รายงานสหกิจศึกษาฉบับสมบูรณ์

การปรับปรุงระบบควบคุมสำหรับเครื่องขึ้นรูปโครงสร้างยาง
Revamping Control System for Tyre Building Machine

นาย ชนาธิป ส่องสว่างธรรม

หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

การปรับปรุงระบบควบคุมสำหรับเครื่องขึ้นรูปโครงสร้างยาง
Revamping Control System for Tyre Building Machine

นาย ชนาธิป ส่องสว่างธรรม

หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ชื่อโครงงานสหกิจศึกษา การปรับปรุงระบบควบคุมสำหรับเครื่องขึ้นรูปโครงสร้างยาง
ชื่อ – สกุล นักศึกษา นายชนาธิป ส่องสว่างธรรม
คณะ วิศวกรรมศาสตร์ **สาขาวิชา** วิศวกรรมอัตโนมัติ
ชื่อ – สกุล อาจารย์นิเทศ ผศ.ดร.ธีรวัฒน์ เทพมณี
รศ.ดร.อัมพวัน จุลเสรีวงศ์
ชื่อ – สกุล ผู้นิเทศงาน คุณศรัณย์ สายกระสุน
คุณไตรทศ ไชยชนะ
ชื่อสถานประกอบการ บริษัท เอ็กซ์เปอร์ต ออโตเมชั่น จำกัด

บทคัดย่อ

โครงงานนี้นำเสนอเทคนิคในการปรับปรุงระบบควบคุมสำหรับเครื่องขึ้นรูปโครงสร้างยาง โดยที่ใช้พีแอลซีรุ่น PLC-5 และตัวควบคุมตำแหน่ง IMCS ที่ล้าสมัยจะถูกแทนที่ด้วยพีแอลซีรุ่น ControlLogix และ ตัวควบคุมตำแหน่งแอนาล็อกเซอร์โวโมดูลรุ่น M02AE ที่ทันสมัยตามลำดับ ในระบบควบคุมใหม่พีแอลซีใช้ในการควบคุมทั้งลำดับการทำงานเครื่องขึ้นรูปโครงสร้างยาง และตำแหน่งของมอเตอร์ ส่วนของแอนาล็อกเซอร์โวรุ่น M02AE ใช้ในการตรวจเช็คตำแหน่งของมอเตอร์ที่ถูกควบคุมแบบป้อนกลับไปยังพีแอลซีนอกจากนี้ยังมีการรวบรวมตัวแปรและแท็กที่ติดตั้งเพื่อแก้ไขในส่วนกับผู้ใช้ (User Interface) และ Human Machine Interface (HMI) ที่มีอยู่ให้สามารถใช้งานได้ในระบบควบคุมใหม่ สามารถยืนยันได้ด้วยผลการการทดสอบ

คำสำคัญ : ControlLogix, อุปกรณ์ควบคุมตำแหน่ง,ตัวควบคุม

Co-operative Title: Revamping Control System for Tyre Building Machine
Student intern name: Mr. Chanatip Songsawangthum
Faculty: Engineering **Department:** Automation Engineering
Advisor name: Asst. Prof. Dr. Teerawat Thepmanee
Assoc. Prof. Dr. Amphawan Julsereewong
Mentor name: Mr. Sarun Saikrasoon
Mr. Trithod Chaichana
Company: Expert Automation Co.,Ltd

ABSTRACT

This project presents a technique to revamp control system for tyre building machine. The out-of-date programmable logic controller (PLC) modeled PLC-5 and position controller modeled IMC-S are replaced by the up-to-date PLC modeled ControlLogix and servo module M02AE, respectively. In the improved system,the new PLC is used for controlling both of machine operation sequence and motor position. The M02AE module is employed for providing position feedback output. Besides, The list of new parameters and tags is also collected to correctthe existing human machine interface(HMI) for use with revamped control system. The workability of the revamped control system is confirmed through test results.

Keyword: ControlLogix , Motion Control , Controller

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้ลุล่วงด้วยดี เนื่องด้วยความอนุเคราะห์จากบริษัท เอ็กซ์เปอร์ต ออโตเมชั่น จำกัด ที่ให้โอกาสในโครงการสหกิจศึกษา อีกทั้งคุณผู้นิเทศงานคุณศรัณย์ สายกระสุน ผู้ดูแลคุณไตรทศ ไชยชนะ และพนักงานบริษัททุกท่านที่ให้ความช่วยเหลือตลอดระยะเวลาสี่เดือน

ขอขอบพระคุณ ผศ.ดร.ธีรวัฒน์ เทพมณีและรศ.ดร.อัมพวัน จุลเสรีวงศ์ ที่ได้ให้คำแนะนำและความช่วยเหลือแก่ผู้จัดทำตลอดมา ขอขอบคุณอาจารย์ภาควิชาวิศวกรรมอัตโนมัติทุกท่านที่ได้ให้คำแนะนำตลอดจนช่วยเหลืออันเป็นประโยชน์ต่อการทำรายงานสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้

ผู้จัดทำรายงานสหกิจศึกษาฉบับสมบูรณ์ ขอขอบพระคุณทุกท่านอย่างสูงที่ให้ การสนับสนุน เอื้อเฟื้อและให้ความอนุเคราะห์ช่วยเหลือ และประโยชน์อันพึงมีจากรายงานสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

ชนาธิป ส่องสว่างธรรม

สารบัญ

หน้า

บทคัดย่อ I
ABSTRACT II
กิตติกรรมประกาศ III
สารบัญ IV
สารบัญภาพ VII
สารบัญตาราง XI
บทที่1 บทนำ 1
1.1 ความเป็นมาและความสำคัญ 1
1.2 วัตถุประสงค์ของโครงงาน 1
1.3 ขอบเขตของปริญญานิพนธ์ 1
1.4 ขั้นตอนการศึกษา 2
1.5 ประโยชน์ที่คาดว่าจะได้รับ 2
บทที่2 ทฤษฎีที่เกี่ยวข้อง 3
2.1 แนวคิดและรูปแบบของกระบวนการผลิตยางจักรยาน 3
2.1.1 แนวคิดของกระบวนการผลิต 3
2.1.2 รูปแบบของกระบวนการผลิต 3
2.2 พีแอลซี (PLC) 5
2.3 โครงสร้างพีแอลซี(PLC) 6
2.3.3 หน่วยรับข้อมูลหรืออินพุท (Input) 7
2.3.4 หน่วยส่งข้อมูลหรือเอาท์พุท (Output) 7
2.3.5 แหล่งจ่ายไฟ (Power Supply) 8
2.4 พีแอลซี PLC-5 8
2.4.1 ประวัติ PLC-5 8
2.4.2 การ Addressing และ Tagging 9
2.4.3 การใช้ Addressing mode 10
2.5 พีแอลซี ControlLogix 11
2.5.1 ประวัติ ControlLogix 11
2.5.2 CPU สำหรับ ControlLogix 11

สารบัญ(ต่อ)

หน้า

2.5.3 I/O Card สำหรับ ControlLogix 12
2.6 จีเอ็มแอล คอมมานเดอร์ (GML Commander) 13
2.6.1 ชุดคำสั่ง HOME 13
2.6.2 ชุดคำสั่ง MOVE 14
2.6.3 ชุดคำสั่ง JOG 17
2.6.5 ชุดคำสั่ง CHANGE DYNAMICS 19
2.6.6 ชุดคำสั่ง STOP MOTION 20
2.6.7 ชุดคำสั่ง FEEDBACK 20
2.6.8 ชุดคำสั่ง RESET FAULT 21
2.6.10 ชุดคำสั่ง ON AXIS 22
2.6.11 ชุดคำสั่ง INPUT 22
2.6.12 ชุดคำสั่ง ON EXPRESSION 23
2.6.13 ชุดคำสั่ง ON TIMEOUT 24
2.6.14 ชุดคำสั่ง IF AXIS FAULT 24
2.6.15 ชุดคำสั่ง ON TASK 25
2.6.18 ชุดคำสั่ง OUTPUT 27
2.6.19 ชุดคำสั่ง EQUATION 27
2.7 ไอเอ็มซีเอส (IMC-S) 29
2.8 Analog Encoder (AE) Servo Module 33
2.9 Motor Driver 35
2.10 Conversion Module 41
2.11 Incremental Encoder 42
บทที่3 ขั้นตอนการดำเนินงาน 43
3.1 แนวคิด 43
3.1.1 รูปแบบของการควบคุมในระบบเก่า 44
3.1.2 รูปแบบของการควบคุมในระบบใหม่ 45
3.2 วิธีการดำเนินงาน 47
3.2.1 ศึกษาการทำงานของระบบเดิม 47

สารบัญ(ต่อ)

หน้า

3.2.2 การออกแบบ Loop Wiring 51
3.2.3 การติดตั้งอุปกรณ์และพีแอลซี 58
3.2.4 การแปลงโปรแกรมลงใน Studio5000 62
3.2.5 การตั้งค่า Motion Group ในแต่ละแกน 71
3.2.7 การ Hook up , Tune , และ Motion Direct Commands 78
3.2.8 การ Intitailization โปรแกรม 81
บทที่4 ผลการดำเนินงาน 82
4.1 กล่าวนำ 82
4.2 ขั้นตอนการทดสอบ 82
บทที่5 สรุปผลและข้อเสนอแนะ 88
5.1 สรุปผล 88
5.2 ปัญหาและวิธีการแก้ไขปัญหา 88
5.2.1 ปัญหาที่พบ 88
5.2.2 วิธีการแก้ไข้ปัญหา 88
5.3 ข้อเสนอแนะ 88
เอกสารอ้างอิง 89

สารบัญภาพ

ภาพที่ **หน้า**

2.1 ขั้นตอนการเตรียมโครงสร้างยาง....4
2.2 โครงสร้างของยางจักรยาน....4
2.3 ตัวอย่างพีแอลซี MicroLogix1100 ของ Allen Bradley....5
2.4 โครงสร้างภายใน PLC....6
2.5 หน่วยความ Processor ของ PLC-5....8
2.6 ตารางคุณสมบัติส่วนหนึ่งของ PLC-5 แต่ละรุ่น....9
2.7 การอ้างอิงตำแหน่งของ PLC-5....9
2.8 รูปแบบของ Addressing....10
2.9 การใช้สล็อท(Slot) และกรุ๊ป(Group) เพื่อหาจำนวนแร็ค(rack)....10
2.10 ราง(Chassis) ของ ControlLogix....11
2.11 CPU ของ ControlLogix5580....12
2.12 ตัวอย่างหน้าต่างการทำงาน GML Commander....13
2.13 ชุดคำสั่ง HOME....13
2.14 ชุดคำสั่ง MOVE....14
2.15 ลักษณะการทำงานของ Phases Shift Move....16
2.16 การทำงานของ Synchronizing Moves....17
2.17 ชุดคำสั่ง JOG....17
2.18 รูปแบบการเคลื่อนที่ของ Trapezoidal jog....18
2.19 ชุดคำสั่ง Gear Axes....18
2.20 ชุดคำสั่ง Change Dynamics....19
2.21 ชุดคำสั่ง Stop Motion....20
2.22 ชุดคำสั่ง Feedback....20
2.23 ชุดคำสั่ง Reset Fault....21
2.24 ชุดคำสั่ง Control Setting....21
2.25 ชุดคำสั่ง On Axis....22
2.26 ชุดคำสั่ง Input....22
2.27 Dedicated , Configured , Miscellaneous....23

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

2.28 ชุดคำสั่ง On Expression....23
2.29 ชุดคำสั่ง On Timeout....24
2.30 ชุดคำสั่ง If Axis Fault....24
2.31 ชุดคำสั่ง On Task....25
2.32 ชุดคำสั่ง Task Control....25
2.33 การทำงานของ Multitasking....26
2.34 การทำงานของ Set Timer....26
2.35 ชุดคำสั่ง Output....27
2.36 ชุดคำสั่ง Equation....27
2.37 IMC-S Motion Controller....29
2.38 IMC-S/234 Connection....30
2.39 คุณสมบัติเฉพาะของ Servo....31
2.40 หน้าการ์ด I/O IMCS....31
2.41 การ์ด 1756-M02AE....33
2.42 เทอรมินัลแบบ Servo module RTB....33
2.43 UNI1401 (ซ้าย) , UNI1403 (ขวา) , UNI1406 (ขวา)....36
2.44 Power Connection UNI1403 และ UNI1406....36
2.45 15-pins Adaptor Terminal....37
2.46 Wiring ของ Connection Terminal UNI1403 และ UNI1406....37
2.47 Power Connection UNI1401....38
2.48 Power Connection ส่วนเสริม Second Encoder....39
2.49 Wiring ของ Connection Terminal UNI1401....40
2.50 Conversion Module(ซ้าย) เทอร์มินอล(ขวา)ของ PLC-5....41
2.51 Base Plate และ Cover Plate....41
2.52 Incremental Encoder....42
3.1 แผนภูมิลำดับการทำงานของกระบวนการ....43
3.2 ฟังก์ชันบล็อกการควบคุมมอเตอร์....44
3.3 ภาพรวมของระบบ Module4 เดิม....44

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

3.4 ภาพรวมของระบบ Module4 ใหม่ 45
3.5 ภาพรวมจากมุมบนของ Module4 48
3.6 ตัวอย่างของ Rotation Tambour Confection 49
3.7 ตัวอย่างของ Loop Wiring 51
3.8 Limit Switch ของ IMCS1 แกน 2 52
3.9 Home Limit Switch ของ IMCS4 แกน 0 53
3.10 Limit Switch ของ IMCS4 แกน 1 53
3.11 Servo Feedback ของ IMCS4 แกน 1 54
3.12 Loop Wiring M02AE ไป UNI1405 แกนที่ 2 55
3.13 Loop Wiring ของ Overtravel ที่การ์ด 1756-IB32 57
3.14 แร็ค Local 58
3.15 RIO1 หรือ แร็คที่ 1 59
3.16 RIO2 หรือ แร็คที่ 2 59
3.17 แร็ค Local,RIO1 และ RIO2 เมื่อทำการประกอบการ์ดลงไป 60
3.18 ตัวอย่างแร็คที่ประกอบ Conversion Module ยังไม่ประกอบ Pre-Wired Cable 61
3.19 หน้าต่าง Translate PLC-5/SLC2.0 62
3.20 หน้าต่างเลือกคอนโทรลเลอร์ใน Studio5000 63
3.21 การทำงานของ IMCS1 ใน GML Commander 64
3.22 หน้าต่าง Controller Organizer 65
3.23 ตัวอย่างบางส่วนของ R000_MainSeq 66
3.24 ตัวอย่างบางส่วนของ R000_MainSeq ส่วน Initialisation 66
3.25 ตัวอย่างบางส่วนของ R000_MainSeq ส่วน Request 67
3.26 บล็อกการทำงาน GML Commader ของ Default Global 68
3.27 เลดเดอร์ R200_ Default Global ของ Default Global 69
3.28 User Constant ใน Data Exchange ใน IMCs1 70
3.29 User Varaible ใน Data Exchange ใน IMCs1 71
3.30 หน้าต่าง Axis Properties ของแกนที่ 1 71
3.31 หน้าต่าง Axis Properties ของแกนที่ 1 การตั้งค่า Limit 72

สารบัญภาพ(ต่อ)

ภาพที่ **หน้า**

3.32 Address และ Alias for ของ RIO_R1 อินพุท 73
3.33 การเชื่อม Name, Alias For และ Base tag 74
3.34 ภาพการแลกเปลี่ยนแท็กระบบใหม่และระบบเก่า 75
3.35 Address ของ RIO Output 76
3.36 เทียบ Address กับแท็กในพีแอลซี ของ RIO Adapter 76
3.37 บิทของ IMCS ที่ 1 และ 2 ใน Studio5000 77
3.38 User Constant ของ IMCS3 78
3.39 หน้าต่าง Hookup ใน การตั้งค่าแกน 79
3.40 Tune ใน การตั้งค่าแกน 80
3.41 หน้าต่าง Motion Direct Commands การทำ JOG 80
3.42 ตัวอย่างเลดเดอร์ใน R300_Intitailisation 81
3.43 หน้าต่าง Monitor Tags ของแกนที่ 1 ขณะทำงาน 81
4.1 การ์ดที่ถูกสร้าง RIO_R1,RIO_R2 และ Local จากซ้ายไปขวาตามลำดับ 84
4.2 หน้าต่างโปรแกรมทั้งหมด RsLogix5 84
4.3 หน้าต่างโปรแกรมทั้งหมด Studio5000 85
4.4 Controller Organizer ของโฟลเดอร์เมื่อเสร็จสิ้น 85
4.4 ภาพติดตามการทำงานของทุกแกน 87

สารบัญตาราง

ตารางที่ **หน้า**

1.1 แสดงระยะเวลาการทำงาน....2
2.1 การเปรียบเทียบของตัวขับมอเตอร์ Drive rating....35
3.1 เปรียบการ์ดของระบบเดิมและระบบใหม่....46
3.2 รายละเอียดชื่อของแต่ละแกน....47
3.3 รายละเอียดของ Loop Wiring แต่ละแกน....56
3.4 รายละเอียดของการ์ดในราง Local....58
3.5 รายละเอียดของการ์ดในราง RIO_R1....59
3.6 รายละเอียดของการ์ดในราง RIO_R2....60
3.7 แสดงรายละเอียด Conversion Module และ Pre-Wired Cable 1492 Series....61
3.8 รายละเอียดของแต่ละแกนใน GML Commander....75
4.1 ผลการทดสอบฮาร์ดแวร์ Overtravel Limit และ Home Limit....82
4.2 ผลการทดสอบฮาร์ดแวร์ตัวควบคุมตำแหน่ง (Motion Control)....83
4.3 การทดสอบลำดับการทำงานของโปรแกรม....86

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันการควบคุมแบบลำดับ (Sequence Control) นิยมใช้พีแอลซี (Programmable logic Control) ซึ่งเป็นอุปกรณ์ควบคุมที่สามารถปรับแก้ไขโปรแกรมตามเงื่อนไขที่ต้องการได้ทั้งนี้ บริษัทผู้ผลิตพีแอลซีมีการพัฒนาขีดความสามารถของพีแอลซีอย่างต่อเนื่อง ทำให้บริษัทผู้ผลิตมีการยกเลิกการผลิตและจำหน่ายพีแอลซีรุ่นเก่าๆ ดังนั้นโรงงานที่ใช้พีแอลซีรุ่นเก่าๆ มีปัญหาในการซ่อมบำรุง จึงเป็นเหตุผลให้โรงงานเหล่านั้นจำเป็นต้องปรับปรุงระบบควบคุม โดยทำการใช้พีแอลซีที่มีจำหน่ายในปัจจุบันทดแทนพีแอลซีรุ่นเดิม เช่นเดียวกับโรงงานผลิตยางแห่งหนึ่งที่ได้ศึกษา พบว่าระบบควบคุมของเครื่องจักรที่ใช้ในสายการผลิตต่าง เป็นระบบที่มีการใช้พีแอลซีในรุ่นที่บริษัทผู้ผลิตยกเลิกการผลิตและจำหน่าย

โครงงานนี้นำเสนอการปรับปรุงระบบควบคุมสำหรับเครื่องขึ้นรูปโครงสร้างยาง โดยระบบเดิมใช้พีแอลซีรุ่น PLC-5 ที่เป็นตัวควบคุมระบบและใช้ไอเอ็มซีเอส (IMC-S) เป็นตัวควบคุมตำแหน่ง (Motion Controller) ของมอเตอร์ ซึ่งปัจจุบันอุปกรณ์ทั้งสองถูกยกเลิกการผลิตและจำหน่าย ทำให้เกิดปัญหาในการซ่อมบำรุงในกรณีที่ต้องการจัดซื้อ ดังนั้นโรงงานที่ศึกษาแห่งนี้มีจึงความต้องการจะปรับปรุงระบบควบคุม โดยเลือกใช้พีแอลซีรุ่น ControlLogix และแอนาล็อกเซอร์โวโมดูล (Analog Servo Module) รุ่น M02AE ทดแทน PLC-5 และ IMC-S ตามลำดับ

1.2 วัตถุประสงค์ของโครงงาน

1. ปรับปรุงระบบควบคุมด้วยการเปลี่ยน PLC-5 และ IMC-S ด้วย ControlLogix และ แอนาล็อกเซอร์โวโมดูล M02AE ตามลำดับ โดยยังคงลำดับการทำงานเครื่องขึ้นรูปโครงสร้างยางแบบเดิม
2. รวบรวมตัวแปรและแท็ก (Tag) เพื่อนำมาใช้ในการแก้ไขในส่วนติดต่อผู้ใช้ (User Interface) หรือ HMI (Human Machine Interface)

1.3 ขอบเขตของปริญญานิพนธ์

1. โปรแกรมของพีแอลซี ControlLogix ที่เขียนด้วย Studio5000 ถูกใช้ในการควบคุมลำดับเครื่องขึ้นรูปโครงสร้างยางและตำแหน่งของมอเตอร์ที่รอการสั่งจากแอนาล็อกเซอร์โวโมดูล M02AE

2. โปรแกรมที่ถูกเขียนขึ้นสำหรับขั้นตอนการตั้งค่า (Setup) ก่อนสั่งการเครื่องขึ้นรูปโครงสร้างยางยางเริ่มทำงาน โดยให้ผู้ปฏิบัติงาน (Operator) ปรับค่ารุ่นของยางที่ต้องการผลิตแล้วปรับค่าสู่เริ่มต้น (Intitialization) ก่อนสั่งสตาร์ท (Start)
3. แท็กและโปรแกรมที่ใช้ในการควบคุมแอนาล็อกเซอร์โวโมดูล M02AE มีรูปแบบการควบคุมตำแหน่งของมอเตอร์ด้วยฟังก์ชันการทำงานแบบเดิม

1.4 ขั้นตอนการศึกษา

1. ศึกษาการทำงานของพีแอลซี PLC-5 และตัวควบคุมตำแหน่ง IMC-S
2. ศึกษาการทำงานของพีแอลซี ControLogix และแอนาล็อกเซอร์โวโมดูล M02AE
3. ศึกษาการเขียนโปรแกรมและย้ายระบบ
4. เขียนโปรแกรมและย้ายระบบ
5. ติดตั้งฮาร์ดแวร์และเดินสายไฟ
6. ทดสอบการทำงานของโปรแกรมและฮาร์ดแวร์ทั้งหมด
7. จัดทำเอกสารสำหรับระบบใหม่

ตารางที่ 1.1 ระยะเวลาการทำงาน

ขั้นตอนการดำเนินงาน	ระยะเวลาดำเนินการ															
	ส.ค.				ก.ย.				ต.ค.				พ.ย.			
	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1. ศึกษาการทำงานระบบเก่า																
2. ศึกษาการทำงานระบบใหม่																
3. เขียนโปรแกรมสำหรับระบบใหม่																
4. ติดตั้งและทดสอบการทำงาน																
5. จัดทำเอกสารสำหรับระบบใหม่																
6. จัดทำรูปเล่มรายงาน																

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. รองรับการทำงานที่ซับซ้อนมากขึ้นและ ระดับงานที่มีความยากสูง
2. เพิ่มประสิทธิภาพในการทำงานของเครื่องจักรให้เกิดสเถียรภาพ
3. ง่ายต่อการแก้ไขและตรวจสอบโปรแกรมในอนาคต

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 แนวคิดและรูปแบบของกระบวนการผลิตยางจักรยาน

2.1.1 แนวคิดของกระบวนการผลิต

ในปัจจุบันนั้นเทคโนโลยีการขึ้นผลิตยางจักรยานมีความซับซ้อนขึ้นส่งผลให้ผลิตภัณฑ์มีคุณภาพมากขึ้น ความรวดเร็วในการผลิตยางจักรยานสูงขึ้น ซึ่งแตกต่างกับในอดีตที่ใช้มนุษย์ในการผลิตร่วมกับเครื่องจักร ในเวลาต่อมาถูกพัฒนาขึ้นโดยใช้ระบบเครื่องจักรเต็มตัว หรือระบบอัตโนมัติ (Automation System) ตามเทคโนโลยีในสมัยนั้นๆ

การผลิตยางของบริษัท จำกัดใช้ระบบอัตโนมัติในการควบคุมเครื่องจักรผลิตยางจักรยานให้เป็นขนาดที่ต้องการอีกทั้งเพื่อควบคุมคุณภาพของยางจักรยานซึ่งกระบวนการผลิตจะมีการผลิตด้วย 3 ขั้นตอนหลักๆ

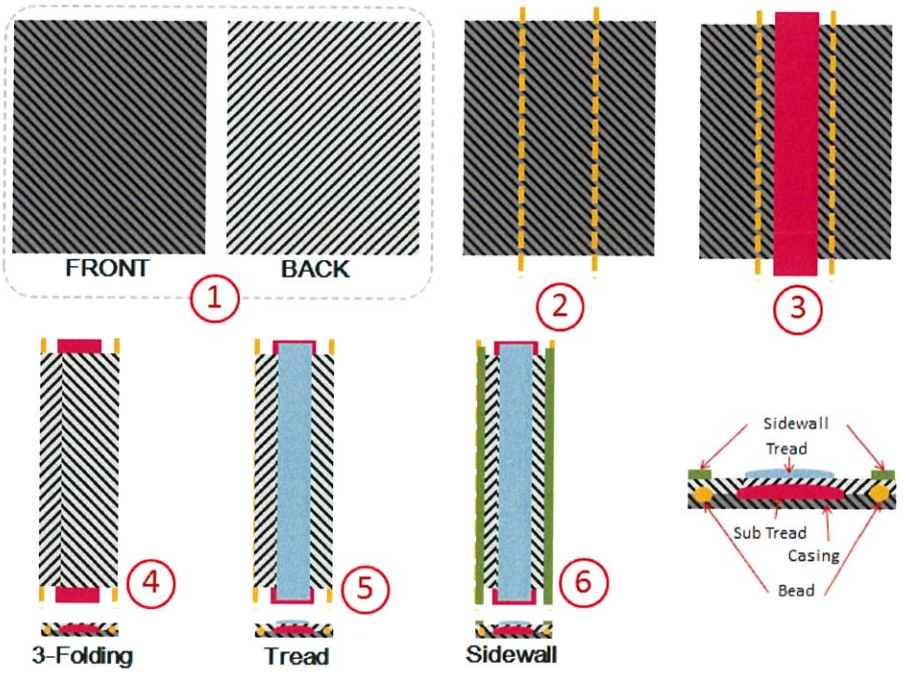
1. การทำแผ่นยาง และส่วนประกอบของยาง
2. การทำโครงสร้างยางสำหรับการขึ้นรูป
3. การขึ้นรูปยาง

ในส่วนที่ได้รับการมอบหมายนั้น คือการปรับปรุงและเปลี่ยนแปลงระบบเตรียมโครงสร้างยางสำหรับการขึ้นรูปที่ทางบริษัทได้ใช้ชื่อว่า Module4 โดยเริ่มแรกนั้นผู้ที่ออกแบบระบบและเครื่องจักรเป็นชาวฝรั่งเศสทำให้การตั้งชื่อ,แท็ก หรือรูปแบบของโปรแกรมจะใช้ภาษาฝรั่งเศสเกือบทั้งหมด

2.1.2 รูปแบบของกระบวนการผลิต

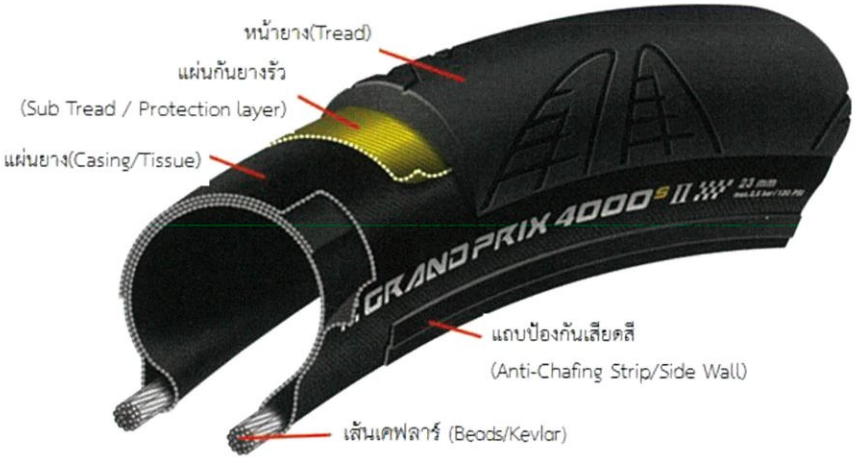
รูปแบบของกระบวนการเตรียมโครงสร้างยางสำหรับการขึ้นรูปนั้นมีความสำคัญต่อผลิตภัณฑ์มากเนื่องจากมีความละเอียดอ่อนในแต่ละขั้นตอน จะมีขั้นตอนในการเตรียมโครงสร้าง 6 ขั้นตอนด้วยกันซึ่งจะสอดคล้องกับการทำงานของเครื่องจักรมีดังนี้ ภาพที่ 2.1

1. เตรียมแผ่นยาง (Casing/Tissue)
2. ใส่เส้นเคฟลาร์ (Beads/Kevlar) บน Casing เพื่อคงรูปจักรยานให้เป็นวง
3. ใส่แผ่นกันยางรั่ว (Sub Tread/Protection layer) บน Casing
4. พับขอบของ Casing ที่ใส่ Beads และ Sub Tread
5. แปะหน้ายาง (Tread) บน Casing ที่พับขอบแล้ว
6. แปะแถบป้องกันเสียดสี (Anti-Chafing Strip/Side Wall)



ภาพที่ 2.1 ขั้นตอนการเตรียมโครงสร้างยาง

เมื่อเสร็จทุกขั้นตอนจะได้ลักษณะคล้ายดังตัวอย่างภาพที่ 2.2 ต่างที่ลักษณะยังไม่เป็นวงกลมเพราะยังไม่ผ่านกระบวนการขึ้นรูป



ภาพที่ 2.2 โครงสร้างของยางจักรยาน

ซึ่งลักษณะของขั้นตอนดังกล่าวจะมาอ้างอิงกับระบบการทำงานของเครื่องจักรการผลิตที่ใช้มอเตอร์ในการควบคุมมีทั้งหมด 14 แกน

2.2 พีแอลซี (PLC)

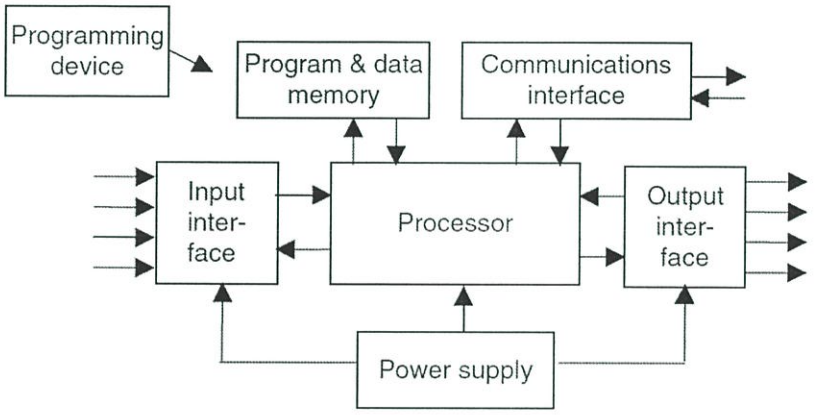
ตัวควบคุมเชิงตรรกะ(Programmable Logic Control) นั้นคืออุปกรณ์ควบคุมชนิดหนึ่งที่สามารถโปรแกรมได้โดยอ้างอิงจาการทำงานทางไฟฟ้าของรีเลย์(Relay)หรือ แมคเนติกคอนเทคเตอร์ (Magnetic Contactor) ที่มีการทำงานรูปแบบตรรกศาสตร์(Logic function) ซึ่งปัจจุบันนั้น ใช้การสร้างโปรแกรมขึ้นมาเพื่อลดการทำงานของรีเลย์หรือ แมคเนติกคอนเทคเตอร์โดยใช้ไมโครคอนโทรลเลอร์ขนาดเล็กในพีแอลซีเป็นตัวควบคุมประมวลผลเงื่อนไขต่างๆ จากอินพุทและเอาท์พุท นอกจากนี้ยังสามารถทำงานร่วมกับอุปกรณ์อื่นๆได้อีก เช่น เครื่องอ่านบาร์โค้ด(Barcode Scanner) เครื่องพิมพ์(Printer) ในปัจจุบันพีแอลซีนอกจากจะใช้งานเดี่ยวๆแล้ว(Stand Alone) ยังนำพีแอลซีหลายๆตัวมาต่อกันด้วยเครือข่าย(Network) ทำให้ระบบมีประสิทธิภาพมากขึ้น ในอุตสาหกรรมจึงมีแนวโน้มการใช้พีแอลซีมากขึ้นเรื่อยๆ

ภาพที่ 2.3 ตัวอย่างพีแอลซี MicroLogix1100 ของ Allen Bradley

การใช้งานพีแอลซีควบคุมเครื่องจักรหรือ อุปกรณ์ต่างๆในโรงงานจะมีข้อได้เปรียบกว่าการใช้ระบบรีเลย์(Relay)ที่จำเป็นจะต้องเดินสายไฟ หรือ Hard-Wired ฉะนั้นเมื่อมีการเปลี่ยนกระบวนการผลิตแล้ว หรือลำดับการทำงานใหม่ จำเป็นต้องเดินสายไฟใหม่ทำให้เสียเวลาและเสียค่าใช้จ่ายมากขึ้น แต่เมื่อใช้พีแอลซีนั้นการเปลี่ยนกระบวนการผลิตนั้นสามารถทำได้โดยการเปลี่ยนโปรแกรมควบคุมภายในพีแอลซี และยังมีการกินไฟที่น้อยกว่า น่าเชื่อถือกว่าระบบรีเลย์

2.3 โครงสร้างพีแอลซี(PLC)

พีแอลซีเป็นอุปกรณ์คล้ายคอมพิวเตอร์ที่ใช้ในอุตสาหกรรมซึ่งประกอบไปด้วย หน่วยประมวลผลกลาง(CPU) , หน่วยความจำ(Memory) , หน่วยรับข้อมูลหรืออินพุท(Input) , หน่วยส่งข้อมูลหรือเอาท์พุท(Output) และหน่วยจ่ายไฟเลี้ยง(Power Supply) พีแอลซีโดยทั่วไปถ้าเป็นพีแอลซีขนาดเล็กจะรวมอยู่ในหน่วยอยู่ในพีแอลซีตัวเดียวแต่ถ้าเป็นพีแอลซีขนาดใหญ่แล้วอาจจะแยกหน่วยต่างๆออกจากกัน



ภาพที่ 2.4 โครงสร้างภายใน PLC

2.3.1 หน่วยประมวลผลกลาง (CPU)

ทำหน้าที่คำนวณและควบคุม ซึ่งเปรียบเสมือนสมองของพีแอลซี ภายในประกอบด้วยวงจรลอจิกหลายชนิดและมีไมโครโปรเซสเซอร์เบส (Micro Processor Based) ใช้แทนอุปกรณ์จำพวกรีเลย์ เคาน์เตอร์/ไทม์เมอร์ และซีเควนเซอร์ เพื่อให้ผู้ใช้สามารถออกแบบวงจรโดยใช้ Relay Ladder Diagram ได้ CPU จะยอมรับข้อมูลจากอุปกรณ์อินพุทต่างๆ จากนั้นจะทำการประมวลผลและเก็บข้อมูลโดยใช้โปรแกรมจากหน่วยความจำ หลังจากนั้นจะส่งส่งข้อมูลที่เหมาะสมและถูกต้องออกไปยังอุปกรณ์เอาท์พุท สมองของระบบ ภายใน CPU จะประกอบไปด้วยวงจร Logic Gate ชนิดต่างๆ หลายชนิด และมี Microprocessor-based ใช้สำหรับแทนอุปกรณ์จำพวกรีเลย์ (Relay) เคาน์เตอร์ (Counter) ไทเมอร์ (Timer) และซีเควนเซอร์ (Sequencers) เพื่อให้ผู้ใช้ได้ออกแบบใช้วงจรรีเลย์แลดเดอร์ลอจิก (Relay Ladder Logic) เข้าไปได้ CPU จะยอมรับ (Read) อินพุต เดต้า (Input Data) จากอุปกรณ์ให้สัญญาณ (Sensing Device) ต่างๆ จากนั้นจะปฏิบัติการและเก็บข้อมูลโดยใช้โปรแกรมจากหน่วยความจำ และส่งข้อมูลที่เหมาะสมถูกต้องไปยังอุปกรณ์ควบคุม (Control Device) แหล่งของกระแสไฟฟ้าตรง (DC Current) สำหรับใช้สร้างโวลต์ต่ำ (Low Level Voltage) ซึ่งใช้โดยโปรเซสเซอร์ (Processor) และไอโอ โมดูล (I/O Modules) และแหล่งจ่ายไฟนี้จะเก็บไว้ที่ CPU หรือแยกออกไปติดตั้งที่จุดอื่นก็ได้ขึ้นอยู่กับผลิตแต่ละรายการ

ประมวลผลของ CPU จากโปรแกรมทำได้โดยรับข้อมูลจากหน่วยอินพุทและเอาท์พุท และส่งข้อมูลสุดท้ายที่ได้จากการประมวลผลไปยังหน่วยเอาท์พุท เรียกว่า การสแกน (Scan) ซึ่งใช้เวลาจำนวนหนึ่งเรียกว่า เวลาสแกน (Scan Time) เวลาในการสแกนแต่ละรอบใช้เวลาประมาณ 1 ถึง 100 msec, (.0001-.01 วินาที) ทั้งนี้ขึ้นอยู่กับข้อมูลและความยาวของโปรแกรม หรือจำนวนอินพุท/เอาท์พุทหรือจำนวนอุปกรณ์ที่ต่อจากพีแอลซี เช่น เครื่องพิมพ์ จอภาพ เป็นต้น อุปกรณ์เหล่านี้จะทำให้เวลาในการสแกนยาวนขึ้น การเริ่มต้นการสแกนเริ่มจากรับคำสั่งของสภาวะของอุปกรณ์จากหน่วยอินพุทมาเก็บไว้ในหน่วยความจำ (Memory) เสร็จแล้วจะทำการปฏิบัติการตามโปรแกรมที่เขียนไว้ทีละคำสั่งจากหน่วยความจำนั้นจนสิ้นสุดแล้วส่งไปที่หน่วยเอาท์พุท ซึ่งการสแกนของพีแอลซีประกอบด้วย

I/O Scan คือ การบันทึกสภาวะข้อมูลของอุปกรณ์ที่อินพุท และให้อุปกรณ์เอาท์พุททำงาน

Program Scan คือ การให้โปรแกรมทำงานตามลำดับก่อนหลัง

2.3.2 หน่วยความจำ (Memory)

ทำหน้าที่เก็บรักษาโปรแกรมและข้อมูลที่ใช้ในการทำงาน โดยขนาดของหน่วยความจำจะถูกแบ่งออกเป็นบิตข้อมูล(Data Bit) ภายในหน่วยความจำ 1 บิต ก็จะมีค่าสภาวะทางลอจิก 0 หรือ 1แตกต่างกันแล้วแต่คำสั่ง ซึ่งพีแอลซี ประกอบด้วยหน่วยความจำสองชนิดคือ ROM และRAM ทำหน้าที่เก็บโปรแกรมของผู้ใช้และข้อมูลที่ใช้ในการปฏิบัติงานของพีแอลซี หน่วยความจำประเภทนี้จะมีแบตเตอรี่เล็กๆ ต่อไว้เพื่อใช้เป็นไฟเลี้ยงข้อมูลเมื่อเกิดไฟดับ การอ่านและการเขียนข้อมูลลงใน RAM ทำได้ง่ายมาก เพราะฉะนั้นจึ่งเหมากับงานในระยะทดลองเครื่องที่มีการเปลี่ยนแปลงแก้ไขโปรแกรมอยู่บ่อยๆ ROM ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของพีแอลซี ตามโปรแกรมของผู้ใช้หน่วยความจำแบบ ROM ยังสามารถแบ่งได้เป็น EPROM ซึ่งจะต้องใช้อุปกรณ์พิเศษในการเขียนและลบโปรแกรม เหมาะกับงานที่ไม่ต้องการเปลี่ยนแปลงโปรแกรม นอกจากนี้ยังมีแบบ EEPROM หน่วยความจำประเภทนี้ไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรม สามารถใช้งานได้เหมือนกับ RAM แต่ไม่ต้องใช้แบตเตอรี่สำรอง แต่ราคาจะแพงกว่าเนื่องจากรวมคุณสมบัติของ ROM และ RAM ไว้ด้วยกัน

2.3.3 หน่วยรับข้อมูลหรืออินพุท (Input)

ทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอกแล้วแปลงสัญญาณให้เป็นสัญญาณที่เหมาะสมแล้วส่งให้หน่วยประมวลผลต่อไป

2.3.4 หน่วยส่งข้อมูลหรือเอาท์พุท (Output)

ทำหน้าที่รับข้อมูลจากตัวประมวลผลแล้วส่งต่อข้อมูลไปควบคุมอุปกรณ์ภายนอก เช่น ควบคุมหลอดไฟ มอเตอร์ และวาล์ว เป็นต้น

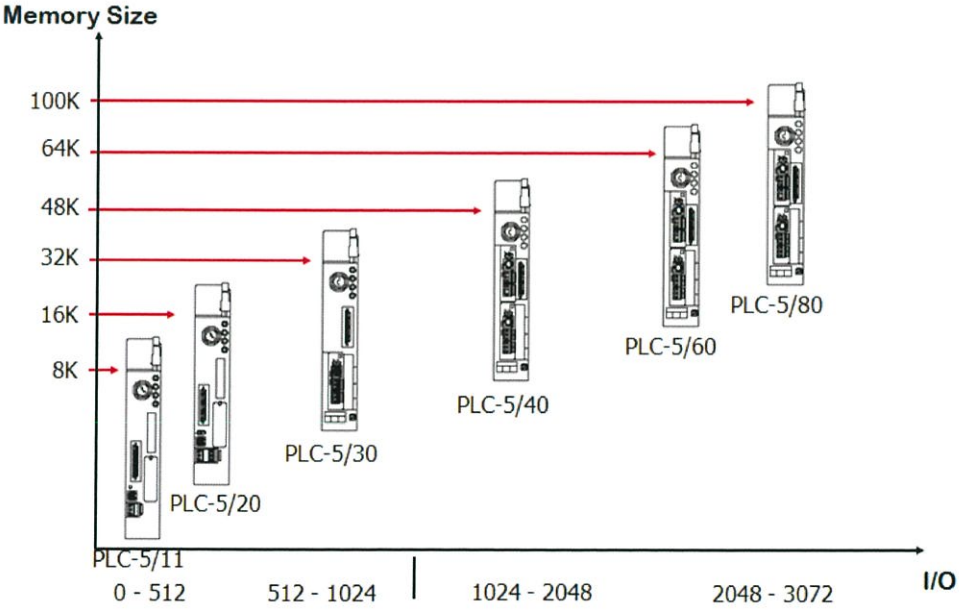
2.3.5 แหล่งจ่ายไฟ (Power Supply)

ทำหน้าที่จ่ายพลังงานและรักษาระดับแรงดันไฟฟ้ากระแสตรงให้กับ CPU Unit หน่วยความจำและหน่วยอินพุท/ เอาท์พุท

2.4 พีแอลซี PLC-5

2.4.1 ประวัติ PLC-5

PLC-5 นั้นเป็นพีแอลซีชนิดหนึ่งของอเลน–แบรดลีย์(Allen Bradley) ซึ่งคุมการผลิตโดยร็อคเวล ออโตเมชั่น(Rockwell Automation)ที่เป็นบริษัทผลิตอุปกรณ์ทางด้านอุตสาหกรรม พีแอลซี-5 นั้นเริ่มต้นการผลิตเมื่อปี ค.ศ.1986 และสิ้นสุดการผลิตเมื่อปี ค.ศ.1995 ในช่วงเวลา 9 ปี นั้นมีพีแอลซี-5 ออกมา 6 รุ่นหลักๆด้วยกัน คือ PLC-5/11 , PLC-5/20 , PLC-5/30 , PLC-5/40 , PLC-5/60 และ PLC-5/80 ซึ่งมีขนาดเมมโมรี่ที่ต่างกันดังภาพที่



ภาพที่ 2.5 หน่วยความ Processor ของ PLC-5

ในอดีตนั้นสิ่งที่ควรคำนึงถึงในการใช้พีแอลซีนั้นคือ หน่วยความจำ ที่ส่งผลต่อการใช้เมมโมรีภายใน อินพุท หรือ เอาท์พุท ในทางกลับพีแอลซีที่มีขนาดหน่วยความจำเยอะ จะมีราคาที่แพงตามขนาดหน่วยความจำที่มากขึ้น ดังนั้นควรเลือกขนาดหน่วยความจำจึงควรเลือกให้เหมาะสมกับกระบวนการผลิต ตัวอย่างเช่น การผลิตยางของบริษัท โมดูลที่ 4 เลือกใช้ PLC-5/60 ซึ่งจะมีขนาดดังภาพที่ 2.5 ที่มีความจุของ I/O สูงถึง 3072 จุด และมีหน่วยความจำอยู่ที่ 64K Word

Processor	Memory (Words)	Local Chassis	Remote Chassis (I/O Racks)	I/O Capacity	Communication
PLC-5/40	48K[3]	1 resident	60[2] (15 I/O racks)	• 2048[1] • 2048 inputs and 2048 outputs using 16- or 32-pt modules	• 4 channels (remote I/O scanner, adapter, DH+ link) • 1 RS-232, RS-422, RS-423 serial port
PLC-5/40L	48K[3]	1 resident up to 16 extended	60[2] (15 I/O racks)	• 2048[1] • 2048 inputs and 2048 outputs using 16- or 32-pt modules	• 2 channels (remote I/O scanner, adapter, DH+ link) • 1 RS-232, RS-422, RS-423 serial port • 1 channel extended local I/O scanner
PLC-5/40E	48K[3]	1 resident (16 rack addressing capability)	60 (15 I/O racks)	• 2048[1] • 2048 inputs and 2048 outputs using 16- or 32-pt modules	• 2 channels (remote I/O scanner, adapter, DH+ link) • 1 RS-232, RS-422, RS-423 serial port • 1 channel Ethernet
PLC-5/40C15	48K[3]	1 resident	60 15 I/O racks	• 2048[1] • 2048 inputs and 2048 outputs using 16- or 32-pt modules	• 2 channels (remote I/O scanner, adapter, DH+ link) • 1 RS-232, RS-422, RS-423 serial port • 1 channel ControlNet
PLC-5/60[3]	64K	1 resident	92[2] (23 I/O racks)	• 3072[1] • 3072 inputs and 3072 outputs using 16- or 32-pt modules	• 4 channels (remote I/O scanner, adapter, DH+ link) • 1 RS-232, RS-422, RS-423 serial port

[1] Any mix of I/O
[2] Maximum of 32 physical devices/channel
[3] Maximum of 57K words per program file and 32K words per data table file

ภาพที่ 2.6 ตารางคุณสมบัติส่วนหนึ่งของ PLC-5 แต่ละรุ่น

2.4.2 การ Addressing และ Tagging

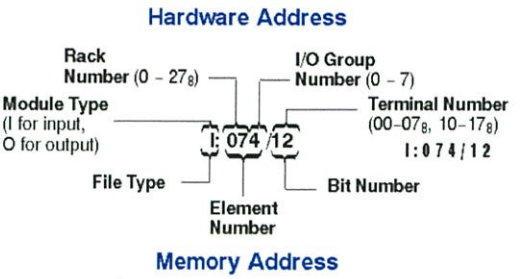
การเรียกใช้แท็ก(Tag) นั้นคือการอ้างอิงของตำแหน่งในหน้าการ์ด(Card) ของพีแอลซี หรือ การเรียกใช้แท็ก ของเมมโมรี่ภายใน การอ้างอิงเมมโมรี่ภายในนั้นจะอ้างจากตารางที่ทาง Allen Bradley กำหนดดังภาพที่ 2.6

File Type	File Type Identifier	File Number	Memory Used in Overhead for Each File (16-bit words)	Memory Used (16-bit words) per Word, Float Word, Character, or Structure
Output Image	O	0	2	1/word
Input Image	I	1	2	1/word
Status	S	2	2	1/word
Bit (binary)	B	3	2	1/word
Timer	T	4[1]	2	3/structure
Counter	C	5[1]	2	3/structure
Control	R	6[1]	2	3/structure
Integer	N	7[1]	2	1/word
Floating-Point	F	8[1]	2	2/float word
ASCII	A	3-999	2	1/2 per character
BCD	D	3-999	2	1/word
Undefined	--	9-999	2	0

[1] This is the default file number. For this file type, you can assign any file number from 3 thru 999.

ภาพที่ 2.7 การอ้างอิงตำแหน่งของ PLC-5

การ Addressing ของ PLC-5 นั้น มีลักษณะเฉพาะตัว เช่น I:074/12 คือการเรียกอินพุทแร็ค(Rack) 07 กรุ๊ป(Group) 04 บิท(Bit) ที่ 12 โดยจะใช้เป็น 8 บิท หรือเลขฐาน 8 ดังภาพ นอกจากนี้เงื่อนไขในการเรียกแร็ค หรือ กรุ๊ป ดังนี้ ในหนึ่งกรุ๊ป จะมี 16 จุด(Point)เสมอ ในหนึ่งแร็คจะมี 8 กรุ๊ปเสมอ



ภาพที่ 2.8 รูปแบบของ Addressing

2.4.3 การใช้ Addressing mode

อย่างที่กล่าวในข้างต้น PLC-5 นั้นหน่วยความจำมีความสำคัญมากดังนั้นจึงจำเป็นต้องจัดการกับ ราง (Chassis) และ สล็อท(Slot) ในฮาร์ดแวร์เพื่อให้เกิดประสิทธภาพสูงสุด มีการแบ่งอยู่สามประเภท

Two-Slot Addressing คือ 2 สล็อ ต่อ 1 กรุ๊ป

One-Slot Addressing คือ 1 สล็อทต่อ 1 กรุ๊ป

Half-Slot Addressing คือ ครึ่งสล็อทต่อ 1 กรุ๊ป

ซึ่งการใช้งานจะมีผลต่อเมมโมรีภายในตัวอย่างเช่น ถ้ามี I/O แบบ 8 จุดโดยใช้ 1 สล็อทเท่ากับว่าในเมมโมรี่ 1 กรุ๊ปแต่ใช้แต่ 8 จุด ดั้งนั้น อีก 8 จุดที่เหลือจะเป็นการจองพื้นที่โดยเสียประโยชน์ทันที ดังภาพที่ 2.9

If Using this Chassis Size	2-slot Addressing, Results In	1-slot Addressing, Results In	1/2-slot Addressing, Results In
4-slot	1/4 rack	1/2 rack	1 rack
8-slot	1/2 rack	1 rack	2 racks
12-slot	3/4 rack	1-1/2 racks	3 racks
16-slot	1 rack	2 racks	4 racks

ภาพที่ 2.9 การใช้สล็อท(Slot) และกรุ๊ป(Group) เพื่อหาจำนวนแร็ค(rack)

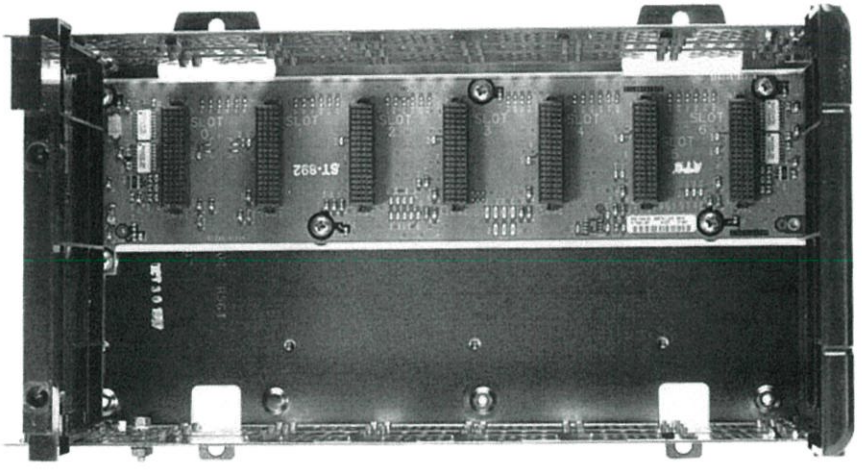
2.5 พีแอลซี ControlLogix

2.5.1 ประวัติ ControlLogix

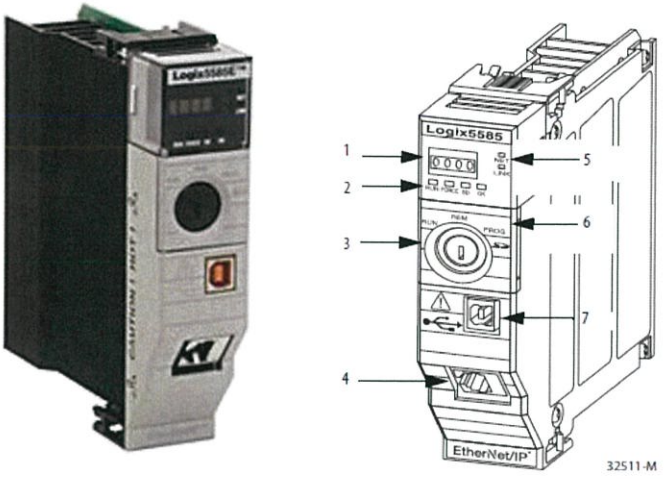
ControlLogix นั้นเป็นชื่อทางการค้าของ Allen Bradley ซึ่งอุปกรณ์ควบคุมระบบขนาดใหญ่ มีประสิทธิภาพสูง ง่ายต่อการใช้งานผสานความสามารถในการจัดการและควบคุมข้อมูล นอกจากนี้ยังมีความน่าเชื่อถือสูง โดย ControlLogix ถูกปล่อยครั้งแรกในปี 1997 มีพื้นฐานมากจาก SLC ซึ่งเป็นผลิตภัณฑ์อีกรุ่นหนึ่งที่ถูกพัฒนามาก่อน และ ControlLogix มีความเร็วสูงกว่า เมมโมรี่มากกว่าทุกผลิตภัณฑ์ที่ผลิตมา เป็นแพลตฟอร์มแรกที่นำการใช้แทกบนพื้นฐานแอดแดรส(Tag Database) มีการใช้การสื่อสาร(Communication) ที่หลากหลายยิ่งขึ้นทำให้ใช้การ์ดแยกเฉพาะ Ethernet , DeviceNet และ ControlNet ต่างจากในอดีตที่ใช้หลายๆพอร์ท(Port) ในการ์ด CPU มีการประยุกต์ใช้กับ Motion Control ได้ซับซ้อนยิ่งขึ้น

2.5.2 CPU สำหรับ ControlLogix

สำหรับ CPU ของจะเป็น ControlLogix5580 ทำงานบนราง(Chassis) เฉพาะ ดังภาพที่ 2.10 การป้อนโปรแกรมจะใช้ 5000 Automation Engineering & Design Environment™ สามารถรองรับ I/O Point ได้ถึง ดิจิตอล 128,000 หรือ แอนาล็อกเซอร์โว 4,000 มีพอร์ท Ethernet และ USB ในตัว



ภาพที่ 2.10 ราง(Chassis) ของ ControlLogix



ภาพที่ 2.11 CPU ของ ControlLogix5580

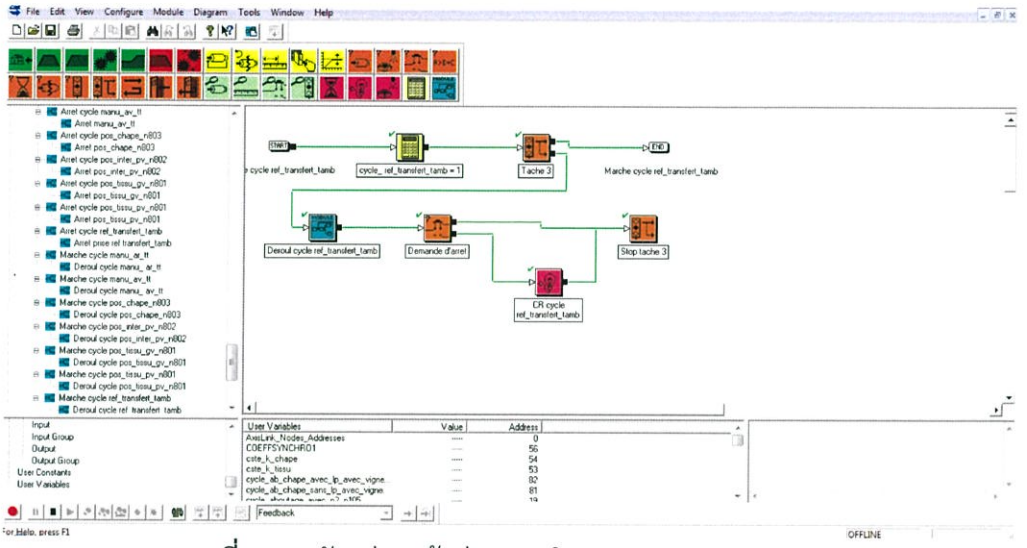
ส่วนประกอบของ CPU ControlLogix5580 จะมีดังนี้ 4-character display , Status Indicators , Mode switch (Remote, Run, Program) , Ethernet Port , Ethernet Status , IndicatorsSD card slot and Reset Button are behind the door และ USB Port ตามลำดับเลขในภาพภาพที่ 2.10 ทางด้านขวามือ

2.5.3 I/O Card สำหรับ ControlLogix

สำหรับ I/O Card นั้นจะใช้เป็น 1756 ซีรี่ย์ ซึ่งจะสามารถใช้กับ ราง(Chassis) ControlLogix พอดี I/O Card แบ่งประเภทการ์ดออก นอกจากนี้ยังสามารถเทอร์มินัลของทางคอนโทรลโลจิก(Removable Terminal Block) กับการ์ด(Card) ออกจากกันได้

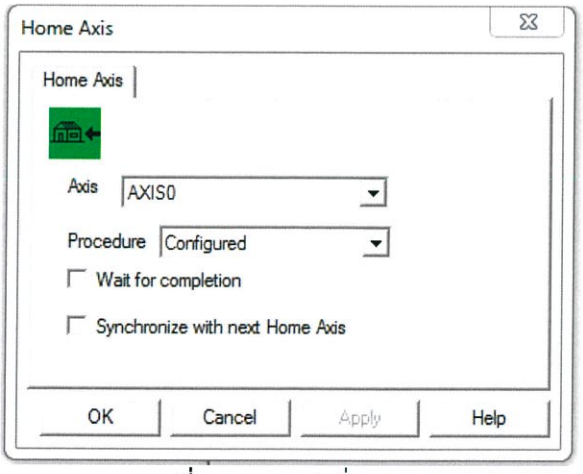
2.6 จีเอ็มแอล คอมมานเดอร์ (GML Commander)

เป็นโปรแกรมที่ถูกพัฒนาเพื่อเขียนเงื่อนไขให้กับ IMC-S ใช้ร่วมกับ RsLogix5 ที่ใช้เขียนโปรแกรมใน PLC-5 ซึ่งในหัวข้อนี้จะยกคำสั่งที่เป็นพื้นฐานที่จำเป็นหรือคำสั่งที่ใช้ในการใช้งานจริงโดยหน้าตาของโปรแกรมจะเป็นลักษณะบล็อกคำสั่งเชื่อมต่อด้วยเส้นเป็นเงื่อนไขตัวอย่างเช่นภาพที่ 2.12



ภาพที่ 2.12 ตัวอย่างหน้าต่างการทำงาน GML Commander

2.6.1 ชุดคำสั่ง HOME



ภาพที่ 2.13 ชุดคำสั่ง HOME

HOME เป็นปุ่มการทำงานเพื่อสั่งเงื่อนไขการทำงานกลับมาที่ตำแหน่งโฮมหรือตำแหน่งเริ่มต้นของซึ่งขึ้นกับการตั้งค่าให้เริ่มต้นของโฮม เงื่อนไขของปุ่มโฮมนั้นไม่สามารถดำเนินงานในแกนจินตภาพได้ แต่ทำในแกนที่จำลองขึ้นนั้นสามารถทำงานได้ใน Oive เนื่องจากเป็นค่าเริ่มต้น ไม่

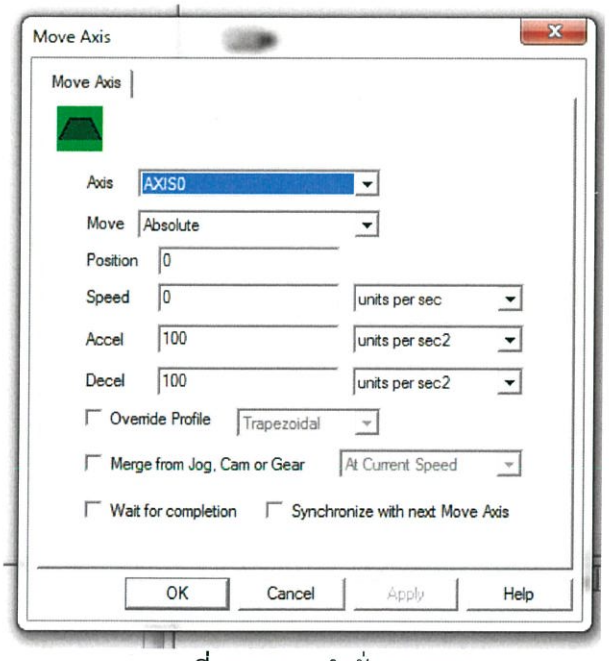
ว่าจะเลือก Oive หรือ Configured ในหน้าเมนูโฮม ผลสุดท้ายของการสั่งจะเป็นผลลัพธ์เดียวกันในเมนู Procedure จะให้ปรับ

1. Configured Home เป็นการโฮมด้วยอ้างอิงจากค่าที่ตั้งค่าไว้แล้ว

2. Oive Home เป็นการหาตำแหน่ง (re-Define) โดยอ้างอิงจากตำแหน่งปัจจุบันของแกนแล้วกลับไปเริ่มตำแหน่งโฮมจากการหาตำแหน่งโดยใช้ Encoder Markerในการคำณวน Wait for Completion คำสั่งจะทำการหยุดเงื่อนไขทั้งหมดไปที่โมชั่นคอนโทรลเลอร์สั่งหยุดโปรแกรมจนกว่าจะมีการเลือก ลำดับการทำงานของโฮม (Homing Sequence)เสร็จสิ้น จากนั้นจึงทำงานในบล็อกต่อไป ใน Procedure นั้นคือการเลือกรูปแบบ

3. Synchronized with next Home Axis เมื่อมีการซิงโครไนส์โฮมด้วย Home Axis อีกตัวโฮมตัวแรกจะทำการทำตามเงื่อนไขของโฮมตัวถัดไปทันที โดยโปรแกรมนั้นเมื่อตั้งค่าการใช้ Synchronized Home แล้วโปรแกรมจะบังคับให้มีโฮมตัวที่สอง

2.6.2 ชุดคำสั่ง MOVE



ภาพที่ 2.14 ชุดคำสั่ง MOVE

MOVE เป็นคำสั่งการเคลื่อนย้ายของแกนไปที่ตำแหน่งสัมบูรณ์ใดๆ (Absolute Position) หรือ ตำแหน่งระยะทางที่เพิ่มขึ้น (Incremental Distance) ที่ความเร่ง (Accel), ความหน่วง(Decel), ตำแหน่ง (Position) หรือความเร็วใด(Speed)ๆ นอกจากนี้ยังสามารถเพิ่มเงื่อนไขได้อีกในคำสั่ง Move มี 6 รูปแบบด้วยกันคือ

1. Absolute Move

2. Incremental Move

3. Phase Shift

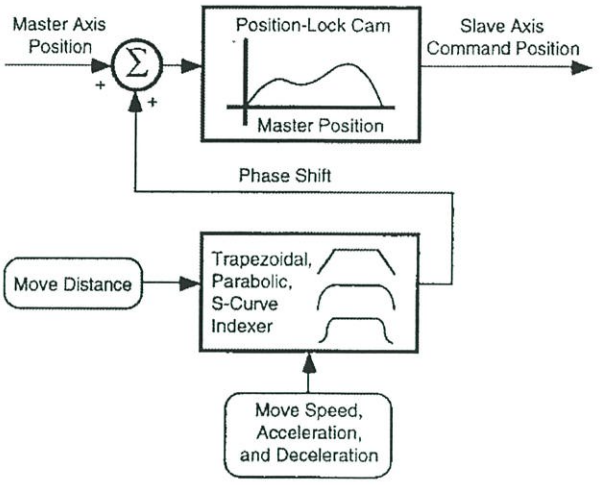
4. Rotary Shortest Path Moves

5. Rotary Positive Moves

6. Rotary Negative Moves

Absolute Move เป็นตัวเลือกของเงื่อนไขการที่ซึ่งเป็นลักษณะเคลื่อนที่แบบสัมบูรณ์สามารถระบุตำแหน่ง ความเร่ง ความเร็ว หรือ ความหน่วง โดยตรงซึ่งการเปลี่ยนจุดปลายของการเคลื่อนที่ (Endpoint) สามารถทำได้ในขณะที่ยังเคลื่อนที่อยู่ โดยเลือกประเภทของการเคลื่อนจากสามรูปแบบ Trapezoidal, S-curve หรือ Parabolic ยกเว้นเมื่อกำลังมีความเร่ง หรือความหน่วง จะไม่สามารถใช้ S-curve หรือ Parabolic ได้ การทำงาน Absolute Move บนแกน Rotary การทำงานแบบ Rotary หรือการทำงานเชิงมุมนั้น คำสั่งคล้ายกับ เชิงเส้น ยกเว้นเมื่อตำแหน่งของแกนนั้นเกินกว่า Unwind parameter ตำแหน่งของแกนจะถูกแก้ ดังนั้น เมื่อค่าไม่เกิน Unwind parameter ในทางเดียวกันค่าตำแหน่งแกนจะไม่ต่ำกว่าไปกว่าศูนย์ เช่น -90° ถ้าเกินกว่าค่า Unwind parameter จะถูกแก้เป็น 270°

Incremental Move เป็นตัวเลือกของเงื่อนไขการที่ซึ่งเป็นลักษณะเคลื่อนที่แบบสัมบูรณ์สามารถระบุตำแหน่ง ความเร่ง ความเร็ว หรือ ความหน่วง โดยตรง ซึ่งการเปลี่ยนจุดปลายในขณะที่กำลังทำงาน แต่ไม่ใช่ในขณะกำลังเร่ง, กำลังหน่วง, หรือ ทำเงื่อนไข S-curve หรือ Parabolic โดยการเพิ่มบล็อก Move อีกหนึ่งบล็อกและเพิ่ม Incremental Move เข้าไป โปรแกรมจะนำระยะทางมารวมกันของทั้งสองบล็อก การทำงาน Incremental Move with Gearing สามารถใช้ Gear Axes Block ในภาพที่ ในขณะที่แกน Slave กำลังทำงานนั้นสามารถทำได้ การทำงานของเกียร์นั้นมีความซับซ้อนสูงการใช้แต่ Incremental Move จะช่วยในการควบคุมแบบ Phase advance/Retard ได้ง่ายขึ้น การทำงาน Incremental Move บนแกน Rotary หรือการทำงานเชิงมุมนั้น คำสั่งคล้ายกับ เชิงเส้น ยกเว้นเมื่อตำแหน่งของแกนนั้นเกินกว่า Unwind parameter ตำแหน่งของแกนจะถูกแก้ ดังนั้น เมื่อค่าไม่เกิน Unwind parameter ในทางเดียวกันค่าตำแหน่งบวกของแกนจะไม่เกินกว่าค่า Unwind parameter และไม่น้อยกว่าศูนย์ เช่น เมื่อมีการใช้ค่ามากกว่า Unwind parameter แกนจะหมุนเกินรอบก่อนที่จะหยุด



ภาพที่ 2.15 ลักษณะการทำงานของ Phases Shift Move

Phases Shift Move สามารถใช้ในการเคลื่อนที่แบบ Phases Shift เพื่อ Shift รูปแบบการหมุนของลูกเบี้ยวให้สัมพันธ์กับแกนหลัก (Master Axis) โดยใช้การ Shift ของเวลา นอกจากนี่ยังสามารถตั้งค่าการทำงานร่วมกันของแกนหลัก และแกนรอง (Synchronization) จากภาพที่ 2.15 จะเห็นการทำงานของ Phases Shift Move ที่จุด Sum ของฟังก์ชันกับแกนหลัก

Rotary Shortest Path Moves การทำงานนี้สำหรับ absolute position ในแกน rotary หรือเชิงมุมเท่านั้นซึ่งจะต้องกำหนด ความเร่ง หรือ ความหน่วง จากนั้นเงื่อนไขนี้จะทำการหาตำแหน่งหรือระยะทางที่ใกลที่สุด โดยที่ค่าบวกจะต้องน้อยกว่าค่า Unwind Value ยกตัวอย่าง เช่น เมื่อกำหนดค่า ต้องการไปที่ตำแหน่ง 225° จากตำแหน่ง 30° ดังนั้น แกนจะหมุนไปทางลบเข้าที่ 225°

1. Rotary Positive Moves คำสั่งในการกำหนดตำแหน่งไปทางบวกซึ่งใช้ใน Rotary

2. Rotary Negative Moves คำสั่งในการกำหนดตำแหน่งไปทางบวกซึ่งใช้ใน Rotary

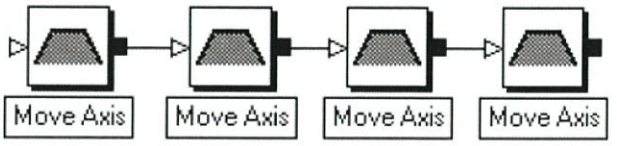
3. Override Profile คำสั่งในการเคลื่อนที่ด้วยความเร็วเพิ่มขึ้น แบบวิธีต่างๆ Trapezoidal, S-curve หรือ Parabolic

Merged form JOG ,CAM or GEAR คำสั่งนี้ใช้สำหรับ การรวมการเคลื่อนที่ สามารใช้ได้ทั้ง absolute or incremental moves และสามารถกำหนด ความเร่ง, ความหน่วง, หรือความเร็ว ได้

4. Current Speed แสดงความเร็วปัจจุบันของแกนนั้น

Programmed Speed กำหนดความเร็วของการหมุน โดยจะอิงตามค่าเริ่มต้นของแกนถ้าไม่ได้กำหนดไว้

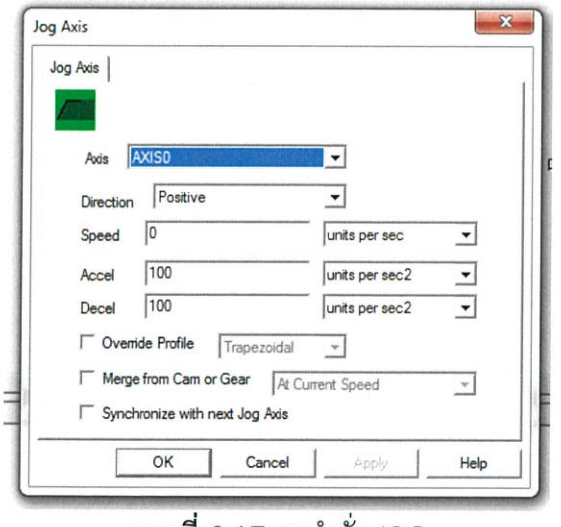
5. Wait for Completion เมื่อใช้คำสั่งนี้แล้วคอนโทรลเลอร์จะหยุดทำงานของบล็อกอื่นจนกว่าจะเคลื่อนเสร็จแล้วจึงจะเริ่มทำงาน ถ้ามีการทำงานซ้อนกัน (Multitasking)



ภาพที่ 2.16 การทำงานของ Synchronizing Moves

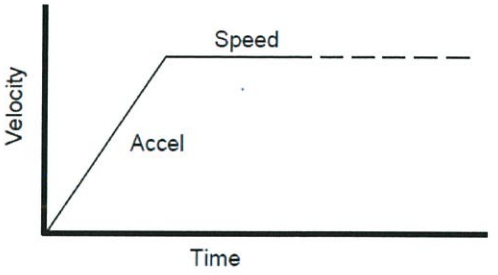
6. Synchronizing with next Moves เมื่อมีการซิงโครไนส์ Move ด้วยอีกตัวโฮมตัวแรกจะทำการทำตามเงื่อนไขของ Move ตัวถัดไปทันที โดยโปรแกรมนั้นเมื่อตั้งค่าการใช้ Synchronized Home แล้วโปรแกรมจะบังคับให้มี Move ตัวที่สอง

2.6.3 ชุดคำสั่ง JOG



ภาพที่ 2.17 ชุดคำสั่ง JOG

JOG เป็นรูปแบบคำสั่งที่ให้แกนใดเคลื่อนที่อย่างต่อเนื่องใน ทิศทาง (Direction), ความเร็ว (Speed) ความหน่วง(Decel) หรือ ความเร่ง(Accel)ใดๆที่กำหนดคำสั่งค่าเริ่มต้นจะใช้เป็นแบบ Trapezoidal

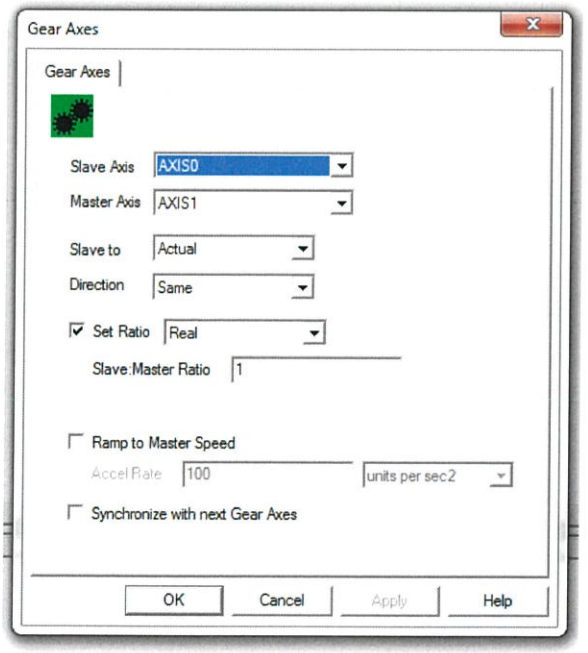


ภาพที่ 2.18 รูปแบบการเคลื่อนที่ของ Trapezoidal jog

สามารถปรับโปรไฟล์การเคลื่อนที่ได้ที่ Override Profile จะสามารถเลือกประเภทของการเคลื่อนที่ได้ 3 ประเภท Trapezoidal , S-Curve และ Parabolic

Synchronizing with next JOG เมื่อมีการซิงโครไนส์ JOG ด้วยอีกตัวโฮมตัวแรกจะทำการทำตามเงื่อนไขของ Move ตัวถัดไปทันที โดยโปรแกรมนั้นเมื่อตั้งค่าการใช้ Synchronized JOG แล้วโปรแกรมจะบังคับให้มี JOG ตัวที่สอง

2.6.4 ชุดคำสั่ง GEAR AXES



ภาพที่ 2.19 ชุดคำสั่ง Gear Axes

Gear Axes เป็นรูปแบบคำสั่งที่ควบคุมระหว่างสองแกนโดยที่มีอัตราส่วนต่อกันอีเล็คทรอนิกส์เกียริ่งจะทำหน้าที่ทำงานร่วมกัน (synchronized) ระหว่างแกนจริง หรือตำแหน่งที่สั่งไปให้เป็นอัตราส่วนต่อกัน โดยแกน Master Axis สามารถตั้งค่าทิศทาง(Direction) เป็น ทิศทางเดียว (Same),ทิศทางตรงกันข้าม(Opposite),ทิศคงที่เสมอ(Unchanged) และ ทิศย้อนกลับคงที่เสมอ

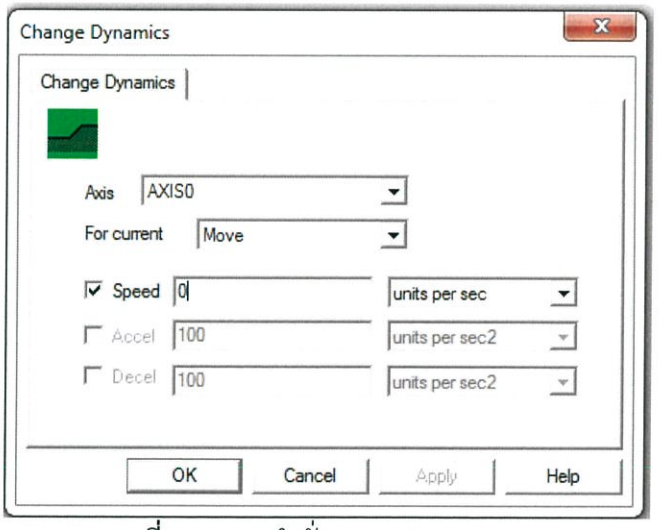
(Reverse) การตั้งค่า Slave to ปรับให้แกน Slave ใช้คำสั่งจากค่าจริง(Actual)ที่คำณวนจาก Encoder หรือ คำสั่ง(Command) ที่อ้างอิงจากค่าที่เราตั้งไว้ในโปรแกรม

การ Set Ratio คือการตั้งค่าอัตราส่วนระหว่าง Master และ Slave Axis จะมี 2 ลักษณะด้วยกัน Real และ Fraction แบบ Real นั้นจะใส่ค่าเป็นอัตราส่วน แต่ใน Fraction จะใส่เป็นค่าการหมุนที่รับจาก Feedback Count ของ Slave Axis และ Master Axis โดย Slave Axis 99,999 และ Master Count ได้ 9-digit ตั้งแต่ 1-999,999,999

Ramp to Master Speed คือการไต่ระดับความเร็วของ Master Axis ใช้การกำหนดความเร่ง (Accel) ของการเคลื่อนที่

Synchronizing with next GEAR เมื่อมีการซิงโครไนส์ GEAR ด้วยอีกตัวโฮมตัวแรกจะทำการทำตามเงื่อนไขของ Move ตัวถัดไปทันที โดยโปรแกรมนั้นเมื่อตั้งค่าการใช้ Synchronized GEAR แล้วโปรแกรมจะบังคับให้มี GEAR ตัวที่สอง

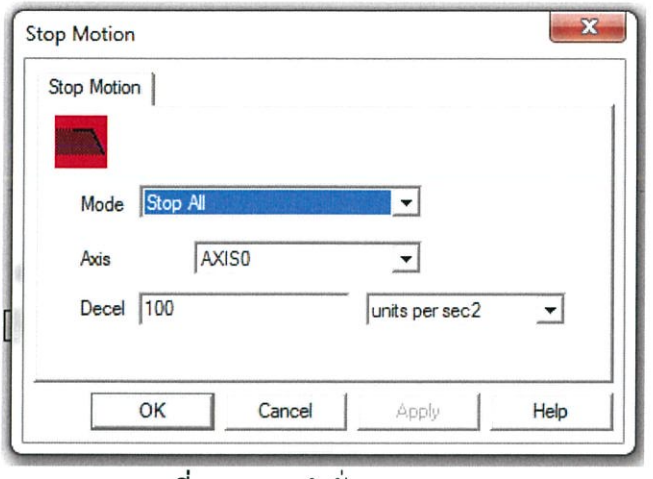
2.6.5 ชุดคำสั่ง CHANGE DYNAMICS



ภาพที่ 2.20 ชุดคำสั่ง Change Dynamics

ฟังก์ชัน CHANGE DYNAMICS เป็นการเปลี่ยนแปลงการเคลื่อนที่ในขณะ Servo Motor ยังทำงานอยู่สามมารถปรับเปลี่ยนการทำงานได้ 2 ลักษณะใน For Current คือ MOVE และ JOG ถ้าเป็นเงื่อนไข MOVE จะปรับไปที่ ความเร็ว(Speed)ใดๆได้ ที่เงื่อนไข JOG จะปรับความเร็ว (Speed)ใดๆได้โดยสามารถระบุ ความเร่ง(Accel) และ ความหน่วง(Decel)ได้

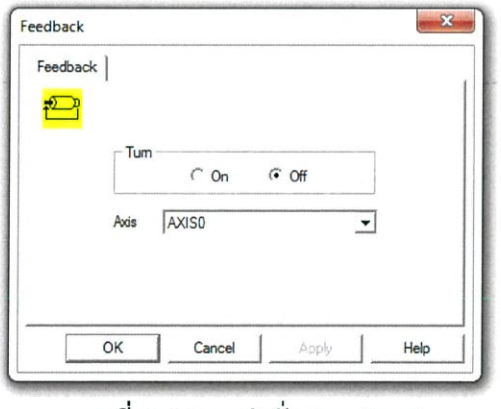
2.6.6 ชุดคำสั่ง STOP MOTION



ภาพที่ 2.21 ชุดคำสั่ง Stop Motion

เป็นฟังก์ชันสำหรับการหยุดแกน ซึ่งกำหนดการหน่วง(Decel) ได้ เงื่อนไขการหยุดจะปรับใน Mode มีดังนี้ Kill control , Stop All , Stop Gearing , Stop Homing , Stop Jog , Stop Move , Stop PCAM , Stop TCAM และ InterPolation

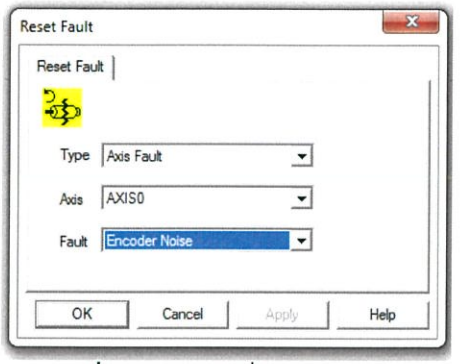
2.6.7 ชุดคำสั่ง FEEDBACK



ภาพที่ 2.22 ชุดคำสั่ง Feedback

ฟังก์ชันการรับ Feedback จาก Counter ของแกนใดๆสามารถตั้งค่าให้ on หรือ off ได้เมื่อถึงฟังก์ชั่นบล็อกนี้

2.6.8 ชุดคำสั่ง RESET FAULT

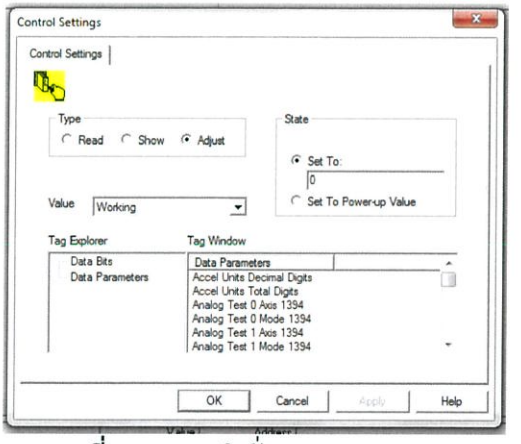


ภาพที่ 2.23 ชุดคำสั่ง Reset Fault

ฟังก์ชันการรีเซทค่าของผิดพลาด(Fault)ที่เกิดจาก Motion Control หรือMotor Driver สามารถเลือกประเภท(Type) 3 ประเภท Axis Fault รีเซทเฉพาะแกนนั้น , All Faults รีเซททุกแกนที่มีการผิดพลาด , Reset 1394 รีเซทคอนโทรลเลอร์ 1394

เลือกการผิดพลาด(Fault) ที่ต้องการจะรีเซท Encoder Noise , Software Overtravel , Hardware Overtravel ,Position Limit Exceeded , Motor/Thermal Fault , Resolver Loss และ Encoder Loss

2.6.9 ชุดคำสั่ง CONTROL SETTING



ภาพที่ 2.24 ชุดคำสั่ง Control Setting

ชุดคำสั่งนี้จะเป็นการนำค่าพารามิเตอร์ใดขึ้นมาใช้ เลือกค่าที่ Tag Window ตามหัวข้อต่าง ใน Tag Explorer โดยเลือกชนิด(Type)ของการทำงานดังนี้

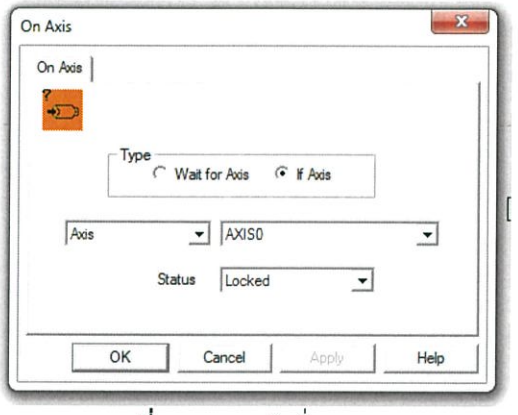
-Show คือ การอ่านค่าปัจจุบันของพารามิเตอร์ขึ้นมาแสดงที่ Interface Port

-Read คือ นำค่าพารามิเตอร์ที่เลือกขึ้นมาเก็บที่ User Variable ที่เลือก

-Adjust คือการนำค่าพารามิเตอร์ที่เลือกมาแก้ไข

นอกจากนี้ค่าที่เลือก(Value) สามารถเลือกได้ เป็น Working ซึ่งเป็นค่าที่ถูก Motion Controller ใช้ในขณะทำงานอยู่หรือ Power-up ซึ่งเป็นค่าเริ่มต้นที่ใส่ไว้ใน Configure Control Options และ Configure Axis ตั้งแต่เริ่มต้น

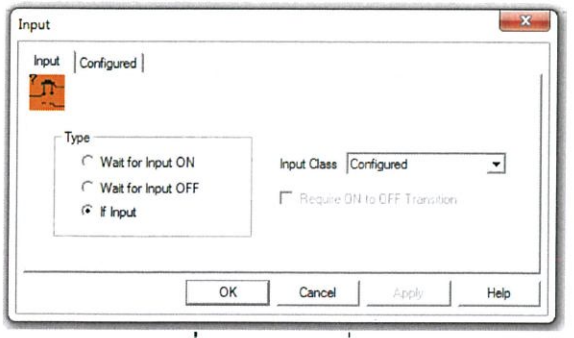
2.6.10 ชุดคำสั่ง ON AXIS



ภาพที่ 2.25 ชุดคำสั่ง On Axis

เป็นชุดคำสั่งสร้างเงื่อนไขของแกนใดๆ จะเลือกชนิด(Type) ของการ ON ว่าเป็นแบบใด Wait For Axis และ If Axis ในแบบแรกนั้นจะหยุดการทำงานของโปรแกรมจนกระทั่ง Axis/Interpolator เงื่อนไข(Status)เป็นจริง ในแบบถัดมา ถ้า Axis/Interpolator เป็นจริงตามเงื่อนไข(Status) จะวิ่งไปทางเส้นบน เงื่อนไขเป็นเท็จจะวิ่งไปทางเส้นล่าง จะมีเงื่อนไข(Status) ดังนี้ Locked , Jogging Done , Moving Done , Homing Done , Output Limited Feedback On , Accelerated , Decelerated , Gearing Opposite , Accelerating , Decelerating , Gearing On และ Gearing Off

2.6.11 ชุดคำสั่ง INPUT



ภาพที่ 2.26 ชุดคำสั่ง Input

ชุดคำสั่ง INPUT ทำไว้เพื่อสร้างเงื่อนไขโดยมี Input เป็นปัจจัยมี 3 หัวข้อ

1. Wait for INPUT ON คือ รอจนกว่าอินพุทจะเป็นจริง

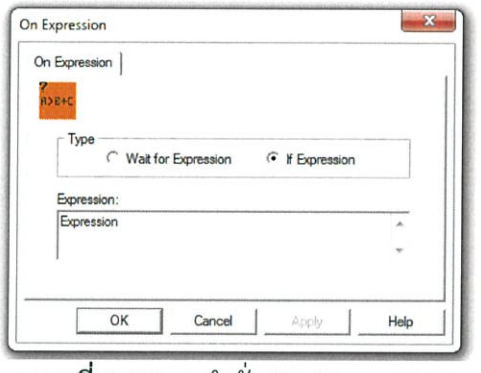
2. Wait for INPUT OFF คือ รอจนกว่าอินพุทจะเป็นเท็จ

3. If INPUT คือเงื่อนไขของการทำงานถ้าเป็นจริงจะวิ่งทางข้างบน และ ถ้าเป็นเท็จจะวิ่งไปทางข้างล่างจะไม่รออินพุทเหมือน 2 กรณีบน

เงื่อนไขของอินพุทจะเลือกได้ใน Input Class มีเงื่อนไข 3 ปะเภท Dedicated , Configured และ Miscellaneous ขึ้นกับเงื่อนไขที่ต้องการดังภาพที่ 2.27 จากซ้ายไปขวาตามลำดับ

ภาพที่ 2.27 Dedicated , Configured , Miscellaneous

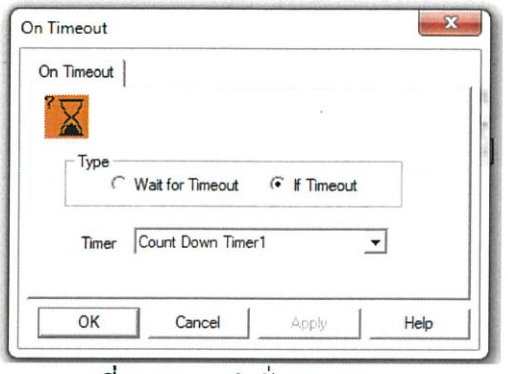
2.6.12 ชุดคำสั่ง ON EXPRESSION



ภาพที่ 2.28 ชุดคำสั่ง On Expression

ชุดคำสั่ง On Expression เป็นชุดคำสั่งเงื่อนไขใดๆ 2 ประเภท คือ Wait for Expression และ If Expression ในแบบแรกเป็นเงื่อนไขที่จะรอที่บล็อกนี้จนกว่าจะเป็นจริงเท่านั้น ในแบบที่สองนั้นจะเป็นเงื่อนไขแบบ ถ้า...แล้ว เงื่อนไขของการทำงานถ้าเป็นจริงจะวิ่งทางข้างบน และถ้าเป็นเท็จจะวิ่งไปทางข้างล่างจะไม่รอ Expression เป็นเงื่อนไขที่กำหนดขึ้นโดยผู้ใช้

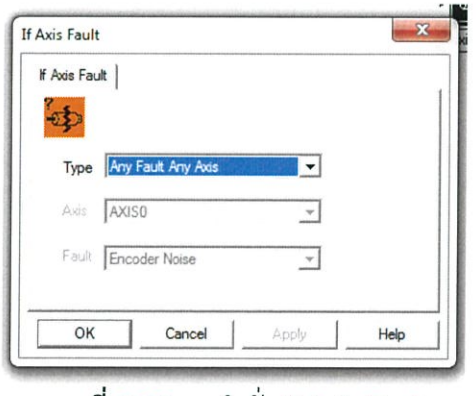
2.6.13 ชุดคำสั่ง ON TIMEOUT



ภาพที่ 2.29 ชุดคำสั่ง On Timeout

ชุดคำสั่ง On Timeout Wait มี 2 ประเภท Wait For Timeout รอจนกว่า Timer จะหมดเวลาจึงจะทำงานต่อ และอีกประเภท ถ้า Timer หมดเวลาวิ่งทางข้างบน แต่ถ้า Timer ยังทำงานอยู่จะวิ่งลงเส้นล่าง

2.6.14 ชุดคำสั่ง IF AXIS FAULT



ภาพที่ 2.30 ชุดคำสั่ง If Axis Fault

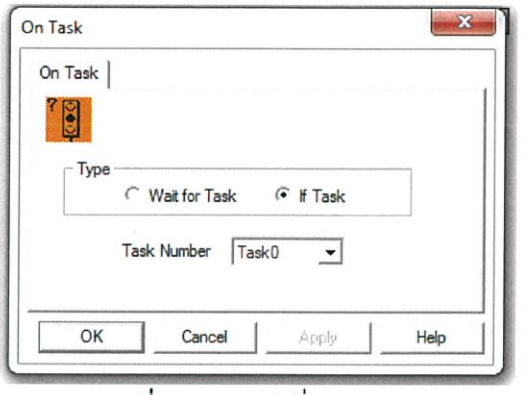
ชุดคำสั่ง If Axis Fault มี 3 ประเภท Any Fault Any Axis , Any Fault Specific Axis และ Specific Fault

-Any Fault Any Axis จะเป็นเงื่อนไขถ้ามีการผิดพลาด (Fault)ใดๆวิ่งทางข้างบน และ ถ้าไม่มีการผิดพลาด(Fault)จะวิ่งไปทางข้างล่าง

-Any Fault Specific Axis คล้ายกับเงื่อนข้างบนเพียงแต่สามารถกำหนดแกนได้

-Specific Fault คล้าบกับเงื่อนไข Any Fault Any Axis สามารถกำหนดแกน และกำหนด เงื่อนไขที่ทำให้เกิดความผิดพลาด(Fault) ได้ Encoder Noise , Software Overtravel , Hardware Overtravel ,Position Limit Exceeded , Drive Fault , Resolver Loss และ Encoder Loss

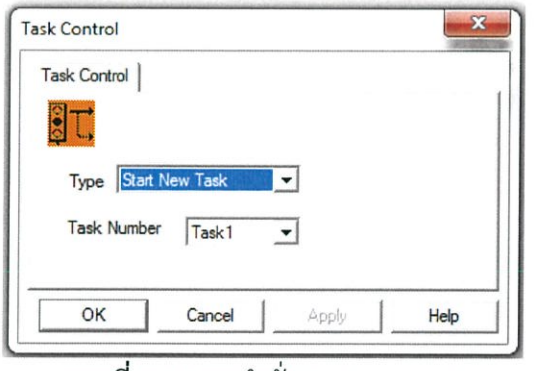
2.6.15 ชุดคำสั่ง ON TASK



ภาพที่ 2.31 ชุดคำสั่ง On Task

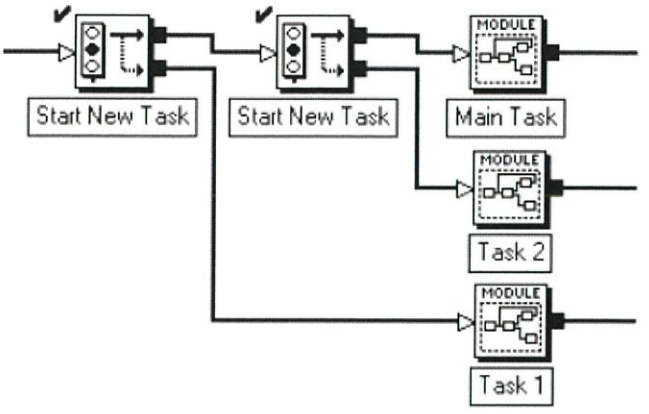
เป็นชุดคำสั่งสร้างเงื่อนไขของงาน(Task) จะเลือกงาน(Task number) ของการทำงาน ว่าเป็นแบบใด Wait For Task และ If Task ในแบบแรกนั้น จะหยุดการทำงานของโปรแกรมจะทำงานต่อเมื่องาน(Task)ที่เลือกไว้ เป็นจริง ในแบบถัดมา จะคล้ายกับแบบแรกแต่เมื่อเงื่อนไขเป็นเท็จจะทำงานตามที่ได้โปรแกรมไว้

2.6.16 ชุดคำสั่ง TASK CONTROL



ภาพที่ 2.32 ชุดคำสั่ง Task Control

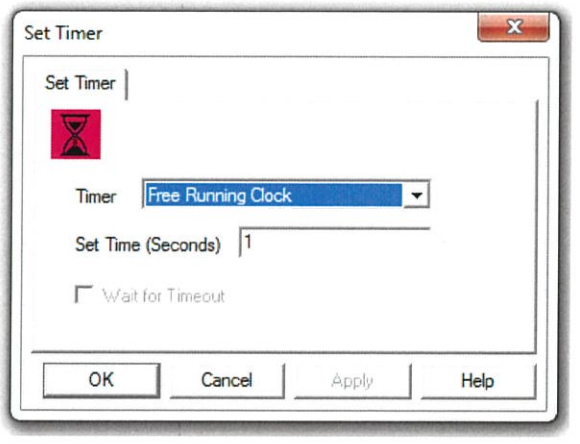
ชุดคำสั่ง Task Control เป็นชุดสำหรับสร้างงานที่ซับซ้อนกว่า ON TASK สามารถเรียกได้ว่าเป็นการทำงานหลายงาน (MultiTasking) ตัวอย่างภาพที่ 2.32 ทั้งสามารถ Start New Task , Stop Current Task , Stop Other Task , Resume Task , Stop Dispatcher และ Restart Dispatcher



ภาพที่ 2.33 การทำงานของ Multitasking

ในภาพที่ 2.33 เป็นตัวอย่างการทำ Multitasking หลายๆ Task พร้อมกันโดย Start New Task

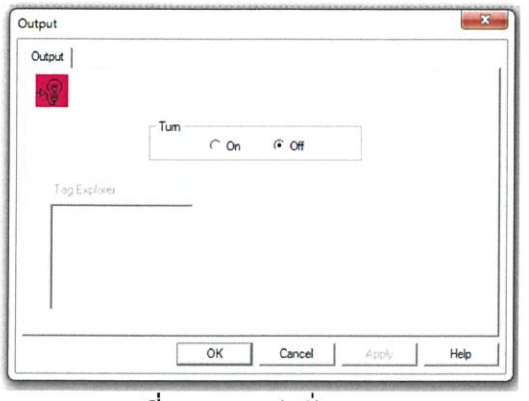
2.6.17 ชุดคำสั่ง SET TIMER



ภาพที่ 2.34 การทำงานของ Set Timer

การทำงานของ Set Timer การตั้งเวลาที่ต้องการแล้วนำไปประยุกต์ใช้ หรือ อาจจะเป็นใช้ตัวนับเวลา หรือ ใช้เป็นตัวจับเวลาโดยการใช้ Wait for Timeout แล้วกำหนดเวลาที่ Set Time หน่วยเป็นวินาที

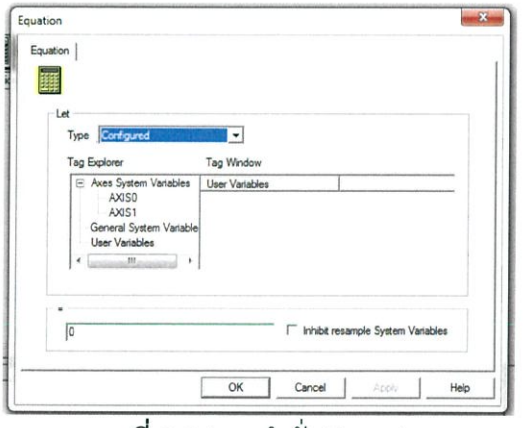
2.6.18 ชุดคำสั่ง OUTPUT



ภาพที่ 2.35 ชุดคำสั่ง Output

ชุดคำสั่ง Output คือคำสั่งให้ Parameter หรือ เอาท์พุทใดๆใน Tag Explorer ทำงานหรือไม่ทำงานด้วยคำสั่ง Turn แล้วจึงเลือก On หรือ Off

2.6.19 ชุดคำสั่ง EQUATION



ภาพที่ 2.36 ชุดคำสั่ง Equation

ชุดคำสั่ง EQUATION เป็นชุดคำสั่งเพื่อกำหนดค่าดังนี้ User Variable , Indirect Variable , Certain System Variables , Discrete Output , Output Group , DH-485 Variable , Cam Profile Point , SLC Variable และ SLC Indirect Variable

การเลือก Type จะบอกว่าจกำหนดค่าประเภทไหน Configured , Master/Slave CAM Position types , Master CAM Time type และ Indirect Variable

1. รายละเอียดของ Configured ที่สามารถกำหนดค่าได้แก่ Axis System Variable , AxisLink I/O output , output group, a DH-485 variable , General System variable , SLC variable หรือ User variable

2. รายละเอียดของ Master/Slave CAM Positionที่สามารถกำหนดค่าได้แก่ Position

Lock Cam Master หรือ Slave Cam position

3. รายละเอียดของ Master CAM Time ที่สามารถกำหนดค่าได้แก่ Time Lock Cam Master Cam point

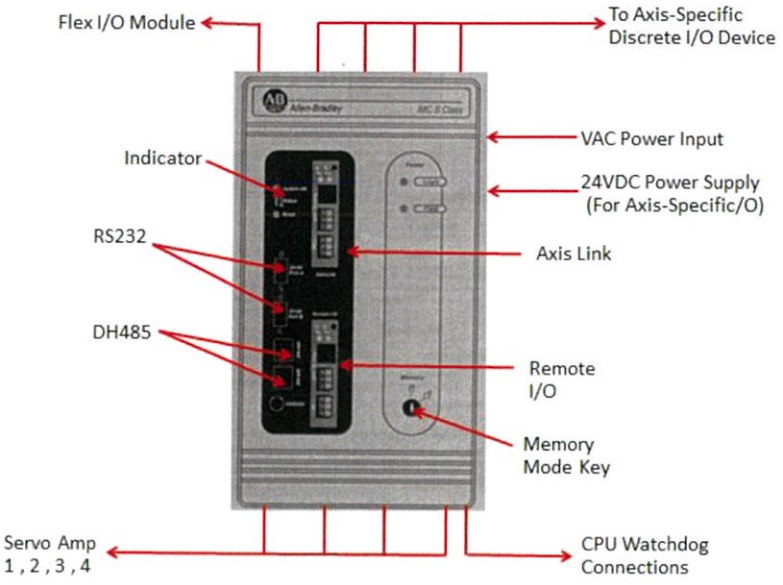
4. รายละเอียดของ Indirect Variable ที่สามารถกำหนดค่าได้แก่ Variable Address , Indirect จะใส่ได้ในช่วง 0 ถึง 1999 , iCODE เวอรชั่น 3 หรือ สูงกว่าจะใส่ได้ในช่วง 0 ถึง 1999 , iCODE ต่ำกว่าเวอรชั่น 3 จะใส่ได้ในช่วง 0 ถึง 999

2.7 ไอเอ็มซีเอส (IMC-S)

เป็นตัวควบคุมตำแหน่ง(Motion Controller) ของทาง Allen Bradley สามารถควบคุมได้ถึง 4 แกน สนับสนุนการควบคุมด้วยตัวเอง(Stand Alone) เหมาะสำหรับการทำงานหลายประเภท ประยุกต์ใช้ในอุตสาหกรรม IMC-S ทำงานร่วมกับตัวขับภายนอก(External Motor Driver) รองรับการทำ Closed loop ที่ใช้ Encoder การสั่งการแบบจุดต่อจุด(Point-to-Point) ด้วยโปรไฟล์ต่างๆ เช่น trapezoidal , parabolic หรือ S-curve หรือควบคุม ตำแหน่ง(Position) ความเร็ว(Speed) ความเร่ง(Acceleration) ความหน่วง(Deceleration) นอกจากนี้ยังทำงานร่วมกันหลายแกน (Interoperation) ทำให้ระบบสามารถทำงานได้ซับซ้อนยิ่งขึ้น ในโครงงานนี้ใช้ รุ่น IMC-S/234

ภาพที่ 2.37 IMC-S Motion Controller

ในภาพที่ 2.37 เป็น Connection ของ IMC-S/234 มีช่องสำหรับ DH485 เป็นโปรโตคอลสื่อสารชนิดหนึ่งสามารถใช้แบบสายอนุกรม(Recommended Serial) ทั้ง RS232 หรือ RS422 โดยการ Dip Switch เป็นตัวตั้งค่า และใช้ AxisLink ได้ ควบคุมได้ถึง 4 แกน



ภาพที่ 2.38 IMC-S/234 Connection

คุณสมบัติบางส่วนของ IMC-S/234

1. ประเภทของ Encoder : Incremental AB quadrature , optically isolated, differential with marker channel

2. Maximum Encoder Frequency : 4,000,000 counts per second (4 MHz). This is equivalent to a channel frequency of 1 MHz in 4X quadrature decode mode

3. Decode Modes : 4X Quadrature, Step/Direction, Count Up/Count Down.

4. Servo Output Resolution : 16 bits, 305 μV or 4.58 μA per bit.

5. Serial bus Information Code : ASCII

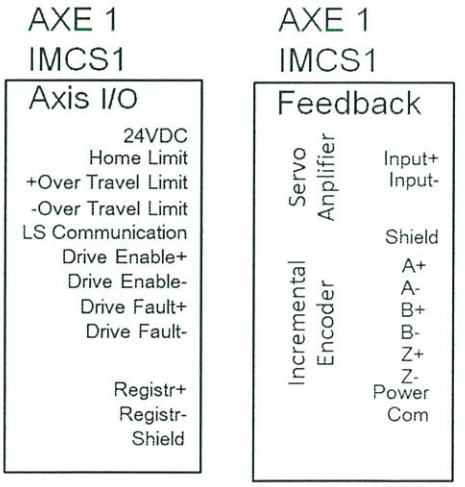
6. DH485 Channel Type : Optically isolated half-duplex RS-485.

7. DH485 Baud Rate : 9,600 or 19.2k Baud (user-selectable).

คุณสมบัติเฉพาะของ Servo ในภาพที่ 2.38 ส่งผลต่อการใช้งานของ Encoder และ Motor ซึ่ง IMC-S ที่เลือกมานั้นเหมาะสมกับการใช้งานแล้ว

Servo Loop Sample and Update Rate	250 Hz to 2 kHz for each of 2 or 4 axes.
Maximum Feedback Frequency	4 MHz (4,000,000 feedback counts per second).
Absolute Position Range	± 1,000,000,000 feedback counts for Linear AxisAxis:Linearlinear axes; R for rotary axes.
Absolute Position Resolution	15 position unit digits or 32 feedback count bits, whichever is less.
Speed Range	0.00001 feedback counts per servo update to 4,000,000 feedback counts per second.
Speed Resolution	15 position unit digits or 15 feedback count bits, whichever is less.
Acceleration/ Deceleration Range	0.00001 feedback counts per servo update2 to 4,000,000,000 feedback counts per second2.
Acceleration/Deceleration Resolution	15 position unit digits or 15 feedback count bits, whichever is less.
Electronic Gearing Gear Ratio Range	0.00001:1 to 9.99999:1 (slave counts : master counts).
Electronic Gearing Gear Ratio Resolution	8 position unit digits or 32 feedback count bits.
Servo Gain Resolution	32 bit floating point.
Servo Output Limit Range	0 to 100%.
Servo Output Limit Resolution	305 µ (voltage output); 4.58µA (current output).

ภาพที่ 2.39 คุณสมบัติเฉพาะของ Servo



ภาพที่ 2.40 หน้าการ์ด I/O IMCS

รายละเอียดของแต่ละขาในหน้าการ์ด

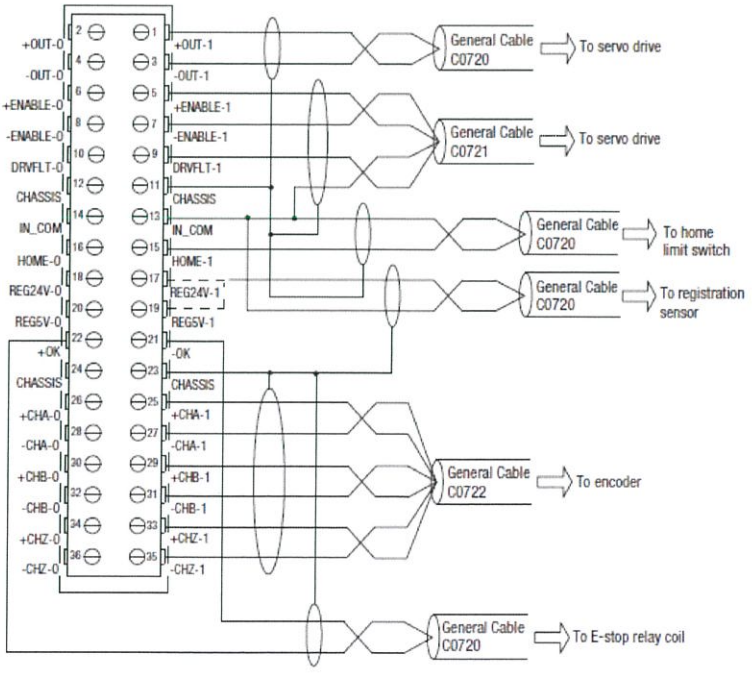
1. 24VDC+ ทำหน้าที่เป็นไฟเลี้ยง (Supply) IMCS
2. HOME Limit ทำหน้าที่รับจากเซนเซอร์ HOME
3. Over Travel Limit ทำหน้าที่รับจากเซนเซอร์ Over Travel
4. LS Communication ทำหน้าสื่อสารกับ Logic solver หรือระบบนิรภัยควบคุมแบบอัตโนมัติ
5. Drive +Enable ทำหน้าที่ส่งสัญญาณ Enable 2 ไปสู่ตัวขับมอเตอร์
6. Drive -Enable ทำหน้าที่เป็นกราวด์ (Ground) 0V ของ Drive+ Enable
7. Drive Fault+ ทำหน้าที่ส่งสัญญาณเกิดการผิดพลาด (Fault)ใดจากอุปกรณ์
8. Drive Fault- ทำหน้าที่เป็นกราวด์ (Ground) 0V ของ Drive Fault+
9. Registr+ ทำหน้าที่เป็นไฟเลี้ยง (Supply) 24V หรือ 5V ได้ให้เซ็นเซอรที่มาต่อ
10. Registr- ทำหน้าที่เป็นกราวด์ (Ground) 0V ของ Registr+
11. Shield สำหรับเชื่อม Shield ของสายอื่นๆเพื่อลดสัญญาณรบกวน (Noise)
12. +A,-A คือ Feedback Encoder Channel A
13. +B,-B คือ Feedback Encoder Channel B
14. +Z,-Z คือ Feedback Encoder Channel Z
15. Power ทำหน้าที่เป็นไฟเลี้ยง 5V ได้ให้ Encoder มาต่อ
16. Input+, Input- ทำหน้าส่งสัญญาณควบคุม Analog เทียบ 2 ขา

2.8 Analog Encoder (AE) Servo Module

การ์ด M02AE จะอยู่การ์ดซีรี่ย์ 1756 ในชื่อเต็มว่า Analog Encoder (AE) Servo Module ซึ่งจะสามารถใช้ร่วมกับราง(Chassis) และ เทอรมินัล(Removable Terminal Block)

ภาพที่ 2.41 การ์ด 1756-M02AE

หน้าที่ของการ์ดนี้จะเป็น Encoder ที่ส่งกลับมาจาก Feedback ในหนึ่งการ์ด(Card) นั้นจะรองรับได้ 2 แกนการ Wiring จะมี 4 แบบ คือ Servo module RTB , 1394 Servo Drive , Ultra 100 Series Drive และ Ultra 200 Series Drive ในงานที่ได้รับมอบหมายนั้นจะใช้แบบ Servo module RTB มีลายละเอียดในแต่ละจุดหน้าการ์ดบนเทอรมินัลในภาพที่ 2.42



ภาพที่ 2.42 เทอรมินัลแบบ Servo module RTB

1. +OUT , -OUT ทำหน้าที่ส่งสัญญาณควบคุมในรูปแบบแอนาล็อกเซอร์โว(Analog)

2. +ENABLE, -ENABLE ทำหน้าที่ส่งสัญญาณ Enable การทำงานพีแอลซีจะเป็นสวิตช์ระหว่าง 2 ขานี้

3. DRVFLT- ทำหน้าที่ส่งสัญญาณเกิดการผิดพลาด (Fault) ใดขึ้นอยู่ที่การตั้งค่าพารามิเตอร์

4. CHASSIS ทำหน้าลดสัญญาณรบกวน (Noise) จะเชื่อมกับ Shield ของสายอื่นๆ

5. IN_COM ทำหน้าเป็นกราวด์ให้ HOME และ เซนเซอร์อื่นๆ

6. REG24V ทำหน้าที่เป็นไฟเลี้ยง 24V (Supply) ให้เซ็นเซอรที่มาต่อ

7. REG5V ทำหน้าที่เป็นไฟเลี้ยง 5V (Supply) ให้เซ็นเซอรที่มาต่อ

8. OK ทำหน้าที่เป็นตัวป้องกันโดย Interlock Safety Switch ต่อกับ E-Stop Relay

9. +CHA,-CHA คือ Feedback Encoder Channel A

10. +CHB,-CHB คือ Feedback Encoder Channel B

11. +CHZ,-CHZ คือ Feedback Encoder Channel Z

รายละเอียดของ Encoder Input โดยสังเขปดังนี้

1. Type: Incremental AB quadrature with marker

2. Mode: 4X quadrature

3. Rate: 4 MHz counts per second maximum

4. Electrical Interface

    4.1 Voltage Range: Optically isolated 5V differential

    4.2 ON Stage: 3.4V to 5.0V

    4.3 OFF Stage: 0V to 1.8V

    4.4 nput impedance: 531 Ohms differential

รายละเอียดของ Servo Output โดยสังเขปดังนี้

1. Type: Analog

2. Voltage Range: 10V

3. Isolation: 20K Ohm

4. Voltage resolution: 16 bits

5. Gain error: ±4%

2.9 Motor Driver

หน้าที่ของตัวขับมอเตอร์(Motor Driver) จะทำหน้าที่ควบคุมไฟฟ้ากำลังเพื่อขับมอเตอร์ให้ทำงานเนื่องมาจากตัวควบคุมตำแหน่ง(Motion Control) กำลังไม่พอในการควบคุมมอเตอร์ นอกจากนี้ตัวขับมอเตอร์ยังสามารถคำสั่งด้วยตัวมันเอง หรือ รับคำสั่งจากอินพุทภายนอกขึ้นกับรุ่นนั้น ในหัวข้อนี้กล่าวถึงตัวขับมอเตอร์ของ Control Techniques รุ่น UNI 1401 , UNI1403 และ UNI1406

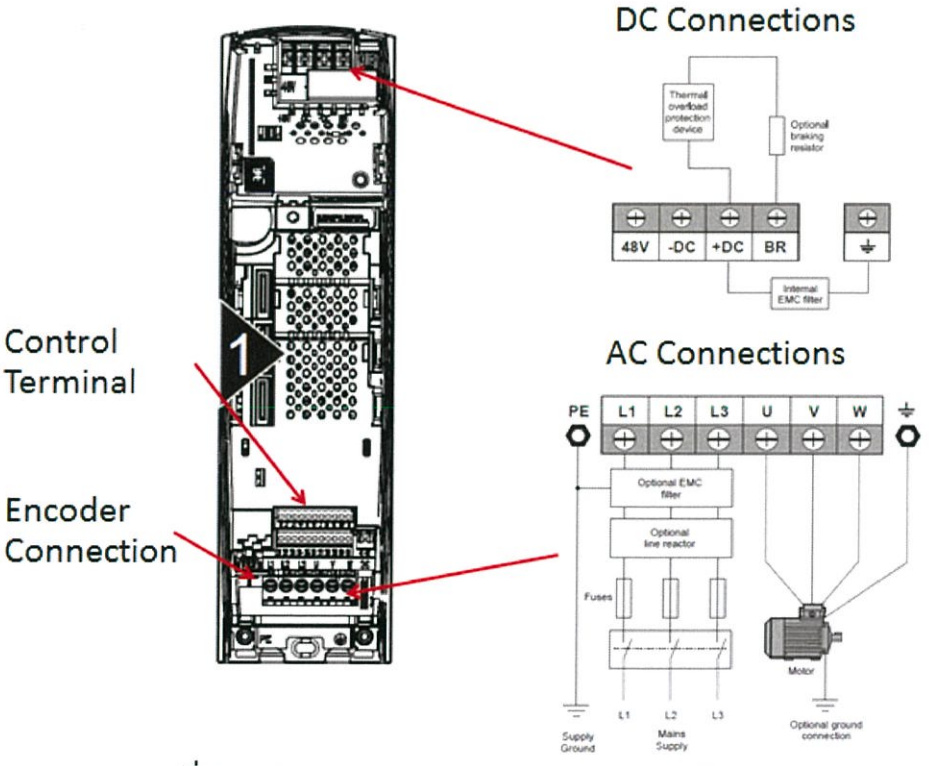
ในรุ่น UNI1401 , UNI1403 , UNI1406 มีลักษณะเดียวกันคุณสมบัติใกล้เคียงกันหมด ในภาพที่ 2.43 จะเห็นว่าทั้ง 3 รุ่นลักษณะภายนอกต่างกันแต่เป็นซีรี่ย์เดียวกันเนื่องมาจากโดยเริ่มต้น UNI DRIVER เป็นของ Control Techniques หลังจากนั้น Emerson ได้ซื้อกิจการและทำ UNI รุ่นใหม่ออกมาโดยเปลี่ยนกรอบภายนอกโดยรวมฟังก์ชันการใช้งานไม่ต่างกัน แตกต่างกันที่ Drive rating และ Connection ทั้งหมด 4 รุ่นสิ่งที่ต่างกันโดยสังเขปคือ Drive Rating จะต่างกันดังตารางที่ 2.1

ตารางที่ 2.1 การเปรียบเทียบของตัวขับมอเตอร์ Drive rating

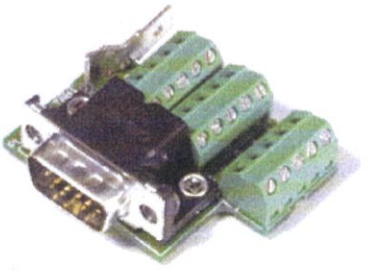
UNIDRIVE		1401	1403	1406
Normal Duty	Maximum Continuous Current output (A)	2.8	5.0	11
	Nominal Power at 220V (kW)	1.1	2.2	5.5
	Motor Power at 230V (Hp)	1.5	3.0	7.5
	Peak Current (A)	3.0	5.5	12.1
Heavy Duty	Maximum Continuous Current output (A)	2.1	4.2	9.5
	Open loop Peak Current (A)	3.1	6.3	14.2
	Closed loop Peak Current (A)	3.6	7.3	16.6
	Nominal Power at 220V (kW)	0.75	1.5	4.0
	Motor Power at 230V (Hp)	1.0	3.0	5.0

ภาพที่ 2.43 UNI1401 (ซ้าย) , UNI1403 (ขวา) , UNI1406 (ขวา)

ภาพที่ 2.44 การ Connection ของ UNI1403 และ UNI1406 มีส่วนที่จำเป็นต้องการ Wiring ใหม่เมื่อทำการย้ายระบบ(Migration) คือ DC Connections , AC Connection , Control Terminal และ Encoder Connection ในส่วนนี้จะใช้พอร์ท(Port) 15-pins แปลงร่วมกับเทอร์มินัล (Terminal) ภาพที่ 2.45

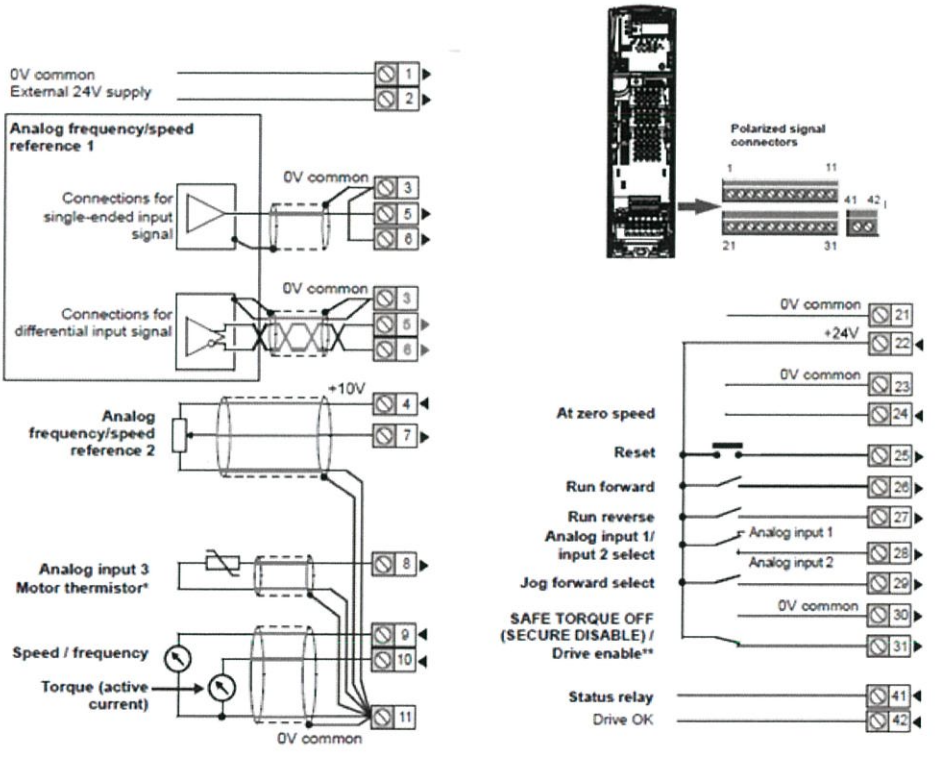


ภาพที่ 2.44 Power Connection UNI1403 และ UNI1406



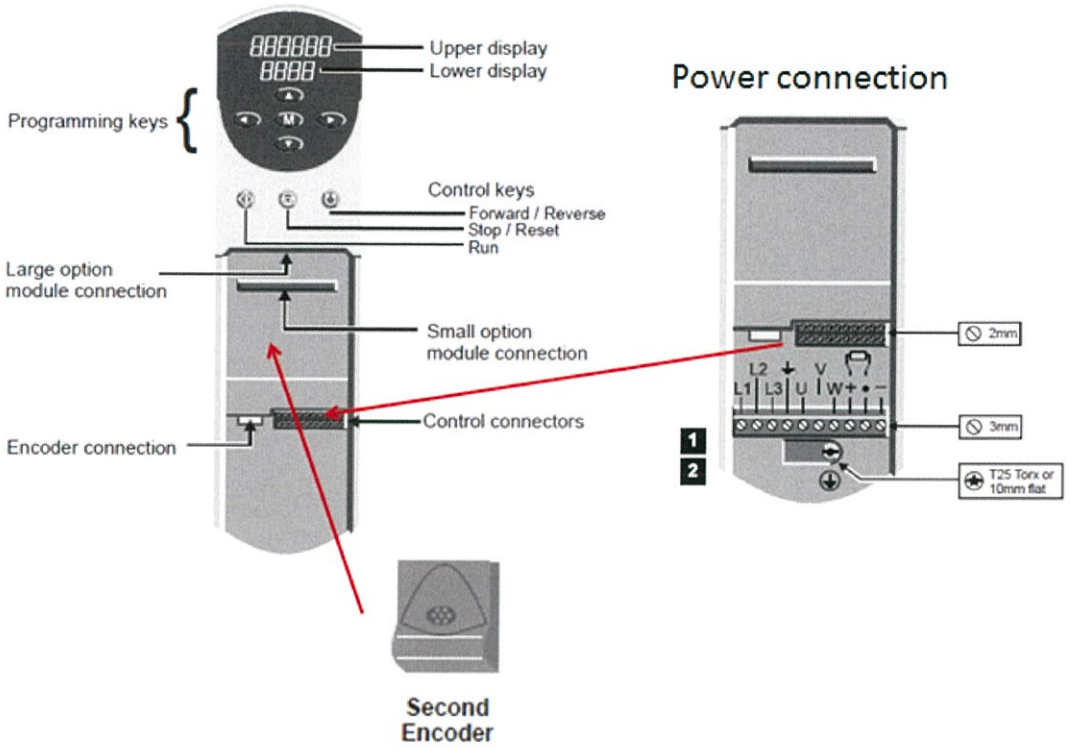
ภาพที่ 2.45 15-pins Adaptor Terminal

ในส่วนของ Control Terminal ของ UNI 1403 และ UNI1406 มี 3 ชิ้น จะเป็นส่วนที่เชื่อมต่อกับตัวควบคุมตำแหน่งดังนั้น ในแต่ละจุดบนหน้าเทอร์มินัลมีวิธีการWiringแต่ละจุดไม่เหมือนในภาพที่ 2.46

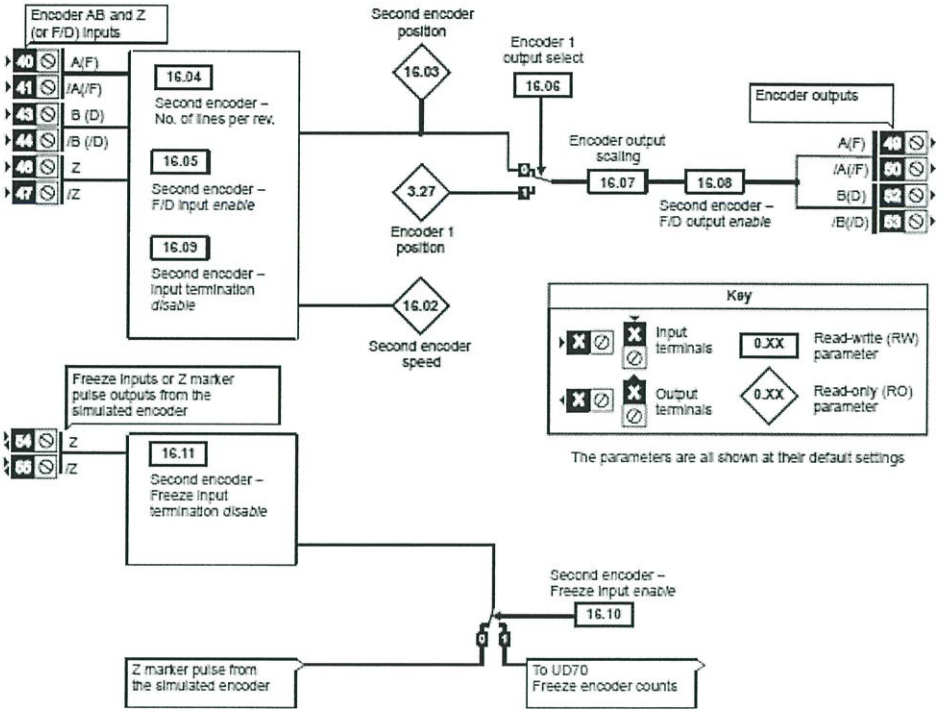


ภาพที่ 2.46 Wiring ของ Connection Terminal UNI1403 และ UNI1406

ภาพที่ 2.47 การ Connection ของ UNI1401 มีส่วนที่จำเป็นต้องการ Wiring ใหม่เมื่อทำการย้ายระบบ(Migration) คือ Control Terminal และ Encoder Connection ยังเพิ่มเติมด้วยโมดูลส่วนเสริมที่เป็น Second Encoder ในภาพที่ 2.48 เป็น Connection ของโมดูลส่วนเสริม

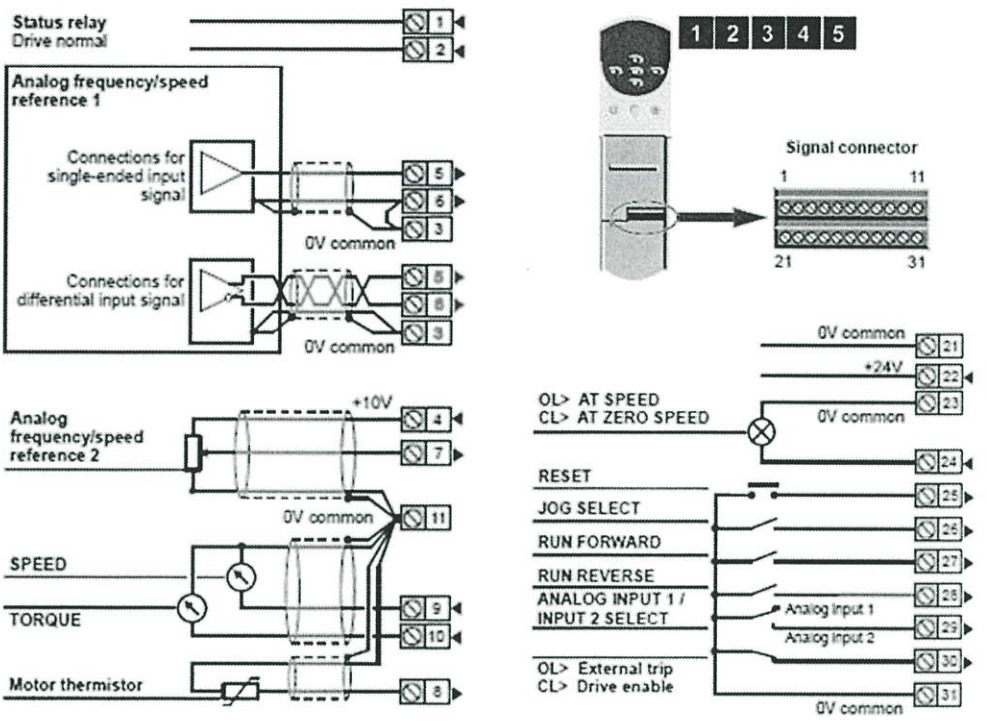


ภาพที่ 2.47 Power Connection UNI1401



ภาพที่ 2.48 Power Connection ส่วนเสริม Second Encoder

ในส่วนของ Control Terminal ของ UNI 1401 มี 2 ชิ้น จะเป็นส่วนที่เชื่อมต่อกับตัวควบคุมตำแหน่งดังนั้น ในแต่ละจุดบนหน้าเทอร์มินัลมีวิธีการ Wiring แต่ละจุดไม่เหมือนในภาพที่ 2.52



ภาพที่ 2.49 Wiring ของ Connection Terminal UNI1401

ในการ Wiring สายนั้น ในบางสายของแต่ละแกนอาจแตกต่างกันไปตามประเภทของเครื่องจักร Encoder เชื่อมต่อที่มาจากมอเตอร์และในขณะเดียวกันจะเชื่อมไปที่ตัวควบคุมตำแหน่งที่มีจุดร่วมอยู่ที่ตัวขับมอเตอร์เช่นภาพที่ 3 การ Wiring ของตัวขับมอเตอร์และตัวควบคุมตำแหน่งจะอยู่ในบทที่ 3

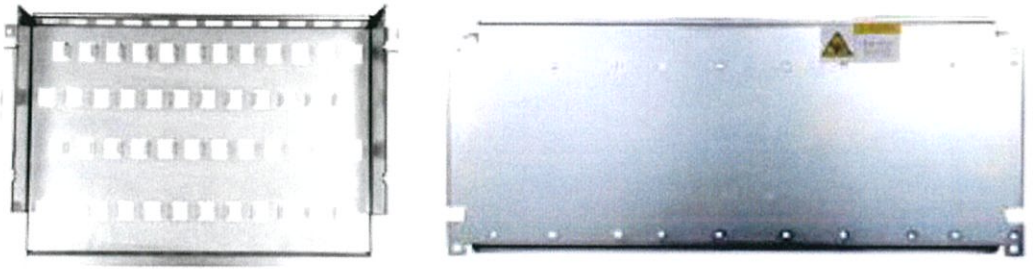
2.10 Conversion Module

Conversion Module จะทำหน้าที่เชื่อมจุดบนหน้าการ์ด(Card) ของอินพุท เอาท์พุท บน ControlLogix กับจุดบนเทอร์มินอล(Terminal) ของ PLC-5 ทำให้ไม่ต้องเสียเวลาในการต่อสาย (Hard-Wired) จุดต่อจุดใหม่ ในภาพที่2.5 ส่วนที่ยื่นออกมาด้านล่างนั้นจะมีพอร์ท(Port) สำหรับ สาย Pre-wired Cable 37-pin หรือ Pre-wired Cable 25-pins

ภาพที่ 2.50 Conversion Module(ซ้าย) เทอร์มินอล(ขวา)ของ PLC-5

Pre-wired Cable 25-pin และ 37-pins(ล่าง)

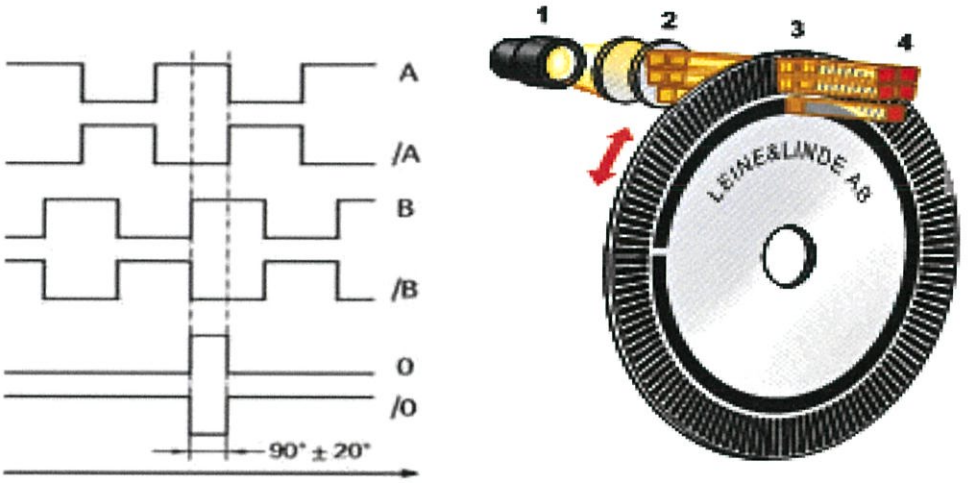
เมื่อนำพีแอลซี ControlLogix มาประกอบรวมกับโดย Conversion Module ติดบน Base Plate และมี Cover Plate มาครอบทับเพื่อให้ราง(Chassis) ของ ControlLogix ติดตั้งเช่นภาพที่ 2.51



ภาพที่ 2.51 Base Plate และ Cover Plate

2.11 Incremental Encoder

เอ็นโคดเดอร์ตำแหน่งแบบเพิ่มค่า (Incremental Position Encoder) ประกอบด้วยแผ่นกลมหรือแผ่นบรรทัดแนวตรง ที่เคลื่อนที่ตามกลไกที่ต้องการวัดระยะทาง ในกรณีของออปติคอลเอ็นโคดเดอร์(Optical Encoder) จะมีการเจาะรูเพื่อให้แสงผ่านไปเป็นระยะ ๆ เพื่อบอกตำแหน่งของแผ่นหมุน โดยจะให้สัญญาณเอาท์พุตเป็นสัญญาณพัลส์ออกมาทุกครั้งที่มีการหมุนแกนเอ็นโคดเดอร์ ทำให้สามารถทราบมุมที่หมุนหรือระยะทางเคลื่อนที่ไปได้ โดยเอาท์พุตปกติจะมีแค่เพียงบิตเดียวหรือสองบิตเพื่อให้บอกทิศทางได้



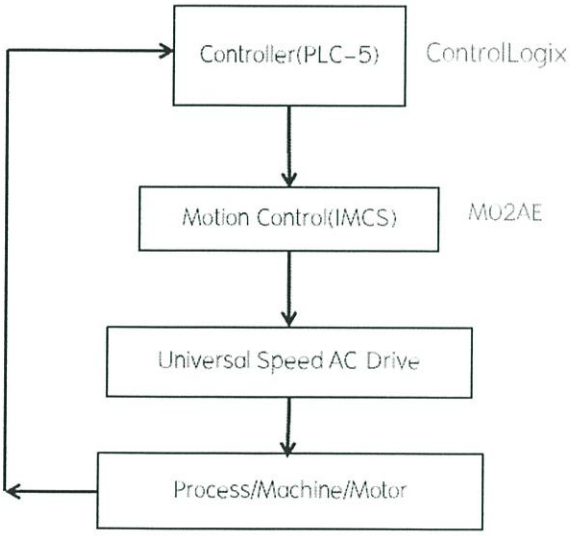
ภาพที่ 2.52 Incremental Encoder

เอ็นโคดเดอร์แบบนี้มีข้อเสียตรงที่ข้อมูลของการเคลื่อนที่จะหายไปหมดเมื่อแหล่งจ่ายไฟฟ้าดับหรือมีการแยกสายสัญญาณออกแม้เพียงชั่วขณะเดียวหรือเมื่อมีการรบกวนของสัญญาณเกิดขึ้น ทำให้ต้องมีการปรับเทียบกับจุดอ้างอิงอยู่ตลอดเวลาเพื่อความถูกต้องไม่สามารถจดจำตำแหน่งแกนหมุนของตัวเองได้ว่าอยู่ที่จุดใด ด้วยเหตุนี้การหมุนกลับไปยังตำแหน่งเริ่มต้น Homing Point นั้นจะทำได้ยาก จึงต้องอาศัยการเก็บข้อมูลพัลส์ตั้งแต่เริ่มต้น เพื่อหาค่าตำแหน่ง ซึ่งในบางครั้งอาจจะอ้างอิงจากจุด Zero Point ซึ่งจะสัญญาณ นอกจากนี้ข้อมูลที่จะนำไปใช้ยังมักจำเป็นที่จะต้องผ่านวงจรนับ (Counter) เพื่อให้ได้ข้อมูลเชิงขนานที่ใช้ในระบบคอมพิวเตอร์

บทที่ 3
ขั้นตอนการดำเนินงาน

3.1 แนวคิด

การย้ายจาก PLC-5 ที่เป็นระบบเก่าไป ControlLogix ที่เป็นระบบใหม่นั้น มีการใช้โปรแกรมคล้ายๆรูปแบบเดิมเนื่องมาจากกระบวนการทำงานยังเป็นรูปแบบเดิมเปลี่ยนคอนโทรลเลอร์ให้มีประสิทธิภาพยิ่งขึ้น ดังนั้นสิ่งที่ควรจะคำนึงถึง คือการยังทำให้กระบวนการทำงานได้ปกติ โดยที่ยังใช้อุปกรณ์ Universal Speed AC Drive ตัวเดิม ดังภาพที่ 3.1



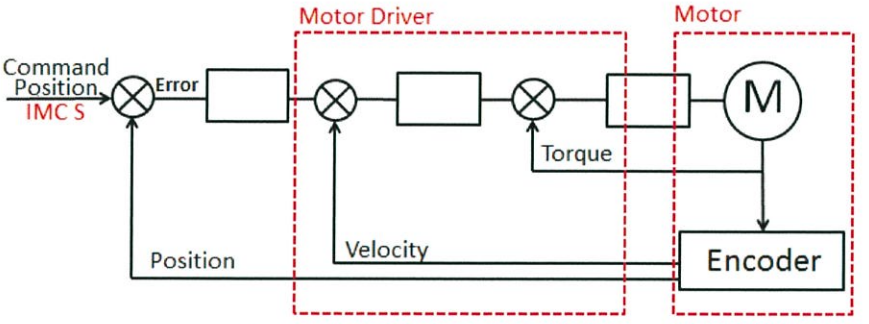
ภาพที่ 3.1 แผนภูมิลำดับการทำงานของกระบวนการ

ดังภาพที่ 3.1 เป็นหลักการทำงาน Controller ส่งค่าคำสั่งในโปรแกรมไปที่ Motion Control จากนั้น Motion Control จะส่งค่าตำแหน่งหรือพารามิเตอร์ ไป Universal Speed AC Drive โดยจะคำณวนค่าที่ได้รับมาเป็นสัญญาณทางไฟฟ้าเพื่อขับมอเตอร์ จากในภาพที่ 3.1 จะถูกเปลี่ยนแปลงอุปกรณ์แทนด้วยตัวอักษรสีแดงทางด้านขวา PLC-5 เป็น ControlLogix และ IMC-S เป็น M02AE

การสื่อสารของ PLC-5 กับ IMC-S นั้นเปรียบสเหมือน IMC-S เป็นอีกแร็คหนึ่ง ในกระบวนการใช้ 14 แกนซึ่งใช้ IMC-S 4 ตัว ดังนั้นเปรียบสเหมือนที่ มีแร็คเพิ่มขึ้นอีก 4 แร็ค แต่ในการปรับปรุงและเปลี่ยนแปลงเป็น ContolLogix นั้นจะใช้การ์ด M02AE ที่จะอยู่บนแร็ค Local แทน

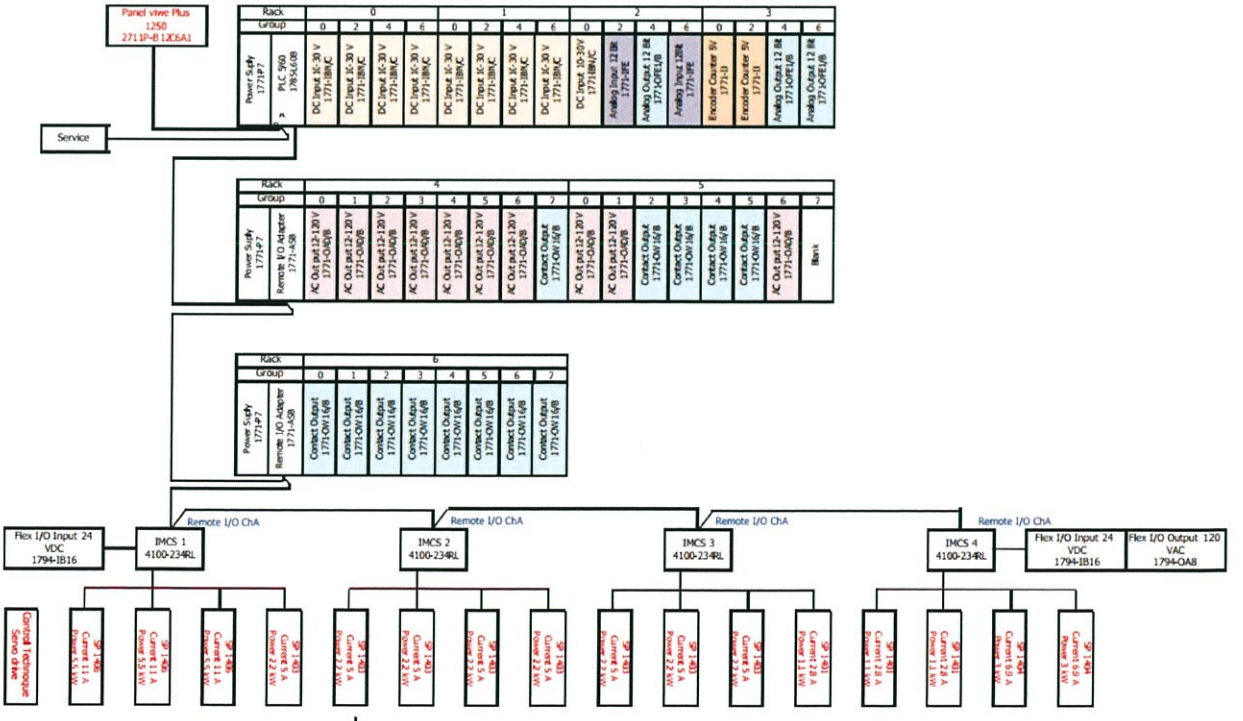
3.1.1 รูปแบบของการควบคุมในระบบเก่า

การควบคุมระบบการขึ้นโครงสร้างยางจะใช้ PLC-5 เป็นพีแอลซี ใช้ IMC-S เป็นตัวควบคุมตำแหน่ง (Motion Control) และตัวขับมอเตอร์ (Motor Driver) เพื่อควบคุมความเร็วแล้วแปลงไปในรูปแบบสัญญาณทางไฟฟ้าไปที่มอเตอร์ในภาพที่ 3.2



ภาพที่ 3.2 ฟังก์ชันบล็อกการควบคุมมอเตอร์

ที่ตู้ไฟฟ้า (Cabinet) จะแบ่งออกเป็น 3 ตู้ คือตู้ควบคุม,ตู้ไฟฟ้ากำลัง และตู้แสดงผล HMI ที่ตู้ควบคุมประกอบไปด้วยอุปกรณ์ PLC-5,I/O Card,Chassis,IMC-S และ I/O Flex Input ที่ตู้ไฟฟ้ากำลังประกอบไปด้วยอุปกรณ์ตัวขับมอเตอร์ UNI 1401 , UNI1403 , UNI1406 , UNIDRIVE M700และแมคเนติกคอนแทคเตอร์ ที่ตู้แสดงผลนั้นมีเพียง PanelView Plus1250 เชื่อมต่อแบบภาพที่ 3.3

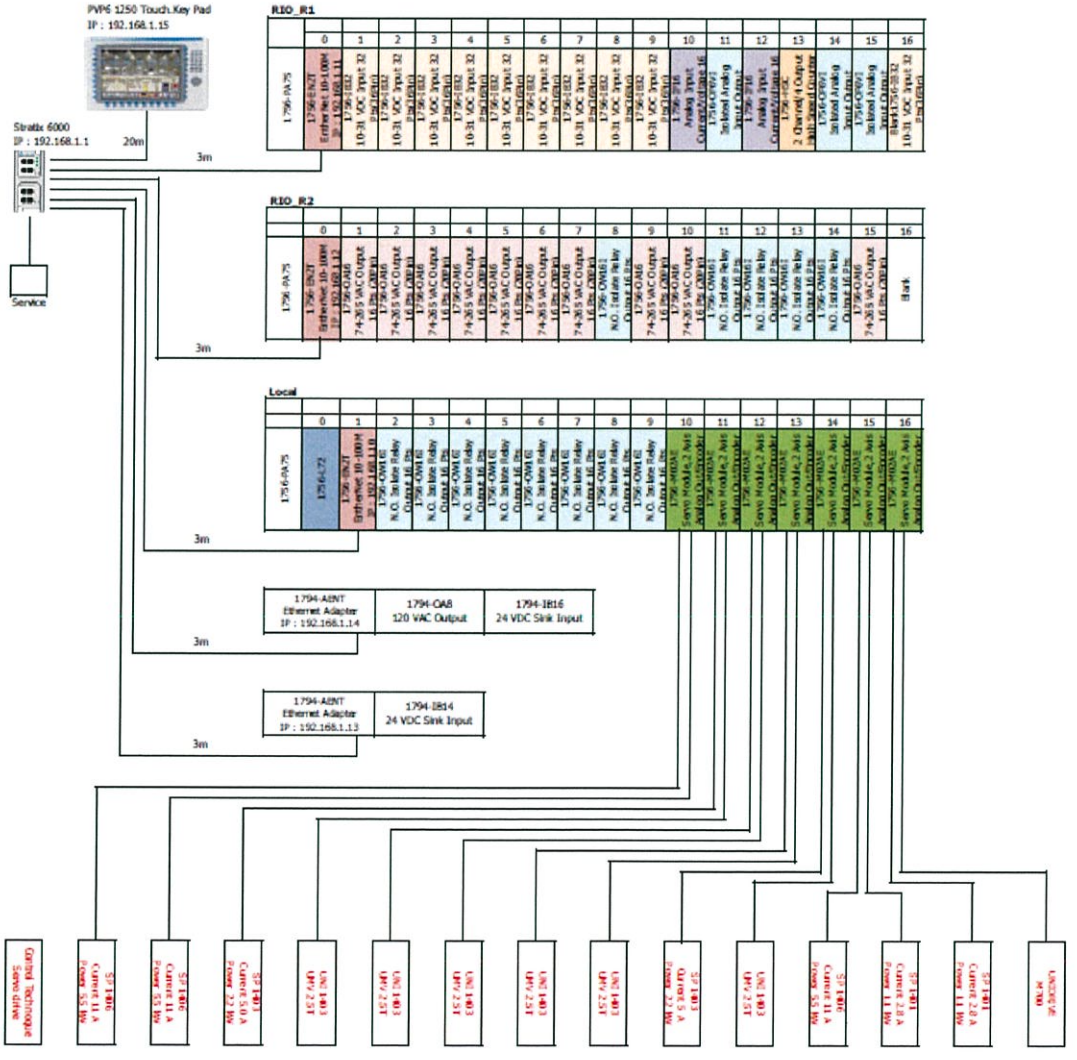


ภาพที่ 3.3 ภาพรวมของระบบ Module4 เดิม

การ Wiring จะมีราง (Chassis) ทั้งหมด 3 รางหรืออาจจะกล่าวในปัจจุบันได้ว่า 3 แร็ค (Rack) แต่ในส่วนโปรแกรมนั้นจะถือว่ามีแร็คทั้งหมด 10 แร็ค (Rack)ด้วยกัน วิธีการหาแร็คนั้นจะอยู่ในหัวข้อ Address Mode ที่ IMC-S จะถือว่าเป็น Remote I/O อีก 4 ชาแนลมีและยัง Flex I/O Input อีกสองหน่วย Flex I/O Output หนึ่งหน่วยที่มาจาก Motor Driver

3.1.2 รูปแบบของการควบคุมในระบบใหม่

การปรับปรุงและเปลี่ยนแปลงระบบ (Migration) จะมีหลักการคือ ดำเนินการเปลี่ยนอุปกรณ์โดยยังคงความสามารถในการทำงานได้เมื่อกลับมาทำงาน หลีกเลี่ยงการปรับเปลี่ยนเครื่องจักรเดิม เป็นที่มาของการเลือกใช้พีแอลซีให้ล้อกับอุปกรณ์เดิม



ภาพที่ 3.4 ภาพรวมของระบบ Module4 ใหม่

ตารางที่ 3.1 เปรียบการ์ดของระบบเดิมและระบบใหม่

ประเภทการ์ด/อุปกรณ์	ระบบเดิม	ระบบใหม่
Input	1771-IBN/C	1756-IB32
Analog Input	1771-IFE(Single End)	1756-IF16(Single End)
Analog Input	1771-IFE(Differential)	1756-IF16(Differential)
Isolated Analog Input	1771-OFE1/B	1756-OF6VI
Counter	1771-IJ x 2	1756-HSC 2 Channel/4Output
AC Output	1771-OAD/B	1756-OA16
Isolated Relay Output	1771-OW16/B	1756-OW16I
Motion Control	IMC-S x 4	M02AE x 7
HMI	PanelViewPlus1250	PanelViewPlus1250
Hub	-	Statix6000
Communication	1771-ASB	1756-EN2T

ในภาพที่ 3.4 จะเห็นว่ายังคงใช้ 3 ราง (Chassis) เช่นเดิมแต่ IMC-S ถูกแทนด้วย M02AE ที่สามารถใส่ร่วมไปในราง (Chassis) ที่แร็ค Local และนำ Flex I/O ต่อแยกเป็นสอง Remote I/O การสื่อสารจะใช้สาย Ethernet ตัวกลางการเชื่อมต่อที่ Hub ที่เป็นผลิตภัณฑ์ของ Allan Bradley คือ Startix6000 นอกจากนี้ M02AE ตัวควบคุมตำแหน่ง สามารถควบคุมได้เพียง 2 แกนเป็นเหตุผลให้ใช้ทั้งหมด 7 ตัวต่างกับ IMC-S ที่สามารถควบคุมได้มากถึง 4 แกนจึงใช้แค่เพียง 4 ตัวเท่านั้นเองใน การแต่ละการ์ด(Card) นั้นจะคล้ายคลึงกับระบบเดิม แต่เปลี่ยนมาใช้ 1756 ซีรี่ย์ ในท้ายสุดทำการเปลี่ยน Panel View Plus 1250 เป็นตัวใหม่แต่ยังคงเป็นรุ่นเดิมMotor Driver ใช้อุปกรณ์เดิมที่มีอยู่ 2 แบบ คือ UNI1401,UNI1403 และ UNI1406 ซึ่งการใช้จะแตกต่างที่ Drive Enable Voltages เป็น 0V,0V และที่ 24V ตามลำดับ

3.2 วิธีการดำเนินงาน

3.2.1 ศึกษาการทำงานของระบบเดิม

ในขั้นตอนแรกของการดำเนินงานจะดูรูปแบบการทำงานของเครื่องจักรในลักษณะใดมีข้อจำกัดทางกายภาพอย่างไรบ้าง สังเกตตำแหน่ง Home ของเครื่องจักร , ตำแหน่ง Hardware Over Travel , ระยะการเคลื่อนที่ , อัตราหน่วง (Decel) , โปรไฟล์ของการหมุน

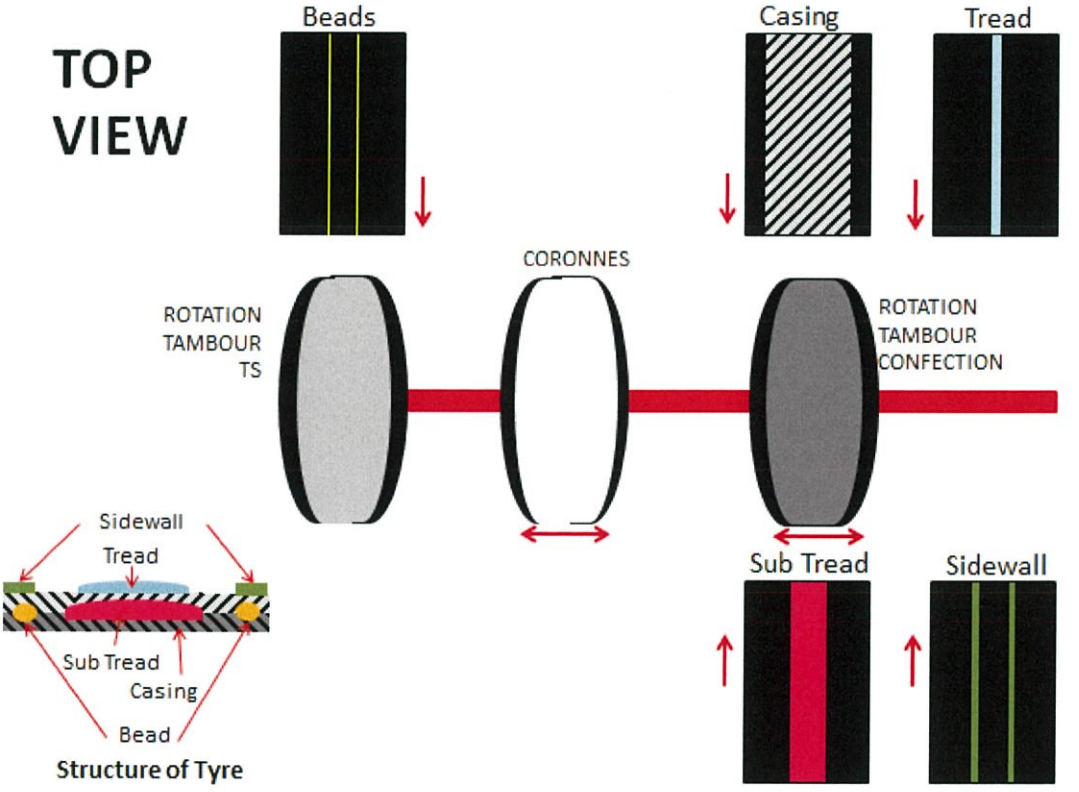
โปรไฟล์การหมุนคือ ประเภทของการเคลื่อนที่ของระบบยังถือเป็นการกำหนดค่าพารามิเตอร์ของโปรแกรม

1. Rotary คือ การเคลื่อนที่เชิงมุมที่เป็นลักษณะหมุนทำให้ในหลายแกนไม่มี Home
2. Linear คือ การเคลื่อนที่เชิงเส้นเป็นลักษณะเคลื่อนที่จากจุดหนึ่งไปอีกจุดหนึ่ง

ตารางที่ 3.2 รายละเอียดชื่อของแต่ละแกน

แกน	ชื่อแกน	โปรไฟล์	Home	Hardware Over Travel	Drive Enable
1	ROTATION TAMBOUR CONFECTION	Rotary	O	X	24V
2	TRANSLATION TAMBOUR CONFECTION	Linear	O	O	0V
3	TRANSFERT TISSU	Linear	O	O	0V
4	TETE TISSU	Rotary	X	X	24V
5	TRANSFERT CHAPE	Linear	O	O	0V
6	TETE CHAPE	Rotary	X	X	0V
7	ROULETEUR CHAPE	Rotary	O	X	24V
8	TRANSFERT PROTECTEUR	Linear	O	O	24V
9	TRANSFERT NAPPE2	Linear	O	O	0V
10	TRANSLATION COURONNES	Linear	O	O	0V
11	ROTATION TAMBOUR TS	Rotary	O	X	0V
12	HAUTEUR POSTE TS	Rotary	X	X	0V
13	LARGEUR COURONNE TS	Linear	O	O	24V
14	GONFLAGE DEGONFLANGE SEAT COURONNES	Linear	O	O	24V

จากการสังเกตพบมีทั้งหมด 14 แกน แต่มีมอเตอร์ถึง 28 ตัว ซึ่งมีมอเตอร์ 14 ตัวที่ไม่ได้สนใจ เนื่องจากสนใจเฉพาะเซอร์โวมอเตอร์เท่านั้น อีก 12 ตัวควบคุมแบบ Open Loop ควบคุมเป็นแบบดิจิตอล 1 หรือ 0 เท่านั้น เมื่อสำรวจในตู้ตัวขับ (Motor Driver) จะมีการ Drive Enable ที่ต่างกัน โดยใช้พีแอลซีเป็นสวิตช์เป็นซิงค์ (Sink)หรือซอร์ส (Source)



ภาพที่ 3.5 ภาพรวมจากมุมบนของ Module4

ภาพรวมการทำงานเมื่อมองจากมุมด้านบนภาพที่ 3.5 จะแบ่งสถานีออกเป็น 3 สถานีหลักๆ มี CORONNES เป็นตัวกลางนำส่งยางมีลักษณะคล้ายตัวจับระหว่าง ROTATION TAMBOUR TS ซึ่งไม่สามารถเคลื่อนที่ได้ทำได้เพียงเป็นแกนหมุนเท่านั้น กับ ROTATION TAMBOUR CONFECTION ที่สามารถเคลื่อนที่ไปมาระหว่างสถานี 2 และ 3 เมื่อเสร็จขั้นตอนพันครบทั้ง 3 สถานี CORONNES จะนำยางที่เสร็จแล้วกลับมาที่ตำแหน่งเริ่มต้นดังภาพ

3.2.1.1 อธิบายการทำงานของแต่ละแกน

-แกนที่1 ROTATION TAMBOUR CONFECTION

แกนนี้มีลักษณะเป็นแกนหมุนทำหน้าที่เป็นแกนกลางในการม้วนส่วนประกอบยางเช่น แผ่นยาง (Casing/Tissue) , แผ่นกันยางรั่ว (Sub Tread / Protection layer) ,แถบป้องกันเสียดสี (Anti-Chafing Strip/Side Wall) และ หน้ายาง (Tread) ให้เป็นลักษณะเป็นวงกลม

ภาพที่ 3.6 ตัวอย่างของ Rotation Tambour Confection

-แกนที่2 TRANSLATION TAMBOUR CONFECTION

แกนนี้ทำหน้าที่เคลื่อน ROTATION TAMBOUR CONFECTION ไปสถานีต่อไปตามลำดับดังนี้

-แกนที่3 TRANSFERT TISSU

แกนนี้มีลักษณะเป็นแขนเพื่อช่วยในการพันรอบ ROTATION TAMBOUR CONFECTION ในช่วงที่จับแผ่นยาง(Casing/Tissue) ไปวางที่ ROTATION TAMBOUR CONFECTION และ ช่วงที่พันจนครบระยะของความยาวยาง

-แกนที่4 TETE TISSU

เป็นแกนลักษณะจ่าย แผ่นยาง(Casing/Tissue) เข้า ROTATION TAMBOUR CONFECTION โดยทำงานร่วมกับ TRANSFERT TISSU ด้วย

-แกนที่5 TRANSFERT CHAPE

แกนนี้มีลักษณะเช่นเดียวกับ TRANSFERT TISSU แตกต่างที่แกนนี้ใช้สำหรับช่วยในการพันรอบของหน้ายาง (Tread) ให้เป็นลักษณะเป็นวงกลม

-แกนที่6 TETE CHAPE

เป็นแกนลักษณะจ่าย หน้ายาง (Tread) เข้า ROTATION TAMBOUR CONFECTION โดยทำงานร่วมกับ TRANSFERT CHAPE ด้วย

-แกนที่7 ROULETEUR CHAPE

เป็นแกนที่ทำหน้าที่บดให้ หน้ายาง (Tread) และแถบป้องกันเสียดสี (Anti-Chafing Strip/Side Wall) แนบไปกับแผ่นยาง(Casing/Tissue) ที่ถูกใส่ เคฟลาร์ (Beads/Kevlar) , แผ่นกันยางรั่ว(Sub Tread / Protection layer) และถูกพับขอบแล้ว

-แกนที่8 TRANSFERT PROTECTEUR

เป็นแกนที่มีลักษณะการทำงานคล้าย TRANSFERT TISSU และ TRANSFERT CHAPE แต่อยู่ในรูปแบบแนวนอน ซึ่งจะทำหน้าจับแถบป้องกันเสียดสี (Anti-Chafing Strip/Side Wall) แปะที่ ROTATION TAMBOUR CONFECTION

- แกนที่9 TRANSFERT NAPPE2

เป็นแกนที่วัดขนาดแผ่นกันยางรั่ว(Sub Tread / Protection layer) แล้วเตรียมนำไปแปะที่ ROTATION TAMBOUR CONFECTION

-แกนที่10 TRANSLATION COURONNES

เป็นแกนที่เคลื่อนที่ของ COURONNES ซึ่งอุปกรณ์ชิ้นนี้ทำหน้าที่เสมือนตัวจับแผ่นยางที่มีลักษณะวงกลม

-แกนที่11 ROTATION TAMBOUR TS

แกนนี้มีลักษณะคล้าย ROTATION TAMBOUR CONFECTION แต่แกนนี้มีหน้าที่สำหรับพันเคฟลาร์ (Beads/Kevlar) เท่านั้นจะมีความพิเศษตรงที่ จะมีร่องสำหรับพัน และมีตัวจับเคฟลาร์ในจังหวะที่พันครบรอบ

-แกนที่12 HAUTEUR POSTE TS

แกนนี้ทำหน้าที่จ่าย และแปะโลโก้ที่แก้มยางโดย ROULETEUR CHAPE จะบดให้ติดกับแผ่นยาง (Casing/Tissue)

-แกนที่13 LARGEUR COURONNE TS

แกนนี้จะอยู่บน COURONNE ซึ่งอุปกรณ์นี้จะปรับระยะการจับเคฟลาร์ (Beads/Kevlar)

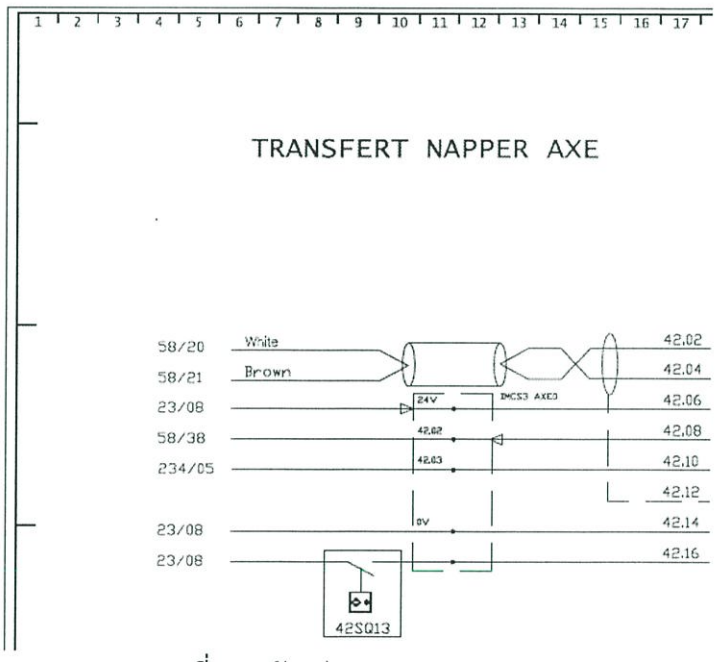
-แกนที่14 GONFLAGE DEGONFLANGE SEAT COURONNES

จะทำอยู่บน COURONNE ทำหน้าที่จับเคฟลาร์ (Beads/Kevlar)ที่ LARGEUR COURONNE TS ได้ปรับไว้ก่อนหน้านอกจากนี้ยังทำหน้าที่จับยางที่ทำเสร็จแล้วมารอ ที่นี้

3.2.2 การออกแบบ Loop Wiring

การสื่อสาร (Communication) ของอุปกรณ์บางประเภทจำเป็นจะต้องมีการ Wiring เฉพาะตัวสิ่งที่สำคัญ คือการดู Loop Wiring เก่า เทียบกับคู่มือ(Manual) ลดการผิดพลาดของการ Wiring อุปกรณ์และ ดูข้อจำกัดของอุปกรณ์ใหม่อีกด้วย

แบบ Wiring เดิมของทางบริษัทมีรูปแบบที่น่าสนใจ คือการเรียนจุดในหน้า และการไปหน้าในภาพที่ 3.7 ที่หัวกระดาษจะมีเลขบอกแนวเช่น 58/20 คือ มาจากหน้าที่ 58 แนวที่ 20 อีกจุดหนึ่งคือ การเรียกตำแหน่งในหน้ากระดาษ เช่น 42.02 คือ อยู่ในหน้าที่ 42 ตำแหน่ง 02 การกำหนดตำแหน่งไม่มีเงื่อนไขในการกำหนดแต่ห้ามซ้ำกันในหน้าเดียวกัน ตัวอย่าง สมมติว่าในหน้า 234 แนวที่05 ต้องการเชื่อมสาย 42.10 จะเรียกดังนี้ในหน้า 234 จะเขียนว่า 42/4 (42.10)

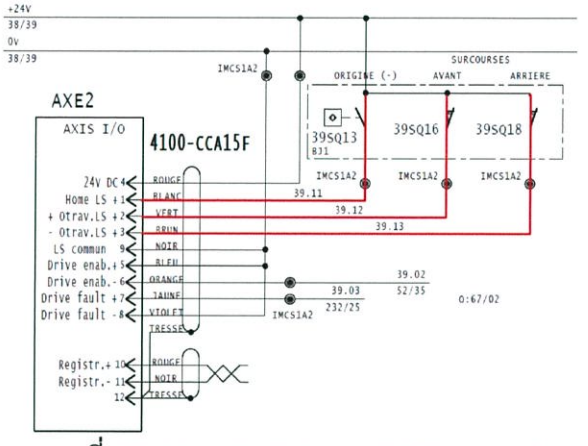


ภาพที่ 3.7 ตัวอย่างของ Loop Wiring

3.2.2.1 Loop Wiring ของ IMCS

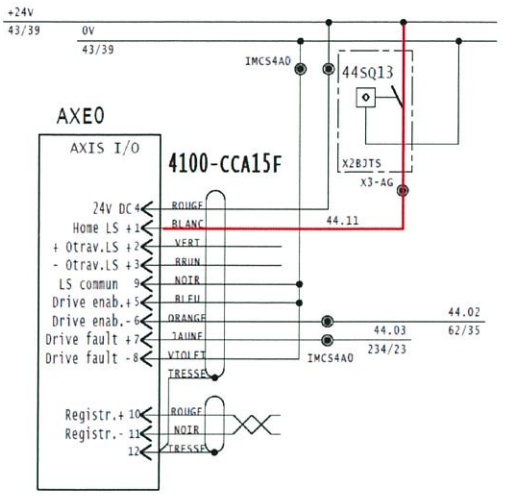
การ Wiring ของตัวควบคุมตำแหน่ง (Motion Control) เดิมจะมีลักษณะคล้ายกันต่างกันที่การต่อ Home และ Over Travel ดังนั้นจะแบ่งออกเป็น 3 กรณี มีHome มีOver Travel, มี Home ไม่มีOver Travel และ ไม่มีHome ไม่มีOver Travel โดยรายละเอียดของแต่ละขาได้กล่าวไว้ในบทที่ 2 IMCS

-มี Home มี Over Travel ในภาพที่ 3.8 เป็น Loop Wiring เดิมของ IMCS1 แกน 2 หรือ TRANSLATION TAMBOUR CONFECTION สังเกตที่เส้นสีแดง ซึ่งจะมาจาก Home Limit Switch และ Over travel Limit Switch



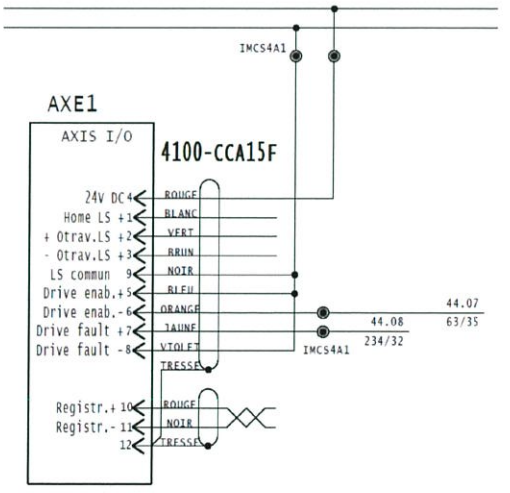
ภาพที่ 3.8 Limit Switch ของ IMCS1 แกน 2

-มี Home ไม่มี Over Travel ในภาพที่ 3.9 เป็น Loop Wiring เดิมของ IMCS4 แกน 0 หรือ ROTATION TAMBOUR TS สังเกตที่เส้นสีแดง ซึ่งจะมาจาก Home Limit Switch



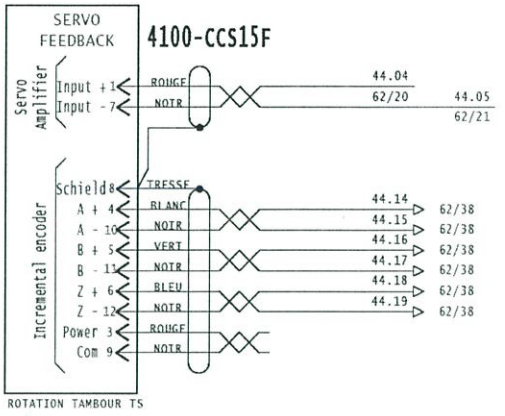
ภาพที่ 3.9 Home Limit Switch ของ IMCS4 แกน 0

-ไม่มี Home ไม่มี Over Travel ในภาพที่ 3.10 เป็น Loop Wiring เดิมของ IMCS4 แกน 1 หรือ HAUTEUR POSTE TS



ภาพที่ 3.10 Limit Switch ของ IMCS4 แกน 1

การ Wiring ของ Servo Feedback นั้นจะมีรูปแบบคล้ายกันทุกแกนจะ Wiring หนึ่งต่อหนึ่งกับ Encoder ด้วยหัว 15-pins



ภาพที่ 3.11 Servo Feedback ของ IMCS4 แกน 1

3.2.2.2 Loop Wiring ของ M02AE

รายละเอียดของแต่ละขาได้กล่าวไว้ในบทที่ 2 M02AE ซึ่งเมื่อนำมาเทียบกับ IMCS ที่ถูกผลิตขึ้นมานานในเทคโนโลยีสมัยนั้น ทำให้เกิดข้อแตกต่างทางกายภาพเป็นหัวข้อ

-ไม่มี Over Travel Limit Switch

-ไม่มี 24VDC เนื่องจากสามารถรับไฟจากราง (Chassis) ได้

-ไม่มี LS Communication สามารถใช้พีแอลซีสื่อสารโดยตรงกับ Safety PLC

-ไม่มี Drive Fault- เพราะมีกราวด์ (Ground)ภายใน

-ไม่มี Input+,Input-

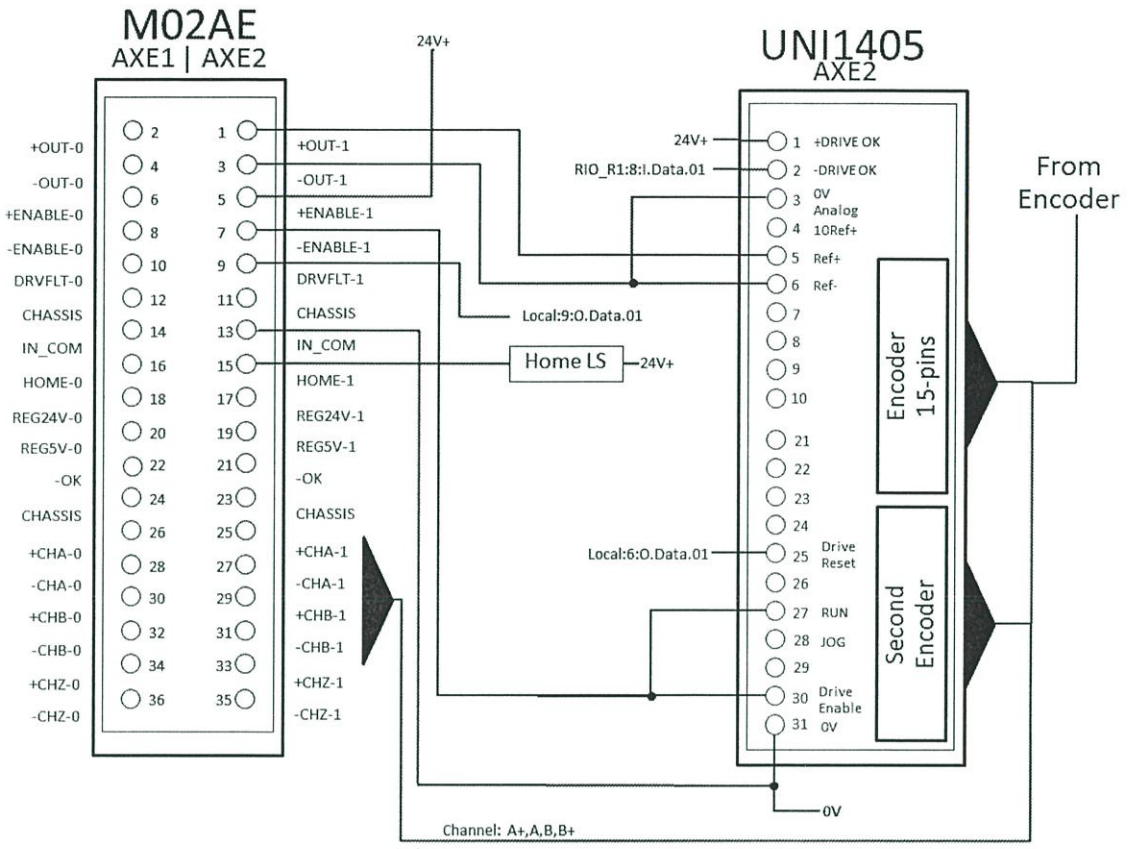
-เพิ่ม OUT+,OUT- ในการส่งสัญญาณไฟฟ้าไปที่ตัวขับมอเตอร์ (Motor Driver)

-นำ Registr+,Registr- ออก

-นำ Mark Encoder หรือ Channel Z ออก

-นำ Power และ Com ออกเพราะสามารถรับไฟได้จากราง

-เปลี่ยนขา Drive Enable เป็นรีเลย์ (Relay) คล้ายสวิตซ์ แทนการส่งสัญญาณไฟฟ้า



ภาพที่ 3.12 Loop Wiring M02AE ไป UNI1405 แกนที่ 2

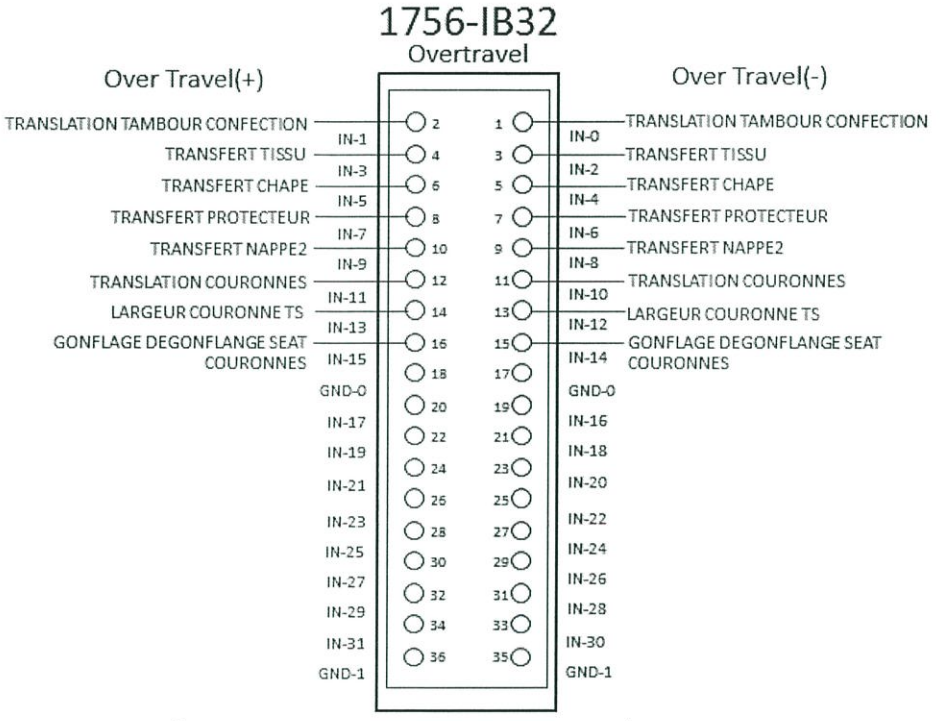
ในภาพที่ 3.12 คือตัวอย่าง Loop Wiring ของ M02AE ไป UNI1405 สิ่งที่เปลี่ยนจาก Wiring ของ IMCS ที่+OUT1 จะต่อกับ Ref+ และ –OUT1 จะต่อกับ Ref- แบบแอนาล็อกเซอร์โว, ENABLE นั้นจะทำเป็นสวิตช์เชื่อมจากไฟเลี้ยง (Supply) 24V ไปเข้าขาที่ 30 Drive Enable ของตัวขับ (Motor Driver) , Drive Fault จะรับค่าจากการ์ดเอาท์พุทเมื่อเกิดการผิดพลาด , Second encoder นั้นจะวิ่งจาก Encoder เข้าพอร์ท 15-pins จากนั้นจะขนานกันเข้า M02AE อีกหนึ่งต่อ

การ Wiring จะเป็นในลักษณะดังกล่าวจะแตกต่างกันที่ Drive Enable อยู่ในตางรางที่ 4 ตำแหน่งหน้าการ์ดแต่ละตัวจะไม่เหมือนกันที่ขา Drive Fault- , Drive OK - และ Drive Reset จะมีตารางให้เห็นในแต่ละแกน แต่ในหน้างานจริงนั้นทำการ Wiring แค่ในฝั่ง M02AE เท่านั้นอุปกรณ์ในกระบวนการ หรือ ตัวขับมอเตอร์(Motor Driver)

ตารางที่ 3.3 รายละเอียดของ Loop Wiring แต่ละแกน

แกน	M02AE	UNI DRIVE	
	Drive Fault+	Drive Ok-	Drive Reset
1	Local:O:9.Data.00	Local:O:8.Data.00	Local:O:6.Data.00
2	Local:O:9.Data.01	Local:O:8.Data.01	Local:O:6.Data.01
3	Local:O:9.Data.02	Local:O:8.Data.02	Local:O:6.Data.02
4	Local:O:9.Data.03	Local:O:8.Data.03	Local:O:6.Data.03
5	Local:O:9.Data.04	Local:O:8.Data.04	Local:O:6.Data.04
6	Local:O:9.Data.05	Local:O:8.Data.05	Local:O:6.Data.05
7	Local:O:9.Data.06	Local:O:8.Data.06	Local:O:6.Data.06
8	Local:O:9.Data.07	Local:O:8.Data.07	Local:O:6.Data.07
9	Local:O:9.Data.08	Local:O:8.Data.08	Local:O:6.Data.08
10	Local:O:9.Data.09	Local:O:8.Data.09	Local:O:6.Data.09
11	Local:O:9.Data.10	Local:O:8.Data.10	Local:O:6.Data.10
12	Local:O:9.Data.11	Local:O:8.Data.11	Local:O:6.Data.11
13	Local:O:9.Data.12	Local:O:8.Data.12	Local:O:6.Data.12
14	Local:O:9.Data.13	Local:O:8.Data.13	Local:O:6.Data.13

เมื่อไม่มีขา Over Travel limit Switch แล้วจะทำให้ระบบไม่สมบูรณ์จึงจำเป็นต้องเพิ่มการ์ด (Card) อินพุทอีกหนึ่งการ์ดเพื่อรับแกนละ 2 ขา จำนวน 8ชุด หรือ 16 จุดบนหน้าการ์ดที่อยู่ใน ราง (Chassis) RIO_R1 สล็อท (Slot)ที่ 16 ซึ่งในภาพที่ 3.13 เป็นชื่อแต่ละแกนที่นำมา Wiring ที่การ์ด 1756-IB32



ภาพที่ 3.13 Loop Wiring ของ Overtravel ที่การ์ด 1756-IB32

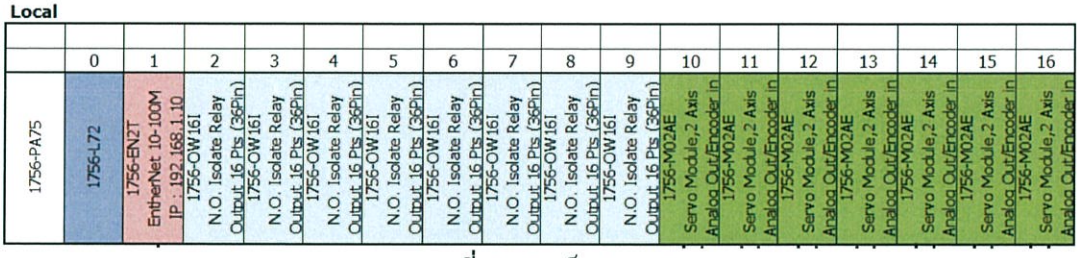
การเรียกใช้ตำแหน่งในอุปกรณ์ (Controller Tag) เพื่อเรียกดูบิทของ OverTravel มีรูปแบบเป็น RIO_R1:I:16.Data.XX โดย XX จะแทนด้วยเลขใน IN-XX เช่น TRANSFERT TISSU ขา + จะเรียกใช้ บิทที่ RIO_R1:I:16.Data.03

3.2.3 การติดตั้งอุปกรณ์และพีแอลซี

3.2.3.1 การติดตั้งพีแอลซี

ลักษณะการเรียงการ์ด (Card) ของพีแอลซีใน ControlLogix นั้นจะพยายามคงความเดิมของระบบเดิมใน PLC-5 ที่มี IMC-S อยู่ เพื่อลดปัญหาการผิดพลาดของแท็ก I/O ใน Terminal ของ Conversion Module ซึ่งจะกล่าวในหัวข้อต่อไป

การวางการ์ด (Card) เดิมนั้นจะแบ่งเป็น 3 แร็ค (Rack) Local ,RIO1 และ RIO2 โดย RIO นั้นเป็นชื่อของ Remote I/O และ Local เป็น แร็ค (Rack) ที่ติดกับซีพียูใน RIO1 นั้นจะเป็นอินพุททั้งหมด RIO2 เป็นเอาท์พุททั้งหมด

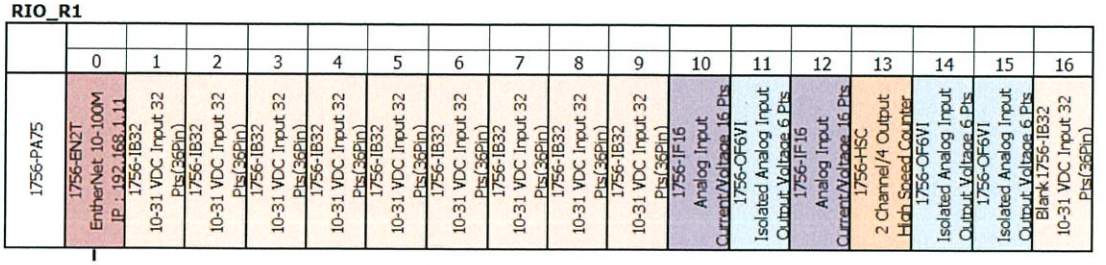


ภาพที่ 3.14 แร็ค Local

ในภาพที่ 3.14 เป็นภาพจำลองของ Local หรือแร็คที่ 0 ในทางซ้ายสุดจะเป็น Power supply ที่ติดมากับกับราง (Chassis) ถัดมาเป็นการ์ดซีพียู ถัดไปการ์ดเชื่อมต่อ (Communication) ถัดมาเป็นการ์ดเอาท์พุท 7 การ์ด และท้ายสุดเป็นการ์ด Analog Encoder จะมีรายละเอียดดังนี้

ตารางที่ 3.4 รายละเอียดของการ์ดในราง Local

การ์ดที่	ชื่อการ์ด	ประเภท
0	1756-L72	CPU
1	1756-EN2T	Ethernet 10-100m
2-9	1756-OW16I	N.O. Isolate Relay Output 16pts
10-16	1756-M02AE	Servo Module, 2 Axis Analog Encoder

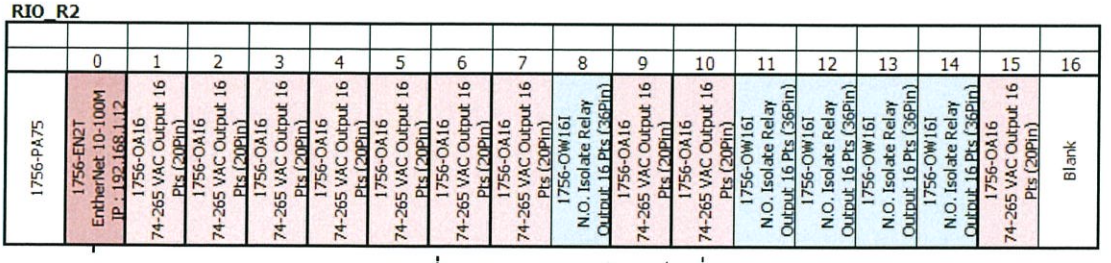


ภาพที่ 3.15 RIO1 หรือ แร็คที่ 1

ในภาพที่ 3.15 เป็นภาพจำลองของ RIO R1 หรือแร็คที่1 ในทางซ้ายสุดจะเป็น Power supply ที่ติดมากับกับราง(Chassis) ถัดมาเป็นการ์ดเชื่อมต่อ(Communication),การ์ดอินพุท 10 การ์ด, แอนาล็อกเซอร์โวอินพุทแบบ 16-point Single-End, แอนาล็อกเซอร์โวอินพุทแบบ 6-point, แอนาล็อกเซอร์โวอินพุทแบบ 16-point, HSC และ แอนาล็อกเซอร์โวอินพุทแบบ 6-point 3 การ์ด

ตารางที่ 3.5 รายละเอียดของการ์ดในราง RIO_R1

การ์ดที่	ชื่อการ์ด	ประเภท
0	1756-EN2T	Ethernet 10-100m
1-9,16	1756-IB32	10-31VDC Input 32pts
10,12	1756-IF16	. Isolate Analog Input 16pts
13	1756-HSC	High Speed Counter
11,14,15	1756-OF6VI	Isolate Analog Input 6pts



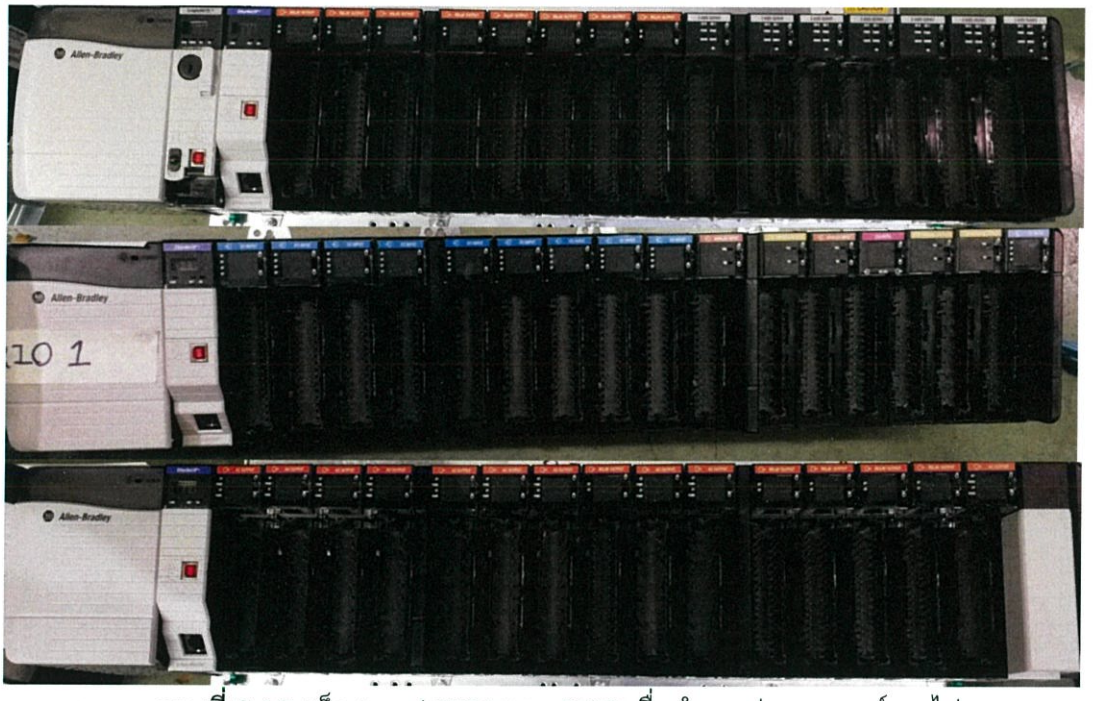
ภาพที่ 3.16 RIO2 หรือ แร็คที่ 2

ในภาพที่ 3.16 เป็นภาพจำลองของ RIO R2 หรือแร็คที่ 2 ในทางซ้ายสุดจะเป็น Power supply ที่ติดมากับกับราง(Chassis) ถัดมาเป็นการ์ดเชื่อมต่อ(Communication),และจะเป็นการ์ดเอาท์พุท 15 การ์ด

ตารางที่ 3.6 รายละเอียดของการ์ดในราง RIO_R2

การ์ดที่	ชื่อการ์ด	ประเภท
0	1756-EN2T	Ethernet 10-100m
1-7,9,10,15	1756-OA16	VAC Output 16pts
8,11-14	1756-OW16I	. N.O. Isolate Relay Output 16pts

เมื่อทำการประกอบแร็คทั้งหมดแล้วจะได้เช่นภาพที่ 3.17 ซึ่งยังไม่ได้เชื่อมกับเทอร์มินอล(Terminal) และ Conversion Module เรียงจากบนไปล่าง Local,RIO1 และ RIO2 ตามลำดับ



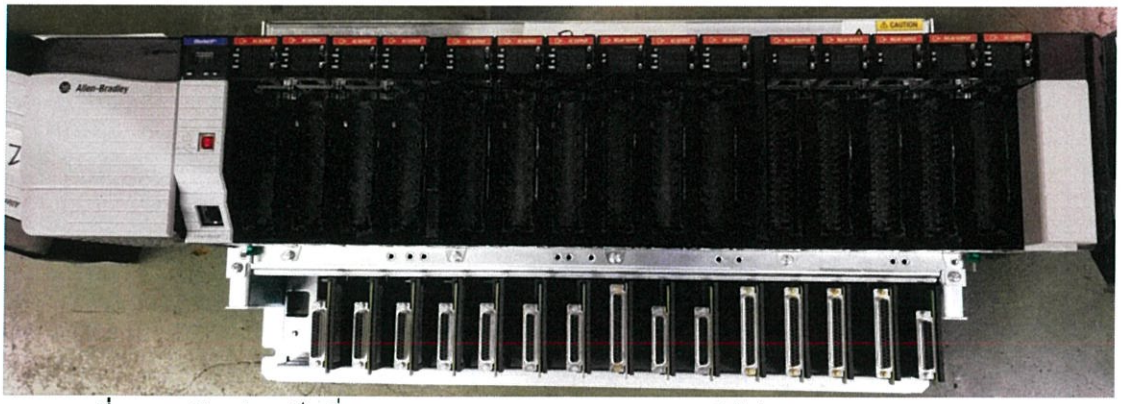
ภาพที่ 3.17 แร็ค Local,RIO1 และ RIO2 เมื่อทำการประกอบการ์ดลงไป

3.2.3.2 การติดตั้ง Conversion Module

ในการติดตั้งพีแอลซีของ ControlLogix นั้นจะใช้ Conversion Module หน้าที่เชื่อมจุดบนหน้าการ์ด(Card) อินพุท เอาท์พุทบน ControlLogix กับจุดบนเทอร์มินอล (Terminal) ของ PLC-5 ทำให้ไม่ต้องเสียเวลาในการต่อสาย(Hard-Wired) จุดต่อจุดใหม่และง่ายต่อการติดตั้ง แต่การเลือกใช้ Conversion Module นั้นจะขึ้นอยู่กับประเภทการ์ด(Card) ใน ControlLogix และยังมีสาย Pre-Wired เป็นตัวเชื่อมซึ่งขึ้นอยู่กับ Conversion Module มีรายละเอียดดังนี้ตารางที่ 3.7

ตารางที่ 3.7 แสดงรายละเอียด Conversion Module และ Pre-Wired Cable 1492 Series

ControlLogix	ประเภท	Conversion Module (1492-CM1771)	Pre-Wired Cable (1492-CONCAB)
1756-IB32	10-31 VDC Input 32 Pts	LD003	CONCAB005Z
1756-IF16	Analog Input 16 Pts Single-End	LA001	CONACAB005A
	Analog Input 16 Pts Differential	LA002	CONACAB005C
1756-OW16I	N.O. Isolate Relay Output 16 Pts	LD011	CONCAB005Y
1756-OF6VI	Isolated Analog Output 6 Pts	LD003	CONCAB005E
1756-OA16	74-265 VAC Output 16 Pts	LD006	CONCAB005X

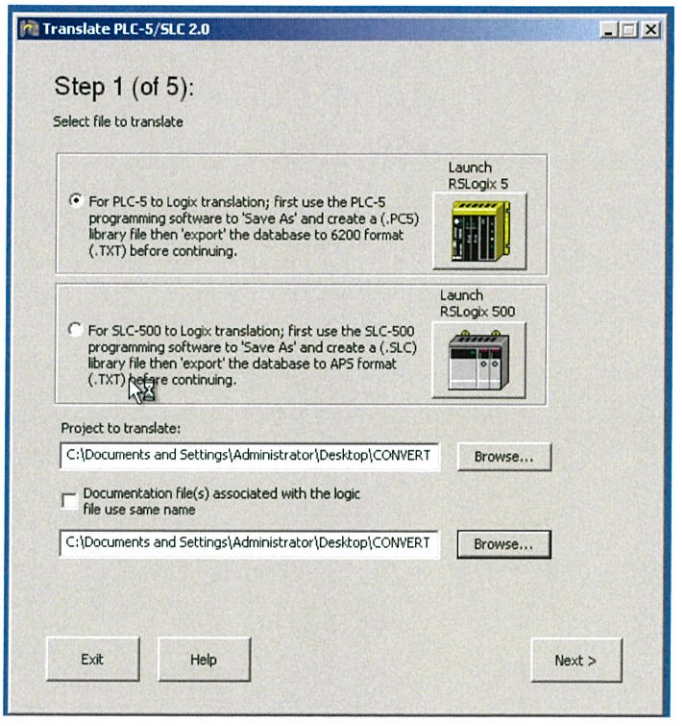


ภาพที่ 3.18 ตัวอย่างแร็คที่ประกอบ Conversion Module ยังไม่ประกอบ Pre-Wired Cable

3.2.4 การแปลงโปรแกรมลงใน Studio5000

3.2.4.1 การแปลงโปรแกรมจาก RsLogix5

โดยปกติแล้วการแปลงโปรแกรมเพื่อให้รองรับสำหรับ Studio5000 นั้นจะมีฟังก์ชันสำหรับการแปลงโปรแกรมแต่เนื่องจากความแตกต่างของโปรแกรม Rslogix5 และ Studio5000 มากจึงทำให้ต้องมีขึ้นตอนในการแปลงจาก Rslogix5 เป็น Rslogix5000 เสียก่อนแล้วจึงแปลงเป็น Studio5000

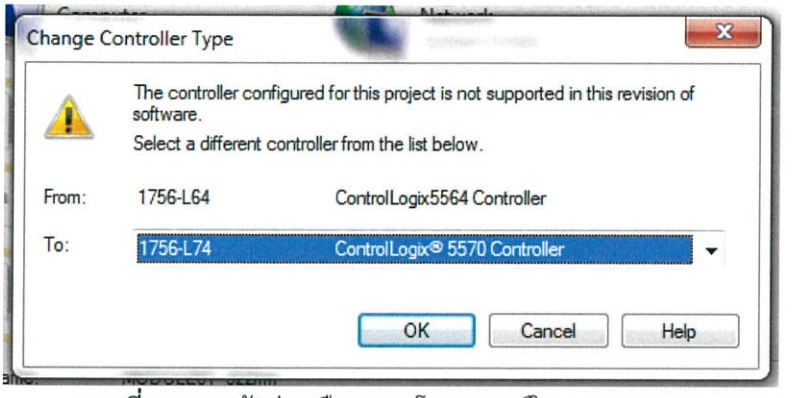


ภาพที่ 3.19 หน้าต่าง Translate PLC-5/SLC2.0

เริ่มแรกจะใช้โปรแกรม Translate PLC-5/SLC 2.0 ในการแปลงโปรแกมจาก RsLogix5 เป็น RsLogix5000 โดยที่การเปลี่ยนนั้นจะใช้ไฟล์นามสกุล Library File(*.PC5) และ RsLogix5(*.RSP) ดังในภาพ ซึ่งการใช้ไฟล์ทั้งสองนั้นมีความแตกต่างนามสกุลไฟล์ Library File(*.PC5) นี้จะเป็นไฟล์สำหรับเก็บข้ออาทิ คอมเม็นท์(Comment) , รายละเอียด(Description) ถ้าหากขาดไฟล์ประเภทไฟล์นี้ไปนั้นจะทำให้ข้อมูลบางส่วนไม่ครบหลังจากการแปลงโปรแกรม RsLogix5(*.RSP) นั้นเป็นไฟล์เก็บข้อมูลของคอนโทรลเลอร์ไม่ว่าจะเป็นเลดเดอร์(Ladder) , แท็ก(Tag) , หรือการตั้งค่าต่างๆ(Configuration)

หลังจากที่ทำการแปลงโปรแกรมมานั้นการ์ดเดิมจะไม่ถูกนำมาด้วยเนื่องจากการ์ดใน PLC-5 และ ControlLogix ไม่คล้ายกัน สิ่งที่ได้หลังจากการแปลงไฟล์นั้น เป็นนามสกุล RsLogix 5000 Export/Import File(*.l5k) ที่เป็นไฟล์เฉพาะสำหรับ RsLogix5000

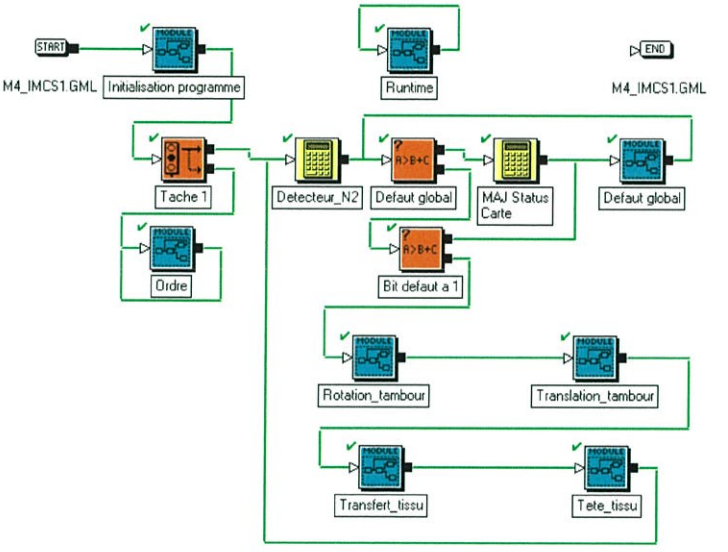
เมื่อได้ไฟล์นามสกุล RsLogix 5000 Export/Import File(*.l5k) มาในขั้นตอนต่อจะทำการแปลงเป็นไฟล์สำหรับ Studio5000 สิ่งที่แตกต่างระหว่างกัน คือ รุ่นของซอฟท์แวร์ (Revision) โดยรุ่นท้ายสุดที่ใช้งานได้ของ RsLogix5000 คือ 20.01.01 และในปัจจุบัน Studio5000 พัฒนาไปถึง รุ่นที่ 30 แล้ว เมื่อเปิดโปรแกรม Studio5000 แล้วทำการนำเข้า(Import) แล้วจะมีหน้าต่างให้เลือกคอนโทรเลอร์ที่ใช้ใน ControlLogix เป็นอันเสร็จสิ้นในการแปลง



ภาพที่ 3.20 หน้าต่างเลือกคอนโทรลเลอร์ใน Studio5000

3.2.4.2 การแปลงโปรแกรมจาก GML Commander

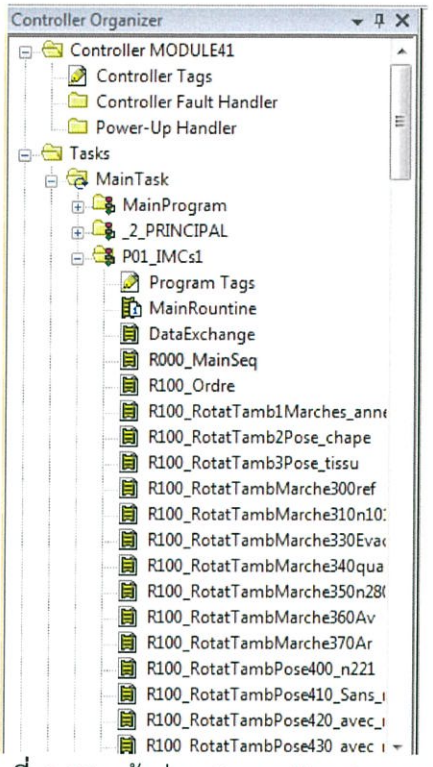
ในหัวข้อนี้จะเป็นการแปลงบล็อกฟังก์ชันใน GML Commander ในรูปแบบเลดเดอร์ จะนำมาแสดงเพียงบางกระบวนการเท่านั้น เนื่องจากในแต่ละกระบวนการจะใช้ลักษณะการทำงานของฟังก์ชันบล็อกคล้ายกันในส่วนของรายละเอียดของหน้าที่แต่ละฟังก์ชันบล็อกจะอยู่ในบทที่ 2 GML Commander ในส่วนของ GML Comander จะแยกเป็น 4 ไฟล์ตาม IMCS เดิม ในภาพที่ 3.21 เป็นการทำงานโดยรวมของ IMCS1



ภาพที่ 3.21 การทำงานของ IMCS1 ใน GML Commander

ภาพที่ 3.21 เริ่มทำงานที่ Start ต่อด้วยบล็อก Initialisation programme คือการสั่ง Intital ก่อนเริ่มกระบวนการเพื่อให้การเคลื่อนที่ของเครื่องจักรกลับสู่สภาวะเริ่มต้นรวมถึงโปรแกรมที่กลับไปสู่ Default ที่ตั้งไว้ ซึ่งกระบวนการเห็นได้ว่าที่ Tache1 คือเงื่อนไขแรกของการทำงาน ที่จะทำสองงานพร้อมๆกันเส้นด้านล่างคือ Ordre ที่เป็นกระบวนการผลิตของ IMCs1 และเส้นบนคือโปรแกรมการปรับค่าพารามิเตอร์(Parameter) ต่างๆตามค่าที่ผู้ใช้กำหนดที่หน้าจอ HMI ในส่วน Runtime นั้นคือโปรแกรมที่ทำงานร่วมกับ HMI

เมื่อนำมาเทียบกับเลดเดอร์(Ladder) ในโปรแกรมแล้วจะมีโฟลเดอร์โปรแกรมจะมี 6 ด้วยกัน คือ Main Program,_2_PRINICPAL,P01_IMCs1, P02_IMCs2, P03_IMCs3 และ P04_IMCs4 ใน 2 โฟลเดอร์โปรแกรมแรกคือ โปรแกรมพื้นฐานการดำเนินงานซึ่งโปรแกรมสามารถแปลงมาจาก RsLogix5 ได้เกือบสมบูรณ์ซึ่งทำให้สมบูรณ์โดยการแก้แท็กเดิมให้ตรงกับแท็กปัจจุบัน ส่วนที่สนใจคือส่วนที่จะทำการแปลงโฟลเดอร์โปรแกรมของ IMCs ที่มี 4 โฟลเดอร์โปรแกรม ดังภาพที่ 3.22 ในโฟลเดอร์โปรแกรมของ IMCs จะมีรูปแบบคล้ายกันหมดจึงทำการยกตัวอย่างเพียง 1 IMCs เท่านั้น

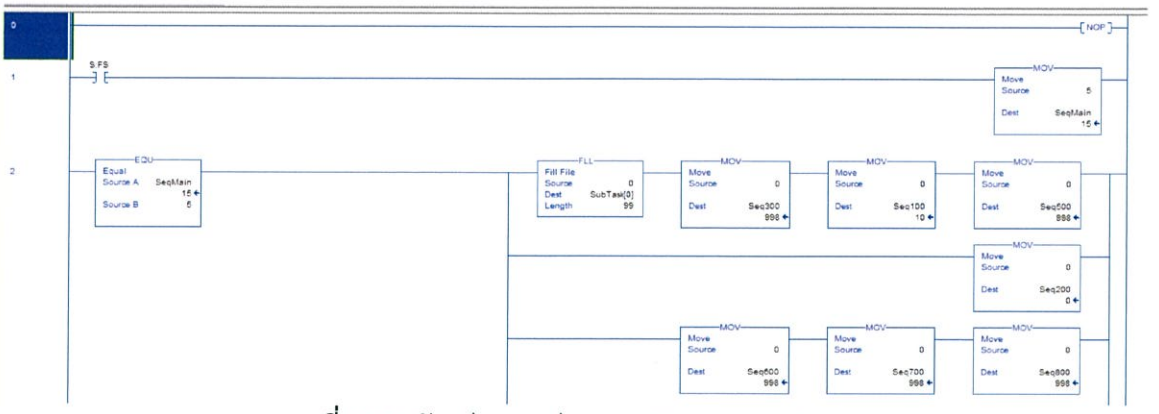


ภาพที่ 3.22 หน้าต่าง Controller Organizer

โฟลเดอร์โปรแกรม P01_IMCs1 มีการเรียกชื่อเลดเดอร์เพื่ออ้างอิงกับบล็อกหลักในภาพที่ 3.22 เป็น RX00_Name ซึ่ง X แทนเลขประจำบล็อกนั้นและ Name คือชื่อบล็อกย่อยในบล็อกหลัก

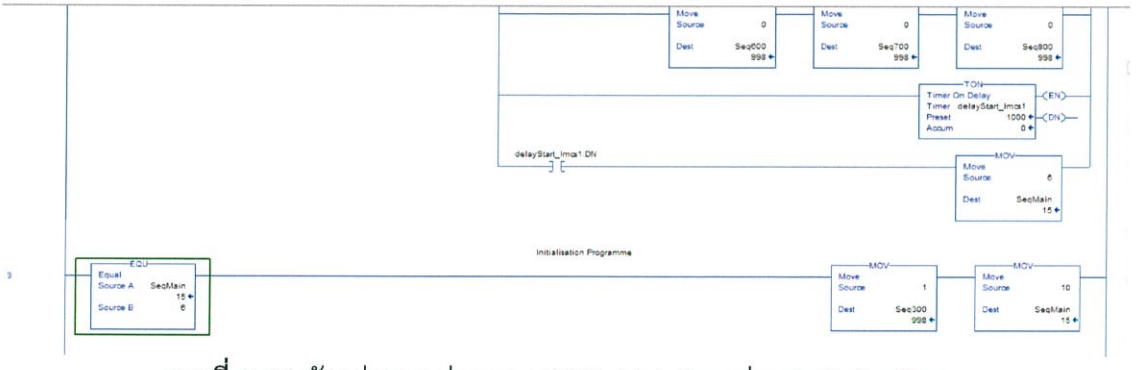
1. Data Exchange แทน การตั้งค่าพารามิเตอร์ของระบบ
2. R000 แทน MainSeq
3. R100 แทน Oredre
4. R200 แทน Default Global
5. R300 แทน Initialisation
6. R500 แทน Rotation Tambour
7. R600 แทน TransTambour
8. R700 แทน TransfertTissu
9. R800 แทน TeteTissu

ในแต่ละหัวข้ออาจมีหัวข้อย่อยอีก เช่น R100



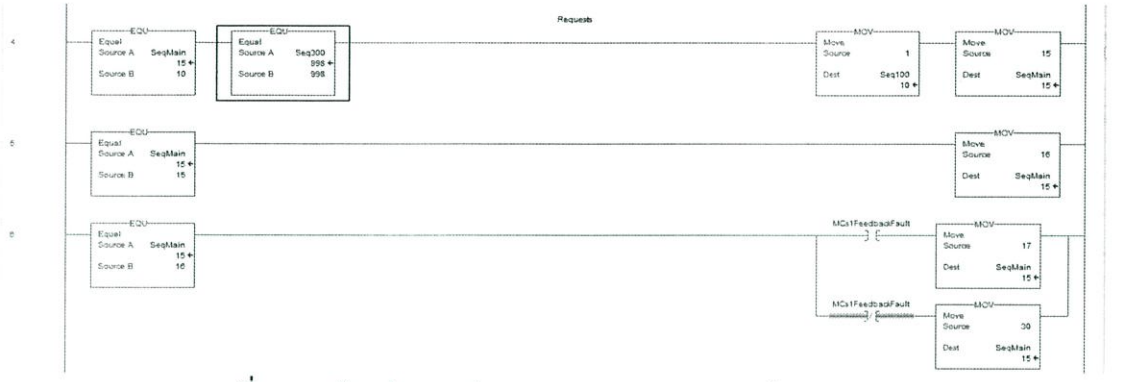
ภาพที่ 3.23 ตัวอย่างบางส่วนของ R000_MainSeq

ที่ R000_MainSeq เป็นเลดเดอร์หลักเพื่อกำหนดทิศทางการทำงานตามเส้นใน GML Commander ภาพที่ 3.23 ในบรรทัดที่1 First Scan แล้ว MOV ค่า 5 เข้า SeqMain ในบรรทัดที่ 2 การเทียบ(Equal) SeqMain เท่ากับ 5 หรือไหม ถ้าเท่ากับ 5 แล้วจะทำการใส่ 0 เข้าใน Seq100, Seq200, Seq300, Seq500, Seq600, Seq700,Seq800 และ SubTask[0] ถึง SubTask[98] จากชื่อที่กล่าวมานั้น คือ ตัวแปรที่สร้างขึ้นเพื่อนำทางคำสั่งไปที่เลดเดอร์ส่วนอื่น



ภาพที่ 3.24 ตัวอย่างบางส่วนของ R000_MainSeq ส่วน Initialisation

ภาพที่ 3.24 เมื่อทำบรรทัดที่ 2 เสร็จ SeqMain จะมีค่าเท่ากับ 6 ที่บรรทัดที่ 3 จะทำการเทียบ(Equal) SeqMain เท่ากับ 6 หรือไหม ถ้าเท่ากันจะส่ง ค่า 1 ไปที่ Seq300 ที่เลดเดอร์ R300_Initialisation และส่ง 10 ไปที่ SeqMain ที่เป็นตัวแปรนำทางของหน้านี้



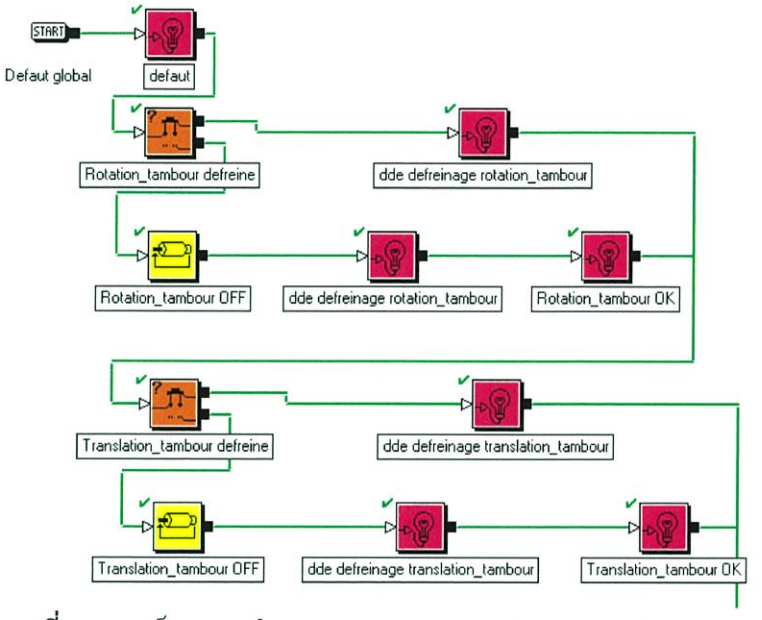
ภาพที่ 3.25 ตัวอย่างบางส่วนของ R000_MainSeq ส่วน Request

ภาพที่ 3.25 เมื่อทำบรรทัดที่ 3 เสร็จ SeqMain จะมีค่าเท่ากับ 10 ที่บรรทัดที่ 3 จะทำการเทียบ(Equal) SeqMain เท่ากับ 10 และ Seq300 เท่ากับ 998 หรือไหม ถ้าเท่ากันจะส่ง ค่า 1 ไปที่ Seq100 ที่เลดเดอร์R100_Ordre และส่ง 15 ไปที่ SeqMain ที่เป็นตัวแปรนำทางของหน้านี้ จะเห็นได้ว่าการตรวจสอบ Seq300 เท่ากับ 998 เพื่อดูการ Initialisation เสร็จหรือยังในหน้า R300_ Initialisation เมื่อเสร็จแล้วจะทำการ MOV ค่า 998 เข้า Seq300 ซึ่งสื่อว่าก่อนจะทำการสั่งเครื่องจักรทำงานหลังจากเปิดเครื่องนั้นต้องทำ Initialisation ก่อนทุกครั้งก่อนจะไปทำเงื่อนไขอื่นๆ ต่อไป

จุดสังเกตความสัมพันธ์ของการทำงานที่เป็นลำดับจะใช้ตัวแปรนำทางในการไปสั่งคำสั่งที่เลดเดอร์อื่นโดยการตรวจสอบ(Equal) เท่ากับเงื่อนไขหรือไหม เมื่อทำเสร็จบรรทัดนั้นๆจะ MOV ค่าลงเพื่อส่งต่อไปส่วนที่ต้องการจะทำแบบนี้ในทุกเลดเดอร์

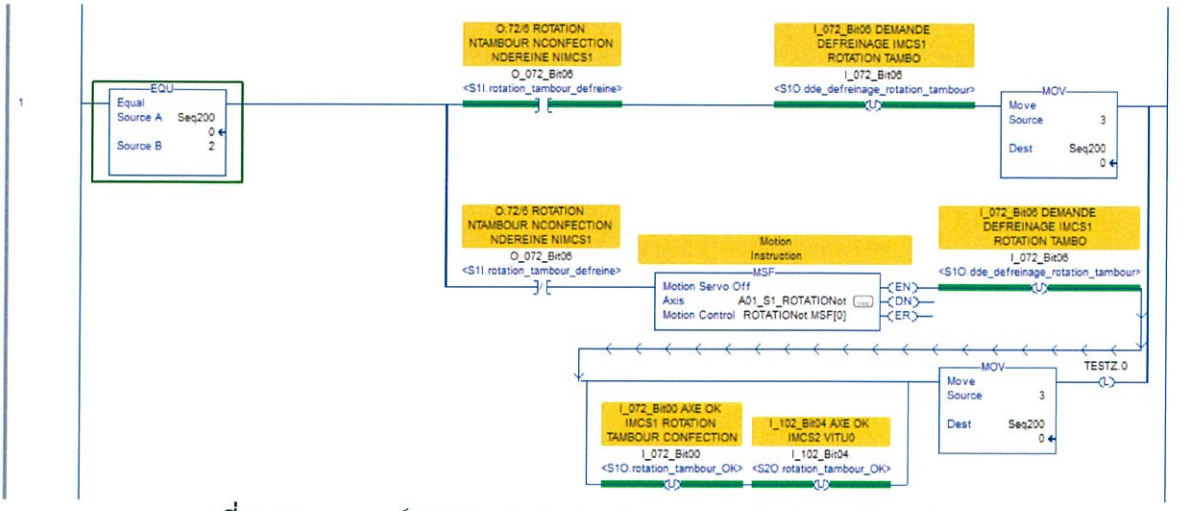
การแปลงเลดเดอร์ถึงแม้คำสั่งทาง Motion ในStudio5000 คล้ายกับคำสั่งในGML Commander แต่ไม่มีฟังก์ชันในการแปลงจึงทำให้การแปลงต้องอาศัยการแปลงโดยเทียบโปรแกรมดูการทำงานใน GML Commander จะยกตัวอย่างมาพอสังเขป

ตัวอย่างที่ 1 Default Global ของ IMCs1



ภาพที่ 3.26 บล็อกการทำงาน GML Commader ของ Default Global

ภาพที่ 3.26 แสดงการทำงานของบล็อกโดยรวม Default Global อธิบายดังนี้ เริ่ม Start สั่ง Output ชื่อ Deafult ทำงานต่อไปจะรับค่าจาก RIO Adapter ด้วยเงื่อนไข if



ภาพที่ 3.27 เลดเดอร์ R200_ Default Global ของ Default Global

ในบรรทัดที่ 1 การตรวจสอบ Seq200 เพื่อดูเงื่อนไขการต่อเนื่องจากบรรทัดที่ 0 จากนั้นจะใส่ Interlock ของ <S1I.rotation_tambour_defreine> จะได้ 2 กรณีคือ

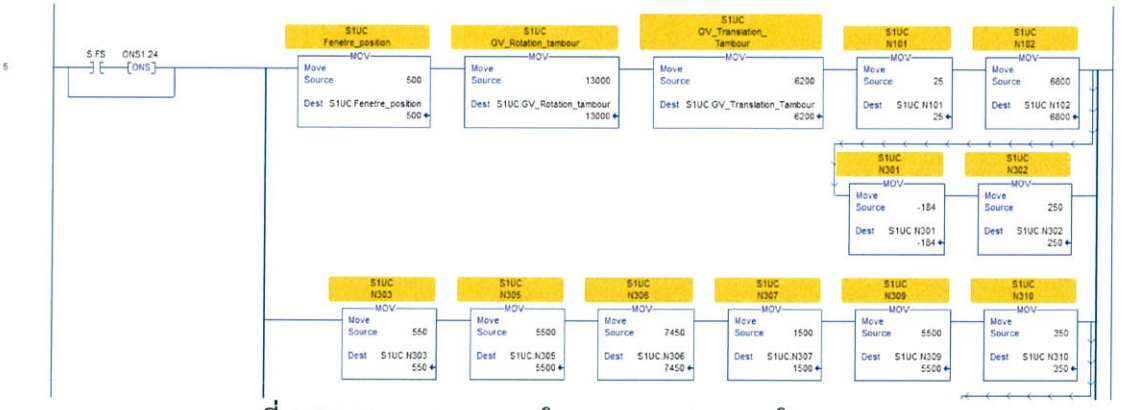
1. กรณี Normally Open จะสั่ง OFF ที่ <S1O.dde defreinage rotation_tambour>
2. กรณี Normally Close จะสั่ง Motion Servo Off แกนที่ 1 ROTATION TAMBOUR CONFECTION และสั่ง OFF ที่ <S1O.dde defreinage rotation_tambour> , <S1O.Rotation_Tambour_OK> และ <S2O.Rotation_Tambour_OK>

แล้วจึงส่งต่อไปบล็อกถัดไปโดย MOV ค่า 3 เพื่อให้บล็อกถัดไปตรวจสอบ เมื่อเทียบกับโปรแกรมใน GML Commander เลดเดอร์ที่กล่าวไปคือฟังก์ชันบล็อก Input แบบ if นั่นเอง จากที่กล่าวมานั้น การแปลงจะใช้การดูลำดับ(Sequence)บล็อกฟังก์ชันและเงื่อนไขของโปรแกรม GML Commander เป็นหลักแล้วจึงสร้างเงื่อนไขการทำงานเฉพาะในเลดเดอร์ นอกจากนี้ในแกน IMCs อื่นจะใช้วิธีการเดียวกัน ไม่ว่าจะเป็นรูปแบบชื่อ ตัวแปรนำทาง หรือ ชื่อเลดเดอร์

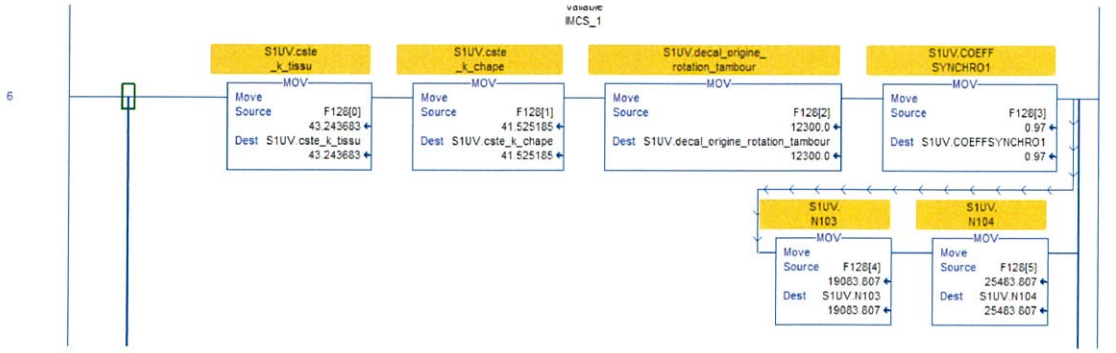
3.2.4.3 การจัดการข้อมูล User Constant และ User Variable

ลักษณะของ User Constant โดยเดิมแล้วในขณะที่ยังอยู่ใน GML Commander เป็นค่าคงที่ฝังตัวลงไปในโปรแกรมเพื่อใช้ในการกำหนดค่า โดยจะสามารถแก้ไขได้ผ่านทางเงื่อนไขในโปรแกรม เช่น ในโหมดผลิตยางรูปแบบ A ใช้ความกว้างหน้า X และ ในโหมดผลิตยางรูปแบบ B ใช้ความกว้างหน้า Y ซึ่งส่งผลทำให้การทำงานของมอเตอร์มีการเปลี่ยนแปลง ดังนั้นจำเป็นต้องแก้ค่า User Constant เพื่อสอดคล้องกัน ในทางเดียวกันลักษณะของ User Variable ใน GML Commander เมื่อเปรียบเทียบกับภาษาเลดเดอร์แล้วคล้ายกับบิทความจำ(Memory Bit) ซึ่ง User Variable จะมีค่าเฉพาะตัวเป็น Address ที่โปรแกรมจะเรียกใช้ แต่เมื่อนำมาใช้ใน ControlLogix แล้วจึงจำเป็นต้องผูก Address กับแท็กใน ControlLogix แทน

การนำเข้าของ User Constant และ User Variable มาใน ControlLogix จำเป็นต้องสร้าง User Define กล่าวถึงในหัวข้อถัดไป ขั้นตอนนี้เป็นการนำค่ามาไว้ใน User Define



ภาพที่ 3.28 User Constant ใน Data Exchange ใน IMCs1

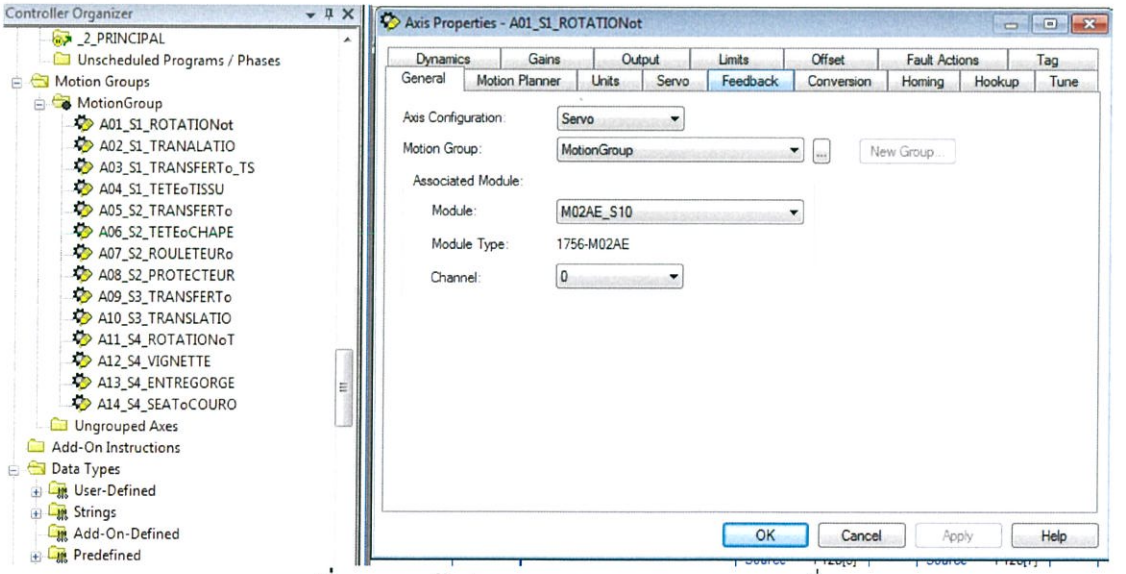


ภาพที่ 3.29 User Varaible ใน Data Exchange ใน IMCs1

ภาพที่ 3.28 และภาพที่ 3.29 การใส่ค่าเข้าไปในแท็กที่สร้างขึ้นจะใช้คำสั่ง MOV ไปปลายทาง(Dest) ในขั้นตอนนี้ต้องทำทุกค่าในทุก IMCs

3.2.5 การตั้งค่า Motion Group ในแต่ละแกน

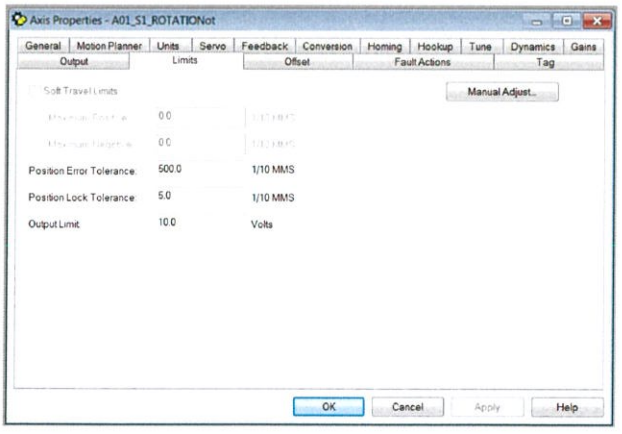
Motion Group ถูกสร้างขึ้นมาจากการ์ด M02AE แล้วทำการสร้างแกน ทั้งหมด 14 แกนเพื่อให้โปรแกรมรับทราบว่ามีการใช้งาน Motion Control นอกจากนี้ยังเพื่อตั้งค่ารูปแบบการทำงานของ Motion Control ที่ Axis Properties



ภาพที่ 3.30 หน้าต่าง Axis Properties ของแกนที่ 1

ในส่วนของการตั้งค่าสิ่งที่สนใจเป็นพิเศษเพราะ เป็นค่าพารามิเตอร์ของการเคลื่อนไหวมีดังนี้

General	> Axis Configuration: Servo
Units	> Position Unit (1/10 MMS)
	> Time Base (Seconds)
Output	> Velocity Scaling: %(1/10 MMS/s)
Limit	> Soft Travel
	> Position Error Tolerance (1/10 MMS)
	> Position Lock Tolerance (1/10 MMS)
Dynamics	> Maximum Speed (1/10 MMS/s)
	> Maximum Acceleration (1/10 MMS/s^2)
	> Maximum Deceleration (1/10 MMS/s^2)
Homing	> Mode
	> Position (1/10 MMS)
	> Sequence
	> Direction (Active Home Sequence Group)



ภาพที่ 3.31 หน้าต่าง Axis Properties ของแกนที่ 1 การตั้งค่า Limit

ภาพที่ 3.31 ในส่วนหน้าต่างของ Limit เป็นการตั้งค่า Soft Travel Limit ที่เป็นการจำกัดการเคลื่อนที่ด้วยซอฟท์แวร์โดยกำหนดระยะทางที่อ้างอิงจาก Encoder หัวข้อ Position Error Tolerance คือ ค่าความเผื่อสำหรับตำแหน่งในย่าน Error , Position Lock Tolerance คือค่าความเผื่อสำหรับตำแหน่งจริงที่เกิดขึ้น และ Output Limit คือ การจำกัดสัญญาณคำสั่งแรงดันไฟฟ้า (โวล์ต)

3.2.6 การใช้ Addressing และ Tagging ลง Studio5000

การแปลงแท็ก(Tag)นั้นเพื่อให้คอนโทรลเลอร์สามารถเชื่อมโยงระหว่างโปรแกรมและหน้าการ์ดได้ ก่อนที่จะสร้างแท็กจำเป็นต้องรู้ว่าแท็กใดที่จำเป็นต้องสร้าง ในส่วนนี้ได้ทำการนำออก(Export)ไฟล์ทั้งหมดของ RsLogix5 เป็น Studio5000 สามารถนำออกได้ไม่ว่าจะเป็นเลดเดอร์(Ladder),แท็ก หรือ คอมเม้นท์ เป็นต้น สิ่งที่เป็นปัญหาแท็กที่เชื่อมโยงกับ GML Commander นั้นจะไม่ถูกใช้อีกต่อไปเพราะจะไม่ใช้ IMCS อีกจึงเป็นสาเหตุทำให้แปลงแท็ก GML Commander ไปในรูปแท็กใหม่ในพีแอลซีก่อน ในขณะเดียวกันเลดเดอร์เดิมที่ได้แปลงมานั้นยังมีการเรียกใช้แท็กใน GML Commander เดิมแต่แท็กนั้นจะเป็นแค่ชื่อที่ไม่ได้เชื่อมโยงกับอะไรไว้ จึงจำเป็นต้องสร้างแท็กใหม่ใน Studio5000 ที่เชื่อมกับพีแอลซีไว้ แล้วจึงนำไปเชื่อมกับแท็กเดิม วิธีนี้ลดความยุ่งยากในการแก้ไขแท็กทุกตัวในเลดเดอร์ให้เป็นแท็กใหม่ ลดความผิดพลาด

3.2.6.1 แปลงแท็กจาก RsLogix5

ส่วนของแท็ก RsLogix5 จะมีลักษณะเฉพาะกล่าวไว้ในบทที่ 2 การ Addressing และ Tagging พีแอลซีเดิมใช้การ Address แบบ 8-bits เลดเดอร์เรียกใช้ตามพีแอลซีเดิม แต่เมื่อทำการติดตั้งการ์ดใหม่การเรียก Address แบบเดิมทำให้โปรแกรมผิดพลาด หลักการคือไม่แก้ไขแท็กที่เลดเดอร์แต่ใช้การ Alias for ไปที่แทกที่สร้างขึ้นใหม่

I_000_Bit00	RIO_R1:1:I.Data.0
I_000_Bit01	RIO_R1:1:I.Data.1
I_000_Bit02	RIO_R1:1:I.Data.2
I_000_Bit03	RIO_R1:1:I.Data.3
I_000_Bit04	RIO_R1:1:I.Data.4
I_000_Bit05	RIO_R1:1:I.Data.5
I_000_Bit06	RIO_R1:1:I.Data.6
I_000_Bit07	RIO_R1:1:I.Data.7
I_000_Bit10	RIO_R1:1:I.Data.8
I_000_Bit11	RIO_R1:1:I.Data.9
I_000_Bit12	RIO_R1:1:I.Data.10
I_000_Bit13	RIO_R1:1:I.Data.11
I_000_Bit14	RIO_R1:1:I.Data.12
I_000_Bit15	RIO_R1:1:I.Data.13
I_000_Bit16	RIO_R1:1:I.Data.14
I_000_Bit17	RIO_R1:1:I.Data.15

ภาพที่ 3.32 Address และ Alias for ของ RIO_R1 อินพุท

ทางซ้ายมือในภาพที่ 3.32 เป็นแท็กที่แปลงมาจาก RsLogix5 อีกทั้งยังเป็นแท็กที่อยู่ในเลดเดอร์ ทางขวามือเป็น Alias for สาเหตุของการตั้งชื่อดังนี้ เพราะเมื่อทำการเพิ่มการ์ดในพีแอลซี ControlLogix โปรแกรมจะสร้าง Controller Tag ให้โดยอัตโนมัติจึงทำการเชื่อมแท็กด้วยการสร้างชื่อเป็นชื่อเดียวกับ Controller Tag จุดสังเกตอีกอย่างหนึ่งการ์ดเก่าและการ์ดใหม่มีขนาด 32 จุดเท่ากันแต่ใน PLC-5 มีการเรียกกรุ๊ปเรียกแร็ค ในControlLogix ไม่มีการเรียกกรุ๊ป เรียกตามหน้าการ์ดจริง ดังนั้นการแปลงแท็กจะนับเทียบหน้าการ์ดจริงว่า ชื่อแท็กใดๆใน RsLogix5 ตรงกับหน้าการ์ดใน PLC-5 ใดแล้วตำแหน่งหน้าการ์ดของ ControlLogix เท่าใดเช่น RIO_R1:1.I.Data.27 มีความหมายว่าเราใช้ชื่อ

รางนี้ว่า RIO_R1 โดยมีการ์ดเชื่อมต่อ(Commuinication) EN2T เป็นตัวระบุของรางนี้ เลขหนึ่งแทนสล็อทที่ I แทนการ์ดประเภทอินพุท และ Data.27 แสดงตำแหน่งบนหน้าการ์ด

	Name	Alias For	Base Tag
	I_001_Bit13	RIO_R1:1:I.Data.27	RIO_R1:I.Slot[1].Data.27
	I_001_Bit14	RIO_R1:1:I.Data.28	RIO_R1:I.Slot[1].Data.28
	I_001_Bit15	RIO_R1:1:I.Data.29	RIO_R1:I.Slot[1].Data.29
	I_001_Bit16	RIO_R1:1:I.Data.30	RIO_R1:I.Slot[1].Data.30
	I_001_Bit17	RIO_R1:1:I.Data.31	RIO_R1:I.Slot[1].Data.31
	I_002_Bit01	RIO_R1.2:I.Data.1	RIO_R1:I.Slot[2].Data.1
	I_002_Bit02	RIO_R1.2:I.Data.2	RIO_R1:I.Slot[2].Data.2
	I_002_Bit03	RIO_R1.2:I.Data.3	RIO_R1:I.Slot[2].Data.3
	I_002_Bit04	RIO_R1.2:I.Data.4	RIO_R1:I.Slot[2].Data.4
	I_002_Bit05	RIO_R1.2:I.Data.5	RIO_R1:I.Slot[2].Data.5
	I_002_Bit06	RIO_R1.2:I.Data.6	RIO_R1:I.Slot[2].Data.6
	I_002_Bit07	RIO_R1.2:I.Data.7	RIO_R1:I.Slot[2].Data.7
	I_002_Bit10	RIO_R1.2:I.Data.8	RIO_R1:I.Slot[2].Data.8
	I_002_Bit11	RIO_R1.2:I.Data.9	RIO_R1:I.Slot[2].Data.9
	I_002_Bit12	RIO_R1.2:I.Data.10	RIO_R1:I.Slot[2].Data.10
	I_002_Bit13	RIO_R1.2:I.Data.11	RIO_R1:I.Slot[2].Data.11
	I_002_Bit15	RIO_R1.2:I.Data.13	RIO_R1:I.Slot[2].Data.13
	I_003_Bit00	RIO_R1.2:I.Data.16	RIO_R1:I.Slot[2].Data.16
	I_003_Bit01	RIO_R1.2:I.Data.17	RIO_R1:I.Slot[2].Data.17
	I_003_Bit02	RIO_R1.2:I.Data.18	RIO_R1:I.Slot[2].Data.18

ภาพที่ 3.33 การเชื่อม Name, Alias For และ Base tag

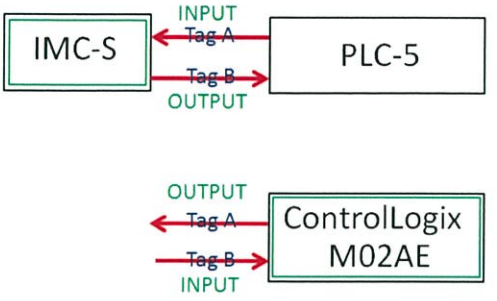
ภาพที่ 3.33 แสดง Base tag คือแท็กที่ถูก Alias for ซึ่ง base tag ในรูปคือแท็กที่สร้างขึ้นจากการเพิ่มการ์ด

3.2.6.2 แปลงแท็กจาก GML Commander

เพื่อไม่ให้เกิดปัญหาในการเคลื่อนย้ายแท็กหาย หรือแท็กไม่สมบูรณ์การย้ายแท็กของ Motion Control ในโปรแกรม GML Commander จะไม่มีการเปลี่ยนใดๆในโปรแกรมเดิมเพียงแต่ทำไฟล์ขึ้นมาเพิ่มเพื่อใส่รายละเอียดการเชื่อมแท็กเข้าด้วยกัน รายละเอียดแท็กมีดังนี้

1. RIO Input
2. RIO Output
3. User Constant
4. User Variable
5. Data Bits

ใน 1 ไฟล์ GML Commander นั้นคือหนึ่งโปรแกรมสำหรับ IMC-S ดังนั้นมี 14 แกน จะมีไฟล์ทั้งหมด 4 ไฟล์ในตารางที่ 3.8 ซึ่งจะไม่สามารถทำงานข้ามไฟล์ได้ 1 IMCS ต่อ 1 ไฟล์ใน GML Commander นอกจากนี้ยังมีข้อสังเกตอีกหนึ่งจุด



ภาพที่ 3.34 ภาพการแลกเปลี่ยนแท็กระบบใหม่และระบบเก่า

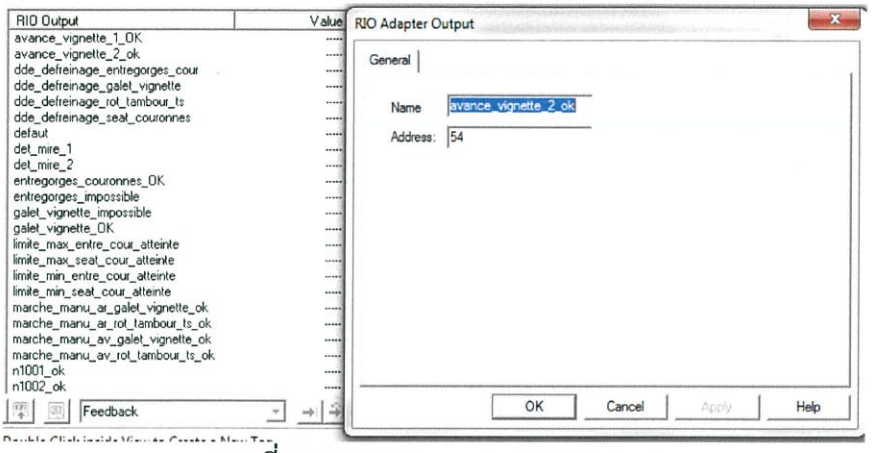
รูปข้างบนภาพที่ 3.34 IMC-S จะรับ Tag A มาจาก PLC-5 ในขณะเดียวกันจะส่ง Tag B ไปที่ PLC-5 เป็นการแลกเปลี่ยนแท็กของระบบเก่า แต่ในระบบใหม่ตัวควบคุมตำแหน่ง(Motion Control) M02AE นั้นอาศัยการประมวลผลร่วมกับ CPU ของ ControlLogix ต่างจากระบบเก่าที่ IMC-S ที่สามารถทำงานเพียงตัวเดียวได้(Stand Alone) ทำให้การเรียกแท็กจะสลับกัน

1. Tag A = Input(เก่า) = Output(ใหม่)
2. Tag B = Output(เก่า) = Input(ใหม่)

ตารางที่ 3.8 รายละเอียดของแต่ละแกนใน GML Commander

IMC-S 1	ROTATION TAMBOUR CONFECTION TRANSLATION TAMBOUR CONFECTION TRANSFERT TISSU TETE TISSU
IMC-S 2	TRANSFERT CHAPE TETE CHAPE ROULETEUR CHAPE TRANSFERT PROTECTEUR
IMC-S 3	TRANSFERT NAPPE2 TRANSLATION COURONNES
IMC-S 4	ROTATION TAMBOUR TS HAUTEUR POSTE TS LARGEUR COURONNE TS GONFLAGE DEGONFLANGE SEAT COURONNES

ในส่วนที่กำลังจะแปลงนั้นคือ RIO Adapter แท็กหลักๆข้างในคือ Input และ Output ที่ใช้ผ่าน Discrete I/O Device



ภาพที่ 3.35 Address ของ RIO Output

ภาพที่ 3.35 หน้าต่าง RIO Adapter Output สิ่งที่สำคัญคือ Address เนื่องจากการอ้างอิงของ IMCS อ้างอิงเป็น Address เลขตั้งแต่ 0-99 เท่ากับว่าสามารถมีได้เพียง 100 ตัวเท่านั้นซึ่ง Address จะสอดคล้องกับ แท็กในพีแอลซี ในโครงงานนี้เป็นแบบ Full-Rack

User-Defined Discrete Outputs	**Correspond to PLC Inputs in I/O Group...**			
0 – 3 *	1 3 5 7	1 3 5	1 3	1
4 – 19	Not Available	2 4 6	2 4	2
20 – 35	Not Available	3 5 7	3 5	3
36 – 51	Not Available	N/A	4 6	4
52 – 67	Not Available	N/A	5 7	5
68 – 83	Not Available	N/A	N/A	6
84 - 99	Not Available	N/A	N/A	7
Starting Group	0 2 4 6	0 2 4	0 2	0
Rack Size	1/4	1/2	3/4	Full

ภาพที่ 3.36 เทียบ Address กับแท็กในพีแอลซี ของ RIO Adapter

ถ้าต้องการหาแท็กในพีแอลซีใดๆ เทียบตารางแล้วหาค่าแล้วนำมาใส่ในรูปแบบ I_XXY_BitZZ โดย XX เป็นแร็ค (Rack)ใดๆ Y เป็นกรุ๊ป (Group)ใดๆ และ Z เป็นบิท (Bit)ใดๆ ตารางดังกล่าวในภาพที่ 3.36 สามารถใช้ได้ทั้งอินพุทและเอาท์พุท

ตัวอย่างที่1

สร้างแท็กพีแอลซีจาก GML Commander โดยเป็นเอาท์พุทอยู่ IMCSที่ 4 ชื่อ n1001_ok ซึ่งอยู่ Address ที่ 37

ประเภท I จากที่กล่าวในข้างต้นอินพุทและเอาท์พุทจะสลับกันเมื่อแปลง Address

แร็คที่ 12 เนื่องจากอยู่ IMCS ที่4 ซึ่งได้กล่าวไว้ในหัวข้อ การ Addressing และ Tagging

กรุ๊ปที่ 4 เนื่องจาก 37 อยู่ใน 36-51

บิทที่ 01 เนื่องจาก ตัวที่ 2 ของ 36-51 แต่การนับแบบฐาน 8 ที่นับเลข 0 ด้วยจึงเป็น 1 ดังนั้นเขียนได้เป็น I_124_Bit01

ตัวอย่างที่2

สร้างแท็กพีแอลซีจาก GML Commander โดยเป็นอินพุทอยู่ IMCSที่1 ชื่อ arret_immediat ซึ่งอยู่ Address ที่ 35

ประเภท O จากที่กล่าวในข้างต้นอินพุทและเอาท์พุทจะสลับกันเมื่อแปลง Address

แร็คที่ 7 เนื่องจากอยู่ IMCS ที่1 ซึ่งได้กล่าวไว้ในหัวข้อ การ Addressing และ Tagging

กรุ๊ปที่ 3 เนื่องจาก 35 อยู่ใน 20-35

บิทที่ 17 เนื่องจาก ตัวสุดท้ายของ 20-35 แต่ด้วยการนับแบบเลขฐาน 8 ที่นับเลข 0 ด้วยจึง ดังนั้นเขียนได้เป็น I_073_Bit17

Scope: MODULE41 Show: All Tags

Name	Alias For
I_076_Bit05	S1O.position_attente_coupe_n311_ok
I_076_Bit07	ArmOutputCam[10].2
I_076_Bit10	ArmOutputCam[10].3
I_076_Bit11	ArmOutputCam[10].7
I_076_Bit12	ArmOutputCam[10].4
I_076_Bit13	ArmOutputCam[10].8
I_076_Bit14	ArmOutputCam[10].5
I_076_Bit15	ArmOutputCam[10].6
I_077_Bit00	S1O.reference_tete_tissu_ok
I_077_Bit11	AENT01:0:I.0
I_077_Bit12	S1O.Defaut_pre_nappe2
I_077_Bit13	S1O.Defaut_abs_nappe2
I_101_Bit10	I[65].8
I_101_Bit14	S2O.defaut
I_102_Bit00	S2O.transfert_chape_OK
I_102_Bit01	S2O.tete_chape_OK
I_102_Bit02	S2O.rouleteur_OK

ภาพที่ 3.37 บิทของ IMCS ที่ 1 และ 2 ใน Studio5000

จากขั้นตอนข้างต้นถือเป็นการเตรียมแท็กเพื่อให้รู้ว่า I_XXY_BitZZ เชื่อมกับแท็กใดใน GML Commander เมื่อต้องการจะสร้างแท็กที่ใช้งานได้จริง จะสร้าง User Define ที่เป็นอินพุทหรือ

เอาท์พุทเดิมใน GML Commander เพื่อกำหนดค่าให้โปรแกรมรู้แล้วทำการเชื่อมกับแท็ก I_XXY_BitZZ โดย Alias for ไปที่ User Define เมื่อทำเสร็จโปรแกรมจะรู้จักกับแท็กที่เคยอยู่ใน GML Commander เช่น ภาพที่ 3.37 I_102_Bit01 จะเชื่อมกับ S2O.tete_Chape_OK การใช้ชื่อ S2O. ด้านหน้าสื่อถึงชื่อที่ตั้ง ถัดมาเป็นเป็นอาเรย์ (Array)ของ User Define ชื่อที่ความหมายดังนี้

S1O. คือ IMCS ที่ 1 Rio Adapter Output

S1I. คือ IMCS ที่ 1 Rio Adapter Input

S2O. คือ IMCS ที่ 2 Rio Adapter Output

S2I. คือ IMCS ที่ 2 Rio Adapter Input

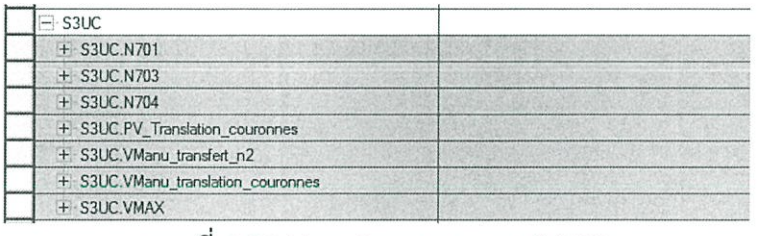
S3O. คือ IMCS ที่ 3 Rio Adapter Output

S3I. คือ IMCS ที่ 3 Rio Adapter Input

S4O. คือ IMCS ที่ 4 Rio Adapter Output

S4I. คือ IMCS ที่ 4 Rio Adapter Input

การย้าย User Constant และ User Variable ทั้ง 2 ค่าจะคล้ายกับ RIO Input/Output แตกต่างที่ไม่มีการผูกค่าไว้กับ address ในพีแอลซีแต่จะเป็นเพียงค่าเมมโมรี่ภายใน ดังนั้นขั้นตอนการทำจะไม่ทำการแปลง Address ใน GML Commander ทำการสร้าง User Define แล้วสร้าง Array ขึ้นมาใช้แทนภาพที่ 3.38 คือตัวอย่างของ User Constant



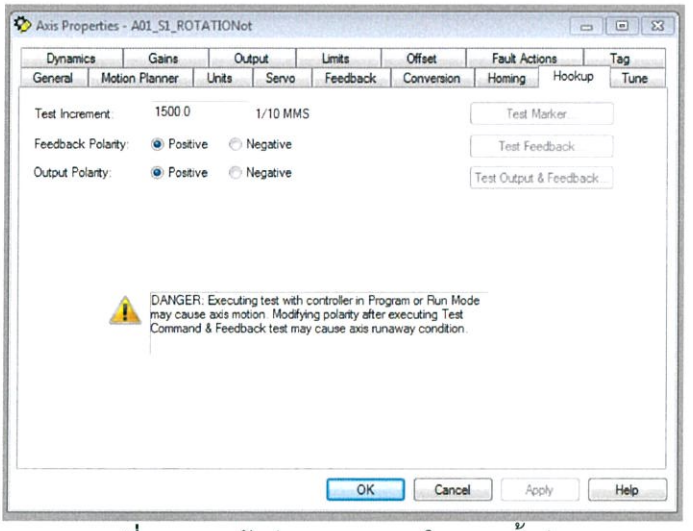
ภาพที่ 3.38 User Constant ของ IMCS3

3.2.7 การ Hook up , Tune , และ Motion Direct Commands

การทำงานของมอเตอร์เป็นอุปกรณ์ชิ้นสุดท้ายของการควบคุมหรืออุปกรณ์ที่สร้างเอาท์พุทออกมาจากกระบวนการคำณวนทั้งหมดจึงมี Encoder ขึ้นมาเพื่อตรวจสอบการทำงานของกระบวนการควบคุมเช่นในภาพที่ 3.2 ซึ่งการทำงานของมอเตอร์ทางกายภาพ (Mechanical) และทางฟ้า (Electrical) ควรสอดคล้องไปในทางเดียวกัน ในทางปฏิบัติเมื่อเราส่งสัญญาณทางไฟฟ้าไปหนึ่งชุดที่มอเตอร์จากตัวขับมอเตอร์ (Motor Driver) มอเตอร์ไม่สามารถสร้างเอาท์พุทเป็นลักษณะเชิงเส้น(Linear)กับคำสั่งทางไฟฟ้าหรืออินพุทกับเงื่อนไขได้เนื่องปัจจัยภายนอกหรือภาระงาน (Load)

ดังนั้นจึงมีการปรับเทียบ(Calibrate) อุปกรณ์ให้เข้าใกล้เชิงเส้นมากที่สุด จึงเป็นที่มาของการ Hook-up และ Tune มอเตอร์

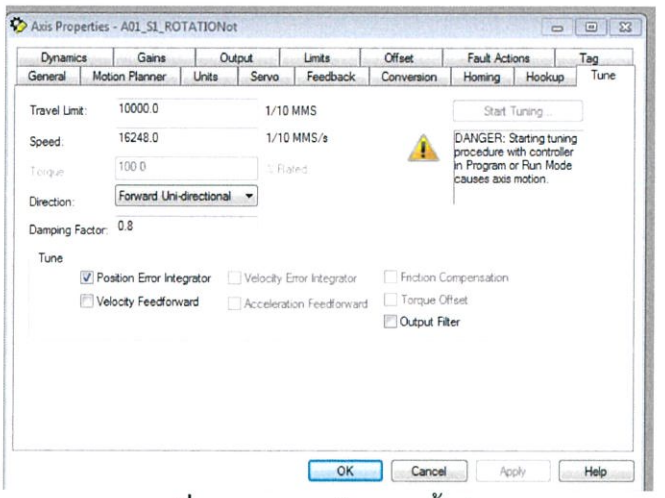
การ Hook-up คือการทดสอบการทำงานของ Feedback , Marker และ Output&Feedback ซึ่งในส่วน Marker นั้นจะไม่ได้ทำการทดสอบเนื่องจาก Encoder ไม่ได้ใช้ชาแนล Z จะทำการทดสอบ Output&Feedback จะเป็นการปรับเทียบทิศทางโดยโปรแกรมจะสั่งการเคลื่อนไหวตามค่าที่ผู้ใช้กำหนด (Test Increment) ไปยังมอเตอร์ สังเกตทิศทางเป็นบวกหรือลบเพื่อกำหนดทิศทางการทำงาน Encoder (Feedback Polarity)



ภาพที่ 3.39 หน้าต่าง Hookup ใน การตั้งค่าแกน

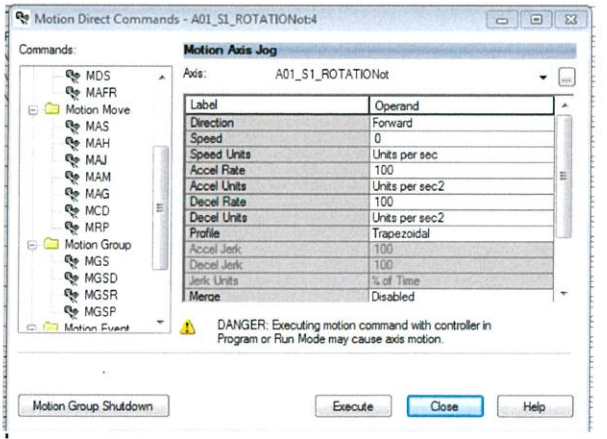
ในภาพที่ 3.39 เมื่อกำหนดการเคลื่อนไหวของ Feedback ที่ 1500/10 MMS ตัวขับมอเตอร์(Motor Driver) คำณวนสัญญาณไฟฟ้าที่ทำให้มอเตอร์หมุนที่ความเร็วของ Encoder เท่านี้ ซึ่งถือเป็นการปรับเทียบการทำงานระหว่างการหมุนมอเตอร์ และคำสั่งไฟฟ้าในหนึ่งระดับ ในส่วนของทิศทางนั้นเป็นไปในทิศทางเดียวกัน

การ Tune เป็นขั้นตอนเพื่อปรับเทียบความเร่ง(Accelation) และความหน่วง (Deceleration) ของมอเตอร์โดยอิงกับสัญญาณทางไฟฟ้าโดย Start Tuning แล้วทำการกรอกค่าความเร่งและความหน่วงลงไปจากนั้นมอเตอร์จะทำงานที่ความเร่งแล้วหน่วงนั้นเพื่อคำณวนแรงบิด (Torque) ที่ใช้ในการทำงานซึ่งค่าความเร่งและความหน่วงจะใช้ค่าที่เป็นความเร่งสูงสุดและความหน่วงสูงสุดของในแต่ละแกนเพื่อปรับแก้ให้ใกล้เคียงค่าที่ใช้จริงมากที่สุด ในขั้นตอนนี้ถือเป็นการปรับแก้ PID ไปโดยอัตโนมัติจากการ Tune ภาพที่ 3.40 เป็นหน้าต่างการ Tune



ภาพที่ 3.40 Tune ใน การตั้งค่าแกน

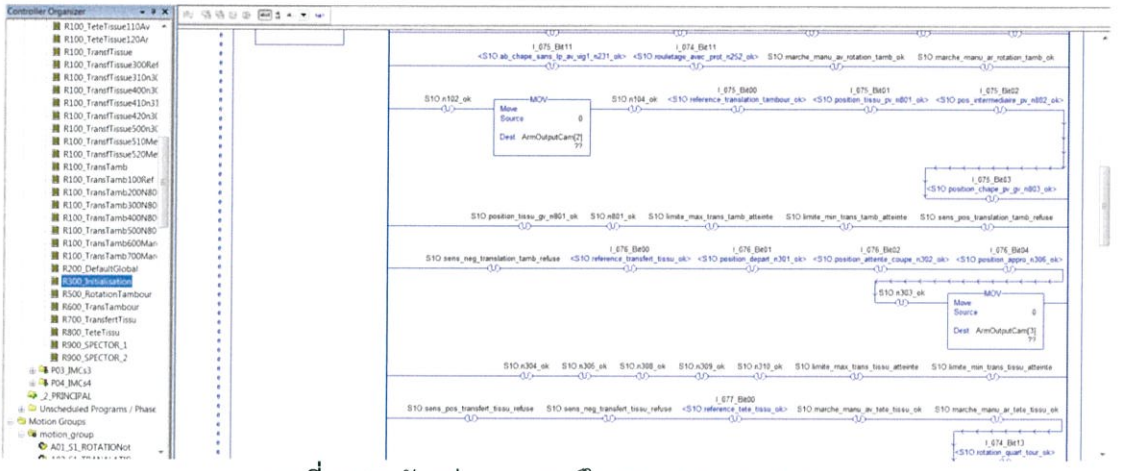
หลังจากทำการปรับตั้งค่าที่ส่งผลต่อการทำงานของมอเตอร์แล้วจะทดสอบ JOG โดยตั่งค่าโปรไฟล์,ความเร่ง,ความหน่วง และความเร็ว ของแกนนั้นและทดสอบ Motion Direct Drive On ด้วยการคำสั่งเป็นแรงดัน(โวล์ต) ทำการติดตามผลการการเคลื่อนที่ควรจะเป็นเพื่อทดสอบการทำงานพอสังเขปของทุกแกนผ่าน Motion Direct Commands ดังภาพที่ 3.41



ภาพที่ 3.41 หน้าต่าง Motion Direct Commands การทำ JOG

3.2.8 การ Intitailization โปรแกรม

การ Intitailization โปรแกรมเป็นหนึ่งในโปรแกรมที่ถูกออกแบบเพื่อคืนค่าเริ่มต้นของโปรแกรมเมื่อมีการเริ่มต้นเครื่องจักร หรือทุกครั้งที่เริ่มพีแอลซีใหม่ ส่งผลให้โปรแกรมจะถูกบังคับให้กลับสู่เริ่มต้นค่าบัฟเฟอร์ที่อยู่เลดเดอร์ที่ค้างจะถูกกำจัดออก อีกทั้งเป็นการเช็คอุปกรณ์ในแต่ละแกนมีการทำงานปกติหรือไหม ในทุกโฟลเดอร์โปรแกรม IMCS จะมีเลดเดอร์อยู่ใช่ชื่อว่า R300_Intitailisation



ภาพที่ 3.42 ตัวอย่างเลดเดอร์ใน R300_Intitailisation

3.2.8 การติดตามผลการทำงานมอเตอร์

เมื่อทำการสั่งทำงานของโปรแกรมใดๆแล้ว การติดตามผลเพื่อตรวจสอบโปรแกรมให้ชัดเจนตามเป้าหมายที่คาดหวังไว้ด้วยหน้าต่าง Monitor Tag ของ AXIS_SERVO ที่เป็นเสมือนแท็กคำสั่งของแต่ละแกนเช่น ภาพที่ 3.43

Name	Value	Description	Constant
A01_S1_ROTATIONot.AttributeErrorID	16#0000	Building Drum Rotation	
A01_S1_ROTATIONot.AccelerationCommand	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.AccelerationFeedback	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.ActualAcceleration	57.811916	Building Drum Rotation	
A01_S1_ROTATIONot.ActualPosition	24.997871	Building Drum Rotation	
A01_S1_ROTATIONot.ActualVelocity	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.AuxPositionFeedback	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.AverageVelocity	0.04624953	Building Drum Rotation	
A01_S1_ROTATIONot.CommandAcceleration	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.CommandPosition	25.0	Building Drum Rotation	
A01_S1_ROTATIONot.CommandVelocity	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.InterpolatedActualPosition	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.InterpolatedCommandPosition	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.MarkerDistance	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.MasterOffset	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.PositionCommand	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.PositionError	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.PositionFeedback	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.PositionIntegratorError	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.Registration1Position	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.Registration2Position	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.ServoOutputLevel	0.0	Building Drum Rotation	
A01_S1_ROTATIONot.StatActualPosition	778.5877	Building Drum Rotation	

Monitor Tags / Edit Tags /

ภาพที่ 3.43 หน้าต่าง Monitor Tags ของแกนที่ 1 ขณะทำงาน

บทที่ 4
ผลการดำเนินงาน

4.1 กล่าวนำ

จากบทที่ 3 ได้มีการกล่าวการทำงานของระบบ , ถึงการย้ายระบบ(Migration) และตั้งค่า (Configuration) ของการเตรียมโครงสร้างยาง Moudule4 โดยผ่านหุ่นยนต์ ซึ่งต้องปรับปรุงระบบทางด้านฮาร์ดแวร์ ซึ่งส่งผลให้ต้องปรับปรุงระบบทางด้านซอฟต์แวร์ด้วย สำหรับเนื้อหาในบทนี้จะกล่าวถึงขั้นตอนการทดสอบ และติดตามผลการทำงาน

4.2 ขั้นตอนการทดสอบ

1. การทดสอบฮาร์ดแวร์ Overtravel Limit และ Home Limit
2. การทดสอบฮาร์ดแวร์ตัวควบคุมตำแหน่ง(Motion Control) และ ตัวขับมอเตอร์
3. การทดสอบลำดับการทำงานของโปรแกรม
4. การทดสอบการทำงานโดยใช้วัตถุดิบจริง

4.3 ผลการทดสอบ

หลังจากทำการติดตั้งฮาร์ดแวร์แล้วก่อนการดำเนินการเริ่มทำงานของอุปกรณ์ จำเป็นจะต้องตรวจสอบการเชื่อมสายทั้งหมดก่อน แล้วจึงเช็ค Overtravel Limit และ Home Limit ตารางที่ 4.1

ตารางที่ 4.1 ผลการทดสอบฮาร์ดแวร์ Overtravel Limit และ Home Limit

	ชื่อแกน	Overtravel Limit	Home Limit	หมายเหตุ
1	ROTATION TAMBOUR CONFECTION	O	O	
2	TRANSLATION TAMBOUR CONFECTION	O	O	
3	TRANSFERT TISSU	X	O	เปลี่ยนเซนเซอร์ใหม่
4	TETE TISSU	O	O	
5	TRANSFERT CHAPE	O	O	
6	TETE CHAPE	O	O	

7	ROULETEUR CHAPE	O	O	
8	TRANSFERT PROTECTEUR	O	O	
9	TRANSFERT NAPPE2	O	O	
10	TRANSLATION COURONNES	O	O	
11	ROTATION TAMBOUR TS	O	O	
12	HAUTEUR POSTE TS	O	O	
13	LARGEUR COURONNE TS	O	O	
14	GONFLAGE DEGONFLANGE SEAT COURONNES	O	O	

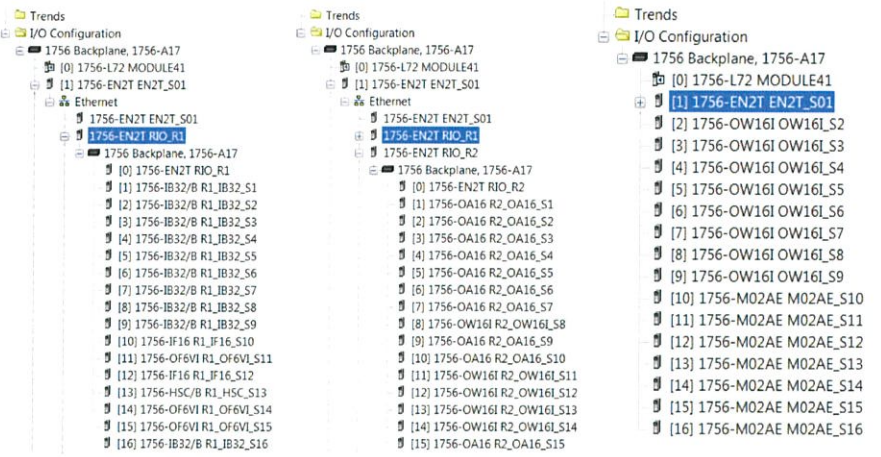
O ผ่าน X ไม่ผ่าน

การทดสอบฮาร์ดแวร์ตัวควบคุมตำแหน่ง(Motion Control) และ ตัวขับมอเตอร์(Motor Driver) เบื่องต้นเพื่อตรวจสอบอุปกรณ์ที่ติดตั้งไปแล้ว เพื่อลดความผิดพลาดจากฮาร์แวร์ตารางที่ 4.2

ตารางที่ 4.2 ผลการทดสอบฮาร์ดแวร์ตัวควบคุมตำแหน่ง (Motion Control)

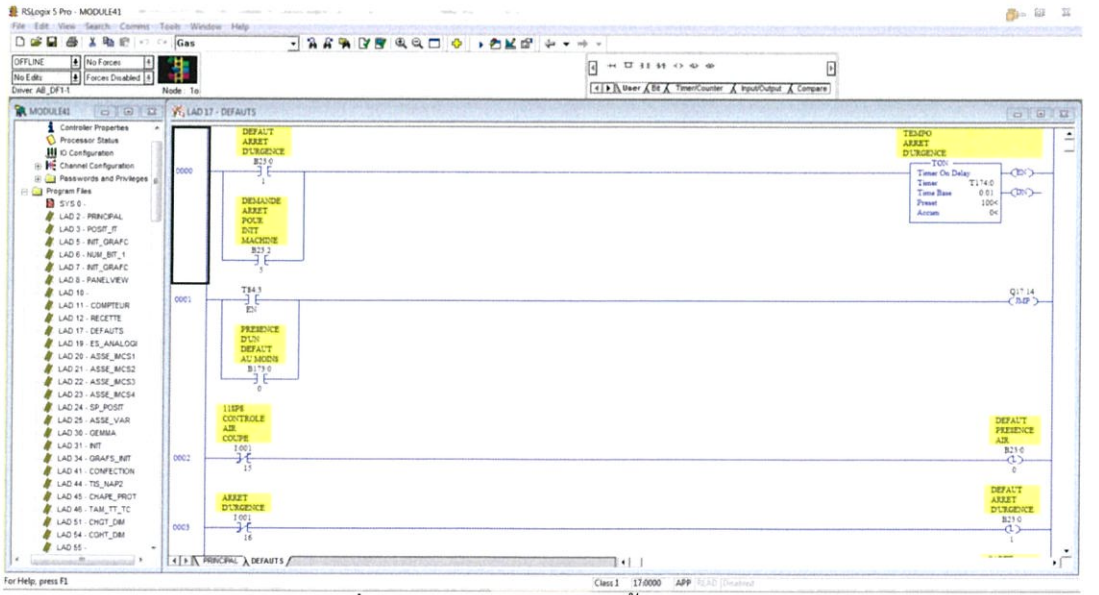
แกนที่	สล็อทที่	การทดสอบฮาร์ดแวร์		หมายเหตุ
		Motion Control	Motor Driver	
1	10	O	O	
2			O	
3	11	O	O	
4			X	อุปกรณ์มีปัญหา
5	12	X	O	การ์ด M02AE ขา ENABLE เสีย
6			O	
7	13	O	O	
8			O	
9	14	O	O	
10			O	
11	15	O	O	
12			O	
13	16	O	O	
14			O	

O ผ่าน X ไม่ผ่าน

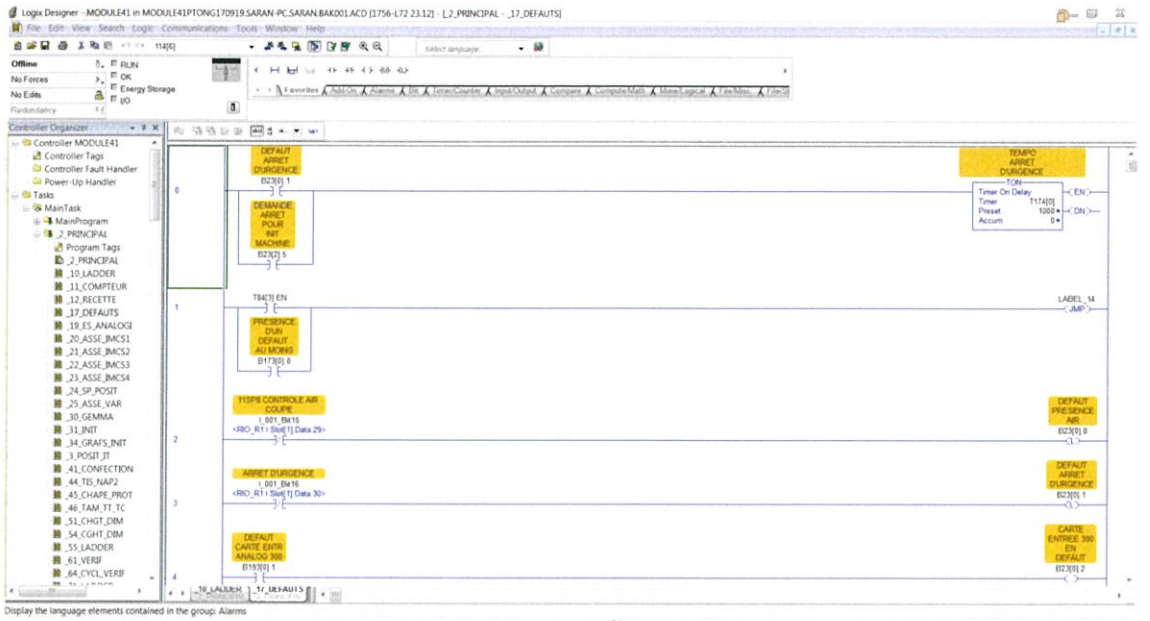


ภาพที่ 4.1 การ์ดที่ถูกสร้าง RIO_R1,RIO_R2 และ Local จากซ้ายไปขวาตามลำดับ

ภาพที่ 4.1 เป็นการ์ดที่ถูกสร้างขึ้นอีกทั้งยังสร้างแท็กจากพีแอลซีจริงและตรวจสอบกับการ์ดจริงแล้ว ในภาพที่ 4.2 และ 4.3 เป็นการเปรียบเทียบโปรแกรมหลังจากการแปลงโปรแกรมจาก RsLogix5 เป็น Studio5000 ซึ่งยังคงไว้การทำงานเดิม นอกจากนี้ยังแปลงแท็กเรียบร้อย เมื่อทำการตรวจสอบโปรแกรมโดยรวมทั้งหมดเทียบลำดับการทำงาน

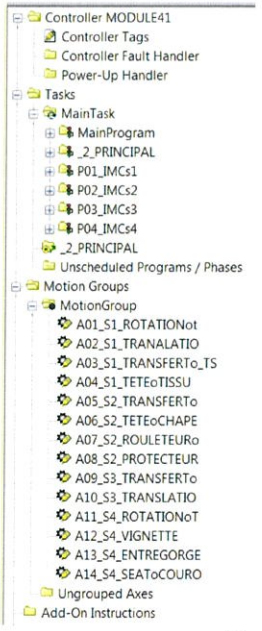


ภาพที่ 4.2 หน้าต่างโปรแกรมทั้งหมด RsLogix5



ภาพที่ 4.3 หน้าต่างโปรแกรมทั้งหมด Studio5000

เมื่อโปรแกรมทั้งหมดพร้อมจะทำงานจะมีโฟล์เดอร์ Controller Organizer ดังภาพที่ 4.4 ซึ่งมี Motion Group ที่ตั้งค่าไว้แล้ว 14 แกน และ 6 โฟล์เดอร์ใน Task



ภาพที่ 4.4 Controller Organizer ของโฟลเดอร์เมื่อเสร็จสิ้น

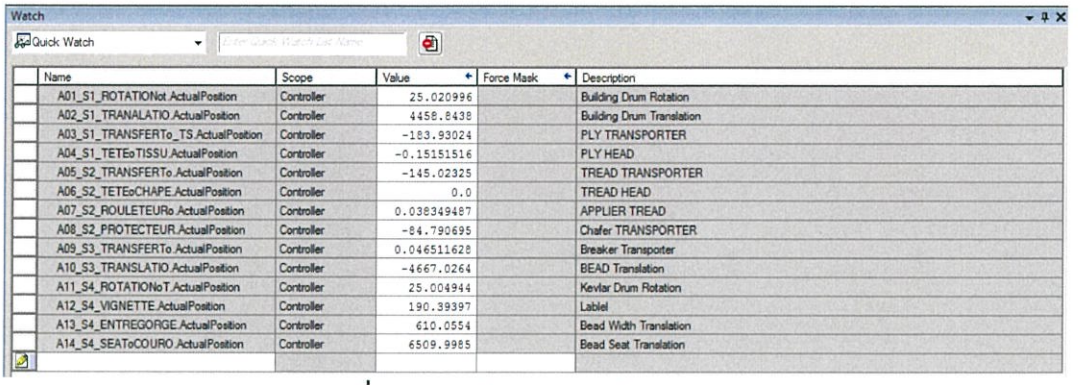
การทดสอบลำดับการทำงานของโปรแกรมที่เครื่องจักรเป็นขั้นตอนทางซอฟท์แวร์ขั้นตอนแรกซึ่งจะรวมไปถึงการตั้งค่าของอุปกณ์ตัวควบคุมตำแหน่ง (Motion Control) และ ตัวขับมอเตอร์(Motor Driver) สรุปได้ดังตารางที่ 4.3

ตารางที่ 4.3 การทดสอบลำดับการทำงานของโปรแกรม

	ชื่อแกน	JOG Test	Direct Drive Move	MAX Speed Test	Tune	Intitailization
1	ROTATION TAMBOUR CONFECTION	O	O	O	O	O
2	TRANSLATION TAMBOUR CONFECTION	O	O	O	O	O
3	TRANSFERT TISSU	O	O	O	O	O
4	TETE TISSU	O	O	O	O	O
5	TRANSFERT CHAPE	O	O	O	O	O
6	TETE CHAPE	O	O	O	O	O
7	ROULETEUR CHAPE	O	O	O	O	O
8	TRANSFERT PROTECTEUR	O	O	O	O	O
9	TRANSFERT NAPPE2	O	O	O	O	O
10	TRANSLATION COURONNES	O	O	O	O	O
11	ROTATION TAMBOUR TS	O	O	O	O	O
12	HAUTEUR POSTE TS	O	O	O	O	O
13	LARGEUR COURONNE TS	O	O	O	O	O
14	GONFLAGE DEGONFLANGE SEAT COURONNES	O	O	O	O	O

O ผ่าน X ไม่ผ่าน

การทดสอบการทำงานโดยใช้วัตถุดิบจริงเพื่อตรวจสอบผลิตภัณฑ์ที่เกิดจากกระบวนการ ในขั้นตอนนี้จะเป็นการสอบเป็นการทดสอบเพียงอุปกรณ์เท่านั้น ว่าสามารถทำงานได้ครบทุกขั้นตอน ติดตามการทำงานโดย Quick Watch ของระบบทั้งหมดในแต่ละแกนในขณะทำงานตามลำดับของเลดเดอร์ในค่า Value คือตำแหน่งที่สร้างขึ้นมา และใน Name แสดงชื่อแกน


Watch

Quick Watch

	Name	Scope	Value	Force Mask	Description
	A01_S1_ROTATIONot.ActualPosition	Controller	25.020996		Building Drum Rotation
	A02_S1_TRANALATIO.ActualPosition	Controller	4458.8438		Building Drum Translation
	A03_S1_TRANSFERTo_TS.ActualPosition	Controller	-183.93024		PLY TRANSPORTER
	A04_S1_TETEoTISSU.ActualPosition	Controller	-0.15151516		PLY HEAD
	A05_S2_TRANSFERTo.ActualPosition	Controller	-145.02325		TREAD TRANSPORTER
	A06_S2_TETEoCHAPE.ActualPosition	Controller	0.0		TREAD HEAD
	A07_S2_ROULETEURo.ActualPosition	Controller	0.038349487		APPLIER TREAD
	A08_S2_PROTECTEUR.ActualPosition	Controller	-84.790695		Chafer TRANSPORTER
	A09_S3_TRANSFERTo.ActualPosition	Controller	0.046511628		Breaker Transporter
	A10_S3_TRANSLATIO.ActualPosition	Controller	-4667.0264		BEAD Translation
	A11_S4_ROTATIONoT.ActualPosition	Controller	25.004944		Kevlar Drum Rotation
	A12_S4_VIGNETTE.ActualPosition	Controller	190.39397		Label
	A13_S4_ENTREGORGE.ActualPosition	Controller	610.0554		Bead Width Translation
	A14_S4_SEAToCOURO.ActualPosition	Controller	6509.9985		Bead Seat Translation

ภาพที่ 4.5 ภาพติดตามการทำงานของทุกแกน

ในการทดสอบผลิตจริง ผู้ปฏิบัติการของบริษัทจะเป็นควบคุมคุณภาพและทดสอบแก้ไขค่าเองทั้งหมดให้ได้ตามต้องการ ดังนั้นขั้นตอนนี้เป็นเพียงการดูภาพรวมการทำงาน

ตารางที่ 4.4 การทดสอบการทำงานโดยใช้วัตถุดิบจริง

ขั้นตอน	การทดสอบ	หมายเหตุ
การม้วนแผ่นยาง (Casing/Tissue)	O	
การใส่แผ่นยางกันรั่ว(Sub Tread)	O	
ติดแถบป้องกันการเสียดสี(Side Wall)	O	
การม้วนเคฟลาร์(Beads/Kevlar)	O	
การติดหน้ายาง(Tread)	O	

O ผ่าน X ไม่ผ่าน

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากการดำเนินงานในการปรับปรุงและเปลี่ยนแปลงระบบ(Migration) ทั้งฮาร์ดแวร์และซอฟท์แวร์ของเครื่องขึ้นโครงสร้างยางผ่านเครื่องจักรระบบอัตโนมัติโดยผู้ใช้งานสามารถควบคุมการผลิตผ่านหน้าจอPanelView ซึ่งการปรับปรุงและเปลี่ยนแปลงระบบของเครื่องจักรมุ่งให้ระบบยังคงเดิมแต่มีประสิทธิภาพมากขึ้น แนวทางในการแก้ไขจะเลือกใช้อุปกรณ์ที่ครอบคุมทำงานเดิม แก้ไขโปรแกรมให้เป็นแนวทางเดิมอิงกับโปรแกรมเดิม นอกจากนี้อุปกรณ์ที่ฉลาดมากขึ้นสามารถดึงประสิทธิภาพออกมาเพิ่มเติมได้อีก

5.2 ปัญหาและวิธีการแก้ไขปัญหา

5.2.1 ปัญหาที่พบ

1. ข้อมูลและเอกสารที่ขาดหายไป ทำให้เกิดความล่าช้า
2. การ Wiring ผิดพลาดที่หน้างานซึ่งเกิดจากสายที่เก่า มีลักษณะตัวอักษรจาง
3. การทำงานผิดพลาดจากการเปลี่ยนแปลงแท็ก
4. อุปกรณ์ที่มีปัญหาจากทางเครื่องจักร และเซนเซอร์บางตัว

5.2.2 วิธีการแก้ไข้ปัญหา

1. ตรวจสอบเอกสารจาก แหล่งข้อมูลภายนอกก่อน
2. ตรวจสอบการทำงานที่หน้างานจริงก่อนเริ่มทำงาน
3. ขอความร่วมมือจาก ผู้ดูแลเครื่องจักรในการแก้ไขอุปกรณ์ที่เครื่องจักร

5.3 ข้อเสนอแนะ

การทำการปรับปรุงและเปลี่ยนแปลงระบบที่เป็นลักษณะเครื่องจักรอัตโนมัติ จำเป็นที่จะมีความเข้าใจลักษณะการทำงานของระบบเคลื่อนไหวทั้งหมด เช่น ตัวขับมอเตอร์(Motor Driver) และยังควรจะมีพื้นฐานการใช้โปรแกรมที่ค่อนข้างดี ซึ่งจะเป็นพื้นฐานที่ดีในการทำงานการอย่างมีประสิทธิภาพ ถูกต้อง รวดเร็ว และครบถ้วน

เอกสารอ้างอิง

[1] PLC-5 และ ControlLogix

แหล่งที่มา: Expert automation's Migration PLC-5 Presentation

[2] Module 4 System Architecture

แหล่งที่มา: Expert Automation LionTyre Module4 Bill of Material

[4] M02AE

แหล่งที่มา: Rockwell Automation

http://literature.rockwellautomation.com/idc/groups/literature/documents/in/1756-in047_-en-p.pdf

[5] IMCs

แหล่งที่มา: Rockwell Automation

http://literature.rockwellautomation.com/idc/groups/literature/documents/td/4100-td023_-en-p.pdf

[6] Unidrive Series

แหล่งที่มา: Emerson Industrial

http://www.emersonindustrial.com/en-EN/documentcenter/ControlTechniques/PDF/catalog/06cat_f2_ac-unidrive_sp.pdf