การควบคุมเซอร์โวมอเตอร์ด้วยเครื่องควบคุม PLC ร่วมกับ HMI และแสดงผลผ่านระบบเครือข่าย Ethernet

Servo Motor Control base on PLC with HMI through Ethernet

เพชร ผดุงยาม
ชูวงศ์ โพธิ์งามวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

การควบคุมเซอร์โวมอเตอร์ด้วยเครื่องควบคุม PLC ร่วมกับ HMI และแสดงผลผ่านระบบเครือข่าย Ethernet

Servo Motor Control base on PLC with HMI through Ethernet

เพชร ผดุงยาม
ชูวงศ์ โพธิ์งามวงศ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

Servo Motor Control base on PLC with HMI through Ethernet

Petch Phadungyam

Chuwong Phongamwong

A THESIS SUBMITTED IN PARTIAL FULFILMENT

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 2014

ปริญญานิพนธ์ปีการศึกษา 2557

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

ห้อข้อปริญญานิพนธ์ การควบคุมเซอร์โวมอเตอร์ด้วยเครื่องควบคุม PLC ร่วมกับ HMI และแสดงผลผ่านระบบเครือข่าย *Ethernet*

Servo Motor Control base on PLC with HMI through Ethernet

นักศึกษาผู้จัดทำ นายเพชร ผดุงยาม **รหัสนักศึกษา** 54010957

นายชูวงศ์ โพธิ์งามวงศ์ **รหัสนักศึกษา** 54010340

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2557

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
อาจารย์สุพรรณ กุลพาณิชย์	

หัวข้อปริญญานิพนธ์	การควบคุมเซอร์โวมอเตอร์ด้วยเครื่องควบคุม PLC ร่วมกับ HMI และแสดงผลผ่านระบบเครือข่าย Ethernet Servo Motor Control base on PLC with HMI through Ethernet
นักศึกษาผู้จัดทา	นายเพชร ผดุงยาม **รหัสนักศึกษา** 54010957
	นายชูวงศ์ โพธิ์งามวงศ์ **รหัสนักศึกษา** 54010340
อาจารย์ที่ปรึกษา	รองศาสตราจารย์สุพรรณ กุลพาณิชย์
ปีการศึกษา	2557

บทคัดย่อ

โครงงานฉบับนี้นำเสนอเป็นการประยุกต์การควบคุมเซอร์โวมอเตอร์ ด้วยเครื่องควบคุม PLC ผ่านทางหน้าจอ Touch Screen แบบกราฟิก และ หน้าจอคอมพิวเตอร์ ผ่านระบบเครือข่าย Ethernet ในรูปแบบของระบบ SCADA ซึ่งประกอบคอมพิวเตอร์ที่โปรแกรมบริหารข้อมูลกลางที่เป็น OPC Server (OLE for Process Control Server) ที่ทำงานอยู่บนโปรแกรม SIMATIC WinCC Graphics Designer เพื่อใช้เป็นศูนย์กลางในการส่งผ่านข้อมูลในรูปแบบ Server – Client สำหรับการประยุกต์ใช้งานนี้ผู้ใช้สามารถควบคุมและติดตามการเคลื่อนที่ของเซอร์โวมอเตอร์ได้ทั้งสองบริเวณ ทั้งจากหน้าจอแสดงผล Tough Screen หรือ จากเครื่องคอมพิวเตอร์ก็ได้ เพราะระบบควบคุมนี้ทำงานบนโครงข่ายแบบ Ethernet ในผลการทดลองจะแสดงให้เห็นถึงฟังค์ชั่น การเคลื่อนที่แบบ Absolute และ Relative โดยผู้ควบคุมสามารถดำเนินการได้อย่างสะดวกและรวดเร็ว

Thesis Title	Servo Motor Control base on PLC with HMI through Ethernet
Authors	Mr. Petch Phadungyam
	Mr. Chuwong Phongamwong
Thesis Advisor	Assoc.Prof. Suphan Kullapanich
Year	2014

Abstract

This project present the application of servo motor control based on PLC with touch screen and monitor through Ethernet protocol in SCADA format includes OPC server (OLE for Process Control Server) that operates on SIMATIC WinCC Graphics Designer program and work as center of data communication in Server-Client format. For this application users can control and monitor the position of servo motor by using 2 kinds of HMI as following Touch screen and monitor because this control system operates on Ethernet protocol. The result of experiment will show that functions of servo motor motion such as absolute move and relative move that the users can use it with ease and quickly.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เพราะได้รับคณาจารย์ภาควิชาวิศวกรรมการวัดคุม ทุกท่านที่ได้มอบคาแนะนำและกำลังใจในการทำวิจัยตลอดมา โดยเฉพาะอย่างยิ่ง ท่าน รศ.สุพรรณ กุลพาณิชย์ ซึ่งเป็นอาจารย์ที่ปรึกษา ที่ให้คำปรึกษา คำแนะนำรวมถึงอุปกรณ์ต่างๆแก่ผู้วิจัยตลอดมา จนทำให้การทำปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี

รวมไปถึงเพื่อนๆ พี่ๆ ที่คอยช่วยให้กาลังใจในการทำวิจัยและความช่วยเหลือในด้านต่างๆ แก่คณะผู้จัดทำ

ที่ลืมเสียมิได้คือ ขอกราบขอบพระคุณคุณพ่อคุณแม่ ที่สนับสนุนและเป็นกำลังใจเสมอมาในการทำปริญญานิพนธ์ฉบับนี้

สุดท้ายนี้ขอขอบคุณภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่มอบสิ่งดีๆตลอดที่คณะผู้จัดทำได้เล่าเรียนมาโดยตลอด คุณค่าและประโยชน์อันพึงมีจาปริญญานิพนธ์ฉบับนี้ คณะผู้จัดทาขอมอบแด่ผู้มีพระคุณทุกท่าน

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I
บทคัดย่อภาษาอังกฤษ.....II
กิตติกรรมประกาศ.....III
สารบัญ.....IV
สารบัญตาราง.....IX
สารบัญภาพ.....X

บทที่ 1 บทนำ.....1
1.1 ความสำคัญของปริญญานิพนธ์.....1
1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1
1.3 ขอบเขตของปริญญานิพนธ์.....2
1.4 ขั้นตอนการศึกษา.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....3
บทที่ 2 ทฤษฎีที่ใช้ในการออกแบบ.....4
2.1 ฮาร์ดแวร์.....4
2.1.1 โครงสร้างหน้าจอทัชสกรีน.....4
2.1.2 การสื่อสารข้อมูลแบบอนุกรม.....5
2.1.2.1 การส่งข้อมูลแบบซิมเพล็กซ์ (Simplex) และดูเพล็กซ์ (Duplex).....6
2.1.2.2 โปรโตคอลของการสื่อสารแบบอนุกรม.....7
2.1.2.3 มาตรฐานสัญญาณอนุกรมแบบ RS-232C.....8
2.1.2.4 พอร์ตสื่อสารอนุกรม.....9
2.1.3 โครงสร้างพื้นฐานของตัวควบคุมพีแอลซี.....10
2.1.4 IEEE 802.3 : Ethernet.....11
2.1.5 TCP/IP.....13
2.1.6 เซอร์โวไดรเวอร์ (Servo Driver).....13

สารบัญ(ต่อ)

หน้า

2.1.6.1 การตั้งค่าสำหรับเซอร์โวไดรเวอร์ G5-series
รุ่น R88D-KP01H....14
2.1.6.2 ชนิดของอินพุตควบคุมสำหรับเซอร์โวไดรฟ์เวอร์....15
2.1.7 เซอร์โวมอเตอร์ (Servo Motor)....16
2.1.7.1 โครงสร้างของเซอร์โวมอเตอร์....17
2.1.7.2 Ball Screw....18
2.1.8 หลักการทำงานของเอ็นโค้ดเดอร์....18
2.1.9 การเชื่อมต่อระหว่างเซอร์โวมอเตอร์ และ Omron CP1H....20
2.1.10 Photoelectric Sensors....22
2.2 ซอฟต์แวร์....24
2.2.1 แลดเดอร์ไดอะแกรม (Ladder Diagrams)....24
2.2.1.1 กลุ่มคำสั่งลอจิกพื้นฐาน....24
2.2.1.1.1 การใช้คำสั่ง LOAD (LD), LOAD NOT (LD NOT)....25
2.2.1.1.2 การใช้คำสั่ง AND, AND NOT....25
2.2.1.1.3 การใช้คำสั่ง OR, OR NOT....26
2.2.1.1.4 การใช้คำสั่ง OUT....26
2.2.1.1.5 การใช้คำสั่ง AND LOAD (AND LD), OR LOAD (OR LD)....26
2.2.1.1.6 การใช้คำสั่ง END (END)....28
2.2.1.1.7 การใช้งานคำสั่ง SFT (10)....28
2.2.1.1.8 การใช้งานคำสั่ง IL (02), ILC (03)....28
2.2.1.1.9 การใช้งานคำสั่ง Differential UP-DIFU(013) และ Differential....29
2.2.1.1.10 การใช้คำสั่ง FB: Move Relative....29
2.2.1.1.11 การใช้คำสั่ง FB: Move Absolute....31
2.2.1.1.12 การใช้คำสั่ง FB : ORIGIN SEARCH....33
2.2.1.1.13 การใช้คำสั่ง FB : Move Velocity....35
2.2.1.1.14 การใช้คำสั่ง FB : Stop....36

สารบัญ(ต่อ)

หน้า

2.2.1.1.15 การใช้คำสั่ง FB : Read Actual Position.....38
2.2.2 OPC (Ole For Process Control).....39
2.2.3 ระบบ SCADA.....44
2.2.3.1 โครงสร้างของ SCADA (Architecture).....44
2.2.3.2 โครงสร้างด้านซอร์ฟแวร์ (Software Architecture).....45
2.2.3.3 โครงสร้างด้านการสื่อสาร (Communications).....47
2.2.3.4 โครงสร้างอินเทอร์เฟส (Interface).....47
2.2.3.5 โครงสร้างความสามารถในการขยายระบบ (Scalability).....47
2.2.3.6 โครงสร้างการสำรองระบบ(Redundancy).....48
2.2.3.7 หน้าที่การทำงาน (Functionality).....48
2.2.3.8 ระบบแสดงผลแบบ MMI (Man Machine Interface).....48
2.2.3.9 ระบบแสดงกราฟสัญญาณแบบต่อเนื่อง (Trending).....49
2.2.3.10ระบบแจ้งเตือน(Alarm).....49
2.2.3.11 การทำงานแบบ Automation.....49
2.2.3.12 การสร้างและพัฒนา (Application Development) 49
2.2.3.13 เครื่องมือในการพัฒนา (Development Tool).....50
บทที่ 3 การออกแบบและการทำงาน.....51
3.1 การเชื่อมต่ออุปกรณ์ทางด้านฮาร์ดแวร์.....52
3.1.1 การเชื่อมต่อระหว่างเซอร์โวมอเตอร์ กับ เครื่องควบคุมพีแอลซี.....52
3.1.2 การเชื่อมต่อระหว่างหน้าจอทัชสกรีน กับ เครื่องควบคุมพีแอลซี.....52
3.1.3 การเชื่อมต่อระหว่างโฟโต้เซนเซอร์ กับ เครื่องควบคุมพีแอลซี.....53
3.1.4 การเชื่อมต่อระหว่างคอมพิวเตอร์ กับ เครื่องควบคุมพีแอลซี.....53
3.2 การเชื่อมต่ออุปกรณ์ทางด้านซอฟต์แวร์.....54
3.2.1 การตั้งค่าเครื่องควบคุมพีแอลซี กับ คอมพิวเตอร์.....54
3.2.2 การตั้งค่าโปรแกรมสำหรับการเชื่อมต่อ OPC Server.....55
3.2.3 การตั้งค่าโปรแกรมสำหรับการเชื่อมต่อ SCADA.....56
3.3 การกำหนดค่า Electronic Gear Integer Setting และ
ชนิดอินพุตของเซอร์โวไดรเวอร์.....56

สารบัญ(ต่อ)

หน้า

3.4 การเขียนโปรแกรมและกราฟิกการทำงานของระบบ....58

3.4.1 การเขียนโปรแกรมสำหรับควบคุมพีแอลซี....58

3.4.2 การเพิ่ม Channel, Device และ Point ของ PLC ใน KEPserverEX5 และการตั้งค่าการเชื่อมต่อ....59

3.4.3 การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอทัชสกรีน....60

3.2.4 การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอ HMI....61

บทที่ 4 ผลการทดลอง....64

4.1 ผลการทดลองการใช้หน้าจอทัชสกรีนในการสั่งการเซอร์โวมอเตอร์และการแสดงผล....64

4.1.1 การดำเนินการทางจุดกำเนิด (Origin Operation)....65

4.1.2 การดำเนินการ JOG (JOG Operation)....67

4.1.3 การดำเนินการ Relative (Relative Operation)....69

4.1.4 การดำเนินการ Absolute (Absolute Operation)....72

4.2 ผลการทดลองการใช้โปรแกรม SIMATIC WinCC Explorer สั่งการเซอร์โวมอเตอร์ และการแสดงผล....74

4.2.1 ส่วนสั่งการของการควบคุมเซอร์โวมอเตอร์....74

4.2.2 ส่วนแสดงผลของการควบคุมเซอร์โวมอเตอร์....77

4.3 การทดสอบการทำงานของหน้าจอ HMI....78

4.3.1 การทดสอบการทำงานของระบบผ่านหน้าจอ HMI ด้วยการกำหนดตัวแปรแบบที่ 1....78

4.3.2 การทดสอบการทำงานของระบบผ่านหน้าจอ HMI ด้วยการกำหนดตัวแปรแบบที่ 2....78

4.3.3 การทดสอบการทำงานของปุ่มหยุดฉุกเฉิน (Emergency)....78

4.3.4 การทดสอบการทำงานของปุ่มหยุด (Deceleration Stop)....79

4.3.5 การทดสอบการทำงานของปุ่มล็อคโรเตอร์ (Servo Lock)....79

4.4 การทดสอบการทำงานของหน้าจอทัชสกรีน....82

สารบัญ(ต่อ)

หน้า

4.4.1 การทดสอบการทำงานของระบบผ่านหน้าจอทัชสกรีนด้วยการกำหนดตัวแปรแบบที่ 1 82
4.4.2 การทดสอบการทำงานของระบบผ่านหน้าจอทัชสกรีนด้วยการกำหนดตัวแปรแบบที่ 2 82
4.4.3 การทดสอบการทำงานของปุ่มหยุดฉุกเฉิน (Emergency) 82
4.4.3 การทดสอบการทำงานของปุ่มล็อคโรเตอร์ (Servo Lock) 82
บทที่ 5 สรุปผลการทดลองและข้อเสนอแนะ 85
5.1 สรุปผลการทดลอง 85
5.2 ข้อเสนอแนะ 86
5.3 อุปสรรคและปัญหา 86

สารบัญตาราง

ตารางที่ **หน้า**

ตารางที่ 2.1 ขาสัญญาณต่างๆในการส่งข้อมูลแบบอนุกรม RS-232 Standard.....10

ตารางที่ 2.2 Electronic Gear Ratio (G1/G2) ของเซอร์โวไดรเวอร์.....15

ตารางที่ 2.3 สายภายในประกอบไปด้วยสายสัญญาณทั้งหมด 50 เส้น.....20

ตารางที่ 2.4 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Move Relative.....30

ตารางที่ 2.5 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง Move Relative.....31

ตารางที่ 2.6 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Move Absolute.....32

ตารางที่ 2.7 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Move Absolute.....33

ตารางที่ 2.8 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : ORIGIN SEARCH.....34

ตารางที่ 2.9 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : ORIGIN SEARCH.....34

ตารางที่ 2.10 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Move Velocity.....35

ตารางที่ 2.11 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Move Velocity.....36

ตารางที่ 2.12 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Stop.....37

ตารางที่ 2.13 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Stop.....37

ตารางที่ 2.14 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Read Actual Position.....38

ตารางที่ 2.15 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Read Actual Position.....39

ตารางที่ 4.1 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 1 ผ่านหน้าจอ HMI.....80

ตารางที่ 4.2 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 2 ผ่านหน้าจอ HMI.....81

ตารางที่ 4.3 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 1 ผ่านหน้าจอ HMI.....83

ตารางที่ 4.4 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 2 ผ่านหน้าจอ HMI.....84

สารบัญรูปภาพ

รูปที่ หน้า

รูปที่ 2.1 หน้าจอทัชสกรีน รุ่น NB5Q-TW00B....4

รูปที่ 2.2 การสื่อสารแบบอนุกรม....7

รูปที่ 2.3 การส่งข้อมูลผ่านการสื่อสารแบบอนุกรม....8

รูปที่ 2.4 ตำแหน่งขาของ DB-9 และ DB-25....9

รูปที่ 2.5 ตัวควบคุม PLC รุ่น CP1H-X40DT-D....11

รูปที่ 2.6 Omron CP1W-CIF41....11

รูปที่ 2.7 เซอร์โวไดรเวอร์ และ เซอร์โวมอเตอร์....14

รูปที่ 2.8 กราฟแสดงพัลส์ที่จ่ายให้กับเซอร์โวไดรฟ์เวอร์....15

รูปที่ 2.9 กราฟแสดงพัลส์ที่จ่ายให้กับเซอร์โวไดรฟ์เวอร์....16

รูปที่ 2.10 หลักการควบคุมเซอร์โวมอเตอร์....17

รูปที่ 2.11 แสดงโครงสร้างของเซอร์โวมอเตอร์....17

รูปที่ 2.12 แสดงโครงสร้างและกราฟการทำงานของเอ็นโค้ดเดอร์....19

รูปที่ 2.13 สาย General Control Cable (R88A-CPG001S)....20

รูปที่ 2.14 แสดงการทำงานของ **Photoelectric Sensors** แบบ Diffuse Mode....22

รูปที่ 2.15 แสดงการทำงานของ **Photoelectric Sensors** แบบ Retroreflective Mode....23

รูปที่ 2.16 แสดงการทำงานของ **Photoelectric Sensors** แบบ Opposed Mode....23

รูปที่ 2.17 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง LOAD (LD), LOAD NOT (LD NOT)....25

รูปที่ 2.18 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND, AND NOT....25

รูปที่ 2.19 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OR, OR NOT....26

รูปที่ 2.20 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง **OUT**....26

รูปที่ 2.21 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND LOAD (AND LD), OR LOAD (OR LD)....27

รูปที่ 2.22 แสดงส่วนประกอบของคำสั่ง SHT(10)....28

รูปที่ 2.23 แสดงส่วนประกอบของคำสั่ง FB : Move Relative....29

รูปที่ 2.24 แสดงส่วนประกอบของคำสั่ง FB : Move Absolute....31

รูปที่ 2.25 แสดงส่วนประกอบของคำสั่ง FB : ORIGIN SEARCH....33

รูปที่ 2.26 แสดงส่วนประกอบของคำสั่ง FB : Move Velocity....35

รูปที่ 2.27 แสดงส่วนประกอบของคำสั่ง FB : Stop....36

รูปที่ 2.28 แสดงส่วนประกอบของ FB : Read Actual Position....38

รูปที่ 2.29 การเชื่อมต่อข้อมูลระหว่างโปรแกรมการใช้งานแม่ข่าย....41

สารบัญรูปภาพ(ต่อ)

รูปที่ หน้า

รูปที่ 2.30 การเชื่อมต่อข้อมูลระหว่างโปรแกรมการใช้งานบนลูกข่าย....41

รูปที่ 2.31 การอ่านข้อมูลจากแม่ข่ายหลายชุด....41

รูปที่ 2.32 การประมวลผลข้อมูลจากแม่ข่ายเดียวกัน....42

รูปที่ 2.33 แสดงการเชื่อมต่อคอมพิวเตอร์เข้ากับเครือข่ายโดยใช้ฮับ....43

รูปที่ 2.34 โครงสร้างด้านฮาร์ดแวร์ของระบบ SCADA....45

รูปที่ 2.35 โครงสร้างด้านซอร์ฟแวร์ของระบบ SCADA....46

รูปที่ 3.1 การเชื่อมต่อทั้งหมดในงานวิจัย....51

รูปที่ 3.2 การเชื่อมต่อระหว่างเซอร์โวมอเตอร์ กับ เครื่องควบคุมพีแอลซี....53

รูปที่ 3.3 การเชื่อมต่อระหว่างโฟโต้เซนเซอร์ กับ เครื่องควบคุมพีแอลซี....53

รูปที่ 3.4 การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ กับ เครื่องควบคุมพีแอลซี....54

รูปที่ 3.5 การกำหนดไอพีแอดเดรสในโปรแกรม Cx – Programmer ของเครื่องควบคุมพีแอลซี...55

รูปที่ 3.6 การเลือกไดร์ฟเวอร์อุปกรณ์ในการตั้งค่าสำหรับการเชื่อมต่อ OPC Server....55

รูปที่ 3.7 แสดงหน้าต่างหลักของโปรแกรม WinCC Explorer....56

รูปที่ 3.8 การ Configure Parameter ของเซอร์โวไดร์ฟ....57

รูปที่ 3.9 หน้าต่างสำหรับการเขียนโปรแกรมภาษาแลดเดอร์....59

รูปที่ 3.10 ภาพแสดงตำแหน่งของ Quick Client....60

รูปที่ 3.11 แสดงหน้าต่างของ Quick Client....60

รูปที่ 3.12 หน้าต่างสำหรับการสร้างกราฟิกหน้าจอทัชสกรีน....61

รูปที่ 3.13 หน้าต่างสำหรับการสร้างกราฟิกหน้าจอ HMI....61

รูปที่ 3.14 กราฟิกหน้าจอ HMI ที่ใช้ในงานวิจัย....62

รูปที่ 3.15 ภาพแสดงหน้าต่างสำหรับเลือกใช้ WinCC OnlineTrendControl....63

รูปที่ 4.1 แสดงหน้าเมนูหลัก....64

รูปที่ 4.2 แสดงหน้าการดำเนินการทางจุดกำเนิด (Origin Operation)....65

รูปที่ 4.3 หน้าต่าง Pop-up สำหรับกำหนดค่าตัวเลขต่างๆ....66

รูปที่ 4.4 แสดงหน้าการดำเนินการ JOG (JOG Operation)....67

รูปที่ 4.5 แสดงหน้าการดำเนินการ Relative (Relative Operation)....69

สารบัญรูปภาพ(ต่อ)

รูปที่ **หน้า**

รูปที่ 4.6 แสดงหน้าต่าง Pop-up สำหรับการเลือกกำหนดค่าต่างๆ.....70

รูปที่ 4.7 แสดงหน้าต่างสำหรับสั่งการเซอร์โวมอเตอร์ และแสดงผลด้วยโปรแกรม Graphic Designer ใน SIMATIC WinCC Explorer.....74

รูปที่ 4.8 แสดงชนิดของตัวแปรต่างๆ ใช้ในการหมุนของเซอร์โวมอเตอร์ และช่องว่างสำหรับกำหนดค่าตัวแปร.....76

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบัน เทคโนโลยีได้ถูกพัฒนาขึ้นอย่างรวดเร็ว มีความสะดวกสบาย และตอบสนอง ต่อความต้องการของผู้ใช้มากยิ่งขึ้น เทคโนโลยีนั้นเข้ามามีบทบาทอย่างมากต่อการดารงชีวิตของมนุษย์ ในปัจจุบันแม้แต่ในทางอุตสาหกรรมก็เช่นกัน เทคโนโลยีก็เข้ามามีบทบาทอย่างมาก เพื่อให้การผลิตในโรงงานหรือการควบคุมต่างๆ มีผลตอบสนองที่รวดเร็ว และสามารถทำงานได้ดีกว่าแรงงานของมนุษย์ โดยในขั้นตอนการผลิตของโรงงานอุตสาหกรรมนั้นจำเป็นจะต้องมีความเที่ยงตรงและความแม่นยำสูง เพื่อให้ชิ้นงานมีคุณภาพ และลดปริมาณชิ้นงานที่เสีย ไม่ว่าจะเป็นการป้อนชิ้นงาน การพับชิ้นงาน หรือเจาะชิ้นงานล้วนแต่ต้องใช้การทำงานของเครื่องมือที่มีความเที่ยงตรงและแม่นยำสูงทั้งสิ้น โดยเครื่องมือต่างๆเหล่านี้จะทำงานไม่ได้ ถ้าขาดโปรแกรมการควบคุมการทำงาน และรูปแบบที่จะให้เครื่องมือทำงาน รวมไปถึงการที่จะใช้เครื่องมือเพื่อควบคุมอะไร และใช้อะไรเป็นตัวขับเคลื่อนเครื่องมือ (เช่น แรง ไฟฟ้า นิวเมติก) ซึ่งในปัจจุบันก็มีเทคโนโลยีที่สามารถเขียนโปรแกรมเพื่อให้อุปกรณ์เหล่านี้ไปควบคุมการทำงานได้

จากความสนในการทำงานของเครื่องจักรในอุตสาหกรรม เราจึงเกิดแนวคิดว่าลักษณะที่ใช้ในการควบคุม และหลักการทำงานเป็นอย่างไร โดยการเขียนคำสั่งให้เครื่องควบคุมพีแอลซี (Programmable Logic Controller: PLC) เพื่อทำการควบคุมเซอร์โวมอเตอร์ ร่วมกับหน้าจอทัชสกรีน และ หน้าจอ HMI สามารถเฝ้ามองการเคลื่อนที่และทำการควบคุมเซอร์โวมอเตอร์ผ่านระบบเครือข่ายแบบ Ethernet โดยใช้ OPC Server เป็นสื่อกลางในการติดต่อสื่อสาร เพื่อทำให้อุปกรณ์ที่มีผู้ผลิตต่างกันสามารถติดต่อสื่อสารและทำงานร่วมกันได้ แสดงผลตอบสนองของระบบในแบบเรียลไทม์ และ เก็บข้อมูลเป็นกราฟเทรนด์เพื่อดูแนวโน้มของการเปลี่ยนแปลงของระบบได้อีกด้วย ซึ่งจะทำให้ผู้ปฏิบัติมีความสะดวกสบายมากขึ้นในการตรวจสอบสถานะการทำงานของเซอร์โวมอเตอร์ได้ตลอดเวลาภายในพื้นที่ที่มีเครือข่าย Ethernet

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เข้าใจหลักการการทำงานของเซอร์โวมอเตอร์ และ เซอร์โวไดร์ฟเวอร์
2. สามารถออกแบบและสร้างโปรแกรมแลดเดอร์ให้เครื่องควบคุมพีแอลซี (Programmable Logic Controller: PLC) เพื่อควบคุมระบบให้เป็นไปตามต้องการได้

3. สามารถสั่งงานการเคลื่อนที่ของเซอร์โวมอเตอร์ด้วยหน้าจอทัชสกรีน เพื่อ สังเกตการณ์การเคลื่อนที่ของเซอร์โวมอเตอร์ ผ่าน RS-232

4. สามารถสั่งงานการเคลื่อนที่ของเซอร์โวมอเตอร์ด้วยหน้าจอ HMI เพื่อ สังเกตการณ์การเคลื่อนที่ของเซอร์โวมอเตอร์ ผ่านระบบเครือข่าย Ethernet

5. เป็นแนวทางในการประยุกต์การใช้งานเซอร์โวมอเตอร์ในรูปแบบต่างๆ

6. เข้าใจหลักการเกี่ยวกับ OPC Server และสามารถใช้ OPC Server เป็นตัวกลางในการรับส่งข้อมูลระหว่างหน้าจอ HMI กับเครื่องควบคุมพีแอลซี ในการสื่อสารข้อมูล

1.3 ขอบเขตของปริญญานิพนธ์

1. สามารถเขียนโปรแกรมแลดเดอร์ให้กับเครื่องควบคุมพีแอลซี เพื่อให้สามารถทำงานตามที่ต้องการ

2. สามารถนำสัญญาณจากเครื่องควบคุมพีแอลซีไปควบคุมการเคลื่อนที่ของเซอร์โวมอเตอร์

3. สามารถรับค่าจากเครื่องควบคุมพีแอลซีมาแสดงผลในหน้าจอทัชสกรีนและหน้าจอ Human Machine Interface (HMI) ผ่านระบบเครือข่ายแบบ Ethernet

4. สามารถสั่งการเครื่องควบคุมพีแอลซีจากหน้าจอ HMI และหน้าจอทัชสกรีน

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของเซอร์โวมอเตอร์ อุปกรณ์ที่เกี่ยวข้อง และ หลักการทำงานของอุปกรณ์

2. ศึกษาการทำงานของหน้าจอทัชสกรีน

3. ศึกษาการใช้โปรแกรม CX-Programmer เพื่อทำการออกแบบและสร้างโปรแกรมแลดเดอร์ให้เครื่องควบคุมพีแอลซี (Programmable Logic Controller: PLC)

4. ศึกษาอุปกรณ์ที่ใช้เชื่อมต่อระหว่างคอมพิวเตอร์กับเครื่องควบคุมพีแอลซี ชนิด USB เช่น ข้อกำหนดต่างๆในการเชื่อมต่อ

5. ศึกษาข้อมูลเกี่ยวกับเครื่องควบคุมพีแอลซี และกำหนดสัญญาณขาเข้า และสัญญาณขาออก

6. ศึกษาการใช้โปรแกรม NB-Designer เพื่อทำการออกแบบและสร้างกราฟิกของหน้าจอทัชสกรีน

7. ศึกษาการกำหนดค่าอัตราทดเกียร์เพื่อทำการออกแบบการเคลื่อนที่ของบอลสกรู

8. สร้างโปรแกรมตามรูปแบบของการทำงานให้กับเครื่องควบคุมพีแอลซี

9. ทำการสร้างกราฟิกของหน้าจอทัชสกรีน ด้วยโปรแกรม NB-Designer โดยจะสามารถสั่งการการเคลื่อนที่ของเซอร์โวมอเตอร์ และแสดงผลทั้งสถานะและตำแหน่งของเซอร์โวมอเตอร์

10. ศึกษาการใช้โปรแกรม KepwareEX5 ซึ่งเป็น OPC Server เพื่อใช้เป็นตัวกลางในการรับส่งข้อมูลระหว่างเครื่องควบคุมพีแอลซีกับคอมพิวเตอร์ผ่านระบบเครือข่าย Ethernet

11. ศึกษาการใช้โปรแกรม SEMATIC WinCC Graphics Designer เพื่อทำการสร้างหน้าจอ HMI

12. ทำการสร้างหน้าจอ HMI ด้วยโปรแกรม SEMATIC WinCC Graphics Designer โดยจะสามารถสั่งการการเคลื่อนที่ของเซอร์โวมอเตอร์ และแสดงผลทั้งสถานะและตำแหน่งของเซอร์โวมอเตอร์

1.5 ประโยชน์ที่คาดว่าจะได้รับ

ในปริญญานิพนธ์นี้เป็นการควบคุมตำแหน่งของเซอร์โวมอเตอร์ โดยการสร้างโปรแกรมแลดเดอร์ผ่านทางคอมพิวเตอร์ให้กับเครื่องควบคุมพีแอลซีเพื่อควบคุมเซอร์โวมอเตอร์ให้ทำงานตามต้องการและแสดงผลการทำงานผ่านหน้าจอทัชสกรีนและหน้าจอมอนิเตอร์แบบเวลาจริง

ซึ่งสามารถนำปริญญานิพนธ์นี้มาอ้างอิง เพื่อจะได้เข้าใจถึงหลักการการทำงานของเซอร์โวมอเตอร์, *OPC Server, หน้าจอทัชสกรีน และเห็นภาพมากขึ้นสำหรับการทำงานในระบบ* อุตสาหกรรม

บทที่ 2
ทฤษฎีที่ใช้ในการออกแบบ

2.1 ฮาร์ดแวร์

2.1.1 โครงสร้างหน้าจอทัชสกรีน

ในปริญญานิพนธ์ฉบับนี้จะแสดงผลและสามารถสั่งการผ่านหน้าจอทัชสกรีน รุ่น NB5Q-TW00B โดยใช้หน่วยความจำข้อมูลของตัวควบคุมพีแอลซี มาใช้แสดงผล แสดงดังรูปที่ 2.1

รูปที่ 2.1 หน้าจอทัชสกรีน รุ่น NB5Q-TW00B

การเชื่อมต่อหน้าจอทัชสกรีน รุ่น NB5Q-TW00B เข้ากับตัวควบคุมพีแอลซี Omron รุ่น CP1H-X40DT-D โดยใช้ สาย RS-232 เชื่อมต่อเข้าหากัน

ช่องสื่อสาร Serial Port COM1 (female) รองรับการสื่อสารแบบ RS-232C เท่านั้น

- ช่องต่อ Serial Port COM1

สามารถดูได้จาก NB5Q-TW00B เป็นตัวอย่าง

COM1 เป็นช่องต่อแบบ 9-pin D-type socket port ช่องต่อนี้รองรับการสื่อสารและฟังก์ชั่น แบบ RS-232C ซึ่งสามารถเชื่อมต่อกับตัวควบคุม (คอนโทรลเลอร์) ที่ใช้ฟังค์ชั่น

ของ RS-232C ได้ อีกทั้งยังสามารถใช้โหลดโปรแกรมและแก้ไขบักได้อีกด้วย ข้อมูลของแต่ละขามีดังนี้

Pins	Signals	I/O	Functions
1	NC	-	-
2	SD	O	Sending data
3	RD	I	Receiving data
4	RS (RTS)	O	Request to send
5	CS (CTS)	I	Clear to send
6	DC+5V	-	DC+5V output (with a maximum current of 250mA)
7	NC	-	-
8	NC	-	-
9	SG	-	Signal ground

ช่องสื่อสาร Serial Port COM2 (female) รองรับการสื่อสารแบบ RS-232C/RS-422A/RS-485

- ช่องต่อ Serial Port COM2

สามารถดูได้จาก NB5Q-TW00B และ NB7W-TW00B เป็นตัวอย่าง COM1 เป็นช่องต่อแบบ 9-pin D-type socket port ช่องต่อนี้รองรับการสื่อสารและฟังค์ชั่น แบบ RS-232C, RS-422A/RS-485 ข้อมูลของแต่ละขามีดังนี้

Pins	Signals	I/O	Functions		
			RS-232C	RS-485	RS-422A
1	NC	I/O	-	-	Sending data (+)
2	SD	O	Sending data	-	-
3	RD	I	Receiving data	-	-
4	Terminal R1	-	-	Terminal resistor 1	
5	Terminal R2	-	-	Terminal resistor 2	
6	RDB+	I/O	-	Send/Receive data (+)	Receiving data (+)
7	SDA-	I/O	-	-	Sending data (-)
8	RDA-	I/O	-	Send/Receive data (-)	Receiving data (-)
9	SG	-	Signal ground		

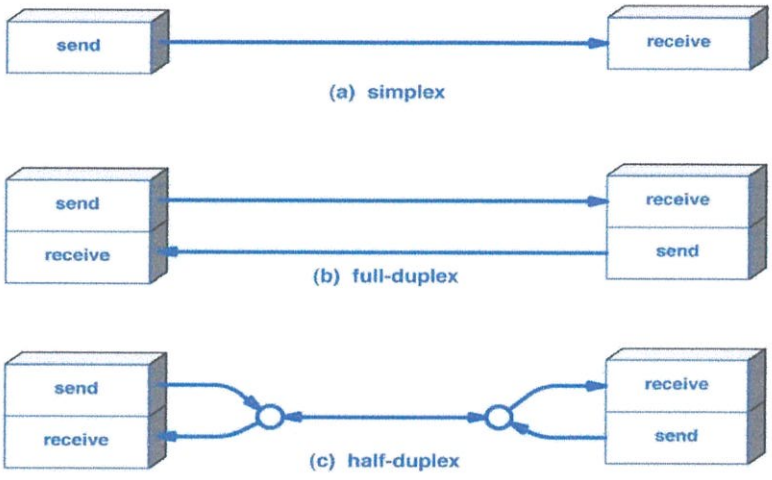
2.1.2 การสื่อสารข้อมูลแบบอนุกรม

ถ้าทำการกล่าวถึงเรื่องของการสื่อสารของข้อมูลแบบอนุกรมแล้วนั้น แสดงว่าจะต้องมีการสื่อสารข้อมูลแบบขนานด้วย ซึ่งการสื่อสารแบบขนานก็คือข้อมูลในทุกๆ บิทในแต่ละเวิร์ดจะถูกส่งออกไปพร้อมๆกัน ขึ้นอยู่กับว่าเวิร์ดดังกล่าวมีขนาดเท่าไร ทั่วไปก็คือ 1 ไบร์ หรือ 8 บิทนั้นเอง การส่งข้อมูลแบบขนานนี้จะมีข้อกำหนดทางด้านระยะทาง ซึ่งโดยทั่วไปจะส่งในระยะไม่เกิน 3 – 5 ฟุต เท่านั้น ทั้งนี้ยังขึ้นอยู่กับอัตราความเร็วที่ใช้ในการส่งข้อมูลด้วย ยิ่งอัตราในการส่งสูงก็จะได้ระยะทางที่มีความสั้นลง การส่งข้อมูลแบบขนานนั้นนิยมในระบบที่ต้องการความเร็วสูง

มากๆ แต่อุปกรณ์ไม่อยู่ห่างกันมากนัก ส่วนการส่งข้อมูลแบบอนุกรมนั้น ข้อมูลจะถูกทยอยส่งออกไปทีละบิทจนหมดครบทั้งเวิร์ด โดยส่งผ่านสัญญาณเพียงเส้นเดียวและในการใช้งานจริงนั้นจะต้องมีสายสัญญาณอีกเส้นเป็นสายสัญญาณกราวด์ (Ground) ดังนั้น เมื่อเราส่งข้อมูลในแบบอนุกรมเราจะสามารถใช้สายสัญญาณอย่างน้อยที่สุดเพียง 2 เส้น ในขณะที่ส่งข้อมูลแบบขนานจะต้องใช้อย่างน้อยเท่ากับจำนวนบิทบวกกับสายสัญญาณระดับแรงดัน Ground อีก 1 เส้น ที่สำคัญการส่งข้อมูลแบบอนุกรมนั้นจะสามารถส่งได้ไกลกว่า เช่น ถ้าส่งตามมาตรฐานของ RS-232 ที่จะกล่าวต่อไปในภายหลังจะสามารถส่งได้ไกลถึง 30 ถึง 40 ฟุต โดยไม่ต้องใช้อุปกรณ์ขับสัญญาณเพิ่มเติมแต่เดิมอย่างใด อย่างไรก็ตามในการส่งข้อมูลยังมีข้อกำหนดบางประการเพื่อให้ได้รับข้อมูลที่ถูกต้อง แม่นยำและมีความน่าเชื่อถือสูง จะต้องมีวิธีตรวจสอบความถูกต้องของข้อมูลได้ในระหว่างที่มีการสื่อสารกันอยู่ด้วย ซึ่งจะได้กล่าวต่อไป

2.1.2.1 การส่งข้อมูลแบบซิมเพล็กซ์ (Simplex) และ ดูเพล็กซ์ (Duplex)

ในการสื่อสารไม่ว่าจะเป็นการสื่อสารข้อมูล หรือเป็นการสื่อสารทั่วๆไปนั้น ย่อมจะต้องประกอบด้วยผู้รับและผู้ส่งแน่นอนตายตัวอยู่ตลอดเวลา เช่น การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์กับเครื่องพิมพ์ เป็นต้น การสื่อสารของอุปกรณ์ที่มีผู้รับและผู้ส่งตายตัวนั้น เราเรียกว่าการสื่อสารแบบซิมพล็กซ์ กล่าวคือ การสื่อสารเป็นไปในลักษณะทิศทางเดียวตลอดเวลาซึ่งจะมีที่ใช้ไม่มากนัก การสื่อสารโดยทั่วไปนั้นจะเป็นแบบดูเพล็กซ์ คือมีทิศทางการสื่อสารเป็นแบบสองทิศทางมีทั้งไปและกลับ โดยการสื่อสารในลักษณะดูเพล็กซ์นั้นแบ่งออกได้เป็น 2 ชนิด คือ แบบฮาร์ฟดูเพล็กซ์ (Half Duplex) นิยมเขียนย่อกันว่า HEX ซึ่งจะมีทิศทางในการสื่อสารในลักษณะที่ผลัดกันเป็นผู้ส่งและผู้รับพร้อมกัน และแบบฟูลดูเพล็กซ์ (Full Duplex) นิยมเขียนย่อว่า FDX จะมีทิศทางการสื่อสารในลักษณะสัญญาณรับสัญญาณหนึ่ง และสัญญาณส่งอีกทิศทางหนึ่ง หรือกล่าวได้อีกนัยหนึ่งว่า สัญญาณรับและส่งจะมีสายตัวนำสัญญาณแยกออกจากันโดยเด็ดขาด

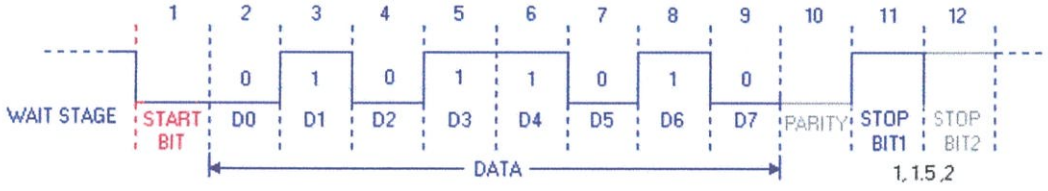


รูปที่ 2.2 การสื่อสารแบบอนุกรม

2.1.2.2 โปรโตคอลของการสื่อสารแบบอนุกรม

เมื่อพิจารณาการส่งข้อมูลแบบอนุกรมให้ดีจะพบว่ามีปัญหาหนึ่งที่เกิดขึ้นอยู่เสมอ ก็คือ การตัดสินว่าข้อมูลที่ได้รับนั้นมีจุดเริ่มต้นที่ใด ดังนั้น จึงมีการกำหนดข้อตกลงในการสื่อสารขึ้นเพื่อแก้ปัญหานี้ โดยข้อตกลงดังกล่าวจะเรียกว่า โปรโตคอล (Protocol) ของการสื่อสารข้อมูลแบบอนุกรม โดยสามารถที่จะแบ่งออกเป็น 2 ประเภทใหญ่ๆ ด้วยกันคือ โปรโตคอลสื่อสารข้อมูลแบบซิงโครนัส (Synchronous) และแบบอะซิงโครนัส (Asynchronous) โดยการสื่อสารข้อมูลแบบซิงโครนัสนี้ ข้อมูลจะถูกทำการส่งออกมาอย่างสม่ำเสมอและช่วงเวลาระหว่างบิทและเวิร์ด จะมีค่าที่เท่ากันเสมอ ส่วนการสื่อสารข้อมูลแบบอะซิงโครนัสจะเป็นหัวใจของการสื่อสารข้อมูลผ่านทางสายโทรศัพท์ ในปัจจุบันซึ่งการสื่อสารแบบนี้ช่วงระยะเวลาระหว่างบิทจะมีค่าที่เท่ากันกับแบบซิงโครนัส แต่จะมีระยะห่างระหว่างเวิร์ดนั้นแตกต่างกันไปเป็นวินาที นาที ชั่วโมง หรือ วัน เป็นต้น ขึ้นอยู่กับฝ่ายรับสามารถรอได้หรือไม่เท่านั้น เมื่อไม่มีข้อกำหนดทางด้านระยะเวลาระหว่างเวิร์ดแล้ว ทางผู้ส่งและผู้รับจะเข้าใจกันตรงกันได้อย่างไร ว่าที่ใดคือจุดที่เริ่มต้นและจุดที่สิ้นสุดของแต่ละเวิร์ด เพื่อแก้ไขปัญหาตรงนี้ จึงมีการกำหนดข้อตกลงเกี่ยวกับรูปแบบของข้อมูลที่ใช้ในการทำการส่งให้ทางผู้รับสามารถเข้าใจว่าจุดใดที่เป็นจุดเริ่มต้นของเวิร์ด ข้อกำหนดดังกล่าวกำหนดให้แต่ละเวิร์ดจะต้องขึ้นต้นด้วยบิทที่ เรียกว่า บิทเริ่มต้น (Start Bit) ซึ่งจะต้องมีข้อมูลเป็นลอจิก 0 เสมอ จากนั้นตามด้วยบิทข้อมูลที่ต้องการส่งมีความยาว 5 ถึง 8 บิท ถัดจากบิทข้อมูลก็จะเป็นบิทพาริตีบิท ซึ่งทำหน้าที่เป็นบิทสำหรับตรวจสอบความถูกต้องของข้อมูลที่ได้รับว่ามีความถูกต้องหรือไม่ บิทพาริตีนี้มี 2 ประเภท คือ อีเวนพาริตี (Even Parity) ซึ่งจะกำหนดจำนวนบิทที่เป็นลอจิก 1 ในบิทที่เป็นข้อมูลมีจำนวนเป็นคี่ ในการส่งข้อมูลบางครั้งอาจจะไม่มีการใช้บิทพาริตีก็ได้ ถ้าหากการสื่อสารในครั้งนั้นมีความน่าเชื่อถือสูง มีสัญญาณรบกวนต่ำเป็นการเพิ่มความเร็วในการสื่อสารได้ด้วยบิทสุดท้าย

ในรูปแบบก็คือ บิทสุดท้าย (Stop Bit) ทำหน้าที่บอกทางผู้รับว่าในขณะนี้ข้อมูลที่ทางผู้รับได้รับนั้นครบเวิร์ดแล้วขอให้เตรียมชุดรับเวิร์ดต่อไปได้ บิทสุดท้ายนี้ถูกกำหนดให้เป็นลอจิก 1 บิท หรือ 2 บิท ก็ได้จากรูปแบบดังกล่าวจะเห็นว่าเรามีรูปแบบสำหรับการสื่อสารมากมาย เช่น 5E1 (5Databit,Even Parity,1 Stop Bit),7E1(7Databit,Even Parity,1 Stop Bit) และ 8E1 (8Databit,Even Parity,1 Stop Bit) เป็นต้น ในการใช้งานทั่วไปเรานิยมใช้กันอยู่เพียง 2 รูปแบบคือ 7E1 และ 8E1 จะเลือกรูปแบบใดขึ้นอยู่กับสภาพของสายส่งสัญญาณว่ามีสัญญาณรบกวนมากเพียงใด ถ้าหากสายส่งมีสัญญาณรบกวนมากก็ควรจะใช้ 7E1 แต่ถ้าสายส่งสัญญาณมีสภาพดีสัญญาณรบกวนต่ำการใช้ 8E1 จะเร็วกว่า เป็นต้น ทั้งนี้ต้องมีการตกลงกันล่วงหน้าระหว่างทางผู้รับและผู้ส่งว่าจะใช้รูปแบบใดในการสื่อสาร ลักษณะของข้อมูลที่ถูกส่งออกไปจะมีลักษณะดังรูปที่ 2.3



รูปที่ 2.3 การส่งข้อมูลผ่านการสื่อสารแบบอนุกรม

2.1.2.3 มาตรฐานสัญญาณอนุกรมแบบ RS232C

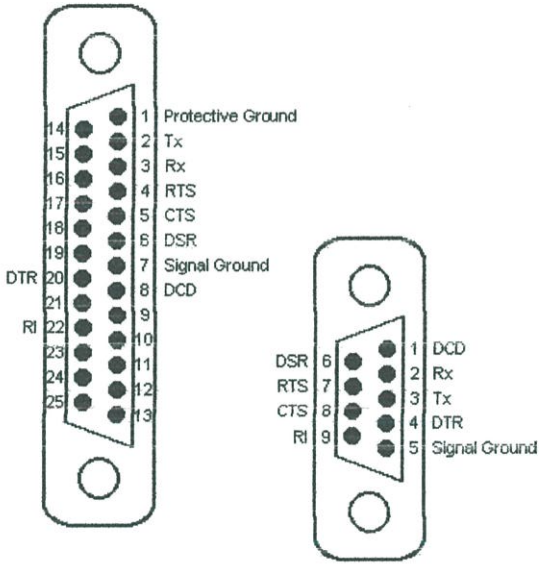
มาตรฐานการสื่อสารข้อมูลแบบอนุกรมกำหนดโดย EIA (Electronics Industries Association) มาตรฐาน RS232C ได้ถูกตีพิมพ์ในปี คศ. 1969 เริ่มต้นจากความต้องการที่จะกำหนดมาตรฐานการเชื่อมต่อระหว่างคอมพิวเตอร์กับโมเด็มในสมัยนั้น

- ตัวอักษร RS แทน “Recommend Standard”
- 232 แทนหมายเลขของมาตรฐาน
- ตัวอักษร C แทนให้รู้ว่ามาตรฐานได้รับการแก้ไขกี่ครั้ง
- ลักษณะสมบัติทางไฟฟ้าของการอินเทอร์เฟสแบบ RS232
- ถูกออกแบบให้ใช้กับอุปกรณ์พวกสัญญาณ Discrete
- ใช้การอินเทอร์เฟสแบบ Unbalanced
- ในแต่ละวงจรใช้ลวดนำในการนำสัญญาณ 1 เส้น และมีสายกราวด์รวมทุกวงจรอีก 1 เส้น
- อัตราเร็วในการส่งข้อมูลมีค่า < 20 กิโลบิทต่อวินาที (Kbps)
- ระยะทางสูงสุดในการส่งข้อมูลมีค่า < 15 เมตร

- ทำให้เกิด Crosstalk ที่มีค่ามาก

2.1.2.4 พอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรม (Serial Communication Device) หรือเรียกว่า "Serial Port" เครื่องคอมพิวเตอร์โดยปกติจะมีพอร์ตชนิดนี้อยู่แล้ว 2 พอร์ต คือ พอร์ตขนาด 9 ขา (9-pins) มีรูปร่างเหมือนสี่เหลี่ยมคางหมู มีเข็มยื่นออกมา 9 เข็ม เรียกหัวชนิดนี้ว่า "DB-9 Connector Male Type" อีกชนิดหนึ่งคือ พอร์ตขนาก 25 ขา (25-pins) มีรูปร่างเช่นเดียวกันกับแบบ 9 ขา แต่มีขนาดที่ยาวกว่า เรียกหัวชนิดนี้ว่า "DB-25 Connector Male Type"



รูปที่ 2.4 ตำแหน่งขาของ DB-9 และ DB-25

ตารางที่ 2.1 ขาสัญญาณต่างๆในการส่งข้อมูลแบบอนุกรม RS-232 Standard

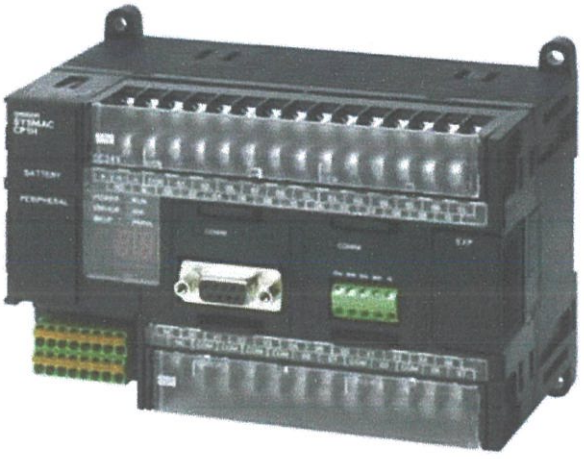
DB-9 pins	DB-25 pins	Function
1	8	Carrier Detect
2	3	Received Data
3	2	Transmitted Data
4	20	Data Terminal Ready
5	7	Signal Ground
6	6	Data Set Ready
7	4	Request to Send
8	5	Clear to Send
9	22	Ring Indicator

มาตรฐานของการรับส่งข้อมูลแบบอนุกรม (RS-232) นี้ได้กำหนดขึ้นเพื่อให้คอมพิวเตอร์ต่างยี่ห้อกัน หรืออุปกรณ์ต่อพ่วงแต่ละชนิดรับส่งข้อมูลกันได้เมื่อทำตามมาตรฐาน ไม่สนใจว่าอุปกรณ์หรือคอมพิวเตอร์นั้นจะผลิตมาจากที่ใด

2.1.3 โครงสร้างพื้นฐานของตัวควบคุมพีแอลซี

ในโปรแกรมเมเบิลลอจิกคอลโทรลเลอร์ (Programmable logic Control : PLC) เป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการทำงานต่างๆ โดย Microprocessor เป็นมันสมองสั่งการที่สำคัญ ตัวควบคุมจะมีส่วนที่เป็นอินพุทและเอาท์พุทที่สามารถต่อออกไปใช้งานได้ทันที ตัวตรวจวัดหรือสวิทตช์ต่างๆ จะต่อเข้ากับอินพุท ส่วนเอาท์พุทจะใช้ต่อออกไปควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรที่เป็นเป้าหมาย เราสามารถสร้างวงจรหรือแบบของการควบคุมได้โดยการป้อนเป็นโปรแกรมคำสั่งเข้าไปในตัวควบคุมพีแอลซี นอกจากนี้ยังสามารถใช้งานร่วมกับอุปกรณ์อื่น เช่น เครื่องอ่านบาร์โค๊ด (Barcode Reader), เครื่องพิมพ์ (Printer) ซึ่งในปัจจุบันนอกจากตัวควบคุมพีแอลซี จะใช้งานแบบเดี่ยว (Stand alone) แล้วยังสามารถต่อ ตัวควบคุมพีแอลซี หลายๆ ตัวเข้าด้วยกัน (Network) เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพมากยิ่งขึ้นด้วยจะเห็นได้ว่าการใช้งาน ตัวควบคุมพีแอลซี มีความยืดหยุ่นมากดังนั้นในโรงงานอุตสาหกรรมต่างๆ จึงเปลี่ยนมาใช้ตัวควบคุมพีแอลซีมากขึ้น

ในส่วนนี้จะขอแสดงถึงโครงสร้างและคุณลักษณะเฉพาะตัวพื้นฐานโดยทั่วไปของ ตัวควบคุมพีแอลซี Omron รุ่น CP1H-X40DT-D ซึ่งเป็นยี่ห้อหนึ่งที่มีมาตรฐานและสามารถใช้งานได้อย่างหลากหลายโดยพิจารณาได้ดังรูปที่ 2.5



รูปที่ 2.5 ตัวควบคุมพีแอลซี รุ่น CP1H-X40DT-D

ตัวควบคุมพีแอลซี Omron รุ่น CP1H-X40DT-D สามารถเชื่อมต่อผ่านระบบ Ethernet ได้ โดยการติดตั้งอุปกรณ์เสริม เพื่อทำการเชื่อมต่อสาย LAN ด้วย Omron CP1W-CIF41 แสดงดังรูปที่ 2.6

รูปที่ 2.6 Omron CP1W-CIF41

2.1.4 IEEE 802.3: Ethernet

Ethernet นับเป็นต้นกำเนิดของเทคโนโลยี LAN เนื่องจาก LAN ส่วนมากหรือเกือบทั้งหมดในปัจจุบันใช้ พื้นฐานของเทคโนโลยีนี้ คุณลักษณะเฉพาะในการทำงานของ Ethernet คือการทำงานแบบที่เรียกว่า การเข้าใช้ระบบเครือข่ายโดยวิธีช่วงชิง หรือ CSMA/CD (Carrier Sense Multiple Access with Collision Detection) โดยมีหลักการทำงานดังนี้

1. ก่อนที่สถานีงานของผู้ใช้จะส่งข้อมูลออกไปยังเครือข่าย จะต้องมีการแจ้งออกไปก่อนเพื่อตรวจสอบดูว่ามีสัญญาณพาหะของผู้ใช้รายอื่นอยู่ในสายหรือไม่
2. เมื่อไม่พบสัญญาณของผู้ใช้อื่น จึงจะเริ่มส่งข้อมูลออกไปได้

3. หากตรวจพบสัญญาณพาหะของผู้ใช้รายอื่นอยู่ จะต้องรอจนกว่าสายจะว่างถึงจะส่งข้อมูลได้

4. ในกรณีที่เกิดปัญหาในการตรวจสอบสัญญาณพาหะ ซึ่งอาจเนื่องมาจากระยะทางของสถานีงานอยู่ห่างกันมาก อาจจะเกิดการชนกันของข้อมูลขึ้นได้ ในกรณีนี้ให้ทั้งทุกๆ สถานีหยุดการส่งข้อมูลขณะนั้น

5. แต่ละสถานีจะทำการสุ่มช่วงระยะเวลาในการรอ เพื่อทำการส่งข้อมูลออกไปใหม่เพื่อไม่ให้มีการชนกันเกิดขึ้นอีก

6. หากยังมีเหตุการณ์ชนกันเกิดขึ้นอีก ก็จะต้องหยุดรอโดยเพิ่มช่วงระยะเวลาในการสุ่มเป็นสองเท่าเพื่อให้ลดโอกาสการชนกันลงและส่งข้อมูลออกไปใหม่ และทำซ้ำเช่นนี้ จนกว่าข้อมูลจะถูกส่งออกไปได้อย่างสมบูรณ์

แม้ว่าระบบ CSMA/CD ดูเหมือนจะเป็นวิธีจัดระเบียบการส่งสัญญาณในระบบเครือข่ายที่ไม่เรียบร้อยนัก แต่ก็ทำงานได้ผลเป็นอย่างดี แต่เมื่อมีจำนวนโหนดบนเครือข่ายมากขึ้น ก็จะทำให้ความน่าจะเป็นในการปะทะกันของ ข้อมูลเพิ่มมากขึ้นด้วย ซึ่งจะส่งผลให้เครือข่ายทำงานช้าลงอย่างหลีกเลี่ยงไม่ได้

ระบบเครือข่าย Ethernet ยังสามารถแบ่งประเภทได้อีก ตามความเร็วและชนิดของสายเคเบิล ดังนี้

- IEEE 802.3 10Base5 (Thick Ethernet)

เป็นระบบเครือข่ายแบบบัส ใช้สายโคแอ็กเชียลอย่างหนา (3/8 นิ้ว) สามารถส่งข้อมูลได้ที่อัตราเร็ว 10 เมกะบิตต่อวินาที ในระยะทางสูงสุดไม่เกิน 500 เมตร และเนื่องจากสายโคแอ็กเชียลอย่างหนาสามารถนำสัญญาณไปได้ไกลกว่าจึงมักถูกใช้เป็นแบคโบน (Backbone) ของระบบเครือข่าย

- IEEE 802.3 10Base2 (Thin Ethernet)

เป็นระบบเครือข่ายแบบบัส ใช้สายโคแอ็กเชียลอย่างบาง (3/16 นิ้ว) สามารถติดตั้งได้ง่ายกว่าแบบแรกและราคาต่ำกว่า สามารถส่งข้อมูลได้ที่อัตราเร็ว 10 เมกะบิตต่อวินาทีในระยะทางสูงสุดไม่เกิน 200 เมตร

- IEEE 802.3 10BaseT (Twisted-pair Ethernet)

เป็นระบบที่จัดการเชื่อมต่อโหนดต่างๆ เข้ากับ Hub เป็นรูปแบบดาว ใช้สายคู่พันเกลียวโดยอาจเป็นแบบไม่มีสิ่งห่อหุ้ม(Unshielded Twisted-pair) หรือแบบมีสิ่งห่อหุ้ม (Shielded Twisted-pair)ก็ได้ มีหัวเชื่อมต่อเป็นแบบ RJ-45 มีลักษณะคล้ายปลั๊กโทรศัพท์ สามารถส่งข้อมูลได้ที่อัตราเร็ว 10 เมกะบิตต่อวินาที โดยมีความยาวของสายระหว่างสถานีงานกับ Hub ไม่เกิน 100 เมตร

- IEEE 802.3u 100BaseX (Fast Ethernet)

มีระบบการเชื่อมต่อแบบดาว สามารถรับส่งข้อมูลได้ที่อัตราเร็ว 100 เมกะบิตต่อวินาทีแบ่งเป็น 3 ประเภท ได้แก่

1. 100BaseT4 ใช้สายคู่พันเกลียวจำนวน 4 คู่
2. 100BaseTX ใช้สายคู่พันเกลียวจำนวน 2 คู่
3. 100BaseFX ใช้เคเบิลใยแก้วนำแสง

2.1.5 TCP/IP

TCP/IP (Transmission Control Protocol/Internet Protocol) เป็นระบบโปรโตคอล การสื่อสารพื้นฐานของระบบอินเตอร์เน็ต มันสามารถใช้เป็น โปรโตคอลในการสื่อสารภายใน เครือข่ายส่วนบุคคล เรียกว่า intranet และ extranet เมื่อมีการติดต่อโดยตรงกับ internet เครื่องคอมพิวเตอร์จะได้รับการคัดลอกโปรแกรม TCP/IP เช่นเดียวกับคอมพิวเตอร์อื่น ๆ เพื่อทำให้ส่งข้อความขอรับสารสนเทศ

TCP/IP เป็นโปรแกรม 2 เลเยอร์ TCP (Transmission Control Protocol) เป็นเลเยอร์ที่สูงกว่า ทำหน้าที่จัดการแยกข้อความหรือไฟล์แลปรกอบให้เหมือนเดิม IP (Internet Protocol) เป็นเลเยอร์ที่ต่ำกว่า ทำหน้าที่จัดการส่วนของที่อยู่ของแต่ละชุดข้อมูล เพื่อทำให้มีปลายทางที่ถูกต้อง เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็น Gateway บนเครือข่ายจะตรวจที่อยู่นี้เพื่อหาจุดหมายในการส่งข้อความ ชุดข้อมูลอาจจะใช้เส้นทางไปยังปลายทางต่างกัน แต่ทั้งหมดจะได้รับการประกอบใหม่ที่ปลายทาง

TCP/IP ใช้ในแบบ client/server ในการสื่อสาร (ระหว่างคอมพิวเตอร์) ซึ่งผู้ใช้คอมพิวเตอร์ (client) เป็นผู้ขอและการบริการได้รับจากคอมพิวเตอร์เครื่องแม่ข่ายในระบบเครือข่าย การสื่อสารของ TCP/IP เป็นแบบจุดต่อจุด (point -to- point) หมายความว่าการสื่อสารแต่ละครั้งเกิดจากจุดหนึ่ง (เครื่อง host เครื่องหนึ่ง) ไปยังจุดอื่นหรือเครื่อง host เครื่องอื่นในเครือข่าย TCP/IP และโปรแกรมประยุกต์ระดับสูงอื่น ที่ใช้ TCP/IP สามารถเรียกว่า "Stateless" เพราะการขอแต่ละ client ได้รับการพิจารณาเป็นการขอใหม่โดยไม่สัมพันธ์กับการขอเดิม (แต่แตกต่างจากการสนทนาทางโทรศัพท์) การที่เป็นพาร์ทของเครือข่ายอิสระแบบ "Stateless" ดังนั้นทุกคนสามารถใช้พาร์ทได้อย่างต่อเนื่อง (หมายเหตุ เลเยอร์ของ TCP จะไม่ "Stateless" ถ้ายังทำการส่งข้อความใดข้อความหนึ่งจะทำการส่งจนกระทั่งชุดข้อมูลนั้นได้รับครบชุด)

2.1.6 เซอร์โวไดรเวอร์ (Servo Driver)

ในปัจจุบันจะรวมส่วนที่เป็นเซอร์โวคอนโทรลเลอร์และ Positioning controller ไว้ด้วยกันซึ่งเซอร์โวคอนโทรลเลอร์จะทำหน้าที่เป็นตัวจ่ายไฟให้กับมอเตอร์ ส่วน Positioning

controller จะทำหน้าที่ควบคุมตำแหน่งการเคลื่อนที่ของมอเตอร์โดยรับคำสั่งมาจากอุปกรณ์ภายนอกเช่น ตัวควบคุมพีแอลซี เป็นต้น

ในปริญญานิพนธ์ฉบับนี้ จะใช้เซอร์โวไดรเวอร์ G5-series รุ่น R88D-KP01H ในการขับเคลื่อนเซอร์โวมอเตอร์

รูปที่2.7 เซอร์โวไดรเวอร์ และ เซอร์โวมอเตอร์

2.1.6.1 การตั้งค่าสำหรับเซอร์โวไดรเวอร์ G5-series รุ่น R88D-KP01H

สำหรับ G5-series รุ่น R88D-KP01H สามารถทำการปรับตั้งค่าได้โดยกดปุ่มที่หน้าปัด หรือใช้โปรแกรม Cx-Driver ในการปรับเปลี่ยนพารามิเตอร์

ผู้ใช้สามารถกำหนดค่าความละเอียดของระบบ ให้เป็นค่าตามต้องการของแต่ละงานได้ โดยการกำหนดค่าลงไปในพารามิเตอร์ ดังนี้

- ถ้าในกรณี Pn008 คือค่า Electronic Gear Integer Setting ป้อนค่าจำนวนพัลส์ต่อรอบที่ต้องการซึ่งเป็นตัวเลขจำนวนเต็มลงใน Pn008 ได้เลย เซอร์โวมอเตอร์ก็จะยึดเอาค่าใน Pn008 ไปใช้เป็นค่าความละเอียดของการเคลื่อนที่
- ถ้าในกรณี Pn008 มีค่าเท่ากับ 0 จะต้องมีการคำนวณค่า Electronic Gear Ratio

ตารางที่2.2 Electronic Gear Ratio (G1/G2) ของเซอร์โวไดรเวอร์

Parameter No.	Value	Condition	Range
Pn009	G1	0.01 ≤ G1/G2 ≤ 100	0 - 2^30
Pn010	G2		1 - 2^30

2.1.6.2 ชนิดของอินพุตควบคุมสำหรับเซอร์โวไดรฟ์เวอร์

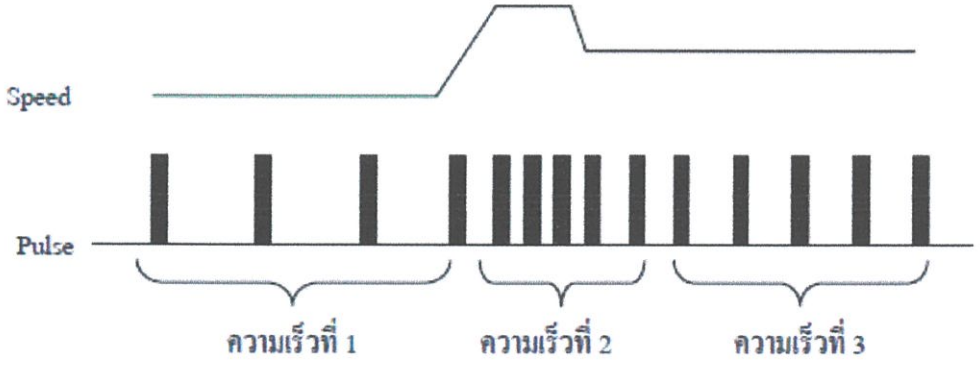
อินพุตของเซอร์โวไดรฟ์เวอร์ ที่ต่อใช้งานกับตัวควบคุมพีแอลซี มีอยู่ด้วยกันหลายแบบแต่ที่นิยมมากที่สุด คือ แบบ Pulse Train และ Linear (หรือ Analog) ซึ่งทั้ง 2 แบบจะเหมาะสมกับงานที่แตกต่างกัน แบบ Pulse train จะเหมาะกับงานควบคุมตำแหน่ง เช่น Feed-to-Cut และ Pick & Place เป็นต้น ส่วนแบบ Linear จะใช้กับงานที่เคลื่อนที่เป็นเส้นโค้งหรือวงกลม ในที่นี้เราจะเน้นการใช้เซอร์โวไดรฟ์เวอร์ที่รับสัญญาณอินพุตแบบ Pulse train เพื่อให้ง่ายในการทำความเข้าใจสำหรับผู้เริ่มต้น

- PULSE TRAIN CONTROL

ตัวควบคุมพีแอลซีจะส่งสัญญาณเป็นพัลส์ให้กับเซอร์โวไดรฟ์เวอร์จากนั้นไดรฟ์เวอร์จะควบคุมการหมุนของมอเตอร์ให้ได้ตามการสั่งงานของตัวควบคุมพีแอลซีระยะในการเคลื่อนที่หรือการหมุนของเซอร์โวมอเตอร์จะขึ้นอยู่กับจำนวนพัลส์ที่ตัวควบคุมพีแอลซีส่งให้ ขณะเดียวกันความเร็วของเซอร์โวมอเตอร์จะขึ้นอยู่กับความเร็วหรือความถี่ของพัลส์ที่ส่งมาจากตัวควบคุมพีแอลซี ดังแสดงในรูปที่ 2.8

ระยะการหมุน α จำนวนพัลส์

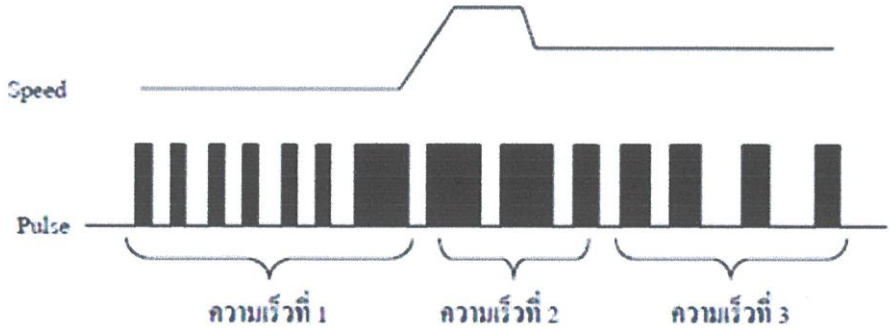
ความเร็ว α ความถี่ของพัลส์



รูปที่ 2.8 กราฟแสดงพัลส์ที่จ่ายให้กับเซอร์โวไดรฟ์เวอร์

- PULSE WIDTH MODULATION (PWM)

การทำงานจะคล้ายๆ กับ Pulse Train คือ ระยะทางการหมุนจะขึ้นอยู่กับจำนวนพัลส์ แต่ความเร็วของมอเตอร์จะขึ้นอยู่กับความกว้างของพัลส์ดังแสดงในรูปที่ 2.9



รูปที่ 2.9 กราฟแสดงพัลส์ที่จ่ายให้กับเซอร์โวไดรฟ์เวอร์

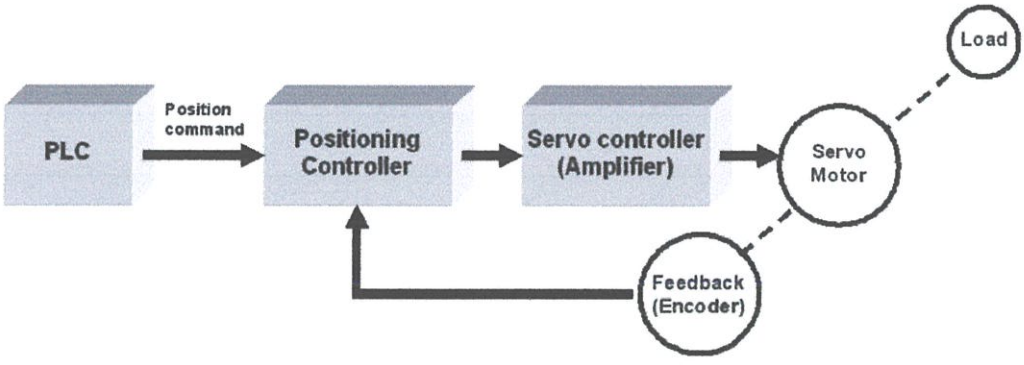
- LINEAR

การควบคุมแบบนี้จะแตกต่างจากทั้งสองแบบที่กล่าวมาข้างต้น เพราะ ตัวควบคุมพีแอลซี จะส่งสัญญาณอนาลอกให้กับเซอร์โวไดรฟ์เวอร์ ซึ่งไดรฟ์เวอร์จะต้องเป็นชนิดที่สามารถรับสัญญาณควบคุมเป็นอนาลอกได้ เช่น ± 10Vdc

2.1.7 เซอร์โวมอเตอร์ (Servo Motor)

ระบบควบคุมเซอร์โวมอเตอร์เบื้องต้น โดยปกติระบบควบคุมเซอร์โวมอเตอร์จะมีองค์ประกอบหลักในการทำงานดังนี้

- มอเตอร์: ทำหน้าที่ขับโหลดมีทั้งชนิดที่มีเบรคและไม่มีเบรค
- เอ็นโค้ดเดอร์ : ติดอยู่กับตัวมอเตอร์เพื่อทำหน้าที่ป้อนกลับการเคลื่อนที่ของมอเตอร์ ดังนั้นเราสามารถทราบตำแหน่งและความเร็วของการหมุนได้จากเอ็นโค้ดเดอร์นี้
- เซอร์โวไดร์เวอร์ : ในปัจจุบันจะรวมส่วนที่เป็นเซอร์โวคอนโทรลเลอร์และ Positioning controller ไว้ด้วยกันซึ่งเซอร์โวคอนโทรลเลอร์จะทำหน้าที่เป็นตัวจ่ายไฟให้กับมอเตอร์ ส่วน Positioning controller จะทำหน้าที่ควบคุมตำแหน่งการเคลื่อนที่ของมอเตอร์โดยรับคำสั่งมาจากอุปกรณ์ภายนอกเช่น ตัวควบคุมพีแอลซี เป็นต้น
- ตัวควบคุมพีแอลซี : จะทำหน้าในการส่งคำสั่งไปยังเซอร์โวไดร์เวอร์ในรูปแบบต่างๆ เช่น อนาลอกและพัลส์ เป็นต้น จากนั้นเซอร์โวไดร์เวอร์จะควบคุมให้มอเตอร์หมุนให้ได้ตำแหน่งและความเร็วตามที่ได้รับคำสั่ง

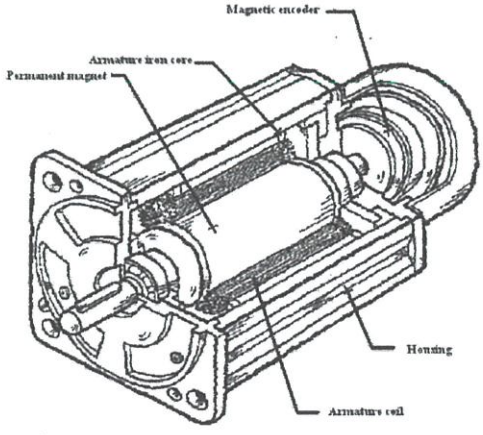


รูปที่ 2.10 หลักการควบคุมเซอร์โวมอเตอร์

จากรูป 2.10 การทำงานจะเริ่มโดย ตัวควบคุมพีแอลซีส่ง Position command ซึ่งเป็นสัญญาณพัลส์ หรือ อนาลอกให้กับ Position controller ที่อยู่ในเซอร์โวไดรฟ์เวอร์ จากนั้น Position controller จะสั่งให้ Amplifier จ่ายกระแสไฟให้กับมอเตอร์เพื่อทำให้มอเตอร์หมุนให้ได้ความเร็วและระยะตามคำสั่งเอ็นโค้ดเดอร์ที่ติดอยู่กับมอเตอร์จะทำหน้าที่ป้อนกลับข้อมูลระยะและความเร็วในการหมุนกลับไปที่ Position controller ซึ่งมันจะมี Counter ทำหน้าที่เปรียบเทียบกับคำสั่งที่ได้รับจากตัวควบคุมพีแอลซี ถ้ายังมีความแตกต่างกันมันส่งสัญญาณไปที่ Amplifier เพื่อสั่งให้มอเตอร์หมุนให้ได้ระยะและความเร็วตามต้องการ

2.1.7.1 โครงสร้างของเซอร์โวมอเตอร์

เซอร์โวมอเตอร์ของออมรอนเป็นประเภท AC Servo Motor แบบ Synchronous Servo Motor มีลักษณะของโครงสร้างดังนี้



รูปที่ 2.11 แสดงโครงสร้างของเซอร์โวมอเตอร์

ส่วนของสเตเตอร์เป็นขดลวดพันในร่องสล็อต ส่วนโรเตอร์เป็นแม่เหล็กถาวร ดังนั้นความสัมพันธ์ของค่าพารามิเตอร์ ต่างๆ จะคล้าย DC motor และมอเตอร์ประเภทนี้ไม่มีแปรงถ่าน

บางทีจึง เรียกว่า เซอร์โวมอเตอร์แบบ DC Brushless ส่วนตัวเอ็นโค้ดเดอร์จะต่ออยู่กับเพลาเดียวกับโรเตอร์

2.1.7.2 Ball Screw

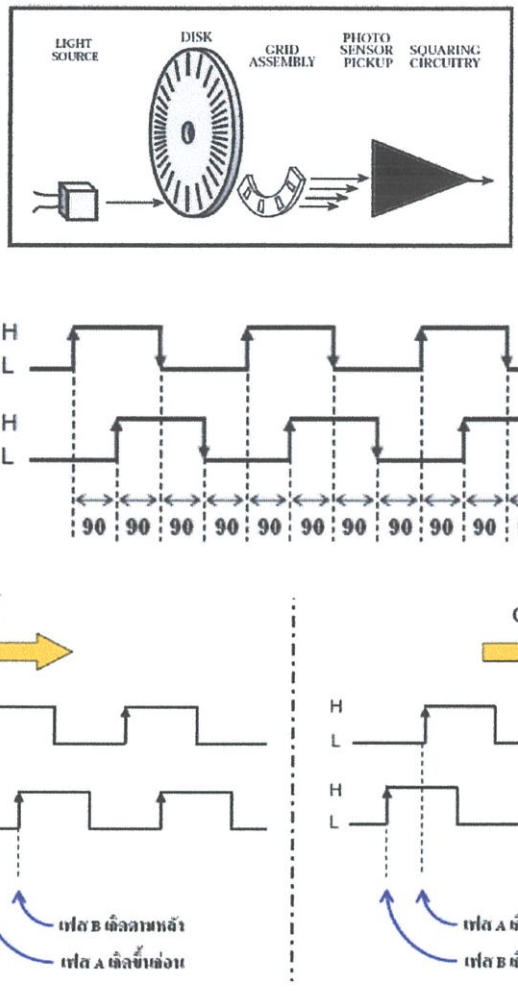
เป็นระบบขับเคลื่อนหรือส่งกำลังประเภทอีกหนึ่ง มีส่วนประกอบหลัก 2 ส่วน ได้แก่ เพลาเกลียว (Screw shaft) เป็นเกลียวกลม และนัต (Nut) ซึ่งมีเม็ดลูกปืนกลม (Ball) จำนวนมากอยู่ภายในตัวนัต ร่องเกลียวกลมบนสกรูและในนัต จะชุบแข็งและเจียระไนผิวเรียบมันเป็นตัวรับน้ำหนักได้ดีและช่วยลดแรงเสียด ทานในการเคลื่อนที่ มี Backlash น้อยกว่า Feed Screw จึงมีประสิทธิภาพและความแม่นยำของระยะการเคลื่อนที่สูง ทำให้ราคาของ Ball Screw สูงด้วยเช่นกัน ปัญหาของ Ballscrew จะมีปัญหากับเครื่อง ที่ทำงานไม้ โดยเฉพาะ MDF จะมีฝุ่น เล็กมาก เข้าไปใน Ballscrew หาก มีการจัดการที่ไม่ดีพอมักเกิดปัญหาเรื่องการอุดตัน ของฝุ่นไม้ในสกรูได้

2.1.8 หลักการทำงานของเอ็นโค้ดเดอร์

เอ็นโค้ดเดอร์ที่นิยมใช้กับเซอร์โวมอเตอร์มีอยู่ด้วยกัน 2 ชนิด ดังนี้

- **เอ็นโค้ดเดอร์แบบ Incremental**

จากรูป 2.12 แสดงส่วนประกอบของเอ็นโค้ดเดอร์ชนิดนี้ โดยลำแสงจะถูกยิงจาก lighting diode ผ่าน fixed disc ไปยัง rotation disc ที่ติดตั้งอยู่บนแกนเพลา โดยมี photo diode เป็นตัวรับแสง ลำแสงจะผ่านรูบน fixed disc และ rotation disc ตามจังหวะของการหมุน ทำให้ได้สัญญาณไฟฟ้าออกมาจาก photo sensor เนื่องจากรูของ A และ B บน fixed disc จะต่างเฟสกันอยู่ 90 องศา ดังนั้นสัญญาณเอาต์พุตทางไฟฟ้าจะได้รูปคลื่นที่ออกมาต่างเฟสกันอยู่ 90 องศา ตามรูปส่วนรูของ Z บนfixed disc จะมีเพียงรูเดียวเท่านั้น ถ้านับค่าเพาส์ที่ได้จากตัวเอ็นโค้ดเดอร์จะเป็นค่ามุมของการหมุนนั่นเอง ส่วนเอาต์พุตเฟส A และ B ที่ต่างเฟสกันอยู่ 90 องศาจะเป็นตัวชี้ถึงทิศทางการหมุนของมอเตอร์ ส่วนเฟส Z หรือ Zero signal เป็นตัวชี้ถึงจุด 0 องศาของการหมุน



รูปที่ 2.12 แสดงโครงสร้างและกราฟการทำงานของเอ็นโค้ดเดอร์

- **เอ็นโค้ดเดอร์แบบ Absolute**

โดยทั่วไป absolute encoder จะมีเอาต์พุตโค้ดให้เลือกเช่น gray code, binary หรือ BCD code แต่สำหรับการเลือกประเภทของ detector ของตัวเซอร์โว ไม่จำเป็นต้องเลือกโค้ดของ เอาต์พุต ให้เลือกแต่เพียงความละเอียดที่ต้องการใช้งานเท่านั้น จะสังเกตเห็นว่าสิ่งที่แตกต่างจาก incremental encoder ก็คือจำนวนของสายสัญญาณเอาต์พุตที่มีจำนวนมากกว่าแบบ incremental ซึ่งจะขึ้นอยู่กับความละเอียดที่เลือกใช้ และอีกประการหนึ่งคือ ความหมายของสัญญาณของ absolute encoder ณ เวลาหนึ่งจะให้ค่าออกมาเป็นค่าสมบูรณ์ ไม่ใช่เป็นค่าที่เปรียบเทียบจากจุดเริ่มต้นเหมือนกับแบบ incremental encoder ดังนั้นถ้าเซอร์โวมอเตอร์ที่มี detector แบบ absolute encoder ก็ไม่จำเป็นจะต้อง search หาตำแหน่งเริ่มต้น (origin search) ใหม่ทุกครั้งที่ ปิดเครื่องแล้วเปิดเครื่อง ขึ้นมาใช้งานใหม่

- LINEAR

การควบคุมแบบนี้จะแตกต่างจากทั้งสองแบบที่กล่าวมาข้างต้น เพราะ ตัวควบคุมพีแอลซีจะส่งสัญญาณอนาลอกให้กับเซอร์โวไดรฟ์เวอร์ ซึ่งไดรฟ์เวอร์จะต้องเป็นชนิดที่สามารถรับสัญญาณควบคุมเป็นอนาลอกได้ เช่น ± 10Vdc

2.1.9 การเชื่อมต่อระหว่างเซอร์โวมอเตอร์ และ Omron CP1H

สำหรับสายควบคุมที่ใช้ในการต่อระหว่างเซอร์โวมอเตอร์ G5-series และ Omron CP1H คือ General Control Cable (R88A-CPG001S) เป็นสายสัญญาณที่มีเพียงคอนเนคเตอร์เชื่อมต่อกับ CN1 ของเซฮร์โวไดรเวอร์เพียงด้านเดียว ส่วนอีกด้านเป็นสายเปล่าเพื่อนำไปต่อกับ CP1H

รูปที่ 2.13 สาย General Control Cable (R88A-CPG001S)

โดยสายภายในประกอบไปด้วยสายสัญญาณทั้งหมด 50 เส้น รายละเลียดแสดงตามตารางต่อไปนี้

ตารางที่ 2.3 สายภายในประกอบไปด้วยสายสัญญาณทั้งหมด 50 เส้น

No.	Color of Wire/Mark	Symbol
1	Orange/Red(1)	+24VCW
2	Orange/Black(1)	+24VCCW
3	Gray/Red(1)	+CW/+PULS/+FA
4	Gray/Black(1)	+CW/+PULS/-FA
5	White/Red(1)	+CCW/+SIGN/+FB
6	White/Black(1)	+CCW/+SIGN/-FB

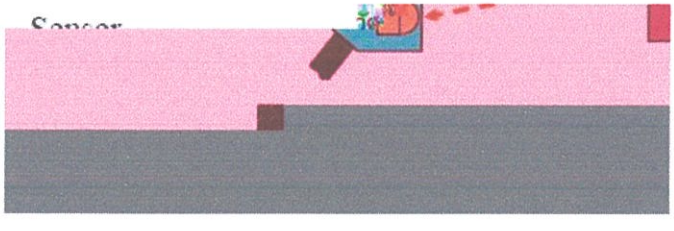
No.	Color of Wire/Mark	Symbol
7	Yellow/Red(1)	+24VIN
8	Pink/Red(1)	SI1
9	Pink/Black(1)	SI2
10	Orange/Red(2)	SO1-
11	Orange/Black(2)	SO1+
12	Yellow/Black(1)	-
13	Gray/Black(2)	SGGND
14	White/red(2)	-
15	White/Black(2)	SGGND
16	Yellow/Red(2)	-
17	Yellow/Black(2) or Pink/Black(2)	SGGND
18	Pink/red(2)	-
19	Orange/Red(5)	Z
20	Gray/Red(2)	-
21	Orange/Red(3)	+A
22	Orange/Black(3)	-A
23	Gray/Red(3)	+Z
24	Gray/Black(3)	-Z
25	Orange/Black(5)	SGGND
26	White/Red(3)	SI3
27	Pink/Black(3)	SI4
28	White/Black(3)	SI5
29	Yellow/Red(3)	SI6
30	Pink/red(3)	SI7
31	Yellow/Black(3)	SI8
32	Gray/Black(4)	SI9
33	Orange/Red(4)	SI10
34	White/Red(4)	SO2-
35	White/Black(4)	SO2+
36	White/Red(4)	ALMCOM

No.	Color of Wire/Mark	Symbol
37	Yellow/Black(4)	/ALM
38	Pink/Red(4)	SO3-
39	Pink/Black(4)	SO3+
40	Gray/Red(4)	-
41	Orange/Black(4)	-
42	Gray/Red(5)	-
43	Gray/Black(5)	-
44	White/Red(5)	+CWLD
45	White/Black(5)	-CWLD
46	Yellow/Red(5)	+CCWLD
47	Yellow/Black(5)	=CCWLD
48	Pink/Black(5)	-B
49	Pink/Red(5)	+B
50	-	-
Shell	-	FG

2.1.10 Photoelectric Sensors

Photoelectric Sensors คืออุปกรณ์ตรวจจับด้วยแสง คือการควบคุมแสงที่ใช้ในกระบวนการผลิตอัตโนมัติต่างๆ โดยทำงานตรวจจับแสงที่มองเห็นหรือแสงที่มองไม่เห็น และตอบสนองการทำงานตามการเปลี่ยนแปลงความเข้มของแสงที่ได้รับ โดย Photoelectric Sensors สามารถแบ่งได้หลักๆ 3 ประเภท

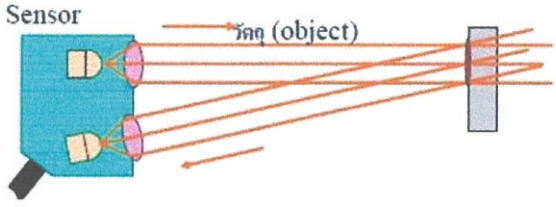
- **Diffuse Mode** (สะท้อนวัตถุโดยตรง)



รูปที่ 2.14 แสดงการทำงานของ **Photoelectric Sensors** แบบ Diffuse Mode

เป็น Sensor ที่อาศัยหลักการยิงแสงไปที่วัตถุ แล้วสะท้อนกลับมา ซึ่ง Sensor ลักษณะนี้นิยมใช้งานโดยทั่วไป เนื่องจากใช้พื้นที่ติดตั้งน้อย เพราะใช้ผิววัตถุที่ตรวจจับเป็นตัวสะท้อนแสงกลับมา

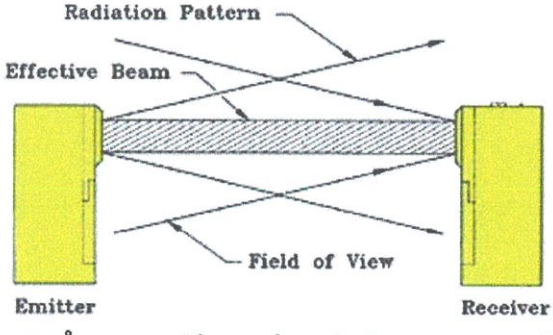
- **Retroreflective Mode** (สะท้อนวัตถุโดยตรงแบบจำกัดลำแสง)



รูปที่ 2.15 แสดงการทำงานของ **Photoelectric Sensors** แบบ Retroreflective Mode

เป็น Sensor ที่ต้องอาศัยแผ่นสะท้อน หรือที่เราเรียกว่า Reflect เป็นตัวสะท้อนแสงกลับมา ซึ่ง Sensorลักษณะนี้สามารถนำไปใช้งานได้ดีในบริเวณที่มีการจำกัดพื้นที่การติดตั้ง นอกจากนี้แผ่นสะท้อน ยังส่งผลทำให้ระยะการตรวจจับวัตถุ สามารถทำได้ไกลขึ้น

- **Opposed Mode** (มีตัวส่งและตัวรับแยกกัน)



รูปที่ 2.16 แสดงการทำงานของ **Photoelectric Sensors** แบบ Opposed Mode

เป็น Sensor แบบที่ใช้ตัวส่งและตัวรับ เนื่องจาก Sensor ลักษณะนี้ มีทั้งตัวส่งและตัวรับ ดังนั้นจึงทำให้ระยะการตรวจจับวัตถุสามารถตรวจจับได้ระยะไกลมากขึ้น นอกจากนี้ในยังสามารถนำไปใช้งานในสภาพแวดล้อมที่มีฝุ่นมากกว่าปกติได้จากประเภทของ Photoelectric Sensor ทั้ง 3 ประเภทข้างต้น ทำให้ในเบื้องต้น เราสามารถตัดสินใจเลือกใช้รูปแบบการติดตั้งได้อย่างเหมาะสม อย่างไรก็ตาม ยังมีปัจจัยอื่นๆ ที่สำคัญ ดังนี้

- สี , พื้นผิวของวัตถุที่ตรวจจับ (Color , Surface) สีต่างๆกัน มีผลต่อการดูดซับของแสง ลักษณะพื้นผิว เช่น ผิวมันวาว ผิวขรุขระ มีผลต่อการสะท้อนของแสง
- ความโตของลำแสง (Beam Pattern) ถ้าความโตของลำแสง มีขนาดใหญ่กว่าวัตถุที่ตรวจจับ ก็จะทำให้แสงนั้นยิงผ่านวัตถุไป บางครั้งทำให้ไม่สามารถตรวจจับได้ ดังนั้นความโตของลำแสง จึงต้องมีขนาดเล็กกว่าวัตถุที่ต้องการตรวจจับ
- ความแตกต่างของสี (Contrast) เนื่องจาก Photoelectric Sensors จะมองเห็นวัตถุที่ต้องการตรวจจับเป็น Gray Scale หรือ เป็นสีเทานั่นเอง ดังนั้น Photo Sensors จะตรวจจับวัตถุที่มีค่า Contrast สูงได้ดีกว่า วัตถุที่มีค่า Contrast ต่ำ

2.2 ซอฟต์แวร์

2.2.1 แลดเดอร์ไดอะแกรม (Ladder Diagrams)

การเขียนโปรแกรม PLC ในปัจจุบันนิยมใช้ภาษาที่เรียกว่าแลดเดอร์ไดอะแกรม (Ladder Diagram) หรือลอจิกแลดเดอร์ (Ladder logic) จัดเป็นภาษาสัญลักษณ์ที่เทียบเคียงมาจากวงจรรีเลย์ สามารถทำความเข้าใจการทำงานได้ง่ายด้วยรูปภาพสัญลักษณ์ จึงทำให้เป็นที่นิยมใช้งานกันอย่างกว้างขว้างในการเขียนโปรแกรมควบคุมพีแอลซี โดยเฉพาะมือใหม่ทั้งหลาย

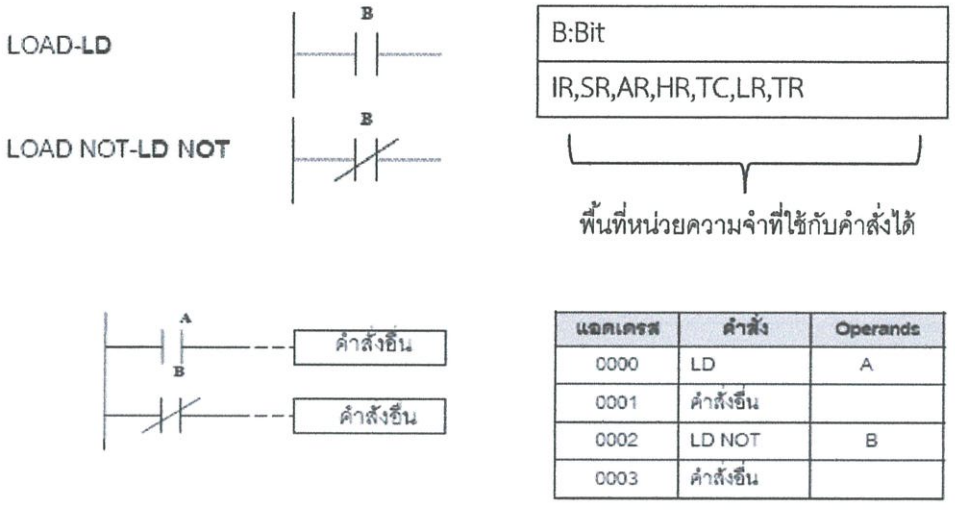
ปัจจุบันการเขียน Ladder ของ PLC ส่วนใหญ่จะใช้ซอฟท์แวร์ในการเขียนภาพสัญลักษณ์ซึ่งทำให้ง่ายและสะดวกมากขึ้นแต่ในการทำงานจริงของ ตัวควบคุมพีแอลซี ไม่ได้ทำงานด้วยรูปภาพแต่จะอาศัยชุดคำสั่ง (Instructions) โดยการเขียนลงในหน่วยความจำเป็นรหัส (Mnemonic code) ซึ่งไม่สามารถจัดเก็บในลักษณะของแลดเดอร์ไดอะแกรมได้โดยตรง ดังนั้นผู้ใช้งานจึงจำเป็นต้องทำความเข้าใจชุดคำสั่งซึ่งเป็นลอจิกพื้นฐานต่างๆ เพื่อสร้างความเข้าใจและสามารถนำไปประยุกต์ใช้งานได้

ท่านสามารถค้นหาข้อมูลตัวควบคุมพีแอลซี หรือคู่มือการใช้งานตัวควบคุมพีแอลซี ได้จากเว็ปไซต์ของ ตัวควบคุมพีแอลซี ยี่ห้อนั้นๆได้ ซึ่งคู่มือดังกล่าวจะมีรายละเอียดเกี่ยวกับการเขียนโปรแกรมด้วยเช่นกัน

2.2.1.1 กลุ่มคำสั่งลอจิกพื้นฐาน

ตัวควบคุมพีแอลซีแต่ละยี่ห้อจะมีรูปแบบของคำสั่ง Mnemonic ที่คล้ายกัน ในที่นี้เราจะอ้างอิงคำสั่ง PLC ของ Omron เพื่อใช้ประกอบคำอธิบาย ส่วนแอดเดรสในตารางคำสั่งหมายถึงตำแหน่งหน่วยความจำที่ ตัวควบคุมพีแอลซีเก็บคำสั่งและลำดับในการประมวลผล การเขียน Ladder ของ Omron น่าจะเป็นต้นแบบในการเรียนรู้ที่ดีสำหรับผู้เริ่มต้น

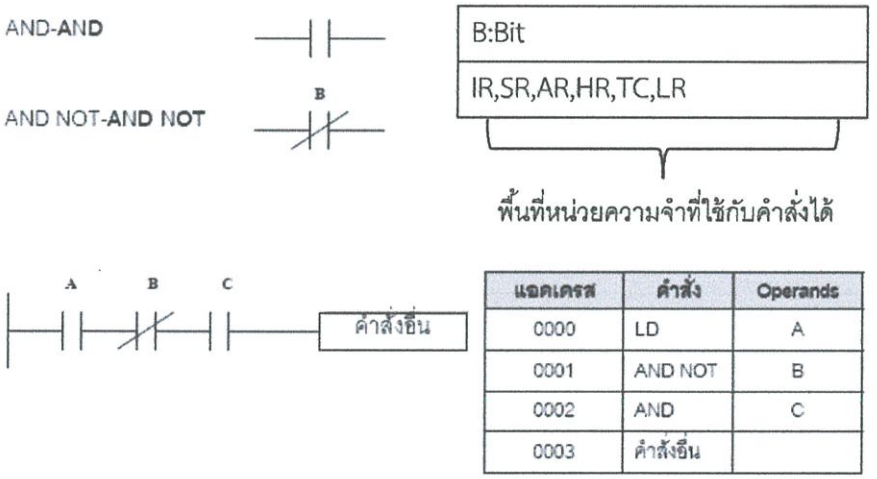
2.2.1.1.1 การใช้คำสั่ง LOAD (LD), LOAD NOT (LD NOT)



แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	คำสั่งอื่น	
0002	LD NOT	B
0003	คำสั่งอื่น	

รูปที่ 2.17 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง LOAD (LD), LOAD NOT (LD NOT)

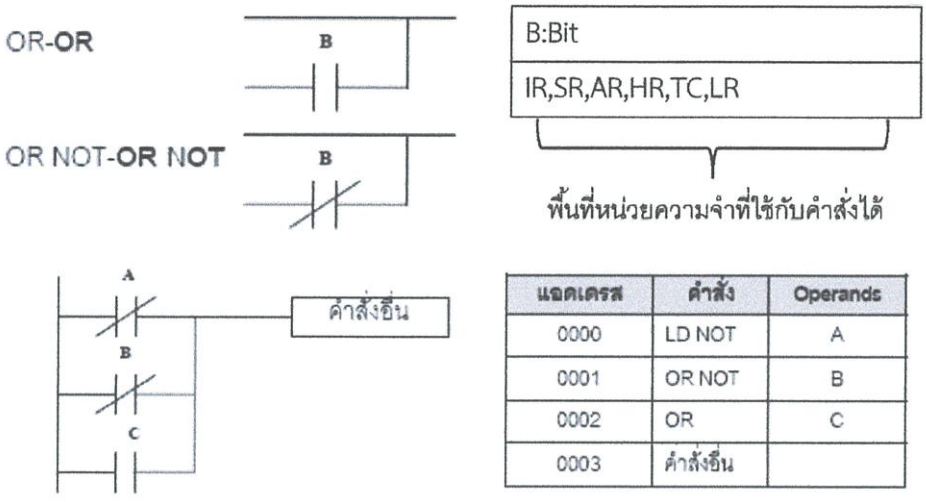
2.2.1.1.2 การใช้คำสั่ง AND, AND NOT



แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	AND NOT	B
0002	AND	C
0003	คำสั่งอื่น	

รูปที่ 2.18 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND, AND NOT

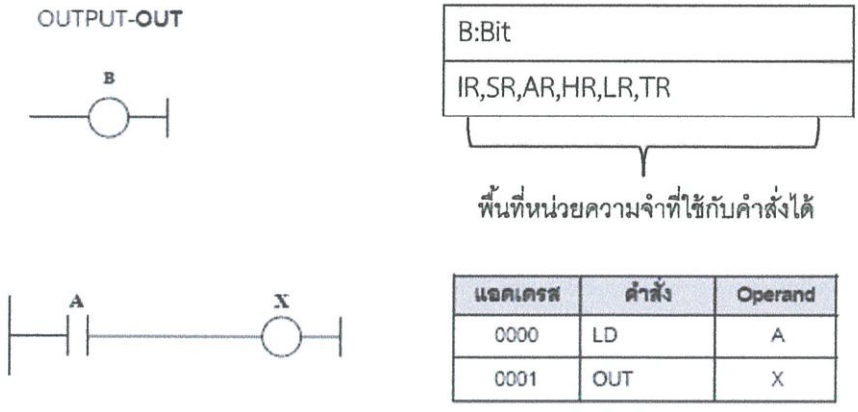
2.2.1.1.3 การใช้คำสั่ง OR, OR NOT



แอดเดรส	คำสั่ง	Operands
0000	LD NOT	A
0001	OR NOT	B
0002	OR	C
0003	คำสั่งอื่น	

รูปที่ 2.19 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OR, OR NOT

2.2.1.1.4 การใช้คำสั่ง OUT

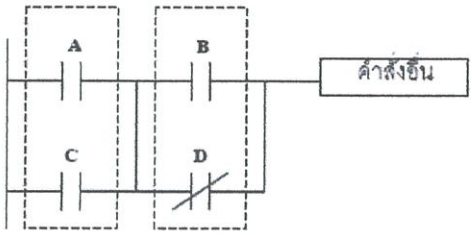


แอดเดรส	คำสั่ง	Operand
0000	LD	A
0001	OUT	X

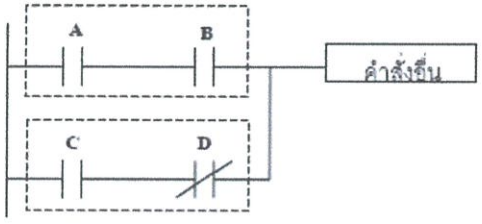
รูปที่ 2.20 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง OUT

2.2.1.1.5 การใช้คำสั่ง AND LOAD (AND LD), OR LOAD (OR LD)

คำสั่งทั้งสองจะทำหน้าที่เชื่อมต่อกลุ่มแลดเดอร์ไดอะแกรม ในกรณีที่ต่ออนุกรม หรือขนานกันมากกว่า 1หน้าสัมผัส ซึ่งการใช้คำสั่ง AND หรือ OR นั้น จะกระทำทีละ 1 หน้าสัมผัสเท่านั้นจึงต้องใช้ AND LD หรือ OR LD

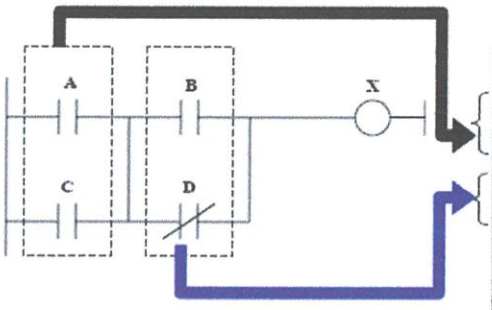


แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	OR	C
0002	LD	B
0003	OR NOT	D
0004	AND LD	

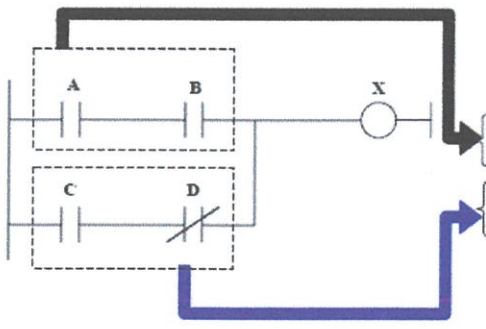


แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	AND	B
0002	LD	C
0003	AND NOT	D
0004	OR LD	---

- AND LD เป็นคำสั่งในการเชื่อมโปรแกรม 2 บล็อก ในแบบอนุกรม



แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	OR	C
0002	LD	B
0003	OR NOT	D
0004	AND LD	---
0005	OUT	X



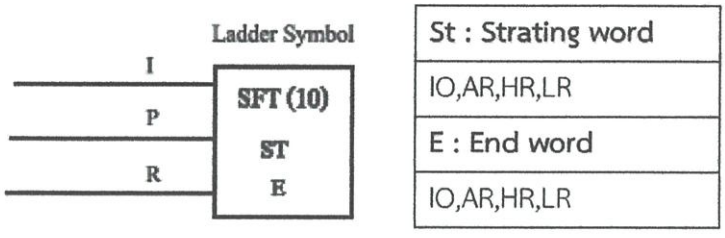
แอดเดรส	คำสั่ง	Operands
0000	LD	A
0001	AND	B
0002	LD	C
0003	AND NOT	D
0004	OR LD	---
0005	OUT	X

รูปที่ 2.21 แสดงแลดเดอร์ไดอะแกรมของคำสั่ง AND LOAD (AND LD), OR LOAD (OR LD)

2.2.1.1.6 การใช้คำสั่ง END (END)

เมื่อสิ้นสุดการเขียนโปรแกรมแล้วจะต้องจบด้วยคำสั่ง END เสมอ ถ้าไม่มีคำสั่งนี้ เมื่อผู้ใช้สั่งให้ ตัวควบคุมพีแอลซี ทำงานจะเกิด Error โดยสังเกตได้จากไฟ Error/Alarm สีแดงจะติดค้าง แต่ซอท์ฟแวร์ของ ตัวควบคุมพีแอลซี หลายยี่ห้อจะช่วยเพิ่มคำสั่งนี้ให้อัตโนมัติเพื่อป้องกันการลืม

2.2.1.1.7 การใช้งานคำสั่ง SFT (10)



รูปที่ 2.22 แสดงส่วนประกอบของคำสั่ง SHT(10)

ST: กำหนด CH หรือ WORD เริ่มต้นของการ SHIFT

E : กำหนด CH หรือ WORD สุดท้ายของการ SHIFT

คำสั่งนี้เป็นคำสั่งเลื่อนข้อมูล " 0 " หรือ " 1 " ของ CHANNEL ที่ต้องการเลื่อนข้อมูล ซึ่งประกอบด้วย 3 อินพุท ดังนี้

DATA INPUT (I) หมายถึง เป็นค่าข้อมูลที่กำหนด " 0 " หรือ " 1 "

PLUSE INPUT (P) หมายถึง เป็นจังหวะพัลซ์หรือสเต็ปของการเลื่อนข้อมูล

RESET INPUT (R) หมายถึงการเซ็ทข้อมูลเพื่อเริ่มต้น

2.2.1.1.8 การใช้งานคำสั่ง IL (02), ILC (03)

คำสั่ง IL และ ILC จะต้องใช้ร่วมกันคือ ถ้าเริ่มต้นมีการใช้คำสั่งด้วย IL เมื่อใดแล้วถ้าต้องการสิ้นสุดการทำงานต้องจบด้วย ILC เงื่อนไขของคำสั่งคือ คอนแทคตรงหน้าส่วนของ IL มีสภาวะ “ON” จะทำให้โปรแกรมที่อยู่ระหว่าง IL และ ILC ทำงานเป็นปกติ แต่ถ้าคอนแทคตำแหน่งดังกล่าวมีสภาวะ “OFF” จะทำให้การทำงานของโปรแกรมระหว่าง IL และ ILC ไม่ทำงาน ในขณะเดียวกัน Output Coil ในช่วงนั้นจะมีสภาวะ “OFF” ด้วย

2.2.1.1.9 การใช้งานคำสั่ง Differential UP-DIFU(013) และ Differential DOWN-DIFD(014)

การใช้งานของคำสั่งนี้ จะทำงานเพียงขอบขาขึ้น หรือขอบขาลงของอินพุตเท่านั้น และจะทำงานเพียง One cycle time เท่านั้น

B:Bit
IR,SR,AR,HR,LR

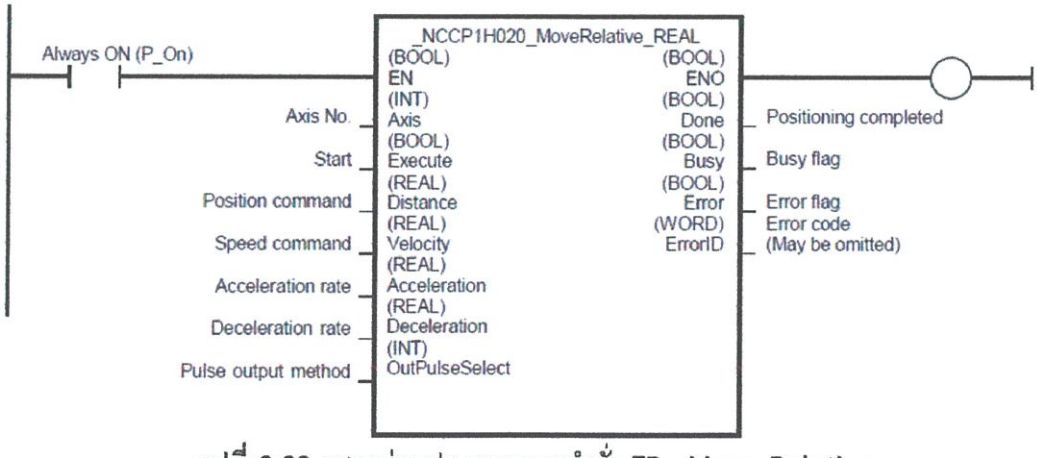
พื้นที่หน่วยความจำที่ใช้กับคำสั่งได้

2.2.1.1.10 การใช้คำสั่ง FB: Move Relative

คำสั่ง Move Relative ซึ่งเป็นคำสั่งที่ทำให้เซอร์โวมอเตอร์แกนที่เราต้องการควบคุมให้เคลื่อนที่ตามตำแหน่ง,ความเร็ว,อัตราเร่งและอัตราหน่วงที่กำหนด โดยจะทำงานต่อจากตำแหน่งเดิมที่มีการเคลื่อนที่

หลักการทำงาน

เมื่อเริ่มต้น บิท A จะทำให้ฟังก์ชั่น Move Relative ทำงาน เซอร์โวมอเตอร์จะหมุนตามตัวแปรที่ตั้งไว้ ในขณะที่ บิท C จะทำงาน (ON) จนกระทั่งเซอร์โวมอเตอร์หมุนถึงตำแหน่งที่เรากำหนดไว้ เมื่อถึงตำแหน่งที่กำหนดแล้ว บิท B จะทำงาน (ON) เพื่อตัดการทำงานของ บิท A ในกรณีที่เกิดความผิดปกติขึ้น(Error) บิท D จะทำงาน (ON) เพื่อตัดการทำงานของ บิท A เช่นกัน



รูปที่ 2.23 แสดงส่วนประกอบของคำสั่ง FB : Move Relative

ตารางที่ 2.4 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Move Relative

Name	Variable Name	Data Type	Default	Range	Description
EN	En	BOOL			1(ON) : ใช้งาน FB 0(OFF) : ไม่ใช้งาน FB
Axis No.	Axis	INT	&0	&0 - &3	&0 Pulse Output 0 &1 Pulse Output 1 &2 Pulse Output 2 &3 Pulse Output 3
Start	Execute	BOOL	0(OFF)		สั่งมอเตอร์ให้เคลื่อนที่
Position Command	Distance	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุระยะทางที่จะให้มอเตอร์เคลื่อที่ไป (pulse)
Speed Command	Velocity	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุความเร็วที่ต้องการ (Hz)
Acceleration Rate	Acceleration	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุอัตราเร่งที่ต้องการ (Hz/4ms)
Deceration Rate	Deceration	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุอัตราหน่วงที่ต้องการ (Hz/4ms)
Pulse output method	Output Pulse Select	INT	&0	&0 - &1	&0 : CW/CCW Output &1 : Pulse + Direction Output

ตารางที่ 2.5 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง Move Relative

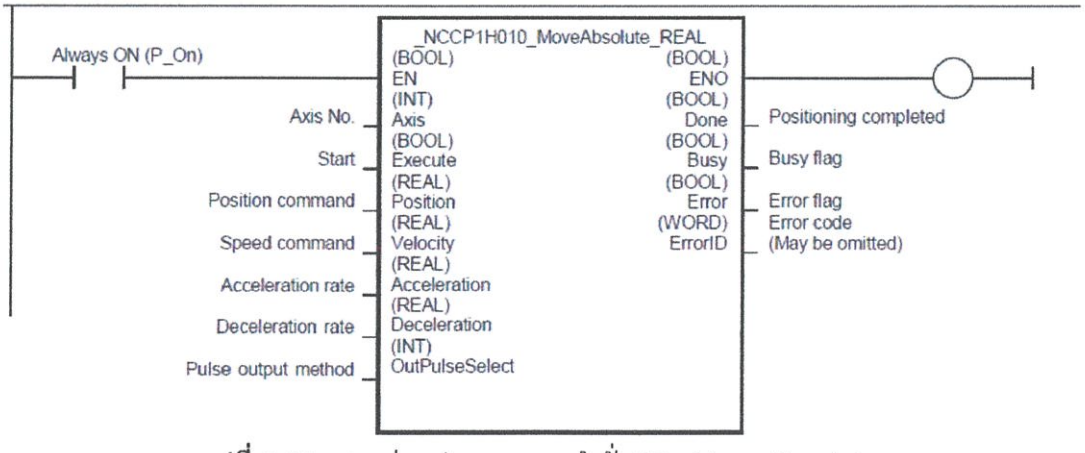
Name	Variable Name	Data Type	Default	Range	Description
ENO	ENO	BOOL			1(ON) : FB ทำงานปกติ 0(OFF) : FB ทำงานผิดปกติ
Positioning Completed	DONE	BOOL			1(ON) : เมื่อมอเตอร์เคลื่อนที่ไปยังตำแหน่งที่ระบุไว้แล้ว
Busy flag	Busy	BOOL			1(ON) : เมื่ออยู่ในกระบวนการทำงาน
Error flag	Error	BOOL			1(ON) : เมื่อเกิดความผิดพลาดในการทำงาน
Error Code (May be omitted)	ErrorID	WORD			แสดง Error Code โดยสามารถเทียบได้จากคู่มือการใช้งานหากนอกเหนือที่มีจะแสดงเป็น #0000

2.2.1.1.11 การใช้คำสั่ง FB: Move Absolute

คำสั่ง Move Absolute ทำหน้าที่สั่งให้เซอร์โวมอเตอร์ในแกนที่ต้องการ หมุนตามตำแหน่ง,ความเร็ว,อัตราเร่ง,อัตราหน่วงที่กำหนด โดยมีเงื่อนไขก่อนใช้คำสั่ง Move Absolute ต้องทำคำสั่ง Origin Search ก่อน

หลักการทำงาน

เมื่อเริ่มต้น บิท A จะทำให้ฟังก์ชั่นนี้ทำงาน เซอร์โวมอเตอร์จะหมุนตามตัวแปรที่กำหนดไว้ ในขณะที่ บิท C จะทำงาน(ON) จนกระทั่งเซอร์โวมอเตอร์หมุนถึงตำแหน่งที่กำหนดไว้ บิท B จะทำงาน(ON) เพื่อตัดสัญญาณ บิท A และหากเกิดความผิดปกติ (Error) บิท D จะทำงาน(ON) เพื่อตัดการทำงานของ บิท A



รูปที่ 2.24 แสดงส่วนประกอบของคำสั่ง FB : Move Absolute

ตารางที่ 2.6 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Move Absolute

Name	Variable Name	Data Type	Default	Range	Description
EN	En	BOOL			1(ON) : ใช้งาน FB 0(OFF) : ไม่ใช้งาน FB
Axis No.	Axis	INT	&0	&0 - &3	&0 Pulse Output 0 &1 Pulse Output 1 &2 Pulse Output 2 &3 Pulse Output 3
Start	Execute	BOOL	0(OFF)		สั่งมอเตอร์ให้เคลื่อนที่
Position Command	Distance	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุระยะทางที่จะให้มอเตอร์เคลื่อที่ไป (pulse)
Speed Command	Velocity	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุความเร็วที่ต้องการ (Hz)
Acceleration Rate	Acceleration	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุอัตราเร่งที่ต้องการ (Hz/4ms)
Deceration Rate	Deceration	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุอัตราหน่วงที่ต้องการ (Hz/4ms)
Pulse output method	Output Pulse Select	INT	&0	&0 - &1	&0 : CW/CCW Output &1 : Pulse + Direction Output

ตารางที่ 2.7 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Move Absolute

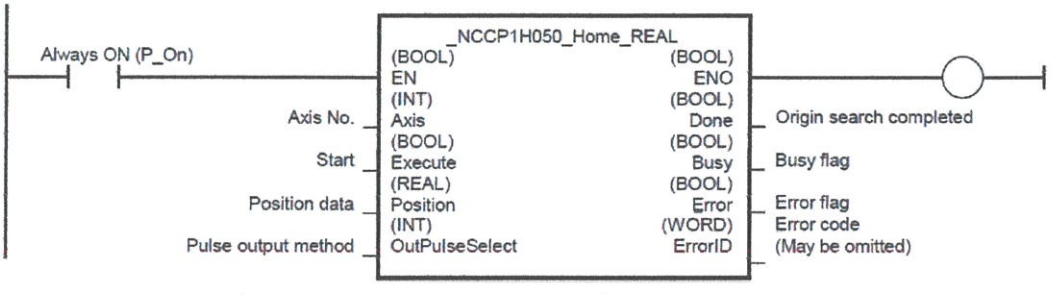
Name	Variable Name	Data Type	Default	Range	Description
EN0	EN0	BOOL			1(ON) : FB ทำงานปกติ 0(OFF) : FB ทำงานผิดปกติ
Positioning Completed	DONE	BOOL			1(ON) : เมื่อมอเตอร์เคลื่อนที่ไปยังตำแหน่งที่ระบุไว้แล้ว
Busy flag	Busy	BOOL			1(ON) : เมื่ออยู่ในกระบวนการทำงาน
Error flag	Error	BOOL			1(ON) : เมื่อเกิดความผิดพลาดในการทำงาน
Error Code (May be omitted)	ErrorID	WORD			แสดง Error Code โดยสามารถเทียบได้จากคู่มือการใช้งานหากนอกเหนือที่มีจะแสดงเป็น #0000

2.2.1.1.12 การใช้คำสั่ง FB : ORIGIN SEARCH

คำสั่ง ORIGIN SEARCH มีหน้าที่สั่งให้เซอร์โวมอเตอร์กลับสู่ตำแหน่งจุดเริ่มต้น (Origin)

หลักการทำงาน

เมื่อเริ่มต้น บิท A จะทำให้ฟังก์ชั่นนี้ทำงาน โดยเซอร์โวมอเตอร์จะเริ่มค้นหาตำแหน่ง Origin เมื่อพบจุด Origin แล้ว บิท B จะทำงาน (ON) เพื่อตัดสัญญาณของ บิท A เซอร์โวจะหยุด ณ ตำแหน่ง Origin หากเกิดความผิดปกติ(Error) บิท D จะทำงาน (ON) เพื่อตัดการทำงานของ บิท A



รูปที่ 2.25 แสดงส่วนประกอบของคำสั่ง FB : ORIGIN SEARCH

ตารางที่ 2.8 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : ORIGIN SEARCH

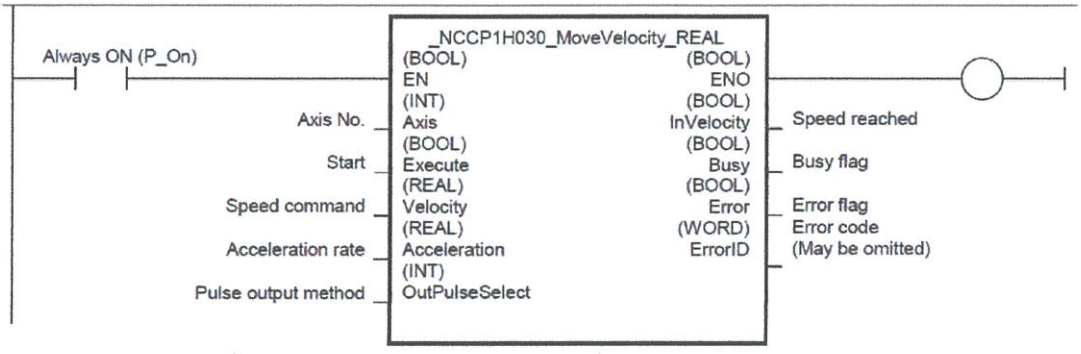
Name	Variable Name	Data Type	Default	Range	Description
EN	EN	BOOL			1(ON) : ใช้งาน FB 0(OFF) : ไม่ใช้งาน FB
Axis No.	Axis	INT	&0	&0 - &3	&0 Pulse Output 0 &1 Pulse Output 1 &2 Pulse Output 2 &3 Pulse Output 3
Start	Execute	BOOL	0(OFF)		ใช้งาน Origin Search
Position Data	Position	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุตำแหน่งที่เคลื่อนที่เป็นจุดเริ่มต้นใหม่ (pulse)
Pulse output method	Output Pulse Select	INT	&0	&0 - &1	&0 : CW/CCW Output &1 : Pulse + Direction Output

ตารางที่ 2.9 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : ORIGIN SEARCH

Name	Variable Name	Data Type	Default	Range	Description
ENO	ENO	BOOL			1(ON) : FB ทำงานปกติ 0(OFF) : FB ทำงานผิดปกติ
Origin search Completed	DONE	BOOL			1(ON) : เมื่อเซอร์โวมอเตอร์เคลื่อนที่ไปยังตำแหน่งเริ่มต้นแล้ว
Busy flag	Busy	BOOL			1(ON) : เมื่ออยู่ในกระบวนการทำงาน
Error flag	Error	BOOL			1(ON) : เมื่อเกิดความผิดพลาดในการทำงาน
Error Code (May be omitted)	ErrorID	WORD			แสดง Error Code โดยสามารถเทียบได้จากคู่มือการใช้งานหากนอกเหนือที่มีจะแสดงเป็น #0000

2.2.1.1.13 การใช้คำสั่ง FB : Move Velocity

คำสั่ง Move Velocity ซึ่งเป็นคำสั่งที่ทำให้เซอร์โวมอเตอร์แกนที่เราต้องการควบคุมให้เคลื่อนที่ตามความเร็ว,อัตราเร่งที่กำหนด โดยจะทำงานต่อจากตำแหน่งเดิมที่มีการเคลื่อนที่



รูปที่ 2.26 แสดงส่วนประกอบของคำสั่ง FB : Move Velocity

ตารางที่ 2.10 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Move Velocity

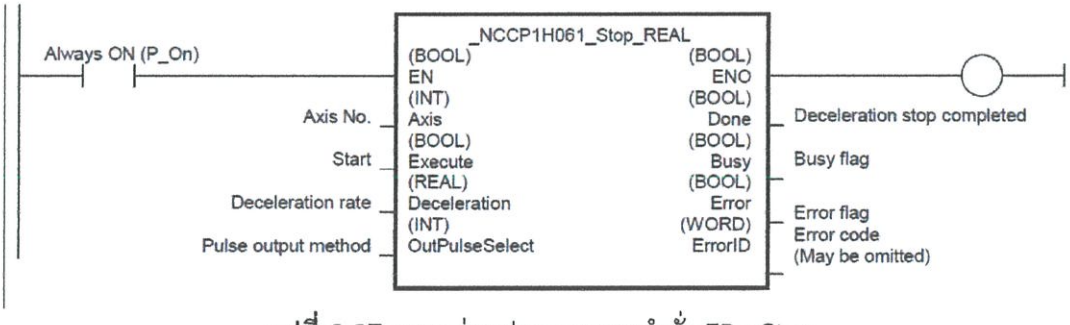
Name	Variable Name	Data Type	Default	Range	Description
EN	En	BOOL			1(ON) : ใช้งาน FB 0(OFF) : ไม่ใช้งาน FB
Axis No.	Axis	INT	&0	&0 - &3	&0 Pulse Output 0 &1 Pulse Output 1 &2 Pulse Output 2 &3 Pulse Output 3
Start	Execute	BOOL	0(OFF)		สั่งมอเตอร์ให้เคลื่อนที่
Speed Command	Velocity	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุความเร็วที่ต้องการ (Hz)
Acceleration Rate	Acceleration	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุอัตราเร่งที่ต้องการ (Hz/4ms)
Pulse output method	Output Pulse Select	INT	&0	&0 - &1	&0 : CW/CCW Output &1 : Pulse + Direction Output

ตารางที่ 2.11 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Move Velocity

Name	Variable Name	Data Type	Default	Range	Description
EN0	EN0	BOOL			1(ON) : FB ทำงานปกติ 0(OFF) : FB ทำงานผิดปกติ
Speed reached	InVelocity	BOOL			1(ON) : เมื่อมอเตอร์มีความเร็วตามที่ระบุไว้
Busy flag	Busy	BOOL			1(ON) : เมื่ออยู่ในกระบวนการทำงาน
Error flag	Error	BOOL			1(ON) : เมื่อเกิดความผิดพลาดในการทำงาน
Error Code (May be omitted)	ErrorID	WORD			แสดง Error Code โดยสามารถเทียบได้จากคู่มือการใช้งานหากนอกเหนือที่มีจะแสดงเป็น #0000

2.2.1.1.14 การใช้คำสั่ง FB : Stop

คำสั่ง Stop เป็นคำสั่งที่ทำให้เซอร์โวมอเตอร์แกนที่เราต้องการควบคุมให้หยุดการเคลื่อนที่ตามอัตราหน่วงที่กำหนด โดยจะทำงานต่อจากตำแหน่งเดิมที่มีการเคลื่อนที่



รูปที่ 2.27 แสดงส่วนประกอบของคำสั่ง FB : Stop

ตารางที่ 2.12 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Stop

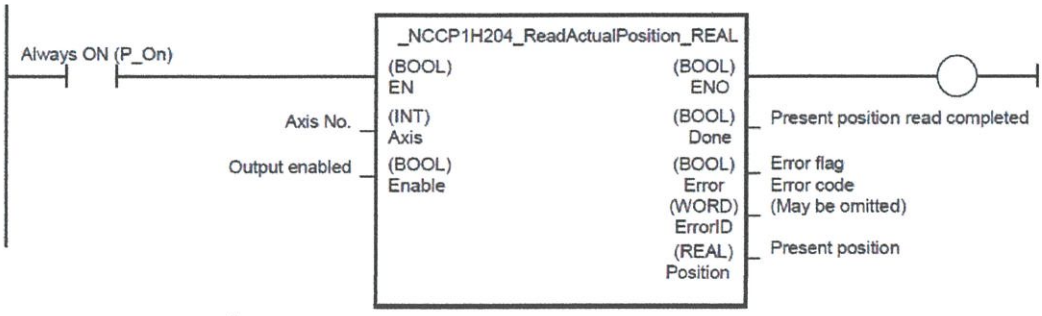
Name	Variable Name	Data Type	Default	Range	Description
EN	EN	BOOL			1(ON) : ใช้งาน FB 0(OFF) : ไม่ใช้งาน FB
Axis No.	Axis	INT	&0	&0 - &3	&0 Pulse Output 0 &1 Pulse Output 1 &2 Pulse Output 2 &3 Pulse Output 3
Start	Execute	BOOL	0(OFF)		ใช้งาน Origin Search
Deceration Rate	Deceration	REAL	(+1.0)	(+1.0) to (+65535.0)	ระบุอัตราหน่วงที่ต้องการ (Hz/4ms)
Pulse output method	Output Pulse Select	INT	&0	&0 - &1	&0 : CW/CCW Output &1 : Pulse + Direction Output

ตารางที่ 2.13 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Stop

Name	Variable Name	Data Type	Default	Range	Description
ENO	ENO	BOOL			1(ON) : FB ทำงานปกติ 0(OFF) : FB ทำงานผิดปกติ
Deceleration Stop Complete	Done	BOOL			1(ON) : เมื่อมอเตอร์หยุดการเคลื่อนที่
Busy flag	Busy	BOOL			1(ON) : เมื่ออยู่ในกระบวนการทำงาน
Error flag	Error	BOOL			1(ON) : เมื่อเกิดความผิดพลาดในการทำงาน
Error Code (May be omitted)	ErrorID	WORD			แสดง Error Code โดยสามารถเทียบได้จากคู่มือการใช้งานหากนอกเหนือที่มีจะแสดงเป็น #0000

2.2.1.1.15 การใช้คำสั่ง FB : Read Actual Position

คำสั่ง Read Actual Position เป็นคำสั่งที่ใช้ระบุตำแหน่งการเคลื่อนที่ของเซอร์โวมอเตอร์ เพื่อนำค่าที่ได้นำไปแสดงผล



รูปที่ 2.28 แสดงส่วนประกอบของ FB : Read Actual Position

ตารางที่ 2.14 แสดงความหมายของตัวแปรอินพุตของคำสั่ง FB : Read Actual Position

Name	Variable Name	Data Type	Default	Range	Description
EN	EN	BOOL			1(ON) : ใช้งาน FB 0(OFF) : ไม่ใช้งาน FB
Axis No.	Axis	INT	&0	&0 - &3	&0 Pulse Output 0 &1 Pulse Output 1 &2 Pulse Output 2 &3 Pulse Output 3
Output Enabled	Enable	BOOL			1(ON) : เปิดใช้งาน 0(OFF) : ปิดใช้งาน

ตารางที่ 2.15 แสดงความหมายของตัวแปรเอาท์พุตของคำสั่ง FB : Read Actual Position

Name	Variable Name	Data Type	Default	Range	Description
ENO	ENO	BOOL			1(ON) : FB ทำงานปกติ 0(OFF) : FB ทำงานผิดปกติ
Present Position Read Complete	Done	BOOL			1(ON) : เมื่อทำการอ่านค่าสำเร็จ
Present Position	Position	REAL	(+1.0)	(+1.0) to (+65535.0)	ตำแหน่งที่อ่านค่าได้
Error flag	*Error*	BOOL			1(ON) : เมื่อเกิดความผิดพลาดในการทำงาน
Error Code (May be omitted)	ErrorID	WORD			แสดง Error Code โดยสามารถเทียบได้จากคู่มือการใช้งานหากนอกเหนือที่มีจะแสดงเป็น #0000

2.2.2 OPC (Ole For Process Control)

OPC เป็นชุดการสื่อสารมาตรฐานสำหรับการเชื่อมต่อและวิธีการที่ได้นำไปใช้ในการสื่อสารกับระบบควบคุมในอุตสาหกรรมกระบวนการผลิตและอุตสาหกรรมแบบอัตโนมัติต่างๆ OPC เป็นเทคโนโลยีที่อยู่บนพื้นฐาน Windows's OLC, Com และ DCOM (Distributed Component Object Model) ซึ่งเทคโนโลยีต่างๆเหล่านี้เป็นตัวที่ใช้กำหนดให้โปรแกรมที่ทำงานบนเครื่องคอมพิวเตอร์แต่ละชนิดมีความสามารถที่ใช้ข้อมูลร่วมกันได้ ผู้ใช้งานอาจเคยได้ยินหรือเคยได้ใช้ความสามารถในการสื่อสารแบบ OLE มาบ้างแล้ว ดังเช่น การเพิ่มเติม Spreadsheet ลงไปบนเอกสาร Word ซึ่ง OLE จะยินยอมให้มีการแก้ไขหรือเพิ่มเติมข้อมูลใน Spreadsheet ได้ตลอดเวลา จากตัวอย่างนั้นเห็นได้ว่าผู้ใช้งานไม่จำเป็นต้องกำหนดรายละเอียดบนโปรแกรมเพิ่มเติมเพียงแต่ใช้เมาส์คลิกไปยังส่วนต่างๆในการทำงานเท่านั้น หลังจากนั้น OLE จะทำการกำหนดให้ Spreadsheet ทำหน้าที่เป็นโปรแกรมแม่ข่าย (OLE Server) สำหรับส่งข้อมูลไปยังเอกสาร Word ซึ่งทำหน้าที่เป็นโปรแกรมลูกข่าย(OLE Client)

OPC จะเป็นรูปแบบการสื่อสารที่ถูกพัฒนามาจาก OLE โดยการเพิ่มคุณลักษณะที่เป็นประโยชน์ต่างๆเข้าไปใน OLE เพื่อทำให้มีความน่าเชื่อถือสูงกับการนำไปใช้งานในระบบควบคุมอุตสาหกรรมกระบวนการผลิต เนื่องจากโปรแกรมแม่ข่ายของ OLE (OLE Client) และการตรวจสอบ

ข้อมูลที่ได้รับว่าถูกต้องหรือไม่ ซึ่งจะเห็นได้ว่าความถูกต้องและความน่าเชื่อถือได้ของการรับส่งข้อมูลจะมีค่าต่ำมาก ดังนั้นจึงไม่เหมาะสมในการนำไปใช้งานกับระบบควบคุมอุตสาหกรรมกระบวนการผลิตได้ ซึ่ง OPC ได้ทำการเพิ่มเติมความสามารถในการตรวจสอบข้อมูลระหว่างการสื่อสารระหว่างโปรแกรมลูกข่าย (Client Application) และโปรแกรมแม่ข่าย (Server Application) จึงทำให้ข้อมูลที่ได้มีความถูกต้องแม่นยำและมีความเชื่อถือได้มากยิ่งขึ้น

ค่าข้อมูลจริงของ OPC จะถูกจัดเตรียมไว้สำหรับเชื่อมต่อร่วมกันระหว่างการสื่อสารกับอุปกรณ์หลายชนิดรวมไปถึงโปรแกรมคอมพิวเตอร์สำหรับใช้ในการควบคุมหรืออุปกรณ์ในกระบวนการผลิต ก่อนหน้าที่ OPC ได้ถูกพัฒนาขึ้นผู้พัฒนาโปรแกรมคอมพิวเตอร์ต้องทำการพัฒนาไดรฟ์เวอร์สำหรับการสื่อสารกับอุปกรณ์แต่ละชนิดอย่างเฉพาะเจาะจง สำหรรับระบบควบคุมที่จะนำไปเชื่อมต่อกับอุปกรณ์นั้นๆ เช่น หน่วยแสดงผลหนึ่งชุดผู้จำหน่ายต้องทำการพัฒนาไดรฟ์เวอร์หลายๆแบบเพื่อใช้สำหรับระบบ PLC หรือระบบ DCS ที่แตกต่างกันไป

สำหรับการใช้งาน OPC ผู้จัดจำหน่ายหน่วยแสดงผลไม่จำเป็นต้องทำการพัฒนาไดรฟ์เวอร์หลายๆแบบเพื่อเตรียมไว้สำหรับเครือข่ายหรือระบบควบคุมที่แตกต่างกัน และในทางกลับกันผู้จัดจำหน่ายเหล่านี้เพียงแต่ทำการพิจารณาไดรฟ์เวอร์ OPC ที่เหมาะสมกับผลิตภัณฑ์ของตนเอง ไว้ใช้ในการเชื่อมต่อเพื่อทำการสื่อสารไปยังไดรฟ์เวอร์ OPC ของผลิตภัณฑ์จากผู้ผลิตรายอื่นๆ

ในปัจจุบันกลไกสื่อสารมาตรฐานสำหรับระบบปฏิบัติการวินโดว์อย่างเช่น OPC ได้มีการจัดเตรียมส่วนเชื่อมต่อสำหรับการใช้งาน (Application Programmer's Interface : API) ของผลิตภัณฑ์หลายๆชนิด โดยระบบนี้สามารถเข้าถึงข้อมูลได้ทางโปรแกรมการใช้งานหรืออาจจะเข้าถึงข้อมูลจากอุปกรณ์ควบคุมโดยผ่านไดรฟ์เวอร์

การใช้งาน OPC ในการเชื่อมต่อระบบควบคุมต่างๆเข้าด้วยกัน เริ่มมีจำนวนเพิ่มขึ้นสำหรับในการเชื่อมต่อระบบควบคุมอื่นๆ (Third party system) เข้ากับระบบควบคุมหลัก ข้อดีที่เห็นได้อย่างชัดเจนนั่นคือเป็นความสามารถในการใช้งานโปรแกรมแบบพิเศษต่างๆบนระบบควบคุมกระบวนการผลิต ข้อดีนี้ทำให้ผู้ใช้งานมีทางเลือกในการใช้งานโปรแกรมแบบพิเศษเหล่านี้ได้อย่างกว้างขวางดังตัวอย่างเช่น Advance Control, Trending, Data logging หรือ Data Conditioning เป็นต้น

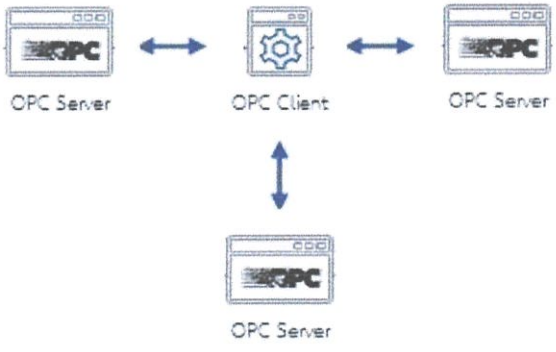
ในการที่ทำให้เข้าใจการทำงานของการสื่อสารแบบ OPC จะต้องมีพื้นฐานความเข้าใจการทำงานของเครือข่ายแบบลูกข่าย (Client) และแม่ข่าย (Server) โดยแม่ข่ายจะเป็นส่วนที่ใช้ในการจัดเตรียมข้อมูลสำหรับลูกข่ายในการนำไปใช้งาน ลูกข่ายจะมีโปรแกรมประมวลผลจากข้อมูลที่ได้ถูกจัดเตรียมไว้ในแม่ข่าย ซึ่งแม่ข่ายจะไม่สามารถสื่อสารกันได้โดยตรงกับแม่ข่ายอื่นๆในระบบหรือบนเครือข่าย โปรแกรมการทำงานอยู่บนลูกข่ายจะถูกใช้เป็นสะพานในการเชื่อมต่อข้อมูลระหว่างแม่ข่ายตั้งแต่สองชุดขึ้นไป ดังแสดงในรูปที่ 2.26

รูปที่ 2.29 การเชื่อมต่อข้อมูลระหว่างโปรแกรมการใช้งานแม่ข่าย

ในทำนองเดียวกันโปรแกรมการทำงานอยู่บนลูกข่ายก็ไม่สามารถสื่อสารกันได้โดยตรงกับลูกข่ายอื่นๆในระบบหรือบนเครือข่าย โปรแกรมการทำงานอยู่บนแม่ข่ายจะถูกใช้เป็นสะพานในการเชื่อมต่อข้อมูลระหว่างลูกข่ายตั้งแต่สองชุดขึ้นไป สามารถแสดงได้ในรูปที่ 2.26

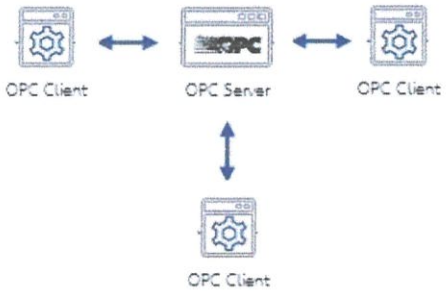
รูปที่ 2.30 การเชื่อมต่อข้อมูลระหว่างโปรแกรมการใช้งานบนลูกข่าย

โปรแกรมการทำงานอยู่บนลูกข่ายอาจจะเข้าถึงข้อมูลที่อยู่บนแม่ข่ายได้หลายชุดในเวลาเดียวกันซึ่งเป็นคุณลักษณะพิเศษของโครงสร้างเครือข่ายแบบลูกข่ายกับแม่ข่าย เมื่อมีการนำไปใช้งานสำหรับการเชื่อมต่อข้อมูลร่วมกันโปรแกรมการใช้งานต่างๆที่ทำงานอยู่บนลูกข่ายจะทำการประมวลผลข้อมูลโดยตรงจากข้อมูลที่ได้จัดเตรียมไว้ในแต่ละแม่ข่ายและจะไม่เชื่อมต่อไปยังข้อมูลจากแม่ข่ายหลายชุดในเวลาเดียวกัน ซึ่งเป็นการสะดวกที่จะกำหนดให้แต่ละแม่ข่ายจัดเตรียมข้อมูลจากอุปกรณ์หรือระบบควบคุมที่แตกต่างกันไป สามารถแสดงได้ในรูปที่ 2.27



รูปที่ 2.31 การอ่านข้อมูลจากแม่ข่ายหลายชุด

ข้อดีอีกข้อหนึ่งของโครงสร้างเครือข่ายแบบลูกข่ายกับแม่ข่ายสามารถกำหนดให้ลูกข่ายหลายชุดเชื่อมต่อเข้ากับแม่ข่ายได้ในเวลาที่ต้องการ ความสามารถนี้จะมีประสิทธิภาพอย่างมากในการใช้ข้อมูลร่วมกันของระบบควบคุม ดังแสดงได้ในรูปที่ 2.28

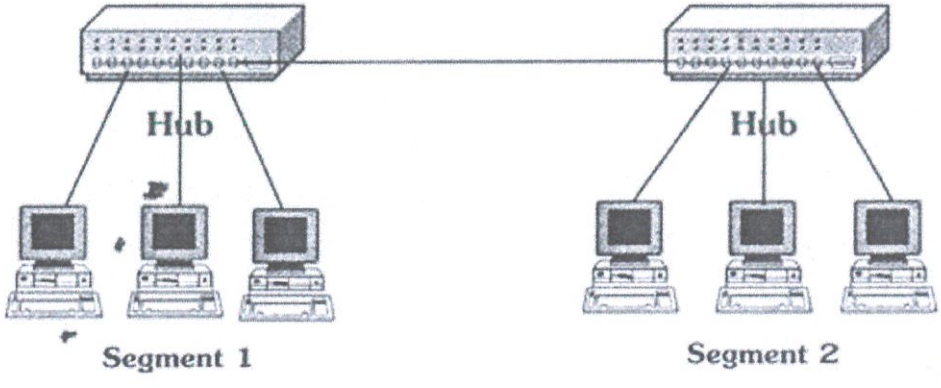


รูปที่ 2.32 การประมวลผลข้อมูลจากแม่ข่ายเดียวกัน

โดยทั่วไปในการควบคุมอุตสาหกรรมกระบวนการผลิตต่างๆ จะแบ่งระบบควบคุมออกเป็น 2 ชนิดคือระบบควบคุมพื้นฐานหรือระบบ DCS และระบบวัดคุมนิรภัย (Safety Instrumented System: SIS) หรือระบบ ESD (Emergency Shut Down system) ซึ่งทั้งสองระบบจะทำงานที่แยกเป็นอิสระต่อกัน ในการใช้งานจริงแล้วระบบวัดคุมนิรภัยจะต้องมีการรับส่งข้อมูลระหว่างระบบควบคุมพื้นฐานตลอดเวลาเนื่องจากระบบวัดคุมนิรภัยจะไม่มีหน่วยแสดงผลเพราะไม่ได้มีวัตถุประสงค์ในการควบคุมกระบวนการผลิต แต่ถึงอย่างไรก็ตามผู้ปฏิบัติการก็ยังต้องมีการตรวจสอบค่าตัวแปรต่างๆจากกระบวนการผลิตในขณะที่เกิดเหตุการณ์ผิดปกติขึ้น ดังนั้นในการออกแบบระบบควบคุมกระบวนการผลิตจึงกำหนดให้ระบบวัดคุมนิรภัยต้องส่งข้อมูลต่างๆที่ต้องการไปแสดงผลยังหน่วยแสดงผลของระบบ DCS โดยผ่านเครือข่ายหลักของระบบควบคุมผ่านการเชื่อมต่อทางโปรแกรม ซึ่ง OPC ก็เป็นทางเลือกหนึ่งที่นิยมใช้กันอย่างแพร่หลายในปัจจุบัน แต่ถ้าเป็นข้อมูลที่เป็นคำสั่งต้องทำการเชื่อมต่อกันด้วยสายไฟเท่านั้นเพื่อความปลอดภัยในการควบคุมกระบวนการผลิต

จากที่กล่าวมาแล้วข้างต้นการสั่งการใดๆจากระบบควบคุมหนึ่งไปยังอีกระบบควบคุมหนึ่งสามารถทำได้โดยผ่านการเชื่อมต่อ OPC ซึ่งในการออกแบบระบบควบคุมสำหรับอุตสาหกรรมกระบวนการผลิตแล้วจะหลีกเลี่ยงการใช้งานรูปแบบนี้ เนื่องจากคำสั่งที่ใช้ในการควบคุมอุปกรณ์ต่างๆในอุตสาหกรรมกระบวนการผลิตต้องการเวลาในการรับส่งข้อมูลที่แน่นอนและมีความเชื่อมั่นสูง และความเร็วของการรับส่งข้อมูลผ่านเครือข่ายการสื่อสารนั้นจะขึ้นอยู่กับปริมาณข้อมูลที่สื่อสารอยู่บนเครือข่าย ดังนั้นถ้าคำสั่งจากต้นทางไปยังอุปกรณ์ปลายทางล่าช้ากว่าเวลาที่ต้องการก็อาจจะทำให้เกิดเหตุการณ์อันตรายขึ้นได้

ในการเชื่อมต่อด้วยมาตรฐาน OPC นั้นเราใช้อุปกรณ์ตัวกลางในการสื่อสารคือ ฮับ (Hub) (รูปที่ 2.29) โดยฮับหรือรีพีตเตอร์ (Repeater) คืออุปกรณ์ที่ใช้เชื่อมต่อกลุ่มของคอมพิวเตอร์ฮับ มีหน้าที่รับส่งเฟรมข้อมูลทุกเฟรมที่ได้รับจากพอร์ตใดพอร์ตหนึ่งไปยังๆทุกๆพอร์ตที่เหลือ คอมพิวเตอร์ที่เชื่อมต่อเข้าฮับ จะแชร์แบนด์วิธหรืออัตราข้อมูลของเครือข่าย ฉะนั้นยิ่งมีคอมพิวเตอร์เชื่อมต่อเข้าฮับมากเท่าใด ยิ่งทำให้แบนด์วิธต่อคอมพิวเตอร์แต่ละเครื่องลดลง ในท้องตลาดปัจจุบันมีฮับหลายชนิดจากหลายบริษัท ข้อแตกต่างระหว่างฮับเหล่านี้ก็เป็นจำพวกพอร์ต สายสัญญาณที่ใช้ ประเภทของเครือข่าย และอัตราข้อมูลที่ฮับรองรับได้การที่อุปกรณ์เครือข่ายอีเธอร์เน็ตสามารถทำงานได้ที่ความเร็ว 2 ระดับ เช่น 10/100 Mbps นั้น ก็เนื่องจากอุปกรณ์เครื่องนั้นมีฟังก์ชันที่สามารถเช็คได้ว่าอุปกรณ์ หรือคอมพิวเตอร์ที่มาเชื่อมต่อกับฮับนั้นสามารถรับส่งข้อมูลได้ที่ความเร็วสูงสุดเท่าใด และอุปกรณ์นั้นก็จะเลือกอัตราข้อมูลสูงสุดที่รองรับทั้งสองฝั่ง ฟังก์ชันนี้จะเรียกว่า "การเจรจาอัตโนมัติ (Auto-Negotiation)" ส่วนใหญ่ฮับหรือสวิสต์ที่ผลิตจะมีฟังก์ชันนี้อยู่ เพื่อให้สามารถเชื่อมต่อเครือข่ายอีเธอร์เน็ตที่ความเร็วต่างกันได้ ถ้ามีอุปกรณ์เครือข่าย หรือคอมพิวเตอร์หลาย ๆ เครื่องเชื่อมต่อเข้ากับฮับและแต่ละโหนดสามารถส่งข้อมูลได้ในอัตราที่ต่างกัน ฮับก็จะเลือกอัตราส่งข้อมูลที่อัตราความเร็วต่ำสุด เนื่องจากคอมพิวเตอร์เหล่านี้จัดอยู่ในคอลลิชั่นโดเมน (Collision Domain) เดียวกัน ตัวอย่างเช่น ถ้า LAN การ์ดของคอมพิวเตอร์เครื่องหนึ่งสามารถรับส่งข้อมูลได้ที่ 10 Mbps ส่วน LAN การ์ดของคอมพิวเตอร์ที่เหลือสามารถรับส่งข้อมูลได้ 10/100 Mbps แล้วคอมพิวเตอร์เหล่านี้เชื่อมต่อเข้ากับฮับเดียวกันที่รองรับอัตราความเร็วที่ 10/100 Mbps เครือข่ายนี้ก็จะทำงานที่ความเร็ว 10 Mbps เท่านั้น แต่ถ้าเป็นสวิทช์อัตราความเร็วจะขึ้นอยู่กับความเร็วของคอมพิวเตอร์เนื่องจากสวิทช์จะแยกคอลลิชั่นโดเมน



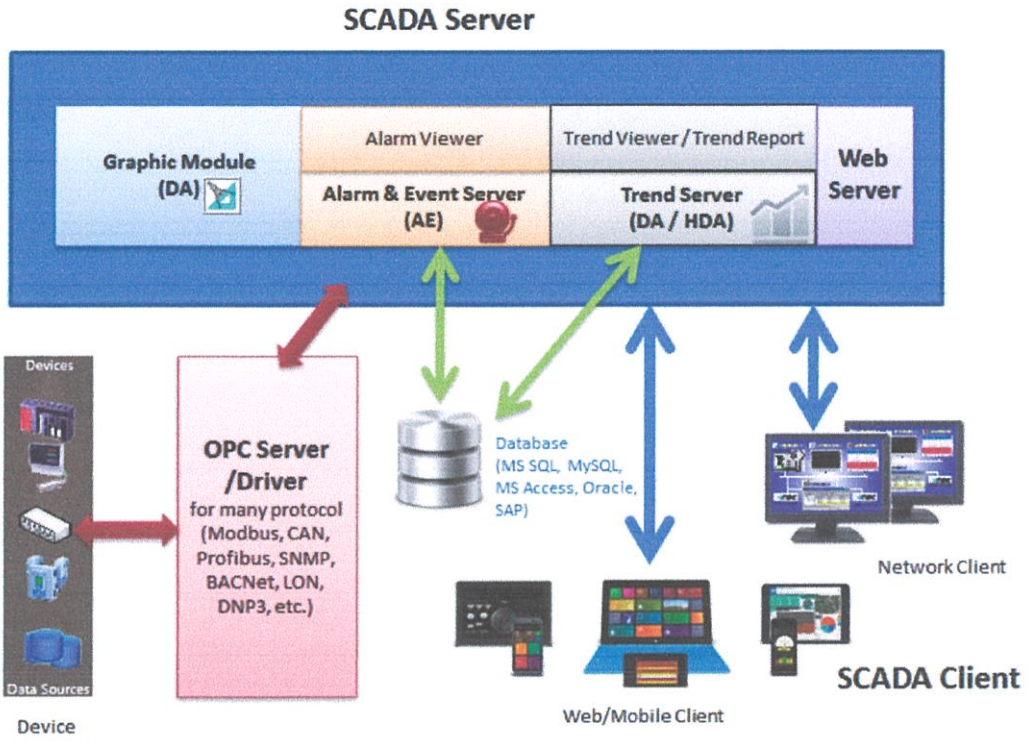
รูปที่ 2.33 แสดงการเชื่อมต่อคอมพิวเตอร์เข้ากับเครือข่ายโดยใช้ฮับ

2.2.3 ระบบ SCADA

SCADA นั้นย่อมาจากคำว่า Supervisory Control And Data Acquisition เป็นระบบตรวจสอบและวิเคราะห์ข้อมูลแบบ Real-time ใช้ในการตรวจสอบสถานะตลอดจนถึงควบคุมการทำงานของระบบควบคุมในอุตสาหกรรมและงานวิศวกรรมต่างๆ เช่น งานด้านโทรคมนาคม สื่อสาร การประปา การบำบัดน้ำเสีย การจัดการด้านพลังงาน อุตสาหกรรมการกลั่นน้ำมันและก๊าซ อุตสาหกรรมเคมี อุตสาหกรรมประกอบยนต์ การขนส่ง กระบวนการนิวเคลียร์ในโรงไฟฟ้า เป็นต้น ตัวอย่างการใช้งานเช่นใช้ SCADA ตรวจสอบข้อมูลเช่นการรั่วไหลของของเหลวที่เกิดขึ้นในท่อขนส่งจากตัวตรวจจับแล้วส่งสัญญาณแจ้งเตือนให้พนักงานทราบ โดยส่งข้อมูลสู่ส่วนกลางของระบบ SCADA เป็นต้นนอกจากนั้น SCADA อาจทำหน้าที่คำนวณและประมวลผลข้อมูลที่ได้จากฮาร์ดแวร์ต่างๆ เช่น ตัวควบคุมพีแอลซี, เครื่องควบคุม , DCS, RTU แล้วแสดงข้อมูลทางหน้าจอ หรือส่งสัญญาณควบคุมฮาร์ดแวร์ดังกล่าว เช่นหากอุณหภูมิของอุปกรณ์สูงเกินพิกัดให้ทำการปิดอุปกรณ์นั้นเป็นต้น โดยสั่งงานผ่าน ตัวควบคุมพีแอลซี หรือ เครื่องควบคุม ที่ติดต่ออยู่ ทั้งนี้ SCADA สามารถเก็บรวบรวมข้อมูลที่ได้จากระบบควบคุมทั้งหมดไว้ในฐานข้อมูลเพื่อให้พนักงานหรือโปรแกรมอื่น ๆ สามารถนาไปใช้งานได้ SCADA นั้นเข้าไปมีส่วนในงานควบคุมทั้งเล็กและใหญ่ที่ต้องการแสดงผลแลกเปลี่ยนข้อมูล หรือควบคุมระบบต่าง ๆ จากส่วนกลาง เพื่อการทางานของระบบรวมที่สัมพันธ์กันมองเห็นภาพรวมได้อย่างชัดเจนและมีความรวดเร็วต่อเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ระบบ SCADA ในปัจจุบันมีความสามารถในการสื่อสาร ควบคุม และประมวลผลข้อมูลจาก I/O ของอุปกรณ์เช่น ตัวควบคุมพีแอลซี, DCS, RTU ได้ถึงระดับที่เกินหนึ่งแสน I/O แล้ว และได้รับการพัฒนาให้มีความสามารถรองรับความต้องการใหม่ ๆ ของผู้ใช้งานอย่างต่อเนื่องตลอดมา

2.2.3.1 โครงสร้างของ SCADA (Architecture)

SCADA แบ่งตามโครงสร้างฮาร์ดแวร์ได้สองระดับคือ Client และ Data Server หรือเรียกสั้น ๆ ว่าServer โดยที่ Client คือคอมพิวเตอร์ที่รับและส่งข้อมูลไปยัง Data Server โดยฝั่ง Client นี้จะแสดงผลการทางานของระบบควบคุมเช่น แสดงเป็นกราฟิก กราฟแบบต่อเนื่อง หรือระบบแจ้งเตือนเมื่อเกิดเหตุการณ์ฉุกเฉินหรือต้องการแจ้งเตือน เป็นต้น ฝั่ง Client สามารถสั่งงานควบคุมไปยัง Data Server เพื่อส่งสัญญาณไปยัง ตัวควบคุมพีแอลซี, DCS หรือ เครื่องควบคุม อีกทอดหนึ่ง ส่วน Data Server จะทาหน้าที่ติดต่อกับ ตัวควบคุมพีแอลซี, DCS, เครื่องควบคุม หรือ RTU ต่าง ๆ เพื่อรับสัญญาณและส่งสัญญาณไปยัง Client และรับการร้องขอจาก Client เพื่อควบคุมอุปกรณ์ ตัวควบคุมพีแอลซี และ เครื่องควบคุม ต่าง ๆ Client และ Data Server ส่วนใหญ่ติดต่อกันผ่านระบบเครือข่าย Ethernet ดังรูปที่ 2.34

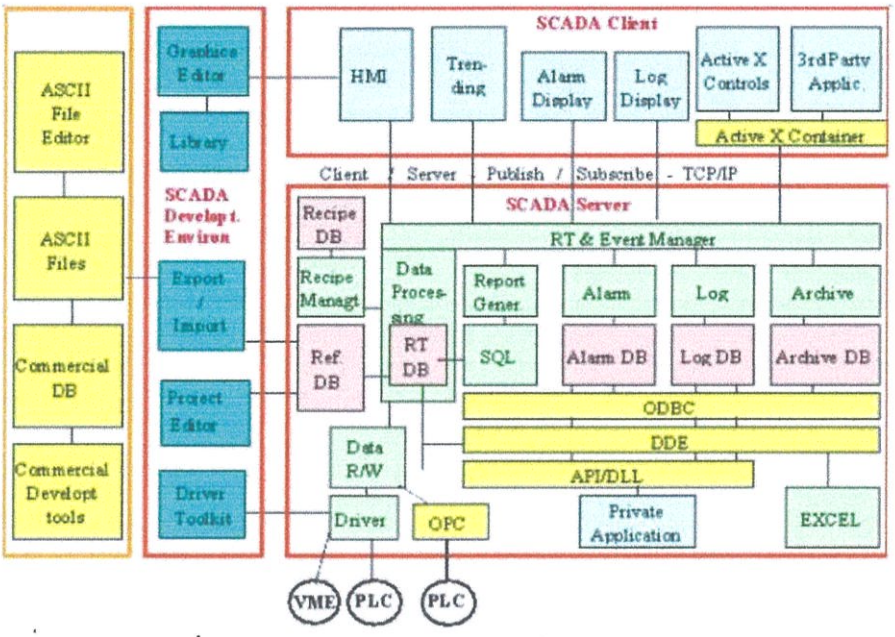


รูปที่ 2.34 โครงสร้างด้านฮาร์ดแวร์ของระบบ SCADA

ที่มา : http://www.eda.co.th/scada.html (09/04/2558)

2.2.3.2 โครงสร้างด้านซอร์ฟแวร์ (Software Architecture)

โครงสร้างด้านซอร์ฟแวร์ของระบบ SCADA นั้นมีข้อที่ต้องทราบคือ SCADA ใช้เทคโนโลยีในการสื่อสารกับฮาร์ดแวร์ (เช่น PLC, DCS) ต่าง ๆ กันไปตามผู้ผลิต เช่นการใช้ Driver เฉพาะของผู้ผลิต SCADA เพื่อสื่อสารกับ PLC, DCS เป็นต้น ซึ่งในปัจจุบันมีการกำหนดมาตรฐานกลางคือ OPC ขึ้นมาเพื่อยุติปัญหาการใช้เทคโนโลยีเฉพาะด้านในการสื่อสาร นอกจากนั้นยังมีความสามารถในการบริการข้อมูลให้กับ Client ที่รวดเร็วและมีเสถียรภาพ โครงสร้างด้านซอร์ฟแวร์ของ SCADA แสดงได้ดังรูปที่ 2.31



รูปที่ 2.35 โครงสร้างด้านซอร์ฟแวร์ของระบบ SCADA

ที่มา : http://www.scadathai.com/articles/475019/What's-SCADA.html (08/09/2558)

จากรูปที่ 2.35 สังเกตได้ว่าในส่วนของ SCADA Server นั้น การติดต่อกับ ตัวควบคุมพีแอลซี หรือ เครื่องควบคุม นั้น ทำได้ทั้งผ่าน Driver หรือ OPC โดยที่ OPC และ Driver สามารถรับคำสั่งแบบ Read / Write เพื่ออ่านข้อมูลจาก ตัวควบคุมพีแอลซี หรือ เขียนข้อมูลเพื่อสั่งงานไปยัง ตัวควบคุมพีแอลซี ได้

SCADA Server ทำหน้าที่จัดการข้อมูล RTDB (Real Time Data Base) ที่ได้จากตัวควบคุมพีแอลซี แล้วส่งให้กับ SCADA Client โดยที่ SCADA Server บางประเภทจะติดต่อกับ SCADA Client ผ่าน DDE Server ซึ่งทำให้สามารถนำเข้าข้อมูลจาก ตัวควบคุมพีแอลซี เข้าสู่โปรแกรมเช่น MS Excel หรือ โปแกรม Client อื่น ๆ ที่ติดต่อกับ DDE Server ได้ SCADA บางตัวจะออกแบบให้ SCADA Server ทำหน้าที่ตรวจจับ Alarm และเก็บไว้ใน Alarm DB หรือเก็บข้อมูลที่เป็น Historian ไว้ใน Log DB เป็นต้นเพื่อส่งให้ Alarm Display และ Log Display ทางฝั่ง SCADA Client ต่อไป

สำหรับส่วน Development Environment นั้นจะขึ้นอยู่กับการออกแบบของ SCADA ซอร์ฟแวร์นั้น ๆ ซึ่งโดยทั่วไปก็จะมีเครื่องมือในการสร้างและจัดการกราฟิก (Graphic Editor) เครื่องมือในการจัดการโปรเจ็คที่สร้างขึ้นมา (Project Editor) มีเครื่องมือในการนำเข้าและส่งออก Text file ที่เก็บค่าคอนฟิกูเรชั่นของการติดต่อกับ Driver หรือ OPC Server ไว้

2.2.3.3 โครงสร้างด้านการสื่อสาร (Communications)

การสื่อสารระหว่าง Client-Server จะสื่อสารผ่านโปรโตคอลโดยทั่วไปเช่น TCP/IP โดย Client จะติดต่อกับพารามิเตอร์หรือ Tag ภายใน Server ที่บริการข้อมูลด้วยรูปแบบที่แตกต่างกันไปตามผู้ผลิต เช่นมีการส่งค่าจาก Server เมื่อค่าของ I/O ของ ตัวควบคุมพีแอลซี มีการเปลี่ยนแปลง เป็นต้น

การสื่อสารกับอุปกรณ์นั้น Server จะทำการตรวจสอบค่าจากอุปกรณ์ตามช่วงเวลาที่ผู้ใช้งานกำหนดไว้ (Defined polling rate) โดยอาจจะต่างกันไปตามพารามิเตอร์ประเภทต่าง ๆ โดยตัว Controller จะส่งค่าพารามิเตอร์ตามที่ถูกร้องขอให้กับ Data Server พร้อมค่าเวลาขณะนั้น (Time Stamp) การสื่อสารกับอุปกรณ์ของ Data Server นั้นอาจเป็นการสื่อสารแบบ Modbus, Profibus, CAN bus เป็นต้น ขึ้นอยู่กับมาตรฐานการสื่อสารของอุปกรณ์นั้น ๆ ว่าเป็นแบบใด ในปัจจุบันมีการสร้าง OPC Server ที่สนับสนุนการติดต่อด้วยมาตรฐานต่างๆเพิ่มขึ้นมากมายจนครอบคลุมอุปกรณ์ทุกประเภท และมีการพัฒนาให้ทั่วถึงไปยังอุปกรณ์ใหม่ ๆ อย่างต่อเนื่อง

2.2.3.4 โครงสร้างอินเทอร์เฟส (Interface)

การติดต่อระหว่าง Data Server กับอุปกรณ์หรือระหว่าง Data Server และ Data Server และกับ Client นั้น มีการผลิตเป็น Driver ออกมามากมายตามเทคนิคเฉพาะของแต่ละผู้ผลิต ต่อมาจึงมีการกำหนดมาตรฐานของอินเทอร์เฟสขึ้นมาเป็น OPC (OLE for Process Control) ซึ่งมีความรวดเร็วในการสื่อสารและบริการข้อมูลโดยมีการจัดตั้ง OPC Foundation ขึ้นเป็นองค์กรหลักในการกำหนดมาตรฐานและถ่ายทอดเทคโนโลยีให้แก่สมาชิก OPC จึงเป็นมาตรฐานกลางที่เปิดกว้างมากที่สุด

การติดต่อกับฐานข้อมูลภายนอกของ SCADA Software นั้น มีการสร้างให้สามารถติดต่อได้ผ่าน ODBC (Open Data Base Connectivity), OLEDB (Linking and Embedding Data Base), DDE (Dynamic Data Exchange) เป็นต้น เพื่อให้สามารถแลกเปลี่ยนข้อมูลหรือทำการเก็บข้อมูลไว้ในฐานข้อมูลรูปแบบต่าง ๆ ในปัจจุบันมีการพัฒนาให้สามารถติดต่อกับโปรแกรม ERP ต่าง ๆ เช่น SAP เป็นต้นได้ด้วย

2.2.3.5 โครงสร้างความสามารถในการขยายระบบ (Scalability)

Scalability คือความสามารถในการรองรับและต่อขยายระบบ SCADA กับส่วนต่าง ๆ เช่น I/O ของอุปกรณ์Controller และจำนวนเครื่อง SCADA Client ที่เพิ่มขึ้น หรือการต่อพ่วงกับระบบ SCADA ของยี่ห้ออื่น ๆ เป็นต้น ถ้าหาก Data Server เป็นแบบ Driver ที่สร้างด้วยเทคโนโลยีเฉพาะในการติดต่อกับอุปกรณ์ ก็เป็นเรื่องลำบากในการต่อขยาย เพราะ Driver บาง

ประเภทสามารถติดต่อได้เฉพาะ SCADA Software บางยี่ห้อเท่านั้น ปัญหานี้เป็นที่วิพากย์ วิจารณ์กันอย่างกว้างขวาง ซึ่งปัจจุบันได้หันมาใช้มาตรฐานกลางคือ OPC เพื่อแก้ไขปัญหานี้

2.2.3.6 โครงสร้างการสำรองระบบ(Redundancy)

SCADA Software ส่วนใหญ่มีความสามารถในการทาสารองระบบของ Data Server โดยที่เมื่อ Data Server เกิดความขัดข้องก็จะสั่งงานให้ Data Server อีกตัวหนึ่งทำงานแทนที่ โดยจะมีการกำหนดคอนฟิกูเรชั่นไว้ที่ Client ว่าจะให้เลือกติดต่อกับ Data Server ตัวไหนเมื่อเกิดความขัดข้องเกิดขึ้น

ในบางครั้งโมดูลที่ทาหน้าที่จัดการด้าน Redundancy นี้อาจจะทาหน้าที่อีกประการหนึ่งคือเป็นจุดพักข้อมูลที่รับมาจาก Data Server เพื่อนำไปส่งให้กับ Client ต่าง ๆ เพราะในกรณีที่มี Client จำนวนมากติดต่ออยู่กับ Data Server ตัวเดียวนั้นอาจมีความล่าช้าในการบริการข้อมูลของ Data Server เพราะต้องให้บริการข้อมูล Client ให้ครบจำนวนก่อนที่จะไปรับข้อมูลใหม่จากอุปกรณ์มาได้ ดังนั้นโมดูลที่ทาหน้าที่ Redundant จึงทำหน้าที่เป็นจุดรับข้อมูลแล้วช่วยส่งต่อให้ Client ต่างๆ อีกทอดหนึ่ง Data Server จะได้ทำหน้าที่บริการข้อมูลให้แก่โหนดเพียงจุดเดียวจึงมีความรวดเร็วในการบริการข้อมูล

2.2.3.7 หน้าที่การทำงาน(Functionality)

การเข้าถึงพารามิเตอร์ของอุปกรณ์ หมายถึงความสามารถในการเข้าถึงกลุ่มของพารามิเตอร์ในอุปกรณ์เช่น I/O ของ ตัวควบคุมพีแอลซี เป็นต้น ความสามารถของ Data Server ในการกำหนดว่าพารามิเตอร์ใด อ่านได้อย่างเดียว เขียนได้อย่างเดียว หรือทั้งอ่านทั้งเขียนเป็นต้น

2.2.3.8 ระบบแสดงผลแบบ MMI (Man Machine Interface)

คือความสามารถในการแสดงผลการทางานของอุปกรณ์ในรูปแบบ กราฟิก ข้อความ สัญลักษณ์ แผนภาพ เป็นต้น โดยสามารถเชื่อมโยงลักษณะการเปลี่ยนแปลงของกราฟิกเหล่านี้กับพารามิเตอร์จาก Data Serverได้ ความสามารถในการสั่งงานผ่านระบบกราฟิกเช่น การปิด/เปิด สวิทซ์บนจอมอนิเตอร์ส่งผลไปยัง I/O ของ ตัวควบคุมพีแอลซี เป็นต้น

ความสามารถในการจัดการกราฟิกเช่น การย่อ ขยาย การกำหนดการเคลื่อนไหวแบบต่าง ๆ เช่น การหมุน การเคลื่อนที่แบบซิกแซ็กตามสัญญาณของ Data Server การแสดงผลสัญญาณในรูปแบบมิเตอร์และเกจวัดแบบต่าง ๆ การนำเข้ากราฟิกประเภทต่างๆ การจัดแบ่งเลเยอร์ เป็นต้น สิ่งเหล่านี้เป็นข้อเปรียบเทียบความสามารถของ SCADA Software ทั้งสิ้น

2.2.3.9 ระบบแสดงกราฟสัญญาณแบบต่อเนื่อง (Trending)

Trending เป็นความสามารถในการพล็อตกราฟต่อเนื่องกันไปบนจอภาพ เพื่อแสดงค่าสัญญาณจาก Data Server โดยอาจจะสามารถพล็อตสัญญาณได้หลายสัญญาณเช่น 8 – 24 สัญญาณ พร้อมกันในหน้าต่างเดียว เพื่อให้สามารถเปรียบเทียบสัญญาณที่พล็อตได้ และไม่จำกัดว่าจะสร้างหน้าต่างพล็อตจานวนเท่าใด Trending อาจมีความสามารถในการซูมสัญญาณที่พล็อตและหยุดการพล็อตเพื่อเลื่อนดูค่าที่พล็อตในแต่ละช่วงเวลาได้ด้วยตัวของผู้ใช้งานเอง นอกจากนั้นการพล็อตอาจสามารถเลือกได้ว่าจะให้เป็นการพล็อตแบบใดเช่น Time plot, Logarithmic plot, Strip Chart, Bar Chart, Circular, X-Y plot เป็นต้น นอกจากนั้นบางผู้ผลิตยังสามารถนาค่า Historian หรือข้อมูลสัญญาณที่เก็บไว้ในฐานข้อมูลออกมาพล็อต ได้อีกด้วย โดย Trending Module นี้อาจเป็นแบบ ActiveX Control คือสามารถนำไปใช้งานในแอพพลิเคชั่นอื่นที่สนับสนุนการนำเข้า ActiveX ได้

2.2.3.10 ระบบแจ้งเตือน(Alarm)

SCADA Software ส่วนใหญ่มีระบบแจ้งเตือนโดย Alarm Display จะรับสัญญาณมาจาก Alarm DB ในฝั่ง SCADA Server โดย Alarm DB สามารถที่จะทำการกำหนดคอนฟิกูเรชั่นว่าจะนำสัญญาณตัวใดมาเป็นตัวพารามิเตอร์ในการแจ้งเตือนบ้าง และมีการแบ่งระดับของ Priority, Limit อย่างไร เป็นต้นระบบแจ้งเตือนยังสามารถที่จะเก็บข้อมูลการแจ้งเตือนไว้ในฐานข้อมูลประเภทต่าง ๆ ได้เช่น MS SQL Server, MS Access, Oracle, MS Excel เป็นต้น และบางยี่ห้อสามารถแสดงออกมาเป็นรายงานในรูปแบบตารางหรือแผนภูมิได้อีกด้วย

2.2.3.11 การทำงานแบบ Automation

เป็นความสามารถที่ SCADA ทำหน้าที่ต่าง ๆ ตามที่กำหนด เช่น ส่งอีเมล์ แสดงข้อความแบบ Instance Message บนหน้าจอ เปิดไปยังหน้าจออื่น ๆ เก็บข้อมูลลงฐานข้อมูล เปิดโปรแกรม หรือรันค่าสั่งสคริปต์ เป็นต้น ตามสัญญาณที่ได้รับจาก Data Server และข้อกำหนดที่สร้างขึ้น

2.2.3.12 การสร้างและพัฒนา (Application Development)

การกำหนดคอนฟิกูเรชั่น ขั้นแรกต้องมีการกำหนดว่าจะติดต่อกับพารามิเตอร์หรือ Tag ใดบ้างจาก Data Server ดังนั้นจะต้องทำการ Define หรือสร้าง Tag ที่ Data Server ก่อนว่า Tag แต่ละตัวหมายถึง Address ที่เท่าใดของอุปกรณ์ (ตัวควบคุมพีแอลซี, DCS, RTU, เครื่องควบคุม ต่างๆ) โดยทั่วไปสามารถทำการนำเข้าคอนฟิกูเรชั่นไฟล์ที่สร้างไว้ก่อนเข้ามาได้ และสามารถ Export ไปยัง Data Server อื่น ๆ ได้ จากนั้นโปรแกรมย่อยอื่น ๆ ของ SCADA Software ฝั่งไคลเอนท์ จึงทำคอนฟิกูเรชั่นตามหน้าที่การทำงานของตนเอง เช่น โมดูลที่มีหน้าที่

แสดงผลกราฟิกก็ต้องกำหนดว่ากราฟิกนั้น ๆ จะเชื่อมโยงกับ Tag ได้จาก Data Server ส่วนโมดูลที่ทาหน้าที่แจ้งเตือนก็ต้องทำคอนฟิกูเรชั่นว่าจะนำ Tag ใด มาเป็นสัญญาณแจ้งเตือน และกำหนดระดับสัญญาณLimitเป็นต้น

2.2.3.13 เครื่องมือในการพัฒนา (Development Tool)

เครื่องมือในการสร้างและพัฒนาระบบ SCADA โดยทั่วไปจะประกอบด้วย

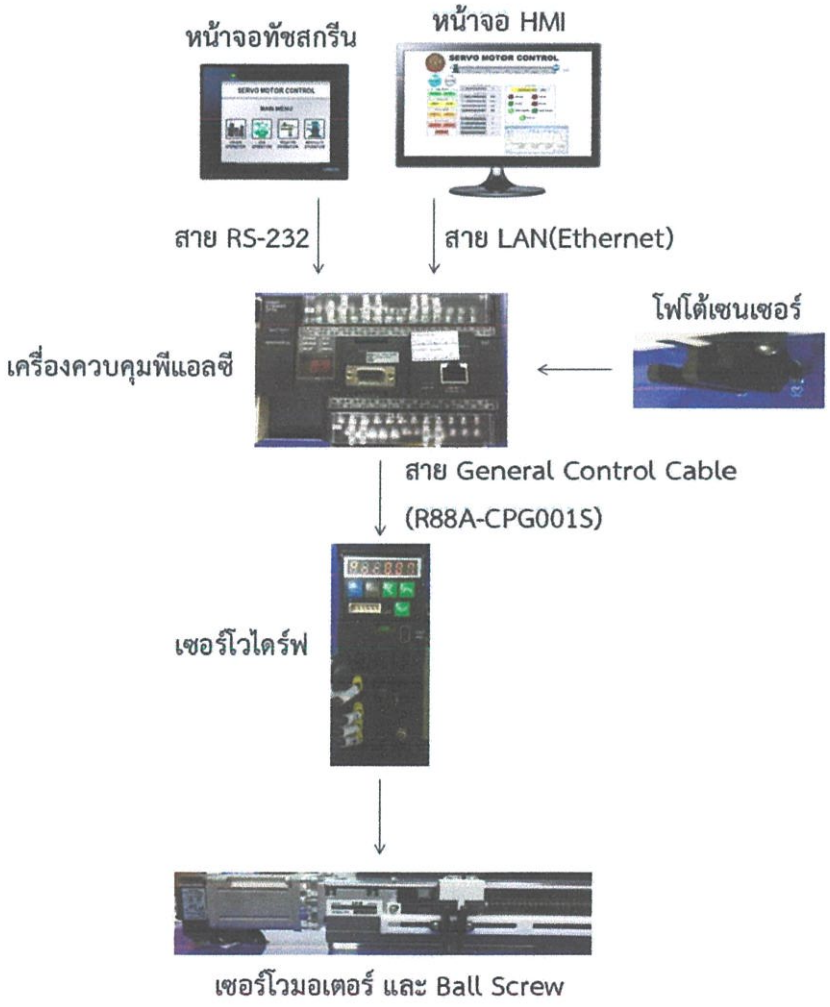
1. เครื่องมือในการสร้างระบบกราฟิก ที่ประกอบด้วยเครื่องมือวาดภาพ เครื่องมือกำหนดเอฟเฟ็กพิเศษต่าง ๆ ไลบรารี่ของกราฟิกสาเร็จรูปในอุตสาหกรรมด้านต่าง ๆ
2. เครื่องมือในการสร้าง Trending
3. เครื่องมือในการสร้างระบบ Alarm
4. เครื่องมือในการกำหนดการติดต่อกับฐานข้อมูลเพื่อทาการเก็บรวบรวมข้อมูลของ Trending และ Alarm ลงไว้ในฐานข้อมูล
5. เครื่องมือในการช่วยสร้าง Script เช่น Java script, VB Script
6. เครื่องมือจัดการด้านความปลอดภัย การแบ่งระดับ User และขอบเขตการใช้งานของ User
7. เครื่องมือในการสร้าง Web application เพื่อให้สามารถควบคุมและตรวจสอบระบบควบคุมผ่าน Web browser ได้

ที่กล่าวมาทั้งหมดนี้เป็นลักษณะของ SCADA และ SCADA Software ส่วนใหญ่ ทั้งนี้คุณผู้อ่านก็คงจะพอเห็นภาพว่าSCADA นั้นสามารถเป็นศูนย์กลางของระบบควบคุมทั้งหมดขององค์กร และมีส่วนช่วยในการตรวจสอบการทางานของระบบให้เป็นไปตามปกติได้อย่างมีประสิทธิภาพและทั่วถึง ภายในเวลาอันรวดเร็ว มีส่วนช่วยในการตัดสินใจในการดาเนินงานจากข้อมูลต่าง ๆ ที่ได้รับจากระบบ SCADA นอกจากนี้เรายังสามารถเชื่อมโยงข้อมูลที่ได้จาก SCADA เข้ากับข้อมูลทางธุรกิจอื่น ๆ เพื่อประมวลผลร่วมกัน เช่น ข้อมูลจานวนของเสียเป็นกิโลกรัมที่ตรวจสอบได้จาก SCADA ถูกนำมาคำนวณร่วมกับค่าใช้จ่ายอื่น ๆ แบบ เรียลไทม์เพื่อสรุปเป็นรายงานค่าใช้จ่ายประจำวันเป็นต้นได้อย่างรวดเร็ว

บทที่ 3

การออกแบบและการทำงาน

ในการเริ่มต้นการออกแบบการจำลองการควบคุมเซอร์โวมอเตอร์ด้วยเครื่องควบคุม PLC ร่วมกับ HMI ผ่านระบบเครือข่าย Ethernet และมี OPC Server เป็นตัวกลางในการรับส่งข้อมูลนั้นต้องกำหนดการทำงานทั้งหมดและส่วนประกอบย่อย ซึ่งส่วนประกอบย่อยแบ่งเป็นสองส่วนหลักๆ คือ ส่วนแบบโครงสร้างการเชื่อมต่อของฮาร์ดแวร์ และ อีกส่วนหนึ่งคือ การคอนฟิกูเรชั่นโปรแกรมและกราฟิก



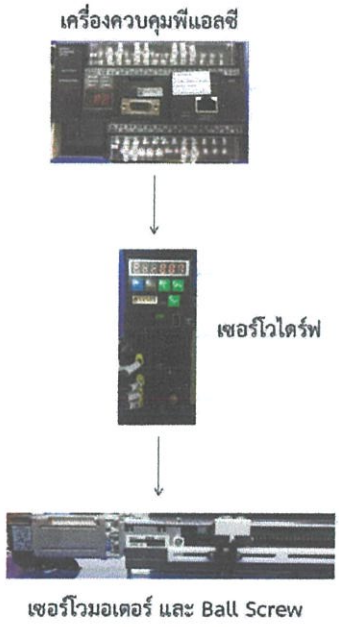
รูปที่ 3.1 การเชื่อมต่อทั้งหมดในงานวิจัย

3.1 การเชื่อมต่ออุปกรณ์ทางด้านฮาร์ดแวร์

การเชื่อมต่อของอุปกรณ์ต่างๆ สามารถอธิบายได้ดังนี้

3.1.1 การเชื่อมต่อระหว่างเซอร์โวมอเตอร์ กับ เครื่องควบคุมพีแอลซี

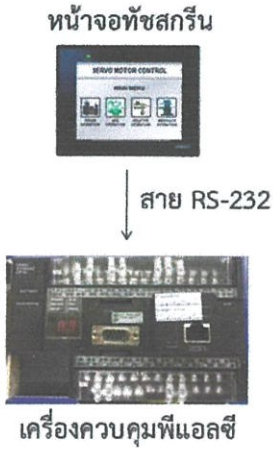
สำหรับสายควบคุมที่ใช้ในการต่อระหว่างเซอร์โวมอเตอร์ G5-series และ Omron CP1H คือ General Control Cable (R88A-CPG001S) เป็นสายสัญญาณที่มีเพียงคอนเนคเตอร์เชื่อมต่อกับ CN1 ของเซอร์โวไดรเวอร์เพียงด้านเดียว ส่วนอีกด้านเป็นสายเปล่าเพื่อนำไปต่อกับ CP1H



รูปที่ 3.2 การเชื่อมต่อระหว่างเซอร์โวมอเตอร์ กับ เครื่องควบคุมพีแอลซี

3.1.2 การเชื่อมต่อระหว่างหน้าจอทัชสกรีน กับ เครื่องควบคุมพีแอลซี

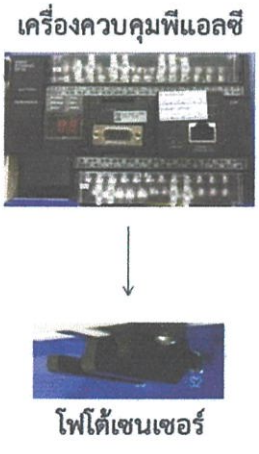
ในการติดต่อกันระหว่างหน้าจอทัชสกรีน กับ เครื่องควบคุมพีแอลซี นั้นจะเชื่อมต่อกันด้วยสาย RS-232 เพื่อสั่งการการควบคุม และ การแสดงผล



รูปที่ 3.2 การเชื่อมต่อระหว่างหน้าจอทัชสกรีน กับ เครื่องควบคุมพีแอลซี

3.1.3 การเชื่อมต่อระหว่างโฟโต้เซนเซอร์ กับ เครื่องควบคุมพีแอลซี

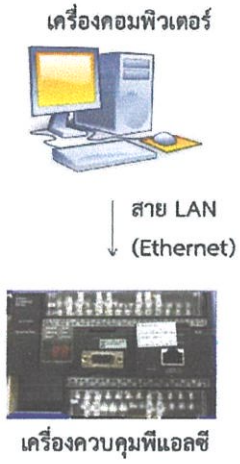
ในการเชื่อมต่อกันระหว่างโฟโต้เซนเซอร์ กับ เครื่องควบคุมพีแอลซีนั้นจะเชื่อมต่อกันด้วยสายไฟหรือสายสัญญาณจากโมดูลอินพุตและเอาท์พุต



รูปที่ 3.3 การเชื่อมต่อระหว่างโฟโต้เซนเซอร์ กับ เครื่องควบคุมพีแอลซี

3.1.4 การเชื่อมต่อระหว่างคอมพิวเตอร์ กับ เครื่องควบคุมพีแอลซี

ในการติดต่อกันระหว่างคอมพิวเตอร์ กับ เครื่องควบคุมพีแอลซี สามารถเชื่อมต่อกันได้โดยผ่านระบบอีเทอร์เน็ตซึ่งเป็นโปรโตคอลของแลน เพื่อทำการดาวน์โหลดโปรแกรมการควบคุมที่ได้เขียนไว้



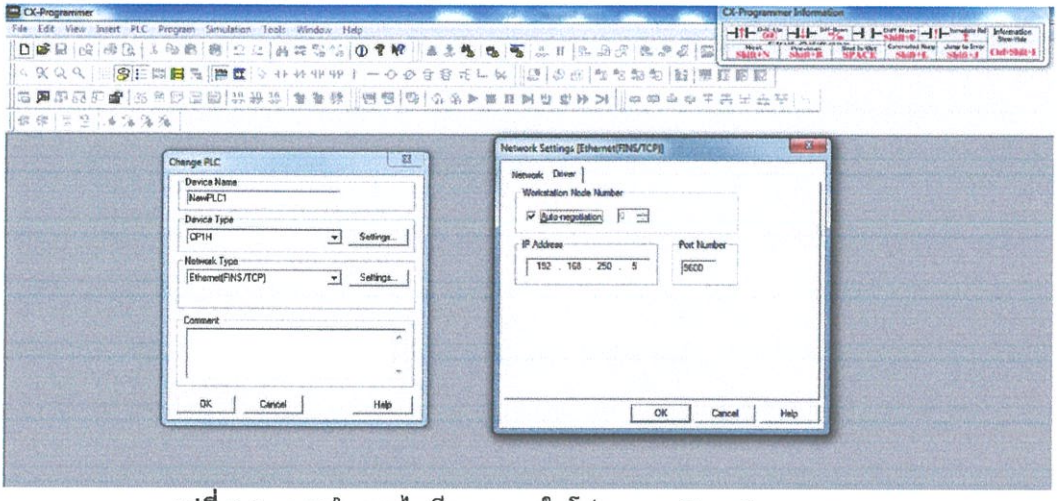
รูปที่ 3.4 การเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์ กับ เครื่องควบคุมพีแอลซี

3.2 การเชื่อมต่ออุปกรณ์ทางด้านซอฟต์แวร์

การเชื่อมต่ออุปกรณ์ซอฟต์แวร์แบ่งออกเป็น 3 ขั้นตอนคือ การตั้งค่าเครื่องควบคุมพีแอลซีกับคอมพิวเตอร์ การตั้งค่าโปรแกรมสำหรับ SCADA และการตั้งค่าโปรแกรมที่ใช้สำหรับการเชื่อมต่อ OPC ซึ่งรายละเอียดตามดังต่อไปนี้

3.2.1 การตั้งค่าเครื่องควบคุมพีแอลซี กับ คอมพิวเตอร์

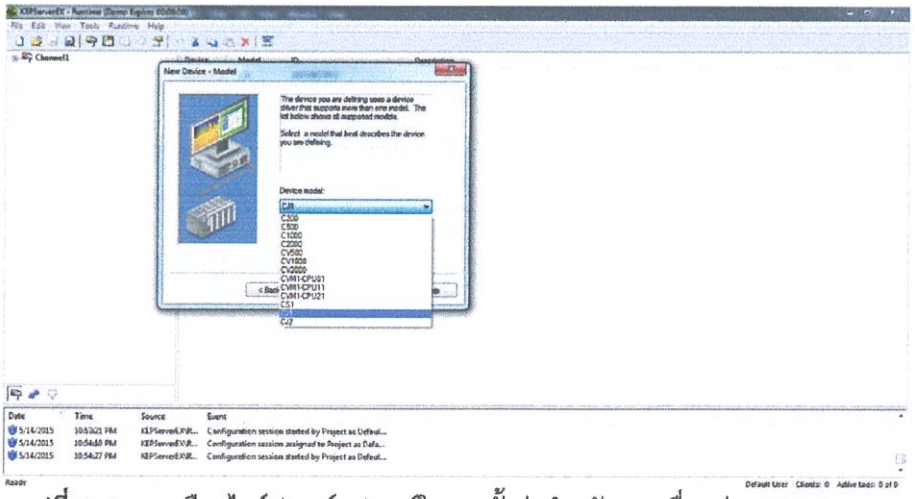
ในการตั้งค่าเครื่องควบคุมพีแอลซีเชื่อมต่อกับคอมพิวเตอร์ สามารถเชื่อมต่อกันได้ผ่านระบบอีเทอร์เน็ตซึ่งเป็นโปรโตคอลของแลน โดยใช้โปรแกรม Cx – Programmer ทำการกำหนดไอพีแอดเดรสของเครื่องควบคุมพีแอลซีและ คอมพิวเตอร์ เป็น 192.168.250.5 และ 192.168.250.30 ตามลำดับ แล้วทำการดาวน์โหลดโปรแกรมที่เขียนไว้ลงไปในเครื่องควบคุมพีแอลซี



รูปที่ 3.5 การกำหนดไอพีแอดเดรสในโปรแกรม Cx – Programmer ของเครื่องควบคุมพีแอลซี

3.2.2 การตั้งค่าโปรแกรมสำหรับการเชื่อมต่อ OPC Server

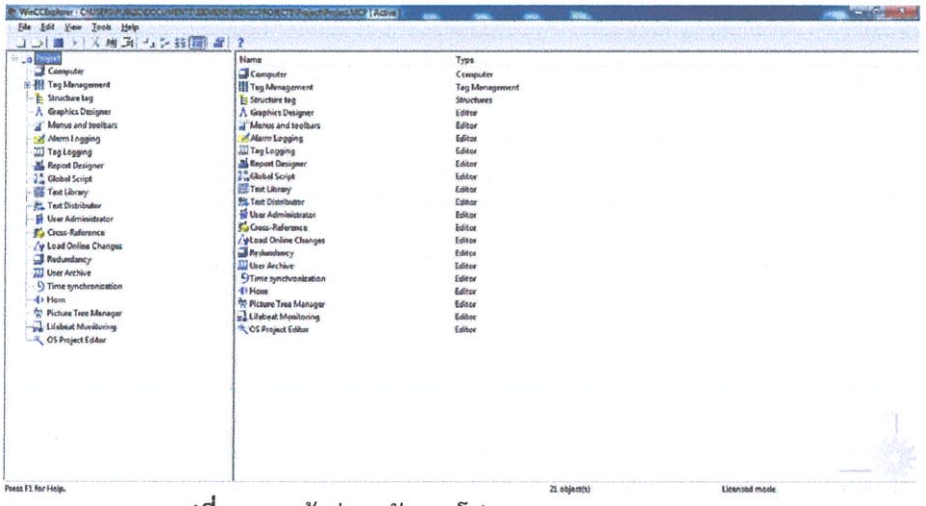
ในงานวิจัยนี้เราเลือกใช้โปรแกรม KEPServerEx V5 มาเป็น OPC Server ซึ่งมีประโยชน์ในการแลกเปลี่ยนข้อมูลกันระหว่าง SCADA พีแอลซี คอมพิวเตอร์ หรืออุปกรณ์อื่นๆ โดยเมื่อเปลี่ยนค่า OPC Tag ที่โหนดใดโหนดหนึ่ง โหนดอื่นๆของ SCADA ที่ติดต่อมาที่ OPC Server นี้ก็จะมีค่าที่ตรงกันกับที่เราเปลี่ยนแปลงนั้น โดยก่อนเริ่มใช้งานต้องมีการกำหนดโครงสร้างของฮาร์ดแวร์ที่เชื่อมต่อและสร้าง Tag ในแอดเดรสที่เราต้องการมอนิเตอร์ค่า ซึ่งสามารถสร้าง Tag ได้มากตามที่ต้องการ โดยจะเลือกดีไวซ์โมเดลเป็น CJ1 เนื่องจากโปรแกรมไม่มีดีไวซ์โมเดล CP1H จึงใช้ CJ1 แทนได้



รูปที่ 3.6 การเลือกไดร์ฟเวอร์อุปกรณ์ในการตั้งค่าสำหรับการเชื่อมต่อ OPC Server

3.2.3 การตั้งค่าโปรแกรมสำหรับการเชื่อมต่อ SCADA

ในการเขียนกราฟิก SCADA สำหรับพีแอลซี Omron ในงานวิจัยนี้เลือกใช้โปรแกรม WinCC (Window Control Center) เวอร์ชั่น 7.0 ซึ่งเป็นซอฟท์แวร์ที่ใช้ในการควบคุมงานทางด้านการผลิตและกระบวนการอัตโนมัติ ผ่านหน้าจอคอมพิวเตอร์ ภายในซอฟท์แวร์ประกอบโมดูลที่ใช้จัดการทางด้านกราฟิก แมสเซสไฟล์ข้อมูล และการทำรายงาน ซึ่งเหมาะสมสำหรับงานทางด้านอุตสาหกรรมการผลิตต่างๆทั่วไป ในขั้นตอนการทดลองเริ่มจากการกำหนดไดร์ฟเวอร์อุปกรณ์ โดยขั้นตอนการกำหนดโครงสร้างต่างๆได้กล่าวไว้ในภาคผนวก ต่อมาคือเขียนกราฟิกและจัดการ Tag ต่างๆ โดย Tag ใน WinCC เป็นได้ทั้งตัวแปรค่าที่ใช้งานจริง เช่น ค่าระดับน้ำแทงค์ หรือเป็นได้ทั้งตัวแปรค่าที่ใช้เฉพาะภายในโปรแกรมที่ใช้ในการคำนวณหรือการจำลองการทำงาน ซึ่งจะแบ่ง Tag ออกเป็นสองประเภทคือ Tag ที่ตัวแปรค่าที่ใช้งานจริงค่าของมันถูกส่งมาจากหน่วยความจำของพีแอลซีหรืออุปกรณ์อื่นๆที่ติดต่อกับโปรแกรมผ่านสายสัญญาณหรือระบบสื่อสารต่างๆนั้นคือ Process Tags และอีกแบบคือ Internal Tags เป็นตัวแปรค่าที่ใช้เฉพาะภายในตัวโปรแกรมใช้ในการคำนวณหรือการจำลองการทำงาน ซึ่งสามารถจัดกลุ่มของ Tags ได้โดยสร้าง Tags Groups



รูปที่ 3.7 หน้าต่างหลักของโปรแกรม WinCC Explorer

3.3 การกำหนดค่า Electronic Gear Integer Setting และ ชนิดอินพุตของเซอร์โวไดรเวอร์

ในงานวิจัยนี้เลือกใช้สัญญาณอินพุตแบบ Pulse Train Control ตัวควบคุมพีแอลซีจะส่งสัญญาณเป็นพัลส์ให้กับเซอร์โวไดรฟ์เวอร์จากนั้นไดรฟ์เวอร์จะควบคุมการหมุนของมอเตอร์ให้ได้ตามการสั่งงานของตัวควบคุมพีแอลซีระยะในการเคลื่อนที่หรือการหมุนของเซอร์โวมอเตอร์จะขึ้นอยู่กับจำนวน

พัลส์ที่ตัวควบคุมพีแอลซีส่งให้ ขณะเดียวกันความเร็วของเซอร์โวมอเตอร์จะขึ้นอยู่กับความเร็วหรือความถี่ของพัลส์ที่ส่งมาจากตัวควบคุมพีแอลซี

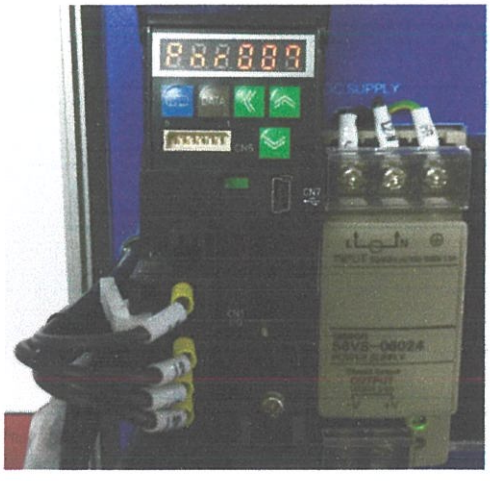
การ Configure Parameter ของเซอร์โวไดร์ฟ กำหนดค่า Parameter 4 ตัว คือ

1. Pn007 : Command Pulse mode Selection
2. Pn008 : Electronic Gear Integer Setting
3. Pn009 : Electronic Gear Ratio(Numerator)
4. Pn010 : Electronic Gear Ratio(Denominator)

การตั้งความละเอียดของจำนวนพัลส์ต่อรอบ ผู้ใช้สามารถกำหนดได้เองตามความต้องการโดยใช้วิธีการตั้งค่า Gear Ratio

Gear Ratio คือ การกำหนดความละเอียดในการเคลื่อนที่ของเซอร์โวมอเตอร์ ซึ่งเป็นการกำหนดความสัมพันธ์ระหว่าง สัญญาณป้อนกลับที่รับมาจากเอ็นโค้ดเดอร์กับจำนวนพัลส์ที่ส่งมาจากตัวควบคุมพีแอลซี ทำให้เราสามารถทราบระยะทางการเคลื่อนที่ได้อย่างถูกต้องตรงกับค่าจริง

ในงานวิจัยนี้ทำการ Configure Parameter ของเซอร์โวไดร์ฟ โดยกำหนดค่า Pn007 = 3 , Pn008 = 4000 , Pn009 = 0 , Pn010 = 0 และ Ball Screw พิตละ 4 มิลลิเมตร ผลตอบสนองที่แสดงออกมา คือ ถ้าตัวควบคุมพีแอลซีทำการจ่ายพัลส์ไปให้เซอร์โวไดร์ฟ 4000 พัลส์ มอเตอร์จะทำการหมุน 1 รอบ และ Ball Screw หมุน 1 รอบ แสดงว่า เมื่อตัวควบคุมพีแอลซีจ่ายพัลส์ออกมา 4000 พัลส์ Ball Screw จะเคลื่อนที่ไปได้ระยะทาง 4 มิลลิเมตร



รูปที่ 3.8 การ Configure Parameter ของเซอร์โวไดร์ฟ

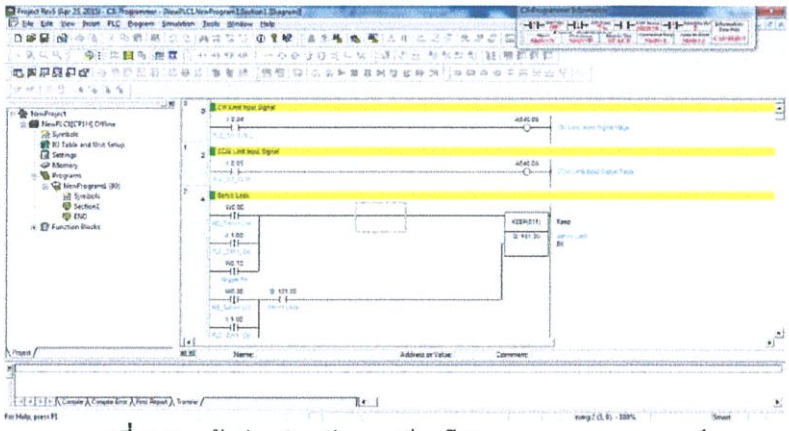
3.4 การเขียนโปรแกรมและกราฟิกการทำงานของระบบ

ในงานวิจัยนี้จะมีการเขียนโปรแกรมออกเป็น 4 อย่างคือการเขียนโปรแกรมควบคุมพีแอลซี , การเพิ่ม Channel, Device และ Point ของPLC ใน KEPserverEX5 และการตั้งค่าการเชื่อมต่อ , การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอทัชสกรีน และ การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอ HMI

3.4.1 การเขียนโปรแกรมสำหรับควบคุมพีแอลซี

ส่วนการเขียนโปรแกรมการทำงานให้กับเครื่องควบคุมพีแอลซี ซึ่งใช้ซอฟต์แวร์คือโปรแกรม Cx-Programmer ภาษาที่ใช้ในการเขียนโปรแกรมการทำงานเขียนโดยใช้ได้หลายภาษาเช่น ภาษาแลดเดอร์, ภาษาบูลลีน, ภาษาบล็อค, ภาษาข้อความภาษาอังกฤษ, ภาษาฟังก์ชั่นชาร์ท เป็นต้น โดยในงานวิจัยนี้เราใช้ภาษาแลดเดอร์ในการเขียนโปรแกรมต่างๆเพื่อใช้ทดลองกราฟิกที่ได้เขียนมารวมไปถึงค่าที่ได้ใน OPC Server ด้วย ซึ่งในที่นี้ได้ออกแบบโปรแกรมแลดเดอร์โดยใช้คำสั่งพื้นฐานทั่วๆ ไปของ PLC และมีการเพิ่มเติมในส่วน Function Boxes ที่ใช้ในการควบคุมตำแหน่งของเซอร์โวมอเตอร์ โดย Function Boxes ที่ใช้ในงานวิจัยนี้ได้แก่

1.) _NCCP1H010_MoveAbsolute_REAL: เป็นฟังก์ชั่นที่ใช้สำหรับสั่งการให้มอเตอร์เคลื่อนที่แบบ Absolute
2.) _NCCP1H020_MoveRelative_REAL: เป็นฟังก์ชั่นที่ใช้สำหรับสั่งการให้มอเตอร์เคลื่อนที่แบบ Relative
3.) _NCCP1H030_MoveVelocity_REAL: เป็นฟังก์ชั่นที่ใช้สำหรับสั่งการให้มอเตอร์หมุนด้วยความเร็วที่กำหนดโดยไม่สนใจถึงระยะทาง
4.) _NCCP1H050_Home_REAL: เป็นฟังก์ชั่นที่ใช้สำหรับสั่งการให้มอเตอร์กลับมาที่จุดกำเนิด
5.) _NCCP1H061_Stop_REAL: เป็นฟังก์ชั่นที่ใช้สำหรับสั่งการให้มอเตอร์หยุดหมุน
6.) _NCCP1H204_ReadActualPosition_REAL: เป็นฟังก์ชั่นที่ใช้สำหรับอ่านค่าตำแหน่งปัจจุบันของมอเตอร์

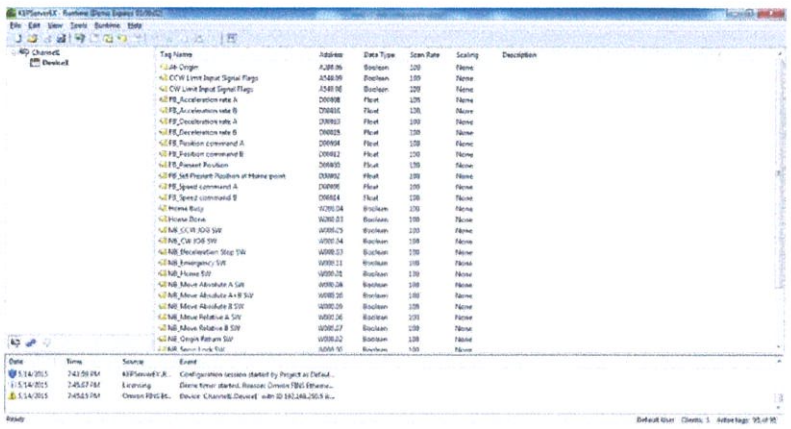


รูปที่ 3.9 หน้าต่างสำหรับการเขียนโปรแกรมภาษาแลดเดอร์

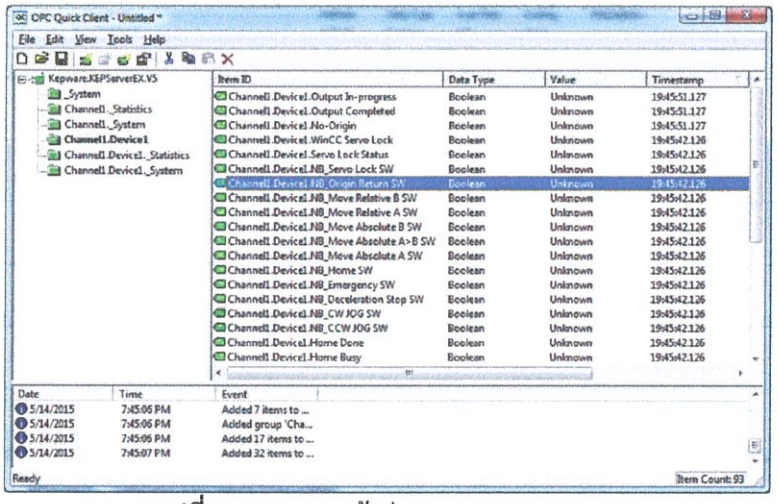
3.4.2 การเพิ่ม Channel, Device และ Point ของ PLC ใน KEPserverEX5 และการตั้งค่าการเชื่อมต่อ

การที่จะเชื่อมต่อ PLC กับหน้าจอ HMI ผ่าน OPC Server นั้น จำเป็นต้องมีการตั้งค่า Channel, Device และ Point ในโปรแกรม KEPserverEX5 ให้ตรงกันกับแอดเดรสที่ต้องการจะเข้าถึงของหน่วยความจำใน PLC ก่อน จึงจะสามารถติดต่อสื่อสารกันได้อย่างถูกต้องผ่าน OPC Server ด้วยระบบเครือข่าย Ethernet

เมื่อทำการเขียนโปรแกรมแลดเดอร์ที่ต้องการทดสอบพร้อมทั้งข้อมูลในแอดเดรสที่ได้ใช้ในการเขียนโปรแกรมแลดเดอร์แล้ว จากนั้นก็เข้าสู่ขั้นตอนของ OPC Server โดยเริ่มจากสร้างโปรเจคใหม่ และทำการเพิ่มชาแนลและตั้งค่าอุปกรณ์ต่างๆในชาแนลที่ได้สร้างเอาไว้และใช้ชื่อว่า Channel 1 ต่อจากนั้นก็ทำการเพิ่ม Tag สำหรับการร้องขอข้อมูล ซึ่งการตั้งค่าแต่ละ Tag มีวิธีการตั้งค่าที่แตกต่างกันตามประเภทของข้อมูล เมื่อทำการเพิ่ม Tag ทั้งหมดเรียบร้อยแล้ว โดยถ้าหากต้องการมอนิเตอร์ค่าต่างๆที่มีการเชื่อมข้อมูลถึงกันแล้ว สามารถดูได้จากการคลิกที่ Quick Client ทางด้านบนของหน้าต่างโปรแกรม (รูปที่ 3.3) และหน้าต่างของ Quick Client แสดงได้ดังรูปที่ 3.10



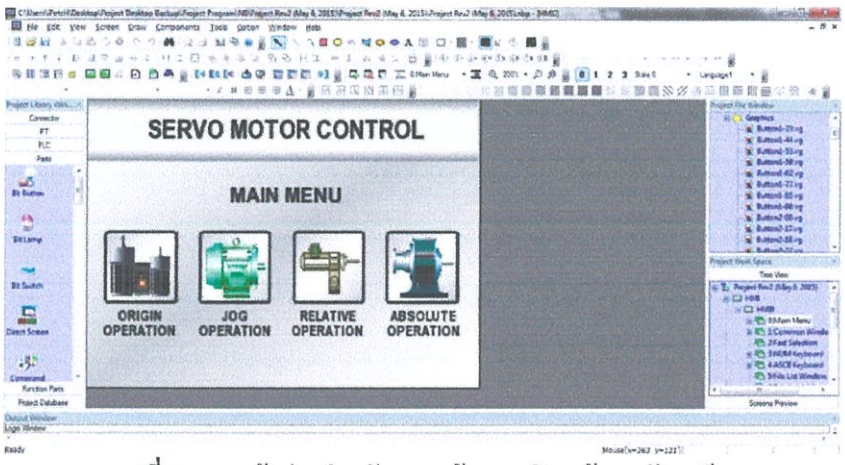
รูปที่ 3.10 ภาพแสดงตำแหน่งของ Quick Client



รูปที่ 3.11 แสดงหน้าต่างของ Quick Client

3.4.3 การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอทัชสกรีน

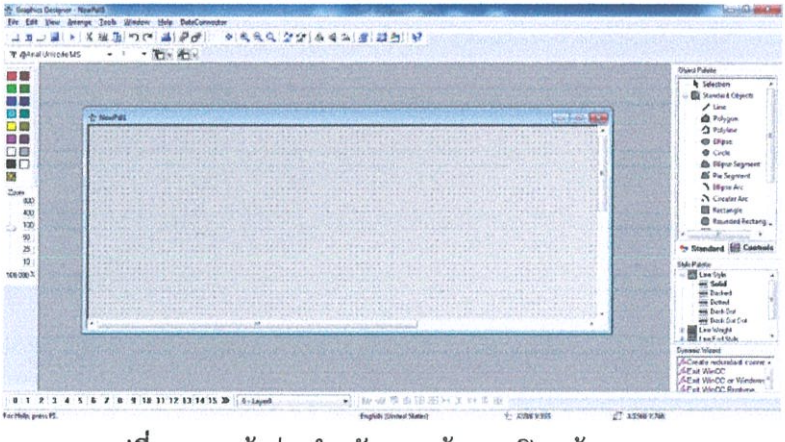
การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอทัชสกรีน โดยใช้โปรแกรม Nb - Designer ที่ใช้สำหรับการออกแบบหน้าจอทัชสกรีนที่เป็นตัวกลางระหว่างผู้ใช้กับเครื่องจักร โปรแกรมมีความสามารถในการสร้างรูปภาพหรือวัตถุที่มีความสามารถในการเข้าถึงข้อมูลจากภายนอก และ มีความสามารถในการเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ได้รับมาตามที่ผู้ใช้กำหนด



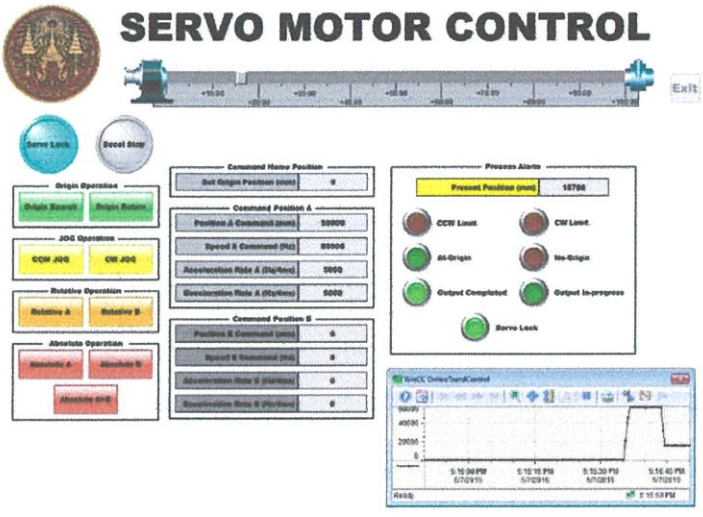
รูปที่ 3.12 หน้าต่างสำหรับการสร้างกราฟิกหน้าจอทัชสกรีน

3.2.4 การสร้างกราฟิกสำหรับการแสดงผลทางหน้าจอ HMI

การเขียนกราฟิกนั้นจะเริ่มจากการใช้งานหน้าต่าง Graphics Designer ซึ่งเป็นส่วนประกอบของโปรแกรม WinCC ที่ใช้สำหรับออกแบบหน้าจอที่เป็นตัวกลางระหว่างผู้ใช้กับเครื่องจักร โดย Graphics Designer มีความสามารถในการสร้างรูปภาพหรือวัตถุที่มีความสามารถในการเข้าถึงข้อมูลจากภายนอก มีความสามารถในการเขียนโปรแกรมเพื่อจัดการกับข้อมูลที่ได้รับมา



รูปที่ 3.13 หน้าต่างสำหรับการสร้างกราฟิกหน้าจอ HMI

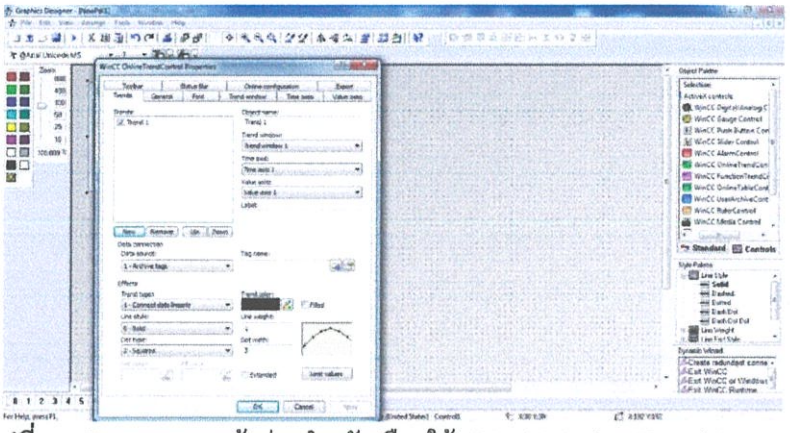


รูปที่ 3.14 กราฟิกหน้าจอ HMI ที่ใช้ในงานวิจัย

นอกจากนี้ Graphics Designer ยังได้เตรียมรูปภาพต่างๆไว้ โดยที่ผู้ใช้ไม่จำเป็นต้องสร้างมาขึ้นใหม่ สามารถนำมาใช้ในการเขียนกราฟิกได้เลย เริ่มจากการสร้างรูปภาพใหม่แล้วกำหนดคุณสมบัติของภาพที่สร้างขึ้นมา ซึ่งการกำหนดคุณสมบัติของภาพนั้น กำหนดได้หลากหลายมากมาย อาทิ เช่น กำหนดให้เป็นปุ่มกด แสดงผลเป็นสีต่างๆคล้ายไฟบอกสถานะ หรือเป็นถังน้ำที่สามารถแสดงสถานะของระดับน้ำได้เป็นต้น คุณสมบัติต่างๆที่เป็นการแสดงผลเหล่านี้สามารถกำหนดได้ในหน้าต่าง Tag Assignment โดยในปริญญานิพนธ์ฉบับนี้เราได้ทำการเขียนกราฟิกจำลองหน้าแผงควบคุมซึ่งแสดงได้ดังภาพ

เมื่อเขียนกราฟิกทั้งหมดเสร็จแล้ว หากต้องการให้กราฟิกที่เขียนนั้นแสดงผลได้นั้นต้องทำการเปลี่ยนโหมดเป็น Runtime Mode เพื่อแสดงผลกราฟิกแผงควบคุมที่ได้เขียนขึ้นมา ซึ่งแสดงได้ดังรูปที่ 3.14

ต่อมาคือขั้นตอนการสร้างเทรนกราฟ เริ่มจากในหน้าต่าง Graphics Design จะมีรายการให้เลือกชื่อว่า WinCC Online TrendControl (รูปที่ 3.15) ซึ่งมีไว้สำหรับแสดงค่าอินพุทหรือเอาท์พุทเป็นเทรนกราฟใน WinCC Runtime โดย แล้วทำการตั้งค่าต่างๆของเทรนกราฟเช่น ชื่อเทรนกราฟ ตั้งค่าเวลาแกน ย่านเวลา ค่าในแกนกราฟ Tags ที่ต้องการแสดงผล เป็นต้น



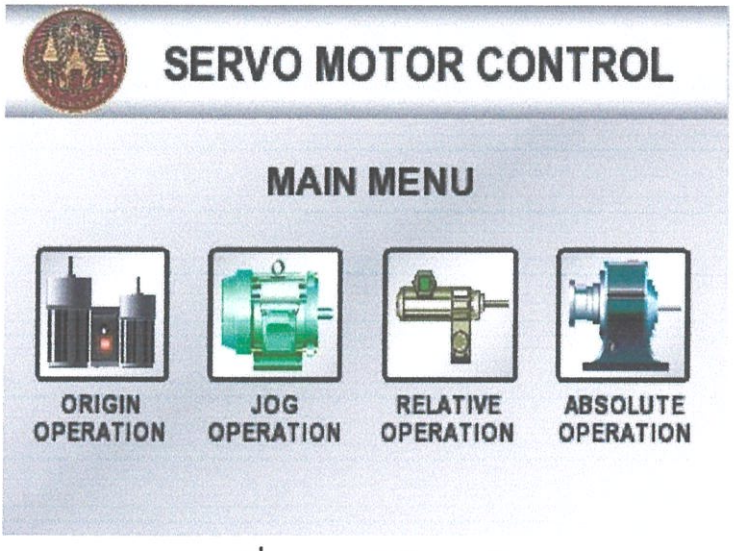
รูปที่ 3.15 ภาพแสดงหน้าต่างสำหรับเลือกใช้ WinCC OnlineTrendControl

บทที่ 4

ผลการทดลอง

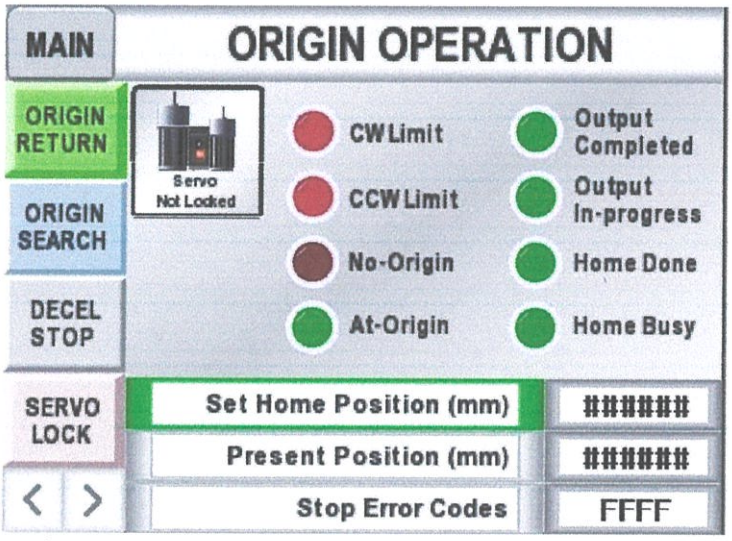
4.1 ผลการทดลองการใช้หน้าจอทัชสกรีนในการสั่งการเซอร์โวมอเตอร์และการแสดงผล

เมื่อทำการกดปุ่ม Power ของชุดควบคุมเซอร์โวมอเตอร์จะปรากฏหน้าเมนูหลัก (Main Menu) บนหน้าจอหน้าจอทัชสกรีนซึ่งจะเป็นหน้าต่างหลักให้ผู้ใช้เลือกรูปแบบการดำเนินการของเซอร์โวมอเตอร์ ซึ่งมีอยู่ด้วยกัน 4 รูปแบบใหญ่ๆ ดังนี้



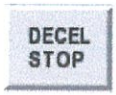
รูปที่ 4.1 แสดงหน้าเมนูหลัก

4.1.1 การดำเนินการทางจุดกำเนิด (Origin Operation) เมื่อสัมผัสที่ ที่หน้าเมนูหลัก จะปรากฏหน้าการดำเนินการทางจุดกำเนิดขึ้นมา



รูปที่ 4.2 แสดงหน้าการดำเนินการทางจุดกำเนิด (Origin Operation)

1. ใช้ในการกลับไปยังหน้าเมนูหลัก
2. และ ใช้ในการเปลี่ยนหน้าจอไปยังหน้าจอถัดไป และหน้าจอก่อนหน้า ตามลำดับ
3. ใช้ในการล็อคโรเตอร์ให้เซอร์โวมอเตอร์พร้อมสำหรับการหมุน
4. ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการค้นหาจุดกำเนิดและสั่งให้บอลสกรูเคลื่อนที่กลับไปยังจุดกำเนิด
5. ใช้ในการสั่งให้เซอร์โวมอเตอร์หมุนให้บอลสกรูเคลื่อนที่เข้าหาจุดกำเนิดในกรณีที่เซอร์โวมอเตอร์ทราบจุดกำเนิดแล้ว

6. 

ใช้ในการสั่งให้เซอร์โวมอเตอร์หยุดการหมุน
7. สามารถกำหนดตำแหน่งของจุดกำเนิดปัจจุบันให้เป็นตำแหน่งที่ต้องการได้โดยการสัมผัสที่ “Set Home Position (mm)” โดยจะปรากฏหน้าต่าง Pop-up ขึ้นมาหลังสัมผัส ใช้สำหรับกำหนดค่าที่ต้องการ ดังรูปที่ 4.3

รูปที่ 4.3 หน้าต่าง Pop-up สำหรับกำหนดค่าตัวเลขต่างๆ

8. สามารถทราบตำแหน่งปัจจุบันของตัวบอลสกรูได้จาก “Present Position (mm)”
9. สามารถทราบค่าความผิดพลาดของการหยุดหมุนของเซอร์โวมอเตอร์ได้จาก “Stop Error Codes”
10. และ ใช้ในการแสดงผลว่าบอลสกรูเคลื่อนที่มาสุดระยะแล้ว โดยจะแสดงสีแดงเมื่อบอลสกรูเคลื่อนที่มาสุดระยะทางทิศ CW และ CCW แล้ว
11. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุน โดยจะแสดงสีเขียวเมื่อเซอร์โวมอเตอร์กำลังหมุน
12. ใช้ในการแสดงผลว่าขณะนี้เซอรโวมอเตอร์ได้ทำการหมุนเสร็จแล้ว และได้ผลลัพธ์ถูกต้องแล้ว โดยจะแสดงสีเขียวเมื่อทำการหมุนเสร็จและได้ผลลัพธ์ตามต้องการ

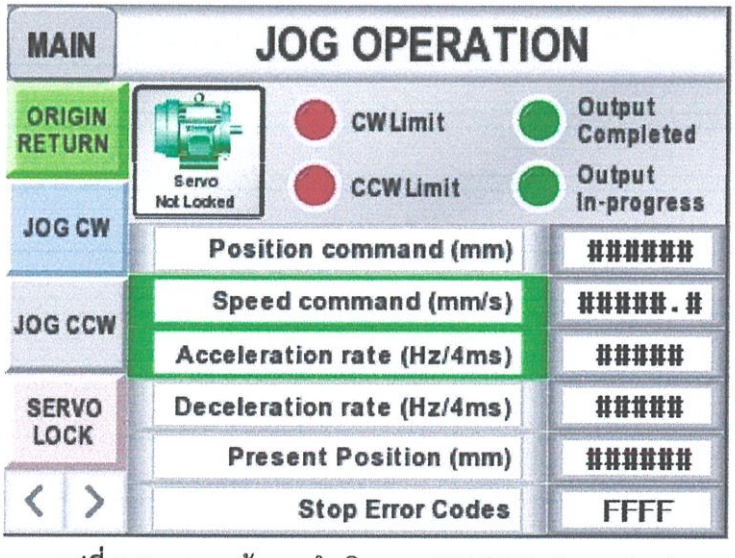
13. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์ไม่มีจุดกำเนิด โดยจะแสดงสีแดงเมื่อเซอร์โวมอเตอร์ไม่มีจุดกำเนิด

14. ใช้ในการแสดงผลว่าขณะนี้บอลสกรูอยู่ในตำแหน่งจุดกำเนิด โดยจะแสดงสีเขียวเมื่อบอลสกรูอยู่ที่ตำแหน่งจุดกำเนิด

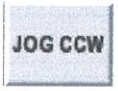
15. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุนโดยมีอัตราเร่งหรืออัตราหน่วง โดยจะแสดงสีเขียวเมื่อมีอัตราเร่งหรืออัตราหน่วง

16. ใช้ในการแสดงว่าขณะนี้เซอร์โวมอเตอร์ได้ทำการหมุนมาถึงตำแหน่งที่ได้กำหนดไว้เรียบร้อยแล้ว โดยจะแสดงเขียวเมื่อเมื่อเซอร์โวมอเตอร์อยู่ในตำแหน่งที่ได้กำหนดไว้

4.1.2 การดำเนินการ JOG (JOG Operation) เมื่อสัมผัสที่ ที่หน้าเมนูหลัก จะปรากฏหน้าการดำเนินการ JOG ขึ้นมา



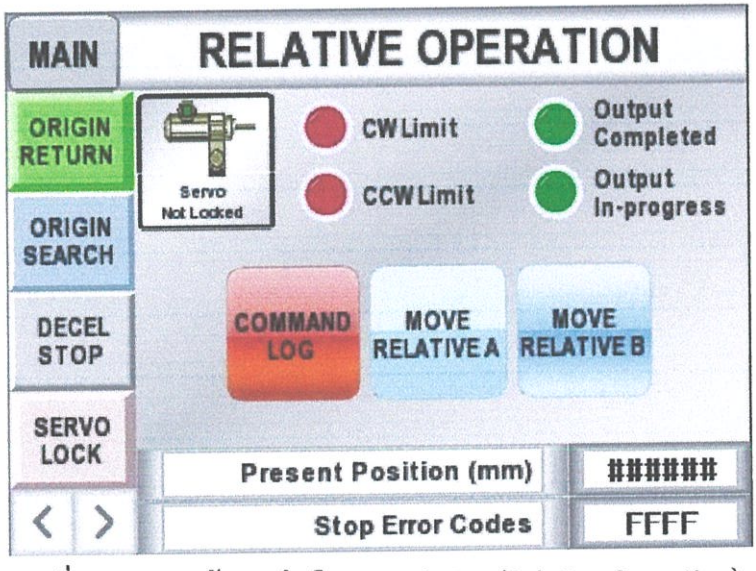
รูปที่ 4.4 แสดงหน้าการดำเนินการ JOG (JOG Operation)

1. ใช้ในการกลับไปยังหน้าเมนูหลัก
2. และ ใช้ในการเปลี่ยนหน้าจอไปยังหน้าจอถัดไป และหน้าจอก่อนหน้า ตามลำดับ
3. ใช้ในการล็อคโรเตอร์ให้เซอร์โวมอเตอร์พร้อมสำหรับการหมุน
4.  ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนทวนเข็มนาฬิกา และหยุดหมุนเมื่อไม่ได้สัมผัส
5. ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนตามเข็มนาฬิกา และหยุดหมุนเมื่อไม่ได้สัมผัส
6. ใช้ในการสั่งให้เซอร์โวมอเตอร์หมุนให้บอลสกรูเคลื่อนที่เข้าหาจุดกำเนิดในกรณีที่เซอร์โวมอเตอร์ทราบจุดกำเนิดแล้ว
7. สามารถกำหนดความเร็วและอัตราเร่งของการหมุนเซอร์โวมอเตอร์ได้โดยสัมผัสที่ “Speed Command (Hz)” และ “Acceleration rate (Hz/4ms)” ตามลำดับ จะปรากฏหน้าต่าง Pop-up ขึ้นมาหลังสัมผัส ใช้สำหรับกำหนดค่าที่ต้องการ ดังรูปที่ 4.3
8. สามารถทราบตำแหน่งปัจจุบันของตัวบอลสกรูได้จาก “Present Position (mm)”
9. สามารถทราบค่าความผิดพลาดของการหยุดหมุนของเซอร์โวมอเตอร์ได้จาก “Stop Error Codes”
10. และ ใช้ในการแสดงผลว่าบอลสกรูเคลื่อนที่มาสุดระยะแล้ว โดยจะแสดงสีแดงเมื่อบอลสกรูเคลื่อนที่มาสุดระยะทางทิศ CW และ CCW แล้ว

11. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุน โดยจะแสดงสีเขียวเมื่อเซอร์โวมอเตอร์กำลังหมุน

12. ใช้ในการแสดงผลว่าขณะนี้เซอรโวมอเตอร์ได้ทำการหมุนเสร็จแล้ว และได้ผลลัพธ์ถูกต้องแล้ว โดยจะแสดงสีเขียวเมื่อทำการหมุนเสร็จและได้ผลลัพธ์ตามต้องการ

4.1.3 การดำเนินการ Relative (Relative Operation) เมื่อสัมผัสที่ ที่หน้าเมนูหลัก จะปรากฏหน้าการดำเนินการ Relative ขึ้นมา



รูปที่ 4.5 แสดงหน้าการดำเนินการ Relative (Relative Operation)

1. ใช้ในการกลับไปยังหน้าเมนูหลัก

2. และ ใช้ในการเปลี่ยนหน้าจอไปยังหน้าจอถัดไป และหน้าจอก่อนหน้า ตามลำดับ

3.

ใช้ในการล็อคโรเตอร์ให้เซอร์โวมอเตอร์พร้อมสำหรับการหมุน

4.

ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Relative ไปยังตำแหน่ง A

5.

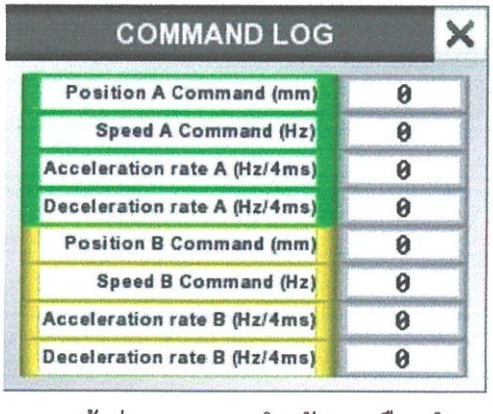
ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Relative ไปยังตำแหน่ง B

6.

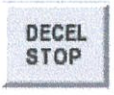
ใช้ในการสั่งให้เซอร์โวมอเตอร์หมุนให้บอลสกรูเคลื่อนที่เข้าหาจุดกำเนิดในกรณีที่เซอร์โวมอเตอร์ทราบจุดกำเนิดแล้ว

7.

ใช้ในการเปิดหน้าต่าง Pop-up ขึ้นมาเพื่อกำหนดค่าตำแหน่งของการเคลื่อนที่, ความเร็วของการเคลื่อนที่, อัตราเร่ง และอัตราหน่วง ของตำแหน่ง A และ B ดังรูปที่ 4.6 สามารถกำหนดค่าตัวแปรต่างๆ ได้เมื่อสัมผัสที่ชื่อตัวแปรนั้นๆ โดยจะปรากฏหน้าต่าง Pop-up ดังรูปที่ 4.3 ขึ้นมาเพื่อกำหนดค่าที่ต้องการ



รูปที่ 4.6 แสดงหน้าต่าง Pop-up สำหรับการเลือกกำหนดค่าต่างๆ

8. 

ใช้ในการสั่งให้เซอร์โวมอเตอร์หยุดการหมุน

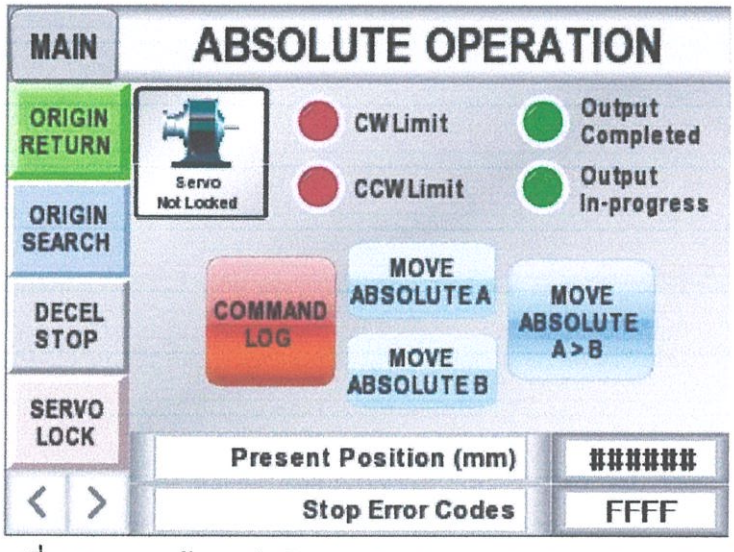
9. สามารถทราบตำแหน่งปัจจุบันของตัวบอลสกรูได้จาก “Present Position (mm)”

10. และ ใช้ในการแสดงผลว่าบอลสกรูเคลื่อนที่มาสุดระยะแล้ว โดยจะแสดงสีแดงเมื่อบอลสกรูเคลื่อนที่มาสุดระยะทางทิศ CW และ CCW แล้ว

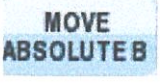
11. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุน โดยจะแสดงสีเขียวเมื่อเซอร์โวมอเตอร์กำลังหมุน

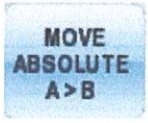
12. ใช้ในการแสดงผลว่าขณะนี้เซอรโวมอเตอร์ได้ทำการหมุนเสร็จแล้ว และได้ผลลัพธ์ถูกต้องแล้ว โดยจะแสดงสีเขียวเมื่อทำการหมุนเสร็จและได้ผลลัพธ์ตามต้องการ

4.1.4 การดำเนินการ Absolute (Absolute Operation) เมื่อสัมผัสที่ ที่หน้าเมนูหลัก จะปรากฏหน้าการดำเนินการ Absolute ขึ้นมา



รูปที่ 4.7 แสดงหน้าการดำเนินการ Absolute (Absolute Operation)

1. ใช้ในการกลับไปยังหน้าเมนูหลัก
2. และ ใช้ในการเปลี่ยนหน้าจอไปยังหน้าจอถัดไป และหน้าจอก่อนหน้า ตามลำดับ
3. ใช้ในการล็อคโรเตอร์ให้เซอร์โวมอเตอร์พร้อมสำหรับการหมุน
4. ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Absolute ไปยังตำแหน่ง A
5.  ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Absolute ไปยังตำแหน่ง B

6. 

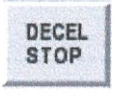
ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Absolute ไปยังตำแหน่ง A และกลับมายังจุดกำเนิด หลังจากนั้นทำการหมุนแบบ Absolute ไปยังตำแหน่ง B และกลับมายังจุดกำเนิด ตามลำดับ

7.

ใช้ในการสั่งให้เซอร์โวมอเตอร์หมุนให้บอลสกรูเคลื่อนที่เข้าหาจุดกำเนิดในกรณีที่เซอร์โวมอเตอร์ทราบจุดกำเนิดแล้ว

8.

ใช้ในการเปิดหน้าต่าง Pop-up ขึ้นมาเพื่อกำหนดค่าตำแหน่งของการเคลื่อนที่, ความเร็วของการเคลื่อนที่, อัตราเร่ง และอัตราหน่วง ของตำแหน่ง A และ B ดังรูปที่ 4.6 สามารถกำหนดค่าตัวแปรต่างๆ ได้เมื่อสัมผัสที่ชื่อตัวแปรนั้นๆ โดยจะปรากฏหน้าต่าง Pop-up ดังรูปที่ 4.3 ขึ้นมาเพื่อกำหนดค่าที่ต้องการ

9. 

ใช้ในการสั่งให้เซอร์โวมอเตอร์หยุดการหมุน

10. สามารถทราบตำแหน่งปัจจุบันของตัวบอลสกรูได้จาก “Present Position (mm)”

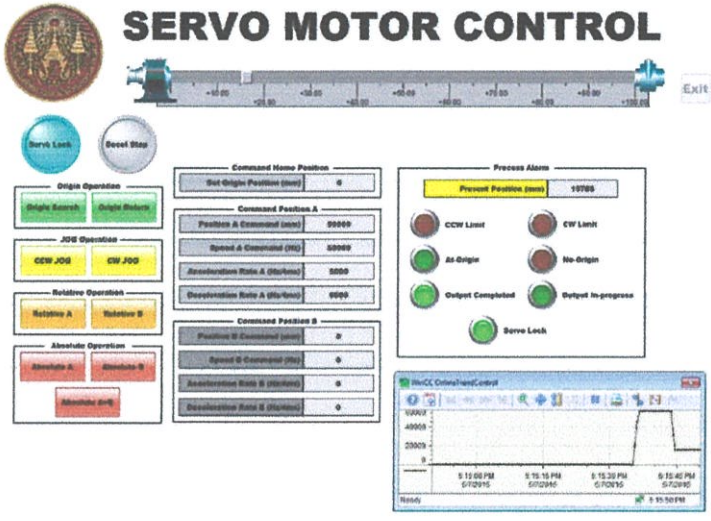
11. และ ใช้ในการแสดงผลว่าบอลสกรูเคลื่อนที่มาสุดระยะแล้ว โดยจะแสดงสีแดงเมื่อบอลสกรูเคลื่อนที่มาสุดระยะทางทิศ CW และ CCW แล้ว

12. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุน โดยจะแสดงสีเขียวเมื่อเซอร์โวมอเตอร์กำลังหมุน

13. ใช้ในการแสดงผลว่าขณะนี้เซอรโวมอเตอร์ได้ทำการหมุนเสร็จแล้ว และได้ผลลัพธ์ถูกต้องแล้ว โดยจะแสดงสีเขียวเมื่อทำการหมุนเสร็จและได้ผลลัพธ์ตามต้องการ

4.2 ผลการทดลองการใช้โปรแกรม SIMATIC WinCC Explorer สั่งการเซอร์โวมอเตอร์ และการแสดงผล

เมื่อทำการ RUN โปรแกรม Graphic Designer สำหรับการควบคุมการทำงานของเซอร์โวมอเตอร์ ใน SIMATIC WinCC Explorer จะปรากฏหน้าต่างสำหรับสั่งการและแสดงผลขึ้นมาดังรูปที่ 4.7 โดยหน้าต่างสำหรับสั่งการเซอร์โวมอเตอร์และแสดงผลนั้นแบ่งออกเป็น 2 ส่วนใหญ่ๆ ดังนี้



รูปที่ 4.7 แสดงหน้าต่างสำหรับสั่งการเซอร์โวมอเตอร์ และแสดงผลด้วยโปรแกรม Graphic Designer ใน SIMATIC WinCC Explorer

4.2.1 ส่วนสั่งการของการควบคุมเซอร์โวมอเตอร์ คือส่วนที่ทำหน้าที่สั่งการ และกำหนดค่าตัวแปรต่างๆ ที่ใช้หมุนของเซอร์โวมอเตอร์

1.

ใช้ในการล็อคโรเตอร์ให้เซอร์โวมอเตอร์พร้อมสำหรับการหมุน

2.

ใช้ในการสั่งให้เซอร์โวมอเตอร์หยุดการหมุน

3.

ใช้ในการสั่งให้เซอร์โวมอเตอร์หยุดการหมุนในกรณีฉุกเฉิน

4.

ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการค้นหาจุดกำเนิดและสั่งให้บอลสกรูเคลื่อนที่กลับไปยังจุดกำเนิด

5.

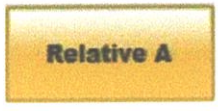
ใช้ในการสั่งให้เซอร์โวมอเตอร์หมุนให้บอลสกรูเคลื่อนที่เข้าหาจุดกำเนิดในกรณีที่เซอร์โวมอเตอร์ทราบจุดกำเนิดแล้ว

6.

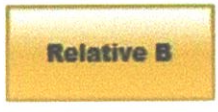
ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนทวนเข็มนาฬิกาและหยุดหมุนเมื่อไม่ได้สัมผัส

7.

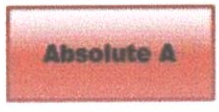
ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนตามเข็มนาฬิกาและหยุดหมุนเมื่อไม่ได้สัมผัส

8. 

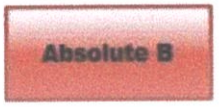
ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Relative ไปยังตำแหน่ง A

9. 

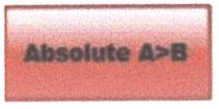
ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Relative ไปยังตำแหน่ง B

10. 

ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Absolute ไปยังตำแหน่ง A

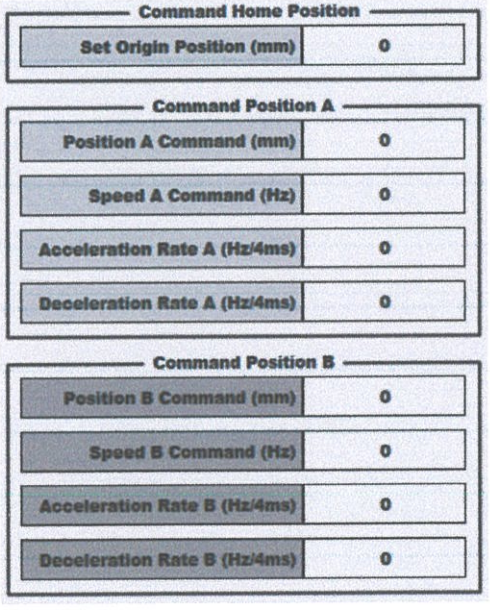
14. 

ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Absolute ไปยังตำแหน่ง B

15. 

ใช้ในการสั่งให้เซอร์โวมอเตอร์ทำการหมุนแบบ Absolute ไปยังตำแหน่ง A และกลับมายังจุดกำเนิด หลังจากนั้นทำการหมุนแบบ Absolute ไปยังตำแหน่ง B และกลับมายังจุดกำเนิด ตามลำดับ

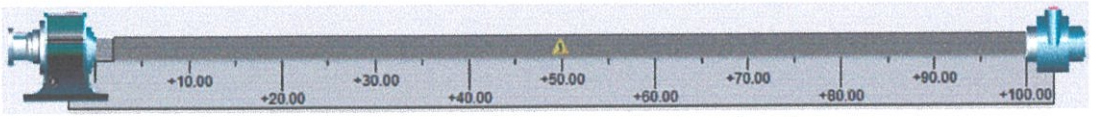
16. สามารถกำหนดค่าตัวแปรต่างๆ ที่ใช้ในการหมุนของเซอร์โวมอเตอร์ โดยการพิมพ์ตัวเลขที่ต้องการใส่ลงไปในช่องว่างหลังตัวแปรที่ต้องการกำหนด ดังรูป 4.8



รูปที่ 4.8 แสดงชนิดของตัวแปรต่างๆ ใช้ในการหมุนของเซอร์โวมอเตอร์และช่องว่างสำหรับกำหนดค่าตัวแปร

4.2.2 ส่วนแสดงผลของการควบคุมเซอร์โวมอเตอร์ คือส่วนที่แสดงผลของการควบคุมต่างๆ ให้ผู้ใช้ได้รับรู้

1. สามารถทราบตำแหน่งปัจจุบันของตัวบอลสกรูได้จาก “Present Position” และภาพแอนิเมชัน ดังรูปที่ 4.8



รูปที่ 4.7 แสดงภาพแอนิเมชันสำหรับแสดงระยะการเคลื่อนที่ของบอลสกรู

2. และ ใช้ในการแสดงผลว่าบอลสกรูเคลื่อนที่มาสุดระยะแล้ว โดยจะแสดงสีแดงเมื่อบอลสกรูเคลื่อนที่มาสุดระยะทางทิศ CW และ CCW แล้ว

3. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุน โดยจะแสดงสีเขียวเมื่อเซอร์โวมอเตอร์กำลังหมุน

4. ใช้ในการแสดงผลว่าขณะนี้เซอรโวมอเตอร์ได้ทำการหมุนเสร็จแล้ว และได้ผลลัพธ์ถูกต้องแล้ว โดยจะแสดงสีเขียวเมื่อทำการหมุนเสร็จและได้ผลลัพธ์ตามต้องการ

5. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์ไม่มีจุดกำเนิด โดยจะแสดงสีแดงเมื่อเซอร์โวมอเตอร์ไม่มีจุดกำเนิด

6. ใช้ในการแสดงผลว่าขณะนี้บอลสกรูอยู่ในตำแหน่งจุดกำเนิด โดยจะแสดงสีเขียวเมื่อบอลสกรูอยู่ที่ตำแหน่งจุดกำเนิด

7. ใช้ในการแสดงผลว่าขณะนี้เซอร์โวมอเตอร์กำลังหมุนโดยมีอัตราเร่งหรืออัตราหน่วง โดยจะแสดงสีเขียวเมื่อมีอัตราเร่งหรืออัตราหน่วง

8. ใช้ในการแสดงว่าขณะนี้เซอร์โวมอเตอร์ได้ทำการหมุนมาถึงตำแหน่งที่ได้กำหนดไว้เรียบร้อยแล้ว โดยจะแสดงเขียวเมื่อเมื่อเซอร์โวมอเตอร์อยู่ในตำแหน่งที่ได้กำหนดไว้

4.3 การทดสอบการทำงานของหน้าจอ HMI

เพื่อศึกษาว่าการสั่งการที่ได้จากส่งข้อมูลจากคอมพิวเตอร์ผ่าน OPC Server ไปหาตัวควบคุมได้ผลการทำงานเป็นไปตามเงื่อนไขที่ต้องการจริง และมีการแสดงผลตรงตามเงื่อนไขการทำงานที่สั่งการได้อย่างถูกต้อง จึงได้ทำการทดสอบการทำงานของระบบด้วยการกำหนดค่าตัวแปรให้กับระบบและสั่งการผ่านหน้าจอ HMI หลังจากนั้นตรวจสอบสถานะของการทำงานของระบบว่าเป็นไปตามที่สั่งการหรือไม่

4.3.1 การทดสอบการทำงานของระบบผ่านหน้าจอ HMI ด้วยการกำหนดตัวแปรแบบที่ 1

เมื่อทำการกำหนดค่าตัวแปรในการเคลื่อนที่ตามตารางที่ 4.1 หลังจากนั้นกดปุ่มให้เซอร์โวมิเตอร์เคลื่อนที่ในแบบต่างๆ บนหน้าจอ HMI พบว่าสามารถสั่งการการทำงานของระบบผ่านหน้าจอ HMI ได้ และสามารถทำการแสดงผลค่าต่างๆ รวมทั้งสถานการทำงานของระบบผ่านหน้าจอ HMI ได้ โดยมีผลการทดลองตามตารางที่ 4.1

4.3.2 การทดสอบการทำงานของระบบผ่านหน้าจอ HMI ด้วยการกำหนดตัวแปรแบบที่ 2

เมื่อทำการกำหนดค่าตัวแปรในการเคลื่อนที่ตามตารางที่ 4.2 หลังจากนั้นกดปุ่มให้เซอร์โวมิเตอร์เคลื่อนที่ในแบบต่างๆ บนหน้าจอ HMI พบว่าสามารถสั่งการการทำงานของระบบผ่านหน้าจอ HMI ได้ และสามารถทำการแสดงผลค่าต่างๆ รวมทั้งสถานการทำงานของระบบผ่านหน้าจอ HMI ได้ โดยมีผลการทดลองตามตารางที่ 4.2

4.3.3 การทดสอบการทำงานของปุ่มหยุดฉุกเฉิน (Emergency)

เมื่อทำการกดปุ่มหยุดฉุกเฉินบนหน้าจอ HMI ขณะที่มอเตอร์มีการหมุนอยู่สามารถทำให้มอเตอร์หยุดหมุนอย่างฉุกเฉินได้จริง

4.3.4 การทดสอบการทำงานของปุ่มหยุด (Deceleration Stop)

เมื่อทำการกดปุ่มหยุดบนหน้าจอ HMI ขณะที่มอเตอร์มีการหมุนอยู่สามารถทำให้มอเตอร์หยุดหมุนได้จริง

4.3.5 การทดสอบการทำงานของปุ่มล็อคโรเตอร์ (Servo Lock)

เมื่อทำการกดปุ่มล็อคโรเตอร์บนหน้าจอ HMI เพื่อสั่งให้โรเตอร์ล็อคหรือปลดล็อคนั้นสามารถสั่งให้เซอร์โวไดรเวอร์ทำการล็อคโรเตอร์หรือปลดล็อคได้จริง

ตารางที่ 4.1 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 1 ผ่านหน้าจอ HMI

Present Position: 0 mm Position A Command: 40 mm Speed A Command: 100 mm/s Acceleration Rate A: 60000 Deceleration Rate A: 60000 Position B Command: 70 mm Speed B Command: 50 mm/s Acceleration Rate B: 5000 Deceleration Rate B: 5000		ผล	Output In progress	Output Complete	At Origin	No Origin	Present Position	Ball Screw Animation	CCW Limit	CW Limit
เหตุ										
Origin Search	ขณะเคลื่อนที่		A			A	S	M		
	ถึงตำแหน่งเป้าหมาย			A	A		S	M		
Relative A	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Relative B	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Absolute A	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Absolute B	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Absolute A>B	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
CCW JOG	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
	ขณะเคลื่อนที่เกินกำหนด						S	M	A	
CW JOG	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
	ขณะเคลื่อนที่เกินกำหนด						S	M		A
Origin Return	ถึงตำแหน่งเป้าหมาย			A	A		S	M		

หมายเหตุ A = แสดงสัญญาณเตือน, S = แสดงค่าได้ถูกต้อง, M = แอนิเมชันขยับได้ถูกต้อง

ตารางที่ 4.2 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 2 ผ่านหน้าจอ HMI

Present Position: 50 mm Position A Command: 40 mm Speed A Command: 100 mm/s Acceleration Rate A: 5000 Deceleration Rate A: 5000 Position B Command: 70 mm Speed B Command: 50 mm/s Acceleration Rate B: 5000 Deceleration Rate B: 5000		ผล	Output In progress	Output Complete	At Origin	No Origin	Present Position	Ball Screw Animation	CCW Limit	CW Limit
เหตุ										
Origin Search	ขณะเคลื่อนที่		A			A	S	M		
	ถึงตำแหน่งเป้าหมาย			A	A		S	M		
Relative A	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Relative B	ขณะเคลื่อนที่		A				S	M		
	ขณะเคลื่อนที่เกินกำหนด						S	M		A
Absolute A	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Absolute B	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
Absolute A>B	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
CCW JOG	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
	ขณะเคลื่อนที่เกินกำหนด						S	M	A	
CW JOG	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A			S	M		
	ขณะเคลื่อนที่เกินกำหนด						S	M		A
Origin Return	ขณะเคลื่อนที่		A				S	M		
	ถึงตำแหน่งเป้าหมาย			A	A		S	M		

หมายเหตุ A = แสดงสัญญาณเตือน, S = แสดงค่าได้ถูกต้อง, M = แอนิเมชันขยับได้ถูกต้อง

4.4 การทดสอบการทำงานของหน้าจอทัชสกรีน

เพื่อศึกษาว่าการสั่งการที่ได้จากส่งข้อมูลจากหน้าจอทัชสกรีน ไปหาตัวควบคุม ได้ผลการทำงานเป็นไปตามเงื่อนไขที่ต้องการจริง และมีการแสดงผลตรงตามเงื่อนไขการทำงานที่สั่งการได้อย่างถูกต้อง จึงได้ทำการทดสอบการทำงานของระบบด้วยการกำหนดค่าตัวแปรให้กับระบบและสั่งการผ่านหน้าจอทัชสกรีน หลังจากนั้นตรวจสอบสถานะของการทำงานของระบบว่าเป็นไปตามที่สั่งการหรือไม่

4.4.1 การทดสอบการทำงานของระบบผ่านหน้าจอทัชสกรีนด้วยการกำหนดตัวแปรแบบที่ 1

เมื่อทำการกำหนดค่าตัวแปรในการเคลื่อนที่ตามตารางที่ 4.3 หลังจากนั้นกดปุ่มให้เซอร์โวมิเตอร์เคลื่อนที่ในแบบต่างๆ บนหน้าจอทัชสกรีน พบว่าสามารถสั่งการการทำงานของระบบผ่านหน้าจอทัชสกรีนได้ และสามารถทำการแสดงผลค่าต่างๆ รวมทั้งสถานการทำงานของระบบผ่านหน้าจอทัชสกรีนได้ โดยมีผลการทดลองตามตารางที่ 4.3

4.4.2 การทดสอบการทำงานของระบบผ่านหน้าจอทัชสกรีนด้วยการกำหนดตัวแปรแบบที่ 2

เมื่อทำการกำหนดค่าตัวแปรในการเคลื่อนที่ตามตารางที่ 4.4 หลังจากนั้นกดปุ่มให้เซอร์โวมิเตอร์เคลื่อนที่ในแบบต่างๆ บนหน้าจอทัชสกรีน พบว่าสามารถสั่งการการทำงานของระบบผ่านหน้าจอทัชสกรีนได้ และสามารถทำการแสดงผลค่าต่างๆ รวมทั้งสถานการทำงานของระบบผ่านหน้าจอทัชสกรีนได้ โดยมีผลการทดลองตามตารางที่ 4.4

4.4.3 การทดสอบการทำงานของปุ่มหยุดฉุกเฉิน (Emergency)

เมื่อทำการกดปุ่มหยุดฉุกเฉินบนหน้าจอทัชสกรีนในขณะที่มอเตอร์มีการหมุนอยู่สามารถทำให้มอเตอร์หยุดหมุนอย่างฉุกเฉินได้จริง

4.4.3 การทดสอบการทำงานของปุ่มหยุด (Deceleration Stop)

เมื่อทำการกดปุ่มหยุดบนหน้าจอทัชสกรีนในขณะที่มอเตอร์มีการหมุนอยู่สามารถทำให้มอเตอร์หยุดหมุนได้จริง

4.4.3 การทดสอบการทำงานของปุ่มล็อคโรเตอร์ (Servo Lock)

เมื่อทำการกดปุ่มล็อคโรเตอร์บนหน้าจอทัชสกรีนเพื่อสั่งให้โรเตอร์ล็อคหรือปลดล็อคนั้นสามารถสั่งให้เซอร์โวไดรเวอร์ทำการล็อคโรเตอร์หรือปลดล็อคได้จริง

ตารางที่ 4.3 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 1 ผ่านหน้าจอ HMI

Present Position: 0 mm Position A Command: 40 mm Speed A Command: 100 mm/s Acceleration Rate A: 60000 Deceleration Rate A: 60000 Position B Command: 70 mm Speed B Command: 50 mm/s Acceleration Rate B: 5000 Deceleration Rate B: 5000		ผล	Output In progress	Output Complete	At Origin	No Origin	CCW Limit	CW Limit
เหตุ								
Origin Search	ขณะเคลื่อนที่		A			A		
	ถึงตำแหน่งเป้าหมาย			A	A			
Relative A	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Relative B	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Absolute A	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Absolute B	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Absolute A>B	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
CCW JOG	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
	ขณะเคลื่อนที่เกินกำหนด						A	
CW JOG	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
	ขณะเคลื่อนที่เกินกำหนด							A
Origin Return	ถึงตำแหน่งเป้าหมาย			A	A			

หมายเหตุ A = แสดงสัญญาณเตือน

ตารางที่ 4.4 การทดสอบการทำงานของเซอร์โวมอเตอร์ด้วยการกำหนดตัวแปรแบบที่ 2 ผ่านหน้าจอ HMI

Present Position: 50 mm Position A Command: 40 mm Speed A Command: 100 mm/s Acceleration Rate A: 5000 Deceleration Rate A: 5000 Position B Command: 70 mm Speed B Command: 50 mm/s Acceleration Rate B: 5000 Deceleration Rate B: 5000		ผล	Output In progress	Output Complete	At Origin	No Origin	CCW Limit	CW Limit
เหตุ								
Origin Search	ขณะเคลื่อนที่		A			A		
	ถึงตำแหน่งเป้าหมาย			A	A			
Relative A	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Relative B	ขณะเคลื่อนที่		A					
	ขณะเคลื่อนที่เกินกำหนด							A
Absolute A	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Absolute B	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
Absolute A>B	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
CCW JOG	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
	ขณะเคลื่อนที่เกินกำหนด						A	
CW JOG	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A				
	ขณะเคลื่อนที่เกินกำหนด							A
Origin Return	ขณะเคลื่อนที่		A					
	ถึงตำแหน่งเป้าหมาย			A	A			

หมายเหตุ A = แสดงสัญญาณเตือน

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

ในปริญญานิพนธ์ฉบับนี้ ได้นำเสนอการควบคุมการทำงานของเซอร์โวมอเตอร์ด้วยเครื่องควบคุมพีแอลซีร่วมกับ HMI ผ่านระบบ Ethernet โดยการใช้โปรแกรม CX-Programmer ในการเขียนชุดคำสั่งให้กับเครื่องควบคุมพีแอลซีเพื่อใช้ควบคุมการทำงานของเซอร์โวมอเตอร์ ทางด้านการสั่งการให้เครื่องควบคุมพีแอลซีสามารถควบคุมการทำงานของเซอร์โวมอเตอร์และด้านการแสดงผลของของระบบนั้นได้ใช้หน้าจอทัชสกรีน และหน้าจอมอนิเตอร์ของคอมพิวเตอร์เข้ามาเป็น HMI โดยสำหรับหน้าจอทัชสกรีนนั้นจะสามารถเข้าถึงหน่วยความจำในเครื่องควบคุมพีแอลซีโดยมีการรับส่งข้อมูลผ่านสาย RS-232 ใช้โปรแกรม NB-Designer ในการออกแบบและวาดกราฟิกให้กับหน้าจอทัชสกรีน และสำหรับหน้าจอมอนิเตอร์ของคอมพิวเตอร์นั้นใช้โปรแกรม SIMATIC WinCC ในการออกแบบและวาดกราฟิกสำหรับทำหน้าที่เป็น HMI ร่วมกับ KEPwareEX5 ซึ่งเป็น OPC server ทำหน้าที่เป็น Server สำหรับการติดต่อสื่อสาร ทำให้คอมพิวเตอร์สามารถติดต่อสื่อสารกันกับเครื่องควบคุมพีแอลซีผ่านสาย Ethernet ได้ ปริญญานิพนธ์ฉบับนี้ถือเป็นการจำลองการทำงานของระบบอุตสาหกรรมที่มีความหลากหลายของอุปกรณ์ สามารถทำให้อุปกรณ์ที่มีผู้ผลิตต่างกันสามารถติดต่อสื่อสารและทำงานร่วมกันได้โดยใช้ OPC Server รวมทั้งมีการสั่งการระบบ, แสดงผลแบบเรียลไทม์ตามการทำงานของระบบที่เป็นอยู่ ณ ปัจจุบันนั้น และเก็บข้อมูลเป็นกราฟเทรนด์เพื่อดูแนวโน้มของการเปลี่ยนแปลงของระบบได้อีกด้วย โดยในภาคเรียนที่ 1 นั้นได้ทำการศึกษาในส่วนของการควบคุมการทำงานของเซอร์โวมอเตอร์ด้วยเครื่องควบคุมพีแอลซี ทั้งในด้านอุปกรณ์ต่างๆ ที่เป็นส่วนประกอบของการควบคุม และในด้านการเขียนชุดคำสั่งให้กับเครื่องควบคุมพีแอลซีในการควบคุมเซอร์โวมอเตอร์ ส่วนในภาคเรียนที่ 2 นั้นได้ทำการออกแบบกราฟิกของหน้าจอทัชสกรีนและออกแบบกราฟิกของหน้าจอ HMI ในการใช้สั่งการและแสดงผลของระบบ รวมทั้งศึกษาเกี่ยวกับ OPC Server เพื่อให้สามารถใช้หน้าจอ HMI ในการสื่อสารข้อมูลกับเครื่องควบคุมพีแอลซีได้

จากผลการทดลอง สามารถควบคุมการทำงานของเซอร์โวมอเตอร์ด้วยเครื่องควบคุมพีแอลซีร่วมกับหน้าจอทัชสกรีน และหน้าจอ HMI ผ่านระบบ Ethernetได้จริง ทั้งแสดงผลและสั่งการได้ ในการ

แสดงผลนั้นค่อนข้างที่จะถูกต้อง สามารถนำโปรแกรมและระบบนี้ไปพัฒนาต่อ และประยุกต์ในการควบคุมอุปกรณ์ในลักษณะอื่นได้หลากหลาย

5.2 ข้อเสนอแนะ

หน้าจอทัชสกรีนรุ่นนี้ที่ใช้ในระบบไม่เหมาะสมที่จะใช้เป็นอุปกรณ์อินพุตของเครื่องควบคุมพีแอลซี เนื่องจากมีช่วงเวลาดีเลย์หลังการสัมผัสค่อนข้างนาน ส่งผลให้สั่งงานได้ช้ากว่าการสัมผัสจริงและอาจจะเป็นผลเสียกับระบบที่ใหญ่ขึ้นได้ หากมีการพัฒนาต่อในอนาคตควรเลือกใช้อุปกรณ์อินพุตชนิดอื่น

5.3 อุปสรรคและปัญหา

ในการเพิ่มแชนแนล เพิ่มอุปกรณ์ เพิ่มกำหนดพ้อยท์ ใน OPC Server นั้นมีความซับซ้อนและมีพ้อยท์เป็นจำนวนมากซึ่งเป็นอุปสรรคในการทำงาน ต้องอาศัยความชำนาญและความรอบคอบในการทำและการตรวจสอบ ระบบจึงจะสามารถทำการสื่อสารระหว่างอุปกรณ์ได้อย่างถูกต้อง เช่นเดียวกันกับการกำหนดแอดเดรสให้กับกราฟิกในหน้าจอทัชสกรีน หรือหน้าจอ HMI ที่มีแอดเดรสจำนวนมาก การกำหนดแอดเดรสผิดพลาดพลาดเพราะความสับสนหรือสะเพร่าอาจเป็นปัญหาหลักที่ทำให้ระบบไม่เป็นไปตามต้องการ ทำให้เสียเวลาไปโดยเปล่าประโยชน์ วิธีการแก้ไข คือต้องมีสติ มีการวางแผนการทำงานให้เป็นระบบและรอบคอบในการทำงาน

ภาคผนวก

บรรณานุกรม

[1] ณรงค์ ตันชีวะวงศ์. ระบบ PLC. สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น)

[2] สุธี พงศาสกุลรับ. ณรงค์ ล่ำดี. การสื่อสารข้อมูลและเครือข่ายคอมพิวเตอร์. สำนักพิมพ์เคทีพี

[3] ผศ.ศุภชัย สุรินทร์วงศ์. มอเตอร์ไฟฟ้า. สมาคมส่งเสริมเทคโนโลยี(ไทย-ญี่ปุ่น)

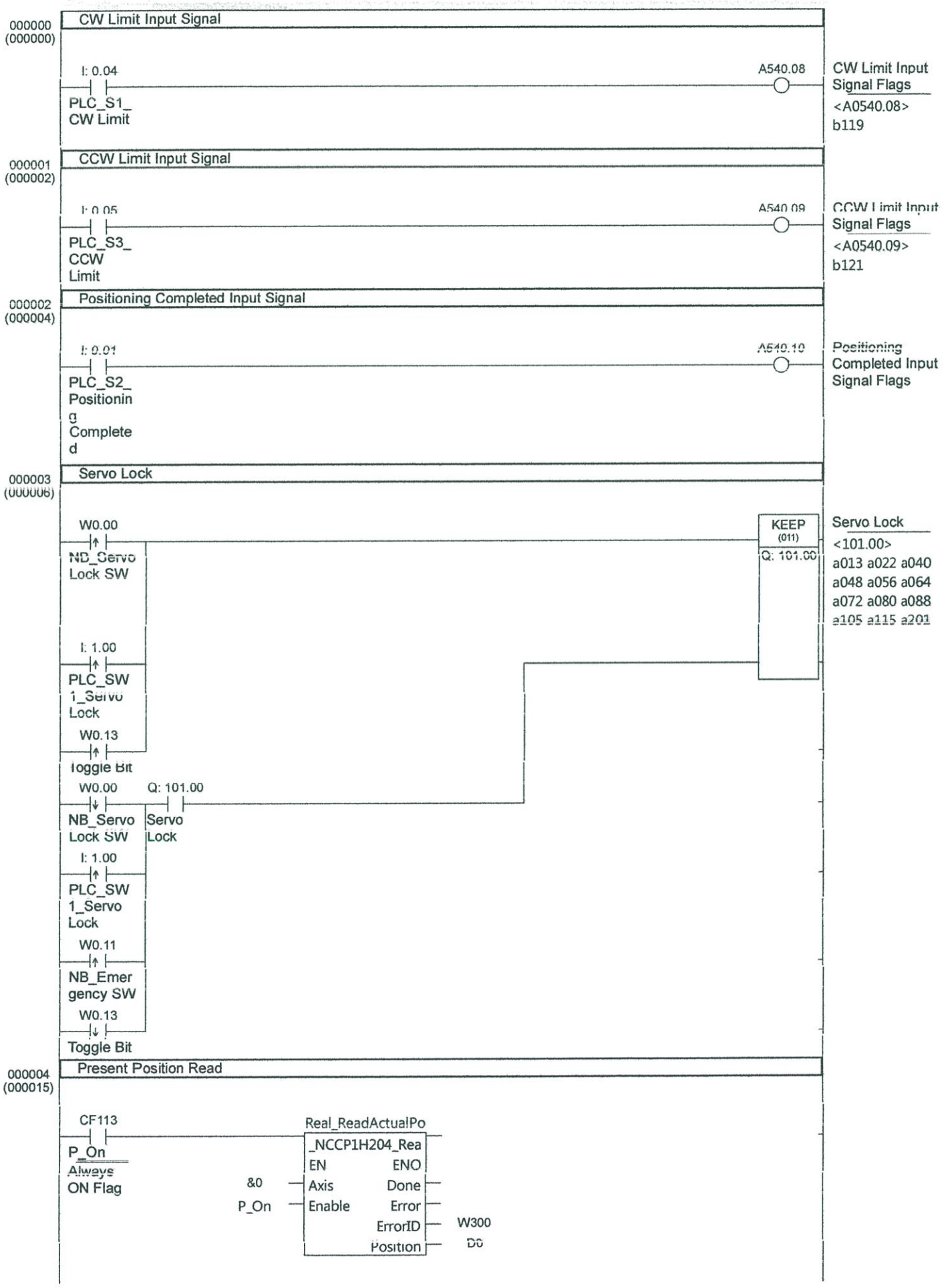
[4] SYSMAC CP Series CP1H CPU Unit. Operation Manual. OMRON

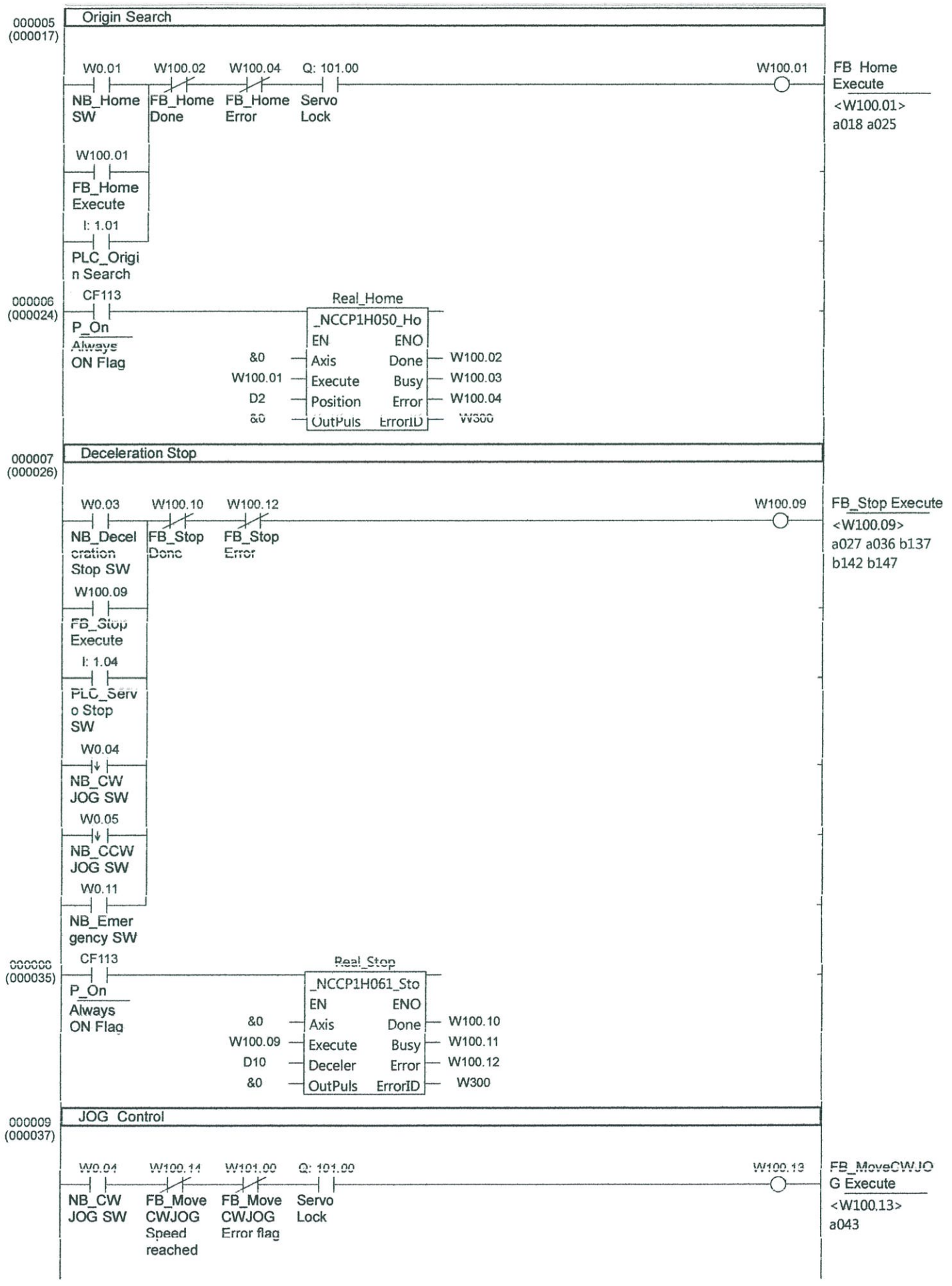
[5] Omron. NB-Series Cx-Designer Introduction Guide

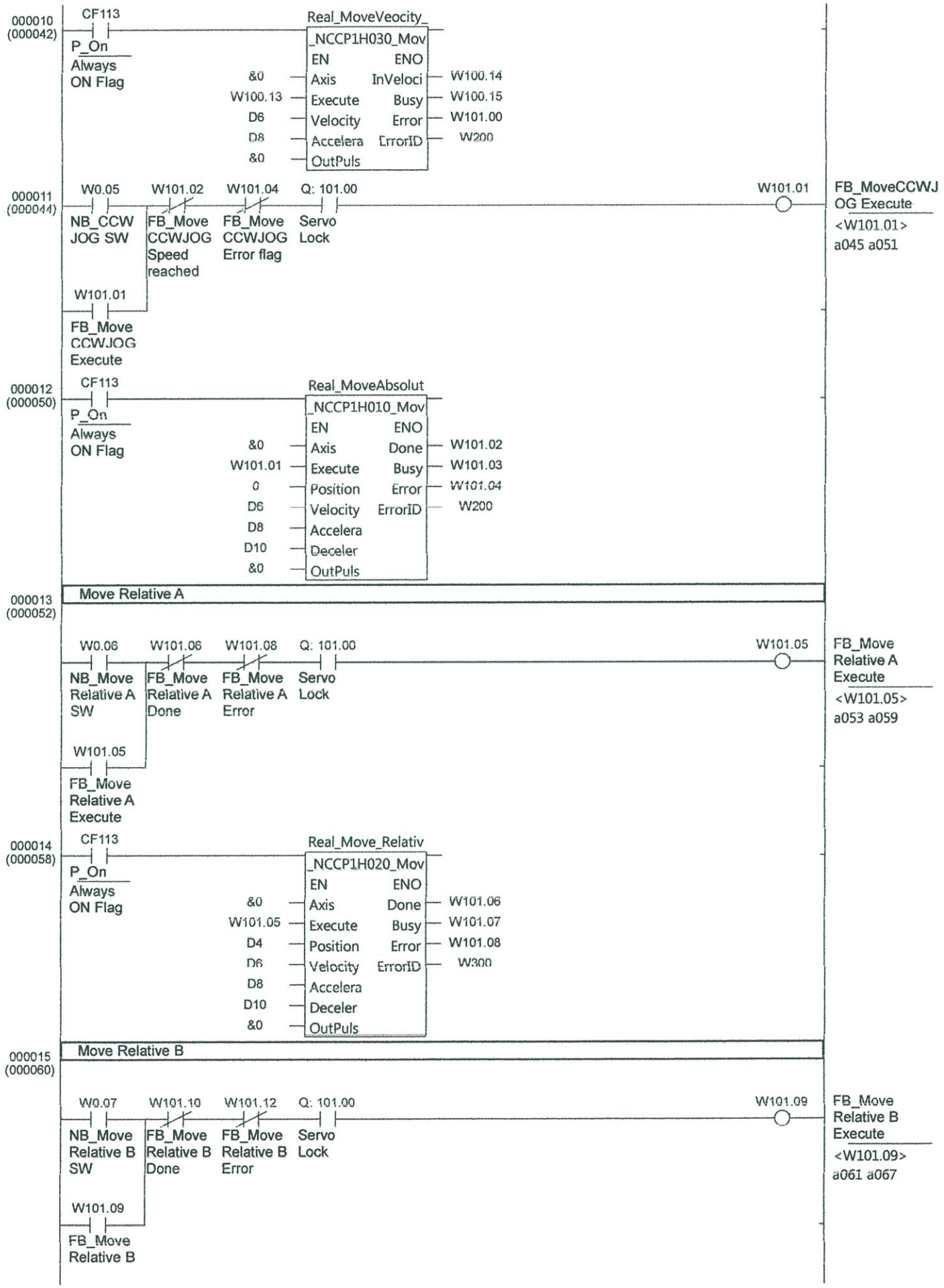
[6] Omron. Cx-One Introduction Guide

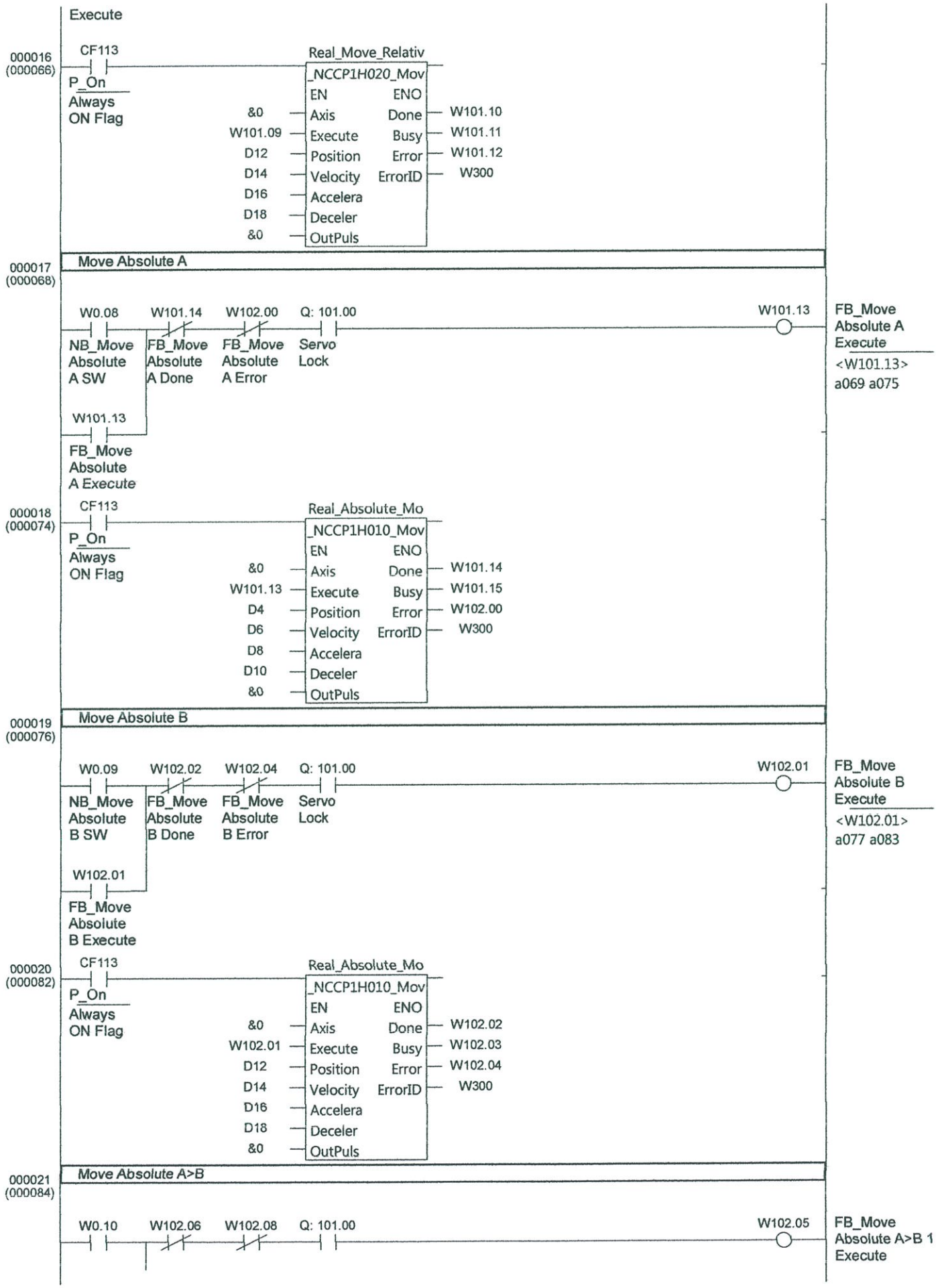
[7] Omron. คู่มือการใช้งานเซอร์โวมอเตอร์ Servo Motor. บริษัท ออมรอน อิเล็กทรอนิกส์ จำกัด

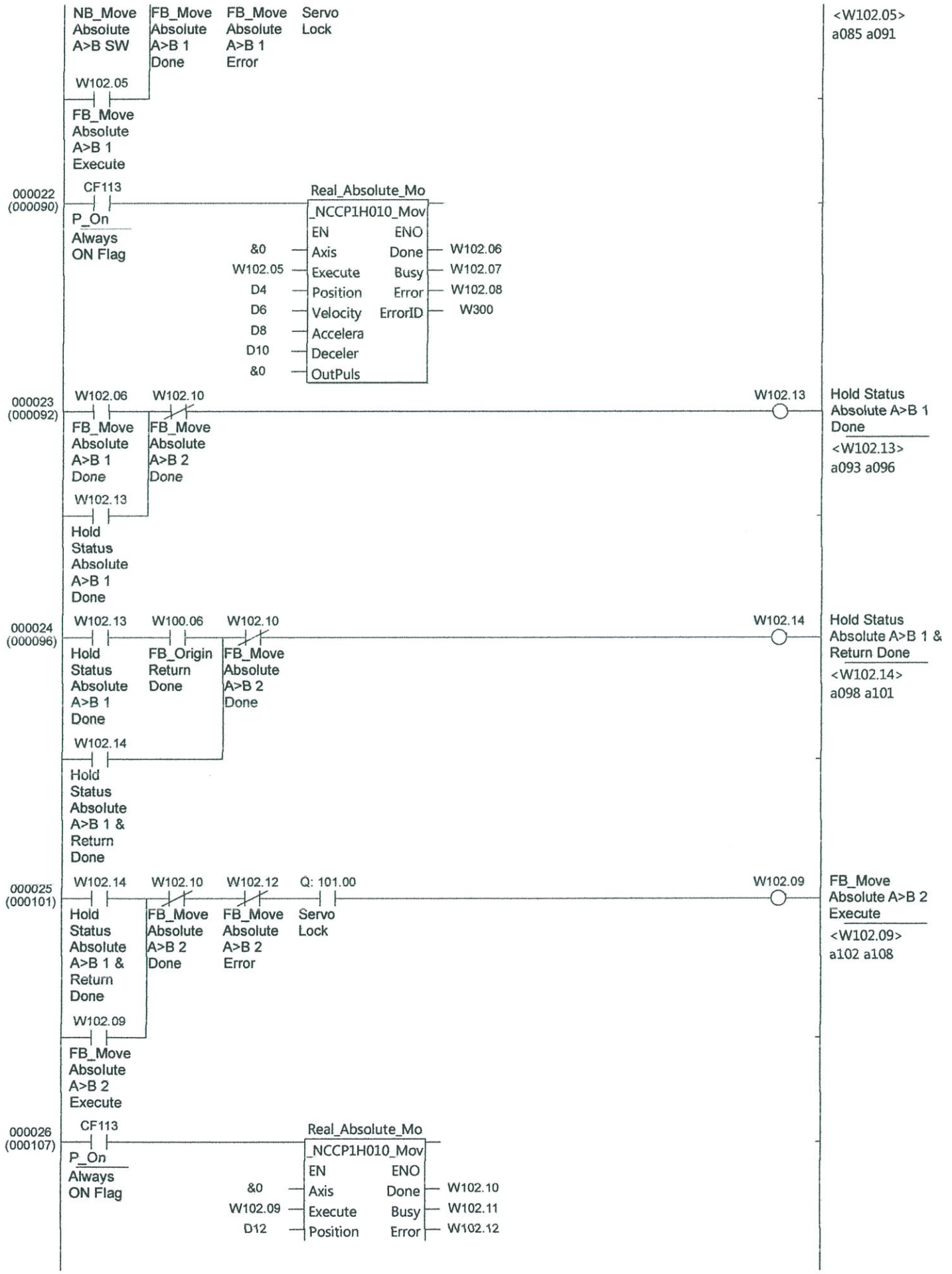
[8] Omron. คู่มือ PLC. บริษัท ออมรอน อิเล็กทรอนิกส์ จำกัด

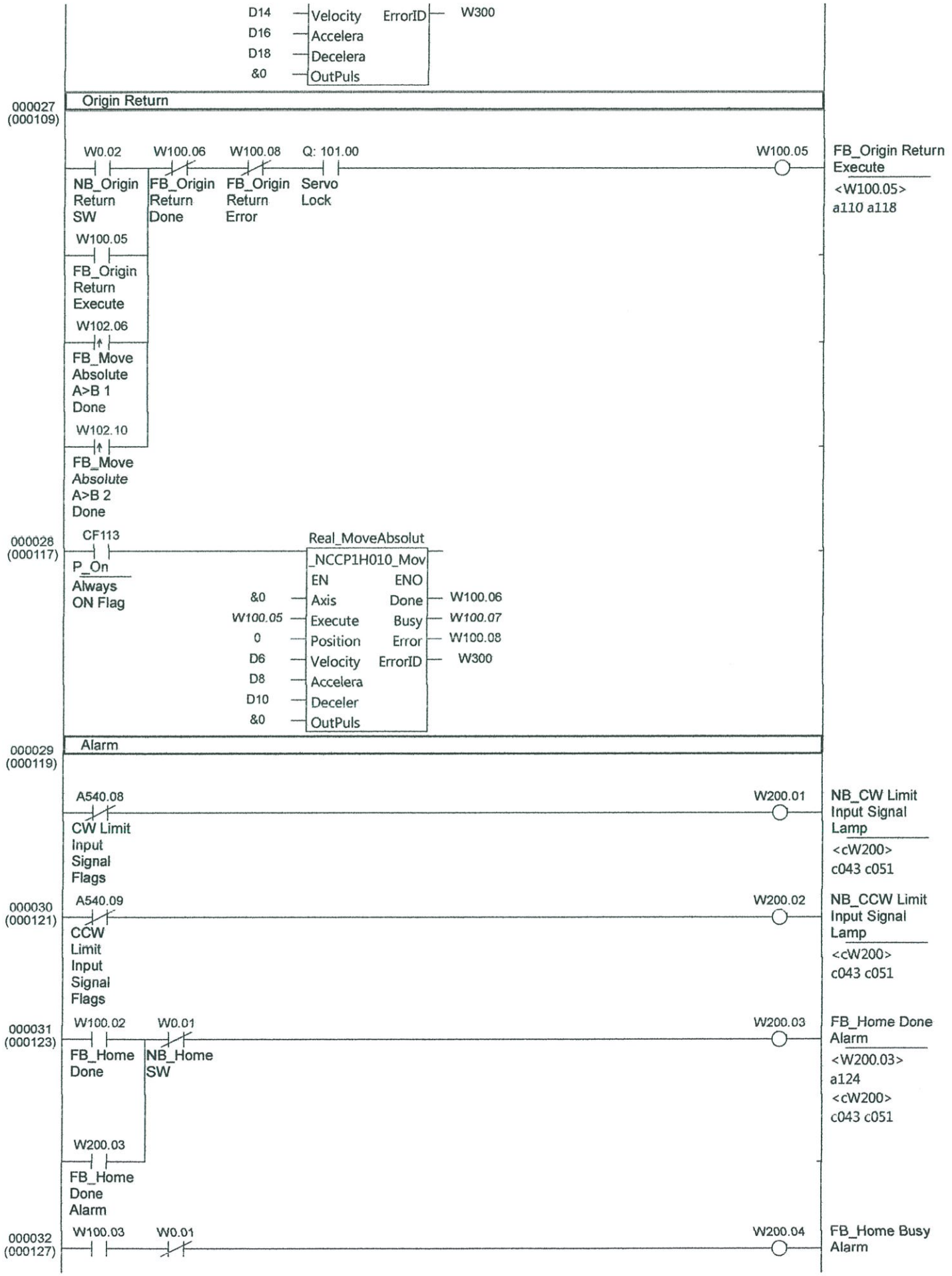

000000
(000000)
CW Limit Input Signal
I: 0.04
PLC_S1_
CW Limit
A540.08
CW Limit Input
Signal Flags
<A0540.08>
b119
000001
(000002)
CCW Limit Input Signal
I: 0.05
PLC_S3_
CCW
Limit
A540.09
CCW Limit Input
Signal Flags
<A0540.09>
b121
000002
(000004)
Positioning Completed Input Signal
I: 0.01
PLC_S2_
Positionin
g
Complete
d
A540.10
Positioning
Completed Input
Signal Flags
000003
(000006)
Servo Lock
W0.00
NB_Servo
Lock SW
I: 1.00
PLC_SW
1_Servo
Lock
W0.13
Toggle Bit
KEEP
(011)
Q: 101.00
Servo Lock
<101.00>
a013 a022 a040
a048 a056 a064
a072 a080 a088
a105 a115 a201
W0.00
NB_Servo
Lock SW
Q: 101.00
Servo
Lock
I: 1.00
PLC_SW
1_Servo
Lock
W0.11
NB_Emer
gency SW
W0.13
Toggle Bit
000004
(000015)
Present Position Read
CF113
P_On
Always
ON Flag
Real_ReadActualPo
_NCCP1H204_Rea
EN
ENO
&0
Axis
Done
P_On
Enable
Error
ErrorID
W300
Position
D0

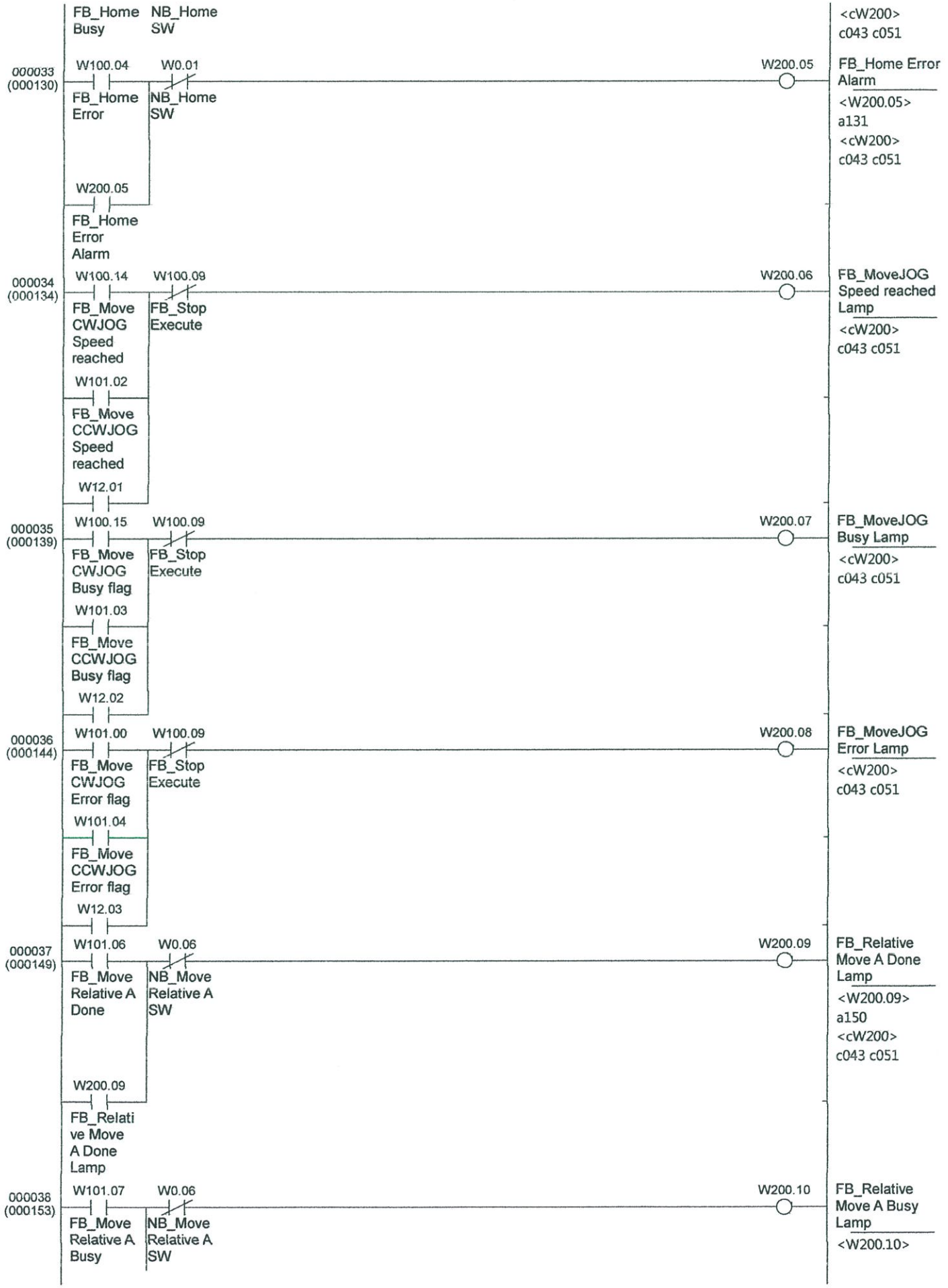

000005
(000017)
Origin Search
W0.01
NB_Home SW
W100.02
FB_Home Done
W100.04
FB_Home Error
Q: 101.00
Servo Lock
W100.01
FB_Home Execute
<W100.01>
a018 a025
W100.01
FB_Home Execute
I: 1.01
PLC_Origin Search
000006
(000024)
CF113
P_On Always ON Flag
Real_Home
_NCCP1H050_Ho
EN
ENO
&0
Axis
Done
W100.02
W100.01
Execute
Busy
W100.03
D2
Position
Error
W100.04
&0
OutPuls
ErrorID
W300
000007
(000026)
Deceleration Stop
W0.03
NB_Deceleration Stop SW
W100.10
FB_Stop Done
W100.12
FB_Stop Error
W100.09
FB_Stop Execute
<W100.09>
a027 a036 b137
b142 b147
W100.09
FB_Stop Execute
I: 1.04
PLC_Servo Stop SW
W0.04
NB_CW JOG SW
W0.05
NB_CCW JOG SW
W0.11
NB_Emergency SW
000008
(000035)
CF113
P_On Always ON Flag
Real_Stop
_NCCP1H061_Sto
EN
ENO
&0
Axis
Done
W100.10
W100.09
Execute
Busy
W100.11
D10
Deceler
Error
W100.12
&0
OutPuls
ErrorID
W300
000009
(000037)
JOG Control
W0.04
NB_CW JOG SW
W100.14
FB_Move CWJOG Speed reached
W101.00
FB_Move CWJOG Error flag
Q: 101.00
Servo Lock
W100.13
FB_MoveCWJOG Execute
<W100.13>
a043

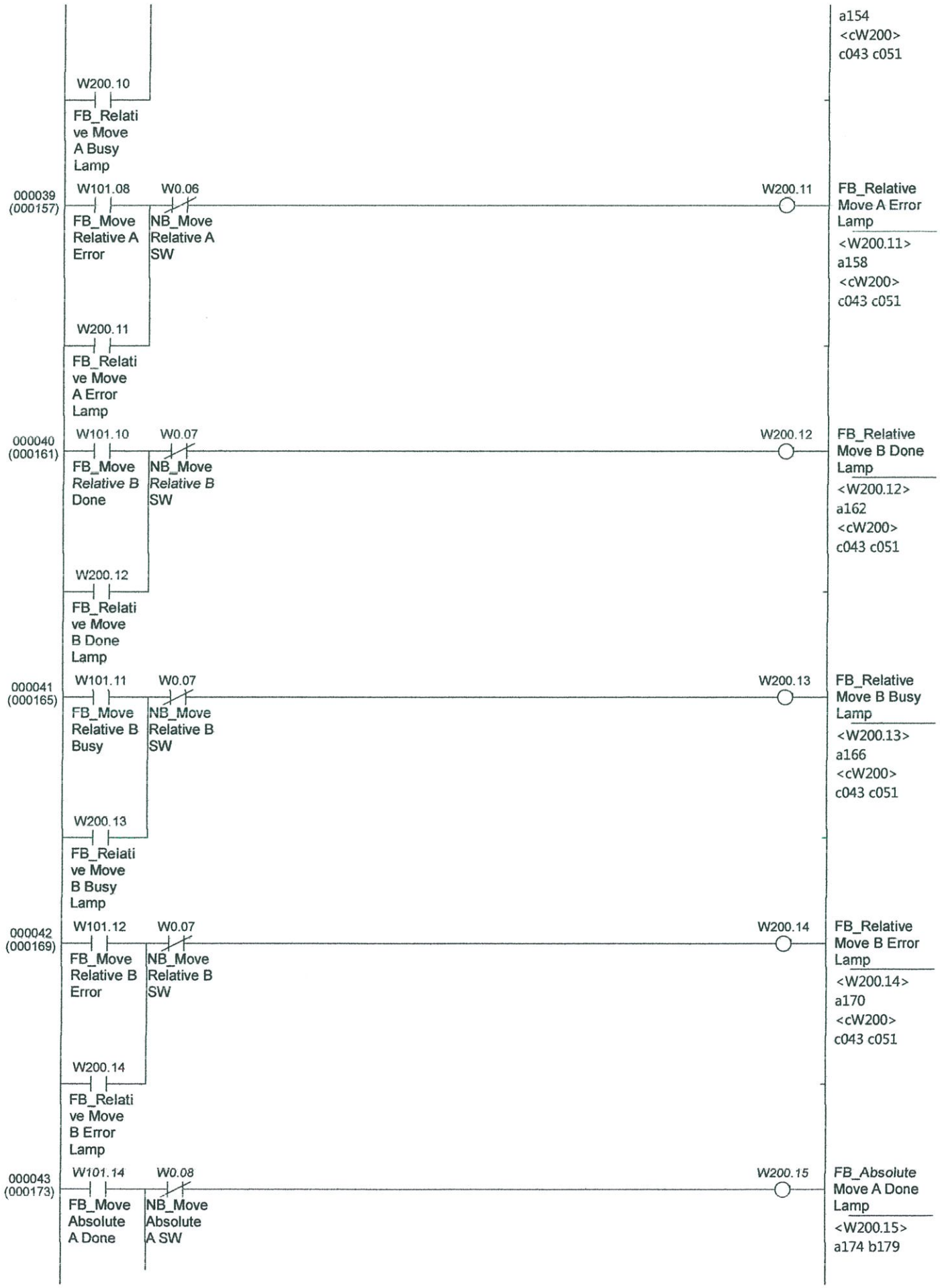

000010
(000042)
CF113
P_On
Always
ON Flag
Real_MoveVeocity_
_NCCP1H030_Mov
EN
ENO
&0
Axis
InVeloci
W100.14
W100.13
Execute
Busy
W100.15
D6
Velocity
Error
W101.00
D8
Accelera
ErrorID
W200
&0
OutPuls
000011
(000044)
W0.05
NB_CCW
JOG SW
W101.02
FB_Move
CCWJOG
Speed
reached
W101.04
FB_Move
CCWJOG
Error flag
Q: 101.00
Servo
Lock
W101.01
FB_MoveCCWJ
OG Execute
<W101.01>
a045 a051
W101.01
FB_Move
CCWJOG
Execute
000012
(000050)
CF113
P_On
Always
ON Flag
Real_MoveAbsolut
_NCCP1H010_Mov
EN
ENO
&0
Axis
Done
W101.02
W101.01
Execute
Busy
W101.03
0
Position
Error
W101.04
D6
Velocity
ErrorID
W200
D8
Accelera
D10
Deceler
&0
OutPuls
000013
(000052)
Move Relative A
W0.06
NB_Move
Relative A
SW
W101.06
FB_Move
Relative A
Done
W101.08
FB_Move
Relative A
Error
Q: 101.00
Servo
Lock
W101.05
FB_Move
Relative A
Execute
W101.05
FB_Move
Relative A
Execute
<W101.05>
a053 a059
000014
(000058)
CF113
P_On
Always
ON Flag
Real_Move_Relativ
_NCCP1H020_Mov
EN
ENO
&0
Axis
Done
W101.06
W101.05
Execute
Busy
W101.07
D4
Position
Error
W101.08
D6
Velocity
ErrorID
W300
D8
Accelera
D10
Deceler
&0
OutPuls
000015
(000060)
Move Relative B
W0.07
NB_Move
Relative B
SW
W101.10
FB_Move
Relative B
Done
W101.12
FB_Move
Relative B
Error
Q: 101.00
Servo
Lock
W101.09
FB_Move
Relative B
Execute
<W101.09>
a061 a067
W101.09
FB_Move
Relative B

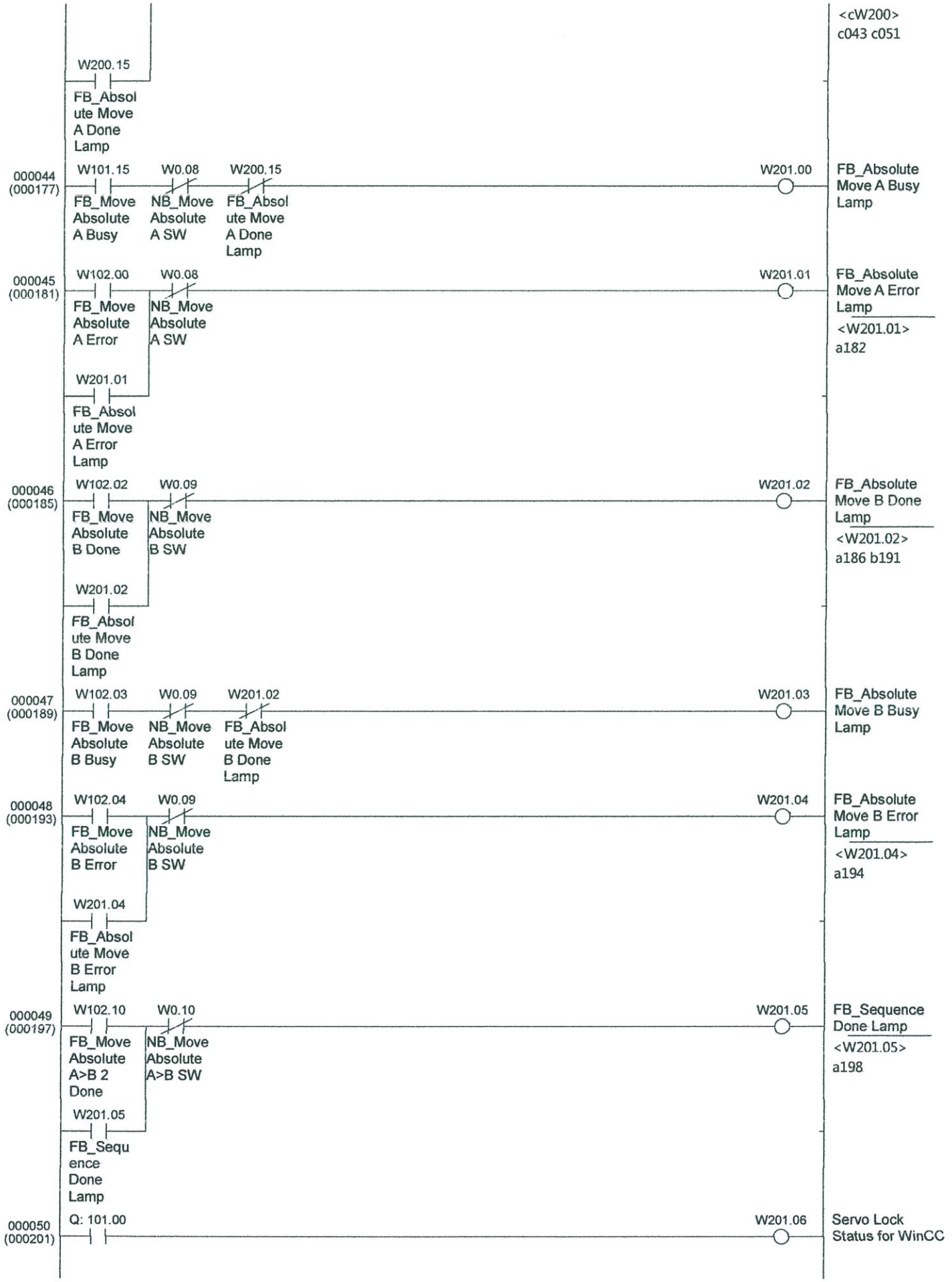

Execute
000016
(000066)
CF113
P_On
Always
ON Flag
Real_Move_Relativ
_NCCP1H020_Mov
EN
ENO
&0
Axis
Done
W101.10
W101.09
Execute
Busy
W101.11
D12
Position
Error
W101.12
D14
Velocity
ErrorID
W300
D16
Accelera
D18
Deceler
&0
OutPuls
000017
(000068)
Move Absolute A
W0.08
W101.14
W102.00
Q: 101.00
W101.13
FB_Move
Absolute A
Execute
<W101.13>
a069 a075
NB_Move
Absolute
A SW
FB_Move
Absolute
A Done
FB_Move
Absolute
A Error
Servo
Lock
W101.13
FB_Move
Absolute
A Execute
000018
(000074)
CF113
P_On
Always
ON Flag
Real_Absolute_Mo
_NCCP1H010_Mov
EN
ENO
&0
Axis
Done
W101.14
W101.13
Execute
Busy
W101.15
D4
Position
Error
W102.00
D6
Velocity
ErrorID
W300
D8
Accelera
D10
Deceler
&0
OutPuls
000019
(000076)
Move Absolute B
W0.09
W102.02
W102.04
Q: 101.00
W102.01
FB_Move
Absolute B
Execute
<W102.01>
a077 a083
NB_Move
Absolute
B SW
FB_Move
Absolute
B Done
FB_Move
Absolute
B Error
Servo
Lock
W102.01
FB_Move
Absolute
B Execute
000020
(000082)
CF113
P_On
Always
ON Flag
Real_Absolute_Mo
_NCCP1H010_Mov
EN
ENO
&0
Axis
Done
W102.02
W102.01
Execute
Busy
W102.03
D12
Position
Error
W102.04
D14
Velocity
ErrorID
W300
D16
Accelera
D18
Deceler
&0
OutPuls
000021
(000084)
Move Absolute A>B
W0.10
W102.06
W102.08
Q: 101.00
W102.05
FB_Move
Absolute A>B 1
Execute


NB_Move Absolute A>B SW
FB_Move Absolute A>B 1 Done
FB_Move Absolute A>B 1 Error
Servo Lock
<W102.05>
a085 a091
W102.05
FB_Move Absolute A>B 1 Execute
000022
(000090)
CF113
P_On
Always ON Flag
Real_Absolute_Mo
_NCCP1H010_Mov
EN
ENO
&0
Axis
Done
W102.06
W102.05
Execute
Busy
W102.07
D4
Position
Error
W102.08
D6
Velocity
ErrorID
W300
D8
Accelera
D10
Deceler
&0
OutPuls
000023
(000092)
W102.06
W102.10
W102.13
FB_Move Absolute A>B 1 Done
FB_Move Absolute A>B 2 Done
Hold Status Absolute A>B 1 Done
<W102.13>
a093 a096
W102.13
Hold Status Absolute A>B 1 Done
000024
(000096)
W102.13
W100.06
W102.10
W102.14
Hold Status Absolute A>B 1 Done
FB_Origin Return Done
FB_Move Absolute A>B 2 Done
Hold Status Absolute A>B 1 & Return Done
<W102.14>
a098 a101
W102.14
Hold Status Absolute A>B 1 & Return Done
000025
(000101)
W102.14
W102.10
W102.12
Q: 101.00
W102.09
Hold Status Absolute A>B 1 & Return Done
FB_Move Absolute A>B 2 Done
FB_Move Absolute A>B 2 Error
Servo Lock
FB_Move Absolute A>B 2 Execute
<W102.09>
a102 a108
W102.09
FB_Move Absolute A>B 2 Execute
000026
(000107)
CF113
P_On
Always ON Flag
Real_Absolute_Mo
_NCCP1H010_Mov
EN
ENO
&0
Axis
Done
W102.10
W102.09
Execute
Busy
W102.11
D12
Position
Error
W102.12


D14
D16
D18
&0
Velocity
Accelera
Decelera
OutPuls
ErrorID
W300
000027
(000109)
Origin Return
W0.02
NB_Origin Return SW
W100.06
FB_Origin Return Done
W100.08
FB_Origin Return Error
Q: 101.00
Servo Lock
W100.05
FB_Origin Return Execute
W100.05
FB_Origin Return Execute
W102.06
FB_Move Absolute A>B 1 Done
W102.10
FB_Move Absolute A>B 2 Done
000028
(000117)
CF113
P_On Always ON Flag
Real_MoveAbsolut
_NCCP1H010_Mov
EN
ENO
&0
Axis
Done
W100.06
W100.05
Execute
Busy
W100.07
0
Position
Error
W100.08
D6
Velocity
ErrorID
W300
D8
Accelera
D10
Deceler
&0
OutPuls
000029
(000119)
Alarm
A540.08
CW Limit Input Signal Flags
W200.01
NB_CW Limit Input Signal Lamp
<cW200>
c043 c051
000030
(000121)
A540.09
CCW Limit Input Signal Flags
W200.02
NB_CCW Limit Input Signal Lamp
<cW200>
c043 c051
000031
(000123)
W100.02
FB_Home Done
W0.01
NB_Home SW
W200.03
FB_Home Done Alarm
<W200.03>
a124
<cW200>
c043 c051
W200.03
FB_Home Done Alarm
000032
(000127)
W100.03
W0.01
W200.04
FB_Home Busy Alarm


FB_Home Busy
NB_Home SW
<cW200> c043 c051
000033 (000130)
W100.04
W0.01
W200.05
FB_Home Error Alarm
FB_Home Error
NB_Home SW
<W200.05> a131 <cW200> c043 c051
W200.05
FB_Home Error Alarm
000034 (000134)
W100.14
W100.09
W200.06
FB_MoveJOG Speed reached Lamp
FB_Move CWJOG Speed reached
FB_Stop Execute
<cW200> c043 c051
W101.02
FB_Move CCWJOG Speed reached
W12.01
000035 (000139)
W100.15
W100.09
W200.07
FB_MoveJOG Busy Lamp
FB_Move CWJOG Busy flag
FB_Stop Execute
<cW200> c043 c051
W101.03
FB_Move CCWJOG Busy flag
W12.02
000036 (000144)
W101.00
W100.09
W200.08
FB_MoveJOG Error Lamp
FB_Move CWJOG Error flag
FB_Stop Execute
<cW200> c043 c051
W101.04
FB_Move CCWJOG Error flag
W12.03
000037 (000149)
W101.06
W0.06
W200.09
FB_Relative Move A Done Lamp
FB_Move Relative A Done
NB_Move Relative A SW
<W200.09> a150 <cW200> c043 c051
W200.09
FB_Relative Move A Done Lamp
000038 (000153)
W101.07
W0.06
W200.10
FB_Relative Move A Busy Lamp
FB_Move Relative A Busy
NB_Move Relative A SW
<W200.10>


a154
<cW200>
c043 c051
W200.10
FB_Relati
ve Move
A Busy
Lamp
000039
(000157)
W101.08
FB_Move
Relative A
Error
W0.06
NB_Move
Relative A
SW
W200.11
FB_Relative
Move A Error
Lamp
<W200.11>
a158
<cW200>
c043 c051
W200.11
FB_Relati
ve Move
A Error
Lamp
000040
(000161)
W101.10
FB_Move
Relative B
Done
W0.07
NB_Move
Relative B
SW
W200.12
FB_Relative
Move B Done
Lamp
<W200.12>
a162
<cW200>
c043 c051
W200.12
FB_Relati
ve Move
B Done
Lamp
000041
(000165)
W101.11
FB_Move
Relative B
Busy
W0.07
NB_Move
Relative B
SW
W200.13
FB_Relative
Move B Busy
Lamp
<W200.13>
a166
<cW200>
c043 c051
W200.13
FB_Relati
ve Move
B Busy
Lamp
000042
(000169)
W101.12
FB_Move
Relative B
Error
W0.07
NB_Move
Relative B
SW
W200.14
FB_Relative
Move B Error
Lamp
<W200.14>
a170
<cW200>
c043 c051
W200.14
FB_Relati
ve Move
B Error
Lamp
000043
(000173)
W101.14
FB_Move
Absolute
A Done
W0.08
NB_Move
Absolute
A SW
W200.15
FB_Absolute
Move A Done
Lamp
<W200.15>
a174 b179


<cW200>
c043 c051
W200.15
FB_Absol ute Move A Done Lamp
000044 (000177)
W101.15
FB_Move Absolute A Busy
W0.08
NB_Move Absolute A SW
W200.15
FB_Absol ute Move A Done Lamp
W201.00
FB_Absolute Move A Busy Lamp
000045 (000181)
W102.00
FB_Move Absolute A Error
W0.08
NB_Move Absolute A SW
W201.01
FB_Absol ute Move A Error Lamp
W201.01
FB_Absolute Move A Error Lamp
<W201.01>
a182
000046 (000185)
W102.02
FB_Move Absolute B Done
W0.09
NB_Move Absolute B SW
W201.02
FB_Absol ute Move B Done Lamp
W201.02
FB_Absolute Move B Done Lamp
<W201.02>
a186 b191
000047 (000189)
W102.03
FB_Move Absolute B Busy
W0.09
NB_Move Absolute B SW
W201.02
FB_Absol ute Move B Done Lamp
W201.03
FB_Absolute Move B Busy Lamp
000048 (000193)
W102.04
FB_Move Absolute B Error
W0.09
NB_Move Absolute B SW
W201.04
FB_Absol ute Move B Error Lamp
W201.04
FB_Absolute Move B Error Lamp
<W201.04>
a194
000049 (000197)
W102.10
FB_Move Absolute A>B 2 Done
W0.10
NB_Move Absolute A>B SW
W201.05
FB_Sequ ence Done Lamp
W201.05
FB_Sequence Done Lamp
<W201.05>
a198
000050 (000201)
Q: 101.00
W201.06
Servo Lock Status for WinCC

Servo
Lock

000051
(000203)

Servo LockToggle SW for Wincc

W0.12
WinCC
Servo
Lock

W0.13
Toggle Bit

KEEP
(011)
W0.13

Toggle Bit
<W000.13>
a008 a012 b204
a206

W0.12
WinCC
Servo
Lock

W0.13
Toggle Bit