

การประยุกต์ PLC ควบคุม SCARA Robot 4 DOF วาดเส้นกราฟิกบนพื้น 2 มิติ
THE PLC APPLICATION TO CONTROL SCARA ROBOT 4 DOF TO
DRAW GRAPHICS ON 2D FLOOR



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2566

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE PLC APPLICATION TO CONTROL SCARA ROBOT 4 DOF TO
DRAW GRAPHICS ON 2D FLOOR



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2023

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

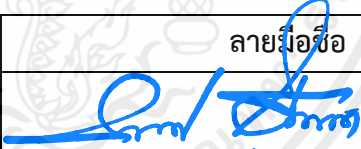

ปริญญาานิพนธ์ปีการศึกษา 2566
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

.....

หัวข้อปริญญาานิพนธ์ การประยุกต์ PLC ควบคุม SCARA Robot 4 DOF วาดเส้นกราฟิกบนพื้น 2 มิติ
THE PLC APPLICATION TO CONTROL SCARA ROBOT 4 DOF TO DRAW
GRAPHICS ON 2D FLOOR

นักศึกษาผู้จัดทำ นายกิมพล แพทย์ไชโย รหัสนักศึกษา 64015109
นายนนทวัฒน์ สีดำ รหัสนักศึกษา 64015065
นายอภิสิทธิ์ อะหาสิเม รหัสนักศึกษา 64015164

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2566

อาจารย์ผู้ควบคุมปริญญาานิพนธ์	ลายมือชื่อ
รศ.ดร. สุพรรณ กุลพานิชย์	
ผศ.ดร. นรินทร์ ธรรมมารักษ์วัฒนะ	

หัวข้อปริญญานิพนธ์	การประยุกต์ PLC ควบคุม SCARA Robot 4 DOF วาดเส้นกราฟิกบนพื้น 2 มิติ	
	THE PLC APPLICATION TO CONTROL SCARA ROBOT 4 DOF TO DRAW GRAPHICS ON 2D FLOOR	
นักศึกษาผู้จัดทำ	นายภิมล แพทย์ไชโย	รหัสนักศึกษา 64015109
	นายนนทวัฒน์ สีดำ	รหัสนักศึกษา 64015065
	นายอภิสิทธิ์ อะหาสิเม	รหัสนักศึกษา 64015164
อาจารย์ที่ปรึกษา	รศ.ดร. สุพรรณ	กุลพาณิชย์
	ผศ.ดร.นรินทร์	ธรรมมารักษ์วัฒน์
ปีการศึกษา	2566	

บทคัดย่อ

โครงการนี้นำเสนอการควบคุมแขนกล SCARA Robot 4DOF ใช้ PLC เป็นเครื่องควบคุมหลักในการสั่งงานเพื่อควบคุมการเคลื่อนที่ของแขนกล ซึ่งประกอบด้วยส่วนหลักๆ 3 ส่วน ได้แก่ ส่วนแรกคือการวิเคราะห์การเคลื่อนที่ของแขนกลตามหลักการของสัญญาณนิมเดนาวิท-ฮาร์เทนเบอร์กประยุกต์ใช้พีชคณิต และตรีโกณมิติโดยการกำหนดองศาการเคลื่อนที่ของแต่ละข้อต่อ และความยาวของก้านต่อ (Link) ของแขนกลให้เป็นตัวแปรที่อยู่ในรูปของสมการคณิตศาสตร์ เพื่อคำนวณหาตำแหน่งส่วนปลายของแขนกล (end-effector) ในขณะเดียวกันนำสมการคณิตศาสตร์ที่ได้มาทำการแปรผลเพิ่มเติมโดยการสร้างแบบจำลองทางคณิตศาสตร์ เพื่อจำลองการเคลื่อนที่ของแขนกลในรูปแบบของกราฟิกคอมพิวเตอร์ด้วยโปรแกรม MATLAB Simulink และสำหรับส่วนที่สองจะเป็นการนำ PLC มาควบคุมแขนกลจริงเพื่อทำการพิสูจน์และเปรียบเทียบความแม่นยำของการเคลื่อนที่กับแบบจำลองทางคณิตศาสตร์ที่สร้างขึ้นว่าการควบคุมแขนกลนั้นมีความถูกต้องมากน้อยเพียงใด ส่วนที่สามเป็นการโปรแกรมให้ แขนกล SCARA Robot 4DOF วาดภาพกราฟิกบนพื้น 2 มิติ มีการออกแบบชุดจับที่ส่วนปลายของแขนกล (end-effector) เพื่อใช้ในการวาดภาพ นอกจากนี้โปรแกรมยังควบคุมให้แขนกลเคลื่อนที่ไปยังพิกัดที่กำหนดแบบอัตโนมัติที่ละตำแหน่งได้ด้วย ซึ่งจากการเก็บบันทึกข้อมูลลักษณะการวาดภาพ และ พิกัดตำแหน่งเป้าหมายของปลายแขนกล มีความถูกต้องอยู่ในเกณฑ์ที่สามารถยอมรับได้

Thesis Title	THE PLC APPLICATION TO CONTROL SCARA ROBOT 4 DOF TO DRAW GRAPHICS ON 2D FLOOR	
Authors	Mr. Peemmapon	Patchaiyo
	Mr. Nontawat	Seedum
	Mr. Apisit	Ahasime
Thesis Advisor	Assoc.Prof.Dr. Suphan	Gulpanich
	Asst.Prof.Dr. Narin	Tammarugwattana
Year	2023	

ABSTRACT

This project presents the control of the SCARA Robot 4DOF mechanical arm. It uses a PLC as the main controller to control the movement of the mechanical arm. It consists of three main parts: The first part is the analysis of the motion of the mechanical arm according to the principles of applied Denavit-Hartenberg signs, algebra, and trigonometry by determining the degree of movement of each joint. and the length of the connecting rod (link) of the mechanical arm are variables in the form of mathematical equations to calculate the position of the tip of the mechanical arm (end-effector) while using the obtained mathematical equations to perform additional transformations by creating mathematical models. To simulate the movement of the mechanical arm in the form of computer graphics with the MATLAB Simulink program, and for the second part, a PLC will be used to control the actual mechanical arm. To verify and compare the accuracy of the movement with the created mathematical model. How accurate is the control of the mechanical arm. The third part is programming the SCARA Robot 4DOF mechanical arm to draw graphics on a 2D floor. The mechanical arm has a gripper design at the end. (end-effector) for use in drawing In addition, the program can control the mechanical arm to automatically move to the specified coordinates, one position at a time. which is from recording data on the characteristics and coordinates of the target position of the mechanical arm tip. The accuracy is within acceptable criteria.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดีเนื่องด้วยการช่วยเหลือจากบุคคลหลายท่านโดยเฉพาะอย่างยิ่งผู้มีพระคุณท่านแรก คณะผู้จัดทำต้องขอขอบคุณ รศ.ดร. สุพรรณ กุลพาณิชย์และ ผศ.ดร. นรินทร์ ธรรมมารักษ์วิวัฒน์อาจารย์ที่ปรึกษา ที่ได้ความรู้ ให้คำแนะนำในการปรับปรุงแก้ไขปัญหาต่างๆ รวมถึงสนับสนุนงบประมาณ สำหรับอุปกรณ์ในการจัดทำโครงการและได้กรุณาตรวจทานความเรียบร้อยของปริญญานิพนธ์ ด้วยความเอาใจใส่ทุกชั้นตอน จนสำเร็จลุล่วงไปได้ด้วยดี

ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมการวัดคุมทุกๆท่าน ที่ได้อบรมสั่งสอนให้ความรู้ในด้านต่างๆคณะกรรมการสอบปริญญานิพนธ์ได้ให้คำแนะนำแก้ไขข้อบกพร่องตลอดการจัดทำที่ผ่านมา

ขอขอบคุณพระบิดา มารดา และครอบครัวของผู้จัดทำรวมถึงขอขอบคุณเพื่อนๆของผู้จัดทำทุกท่าน ที่ได้คอยให้คำปรึกษาแลกเปลี่ยนความรู้ ช่วยเหลือสนับสนุนในทุกๆด้าน จนโครงการสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ขอขอบคุณผู้เขียนหนังสือเอกสารอ้างอิงและเว็บไซต์ต่างๆ ที่คณะผู้จัดทำได้นำมาใช้ อ้างอิงประกอบการศึกษา จนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงโดยสมบูรณ์ที่สุด

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญ(ต่อ).....	V
สารบัญ(ต่อ).....	VI
สารบัญ(ต่อ).....	VII
สารบัญ(ต่อ).....	VIII
สารบัญตาราง.....	IX
สารบัญรูป.....	X
สารบัญรูป(ต่อ).....	XI
สารบัญรูป(ต่อ).....	XII
สารบัญรูป(ต่อ).....	XIII
สารบัญรูป(ต่อ).....	XIV
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	2
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา.....	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง.....	4
2.1 หลักการพื้นฐานของแขนง.....	4
2.1.1 โครงสร้างของแขนง.....	5

สารบัญ(ต่อ)

	หน้า
2.1.1.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์	5
2.1.2 องศาเสรี (Degree of freedom).....	6
2.1.3 พื้นที่การทำงาน (Workspace).....	7
2.2 ทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล.....	7
2.2.1 การเคลื่อนที่ของวัตถุแข็งเกร็ง.....	7
2.2.2 การหมุนเฟรมใน 3 มิติ.....	8
2.2.3 การเคลื่อนที่จากการเลื่อน	12
2.2.4 การแปลงเอกพันธ์	13
2.2.5 สายโซ่จลนศาสตร์	15
2.2.6 การหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับของ SCARA Robot 4DOF	15
2.3 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC).....	18
2.3.1 โครงสร้างและส่วนประกอบของ PLC	18
2.3.1.1 หน่วยประมวลผลกลาง (CPU : Center Processing Unit).....	19
2.3.1.2 หน่วยความจำ (Memory Unit).....	19
2.3.1.3 หน่วยอินพุต/เอาต์พุต (Input/Output Unit).....	20
2.3.1.4 แหล่งจ่ายไฟ (Power Supply Unit)	20
2.3.1.5 เครื่องป้อนโปรแกรม (Programming Device)	20
2.3.2 ภาษาที่ใช้ในการเขียนโปรแกรมให้กับ PLC	21
2.3.2.1 Ladder Diagram (LD)	21
2.3.2.2 Function Block Diagram (FBD)	22
2.3.2.3 Sequential Flow Chart (SFC).....	22
2.3.2.4 Instruction List (IL).....	22
2.3.2.5 Structure Text Language (ST)	22
2.3.3 การทำงานของ PLC.....	22

สารบัญ(ต่อ)

	หน้า
2.3.4 การสแกนอินพุตและเอาต์พุต.....	23
2.3.5 ข้อมูลในหน่วยความจำของ PLC.....	24
2.4 ระบบเลขฐาน.....	24
2.4.1 ระบบเลขฐาน 2 (Binary Number System).....	24
2.4.2 ระบบเลขฐาน 8 (Octal Number System).....	25
2.4.3 ระบบเลขฐาน 10 (Decimal Number System).....	25
2.4.4 ระบบเลขฐาน 16 (Hexadecimal Number System).....	25
2.5 OLE For Process Control (OPC Server).....	26
2.5.1 KEPServerEX6.....	26
2.5.2 Dynamic Data Exchange (DDE).....	27
2.5.2.1 แบบ Hot Link.....	27
2.5.2.2 แบบ Cold Link.....	27
2.5.3 การติดต่อสื่อสารผ่านพอร์ตอนุกรม RS232.....	28
2.6 มอเตอร์แบบสเต็ป.....	29
2.6.1 โครงสร้างและหลักการทำงานของมอเตอร์แบบสเต็ป.....	29
2.6.2 ชนิดของมอเตอร์แบบสเต็ป.....	30
2.6.2.1 มอเตอร์แบบสเต็ปชนิดไปโพลาร์.....	30
2.6.3 การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป.....	30
2.6.3.1 แบบฟลูสเต็ม 1 เฟส (Full Step 1 Phase).....	30
2.6.3.2 แบบฟลูสเต็ม 2 เฟส (Full Step 2 Phase).....	31
2.6.3.3 แบบครึ่งสเต็ป (Half Step).....	31
2.6.4 วงจรขับมอเตอร์.....	32
2.7 โฟโตอิเล็กทรอนิกส์เซอร์.....	33
2.7.1 หลักการโฟโตอิเล็กทรอนิกส์เซอร์ที่ใช้ในปริยญาณิพนธ์.....	33
2.8 ความสัมพันธ์ระหว่างระยะทางและอัตราเร็ว.....	34

สารบัญ(ต่อ)

	หน้า
2.8.1 ระยะทาง (Distance)	34
2.8.2 การกระจัด (Displacement)	34
2.8.3 อัตราเร็ว (Speed)	35
บทที่ 3 วิธีการดำเนินงาน	37
3.1 ภาพรวมการออกแบบการทำงาน.....	37
3.2 การศึกษาพื้นฐานการเคลื่อนที่ของแขนกล.....	39
3.3 การประยุกต์ใช้โปรแกรมแบบจำลองทางคณิตศาสตร์	41
3.4 การออกแบบโปรแกรม PLC ควบคุมการทำงานของแขนกล.....	45
3.4.1 การเขียนโปรแกรม PLC Ladder ควบคุมการทำงานของมอเตอร์แบบสเต็ป.....	45
3.4.2 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่.....	48
ของแขนกลแบบย้อนกลับ	
3.4.3 การเขียนโปรแกรม PLC คำนวณความเร็วการเคลื่อนที่ของมอเตอร์	50
3.4.4 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่เป็นแบบอัตโนมัติ.....	51
เพื่อใช้ในการวาดภาพ 2 มิติ	
3.4.5 การออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน	54
3.5 การเชื่อมต่อ KEPServerEX6 เข้ากับ PLC	56
3.5.1 การเชื่อมต่อ Microsoft Excel กับ KEPServerEX6	56
3.5.2 การเชื่อมต่อ KEPServerEX6 กับ CX-Programmer	58
3.6 การออกแบบตัวยึดปากกาของแขนกล SCARA Robot 4DOF	59
บทที่ 4 ผลการทดลอง	62
4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่.....	62
ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทาง คณิตศาสตร์	
4.2 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพ 2 มิติที่หลากหลายรูปแบบ ..	65

สารบัญตาราง

ตารางที่	หน้า
2.1 ลักษณะของข้อต่อที่ใช้ในแกนกล	6
2.2 ตารางตัวอย่างความสัมพันธ์ระหว่างเลขฐานแต่ละระบบ	25
2.3 ลำดับการจ่ายกระแสกระตุ้นแบบฟลูสเต็มป์ 1 เฟส.....	31
2.4 ลำดับการจ่ายกระแสกระตุ้นแบบฟลูสเต็มป์ 2 เฟส.....	31
2.5 ลำดับการจ่ายกระแสกระตุ้นแบบครึ่งสเต็มป์	32
4.1 การทดลองควบคุมแกนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแกนกล.....	63
แบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแกนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด $(+x +y)$	
4.2 การทดลองควบคุมแกนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแกนกล.....	63
แบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแกนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด $(+x -y)$	
4.3 การทดลองควบคุมแกนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแกนกล.....	64
แบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแกนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด $(-x +y)$	
4.4 การทดลองควบคุมแกนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแกนกล.....	64
แบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแกนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด $(-x -y)$	
4.5 การทดลองในการควบคุมแกนกลให้มีการเคลื่อนที่วาดภาพสามเหลี่ยม 2 มิติ.....	65
4.6 การทดลองในการควบคุมแกนกลให้มีการเคลื่อนที่วาดภาพสี่เหลี่ยม 2 มิติ.....	66
4.7 การทดลองในการควบคุมแกนกลให้มีการเคลื่อนที่วาดภาพสี่เหลี่ยมผืนผ้า 2 มิติ.....	67
4.8 การทดลองในการควบคุมแกนกลให้มีการเคลื่อนที่วาดภาพสี่เหลี่ยมคางหมู 2 มิติ.....	68
4.9 การทดลองในการควบคุมแกนกลให้มีการเคลื่อนที่วาดภาพนาฬิกาทราย 2 มิติ.....	69
4.10 แสดงเปอร์เซ็นต์ค่าความคลาดเคลื่อนจริงที่แกนกลเคลื่อนที่ไปในพิกัดแต่ละตำแหน่ง.....	70
ในการวาดภาพ 2 มิติ	

สารบัญรูป

รูปที่	หน้า
2.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์ของแขนกล	5
2.2 ตัวอย่างภาพประกอบของแขนกล.....	6
2.3 การหมุนของเฟรมใน 3 มิติ	8
2.4 การหมุนของเฟรมใน 3 มิติ	8
2.5 เฟรมที่เกิดจากการหมุนรอบแกน Z.....	9
2.6 การติดตั้งเฟรมเข้ากับวัตถุแข็งเกรง.....	11
2.7 การเคลื่อนที่ของเฟรมโดยการเลื่อน.....	13
2.8 มุมมองด้านบนแสดงความสัมพันธ์ของเฟรมที่ 4 เทียบกับเฟรมอ้างอิง.....	16
2.9 โครงสร้างสถาปัตยกรรมของ PLC.....	18
2.10 ตัวอย่างการต่อคอมพิวเตอร์เข้ากับ PLC.....	21
2.11 วงรอบลำดับการทำงานของ PLC	23
2.12 การประยุกต์ใช้ KEP SERVEREX6 กับงานด้านต่าง ๆ.....	27
2.13 ส่วนประกอบของระบบ DYNAMIC DATA EXCHANGE (DDE).....	28
2.14 ตัวอย่างการเชื่อมต่อคอมพิวเตอร์เข้ากับ PLC ผ่านพอร์ตอนุกรม RS232	28
2.15 มอเตอร์แบบสเต็ปขณะหยุดนิ่ง และขณะหมุนไปได้ครึ่งสเต็ป	29
2.16 โครงสร้างของมอเตอร์แบบสเต็ปชนิดไบโพลาร์.....	30
2.17 การต่อใช้งาน MICRO STEP DRIVER กับเครื่องควบคุม	33
2.18 การทำงานของโฟโตอิเล็กทริกเซนเซอร์แบบ U-TYPE.....	34
2.19 ระยะเวลาและการกระจัด	34
3.1 ภาพรวมการทำงานของโครงการ.....	37
3.2 ศึกษาพื้นฐานที่เกี่ยวข้องเรื่องพิกัดเฟรมและการแปลง.....	39
3.3 ศึกษาพื้นฐานที่เกี่ยวข้องคำศัพท์ในวิทยาการหุ่นยนต์.....	39
3.4 ศึกษาพื้นฐานที่เกี่ยวข้องการจำลองเชิงจลนศาสตร์.....	40
3.5 ศึกษาพื้นฐานที่เกี่ยวข้องการคำนวณจลนศาสตร์ของแขนกล.....	40

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.6 ตัวอย่างการศึกษาพื้นฐานที่เกี่ยวข้องกับการเคลื่อนที่ของแขนกล.....	40
3.7 ระยะเวลาของแขนกลSCARA Robot 4DOF.....	41
3.8 แสดง Flow chart เงื่อนไขการทำงานของแบบจำลองทางคณิตศาสตร์.....	43
3.9 การเขียนโปรแกรมสำหรับคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ	44
3.10 หน้าต่างโปรแกรมการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ	44
3.11 แสดงหน้าต่าง GUI และโมเดลแขนกลจำลอง.....	45
3.12 CX-Programmer Settings	46
3.13 การตั้งค่า Pulse Output.....	46
3.14 โปรแกรมการควบคุมมอเตอร์แบบสแต็ปของแขนกล.....	47
3.15 แสดง Flowchart เงื่อนไขการทำงานของโปรแกรม PLC ควบคุมการทำงานของแขนกล.....	48
3.16 Function Block ของการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ.....	49
3.17 Structure Text ในการคำนวณหาค่ามุมหรือระยะเลื่อนของข้อต่อแขนกล	49
3.18 Function Block ของการคำนวณหาค่าความเร็วของข้อต่อแต่ละแกนในการเคลื่อนที่.....	50
3.19 Structure Text ในการคำนวณหาค่าความเร็วของข้อต่อแต่ละแกนในการเคลื่อนที่.....	50
3.20 เงื่อนไขการทำงานของโปรแกรมควบคุมแบบอัตโนมัติ.....	51
3.21 โปรแกรมสร้างช่องทางเรียกใช้ข้อมูลแบบ Indirect Address.....	52
3.22 โปรแกรมเก็บค่าที่ได้จากการคำนวณไว้ในช่องทางเดียวกับการสร้าง Indirect Address.....	52
3.23 โปรแกรมควบคุมการเคลื่อนที่แบบอัตโนมัติโดยใช้ Shift register.....	53
3.24 โปรแกรมควบคุมการเคลื่อนที่โดยใช้ Complete Flag ของ Pulse Outแต่ละแกน.....	53
3.25 โปรแกรมการส่งค่าไปยังฟังก์ชัน PULS(886) และ SPED(885).....	54
3.26 Main Menu (HMI).....	54
3.27 หน้าต่าง Sequence Control Mode(HMI).....	55
3.28 หน้าต่าง Data Pulse(HMI).....	55
3.29 หน้าต่าง DATA Speed(HMI).....	56
3.30 การตั้งค่า Microsoft Excel.....	56

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.31 ช่องทางการส่งข้อมูลจาก Microsoft Excel ไปสู่ PLC.....	57
3.32 สร้าง Tag data.....	57
3.33 สร้าง Advance tag.....	58
3.34 การเชื่อมต่อระหว่างคอมพิวเตอร์กับ PLC.....	58
3.35 การส่งข้อมูลระหว่างต้นทาง Microsoft Excel และปลายทาง Data memory PLC.....	59
3.36 ระยะเวลาช่องว่างของส่วนปลายเพื่อออกแบบเครื่องมือปกปากกา.....	59
3.37 ออกแบบเครื่องมือโดยใช้โปรแกรม Auto CAD.....	60
3.38 พิมพ์เครื่องมือของแขนกล SCARA Robot 4DOF.....	60
3.39 ติดตั้งปกปากกากับแขนกลSACARA Robot 4DOF.....	61
3.40 ติดตั้งปากกาและปกปากกากับแขนกลSACARA Robot 4DOF.....	61
ก-1 คุณสมบัติCP1Hหน้าที่ 1	77
ก-2 คุณสมบัติCP1Hหน้าที่ 2	78
ก-3 คุณสมบัติCP1Hหน้าที่ 3	79
ก-4 คุณสมบัติCP1Hหน้าที่ 4	80
ก-5 คุณสมบัติCP1Hหน้าที่ 5	81
ก-6 คุณสมบัติCP1Hหน้าที่ 6	82
ก-7 คุณสมบัติCP1Hหน้าที่ 7	83
ก-8 คุณสมบัติCP1Hหน้าที่ 8	84
ก-9 คุณสมบัติCP1Hหน้าที่ 9	85
ก-10 คุณสมบัติของHMI Omron NB5QW00Bหน้าที่ 1.....	86
ก-11 คุณสมบัติของHMI Omron NB5QW00Bหน้าที่ 2.....	87
ก-12 คุณสมบัติของHMI Omron NB5QW00Bหน้าที่ 3.....	88
ก-13 คุณสมบัติของHMI Omron NB5QW00Bหน้าที่ 4.....	86
ข-1 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 1.....	92
ข-2 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 2.....	93

สารบัญญรูป(ต่อ)

รูปที่	หน้า
ข-3 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 3.....	94
ข-4 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 4.....	95
ข-5 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 5.....	96
ข-6 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 6.....	97
ข-7 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 7.....	98
ข-8 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 8.....	99
ข-9 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 9.....	100
ข-10 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 10.....	101
ค-2 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 1.....	103
ค-2 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 2.....	104
ค-3 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 3.....	105
ค-4 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 4.....	106
ค-5 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 5.....	107
ค-6 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 6.....	108
ค-7 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 7.....	109
ค-8 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 8.....	110
ค-9 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 9.....	111
ค-10 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 10.....	112
ค-11 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 11.....	113
ค-12 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 12.....	114
ค-13 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 13.....	115
ค-14 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 14.....	116
ค-15 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 15.....	117
ค-16 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 16.....	118
ค-17 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 17.....	119

สารบัญรูป(ต่อ)

รูปที่	หน้า
ค-18 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 18	120
ค-19 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 19	121
ค-20 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 20	122
ค-21 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 20	123
ค-22 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 22	124
ค-23 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 23	125
ค-24 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 24	126
ค-25 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 25	127
ค-26 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 26	128
ค-27 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 27	129
ค-28 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 28	130
ค-29 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 29	131
ค-30 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 30	132
ค-31 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 31	133
ค-32 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 32	134
ค-33 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 33	135
ค-34 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 34	136
ค-35 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 35	137
ค-36 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 36	138
ค-37 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 37	139

บทที่ 1

บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

อุตสาหกรรมในประเทศไทยปัจจุบันได้มีการพัฒนาเทคโนโลยี เกี่ยวกับระบบอัตโนมัติและมีแนวโน้มในการปรับตัวเข้ากับอุตสาหกรรมอัตโนมัติ เช่น การนำแขนกลต่างๆ เข้ามาทำงานแทนมนุษย์ มีบทบาทสำคัญในกระบวนการผลิตมากขึ้น โดยปัจจุบันมีหลากหลายอุตสาหกรรมที่นำแขนกลเข้ามาใช้งาน เช่น อุตสาหกรรมยานยนต์ อุตสาหกรรมแปรรูป อุตสาหกรรมอาหาร เป็นต้น ทั้งนี้เพื่อต้องการยกระดับและเพิ่มความสามารถของกระบวนการผลิตให้สูงขึ้น และต้องการที่จะลดต้นทุนการผลิต เนื่องจากแขนกลสามารถทำงานได้อย่างต่อเนื่อง เกินกว่าขีดจำกัดของแรงงานมนุษย์ อีกทั้งแขนกลยังสามารถทำงานได้ถูกต้องและมีความเที่ยงตรงที่มากกว่า ทำให้สามารถลดความผิดพลาดที่อาจจะเกิดขึ้นในกระบวนการผลิตได้ดีกว่าการทำงานของมนุษย์

โดยในควบคุมการทำงานของแขนกล ให้สามารถเคลื่อนที่ไปยังตำแหน่งเป้าหมายใดๆ ได้นั้น สามารถแบ่งประเภทของการควบคุมออกเป็น 2 วิธี อ้างอิงตามทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล ซึ่งจะประกอบด้วย การควบคุมโดยการป้อนค่ามุมของแต่ละข้อต่อของแขนกล (Joint Angle) ที่ต้องการให้เคลื่อนที่ไป แล้วนำมาคำนวณหาค่าของตำแหน่งสุดท้ายของส่วนปลายแขนกล (x, y, z) ด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้า และการควบคุมอีกรูปแบบหนึ่งคือการป้อนค่าตำแหน่งสุดท้ายของส่วนปลายแขนกล (x, y, z) แล้วทำการคำนวณกลับด้วยจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับเพื่อหาค่าของมุมแต่ละข้อต่อของแขนกล (Joint Angle) ที่จะทำให้แขนกลเคลื่อนที่ไปยังจุดนั้นๆ ได้ ดังนั้นจะเห็นได้ว่าทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลทั้งแบบไปข้างหน้าและแบบย้อนกลับนั้นสามารถนำมาวิเคราะห์เพื่อใช้ในการควบคุมแขนกลให้เคลื่อนที่ไปตามที่กำหนดได้

ทางด้านผู้จัดทำโครงการได้ประยุกต์ใช้ PLC ควบคุมแขนกล SCARA Robot 4DOF ในการเคลื่อนที่ให้เป็นไปตามทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับเพียงส่วนเดียว มีการใช้แบบจำลองการเคลื่อนที่ทางคณิตศาสตร์ จำลองการเคลื่อนที่ของแขนกล SCARA Robot 4DOF เป็นการตรวจสอบว่าค่าอินพุตที่ป้อนนั้นมีค่าที่ถูกต้องและอยู่ในระยะขอบเขตการทำงานของแขนกลหรือไม่ เพื่อให้สามารถนำไปป้อนให้กับแขนกลในทางปฏิบัติจริง และเปรียบเทียบความถูกต้องของค่ามุมมองแต่ละแกนระหว่างแบบจำลองทางคณิตศาสตร์กับแขนกล SCARA Robot 4DOF จริง ว่ามีความแตกต่างกันน้อยเพียงใด รวมถึงเพิ่มการทำงานให้เคลื่อนที่เป็นลำดับขั้นตอนเพื่อให้แขนกล SCARA Robot

4DOF วาดภาพบนพื้น 2 มิติได้ โดยมีการพัฒนาโปรแกรมจากทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับให้สามารถรับค่าอินพุตได้มากขึ้น ตามรูปแบบต่างๆ และมีการออกแบบเครื่องมือส่วนปลายของแขนกล SCARA Robot 4DOF ในลักษณะเป็นปากกา เพื่อให้สามารถวาดออกมาเป็นรูปภาพ 2 มิติ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อประยุกต์ใช้เครื่องควบคุม PLC ให้ทำการควบคุมแขนกล SCARA Robot 4DOF ให้สามารถเคลื่อนที่ได้ตามทฤษฎีจลนศาสตร์การเคลื่อนที่แบบย้อนกลับ
2. เพื่อศึกษาการใช้งานและการควบคุมตำแหน่งของ Stepping Motor สำหรับการเคลื่อนที่ของข้อต่อ (Joint) ของแขนกล SCARA Robot 4DOF
3. เพื่อควบคุมการเคลื่อนที่ของแขนกลให้ได้ตามพิกัดค่าแกน จากการออกแบบกราฟิก 2 มิติด้วยข้อมูลจาก G-code
4. เพื่อศึกษาวิธีการออกแบบแบบจำลอง แสดงผลการเคลื่อนที่ของแขนกล SCARA Robot ในรูปแบบของกราฟิกด้วยโปรแกรม MATLAB Simulink

1.3 ขอบเขตของปริญญานิพนธ์

1. ศึกษาแขนกล SCARA Robot 4DOF โดยมีชุดขับเคลื่อนเป็น Stepping Motor
2. ออกแบบ Graphic Editor เพื่อเป็นข้อมูลกำหนดตำแหน่งให้ SCARA Robot เคลื่อนที่เพื่อวาดเป็นเส้นกราฟิก
3. มีการจัดเก็บข้อมูลให้อยู่ในรูปแบบของ Text File หรือใน Excel File
4. ใช้เครื่องควบคุม PLC ที่มี High Speed O/P ไม่น้อยกว่า 4 แกน
5. การควบคุมตำแหน่งของแขนกล SCARA Robot 4DOF สามารถพิสูจน์ได้จากสมการจลนศาสตร์การเคลื่อนที่แบบย้อนกลับ ด้วยวิธีการ Trigonometry
6. สร้างแบบจำลองการเคลื่อนที่ของแขนกล SCARA Robot 4DOF ด้วยโปรแกรม MATLAB Simulink ที่สอดคล้องกับแขนกล
7. ใช้แขนกล SCARA Robot 4DOF วาดภาพกราฟิกบนพื้น 2 มิติ

1.4 ขั้นตอนการศึกษา

1. ศึกษาพื้นฐานการเคลื่อนที่ของแขนกล SACARA Robot 4 DOF และวิธีการวิเคราะห์หาสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ ที่สัมพันธ์กับการเคลื่อนที่ของแขนกล
2. ศึกษาวิธีในการเขียนโปรแกรม PLC เพื่อควบคุมการทำงานของ Stepping Motor ด้วย ภาษาแบบ Ladder, Structure Text และ Function Block

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. สามารถควบคุมการเคลื่อนที่ของแขนกล SCARA Robot 4DOF ตามทฤษฎีจลนศาสตร์การเคลื่อนที่แบบย้อนกลับได้
2. สร้างองค์ความรู้และเทคนิคการเขียนโปรแกรมเครื่องควบคุม PLC ที่ใช้เป็นเครื่องควบคุมการทำงานของแขนกล ด้วยภาษา Ladder และ Structure Text ไปประยุกต์ใช้กับเครื่องจักรกลประเภทอื่นๆ หรือนำไปประกอบวิชาชีพได้อย่างสมบูรณ์
3. สามารถเตรียมและสร้างเส้นทางการเดินของแขนกลจากโปรแกรม Editor และ ข้อมูล G code ไปควบคุมการทำงานของแขนกลให้ได้ตามการออกแบบ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ในการทำปริญญานิพนธ์ PLC ควบคุมแขนกล SCARA Robot 4DOF ตามหลักการจลนศาสตร์ การเคลื่อนที่แบบย้อนกลับ ทางคณะผู้จัดทำได้ทำการศึกษาเอกสารที่เกี่ยวข้องต่างๆไม่ว่าจะเป็นหนังสือ ข้อมูลจากสื่อต่างๆและบทความ ที่เกี่ยวข้องกับการทำชิ้นงาน SCARA Robot 4DOF และปริญญานิพนธ์ ซึ่งเป็นส่วนอ้างอิงในการกล่าวถึงทฤษฎีและหลักการทำงานของอุปกรณ์ต่างๆและข้อมูลความรู้ที่เกี่ยวข้อง รวมไปถึงการทดสอบชิ้นงานซึ่งจะมีการอธิบายตามเนื้อหาดังต่อไปนี้

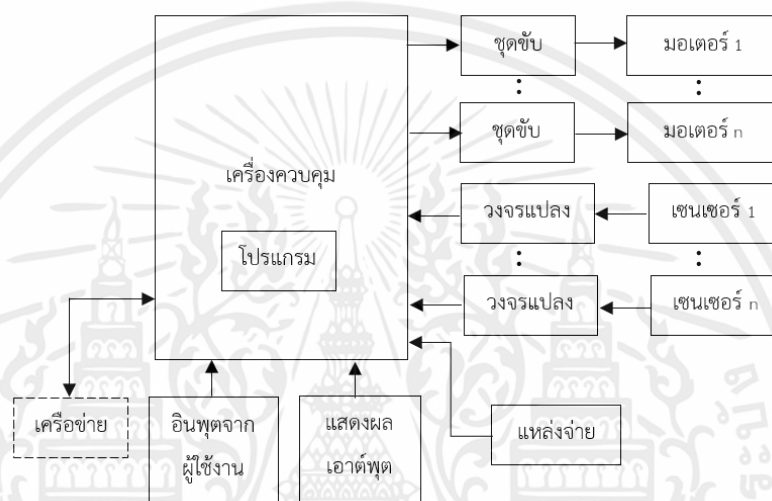
- 2.1 หลักการพื้นฐานของแขนกล
- 2.2 ทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล
- 2.3 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)
- 2.4 ระบบเลขฐาน
- 2.5 OLE for Process Control (OPC Sever)
- 2.6 มอเตอร์แบบสแต็ป
- 2.7 โฟโตอิเล็กทริกเซนเซอร์
- 2.8 ความสัมพันธ์ระหว่างระยะทางและอัตราเร็ว

2.1 หลักการพื้นฐานของแขนกล

ความหมายของแขนกลนั้น คือตัวจัดการ (Manipulator) ที่สามารถทำหน้าที่ได้หลายอย่าง และสามารถโปรแกรมการทำงานใหม่ได้ ที่ออกแบบมาเพื่อใช้งานสำหรับการเคลื่อนย้ายวัสดุชิ้นงาน หรือทำงานในด้านอื่นๆ โดยที่การเคลื่อนที่จะถูกเขียนโปรแกรมเอาไว้ผ่านเครื่องควบคุมทำให้แขนกล สามารถทำงานได้อย่างมีประสิทธิภาพและแม่นยำ ซึ่งจากนิยามของคำว่า “การทำหน้าที่ได้หลายอย่าง” และ “การโปรแกรมใหม่ได้” ทำให้แขนกลนั้นมีความแตกต่างไปจากเครื่องจักรอัตโนมัติ ประเภทอื่นที่มีการควบคุมผ่านคอมพิวเตอร์และเครื่องควบคุมเหมือนกัน เช่น เครื่องซีเอ็นซี หรือ เครื่องพิมพ์ 3 มิติ เพราะว่า แขนกลมีความยืดหยุ่นในการทำงานที่มากกว่า ตัวอย่างเช่น เครื่องซีเอ็นซี ที่มีการทำงานที่เพียงแค่งัด ชิ้นงานในพิกัด 3 มิติ และไม่สามารถนำมาใช้งานในด้านอื่น อย่างเช่น การเคลื่อนย้ายสิ่งของได้ แต่แขนกลนั้นสามารถนำมามาติดหัวกัดและเขียนโปรแกรมให้กัดชิ้นงาน หรือ สามารถที่จะเปลี่ยนจากหัวกัดเป็นหัว เครื่องมืออื่น ๆ และเขียนโปรแกรมใหม่ เพื่อนำไปใช้งานในการเคลื่อนย้ายวัตถุ หรือเชื่อมชิ้นงานได้

2.1.1 โครงสร้างของแขนกล

โครงสร้างของแขนกลประกอบด้วยอุปกรณ์ และองค์ประกอบต่าง ๆ ซึ่งสามารถแบ่ง ออกได้เป็น 2 ส่วน คือ ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์ และส่วนประกอบทางกลของแขนกล โดยสามารถอธิบายได้ดังนี้



รูปที่ 2.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์ของแขนกล



2.1.1.1 ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์

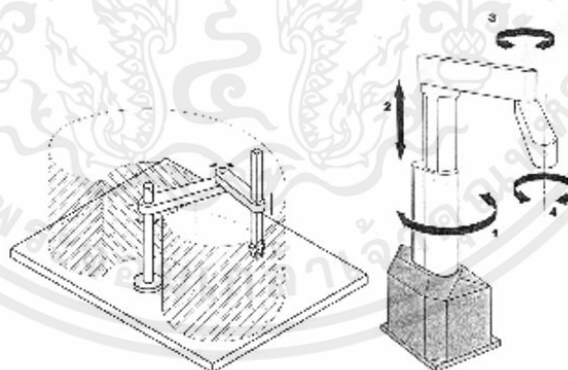
ส่วนประกอบทางไฟฟ้าและคอมพิวเตอร์พื้นฐานของแขนกลนั้น ดังรูปที่ 2.1 ซึ่ง จะประกอบไปด้วยตัวประมวลผลซึ่งเป็นเครื่องควบคุมที่ทำหน้าที่ประมวลผล และควบคุมการ ทำงานของ แขนกลตามโปรแกรมที่มีการเขียนเอาไว้ โดยที่เครื่องควบคุมจะรับค่าข้อมูล ซึ่งเป็นค่า ตำแหน่งต่างๆ จาก ผู้ใช้ผ่านอุปกรณ์อินพุต เช่น HMI ส่วนมอเตอร์เปรียบเสมือนเป็นส่วนขับเคลื่อน ใน การเคลื่อนที่เพื่อให้ แขนกลเคลื่อนที่ไปสู่ตำแหน่งเป้าหมาย และแขนกลจะทำงานได้อย่างถูกต้อง และแม่นยำได้ต้องมี เซนเซอร์หรืออุปกรณ์ตรวจจับ เพื่อใช้วัดค่าตำแหน่งการเคลื่อนที่ของข้อต่อ หรือ อาจจะเป็นเซนเซอร์เพื่อ ป้องกันการเคลื่อนที่ที่ผิดพลาดของแขนกล และส่วนประกอบทั้งหมดในส่วน นี้จะมีแหล่งจ่าย เพื่อให้ อุปกรณ์ทุกส่วนนั้นสามารถทำงานได้

2.1.1.2 ส่วนประกอบทางกล

ส่วนประกอบทางกลเป็นส่วนประกอบที่มีความสำคัญสำหรับการนำไป วิเคราะห์ทางจลนศาสตร์ ซึ่งส่วนประกอบทางกลจะประกอบด้วย 1) ฐาน (Base) ซึ่งเป็นส่วนที่ถูกยึดติดอยู่กับที่ 2) ข้อต่อ (Joints) เป็นจุดเชื่อมต่อระหว่างส่วนต่างๆในแขนกลและเป็นส่วนที่เคลื่อนที่ได้ของแขนกล ซึ่งข้อต่อจะแบ่งออกเป็น 2 ประเภท แสดงดังตารางที่ 2.1 3) ก้านต่อ (Links) เป็นส่วน ที่เชื่อมต่อระหว่างข้อต่อแต่ละจุด 4) ส่วนปลายของแขนกล หรือ เครื่องมือปลายแขน (End Effector) เป็นส่วนที่ติดตั้งเครื่องมือเพื่อใช้งานต่าง ๆ และเป็นส่วนที่สัมผัสกับชิ้นงาน

ตารางที่ 2.1 ลักษณะของข้อต่อที่ใช้ในแขนกล

ประเภท	อักษรย่อ	สัญลักษณ์	คำอธิบาย
แบบหมุน (Revolute)	R		การเคลื่อนที่แบบหมุนรอบแกน
แบบเชิงเส้น หรือ เลื่อน (Prismatic)	P		การเคลื่อนที่แบบเชิงเส้นเลื่อนไปตามแกน



รูปที่ 2.2 ตัวอย่างภาพประกอบของแขนกล

2.1.2 องศาเสรี (Degree of freedom)

ในที่ในระบบ 3 มิติในการเคลื่อนที่ในระบบ 3 มิติสามารถอธิบายได้โดยพารามิเตอร์ที่เรียกว่าค่าตำแหน่ง และมุมการหมุนองศาเสรี (Degree of freedom) ซึ่งเป็นจำนวนของพารามิเตอร์ที่บ่งบอกถึงการ เคลื่อนที่ของวัตถุที่เป็นอิสระ โดยค่าองศาเสรีจะมีมากหรือน้อยจะขึ้นอยู่กับโครงสร้างทาง

กายภาพของวัตถุว่าถูกจำกัดอยู่หรือไม่ ในทางแขนกลนั้นค่าพารามิเตอร์องศาเสรี จะบอกถึงจำนวนของ จุดข้อต่อที่สามารถเคลื่อนที่ได้ของแขนกล ไม่ว่าจะเป็นการเคลื่อนที่แบบหมุน หรือแบบเลื่อนก็ตาม

2.1.3 พื้นที่การทำงาน (Workspace)

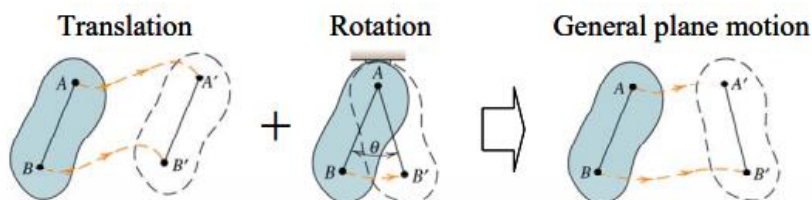
พื้นที่การทำงานของแขนกล (Workspace) คือ พื้นที่ หรือ ปริมาตร ทั้งหมดที่ส่วน ปลายของแขนกลนั้นสามารถเข้าถึงได้ ซึ่งก็คือพื้นที่ที่แขนกลสามารถเคลื่อนที่ หรือ ทำงาน ไปยังจุด นั้นได้ โดยที่ไม่ถูกขวางกั้น หรือเกิดความเสียหาย

2.2 ทฤษฎีจลนศาสตร์การเคลื่อนที่ของแขนกล

ทฤษฎีจลนศาสตร์ เป็นการอธิบายถึงการเคลื่อนที่ของวัตถุโดยที่ไม่พิจารณาแรงที่มากระทำ กับวัตถุและพลวัต ทำให้ตำแหน่งทางจลนศาสตร์จะขึ้นอยู่กับตำแหน่งทางเรขาคณิตของวัตถุ ในส่วนของการคิดจลนศาสตร์ของแขนกลนั้นจะขึ้นอยู่กับโครงสร้างของข้อต่อและก้านต่อ ซึ่งจลนศาสตร์ของแขนกลนั้นสามารถแบ่งออกได้เป็น 2 ส่วน คือ จลนศาสตร์แบบไปข้างหน้า (Forward Kinematics) ซึ่งเป็นการหาพิกัดและทิศทางในรูปของการแปลงเอกพันธ์ของส่วนปลายของแขนกล โดยที่ทราบ ตำแหน่งและมุมของข้อต่อทั้งหมด ในทางกลับกันจลนศาสตร์แบบย้อนกลับ (Inverse Kinematics) จะเป็นการหาค่าตำแหน่งและมุมของข้อต่อ โดยที่ทราบพิกัดตำแหน่งหรือเมทริกซ์เอกพันธ์ของส่วน ปลายของแขนกลเมื่อเทียบกับเฟรมฐาน ซึ่งหลักการและวิธีการได้มาซึ่งสมการจลนศาสตร์ทั้ง 2 แบบ จะอธิบายในส่วนตัวไป

2.2.1 การเคลื่อนที่ของวัตถุแข็งเกร็ง

การเคลื่อนที่ของแขนกลจะมีลักษณะการเคลื่อนที่ที่เป็นแบบการเคลื่อนที่ของวัตถุแข็งเกร็ง กล่าวคือ วัตถุแข็งเกร็งจะไม่เปลี่ยนแปลงในขณะที่แขนกลนั้นเคลื่อนที่ โดยที่ตำแหน่งสัมพัทธ์ระหว่างอนุภาคภายในไม่มีการเปลี่ยนแปลง การเคลื่อนที่ของวัตถุแข็งเกร็งนั้นจะประกอบด้วย การ หมุน (Rotation) และการเลื่อน (Translation) ซึ่งก็คือการเคลื่อนที่ของข้อต่อแขนกลนั่นเอง ดังนั้นการศึกษาวิเคราะห์การเคลื่อนที่ของแขนกลจึงนำวิธีการวิเคราะห์การเคลื่อนที่ของวัตถุแข็งเกร็งมาใช้ ซึ่งวิธีการในการวิเคราะห์การเคลื่อนที่ของวัตถุแข็งเกร็งคือการติดเฟรมพิกัด (Coordinate frame) หรือเฟรม $\{i\}$ ที่วัตถุ ณ ตำแหน่งนั้น ๆ โดย i คือ หมายเลขหรือชื่อเฟรม ซึ่งเฟรมก็คือพิกัดคาร์ทีเซียน ที่ประกอบไปด้วย แกน (x, y, z) ที่ตั้งฉากกันตามกฎมือขวา โดยเฟรมนี้จะยึดติดและเคลื่อนที่ไปพร้อมกับวัตถุสามารถอธิบายได้ว่าการเคลื่อนที่ของวัตถุแข็งเกร็งนั้นเป็นการหมุนและเคลื่อนที่ของเฟรม ณ ตำแหน่งนั้นๆเทียบกับเฟรมอ้างอิง

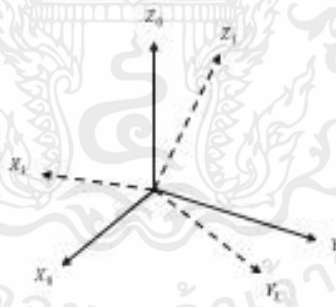


รูปที่ 2.3 การหมุนของเฟรมใน 3 มิติ

2.2.2 การหมุนเฟรมใน 3 มิติ

การหมุนของเฟรมใน 3 มิติทำได้โดยการฉายแกน x, y, z ของเฟรม $\{1\}$ ที่มีการหมุนลงบนแกนหลักของเฟรมอ้างอิง $\{0\}$ โดยใช้วิธีของผลคูณจุด (Dot product) และกำหนดให้ แกน x, y, z ของเฟรมทั้ง 2 เป็นเวกเตอร์หนึ่งหน่วย (Unit vector) จะได้เมทริกซ์การหมุน (Rotation Matrix) ที่สัมพันธ์กับการหมุนระหว่าง 2 เฟรมได้ว่า

$${}^0R_1 = \begin{bmatrix} x_1 & x_0 & y_1 & x_0 & z_1 & x_0 \\ x_1 & y_0 & y_1 & y_0 & z_1 & y_0 \\ x_1 & z_0 & y_1 & z_0 & z_1 & z_0 \end{bmatrix} \quad (2.1)$$



รูปที่ 2.4 การหมุนของเฟรมใน 3 มิติ

ในทางกลับกันหากต้องเริ่มจากเฟรม $\{1\}$ และหมุนให้เฟรมเป็น $\{0\}$ จะได้เมทริกซ์การหมุนเป็น

$${}^0R_1 = \begin{bmatrix} x_0 & x_1 & y_0 & x_1 & z_0 & x_1 \\ x_0 & y_1 & y_0 & y_1 & z_0 & y_1 \\ x_0 & z_1 & y_0 & z_1 & z_0 & z_1 \end{bmatrix} \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยคุณสมบัติสลับที่ของผลคูณจุด ($x_i y_j = x_j y_i$) จะได้เป็น

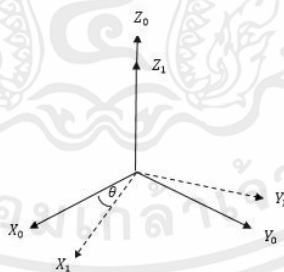
$${}^0R_I = {}^0R_I^T \quad (2.3)$$

ซึ่งในมุมมองของทางเรขาคณิตและตรีโกณมิติ การหมุนเฟรมจากเฟรม {1} เป็นเฟรม {0} เป็นการดำเนินการผูกผันกับเมทริกซ์การหมุนโดยการนำทรานสโพสของสมการเมทริกซ์การหมุนคูณกับตัวเอง ดังสมการที่ (2.4) ซึ่งเมทริกซ์ที่มีคุณสมบัติเช่นนี้จะถูกเรียกว่า เมทริกซ์เชิงตั้งฉาก (Orthogonal Matrix) หรือ $R \in SO(n)$ ซึ่ง $SO(n)$ คือกลุ่มเชิงตั้งฉากพิเศษ (Special Orthogonal group) จำนวน n

$${}^0R_I^T = {}^0R_I^{-1} \quad (2.4)$$

โดยในลำดับถัดไปจะเป็นการหาเมทริกซ์การหมุนรอบแกน z จากรูปที่ 2.5 เฟรม {1} เกิดจากการหมุนรอบแกน z_0 ของเฟรม {0} เป็นมุม θ โดยใช้สมการความสัมพันธ์ของผลคูณจุดกับ เวกเตอร์ ใน 3 มิติ จะได้

$$v_0 v_0 = \|v_0\| \|v_1\| \cos(\theta) \quad (2.5)$$



รูปที่ 2.5 เฟรมที่เกิดจากการหมุนรอบแกน Z

จากนั้นทำการแทนที่ค่าสมการที่ 2.5 ในตัวแปรแต่ละตัวในสมการที่ 2.1 และจัดรูปสมการให้เหลือตัวแปร θ ตัวเดียวในการแทนค่าตรีโกณมิติจาก

$$x_0 = \cos(\theta) x_1 - \sin(\theta) y_1$$

$$y_0 = \sin(\theta) x_1 - \cos(\theta) y_1 \quad (2.6)$$

$$z_0 = z_1$$

และฟอร์มของเวกเตอร์หนึ่งหน่วยมีค่าเท่ากับ 1 ดังนั้นจะได้เมทริกซ์การหมุนจะมีค่า

$${}^0R_1 = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

ซึ่งเมทริกซ์การหมุนรอบแกนในสมการที่ (2.7) คือ เมทริกซ์การหมุนพื้นฐาน (pure rotation matrix) โดยเป็นการหมุนรอบแกน 2 ซึ่งสามารถแทนด้วยสัญลักษณ์ $R_{z,\theta}$ จะได้สมการเมทริกซ์การหมุนพื้นฐานรอบแกน Z เป็น

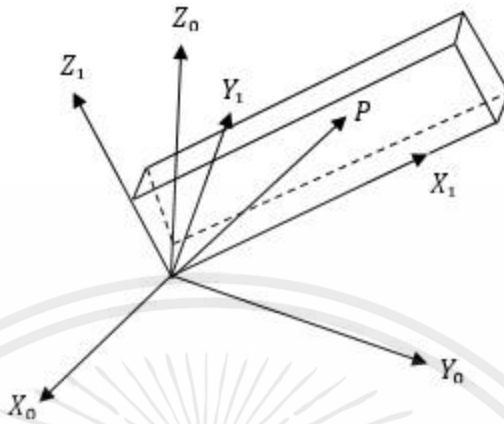
$$R_{z,\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

ในทำนองเดียวกันสามารถหาเมทริกซ์การหมุนพื้นฐานรอบแกน X และ Y ได้ดังนี้

$$R_{z,\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \quad (2.9)$$

$$R_{y,\beta} = \begin{bmatrix} \cos(\beta) & 0 & \sin(\beta) \\ 0 & 1 & 0 \\ -\sin(\beta) & 0 & \cos(\beta) \end{bmatrix} \quad (2.10)$$

ในการประยุกต์เมทริกซ์การหมุนเพื่อใช้ในการวิเคราะห์การเคลื่อนที่ของแขนกล ทำได้โดยการติดเฟรมเข้ากับข้อต่อและก้านต่อของแขนกล และทำการหาความสัมพันธ์ของเฟรมที่หมุนเปรียบเทียบกับเฟรมอ้างอิง จากรูปที่ 2.4 วัตถุแข็งเกร็งมีมุมหนึ่งติดกับจุดกำเนิดของเฟรมอ้างอิง {0} ซึ่งมีการหมุนได้รอบทิศทาง และถูกยึดติดอยู่กับที่ไม่ให้เลื่อนออกจากจุดกำเนิดของเฟรม {0} และติดเฟรม {1} เข้ากับวัตถุ โดยกำหนดให้มีจุดกำเนิดเดียวกันกับเฟรม {0} และหมุนไปพร้อมกับวัตถุ โดยมีจุด P ทำหน้าที่ในการบอกตำแหน่งของวัตถุที่สัมพันธ์กับเฟรม {0}



รูปที่ 2.6 การติดตั้งเฟรมเข้ากับวัตถุแข็งเกรง

เมื่อทำการพิจารณาที่จุด P ซึ่งถูกชี้โดยเวกเตอร์ P สมมติว่า 1P เป็นตำแหน่งของจุด P เทียบเฟรม $\{1\}$ จะได้

$$P = ux_1 + vy_1 = wz_1 \quad (2.11)$$

สามารถหาพิกัดของจุด P เทียบกับ O_0 ดังนี้

$${}^0P = \begin{bmatrix} P \cdot x_0 \\ P \cdot y_0 \\ P \cdot z_0 \end{bmatrix} \quad (2.12)$$

เมื่อทำการแทนค่าสมการ และเปรียบเทียบกับนิยามของเมทริกซ์การหมุนตามสมการ จะเขียนใหม่ได้เป็น

$${}^0P = \begin{bmatrix} x_1 & x_0 & y_1 & x_0 & z_1 & x_0 \\ x_1 & y_0 & y_1 & y_0 & z_1 & y_0 \\ x_1 & z_0 & y_1 & z_0 & z_1 & z_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \text{ หรือ } {}^0P = {}^0R_1 {}^1P \quad (2.13)$$

ในกรณีที่มีการหมุนของเฟรมมากกว่า $\{1\}$ ครั้ง สามารถนำเมทริกซ์ของการหมุนในแต่ละครั้งมาทำการคูณกัน โดยจะต้องพิจารณาตามลำดับการหมุนแต่ละครั้งเทียบกับเฟรมสุดท้าย

จากสมการเป็นเมทริกซ์การหมุนจากเฟรม {0} เป็น {1} สมมติว่ามีการหมุนเฟรมอีกครั้ง จากเฟรม {1} ไป {2} สามารถเขียนความสัมพันธ์ของจุด P เมื่อเทียบกับเฟรม 0,1,2 ดังนี้

$${}^0p = {}^0R_1 {}^1p \quad (2.14)$$

$${}^0p = {}^0R_2 {}^2p \quad (2.15)$$

$${}^0p = {}^0R_1 {}^1R_2 {}^2p \quad (2.16)$$

ดังนั้น

$${}^0R_2 = {}^0R_1 {}^1R_2 \quad (2.17)$$

ในการหาเมทริกซ์การหมุนรวมรอบแกน x, y, z เมื่อมีการหมุนเฟรมหลายครั้ง ลำดับ ของ การหมุนจะพิจารณาจากชนิดของเฟรมที่มีการหมุน โดยหากเป็นการหมุนเทียบเฟรมปัจจุบันให้ คุณเมทริกซ์การหมุนทางด้านหลัก แต่ถ้าเป็นการหมุนเทียบเฟรมอ้างอิงให้คุณเมทริกซ์การหมุนทางด้านหน้า ดังนั้นการคูณเมทริกซ์การหมุนรอบแกน x, y, z จะมีลำดับดังนี้

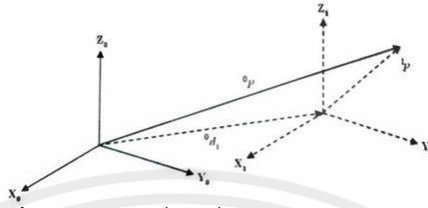
$$R = R_{x,\delta} R_{z,\alpha} R_{x,\theta} R_{z,\beta} \quad (2.18)$$

และเมื่อ α และ δ คือมุมของการหมุนรอบแกนของเฟรมอ้างอิง ส่วน θ และ β คือมุมของ การหมุนรอบแกนของเฟรมปัจจุบัน

2.2.3 การเคลื่อนที่จากการเคลื่อน

การเคลื่อนที่ของแขนกลจะเป็นการผสมระหว่างการเคลื่อนที่แบบเลื่อนและแบบหมุนใน การพิจารณาการเคลื่อนของเฟรม ดังรูปที่ 2.7 เฟรม {1} เกิดจากการเลื่อนเฟรม {0} ไปยังตำแหน่ง P โดยที่ ทั้ง 2 เฟรมนั้นมีทิศทางที่เหมือนกัน เมื่อ 1P คือตำแหน่งของจุด P เทียบกับเฟรม {1} การหา 0P หรือ ตำแหน่งของจุด P เทียบกับเฟรม {0} จะหาได้จากการบวกเวกเตอร์ 1P กับเวกเตอร์ที่ชี้ไปยัง จุดกำเนิด ของเฟรม {1} ดังสมการ

$${}^0p = {}^1p + {}^0d_1 \quad (2.19)$$



รูปที่ 2.7 การเคลื่อนที่ของเฟรมโดยการเลื่อน

แต่ในการเลื่อนจากเฟรม {0} เป็น {1} นั้นไม่จำเป็นที่จะต้องมีการกำหนดทิศทางให้มีทิศทางเดียวกัน ซึ่งการหมุนใด ๆ ของเฟรม {1} เทียบกับเฟรม {0} สามารถแทนได้ดั่งเมทริกซ์การหมุน 0R_1 ดังนั้น

$${}^0p = {}^0R_1 {}^1p + {}^0d_1 \quad (2.20)$$

วิธีข้างต้นนี้สามารถใช้สำหรับการเปรียบเทียบพิกัดของระบบที่มีจำนวนหลายเฟรมได้ แต่จะมีความยาวและซับซ้อนมากขึ้นตามจำนวนของเฟรมที่มี เนื่องจากรูปของสมการมีการพจน์ของการคูณและการบวกซ้อนกัน

2.2.4 การแปลงเอกพันธ์

จากการวิเคราะห์การหมุนใน 3 มิติที่มีการหมุนเฟรมหลาย ๆ ครั้งนั้นทำให้ได้สมการซึ่งเป็นการคูณเมทริกซ์การหมุนทั้งหมดเข้าด้วยกัน ดังสมการที่ (2.18) ขณะที่การวิเคราะห์การเลื่อนของเฟรมจะใช้วิธีการบวกเวกเตอร์เวกเตอร์ที่ชี้ไปยังจุดกำเนิดของเฟรมปัจจุบันคูณกับเมทริกซ์การหมุน ดังสมการที่ (2.19) ซึ่งทั้ง 2 สมการนี้สามารถใช้ในการอธิบายการหมุนและเลื่อนของเฟรมที่ติดต่อกับวัตถุได้ เพียงแต่ต้องมีการจัดรูปสมการใหม่ ซึ่งสามารถเขียนสมการใหม่ให้อยู่ในรูปเมทริกซ์ได้เป็น

$$T = \begin{bmatrix} R & d \\ 0 & 1 \end{bmatrix}, R \in SO(3), d \in \mathbb{R}^3 \quad (2.21)$$

เมทริกซ์ T ตามสมการที่ (2.21) คือเมทริกซ์การแปลงเอกพันธ์ (Homogeneous Transformation) ซึ่งเป็นเมทริกซ์ที่รวมเมทริกซ์การหมุนและเมทริกซ์การเลื่อนเข้าด้วยกัน เพื่อลดรูป

สมการให้สั้นลงและเหมาะสมกว่าสมการที่ (2.18) และสมการที่ (2.19) โดยการเมทริกซ์ของการแปลงเอกพันธ์จะอยู่ในรูปสมการดังนี้

$${}^{i-1}T_i = \begin{bmatrix} n_x & o_x & a_x & d_x \\ n_y & o_y & a_y & d_y \\ n_z & o_z & a_z & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n & o & a & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.22)$$

เมื่อ $n = [n_x \ n_y \ n_z]^T$, $o = [o_x \ o_y \ o_z]^T$, $a = [a_x \ a_y \ a_z]^T$ คือเวกเตอร์แทน ทิศทางของ x, y, z ตามลำดับ เทียบกับเฟรม {i} ส่วน $d = [d_x \ d_y \ d_z]^T$ คือเวกเตอร์บอกพิกัดของจุด กำเนิดของเฟรม {i} เทียบกับเฟรม {0}

การแปลงเอกพันธ์พื้นฐานจะประกอบด้วยเมทริกซ์การเลื่อนและเมทริกซ์การหมุนรวม 6 สมการ ดังนี้

$$Transl_{x,a} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Trot_{x,\phi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi & 0 \\ 0 & \sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

$$Transl_{y,b} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Trot_{y,\theta} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.24)$$

$$Transl_{z,c} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad Trot_{z,\psi} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.25)$$

ซึ่งเมทริกซ์ $Transl_{x,a}$, $Transl_{y,b}$, $Transl_{z,c}$ คือเมทริกซ์การเลื่อนในแนวแกน x, y, z ตามลำดับ และเมทริกซ์ $Trot_{x,\phi}$, $Trot_{y,\theta}$, $Trot_{z,\psi}$ คือเมทริกซ์การหมุนรอบแกน x, y, z ตามลำดับ

2.2.5 สายโซ่จลนศาสตร์

การนำหลักการวิเคราะห์การเคลื่อนที่ของวัตถุแข็งเกร็ง และการแปลงเอกพันธ์ที่ได้มีการกล่าวถึงในหัวข้อที่ผ่านมาข้างต้นนำมาประยุกต์ใช้ในการวิเคราะห์หาจลนศาสตร์การเคลื่อนที่ของแขนกลนั้นทำได้โดยการติดเฟรมให้กับข้อต่อทั้งหมดจากฐานไปสู่ส่วนปลายของแขนกล ซึ่งจากการศึกษาของศาเสรีใน 3 มิติ พบว่าการติดเฟรมให้แขนกลโดยทั่วไปจะทำให้แขนกลมีองศาเสรี 6 ตัวแปรอิสระ (3 ตัวแปรตำแหน่งและ 3 ตัวแปรทิศทาง) ทำให้ในเมทริกซ์เอกพันธ์มีตัวแปรที่ไม่ทราบค่า จำนวนมากและตัวแปรจะมีค่าเพิ่มขึ้นตามจำนวนของก้านต่อแขนกล ดังนั้นจึงทำการลดจำนวนของตัวแปรลง โดยตั้งสมมติฐานกำหนดให้แต่ละข้อต่อมีองศาเสรีเท่ากับ 1 ทำให้ฟังก์ชันของตัวแปรเหลือเพียง 1 ตัวแปรไม่ทราบค่า เรียกว่า ตัวแปรข้อต่อ (Joint variable) ซึ่งจะใช้สัญลักษณ์ q_i เมื่อ q_i มีค่าเท่ากับ θ_i ซึ่งเป็นองศาการหมุนสำหรับข้อต่อแบบหมุน หรือ q_i จะมีค่าเท่ากับ d_i ซึ่งเป็นระยะทางที่เคลื่อนที่ของข้อต่อแบบเลื่อน

จากวิธีการลดตัวแปรในข้างต้นสามารถกำหนดเมทริกซ์เอกพันธ์ที่อธิบายการเคลื่อนที่ของเฟรมสำหรับแต่ละข้อได้ดังนี้

$$A_i = A_j(q_j) \quad (2.26)$$

ดังนั้นสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าจะเขียนได้เป็น

$${}^0T = A_1(q_1)A_2(q_2) \dots A_n(q_n) \quad (2.27)$$

2.2.6 การหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับของ SCARA Robot 4DOF

จลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับจะใช้วิธีการคำนวณค่าตัวแปรของแต่ละข้อต่อโดยกำหนดตำแหน่งที่ปลายของแขนกล โดยวิธีการคำนวณจลนศาสตร์แบบย้อนกลับนี้ในบางครั้งมีได้หลายคำตอบ (Several Solution) หรือไม่สามารหาคำตอบได้จากข้อจำกัดในเรื่องของพื้นที่การทำงานของแขนกลที่จะสามารถเคลื่อนที่ไปได้ สำหรับในบริภูณานิพนธ์ฉบับนี้การวิเคราะห์และคำนวณหาจลนศาสตร์การเคลื่อนที่ของแขนกลแบบผกผันจะใช้วิธีรูปแบบปิด (Closed Form) ซึ่งจะเป็นการใช้พีชคณิตและตรีโกณมิติในการคำนวณหา ซึ่งสามารถอธิบายหลักการในการคำนวณหาได้ดังนี้

$$x^2 + y^2 = L_1^2 + L_2^2 + 2L_1L_2 \cos(\theta_2) \quad (2.28)$$

ทำการจัดรูปสมการใหม่

$$\cos(\theta_2) = \frac{x^2 + y^2 - L_1^2 + L_2^2}{2L_1L_2} \quad (2.29)$$

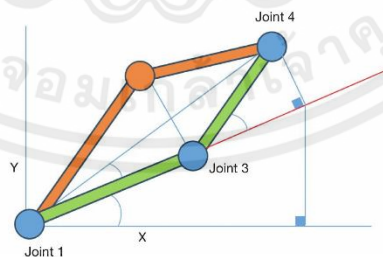
ซึ่ง $\cos(\theta_2)$ จะมีค่าอยู่ในช่วง ± 1 ซึ่งหากทำการแทนค่าตัวแปรและคำนวณตามสมการที่ (2.29) แล้วทำให้เทอมของสมการในด้านขวามีค่าอยู่นอกช่วงของ ± 1 แสดงว่าค่าตำแหน่งพิกัด x, y ที่กำหนดนั้นมีค่าอยู่นอกพื้นที่การทำงานของแขนกล ทำให้ไม่สามารถหาคำตอบที่เป็นไปได้

จากสมการที่ (2.29) สามารถหาคำตอบของสมการได้โดยการทำ \cos^{-1} ซึ่งการทำด้วยวิธีนี้จะทำให้ได้คำตอบเดียว แต่สำหรับการเคลื่อนที่ของแขนกลนั้นจะมีการเคลื่อนที่ไปในทิศทางทั้ง 4 จุดภาค ทำให้คำตอบของจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับอาจมีได้หลายคำตอบ ดังนั้นจากความสัมพันธ์ของตรีโกณมิติ

$$\begin{aligned} \text{จาก} \quad & \sin^2(\theta_2)\cos^2(\theta_2) = 1 \\ \text{เป็น} \quad & \sin^2(\theta_2) = \sqrt{1 - \cos^2(\theta_2)} \end{aligned} \quad (2.30)$$

ดังนั้นสามารถหาค่ามุมของข้อต่อที่ 3 หรือ θ_2 ได้จาก

$$\theta_2 = \text{Atan2}(\sin(\theta_2), \cos(\theta_2)) \quad (2.31)$$



รูปที่ 2.8 มุมมองด้านบนแสดงความสัมพันธ์ของเฟรมที่ 4 เทียบกับเฟรมอ้างอิง

จากรูปที่ 2.8 แสดงให้เห็นว่าค่า θ_2 ที่ได้จากการคำนวณตามสมการที่ (2.31) นั้นสามารถมีคำตอบของสมการได้ 2 คำตอบ ซึ่งเกิดจากค่าของ $\cos(\theta_2)$ ที่ทำให้เกิด $\sin(\theta_2)$ จำนวน 2 ค่า ซึ่งคำตอบ

ที่ได้นั้นจะเป็นการระบุท่าทางของแขนกลในกรณีที่แขนกลเคลื่อนที่ไปทางด้านบวกหรือเคลื่อนที่ไปทางด้านลบ

ต่อไปจะเป็นการหาค่าของมุมข้อต่อที่ 1 หรือ θ_1 จากรูปที่ 2.10 แสดงความสัมพันธ์ของมุมได้เป็น

$$\tan(\beta) = \frac{L_2 \sin(\theta_2)}{L_1 + L_2 \cos(\theta_2)} \quad (2.32)$$

$$\beta = \text{Atan}(L_2 \sin(\theta_2), L_1 + L_2 \cos(\theta_2)) \quad (2.33)$$

เมื่อ

$$\beta + \theta_1 = \text{Atan2}(x, y) \quad (2.34)$$

ดังนั้นมุมของข้อต่อที่ 1 จะมีค่าเท่ากับ

$$\theta_1 = \text{Atan2}(y, x) - \beta \quad (2.35)$$

$$\theta_1 = \text{Atan2}(y, x) - \text{Atan}(L_2 \sin(\theta_2), L_1 + L_2 \cos(\theta_2)) \quad (2.36)$$

การหาค่าของข้อต่อที่ 2 หรือ d_1 ซึ่งเป็นข้อต่อแบบเลื่อน ซึ่งสามารถคำนวณได้โดยการรวมความสูงของส่วนปลายของแขนกลเทียบกับฐาน ได้สมการเป็น

$$d_1 = z_1 + d_4 - d_5 - d_0 \quad (2.37)$$

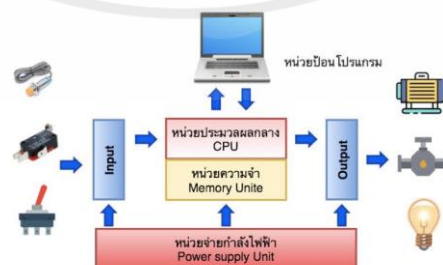
จากสมการที่ (2.31), (2.36) และ (2.37) เมื่อทำการป้อนอินพุตซึ่งเป็นค่าตำแหน่งพิกัดของส่วนปลายของแขนกลและทำการแทนค่าค่าที่ลงในตัวแปรทั้งหมดจะสามารถคำนวณหาค่าของมุมข้อต่อต่างๆ สำหรับจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับได้

2.3 โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ (PLC)

โปรแกรมเมเบิลลอจิกคอนโทรลเลอร์ หรือที่นิยมเรียกกันว่า PLC เป็นอุปกรณ์ควบคุมการทำงาน of เครื่องจักรหรือกระบวนการทำงานต่างๆ ซึ่ง PLC ถูกสร้างขึ้นมาเพื่อใช้ทดแทนวงจรรีเลย์ แบบเก่าที่มีข้อจำกัดในเรื่องของการเดินสายไฟเพื่อเปลี่ยนแปลงเงื่อนไขของการควบคุมที่มีความยุ่งยาก และเสถียรภาพของการควบคุมจะลดลงเมื่อมีการใช้งานไปในระยะเวลานาน ดังนั้นจึงมีการนำเอา PLC เข้ามาทำงานทดแทน โดยที่ PLC วงจรภายในของ PLC จะไม่มีโครโพรเซสเซอร์ที่ทำหน้าที่เป็นหน่วยประมวลผลและสั่งการทำให้มีความสามารถเปลี่ยนเงื่อนไขของการควบคุมใหม่ได้ โดยเพียงการปรับแต่งโปรแกรมการทำงานใหม่หรือเขียนโปรแกรมใหม่ นอกจากนั้น PLC จะมีส่วนที่เชื่อมต่อกับอินพุตและเอาต์พุตใช้งานได้โดยตรงโดยอุปกรณ์ตรวจรู้ เซนเซอร์ หรือสวิตช์ต่างๆ จะต่อเข้ากับอินพุต ส่วนเอาต์พุตจะต่อออกไปควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรที่เป็นเป้าหมาย และ PLC ยังสามารถใช้งานร่วมกับอุปกรณ์อื่นได้ ตัวอย่างเช่น เครื่องอ่านบาร์โค้ด (Barcode Reader) เครื่องพิมพ์ (Printer) หรือระบบ RFID เป็นต้น โดย PLC นั้นสามารถแบ่งการใช้งานออกเป็น 2 รูปแบบ ตามรูปแบบของการต่อใช้งาน ซึ่งก็คือการใช้งานแบบเดี่ยว (Standalone) และ การใช้งานแบบต่อหลายๆ ตัวเข้าด้วยกันรวมเป็นระบบเครือข่าย (Network) ซึ่งจะเห็นได้ว่า PLC นั้นค่อนข้างที่จะมีความยืดหยุ่นในการใช้งานและถือเป็นอุปกรณ์ควบคุมที่สำคัญอย่างยิ่งในโรงงานอุตสาหกรรม สำหรับการควบคุมเครื่องจักรให้มีการทำงานที่เป็นแบบอัตโนมัติ เพื่อเพิ่มประสิทธิภาพ ของกระบวนการผลิตให้มากขึ้น

2.3.1 โครงสร้างและส่วนประกอบของ PLC

โครงสร้างของเครื่องควบคุมโปรแกรมเมเบิลลอจิกคอนโทรลหรือ PLC มีส่วนประกอบที่สำคัญ ซึ่งสามารถแบ่งออกได้เป็น 5 ส่วน ซึ่งประกอบด้วย หน่วยประมวลผลกลาง หน่วยความจำ หน่วยอินพุต/เอาต์พุต แหล่งจ่ายไฟ และเครื่องป้อนโปรแกรม



รูปที่ 2.9 โครงสร้างสถาปัตยกรรมของ PLC

2.3.1.1 หน่วยประมวลผลกลาง (CPU : Center Processing Unit)

หน่วยประมวลผลกลาง หรือ CPU จะมีวงจรที่ประกอบด้วยวงจรลอจิกเกตหลายๆชนิดและไมโครโพรเซสเซอร์อยู่ภายใน ซึ่งมีทำหน้าที่ในการควบคุมประมวลผลชุดคำสั่งข้อมูลหรือโปรแกรมตามที่ได้รับมาจากอุปกรณ์ส่วนอินพุต และเก็บข้อมูลทั้งหมดนั้นไว้ที่หน่วยความจำหลังจากนั้นเมื่อทำการประมวลผลเสร็จสิ้นข้อมูลที่เป็นผลของการประมวลก็จะถูกส่งไปยังส่วนของอุปกรณ์เอาต์พุตตามเงื่อนไขของโปรแกรมที่ได้รับ โดยที่เวลาในการประมวลผลนั้นจะขึ้นอยู่กับเวลาสแกน (Scan Time) ซึ่งจะช้าหรือเร็วมีสาเหตุมาจากหลากหลายปัจจัย เช่น จำนวนข้อมูลของโปรแกรมที่ต้องประมวลผล หรือความยาวและลำดับชั้นของเงื่อนไขของโปรแกรมคำสั่ง เป็นต้น

2.3.1.2 หน่วยความจำ (Memory Unit)

หน่วยความจำ (Memory Unit) ทำหน้าที่ในการเก็บบันทึกโปรแกรมและข้อมูลต่างๆ ซึ่งในระหว่างที่ PLC มีการทำงานก็จะมีการดึงข้อมูลต่างๆ และโปรแกรมที่ได้มีการเก็บไว้ในหน่วยความจำออกมาเพื่อทำการประมวลผลการทำงาน โดยขนาดของหน่วยความจำจะถูกแบ่งออกเป็นบิตข้อมูล (Data Bit) โดยภายในหน่วยความจำ 1 บิตจะมีค่าสถานะทางลอจิกเป็น 0 หรือ 1 หน่วยความจำของ PLC จะประกอบไปด้วยหน่วยความจำ 2 ชนิด คือ หน่วยความจำชนิดชั่วคราวและ หน่วยความจำถาวร ซึ่งสามารถอธิบายหน่วยความจำแต่ละชนิดได้ดังนี้

1) หน่วยความจำชั่วคราว (RAM : Random Access Memory) ทำหน้าที่ เป็นหน่วยความจำในการเก็บโปรแกรมของผู้ใช้ และข้อมูลที่ใช้ในการทำงานของ PLC ซึ่งหน่วยความจำชั่วคราวนี้ถือว่าเป็นหน่วยความจำมาตรฐานของ PLC ที่ใช้งานในการเก็บโปรแกรมที่มีการแก้ไขบ่อยๆ ซึ่งหน่วยความจำประเภทนี้จะมีแบตเตอรี่สำรองขนาดเล็กๆ ต่อเอาไว้เพื่อใช้เลี้ยงข้อมูลที่ถูกระงับไว้และป้องกันข้อมูลที่ถูกระงับไว้ใน เมื่อเกิดไฟดับหรือไม่มีไฟเลี้ยงจ่ายให้กับ PLC

2) หน่วยความจำถาวร (ROM : Read Only Memory) ทำหน้าที่เป็นหน่วยความจำในการเก็บโปรแกรมระบบของ PLC และใช้เป็นพื้นที่สำรองข้อมูลเพื่อป้องกันกรณีที่โปรแกรม หรือข้อมูลในหน่วยความจำชั่วคราวหายไปซึ่งหน่วยความจำนี้ไม่จำเป็นต้องมีแบตเตอรี่สำรองข้อมูล แต่มีข้อจำกัดคือหน่วยความจำชนิดนี้สามารถทำการโปรแกรมใหม่ได้ แต่ไม่สามารถลบได้ อีกทั้งหากมีการเสียหายแล้วไม่สามารถซ่อมได้และมีเวลาในการเปิดข้อมูล (Time Access) ช้ากว่าหน่วยความจำแบบชั่วคราว จึงจำเป็นที่จะต้องมีการใช้หน่วยความจำทั้ง 2 ชนิดร่วมกันเพื่อให้การทำงานมีความถูกต้องสมบูรณ์

2.3.1.3 หน่วยอินพุต/เอาต์พุต (Input/Output Unit)

หน่วยอินพุต/เอาต์พุต (Input/Output Unit) จะถูกต่อเข้ากับชุดควบคุมเพื่อรับและส่งค่าสถานะและค่าสัญญาณต่างๆ ไปยังอุปกรณ์ที่มาต่อใช้งาน ซึ่งสามารถแบ่งส่วนของการทำงานออกเป็น 2 ส่วนได้ดังนี้

1) หน่วยอินพุตของ PLC จะทำหน้าที่รับสัญญาณอินพุตเข้ามาแล้วแปลงสัญญาณเพื่อที่จะส่งเข้าไปภายใน PLC ซึ่งอุปกรณ์อินพุต (Input Device) ต่างๆ ที่จะนำมาต่อกับภาคอินพุตได้นั้น เช่น Relay, Limit Switch, VI Inverter, Encoder, Temperature Controller, Photoelectric Sensor เพื่อส่งไปยัง CPU เพื่อประมวลผลตามโปรแกรมคำสั่งของผู้ใช้ โดยหน่วยอินพุตจะมีหน้าที่การทำงานดังต่อไปนี้

- แปลงระดับสัญญาณเข้าไปให้เป็นระดับสัญญาณดิจิทัลหรือแอนาล็อกที่เหมาะสมให้กับระบบการทำงานของหน่วยประมวลผลกลาง
- แบ่งสัญญาณภายนอกและภายในออกจากกัน (Isolate) เพื่อที่จะป้องกันไม่ให้หน่วยประมวลผลกลางได้รับความเสียหายแก่ปัญหาการสนสะเทือนของหน้าสัมผัส

2) หน่วยเอาต์พุตของ PLC ทำหน้าที่ส่งสัญญาณออกไปขับโหลดชนิดต่างๆ ตามเงื่อนไขที่ได้มีการเขียนโปรแกรมเอาไว้ ซึ่งหน่วยเอาต์พุตทำหน้าที่รับข้อมูลจากหน่วยประมวลผลกลางแล้วส่งต่อข้อมูลไปควบคุมอุปกรณ์ภายนอก เช่น ควบคุมหลอดไฟ มอเตอร์ และวาล์ว เป็นต้น

2.3.1.4 แหล่งจ่ายไฟ (Power Supply Unit)

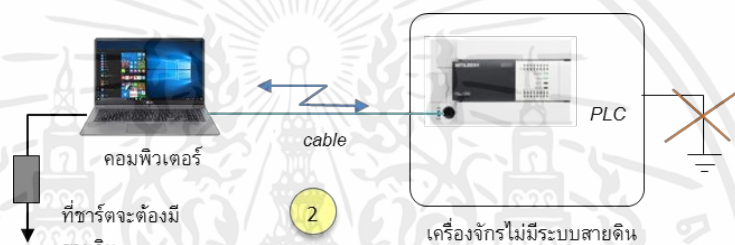
มีหน้าที่ในการจ่ายกำลังไฟฟ้าและรักษาระดับแรงดันให้กับส่วนต่างๆ ของ PLC ซึ่งแหล่งจ่ายไฟของ PLC นั้นจะแบ่งออกเป็น 2 ชุด ประกอบด้วย ชุดแรกจะมีไว้สำหรับอุปกรณ์และวงจรภายใน และชุดที่สองจะมีไว้สำหรับการต่อวงจรของส่วนอินพุตและเอาต์พุต โดยระดับแรงดันของ PLC ที่มีใช้งานนั้น จะมีอยู่ด้วยกัน 3 ระดับ คือ 24Vdc, 120Vac และ 240Vac ซึ่งจะมีหลักในการเลือกโดยเลือกใช้งานพิจารณาจากชนิด และจำนวนของอินพุต/เอาต์พุตที่ใช้

2.3.1.5 เครื่องป้อนโปรแกรม (Programming Device)

ในการสั่งงานให้ PLC มีการทำงานผู้ใช้จะต้องมีการป้อนโปรแกรมคำสั่งให้กับ PLC เพื่อให้ PLC ทำการประมวลผลโปรแกรมตามข้อมูลที่ได้รับให้เป็นไปตามเงื่อนไข และส่งสัญญาณไปขับเคลื่อนเอาต์พุตให้ทำงานตามที่ต้องการ ซึ่งในการป้อนโปรแกรมคำสั่งจะอาศัยการป้อนผ่านอุปกรณ์ที่เชื่อมต่อกับ PLC ผ่านการเชื่อมต่อในรูปแบบต่างๆ ไม่ว่าจะเป็นการเชื่อมต่อผ่านพอร์ต อนุกรม หรือการ

เชื่อมต่อผ่านเครือข่าย โดยในปริญญาบัตรฉบับนี้จะมีอุปกรณ์ที่ใช้ในการเชื่อมต่อ กับ PLC เพื่อใช้สำหรับการป้อนโปรแกรมคำสั่งอยู่ด้วยกัน 2 ประเภท คือ คอมพิวเตอร์ส่วนบุคคล และ Human Machine Interface (HMI)

1) คอมพิวเตอร์ส่วนบุคคล (Personal Computer) คือการใช้ซอฟต์แวร์คอมพิวเตอร์ในที่นี้คือโปรแกรม CX-Programmer สำหรับการติดต่อระหว่าง PLC กับคอมพิวเตอร์ส่วนบุคคลในการควบคุมป้อนหรือแก้ไขโปรแกรมคำสั่งกำหนดสัญญาณอินพุต/เอาต์พุต กำหนดลักษณะการตั้งค่าและกำหนดพื้นที่หน่วยความจำให้กับ PLC ให้สามารถทำตามเงื่อนไขที่มีการกำหนดได้



รูปที่ 2.10 ตัวอย่างการต่อคอมพิวเตอร์เข้ากับ PLC

2) Human Machine Interface (HMI) คือการควบคุมผ่านระบบ อินเทอร์เน็ตหรือผ่านหน้าจอ ที่สามารถแสดงผล สั่งการ และจัดการข้อมูล ต่างๆ ผ่านหน้าจอแสดงผล โดยใช้การติดต่อระหว่าง PLC กับ HMI เพื่อรับส่งคำสั่งสัญญาณหรือข้อมูลต่างๆ ผ่านหน่วยประมวลผลกลางแล้วนำไปควบคุมอุปกรณ์เอาต์พุตต่อไป

2.3.2 ภาษาที่ใช้ในการเขียนโปรแกรมให้กับ PLC

การเขียนโปรแกรมคำสั่งให้ PLC ทำงานตามเงื่อนไขที่ต้องการตามมาตรฐาน IEC-31131-3 ได้มีการกำหนดภาษาที่ใช้ในการเขียนโปรแกรมของ PLC ไว้ทั้งหมด 5 ภาษาดังต่อไปนี้

2.3.2.1 Ladder Diagram (LD)

เป็นภาษาที่เขียนอยู่ในรูปแบบของกราฟิก ซึ่งมีพื้นฐานมาจากวงจรควบคุม แบบรีเลย์และวงจรไฟฟ้า ซึ่ง Ladder Diagram จะประกอบด้วย ราง (Rail) เพื่อใช้สำหรับเชื่อมต่อกับอุปกรณ์อินพุต ซึ่งที่เป็นสวิตช์หน้าสัมผัสและเป็นทางผ่านของกระแส และขดลวดหรือคอยล์ (Coil) เพื่อใช้สำหรับเชื่อมต่อกับอุปกรณ์ที่เป็นเอาต์พุต

2.3.2.2 Function Block Diagram (FBD)

เป็นภาษาที่มีรูปแบบการเขียน และฟังก์ชันการทำงานเช่นเดียวกับภาษา Ladder Diagram แต่จะมีการเชื่อมต่อกันเป็นโครงข่าย โดยวิธีการเขียนนั้นจะมีลักษณะเป็นฟังก์ชันบล็อกไดอะแกรมที่มีพื้นฐานมาจากลอจิกไดอะแกรม

2.3.2.3 Sequential Flow Chart (SFC)

เป็นภาษาที่มีโครงสร้างการทำงานเป็นแบบลำดับขั้น ซึ่งจะประกอบด้วย ส่วนประกอบ 2 ส่วน ได้แก่ ลำดับขั้น (Step) ของคำสั่งในการปฏิบัติงานในแต่ละขั้นตอน และเงื่อนไข (Condition) ที่กำหนดให้กระทำคำสั่งในแต่ละขั้นตอน ซึ่งสามารถกำหนดลักษณะการทำงานของโปรแกรมให้เป็นไปตามที่ต้องการได้

2.3.2.4 Instruction List (IL)

เป็นภาษาที่เขียนอยู่ในรูปของข้อความ ซึ่งจะมีลักษณะคล้ายกับภาษา Assembly หรือภาษาเครื่อง (Machine Code) โดยที่ภายในหนึ่งคำสั่งควบคุมจะประกอบด้วย 2 ส่วน ได้แก่ ส่วนปฏิบัติการ (Operator) และส่วนที่ถูกดำเนินการ (Operand)

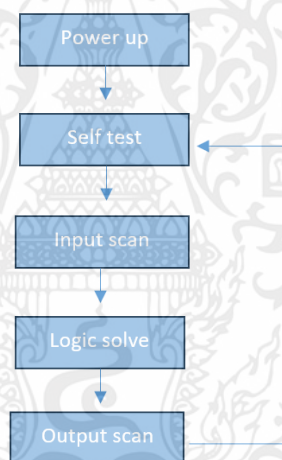
2.3.2.5 Structure Text Language (ST)

เป็นภาษาระดับสูงที่มีพื้นฐานกำหนดมาจากภาษา Pascal ซึ่งจะประกอบไปด้วย นิพจน์ และคำสั่ง ซึ่งคำสั่งโดยทั่วไปในการเขียนด้วย Structure Text จะอยู่ในรูปของคำสั่งที่เกี่ยวข้องกับการคำนวณค่าพารามิเตอร์ต่างๆ ด้วยสมการทางคณิตศาสตร์ การเลือกเงื่อนไขการทำงาน เช่น IF...THEN...ELSE เป็นต้น และคำสั่งในรูปของการทำงานแบบวนซ้ำ เช่น FOR, WHILE เป็นต้น

2.3.3 การทำงานของ PLC

การทำงานของ PLC ส่วนใหญ่แล้วจะมีลำดับขั้นของการทำงานพื้นฐานอยู่ 4 ขั้นตอน และจะมีการทำงานวนซ้ำๆกันหลายครั้งภายในเวลาหนึ่งวินาทีการทำงานของ PLC จะเริ่มต้นเมื่อมีการจ่ายแรงดันให้กับ PLC ซึ่ง PLC จะทำการตรวจสอบสถานะการทำงานของอุปกรณ์ฮาร์ดแวร์และซอฟต์แวร์ เพื่อที่จะทำการตรวจหาข้อผิดพลาดต่างๆ หากไม่มีปัญหาใดๆ PLC จะนำเอาสัญญาณอินพุต ที่เป็นข้อมูลต่างๆ จากอุปกรณ์ในหน่วยอินพุตเข้ามาเก็บไว้ในหน่วยความจำซึ่งขั้นตอนนี้จะถูกเรียกว่าสแกนอินพุต (Input Scan) หลังจากนั้น PLC จะทำการประมวลผลตามโปรแกรมคำสั่งที่ผู้เขียนเอาไว้โดยใช้

ข้อมูลจากหน่วยความจำประกอบขึ้นตอนการประมวลผลนี้เรียกว่าสแกนลอจิก (Logic Scan) ในขณะที่ PLC ทำการประมวลผลตามเงื่อนไขของโปรแกรมคำสั่งนั้นค่า เอาต์พุตของ PLC ก็จะเปลี่ยนแปลงไปตามเงื่อนไขต่างๆ ของโปรแกรมโดยที่การเปลี่ยนแปลงของค่าสถานะทางเอาต์พุตนี้จะถูกเก็บไว้ในหน่วยความจำเช่นกัน และจะเก็บไว้ที่หน่วยความจำชั่วคราวเพียงเท่านั้น เมื่อ PLC ทำการสแกนลอจิกเสร็จสิ้นแล้วข้อมูลเอาต์พุตในหน่วยความจำชั่วคราวนี้จะถูกส่งไปที่หน่วยเอาต์พุต เพื่อให้อุปกรณ์หรือโหลดที่ต่ออยู่นั้นมีสถานะทำงานหรือไม่ทำงานตามเงื่อนไขและผลลัพธ์ที่ได้จากการประมวลผล ซึ่งเรียกขั้นตอนนี้ว่า สแกนเอาต์พุต (Output Scan) ซึ่ง เมื่อสิ้นสุดการสแกนเอาต์พุต PLC ก็จะกลับไปเริ่มต้นการทำงานใหม่ ซึ่งกระบวนดังกล่าวนี้จะใช้เวลา อยู่ที่ 5 - 10 ms หรือเร็วกว่าขึ้นอยู่กับความสามารถและความเร็วในการทำงานของหน่วย ประมวลผลกลางของ PLC ด้วย



รูปที่ 2.11 วงรอบลำดับการทำงานของ PLC

2.3.4 การสแกนอินพุตและเอาต์พุต

เมื่อถึงขั้นตอนของการสแกนอินพุตอุปกรณ์อินพุตต่างๆ ที่ต่อเข้ากับ PLC นั้นถูกสแกนค่าข้อมูลหรือค่าสถานะต่างๆ จะถูกนำไปเก็บไว้ในหน่วยความจำในทางกลับกันเมื่อถึงขั้นตอนของการสแกนเอาต์พุตอุปกรณ์เอาต์พุตที่ต่อกับ PLC ที่ถูกสแกนจะทำการคัดลอกข้อมูลจากในหน่วยความจำส่งออกไปให้กับอุปกรณ์เอาต์พุต ซึ่งการสแกนลอจิกหรือในการประมวลผลโปรแกรม PLC จะใช้ค่าหรือข้อมูลที่อยู่ภายในหน่วยความจำเท่านั้น โดยที่จะไม่สนใจค่า หรือข้อมูลจริงของอินพุตและเอาต์พุตในขณะนั้นเลย โดยที่ต้องเป็นเช่นนั้นก็เพราะว่าถ้าค่าข้อมูลที่ใช้ในการประมวลผลทั้งทางด้านอินพุตและเอาต์พุตมีการเปลี่ยนแปลงไปตลอดเวลาจะทำให้ PLC มีการทำงานที่ช้ามากเพราะแทนที่ PLC จะประมวลผลโปรแกรม

ในแต่ละคำสั่งให้จบทั้งโปรแกรมไปแต่กลับที่จะต้องมาคัดลอกข้อมูลใหม่ที่มีการเปลี่ยนแปลงทุกครั้ง ซึ่งทำให้การประมวลผลโปรแกรมไม่สามารถทำได้โดยง่าย

2.3.5 ข้อมูลในหน่วยความจำของ PLC

PLC จะมีการเก็บข้อมูลของค่าสถานะจากการวัดของอุปกรณ์ตรวจรู้หรือเซนเซอร์ หรือค่าสัญญาณต่างๆ จากหน่วยอินพุตและหน่วยเอาต์พุตเอาไว้ในหน่วยความจำของ PLC ซึ่งข้อมูลที่ถูกจัดเก็บนั้นจะถูกแบ่งออกเป็น 3 ประเภท โดยอ้างอิงตามขนาดของข้อมูล ซึ่งได้แก่ บิต (Bit), ดิจิต (Digit), ไบต์ (Byte) และเวิร์ด (Word) ซึ่งสามารถอธิบายความสัมพันธ์ได้ดังนี้

1 เวิร์ด = 2 ไบต์

1 ไบต์ = 2 ดิจิต

1 ดิจิต = 4 บิต

2.4 ระบบเลขฐาน

เนื่องจากในปฏิญญาพันธบัตรฉบับนี้ใช้ PLC เป็นเครื่องควบคุมหลัก ซึ่งการประมวลผลข้อมูล เป็นการประมวลผลที่อยู่ในรูปของข้อมูลแบบดิจิตอล เช่นเดียวกับคอมพิวเตอร์ หรือเครื่องควบคุม อื่น ๆ ที่มีใช้งานอยู่ในปัจจุบัน ดังนั้นระบบเลขฐานจึงเป็นพื้นฐานที่สำคัญในการแปลงข้อมูลต่าง ๆ ให้สามารถเข้าถึงร่วมกันได้ ซึ่งระบบเลขฐานที่ถูกนำมาใช้งานบน PLC ในการเขียนโปรแกรมคำสั่ง ควบคุม หรือใช้งานในด้านต่าง ๆ นั้น จะมีอยู่ด้วยกัน 4 ระบบ ได้แก่

2.4.1 ระบบเลขฐาน 2 (Binary Number System)

ข้อมูลสำหรับระบบเลขฐาน 2 จะประกอบไปด้วยเลข 2 ตัว คือ 0 กับ 1 ซึ่งเป็นเลขฐานที่ใช้สำหรับการประมวลผลในภาษาคอมพิวเตอร์ เพราะจากการที่อุปกรณ์ทางไฟฟ้าหรือวงจรดิจิตอลนั้นมีสถานะการทำงานอยู่ด้วยกัน 2 สถานะคือ เปิด กับ ปิด ซึ่งเทียบได้กับข้อมูล 0 กับ 1 ของเลขฐาน 2 ตามลำดับ ซึ่งการใช้เลขฐาน 10 อาจไม่เป็นที่นิยมนักในสำหรับใช้งานกับคอมพิวเตอร์และอาจเกิดปัญหาอื่นๆขึ้นได้หรือแม้แต่ในอุปกรณ์ทางไฟฟ้าก็อาจจะต้องทำให้มีการแบ่งสถานะออกเป็น 10 สถานะ ซึ่งทำให้เกิดความยุ่งยาก ซึ่งในการเก็บข้อมูลด้วยระบบเลขฐาน 2 นี้ จะมีการจัดเก็บโดยการเก็บเป็นกลุ่มเลขฐาน 2 จำนวนหลายๆบิตนำมาต่อกันโดยที่จะมีความยาวตามขนาดของข้อมูลที่ต้องการจัดเก็บ

2.4.2 ระบบเลขฐาน 8 (Octal Number System)

เป็นระบบเลขฐานที่ประกอบด้วยตัวเลขจำนวน 8 ตัว ประกอบด้วยเลข 0, 1, 2, 3, 4, 5, 6, 7 ซึ่งระบบเลขฐาน 8 นั้น เป็นระบบเลขฐานที่ถูกนำมาใช้แทนระบบเลขฐาน 2 สำหรับในการเก็บข้อมูล ที่มีขนาดใหญ่ขึ้น

2.4.3 ระบบเลขฐาน 10 (Decimal Number System)

เป็นระบบเลขฐานที่ถูกนำมาใช้เป็นมาตรฐานที่สามารถพบเห็นได้ทั่วไปในชีวิตประจำวัน ซึ่งประกอบด้วยเลขจำนวน 10 ตัว ประกอบด้วยเลข 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 ซึ่ง ระบบเลขฐาน 10 นี้ เป็นระบบเลขฐานที่สามารถนำไปคำนวณค่าทางคณิตศาสตร์ได้ง่ายกว่าระบบเลข ฐานทั้งหมด MI

2.4.4 ระบบเลขฐาน 16 (Hexadecimal Number System)

เป็นระบบเลขฐานที่ถูกนำมาใช้แสดงแทนระบบเลขฐาน 2 ในการเก็บข้อมูลที่มีจำนวนมาก สำหรับการประมวลผลทางคอมพิวเตอร์ เพราะถ้าใช้ระบบเลขฐาน 2 ในการเก็บข้อมูลที่มีจำนวนมากๆแล้วนั้น จะทำให้ต้องเขียนแทนด้วยเลขฐาน 2 ที่มีขนาดหลายบิต แต่หากเขียนแทนด้วย เลขฐาน 16 จะเป็นการลดรูปทำให้เหลือเพียงจำนวนไม่กี่หลัก ซึ่งใน PLC ก็ได้มีการใช้ระบบเลขฐาน 16 เป็นหลักในการแสดงค่าของข้อมูลที่ถูกเก็บไว้ในหน่วยความจำ เพื่อแทนเลขฐาน 2 ที่ใช้ในการ ประมวลผล ด้วยเช่นกัน ระบบเลขฐาน 16 จะประกอบไปด้วยตัวเลขจำนวน 10 เลข และตัวอักษรอีก 6 ตัว ซึ่ง ประกอบด้วย 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 และ A, B, C, D, E, F

ตารางที่ 2.2 ตารางตัวอย่างความสัมพันธ์ระหว่างเลขฐานแต่ละระบบ

เลขฐาน 2	เลขฐาน 8	เลขฐาน 10	เลขฐาน 16
0000	0	0	0
0001	1	1	1
0010	2	2	2
0011	3	3	3
0100	4	4	4
0101	5	5	5
0110	6	6	6

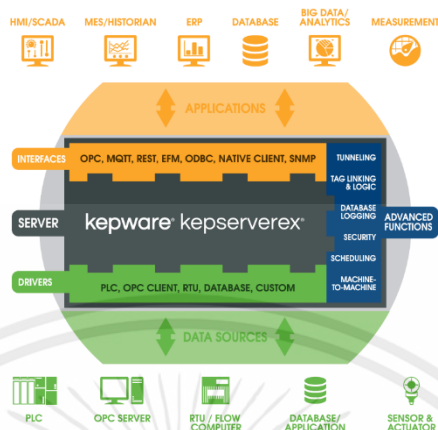
0111	7	7	7
1000	10	8	8
1001	11	9	9
1010	12	10	A
1011	13	11	B
1100	14	12	C
1101	15	13	D
1110	16	14	E
1111	17	15	F

2.5 OLE For Process Control (OPC Server)

OLE for Process Control หรือ OPC Server ทำหน้าที่ในการเป็นตัวกลางในการติดต่อสื่อสารข้อมูลระหว่างอุปกรณ์ทางด้านอุตสาหกรรม เช่น เครื่องจักร HMI หรือ PLC กับระบบของผู้ใช้ ซึ่งใช้มาตรฐานการสื่อสารในรูปแบบของ server และ client โดย OPC Server จะทำการแปลงโปรโตคอลการเชื่อมต่อของอุปกรณ์ต่างๆ ที่เชื่อมต่ออยู่นั้นให้เป็นโปรโตคอลเดียวกัน คือ OPC โปรโตคอลเพื่อให้สามารถรับ-ส่งและใช้ข้อมูลต่างๆ ร่วมกันได้.

2.5.1 KEPServerEX6

ในปฏิญานิพนธ์ฉบับนี้ จะมีการใช้ OPC Server เชื่อมต่อระหว่าง PLC กับ คอมพิวเตอร์ สำหรับการรับส่งข้อมูลตำแหน่งส่วนปลายของแขนกลที่อยู่ในรูปของ G-Code บน ฐานข้อมูลซึ่งอยู่บนตารางของโปรแกรม Microsoft Excel กับหน่วยความจำของ PLC เพื่อใช้ในการป้อนโปรแกรมคำสั่ง สำหรับการควบคุมแขนกล SCARA Robot 4DOF ให้มีการเคลื่อนที่ไปในหลายๆ ตำแหน่ง โดยเลือกใช้ KEPServerEX6 เป็น OPC Server ซึ่งในการใช้งานจะต้องทราบถึงโครงสร้าง



รูปที่ 2.12 การประยุกต์ใช้ KEP ServerEX6 กับงานด้านต่าง ๆ

2.5.2 Dynamic Data Exchange (DDE)

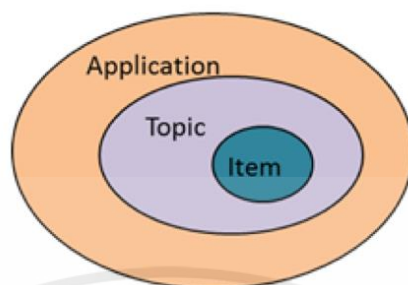
ในการดึงข้อมูลจากฐานข้อมูลบนโปรแกรม Microsoft Excel นั้นจะไม่สามารถเชื่อมต่อกัน OPC Server โดยตรงได้ ซึ่งจะต้องมีเครื่องมือที่เป็นตัวกลางในเชื่อมโยงข้อมูลจากบนโปรแกรม Microsoft Excel ไปยัง OPC Server เพื่อเก็บข้อมูลไปที่หน่วยจำ PLC หรือบนโปรแกรม CX-Programmer แบบเรียลไทม์ ซึ่ง Dynamic Data Exchange หรือ DDE นั้นเป็นมาตรฐานในการติดต่อสื่อสารระหว่างโปรแกรมที่ทำงานบนระบบปฏิบัติการ Windows ซึ่งการแลกเปลี่ยนข้อมูลจะประกอบด้วยส่วนประกอบที่สำคัญ 3 ส่วน ได้แก่ โปรแกรมที่ติดต่อด้วย, กลุ่มที่ข้อมูลอยู่ และข้อมูลที่ต้องการ และสามารถแบ่งชนิดของการเชื่อมต่อได้ 2 วิธี คือ

2.5.2.1 แบบ Hot Link

เป็นการเชื่อมต่อผ่าน Dynamic Data Exchange (DDE) เพื่อใช้สำหรับอ่านข้อมูลเพียงอย่างเดียว (Read only) ซึ่งจะมีการเชื่อมต่อตลอดเวลาตลอดการใช้งาน

2.5.2.2 แบบ Cold Link

เป็นการเชื่อมต่อผ่าน Dynamic Data Exchange (DDE) ที่สามารถอ่านและเขียนข้อมูลลงไปได้ ซึ่งจะถูกกำหนดเงื่อนไขหรือเวลาในการเชื่อมต่อเอาไว้



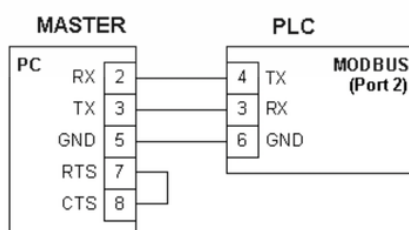
รูปที่ 2.13 ส่วนประกอบของระบบ Dynamic Data Exchange (DDE)

2.5.3 การติดต่อสื่อสารผ่านพอร์ตอนุกรม RS232

สำหรับการเชื่อมต่อคอมพิวเตอร์ เพื่อใช้ในการรับ-ส่งข้อมูลกับ PLC ผ่าน OPC Server นั้น จะใช้การติดต่อสื่อสารผ่านพอร์ตอนุกรมโดยใช้ RS232 ซึ่ง RS232 คือมาตรฐานการรับ-ส่งข้อมูลดิจิทัลแบบอนุกรม (Serial Communication) ที่มีการรับส่งข้อมูลกันเป็นแบบอะซิงโครนัส 2 ทิศทาง หรือ Full duplex กล่าวคืออุปกรณ์ที่ติดต่อสื่อสารกันอยู่นั้นสามารถรับและส่งข้อมูลได้พร้อมๆ กันในเวลาเดียวกัน ซึ่งการรับ-ส่งข้อมูล สำหรับสถานะลอจิก 0 นั้นจะมีระดับสัญญาณตั้งแต่ 3 โวลต์ ไปจนถึง 15 โวลต์ และสำหรับสถานะลอจิก 1 จะมีระดับแรงดันที่ -3 โวลต์ ไปจนถึง -15 โวลต์ ซึ่งการนำไปใช้งานกับวงจรดิจิทัลนั้นจึงต้องมีอุปกรณ์ในการแปลงระดับแรงดันให้เหลือเพียง 0-5 โวลต์ สำหรับภาคส่งข้อมูล และสำหรับภาครับข้อมูลก็จะต้องเปลี่ยนระดับแรงดันให้สูงกว่า 43 หรือต่ำกว่า -3 ซึ่งในการรับ-ส่งข้อมูลจะประกอบไปด้วยสายเชื่อมต่อทั้งหมด 3 เส้น ดังรูปที่ 2.14 ซึ่งสามารถ อธิบายได้ ดังนี้

- 1) Tx (Transmit Data) คือ สายที่ใช้สำหรับการส่งข้อมูล
- 2) Rx (Receive Data) คือ สายที่ใช้สำหรับรับข้อมูล
- 3) GND (Signal Ground) คือ สายกราวด์ หรือสายอ้างอิงในการเทียบศักย์แรงดันไฟ ฟา

ที่ 0 โวลต์



รูปที่ 2.14 ตัวอย่างการเชื่อมต่อคอมพิวเตอร์เข้ากับ PLC ผ่านพอร์ตอนุกรม RS232

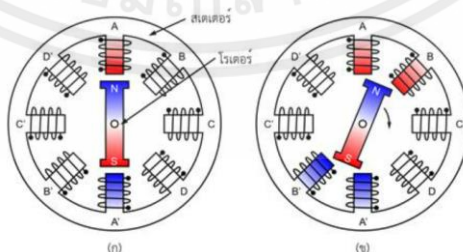
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 มอเตอร์แบบสเต็ป

มอเตอร์แบบสเต็ป หรือที่เรียกว่าสเต็ปปีงมอเตอร์ (Stepping Motor) คือมอเตอร์ไฟฟ้าชนิดหนึ่ง ที่ถูกนำมาประยุกต์ใช้สำหรับการควบคุมที่มีการกำหนดตำแหน่ง และทิศทางให้กับกลไกการทำงานของเครื่องจักรที่ต้องการความแม่นยำและความแน่นอนที่สูง ซึ่งการหมุนของมอเตอร์ชนิดนี้ จะมีความแตกต่างจากมอเตอร์ชนิดอื่นๆ นั่นก็คือว่าการหมุนของมอเตอร์แบบสเต็ปนั้นจะมีการหมุนที่ แบ่งออกเป็นทีละขั้น หรือสเต็ป โดยการหมุนไปแต่ละสเต็ปนั้นจะมีระยะที่เท่า ๆ กัน เช่น หมุนไป สเต็ปละ 0.9 องศา, 1.8 องศา, 3.6 องศา เป็นต้น ซึ่งหลักการทำงานของมอเตอร์ชนิดนี้ จะใช้สัญญาณ พัลส์ในการขับเคลื่อน มอเตอร์ให้มีการหมุนมอเตอร์ชนิดนี้นั้นสามารถหมุนได้รอบแกน 360 องศา โดยที่มุมในการหมุนและความเร็วในการหมุนนั้นจะมีค่าตามสัดส่วนที่สัมพันธ์กับสัญญาณพัลส์ที่จ่ายให้กับมอเตอร์

2.6.1 โครงสร้างและหลักการทำงานของมอเตอร์แบบสเต็ป

โครงสร้างภายในของมอเตอร์แบบสเต็ป จะแสดงดังรูปที่ 2.15 ประกอบด้วยสเตเตอร์ (Stator) ที่มีขดลวดพันอยู่ และส่วนที่เป็นแกนหมุนหรือโรเตอร์ (Rotor) เป็นแม่เหล็กถาวร โดยหลักการทำงานของมอเตอร์แบบสเต็ปคือ เมื่อจ่ายกระแสให้ไหลผ่านขดลวดสเตเตอร์จะทำให้เกิดการเหนี่ยวนำของสนามแม่เหล็กขึ้นแรงดึงดูดที่เกิดขึ้นจากสนามแม่เหล็กนี้จะทำให้เกิดการหมุนของโรเตอร์ ซึ่งกระแสที่จ่ายให้กับมอเตอร์แบบสเต็ปนั้นจะอยู่ในรูปแบบของสัญญาณพัลส์ทำให้ต้องใช่วงจรอิเล็กทรอนิกส์หรือ วงจรขับมอเตอร์ในการกำเนิดสัญญาณก่อนป้อนให้กับมอเตอร์ โดยเมื่อทำการจ่ายกระแสไปยังขดลวดแต่ละขดอย่างต่อเนื่องจะมีผลทำให้มอเตอร์แบบสเต็ปนั้นมีการหมุนอย่างต่อเนื่อง และหากต้องการให้มอเตอร์มีการหมุนไปตามจำนวนของสเต็ปที่กำหนดนั้น สามารถทำได้โดยการจ่ายกระแสที่เป็นสัญญาณพัลส์ผ่านขดลวดตามจำนวนสเต็ปที่ต้องการ



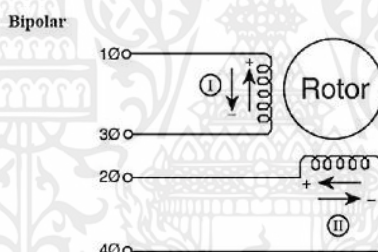
รูปที่ 2.15 มอเตอร์แบบสเต็ปขณะหยุดนิ่ง และขณะหมุนไปได้ครึ่งสเต็ป

2.6.2 ชนิดของมอเตอร์แบบสเต็ป

มอเตอร์แบบสเต็ปสามารถแบ่งออกได้เป็น 2 ชนิด คือ ชนิดยูนิโพลาร์ (Unipolar) และ ชนิดไบโพลาร์ (Bipolar) ซึ่งทั้งสองชนิดเป็นมอเตอร์แบบ 2 เฟส โดยแต่ละแบบจะมีโครงสร้างและจุดเชื่อมต่อที่แตกต่างกัน โดยในปฏิญญาฉบับนี้ขออธิบายเพียงเฉพาะมอเตอร์แบบสเต็ปชนิดไบโพลาร์ ซึ่งมีรายละเอียด ดังนี้

2.6.2.1 มอเตอร์แบบสเต็ปชนิดไบโพลาร์

มอเตอร์แบบสเต็ปชนิดไบโพลาร์จะมีโครงสร้างที่มีขดลวดภายในจำนวน 2 ชุด แบ่งออกเป็นชุด A และชุด B และไม่มีจุดต่อสำหรับต่อแรงดันไฟบวกการหมุนของมอเตอร์นั้นจะทำการจ่ายกระแสกระตุ้นผ่านเฟสแต่ละเฟสของขดลวดทั้ง 2 ชุด ทำให้จึงมีสายต่อออกมาใช้งานเพียงจำนวน 4 เส้น โดยโครงสร้างของมอเตอร์แบบสเต็ปชนิดไบโพลาร์ แสดงดังรูปที่ 2.16



รูปที่ 2.16 โครงสร้างของมอเตอร์แบบสเต็ปชนิดไบโพลาร์

2.6.3 การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป

การควบคุมและสั่งการให้มอเตอร์แบบสเต็ปมีการหมุนหรือเคลื่อนที่ไปที่ละสเต็ปๆ ได้ นั้นสามารถทำได้โดยการจ่ายกระแสเข้าไปที่ขดลวดสเตเตอร์อย่างทีกล่าวไปข้างต้น ซึ่งการจ่ายกระแสนั้นจะทำได้ในลักษณะที่เป็นลำดับและมีการวนรอบการทำตามเดิม โดยวิธีการจ่ายกระแสนั้นสามารถแบ่งออกเป็น 3 แบบ ได้แก่ แบบฟลูสเต็ม 1 เฟส, แบบฟลูสเต็ม 2 เฟส และแบบครึ่งสเต็ป

2.6.3.1 แบบฟลูสเต็ม 1 เฟส (Full Step 1 Phase)

การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป แบบฟลูสเต็ม 1 เฟส จะเป็นการจ่ายกระแสเพื่อขับมอเตอร์ไปที่ละขด โดยเรียงลำดับจากขดลวดที่ 1 2 3 4 หรือ 4 3 2 1 ตามทิศทางการหมุน

ตารางที่ 2.3 ลำดับการจ่ายกระแสกระตุ้นแบบฟูลสเต็ป 1 เฟส

สเต็ปที่	เฟส 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน
5	ทำงาน	-	-	-
6	-	ทำงาน	-	-

2.6.3.2 แบบฟูลสเต็ป 2 เฟส (Full Step 2 Phase)

การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ป แบบฟูลสเต็ป 2 เฟส นั้นจะ เป็นการกระตุ้นไปที่ละขดลวดเช่นเดียวกับแบบ 1 เฟส แต่จะแจกจ่ายกันตรงที่แบบ 2 เฟสนั้นจะ กระตุ้น ขดลวดทีละ 2 ขด ที่อยู่ลำดับใกล้เคียงกัน เช่น 1 และ 2 2 และ 3 3 และ 4 4 และ 1 หรือ 1 และ 4 4 และ 3 3 และ 2 2 และ 1 ตามทิศทางการหมุนของมอเตอร์ ลำดับการกระตุ้นสามารถดูได้จากตาราง

ตารางที่ 2.4 ลำดับการจ่ายกระแสกระตุ้นแบบฟูลสเต็ป 2 เฟส

สเต็ปที่	เฟส 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน
5	ทำงาน	ทำงาน	-	-
6	-	ทำงาน	-	-

2.6.3.3 แบบครึ่งสเต็ป (Half Step)

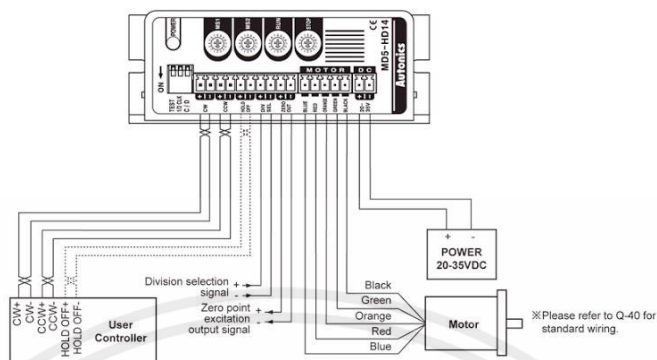
การจ่ายกระแสกระตุ้นให้กับมอเตอร์แบบสเต็ปเป็นการจ่ายกระแสไปที่ขดลวดที่ 1 และขดลวดที่ 2 สลับกันไป โดยการกระตุ้นแบบนี้จะเป็นการเพิ่มจำนวนรอบ ของการกระตุ้นให้มากขึ้น ทำให้มอเตอร์แบบสเต็ปหมุนเร็วขึ้น และมีความแม่นยำที่มากขึ้น ซึ่งลำดับการกระตุ้นตามตารางที่ 2.6

ตารางที่ 2.5 ลำดับการจ่ายกระแสกระตุ้นแบบครึ่งสเต็ป

สเต็ปที่	เฟส 1	เฟสที่ 2	เฟสที่ 3	เฟสที่ 4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	ทำงาน	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	-
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน
9	ทำงาน	-	-	ทำงาน
10	ทำงาน	ทำงาน	-	-

2.6.4 วงจรขับมอเตอร์

ในการขับมอเตอร์โดยปกติจะป้อนแรงดันไฟตรงให้โดยตรง ทำให้มอเตอร์ทำงานเต็มกำลัง ซึ่งอาจมีความเร็วมากเกินไป ส่งผลให้ไม่เกิดความไม่เสถียรในการทำงาน และอาจทำให้เกิดความยากและความผิดพลาดในการควบคุมองศาการหมุนของมอเตอร์ ดังนั้นการปรับความเร็วของมอเตอร์จึงใช้วิธีลดแรงดันไฟฟ้าที่ป้อนให้กับมอเตอร์ วิธีที่นิยมคือการป้อนพัลส์ไปขับมอเตอร์แทน แล้วปรับความกว้างพัลส์ช่วงบวก เพื่อให้ได้ค่าแรงดันเฉลี่ยตามต้องการ วิธีการนี้เรียกว่า พัลส์วิดธ์มอดูเลเตอร์ (PWM) โดยการควบคุมอาศัยหลักการทำงานของอุปกรณ์สารกึ่งตัวนำต่างๆ เช่น ตรีเล็ท ทรานซิสเตอร์ ไอซี หรือมอสเฟต เป็นต้น ซึ่งวงจรขับมอเตอร์ที่ใช้ในปริญญาโทฉบับนี้ คือ Microstep driver เป็นไดรเวอร์สเต็ปเปอร์มอเตอร์แบบสองเฟสสร้างขึ้นจาก IC TB6600HG ทำหน้าที่ในการควบคุมความเร็วและทิศทางของมอเตอร์ที่สามารถจ่ายกระแสได้ 0-4 แอมป์ และจ่ายแรงดันไฟกระแสตรงได้ 9-24 โวลต์



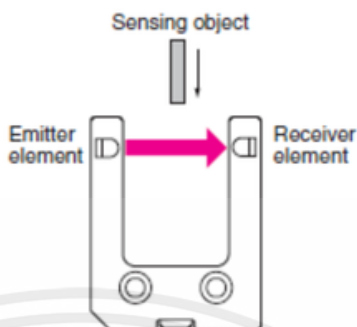
รูปที่ 2.17 การต่อใช้งาน Micro step Driver กับเครื่องควบคุม

2.7 โฟโตอิเล็กทรอนิกส์เซนเซอร์

โฟโตอิเล็กทรอนิกส์เซนเซอร์ คือเซนเซอร์ที่ใช้สำหรับการตรวจจับการเคลื่อนไหวของวัตถุ หรือตรวจสอบขนาดรูปร่างของวัตถุ โดยที่เซนเซอร์ชนิดนี้ทำงานโดยที่ไม่ต้องมีการสัมผัสตัววัตถุ แต่จะอาศัยหลักการในการส่งและรับแสงระหว่างภาคส่งแสง (Emitter) ภาคตัวรับแสง (Receiver) ลักษณะของการตรวจจับนั้นเกิดจากการที่ลำแสงจากตัวส่งแสงส่งไปสะท้อนกับวัตถุหรือถูกขวางกั้นด้วยวัตถุ ส่งผลให้ตัวรับแสงนั้นมีสภาวะการทำงานที่เปลี่ยนไป และทำให้เกิดสัญญาณเอาต์พุตเพื่อนำไปใช้งานต่อไป

2.7.1 หลักการโฟโตอิเล็กทรอนิกส์เซนเซอร์ที่ใช้ในปริยญาณิพนธ์

โฟโตอิเล็กทรอนิกส์เซนเซอร์ที่นำมาใช้ในปริยญาณิพนธ์ฉบับนี้คือ PM-K44 เป็นโฟโตอิเล็กทรอนิกส์เซนเซอร์ที่มีขนาดเล็กมีวงจรรขยายสัญญาณในตัว ลักษณะโครงสร้างเป็นรูปตัว U หรือก้ามปู ซึ่งมีวงจรรควบคุมที่เป็นทั้งภาคส่งแสง (Emitter) และภาครับแสง (Receiver) หลักการทำงานของเซนเซอร์นั้นจะอาศัยหลักการเดียวกันกับโฟโตอิเล็กทรอนิกส์เซนเซอร์ชนิดลำแสงผ่านตลอด (Through Beam Photoelectric Sensor) ซึ่งมีหลักการคือให้ภาคส่งแสง และภาครับแสงนั้นอยู่ตรงข้ามกันในขณะที่มีสภาวะปกติภาครับแสงจะสามารถรับลำแสงที่ส่งออกมาจากตัวส่งได้ตลอดเวลา ซึ่งเมื่อมีวัตถุหรือชิ้นงานผ่านหน้าเซนเซอร์ และไปขวางลำแสงที่ส่งมาจากภาคส่งแสงทำให้ภาครับแสงไม่สามารถรับสัญญาณได้ก็จะเป็นการทำให้วงจรภายในของเซนเซอร์นั้นตรวจรู้ได้ว่ามีวัตถุหรือชิ้นงานขวางอยู่ และจะทำให้สถานะเอาต์พุตของภาครับแสงเปลี่ยนไป ซึ่งจะเรียกลักษณะการทำงานแบบนี้ ว่า Dark On หรือ Dark Operate



รูปที่ 2.18 การทำงานของโฟโตอิเล็กทริกเซนเซอร์แบบ U-type

2.8 ความสัมพันธ์ระหว่างระยะทางและอัตราเร็ว

2.8.1 ระยะทาง (Distance)

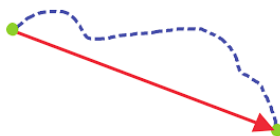
การเคลื่อนที่ของวัตถุ เช่น รถยนต์ สัตว์ วัตถุตกในอากาศ พบว่าตำแหน่งของวัตถุมีการเปลี่ยนไปจากเดิม หรือกล่าวได้ว่าวัตถุจะเลื่อนจากตำแหน่งเดิมไปยังตำแหน่งใหม่ ซึ่งจะเรียกการเคลื่อนที่ เช่นนี้ว่า การเคลื่อนที่แบบเลื่อนตำแหน่ง (translation motion) ถ้าเราทราบตำแหน่งเริ่มต้นเส้นทางการเคลื่อนที่และตำแหน่งสุดท้ายของการเคลื่อนที่ ก็จะได้ระยะทางจากความยาวตามเส้นทางการเคลื่อนที่นั้น ระยะทางใช้สัญลักษณ์ “ s ” เป็นปริมาณสเกลลาร์ คือมีแต่ขนาดเพียงอย่างเดียว แต่ไม่บอกทิศทางมีหน่วยเป็น เมตร (m)

2.8.2 การกระจัด (Displacement)

เมื่อวัตถุมีการเคลื่อนที่จากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งการบอกตำแหน่งใหม่เทียบกับตำแหน่งเดิม เพื่อให้เข้าใจได้ชัดเจนต้องบอกทั้งระยะทางและทิศทางปริมาณที่บอกให้ทราบถึงการเปลี่ยนตำแหน่ง เรียกว่า การกระจัด

การกระจัด ใช้สัญลักษณ์ “ s' ” เป็นปริมาณเวกเตอร์ มีหน่วยเป็น เมตร (m)

การกระจัดหาได้จากเส้นตรงที่เขียนหัวลูกศรกำกับโดยลากจากจุดเริ่มต้นไปยังจุดสุดท้ายของการเคลื่อนที่ ความยาวของเส้นตรงแทนขนาดของการกระจัดและทิศที่หัวลูกศรชี้จะแทนทิศของการกระจัด



รูปที่ 2.19 ระยะทางและการกระจัด

2.8.3 อัตราเร็ว (Speed)

เมื่อวัตถุมีการเคลื่อนที่ปริมาณที่บอกได้ว่าวัตถุนั้นเคลื่อนที่ได้เร็วมากหรือน้อยนั้น คือ อัตราเร็ว ซึ่งโดยทั่วไป อัตราเร็ว หมายถึง ระยะทางที่วัตถุเคลื่อนที่ได้ในหนึ่งเวลา หรือ อัตราการเปลี่ยน ระยะทาง อัตราเร็วเป็นปริมาณสเกลลาร์ ใช้สัญลักษณ์ “ v ” มีหน่วย เมตรต่อวินาที (m/s)

$$\text{อัตราเร็ว} = \frac{\text{ระยะทาง}}{\text{เวลา}}$$

$$v = \frac{s}{t} \quad (2.38)$$

เมื่อ v คือ อัตราเร็วของวัตถุ มีหน่วย เมตรต่อวินาที (m / s)

s คือ ระยะทางของวัตถุ มีหน่วย เมตร (m)

t คือ เวลาที่วัตถุใช้ในการเคลื่อนที่ มีหน่วย วินาที (s)

อัตราเร็วสามารถแบ่งออกได้ ดังนี้

อัตราเร็วเฉลี่ย (v_{av}) คือ ระยะทางที่วัตถุเคลื่อนที่ได้ในหนึ่งหน่วยเวลา (ในช่วงเวลาหนึ่งที่ กำลังพิจารณาเท่านั้น) สามารถเขียนเป็นสมการได้

$$v_{av} = \frac{\Delta s}{\Delta t} \quad (2.39)$$

v_{av} คือ อัตราเร็วของวัตถุ มีหน่วย เมตรต่อวินาที (m / s)

Δs คือ ระยะทางของวัตถุ มีหน่วย เมตร (m)

Δt คือ เวลาที่วัตถุใช้ในการเคลื่อนที่ มีหน่วย วินาที (s)

อัตราเร็วขณะใดขณะหนึ่ง (v_t) คือ ระยะทางที่เคลื่อนที่ได้ในหนึ่งหน่วยเวลา เมื่อช่วงเวลา ที่เคลื่อนที่น้อยมากๆ สามารถเขียนสมการได้

$$v_t = \frac{\Delta s}{\Delta t} \quad (2.40)$$

เมื่อ v_t คือ อัตราเร็วขณะใดขณะหนึ่งมีหน่วยเป็น เมตรต่อวินาที (m / s)

อัตราเร็วคงที่ (v) คือ การบอกให้ทราบว่าวัตถุมีการเคลื่อนที่อย่างสม่ำเสมอไม่ว่าจะพิจารณาในช่วงเวลาใด ๆ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

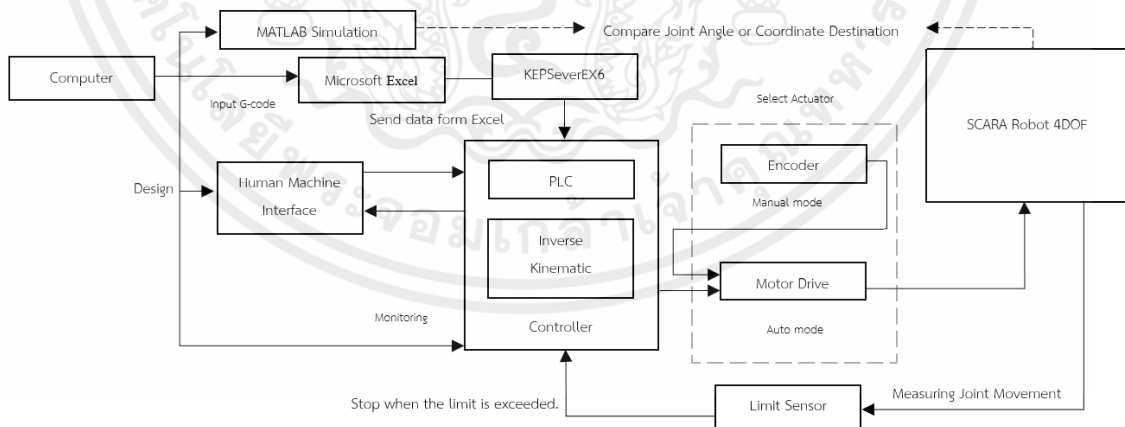
บทที่ 3

วิธีการดำเนินงาน

บทนี้จะกล่าวถึงวิธีการดำเนินงานจัดทำปฏิญานิพนธ์ซึ่งจะมีเนื้อหาที่เกี่ยวข้องกับกระบวนการในการทำงาน ขั้นตอนและวิธีการในการจัดทำตลอดจนการประยุกต์ใช้หลักการและทฤษฎีต่างๆ เพื่อให้สามารถจัดทำโครงการประยุกต์ใช้ PLC ในการควบคุมการเคลื่อนที่ของแขนกล SCARA Robot 4DOF ให้เป็นไปตามจลนศาสตร์การเคลื่อนที่ให้บรรลุวัตถุประสงค์ตามที่ได้กำหนดไว้ โดยสามารถจำแนกขั้นตอนในการดำเนินงานได้ดังต่อไปนี้

- 3.1 ภาพรวมการออกแบบการทำงาน
- 3.2 การศึกษาพื้นฐานการเคลื่อนที่ของแขนกล
- 3.3 การประยุกต์ใช้โปรแกรมแบบจำลองทางคณิตศาสตร์
- 3.4 การออกแบบโปรแกรม PLC ควบคุมการทำงานของแขนกล
- 3.5 การออกแบบตัวยึดปากกาของแขนกล SCARA Robot 4DOF
- 3.6 การเชื่อมต่อ OPC Server KEPServerEX6 เข้ากับ PLC

3.1 ภาพรวมการออกแบบการทำงาน



รูปที่ 3.1 ภาพรวมการทำงานของโครงการ

ในส่วนแรกจะเป็นการกล่าวถึงภาพรวมการทำงานทั้งหมดของโครงการการประยุกต์ใช้ PLC ควบคุม SCARA Robot 4DOF ภาพกราฟิก Graphics บนพื้น 2 มิติ เป็นการประยุกต์เอา PLC มาใช้สำหรับการควบคุมการเคลื่อนที่ของแขนกล SCARA Robot 4DOF โดยที่ภายใน PLC จะมีการเขียนโปรแกรมในการคำนวณค่าตามจลศาสตร์การเคลื่อนที่ของแขนกล ซึ่งค่าเอาต์พุตจาก PLC ที่เป็นสัญญาณพัลส์และทิศทางหมุนของมอเตอร์จะถูกส่งไปยังวงจรขับเพื่อขับมอเตอร์ให้ข้อต่อแขนกลเคลื่อนที่ไปยังตำแหน่งเป้าหมายที่กำหนด โดยที่การเคลื่อนที่ของแขนกลนั้นจะถูกกำกับขอบเขตในการเคลื่อนที่ได้ด้วยเซนเซอร์เพื่อป้องกันกรณีที่เกิดการทำงานผิดพลาด ซึ่งในส่วนของการประยุกต์ PLC เพื่อควบคุมแขนกลข้างต้นนี้ จะมีการทำงานร่วมกับส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) ซึ่งมีการออกแบบมาเพื่อทำหน้าที่ให้ผู้ใช้งานได้ทำการกำหนดค่าตำแหน่งเป้าหมายของการควบคุมแขนกล และแสดงค่าเอาต์พุตของการคำนวณและประมวลผลให้ผู้ใช้งานได้ทำการบันทึกค่า นอกจากนี้ยังมีการใช้แบบจำลองทางคณิตศาสตร์ในรูปแบบของกราฟิกคอมพิวเตอร์ผ่านการสร้าง Graphical user interface หรือ GUI บนโปรแกรม MATLAB เพื่อนำมาใช้สำหรับการจำลองการเคลื่อนที่ของแขนกลในการเคลื่อนที่ไปยังตำแหน่งต่าง ๆ อีกทั้งยังใช้สำหรับเป็นเครื่องมือในการตรวจสอบความถูกต้องของการเคลื่อนที่ของแขนกลที่มีการเคลื่อนที่ไป เมื่อแขนกลสามารถเคลื่อนที่ไปยังตำแหน่งที่ต้องการแล้วจะเพิ่มฟังก์ชันให้กับแขนกลโดยติดตั้งเครื่องมือปากกาและการเคลื่อนที่ให้เป็นลำดับขั้นตอนหรือสามารถวาดภาพ Graphics บนพื้น 2 มิติได้ นอกจากนี้ยังมีการสร้างการเชื่อมต่อกับ OPC Server ผ่าน KEPServerEX6 เพื่อใช้สำหรับการรับส่งชุดข้อมูลของพิกัดตำแหน่งเพื่อใช้สำหรับการควบคุมแขนกลให้มีการทำงานแบบอัตโนมัติในการเคลื่อนที่ไปยังตำแหน่งใดๆ ตามที่มีการกำหนดไว้

3.2 การศึกษาพื้นฐานการเคลื่อนที่ของแขนกล

เป็นการศึกษาพื้นฐานที่เกี่ยวข้องกับแขนกลหรือหุ่นยนต์เริ่มจากเรื่องเฟรมการหมุนทั้งใน 2 มิติ และ 3 มิติ ตามลำดับหัวข้อ ดังรูปที่ 3.2 การเรียนพื้นฐานดังกล่าวเป็นการต่อยอดไปยังเรื่องสมการ จลนศาสตร์การเคลื่อนที่แบบย้อนกลับ ทำให้ได้สมการของแขนกลที่ใช้ในการหาค่ามุมของแต่ละแกน

Description	Curriculum	Reviews
Coordinate Frame and Transformation (พิกัดเฟรมและการแปลง)		
1 Part 01: Coordinate Frame (เฟรมพิกัด)		12
2 Part 02: Reference Frame (เฟรมอ้างอิง)		12
3 Part 03: Rotation in 2D (การหมุนใน 2 มิติ)		22
4 Part 04: Rotation Matrix (เมทริกซ์การหมุน)		17
5 Part 05: Operation Rotation in 3D (การหมุนใน 3 มิติ)		7
6 Part 06: Sequence of Rotation (ลำดับการหมุน)		14
7 Part 07: Description Orientation (การอธิบายทิศทาง)		12
8 Part 08: Aircraft's Orientation I (ทิศทางของอากาศยาน 1)		15
9 Part 09: Aircraft's Orientation II (ทิศทางของอากาศยาน 2)		9
10 Part 10: Position in 3D (ตำแหน่งใน 3 มิติ)		15
11 Part 11: Operation Translation (การเลื่อน)		11
12 Part 12: Pose and Transformation (ท่าทางและการแปลง)		12
13 Part 13: Example of multiple frames (ยกตัวอย่างกรณีหลายเฟรม)		14
14 Part 14: Homogeneous Transformation (การแปลงแบบเอกพันธ์)		10
15 Part 15: Standard Transformation (การแปลงแบบมาตรฐาน)		14
16 Quiz unit 1 เฟรมและการแปลง		13 questions

รูปที่ 3.2 ศึกษาพื้นฐานที่เกี่ยวข้องเรื่องพิกัดเฟรมและการแปลง

Description	Curriculum	Reviews
Basic Robotics Terminology (คำศัพท์ที่ใช้ในวิทยาการหุ่นยนต์)		
1 Part 01: Kinematics (จลนศาสตร์)		11
2 Part 02: Degree of Freedom (DOF) (องศาอิสระ)		10
3 Part 03: Example of Joints (ยกตัวอย่างองศาอิสระประเภทต่าง)		16
4 Part 04: Example of 2D Robot (ยกตัวอย่างหุ่นยนต์ใน 2 มิติ)		11
5 Part 05: Serial Manipulator Terminology (คำศัพท์ที่ใช้กับแขนกลอนุกรม)		14
6 Part 06: Types of Serial Manipulators (แขนกลอนุกรมประเภทต่าง)		8
7 Part 07: Convention of n-DoF Manipulators (สัจพจน์ของแขนกลอนุกรม)		14
8 Part 08: Importance of Robotics Manipulator (ความสำคัญของหุ่นยนต์แขนกล)		16
9 Quiz unit 2 Basic Robotics Terminology		7 questions

รูปที่ 3.3 ศึกษาพื้นฐานที่เกี่ยวข้องคำศัพท์ในวิทยาการหุ่นยนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Kinematic Modelling of Serial Manipulator (การจำลองหุ่นยนต์เชิงจลนศาสตร์)		
1	Part 01: Types of Kinematics (จลนศาสตร์ประเภทต่าง)	14
2	Part 02: Setting Frame (การกำหนดเฟรม)	10
3	Part 03: Conditions of Frame Setting (เงื่อนไขการกำหนดเฟรม)	13
4	Part 04: Denavit-Hartenberg Parameter (DH พารามิเตอร์)	16
5	Part 05: Denavit-Hartenberg Table (ตาราง DH)	10
6	Part 06: Forward Pose Kinematics (การคำนวณท่าทางจลนศาสตร์)	16
7	Quiz unit 3 Kinematic Modelling of Serial Manipulator	6 questions

รูปที่ 3.4 ศึกษาพื้นฐานที่เกี่ยวข้องกับการจำลองเชิงจลนศาสตร์

Forward Kinematics of Serial Manipulators (การคำนวณจลนศาสตร์ของแขนกลอนุกรม)		
1	Part 01: Inverse Kinematics (การคำนวณจลนศาสตร์ย้อนกลับ)	6
2	Part 02: IK: Geometric Approach I (การคำนวณโดยการใช้เรขาคณิต)	11
3	Part 03: IK: Geometric Approach II (การคำนวณโดยการใช้เรขาคณิต)	7
4	Part 04: Example of 2D RR Planar I (ตัวอย่างหุ่นยนต์พ्लันนา 2 มิติ)	15
5	Part 05: Example of 2D RR Planar II (ตัวอย่างหุ่นยนต์พ्लันนา 2 มิติ)	10
6	Part 06: Example of 3 DOF RRR Manipulator I (ตัวอย่างแขนกล 3 องศาอิสระ)	12
7	Part 07: Example of 3 DOF RRR Manipulator II (ตัวอย่างแขนกล 3 องศาอิสระ)	5
8	Part 08: IK: Algebraic Approach I (การคำนวณโดยการใช้พีชคณิต)	10
9	Part 09: IK: Algebraic Approach II (การคำนวณโดยการใช้พีชคณิต)	9
10	Part 10: IK Algebraic Approach III (การคำนวณโดยการใช้พีชคณิต)	10
11	Part 11: Example: Spherical Wrist (ยกตัวอย่างข้อต่อมือแขนกล)	10
12	Part 12: Inverse Pose Kinematics: Articulated Robot I (การคำนวณท่าทางจลนศาสตร์ย้อนกลับ)	9
13	Part 13: Inverse Pose Kinematics: Articulated Robot II (การคำนวณท่าทางจลนศาสตร์ย้อนกลับ)	14
14	Quiz unit 4 Forward Kinematics of Serial Manipulators	8 questions

รูปที่ 3.5 ศึกษาพื้นฐานที่เกี่ยวข้องกับการคำนวณจลนศาสตร์ของแขนกล

Section 3
Kinematic Modelling of Serial Manipulator (การจำลองหุ่นยนต์เชิงจลนศาสตร์)

Section 4
Forward Kinematics of Serial Manipulators (การคำนวณจลนศาสตร์ของแขนกลอนุกรม)

- 1 Part 01: Inverse Kinematics (... 6
- 2 Part 02: IK: Geometric Appro... 11
- 3 Part 03: IK: Geometric Appr... 7
- 4 Part 04: Example of 2D RR P... 15
- 5 Part 05: Example of 2D RR P... 10
- 6 Part 06: Example of 3 DOF R... 12

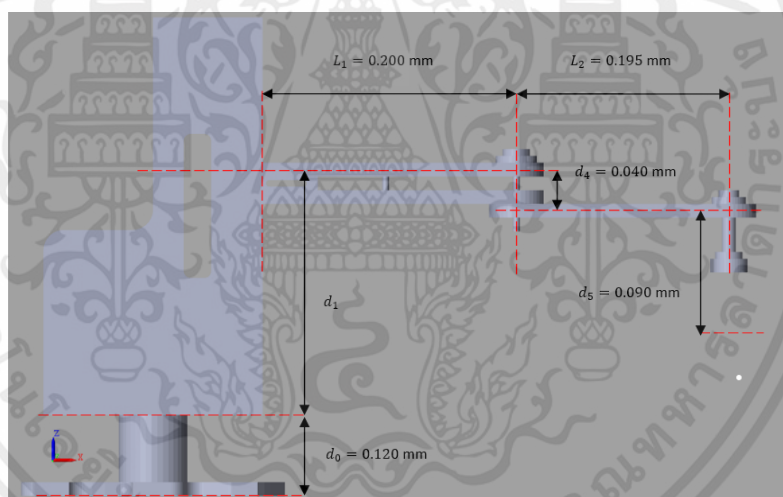
รูปที่ 3.6 ตัวอย่างการศึกษาพื้นฐานที่เกี่ยวข้องกับการเคลื่อนที่ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การประยุกต์ใช้โปรแกรมแบบจำลองทางคณิตศาสตร์

มีการใช้แบบจำลองแขนกล SCARA Robot 4DOF ดังรูปภาพ 3.12 ร่วมกับโปรแกรมแบบจำลองทางคณิตศาสตร์ เพื่อตรวจสอบค่ามุมของแต่ละแกน เมื่อป้อนค่าตำแหน่งพิกัด(x, y, z)ปลายแขนกล

ใช้โปรแกรม MATLAB R2022b MATLAB GUI ในการประยุกต์ใช้แบบจำลองการเคลื่อนที่ของแขนกล จากสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ ทำให้ได้สมการจลนศาสตร์การเคลื่อนที่ของแขนกล SCARA Robot 4DOF จลนศาสตร์การเคลื่อนที่แบบย้อนกลับดังสมการที่ (2.31), (2.36) และ (2.37) เมื่อทำการแทนค่าของความยาวก้านต่อลงใน สมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ โดยที่ d_0 มีค่าเป็น 0.120 m, d_4 มีค่าเป็น -0.040 m, d_5 มีค่าเป็น -0.090 m, L_1 มีค่าเป็น 0.200 m และ L_2 มีค่าเป็น 0.195 m จะทำให้ได้สมการใหม่เป็นสมการจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับจะได้



รูปที่ 3.7 ระยะของแขนกลSCARA Robot 4DOF

- x คือ พิกัดแนวแกน X
- y คือ พิกัดแนวแกน y
- L_1 คือ ระยะก้าน 1 ของแขนกล
- L_2 คือ ระยะก้าน 2 ของแขนกล
- d_0 คือ ระยะฐานของแขนกล
- d_4 คือ ระยะห่างระหว่าง L_1 และ L_2
- d_5 คือ ระยะห่างระหว่าง L_2 ถึงปลายปากกา

$$\frac{x^2+y^2-(L_1)^2-(L_2)^2}{2L_1L_2} = \cos(\theta_2) \quad (3.1)$$

กำหนดเป็นตัวแปรดังนี้

$$C_2 = \cos(\theta_2) \quad (3.2)$$

$$S_2 = \pm \sqrt{1-C_2^2} \quad (3.3)$$

$$\theta_2 = \text{atan2}(S_2, C_2) \quad (3.4)$$

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(L_2 S_2, L_1 + L_2 C_2) \quad (3.5)$$

แทนค่า

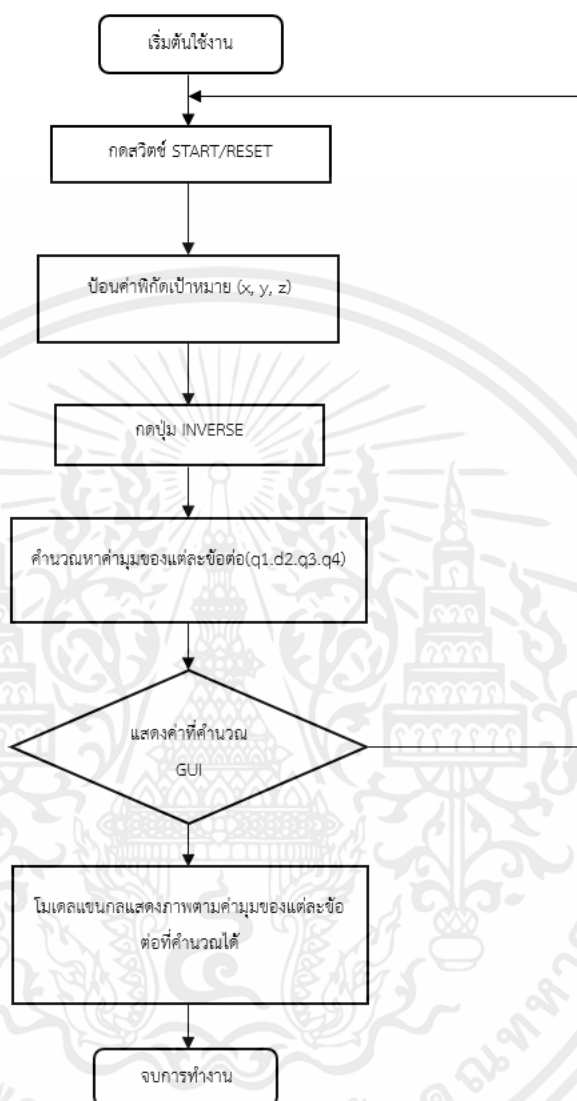
$$\theta_2 = \text{atan2}(\sqrt{1-\cos^2} \frac{x^2+y^2-(0.200)^2-(0.195)^2}{2(0.200)(0.195)}) \quad (3.6)$$

$$\theta_1 = \text{atan2}(y, x) - \text{atan2}(0.195 S_2, 0.200 + 0.195 C_2) \quad (3.7)$$

เนื่องจากแขนกลมีระยะการเคลื่อนที่ที่จำกัด ค่า (x, y) ที่สามารถนำมาคำนวณได้ต้องมีค่าน้อยกว่าระยะก้านแขนกลตั้งสมการ

$$\sqrt{x^2+y^2} \leq L_1 + L_2 \quad (3.8)$$

จากนั้นได้ทำการสร้าง GUI โดยนำเอาสมการที่ (3.1) จนถึง (3.7) มาประยุกต์ใช้ โดยกำหนดให้มีเงื่อนไขในการทำงาน ดังนี้



รูปที่ 3.8 แสดง Flow chart เงื่อนไขการทำงานของแบบจำลองทางคณิตศาสตร์

จากรูปภาพด้านล่างเป็นโปรแกรมการออกแบบหน้าจอและเงื่อนไขการทำงานตามปุ่มคำสั่งของแบบจำลองการเคลื่อนที่ทางคณิตศาสตร์โดยมีการใช้สมการจลนศาสตร์การเคลื่อนที่แบบย้อนกลับมาคำนวณ เมื่อทำการเริ่มโปรแกรมจะแสดงหน้าจอ ดังรูปที่ 3.11

```

72 % Get default command line output from handles structure
73 varargin{1} = handles.output;
74
75
76 % --- Executes on button press in inverse.
77 function inverse_callback(hObject, eventdata, handles)
78 ModelName = 'updateSCARA';
79 global var;
80 x=get(handles.x,'string');
81 x = str2num(x);
82 if ((x <= -350)&&(x >=350))
83     msgbox('Please Select PX Value In Range -350-->350');
84 end
85 y=get(handles.y,'string');
86 y = str2num(y);
87 if ((y <= -350)&&(y >=350))
88     msgbox('Please Select PY Value In Range -350-->350');
89 end
90 z=get(handles.z,'string');
91 z = str2num(z);
92 if ((z <= 20)&&(z >= 180))
93     msgbox('Please Select PZ Value In Range 20-->180');
94 end
95 xi = x/1000;
96 yi = y/1000;
97 zi = z/1000;
98 d0 = 0.12; d4 = -0.039325; d5 = -0.06;
99 L2 = 0.20; L3 = 0.15;
100 C2 = ((xi)^2 + (yi)^2 - (L2)^2 - (L3)^2)/(2*L2*L3);
101 C2f = round(C2/1000)/1000;
102 S2 = 1*sqrt(1-(C2f)^2);
103 if (S2 <= -1)
104     msgbox('Robot Crash Close Program to Restart');at
105 end
106 q3 = atan2(S2,C2f)*(-1);
107 q3In = q3*(-1);
108 q1 = atan2d(yi,xi)-atan2d(L3*S2,L2+(L3*C2f));
109 q2 = (pi + d4 - d5 - d0)*1000 ;
110 q4 = 0;
111 guidata(hObject, handles);
112 set(handles.q1,'string',q1);
113 set_param(hObject,'Slider Gain','Gain',num2str(q1));
114 set(handles.q2,'string',q2);
115 set_param(hObject,'Slider Gain1','Gain',num2str(q2));
116 set(handles.q3,'string',q3In);
117 set_param(hObject,'Slider Gain2','Gain',num2str(q3In));
118 set(handles.q4,'string',q4);
119 set_param(hObject,'Slider Gain3','Gain',num2str(q4));
120 % hObject handle to inverse (see GCBO)
121 % eventdata reserved - to be defined in a future version of MATLAB
122 % handles structure with handles and user data (see GUIDATA)
123

```

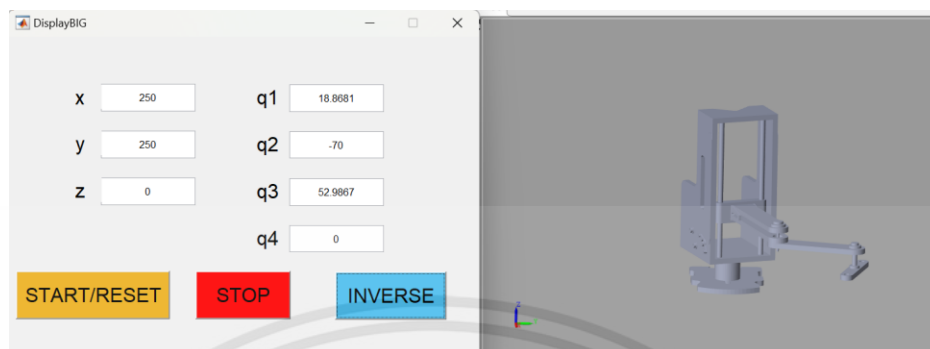
รูปที่ 3.9 การเขียนโปรแกรมสำหรับคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ

เมื่อเขียนโปรแกรมจากดังรูปภาพที่ 3.9 จะได้หน้าจอแสดงผลตามรูปภาพด้านล่างเพื่อป้อนค่าพิกัดตำแหน่งและสังเกตค่ามุมของโมเดลแขนกลจำลอง

The image shows a MATLAB GUI window titled 'DisplayBIG'. It contains a grid of input fields for numerical data. The first column has fields for 'x', 'y', and 'z'. The second column has fields for 'q1', 'q2', 'q3', and 'q4'. Below the grid are three buttons: 'START/RESET' (yellow), 'STOP' (red), and 'INVERSE' (blue). The window has standard Windows-style title bars with minimize, maximize, and close buttons.

รูปที่ 3.10 หน้าต่างโปรแกรมการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงหน้าต่าง GUI และโมเดลแขนกลจำลอง

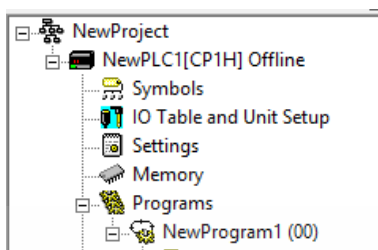
จากรูปที่ 3.10 จะแสดงหน้าต่างของ GUI แบบจำลองทางคณิตศาสตร์เป็นการคำนวณโดยผู้ใช้อ้อนค่าพิกัด (x, y, z) ตำแหน่งเป้าหมายของแขนกลคำนวณออกเป็นค่ามุมของแต่ละแกนของแขนกล (q1, q2, q3, q4) ส่วนการทำงานปุ่มคำสั่ง START/RESET จะเป็นการ RESET ค่าพิกัด (x, y, z) ให้มีค่า = 0 ปุ่ม INVERSE จะเป็นการคำนวณตามสมการจลนศาสตร์การเคลื่อนที่แบบย้อนกลับและปุ่ม STOP จะเป็นการออกจากโปรแกรมภาพของโมเดลแขนกลจำลองจะเป็นการใช้ค่ามุมของแต่ละแกนของแขนกลที่ได้จากการคำนวณในการแสดงผลภาพ

3.4 การออกแบบโปรแกรม PLC ควบคุมการทำงานของแขนกล

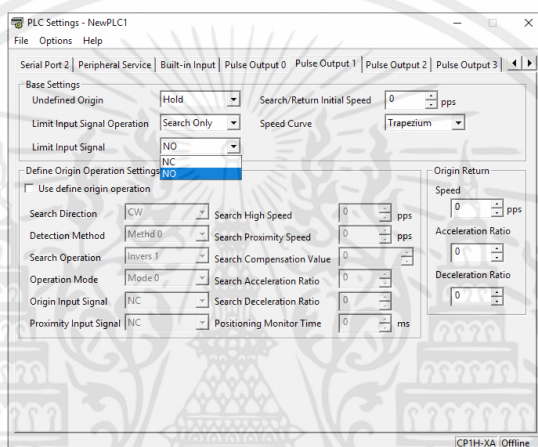
การออกแบบโปรแกรมของการควบคุมแขนกล SCARA Robot 4DOF มีการกำหนดเงื่อนไขในการทำงานและเขียนโปรแกรมควบคุมให้กับ PLC รวมไปถึงการออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน (HMI) ซึ่งสามารถอธิบายวิธีการดำเนินการได้ดังนี้

3.4.1 การเขียนโปรแกรม PLC Ladder ควบคุมการทำงานของมอเตอร์แบบสเต็ป

ใช้โปรแกรม CX-Programmer ในการเขียนโปรแกรมเพื่อควบคุมการทำงานของมอเตอร์ในแต่ละแกนที่เป็นอิสระต่อกัน เริ่มจากการตั้งค่า Pulse Output 0, Pulse Output 1, Pulse Output 2, Pulse Output 3 ให้ตรงกับลักษณะการทำงานของมอเตอร์ไปที่ Settings ตามรูปที่ 3.12 สังเกต Base Settings ที่ limit input signal ตั้งค่าให้เป็น NO ตามรูปที่ 3.13 เนื่องจากที่แขนกลมีลิมิตสวิตช์ป้องกันการเคลื่อนที่จากพื้นที่การทำงานจะติดตั้งลิมิตสวิตช์ชนิด NO ป้องกันไว้

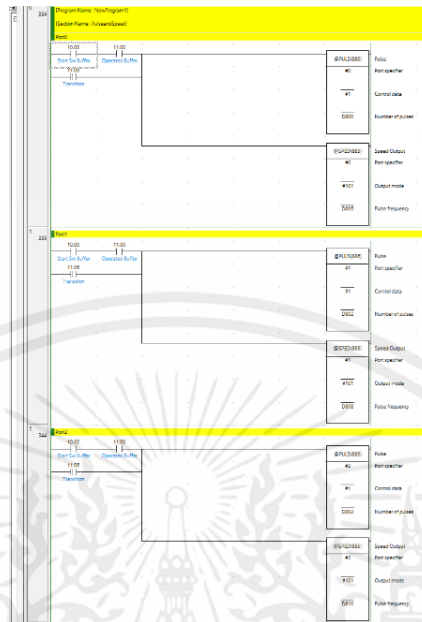


รูปที่ 3.12 CX-Programmer Settings



รูปที่ 3.13 การตั้งค่า Pulse Output

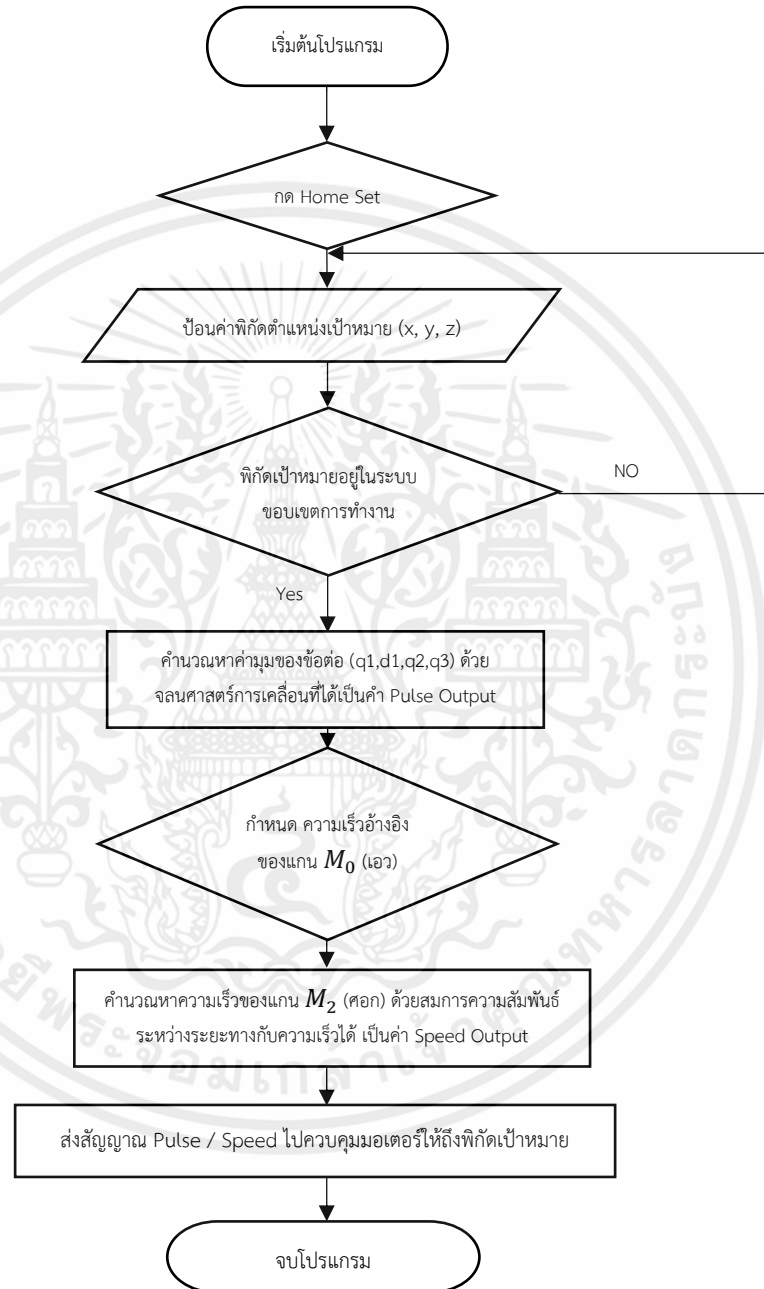
ฟังก์ชันที่ใช้ในการขับเคลื่อนมอเตอร์จะมีอยู่ 2 ฟังก์ชันคือ PLUS(886) ใช้สำหรับกำหนดระยะทางให้การเคลื่อนที่ของมอเตอร์และ SPED(885) ใช้กำหนดทิศทางและความเร็วในการเคลื่อนที่ของมอเตอร์



รูปที่ 3.14 โปรแกรมการควบคุมมอเตอร์แบบสแต็ปของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

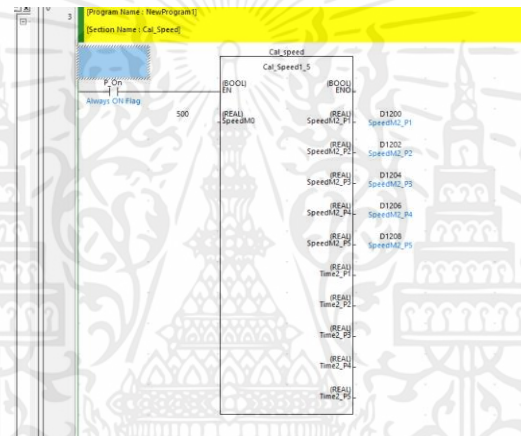
3.4.2 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ



รูปที่ 3.15 แสดง Flowchart เงื่อนไขการทำงานของโปรแกรม PLC ควบคุมการทำงานของแขนกล

3.4.3 การเขียนโปรแกรม PLC คำนวณความเร็วการเคลื่อนที่ของมอเตอร์

ในการเคลื่อนที่ของแขนกลไปยังตำแหน่งเป้าหมายใดๆ การเคลื่อนที่ของข้อต่อแต่ละแกนนั้นจะต้องมีความสัมพันธ์กัน กล่าวคือข้อต่อแต่ละแกนนั้นจะต้องเคลื่อนที่ไปยังตำแหน่งเป้าหมาย โดยใช้ระยะเวลาที่เท่าๆ กัน ซึ่งได้มีการเขียนโปรแกรมด้วยภาษา Structure Text ในการคำนวณหาความเร็วของแกนข้อต่อ โดยใช้หลักการความสัมพันธ์ระหว่างระยะทางและอัตราเร็ว ซึ่งใช้ค่าที่ได้จากการคำนวณระยะทาง จาก Function Block ของการคำนวณจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ มาคำนวณหาความเร็วการเคลื่อนที่ของแกนที่ 3 (elbow) ดังนี้



รูปที่ 3.18 Function Block ของการคำนวณค่าความเร็วของข้อต่อแต่ละแกนในการเคลื่อนที่

```

Internals  Inputs  Outputs  In Out  Externals

A := ABS(BasePulse_1);
B := ABS(BasePulse_2);
C := ABS(BasePulse_3);
D := ABS(BasePulse_4);
E := ABS(BasePulse_5);

K := ABS(ShoulderPulse_1);
L := ABS(ShoulderPulse_2);
M := ABS(ShoulderPulse_3);
N := ABS(ShoulderPulse_4);
O := ABS(ShoulderPulse_5);

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(A)/(Time2_P1);
Time2_P1 := ABS(A)/(SpeedM0);
SpeedM2_P1 := ABS(K)/(Time2_P1);
END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(B)/(Time2_P2);
Time2_P2 := ABS(B-A)/(SpeedM0);
SpeedM2_P2 := ABS(L)/(Time2_P2);
END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(C-B)/(Time2_P3);
Time2_P3 := ABS(C-B)/(SpeedM0);
SpeedM2_P3 := ABS(M-L)/(Time2_P3);
END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(D-C)/(Time2_P4);
Time2_P4 := ABS(D-C)/(SpeedM0);
SpeedM2_P4 := ABS(N-M)/(Time2_P4);
END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(E-D)/(Time2_P5);
Time2_P5 := ABS(E-D)/(SpeedM0);
SpeedM2_P5 := ABS(O-N)/(Time2_P5);
END_IF;

```

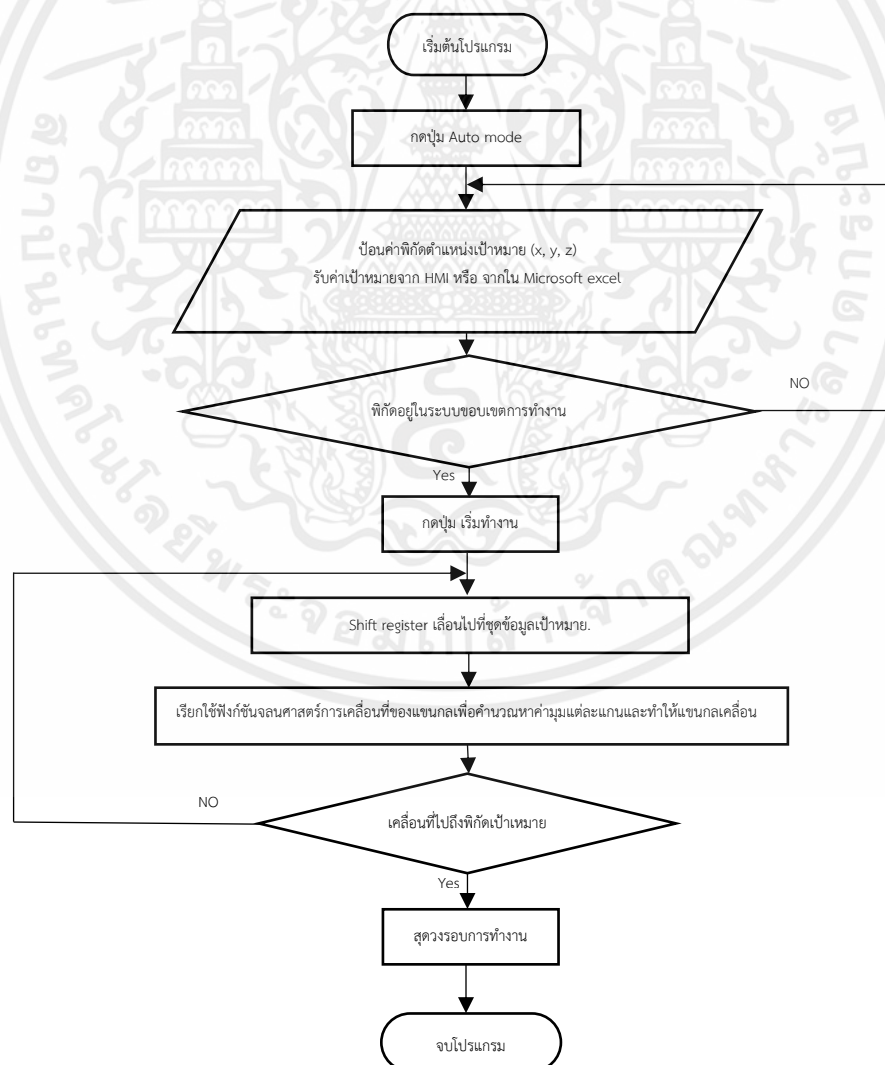
รูปที่ 3.19 Structure Text ในการคำนวณค่าความเร็วของข้อต่อแต่ละแกนในการเคลื่อนที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.4 การเขียนโปรแกรม PLC ควบคุมให้มีการเคลื่อนที่เป็นแบบอัตโนมัติเพื่อใช้ในการวาด

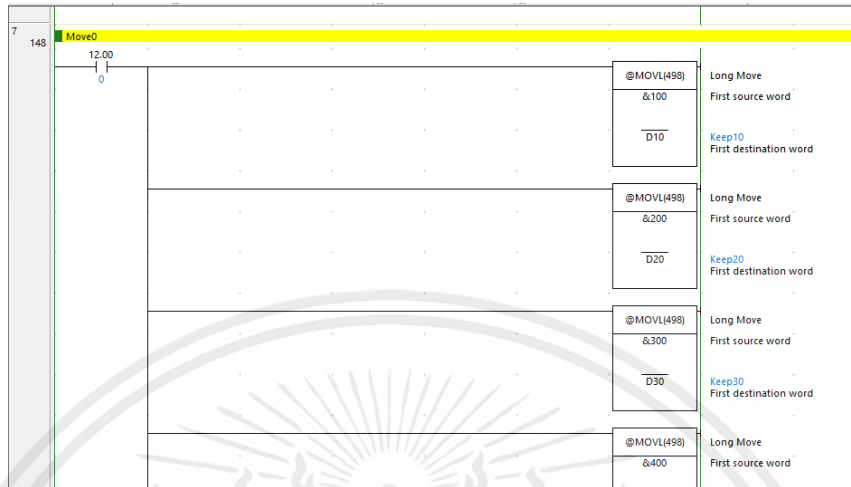
ภาพ 2 มิติ

การประยุกต์ใช้แขนกลเพื่อใช้ในการวาดภาพ 2 มิตินั้นจะมีการทำงานที่มีการเคลื่อนที่ไปในหลาย ๆ จุด ดังนั้นจึงได้มีการออกแบบโปรแกรมเพื่อควบคุมแขนกลโดยให้ผู้ใช้สามารถป้อนค่าอินพุตได้หลายค่า และให้แขนกลมีการเคลื่อนที่เป็นวงรอบ ซึ่งจะใช้การเขียนด้วยภาษา Ladder มีการเรียกใช้ข้อมูลแบบ Indirect address การชี้ข้อมูลแบบ Indirect address มีวัตถุประสงค์เพื่อใช้จัดการเก็บข้อมูลตำแหน่งและความเร็วลงในหน่วยความจำตามลำดับก่อน,หลังให้ง่ายและสะดวก เมื่อผู้พัฒนาต้องการจัดเก็บหรือนำข้อมูลออกมาใช้งานได้อย่างเป็นระบบ นอกจากนี้ยังมีการใช้ Shift register ร่วมกับ Complete Flag เพื่อลำดับชุดข้อมูลออกมาทีละตำแหน่ง เขียนลักษณะเงื่อนไขของโปรแกรมได้ดังนี้



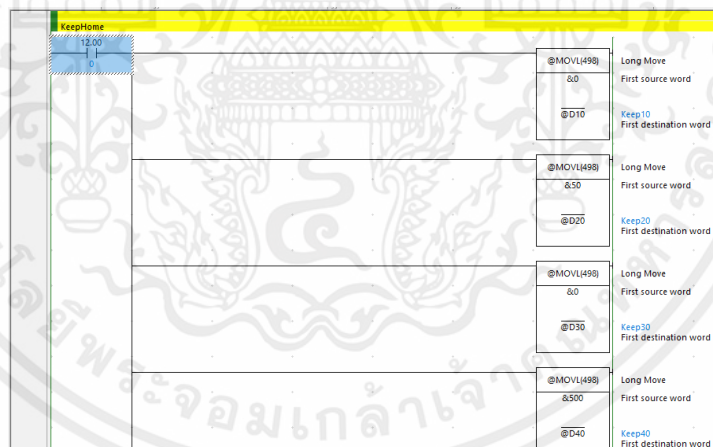
รูปที่ 3.20 เงื่อนไขการทำงานของโปรแกรมควบคุมแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

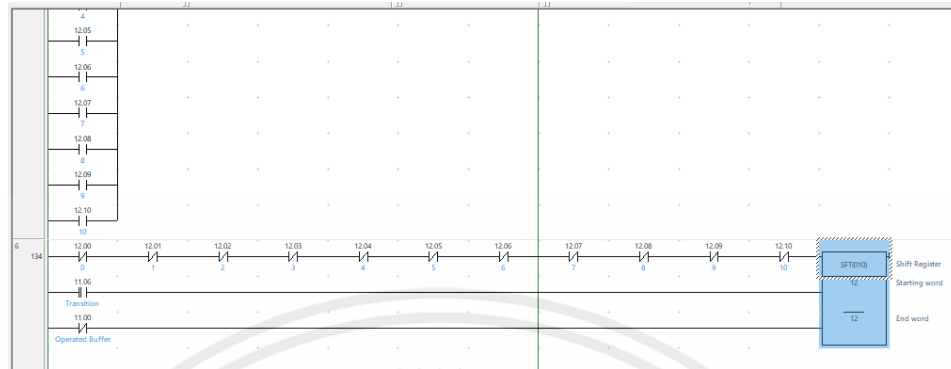


รูปภาพ 3.21 โปรแกรมสร้างช่องทางเรียกใช้ข้อมูลแบบ Indirect Address

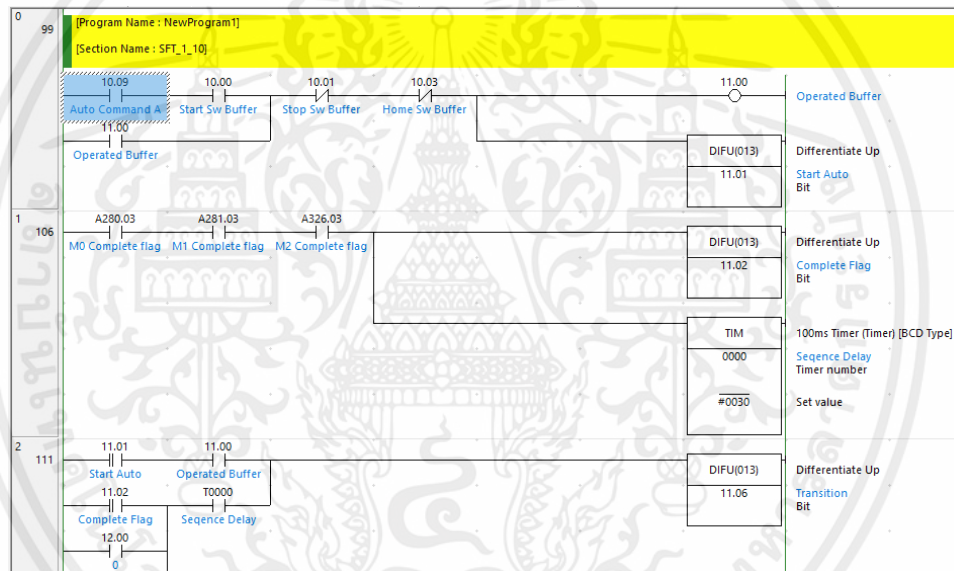
เมื่อได้ค่า Pulse และ Speed จากการคำนวณ จะนำมาเก็บไว้หมายเลขเดียวกับช่องทาง
การสร้าง Indirect Address เพื่อรอการส่งออกไปยังฟังก์ชัน PULS(886) และ SPED(885)



รูปที่ 3.22 โปรแกรมเก็บค่าที่ได้จากการคำนวณไว้ในช่องทางเดียวกับการสร้าง Indirect Address

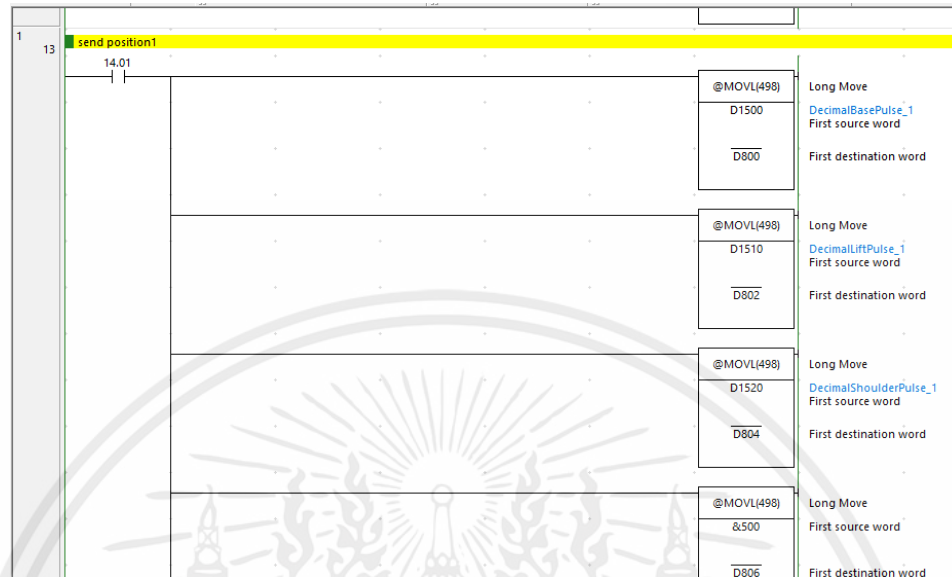


รูปที่ 3.23 โปรแกรมควบคุมการเคลื่อนที่แบบอัตโนมัติโดยใช้ Shift register



รูปที่ 3.24 โปรแกรมควบคุมการเคลื่อนที่โดยใช้ Complete Flag ของ Pulse Out แต่ละแกน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

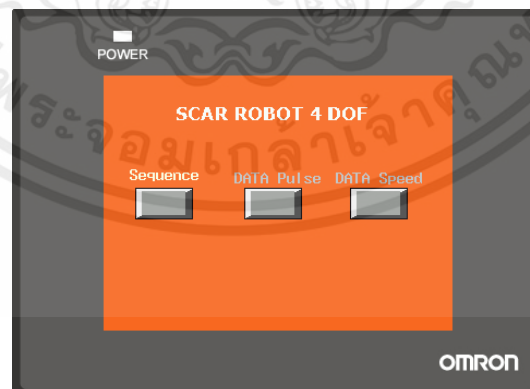


รูปที่ 3.25 โปรแกรมการส่งค่าไปยังฟังก์ชัน PULS(886) และ SPED(885)

3.4.5 การออกแบบส่วนเชื่อมต่อกับผู้ใช้งาน

ในการออกแบบส่วนเชื่อมต่อกับผู้ใช้งานได้มีการออกแบบเพื่อให้ผู้ใช้งานทำการป้อนค่าอินพุตซึ่งเป็นค่าตำแหน่งเป้าหมายของการควบคุมแขนกล ซึ่งแบ่งออกได้เป็น 4 หน้าต่างดังนี้

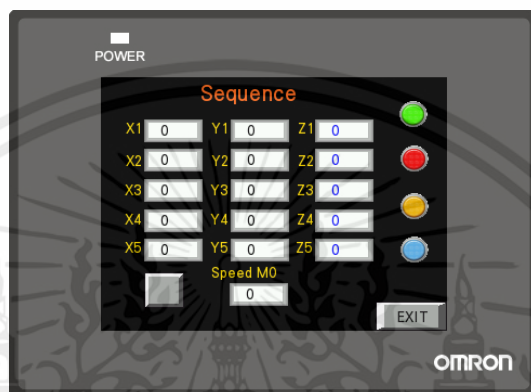
1.) Main Menu เป็นหน้าต่างสำหรับเลือกการควบคุมการเคลื่อนที่ในการวาดภาพ 2 มิติ หรือการแสดงผลข้อมูลรูปแบบ Pulse และ Speed



รูปที่ 3.26 Main Menu (HMI)

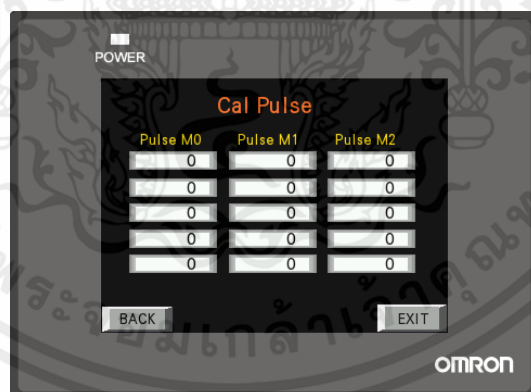
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) Sequence เป็นหน้าต่างสำหรับการควบคุมตามจลนศาสตร์การเคลื่อนที่ของแกนกลแบบย้อนกลับ ซึ่งจะประกอบไปด้วยส่วนของการป้อนค่าพิกัดตำแหน่งของส่วนปลายแขนกล ส่วนป้อนค่า Speed อ่างอิง และปุ่มควบคุมการทำงาน



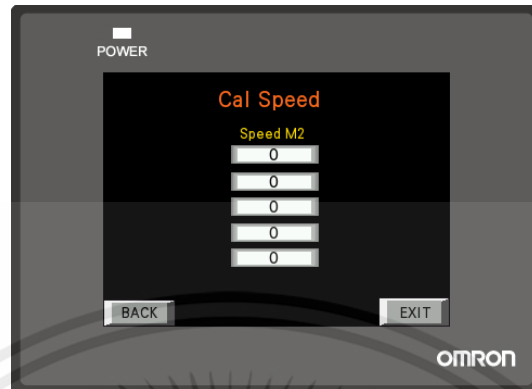
รูปที่ 3.27 หน้าต่าง Sequence Control Mode(HMI)

3.) Data Pulse เป็นหน้าต่างสำหรับการแสดงผลค่า Pulse ของ แต่ละแกน



รูปที่ 3.28 หน้าต่าง Data Pulse(HMI)

4.) Data Speed เป็นหน้าต่างสำหรับการแสดงผลค่า Speed ของ แต่ละแกน

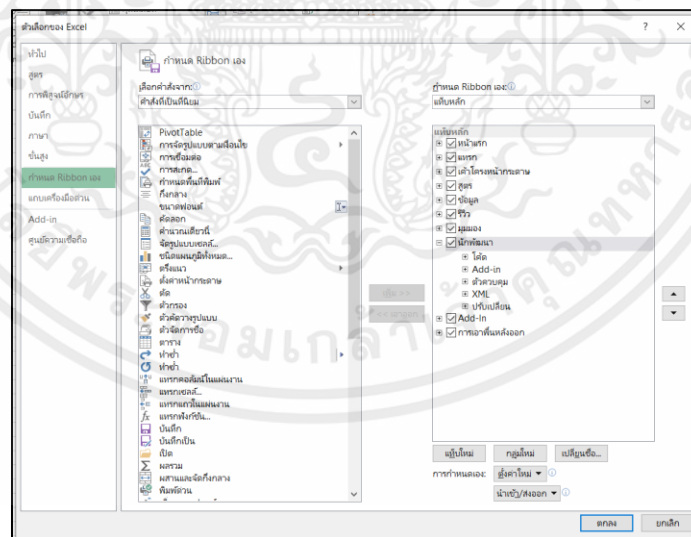


รูปที่ 3.29 หน้าต่าง DATA Speed(HMI)

3.5 การเชื่อมต่อ KEPServerEX6 เข้ากับ PLC

3.5.1 การเชื่อมต่อ Microsoft Excel กับ KEPServerEX6

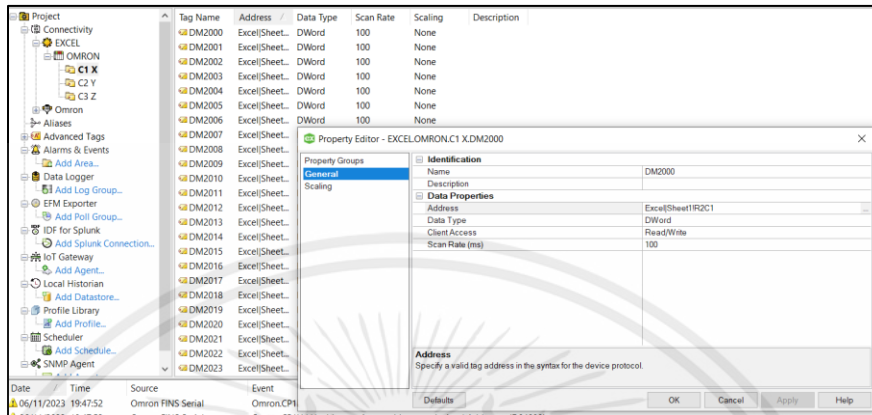
ทำการตั้งค่า Microsoft Excel ให้เป็นโหมดผู้พัฒนาเพื่อทำให้ตัวโปรแกรมสามารถนำเข้าและส่งออกข้อมูลแบบเรียลไทม์ได้



รูปที่ 3.30 การตั้งค่า Microsoft Excel

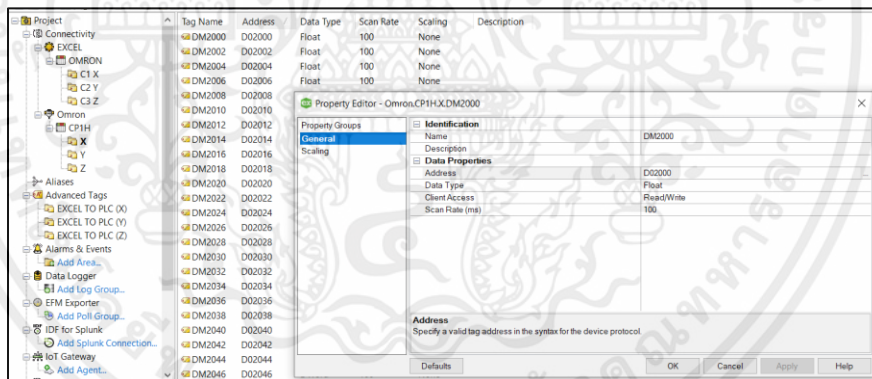
สร้าง channel เพื่อเป็นเหมือนช่องทางการส่งข้อมูลจาก Microsoft Excel ไปสู่ PLC และสร้าง address ของแต่ละคอลัมน์เพื่อที่จะดึงข้อมูลชุดพิกัดตัวเลขมาใช้เช่น R2C1 คือ Row ที่ 2 คอลัมน์ที่ 1 ของโปรแกรม Microsoft Excel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



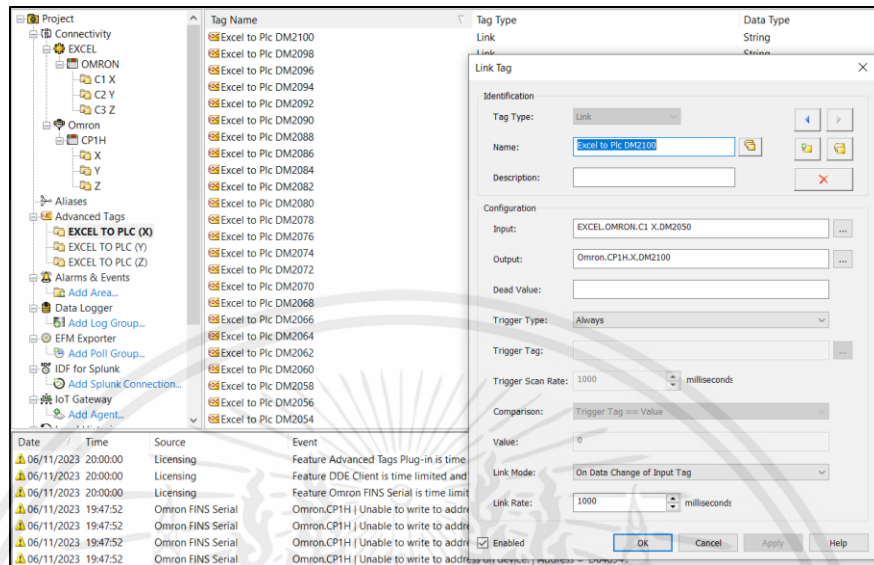
รูปที่ 3.31 ช่องทางการส่งข้อมูลจาก Microsoft Excel ไปสู่ PLC

สร้าง Tag data เพื่อที่จะเตรียมส่งข้อมูลนั้นๆไปยัง Data memory ที่ PLC ตัวอย่างในภาพที่ 3.1 คือการตั้งค่า address ให้ส่ง R2C1 ไปยัง Data memory ที่ D2000



รูปที่ 3.32 สร้าง Tag data

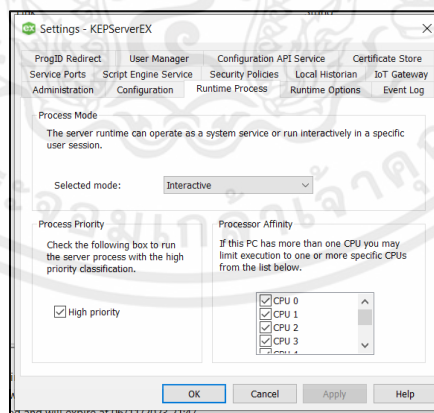
สร้าง Advance tag เพื่อกำหนด Input Output ในการส่งรวมถึงการเลือกรุ่น PLC ปลายทางที่จะส่งไป



รูปที่ 3.33 สร้าง Advance tag

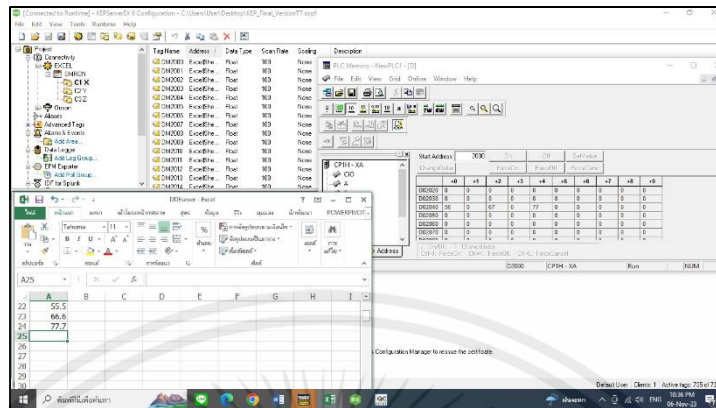
3.5.2 การเชื่อมต่อ KEPServerEX6 กับ CX-Programmer

เมื่อทำการตั้งค่า Microsoft Excel และ KEPServerEX6 เรียบร้อยแล้วขั้นตอนต่อไปจะทำการเชื่อมต่อคอมพิวเตอร์และ PLC เข้าด้วยกันผ่านสาย RS232 และทำการเปิด Sever KEP SeverEX6 ให้เป็นโหมด Interactive เพื่อให้สามารถทำการส่งข้อมูลไปยัง PLC ได้



รูปที่ 3.34 การเชื่อมต่อระหว่างคอมพิวเตอร์กับ PLC

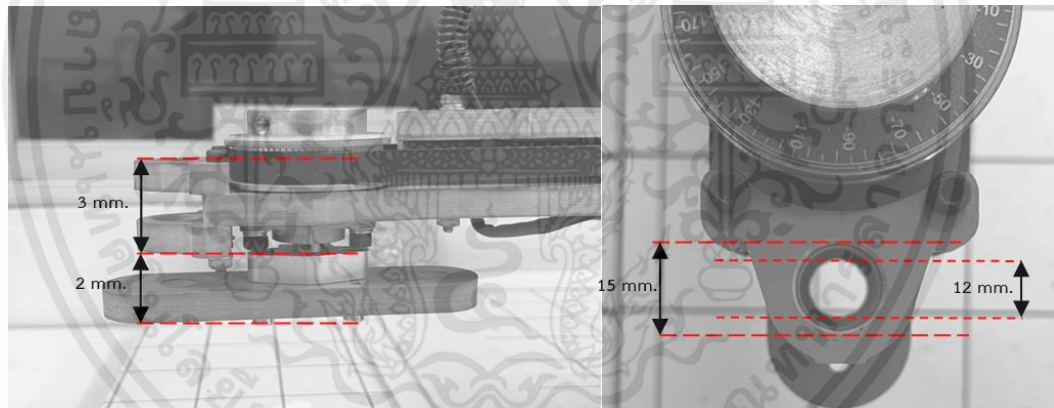
ทดสอบระบบการส่งข้อมูลไปยังปลายทางโดยต้นทางจะเป็น Microsoft Excel และปลายทางจะเป็น Data memory PLC ดังรูปที่...แสดงการทำงานว่าระบบสามารถทำงานได้



รูปที่ 3.35 การส่งข้อมูลระหว่างต้นทาง Microsoft Excel และปลายทาง Data memory PLC

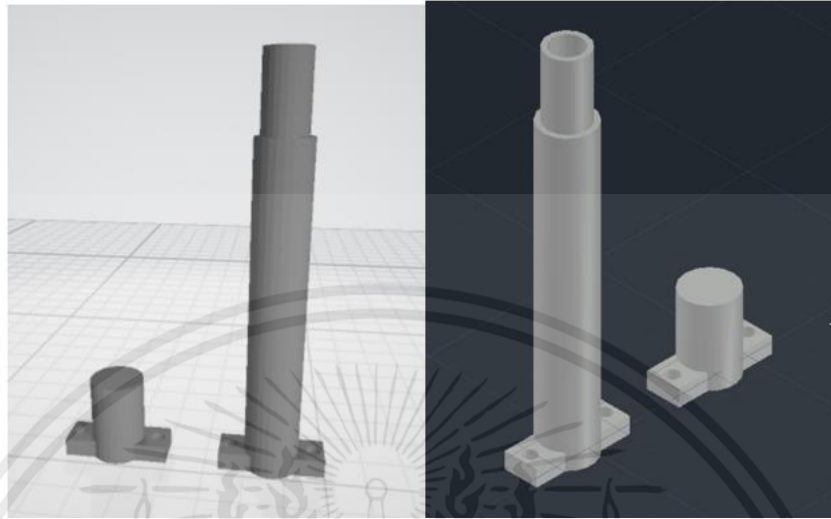
3.6 การออกแบบตัวยึดปากกาของแขนกล SCARA Robot 4DOF

วัดระยะก้านต่อของแขนกล SCARA Robot 4DOF แสดงระยะดังรูปที่ 3.36



รูปที่ 3.36 ระยะช่องว่างของส่วนปลายเพื่อออกแบบเครื่องมือปากกา

ทำการออกแบบจากรยะที่วัดได้ด้วยโปรแกรม Auto CAD โดยมีส่วนของตัวปากกาและส่วนฝาของปากกามีรูยึดระหว่างทั้งสองส่วน



รูปที่ 3.37 ออกแบบเครื่องมือโดยใช้โปรแกรม Auto CAD

ทำการบันทึกไฟล์เพื่อนำไปพิมพ์ออกมาเป็นชิ้นงานดังรูปที่ 3.38 เมื่อได้ชิ้นงานทำการตัดแต่งและนำปากกาขนาดขนาดความกว้าง 10 มิลลิเมตร ติดตั้งเข้ากับปอกปากกา ภายในฝาจจะมีสปริงเพื่อลดแรงกดจากแกน Z ของแขนกล

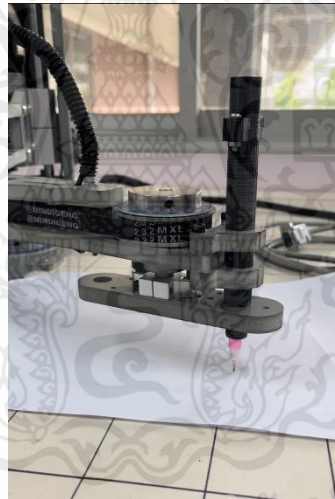


รูปที่ 3.38 พิมพ์เครื่องมือของแขนกล SCARA Robot 4DOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.39 ติดตั้งปอกปากกากับแขนกลSACARA Robot 4DOF



รูปที่ 3.40 ติดตั้งปากกาและปอกปากกากับแขนกลSACARA Robot 4DOF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

ในบทนี้จะเป็นการทดสอบและหาประสิทธิภาพของการประยุกต์ PLC ควบคุมแขนกล SCARA Robot 4DOF ในการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับให้เป็นไปตามวัตถุประสงค์ที่ได้มีการกำหนดไว้ ซึ่งสามารถแบ่งผลของการทดลองออกเป็น 2 หัวข้อดังนี้

4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์

4.2 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพ 2 มิติที่หลากหลายรูปแบบ

4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์

การทดลองนี้เป็นการควบคุมการเคลื่อนที่ของแขนกลโดยมีการกำหนดอินพุตเป็นพิกัดตำแหน่งเป้าหมาย คือค่า (x, y, z) ที่ตำแหน่งปลายแขนกล เมื่อแขนกลเคลื่อนที่ไปยังพิกัดตำแหน่งเรียบร้อยแล้วทำการเก็บบันทึกค่ามุมของแต่ละข้อต่อ นำมาเปรียบเทียบกับการจำลองการเคลื่อนที่ของแขนกลบนแบบจำลองทางคณิตศาสตร์ เพื่อตรวจสอบความถูกต้อง ทำการทดลองซ้ำจำนวน 5 ครั้ง มีการกำหนดพิกัดตำแหน่งในการเคลื่อนที่ ทั้งหมด 4 จตุภาค (Quadrant) ซึ่งประกอบไปด้วยพิกัด $(+x +y)$, $(+x -y)$, $(-x +y)$ และ $(-x -y)$ ตามลำดับ ผลลัพธ์การทดลองได้ดังนี้

ตารางที่ 4.1 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด (+x +y)

ครั้งที่	กำหนดอินพุตเป็นพิกัดตำแหน่งเป้าหมาย(มม.)			ค่ามุมของข้อต่อจากแบบจำลอง (องศา)				ค่ามุมของข้อต่อจากแขนกลจริง(องศา)			
	x	y	z	θ_1	θ_2	θ_3	θ_4	θ_1	θ_2	θ_3	θ_4
1	250	300	0	41.6775	-70	17.2539	0	45	-70	15	0
2	250	300	0	41.6775	-70	17.2539	0	45	-70	15	0
3	250	300	0	41.6775	-70	17.2539	0	45	-70	15	0
4	250	300	0	41.6775	-70	17.2539	0	45	-70	15	0
5	250	300	0	41.6775	-70	17.2539	0	45	-70	15	0

ตารางที่ 4.2 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด (+x -y)

ครั้งที่	กำหนดอินพุตเป็นพิกัดตำแหน่งเป้าหมาย(มม.)			ค่ามุมของข้อต่อจากแบบจำลอง (องศา)				ค่ามุมของข้อต่อจากแขนกลจริง(องศา)			
	x	y	z	θ_1	θ_2	θ_3	θ_4	θ_1	θ_2	θ_3	θ_4
1	250	-300	0	-58.711	-70	-17.25	0	-60	-70	15	0
2	250	-300	0	-58.711	-70	-17.25	0	-60	-70	15	0
3	250	-300	0	-58.711	-70	-17.25	0	-60	-70	15	0
4	250	-300	0	-58.711	-70	-17.25	0	-60	-70	15	0
5	250	-300	0	-58.711	-70	-17.25	0	-60	-70	15	0

ตารางที่ 4.3 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด $(-x + y)$

ครั้งที่	กำหนดอินพุตเป็นพิกัดตำแหน่งเป้าหมาย(มม.)			ค่ามุมของข้อต่อจากแบบจำลอง (องศา)				ค่ามุมของข้อต่อจากแขนกลจริง(องศา)			
	x	y	z	θ_1	θ_2	θ_3	θ_4	θ_1	θ_2	θ_3	θ_4
1	-250	300	0	121.289	-70	17.2539	0	125	-70	15	0
2	-250	300	0	121.289	-70	17.2539	0	125	-70	15	0
3	-250	300	0	121.289	-70	17.2539	0	125	-70	15	0
4	-250	300	0	121.289	-70	17.2539	0	125	-70	15	0
5	-250	300	0	121.289	-70	17.2539	0	125	-70	15	0

ตารางที่ 4.4 การทดลองควบคุมแขนกลให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ เปรียบเทียบกับการเคลื่อนที่ของแขนกลในแบบจำลองทางคณิตศาสตร์ ที่พิกัด $(-x - y)$

ครั้งที่	กำหนดอินพุตเป็นพิกัดตำแหน่งเป้าหมาย(มม.)			ค่ามุมของข้อต่อจากแบบจำลอง (องศา)				ค่ามุมของข้อต่อจากแขนกลจริง(องศา)			
	x	y	z	θ_1	θ_2	θ_3	θ_4	θ_1	θ_2	θ_3	θ_4
1	-200	-300	0	-147.48	-70	48.2409	0	-150	-70	45	0
2	-200	-300	0	-147.48	-70	48.2409	0	-150	-70	45	0
3	-200	-300	0	-147.48	-70	48.2409	0	-150	-70	45	0
4	-200	-300	0	-147.48	-70	48.2409	0	-150	-70	45	0
5	-200	-300	0	-147.48	-70	48.2409	0	-150	-70	45	0

4.2 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพ 2 มิติที่หลากหลาย

รูปแบบ

การทดลองนี้เป็นการควบคุมการเคลื่อนที่ของแขนกลให้มีการวาดภาพกราฟิก 2 มิติแบบอัตโนมัติโดยมีการกำหนดค่าอินพุตเป็นพิกัดตำแหน่งเป้าหมายหลายๆตำแหน่งให้แขนกลเคลื่อนที่ตามลำดับ โดยจะมีการกำหนดพิกัดเป็นรูปเลขาคณิต 5 รูปแบบ คือ สามเหลี่ยม สี่เหลี่ยมจัตุรัส สี่เหลี่ยมผืนผ้า สี่เหลี่ยมคางหมู และรูปนาฬิกาทราย และจะทำการบันทึกผลตามค่าพิกัดที่กำหนดให้แขนกลทำการวาดที่ละจุด

ตารางที่ 4.5 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพสามเหลี่ยม 2 มิติ

ครั้งที่	พิกัดอินพุตที่กำหนดให้เป็นเป้าหมาย			พิกัดที่แขนกลเคลื่อนที่ไป		
	x	y	z	x	y	z
1	250	300	0	255	300	0
	150	300	0	155	300	0
	250	200	0	255	200	0
	250	300	0	255	300	0
เปอร์เซ็นต์ความผิดพลาด				2.222%	0	0
2	250	-300	0	255	-300	0
	150	-300	0	155	-300	0
	250	-200	0	255	-200	0
	250	-300	0	255	-300	0
เปอร์เซ็นต์ความผิดพลาด				2.222%	0	0
3	200	-200	0	204	-200	0
	200	-250	0	203	-250	0
	150	-250	0	153	-250	0
	200	-200	0	204	-200	0
เปอร์เซ็นต์ความผิดพลาด				1.867%	0	0

ตารางที่ 4.6 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพสี่เหลี่ยม 2 มิติ

ครั้งที่	พิกัดอินพุตที่กำหนดให้เป็นเป้าหมาย			พิกัดที่แขนกลเคลื่อนที่ไป		
	x	y	z	x	y	z
1	250	250	0	247	250	0
	200	250	0	197	250	0
	200	200	0	197	200	0
	250	250	0	247	250	0
	250	250	0	247	250	0
เปอร์เซ็นต์ความผิดพลาด				1.304%	0	0
2	250	-250	0	250	-250	0
	200	-250	0	198	-250	0
	200	-200	0	198	-200	0
	250	-200	0	247	-200	0
	250	-250	0	250	-250	0
เปอร์เซ็นต์ความผิดพลาด				0.609%	0	0
3	150	-250	0	149	-248	0
	150	-300	0	149	-298	0
	100	-300	0	99	-298	0
	100	-250	0	99	-248	0
	150	-250	0	149	-248	0
เปอร์เซ็นต์ความผิดพลาด				0.769%	0.741%	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.7 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพสี่เหลี่ยมผืนผ้า 2 มิติ

ครั้งที่	พิกัดอินพุตที่กำหนดให้เป็นเป้าหมาย			พิกัดที่แขนกลเคลื่อนที่ไป		
	x	y	z	x	y	z
1	250	300	0	245	300	0
	250	300	0	245	300	0
	150	200	0	145	200	0
	150	200	0	145	200	0
	250	300	0	245	300	0
เปอร์เซ็นต์ความผิดพลาด				2.381%	0	0
2	250	300	0	245	300	0
	200	300	0	195	300	0
	200	200	0	195	200	0
	250	200	0	245	200	0
	250	300	0	245	300	0
เปอร์เซ็นต์ความผิดพลาด				2.174%	0	0
3	250	-300	0	252	-298	0
	200	-300	0	201	-298	0
	200	-200	0	199	-198	0
	250	-200	0	248	-198	0
	250	-300	0	252	-298	0
เปอร์เซ็นต์ความผิดพลาด				0.174%	0.769%	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.8 การทดลองในการควบคุมแกนกลให้มีการเคลื่อนที่วาดภาพสี่เหลี่ยมคางหมู 2 มิติ

ครั้งที่	พิกัดอินพุตที่กำหนดให้เป็นเป้าหมาย			พิกัดที่แกนกลเคลื่อนที่ไป		
	x	y	z	x	y	z
1	250	-300	0	250	-298	0
	200	-250	0	200	-248	0
	200	-200	0	200	-198	0
	250	-200	0	250	-198	0
	250	-300	0	250	-298	0
เปอร์เซ็นต์ความผิดพลาด				0	0.800%	0
2	250	300	0	245	300	0
	200	250	0	195	250	0
	200	200	0	195	200	0
	250	200	0	245	200	0
	250	300	0	245	300	0
เปอร์เซ็นต์ความผิดพลาด				2.174%	0	0
3	250	300	0	245	300	0
	200	300	0	195	300	0
	200	250	0	195	250	0
	250	300	0	245	300	0
	250	300	0	245	300	0
เปอร์เซ็นต์ความผิดพลาด				2.174%	0	0

ตารางที่ 4.9 การทดลองในการควบคุมแขนกลให้มีการเคลื่อนที่วาดภาพนาฬิกาทราย 2 มิติ

ครั้งที่	พิกัดอินพุตที่กำหนดให้เป็นเป้าหมาย			พิกัดที่แขนกลเคลื่อนที่ไป		
	x	y	z	x	y	z
1	250	250	0	245	253	0
	200	300	0	195	303	0
	200	250	0	195	253	0
	250	300	0	245	303	0
	250	250	0	245	253	0
เปอร์เซ็นต์ความผิดพลาด				2.174%	1.111%	0
2	250	-250	0	252	-247	0
	200	-300	0	202	-297	0
	200	-250	0	202	-247	0
	250	-300	0	252	-297	0
	250	-250	0	252	-247	0
เปอร์เซ็นต์ความผิดพลาด				0.870%	1.111%	0
3	250	-250	0	252	-248	0
	200	-250	0	202	-248	0
	250	-300	0	252	-298	0
	200	-300	0	202	-298	0
	250	-250	0	252	-248	0
เปอร์เซ็นต์ความผิดพลาด				0.870%	0.741%	0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.10 แสดงเปอร์เซ็นต์ค่าความคลาดเคลื่อนจริงที่แขนกลเคลื่อนที่ไปในพิกัดแต่ละตำแหน่งในการวาดภาพ 2 มิติ

รูปแบบ	x	y	z
สามเหลี่ยม	2.222%	0	0
	2.222%	0	0
	1.867%	0	0
ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน	2.1036%	0	0
สี่เหลี่ยม	1.304%	0	0
	0.609%	0	0
	0.796%	0.741%	0
ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน	0.903%	0.247%	0
สี่เหลี่ยมผืนผ้า	2.381%	0	0
	2.174%	0	0
	0.174%	0.769%	0
ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน	1.5763%	0.2563%	0
สี่เหลี่ยมคางหมู	0	0.8000%	0
	2.174%	0	0
	2.174%	0	0
ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน	1.4493%	0.2666%	0
นาฬิกาทราย	2.174%	1.111%	0
	0.870%	1.111%	0
	0.870%	0.741%	0
ค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน	1.3046%	0.9876%	0

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ปริญญาโทฉบับนี้ได้ทำการประยุกต์ PLC เพื่อควบคุมแขนกล SCARA Robot 4DOF ให้มีการเคลื่อนที่ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ ซึ่งได้มีการออกแบบโปรแกรมการทำงานซึ่งมีการรับค่าข้อมูลอินพุตจากผู้ใช้งาน จากนั้นมาทำการคำนวณเพื่อหาค่ามุมของข้อต่อแขนกลตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบย้อนกลับ โดยค่าที่ได้ก็จะถูกนำมาคำนวณเพื่อหาค่าของสัญญาณพัลส์และ ความเร็วของมอเตอร์เพื่อนำไปขับมอเตอร์ให้แขนกลเกิดการเคลื่อนที่ไปยังตำแหน่งเป้าหมายตามที่กำหนด โดยมีการอ้างอิงมาจากบทที่ 4 ผลการทดลองในตารางที่ 4.1 โดยการทดลองนี้เป็นการกำหนดอินพุตขึ้นมาทั้งหมด 5 ชุดเป็นชุดพิกัดของแกน (x, y, z) โดยจะทำการทดลองในโปรแกรม MATLAB Simulink เป็นตัวอ้างอิง และทดลองกับแขนกล SCARA Robot 4DOF เป็นตัวเปรียบเทียบเพื่อหาค่าของมุมข้อต่อ ณ จุดใดๆที่กำหนด โดยจะมีการเปรียบเทียบความคลาดเคลื่อนว่าแขนกล SCARA Robot 4DOF มีความคลาดเคลื่อนจาก SCARA Robot 4DOF ในโปรแกรม MATLAB Simulink ไปมากน้อยเท่าไรจากมุมที่ถูกคำนวณมาในโปรแกรม MATLAB Simulink แล้วก็ทำการบันทึกผล ตารางที่ 4.10 แสดงเปอร์เซ็นต์ค่าความคลาดเคลื่อนจริงที่แขนกลเคลื่อนที่ไปในพิกัดแต่ละตำแหน่งในการวาดภาพ 2 มิติ สามเหลี่ยม สี่เหลี่ยม สี่เหลี่ยมผืนผ้า สี่เหลี่ยมคางหมู และ รูปนาฬิกาทราย ผลการทดลองปรากฏว่ารูปสามเหลี่ยมมีการทดลองให้แขนกลวาด 3 ครั้งมีค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อนของแกน x อยู่ที่ 2.1036% แกน y อยู่ที่ 0% แกน Z อยู่ที่ 0% รูปสี่เหลี่ยมมีการทดลองให้แขนกลวาด 3 ครั้งมีค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน ของแกน x อยู่ที่ 0.903% แกน y อยู่ที่ 0.247% แกน Z อยู่ที่ 0% รูปสี่เหลี่ยมผืนผ้ามีการทดลองให้แขนกลวาด 3 ครั้งมีค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน ของแกน x อยู่ที่ 0.174% แกน y อยู่ที่ 0.769% แกน Z อยู่ที่ 0% รูปสี่เหลี่ยมคางหมูมีการทดลองให้แขนกลวาด 3 ครั้งมีค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน ของแกน x อยู่ที่ 1.4493% แกน y อยู่ที่ 0.2666% แกน Z อยู่ที่ 0% รูปนาฬิกาทรายมีการทดลองให้แขนกลวาด 3 ครั้งมีค่าเฉลี่ยเปอร์เซ็นต์ความคลาดเคลื่อน ของแกน x อยู่ที่ 1.3046% แกน y อยู่ที่ 0.9876% แกน Z อยู่ที่ 0%

5.2 ปัญหาและอุปสรรคที่พบในระหว่างดำเนินการ

1) ปัญหาเกี่ยวกับค่าที่ได้จากการคำนวณทั้งจำนวนพัลส์ที่ใช้เป็นระยะการเคลื่อนที่และความเร็วของมอเตอร์แต่ละแกน จากฟังก์ชัน PULS(886) และ SPED(885) ไม่สามารถใช้ข้อมูลตัวเลขจำนวนจริงหรือประเภทเลขทศนิยมได้ (Float) ทางผู้จัดทำโครงการได้ใช้ฟังก์ชัน FIXL(451) Floating to 32 bit ในการแปลงตัวเลขจากการคำนวณที่เป็นจำนวนจริงให้เป็นจำนวนเต็ม(Integer) เพื่อใช้ในฟังก์ชัน PULSE(886) SPED(885) และเมื่อแปลงแล้วจะทำให้ความเร็วของมอเตอร์นั้นลดลงจึงทำให้เกิดเป็นเส้นโค้งในการวาดภาพ 2 มิติ

2.ปัญหาที่เกี่ยวกับโปรแกรมโดยเฉพาะโปรแกรมที่ใช้ในการส่งค่าพิกัดหรือชุดตัวเลข G-code ไปยัง PLC จะต้องมีการสร้างเชิร์ฟเวอร์ขึ้นมาเพื่อเป็นตัวกลางในการส่งข้อมูลจากโปรแกรม Microsoft excel ไปยัง Data memory ของ PLC โดยการส่งนั้นจะมีความยุ่งยากในทั้งเรื่องของการตั้งชื่อคอลัมน์ในแต่ละคอลัมน์ของโปรแกรม Microsoft excel เพื่อที่จะได้ส่งค่าที่ต้องการส่งในคอลัมน์นั้นๆได้ถูก และการที่ส่งข้อมูลชุดตัวเลขนั้นจะต้องมีการเรียนรู้เกี่ยวกับ Data type ของชุดข้อมูลที่ผู้ส่ง เช่น การส่งแบบ Dword float หรือString โดยข้อมูลบางชนิดจะไม่สามารถส่งได้หากใช้ Data type ผิด เช่น การส่งเลขทศนิยมจะต้องใช้ Data type float การส่งข้อมูลแบบตัวหนังสือจะต้องใช้ Data type String เป็นต้น

3. ปัญหาที่เกี่ยวกับการวาดภาพกราฟิกบนพื้น 2 มิติ เรื่องของการออกแบบเครื่องมือปกอกปากกาส่วนปลายที่ใช้กับ SACARA Robot 4DOF เนื่องจากมีระยะช่องว่างระหว่างตัวปากกากับตัวปกที่ทำให้เกิดการคลาดเคลื่อนในการวาดภาพเล็กน้อย ในขณะที่แขนกลเคลื่อนที่วาดภาพกราฟิกอยู่ปากกาจะมีการสั่นทำให้ภาพที่ได้เป็นเส้นปะหรือไม่เป็นเส้นเดียวกัน

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

ข้อเสนอแนะค่าพิกัดที่ควรป้อนต้องไม่เกินระยะของแขนกล SACARA Robot 4DOF หรือตามสมการที่ 3.8 เมื่อป้อนค่าพิกัดตำแหน่งเป้าหมายเรียบร้อยแล้วควรสังเกตค่าความเร็วของแต่ละแกนที่คำนวณได้ ค่าที่แนะนำไม่ควรเกิน 2500 หากมากกว่าจะทำให้การเคลื่อนที่ของแขนกลนั้นเร็วอาจเกิดการเสียหายได้ ไม่ควรนำมือไปขัดขวางการเคลื่อนที่ของแขนกล หากมีข้อผิดพลาดในการเคลื่อนที่ไม่ตรงกับตำแหน่งพิกัดเป้าหมาย สามารถเปลี่ยนการควบคุมในโหมด manual ด้วยผู้ใช้งาน

แนวทางการพัฒนาต่อสำหรับ SCARA Robot 4DOF ในส่วนของ Ladder diagram การทำให้แขนกลเคลื่อนที่เป็นลำดับขั้นตอนสามารถทำได้หลายรูปแบบแล้วแต่การวิเคราะห์ของผู้พัฒนา สำหรับ Ladder diagram ในโครงการนี้สามารถป้อนค่าพิกัดตำแหน่งเป้าหมายเพียง 5 ตำแหน่งเท่านั้น สามารถ

พัฒนาเพิ่มตำแหน่งให้มากขึ้นหรือปรับเปลี่ยนรูปแบบลำดับขั้นตอนที่แตกต่างออกไป และเพิ่มการทำงานของมอเตอร์แกนที่ 4 ให้สามารถใช้งานได้ ส่วนนี้จะอยู่ที่ปลายแขนกล ซึ่งทางผู้จัดทำโครงการยังไม่ได้เขียนโปรแกรมใช้งาน และในส่วนของ structure text การคำนวณควรรศึกษาสมการจลนศาสตร์การเคลื่อนที่แบบย้อนกลับให้ถี่ถ้วน สมการจะมีการเปลี่ยนแปลงเมื่อเพิ่มการเคลื่อนที่ให้แกนที่ 4 ทำให้การควบคุมมีความซับซ้อนแต่สามารถเคลื่อนที่ไปยังตำแหน่งจุดอับของแขนกลได้มาก ในส่วนของโปรแกรมคือจะเลือกใช้โปรแกรมที่สร้างเซิร์ฟเวอร์ ผู้จัดทำใช้เป็นเวอร์ชันทดลองใช้โดยจะสามารถรันเซิร์ฟเวอร์ได้เพียง 2 ชั่วโมงเท่านั้นหากครบกำหนดเวลาแล้วตัวของเซิร์ฟเวอร์ก็จะถูกปิดลงทำให้ต้องทำการรีเซิร์ฟเวอร์ใหม่ ทำให้ทำงานเกิดการสะดุดขณะทดลองงาน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] by Courtney Taylor สถิติ: Degrees of Freedom <https://th.eferrit.com/degrees-of-freedom/>
- [2] พงษ์ศักดิ์ ชินนาบุญ และวีระชัย ลี้มพรชัยเจริญ จลศาสตร์ของการเคลื่อนที่เชิงเส้น <https://pws.npru.ac.th/nattapon/data/files/>
- [3] <https://learn.thairobotics.org/courses/basic-mechanics-statics/>
- [4] เพชรฟ้า ชาติสมบูรณ์ ภาณุวัฒน์ พลงาม อภิวัดน์ คำเชื่อน การประยุกต์ PLC ควบคุม SCARA Robot 4DOF ตามจลนศาสตร์การเคลื่อนที่ของแขนกลแบบไปข้างหน้าและแบบย้อนกลับ <https://opacimages.lib.kmitl.ac.th/medias/>
- [5] Programmable Logic Controller AMCI, REALPARS, ADVANCE <https://flutech.co.th/what-is-a-plc-or-programmable-logic-controller/>
- [6] (AWS) หน่วยประมวลผลกลาง <https://aws.amazon.com/th/what-is/cpu/>
- [7] Memory Unit <https://sinanew.wordpress.com/> <https://sinanew.wordpress.com>
- [8] สำนักเทคโนโลยีสารสนเทศ Power Supply <https://www.dol.go.th/it/PagesSupply.aspx>
- [9] Program_Basics_fod_tha https://www.mitsubishielectric.com/fa/assist/e-learning/pdf/tha/1-Program_Basics_fod_tha
- [10] อ.พินันทา ฉัตรวัฒนา Logic System http://km.pmitc.ac.th/files/15111113135251983_1711130993753
- [11] Margaret Rouse OLE for Process Control <https://www.techopedia.com/definition/1307/ole-for-process-control-opc>
- [12] KEPServerEX6 Case Studies KEPServerEX6 <https://www.ptc.com/en/products/kepware/kepserverex>
- [13] About Dynamic Data Exchange Microsoft <https://learn.microsoft.com/en-us/windows/win32/dataxchg/about-dynamic-data-exchange>
- [14] STEPPING MOTOR http://narong.ece.engr.tu.ac.th/ei444/document/15-stepper_motor



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OMRON

CP series CP1H CPU Unit CP1H-X□□D□-□/CP1H-Y□□D□-□ CP1H-XA□□D□-□

4 Axis Position Control and Comprehensive Programmable Controller

- The CP1H-X with pulse outputs for 4 axes.
- The CP1H-Y with 1-MHz pulse I/O.
- The CP1H-XA with pulse outputs for 4 axes and built-in analog I/O.



Features

- Pulse output for 4 axes. Advanced power for high-precision positioning control.
- High-speed counters. Differential phases for 4 axes.
- Easily handles multi-axis control with a single unit.
- Eight interrupt inputs are built in. Faster processing of approximately 500 instructions speeds up the entire system.
- Serial communications. Two ports. Select Option Boards for either RS-232C or RS-485 communications.
- Ethernet Communications. Enabled by using an Option Board. Two ports can be used as an Ethernet port to perform. Ethernet communications between the CP1H and a host computer.
- Built-in Analog I/O. XA CPU Units provide 4 input words and 2 output words.
- USB Peripheral Port. Another standard feature.
- The structured text (ST) language. Makes math operations even easier.
- Can be used for the CP1W series and CJ series Unit. The extensibility of it is preeminently good.
- LCD displays and settings. Enabled using Option Board.

OMRON

1

รูปที่ ก-1 คุณสมบัติ CP1H หน้าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1H

Model Number Structure

■ Model Number Legend (Not all models that can be represented with the model number legend can necessarily be produced.)

CP1H-□□D□-□
 (1) (2) (3) (4)




- | | |
|---|-----------------------------------|
| 1. Class | 3. Output classification |
| X : Basic model | R : Relay output |
| XA : Built-in analog I/O terminals | T : Transistor Output (sinking) |
| Y : Dedicated pulse I/O terminals | T1 : Transistor Output (sourcing) |
| 2. Number of Built-In number I/O points | 4. Power supply |
| 40 : 40 I/O points | A : AC |
| 20 : 20 I/O points | D : DC |

Ordering Information

Applicable standards







Refer to the OMRON website (www.ia.omron.com) or ask your OMRON representative for the most recent applicable standards for each model.

■ CPU Units

CPU Unit	CPU type	Specifications				Model
		Power supply	Output method	Inputs	Outputs	
 CP1H-X CPU Units Memory capacity: 20K steps High-speed counters: 100 kHz, 4 axes Pulse outputs: 100 kHz, 4 axes (Models with transistor outputs only)		AC power supply	Relay output	24	16	CP1H-X40DR-A
		DC power supply	Transistor output (sinking)			CP1H-X40DT-D
			Transistor output (sourcing)			CP1H-X40DT1-D
 CP1H-XA CPU Units Memory capacity: 20K steps High-speed counters: 100 kHz, 4 axes Pulse outputs: 100 kHz, 4 axes (Models with transistor outputs only) Analog inputs: 4 Analog outputs: 2		AC power supply	Relay output	24	16	CP1H-XA40DR-A
		DC power supply	Transistor output (sinking)			CP1H-XA40DT-D
			Transistor output (sourcing)			CP1H-XA40DT1-D
 CP1H-Y CPU Units Memory capacity: 20K steps High-speed counters: 1 MHz, 2 axes 100 kHz, 2 axes Pulse outputs: 1 MHz, 2 axes 100 kHz, 2 axes		DC power supply	Transistor output (sinking)	12 + line-driver input, 2 axes	8 + line-driver output, 2 axes	CP1H-Y20T-D

Note: 1. CP1H PLCs are supported by CX-Programmer version 6.1 or higher.
 2. Purchase a separately sold Option Unit if you will use RS-232C, RS-422A/485, Ethernet, or LCD.

■ Options for CPU Units

Name	Specifications	Model
 RS-232C Option Board		CP1W-CIF01
 RS-422A/485 Option Board	Can be mounted in either CPU Unit Option Board slot 1 or 2.	CP1W-CIF11
 RS-422A/485 (Isolated-type) Option Board		CP1W-CIF12-V1
 Ethernet Option Board	Can be mounted in either CPU Unit Option Board slot 1 or 2. *	CP1W-CIF41
 LCD Option Board	Can be mounted only in the CPU Unit Option Board slot 1.	CP1W-DAM01
 Memory Cassette	Can be used for backing up programs or auto-booting.	CP1W-ME05M

* When using CP1W-CIF41 Ver.1.0, one Ethernet port can be added.

Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
 Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

CP1H

General Specifications

Item	Type	AC power supply models	DC power supply models
	Model	CP1H-□□□-A	CP1H-□□□-D
Power supply		100 to 240 VAC 50/60 Hz	24 VDC
Operating voltage range		85 264 VAC	20.4 to 26.4 VDC (with 4 or more Expansion Units and Expansion I/O Units: 21.6 to 26.4 VDC)
Power consumption		100 VA max. (CP1H-□□□-A)(page 28)	50 W max. (CP1H-□□□-D)(page 28)
Inrush current (See note.)		100 to 120 VAC inputs: 20 A max. (for cold start at room temperature) 8 ms max. 200 to 240 VAC inputs: 40 A max. (for cold start at room temperature), 8 ms max.	30 A max. (for cold start at room temperature) 20 ms max.
External power supply		300 mA at 24 VDC	None
Insulation resistance		20 M Ω min. (at 500 VDC) between the external AC terminals and GR terminals	No insulation between primary and secondary for DC power supply
Dielectric strength		2,300 VAC at 50/60 Hz for 1 min between the external AC and GR terminals, leakage current: 5 mA max.	No insulation between primary and secondary for DC power supply
Noise immunity		Conforms to IEC 61000-4-4, 2 kV (power supply line)	
Vibration resistance		Conforms to JIS C60068-2-6, 10 to 57 Hz, 0.075-mm amplitude, 57 to 150 Hz, acceleration: 9.8 m/s ² in X, Y, and Z directions for 80 minutes each. Sweep time: 8 minutes \times 10 sweeps = total time of 80 minutes)	
Shock resistance		Conforms to JIS C60068-2-27, 147 m/s ² three times each in X, Y, and Z directions	
Ambient operating temperature		0 to 55°C	
Ambient humidity		10% to 90% (with no condensation)	
Ambient operating environment		No corrosive gas	
Ambient storage temperature		-20 to 75°C (Excluding battery)	
Power holding time		10 ms min.	2 ms min.

Note: The above values are for a cold start at room temperature for an AC power supply, and for a cold start for a DC power supply.

- A thermistor (with low-temperature current suppression characteristics) is used in the inrush current control circuitry for the AC power supply. The thermistor will not be sufficiently cooled if the ambient temperature is high or if a hot start is performed when the power supply has been OFF for only a short time. In those cases the inrush current values may be higher (as much as two times higher) than those shown above. Always allow for this when selecting fuses and breakers for external circuits.
- A capacitor charge-type delay circuit is used in the inrush current control circuitry for the DC power supply. The capacitor will not be charged if a hot start is performed when the power supply has been OFF for only a short time, so in those cases the inrush current values may be higher (as much as two times higher) than those shown above.

CP1H

Performance Specifications

Item	Type Models	CP1H-XA CPU Units	CP1H-X CPU Units	CP1H-Y CPU Units
		CP1H-XA□□□□□	CP1H-X□□□□□	CP1H-Y□□□□□
Control method		Stored program method		
I/O control method		Cyclic scan with immediate refreshing		
Program language		Ladder diagram		
Function blocks		Maximum number of function block definitions: 128 Maximum number of instances: 256 Languages usable in function block definitions: Ladder diagrams, structured text (ST)		
Instruction length		1 to 7 steps per instruction		
Instructions		Approx. 500 (function codes: 3 digits)		
Instruction execution time		Basic instructions: 0.10 μs min. Special instructions: 0.15 μs min.		
Common processing time		0.7 ms		
Program capacity		20K steps		
Number of tasks		288 (32 cyclic tasks and 256 interrupt tasks)		
Scheduled interrupt tasks		1 (interrupt task No. 2, fixed)		
	Input interrupt tasks	8 (interrupt task No. 140 to 147, fixed) (Interrupt tasks can also be specified and executed for high-speed counter interrupts.)	6 (interrupt task No. 140 to 145, fixed)	
Maximum subroutine number		256		
Maximum jump number		256		
I/O areas (See note.)	Input bits	272bits (17 words) : CIO 0.00 to 16.15		
	Output bits	272bits (17 words) : CIO 100.00 to 116.16		
	Built-in Analog Inputs	CIO 200 to CIO 203	---	
	Built-in Analog Outputs	CIO 210 to CIO 211	---	
	Serial PLC Link Area	1,440 bits (90 words) : CIO 3100.00 to CIO 3189.15 (CIO 3100 to CIO 3189)		
Work bits	8,192 bits (512 words) : W0.00 to W511.15 (W0 to W511) CIO Area: 37,504 bits (2,344 words) : CIO 3800.00 to CIO 6143.15 (CIO 3800 to CIO 6143)			
TR Area	16 bits: TR0 to TR15			
Holding Area	8,192 bits (512 words) : H0.00 to H511.15 (H0 to H511)			
AR Area	Read-only (Write-prohibited): 7168 bits (448 words) : A0.00 to A447.15 (A0 to A447) Read/Write: 8192 bits (512 words) : A448.00 to A959.15 (A448 to A959)			
Timers	4,096 bits: T0 to T4095			
Counters	4,096 bits: C0 to C4095			
DM Area	32 Kwords: D0 to D32767			
Data Register Area	16 registers (16 bits): DR0 to DR15			
Index Register Area	16 registers (32 bits): IR0 to IR15			
Task Flag Area	32 flags (32 bits): TK0000 to TK0031			
Trace Memory	4,000 words (500 samples for the trace data maximum of 31 bits and 6 words.)			
Memory Cassette	A special Memory Cassette (CP1W-MED5M) can be mounted. Note: Can be used for program backups and auto-booting.			
Clock function	Supported. Accuracy (monthly deviation): -4.5 min to -0.5 min (ambient temperature: 55°C), -2.0 min to +2.0 min (ambient temperature: 25°C), -2.5 min to +1.5 min (ambient temperature: 0°C)			
Communications functions	One built-in peripheral port (USB 1.1): For connecting Support Software only.			
	A maximum of two Serial Communications Option Boards can be mounted. A maximum of two Ethernet Option Boards can be mounted. When using CP1W-CIF41 Ver.1.0, one Ethernet Option Board can be mounted.			
Memory backup	Flash memory: User programs, parameters (such as the PLC Setup), comment data, and the entire DM Area can be saved to flash memory as initial values. Battery backup: The Holding Area, DM Area, and counter values (flags, PV) are backed up by a battery.			
Battery service life	5 years at 25°C. (Use the replacement battery within two years of manufacture.)			
Built-in input terminals	40 (24 inputs, 16 outputs)	20 (12 inputs, 8 outputs) Line-driver inputs: Two axes for phases A, B, and Z Line-driver outputs: Two axes for CW and CCW		
Number of connectable Expansion (I/O) Units	CP Expansion I/O Units: 7 max.; CJ-series Special I/O Units or CPU Bus Units: 2 max.			
Max. number of I/O points	320 (40 built in + 40 per Expansion (I/O) Unit × 7 Units)		300 (20 built in + 40 per Expansion (I/O) Unit × 7 Units)	
Interrupt inputs	8 inputs (Shared by the external interrupt inputs (counter mode) and the quick-response inputs.)		6 inputs (Shared by the external interrupt inputs (counter mode) and the quick-response inputs.)	
Interrupt input counter mode	8 inputs (Response frequency: 5 kHz max. for all interrupt inputs), 16 bits Up or down counters		6 inputs (Response frequency: 5 kHz max. for all interrupt inputs), 16 bits Up or down counters	
Quick-response inputs	8 points (Min. input pulse width: 50 μs max.)		6 points (Min. input pulse width: 50 μs max.)	
Scheduled interrupts	1			

CP1H

Type		CP1H-XA CPU Units	CP1H-X CPU Units	CP1H-Y CPU Units
Item	Models	CP1H-XA□□□□□	CP1H-X□□□□□	CP1H-Y□□□□□
High-speed counters		4 inputs: Differential phases (4x), 50 kHz or Single-phase (pulse plus direction, up/down, increment), 100 kHz Value range: 32 bits, Linear mode or ring mode Interrupts: Target value comparison or range comparison		2 inputs: Differential phases (4x), 500 kHz or Single-phase, 1 MHz and 2 inputs: Differential phases (4x), 50 kHz or Single-phase (pulse plus direction, up/down, increment), 100 kHz Value range: 32 bits, Linear mode or ring mode Interrupts: Target value comparison or range comparison
Pulse outputs (models with transistor outputs only)	Pulse outputs	Trapezoidal or S-curve acceleration and deceleration (Duty ratio: 50% fixed) 4 outputs, 1 Hz to 100 kHz (CCW/CW or pulse plus direction)		Trapezoidal or S-curve acceleration and deceleration (Duty ratio: 50% fixed) 2 outputs, 1 Hz to 1 MHz (CCW/CW or pulse plus direction) 2 outputs, 1 Hz to 100 kHz (CCW/CW or pulse plus direction)
	PWM outputs	Duty ratio: 0.0% to 100.0% (Unit: 0.1%) 2 outputs, 0.1 to 6553.5 Hz (Accuracy: ±5% at 1 kHz)		
Built-in analog I/O terminals		4 analog inputs and 2 analog outputs	None	
Analog control		1 (Setting range: 0 to 255)		
External analog input		1 input (Resolution: 1/256, Input range: 0 to 10 V), not isolated		

Note: The memory areas for CJ-series Special I/O Units and CPU Bus Units are allocated at the same as for the CJ-series. For details, refer to the CJ Series catalog (Cat. No. P052).

รูปที่ ก-5 คุณสมบัติ CP1H หน้าที่ 5

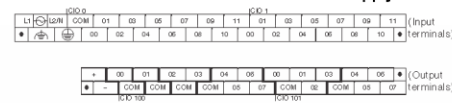
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1H

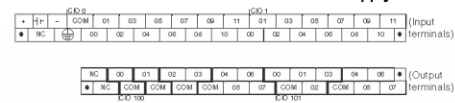
Built-in Inputs / Built-in Outputs

■ Terminal Block Arrangement

● CP1H-XA and X CPU Units with AC Power Supply



● CP1H-XA and X CPU Units with DC Power supply



■ Built-in Input Area

● CP1H-XA and X CPU Units

PLC Setup	Input operation			High-speed counter operation	Pulse output origin search function set to be used.
	Normal inputs	Interrupt inputs	Quick-response inputs	High-speed counters	
CIO 0	00	Normal input 0	Interrupt input 0	Quick-response input 0	Pulse 0: Origin input signal
	01	Normal input 1	Interrupt input 1	Quick-response input 1	Pulse 0: Origin proximity input signal
	02	Normal input 2	Interrupt input 2	Quick-response input 2	Pulse output 1: Origin input signal
	03	Normal input 3	Interrupt input 3	Quick-response input 3	Pulse output 1: Origin proximity input signal
	04	Normal input 4			High-speed counter 2 (phase-A, increment, or count input)
	05	Normal input 5			High-speed counter 2 (phase-B, decrement, or direction input)
	06	Normal input 6			High-speed counter 1 (phase-A, increment, or count input)
	07	Normal input 7			High-speed counter 1 (phase-B, decrement, or direction input)
	08	Normal input 8			High-speed counter 0 (phase-A, increment, or count input)
	09	Normal input 9			High-speed counter 0 (phase-B, decrement, or direction input)
	10	Normal input 10			High-speed counter 3 (phase-A, increment, or count input)
	11	Normal input 11			High-speed counter 3 (phase-B, decrement, or direction input)
CIO 1	00	Normal input 12	Interrupt input 4	Quick-response input 4	High-speed counter 3 (phase-Z/reset)
	01	Normal input 13	Interrupt input 5	Quick-response input 5	Pulse output 2: Origin input signal
	02	Normal input 14	Interrupt input 6	Quick-response input 6	Pulse output 2: Origin proximity input signal
	03	Normal input 15	Interrupt input 7	Quick-response input 7	Pulse output 3: Origin input signal
	04	Normal input 16			Pulse output 3: Origin proximity input signal
	05	Normal input 17			
	06	Normal input 18			
	07	Normal input 19			
	08	Normal input 20			
	09	Normal input 21			
	10	Normal input 22			
	11	Normal input 23			

■ Built-in Output Area

● CP1H-XA and CP1H-X CPU Units

Instruct-ions	When the instructions to the right are not executed	When a pulse output instruction (SPED, ACC, PLS2, or ORG) is executed		When the origin search function is set to be used in the PLC Setup, and an origin search is executed by the ORG instruction	When the PWM instruction is executed	
		CW/CCW	Pulse plus direction	When the origin search function is used	Variable duty ratio pulse output	
PLC Setup	Normal outputs	Fixed duty ratio pulse outputs			PWM output	
	CIO 100	00	Normal output 0	Pulse output 0 (CW)	Pulse output 0 (pulse)	
		01	Normal output 1	Pulse output 0 (CCW)	Pulse output 1 (pulse)	
		02	Normal output 2	Pulse output 1 (CW)	Pulse output 0 (direction)	
		03	Normal output 3	Pulse output 1 (CCW)	Pulse output 1 (direction)	
		04	Normal output 4	Pulse output 2 (CW)	Pulse output 2 (pulse)	
		05	Normal output 5	Pulse output 2 (CCW)	Pulse output 2 (direction)	
		06	Normal output 6	Pulse output 3 (CW)	Pulse output 3 (pulse)	
07		Normal output 7	Pulse output 3 (CCW)	Pulse output 3 (direction)		
CIO 101	00	Normal output 8			PWM output 0	
	01	Normal output 9			PWM output 1	
	02	Normal output 10			Origin search 0 (Error counter reset output)	
	03	Normal output 11			Origin search 1 (Error counter reset output)	
	04	Normal output 12			Origin search 2 (Error counter reset output)	
CIO 101	05	Normal output 13			Origin search 3 (Error counter reset output)	
	06	Normal output 14				
	07	Normal output 15				

CP1H

I/O Specifications for CPU Units

Input Specifications

ITEM	Specifications		
	High-speed counter inputs (phases A and B)	Interrupt inputs and quick-response inputs	Normal inputs
CP1H-XA/X CPU Units	CIO 0.04 to CIO 0.11	CIO 0.00 to CIO 0.03 and CIO 1.00 to CIO 1.03	CIO 1.04 to CIO 1.11
CP1H-Y CPU Units	CIO 0.04, CIO 0.05, CIO 0.10, CIO 0.11	CIO 0.00, CIO 0.01 and CIO 1.00 to CIO 1.03	CIO 1.04, CIO 1.05
Input voltage	24 VDC +10%/-15%		
Applicable sensors	2-wire sensors or 3-wire sensors		
Input impedance	3.0 kΩ		4.7 kΩ
Input current	7.5 mA typical		5 mA typical
ON voltage	17.0 VDC min.		14.4 VDC min.
OFF voltage/current	1 mA max. at 5.0 VDC		
ON delay	2.5 μs max.	50 μs max.	1 ms max.
OFF delay	2.5 μs max.	50 μs max.	1 ms max.
Circuit configuration			

High-speed Counter Function Input Specifications

CP1H-XA/X CPU Units (Input bits: CIO 0.04 to CIO 0.11)
 CP1H-Y CPU Units (Input bits: CIO 0.04, CIO 0.05, CIO 0.10, CIO 0.11)

Item	Specifications
ON/OFF delay	<ul style="list-style-type: none"> Pulse plus direction input mode Increment mode Up/down input mode Differential phase input mode

Interrupt Input Counter Mode

CP1H-XA/X CPU Units (Input bits: CIO 0.00 to CIO 0.03, CIO 1.00 to CIO 1.03)
 CP1H-Y CPU Units (Input bits: CIO 0.00, CIO 0.11, CIO 1.00 to CIO 1.03)

Item	Specifications
ON/OFF delay	

High-speed Counter Inputs (Line-driver Inputs)

CP1H-Y CPU Units

Item	Specifications	
High-speed counter inputs	Phases A and B	Phase Z
Input voltage	RS-422A line-driver, AM26LS31 or equivalent Note: The power supply voltage on the line-driver must be 5 V±5% max.	
Input type	Line-driver input	
Input current	10 mA typical	13 mA typical
Circuit configuration		
ON/OFF delay	<ul style="list-style-type: none"> Pulse plus direction input mode Increment mode Up/down input mode Differential phase input mode 	

CP1H

● Pulse outputs

CP1H-XA/X CPU Units: Output bits CIO 100.00 to CIO 100.07
 CP1H-Y CPU Units: Output bits CIO100.04 to CIO 100.07

Item	Specifications
Max. switching capacity	30 mA at 4.75 to 26.4 VDC
Min. switching capacity	7 mA at 4.75 to 26.4 VDC
Max. output frequency	100 kHz
Output waveform	

- Note: 1. The above values assume a resistive load and do not consider the impedance of the cable connecting the load.
 2. The pulse widths during actual use may be smaller than the ones shown above due to pulse distortion caused by connecting cable impedance.
 3. The OFF and ON refer to the output transistor. The output transistor is ON at level "L".

● Pulse Outputs (Line-driver Outputs)

CP1H-Y CPU Units

Item	Specifications
Pulse outputs	Line-driver outputs, Am26LS31 or equivalent
Max. output current	20 mA
Max. output frequency	1 MHz
Circuit configuration	

Note: Connect a load of 20 mA or less to the output. The Unit may be damaged if a current of more than 20 mA is output.

■ Analog I/O Specifications (CP1H-XA CPU Units Only)

Item		Voltage I/O	Current I/O
Analog Input Section	Number of analog inputs	4	
	Input signal range	0 to 5 V, 1 to 5 V, 0 to 10 V, or -10 to 10 V	0 to 20 mA or 4 to 20 mA
	Max. rated input	±15 V	±30 mA
	External input impedance	1 MΩ min.	Approx. 250 Ω
	Resolution	1/6,000 or 1/12,000 (full scale)	
	Overall accuracy	25°C: ±0.3% full scale/0 to 55°C: ±0.6% full scale	25°C: ±0.4% full scale/0 to 55°C: ±0.8% full scale
	A/D conversion data	Full scale for -10 to 10 V: F448 (E890) to 0BB8 (1770) hex Full scale for other ranges: 0000 to 1770 (2EE0) hex	
	Averaging	Supported (Set for individual inputs in the PLC Setup.)	
Analog Output Section	Open-circuit detection	Supported (Value when disconnected: 8000 Hex)	
	Number of outputs	2	
	Output signal range	0 to 5 V, 1 to 5 V, 0 to 10 V, -10 to 10 V	0 to 20 mA or 4 to 20 mA
	Allowable external output load resistance	1 kΩ min.	600 Ω max.
	External output impedance	0.5 Ω max.	---
	Resolution	1/6000 or 1/12000 (full scale)	
	Overall accuracy	25°C: ±0.4% of full scale, 0 to 55°C: ±0.8% of full scale	
	D/A conversion data	Full scale for -10 to 10 V: F448 (E890) to 0BB8 (1770) hex Full scale for other ranges: 0000 to 1770 (2EE0) hex	
Conversion time	1 ms/point		
Isolation method	Photocoupler isolation between analog I/O terminals and internal circuits. No isolation between analog I/O signals.		

Built-in Analog Input Switch (Factory Settings)



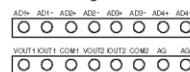
● Pulse outputs

CP1H-XA/X/Y CPU Units: Output bits CIO101.00, CIO 101.01

Item	Specifications
Max. switching capacity	30 mA at 4.75 to 26.4 VDC
Max. output frequency	1 kHz
PWM output precision	ON duty +5%, -0% at output frequency of 1 kHz
Output waveform	

- Note: 1. The above values assume a resistive load and do not consider the impedance of the cable connecting the load.
 2. The pulse widths during actual use may be smaller than the ones shown above due to pulse distortion caused by connecting cable impedance.
 3. The OFF and ON refer to the output transistor. The output transistor is ON at level "L".

Built-in Analog I/O Terminal Block Arrangement

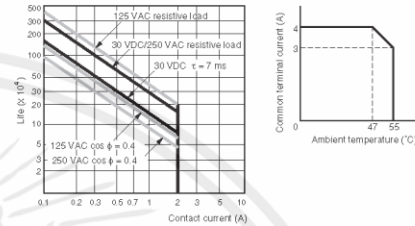


CP1H

■ Output Specifications
● CPU Units with Relay Outputs

Item	Specifications	
Max. switching capacity	2 A, 250 VAC (cosφ = 1), 2 A, 24 VDC 4 A/common	
Min. switching capacity	5 VDC, 10 mA	
Service life of relay	Electrical	
	Resistive load	100,000 operations (24 VDC)
	Inductive load	48,000 operations (250 VAC, cosφ = 0.4)
Mechanical	20,000,000 operations	
ON delay	15 ms max.	
OFF delay	15 ms max.	
Circuit configuration		

Note: Under the worst conditions, the service life of output contacts is as shown on the left.
The service life of relays is as shown in the following diagram as a guide line.



● CPU Units with Transistor Outputs (Sinking/Sourcing)

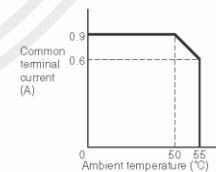
Item	Specifications		
CP1H-XA/X CPU Units	CIO 100.00 to CIO 100.07	CIO 101.00, CIO 101.01	CIO 101.02 to CIO 101.07
CP1H-Y CPU Units	CIO 100.04 to CIO 100.07	CIO 101.00, CIO 101.01	CIO 101.02, CIO 101.03
Max. switching capacity	4.5 to 30 VDC: 300 mA/point, 0.9 A/common, 3.6 A/Unit *1*2		
Min. switching capacity	4.5 to 30 VDC, 1 mA		
Leakage current	0.1 mA max.		
Residual voltage	0.6 V max.	1.5 V max.	
ON delay	0.1 ms max.		
OFF delay	0.1 ms max.		1 ms max.
Fuse	1/common *3		
Circuit configuration			

Note: 1. Do not apply a voltage or connect a load to an output terminal exceeding the maximum switching capacity.

*3 Also do not exceed 0.9 A for the total for CIO 100.00 to CIO 100.03. (CIO 100.00 to CIO 100.03 is different common.)

*4 A maximum of 0.9 A per common can be switched at an ambient temperature of 50 °C.

*5 Fuses cannot be replaced by the user.



รูปที่ ก-9 คุณสมบัติ CP1H หน้าที่ 9

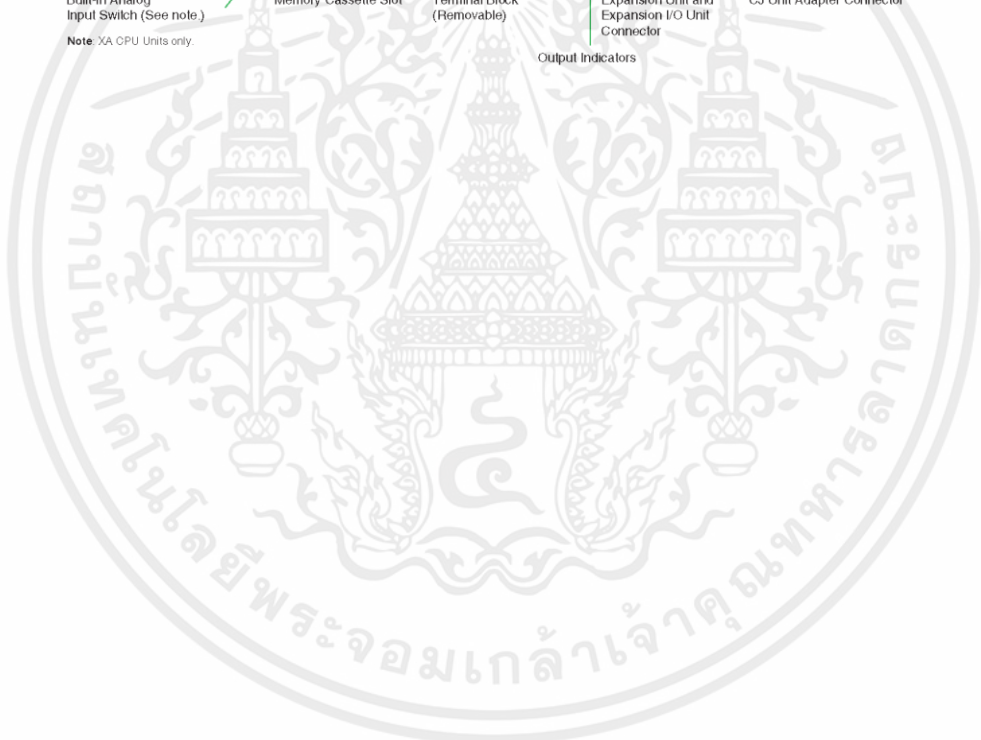
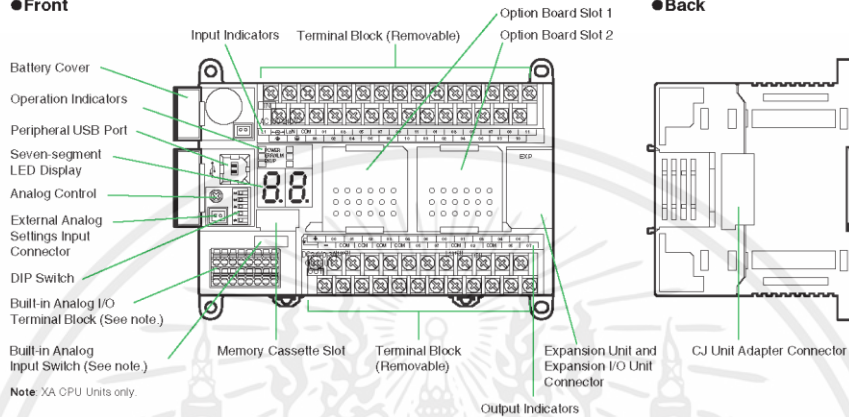
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP1H

External Interfaces

■ CPU Unit Nomenclature

● Front



รูปที่ ก-10 คุณสมบัติ CP1H หน้าที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

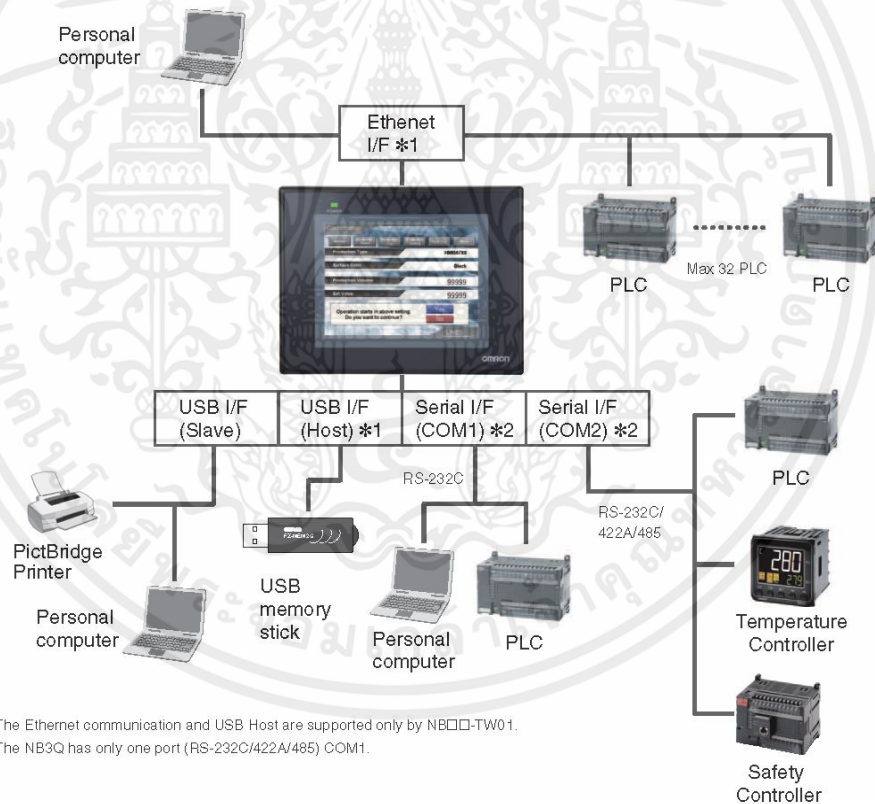
Programmable Terminals NB series

The feature-rich, economic programmable terminal

- More than 65,000 display colors TFT, color touch-screen for all models
- Available in sizes ranging from 3 to 10 inches
- Long-life LED backlight
- Serial, USB or Ethernet communication
- USB memory stick support



System Configuration



Windows is registered trademarks of Microsoft Corporation in the USA and other countries.
Microsoft product screen shot(s) reprinted with permission from Microsoft Corporation.
Apple and Apple logo are registered trademarks of Apple Inc. in the USA and other countries. App Store is a service mark of Apple Inc.
Google Play and the Google Play logo are trademarks of Google LLC.
Safari is trademark of Apple Inc., registered in the U.S. and other countries.
Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.
The product photographs and figures that are used in this catalog may vary somewhat from the actual products.

NB Series

Ordering Information

Programmable Terminals

Product name	Specifications	Model
NB3Q	3.5 inch, TFT LCD, Color, 320 × 240 dots	NB3Q-TW00B
	3.5 inch, TFT LCD, Color, 320 × 240 dots, USB Host, Ethernet	NB3Q-TW01B
NB5Q	5.6 inch, TFT LCD, Color, 320 × 234 dots	NB5Q-TW00B
	5.6 inch, TFT LCD, Color, 320 × 234 dots, USB Host, Ethernet	NB5Q-TW01B
NB7W	7 inch, TFT LCD, Color, 800 × 480 dots	NB7W-TW00B
	7 inch, TFT LCD, Color, 800 × 480 dots, USB Host, Ethernet	NB7W-TW01B
NB10W	10.1 inch, TFT LCD, Color, 800 × 480 dots, USB Host, Ethernet	NB10W-TW01B

Options

Product name	Specifications	Model
Software	Supported Operating Systems: Windows 10, Windows 8.1, Windows 8, Windows 7, Windows Vista®, Windows XP (SP3 or higher). Note: Except for Windows XP 64-bit version Download from Omron's regional websites.	NB-Designer *
NB-to-PLC Connecting cable	For NB to PLC via RS-232C (CP/CJ/CS), 2m	XW2Z-200T
	For NB to PLC via RS-232C (CP/CJ/CS), 5m	XW2Z-500T
	For NB to PLC via RS-422A/485, 2m	NB-RSEXT-2M
Display protective sheets	For the NB3Q contains 5 sheets	NB3Q-KBA04
	For the NB5Q contains 5 sheets	NB5Q-KBA04
	For the NB7W contains 5 sheets	NB7W-KBA04
	For the NB10W contains 5 sheets	NB10W-KBA04
Attachment	Mounting bracket for NT31/NT31C series to NB5Q series	NB5Q-ATT01

Note: Use a standard USB cable (Type A male to Type B male) or standard Ethernet cable (10 BASE-T/100 BASE-TX twisted-pair cable) to connect the NB series to software (personal computer).
Use a standard Ethernet cable (10 BASE-T/100 BASE-TX twisted-pair cable) to connect the NB series to a PLC. For detail, refer to the NB Series Setup Manual (Cat. No. V107).

- * The NB5Q-TW01B and NB7W-TW01B are supported by NB-Designer version 1.10 or higher.
- * The NB3Q-TW01B and NB10W-TW01B are supported by NB-Designer version 1.20 or higher.

System Configuration

Items	Minimum Configurations
Operating System (OS)	Microsoft Windows XP (SP3 or higher) Microsoft Windows Vista (32-bit or 64-bit edition) Microsoft Windows 7 (32-bit or 64-bit edition) Microsoft Windows 8 (32-bit or 64-bit edition) Microsoft Windows 8.1 (32-bit or 64-bit edition) Microsoft Windows 10 (32-bit or 64-bit edition)
CPU	Windows computers with CPU recommended for OS
Memory	512 MB or more
Hard Disk	2.5 GB or more, with the disk space more than 800 MB at least
Display	Supports the display with resolution of 800 × 600 and 16-bit high color (1024 × 768 and 32-bit true color are recommended.)
Communication ports	RS-232C Port, USB Port, Ethernet Port

Recommended USB Memory

Product name	Specifications	Model
USB memory stick	Capacity: 2 GB	FZ-MEM2G
	Capacity: 8 GB	FZ-MEM8G

รูปที่ ก-11 แสดงคุณสมบัติของHMI Omron NB5QTW00Bหน้าที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NB Series

Specifications

HMI

Specifications	NB3Q		NB5Q		NB7W		NB10W
	TW00B	TW01B	TW00B	TW01B	TW00B	TW01B	TW01B
Display type	3.5" TFT LCD		5.6" TFT LCD		7" TFT LCD		10.1" TFT LCD
Display resolution (H x V)	320 x 240		320 x 234		800 x 480		800 x 480
Number of colors	65,536						
Backlight	LED						
Backlight lifetime	50,000 hours of operating time at the normal temperature (25 °C)*						
Touch panel	Analog resistive membrane type, resolution 1024 x 1024, life: 1 million touch operations						
Dimensions in mm (H x W x D)	103.8 x 129.8 x 52.8		142 x 184 x 46		148 x 202 x 46		210.8 x 268.8 x 54.0
Weight	300 g max.	305 g max.	620 g max.	627 g max.	710 g max.	715 g max.	1525 g max.

* This is the estimated time when the luminous intensity is decreased by 50% per LED at room temperature and humidity. It is a typical value.

Functionality

Specifications	NB3Q		NB5Q		NB7W		NB10W
	TW00B	TW01B	TW00B	TW01B	TW00B	TW01B	TW01B
Internal memory	128MB (including system area)						
Memory interface	---	USB Memory	---	USB Memory	---	USB Memory	USB Memory
Serial (COM1)	RS-232C/422A/485 (not isolated), Transmission distance: 15m Max. (RS-232C), 500m Max. (RS-422A/485), Connector: D-Sub 9-pin		RS-232C, Transmission distance: 15 m Max., Connector: D-Sub 9-pin				
Serial (COM2)	---		RS-232C/422A/485 (not isolated), Transmission distance: 15m Max. (RS-232C), 500m Max. (RS-422A/485), Connector: D-Sub 9-pin				
USB Host	Equivalent to USB 2.0 full speed, type A, Output power 5V, 150mA						
USB Slave	Equivalent to USB 2.0 full speed, type B, Transmission distance: 5m						
Printer connection	PictBridge support						
Ethernet	---	10/100 base-T	---	10/100 base-T	---	10/100 base-T	10/100 base-T

General

Specifications	NB3Q		NB5Q		NB7W		NB10W
	TW00B	TW01B	TW00B	TW01B	TW00B	TW01B	TW01B
Line voltage	20.4 to 27.6 VDC (24 VDC -15 to 15%)						
Power consumption	5 W	9 W	6 W	10 W	7 W	11 W	14 W
Battery lifetime	5 years (at 25 °C)*						
Enclosure rating (front side)	Front operation part: IP65 (Dust proof and drip proof only from the front of the panel)						
Obtained standards	EC Directives, KC, cUL508						
Operating environment	No corrosive gases.						
Noise immunity	Compliant with IEC61000-4-4, 2KV (Power cable)						
Ambient operating temperature	0 to 50 °C						
Ambient operating humidity	10% to 90% RH (without condensation)						

Note: Flash memory can be written to up 100,000 times.

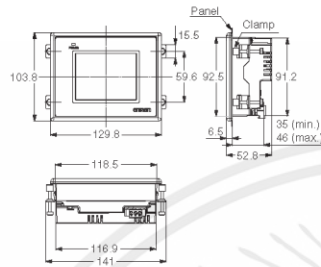
* OMRON accepts NB unit including the display device for repair and analysis within five years of the manufacture date.

NB Series

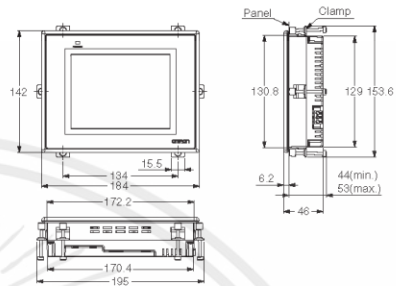
Dimensions

(Units: mm)

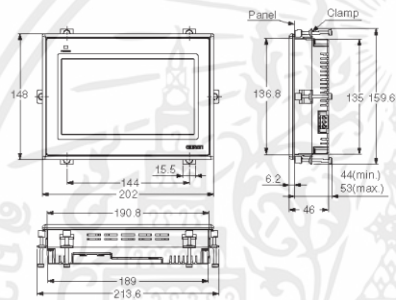
NB3Q



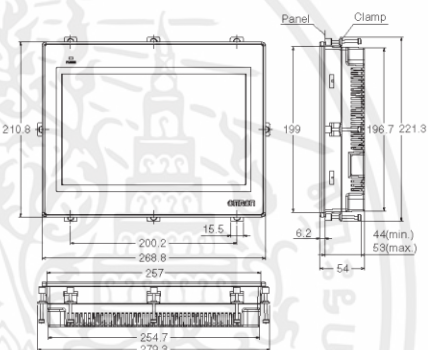
NB5Q



NB7W



NB10W



Model	Panel cutout (H x V mm)
NB3Q	119.0 (+0.5/-0) x 93.0 (+0.5/-0)
NB5Q	172.4 (+0.5/-0) x 131.0 (+0.5/-0)
NB7W	191.0 (+0.5/-0) x 137.0 (+0.5/-0)
NB10W	258.0 (+0.5/-0) x 200.0 (+0.5/-0)

Applicable panel thickness: 1.6 to 4.8 mm

Related Manuals

Cat. No	Model	Name
V106	NB-Designer	NB Series NB-Designer Operation Manual
V107	NB3Q, NB5Q, NB7W, NB10W	NB Series Setup Manual
V108	NB3Q, NB5Q, NB7W, NB10W	NB Series Host Connection Manual
V109	NB3Q, NB5Q, NB7W, NB10W	NB Series Startup Guide

รูปที่ ก-13 แสดงคุณสมบัติของHMI Omron NB5QTW00Bหน้าที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:15 น. D:\INS\ป... \SCARAFINISH DataFile.m 1 of 4

```
% Simscape(TM) Multibody(TM) version: 7.4

% This is a model data file derived from a Simscape Multibody Import XML file using
the smimport function.
% The data in this file sets the block parameter values in an imported Simscape
Multibody model.
% For more information on this file, see the smimport function help page in the
Simscape Multibody documentation.
% You can modify numerical values, but avoid any other changes to this file.
% Do not add code to this file. Do not edit the physical units shown in comments.

%%VariableName:smiData

%===== RigidTransform =====%

%Initialize the RigidTransform structure array by filling in null values.
smiData.RigidTransform(9).translation = [0.0 0.0 0.0];
smiData.RigidTransform(9).angle = 0.0;
smiData.RigidTransform(9).axis = [0.0 0.0 0.0];
smiData.RigidTransform(9).ID = '';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(1).translation = [-77.500000000000014 0 -10.000000000000009];
% mm
smiData.RigidTransform(1).angle = 0; % rad
smiData.RigidTransform(1).axis = [0 0 0];
smiData.RigidTransform(1).ID = 'B[Part armm-2--:arm 1 update-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(2).translation = [200.00000000000011 6.4392935428259079e-15
4.999999999999964]; % mm
smiData.RigidTransform(2).angle = 3.1415926535897927; % rad
smiData.RigidTransform(2).axis = [-1 -2.606250810036893e-32 7.8383362686922011e-17];
smiData.RigidTransform(2).ID = 'F[Part armm-2--:arm 1 update-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(3).translation = [64.99999999999986 -42.50000000000036
9.99999999999947]; % mm
smiData.RigidTransform(3).angle = 0; % rad
smiData.RigidTransform(3).axis = [0 0 0];
smiData.RigidTransform(3).ID = 'B[model new-1--:arm 1 update-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(4).translation = [8.1989970368567811e-14 -42.49999999999929
-102.28469758650316]; % mm
smiData.RigidTransform(4).angle = 3.9849038017436856e-16; % rad
smiData.RigidTransform(4).axis = [-0.8862749303750177 -0.46305594690557628
0.0097947855585369791];
smiData.RigidTransform(4).ID = 'F[model new-1--:arm 1 update-1]';
```

รูปที่ ข-1 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:15 น. D:\INS\ป... \SCARAFINISH DataFile.m 2 of 4

```

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(5).translation = [77.49999999999957 0 30]; % mm
smiData.RigidTransform(5).angle = 0; % rad
smiData.RigidTransform(5).axis = [0 0 0];
smiData.RigidTransform(5).ID = 'B[Part armm-2--:GRP3-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(6).translation = [1.7054379224114983e-13 3.1569464429991513e-15
-9.9999999999999893]; % mm
smiData.RigidTransform(6).angle = 1.188027207435054e-16; % rad
smiData.RigidTransform(6).axis = [-0.99825704317902098 -0.059015893990333704
3.4995142346445853e-18];
smiData.RigidTransform(6).ID = 'F[Part armm-2--:GRP3-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(7).translation = [0 0 60.000000000000007]; % mm
smiData.RigidTransform(7).angle = 0; % rad
smiData.RigidTransform(7).axis = [0 0 0];
smiData.RigidTransform(7).ID = 'B[scara part platel-1--:model new-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(8).translation = [4.0924152161301874e-16 -7.8603689502771822e-16
1.1936485775231347]; % mm
smiData.RigidTransform(8).angle = 7.4242114836850522e-16; % rad
smiData.RigidTransform(8).axis = [0.88698466631402351 0.46179887583644075
1.5205101456367772e-16];
smiData.RigidTransform(8).ID = 'F[scara part platel-1--:model new-1]';

%Translation Method - Cartesian
%Rotation Method - Arbitrary Axis
smiData.RigidTransform(9).translation = [0 0 0]; % mm
smiData.RigidTransform(9).angle = 0; % rad
smiData.RigidTransform(9).axis = [0 0 0];
smiData.RigidTransform(9).ID = 'RootGround[scara part platel-1]';

%===== Solid =====%
%Center of Mass (CoM) %Moments of Inertia (MoI) %Product of Inertia (PoI)

%Initialize the Solid structure array by filling in null values.
smiData.Solid(5).mass = 0.0;
smiData.Solid(5).CoM = [0.0 0.0 0.0];
smiData.Solid(5).MoI = [0.0 0.0 0.0];
smiData.Solid(5).PoI = [0.0 0.0 0.0];
smiData.Solid(5).color = [0.0 0.0 0.0];
smiData.Solid(5).opacity = 0.0;
smiData.Solid(5).ID = '';

%Inertia Type - Custom

```

รูปที่ ข-2 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:15 น. D:\INS\ป... \SCARAFINISH DataFile.m 3 of 4

```

%Visual Properties - Simple
smiData.Solid(1).mass = 0.23245920110780138; % kg
smiData.Solid(1).CoM = [79.070514375386978 0 20.336344557641141]; % mm
smiData.Solid(1).MoI = [100.41430987557122 1215.5596359498682 1260.0793683003251]; % kg*mm^2
smiData.Solid(1).PoI = [0 -9.4550395828573954 0]; % kg*mm^2
smiData.Solid(1).color = [1 0 1];
smiData.Solid(1).opacity = 1;
smiData.Solid(1).ID = 'arm 1 update*:Default';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(2).mass = 0.36767477042468105; % kg
smiData.Solid(2).CoM = [0 0 13.010457473975]; % mm
smiData.Solid(2).MoI = [542.98660039441972 935.02488164441854 1303.0845636743786]; % kg*mm^2
smiData.Solid(2).PoI = [0 0 0]; % kg*mm^2
smiData.Solid(2).color = [0.792156862745098 0.81960784313725488 0.9333333333333335];
smiData.Solid(2).opacity = 1;
smiData.Solid(2).ID = 'scara part platel*:Default';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(3).mass = 0.07806139244147782; % kg
smiData.Solid(3).CoM = [9.7078854118089168 0 -4.3842491647076471]; % mm
smiData.Solid(3).MoI = [12.553223579355478 234.06105239888066 240.71409299613347]; % kg*mm^2
smiData.Solid(3).PoI = [0 -3.2585207001721535 0]; % kg*mm^2
smiData.Solid(3).color = [0.99607843137254903 0.87058823529411766 0.76078431372549016];
smiData.Solid(3).opacity = 1;
smiData.Solid(3).ID = 'Part armm*:Default';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(4).mass = 0.95391429178004106; % kg
smiData.Solid(4).CoM = [16.363024795836949 -0.029052429940401782 111.77402118279312]; % mm
smiData.Solid(4).MoI = [12182.738485580354 11209.209053415281 4882.9772897481062]; % kg*mm^2
smiData.Solid(4).PoI = [-1.6739029959879579 -1660.2421249324589 -1.1430716865698562]; % kg*mm^2
smiData.Solid(4).color = [1 0.75294117647058822 0.75294117647058822];
smiData.Solid(4).opacity = 1;
smiData.Solid(4).ID = 'model new*:Default';

%Inertia Type - Custom
%Visual Properties - Simple
smiData.Solid(5).mass = 0.032200000543235234; % kg
smiData.Solid(5).CoM = [0 0 -4.999999999999999]; % mm
smiData.Solid(5).MoI = [47.948404079559381 1.9827409525383157 49.399504904622852]; % kg*mm^2
smiData.Solid(5).PoI = [0 0 0.89344651098614125]; % kg*mm^2

```

รูปที่ ข-3 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:15 น. D:\INS\ป้...\SCARAFINISH DataFile.m 4 of 4

```

smiData.Solid(5).color = [0.792156862745098 0.81960784313725488
0.9333333333333335];
smiData.Solid(5).opacity = 1;
smiData.Solid(5).ID = 'GRP3*:Default';

%===== Joint =====%
%X Revolute Primitive (Rx) %Y Revolute Primitive (Ry) %Z Revolute Primitive (Rz)
%X Prismatic Primitive (Px) %Y Prismatic Primitive (Py) %Z Prismatic Primitive (Pz)
%Spherical Primitive (S)
%Constant Velocity Primitive (CV) %Lead Screw Primitive (LS)
%Position Target (Pos)

%Initialize the CylindricalJoint structure array by filling in null values.
smiData.CylindricalJoint(1).Rz.Pos = 0,0;
smiData.CylindricalJoint(1).Pz.Pos = 0,0;
smiData.CylindricalJoint(1).ID = '';

smiData.CylindricalJoint(1).Rz.Pos = 0.21599626479527884; % deg
smiData.CylindricalJoint(1).Pz.Pos = 0; % mm
smiData.CylindricalJoint(1).ID = '[scara part platel-1:-:model new-1]';

%Initialize the PrismaticJoint structure array by filling in null values.
smiData.PrismaticJoint(1).Pz.Pos = 0,0;
smiData.PrismaticJoint(1).ID = '';

smiData.PrismaticJoint(1).Pz.Pos = 0; % m
smiData.PrismaticJoint(1).ID = '[model new-1:-:arm 1 update-1]';

%Initialize the RevoluteJoint structure array by filling in null values.
smiData.RevoluteJoint(2).Rz.Pos = 0,0;
smiData.RevoluteJoint(2).ID = '';

smiData.RevoluteJoint(1).Rz.Pos = -0.28077843041121475; % deg
smiData.RevoluteJoint(1).ID = '[Part armm-2:-:arm 1 update-1]';

smiData.RevoluteJoint(2).Rz.Pos = 178.30412923775256; % deg
smiData.RevoluteJoint(2).ID = '[Part armm-2:-:GRP3-1]';

```

รูปที่ ข-4 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:16 น. D:\INS\ปี2\เทอม2\PRO...\DisplayBIG.m 1 of 7

```

function varargout = DisplayBIG(varargin)
% DISPLAYBIG MATLAB code for DisplayBIG.fig
%   DISPLAYBIG, by itself, creates a new DISPLAYBIG or raises the existing
%   singleton*.
%
%   H = DISPLAYBIG returns the handle to a new DISPLAYBIG or the handle to
%   the existing singleton*.
%
%   DISPLAYBIG('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in DISPLAYBIG.M with the given input arguments.
%
%   DISPLAYBIG('Property','Value',...) creates a new DISPLAYBIG or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before DisplayBIG_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to DisplayBIG_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
%
% Edit the above text to modify the response to help DisplayBIG
%
% Last Modified by GUIDE v2.5 29-Mar-2023 23:33:06
%
% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @DisplayBIG_OpeningFcn, ...
                  'gui_OutputFcn',  @DisplayBIG_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before DisplayBIG is made visible.
function DisplayBIG_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to DisplayBIG (see VARARGIN)

```

รูปที่ 20 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:16 น. D:\INS\ปี2\เทอม2\PRO...\DisplayBIG.m 2 of 7

```
% Choose default command line output for DisplayBIG
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes DisplayBIG wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = DisplayBIG_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on button press in inverse.
function inverse_Callback(hObject, eventdata, handles)
modelName = 'updateSCARA';
global var;
x=get(handles.x,'string');
x = str2num(x);
if ((x <= -350)&(x >=350))
    msgbox('Please Select PX Value In Range -350-->350');
end
y=get(handles.y,'string');
y = str2num(y);
if ((y <= -350)&(y >=350))
    msgbox('Please Select PY Value In Range -350-->350');
end
z=get(handles.z,'string');
z = str2num(z);
if ((z <= 20)&(z >= 180))
    msgbox('Please Select PZ Value In Range 0-->180');
end
xi = x/1000;
yi = y/1000;
zi = z/1000;
d0 = 0.12; d4 = -0.040; d5 = -0.090;
L2 = 0.20; L3 = 0.195;
C2 = ((xi)^2 + (yi)^2 - (L2)^2 - (L3)^2)/(2*L2*L3);
C2f = round(C2*1000)/1000;
S2 = 1*sqrt(1-(C2f)^2);
if (S2 <= -1)
    msgbox('Robot Crash Close Program to Restart');at
end
q3 = atan2d(S2,C2f)*(-1);
q3ln = q3*(-1)
q1 = atan2d((yi), (xi))-atan2d(L3*S2,L2+(L3*C2f));
```

รูปที่ 21 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:16 น. D:\INS\ปี2\เทอม2\PRO...\DisplayBIG.m 3 of 7

```

q2 = (zi + d4 - d5 - d0)*1000 ;
q4 = 0;
guidata(hObject, handles);
set(handles.q1,'string',q1);
set_param([modelName '/Slider Gain'],'Gain',num2str(q1))
set(handles.q2,'string',q2);
set_param([modelName '/Slider Gain1'],'Gain',num2str(q2))
set(handles.q3,'string',q3ln);
set_param([modelName '/Slider Gain2'],'Gain',num2str(q3ln))
set(handles.q4,'string',q4);
set_param([modelName '/Slider Gain3'],'Gain',num2str(q4))
% hObject handle to inverse (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

function x_Callback(hObject, eventdata, handles)
% hObject handle to x (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of x as text
% str2double(get(hObject,'String')) returns contents of x as a double

% --- Executes during object creation, after setting all properties.
function x_CreateFcn(hObject, eventdata, handles)
% hObject handle to x (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function y_Callback(hObject, eventdata, handles)
% hObject handle to y (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of y as text
% str2double(get(hObject,'String')) returns contents of y as a double

% --- Executes during object creation, after setting all properties.
function y_CreateFcn(hObject, eventdata, handles)
% hObject handle to y (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB

```

รูปที่ 22 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:16 น. D:\INS\ปี2\เทอม2\PRO...\DisplayBIG.m 4 of 7

```
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function z_Callback(hObject, eventdata, handles)
% hObject    handle to z (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of z as text
%       str2double(get(hObject,'String')) returns contents of z as a double

% --- Executes during object creation, after setting all properties.
function z_CreateFcn(hObject, eventdata, handles)
% hObject    handle to z (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function q1_Callback(hObject, eventdata, handles)
% hObject    handle to q1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of q1 as text
%       str2double(get(hObject,'String')) returns contents of q1 as a double

% --- Executes during object creation, after setting all properties.
function q1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to q1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUiControlBackgroundColor'))
```

รูปที่ 23 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:16 น. D:\INS\ปี2\เทอม2\PRO...\DisplayBIG.m 5 of 7

```

        set(hObject,'BackgroundColor','white');
    end

function q2_Callback(hObject, eventdata, handles)
% hObject    handle to q2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of q2 as text
%        str2double(get(hObject,'String')) returns contents of q2 as a double

% --- Executes during object creation, after setting all properties.
function q2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to q2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function q3_Callback(hObject, eventdata, handles)
% hObject    handle to q3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of q3 as text
%        str2double(get(hObject,'String')) returns contents of q3 as a double

% --- Executes during object creation, after setting all properties.
function q3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to q3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function q4_Callback(hObject, eventdata, handles)

```

รูปที่ 24 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

18/11/2566 21:16 น. D:\INS\ปี2\เทอม2\PRO...\DisplayBIG.m 6 of 7

```

% hObject    handle to q4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of q4 as text
%        str2double(get(hObject,'String')) returns contents of q4 as a double

% --- Executes during object creation, after setting all properties.
function q4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to q4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(
(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
modelName = 'updateSCARA';
global var;
q1 = 0;
set(handles.q1,'string',num2str(q1));
set_param([modelName '/Slider Gain'],'Gain',num2str(q1))
q2 = 0;
set(handles.q2,'string',num2str(q2));
set_param([modelName '/Slider Gain1'],'Gain',num2str(q2))
q3 = 0;
set(handles.q3,'string',num2str(q3));
set_param([modelName '/Slider Gain2'],'Gain',num2str(q3))
q4 = 0;
set(handles.q4,'string',num2str(q4));
set_param([modelName '/Slider Gain3'],'Gain',num2str(q4))
set(handles.x,'string',num2str(350));
set(handles.y,'string',num2str(0));
set(handles.z,'string',num2str(100));

% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
close;
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

รูปที่ 25 โปรแกรมแบบจำลองทางคณิตศาสตร์หน้าที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Variable Type	Name	Data Type	Retained	AT	Initial Value	Comment
Inputs	EN	BOOL	No		FALSE	Controls execution of the Function Block.
Inputs	X1	REAL	No		0.0	
Inputs	Y1	REAL	No		0.0	
Inputs	Z1	REAL	No		0.0	
Inputs	EE1	REAL	No		0.0	
Inputs	X2	REAL	No		0.0	
Inputs	Y2	REAL	No		0.0	
Inputs	Z2	REAL	No		0.0	
Inputs	EE2	REAL	No		0.0	
Inputs	X3	REAL	No		0.0	
Inputs	Y3	REAL	No		0.0	
Inputs	Z3	REAL	No		0.0	
Inputs	EE3	REAL	No		0.0	
Inputs	X4	REAL	No		0.0	
Inputs	Y4	REAL	No		0.0	
Inputs	Z4	REAL	No		0.0	
Inputs	EE4	REAL	No		0.0	
Inputs	X5	REAL	No		0.0	
Inputs	Y5	REAL	No		0.0	
Inputs	Z5	REAL	No		0.0	
Inputs	EE5	REAL	No		0.0	
Outputs	ENO	BOOL	No		FALSE	Indicates successful execution of the Function Block.
Outputs	BASEPulse_1	REAL	No		0.0	
Outputs	BASEPulse_2	REAL	No		0.0	
Outputs	BASEPulse_3	REAL	No		0.0	
Outputs	BASEPulse_4	REAL	No		0.0	
Outputs	BASEPulse_5	REAL	No		0.0	
Outputs	LIFTPulse_1	REAL	No		0.0	
Outputs	LIFTPulse_2	REAL	No		0.0	
Outputs	LIFTPulse_3	REAL	No		0.0	
Outputs	LIFTPulse_4	REAL	No		0.0	
Outputs	LIFTPulse_5	REAL	No		0.0	
Outputs	SHOULDERPulse_1	REAL	No		0.0	
Outputs	SHOULDERPulse_2	REAL	No		0.0	
Outputs	SHOULDERPulse_3	REAL	No		0.0	
Outputs	SHOULDERPulse_4	REAL	No		0.0	
Outputs	SHOULDERPulse_5	REAL	No		0.0	
Outputs	THETA_ER	BOOL	No		FALSE	
Outputs	THETA_ER_2	BOOL	No		FALSE	
Outputs	THETA_ER_3	BOOL	No		FALSE	
Outputs	THETA_ER_4	BOOL	No		FALSE	
Outputs	THETA_ER_5	BOOL	No		FALSE	
Outputs	POS_ER_1	BOOL	No		FALSE	
Outputs	POS_ER_2	BOOL	No		FALSE	
Outputs	POS_ER_3	BOOL	No		FALSE	
Outputs	POS_ER_4	BOOL	No		FALSE	
Outputs	POS_ER_5	BOOL	No		FALSE	
Internals	L2	REAL	No		200.0	
Internals	L3	REAL	No		195.0	
Internals	dd0	REAL	No		120.0	
Internals	dd4	REAL	No		-40.0	
Internals	dd5	REAL	No		-90.0	
Internals	X1min	REAL	No		-350.0	
Internals	X1max	REAL	No		350.0	
Internals	Y1min	REAL	No		-350.0	

รูปที่ ค-1 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่หน้าที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Internals	Y1max	REAL	No	350.0	
Internals	Z1min	REAL	No	0.0	
Internals	Z1max	REAL	No	179.0	
Internals	BaseThetaMax	REAL	No	180.0	
Internals	BaseThetaMin	REAL	No	-180.0	
Internals	LiftMax	REAL	No	80.0	
Internals	LiftMin	REAL	No	-80.0	
Internals	ShoulderThetaMax	REAL	No	145.0	
Internals	ShoulderThetaMin	REAL	No	-145.0	
Internals	EEMax	REAL	No	180.0	
Internals	EEMin	REAL	No	-180.0	
Internals	Cos2_1	REAL	No	0.0	
Internals	Cos2_2	REAL	No	0.0	
Internals	Cos2_3	REAL	No	0.0	
Internals	Cos2_4	REAL	No	0.0	
Internals	Cos2_5	REAL	No	0.0	
Internals	S2_1	REAL	No	0.0	
Internals	S2_2	REAL	No	0.0	
Internals	S2_3	REAL	No	0.0	
Internals	S2_4	REAL	No	0.0	
Internals	S2_5	REAL	No	0.0	
Internals	Beta_1	REAL	No	0.0	
Internals	Beta_2	REAL	No	0.0	
Internals	Beta_3	REAL	No	0.0	
Internals	Beta_4	REAL	No	0.0	
Internals	Beta_5	REAL	No	0.0	
Internals	Radius1	REAL	No	0.0	
Internals	Radius2	REAL	No	0.0	
Internals	Radius3	REAL	No	0.0	
Internals	Radius4	REAL	No	0.0	
Internals	Radius5	REAL	No	0.0	
Internals	BASE_1	REAL	No	0.0	
Internals	BASE_2	REAL	No	0.0	
Internals	BASE_3	REAL	No	0.0	
Internals	BASE_4	REAL	No	0.0	
Internals	BASE_5	REAL	No	0.0	
Internals	LIFT_1	REAL	No	0.0	
Internals	LIFT_2	REAL	No	0.0	
Internals	LIFT_3	REAL	No	0.0	
Internals	LIFT_4	REAL	No	0.0	
Internals	LIFT_5	REAL	No	0.0	
Internals	SHOULDER_1	REAL	No	0.0	
Internals	SHOULDER_2	REAL	No	0.0	
Internals	SHOULDER_3	REAL	No	0.0	
Internals	SHOULDER_4	REAL	No	0.0	
Internals	SHOULDER_5	REAL	No	0.0	

[Function Block Name : Cal_Pulse1_5]

```

Radius1 := sqrt((X1*X1) + (Y1*Y1) + (Z1*Z1));
Radius2 := sqrt((X2*X2) + (Y2*Y2) + (Z2*Z2));
Radius3 := sqrt((X3*X3) + (Y3*Y3) + (Z3*Z3));
Radius4 := sqrt((X4*X4) + (Y4*Y4) + (Z4*Z4));
Radius5 := sqrt((X5*X5) + (Y5*Y5) + (Z5*Z5));

```

```

IF ((Radius1 > 405.0 OR Radius1 < 150.0) OR (X1 > X1max OR X1 < X1min) OR (Y1 > Y1max OR Y1 < Y1min) OR (Z1 > Z1max OR Z1 < Z1min))

```

รูปที่ 26 แสดงโปรแกรม PLC ควบคุมแขนกลหน้า ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

END_IF;

IF(((BASE_1) > BaseThetaMax OR (BASE_1) < BaseThetaMin) OR ((LIFT_1) > LiftMax OR (LIFT_1) < LiftMin) OR
((SHOULDER_1) > ShoulderThetaMax OR (SHOULDER_1) < ShoulderThetaMin) OR (EE1 > EEMax OR EE1 < EEMin))
THEN THETA_ER := TRUE;
ELSE
  THETA_ER := FALSE;
  BASEPulse_1 := BASE_1*90.0;
  LIFTPulse_1 := LIFT_1*200.0;
  SHOULDERPulse_1 := SHOULDER_1*52.941;
END_IF;

END_IF;
IF(((BASE_2) > BaseThetaMax OR (BASE_2) < BaseThetaMin) OR ((LIFT_2) > LiftMax OR (LIFT_2) < LiftMin) OR
((SHOULDER_2) > ShoulderThetaMax OR (SHOULDER_2) < ShoulderThetaMin) OR (EE2 > EEMax OR EE2 < EEMin))
THEN THETA_ER_2 := TRUE;
ELSE THETA_ER_2 := FALSE;
  BASEPulse_2 := BASE_2*90.0;
  LIFTPulse_2 := LIFT_2*200.0;
  SHOULDERPulse_2 := SHOULDER_2*52.941;
END_IF;

IF(((BASE_3) > BaseThetaMax OR (BASE_3) < BaseThetaMin) OR ((LIFT_3) > LiftMax OR (LIFT_3) < LiftMin) OR
((SHOULDER_3) > ShoulderThetaMax OR (SHOULDER_3) < ShoulderThetaMin) OR (EE3 > EEMax OR EE3 < EEMin))
THEN
  THETA_ER_3 := TRUE;
ELSE
  THETA_ER_3 := FALSE;
  BASEPulse_3 := BASE_3*90.0;
  LIFTPulse_3 := LIFT_3*200.0;
  SHOULDERPulse_3 := SHOULDER_3*52.941;
END_IF;

END_IF;
IF(((BASE_4) > BaseThetaMax OR (BASE_4) < BaseThetaMin) OR ((LIFT_4) > LiftMax OR (LIFT_4) < LiftMin) OR
((SHOULDER_4) > ShoulderThetaMax OR (SHOULDER_4) < ShoulderThetaMin) OR (EE4 > EEMax OR EE4 < EEMin))
THEN
  THETA_ER_4 := TRUE;
ELSE
  THETA_ER_4 := FALSE;
  BASEPulse_4 := BASE_4*90.0;
  LIFTPulse_4 := LIFT_4*200.0;
  SHOULDERPulse_4 := SHOULDER_4*52.941;
END_IF;

END_IF;
IF(((BASE_5) > BaseThetaMax OR (BASE_5) < BaseThetaMin) OR ((LIFT_5) > LiftMax OR (LIFT_5) < LiftMin) OR
((SHOULDER_5) > ShoulderThetaMax OR (SHOULDER_5) < ShoulderThetaMin) OR (EE5 > EEMax OR EE5 < EEMin))
THEN
  THETA_ER_5 := TRUE;
ELSE
  THETA_ER_5 := FALSE;
  BASEPulse_5 := BASE_5*90.0;
  LIFTPulse_5 := LIFT_5*200.0;
  SHOULDERPulse_5 := SHOULDER_5*52.941;
END_IF;

END_IF;

```

รูปที่ 27 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IF(X2 > (-1.0) AND Y2 > (-1.0) )
THEN BASE_2 := RAD_TO_DEG ((ATAN(Y2/X2))- (ATAN(Beta_2)));
END_IF;
IF(X3 > (-1.0) AND Y3 > (-1.0) )
THEN BASE_3 := RAD_TO_DEG ((ATAN(Y3/X3))- (ATAN(Beta_3)));
END_IF;
IF(X4 > (-1.0) AND Y4 > (-1.0) )
THEN BASE_4 := RAD_TO_DEG ((ATAN(Y4/X4))- (ATAN(Beta_4)));
END_IF;
IF(X5 > (-1.0) AND Y5 > (-1.0) )
THEN BASE_5 := RAD_TO_DEG ((ATAN(Y5/X5))- (ATAN(Beta_5)));
END_IF;

IF(X1 < (-1.0) AND Y1 > (-1.0) ) THEN
BASE_1 := 90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y1/X1))- (ATAN(Beta_1)))));
END_IF;
IF(X2 < (-1.0) AND Y2 > (-1.0) ) THEN
BASE_2 := 90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y2/X2))- (ATAN(Beta_2)))));
END_IF;
IF(X3 < (-1.0) AND Y3 > (-1.0) ) THEN
BASE_3 := 90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y3/X3))- (ATAN(Beta_3)))));
END_IF;
IF(X4 < (-1.0) AND Y4 > (-1.0) ) THEN
BASE_4 := 90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y4/X4))- (ATAN(Beta_4)))));
END_IF;
IF(X5 < (-1.0) AND Y5 > (-1.0) ) THEN
BASE_5 := 90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y5/X5))- (ATAN(Beta_5)))));
END_IF;

IF(X1 > (-1.0) AND Y1 < (-1.0) ) THEN
BASE_1 := (-1.0) * ABS(RAD_TO_DEG ((ATAN(Y1/X1))- (ATAN(Beta_1)))));
END_IF;
IF(X2 > (-1.0) AND Y2 < (-1.0) ) THEN
BASE_2 := (-1.0) * ABS(RAD_TO_DEG ((ATAN(Y2/X2))- (ATAN(Beta_2)))));
END_IF;
IF(X3 > (-1.0) AND Y3 < (-1.0) ) THEN
BASE_3 := (-1.0) * ABS(RAD_TO_DEG ((ATAN(Y3/X3))- (ATAN(Beta_3)))));
END_IF;
IF(X4 > (-1.0) AND Y4 < (-1.0) ) THEN
BASE_4 := (-1.0) * ABS(RAD_TO_DEG ((ATAN(Y4/X4))- (ATAN(Beta_4)))));
END_IF;
IF(X5 > (-1.0) AND Y5 < (-1.0) ) THEN
BASE_5 := (-1.0) * ABS(RAD_TO_DEG ((ATAN(Y5/X5))- (ATAN(Beta_5)))));
END_IF;

IF(X1 < (-1.0) AND Y1 < (-1.0) ) THEN
BASE_1 := (-1.0) * (90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y1/X1))- (ATAN(Beta_1))))));
END_IF;
IF(X2 < (-1.0) AND Y2 < (-1.0) ) THEN
BASE_2 := (-1.0) * (90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y2/X2))- (ATAN(Beta_2))))));
END_IF;
IF(X3 < (-1.0) AND Y3 < (-1.0) ) THEN
BASE_3 := (-1.0) * (90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y3/X3))- (ATAN(Beta_3))))));
END_IF;
IF(X4 < (-1.0) AND Y4 < (-1.0) ) THEN
BASE_4 := (-1.0) * (90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y4/X4))- (ATAN(Beta_4))))));
END_IF;
IF(X5 < (-1.0) AND Y5 < (-1.0) ) THEN
BASE_5 := (-1.0) * (90.0 + (90.0 - ABS(RAD_TO_DEG ((ATAN(Y5/X5))- (ATAN(Beta_5))))));
END_IF;

```

รูปที่ 28 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

END_IF;

IF(((BASE_1) > BaseThetaMax OR (BASE_1) < BaseThetaMin) OR ((LIFT_1) > LiftMax OR (LIFT_1) < LiftMin) OR
((SHOULDER_1) > ShoulderThetaMax OR (SHOULDER_1) < ShoulderThetaMin) OR (EE1 > EEMax OR EE1 < EEMin))
THEN THETA_ER := TRUE;
ELSE
  THETA_ER := FALSE;
  BASEPulse_1 := BASE_1*90.0;
  LIFTPulse_1 := LIFT_1*200.0;
  SHOULDERPulse_1 := SHOULDER_1*52.941;
END_IF;

END_IF;
IF(((BASE_2) > BaseThetaMax OR (BASE_2) < BaseThetaMin) OR ((LIFT_2) > LiftMax OR (LIFT_2) < LiftMin) OR
((SHOULDER_2) > ShoulderThetaMax OR (SHOULDER_2) < ShoulderThetaMin) OR (EE2 > EEMax OR EE2 < EEMin))
THEN THETA_ER_2 := TRUE;
ELSE THETA_ER_2 := FALSE;
  BASEPulse_2 := BASE_2*90.0;
  LIFTPulse_2 := LIFT_2*200.0;
  SHOULDERPulse_2 := SHOULDER_2*52.941;
END_IF;

IF(((BASE_3) > BaseThetaMax OR (BASE_3) < BaseThetaMin) OR ((LIFT_3) > LiftMax OR (LIFT_3) < LiftMin) OR
((SHOULDER_3) > ShoulderThetaMax OR (SHOULDER_3) < ShoulderThetaMin) OR (EE3 > EEMax OR EE3 < EEMin))
THEN
  THETA_ER_3 := TRUE;
ELSE
  THETA_ER_3 := FALSE;
  BASEPulse_3 := BASE_3*90.0;
  LIFTPulse_3 := LIFT_3*200.0;
  SHOULDERPulse_3 := SHOULDER_3*52.941;
END_IF;

END_IF;
IF(((BASE_4) > BaseThetaMax OR (BASE_4) < BaseThetaMin) OR ((LIFT_4) > LiftMax OR (LIFT_4) < LiftMin) OR
((SHOULDER_4) > ShoulderThetaMax OR (SHOULDER_4) < ShoulderThetaMin) OR (EE4 > EEMax OR EE4 < EEMin))
THEN
  THETA_ER_4 := TRUE;
ELSE
  THETA_ER_4 := FALSE;
  BASEPulse_4 := BASE_4*90.0;
  LIFTPulse_4 := LIFT_4*200.0;
  SHOULDERPulse_4 := SHOULDER_4*52.941;
END_IF;

END_IF;
IF(((BASE_5) > BaseThetaMax OR (BASE_5) < BaseThetaMin) OR ((LIFT_5) > LiftMax OR (LIFT_5) < LiftMin) OR
((SHOULDER_5) > ShoulderThetaMax OR (SHOULDER_5) < ShoulderThetaMin) OR (EE5 > EEMax OR EE5 < EEMin))
THEN
  THETA_ER_5 := TRUE;
ELSE
  THETA_ER_5 := FALSE;
  BASEPulse_5 := BASE_5*90.0;
  LIFTPulse_5 := LIFT_5*200.0;
  SHOULDERPulse_5 := SHOULDER_5*52.941;
END_IF;

END_IF;

```

รูปที่ 29 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Variable Type	Name	Data Type	Retained	AT	Initial Value	Comment
Inputs	EN	BOOL	No		FALSE	Controls execution of the Function Block.
Inputs	SpeedM0	REAL	No		500.0	
Outputs	ENO	BOOL	No		FALSE	Indicates successful execution of the Function Block.
Outputs	SpeedM2_P1	REAL	No		0.0	
Outputs	SpeedM2_P2	REAL	No		0.0	
Outputs	SpeedM2_P3	REAL	No		0.0	
Outputs	SpeedM2_P4	REAL	No		0.0	
Outputs	SpeedM2_P5	REAL	No		0.0	
Outputs	Time2_P1	REAL	No		0.0	
Outputs	Time2_P2	REAL	No		0.0	
Outputs	Time2_P3	REAL	No		0.0	
Outputs	Time2_P4	REAL	No		0.0	
Outputs	Time2_P5	REAL	No		0.0	
Internals	A	REAL	No		0.0	
Internals	B	REAL	No		0.0	
Internals	C	REAL	No		0.0	
Internals	D	REAL	No		0.0	
Internals	E	REAL	No		0.0	
Internals	K	REAL	No		0.0	
Internals	L	REAL	No		0.0	
Internals	M	REAL	No		0.0	
Internals	N	REAL	No		0.0	
Internals	O	REAL	No		0.0	
Internals	BasePulse_1	REAL	No	D1800		
Internals	ShoulderPulse_1	REAL	No	D1820		
Internals	BasePulse_2	REAL	No	D1802		
Internals	ShoulderPulse_2	REAL	No	D1822		
Internals	BasePulse_3	REAL	No	D1804		
Internals	ShoulderPulse_3	REAL	No	D1824		
Internals	BasePulse_4	REAL	No	D1806		
Internals	BasePulse_5	REAL	No	D1808		
Internals	ShoulderPulse_4	REAL	No	D1826		
Internals	ShoulderPulse_5	REAL	No	D1828		
Internals	V	REAL	No		500.00	

```

[Function Block Name : Cal_Speed1_5]

A := ABS(BasePulse_1) ;
B := ABS(BasePulse_2) ;
C := ABS(BasePulse_3) ;
D := ABS(BasePulse_4) ;
E := ABS(BasePulse_5) ;

K := ABS(ShoulderPulse_1) ;
L := ABS(ShoulderPulse_2) ;
M := ABS(ShoulderPulse_3) ;
N := ABS(ShoulderPulse_4) ;
O := ABS(ShoulderPulse_5) ;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(A)/(Time2_P1) ;
Time2_P1 := ABS(A)/(SpeedM0) ;
SpeedM2_P1 := ABS(K)/(Time2_P1) ;
    
```

รูปที่ 30 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(B-A)/(Time2_P2);
Time2_P2 := ABS(B-A)/(SpeedM0);
SpeedM2_P2 := ABS(L-K)/(Time2_P2);
END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(C-B)/(Time2_P3);
Time2_P3 := ABS(C-B)/(SpeedM0);
SpeedM2_P3 := ABS(M-L)/(Time2_P3);
END_IF;

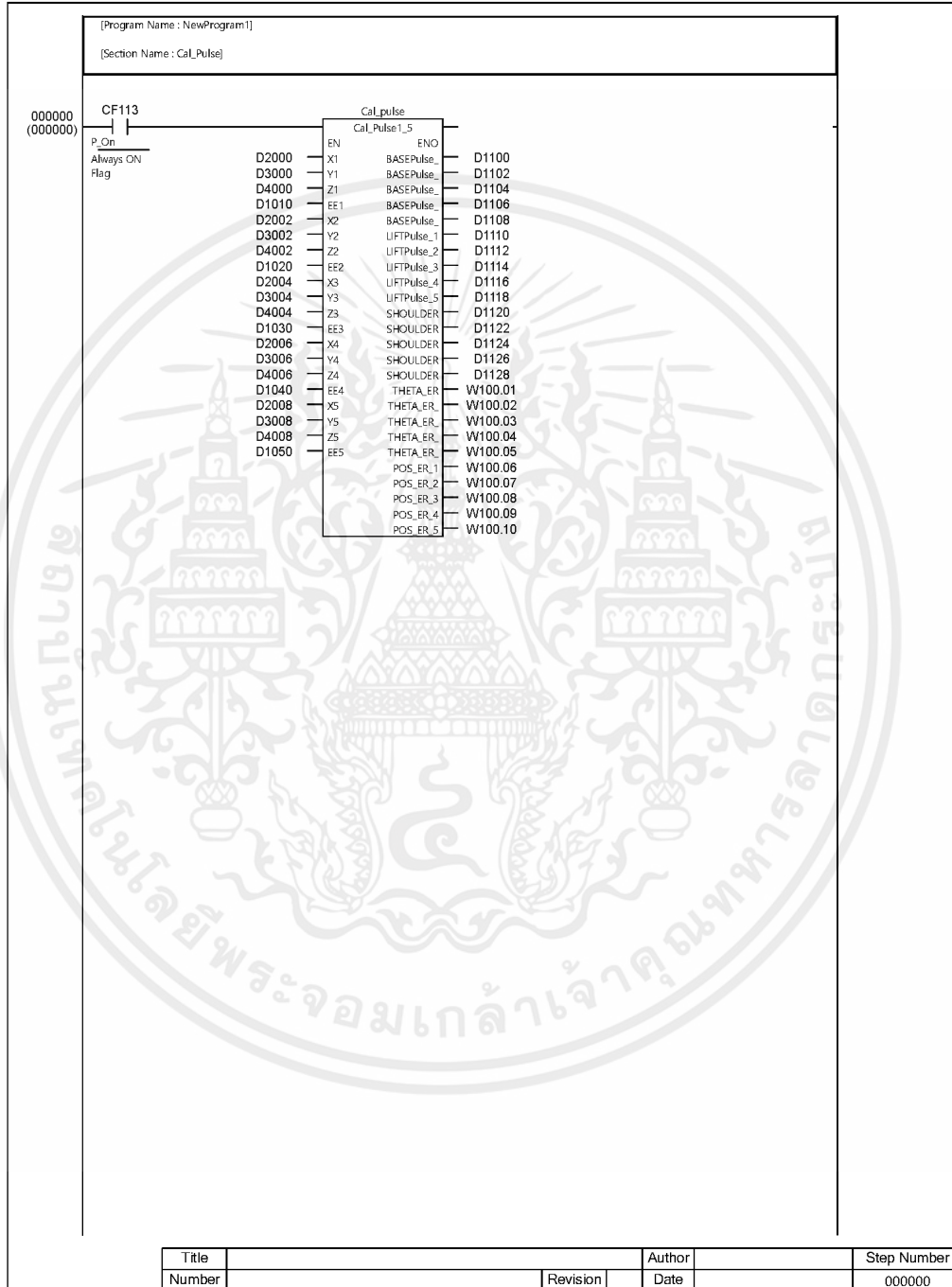
IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(D-C)/(Time2_P4);
Time2_P4 := ABS(D-C)/(SpeedM0);
SpeedM2_P4 := ABS(N-M)/(Time2_P4);
END_IF;

IF(BasePulse_1 > 300.0 OR BasePulse_1 < -300.0) THEN
V := ABS(E-D)/(Time2_P5);
Time2_P5 := ABS(E-D)/(SpeedM0);
SpeedM2_P5 := ABS(O-N)/(Time2_P5);
END_IF;

```

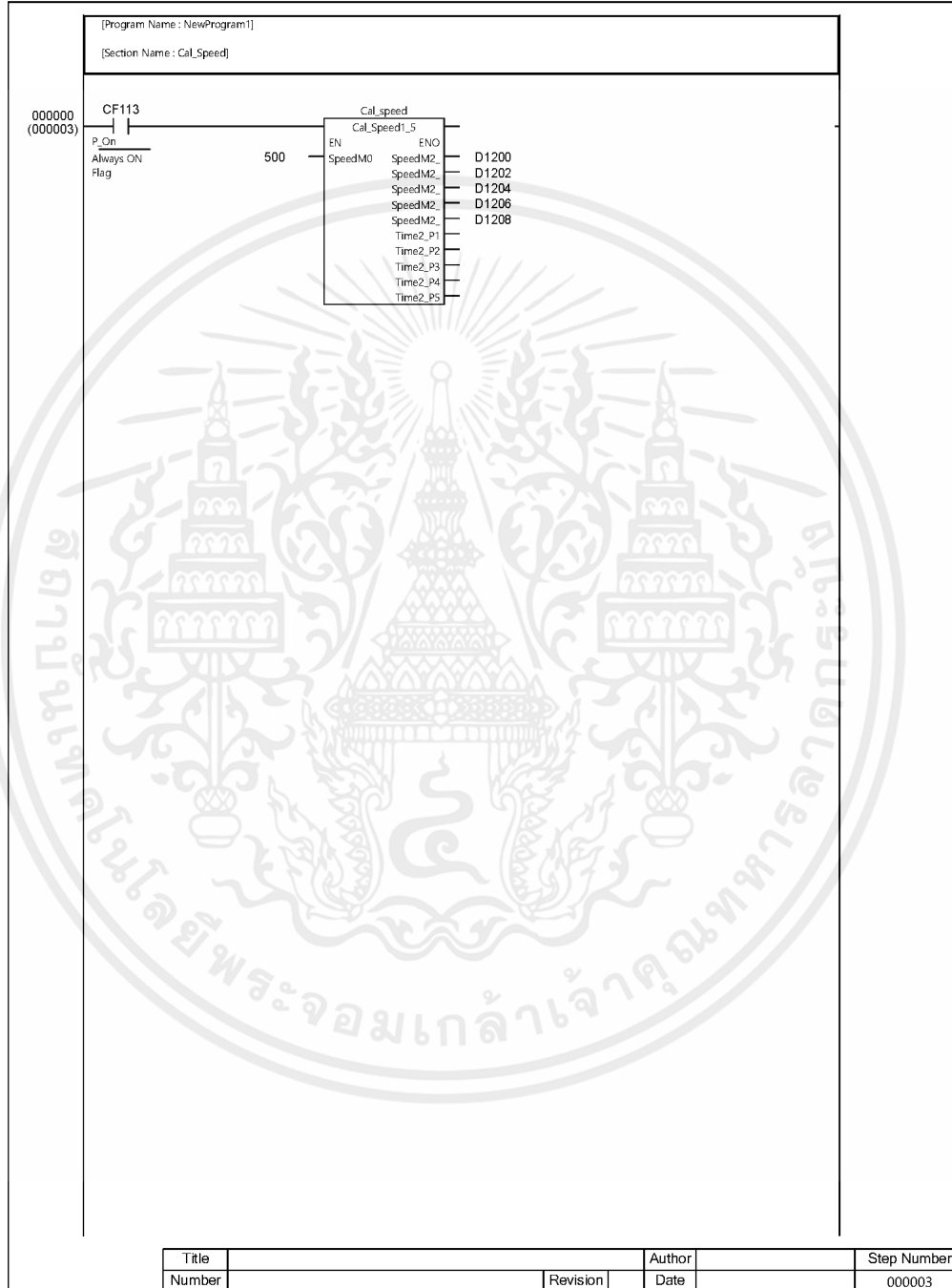
รูปที่ 31 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



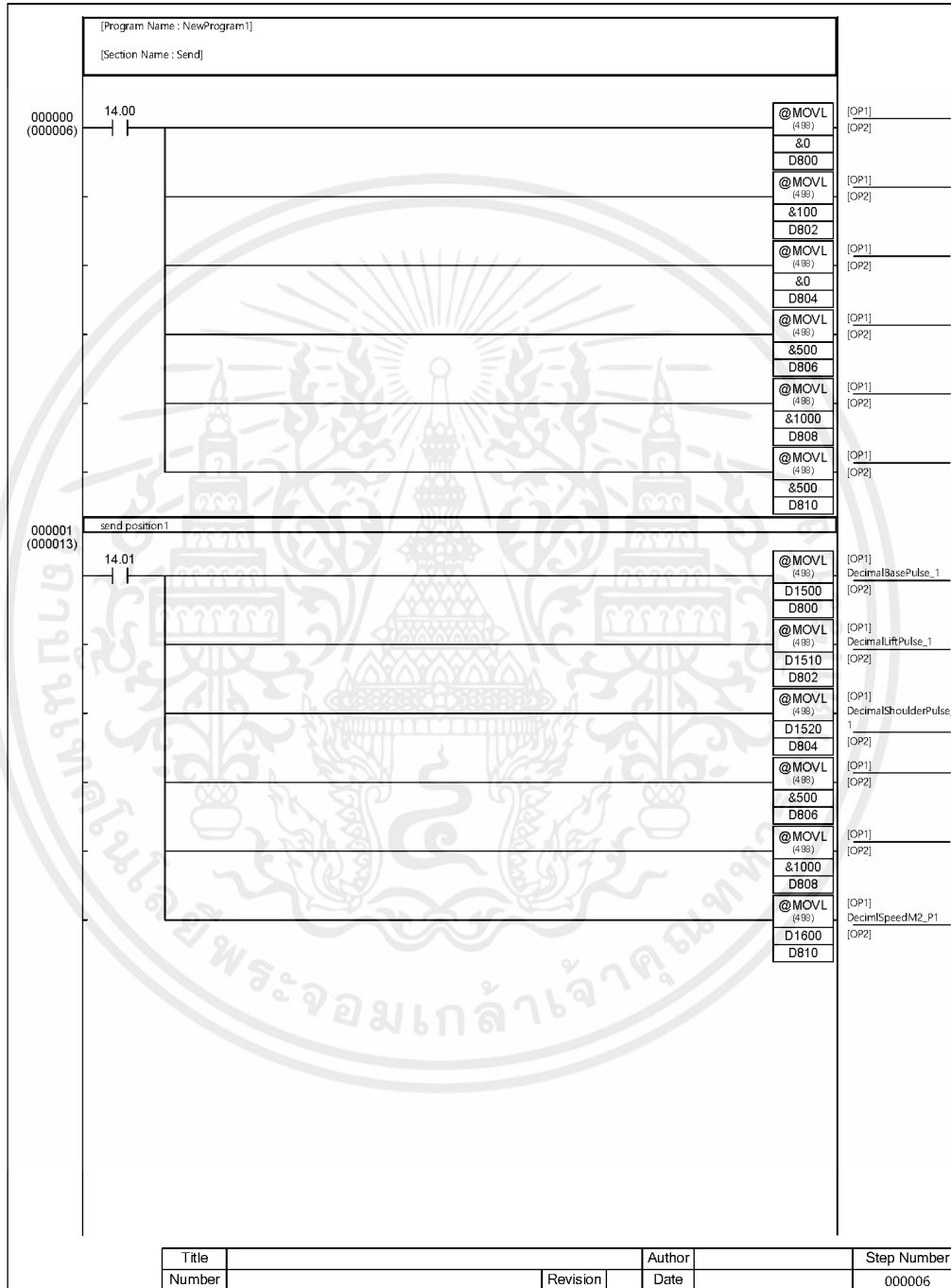
รูปที่ 32 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



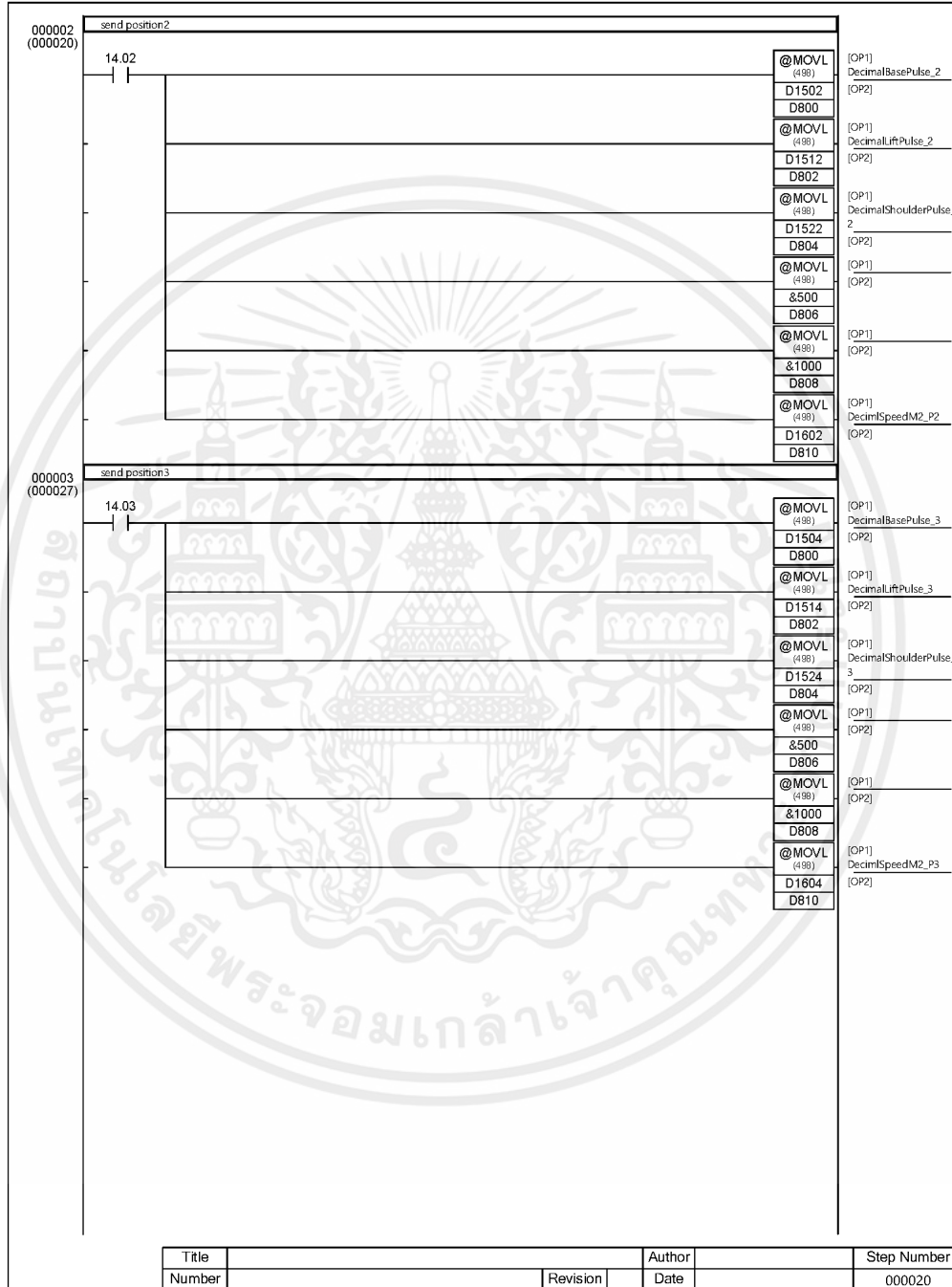
รูปที่ 33 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



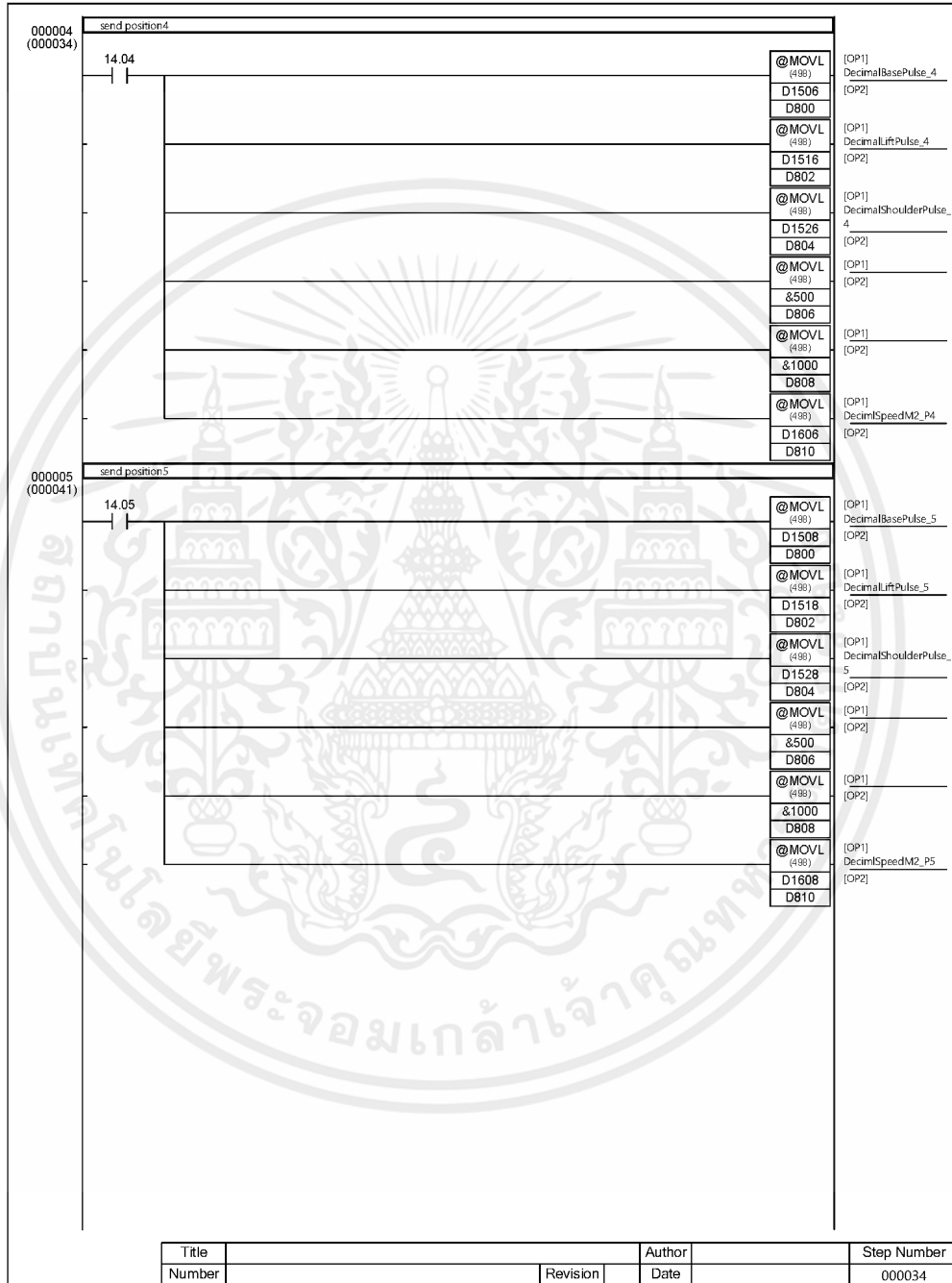
รูปที่ 34 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



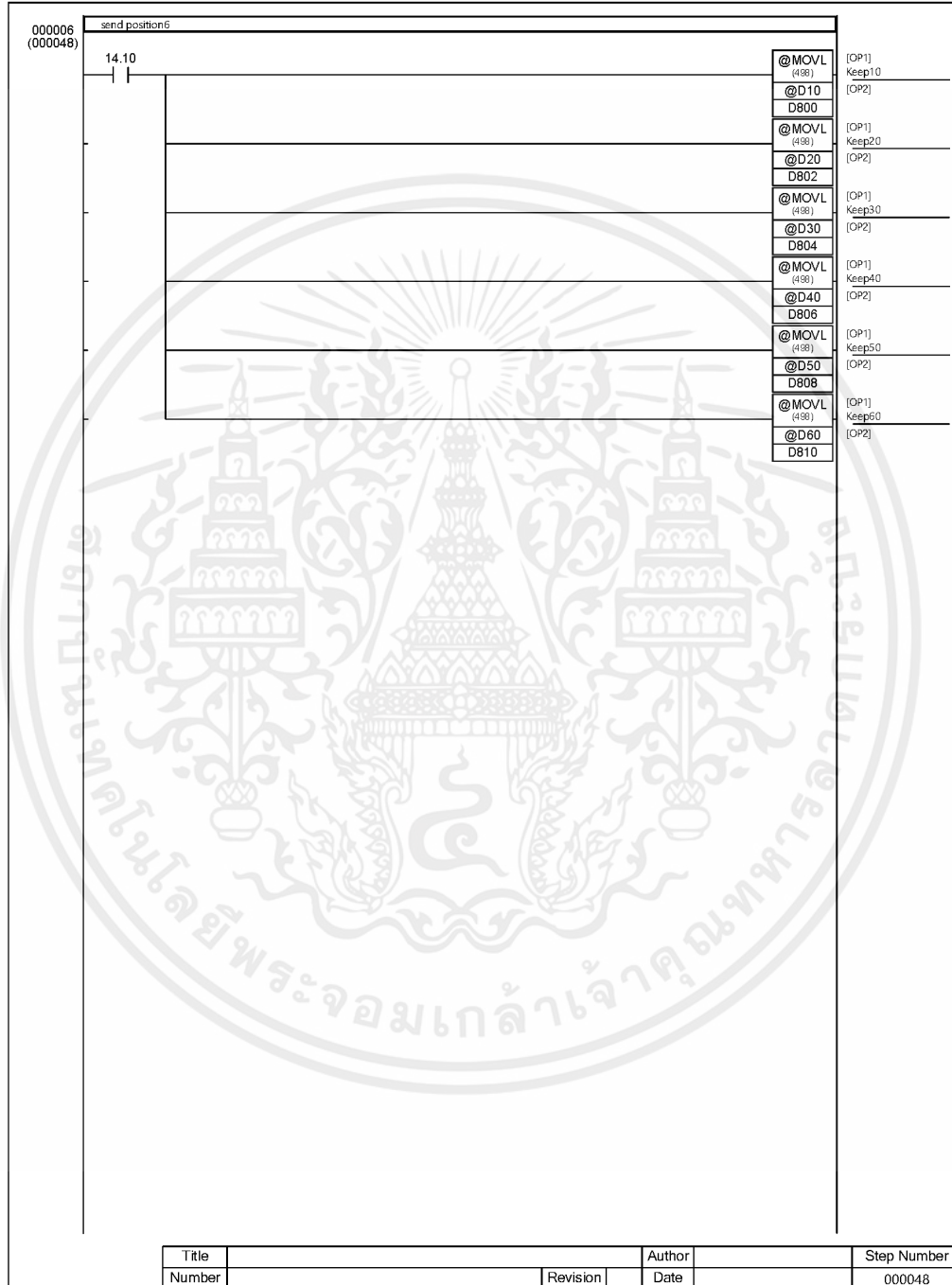
รูปที่ 35 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



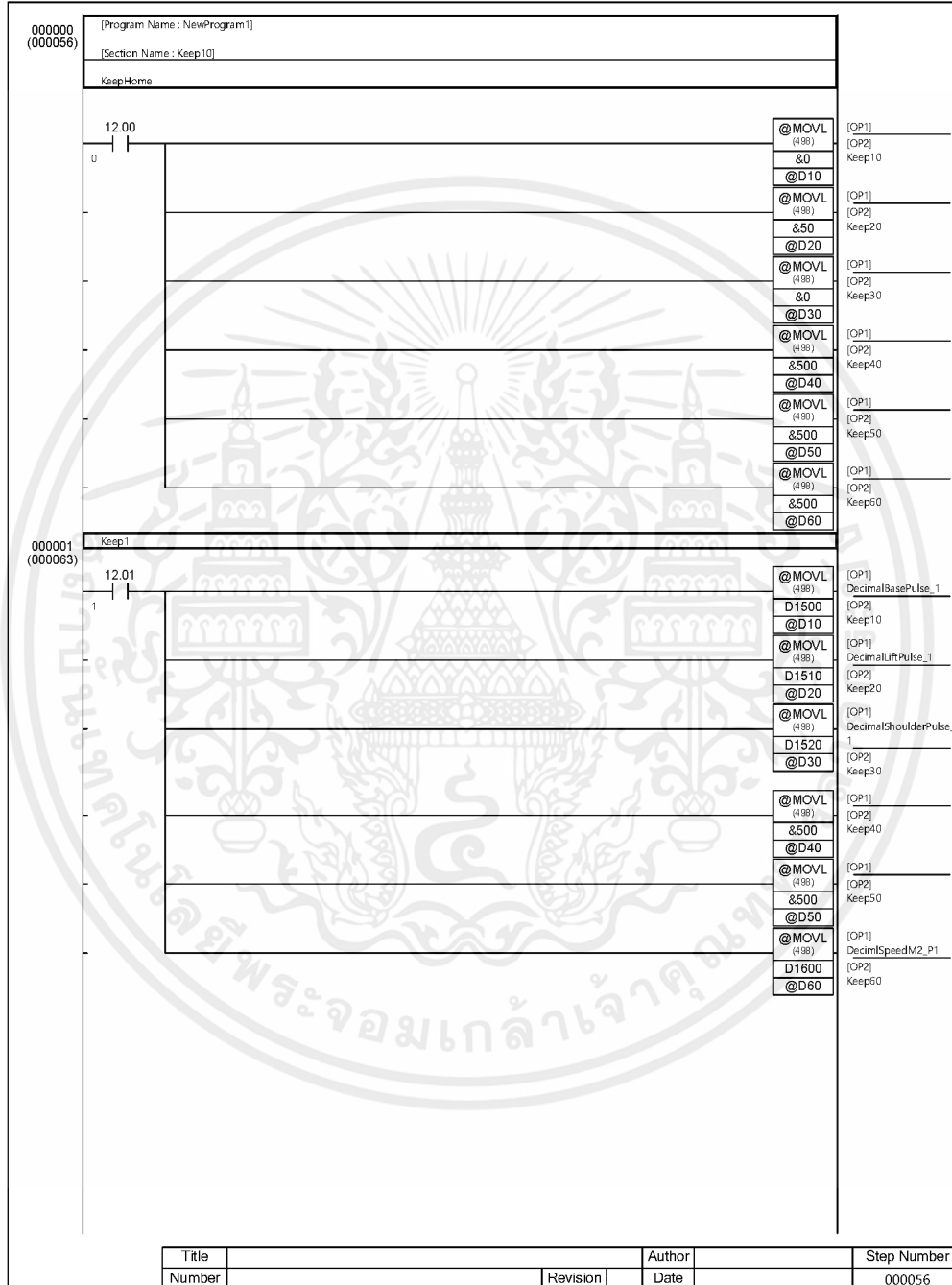
รูปที่ 36 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



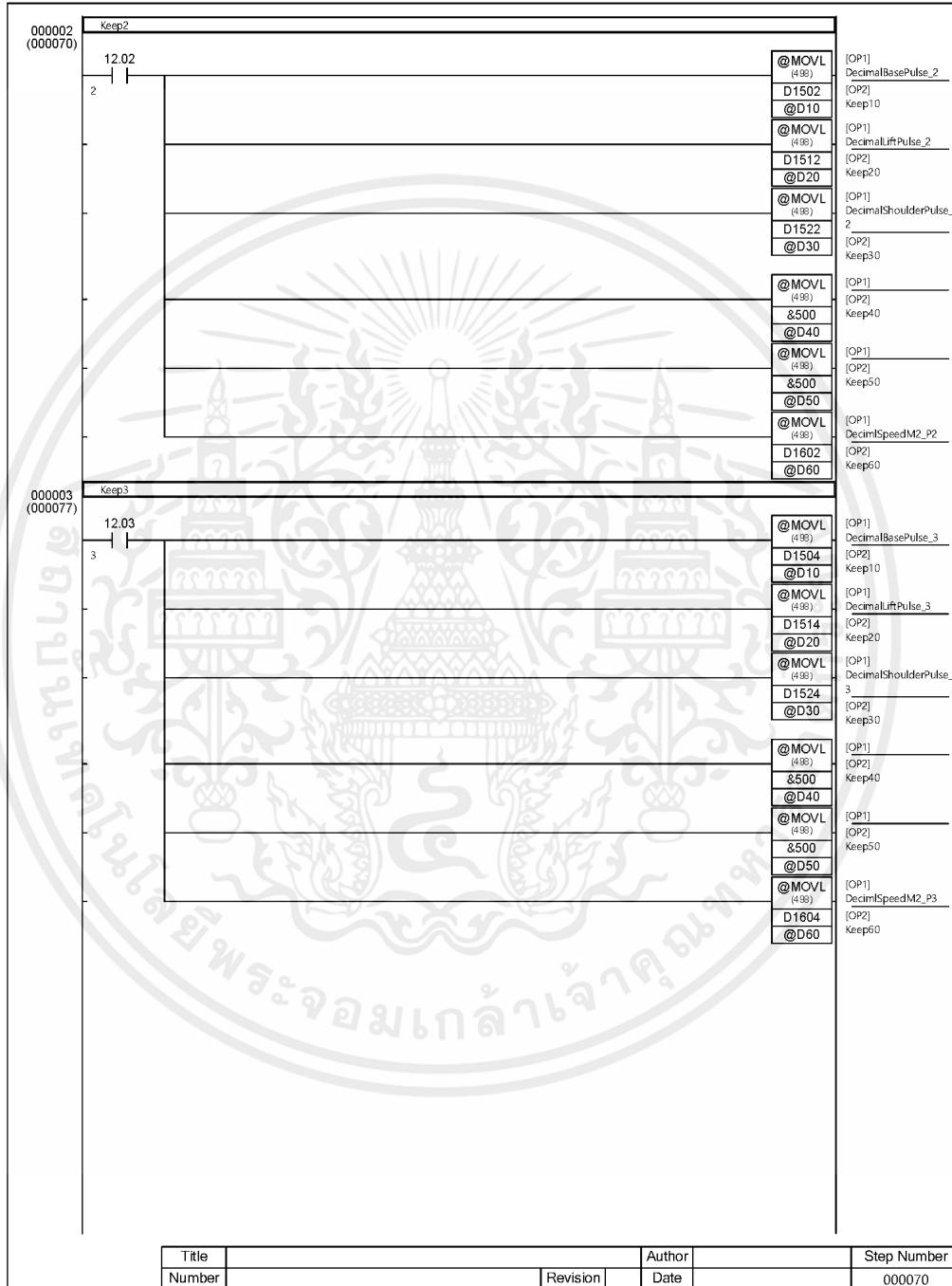
รูปที่ 37 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



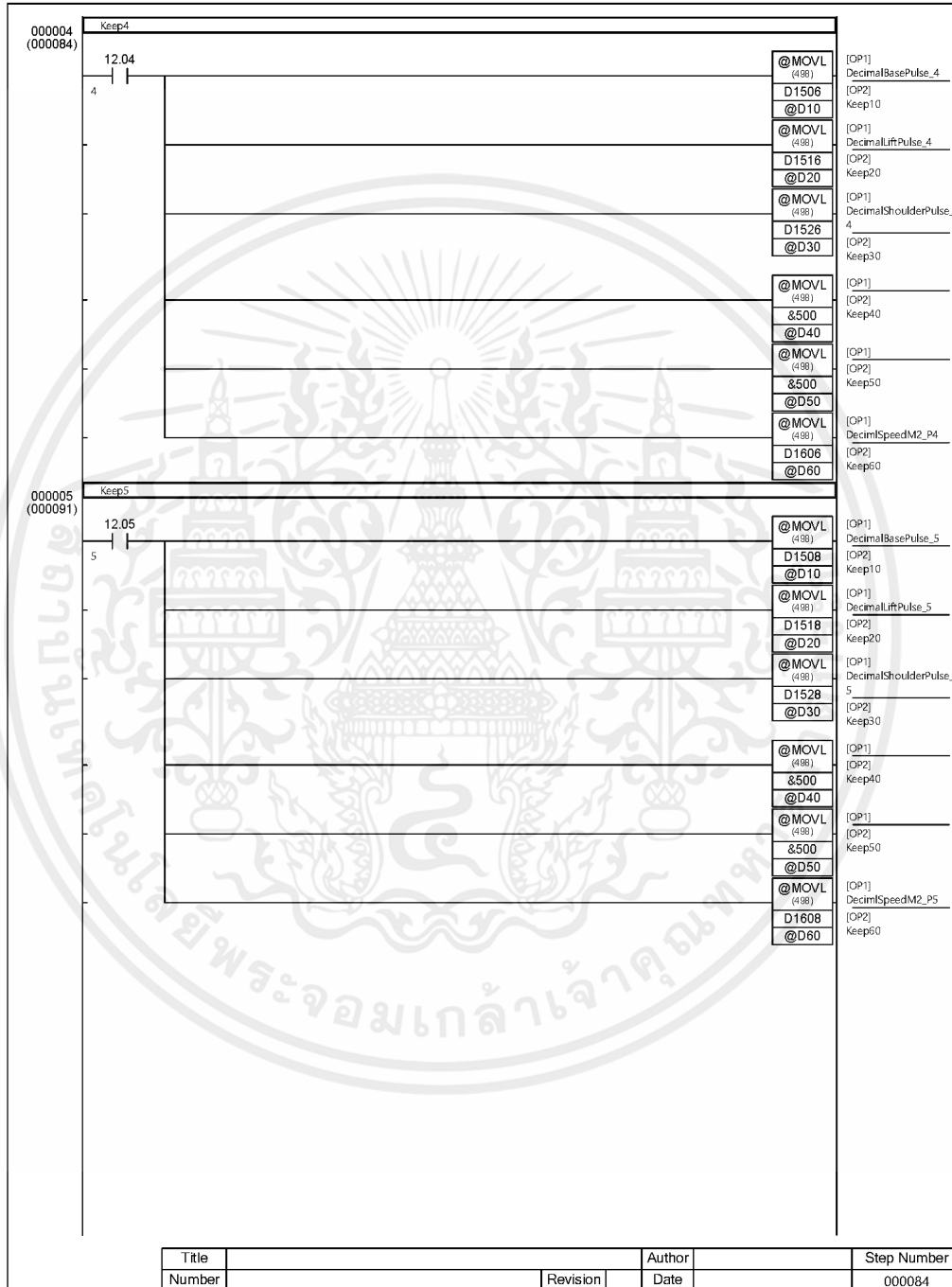
รูปที่ 38 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



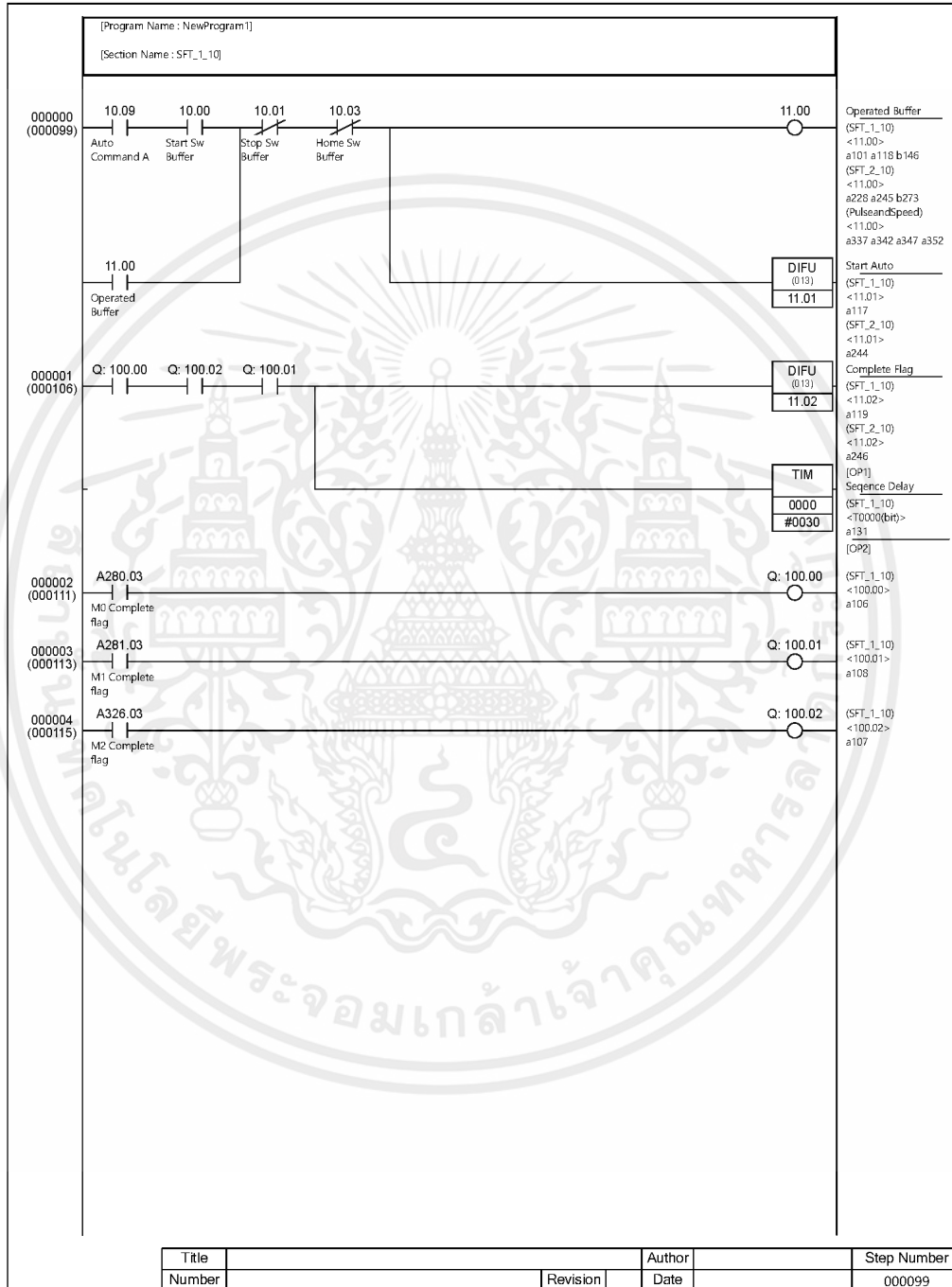
รูปที่ 39 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



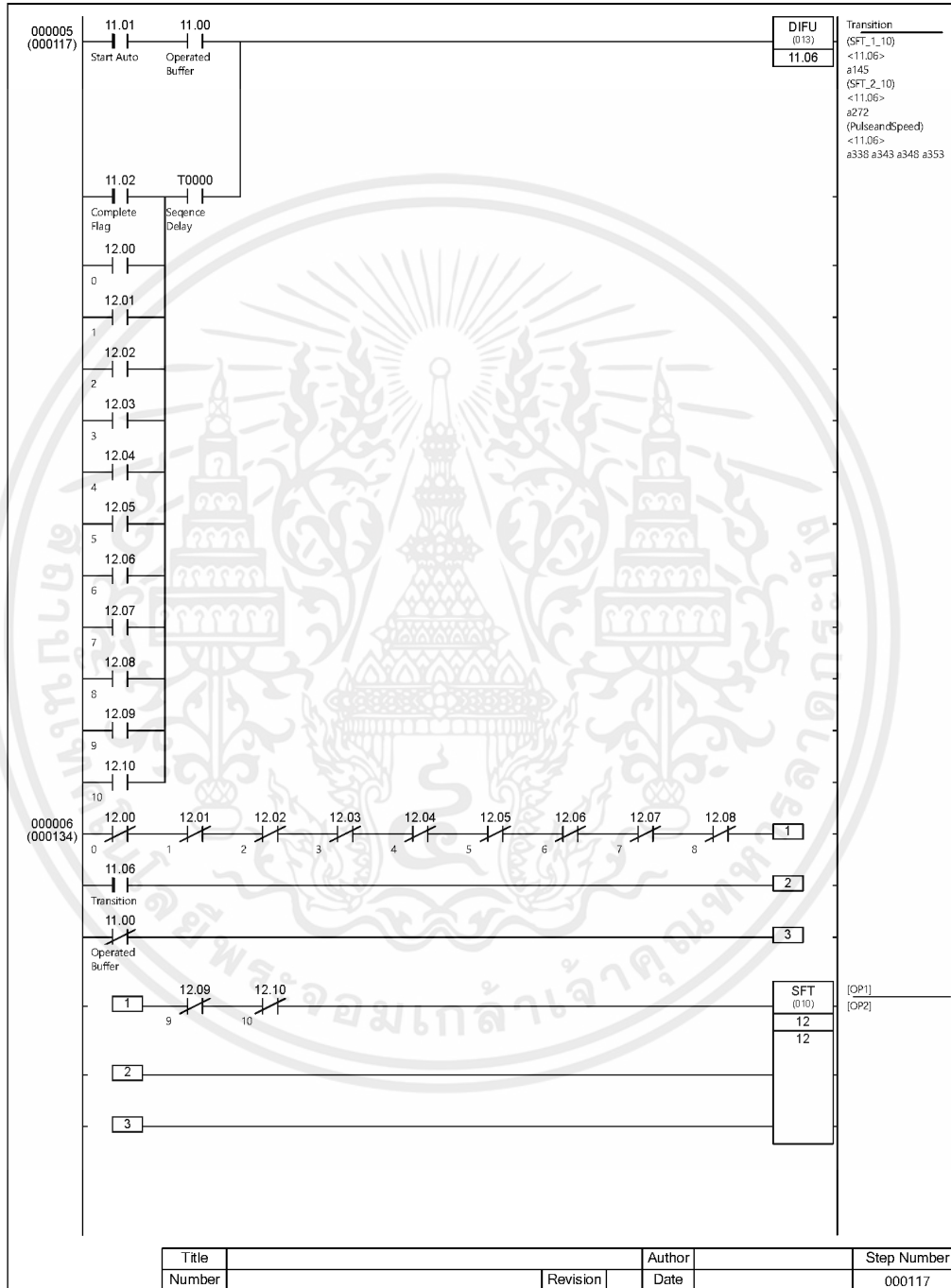
รูปที่ 40 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



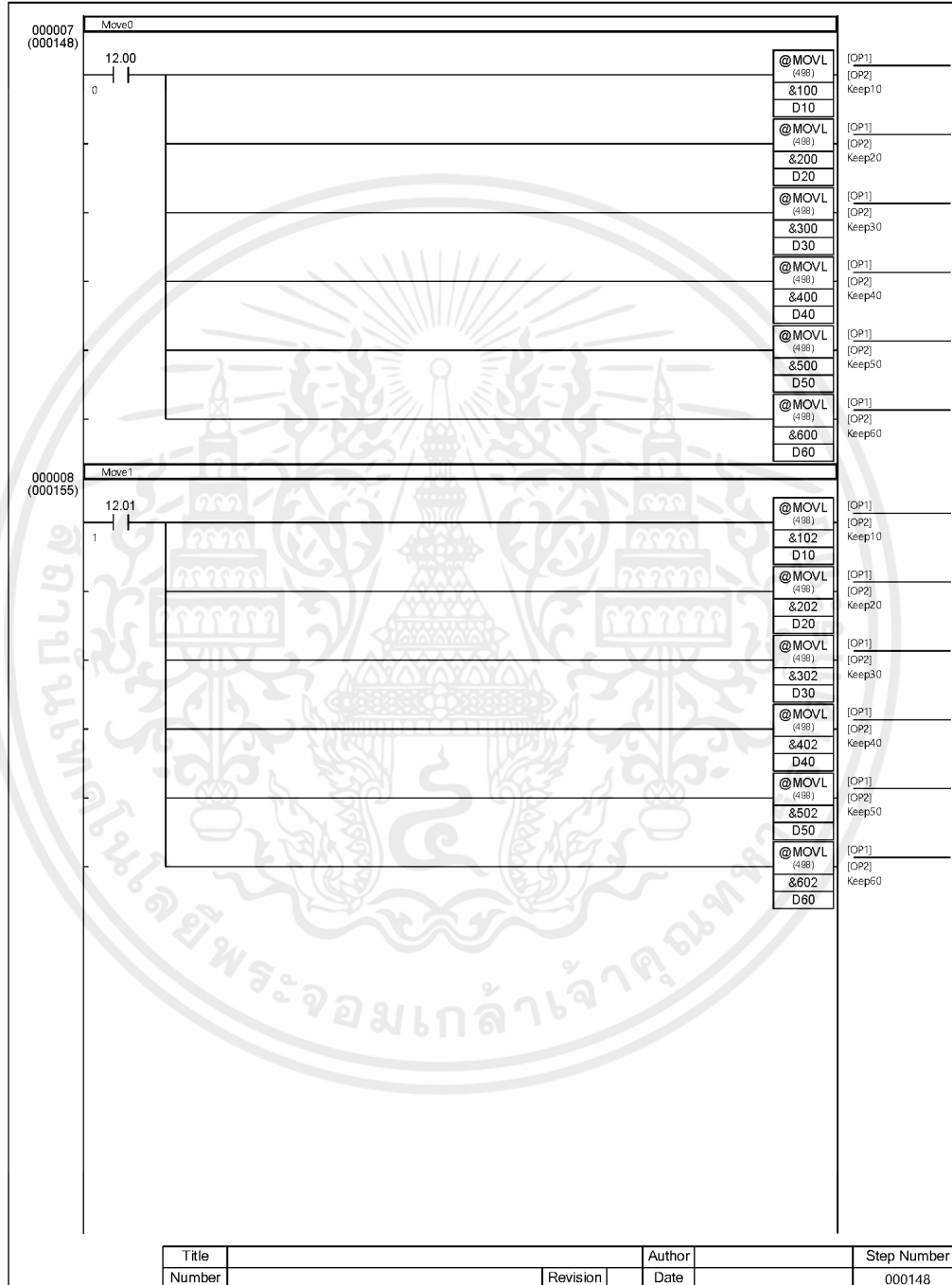
รูปที่ 41 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



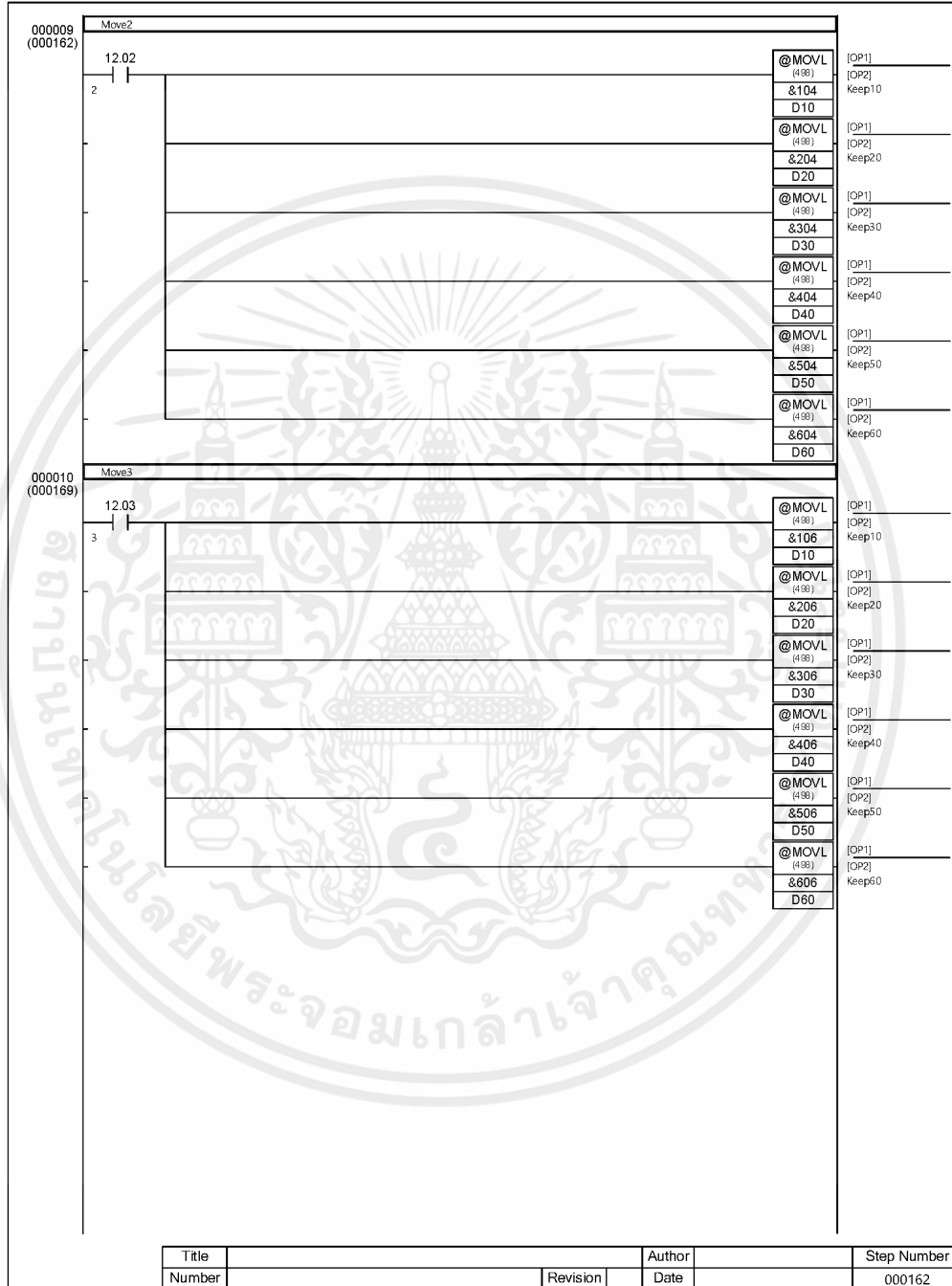
รูปที่ 42 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



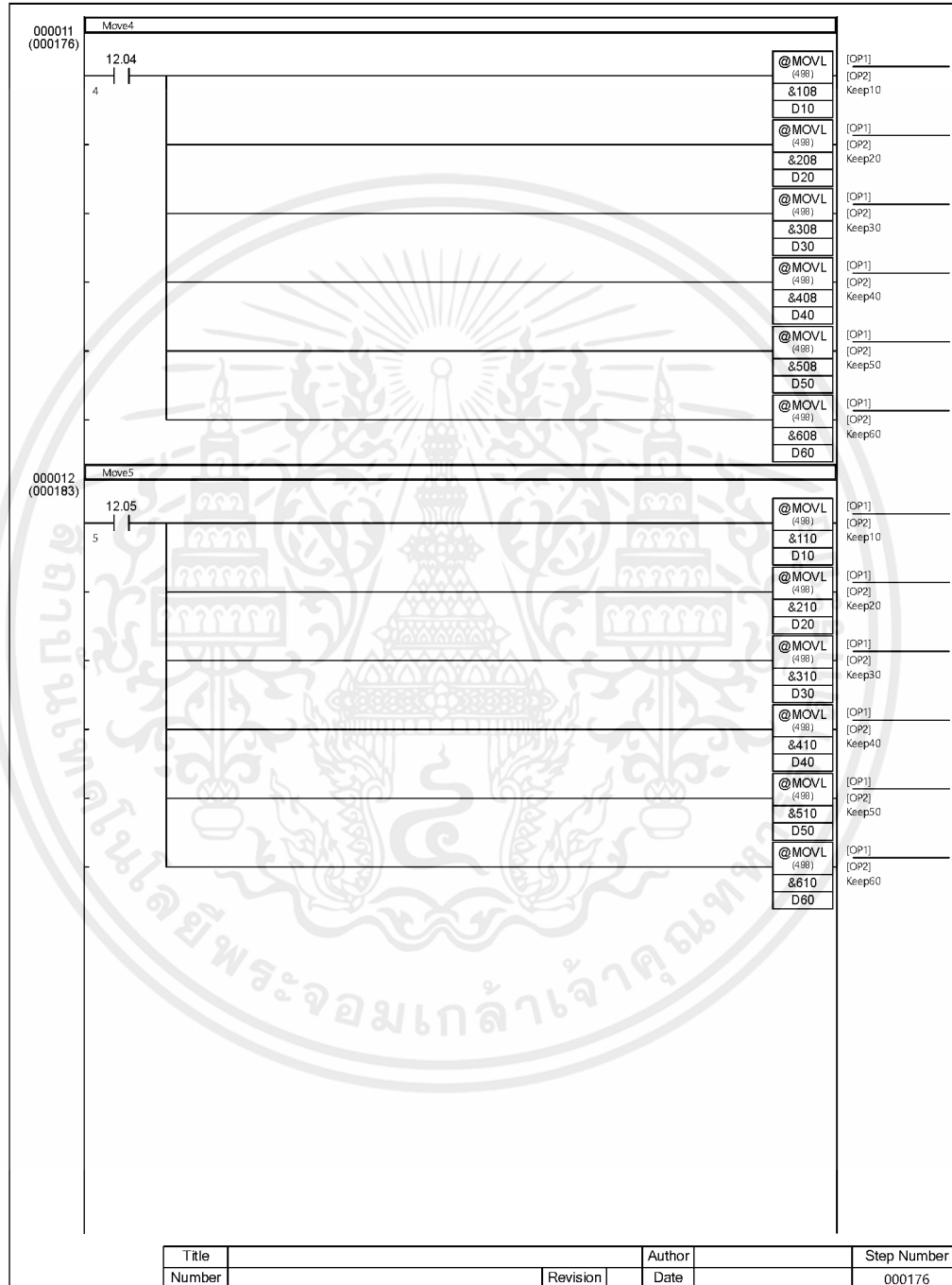
รูปที่ 43 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 44 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



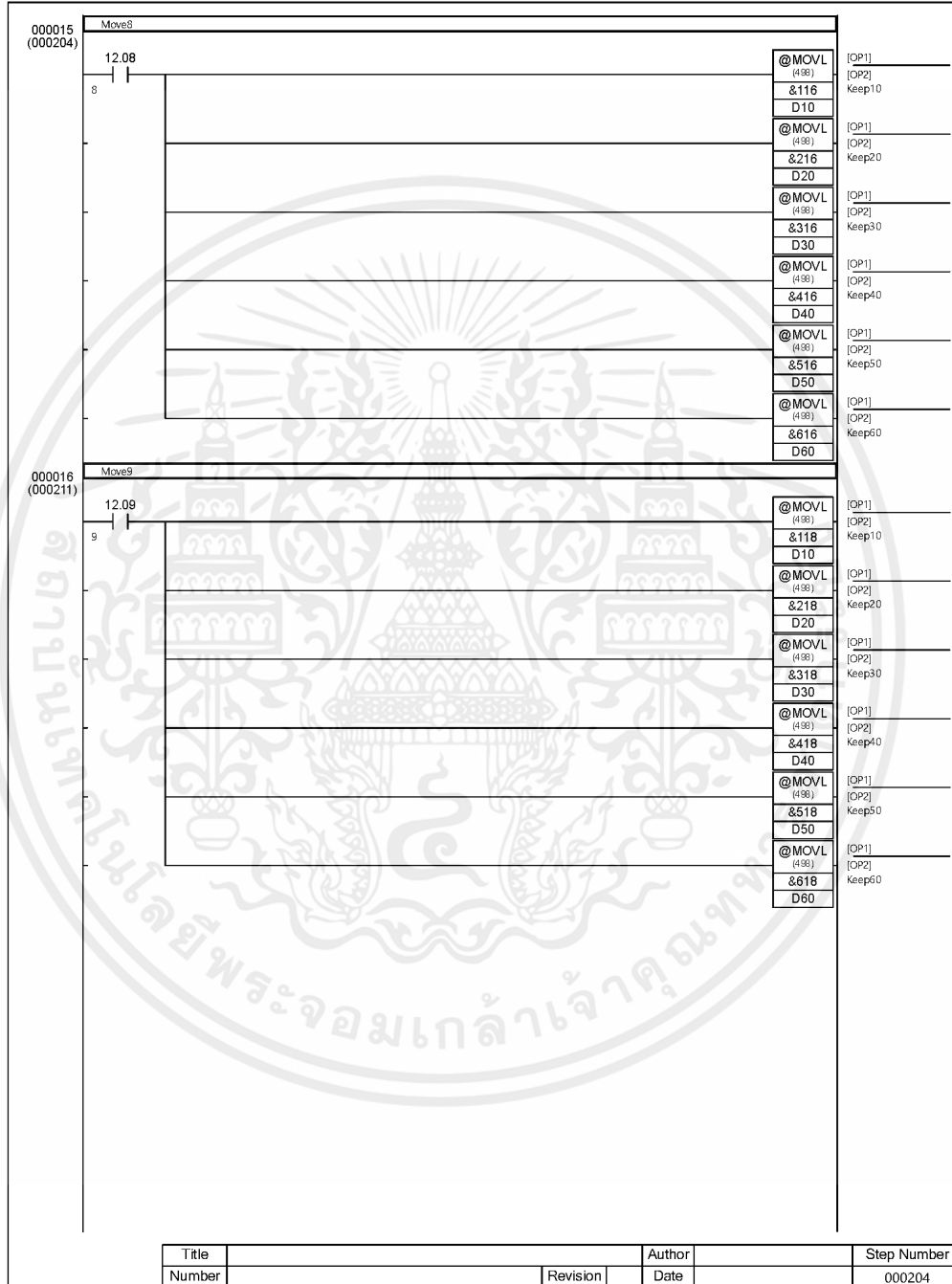
รูปที่ 45 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000013 (000190)	Move6						
	12.06						
	6		@MOV L (498)	&112	[OP1] [OP2]	Keep10	
			D10				
			@MOV L (498)	&212	[OP1] [OP2]	Keep20	
			D20				
			@MOV L (498)	&312	[OP1] [OP2]	Keep30	
			D30				
			@MOV L (498)	&412	[OP1] [OP2]	Keep40	
			D40				
		@MOV L (498)	&512	[OP1] [OP2]	Keep50		
		D50					
		@MOV L (498)	&612	[OP1] [OP2]	Keep60		
		D60					
000014 (000197)	Move7						
	12.07						
	7		@MOV L (498)	&114	[OP1] [OP2]	Keep10	
			D10				
			@MOV L (498)	&214	[OP1] [OP2]	Keep20	
			D20				
			@MOV L (498)	&314	[OP1] [OP2]	Keep30	
			D30				
			@MOV L (498)	&414	[OP1] [OP2]	Keep40	
			D40				
		@MOV L (498)	&514	[OP1] [OP2]	Keep50		
		D50					
		@MOV L (498)	&614	[OP1] [OP2]	Keep60		
		D60					
Title		Author	Step Number				
Number	Revision	Date	000190				

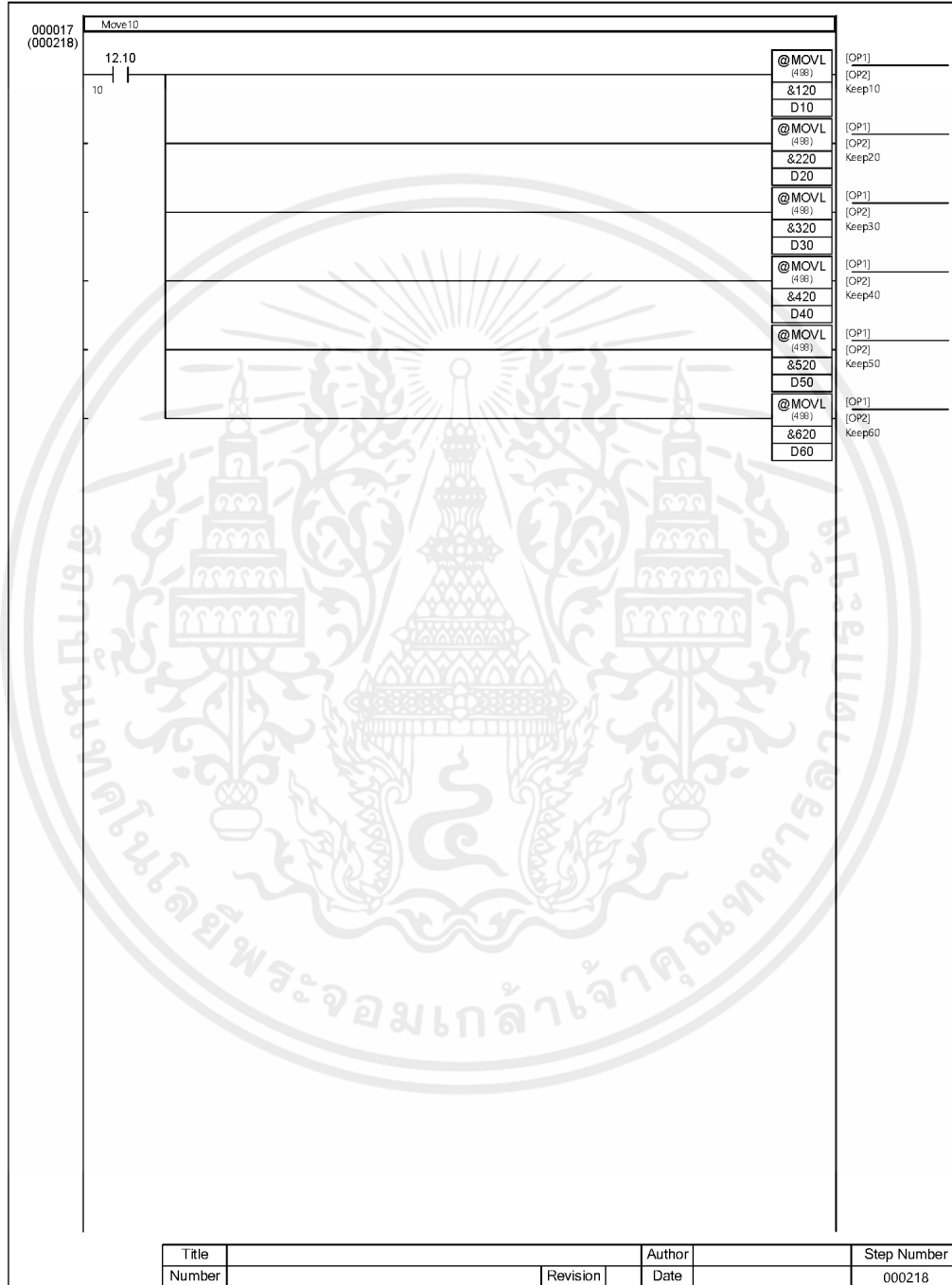
รูปที่ 46 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



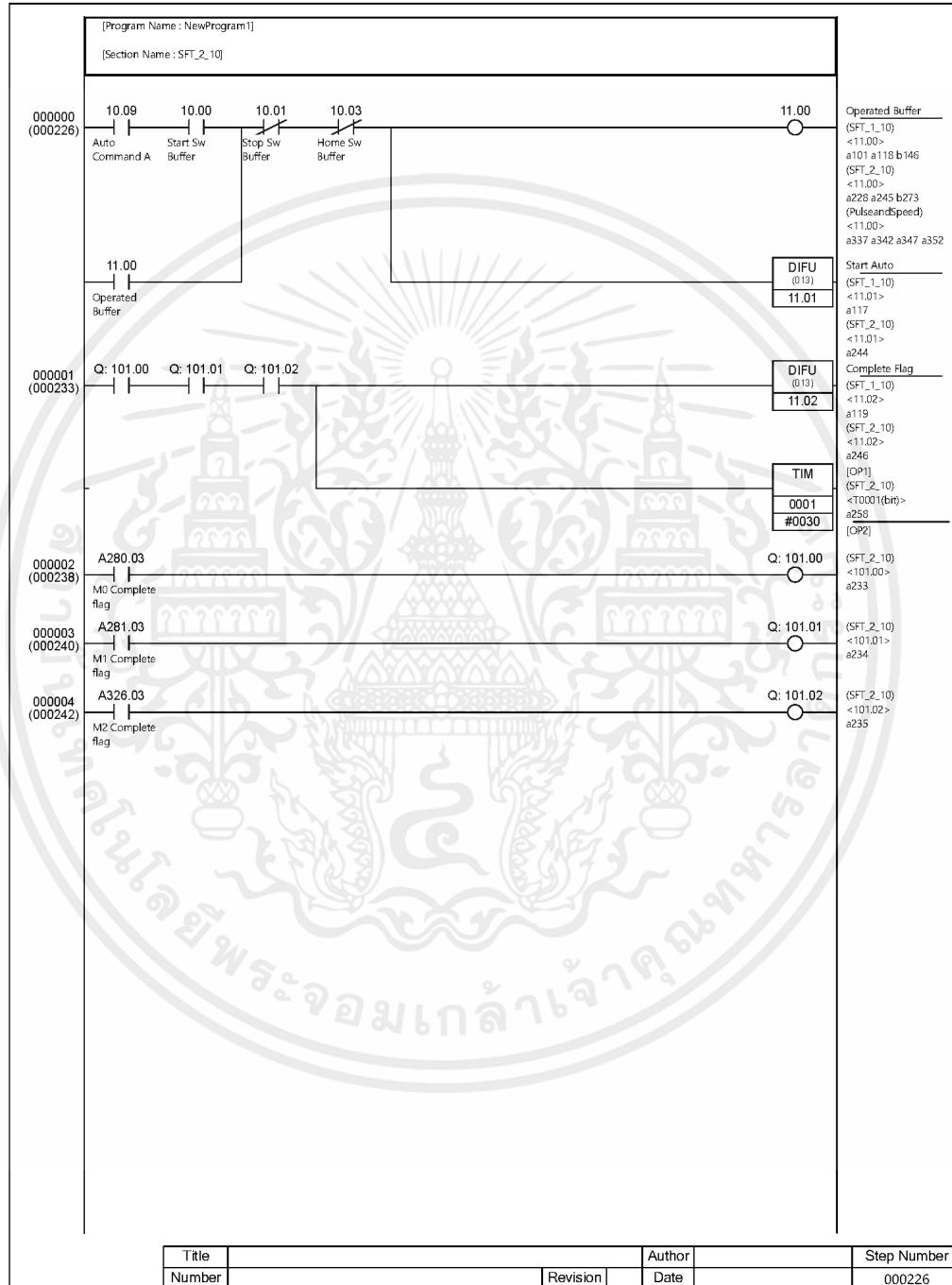
รูปที่ 47 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 23

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



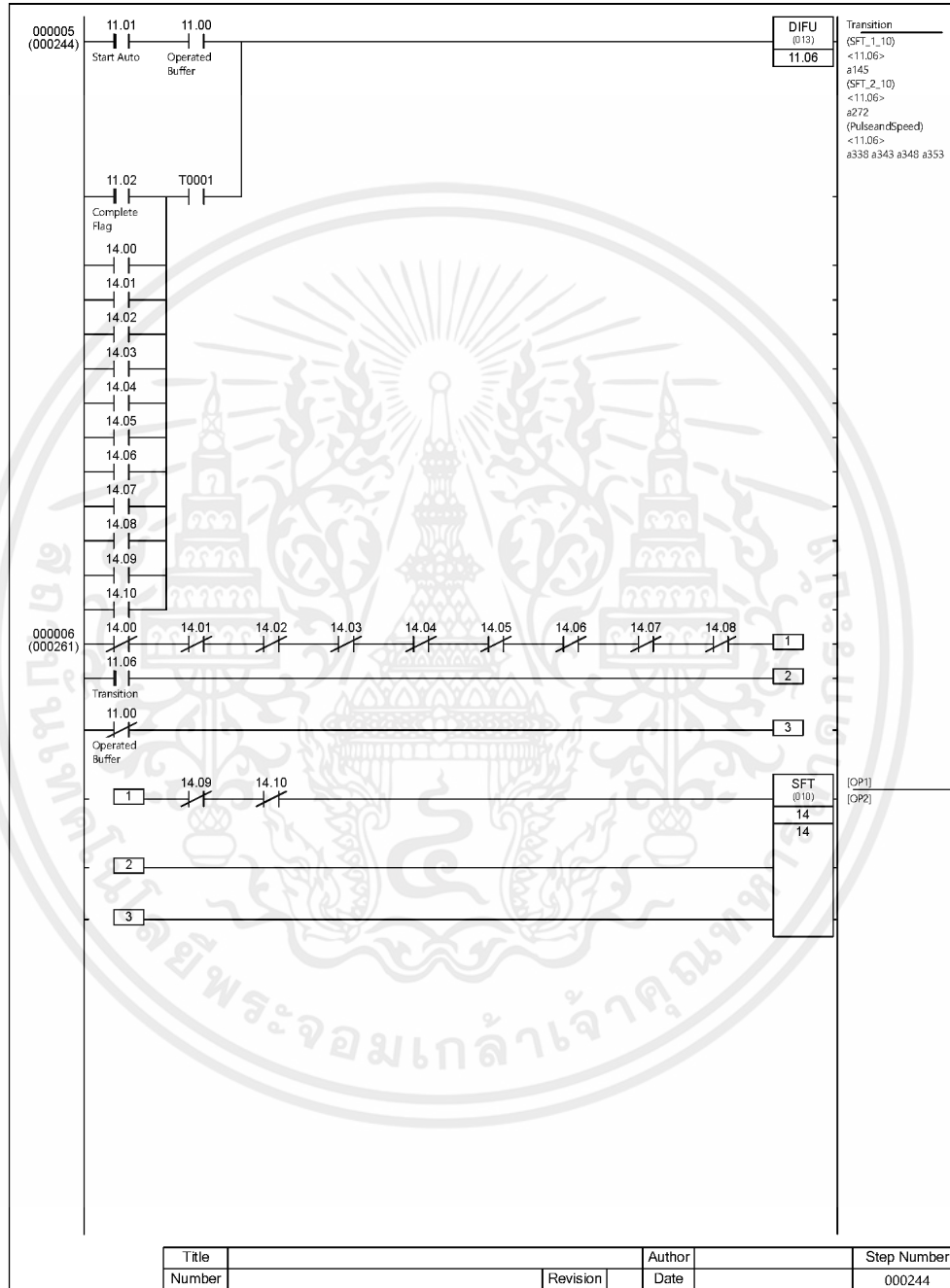
รูปที่ 48 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



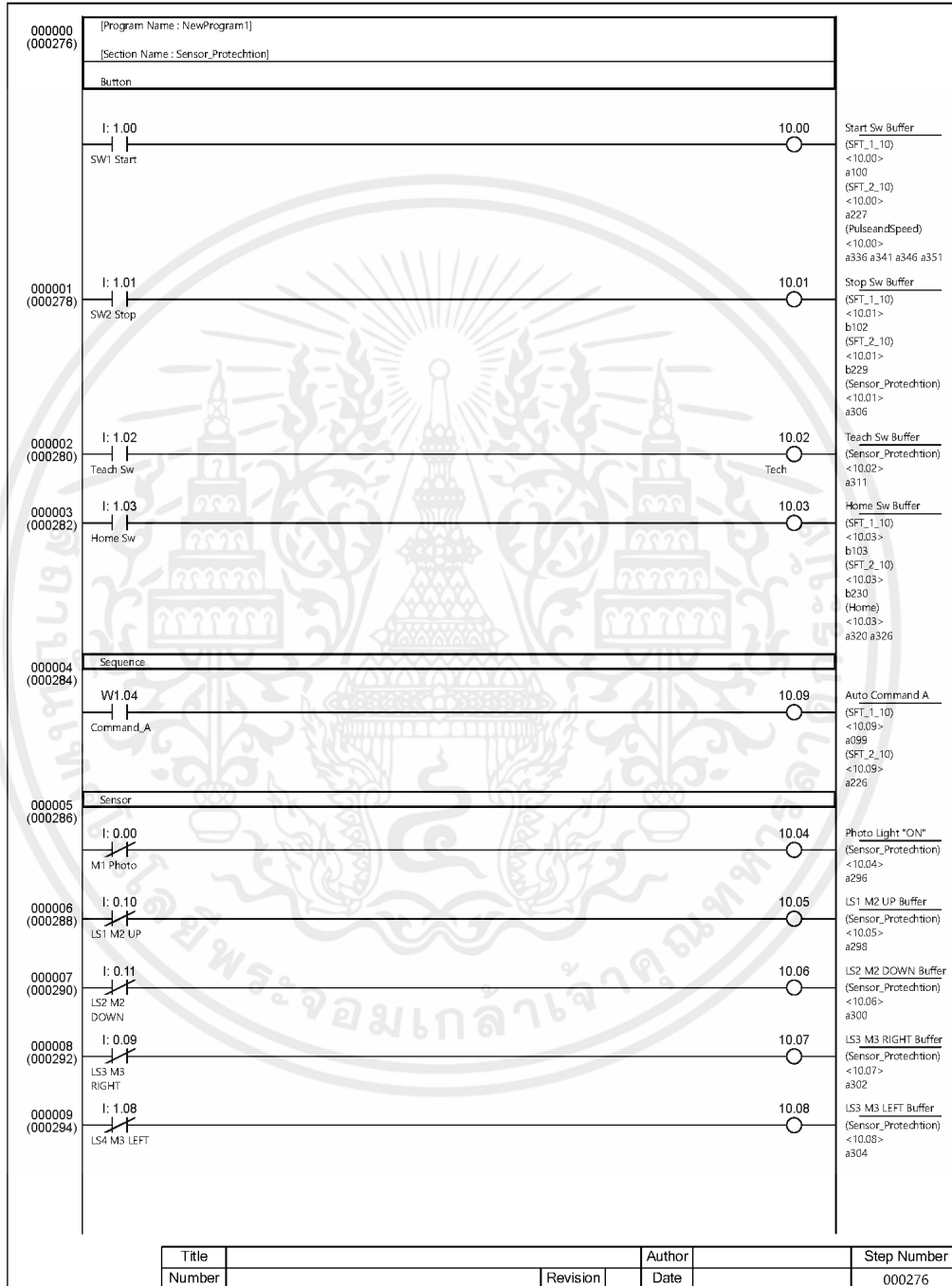
รูปที่ 49 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



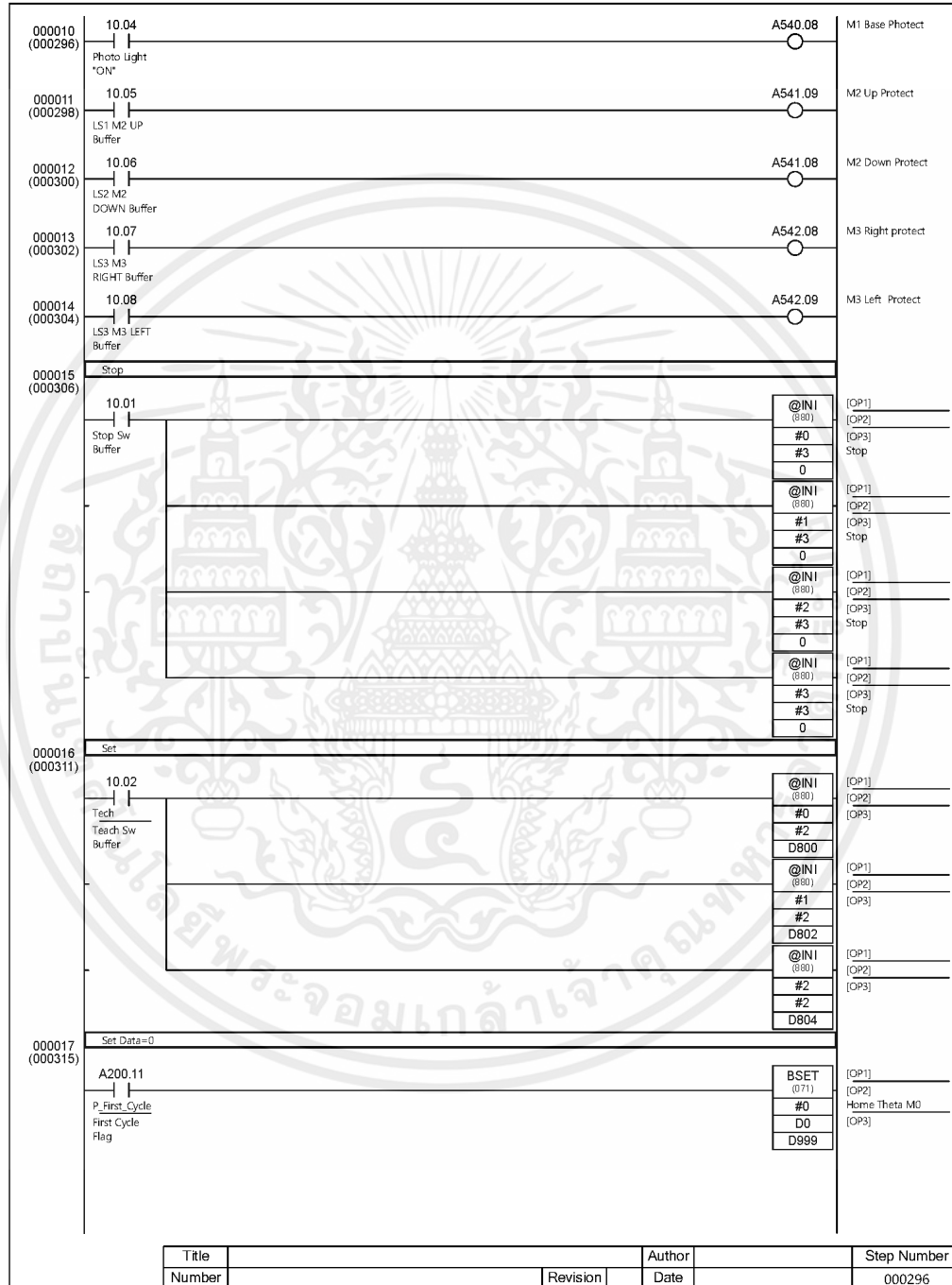
รูปที่ 50 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 51 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



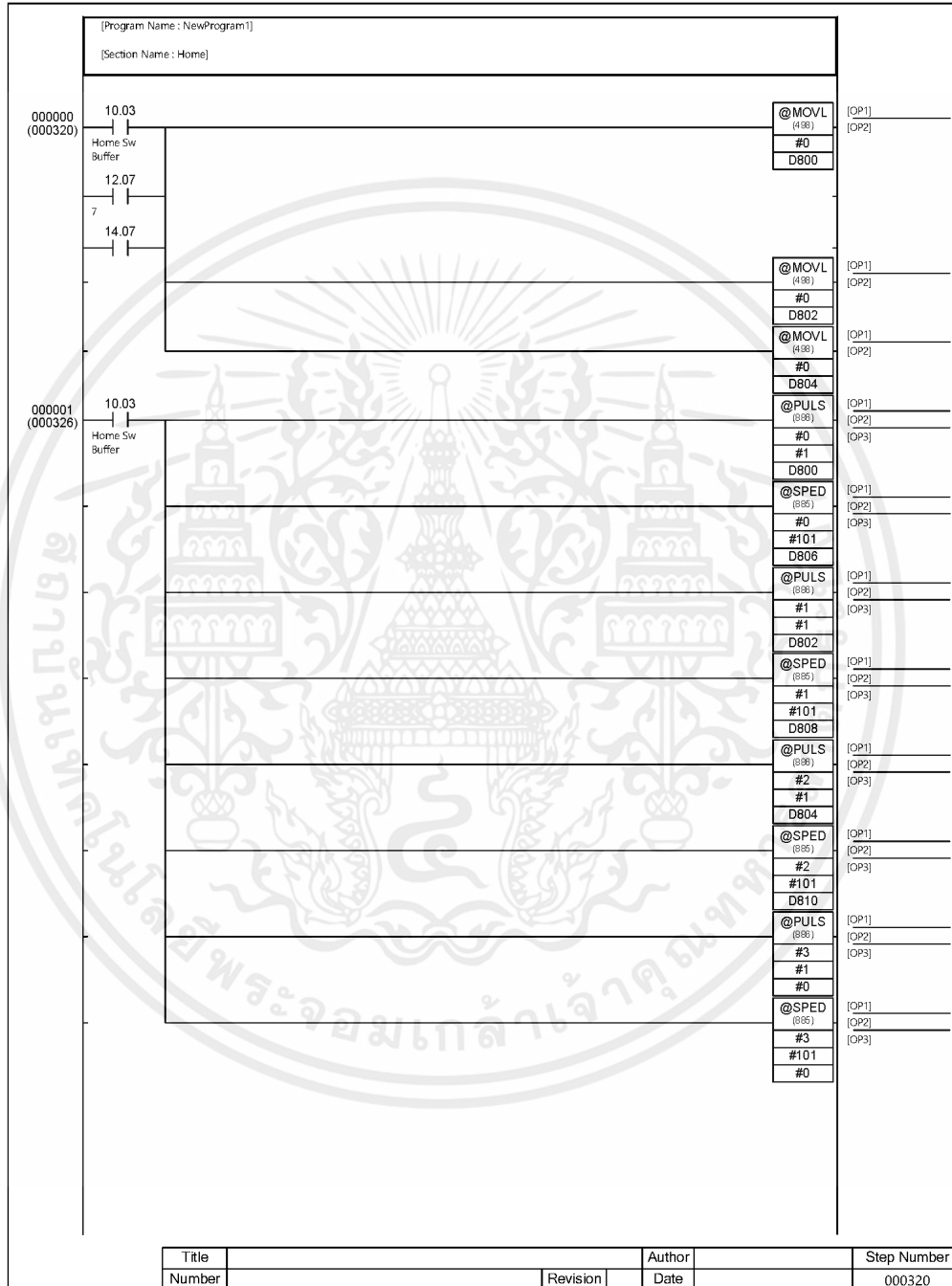
รูปที่ 52 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000018 (000317)	A200.11 P_First_Cycle First Cycle Flag	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">BSET</td> <td style="font-size: small;">(OP1)</td> </tr> <tr> <td style="text-align: center;">(01)</td> <td style="font-size: small;">(OP2)</td> </tr> <tr> <td style="text-align: center;">#0</td> <td style="font-size: small;">(OP3)</td> </tr> <tr> <td style="text-align: center;">D1040</td> <td></td> </tr> <tr> <td style="text-align: center;">D1700</td> <td></td> </tr> </table>	BSET	(OP1)	(01)	(OP2)	#0	(OP3)	D1040		D1700		
BSET	(OP1)												
(01)	(OP2)												
#0	(OP3)												
D1040													
D1700													
<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 30%; text-align: center;">Title</td> <td style="width: 30%; text-align: center;">Author</td> <td style="width: 40%; text-align: center;">Step Number</td> </tr> <tr> <td style="text-align: center;">Number</td> <td style="text-align: center;">Date</td> <td style="text-align: center;">000317</td> </tr> <tr> <td style="width: 40%;"></td> <td style="width: 10%; text-align: center;">Revision</td> <td></td> </tr> </table>		Title	Author	Step Number	Number	Date	000317		Revision				
Title	Author	Step Number											
Number	Date	000317											
	Revision												

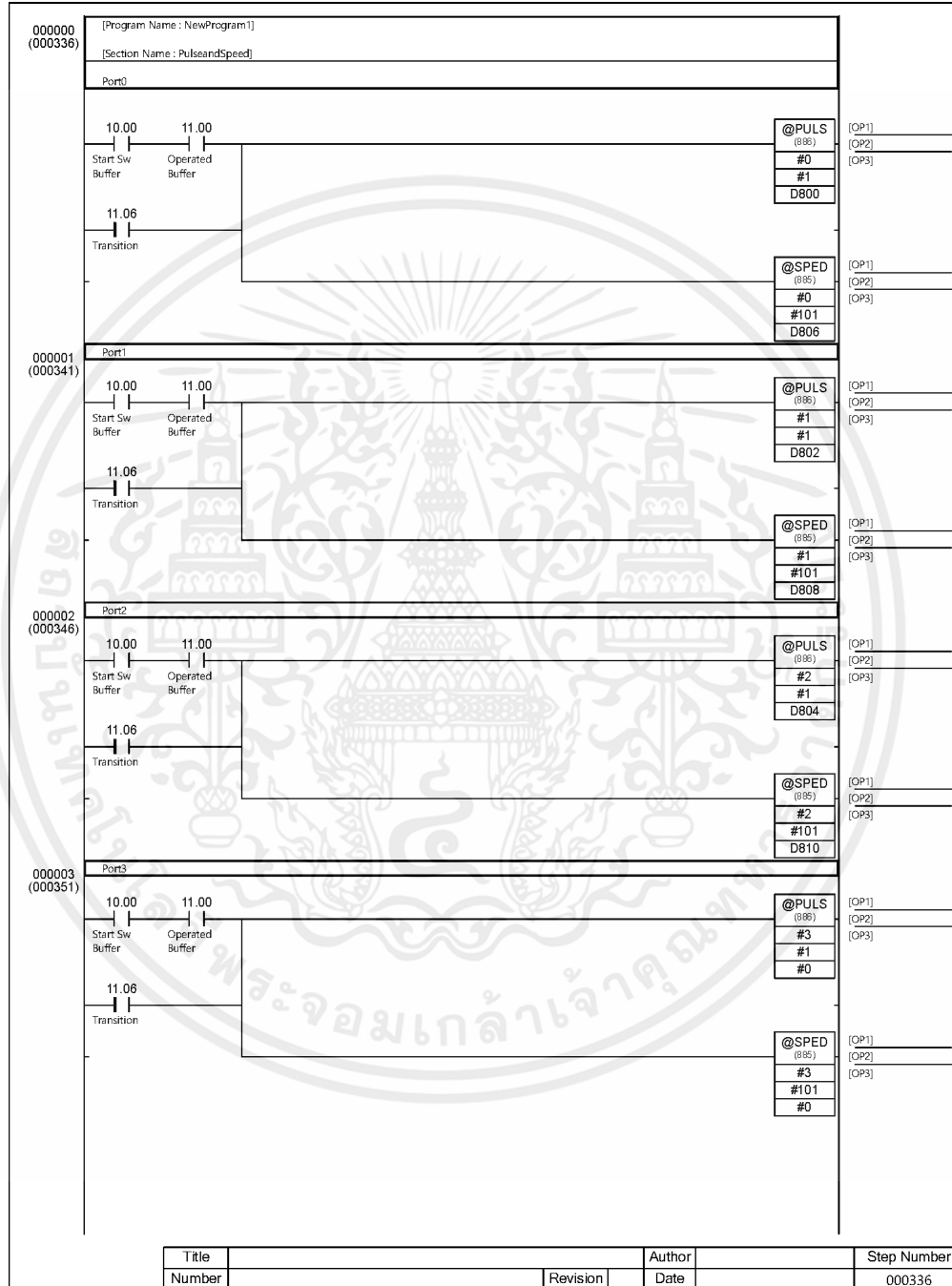
รูปที่ 53 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



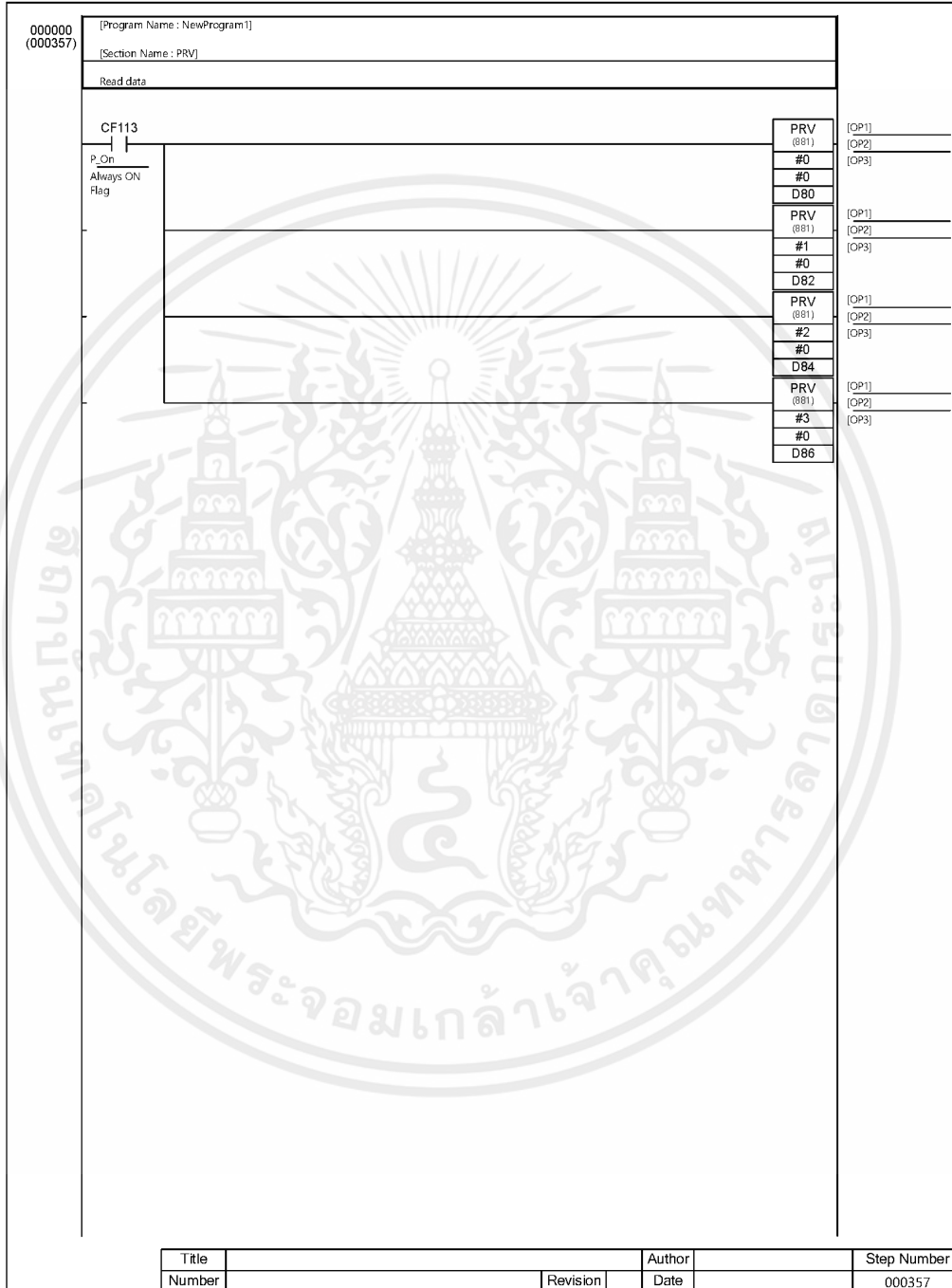
รูปที่ 54 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



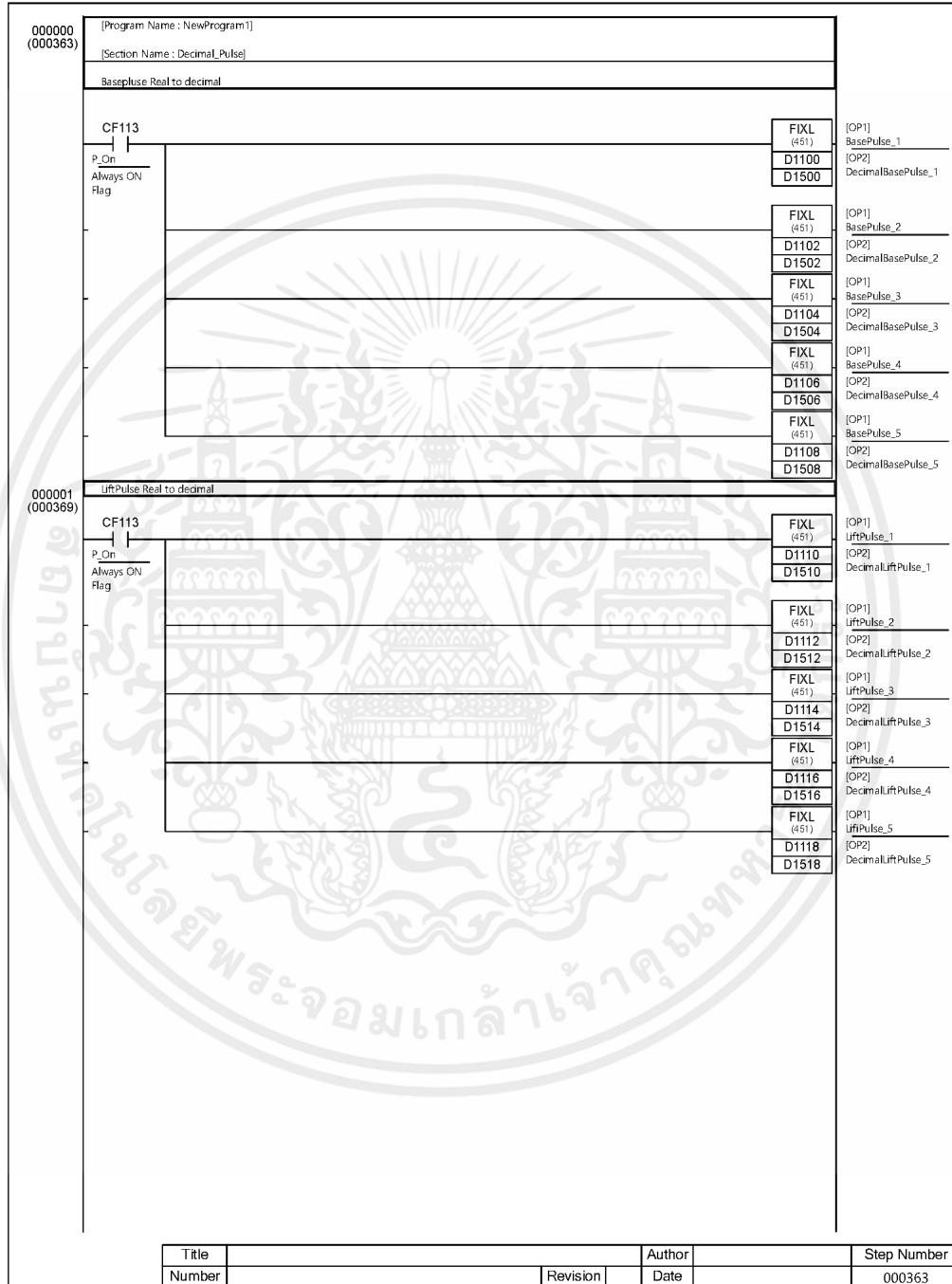
รูปที่ 55 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



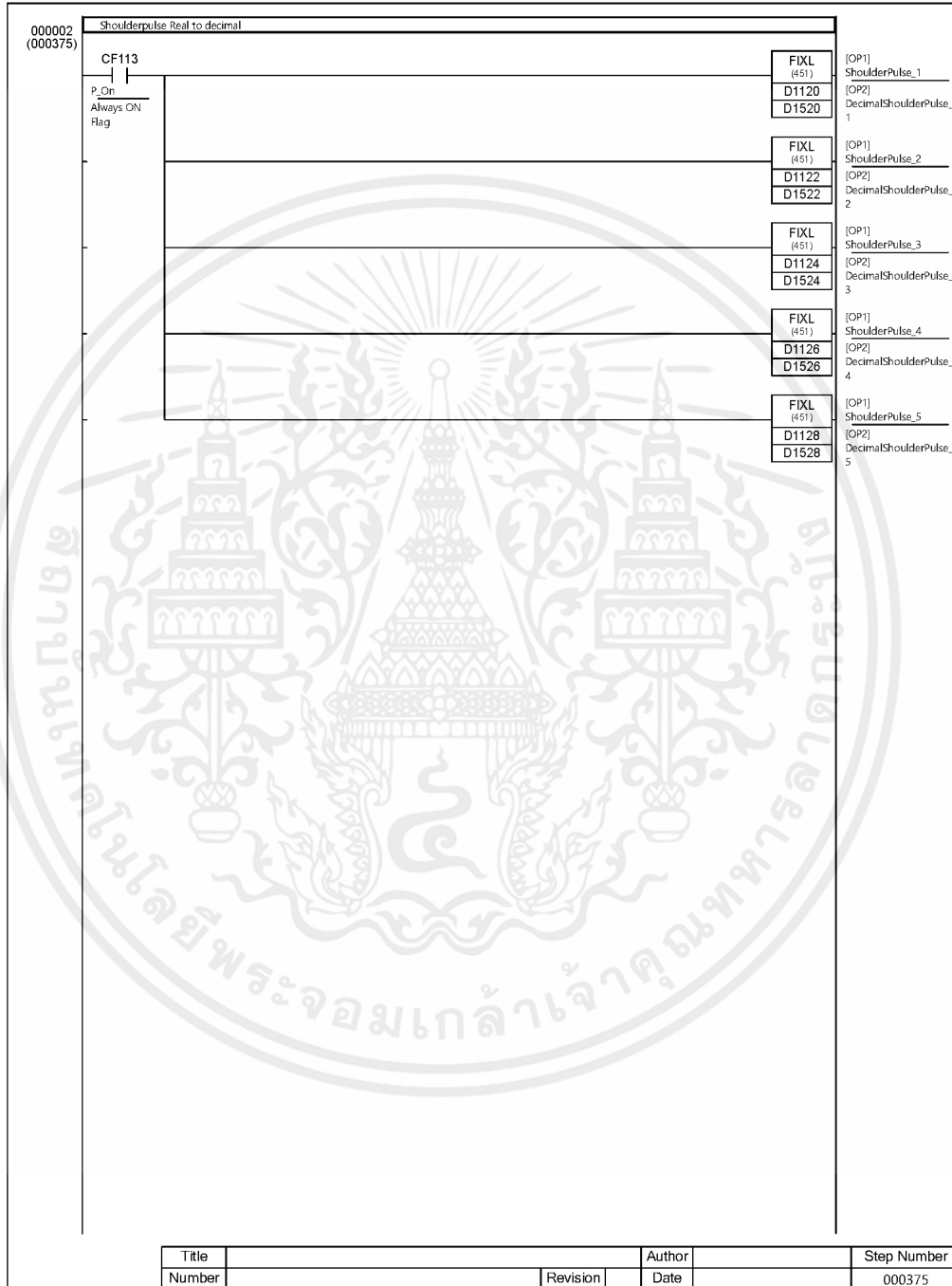
รูปที่ 56 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 32

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



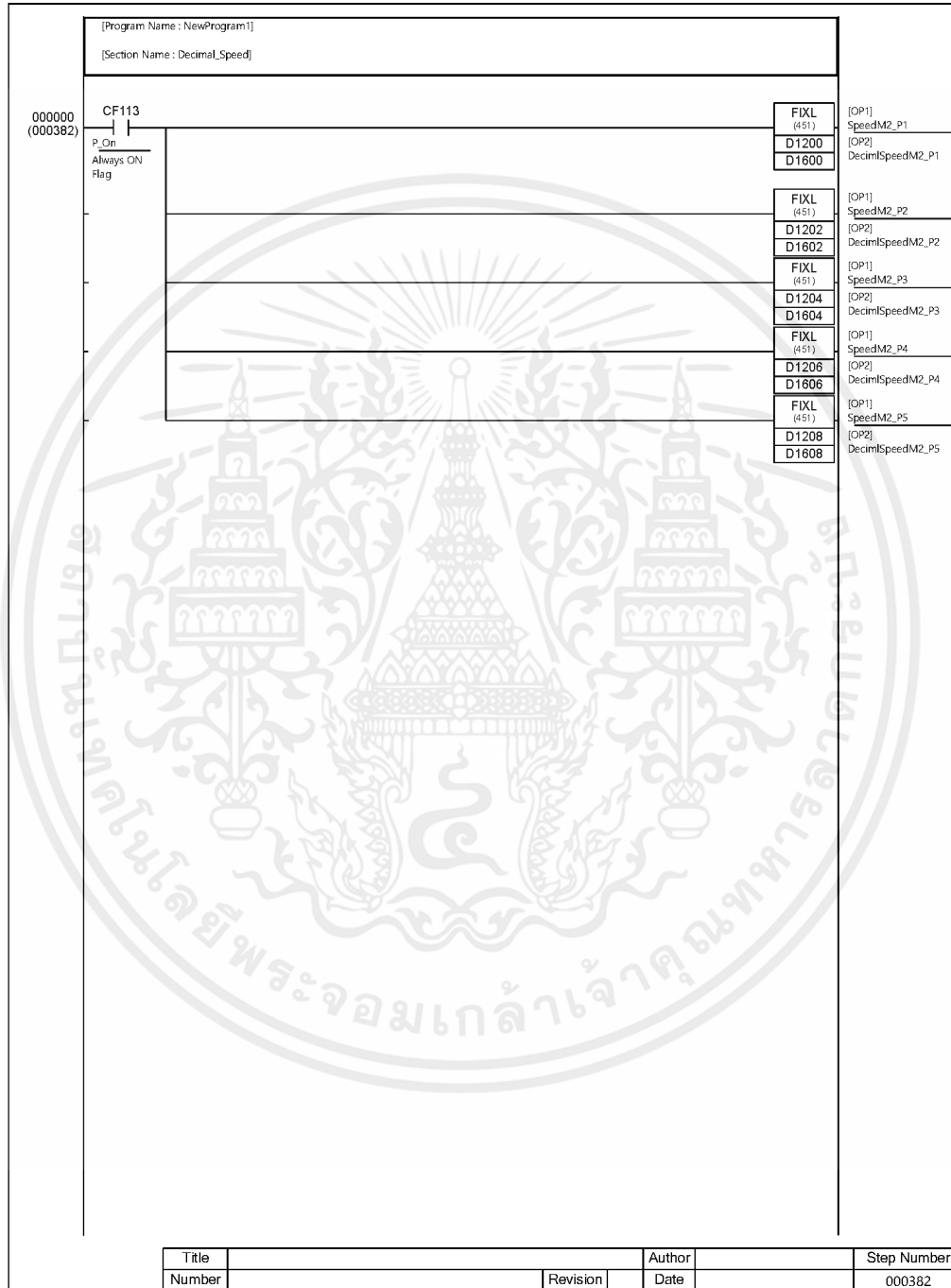
รูปที่ 57 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



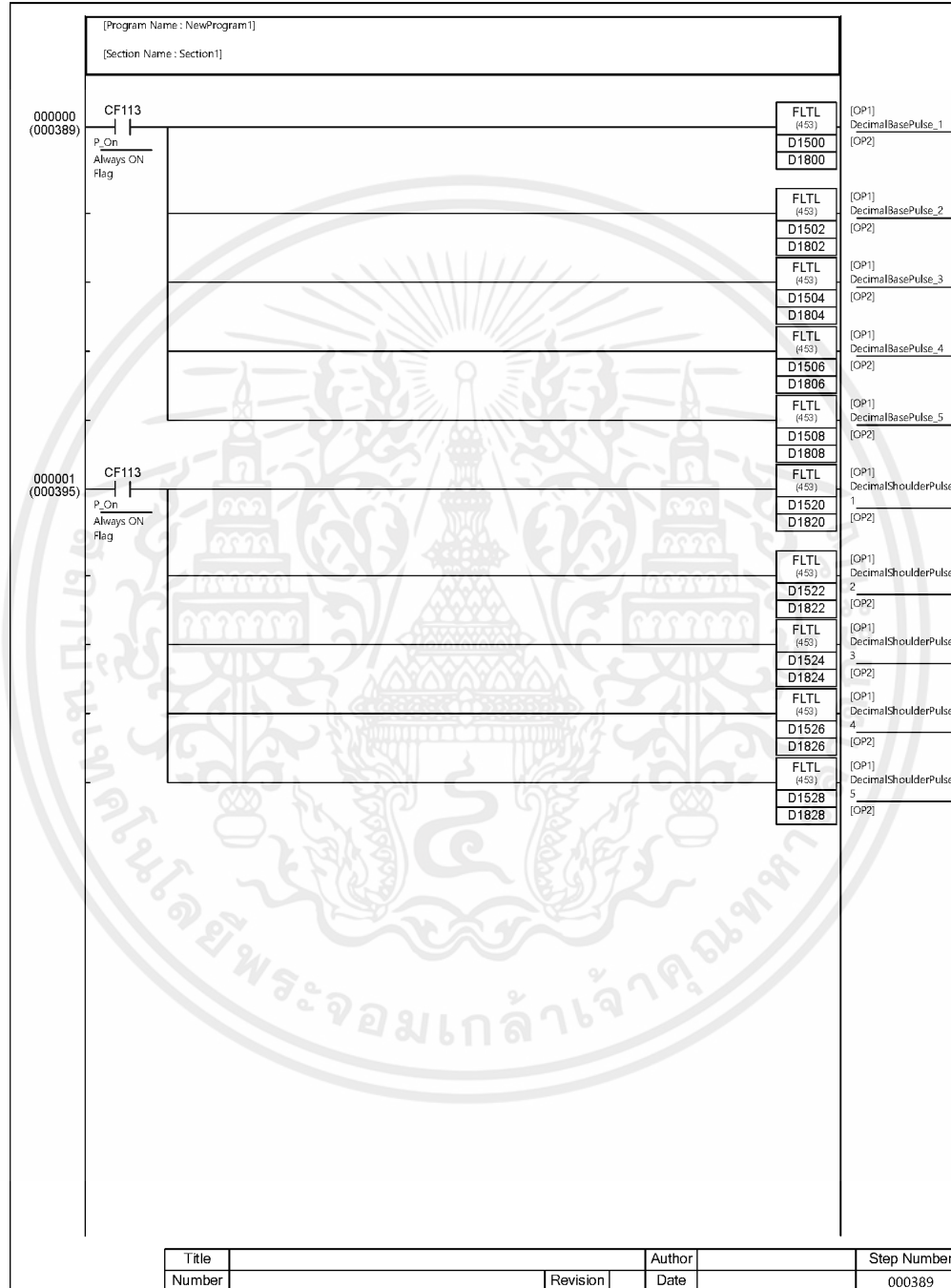
รูปที่ 58 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 59 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 60 แสดงโปรแกรม PLC ควบคุมแขนกลหน้าที่ 36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

