



รายงานสหกิจศึกษาฉบับสมบูรณ์

การควบคุมกระบวนการอัดเม็ดอาหารสัตว์ผ่านระบบสกาตา
Extrusion Process Control via SCADA

นางสาวปรีชมน ต้นไธง

ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559



T148594

รายงานสหกิจศึกษาฉบับสมบูรณ์

การควบคุมกระบวนการอัดเม็ดอาหารสัตว์ผ่านระบบสกาตา
Extrusion Process Control via SCADA

นางสาวปรีชมน ต้นไธง

เลขหมู่.....
เลขทะเบียน 148594
วันเดือนปี 6 พ.ย. 2560

b. 14872210
f.

ภาควิชาวิศวกรรมการผลิตและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

ชื่อโครงการสหกิจศึกษา การควบคุมกระบวนการอัดเม็ดอาหารสัตว์ผ่านระบบสกาตา

ชื่อ-สกุล นักศึกษา นางสาวปวิชนัน ต้นไธง

คณะ วิศวกรรมศาสตร์

ภาควิชา วิศวกรรมการวัดและควบคุม

ชื่อ-สกุล อาจารย์นิเทศ รศ.ดร.เกษตร์ ศิริสันติสัมฤทธิ์

ชื่อ-สกุล ผู้นิเทศงาน คุณชำนาญ ตุ่มตอง

ชื่อสถานประกอบการ บริษัท อินคอนโทรล จำกัด

บทคัดย่อ

รายงานสหกิจศึกษาฉบับสมบูรณ์นี้ได้อธิบายถึงการศึกษาและการออกแบบการควบคุมกระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ทรูชัน (Extrusion) ด้วยตัวควบคุมเชิงตรรกที่สามารถโปรแกรมได้ (PLC: Programmable Logic Controller) ผ่านหน้าจอ HMI (Human-Machine Interface) ของระบบสกาตา (SCADA: Supervisory Control and Data Acquisition) เพื่อปรับปรุงกระบวนการผลิตให้ทำงานได้ตามเป้าหมาย ลดข้อผิดพลาดในการทำงาน รวมทั้งเพิ่มประสิทธิภาพและความปลอดภัยในการดำเนินงาน โดยการออกแบบการควบคุมประกอบด้วย การเขียนโปรแกรมควบคุม PLC และการออกแบบหน้าจอแสดงผลและทดสอบการทำงานของระบบเพื่อให้พร้อมต่อการติดตั้งร่วมกับระบบอื่นๆ ที่จำเป็นในการควบคุมกระบวนการ เช่น ระบบฐานข้อมูล

คำสำคัญ: PLC, HMI, SCADA

Co-operative Research Title: Extrusion Process Control via SCADA

Student Intern Name: Ms. Parichamon Tantha-ngai

Faculty: Engineering **Department:** Instrumentation and Control Engineering

Advisor Name: Assoc. Prof. Dr. Kaset Sirisantisamrid

Mentor Name: Mr. Chumnan Toomtong

Company: InControl Company Limited

ABSTRACT

This cooperative educational report is a description of study and design extrusion process control by PLC (Programmable Logic Controller) via HMI (Human-Machine Interface) of SCADA (Supervisory Control and Data Acquisition) system. The objective of research is to improve the process by reducing the possibility of fault and increasing efficiency along with safety in the operation. The task consists of writing PLC controlling program and designing HMI display. The last procedure is testing of system operation to make it ready for installation with other important systems later, such as database system.

Keywords: PLC, HMI, SCADA

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับสมบูรณ์นี้สำเร็จลุล่วงได้อย่างดี เนื่องด้วยการสนับสนุนจากบริษัทอินคอนโทรล จำกัด ซึ่งได้ให้โอกาสแก่นักศึกษาสหกิจในการเข้าไปปฏิบัติงานในองค์กร รวมทั้งให้ความรู้และการดูแลที่อบอุ่นตลอดระยะเวลาการปฏิบัติงานสหกิจศึกษา จึงขอขอบพระคุณบริษัท อินคอนโทรล จำกัด และคุณชำนาญ ตุ่มตอง ผู้นิเทศงาน รวมถึงพนักงานทุกท่านที่ได้มอบประสบการณ์อันมีค่าในครั้งนี้

ทั้งนี้ ขอขอบพระคุณ รศ.ดร.เกษตร์ ศิริสันติสัมฤทธิ์ และคณาจารย์สาขาวิชาวิศวกรรมการวัดคุมทุกท่าน ที่ได้ให้ความช่วยเหลือและคำปรึกษาอันเป็นประโยชน์ในการปฏิบัติงานและการจัดทำรายงานสหกิจศึกษาฉบับสมบูรณ์นี้ด้วยดีเสมอมา

นางสาวปริชมน ต้นไถง

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VII
สารบัญภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	1
1.3 ขอบเขตของโครงการ.....	2
1.4 วิธีดำเนินงาน.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 อุตสาหกรรมอาหารสัตว์สำเร็จรูป.....	4
2.1.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป.....	4
2.1.1.1 การรับวัตถุดิบเข้าสายการผลิต (Raw Material Receiving).....	4
2.1.1.2 การบดวัตถุดิบ (Grinding).....	5
2.1.1.3 การผสมวัตถุดิบ (Mixing).....	5
2.1.1.4 การอัดเม็ด (Pelleting).....	5
2.1.1.5 การบรรจุหีบห่อ (Packing).....	5
2.1.2 วิธีการอัดเม็ดอาหารสัตว์.....	6
2.1.2.1 การอัดเม็ดอาหารแบบจมน้ำ (Sinking Pellets).....	6
2.1.2.2 การอัดเม็ดอาหารแบบลอยน้ำ (Floating Pellets).....	6
2.2 ความรู้พื้นฐานเกี่ยวกับพีแอลซี (PLC).....	7
2.2.1 ความหมายของพีแอลซี.....	7
2.2.2 โครงสร้างของพีแอลซี.....	8
2.2.2.1 ซีพียู (CPU; Central Processing Unit).....	8
2.2.2.2 หน่วยความจำ (Memory Unit).....	8
1) หน่วยความจำชั่วคราว (RAM: Random Access Memory).....	9
2) หน่วยความจำถาวร (ROM: Read Only Memory).....	9
2.1) PROM (Programmable ROM).....	9
2.2) EPROM (Erasable Programmable ROM).....	9
2.3) EEPROM (Electrical Erasable Programmable ROM).....	9
2.2.2.3 ภาควินพุต (Input Unit).....	9
2.2.2.4 ภาควเอาต์พุต (Output Unit).....	9

สารบัญ (ต่อ)

	หน้า
2.2.2.5 ภาคแหล่งจ่ายพลังงาน (Power Supply Unit).....	10
2.2.3 ภาษาที่ใช้เขียนพีแอลซี.....	11
2.2.3.1 LD (Ladder Diagram).....	11
2.2.3.2 FBD (Function Block Diagram).....	11
2.2.3.3 IL (Instruction List).....	12
2.2.3.4 ST (Structure Text).....	12
2.2.3.5 SFC (Sequential Flow Chart).....	12
2.3 ความรู้พื้นฐานเกี่ยวกับระบบสกาดา (SCADA).....	13
2.3.1 ความหมายของสกาดา.....	13
2.3.2 โครงสร้างของระบบสกาดา.....	14
2.3.2.1 โครงสร้างด้านฮาร์ดแวร์ (Hardware Architecture).....	14
2.3.2.2 โครงสร้างด้านซอฟต์แวร์ (Software Architecture).....	14
2.3.2.3 โครงสร้างด้านการสื่อสาร (Communications).....	15
2.3.2.4 โครงสร้างด้านอินเทอร์เฟซ (Interface).....	15
2.3.2.5 โครงสร้างความสามารถในการขยายระบบ (Scalability).....	16
2.3.2.6 โครงสร้างการสำรองระบบ (Redundancy).....	16
2.3.3 หน้าที่การทำงานของสกาดา (SCADA Functionality).....	16
2.3.3.1 การเข้าถึงพารามิเตอร์ของอุปกรณ์.....	16
2.3.3.2 ระบบแสดงผลแบบ HMI (Human-Machine Interface).....	16
2.3.3.3 ระบบแสดงกราฟสัญญาณแบบต่อเนื่อง (Trending).....	17
2.3.3.4 ระบบแจ้งเตือน (Alarm).....	17
2.3.3.5 การทำงานแบบ Automation.....	17
2.3.4 การสร้างและพัฒนาระบบสกาดา (Application Development).....	17
2.3.4.1 การกำหนดคอนฟิกูเรชัน.....	17
2.3.4.2 เครื่องมือในการพัฒนา (Development Tool).....	17
2.4 ความสำคัญของ OPC.....	18
2.5 ซอฟต์แวร์ที่เกี่ยวข้อง.....	19
2.5.1 Studio 5000.....	19
2.5.1.1 ความต้องการด้านฮาร์ดแวร์.....	19
2.5.1.2 ความต้องการด้านซอฟต์แวร์ระบบปฏิบัติการ.....	19
2.5.2 Wonderware InTouch HMI 2014 R2.....	20
2.5.2.1 InTouch Application Manager.....	20
2.5.2.2 InTouch WindowMaker.....	20
2.5.2.3 InTouch WindowViewer.....	21
2.5.3 ArchestrA IDE (Integrated Development Environment).....	22

สารบัญ (ต่อ)

	หน้า
2.5.4 RSLinx Classic.....	22
2.5.5 SMC (System Management Console).....	23
2.5.6 RSLogix Emulate 5000 Chassis Monitor.....	23
บทที่ 3 การออกแบบโปรแกรมควบคุมจำลองพีแอลซี.....	25
3.1 ออกแบบโปรแกรมควบคุม.....	25
3.1.1 ศึกษากระบวนการ.....	25
3.1.2 จัดทำรายชื่ออุปกรณ์.....	26
3.1.3 ออกแบบโปรแกรมควบคุม.....	27
3.2 สร้างวัตถุ (Object) ที่ใช้ในจอแสดงผล.....	31
3.2.1 สร้างกราฟิกให้กับวัตถุ.....	31
3.2.2 สร้างวัตถุแม่แบบ.....	32
3.2.3 สร้าง Instance.....	34
3.3 ออกแบบหน้าจอแสดงผล.....	35
3.4 ทดสอบการทำงานของโปรแกรมควบคุมและส่วนแสดงผล.....	35
3.4.1 เชื่อมต่อโปรแกรมควบคุมและส่วนแสดงผลด้วย OPC.....	35
3.4.2 ทดสอบการทำงานของระบบ.....	37
บทที่ 4 ผลการดำเนินงาน.....	40
4.1 โปรแกรมควบคุม.....	40
4.1.1 Main Program.....	40
4.1.2 Subroutine.....	40
4.2 ส่วนแสดงผล.....	41
4.2.1 ภาพรวมของระบบ.....	41
4.2.2 ฟังก์ชันการแสดงผลของอุปกรณ์.....	43
4.2.2.1 Pop Up ของอุปกรณ์.....	43
4.2.2.2 การแสดงผลเมื่ออุปกรณ์ถูกสั่งงาน.....	45
1) Auto Mode.....	46
2) Manual Mode.....	46
3) Maintenance Status.....	47
4.2.2.3 การแสดงผลเมื่อเกิดความผิดปกติ.....	48
บทที่ 5 สรุปผลและข้อเสนอแนะ.....	49
เอกสารอ้างอิง.....	50
ประวัติผู้เขียน.....	51

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินงาน.....	3

สารบัญภาพ

ภาพที่	หน้า
2.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป.....	4
2.2 การทำงานของหน้าแวนและใบมีดในเครื่อง Steam Pelleting.....	7
2.3 เครื่อง Extruder.....	7
2.4 โครงสร้างของพีแอลซี.....	8
2.5 อุปกรณ์ที่ใช้ต่อกับภาคอินพุตของพีแอลซี.....	10
2.6 โหลดที่ใช้ต่อกับภาคเอาต์พุตของพีแอลซี.....	10
2.7 ภาษา Ladder Diagram.....	11
2.8 ภาษา Function Block Diagram.....	11
2.9 ภาษา Instruction List.....	12
2.10 ภาษา Structure Text.....	12
2.11 ภาษา Sequential Flow Chart.....	13
2.12 โครงสร้างด้านฮาร์ดแวร์ของระบบ SCADA.....	14
2.13 โครงสร้างด้านซอฟต์แวร์ของระบบ SCADA.....	15
2.14 การติดต่อสื่อสารระหว่างฮาร์ดแวร์กับซอฟต์แวร์โดยใช้ OPC เป็นตัวกลาง.....	18
2.15 ซอฟต์แวร์ Studio 5000.....	19
2.16 ซอฟต์แวร์ InTouch Application Manager.....	20
2.17 ซอฟต์แวร์ InTouch WindowMaker.....	21
2.18 ซอฟต์แวร์ InTouch WindowViewer.....	21
2.19 ซอฟต์แวร์ ArchestrA IDE.....	22
2.20 ซอฟต์แวร์ RSLinx Classic.....	23
2.21 ซอฟต์แวร์ System Management Console.....	24
2.22 ซอฟต์แวร์ RSLogix Emulate 5000 Chassis Monitor.....	24
3.1 แผนผังแสดงความต้องการของลูกค้า.....	25
3.2 แผนภาพ P&ID.....	26
3.3 คำอธิบายชื่ออุปกรณ์.....	26
3.4 รายชื่ออุปกรณ์ที่จำแนกตามประเภท.....	27
3.5 รายการตัวแปรที่จะใช้ในโปรแกรมควบคุม.....	27
3.6 Flowchart สำหรับเขียนโปรแกรม.....	28
3.7 การสร้างตัวแปรให้กับประเภทของอุปกรณ์.....	29
3.8 การตั้งค่าชนิดของพีแอลซี.....	29
3.9 การสร้างตัวแปรให้กับชื่อของอุปกรณ์.....	30
3.10 โปรแกรมควบคุมที่เขียนด้วยภาษา Ladder Diagram.....	30

สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.11 การสร้างกราฟิกให้กับวัตถุ.....	31
3.12 การวาดกราฟิก.....	31
3.13 การเพิ่ม Animation Link ให้กับรูปกราฟิก.....	32
3.14 การสร้างวัตถุแม่แบบ.....	32
3.15 การสร้าง Attribute ให้กับวัตถุแม่แบบ.....	33
3.16 การเขียน Script ให้กับวัตถุแม่แบบ.....	33
3.17 การเพิ่มกราฟิกให้กับวัตถุแม่แบบ.....	34
3.18 การสร้าง Instance.....	34
3.19 การสร้างหน้าจอแสดงผล.....	35
3.20 แผนผังการรับ-ส่งข้อมูลระหว่างซอฟต์แวร์.....	35
3.21 การตั้งค่าการติดต่อสื่อสารด้วย Topic Name ของ OPC Protocol.....	36
3.22 การจับคู่ Topic Name กับอุปกรณ์ที่ต้องการ.....	36
3.23 การสร้าง OPC Object.....	37
3.24 การสร้าง OPC Group Object.....	37
3.25 การ Run โปรแกรมควบคุมในพีแอลซีจำลอง.....	38
3.26 หน้าจอแสดงผลในซอฟต์แวร์ InTouch WindowViewer.....	38
3.27 ตัวอย่างการทดสอบระบบ.....	39
4.1 Main Program ของโปรแกรมควบคุม.....	40
4.2 Subroutine ควบคุมการทำงานของ On - Off Motor.....	41
4.3 ภาพรวมของกระบวนการขณะที่ยังไม่มีอุปกรณ์ทำงาน.....	42
4.4 ภาพรวมของกระบวนการขณะที่อุปกรณ์บางส่วนทำงานและบางส่วนเกิดความผิดปกติ.....	42
4.5 ภาพรวมของกระบวนการเมื่อเกิด Communication Fault ระหว่างซอฟต์แวร์ที่ใช้ แสดงผลและซอฟต์แวร์ที่เก็บกราฟิกของอุปกรณ์.....	43
4.6 Pop Up ของ On - Off Motor (มอเตอร์สำหรับเปิด - ปิด เท่านั้น).....	43
4.7 Pop Up ของ Left - Right Motor (มอเตอร์หมุนกลับทิศทางได้).....	44
4.8 Pop Up ของ 2 - Way Valve (วาล์ว 2 ทาง).....	44
4.9 Pop Up ของ 3 - Way Valve (วาล์ว 3 ทาง).....	45
4.10 Pop Up ของ Sliding Valve (วาล์วสไลด์เปิด - ปิด).....	45
4.11 การแสดงผลเมื่ออุปกรณ์ทำงานอยู่ใน Auto Mode.....	46
4.12 การแสดงผลเมื่ออุปกรณ์ทำงานอยู่ใน Manual Mode.....	47
4.13 การแสดงผลเมื่ออุปกรณ์อยู่ในสถานะ Maintenance.....	47
4.14 การแสดงผลเมื่อเกิดความผิดปกติขึ้นกับอุปกรณ์.....	48

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญ

บริษัทอินคอนโทรล จำกัด เป็นบริษัทที่ดำเนินธุรกิจเกี่ยวกับการออกแบบ ปรับปรุงระบบควบคุมแบบอัตโนมัติ ตั้งแต่การควบคุมเครื่องจักรไปจนถึงกระบวนการที่ซับซ้อน ครอบคลุมกลุ่มลูกค้า ตั้งแต่โรงงานอุตสาหกรรม ระบบสาธารณูปโภคพื้นฐาน ไปจนถึงส่วนการก่อสร้างอาคารต่างๆ ที่ต้องการระบบอัตโนมัติ โดยสามารถดำเนินการได้ทั้งด้านเครื่องมือตรวจวัด ระบบควบคุม HMI (Human-Machine Interface) ระบบฐานข้อมูล และ MES (Manufacturing Execution Systems)

ทั้งนี้บริษัทอินคอนโทรล จำกัด ได้รับมอบหมายให้ดำเนินการปรับปรุงระบบควบคุมการผลิตของบริษัทกรุงเทพอาหาร จำกัด (มหาชน) ซึ่งเป็นลูกค้าในกลุ่มอุตสาหกรรมการผลิตอาหารสัตว์สำเร็จรูป ดำเนินธุรกิจการผลิตอาหารสัตว์หลายชนิด ได้แก่ อาหารไก่ อาหารสุกร อาหารวัว อาหารปลา อาหารกุ้ง เป็นต้น เนื่องจากทางบริษัทกรุงเทพอาหาร จำกัด (มหาชน) เล็งเห็นว่าอุตสาหกรรมการผลิตอาหารสัตว์สำเร็จรูปในปัจจุบันเริ่มขยายตัวเพิ่มขึ้นตามความต้องการของธุรกิจปศุสัตว์ อันเป็นผลมาจากอุปสงค์ที่สูงขึ้นของผู้บริโภค ดังนั้นจึงมีความต้องการที่จะดำเนินโครงการปรับปรุงการควบคุมกระบวนการผลิต จากเดิมคือแบบใช้คนควบคุม (Manual Control) มาเป็นแบบอัตโนมัติ เพื่อเพิ่มประสิทธิภาพในการผลิต อันจะเป็นผลไปสู่การเพิ่มปริมาณ และคุณภาพของผลิตภัณฑ์ให้สามารถแข่งขันในกลุ่มอุตสาหกรรมเดียวกันได้

การผลิตอาหารสัตว์สำเร็จรูป จะมีกระบวนการอัดเม็ดแบบเอ็กซ์ทรูชัน (Extrusion) ซึ่งใช้ผลิตอาหารเม็ดแบบลอยน้ำ เพื่อให้เหมาะสมกับพฤติกรรมการกินอาหารของสัตว์น้ำบางชนิด โดยการควบคุมกระบวนการจะใช้ผู้ปฏิบัติงานเพื่อเปิด-ปิดสวิตช์ของอุปกรณ์ต่างๆ ประกอบด้วย วาล์วควบคุมน้ำและไอน้ำ เพื่อปรับอุณหภูมิและเพิ่มความชื้นให้วัตถุดิบ มอเตอร์ของเครื่องอัดเม็ด เครื่องอบแห้ง และสายพานลำเลียง เป็นต้น ซึ่งการควบคุมแบบนี้อาจทำให้เกิดความผิดพลาดได้ง่าย ทั้งต่อคน เครื่องจักร และวัตถุดิบ เนื่องจากฐานข้อมูลการผลิตยังไม่เป็นฐานข้อมูลเดียวกัน การจะควบคุมเครื่องจักรทั้งกระบวนการให้ทำงานสัมพันธ์กันได้อาศัยเพียงการสื่อสารระหว่างผู้ปฏิบัติงานเท่านั้น ซึ่งไม่เพียงพอต่อการควบคุมให้มีประสิทธิภาพ ทางบริษัทกรุงเทพอาหาร จำกัด (มหาชน) จึงมีแนวคิดให้นำเครื่องควบคุมเชิงตรรกที่สามารถโปรแกรมได้ (PLC: Programmable Logic Controller) มาใช้ควบคุมกระบวนการ ควบคู่กับระบบสกาตา (SCADA: Supervisory Control and Data Acquisition) ซึ่งสามารถควบคุม ตรวจสอบ วิเคราะห์ และเก็บข้อมูลการผลิตแบบเรียลไทม์ (Real Time) ได้

โครงการนี้จึงได้ศึกษาและออกแบบระบบควบคุมกระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ทรูชันด้วย PLC ผ่านหน้าจอ HMI ของระบบ SCADA เพื่อปรับปรุงกระบวนการผลิตของบริษัทกรุงเทพอาหาร จำกัด (มหาชน) ให้เป็นไปตามเป้าหมาย ลดข้อผิดพลาดในการทำงาน ตลอดจนเพิ่มประสิทธิภาพและความปลอดภัยในกระบวนการผลิต

1.2 วัตถุประสงค์ของโครงการ

1. เพื่อศึกษาและออกแบบโปรแกรมควบคุมกระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ทรูชันโดย

ใช้ PLC

2. เพื่อออกแบบส่วนแสดงผลของระบบ SCADA ให้อำนวยความสะดวกแก่ผู้ปฏิบัติงานในการสังเกตและสั่งการระบบ

1.3 ขอบเขตของโครงการงาน

1. ศึกษาและออกแบบโปรแกรมของ PLC เพื่อใช้ควบคุมกระบวนการผลิต โดยใช้ซอฟต์แวร์ Studio 5000 จาก PLC ยี่ห้อ Allen-Bradley

2. ออกแบบส่วนแสดงผลโดยใช้ซอฟต์แวร์ ArchestrA IDE และ Wonderware InTouch เพื่อให้สามารถสั่งการกระบวนการผลิตและแสดงสถานะการทำงานของอุปกรณ์ผ่านทางหน้าจอ HMI ได้

3. เชื่อมต่อส่วนแสดงผลของระบบ SCADA กับส่วนควบคุมคือโปรแกรม PLC ผ่านทาง OPC (OLE for Process Control) คือซอฟต์แวร์ RSLinx Classic และ System Management Console เพื่อทดสอบและแก้ไขการทำงานของส่วนแสดงผลกับโปรแกรม PLC ให้สอดคล้องกัน

1.4 วิธีดำเนินงาน

1. ศึกษาการใช้งานซอฟต์แวร์สำหรับสร้างโปรแกรมควบคุม และสร้างกราฟิกของหน้าจอ HMI
2. ศึกษากระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ทรูชัน
3. ออกแบบและสร้างกราฟิกของหน้าจอ HMI
4. ออกแบบแผนผังแสดงขั้นตอนการทำงานของโปรแกรมควบคุม
5. สร้างโปรแกรมควบคุมของ PLC ตามรูปแบบในแผนผัง
6. เชื่อมต่อส่วนแสดงผล HMI ของระบบ SCADA กับตัวควบคุม ผ่านทาง OPC
7. ทดสอบและปรับปรุงการทำงานของระบบ

ตารางที่ 1.1 แผนการดำเนินงาน

แผนการดำเนินงาน	สัปดาห์	วัน-เดือน-ปี															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
ศึกษาการใช้งานซอฟต์แวร์ที่เกี่ยวข้อง	■	■	■														
ศึกษากระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ทรูชั่น				■	■												
ออกแบบและสร้างกราฟิกของหน้าจอ HMI					■	■	■										
ออกแบบแผนผังแสดงขั้นตอนการทำงานของโปรแกรมควบคุม							■	■									
สร้างโปรแกรมควบคุมของ PLC ตามรูปแบบในแผนผัง									■	■	■	■					
ทดสอบและปรับปรุงการทำงานของระบบ												■	■	■			
จัดทำรายงานสหกิจศึกษา														■	■	■	■

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ผู้ปฏิบัติงานสามารถสังเกตและสั่งการกระบวนการได้สะดวกผ่านทางหน้าจอ HMI
2. ลดโอกาสการเกิดข้อผิดพลาดในกระบวนการทั้งต่อคน เครื่องจักร และวัตถุดิบ ทำให้กระบวนการเป็นไปตามเป้าหมาย ลดต้นทุนที่อาจสูญเสียไปจากความผิดพลาด เช่น เวลา ค่าซ่อมบำรุง เป็นต้น
3. ช่วยเพิ่มความปลอดภัยในการดำเนินงาน
4. เพิ่มประสิทธิภาพในกระบวนการผลิต ซึ่งจะนำไปสู่การเพิ่มผลิตผลและคุณภาพของผลิตภัณฑ์

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

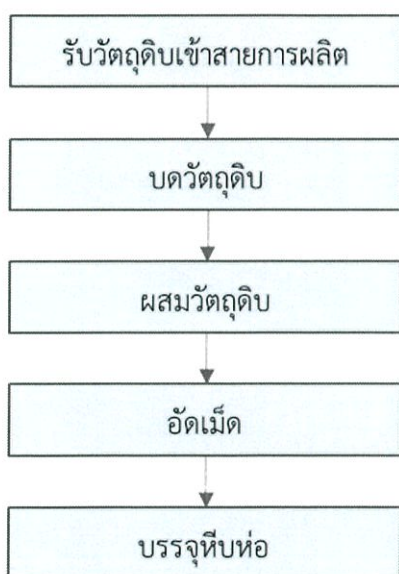
2.1 อุตสาหกรรมอาหารสัตว์สำเร็จรูป

ปัจจุบันการเลี้ยงสัตว์ได้ปรับรูปแบบจากการเลี้ยงแบบครัวเรือน มาเป็นการเลี้ยงเชิงพาณิชย์จึงนิยมใช้อาหารสัตว์สำเร็จรูปมากขึ้น เนื่องจากสะดวกและให้ผลคุ้มค่า ดังนั้นความต้องการอาหารสัตว์สำเร็จรูปจึงมีแนวโน้มสูงขึ้นเรื่อยๆ ส่งผลให้อุตสาหกรรมการผลิตอาหารสัตว์สำเร็จรูปในประเทศขยายตัวเพิ่มขึ้นเป็นลำดับ

การเจริญเติบโตทางด้านเศรษฐกิจส่งผลต่อการพัฒนาของภาคเกษตรกรรมและปศุสัตว์ อาหารสัตว์จึงเป็นปัจจัยสำคัญที่มีส่วนช่วยพัฒนาการปศุสัตว์ หากจะพัฒนาให้แข่งขันกับตลาดในต่างประเทศได้จะต้องลดต้นทุนการผลิต ซึ่งอาหารสัตว์เป็นต้นทุนที่สูงถึงร้อยละ 60-70 ของต้นทุนทั้งหมด ส่งผลถึงอุตสาหกรรมอาหารสัตว์ที่จะต้องพัฒนากระบวนการผลิตให้มีคุณภาพ และประสิทธิภาพที่สูงขึ้น

2.1.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป

โดยทั่วไป กระบวนการที่สำคัญในการผลิตอาหารสัตว์สำเร็จรูปในอุตสาหกรรม มีดังนี้



ภาพที่ 2.1 กระบวนการผลิตอาหารสัตว์สำเร็จรูป

2.1.1.1 การรับวัตถุดิบเข้าสายการผลิต (Raw Material Receiving)

ในกระบวนการผลิตอาหารสัตว์จะมีการใช้วัตถุดิบทั้งที่แปรรูปแล้ว และยังไม่ได้แปรรูป สิ่งสำคัญในขั้นตอนนี้คือ การจัดเก็บวัตถุดิบเพื่อรอการผลิตไม่ให้เกิดวัตถุดิบค้างในคลัง เพราะวัตถุดิบที่ใช้เกือบทั้งหมดเป็นวัตถุดิบสดยังไม่ผ่านการแปรรูป เช่น ปลาป่น เปลือกกุ้ง สามารถเก็บได้เพียงช่วงระยะเวลาสั้นๆ เท่านั้น หากเก็บวัตถุดิบไว้นานจะเกิดการย่อยสลาย และเกิดเชื้อรา แบคทีเรีย ส่งผลเสียต่อคุณภาพของผลิตภัณฑ์ อันจะนำไปสู่ผลกระทบต่อสัตว์ที่บริโภคอาหาร

2.1.1.2 การบดวัตถุดิบ (Grinding)

การบดวัตถุดิบมีความจำเป็นในกรณีที่วัตถุดิบมีขนาดใหญ่และไม่ละเอียด เช่น เมล็ดถั่วเหลือง เมล็ดข้าวโพด มันเส้น ปลายข้าว และเปลือกกุ้ง โดยวัตถุดิบเหล่านี้จะต้องนำมาบดให้ละเอียดด้วยเครื่องบดอาหาร สำหรับวัตถุดิบขนาดเล็ก เช่น ปลาปน รำข้าว แป้งสาลี แป้งมันสำปะหลัง ไม่จำเป็นต้องบดให้ละเอียด แต่ถ้าเป็นการผลิตอาหารในโรงงานอาหารสัตว์น้ำ ซึ่งมีการควบคุมอุณหภูมิอย่างดี จะบดวัตถุดิบทุกประเภทให้ละเอียดเท่ากันทั้งหมด โดยอาจบด 1 หรือ 2 ครั้ง แล้วแต่กรณีว่ามีขนาดเล็กใหญ่ประการใด สำหรับวัตถุดิบจำพวกวิตามิน แร่ธาตุ และกรดอะมิโนสังเคราะห์ จะมีขนาดเล็กละเอียดอยู่แล้ว จึงไม่ต้องนำมาบด อีกทั้งความร้อนที่เกิดจากการบด จะทำให้วิตามิน แร่ธาตุ และกรดอะมิโนสังเคราะห์ถูกทำลายไปอย่างรวดเร็ว ดังนั้นจึงนำไปผสมใช้ได้โดยไม่ต้องบด โดยหลักการแล้วการบดวัตถุดิบอาหารสัตว์ให้ละเอียดนั้น จะช่วยเพิ่มพื้นที่ผิวของวัตถุดิบให้มากขึ้น ซึ่งมีผลให้สามารถผสมวัตถุดิบให้เป็นเนื้อเดียวกันได้ง่าย และช่วยให้อาหารเม็ดที่อัดออกมามีความหนาแน่นและยึดตัวดี นอกจากนี้ยังมีส่วนช่วยให้สัตว์สามารถย่อยและดูดซึมธาตุอาหารไปใช้ประโยชน์ได้ดีขึ้น หรือสัมประสิทธิภาพย่อยอาหารของสัตว์สูงขึ้น

2.1.1.3 การผสมวัตถุดิบ (Mixing)

จุดมุ่งหมายของการผสมวัตถุดิบก็เพื่อให้ส่วนประกอบต่างๆ ผสมคลุกเคล้าเป็นเนื้อเดียวกัน วัตถุดิบทุกชนิดกระจายตามเนื้อของส่วนผสมอย่างทั่วถึง กล่าวคือ หลังจากการผสมแล้วหากแบ่งส่วนผสมมาตรวจหาส่วนประกอบ ส่วนผสมที่เกิดจากการผสมที่ดีจะต้องมีส่วนประกอบของวัตถุดิบครบถ้วนทั้งชนิดและปริมาณตรงตามที่ระบุในสูตรอาหารก่อนการผสม การผสมวัตถุดิบอาจแบ่งออกเป็น 2 ขั้นตอน คือ การผสมวิตามินและแร่ธาตุ และการผสมวัตถุดิบอื่น การผสมวิตามิน และแร่ธาตุมักผสมเตรียมไว้ต่างหากก่อนการผสมวัตถุดิบอื่นๆ เนื่องจากวิตามินและแร่ธาตุที่จะใส่เป็นส่วนผสมในอาหารมีปริมาณน้อย จึงนิยมผสมวิตามินหรือแร่ธาตุกับวัตถุดิบชนิดอื่นเป็นวิตามินรวมหรือแร่ธาตุรวมก่อน แล้วจึงนำวิตามินรวมและแร่ธาตุรวมไปผสมกับวัตถุดิบอื่นตามต้องการ วิธีการนี้ช่วยให้ส่วนผสมของอาหารมีวิตามินและแร่ธาตุตามต้องการ และยังทำให้วิตามินและแร่ธาตุซึ่งเป็นส่วนผสมที่มีปริมาณน้อยสามารถกระจายตัวทั่วเนื้ออาหารได้

2.1.1.4 การอัดเม็ด (Pelleting)

การอัดเม็ดมีวัตถุประสงค์เพื่อต้องการให้วัตถุดิบที่ผ่านการผสมแล้วเปลี่ยนรูปเป็นอาหารเม็ด ซึ่งมีคุณสมบัติเหมาะสมแก่การนำไปใช้เลี้ยงสัตว์ เนื่องจากการอัดเม็ดช่วยลดความเป็นฝุ่นของอาหาร และไม่ทำให้คุณค่าจากอาหารที่มีชิ้นส่วนเล็กหรือเป็นผงที่ฟุ้งกระจายได้สูญหายไป การเก็บอาหารใช้พื้นที่น้อยลง เนื่องจากมีความหนาแน่นของเนื้อวัตถุดิบสูง และส่งผลให้อาหารไม่ร่วนง่ายเมื่อเกิดการกระทบกระเทือนจากการขนส่ง นอกจากนี้ ความร้อนที่เกิดขึ้นในขณะอัดเม็ดจะทำให้แบ่งบางส่วนสุกหรือเปลี่ยนโครงสร้างเอนไซม์ของสัตว์ก็ทำการย่อยได้ง่ายขึ้น หรือจุลินทรีย์ทำงานได้มากกว่า แรงอัดทำให้ผนังเซลล์ของพืชแตกจึงช่วยให้ย่อยได้ดีขึ้น เพิ่มการใช้ประโยชน์ของส่วนที่เป็นเยื่อใยมากขึ้น อีกทั้งยังช่วยป้องกันไม่ให้สัตว์เลือกกินวัตถุดิบที่ชอบเท่านั้น เพราะในอาหารเม็ดมีวัตถุดิบทั้งหมดครบถ้วนในปริมาณที่สม่ำเสมอ

2.1.1.5 การบรรจุหีบห่อ (Packing)

หลังจากอาหารเม็ดแห้งแล้ว และมีอุณหภูมิใกล้เคียงกับอุณหภูมิห้อง จึงพร้อมที่จะได้รับการบรรจุเข้าหีบห่อเพื่อเก็บหรือจำหน่ายต่อไป ในการบรรจุจะมีการบันทึกรายละเอียดที่จำเป็นบนหีบห่อด้วย เช่น รหัส วันบรรจุ น้ำหนักอาหาร และคุณค่าทางอาหาร สิ่งที่สำคัญที่สุดในขั้นตอนนี้คือ

สถานที่จัดเก็บอาหารต้องมีความร้อนและความชื้นไม่สูงเกินไป เพราะจะทำให้อาหารมีราเกิดขึ้น วัสดุที่ใช้บรรจุควรมีคุณสมบัติพิเศษในการป้องกันความชื้นไม่ให้ซึมผ่านไปสัมผัสกับอาหารได้ ทั้งนี้เพื่อป้องกันไม่ให้อาหารเสื่อมคุณภาพเร็ว

2.1.2 วิธีการอัดเม็ดอาหารสัตว์

การอัดเม็ดอาหารสัตว์นั้น มีวัตถุประสงค์เพื่อให้ได้อาหารที่มีลักษณะทางกายภาพและทางเคมีเหมาะสมกับความต้องการและพฤติกรรมการกินอาหารของสัตว์ต่างชนิดกัน ซึ่งสามารถแบ่งวิธีการอัดเม็ดได้เป็น 2 แบบ ตามชนิดของอาหารเม็ดที่ต้องการ ได้แก่

2.1.2.1 การอัดเม็ดอาหารแบบจมน้ำ (Sinking Pellets)

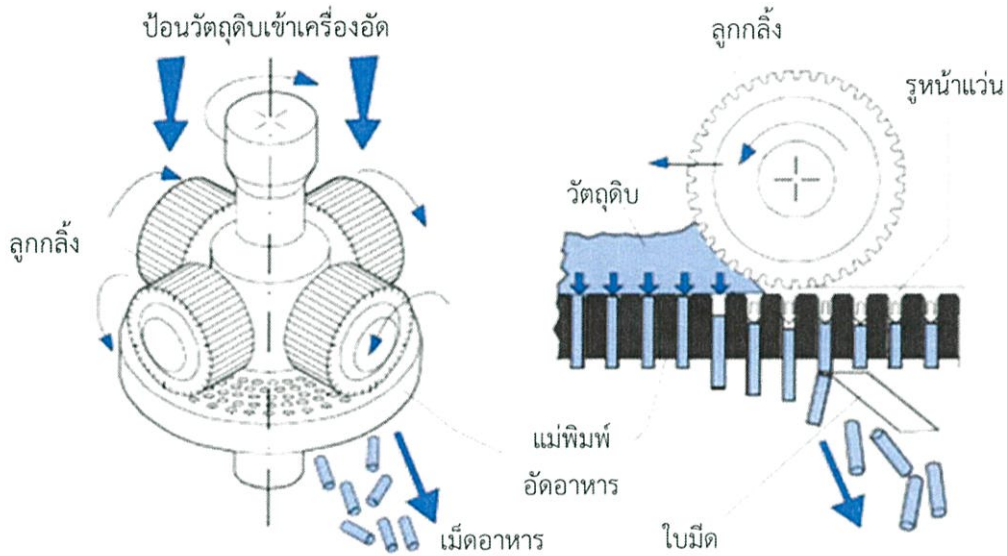
การอัดเม็ดอาหารแบบจมน้ำ ใช้กับปลาที่กินสัตว์หรือกินอาหารตามพื้น และสัตว์กลุ่มครัสเตเชีย โดยจะใช้เครื่องอัดเม็ดแบบ Steam Pelleting ซึ่งใช้ความร้อน ความดัน และความชื้น วัตถุดิบจะถูกลำเลียงผ่านห้องผสมน้ำหรือไอน้ำ (Steam Condition Chamber) ทำให้ความชื้นเพิ่มขึ้นประมาณ 15-18% และมีอุณหภูมิประมาณ 85-90°C ความร้อนจากไอน้ำทำให้แบ่งกลายเป็นเจลช่วยให้อาหารจับตัวกันดีขึ้น จากนั้นอาหารจะเคลื่อนที่มายังห้องอัดเม็ด (Pelleter) โดยอาหารผสมจะเคลื่อนที่มาระหว่างลูกกลิ้งและหน้าแวนซึ่งหมุนในทิศทางเดียวกัน อาหารจะถูกลูกกลิ้งอัดผ่านรูหน้าแวนออกมาเป็นแท่งตามขนาดของรูที่หน้าแวน จากนั้นจึงถูกตัดให้มีขนาดสั้นหรือยาวขึ้นกับระยะห่างระหว่างใบมีดและหน้าแวน ลักษณะการทำงานของหน้าแวนและใบมีดแสดงดังภาพที่ 2.2 ขนาดของเม็ดอาหารส่วนใหญ่ประมาณ 2.5-6 มิลลิเมตร ขึ้นกับขนาดและชนิดของสัตว์ที่จะกิน ความยาวของเม็ดอาหารประมาณ 1.5-2 เท่าของเส้นผ่าศูนย์กลางเม็ดอาหาร เมื่ออัดอาหารออกมาภายนอกจะมีความชื้นประมาณ 10% ทำให้แห้งโดยวิธีใช้ความร้อน การอัดเม็ดอาหารด้วยวิธีนี้ทำลายโครงสร้างอาหารน้อยกว่าวิธี Extrusion แต่มีข้อควรระวังคือ ต้องมีปริมาณไขมันที่เหมาะสม ถ้ามีน้อยเกินไปประมาณ 2-3% จะทำให้อัดอาหารยาก และหากมีมากเกินไปกว่า 10% จะทำให้อาหารไม่คงรูป ในสูตรอาหารที่มีปริมาณไขมันมากนิยมสเปรย์ไขมันเคลือบผิวอาหารหลังจากอัดเม็ดอาหารแล้ว

2.1.2.2 การอัดเม็ดอาหารแบบลอยน้ำ (Floating Pellets)

การอัดเม็ดอาหารแบบลอยน้ำใช้กับปลากินพืช โดยจะใช้เครื่องอัดเม็ดอาหารแบบ Extruder ซึ่งจะให้อาหารสุกด้วยความร้อน ความดัน และการเสียดสีของอาหารกับพื้นที่ของเครื่องอัดเม็ด วัตถุดิบส่วนใหญ่จะประกอบด้วยแป้งและโปรตีน โดยวัตถุดิบเหล่านี้จะถูกลำเลียงผ่านห้องผสมไอน้ำ (Mixing Cylinder หรือ Conditioning Cylinder) ให้ความชื้น คลุกให้เข้ากัน ซึ่งความชื้นในระยะนี้จะมีค่าประมาณ 25% ในช่วงนี้อาหารจะสุกเนื่องจากไอน้ำความร้อนสูง ส่งผลให้อาหารมีกลิ่นหอม และถูกย่อยได้ง่ายขึ้น จากนั้นจะถูกส่งเข้าส่วนที่ใช้สำหรับอัดเม็ดคือเครื่อง Extruder ซึ่งมีลักษณะดังภาพที่ 2.3 ภายในเครื่องประกอบด้วยสกรูตัวใหญ่ ทำหน้าที่เป็นตัวอัดอาหารผ่านท่อเกลียว ซึ่งหมุนด้วยความเร็ว 200 - 500 รอบต่อนาที สกรูจะเป็นตัวบีบอัดและเกิดความร้อนจากการเสียดสีภายในอาหาร ซึ่งทำให้อาหารมีลักษณะขึ้นเหนียวและสุกด้วยเวลาสั้นประมาณ 15 - 30 วินาที จากนั้นจึงถูกบีบอัดผ่านหน้าแวนที่ปลายกระบอกรัด โดยจะอัดออกเป็นเส้นยาวหรือเป็นแท่งขึ้นอยู่กับความร้อนที่เกิดขึ้นจากการอัด ความหนาแน่นของเนื้ออาหาร และขนาดของหน้าแวนปลายกระบอกรัด เมื่ออาหารสัมผัสกับอุณหภูมิและความดันภายนอกเครื่องอัดที่ต่ำกว่าอย่างกระทันหัน น้ำในเม็ดอาหารจะระเหยออก ทำให้มีลักษณะเนื้อฟูขึ้น มีรูพรุนในเนื้ออาหารสูง จึงความหนาแน่นน้อยกว่าน้ำ และสามารถลอยน้ำได้

โดยปกติการอัดอาหารด้วย Extruder จะมีความชื้นสูงกว่าการอัดอาหารแบบ Steam Pelleted ซึ่งมีความชื้นประมาณ 18-21% ดังนั้นอาหารแบบนี้ควรอบให้แห้งด้วยความร้อนเพื่อสามารถเก็บรักษาได้นานขึ้น โดยอาหารที่อบแล้วควรมีความชื้นประมาณ 10-12%

ข้อดีของอาหารแบบลอยน้ำคือทำให้สังเกตการกินอาหารของปลาได้ และขั้นตอนการอัดเม็ดทำให้แป้งเกือบทุกชนิดช่วยเพิ่มอัตราการย่อยได้ของปลา แต่มีข้อเสียคือในการผลิตต้องใช้ อุณหภูมิและความดันสูง ทำให้วิตามินบางชนิดสลายตัวได้ เช่น วิตามินซี ดังนั้นต้องมีการคำนวณอัตราการสูญเสียวิตามินในระหว่างการผลิตด้วย



ภาพที่ 2.2 การทำงานของหน้าแวนและไบเม็ดในเครื่อง Steam Pelleting



ภาพที่ 2.3 เครื่อง Extruder

2.2 ความรู้พื้นฐานเกี่ยวกับพีแอลซี (PLC)

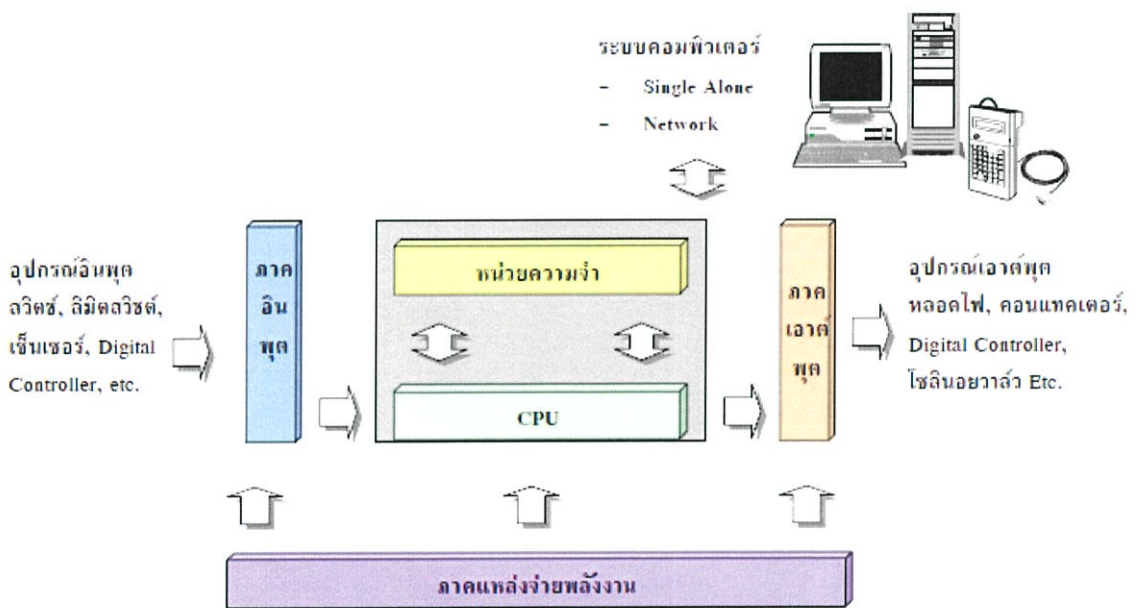
2.2.1 ความหมายของพีแอลซี

พีแอลซี (PLC) ย่อมาจาก Programmable Logic Controller ซึ่งหมายถึงตัวควบคุมเชิงตรรกะที่สามารถโปรแกรมได้ ในยุโรปมักเรียกว่า “ซีควนเซอร์” (Sequencer Controller) เรียกย่อๆว่า SC แต่สมาคมผู้ผลิตอุปกรณ์ไฟฟ้าญี่ปุ่น (JEMA) มักเรียก “พีซี” (PC) หรือ “พีแอลซี” (PLC)

พีแอลซีเป็นอุปกรณ์ควบคุมการทำงานของเครื่องจักรหรือกระบวนการต่าง ๆ โดยภายในจะมีไมโครโพรเซสเซอร์ทำหน้าที่เป็นตัวสั่งการ มีหน่วยความจำสำหรับเก็บโปรแกรมควบคุมเครื่องจักรหรือกระบวนการ มีอินพุตที่สามารถต่อเข้ากับตัวตรวจรู้หรือสวิตซ์ต่าง ๆ และมีส่วนของเอาต์พุตที่สามารถต่อเข้ากับอุปกรณ์กระทำการของเครื่องจักร เช่น วงจรแม่เหล็ก (Magnetic) วาล์วต่างๆ เป็นต้น รูปแบบของการควบคุมเครื่องจักรสามารถเปลี่ยนแปลงได้โดยการเปลี่ยนโปรแกรมตั้งงานพีแอลซี โดยจะมีฟังก์ชันต่างๆ สำหรับการควบคุมอยู่ภายใน เช่น รีเลย์ รีจิสเตอร์ ตัวตั้งเวลา ตัวนับ และตัวเปรียบเทียบ เป็นต้น การเชื่อมต่อของอุปกรณ์เหล่านี้สามารถทำได้โดยซอฟต์แวร์

2.2.2 โครงสร้างของพีแอลซี

โครงสร้างภายในของพีแอลซีแต่ละส่วนจะร่วมกันทำงานเป็นระบบควบคุม โดยจะประกอบไปด้วยส่วนสำคัญ 5 ส่วนดังนี้



ภาพที่ 2.4 โครงสร้างของพีแอลซี

2.2.2.1 ซีพียู (CPU; Central Processing Unit)

ซีพียูหรือหน่วยประมวลผลกลาง ทำหน้าที่ประมวลผลการทำงานตามคำสั่งของส่วนต่างๆ ตามที่ได้รับมา ผลลัพธ์จากการประมวลผลก็จะถูกส่งออกไปยังส่วนต่างๆ ตามที่ระบุไว้ด้วยคำสั่ง โดยความเร็วในการประมวลผลขึ้นอยู่กับขนาดของซีพียู และขนาดของโปรแกรมด้วย ปกติแล้วซีพียูจะใช้ไมโครโพรเซสเซอร์ขนาดตั้งแต่ 4 บิต, 8 บิต, 16 บิต, 32 บิต, 64 บิต หรือ 120 บิต มาทำงาน โดยที่ซีพียูแต่ละขนาดก็จะมีขีดความสามารถจำกัดไม่เท่ากัน จึงทำให้ PLC แต่ละรุ่นมีความสามารถต่างกัน หรือภายใน PLC บางรุ่นจะใช้ไมโครโพรเซสเซอร์ถึง 2 ตัวช่วยกันทำงาน เวลาการประมวลผลก็จะเร็วกว่า PLC ที่ใช้ไมโครโพรเซสเซอร์เพียงตัวเดียว

2.2.2.2 หน่วยความจำ (Memory Unit)

หน่วยความจำเป็นอุปกรณ์ที่ใช้เก็บโปรแกรมและข้อมูลต่างๆ ของ PLC กรณีที่สั่ง RUN PLC ก็จะนำเอาโปรแกรมและข้อมูลในหน่วยความจำมาประมวลผลการทำงาน สำหรับหน่วยความจำที่ใช้งานอยู่ใน PLC มีด้วยกัน 2 แบบคือ

1) หน่วยความจำชั่วคราว (RAM: Random Access Memory)

ใช้เก็บโปรแกรมและข้อมูลที่ทำงานจากการสั่ง RUN/STOP PLC โปรแกรมและข้อมูลที่สร้างขึ้นโดยผู้ใช้ก็ถูกเก็บในส่วนนี้ด้วย คุณสมบัติของ RAM เมื่อไม่มีไฟเลี้ยงจะทำให้โปรแกรมและข้อมูลหายไปทันที ดังนั้นภายใน 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 โดย ROM แบ่งออกเป็น 3 ชนิด ได้แก่

2.1) PROM (Programmable ROM)

จัดเป็น ROM รุ่นแรก เขียนข้อมูลลงชิปได้เพียงครั้งเดียว ถ้าเขียนข้อมูลไม่สมบูรณ์ ชิพก็จะเสียทันทีไม่สามารถนำกลับมาเขียนใหม่ได้อีก

2.2) EPROM (Erasable Programmable ROM)

ถูกพัฒนาขึ้นมาจาก PROM ทำให้สามารถเขียนข้อมูลลงชิปได้หลายครั้ง เพียงแค่นำชิปไปฉายแสงอุลตราไวโอเลตก็จะเป็นการลบข้อมูลด้วยสัญญาณทางไฟฟ้าได้เลย จึงทำให้เกิดความสะดวกสบายมากขึ้น แต่เรื่องเวลาในการเข้าถึงข้อมูลก็ยังช้ากว่า RAM อยู่

2.3) EEPROM (Electrical Erasable Programmable ROM)

เป็นอุปกรณ์ที่รวมคุณสมบัติของ ROM และ RAM ไว้ด้วยกัน เนื่องจากไม่ต้องใช้เครื่องมือพิเศษในการเขียนและลบโปรแกรม โดยสามารถใช้งานได้เหมือนกับ RAM แต่ไม่ต้องใช้แบตเตอรี่สำรอง ราคาจึงแพงกว่า

2.2.2.3 ภาคอินพุต (Input Unit)

ทำหน้าที่รับสัญญาณจากอุปกรณ์ภายนอกเข้ามา แล้วแปลงให้เป็นสัญญาณที่เหมาะสม ก่อนส่งให้หน่วยประมวลผลต่อไป ซึ่งอุปกรณ์ต่างๆ ที่นำมาต่อกับภาคอินพุตได้ สามารถจำแนกเป็นกลุ่ม ได้ดังภาพที่ 2.5 โดยแต่ละกลุ่มอุปกรณ์จะมีวิธีต่อวงจรเข้าภาคอินพุตที่แตกต่างกันออกไป เวลาใช้งานจึงต้องศึกษาข้อมูลเพิ่มเติมของอุปกรณ์แต่ละชนิดก่อน เพื่อความเข้าใจขั้นตอนการทำงาน และสามารถต่อวงจรได้ถูกต้อง

2.2.2.4 ภาคเอาต์พุต (Output Unit)

ภาคเอาต์พุตของ PLC ทำหน้าที่ส่งสัญญาณออกไปขับโหลดชนิดต่างๆ ตามเงื่อนไขที่โปรแกรมไว้ ชนิดของโหลดที่สามารถนำมาต่อกับภาคเอาต์พุตแยกออกเป็นกลุ่มได้ดังภาพที่ 2.6 โดยแต่ละกลุ่มก็จะควบคุมลักษณะของงานแตกต่างกันไปตามคุณสมบัติของอุปกรณ์นั้นๆ การต่อวงจรเข้าภาคเอาต์พุตก็มีมาตรฐานทางอุตสาหกรรมกำกับอยู่เช่นกัน จึงทำให้ผู้ใช้ไม่ต้องใช้อุปกรณ์เสริมมาก เพียงแต่ดูรายละเอียดการต่อให้เข้าใจก็เพียงพอแล้ว



push button switch



proximity switch



encoder



digital signal controller



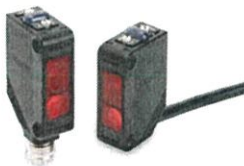
limit switch



relay



temperature controller



photoelectric sensor



digital thumbwheel switch



Inverter

ภาพที่ 2.5 อุปกรณ์ที่ใช้ต่อกับภาคอินพุตของพีแอลซี



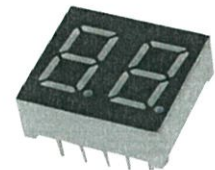
lamp indicator



buzzer



relay



7-segment display



magnetic contactor



temperature controller



drive

ภาพที่ 2.6 โหลดที่ใช้ต่อกับภาคเอาต์พุตของพีแอลซี

2.2.2.5 ภาคแหล่งจ่ายพลังงาน (Power Supply Unit)

ภาคแหล่งจ่ายพลังงาน จะทำหน้าที่จ่ายพลังงานให้กับอุปกรณ์ภายใน PLC ได้แก่ อุปกรณ์ไอซี, ไฟเลี้ยงวงจรกำหนดการทำงานแบบต่างๆ เป็นต้น นอกจากนี้ยังจ่ายพลังงานเลี้ยงวงจรที่จะนำมาต่อกับ PLC ทั้งภาคอินพุตและเอาต์พุตด้วย ซึ่งแหล่งจ่ายพลังงานของ PLC จะแบ่งออกเป็น 2 ชุด

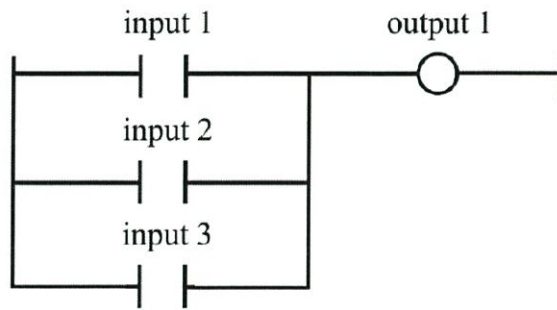
ชุดหนึ่งสำหรับอุปกรณ์และวงจรภายในแต่ละโมดูลต่างๆ ของ PLC อีกชุดหนึ่งเป็นตัวจ่ายพลังงาน 24 VDC (Service Unit 24 VDC) สำหรับการต่อวงจรภาคอินพุตหรือเอาต์พุตก็ได้ ในการใช้งานจริง แหล่งจ่ายจะถูกออกแบบมา 2 ลักษณะตามโครงสร้างภายนอกของ PLC คือ แหล่งจ่ายชนิดที่รวมอยู่ในตัว PLC กับชนิดที่แยกออกมาเป็นโมดูล (Module)

2.2.3 ภาษาที่ใช้เขียนพีแอลซี

พีแอลซีแต่ละยี่ห้อจะใช้ภาษาในการเขียนโปรแกรมเพื่อสั่งงานแตกต่างกัน ซึ่งตามมาตรฐาน IEC1131-3 ได้แบ่งมาตรฐานภาษาต่างๆ ออกเป็น 5 แบบคือ

2.2.3.1 LD (Ladder Diagram)

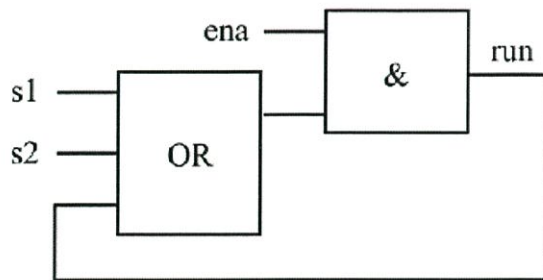
เป็นภาษาที่เขียนอยู่ในรูปกราฟิก ซึ่งมีพื้นฐานมาจากวงจรควบคุมแบบรีเลย์และวงจรไฟฟ้า โดยแลตเตอร์ไดอะแกรมจะประกอบด้วยราง (Rail) ทั้งซ้ายและขวาของไดอะแกรม เพื่อใช้สำหรับเชื่อมต่ออุปกรณ์ที่เป็นสวิตช์หน้าสัมผัส ซึ่งจะเป็นทางผ่านของกระแสและมีขดลวดหรือคอยล์เป็นเอาต์พุต



ภาพที่ 2.7 ภาษา Ladder Diagram

2.2.3.2 FBD (Function Block Diagram)

เป็นภาษาที่เขียนในรูปแบบกราฟิกเช่นเดียวกัน และเชื่อมต่อกันเป็นโครงข่าย โดยมีพื้นฐานมาจากหลักการของลอจิกไดอะแกรม (Logic Diagram)



ภาพที่ 2.8 ภาษา Function Block Diagram

2.2.3.3 IL (Instruction List)

เป็นภาษาที่เขียนในรูปข้อความจะมีลักษณะคล้ายกับภาษาแอสเซมบลี (Assembly) และภาษาเครื่อง (Machine Code)

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, *)
			(* result := a1 + (a2*(a3 - a4) * a5) *)
	ADD	a6	(* a1 + (a2 * (a3 - a4) * a5) + a6 *)
	ST	res	(* store current result in res *)

ภาพที่ 2.9 ภาษา Instruction List

2.2.3.4 ST (Structure Text)

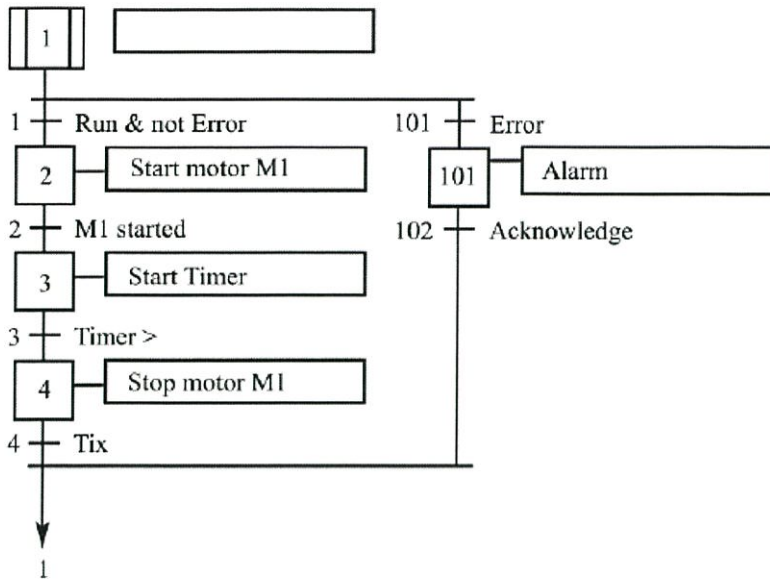
เป็นภาษาระดับสูง โดยมีพื้นฐานมาจากภาษาปาสคัล (Pascal) ซึ่งประกอบด้วย นิพจน์ และคำสั่ง โดยคำสั่งทั่วไปจะเกี่ยวกับการเลือกการทำงาน เช่น IF...THEN...ELSE และ คำสั่งเกี่ยวข้องกับการทำงานซ้ำ เช่น FOR, WHILE เป็นต้น

```
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
```

ภาพที่ 2.10 ภาษา Structure Text

2.2.3.5 SFC (Sequential Flow Chart)

เป็นภาษาที่รองรับการเขียนโปรแกรมที่มีโครงสร้างการทำงานแบบเป็นลำดับหรือ ซีควเอนซ์ ซึ่งจะประกอบด้วย Step (การปฏิบัติการย่อย) และ Transition (เงื่อนไขที่กำหนดให้ปฏิบัติงานตามคำสั่งย่อย) นอกจากนี้ยังสามารถกำหนดลักษณะการทำงานเป็นแบบ Linear, Alternative และ Parallel Step Sequence ได้ด้วย



ภาพที่ 2.11 ภาษา Sequential Flow Chart

2.3 ความรู้พื้นฐานเกี่ยวกับระบบสกาตา (SCADA)

2.3.1 ความหมายของสกาตา

สกาตา (SCADA) ย่อมาจาก Supervisory Control and Data Acquisition เป็นระบบตรวจสอบและวิเคราะห์ข้อมูลแบบ Real-time ใช้ในการตรวจสอบสถานะ ตลอดจนควบคุมการทำงานของระบบควบคุมในอุตสาหกรรมและงานวิศวกรรมต่างๆ เช่น งานด้านโทรคมนาคมสื่อสาร การประปา การบำบัดน้ำเสีย การจัดการด้านพลังงาน อุตสาหกรรมการกลั่นน้ำมันและก๊าซ อุตสาหกรรมเคมี อุตสาหกรรมประกอบรถยนต์ การขนส่ง กระบวนการนิวเคลียร์ในโรงไฟฟ้า เป็นต้น ตัวอย่างการใช้งาน เช่น ใช้ SCADA ตรวจสอบข้อมูลการรั่วไหลของของเหลวที่เกิดขึ้นในท่อขนส่งจากตัวตรวจจับ แล้วส่งสัญญาณแจ้งเตือนให้พนักงานทราบ โดยส่งข้อมูลสู่ส่วนกลางของระบบ SCADA เป็นต้น นอกจากนี้ SCADA อาจทำหน้าที่คำนวณและประมวลผลข้อมูลที่ได้จากฮาร์ดแวร์ต่าง ๆ เช่น PLC, Controller, DCS, RTU แล้วแสดงข้อมูลทางหน้าจอ หรือส่งสัญญาณควบคุมฮาร์ดแวร์ดังกล่าว เช่น หากอุณหภูมิของอุปกรณ์สูงเกินพิกัด ให้ทำการปิดอุปกรณ์นั้น เป็นต้น โดยสั่งงานผ่าน PLC หรือ Controller ที่ติดต่อยู่ ทั้งนี้ SCADA สามารถเก็บรวบรวมข้อมูลที่ได้จากระบบควบคุมทั้งหมดไว้ในฐานข้อมูลเพื่อให้พนักงานหรือโปรแกรมอื่นๆ สามารถนำไปใช้งานได้ SCADA ได้เข้าไปมีส่วนในงานควบคุมทั้งขนาดเล็กและใหญ่ที่ต้องการแสดงผล แลกเปลี่ยนข้อมูล หรือควบคุมระบบต่างๆ จากส่วนกลาง เพื่อการทำงานของระบบรวมทั้งสัมพันธ์กัน มองเห็นภาพรวมได้อย่างชัดเจน และมีความรวดเร็วต่อเหตุการณ์ต่างๆ ที่เกิดขึ้น ระบบ SCADA ในปัจจุบันมีความสามารถในการสื่อสาร ควบคุม และประมวลผลข้อมูลจาก I/O ของอุปกรณ์เช่น PLC, DCS, RTU ได้ถึงระดับที่เกินหนึ่งแสน I/O แล้ว และได้รับการพัฒนาให้มีความสามารถรองรับความต้องการใหม่ๆ ของผู้ใช้งานอย่างต่อเนื่องตลอดมา

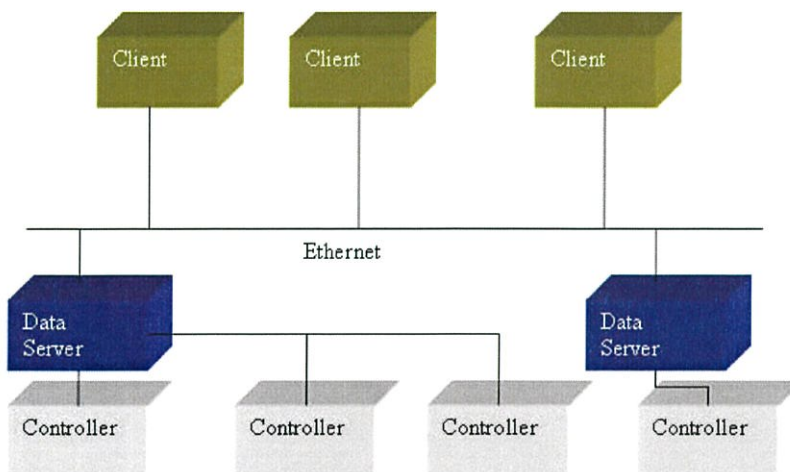
2.3.2 โครงสร้างของระบบสกาดา

2.3.2.1 โครงสร้างด้านฮาร์ดแวร์ (Hardware Architecture)

SCADA แบ่งตามโครงสร้างฮาร์ดแวร์ได้สองระดับคือ Client และ Data Server หรือเรียกสั้น ๆ ว่า Server โดยที่ Client คือคอมพิวเตอร์ที่รับและส่งข้อมูลไปยัง Data Server โดยฝั่ง Client นี้จะแสดงผลการทำงานของระบบควบคุม เช่น แสดงเป็นกราฟิก กราฟแบบต่อเนื่อง หรือระบบแจ้งเตือนเมื่อเกิดเหตุการณ์ฉุกเฉินหรือต้องการแจ้งเตือน เป็นต้น ฝั่ง Client สามารถสั่งงานควบคุมไปยัง Data Server เพื่อให้ส่งสัญญาณไปยัง PLC, DCS หรือ Controller อีกทอดหนึ่ง โดย Data Server จะติดต่อกับ PLC, DCS, Controller หรือ RTU ต่าง ๆ เพื่อรับสัญญาณและส่งไปยัง Client ควบคู่ไปกับการรับ การร้องขอจาก Client มาใช้ควบคุมอุปกรณ์ PLC และ Controller ต่างๆ ซึ่ง Client และ Data Server ส่วนใหญ่ติดต่อกันผ่านระบบเครือข่าย Ethernet ดังภาพที่ 2.12 และ Controller จะติดต่อกับ อุปกรณ์ Field Instrument ต่าง ๆ เช่น เซ็นเซอร์ รีเลย์ เพื่อนำสัญญาณมาให้ Data Server

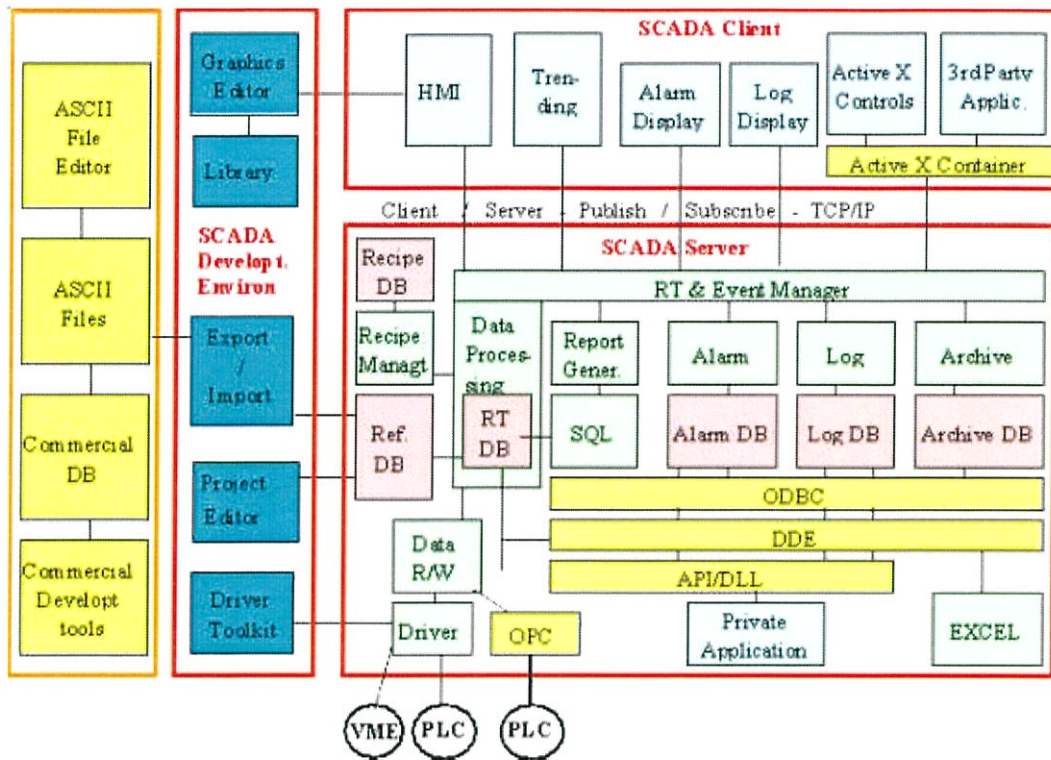
2.3.2.2 โครงสร้างด้านซอฟต์แวร์ (Software Architecture)

เนื่องจาก SCADA ใช้เทคโนโลยีในการสื่อสารกับฮาร์ดแวร์ที่แตกต่างกันไปตามผู้ผลิต เช่น การใช้ Driver เฉพาะของผู้ผลิตเพื่อสื่อสารกับ PLC, DCS ซึ่งในปัจจุบันมีการกำหนดมาตรฐานกลางคือ OPC ขึ้นมาเพื่อยุติปัญหาการใช้เทคโนโลยีเฉพาะด้านการสื่อสาร นอกจากนั้นยังมีความสามารถในการบริการข้อมูลให้กับ Client ที่รวดเร็วและมีเสถียรภาพ ทำให้ SCADA Server สามารถติดต่อกับ PLC หรือ Controller ผ่าน Driver หรือ OPC ก็ได้ โดยที่ OPC และ Driver จะรับคำสั่งแบบ Read/Write เพื่ออ่านข้อมูลจาก PLC หรือเขียนข้อมูลเพื่อสั่งงานไปยัง PLC โดยโครงสร้างด้านซอฟต์แวร์ของ SCADA สามารถแสดงได้ดังภาพที่ 2.13



ภาพที่ 2.12 โครงสร้างด้านฮาร์ดแวร์ของระบบ SCADA

SCADA Server จะทำหน้าที่จัดการข้อมูล RTDB (Real Time Data Base) ที่ได้จาก PLC แล้วส่งให้กับ SCADA Client โดย SCADA Server บางประเภทจะติดต่อกับ SCADA Client ผ่าน DDE Server ซึ่งทำให้สามารถนำข้อมูลจาก PLC เข้าสู่โปรแกรมเช่น MS Excel หรือ โปรแกรม Client อื่นๆ ที่ติดต่อกับ DDE Server ได้ และในบางครั้ง SCADA Server อาจถูกออกแบบให้ทำหน้าที่



ภาพที่ 2.13 โครงสร้างด้านซอฟต์แวร์ของระบบ SCADA

ตรวจจับ Alarm และเก็บไว้ใน Alarm DB หรือเก็บข้อมูลที่เป็น Historian ไว้ใน Log DB เป็นต้น เพื่อส่งให้ Alarm Display และ Log Display ทางฝั่ง SCADA Client ต่อไป

สำหรับส่วน Development Environment จะขึ้นอยู่กับกรอบการออกแบบซอฟต์แวร์ของผู้ผลิตแต่ละราย ซึ่งโดยทั่วไปมักจะมีเครื่องมือในการสร้างและจัดการกราฟิก (Graphic Editor) เครื่องมือในการจัดการโปรเจกต์ที่สร้างขึ้น (Project Editor) มีเครื่องมือในการนำเข้าและส่งออก Text File ที่เก็บค่า Configuration ของการติดต่อกับ Driver หรือ OPC Server ไว้

2.3.2.3 โครงสร้างด้านการสื่อสาร (Communications)

การสื่อสารระหว่าง Client-Server จะสื่อสารผ่านโปรโตคอลโดยทั่วไป เช่น TCP/IP โดย Client จะติดต่อกับพารามิเตอร์หรือ Tag ภายใน Server ที่บริการข้อมูลด้วยรูปแบบที่แตกต่างกันไปตามผู้ผลิต เช่น มีการส่งค่าจาก Server เมื่อค่า I/O ของ PLC เปลี่ยนแปลง เป็นต้น

การสื่อสารระหว่าง Server กับอุปกรณ์นั้น Server จะตรวจสอบค่าจากอุปกรณ์ตามช่วงเวลาที่ใช้กำหนดไว้ (Defined Polling Rate) โดยอาจจะต่างกันไปตามพารามิเตอร์ประเภทต่างๆ ตัว Controller จะส่งค่าพารามิเตอร์ตามที่ถูกร้องขอให้กับ Data Server พร้อมค่าเวลาขณะนั้น (Time Stamp) การสื่อสารกับอุปกรณ์ของ Data Server อาจเป็นการสื่อสารแบบ Modbus, Profibus, CAN bus เป็นต้น ขึ้นอยู่กับมาตรฐานการสื่อสารของอุปกรณ์นั้นๆ ซึ่งในปัจจุบันมีการสร้าง OPC Server ที่สนับสนุนการติดต่อด้วยมาตรฐานต่างๆ เพิ่มขึ้นมากมายจนครอบคลุมอุปกรณ์ทุกประเภท

2.3.2.4 โครงสร้างด้านอินเทอร์เฟซ (Interface)

การติดต่อกันระหว่าง Data Server กับอุปกรณ์ หรือระหว่าง Data Server กับ Client หรือแม้กระทั่งระหว่าง Data Server เองนั้นมีการผลิตเป็น Driver ออกมามากมายตามเทคนิค

เฉพาะของแต่ละผู้ผลิต ต่อมาจึงมีการกำหนดมาตรฐานของอินเทอร์เฟซขึ้นมาเป็น OPC (OLE for Process Control) ซึ่งมีความรวดเร็วในการสื่อสารและบริการข้อมูลโดยมีการจัดตั้ง OPC Foundation ขึ้นเป็นองค์กรหลักในการกำหนดมาตรฐานและถ่ายทอดเทคโนโลยีให้แก่สมาชิก OPC จึงเป็นมาตรฐานกลางที่เปิดกว้างมากที่สุด

การติดต่อกับฐานข้อมูลภายนอกของ SCADA Software นั้น มีการสร้างให้สามารถติดต่อได้ผ่าน ODBC (Open Data Base Connectivity), OLEDB (Linking and Embedding Data Base), DDE (Dynamic Data Exchange) เป็นต้น เพื่อให้สามารถแลกเปลี่ยนหรือเก็บข้อมูลไว้ในฐานข้อมูลรูปแบบต่างๆ ในปัจจุบันมีการพัฒนาให้สามารถติดต่อกับโปรแกรม ERP ต่างๆ ได้ด้วย เช่น SAP เป็นต้น

2.3.2.5 โครงสร้างความสามารถในการขยายระบบ (Scalability)

Scalability คือความสามารถในการรองรับและต่อขยายระบบ SCADA กับส่วนต่างๆ เช่น I/O ของอุปกรณ์ Controller และจำนวนเครื่อง SCADA Client ที่เพิ่มขึ้น หรือการต่อพ่วงกับระบบ SCADA ของยี่ห้ออื่นๆ เป็นต้น ถ้าหาก Data Server เป็นแบบ Driver ที่สร้างด้วยเทคโนโลยีเฉพาะในการติดต่อกับอุปกรณ์ ก็จะทำให้การต่อขยายเป็นเรื่องยุ่งยาก เพราะ Driver บางประเภทสามารถติดต่อได้เฉพาะ SCADA Software บางยี่ห้อเท่านั้น ซึ่งปัจจุบันได้มีมาตรฐานกลางคือ OPC เพื่อแก้ไขปัญหานี้

2.3.2.6 โครงสร้างการสำรองระบบ (Redundancy)

SCADA Software ส่วนใหญ่มีความสามารถในการทำสำรองระบบของ Data Server คือเมื่อ Data Server เกิดความขัดข้องก็จะสั่งงานให้ Data Server อีกตัวหนึ่งทำงานแทน โดยจะมีการกำหนด Configuration ไว้ที่ Client ว่าจะให้เลือกติดต่อกับ Data Server ตัวไหนเมื่อเกิดความขัดข้อง และในบางครั้งโมดูลที่ทำหน้าที่จัดการด้าน Redundancy นี้ อาจจะทำหน้าที่อีกประการหนึ่งคือ เป็นจุดพักข้อมูลที่รับมาจาก Data Server เพื่อนำไปส่งให้กับ Client ต่างๆ เพราะในกรณีที่มี Client จำนวนมากติดต่อกับ Data Server ตัวเดียว อาจมีความล่าช้าในการบริการข้อมูลของ Data Server เพราะต้องให้บริการข้อมูล Client ให้ครบจำนวนก่อนที่จะไปรับข้อมูลใหม่จากอุปกรณ์มาได้ ดังนั้นโมดูลที่ทำหน้าที่ Redundant จึงทำหน้าที่เป็นจุดรับข้อมูล แล้วช่วยส่งต่อให้ Client ต่างๆอีกทอดหนึ่ง Data Server จะได้ทำหน้าที่บริการข้อมูลให้แก่โหนดเพียงจุดเดียว จึงมีความรวดเร็วในการบริการข้อมูล

2.3.3 หน้าที่การทำงานของสกาตา (SCADA Functionality)

2.3.3.1 การเข้าถึงพารามิเตอร์ของอุปกรณ์

หมายถึงความสามารถในการเข้าถึงกลุ่มของพารามิเตอร์ในอุปกรณ์เช่น I/O ของ PLC เป็นต้น ความสามารถของ Data Server ในการกำหนดว่าพารามิเตอร์ใดอ่านได้อย่างเดียว เขียนได้อย่างเดียว หรือทั้งอ่านทั้งเขียน เป็นต้น

2.3.3.2 ระบบแสดงผลแบบ HMI (Human-Machine Interface)

คือความสามารถในการแสดงผลการทำงานของอุปกรณ์ในรูปแบบกราฟิก, ข้อความ, สัญลักษณ์, แผนภาพ เป็นต้น โดยสามารถเชื่อมโยงลักษณะการเปลี่ยนแปลงของกราฟิกเหล่านี้กับพารามิเตอร์จาก Data Server ได้ ความสามารถในการสั่งงานผ่านระบบกราฟิกเช่น การปิด/เปิดสวิตช์บนจอมอนิเตอร์ ส่งผลไปยัง I/O ของ PLC ความสามารถในการจัดการกราฟิก เช่น การย่อ-ขยาย การกำหนดการเคลื่อนไหวแบบต่างๆ เช่น การหมุน การเคลื่อนที่แบบซิกแซกตามสัญญาณของ Data Server การ

แสดงผลสัญญาณในรูปแบบมิเตอร์และเกจวัดแบบต่างๆ การนำเข้ากราฟิกประเภทต่างๆ การจัดแบ่งเลย์เออร์ เป็นต้น

2.3.3.3 ระบบแสดงกราฟสัญญาณแบบต่อเนื่อง (Trending)

Trending คือความสามารถในการพล็อตกราฟต่อเนื่องกันไปบนจอภาพเพื่อแสดงค่าสัญญาณจาก Data Server โดยสามารถพล็อตสัญญาณได้หลายสัญญาณ เช่น 8 – 24 สัญญาณพร้อมกันในหน้าต่างเดียว เพื่อให้สามารถเปรียบเทียบสัญญาณที่พล็อตได้ และไม่จำกัดว่าจะสร้างหน้าต่างพล็อตจำนวนเท่าใด หรืออาจมีความสามารถในการซูมสัญญาณที่พล็อต และหยุดการพล็อตเพื่อเลื่อนดูค่าที่พล็อตในแต่ละช่วงเวลาได้ด้วยตัวของผู้ใช้งานเอง นอกจากนี้การพล็อตอาจสามารถเลือกได้ว่าจะให้เป็นแบบใด เช่น Time Plot, Logarithmic Plot, Strip Chart, Bar Chart, Circular, X-Y Plot เป็นต้น นอกจากนี้ผู้ผลิตบางรายยังสามารถนำค่า Historian หรือข้อมูลสัญญาณที่เก็บไว้ในฐานข้อมูลออกมาพล็อตได้อีกด้วย โดย Trending Module นี้ อาจเป็นแบบ ActiveX Control คือสามารถนำไปใช้งานในแอปพลิเคชันอื่นที่สนับสนุนการนำเข้า ActiveX ได้

2.3.3.4 ระบบแจ้งเตือน (Alarm)

SCADA Software ส่วนใหญ่มีระบบแจ้งเตือนโดย Alarm Display จะรับสัญญาณมาจาก Alarm DB ในฝั่ง SCADA Server โดย Alarm DB สามารถจะทำการกำหนดคอนฟิกูเรชันว่าจะนำสัญญาณตัวใดมาเป็นพารามิเตอร์ในการแจ้งเตือนบ้าง และมีการแบ่งระดับของ Priority Limit อย่างเป็นต้น นอกจากนี้ ระบบแจ้งเตือนยังสามารถเก็บข้อมูลการแจ้งเตือนไว้ในฐานข้อมูลประเภทต่างๆ ได้เช่น MS SQL Server, MS Access, Oracle, MS Excel เป็นต้น และบางยี่ห้อสามารถแสดงออกมาเป็นรายงานในรูปแบบตารางหรือแผนภูมิได้อีกด้วย

2.3.3.5 การทำงานแบบ Automation

เป็นความสามารถที่ SCADA ทำหน้าที่ต่างๆตามที่กำหนด เช่น ส่งอีเมล แสดงข้อความแบบ Instance Message บนหน้าจอ เปิดไปยังหน้าจออื่นๆ เก็บข้อมูลลงฐานข้อมูล เปิดโปรแกรมหรือรันคำสั่งสคริปต์ตามสัญญาณที่ได้รับจาก Data Server และข้อกำหนดที่สร้างขึ้น

2.3.4 การสร้างและพัฒนาระบบสกาตา (Application Development)

2.3.4.1 การกำหนดคอนฟิกูเรชัน

การกำหนดคอนฟิกูเรชัน ขั้นแรกต้องกำหนดว่าจะติดต่อกับพารามิเตอร์หรือ Tag ใดบ้างจาก Data Server ดังนั้นจะต้อง Define หรือสร้าง Tag ที่ Data Server ก่อนว่า Tag แต่ละตัวหมายถึง Address ที่เท่าใดของอุปกรณ์ (PLC, DCS, RTU, Controller ต่างๆ) โดยทั่วไปสามารถนำเข้าคอนฟิกูเรชันไฟล์ที่สร้างไว้ก่อนได้ และสามารถ Export ไปยัง Data Server อื่นๆ จากนั้นโปรแกรมย่อยอื่นๆของ SCADA Software ฝั่ง Client จึงทำคอนฟิกูเรชันตามหน้าที่การทำงานของตนเอง เช่น โมดูลที่มีหน้าที่แสดงผลกราฟก็ต้องกำหนดว่ากราฟนั้นๆ จะเชื่อมโยงกับ Tag ใดจาก Data Server ส่วนโมดูลที่มีหน้าที่แจ้งเตือนก็ต้องทำคอนฟิกูเรชันว่าจะนำ Tag ใดมาเป็นสัญญาณแจ้งเตือน และกำหนดระดับสัญญาณ Limit เป็นต้น

2.3.4.2 เครื่องมือในการพัฒนา (Development Tool)

เครื่องมือในการสร้างและพัฒนาระบบ SCADA โดยทั่วไปจะประกอบด้วย

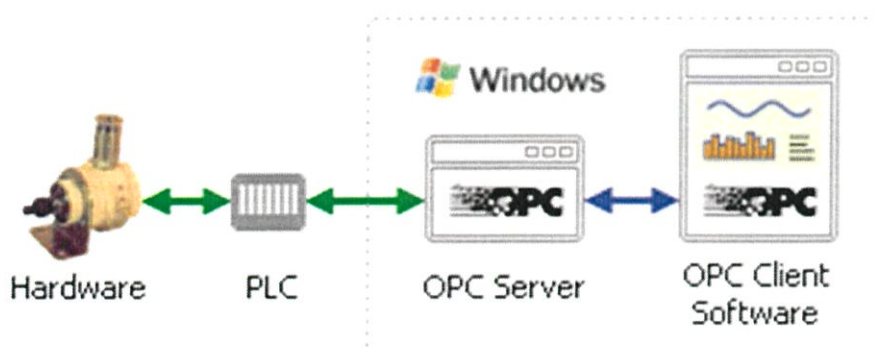
1) เครื่องมือในการสร้างระบบกราฟิก ที่ประกอบด้วยเครื่องมือวาดภาพ, เครื่องมือกำหนดเอฟเฟกต์พิเศษต่างๆ, โลบรารีของกราฟิกสำเร็จรูปในอุตสาหกรรมด้านต่างๆ

- 2) เครื่องมือในการสร้าง Trending
- 3) เครื่องมือในการสร้างระบบ Alarm
- 4) เครื่องมือในการกำหนดการติดต่อกับฐานข้อมูลเพื่อทำการเก็บรวบรวมข้อมูลของ Trending และ Alarm ลงไว้ในฐานข้อมูล
- 5) เครื่องมือในการช่วยสร้าง Script เช่น Java Script, VB Script
- 6) เครื่องมือจัดการด้านความปลอดภัย การแบ่งระดับ User และขอบเขตการใช้งานของ User
- 7) เครื่องมือในการสร้าง Web Application เพื่อให้สามารถควบคุม และตรวจสอบระบบควบคุมผ่าน Web Browser ได้

2.4 ความสำคัญของ OPC

OPC ย่อมาจาก Object Linking and Embedding for Process Control เป็นมาตรฐานที่ใช้ในการติดต่อสื่อสารกันของซอฟต์แวร์ ซึ่งทำให้ซอฟต์แวร์บนคอมพิวเตอร์สามารถสื่อสารกับอุปกรณ์ทางอุตสาหกรรมต่างๆ ได้ โดยการใช้งานจะอยู่ในรูปแบบของ Server และ Client กล่าวคือ OPC Server เป็นซอฟต์แวร์ที่แปลงโปรโตคอลที่ใช้ในการสื่อสารของฮาร์ดแวร์ คือ PLC มาเป็นโปรโตคอลของ OPC ส่วน OPC Client เป็นซอฟต์แวร์ที่ต้องการใช้ข้อมูลจาก OPC Server ในการรับหรือส่งข้อมูลกับฮาร์ดแวร์ต่างๆ ซึ่งรูปแบบการติดต่อสื่อสารโดยใช้ OPC เป็นตัวกลางแสดงดังภาพที่ 2.14

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



ภาพที่ 2.14 การติดต่อสื่อสารระหว่างฮาร์ดแวร์กับซอฟต์แวร์โดยใช้ OPC เป็นตัวกลาง

2.5 ซอฟต์แวร์ที่เกี่ยวข้อง

2.5.1 Studio 5000

Studio 5000 เป็นซอฟต์แวร์สำหรับตั้งค่าและสร้างโปรแกรม เพื่อใช้งานในตัวควบคุมยี่ห้อ Allen-Bradley ตระกูล Logix 5000 ด้วยคุณสมบัติที่ใช้งานง่าย จึงเพิ่มความสะดวกให้แก่ผู้ใช้งานในการออกแบบระบบควบคุม ประหยัดเวลาในการดำเนินงาน และช่วยเพิ่มประสิทธิภาพในการผลิตให้มากยิ่งขึ้น

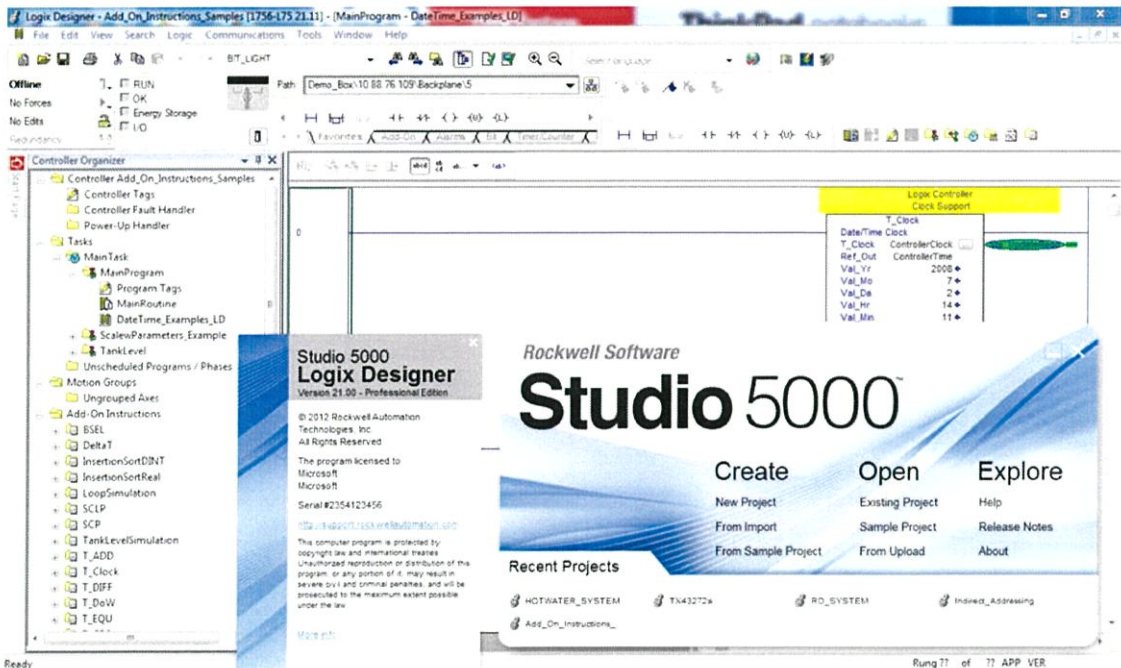
คุณสมบัติของเครื่องคอมพิวเตอร์ที่พร้อมใช้งานโปรแกรม Studio 5000 ควรมีลักษณะดังนี้

2.5.1.1 ความต้องการด้านฮาร์ดแวร์

- คอมพิวเตอร์ประมวลผลด้วยซีพียู Intel Core i5 หรือสูงกว่า
- หน่วยความจำ RAM 8 GB ขึ้นไป
- พื้นที่ว่างที่ต้องการบนฮาร์ดดิสก์ 20 GB
- กราฟิก ทำงานได้ดีกับซอฟต์แวร์ DirectX 9 ที่มีไดรเวอร์ WDDM (Windows Display Driver Model) เวอร์ชัน 1.0 ขึ้นไป
- ซีดีรอมไดรฟ์
- อุปกรณ์ซีเมาส์

2.5.1.2 ความต้องการด้านซอฟต์แวร์ระบบปฏิบัติการ

- 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 (server version of Windows 8)



ภาพที่ 2.15 ซอฟต์แวร์ Studio 5000

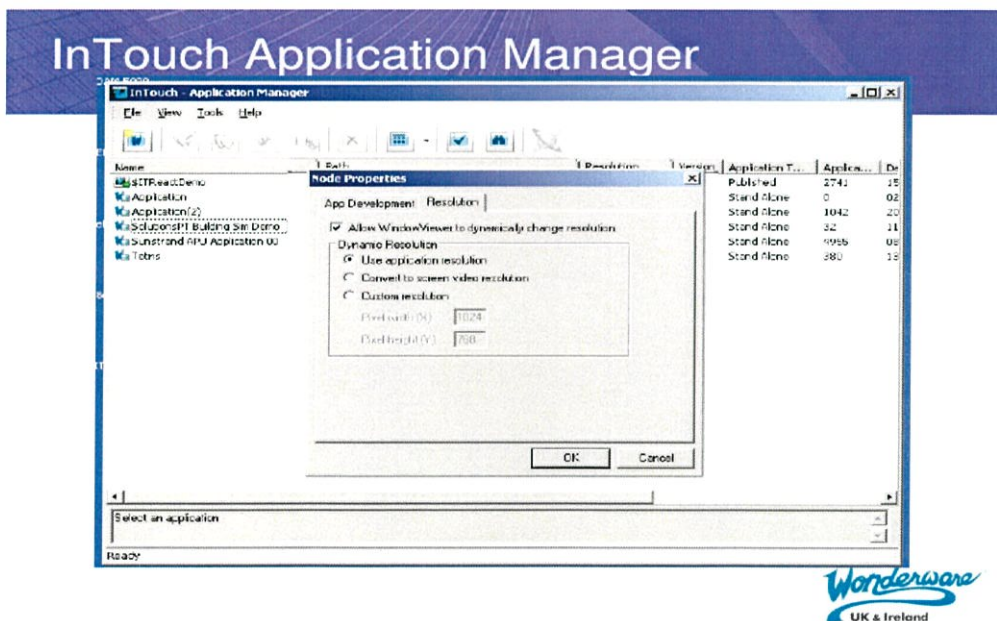
2.5.2 Wonderware InTouch HMI 2014 R2

Wonderware InTouch HMI 2014 R2 เป็นซอฟต์แวร์ HMI จากบริษัท Invensys Process Systems ใช้สำหรับสร้างกราฟิกในอุตสาหกรรม ซึ่งโปรแกรมที่เขียนจะแสดงกระบวนการต่างๆ ในระบบควบคุมผ่านหน้าจอ HMI โดยที่ผู้ใช้หรือผู้ควบคุมไม่จำเป็นต้องอยู่หน้างาน เพื่อสังเกตกระบวนการ ทำให้เกิดความสะดวกต่อผู้ดูแลทั้งยังช่วยลดต้นทุนการจ้างงานอีกด้วย

เนื่องจาก HMI/SCADA เกิดจากความต้องการของผู้ใช้งานในการเข้าไปควบคุมระบบที่มี PLC เป็นตัวควบคุมอยู่ ในขณะที่ PLC มีการควบคุมกระบวนการโดยอัตโนมัติและอยู่กระจัดกระจายกันไปยากที่จะนำข้อมูลแต่ละส่วนมาประมวลผลรวมกันด้วยการเก็บข้อมูลแบบ Manual แต่ HMI/SCADA จะนำข้อมูลจาก PLC ส่งผ่านโครงข่ายการสื่อสารแบบต่างๆ และรวบรวมข้อมูลเข้าด้วยกัน ทำให้สามารถสั่งการได้โดยผู้เชี่ยวชาญ

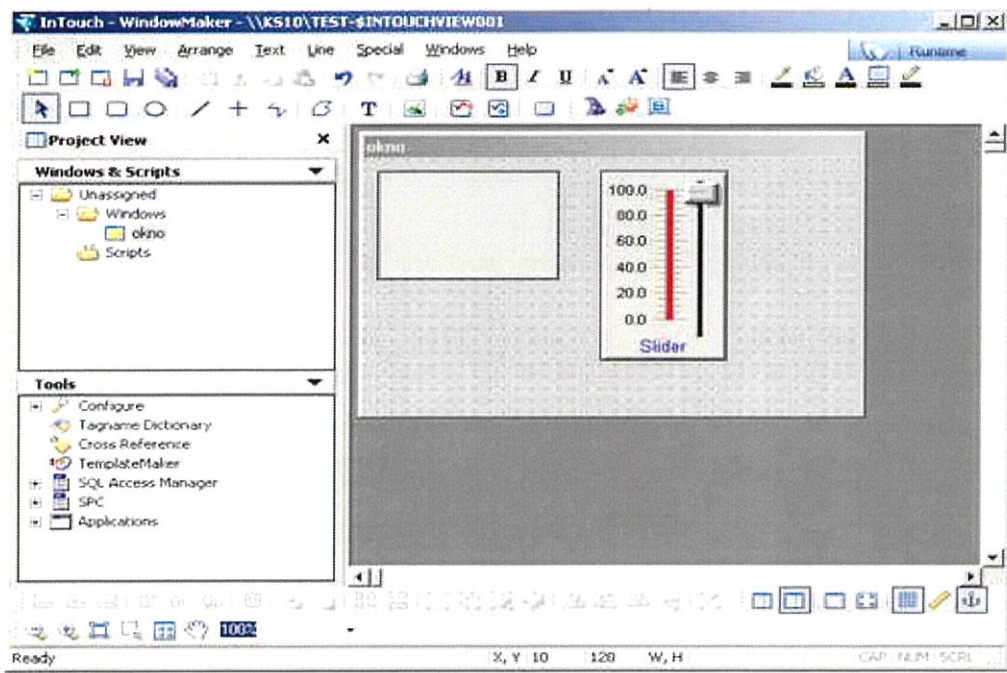
InTouch สามารถทำงานบนระบบปฏิบัติการ Windows ของ Microsoft ได้ และประกอบไปด้วย 3 โปรแกรมที่สำคัญ ได้แก่

2.5.2.1 InTouch Application Manager ใช้บริหารจัดการโปรแกรมประยุกต์ที่ผู้ใช้สร้างขึ้น กำหนดค่าของ WindowViewer กำหนดความละเอียดของกราฟิก และยังมี DBDump และ DBLoad ซึ่งเป็นตัวเก็บฐานข้อมูลของกราฟิกโดยสามารถนำออกมาเป็นไฟล์ Excel ได้ ทำให้บริหารจัดการง่ายขึ้น



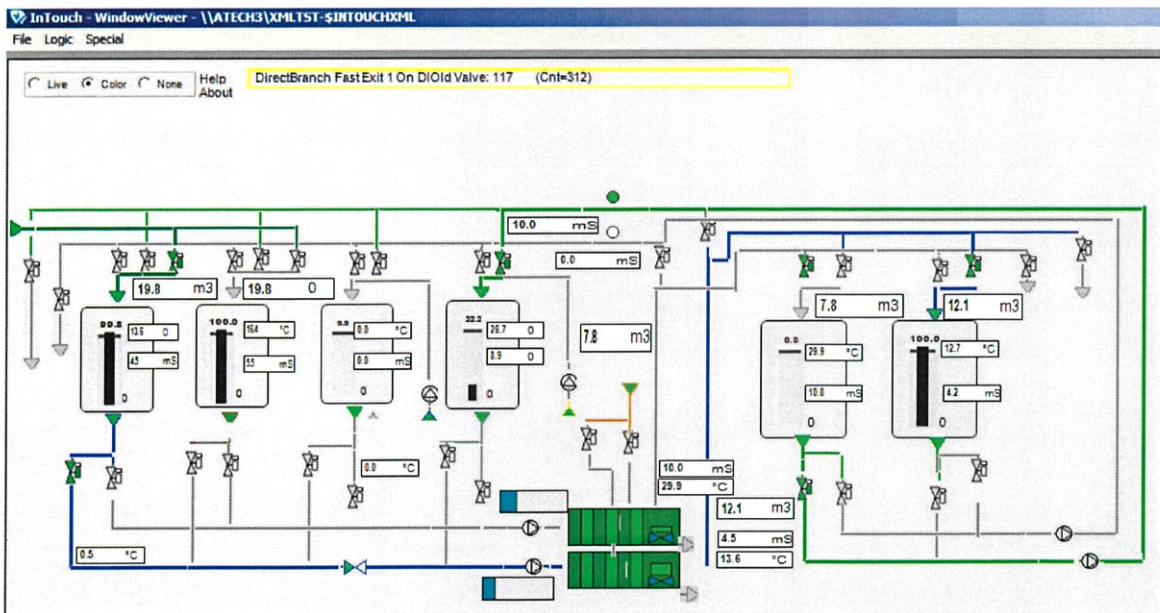
ภาพที่ 2.16 ซอฟต์แวร์ InTouch Application Manager

2.5.2.2 InTouch WindowMaker เป็นโปรแกรมสร้าง กำหนดค่า และแก้ไขกราฟิก โดยมีเครื่องมือในการวาดกราฟิก เขียนสคริป และมี Factory Symbol สำเร็จรูปที่สามารถนำมาใช้ได้อย่างสะดวก นอกจากนี้ ยังใช้ในการกำหนดค่าให้สามารถเชื่อมต่อ Industrial I/O Systems กับ Microsoft Windows Applications อื่นๆ ได้



ภาพที่ 2.17 ซอฟต์แวร์ InTouch WindowMaker

2.5.2.3 InTouch WindowViewer เป็นหน้าต่างแสดงผลเมื่อรันใหม่จาก WindowMaker สามารถบันทึกข้อมูลที่ผ่านมาและรายงานผล ตลอดจนแสดงสัญญาณเตือน เมื่อเกิดความผิดปกติขึ้นกับกระบวนการตามที่โปรแกรมไว้ จึงช่วยลดการสูญเสียจากอุบัติเหตุได้



ภาพที่ 2.18 ซอฟต์แวร์ InTouch WindowViewer

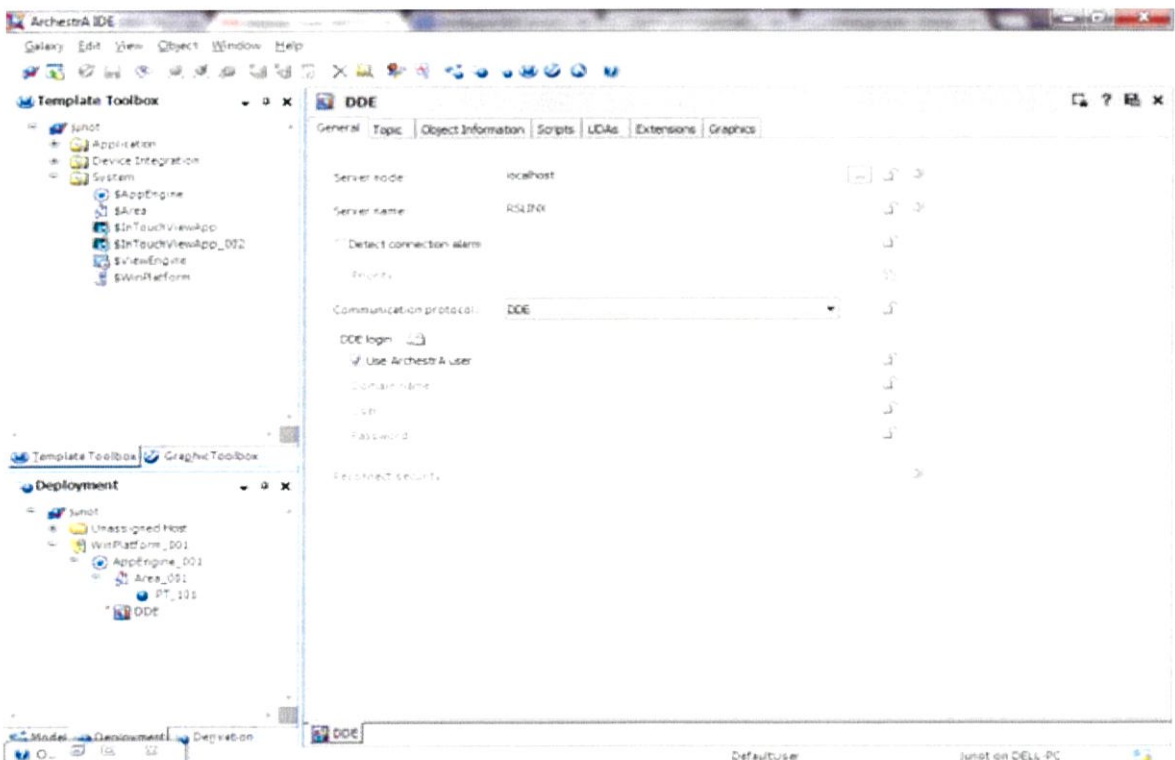
2.5.3 ArchestrA IDE (Integrated Development Environment)

ArchestrA IDE เป็นซอฟต์แวร์ที่จะช่วยสร้างสภาพแวดล้อม (Environment) ในคอมพิวเตอร์ให้เหมาะแก่การเขียนโปรแกรม โดยมีองค์ประกอบหรือตัวช่วยต่างๆที่จะช่วยให้ผู้พัฒนาแอปพลิเคชันสามารถทำงานได้อย่างรวดเร็ว ถูกต้อง แม่นยำ ทั้งยังสามารถตรวจสอบระบบที่จัดทำได้ทั้งเชิงลึกหรือตื้น ทำให้การพัฒนางานต่างๆเกิดความสะดวกมากขึ้น

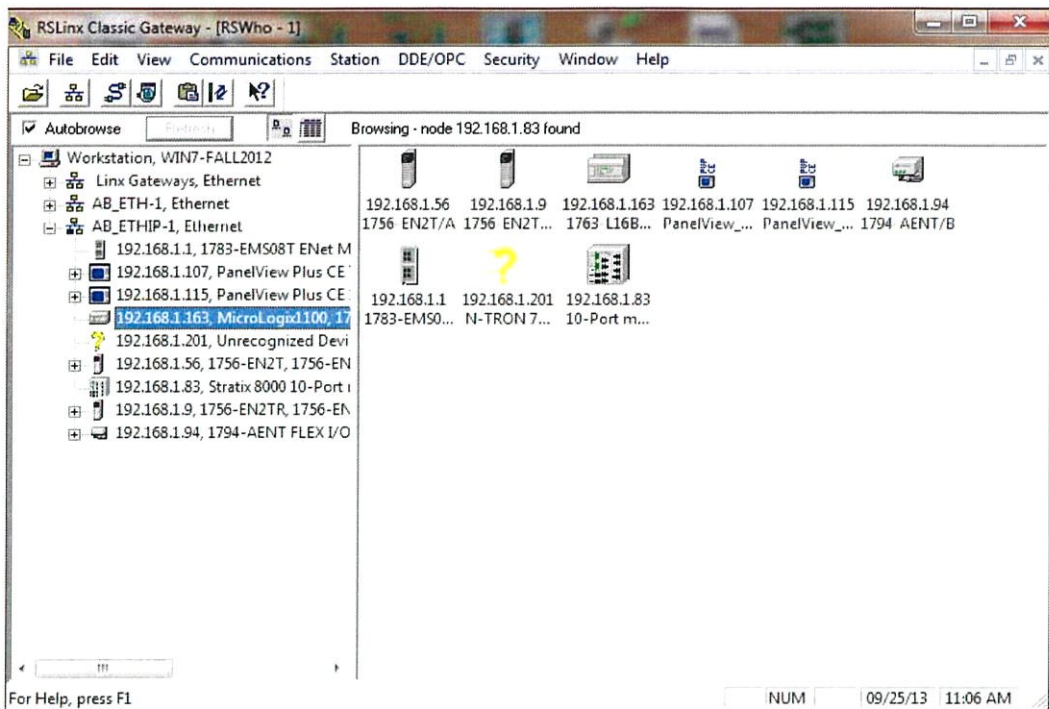
ArchestrA IDE นั้นทำหน้าที่เป็น System Platform กล่าวคือ เป็นระบบที่ใช้เก็บวัตถุ (Object) ต่างๆที่ผู้ใช้สร้างขึ้น ซึ่งประกอบไปด้วยกราฟิกและคำสั่งที่เกี่ยวข้อง ซึ่งระบบนี้สามารถขยายเพิ่มเติมได้ตามการออกแบบของผู้ใช้งาน โดย ArchestrA IDE สามารถทำได้ทั้งการออกแบบ พัฒนา และเก็บวัตถุ เพื่อ Deploy หรือส่งวัตถุไปใช้ในคอมพิวเตอร์ที่ต้องการซึ่งอยู่ในเน็ตเวิร์กเดียวกัน และด้วยคุณสมบัตินี้ จึงมีประโยชน์ต่อการพัฒนาแอปพลิเคชันขนาดใหญ่ ที่มีวัตถุในลักษณะเดียวกันเป็นจำนวนมาก ลดความซ้ำซ้อนในการแก้ไขคุณสมบัติของวัตถุทีละตัว การจัดการวัตถุจึงมีความสะดวกและรวดเร็ว

2.5.4 RSLinx Classic

เป็นซอฟต์แวร์ที่ใช้เชื่อมต่อพีแอลซีกับคอมพิวเตอร์ เพื่อดาวนโหลดโปรแกรมที่มีอยู่ในคอมพิวเตอร์ให้พีแอลซี, อัปโหลดโปรแกรมจากพีแอลซีขึ้นมาตรวจสอบและแก้ไขในคอมพิวเตอร์, สั่ง Run/Stop พีแอลซี หรือทดสอบการทำงานของโปรแกรมที่สร้างขึ้น เป็นต้น ทั้งนี้จะต้องมีไดรเวอร์ซึ่งเป็นโปรแกรมย่อยสำหรับควบคุมอุปกรณ์ภายนอกที่นำมาเชื่อมต่อกับคอมพิวเตอร์หรือเรียกว่า โปรเซสเซอร์ ซึ่งได้แก่พีแอลซีนั่นเอง ไดรเวอร์ที่ใช้ขึ้นอยู่กับคุณสมบัติทางกายภาพของการเชื่อมต่อระหว่างคอมพิวเตอร์กับโปรเซสเซอร์ เช่น RS-232, RS-485, RS-422 หรือ Ethernet เป็นต้น



ภาพที่ 2.19 ซอฟต์แวร์ ArchestrA IDE



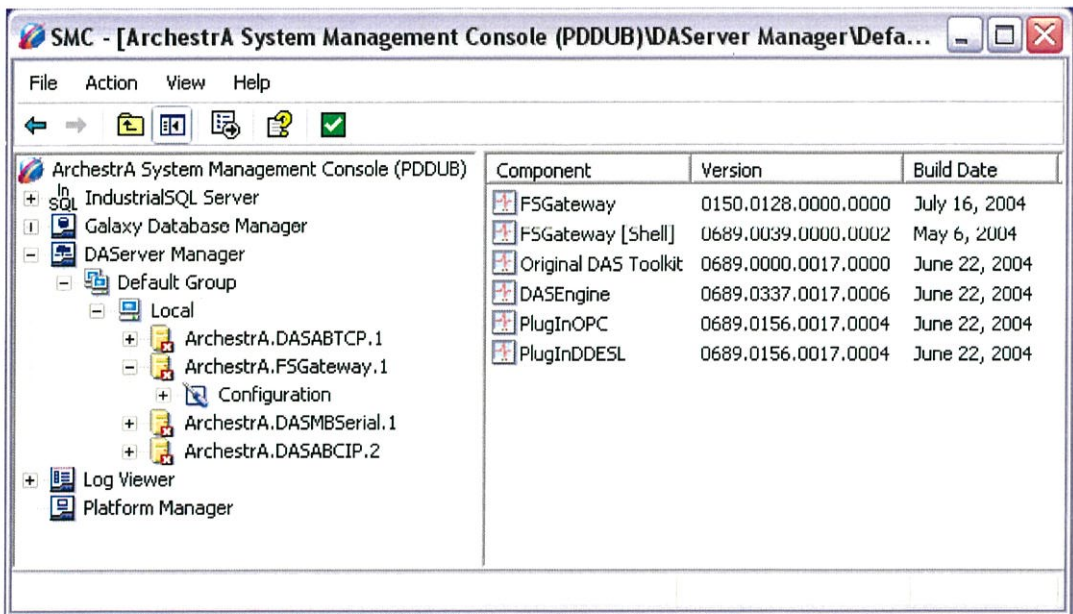
ภาพที่ 2.20 ซอฟต์แวร์ RSLogix Classic

2.5.5 SMC (System Management Console)

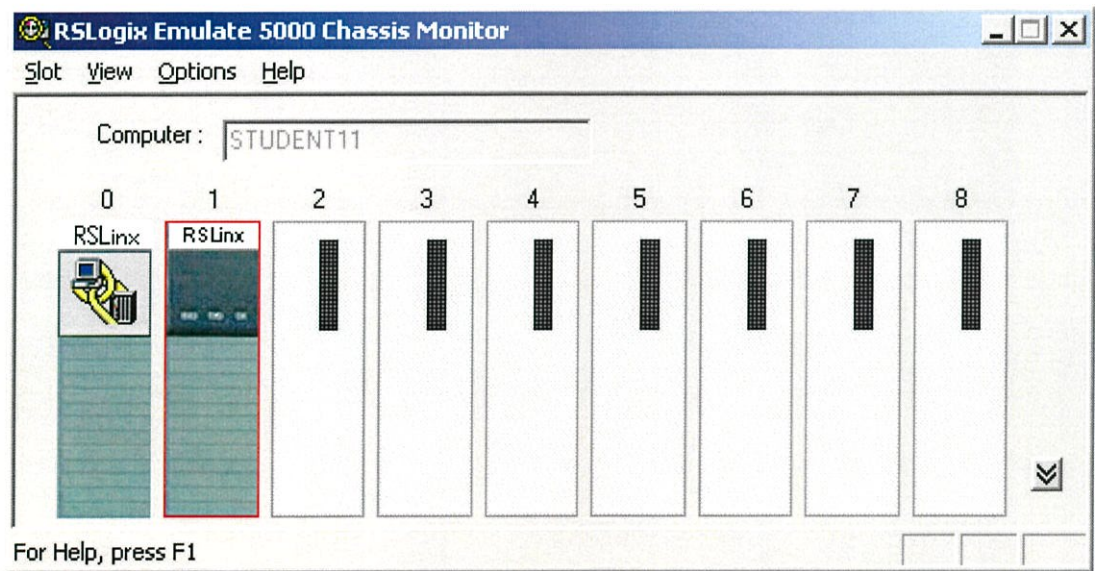
SMC เป็นโปรแกรมที่เป็นตัวกลางสำหรับเชื่อมต่อซอฟต์แวร์ต่างๆ ให้สามารถติดต่อสื่อสารกันได้ หรือเป็น OPC Software นั้นเอง โดยสามารถเชื่อมต่อโปรแกรมจาก PLC กับส่วนแสดงผลกราฟิกได้ ด้วยการตั้งค่า Topic Name ให้สอดคล้องกันทั้ง Server และ Client นอกจากนี้ SMC สามารถวิเคราะห์ความผิดพลาด และยังเป็นเครื่องมือการจัดการเครือข่ายคอมพิวเตอร์, บริการ และส่วนประกอบอื่นๆ

2.5.6 RSLogix Emulate 5000 Chassis Monitor

RSLogix Emulate 5000 เป็นโปรแกรมจำลองตัวควบคุมยี่ห้อ Allen-Bradley ตระกูล Logix 5000 ซึ่งได้แก่ ControlLogix, CompactLogix, FlexLogix, SoftLogix5800 และ DriveLogix เพื่อจำลองฟังก์ชันการทำงานของพีแอลซีโดยไม่ต้องใช้พีแอลซีจริง จึงมีประโยชน์ต่อการทดสอบและแก้ไขโปรแกรมก่อนการทำ Commissioning และ Startup กระบวนการ



ภาพที่ 2.21 ซอฟต์แวร์ System Management Console



ภาพที่ 2.22 ซอฟต์แวร์ RSLogix Emulate 5000 Chassis Monitor

บทที่ 3

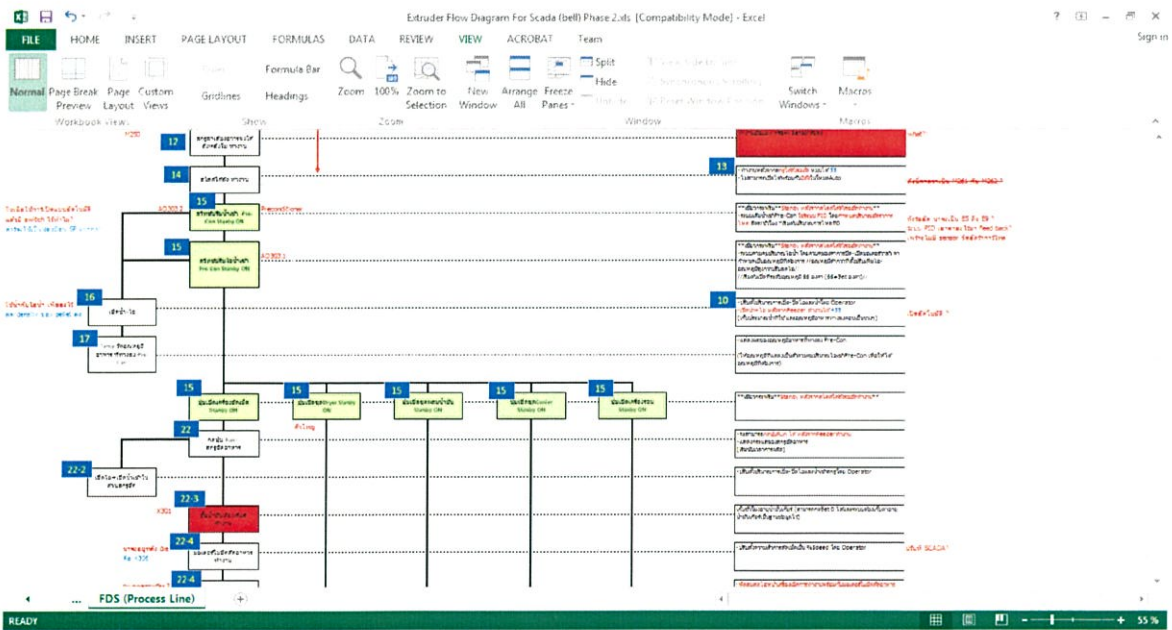
การออกแบบโปรแกรมควบคุมจำลองพีแอลซี

ในการออกแบบโปรแกรมควบคุมและส่วนแสดงผลของกระบวนการอัดเม็ดอาหารสัตว์ด้วยวิธีเอ็กซ์-ทูล์ซึ้นนั้น จะต้องศึกษากระบวนการให้เข้าใจว่ามีขั้นตอนการทำงานอย่างไร เครื่องจักรที่เกี่ยวข้องแต่ละชนิดมีคุณสมบัติและข้อจำกัดใดบ้าง เพื่อให้สามารถออกแบบโปรแกรมได้อย่างถูกต้อง โดยมีการทดลองฟังก์ชันการทำงานของระบบควบคุมด้วยซอฟต์แวร์จำลองพีแอลซี ตรวจสอบและแก้ไขจนได้โปรแกรมควบคุมและส่วนแสดงผลตรงตามความต้องการของลูกค้า ซึ่งวิธีการดำเนินงานดังกล่าวสามารถแบ่งเป็นขั้นตอนได้ดังนี้

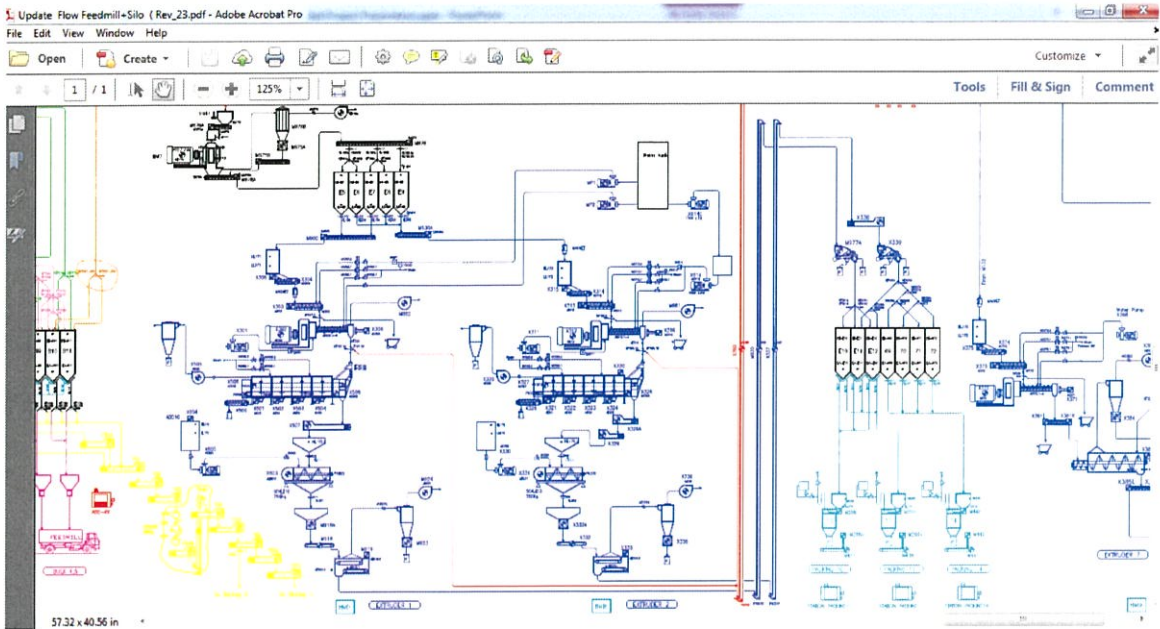
3.1 ออกแบบโปรแกรมควบคุม

3.1.1 ศึกษากระบวนการ

ศึกษาขั้นตอนการทำงานของกระบวนการอัดเม็ดแบบเอ็กซ์ทูล์ซึ้น จากแผนผังแสดงความต้องการของลูกค้า ประกอบกับแผนภาพ P&ID และคำอธิบายชื่ออุปกรณ์ต่างๆ เพื่อทราบถึงขั้นตอนของกระบวนการ เจริญไซ อินพุต และเอาต์พุตที่เกี่ยวข้อง



ภาพที่ 3.1 แผนผังแสดงความต้องการของลูกค้า



ภาพที่ 3.2 แผนภาพ P&ID

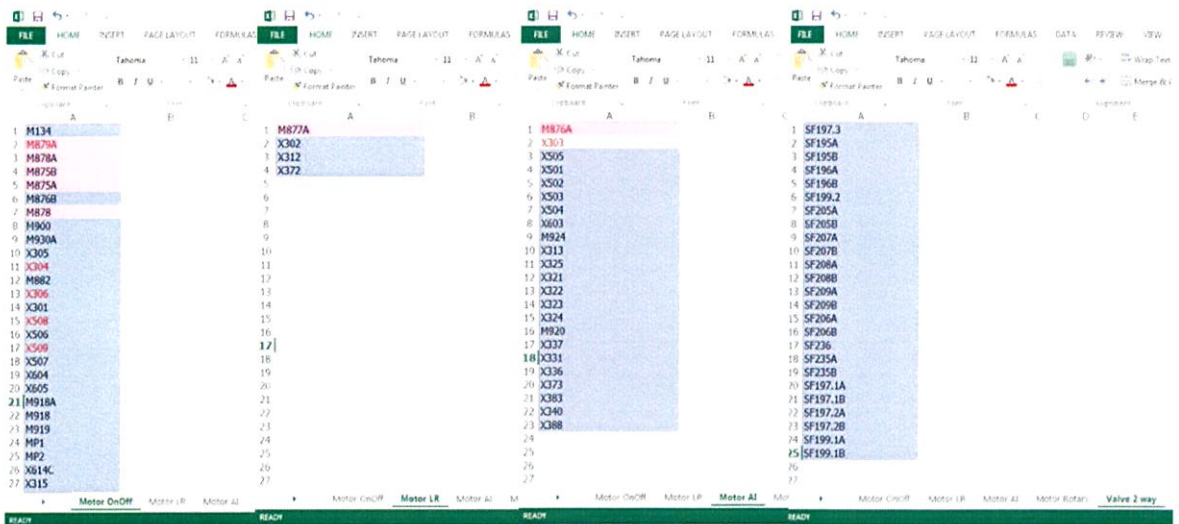
IO List PLC Feedmill - Silo Rev.06 (verawat 23-02-16).xlsx - Excel

LINE	LINE I/O	LINE DESCRIPTION	PLC Configuration										Set Analog							
			IN	COIL	OUT	Limit SW	Level	A.R.T	A. Affinity	Alarm	Stop	Start	Input	Output	Unit	Min	Max	Low	High	Alarm
1	PLC 2 sw emergency																			
2	AL1	Alarm EXT																		
3	AL2	Alarm EXT																		
4	AL3	Alarm EXT																		
5	SL100	OP - Side Bin 1 to HMT																		
6	SL101	OP - Side Bin 2 to HMT																		
7	SL102	OP - Side Bin 3 to HMT																		
8	SL103	OP - Side Bin 4 to HMT																		
9	LL110	Low stopper																		
10	Ev 104	Poppe Hammer																		
11	ML105	Al - Flow SW																		
12	ML106	Feeder HMT																		
13	AD104	Inv. Feeder HMT																		
14	ML107	Hammer M1 7 Right																		
15	ML108	Hammer M10 Left																		
16	ML109	Hammer M11																		
17	ML110	Hammer M12																		
18	ML111	Hammer M13																		
19	ML112	Hammer M14																		
20	ML113	Hammer M15																		
21	ML114	Hammer M16																		
22	ML115	Hammer M17																		
23	ML116	Hammer M18																		
24	ML117	Hammer M19																		
25	ML118	Hammer M20																		
26	ML119	Hammer M21																		
27	ML120	Hammer M22																		
28	ML121	Hammer M23																		
29	ML122	Hammer M24																		
30	ML123	Hammer M25																		

ภาพที่ 3.3 คำอธิบายชื่ออุปกรณ์

3.1.2 จัดทำรายชื่อบุคลากร

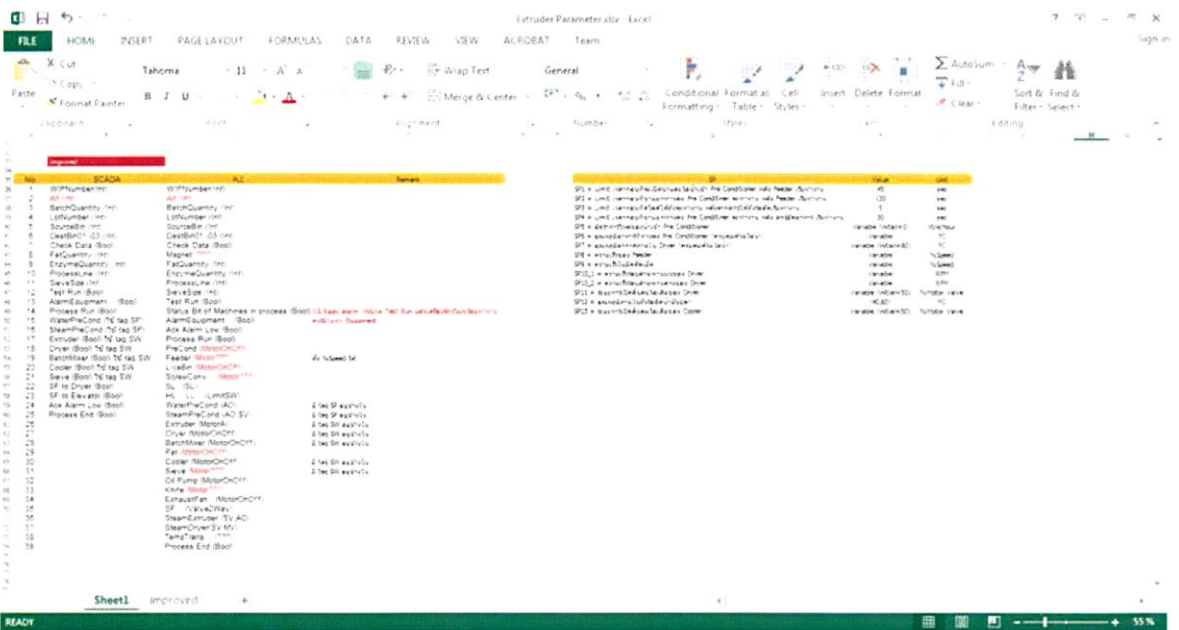
จำแนกประเภทอุปกรณ์เพื่อง่ายต่อการสร้างวัตถุ (Object) และเขียนโปรแกรมควบคุม ดังภาพที่ 3.4 เนื่องจากอุปกรณ์ชนิดเดียวกันจะมีคุณสมบัติส่วนใหญ่เหมือนกัน การจำแนกประเภทอุปกรณ์จึงทำให้สามารถสร้างวัตถุและโปรแกรมที่เป็นต้นแบบสำหรับอุปกรณ์แต่ละประเภทได้ ซึ่งทำให้เกิดความสะดวกและรวดเร็วสำหรับระบบขนาดใหญ่



ภาพที่ 3.4 รายชื่ออุปกรณ์ที่จำแนกตามประเภท

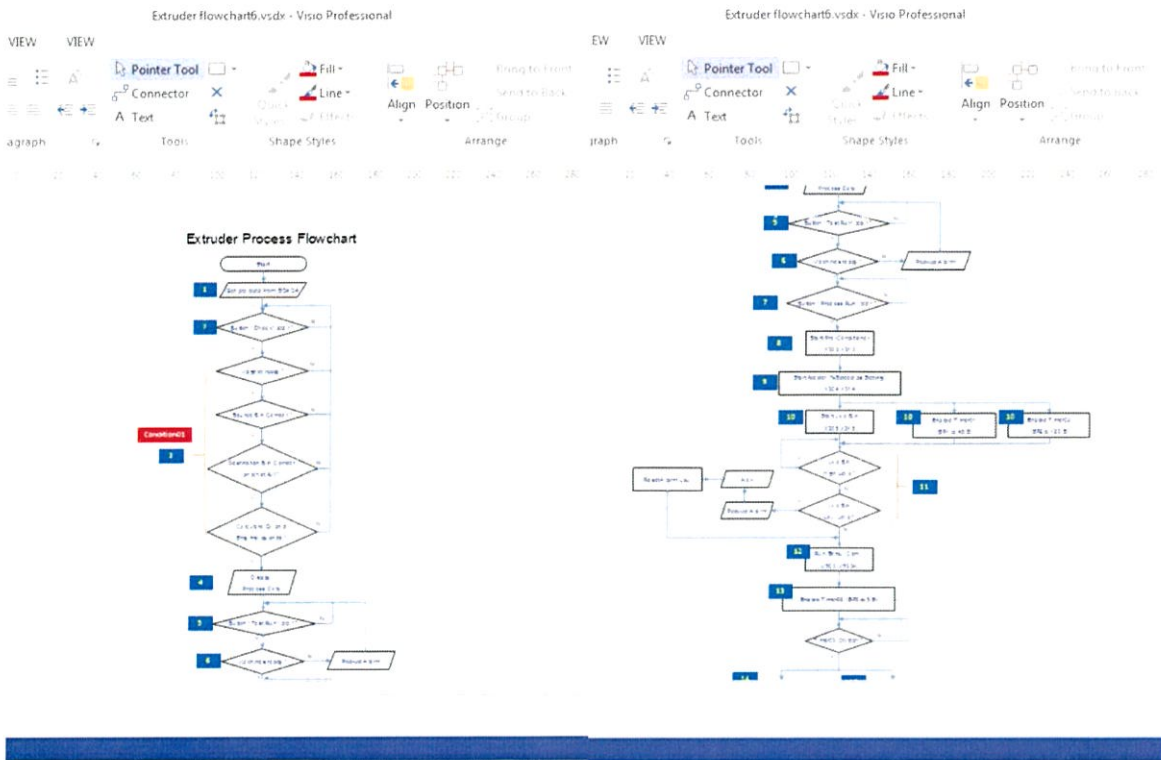
3.1.3 ออกแบบโปรแกรมควบคุม

1) วางแผนตัวแปรที่จะใช้ในการเขียนโปรแกรม แล้วจัดทำเป็นรายการ ประกอบด้วย ชื่อตัวแปร และประเภทตัวแปร เช่น Boolean, Integer, String เป็นต้น



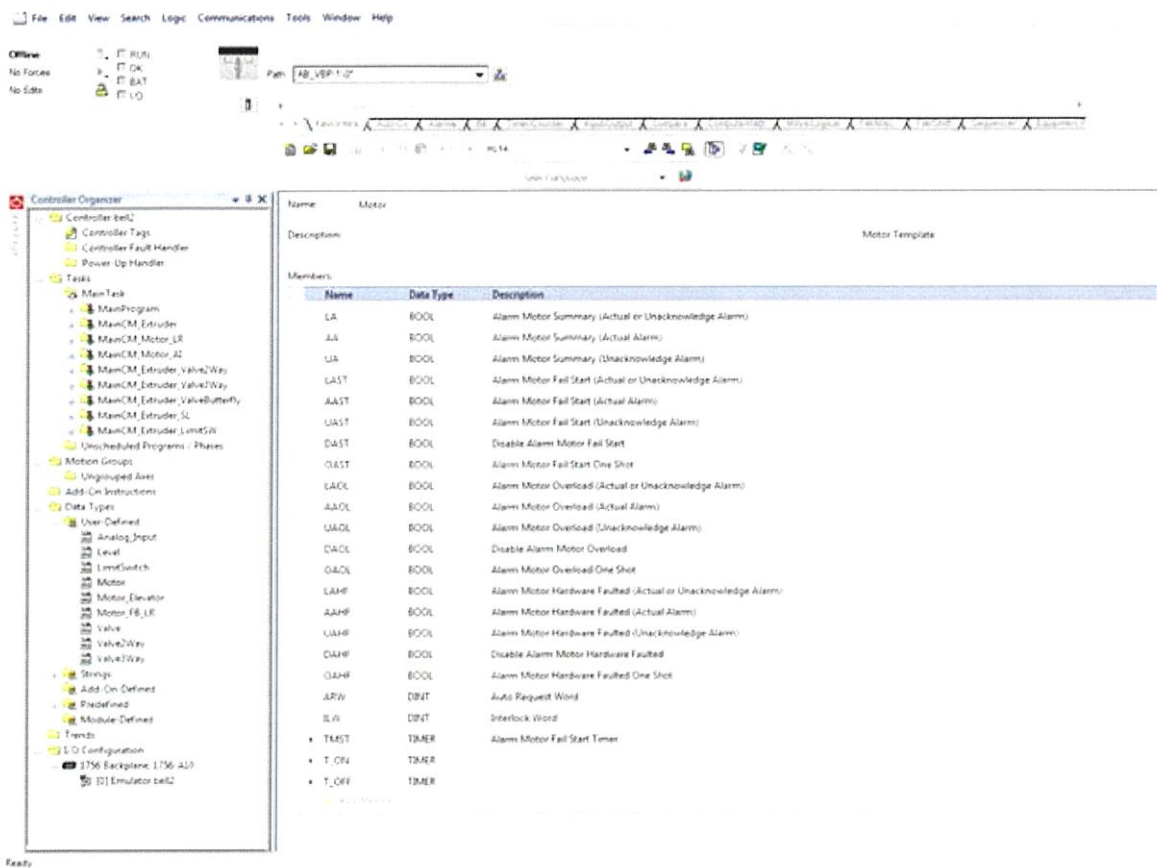
ภาพที่ 3.5 รายการตัวแปรที่จะใช้ในโปรแกรมควบคุม

2) สร้างผังงาน (Flowchart) เพื่อเป็นแนวทางในการเขียนโปรแกรมและง่ายต่อการติดตามหรือตรวจแก้ไขโปรแกรมในภายหลัง

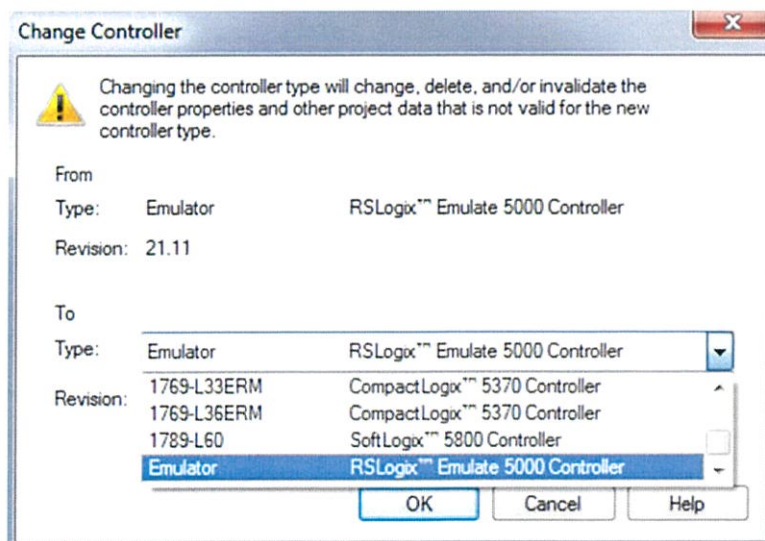


ภาพที่ 3.6 Flowchart สำหรับเขียนโปรแกรม

3) สร้างตัวแปรให้กับอุปกรณ์แต่ละประเภท ในซอฟต์แวร์ Studio 5000 เนื่องจากอุปกรณ์หนึ่งประเภทจำเป็นต้องใช้ตัวแปรหลายตัวในการควบคุม เช่น มอเตอร์ต้องใช้ตัวแปร Timer, Auto Request, Manual Request, Interlock เป็นต้น โดยตั้งค่าชนิดของตัวควบคุมในโปรแกรมให้เป็น RSLogix Emulate 5000 ดังภาพที่ 3.8 เพื่อสะดวกในการทดสอบโปรแกรมโดยใช้ซอฟต์แวร์เป็นตัวจำลองพีแอลซี

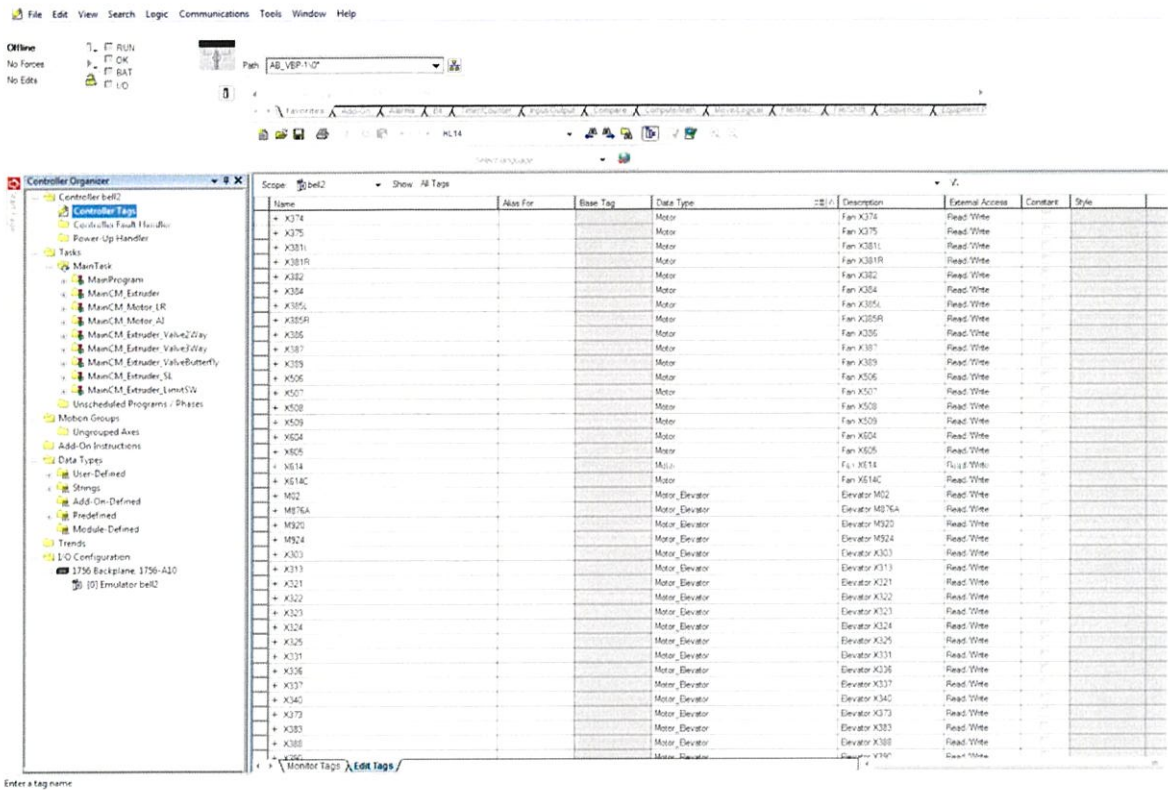


ภาพที่ 3.7 การสร้างตัวแปรให้กับประเภทของอุปกรณ์



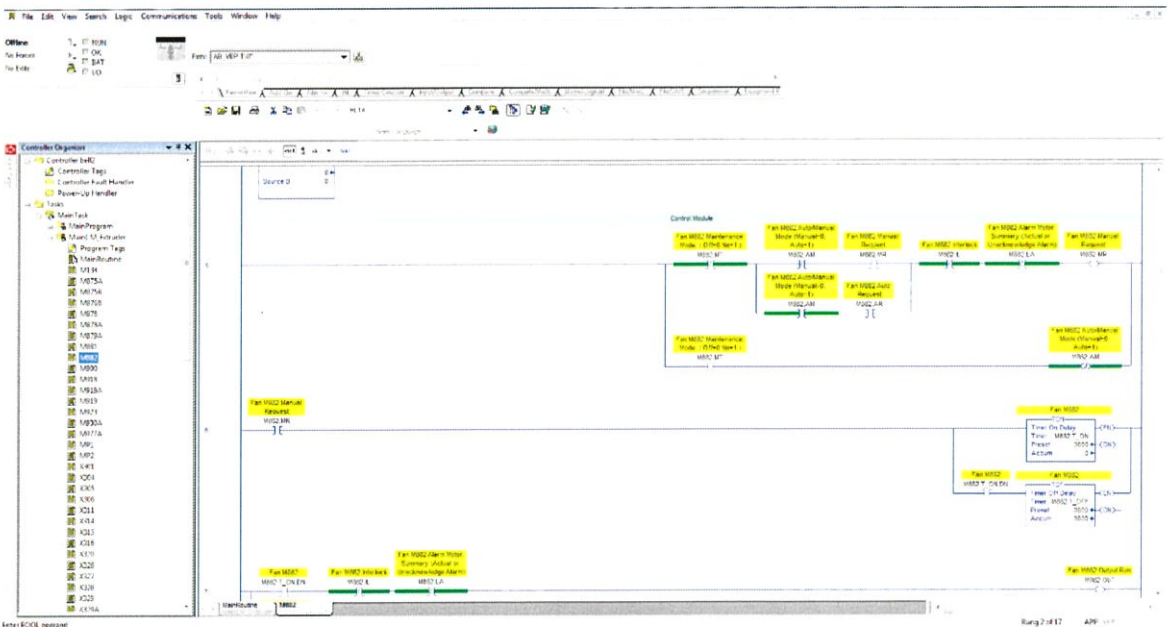
ภาพที่ 3.8 การตั้งค่าชนิดของพีแอลซี

4) สร้างตัวแปรให้กับชื่อของอุปกรณ์ (Tag Name) ทุกตัว โดยเลือกชนิดของตัวแปรให้เหมาะสมกับประเภทของอุปกรณ์ ดังภาพที่ 3.9 เช่น ชื่ออุปกรณ์ M124 เลือกชนิดตัวแปรเป็น Motor_OnOff ที่ได้สร้างไว้ในข้อ 3) เป็นต้น เพื่อให้สามารถนำชื่อของอุปกรณ์มาใช้เขียนโปรแกรมได้



ภาพที่ 3.9 การสร้างตัวแปรให้กับชื่อของอุปกรณ์

5) เขียนโปรแกรมตาม Flowchart ที่ได้วางแผนไว้ โดยใช้ภาษา Ladder Diagram

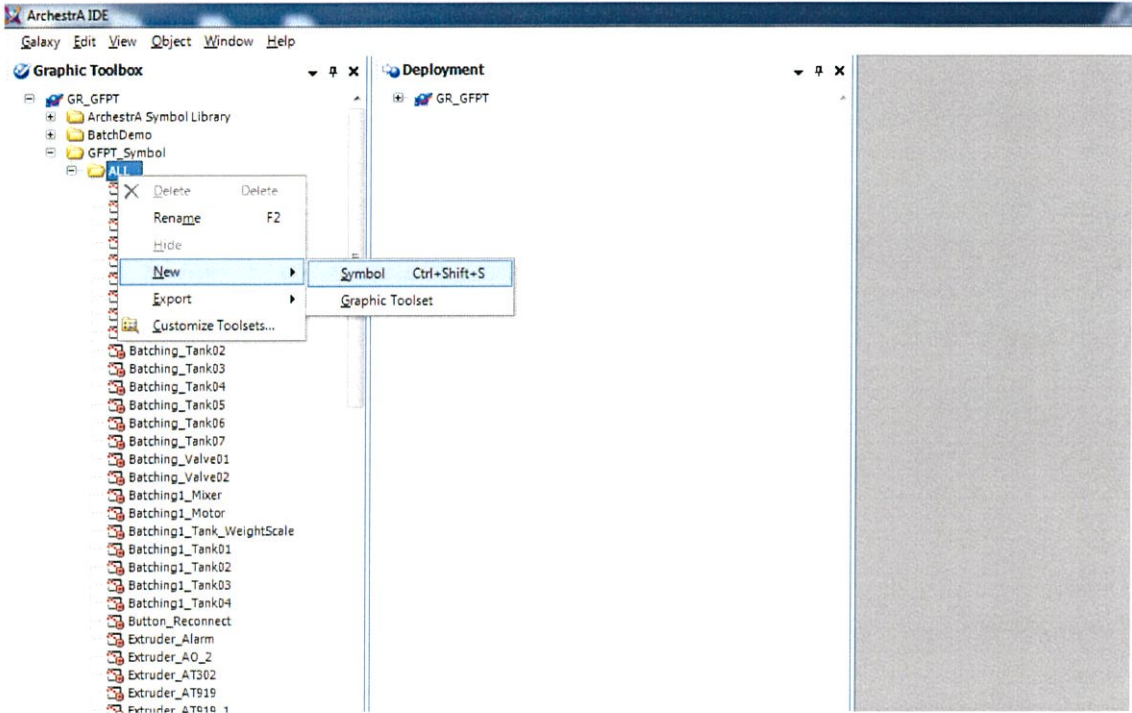


ภาพที่ 3.10 โปรแกรมควบคุมที่เขียนด้วยภาษา Ladder Diagram

3.2 สร้างวัตถุ (Object) ที่ใช้ในจอแสดงผล

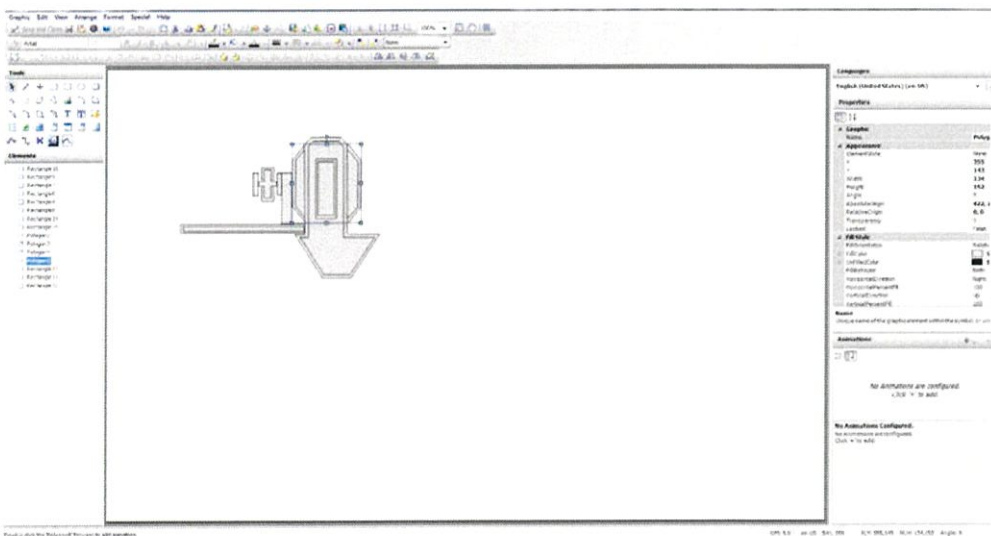
3.2.1 สร้างกราฟิกให้กับวัตถุ

1) สร้างกราฟิกให้กับวัตถุในซอฟต์แวร์ ArchestrA IDE โดยตั้งชื่อให้สอดคล้องกับชื่อของอุปกรณ์ ซึ่งกราฟิกนี้จะเก็บไว้ในส่วนของ Graphic Toolbox



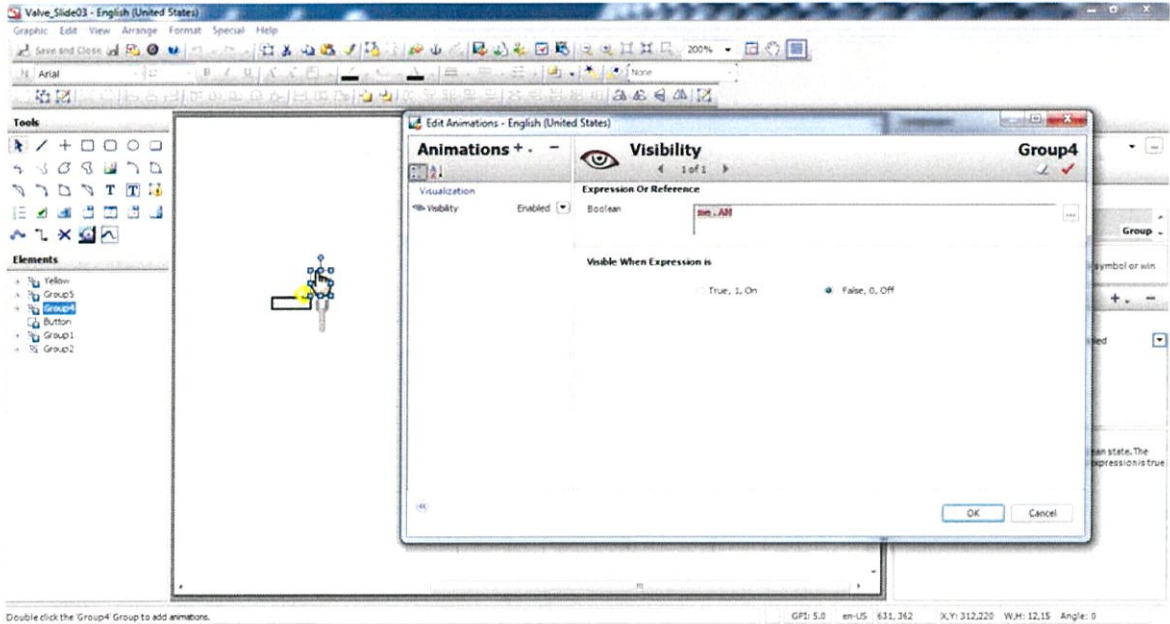
ภาพที่ 3.11 การสร้างกราฟิกให้กับวัตถุ

2) วาดกราฟิกโดยอ้างอิงลักษณะอุปกรณ์จากแผนภาพ P&ID



ภาพที่ 3.12 การวาดกราฟิก

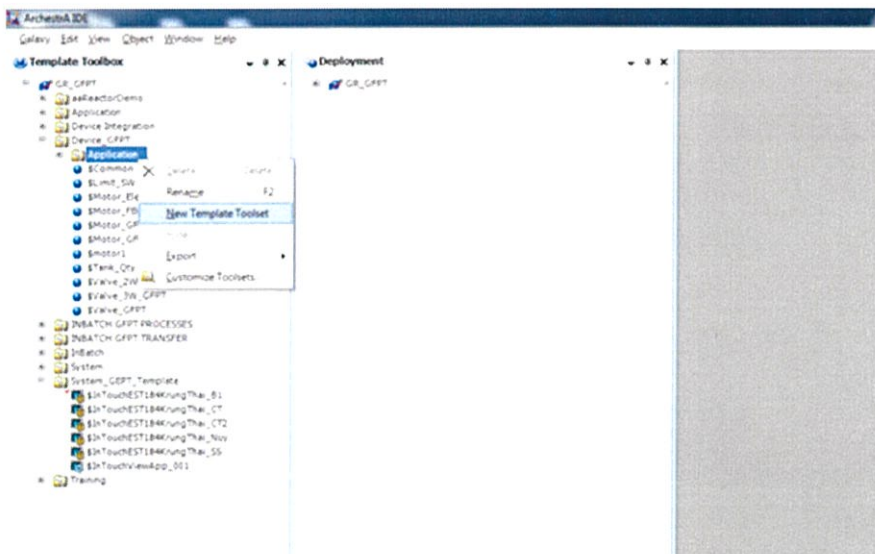
3) สร้าง Animation Link ให้กับรูปกราฟิกที่วาด โดย Animation Link นี้จะเป็นตัวกำหนดว่ารูปกราฟิกจะแสดงผลอย่างไร เมื่อมีการสั่งงานในแบบต่างๆดังตัวอย่างในภาพที่ 3.13 ซึ่งเป็นการเพิ่ม Animation Link ให้กับสัญลักษณ์รูปมือของ Slide Valve เพื่อให้รูปมือนี้ปรากฏบนจอแสดงผลเมื่อตัวแปร Manual Request จากโปรแกรมควบคุมมีสถานะเป็น 1 เท่านั้น



ภาพที่ 3.13 การเพิ่ม Animation Link ให้กับรูปกราฟิก

3.2.2 สร้างวัตถุแม่แบบ

1) สร้างวัตถุแม่แบบโดยตั้งชื่อให้สอดคล้องกับประเภทของอุปกรณ์ ซึ่งวัตถุแม่แบบนี้จะเก็บไว้ในส่วนของ Template Toolbox ดังภาพที่ 3.14

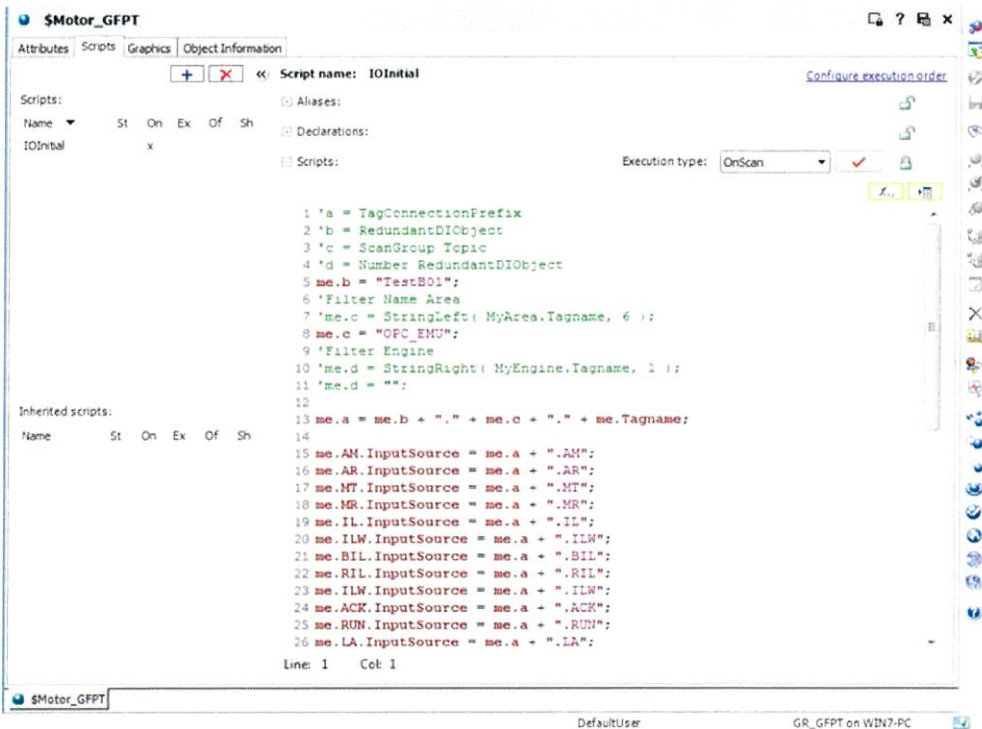


ภาพที่ 3.14 การสร้างวัตถุแม่แบบ

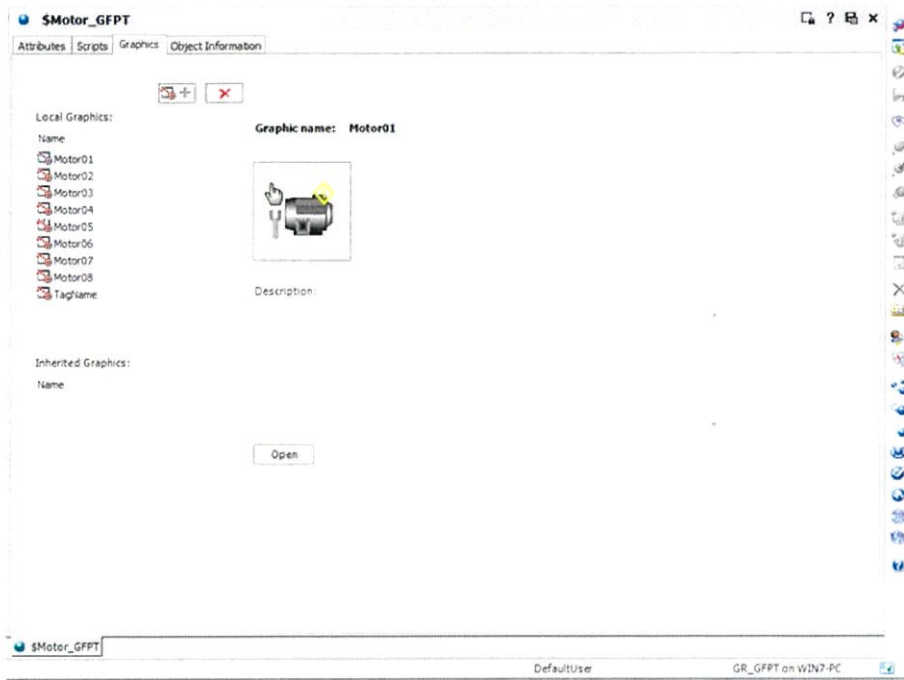
2) เพิ่ม Attribute คือตัวแปรที่เกี่ยวข้องในการแสดงผลของวัตถุ และสร้าง Script เพื่อควบคุมคุณลักษณะการแสดงผล จากการเรียกใช้และสั่งงานวัตถุโดยโปรแกรมของพีแอลซี จากนั้นเพิ่มรูปภาพที่สร้างไว้ในข้อ 3.2.1 ให้สอดคล้องกับอุปกรณ์แต่ละประเภท



ภาพที่ 3.15 การสร้าง Attribute ให้กับวัตถุแม่แบบ



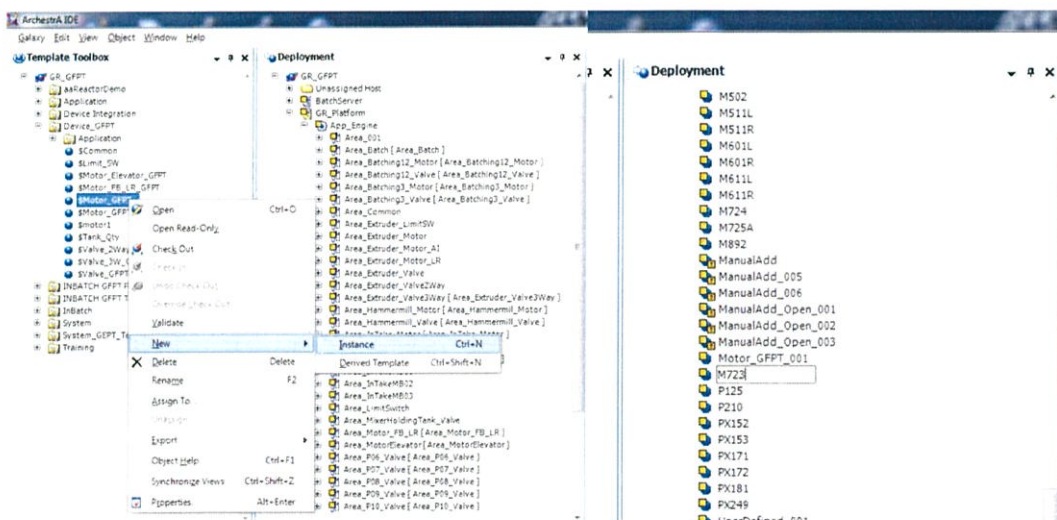
ภาพที่ 3.16 การเขียน Script ให้กับวัตถุแม่แบบ



ภาพที่ 3.17 การเพิ่มกราฟิกให้กับวัตถุแม่แบบ

3.2.3 สร้าง Instance

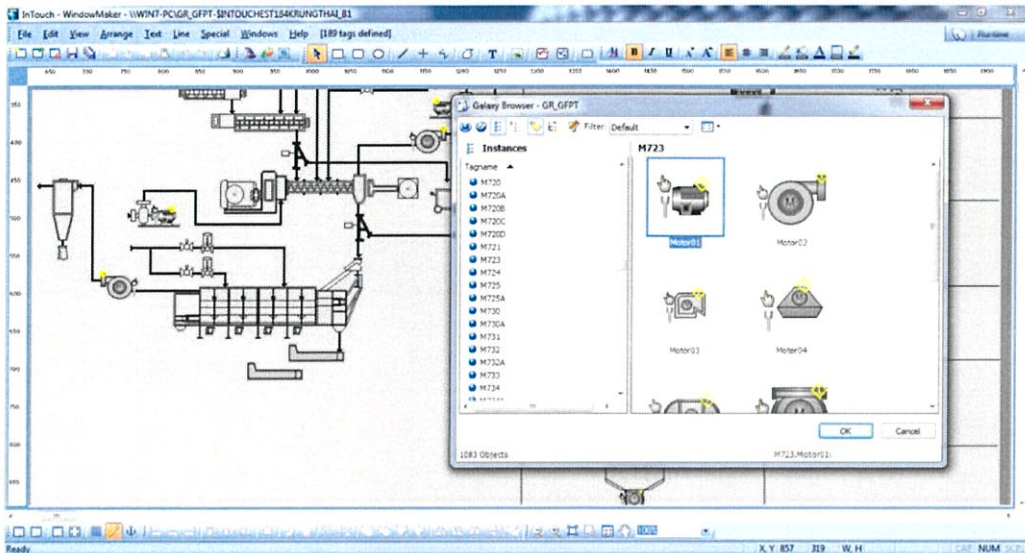
สร้าง Instance ซึ่งเปรียบเสมือนวัตถุตัวลูกที่ได้รับการถ่ายทอดคุณลักษณะทั้งหมดมาจากวัตถุแม่แบบทั้ง Attribute, Script และ Graphic โดยคุณลักษณะที่ได้รับการถ่ายทอดมานี้ จะไม่สามารถเปลี่ยนแปลงแก้ไขที่ตัวลูกได้ แต่การเปลี่ยนแปลงคุณลักษณะที่ตัวแม่แบบจะทำให้ตัวลูกที่ได้รับการถ่ายทอดคุณลักษณะดังกล่าวมานั้นเปลี่ยนแปลงตามไปด้วย ทั้งนี้อาจสร้างคุณลักษณะย่อยเพิ่มเติม ซึ่งแตกต่างกันไปที่ตัวลูกแต่ละตัวได้ การจัดการวัตถุในลักษณะดังกล่าวนี้ ทำให้เกิดความสะดวกรวดเร็วในการสร้างและแก้ไขปรับปรุงคุณลักษณะของวัตถุสำหรับระบบขนาดใหญ่ซึ่งมีอุปกรณ์เป็นจำนวนมาก โดยต้องสร้างวัตถุตัวลูกจากวัตถุแม่แบบของอุปกรณ์ประเภทเดียวกัน



ภาพที่ 3.18 การสร้าง Instance

3.3 ออกแบบหน้าจอแสดงผล

จัดวางวัตถุที่สร้างไว้ขึ้นเป็นระบบที่ถูกต้องด้วยซอฟต์แวร์ InTouch WindowMaker โดยอ้างอิงรูปแบบของระบบจากแผนภาพ P&ID และออกแบบหน้าจอแสดงผลให้ง่ายต่อการสังเกตของผู้ปฏิบัติงาน

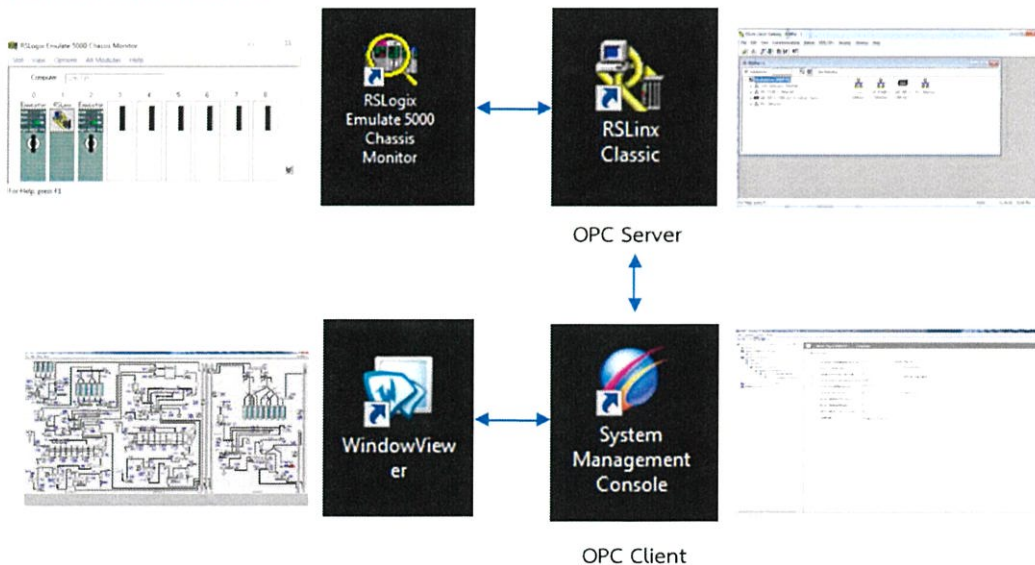


ภาพที่ 3.19 การสร้างหน้าจอแสดงผล

3.4 ทดสอบการทำงานของโปรแกรมควบคุมและส่วนแสดงผล

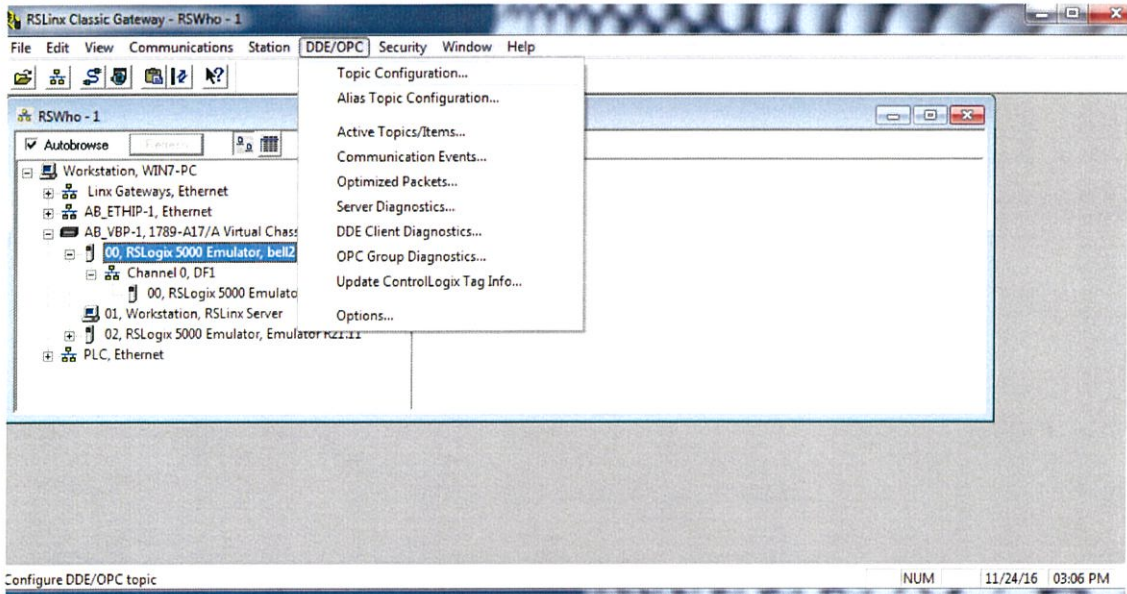
3.4.1 เชื่อมต่อโปรแกรมควบคุมและส่วนแสดงผลด้วย OPC

เนื่องจากส่วนแสดงผลและโปรแกรมควบคุมจะต้องสื่อสารแลกเปลี่ยนข้อมูลระหว่างกัน จึงต้องทำการทดสอบการทำงานของระบบว่ามีความถูกต้องสัมพันธ์กันหรือไม่ โดยใช้ OPC เป็นตัวกลางในการรับ-ส่งข้อมูลระหว่างซอฟต์แวร์ที่เป็น OPC Server และ OPC Client ซึ่งได้แก่ซอฟต์แวร์ RSLinx Classic และ System Management Console ตามลำดับ แผนผังการรับ-ส่งข้อมูลแสดงดังภาพที่ 3.20 โดยมีขั้นตอนการเชื่อมต่อดังนี้

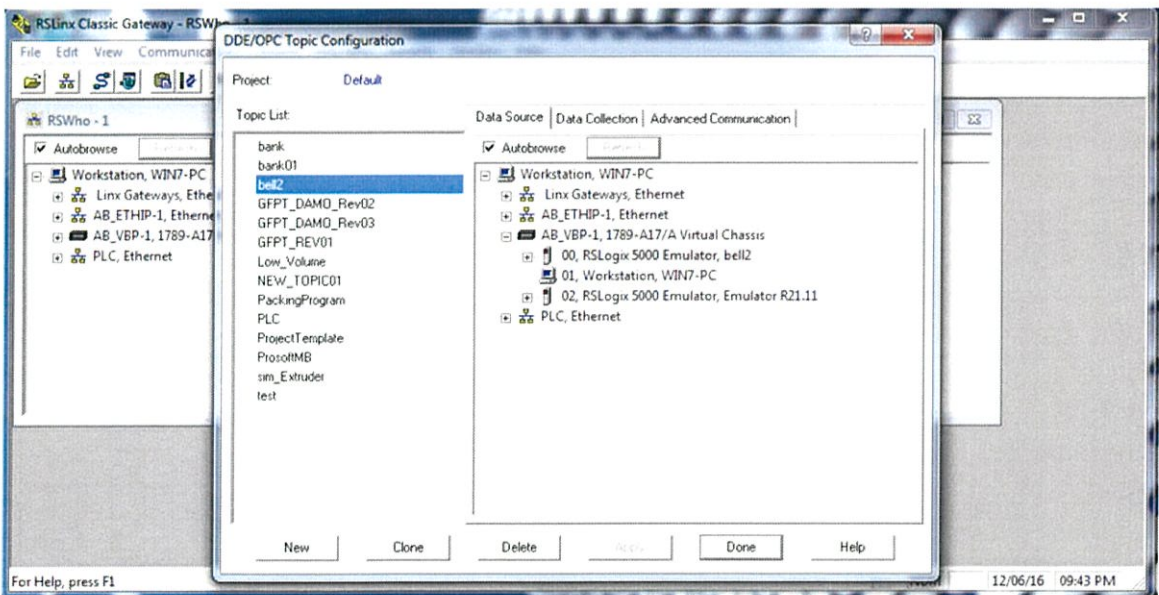


ภาพที่ 3.20 แผนผังการรับ-ส่งข้อมูลระหว่างซอฟต์แวร์

1) ตั้งค่าการติดต่อสื่อสารในซอฟต์แวร์ RSLinx Classic ให้เป็น OPC Protocol ดังภาพที่ 3.21 ซึ่งจะติดต่อสื่อสารกับซอฟต์แวร์ที่มี Topic Name ชื่อตรงกันเท่านั้น จากนั้นตั้งชื่อ Topic Name แล้วจับคู่กับอุปกรณ์ที่จะดึงข้อมูลขึ้นมาเก็บเพื่อใช้เป็น OPC Server ดังภาพที่ 3.22 ซึ่งอุปกรณ์ในที่นี้คือพีแอลซีจำลองจากซอฟต์แวร์ RSLogix Emulate 5000 Chassis Monitor

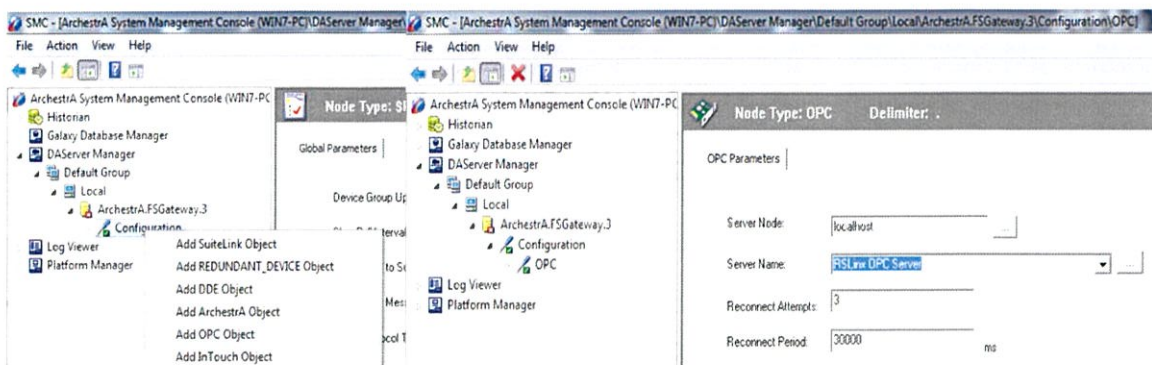


ภาพที่ 3.21 การตั้งค่าการติดต่อสื่อสารด้วย Topic Name ของ OPC Protocol



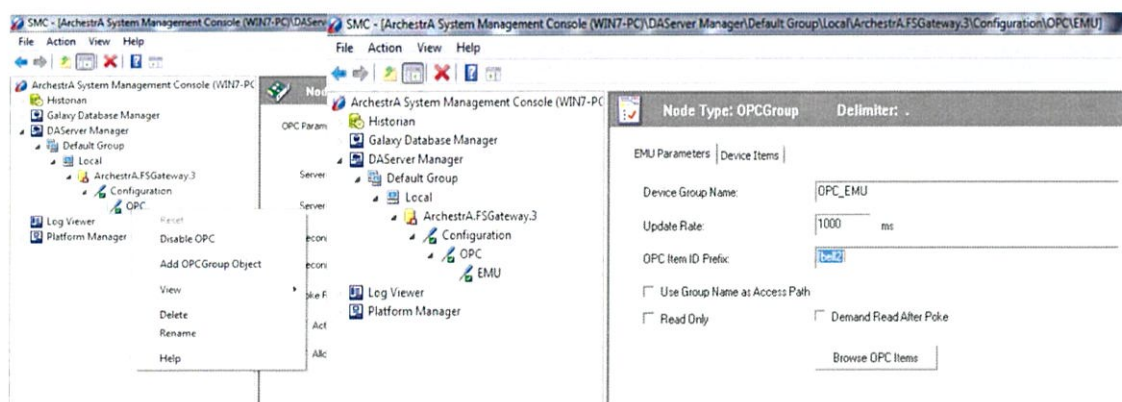
ภาพที่ 3.22 การจับคู่ Topic Name กับอุปกรณ์ที่ต้องการ

2) สร้าง OPC Object ขึ้นในซอฟต์แวร์ System Management Console ซึ่งใช้เป็น OPC Client โดยเลือก Server Name เป็น RSLinx OPC Server ดังภาพที่ 3.23 เพื่อให้สามารถติดต่อกับซอฟต์แวร์ RSLinx Classic ด้วย OPC Protocol ได้



ภาพที่ 3.23 การสร้าง OPC Object

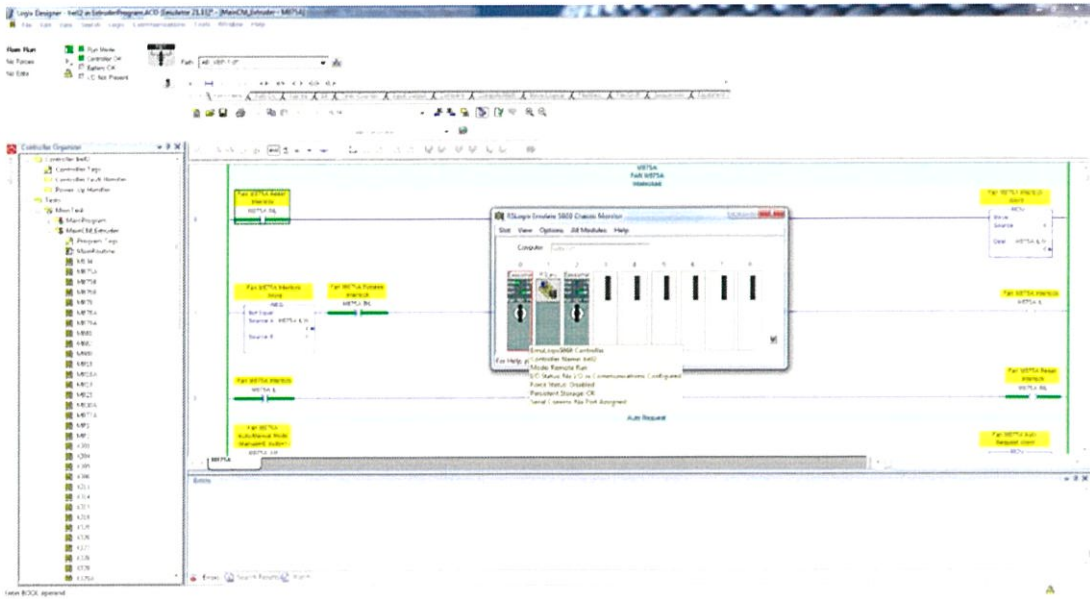
3) สร้าง OPC Group Object ขึ้นจาก OPC Object ที่ได้สร้างไว้แล้วในข้อ 2) เนื่องจาก OPC Object แต่ละ Group นั้นจะสามารถติดต่อกับซอฟต์แวร์ที่มี Topic Name ชื่อเดียวกัน จึงต้องตั้งชื่อ Topic Name ให้กับแต่ละ Group โดยไม่ซ้ำกันและชื่อนั้นต้องเหมือนกับ Topic Name ของ OPC Server ที่ต้องการรับ-ส่งข้อมูลด้วย



ภาพที่ 3.24 การสร้าง OPC Group Object

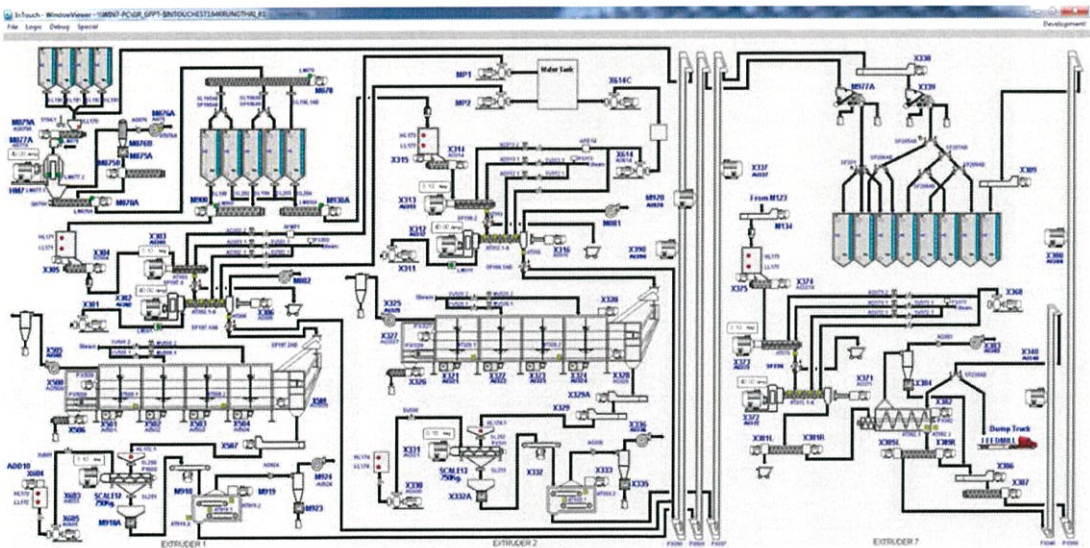
3.4.2 ทดสอบการทำงานของระบบ

1) ดาวน์โหลดโปรแกรมควบคุมจากซอฟต์แวร์ Studio 5000 ลงในพีแอลซีจำลองคือซอฟต์แวร์ RSLogix Emulate 5000 Chassis Monitor แล้วสั่ง Online PLC จะปรากฏแถบสีเขียวขึ้นในโปรแกรมของ Studio 5000 ดังภาพที่ 3.25



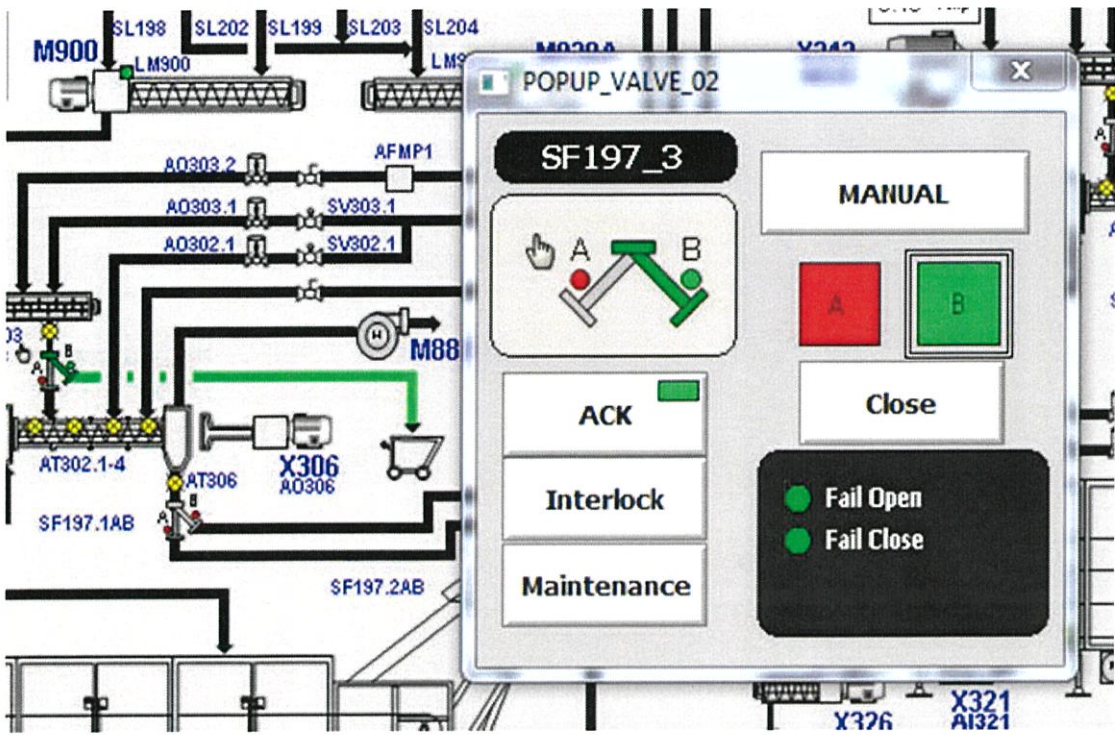
ภาพที่ 3.25 การ Run โปรแกรมควบคุมในพีแอลซีจำลอง

2) เปิดหน้าจอแสดงผลที่สร้างขึ้น โดยใช้ซอฟต์แวร์ InTouch WindowViewer



ภาพที่ 3.26 หน้าจอแสดงผลในซอฟต์แวร์ InTouch WindowViewer

3) ทดสอบการทำงานของระบบโดยการสั่งงานอุปกรณ์แต่ละตัวในหน้าจอแสดงผล จากนั้นดูผลลัพธ์ว่ามีการแสดงคุณลักษณะสัมพันธ์กับโปรแกรมควบคุมตามที่ต้องการหรือไม่ แล้วทำการแก้ไขปรับปรุงจนกว่าจะได้ระบบที่ถูกต้อง ตัวอย่างการทดสอบระบบแสดงดังภาพที่ 3.27



ภาพที่ 3.27 ตัวอย่างการทดสอบระบบ

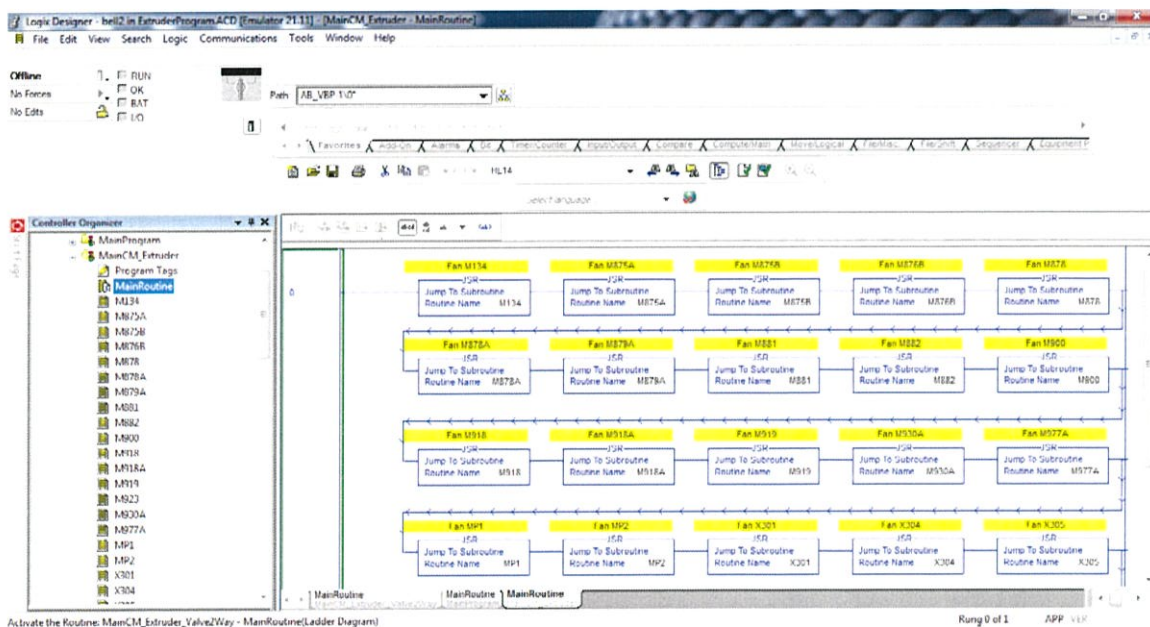
บทที่ 4 ผลการดำเนินงาน

4.1 โปรแกรมควบคุม

โปรแกรมควบคุมจะแบ่งเป็น 2 ส่วนหลักๆ ได้แก่ Main Program และ Subroutine

4.1.1 Main Program

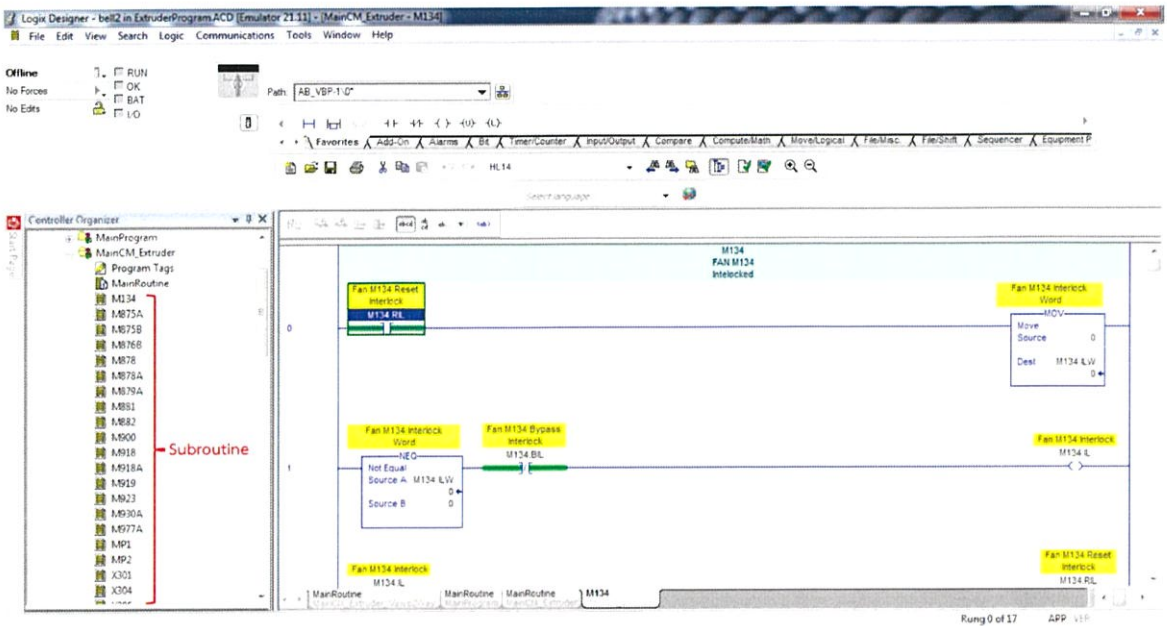
เป็นโปรแกรมควบคุมขั้นตอนการทำงานของกระบวนการโดยรวม เช่น หากมีอุปกรณ์ตัวหนึ่งทำงานอยู่ จะสามารถสั่งอุปกรณ์ตัวใดให้ทำงานได้อีกบ้าง ภายในกี่วินาที เป็นต้น โดยจะมีตัวแปรในการส่งค่าไปที่ Subroutine ที่เกี่ยวข้อง เพื่อไปควบคุมอุปกรณ์ในสายการผลิตต่อไป



ภาพที่ 4.1 Main Program ของโปรแกรมควบคุม

4.1.2 Subroutine

เป็นโปรแกรมควบคุมการทำงานของอุปกรณ์ประเภทต่างๆ เช่น มอเตอร์แบบ On - Off, มอเตอร์แบบกลับทิศทางการหมุนได้, วาล์ว 2 ทาง และ วาล์วแบบสไลด์เปิด - ปิด เป็นต้น ซึ่งหลักการการทำงานของอุปกรณ์แต่ละประเภทมีความแตกต่างกัน จึงต้องเขียนโปรแกรมที่ต่างกันเพื่อควบคุมการทำงาน เช่น ควบคุมการเปิด, การหมุน และการแจ้งเตือนกรณีเกิดความผิดปกติ โดยตัวอย่าง Subroutine สำหรับควบคุมการทำงานของมอเตอร์ชนิด On - Off แสดงดังภาพที่ 4.2



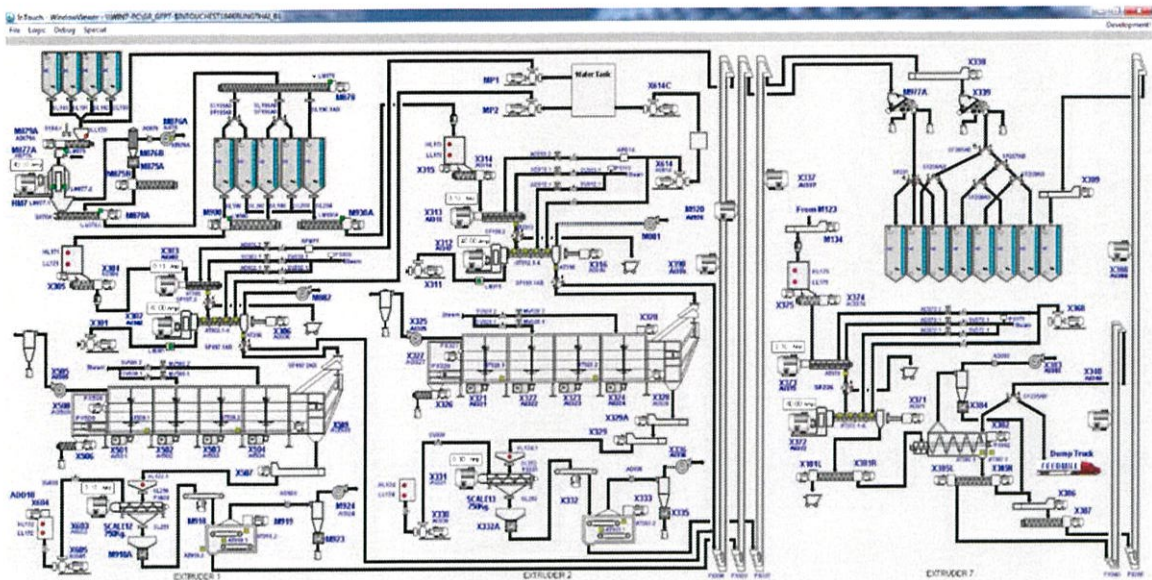
ภาพที่ 4.2 Subroutine ควบคุมการทำงานของ On – Off Motor

การแบ่งโปรแกรมออกเป็น 2 ส่วนข้างต้น จะช่วยประหยัดเวลาและหน่วยความจำในการประมวลผลของพีแอลซี เนื่องจากพีแอลซีจะประมวลผลเฉพาะที่ Main Program และเรียกใช้ Subroutine ของอุปกรณ์ที่เกี่ยวข้องเท่านั้น นอกจากนี้ยังช่วยให้สะดวกต่อการแก้ไขโปรแกรมสำหรับอุปกรณ์แต่ละชนิด ซึ่งจะทำให้ได้ง่ายกว่าการเก็บโปรแกรมทุกอย่างไว้ใน Main Program ที่เดียว

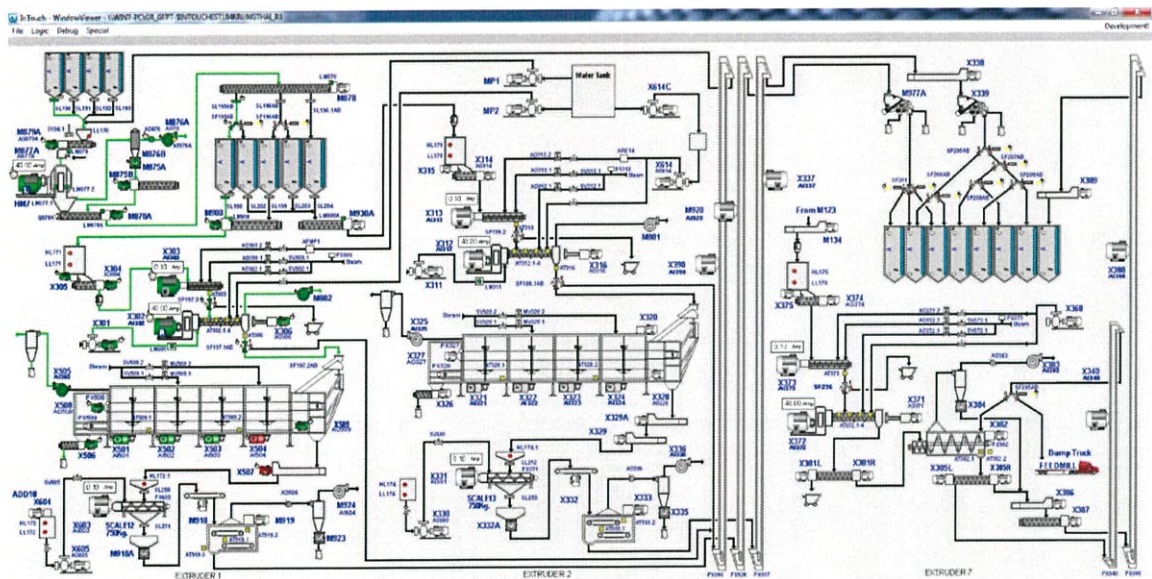
4.2 ส่วนแสดงผล

4.2.1 ภาพรวมของระบบ

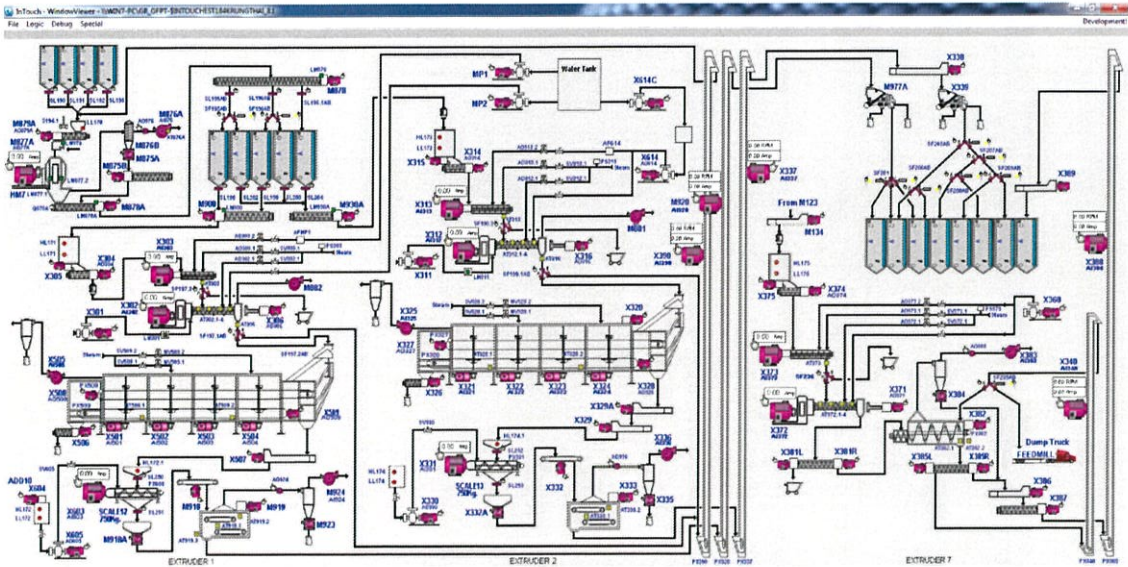
แสดงอุปกรณ์ทุกตัวในกระบวนการที่เชื่อมต่อกันด้วยท่อและสายพานลำเลียง ผู้ปฏิบัติงานจึงสามารถสังเกตการทำงานของกระบวนการโดยรวมได้ในหน้าจอเดียว โดยใช้สีเขียวแสดงสถานะอุปกรณ์ที่ทำงานอยู่ สีดำแสดงสถานะอุปกรณ์ที่ไม่ได้ทำงาน สีแดงแสดงสถานะอุปกรณ์ที่ผิดปกติ (แสดง Alarm) และสีม่วงแสดงสถานะอุปกรณ์ที่เกิดความผิดพลาดด้านการติดต่อสื่อสาร (Communication Fault) ระหว่างซอฟต์แวร์แสดงผล (InTouch WindowViewer) และซอฟต์แวร์สำหรับเก็บกราฟิกของอุปกรณ์ (Archestra IDE)



ภาพที่ 4.3 ภาพรวมของกระบวนการขณะที่ยังไม่มีอุปกรณ์ทำงาน



ภาพที่ 4.4 ภาพรวมของกระบวนการขณะที่อุปกรณ์บางส่วนทำงานและบางส่วนเกิดความผิดปกติ

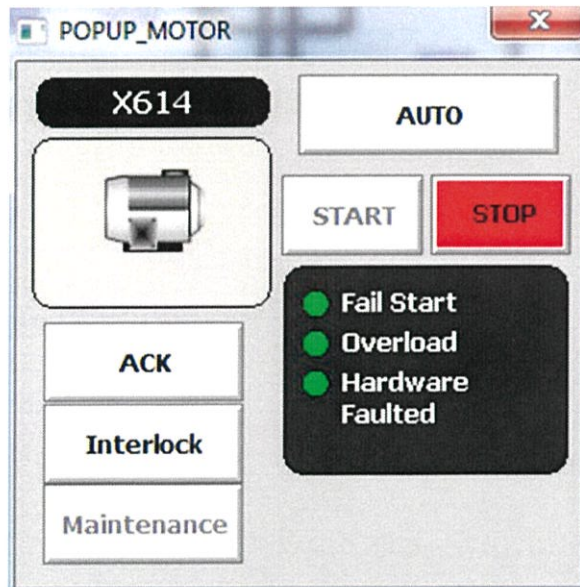


ภาพที่ 4.5 ภาพรวมของกระบวนการเมื่อเกิด Communication Fault ระหว่างซอฟต์แวร์ที่ใช้แสดงผลและซอฟต์แวร์ที่เก็บกราฟิกของอุปกรณ์

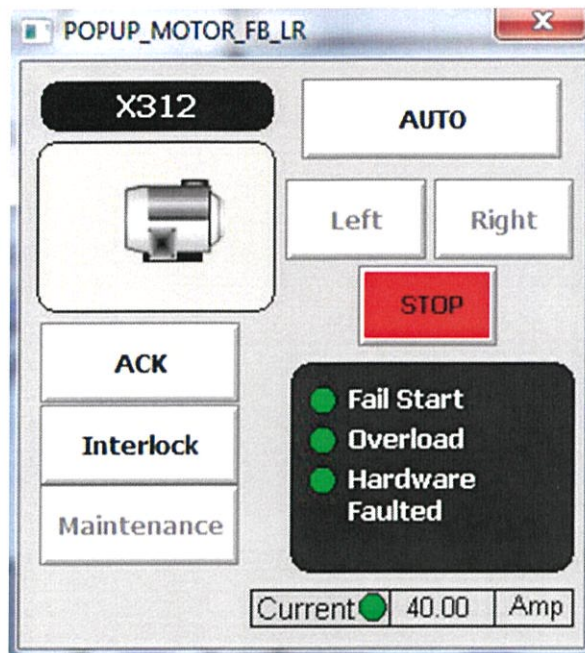
4.2.2 ฟังก์ชันการแสดงผลของอุปกรณ์

4.2.2.1 Pop Up ของอุปกรณ์

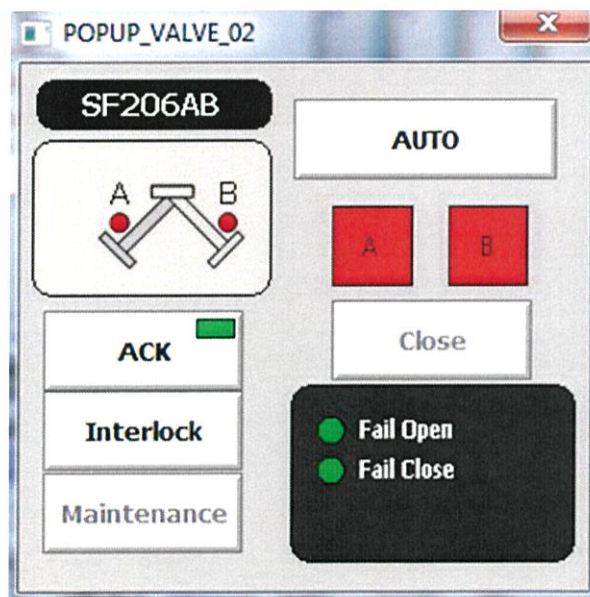
เมื่อกดเลือกอุปกรณ์จากหน้าจอ HMI จะปรากฏ Pop Up ซึ่งมีองค์ประกอบเช่น ชื่ออุปกรณ์ (Tag Name), สัญลักษณ์, โหมดการทำงาน, แถบแสดง Alarm และแถบคำสั่งที่เกี่ยวข้องของอุปกรณ์ชนิดต่างๆ เป็นต้น โดย Pop Up ของอุปกรณ์แต่ละชนิดมีดังนี้



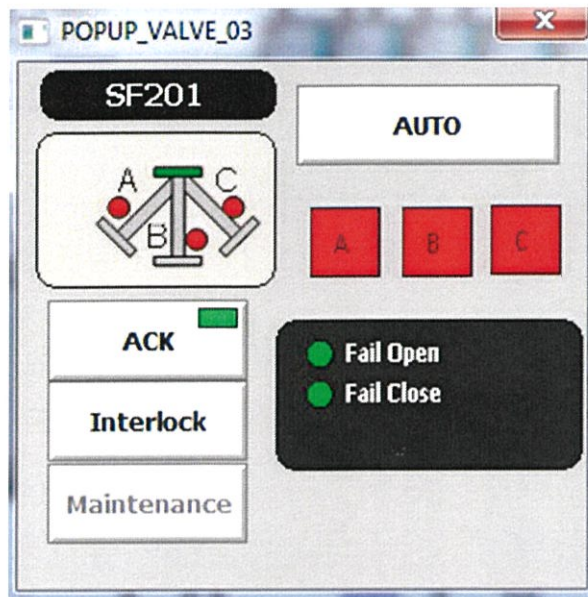
ภาพที่ 4.6 Pop Up ของ On - Off Motor (มอเตอร์สำหรับเปิด - ปิด เท่านั้น)



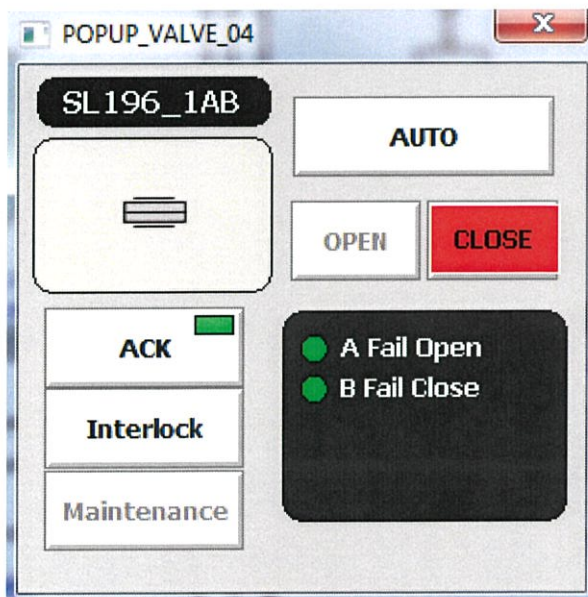
ภาพที่ 4.7 Pop Up ของ Left – Right Motor (มอเตอร์หมุนกลับทิศทางได้)



ภาพที่ 4.8 Pop Up ของ 2 – Way Valve (วาล์ว 2 ทาง)



ภาพที่ 4.9 Pop Up ของ 3 – Way Valve (วาล์ว 3 ทาง)



ภาพที่ 4.10 Pop Up ของ Sliding Valve (วาล์วสไลด์เปิด – ปิด)

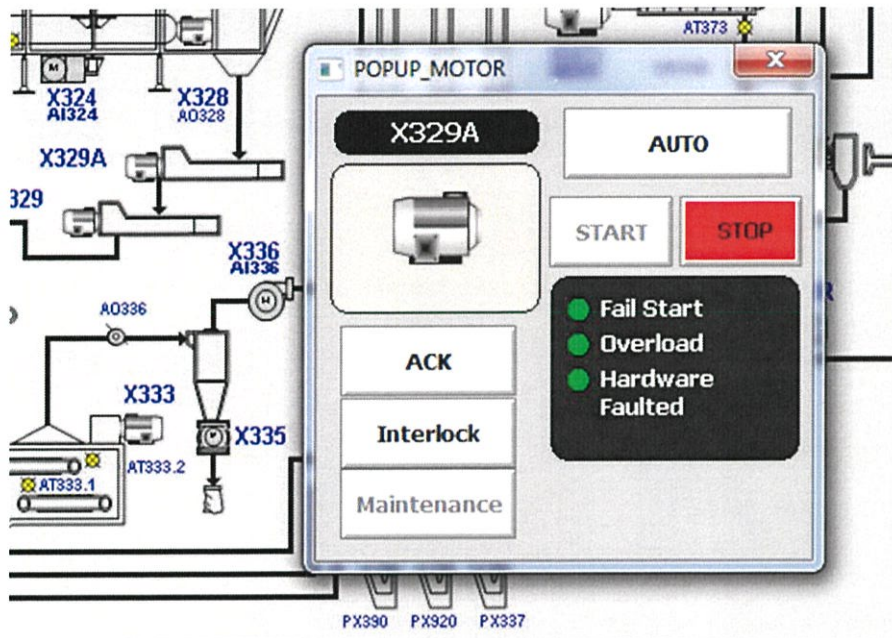
4.2.2.2 การแสดงผลเมื่ออุปกรณ์ถูกสั่งงาน

จากแถบคำสั่งของอุปกรณ์ชนิดต่างๆ สามารถสั่งงานให้อุปกรณ์ทำงานได้ 2 โหมด ได้แก่ Manual และ Auto Mode โดย Manual Mode นั้น ผู้ปฏิบัติงานสามารถสั่งงานอุปกรณ์ได้ที่ละตัว จากหน้าจอ HMI ส่วน Auto Mode ผู้ปฏิบัติงานจะต้องระบุถึงต้นทางที่จะนำวัตถุดิบมาอัดเม็ด และถึงปลายทางที่จะนำอาหารเม็ดไปเก็บ โดยการสั่งงานจากหน้าจอใน Auto Mode เมื่อกดปุ่มเริ่มการทำงาน อุปกรณ์แต่ละตัวก็จะทำงานไปตามลำดับขั้นตอนที่โปรแกรมไว้โดยอัตโนมัติ แต่ในกรณีที่ต้องมีการซ่อมแซม อุปกรณ์ในระหว่างการผลิต ผู้ปฏิบัติงานจะต้องกดปุ่มคำสั่ง Maintenance จาก Pop Up ของอุปกรณ์ที่ต้องการซ่อมแซม ซึ่งจะทำให้อุปกรณ์นั้นหยุดการทำงานและไม่สามารถสั่งงานได้จนกว่าจะยกเลิกสถานะ

Maintenance ทั้งนี้เพื่อความปลอดภัยของช่างที่ซ่อมแซมอุปกรณ์ โดยตัวอย่างของการแสดงผลเมื่ออุปกรณ์ถูกสั่งงานมีดังนี้

1) Auto Mode

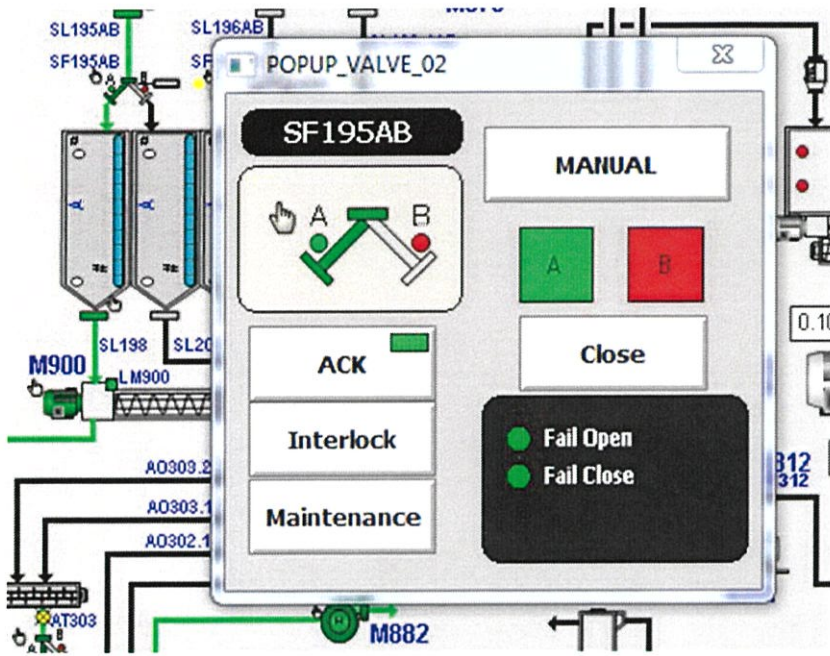
เมื่อกระบวนการทำงานอยู่ใน Auto Mode การแสดงผลของอุปกรณ์จะเป็นไปตามการสั่งงานจากโปรแกรมของพีแอลซีเท่านั้น ผู้ปฏิบัติงานจะไม่สามารถกดสั่งงานอุปกรณ์ได้จากหน้าจอ HMI



ภาพที่ 4.11 การแสดงผลเมื่ออุปกรณ์ทำงานอยู่ใน Auto Mode

2) Manual Mode

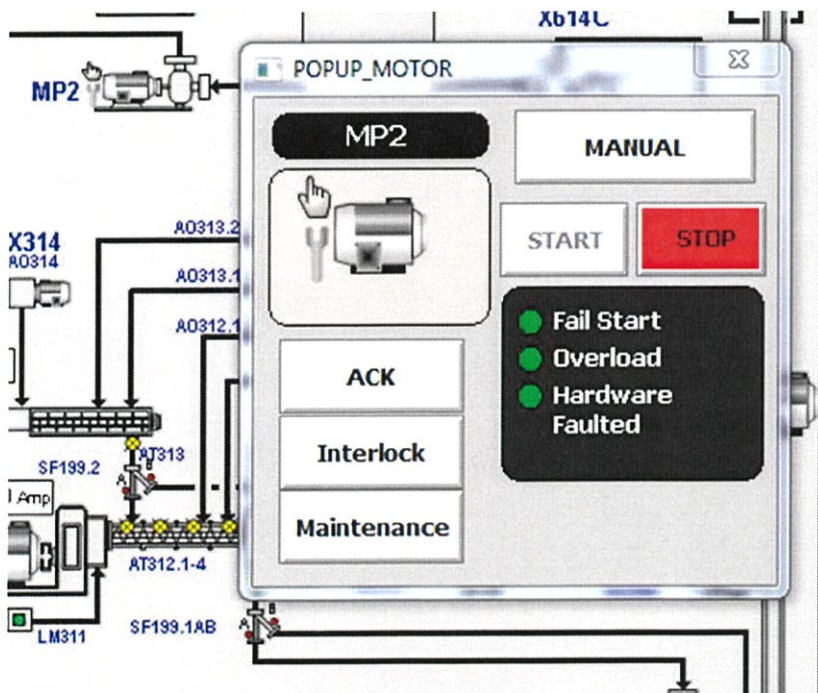
เมื่อกระบวนการทำงานอยู่ใน Manual Mode ผู้ปฏิบัติงานจะสามารถสั่งงานอุปกรณ์แต่ละตัวได้จากหน้าจอ HMI เช่น สั่งเปิด - ปิดมอเตอร์, สั่งเปิดวาล์ว 2 ทางในด้าน A หรือ B และสั่งให้มอเตอร์เปลี่ยนทิศทางการหมุน เป็นต้น



ภาพที่ 4.12 การแสดงผลเมื่ออุปกรณ์ทำงานอยู่ใน Manual Mode

3) Maintenance Status

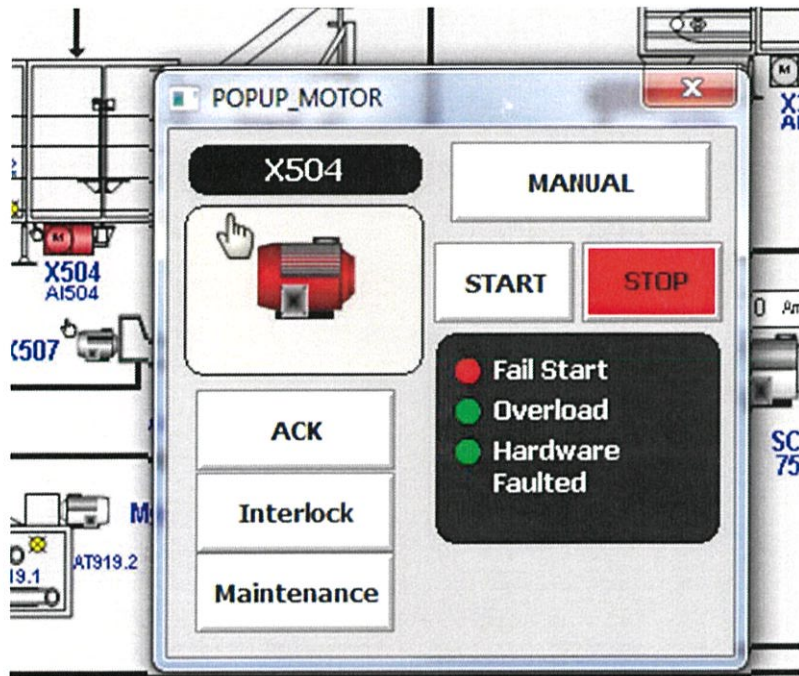
เมื่อต้องการซ่อมแซมอุปกรณ์ การกดสถานะ Maintenance ให้กับอุปกรณ์ที่ต้องการ จะช่วยเพิ่มความปลอดภัยในการทำงาน เพราะอุปกรณ์จะหยุดทำงานและไม่สามารถสั่งงานได้ จนกว่าจะยกเลิกสถานะนี้



ภาพที่ 4.13 การแสดงผลเมื่ออุปกรณ์อยู่ในสถานะ Maintenance

4.2.2.3 การแสดงผลเมื่อเกิดความผิดปกติ

เมื่อเกิดความผิดปกติขึ้นกับอุปกรณ์ สัญลักษณ์ของอุปกรณ์จะมีสีแดงกระพริบอยู่ตลอดเวลา และ Pop Up ของอุปกรณ์จะแสดงต้นเหตุของความผิดปกติว่าเกิดจาก Fail Start/Overload หรือ Hardware Faulted โดยการแสดง Alarm นี้จะยังคงอยู่จนกว่าอุปกรณ์จะกลับสู่สภาวะปกติ หรือผู้ปฏิบัติงานกด Acknowledge (ปุ่ม ACK) จากหน้าจอแสดงผล



ภาพที่ 4.14 การแสดงผลเมื่อเกิดความผิดปกติขึ้นกับอุปกรณ์

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

จากการดำเนินงานออกแบบโปรแกรมควบคุมและส่วนแสดงผลของกระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ท루ชัน ได้ผลลัพธ์เป็นโปรแกรมควบคุมของพีแอลซีซึ่งเขียนด้วยภาษา Ladder Diagram และหน้าจอแสดงผล HMI ซึ่งจะอำนวยความสะดวกในการสังเกตและสั่งงานกระบวนการโดยผู้ปฏิบัติงาน ในส่วนของโปรแกรมควบคุมที่แบ่งเป็น Main Program และ Subroutine ช่วยให้เกิดความสะดวกต่อการแก้ไขลำดับขั้นตอนการทำงานของกระบวนการ หรือฟังก์ชันการทำงานของอุปกรณ์แต่ละประเภท และใน ส่วนของการแสดงผลซึ่งมีการเก็บวัตถุกราฟิกต่างๆ อย่างเป็นระบบ ส่งผลให้การแก้ไขและจัดการคุณสมบัติ ของวัตถุรวดเร็ว ลดความซ้ำซ้อนและความผิดพลาดที่อาจเกิดขึ้นได้ ซึ่งจากผลลัพธ์ข้างต้นนี้จะได้องค์ประกอบที่สำคัญในการควบคุมกระบวนการอัดเม็ดอาหารสัตว์แบบเอ็กซ์ทรูชัน ที่พร้อมจะนำไปติดตั้ง และทดสอบร่วมกับระบบอื่นๆ ซึ่งเกี่ยวข้องกับการควบคุมกระบวนการ เช่น ระบบฐานข้อมูล (Database) จึงเป็นประโยชน์ในการพัฒนาการควบคุมกระบวนการให้มีประสิทธิภาพมากยิ่งขึ้น

5.2 ปัญหาและแนวทางแก้ไข

ปัญหาที่พบระหว่างการปฏิบัติงานสหกิจศึกษาคือ ข้อมูลการทำงานของกระบวนการจากแผนผัง แสดงความต้องการของลูกค้าไม่ชัดเจน รวมถึงมีการปรับเปลี่ยนบ่อยครั้ง เนื่องจากผู้เขียนแผนไม่ใช่ผู้ที่ คำนึงเกี่ยวกับอุปกรณ์ในสายการผลิตเท่ากับผู้ปฏิบัติงานซึ่งทำงานอยู่ที่หน้างานเป็นประจำ หรือทางบริษัท ลูกค้าอาจยังไม่เชี่ยวชาญทางด้าน Automation มากนัก เพราะอยู่ในช่วงเริ่มปรับปรุงกระบวนการใหม่ ดังนั้น แนวทางแก้ไขที่ทำได้คือปรึกษากับผู้นิเทศงานถึงปัญหาในรายละเอียดต่างๆ เพื่อให้ผู้นิเทศงานติดต่อ ประสานงานกับลูกค้าเพื่อขอข้อมูลที่ชัดเจน รวมถึงการรายงานผลงานเป็นระยะ เพื่อดูแลตอบรับของบริษัท ลูกค้าว่าต้องการแก้ไขเพิ่มเติมส่วนใดอีกบ้าง ซึ่งจะประหยัดเวลาและทรัพยากรมากกว่าการทำงานจนเสร็จ ทั้งหมดแล้วต้องแก้ไขในภายหลัง

5.3 ข้อเสนอแนะ

ในการพัฒนาระบบควบคุมกระบวนการต่างๆ ซึ่งลูกค้าส่วนใหญ่มักมีความต้องการที่หลากหลาย สิ่งที่สำคัญที่สุดคือการวางโครงสร้างของโปรแกรมควบคุมอย่างเป็นระบบ ง่ายต่อการเข้าใจและการจัดการ เพื่อช่วยลดความยุ่งยากและประหยัดเวลาในการแก้ไขโปรแกรม เนื่องมาจากการปรับเปลี่ยนแนวคิดและ อุปกรณ์ในภายหลัง

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 หล้าหมั่น และคณะ; เทคโนโลยีการผลิตอาหารสัตว์อัดเม็ดประจำปี 2556; แหล่งที่มา: <http://www.agro.cmu.ac.th/absc/data/57/57-008.pdf> (สืบค้นวันที่ 3 พฤศจิกายน 2559)
- [3] ชูศักดิ์ ลิ้มสกุล 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.rockwellautomation.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 - นามสกุล นางสาวปริชมน ต้นไธง

วัน เดือน ปีเกิด 20 กันยายน 2537

ที่อยู่ 559/165 หมู่ 2 ถนนประชาพัฒนา แขวงทับยาว เขตลาดกระบัง กรุงเทพฯ 10520

E-mail parich.bell@gmail.com

โทรศัพท์ 097-1927267

ประวัติการศึกษา

- พ.ศ.2553 – 2555 ระดับมัธยมศึกษาตอนปลาย โรงเรียนนวมินทราชินูทิศ หอวัง นนทบุรี
- พ.ศ.2556 – ปัจจุบัน วิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหารลาดกระบัง

ประสบการณ์

- นักศึกษาฝึกงาน แผนก Maintenance Engineering
บริษัท ปตท. สำรวจและผลิตปิโตรเลียม จำกัด (มหาชน)
- นักศึกษาโครงการสหกิจศึกษา บริษัท อินคอนโทรล จำกัด