รายงานสหกิจศึกษาฉบับสมบูรณ์

การออกแบบระบบควบคุมอัตโนมัติด้วย PLC ผ่าน SCADA สำหรับกระบวนการบรรจุอาหารสัตว์

Automatic Control System Design by PLC via SCADA for Feed Mill Packing Process

นายสหัสชัย วิชาวิทยากุล

ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

การออกแบบระบบควบคุมอัตโนมัติด้วย PLC ผ่าน SCADA สำหรับกระบวนการบรรจุอาหารสัตว์

Automatic Control System Design by PLC via SCADA for Feed Mill Packing Process

นายสหัสชัย วิชาวิทยากุล

ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

ชื่อโครงงานสหกิจศึกษา การออกแบบระบบควบคุมอัตโนมัติด้วย PLC ผ่าน SCADA สำหรับกระบวนการบรรจุอาหารสัตว์

ชื่อ-สกุล นักศึกษา นายสหัสชัย วิชาวิทยากุล

คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมการวัดและควบคุม

ชื่อ-สกุล อาจารย์นิเทศ รศ.ดร.วิศรุต ศรีรัตนะ

ชื่อ-สกุล ผู้นิเทศงาน นายชำนาญ ตุมตอง

ชื่อสถานประกอบการ บริษัท อินคอนโทรล จำกัด

บทคัดย่อ

รายงานสหกิจศึกษาฉบับสมบูรณ์นี้ได้อธิบายถึงการศึกษาและการออกแบบระบบควบคุมอัตโนมัติสำหรับกระบวนการบรรจุอาหารสัตว์ให้กับบริษัทลูกค้าของ บริษัท อินคอนโทรล จำกัด ในนามของวิศวกรอัตโนมัติของทาง บริษัท อินคอนโทรล จำกัด โดยการใช้งานตัวควบคุมเชิงตรรกะที่สามารถเขียนโปรแกรมได้ ผ่านทางหน้าจอควบคุมด้วยบุคคลของระบบสกาดา เพื่อเป็นการปรับปรุงกระบวนการผลิตให้ทำงานได้ตามเป้าหมาย ลดข้อผิดพลาดในการทำงาน รวมทั้งเพิ่มประสิทธิภาพและความปลอดภัยในการดำเนินงาน โดยการออกแบบการควบคุมประกอบด้วย การเขียนโปรแกรมควบคุม การออกแบบหน้าจอที่สามารถควบคุมรวมทั้งแสดงผลได้ และการทดสอบการทำงานของระบบ เพื่อให้พร้อมต่อการติดตั้งร่วมกับระบบอื่น ๆ ที่จำเป็นในการควบคุมกระบวนการ

คำสำคัญ: โรงงานอาหารสัตว์, ตัวควบคุมเชิงตรรกะที่สามารถเขียนโปรแกรมได้, หน้าจอควบคุมด้วยบุคคล, สกาดา, การทดสอบการทำงานของระบบ

Co-operative Research Title: Automatic Control System Design by PLC via SCADA for Feed Mill Packing Process

Student Intern Name: Mr. Sahaschai Wichawittayakun

Faculty: Engineering **Department:** Instrumentation Engineering

Advisor Name: Assoc. Prof. Dr. Witsarut Sriratana

Mentor Name: Mr. Chumnan Toomtong

Company: InControl Company Limited

ABSTRACT

This cooperative educational report is about of studying and designing of packing process of animal feed mill automatic control system with PLC (Programmable Logic Controller) via HMI (Human-Machine Interface) of SCADA (Supervisory Control and Data Acquisition) system to improve the process production for achieve the objectives by reducing the possibility of fault and increasing efficiency with safety in the operation. In the part of procedure consists of programing the PLC controller program, designing the HMI display from SCADA software and the last is commissioning operation system for preparing the process production to ready for installation with other important systems.

Keywords: Feed Mill, PLC, HMI, SCADA, Commissioning

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับสมบูรณ์นี้สำเร็จลุล่วงไปด้วยดี เนื่องด้วยการสนับสนุนเป็นอย่างดีจากทาง บริษัท อินคอนโทรล จำกัด ซึ่งได้ให้โอกาสแก่นักศึกษาสหกิจในการเข้าไปปฏิบัติงานในองค์กร รวมทั้งให้ความรู้และการดูแลที่อบอุ่นตลอดระยะเวลาการปฏิบัติงานสหกิจศึกษา จึงขอขอบพระคุณทางด้าน บริษัท อินคอนโทล จำกัด และ นายชำนาญ ตุมตอง ผู้นิเทศงาน รวมถึงพนักงานทุกท่านที่ได้มอบประสบการณ์อันมีค่ายิ่งในครั้งนี้

ทั้งนี้ ต้องขอขอบพระคุณ รศ.ดร.วิศรุต ศรีรัตนะ และคณาจารย์สาขาวิชาวิศวกรรมการวัดและควบคุมทุกท่าน ที่ได้ให้ความช่วยเหลือและคำปรึกษาอันเป็นประโยชน์ในการปฏิบัติงานและการจัดทำโครงงานสหกิจศึกษาฉบับสมบูรณ์นี้ด้วยดีเสมอมา

นายสหัสชัย วิชาวิทยากุล

สารบัญ

หน้า

บทคัดย่อ I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 2

1.4 วิธีดำเนินงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 4

2.1 อุตสาหกรรมอาหารสัตว์สำเร็จรูป 4

2.1.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป 4

2.1.2 ปัญหา และอุปสรรคด้านการผลิตอาหารสัตว์ 7

2.2 ความรู้พื้นฐานเกี่ยวกับ PLC และ SCADA 8

2.2.1 ความหมายของ PLC 8

2.2.2 โครงสร้างของ PLC 8

2.2.3 ภาษาที่ใช้เขียน PLC 13

2.3.1 ความหมายของ SCADA 15

2.3.2 โครงสร้างของระบบ SCADA 16

2.4 ความสำคัญของ OPC 18

สารบัญ (ต่อ)

หน้า

2.5 ซอฟต์แวร์ที่เกี่ยวข้อง 19

2.5.1 Studio 5000 Logix Designer 19

2.5.2 Wonderware InTouch HMI 2014 R2 20

2.5.3 ArchestrA IDE (Integrated Development Environment) 22

2.5.4 SMC (System Management Console) 22

2.5.5 RSLinx Classic 23

2.5.6 RSLogix Emulate 5000 Chassis Monitor 23

บทที่ 3 การออกแบบระบบควบคุมของกระบวนการบรรจุอาหารสัตว์ 24

3.1 สรุปสิ่งจำเป็นสำหรับการออกแบบระบบควบคุม 24

3.1.1 P&ID (Piping and Instrumentation Diagram) 24

3.1.2 FDS (Functional Design Specification) 25

3.1.3 รายชื่ออุปกรณ์ที่ทำการควบคุม 25

3.2 ทำการเขียนโปรแกรมควบคุม PLC 26

3.2.1 สร้าง Tag Name ให้อุปกรณ์บน PLC 26

3.2.2 ออกแบบ Flowchart Diagram 27

3.2.3 เขียนโปรแกรมควบคุม PLC 28

3.2.4 ทำการทดสอบระบบควบคุม PLC 28

3.3 ทำการออกแบบหน้าจอควบคุม HMI ผ่านระบบ SCADA 29

3.3.1 สร้าง Object และเพิ่ม Attribute ในระบบ SCADA 29

3.3.2 ออกแบบกราฟฟิกลงใน Object 30

3.3.3 ออกแบบหน้าจอควบคุม HMI 31

3.4 ทำการเชื่อมต่อระหว่าง OPC SERVER และ OPC CLIENT 33

3.4.1 ตั้งค่า Topic Name ของ OPC Server 34

3.4.2 ตั้งค่า Topic Name ของ OPC Client 35

3.5 ทำการทดสอบการสั่งการ PLC ผ่านหน้าจอควบคุม HMI 37

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการดำเนินงาน 39

บทที่ 5 สรุปผลและข้อเสนอแนะ 44

5.1 สรุปผล 44

5.2 ปัญหาและแนวทางแก้ไข 44

5.3 ข้อเสนอแนะ 44

เอกสารอ้างอิง 45

ประวัติผู้เขียน 46

สารบัญตาราง

ตารางที่ **หน้า**

1.1 แผนการดำเนินงาน 3
2.1 คุณสมบัติของอุปกรณ์ที่พร้อมใช้งานโปรแกรม Studio 5000 Logix Designer 19
2.2 คุณสมบัติของอุปกรณ์ที่พร้อมใช้งานโปรแกรม Wonderware InTouch HMI 2014 20

สารบัญรูป

รูปที่ **หน้า**

2.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป 4

2.2 การรับวัตถุดิบเข้าสายการผลิต 5

2.3 การบดวัตถุดิบ 5

2.4 การผสมวัตถุดิบ 6

2.5 การอัดเม็ด 6

2.6 การบรรจุหีบห่อ 7

2.7 โครงสร้างของ PLC 8

2.8 ลักษณะของหน่วยประมวลผลกลางของ PLC ยี่ห้อ ALLEN-BRADLEY SLC 500 9

2.9 ลักษณะของ ROM ยี่ห้อ AMERICAN MEGATRENDS AMIBIOS 586 10

2.10 ตัวอย่างอุปกรณ์ที่ใช้ส่งสัญญาณให้หน่วยอินพุตของ PLC 11

2.11 ตัวอย่างอุปกรณ์ที่รับสัญญาณจากหน่วยเอาต์พุตของ PLC 11

2.12 ลักษณะของแหล่งจ่ายพลังงาน ยี่ห้อ OMRON SBAS-240006 12

2.13 ภาษา LADDER DIAGRAM 13

2.14 ภาษา FUNCTION BLOCK DIAGRAM 13

2.15 ภาษา INSTRUCTION LIST 14

2.16 ภาษา STRUCTURE TEXT 14

2.17 ภาษา SEQUENTIAL FLOW CHART 15

2.18 ลักษณะของระบบควบคุม โดยใช้ระบบ SCADA 16

2.19 โครงสร้างของระบบ SCADA 17

2.20 การติดต่อสื่อสารระหว่าง PLC และ OPC CLIENT ผ่านทาง OPC SERVER 18

2.21 ลักษณะของซอฟต์แวร์ STUDIO 5000 LOGIX DESIGNER 19

2.22 โลโก้ซอฟต์แวร์ WONDERWARE INTOUCH HMI 2014 R2 20

2.23 ลักษณะของซอฟต์แวร์ INTOUCH WINDOWMAKER 21

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

2.24 ลักษณะของซอฟต์แวร์ INTOUCH WINDOWVIEWER 21

2.25 ลักษณะของซอฟต์แวร์ ARCHESTRA IDE 22

2.26 ลักษณะของซอฟต์แวร์ SMC 22

2.27 ลักษณะของซอฟต์แวร์ RSLINX CLASSIC 23

2.28 ลักษณะของซอฟต์แวร์ RSLOGIX EMULATE 5000 CHASSIS MONITOR 23

3.1 แผนภาพ P&ID ของกระบวนการบรรจุอาหารสัตว์ 24

3.2 FDS ของกระบวนการบรรจุอาหารสัตว์ 25

3.3 รายชื่ออุปกรณ์ที่ใช้ในการควบคุมของกระบวนการบรรจุอาหารสัตว์ 25

3.4 รายชื่อ TAG NAME ของอุปกรณ์บน PLC 26

3.5 ตัวอย่าง FLOWCHART DIAGRAM สำหรับการเขียนโปรแกรมควบคุม PLC 27

3.6 ตัวอย่างการเขียนโปรแกรมควบคุม PLC 28

3.7 การทดสอบระบบควบคุม PLC โดยใช้โปรแกรม RSLOGIX EMULATE 5000 28

3.8 การสร้าง OBJECT ในซอฟต์แวร์ ARCHESTRA IDE 29

3.9 การเพิ่ม ATTRIBUTE ลงใน OBJECT ในซอฟต์แวร์ ARCHESTRA IDE 29

3.10 การสร้างกราฟฟิกลงใน OBJECT 30

3.11 การฝัง ATTRIBUTE ลงในกราฟฟิก 30

3.12 การเขียน SCRIPT เพื่อให้ ATTRIBUTE เชื่อมโยงค่ากับ OPC SERVER 31

3.13 การสร้าง INSTANCE FILE จาก INTOUCH WINDOWMAKER 31

3.14 การสร้าง WINDOW เพื่อออกแบบหน้าจอควบคุม 32

3.15 การดึงกราฟฟิกใน OBJECT มาใช้ในซอฟต์แวร์ INTOUCH WINDOWMAKER 32

3.16 แผนผังการเชื่อมต่อข้อมูลระหว่าง OPC SERVER และ OPC CLIENT 33

3.17 การตั้งค่า TOPIC NAME 34

3.18 การจับคู่ TOPIC NAME กับ PLC จำลอง 34

3.19 การสร้าง OPC OBJECT ในซอฟต์แวร์ ARCHESTRA SMC 35

3.20 การตั้งค่า OPC OBJECT ในซอฟต์แวร์ ARCHESTRA SMC 36

สารบัญรูป (ต่อ)

รูปที่ **หน้า**

3.21 การตั้งค่า OPC OBJECT ในซอฟต์แวร์ ARCHESTRA SMC....36
3.22 การแสดงผลของ ATTRIBUTE ในซอฟต์แวร์ ARCHESTRA SMC เมื่อเชื่อมต่อสำเร็จ....37
3.23 การเปิดหน้าต่าง INTOUCH WINDOWVIEWER จาก INTOUCH WINDOWMAKER....37
3.24 หน้าจอควบคุมที่ออกแบบไว้ แสดงบน INTOUCH WINDOWVIEWER....38
3.25 ตัวอย่างของการสั่งการที่เป็นไปตามตรรกะ PLC....38
4.1 (ก) หน้าจอควบคุม HMI ของกระบวนการบรรจุอาหารสัตว์หน้า 1....39
4.1 (ข) หน้าจอควบคุม HMI ของกระบวนการบรรจุอาหารสัตว์หน้า 2....39
4.2 การกรอกค่า STATION PACKING และ SOURCE BIN....40
4.3 แสดงหน้าต่างแจ้งเตือน....40
4.4 การกรอกข้อมูลที่ถูกต้อง และพร้อมสำหรับการกดปุ่ม START....41
4.5 แสดงการไล่ลำดับการเปิดอุปกรณ์ใน STATION PACKING....41
4.6 แสดงปุ่ม STOP ที่สามารถกดได้หลังจากอุปกรณ์ทุกตัวใน STATION PACKING เปิดหมดแล้ว....42
4.7 แสดงสถานะของกระบวนการในสถานะ HELD....42
4.8 แสดงสถานะของกระบวนการในสถานะ ABORTED....43
4.9 แสดงอุปกรณ์ที่เกิด FAULT ในกระบวนการที่กำลังทำงานอยู่....43

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัท อินคอนโทรล จำกัด เป็นบริษัทที่ดำเนินธุรกิจเกี่ยวกับการออกแบบ ปรับปรุงระบบ ควบคุมแบบอัตโนมัติ ตั้งแต่การควบคุมเครื่องจักรไปจนถึงกระบวนการที่ซับซ้อน ครอบคลุมกลุ่มลูกค้าตั้งแต่โรงงานอุตสาหกรรม ระบบสาธารณูปโภคพื้นฐาน ไปจนถึงส่วนการก่อสร้างอาคารต่าง ๆ ที่ต้องการระบบอัตโนมัติ โดยสามารถดำเนินการได้ทั้งด้านเครื่องมือตรวจวัด ระบบควบคุม HMI (Human-Machine Interface) ระบบฐานข้อมูล และ MES (Manufacturing Execution Systems)

ทั้งนี้ บริษัท อินคอนโทรล จำกัด ได้รับมอบหมายให้ดำเนินการปรับปรุงระบบควบคุมการผลิตของบริษัทลูกค้า ซึ่งเป็นบริษัทในกลุ่มอุตสาหกรรมการผลิตอาหารสัตว์สำเร็จรูป เนื่องจากทางบริษัทลูกค้าเล็งเห็นว่าอุตสาหกรรมการผลิตอาหารสัตว์สำเร็จรูปในปัจจุบันเริ่มขยายตัวเพิ่มขึ้นตามความต้องการของธุรกิจปศุสัตว์ อันเป็นผลมาจากอุปสงค์ที่สูงขึ้นของผู้บริโภค ดังนั้น จึงมีความต้องการที่จะดำเนินโครงการปรับปรุงการควบคุมกระบวนการผลิต จากเดิมคือแบบใช้คนควบคุม (Manual Control) มาเป็นแบบควบคุมอัตโนมัติ (Automatic Control) เพื่อเพิ่มประสิทธิภาพในการผลิต และเป็นผลไปสู่การเพิ่มปริมาณและคุณภาพของผลิตภัณฑ์ให้สามารถแข่งขันในกลุ่มอุตสาหกรรม

ด้วยเหตุนี้ นักศึกษาสหกิจจึงได้มีโอกาสเป็นส่วนหนึ่งในการดำเนินงานของการทำโครงงานระบบควบคุมอัตโนมัติของกระบวนการผลิตอาหารสัตว์ ในนามของพนักงานของ บริษัท อินคอนโทรล จำกัด ในตำแหน่งวิศวกรอัตโนมัติ และได้รับมอบหมายหน้าที่ให้ออกแบบระบบควบคุมกระบวนผลิตอาหารสัตว์ในส่วนของระบบบรรจุอาหารสัตว์ โดยการใช้เขียนโปรแกรม PLC ผ่านหน้าจอ HMI ของระบบ SCADA เพื่อปรับปรุงกระบวนการผลิตของบริษัทลูกค้า ให้เป็นไปตามเป้าหมาย ลดข้อผิดพลาดในการทำงาน ตลอดจนเพิ่มประสิทธิภาพ และความปลอดภัยในกระบวนการผลิต

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาและออกแบบระบบควบคุมอัตโนมัติกระบวนการบรรจุอาหารสัตว์ด้วย PLC
2. เพื่อออกแบบส่วนแสดงผล HMI ของระบบ SCADA ให้อำนวยความสะดวกแก่ผู้ปฏิบัติงานในการสังเกต และสั่งการระบบ
3. เพื่อพัฒนากระบวนการบรรจุอาหารสัตว์ของบริษัทลูกค้า จากระบบควบคุมโดยบุคคล เป็นระบบควบคุมอัตโนมัติ

1.3 ขอบเขตของโครงงาน

1. ศึกษาการใช้งานซอฟต์แวร์ Studio 5000 Logix Designer เพื่อเขียนระบบควบคุม PLC
2. ศึกษาการใช้งานซอฟต์แวร์ ArchestrA IDE และ Wonderware InTouch เพื่อออกแบบหน้าจอ HMI บนระบบ SCADA
3. ศึกษาการใช้งานซอฟต์แวร์ RSLinx Classic และ System Management Console เพื่อเชื่อมต่อ PLC และหน้าจอ HMI เข้าด้วยกัน และจำลองการทำงาน รวมถึงแก้ไขการทำงานของส่วนแสดงผลกับโปรแกรม PLC ให้สอดคล้องกัน

1.4 วิธีดำเนินงาน

1. ศึกษาการใช้งานซอฟต์แวร์ที่เกี่ยวข้อง
2. ศึกษากระบวนการผลิตของกระบวนการบรรจุอาหารสัตว์ของบริษัทลูกค้า
3. ออกแบบและสร้างกราฟิกของหน้าจอ HMI
4. ออกแบบแผนผังแสดงขั้นตอนการทำงาน (Flow Chart) ของโปรแกรมควบคุม PLC และทำการสร้างโปรแกรมควบคุมของ PLC ตามที่ได้ออกแบบไว้
5. เชื่อมต่อส่วนแสดงผล HMI ของระบบ SCADA กับ PLC ผ่านทาง OPC
6. ทดสอบ และปรับปรุงการทำงานของระบบ

ตารางที่ 1.1 แผนการดำเนินงาน

แผนการดำเนินงาน	วัน/เดือน/ปี	7/8/2560	14/8/2560	21/8/2560	28/8/2560	4/9/2560	11/9/2560	18/9/2560	25/9/2560	2/10/2560	9/10/2560	16/10/2560	23/10/2560	30/10/2560	6/11/2560	13/11/2560	20/11/2560
	สัปดาห์ที่	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ศึกษาการใช้งานซอฟต์แวร์ที่เกี่ยวข้อง		✓	✓	✓													
ศึกษากระบวนการผลิตการจัดห่ออาหารสัตว์					✓	✓											
ออกแบบและสร้างกราฟิกของหน้าจอ HMI						✓	✓	✓									
ออกแบบแผนผังแสดงขั้นตอนการทำงานของโปรแกรมควบคุม								✓	✓								
สร้างโปรแกรมควบคุมของ PLC ตามรูปแบบในแผนผัง										✓	✓	✓	✓				
ทดสอบ และปรับปรุงการทำงานของระบบ													✓	✓	✓		
จัดทำรายงานสหกิจศึกษา														✓	✓	✓	✓

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ปฏิบัติงานสามารถสังเกต และสั่งการกระบวนการได้สะดวกผ่านทางหน้าจอ HMI
2. ลดโอกาสการเกิดข้อผิดพลาดในกระบวนการทั้งต่อคน เครื่องจักร และวัตถุดิบ ทำให้กระบวนการเป็นไปตามเป้าหมาย
3. ช่วยเพิ่มความปลอดภัยในการดำเนินงาน
4. เพิ่มประสิทธิภาพในกระบวนการผลิต

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

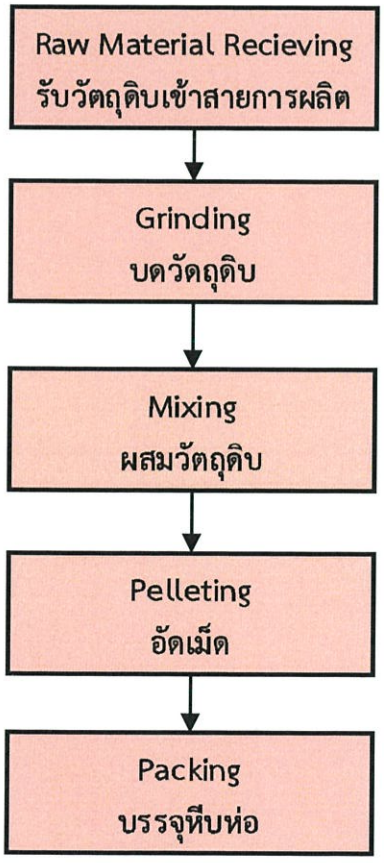
2.1 อุตสาหกรรมอาหารสัตว์สำเร็จรูป

ปัจจุบันการเลี้ยงสัตว์ได้ปรับรูปแบบจากการเลี้ยงแบบครัวเรือน มาเป็นการเลี้ยงเชิงพาณิชย์ จึงนิยมใช้อาหารสัตว์สำเร็จรูปมากขึ้น เนื่องจาก มีความสะดวก และให้ผลคุ้มค่า ดังนั้น ความต้องการของการให้อาหารสัตว์สำเร็จรูป จึงมีแนวโน้มสูงขึ้นเรื่อย ๆ ส่งผลให้อุตสาหกรรมการผลิตอาหารสัตว์สำเร็จรูปในประเทศขยายตัวเพิ่มมากขึ้นเป็นลำดับ

การเจริญเติบโตทางด้านเศรษฐกิจส่งผลต่อการพัฒนาของภาคเกษตรกรรม และการผลิตอาหารสัตว์จึงเป็นปัจจัยสำคัญที่มีส่วนช่วยพัฒนาการปศุสัตว์ หากจะพัฒนาให้แข่งขันกับตลาดในต่างประเทศได้จะต้องลดต้นทุนการผลิต ซึ่งอาหารสัตว์เป็นต้นทุนที่สูงถึงร้อยละ 60-70 ของต้นทุนทั้งหมด ส่งผลให้อุตสาหกรรมของอาหารสัตว์ที่จะต้องพัฒนากระบวนการผลิตให้มีคุณภาพ และประสิทธิภาพที่สูงขึ้น

2.1.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป

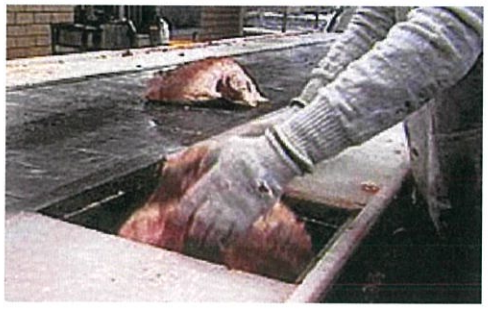
โดยทั่วไป กระบวนการที่สำคัญในการผลิตอาหารสัตว์สำเร็จรูปในอุตสาหกรรมมีขั้นตอนแสดงดังรูปที่ 2.1 ดังนี้



รูปที่ 2.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป

2.1.1.1 การรับวัตถุดิบเข้าสายการผลิต (Raw Material Receiving)

ในกระบวนการผลิตอาหารสัตว์จะมีการใช้วัตถุดิบทั้งที่แปรรูปแล้ว และยังไม่ได้ผ่านการแปรรูป สิ่งสำคัญในขั้นตอนนี้ คือ การจัดเก็บวัตถุดิบเพื่อรอการผลิตไม่ให้เกิดวัตถุดิบตกค้าง ดังแสดงในรูปที่ 2.2 เพราะวัตถุดิบที่ใช้เกือบทั้งหมดเป็นวัตถุดิบสดยังไม่ผ่านการแปรรูป เช่น ปลาป่น เปลือกกุ้ง สามารถเก็บได้เพียงช่วง ระยะเวลาสั้น ๆ เท่านั้น หากเก็บวัตถุดิบไว้นานจะเกิดการย่อยสลาย และเกิดเชื้อรา แบคทีเรีย ส่งผลเสียต่อ คุณภาพของผลิตภัณฑ์ นำไปสู่ผลกระทบต่อสัตว์ที่บริโภคอาหาร



รูปที่ 2.2 การรับวัตถุดิบเข้าสายการผลิต

2.1.1.2 การบดวัตถุดิบ (Grinding)

การบดวัตถุดิบมีความจำเป็นในกรณีที่วัตถุดิบมีขนาดใหญ่ และไม่ละเอียด เช่น เมล็ดถั่วเหลือง เมล็ดข้าวโพด มันเส้น ปลายข้าว และเปลือกกุ้ง โดยวัตถุดิบเหล่านี้จะต้องนำมาบดให้ละเอียดด้วยเครื่องบดอาหาร สำหรับวัตถุดิบขนาดเล็ก เช่น ปลาป่น รำข้าว แป้งสาลี แป้งมัน สำปะหลัง ไม่จำเป็นต้องบดให้ละเอียด ดังแสดงในรูปที่ 2.3 สำหรับวัตถุดิบจำพวกวิตามิน แร่ธาตุ และกรดอะมิโนสังเคราะห์ จะมีขนาดละเอียดอยู่แล้ว จึงไม่ต้องนำมาบด อีกทั้งความร้อนที่เกิดจากการบด จะทำให้วิตามิน แร่ธาตุ และกรดอะมิโนสังเคราะห์ถูกทำลายไปอย่างรวดเร็ว

รูปที่ 2.3 การบดวัตถุดิบ

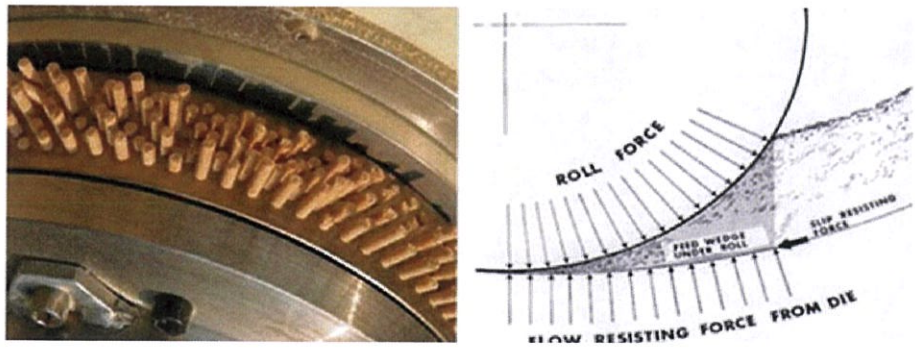
2.1.1.3 การผสมวัตถุดิบ (Mixing)

จุดมุ่งหมายของการผสมวัตถุดิบก็เพื่อให้ส่วนประกอบต่าง ๆ ผสมคลุกเคล้าเป็นเนื้อเดียวกัน ดังแสดงในรูปที่ 2.4 วัตถุดิบทุกชนิดกระจายตามเนื้อของส่วนผสมอย่างทั่วถึง กล่าวคือ หลังจากการผสมแล้วหากแบ่งส่วนผสมมาตรวจหาส่วนประกอบ ส่วนผสมที่เกิดจากการผสมที่ดีจะต้องมีส่วนประกอบของวัตถุดิบครบถ้วนตรงตามที่ระบุในสูตรอาหารก่อนการผสม

รูปที่ 2.4 การผสมวัตถุดิบ

2.1.1.4 การอัดเม็ด (Pelleting)

การอัดเม็ดมีวัตถุประสงค์เพื่อต้องการให้วัตถุดิบที่ผ่านการผสมแล้วเปลี่ยนรูปเป็นอาหารเม็ด ซึ่งมีคุณสมบัติเหมาะแก่การนำไปใช้เลี้ยงสัตว์ เนื่องจาก การอัดเม็ดช่วยลดความเป็นฝุ่นของอาหาร และไม่ทำให้คุณค่าจากอาหารที่มีชิ้นส่วนเล็กหรือเป็นผงที่ฟุ้งกระจายได้สูญเสียไป การเก็บอาหาร ใช้พื้นที่น้อยลง ดังแสดงในรูปที่ 2.5 เนื่องจาก มีความหนาแน่นของเนื้อวัตถุสูง อีกทั้งยังส่งผลให้อาหารไม่ร่วนง่ายเมื่อเกิดการกระทบ กระเทือนจากการขนส่ง นอกจากนี้ ความร้อนที่เกิดขึ้นในขณะอัดเม็ดจะทำให้แป้งบางส่วนสุกหรือเปลี่ยนโครงสร้างเอนไซม์ของสัตว์ก็จะทำการย่อยได้ง่ายขึ้น หรือจุลินทรีย์ทำงานได้มากกว่า เพิ่มการใช้ประโยชน์ของส่วนที่เป็นเยื่อใยมากขึ้น อีกทั้งยังช่วยป้องกันไม่ ให้สัตว์เลือกกินวัตถุดิบที่ชอบเท่านั้น เพราะในอาหารเม็ดมีวัตถุดิบทั้งหมดครบถ้วนในปริมาณที่สม่ำเสมอกัน



รูปที่ 2.5 การอัดเม็ด

2.1.1.5 การบรรจุหีบห่อ (Packing)

หลังจากอาหารเม็ดแห้งแล้ว และมีอุณหภูมิใกล้เคียงกับอุณหภูมิห้อง จึงพร้อมที่จะได้รับการบรรจุเข้าหีบห่อเพื่อเก็บหรือจำหน่ายต่อไป ในการบรรจุจะมีการบันทึกรายละเอียดที่จำเป็นบนหีบห่อด้วย เช่น รหัส วันบรรจุ น้ำหนักอาหาร และคุณค่าทางอาหาร ดังแสดงในรูปที่ 2.6 สิ่งที่สำคัญสุดในขั้นตอนนี้คือ สถานที่จัดเก็บอาหารต้องมีความร้อนและความชื้นไม่สูงเกินไป เพราะจะทำให้อาหารมีราเกิดขึ้น วัสดุที่ใช้บรรจุควรมีคุณสมบัติพิเศษในการป้องกันความชื้นไม่ให้ซึมผ่านไปสัมผัสกับอาหารและเพื่อป้องกันไม่ให้อาหารเสื่อมคุณภาพเร็ว

รูปที่ 2.6 การบรรจุหีบห่อ

2.1.2 ปัญหา และอุปสรรคด้านการผลิตอาหารสัตว์

ปัญหา และอุปสรรคด้านการผลิตอาหารสัตว์ที่พบในอุตสาหกรรมการผลิตอาหารสัตว์ที่พบ คือ

- วัตถุดิบที่ใช้เป็นส่วนผสมอาหารสัตว์บางชนิดเกิดการขาดแคลน เนื่องจาก อยู่นอกฤดูกาลเพาะปลูกหรือธรรมชาติเปลี่ยนแปลงผลผลิตออกไม่ตรงฤดูกาล
- วัตถุดิบที่เป็นส่วนผสมอาหารสัตว์มีราคาแพง เนื่องจาก วัตถุดิบมีน้อย ต้องแข่งขันกันรับซื้อ

2.2 ความรู้พื้นฐานเกี่ยวกับ PLC และ SCADA

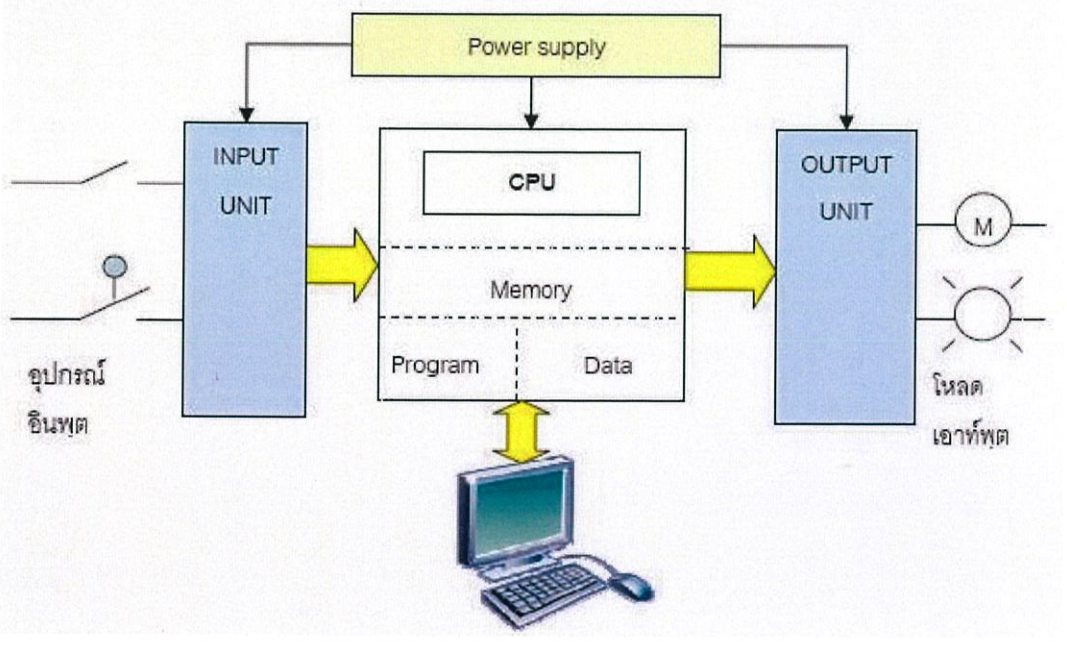
2.2.1 ความหมายของ PLC

พีแอลซี (PLC) ย่อมาจาก Programmable Logic Controller ซึ่งหมายถึง ตัวควบคุมเชิงตรรกะที่สามารถโปรแกรมได้ ในยุโรปมักเรียกว่า "ตัวควบคุมซีเควนเซอร์" (Sequencer Controller) เรียกย่อ ๆ ว่า SC แต่สมาคมผู้ผลิตอุปกรณ์ไฟฟ้า (JEMA) มักเรียก "พีซี" (PC) หรือ "พีแอลซี" (PLC)

พีแอลซีเป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการต่าง ๆ โดยภายในจะมีไมโครโปรเซสเซอร์ทำหน้าที่เป็นตัวสั่งการ มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมเครื่องจักรหรือกระบวนการ มีอินพุตที่สามารถต่อเข้ากับตัวตรวจรู้หรือสวิตช์ต่าง ๆ และมีส่วนของเอาท์พุตที่สามารถต่อเข้ากับอุปกรณ์กระทำการของเครื่องจักร เช่น วงจรแม่เหล็ก (Magnetic) วาล์วต่าง ๆ เป็นต้น รูปแบบของการควบคุมเครื่องจักรสามารถเปลี่ยนแปลงได้โดยการเปลี่ยนโปรแกรมสั่งงาน PLC โดยจะที่มีฟังก์ชันต่าง ๆ สำหรับการควบคุมอยู่ภายใน เช่น รีเลย์ รีจีสเตอร์ ตัวตั้งเวลา ตัวนับ และตัวเปรียบเทียบ เป็นต้น การเชื่อมต่อของอุปกรณ์เหล่านี้สามารถทำได้โดยซอฟต์แวร์

2.2.2 โครงสร้างของ PLC

โครงสร้างภายในของ PLC แต่ละส่วนจะร่วมกันทำงานเป็นระบบควบคุม โดยจะประกอบไปด้วยส่วนสำคัญ 5 ส่วน ดังแสดงในรูปที่ 2.7 ดังนี้



รูปที่ 2.7 โครงสร้างของ PLC

2.2.2.1 ซีพียู (CPU; Central Processing Unit)

ซีพียูหรือหน่วยประมวลผลกลาง ดังแสดงในรูปที่ 2.8 ทำหน้าที่ประมวลผลการทำงานตามคำสั่งของส่วนต่าง ๆ ตามที่ได้รับมา ผลลัพธ์จากการประมวลผลจะถูกส่งไปยังส่วนต่าง ๆ ตามที่ระบุไว้ด้วยคำสั่ง โดยความเร็วในการประมวลผลขึ้นอยู่กับการเลือกขนาดของซีพียู และขนาดของโปรแกรมด้วย

ปกติแล้วซีพียูจะใช้ไมโครโปรเซสเซอร์ขนาดตั้งแต่ 4 บิต, 8 บิต, 16 บิต, 32 บิต, 64 บิต หรือ 120 บิต มาทำงาน โดยที่ซีพียูแต่ละขนาดก็จะมีความสามารถจำกัดไม่เท่ากัน จึงทำให้ PLC แต่ละรุ่น มีความสามารถต่างกันหรือภายใน PLC บางรุ่นจะใช้ไมโครโปรเซสเซอร์ถึง 2 ตัว ช่วยกันทำงาน เวลาการประมวลผลก็จะเร็วกว่า PLC ที่ใช้ไมโครโปรเซสเซอร์เพียงแค่ตัวเดียว

รูปที่ 2.8 ลักษณะของหน่วยประมวลผลกลางของ PLC ยี่ห้อ Allen-Bradley SLC 500

2.2.2.2 หน่วยความจำ (Memory Unit)

หน่วยความจำเป็นอุปกรณ์ที่ใช้เก็บโปรแกรม และข้อมูลต่าง ๆ ของ PLC ดังแสดงในรูปที่ 2.9 กรณีที่สั่ง RUN PLC ก็จะนำเอาโปรแกรมและข้อมูลในหน่วยความจำมาประมวลผลการทำงาน สำหรับหน่วยความจำที่ใช้งานอยู่ใน PLC มีด้วยกัน 2 แบบ คือ

1) หน่วยความจำชั่วคราว (RAM: Random Access Memory)

ใช้เก็บโปรแกรม และข้อมูลที่ทำงานจากการสั่ง RUN หรือ STOP โปรแกรม และข้อมูลที่สร้างขึ้นโดยผู้ใช้ก็ถูกเก็บในส่วนนี้ด้วย คุณสมบัติของ RAM เมื่อไม่มีไฟเลี้ยงจะทำให้โปรแกรมและข้อมูลหายไปทันที ดังนั้น ภายใน PLC จึงมีแบตเตอรี่สำรองข้อมูล (Backup Battery) เพื่อใช้สำรองข้อมูล (Backup Data) กรณีที่ไฟหลัก (Main Power Supply) ไม่จ่ายไฟให้กับ PLC

2) หน่วยความจำถาวร (ROM: Read Only Memory)

ใช้จัดเก็บซอฟต์แวร์ระบบ (System Software) เป็นชุดสำรองโปรแกรมและข้อมูล (Backup Program and Data) เพื่อป้องกันกรณีที่โปรแกรมและข้อมูลใน RAM หายไป ผู้ใช้สามารถถ่ายโปรแกรมและข้อมูลเข้าไปที่ RAM ใหม่ได้ ROM ไม่จำเป็นต้องมีแบตเตอรี่สำรองข้อมูล แต่ก็มีปัญหาเรื่องเวลาในการเข้าถึงข้อมูล (Time Access) ที่ช้ากว่า RAM ดังนั้น PLC จึงมีหน่วยความจำใช้งานทั้ง RAM และ ROM ร่วมกันอยู่ โดยที่ ROM แบ่งออกเป็น 3 ชนิด ได้แก่

2.1) PROM (Programmable ROM)

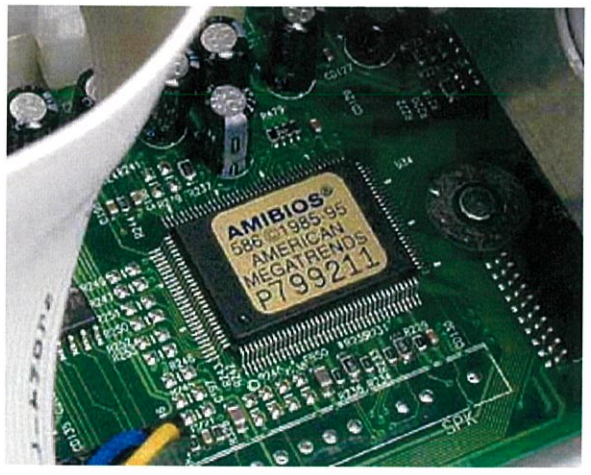
จัดเป็น ROM รุ่นแรก เขียนข้อมูลลงชิพได้เพียงครั้งเดียว ถ้าเขียนข้อมูลไม่สมบูรณ์ ชิพก็จะเสียทันทีไม่สามารถนำกลับมาเขียนใหม่ได้อีก

2.2) EPROM (Erasable Programmable ROM)

ถูกพัฒนาขึ้นมาจาก PROM ทำให้สามารถเขียนข้อมูลลงชิพได้หลายครั้ง เพียงแค่นำชิพไปฉายแสงอุลตร้าไวโอเลตก็จะเป็นการลบข้อมูลด้วยสัญญาณทางไฟฟ้าได้เลย จึงทำให้เกิดความสะดวกสบายมากขึ้น แต่เรื่องเวลาในการเข้าถึงข้อมูลก็ยังช้ากว่า RAM อยู่

2.3) EEPROM (Electrical Erasable Programmable ROM)

เป็นอุปกรณ์ที่รวมคุณสมบัติของ ROM และ RAM ไว้ด้วยกัน เนื่องจากไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรม โดยสามารถใช้งานได้เหมือนกับ RAM แต่ไม่ต้องใช้แบตเตอรี่สำรอง ราคาจึงแพงกว่า



รูปที่ 2.9 ลักษณะของ ROM ยี่ห้อ AMERICAN MEGATRENDS AMIBIOS 586

2.2.2.3 อุปกรณ์ภาคอินพุต (Input Unit)

ทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอกเข้ามา แล้วแปลงให้เป็นสัญญาณที่เหมาะสม ก่อนส่งให้หน่วยประมวลผลต่อไป ซึ่งอุปกรณ์ต่าง ๆ ที่นำมาต่อกับภาคอินพุตได้สามารถจำแนกเป็นกลุ่มได้ดังรูปที่ 2.10 โดยแต่ละกลุ่มอุปกรณ์จะมีวิธีต่อวงจรเข้าภาคอินพุตที่แตกต่างกันออกไป เวลาใช้งานจึงต้องศึกษาข้อมูลเพิ่มเติมของอุปกรณ์แต่ละชนิดก่อน เพื่อความเข้าใจขั้นตอนการทำงาน และสามารถต่อวงจรได้ถูกต้อง

รูปที่ 2.10 ตัวอย่างอุปกรณ์ที่ใช้ส่งสัญญาณให้หน่วยอินพุตของ PLC

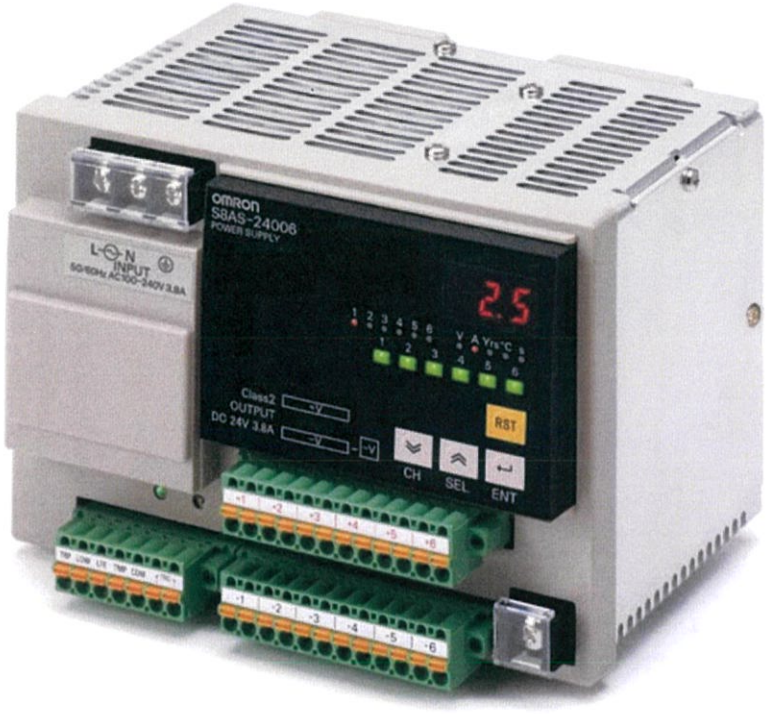
2.2.2.4 อุปกรณ์ภาคเอาต์พุต (Output Unit)

ภาคเอาต์พุตของ PLC ทำหน้าที่ส่งสัญญาณออกไปขับโหลดชนิดต่าง ๆ ตามเงื่อนไขที่โปรแกรมไว้ ชนิดของโหลดที่สามารถนำมาต่อกับภาคเอาต์พุตแยกออกเป็นกลุ่มได้ดังรูปที่ 2.11 โดยแต่ละกลุ่มก็จะควบคุมลักษณะของงานแตกต่างกันไปตามคุณสมบัติของอุปกรณ์นั้น ๆ การต่อวงจรเข้าภาคเอาต์พุตก็มีมาตรฐานทางอุตสาหกรรมกำกับอยู่เช่นกัน จึงทำให้ผู้ใช้ไม่ต้องใช้อุปกรณ์เสริมมาก เพียงแต่ดูรายละเอียดการต่อให้เข้าใจก็เพียงพอแล้ว

รูปที่ 2.11 ตัวอย่างอุปกรณ์ที่รับสัญญาณจากหน่วยเอาต์พุตของ PLC

2.2.2.5 ภาคแหล่งจ่ายพลังงาน (Power Supply Unit)

ภาคแหล่งจ่ายพลังงาน ซึ่งมีลักษณะดังรูปที่ 2.12 จะทำหน้าที่จ่ายพลังงานให้กับอุปกรณ์ภายใน PLC เช่น อุปกรณ์จำพวกไอซี (IC; Integrated Circuit), ไฟเลี้ยงวงจรกำหนดการทำงานแบบต่าง ๆ เป็นต้น นอกจากนี้ยังจ่ายพลังงานเลี้ยงวงจรที่จะนำมาต่อกับ PLC ทั้งภาคอินพุต และเอาต์พุตด้วย ซึ่งแหล่งจ่ายพลังงานของ PLC จะแบ่งออกเป็น 2 ชุด โดยชุดแรกจะใช้สำหรับอุปกรณ์ และวงจรภายในแต่ละโมดูลต่าง ๆ ของ PLC อีกชุดหนึ่งเป็นตัวจ่ายพลังงานแรงดันไฟฟ้า 24 VDC (Service Unit 24 VDC) สำหรับการต่อวงจรภาคอินพุตหรือเอาต์พุตก็ได้ ในการใช้งานจริง แหล่งจ่ายจะถูกออกแบบมา 2 ลักษณะตามโครงสร้างภายนอกของ PLC คือ แหล่งจ่ายชนิดที่รวมอยู่ในตัว PLC กับชนิดที่แยกออกมาเป็นโมดูล



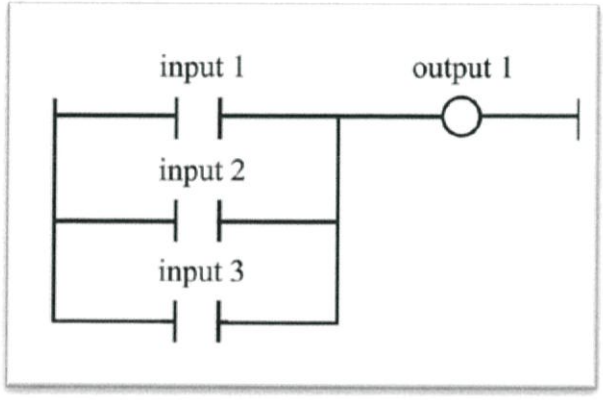
รูปที่ 2.12 ลักษณะของแหล่งจ่ายพลังงาน ยี่ห้อ OMRON S8AS-240006

2.2.3 ภาษาที่ใช้เขียน PLC

พีแอลซีแต่ละยี่ห้อจะใช้ภาษาในการเขียนโปรแกรมเพื่อสั่งงานแตกต่างกัน ซึ่งตามมาตรฐาน IEC 61131-3 ซึ่งเป็นมาตรฐานที่ว่าด้วยภาษาในการเขียนโปรแกรม ได้แบ่งมาตรฐานภาษาต่าง ๆ ออกเป็น 5 แบบ อันได้แก่

2.2.3.1 LD (Ladder Diagram)

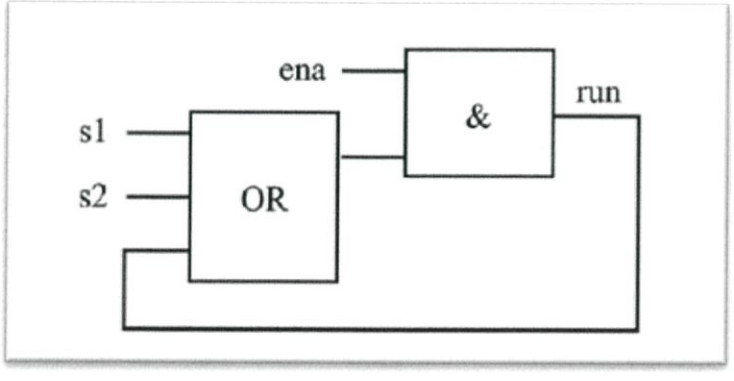
เป็นภาษาที่เขียนอยู่ในรูปกราฟิก ซึ่งมีพื้นฐานมาจากวงจรควบคุมแบบรีเลย์ และวงจรไฟฟ้า ใช้งานได้ง่าย โดยที่แลดเดอร์ไดอะแกรมจะประกอบด้วยราง (Rail) ทั้งซ้าย และขวาของไดอะแกรม เพื่อใช้สำหรับเชื่อมต่ออุปกรณ์ที่เป็นสวิตช์หน้าสัมผัส ซึ่งจะเป็นทางผ่านของกระแสและมีขดลวดหรือคอยล์เป็นเอาต์พุต ดังแสดงในรูปที่ 2.13



รูปที่ 2.13 ภาษา Ladder Diagram

2.2.3.2 FBD (Function Block Diagram)

เป็นภาษาที่เขียนในรูปแบบกราฟิกเช่นเดียวกัน และเชื่อมต่อกันเป็นโครงข่าย โดยที่มีพื้นฐานมาจากหลักการของลอจิกไดอะแกรม (Logic Diagram) ดังแสดงในรูปที่ 2.14



รูปที่ 2.14 ภาษา Function Block Diagram

2.2.3.3 IL (Instruction List)

เป็นภาษาที่เขียนในรูปข้อความจะมีลักษณะคล้ายกับภาษาแอสแซมบลี้ (Assembly) และภาษาเครื่อง (Machine Code) ดังแสดงในรูปที่ 2.15

```
Label:   LD      a1      (* result := a1 *)
         ADD(    a2      (* delayed ADD, result := a2 *)
         MUL(    a3      (* delayed MUL, result := a3 *)
         SUB     a4      (* result := a3 -a4 *)
         )               (* execute delayed MUL, *)
                         (* resule := a1+ (a2*(a3  - a4)  * a5) *)
         ADD     a6      (* a1 + (a2 * (a3 - a4) * a5) + a6 *)
         ST      res     (* store current result in res *)
```

รูปที่ 2.15 ภาษา Instruction List

2.2.3.4 ST (Structure Text)

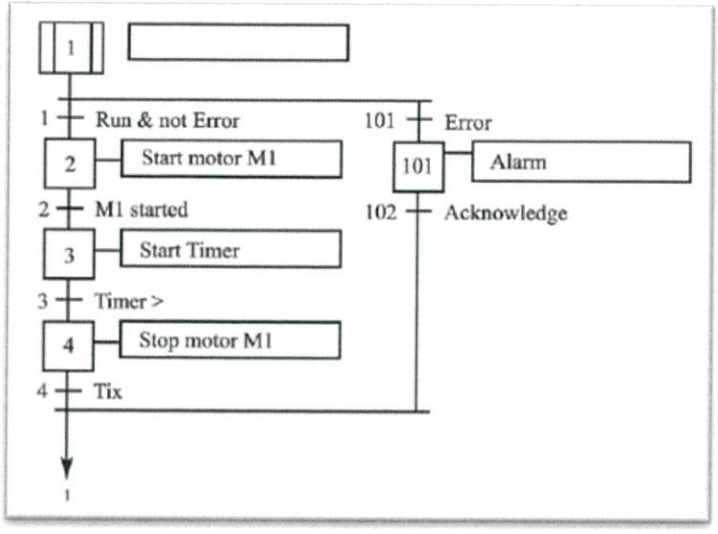
เป็นภาษาระดับสูง โดยมีพื้นฐานมาจากภาษาปาสคัล (Pascal) ซึ่งประกอบด้วยนิพจน์ และคำสั่ง โดยคำสั่งทั่วไปจะเกี่ยวกับการเลือกการทำงาน เช่น IF...THEN...ELSE และคำสั่งที่เกี่ยวข้องกับการทำงานซ้ำ เช่น FOR, WHILE เป็นต้น ดังแสดงในรูปที่ 2.16

```
D := B*B  - 4*A*C ;
IF D  < 0.0 THEN Nroots := 0 ;
ELSIF D = 0.0 THEN
          Nroots:=1 ;
          X1  :=  -B/(2.0*A) ;
ELSE Nroots := 2;
          X1 := (-B+sqrt(D))/(2.0*A) ;
          X2 := (-B-sqrt(D))/(2.0*A) ;
END_IF
```

รูปที่ 2.16 ภาษา Structure Text

2.2.3.5 SFC (Sequential Flow Chart)

เป็นภาษาที่รองรับการเขียนโปรแกรมที่มีโครงสร้างการทำงานแบบเป็นลำดับหรือซีเควนซ์ ซึ่งจะประกอบด้วย การปฏิบัติการย่อย (Step) และเงื่อนไขที่กำหนดให้ปฏิบัติงานตามคำสั่งย่อย(Transition) นอกจากนี้ยังสามารถกำหนดลักษณะการทำงานเป็นแบบ Liner, Alternative และ Parallel Step Sequence ได้ด้วย ดังแสดงในรูปที่ 2.17



รูปที่ 2.17 ภาษา Sequential Flow Chart

2.3 ความรู้พื้นฐานเกี่ยวกับระบบ SCADA

2.3.1 ความหมายของ SCADA

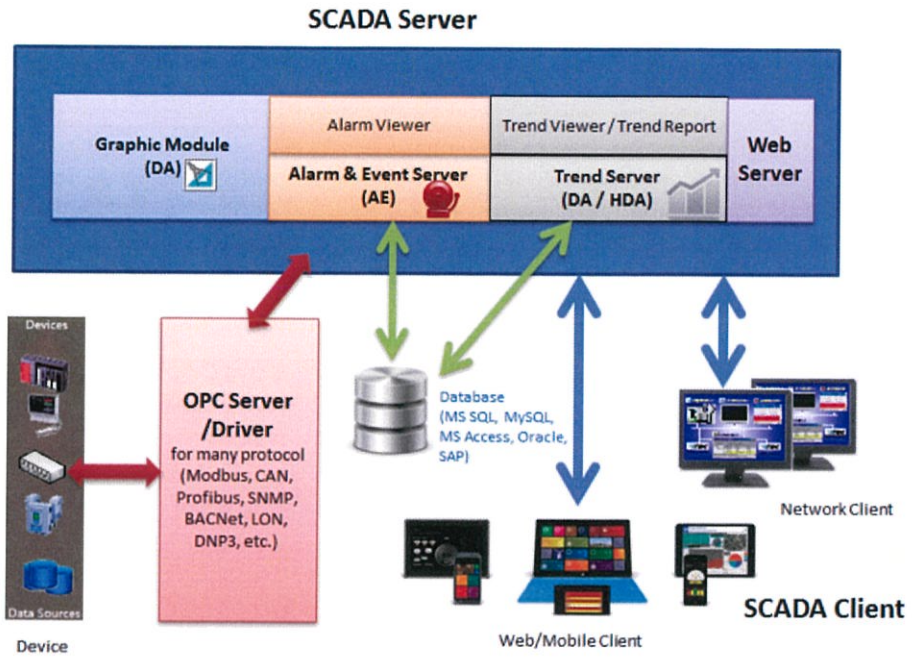
สกาดา (SCADA) ย่อมาจาก Supervisory Control and Data Acquisition เป็นระบบตรวจสอบและวิเคราะห์ข้อมูลแบบ Real-time ใช้ในการตรวจสอบสถานะ ตลอดจนควบคุมการทำงานของระบบควบคุมในอุตสาหกรรม และงานวิศวกรรมต่าง ๆ เช่น งานด้านโทรคมนาคมสื่อสาร การประปา การบำบัดน้ำเสีย การจัดการด้านพลังงาน อุตสาหกรรมการกลั่นน้ำมันและก๊าซ อุตสาหกรรมเคมี อุตสาหกรรมประกอบรถยนต์ การขนส่ง กระบวนการนิวเคลียร์ในโรงไฟฟ้า เป็นต้น ตัวอย่างการใช้งาน เช่น ใช้ SCADA ตรวจสอบข้อมูลการรั่วไหลของของเหลวที่เกิดขึ้นในท่อขนส่งจากตัวตรวจจับ แล้วส่งสัญญาณแจ้งเตือนให้พนักงานทราบ โดยส่งข้อมูลสู่ส่วนกลางของระบบ SCADA เป็นต้น นอกจากนั้น SCADA อาจทำหน้าที่คำนวนและประมวลผลข้อมูลที่ได้จากฮาร์ดแวร์ต่าง ๆ เช่น PLC, Controller, DCS, RTU แล้วแสดงข้อมูลทางหน้าจอ หรือส่งสัญญาณควบคุมฮาร์ดแวร์ดังกล่าว เช่น หากอุณหภูมิของอุปกรณ์สูงเกินพิกัด ให้ทำการปิดอุปกรณ์นั้น เป็นต้น โดยสั่งงานผ่าน PLC หรือ Controller ที่ติดต่ออยู่ ดังรูปที่ 2.18

ทั้งนี้ SCADA สามารถเก็บรวบรวมข้อมูลที่ได้จากระบบควบคุมทั้งหมดไว้ในฐานข้อมูลเพื่อให้พนักงาน หรือโปรแกรมอื่น ๆ สามารถนำไปใช้งานได้ SCADA ได้เข้าไปมีส่วนในงานควบคุมทั้งขนาดเล็กและใหญ่ที่ต้องการแสดงผล แลกเปลี่ยนข้อมูล หรือควบคุมระบบต่างๆ จากส่วนกลาง เพื่อการทำงานของระบบรวมที่สัมพันธ์กัน มองเห็นภาพรวมได้อย่างชัดเจน และมีความรวดเร็วต่อเหตุการณ์ต่าง ๆ ที่เกิดขึ้น ระบบ SCADA ในปัจจุบันมีความสามารถในการสื่อสาร ควบคุม และประมวลผลข้อมูลจาก I/O ของอุปกรณ์เช่น PLC, DCS, RTU ได้ถึงระดับที่เกินหนึ่งแสน I/O และได้รับการพัฒนาให้มีความสามารถรองรับความต้องการใหม่ ๆ ของผู้ใช้งานอย่างต่อเนื่องตลอดมา

รูปที่ 2.18 ลักษณะของระบบควบคุม โดยใช้ระบบ SCADA

2.3.2 โครงสร้างของระบบ SCADA

SCADA โดยทั่วไปประกอบด้วยส่วนรับข้อมูลจากอุปกรณ์ (Device) ต่าง ๆ ที่เรียกว่า OPC Server หรือ OPC Driver ในปัจจุบันนิยมใช้ OPC Server มากกว่า เนื่องจาก มีมาตฐานที่ไม่ขึ้นกับค่ายหรือยี่ห้อ ซ้ำยังมีให้เลือกซื้อจากผู้ผลิตจำนวนมาก และมีองค์กรณกลาง คือ OPC Foundation เป็นผู้ดูแลตรวจสอบมาตรฐานและพัฒนามาตรฐานของ OPC ซึ่งย่อมาจาก OLE for Process Control เมื่อ OPC Server ซึ่งเป็นซอฟต์แวร์รับส่งข้อมูลระหว่าง Device ส่งให้ซอฟต์แวร์ SCADA เพื่อนำไปแสดงผลในแบบกราฟิก เก็บข้อมูล บันทึกค่าสัญญาณ รวมถึงมีการแจ้งเตือน ดังรูปที่ 2.19



รูปที่ 2.19 โครงสร้างของระบบ SCADA

โมดูลกราฟิก (Graphic Module) จะรับข้อมูลจาก OPC Server หรือ OPC Driver เพื่อนำไปแสดงผลแบบ Real Time ข้อมูลในส่วนนี้เรียกว่า DA (Data Access)

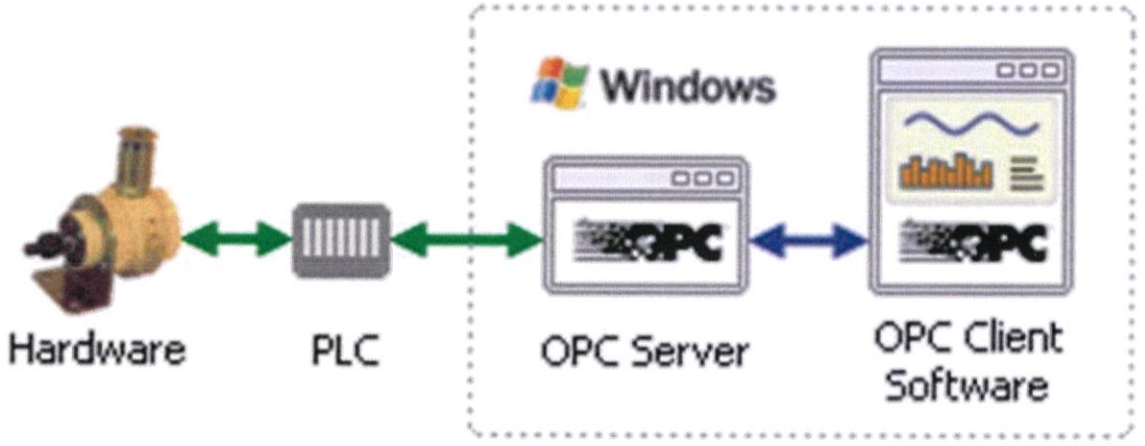
โมดูลตรวจจับ และแจ้งเตือน (Alarm) ก็รับค่าแบบ DA จาก OPC Server เช่นกัน จากนั้นนำไปประมวลผลเป็นการแจ้งเตือนที่เรียกว่า AE (Alarm and Event) และเก็บข้อมูลแจ้งเตือนไว้แสดงผลย้อนหลังซึ่งใช้ในการวิเคราะห์ความขัดข้องต่าง ๆ ที่เกิดขึ้นซึ่งเป็นส่วน สำคัญที่ใช้คำนวณค่า Down Time หรือ OLE ต่อไป

โมดูลเก็บข้อมูล และพล็อตสัญญาณ (Trend) รับข้อมูล DA จาก OPC Server ในส่วนหนึ่งมาแสดงผลแบบ Real Time และอีกส่วนหนึ่งเก็บไว้ในฐานข้อมูลเพื่อนำกลับมาแสดงผลย้อนหลังได้ เรียกว่าทำหน้าที่เป็น HDA Server (History Data Access Server)

SCADA ที่ประกอบด้วย 3 โมดูลข้างต้นเป็นหลัก จะสามารถบริการข้อมูลให้กับ Client แบบต่าง ๆ เพื่อนำไปแสดงผลและสั่งการจากผู้ที่เกี่ยวข้อง และหากมีความสามารถในการแสดงผลผ่านเว็บแล้วก็จำเป็นต้องมี Web Server บริการข้อมูลใน SCADA Server ด้วยระบบ SCADA ยังสามารถทำงานในรูปแบบยระบบเครือข่าย SCADA ขนาดใหญ่ขึ้นได้เช่น SCADA Server หนึ่งสามารถรวบรวมข้อมูลจาก SCADA Server ย่อยอื่น ๆ ผ่าน OPC Server เป็นต้น

2.4 ความสำคัญของ OPC

OPC ย่อมาจาก "Object Linking and Embedding for Process Control" เป็นมาตรฐานที่ใช้ในการติดต่อสื่อสารกันของซอฟต์แวร์ ซึ่งทำให้ซอฟต์แวร์บนคอมพิวเตอร์สามารถสื่อสารกับอุปกรณ์ทางอุตสาหกรรมต่าง ๆ ได้ โดยการใช้งานจะอยู่ในรูปแบบของ Server และ Client กล่าวคือ OPC Server เป็นซอฟต์แวร์ที่แปลงโปรโตคอลที่ใช้ในการสื่อสารของฮาร์ดแวร์ คือ PLC มาเป็นโปรโตคอลของ OPC ส่วน OPC Client เป็นซอฟต์แวร์ที่ต้องการใช้ข้อมูลจาก OPC Server ในการรับหรือส่งข้อมูลกับฮาร์ดแวร์ต่าง ๆ ซึ่งรูปแบบการติดต่อสื่อสารโดยใช้ OPC เป็นตัวกลาง ดังแสดงในรูปที่ 2.20



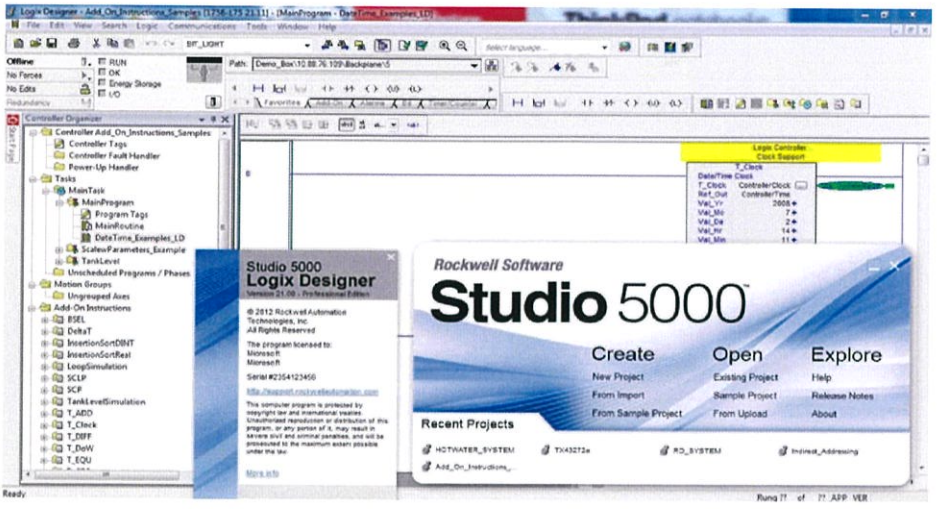
รูปที่ 2.20 การติดต่อสื่อสารระหว่าง PLC และ OPC Client ผ่านทาง OPC Server

OPC เป็นมาตรฐานกลาง (Open Standard) ที่เปิดกว้างแก่ผู้ใช้งาน จึงช่วยลดต้นทุนและเพิ่มทางเลือกให้ผู้ผลิตอุปกรณ์ สามารถพัฒนาระบบสื่อสารข้อมูลของตนให้เป็นไปตามมาตรฐานเดียวกันได้ ทั้งนี้การประยุกต์ใช้ส่วนใหญ่จะเป็นการช่วยให้อุปกรณ์ต่างค่ายสามารถแลกเปลี่ยนข้อมูลกันได้ จึงทำให้การรวบรวมข้อมูลจากอุปกรณ์ที่หลากหลายทั้งในเชิงข้อมูลและรูปแบบมาตรฐาน สามารถทำได้ง่ายขึ้น บางครั้งอาจใช้วิธีการนำข้อมูลจากอุปกรณ์ไปเก็บรวบรวมในระบบฐานข้อมูล เช่น SQL Server ไว้ก่อน จากนั้นค่อยให้ระบบบริหารจัดการทรัพยากร เช่น HMI หรือ SCADA มานำข้อมูลไปใช้อีกที ซึ่งก็มีข้อเด่นข้อด้อยแตกต่างกันไป ทั้งนี้ขึ้นอยู่กับรูปแบบการใช้ข้อมูลที่ต้องการ

2.5 ซอฟต์แวร์ที่เกี่ยวข้อง

2.5.1 Studio 5000 Logix Designer

Studio 5000 Logix Designer เป็นซอฟต์แวร์สำหรับตั้งค่าและสร้างโปรแกรม เพื่อใช้งานตัวควบคุม ยี่ห้อ Allen-Bradley ตระกูล Logix 5000 และมีหน้าตาของซอฟต์แวร์ ดังรูปที่ 2.21



รูปที่ 2.21 ลักษณะของซอฟต์แวร์ Studio 5000 Logix Designer

ด้วยคุณสมบัติที่ใช้งานง่าย จึงเพิ่มความสะดวกให้แก่ผู้ใช้งานในการออกแบบระบบควบคุม ประหยัดเวลาในการดำเนินงาน และช่วยเพิ่มประสิทธิภาพในการผลิตให้มากยิ่งขึ้น และเครื่องคอมพิวเตอร์ที่จะใช้งานโปรแกรมควรมีคุณสมบัติ ดังตารางที่ 2.1

ตารางที่ 2.1 คุณสมบัติของอุปกรณ์ที่พร้อมใช้งานโปรแกรม Studio 5000 Logix Designer

ความต้องการด้านฮาร์ดแวร์	ความต้องการด้านซอฟต์แวร์ระบบปฏิบัติการ
• คอมพิวเตอร์ประมวลผลด้วยซีพียู Intel Core i5 หรือสูงกว่า • หน่วยความจำ RAM 8 GB ขึ้นไป • พื้นที่ว่างที่ต้องการบนฮาร์ดดิสก์ 20 GB • กราฟิก ทำงานได้ดีกับซอฟต์แวร์ DirectX 9	• Microsoft Windows 7 Professional (64-bit) Service Pack 1 • Microsoft 7 Home Premium (32-bit) Service Pack 1 • Microsoft Windows Server 2008 R2 Standard Edition Service Pack 1 • Microsoft Windows 8 Professional (64-bit) • Microsoft Windows 8.1 Professional (64-bit) • Microsoft Windows Server 2012 Standard Edition

2.5.2 Wonderware InTouch HMI 2014 R2

Wonderware InTouch HMI 2014 R2 เป็นซอฟต์แวร์ HMI จากบริษัท Invensys Process Systems ใช้สำหรับสร้างกราฟิกในอุตสาหกรรม ดังแสดงในรูป 2.22 เป็นโปรแกรมที่เขียนจะแสดงกระบวนการต่าง ๆ ในระบบควบคุมผ่านหน้าจอ HMI โดยที่ผู้ใช้หรือผู้ควบคุมไม่จำเป็นต้องอยู่ที่หน้างาน เพื่อสังเกตกระบวนการ ทำให้เกิดความสะดวกต่อผู้ดูแล

รูปที่ 2.22 โลโก้ซอฟต์แวร์ Wonderware InTouch HMI 2014 R2

เนื่องจาก HMI หรือ SCADA เกิดจากความต้องการของผู้ใช้งานในการเข้าไปควบคุมระบบที่มี PLC เป็นตัวควบคุมอยู่ ในขณะที่ PLC มีการควบคุมกระบวนการโดยอัตโนมัติ และอยู่กระจัดกระจาย ยากที่จะนำข้อมูลแต่ละส่วนมาประมวลผลร่วมกันด้วยการเก็บข้อมูลแบบ Manual แต่ HMI หรือ SCADA จะนำข้อมูลจาก PLC ส่งผ่านโครงข่ายการสื่อสารแบบต่าง ๆ และรวบรวมข้อมูลเข้าด้วยกัน ทำให้สามารถสั่งการได้โดยผู้เชี่ยวชาญ และเครื่องคอมพิวเตอร์ที่จะใช้งานโปรแกรมควรมีคุณสมบัติดังแสดงในตารางที่ 2.2

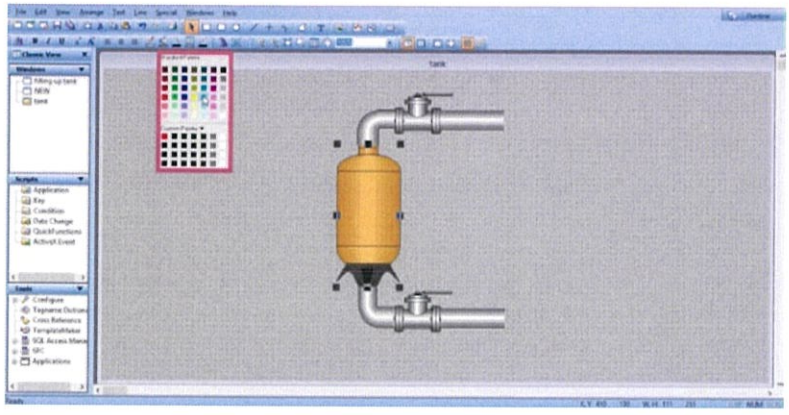
ตารางที่ 2.2 คุณสมบัติของอุปกรณ์ที่พร้อมใช้งานโปรแกรม Wonderware InTouch HMI 2014 R2

ความต้องการของระบบ Microsoft SQL Server	ความต้องการด้านซอฟต์แวร์ ระบบปฏิบัติการ
• Microsoft SQL 2012 SP2 Express, Standard and Enterprise (32/64-bit) • Microsoft SQL 2014 Express, Standard and Enterprise (NSP, SP1, SP2) (32/64-bit) • Microsoft SQL 2014 SP1 Express, Standard and Enterprise(32/64-bit) • Microsoft SQL 2016 Express (No SP, SP1), Standard and Enterprise(32/64-bit)	• Microsoft SQL 2012 SP2 Express, Standard and Enterprise (32/64-bit) • Microsoft SQL 2014 Express, Standard and Enterprise (NSP, SP1, SP2) (32/64-bit) • Microsoft SQL 2014 SP1 Express, Standard and Enterprise(32/64-bit) • Microsoft SQL 2016 Express (No SP, SP1), Standard and Enterprise(32/64-bit)

Wonderware InTouch HMI 2014 R2 สามารถทำงานบนระบบปฏิบัติการ Windows ของ Microsoft ได้ และประกอบด้วย 2 โปรแกรมที่สำคัญ ที่ใช้สำหรับสร้างหน้าต่าง HMI อันได้แก่

2.5.2.1 InTouch WindowMaker

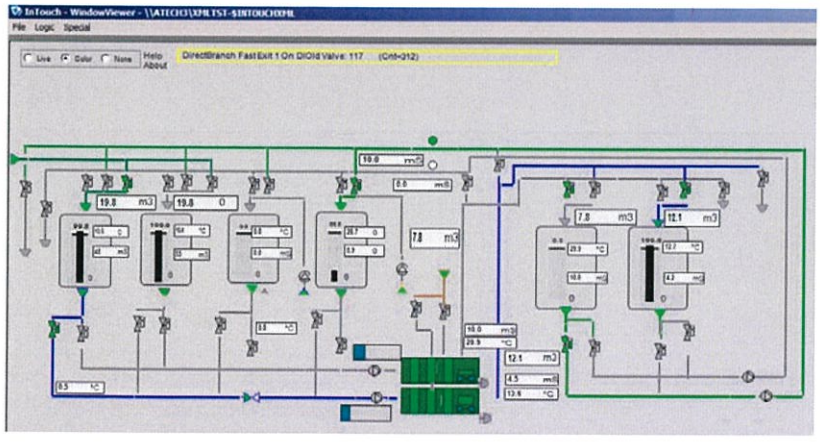
เป็นโปรแกรมสร้างที่ใช้สำหรับสร้าง และแก้ไขกราฟิก ดังรูปที่ 2.23 โดยมีเครื่องมือในการวาดกราฟิก เขียนสคริป อีกทั้งยังมี Factory Symbol สำเร็จรูปที่สามารถนำมาใช้ได้อย่างสะดวก นอกจากนี้ ยังใช้ในการกำหนดค่าให้สามารถเชื่อมต่อ Industrial I/O Systems กับ Microsoft Windows Applications อื่น ๆ ได้



รูปที่ 2.23 ลักษณะของซอฟต์แวร์ Intouch WindowMaker

2.5.2.2 InTouch WindowViewer

เป็นหน้าต่างแสดงผลเมื่อ Runtime จาก WindowMaker ดังรูปที่ 2.24 สามารถบันทึกข้อมูลที่ผ่านมาและรายงานผล ตลอดจนแสดงสัญญาณเตือน เมื่อเกิดความผิดปกติขึ้นกับกระบวนการตามที่โปรแกรมไว้ จึงช่วยลดการสูญเสียจากอุบัติเหตุได้



รูปที่ 2.24 ลักษณะของซอฟต์แวร์ Intouch WindowViewer

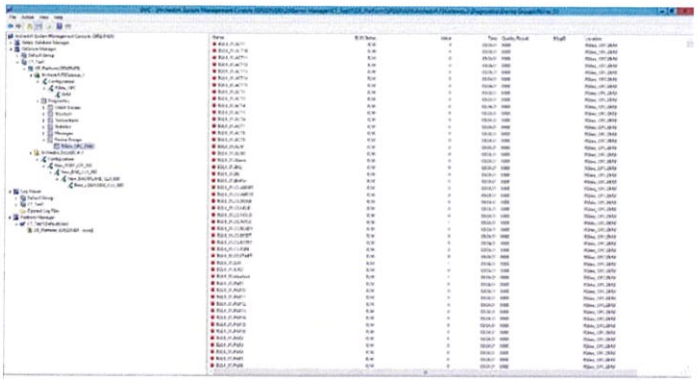
2.5.3 ArchestrA IDE (Integrated Development Environment)

ArchestrA IDE ดังรูปที่ 2.25 นั้นทำหน้าที่เป็น System Platform กล่าวคือ เป็นระบบที่ใช้เก็บวัตถุ (Object) ต่าง ๆ ที่ผู้ใช้สร้างขึ้นในรูปแบบของข้อมูล หรือกราฟฟิก ซึ่งประกอบไปด้วยกราฟิกและคำสั่งที่เกี่ยวข้อง ซึ่งระบบนี้สามารถขยายเพิ่มเติมได้ตามการออกแบบของผู้ใช้งาน โดย ArchestrA IDE สามารถทำได้ทั้งการออกแบบ พัฒนา และเก็บวัตถุ เพื่อ Deploy หรือส่งวัตถุไปใช้ได้ในคอมพิวเตอร์ที่ต้องการซึ่งอยู่ใน Network เดียวกัน และด้วยคุณสมบัตินี้เอง จึงมีประโยชน์ต่อการพัฒนาแอปพลิเคชันขนาดใหญ่ ที่มีวัตถุในลักษณะเดียวกันเป็นจำนวนมาก ลดความซ้ำซ้อนในการแก้ไขคุณสมบัติของวัตถุทีละตัว การจัดการวัตถุจึงมีความสะดวกและรวดเร็ว

รูปที่ 2.25 ลักษณะของซอฟต์แวร์ ArchestrA IDE

2.5.4 SMC (System Management Console)

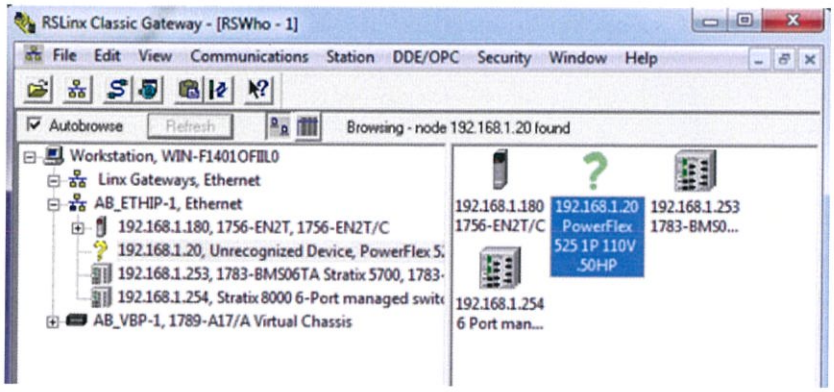
SMC ดังรูปที่ 2.26 เป็นโปรแกรมที่เป็นตัวกลางสำหรับเชื่อมต่อซอฟต์แวร์ต่าง ๆ ให้สามารถติดต่อสื่อสารกันได้ หรือเป็น OPC Server โดยสามารถเชื่อมต่อโปรแกรมจาก PLC กับส่วนแสดงผลกราฟิกได้ ด้วยการตั้งค่า Topic Name ให้สอดคล้องกันทั้ง Server และ Client นอกจากนี้ SMC ยังสามารถวิเคราะห์ความผิดพลาด และยังเป็นเครื่องมือในการจัดการเครือข่ายคอมพิวเตอร์



รูปที่ 2.26 ลักษณะของซอฟต์แวร์ SMC

2.5.5 RSLinx Classic

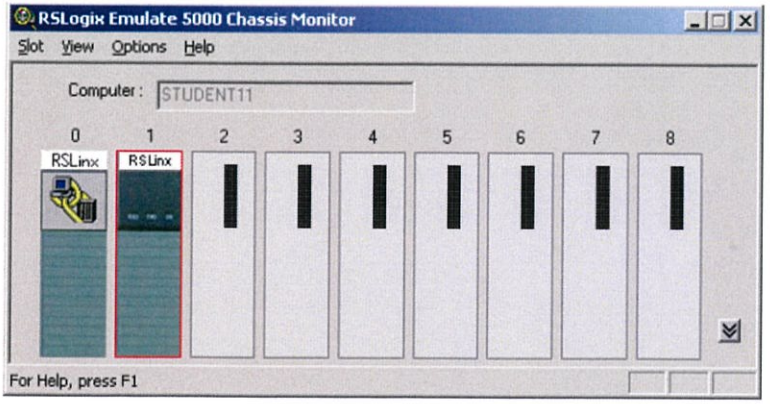
เป็นซอฟต์แวร์ที่ใช้เชื่อมต่อพีแอลซีกับคอมพิวเตอร์ ดังรูปที่ 2.27 เพื่อดาวน์โหลดโปรแกรมที่มีอยู่ในคอมพิวเตอร์ให้พีแอลซี หรืออัพโหลดโปรแกรมจากพีแอลซีขึ้นมาตรวจสอบแก้ไขในคอมพิวเตอร์ และสารมารถสั่ง Run หรือ Stop พีแอลซี หรือทดสอบการทำงานของโปรแกรมที่สร้างขึ้น เป็นต้น ทั้งนี้จะต้องมีไดรเวอร์ซึ่งเป็นโปรแกรมย่อยสำหรับควบคุมอุปกรณ์ภายนอกที่นำมาเชื่อมต่อกับคอมพิวเตอร์หรือเรียกว่า โปรเซสเซอร์ ซึ่งได้แก่พีแอลซีนั่นเอง ไดรเวอร์ที่ใช้ขึ้นอยู่กับคุณสมบัติทางกายภาพของการเชื่อมต่อระหว่างคอมพิวเตอร์กับโปรเซสเซอร์ เช่น RS-232, RS-485, RS-422 หรือ Ethernet เป็นต้น



รูปที่ 2.27 ลักษณะของซอฟต์แวร์ RSLinx Classic

2.5.6 RSLogix Emulate 5000 Chassis Monitor

RSLogix Emulate 5000 ดังรูปที่ 2.28 เป็นโปรแกรมจำลองตัวควบคุมของยี่ห้อ Allen-Bradley ในตระกูล Logix 5000 ซึ่งได้แก่ ControlLogix, CompactLogix, FlexLogix, SoftLogix5800 และ DriveLogix เพื่อจำลองฟังก์ชันการทำงานของพีแอลซีโดยไม่ต้องใช้พีแอลซีจริง จึงมีประโยชน์ต่อการทดสอบและแก้ไขโปรแกรมก่อนการทำ Commissioning และ Startup กระบวนการ



รูปที่ 2.28 ลักษณะของซอฟต์แวร์ RSLogix Emulate 5000 Chassis Monitor

บทที่ 3
การออกแบบระบบควบคุมของกระบวนการบรรจุอาหารสัตว์

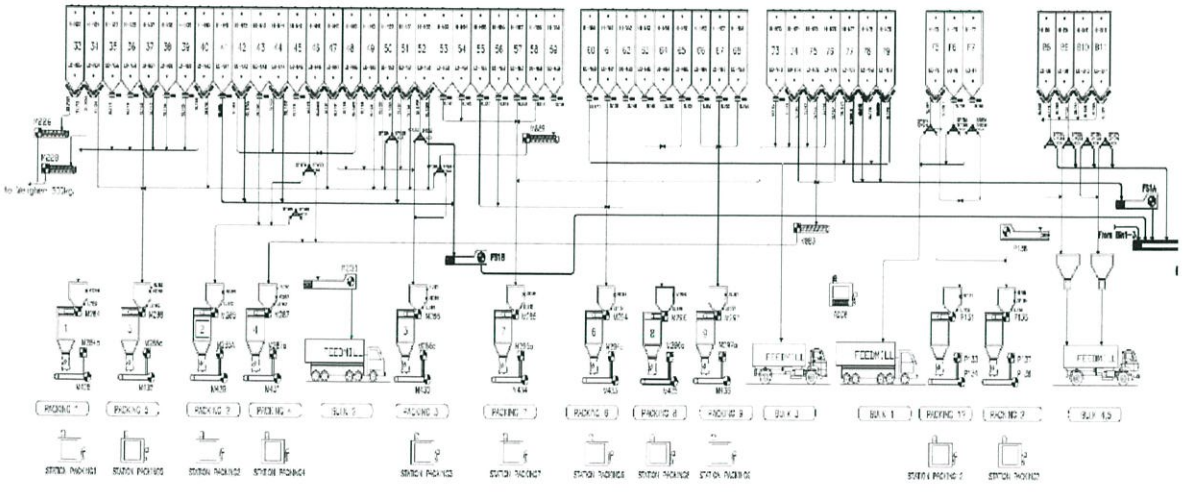
ในการออกแบบระบบควบคุมและส่วนแสดงผลของกระบวนการบรรจุอาหารสัตว์ จะต้องเริ่มจากการศึกษากระบวนการให้เข้าใจว่ามีขั้นตอนการทำงานอย่างไร เครื่องจักรที่เกี่ยวข้องแต่ละชนิดมีคุณสมบัติและข้อจำกัดใดบ้าง เพื่อให้สามารถออกแบบโปรแกรมได้อย่างถูกต้อง โดยมีการทดลองฟังก์ชันการทำงานของระบบควบคุมด้วยซอฟต์แวร์จำลองพีแอลซี ตรวจสอบ และแก้ไขจนได้โปรแกรมควบคุม และส่วนแสดงผลตรงตามความต้องการของลูกค้า ซึ่งวิธีการดำเนินงานดังกล่าวสามารถแบ่งเป็นขั้นตอนได้ ดังนี้

3.1 สรุปสิ่งจำเป็นสำหรับการออกแบบระบบควบคุม

ในการออกแบบระบบควบคุมของกระบวนการบรรจุอาหารสัตว์นี้ มีส่วนสำคัญที่ไว้ใช้พิจารณาในการออกแบบอยู่ 3 ส่วนหลัก ๆ ด้วยกัน อันได้แก่

3.1.1 P&ID (Piping and Instrumentation Diagram)

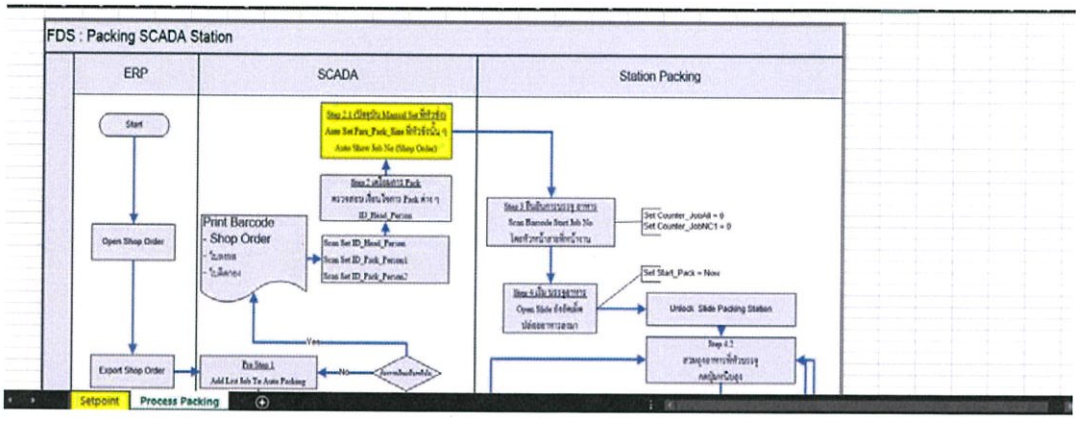
P&ID ดังรูปที่ 3.1 เป็นแผนภาพที่ใช้แสดงถึงรายละเอียดต่าง ๆ ของอุปกรณ์การวัดและควบคุม รวมถึงการเชื่อมต่อของอุปกรณ์แต่ตัวในกระบวนการนั้น ๆ ในรูปของสัญลักษณ์ ซึ่งทางบริษัทลูกค้าได้จัดเตรียมไว้ให้ เพื่อใช้ในการออกแบบกราฟฟิก ในหน้าจอควบคุม HMI ให้สอดคล้องกับ P&ID



รูปที่ 3.1 แผนภาพ P&ID ของกระบวนการบรรจุอาหารสัตว์

3.1.2 FDS (Functional Design Specification)

FDS ดังรูปที่ 3.2 คือ ข้อกำหนดในการออกแบบการทำงานของกระบวนการ เป็นสิ่งที่ใช้บอกว่าการทำงานของกระบวนการนี้ต้องมีขั้นตอนการทำงานอย่างไร ในส่วนนี้ทางบริษัทลูกค้าจะเป็นฝ่ายที่กำหนดความต้องการนี้ขึ้น อาจจะอยู่ในรูปของแผนภาพการไหล (Flowchart Diagram) หรืออาจจะเป็นแบบหัวข้อการทำงานทีละขั้นตอน ทั้งนี้จะขึ้นอยู่กับบริษัทของทางลูกค้า ซึ่งในการออกแบบครั้งนี้ ทางลูกค้าได้ให้ข้อมูลมาในรูปแบบของ Flowchart Diagram และนำข้อมูลในส่วนนี้มาใช้พิจารณาประกอบการเขียนโปรแกรมควบคุมด้วย PLC



รูปที่ 3.2 FDS ของกระบวนการบรรจุอาหารสัตว์

3.1.3 รายชื่ออุปกรณ์ที่ทำการควบคุม

ในส่วนนี้ทางบริษัทลูกค้าจะทำการจัดทำรายชื่อของอุปกรณ์ทั้งหมดที่ต้องการจะทำการควบคุมที่สอดคล้องกับ P&ID และ FDS ดังแสดงในรูปที่ 3.3 เพื่อให้ทางผู้ออกแบบนำไปใช้สร้าง Tag Name ให้กับ PLC และสร้าง Attribute ให้กับระบบ SCADA ในลำดับถัดไป

PACKING PROCESS

EQUIPMENT MODULE NAME	DEVICE SEQUENCE				TANK NUMBER	INTERLOCK WITH
PACKING 1	M428	M284a	M284	SL110	33	PACKING 5
				SL111	35	
				SL112	36	
				SL113	37	
				SL114	38	
				SL115	39	
				SL116	40	
PACKING 5	M432	M288a	M288	SL128	34	
				SL129	37	PACKING 3
				SL130	40	
				SL131	47	
				SL132	49	

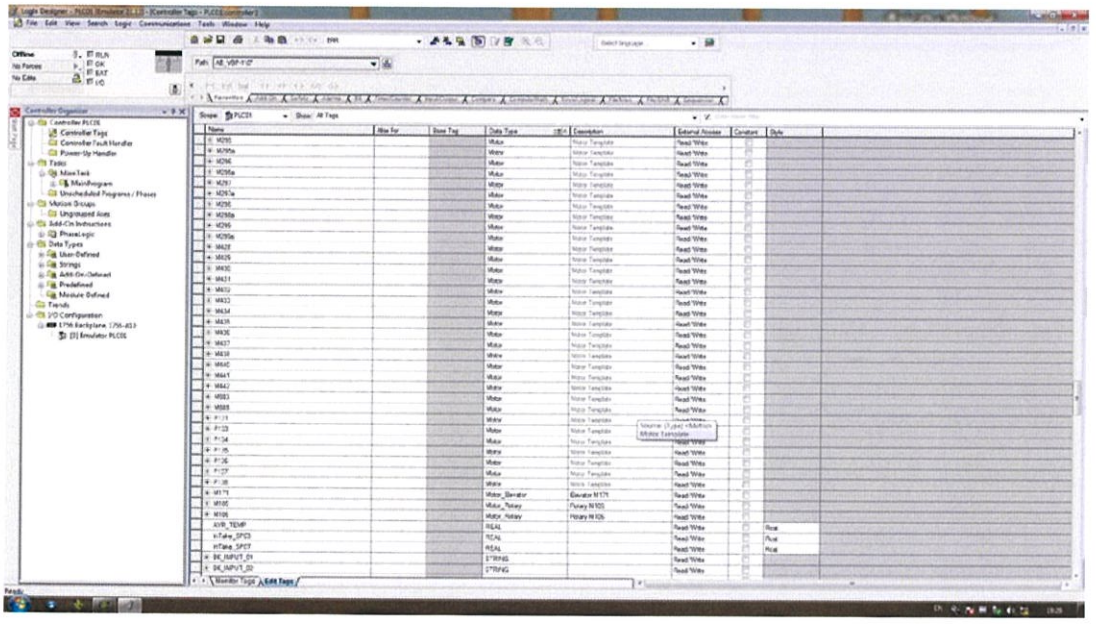
รูปที่ 3.3 รายชื่ออุปกรณ์ที่ใช้ในการควบคุมของกระบวนการบรรจุอาหารสัตว์

3.2 ทำการเขียนโปรแกรมควบคุม PLC

จากข้อมูลในหัวข้อ 3.1 เราจะนำมาใช้ในการเขียนโปรแกรมควบคุม PLC ซึ่งในหัวข้อนี้จะทำการใช้โปรแกรมควบคุม PLC คือ Studio 5000 Logix Designer ในการเขียนโปรแกรม โดยมีขั้นตอนในการทำด้วยกัน 4 ขั้นตอน ดังต่อไปนี้

3.2.1 สร้าง Tag Name ให้อุปกรณ์บน PLC

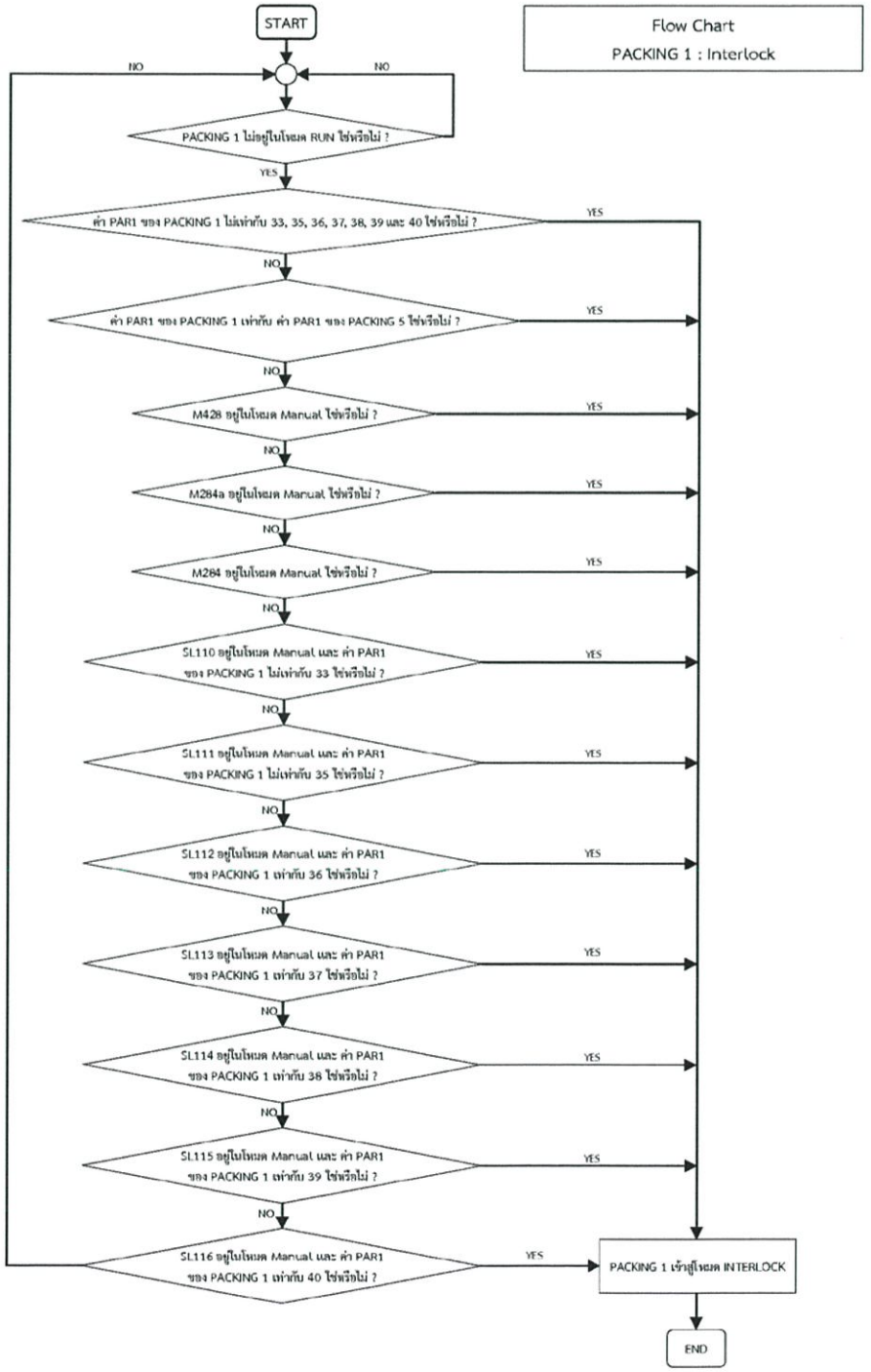
ในขั้นตอนแรกนี้ เราจะทำการสร้าง Tag Name ในซอฟต์แวร์ Studio 5000 Logix Designer ดังแสดงในรูปที่ 3.4 ให้อุปกรณ์ที่ต้องการจะควบคุม รวมถึงสวิตซ์ การแจ้งเตือน (Alarm) การที่ระบบเกิดการทำงานขัดแย้งซึ่งกันและกัน (Interlock) ซึ่งที่กล่าวมานั้น ล้วนจำเป็นจะต้องมี Tag Name เพื่อที่โปรแกรมจะได้เข้าถึงส่วนแสดงผลนั้น ๆ ได้ โดยนำข้อมูลรายชื่ออุปกรณ์ที่ทำการควบคุมมาใช้พิจารณาในการสร้าง Tag Name ให้สอดคล้องกัน



รูปที่ 3.4 รายชื่อ Tag Name ของอุปกรณ์บน PLC

3.2.2 ออกแบบ Flowchart Diagram

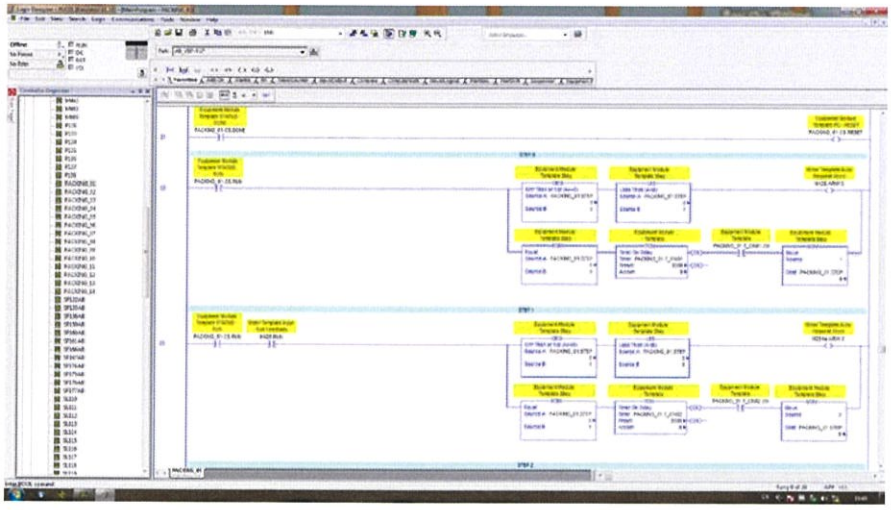
สำหรับการเขียนโปรแกรมควบคุม PLC จำเป็นจะต้องมีลำดับขั้นตอนในการออกแบบการทำงานของอุปกรณ์ทุกตัวเป็นลำดับ เพื่อให้เขียนการทำงานได้อย่างเป็นระบบ รวมถึงง่ายต่อการแก้ไขและตรวจสอบ ในขั้นตอนนี้จึงทำการสร้าง Flowchart Diagram เพื่อเป็นแนวทางในการออกแบบการทำงานอย่างเป็นระบบ ดังแสดงในรูปที่ 3.5



รูปที่ 3.5 ตัวอย่าง Flowchart Diagram สำหรับการเขียนโปรแกรมควบคุม PLC

3.2.3 เขียนโปรแกรมควบคุม PLC

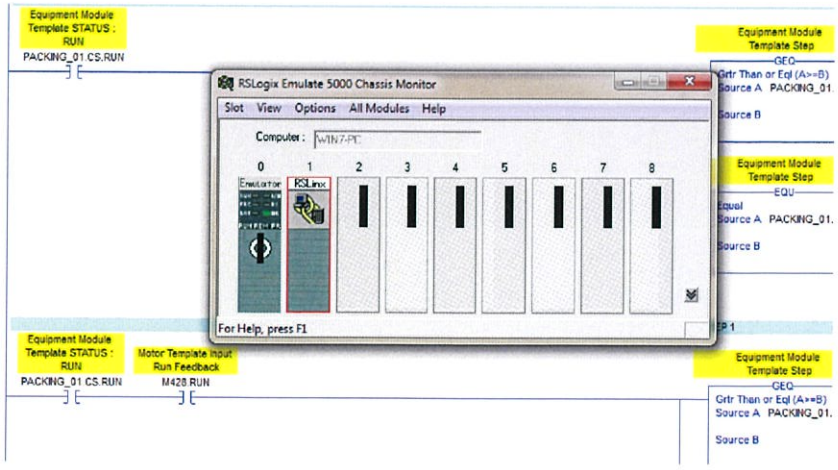
หลังจากที่ทำการออกแบบ Flowchart Diagram เสร็จ ก็เริ่มต้นเขียนโปรแกรมควบคุม PLC ตามกระบวนการที่ได้วางแผนไว้ใน Flowchart Diagram ข้างต้น ซึ่งในที่นี้ได้ทำการเขียนโปรแกรมควบคุม PLC โดยใช้ภาษา Ladder Diagram เนื่องจาก มีความสะดวก ง่ายต่อการอ่านกระบวนการ และการตรวจสอบแก้ไข ดังแสดงในรูปที่ 3.6



รูปที่ 3.6 ตัวอย่างการเขียนโปรแกรมควบคุม PLC

3.2.4 ทำการทดสอบระบบควบคุม PLC

หลังจากที่ทำการเขียนระบบควบคุม PLC เสร็จ จะต้องนำมาทดสอบความพร้อมสำหรับการทำงานว่าถูกต้อง และเป็นไปตาม Flowchart Diagram หรือไม่ โดยการทำการจำลอง PLC ขึ้นมาในโปรแกรม RSLogix Emulate 5000 Chassis Monitor และทำการ Download โปรแกรมควบคุม PLC ที่ได้เขียนไว้ ไปที่ตัว PLC ที่จำลองขึ้นมา ดังแสดงในรูปที่ 3.7 เพื่อทำการทดสอบ และแก้ไขจุดที่บกพร่องจนระบบสามารถทำงานได้ตามเป้าหมาย



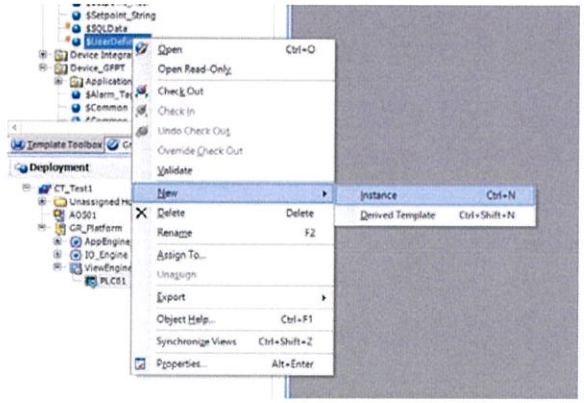
รูปที่ 3.7 การทดสอบระบบควบคุม PLC โดยใช้โปรแกรม RSLogix Emulate 5000 Chassis Monitor

3.3 ทำการออกแบบหน้าจอควบคุม HMI ผ่านระบบ SCADA

หลังจากที่ทำการเขียนโปรแกรมควบคุม PLC เสร็จแล้ว ก็มาทำในส่วนแสดงผลและควบคุมผ่านทางหน้าจอควบคุม HMI ผ่านระบบ SCADA ในส่วนของขั้นตอนนี้จะทำการใช้ซอฟต์แวร์ ArchestrA IDE และ Wonderware InTouch HMI 2014 R2 ในการออกแบบหน้าจอควบคุม HMI ผ่านระบบ SCADA ซึ่งแบ่งขั้นตอนออกเป็น 3 ขั้นตอน ดังต่อไปนี้

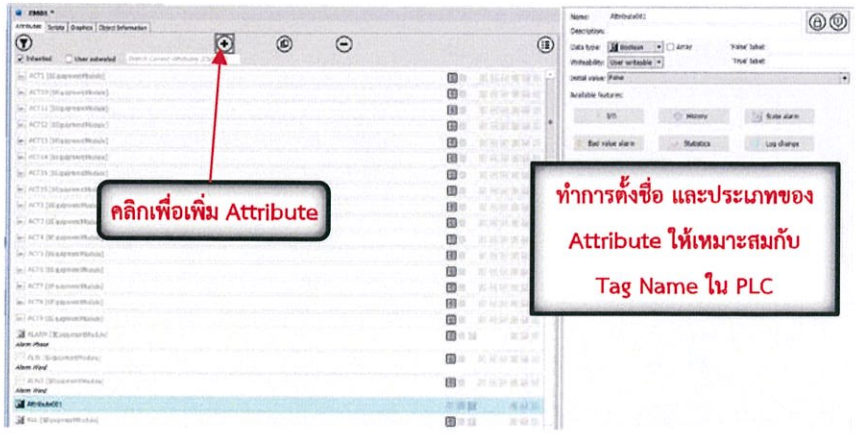
3.3.1 สร้าง Object และเพิ่ม Attribute ในระบบ SCADA

ก่อนที่จะทำการออกแบบกราฟฟิกนั้น ต้องทำการสร้างวัตถุ (Object) ดังแสดงในรูปที่ 3.8



รูปที่ 3.8 การสร้าง Object ในซอฟต์แวร์ ArchestrA IDE

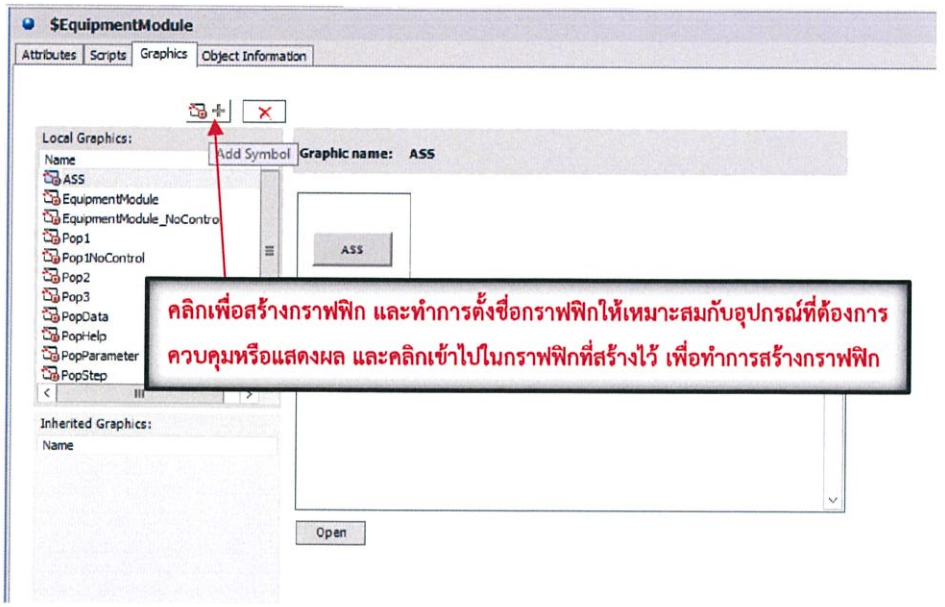
Object เปรียบได้กับตัวแทนของกระบวนการย่อย เพื่อสร้างที่อยู่ให้กับสมาชิก (Attribute) ดังแสดงในรูปที่ 3.9 ซึ่งเปรียบได้กับตัวแทนของอุปกรณ์ที่ต้องการควบคุมในกระบวนการย่อย ของระบบ SCADA โดยใช้ซอฟต์แวร์ ArchestrA IDE เพื่อที่จะทำการเชื่อมต่อระหว่าง OPC Server นั่นคือ Tag Name ของทางฝั่ง PLC และ OPC Client นั่นคือ Attribute ของทางฝั่งระบบ SCADA



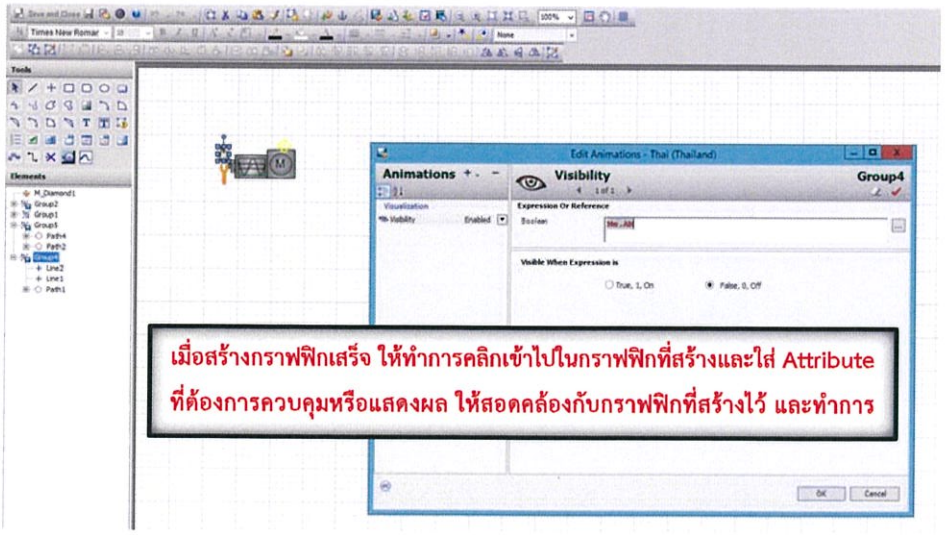
รูปที่ 3.9 การเพิ่ม Attribute ลงใน Object ในซอฟต์แวร์ ArchestrA IDE

3.3.2 ออกแบบกราฟฟิกลงใน Object

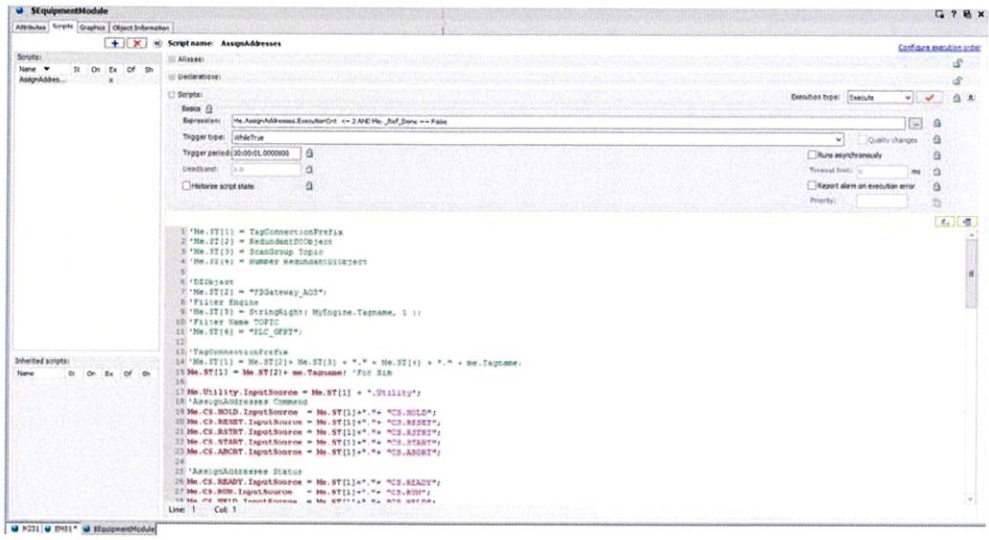
ในขั้นตอนนี้เราจะทำการออกแบบกราฟฟิกลงใน Object และฝัง Attribute ลงในกราฟฟิก เพื่อให้กราฟฟิกควบคุม และแสดงผลของ Attribute ต่าง ๆ ที่ต้องทำการควบคุม และแสดงผลออกมาทางหน้าจอควบคุม HMI พร้อมทั้งเขียนสคริปให้ Attribute เชื่อมโยงค่ากับ OPC Server โดยมีขั้นตอนการออกแบบตาม รูปที่ 3.10, รูปที่ 3.11 และ รูปที่ 3.12 ตามลำดับ



รูปที่ 3.10 การสร้างกราฟฟิกลงใน Object



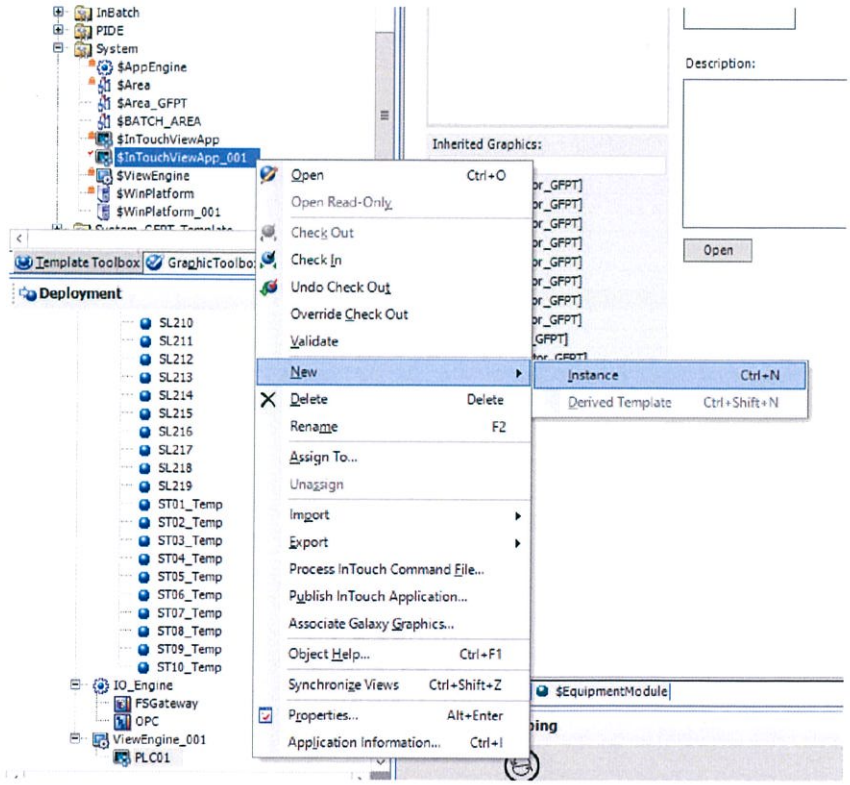
รูปที่ 3.11 การฝัง Attribute ลงในกราฟฟิก



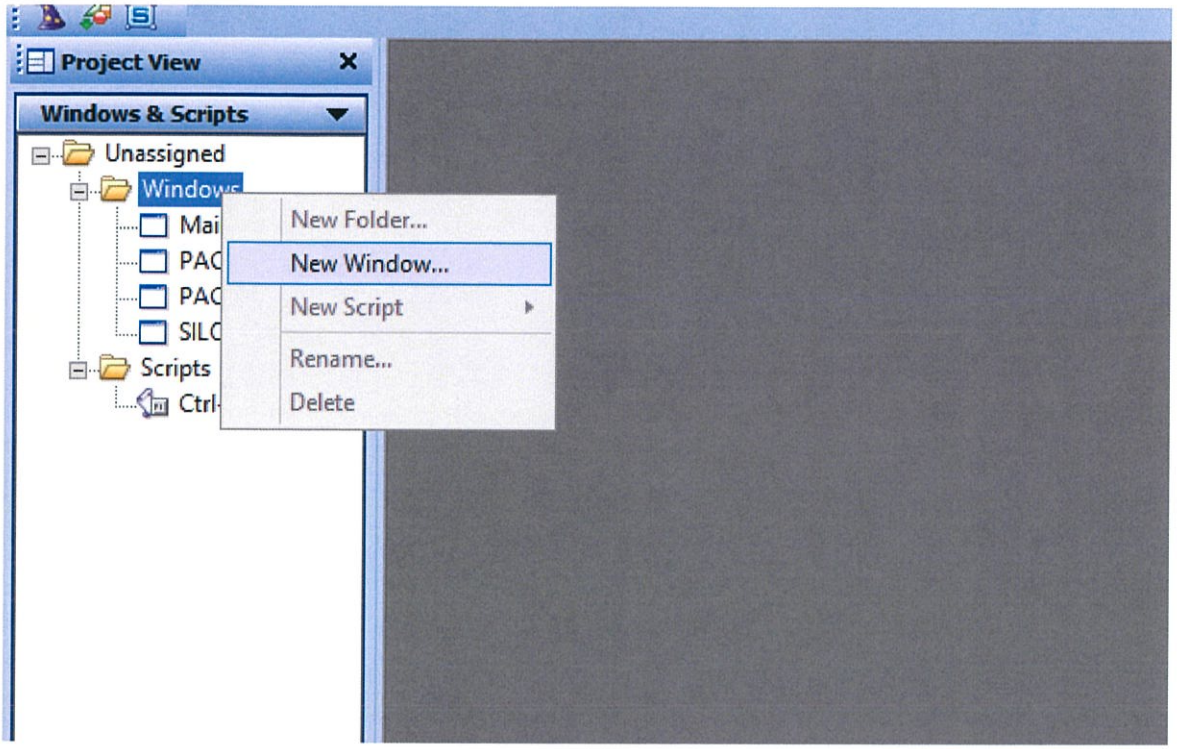
รูปที่ 3.12 การเขียน Script เพื่อให้ Attribute เชื่อมโยงค่ากับ OPC Server

3.3.3 ออกแบบหน้าจอควบคุม HMI

เมื่อทำการสร้างกราฟฟิกเสร็จเรียบร้อยแล้ว ก็ทำการสร้างหน้าจอควบคุมโดยใช้ซอฟต์แวร์ Wonderware InTouch HMI 2014 R2 โดยการสร้างไฟล์คัดลอกแม่แบบ (Instance) ขึ้นมาจากไฟล์ InTouch WindowMaker ที่อยู่ในซอฟต์แวร์ ArchestrA IDE และทำการออกแบบหน้าจอควบคุม โดยมีขั้นตอนการออกแบบตาม รูปที่ 3.13, รูปที่ 3.14 และ รูปที่ 3.15 ตามลำดับ



รูปที่ 3.13 การสร้าง Instance File จาก InTouch WindowMaker

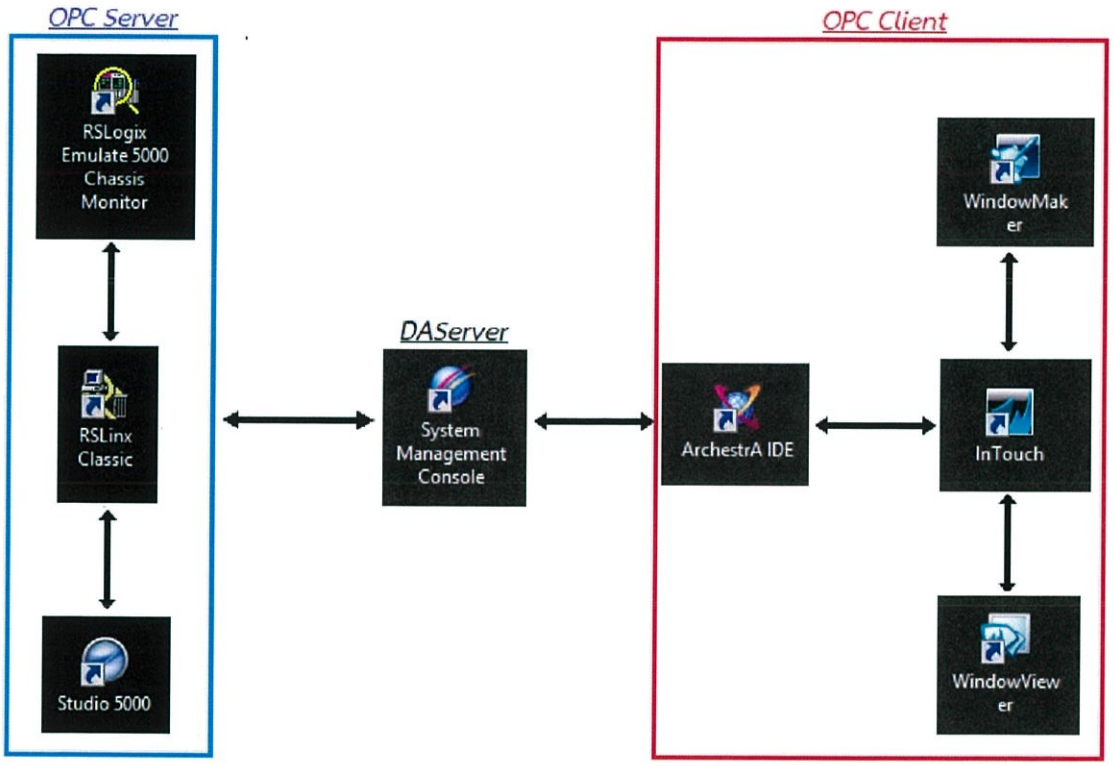


รูปที่ 3.14 การสร้าง Window เพื่อออกแบบหน้าจอควบคุม ในซอฟต์แวร์ InTouch WindowMaker

รูปที่ 3.15 การดึงกราฟฟิกใน Object มาใช้ในซอฟต์แวร์ InTouch WindowMaker

3.4 ทำการเชื่อมต่อระหว่าง OPC Server และ OPC Client

หลังจากที่ทำการเขียนโปรแกรมควบคุม PLC และออกแบบหน้าจอควบคุม HMI เสร็จเรียบร้อยแล้ว ในขั้นตอนนี้จะเป็นขั้นตอนการเชื่อมโยงข้อมูลระหว่าง PLC และระบบ SCADA เพื่อทำการสั่งการ PLC ผ่านทางหน้าจอควบคุม HMI ผ่านทาง DAServer (Data Access Server) ซึ่งเป็นส่วนกลางในการทำหน้าที่เชื่อมต่อระหว่าง OPC Server และ OPC Client เข้าด้วยกัน ซึ่งซอฟต์แวร์ที่ใช้ทำหน้าที่เป็น DAServer คือ ซอฟต์แวร์ ArchestrA SMC (System Management Console) โดยมีแผนผังอย่างคร่าว ๆ ดังแสดงในรูปที่ 3.16

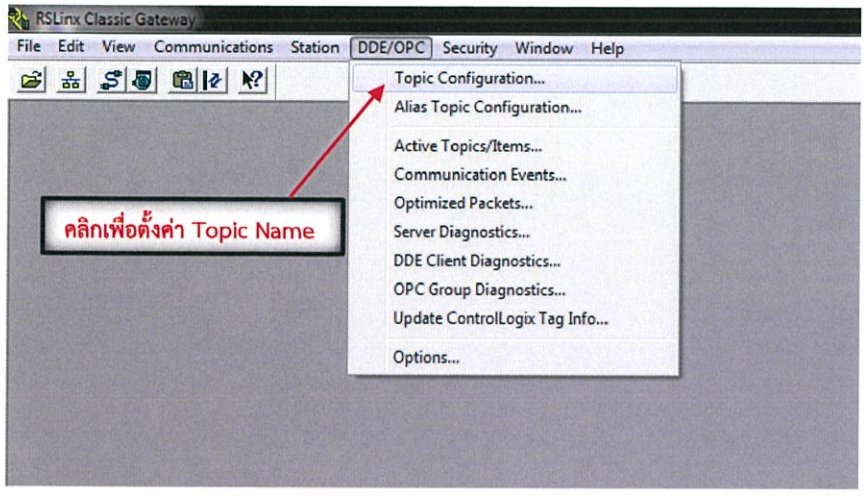


รูปที่ 3.16 แผนผังการเชื่อมต่อข้อมูลระหว่าง OPC Server และ OPC Client

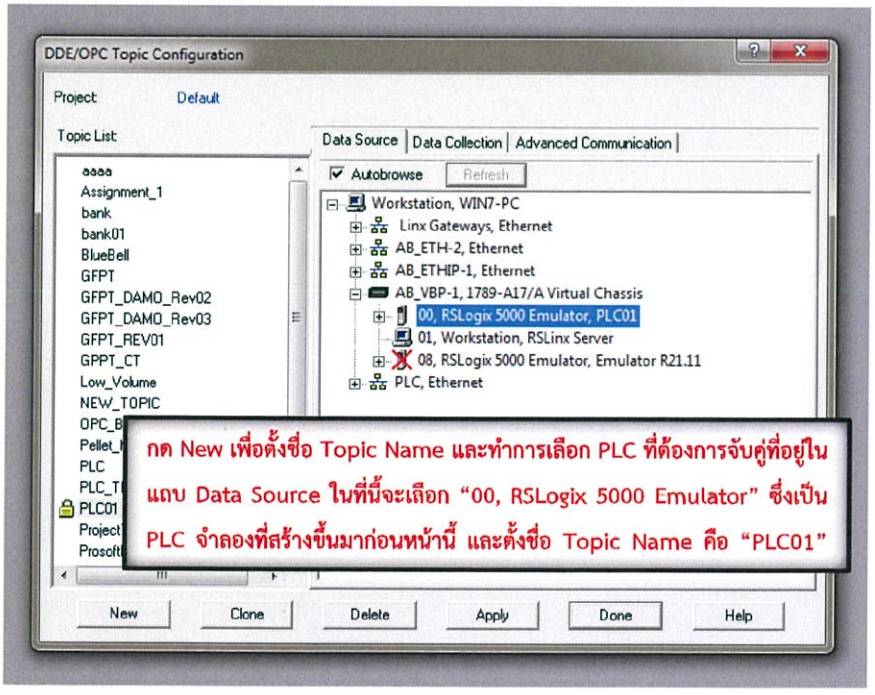
ในการเชื่อมต่อระหว่าง OPC Server และ OPC Client ผ่าน DAServer จำเป็นที่ต้องทำการตั้งค่าชื่อหัวข้อ (Topic Name) ของทั้ง OPC Server และ OPC Client ให้ตรงกัน และ IP Address ของทั้งสองเครื่องที่ใช้งานซอฟต์แวร์ต้องอยู่ในวงศ์เดียวกัน เพื่อรับส่งข้อมูลผ่าน DAServer ซึ่งสามารถแบ่งเป็นขั้นตอนในการตั้งค่าออกเป็น 2 ขั้นตอน ดังนี้

3.4.1 ตั้งค่า Topic Name ของ OPC Server

ในขั้นตอนนี้จะเป็นการตั้งค่า Topic Name ของทางด้าน OPC Server นั่นคือ ซอฟต์แวร์ RSLinx Classic ให้ตรงกับ OPC Client ในที่นี้จะทำการตั้งชื่อ Topic Name คือ “PLC01” จากนั้นนำ Topic Name ที่ตั้งชื่อมาจับคู่กับ PLC จำลองที่สร้างขึ้นในซอฟต์แวร์ RSLogix Emulate 5000 Chassis Monitor โดยมีขั้นตอนการตั้งค่าตาม รูปที่ 3.17 และ รูปที่ 3.18 ตามลำดับ



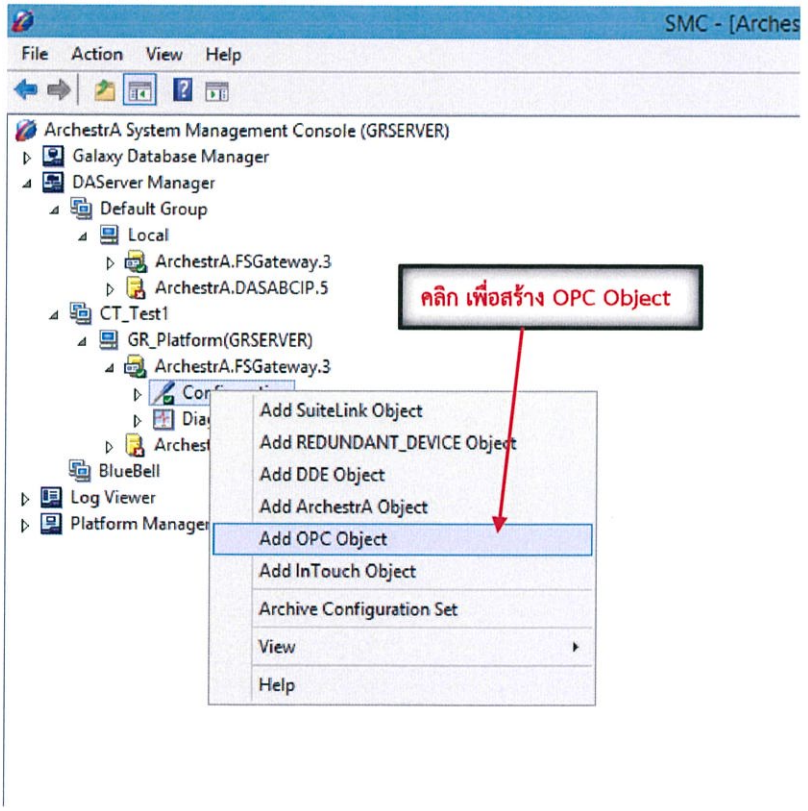
รูปที่ 3.17 การตั้งค่า Topic Name



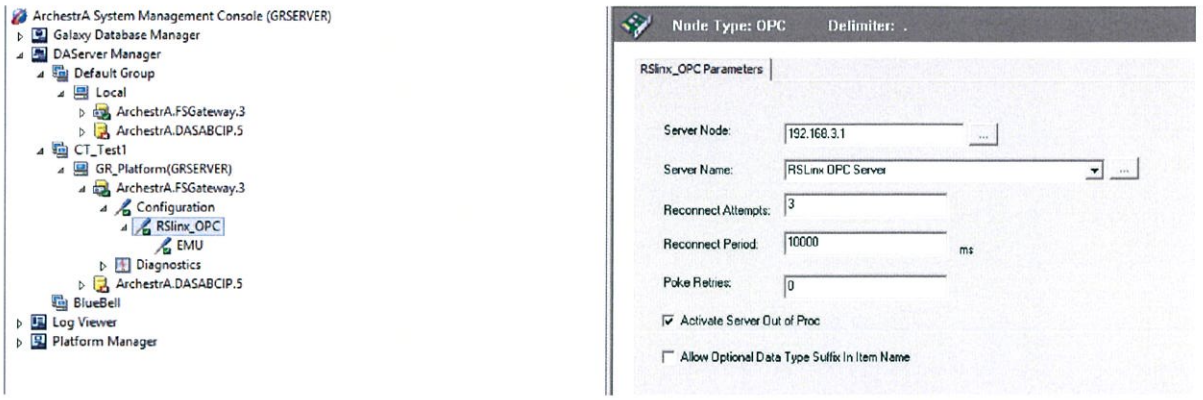
รูปที่ 3.18 การจับคู่ Topic Name กับ PLC จำลอง

3.4.2 ตั้งค่า Topic Name ของ OPC Client

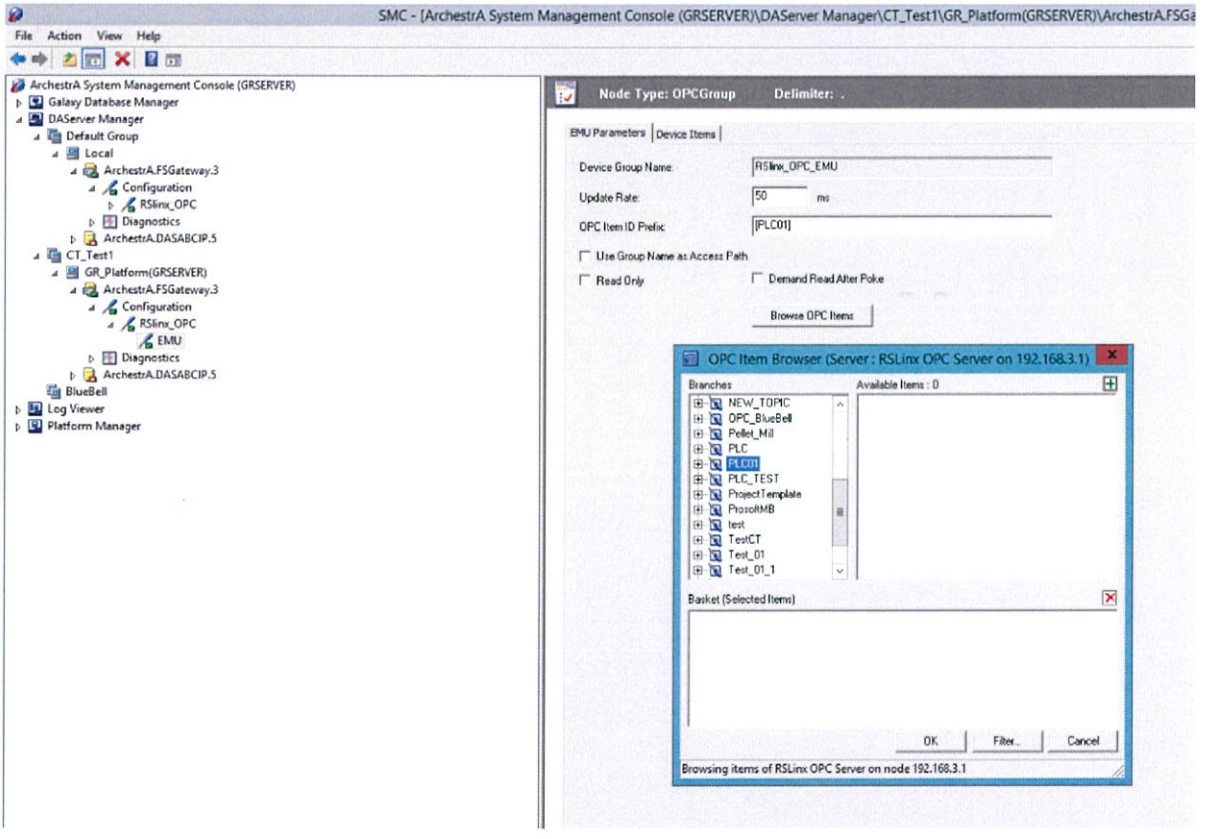
ในขั้นตอนนี้จะเป็นการตั้งค่า Topic Name ของทางด้าน OPC Client นั่นคือ ซอฟต์แวร์ ArchestrA SMC ให้ตรงกับ OPC Server โดยเริ่มจากการสร้าง OPC Object ใน Node Group ที่ทำการใช้งาน ในที่นี้จะใช้ FSGateway เป็นตัวส่งข้อมูลของ OPC Client จึงทำการสร้าง OPC Object ลงใน FSGateway ของ Node Group: CT_Test1 และตั้งชื่อว่า RSlinx_OPC จากนั้นทำการใส่ Server Node นั่นคือ IP Address ของเครื่องที่เป็น OPC Server และใส่ Server Name เพื่อระบุชื่อให้กับ OPC ที่ต้องการเชื่อต่อในครั้งนี้ ในที่นี้กำหนดให้ชื่อ "RSLinx OPC Server" และจะได้แถบ "EMU" ขึ้นมาในแถบ RSlinx_OPC จากนั้น ให้ทำการกด "Browse OPC Items" เพื่อเลือก Topic Name ใน OPC Server ที่เคยสร้างไว้ นั่นคือ "PLC01" เมื่อเชื่อมต่อสำเร็จ จะปรากฏรายชื่อ Attribute ที่เคยสร้างไว้ ในแถบ RSlinx_OPC_EMU และจะแสดงสัญลักษณ์สีเขียว โดยมีขั้นตอนการตั้งค่าตาม รูปที่ 3.19, รูปที่ 3.20, รูปที่ 3.21 และ รูปที่ 3.22 ตามลำดับ



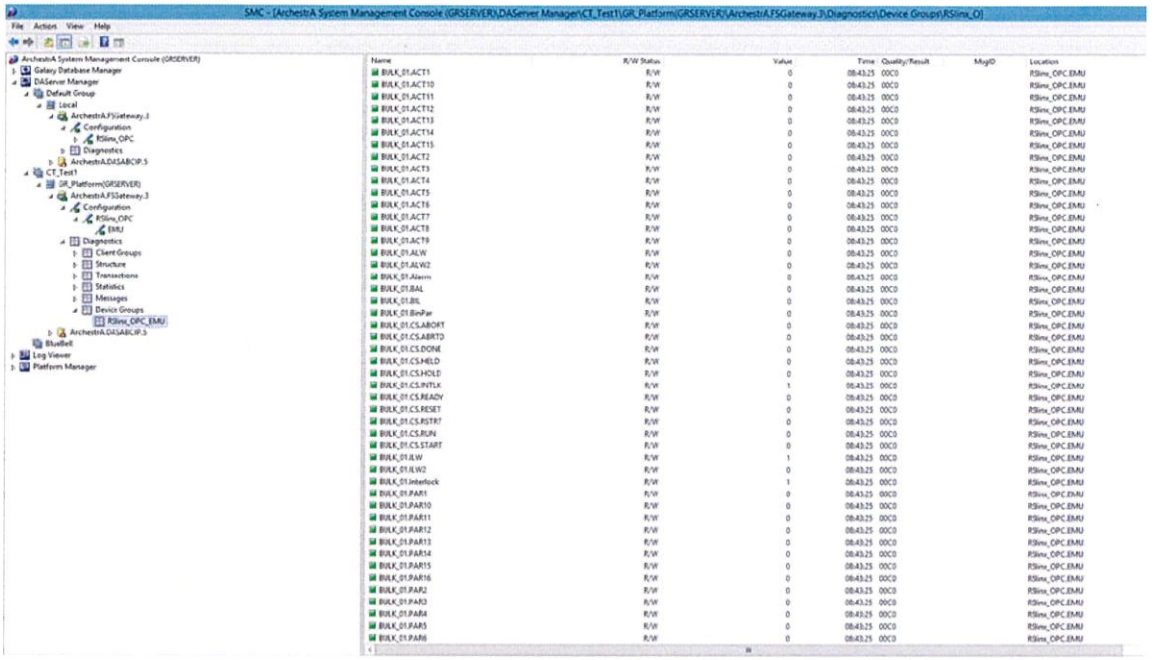
รูปที่ 3.19 การสร้าง OPC Object ในซอฟต์แวร์ ArchestrA SMC



รูปที่ 3.20 การตั้งค่า OPC Object ในซอฟต์แวร์ ArchestrA SMC



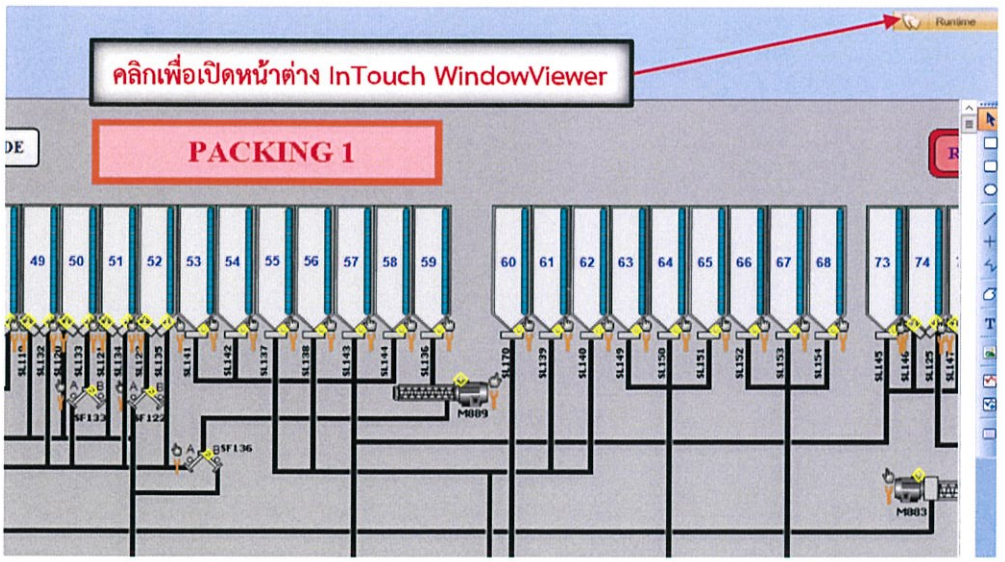
รูปที่ 3.21 การตั้งค่า OPC Object ในซอฟต์แวร์ ArchestrA SMC



รูปที่ 3.22 การแสดงผลของ Attribute ในซอฟต์แวร์ ArchestrA SMC เมื่อเชื่อมต่อสำเร็จ

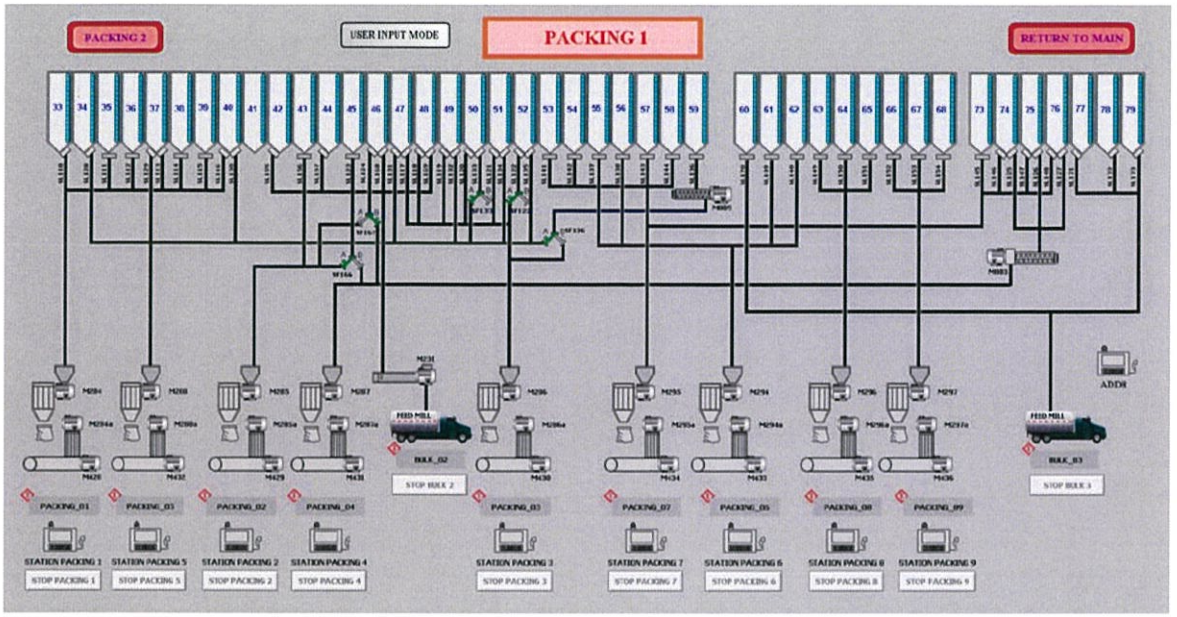
3.5 ทำการทดสอบการสั่งการ PLC ผ่านหน้าจอควบคุม HMI

หลังจากที่ทำการเชื่อมต่อ OPC Server กับ OPC Client เสร็จเรียบร้อยแล้ว ก็ทำการทดสอบการสั่งการ PLC ผ่านหน้าจอควบคุม HMI ดังแสดงในรูปที่ 3.23 โดยการกดปุ่ม Runtime ในซอฟต์แวร์ InTouch WindowMaker



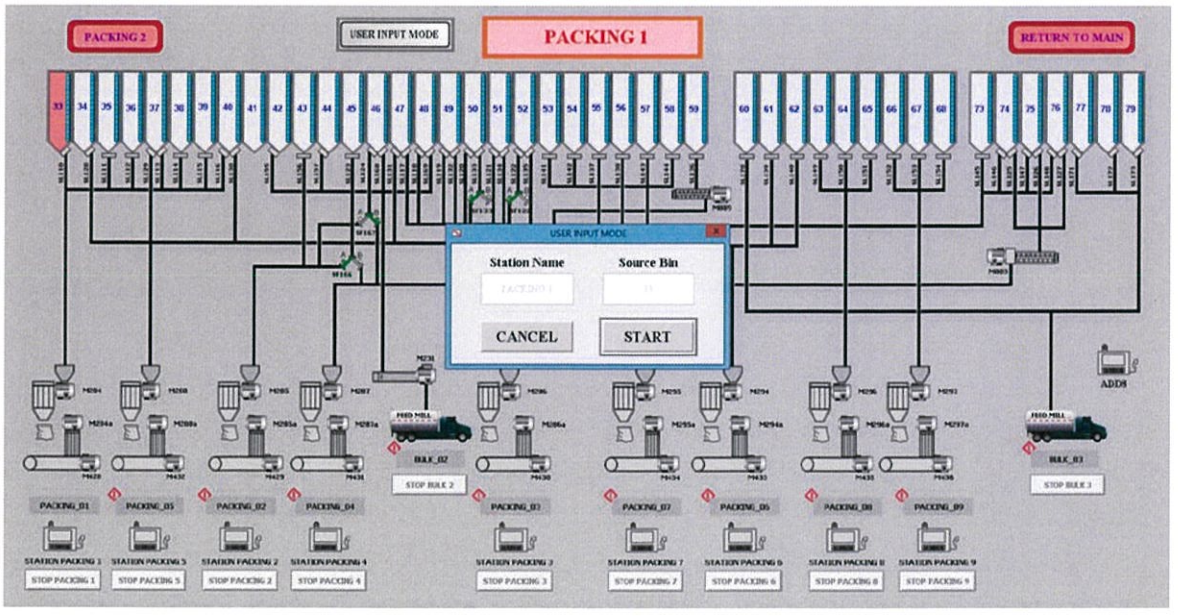
รูปที่ 3.23 การเปิดหน้าต่าง InTouch WindowViewer จาก InTouch WindowMaker

หลังจากกดปุ่ม Runtime จะปรากฏหน้าต่างขึ้นมา นั่นคือ หน้าจอควบคุม HMI ดังแสดงในรูปที่ 3.24



รูปที่ 3.24 หน้าจอควบคุมที่ออกแบบไว้ แสดงบน InTouch WindowViewer

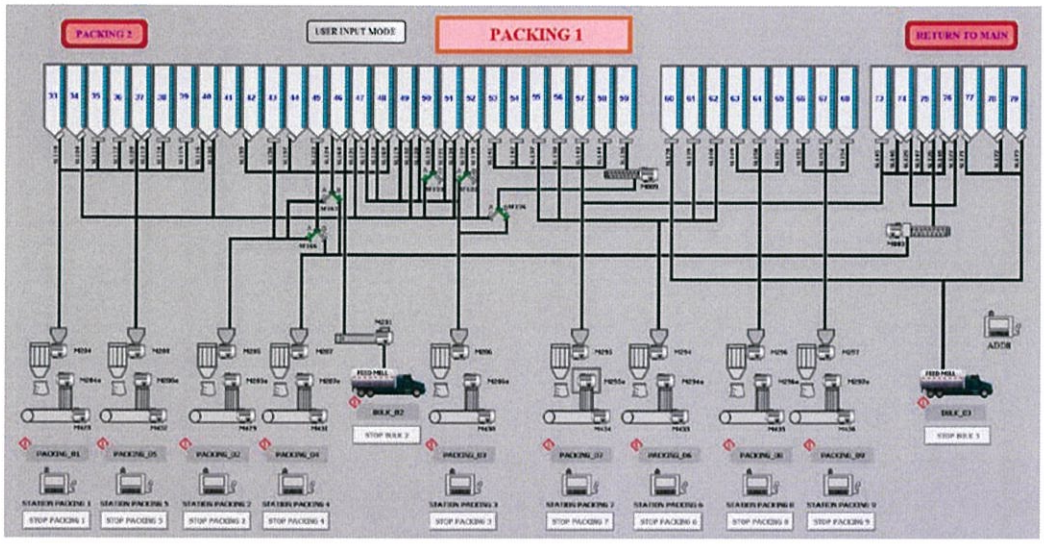
จากนั้น จะสามารถสั่งการ PLC ได้จากหน้าจอ HMI โดยรูปที่ 3.25 จะเป็นตัวอย่างของการสั่งการที่เป็นไปตามตรรกะ PLC



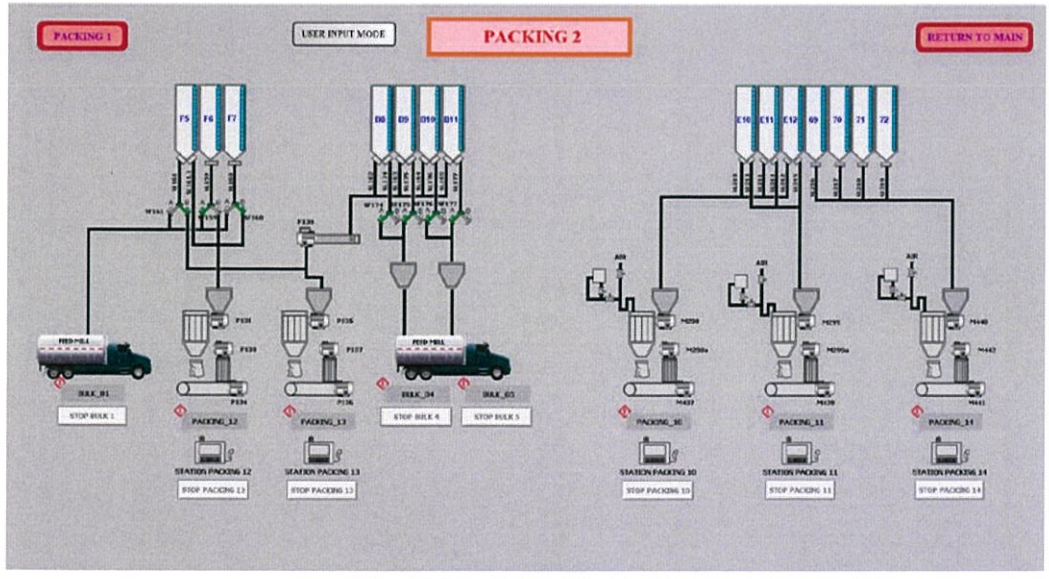
รูปที่ 3.25 ตัวอย่างของการสั่งการที่เป็นไปตามตรรกะ PLC ที่ถูกแสดงบน InTouch WindowViewer

บทที่ 4
ผลการดำเนินงาน

ในส่วนผลการดำเนินงาน ได้ผลลัพธ์ คือ การออกแบบระบบควบคุมอัตโนมัติด้วย PLC ผ่าน SCADA สำหรับกระบวนการการบรรจุอาหารสัตว์ และมีการทำงานตามแบบที่ลูกค้าทำการเสนอมา โดยมีข้อกำหนดในการออกแบบการทำงานของกระบวนการ (FDS) ในรูปแบบของ Flowchart Diagram และกำหนดรูปร่างของโครงสร้างหน้าจอควบคุม ดังแสดงในรูปที่ 4.1 (ก) และ 4.2 (ข) ตามลำดับ ได้มีความใกล้เคียงกับ P&ID ที่ทางลูกค้าต้องการได้มากที่สุดเท่าที่จะเป็นไปได้



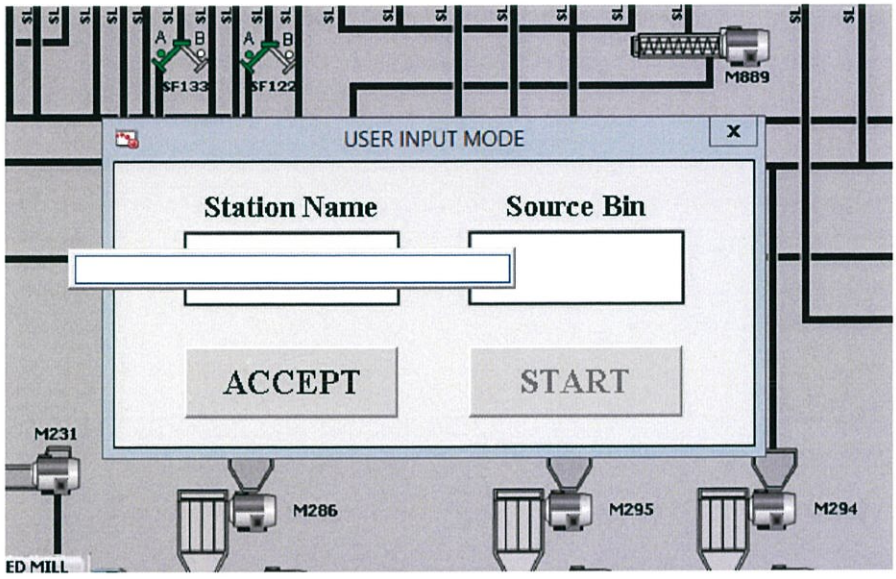
รูปที่ 4.1(ก) หน้าจอควบคุม HMI ของกระบวนการบรรจุอาหารสัตว์หน้า 1



รูปที่ 4.1(ข) หน้าจอควบคุม HMI ของกระบวนการบรรจุอาหารสัตว์หน้า 2

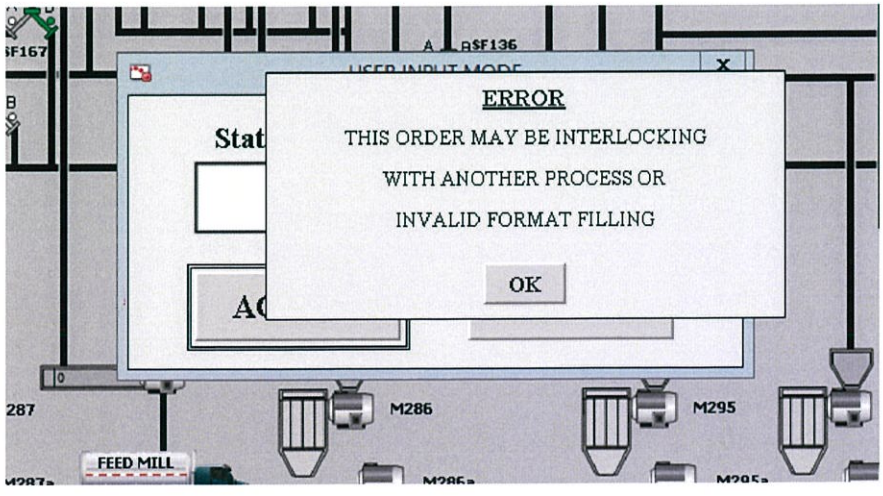
ทางผู้จัดทำสามารถออกแบบโปรแกรมควบคุม PLC ได้ตามเป้าหมายที่วางไว้ใน FDS ได้อย่างเสร็จสมบูรณ์ โดยการทำงานของตรรกะใน FDS สามารถสรุปการทำงานทั้งหมดเป็นกรณีต่าง ๆ โดยมีการควบคุม และการแสดงผลของหน้าจอควบคุม HMI ตามกรณี ดังต่อไปนี้

1) สามารถทำการเลือกชื่อสถานีบรรจุ (Station Packing) และหมายเลขถัง (Source Bin) โดยการป้อนค่าใส่หน้าจอควบคุม HMI ดังแสดงในรูปที่ 4.2



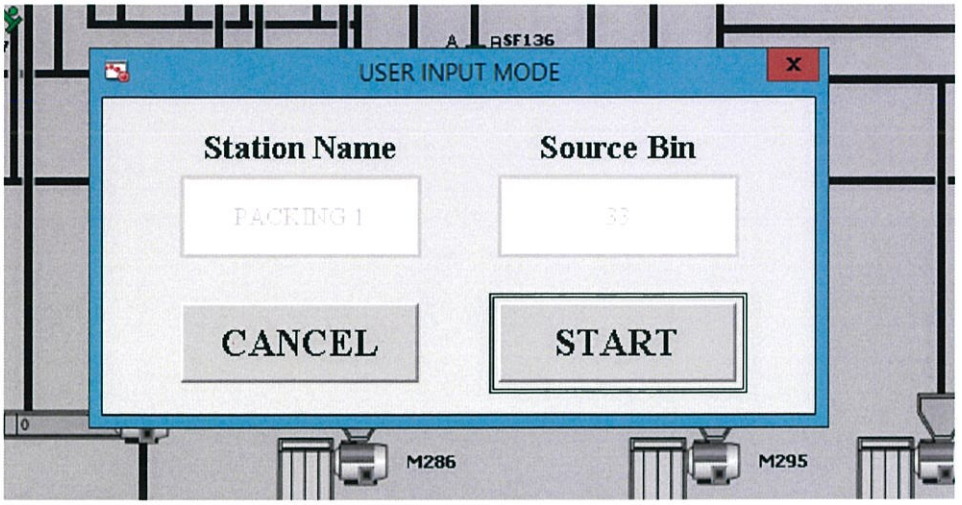
รูปที่ 4.2 การกรอกค่า Station Packing และ Source Bin

2) เมื่อทำการป้อนค่าเสร็จ และทำการกดปุ่ม ACCEPT ถ้า Station Packing และ Source Bin ไม่สอดคล้องกัน หรือมีการใส่ค่าผิดรูปแบบ หรือ Station Packing และ Source Bin ที่ใส่ไปมีการทำงานทับซ้อน (Interlock) กับกระบวนการอื่น ให้ทำการแสดงหน้าต่างแจ้งเตือน (Pop-up) และแสดงสถานะ Interlocked ดังแสดงในรูปที่ 4.3



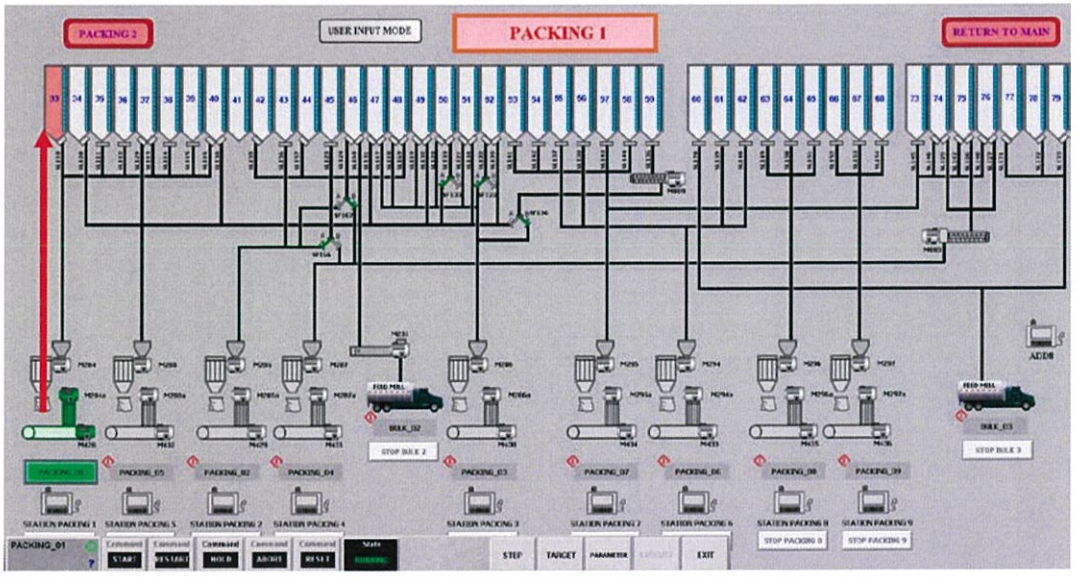
รูปที่ 4.3 แสดงหน้าต่างแจ้งเตือน

3) ถ้าอยู่นอกเหนือจากกระบวนการใน ข้อที่ 2) แสดงว่า Station Packing ที่เลือกสามารถทำงานได้ และให้แสดงสถานะ Ready และพร้อมสำหรับทำการกดปุ่ม Start เพื่อเริ่มต้นการเปิดอุปกรณ์ใน Station Packing นั้น ไปยัง Source Bin ที่ได้เลือกไว้ ดังแสดงในรูปที่ 4.4



รูปที่ 4.4 การกรอกข้อมูลที่ถูกต้อง และพร้อมสำหรับการกดปุ่ม Start

4) การเริ่มต้นของอุปกรณ์ใน Station Packing จะต้องเปิดอุปกรณ์ที่อยู่ปลายทางไล่มาจนถึงต้นทาง ดังแสดงในรูปที่ 4.5



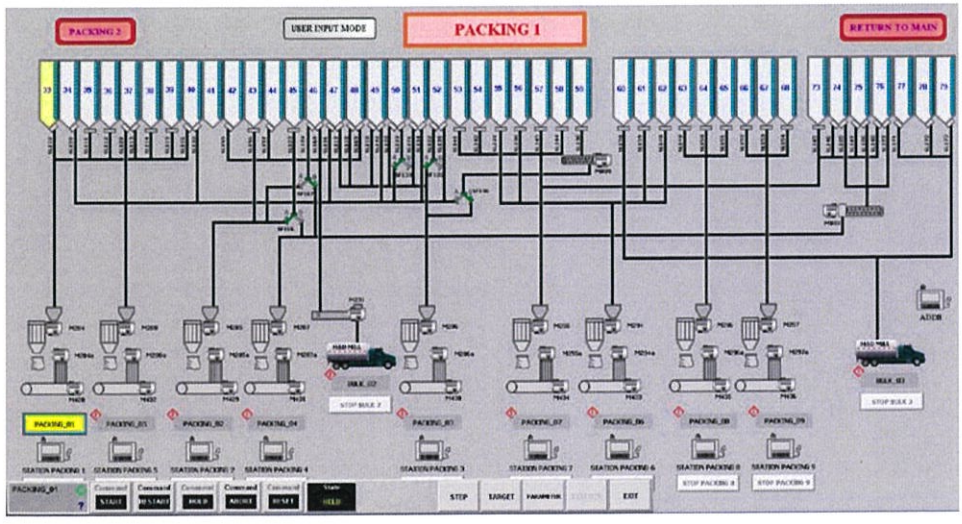
รูปที่ 4.5 แสดงการไล่ลำดับการเปิดอุปกรณ์ใน Station Packing

5) เมื่อทำการเริ่มต้นระบบจะแสดงสถานะ Running จนกว่าจะกดปุ่ม Stop ของ Station Packing ที่ได้เลือกไว้ ซึ่งปุ่ม Stop สามารถใช้งานได้ก็ต่อเมื่ออุปกรณ์ทุกตัวที่อยู่ระหว่างต้นทาง และปลายทางใน Station Packing และ Source Bin ที่เลือกไว้ จะต้องเปิดใช้งานหมดทุกตัว ดังแสดงในรูปที่ 4.6

รูปที่ 4.6 แสดงปุ่ม Stop ที่สามารถกดได้หลังจากอุปกรณ์ทุกตัวใน Station Packing เปิดหมดแล้ว

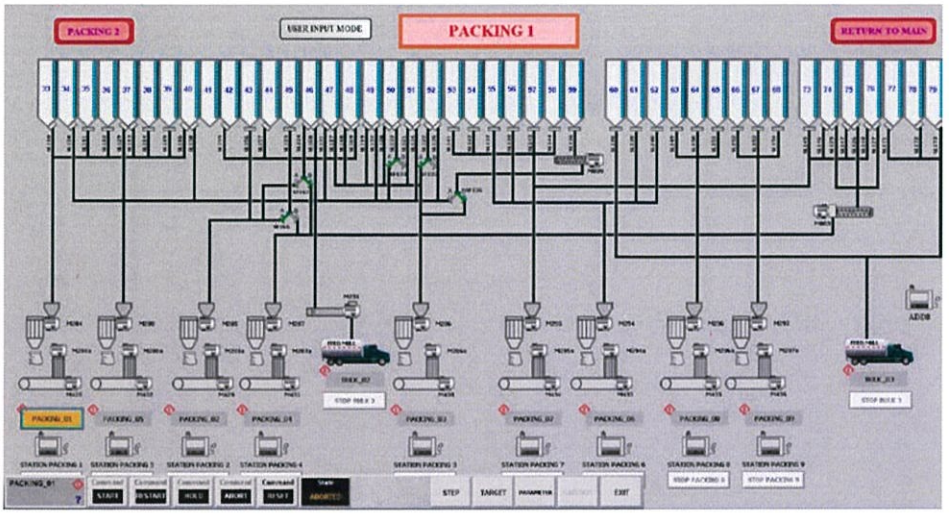
6) เมื่อทำการกดปุ่ม Stop จะทำการปิดอุปกรณ์ทุกตัวจากต้นทางไล่มาจนถึงปลายทาง

7) เมื่อระบบอยู่ในสถานะ Running สามารถกดปุ่ม Hold เพื่อหยุดกระบวนการชั่วคราวได้ และในการหยุดชั่วคราวนี้มีผลทำให้อุปกรณ์ทุกตัวที่ถูกใช้งานใน Station Packing นั้นปิดตัวลงพร้อมกันหมดพร้อมทั้งแสดงสถานะ Held ดังแสดงในรูปที่ 4.7



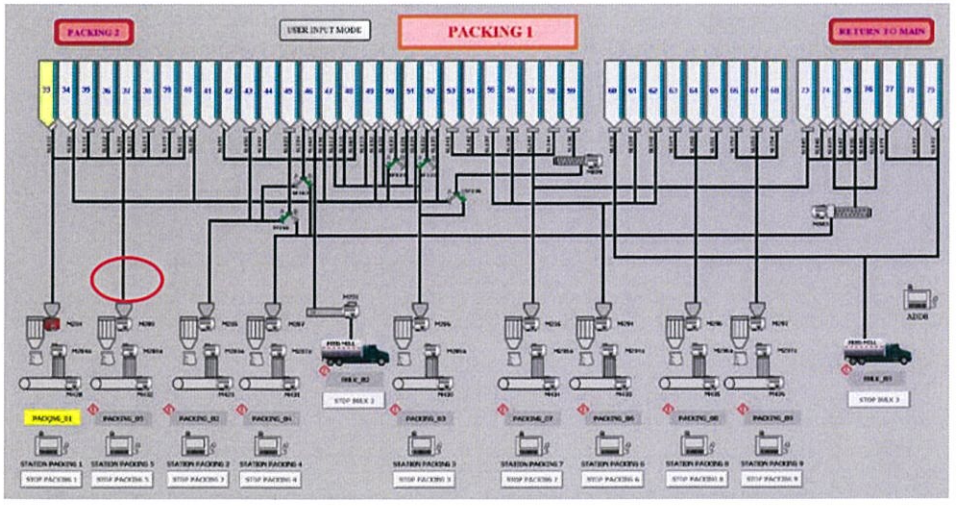
รูปที่ 4.7 แสดงสถานะของกระบวนการในสถานะ Held

8) เมื่ออยู่ในสถานะ Held ถ้ากดปุ่ม Restart จะทำกระบวนการเข้าสู่สถานะ Running อีกครั้ง และยังคงสถานะของอุปกรณ์ตอนก่อนกดปุ่ม Hold แต่ถ้ากดปุ่ม Abort จะเป็นการยุติการทำงานของกระบวนการ พร้อมทั้งเข้าสู่สถานะ Aborted และต้องทำการกดปุ่ม Reset เพื่อล้างค่าพารามิเตอร์ต่าง ๆ และเตรียมพร้อมสำหรับการสั่งการในครั้งใหม่ ซึ่งปุ่ม Reset สามารถกดได้ก็ต่อเมื่ออยู่ในสถานะ Aborted ดังแสดงในรูปที่ 4.8



รูปที่ 4.8 แสดงสถานะของกระบวนการในสถานะ Aborted

9) ในขณะที่กระบวนการของ Station Packing อยู่ในสถานะ Running ถ้ามีอุปกรณ์อย่างน้อย 1 ตัว ในกระบวนการที่เลือกเกิดการทำงานผิดพลาด (Fault) กระบวนการนั้นจะเข้าสู่สถานะ Held อัตโนมัติ โดยที่ไม่สามารถกด Restart ได้ และทำการ Alarm ที่อุปกรณ์ตัวนั้นในหน้าจอควบคุม HMI ซึ่งจะ Alarm เรื่อย ๆ ถึงแม้จะกดปุ่ม Abort และ Reset แล้วก็ตาม จนกว่าอุปกรณ์ตัวนั้นจะกลับมาทำงานได้อย่างปกติ ดังแสดงในรูปที่ 4.9



รูปที่ 4.9 แสดงอุปกรณ์ที่เกิด Fault ในกระบวนการที่กำลังทำงานอยู่

บทที่ 5
สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากการดำเนินงานออกแบบโปรแกรมควบคุมและส่วนแสดงผลของกระบวนการบรรจุอาหารสัตว์แบบเอ็กซ์ทรูชัน ได้ผลลัพธ์เป็นโปรแกรมควบคุมด้วย PLC ซึ่งเขียนด้วยภาษา Ladder Diagram และหน้าจอควบคุมและแสดงผล HMI ซึ่งจะอำนวยความสะดวกในการสังเกต และสั่งงานกระบวนการโดยผู้ปฏิบัติงาน ในส่วนของโปรแกรมควบคุมสามารถควบคุม และแสดงผลได้สอดคล้องกับความต้องการของความต้องการของลูกค้า รวมถึงช่วยให้เกิดความสะดวกต่อการแก้ไขลำดับขั้นตอนการทำงานของกระบวนการ หรือฟังก์ชันการทำงานของอุปกรณ์แต่ละประเภท และในส่วนของการแสดงผลซึ่งมีการเก็บวัตถุกราฟิกต่าง ๆ อย่างเป็นระบบ ส่งผลให้การแก้ไขและจัดการคุณสมบัติของวัตถุรวดเร็ว ลดความซ้ำซ้อนและความผิดพลาดที่อาจเกิดขึ้นได้ ซึ่งจากผลลัพธ์ข้างต้นนี้จะได้องค์ประกอบที่สำคัญในการควบคุมกระบวนการบรรจุอาหารสัตว์ ที่พร้อมจะนำไปติดตั้งและทดสอบร่วมกับระบบอื่น ๆ ซึ่งเกี่ยวข้องกับการควบคุมกระบวนการ เช่น ระบบฐานข้อมูล (Database) จึงเป็นประโยชน์ในการพัฒนาการควบคุมให้มีประสิทธิภาพมากยิ่งขึ้น

5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่พบระหว่างการปฏิบัติงานสหกิจศึกษาคือ ข้อมูลการทำงานของกระบวนการจากแผนผังแสดงความต้องการของลูกค้าไม่ชัดเจน รวมถึงมีการปรับเปลี่ยนบ่อยครั้ง เนื่องจากผู้เขียนแผนไม่ใช่ผู้ที่คุ้นเคยกับอุปกรณ์ในสายการผลิตเท่ากับผู้ปฏิบัติงานซึ่งทำงานอยู่ที่หน้างานเป็นประจำ หรือทางบริษัทลูกค้าอาจยังไม่เชี่ยวชาญทางด้าน Automation มากนัก เพราะอยู่ในช่วงเริ่มปรับปรุงกระบวนการใหม่ ดังนั้น แนวทางแก้ไขที่ทำได้คือปรึกษากับผู้นิเทศงานถึงปัญหาในรายละเอียดต่าง ๆ เพื่อให้ผู้นิเทศงานติดต่อประสานงานกับลูกค้าเพื่อขอข้อมูลที่ชัดเจน รวมถึงการรายงานผลงานเป็นระยะ เพื่อดูผลตอบรับของบริษัทลูกค้าว่าต้องการแก้ไขเพิ่มเติมส่วนใดอีกบ้าง ซึ่งจะประหยัดเวลาและทรัพยากรมากกว่าการทำงานจนเสร็จทั้งหมดแล้วต้องแก้ไขในภายหลัง

5.3 ข้อเสนอแนะ

ในการพัฒนาระบบควบคุมกระบวนการต่าง ๆ ซึ่งลูกค้าส่วนใหญ่มักมีความต้องการที่หลากหลาย สิ่งที่สำคัญที่สุดคือการวางโครงสร้างของโปรแกรมควบคุมอย่างเป็นระบบ ง่ายต่อการเข้าใจและการจัดการ เพื่อช่วยลดความยุ่งยากและประหยัดเวลาในการแก้ไขโปรแกรม เนื่องมาจากการปรับเปลี่ยนแนวคิดและอุปกรณ์ในภายหลัง

เอกสารอ้างอิง

[1] Food and Agriculture Organization of the United Nations; Effects of Processing on the Nutritional Value of Feeds; แหล่งที่มา:
http://www.fao.org/docrep/x5738e/x5738e0l.htm#TopOfPage
(สืบค้นวันที่ 21 กันยายน 2559)

[2] อังคณา หลักหมั้น และคณะ; เทคโนโลยีการผลิตอาหารสัตว์อัดเม็ดประจำปี 2556; แหล่งที่มา:
http://www.agro.cmu.ac.th/absc/data/57/57-008.pdf
(สืบค้นวันที่ 3 พฤศจิกายน 2559)

[3] ชูศักดิ์ ลิ่มสกุล และคณะ; การใช้งานพีแอลซี; แหล่งที่มา:
lectric2.ee.psu.ac.th/~kanadit/S5W/plc _s5w.pdf
(สืบค้นวันที่ 18 พฤศจิกายน 2559)

[4] แนะนำให้รู้จัก PLC; แหล่งที่มา:
www.tatc.ac.th/files/0902050883921_1106010774824.pdf
(สืบค้นวันที่ 18 พฤศจิกายน 2559)

[5] SCADA คืออะไร; แหล่งที่มา:
http://www.binaryadvance.com/SCADA.html
(สืบค้นวันที่ 18 พฤศจิกายน 2559)

[6] Cogent Real-Time Systems Inc.; What is OPC; แหล่งที่มา:
http://www.opcdatahub.com/WhatIsOPC.html
(สืบค้นวันที่ 25 ธันวาคม 2559)

[7] Rockwell Automation Inc.; STUDIO 5000 LOGIX DESIGNER; แหล่งที่มา:
https://www.rock wellautomation.com/rockwellsoftware/products/studio5000-logix-designer.page# system-requirements
(สืบค้นวันที่ 25 ธันวาคม 2559)

[8] A Quick Tutorial on RSLogix Emulator 5000; แหล่งที่มา:
http://www.plcdev.com/a_quick_tutorial_on_rslogix_emulator_5000
(สืบค้นวันที่ 25 ธันวาคม 2559)

ประวัติผู้เขียน

ชื่อ-สกุล นายสหัสชัย วิชาวิทยากุล

วัน เดือน ปีเกิด 6 กรกฎาคม พ.ศ.2538

ที่อยู่ 3/11 ซอย สวนสยาม 28 แยก 1 แขวง คันนายาส เขต คันนายาว กรุงเทพฯ 10230

E-mail cgamezsawi@gmail.com

โทรศัพท์ 085-332-3847

ประวัติการศึกษา

- พ.ศ.2554 - 2556: ระดับมัธยมศึกษาตอนปลาย โรงเรียนเตรียมอุดมศึกษาน้อมเกล้า
- พ.ศ.2557 - ปัจจุบัน: วิศวกรรมศาสตร์บัณฑิต สาขาวิชา วิศวกรรมการวัดคุม
  ภาควิชา วิศวกรรมการวัดและควบคุม คณะ วิศวกรรมศาสตร์
  สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ประสบการณ์

- นักศึกษาฝึกงาน แผนก วิศวกรไฟฟ้าและเครื่องมือวัด
  บริษัท ไทยนิปปอน สตีล แอนด์ ซูมิคิน เอ็นจิเนียริ่ง แอนด์ คอนสตรัคชั่น คอร์ปอเรชั่น จำกัด
- นักศึกษาโครงการสหกิจศึกษา แผนก วิศวกรอัตโนมัติ บริษัท อินคอนโทรล จำกัด