รายงานสหกิจศึกษาฉบับสมบูรณ์

ศูนย์กลางรวบรวมข้อมูลด้วย PLC สำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์

Data Concentrator for Animal Feed Packaging System by Using a PLC

นางสาวจารวี อ่อนนิวัฒนานนท์

ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2561

รายงานสหกิจศึกษาฉบับสมบูรณ์

ศูนย์กลางรวบรวมข้อมูลด้วย PLC สำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์

Data Concentrator for Animal Feed Packaging System by Using a PLC

นางสาวจารวี อ่อนนิวัฒนานนท์

ภาควิชาวิศวกรรมการวัดและควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานสหกิจศึกษา ศูนย์กลางรวบรวมข้อมูลด้วย PLC สำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์

ชื่อ-สกุล นักศึกษา นางสาวจารวี อ่อนนิวัฒนานนท์

คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมการวัดและควบคุม

ชื่อ - สกุล อาจารย์นิเทศ รศ.ดร.วิศรุต ศรีรัตนะ

ชื่อ - สกุล ผู้นิเทศ นายศุภณัฐ ตุงคะบูรณะ

ชื่อสถานประกอบการ บริษัท พีเอส เอนจิเนียริ่ง คอนซัลแตนท์ จำกัด

บทคัดย่อ

โครงงานสหกิจศึกษาได้จัดทำขึ้นเพื่อ ออกแบบศูนย์กลางรวบรวมข้อมูลของกระบวนการบรรจุภัณฑ์อาหารสัตว์ด้วย PLC ของทางบริษัท Rockwell Automation โดยใช้โปรแกรม Studio5000 ในการเขียนโปรแกรม PLC และโปรแกรม FactoryTalk View Studio ในการสร้างส่วนแสดงผลข้อมูลของกระบวนการ ซึ่งศูนย์กลางรวบรวมข้อมูลมีหน้าที่ในการรับส่งข้อมูลระหว่างระบบ MES กับระบบ PCS และรวบรวมข้อมูลในกระบวนการจากเครื่องจักรแต่ละตัวในระบบ PCS ส่งต่อข้อมูลที่ได้ให้กับระบบ MES เพื่อนำข้อมูลไปใช้ในการวิเคราะห์ความผิดพลาดของกระบวนการบรรจุภัณฑ์ที่ทำให้การผลิตสินค้าไม่ได้ตามจำนวนที่กำหนดไว้ ซึ่งความผิดพลาดของกระบวนการอาจเกิดจากการพิมพ์วันที่ของการผลิตสินค้าผิดหรือน้ำหนักของสินค้าเกินจากค่ามาตรฐานที่ตั้งไว้ โดยศูนย์กลางรวบรวมข้อมูลจะช่วยให้การสั่งดำเนินการผลิตจากระบบ MES ไปยังเครื่องจักรหลาย ๆ เครื่องในระบบ PCS มีความสะดวกและมีถูกต้องของข้อมูลมากยิ่งขึ้น และสามารถนำข้อมูลในกระบวนการไปใช้ในการวิเคราะห์ในระบบ MES ได้อย่างรวดเร็วทำให้โรงงานสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้อย่างทันท่วงที

คำสำคัญ : PLC, MES, PCS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Co-operative Title: Data Concentrator for Animal Feed Packaging System by Using a PLC
Student Intern Name: Ms.Jarawee Onniwattananon
Faculty: Engineering **Department:** Instrumentation and Control Engineering
Advisor Name: Assoc.Prof.Dr.Witsarut Sriratana
Mentor Name: Mr.Supanut Tungkaburana
Company: PS Engineering Consultant Company Limited

ABSTRACT

Cooperative education project is make to designed data concentrator for animal feed packaging system by using a PLC of Rockwell Automation. Data concentrator using Studio 5000 to program PLC and FactoryTalk View Studio to create display. The operating principle of data concentrator is transfer data between Manufacturing Execution Systems and Process Control System, collect data in the process from each machine in the PCS and forward the data to the MES for used to analyze the failure of the packaging process, which results in the production does not meet the specified number. Process failures may be caused by incorrect date of manufacture or weight of the product exceeds the standard. Data concentrator will make the production order from MES to machines in PCS convenient and accurate, process data can be used for analysis in the MES system quickly to enable factory can correct any errors that occur in a timely manner.

Keywords: PLC, MES, PCS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานสหกิจศึกษาเรื่องศูนย์กลางรวบรวมข้อมูลด้วย PLC สำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์ สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากได้รับการสนับสนุนจาก บริษัท พีเอส เอนจิเนียริ่ง คอนซัลแตนท์ จำกัด ที่ได้เปิดโอกาสให้ผู้จัดทำได้เข้าร่วมโครงงานสหกิจศึกษาตลอดระยะเวลาหนึ่งภาคเรียนการศึกษา ผู้จัดทำขอขอบพระคุณความช่วยเหลือต่าง ๆ จาก คุณศุภณัฐ ตุงคะบูรณะ ผู้นิเทศงานและบุคลากรภายในบริษัทที่ได้ให้คำแนะนำ และให้คำปรึกษา มอบความรู้และประสบการณ์ต่าง ๆ ในการทำงานอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอขอบพระคุณ รศ.ดร.วิศรุต ศรีรัตนะ อาจารย์นิเทศและคณาจารย์สาขาวิชาวิศวกรรมการวัดคุมและควบคุมทุกท่าน ที่ได้ให้ความช่วยเหลือและให้คำปรึกษาในการแก้ไขปัญหาต่าง ๆ ตลอดระยะเวลาการทำสหกิจศึกษา ให้สำเร็จลุล่วงไปได้ด้วยดี

นางสาวจารวี อ่อนนิวัฒนานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีการดำเนินการโครงงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง 5

2.1 กล่าวนำ 5

2.2 โครงสร้างของระบบอัตโนมัติ 5

2.2.1 Enterprise Resource Planning (ERP) 5

2.2.2 Manufacturing Execution Systems (MES) 6

2.2.3 Concentrator 6

2.2.4 Supervisory Control and Data Acquisition (SCADA) 8

2.3 Programmable Logic Controller (PLC) 11

2.3.1 ความหมายของ PLC 11

2.3.2 โครงสร้างของ PLC 11

2.3.3 ความสามารถของ PLC 15

2.3.4 ขนาดของ PLC 16

2.3.5 ภาษาที่ใช้เขียนโปรแกรม 17

2.4 โครงสร้างด้านการสื่อสาร (Communications) 20

2.4.1 Ethernet 20

2.4.2 Modbus 21

2.4.3 OPC 25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.5 โปรแกรมที่ใช้ในการทำงาน....27
2.5.1 โปรแกรม Studio5000....27
2.5.2 โปรแกรม Studio5000 Logix Emulate....27
2.5.3 โปรแกรม FactoryTalk View Studio....28
บทที่ 3 วิธีดำเนินการออกแบบระบบศูนย์กลางรวบรวมข้อมูล....29
3.1 กล่าวนำ....29
3.2 แบบจำลองของโรงงานและกระบวนการบรรจุภัณฑ์อาหารสัตว์....29
3.3 จัดทำเอกสารการจับคู่ข้อมูลและกำหนด Address ให้กับพารามิเตอร์ของกระบวนการ....31
3.4 การออกแบบฟังก์ชั่นการทำงาน (Function design)....33
3.4.1 Process Cell Level....35
3.4.2 Process Unit Level....44
3.5 ขั้นตอนการเขียนโปรแกรม Studio 5000....52
3.5.1 การสร้าง New Program....52
3.5.2 การสร้าง New Routine....53
3.5.3 การสร้าง User-Defined Data Types....54
3.5.4 การสร้าง Local Tag....56
3.5.5 การสร้าง Controller Tag....57
3.5.6 การออนไลน์โปรแกรม....59
3.5.7 การแก้ไขออนไลน์....61
3.5.8 การทดสอบโปรแกรม โดยใช้โปรแกรม Studio5000 Logix Emulate....63
3.6 ขั้นตอนการเขียนโปรแกรม FactoryTalk View Studio....74
3.6.1 การเปิดโปรแกรม FactoryTalk View Site Edition....74
3.6.2 การสร้าง New Display....75
3.6.3 การสร้าง Global Objects....77
3.6.4 การสร้าง Parameters....78
3.6.5 การสร้าง Client หรือ สร้าง Runtime....79
บทที่ 4 ผลการทดสอบระบบศูนย์กลางรวบรวมข้อมูล....84
4.1 กล่าวนำ....84

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.2 ผลการทดสอบการทำงานของโปรแกรม....84
4.2.1 Order Preview....84
4.2.2 Load Sequence....85
4.2.3 Start Unit PO....86
4.2.4 Consume....88
4.2.5 Produce....89
4.2.6 Complete Unit PO....91
4.2.7 Complete Sequence....92
4.2.8 Complete POI....94
4.3 ผลการทดสอบการแสดงผลข้อมูลผ่านระบบสกาดา (SCADA)....95
4.3.1 Order Preview....95
4.3.2 Load Sequence....96
4.3.3 Start Unit PO....97
4.3.4 Consume....98
4.3.5 Produce....98
4.3.6 Complete Unit PO....99
4.3.7 Complete Sequence....99
4.3.8 Complete POI....100
บทที่ 5 สรุปผลและข้อเสนอแนะ....101
5.1 สรุปผลการดำเนินงาน....101
5.2 ปัญหาที่พบและแนวทางการแก้ไข....101
5.2.1 ปัญหาที่พบ....101
5.2.2 แนวทางการแก้ปัญหา....101
5.3 ข้อเสนอแนะ....102
บรรณานุกรม....103
ประวัติผู้เขียน....104

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

1.1 แผนการดำเนินงาน....3

2.1 การจำแนกขนาดของ PLC....17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ **หน้า**

2.1 โครงสร้างของระบบอัตโนมัติ....5
2.2 คอนเซนเทรเตอร์ในระบบเครือข่าย....6
2.3 Point-to-Point Configuration....8
2.4 Point-to-Multipoint Configuration....9
2.5 ส่วนประกอบของ PLC....12
2.6 ลำดับการทำงานในหนึ่งรอบการทำงานของ PLC....13
2.7 ตัวอย่างอุปกรณ์อินพุตที่ต่อเข้ากับ PLC....14
2.8 ตัวอย่างอุปกรณ์เอาท์พุตที่ต่อเข้ากับ PLC....15
2.9 Ladder Diagram....18
2.10 Function block diagram....18
2.11 Instruction list....19
2.12 Structure text....19
2.13 SFC (Sequential Function)....20
2.14 Ethernet LAN Diagram....21
2.15 การติดต่อสื่อสารแบบ Master/Slave....22
2.16 ลักษณะเฟรมข้อมูลของ MODBUS RTU....23
2.17 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS RTU....23
2.18 ลักษณะเฟรมข้อมูลของ MODBUS ASCII....24
2.19 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS ASCII....24
2.20 ลักษณะการสื่อสารของ MODBUS TCP/IP....25
2.21 การแปลง MODBUS Serial เป็น MODBUS Ethernet....25
2.22 ลักษณะการสื่อสารของ OPC....26
2.23 การเชื่อมต่อ OPC กับอุปกรณ์ต่าง ๆ....26
2.24 โปรแกรม Studio 5000....27
2.25 โปรแกรม Studio5000 Logix Emulate....27
2.26 โปรแกรม FactoryTalk View Studio....28
3.1 แบบจำลองโรงงาน....29
3.2 กระบวนการบรรจุภัณฑ์อาหารสัตว์ (สายการผลิตที่11)....30
3.3 กระบวนการบรรจุภัณฑ์อาหารสัตว์ (สายการผลิตที่11) (ต่อ)....30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.4 ตัวอย่างแผนภาพ PACKING MACHINE INTERFACE SCOPE 31
3.5 ตัวอย่างเอกสารของ MOM....32
3.6 ตัวอย่างเอกสารการจับคู่ข้อมูลของบริษัท KHS (สายการผลิตที่11)....32
3.7 ตัวอย่างเอกสารการจับคู่ข้อมูลของบริษัท KHS (สายการผลิตที่11) (ต่อ)....33
3.8 ภาพรวมการทำงานของศูนย์กลางรวบรวมข้อมูล....33
3.9 Array ในศูนย์กลางรวบรวมข้อมูลที่ใช้เก็บ 5 PO....35
3.10 Array ของฟังก์ชั่น Handshake ในขั้นตอน Order Preview....35
3.11 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Order Preview....36
3.12 Array ของ MES_HMIUnit.Context ในขั้นตอน Load Sequence....37
3.13 Array ของฟังก์ชั่น Handshake ในขั้นตอน Load Sequence....38
3.14 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Load Sequence....38
3.15 Array ของ MES_HMIUnit.Context ในขั้นตอน Complete Sequence....39
3.16 Array ของฟังก์ชั่น Handshake ในขั้นตอน Complete Sequence....40
3.17 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Complete Sequence....41
3.18 Array ของ MES_HMIUnit.CompletePOI ในขั้นตอน Complete POI....42
3.19 Array ของฟังก์ชั่น Handshake ในขั้นตอน Complete POI....42
3.20 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Complete POI....43
3.21 Array ของ MES_Object.Context ในขั้นตอน Start Unit PO....44
3.22 Array ของฟังก์ชั่น Handshake ในขั้นตอน Start Unit PO....45
3.23 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Start Unit PO....45
3.24 Array ของ MES_Object.CONS ในขั้นตอน Consume....46
3.25 Array ของฟังก์ชั่น Handshake ในขั้นตอน Consume....47
3.26 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Consume....47
3.27 Array ของ MES_Object.PROD ในขั้นตอน Produce....48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.28 Array ของฟังก์ชั่น Handshake ในขั้นตอน Produce....49
3.29 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Produce....49
3.30 Array ของ MES_Object.Context ในขั้นตอน Complete Unit PO....50
3.31 Array ของฟังก์ชั่น Handshake ในขั้นตอน Complete Unit PO....51
3.32 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Complete Unit PO....51
3.33 การเพิ่มโปรแกรมที่ใช้ลงใน MainTask....52
3.34 การตั้งชื่อโปรแกรม....53
3.35 การสร้าง Routine สำหรับเขียนโปรแกรมใหม่....53
3.36 การตั้งชื่อ Routine และกำหนดภาษาเขียนโปรแกรม....54
3.37 การสร้าง Data Type....54
3.38 การตั้งชื่อ Data Type และกำหนดภาษาเขียนโปรแกรม....55
3.39 ตัวอย่าง User-Defined Data Types....55
3.40 การสร้าง Local Tag สำหรับ Tag ที่ใช้ในภายใน Main Program....56
3.41 การตั้งชื่อ Local Tag และกำหนดรายละเอียดต่าง ๆ....56
3.42 ตัวอย่าง Local Tag....57
3.43 การสร้าง Controller Tag สำหรับ Tag ที่ใช้ทั้งภายนอกหรือภายในโปรแกรม....57
3.44 การตั้งชื่อ Controller Tag และกำหนดรายละเอียดต่าง ๆ....58
3.45 ตัวอย่าง Controller Tag....58
3.46 การออนไลน์โปรแกรม PLC....59
3.47 การเลือกคอนโทรลเลอร์ที่ต้องการใช้ในการออนไลน์โปรแกรม PLC....59
3.48 การดาวน์โหลดโปรแกรม PLC ลงในคอนโทรลเลอร์....60
3.49 การสั่งให้คอนโทรลเลอร์ Run โปรแกรม....60
3.50 การ Online Edit โปรแกรม....61
3.51 การแก้ไขลอจิกในโปรแกรม....61
3.52 ปุ่ม Verify Routine....62
3.53 หน้าต่างแสดง error....62
3.54 ปุ่ม Finalize All Edits....62
3.55 การยืนยันการแก้ไข....63

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.56	ตัวอย่างลอจิกที่ทำการแก้ไขแล้ว	63
3.57	การสร้าง PLC จำลอง	64
3.58	การเลือกโมดูลของคอนโทรลเลอร์	64
3.59	การกำหนดรายละเอียดต่าง ๆ ของคอนโทรลเลอร์	64
3.60	การกำหนดรายละเอียดต่าง ๆ ของคอนโทรลเลอร์ (ต่อ)	65
3.61	การสั่งให้คอนโทรลเลอร์จำลอง Run โปรแกรม	65
3.62	การเลือกคอนโทรลเลอร์ที่ต้องการทดสอบ	66
3.63	การเปลี่ยนชนิดของคอนโทรลเลอร์	66
3.64	การเปลี่ยนชนิดของคอนโทรลเลอร์ (ต่อ)	67
3.65	การยืนยันการเปลี่ยนชนิดของคอนโทรลเลอร์	67
3.66	การเปลี่ยนหมายเลข Slot	68
3.67	ตัวอย่างคอนโทรลเลอร์ที่เปลี่ยนเป็น Emulator	68
3.68	ตัวอย่างโปรแกรม Simulation	69
3.69	ตัวอย่างโปรแกรม Simulation (ต่อ)	70
3.70	ตัวอย่างโปรแกรม BufferConsume	71
3.71	ตัวอย่างโปรแกรม BufferProduce	72
3.72	ตัวอย่างโปรแกรม ProcessStepPC	73
3.73	ตัวอย่างโปรแกรม ProcessStep	73
3.74	ตัวอย่างคอนโทรลเลอร์ที่เปลี่ยนเป็น Emulator	74
3.75	การเลือกไฟล์โปรแกรมที่ต้องการเปิด	74
3.76	การสร้าง Displays	75
3.77	การตั้งค่าหน้าจอ	75
3.78	การตั้งค่าหน้าจอ (ต่อ)	76
3.79	ตัวอย่างหน้าจอที่ถูกสร้างขึ้น	76
3.80	การตั้งชื่อหน้าจอ	77
3.81	การตั้งชื่อหน้าจอ (ต่อ)	77
3.82	การสร้าง Global Objects	77
3.83	ตัวอย่าง Global Objects ที่ถูกนำไปใช้ในโปรแกรมต่าง ๆ	78
3.84	การสร้างพารามิเตอร์	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.85 ตัวอย่างพารามิเตอร์ ที่ถูกนำไปใช้ในโปรแกรมต่าง ๆ....79
3.86 การสร้าง Runtime....79
3.87 การสร้าง Runtime (ต่อ)....80
3.88 การตั้งชื่อ Runtime....80
3.89 การเลือกชนิดของ SE client application....81
3.90 การเลือกโปรเจคที่จะสร้าง Runtime....81
3.91 การกำหนดหน้าแรกของ Runtime....82
3.92 การกำหนดขนาดของหน้าจอ Runtime....82
3.93 การเลือกรูปแบบการ Save Runtime....83
3.94 ตัวอย่างหน้าจอ Runtime....83
4.1 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Order Preview....84
4.2 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Done ในขั้นตอน Order Preview....85
4.3 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Load Sequence....85
4.4 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Load Sequence....86
4.5 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Load Sequence....86
4.6 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Load Sequence....86
4.7 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Start Unit PO....87
4.8 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Start Unit PO....87
4.9 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Start Unit PO....87
4.10 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Start Unit PO....88
4.11 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Start Unit PO (ต่อ)....88
4.12 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Consume....88
4.13 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Consume....89
4.14 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Consume....89
4.15 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Consume....89
4.16 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Produce....90
4.17 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Produce....90
4.18 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Produce....90
4.19 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Produce....91
4.20 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Complete Unit PO....91

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.21 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Complete Unit PO	91
4.22 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Complete Unit PO	92
4.23 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete Unit PO	92
4.24 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete Unit PO (ต่อ)	92
4.25 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Complete Sequence	93
4.26 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Complete Sequence	93
4.27 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Complete Sequence	94
4.28 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete Sequence	94
4.29 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Complete POI	94
4.30 ผลการทดสอบ PCS ส่งบิต Trigger ในขั้นตอน Complete POI	95
4.31 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Complete POI	95
4.32 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete POI	95
4.33 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Order Preview	96
4.34 ผลการทดสอบการแสดงค่าพารามิเตอร์ต่าง ๆ ในขั้นตอน Order Preview	96
4.35 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Load Sequence	97
4.36 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Start Unit PO	97
4.37 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Consume	98
4.38 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Produce	98
4.39 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Complete Unit PO	99
4.40 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Complete Sequence	100
4.41 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Complete POI	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของโครงงาน

ในปัจจุบันธุรกิจการผลิตและโรงงานอุตสาหกรรมส่วนใหญ่มักนิยมใช้ระบบ Manufacturing Execution Systems (MES) เป็นหัวใจหลักในการควบคุมการผลิตและการเพิ่มประสิทธิภาพของสินค้า ดังนั้นการรวบรวมข้อมูลในระบบ Process Control System (PCS) มาใช้ในการวิเคราะห์เพื่อแก้ไขความผิดพลาดของกระบวนการทำงานจึงเป็นสิ่งที่จำเป็นต่อธุรกิจ เทคโนโลยีที่จะมามีบทบาทในการจัดเก็บและรวบรวมข้อมูลของกระบวนการต่าง ๆ ในโรงงานอุตสาหกรรมนั้นก็คือศูนย์กลางรวบรวมข้อมูลซึ่งจะช่วยให้การสั่งดำเนินการผลิตจากระบบ MES ไปยังเครื่องจักรในระบบ PCS มีความสะดวกและมีถูกต้องของข้อมูลมากยิ่งขึ้นและสามารถนำข้อมูลไปใช้ในการวิเคราะห์ในระบบ MES ได้อย่างรวดเร็วทำให้โรงงานสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้อย่างทันท่วงที

บริษัท พีเอส เอนจิเนียริ่ง คอนซัลแตนท์ จำกัด เป็นบริษัทที่มีความเชี่ยวชาญในเรื่องการออกแบบระบบควบคุม, ตู้ควบคุม, การเขียนโปรแกรม PLC หลากหลายยี่ห้อ, การติดตั้งระบบต่าง ๆ รวมไปถึงการซ่อมบำรุงและการปรับปรุงระบบ (Revamp) อย่างบริษัท ร็อคเวล ออโตเมชั่น ไทย จำกัด ได้ทำการจ้างบริษัท พีเอส เอนจิเนียริ่ง คอนซัลแตนท์ จำกัด ให้ดำเนินการออกแบบศูนย์กลางรวบรวมข้อมูล สำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์ของ บริษัท มาร์ส เพ็ทแคร์ (ประเทศไทย) จำกัด ซึ่งศูนย์กลางรวบรวมข้อมูล ทำหน้าที่ในการรับส่งข้อมูลระหว่างระบบ MES กับระบบ PCS และรวบรวมข้อมูลจากเครื่องจักรแต่ละตัวในระบบ PCS ส่งต่อข้อมูลที่ได้ให้กับ MES เพื่อนำข้อมูลที่ได้ไปใช้ในการวิเคราะห์และควบคุมการผลิตให้เกิดประสิทธิภาพสูงสุด โดยศูนย์กลางรวบรวมข้อมูลจะใช้ PLC ในการควบคุมการทำงาน และในส่วนของการแสดงผลผ่านระบบสกาดาจะแสดงข้อมูลที่ส่งให้กับเครื่องจักรในระบบ PCS และแสดงขั้นตอนการทำงานของศูนย์กลางรวบรวมข้อมูล โดยซอฟแวร์ที่ใช้ในการออกแบบหน้าจอเพื่อแสดงผลข้อมูลคือโปรแกรม FactoryTalk View Studio และซอฟแวร์ที่ใช้ในการเขียนโปรแกรม PLC คือโปรแกรม Studio5000

สำหรับโครงงานสหกิจศึกษาฉบับนี้จึงได้จัดทำขึ้นเพื่อศึกษาด้านซอฟแวร์เป็นหลัก เพื่อให้นักศึกษาได้เรียนรู้เกี่ยวกับการเขียนโปรแกรมควบคุมการทำงานของ PLC และการออกแบบหน้าจอแสดงผลข้อมูลผ่านระบบสกาดา รวมทั้งการเชื่อมต่อศูนย์กลางรวบรวมข้อมูลเข้ากับระบบ MES และ PCS ทั้งนี้เพื่อให้มีความปลอดภัยในการจัดเก็บและเรียกใช้ข้อมูล จึงต้องมีการศึกษาและทำความเข้าใจในหลักการทำงานของศูนย์กลางรวบรวมข้อมูลเป็นอย่างดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อศึกษาหลักการทำงานของศูนย์กลางรวบรวมข้อมูล

1.2.2 เพื่อออกแบบศูนย์กลางรวบรวมข้อมูล สำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์ให้สามารถใช้งานได้จริง

1.2.3 เพื่อออกแบบหน้าจอแสดงผลผ่านระบบสกาดาให้สามารถแสดงผลข้อมูลได้อย่างถูกต้อง

1.2.4 เพื่อฝึกตนเองให้เป็นผู้มีความมุ่งมั่น ตั้งใจ อดทนและตรงต่อเวลา ซึ่งตรงกับอัตลักษณ์ของสถาบัน “ซื่อสัตย์ ใฝ่รู้ สู้งาน”

1.3 ขอบเขตของโครงงาน

1.3.1 ศึกษาการใช้โปรแกรม Studio5000 เพื่อนำมาใช้ในการเขียนโปรแกรมควบคุมการทำงานของศูนย์กลางรวบรวมข้อมูล

1.3.2 ออกแบบหน้าจอแสดงผลข้อมูล โดยใช้โปรแกรม FactoryTalk View Studio

1.3.3 ศึกษาและออกแบบศูนย์กลางรวบรวมข้อมูลโดยใช้ PLC ในการควบคุมการทำงาน

1.4 วิธีการดำเนินการโครงงาน

1.4.1 ศึกษาการใช้งานโปรแกรม Studio5000

1.4.2 ศึกษาหลักการทำงานของศูนย์กลางรวบรวมข้อมูล

1.4.3 จัดทำเอกสารการจับคู่ข้อมูล (Data Mapping) และกำหนดที่อยู่ (Address) ให้กับพารามิเตอร์ของกระบวนการ

1.4.4 เขียนโปรแกรมควบคุมการทำงานของศูนย์กลางรวบรวมข้อมูล

1.4.5 เขียนหน้าจอกราฟิกแสดงผลข้อมูล

1.4.6 เชื่อมต่อศูนย์กลางรวบรวมข้อมูลในโปรแกรม Studio5000 กับ PLC

1.4.7 เชื่อมต่อกราฟิกที่ใช้แสดงผลข้อมูลในโปรแกรม FactoryTalk View Studio กับ PLC

1.4.8 ทดสอบการทำงานของระบบ

1.4.9 Factory Acceptance Test (FAT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 1.1 แผนการดำเนินงาน

แผนการดำเนินงาน	เดือน	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
	สัปดาห์	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ศึกษาโปรแกรมที่ใช้ในโครงงาน																	
ศึกษาการทำงานของศูนย์กลางรวบรวมข้อมูล																	
จัดทำเอกสารการจับคู่ข้อมูล																	
เขียนโปรแกรมควบคุมการทำงานของศูนย์กลางรวบรวมข้อมูล																	
เขียนหน้าจอกราฟิกแสดงผลข้อมูล																	
เชื่อมต่อโปรแกรม Studio5000 กับ PLC																	
เชื่อมต่อโปรแกรม FactoryTalk View Studio กับ PLC																	
ทดสอบการทำงานของระบบ																	
Factory Acceptance Test																	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ผู้ปฏิบัติงานสามารถใช้งานศูนย์กลางรวบรวมข้อมูลได้จริง

1.5.2 ได้รับความรู้และความเข้าใจในกระบวนการทำงานของศูนย์กลางรวบรวมข้อมูล

1.5.3 ศูนย์กลางรวบรวมข้อมูลสามารถรับส่งข้อมูลกับ MES และ PCS ได้อย่างถูกต้อง

1.5.4 หน้าจอสามารถแสดงผลข้อมูลและแสดงขั้นตอนการทำงานของศูนย์กลางรวบรวมข้อมูลผ่านระบบสกาดาได้อย่างถูกต้อง

1.5.5 ฝึกการทำงานอย่างเป็นระบบ และสามารถแก้ไขปัญหาที่เกิดขึ้นได้

1.5.6 ฝึกตนเองให้เป็นผู้มีความมุ่งมั่น ตั้งใจ อดทนและตรงต่อเวลา ซึ่งตรงกับอัตลักษณ์ของสถาบัน “ซื่อสัตย์ ใฝ่รู้ สู้งาน”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

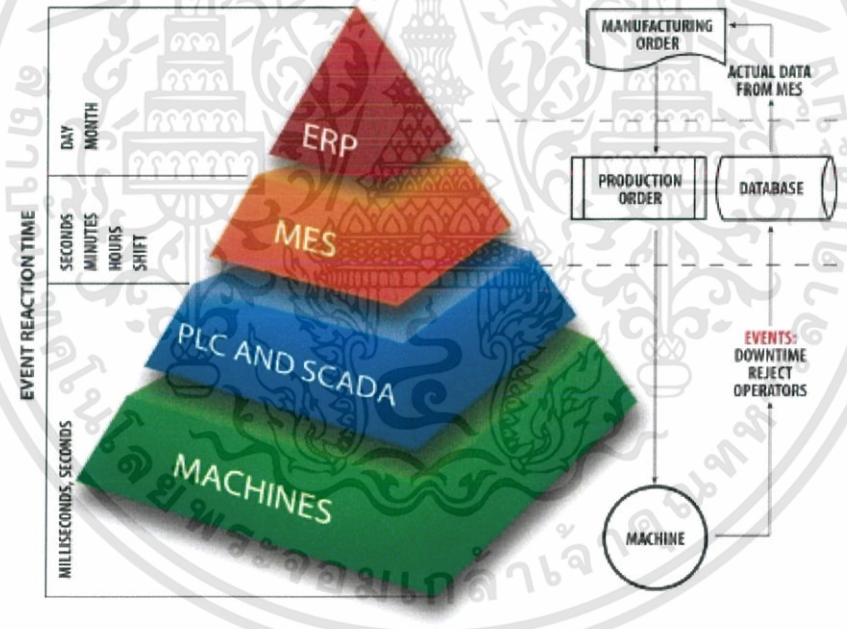
บทที่ 2
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 กล่าวนำ

ในการออกแบบศูนย์กลางรวบรวมข้อมูลที่มี PLC เป็นคอนโทรลเลอร์จะต้องมีความรู้และความเข้าใจในเรื่องต่าง ๆ เพื่อให้สามารถสร้างศูนย์กลางรวบรวมข้อมูลที่สามารถใช้งานได้จริง ซึ่งความรู้พื้นฐานที่ต้องใช้ประกอบด้วย โครงสร้างของระบบอัตโนมัติ, PLC, ภาษาที่ใช้ในการเขียนโปรแกรม และโปรโตคอลที่ใช้ในการสื่อสารระหว่างศูนย์กลางรวบรวมข้อมูลกับระบบ MES และ PCS รวมถึงโปรแกรมที่ใช้ในการทำงานในโครงงานนี้

2.2 โครงสร้างของระบบอัตโนมัติ

โครงสร้างของระบบอัตโนมัติประกอบไปด้วยระบบต่าง ๆ ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 โครงสร้างของระบบอัตโนมัติ

2.2.1 Enterprise Resource Planning (ERP) [1]

ERP เป็นระบบการวางแผนเพื่อให้กระบวนการผลิต การซื้อขายและการใช้ทรัพยากรต่าง ๆ ขององค์กร ได้อย่างมีประสิทธิภาพและมีประโยชน์สูงสุด ทั้งนี้เพื่อตอบสนองต่อความต้องการของลูกค้าและชิงความได้เปรียบในทางธุรกิจ ซึ่งในกระบวนการผลิตเมื่อได้รับคำสั่งมาจากระบบ ERP แล้วระบบคอมพิวเตอร์จะเชื่อมต่อการสื่อสารโดยอัตโนมัติ เพื่อให้การวางแผนจากระบบ ERP เชื่อมต่อกับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมการผลิตหรือระบบ MES โดยจะทำให้การเชื่อมต่อระหว่างการผลิตและการวางแผนสามารถทำงานสอดประสานกันได้โดยอัตโนมัติ

2.2.2 Manufacturing Execution Systems (MES) [2]

MES หรือ ระบบดำเนินการผลิต คือ ระบบการคำนวณที่ใช้ในการผลิต โดยระบบ MES ทำงานในแบบเวลาจริงเพื่อที่สามารถควบคุมองค์ประกอบต่าง ๆ ในการผลิตได้ในเวลาเดียวกัน เช่น การนำเข้าข้อมูล ส่วนบุคคล เครื่องจักรและการสนับสนุน ระบบ MES อาจทำงานในพื้นที่ต่าง ๆ เช่น การบริหารจัดการสินค้าตลอดอายุการใช้งาน การกำหนดแหล่งที่มา การสั่งซื้อและการจัดส่ง การวิเคราะห์การผลิตสำหรับประสิทธิภาพอุปกรณ์ในภาพรวมหรือ Overall Equipment Effectiveness (OEE) รวมทั้งการติดตามและการสืบค้นอุปกรณ์ ทั้งนี้ระบบ MES ได้สร้างขั้นตอนการผลิตที่ไม่มีข้อผิดพลาดและให้บริการตอบรับในแบบเวลาจริงสำหรับการเปลี่ยนแปลงต่าง ๆ ที่ต้องการ

2.2.3 Concentrator [3]

คอนเซนเทรเตอร์ (Concentrator) มีชื่อเรียกอีกอย่างหนึ่งว่าหน่วยประมวลผลทางการสื่อสาร (Communications Processor) โดยมากจะเป็นคอมพิวเตอร์อีกตัวหนึ่งที่ทำหน้าที่เฉพาะ ส่วนใหญ่จะมีหน่วยความจำสำรองพ่วงติดอยู่กับคอนเซนเทรเตอร์ ซึ่งทำหน้าที่รวมข้อมูลที่ส่งเข้ามาด้วยความเร็วต่ำ จากนั้นจะนำข้อมูลที่รวมกันแล้ว ส่งผ่านสายส่งความเร็วสูงไปยังเครื่องคอมพิวเตอร์อีกต่อหนึ่ง ดังรูปที่ 2.2 จะแสดงให้เห็นว่าคอนเซนเทรเตอร์ทำหน้าที่รับส่งข้อมูลระหว่างระบบ MES กับระบบ PCS

รูปที่ 2.2 คอนเซนเทรเตอร์ในระบบเครือข่าย

2.2.3.1 หลักการทำงานของคอนเซนเทรเตอร์

คอนเซนเทรเตอร์เป็นอุปกรณ์อีกชนิดหนึ่งที่ทำหน้าที่คล้ายกับมักซ์ คือ รวมสัญญาณจากสายสื่อสารหลายเส้นเข้าด้วยกันเพื่อส่งออกทางสายสื่อสารเพียงเส้นเดียว แต่ก็มีความต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรายละเอียดและวิธีนำมาใช้งาน มักซ์จำเป็นต้องใช้งานเป็นคู่เสมอในขณะที่คอนเซนเทรเตอร์ใช้เพียงเครื่องเดียวและยังมีขีดความสามารถในการประมวลผลและเก็บข้อมูลได้ด้วยซึ่งมีกลไกหลายขั้นตอนดังนี้

1) การใช้บัฟเฟอร์ (Buffering) ข้อมูลที่ส่งมายังคอนเซนเทรเตอร์มาจากหลายอุปกรณ์และหลายรูปแบบ ดังนั้นจึงต้องมีการจัดเก็บข้อมูลด้วยบัฟเฟอร์เพื่อผ่านการจัดการของคอนเซนเทรเตอร์ต่อไป
2) จองเนื้อที่หน่วยความจำและควบคุมการจัดคิว (Allocation of Storage and Control of Queues) การจองเนื้อที่หน่วยความจำนี้บางครั้งมีขั้นตอนที่สลับซับซ้อนมากบางครั้งข้อมูลจากอุปกรณ์รับส่งข้อมูลปลายทางหลาย ๆ เครื่องมีการส่งเข้ามาพร้อม ๆ กัน ซึ่งจะใช้วิธีแก้ปัญหาโดยการจองเนื้อที่หน่วย ความจำแบบไม่คงที่ (Dynamic allocation) จากนั้นก็จะมีการจัดคิวการทำงานที่จัดการกับข้อมูลก่อนหลัง แล้วจึงส่งผ่านกระแสข้อมูลที่รวมกันแล้วไปยังเครื่องคอมพิวเตอร์
3) รับข่าวสารจากอุปกรณ์รับส่งข้อมูลปลายทาง คอนเซนเทรเตอร์จะมีวงจรที่ต่อพ่วงกับอุปกรณ์ที่ส่งผ่านข้อมูลด้วยความเร็วต่ำหลายเครื่อง และต้องคอยตรวจสอบว่าเมื่อไรจะมีข้อมูลส่งเข้ามา การรอคอยข้อมูลเข้านี้เป็นไปในลักษณะที่ไม่แน่นอนว่าจะเกิดขึ้นเมื่อใดและมาจากสายส่งเส้นไหน ด้วยเหตุนี้คอนเซนเทรเตอร์จึงต้องมีการตรวจหา (Scan) ไปตามสายต่าง ๆ ด้วยความเร็วสูงเพื่อช่วยป้องกันสัญญาณสูญหายหรือผิดเพี้ยนไป
4) รวมข้อมูลเพื่อส่งผ่านในสายส่งความเร็วสูง เพื่อรวบรวมข้อมูลที่ได้รับมาเปลี่ยนรหัส จากนั้นก็จะจัดข้อมูลเป็นกลุ่มโดยจะต้องให้เครื่องคอมพิวเตอร์ทราบด้วยว่าข้อมูลกลุ่มนั้นมาจากสถานีไหนจึงต้องเพิ่มรหัสประจำสถานีไว้ที่ส่วนต้นของกลุ่มข้อมูลจึงจะอยู่ในสภาพพร้อมที่จะส่งข้อมูล
5) ตรวจสอบข้อผิดพลาดในการส่งข้อมูลด้วยความเร็วสูงแบบซิงโครนัส ซึ่งจะมีการตรวจสอบโดยใช้แพริตี้บิต

จากหลักการจะเห็นได้ว่า คอนเซนเทรเตอร์เครื่องหนึ่งถูกนำมาวางไว้ระหว่างโฮสต์และเทอร์มินอลจำนวนหนึ่ง คอนเซนเทรเตอร์จะรับข้อมูลเข้ามาจากเครื่องเทอร์มินอลทำการตรวจสอบความถูกต้องของข้อมูลและเก็บข้อมูลนั้นไว้เพื่อรอการนำส่งและส่งข้อมูลนั้นไปยังโฮสต์ในที่สุด ถ้าโฮสต์และเทอร์มินอลอยู่ห่างจากกัน คอนเซนเทรเตอร์จะถูกวางไว้ที่ฝั่งเทอร์มินอลเพื่อให้มีสายสื่อสารเพียงเส้นเดียวเชื่อมไปที่โฮสต์ เนื่องจากมีตัวประมวลผลและหน่วยบันทึกข้อมูลในตัวเอง คอนเซนเทรเตอร์จึงสามารถทำงานร่วมกับเทอร์มินอลได้ แม้ว่าสายสื่อสารที่ติดต่อกับโฮสต์จะเสียหายหรือถูกยกเลิกเป็นการชั่วคราว ข้อมูลจากเทอร์มินอลจะถูกเก็บรักษาไว้อย่างดีและนำส่งต่อไปยังโฮสต์เมื่อสายสื่อสารสามารถใช้งานได้ตามปกติ การโต้ตอบระหว่างผู้ใช้กับโปรแกรมบางอย่างที่เทอร์มินอลก็สามารถทำงานได้โดยใช้ตัวประมวลผลที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คอนเซนเทรเตอร์แทนได้ นอกจากนั้นแล้วคอนเซ็นเทรเตอร์สามารถเชื่อมต่อเทอร์มินอลจำนวนหนึ่งเข้ากับโฮลต์ หลายเครื่องเพื่อเพิ่มประสิทธิภาพในการทำงาน

2.2.4 Supervisory Control and Data Acquisition (SCADA) [4]

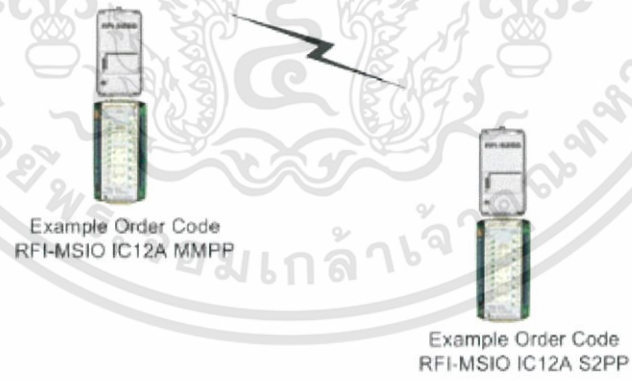
สกาดาย่อมาจากคำว่า Supervisory Control and Data Acquisition คือ ระบบการส่งข้อมูลในระยะไกล เพื่อใช้ในการตรวจสอบ เก็บข้อมูล และควบคุมกระบวนการผลิตต่าง ๆ ที่มีหน่วยควบคุมอยู่ห่างไกลกับกระบวนการผลิต โดยจะมีการสื่อสารข้อมูลแบบดิจิตอลผ่านทางระบบเครือ ข่ายคมนาคม องค์ประกอบหลักของสกาดาได้แก่ หน่วยติดต่อและปฏิบัติการของผู้ใช้ระดับบนหน่วยควบคุมระยะไกล หน่วยติดต่อระยะไกล และกระบวนการผลิต

ระบบสกาดาเป็นการรวมขบวนการ 2 ขบวนการเข้าด้วยกัน คือ

1) Telemetry System เป็นเทคนิคที่ใช้ในการส่งและรับข้อมูลผ่านสื่อกลาง โดยข้อมูลนั้นสามารถวัดได้ ข้อมูลเหล่านี้จะถูกส่งไปอีกสถานที่หนึ่งโดยผ่านสื่อกลางต่าง ๆ เช่น เคเบิล สายโทรศัพท์ หรือคลื่นวิทยุ
2) Data Acquisition เป็นวิธีการเข้าถึงและควบคุมข้อมูลจากอุปกรณ์ที่ถูกควบคุม หรือถูกตรวจสอบอยู่ โดยที่ข้อมูลที่ได้จะถูกส่งไปให้ระบบ Telemetry System เพื่อทำการส่งต่อไป

ระบบสกาดาแบ่งออกเป็นสองรูปแบบ คือ

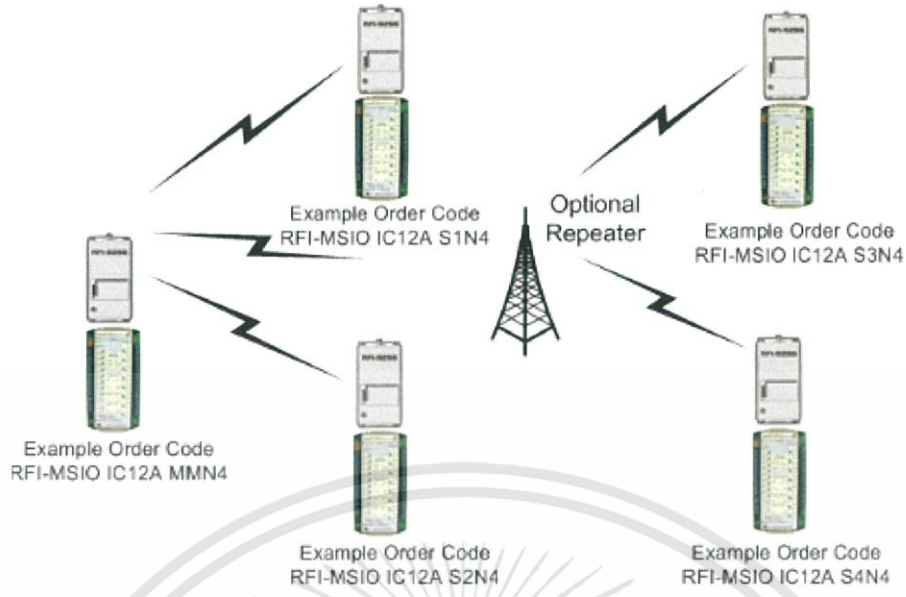
1) Point-to-Point Configuration เป็นการควบคุมที่ใช้หน่วยควบคุมในการควบคุมกระบวนการผลิตเพียงกระบวนการเดียว ดังรูปที่ 2.3



รูปที่ 2.3 Point-to-Point Configuration

2) Point-to-Multipoint Configuration เป็นการควบคุมใช้หน่วยควบคุมเดียวในการควบคุมกระบวนผลิตการหลายกระบวนการ ดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 Point-to-Multipoint Configuration

2.2.4.1 ส่วนประกอบของสกาดา

1) Field Instrumentation เป็นส่วนของเครื่องมือหรือเซนเซอร์ที่เชื่อมต่อกับเครื่องจักรหรืออุปกรณ์ที่ถูกควบคุมหรือตรวจสอบ โดยจะเปลี่ยนค่าปริมาณทางฟิสิกส์ให้เป็นปริมาณทางไฟฟ้า ซึ่งอาจจะอยู่ในรูปของอนาล็อกหรือดิจิตอล
2) Remote Station เป็นส่วนที่ทำการรวบรวมข้อมูลจากเครื่องจักรหรืออุปกรณ์ และส่งไปยังศูนย์กลางระบบสกาดา (SCADA)
3) Communication Network เป็นการส่งข้อมูลดิจิตอลระหว่างสถานที่หนึ่งไปยังสถานที่หนึ่ง โดยผ่านตัวกลางในการติดต่อสื่อสาร เช่น สายเคเบิล คลื่นวิทยุ
4) Central Monitoring Station (CMS) เป็นศูนย์กลางระบบสกาดาโดยรับข้อมูลมาประมวลผล และทำการแสดงกระบวนการบนหน้าจอคอมพิวเตอร์ ประกอบด้วยซอฟต์แวร์ และฮาร์ดแวร์

2.2.4.2 ฐานข้อมูลของสกาดา

1) Real-time Database Servers เป็นระบบฐานข้อมูลที่ใช้จัดการและเก็บค่าของกระบวนการ ณ เวลาปัจจุบันในขณะใด ๆ ค่าเวลาจริงจะเปลี่ยนแปลงไปตามสภาพของกระบวนการที่เปลี่ยนแปลงไปตามเวลา
2) Historical Database Servers เป็นระบบฐานข้อมูลที่ใช้จัดการและจัดเก็บค่า Historical Dataของกระบวนการเพื่อใช้ในการ Trending, Logging,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Statistic และ Report มาตรฐาน Protocol ของสกาดาปัจจุบันมีสกาดามาตรฐาน Protocols มากกว่า 200 โปรโตคอลทั่วโลก

2.2.4.3 มาตรฐานโปรโตคอลที่ใช้กันในปัจจุบัน

1) ASCII (American Standard Code for Information Interchange) เป็นโปรโตคอลที่ใช้ในการสื่อสารของคอมพิวเตอร์ที่รู้จักกันอย่างแพร่หลาย และเป็นสากล
2) CAP (Compressed ASCII Protocol) เป็น RTU Protocol ที่ดีที่สุด เป็นภาษาที่คนสามารถเข้าใจได้ มีความน่าเชื่อถือ เร็ว และมีความปลอดภัยสูง
3) Modbus เป็น point-to-point PLC protocol ที่ใช้กันทุกแห่ง แต่มีข้อเสียคือ เป็นภาษาที่คนไม่สามารถอ่านเข้าใจได้
4) Modbus X พัฒนามาจาก Modbus ทำให้สามารถอ่านและสามารถสร้างจำนวนบวกและลบได้
5) IEEE 32 บิต Signal Format Floating Point เป็นมาตรฐานของโรงงานอุตสาหกรรม สำหรับส่งตัวเลข 32 บิต ด้วยความถูกต้อง

2.2.4.4 องค์ประกอบของสกาดา

ผู้ใช้สามารถตรวจสอบและควบคุมกระบวนการผลิตภายในโรงงานอุตสาหกรรมเป็นระยะทางไกลได้โดย หน่วยติดต่อและปฏิบัติการของผู้ใช้ระดับบนเป็นเครื่องมือปฏิบัติการของผู้ใช้สำหรับตรวจสอบและควบคุมกระบวนการผลิตเชื่อมต่อกับหน่วยควบคุมระยะไกล หน่วยควบคุมระยะไกลติดต่อกับหน่วยติดต่อระยะไกลโดยการสื่อสารข้อมูลแบบดิจิตอลทางระบบ

2.2.4.5 งานที่ต้องใช้สกาดา

งานการตรวจสอบ การเก็บรวบรวมข้อมูลของกระบวนการผลิต และการบริหารระบบควบคุมกลุ่มโรงงานอุตสาหกรรมขนาดใหญ่บริเวณกระบวนการผลิตครอบคลุมพื้นที่กว้าง หรือโรงงานอุตสาหกรรมมีกระบวนการผลิตอิสระติดตั้งกระจัดกระจายทั่วบริเวณพื้นที่การผลิต รวมถึงระบบสาธารณูปโภคต่าง ๆ

2.2.4.7 ประโยชน์ของการนำระบบสกาดามาใช้

ในปัจจุบันหลายหน่วยงานได้มีการนำระบบสกาดานี้ มาใช้ในการเฝ้าดูข้อมูลเพื่อช่วยในการตัดสินใจในการควบคุมระบบ เช่น การไฟฟ้านครหลวง , การประปานครหลวง , ปตท. ฯลฯ ซึ่งเหตุผลที่หลายหน่วยงานเลือกใช้ระบบนี้เพราะเหตุผลต่าง ๆ ดังต่อไปนี้

1) การควบคุมจะเป็นไปอย่างต่อเนื่องและครอบคลุมตลอดพื้นที่
2) เพิ่มประสิทธิภาพในการตรวจสอบและการบำรุงรักษาเครื่องจักรกับอุปกรณ์เนื่องจากสามารถทำการบำรุงรักษา หรือทำ PM ก่อนที่เครื่องจักรจะชำรุด
3) ก่อให้เกิดเสถียรภาพและความน่าเชื่อถือไว้วางใจ
4) ลูกค้าจะได้รับบริการอันรวดเร็ว ทันสมัย แม่นยำ และยุติธรรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) การติดตามข้อมูล และการประเมินผลต่าง ๆ เป็นไปอย่างรวดเร็วและถูกต้อง

6) ประหยัดแรงงานและกำลังคน และทำให้ประหยัดค่าใช้จ่าย

2.3 Programmable Logic Controller (PLC) [5]

PLC เป็นอุปกรณ์ชนิดโซลิด – สเตท (Solid State) ที่ทำงานแบบลอจิก (Logic Functions) การออกแบบการทำงานของ PLC จะคล้ายกับหลักการทำงานของคอมพิวเตอร์ จะประกอบ ด้วยอุปกรณ์ที่เรียกว่า Solid-State Digital Logic Elements ใช้สำหรับควบคุมกระบวนการทำงานของเครื่องจักรและอุปกรณ์ในโรงงานอุตสาหกรรม

การใช้ PLC สำหรับควบคุมเครื่องจักรหรืออุปกรณ์ต่าง ๆ ในโรงงานอุตสาหกรรมจะมีข้อได้เปรียบกว่าการใช้ระบบของรีเลย์ (Relay) ซึ่งจำเป็นจะต้องเดินสายไฟฟ้า หรือที่เรียกว่า Hard Wired ฉะนั้นเมื่อมีความจำเป็นที่ต้องเปลี่ยนกระบวนการผลิตหรือลำดับการทำงานใหม่ ก็ต้องเดินสาย ไฟฟ้าใหม่ ซึ่งเสียเวลาและเสียค่าใช้จ่ายสูง แต่เมื่อเปลี่ยนมาใช้ PLC แล้ว การเปลี่ยนกระบวนการผลิตหรือลำดับการทำงานใหม่นั้นทำได้โดยการเปลี่ยนโปรแกรมใหม่เท่านั้น นอกจากนี้แล้ว PLC ยังใช้ระบบโซลิด – สเตท ซึ่งน่าเชื่อถือกว่าระบบเดิม การกินกระแสไฟฟ้าน้อยกว่า และสะดวกกว่าเมื่อต้องการขยายขั้นตอนการทำงานของเครื่องจักร

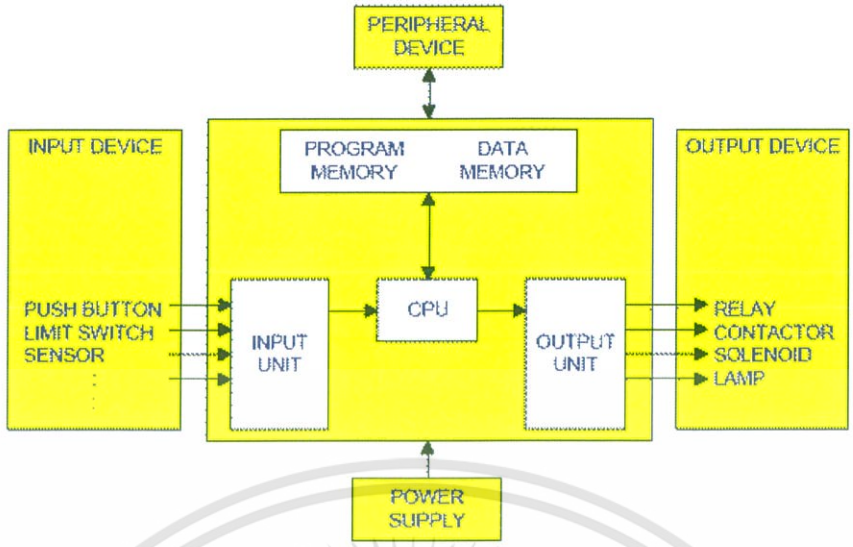
2.3.1 ความหมายของ PLC

PLC เป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการทำงานต่าง ๆ โดยภายในมี Microprocessor เป็นสมองสั่งการที่สำคัญ PLC จะมีส่วนที่เป็นอินพุตและเอาต์พุตที่สามารถต่อออกไปใช้งานได้ทันที ตัวตรวจวัดหรือสวิตช์ต่าง ๆ จะต่อเข้ากับอินพุต ส่วนเอาต์พุตจะใช้ต่อออกไปควบคุมการทำงานของอุปกรณ์หรือเครื่องจักรที่เป็นเป้าหมาย เราสามารถสร้างวงจรหรือแบบของการควบคุมได้โดยการป้อนเป็นโปรแกรมคำสั่งเข้าไปใน PLC นอก จากนี้ยังสามารถใช้งานร่วมกับอุปกรณ์อื่นเช่นเครื่องอ่านบาร์โค๊ด (Barcode Reader) เครื่องพิมพ์ (Printer) ซึ่งในปัจจุบันนอกจาก PLC จะใช้งานแบบเดี่ยว (Standalone) แล้วยังสามารถต่อ PLC หลาย ๆ ตัวเข้าด้วยกัน (Network) เพื่อควบคุมการทำงานของระบบให้มีประสิทธิภาพมากยิ่งขึ้นด้วยจะเห็นได้ว่าการใช้งาน PLC มีความยืดหยุ่นมาก ดังนั้นในโรงงานอุตสาหกรรมต่าง ๆ จึงเปลี่ยนมาใช้ PLC มากขึ้น

2.3.2 โครงสร้างของ PLC [6]

ลักษณะโครงสร้างภายในของ PLC ประกอบไปด้วยส่วนต่างๆ ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 ส่วนประกอบของ PLC

2.3.2.1 ตัวประมวลผล (CPU)

ทำหน้าที่คำนวณและควบคุม ซึ่งเปรียบเสมือนสมองของ PLC ภายในประกอบด้วยวงจรลอจิกหลายชนิดและมีไมโครโปรเซสเซอร์เบส (Micro Processor Based) ใช้แทนอุปกรณ์จำพวกรีเลย์ เคาน์เตอร์/ไทม์เมอร์ และซีเควนเซอร์ เพื่อให้ผู้ใช้สามารถออกแบบวงจรโดยใช้ Relay Ladder Diagram ได้ CPU จะยอมรับข้อมูลจากอุปกรณ์อินพุทต่าง ๆ จากนั้นจะทำการประมวลผลและเก็บข้อมูลโดยใช้โปรแกรมจากหน่วยความจำ หลังจากนั้นจะส่งข้อมูลที่เหมาะสมและถูกต้องออกไปยังอุปกรณ์เอาท์พุท

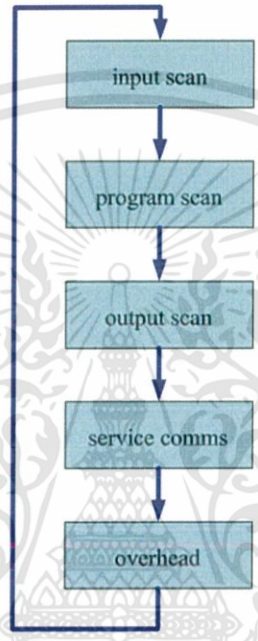
สำหรับโปรแกรมลอจิกที่เราป้อนลงใน PLC นั้น เมื่อ PLC ทำงาน การประมวลผลโปรแกรมของ PLC นั้นจะเป็นรอบ เราเรียกว่ารอบการทำงาน (Operating Cycle) ในหนึ่งรอบการทำงานของ PLC นั้นจะมีการทำงานเป็นลำดับตามแสดงในรูปที่ 2.6 โดยมีรายละเอียดแต่ละขั้นตอนดังต่อไปนี้

1) สแกนอินพุต (Input Scan) เป็นเวลาที่ PLC ใช้สำหรับอ่านข้อมูลอินพุตของ PLC โดยเวลาที่ใช้นั้นเป็น ไมโครวินาที
2) สแกนโปรแกรม (Program Scan) เป็นเวลาที่ PLC ใช้สำหรับปฏิบัติการคำสั่งในโปรแกรมลอจิกก็คือโปรแกรมที่เราเขียนนั่นเอง เวลาที่ใช้นั้นขึ้นอยู่กับจำนวนคำ สั่งที่ใช้หรือจำนวนโปรแกรมที่เราเขียนว่ามีขนาดแค่ไหน
3) สแกนเอาต์พุต (Output Scan) เป็นเวลาที่ PLC ใช้สำหรับเขียนข้อมูลเอาต์พุตของ PLC โดยเวลาที่ใช้นั้นเป็นไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การบริการสื่อสาร (Service Communication) เป็นเวลาที่ PLC ใช้สื่อสารกับอุปกรณ์อื่นเช่น คอมพิวเตอร์ หรือ อุปกรณ์สำหรับโปรแกรม (Hand Help)

5) Housekeeping และ Overhead เป็นเวลาที่ PLC ใช้ในการจัดการดำเนินการให้ PLC มีความพร้อมในการใช้งาน เช่น การตรวจสอบและจัดระเบียบหน่วยความจำ และเวลาส่วนเกินที่จำเป็น



รูปที่ 2.6 ลำดับการทำงานในหนึ่งรอบการทำงานของ PLC

2.3.2.2 หน่วยความจำ

ประกอบด้วย หน่วยความจำชนิด RAM และ ROM หน่วยความจำชนิด RAM ทำหน้าที่เก็บโปรแกรมของผู้ใช้และข้อมูลสำหรับใช้ในการปฏิบัติงานขอ PLC ส่วน ROM ทำหน้าที่เก็บโปรแกรมสำหรับใช้ในการปฏิบัติงานของ PLC ตามโปรแกรมของผู้ใช้ ROM ย่อมาจาก Read Only Memory สามารถโปรแกรมได้แต่ลบไม่ได้ ถ้าชำรุดแล้วซ่อมไม่ได้

1) RAM (Random Access Memory) หน่วยความจำประเภทนี้จะมีแบตเตอรี่เล็ก ๆ ต่อไว้ เพื่อใช้เลี้ยงข้อมูลเมื่อเกิดไฟดับ การอ่านและเขียนโปรแกรมลงใน RAM ทำได้ง่ายมาก จึงเหมาะกับการใช้งานในระยะทดลองเครื่องที่มีการเปลี่ยนแปลงแก้ไขโปรแกรมบ่อย ๆ

2) EPROM (Erasable Programmable Read Only Memory) หน่วยความจำชนิด EPROM นี้จะต้องใช้เครื่องมือพิเศษในการเขียน

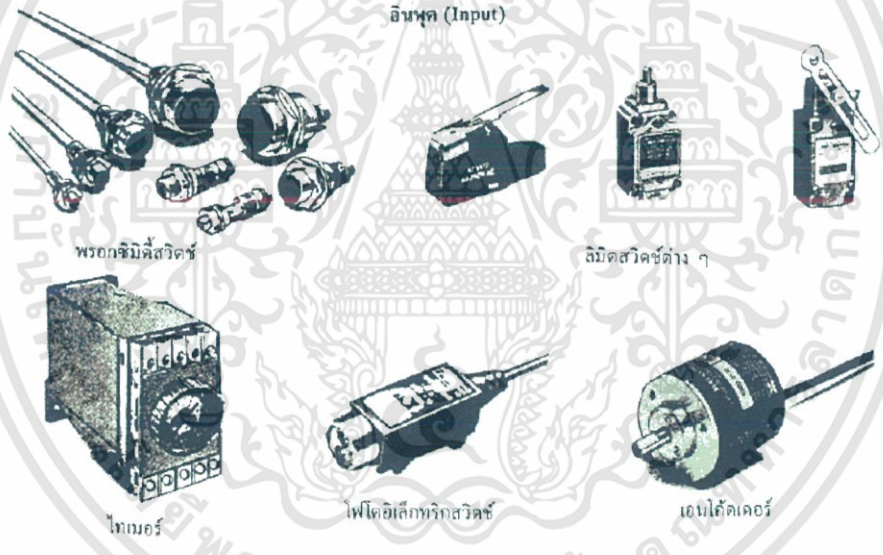
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม การลบโปรแกรมทำได้โดยใช้แสงอัลตราไวโอเลตหรือตากแดดร้อน ๆ นาน ๆ มีข้อดีตรงที่โปรแกรมจะไม่สูญหายแม้ไฟดับ จึงเหมาะกับการใช้งานที่ไม่ต้องเปลี่ยนโปรแกรม

3) EEPROM (Electrical Erasable Programmable Read Only Memory) หน่วยความจำชนิดนี้ไม่ต้องใช้เครื่องมือพิเศษในการเขียน และลบโปรแกรมโดยใช้วิธีการทางไฟฟ้าเหมือนกับ RAM นอกจากนั้นก็ไม่จำเป็นต้องมีแบตเตอรี่สำรองไฟเมื่อไฟดับ ราคาจะแพงกว่า แต่จะรวมคุณสมบัติที่ดีของทั้ง RAM และ EPROM เอาไว้ด้วยกัน

2.3.2.3 หน่วยอินพุต-เอาต์พุต (Input-Output Unit)

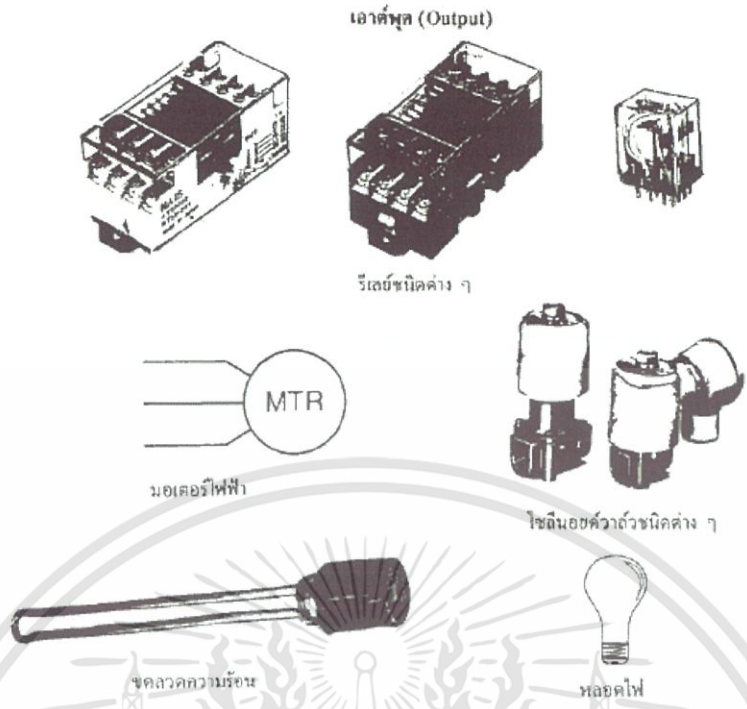
1) หน่วยอินพุต ทำหน้าที่ รับสัญญาณจากอุปกรณ์ภายนอกแล้วแปลงสัญญาณให้เป็นสัญญาณที่เหมาะสมแล้วส่งให้หน่วยประมวลผลต่อไป ตัวอย่างอุปกรณ์อินพุตดังรูปที่ 2.7



รูปที่ 2.7 ตัวอย่างอุปกรณ์อินพุตที่ต่อเข้ากับ PLC

2) หน่วยเอาต์พุต ทำหน้าที่รับข้อมูลจากตัวประมวลผลแล้วส่งต่อข้อมูลไปควบคุมอุปกรณ์ภายนอกเช่น ควบคุมหลอดไฟ มอเตอร์ และวาล์ว เป็นต้น ตัวอย่างอุปกรณ์เอาต์พุตดังรูปที่ 2.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ตัวอย่างอุปกรณ์เอาท์พุตที่ต่อเข้ากับ PLC

2.3.2.4 แหล่งจ่ายไฟ (Power Supply)

ทำหน้าที่จ่ายพลังงานและรักษาระดับแรงดันไฟฟ้ากระแสตรงให้กับ CPU Unit หน่วยความจำและหน่วยอินพุท/ เอาท์พุท

2.3.2.5 อุปกรณ์ต่อร่วม (Peripheral Devices)

1) PROGRAMMING CONSOLE
2) EPROM WRITER
3) PRINTER
4) GRAPHIC PROGRAMMING
5) CRT MONITOR
6) HANDHELD
7) อื่น ๆ

2.3.3 ความสามารถของ PLC

2.3.3.1 งานที่ทำตามลำดับก่อนหลัง (Sequence Control)

1) การทำงานของระบบรีเลย์
2) การทำงานของไทเมอร์ เคาน์เตอร์
3) การทำงานของ P.C.B. Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การทำงานในระบบกึ่งอัตโนมัติ ระบบอัตโนมัติ หรืองานที่เป็นกระบวนการทำงานของเครื่องจักรกลต่าง ๆ

2.3.3.2 งานควบคุมสมัยใหม่ (Sophisticated Control)

1) การทำงานทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร
2) การควบคุมแบบอนาล็อก (Analog Control) เช่น การควบคุมอุณหภูมิ (Temperature) การควบคุมความดัน (Pressure) เป็นต้น
3) การควบคุม P.I.D. (Proportional-Intergral-Derivation)
4) การควบคุมเซอร์โวมอเตอร์ (Sevo-motor Control)
5) การควบคุม Stepper-motor
6) Information Handling

2.3.3.3 การควบคุมเกี่ยวกับงานอำนวยการ (Supervisory Control)

1) งานสัญญาณเตือน (Alarm) และ Process Monitoring
2) Fault Diagnostic and Monitoring
3) งานต่อร่วมกับคอมพิวเตอร์ (RS-232C/RS422)
4) Printer/ASCII Interfacing
5) งานควบคุมอัตโนมัติในโรงงานอุตสาหกรรม
6) LAN (Local Area Network)
7) WAN (Wide Area Network)
8) FA., FMS., CIM. เป็นต้น

2.3.4 ขนาดของ PLC

เนื่องจากในปัจจุบันมีการนำ PLC ไปใช้งานอย่างกว้างขวาง ซึ่งในการนำเอา PLC ไปใช้งานในแต่ละชนิดนั้น จะพิจารณาจากขนาดของงานที่จะนำไปควบคุมเป็นหลัก ดังนั้นจึงเป็นผลให้ผู้ผลิต PLC ทำการผลิต PLC ออกมาหลากหลายระดับซึ่งในแต่ละระดับก็มีสมรรถภาพที่แตกต่างกัน เพื่อให้เหมาะสมกับการใช้งานในแต่ละประเภท

โดยทั่วไปแล้วนั้นการแบ่งขนาดของ PLC จะพิจารณาจากขนาดของหน่วยความจำโปรแกรม (Program memory) และจำนวนของอินพุต และเอาต์พุต (Input / Output channels) สูงสุดที่ระบบ PLC นั้น สามารถที่จะรองรับได้ ซึ่งจากตารางที่ 2.1 จะแสดงการจำแนก PLC ตามขนาด ของหน่วยความจำโปรแกรม และจำนวนของอินพุต และเอาต์พุต แต่อย่างไรก็ตามในการพิจารณาคุณสมบัติของ PLC เพื่อนำไปใช้งานจะต้องพิจารณาถึงองค์ ประกอบหรือคุณสมบัติอื่น ๆ ประกอบด้วย เช่น Processor, Language facilities, Communication port, Function operations, Expansion capability, Cycle time เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.1 การจำแนกขนาดของ PLC

ขนาดของ PLC	จำนวน I/O สูงสุด	หน่วยความจำโปรแกรม
ขนาดเล็ก (Small size)	ไม่เกิน 128 / 128	4 Kbyte (2,000 Statements)
ขนาดกลาง (Medium size)	ไม่เกิน 1024 / 1024	16 Kbyte (8,000 Statements)
ขนาดใหญ่ (Large size)	ไม่เกิน 2048 / 2048	64 Kbyte (32,000 Statements)
ขนาดใหญ่มาก (Very large size)	ประมาณ 8192 / 8192	256 Kbyte (128,000 Statements)

2.3.5 ภาษาที่ใช้เขียนโปรแกรม [7]

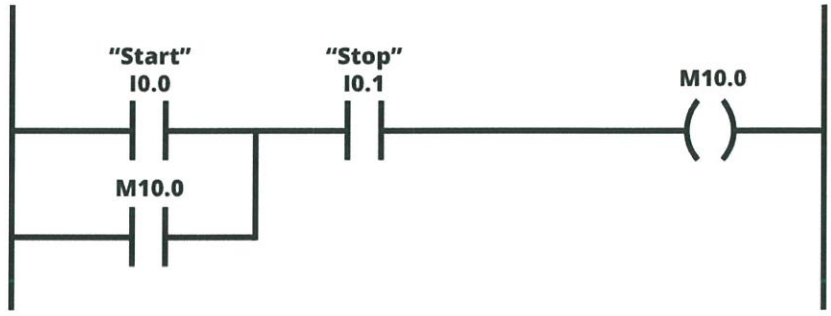
ภาษาที่ใช้ในการเขียนโปรแกรม ตามมาตรฐาน IEC จะประกอบด้วย 5 ภาษา คือ Ladder Diagram, Function Block Diagram, Instruction List, Structure Text, Sequential Function Chart ถึงแม้ว่าลักษณะโครงสร้างของแต่ละภาษาจะมีความ แตกต่างกัน แต่ในแต่ละภาษจะมีส่วนประกอบต่าง ๆ ในโปรแกรมมีลักษณะเดียวกัน ตามมาตรฐาน IEC 1131 – 3 เช่น ลักษณะของการประกาศตัวแปร ฟังก์ชั่น และฟังก์ชั่นบล็อก เป็นต้น แต่อย่างไร ก็ตามเราสามารถที่จะเขียนโปรแกรมในรูปแบบการเขียนต่าง ๆ มารวมกันได้และมีฟังก์ชั่นพื้นฐานดังนี้

1) LD เป็นการรับค่าเบื้องต้นของบล็อก
2) OR การ or หรือ การต่อแบบขนาน
3) AND การ and หรือ การต่อแบบอนุกรม
4) NOT การ not หรือ การกลับค่า
5) OUT เอาต์พุตแบบรีเลย์
6) TIM เอาต์พุตแบบไทเมอร์หรือตัวจับเวลา
7) CNT เอาต์พุตแบบเคาเตอร์หรือตัวนับ
8) KEEP เอาต์พุตแบบรีเลย์แบบมีแลตช์หรือค้างสถานะ
9) END คำสั่งจบโปรแกรม

2.3.5.1 LD (Ladder diagram)

แลดเดอร์ไดอะแกรมเป็นการเขียนอยู่ในรูปกราฟิก ซึ่งมีพื้นฐานมาจากวงจรควบคุมแบบรีเลย์และวงจรไฟฟ้า ซึ่งแลดเดอร์ไดอะแกรม จะประกอบด้วยราง (Rail) ทั้งซ้ายขวาของไดอะแกรมเพื่อใช้สำหรับเชื่อมต่อปกรณ์ที่เป็นสวิตช์หน้าสัมผัส เพื่อเป็นทางผ่านของกระแสและมีขดลวดหรือคอยล์เป็นเอาต์พุต ดังรูปที่ 2.9

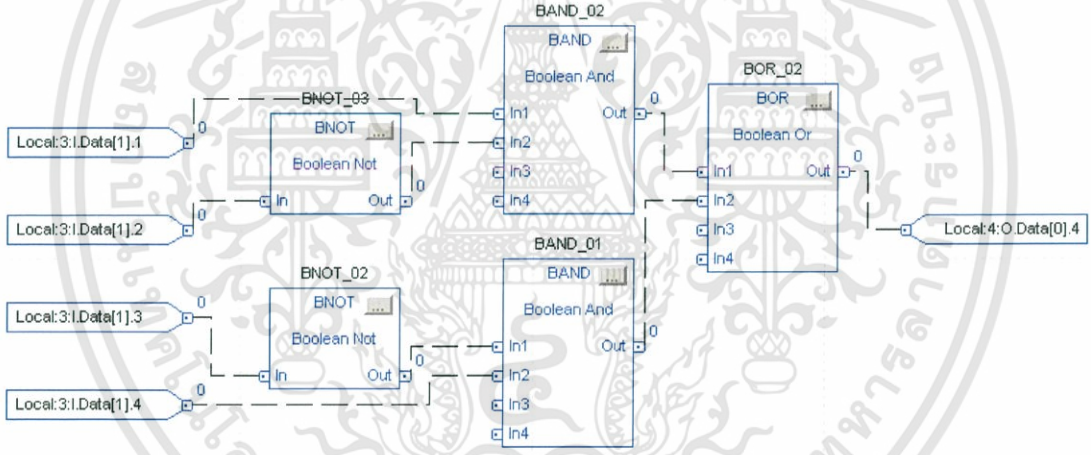
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 Ladder Diagram

2.3.5.2 FBD (Function block diagram)

FBD เป็นภาษาที่ฟังก์ชั่นการทำงานในรูปแบบในรูปของกราฟิกเช่นเดียวกับแลดเดอร์ไดอะแกรมและมาการเชื่อมต่อเข้ากันเป็นโครงข่าย ซึ่งในการเขียนโปรแกรมในรูปแบบของฟังก์ชั่นบล็อกไดอะแกรมจะมีพื้นฐานมาจากลอจิกไดอะแกรม ดังรูปที่ 2.10



รูปที่ 2.10 Function block diagram

2.3.5.3 IL (Instruction list)

IL เป็นภาษาที่เขียนในรูปแบบของข้อความ มีลักษณะคล้ายกับภาษาแอสแซมบี้ (Assembly) และภาษาเครื่อง (Machine code) ดังรูปที่ 2.11 และส่วนที่ถูกดำเนินการ (Operand) จะเห็นว่าในภาษาปัจจุบัน LD, FBD และ IL เป็นภาษาที่บริษัทผู้ผลิต PLC/PC ในปัจจุบันนั้นกำหนดให้ใช้ในการเขียนโปรแกรม PLC ซึ่งในแต่ละบริษัทจะมีการพัฒนารูปแบบของฟังก์ชั่นและฟังก์ชั่นบล็อกมีความแตกต่างกัน ดังนั้นในการเขียนโปรแกรมควบคุมการทำงานของ PLC ในแต่ละยี่ห้อ ควรที่จะศึกษารูปแบบของฟังก์ชั่นบล็อกให้เข้าใจเสียก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Label	Opcode	Operand	Comment
START:	LD	%I:000/00	(* Load input bit 00 *)
	AND(	%I:000/01	(* Start a branch and load input bit 01 *)
	OR(	%I:000/02	(* Load input bit 02 *)
	ANDN	%I:000/03	(* Load input bit 03 and invert *)
	)		
	)		
	ST	%O:001/00	(* SET the output bit 00 *)

รูปที่ 2.11 Instruction list

2.3.5.4 ST (Structure text)

ST เป็นภาษาในระดับสูงซึ่งโดยพื้นฐานแล้วมาจากภาษา Pascal ซึ่งจะประกอบด้วยนิพจน์และคำสั่งโดยคำสั่งทั่วไปแล้วจะอยู่ในรูปของคำสั่งเกี่ยวกับการเลือกการทำงาน เช่น IF...THEN...ELSE และ คำสั่งเกี่ยวข้องกับการทำงานซ้ำ เช่น FOR, WHILE เป็นต้น ดังรูปที่ 2.12

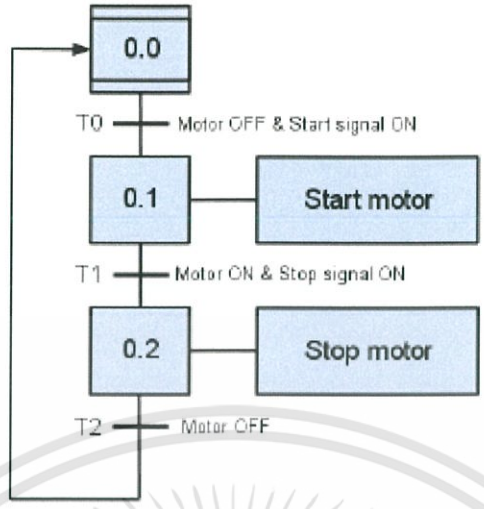
```
F8:10 := 0;
WHILE (N7:0 < 5) DO
  F8:10 := F8:10 + F8:[N7:0];
  N7:0 := N7:0 + 1;
END_WHILE;
```

รูปที่ 2.12 Structure text

2.3.5.5 SFC (Sequential Function)

SFC เป็นภาษาที่ในการเขียนโปรแกรมที่มีโครงสร้างการทำงานเป็นแบบลำดับหรือซีเควนซ์ ส่วนประกอบของภาษา SFC ประกอบด้วย Step (ปฏิบัติการย่อย) และTransition (เงื่อนไขที่กำหนดให้ปฏิบัติงานตามคำสั่งย่อย) ดังรูปที่ 2.13 นอกจากนี้ยังสามารถกำหนดลักษณะการทำงานเป็นแบบ Alternative, Parallel Step Sequence และ Liner เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



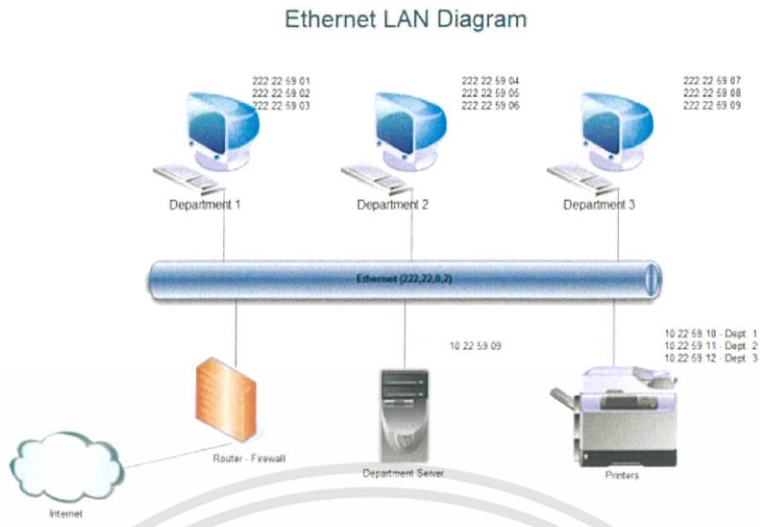
รูปที่ 2.13 SFC (Sequential Function)

2.4 โครงสร้างด้านการสื่อสาร (Communications)

2.4.1 Ethernet [8]

Ethernet เป็นเทคโนโลยีเครือข่ายที่ได้รับความนิยมมากเพราะเป็นการส่งข้อมูลด้วยความเร็วสูง ซึ่งในช่วงแรกที่มีการพัฒนาระบบ Ethernet สามารถที่จะส่งผ่านข้อมูลด้วยความเร็ว 10 เมกะบิตต่อวินาที (Mbps) แต่ปัจจุบันมีการพัฒนาเทคโนโลยีใหม่ที่เรียกว่า Fast Ethernet และ Gigabit Ethernet ที่ทำความเร็วได้ถึง 100 เมกะบิตต่อวินาที (Mbps) หรือ 1 Gbps และ 1000 เมกะบิตต่อวินาที (Mbps) หรือ 10 GbE ตามลำดับ โดยมีการควบคุมมาตรฐานของ Ethernet ด้วยสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers) หรือ IEEE ซึ่งเป็นองค์กรที่ไม่หวังผลกำไรที่ค่อยดูแลและพัฒนาเทคโนโลยีทางไฟฟ้าและคอมพิวเตอร์ซึ่งก็รวมถึง Ethernet ด้วยเช่นกัน ตัวอย่างการเชื่อมต่อแบบ Ethernet แสดงดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 Ethernet LAN Diagram

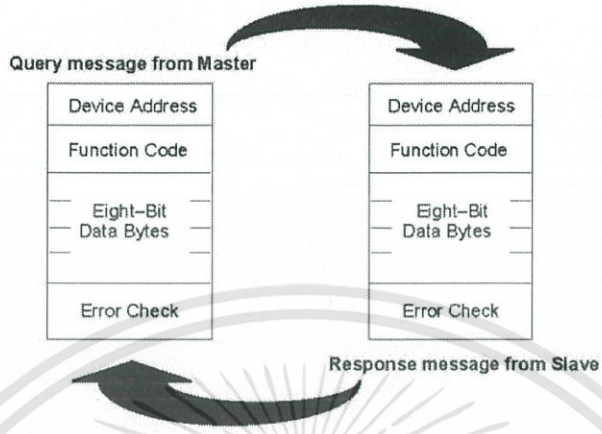
Ethernet เป็นการสื่อสารแบบโพรโทคอล (Protocol) ของ LAN ชนิดหนึ่งที่พัฒนาขึ้นโดย 3 บริษัทใหญ่คือบริษัท Xerox Corporation, Digital Equipment Corporation (DEC) และ Intel ในปี ค.ศ. 1976 หรืออาจจะเรียกการสื่อสารแบบนี้ว่าเป็นการสื่อสารระดับล่างก็ได้เช่นกัน ระบบการส่งแบบ Ethernet นั้นเป็นระบบการส่งที่เรียกว่า CSMA/CD (Carrier Sense Multiple Access / Collision Detection) โดยอธิบายหลักการทำงานได้ดังนี้ ในการส่งข้อมูลแต่ละครั้งจะทำการส่งได้เพียงคนเดียวเท่านั้น แต่ถ้าในเวลาเดียวกันมีการส่งข้อมูลมาพร้อมกัน มากกว่า 1 คนด้วยกัน ซึ่งเรียกว่า “Collision” อุปกรณ์คอมพิวเตอร์แต่ละตัวจะมีการตรวจสอบ Collision เมื่ออุปกรณ์คอมพิวเตอร์ตรวจสอบว่ามีการส่งข้อมูลมาพร้อมกัน อุปกรณ์เหล่านั้นจะหยุดส่งข้อมูลเพียงช่วงเวลาหนึ่ง และจะทำการส่งใหม่ เวลาที่หยุดรอนั้นจะเป็นการสุ่มแบบสถิติ ทำให้การเกิด Collision อีกครั้งจะเป็นไปได้น้อยมาก ๆ แต่ถ้าเกิดการ Collision จริง ๆ อุปกรณ์คอมพิวเตอร์เหล่านั้นก็จะวนกลับมารอและสุ่มเวลาอีกรอบจนกว่าจะไม่พบการ Collision อีก

2.4.2 Modbus [9]

โปรโตคอล MODBUS เป็นการสื่อสารข้อมูลในลักษณะ Master/Slave ดังแสดงในรูปที่ 2.15 ซึ่งเป็นการสื่อสารจากอุปกรณ์แม่ (Master) เครื่องเดียว ส่วนใหญ่มักเป็นซอฟต์แวร์คอมพิวเตอร์หรืออุปกรณ์แสดงผล HMI ไปยังอุปกรณ์ลูก (Slave) ได้หลาย ๆ เครื่อง โดยสามารถกำหนดหมายเลขอุปกรณ์ได้สูงสุด 255 เครื่อง โดยมีลักษณะการส่งข้อมูล 2 แบบ คือ ข้อมูลแบบแอสกี (ASCII) และข้อมูลแบบเลขฐานสอง (Binary) ในโปรโตคอล MODBUS ที่สื่อสารข้อมูลแบบ ASCII จะเรียก MODBUS ASCII และโปรโตคอล MODBUS ที่สื่อสารข้อมูลแบบเลขฐานสอง จะเรียก MODBUS RTU ทำให้มีความแตกต่างในการกำหนดค่าพอร์ตสื่อสาร การรับส่งข้อมูลด้วยโปรโตคอล MODBUS สามารถเลือกได้ 2 โหมด คือ โหมด ASCII และโหมด RTU ซึ่งทั้ง 2 โหมดนี้มีความแตกต่างกันที่การกำหนดรูปแบบของชุดข้อมูลภายในเฟรม จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เลือกโหมดใดก็ได้แต่มีเงื่อนไขว่า อุปกรณ์ทุกตัวที่ต่อร่วมกันอยู่ในบัสหรือเครือข่ายเดียวกัน จะต้องตั้งให้เลือกใช้โหมดเดียวกันทั้งหมด

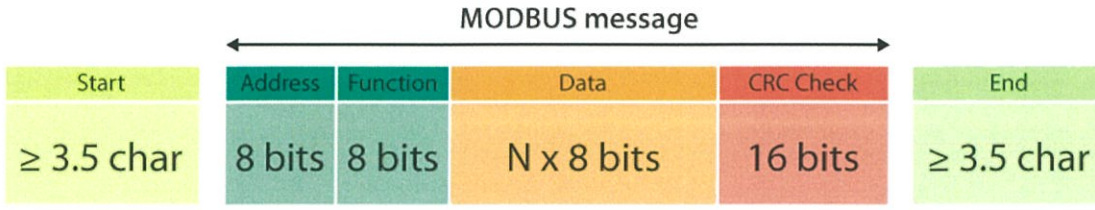


รูปที่ 2.15 การติดต่อสื่อสารแบบ Master/Slave

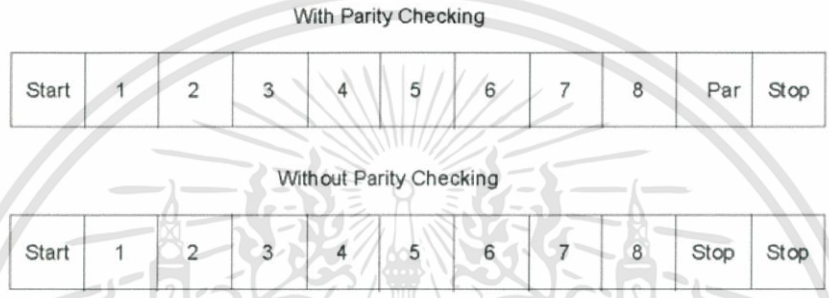
2.4.2.1 MODBUS RTU

เฟรมข้อมูลในโหมด RTU ประกอบด้วยข้อมูลแสดงตำแหน่งแอดเดรส 1 ไบต์, หมายเลขฟังก์ชัน 1 ไบต์, ข้อมูลที่ทำการรับส่งจำนวนมากสุดไม่เกิน 252 ไบต์ และรหัสตรวจสอบความถูกต้องของข้อมูลแบบ CRC (Cyclical Redundancy Checking) ขนาด 2 ไบต์ ค่า CRC นี้เป็นค่าที่คำนวณมาจากข้อมูลทุกไบต์ ไม่รวมบิต Start, Stop และ Parity Check ดังแสดงในรูปที่ 2.16 โดยที่ตัว Slave ตัวที่ส่งข้อมูลออกมาจะสร้างรหัส CRC แล้วส่งตามท้ายไบต์ข้อมูลออกมา หลังจากนั้นเมื่อ Master ได้รับเฟรมข้อมูลและถอดข้อมูลออกจากเฟรมแล้วจะทำการคำนวณค่า CRC ตามสูตรเดียวกับ Slave เพื่อทำการเปรียบเทียบค่า CRC ทั้ง 2 ค่าว่าตรงกันหรือไม่ หากไม่ตรงกันแสดงว่าเกิดความผิดพลาดในการรับส่งข้อมูล ในโหมด RTU การรับส่งข้อมูล 1 ไบต์ ไม่ว่าจะเป็นข้อมูลส่วนใดภายในเฟรมจะต้องทำการส่งบิตข้อมูลรวม 11 บิต คือ บิตเริ่มต้น (Start) 1 บิต, บิตข้อมูล 8 บิต, บิตตรวจสอบ Parity ของข้อมูล 1 บิตและบิตหยุด 1 บิต (Stop) 1 บิต หรือหากเลือกแบบไม่มีบิต Parity ก็จะเป็นแบบ Stop แทน 2 บิต ดังแสดงในรูปที่ 2.17 สำหรับการกำหนดให้มีบิต Parity นั้น สามารถเลือกเป็นแบบคู่ (Even Parity) หรือคี่ (Odd Parity) ก็ได้ และหากต้องการออกแบบให้สอดคล้องกับอุปกรณ์ที่มีใช้กันทั่วไปมากที่สุด ควรเลือกแบบคู่โดยที่สามารถปรับเปลี่ยนเป็นแบบคี่หรือไม่มีการตรวจสอบ Parity (No Parity) ได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 ลักษณะเฟรมข้อมูลของ MODBUS RTU



รูปที่ 2.17 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS RTU

2.4.2.2 MODBUS ASCII

การรับส่งข้อมูลในโหมด ASCII นั้นมีความแตกต่างจากโหมด RTU ตรงที่ในโหมด RTU ข้อมูลที่จะส่งขนาด 1 ไบต์ นำมารวมกับบิตประกอบต่าง ๆ ก็สามารถส่งออกไปได้เลย แต่สำหรับโหมด ASCII จะมองข้อมูล 1 ไบต์ นั้นออกมาเป็นตัวอักษร 2 ตัว เช่น ค่า 0x5B ซึ่งเป็นเลขฐานสิบหก ก็จะถูกมองเป็นตัวอักษร '5' และตัวอักษร 'B' จากนั้นก็จะทำการค้นหารหัส ASCII ของตัวอักษรทั้ง 2 ตัวนั้น ซึ่งได้แก่ 0x35 สำหรับ '5' และ 0x42 สำหรับ 'B' แล้วทำการส่งรหัส ASCII ทั้ง 2 ค่านี้ออกไป ซึ่งจะได้ผลเท่ากับการส่งค่า0x5Bซึ่งเป็นข้อมูลขนาด1ไบต์ในโหมด RTU ดังแสดงในรูปที่ 2.18 และ 2.19 จะเห็นได้ว่าการส่งข้อมูลในโหมด ASCII จะต้องทำงานมากกว่าการส่งข้อมูลในโหมด RTU ซึ่งทำให้อัตราเร็วในการสื่อสารมีค่าต่ำกว่า สาเหตุที่เป็นแบบนี้ก็เพราะว่า โหมด ASCII ได้ถูกออกแบบมาสำหรับอุปกรณ์ที่ไม่มีความสามารถในการกำหนดช่วงระยะห่างของเวลาในการส่งเฟรมข้อมูล อย่างเช่นในโหมด RTU ที่อุปกรณ์สามารถกำหนดได้ว่าจะส่งเฟรมข้อมูลแต่ละเฟรมออกมาด้วยเวลาห่างกันเท่าใด และอุปกรณ์ที่รอรับข้อมูลก็ต้องสามารถตรวจจับและแยกแยะได้ว่าเฟรมข้อมูลแต่ละเฟรมที่รับเข้ามานั้น มีระยะเวลาห่างกันภายในช่วงเวลาที่กำหนดหรือไม่ เพื่อทำให้สามารถตรวจสอบหาจุดเริ่มต้นและจุด สิ้นสุดของเฟรมข้อมูลแต่ละเฟรมได้ แต่ในความเป็นจริงยังมีอุปกรณ์อีกหลายชนิดที่ไม่มีความสามารถพิเศษนี้ จึงต้องใช้วิธีอื่นที่จะช่วยให้สามารถรับรู้จุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลได้ นั่นได้แก่โหมด ASCII ซึ่งในโหมดนี้จะเริ่มต้นเฟรมข้อมูลด้วยการส่งรหัส ASCII ที่กำหนดให้หมายถึงจุดเริ่มต้น คือ 0x3A ซึ่งตรงกับตัวอักษร ':' ตามด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรสของ Slave, หมายเลขฟังก์ชัน, ข้อมูล, รหัสตรวจสอง RLC และรหัส ASCII 2 ตัว ที่กำหนดให้หมายถึงจุดสิ้นสุด คือ รหัส 0x0D และ 0x0A คือรหัส CR (Carriage Return) และ LF (Line Feed) ตามลำดับ โดยในขณะที่บัสข้อมูลว่างจากการรับส่งข้อมูล อุปกรณ์ทุกตัวจะคอยตรวจสอบข้อมูลในบัสว่ามีการส่งรหัส ASCII ของ ':' ออกมาหรือไม่ ถ้ามีก็จะรับรู้ว่าขณะนี้ได้มีการเริ่มต้นส่งเฟรมข้อมูลออกมาแล้ว ก็จะเข้ากระบวนการรับข้อมูลต่อไป

START	ADDRESS	FUNCTION	DATA	LRC CHECK	END
1 CHAR :	2 CHARS	2 CHARS	*n* CHARS	2 CHARS	2 CHARS CRLF

Modbus Frame Structure-ASCII Mode

รูปที่ 2.18 ลักษณะเฟรมข้อมูลของ MODBUS ASCII

รูปที่ 2.19 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS ASCII

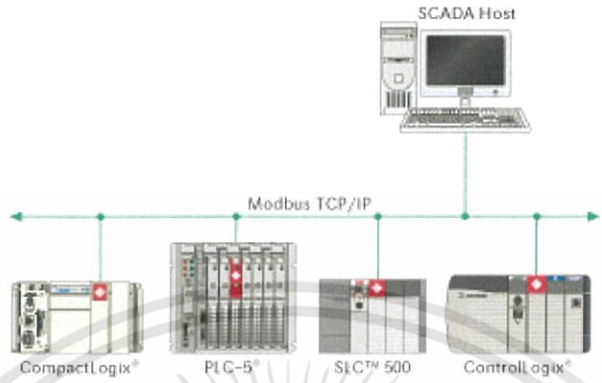
MODBUS จะบริการให้อุปกรณ์ติดต่อสื่อสารกันผ่าน Serial Port (RS-232/422/485) แต่ในปัจจุบันได้มีการพัฒนาให้อุปกรณ์สามารถติดต่อสื่อสารกับอุปกรณ์ที่อยู่บนเครือข่าย Ethernet ซึ่งอุปกรณ์ที่ใช้การสื่อสารแบบ MODBUS Protocol ส่วนใหญ่จะเป็น PLCs, DCSs, HMIs, Instruments อย่างไรก็ตาม MODBUS จำเป็นต้องมีอุปกรณ์จำพวก Gateway และ Bridge ในการติดต่อสื่อสารระหว่าง Serial Line กับ Ethernet

2.4.2.3 MODBUS TCP/IP

MODBUS TCP/IP ถูกพัฒนาขึ้นโดยมีวัตถุประสงค์เพื่อจะนำการสื่อสารแบบ Internet มาใช้กับอุปกรณ์จำพวก Ethernet Device ระยะในการใช้งานสำหรับการเดินสาย (สาย LAN) คือ 100 เมตร โดยสามารถขยายระยะในการสื่อสารได้โดยการใช้อุปกรณ์ Repeater หรือในระบบ LAN จะเรียกอุปกรณ์นี้ว่า Hub หรือ Switch ก็จะสามารถลากสายได้อีก 100 เมตร และยังสามารถต่อ Repeater

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขยายระยะทางได้โดยไม่จำกัด ในการสื่อสารโดยทั่วไปมีความเร็ว 100,000,000 บิตต่อวินาที (100 Mbps) และเชื่อมต่ออุปกรณ์ได้ไม่จำกัดจำนวน ตัวอย่างการสื่อสารของ MODBUS TCP/IP จะแสดงดังรูปที่ 2.20



รูปที่ 2.20 ลักษณะการสื่อสารของ MODBUS TCP/IP

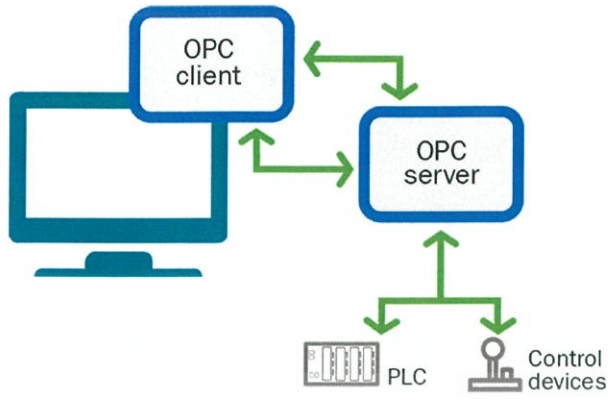
MODBUS ASCII/RTU ที่จะติดต่อสื่อสารกับ MODBUS TCP เพื่อให้ใช้งานในเครือข่าย Ethernet จะใช้ Gateway ติดต่อและแปลงรูปแบบการสื่อสารข้อมูล ดังแสดงในรูปที่ 2.21 โดยการสื่อสารของ MODBUS RTU/ASCII จะเป็นการสื่อสารผ่านทาง RS-232/422/485 นั้นจะถูก Gateway แปลงให้เป็น MODBUS TCP เพื่อใช้ในการติดต่อสื่อสารในเครือข่าย Ethernetต่อไป

รูปที่ 2.21 การแปลง MODBUS Serial เป็น MODBUS Ethernet

2.4.3 OPC [10]

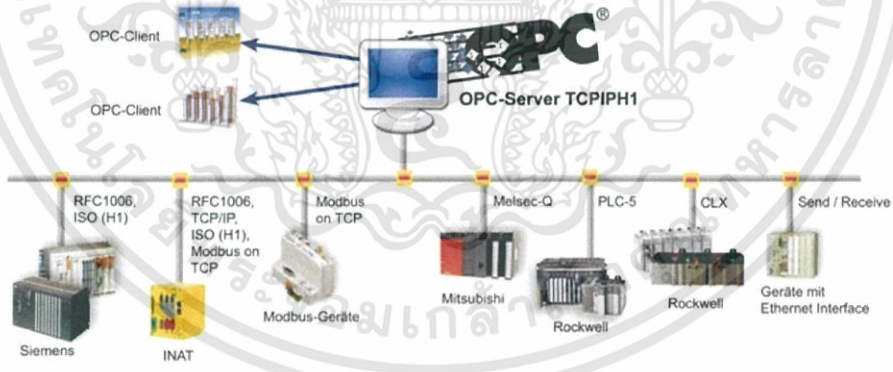
OPC (OLE For Process Control) เป็นตัวกลางในการติดต่อสื่อสารของ 2 ฝ่าย แบ่งเป็น Client และ Server หากผู้ใช้งานมี Controller (PLC, DCS) แต่ต้องการจะสื่อสารกับอุปกรณ์ควบคุมอื่น ๆ เช่น HMI, SCADA หรือ Remote Unit ต่าง ๆ ที่คนละยี่ห้อกันเพื่อให้สามารถสื่อสารกันได้จะต้องใช้ OPC เปรียบเทียบคือเป็นเสมือนตัวแปลภาษาของอุปกรณ์ ดังแสดงในรูปที่ 2.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.22 ลักษณะการสื่อสารของ OPC

ด้วยรูปแบบมาตรฐานของข้อมูลแบบเปิด ทำให้ผู้ผลิตอุปกรณ์ทั้งหลายบนโลกใบนี้สามารถพัฒนาระบบสื่อสารข้อมูลของตนให้เป็นไปตามมาตรฐานเดียวกันบ้างก็เป็น Server (ผู้ให้ข้อมูลซึ่งก็มักจะเป็นอุปกรณ์หรือเครื่องมือต่าง ๆ ในโรงงาน เช่น Sensor, Controller, PLC, หรือ HMI) กับ Client (ผู้ใช้ข้อมูลซึ่งมักจะเป็นระบบการบริหารจัดการทรัพยากรต่าง ๆ เช่น HMI , SCADA) ทั้งนี้การประยุกต์ใช้ส่วนใหญ่จะเป็นการช่วยให้มีการแลกเปลี่ยนข้อมูลจากอุปกรณ์ต่างค่ายกัน หรือการรวบรวมข้อมูลจากอุปกรณ์ต่าง ๆ ที่หลากหลายในเชิงของข้อมูลมีรูปแบบและมาตรฐานที่แตกต่างกันทำได้ง่ายขึ้น ดังแสดงในรูปที่ 2.23



รูปที่ 2.23 การเชื่อมต่อ OPC กับอุปกรณ์ต่าง ๆ

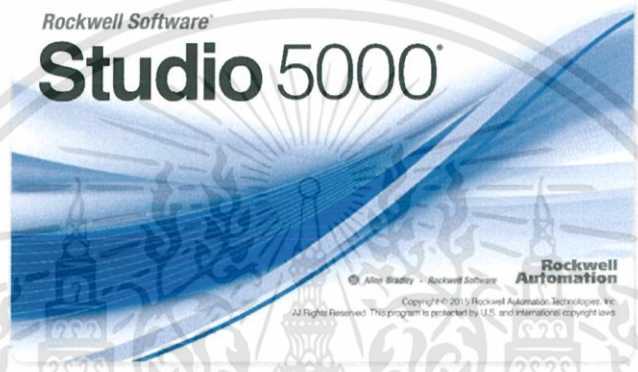
บางครั้งผู้ใช้งานอาจใช้วิธีการอื่น ๆ เช่น นำข้อมูลต่าง ๆ จากอุปกรณ์ไปรวบรวมไว้ในระบบฐาน ข้อมูล เช่น SQL Server จากนั้นให้ระบบบริหารจัดการทรัพยากรค่อยมานำข้อมูลจากฐานข้อมูลไปใช้อีกที ซึ่งก็มีข้อเด่นข้อด้อยแตกต่างกันไปทั้งนี้ก็ขึ้นอยู่กับความต้องการใช้ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 โปรแกรมที่ใช้ในการทำงาน

2.5.1 โปรแกรม Studio5000

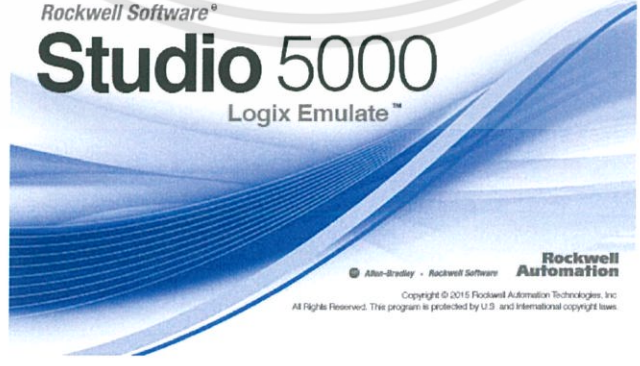
Studio5000 เป็นซอร์ฟแวร์สำหรับใช้ในการเขียนโปรแกรมควบคุมการทำงานของ Control Logix PLC, CompactLogix PLC หรือฮาร์ดแวร์อื่นของ Rockwell ที่รองรับการทำงานบน Studio5000 ซึ่งสนับสนุนอยู่ 4 ภาษาหลัก ๆ ได้แก่ Ladder, Function Block, Sequential Function Chart และ Structure Text นอกจากนี้โปรแกรมยังมีสามารถในการ Debug และ Online Edit เพื่อแก้ไขในขณะ PLC ทำงานอยู่ได้อีกด้วย หน้าต่างของโปรแกรม Studio5000 จะแสดงดังรูปที่ 2.24



รูปที่ 2.24 โปรแกรม Studio 5000

2.5.2 โปรแกรม Studio5000 Logix Emulate

Studio5000 Logix Emulate เป็นซอร์ฟแวร์จำลองการทำงานของ Logix5000 Contoller (ControlLogix, CompactLogix, FlexLogix,SoftLogix และ DriveLogix Controller) ทำให้สามารถทดสอบโปรแกรม Studio5000 ได้โดยไม่ต้องมี Controller และ I/O โมดูลจริง หน้าต่างของโปรแกรม Studio5000 Logix Emulate จะแสดงดังรูปที่ 2.25



รูปที่ 2.25 โปรแกรม Studio5000 Logix Emulate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.3 โปรแกรม FactoryTalk View Studio

FactoryTalk View Studio เป็นซอร์ฟแวร์หนึ่งสำหรับการออกแบบกราฟิกและใช้งาน HMI/SCADA สามารถเชื่อมต่อกับ Controller ได้หลากหลายยี่ห้อและหลายตัวในหน้าจอเดียวกัน เช่น Allen Bradley, Siemens เป็นต้น ผ่าน DDE/OPC server หรือ suit link และ FactoryTalk View Studio สามารถสั่งงานควบคุมกระบวนการได้จากกราฟิกที่เขียนได้โดยตรงเพียง กำหนด tag ให้ตรงกับ tag ที่ต้องการสั่งงานและควบคุมใน controller หน้าต่างของโปรแกรม FactoryTalk View Studio จะแสดงดังรูปที่ 2.26

รูปที่ 2.26 โปรแกรม FactoryTalk View Studio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

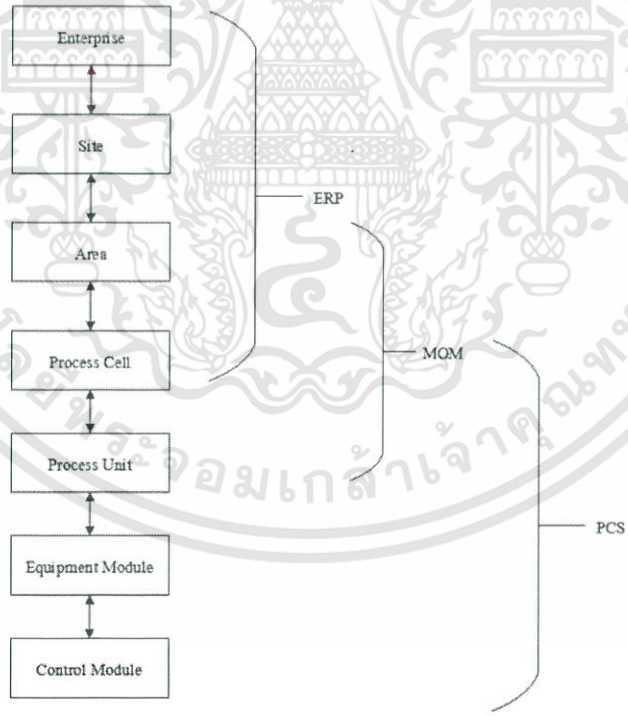
บทที่ 3
วิธีดำเนินการออกแบบศูนย์กลางรวบรวมข้อมูล

3.1 กล่าวนำ

ในการออกแบบศูนย์กลางรวบรวมข้อมูลที่สามารถใช้ได้จริงนั้นจะต้องมีการศึกษาโครงสร้างของระบบการบริหารจัดการข้อมูลของกระบวนการบรรจุภัณฑ์อาหารสัตว์ ว่ามีการรับส่งข้อมูลอย่างไร ส่งข้อมูลด้วยวิธีใดได้บ้างและข้อมูลที่ต้องการส่งมีอะไรบ้าง เพื่อที่จะสามารถออกแบบศูนย์กลางรวบรวมข้อมูลได้อย่างมีประสิทธิภาพและสอดคล้องกับฟังก์ชั่นการทำงานของระบบ MES โดยมีการออกแบบและทดสอบ ซึ่งสามารถอธิบายได้ตามขั้นตอนดังนี้

3.2 ศึกษาแบบจำลองของโรงงานและกระบวนการบรรจุภัณฑ์อาหารสัตว์

ในขั้นตอนแรกของการทำงาน จำเป็นต้องศึกษาแบบจำลองโรงงานและกระบวนการบรรจุภัณฑ์อาหารสัตว์ เพื่อให้สามารถออกแบบฟังก์ชั่นการทำงานของศูนย์กลางรวบรวมข้อมูลมีความสอดคล้องกับการทำงานของเครื่องจักรในระบบ PCS

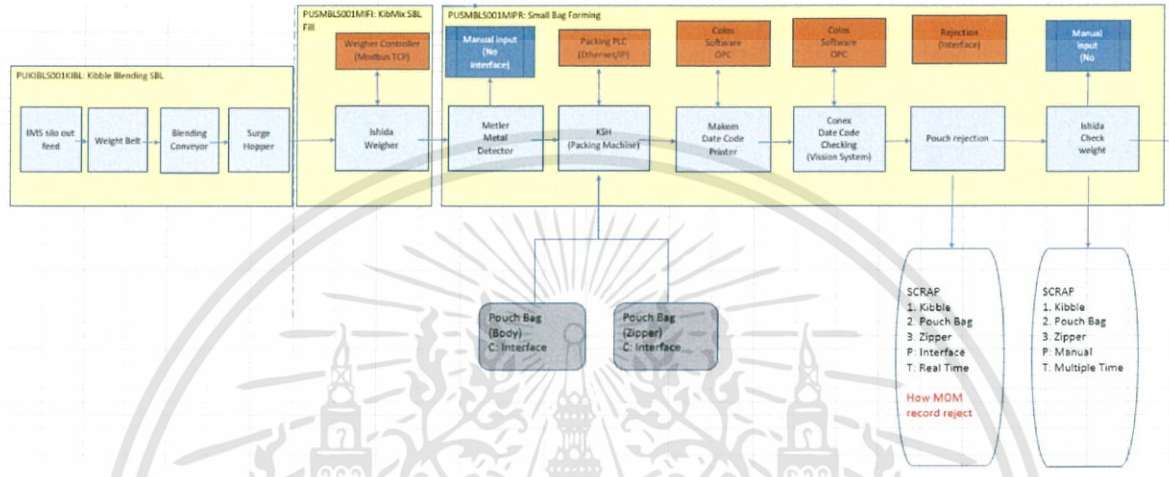


รูปที่ 3.1 แบบจำลองโรงงาน

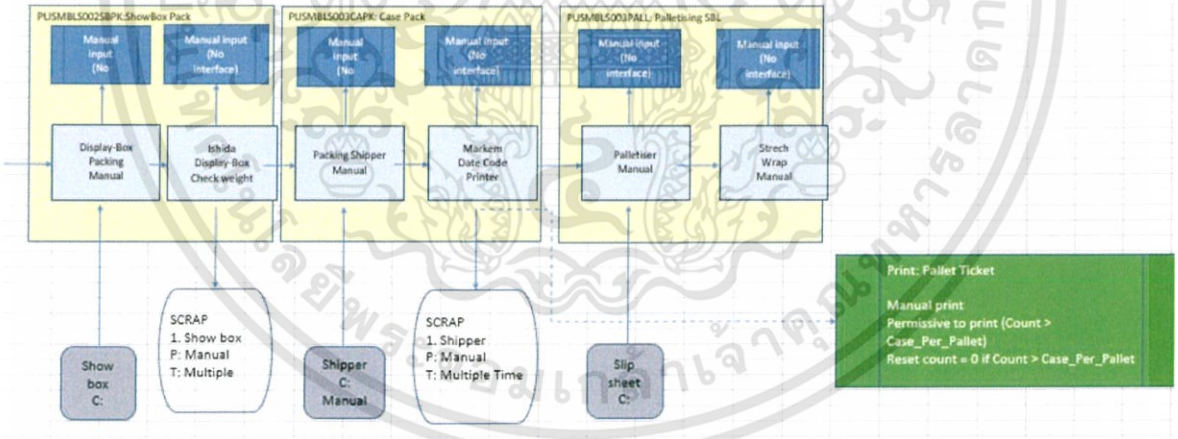
จากรูปที่ 3.1 จะแสดงให้เห็นภาพรวมของการทำงานในโรงงานว่าแต่ละส่วนทำงานสอดคล้องกันได้อย่างไร จากด้านบนสุดของแบบจำลองคือ Enterprise เป็นระบบการบริหารจัดการของโรงงาน ถัดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาคือ Site คอยรับคำสั่งจาก Enterprise ว่าสั่งให้ดำเนินการทำงานในส่วนไหน, Area คือ พื้นที่การผลิต เช่น พื้นที่ Sanook และพื้นที่ Jaroen เป็นพื้นที่การผลิตที่ได้รับมอบหมายให้สร้างศูนย์กลางรวบรวมข้อมูล , Process Cell คือ สายการผลิตและใน Process Cell จะประกอบด้วยหลายๆ Process Unit ซึ่งเป็นหน่วยการทำงานย่อยของสายการผลิตนั้น ๆ ซึ่งใน Process Unit จะประกอบด้วยโมดูลควบคุมการทำงานของเครื่องจักรและเครื่องจักรที่ใช้ในการผลิต



รูปที่ 3.2 กระบวนการบรรจุภัณฑ์อาหารสัตว์ (สายการผลิตที่11)



รูปที่ 3.3 กระบวนการบรรจุภัณฑ์อาหารสัตว์ (สายการผลิตที่11) (ต่อ)

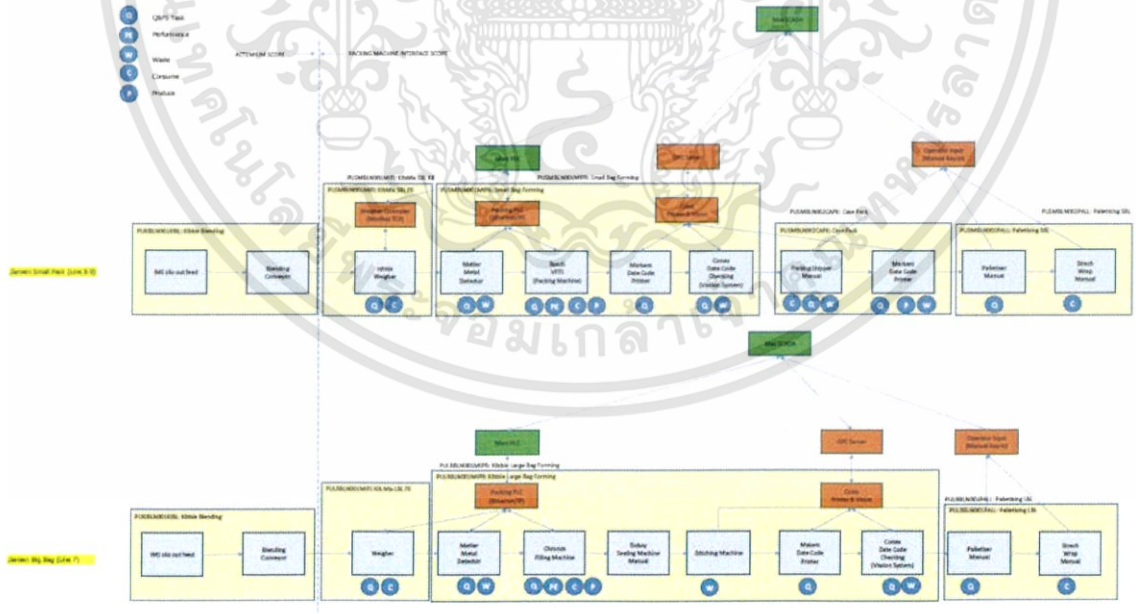
ในรูปที่ 3.2 และ 3.3 คือกระบวนการบรรจุภัณฑ์อาหารสัตว์ของสายการผลิตที่ 11 ซึ่งการทำงานของกระบวนการจะเริ่มจากนำผลิตภัณฑ์ (ลักษณะเป็นเม็ด) ที่ผลิตได้จากกระบวนการผลิต ลำเลียงผ่านสายพานมาชั่งน้ำหนักก่อนเข้าไปตรวจโลหะที่ปะปนมากับผลิตภัณฑ์ หลังจากนั้นผลิตภัณฑ์จะถูกส่งมาบรรจุใส่ถุงที่ได้ทำการขึ้นรูปและตรวจการพิมพ์วันที่ผลิต/หมดอายุไว้แล้ว จากนั้นจะทำการปิดล็อคถุงแล้วส่งถุงไปตรวจสอบน้ำหนักว่าตรงตามมาตรฐานที่กำหนดไว้หรือไม่ ถ้าน้ำหนักไม่ได้มาตรฐานจะถูกนำออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากกระบวนการ ส่วนถุงที่ผ่านการตรวจสอบจะถูกนำไปบรรจุลงในกล่องกระดาษขนาดเล็กแล้วทำการตรวจสอบน้ำหนักอีกครั้ง ก่อนนำไปไปบรรจุลงกล่องลังขนาดใหญ่อีกที หลังจากนั้นจะทำการพิมพ์วันที่ผลิต/หมดอายุลงบนกล่อง แล้วนำกล่องไปวางรวมกันบนลังไม้ (Pallet) เพื่อทำการพันพลาสติกใสรอบ ๆ กล่องสำหรับการเตรียมนำออกจำหน่าย

3.3 จัดทำเอกสารการจับคู่ข้อมูลและกำหนด Address ให้กับพารามิเตอร์ของกระบวนการ

ทำการกำหนด Address ให้กับ Tag ที่ทาง Manufacturing Operations Management (MOM) ซึ่งเป็นระบบ MES ของทางบริษัท Rockwell Automation ต้องใช้ในการรับส่งข้อมูล โดยทำการจัดทำเอกสารเพื่อขอ Address กับทางบริษัทผู้ผลิตเครื่องจักรใน PCS เพื่อให้มีความสอดคล้องกันของ Tag ที่ใช้ในการรับส่งข้อมูล โดยจะเริ่มจากการนำเอกสารที่ทาง MOM ส่งมาให้ว่าต้องการ Tag อะไรบ้าง ซึ่งก็คือพารามิเตอร์ของกระบวนการ ดังรูปที่ 3.5 มาเปรียบเทียบกับแผนภาพ Packing Machine Interface Scope ดังรูปที่ 3.4 ซึ่งมีรายละเอียดของ Process Unit, Process Cell, เครื่องจักรต่าง ๆ ที่ใช้ในการทำงาน รวมถึงบริษัทผู้ผลิตเครื่องจักร และการเชื่อมต่อการสื่อสารของข้อมูลในกระบวนการบรรจุภัณฑ์อาหารสัตว์กับ PLC และทำการศึกษาว่ามีข้อมูลอะไรบ้างของเครื่องจักรที่ต้องการส่งกลับไปให้ MOM ซึ่งจะมีพารามิเตอร์ที่ใช้ตรวจสอบคุณภาพของกระบวนการ (Quality Check Parameter), ข้อมูลของวัสดุที่ถูกใช้ในกระบวนการ (Consumption), วัสดุที่ผลิตได้จากกระบวนการบรรจุภัณฑ์ (Production) และวัสดุที่ไม่ได้ตามมาตรฐานที่กำหนดไว้จะถูกนำออกจากกระบวนการ (Waste)



รูปที่ 3.4 ตัวอย่างแผนภาพ PACKING MACHINE INTERFACE SCOPE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Quality Check Parameter –

Process step	Quality inspection point	Quality check parameter	Operation Tagged	Inspection frequency	Sort of data	Tag Path	Data type	Specification
	Metal detector	Metal detection and rejection test (3 scenario)		Every change over/2 hr & before start up and shut down. Every shift	PCS		Text	Pass/Fail
		Metal sample found	PUSMBLS001MIPR	Every detection	PCS		Numeric	
		Weight of metal rejection		Every change over/2 hr & before start up and shut down. Every shift	PCS		Numeric	
Packing Operation Small Bag	Date coding	Visual check on correct date code	PUSMBLS001MIPR	Every change over/2 hr	PCS		Capture photo	
	Vision system	Rejection test	PUSMBLS001MIPR	Every change over/2 hr	PCS		Text	Pass/Fail
		Number of rejection		Every 2 hrs	PCS		Numeric	
		Correct PO bag vs display box		Every change over/2 hr	PCS		Text	Pass/Fail
	Packing shipper	Visual check on correct date code	PUSMBLS003CAPK	Every change over/2 hr	PCS		Text	Pass/Fail
		Correct PO bag vs shipper		Every change over/2 hr	PCS		Text	Pass/Fail

รูปที่ 3.5 ตัวอย่างเอกสารของ MOM

ในเอกสารการจับคู่ข้อมูลจะประกอบไปด้วย Process Cell, Process Unit, Parameter, Protocol, Data type, Tag Path หรือ Address โดยเอกสารจะแยกตามบริษัทผู้ผลิตเครื่องจักรและสายการผลิต ดังรูปที่ 3.6 และ 3.7

Sanook Small Bag (Line 11)

1. Quality Check Parameter

Process Cell	Process Unit	Process Unit Description	Vendor	Parameter	Inspection frequency	Protocol	Usage type	Data type	Tag Path or Address
PCSMBLS001	PUSMBLS001MIPR	Small Bag Forming	Metler	Metal detection and rejection test (3 scenario)	Every change over/2 hr & before start up and shut down, Every shift	SCADA	Output	Text	Create new tag at scada
				Metal sample found	Every detection			Numeric	
				Weight of metal rejection	Every change over/2 hr & before start up and shut down, Every shift			Numeric	

2. SRC's / Process Parameters from MOM to PCS for Packing Line

Process Cell	Process Unit	Process Unit Description	Vendor	Parameter	Protocol	Usage type	Data type	Tag Path or Address
PCSMBLS001	PUSMBLS001MIPR	Small Bag Forming	KHS	Pack Format	Ethernet IP	Input (PCS to KHS)	String	Type[0].Type_Name
				Target Packing Rate (1-100)			Decimal	HMI_Rcp_VM_Run_Speed
			KHS	Timing Hopper (Open, Close)			Decimal	??? Each Timing channel has 4 parameter.
				KSH Program (1-100)			Long	HMI_Type_Active (read only)
				Roll Position (0-300mm)			Decimal	Type[0].MP_Mechanical_Setting :[XX] XX is the mech. Setting number on the screen.
				Forming Set (Pack Size)			Decimal	
				Funnel (S/M/L)			String	
				Side Seal Position			Decimal	
				Side Seal Heat			Decimal	
				End Seal Heat			Decimal	
				Seal Condition at Zipper area			Decimal	
				Peg hole (on/off)			Boolean	HMI_Rcp_Hanger_Hole_Punch
			KHS	Metal Detector Left Program	Ethernet IP	Input (PCS to KHS)	Long	???????
				Metal Detector Right Program			Long	???????
PCSMBLS003	PUSMBLS003PALL	Palletizing LBL		Number of layer	SCADA	Input	Long	Create new tag at scada
				Case per layer			Long	
				Case per pallet			Long	

รูปที่ 3.6 ตัวอย่างเอกสารการจับคู่ข้อมูลของบริษัท KHS (สายการผลิตที่11)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Consumption from Packing Line to PCS for MOM

Process Cell	Process Unit	Process Unit Description	Vendor	Parameter	Protocol	Usage type	Data type	Tag Path or Address
PCSMBLS001	PUSMBLS001MIPR	Small Bag Forming	KSH	MatID (Material Type)	Ethernet IP	Output (KSH to PCS)	STRING	
				MatDesc (Material Description)			STRING	
				Quantity (Material Quantity)			REAL	
				UoM (Unit of Measure)			STRING	
				Density (Material Density)			REAL	
				CarrierID (Material Carrier Identification)			STRING	
				BatchID (Material Batch Identification)			STRING	
				SublotID (Material Sublot Identification)			STRING	

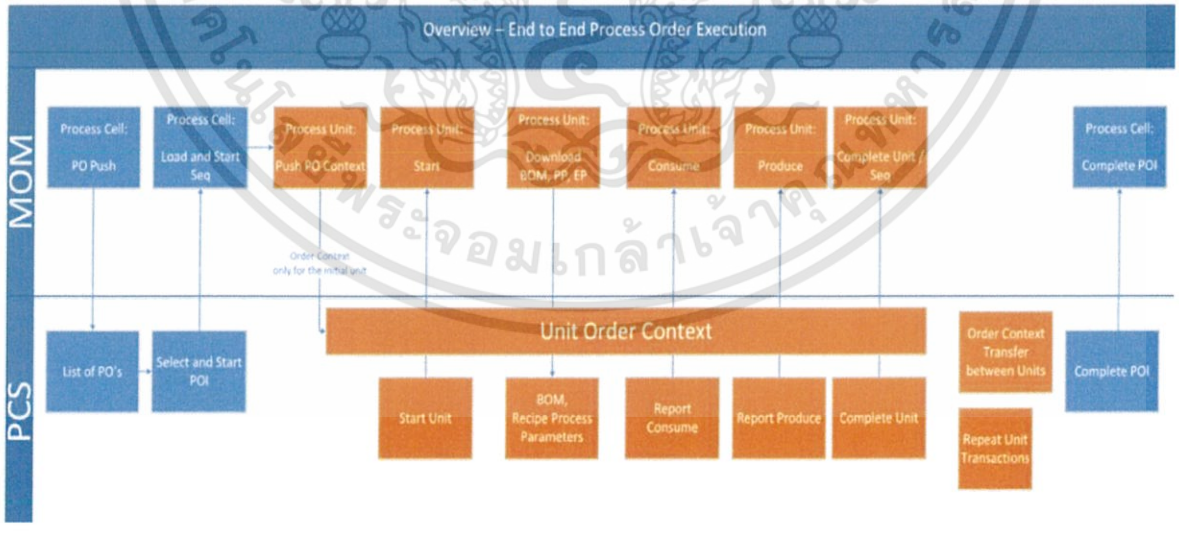
4. Production from Packing Line to PCS for MOM

Process Cell	Process Unit	Process Unit Description	Vendor	Parameter	Protocol	Usage type	Data type	Tag Path or Address
PCSMBLS001	PUSMBLS001MIPR	Small Bag Forming	KSH	MatID (Material Type)	Ethernet IP	Output (KSH to PCS)	STRING	
				MatDesc (Material Description)			STRING	
				Quantity (Material Quantity)			REAL	
				UoM (Unit of Measure)			STRING	
				Density (Material Density)			REAL	
				CarrierID (Material Carrier Identification)			STRING	
				BatchID (Material Batch Identification)			STRING	
				SublotID (Material Sublot Identification)			STRING	

รูปที่ 3.7 ตัวอย่างเอกสารการจับคู่ข้อมูลของบริษัท KHS (สายการผลิตที่11) (ต่อ)

3.4 การออกแบบฟังก์ชั่นการทำงาน (Function design)

การดำเนินการสร้างศูนย์กลางรวบรวมข้อมูล สิ่งแรกที่ต้องปฏิบัติ คือ การออกแบบลำดับขั้นตอนการทำงานของโปรแกรม PLC ให้มีการทำงานอย่างเป็นลำดับขั้นตอน โดยการออกแบบภาพรวมของระบบจะเขียนขึ้นตามแผนภาพดังรูปที่ 3.8 แสดงขั้นตอนการทำงานของศูนย์กลางรวบรวมข้อมูลซึ่งทำหน้าที่เป็นศูนย์กลางรับส่งข้อมูลระหว่าง MOM กับเครื่องจักรใน PCS



รูปที่ 3.8 ภาพรวมการทำงานของศูนย์กลางรวบรวมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.8 การทำงานของศูนย์กลางรวบรวมข้อมูลจะมีความเชื่อมโยงของสองส่วน ส่วนแรก คือ MOM และส่วนที่สองคือ PCS ซึ่งศูนย์กลางรวบรวมข้อมูลจะทำงานอยู่ในส่วนนี้ การทำงานในขั้นตอนแรกจะเริ่มต้นที่ระดับ Process Cell (สี่เหลี่ยมสีฟ้า)

1) MOM จะส่ง Process Order (PO) มาให้กับศูนย์กลางรวบรวมข้อมูล
2) ผู้ปฏิบัติงาน (Operator) จะทำการเลือก PO ที่ต้องใช้ในกระบวนการและสั่งให้เริ่มต้น Process Order Item (POI) โดยส่ง PO ที่ได้ทำการเลือกแล้วกลับไปให้ MOM
3) MOM จะทำการโหลด Sequence และเริ่มการทำงานของ Sequence

ขั้นตอนต่อไปจะอยู่ในระดับ Process Unit (สี่เหลี่ยมสีส้ม)

4) ใน Process Unit แรก MOM จะทำการโหลด PO Context ให้กับศูนย์กลางรวบรวมข้อมูล ในส่วนของ Process Unit ถัดไป ศูนย์กลางรวบรวมข้อมูลจะต้องทำการโหลด PO Context ให้กับ Process Unit
5) เมื่อได้รับ PO Context แล้วศูนย์กลางรวบรวมข้อมูลจะทำการสั่งให้เริ่มต้น Unit โดยคำสั่งจะถูกส่งไปให้ MOM ทำการเริ่มต้น Unit
6) MOM จะทำการดาวน์โหลด Bill of Material (BOM), Process Parameter (PP), Engineering Parameter (EP) ให้กับศูนย์กลางรวบรวมข้อมูล
7) เมื่อได้รับข้อมูลของวัสดุที่ถูกใช้ในกระบวนการจากเครื่องจักรใน PCS แล้วศูนย์กลางรวบรวมข้อมูลจะรายงาน Consumption ไปให้กับ MOM
8) เมื่อได้รับข้อมูลของวัสดุที่ผลิตได้จากกระบวนการบรรจุภัณฑ์แล้ว ศูนย์กลางรวบรวมข้อมูลจะรายงาน Production ไปให้กับ MOM
9) ศูนย์กลางรวบรวมข้อมูลจะทำการสั่งให้ Compete Unit โดยคำสั่งจะถูกส่งไปให้ MOM ทำการ Compete Unit
10) ถ้ามี Process Unit เดียว MOM จะทำการ Compete Sequence เลย หากมี Process Unit อื่น ๆ ใน Process Cell อีก ศูนย์กลางรวบรวมข้อมูลจะต้องทำการโหลด PO Context จาก MOM ให้กับ Process Unit และดำเนินการในขั้นตอนที่ 5) – 9) ให้สำเร็จ

ขั้นตอนต่อไปจะอยู่ในระดับ Process Cell (สี่เหลี่ยมสีฟ้า)

11) เมื่อดำเนินการใน Process Cell ครบทุก Process Unit แล้วศูนย์กลางรวบรวมข้อมูลจะทำการสั่งให้ Compete POI โดยคำสั่งจะถูกส่งไปให้ MOM ทำการ Compete POI

ในหัวข้อถัดไป จะเป็นการอธิบายฟังก์ชั่นการทำงานของโปรแกรมศูนย์กลางรวบรวมข้อมูลอย่างละเอียด โดยจะแบ่งหัวข้อหลักตามการทำงานใน Process Cell และ Process Unit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1 Process Cell Level

3.4.1.1 Order Preview

เป็นฟังก์ชั่นสำหรับการส่ง PO มาให้กับ PLC ใน PCS ซึ่งเป็นขั้นตอนแรกของระบบ โดยการทำงานจะเริ่มจาก Operator สั่งเริ่มต้นการทำงานโดยส่ง Ready flag ไปให้กับ MOM เมื่อ Ready flag มีค่าเท่ากับ 1 ศูนย์กลางรวบรวมข้อมูลจะอนุญาตให้ MOM ทำการดาวน์โหลด 5 PO มาเก็บไว้ใน array (MES_HMIUnit.OrderPreview.Data.Item) ที่ศูนย์กลางรวบรวมข้อมูลดังรูปที่ 3.9

MES_HMIUnit.OrderPreview	{...}
− MES_HMIUnit.OrderPreview.Data	{...}
− MES_HMIUnit.OrderPreview.Data.Item	{...}
− MES_HMIUnit.OrderPreview.Data.Item[0]	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[0].PO	'2222250'
+ MES_HMIUnit.OrderPreview.Data.Item[0].POItem	'2222250_MRSTH_SizeRed...
+ MES_HMIUnit.OrderPreview.Data.Item[0].Sequence	0
− MES_HMIUnit.OrderPreview.Data.Item[0].Material	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[0].Material.MatID	'1089295'
+ MES_HMIUnit.OrderPreview.Data.Item[0].Material.MatDesc	'1089295 Description'
MES_HMIUnit.OrderPreview.Data.Item[0].Material.Quantity	1100.0
+ MES_HMIUnit.OrderPreview.Data.Item[0].Material.UoM	'kg'
MES_HMIUnit.OrderPreview.Data.Item[0].Material.Density	0.0
+ MES_HMIUnit.OrderPreview.Data.Item[0].Material.CarrierID	''
+ MES_HMIUnit.OrderPreview.Data.Item[0].Material.BatchID	''
+ MES_HMIUnit.OrderPreview.Data.Item[0].Material.SublotID	''
+ MES_HMIUnit.OrderPreview.Data.Item[0].TargetRuns	5
+ MES_HMIUnit.OrderPreview.Data.Item[0].TimeStart	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[0].TimeComplete	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[0].Waiver	'NA'
+ MES_HMIUnit.OrderPreview.Data.Item[0].RunNumber	0
+ MES_HMIUnit.OrderPreview.Data.Item[1]	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[2]	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[3]	{...}
+ MES_HMIUnit.OrderPreview.Data.Item[4]	{...}

รูปที่ 3.9 Array ในศูนย์กลางรวบรวมข้อมูลที่ใช้เก็บ 5 PO

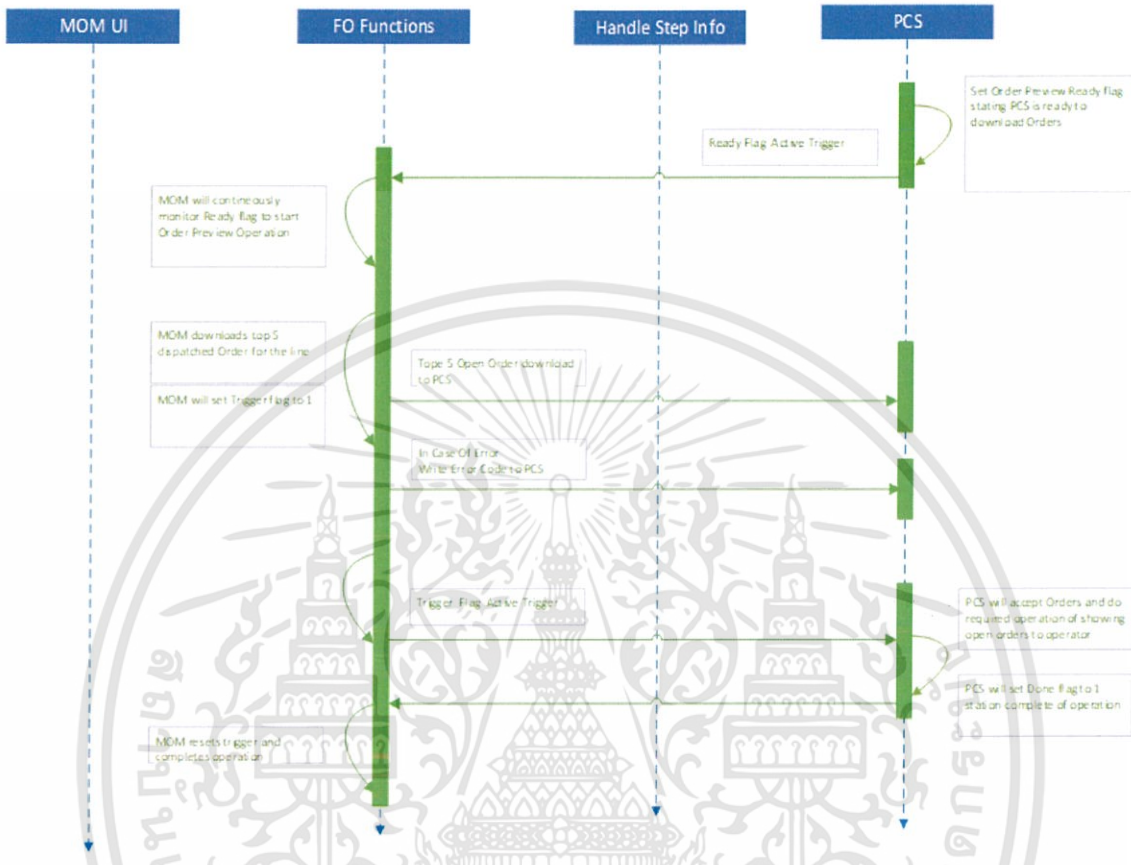
หลังจากทำการดาวน์โหลดเสร็จ MOM จะทำการเซ็ท Trigger flag ให้มีค่าเป็น 1 ดังรูปที่ 3.10 ศูนย์กลางรวบรวมข้อมูลจะทำการ acknowledges โดยการเซ็ท Done flag ให้มีค่าเป็น 1 หลังจาก MOM ได้รับ Done flag แล้ว จะทำการเคลียร์ Trigger flag ในศูนย์กลางรวบรวมข้อมูล

MES_HMIUnit.OrderPreview.Handshake	{...}
MES_HMIUnit.OrderPreview.Handshake.Ready	1
MES_HMIUnit.OrderPreview.Handshake.Trigger	1
MES_HMIUnit.OrderPreview.Handshake.Done	0
+ MES_HMIUnit.OrderPreview.Handshake.ErrorCode	0
+ MES_HMIUnit.OrderPreview.Handshake.Mode	0
+ MES_HMIUnit.OrderPreview.Handshake.Response	0
+ MES_HMIUnit.OrderPreview.Handshake.Config	{...}
+ MES_HMIUnit.OrderPreview.Handshake.Counter	{...}

รูปที่ 3.10 Array ของฟังก์ชั่น Handshake ในขั้นตอน Order Preview

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.11



รูปที่ 3.11 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Order Preview

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Ready flag เป็น 1 เพื่อเป็นการยืนยันว่าศูนย์กลางรวบรวมข้อมูลพร้อมสำหรับการดาวน์โหลด PO
2) เมื่อได้รับ Trigger บิตจาก MOM แล้วศูนย์กลางรวบรวมข้อมูลจะทำการอัพเดท 5 PO ที่จะถูกดำเนินการในสายการผลิตและทำการทริกเกอร์บิต Done

ขอบเขตการทำงานของ MOM

1) MOM จะทำการตรวจสอบ ถ้า Ready flag มีค่าเท่ากับ 1 เมื่อเงื่อนไขนี้เป็นจริง MOM จะทำการดาวน์โหลด 5 PO ลงในศูนย์กลางรวบรวมข้อมูล
2) MOM จะทำการเซ็ท Trigger เป็น 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) MOM จะรอจนกระทั่งศูนย์กลางรวบรวมข้อมูลส่ง Done flag มาให้ เพื่อจบการทำงานของขั้นตอน Order Preview

-ถ้าได้รับ Done flag เท่ากับ 1 MOM จะทำการรีเซ็ท Trigger flag

-ถ้าไม่ได้รับ Done flag จนกระทั่งหมดเวลาแล้ว MOM จะทำการส่ง Error message และ ทำการรีเซ็ท trigger flag

3.4.1.2 Load Sequence

เป็นฟังก์ชั่นสำหรับการดำเนินการโหลด Sequence (สร้าง Sequence และเริ่มต้น Sequence) ซึ่งเป็นขั้นตอนต่อจาก Order Preview โดย Operator จะทำการเลือกคำสั่งที่ต้องการใช้งานจาก 5 PO ที่ถูกจัดเก็บอยู่ใน array (MES_HMIUnit.OrderPreview.Data.Item) แล้วศูนย์กลางรวบรวมข้อมูลจะทำการคัดลอกข้อมูลมาไว้ใน array (MES_HMIUnit.Context) ดังรูปที่ 3.12

MES_HMIUnit	{...}
− MES_HMIUnit.Context	{...}
+ MES_HMIUnit.Context.PO	'ORDER001'
+ MES_HMIUnit.Context.POItem	'ORDER001_ORDERITEM_01'
+ MES_HMIUnit.Context.Sequence	1
− MES_HMIUnit.Context.Material	{...}
+ MES_HMIUnit.Context.Material.MatID	'1089295'
+ MES_HMIUnit.Context.Material.MatDesc	''
MES_HMIUnit.Context.Material.Quantity	1100.0
+ MES_HMIUnit.Context.Material.UoM	'kg'
MES_HMIUnit.Context.Material.Density	0.0
+ MES_HMIUnit.Context.Material.CarrierID	''
+ MES_HMIUnit.Context.Material.BatchID	''
+ MES_HMIUnit.Context.Material.SublotID	''
+ MES_HMIUnit.Context.TargetRuns	0
+ MES_HMIUnit.Context.TimeStart	{...}
+ MES_HMIUnit.Context.TimeComplete	{...}
+ MES_HMIUnit.Context.Waiver	''
+ MES_HMIUnit.Context.RunNumber	0

รูปที่ 3.12 Array ของ MES_HMIUnit.Context ในขั้นตอน Load Sequence

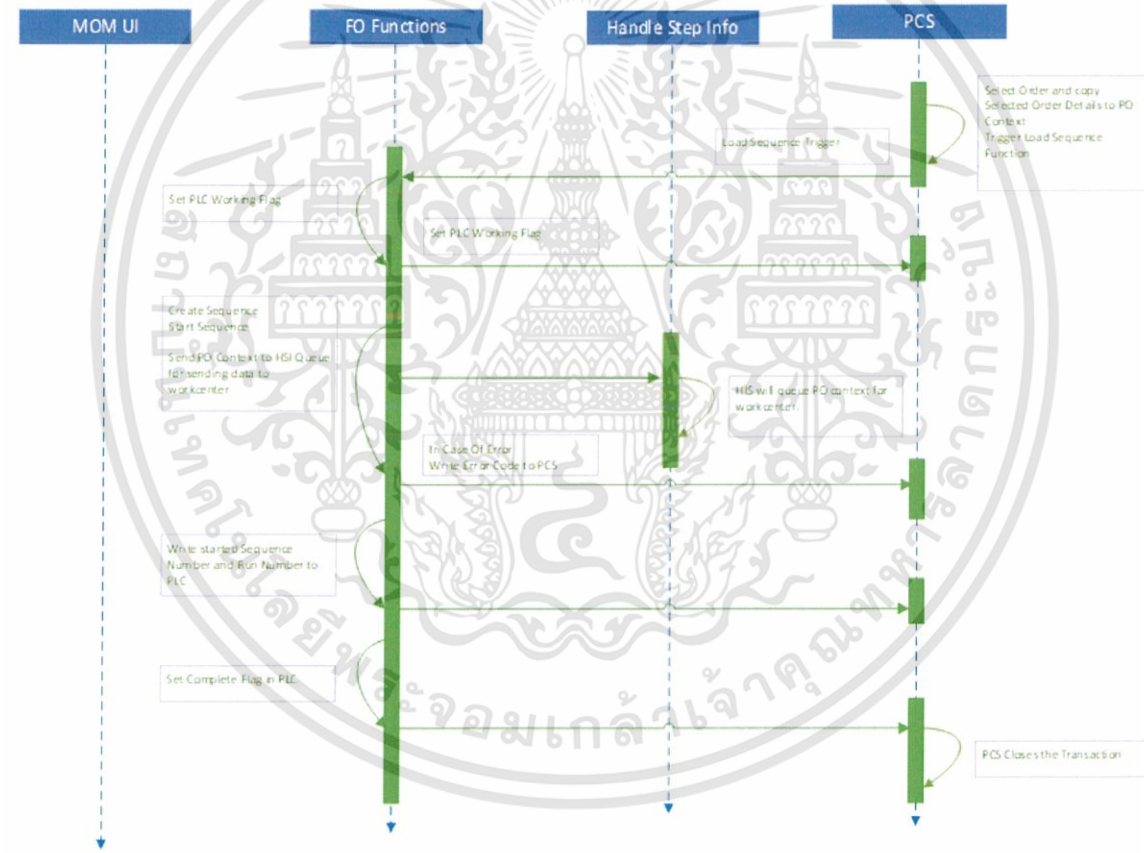
หลังจากทำการคัดลอกข้อมูลเสร็จแล้ว ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Load Sequence trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Load Sequence trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการสร้าง Sequence และเริ่มต้น Sequence หลังจากนั้น MOM จะทำการเขียน Sequence Number และ run Number ให้กับศูนย์กลางรวบรวมข้อมูลแล้วจึงทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Load Sequence ดังรูปที่ 3.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MES_HMIUnit.LoadSequence	{...}
MES_HMIUnit.LoadSequence.Trigger	0
MES_HMIUnit.LoadSequence.Working	0
MES_HMIUnit.LoadSequence.Completed	0
⊞ MES_HMIUnit.LoadSequence.Error	0
⊞ MES_HMIUnit.LoadSequence.Mode	0
⊞ MES_HMIUnit.LoadSequence.Response	0
⊞ MES_HMIUnit.LoadSequence.Config	{...}
⊞ MES_HMIUnit.LoadSequence.Counter	{...}

รูปที่ 3.13 Array ของฟังก์ชั่น Handshake ในขั้นตอน Load Sequence

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.14



รูปที่ 3.14 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Load Sequence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) Operator จะทำการอัพเดท PLC tag (เลือก PO context ที่ต้องใช้งานและข้อมูลของ Material Id)
2) ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Load Sequence trigger เป็น 1 เพื่อเริ่มการทำงานในขั้นตอน Load Sequence

ขอบเขตการทำงานของ MOM

1) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูลและจะทำการตรวจสอบว่ามีการเริ่มต้นของ Sequence ใด ๆ หรือไม่ ถ้าไม่มีการเริ่มต้นของ Sequence MOM จะสร้าง Sequence ใหม่สำหรับ PO context
2) MOM จะเริ่มต้นการสร้าง Sequence ถัดไปสำหรับ PO context
3) PO context สำหรับ Sequence จะถูกดาวน์โหลดไปยัง PO เป้าหมายตามขั้นตอนของสูตรควบคุม Sequence ที่กำหนดค่าไว้ สำหรับการดาวน์โหลด PO context ในครั้งแรก
4) MOM จะอัพเดท Sequence Number ใน PO context ของ Process Unit

3.4.1.3 Complete Sequence

เป็นฟังก์ชั่นสำหรับการเปลี่ยนสถานะ Sequence เป็น Completed ซึ่งเป็นขั้นตอนต่อจาก Complete Unit PO โดย Operator จะทำการเลือกคำสั่งที่ต้องการใช้งานจากคำสั่งทั้งหมดที่ถูกจัดเก็บอยู่ใน array (MES_HMIUnit.OrderPreview.Data.Item) แล้วศูนย์กลางรวบรวมข้อมูลจะทำการคัดลอกข้อมูลมาไว้ใน array (MES_HMIUnit.Context) ดังรูปที่ 3.15

MES_HMIUnit	{...}
− MES_HMIUnit.Context	{...}
+ MES_HMIUnit.Context.PO	'ORDER001'
+ MES_HMIUnit.Context.POItem	'ORDER001_ORDERITEM_01'
+ MES_HMIUnit.Context.Sequence	1
− MES_HMIUnit.Context.Material	{...}
+ MES_HMIUnit.Context.Material.MatID	'1089295'
+ MES_HMIUnit.Context.Material.MatDesc	''
MES_HMIUnit.Context.Material.Quantity	1100.0
+ MES_HMIUnit.Context.Material.UoM	'kg'
MES_HMIUnit.Context.Material.Density	0.0
+ MES_HMIUnit.Context.Material.CarrierID	''
+ MES_HMIUnit.Context.Material.BatchID	''
+ MES_HMIUnit.Context.Material.SublotID	''
+ MES_HMIUnit.Context.TargetRuns	0
+ MES_HMIUnit.Context.TimeStart	{...}
+ MES_HMIUnit.Context.TimeComplete	{...}
+ MES_HMIUnit.Context.Waiver	''
+ MES_HMIUnit.Context.RunNumber	0

รูปที่ 3.15 Array ของ MES_HMIUnit.Context ในขั้นตอน Complete Sequence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

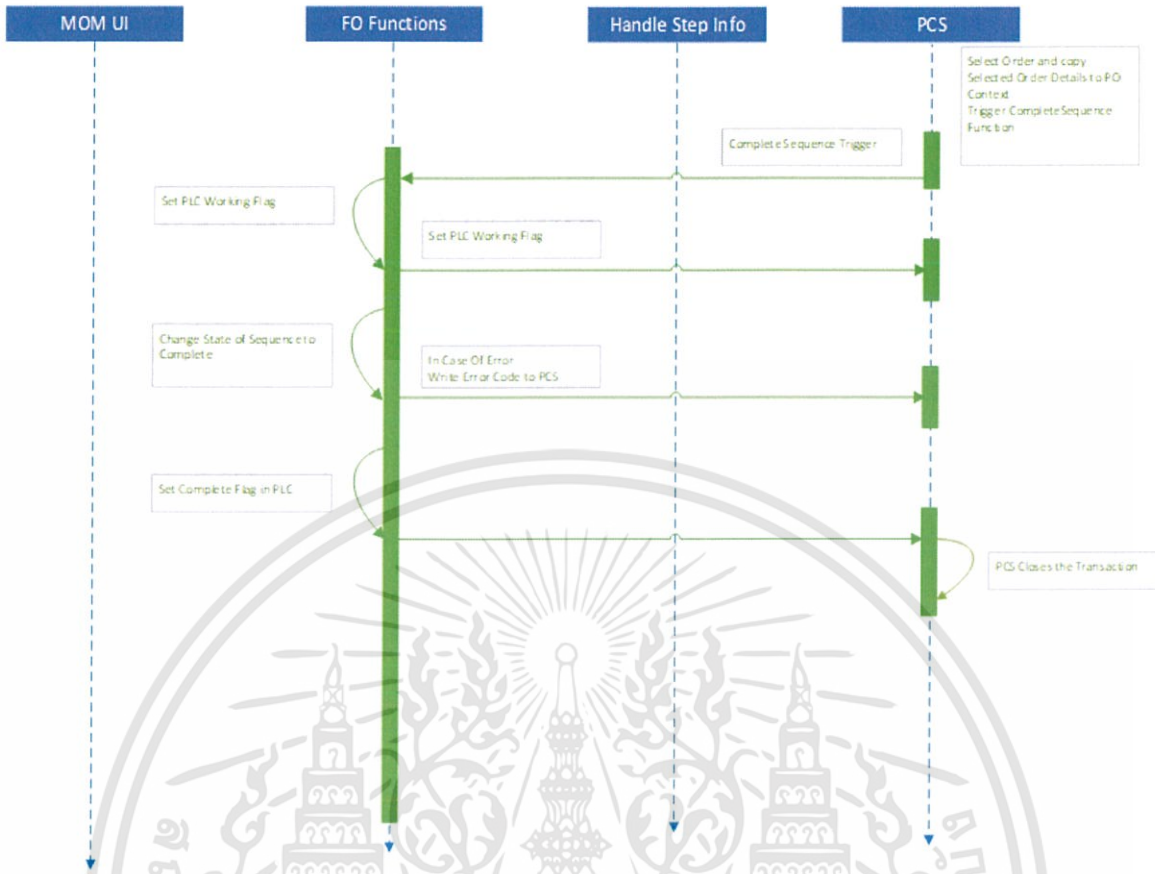
หลังจากทำการคัดลอกข้อมูลเสร็จแล้ว ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Complete Sequence trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Complete Sequence trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการเปลี่ยนสถานะ Sequence เป็น Complete หลังจากนั้น MOM จะทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Complete Sequence เมื่อขั้นตอน Complete Sequence เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเคลียร์ Handshake Tag ทั้งหมดให้มีค่าเป็น 0 ดังรูปที่ 3.16

MES_HMIUnit.CompleteSequence	{...}
MES_HMIUnit.CompleteSequence.Trigger	0
MES_HMIUnit.CompleteSequence.Working	0
MES_HMIUnit.CompleteSequence.Completed	0
⊞ MES_HMIUnit.CompleteSequence.Error	0
⊞ MES_HMIUnit.CompleteSequence.Mode	0
⊞ MES_HMIUnit.CompleteSequence.Response	0
⊞ MES_HMIUnit.CompleteSequence.Config	{...}
⊞ MES_HMIUnit.CompleteSequence.Counter	{...}

รูปที่ 3.16 Array ของฟังก์ชั่น Handshake ในขั้นตอน Complete Sequence

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Complete Sequence

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) Operator จะทำการอัพเดท PLC tag (เลือก Process Order context ที่ต้องใช้งานสำหรับ Sequence ที่ต้องการ Completed)
2) ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Complete Sequence trigger เป็น 1 เพื่อเริ่มการทำงานในขั้นตอน Complete Sequence

ขอบเขตการทำงานของ MOM

1) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูล
2) MOM จะทำการตรวจสอบว่าทุกขั้นตอนของกระบวนการใน Sequence พร้อมสำหรับการ Complete หรือไม่ ถ้าเป็นไปตามเงื่อนไข MOM จะทำการ Complete ทุกขั้นตอน
3) MOM จะทำการเปลี่ยนสถานะ Sequence เป็น Complete
4) MOM จะทำการเซ็ท Complete flag ให้กับศูนย์กลางรวบรวมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.4 Complete POI

เป็นฟังก์ชั่นสำหรับการ Complete POI ซึ่งเป็นขั้นตอนต่อจาก Complete Sequence โดย Operator จะทำการเลือกคำสั่งที่ต้องการใช้งานจากคำสั่งทั้งหมดที่ถูกจัดเก็บอยู่ใน array (MES_HMIUnit.OrderPreview.Data.Item) แล้วศูนย์กลางรวบรวมข้อมูลจะทำการคัดลอกข้อมูลมาไว้ใน array (MES_ HMIUnit.CompletePOI.Data) ดังรูปที่ 3.18

MES_HMIUnit.CompletePOI	{...}
− MES_HMIUnit.CompletePOI.Data	{...}
+ MES_HMIUnit.CompletePOI.Data.PO	'2222247'
+ MES_HMIUnit.CompletePOI.Data.POItem	'2222247_MRSTH_SizeRed...
+ MES_HMIUnit.CompletePOI.Data.Sequence	0
+ MES_HMIUnit.CompletePOI.Data.Material	{...}
+ MES_HMIUnit.CompletePOI.Data.TargetRuns	0
+ MES_HMIUnit.CompletePOI.Data.TimeStart	{...}
+ MES_HMIUnit.CompletePOI.Data.TimeComplete	{...}
+ MES_HMIUnit.CompletePOI.Data.Waiver	''
+ MES_HMIUnit.CompletePOI.Data.RunNumber	0
− MES_HMIUnit.CompletePOI.Handshake	{...}

รูปที่ 3.18 Array ของ MES_HMIUnit.CompletePOI ในขั้นตอน Complete POI

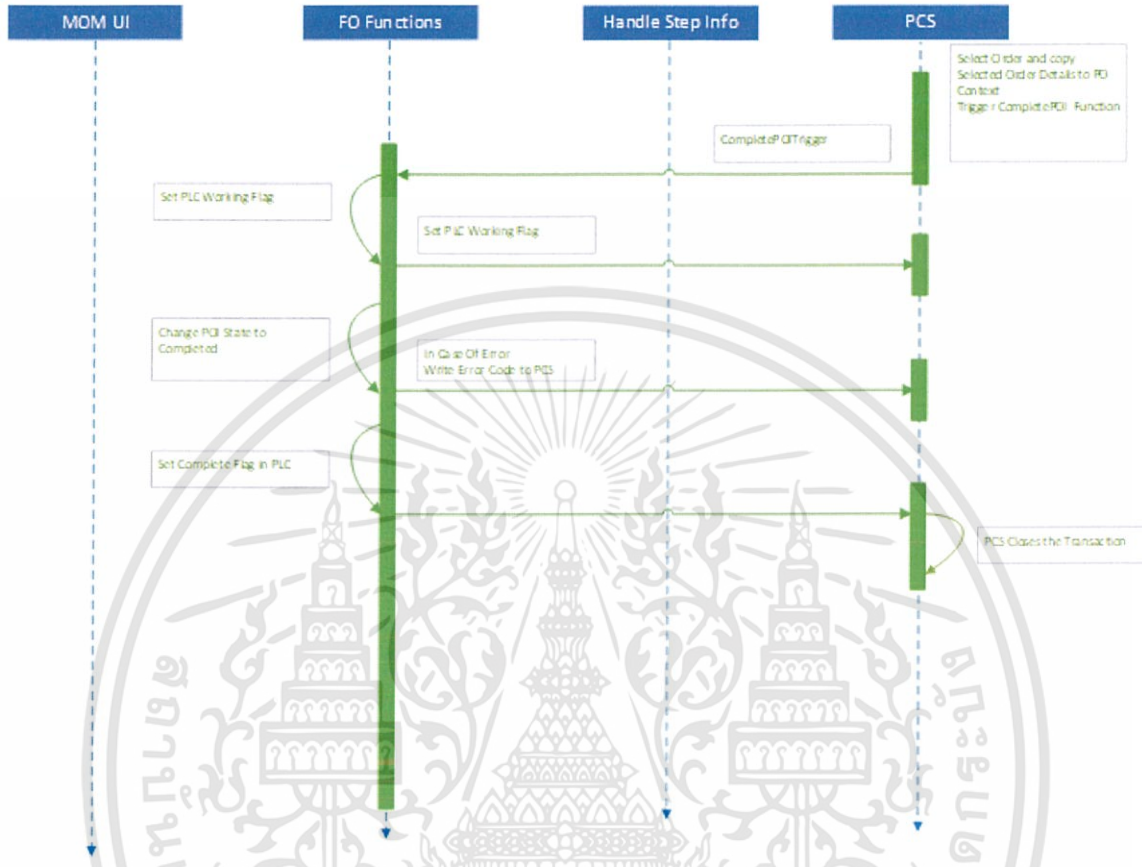
หลังจากทำการคัดลอกข้อมูลเสร็จแล้ว ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Complete POI trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Complete POI trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการเปลี่ยนสถานะของ POI เป็น Completed หลังจากนั้น MOM จะทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Complete POI เมื่อขั้นตอน Complete POI เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเคลียร์ Handshake Tag ทั้งหมดให้มีค่าเป็น 0 ดังรูปที่ 3.19

MES_HMIUnit.CompletePOI.Handshake	{...}
MES_HMIUnit.CompletePOI.Handshake.Trigger	0
MES_HMIUnit.CompletePOI.Handshake.Working	0
MES_HMIUnit.CompletePOI.Handshake.Completed	0
+ MES_HMIUnit.CompletePOI.Handshake.Error	0
+ MES_HMIUnit.CompletePOI.Handshake.Mode	0
+ MES_HMIUnit.CompletePOI.Handshake.Response	0
+ MES_HMIUnit.CompletePOI.Handshake.Config	{...}
+ MES_HMIUnit.CompletePOI.Handshake.Counter	{...}
MES_HMIUnit.CompletePOI.HandshakeAOI	{...}

รูปที่ 3.19 Array ของฟังก์ชั่น Handshake ในขั้นตอน Complete POI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.20



รูปที่ 3.20 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Complete POI

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) Operator จะทำการอัพเดท PLC tag (เลือก Process Order context ที่ต้องใช้งานสำหรับ POI ที่ต้องการ Completed)
2) ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Complete POI trigger เป็น 1 เพื่อเริ่มการทำงานในขั้นตอน Complete POI

ขอบเขตการทำงานของ MOM

1) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูล
2) MOM จะทำการตรวจสอบว่าไม่มี Sequence สำหรับ POI อยู่ในสถานะ Running ถ้าเป็นไปตามเงื่อนไข MOM จะส่ง error message ไปยังศูนย์กลางรวบรวมข้อมูลและหยุดการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) MOM จะทำการเปลี่ยนสถานะของ POI เป็น Complete

4) MOM จะทำการเซ็ท Complete flag ให้กับศูนย์กลางรวบรวมข้อมูล

3.4.2 Process Unit Level

3.4.2.1 Start Unit PO

เป็นฟังก์ชั่นสำหรับการดำเนินการเริ่มต้นของ Process Unit ซึ่งเป็นขั้นตอนต่อจาก Load Sequence โดย MOM จะทำการคัดลอกคำสั่งที่ถูกจัดเก็บอยู่ใน array (MES_HMIUnit.Context) มาให้กับ Process Unit แรกของ Process Cell โดยข้อมูลที่ถูกคัดลอกมาจะถูกจัดเก็บอยู่ใน array (MES_Object.Context) ในส่วนของ Process Unit ที่เหลือ ศูนย์กลางรวบรวมข้อมูลจะทำการคัดลอกคำสั่งที่ถูกจัดเก็บอยู่ใน array (MES_HMIUnit.Context) มาไว้ใน array (MES_Object.Context) ดังรูปที่ 3.21

− MES_Object	{...}
− MES_Object.Context	{...}
+ MES_Object.Context.PO	'ORDER001'
+ MES_Object.Context.POItem	'ORDER001_ORDERITEM_01'
+ MES_Object.Context.Sequence	1
− MES_Object.Context.Material	{...}
+ MES_Object.Context.Material.MatID	'1089295'
+ MES_Object.Context.Material.MatDesc	'1089295 Description'
MES_Object.Context.Material.Quantity	1100.0
+ MES_Object.Context.Material.UoM	'kg'
MES_Object.Context.Material.Density	0.0
+ MES_Object.Context.Material.CarrierID	''
+ MES_Object.Context.Material.BatchID	''
+ MES_Object.Context.Material.SublotID	''
+ MES_Object.PO_Status	0

รูปที่ 3.21 Array ของ MES_Object.Context ในขั้นตอน Start Unit PO

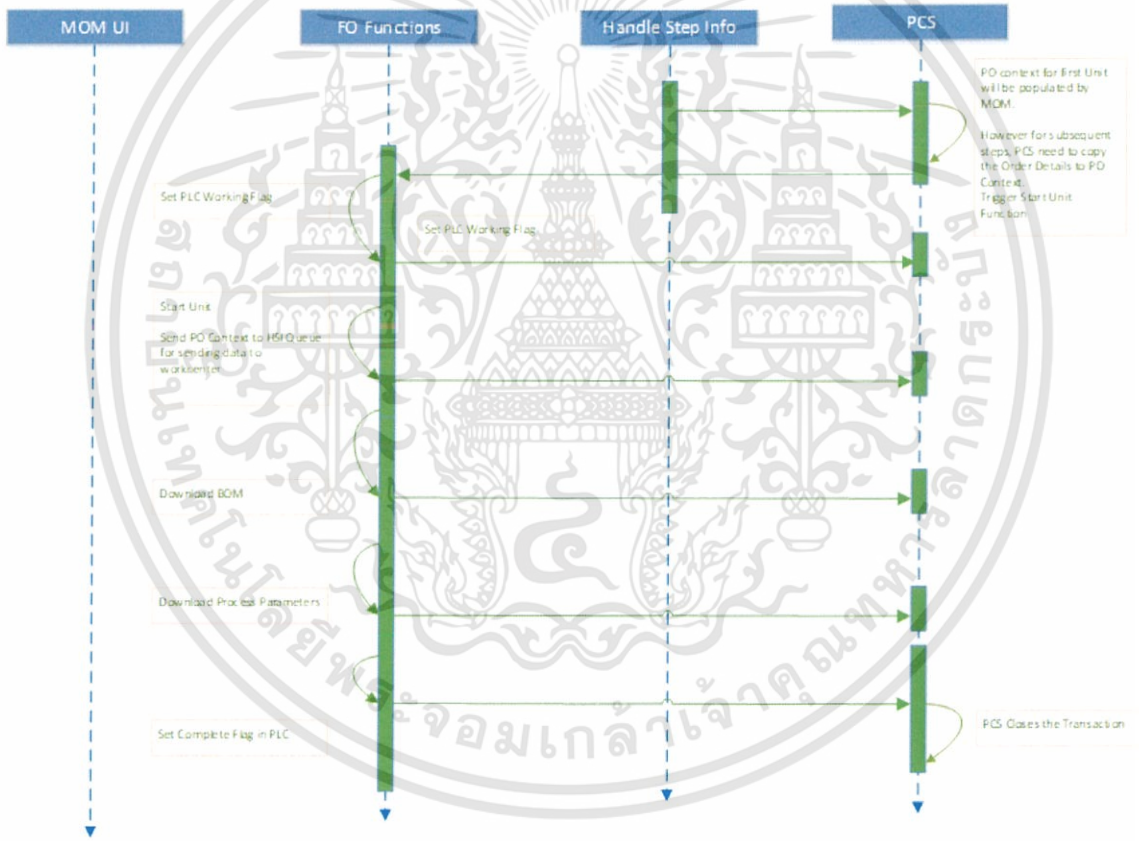
หลังจากทำการคัดลอกข้อมูลเสร็จแล้ว ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Object Start trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Object Start trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการเริ่มต้น Unit โดยการดาวน์โหลด Bill of Material (BOM), Process Parameters (PP) มาเก็บไว้ใน array ที่ศูนย์กลางรวบรวมข้อมูล หลังจากนั้น MOM จะทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Start Unit PO เมื่อขั้นตอน Start Unit PO เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเคลียร์ Handshake Tag ทั้งหมดให้มีค่าเป็น 0 ดังรูปที่ 2.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

⊟ MES_Object.Start	{...}
MES_Object.Start.Trigger	0
MES_Object.Start.Working	0
MES_Object.Start.Completed	0
⊞ MES_Object.Start.Error	0
⊞ MES_Object.Start.Mode	0
⊞ MES_Object.Start.Response	0
⊞ MES_Object.Start.Config	{...}
⊞ MES_Object.Start.Counter	{...}

รูปที่ 3.22 Array ของฟังก์ชั่น Handshake ในขั้นตอน Start Unit PO

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.23



รูปที่ 3.23 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Start Unit PO

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Object Start trigger เป็น 1 เพื่อเริ่มการทำงานในขั้นตอน Start Unit Process Order (PO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการทำงานของ MOM

1) MOM จะทำการอัพเดท PLC tag (คัดลอก PO context ที่ต้องใช้งาน) สำหรับ Process Unit แรก ในส่วนของ Process Unit ที่เหลือ ศูนย์กลางรวบรวมข้อมูลจะต้องทำการคัดลอก PO context
2) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูล
3) MOM จะทำการเริ่มต้น Process Unit สำหรับ PO context
4) Bill of Material (BOM) และ Process Parameters (PP) จะถูกดาวน์โหลดไปยัง PLC tag ของ Process Unit

3.4.2.2 Consume

เป็นฟังก์ชั่นสำหรับการดำเนินการรายงาน Consumption ซึ่งเป็นขั้นตอนต่อจาก Start Unit PO โดย Operator จะทำการกรอกรายละเอียดของวัสดุที่ถูกใช้ไปในกระบวนการบรรจุภัณฑ์อาหารสัตว์ให้กับศูนย์กลางรวบรวมข้อมูล และข้อมูลจะถูกศูนย์กลางรวบรวมข้อมูลนำมาจัดเก็บอยู่ใน array (MES_Object.CONS) ดังรูปที่ 3.24

MES_Object.CONS	{...}
MES_Object.CONS.Material	{...}
MES_Object.CONS.Material.MatID	'1021133'
MES_Object.CONS.Material.MatDesc	''
MES_Object.CONS.Material.Quantity	10.0
MES_Object.CONS.Material.UoM	'kg'
MES_Object.CONS.Material.Density	0.0
MES_Object.CONS.Material.CarrierID	''
MES_Object.CONS.Material.BatchID	''
MES_Object.CONS.Material.SublotID	'S1490271035834'

รูปที่ 3.24 Array ของ MES_Object.CONS ในขั้นตอน Consume

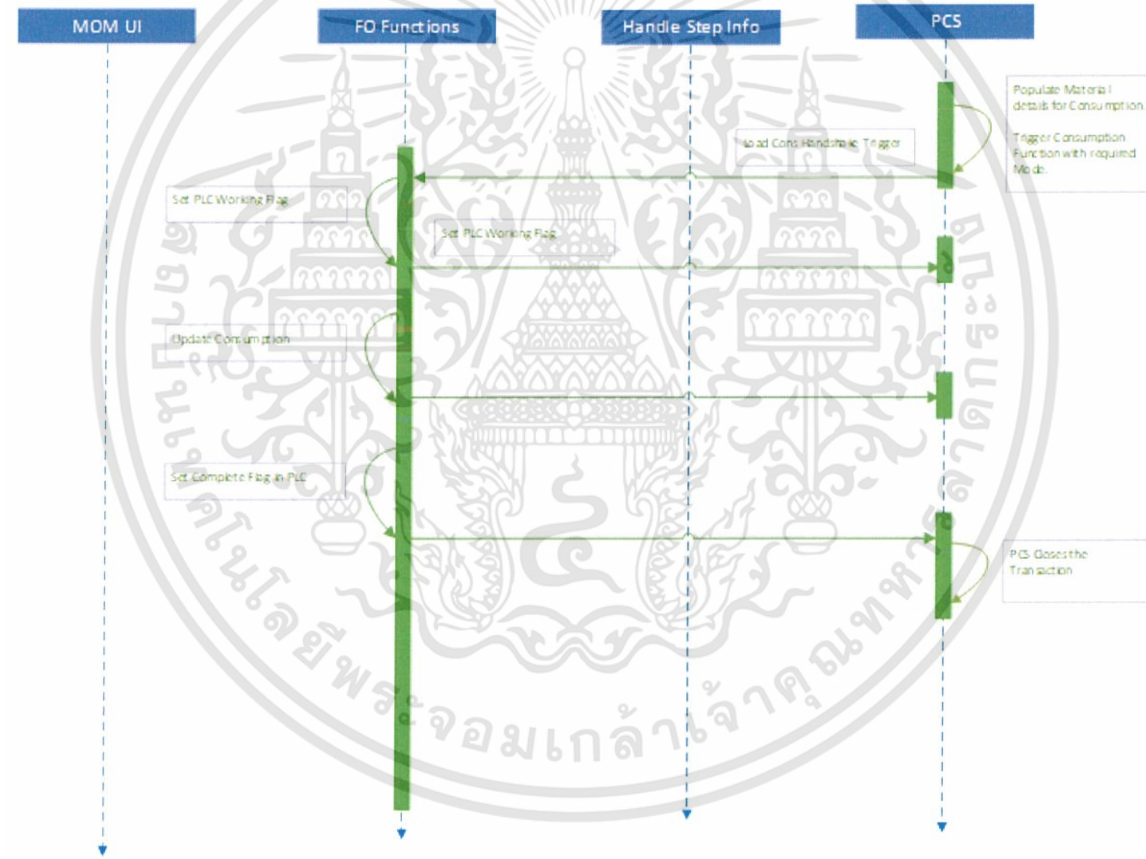
หลังจากข้อมูลถูกจัดเก็บอยู่ใน array (MES_Object.CONS) แล้ว ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Object Cons trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Object CONS trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการอัพเดท Consumption หลังจากนั้น MOM จะทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Consume เมื่อขั้นตอน Consume เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเคลียร์ Handshake Tag ทั้งหมดให้มีค่าเป็น 0 ดังรูปที่ 3.25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

⊟ MES_Object.CONS.Handshake	{...}
MES_Object.CONS.Handshake.Trigger	0
MES_Object.CONS.Handshake.Working	0
MES_Object.CONS.Handshake.Completed	0
⊞ MES_Object.CONS.Handshake.Error	0
⊞ MES_Object.CONS.Handshake.Mode	2
⊞ MES_Object.CONS.Handshake.Response	0
⊞ MES_Object.CONS.Handshake.Config	{...}
⊞ MES_Object.CONS.Handshake.Counter	{...}

รูปที่ 3.25 Array ของฟังก์ชั่น Handshake ในขั้นตอน Consume

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.26



รูปที่ 3.26 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Consume

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) Operator จะทำการอัพเดท PLC tag (กรอกรายละเอียดของวัสดุที่ถูกใช้ไปในกระบวนการบรรจุภัณฑ์อาหารสัตว์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ศูนย์กลางรวบรวมข้อมูลจะทำการทริกเกอร์ Consumption สำหรับ PO context โดยใช้โหมดของ Consumption

ขอบเขตการทำงานของ MOM

1) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูล
2) MOM จะดำเนินการรับข้อมูล Consumption โดยการใช้โหมดจากศูนย์กลางรวบรวมข้อมูล

3.4.2.3 Produce

เป็นฟังก์ชั่นสำหรับการดำเนินการรายงาน Production ซึ่งเป็นขั้นตอนต่อจาก Consume โดย Operator จะทำการกรอกรายละเอียดของวัสดุที่ผลิตได้ในกระบวนการบรรจุภัณฑ์อาหารสัตว์และวัสดุที่ไม่ได้ตามมาตรฐานที่กำหนดไว้ให้กับศูนย์กลางรวบรวมข้อมูล และข้อมูลจะถูกศูนย์กลางรวบรวมข้อมูลนำมาจัดเก็บอยู่ใน array (MES_Object.PROD) ดังรูปที่ 3.27

MES_Object.PROD	{...}
MES_Object.PROD.Material	{...}
MES_Object.PROD.Material.MatID	'1089295'
MES_Object.PROD.Material.MatDesc	''
MES_Object.PROD.Material.Quantity	101.0
MES_Object.PROD.Material.UoM	'kg'
MES_Object.PROD.Material.Density	0.0
MES_Object.PROD.Material.CarrierID	'MARSTH_SU03'
MES_Object.PROD.Material.BatchID	''
MES_Object.PROD.Material.SublotID	''

รูปที่ 3.27 Array ของ MES_Object.PROD ในขั้นตอน Produce

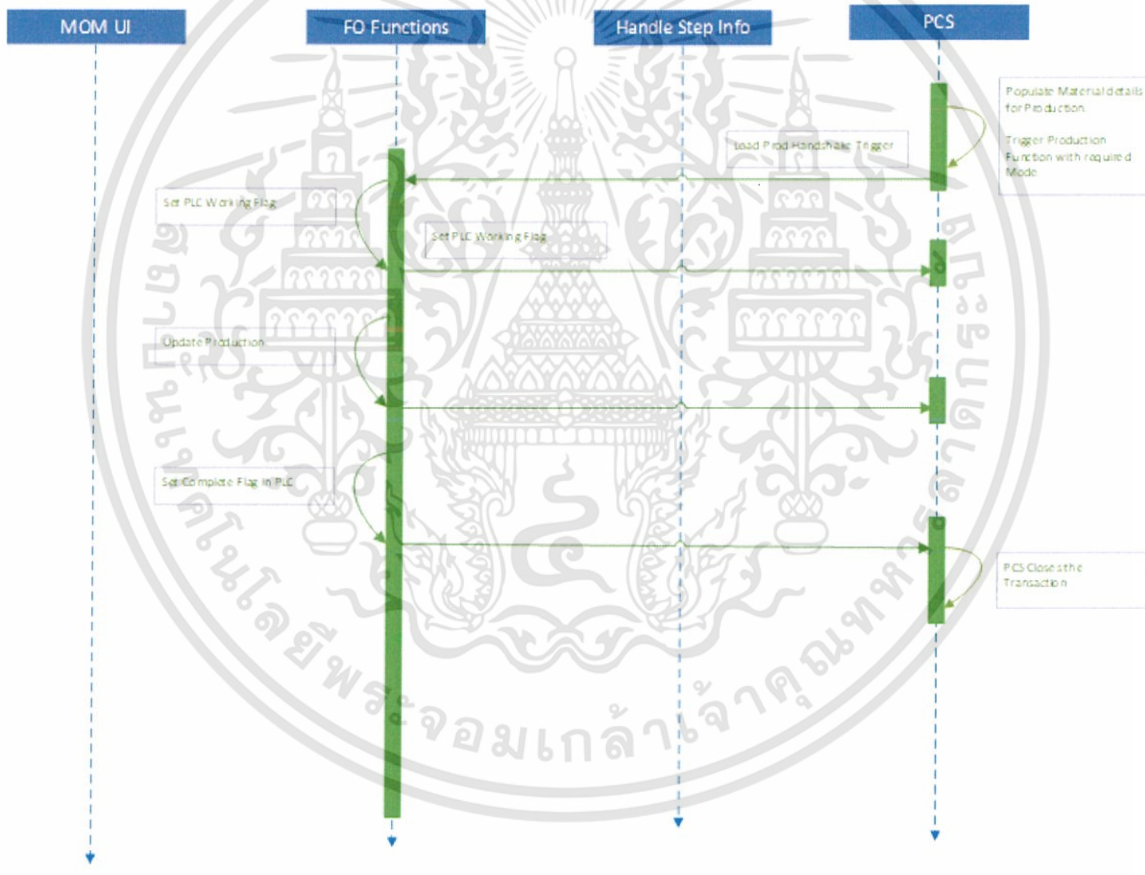
หลังจากข้อมูลถูกจัดเก็บอยู่ใน array (MES_Object.PROD) แล้ว ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Object PROD trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Object PROD trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการอัพเดท Production หลังจากนั้น MOM จะทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Produce เมื่อขั้นตอน Produce เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเคลียร์ Handshake Tag ทั้งหมดให้มีค่าเป็น 0 ดังรูปที่ 3.28

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

⊟ MES_Object.PROD.Handshake	{...}
MES_Object.PROD.Handshake.Trigger	0
MES_Object.PROD.Handshake.Working	0
MES_Object.PROD.Handshake.Completed	0
⊞ MES_Object.PROD.Handshake.Error	0
⊞ MES_Object.PROD.Handshake.Mode	1
⊞ MES_Object.PROD.Handshake.Response	0
⊞ MES_Object.PROD.Handshake.Config	{...}
⊞ MES_Object.PROD.Handshake.Counter	{...}

รูปที่ 3.28 Array ของฟังก์ชั่น Handshake ในขั้นตอน Produce

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.29



รูปที่ 3.29 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Produce

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) Operator จะทำการอัพเดท PLC tag (กรอกรายละเอียดของวัสดุที่ผลิตได้ในกระบวนการบรรจุภัณฑ์อาหารสัตว์)
2) ศูนย์กลางรวบรวมข้อมูลจะทำการทริกเกอร์ Production สำหรับ PO context โดยใช้โหมดของ Production

ขอบเขตการทำงานของ MOM

1) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูล
2) MOM จะดำเนินการรับข้อมูล Production และ Waste โดยการใช้โหมดจาก ศูนย์กลางรวบรวมข้อมูล

3.4.2.4 Complete Unit PO

เป็นฟังก์ชั่นสำหรับการดำเนินการ complete ของ Process Unit ซึ่งเป็นขั้นตอนต่อจาก Produce โดยคำสั่งจะถูกจัดเก็บอยู่ใน array (MES_Object.Context) ดังรูปที่ 3.30

− MES_Object	{...}
− MES_Object.Context	{...}
+ MES_Object.Context.PO	'ORDER001'
+ MES_Object.Context.POItem	'ORDER001_ORDERITEM_01'
+ MES_Object.Context.Sequence	1
− MES_Object.Context.Material	{...}
+ MES_Object.Context.Material.MatID	'1089295'
+ MES_Object.Context.Material.MatDesc	'1089295 Description'
MES_Object.Context.Material.Quantity	1100.0
+ MES_Object.Context.Material.UoM	'kg'
MES_Object.Context.Material.Density	0.0
+ MES_Object.Context.Material.CarrierID	''
+ MES_Object.Context.Material.BatchID	''
+ MES_Object.Context.Material.SublotID	''
+ MES_Object.PO_Status	0

รูปที่ 3.30 Array ของ MES_Object.Context ในขั้นตอน Complete Unit PO

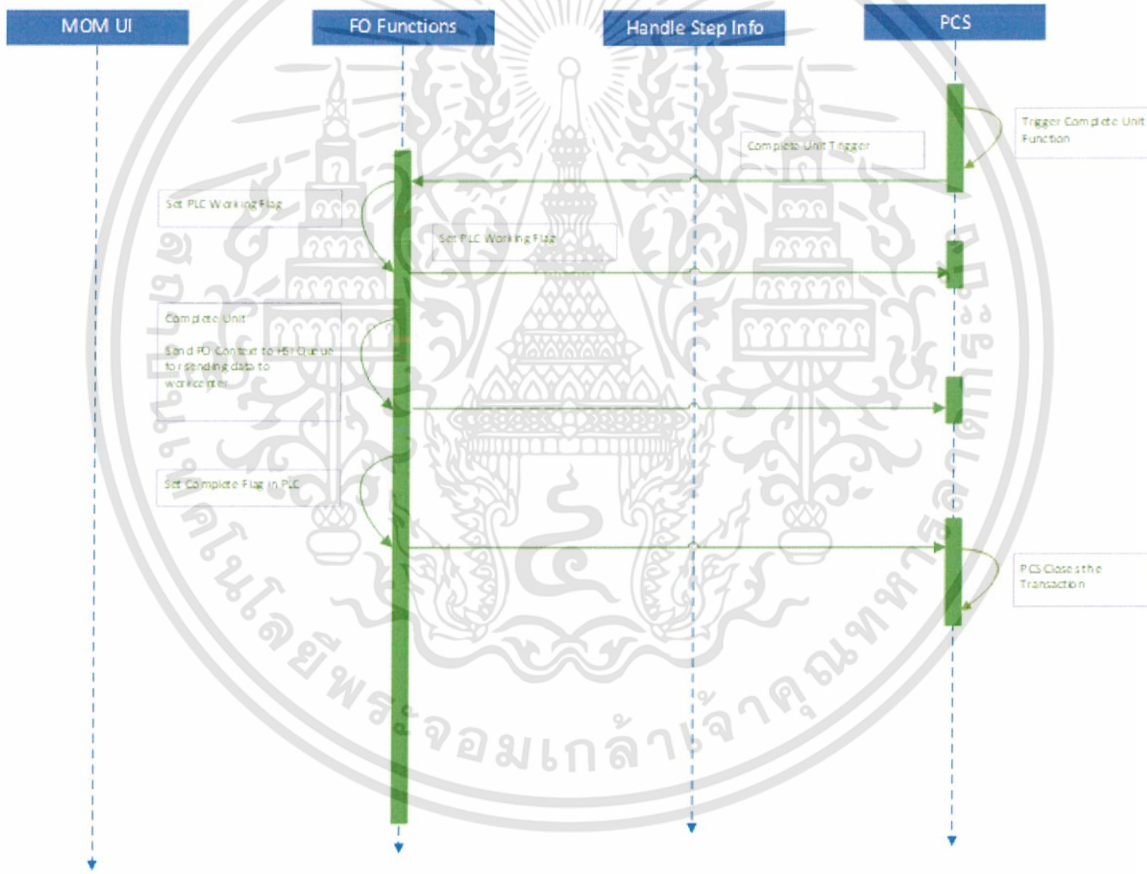
หลังจากคำสั่งถูกจัดเก็บอยู่ใน array (MES_Object.Context) แล้วและขั้นตอนของ Start Unit PO, Consume และ Produce เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเซ็ท Complete Unit trigger ให้มีค่าเป็น 1 เพื่อเริ่มต้นการทำงาน หลังจากที่ MOM ได้รับ Complete Unit trigger แล้วจะทำการเซ็ท Working Flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลและทำการ Complete Unit หลังจากนั้น MOM จะทำการเซ็ท Complete flag ส่งกลับมาให้ศูนย์กลางรวบรวมข้อมูลเพื่อจบการทำงานในขั้นตอน Complete Unit PO เมื่อขั้นตอน Complete Unit PO เสร็จสมบูรณ์ ศูนย์กลางรวบรวมข้อมูลจะทำการเคลียร์ Handshake Tag ทั้งหมดให้มีค่าเป็น 0 ดังรูปที่ 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MES_Object.Complete	{...}
MES_Object.Complete.Trigger	0
MES_Object.Complete.Working	0
MES_Object.Complete.Completed	0
MES_Object.Complete.Error	0
MES_Object.Complete.Mode	0
MES_Object.Complete.Response	0
MES_Object.Complete.Config	{...}
MES_Object.Complete.Counter	{...}

รูปที่ 3.31 Array ของฟังก์ชั่น Handshake ในขั้นตอน Complete Unit PO

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS และ MOM จะแสดงดังแผนผังการทำงานแบบลำดับ (Sequence Diagram) รูปที่ 3.32



รูปที่ 3.32 แผนผังการทำงานแบบลำดับ (Sequence Diagram) แสดงขั้นตอนการทำงานของ Complete Unit PO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขอบเขตการทำงานของศูนย์กลางรวบรวมข้อมูลใน PCS

1) ศูนย์กลางรวบรวมข้อมูลจะทำการทริกเกอร์ Unit Complete ของ PO context

ขอบเขตการทำงานของ MOM

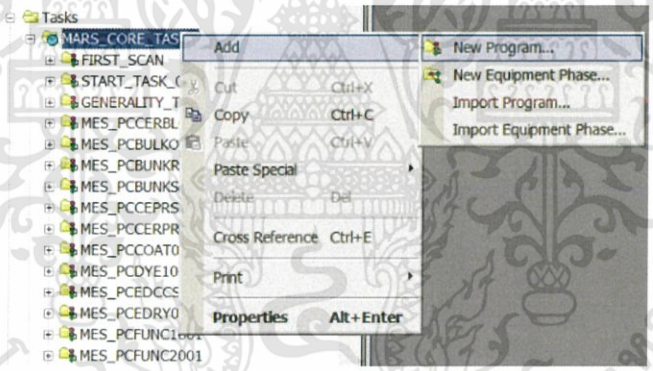
1) MOM จะนำ PO context มาจากศูนย์กลางรวบรวมข้อมูล
2) MOM จะทำการ Complete Unit สำหรับ PO context

3.5 ขั้นตอนการเขียนโปรแกรม Studio 5000

เนื่องจากโปรแกรมของศูนย์กลางรวบรวมข้อมูลสำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์ที่ได้สร้างขึ้น ได้ทำการเขียนโปรแกรมต่อจากโปรแกรมของกระบวนการผลิตอื่น ๆ ในโรงงาน ดังนั้นในขั้นตอนแรกจะกล่าวถึงการสร้าง New Program

3.5.1 การสร้าง New Program

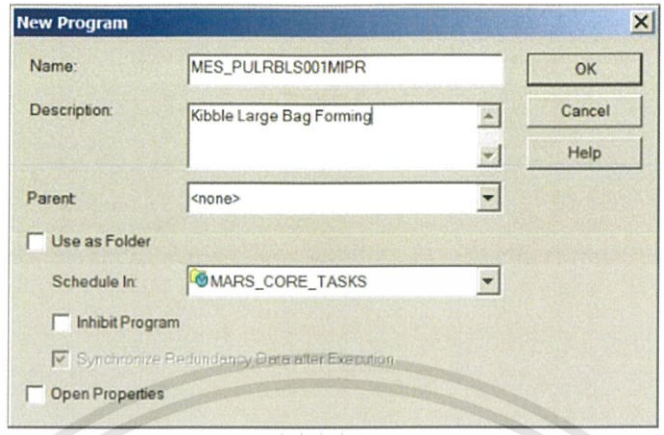
1) คลิกขวาที่ MainTask แล้วเลือก Add > New Program เพื่อเพิ่มโปรแกรมที่ใช้เข้าไป ดังรูปที่ 3.33



รูปที่ 3.33 การเพิ่มโปรแกรมที่ใช้ลงใน MainTask

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

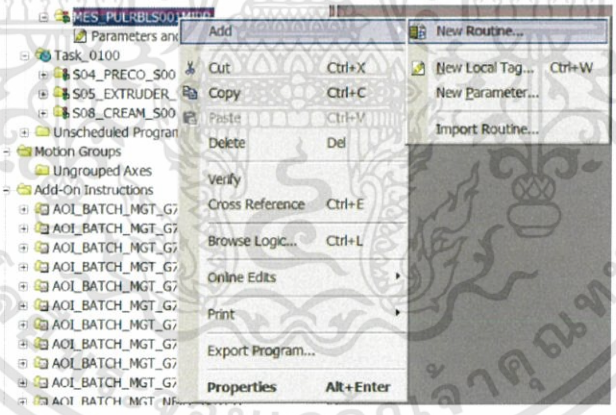
2) ตั้งชื่อโปรแกรมและใส่คำอธิบาย แล้วคลิก OK ดังรูปที่ 3.34



รูปที่ 3.34 การตั้งชื่อโปรแกรม

3.5.2 การสร้าง New Routine

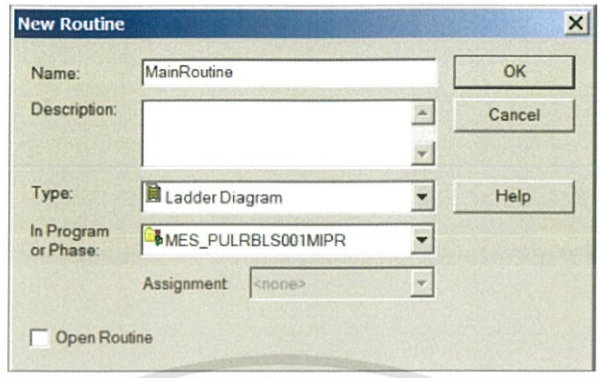
1) คลิกขวาที่ MainProgram เลือก Add > New Routine เพื่อสร้าง Routine ซึ่งเป็นพื้นที่หรือหน้าต่างสำหรับเขียนโปรแกรม PLC ดังรูปที่ 3.35



รูปที่ 3.35 การสร้าง Routine สำหรับเขียนโปรแกรมใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

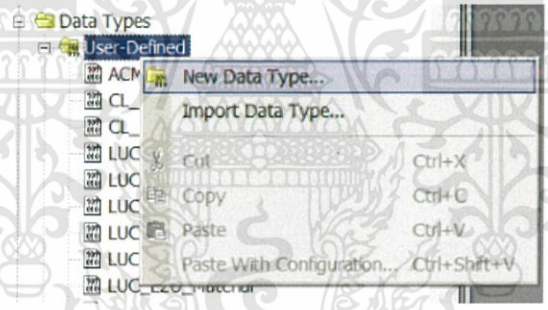
2) เลือกภาษาในการเขียนโปรแกรม และตั้งชื่อ Routine แล้วคลิก OK ดังรูปที่ 3.36



รูปที่ 3.36 การตั้งชื่อ Routine และกำหนดภาษาเขียนโปรแกรม

3.5.3 การสร้าง User-Defined Data Types

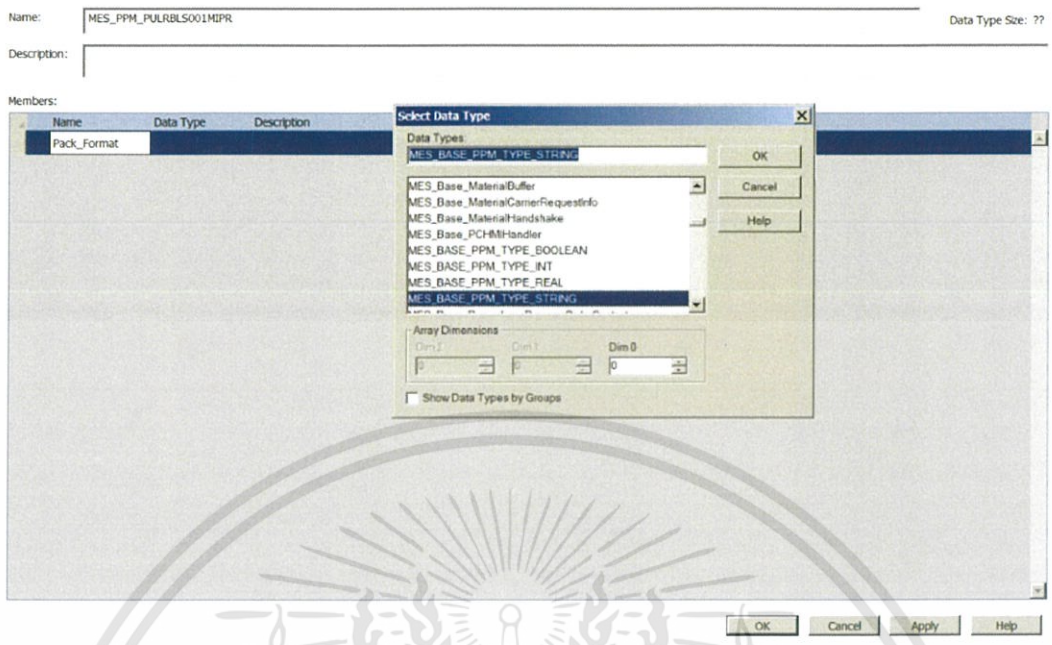
1) คลิกขวาที่ User-Defined เลือก New Data Type เพื่อสร้างชนิดข้อมูล (Data Type) ซึ่งนำไปใช้ในการสร้าง Tag ต่าง ๆ ดังรูปที่ 3.37



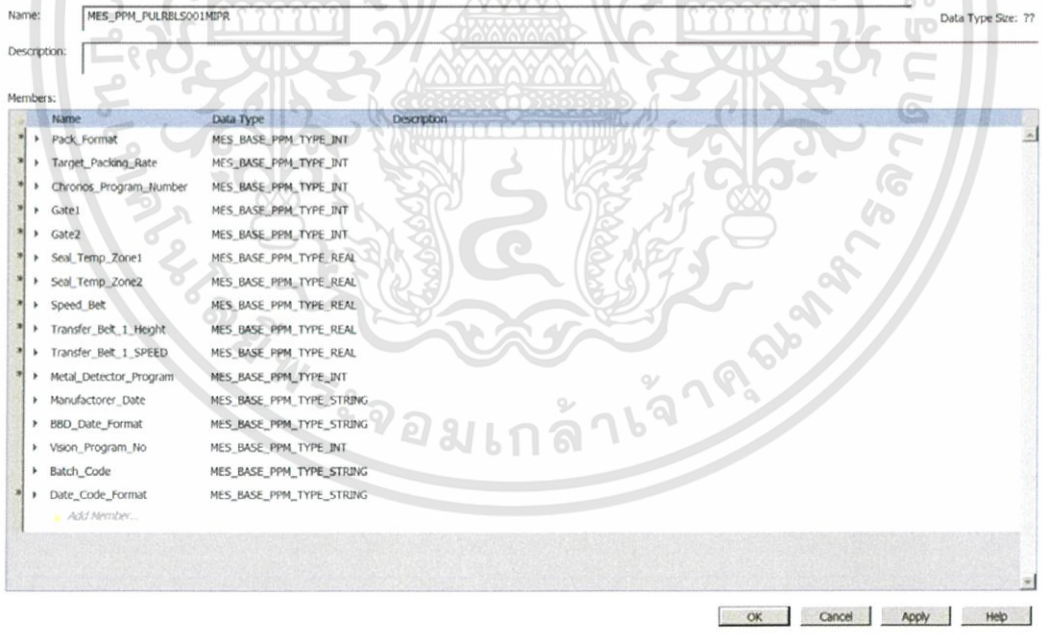
รูปที่ 3.37 การสร้าง Data Type

2) พิมพ์ชื่อ Data Type ลงในช่อง Name และป้อนคำอธิบายลงในส่วนของ Description จากนั้นป้อนชื่อสมาชิกและ กำหนด Data type จนครบทั้งหมด เสร็จแล้วคลิก OK ดังรูปที่ 3.38 และตัวอย่างของ Data type จะแสดงดังรูปที่ 3.39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.38 การตั้งชื่อ Data Type และกำหนดภาษาเขียนโปรแกรม

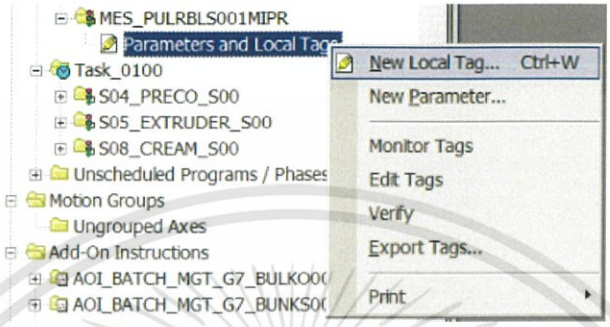


รูปที่ 3.39 ตัวอย่าง User-Defined Data Types

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

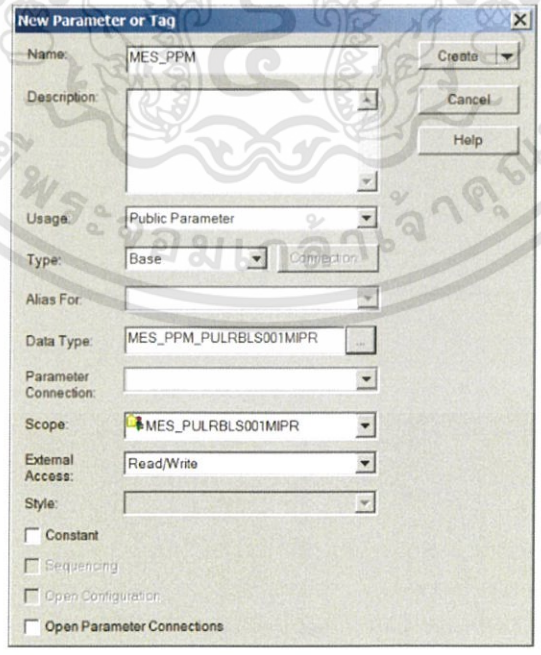
3.5.4 การสร้าง Local Tag

1) คลิกขวาที่ Parameter and Local Tags เลือก New Local Tag เพื่อสร้าง Local Tag ซึ่งเป็น Tag ที่สามารถมองเห็นและใช้งานได้เฉพาะในโปรแกรมนั้น ๆ เท่านั้น ดังรูปที่ 3.40



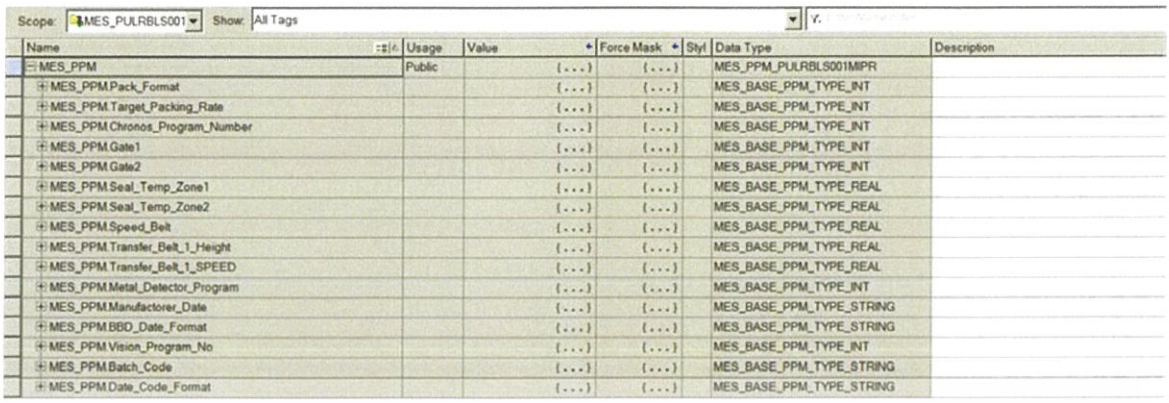
รูปที่ 3.40 การสร้าง Local Tag สำหรับ Tag ที่ใช้ในภายใน Main Program

2) ตั้งชื่อ Tag แล้วเลือก Usage เป็น Public, Local, Input, Output, InOut ซึ่ง Public ใช้กับข้อมูลขนาดใหญ่ที่ต้องใช้ร่วมกันระหว่างโปรแกรม และกำหนดชนิด (Type) ของ Tag ซึ่งใน ControlLogix มีอยู่ 4 ชนิด คือ Base, Alias, Produced และ Consumed แล้วเลือก Data type เป็น Bool , Integer, Real และ UDT อื่น ๆ แล้วคลิก OK ดังรูปที่ 3.41 และตัวอย่างของ Local Tag จะแสดงดังรูปที่ 3.42



รูปที่ 3.41 การตั้งชื่อ Local Tag และกำหนดรายละเอียดต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



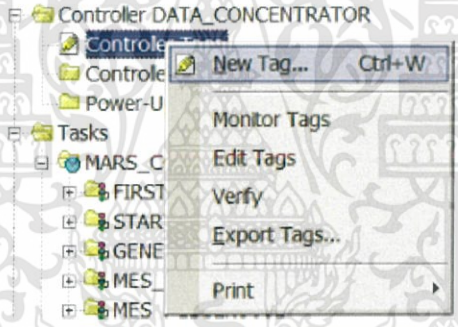
Scope: MES_PULRBLS001 Show: All Tags

Name	Usage	Value	Force Mask	Styl	Data Type	Description
MES_PPM	Public	{...}	{...}		MES_PPM_PULRBLS001MIPR	
MES_PPM.Pack_Format		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Target_Packing_Rate		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Chronos_Program_Number		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Gate1		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Gate2		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Seal_Temp_Zone1		{...}	{...}		MES_BASE_PPM_TYPE_REAL	
MES_PPM.Seal_Temp_Zone2		{...}	{...}		MES_BASE_PPM_TYPE_REAL	
MES_PPM.Speed_Belt		{...}	{...}		MES_BASE_PPM_TYPE_REAL	
MES_PPM.Transfer_Belt_1_Height		{...}	{...}		MES_BASE_PPM_TYPE_REAL	
MES_PPM.Transfer_Belt_1_SPEED		{...}	{...}		MES_BASE_PPM_TYPE_REAL	
MES_PPM.Metal_Detector_Program		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Manufactorer_Date		{...}	{...}		MES_BASE_PPM_TYPE_STRING	
MES_PPM.BBD_Date_Format		{...}	{...}		MES_BASE_PPM_TYPE_STRING	
MES_PPM.Vision_Program_No		{...}	{...}		MES_BASE_PPM_TYPE_INT	
MES_PPM.Batch_Code		{...}	{...}		MES_BASE_PPM_TYPE_STRING	
MES_PPM.Date_Code_Format		{...}	{...}		MES_BASE_PPM_TYPE_STRING	

รูปที่ 3.42 ตัวอย่าง Local Tag

3.5.5 การสร้าง Controller Tag

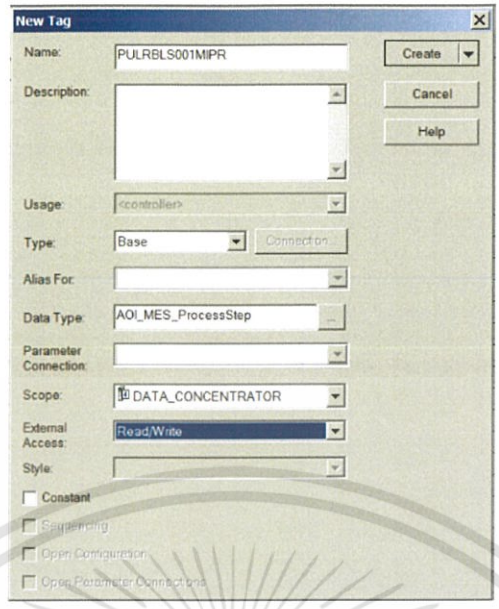
1) คลิกขวาที่ Controller Tags เลือก New Tag เพื่อสร้าง Controller Tag ซึ่งเป็น Tag ที่สามารถมองเห็นและใช้งานได้จากทั้งภายนอกหรือภายในโปรแกรม ดังรูปที่ 3.43



รูปที่ 3.43 การสร้าง Controller Tag สำหรับ Tag ที่ใช้ทั้งภายนอกหรือภายในโปรแกรม

2) ตั้งชื่อ Tag แล้วกำหนดชนิด (Type) ของ Tag ซึ่งใน ControlLogix มีอยู่ 4 ชนิด คือ Base, Alias, Produced และ Consumed แล้วเลือก Data type เป็น Bool , Integer, Real และ UDT อื่น ๆ แล้วคลิก OK ดังรูปที่ 3.44 และตัวอย่างของ Controller Tag จะแสดงดังรูปที่ 3.45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.44 การตั้งชื่อ Controller Tag และกำหนดรายละเอียดต่าง ๆ

Scope: DATA_CONCENTF Show: All Tags

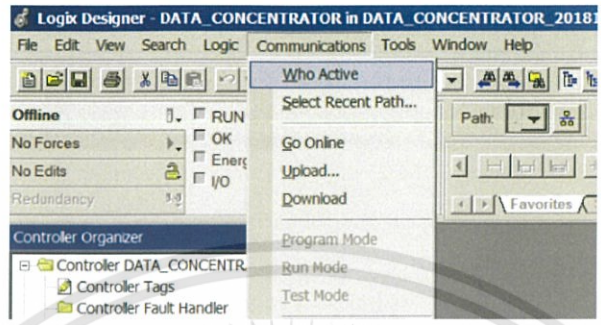
Name	Value	Force Mask	Styl	Data Type	Description
PULRBLS001MIPR	{...}	{...}		AOI_MES_ProcessStep	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.EnableIn	1		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.EnableOut	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.StartMode	0		De	DINT	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.CompleteMode	0		De	DINT	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.ConsumeMode	0		De	DINT	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.ProduceMode	0		De	DINT	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.RequestStepInfoMode	0		De	DINT	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerConsume	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerProduce	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerRequestStepInfo	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.AllowExternalTrigger	1		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerStartExternal	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerCompleteExternal	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerConsumeExternal	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerProduceExternal	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.TriggerRequestStepInfoExternal	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.ResetExternal	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.OrderContextAvailable	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.StepInfoInvalid	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.ReadyToStart	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.Starting	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.StartCompleted	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.StartError	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.StartTimeout	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.StartResetting	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.Start_Trigger	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.Start_Working	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.Start_Completed	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.ReadyToComplete	0		De	BOOL	Process Step instruction for Handling Order Context drive
PULRBLS001MIPR.Completing	0		De	BOOL	Process Step instruction for Handling Order Context drive

รูปที่ 3.45 ตัวอย่าง Controller Tag

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

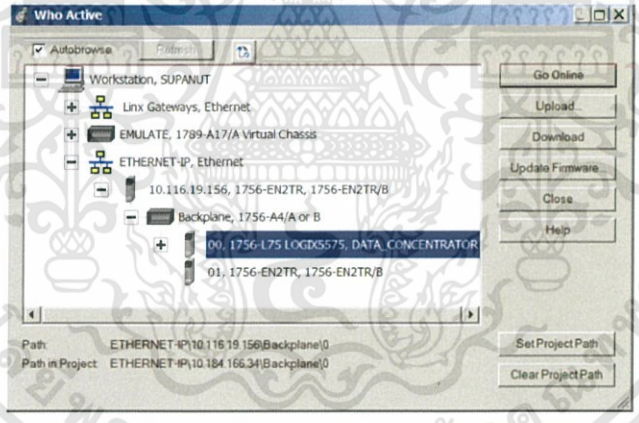
3.5.6 การออนไลน์โปรแกรม

1) คลิกขวาที่ Communications เลือก Who Active เพื่อทำการเลือกคอนโทรลเลอร์ที่ต้องการใช้ในการออนไลน์โปรแกรม PLC ดังรูปที่ 3.46



รูปที่ 3.46 การออนไลน์โปรแกรม PLC

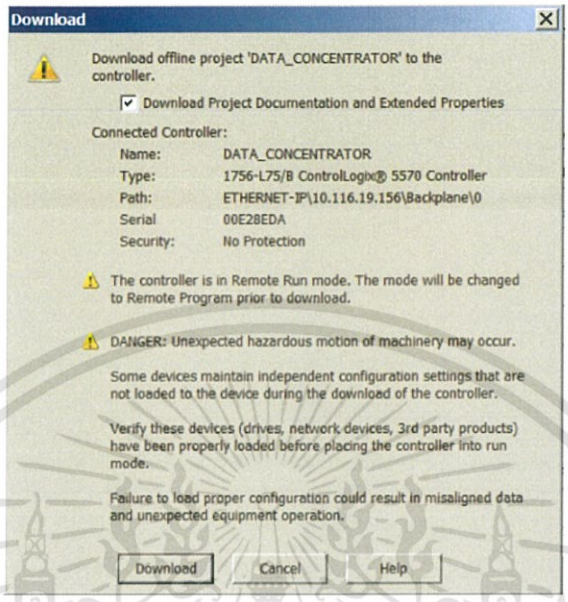
2) เลือกคอนโทรลเลอร์ที่ต้องการใช้ในการออนไลน์โปรแกรม PLC แล้วคลิก OK ดังรูปที่ 3.47



รูปที่ 3.47 การเลือกคอนโทรลเลอร์ที่ต้องการใช้ในการออนไลน์โปรแกรม PLC

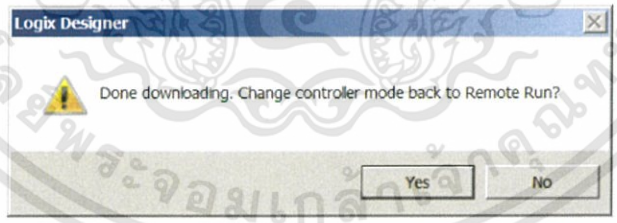
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) คลิกที่ Download เพื่อทำการดาวน์โหลดโปรแกรม PLC ลงในคอนโทรลเลอร์ ดังรูปที่ 3.48



รูปที่ 3.48 การดาวน์โหลดโปรแกรม PLC ลงในคอนโทรลเลอร์

4) หลังจากดาวน์โหลดโปรแกรมลงคอนโทรลเลอร์แล้ว เราสามารถสั่งให้คอนโทรลเลอร์ Run โปรแกรมได้ 2 วิธีคือ บิดสวิตซ์ไปยังตำแหน่ง “REM” แล้วคลิก Yes ดังรูปที่ 3.49 หรือ บิดสวิตซ์เลือกโหมดของPLC ไปที่ตำแหน่ง “RUN”

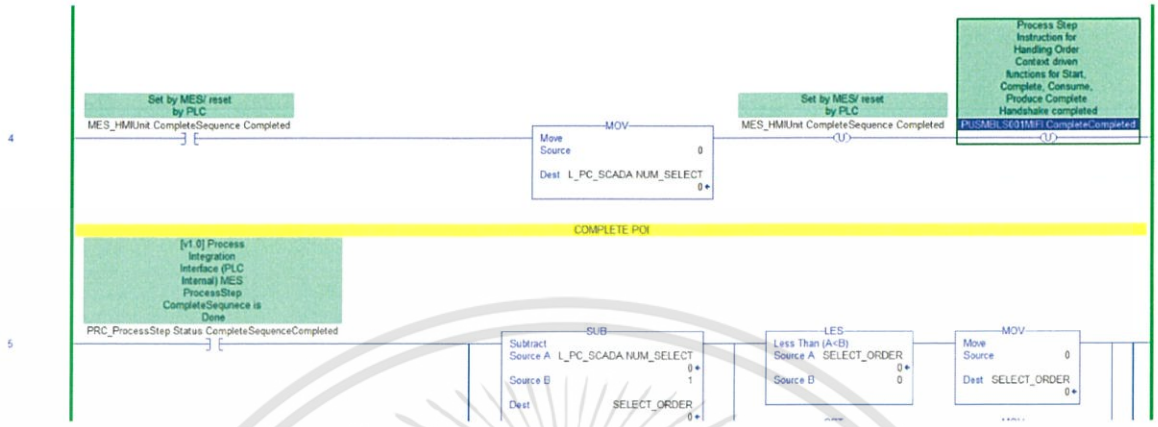


รูปที่ 3.49 การสั่งให้คอนโทรลเลอร์ Run โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

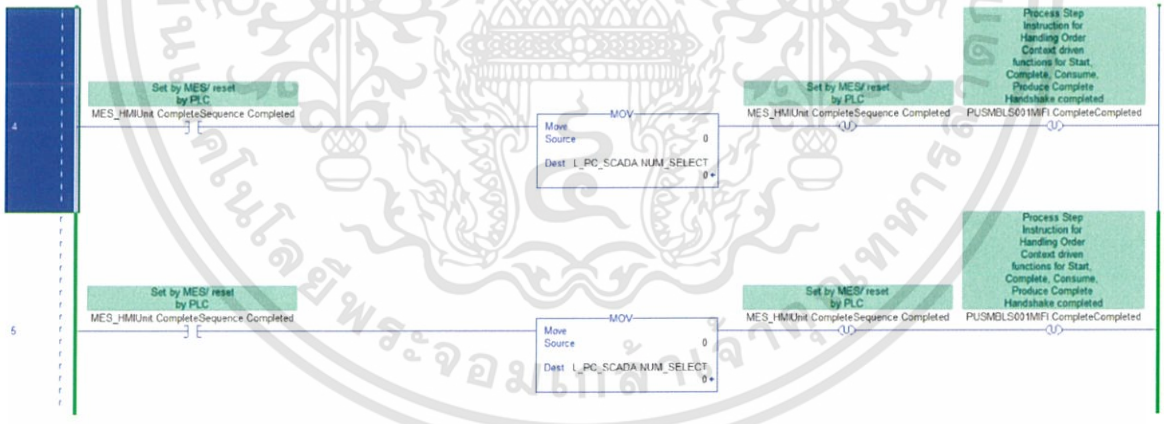
3.5.7 การแก้ไขออนไลน์

1) ออนไลน์โปรแกรมแล้ว ดับเบิ้ลคลิก Rung ที่ต้องการแก้ไข ดังรูปที่ 3.50



รูปที่ 3.50 การ Online Edit โปรแกรม

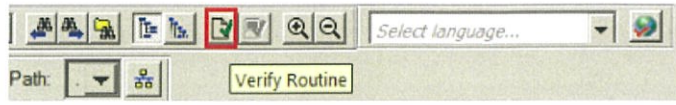
2) จะเห็นว่า Rung ที่เราต้องการแก้ไขถูกคัดลอกขึ้นมาอีก 1 ชุด โดย Rung ที่ขึ้นต้นด้วย “i” จะเป็น Rung ที่เราสามารถแก้ไขได้ ส่วน Rung ที่นำหน้าด้วย “r” จะเป็น Rung ต้นฉบับที่PLC ใช้ Run อยู่ จากนั้นให้ทำการแก้ไขลอจิก ดังรูปที่ 3.51



รูปที่ 3.51 การแก้ไขลอจิกในโปรแกรม

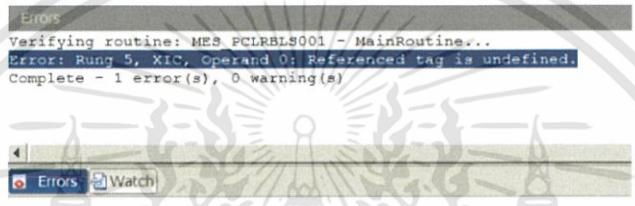
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) คลิกที่ “Verify Routine” เพื่อตรวจสอบความถูกต้องของโปรแกรม ดังรูปที่ 3.52



รูปที่ 3.52 ปุ่มVerify Routine

4) ถ้าไม่มี error จะแสดงจำนวน error เป็น 0 ถ้ามี error เกิดขึ้น ต้องทำการแก้ไข error ก่อน ถึงสามารถ Download โปรแกรมลงPLC ได้ ดังรูปที่ 3.53



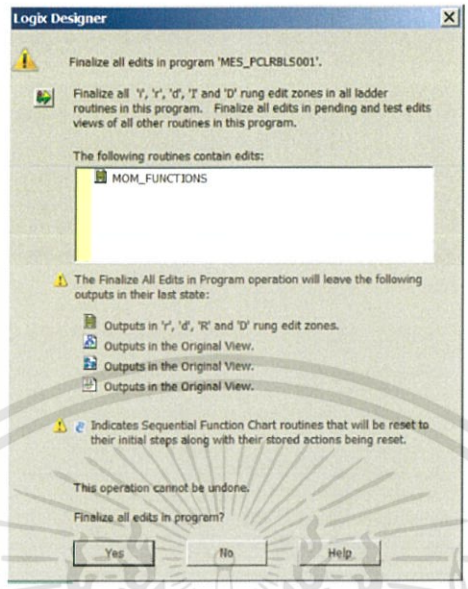
รูปที่ 3.53 หน้าต่างแสดง error

5) ถ้าต้องการยืนยันการแก้ไขและใช้งานลอจิกที่เพิ่มเข้าไป ให้คลิกที่ “Finalize All Edits” ดังรูปที่ 3.54

รูปที่ 3.54 ปุ่ม Finalize All Edits

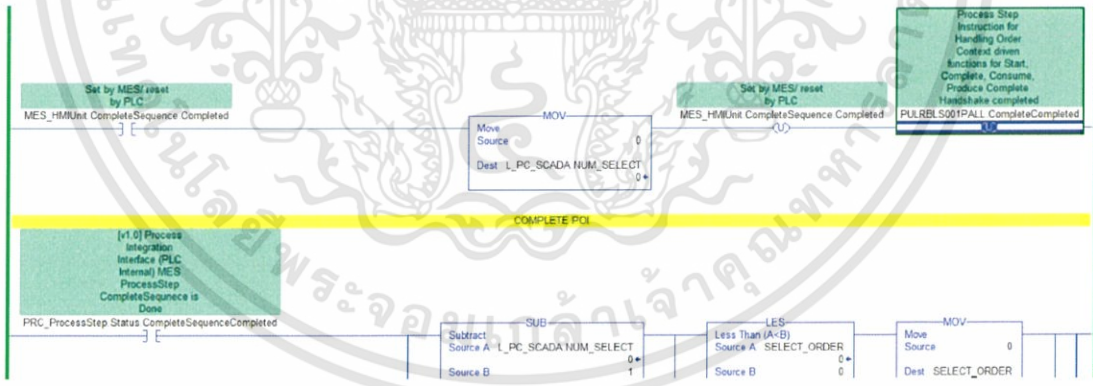
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) คลิกที่ Yes เพื่อเป็นการยืนยันการแก้ไขอีกครั้ง ดังรูปที่ 3.55



รูปที่ 3.55 การยืนยันการแก้ไข

7) Studio 5000 จะลบ Rung ต้นฉบับแล้วนำ Rung ที่ทำการแก้ไขมาใช้งาน ดังรูปที่ 3.56

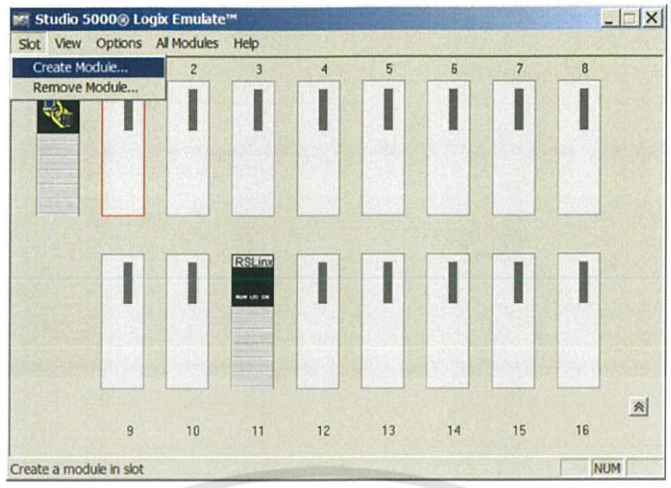


รูปที่ 3.56 ตัวอย่างลอจิกที่ทำการแก้ไขแล้ว

3.5.8 การทดสอบโปรแกรม โดยใช้โปรแกรม Studio5000 Logix Emulate

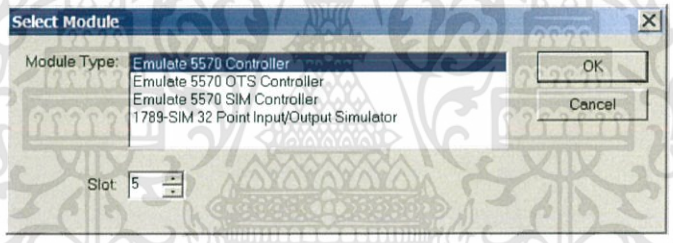
1) เปิดโปรแกรม Studio5000 Logix Emulate ขึ้นมา คลิกที่ Slot เลือก Create Module เพื่อสร้างPLC จำลอง ในกรณีที่ต้องการทดสอบโปรแกรมโดยไม่มีคอนโทรลเลอร์และ I/O โมดูลจริง ดังรูปที่ 3.57

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



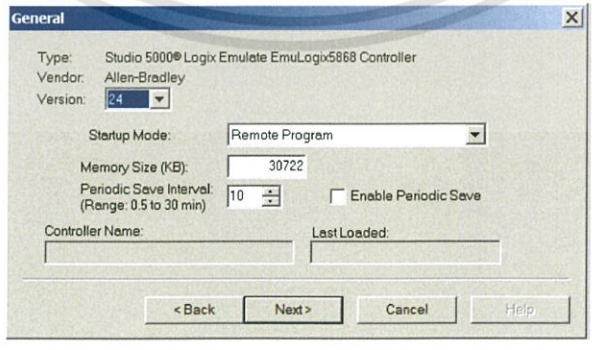
รูปที่ 3.57 การสร้าง PLC จำลอง

2) เลือกโมดูลเป็น Emulate 5570 Controller แล้วเลือก Slot ตาม Slot ที่คอนโทรลเลอร์จริงติดตั้งอยู่ จากนั้นคลิก OK ดังรูปที่ 3.58



รูปที่ 3.58 การเลือกโมดูลของคอนโทรลเลอร์

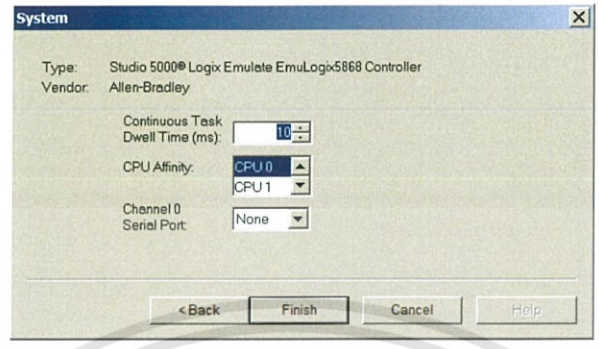
3) เลือก Version เลือก Startup Mode, Memory และ Periodic Save Interval ของคอนโทรลเลอร์ที่ใช้งานจากนั้นคลิกที่ Next ดังรูปที่ 3.59



รูปที่ 3.59 การกำหนดรายละเอียดต่าง ๆ ของคอนโทรลเลอร์

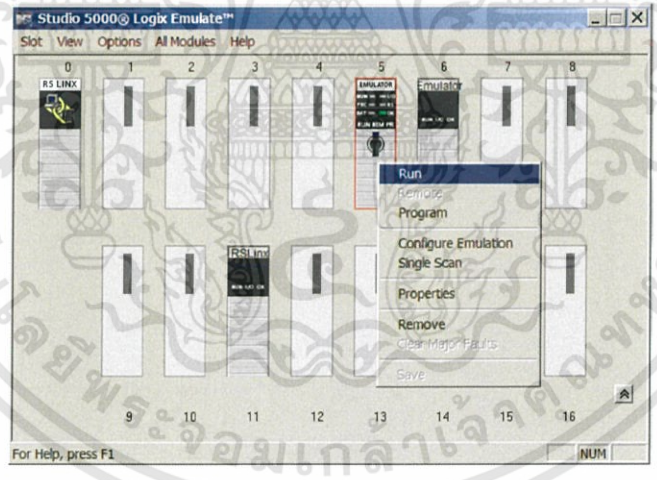
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) เลือก Continuous Task Dwell Time (ms) , CPU Affinity และ Channel 0 Serial Port ที่ต้องการ จากนั้นคลิกที่ Finish ดังรูปที่ 3.60



รูปที่ 3.60 การกำหนดรายละเอียดต่าง ๆ ของคอนโทรลเลอร์ (ต่อ)

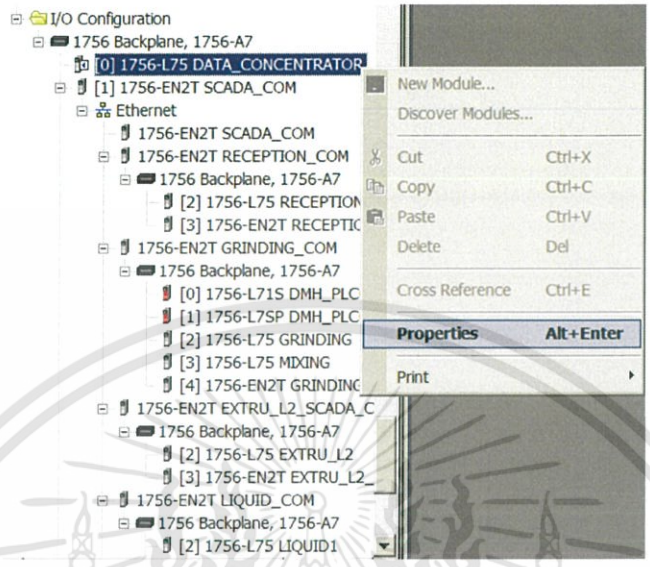
5) หลังจากกำหนดรายละเอียดต่าง ๆ ให้คอนโทรลเลอร์แล้ว เราสามารถสั่งให้คอนโทรลเลอร์ Run โปรแกรมได้ คลิกขวาที่ Slot ที่คอนโทรลเลอร์จำลองติดตั้งอยู่ แล้วเลือก Run ดังรูปที่ 3.61



รูปที่ 3.61 การสั่งให้คอนโทรลเลอร์จำลอง Run โปรแกรม

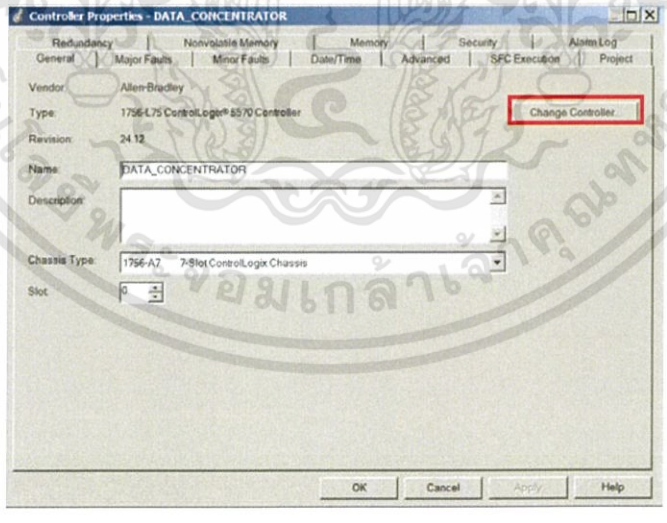
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) เปิดโปรแกรม Studio5000 ไปที่ I/O Configuration คลิกขวาที่คอนโทรลเลอร์ที่ต้องการทดสอบ แล้วเลือก Properties ดังรูปที่ 3.62



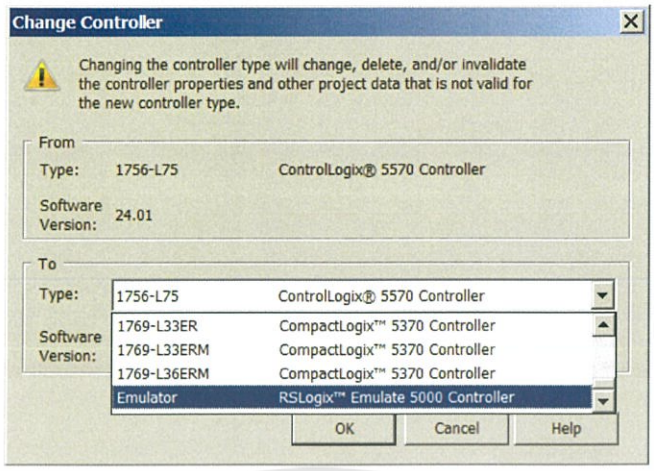
รูปที่ 3.62 การเลือกคอนโทรลเลอร์ที่ต้องการทดสอบ

7) คลิกที่ Change Controller แล้วเปลี่ยน Controller Type เป็น Emulate Controller จากนั้นคลิกที่ OK ดังรูปที่ 3.63 และรูปที่ 3.64 ตามลำดับ



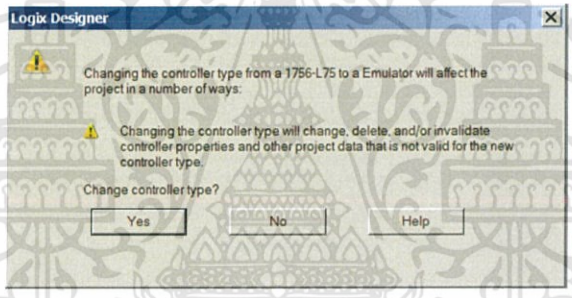
รูปที่ 3.63 การเปลี่ยนชนิดของคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.64 การเปลี่ยนชนิดของคอนโทรลเลอร์ (ต่อ)

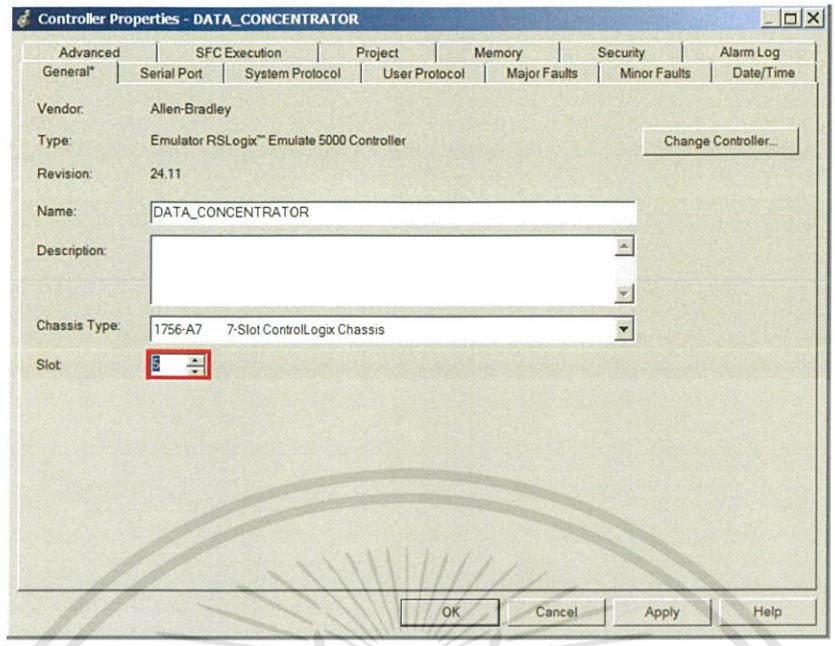
8) คลิกปุ่ม Yes เพื่อยืนยันการเปลี่ยน Controller Type ดังรูปที่ 3.65



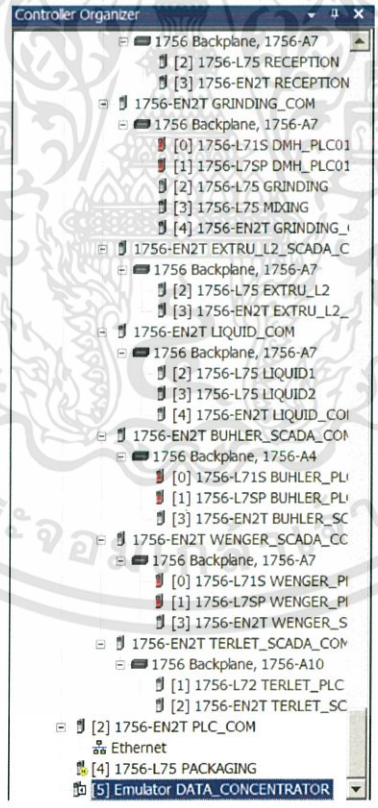
รูปที่ 3.65 การยืนยันการเปลี่ยนชนิดของคอนโทรลเลอร์

9) หลังจากเปลี่ยน Controller Type เป็น Emulator RSLogix แล้ว ให้เปลี่ยนหมายเลขของ Slot ให้ตรงกับที่สร้างไว้ใน Studio5000 Logix Emulate หลังจากนั้นคลิกปุ่ม OK เพื่อปิดหน้าต่าง Controller Properties ดังรูปที่ 3.66 และตัวอย่างของคอนโทรลเลอร์ที่เปลี่ยนเป็น Emulator จะแสดงดังรูปที่ 3.67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



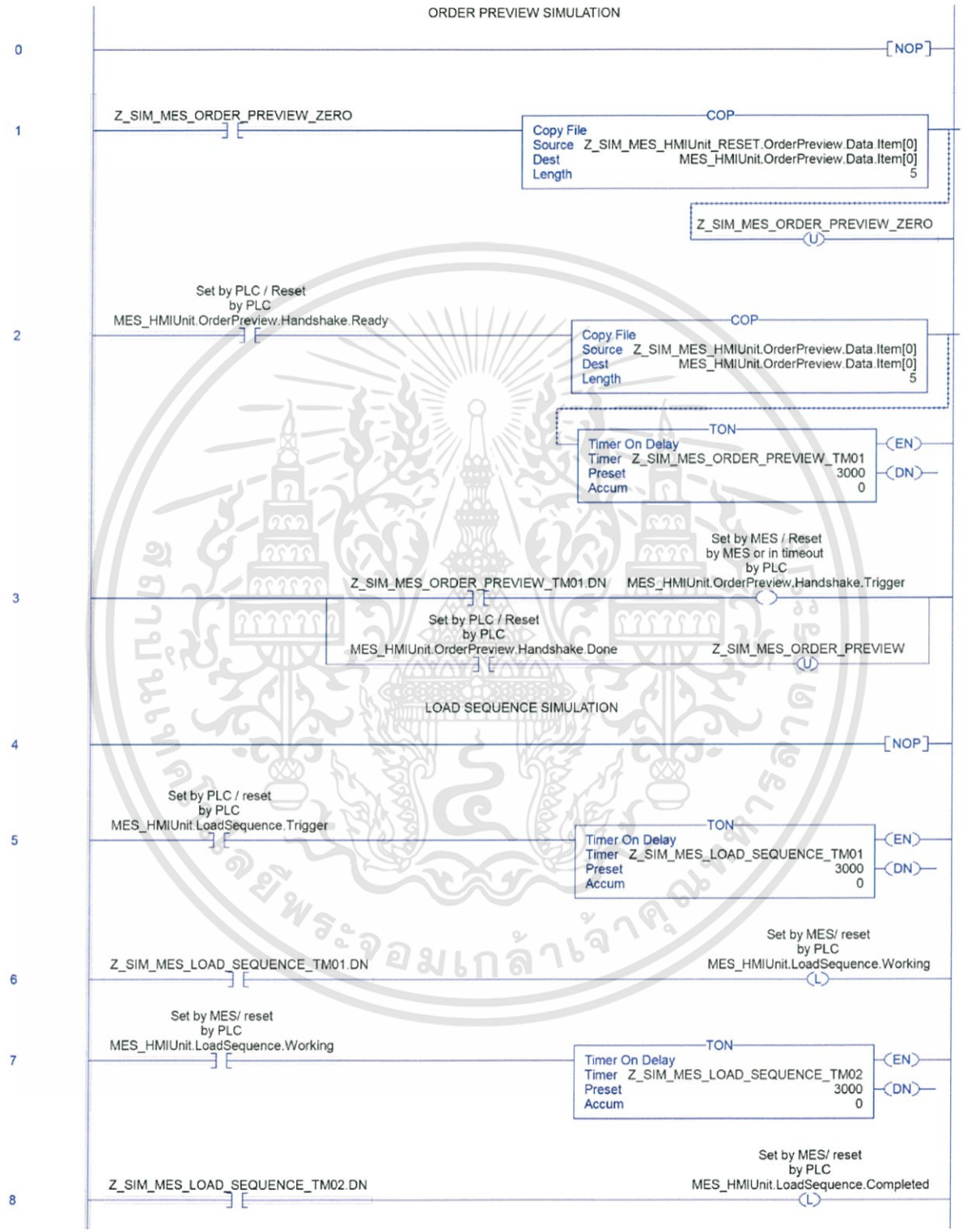
รูปที่ 3.66 การเปลี่ยนหมายเลข Slot



รูปที่ 3.67 ตัวอย่างคอนโทรลเลอร์ที่เปลี่ยนเป็น Emulator

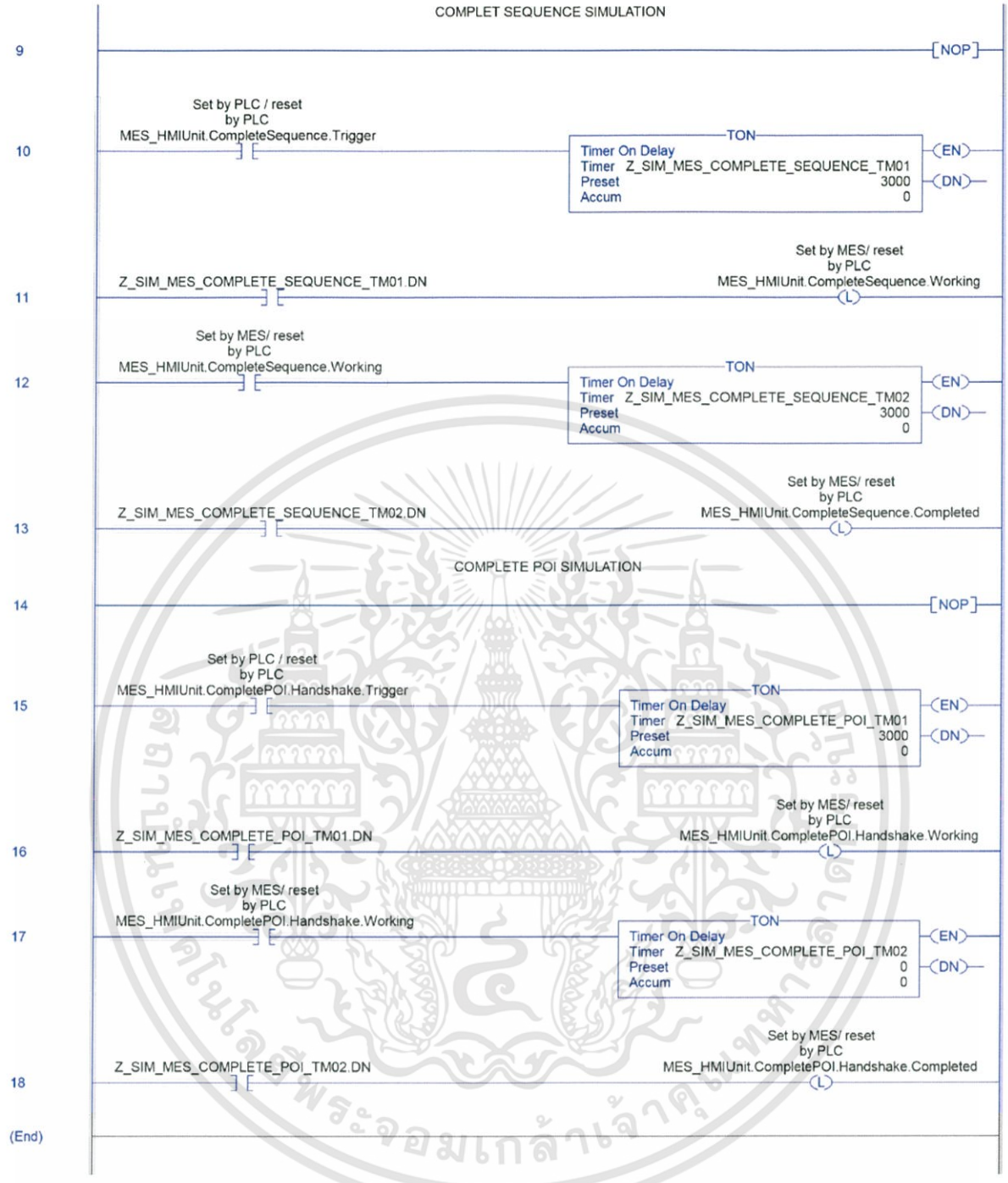
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของโปรแกรมศูนย์กลางรวบรวมข้อมูลที่ได้สร้างขึ้นบางส่วน จะแสดงดังรูปที่ 3.68, 3.69, 3.70, 3.71, 3.72 และ 3.73



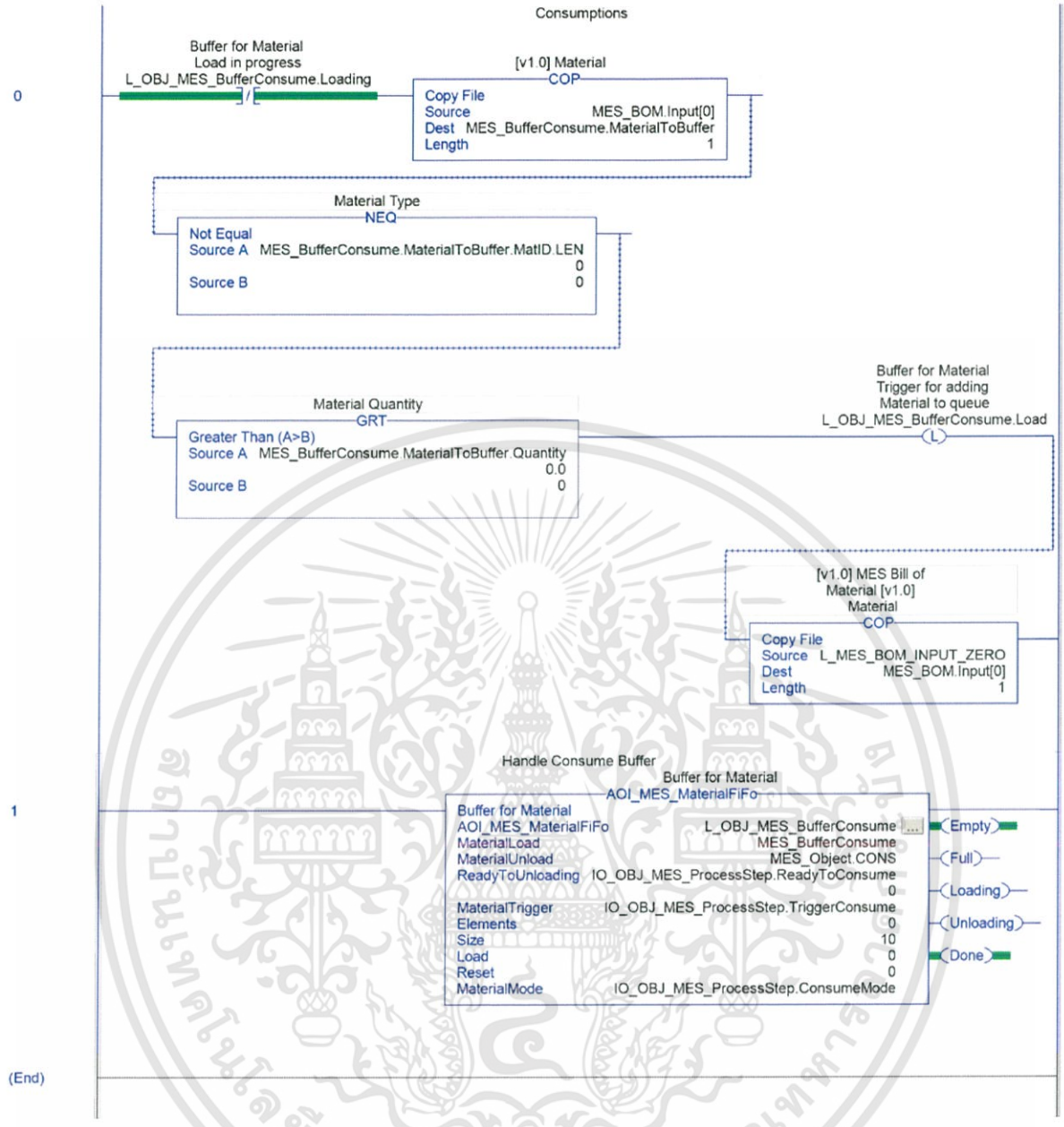
รูปที่ 3.68 ตัวอย่างโปรแกรม Simulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



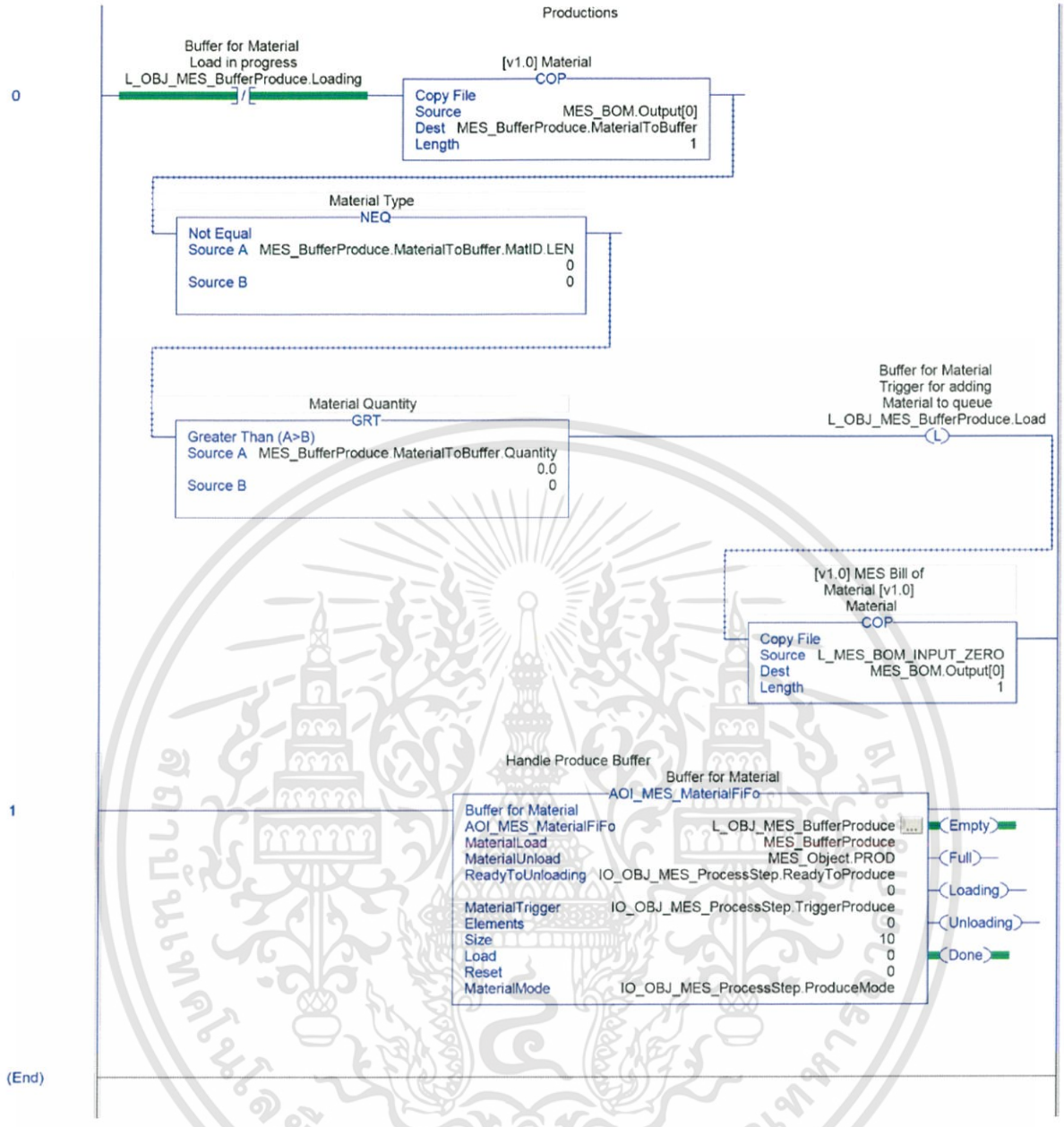
รูปที่ 3.69 ตัวอย่างโปรแกรม Simulation (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



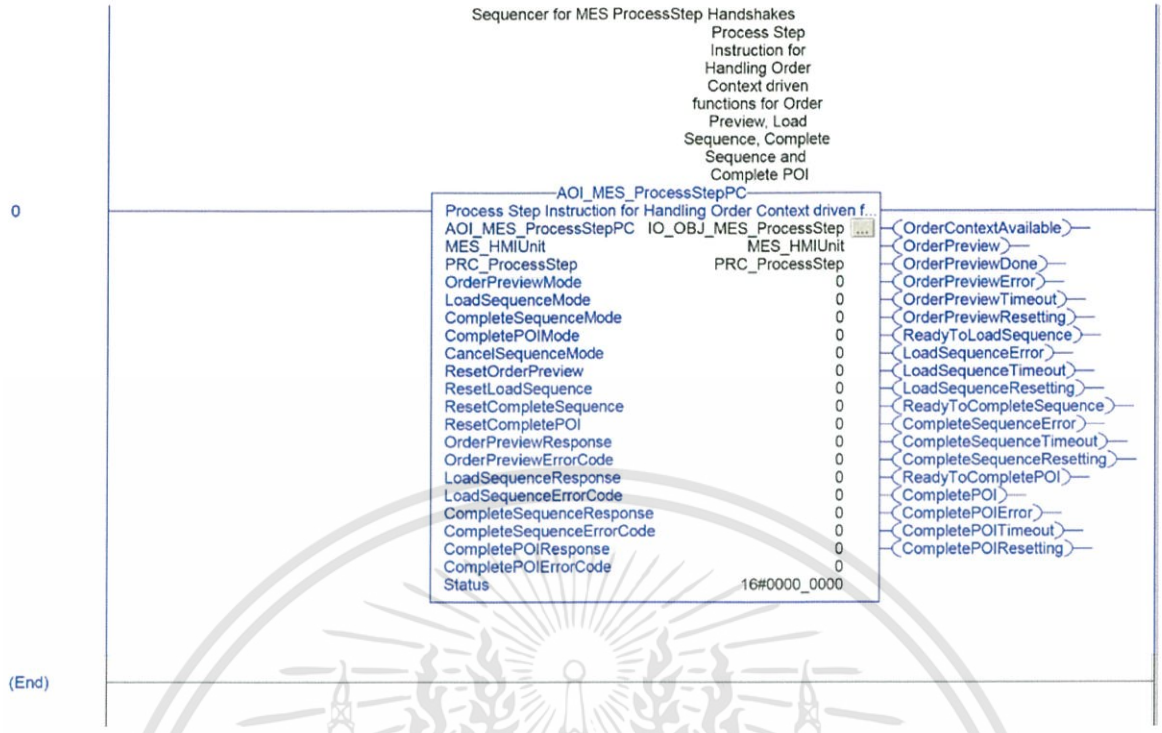
รูปที่ 3.70 ตัวอย่างโปรแกรม BufferConsume

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

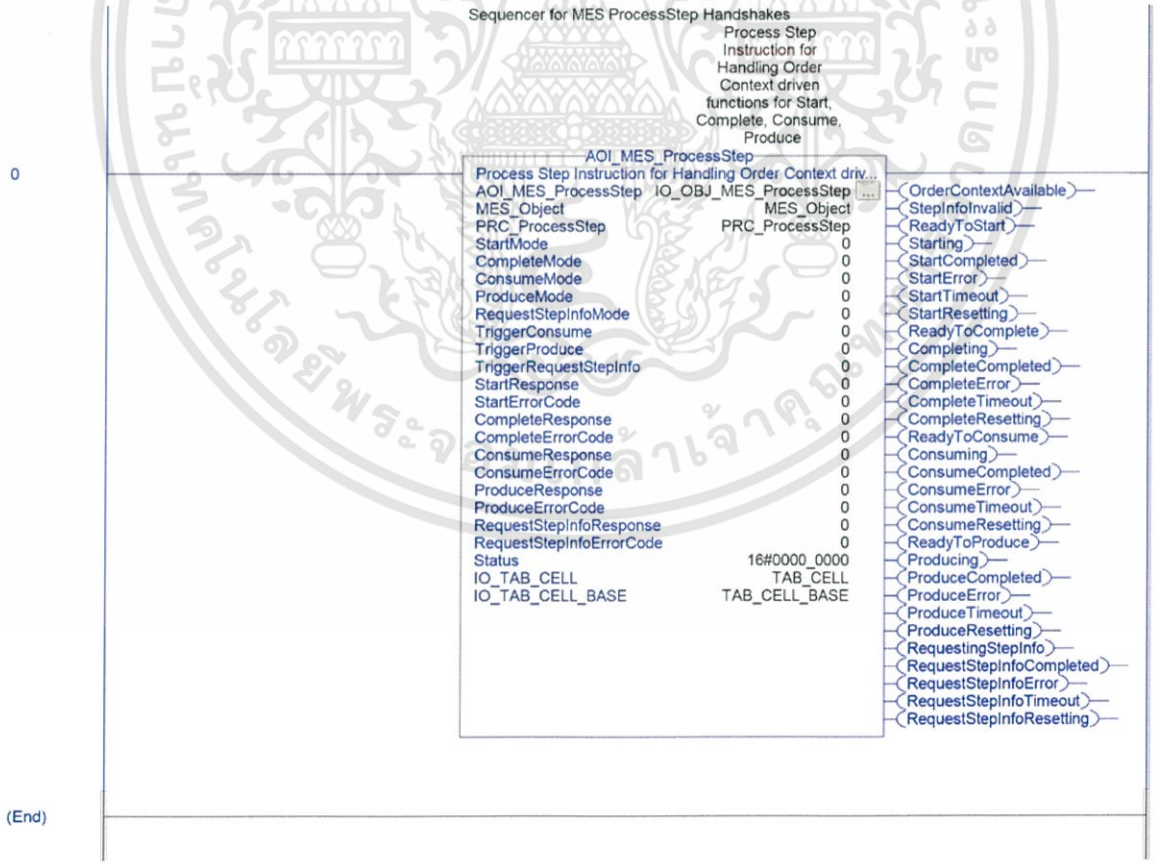


รูปที่ 3.71 ตัวอย่างโปรแกรม BufferProduce

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.72 ตัวอย่างโปรแกรม ProcessStepPC



รูปที่ 3.73 ตัวอย่างโปรแกรม ProcessStep

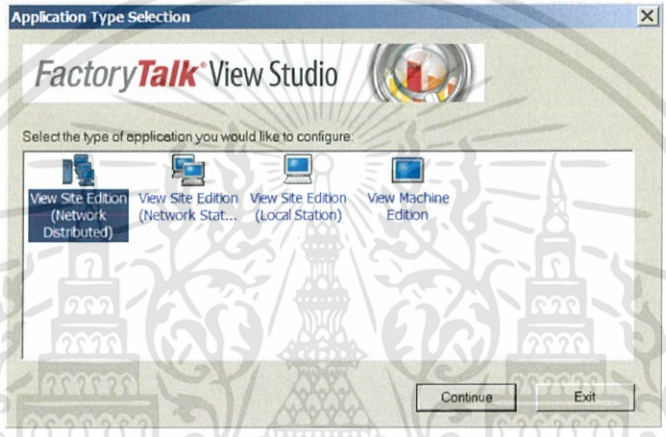
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ขั้นตอนการเขียนโปรแกรม FactoryTalk View Studio

เนื่องจากส่วนแสดงผลของศูนย์กลางรวบรวมข้อมูลสำหรับกระบวนการบรรจุภัณฑ์อาหารสัตว์ที่ได้สร้างขึ้น ได้ทำการเขียนโปรแกรมต่อจากส่วนแสดงผลของกระบวนการผลิตอื่น ๆ ในโรงงาน ดังนั้นในขั้นตอนแรกจะกล่าวถึงการเปิดโปรแกรมที่สร้างไว้แล้ว

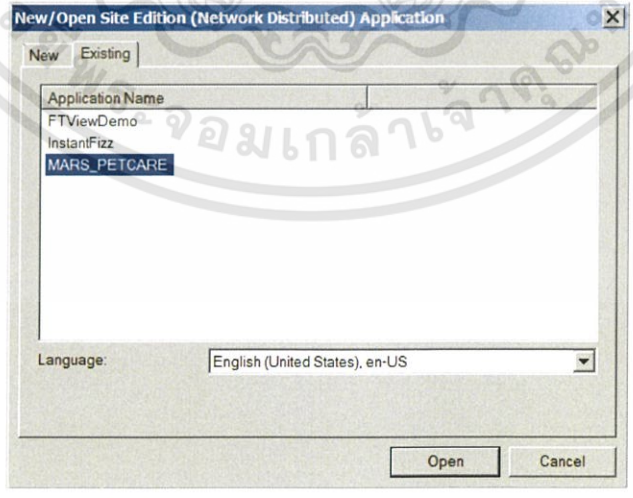
3.6.1 การเปิดโปรแกรม FactoryTalk View Site Edition

1) เปิดโปรแกรม FactoryTalk View Studio เลือก View Site Edition (Network Distributed) ซึ่งเป็นชนิดของแอปพลิเคชันที่มีความสามารถในการทำงานสูงสุดของโปรแกรม แล้วคลิกที่ Continue ดังรูปที่ 3.74



รูปที่ 3.74 ตัวอย่างคอนโทรลเลอร์ที่เปลี่ยนเป็น Emulator

2) เลือกไฟล์โปรแกรมที่ต้องการเปิด แล้วคลิกที่ Open ดังรูปที่ 3.75



รูปที่ 3.75 การเลือกไฟล์โปรแกรมที่ต้องการเปิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 การสร้าง New Display

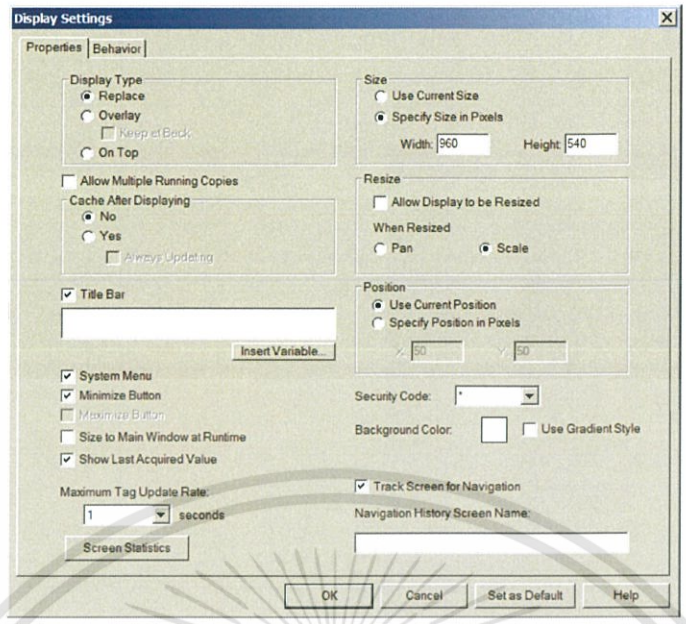
1) คลิกขวาที่ Displays แล้วเลือก New เพื่อเพิ่มหน้าจอที่ใช้วาดกราฟิกเข้าไป ดังรูปที่ 3.76

รูปที่ 3.76 การสร้าง Displays

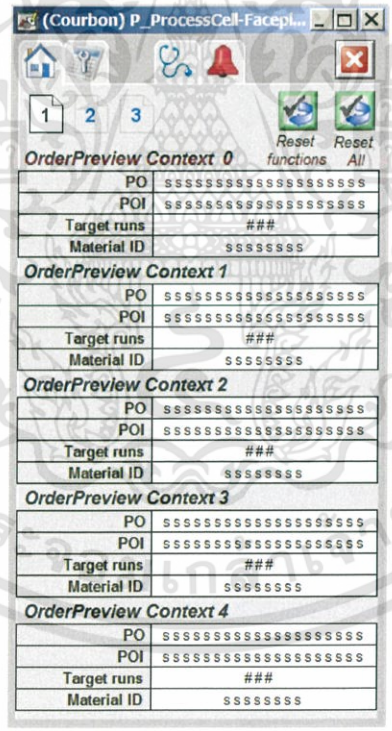
2) คลิกขวาที่หน้าจอ แล้วเลือก Display Settings เพื่อทำการตั้งค่ารายละเอียดต่าง ๆ ของหน้าจอ ดังรูปที่ 3.77 และรูปที่ 3.78 ตัวอย่างของหน้าจอที่ถูกสร้างขึ้นจะแสดงดังรูปที่ 3.79

รูปที่ 3.77 การตั้งค่าหน้าจอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



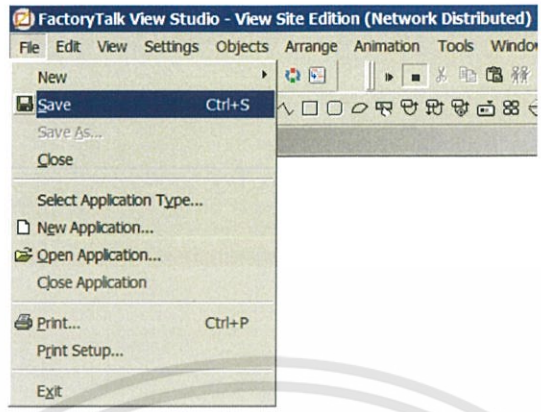
รูปที่ 3.78 การตั้งค่าหน้าจอ (ต่อ)



รูปที่ 3.79 ตัวอย่างหน้าจอที่ถูกสร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) คลิกที่ File เลือก Save แล้วทำการตั้งชื่อหน้าจอ คลิก OK ดังรูปที่ 3.80 และรูปที่ 3.81

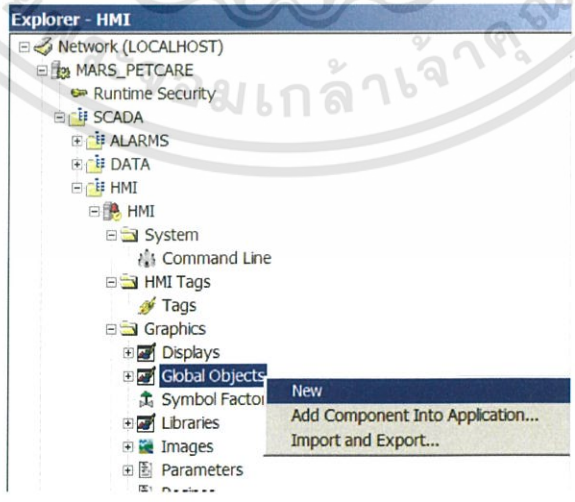


รูปที่ 3.80 การตั้งชื่อหน้าจอ

รูปที่ 3.81 การตั้งชื่อหน้าจอ (ต่อ)

3.6.3 การสร้าง Global Objects

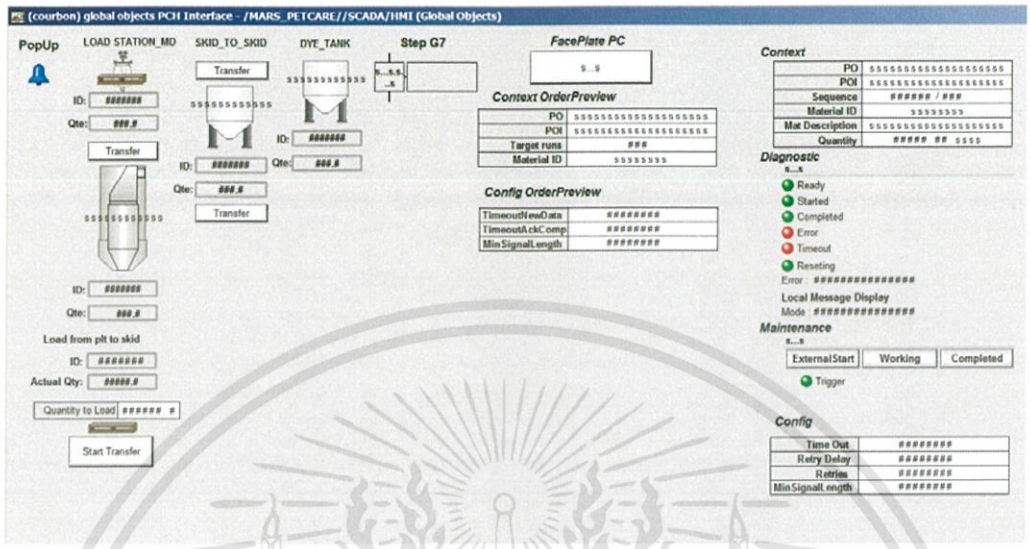
1) คลิกขวาที่ Global Objects แล้วเลือก New เพื่อเพิ่มหน้าจอที่ใช้วาดกราฟิกของอุปกรณ์ต่าง ๆ ที่จะถูกใช้ร่วมกันในโปรแกรม ดังรูปที่ 3.82



รูปที่ 3.82 การสร้าง Global Objects

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

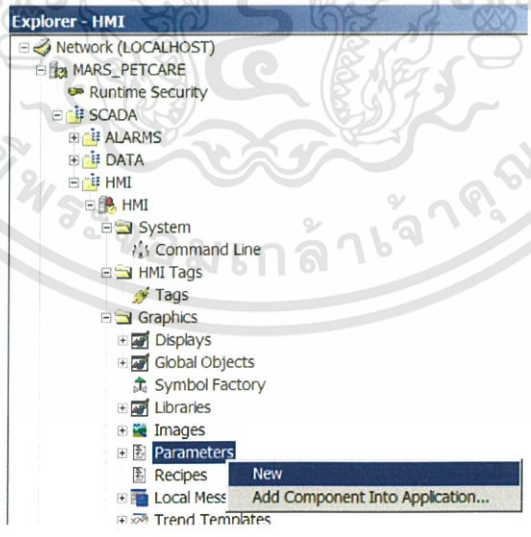
2) ทำการตั้งค่ารายละเอียดต่าง ๆ ของหน้าจอเช่นเดียวกับการสร้าง New Display ตัวอย่างของ Global Objects ที่ถูกสร้างขึ้นจะแสดงดังรูปที่ 3.83



รูปที่ 3.83 ตัวอย่าง Global Objects ที่ถูกนำไปใช้ในโปรแกรมต่าง ๆ

3.6.4 การสร้าง Parameters

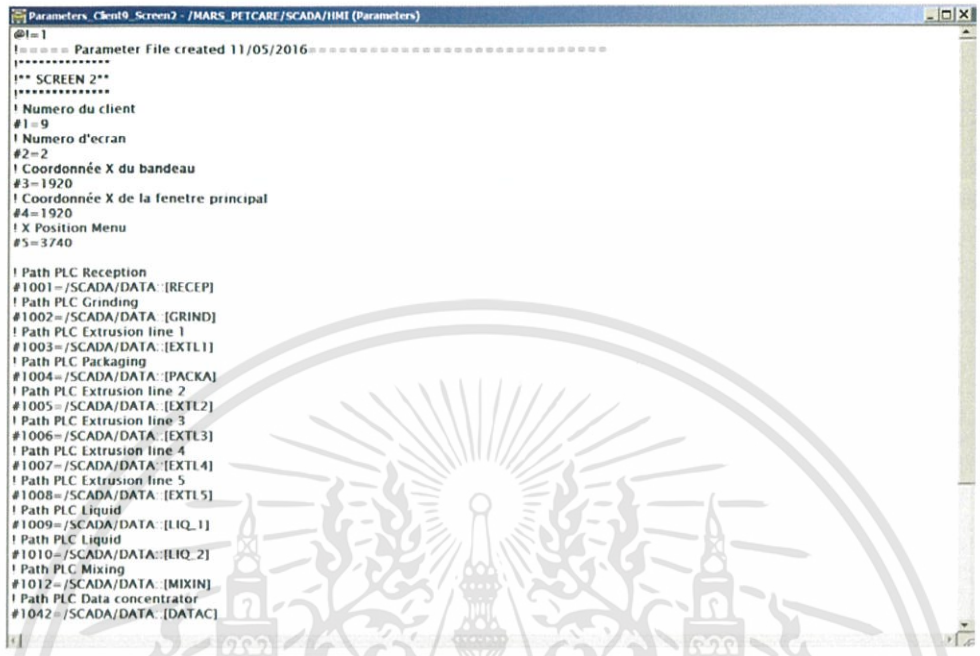
1) คลิกขวาที่ Parameters แล้วเลือก New เพื่อเพิ่มไฟล์สำหรับการสร้างพารามิเตอร์ต่าง ๆ ที่จะถูกนำไปใช้ในโปรแกรม ดังรูปที่ 3.84



รูปที่ 3.84 การสร้างพารามิเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทำการกำหนดพารามิเตอร์ต่าง ๆ ที่จะถูกนำไปใช้ในโปรแกรม ดังตัวอย่างของพารามิเตอร์ที่ถูกสร้างขึ้นในรูปที่ 3.85



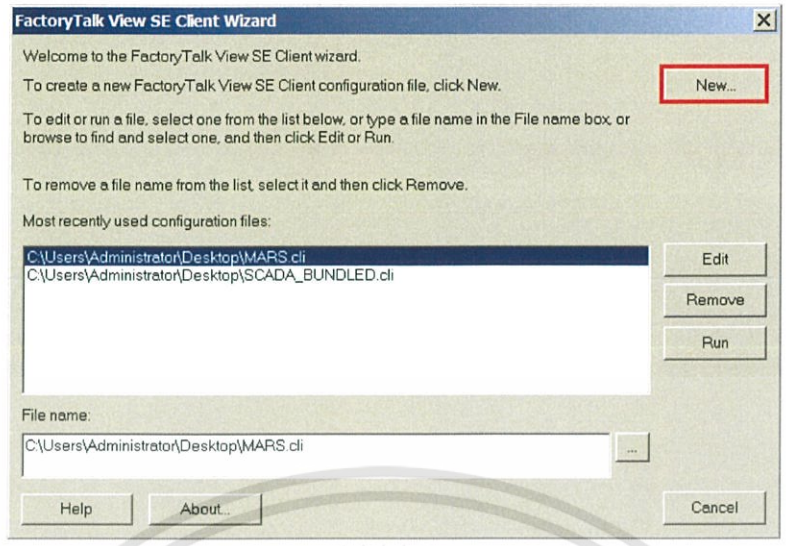
รูปที่ 3.85 ตัวอย่างพารามิเตอร์ ที่ถูกนำไปใช้ในโปรแกรมต่าง ๆ

3.6.5 การสร้าง Client หรือ สร้าง Runtime

1) คลิกที่ Launch SE Client ดังรูปที่ 3.86 จะได้หน้าต่างดังรูปที่ 3.87 คลิกที่ New เพื่อสร้าง Runtime

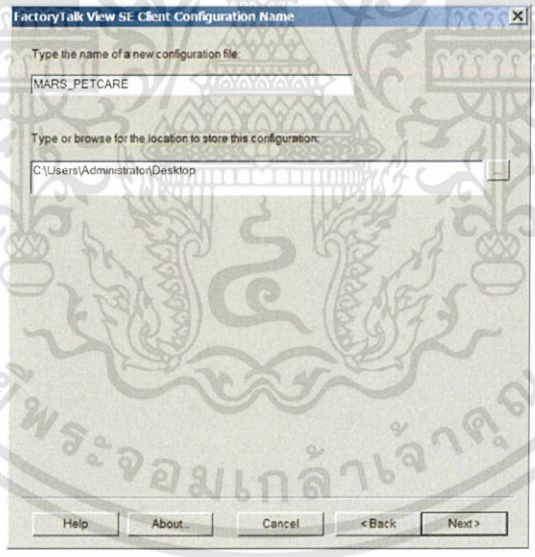
รูปที่ 3.86 การสร้าง Runtime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.87 การสร้าง Runtime (ต่อ)

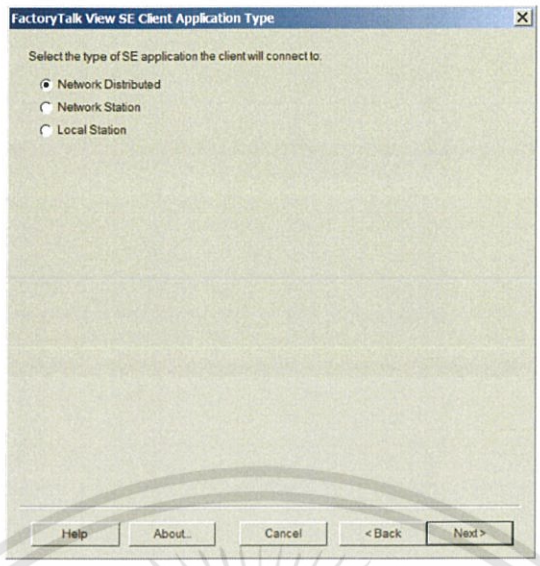
2) ตั้งชื่อ Runtime และเลือก Location สำหรับเก็บไฟล์ Runtime แล้วคลิกที่ Next ดังรูปที่ 3.88



รูปที่ 3.88 การตั้งชื่อ Runtime

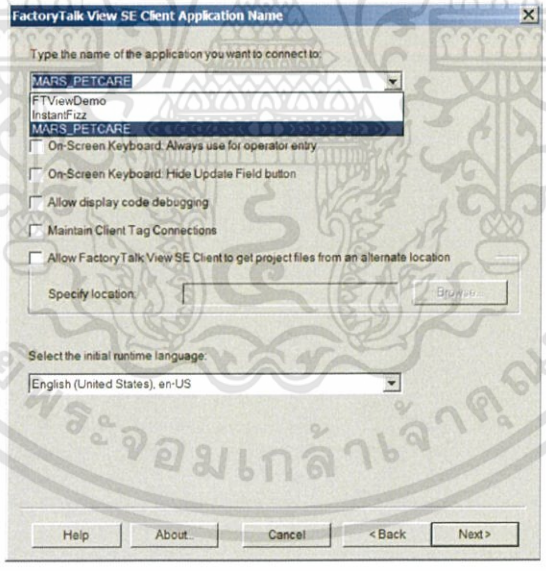
3) เลือกชนิดของ SE client application ที่ต้องการเชื่อมต่อ ให้เลือกเป็น Network Distributed ซึ่งเป็นชนิดของแอปพลิเคชันที่มีความสามารถในการทำงานสูงสุดของโปรแกรมเหมาะสำหรับสกาดาหลักที่ใช้ภายในโรงงาน แล้วคลิกที่ Next ดังรูปที่ 3.89

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.89 การเลือกชนิดของ SE client application

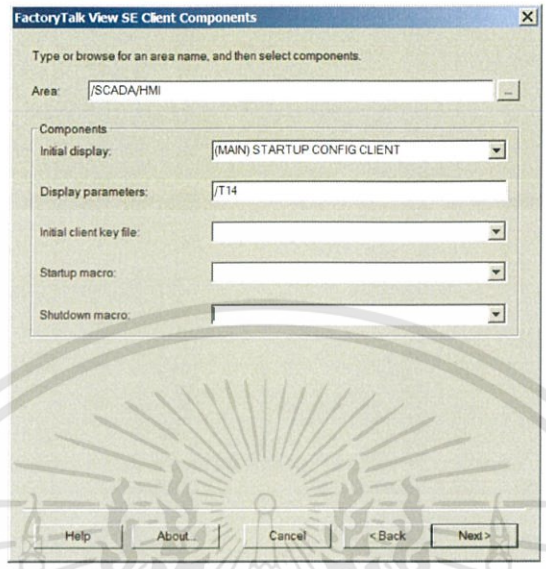
4) เลือกโปรเจคที่ต้องการจะสร้าง Runtime และภาษาที่ใช้ในการเริ่มต้น Runtime แล้วคลิกที่ Next ดังรูปที่ 3.90



รูปที่ 3.90 การเลือกโปรเจคที่จะสร้าง Runtime

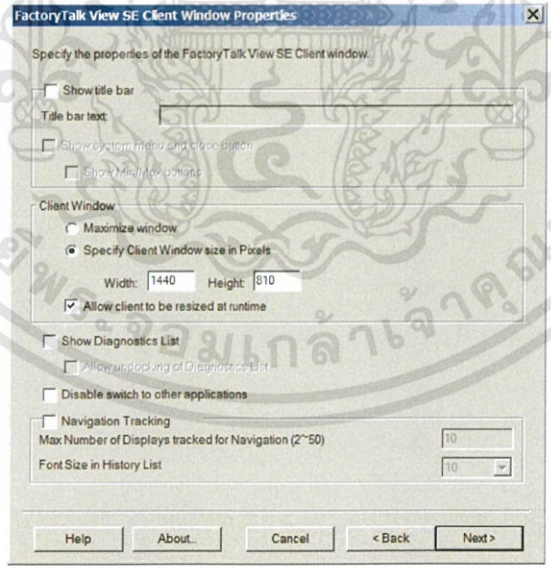
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) กำหนดหน้าแรกของ Runtime และพารามิเตอร์ที่ใช้ในหน้านั้น แล้วคลิกที่ Next ดังรูปที่ 3.91



รูปที่ 3.91 การกำหนดหน้าแรกของ Runtime

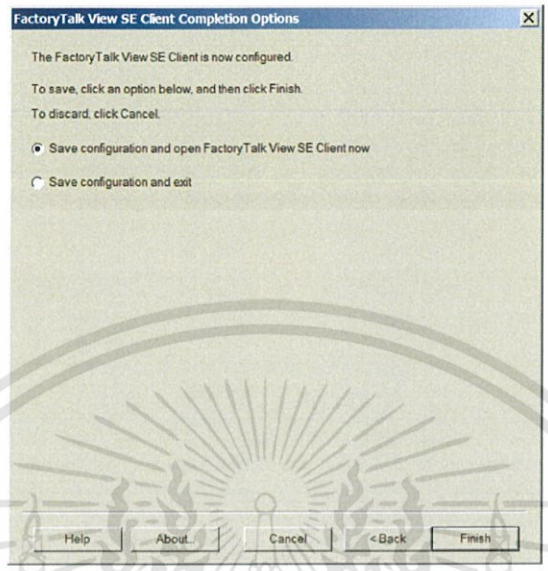
6) กำหนดขนาดของหน้าจอ Runtime แล้วคลิกที่ Next ดังรูปที่ 3.92



รูปที่ 3.92 การกำหนดขนาดของหน้าจอ Runtime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) เลือกรูปแบบการ Save แล้วคลิกที่ Finish ดังรูปที่ 3.93 ตัวอย่างของ Runtime ที่ถูกสร้างขึ้นจะแสดงดังรูปที่ 3.94



รูปที่ 3.93 การเลือกรูปแบบการ Save Runtime

รูปที่ 3.94 ตัวอย่างหน้าจอ Runtime

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดสอบศูนย์กลางรวบรวมข้อมูล

4.1 กล่าวนำ

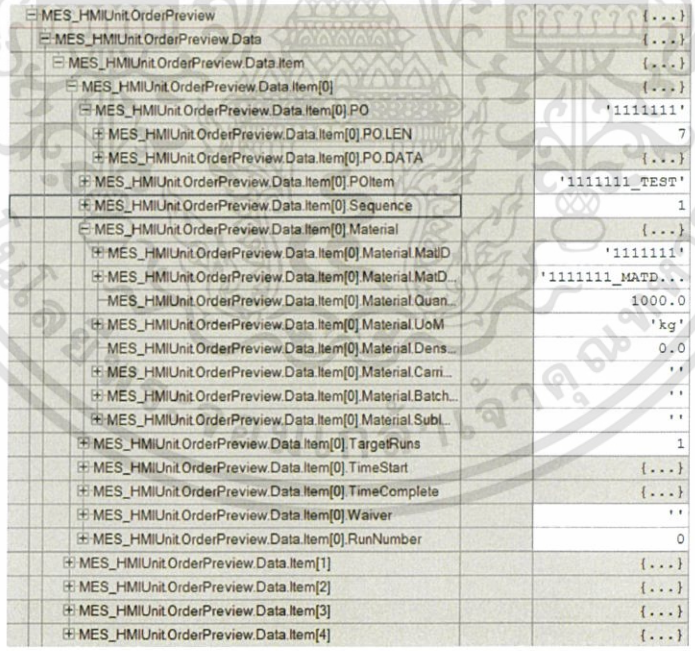
ในบทนี้จะกล่าวถึงผลการทดสอบของศูนย์กลางรวบรวมข้อมูลในแต่ละขั้นตอนกับ MOM โดยจะแบ่งการทดสอบออกเป็น การทดสอบการทำงานของโปรแกรมและการทดสอบการทำงานของส่วนแสดงผลข้อมูลผ่านระบบสกาดา

4.2 ผลการทดสอบการทำงานของโปรแกรม

ในการทดสอบการทำงานของโปรแกรม จะทำการทดสอบทีละขั้นตอนตามที่ได้ออกแบบฟังก์ชั่นการทำงานและเขียนโปรแกรม PLC ไว้ และค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน PO จะเป็นค่าที่ถูกกำหนดขึ้นสำหรับการทดสอบเท่านั้น ไม่ใช่ค่าที่แท้จริงของกระบวนการ

4.2.1 Order Preview

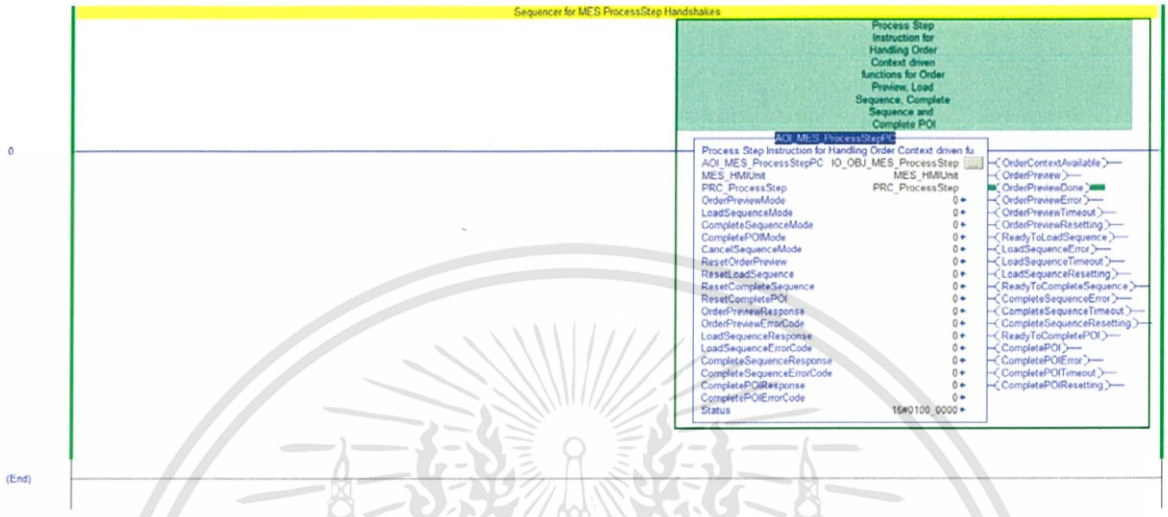
ในขั้นตอน Order Preview จะทำการทดสอบโดยส่งบิต Ready ไปให้กับ MOM แล้วทำการสังเกตค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMI Unit.OrderPreview.Data.Item) ดังรูปที่ 4.1



รูปที่ 4.1 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Order Preview

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.1 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMIUnit.Order Preview.Data.Item) จะถูก MOM ดาวน์โหลดข้อมูลของ PO มาให้และหลังจากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Done กลับไปเพื่อแสดงว่าจบการทำงานในขั้นตอน Order Preview แล้วดังรูปที่ 4.2



รูปที่ 4.2 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Done ในขั้นตอน Order Preview

4.2.2 Load Sequence

ในขั้นตอน Load Sequence จะทำการทดสอบโดยเลือกคำสั่งที่ต้องการใช้งานจาก 5 PO (ในการทดลองเลือกคำสั่งที่ 1) แล้วทำการสังเกตค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMI Unit.Context) ดังรูปที่ 4.3

MES_HMIUnit	Public	{...}
MES_HMIUnit.Context		{...}
MES_HMIUnit.Context.PO		'1111111'
MES_HMIUnit.Context.PO.LEN		7
MES_HMIUnit.Context.PO.DATA		{...}
MES_HMIUnit.Context.POItem		'1111111_TEST'
MES_HMIUnit.Context.Sequence		1
MES_HMIUnit.Context.Material		{...}
MES_HMIUnit.Context.Material.MatID		'1111111'
MES_HMIUnit.Context.Material.MatDesc		'1111111_MATD...
MES_HMIUnit.Context.Material.Quantity		1000.0
MES_HMIUnit.Context.Material.UoM		'kg'
MES_HMIUnit.Context.Material.Dehsity		0.0
MES_HMIUnit.Context.Material.CarrierID		''
MES_HMIUnit.Context.Material.BatchID		''
MES_HMIUnit.Context.Material.SublotID		''
MES_HMIUnit.Context.TargetRuns		1
MES_HMIUnit.Context.TimeStart		{...}
MES_HMIUnit.Context.TimeComplete		{...}
MES_HMIUnit.Context.Waiver		''
MES_HMIUnit.Context.RunNumber		0
MES_HMIUnit.OrderPreview		{...}
MES_HMIUnit.PO_Status		4
MES_HMIUnit.LoadSequence		{...}
MES_HMIUnit.CancelSequence		{...}
MES_HMIUnit.CompletePOI		{...}
MES_HMIUnit.CompleteSequence		{...}

รูปที่ 4.3 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Load Sequence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.3 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMIUnit.Order Preview.Data.Item) ถูกคัดลอกมาไว้ใน array (MES_HMIUnit.Context) และหลังจากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.4 และ MOM จะส่งบิต Working และ Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Load Sequence แล้วดังรูปที่ 4.5 และรูปที่ 4.6 ตามลำดับ

⊟ MES_HMIUnit.LoadSequence		{...}
MES_HMIUnit.LoadSequence.Trigger		1
MES_HMIUnit.LoadSequence.Working		0
MES_HMIUnit.LoadSequence.Completed		0
⊞ MES_HMIUnit.LoadSequence.Error		0
⊞ MES_HMIUnit.LoadSequence.Mode		0
⊞ MES_HMIUnit.LoadSequence.Response		0
⊞ MES_HMIUnit.LoadSequence.OwnerMode		0
⊞ MES_HMIUnit.LoadSequence.OperationMode		0
⊞ MES_HMIUnit.LoadSequence.Config		{...}
⊞ MES_HMIUnit.LoadSequence.Counter		{...}

รูปที่ 4.4 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Load Sequence

⊟ MES_HMIUnit.LoadSequence		{...}
MES_HMIUnit.LoadSequence.Trigger		0
MES_HMIUnit.LoadSequence.Working		1
MES_HMIUnit.LoadSequence.Completed		0
⊞ MES_HMIUnit.LoadSequence.Error		0
⊞ MES_HMIUnit.LoadSequence.Mode		0
⊞ MES_HMIUnit.LoadSequence.Response		0
⊞ MES_HMIUnit.LoadSequence.OwnerMode		0
⊞ MES_HMIUnit.LoadSequence.OperationMode		0
⊞ MES_HMIUnit.LoadSequence.Config		{...}
⊞ MES_HMIUnit.LoadSequence.Counter		{...}

รูปที่ 4.5 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Load Sequence

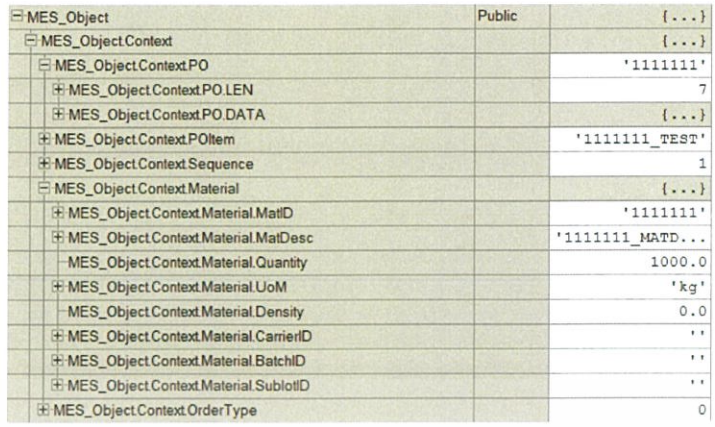
⊟ MES_HMIUnit.LoadSequence		{...}
MES_HMIUnit.LoadSequence.Trigger		0
MES_HMIUnit.LoadSequence.Working		0
MES_HMIUnit.LoadSequence.Completed		1
⊞ MES_HMIUnit.LoadSequence.Error		0
⊞ MES_HMIUnit.LoadSequence.Mode		0
⊞ MES_HMIUnit.LoadSequence.Response		0
⊞ MES_HMIUnit.LoadSequence.OwnerMode		0
⊞ MES_HMIUnit.LoadSequence.OperationMode		0
⊞ MES_HMIUnit.LoadSequence.Config		{...}
⊞ MES_HMIUnit.LoadSequence.Counter		{...}

รูปที่ 4.6 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Load Sequence

4.2.3 Start Unit PO

จากรูปที่ 4.7 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMIUnit.Context) ถูกคัดลอกมาไว้ใน array (MES_Object.Context)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



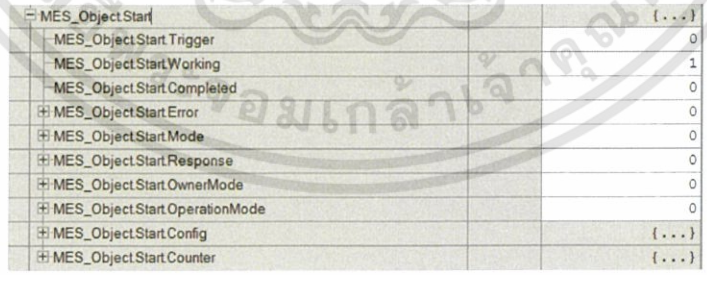
MES_Object	Public	{...}
MES_Object.Context		{...}
MES_Object.Context.PO		'1111111'
MES_Object.Context.PO.LEN		7
MES_Object.Context.PO.DATA		{...}
MES_Object.Context.POItem		'1111111_TEST'
MES_Object.Context.Sequence		1
MES_Object.Context.Material		{...}
MES_Object.Context.Material.MatID		'1111111'
MES_Object.Context.Material.MatDesc		'1111111_MATD...
MES_Object.Context.Material.Quantity		1000.0
MES_Object.Context.Material.UoM		'kg'
MES_Object.Context.Material.Density		0.0
MES_Object.Context.Material.CarrierID		''
MES_Object.Context.Material.BatchID		''
MES_Object.Context.Material.SublotID		''
MES_Object.Context.OrderType		0

รูปที่ 4.7 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Start Unit PO

หลังจากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.8 และ MOM จะส่งบิต Working และ Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Start Unit PO แล้วดังรูปที่ 4.9 และรูปที่ 4.10, 4.11 ตามลำดับ

MES_Object.Start		{...}
MES_Object.Start.Trigger		1
MES_Object.Start.Working		0
MES_Object.Start.Completed		0
MES_Object.Start.Error		0
MES_Object.Start.Mode		0
MES_Object.Start.Response		0
MES_Object.Start.OwnerMode		0
MES_Object.Start.OperationMode		0
MES_Object.Start.Config		{...}
MES_Object.Start.Counter		{...}

รูปที่ 4.8 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Start Unit PO



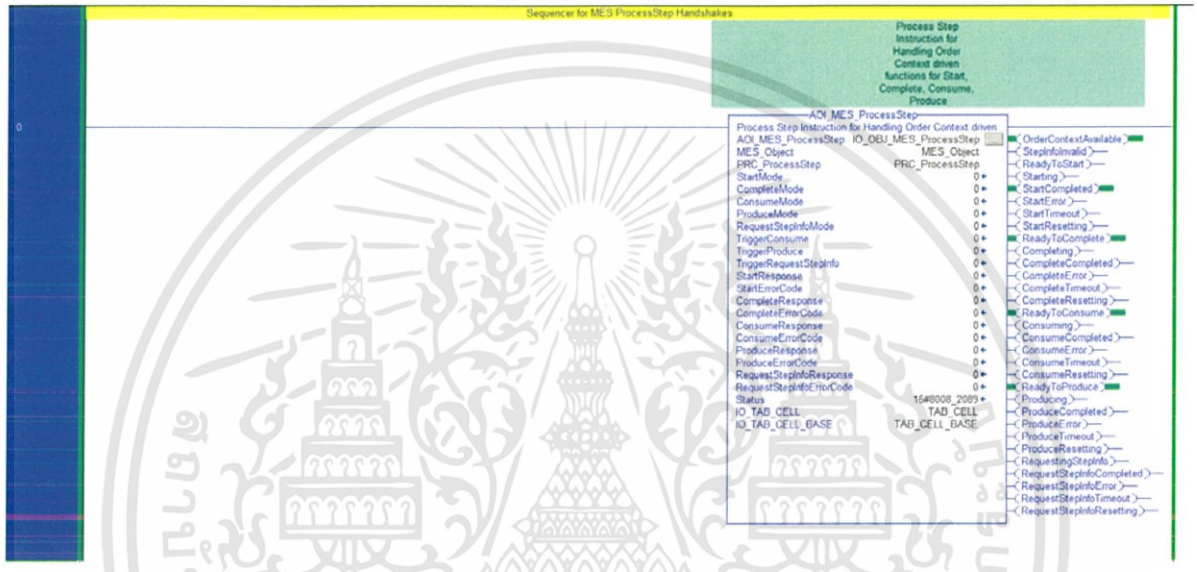
MES_Object.Start		{...}
MES_Object.Start.Trigger		0
MES_Object.Start.Working		1
MES_Object.Start.Completed		0
MES_Object.Start.Error		0
MES_Object.Start.Mode		0
MES_Object.Start.Response		0
MES_Object.Start.OwnerMode		0
MES_Object.Start.OperationMode		0
MES_Object.Start.Config		{...}
MES_Object.Start.Counter		{...}

รูปที่ 4.9 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Start Unit PO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MES_Object.Start		{...}
MES_Object.Start.Trigger		0
MES_Object.Start.Working		0
MES_Object.Start.Completed		1
MES_Object.Start.Error		0
MES_Object.Start.Mode		0
MES_Object.Start.Response		0
MES_Object.Start.OwnerMode		0
MES_Object.Start.OperationMode		0
MES_Object.Start.Config		{...}
MES_Object.Start.Counter		{...}

รูปที่ 4.10 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Start Unit PO



รูปที่ 4.11 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Start Unit PO (ต่อ)

4.2.4 Consume

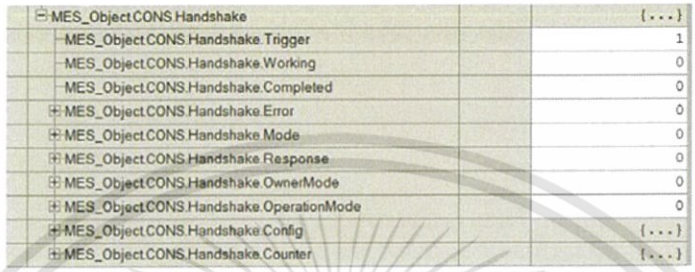
ในขั้นตอน Consume จะทำการทดสอบโดยกรอกรายละเอียดของวัสดุที่ถูกใช้ไปในกระบวนการให้กับศูนย์กลางรวบรวมข้อมูลแล้วทำการสังเกตค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_Object.CONS) ดังรูปที่ 4.12

MES_Object.CONS		{...}
MES_Object.CONS.Material		{...}
MES_Object.CONS.Material.MatID		'00000000'
MES_Object.CONS.Material.MatDesc		'00000'
MES_Object.CONS.Material.Quantity		1000.0
MES_Object.CONS.Material.UoM		'KG'
MES_Object.CONS.Material.Density		10.0
MES_Object.CONS.Material.CarrierID		'00000'
MES_Object.CONS.Material.BatchID		'00000'
MES_Object.CONS.Material.SublotID		'00000'

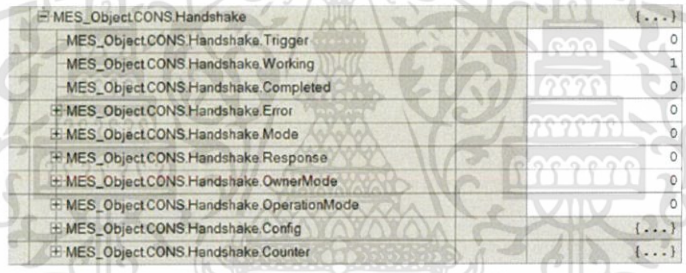
รูปที่ 4.12 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Consume

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

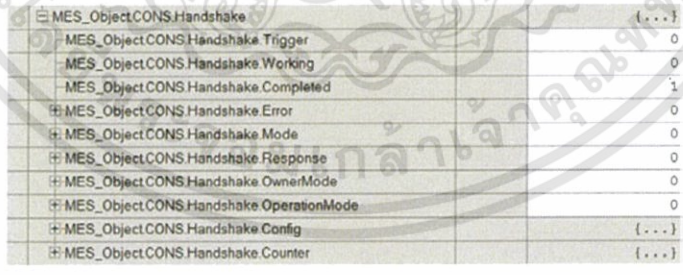
จากรูปที่ 4.12 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_Object.CONS) ได้เก็บข้อมูลต่าง ๆ ของวัสดุที่ถูกใช้ไปในกระบวนการไว้แล้ว จากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.13 และ MOM จะส่งบิต Working กลับมาให้ดังรูปที่ 4.14 และจะทำการอัพเดทข้อมูลต่าง ๆ ของวัสดุที่ได้ถูกใช้ไปในกระบวนการ แล้วจะส่งบิต Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Consume แล้วดังรูปที่ 4.15



รูปที่ 4.13 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Consume



รูปที่ 4.14 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Consume



รูปที่ 4.15 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Consume

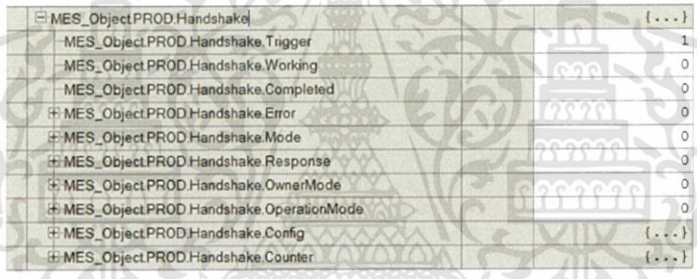
4.2.5 Produce

ในขั้นตอน Produce จะทำการทดสอบโดยกรอกรายละเอียดของวัสดุที่ผลิตได้ในกระบวนการให้กับศูนย์กลางรวบรวมข้อมูลแล้วทำการสังเกตค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_Object.PROD) ดังรูปที่ 4.16

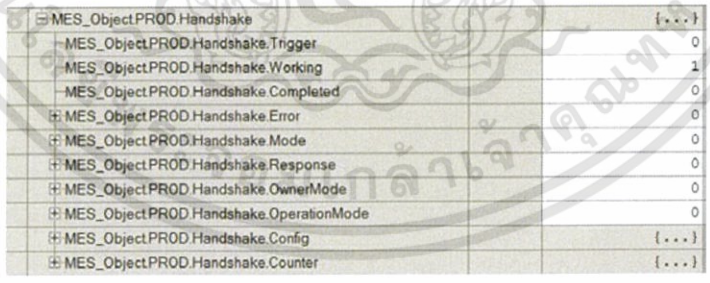
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.16 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Produce

จากรูปที่ 4.16 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_Object.PROD) ได้เก็บข้อมูลต่าง ๆ ของวัสดุที่ผลิตได้ในกระบวนการไว้แล้ว จากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.17 และ MOM จะส่งบิต Working กลับมาให้ดังรูปที่ 4.18 และจะทำการอัพเดทข้อมูลต่าง ๆ ของวัสดุที่ผลิตได้ในกระบวนการ แล้วจะส่งบิต Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Produce แล้วดังรูปที่ 4.19

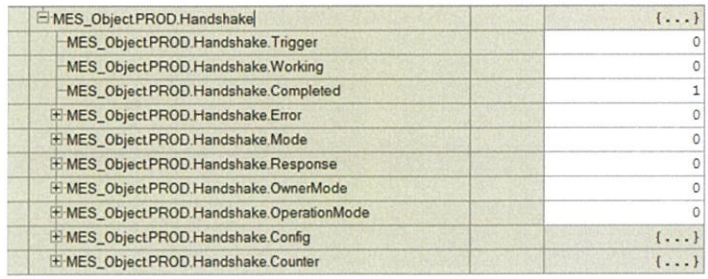


รูปที่ 4.17 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Produce



รูปที่ 4.18 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Produce

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

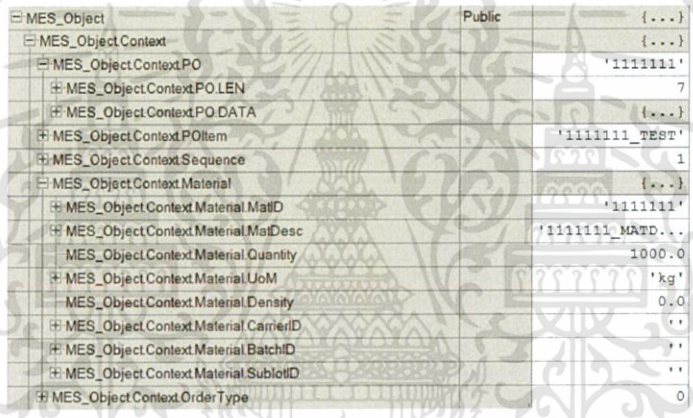


MES_Object.PROD.Handshake		{...}
MES_Object.PROD.Handshake.Trigger		0
MES_Object.PROD.Handshake.Working		0
MES_Object.PROD.Handshake.Completed		1
MES_Object.PROD.Handshake.Error		0
MES_Object.PROD.Handshake.Mode		0
MES_Object.PROD.Handshake.Response		0
MES_Object.PROD.Handshake.OwnerMode		0
MES_Object.PROD.Handshake.OperationMode		0
MES_Object.PROD.Handshake.Config		{...}
MES_Object.PROD.Handshake.Counter		{...}

รูปที่ 4.19 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Produce

4.2.6 Complete Unit PO

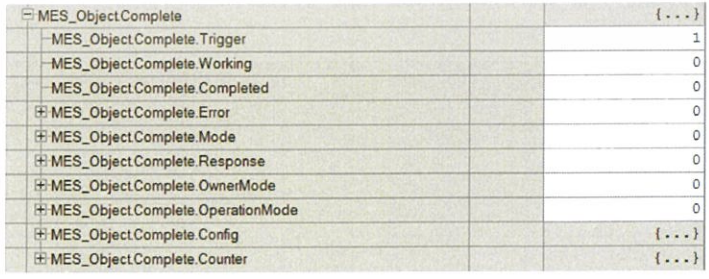
จากรูปที่ 4.20 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMIUnit.Context) ถูกคัดลอกมาไว้ใน array (MES_Object.Context)



MES_Object	Public	{...}
MES_Object.Context		{...}
MES_Object.Context.PO		'1111111'
MES_Object.Context.PO.LEN		7
MES_Object.Context.PO.DATA		{...}
MES_Object.Context.POItem		'1111111_TEST'
MES_Object.Context.Sequence		1
MES_Object.Context.Material		{...}
MES_Object.Context.Material.MatID		'1111111'
MES_Object.Context.Material.MatDesc		'1111111_MATD...
MES_Object.Context.Material.Quantity		1000.0
MES_Object.Context.Material.UoM		'kg'
MES_Object.Context.Material.Density		0.0
MES_Object.Context.Material.CarrierID		''
MES_Object.Context.Material.BatchID		''
MES_Object.Context.Material.SublotID		''
MES_Object.Context.OrderType		0

รูปที่ 4.20 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Complete Unit PO

หลังจากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.21 และ MOM จะส่งบิต Working และ Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Complete Unit PO แล้วดังรูปที่ 4.22 และรูปที่ 4.23 ตามลำดับ



MES_Object.Complete		{...}
MES_Object.Complete.Trigger		1
MES_Object.Complete.Working		0
MES_Object.Complete.Completed		0
MES_Object.Complete.Error		0
MES_Object.Complete.Mode		0
MES_Object.Complete.Response		0
MES_Object.Complete.OwnerMode		0
MES_Object.Complete.OperationMode		0
MES_Object.Complete.Config		{...}
MES_Object.Complete.Counter		{...}

รูปที่ 4.21 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Complete Unit PO

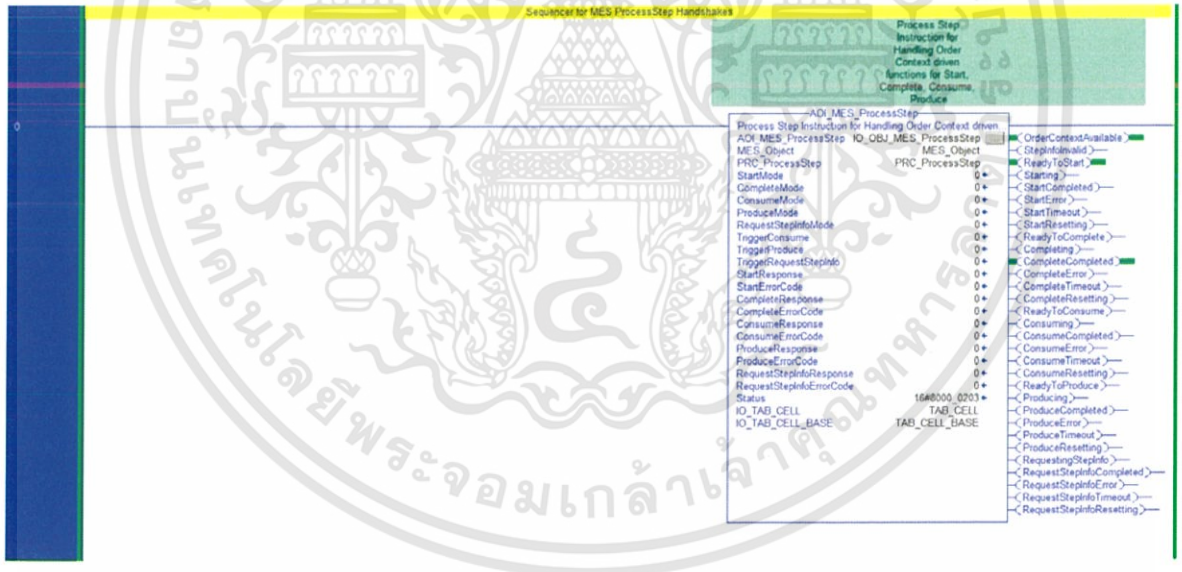
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MES_Object.Complete		{...}
MES_Object.Complete.Trigger		0
MES_Object.Complete.Working		1
MES_Object.Complete.Completed		0
MES_Object.Complete.Error		0
MES_Object.Complete.Mode		0
MES_Object.Complete.Response		0
MES_Object.Complete.OwnerMode		0
MES_Object.Complete.OperationMode		0
MES_Object.Complete.Config		{...}
MES_Object.Complete.Counter		{...}

รูปที่ 4.22 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Complete Unit PO

MES_Object.Complete		{...}
MES_Object.Complete.Trigger		0
MES_Object.Complete.Working		0
MES_Object.Complete.Completed		1
MES_Object.Complete.Error		0
MES_Object.Complete.Mode		0
MES_Object.Complete.Response		0
MES_Object.Complete.OwnerMode		0
MES_Object.Complete.OperationMode		0
MES_Object.Complete.Config		{...}
MES_Object.Complete.Counter		{...}

รูปที่ 4.23 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete Unit PO

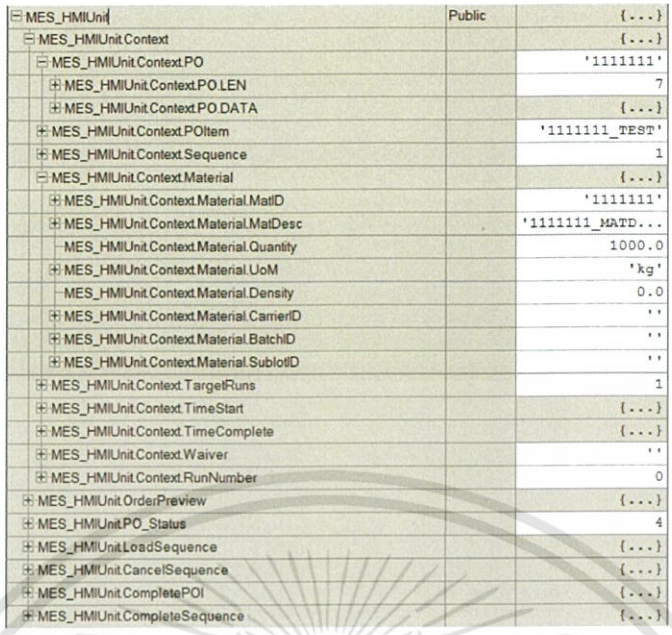


รูปที่ 4.24 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete Unit PO (ต่อ)

4.2.7 Complete Sequence

ในขั้นตอน Complete Sequence จะทำการทดสอบโดยเลือกคำสั่งที่ต้องการใช้งานจาก 5 PO แล้วทำการสังเกตค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMI Unit.Context) ดังรูปที่ 4.25

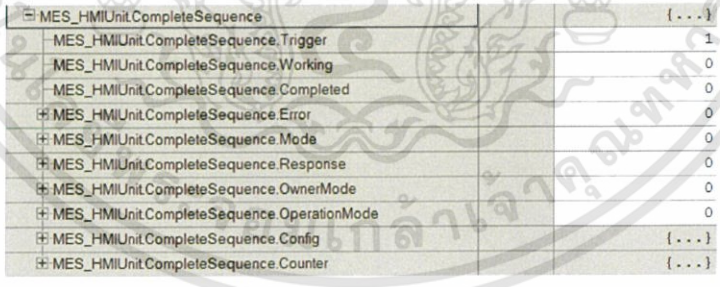
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MES_HMIUnit	Public	{...}
MES_HMIUnit.Context		{...}
MES_HMIUnit.Context.PO		'1111111'
MES_HMIUnit.Context.PO.LEN		7
MES_HMIUnit.Context.PO.DATA		{...}
MES_HMIUnit.Context.POItem		'1111111_TEST'
MES_HMIUnit.Context.Sequence		1
MES_HMIUnit.Context.Material		{...}
MES_HMIUnit.Context.Material.MatID		'1111111'
MES_HMIUnit.Context.Material.MatDesc		'1111111_MATD...
MES_HMIUnit.Context.Material.Quantity		1000.0
MES_HMIUnit.Context.Material.UoM		'kg'
MES_HMIUnit.Context.Material.Density		0.0
MES_HMIUnit.Context.Material.CarrierID		''
MES_HMIUnit.Context.Material.BatchID		''
MES_HMIUnit.Context.Material.SubLotID		''
MES_HMIUnit.Context.TargetRuns		1
MES_HMIUnit.Context.TimeStart		{...}
MES_HMIUnit.Context.TimeComplete		{...}
MES_HMIUnit.Context.Waiver		''
MES_HMIUnit.Context.RunNumber		0
MES_HMIUnit.OrderPreview		{...}
MES_HMIUnit.PO_Status		4
MES_HMIUnit.LoadSequence		{...}
MES_HMIUnit.CancelSequence		{...}
MES_HMIUnit.CompletePOI		{...}
MES_HMIUnit.CompleteSequence		{...}

รูปที่ 4.25 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Complete Sequence

จากรูปที่ 4.25 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMIUnit.Order Preview.Data.Item) ถูกคัดลอกมาไว้ใน array (MES_HMIUnit.Context) และหลังจากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.26 และ MOM จะส่งบิต Working และ Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Complete Sequence แล้วดังรูปที่ 4.27 และรูปที่ 4.28 ตามลำดับ



MES_HMIUnit.CompleteSequence		{...}
MES_HMIUnit.CompleteSequence.Trigger		1
MES_HMIUnit.CompleteSequence.Working		0
MES_HMIUnit.CompleteSequence.Completed		0
MES_HMIUnit.CompleteSequence.Error		0
MES_HMIUnit.CompleteSequence.Mode		0
MES_HMIUnit.CompleteSequence.Response		0
MES_HMIUnit.CompleteSequence.OwnerMode		0
MES_HMIUnit.CompleteSequence.OperationMode		0
MES_HMIUnit.CompleteSequence.Config		{...}
MES_HMIUnit.CompleteSequence.Counter		{...}

รูปที่ 4.26 ผลการทดสอบศูนย์กลางรวบรวมข้อมูลส่งบิต Trigger ในขั้นตอน Complete Sequence

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MES_HMIUnit.CompleteSequence		{...}
MES_HMIUnit.CompleteSequence.Trigger		0
MES_HMIUnit.CompleteSequence.Working		1
MES_HMIUnit.CompleteSequence.Completed		0
MES_HMIUnit.CompleteSequence.Error		0
MES_HMIUnit.CompleteSequence.Mode		0
MES_HMIUnit.CompleteSequence.Response		0
MES_HMIUnit.CompleteSequence.OwnerMode		0
MES_HMIUnit.CompleteSequence.OperationMode		0
MES_HMIUnit.CompleteSequence.Config		{...}
MES_HMIUnit.CompleteSequence.Counter		{...}

รูปที่ 4.27 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Complete Sequence

MES_HMIUnit.CompleteSequence		{...}
MES_HMIUnit.CompleteSequence.Trigger		0
MES_HMIUnit.CompleteSequence.Working		0
MES_HMIUnit.CompleteSequence.Completed		1
MES_HMIUnit.CompleteSequence.Error		0
MES_HMIUnit.CompleteSequence.Mode		0
MES_HMIUnit.CompleteSequence.Response		0
MES_HMIUnit.CompleteSequence.OwnerMode		0
MES_HMIUnit.CompleteSequence.OperationMode		0
MES_HMIUnit.CompleteSequence.Config		{...}
MES_HMIUnit.CompleteSequence.Counter		{...}

รูปที่ 4.28 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete Sequence

4.2.8 Complete POI

ในขั้นตอน Complete POI จะทำการทดสอบโดยเลือก order ที่ต้องการใช้งานจาก 5 PO แล้วทำการสังเกตค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (HMI Unit.CompletePOI.Data) ดังรูปที่ 4.29

MES_HMIUnit.CompletePOI		{...}
MES_HMIUnit.CompletePOI.Data		{...}
MES_HMIUnit.CompletePOI.Data.PO		'1111111'
MES_HMIUnit.CompletePOI.Data.POItem		'1111111_TEST'
MES_HMIUnit.CompletePOI.Data.Sequence		1
MES_HMIUnit.CompletePOI.Data.Material		{...}
MES_HMIUnit.CompletePOI.Data.TargetRuns		1
MES_HMIUnit.CompletePOI.Data.TimeStart		{...}
MES_HMIUnit.CompletePOI.Data.TimeComplete		{...}
MES_HMIUnit.CompletePOI.Data.Waiver		''
MES_HMIUnit.CompletePOI.Data.RunNumber		0

รูปที่ 4.29 ผลการทดสอบการทำงานของโปรแกรมในขั้นตอน Complete POI

จากรูปที่ 4.29 จะเห็นได้ว่าค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน array (MES_HMIUnit.Order Preview.Data.Item) ถูกคัดลอกมาไว้ใน array (MES_HMIUnit.Context) และหลังจากนั้นศูนย์กลางรวบรวมข้อมูลจะส่งบิต Trigger ให้กับ MOM ดังรูปที่ 4.30 และ MOM จะส่งบิต Working และ Complete กลับมาให้ เพื่อแสดงว่าจบการทำงานในขั้นตอน Complete POI แล้วดังรูปที่ 4.31 และรูปที่ 4.32 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MES_HMIUnit.CompletePOI.Handshake	{...}
MES_HMIUnit.CompletePOI.Handshake.Trigger	1
MES_HMIUnit.CompletePOI.Handshake.Working	0
MES_HMIUnit.CompletePOI.Handshake.Completed	0
MES_HMIUnit.CompletePOI.Handshake.Error	0
MES_HMIUnit.CompletePOI.Handshake.Mode	0
MES_HMIUnit.CompletePOI.Handshake.Response	0
MES_HMIUnit.CompletePOI.Handshake.OwnerMode	10000
MES_HMIUnit.CompletePOI.Handshake.OperationMode	30000
MES_HMIUnit.CompletePOI.Handshake.Config	{...}
MES_HMIUnit.CompletePOI.Handshake.Counter	{...}

รูปที่ 4.30 ผลการทดสอบ PCS ส่งบิต Trigger ในขั้นตอน Complete POI

MES_HMIUnit.CompletePOI.Handshake	{...}
MES_HMIUnit.CompletePOI.Handshake.Trigger	0
MES_HMIUnit.CompletePOI.Handshake.Working	1
MES_HMIUnit.CompletePOI.Handshake.Completed	0
MES_HMIUnit.CompletePOI.Handshake.Error	0
MES_HMIUnit.CompletePOI.Handshake.Mode	0
MES_HMIUnit.CompletePOI.Handshake.Response	0
MES_HMIUnit.CompletePOI.Handshake.OwnerMode	10000
MES_HMIUnit.CompletePOI.Handshake.OperationMode	30000
MES_HMIUnit.CompletePOI.Handshake.Config	{...}
MES_HMIUnit.CompletePOI.Handshake.Counter	{...}

รูปที่ 4.31 ผลการทดสอบ MOM ส่งบิต Working ในขั้นตอน Complete POI

MES_HMIUnit.CompletePOI.Handshake	{...}
MES_HMIUnit.CompletePOI.Handshake.Trigger	0
MES_HMIUnit.CompletePOI.Handshake.Working	0
MES_HMIUnit.CompletePOI.Handshake.Completed	1
MES_HMIUnit.CompletePOI.Handshake.Error	0
MES_HMIUnit.CompletePOI.Handshake.Mode	0
MES_HMIUnit.CompletePOI.Handshake.Response	0
MES_HMIUnit.CompletePOI.Handshake.OwnerMode	10000
MES_HMIUnit.CompletePOI.Handshake.OperationMode	30000
MES_HMIUnit.CompletePOI.Handshake.Config	{...}
MES_HMIUnit.CompletePOI.Handshake.Counter	{...}

รูปที่ 4.32 ผลการทดสอบ MOM ส่งบิต Complete ในขั้นตอน Complete POI

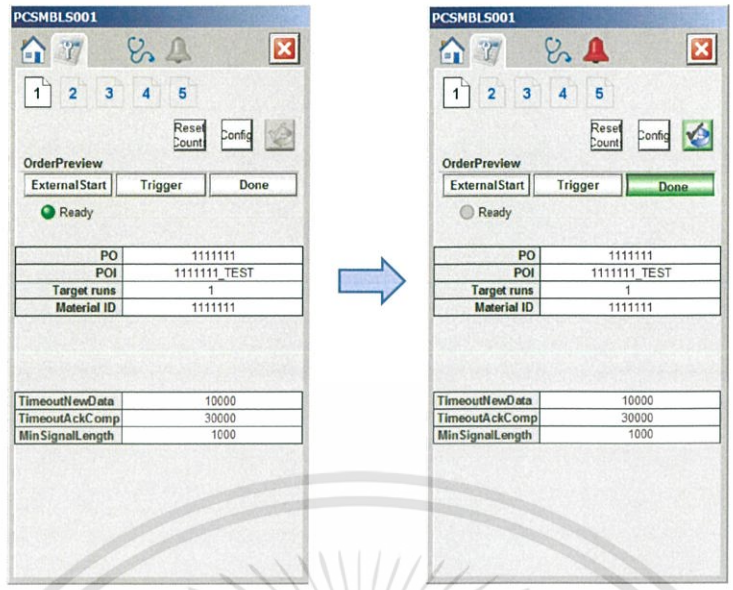
4.3 ผลการทดสอบการแสดงผลข้อมูลผ่านระบบสกาดา (SCADA)

ในการทดสอบการแสดงผลข้อมูลผ่านระบบสกาดา จะทำการทดสอบทีละขั้นตอนตามที่ได้ออกแบบฟังก์ชั่นการทำงานและเขียนโปรแกรม PLC ไว้ และค่าพารามิเตอร์ต่าง ๆ ที่อยู่ใน PO จะเป็นค่าที่ถูกกำหนดขึ้นสำหรับการทดสอบเท่านั้น ไม่ใช่ค่าที่แท้จริงของกระบวนการ เช่นเดียวกับการทดสอบการทำงานของโปรแกรม

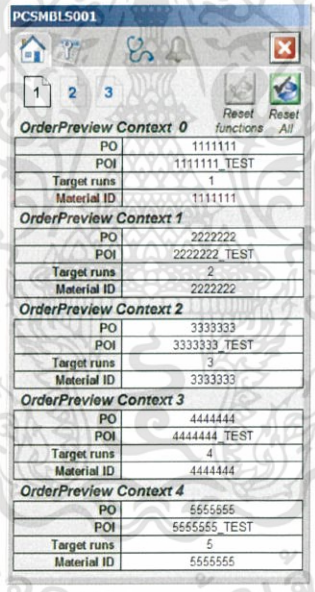
4.3.1 Order Preview

จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Ready และบิต Done ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ ดังรูปที่ 4.33 รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ที่ MOM ดาวน์โหลดมาให้ได้อย่างถูกต้องดังรูปที่ 4.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Order Preview

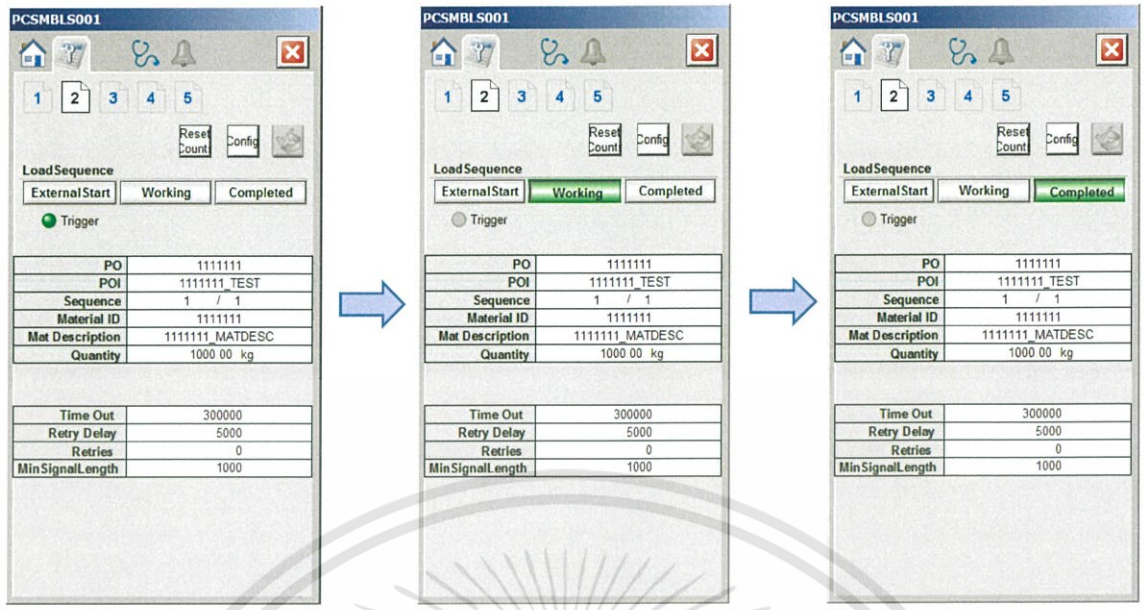


รูปที่ 4.34 ผลการทดสอบการแสดงค่าพารามิเตอร์ต่าง ๆ ในขั้นตอน Order Preview

4.3.2 Load Sequence

จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ของ PO ที่ถูกเลือกจาก 5 PO ได้อย่างถูกต้องดังรูปที่ 4.35

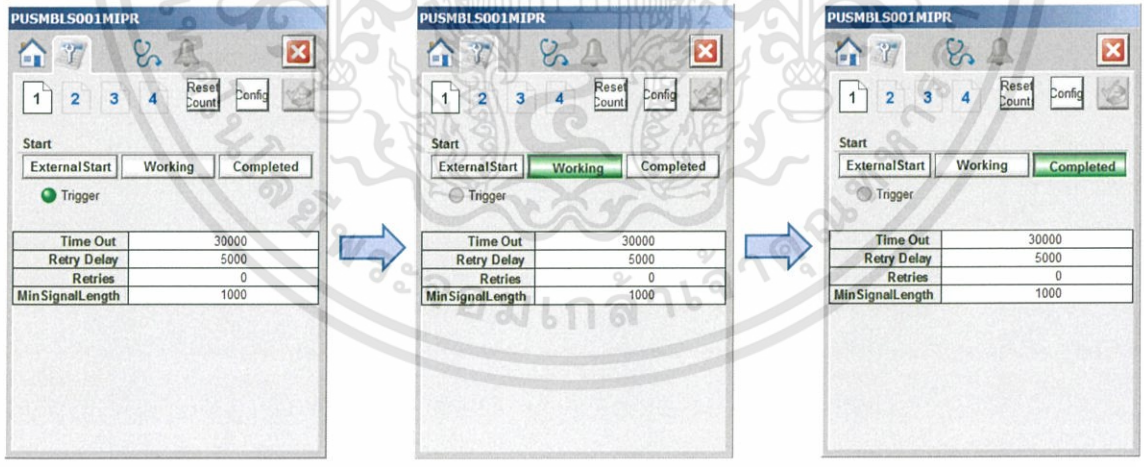
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.35 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Load Sequence

4.3.3 Start Unit PO

จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ได้อย่างถูกต้องดังรูปที่ 4.36

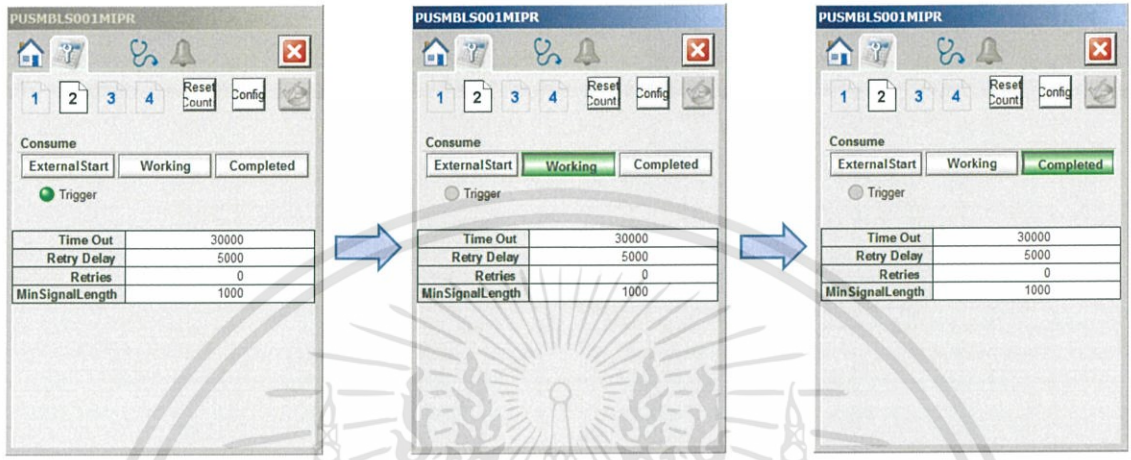


รูปที่ 4.36 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Start Unit PO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.4 Consume

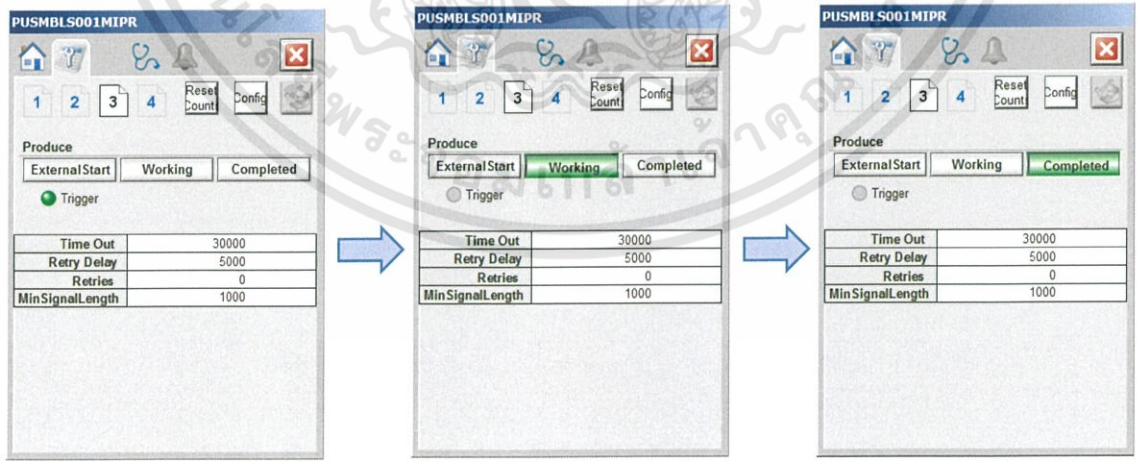
จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ได้อย่างถูกต้องดังรูปที่ 4.37



รูปที่ 4.37 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Consume

4.3.5 Produce

จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ได้อย่างถูกต้องดังรูปที่ 4.38

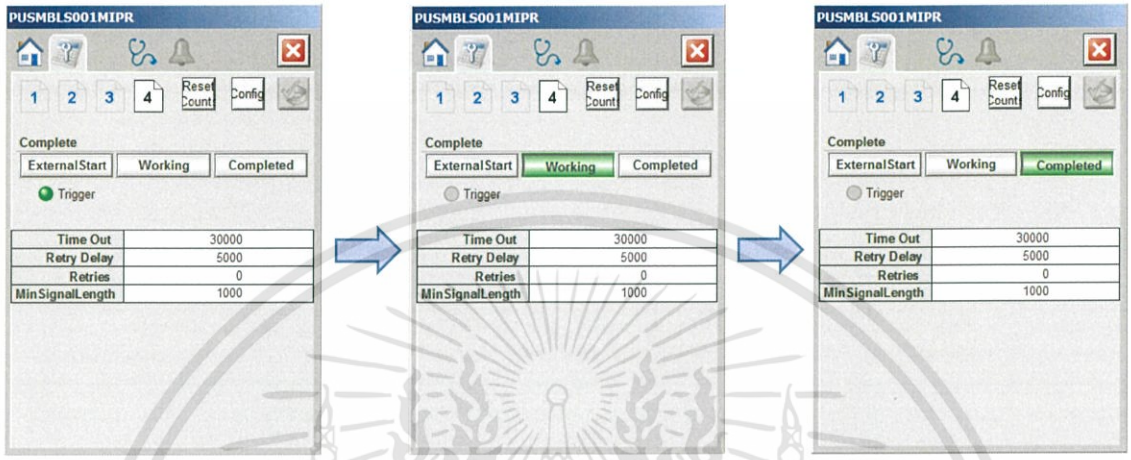


รูปที่ 4.38 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Produce

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 Complete Unit PO

จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ได้อย่างถูกต้องดังรูปที่ 4.39



รูปที่ 4.39 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Complete Unit PO

4.3.7 Complete Sequence

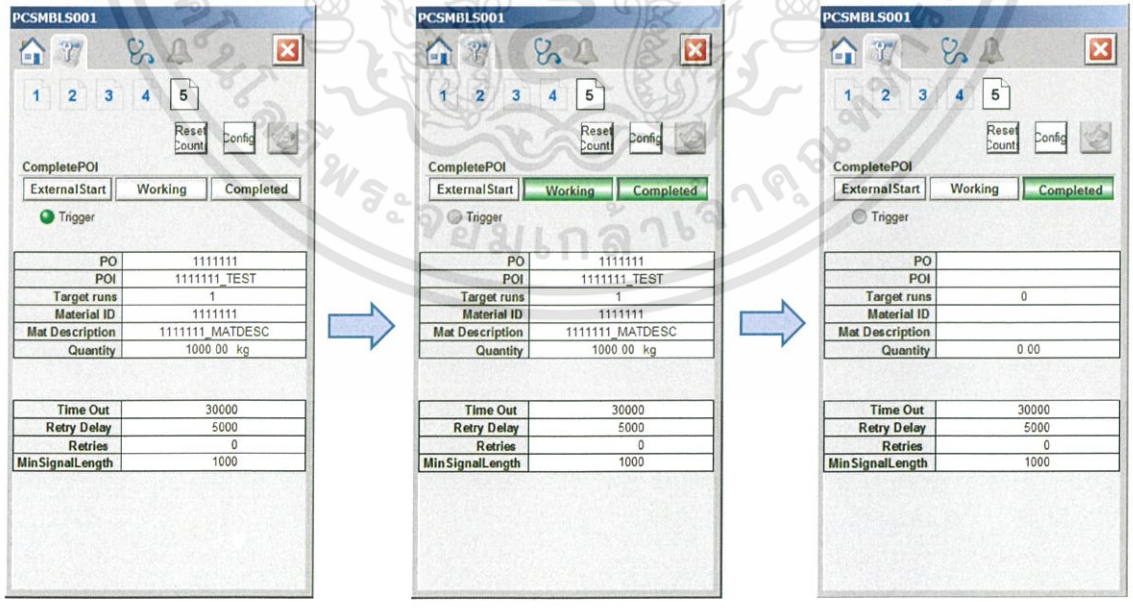
จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ของ PO ที่ถูกเลือกจาก 5 PO ได้อย่างถูกต้องดังรูปที่ 4.40

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.40 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Complete Sequence

4.3.8 Complete POI

จากการทดสอบกราฟิกของสกาดาสามารถสั่งงานและแสดงค่าพารามิเตอร์ต่าง ๆ ของกระบวนการได้ ไม่ว่าจะเป็นการแสดงสถานะการทำงานของบิต Trigger, Working และ Complete ซึ่งทำงานได้ตามเงื่อนไขที่ถูกกำหนดไว้ รวมทั้งการแสดงค่าพารามิเตอร์ต่าง ๆ ของ PO ที่ถูกเลือกจาก 5 PO ได้อย่างถูกต้องดังรูปที่ 4.41



รูปที่ 4.41 ผลการทดสอบการแสดงสถานะการทำงานในขั้นตอน Complete POI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินการออกแบบ ศูนย์กลางรวบรวมข้อมูลของกระบวนการบรรจุภัณฑ์อาหารสัตว์และส่วนแสดงผลข้อมูลในกระบวนการ พบว่า ศูนย์กลางรวบรวมข้อมูลมีการทำงานอย่างเป็นลำดับตามฟังก์ชั่นที่กำหนดไว้ สามารถรับส่งข้อมูลกับ MOM ได้อย่างถูกต้อง 100 เปอร์เซ็นทุกขั้นตอนตรงตามความต้องการของลูกค้า และในการรับข้อมูลจากเครื่องจักรที่อยู่ในระบบ PCS ได้ทำการจำลองข้อมูลส่งให้กับ MOM ซึ่งพบว่าศูนย์กลางรวบรวมข้อมูลสามารถรับส่งข้อมูลให้กับ MOM ได้ถูกต้อง และสามารถส่งข้อมูลได้ตามเวลาจริง ในส่วนของการแสดงผลข้อมูลผ่านระบบสกาดา ศูนย์กลางรวบรวมข้อมูลสามารถนำข้อมูลของกระบวนการแสดงผลผ่านระบบสกาดาได้อย่างถูกต้อง ซึ่งส่วนแสดงผลนั้นจะแสดงค่าพารามิเตอร์ต่าง ๆ ที่ต้องนำไปใช้ในกระบวนการบรรจุภัณฑ์อาหารสัตว์ รวมถึงแสดงสถานะของขั้นตอนการทำงานต่าง ๆ เพื่อให้ผู้ปฏิบัติงานสามารถดำเนินงานในขั้นต่อไปได้อย่างถูกต้องและช่วยให้การสั่งดำเนินการผลิตจาก MOM ไปที่เครื่องจักรหลาย ๆ เครื่องในระบบ PCS มีความสะดวกและมีถูกต้องของข้อมูลมากยิ่งขึ้น และสามารถนำข้อมูลในกระบวนการไปใช้ในการวิเคราะห์ในระบบ MES ได้อย่างรวดเร็ว

5.2 ปัญหาที่พบและแนวทางการแก้ไข

5.2.1 ปัญหาที่พบ

1) ข้อจำกัดทางด้านภาษา เนื่องจากโครงงานนี้เป็นการทำงานร่วมกันของหลายบริษัทต่างชาติ การสื่อสารภาษาอังกฤษจึงเป็นสิ่งจำเป็นมากในการทำงานกับผู้ร่วมงานที่เป็นชาวต่างชาติ
2) ผู้เกี่ยวข้องกับโครงงานนี้ มีความเข้าใจที่ไม่ตรงกันเกี่ยวกับลักษณะของโครงงาน เช่น การทำงานของฟังก์ชั่นในโปรแกรม PLC และบนสกาดาเป็นต้น
3) ความรู้ด้านการเขียนโปรแกรม PLC และสกาดายังมีไม่เพียงพอ

5.2.2 แนวทางการแก้ปัญหา

1) หมั่นฝึกฝนตนและพัฒนาตนเองให้มีความพร้อมทางด้านภาษาอยู่เสมอ
2) ชี้แจงและทำความเข้าใจกับทุก ๆ ฝ่ายที่เกี่ยวข้องกับโครงงานนี้ ให้มีความเข้าใจที่ตรงกัน เพื่อใช้บรรลุตามวัตถุประสงค์ที่ตั้งไว้
3) ศึกษาวิธีการใช้งานโปรแกรมจากอินเตอร์เน็ต สอบถามเทคนิคในการใช้งานเพิ่มเติมและความรู้ต่าง ๆ จากพี่ ๆ ในบริษัท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะ

ในการออกแบบศูนย์กลางรวบรวมข้อมูลจำเป็นต้องรู้เงื่อนไขการทำงานของเครื่องจักรต่าง ๆ ในกระบวนการบรรจุภัณฑ์อาหารสัตว์ จึงต้องมีการศึกษาและไปดูที่สถานที่จริงเพื่อให้การออกแบบศูนย์กลางรวบรวมข้อมูลนั้นเหมาะสมกับการดำเนินงานของเครื่องจักร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] Industrial Revolution. 2559. "Enterprise Resource Planning (ERP)". [ระบบออนไลน์]. แหล่งที่มา : https://www.irev.co.th/erpmes/ (2 พฤศจิกายน 2561)

[2] ASTech Pte. 2561. "Manufacturing Execution Systems". [ระบบออนไลน์]. แหล่งที่มา : http://astech.com.sg/th/?p=644 (2 พฤศจิกายน 2561)

[3] แพร ปานประโคน. 2557. "คอนเซนเทรเตอร์ (Concentrator)". [ระบบออนไลน์]. แหล่งที่มา : http://panprakorn. blogspot.com/2014/03/concentrator-communications-processor -2.html (2 พฤศจิกายน 2561)

[4] Jakarin Automation. 2558. "ความรู้เบื้องต้นเกี่ยวกับระบบ SCADA". [ระบบออนไลน์]. แหล่งที่มา : http://mechatronnic2day.blogspot.com/2015/03/scada-1.html (2 พฤศจิกายน 2561)

[5] Chatchai Ribruamsup. 2560. "PLC คือ อะไร". [ระบบออนไลน์]. แหล่งที่มา : http://www .advance-electronic.com/blog/detail/113/th/PLC-คือ-อะไร.html (2 พฤศจิกายน 2561)

[6] Star circuit. 2559. "โครงสร้างโดยทั่วไปของ PLC". [ระบบออนไลน์]. แหล่งที่มา : http:// www.star-circuit.com/article/PLC.html (2 พฤศจิกายน 2561)

[7] Autocentrated. 2556. "ภาษาที่ใช้ในการเขียนโปรแกรมมาตรฐาน IEC1131–3". [ระบบออนไลน์]. แหล่งที่มา : http://autocentrated.com/index.php (2 พฤศจิกายน 2561)

[8] Kred4u. 2561. "Ethernet คืออะไร". [ระบบออนไลน์]. แหล่งที่มา : https://www.xn12cg 1cxchd0a2gzc1c5d5a.net/ethernet/ (2 พฤศจิกายน 2561)

[9] PONGSAKRIVERPLUS. 2560. "PLC Protocol". [ระบบออนไลน์]. แหล่งที่มา : https:// riverplusblog.com/ 2011/ 08/ 18/plc-protocol-การสื่อสารแบบ-modbus-protocol/ (2 พฤศจิกายน 2561)

[10] Sutinnakorn Pankham. 2556. "OLE For Process Control (OPC)". [ระบบออนไลน์]. แหล่งที่มา : http://automationreview.blogspot.com/2013/10/ole-for-processcontrol -opc.html (2 พฤศจิกายน 2561)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

ชื่อ – นามสกุล นางสาวจารวี อ่อนนิวัฒนานนท์
วัน เดือน ปีเกิด 18 เมษายน 2540
ที่อยู่ 221/1 หมู่ 7 ต.ลาดยาว อ.ลาดยาว จ.นครสวรรค์ 60150
E-mail onniwattananon5151@gmail.com
โทรศัพท์ 08 8150 3152

ประวัติการศึกษา

- พ.ศ. 2555 - 2557 ระดับมัธยมศึกษาตอนปลาย โรงเรียนสตรีนครสวรรค์ จังหวัดนครสวรรค์
- พ.ศ. 2558 - ปัจจุบัน วิศวกรรมศาสตร์บัณฑิต หลักสูตรวิศวกรรมการวัดคุม
  ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
  สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์

- นักศึกษาฝึกงาน แผนก DCS & Control System
  บริษัท พีทีที โกลบอล เคมิคอล จำกัด (มหาชน)
- นักศึกษาโครงการสหกิจศึกษา แผนก Project Engineering
  บริษัท พีเอส เอ็นจิเนียริ่ง คอนซัลแตนท์ จำกัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้