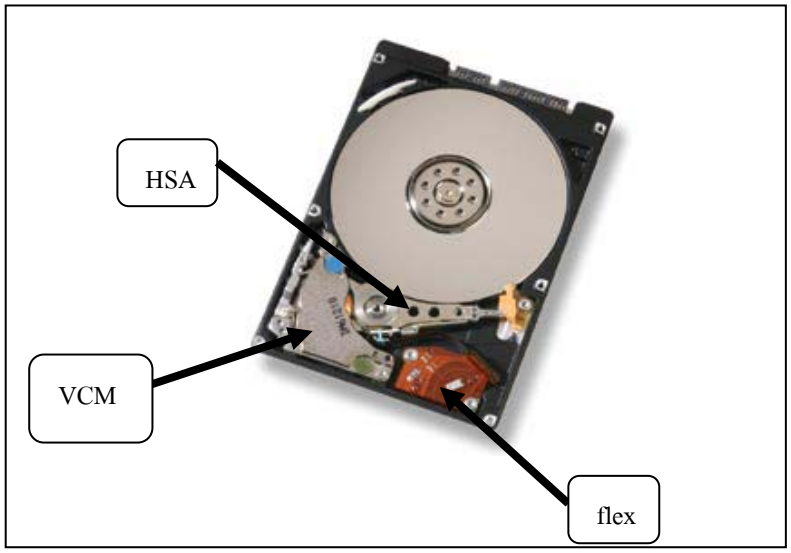
1) เลือกชนิดของโปรแกรมประยุกต์
2) สร้างส่วนติดต่อกับผู้ใช้
3) เขียนชุดคำสั่งเพื่อรองรับเหตุการณ์ ที่อาจเกิดขึ้นกับแต่ละตัวควบคุม
4) การทดสอบ ตรวจสอบ และดักจับข้อผิดพลาด
5) คอมไพล์โปรเจ็กต์ให้เป็นโปรแกรมประยุกต์ที่สมบูรณ์ (เช่น *.exe หรือ *.dll เป็นต้น)

บทที่ 3
การดำเนินงานวิจัย

เพื่อทำการออกแบบเครื่องต้นแบบในกระบวนการถอด HSA/VCM โดยอัตโนมัติ ให้สามารถลดเวลา ความเสียหายและลดสารปนเปื้อนจากกระบวนการถอดโดยแรงงานคนในปัจจุบัน ทั้งนี้จึงต้องทำการศึกษากระบวนการถอด HSA/VCM ในสายการผลิตที่คนทำงานอยู่ในปัจจุบันของฮาร์ดดิสก์ไดรฟ์ขนาด 2.5 นิ้ว เพื่อเป็นแนวทางในการออกแบบ จากนั้นดำเนินการออกแบบเครื่องต้นแบบด้วยโปรแกรม SolidWorks 2008 เมื่อได้รูปแบบของเครื่องต้นแบบแล้วจึงดำเนินการสร้างเครื่องต้นแบบ และออกแบบระบบควบคุมการทำงานอัตโนมัติให้กับเครื่องต้นแบบ ในส่วนของกระบวนการขันสกรูออก ดูดสกรู และสารปนเปื้อนที่เกิดจากกระบวนการขันสกรู และกด flex ด้วยไมโครคอนโทรลเลอร์ ดังนี้

3.1 กระบวนการถอด HSA/VCM โดยมนุษย์

ในกระบวนการถอด HSA/VCM นั้นอุปกรณ์ที่จะต้องทำการถอดประกอบหลังจากดำเนินการขันสกรูที่ตำแหน่ง HSA ยึดกับฐานออก มีดังนี้

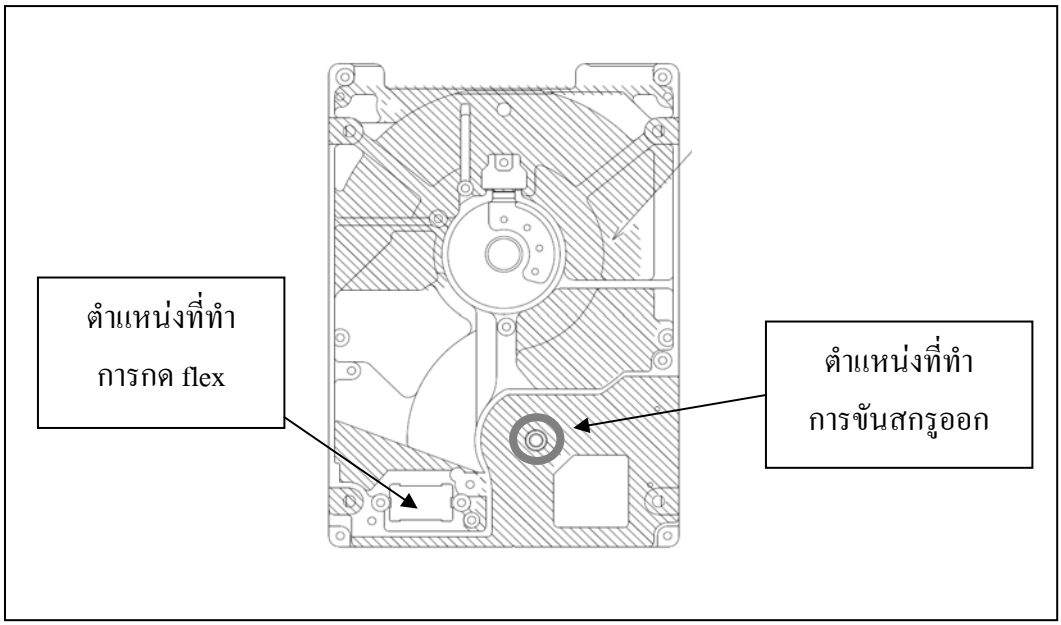


รูปที่ 3.1 อุปกรณ์ที่ดำเนินการถอดหลังจากกระบวนการขันสกรูออกที่ตำแหน่ง HSA ยึดกับฐาน

กระบวนการถอด HSA/VCM โดยแรงงานคนของฮาร์ดดิสก์ไดรฟ์ขนาด 2.5 นิ้ว ของบริษัทผู้ผลิต ขั้นตอนของกระบวนการถอด HSA/VCM จะทำการขันสกรูออก 1 ตำแหน่ง ซึ่งเป็นตำแหน่งที่ยึด HSA กับฐานไว้ โดยมีชุดจับยึดชิ้นงานรองรับฮาร์ดดิสก์ไดรฟ์เพื่อช่วยในการถอด HSA/VCM ดังรูปที่ 3.2โดยมีขั้นตอนดังนี้

พนักงานวางฮาร์ดดิสก์ไดรฟ์ลงบนชุดจับยึดชิ้นงาน โดยคว่ำด้านหน้าฮาร์ดดิสก์ไดรฟ์ลง ลักษณะการวางดังรูปที่ 3.3 และพนักงานหมุนชุดกลไกเพื่อทำให้ฮาร์ดดิสก์ไดรฟ์ลงมาที่ตำแหน่งปฏิบัติงาน ณ บริเวณนี้ จะมีชุดรองรับ HSA และ VCM โดยพนักงานใช้อุปกรณ์กด flex ทำการกด flex ให้หลุดออกจากฐานของฮาร์ดดิสก์ไดรฟ์ และใช้ไขควงทำการขันสกรูที่ยึด HSA กับฐานออก เมื่อขันสกรูออกเรียบร้อยแล้ว พนักงานใช้ท่อสุญญากาศดูดตัวสกรู และอนุภาคต่าง ๆ ที่เกิดจากกระบวนการขันสกรูออก จากนั้นนำอุปกรณ์กด flex ออก และหมุนชุดกลไกขึ้นเพื่อยกฮาร์ดดิสก์ไดรฟ์ขึ้น หยิบ VCM HSA และ flex ที่วางอยู่บนอุปกรณ์รองรับออกไว้ที่ถาด สำหรับ HSA ต้องทำการใส่ head clip เพื่อป้องกันไม่ให้แขนหัวอ่านติดกัน

รูปที่ 3.2 ชุดจับยึดชิ้นงานของกระบวนการถอด HSA/VCM โดยมนุษย์



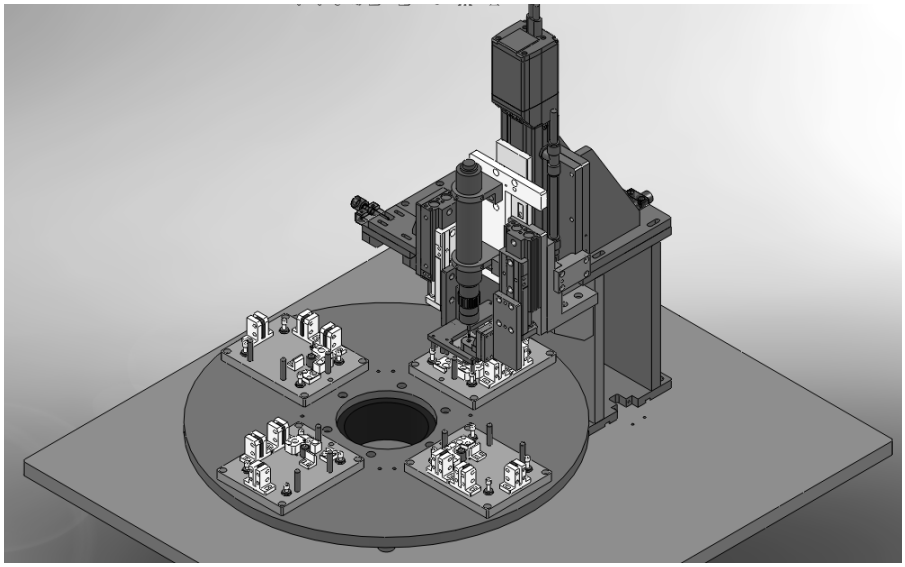
รูปที่ 3.3 ตำแหน่งที่ทำการขันสกรูออก และกด flex

3.2 การออกแบบเครื่องจักรและระบบกลไก

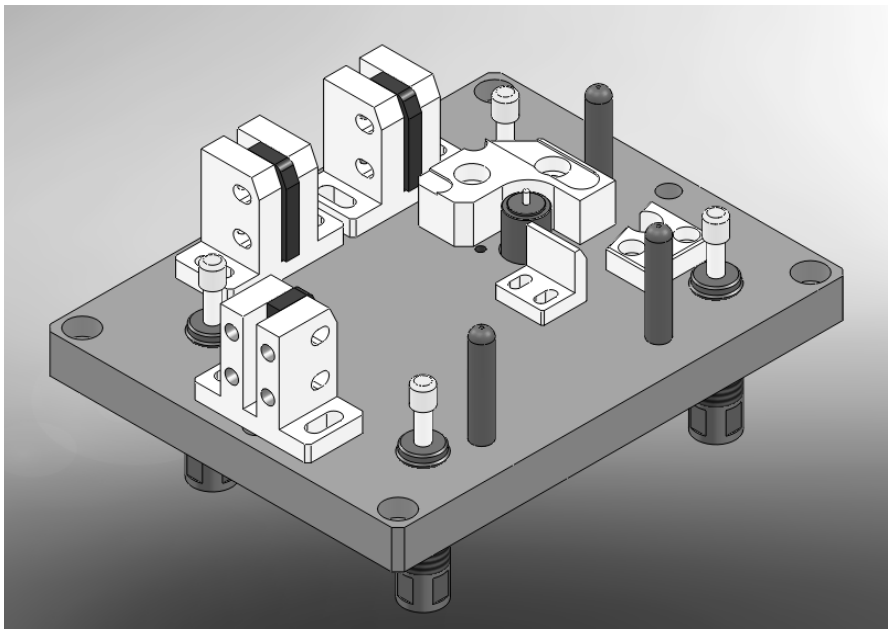
3.2.1 การออกแบบเครื่องจักรด้วยโปรแกรม SolidWorks

การออกแบบเครื่องต้นแบบจะทำการออกแบบโดยใช้โปรแกรม SolidWorks รุ่น 2008 โดยแนวคิดในการออกแบบเครื่องที่ใช้ในการถอด HSA/VCM ของฮาร์ดดิสก์ไดรฟ์ขนาด 2.5 นิ้ว จะออกแบบเครื่องต้นแบบให้สามารถวางชิ้นงานได้ 4 ชิ้น และจะใช้ลักษณะการทำงานเป็นแบบหมุนชิ้นงาน โดยในการหมุนชิ้นงานแต่ละครั้งจะทำการหมุนครั้งละ 90 องศา ลักษณะของเครื่องต้นแบบที่ทำการออกแบบแสดงดังรูปที่ 3.4 โดยการออกแบบโครงสร้างในส่วนต่าง ๆ ของเครื่องจักรต้นแบบจะคำนึงถึงขนาดสัดส่วนที่เหมาะสมในการใช้งาน พื้นที่การใช้งาน และความแข็งแรงในการรองรับอุปกรณ์

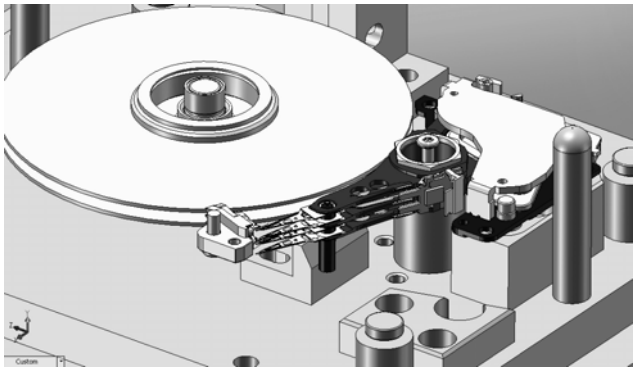
ในส่วนของการออกแบบชุดจับยึดชิ้นงานจะถูกออกแบบให้รองรับฮาร์ดดิสก์ไดรฟ์ และ HSA/VCM โดยจะอ้างอิงจากชุดจับยึดชิ้นงานที่ใช้ในกระบวนการถอด HSA/VCM โดยแรงงานคนในกระบวนการปัจจุบัน ซึ่งชุดจับยึดชิ้นงานจะแตกต่างกันในส่วนของอุปกรณ์ที่ทำการรองรับฮาร์ดดิสก์ไดรฟ์ โดยจะเปลี่ยนมาใช้อุปกรณ์กันสะท้อน (Shock absorber) แทนการใช้แรงงานคนในการหมุนชุดกลไก อุปกรณ์นี้ใส่เพื่อลดการเสียหายของ HSA/VCM ที่เกิดจากแรงกระแทกโดยตรงจากการกดฮาร์ดดิสก์ไดรฟ์ ดังรูปที่ 3.5 โดยฮาร์ดดิสก์ไดรฟ์จะถูกกดลงมาได้เนื่องมาจากแรงที่ใช้ในการกดยึดชิ้นงานจากชุดกระบอกสูบลม และจากรูปชุดจับยึดชิ้นงานจะมีตัวที่ใช้ช่วยในการหลุดของ HSA ออกจากแผ่นข้อมูล เพื่อไม่ให้แผ่นข้อมูลเกิดความเสียหาย ในขณะทำการขันสกรูออกในตำแหน่งที่ HSA ยึดกับฐาน ลักษณะดังรูปที่ 3.6



รูปที่ 3.4 ลักษณะของเครื่องต้นแบบที่ทำการออกแบบด้วยโปรแกรม SolidWorks

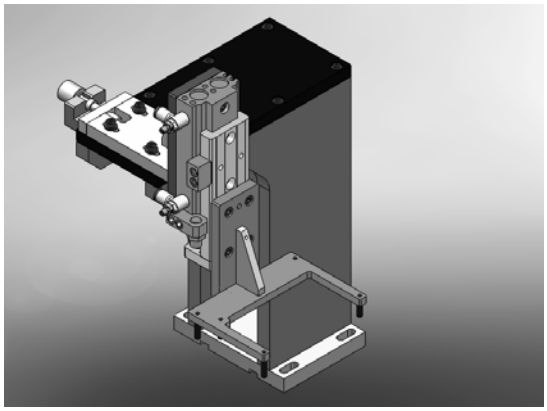


รูปที่ 3.5 ชุดจับยึดชิ้นงานที่ออกแบบ

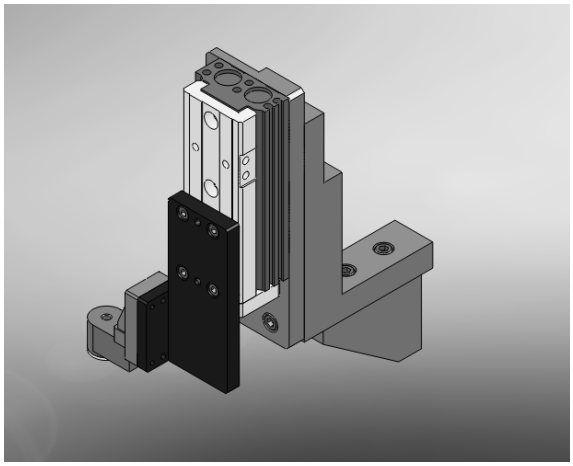


รูปที่ 3.6 การหลุดของ HSA ออกจากแผ่นข้อมูล

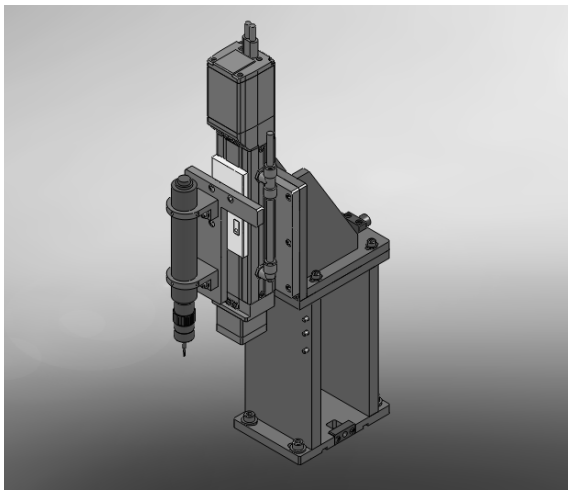
กระบวนการทำงานในส่วนของการขันสกรู จะเลือกใช้กระบอกสูบที่มีความแม่นยำในการกำหนดตำแหน่งการเคลื่อนที่ในแนวแกน y จำนวน 1 ชุด เพื่อจับยึดไขควง ส่วนขั้นตอนการกดฮาร์ดดิสก์ไดรฟ์ และ flex และขั้นตอนการดูดสกรูและสารปนเปื้อนในกระบวนการ จะเลือกใช้กระบอกสูบที่ไม่ต้องการความแม่นยำสูงในการกำหนดตำแหน่งการเคลื่อนที่ในแนว y ของแต่ละขั้นตอนอย่างละ 1 ชุด ในการออกแบบชุดที่ใช้ยึดอุปกรณ์การเคลื่อนที่ในแนวเส้นตรงของเครื่องต้นแบบในแต่ละส่วนจะทำการออกแบบให้มีความเหมาะสมกับพื้นที่ใช้งาน การรองรับน้ำหนัก ความสะดวกในการถอดประกอบ และในการติดตั้งให้สามารถปรับตำแหน่งได้ง่าย สำหรับชุดยึดของกระบอกสูบทั้ง 3 ชุด จะออกแบบให้ติดตั้งแยกกัน เพื่อความสะดวกในการติดตั้ง และ การสั่งงานควบคุมชุดอุปกรณ์ เนื่องจากในการสั่งงานของอุปกรณ์แต่ละชุดจะใช้คำสั่งใน การควบคุมแยกจากกัน



รูปที่ 3.7 ชุดขาตั้งยึดกระบอกสูบลมของการกดฮาร์ดดิสก์ไดรฟ์ และ flex



รูปที่ 3.8 ชุดขาตั้งยึดกระบอกสูบลมของชุดท่อสุญญากาศ



รูปที่ 3.9 ชุดขาตั้งยึดกระบอกสูบไฟฟ้าของชุดไขควง

การขึ้นรูปชิ้นงานของเครื่องต้นแบบ สำหรับวัสดุที่เลือกใช้ของชิ้นส่วนต่าง ๆ ขึ้นอยู่กับการนำไปใช้งานในภาคอุตสาหกรรม ซึ่งสำหรับเครื่องต้นแบบนี้เป็นเครื่องจักรที่นำไปใช้งานในอุตสาหกรรมฮาร์ดดิสก์ไดรฟ์ ดังนั้นวัสดุที่เลือกใช้ควรมีคุณสมบัติคือ ใช้โลหะจำพวกเหล็กชนิดที่ไม่เกิดสนิม เพื่อป้องกันสิ่งปนเปื้อนที่จะเกิดขึ้น ส่วนวัสดุที่เป็นพลาสติกจะต้องป้องกันการเกิดสิ่งปนเปื้อน และไฟฟ้าสถิต เนื่องจากวัสดุที่ใช้จะต้องคำนึงถึงการใช้งานในห้องสะอาด การเกิดสิ่งปนเปื้อน และไฟฟ้าสถิต ที่มีผลต่อความเสียหายของชิ้นส่วนของฮาร์ดดิสก์ไดรฟ์ สำหรับเครื่องต้นแบบวัสดุที่เลือกใช้ในส่วนที่เป็นโลหะ คือ SUS304 และ SUS440C เป็นโลหะจำพวกเหล็ก ชนิดเหล็กกล้าไร้สนิม หรือที่เรียกว่าสแตนเลส (Stainless steel) โดย SUS304 เป็นวัสดุที่ง่าย

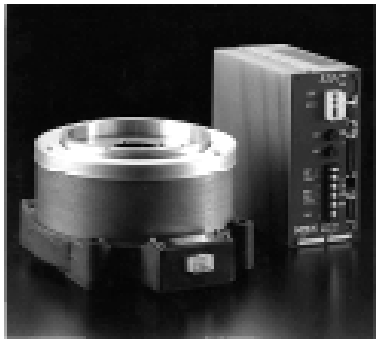
ต่อการขึ้นรูป และป้องกันการเกิดสนิมได้เป็นอย่างดี ส่วน SUS440C จะดีกว่า SUS304 ตรงที่สามารถชุบแข็งได้ดี จึงทนต่อการเสียดสีสูง และทนต่อการกัดกร่อนได้ดีกว่า และวัสดุอีกชนิดที่เป็นพลาสติกคือ ULTEM1000 สำหรับ ULTEM1000 เป็นวัสดุประเภทพลาสติก หรือไฮเทคพอลิเมอร์ คือพอลิเมอร์ที่ได้รับการพัฒนาขึ้นมาเพื่อใช้ในเทคโนโลยีชั้นสูง มีคุณสมบัติคือ ใส ทนความร้อนได้สูง ทนต่อสารเคมี และไม่นำไฟฟ้า สำหรับเครื่องต้นแบบจะใช้ในส่วนของชุดจับยึดชิ้นงาน ชุดกดฮาร์ดดิสก์ไดรฟ์ และ flex และชุดท่อสุญญากาศ ในส่วนของตำแหน่งของชุดอุปกรณ์ที่ต้องมีการกระทบกับฮาร์ดดิสก์ไดรฟ์ในกระบวนการทำงาน ที่อาจทำให้ชิ้นส่วนภายในของฮาร์ดดิสก์ไดรฟ์เกิดความเสียหายได้

3.2.2 ชุดส่วนประกอบที่ใช้ในเครื่องต้นแบบ

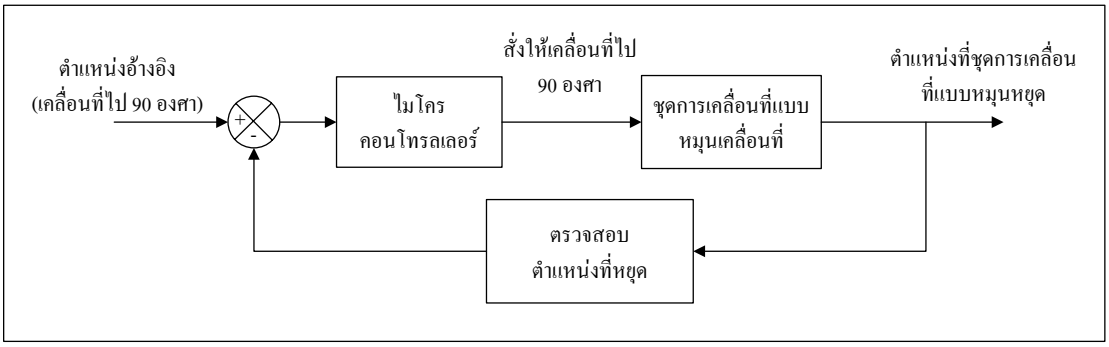
อุปกรณ์ที่เลือกใช้เป็นส่วนประกอบของเครื่องต้นแบบ มีรายละเอียดดังนี้

3.2.2.1 ชุดการเคลื่อนที่แบบหมุน

อุปกรณ์ที่เลือกใช้ในการหมุนชุดจับยึดชิ้นงานไปยังตำแหน่งทำงานของเครื่องต้นแบบ ผู้วิจัยเลือกใช้อุปกรณ์ที่มีการใช้งานอยู่แล้วในบริษัทผู้ผลิต คือ ชุดการเคลื่อนที่แบบหมุนของบริษัท NSK รุ่น YS3008FN001 รองรับค่าแรงบิดได้มากสุดเท่ากับ 8 นิวตันเมตร และรองรับน้ำหนักได้ไม่เกิน 450 กิโลกรัม ดังรูปที่ 3.7 ซึ่งจากการออกแบบอุปกรณ์ที่วางบนชุดการเคลื่อนที่แบบหมุน น้ำหนักโดยประมาณจากการเลือกใช้วัสดุสำหรับชิ้นส่วนต่าง ๆ เมื่อคำนวณจากโปรแกรม SolidWorks 2008 มีค่าประมาณ 30 กิโลกรัม จะเห็นว่าอุปกรณ์ชุดการเคลื่อนที่แบบหมุนชุดนี้สามารถรองรับน้ำหนักได้ ซึ่งหากไม่ใช้อุปกรณ์ตัวดังกล่าวนี้ก็สามารถใช้อุปกรณ์อื่นที่สามารถตอบสนองต่อลักษณะการทำงาน และรองรับภาระกรรมดังกล่าวได้ โดยแสดงแผนภาพระบบของการทำงานของชุดการเคลื่อนที่แบบหมุนของเครื่องต้นแบบได้ดังรูปที่ 3.11



รูปที่ 3.10 ชุดการเคลื่อนที่แบบหมุนของบริษัท NSK รุ่น YS3008FN001

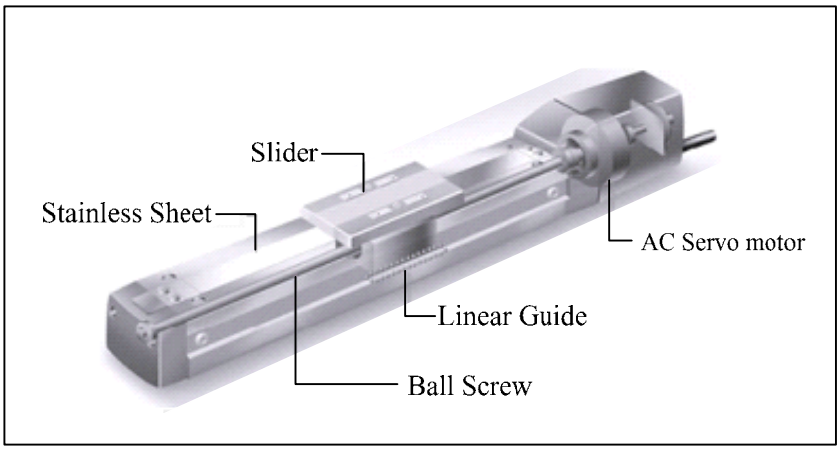


รูปที่ 3.11 แผนภาพระบบของชุดการเคลื่อนที่แบบหมุน

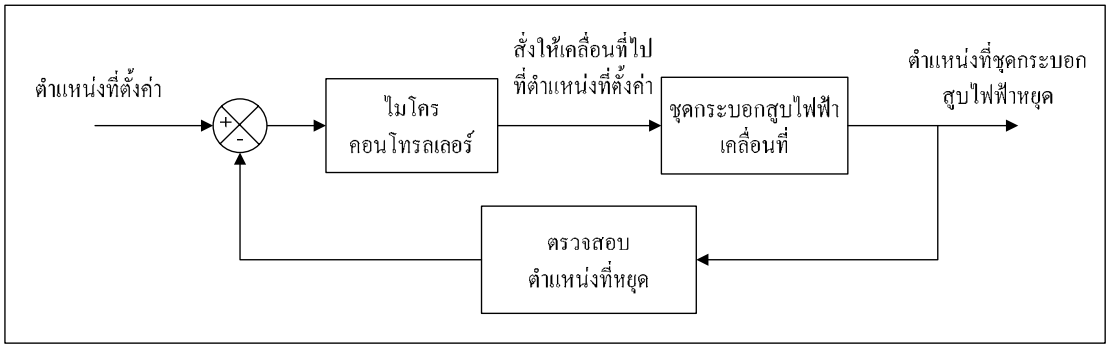
3.2.2.2 ชุดการเคลื่อนที่ในแนวเส้นตรง

ชุดอุปกรณ์การเคลื่อนที่ในแนวเส้นตรงสำหรับเครื่องต้นแบบที่ใช้ในงานวิจัยนี้ เลือกใช้จำนวน 3 ชุด คือ

ชุดที่หนึ่ง สำหรับการเคลื่อนที่ของชุดไขควง ซึ่งการเลือกใช้กระบอกสูบในการเคลื่อนที่ของชุดไขควงนี้จะต้องเป็นกระบอกสูบที่ให้ความแม่นยำและเที่ยงตรงสูง เนื่องจากจะต้องกำหนดตำแหน่งในการเคลื่อนที่ของการขันสกรูที่ชัดเจน เพราะถ้ากำหนดตำแหน่งได้ไม่เที่ยงตรงและไม่แม่นยำ จะทำให้การขันสกรูผิดพลาดและทำให้เกิดการเสียหายต่ออุปกรณ์ของฮาร์ดดิสก์ไดรฟ์ได้ จากการออกแบบกระบอกสูบนี้ต้องรับน้ำหนักในส่วนของไขควง และชุดอุปกรณ์ยึดไขควง ซึ่งประมาณน้ำหนักได้ 2 กิโลกรัม และระยะที่เคลื่อนที่ในการปฏิบัติงานคือประมาณ 50 มิลลิเมตร โดยผู้วิจัยเลือกใช้อุปกรณ์ที่มีการใช้งานอยู่แล้วในบริษัทผู้ผลิต คืออุปกรณ์ของบริษัท IAI รุ่น RCP2CR-SA6-I-PM-2-100-P1-S-BL-JP ดังแสดงในรูปที่ 3.8 มีคุณสมบัติคือเป็นกระบอกสูบไฟฟ้า สามารถรับน้ำหนักในแนวดิ่งได้ 6 กิโลกรัม รับน้ำหนักในแนวระนาบได้ 12 กิโลกรัม ปรับค่าความเร็วได้ตั้งแต่ 1-150 มิลลิเมตรต่อวินาที และมีช่วงของกระบอกสูบ 100 มิลลิเมตร อุปกรณ์รุ่นนี้จะเหมาะสำหรับการใช้งานในห้องสะอาด เนื่องจากการทำงานของบริษัทผู้ผลิตมีการควบคุมเรื่องฝุ่นผง ซึ่งหากไม่ใช้อุปกรณ์รุ่นดังกล่าว ก็สามารถใช้อุปกรณ์อื่นที่สามารถตอบสนองต่อลักษณะการทำงานและรองรับภาระกรรมดังที่กล่าวมาได้ โดยแสดงแผนภาพระบบของการทำงานของชุดกระบอกสูบไฟฟ้าของเครื่องต้นแบบได้ดังรูปที่ 3.13



รูปที่ 3.12 ชุดกระบอกสูบไฟฟ้าของบริษัท IAI รุ่น RCP2CR-SA6-I-PM-2-100-P1-S-BL-JP



รูปที่ 3.13 แผนภาพระบบของชุดกระบอกสูบไฟฟ้า

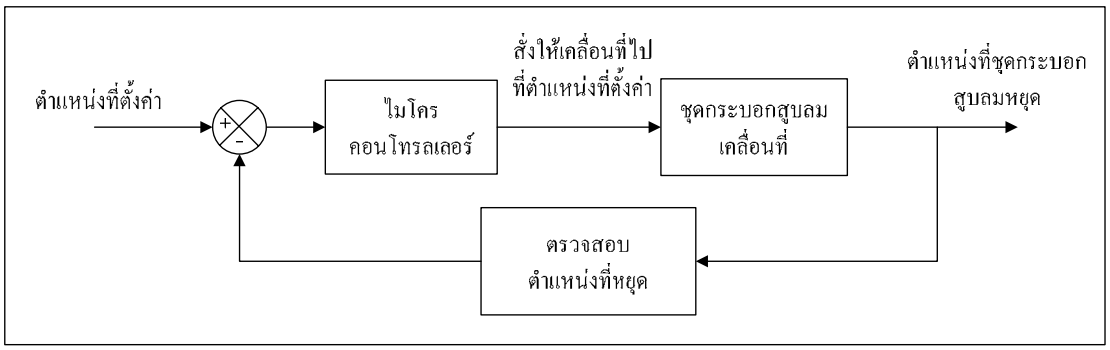
ชุดที่สอง สำหรับการเคลื่อนที่ของชุดท่อสุญญากาศ กระบอกสูบที่เลือกใช้กับการเคลื่อนที่ของชุดท่อสุญญากาศ ไม่จำเป็นต้องใช้กระบอกสูบที่มีความแม่นยำสูงมาก ต้องการเพียงเลือกช่วงการทำงานของกระบอกสูบให้เหมาะสมกับการใช้งาน ลักษณะการเคลื่อนที่ และ การรับน้ำหนัก ซึ่งระยะการทำงานที่ใช้ในการเคลื่อนที่จากการออกแบบมีค่าประมาณ 30 มิลลิเมตร รับน้ำหนักชุดท่อสุญญากาศประมาณ 150 กรัม โดยผู้วิจัยได้เลือกใช้อุปกรณ์ที่มีอยู่แล้วในบริษัทผู้ผลิต คืออุปกรณ์ของบริษัท SMC รุ่น MXQ12-50CS ดังรูปที่ 3.9 ซึ่งมีคุณสมบัติคือ เป็นกระบอกสูบลม ใช้ลมในการเคลื่อนที่ ระยะของกระบอกสูบ 0-50 มิลลิเมตร รับน้ำหนัก 190 กรัม ให้แรงดัน 0.15 ถึง 0.7 MPa และให้แรงกดขนาด 45-158 นิวตัน

ชุดที่สาม สำหรับการเคลื่อนที่ของชุดกดฮาร์ดดิสก์ไดรฟ์ และกด flex โดยกระบอกสูบที่ใช้ในส่วนนี้ต้องรองรับแรงดันกลับจากอุปกรณ์กันสะท้อน ที่ช่วยในการดูดซับแรงกดของกระบอกสูบ จากการคำนวณแรงต้านของอุปกรณ์กันสะท้อน 4 ตัว แรงที่ดันกลับของทั้ง

4 ตัวประมาณ 24 นิวตัน ผู้วิจัยจึงได้เลือกใช้ที่มีอยู่แล้วในบริษัทผู้ผลิต เป็นอุปกรณ์ของบริษัท SMC รุ่นเดียวกับที่ใช้สำหรับการเคลื่อนที่ของชุดท่อสุญญากาศ

กระบอกสูบสองชุดหลังนี้เป็นแบบกระบอกสูบชนิด Air slide table โดยเป็นกระบอกสูบที่เคลื่อนที่ในแนวเส้นตรง ที่ใช้ลมเคลื่อนที่โดยไฟฟ้าสั่งงานลิ้นโซลินอยด์ (Solenoid valve) เนื่องจากการทำงานในส่วนของชุดกดฮาร์ดดิสก์ไดรฟ์ และกด flex และ ชุดท่อสุญญากาศไม่เน้นในเรื่องความแม่นยำมาก ทั้งนี้ถ้าหากไม่ใช้กระบอกสูบลมรุ่นดังที่กล่าวมาก็สามารถเลือกใช้รุ่นอื่นที่มีลักษณะการทำงานและรองรับภาระกรรมตามที่กำหนดได้ โดยแสดงแผนภาพระบบของการทำงานของชุดกระบอกสูบลมของเครื่องต้นแบบได้ดังรูปที่ 3.15

รูปที่ 3.14 ชุดกระบอกสูบลมของบริษัท SMC รุ่น MXQ



รูปที่ 3.15 แผนภาพระบบของชุดกระบอกสูบลม

3.2.2.3 อุปกรณ์ปรับแรงดันลมและกรองอากาศ

สำหรับการปรับลดแรงดันลมที่มาจากต้นทางก่อนจะเข้าสู่กระบอกสูบลม เพื่อให้แรงดันลมที่เข้ามามีค่าไม่มากจนเกินไป จะใช้อุปกรณ์ปรับแรงดันลม ที่เรียกว่า Filter regulator สำหรับในเครื่องต้นแบบเลือกใช้ของบริษัท SMC รุ่น AW20-02BG ดังรูปที่ 3.8 มีคุณสมบัติ คือปรับค่าแรงดันได้ในช่วง 0.05-0.85 MPa

รูปที่ 3.16 Filter regulator ของบริษัท SMC รุ่น AW20-02BG

3.2.2.4 อุปกรณ์ปรับความเร็วลม

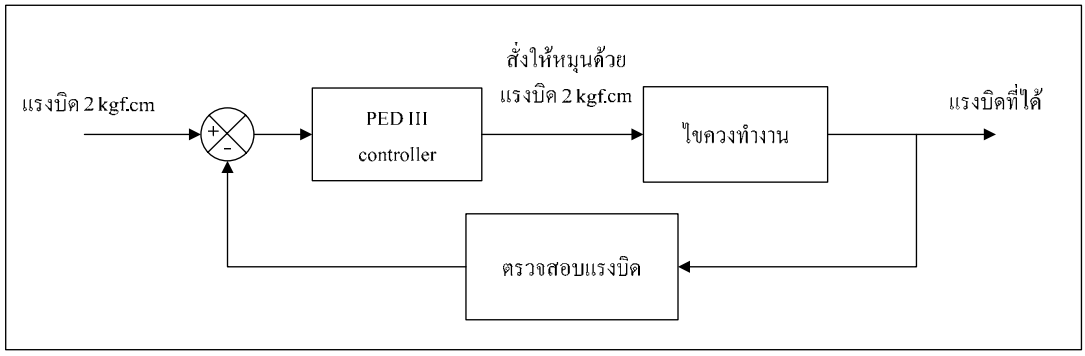
สำหรับการปรับแรงดันเพื่อควบคุมความเร็วในการเคลื่อนที่ของชุดกระบอกสูบลมให้สามารถเคลื่อนที่ได้ช้าหรือเร็วจะใช้อุปกรณ์ที่เรียกว่า Speed control fitting โดยเลือกของบริษัท SMC รุ่น AS1201F-M5-04S ดังรูปที่ 3.9

รูปที่ 3.17 Speed control fitting ชนิด Elbow ของบริษัท SMC

3.2.2.5 ชุดอุปกรณ์ขันสกรู

อุปกรณ์ที่ใช้ในการขันสกรูจะใช้ไขควง (Screwdriver) ของบริษัท Technart รุ่น A ซึ่งไขควงที่ใช้เป็นไขควงไฟฟ้า ทำงานโดยการสั่งงานจากกล่องควบคุม รุ่นของกล่องควบคุมคือ PED III (PEDIII-h) ปรับค่าแรงบิดได้ตั้งแต่ 0.3–8.5 กิโลกรัมแรงเซนติเมตร ในการสั่งงานสำหรับการขันสกรูเข้าเมื่อถึงจุดปฏิบัติงาน ช่วงแรกจะทำการให้แรงบิดต่ำ ๆ ที่ค่า 1 กิโลกรัมแรงเซนติเมตร (0.10 นิวตันเมตร) ก่อน เพื่อให้ดอกสกรูลงในแฉกของหัวสกรู แล้วจึงให้ค่าแรงบิดประมาณ 2-4 กิโลกรัมแรงเซนติเมตร (0.25-0.40 นิวตันเมตร) ในการขันสกรูเข้า สำหรับเครื่องต้นแบบการทำงานของไขควงจะเป็นในส่วนของการขันสกรูออก จากการศึกษากระบวนการถอด HSA/VCM โดยแรงงานคน ค่าแรงบิดที่เหมาะสำหรับการคลายสกรูมีค่าประมาณ 1.5-2.1 กิโลกรัมแรงเซนติเมตร คิดมาจากค่าแรงบิดที่ใช้ขันสกรูเข้าอยู่ที่ 3 กิโลกรัมแรงเซนติเมตร ซึ่งค่าแรงบิดที่ใช้ขันออกจะอยู่ที่ประมาณร้อยละ 50-70 ของการขันสกรูเข้า สาเหตุที่ทำการเลือกอุปกรณ์นี้ เนื่องจากบริษัทผู้ผลิตใช้ในกระบวนการผลิตฮาร์ดดิสก์ไดรฟ์ โดยแสดงแผนภาพระบบของการทำงานของชุดอุปกรณ์ขันสกรูของเครื่องต้นแบบได้ดังรูปที่ 3.19

รูปที่ 3.18 ชุดอุปกรณ์ขันสกรูของบริษัท Technart



รูปที่ 3.19 แผนภาพระบบของชุดอุปกรณ์ขันสกรู

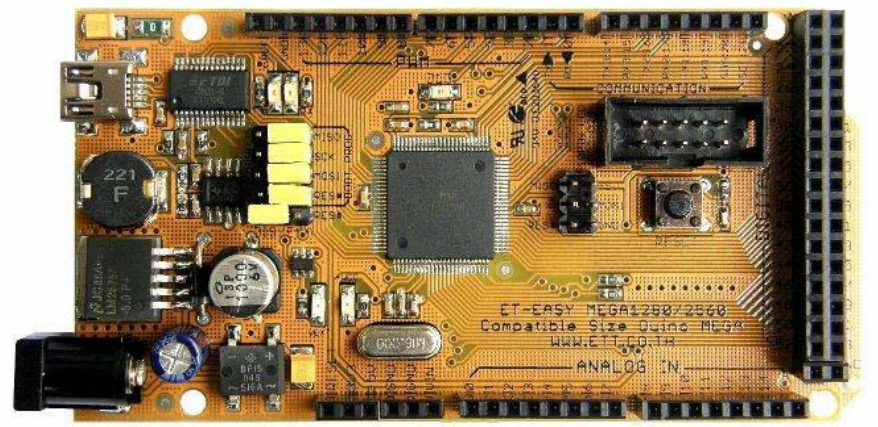
ผู้จัดทำงานวิจัยได้ทำการเลือกอุปกรณ์ต่าง ๆ ที่ใช้ในเครื่องต้นแบบในกระบวนการถอด HSA/VCM โดยขึ้นอยู่กับองค์ประกอบต่าง ๆ ดังนี้

1. หน้าที่และลักษณะในการทำงาน
2. ความแม่นยำและเที่ยงตรงบริเวณตำแหน่งที่ทำงาน
3. ภาระงานที่อุปกรณ์จะได้รับ
4. อุปกรณ์ที่มีอยู่ในบริษัทผู้ผลิต
5. ราคาที่เหมาะสมกับคุณสมบัติของอุปกรณ์

ทั้งนี้การเลือกใช้อุปกรณ์สามารถเลือกใช้อุปกรณ์อื่นที่มีคุณสมบัติ ลักษณะการทำงานและรับภาระการทำงานได้ตามที่ออกแบบแทนกันได้ เนื่องจากอุปกรณ์รุ่นอื่นอาจมีประสิทธิภาพดีกว่า

3.3 การควบคุมระบบการทำงานของเครื่องจักร

ในการควบคุมการทำงานของอุปกรณ์ต่าง ๆ ของเครื่องต้นแบบจะควบคุมด้วยไมโครคอนโทรลเลอร์ เนื่องจากเป็นการสร้างเครื่องต้นแบบ เพื่อต้องการดูผลของการทำงานซึ่งลักษณะการใช้งาน และขั้นตอนทำงานของเครื่องต้นแบบที่ทำการออกแบบมีขั้นตอนในการทำงานไม่ซับซ้อนมาก และ ปัจจุบันนี้ระบบควบคุมแบบไมโครคอนโทรลเลอร์ มีการพัฒนาในส่วนต่าง ๆ มากขึ้นเช่น การใช้งาน ความสามารถในการรองรับการเก็บข้อมูล การประมวลผล การเขียนโปรแกรมควบคุม และมีราคาที่ไม่แพง เป็นต้น โดยไมโครคอนโทรลเลอร์ที่เลือกใช้คือ รุ่น ATMEGA 1280 ตระกูล AVR เป็นบอร์ดของบริษัทอีทีที รุ่น ET-EASY MEGA1280 (DUINO MEGA) ดังรูปที่ 3.11



รูปที่ 3.20 บอร์ด ET-EASY MEGA1280 (DUINO MEGA)
ใช้ชิพรุ่น ATMEGA 1280

บอร์ดรุ่น ET-EASY MEGA1280 (DUINO MEGA) ของบริษัทอีทีที ได้เลือกใช้ชิพไมโครคอนโทรลเลอร์ ATmega1280 ที่เป็นชิพตระกูล AVR ของบริษัท Atmel รองรับการเขียนโปรแกรมภาษาซีของ Arduino ที่เป็นหน่วยประมวลผลหลัก โดยชิพรุ่นนี้มีหน่วยความจำแฟลชสำหรับเก็บเขียนโปรแกรม 128 กิโลไบต์ มีหน่วยความจำแรม 8 กิโลไบต์ มี EEPROM อีก 4 กิโลไบต์สาหรับใช้เป็นที่เก็บข้อมูลถาวรได้เมื่อไม่มีไฟเลี้ยง และยังมีขานำเข้า/ส่งออก (I/O) สำหรับต่อใช้งานทั่วไป 86 ขา มีคำสั่งควบคุมอัตราการหมุนของมอเตอร์ (Pulse Width Modulation : PWM) ที่กำหนดความละเอียดได้ระดับ 16 บิตให้ใช้งานถึง 12 ช่องสัญญาณ มีช่องสื่อสารแบบอนุกรม 4 พอร์ต และสามารถแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog-to-Digital Converters : ADC) อีก 16 ช่องสัญญาณ ซึ่งจะเห็นว่ามีความสามารถพื้นฐานที่มากพอสำหรับงานควบคุมที่หลากหลาย

3.3.1 การพัฒนาโปรแกรมของบอร์ด ET-EASY MEGA1280 (DUINO MEGA)

สามารถทำได้ใน 2 รูปแบบการพัฒนา คือ

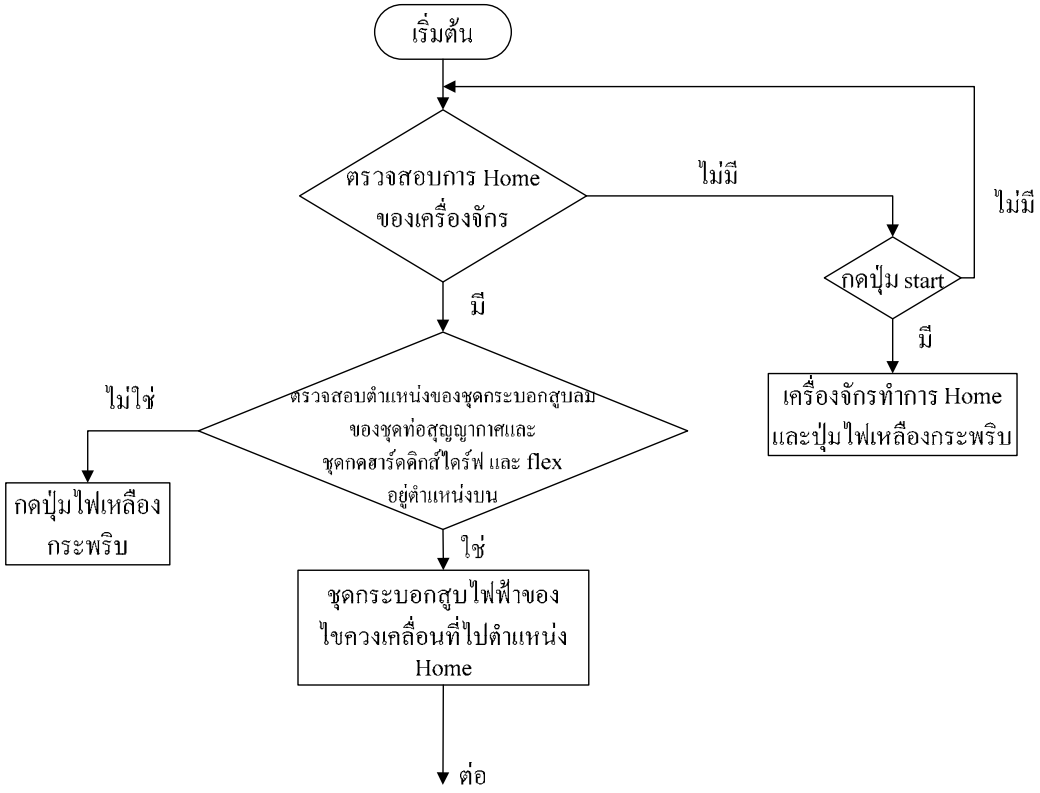
1. ARDUINO PROJECT เป็นรูปแบบการพัฒนาโปรแกรมโดยใช้โปรแกรมและชุดคำสั่งด้วย ภาษาซี (C++) ของ ARDUINO PROJECT

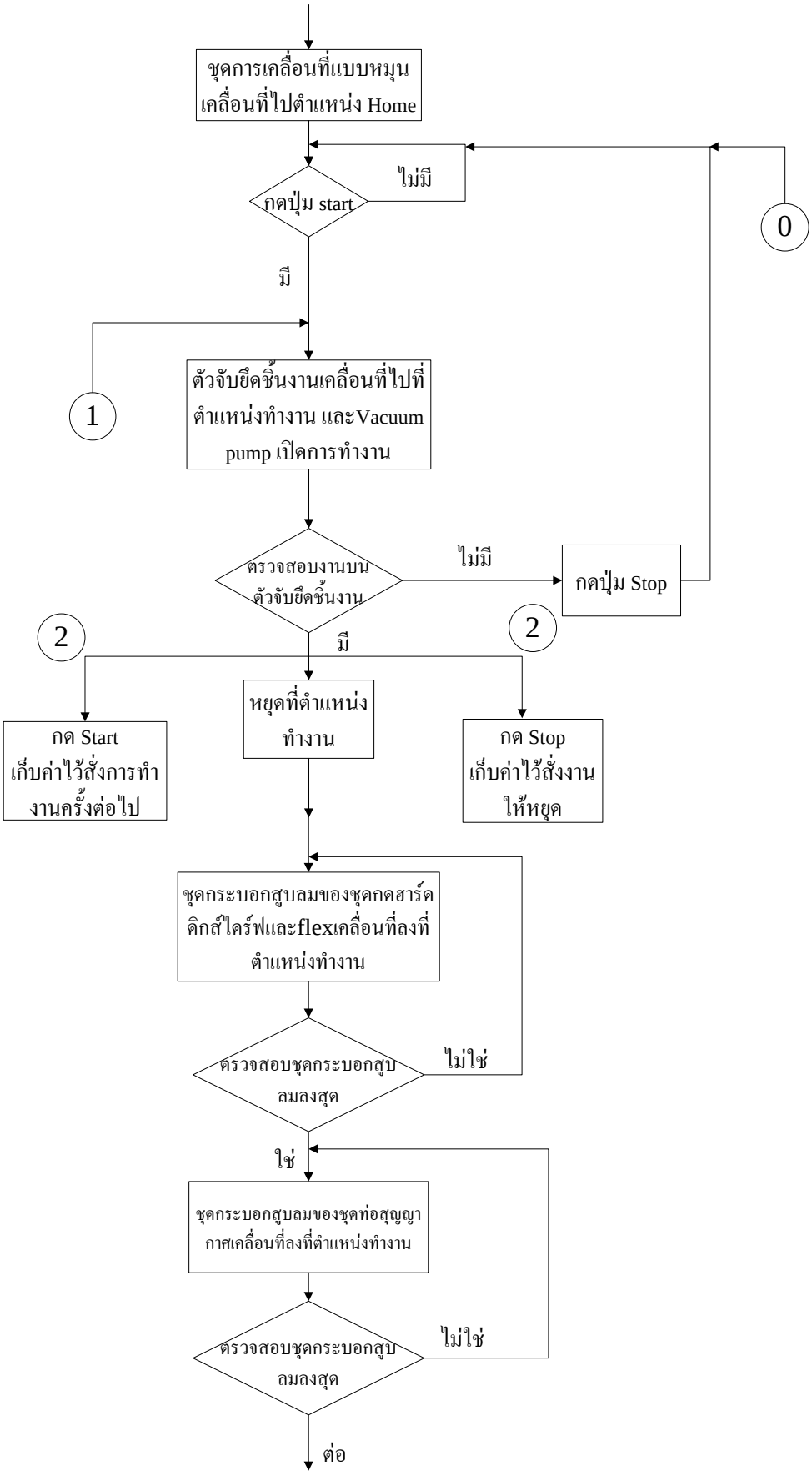
2. AVR MICRO CONTROLLER เป็นรูปแบบการพัฒนาโปรแกรมตามรูปแบบของ AVR ปกติ ซึ่งสามารถเลือกใช้งานโปรแกรมภาษาใด ๆ ที่รองรับการใช้งานร่วมกับ AVR เบอร์ ATMEGA328 เช่น ภาษาเบสิก BASCOM-AVR ภาษาซี CODE VISION และ WIN AVR เป็นต้น โดยใช้การดาวน์โหลดผ่าน BOOTLOADER หรือผ่านทางขั้วต่อ AVR ISP แบบ IDE 10PIN ซึ่งจะต้องมีชุดดาวน์โหลดต่อเพิ่ม

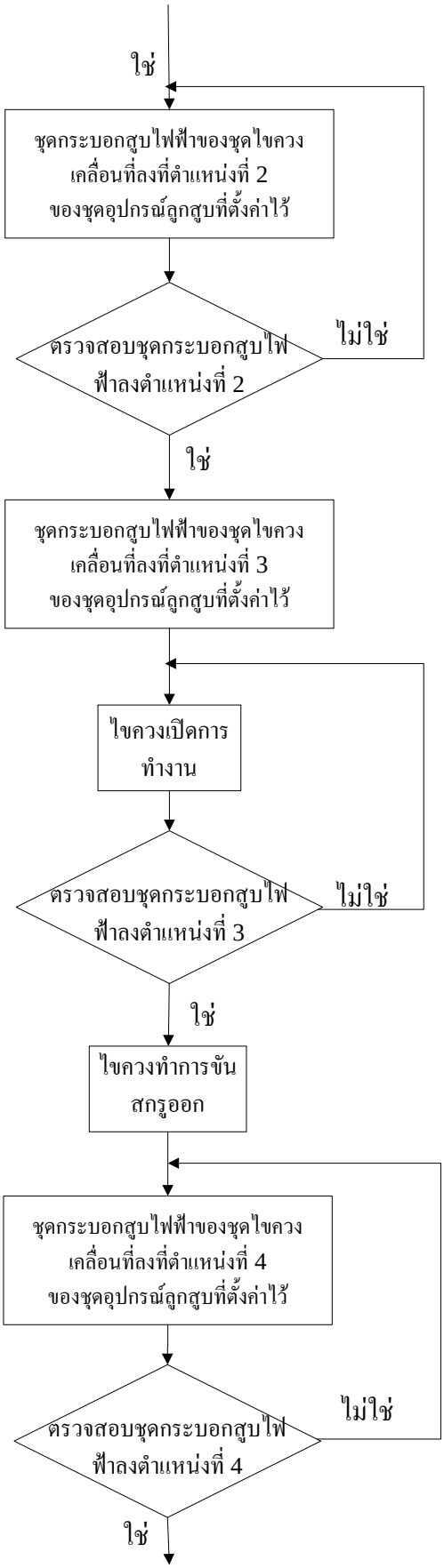
3.3.2 การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ ตระกูล AVR ด้วย Arduino

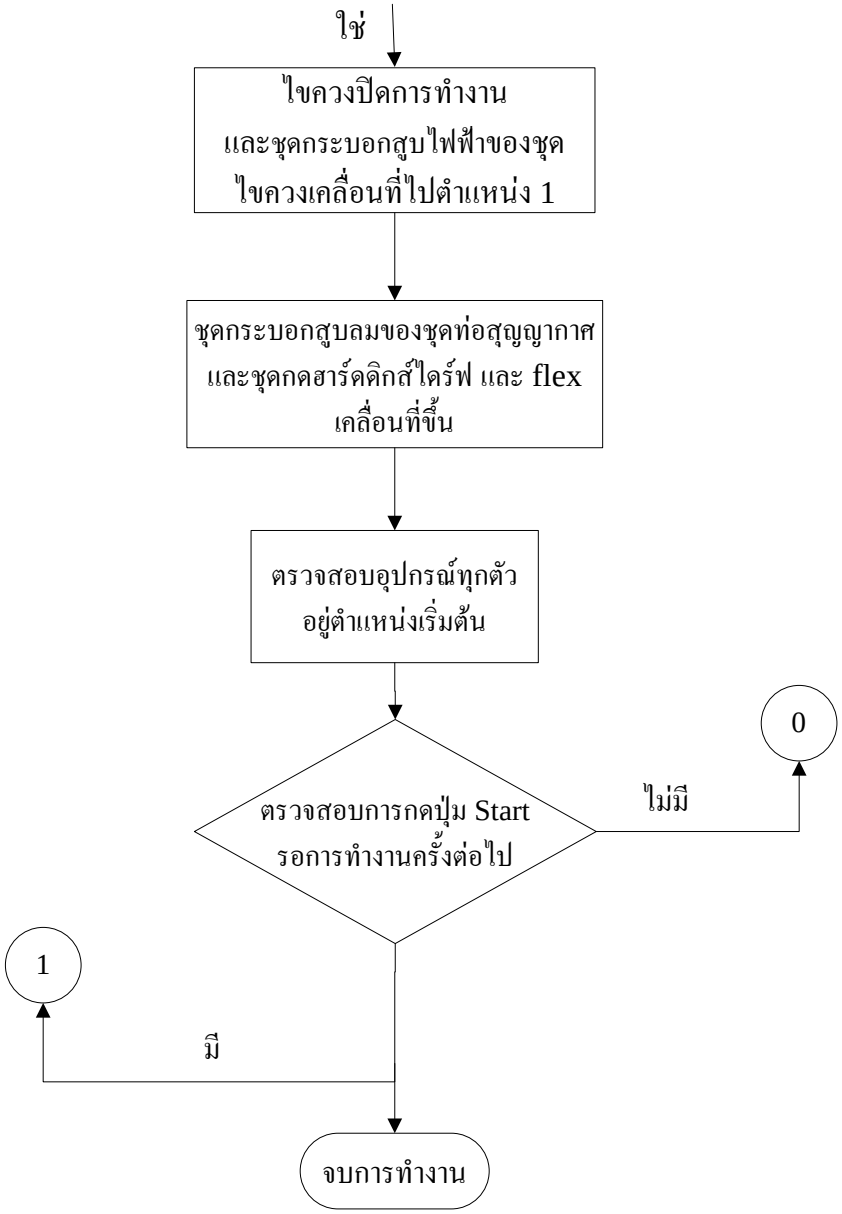
โปรแกรมภาษาของ Arduino จะใช้ภาษาซีพลัสพลัส (C++) ซึ่งเป็นรูปแบบของโปรแกรมภาษาซีประยุกต์แบบหนึ่ง ที่มีโครงสร้างของตัวภาษาโดยรวมใกล้เคียงกันกับภาษาซีมาตรฐาน (ANSI-C) อื่น ๆ เพียงแต่ได้มีการปรับปรุงรูปแบบในการเขียนโปรแกรมบางส่วนที่ผิดเพี้ยนไปจากภาษาซีมาตรฐานเล็กน้อย เพื่อช่วยลดความยุ่งยากในการเขียนโปรแกรมและให้ผู้เขียนโปรแกรมสามารถเขียนโปรแกรมได้ง่ายและสะดวกมากขึ้นกว่าการเขียนภาษาซีตามแบบมาตรฐาน โดยตรง ซึ่งในความเป็นจริงแล้วในการเขียนโปรแกรมของ Arduino เราสามารถใช้คำสั่งต่าง ๆ ที่เป็นคำสั่งตามมาตรฐานของภาษาซีมาตรฐานเข้ามาใช้ในการเขียนโปรแกรมได้ทันที โดยรูปแบบการเขียนโปรแกรม และการใช้งานคำสั่งต่าง ๆ นั้นสามารถอ้างอิงจากหนังสือ ตำราของภาษาซีมาตรฐานได้โดยตรง

3.3.3 แผนผังของโปรแกรมควบคุมการทำงานของเครื่องต้นแบบที่ควบคุมด้วยไมโครคอนโทรลเลอร์




ชุดการเคลื่อนที่แบบหมุน เคลื่อนที่ไปตำแหน่ง Home
กดปุ่ม start
ไม่มี
มี
0
1
ตัวจับยึดชิ้นงานเคลื่อนที่ไปที่ ตำแหน่งทำงาน และVacuum pump เปิดการทำงาน
ตรวจสอบงานบน ตัวจับยึดชิ้นงาน
ไม่มี
กดปุ่ม Stop
2
2
มี
กด Start เก็บค่าไว้สั่งการทำ งานครั้งต่อไป
หยุดที่ตำแหน่ง ทำงาน
กด Stop เก็บค่าไว้สั่งงาน ให้หยุด
ชุดกระบอกสูบลมของชุดกดฮาร์ด ดิกส์ไดร์ฟและflexเคลื่อนที่ลงที่ ตำแหน่งทำงาน
ตรวจสอบชุดกระบอกสูบ ลมลงสุด
ไม่ใช่
ใช่
ชุดกระบอกสูบลมของชุดท่อสุญญา กาศเคลื่อนที่ลงที่ตำแหน่งทำงาน
ตรวจสอบชุดกระบอกสูบ ลมลงสุด
ไม่ใช่
ต่อ


ใช่
ชุดกระบอกสูบไฟฟ้าของชุดไขควง
เคลื่อนที่ลงที่ตำแหน่งที่ 2
ของชุดอุปกรณ์ลูกสูบที่ตั้งค่าไว้
ตรวจสอบชุดกระบอกสูบไฟ
ฟ้าลงตำแหน่งที่ 2
ไม่ใช่
ใช่
ชุดกระบอกสูบไฟฟ้าของชุดไขควง
เคลื่อนที่ลงที่ตำแหน่งที่ 3
ของชุดอุปกรณ์ลูกสูบที่ตั้งค่าไว้
ไขควงเปิดการ
ทำงาน
ตรวจสอบชุดกระบอกสูบไฟ
ฟ้าลงตำแหน่งที่ 3
ไม่ใช่
ใช่
ไขควงทำการขัน
สกรูออก
ชุดกระบอกสูบไฟฟ้าของชุดไขควง
เคลื่อนที่ลงที่ตำแหน่งที่ 4
ของชุดอุปกรณ์ลูกสูบที่ตั้งค่าไว้
ตรวจสอบชุดกระบอกสูบไฟ
ฟ้าลงตำแหน่งที่ 4
ไม่ใช่
ใช่



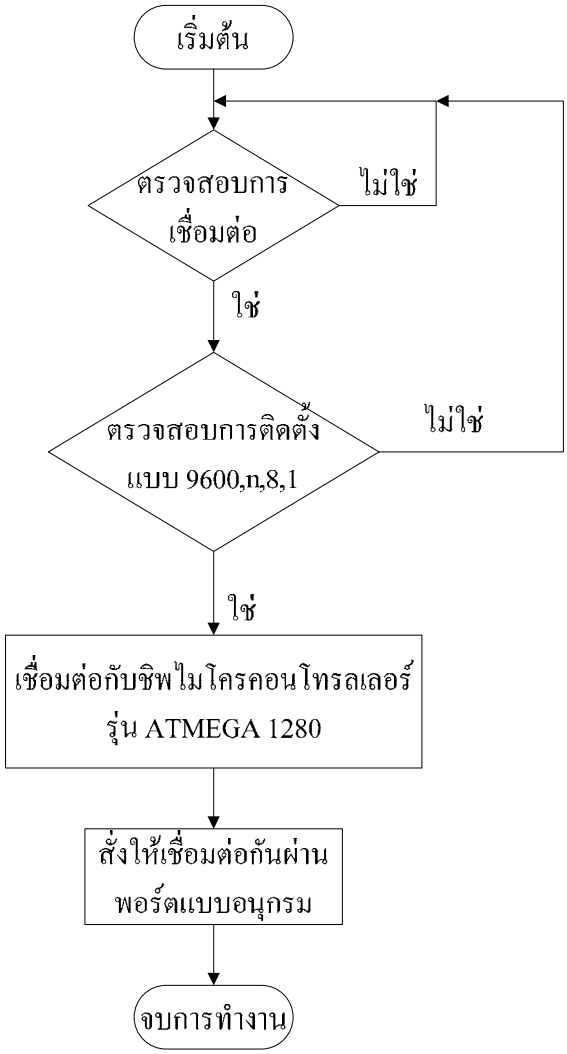
รูปที่ 3.21 แผนผังของโปรแกรมควบคุมการทำงานของเครื่องต้นแบบ

3.3.4 การติดต่อและควบคุมการทำงานของอุปกรณ์ต่าง ๆ ผ่านพอร์ตอนุกรมกับ Microsoft Visual Basic 6.0

จากการออกแบบ และควบคุมระบบการทำงานของเครื่องต้นแบบด้วยไมโครคอนโทรลเลอร์ เมื่อต้องการให้มีการติดต่อ และควบคุมการทำงานของอุปกรณ์ต่าง ๆ ระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ จึงต้องมีการสร้าง Graphic User Interface (GUI) เพื่อสร้างหน้าจอกราฟฟิกบนคอมพิวเตอร์ติดต่อสื่อสารกับไมโครคอนโทรลเลอร์ โดยใช้โปรแกรม

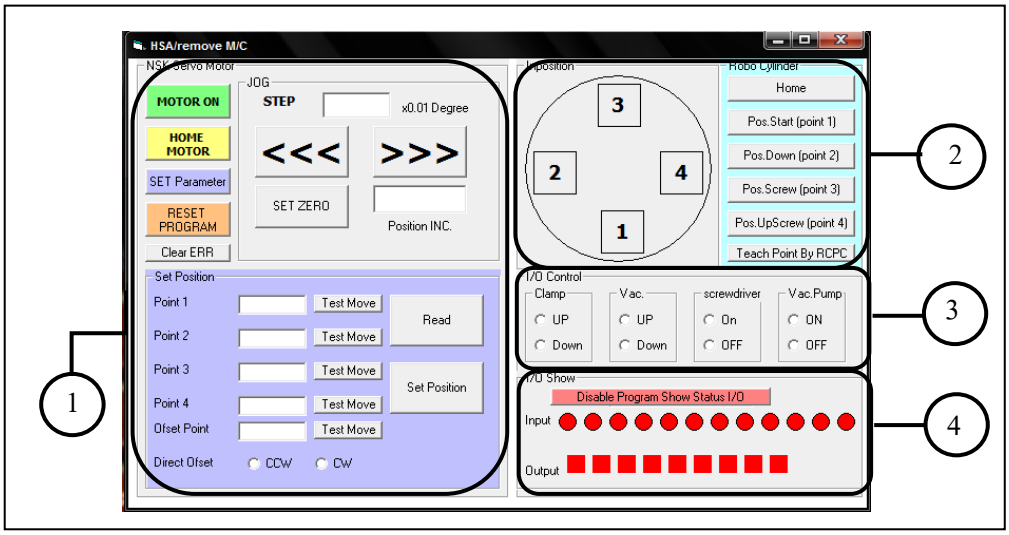
Microsoft Visual Basic 6.0 ซึ่งจากเครื่องต้นแบบ อุปกรณ์ที่ต้องการติดต่อ และควบคุมมีดังนี้ คือ ชุดการเคลื่อนที่แบบหมุน ชุดกระบอกสูบลม ชุดกระบอกสูบไฟฟ้า และชุดไขควง

3.3.4.1 แผนผังแสดงขั้นตอนการติดต่อสื่อสารไมโครคอนโทรลเลอร์กับคอมพิวเตอร์โดยใช้โปรแกรม Microsoft visual basic 6.0

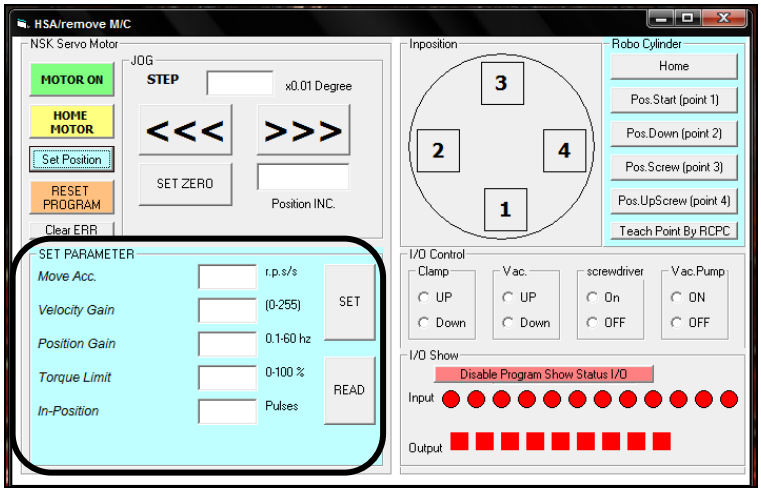


รูปที่ 3.22 แผนผังแสดงขั้นตอนการติดต่อสื่อสารไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

3.3.4.2 หน้าต่างการติดต่อและควบคุมอุปกรณ์โดยใช้โปรแกรม Microsoft Visual Basic 6.0



A) แสดงส่วนการตั้งค่าตำแหน่ง



B) แสดงส่วนของการตั้งค่าตัวแปรของชุดการเคลื่อนที่แบบหมุน

รูปที่ 3.23 หน้าต่างการติดต่อและควบคุมอุปกรณ์ของเครื่องต้นแบบ A) แสดงส่วนการตั้งค่าตำแหน่ง B) แสดงส่วนของการตั้งค่าตัวแปรของชุดการเคลื่อนที่แบบหมุน

จากหน้าต่างการติดต่อและควบคุมอุปกรณ์โดยใช้โปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 รูปที่ 3.23 A และ B อธิบายส่วนต่าง ๆ ได้ดังนี้

ส่วนที่ 1 ติดต่อและควบคุมชุดการเคลื่อนที่แบบหมุน ในส่วนนี้จะมีฟังก์ชันที่ใช้ติดต่อและควบคุมอุปกรณ์ดังนี้ คือ

1. สามารถสั่งงานเปิดและปิดอุปกรณ์ได้
2. สั่งงานให้อุปกรณ์เคลื่อนที่ไปในตำแหน่งเริ่มต้น (Home)
3. ตั้งค่าตำแหน่งสำหรับการทำงาน
4. ตั้งค่าตัวแปรที่ใช้ในการควบคุมอุปกรณ์ให้สามารถทำงานให้เหมาะสมกับกระบวนการ คือ ความเร่ง ความเร็ว Position gain Torque limit และ In-position
5. สามารถตั้งค่าใหม่ของโปรแกรมได้ โดยกดปุ่ม RESET PROGRAM เมื่อสั่งให้ทำงานแล้วเกิดความผิดพลาดขึ้น

ส่วนที่ 2 ติดต่อและควบคุมชุดกระบอกสูบไฟฟ้า ในส่วนนี้จะมีฟังก์ชันที่ใช้ติดต่อและควบคุมอุปกรณ์ดังนี้ คือ

1. ตั้งค่าตำแหน่งในการเคลื่อนที่ของอุปกรณ์ จำนวน 5 ตำแหน่ง โดยในส่วนนี้จะเชื่อมต่อกับโปรแกรมของอุปกรณ์ที่มีมาให้ ที่มีชื่อว่า RCPC ซึ่งโปรแกรมของชุดลูกสูบไฟฟ้าจะสามารถปรับตั้งค่าตำแหน่ง ความเร็ว และเวลาในการเคลื่อนที่ได้ โดยเมื่อกดปุ่มคำสั่ง Teach Point By RCPC หน้าจอของโปรแกรมจะขึ้นให้ตั้งค่าต่าง ๆ ลักษณะของหน้าต่างโปรแกรมดังรูปที่ 3.15
2. ทดสอบการทำงาน โดยสั่งงานให้อุปกรณ์เคลื่อนที่ไปในตำแหน่งที่ต้องการ

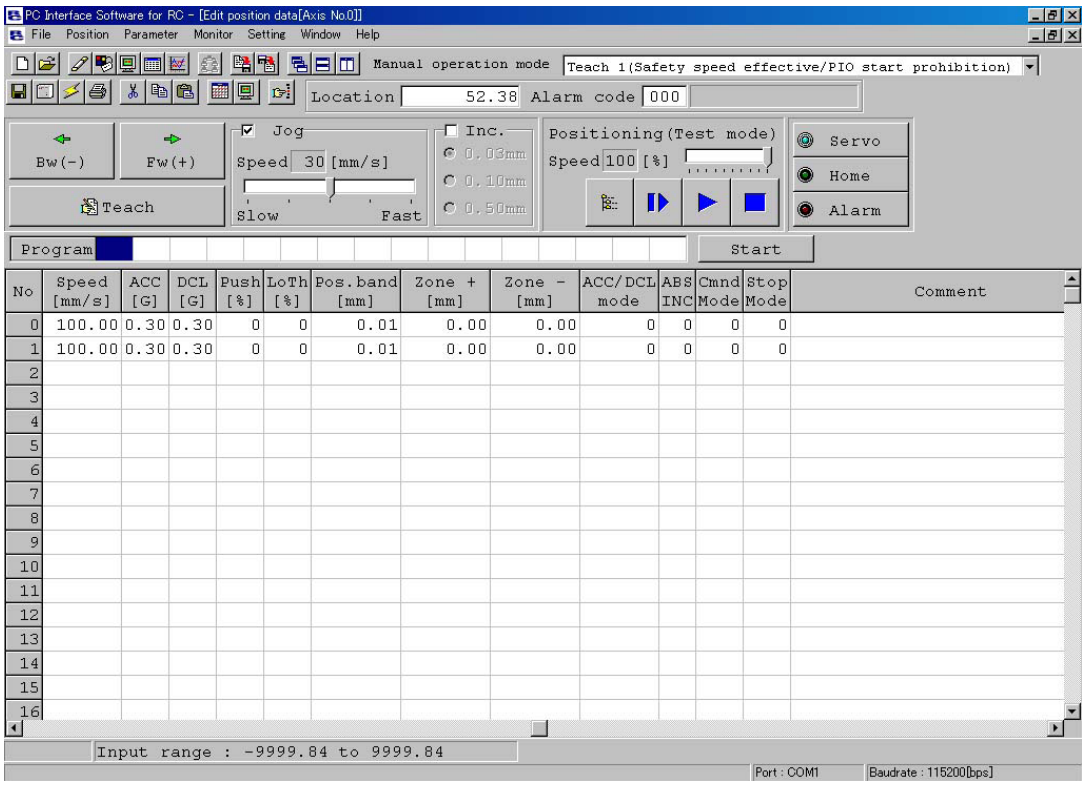
ส่วนที่ 3 ติดต่อและควบคุมชุดกระบอกสูบลมของชุดกดฮาร์ดดิสกไดรฟ์ และ Flex ชุดกระบอกสูบลมของชุดท่อสุญญากาศ การสั่งงานการทำงานของไขควง และการเปิดปิดปั๊มของท่อสุญญากาศ (Vacuum Pump) โดยคำสั่งที่ใช้ติดต่อและควบคุมอุปกรณ์ของแต่ละตัวมีรายละเอียดดังนี้

1. การติดต่อและควบคุมชุดกระบอกลมของชุดกดฮาร์ดดิสก์ไดรฟ์ และ Flex ชุดกระบอกลมของชุดท่อสุญญากาศ จะสั่งงานในส่วนของการเคลื่อนที่ขึ้นและลง
2. การสั่งงานไขควงและการเปิด-ปิดปั๊มของท่อสุญญากาศ เพื่อทดสอบการทำงาน โดยตั้งค่าการทำงานของไขควงในการขันสกรูออกให้แรงบิดที่ค่า 2.0 กิโลกรัมแรงเซนติเมตร ซึ่งการตั้งค่าคุณสมบัติของไขควงสำหรับการทำงานสามารถตั้งค่าได้จากโปรแกรมของอุปกรณ์

ส่วนที่ 4 แสดงไฟสถานะการทำงานของบอร์ดไมโครคอนโทรลเลอร์

สำหรับหน้างต่างการติดต่อและควบคุมอุปกรณ์จะใช้งานเมื่อต้องการทดสอบ และติดตั้งตำแหน่งการทำงานของเครื่องต้นแบบ ซึ่งค่าต่าง ๆ ที่ใช้ในการติดตั้งอุปกรณ์แต่ละตัวจะเป็น

ค่าที่ได้จากคู่มือการใช้งานของอุปกรณ์ตามคุณสมบัติของอุปกรณ์ โดยในการตั้งค่าการทำงานของอุปกรณ์จะต้องตั้งค่าให้สามารถทำงานให้เหมาะสมและสอดคล้องกัน



รูปที่ 3.24 หน้าต่างการตั้งค่าของโปรแกรม RCPC

บทที่ 4
ผลการทดลองและวิเคราะห์ผลการทดลอง

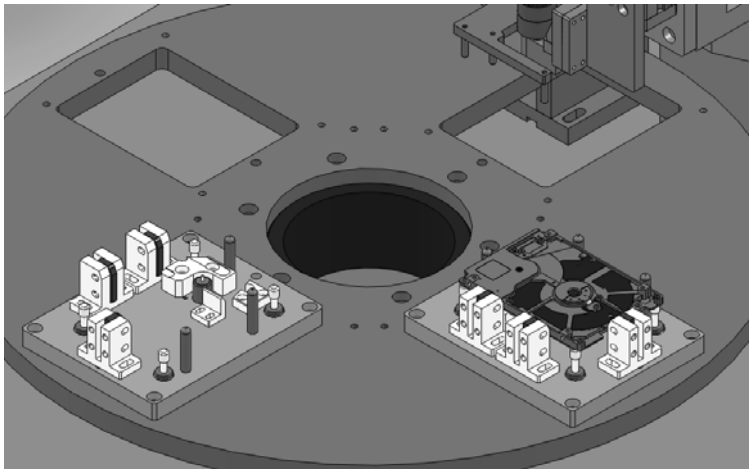
4.1 ลำดับการทำงานของเครื่องต้นแบบ

จากการศึกษากระบวนการต่าง ๆ ในการถอด HSA/VCM โดยใช้แรงงานคน เมื่อทำการออกแบบเครื่องต้นแบบด้วยโปรแกรม SolidWorks 2008 และดำเนินการสร้างเครื่องต้นแบบที่ใช้ในกระบวนการถอด HSA/VCM ได้ลักษณะของเครื่องต้นแบบดังแสดงในรูปที่ 4.1

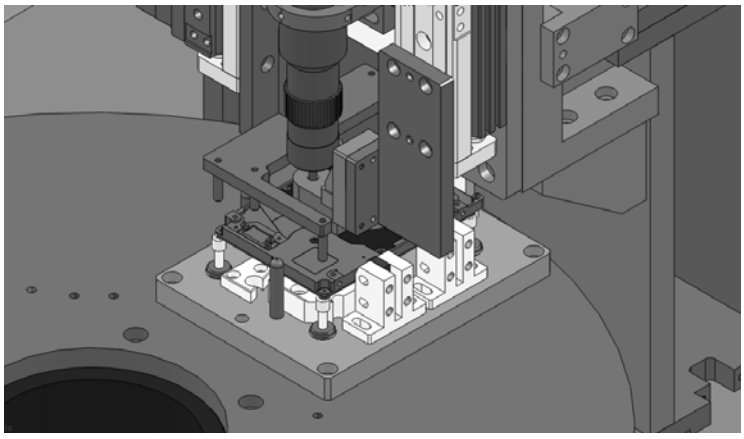
รูปที่ 4.1 เครื่องต้นแบบ

เครื่องต้นแบบที่สร้างขึ้น จำนวนของชุดจับยึดชิ้นงานจะดำเนินการสร้างขึ้นเพียง 2 ชุด เพื่อลดค่าใช้จ่ายในการขึ้นรูปชิ้นงาน ดังนั้นในการทดสอบการทำงานของเครื่องต้นแบบ จะทดสอบการทำงานได้แค่ 2 กระบวนการ ซึ่งจะไม่สามารถทดสอบได้ครบรอบวัฏจักรของกระบวนการ โดยมีขั้นตอนการทำงานของกระบวนการดังนี้

1. พนักงานนำฮาร์ดดิสก์ไดรฟ์วางบนชุดจับยึดชิ้นงาน และพนักงานกดสวิตซ์สั่งงานให้ชุดการเคลื่อนที่แบบหมุนทำการหมุน 90 องศา เพื่อเคลื่อนที่นำชุดจับยึดชิ้นงานเข้ามา ณ ตำแหน่งทำงาน

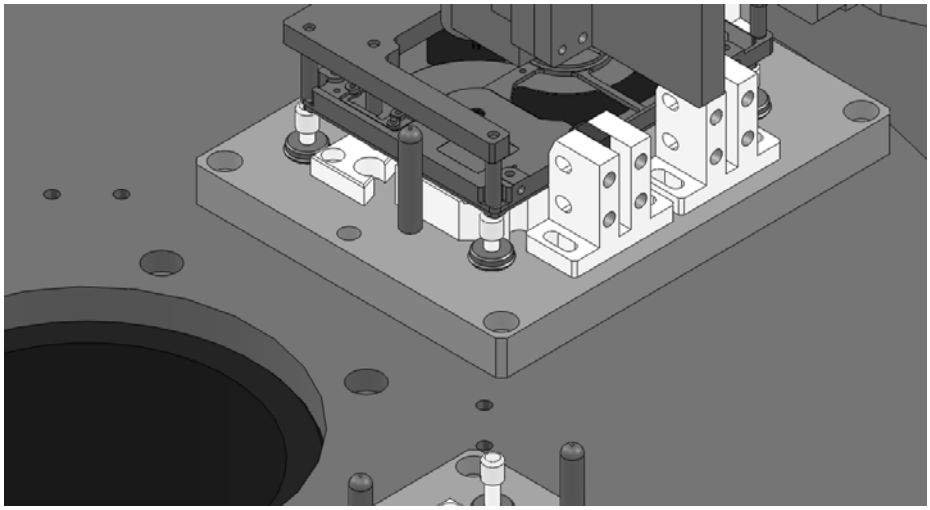


รูปที่ 4.2 การวางฮาร์ดดิสก์ไดรฟ์บนชุดจับยึดชิ้นงาน



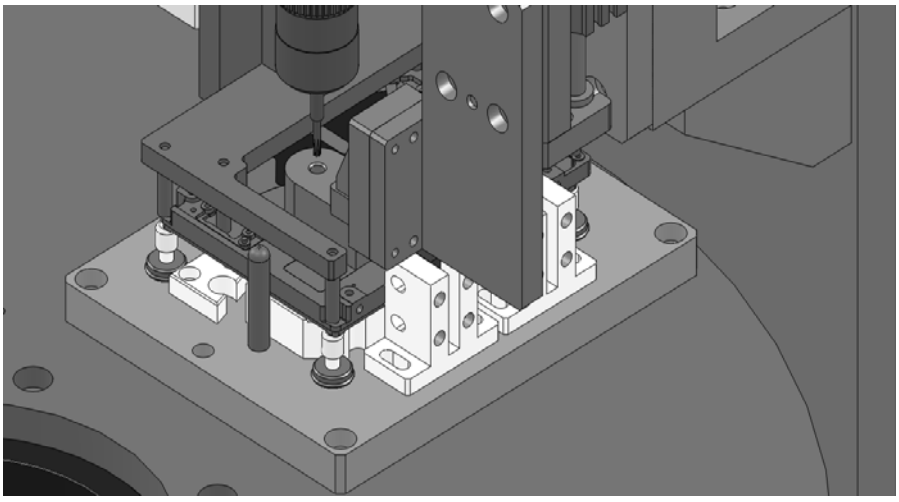
รูปที่ 4.3 ชิ้นงานเคลื่อนที่เข้ามาที่ตำแหน่งทำงาน

2. ชุดกระบอกสูบลมยึดอุปกรณ์กดฮาร์ดดิสก์ไดรฟ์ และกด flex จะเคลื่อนที่ลงมากดฮาร์ดดิสก์ไดรฟ์พร้อมทั้งกด flex ให้หลุด ในขณะเดียวกัน HSA จะหมุนออกจากแผ่นข้อมูล โดยตัวที่ช่วยในการหลุดของ HSA ออกจากแผ่นข้อมูล เพื่อไม่ให้แผ่นข้อมูลเสียหายในขณะทำการขันสกรูออก



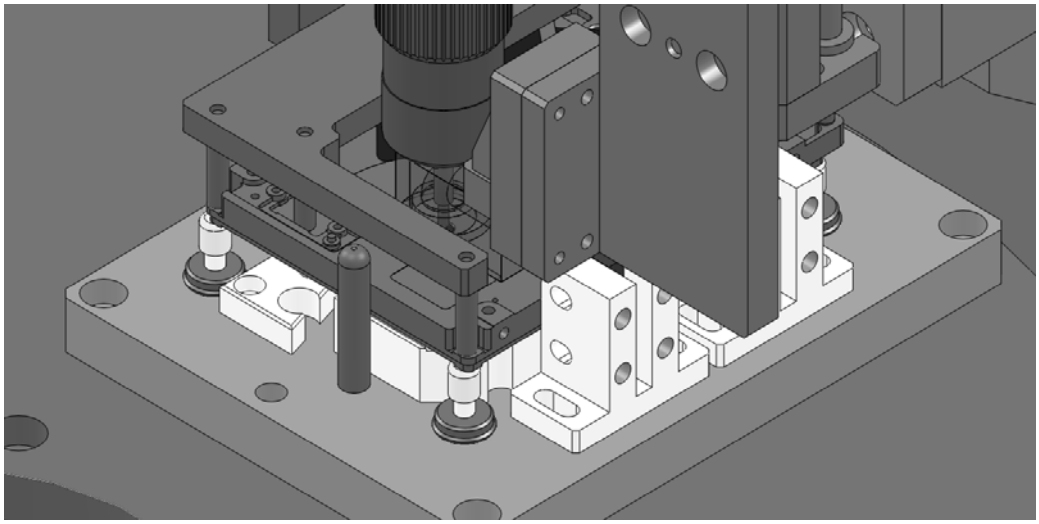
รูปที่ 4.4 การกดฮาร์ดดิสก์ไดรฟ์และกด flex

3. ชุดกระบอกสูบลมยึดชุดท่อสุญญากาศเคลื่อนที่ลงมา ณ ตำแหน่งทำงานและเป็นแนวทางนำเพื่อให้ชุดไขควงเคลื่อนที่ลงมาขันสกรูออก



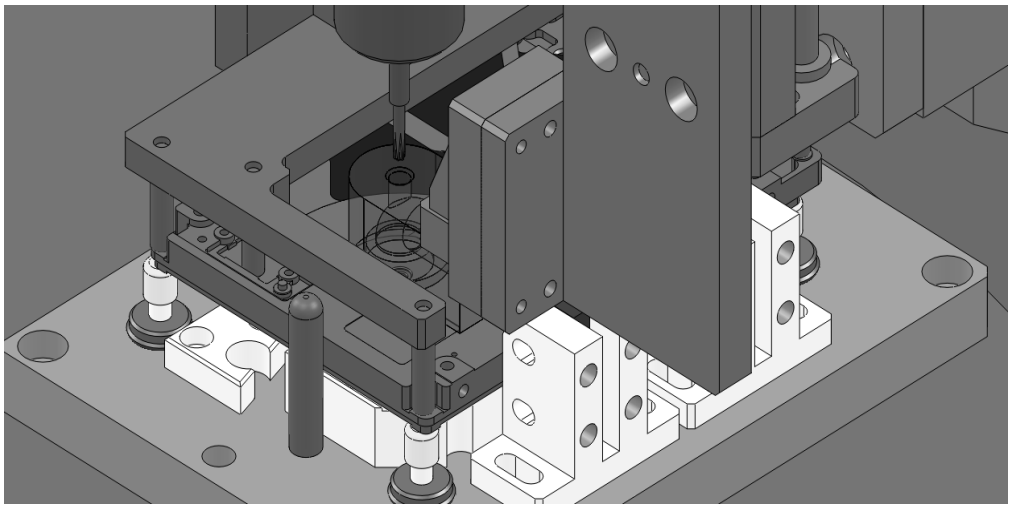
รูปที่ 4.5 ตำแหน่งปฏิบัติงานของชุดกระบอกสูบลมยึดชุดท่อสุญญากาศ

4. ชุดกระบอกสูบไฟฟ้ายึดไขควงเคลื่อนที่ลงมาเพื่อทำการขันสกรูออก ในขณะเดียวกันชุดท่อสุญญากาศจะดูดอนุภาคต่าง ๆ ที่เกิดจากกระบวนการขันสกรูออก



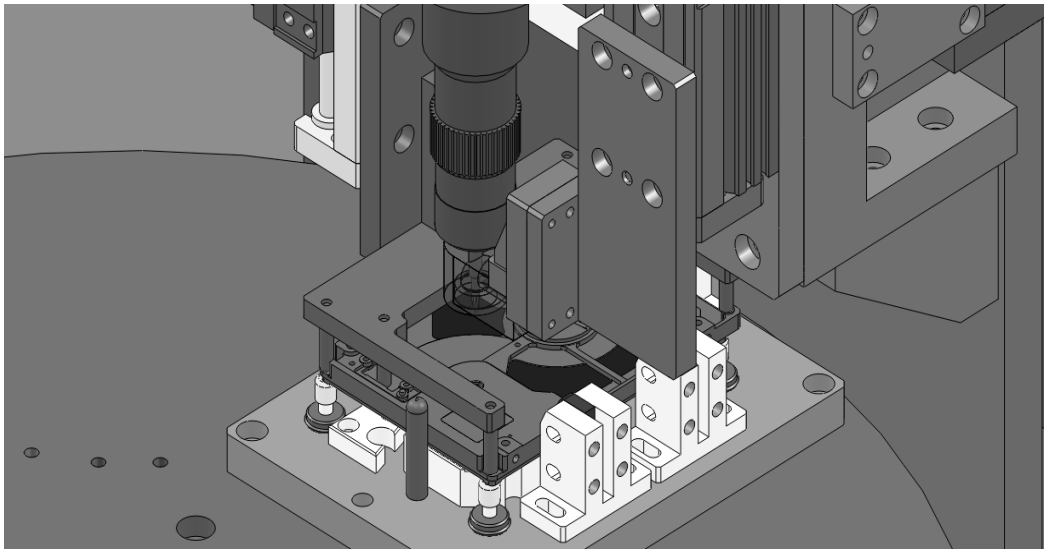
รูปที่ 4.6 ตำแหน่งปฏิบัติงานของชุดกระบอกสูบไฟฟ้ายึดไขควง

5. ขันสกรูออกเสร็จ ชุดกระบอกสูบไฟฟ้ายึดไขควงเคลื่อนที่ขึ้น ในขณะเดียวกันชุดท่อสุญญากาศยังคงทำงานเพื่อดูดอนุภาคต่าง ๆ และสกรูไปที่บริเวณที่จัดเก็บ



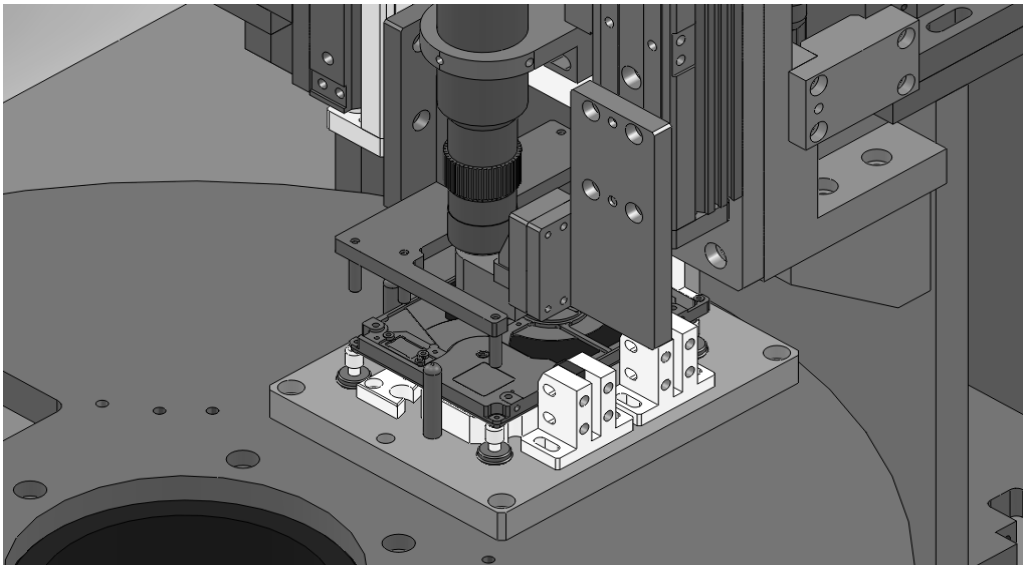
รูปที่ 4.7 ตำแหน่งของชุดกระบอกสูบไฟฟ้ายึดไขควงเมื่อเคลื่อนที่ขึ้นไปตำแหน่งบน

6. ชุดกระบอกสูบลมยึดชุดท่อสุญญากาศเคลื่อนที่ขึ้น



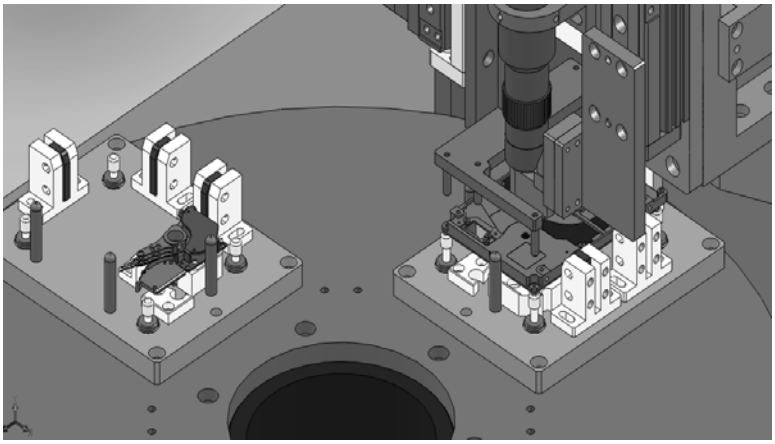
รูปที่ 4.8 ตำแหน่งของชุดกระบอกสูบลมยึดชุดท่อสุญญากาศเมื่อเคลื่อนที่ขึ้น

7. ชุดกระบอกสูบลมยึดอุปกรณ์กดฮาร์ดดิสก์ไดรฟ์ และกด flex เคลื่อนที่ขึ้น



รูปที่ 4.9 ตำแหน่งชุดกระบอกสูบลมยึดอุปกรณ์กดฮาร์ดดิสก์ไดรฟ์ และกด flex เมื่อเคลื่อนที่ขึ้น

8. ชุดการเคลื่อนที่แบบหมุนทำการหมุน 90 องศา เพื่อเคลื่อนที่ชุดจับยึดชิ้นงานออกมาจากตำแหน่งทำงาน และเคลื่อนที่นำชุดจับยึดชิ้นงานและชิ้นงานชุดใหม่เข้าสู่ตำแหน่งทำงาน และจากนั้นพนักงานจะทำการหยิบ HAS VCM และ flex ที่วางอยู่บนอุปกรณ์รองรับออกไว้ที่ถาดวาง โดย HSA จะต้องใส่ head clip เพื่อป้องกันไม่ให้หัวอ่านและเขียนข้อมูลติดกัน



รูปที่ 4.10 ลักษณะของ HSA/VCM และ flex อยู่บนที่พักบนชุดจับยึดชิ้นงาน

4.2 การปรับตั้งค่าตำแหน่งและตัวแปรของเครื่องต้นแบบ

การติดตั้งเครื่องจักรเพื่อให้เครื่องจักรสามารถทำงานได้มีประสิทธิภาพ จะต้องมีการปรับตั้งค่าตัวแปรต่างๆของอุปกรณ์ที่ติดตั้งในเครื่องจักร สำหรับการปรับตั้งค่าของอุปกรณ์ต่าง ๆ ผ่านหน้าต่างการติดต่อ และควบคุมอุปกรณ์โดยผ่านหน้าต่างการติดต่อที่เรียกว่า GUI โดยใช้โปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 อุปกรณ์ชุดแรกที่ต้องทำการปรับตั้งค่าตัวแปร และตำแหน่งในการติดตั้งของเครื่องต้นแบบ คือ ชุดการเคลื่อนที่แบบหมุนโดยจะต้องปรับตำแหน่งให้ตรงกับชุดอุปกรณ์กดฮาร์ดดิสก์ไดรฟ์ และกด flex และอุปกรณ์ชุดที่สองที่ต้องปรับค่าตำแหน่ง และตัวแปรคือชุดกระบอกสูบไฟฟ้ายึดไขควง สำหรับการปรับค่าตำแหน่ง และตัวแปรในการเคลื่อนที่ของชุดกระบอกสูบไฟฟ้ายึดไขควง จะใช้โปรแกรมของชุดกระบอกสูบไฟฟ้าที่มีมาให้ในการปรับตั้งค่า และจากนั้นจะดำเนินการปรับค่าตำแหน่งของไขควงให้ตรงกับตำแหน่งในการขันสกรู ส่วนค่าแรงบิดในการขันสกรูออกของไขควงสามารถปรับค่าได้จากกล่องควบคุม โดยค่าแรงบิดในการขันสกรูออกของเครื่องต้นแบบใช้ค่าแรงบิดเท่ากับ 2.0 กิโลกรัมแรงเซนติเมตร เมื่อปรับตำแหน่งของชุดไขควงแล้ว อุปกรณ์ชุดสุดท้ายที่ต้องปรับตำแหน่งคือ ชุดกระบอกสูบลมยึดชุดท่อสุญญากาศโดยปรับตำแหน่งให้ตรงกับไขควง

เครื่องต้นแบบที่ทำการสร้างขึ้นนั้น นอกจากจะต้องทำงานได้ตามวัตถุประสงค์แล้ว ยังต้องคำนึงถึงเวลา และแรงกระแทกที่มีผลต่อความเสียหายของชิ้นงานด้วย จึงได้ทำการทดลองปรับค่าอุปกรณ์เพื่อหาเวลาที่ดีที่สุดของกระบวนการถอด HSA/VCM โดยปรับค่าแรงดัน ความเร็ว และความเร่งในการเคลื่อนที่ที่มีผลต่อเวลา และแรงที่กระทำต่อชิ้นงานจากอุปกรณ์ดังนี้ คือ ชุดการเคลื่อนที่แบบหมุน ชุดกระบอกสูบไฟฟ้ายึดไขควง ชุดกระบอกสูบลมยึดชุดท่อสุญญากาศ และชุดกระบอกสูบลมยึดอุปกรณ์กดฮาร์ดดิสก์ไดรฟ์ และกด flex เพื่อหาว่าที่ตำแหน่ง และค่าตัวแปรใดสามารถใช้เวลาในการทำงานได้น้อยที่สุดและไม่ทำให้ชิ้นงานเสียหายหลังจากกระบวนการขันสกรูออก

จากการทดลองปรับค่าอุปกรณ์จำนวน 10 ครั้ง เมื่อนำเวลาที่ได้จากการทดลองมารวมกับเวลาในขั้นตอนต่าง ๆ ในกระบวนการ สามารถแสดงเวลารวมในแต่ละครั้งได้ดังตารางที่ 4.1

จากตารางที่ 4.1 ดังนั้นเวลาที่ใช้ในกระบวนการถอด HSA/VCM ที่ใช้เวลาในการทำงานได้น้อยที่สุดคือ 13.23 วินาที แบ่งเวลาในแต่ละขั้นตอนการทำงานได้ดังตารางที่ 4.2

ตารางที่ 4.1 ตารางบันทึกเวลาจากการปรับค่าอุปกรณ์จำนวน 10 ครั้ง

No.	Cycle time per process (s)
1	14.03
2	13.89
3	13.84
4	13.74
5	13.62
6	13.53
7	13.46
8	13.41
9	13.31
10	13.23

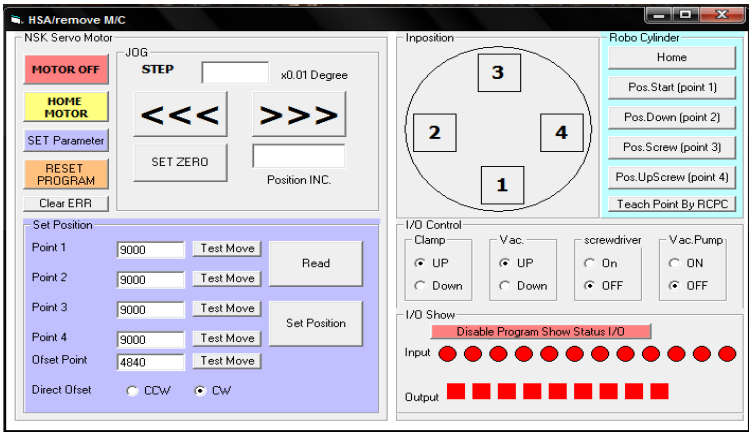
ตารางที่ 4.2 แสดงเวลาที่ใช้ในแต่ละขั้นตอนการทำงาน

STEP	PROCESS	TIME(SEC)
1	Operator load unit	2
2	Servo motor rotate 90 degree, pneumatic cylinder of clamp and vacuum move down	0.4
3	Electrical cylinder of screw driver down , screw driver remove screw and drive up	5.53
4	Pneumatic cylinder of clamp and vacuum move up, Servo motor forward 90 degree	1.2
5	Operator separate HSA,VCM and Flex then load unit out	4
Total time		13.23

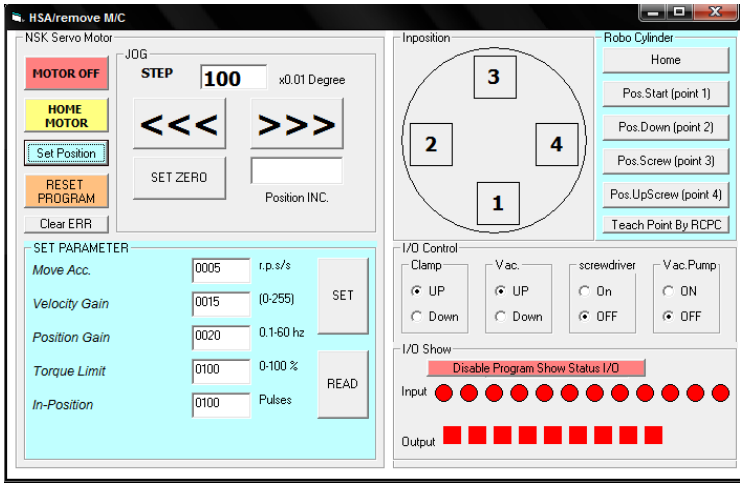
เวลาในการปฏิบัติงานตลอดทั้งกระบวนการจากกระบวนการเดิมที่ใช้มนุษย์ในการถอดประกอบนั้นใช้เวลา 19.52 วินาที เมื่อเปรียบเทียบเวลาจากเครื่องต้นแบบที่ทำการออกแบบนี้จะมีการลดของเวลาลงได้ประมาณร้อยละ 30 ของกระบวนการเดิม และที่เวลานี้เมื่อดูผลของชิ้นงานหลังจากกระบวนการขันสกรูออกพบว่า ชิ้นงานไม่เกิดความเสียหาย

รูปที่ 4.11 ลักษณะของชิ้นงานภายหลังจากกระบวนการขันสกรูออก

จากค่าที่ให้เวลาในการทำงานที่ดีที่สุด และชิ้นงานภายหลังจากกระบวนการขันสกรูออกที่ไม่เกิดความเสียหาย สามารถแสดงการปรับตั้งค่าตำแหน่งในการเคลื่อนที่ และตัวแปรต่าง ๆ ของอุปกรณ์ได้ดังนี้คือ ชุดการเคลื่อนแบบหมุนแสดงการปรับตั้งค่าดังรูปที่ 4.12 A และ B) และการปรับตั้งค่าของชุดกระบอกสูบไฟฟ้ายึดไขควง โดยปรับค่าตำแหน่ง และตัวแปรที่โปรแกรม RCPC ของอุปกรณ์แสดงดังรูปที่ 4.13

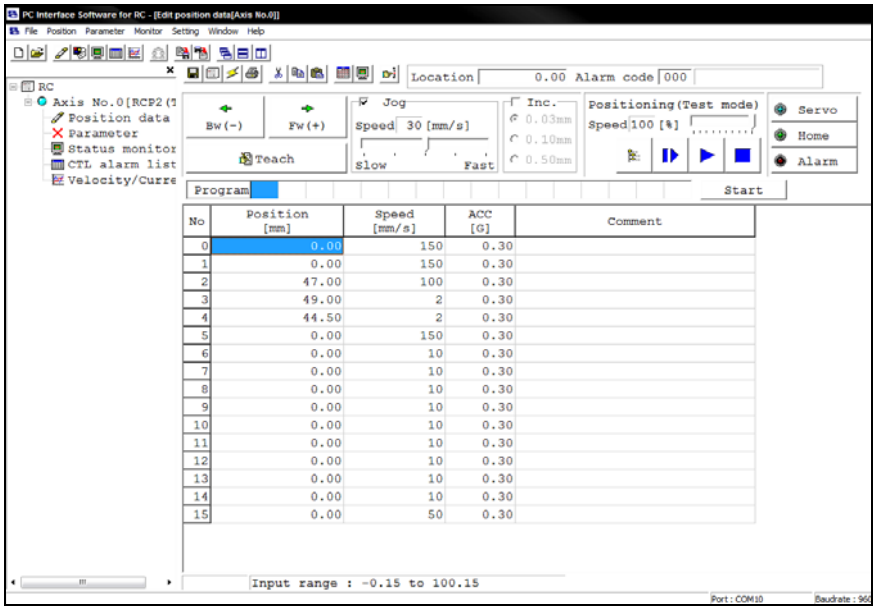


A) ค่าตำแหน่งที่ปรับ



B) ค่าตัวแปรที่ปรับ

รูปที่ 4.12 การปรับค่าที่ให้เวลาในการทำงานดีที่สุดของชุดการเคลื่อนที่แบบหมุน
A) ค่าตำแหน่งที่ทำ B) ค่าตัวแปรที่ปรับ



รูปที่ 4.13 ค่าตำแหน่ง และตัวแปรที่ทำการปรับที่ให้เวลาในการทำงานดีที่สุดของชุดกระบอกสูบไฟฟ้ายึดไขควง

4.3 การวิเคราะห์แรงกระแทก

จากการออกแบบการทำงานของเครื่องต้นแบบ การเสียหายของชิ้นส่วนต่างๆของชิ้นงานเกิดขึ้นมาจากแรงกระแทกในกระบวนการทำงาน โดยการวิเคราะห์แรงกระแทกจะพิจารณาจากความเร่งสูงสุดที่เกิดขึ้น เพราะบ่งบอกถึงแรงสูงสุดที่กระทำ ดังนั้นจึงต้องทำการออกแบบให้ความเร่งของวัตถุที่เกิดขึ้นไม่เกินค่าที่บริษัทผู้ผลิตยอมรับได้ คือที่ค่าเท่ากับ 60 G สำหรับกระบวนการที่มีชิ้นส่วนของชุดหัวอ่าน (HSA) ซึ่งจากการเลือกใช้อุปกรณ์ในการทำงานของเครื่องต้นแบบ สามารถวิเคราะห์ค่าความเร่งของวัตถุที่เกิดขึ้นจากขั้นตอนการทำงานต่าง ๆ ของกระบวนการได้จากค่าคุณสมบัติของอุปกรณ์ โดยแรงกระแทกที่มาจากขั้นตอนต่าง ๆ มีดังนี้ คือ

1. การกดยึดฮาร์ดดิสก์ไดรฟ์และกด Flex

ความเสียหายที่เกิดขึ้นกับชิ้นส่วนต่าง ๆ จะเกิดในช่วงที่ชุดกดยึดฮาร์ดดิสก์ไดรฟ์ และกด flex ลงมากระทบชิ้นงาน มีการกระแทกเกิดขึ้น 3 ทิศทาง คือ ทิศทางแกน x และ y มาจากชิ้นงานกระทบกับชุดจับยึดชิ้นงานทางด้านข้าง และทิศทางแกน z มาจากชิ้นงานรับแรงกระแทกจากชุดกดยึดฮาร์ดดิสก์ไดรฟ์ และกด flex โดยมาจากการเคลื่อนที่ของชุดกระบอกสูบลมที่เคลื่อนที่ลงมากดยึดฮาร์ดดิสก์ไดรฟ์ และกด flex ซึ่งค่าความเร่งของวัตถุที่มากระทำสามารถคำนวณหาได้ดังนี้

แรงกระแทกที่เกิดขึ้นมาจากแรงดันลมของกระบอกสูบ จากค่าคุณสมบัติของกระบอกสูบลมที่เลือกใช้ รุ่น MXQ12-50CS ให้แรงดัน 0.15-0.7 MPa และให้แรงกดขนาด 45-158 นิวตัน โดยในการคำนวณจะเลือกใช้ค่าแรงที่มากที่สุด เพื่อทำการวิเคราะห์ค่าความเร่งของวัตถุที่มากระทำที่ค่าแรงมากที่สุดสามารถยอมรับได้หรือไม่

จากคุณสมบัติของกระบอกสูบลม และชิ้นงานที่ใช้ในกระบวนการ แรงดันกระบอกสูบลม (P) 0.7 MPa ให้แรงกดเท่ากับ 178 นิวตัน โดยค่าแรงกดหามาจาก

$$F = P \times A \tag{4.1}$$

เมื่อ Diameter piston (D) = 6 mm ดังนั้น Piston area (A) $= \frac{\pi D^2}{4} = 266\ mm^2$ จากสมการการเคลื่อนที่ 4.2

$$\sum F = ma \tag{4.2}$$

เมื่อ $m_{total} = mass_{clamp} + mass_{cylinder}$ โดย $mass_{clamp} = 0.400\ kg, mass_{cylinder} = 0.525\ kg$ ดังนั้น $m_{total} = 0.400 + 0.525 = 0.925\ kg$ แทนค่า F และ m ลงในสมการที่ (4.2) จะได้ $a = 170.81\ {}^{m}\!/_{s^2}$ วัดในหน่วยของความเร่งเนื่องจากแรงโน้มถ่วง ได้เป็น g = 17.41 G

2. การเคลื่อนที่ของชุดท่อสุญญากาศ

ความเสียหายที่เกิดขึ้นกับชิ้นส่วนต่าง ๆ จะเกิดในช่วงชุดท่อสุญญากาศลงมากระทบชิ้นงาน มีการกระแทกเกิดขึ้น 1 ทิศทาง คือ ทิศทางแกน z มาจากการกระแทกของการเคลื่อนที่ของชุดกระบอกสูบลมที่เคลื่อนที่ลงมา ณ ตำแหน่งบนชิ้นงาน เพื่อเป็นแนวทางนำเพื่อให้ชุดไขควงเคลื่อนที่ลงมาขันสกรูออก และดูดสกรู และอนุภาคต่าง ๆ ที่เกิดจากกระบวนการขันสกรูออก ซึ่งแรงกระแทกที่เกิดขึ้นในขั้นตอนนี้มาจากแรงดันลมของกระบอกสูบ โดยกระบอกสูบลมที่ใช้เป็นรุ่นเดียวกับขั้นตอนการกดยึดฮาร์ดดิสก์ไดรฟ์ และกด flex ดังนั้นค่าความเร่งของวัตถุที่มากระทำที่ได้จากการคำนวณจะมีค่าดังนี้

จากคุณสมบัติของกระบอกสูบลม และชิ้นงานที่ใช้ในกระบวนการ แรงดันกระบอกสูบลม (P) 0.7 MPa ให้แรงกดเท่ากับ 178 นิวตัน โดยค่าแรงกดหามาจากสมการที่ 4.1 เมื่อ Diameter piston (D) = 6 mm ดังนั้น Piston area (A) = 266 mm^2 และจากสมการ การเคลื่อนที่ (4.2) เมื่อ $m_{total} = mass_{vacuum} + mass_{cylinder}$ โดย $mass_{vacuum} = 0.130\ kg$ และ $mass_{cylinder} = 0.525\ kg$

ดังนั้น $m_{total} = 0.400 + 0.525 = 0.925\ kg$ แทนค่า F และ m ลงในสมการที่ 4.2 จะได้ $a = 170.81\ {}^{m}\!/_{s^2}$ วัดในหน่วยของ Gravity ได้เป็น $g = 24.59\ G$

3. การเคลื่อนที่ของชุดไขควง

ความเสียหายที่เกิดขึ้นกับชิ้นส่วนต่าง ๆ จะเกิดในช่วงชุดไขควงลงมากระทบชิ้นงาน บริเวณตัวสกรู มีการกระแทกเกิดขึ้น 1 ทิศทาง คือ ทิศทางแกน z มาจากการกระแทกของเคลื่อนที่ของชุดกระบอกสูบไฟฟ้าที่เคลื่อนที่ลงมาตำแหน่งบนชิ้นงาน ซึ่งค่าความเร่งของวัตถุที่มากระทำที่เกิดขึ้นจากคุณสมบัติของกระบอกสูบไฟฟ้าที่เลือกใช้ รุ่น RCP2CR-SA6-I-PM-2-100-P1-S-BL-JP ให้ความเร่ง 0.1-0.3G และความเร็ว 1-150 mm/s ซึ่งจากค่าความเร่งของอุปกรณ์จะเห็นได้ว่ามีค่าไม่เกินค่าที่บริษัทผู้ผลิตกำหนด ถ้ากำหนดค่าความเร่งที่ค่ามากที่สุดในการทำงาน

จากการวิเคราะห์ค่าความเร่งของวัตถุที่มากระทำที่ทำให้เกิดการกระแทกกับชิ้นงานในขั้นตอนการทำงานต่าง ๆ ในกระบวนการ จากการคำนวณจะเห็นว่าค่าความเร่งของวัตถุที่มากระทำที่เกิดขึ้นไม่เกินค่าความเร่งที่ทางบริษัทผู้ผลิตกำหนดไว้

บทที่ 5
สรุปและข้อเสนอแนะ

งานวิจัยนี้เป็นการสร้างเครื่องต้นแบบในกระบวนการถอด HSA/VCM โดยอัตโนมัติในขั้นตอนการขันสกรูในตำแหน่งที่ยึด HSA กับฐานออก กด flex ดูดสกรู และอนุภาคต่าง ๆ ภายหลังจากขั้นตอนขันสกรูออก จากการทำการทดสอบการทำงานของเครื่องต้นแบบ จะเห็นได้ว่าเครื่องต้นแบบใช้เวลาในการกระบวนการเท่ากับ 13.23 วินาที สามารถช่วยลดเวลาได้ประมาณร้อยละ 30 จากกระบวนการเดิมที่ใช้แรงงานคน และจากการวิเคราะห์ค่าความเร่งของวัตถุที่มากระทำที่เกิดขึ้นในกระบวนการจากขั้นตอนในการทำงานต่าง ๆ จากการคำนวณค่าความเร่งของวัตถุที่มากระทำค่าที่ได้ไม่เกินค่าที่ทางบริษัทผู้ผลิตกำหนดที่ค่า 60G ซึ่งค่าความเร่งของวัตถุที่มากระทำจากขั้นตอนการกดยึดฮาร์ดดิสก์ไดรฟ์ และกด flex มีค่าเท่ากับ 17.41G ขั้นตอนการเคลื่อนที่ของชุดท่อสุญญากาศลงมาที่ชิ้นงานมีค่าเท่ากับ 24.59G และขั้นตอนการเคลื่อนที่ของชุดไขควงลงมาที่ชิ้นงานบริเวณสกรูมีค่าเท่ากับ 0.3G จะเห็นว่าการลดปัญหาความเสียหายต่อชิ้นงานภายหลังจากกระบวนการถอดประกอบ และกระบวนการทำงานสามารถทำงานได้ได้ดีในระดับหนึ่ง เนื่องจากค่าความเร่งของวัตถุที่มากระทำที่ได้เกิดจากการคำนวณ จะต้องมีการนำเครื่องต้นแบบเข้าไปทดสอบค่าความเร่งของวัตถุที่มากระทำที่บริษัทผู้ผลิต เพื่อที่จะได้ค่าความเร่งของวัตถุที่มากระทำที่เกิดขึ้นจริงจากกระบวนการทำงานของเครื่องต้นแบบ

สำหรับการเครื่องต้นแบบที่สร้างขึ้นถ้าจะนำไปใช้ในอุตสาหกรรมจริง เพื่อลดการนำเข้าเครื่องจักรจากต่างประเทศ ยังคงต้องปรับปรุง และพัฒนาในส่วนต่าง ๆ ดังนี้คือ

1. อุปกรณ์ เนื่องจากใช้อุปกรณ์ที่มีอยู่แล้วในบริษัทผู้ผลิต ดังนั้นการเลือกใช้อุปกรณ์จึงมีข้อจำกัดในการเลือกใช้งานให้สามารถทำงานได้อย่างมีประสิทธิภาพ
2. ชุดจับยึดชิ้นงาน ควรออกแบบชุดจับยึดชิ้นงานให้สามารถใช้งานให้มีประสิทธิภาพเพิ่มมากขึ้น ในส่วนของการยึดจับชิ้นงานไม่ให้เคลื่อนที่ เนื่องจากจะเห็นได้ว่าเมื่อนำชิ้นงานวางบนชุดจับยึดชิ้นงานชิ้นงานไม่มีการถูกจับยึด ทำให้เมื่อชิ้นงานเคลื่อนที่ไปที่ตำแหน่งทำงานชิ้นงานอาจเกิดการเคลื่อนที่ได้ และจะทำให้ตำแหน่งในการขันสกรูออกคลาดเคลื่อนไม่ตรงตำแหน่ง และชุดรองรับชิ้นงานในส่วนต่าง ๆ เช่น การเคลื่อนที่ของชุดหัวอ่านออกจากแผ่นดิสก์ ควรให้มีกลไกเพิ่มเข้ามาช่วยในการเคลื่อนที่ออก เพื่อป้องกันความเสียหายของชุดหัวอ่านให้เพิ่มมากขึ้น

3. ระบบควบคุม สำหรับการควบคุมระบบการทำงานของเครื่องจักรอัตโนมัติ เมื่อนำไปพัฒนาใช้ในกระบวนการทางอุตสาหกรรมขนาดใหญ่ ที่มีความซับซ้อนมาก และคำนึงถึงสภาพแวดล้อมที่ใช้โรงงานอุตสาหกรรม เช่นความร้อน ความเย็น ระบบไฟฟ้ารบกวน การสั่นสะเทือน และการกระแทก ควรใช้ระบบควบคุมแบบ Programmable Logic Controller (PLC) เนื่องจาก PLC ถูกออกแบบ และสร้างขึ้นเพื่อให้ทนต่อสภาพแวดล้อมในโรงงานอุตสาหกรรมโดยเฉพาะ การโปรแกรมและการใช้งาน PLC ทำได้ง่ายไม่ยุ่งยาก มีระบบการตรวจสอบตัวเองตั้งแต่ช่วงติดตั้ง จนถึงช่วงการใช้งาน ทำให้การบำรุงรักษาทำได้ง่าย ตลอดจน PLC ถูกพัฒนาให้มีความสามารถในการตัดสินใจสูงขึ้นเรื่อย ๆ ทำให้การใช้งานสะดวก

รายการอ้างอิง

มนูกิจ พานิชกุล และอรรณพ เรืองวิเศษ (แปลและเรียบเรียง). (2548). **แนวคิดและวิธีการออกแบบเครื่องจักรกล.** กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น)

ดอนสัน ปงผาบ. (2548). **Microcontroller and application**. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น).

อภิชาติ ภู่พลับ. (2552). **เขียนโปรแกรมควบคุม Microcontroller ด้วยภาษา C, Assembly และ Visual Basic**. (พิมพ์ครั้งที่ 1). นนทบุรี : บริษัท ไอดีซี อินโฟ ดิสทริบิวเตอร์ เซ็นเตอร์ จำกัด.

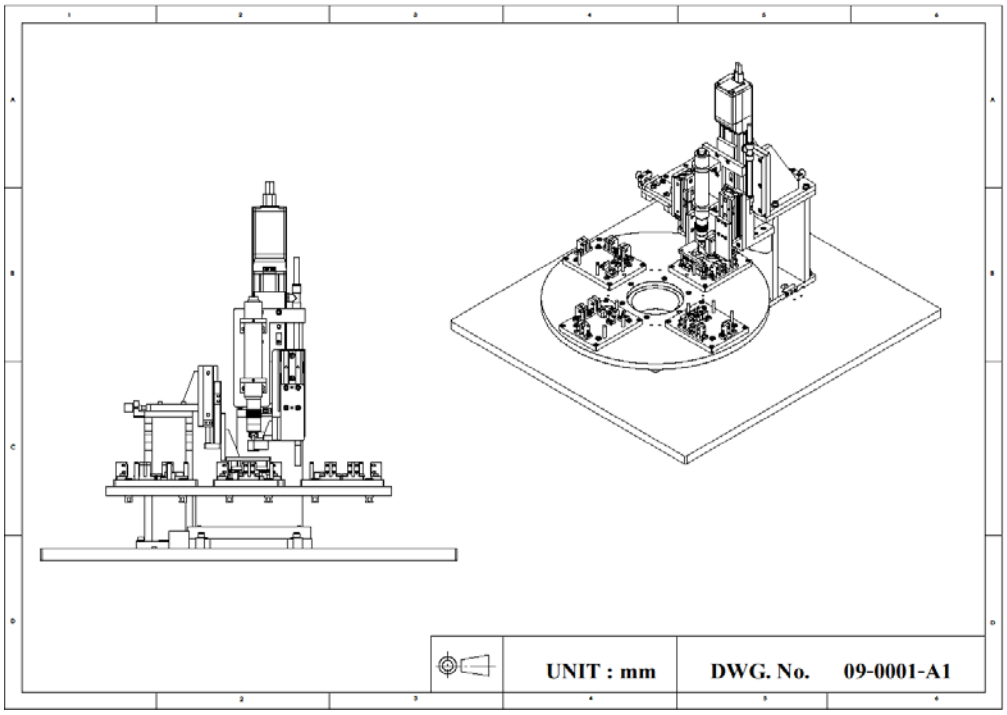
โอภาส ศิริครรชิตถาวร, วรพจน์ กรแก้ววัฒนกุล และชัยวัฒน์ ลิ้มพรจิตรวิไล. (2537). **เรียนรู้ระบบควบคุมอย่างง่ายด้วยโปรแกรมภาษา C กับ Arduino และบอร์ดไมโครคอนโทรลเลอร์ POP-168**. กรุงเทพฯ : บริษัท อินโนเวตีฟ เอ็กเพอริเมนต์ จำกัด.

Joseph, E.S. Charles, R.M. and Richard, G.B. (1989) **Mechanical Engineering Design**. (7^{th} edition), New York, McGrawHill.

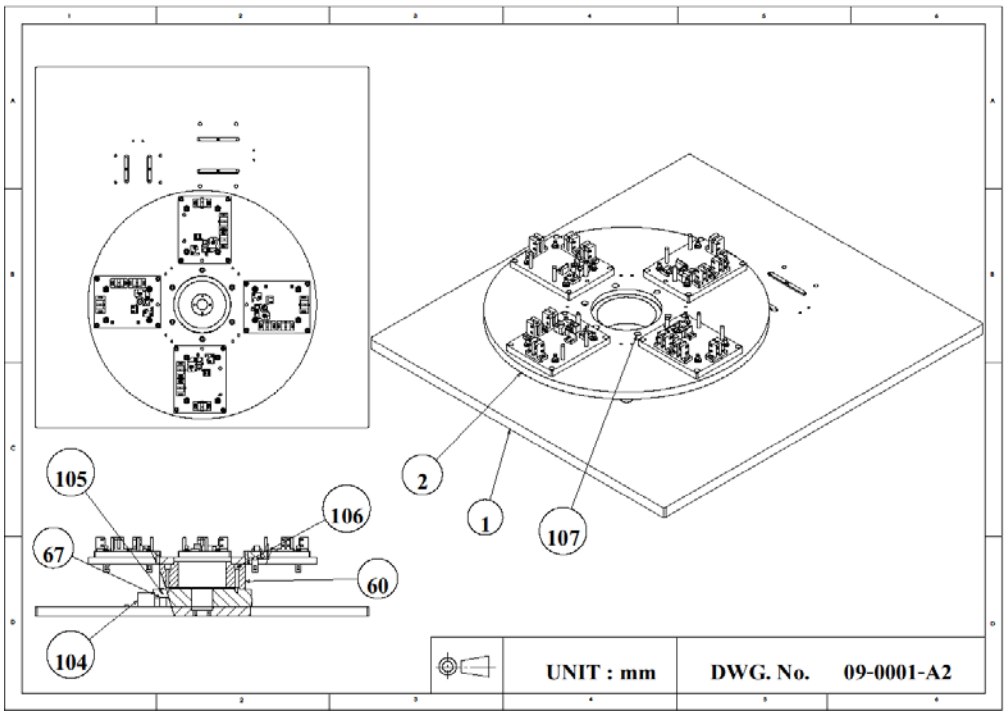
Katsuhiko, O. (2002) **Modern Control Engineering.** (4^{th} Edition), USA, Prentice Hall.

ภาคผนวก ก

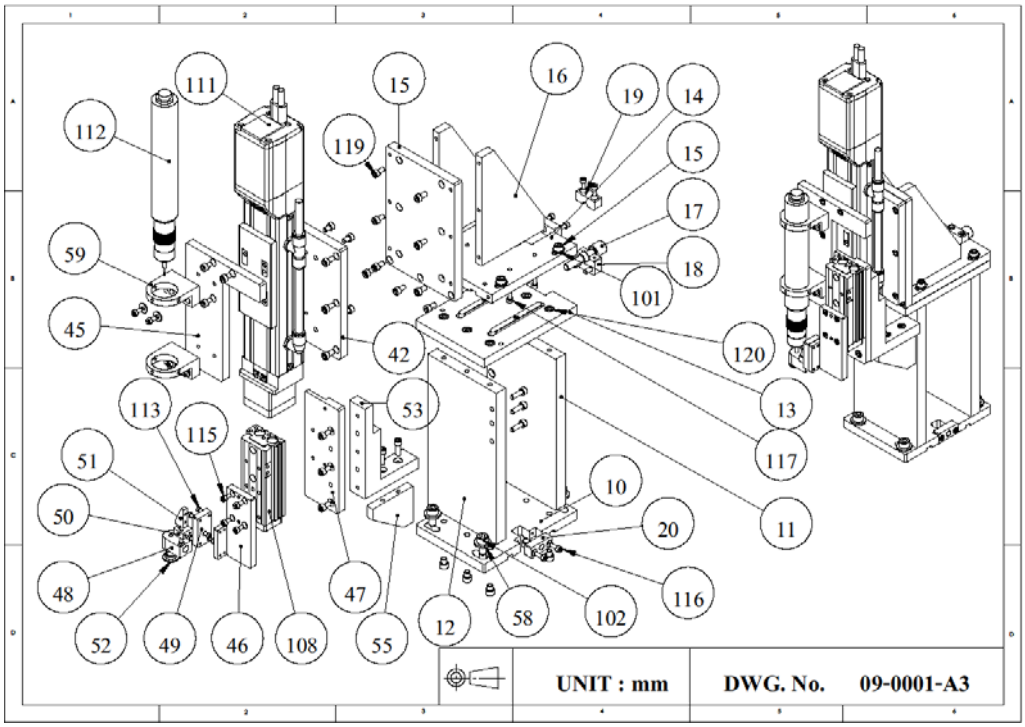
รายละเอียดขนาดชิ้นส่วนเครื่องต้นแบบ



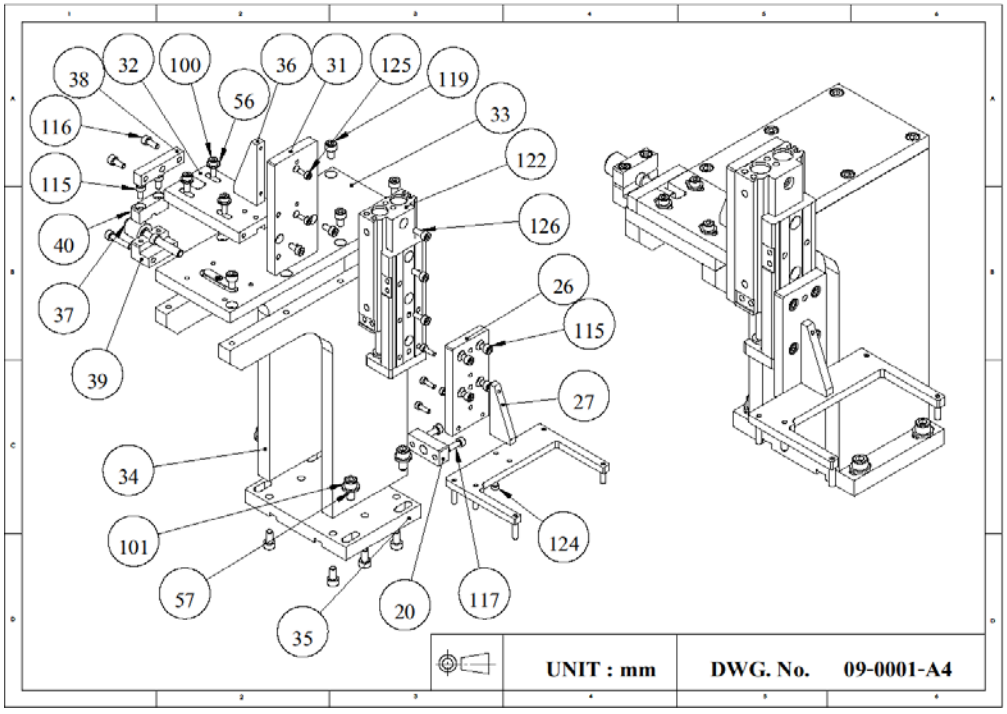
รูป ก.1 องค์ประกอบของเครื่องต้นแบบ



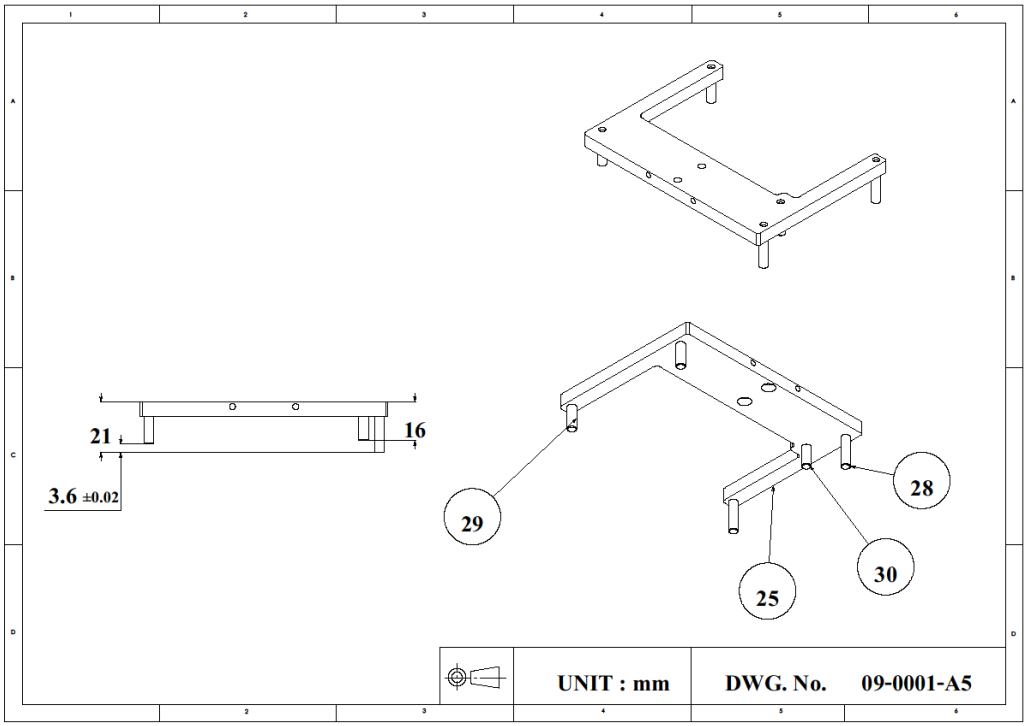
รูป ก.2 หมายเลขของชิ้นส่วนต่าง ๆ ของเครื่องต้นแบบ



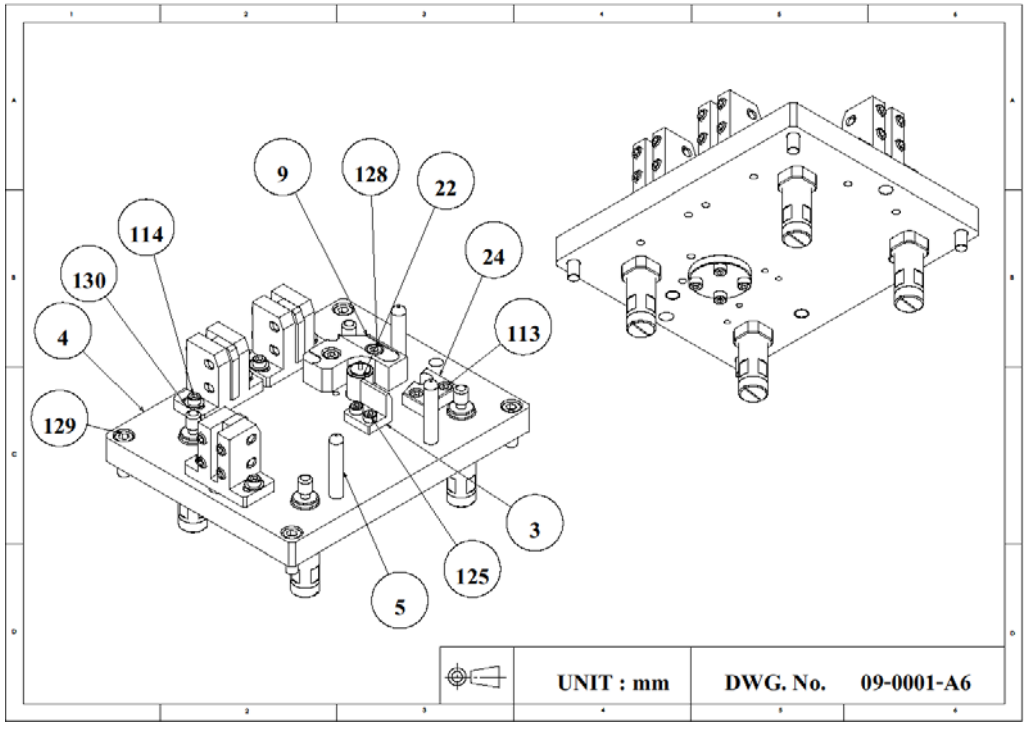
รูป ก.3 หมายเลขของชิ้นส่วนในส่วนชุดขาตั้งยึดกระบอกสูบไฟฟ้า



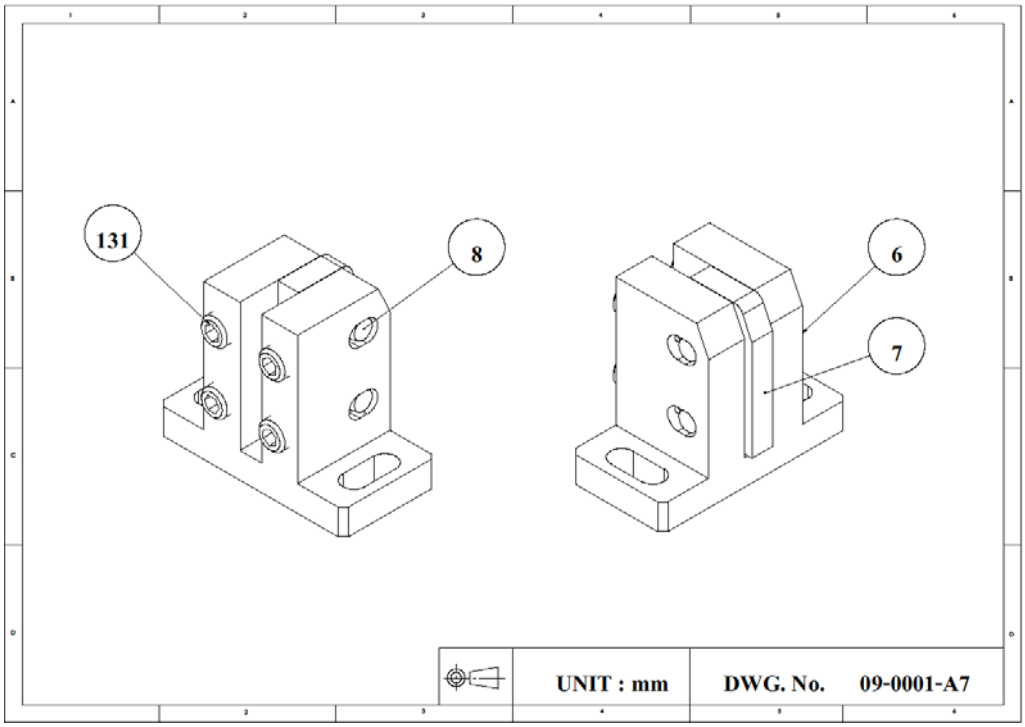
รูป ก.4 หมายเลขของชิ้นส่วนในส่วนชุดขาตั้งยึดกระบอกสูบลมของชุดกดฮาร์ดดิส์ไดรฟ์



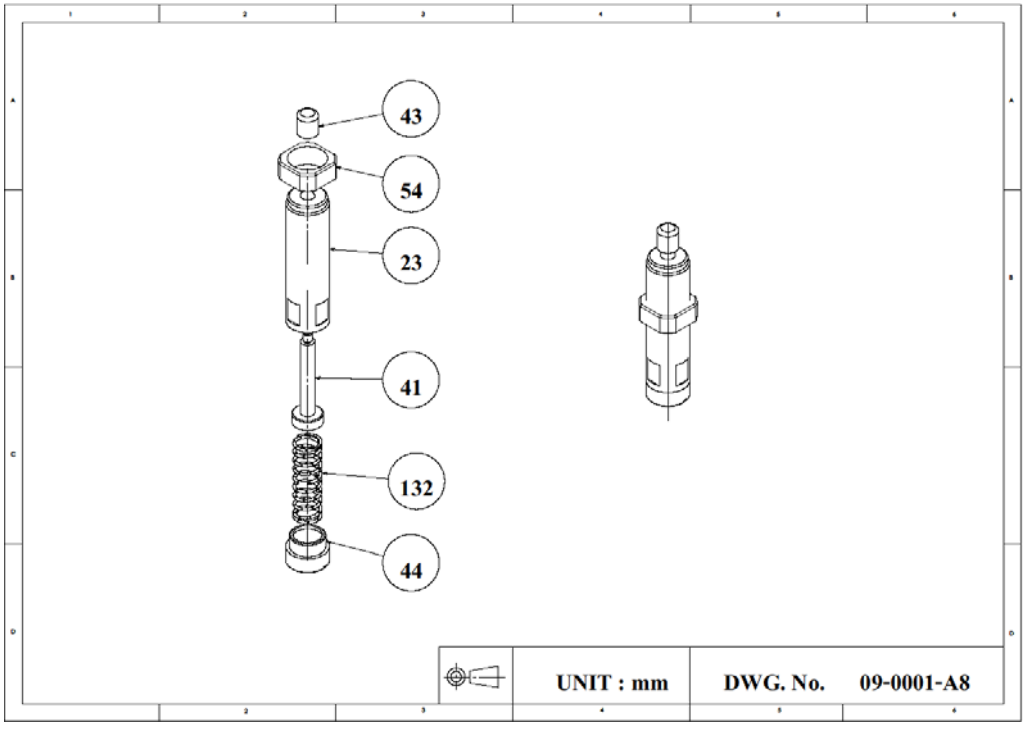
รูป ก.5 หมายเลขของชิ้นส่วนของชุดกดฮาร์ดดิสก์ไดรฟ์



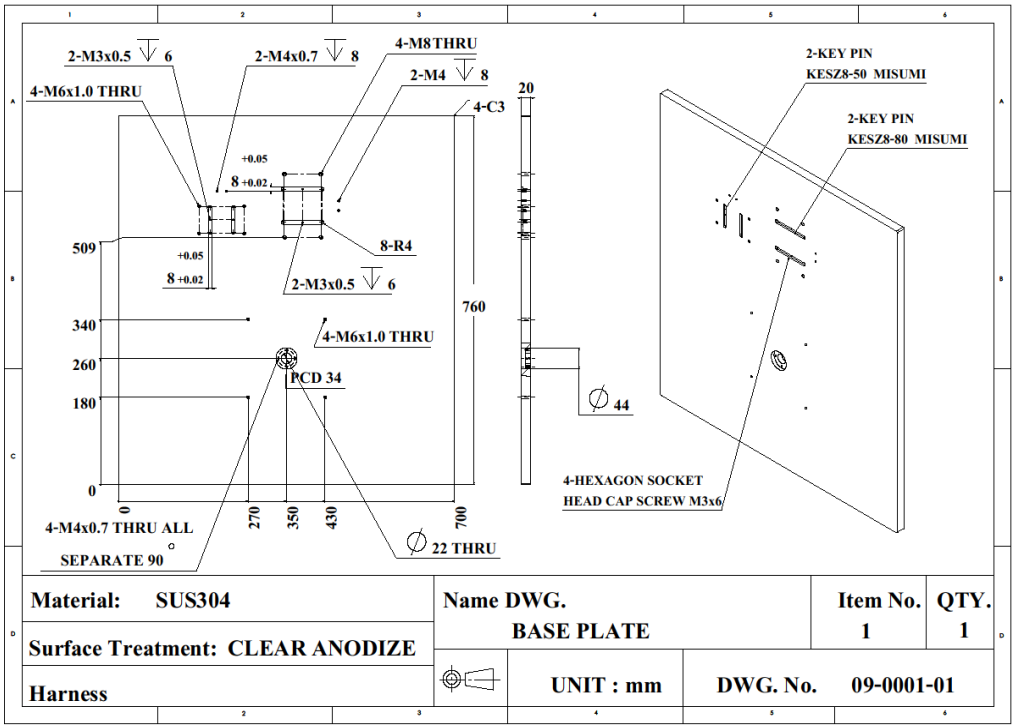
รูป ก.6 หมายเลขชิ้นส่วนของชุดจับยึดชิ้นงาน



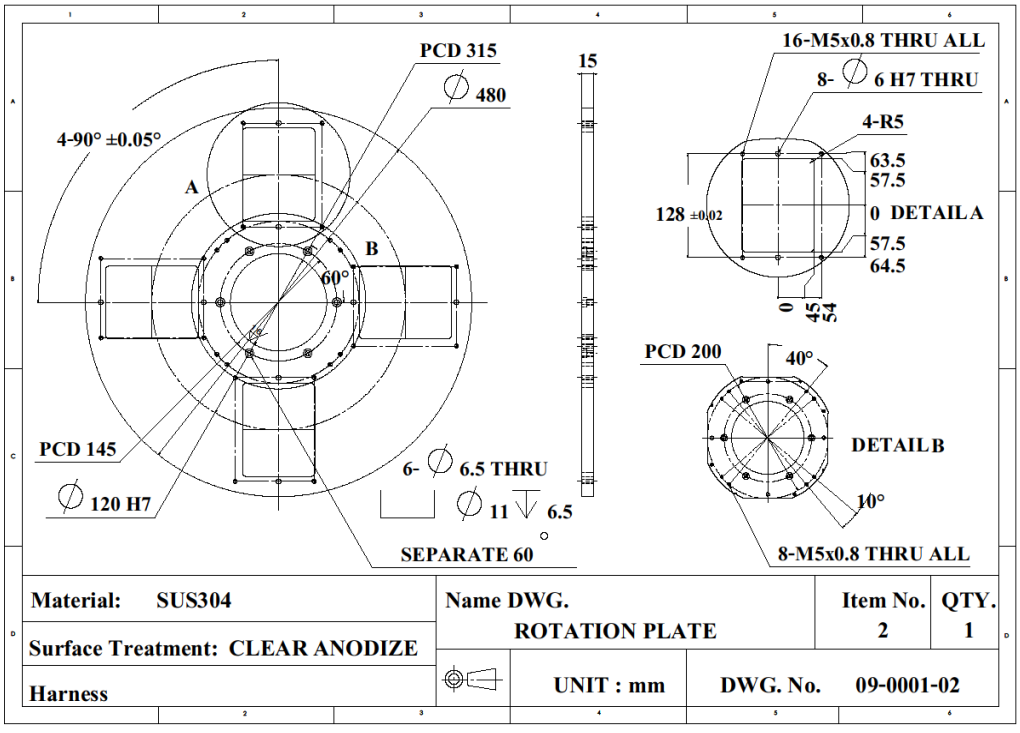
รูป ก.7 หมายเลขชิ้นส่วนของชุด HOUSING



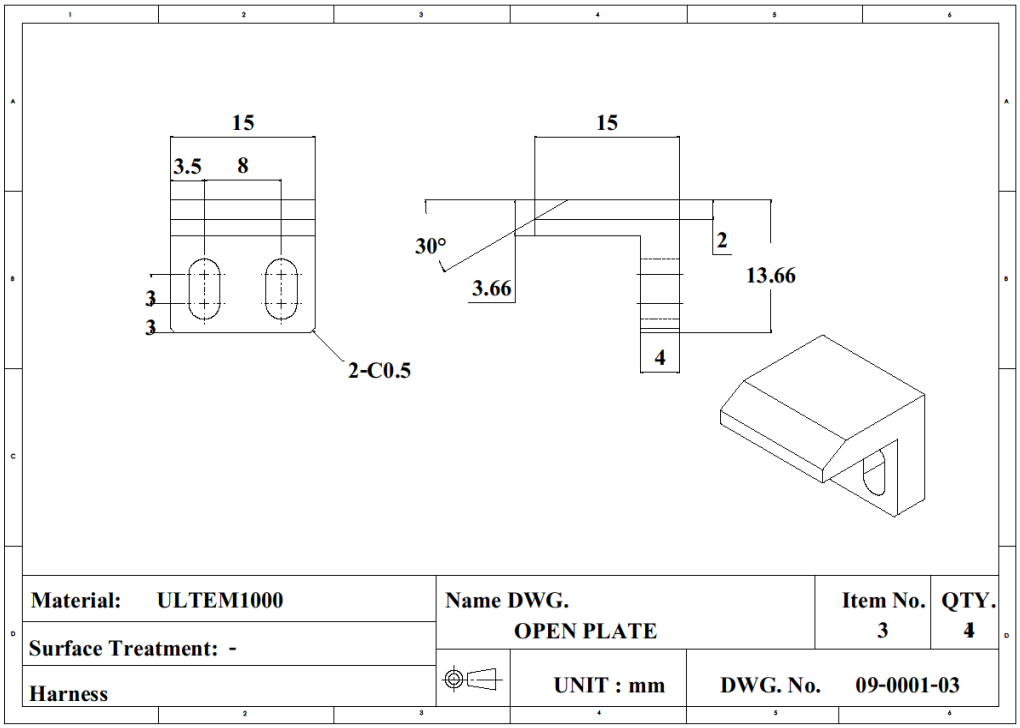
รูป ก.8 หมายเลขชิ้นส่วนของ ABSORBER



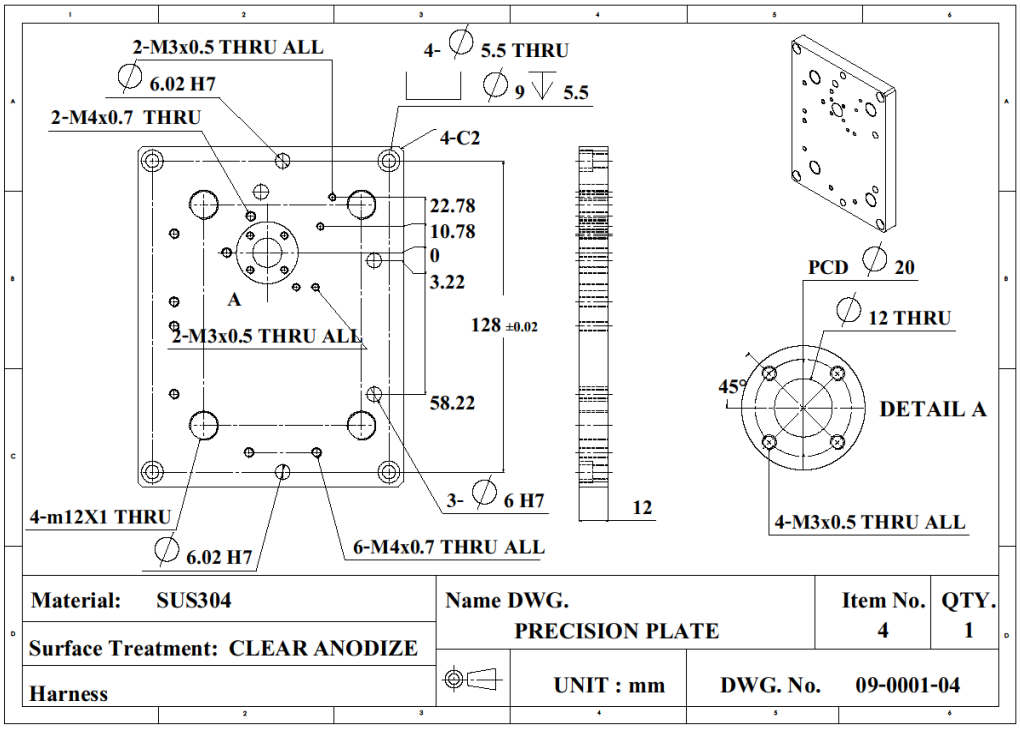
รูปที่ ก.9 รายละเอียดขนาดของ BASE PLATE



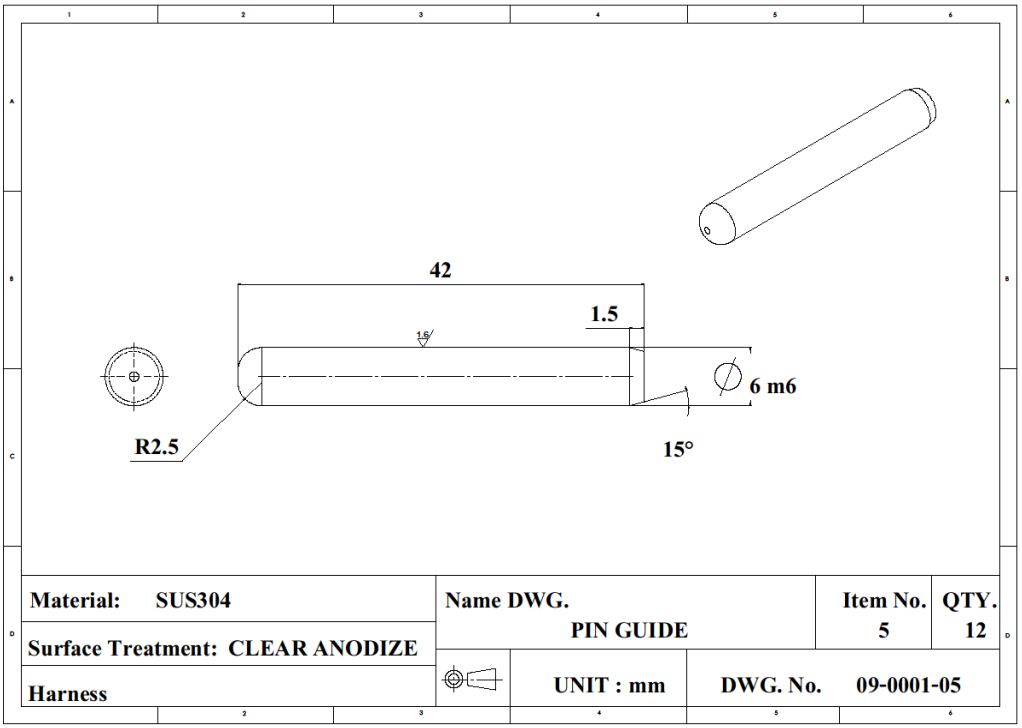
รูปที่ ก.10 รายละเอียดขนาดของ ROTATION PLATE



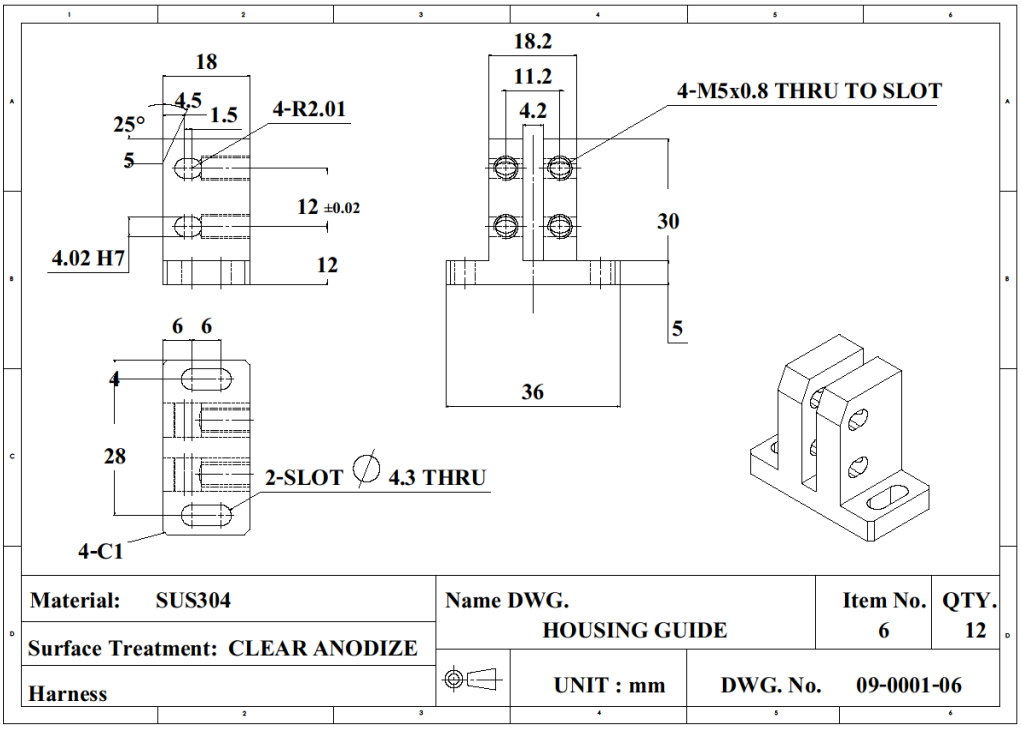
รูปที่ ก.11 รายละเอียดขนาดของ OPEN PLATE



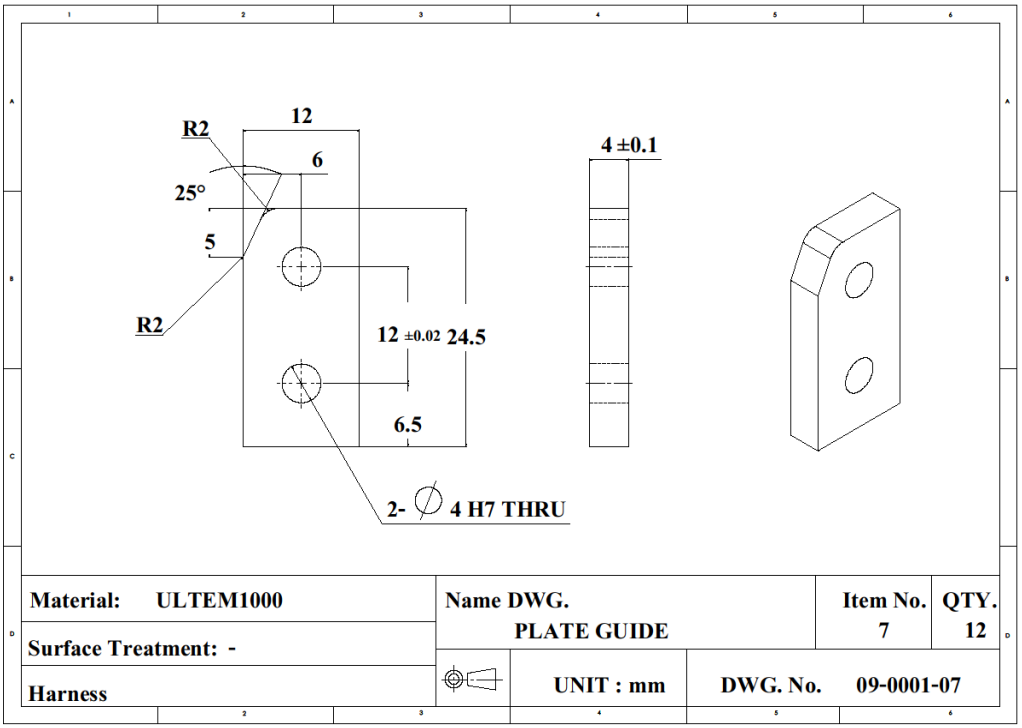
รูปที่ ก.12 รายละเอียดขนาดของ PRECISION PLATE



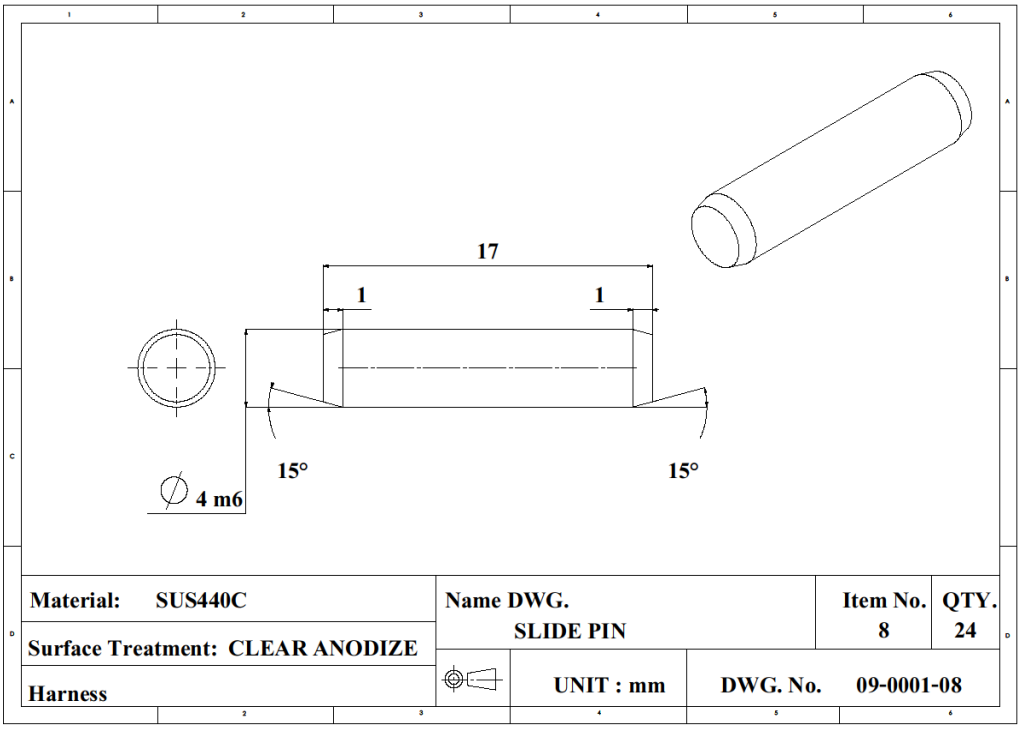
รูปที่ ก.13 รายละเอียดขนาดของ PIN GUIDE



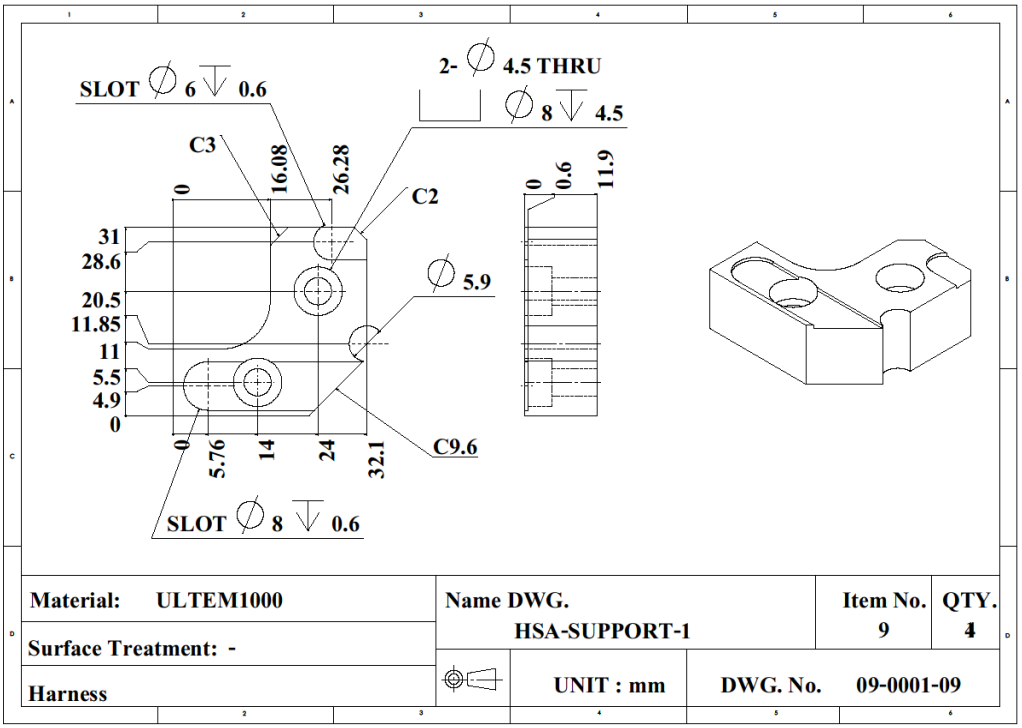
รูปที่ ก.14 รายละเอียดขนาดของ HOUSING GUIDE



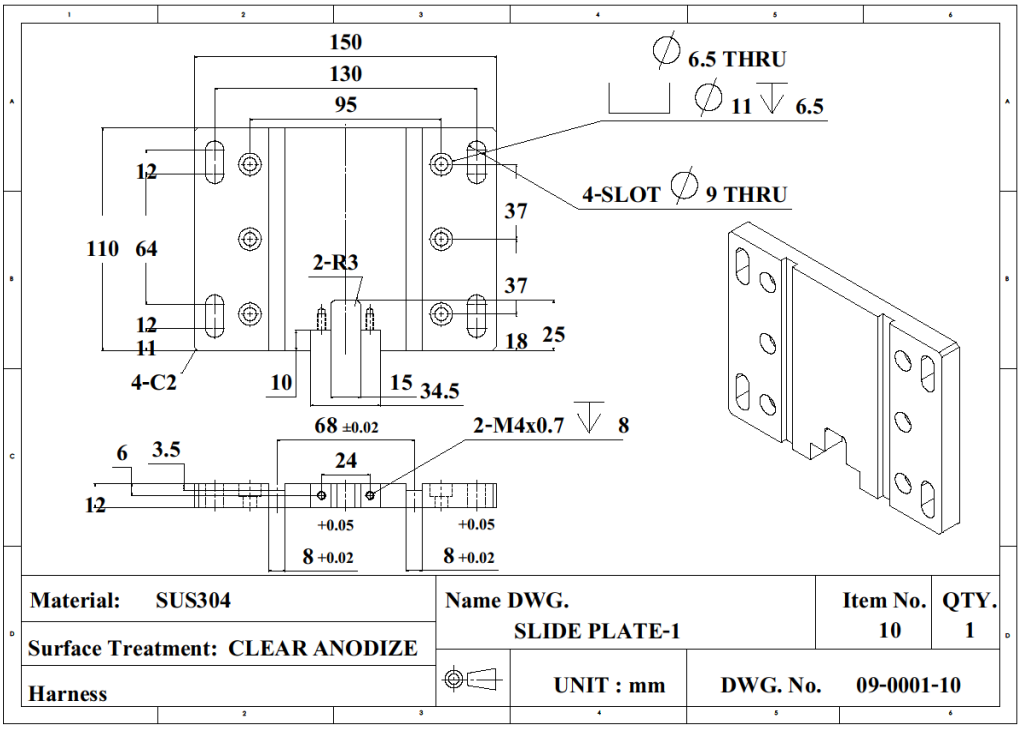
รูปที่ ก.15 รายละเอียดขนาดของ PLATE GUIDE



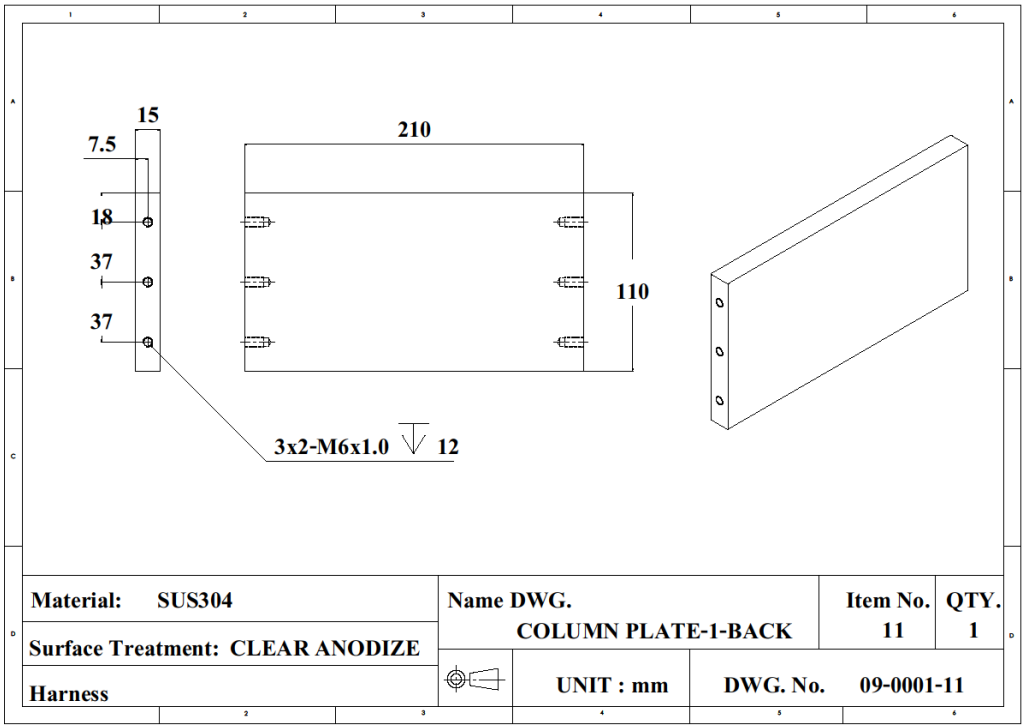
รูปที่ ก.16 รายละเอียดขนาดของ SLIDE PIN



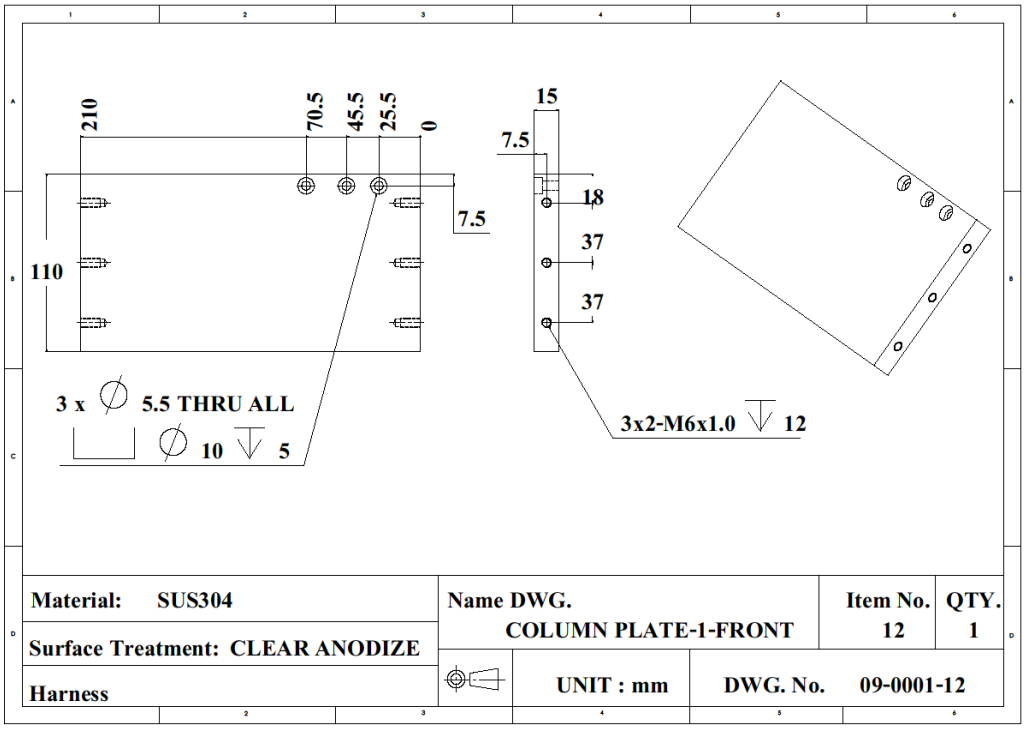
รูปที่ ก.17 รายละเอียดขนาดของ HSA SUPPORT-1



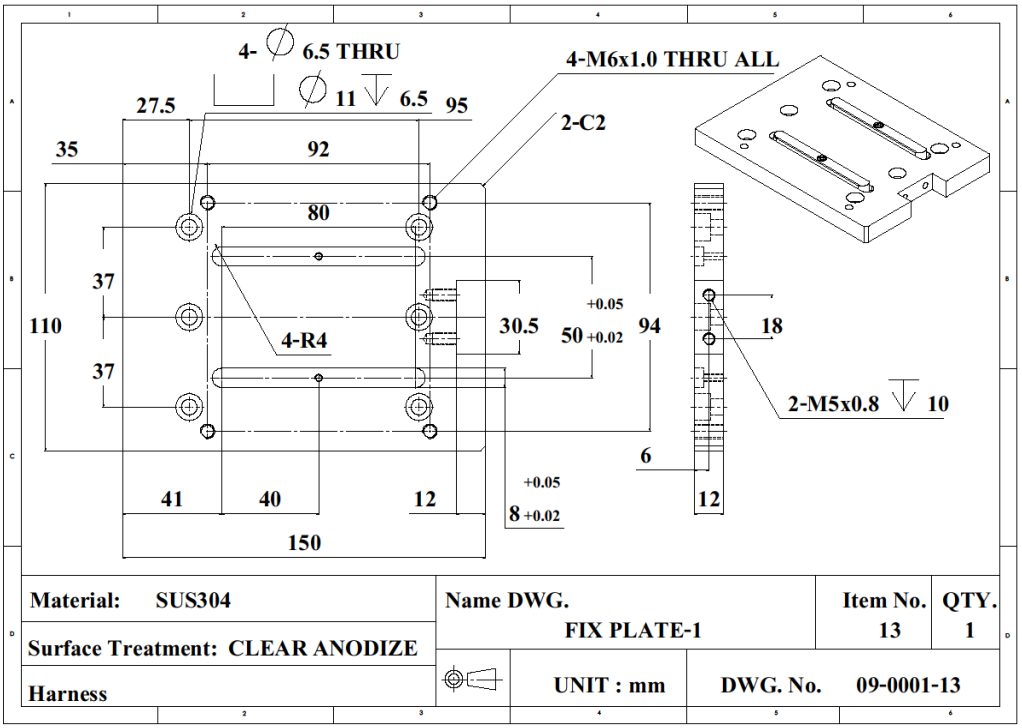
รูปที่ ก.18 รายละเอียดขนาดของ SLIDE PLATE-1



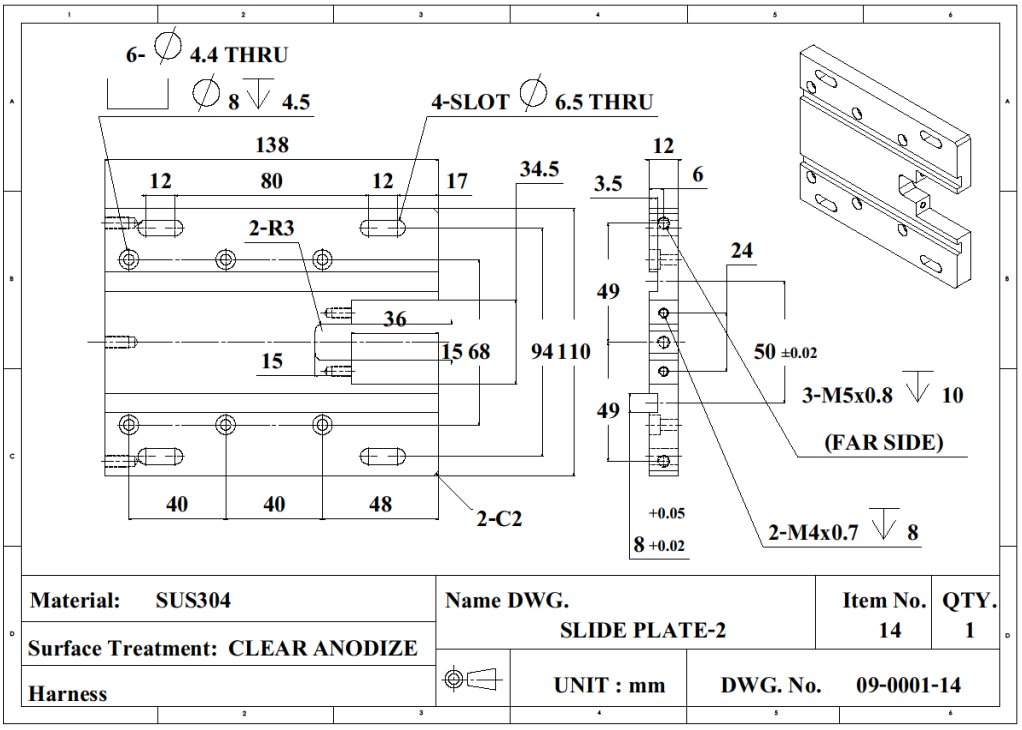
รูปที่ ก.19 รายละเอียดขนาดของ COLUMN PLATE-1-BACK



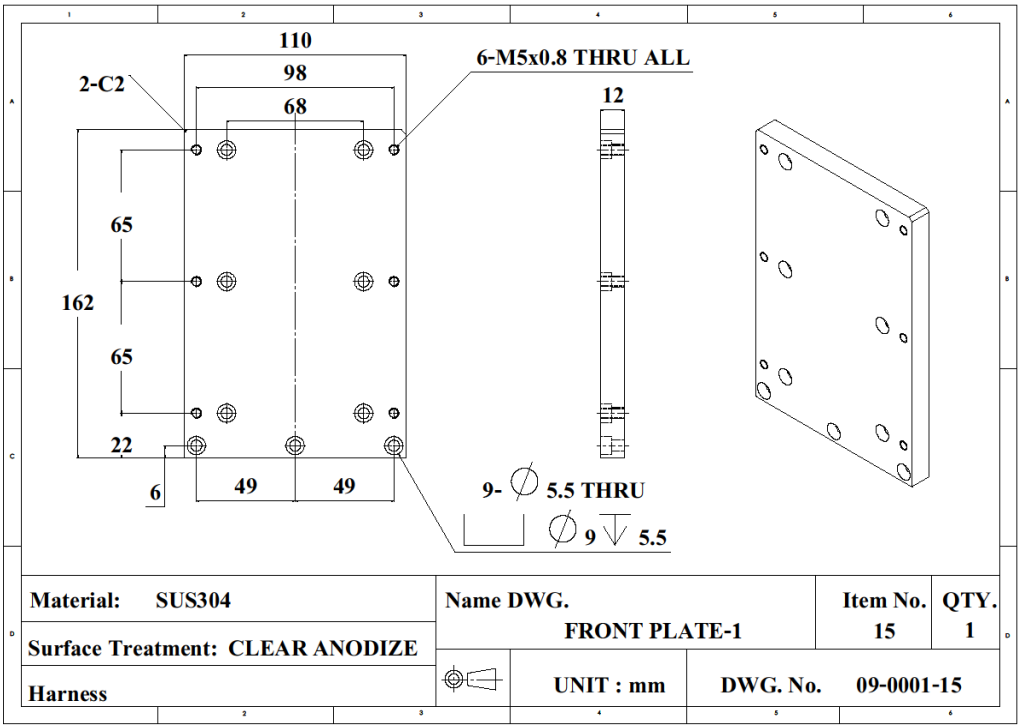
รูปที่ ก.20 รายละเอียดขนาดของ COLUMN PLATE-1-FRONT



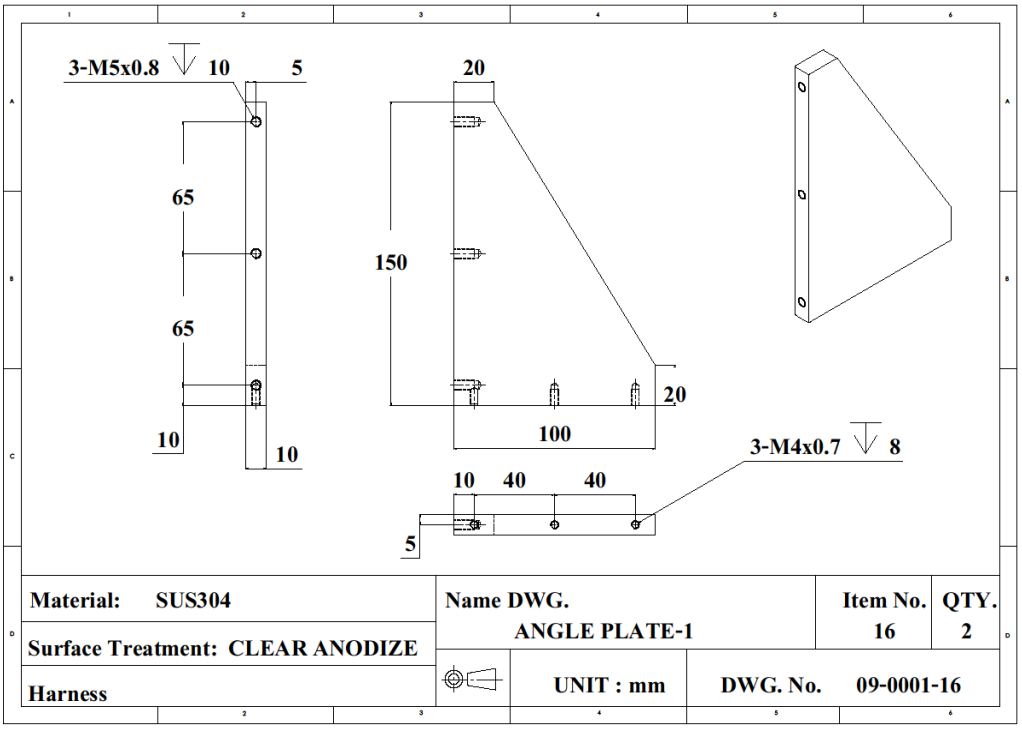
รูปที่ ก.21 รายละเอียดขนาดของ FIX PLATE-1



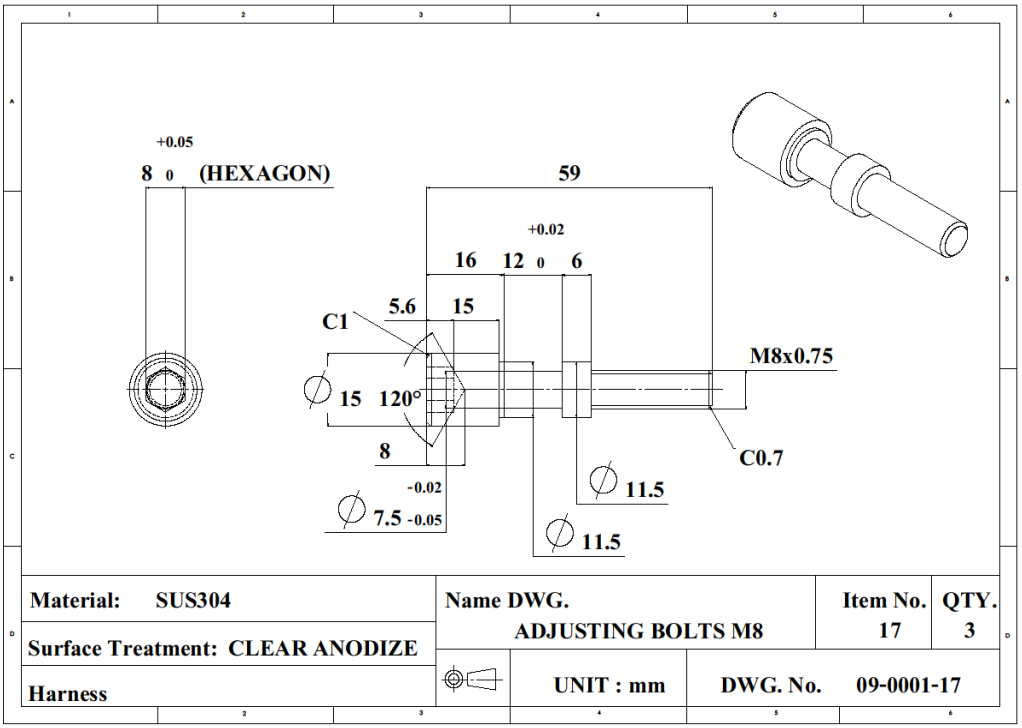
รูปที่ ก.22 รายละเอียดขนาดของ SLIDE PLATE-2



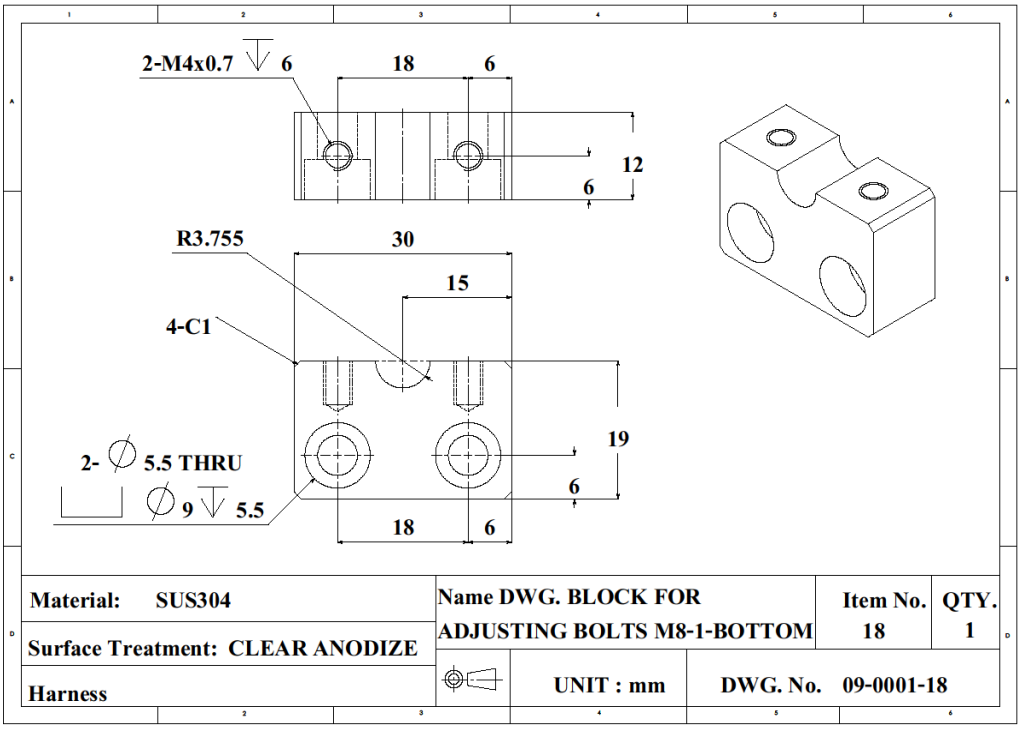
รูปที่ ก.23 รายละเอียดขนาดของ FRONT PLATE-1



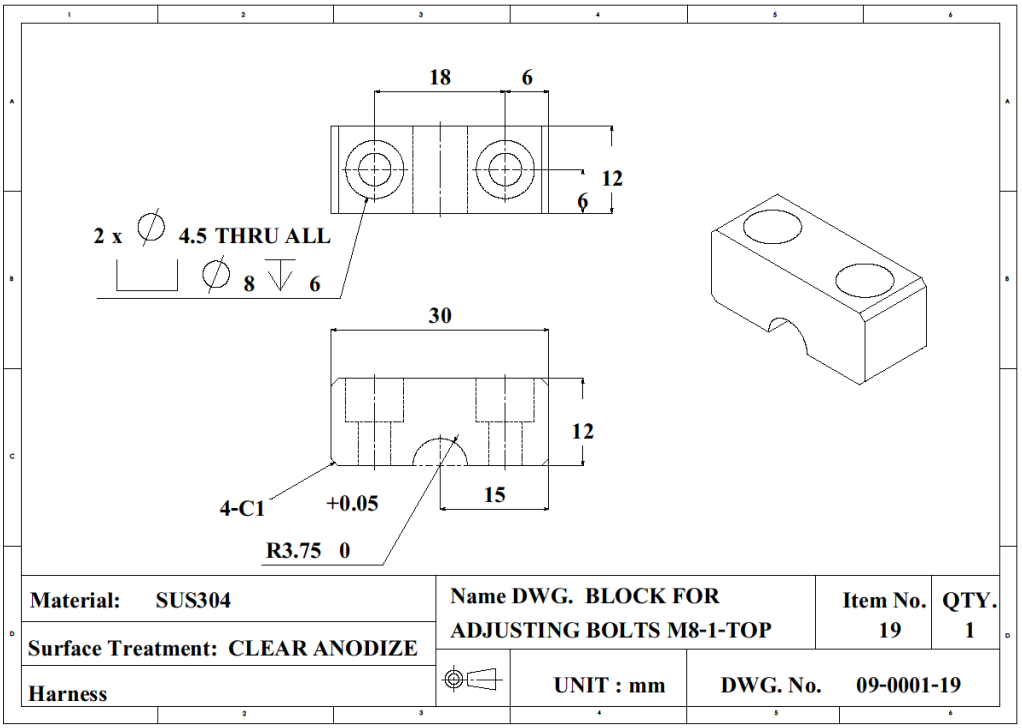
รูปที่ ก.24 รายละเอียดขนาดของ ANGLE PLATE-1



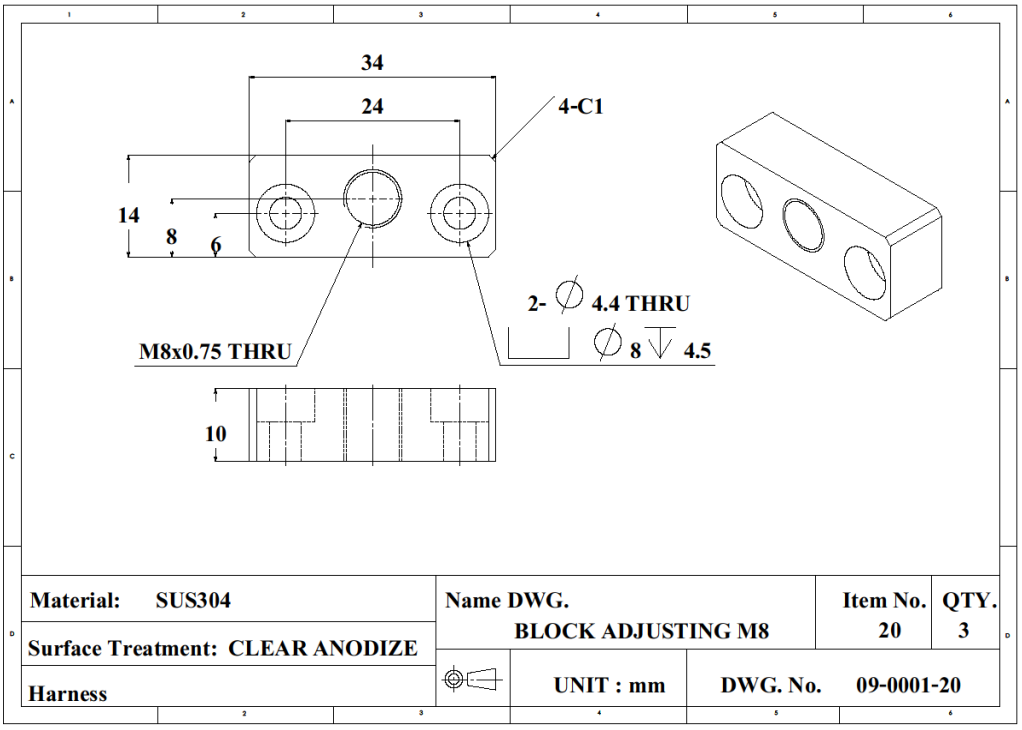
รูปที่ ก.25 รายละเอียดขนาดของ ADJUSTING BOLTS M8



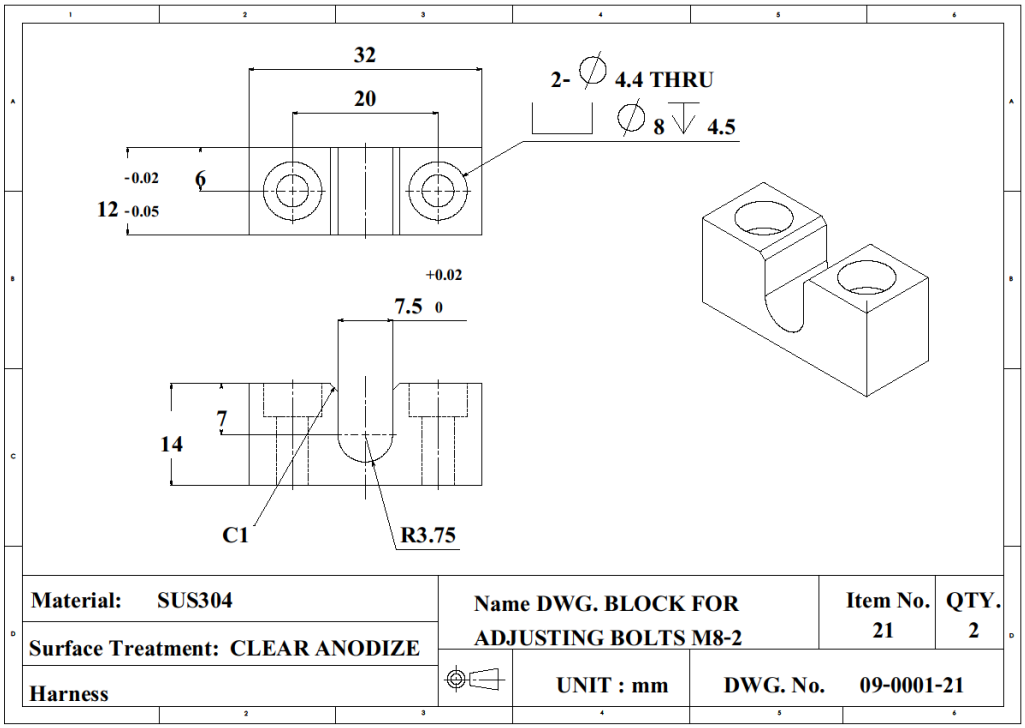
รูปที่ ก.26 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M8-1-BOTTOM



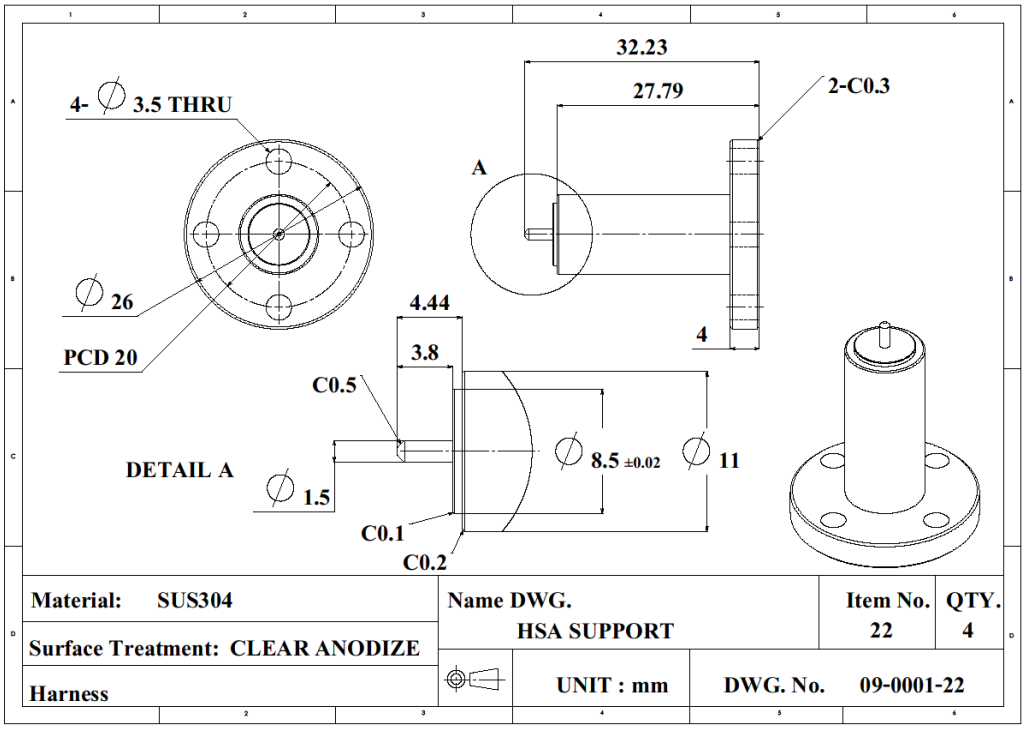
รูปที่ ก.27 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M8-1-TOP



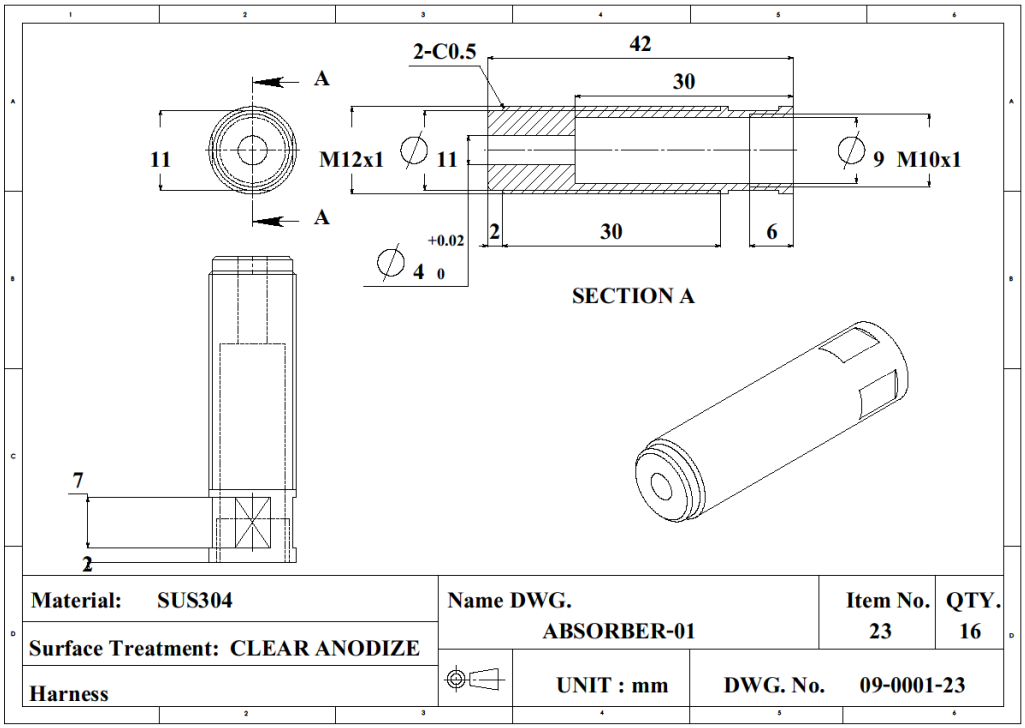
รูปที่ ก.28 รายละเอียดขนาดของ BLOCK ADJUSTING M8



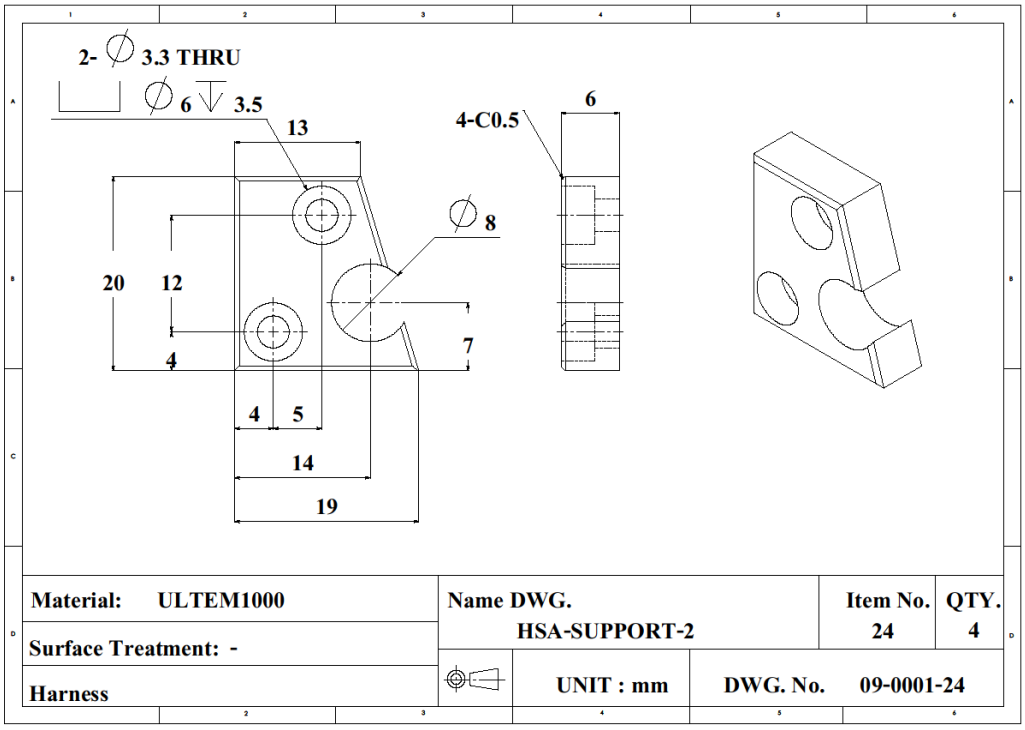
รูปที่ ก.29 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M8-2



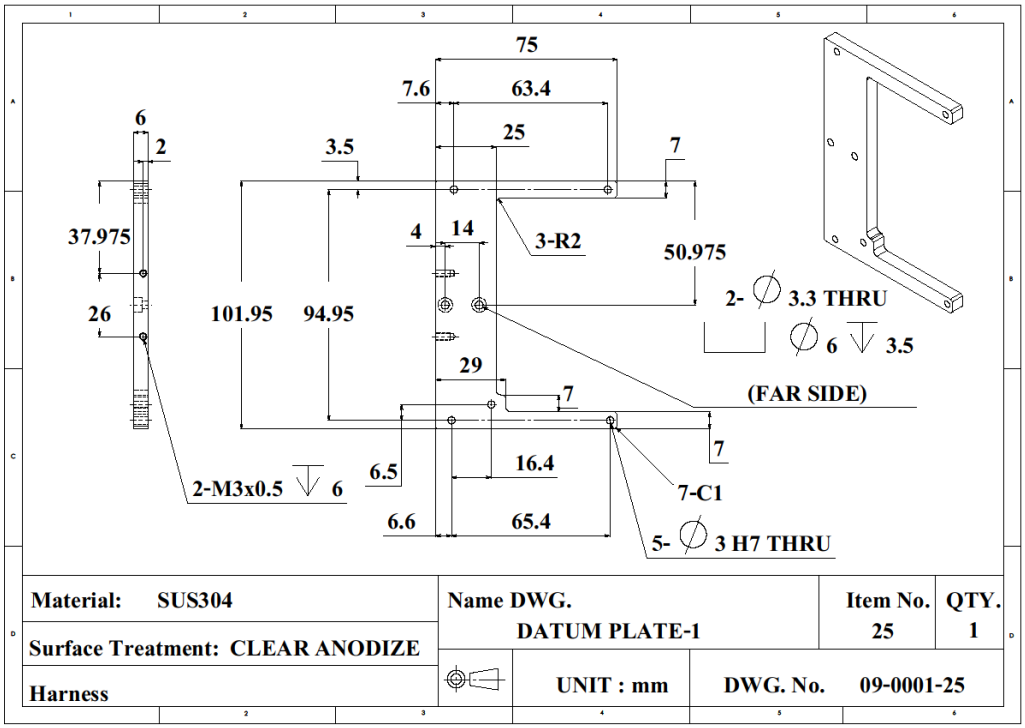
รูปที่ ก.30 รายละเอียดขนาดของ HSA SUPPORT



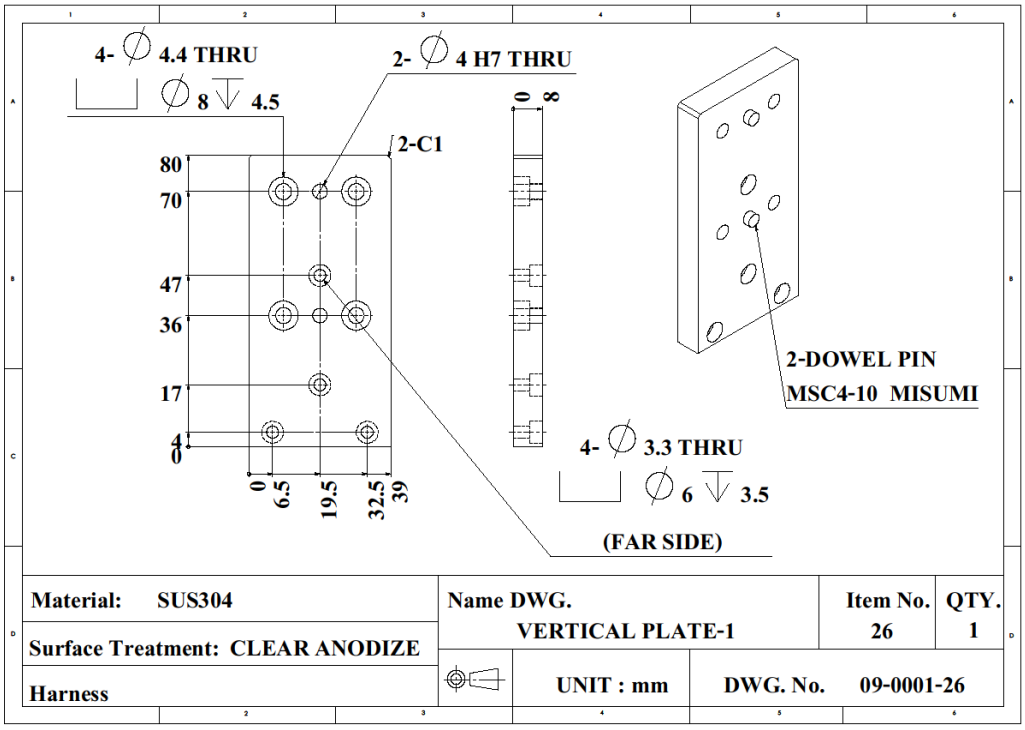
รูปที่ ก.31 รายละเอียดขนาดของ ABSORBER-01



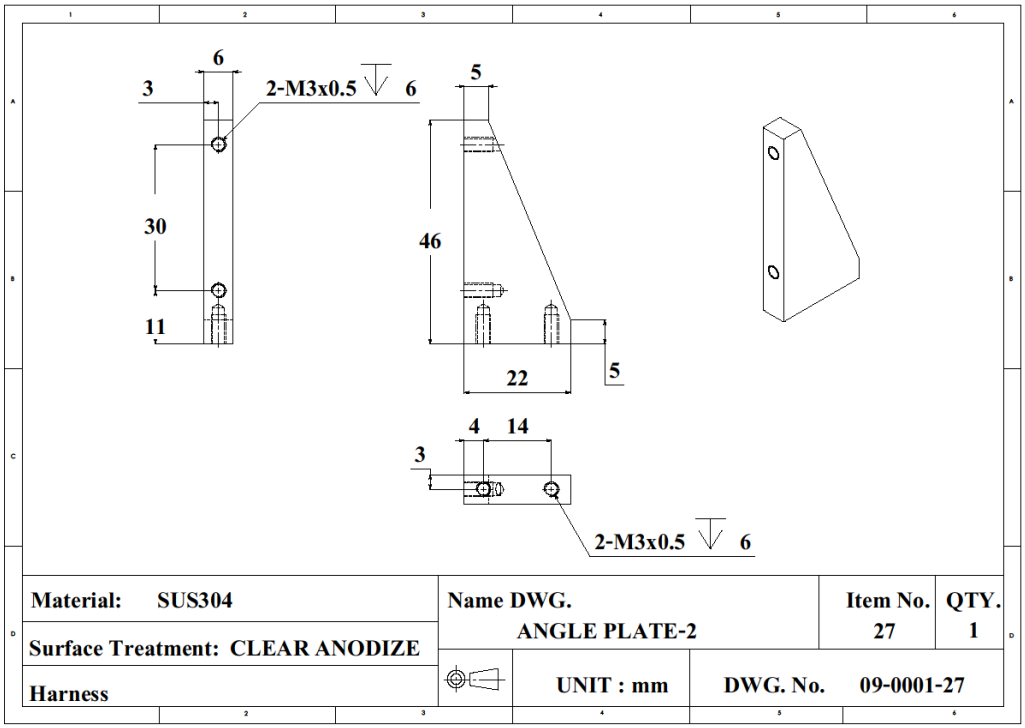
รูปที่ ก.32 รายละเอียดขนาดของ HSA SUPPORT-2



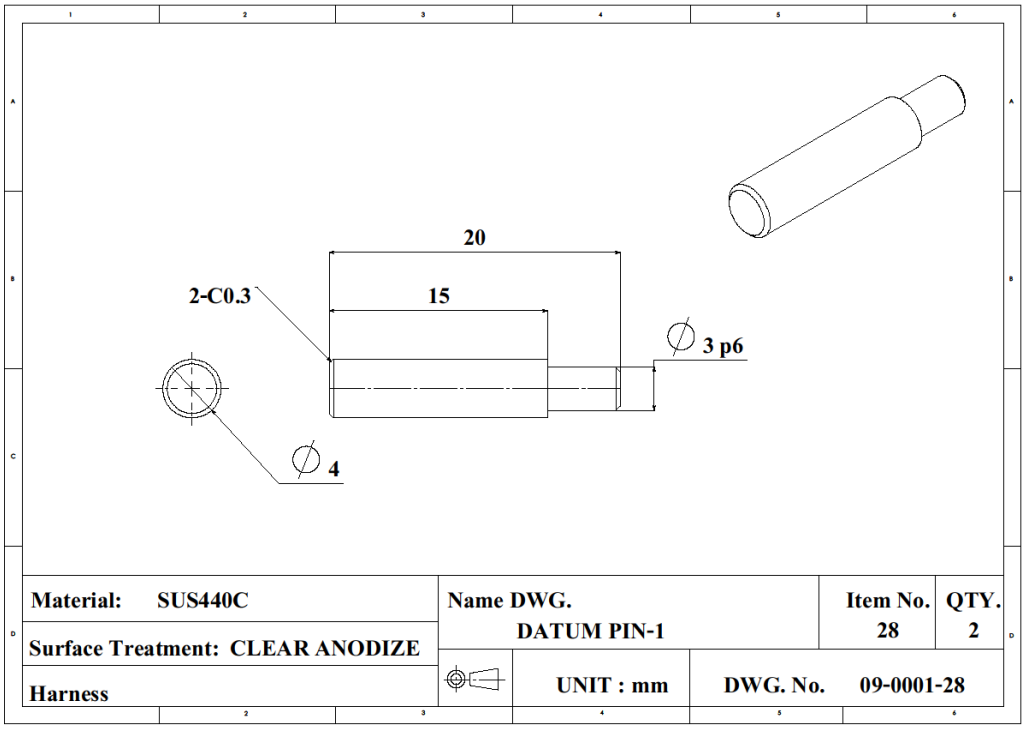
รูปที่ ก.33 รายละเอียดขนาดของ DATUM PLATE-1



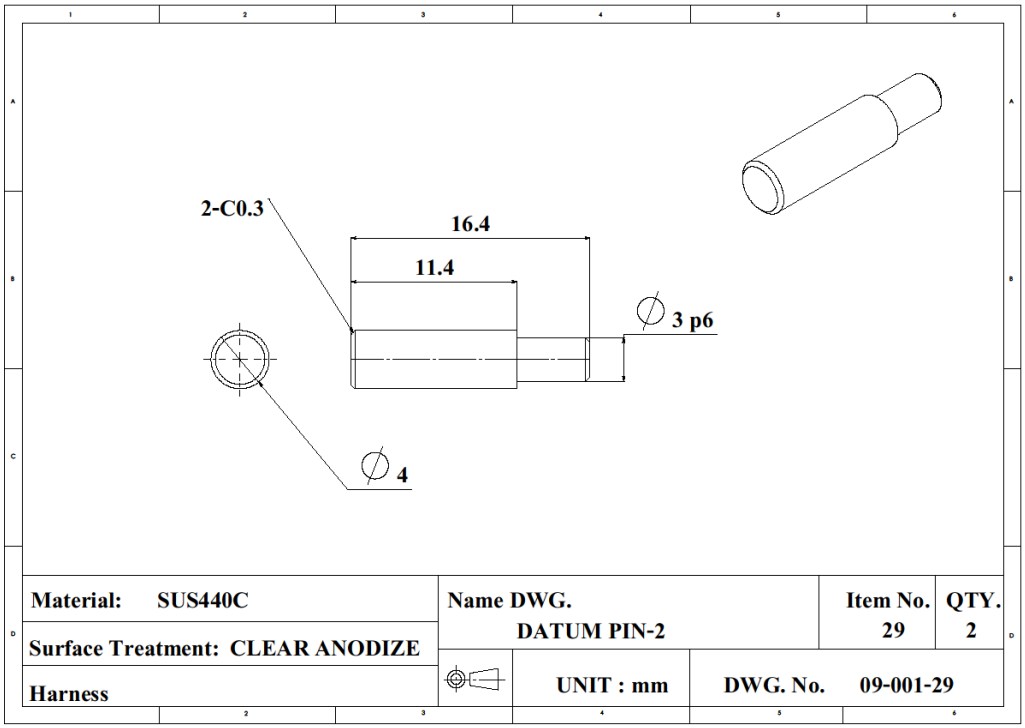
รูปที่ ก.34 รายละเอียดขนาดของ VERTICAL PLATE-1



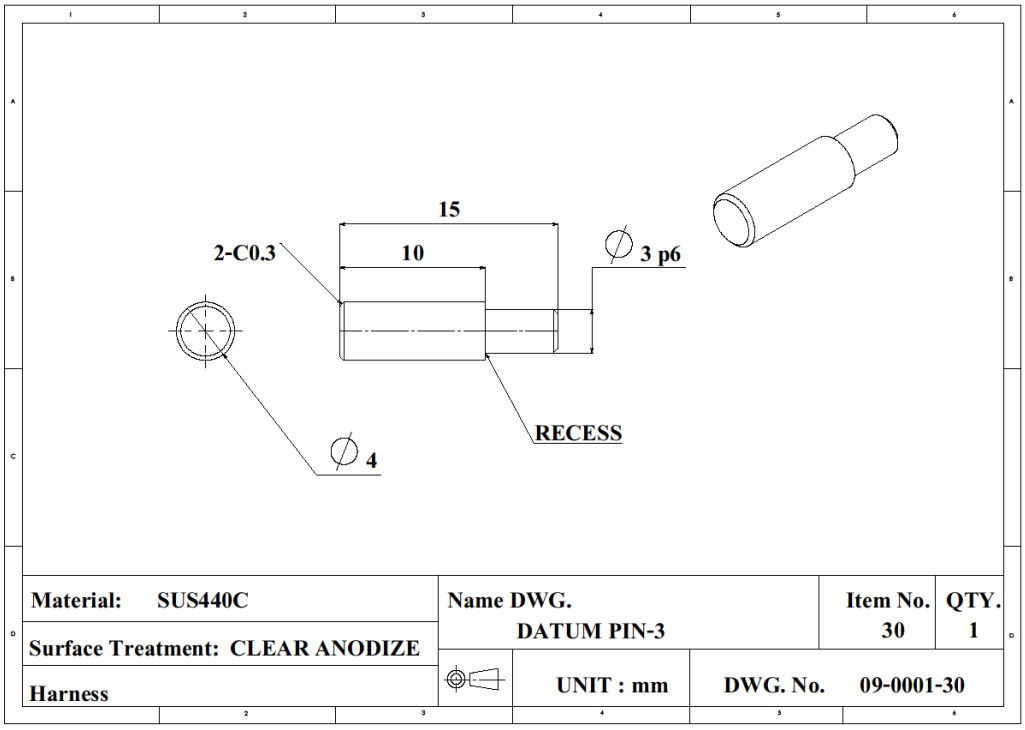
รูปที่ ก.35 รายละเอียดขนาดของ ANGLE PLATE-2



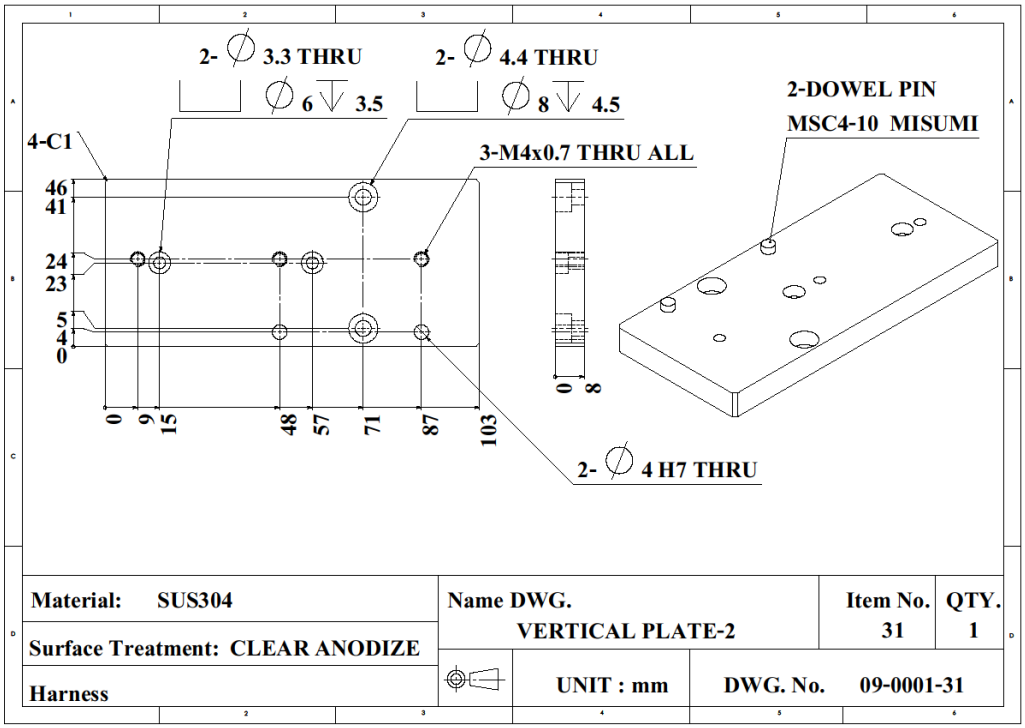
รูปที่ ก.36 รายละเอียดขนาดของ DATUM PIN-1



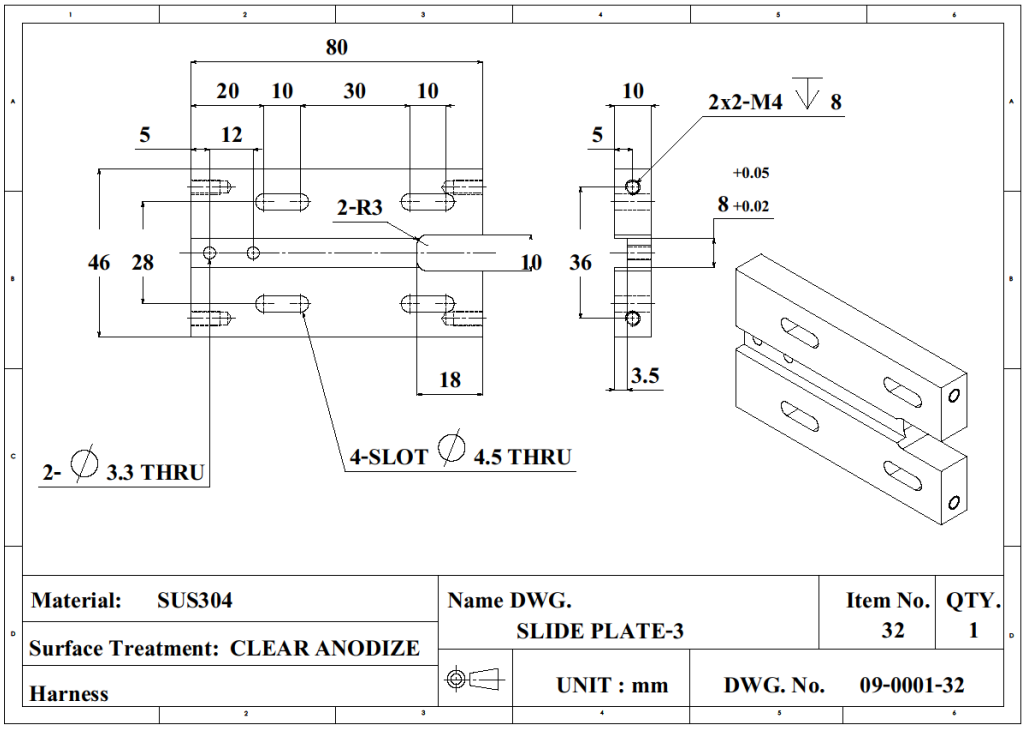
รูปที่ ก.37 รายละเอียดขนาดของ DATUM PIN-2



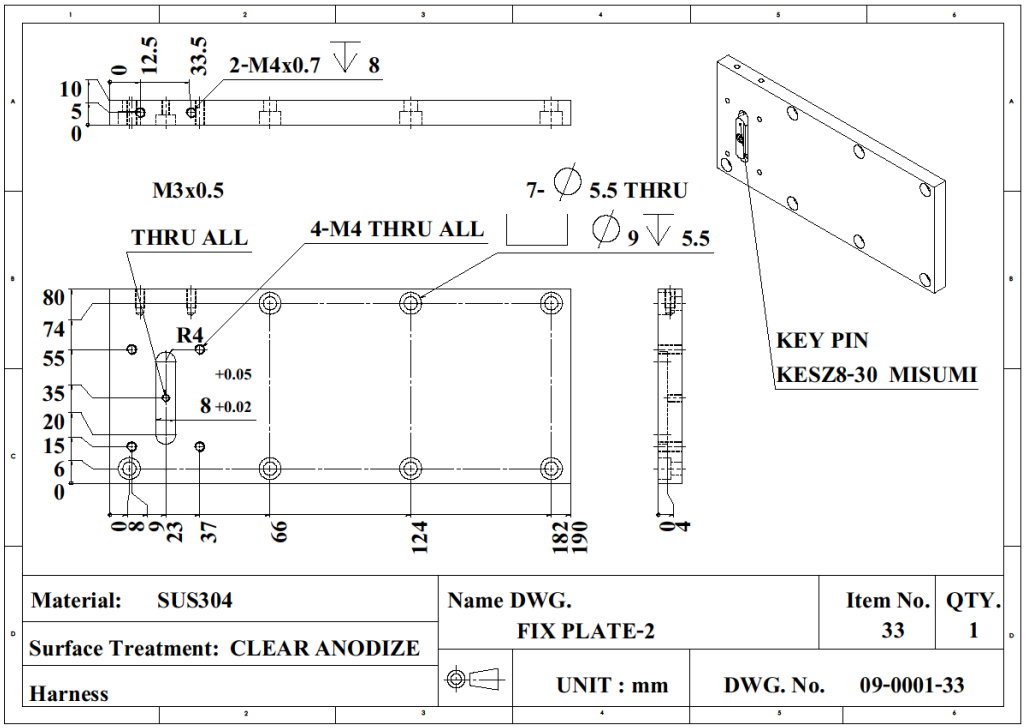
รูปที่ ก.38 รายละเอียดขนาดของ DATUM PIN-3



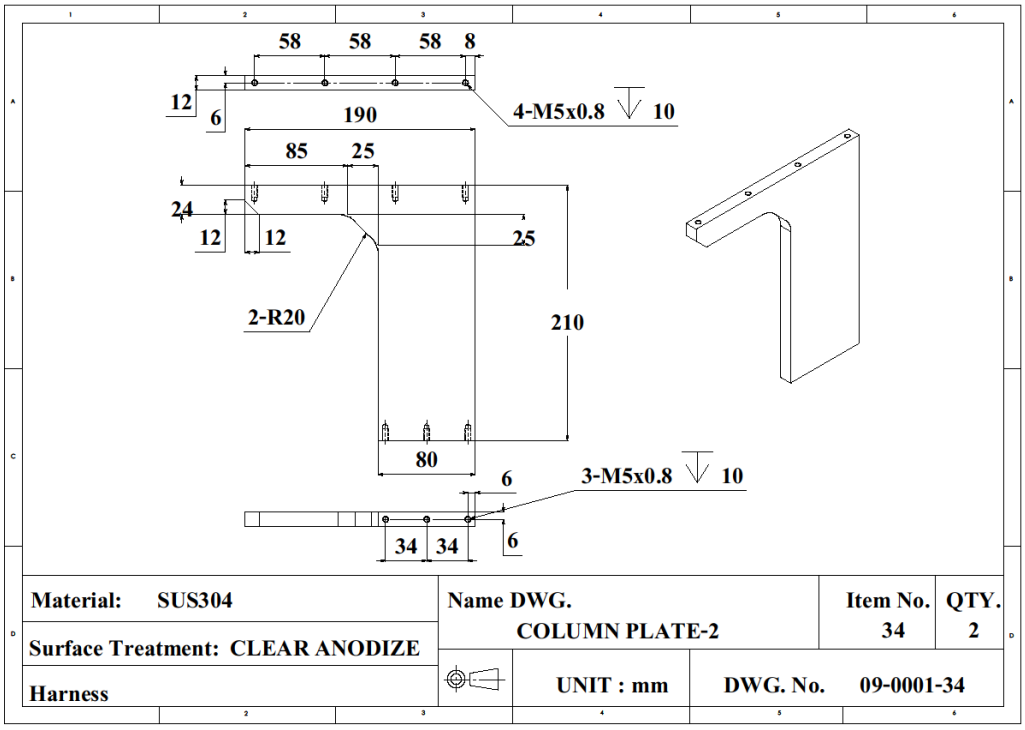
รูปที่ ก.39 รายละเอียดขนาดของ VERTICAL PLATE-2



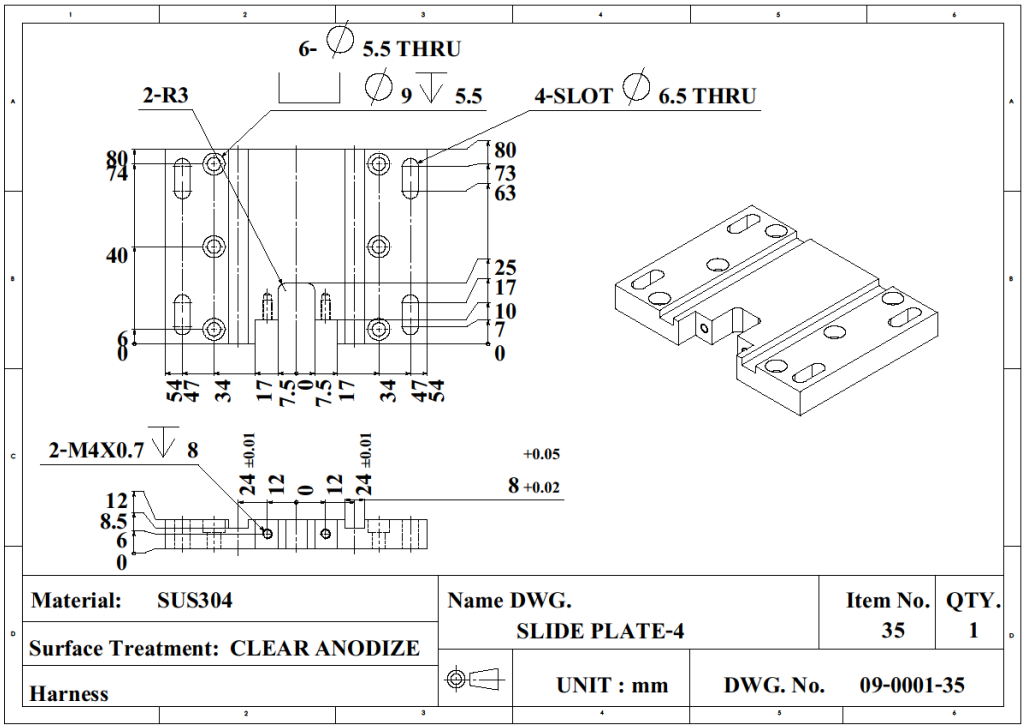
รูปที่ ก.40 รายละเอียดขนาดของ SLIDE PLATE-3



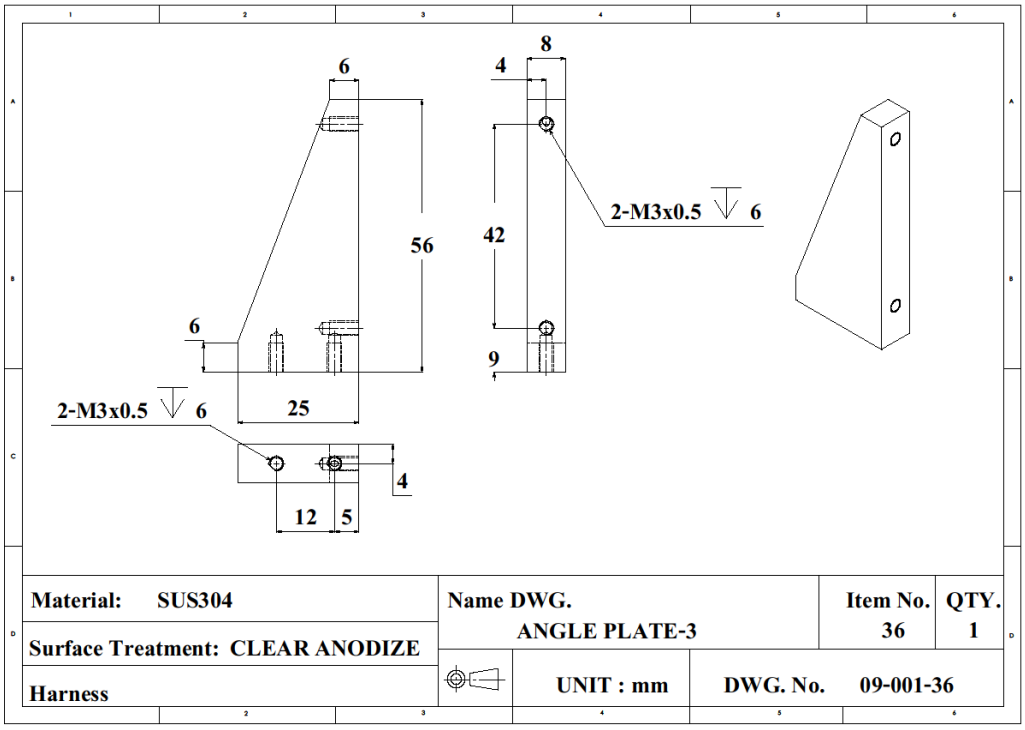
รูปที่ ก.41 รายละเอียดขนาดของ FIX PLATE-2



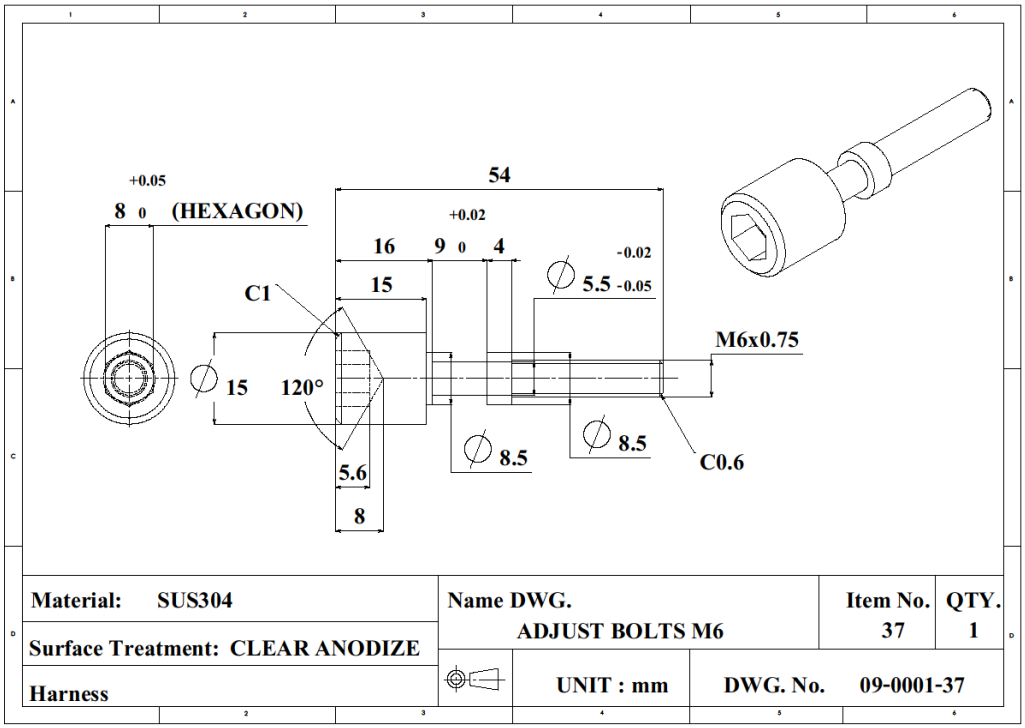
รูปที่ ก.42 รายละเอียดขนาดของ COLUMN PLATE-2



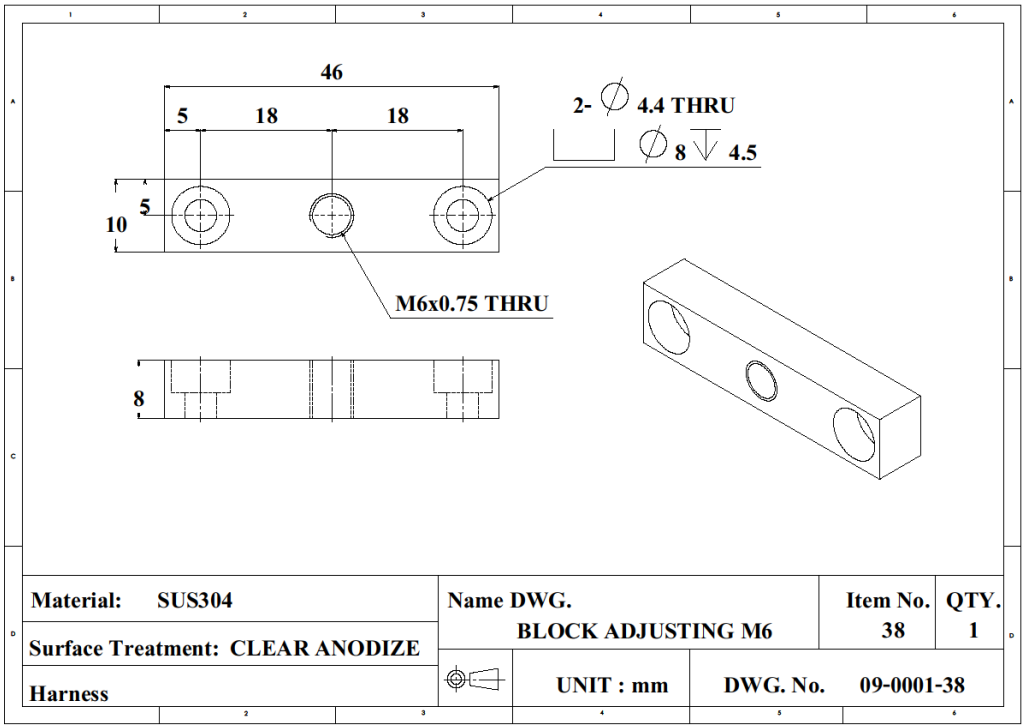
รูปที่ ก.43 รายละเอียดขนาดของ SLIDE PLATE-4



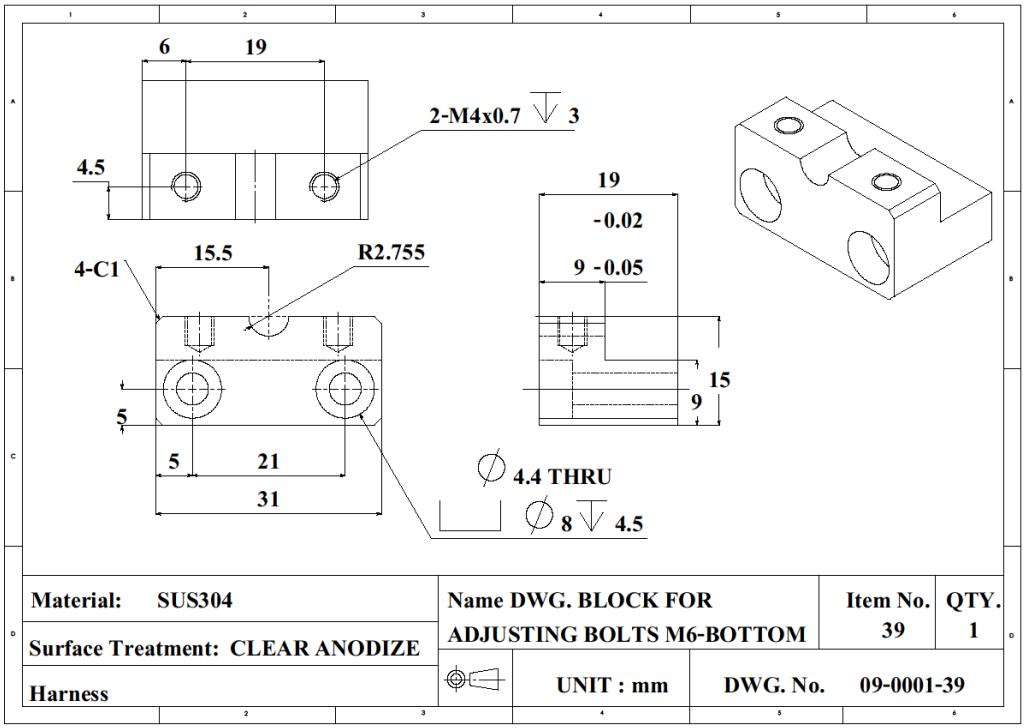
รูปที่ ก.44 รายละเอียดขนาดของ ANGLE PLATE-3



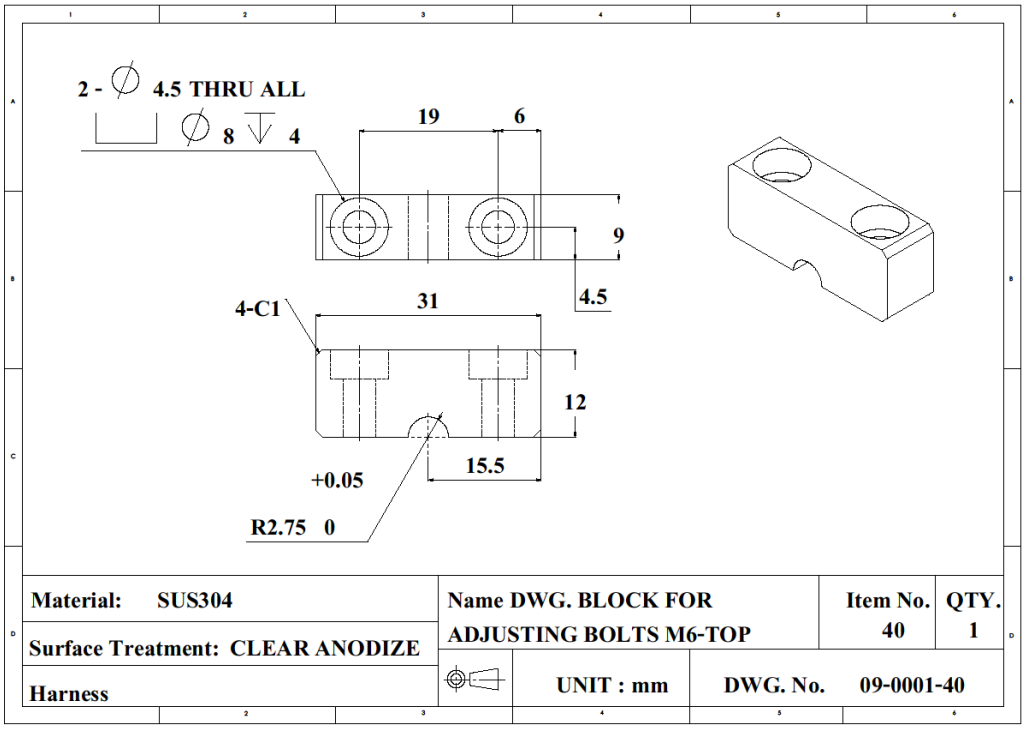
รูปที่ ก.45 รายละเอียดขนาดของ ADJUST BOLTS M6



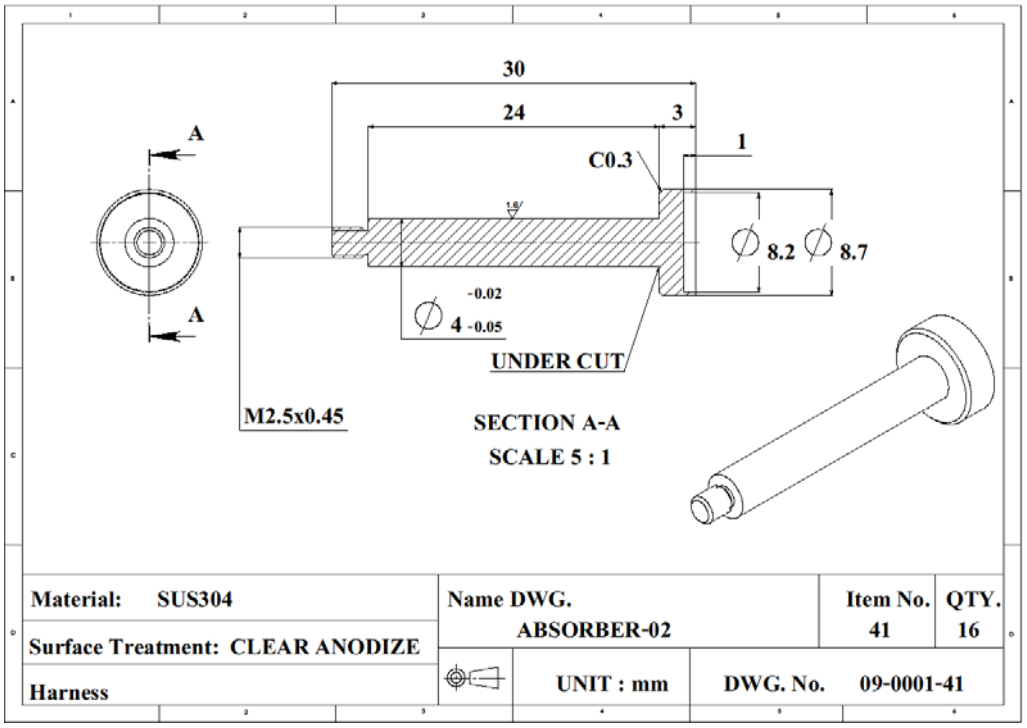
รูปที่ ก.46 รายละเอียดขนาดของ BLOCK ADJUSTING M6



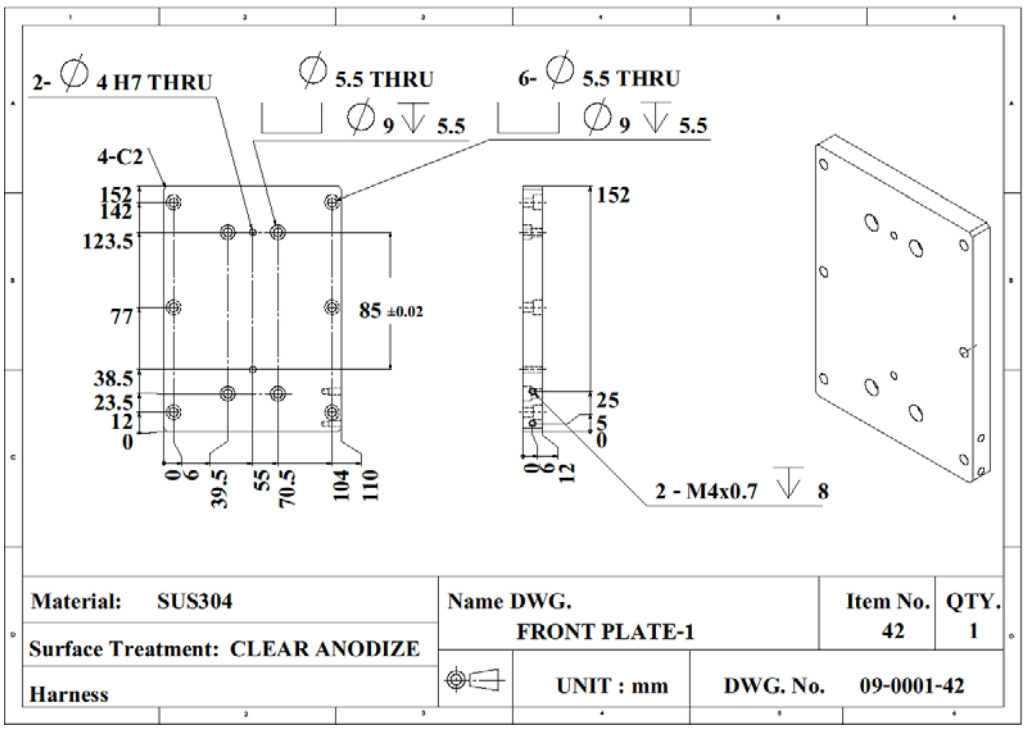
รูปที่ ก.47 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M6-BOTTOM



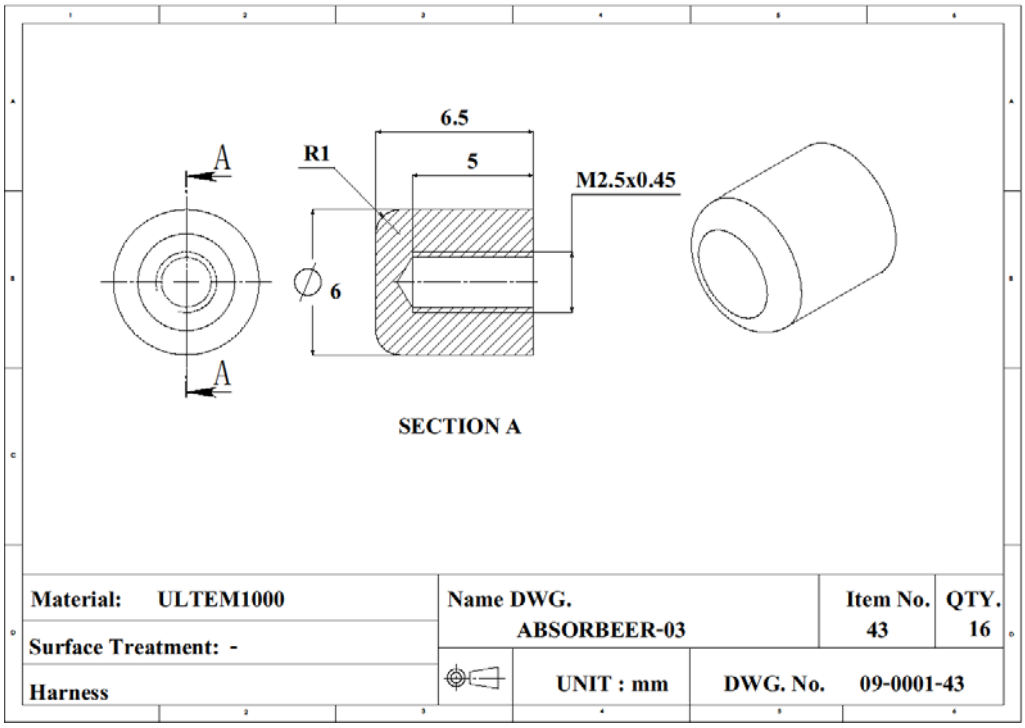
รูปที่ ก.48 รายละเอียดขนาดของ BLOCK FOR ADJUSTING BOLTS M6-TOP



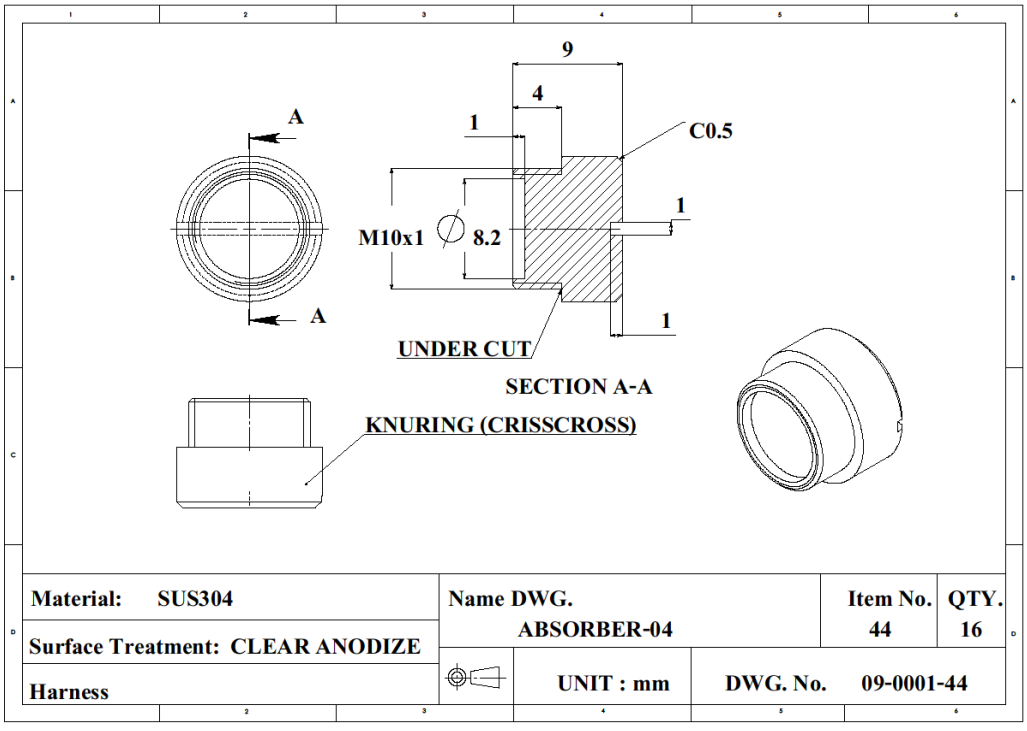
รูปที่ ก.49 รายละเอียดขนาดของ ABSORBER-02



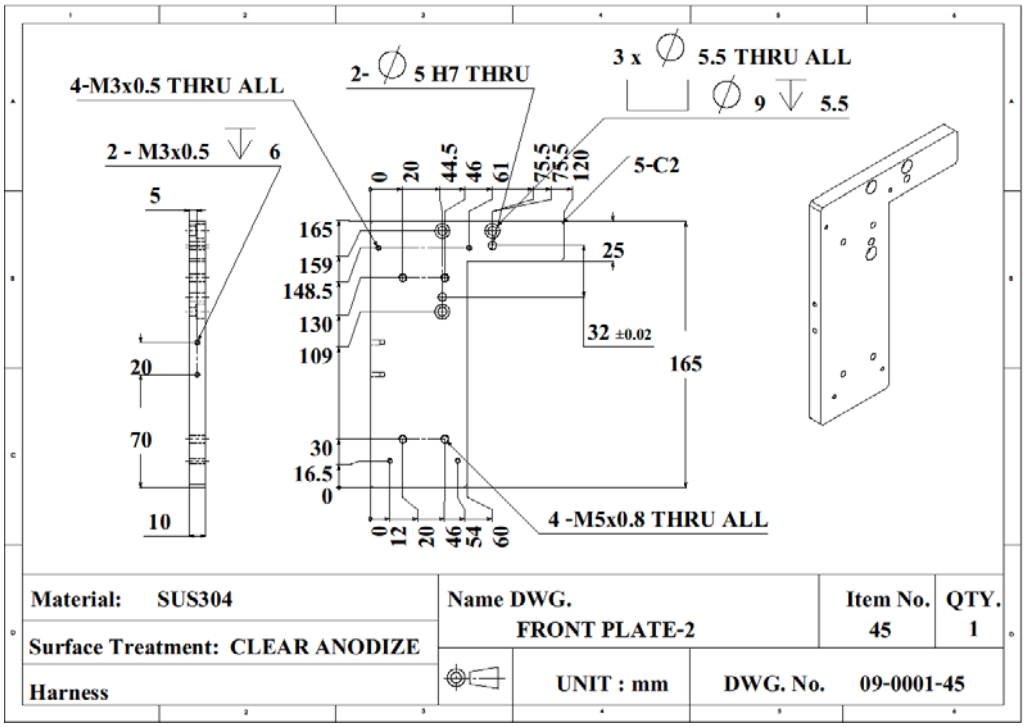
รูปที่ ก.50 รายละเอียดขนาดของ FRONT PLATE-1



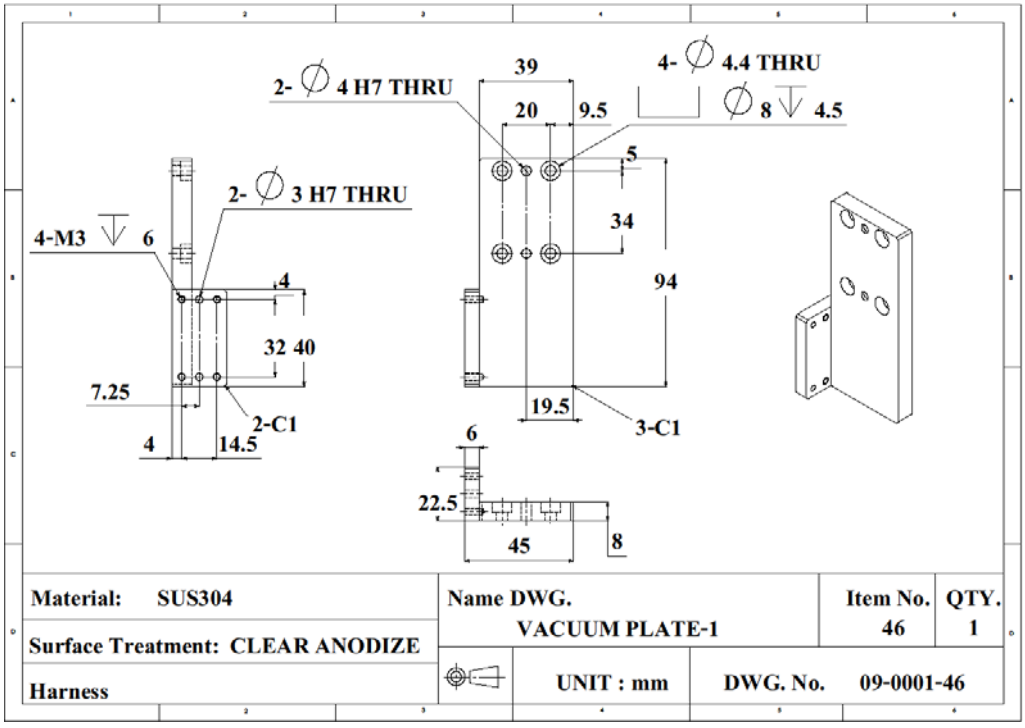
รูปที่ ก.51 รายละเอียดขนาดของ ABSORBER-03



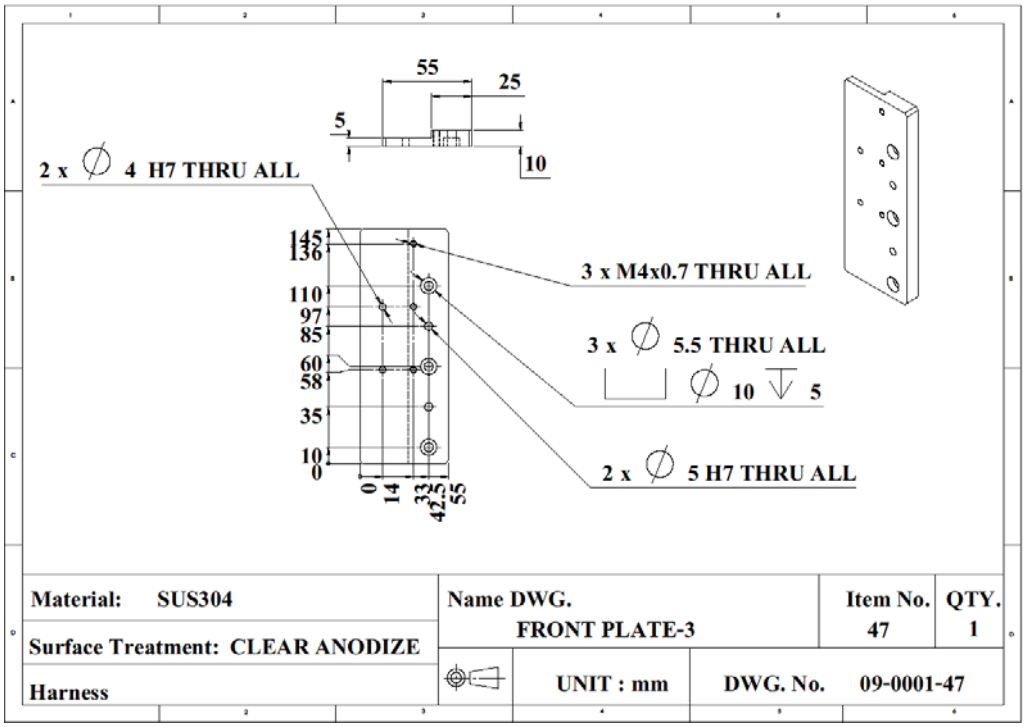
รูปที่ ก.52 รายละเอียดขนาดของ ABSORBER-04



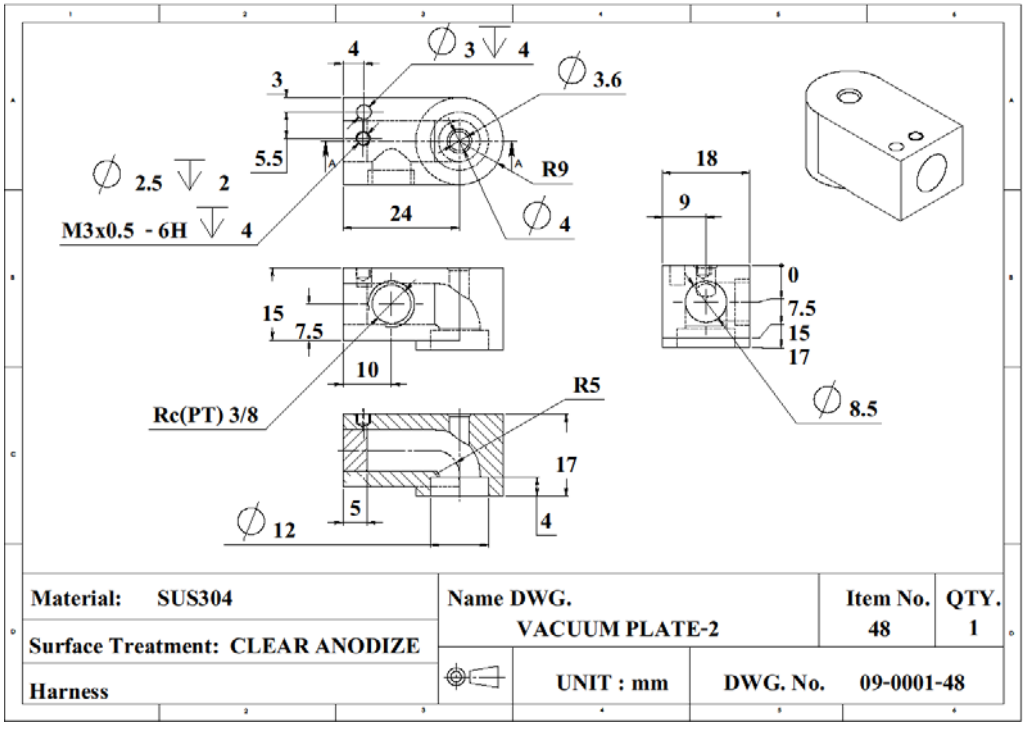
รูปที่ ก.53 รายละเอียดขนาดของ FRONT PLATE-2



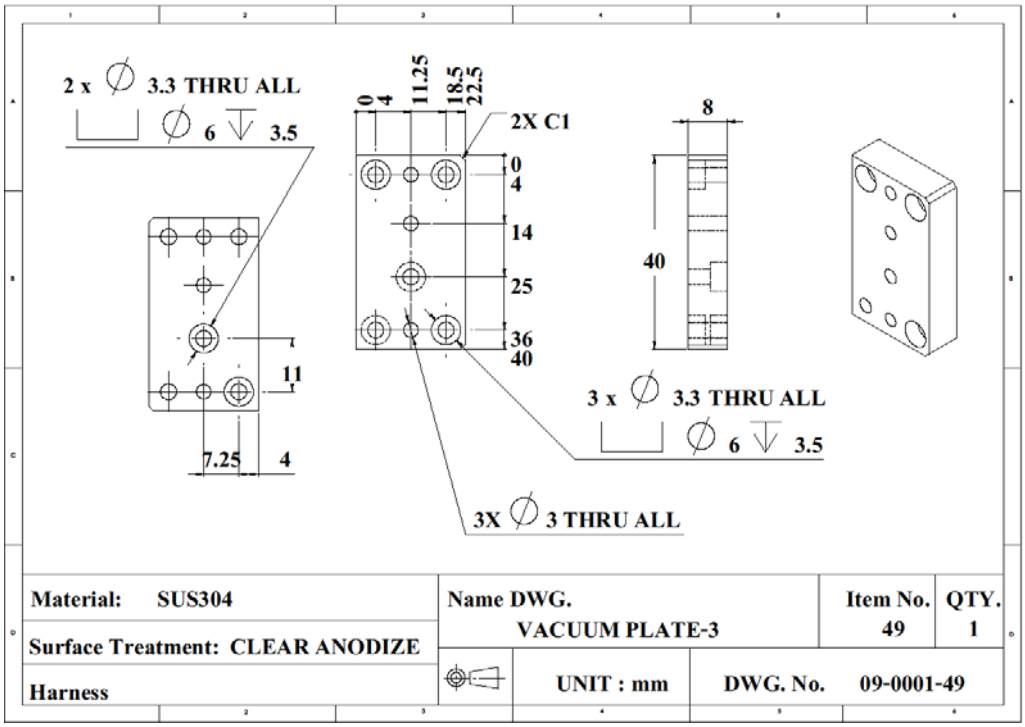
รูปที่ ก.54 รายละเอียดขนาดของ VACUUM PLATE-1



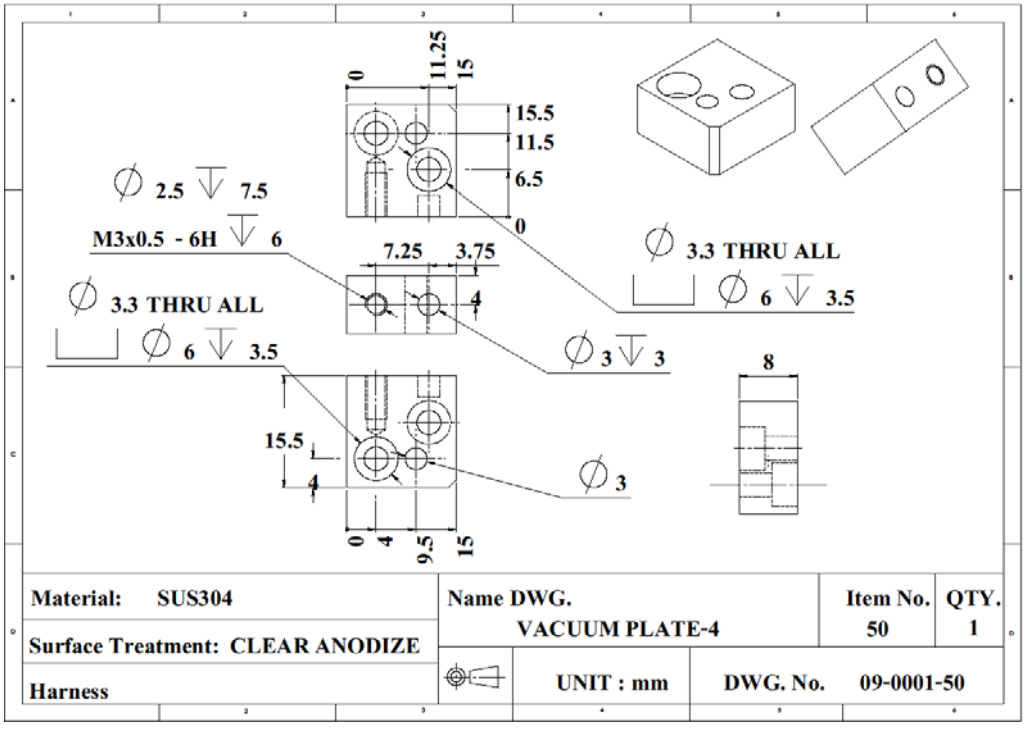
รูปที่ ก.55 รายละเอียดขนาดของ FRONT PLATE-3



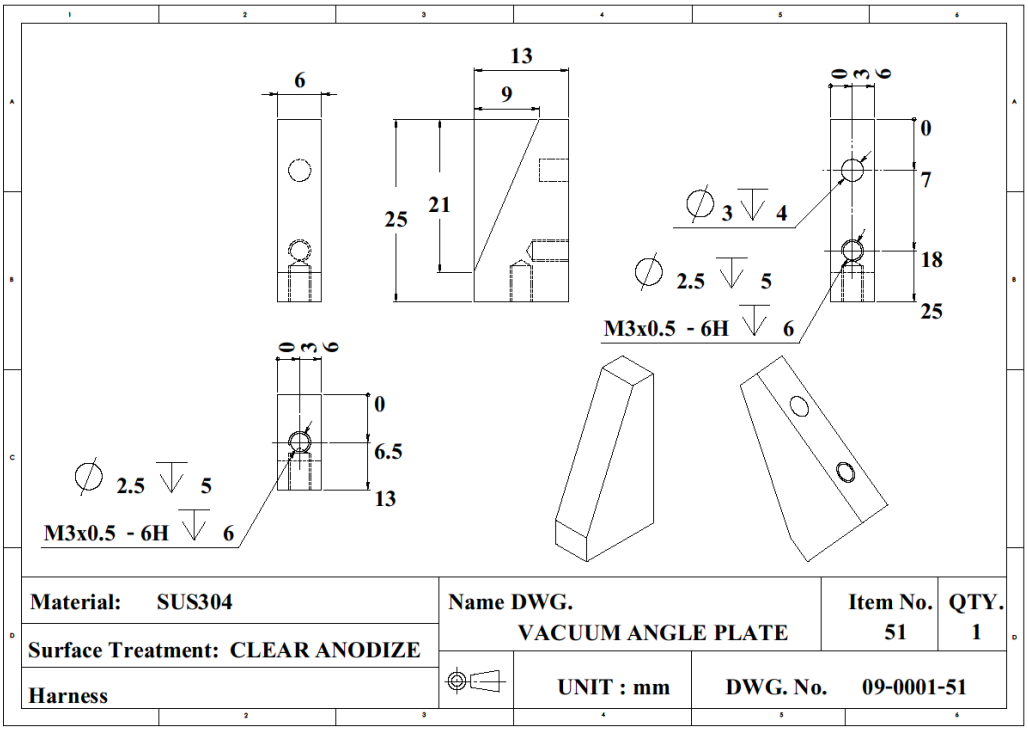
รูปที่ ก.56 รายละเอียดขนาดของ VACUUM PLATE-2



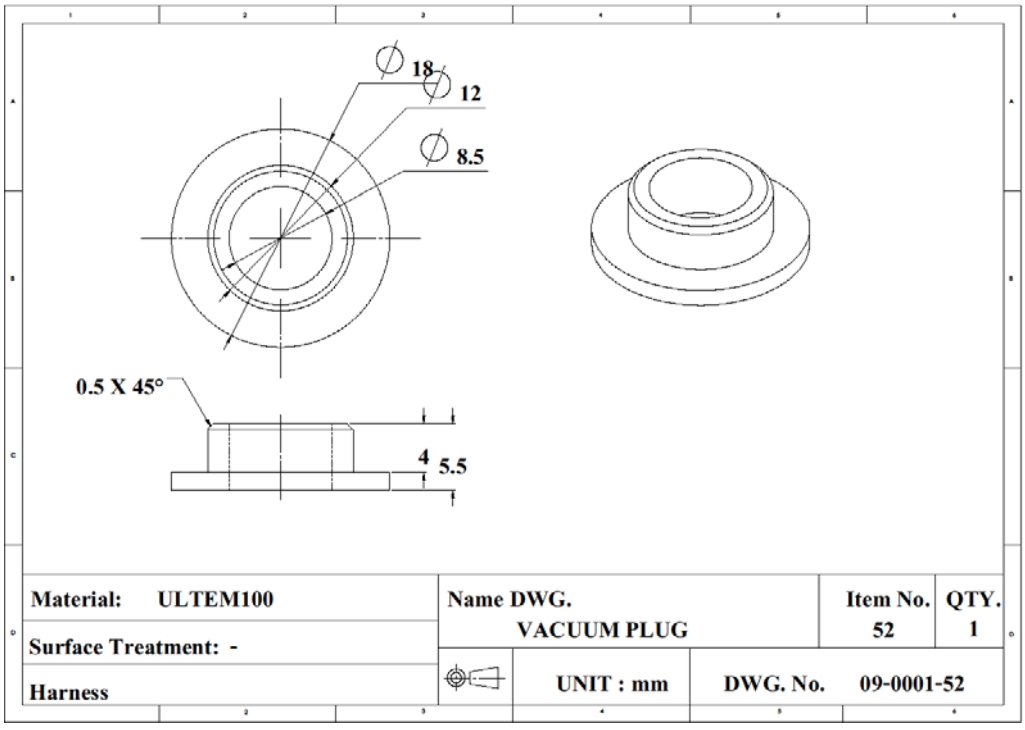
รูปที่ ก.57 รายละเอียดขนาดของ VACUUM PLATE-3



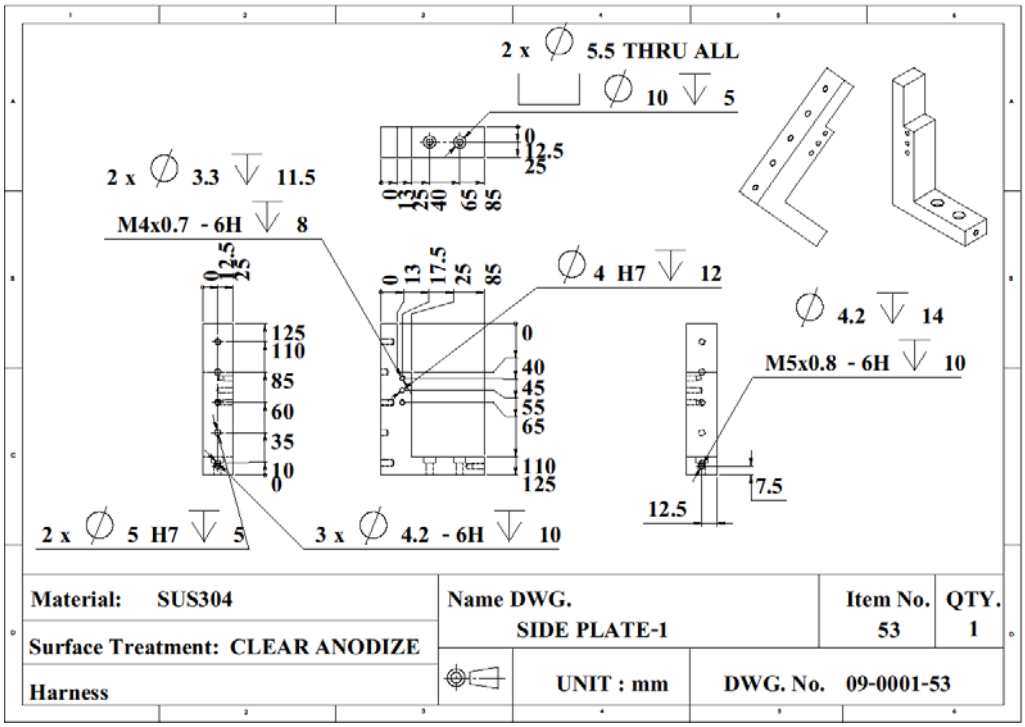
รูปที่ ก.58 รายละเอียดขนาดของ VACUUM PLATE-4



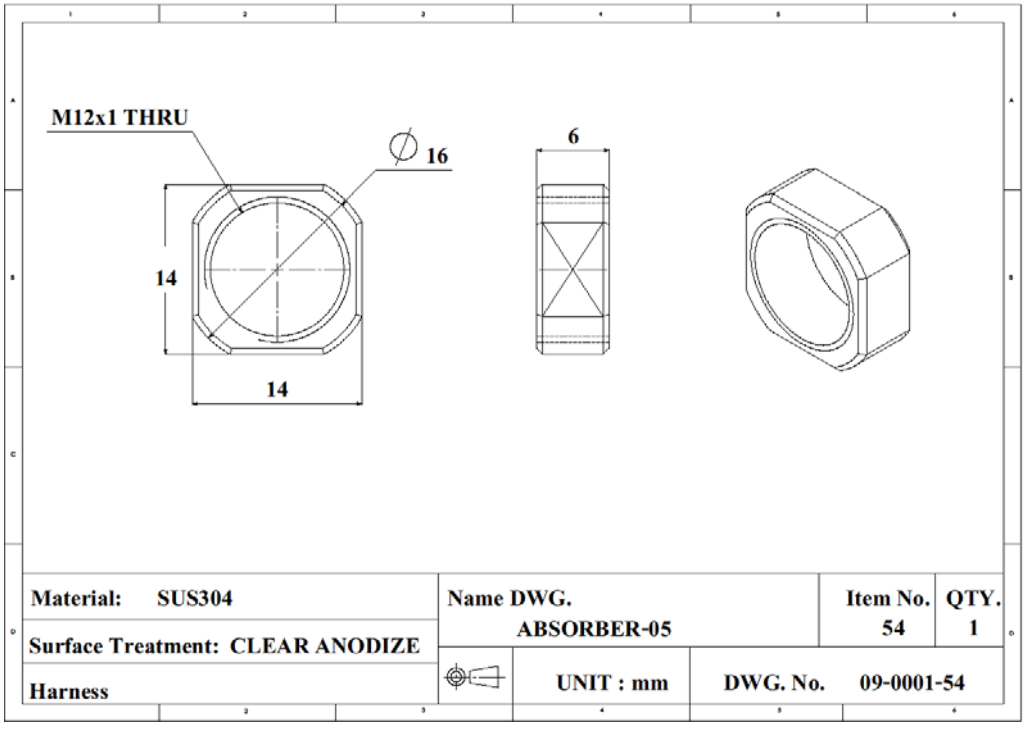
รูปที่ ก.59 รายละเอียดขนาดของ VACUUM ANGLE PLATE



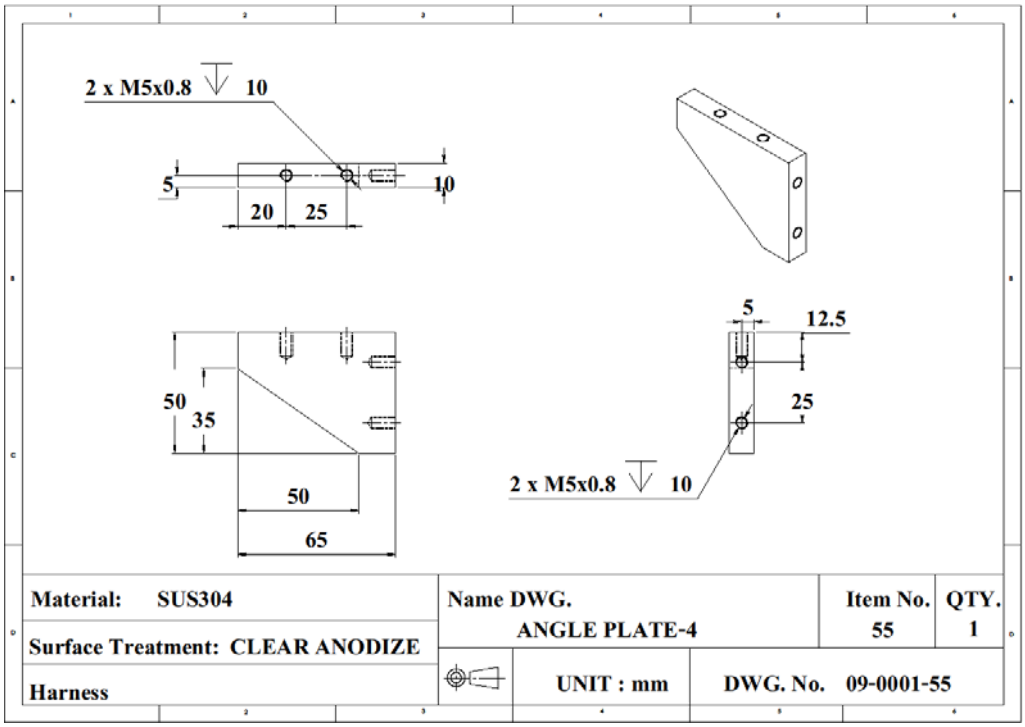
รูปที่ ก.60 รายละเอียดขนาดของ VACUUM PLUG



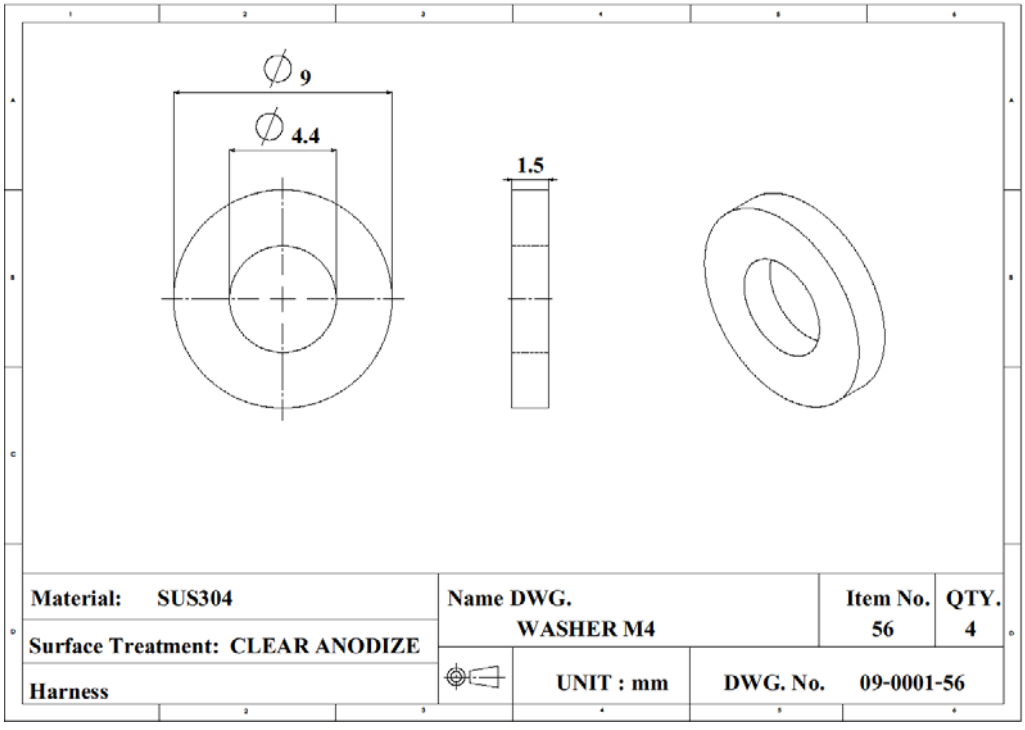
รูปที่ ก.61 รายละเอียดขนาดของ SIDE PLATE-1



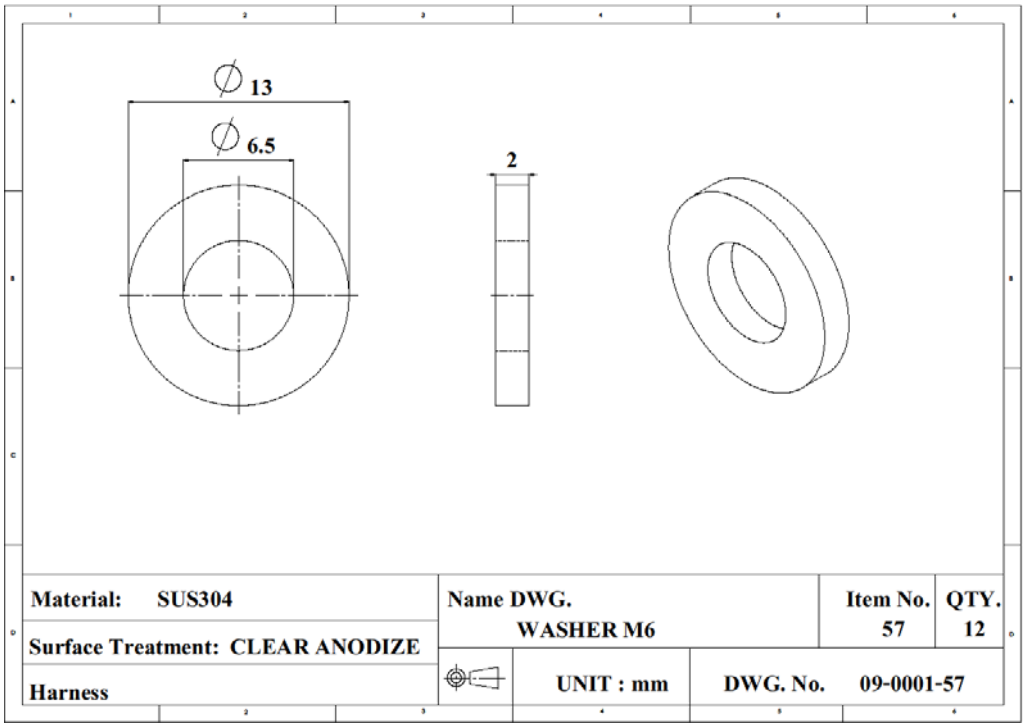
รูปที่ ก.62 รายละเอียดขนาดของ ABSORBER-05



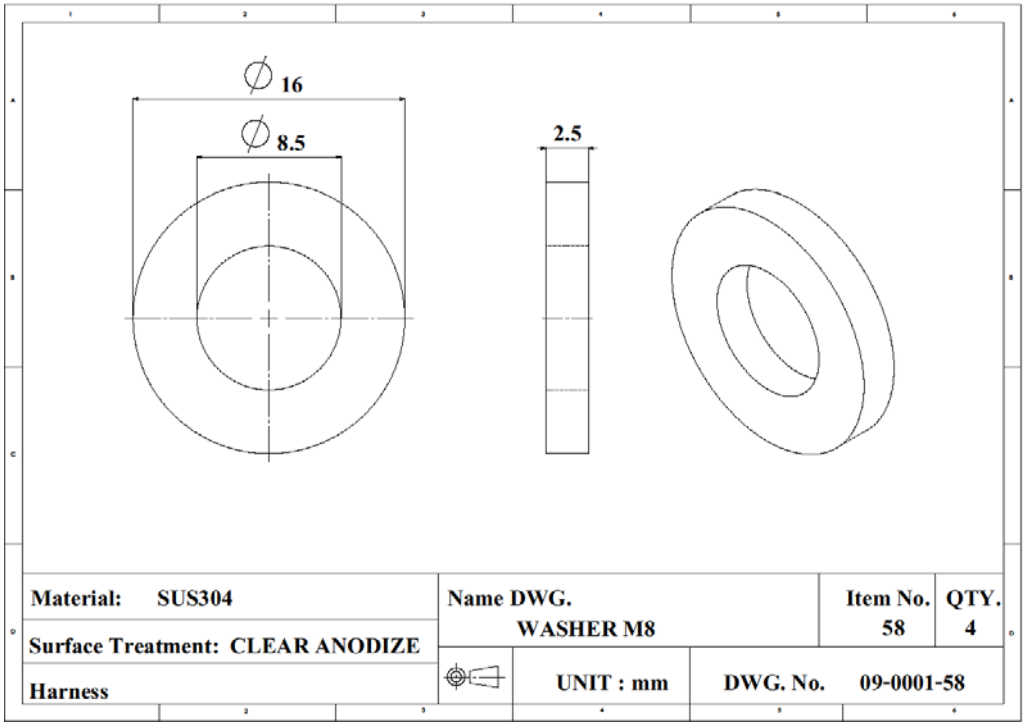
รูปที่ ก.63 รายละเอียดขนาดของ ANGLE PLATE-4



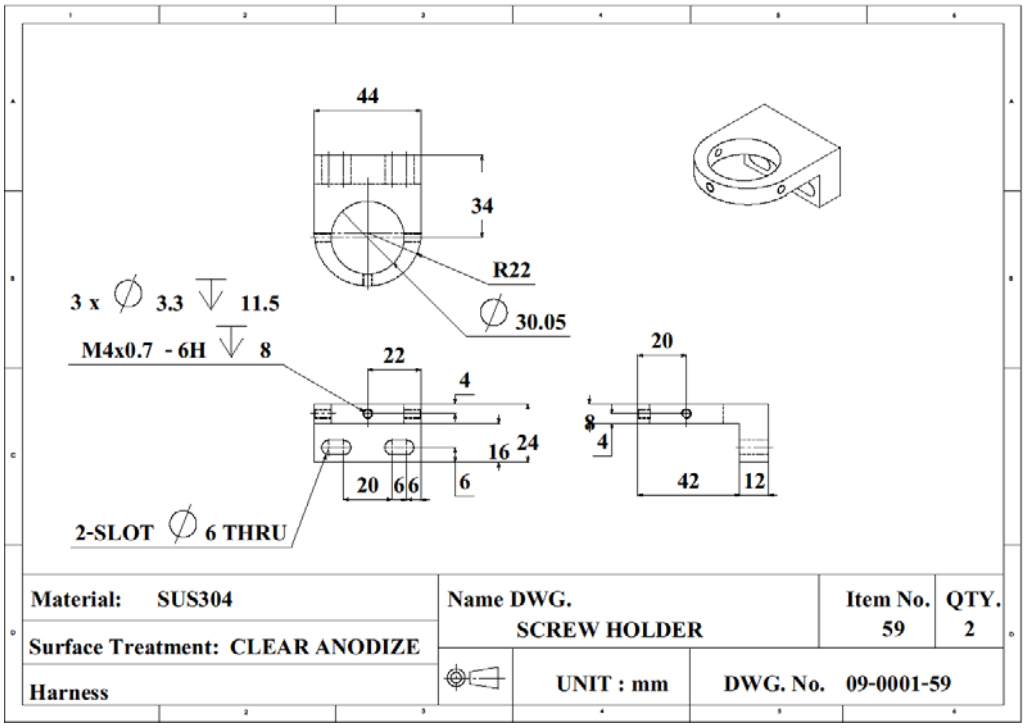
รูปที่ ก.64 รายละเอียดขนาดของ WASHER M4



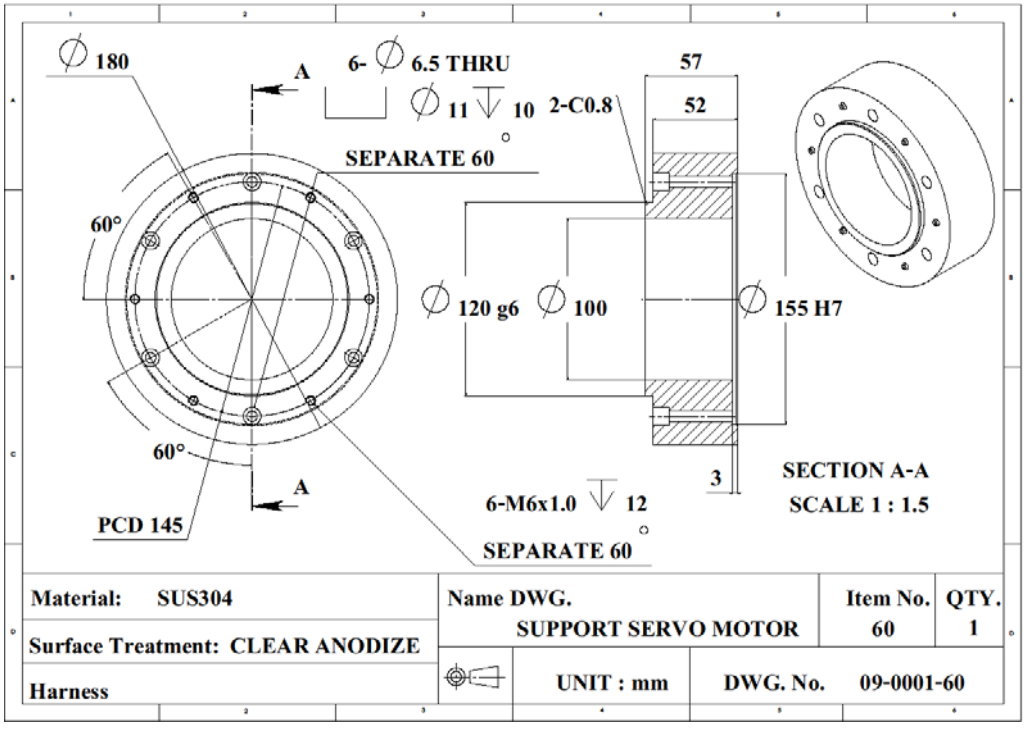
รูปที่ ก.65 รายละเอียดขนาดของ WASHER M6



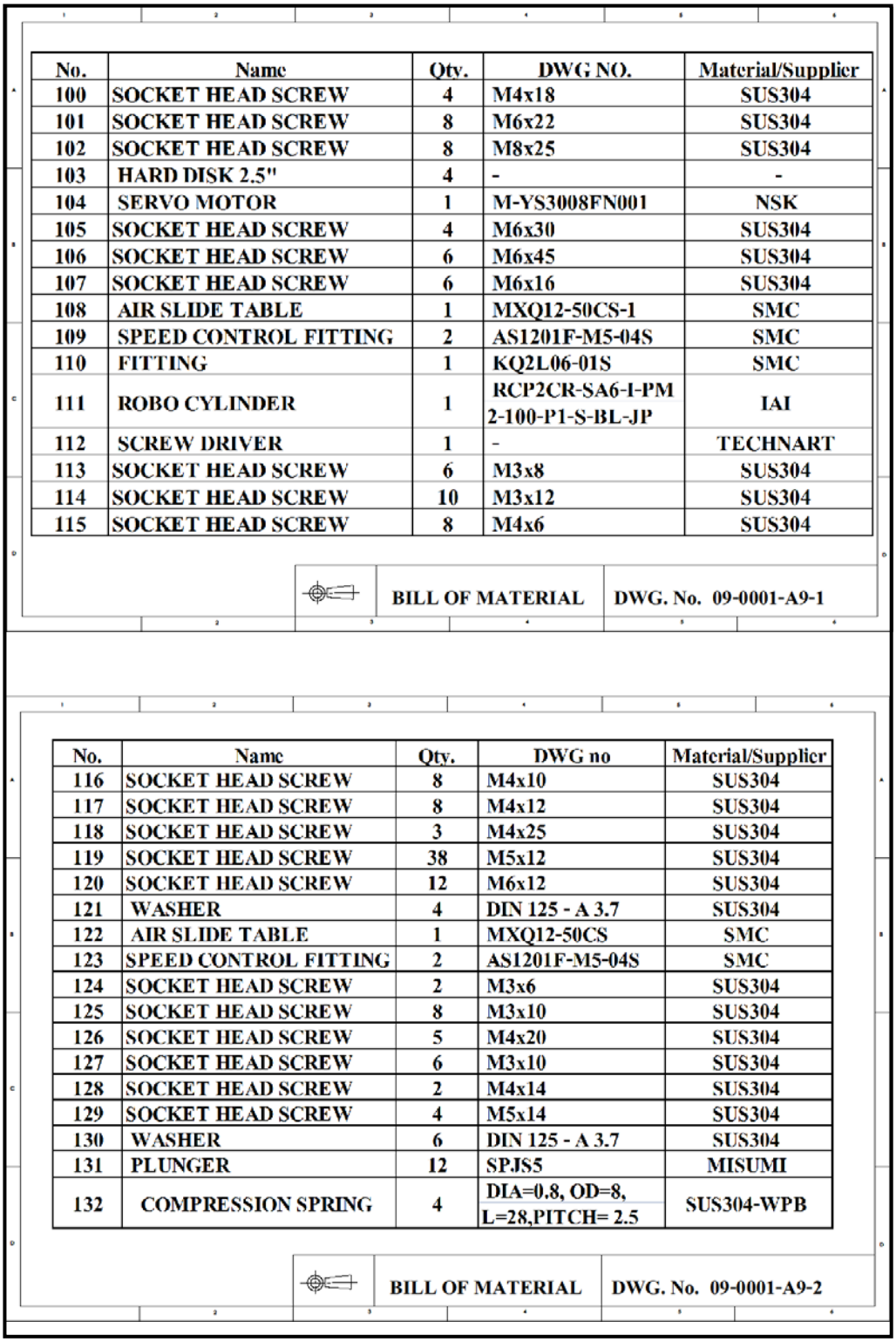
รูปที่ ก.66 รายละเอียดขนาดของ WASHER M8



รูปที่ ก.67 รายละเอียดขนาดของ SCREW HOLDER



รูปที่ ก.68 รายละเอียดขนาดของ SUPPORT SERVO MOTOR



No.	Name	Qty.	DWG NO.	Material/Supplier
100	SOCKET HEAD SCREW	4	M4x18	SUS304
101	SOCKET HEAD SCREW	8	M6x22	SUS304
102	SOCKET HEAD SCREW	8	M8x25	SUS304
103	HARD DISK 2.5"	4	-	-
104	SERVO MOTOR	1	M-YS3008FN001	NSK
105	SOCKET HEAD SCREW	4	M6x30	SUS304
106	SOCKET HEAD SCREW	6	M6x45	SUS304
107	SOCKET HEAD SCREW	6	M6x16	SUS304
108	AIR SLIDE TABLE	1	MXQ12-50CS-1	SMC
109	SPEED CONTROL FITTING	2	AS1201F-M5-04S	SMC
110	FITTING	1	KQ2L06-01S	SMC
111	ROBO CYLINDER	1	RCP2CR-SA6-I-PM 2-100-P1-S-BL-JP	IAI
112	SCREW DRIVER	1	-	TECHNART
113	SOCKET HEAD SCREW	6	M3x8	SUS304
114	SOCKET HEAD SCREW	10	M3x12	SUS304
115	SOCKET HEAD SCREW	8	M4x6	SUS304

BILL OF MATERIAL DWG. No. 09-0001-A9-1

No.	Name	Qty.	DWG no	Material/Supplier
116	SOCKET HEAD SCREW	8	M4x10	SUS304
117	SOCKET HEAD SCREW	8	M4x12	SUS304
118	SOCKET HEAD SCREW	3	M4x25	SUS304
119	SOCKET HEAD SCREW	38	M5x12	SUS304
120	SOCKET HEAD SCREW	12	M6x12	SUS304
121	WASHER	4	DIN 125 - A 3.7	SUS304
122	AIR SLIDE TABLE	1	MXQ12-50CS	SMC
123	SPEED CONTROL FITTING	2	AS1201F-M5-04S	SMC
124	SOCKET HEAD SCREW	2	M3x6	SUS304
125	SOCKET HEAD SCREW	8	M3x10	SUS304
126	SOCKET HEAD SCREW	5	M4x20	SUS304
127	SOCKET HEAD SCREW	6	M3x10	SUS304
128	SOCKET HEAD SCREW	2	M4x14	SUS304
129	SOCKET HEAD SCREW	4	M5x14	SUS304
130	WASHER	6	DIN 125 - A 3.7	SUS304
131	PLUNGER	12	SPJS5	MISUMI
132	COMPRESSION SPRING	4	DIA=0.8, OD=8, L=28,PITCH= 2.5	SUS304-WPB

BILL OF MATERIAL DWG. No. 09-0001-A9-2

รูปที่ ก.69 รายการหมายเลขของชิ้นส่วน

ภาคผนวก ข

รายละเอียดโปรแกรม

ข.1 โปรแกรมควบคุมและสั่งงานการทำงานของเครื่องต้นแบบ

เครื่องจักรต้นแบบสำหรับงานวิจัยนี้เลือกใช้ระบบการควบคุมแบบไมโครคอนโทรลเลอร์ โดยไมโครคอนโทรลเลอร์ที่เลือกใช้คือรุ่น ATMEGA 1280 ตระกูล AVR เป็นบอร์ดของบริษัท อีทีที รุ่น ET-EASY MEGA1280 (DUINO MEGA) ซึ่งบอร์ดรุ่นนี้รองรับการเขียนโปรแกรมภาษาซี ของ Arduino ที่ใช้ปืนหน่วยประมวลผลหลัก สำหรับรายละเอียดของโปรแกรมควบคุม และสั่งงาน การทำงานเครื่องจักรต้นแบบที่เขียนด้วยภาษาซีของ Arduino มีดังนี้

```
#include <EEPROM.h>
int Rb2;
int Rb3;
int CheckHome;
int StatusHome=0;
int CheckStart;
int StatusStart=0;
int CheckStop;
int CheckReset;
int CheckNSK;
int CheckPosNSK;
int CountLoop=0;
int data_nsk;
int ofset;
int CheckSenClampUp;
int CheckSenClampDown;
int CheckSenVacUP;
int CheckSenVacDown;
int CheckSenOpject;
int SenEmg;
int CountOffPump;
//============================data NSK
int Ofset_1;
```

```
int Ofset_10;
int Ofset_100;
int Ofset_1000;
unsigned int Ofset;

int plt1_1;
int plt1_10;
int plt1_100;
int plt1_1000;
unsigned int plt1;

int plt2_1;
int plt2_10;
int plt2_100;
int plt2_1000;
unsigned int plt2;

int plt3_1;
int plt3_10;
int plt3_100;
int plt3_1000;
unsigned int plt3;

int plt4_1;
int plt4_10;
int plt4_100;
int plt4_1000;
unsigned int plt4;

// Input
```

```
const int Sw_Home=23;
const int Sw_Reset=24;
const int Sw_Stop=25;
const int Sw_Start=22;
const int SenUpClamp=27;
const int SenDownClamp=26;
const int SenUpVac=29;
const int SenDownVac=28;

const int NskReady=31;
const int NskInPos=30;
const int SenOpject=33;
const int SenEMGROBO=32;
const int Sp3=35;
const int Sp4=34;
const int Sp5=37;
const int Sp6=36;

//  Output
const int Lamp_Home=39;
const int Lamp_Reset=40;
const int Lamp_Stop=41;
const int Lamp_Start=38;
const int SolClamp=43;
const int SolVac=42;
const int OnDrive=45;
const int OnPump=44;

const int OnEMGROBO=47;
const int Spo2=46;
```

```
const int Spo3=49;
const int Spo4=48;
const int Spo5=51;
const int Spo6=50;
const int Spo7=53;
const int Spo8=52;

//Rtx data Robo
byte  Robobyte[16];

void setup()
{
  // initialize both serial ports:
  Serial.begin(9600);
  Serial3.begin(9600);
  Serial2.begin(9600);
  // Input
pinMode (Sw_Home,INPUT);
pinMode(Sw_Reset,INPUT);
pinMode(Sw_Stop,INPUT);
pinMode(Sw_Start,INPUT);
pinMode(SenUpClamp,INPUT);
pinMode(SenDownClamp,INPUT);
pinMode(SenUpVac,INPUT);
pinMode(SenDownVac,INPUT);
pinMode(NskReady,INPUT);
pinMode(NskInPos,INPUT);
pinMode(SenOpject,INPUT);
pinMode(SenEMGROBO,INPUT);
pinMode(Sp3,INPUT);
```

```
pinMode(Sp4,INPUT);
pinMode(Sp5,INPUT);
pinMode(Sp6,INPUT);
// Output
pinMode(Lamp_Home,OUTPUT);
pinMode(Lamp_Reset,OUTPUT);
pinMode(Lamp_Stop,OUTPUT);
pinMode(Lamp_Start,OUTPUT);
pinMode(SolClamp,OUTPUT);
pinMode(OnEMGROBO,OUTPUT);
pinMode(SolVac,OUTPUT);
pinMode(Spo2,OUTPUT);
pinMode(OnPump,OUTPUT);
pinMode(OnDrive,OUTPUT);
pinMode(Spo3,OUTPUT);
pinMode(Spo4,OUTPUT);
pinMode(Spo5,OUTPUT);
pinMode(Spo6,OUTPUT);
pinMode(Spo7,OUTPUT);
pinMode(Spo8,OUTPUT);
CountLoop=1;
digitalWrite(OnEMGROBO,HIGH);
}

void loop()
{
 SenEmg=digitalRead(SenEMGROBO);
   if (SenEmg==HIGH)
   {
    digitalWrite(OnEMGROBO,LOW);
```

```
  }
CheckNSK=digitalRead(NskReady);
if (CheckNSK==HIGH)
{
  do
  {
 CheckReset=digitalRead(Sw_Reset);
digitalWrite(Lamp_Reset,HIGH);
  }
  while(CheckReset !=LOW);
  CheckReset=digitalRead(Sw_Reset);
   if (CheckReset==LOW);
{
  Serial3.print("CL");//RESET NSK
  Serial3.print(0x0d,BYTE);//RESET NSK
}
}
//Conect To Pc
if (Serial.available())
{
do
{
 if (Serial.available())
 {
 int Rb2 = Serial.read();
    // Conect To NSK
 if (Rb2 == '<')
     {
       Rb3=0;
do
```

```
 {
 if (Serial.available())
 {
  Rb3 = Serial.read();
  Serial3.print(Rb3, BYTE);
 }
  if (Serial3.available())
 {
  int inByte = Serial3.read();
  Serial.print(inByte, BYTE);
 }
 if (Serial2.available())
 {
  int inByte = Serial2.read();
 }
 }
  while(Rb3!='@');
  goto nx;
     }
  else if (Rb2 == '!')
  {
    goto nx;
  }
  else if (Rb2 == '^')
  {
Serial.print(digitalRead(Sw_Home));                    //1
Serial.print(digitalRead(Sw_Reset));                   //2
Serial.print(digitalRead(Sw_Stop));                    //3
Serial.print(digitalRead(Sw_Start));          //4
Serial.print(digitalRead(SenUpClamp));                 //5
```

```
Serial.print(digitalRead(SenDownClamp));      //6
Serial.print(digitalRead(SenUpVac));                  //7
Serial.print(digitalRead(SenDownVac));                //8
Serial.print(digitalRead(NskReady));                  //9
Serial.print(digitalRead(NskInPos));                  //10
Serial.print(digitalRead(SenOpject));                 //11
Serial.print(digitalRead(SenEMGROBO));
Serial.print(digitalRead(Lamp_Home));                 //12
Serial.print(digitalRead(Lamp_Reset));                //13
Serial.print(digitalRead(Lamp_Stop));                 //14
Serial.print(digitalRead(Lamp_Start));                //15
Serial.print(digitalRead(SolClamp));                  //16
Serial.print(digitalRead(SolVac));            //17
Serial.print(digitalRead(OnPump));                    //18
Serial.print(digitalRead(OnDrive));           //19
Serial.println();
   }
    else if (Rb2 == '#')
    {
      int d=0;
      do
      {
        if (Serial.available())
        {
        data_nsk  = Serial.read();
        EEPROM.write(d,data_nsk);
        d++;
        }
      }
      while(d!=41);
```

```
// Set Parameter NSK
Serial3.print("MA");                                    //Move Acc. NSK
Serial3.print(EEPROM.read(20),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(21),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(22),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(23),BYTE);          //Acc. NSK
Serial3.print(0x0d,BYTE);                          //Enter
delay(100);
Serial3.print("VG");                                    //Move Acc. NSK
Serial3.print(EEPROM.read(24),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(25),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(26),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(27),BYTE);          //Acc. NSK
Serial3.print(0x0d,BYTE);                          //Enter
delay(100);
Serial3.print("PG");                                    //Move Acc. NSK
Serial3.print(EEPROM.read(28),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(29),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(30),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(31),BYTE);          //Acc. NSK
Serial3.print(0x0d,BYTE);                          //Enter
delay(100);
Serial3.print("TL");                                    //Move Acc. NSK
Serial3.print(EEPROM.read(32),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(33),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(34),BYTE);          //Acc. NSK
Serial3.print(EEPROM.read(35),BYTE);          //Acc. NSK
Serial3.print(0x0d,BYTE);                          //Enter
delay(100);
Serial3.print("IN");                                    //Move Acc. NSK
```

```
 Serial3.print(EEPROM.read(36),BYTE);                    //Acc. NSK
 Serial3.print(EEPROM.read(37),BYTE);                    //Acc. NSK
 Serial3.print(EEPROM.read(38),BYTE);                    //Acc. NSK
 Serial3.print(EEPROM.read(39),BYTE);                    //Acc. NSK
 Serial3.print(0x0d,BYTE);                               //Enter
}
else if (Rb2 == '$')
{
  int e=0;
  do
  {
    Serial.print(EEPROM.read(e));
    e++;
  }
  while(e!=41);
  //go to nx;
}
else if (Rb2 == 'a')
{
 digitalWrite(Lamp_Home,HIGH);
 //go to nx;
}
else if (Rb2 == 'A')
{
 digitalWrite(Lamp_Home,LOW);
 //go to nx;
}
else if (Rb2 == 'b')
{
 digitalWrite(Lamp_Reset,HIGH);
```

```
 //go to nx;
}
else if (Rb2 == 'B')
{
 digitalWrite(Lamp_Reset,LOW);
 //go to nx;
}
else if (Rb2 == 'c')
{
 digitalWrite(Lamp_Stop,HIGH);
 //go to nx;
}
else if (Rb2 == 'C')
{
 digitalWrite(Lamp_Stop,LOW);
 //go to nx;
}
else if (Rb2 == 'd')
{
 digitalWrite(Lamp_Start,HIGH);
 //go to nx;
}
else if (Rb2 == 'D')
{
 digitalWrite(Lamp_Start,LOW);
 //go to nx;
}
else if (Rb2 == 'e')
{
 digitalWrite(SolClamp,HIGH);
```

```
// go to nx;
}
else if (Rb2 == 'E')
{
 digitalWrite(SolClamp,LOW);
 //go to nx;
}
else if (Rb2 == 'f')
{
 digitalWrite(SolVac,HIGH);
 //go to nx;
}
else if (Rb2 == 'F')
{
 digitalWrite(SolVac,LOW);
 //go to nx;
}
else if (Rb2 == 'g')
{
 digitalWrite(OnPump,HIGH);
 //go to nx;
}
else if (Rb2 == 'G')
{
 digitalWrite(OnPump,LOW);
 //go to nx;
}
else if (Rb2 == 'h')
{
 digitalWrite(OnDrive,HIGH);
```

```
      //go to nx;
     }
     else if (Rb2 == 'H')
     {
      digitalWrite(OnDrive,LOW);
      //go to nx;
     }
     else if (Rb2 == 'i')
     {
      digitalWrite(OnEMGROBO,HIGH);
      //go to nx;
     }
     else if (Rb2 == 'I')
     {
      digitalWrite(OnEMGROBO,LOW);
     //go to nx;
     }
    Serial2.print(Rb2, BYTE);
   }
   if (Serial2.available())
   {
    int inByte = Serial2.read();
    Serial.print(inByte, BYTE);
   }
}
  while(Rb2!='!');
  // end connect PC
  }
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
  nx:
```

```
CheckHome=digitalRead(Sw_Home);                          //Check Switch Home
if (CheckHome==LOW)
{
CheckSenClampUp=digitalRead(SenUpClamp);
CheckSenVacUP=digitalRead(SenUpVac);
if (CheckSenClampUp==HIGH || CheckSenVacUP==HIGH)
{
 digitalWrite(Lamp_Reset,HIGH);
 delay(250);
 Serial.print("Clamp Down");
 digitalWrite(Lamp_Reset,LOW);
 delay (250);
goto nx;
}
  Serial2.flush();
  digitalWrite(Lamp_Home,HIGH);
  Serial2.print("┐ 0o07000000007A └ ");                  //Home Robo
  if (Serial2.available())
{
  int inByte = Serial2.read();
}
  delay(2000);
do
{
CheckHome=digitalRead(Sw_Home);
}
while(CheckHome!=HIGH);
Serial2.flush();
NotHome:
 int i=0;
```

```
Serial2.print("┐ 0n000000000082 └"); // Check status
do
{
if (Serial2.available()>0)
   {
    Robobyte[i]=Serial2.read();
    i++;
   }
}
while(i!=16);
//Serial.println(Robobyte[10]);
delay(5);
if (Robobyte[10]=='F')
{
Serial3.print(0x0d,BYTE);                    //Home NSK
delay(100);
Serial3.print("HS");                         //Home NSK
Serial3.print(0x0d,BYTE);                    //Home NSK
 Posnsk1:
CheckPosNSK=digitalRead(NskInPos);
 if (CheckPosNSK==HIGH)
 {
  goto Posnsk1;
 }
 //offset NSK
 delay(500);
 ofset = EEPROM.read(40);
if (ofset =='z')                             //Ofset CCW
{
  Serial3.print("ID-");
```

```
  Serial3.print(EEPROM.read(0),BYTE);
  Serial3.print(EEPROM.read(1),BYTE);
  Serial3.print(EEPROM.read(2),BYTE);
  Serial3.print(EEPROM.read(3),BYTE);
  Serial3.print(0x0d,BYTE);                                    //Home NSK
  digitalWrite(Lamp_Home,LOW);
 }
 else if (ofset =='Z')                                         //Offset CCW
 {
  Serial3.print("ID");
  Serial3.print(EEPROM.read(0),BYTE);
  Serial3.print(EEPROM.read(1),BYTE);
  Serial3.print(EEPROM.read(2),BYTE);
  Serial3.print(EEPROM.read(3),BYTE);
  Serial3.print(0x0d,BYTE);//Home NSK
  digitalWrite(Lamp_Home,LOW);
 }
digitalWrite(Lamp_Stop,LOW);
StatusHome=1;
 }
 else
 {
//Serial.println(Robobyte[10]);
  goto NotHome;
 }
 }
 // Indicator Home
 Home:
 if (StatusStart==1 && StatusHome==0)
 {
```

```
  digitalWrite(Lamp_Home,HIGH);
  delay (250);
  digitalWrite(Lamp_Home,LOW);
  delay (250);
  //Serial.println(StatusStart);
  //Serial.println(StatusHome);
}

//=============================================================Loop Run
CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
  StatusStart=1;
  if (StatusHome==0)
  {
    digitalWrite(OnPump,HIGH);
    goto Home;
  }
  Serial2.print("┐ 0Q301010000009A └");                    //Move Robo Pos1
  SenEmg=digitalRead(SenEMGROBO);
  if (SenEmg==HIGH)
  {
   digitalWrite(OnEMGROBO,LOW);
  }
  delay (45);
   Serial2.flush();
 Check_RoboposStart:
 SenEmg=digitalRead(SenEMGROBO);
  if (SenEmg==HIGH)
  {
```

```
   digitalWrite(OnEMGROBO,LOW);
  }
    int i=0;
Serial2.print("┐ 0n000000000082 ┗ ");                    // Check status
do
{
if (Serial2.available()>0)                              // Robo Cylinder
   {
    Robobyte[i]=Serial2.read();
    //Serial.print(Robobyte[i]);
    i++;
   }
}
while(i!=16);
//Serial.println(Robobyte[11]);
delay(5);
if (Robobyte[11]=='1')
{
  CheckStop=digitalRead(Sw_Stop);
  if (CheckStop==LOW)
  {
    goto Mc_Stop;
  }
  //Serial.println(Robobyte[11]);
  goto nxStep;
}
else
{
goto Check_RoboposStart;
}
```

```
do
{
nxStep:                                                    //Begin LOOP RUN
 if (StatusStart==0)
 {
digitalWrite(Lamp_Start,LOW);
CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
CheckStop=digitalRead(Sw_Stop);
CountOffPump++;
if (CountOffPump==60)                                      //delay Off Pump
{
  digitalWrite(OnPump,LOW);
}
  if (CheckStop==LOW)
  {
    Serial.print("Stop");
    goto Mc_Stop;
  }
  int i=0;
 for (int i=0; i <= 20; i++)
 {
   delay(7);
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
```

```
 StatusStart=1;
  goto nxStep;
}
 }
digitalWrite(Lamp_Start,HIGH);
 for (int i=0; i <= 20; i++)
 {
   delay(7);
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 goto nxStep;
}
 }
goto nxStep;
 }
 CountOffPump=0;
 StatusStart=0;
digitalWrite(OnPump,HIGH);
digitalWrite(Lamp_Start,LOW);
digitalWrite(Lamp_Stop,LOW);
CheckupClamp:
CheckSenClampUp=digitalRead(SenUpClamp);
CheckSenVacUP=digitalRead(SenUpVac);
if (CheckSenClampUp==HIGH || CheckSenVacUP==HIGH)
{
 digitalWrite(Lamp_Reset,HIGH);
 delay(250);
 Serial.print("Clamp Down");
```

```
 digitalWrite(Lamp_Reset,LOW);
 delay (250);
goto CheckupClamp;
}
//Select Move Fixture Rotary
movensk:
switch (CountLoop)
{
  //Serial.print("CountLoop  ");
  //Serial.println(CountLoop);
case 1:
  Serial.print(digitalRead(Sw_Home));                    //1
Serial.print(digitalRead(Sw_Reset));                     //2
Serial.print(digitalRead(Sw_Stop));                      //3
Serial.print(digitalRead(Sw_Start));                     //4
Serial.print(digitalRead(SenUpClamp));                   //5
Serial.print(digitalRead(SenDownClamp));                 //6
Serial.print(digitalRead(SenUpVac));                     //7
Serial.print(digitalRead(SenDownVac));                   //8
Serial.print(digitalRead(NskReady));                     //9
Serial.print(digitalRead(NskInPos));                     //10
Serial.print(digitalRead(SenOpject));                    //11
Serial.print(digitalRead(SenEMGROBO));
Serial.print(digitalRead(Lamp_Home));                    //12
Serial.print(digitalRead(Lamp_Reset));                   //13
Serial.print(digitalRead(Lamp_Stop));                    //14
Serial.print(digitalRead(Lamp_Start));                   //15
Serial.print(digitalRead(SolClamp));                     //16
Serial.print(digitalRead(SolVac));                       //17
Serial.print(digitalRead(OnPump));                       //18
```

```
Serial.print(digitalRead(OnDrive));                                    //19
Serial.println();
     Serial3.print("ID-");
    Serial3.print(EEPROM.read(4),BYTE);
    Serial3.print(EEPROM.read(5),BYTE);
    Serial3.print(EEPROM.read(6),BYTE);
    Serial3.print(EEPROM.read(7),BYTE);
    Serial3.print(0x0d,BYTE);                                  //Home NSK
   Serial3.print(0x0d,BYTE);                                   //NSK
   break;
   case 2:
   Serial.print(digitalRead(Sw_Home));                              //1
Serial.print(digitalRead(Sw_Reset));                                //2
Serial.print(digitalRead(Sw_Stop));                                 //3
Serial.print(digitalRead(Sw_Start));                                //4
Serial.print(digitalRead(SenUpClamp));                              //5
Serial.print(digitalRead(SenDownClamp));                            //6
Serial.print(digitalRead(SenUpVac));                                //7
Serial.print(digitalRead(SenDownVac));                              //8
Serial.print(digitalRead(NskReady));                                //9
Serial.print(digitalRead(NskInPos));                                //10
Serial.print(digitalRead(SenOpject));                               //11
Serial.print(digitalRead(SenEMGROBO));
Serial.print(digitalRead(Lamp_Home));                               //12
Serial.print(digitalRead(Lamp_Reset));                              //13
Serial.print(digitalRead(Lamp_Stop));                               //14
Serial.print(digitalRead(Lamp_Start));                              //15
Serial.print(digitalRead(SolClamp));                                //16
Serial.print(digitalRead(SolVac));                                  //17
Serial.print(digitalRead(OnPump));                                  //18
```

```
Serial.print(digitalRead(OnDrive));                                    //19
Serial.println();
     Serial3.print("ID-");
    Serial3.print(EEPROM.read(8),BYTE);
    Serial3.print(EEPROM.read(9),BYTE);
    Serial3.print(EEPROM.read(10),BYTE);
    Serial3.print(EEPROM.read(11),BYTE);
    Serial3.print(0x0d,BYTE);                                  //Home NSK
   Serial3.print(0x0d,BYTE);                                   //NSK
   break;
   case 3:
   Serial.print(digitalRead(Sw_Home));                              //1
Serial.print(digitalRead(Sw_Reset));                                //2
Serial.print(digitalRead(Sw_Stop));                                 //3
Serial.print(digitalRead(Sw_Start));                                //4
Serial.print(digitalRead(SenUpClamp));                              //5
Serial.print(digitalRead(SenDownClamp));                            //6
Serial.print(digitalRead(SenUpVac));                                //7
Serial.print(digitalRead(SenDownVac));                              //8
Serial.print(digitalRead(NskReady));                                //9
Serial.print(digitalRead(NskInPos));                                //10
Serial.print(digitalRead(SenOpject));                               //11
Serial.print(digitalRead(Lamp_Home));                               //12
Serial.print(digitalRead(Lamp_Reset));                              //13
Serial.print(digitalRead(Lamp_Stop));                               //14
Serial.print(digitalRead(Lamp_Start));                              //15
Serial.print(digitalRead(SolClamp));                                //16
Serial.print(digitalRead(SolVac));                                  //17
Serial.print(digitalRead(OnPump));                                  //18
Serial.print(digitalRead(OnDrive));                                 //19
```

```
Serial.println();
     Serial3.print("ID-");
    Serial3.print(EEPROM.read(12),BYTE);
    Serial3.print(EEPROM.read(13),BYTE);
    Serial3.print(EEPROM.read(14),BYTE);
    Serial3.print(EEPROM.read(15),BYTE);
    Serial3.print(0x0d,BYTE);                                    //Home NSK
   Serial3.print(0x0d,BYTE);                                     //NSK
   break;
   case 4:
   Serial.print(digitalRead(Sw_Home));                                 //1
Serial.print(digitalRead(Sw_Reset));                                   //2
Serial.print(digitalRead(Sw_Stop));                                    //3
Serial.print(digitalRead(Sw_Start));                                   //4
Serial.print(digitalRead(SenUpClamp));                                 //5
Serial.print(digitalRead(SenDownClamp));                               //6
Serial.print(digitalRead(SenUpVac));                                   //7
Serial.print(digitalRead(SenDownVac));                                 //8
Serial.print(digitalRead(NskReady));                                   //9
Serial.print(digitalRead(NskInPos));                                   //10
Serial.print(digitalRead(SenOpject));                                  //11
Serial.print(digitalRead(Lamp_Home));                                  //12
Serial.print(digitalRead(Lamp_Reset));                                 //13
Serial.print(digitalRead(Lamp_Stop));                                  //14
Serial.print(digitalRead(Lamp_Start));                                 //15
Serial.print(digitalRead(SolClamp));                                   //16
Serial.print(digitalRead(SolVac));                                     //17
Serial.print(digitalRead(OnPump));                                     //18
Serial.print(digitalRead(OnDrive));                                    //19
Serial.println();
```

```
 Serial3.print("ID-");
 Serial3.print(EEPROM.read(16),BYTE);
 Serial3.print(EEPROM.read(17),BYTE);
 Serial3.print(EEPROM.read(18),BYTE);
 Serial3.print(EEPROM.read(19),BYTE);
 Serial3.print(0x0d,BYTE);                                    //Home NSK
Serial3.print(0x0d,BYTE);                                     //NSK
break;
}
Posnsk:
CheckPosNSK=digitalRead(NskInPos);
if (CheckPosNSK==HIGH)
{
 goto Posnsk;
}
else
{
 //==================================Check disk ว่ามีหรือไม่มีถ้าไม่มีจะข้าม
 CheckSenOpject=digitalRead(SenOpject);
 if (CheckSenOpject==HIGH)
 {
   CheckStop=digitalRead(Sw_Stop);
if (CheckStop==LOW)
{
 goto Mc_Stop;
}
  CountLoop=CountLoop+1;
 if (CountLoop==5)
 {
 CountLoop=1;
```

```
    }
    goto movensk;
   }
   //=========================================ปิดการ Check
digitalWrite(SolClamp,HIGH);                    // Clamp ลง
delay(50);
  CheckDown:
 CheckSenClampDown=digitalRead(SenDownClamp);   // Check Sensor ตอนลง
 if (CheckSenClampDown==HIGH)
 {
  delay(150);
  Serial.print("Clamp Down");
  digitalWrite(Lamp_Reset,LOW);
  delay (150);
 goto CheckDown;
 }
digitalWrite(SolVac,HIGH);                      // Vacuum ลง
delay(50);
CheckVacDown:
 CheckSenVacDown=digitalRead(SenDownVac);       // Check Sensor ตอนลง
 if (CheckSenVacDown==HIGH)
 {
  digitalWrite(Lamp_Reset,HIGH);
 digitalWrite(Lamp_Reset,HIGH);
  delay(150);
  Serial.print("Clamp Down");
  digitalWrite(Lamp_Reset,LOW);
  delay (150);
 goto CheckVacDown;
 }
```

```
 Serial2.print("┐ 0Q30101000009A └ ");                    //Move Robo Pos1
 ////////////////////////////////////////////////////////////////////////////////////////////////////ชั้นที่ 1
    if (SenEmg==HIGH)
 {
  digitalWrite(OnEMGROBO,LOW);
 }
 delay (45);
  Serial2.flush();
Check_Robopos1:
SenEmg=digitalRead(SenEMGROBO);
 if (SenEmg==HIGH)
 {
  digitalWrite(OnEMGROBO,LOW);
 }
   int i=0;
Serial2.print("┐ 0n000000000082 └ ");                    // Check status
do
{
if (Serial2.available()>0)                               // Robo Cylinder
   {
    Robobyte[i]=Serial2.read();
    //Serial.print(Robobyte[i]);
    i++;
   }
}
while(i!=16);
//Serial.println(Robobyte[11]);
delay(5);
if (Robobyte[11]=='1')
{
```

```
 CheckStop=digitalRead(Sw_Stop);
 if (CheckStop==LOW)
 {
  Serial.print("Stop");
  goto Mc_Stop;
 }
 CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
 //Serial.println(Robobyte[11]);
 delay(5);
Serial2.print("┐ 0Q301020000099 └ ");                    //Move Robo Pos2
////////////////////////////////////////////////////////////////////////////////////////////////////ขั้นที่ 2
   if (SenEmg==HIGH)
 {
  digitalWrite(OnEMGROBO,LOW);
 }
 delay (45);
  Serial2.flush();
Check_Robopos2:
SenEmg=digitalRead(SenEMGROBO);
 if (SenEmg==HIGH)
 {
  digitalWrite(OnEMGROBO,LOW);
 }
   int i=0;
Serial2.print("┐ 0n000000000082 └ ");                    // Check status
```

```
do
{
if (Serial2.available()>0)                                          // Robo Cylinder
   {
    Robobyte[i]=Serial2.read();
    //Serial.print(Robobyte[i]);
    i++;
   }
}
while(i!=16);
if (Robobyte[11]=='2')
{
 CheckStop=digitalRead(Sw_Stop);
 if (CheckStop==LOW)
 {
  goto Mc_Stop;
 }
 CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
 //Serial.println(Robobyte[11]);
 delay(5);
                        ////////เปิดดอก สกรู///////////////
//////////////////////////////////////////////////////////////////////////////////////////////////////////////
 digitalWrite(OnDrive,HIGH);
 // Check Alarm  Drive
```

```
// ปิด Check Alarm

  Serial2.print("┐ 0Q301030000098 └ ");                    //Move Robo Pos3
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////ชั้นที่ 3
     if (SenEmg==HIGH)
   {
    digitalWrite(OnEMGROBO,LOW);
   }
   delay (45);
    Serial2.flush();
  Check_Robopos3:
  SenEmg=digitalRead(SenEMGROBO);
   if (SenEmg==HIGH)
   {
    digitalWrite(OnEMGROBO,LOW);
   }
     int i=0;
 Serial2.print("┐ 0n000000000082 └ ");                          // Check status
 do
 {
 if (Serial2.available()>0)                                    // Robo Cylinder
    {
      Robobyte[i]=Serial2.read();
      //Serial.print(Robobyte[i]);
      i++;
    }
 }
 while(i!=16);
 if (Robobyte[11]=='3')
 {
```

```
 CheckStop=digitalRead(Sw_Stop);
 if (CheckStop==LOW)
 {
  goto Mc_Stop;
 }
//Serial.println(Robobyte[11]);
delay(5);
CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
StatusStart=1;
digitalWrite(Lamp_Start,HIGH);
}
delay(500);
Serial2.print("┐ 0Q301040000097 └ ");                    //Move Robo Pos4
////////////////////////////////////////////////////////////////////////////////////////////////////////////ชั้นที่ 4
   if (SenEmg==HIGH)
 {
  digitalWrite(OnEMGROBO,LOW);
 }
 delay (45);
  Serial2.flush();
Check_Robopos4:
SenEmg=digitalRead(SenEMGROBO);
 if (SenEmg==HIGH)
 {
  digitalWrite(OnEMGROBO,LOW);
 }
   int i=0;
Serial2.print("┐ 0n000000000082 └ ");                    // Check status
```

```
do
{
if (Serial2.available()>0)                                          // Robo Cylinder
   {
    Robobyte[i]=Serial2.read();
    //Serial.print(Robobyte[i]);
    i++;
   }
}
while(i!=16);
if (Robobyte[11]=='4')
{
 digitalWrite(OnDrive,LOW);                                         //เปิด Screw Driver
 CheckStop=digitalRead(Sw_Stop);
 if (CheckStop==LOW)
 {
  goto Mc_Stop;
 }
 //Serial.println(Robobyte[11]);
 delay(5);
 CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
delay(1000);                                                        //หน่วงเวลายก Vacuum
Serial2.print("┐ 0Q301010000009A └ ");                             //Move Robo Pos1
  ////////////////////////////////////////////////////////////////////////////////////////////////////////////ชั้นที่ 5
     if (SenEmg==HIGH)
```

```
  {
   digitalWrite(OnEMGROBO,LOW);
  }
  delay (45);
   Serial2.flush();
 Check_Robopos5:
 SenEmg=digitalRead(SenEMGROBO);
  if (SenEmg==HIGH)
  {
   digitalWrite(OnEMGROBO,LOW);
  }
    int i=0;
Serial2.print("┐ 0n000000000082 └ ");                    // Check status
do
{
if (Serial2.available()>0)                               // Robo Cylinder
   {
    Robobyte[i]=Serial2.read();
    //Serial.print(Robobyte[i]);
    i++;
   }
}
 while(i!=16);
//Serial.println(Robobyte[11]);
delay(5);
 if (Robobyte[11]=='1')
 {
 ////////////////////////////////////////////////////////////////////////////จุดสิ้นสุด 1 รอบ
 digitalWrite(SolVac,LOW);                               //ยก Vacuum
 CheckVac:
```

```
CheckSenVacUP=digitalRead(SenUpVac);              //Check Sensor ตอนยก Vacuum
if (CheckSenVacUP==HIGH)
{
 digitalWrite(Lamp_Reset,HIGH);
 delay(250);
 Serial.print("Clamp Down");
 digitalWrite(Lamp_Reset,LOW);
 delay (250);
goto CheckVac;
}
delay(800);                                       //หน่วงเวลายก Clamp
digitalWrite(SolClamp,LOW);                       //ยก Clamp
 CheckCam:
CheckSenClampUp=digitalRead(SenUpClamp);          //Check Sensor ตอนยก  Clamp
if (CheckSenClampUp==HIGH)
{
 digitalWrite(Lamp_Reset,HIGH);
 delay(250);
 Serial.print("Clamp Down");
 digitalWrite(Lamp_Reset,LOW);
 delay (250);
goto CheckCam;
}

  CountLoop=CountLoop+1;
if (CountLoop==5)
{
  CountLoop=1;
}
goto nxStep;
```

```
}
else
{
 //Serial.println(Robobyte[11]);
 delay(5);
 CheckStop=digitalRead(Sw_Stop);
  if (CheckStop==LOW)
  {
    goto Mc_Stop;
  }
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
  goto Check_Robopos5;//=======================================ชั้นที่ 5
}
}
else
{
//Serial.println(Robobyte[11]);
delay(5);
CheckStop=digitalRead(Sw_Stop);
  if (CheckStop==LOW)
  {
    goto Mc_Stop;
  }
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
```

```
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
  goto Check_Robopos4;//=======================================ชั้นที่ 4
}
}
}
else
{
  //Serial.println(Robobyte[11]);
  delay(5);
  CheckStop=digitalRead(Sw_Stop);
  if (CheckStop==LOW)
  {
    goto Mc_Stop;
  }
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
  goto Check_Robopos3;//=======================================ชั้นที่ 3
}
//Serial.println(Robobyte[11]);
delay(5);
CheckStop=digitalRead(Sw_Stop);
  if (CheckStop==LOW)
  {
```

```
    goto Mc_Stop;
  }
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
goto Check_Robopos2;//=========================================ชั้นที่ 2
}
else
{
  //Serial.println(Robobyte[11]);
  delay(5);
  CheckStop=digitalRead(Sw_Stop);
  if (CheckStop==LOW)
  {
    goto Mc_Stop;
  }
  CheckStart=digitalRead(Sw_Start);
if (CheckStart==LOW)
{
 StatusStart=1;
 digitalWrite(Lamp_Start,HIGH);
}
  goto Check_Robopos1;//=========================================ชั้นที่ 1
}
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
 }
```

```
  CheckStop=digitalRead(Sw_Stop);
 }
 while(CheckStop!=LOW);
 Mc_Stop:
 digitalWrite(Lamp_Start,LOW);
 digitalWrite(Lamp_Stop,HIGH);
 }
 CheckReset=digitalRead(Sw_Reset);
 if (CheckReset==LOW)
 {
 digitalWrite(Lamp_Reset,HIGH);
 digitalWrite(SolClamp,LOW);
 delay (1000);
 Serial2.print("┐ 0o07000000007A └");                    //Home Robo Cylinder
 delay (2000);
 Serial3.print("ID-");
  Serial3.print(EEPROM.read(12),BYTE);
  Serial3.print(EEPROM.read(13),BYTE);
  Serial3.print(EEPROM.read(14),BYTE);
  Serial3.print(EEPROM.read(15),BYTE);
  Serial3.print(0x0d,BYTE);                              //Home NSK
 Serial3.print(0x0d,BYTE);                               //NSK
  delay(1000);
 }
}
```

ข.2 โปรแกรมติดต่อและควบคุมการทำงานของอุปกรณ์ต่าง ๆ ระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์

การติดต่อ และควบคุมการทำงานของอุปกรณ์ต่าง ๆ ของเครื่องจักรต้นแบบระหว่างคอมพิวเตอร์กับไมโครคอนโทรลเลอร์ เมื่อสร้างหน้าจอกราฟฟิกบนคอมพิวเตอร์เพื่อติดต่อสื่อสาร

กับไมโครคอนโทรลเลอร์โดยใช้โปรแกรมไมโครซอฟท์วิชวลเบสิก 6.0 แล้ว สำหรับรายละเอียดของโปรแกรมมีดังนี้

```
Dim rr As String
Dim Ofset As Double
Dim Datansk As String
Dim Direct As String
Dim flag As Integer

Private Sub CmdClear_Click()
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "CL" + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
End Sub

Private Sub cmdDis_Click()
If cmdDis.Caption = "Disable Program Show Status I/O" Then
Timer3.Enabled = False
Timer2.Enabled = True
cmdDis.BackColor = &H80FF80
cmdDis.Caption = "Enable Program Show Status I/O"
Else
Timer2.Enabled = False
cmdDis.BackColor = &H8080FF
cmdDis.Caption = "Disable Program Show Status I/O"
MSComm1.Output = "!"
```

```
Timer3.Enabled = True
End If
End Sub

Private Sub cmddown_Click()
Dim aa As String
flag = 1
aa = Chr$(&H2) + "0Q301020000099" + Chr$(&H3)
MSComm1.Output = aa
End Sub

Private Sub cmdHOME_Click()
Dim aa As String
flag = 1
aa = Chr$(&H2) + "0o07000000007A" + Chr$(&H3)
MSComm1.Output = aa
Delay (25)
MSComm1.Output = "!"
End Sub

Private Sub cmdOpen_Click()
Dim my As Integer
Timer3.Enabled = False
If cmdOpen.Caption = "Teach Point By RCPC" Then
MSComm1.PortOpen = False
my = Shell("C:\RcPc\RcPc.exe", vbNormalFocus)
cmdOpen.Caption = "Reset Commport"
tmtfocus.Enabled = True
FrameNSK.Enabled = False
FrameIO.Enabled = False
```

```
cmdHOME.Enabled = False
cmdstart.Enabled = False
cmddown.Enabled = False
cmdscrew.Enabled = False
cmdUpScrew.Enabled = False
Else
MSComm1.Settings = "9600,n,8,1"
On Error GoTo nx
MSComm1.PortOpen = True
cmdOpen.Caption = "Teach Point By RCPC"
tmtfocus.Enabled = False
FrameNSK.Enabled = True
FrameIO.Enabled = True
cmdHOME.Enabled = True
cmdstart.Enabled = True
cmddown.Enabled = True
cmdscrew.Enabled = True
cmdUpScrew.Enabled = True
cmdtestmove.Enabled = True
End If
GoTo exx
nx:
MsgBox "â»Ãá¡ÃÁ RCPC à»Ô´ÍÂÙè¡ÃØ³Ò»Ô´â»Ãá¡ÃÁ¡èÍ¹"
exx:
End Sub



Private Sub cmdPara_Click()
If F_para.Visible = False Then
F_para.Left = 120
```

```
F_para.Top = 3000
F_para.Height = 3135
F_para.Width = 5055
cmdPara.Caption = "Set Position"
cmdPara.BackColor = &HFFFFC0
F_para.Visible = True
F_Position.Visible = False
Else
cmdPara.Caption = "Set Parameter"
cmdPara.BackColor = &HFFC0C0
F_para.Visible = False
F_Position.Visible = True
End If
End Sub

Private Sub cmdRead_Click()
MSComm1.Output = "$"
tmtread.Enabled = True
End Sub

Private Sub cmdscrew_Click()
Dim aa As String
flag = 1
aa = Chr$(&H2) + "0Q301030000098" + Chr$(&H3)
MSComm1.Output = aa
End Sub

Private Sub cmdSet_Click()
txtOfset.Text = Format(txtOfset.Text, "0000")
txtp1.Text = Format(txtp1.Text, "0000")
```

```
txtp2.Text = Format(txtp2.Text, "0000")
txtp3.Text = Format(txtp3.Text, "0000")
txtp4.Text = Format(txtp4.Text, "0000")
txtMA.Text = Format(txtMA.Text, "0000")
txtVG.Text = Format(txtVG.Text, "0000")
txtPG.Text = Format(txtPG.Text, "0000")
txtTL.Text = Format(txtTL.Text, "0000")
txtIP.Text = Format(txtIP.Text, "0000")
MSComm1.Output = "#"
Delay (1)
MSComm1.Output = txtOfset.Text + txtp1.Text + txtp2.Text + txtp3.Text +
txtp4.Text + txtMA.Text + txtVG.Text + txtPG.Text + txtTL.Text + txtIP.Text + Direct
Delay (50)
MSComm1.Output = "!"
End Sub

Private Sub cmdSetPara_Click()
txtOfset.Text = Format(txtOfset.Text, "0000")
txtp1.Text = Format(txtp1.Text, "0000")
txtp2.Text = Format(txtp2.Text, "0000")
txtp3.Text = Format(txtp3.Text, "0000")
txtp4.Text = Format(txtp4.Text, "0000")
txtMA.Text = Format(txtMA.Text, "0000")
txtVG.Text = Format(txtVG.Text, "0000")
txtPG.Text = Format(txtPG.Text, "0000")
txtTL.Text = Format(txtTL.Text, "0000")
txtIP.Text = Format(txtIP.Text, "0000")
MSComm1.Output = "#"
Delay (1)
```

```
MSComm1.Output = txtOfset.Text + txtp1.Text + txtp2.Text + txtp3.Text + txtp4.
Text + txtMA.Text + txtVG.Text + txtPG.Text + txtTL.Text + txtIP.Text + Direct
End Sub

Private Sub cmdstart_Click()
Dim aa As String
flag = 1
aa = Chr$(&H2) + "0Q30101000009A" + Chr$(&H3)
MSComm1.Output = aa
Delay (500)
rr = ""
aa = Chr$(&H2) + "0n000000000082" + Chr$(&H3)
MSComm1.Output = aa
rr = MSComm1.Input
End Sub

Private Sub cmdT1_Click()
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "ID" + txtp1.Text + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
End Sub

Private Sub cmdT2_Click()
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
```

```
MSComm1.Output = "ID" + txtp2.Text + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
End Sub

Private Sub cmdT3_Click()
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "ID" + txtp3.Text + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
End Sub

Private Sub cmdT4_Click()
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "ID" + txtp4.Text + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
End Sub

Private Sub cmdUpScrew_Click()
Dim aa As String
flag = 1
aa = Chr$(&H2) + "0Q301040000097" + Chr$(&H3)
```

```
MSComm1.Output = aa
End Sub

Private Sub CmdZero_Click()
Ofset = 0
txtPosNsk.Text = Ofset
End Sub

Private Sub Command1_Click()
Command1.Enabled = False
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "ID" + "-" + Txt_step.Text + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
Ofset = Ofset - Txt_step.Text
If Ofset < (-36000) Then
Ofset = Ofset + 36000
End If
If Ofset < 0 Then
Option1.Value = True
Direct = "z"
End If
Delay (500)
txtPosNsk.Text = Ofset
Command1.Enabled = True
rr = MSComm1.Input
End Sub
```

```
Private Sub Command12_Click()
Dim aa As String
aa = Chr$(&H2) + "0o07000000007A" + Chr$(&H3)
MSComm1.Output = aa
Delay (700)
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "HS" + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
Ofset = 0
txtPosNsk.Text = Ofset
rr = MSComm1.Input
End Sub

Private Sub Command2_Click()
Command2.Enabled = False
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "ID" + Txt_step.Text + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
Ofset = Ofset + Txt_step.Text
If Ofset > 36000 Then
Ofset = Ofset - 36000
End If
If Ofset >= 0 Then
Option2.Value = True
```

```
Direct = "Z"
End If
Delay (500)
txtPosNsk.Text = Ofset
Command2.Enabled = True
rr = MSComm1.Input
End Sub

Private Sub Command3_Click()
Timer3.Enabled = False
Command3.Enabled = False
Delay (50)
MSComm1.PortOpen = False
Delay (1000)
MSComm1.PortOpen = True
Command3.Enabled = True
End Sub

Private Sub Form_Load()
MSComm1.Settings = "9600,n,8,1"
On Error GoTo ex
MSComm1.PortOpen = True
F_para.Visible = False
cmdPara.BackColor = &HFFC0C0
OptClampUp.Value = True
OptVacUp.Value = True
OptPOff.Value = True
OptdriveOff.Value = True
Frame13.Visible = False
M_ON_OFF.Caption = "MOTOR OFF"
```

```
M_ON_OFF.BackColor = &H8080FF
Delay (1500)
cmdRead_Click
GoTo exxx
ex:
MsgBox "CommPort à»Ô´ÍÂÙè¡èÍ¹ ËÃ×ÍäÁèÊÒÁÒÃ¶àª×èÍÁµèÍä´é "
F_para.Visible = False
cmdPara.BackColor = &HFFC0C0
Frame13.Visible = False
'End
exxx:
End Sub

Private Sub Label21_Click()
If Label21.BackColor = &HFF& Then
MSComm1.Output = "a"
Label21.BackColor = &HFF00&
Else
MSComm1.Output = "A"
Label21.BackColor = &HFF&
End If
End Sub
Private Sub Label23_Click()
If Label23.BackColor = &HFF& Then
MSComm1.Output = "d"
Label23.BackColor = &HFF00&
Else
MSComm1.Output = "D"
Label23.BackColor = &HFF&
End If
```

```
End Sub


Private Sub Label24_Click()
If Label24.BackColor = &HFF& Then
MSComm1.Output = "c"
Label24.BackColor = &HFF00&
Else
MSComm1.Output = "C"
Label24.BackColor = &HFF&
End If
End Sub
Private Sub Label25_Click()
If Label25.BackColor = &HFF& Then
MSComm1.Output = "b"
Label25.BackColor = &HFF00&
Else
MSComm1.Output = "B"
Label25.BackColor = &HFF&
End If
End Sub
Private Sub Label26_Click()
If Label26.BackColor = &HFF& Then
MSComm1.Output = "e"
Label26.BackColor = &HFF00&
OptClampDown.Value = True
Else
MSComm1.Output = "E"
Label26.BackColor = &HFF&
OptClampUp.Value = True
End If
```

```
End Sub
Private Sub Label27_Click()
If Label27.BackColor = &HFF& Then
MSComm1.Output = "f"
Label27.BackColor = &HFF00&
OptVacDown.Value = True
Else
MSComm1.Output = "F"
Label27.BackColor = &HFF&
OptVacUp.Value = True
End If
End Sub
Private Sub Label28_Click()
If Label28.BackColor = &HFF& Then
MSComm1.Output = "g"
Label28.BackColor = &HFF00&
OptPOn = True
Else
MSComm1.Output = "G"
Label28.BackColor = &HFF&
OptPOff.Value = True
End If
End Sub
Private Sub Label29_Click()
If Label29.BackColor = &HFF& Then
MSComm1.Output = "h"
Label29.BackColor = &HFF00&
OptdriveOn.Value = True
Else
MSComm1.Output = "H"
```

```
Label29.BackColor = &HFF&
OptdriveOff.Value = True
End If
End Sub
Private Sub Label30_Click()
If Label30.BackColor = &HFF& Then
MSComm1.Output = "i"
Label30.BackColor = &HFF00&
Else
MSComm1.Output = "I"
Label30.BackColor = &HFF&
End If
End Sub

Private Sub M_ON_OFF_Click()
Dim Data_Echo As String
If M_ON_OFF.Caption = "MOTOR OFF" Then
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
MSComm1.Output = "MO" + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
M_ON_OFF.Caption = "MOTOR ON"
M_ON_OFF.BackColor = &H80FF80
Else
MSComm1.Output = "<"
MSComm1.Output = Chr$(&HD)
Delay (20)
```

```
MSComm1.Output = "SV" + Chr$(&HD)
Delay (25)
MSComm1.Output = "@"
rr = MSComm1.Input
M_ON_OFF.Caption = "MOTOR OFF"
M_ON_OFF.BackColor = &H8080FF
End If
End Sub

Private Sub OptClampDown_Click()
MSComm1.Output = "e"
Label26.BackColor = &HFF00&
End Sub

Private Sub OptClampUp_Click()
MSComm1.Output = "E"
Label26.BackColor = &HFF&
End Sub

Private Sub OptdriveOff_Click()
MSComm1.Output = "H"
Label29.BackColor = &HFF&
End Sub

Private Sub OptdriveOn_Click()
MSComm1.Output = "h"
Label29.BackColor = &HFF00&
End Sub

Private Sub Option1_Click()
```

```
Direct = "z"
End Sub

Private Sub Option2_Click()
Direct = "Z"
End Sub

Private Sub OptPOff_Click()
MSComm1.Output = "G"
Label28.BackColor = &HFF&
End Sub

Private Sub OptPOn_Click()
MSComm1.Output = "g"
Label28.BackColor = &HFF00&
End Sub

Private Sub OptVacDown_Click()
MSComm1.Output = "f"
Label27.BackColor = &HFF00&
End Sub

Private Sub OptVacUp_Click()
MSComm1.Output = "F"
Label27.BackColor = &HFF&
End Sub

Private Sub Timer1_Timer()
Dim i As Integer
For i = 0 To 11 Step 1
```

```
ShapeIN(i).FillStyle = 0
ShapeIN(i).FillColor = &HFF&
Next
Timer1.Enabled = False
End Sub

Private Sub Timer2_Timer()
Dim datarr As String
Dim sw_HOME As String
Dim sw_RESET As String
Dim sw_STOP As String
Dim sw_START As String
Dim Clampup As String
Dim ClampDown As String
Dim Vacup As String
Dim VacDown As String
Dim NskReady As String
Dim NskIN As String
Dim Opject As String
Dim LHome As String
Dim LReset As String
Dim LStop As String
Dim LStart As String
Dim SolClamp As String
Dim SolVac As String
Dim Screw As String
Dim OnPump As String
Dim ONrobo As String
MSComm1.Output = "^"
Delay (50)
```

```
datarr = MSComm1.Input
Label13 = datarr
sw_HOME = Left(datarr, 1)
If sw_HOME = "0" Then
ShapeIN(0).FillColor = &HFF00&
Else
ShapeIN(0).FillColor = &HFF&
End If

sw_RESET = Mid(datarr, 2, 1)
If sw_RESET = "0" Then
ShapeIN(3).FillColor = &HFF00&
Else
ShapeIN(3).FillColor = &HFF&
End If

sw_STOP = Mid(datarr, 3, 1)
If sw_STOP = "0" Then
ShapeIN(2).FillColor = &HFF00&
Else
ShapeIN(2).FillColor = &HFF&
End If

sw_STRAT = Mid(datarr, 4, 1)
If sw_STRAT = "0" Then
ShapeIN(1).FillColor = &HFF00&
Else
ShapeIN(1).FillColor = &HFF&
End If
Clampup = Mid(datarr, 5, 1)
```

```
If Clampup = "0" Then
ShapeIN(4).FillColor = &HFF00&
Else
ShapeIN(4).FillColor = &HFF&
End If
ClampDown = Mid(datarr, 6, 1)
If ClampDown = "0" Then
ShapeIN(5).FillColor = &HFF00&
Else
ShapeIN(5).FillColor = &HFF&
End If
Vacup = Mid(datarr, 7, 1)
If Vacup = "0" Then
ShapeIN(6).FillColor = &HFF00&
Else
ShapeIN(6).FillColor = &HFF&
End If
VacDown = Mid(datarr, 8, 1)
If VacDown = "0" Then
ShapeIN(7).FillColor = &HFF00&
Else
ShapeIN(7).FillColor = &HFF&
End If
NskReady = Mid(datarr, 9, 1)
If NskReady = "0" Then
ShapeIN(8).FillColor = &HFF00&
Else
ShapeIN(8).FillColor = &HFF&
End If
NskIN = Mid(datarr, 10, 1)
```

```
If NskIN = "0" Then
ShapeIN(9).FillColor = &HFF00&
Else
ShapeIN(9).FillColor = &HFF&
End If
Opject = Mid(datarr, 11, 1)
If Opject = "0" Then
ShapeIN(10).FillColor = &HFF00&
Else
ShapeIN(10).FillColor = &HFF&
End If
EMG = Mid(datarr, 12, 1)
If EMG = "0" Then
ShapeIN(11).FillColor = &HFF00&
Else
ShapeIN(11).FillColor = &HFF&
End If
LHome = Mid(datarr, 13, 1)
If LHome = "1" Then
Label21.BackColor = &HFF00&
Else
Label21.BackColor = &HFF&
End If
LReset = Mid(datarr, 14, 1)
If LReset = "1" Then
Label25.BackColor = &HFF00&
Else
Label25.BackColor = &HFF&
End If
LStop = Mid(datarr, 15, 1)
```

```
If LStop = "1" Then
Label24.BackColor = &HFF00&
Else
Label24.BackColor = &HFF&
End If
LStart = Mid(datarr, 16, 1)
If LStart = "1" Then
Label23.BackColor = &HFF00&
Else
Label23.BackColor = &HFF&
End If
SolClamp = Mid(datarr, 17, 1)
If SolClamp = "1" Then
Label26.BackColor = &HFF00&
Else
Label26.BackColor = &HFF&
End If
SolVac = Mid(datarr, 18, 1)
If SolVac = "1" Then
Label27.BackColor = &HFF00&
Else
Label27.BackColor = &HFF&
End If
Screw = Mid(datarr, 19, 1)
If Screw = "1" Then
Label28.BackColor = &HFF00&
Else
Label28.BackColor = &HFF&
End If
OnPump = Mid(datarr, 20, 1)
```

```
If OnPump = "1" Then
Label29.BackColor = &HFF00&
Else
Label29.BackColor = &HFF&
End If
ONrobo = Mid(datarr, 21, 1)
If ONrobo = "1" Then
Label30.BackColor = &HFF00&
Else
Label30.BackColor = &HFF&
End If
End Sub

Private Sub Timer3_Timer()
Dim rxt As String
Dim rxz As String
Dim datarr As String
Dim sw_HOME As String
Dim sw_RESET As String
Dim sw_STOP As String
Dim sw_START As String
Dim Clampup As String
Dim ClampDown As String
Dim Vacup As String
Dim VacDown As String
Dim NskReady As String
Dim NskIN As String
Dim Opject As String
Dim LHome As String
Dim LReset As String
```

```
Dim LStop As String
Dim LStart As String
Dim SolClamp As String
Dim SolVac As String
Dim Screw As String
Dim OnPump As String
Dim ONrobo As String
tmtread.Enabled = False
rxt = MSComm1.Input
If rxt = "" Then
Label13 = Label13
Else
Label13 = rxt
Text1.Text = Label13
datarr = Text1.Text
If flag = 0 Then
Frame12.Visible = True
Delay (300)
Frame12.Visible = False
Frame13.Visible = True
Select Case lbplt1.Caption
Case 1
lbplt11.Caption = 2
lbplt22.Caption = 3
lbplt33.Caption = 4
lbplt44.Caption = 1
Case 2
lbplt11.Caption = 3
lbplt22.Caption = 4
lbplt33.Caption = 1
```

```
lbplt44.Caption = 2
Case 3
lbplt11.Caption = 4
lbplt22.Caption = 1
lbplt33.Caption = 2
lbplt44.Caption = 3
Case 4
lbplt11.Caption = 1
lbplt22.Caption = 2
lbplt33.Caption = 3
lbplt44.Caption = 4
End Select
Delay (300)
Frame12.Visible = True
Frame13.Visible = False
Select Case lbplt1.Caption
Case 1
lbplt1.Caption = 2
lbplt2.Caption = 3
lbplt3.Caption = 4
lbplt4.Caption = 1
Case 2
lbplt1.Caption = 3
lbplt2.Caption = 4
lbplt3.Caption = 1
lbplt4.Caption = 2
Case 3
lbplt1.Caption = 4
lbplt2.Caption = 1
lbplt3.Caption = 2
```

```
lbplt4.Caption = 3
Case 4
lbplt1.Caption = 1
lbplt2.Caption = 2
lbplt3.Caption = 3
lbplt4.Caption = 4
End Select
End If
flag = 0
sw_HOME = Left(datarr, 1)
If sw_HOME = "0" Then
ShapeIN(0).FillColor = &HFF00&
Else
ShapeIN(0).FillColor = &HFF&
End If

sw_RESET = Mid(datarr, 2, 1)
If sw_RESET = "0" Then
ShapeIN(3).FillColor = &HFF00&
Else
ShapeIN(3).FillColor = &HFF&
End If

sw_STOP = Mid(datarr, 3, 1)
If sw_STOP = "0" Then
ShapeIN(2).FillColor = &HFF00&
Else
ShapeIN(2).FillColor = &HFF&
End If
```

```
sw_STRAT = Mid(datarr, 4, 1)
If sw_STRAT = "0" Then
ShapeIN(1).FillColor = &HFF00&
Else
ShapeIN(1).FillColor = &HFF&
End If
Clampup = Mid(datarr, 5, 1)
If Clampup = "0" Then
ShapeIN(4).FillColor = &HFF00&
Else
ShapeIN(4).FillColor = &HFF&
End If
ClampDown = Mid(datarr, 6, 1)
If ClampDown = "0" Then
ShapeIN(5).FillColor = &HFF00&
Else
ShapeIN(5).FillColor = &HFF&
End If
Vacup = Mid(datarr, 7, 1)
If Vacup = "0" Then
ShapeIN(6).FillColor = &HFF00&
Else
ShapeIN(6).FillColor = &HFF&
End If
VacDown = Mid(datarr, 8, 1)
If VacDown = "0" Then
ShapeIN(7).FillColor = &HFF00&
Else
ShapeIN(7).FillColor = &HFF&
End If
```

```
NskReady = Mid(datarr, 9, 1)
If NskReady = "0" Then
ShapeIN(8).FillColor = &HFF00&
Else
ShapeIN(8).FillColor = &HFF&
End If
NskIN = Mid(datarr, 10, 1)
If NskIN = "0" Then
ShapeIN(9).FillColor = &HFF00&
Else
ShapeIN(9).FillColor = &HFF&
End If
Opject = Mid(datarr, 11, 1)
If Opject = "0" Then
ShapeIN(10).FillColor = &HFF00&
Else
ShapeIN(10).FillColor = &HFF&
End If
EMG = Mid(datarr, 12, 1)
If EMG = "0" Then
ShapeIN(11).FillColor = &HFF00&
Else
ShapeIN(11).FillColor = &HFF&
End If
LHome = Mid(datarr, 13, 1)
If LHome = "1" Then
Label21.BackColor = &HFF00&
Else
Label21.BackColor = &HFF&
End If
```

```
LReset = Mid(datarr, 14, 1)
If LReset = "1" Then
Label25.BackColor = &HFF00&
Else
Label25.BackColor = &HFF&
End If
LStop = Mid(datarr, 15, 1)
If LStop = "1" Then
Label24.BackColor = &HFF00&
Else
Label24.BackColor = &HFF&
End If
LStart = Mid(datarr, 16, 1)
If LStart = "1" Then
Label23.BackColor = &HFF00&
Else
Label23.BackColor = &HFF&
End If
SolClamp = Mid(datarr, 17, 1)
If SolClamp = "1" Then
Label26.BackColor = &HFF00&
Else
Label26.BackColor = &HFF&
End If
SolVac = Mid(datarr, 18, 1)
If SolVac = "1" Then
Label27.BackColor = &HFF00&
Else
Label27.BackColor = &HFF&
End If
```

```
Screw = Mid(datarr, 19, 1)
If Screw = "1" Then
Label28.BackColor = &HFF00&
Else
Label28.BackColor = &HFF&
End If
OnPump = Mid(datarr, 20, 1)
If OnPump = "1" Then
Label29.BackColor = &HFF00&
Else
Label29.BackColor = &HFF&
End If
End If
End Sub

Private Sub tmtfocus_Timer()
Delay (500)
cmdOpen.BackColor = &HFF&
Delay (500)
cmdOpen.BackColor = &H8000000F
End Sub

Private Sub tmtread_Timer()
Dim Optoint As String
Datansk = MSComm1.Input
txtOfset.Text = Left(Datansk, 4)
txtp1.Text = Mid(Datansk, 5, 4)
txtp2.Text = Mid(Datansk, 9, 4)
txtp3.Text = Mid(Datansk, 13, 4)
txtp4.Text = Mid(Datansk, 17, 4)
```

```
txtMA.Text = Mid(Datansk, 21, 4)
txtVG.Text = Mid(Datansk, 25, 4)
txtPG.Text = Mid(Datansk, 29, 4)
txtTL.Text = Mid(Datansk, 33, 4)
txtIP.Text = Mid(Datansk, 37, 4)
Optoint = Right(Datansk, 1)
If Optoint = "z" Then
Option1.Value = True
Else
Option2.Value = True
End If
tmtread.Enabled = False
Timer3.Enabled = True
End Sub
```

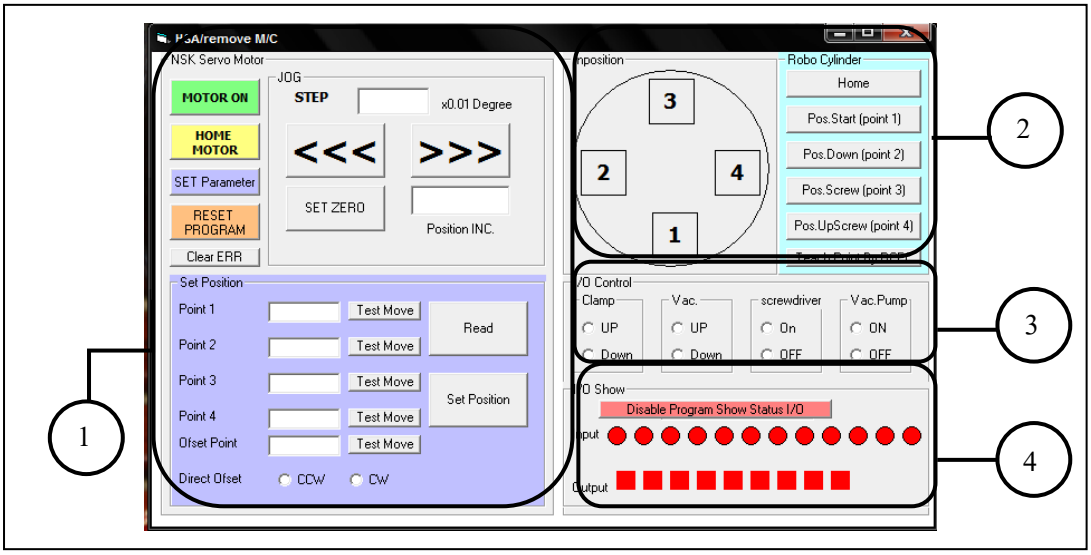
ภาคผนวก ค

คู่มือการใช้งานเครื่องต้นแบบ

ค. 1 การติดตั้งเครื่องจักรต้นแบบก่อนการใช้งาน

ก่อนการใช้งานเครื่องจักร ควรปรับตั้งค่าตำแหน่งให้ตรงตำแหน่งการทำงาน และตัวแปรต่าง ๆ ในการทำงานให้ถูกต้องเหมาะสม สำหรับเครื่องจักรต้นแบบการปรับตั้งค่ามีขั้นตอนดังนี้

เปิดโปรแกรม Setup machine ที่เครื่องคอมพิวเตอร์ และเชื่อมต่อโปรแกรมกับบอร์ดไมโครคอนโทรลเลอร์ โดยเสียบสาย USB จากบอร์ดไมโครคอนโทรลเลอร์ของเครื่องจักรมาที่เครื่องคอมพิวเตอร์ เมื่อมีการเชื่อมต่อจะปรากฏหน้าต่างของโปรแกรมดังรูปที่ ค.1



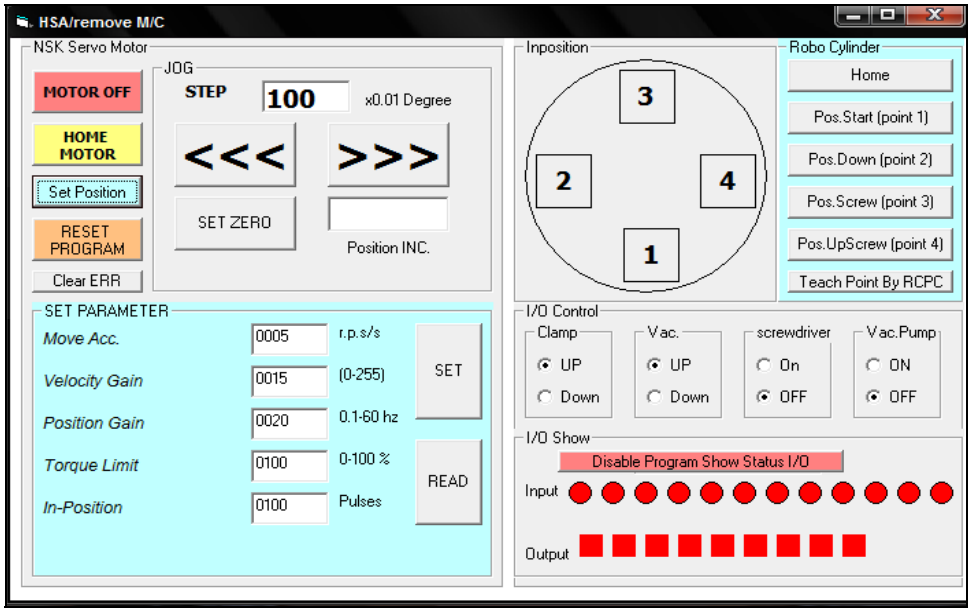
รูปที่ ค. 1 หน้าต่างโปรแกรม Setup machine

1. ปรับค่าตำแหน่ง และตัวแปรของชุดการเคลื่อนที่แบบหมุน

1.1 กดปุ่ม MOTOR ON เปิดการทำงานของชุดการเคลื่อนที่แบบหมุน

1.2 กดปุ่ม HOME MOTOR ตั้งค่าตัวแปรของชุดการเคลื่อนที่แบบหมุน โดยกดที่ปุ่ม SET Parameter หน้าต่างของโปรแกรมในส่วนของชุดการเคลื่อนที่แบบหมุนจะเปลี่ยนไปที่หน้าการตั้งค่าตัวแปร โดยค่าที่เลือกใช้ เลือกค่าจากคู่มือของอุปกรณ์ เลือกจากการใช้การเชื่อมต่อแบบ RS-232 โดยจากการทดสอบการทำงานของเรื่องจักร ค่าตัวแปรที่ให้การทำงานดีที่สุดมีค่าดังรูปที่ ค.2 เมื่อใส่ค่าแล้วให้กด SET และกด READ เพื่อส่งข้อมูลไปที่บอร์ดไมโครคอนโทรลเลอร์

1.3 ปรับตำแหน่งของชุดจับยึดชิ้นงานให้ตรงตำแหน่งทำงาน โดยกดที่ปุ่ม Set position ซึ่งตำแหน่งทำงานอ้างอิงจากชุดกดฮาร์ดดิสไดรฟ์ และ flex โดยในการหาตำแหน่งแรกในการทำงาน เมื่อกด Home motor ชุดการเคลื่อนที่แบบหมุนจะเคลื่อนที่ทำให้ชุดจับยึดชิ้นงาน



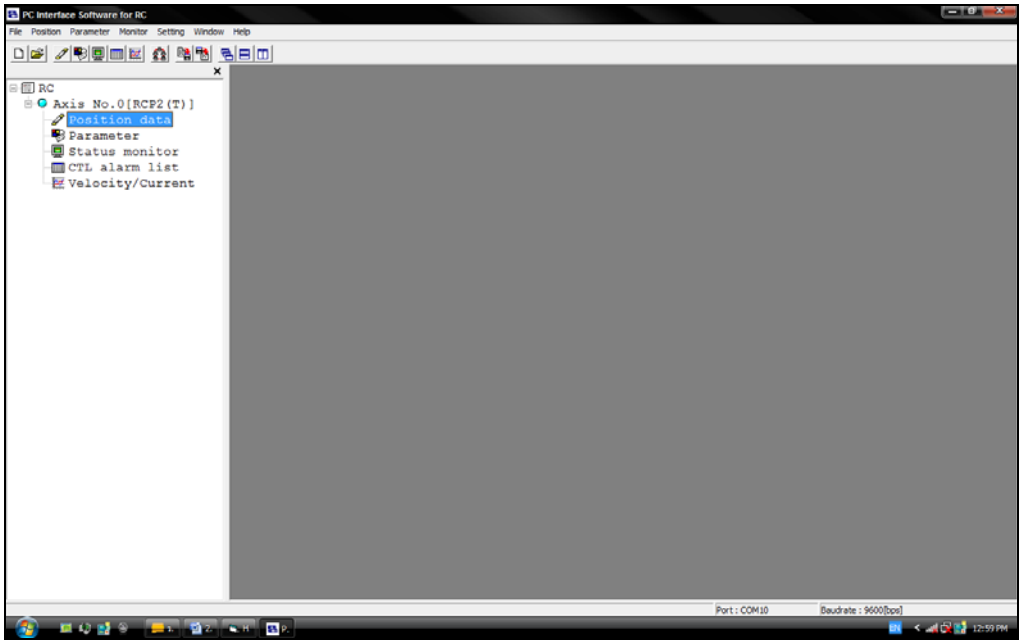
รูปที่ ค. 2 หน้าต่างโปรแกรมในส่วนการตั้งค่าตัวแปรของชุดการเคลื่อนที่แบบหมุน

เคลื่อนที่ไปที่ตำหแน่งใด ๆ จากนั้นเลือกตัวจับยึดชิ้นงาน 1 ชุด เพื่อเป็นตัวอ้างอิงในการหาระยะที่จะไปที่ตำแหน่งแรกในการทำงาน โดยการเคลื่อนที่ไปที่ตำแหน่งทำงาน ของตำแหน่งทำงานที่ 1 ใช้การเคลื่อนที่โดยกดปุ่ม JOG ซึ่งสามารถเลือกค่าในการเคลื่อนที่ได้ เพื่อให้ได้ตำแหน่งที่ตรงกับชุดกดฮาร์ดดิสก์ไดรฟ์ เมื่อได้ค่าที่ตำแหน่งแรกแล้วให้กด set zero และนำค่าที่ช่อง position INC. ไปใส่ที่ช่อง offset point จากนั้นที่ตำแหน่งต่อไป ให้กด jog เพื่อเคลื่อนที่ไปหาตำแหน่งที่ 2 3 และ 4 ตามลำดับ โดยทำเช่นเดียวกันในทุกตำแหน่ง ซึ่งค่าที่ใส่ที่ช่อง Point 1 2 3 และ 4 จะมีค่าเท่ากัน เนื่องจากในการเคลื่อนที่ของชุดการเคลื่อนที่แบบหมุน จะให้การเคลื่อนที่แต่ละครั้งเท่ากับ 90 องศา ก็คือ เท่ากับ 9000 x 0.01 degree เมื่อใส่ค่าเรียบร้อยแล้วให้กด set position และ Read ตามลำดับ

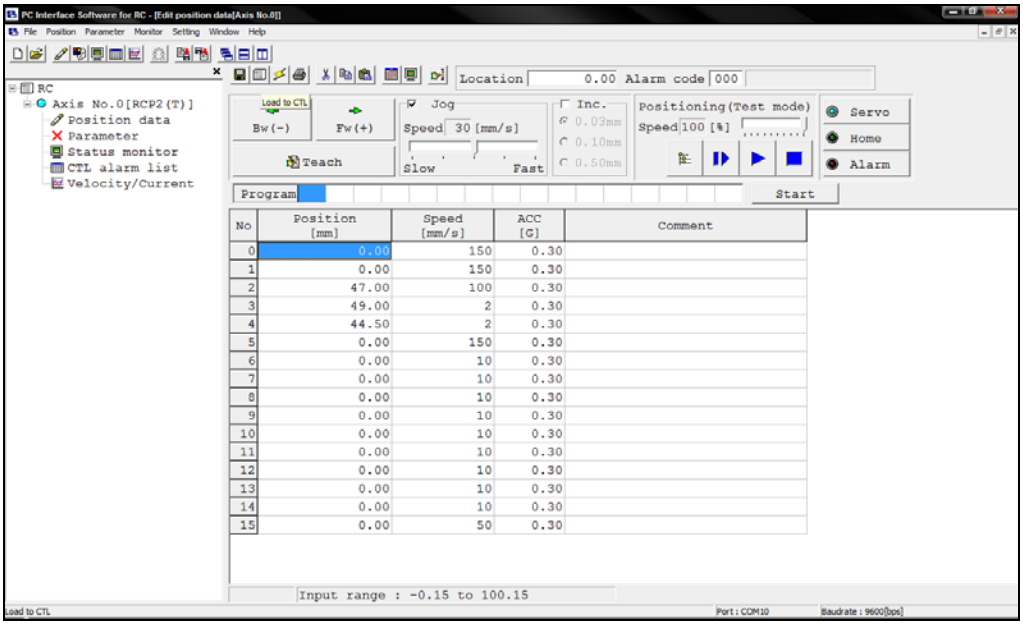
1. ปรับค่าตำแหน่ง และตัวแปรของชุดกระบอกสูบไฟฟ้าของชุดไขควง

สำหรับชุดกระบอกสูบไฟฟ้าของชุดไขควง ปรับค่าต่าง ๆ ที่โปรแกรม RCPC ของอุปกรณ์ โดยปรับตั้งค่าดังนี้

2. กดปุ่ม Teach point by RCPC เพื่อปรับค่าตำแหน่งและตัวแปร เมื่อเข้าสู่โปรแกรม จะได้ลักษณะหน้าต่างการตั้งค่า ดังรูปที่ ค.3 และคลิกเครื่องหมายบวกที่คำสั่ง Axis NO.0 [RCP2(T)] และให้ ดับเบิ้ลคลิกที่ position data เพื่อเข้าสู่การปรับตั้งค่าตำแหน่งของอุปกรณ์ เมื่อปรับตั้งค่าตำแหน่งและตัวแปรของอุปกรณ์เรียบร้อยแล้ว ให้กดปุ่ม Load to CTL เพื่อส่งค่าไปยังกล่องควบคุมของชุดกระบอกสูบไฟฟ้า



รูปที่ ค. 3 หน้าต่างการเลือกการตั้งค่าตำแหน่ง และตัวแปรของโปรแกรม RCPC



รูปที่ ค. 4 หน้าต่างการใส่ค่าตำแหน่ง ตัวแปร และการส่งข้อมูลไปชุดกล่องควบคุม

การเคลื่อนที่ของชุดกระบอกสูบไฟฟ้า ปรับให้เคลื่อนที่ได้ 5 ตำแหน่ง คือ

ตำแหน่งที่ 1 (NO.0) ตำแหน่ง Home คือ ตำแหน่งบนสุดในการเคลื่อนที่ของกระบอกสูบ

ตำแหน่งที่ 2 (NO.1) ตำแหน่งที่เคลื่อนที่จากตำแหน่ง Home ลงมา โดยกำหนดระยะไม่ให้เข้าใกล้ฮาร์ดดิสก์ไดรฟ์มาก หรือไม่ต้องกำหนดการเคลื่อนที่ คือใส่ค่าเท่ากับศูนย์ สำหรับเครื่องจักรต้นแบบที่ตำแหน่งนี้ กำหนดใส่ค่าเท่ากับศูนย์ เนื่องจากเมื่อทดสอบการเคลื่อนที่ที่ตำแหน่งต่าง ๆ และจับเวลา ที่ตำแหน่งนี้ให้เวลาในการทำงานรวมทั้งกระบวนการดีที่สุด เพราะไม่ต้องเสียเวลาในการเคลื่อนที่มาอีกหนึ่งระยะ

ตำแหน่งที่ 3 (NO.2) เคลื่อนที่จากตำแหน่ง 2 ลงมา ระยะของตำแหน่งนี้จะเข้าใกล้ฮาร์ดดิสก์ไดรฟ์

ตำแหน่งที่ 4 (NO.3) เป็นตำแหน่งที่เคลื่อนที่จากตำแหน่ง 3 ลงมา คือ ดอกสกรูของไขควงถึงตัวสกรูที่จะทำการขัน ดังนั้นในการเคลื่อนที่มาที่ตำแหน่งนี้ไม่ควรกำหนดความเร็วมีค่ามาก เพื่อป้องกันไม่ให้สกรูเสียหาย และฮาร์ดดิสก์ไดรฟ์ได้รับแรงกระแทก เมื่อถึงที่ตำแหน่งนี้ไขควงจะเปิดการทำงาน และทำการขันสกรูออก

ตำแหน่งที่ 5 (NO.4) เคลื่อนที่จากตำแหน่งที่ 4 กลับขึ้นไปที่ระยะใกล้ตำแหน่งบนสุด

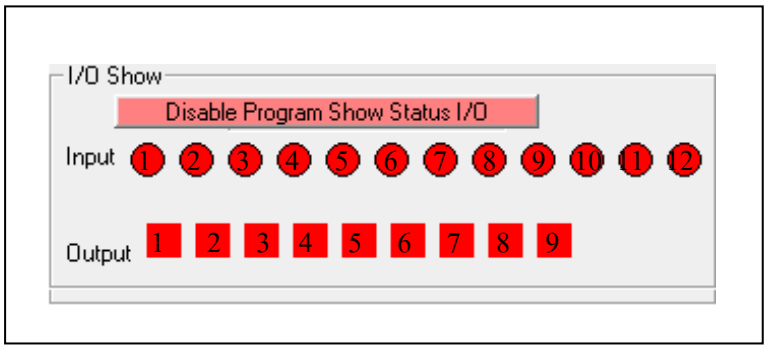
ในการปรับค่าของตำแหน่ง ให้นำเมาส์ไปคลิกที่ช่อง position ของ NO.ที่ต้องการตั้งค่า และปรับตำแหน่งขึ้นลงโดยกดที่ปุ่ม FW+ หรือ BW- ซึ่งสามารถกำหนดการเคลื่อนที่ได้ที่ปุ่ม JOG เมื่อได้ตำแหน่งที่ต้องการแล้ว ค่าตำแหน่งจะแสดงที่ช่อง Location ให้กดปุ่ม Teach ค่าที่ต้องการจะไปแสดงที่ตำแหน่งที่เลือกไว้ ซึ่งการเคลื่อนที่จากตำแหน่งหนึ่ง ไปหาอีกตำแหน่งหนึ่งสามารถกำหนดความเร็ว และความเร่งในการเคลื่อนที่ได้ โดยค่าความเร่งสำหรับชุดกระบอกสูบไฟฟ้าจะมีคุณสมบัติให้กำหนดตามการใช้งาน สามารถดูได้จากรายละเอียดของอุปกรณ์ เมื่อกำหนดตำแหน่งในการเคลื่อนที่ครบแล้ว สามารถทดสอบการเคลื่อนที่ทั้งหมด โดยกดปุ่มทดสอบการทำงาน

3. I/O control

แผงควบคุมนี้ ใช้สำหรับทดสอบการทำงานของชุดกระบอกสูบลมของชุดกดฮาร์ดดิสก์ไดรฟ์ และ flex ชุดท่อสุญญากาศ การทดสอบการทำงานของไขควง และปั๊ม

4. I/O show

แสดงไฟสถานะของบอร์ดไมโครคอนโทรลเลอร์ เมื่อมีการเชื่อมต่อกับบอร์ดไมโครคอนโทรลเลอร์ และมีการสั่งงานอุปกรณ์ใด ๆ แสดงรายละเอียดของไฟสถานะแต่ละตัวดังนี้



รูปที่ ค. 5 ไฟสถานะของบอร์ดไมโครคอนโทรลเลอร์

4.1 ไฟสถานะ Input

หมายเลข 1 ไฟแสดงสถานะของการ Home เครื่องจักร คือเมื่อมีการกดปุ่ม HOME เพื่อสั่งการทำงานทั้งระบบให้เข้าสู่ตำแหน่งเริ่มต้น ไฟตำแหน่งนี้จะติด

หมายเลข 2 ไฟแสดงสถานะของปุ่ม Start เมื่อมีการกดปุ่ม start ไฟตำแหน่งนี้จะติด

หมายเลข 3 ไฟแสดงสถานะของปุ่มStop

หมายเลข 4 ไฟแสดงสถานะของปุ่ม Reset

หมายเลข 5 ไฟแสดงสถานะของ sensor ของชุดกระบอกสูบลมยึดชุดกดฮาร์ดดิสก์ไดรฟ์ และ Flex ไฟติดเมื่อชุดกระบอกสูบลมอยู่ตำแหน่งบน

หมายเลข 6 ไฟแสดงสถานะ sensor ของชุดกระบอกสูบลมยึดชุดกดฮาร์ดดิสก์ไดรฟ์ และ Flex ไฟติดเมื่อชุดกระบอกสูบลมอยู่ตำแหน่งล่าง

หมายเลข 7 ไฟแสดงสถานะของ sensor ของชุดกระบอกสูบลมยึดชุดท่อสุญญากาศ ไฟติดเมื่อชุดกระบอกสูบลมอยู่ตำแหน่งบน

หมายเลข 8 ไฟแสดงสถานะของ sensor ของชุดกระบอกสูบลมยึดชุดท่อสุญญากาศ ไฟติดเมื่อชุดกระบอกสูบลมอยู่ตำแหน่งล่าง

หมายเลข 9 ไฟแสดงสถานะของชุดการเคลื่อนที่แบบหมุน เมื่อมีการเปิดเครื่อง และ Home ทั้งระบบให้เข้าสู่ตำแหน่งเริ่มต้น ไฟที่ตำแหน่งนี้จะติดตลอดเพื่อแสดงว่าชุดการเคลื่อนที่แบบหมุน พร้อมทำงานตลอด ไฟจะดับเมื่อเกิดข้อผิดพลาดขึ้นกับชุดการเคลื่อนที่แบบหมุน

หมายเลข 10 ไฟแสดงสถานะการทำงานของชุดการเคลื่อนที่แบบหมุนคือ เมื่อมีการสั่งงานให้เคลื่อนที่ไปที่ตำแหน่งทำงาน และมีการเคลื่อนที่ไปตำแหน่งที่กำหนด ไฟจะกระพริบและบอกให้ทราบว่าสามารถทำงานขั้นตอนต่อไปได้

หมายเลข 11 ไฟแสดงสถานะของ sensor ตรวจสอบชิ้นงาน คือ เมื่อมีการเคลื่อนที่ชุดจับยึดชิ้นงานเข้ามาที่ตำแหน่งทำงาน ที่ตำแหน่งนี้จะมี sersor ตรวจสอบว่ามีชิ้นงานหรือไม่ ถ้ามีไฟจะติด และจะทำงานขั้นตอนต่อไป แต่ถ้าไม่มีชิ้นงาน ก็จะข้ามชุดจับยึดชิ้นงานนี้เพื่อเคลื่อนที่ชุดจับยึดชิ้นงานชุดต่อไปเข้ามาที่ตำแหน่งทำงาน

หมายเลข 12 ไฟแสดงสถานะของ sensor ตรวจสอบตำแหน่งของชุดกระบอกสูบไฟฟ้ายึดไขควง ที่ตำแหน่งนี้สถานะของไฟจะติดตลอด คือเมื่อชุดกระบอกสูบไฟฟ้ายึดไขควงมีการเคลื่อนที่ จะมีการตรวจสอบตลอดว่าเคลื่อนที่ลงมาต่ำกว่าตำแหน่งที่กำหนดไว้หรือไม่ และถ้าไฟที่สถานะนี้ดับแสดงว่ามีการเคลื่อนที่ลงมาต่ำกว่าตำแหน่งที่กำหนด

4.2 ไฟสถานะ Output

หมายเลข 1 ไฟแสดงสถานะของปุ่ม HOME เมื่อมีการกดปุ่มสั่งงาน

หมายเลข 2 ไฟแสดงสถานะของปุ่ม Start เมื่อมีการกดปุ่มไฟจะติด

หมายเลข 3 ไฟแสดงสถานะของของปุ่ม Stop เมื่อมีการกดปุ่มไฟจะติด

หมายเลข 4 ไฟแสดงสถานะของของปุ่ม Reset เมื่อมีการกดปุ่มไฟจะติด

หมายเลข 5 ไฟแสดงสถานะของกระบอกสูบลมของชุดกดฮาร์ดดิกส์ไดรฟ์ และกด flex เมื่อกดสั่งงานให้เคลื่อนที่ลงมาไฟจะติด

หมายเลข 6 ไฟแสดงสถานะของกระบอกสูบลมของชุดท่อสุญญากาศเมื่อกดสั่งงานให้เคลื่อนที่ลงมาไฟจะติด

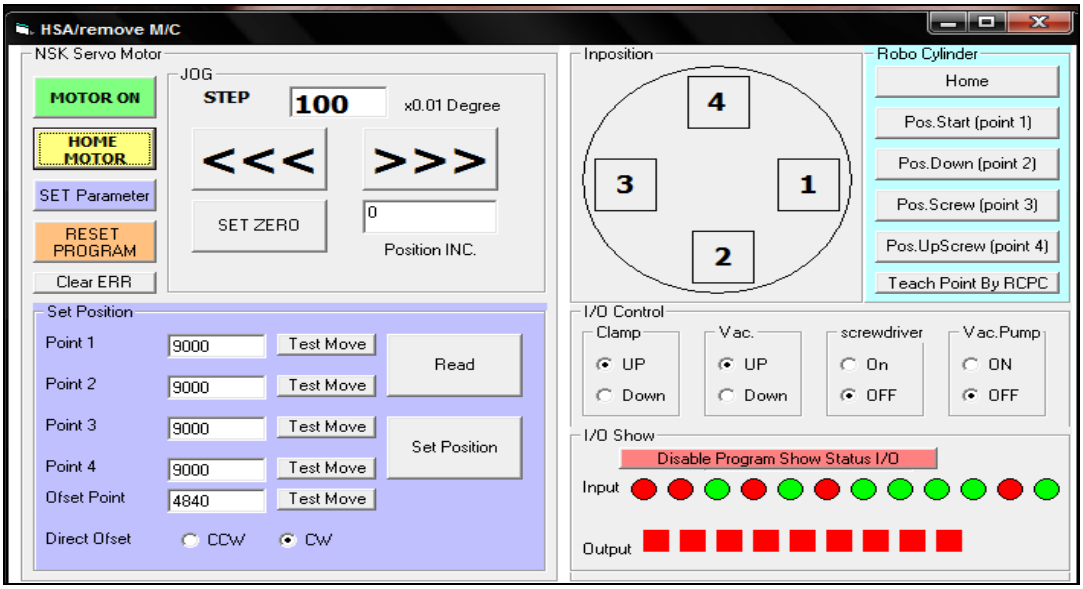
หมายเลข 7 ไฟแสดงสถานะของไขควง เมื่อมีการเปิดการทำงานของไขควงไฟจะติด

หมายเลข 8 ไฟแสดงสถานะของปั๊ม เมื่อมีการเปิดการทำงานไฟจะติด

หมายเลข 9 ไม่มีการใช้งานไฟในตำแหน่งนี้

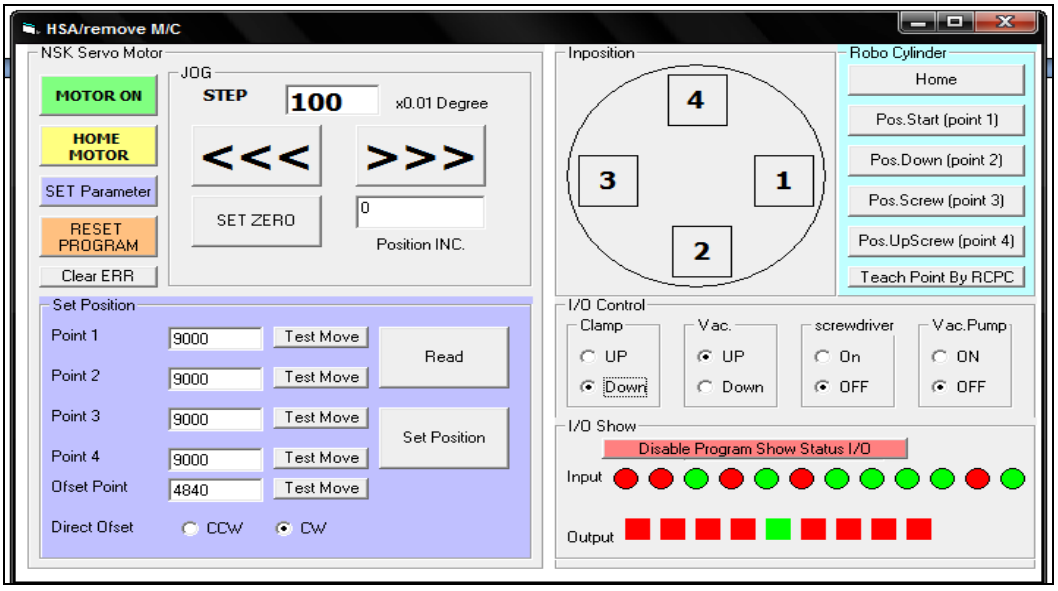
แสดงตัวอย่างเมื่อกดปุ่มสั่งงานคำสั่งต่าง ๆ ไฟสถานะจะแสดง ดังนี้

1. กดปุ่ม HOME MOTOR ไฟแสดงที่บอร์ดไมโครคอนโทรลเลอร์จะแสดงดังรูป



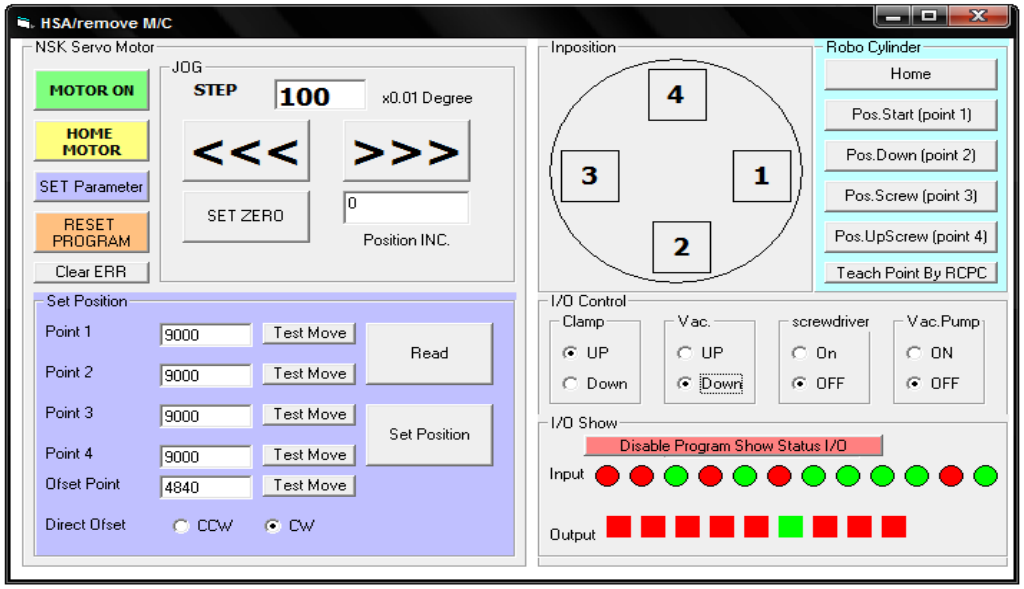
รูปที่ ค. 6 หน้าต่างแสดงไฟสถานะเมื่อกดปุ่ม HOME MOTOR

2. กด Clamp down ไฟติดดังรูป



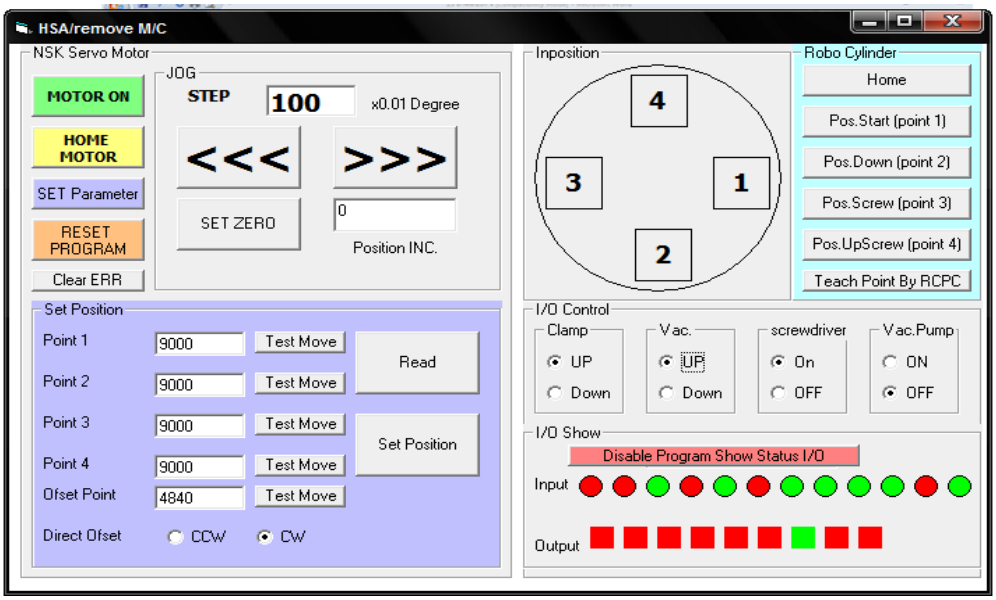
รูปที่ ค. 7 หน้าต่างแสดงไฟสถานะเมื่อกด Clamp down

3. กด Vacuum down ไฟติดดังรูป



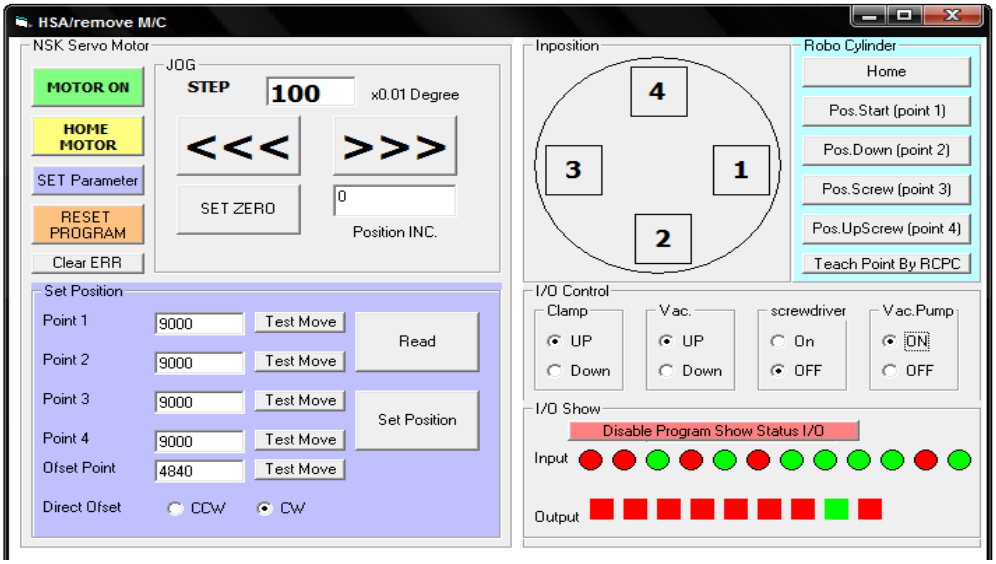
รูปที่ ค. 8 หน้าต่างแสดงไฟสถานะเมื่อกด Vacuum down

4. กดเปิดสกรู ไฟติดดังรูป



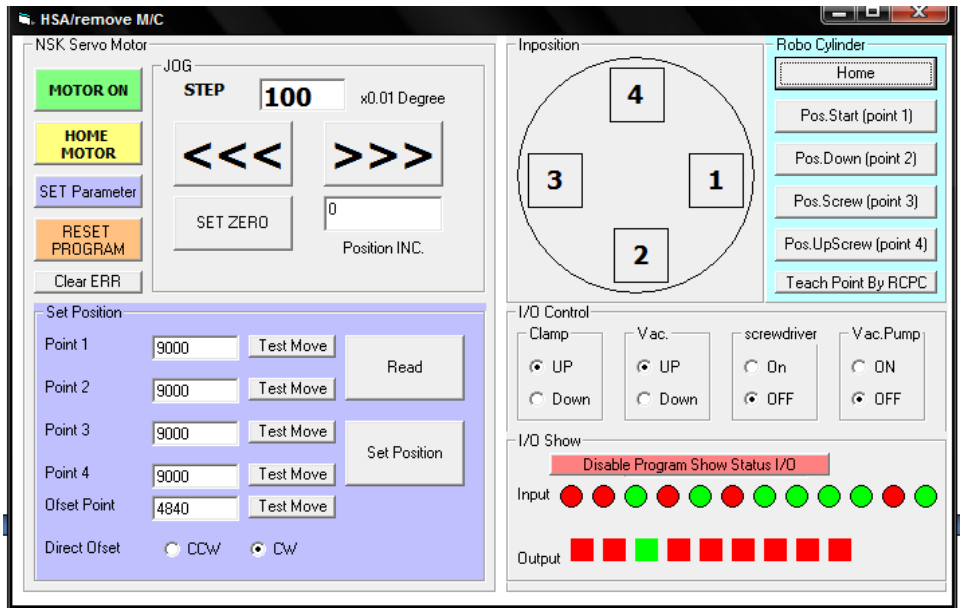
รูปที่ ค. 9 หน้าต่างแสดงไฟสถานะเมื่อกดเปิดสกรู

5. กดเปิดปั๊ม ไฟติดดังรูป



รูปที่ ค. 10 หน้าต่างแสดงไฟสถานะเมื่อกดเปิดปั๊ม

6. กดปุ่ม HOME ของชุดกระบอกสูบไฟฟ้า ไฟติดดังรูป



รูปที่ ค. 11 หน้าต่างแสดงไฟสถานะเมื่อกดปุ่ม HOME ของชุดกระบอกสูบไฟฟ้า

5. ปรับตั้งค่าลูกสูบยึดไขควง จะต้องปรับตำแหน่งให้หัวของดอกสกรูตรงตำแหน่งกับสกรู โดยปรับที่ชิ้นส่วนขาตั้ง และชิ้นส่วนที่ใช้ยึดไขควง เนื่องจากออกแบบให้สามารถปรับได้

6. ปรับตำแหน่งชุดท่อสุญญากาศให้ตรงกับดอกสกรู เนื่องจากใช้ชุดท่อสุญญากาศเป็นแนวทางในการเคลื่อนที่ของชุดไขควง โดยปรับที่ชิ้นส่วนขาตั้ง และชิ้นส่วนที่ใช้ยึดท่อสุญญากาศ จากนั้นทดสอบการเคลื่อนที่ขึ้นลงของชุดกดฮาร์ดดิสก์ไดรฟ์ และกด Flex และชุดท่อสุญญากาศ ว่าเคลื่อนที่ขึ้น และลงสุดหรือไม่ โดยกดที่แผงควบคุม I/O control ที่แผงควบคุม I/O control สามารถทดสอบการทำงานของชุดไขควงได้

ค. 2 ขั้นตอนการใช้งานเครื่องจักรต้นแบบ

เมื่อตั้งค่าตำแหน่ง และตัวแปรของอุปกรณ์ในเครื่องจักรต้นแบบเรียบร้อยแล้ว เริ่มใช้งานเครื่องจักรต้นแบบดังนี้

1. กดปุ่ม Home (ปุ่มสีส้ม) เพื่อให้อุปกรณ์ทุกชิ้นเคลื่อนที่ไปในตำแหน่งเริ่มต้น เมื่อตรวจสอบว่าไม่มีข้อผิดพลาด ปุ่มสีส้มจะติดขึ้น พนักงานวางฮาร์ดดิสก์ไดรฟ์ ที่ตัวจับยึดชิ้นงาน

2. กดปุ่ม start (ปุ่มสีเขียว) ชิ้นงานเคลื่อนที่ไปที่ตำแหน่งทำงาน และดำเนินการตามกระบวนการที่ตั้งไว้จนจบกระบวนการ ถ้าไม่กดปุ่ม start ต่อ เครื่องจะหยุดทำงานหลังจากกระบวนการแรกเสร็จ 60 วินาที และให้กดปุ่ม stop (ปุ่มสีแดง) เพื่อหยุดการทำงาน แต่ถ้ากดปุ่ม start ต่อ แล้วไม่ได้ใส่ฮาร์ดดิกส์ไดรฟ์ไว้ที่ชุดจับยึดชิ้นงานตัวต่อไป sensor จะตรวจสอบว่าไม่มีชิ้นงาน ชุดการเคลื่อนที่แบบหมุนก็จะเคลื่อนที่ไปเรื่อยๆจนกว่าจะกดปุ่ม stop ชุดการเคลื่อนที่แบบหมุนถึงจะหยุด จากนั้นให้ใส่ชิ้นงานและกดปุ่ม start เริ่มการกระบวนการทำงานใหม่ และเมื่อมีการเกิดข้อผิดพลาดในกระบวนการขึ้น ให้กดปุ่ม Reset (ปุ่มสีเหลือง) เพื่อให้หยุดการทำงาน และกดปุ่ม HOME เพื่อเข้าสู่ตำแหน่งเริ่มต้นในการทำงานใหม่

ภาคผนวก ง

รายชื่อบทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

บทความวิชาการที่ได้รับการตีพิมพ์เผยแพร่

Kaewkamsai, P., Nerakae, K., and Chamniprasart, K. (2010). **Development of HSA/VCM Automatic Removal Machine.** Proceedings of International Conference on Data Storage Technology (DST-CON) Bangkok, Thailand (Submit).

อยู่ในระหว่างรอการนำเสนอผลงานทางวิชาการโดยได้รับการตอบรับแล้ว

[DST-CON2010] Acceptance Notification

From: **DSTCON 2010** (dstcon.kmitl@gmail.com)
Sent: Friday, April 23, 2010 8:44:48 AM
To: anna_nicha@hotmail.com; air_me10@hotmail.com; kontorn@sut.ac.th
Cc: DST-CON2010 (dstcon2010@gmail.com)

Dear Kaewkamsai Phachirarat,Nerakae Krissana,Chamniprasart Kontorn;
Paper ID: 20100084
Paper Title: Development of HSA/VCM automatic removal machine

Congratulations!
The DST-CON2010 Technical Program Committee has completed the review process, and we are pleased to inform you that the paper named above has been ACCEPTED for presentation. The said paper also will be published in the conference Proceedings.
Included at the end of this e-mail message are the reviewers' comments on your paper. Please revise your paper according to reviewers' comments.
BE NOTED! The final manuscript MUST BE LIMITED to only 4 pages. The final manuscript and copyright form, MUST be uploaded to the http://www.dst-con2010.org web site ONLY by Monday April 26, 2010. Violating this note may cause your paper being unpublished.

Again, congratulations; we are looking forward to seeing you in DST-CON2010, BITEC, Bangkok.

Best regards,
S.Choomchuay
Technical Program Committee Chair, DST-CON2010

==========

Reviewers' Comments:

(1).	Strength of the paper
	#good illustration
(2).	Weakness of the paper
	# Grammatical errorneous # Lack of details of microcontroller impleme Worl flow diagram should be included.

Development of HSA/VCM automatic removal machine

Phachirarat Kaewkamsai, Krissana Nerakae, Kontorn Chamniprasart
School of Mechanical Engineering, Institute of Engineering, Suranaree University of Technology,
Nakhon Ratchasima, 30000, Thailand
anna_nicha@hotmail.com, air_me10@hotmail.com, kontorn@sut.ac.th

Abstract— **This research represents the prototype machine that uses to produce and assemble Hard Disk Drive in part of the Head Stack Assembly (HSA) removal and control set of the Voice Coil Motor (VCM) movement. Due to original production, Hard Disk Drive were assembled completely. It was tested and found the error of HSA movement. We solved them by manual removal. We found that they occurred the problem of cleanliness and damage of other components. This research requires to create the prototype machine for Head Stack Assembly (HSA) and Voice Coil Motor (VCM) automatic removal that control by Microcontroller. This research has four parts. The first part is the design of screw removal system. The second part is the design of screw suck system. The third part is the contaminant in production of automatic removal and the last one is the design of work procedure. Researcher expects HSA/VCM automatic removal machine will reduce error and contamination, time-consuming of the human work process, improve quality of the product and reduce the production capital. The test result and discussions are included in this report.**

Keywords-component —HSA, VCM, Hard Disk Drive

I. INTRODUCTION

Manufacturing Technology of hard disk drive is more concern with automation system which can improve manufacturing process efficiency and reduce manufacturing cost by reducing cycle time that come from
manual operations.

II. METHOD

HSA / VCM removal process of Hard Disk Drive 2.5 inch starting from removing screw at base of Hard Disk Drive to detach HSA and Base. (as in Fig.1)

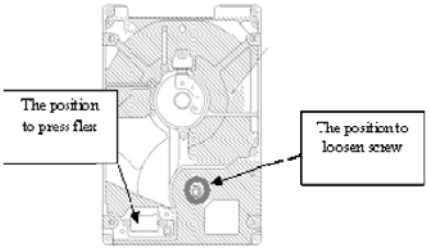


Fig.1 shows the screw position to be loosen and to press Flex

A. Design of machine and its mechanism

Machine design is [1] designed by SolidWork 2008 by based on an idea of designing equipment for disassembly HAS/VCM in 2.5 inches hard disk drive. The equipment will be designed to place 4 pieces of part.Parts will be rotated every 90 degree.

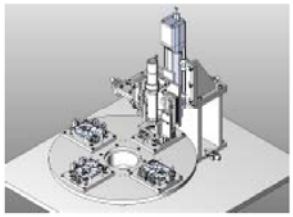


Fig.2 shows the model prototype machine designed by Solidwork software

Fixture is designed for Hard Disk Drive and HSA/VCM which is referred from manual fixture for HSA/VCM disassemble process works by worker as figure 3. Difference between manual and machine process is plate for placing hard disk drive which is changed to be an absorber to substitute manual rotation. The absorber can reduce HSA/VCM destroyed by direct impact from hard disk drive due air fixture force. There is fixture to support HSA when it falls off from disk to prevent disk destroyed while screw loose. As figure 5

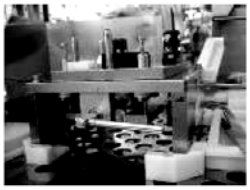


Fig.3 shows fixtures in manual HSA/VCM removal process

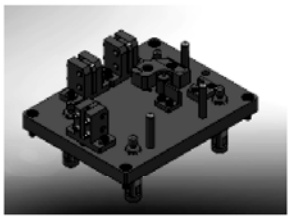


Fig.4 shows the designed fixtures

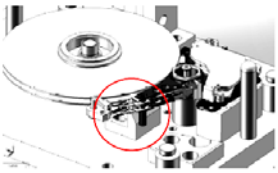
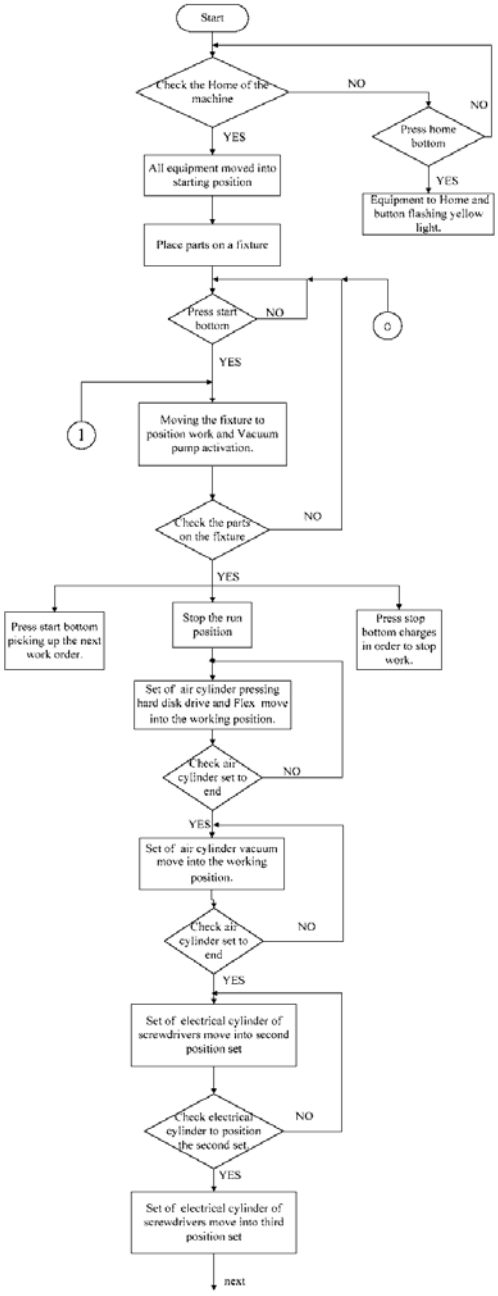


Fig.5 shows HSA removal from disk using ramp

Manual loosing screw process is substituted by electrical screw which is commanded by control box. Torque of loosing screw is about 1.5-2.1 kgf.cm. The torque is calculated from tightening screw torque which is 3 kgf.cm. Normally, loosing screw torque is 50-70% of tightening screw torque. Electrical cylinder is used for high accuracy of moving screw set toward a working position.

B. Machine controlling

Prototype machine running is controlled by microcontroller [2] due to machine controlling running and designing are not so complicated because it just made for study the result. Nowadays, microcontroller system is improved in many areas such as it's easy to use, ability of data storage is improved, processing is better, controller program development is not complicate and inexpensive price. Microcontrollers used for the machine are ATMEGA 1280, AVR series which is developed by ETT company, ET-EASY MEGA 1280 (DUINO MEGA) to support development program by C language of Arduino which is the main processing unit. Flow chart of prototype machine controlling by microcontroller is shown in figure 6.



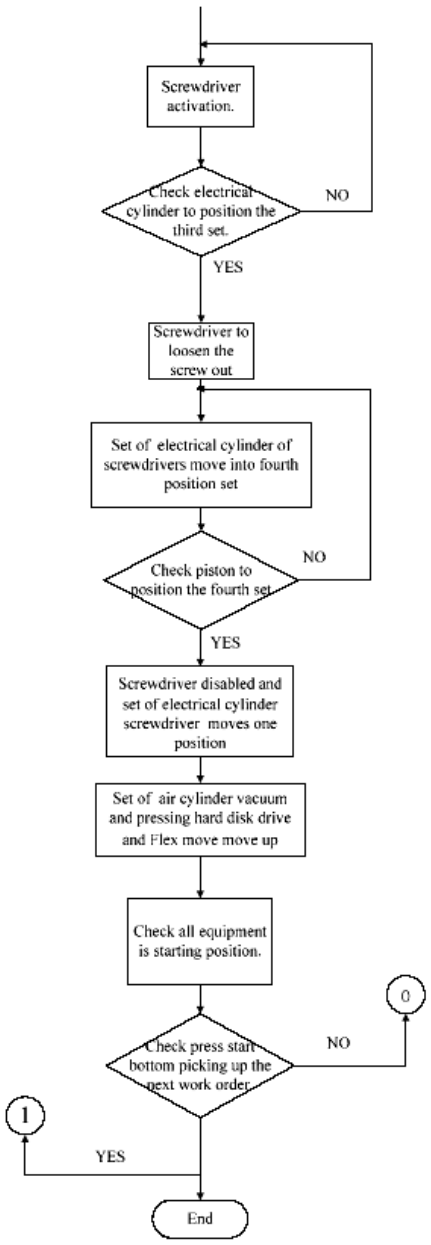


Fig. 6 Flow chart of prototype machine controlling

As designing and controlling master machine by microcontroller, synchronization between computer and microcontroller performs by Microsoft Visual Basic 6.0 software to develop a Graphic User Interface (GUI) on computer. As from the master machine, connection and controller equipment are rotating movement set, pneumatic cylinder, electrical cylinder set and screw set.

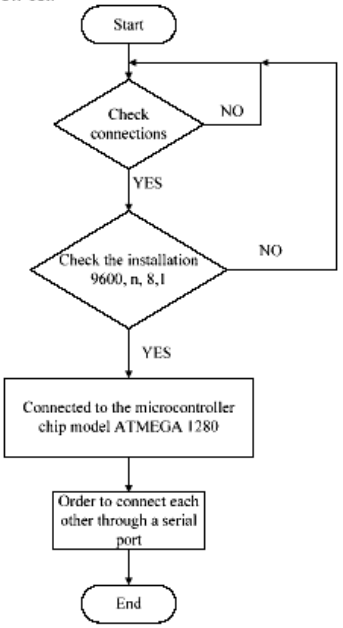


Figure 7 Flow chart of microcontroller and computer synchronization

III. EXPERIMENT AND ITS ANALYSIS

From the study of the procedures in removal of HSA/VCM by man, When designing prototype machine with Solidworks 2008 and operated a prototype machine used in the process of removing HSA / VCM characteristics of the prototype machine was completed as Figure 8.

Fig. 12 shows the complete prototype machine

The built prototype machine, other than working in accordance with the objectives, will also consider about time. Therefore, the machine's rates: distance, speed and motion accelerator of Servo motor and Electrical cylinder were adjusted and tested to find the least working time.

From the 10-times machine rate test, Table 1 shows the total time of each test including all procedures.

TABLE I
SHOWS TEST TIME RECORD

No.	Total time of process (s)
1	14.03
2	13.89
3	13.84
4	13.74
5	13.62
6	13.53
7	13.46
8	13.41
9	13.31
10	13.23

Therefore, the removal test of HSA/VCM which consumes the least time is 13.23 seconds.

TABLE II
SHOWS TIME CONSUMING AT EACH PROCEDURE

Step	Process	Time(s)
1	Operator load unit	2.0
2	Servo motor rotate 90 degree, pneumatic cylinder of clamp and vacuum move down	0.4
3	Electrical cylinder of screw driver down , screw driver remove screw and drive up	5.53
4	Pneumatic cylinder of clamp and vacuum move up, Servo motor forward 90 degree	1.2
5	Operator separate HSA,VCM and Flex then load unit out	4
Total time		13.23

The total manual operation time consumes 19.52 seconds, and when comparing with the taken time of the model machine, the time reduces about 30 percent of the former process. At this total time, the result of the removal of the workpiece was found no damage.

IV. CONCLUSION AND SUGGESTION

This research is for building the prototype automatic machine for the HSA/VCM removal processes by releasing a screw holding HSA, pressing Flex, and vacuum a screw and particles after the screw is released. From the operation test of the prototype machine, it was found that the prototype machine can moderately help reducing time consuming and damages of HSA unit after the removal process. Because the prototype machine was created by using the available devices from the manufacture, the limitation of the devices is included. If this machine will be applied in the real industry to reduce machine importing from overseas, it still requires some improvement and development in the following components.

1) The chosen device is limitation from available devices of manufacturer
2) The development of the fixture should be designed to be able to hold a HSA unit more efficiently.
3) The controller should consider on the industrial environment and operation accuracy.

ACKNOWLEDGEMENT

This research can be finished in accordance with all objectives; the researcher wishes to thank Hitachi GST for fruitful information and every person having given consults, suggestions, instructions, support and encouragement well through all the papers. Thank also to NECTEC for grant to support in this research.

REFERENCES

[1] Joseph E.Shigley, Charles R.Mischke and Richard G.Budynas. Mechanical Engineering Design. Seventh edition. McGrawHill.
[2] Dornsun Pongphab. (2006). Microcontroller and application. Bangkok : Technology promotion association (Thailand-Japan)

ประวัติผู้เขียน

นางสาวพชิรารัชต์ แก้วคำไสย์ เกิดเมื่อวันที่ 27 พฤษภาคม 2527 ที่อำเภอเมือง จังหวัดน่าน เริ่มการศึกษาตั้งแต่อนุบาลถึงระดับประถมศึกษาปีที่ 6 ที่โรงเรียนสาธิตมหาวิทยาลัยราชภัฏวไลยอลงกรณ์ ในพระบรมราชูปถัมภ์ จังหวัดปทุมธานี และมัธยมศึกษาปีที่ 1 ถึง 6 ที่โรงเรียนธรรมศาสตร์คลองหลวงวิทยาคม จังหวัดปทุมธานี สำเร็จการศึกษาวิศวกรรมศาสตรบัณฑิต (วิศวกรรมเครื่องกล) สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา เมื่อ พ.ศ. 2549 ภายหลังสำเร็จการศึกษาได้เข้าศึกษาต่อระดับปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ในขณะที่ศึกษาต่อในระดับปริญญาโท ได้มีประสบการณ์การสอนปฏิบัติการของสาขาวิชาวิศวกรรมเครื่องกลจำนวน 4 รายวิชา ได้แก่ (1) ปฏิบัติการวิศวกรรมเครื่องกล 1 (2) วิชาเขียนแบบวิศวกรรม 1 (3) วิชาเขียนแบบวิศวกรรม 2 (4) วิชาชา MATLAB สำหรับวิศวกรรมเครื่องกล