รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบสกาดาสำหรับกระบวนการอัดเม็ดอาหารสัตว์
SCADA System for Animal-Feed Pelleting Process

นายณัฐปคัลภ์ หรรษคุณาฒัย

หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

T148522

รายงานสหกิจศึกษาฉบับสมบูรณ์

ระบบสกาดาสำหรับกระบวนการอัดเม็ดอาหารสัตว์
SCADA System for Animal-Feed Pelleting Process

นายณัฐปคัลภ์ หรรษคุณาฒัย

ว.สว.
ส243365
2559
ระบบหมู่ 148522
เลขทะเบียน
วัน.เดือน.ปี 30 มี.ค. 2560
b12870365
b0026869

หลักสูตรวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอัตโนมัติ
ภาควิชาวิศวกรรมการวัดและควบคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2559

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชื่อโครงงานสหกิจศึกษา ระบบสกาดาสำหรับกระบวนการอัดเม็ดอาหารสัตว์
ชื่อ – สกุล นักศึกษา นายณัฐปคัลภ์ หรรษคุณาฒัย
คณะ วิศวกรรมศาสตร์ **สาขาวิชา** วิศวกรรมอัตโนมัติ
ชื่อ – สกุล อาจารย์นิเทศ รศ.ดร.ไสว พงศ์สวัสดิ์
ผศ.ดร.ธีรวัฒน์ เทพมณี
ชื่อ – สกุล ผู้นิเทศงาน นายชำนาญ ตุมทอง
ชื่อสถานประกอบการ บริษัท อินคอนโทรล จำกัด

บทคัดย่อ

โครงการนี้นำเสนอระบบสกาดาสำหรับการควบคุมกระบวนการอัดเม็ดของโรงงานผลิตอาหารสัตว์แห่งหนึ่ง ระบบสกาดาที่นำเสนอประกอบไปด้วยพีแอลซีรุ่น ControlLogix 5570 เวิร์คสเตชันสำหรับการแสดงผลและติดต่อกับผู้ใช้งานด้วยซอฟท์แวร์ ArchestrA IDE เซิร์ฟเวอร์สำหรับฐานข้อมูล และเซิร์ฟเวอร์สำหรับเอโอเอส โดยที่อุปกรณ์ฮาร์ดแวร์ทั้งหมดติดตั้งบนเครือข่าย EtherNet/IP ระบบสกาดาใหม่นี้ถูกสร้างขึ้นเพื่อใช้งานแทนระบบเดิมซึ่งใช้แผงแสดงการทำงาน ทั้งนี้เพื่อเป็นการปรับปรุงประสิทธิภาพการควบคุมกระบวนการ จากผลการทดลองยืนยันได้ว่าระบบสกาดาที่สร้างขึ้นสามารถทำงานได้อย่างถูกต้อง

คำสำคัญ : กระบวนการอัดเม็ดอาหารสัตว์, ระบบ SCADA, การควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Co-operative Title: SCADA System for Animal-Feed Pelleting Process
Student intern name: Mr. Natpakal Hansakuntai
Faculty: Engineering **Department:** Automation Engineering
Advisor name: Assoc. Prof. Dr. Sawai Pongsawatd
Asst. Prof. Dr. Teerawat Thepmanee
Mentor name: Mr. Chumnan Tumtong
Company: InControl Company Limited

ABSTRACT

This project presents a supervisory control and data acquisition (SCADA) system for controlling pelleting process of an animal feed manufacturer. The proposed SCADA consists of a programmable logic controller (PLC) modeled ControlLogix 5570, a human machine interface (HMI) workstation installed ArchestrA IDE software, historian data server, and two Automation Object Servers (AOSs). All hardware components are connected to the EtherNet/IP network. The new SCADA system is implemented to replace the previous system using Mimic Diagram Panel for improving the control effectiveness. Experimental results confirm that the implemented SCADA system can be correctly operated.

Keywords : Animal-Feed Pelleting Process, SCADA System, Control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

รายงานสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้ลุล่วงด้วยดี เนื่องด้วยความอนุเคราะห์จากบริษัท อินคอนโทรล จำกัด ที่ให้โอกาสสำหรับโครงการสหกิจศึกษา อีกทั้งคุณชำนาญ ตุมทอง ผู้นิเทศงาน และพนักงานบริษัททุกท่านที่ให้ความช่วยเหลือ และให้ประสบการณ์ในการทำงานตลอดระยะเวลาสี่เดือน

ขอขอบพระคุณ รศ.ดร.ไสว พงศ์สวัสดิ์ และผศ.ดร.ธีรวัฒน์ เทพมณี ที่ได้ให้ความเมตตา และคำแนะนำแก่ผู้จัดทำตลอดมา ขอขอบพระคุณอาจารย์ภาควิชาวิศวกรรมอัตโนมัติทุกท่านที่ได้ให้คำแนะนำและช่วยเหลืออันเป็นประโยชน์ต่อการทำรายงานสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้

คุณค่า และประโยชน์อันพึงมีจากรายงานสหกิจศึกษาฉบับสมบูรณ์ฉบับนี้ผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

ณัฐปคัลภ์ หรรษคุณาตัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ **หน้า**

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญภาพ VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมา และความสำคัญ 1

1.2 วัตถุประสงค์ของโครงงาน 1

1.3 ขอบเขตของโครงงาน 1

1.4 วิธีดำเนินการโครงงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับ 2

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง 3

2.1 กล่าวนำ 3

2.2 การอัดเม็ดอาหารสัตว์ 3

2.2.1 อาหารสัตว์อัดเม็ด 3

2.2.2 การผลิตสัตว์อาหารแบบอัดเม็ด 4

2.3 ระบบ SCADA 7

บทที่ 3 การดำเนินงาน 10

3.1 กล่าวนำ 10

3.2 ระบบควบคุมกระบวนการอัดเม็ดอาหารสัตว์ที่สร้างขึ้น 10

3.2.1 โครงสร้างของระบบโดยภาพรวม 10

3.2.2 รายละเอียดในส่วนฮาร์ดแวร์ 11

3.2.3 ซอฟท์แวร์ที่ใช้งาน 12

3.3 ส่วนโปรแกรมควบคุม 14

3.4 ส่วนแสดงผล 21

บทที่ 4 ผลการดำเนินงาน 46

4.1 กล่าวนำ 46

4.2 ผลการทดสอบระบบ 46

4.2.1 การทดสอบการเชื่อมต่อส่วนแสดงผล และตัวควบคุม 46

4.2.2 การทดสอบการควบคุมอุปกรณ์ในกระบวนการผ่านส่วนแสดงผล 48

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

บทที่ **หน้า**

บทที่ 5 สรุปผล ปัญหา และข้อเสนอแนะ 53

5.1 สรุปผลการดำเนินงาน 53

5.2 ปัญหา และวิธีการแก้ไข 53

5.2.1 ปัญหาที่พบ 53

5.2.2 วิธีการแก้ไข 53

5.3 ข้อเสนอแนะ 53

เอกสารอ้างอิง 54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ **หน้า**

1.1 แผนการดำเนินงาน....2
3.1 ตัวอย่าง Data Type ของมอเตอร์....16
3.2 การใช้สีในการกำหนดสถานะอุปกรณ์....31
3.3 Setpoint ในการตรวจสอบ Alarm....36
3.4 เงื่อนไขในการเกิด Alarm....36

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

ภาพที่ **หน้า**

2.1 ตัวอย่างอาหารสัตว์อัดเม็ดขนาดต่าง ๆ 3
2.2 กระบวนการผลิตอาหารสัตว์อัดเม็ด 4
2.3 ตัวอย่างเครื่องบดแบบค้อนเหวี่ยง 5
2.4 ตัวอย่างเครื่องผสมแบบตั้ง 5
2.5 ตัวอย่างเครื่องผสมแบบริบบิ้นถังนอน 6
2.6 ตัวอย่างเครื่องอัดเม็ด 6
2.7 องค์ประกอบของระบบ SCADA 7
2.8 สถาปัตยกรรมของระบบ SCADA ยุคแรก 8
2.9 สถาปัตยกรรมของระบบ SCADA ยุคที่สอง 8
2.10 สถาปัตยกรรมของระบบ SCADA ยุคที่สาม 9
3.1 Network Architecture ของระบบผลิตอาหารสัตว์อัดเม็ด 10
3.2 ControlLogix 5570 11
3.3 โครงสร้างของซอฟท์แวร์ ArchestrA IDE 12
3.4 การทำงานประสานกันของซอฟแวร์ ArchestrA IDE 12
3.5 ซอฟท์แวร์ ArchestrA IDE 13
3.6 ซอฟท์แวร์ Wonderware InTouch 13
3.7 ซอฟท์แวร์ Wonderware InTouch 13
3.8 ซอฟท์แวร์ Studio 5000 14
3.9 ซอฟท์แวร์ Studio 5000 14
3.10 การสร้างโปรเจคใหม่ของ Studio 5000 15
3.11 การเลือกตัวควบคุมสำหรับโปรเจค 16
3.12 การนำเข้า Data Tpye 17
3.13 Data Tpye ที่นำเข้า 17
3.14 การสร้าง Tag ใหม่ 17
3.15 การกำหนดค่า Tag ใหม่ 18
3.16 Tag ที่สร้างขึ้นมาใหม่ 18
3.17 การสร้างโปรแกรมควบคุมใหม่ 19
3.18 โปรแกรมควบคุมที่สร้างขึ้นใหม่ 19
3.19 การสร้าง Routine ใหม่ 19
3.20 การกำหนดค่า Routine ใหม่ 20
3.21 ตัวอย่าง Routine ที่สร้างขึ้นใหม่ 20
3.22 ส่วนประกอบหลักของ Control Module 20
3.23 ตัวอย่างโปรแกรมควบคุมของอุปกรณ์ 21
3.24 Flow Diagram ของกระบวนการอัดเม็ดอาหารสัตว์ 21
3.25 หน้าแรกของการสร้าง Galaxy 22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.26 การ Creat New Galaxy 22
3.27 การ Connect Galaxy 23
3.28 การสร้าง Engine ใหม่ 23
3.29 ตัวอย่าง Engine ใน Platform 23
3.30 การสร้าง Area ใหม่ 24
3.31 ตัวอย่าง Area ใน Engine 24
3.32 การสร้าง Template ใหม่ 25
3.33 ตัวอย่าง Template 25
3.34 ตัวอย่าง Attributes ใน Template 26
3.35 ตัวอย่าง Scripts ใน Template 26
3.36 ตัวอย่าง Graphics ใน Template 27
3.37 การสร้าง Object ใหม่ 27
3.38 ตัวอย่าง Object ที่ยังไม่ได้ย้ายเข้าไปใน Area 28
3.39 ตัวอย่าง Object ใน Area 28
3.40 การสร้าง Symbol ใหม่ 28
3.41 Symbol ถังเก็บวัตถุดิบ 29
3.42 Graphics มอเตอร์ใน Template 29
3.43 Animations สำหรับ Graphics 29
3.44 ตัวอย่างการตั้งค่า Value Data 30
3.45 ตัวอย่างการตั้งค่า Action Script 30
3.46 ตัวอย่างการตั้งค่า Fill Style 30
3.47 การสร้าง InTouch Application ใหม่ 31
3.48 การเปิด InTouch Application ที่สร้างขึ้นใหม่ 32
3.49 การสร้างหน้า HMI ใหม่ 32
3.50 การกำหนดข้อมูลของหน้า HMI ใหม่ 32
3.51 การ Import หน้า HMI 33
3.52 การเปิดหน้า HMI ที่เคยสร้างไว้ 33
3.53 การเพิ่ม Symbol หรือ Graphic ลงในหน้า HMI 33
3.54 การเพิ่ม Symbol ลงในหน้า HMI 34
3.55 การเพิ่ม Graphic ลงในหน้า HMI 34
3.56 ตัวอย่าง Popup สำหรับอุปกรณ์ 35
3.57 ตัวอย่าง Popup Alarm 35
3.58 ตัวอย่าง Scripts สำหรับหน้าจอ HMI 37
3.59 ตัวอย่าง Application 37
3.60 ตัวอย่าง Key 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

ภาพที่ **หน้า**

3.61 ตัวอย่าง Condition 38
3.62 ตัวอย่าง Data Change 39
3.63 ตัวอย่าง QuickFunctios 39
3.64 หน้า HMI Pellet Mill ส่วนที่ 1 40
3.65 หน้า HMI Pellet Mill ส่วนที่ 2 40
3.66 หน้า HMI Pellet Mill ส่วนที่ 3 41
3.67 หน้า HMI Pellet Mill ส่วนที่ 4 41
3.68 ลำดับการเชื่อมต่อของส่วนแสดงผล และ PLC 42
3.69 การสร้าง Topic ใหม่สำหรับตัวควบคุม 43
3.70 การเลือกตัวควบคุมให้กับ Topic 43
3.71 การสร้าง OPC Object ใน DAServer Manager 44
3.72 การตั้งค่า OPC Object ใน DAServer Manager 44
3.73 การตั้งค่า OPC ใน ArchestrA IDE 45
3.74 การเพิ่ม Topic ใน ArchestrA IDE 45
4.1 การทดสอบ Download โปรแกรม PLC 46
4.2 การเปลี่ยนโหมด PLC 47
4.3 การ Deploy Object ของ HMI 47
4.4 การ Activate Server ใน SMC 48
4.5 I/O สามารถรับ-ส่งค่าได้ 48
4.6 การทำงานของมอเตอร์ในโหมด Auto 49
4.7 โปรแกรมควบคุมในโหมด Auto 49
4.8 การทำงานของมอเตอร์ในโหมด Manual 49
4.9 โปรแกรมควบคุมในโหมด Manual 50
4.10 มอเตอร์ในสถานะ Start 50
4.11 มอเตอร์ในสถานะ Stop 50
4.12 โปรแกรมควบคุมเมื่อ Start อุปกรณ์ 51
4.13 การทำงานของมอเตอร์ในโหมด Maintenance 51
4.14 โปรแกรมควบคุมในโหมด Maintenance 52
4.15 ตัวอย่างมอเตอร์ Fail Start 52
4.16 โปรแกรมควบคุมเมื่อเกิด Alarm Fail Start 52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา และความสำคัญ

บริษัทผลิตอาหารสัตว์แห่งหนึ่งเป็นผู้ผลิต และจำหน่ายอาหารสัตว์ โดยธุรกิจการผลิตอาหารสัตว์ของบริษัทนี้ มีผลิตภัณฑ์หลายชนิด ได้แก่ อาหารไก่ไข่ อาหารไก่เนื้อ อาหารสุกร อาหารวัว อาหารปลา อาหารกบ อาหารกุ้ง อาหารแพะ อาหารโคเนื้อ อาหารโคนม และอาหารนกกระทา เป็นต้น

จากการศึกษาการผลิตอาหารสัตว์ที่โรงงานแห่งหนึ่ง ซึ่งเป็นโรงงานของบริษัทแห่งนี้ พบว่าการผลิตอาหารสัตว์มี 3 กระบวนการหลัก คือ การบด (Grinding) การผสม (Mixing) และการอัดเม็ด (Pelleting) โดยมีการควบคุมกระบวนการผลิตด้วยผู้ปฏิบัติงานซึ่งผู้ปฏิบัติงานต้องทำการสั่งการเปิด-ปิดอุปกรณ์ในสายการผลิต เช่น มอเตอร์สายพาน เครื่องอัดเม็ด และวาล์วเปิด-ปิด ผ่าน Mimic Diagram Panel และต้องมีการประสานงานระหว่างผู้ปฏิบัติงานในแต่ละขั้นตอนการผลิต อีกทั้งข้อมูลการผลิตถูกบันทึกด้วยมือ (Manual Record) ซึ่งอาจเกิดความผิดพลาดได้ เนื่องจากประสิทธิภาพของการผลิตขึ้นอยู่กับประสิทธิภาพของผู้ปฏิบัติงานแต่ละบุคคล เพื่อลดข้อจำกัดจากการใช้ผู้ปฏิบัติงานในการควบคุมการผลิตอาหารสัตว์ ทางโรงงานจึงต้องการปรับปรุงการผลิตอาหารสัตว์ให้มีการทำงานแบบอัตโนมัติ รวมถึงการบันทึก และการจัดเก็บข้อมูลต่าง ๆ เกี่ยวกับการผลิตในรูปของไฟล์อิเล็กทรอนิกส์ ในโครงงานนี้มีการอธิบายเฉพาะการออกแบบ และสร้างระบบควบคุมอัตโนมัติสำหรับกระบวนการอัดเม็ดอัดเม็ดอาหารสัตว์เท่านั้น

ระบบควบคุมสำหรับกระบวนการอัดเม็ดอาหารสัตว์ที่สร้างขึ้นเป็นระบบควบคุมบนพื้นฐานสกาดา (Supervisory Control And Data Acquisition: SCADA) ที่มีตัวควบคุมพีแอลซี (Programmable Logic Controller: PLC) และส่วนแสดงผลและติดต่อผู้ใช้งาน (Human Machine Interface: HMI) โดยในส่วน HMI ใช้สำหรับการติดตามผล และควบคุมการทำงานของอุปกรณ์ต่าง ๆ เช่น มอเตอร์ และ วาล์วเปิด-ปิด เพื่อควบคุมกระบวนการผลิตเป็นไปตามเป้าหมาย นอกจากนี้ การใช้ระบบควบคุมอัตโนมัติที่สร้างขึ้นยังช่วยลดข้อผิดพลาดในการทำงานของผู้ปฏิบัติงาน และลดต้นทุนในการจ้างแรงงาน อีกทั้งเพิ่มประสิทธิภาพในการผลิตอาหารสัตว์อีกด้วย

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ศึกษากระบวนการผลิตอาหารสัตว์แบบอัดเม็ด

1.2.2 เขียนโปรแกรมสำหรับ PLC ในการควบคุมอุปกรณ์ในกระบวนอัดเม็ดอาหารสัตว์

1.2.3 สร้างส่วนแสดงผล HMI ของระบบ SCADA

1.3 ขอบเขตของโครงงาน

1.3.1 ศึกษา และเขียนโปรแกรม PLC รุ่น ControlLogix 5570 ในการควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์ โดยใช้ซอฟท์แวร์ Studio 5000

1.3.2 ศึกษา และสร้างส่วนแสดงผลสำหรับแสดงสถานะการทำงาน และควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์ โดยใช้ซอฟท์แวร์ ArchestrA IDE

เอกสารนี้เป็นลิขสิทธิ์ของมหาวิทยาลัยสงขลานครินทร์ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 วิธีดำเนินการโครงงาน

1.4.1 ค้นคว้าหาข้อมูลเกี่ยวกับกระบวนการอัดเม็ดอาหารสัตว์ และซอฟท์แวร์ที่เกี่ยวข้องสำหรับสร้างโปรแกรมควบคุม และส่วนแสดงผล

1.4.2 เขียนโปรแกรมควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์

1.4.3 สร้างส่วนแสดงผลของระบบ SCADA สำหรับแสดงผล และควบคุมการทำงานของอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์

1.4.4 ทดสอบ และปรับปรุงการทำงานของระบบ

ตารางที่ 1.1 แผนการดำเนินงาน

แผนการดำเนินงาน	DATE	8-ส.ค.-59	15-ส.ค.-59	22-ส.ค.-59	29-ส.ค.-59	5-ก.ย.-59	12-ก.ย.-59	19-ก.ย.-59	26-ก.ย.-59	3-ต.ค.-59	10-ต.ค.-59	17-ต.ค.-59	24-ต.ค.-59	31-ต.ค.-59	7-พ.ย.-59	14-พ.ย.-59	21-พ.ย.-59
	WEEK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ศึกษากระบวนผลิตอาหารสัตว์แบบอัดเม็ด																	
ศึกษา และค้นคว้าข้อมูลของซอฟท์แวร์ที่เกี่ยวข้อง																	
สร้างโปรแกรม PLC สำหรับควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์																	
สร้างส่วนแสดงผลของระบบ SCADA																	
ตรวจสอบ และทดสอบการทำงานของระบบ																	
จัดทำรูปเล่มสหกิจศึกษาฉบับสมบูรณ์																	

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ช่วยเพิ่มความสะดวกแก่ผู้ปฏิบัติงาน

1.5.2 ลดค่าใช้จ่ายในการว่าจ้างผู้ปฏิบัติงาน

1.5.3 ลดข้อผิดพลาดในการทำงาน

1.5.4 กระบวนการผลิตมีประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 กล่าวนำ

ในบทนี้กล่าวถึงทฤษฎีที่เกี่ยวข้องกับโครงงาน ได้แก่ การอัดเม็ดอาหารสัตว์ อาหารสัตว์อัดเม็ด และระบบสกาดา

2.2 การอัดเม็ดอาหารสัตว์ [1]-[3]

2.2.1 อาหารสัตว์อัดเม็ด

อาหารสัตว์อัดเม็ด ดังตัวอย่างในภาพที่ 2.1 เป็นอาหารสัตว์ที่มีคุณค่าของอาหารสูงเพราะปริมาณที่กินได้มากขึ้น โดยขนาดของเม็ดอาหารที่อัดต้องมีขนาดเหมาะสมนานพอที่ทำให้อยู่ในกระเพาะรูเมนจนเซลลูโลสถูกแบคทีเรียย่อยเพื่อเปลี่ยนเป็นกรดไขมันที่ระเหยได้ แล้วนำไปใช้เป็นพลังงานได้ แต่ถ้าขนาดเม็ดที่อัดมีขนาดเล็กเกินไปอาหารจะเคลื่อนผ่านไปเร็ว และไปถูกย่อยในลำไส้เล็ก โดยทั่วไปพบว่าอาหารอัดเม็ดจะเคลื่อนที่ผ่านทางเดินอาหารเร็วขึ้นทำให้การย่อยได้ของเยื่อใยต่ำลง นอกจากนี้ยังทำให้สัตว์เคี้ยวเอื้องกินอาหารได้มากขึ้นทำให้อาหารเคลื่อนที่เร็ว โอกาสของการย่อยได้ของอาหารโดยรวมมักต่ำลง ในสัตว์เคี้ยวเอื้องที่อายุยังน้อยได้ประโยชน์จากการใช้อาหารอัดเม็ดมากกว่า ดังนั้น การอัดเม็ดอาหารหยาบต้องคำนึงถึงสัตว์ที่เลี้ยง ขนาดของเม็ด และค่าใช้จ่ายในการอัดตัวอย่าง เช่น การอัดกากสับปะรดหรืออัลฟัลฟ่า ใช้ประโยชน์ในแง่การเก็บรักษา และขนส่ง แต่เพิ่มคุณค่าได้สูงกว่าการไม่อัดเม็ด

ภาพที่ 2.1 ตัวอย่างอาหารสัตว์อัดเม็ดขนาดต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 การผลิตสัตว์อาหารแบบอัดเม็ด

การผลิตอาหารสัตว์แบบอัดเม็ดสามารถแบ่งกระบวนการผลิตออกเป็น 3 กระบวนการหลัก คือ การบด (Grinding) การผสม (Mixing) และการอัดเม็ด (Pelleting) ดังแสดงในภาพที่ 2.2

ภาพที่ 2.2 กระบวนการผลิตอาหารสัตว์อัดเม็ด

2.2.2.1 การบด (Grinding)

การบด หรือการลดขนาดของวัตถุดิบ ถือเป็นกระบวนการหลักของอุตสาหกรรมอาหารสัตว์ มีหลายเหตุผลที่ต้องนำวัตถุดิบเข้าเครื่องบดนี้ ได้แก่ เพื่อลดขนาดของเส้นใย หรืออนุภาคลง ต้องการลดความชื้นลงโดยการเติมอากาศ เพื่อเติมสารต่าง ๆ ในขณะบด หรือผสม เพื่อเพิ่มความสะดวกในการจัดการ หรือจัดเก็บ เป็นต้น โดยปกติการบดช่วยในเรื่องของการย่อยอาหาร และการดูดซับสารอาหาร การผสมวัตถุดิบ การอัดเม็ดอาหารสัตว์ และเพิ่มความหนาแน่นของส่วนผสมอีกด้วย

การบดวัตถุดิบอาหารสัตว์มีความจำเป็นในกรณีที่วัตถุดิบอาหารสัตว์มีขนาดใหญ่และไม่ละเอียด เช่น เมล็ดถั่วเหลือง เมล็ดข้าวโพด มันเส้น ปลายข้าว และเปลือกกุ้ง โดยวัตถุดิบเหล่านี้ต้องนำมาบดให้ละเอียดด้วยเครื่องบดอาหาร สำหรับวัตถุดิบอาหารสัตว์ที่มีขนาดเล็ก เช่น ปลาป่น รำข้าว แป้งสาลี แป้งมันสำปะหลัง ไม่จำเป็นต้องบดให้ละเอียด แต่ถ้าเป็นการผลิตอาหารในโรงงานอาหารสัตว์น้ำ ซึ่งมีการควบคุมอุณหภูมิอย่างดีจะบดวัตถุดิบอาหารสัตว์ทุกประเภทให้มีขนาดละเอียดเท่ากันหมด โดยอาจบด 1 หรือ2 ครั้ง แล้วแต่กรณีว่ามีขนาดเล็กใหญ่ประการใด สำหรับวัตถุดิบอาหารสัตว์จำพวกวิตามิน แร่ธาตุ และกรดอะมิโนสังเคราะห์ มีขนาดละเอียดอยู่แล้วจึงไม่ต้องนำมาบดอีก และอีกทั้งความร้อนที่เกิดจากการบดทำให้วิตามินแร่ธาตุ และกรดอะมิโนสังเคราะห์ถูกทำลายไปอย่างรวดเร็ว ดังนั้นจึงนำไปผสมใช้ได้เลยโดยไม่ต้องบด โดยหลักการแล้วการบดวัตถุดิบอาหารสัตว์ให้ละเอียดนั้นช่วยเพิ่มพื้นที่ผิวของวัตถุดิบอาหารสัตว์ให้มากขึ้น ซึ่งมีผลทำให้สามารถผสมวัตถุดิบอาหารสัตว์ให้เป็นเนื้อเดียวกันได้ง่ายขึ้น และช่วยให้อาหารเม็ดที่อัดออกมามีความหนาแน่น และยึดตัวดีขึ้น นอกจากนี้การบดวัตถุดิบอาหารสัตว์ให้ละเอียดยังมีส่วนช่วยให้ปลาสามารถย่อย และดูดซึมธาตุอาหารไปใช้ประโยชน์ได้ดีขึ้น หรือสัมประสิทธิ์การย่อยอาหารสัตว์สูงขึ้น

เครื่องบดที่นิยมใช้สำหรับการผลิตมีหลายประเภท เช่น เครื่องบดแบบปั่น (Blender) เครื่องบดแบบโม่ (Burr mill, Roller mill) และที่นิยมใช้มากที่สุดเครื่องบดแบบค้อนเหวี่ยง (Hammer mill) ดังตัวอย่างที่แสดงในภาพที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 2.3 ตัวอย่างเครื่องบดแบบค้อนเหวี่ยง

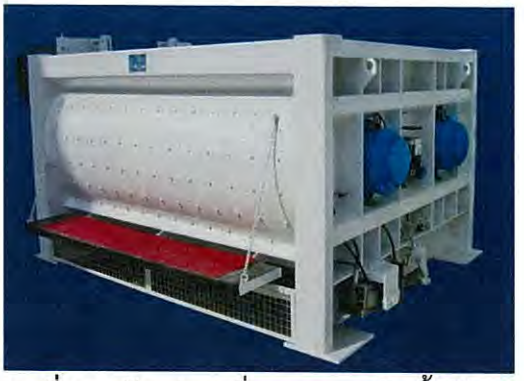
2.2.2.2 การผสม (Mixing)

การผสม เป็นกระบวนการที่ช่วยให้วัตถุดิบที่กำหนดเป็นส่วนประกอบของอาหารผสมคลุกเคล้าเป็นเนื้อเดียวกัน วัตถุดิบทุกชนิดกระจายตามเนื้อของส่วนผสมอย่างทั่วถึง กล่าวคือ หลังจากการผสมแล้ว หากแบ่งส่วนผสมมาตรวจหาส่วนประกอบของวัตถุดิบ ส่วนผสมที่เกิดจากการผสมที่ดีต้องมีวัตถุดิบครบถ้วนทั้งชนิด และปริมาณตรงตามที่ระบุในสูตรอาหารก่อนการผสม การผสมวัตถุดิบมักทำเพื่อการผสมวิตามิน และแร่ธาตุ ซึ่งมักผสมเตรียมไว้ต่างหากก่อนการผสมวัตถุดิบอื่นๆ เนื่องจากวิตามิน และแร่ธาตุที่ใส่เป็นส่วนผสมในอาหารมีปริมาณน้อย จึงนิยมผสมวิตามิน หรือแร่ธาตุกับวัตถุดิบชนิดอื่นเป็นวิตามินรวม หรือแร่ธาตุรวมก่อน แล้วจึงนำวิตามินรวม และแร่ธาตุรวมไปผสมกับวัตถุดิบอื่นตามต้องการ ต่อไปวิธีการเจือจางวิตามิน หรือแร่ธาตุกับวัตถุดิบเช่นนี้ช่วยให้ส่วนผสมของอาหารมีวิตามิน และแร่ธาตุตามต้องการ นอกจากนั้นก็เพื่อให้วิตามิน และแร่ธาตุซึ่งเป็นส่วนผสมที่มีปริมาณน้อยกระจายทั่วเนื้ออาหาร

เครื่องผสมมีหลายแบบ หลายขนาด และมีคุณสมบัติแตกต่างกันตามที่ผู้ใช้เลือก เช่น เครื่องผสมแบบตั้ง (Vertical Mixer) ดังตัวอย่างภาพที่ 2.4 เครื่องผสมแบบริบบิ้นถังนอน (Horizontal Ribbon Mixer) ดังตัวอย่างภาพที่ 2.5 เครื่องผสมแบบใบพายถังนอน (Horizontal Shaft Paddle Type Mixer)

ภาพที่ 2.4 ตัวอย่างเครื่องผสมแบบตั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.5 ตัวอย่างเครื่องผสมแบบริบบิ้นถังนอน

2.2.2.3 การอัดเม็ด (Pelleting)

การอัดเม็ด เป็นการนำเอาวัตถุดิบอาหารที่บดแล้ว หรือผสมตามต้องการมาคลุกรวมกันแล้วอัดผ่านรูตะแกรง ขนาดของเม็ดอาหารมีขนาดเล็ก หรือขนาดใหญ่ขึ้นกับความสัมพันธ์ของรูตะแกรง ความร้อน และแรงอัด การอัดเม็ดเป็นการลดการเป็นฝุ่นของอาหาร เช่น การอัดเม็ดมันสำปะหลังเพิ่มความน่ากิน และไม่ทำให้สูญเสียส่วนประกอบทางสารอาหารจากอาหารที่มีขนาดชิ้นส่วนเล็ก หรือเป็นผงไม่ให้ฟุ้งกระจายไป การจัดเก็บใช้พื้นที่น้อยลงเนื่องจากมีความหนาแน่นของเนื้อวัตถุสูง นอกจากนี้ยังทำให้อาหารมีคุณค่าสูงขึ้น ความร้อนที่เกิดขึ้นในขณะที่อัดผ่านรูตะแกรงทำให้แป้งบางส่วนสุก หรือเปลี่ยนโครงสร้างทำให้เอนไซม์ของสัตว์ทำการย่อยได้ง่ายขึ้น หรือจุลินทรีย์ทำงานได้มากกว่า แรงอัดทำให้ผนังเซลล์ของพืชแตกทำให้ย่อยได้ง่ายขึ้น การอัดนี้สามารถใช้กับวัตถุดิบที่เป็นอาหารทั้งชนิดเดียวกัน เช่น เมล็ดพืช หรือพืชแห้งที่ผ่านการสับแล้ว และอาหารผสม ขนาดของวัตถุดิบที่ใช้มีขนาดใกล้เคียงกันโดยทั่วไปใช้ขนาดลอดรูตะแกรง 2 - 2.5 มิลลิเมตรทำให้จับตัวเป็นเม็ดแข็ง และไม่ร่วนง่ายเมื่อกระทบกระเทือนจากการขนส่ง การใช้ไอน้ำผสมก่อนการอัดเม็ดทำให้อัดง่ายขึ้น และการสึกหรอของรูตะแกรงลดลงด้วย ปริมาณไอน้ำต้องพอเหมาะถ้ามากเกินไปเม็ดอาหารทำให้นิ่ม เมื่ออัดแล้วควรทิ้งระยะเวลาให้เย็นลงก่อนบรรจุ ปล่อยให้ความร้อนจากการอัด และไอน้ำแทรกอยู่ในเม็ดอาหารระเหยไป ถ้ารีบบรรจุไอน้ำที่คายออกมาในภาชนะจะทำให้อาหารสัตว์ขึ้นราได้ง่าย โดยตัวอย่างเครื่องอัดเม็ดแสดงในภาพที่ 2.6

ภาพที่ 2.6 ตัวอย่างเครื่องอัดเม็ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบ SCADA [4]-[6]

SCADA (Supervisory Control And Data Acquisition) เป็นระบบตรวจสอบ และวิเคราะห์ข้อมูลแบบ Real-time ใช้ในการตรวจสอบสถานะตลอดจนถึงควบคุมการทำงานของระบบควบคุมในอุตสาหกรรม และงานวิศวกรรมต่าง ๆ เช่น งานด้านโทรคมนาคมสื่อสาร การประปา การจัดการด้านพลังงาน อุตสาหกรรมการกลั่นน้ำมัน อุตสาหกรรมเคมี และกระบวนการนิวเคลียร์ในโรงไฟฟ้า ตัวอย่างการใช้งาน เช่น ใช้ ระบบ SCADA ตรวจสอบข้อมูลของการรั่วไหลของของเหลวที่เกิดขึ้นในท่อขนส่งจากตัวตรวจจับแล้วส่งสัญญาณแจ้งเตือนให้ผู้ปฏิบัติงานทราบ โดยส่งข้อมูลเข้าสู่ส่วนกลางของระบบ SCADA

นอกจากนั้น SCADA อาจทำหน้าที่คำนวณ และประมวลผลข้อมูลที่ได้จากฮาร์ดแวร์ต่าง ๆ เช่น PLC, Controller, DCS, RTU แล้วแสดงข้อมูลทางหน้าจอ หรือส่งสัญญาณควบคุมฮาร์ดแวร์ดังกล่าว เช่น หากอุณหภูมิของอุปกรณ์สูงเกินพิกัดให้ทำการปิดอุปกรณ์นั้น เป็นต้น โดยสั่งงานผ่าน PLC หรือ Controller ที่ติดต่ออยู่ ทั้งนี้ระบบ SCADA สามารถเก็บรวบรวมข้อมูลที่ได้จากระบบควบคุมทั้งหมดไว้ในฐานข้อมูลเพื่อให้ผู้ปฏิบัติงาน หรือโปรแกรมอื่น ๆ สามารถนำไปใช้งานได้ ระบบ SCADA นั้นเข้าไปมีส่วนในงานควบคุมทั้งเล็ก และใหญ่ที่ต้องการแสดงผล แลกเปลี่ยนข้อมูล หรือควบคุมระบบต่าง ๆ จากส่วนกลาง เพื่อการทำงานของระบบรวมที่สัมพันธ์กัน มองเห็นภาพรวมได้อย่างชัดเจน และมีความรวดเร็วต่อเหตุการณ์ต่าง ๆ ที่เกิดขึ้น

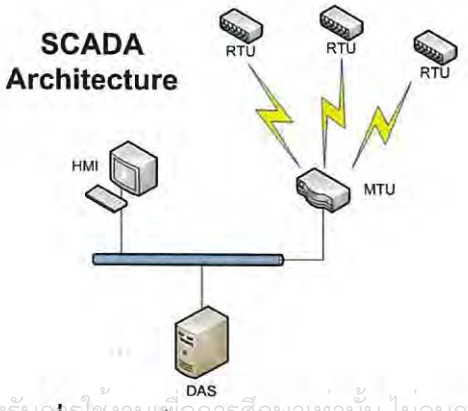
2.3.1 องค์ประกอบของระบบ SCADA มีส่วนประกอบหลัก 4 ส่วน ดังตัวอย่างภาพที่ 2.7

1) Field Data Interface Devices ทำหน้าที่เสมือน หู และตาของระบบ SCADA เพื่อทำการรวบรวมข้อมูล และส่งไปยังศูนย์กลางระบบ SCADA ซึ่งอาจเป็น RTUs (Remote Terminal Unit) หรือ PLCs ที่ได้เชื่อต่อกับอุปกรณ์ภาคสนาม หรือเซ็นเซอร์ที่เชื่อมต่อกับเครื่องจักร หรืออุปกรณ์ที่ถูกควบคุม หรือถูกตรวจสอบ

2) Communication Network เป็นการส่งข้อมูลดิจิตอลระหว่างสถานที่หนึ่งไปยังสถานที่หนึ่ง โดยผ่านตัวกลางในการติดต่อสื่อสาร เช่น สายเคเบิล คลื่นวิทยุ หรือดาวเทียม เป็นต้น

3) Central Host Computer เป็นคอมพิวเตอร์ หรือเครือข่ายของเซิร์ฟเวอร์คอมพิวเตอร์ที่ทำหน้าที่เป็นศูนย์กลางของระบบ SCADA โดยรับข้อมูลมาจากส่วนของ RTUs หรือ PLCs เพื่อประมวลผล และแสดงกระบวนการในรูปแบบที่ผู้ปฏิบัติงานสามารถทำงานได้

4) Operator Workstations เป็นส่วนของคอมพิวเตอร์ที่ทำการเชื่อมต่อกับ Central Host Computer ผ่านทาง LAN หรือ WAN ซึ่งกระทำเหมือนเป็น Client ที่เรียกข้อมูลจาก Server หรือ Central Host Computer เพื่อนำมาแสดงข้อมูลที่ต้องการให้แก่ผู้ปฏิบัติงาน

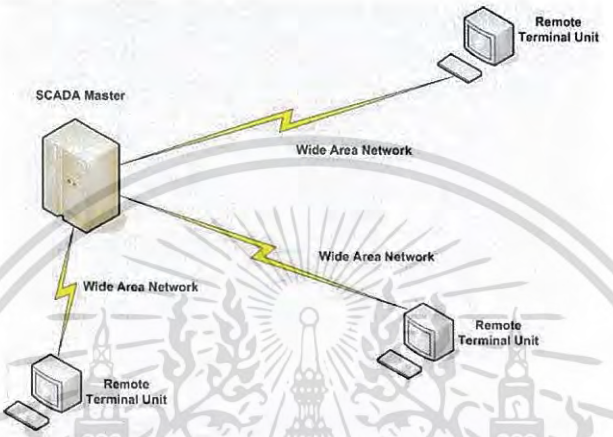


ภาพที่ 2.7 องค์ประกอบของระบบ SCADA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

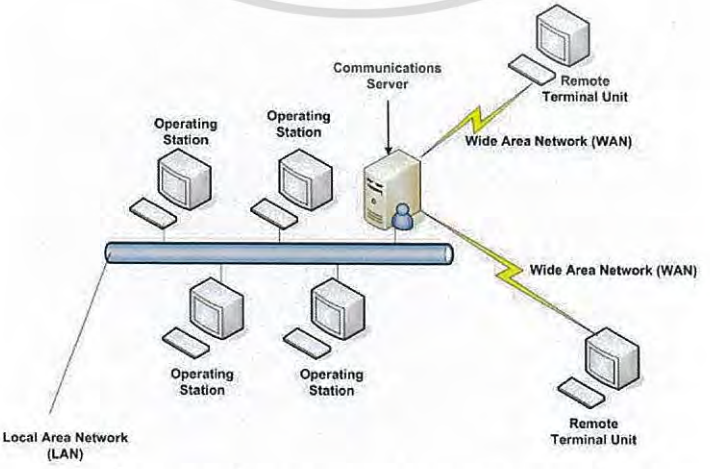
2.3.2 สถาปัตยกรรมของระบบ SCADA มีวิวัฒนาการ 3 ยุคดังนี้

1) ยุคแรก Monolithic ในยุคแรกนี้การคำนวณถูกกระทำโดยคอมพิวเตอร์เมนเฟรม แต่ระบบเครือข่าย (Networking) ยังไม่เกิดในเวลาที่ระบบ SCADA ได้รับการพัฒนา ดังนั้นระบบ SCADA เป็นระบบที่อิสระไม่มีการเชื่อมต่อกับระบบอื่น ๆ เครือข่ายส่วนใหญ่ได้รับการออกแบบมาโดยผู้ขาย RTU ในการสื่อสารกับ RTU. โปรโตคอลการสื่อสารที่มักถูกนำมาใช้เป็นกรรมสิทธิ์เฉพาะของผู้ขายในเวลานั้น ดังภาพที่ 2.8 แสดงสถาปัตยกรรมของระบบ SCADA ในยุคแรก



ภาพที่ 2.8 สถาปัตยกรรมของระบบ SCADA ยุคแรก

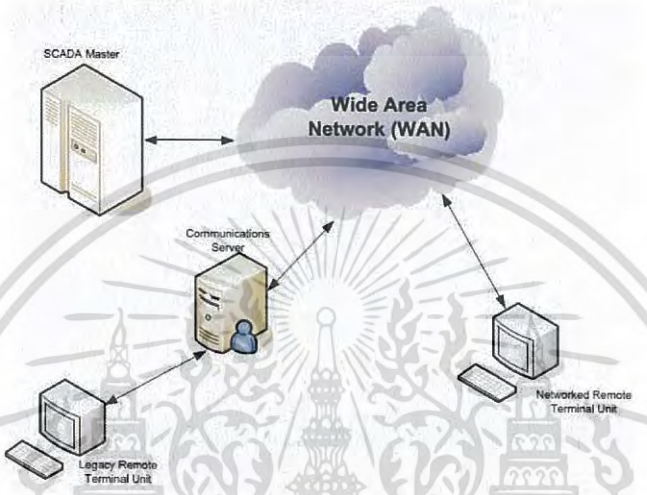
2) ยุคที่สอง Distributed การประมวลผลถูกกระจายไปหลายสถานีที่มีการเชื่อมต่อผ่านระบบ LAN และมีการใช้ข้อมูลร่วมกันในเวลาจริง แต่ละสถานีจึงรับผิดชอบสำหรับงานเฉพาะทำให้ขนาดและค่าใช้จ่ายของแต่ละสถานีน้อยกว่าสถานีในรุ่นแรก โปรโตคอลเครือข่ายที่ใช้ก็ยังคงเป็นกรรมสิทธิ์เฉพาะของผู้ผลิต ซึ่งนำไปสู่ปัญหาด้านความปลอดภัยของระบบ SCADA ที่ได้รับความสนใจจากแฮกเกอร์ เนื่องจากโปรโตคอลที่เป็นกรรมสิทธิ์เฉพาะจึงมีคนน้อยมากที่รู้ถึงวิธีการกำหนดวิธีการรักษาความปลอดภัยของการติดตั้งนอกจากนักพัฒนา และแฮกเกอร์ ดังนั้น ความปลอดภัยของการติดตั้งระบบ SCADA มักถูกประเมินเกินกว่าความเป็นจริงอย่างมาก ดังภาพที่ 2.9 แสดงสถาปัตยกรรมของระบบ SCADA ในยุคที่สอง



ภาพที่ 2.9 สถาปัตยกรรมของระบบ SCADA ยุคที่สอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ยุคที่สาม Networked เนื่องจากมีการใช้งานของโปรโตคอลมาตรฐาน อีกทั้งหลายระบบ SCADA ที่อยู่ในเครือข่ายสามารถเข้าถึงได้จากอินเทอร์เน็ต ทำให้ระบบอาจมีความเสี่ยงการโจมตีระยะไกล ในทางตรงกันข้ามการใช้โปรโตคอลมาตรฐาน และเทคนิคการรักษาความปลอดภัยนี้สามารถช่วยให้การปรับปรุง และดูแลรักษาความปลอดภัยของระบบ SCADA ได้ทันท่วงที ดังภาพที่ 2.10 แสดงสถาปัตยกรรมของระบบ SCADA ในยุคที่สาม



ภาพที่ 2.10 สถาปัตยกรรมของระบบ SCADA ยุคที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

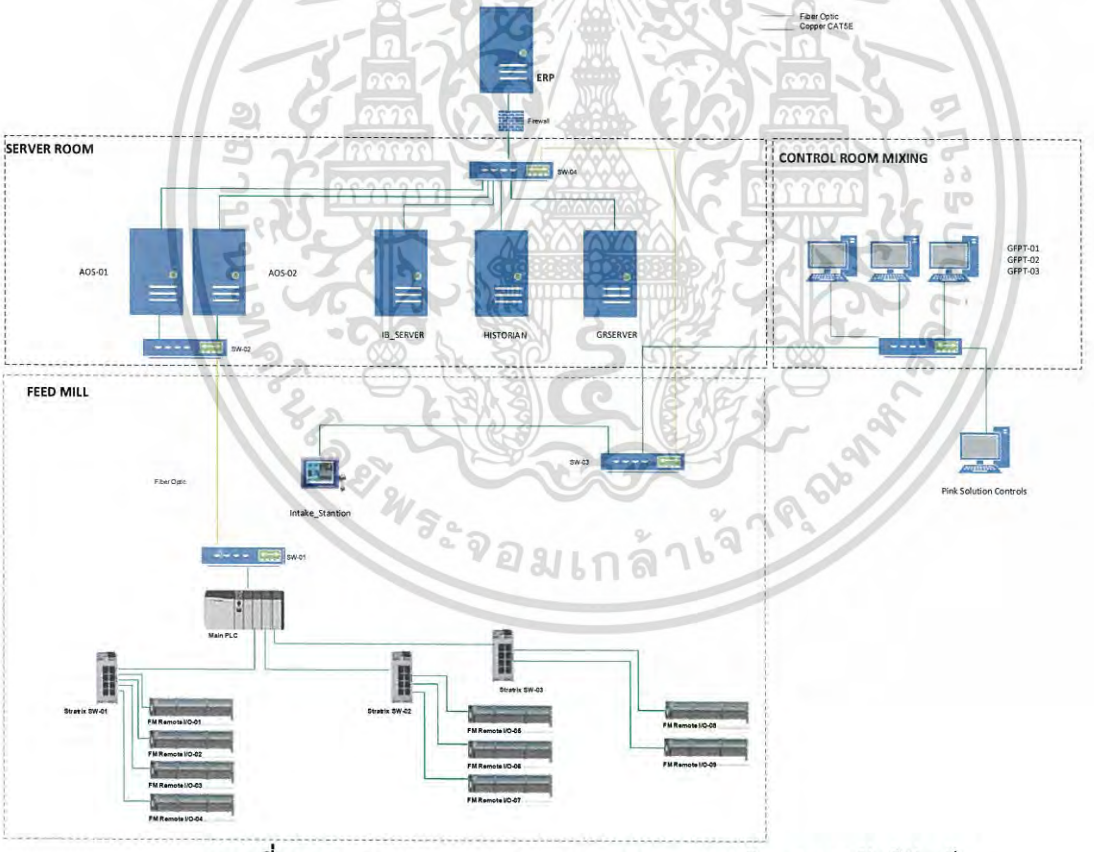
บทที่ 3
การดำเนินงาน

3.1 กล่าวนำ

การสร้างระบบควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์นั้น ประกอบด้วย 2 ส่วน ได้แก่ ส่วนโปรแกรมควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์ และส่วนแสดงผลกระบวนการอัดเม็ดอาหารสัตว์ โดยโครงการนี้ได้ทำออกแบบระบบที่ใช้ในการควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์ให้สามารถควบคุมอุปกรณ์ในกระบวนการผ่านทางส่วนแสดงผลของ ระบบ SCADA ซึ่งช่วยเพิ่มความสะดวกแก่ผู้ปฏิบัติงาน ลดข้อผิดพลาดในการทำงาน อีกทั้งยังช่วยเพิ่มประสิทธิภาพในการผลิตอีกด้วย

3.2 ระบบควบคุมกระบวนการอัดเม็ดอาหารสัตว์ที่สร้างขึ้น

3.2.1 โครงสร้างของระบบโดยภาพรวม



ภาพที่ 3.1 Network Architecture ของระบบผลิตอาหารสัตว์อัดเม็ด

ภาพที่ 3.1 แสดง Network Architecture ของระบบการผลิตอาหารสัตว์แบบอัดเม็ดซึ่งเป็นระบบที่ใช้ควบคุมกระบวนการหลัก ๆ ทั้ง 3 กระบวนการ คือ การบด (Grinding) การผสม(Mixing) และการอัดเม็ด (Pelleting) สำหรับระบบควบคุมนี้แบ่งออกเป็น 3 ส่วนหลัก คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

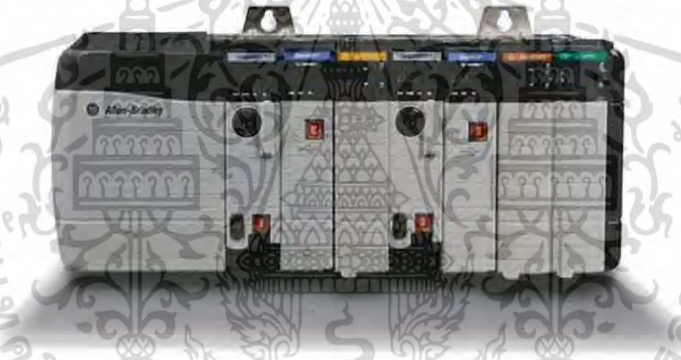
1) Server Room เป็นส่วนของเซิร์ฟเวอร์คอมพิวเตอร์ที่ใช้ในการเก็บรวบรวมข้อมูลต่าง ๆ ซึ่งเก็บเป็นส่วน ๆ ดังนี้ AOS และ GRSERVER เก็บส่วนแสดงผลของระบบ SCADA ทั้งหมด InBatch Server เก็บสูตรส่วนผสมของการผลิตของอาหารสัตว์ชนิดต่าง ๆ และ Historian เก็บจำนวน สถิติของอาหารสัตว์ที่ผลิตออกมา

2) Control Room Mixing เป็นส่วนของห้องควบคุมที่ใช้ติดตามผล และควบคุมอุปกรณ์ในการผลิตทั้งหมด ซึ่งอุปกรณ์ในห้องนี้เชื่อมต่อกับ Server Room เพื่อนำข้อมูลในระบบมาแสดงผลติดตามผล และควบคุม

3) Feed Mill เป็นส่วนของอุปกรณ์ควบคุม เช่น PLC, RTUs, Stations ที่ใช้ควบคุมอุปกรณ์การผลิต โดย RTUs มีทั้งหมด 9 หน่วย ได้เชื่อมต่อกับ PLC ซึ่งเป็นตัวควบคุมผ่านโปรโตคอล EtherNet/IP

3.2.2 รายละเอียดในส่วนฮาร์ดแวร์

3.2.2.1 ControlLogix 5570 [7]



ภาพที่ 3.2 ControlLogix 5570

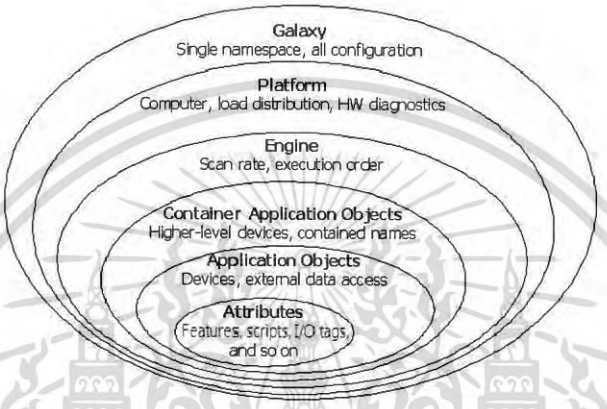
ControlLogix 5570 จากตัวอย่างในภาพที่ 3.2 เป็นผลิตภัณฑ์ประเภท PLC ของบริษัท Rockwell Automation ซึ่งอยู่ในตระกูล ControlLogix 5000 มีประสิทธิภาพในการควบคุมระบบหรือกระบวนการขนาดใหญ่ได้ อีกทั้งยังมีมาตรฐานด้านความปลอดภัยในการใช้งานในกระบวนการประเภท Motion และ Discrete มีความสามารถในการเชื่อมต่อมากมาย ได้แก่ EtherNet/IP, ControlNet, DeviceNet, Data Highway Plus, Remote I/O และ SynchLink เป็นต้น โดย PLC นี้มีจำนวน อินพุต/เอาท์พุต ที่สามารถต่อได้สูงสุด 128,000 จุดสำหรับดิจิตอล และ 4,000 จุดสำหรับอนาลอก โดยมีหน่วยความจำสูงสุด คือ 32 MB และใช้ซอฟท์แวร์ Studio 5000 ในการเขียนโปรแกรมสำหรับควบคุม PLC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ซอฟท์แวร์ที่ใช้งาน

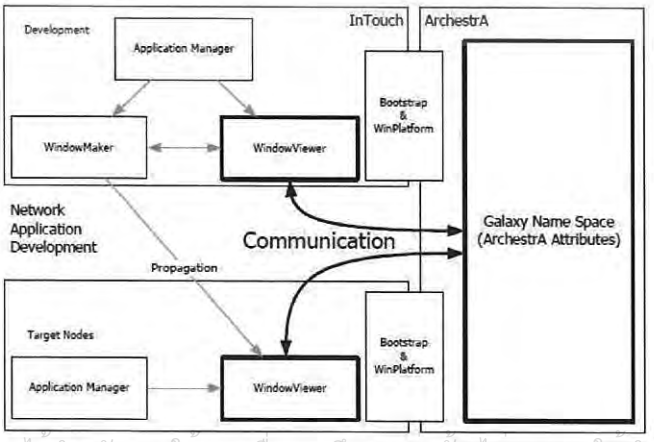
3.2.3.1 ArchestrA IDE [8]

ArchestrA IDE เป็นซอฟท์แวร์ที่ครอบคลุมเกี่ยวกับข้อมูลต่าง ๆ ของระบบควบคุมแบบอัตโนมัติที่ออกแบบมาเพื่อให้สามารถใช้ประโยชน์จากระบบเดิม ๆ ผ่านทางเทคโนโลยีของมาตรฐานอุตสาหกรรมใหม่ โดยมีส่วนประกอบที่แบ่งออกเป็นลำดับดังภาพที่ 3.3 ซึ่งแต่ละส่วนมีการทำงานประสานกันดังภาพที่ 3.4 และภาพที่ 3.5 แสดง UI ของซอฟแวร์นี้



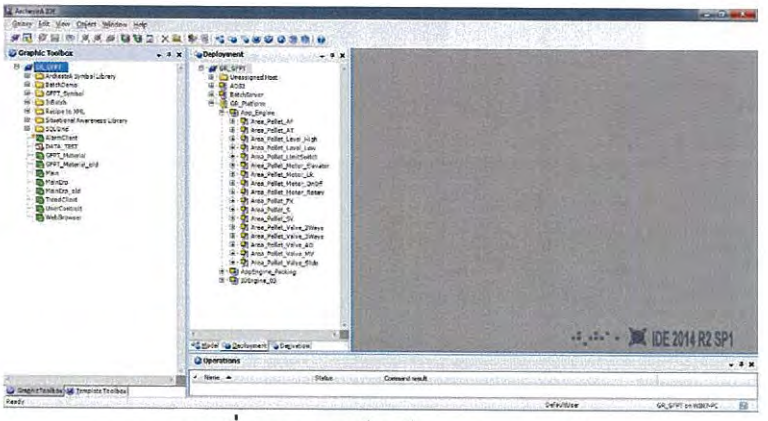
ภาพที่ 3.3 โครงสร้างของซอฟท์แวร์ ArchestrA IDE

1) Galaxy เป็นส่วนรวบรวมข้อมูลต่าง ๆ ไว้ทั้งหมดเสมือนเป็นกลุ่มของทั้งระบบ
2) Platform เปรียบเสมือนคอมพิวเตอร์ที่รวบรวมข้อมูลในระบบ
3) Engine เปรียบเสมือนลำดับที่ทำงานของคอมพิวเตอร์
4) Area เปรียบเสมือนส่วนรวบรวมอุปกรณ์ในระบบไว้ด้วยกัน
5) Object เปรียบเสมือนอุปกรณ์ของระบบที่ใช้ในการควบคุม
6) Attribute เป็นส่วนที่เชื่อมต่อกับส่วนอินพุต/เอาท์พุตของอุปกรณ์ในระบบ



ภาพที่ 3.4 การทำงานประสานกันของซอฟแวร์ ArchestrA IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

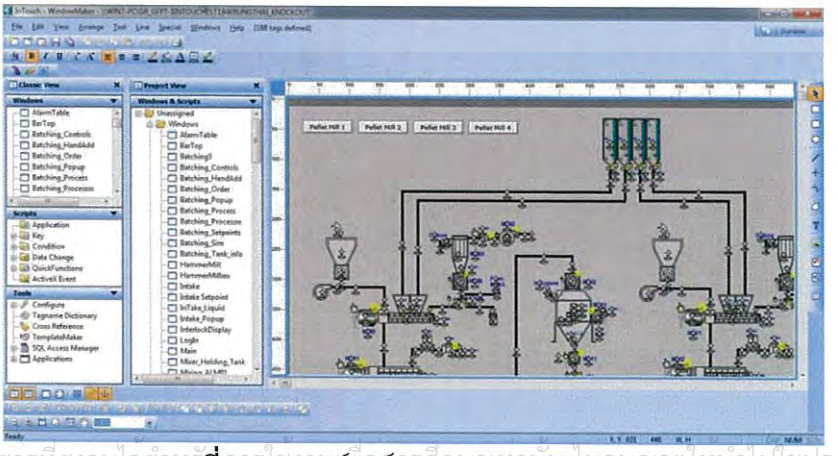


ภาพที่ 3.5 ซอฟท์แวร์ ArchestrA IDE

3.2.3.2 Wonderware InTouch [9]

Wonderware InTouch เป็นซอฟท์แวร์สำหรับการเขียนกราฟฟิกในอุตสาหกรรม ทั้ง HMI และระบบ SCADA จากบริษัท Invensys Process Systems ซึ่งปัจจุบันอยู่ภายใต้การบริหารโดย Schneider Electric โดยกราฟฟิกที่สร้างขึ้นแสดงกระบวนการต่าง ๆ ในระบบควบคุมของอุตสาหกรรม ผ่านส่วนแสดงผล ช่วยให้ผู้ปฏิบัติงานสามารถเห็นภาพรวมของกระบวนการ โดย UI ของซอฟแวร์นี้แสดงใน ภาพที่ 3.6 และภาพที่ 3.7

ภาพที่ 3.6 ซอฟท์แวร์ Wonderware InTouch



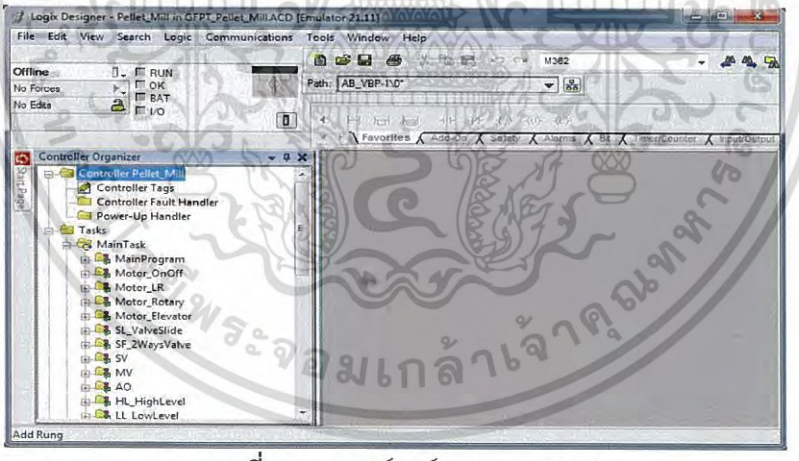
ภาพที่ 3.7 ซอฟท์แวร์ Wonderware InTouch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3.3 Studio 5000 [10]

Studio 5000 Logix Designer เป็นซอฟท์แวร์ที่ใช้สำหรับการออกแบบ พัฒนาและกำหนดค่าโปรแกรมที่ใช้สำหรับ Allen-Bradley PLC ตระกูล Logix 5000 และอุปกรณ์อื่น ๆ ที่เกี่ยวข้อง โดยทั่วไปแล้วมักใช้การเขียนโปรแกรมให้กับ PLC ซึ่งซอฟท์แวร์ Studio 5000 สามารถใช้ได้กับ PLC ตระกูล Logix 5000 ที่มีเฟิร์มแวร์ตั้งแต่เวอร์ชัน 21 ขึ้นไป โดย UI ของซอฟแวร์นี้แสดงในภาพที่ 3.8 และภาพที่ 3.9

ภาพที่ 3.8 ซอฟท์แวร์ Studio 5000



ภาพที่ 3.9 ซอฟท์แวร์ Studio 5000

3.3 ส่วนโปรแกรมควบคุม

ในการสร้างโปรแกรมควบคุมกระบวนการนั้นจำเป็นต้องรู้กระบวนการ ลำดับการทำงานของกระบวนการที่ต้องการสร้างโปรแกรมควบคุม อีกทั้งยังต้องทราบรายการอุปกรณ์ต่าง ๆ ในกระบวนการเพื่อนำมากำหนดจำนวนอินพุต/เอาท์พุตของส่วนโปรแกรมควบคุมเพื่อใช้ควบคุมอุปกรณ์ และส่งค่ามาแสดงผลในส่วนแสดงผลของกระบวนการ จากการศึกษาข้างต้นทำให้ทราบจำนวนอุปกรณ์ที่ใช้ในการควบคุมกระบวนการดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

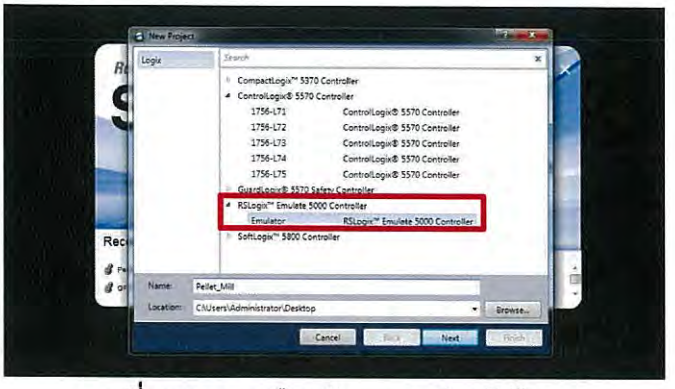
-	มอเตอร์ธรรมดา (Motor On-Off)	89	ตัว
-	มอเตอร์สองทิศทาง (Motor Forward-Reverse)	6	ตัว
-	มอเตอร์สำหรับยกของ (Motor Elevator)	6	ตัว
-	มอเตอร์แบบเลือกตำแหน่ง (Motor Rotary)	5	ตัว
-	วาล์วสไลด์ (Slide Valve)	24	ตัว
-	วาล์วสองทางเลือก (2 Ways Valve)	30	ตัว
-	วาล์วสามทางเลือก (3 Ways Valve)	2	ตัว
-	วาล์วเปิด-ปิด เป็น% (Butterfly Valve)	8	ตัว
-	เซนเซอร์ตรวจระดับของถัง (Level Switch)	34	ตัว
-	เซนเซอร์ตรวจจับการไหลของสายพาน (Proximity Switch)	18	ตัว
-	เซนเซอร์ตรวจวัดอุณหภูมิ (RTD)	22	ตัว
-	ลิมิตสวิตช์ (Limit Switch)	72	ตัว

เมื่อทราบรายการอุปกรณ์ในกระบวนการทั้งหมดจึงนำไปสร้าง I/O List เพื่อใช้ในการสร้าง Tag ของโปรแกรมควบคุม โดยใช้ PLC คือ ControlLogix 5575 และใช้ซอฟท์แวร์ Studio 5000 ในการสร้างและจัดการโปรแกรมควบคุม แต่เนื่องจากโครงการนี้เป็นส่วนของโครงการที่ยังอยู่ในขั้นตอนการพัฒนาและทดสอบจึงใช้ตัวควบคุมแบบจำลอง คือ ซอฟท์แวร์ RSLinx Emulate 5000 แทน PLC จริง ขั้นตอนการสร้างส่วนโปรแกรมควบคุมมีดังต่อไปนี้

3.3.1 สร้างโปรเจคสำหรับโปรแกรมควบคุม โดยการเปิดซอฟท์แวร์ Studio 5000 ขึ้นมา จากนั้นคลิกที่ New Project แล้วจึงเลือกตัวควบคุม RSLinx Emulate 5000 Controller > Emulator แล้วจึงตั้งชื่อโปรเจค ดังภาพที่ 3.10 และภาพที่ 3.11

ภาพที่ 3.10 การสร้างโปรเจคใหม่ของ Studio 5000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.11 การเลือกตัวควบคุมสำหรับโปรเจค

3.3.2 การสร้าง Tag สำหรับโปรแกรมควบคุมนั้นต้องสร้างตามคุณลักษณะของอุปกรณ์แต่ชนิดที่ต่างกัน โดย Tag ที่สร้างขึ้นบนซอฟแวร์ Studio 5000 นี้สามารถเลือก Data Type (ชนิดของ Tag) ได้ ซึ่ง Data Type แต่ละชนิดมี Tag ย่อยภายในเพื่อให้มีคุณลักษณะที่ตรงกับอุปกรณ์นั้น ๆ ดังตัวอย่าง Data Type ของมอเตอร์ ตามตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่าง Data Type ของมอเตอร์

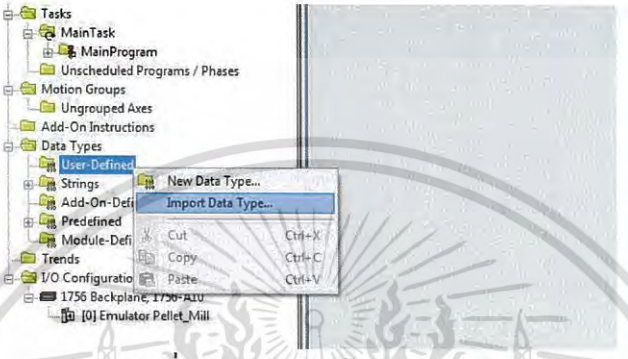
Tag ย่อย	Description
AM	Auto/Manual Mode
MT	Maintenance Mode
AR	Auto Request
MR	Manual Request
IL	Interlock
BIL	Bypass Interlock
RIL	Reset Interlock
ACK	Acknowledge
RUN	Input Run Feedback
OL	Input Overload
HF	Input Hardware Faulted
OUT	Output Run
AA	Alarm Motor Summary (Actual Alarm)
UA	Alarm Motor Summary (Unacknowledge Alarm)
AAST	Alarm Motor Fail Start (Actual Alarm)
UAST	Alarm Motor Fail Start (Unacknowledge Alarm)
AAOL	Alarm Motor Overload (Actual Alarm)
UAOL	larm Motor Overload (Unacknowledge Alarm)
AAHF	Alarm Motor Hardware Faulted (Actual Alarm)
UAHF	Alarm Motor Hardware Faulted (Unacknowledge Alarm)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

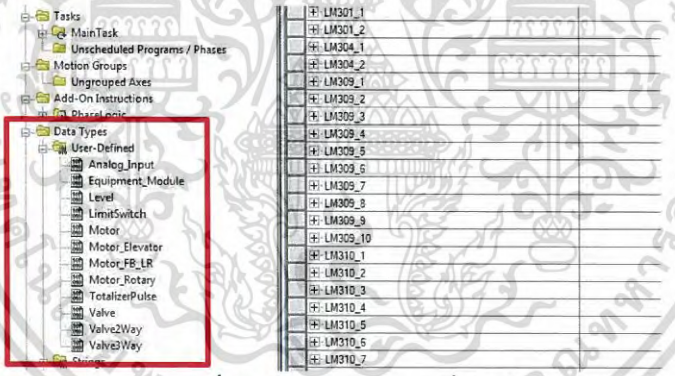
หอสมุดกลาง พระจอมเกล้าลาดกระบัง

สำหรับขั้นตอนการสร้าง Tag ให้กับอุปกรณ์มีดังต่อไปนี้

1) สร้าง Data Type หรือนำเข้า Data Type ก่อนการสร้าง Tag ให้กับอุปกรณ์แต่ละชนิด สำหรับขั้นตอนนี้เป็นการนำเข้า Data Type ที่สร้างขึ้นมาแล้ว โดยคลิกขวาที่ User-Defined > Import Data Type... ดังภาพที่ 3.12 แล้วเลือก Data Type ที่ต้องการนำเข้า จากนั้นคลิก Import แล้วคลิก OK ทำให้ได้ Data Type ของดังภาพที่ 3.13

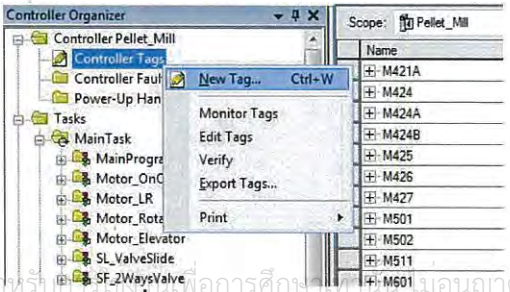


ภาพที่ 3.12 การนำเข้า Data Tpye



ภาพที่ 3.13 Data Tpye ที่นำเข้า

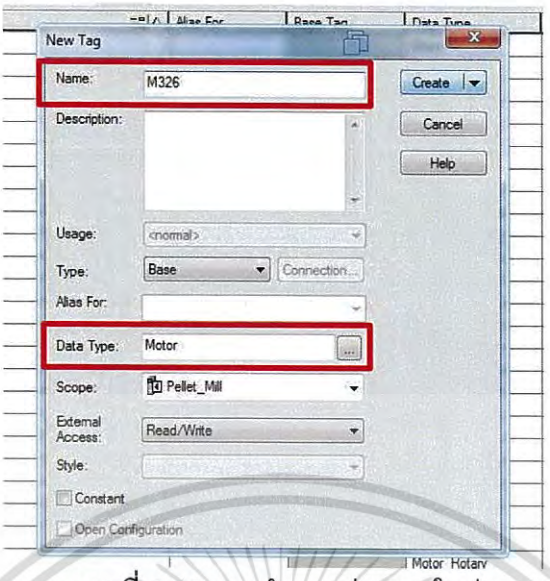
2) การสร้าง Tag ให้กับโปรแกรมควบคุมโดยคลิกขวาที่ Controller Tags > New Tag... แล้วจึงตั้งชื่อ Tag และเลือก Data Type ให้ตรงกับชนิดของอุปกรณ์ ดังภาพที่ 3.14 และภาพที่ 3.15 แล้วคลิก Create จึงได้ Tag ที่สร้างขึ้นใหม่ตามต้องการดังภาพที่ 3.16



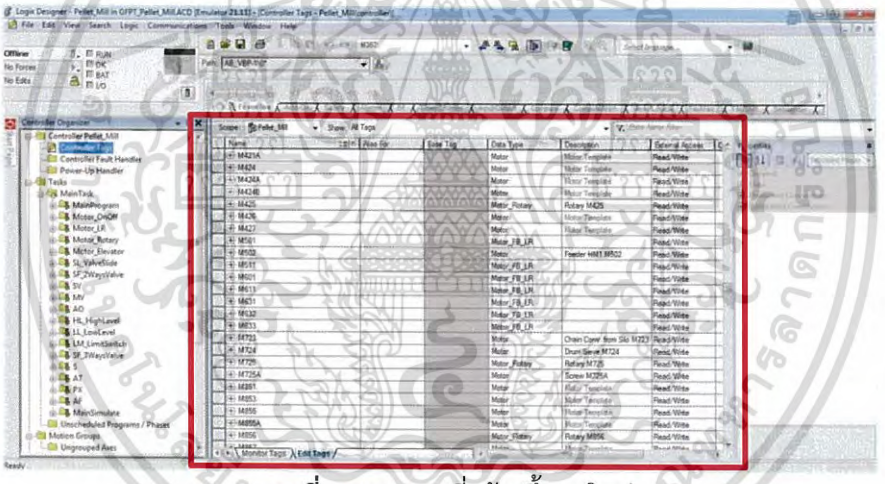
ภาพที่ 3.14 การสร้าง Tag ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

148522



ภาพที่ 3.15 การกำหนดค่า Tag ใหม่

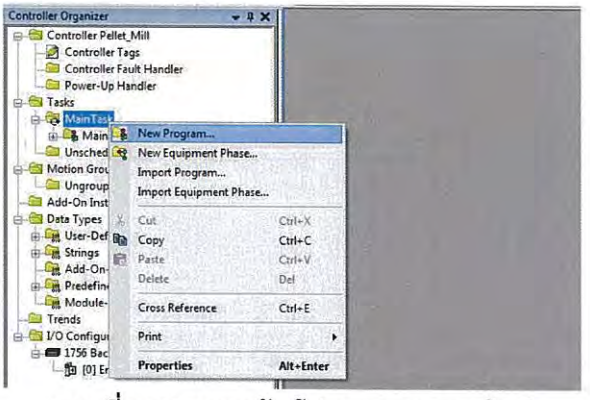


ภาพที่ 3.16 Tag ที่สร้างขึ้นมาใหม่

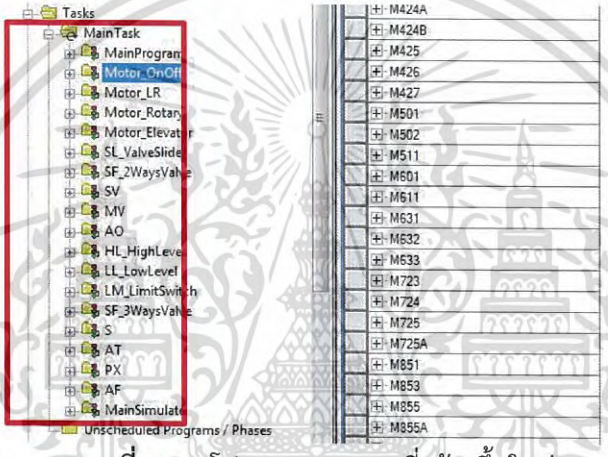
3.3.3 การสร้างโปรแกรมควบคุมอุปกรณ์ในกระบวนการ โดยอุปกรณ์ต่างชนิดกันจึงต้องมีโปรแกรมการควบคุมที่ต่างกันไป มีวิธีการดังนี้

1) สร้างโปรแกรมควบคุม เริ่มจากคลิกขวาที่ MainTask > New Program... แล้วจึงเปลี่ยนชื่อโปรแกรมตามชนิดของตัวอุปกรณ์ เช่น Motor_OnOff, Motor_LR, Motor_Elevator เป็นต้น ดังภาพที่ 3.17 และภาพที่ 3.18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

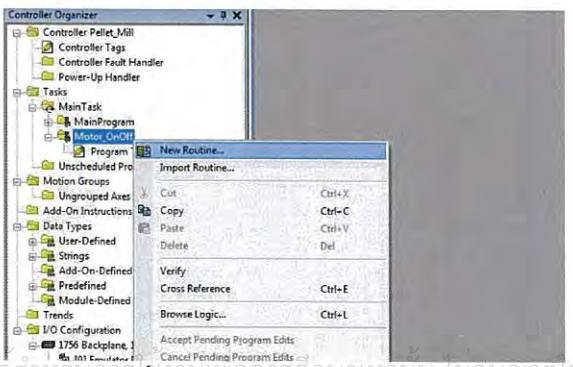


ภาพที่ 3.17 การสร้างโปรแกรมควบคุมใหม่



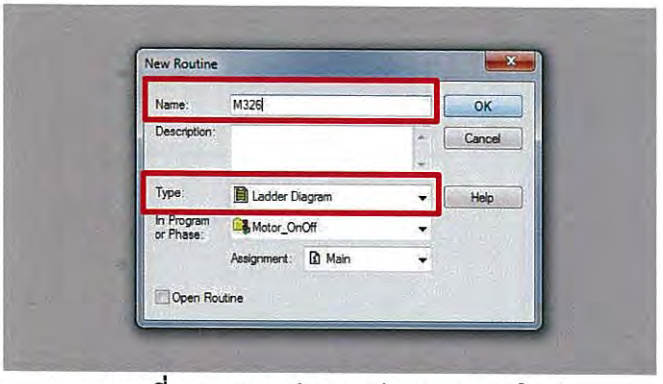
ภาพที่ 3.18 โปรแกรมควบคุมที่สร้างขึ้นใหม่

2) การสร้าง Routine สำหรับเขียนโปรแกรมที่ใช้สำหรับควบคุมอุปกรณ์ต่าง ๆ โดยภาษาที่ใช้ คือ Ladder Diagram โดยสร้าง Routine ของอุปกรณ์ให้ตรงกับชนิดของอุปกรณ์นั้น ๆ เช่น M326 เป็นมอเตอร์ธรรมดา (Motor_OnOff) เริ่มจากคลิกขวาที่ Motor_OnOff > New Routine... จากนั้นเปลี่ยนชื่อเป็น M326 โดยเลือก Type (ภาษาของโปรแกรม PLC) เป็น Ladder Diagram แล้วคลิก OK ดังภาพที่ 3.19 ถึงภาพที่ 3.21

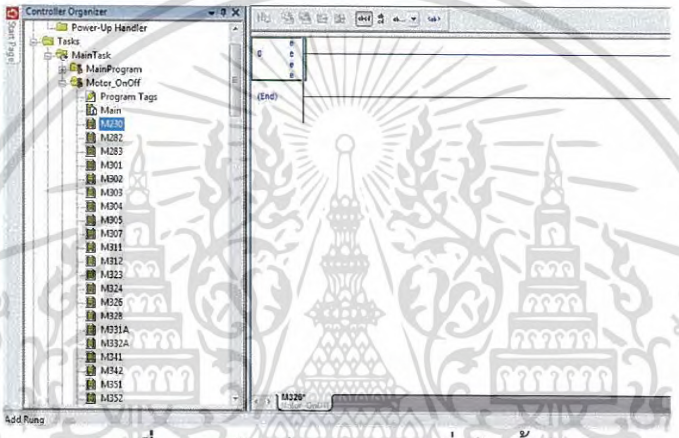


ภาพที่ 3.19 การสร้าง Routine ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

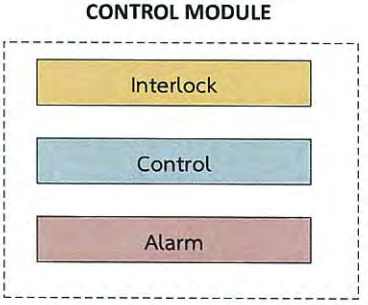


ภาพที่ 3.20 การกำหนดค่า Routine ใหม่



ภาพที่ 3.21 ตัวอย่าง Routine ที่สร้างขึ้นใหม่

เมื่อสร้าง Routine ให้กับอุปกรณ์ทุกตัวแล้วจึงเขียนโปรแกรมสำหรับตัวควบคุมอุปกรณ์ทุกตัวในกระบวนการ โดยรูปแบบการสร้างโปรแกรมควบคุมอุปกรณ์เป็นการนำข้อมูลพื้นฐานที่ใช้ควบคุมอุปกรณ์ เช่น อินพุต เอาท์พุต รูปแบบลักษณะการควบคุม มาสร้างเป็นฟังก์ชั่นการควบคุม ซึ่งเรียกว่า Control module ประกอบไปด้วย 3 ส่วนหลัก ๆ คือ Control, Alarm และ Interlock ดังตัวอย่างภาพที่ 3.22 และโปรแกรมควบคุมตัวอย่างในภาพที่ 3.23



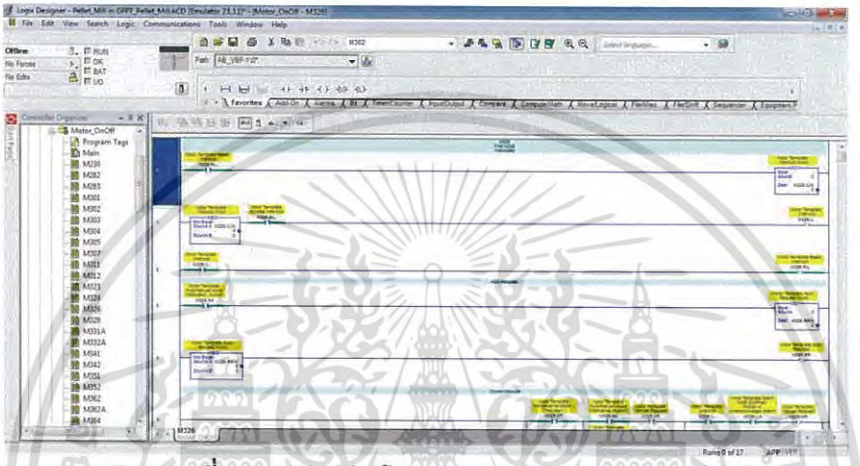
ภาพที่ 3.22 ส่วนประกอบหลักของ Control Module

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- Control คือ ส่วนควบคุมอุปกรณ์ เช่น การเลือกโหมดควบคุม Auto Manual การสั่ง Start Stop อุปกรณ์ การแสดงสถานะอุปกรณ์ เป็นต้น

- Alarm คือ ส่วนแจ้งเตือนเมื่ออุปกรณ์มีการทำงานที่ผิดพลาด เช่น ผู้ปฏิบัติงานสั่ง Open Valve แล้ว แต่อุปกรณ์จริงยังคงสถานะ Close จึงเกิดการแจ้งเตือนไปยังผู้ปฏิบัติงาน

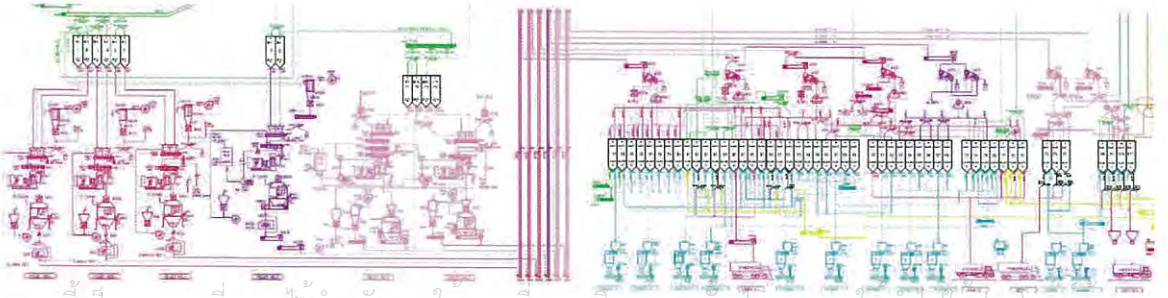
- Interlock คือ เงื่อนไข ข้อห้าม ในการควบคุมอุปกรณ์นั้น ๆ เช่น เมื่อวัตถุดิบในถังเก็บอยู่ในระดับ Low ต้องไม่สามารถทำสั่งงาน Start มอเตอร์สายพานนำวัตถุดิบได้



ภาพที่ 3.23 ตัวอย่างโปรแกรมควบคุมของอุปกรณ์

3.4 ส่วนแสดงผล

ในการสร้างส่วนแสดงผลเพื่อใช้ควบคุม และแสดงสถานะเปิด-ปิดของอุปกรณ์ต่าง ๆ ได้แก่ มอเตอร์สายพานลำเลียง มอเตอร์สองทิศทาง มอเตอร์แบบหมุนเลือกตำแหน่ง วาล์วสองทางเลือก วาล์วสามทางเลือก วาล์วสไลด์ และเซนเซอร์ต่าง ๆ เป็นต้น อีกทั้งยังใช้แสดงผลค่าของกระแส และความเร็วรอบของมอเตอร์ในกระบวนการอัดเม็ดอาหารสัตว์โดยใช้ซอฟท์แวร์ ArchestrA IDE .ในการจัดการส่วนแสดงผลต่าง ๆ ซึ่งเรียกระบบนี้ว่า System Platform และใช้ซอฟท์แวร์ Wonderware Intouch ในการสร้าง HMI ซึ่งซอฟท์แวร์ตัวนี้ถูกติดตั้งมาพร้อมกันกับซอฟท์แวร์ ArchestrA IDE โดย HMI ต้องถูกสร้างให้เหมาะสมกับการใช้งาน และใกล้เคียงวิธีควบคุมเดิมที่ใช้ Mimic Diagram Panel เป็นส่วนควบคุม ดังตัวอย่างภาพที่ 3.24 เพื่อให้ผู้ปฏิบัติงานคุ้นเคย และสามารถใช้งานระบบควบคุมใหม่นี้ได้อย่างรวดเร็ว อีกทั้งยังช่วยเพิ่มความสะดวก และลดความผิดพลาดในการปฏิบัติงาน



ภาพที่ 3.24 Flow Diagram ของกระบวนการอัดเม็ดอาหารสัตว์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

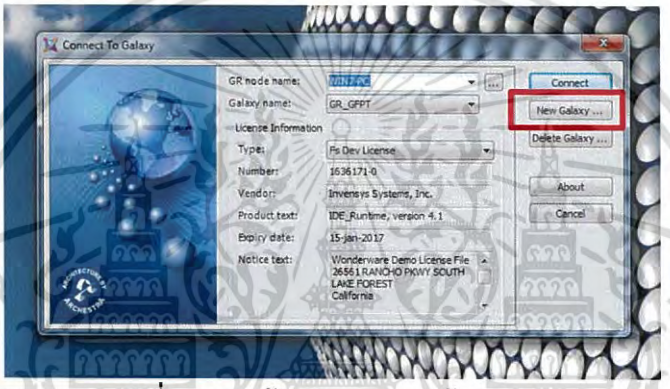
ในการสร้างส่วนแสดงผลที่ใช้สำหรับติดตามผล และควบคุมกระบวนการอัดเม็ดอาหารสัตว์ จำเป็นต้องออกแบบให้เหมือน หรือคล้ายคลึงกันกับส่วนควบคุมของระบบเดิม เพื่อให้ผู้ปฏิบัติงานใช้งานได้สะดวก และรวดเร็ว มีขั้นตอนดังต่อไปนี้

3.4.1 การสร้าง Galaxy ของระบบ

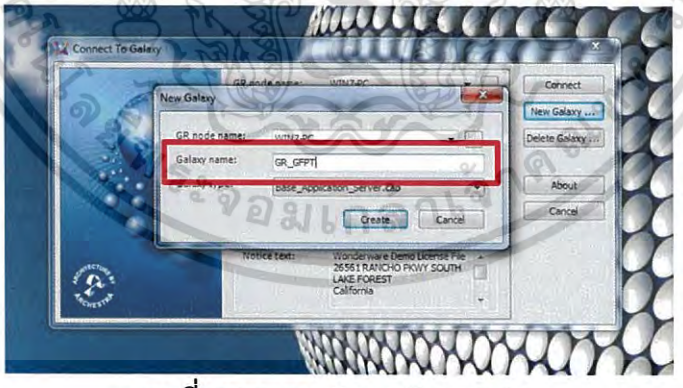
ในการสร้างโปรเจคขึ้นมาบนซอฟท์แวร์ ArchestrA IDE นั้นเรียกว่า Galaxy ซึ่งเป็นส่วนที่รวบรวมข้อมูลทั้งหมดไว้ ในการสร้าง Galaxy มีขั้นตอนดังนี้

1) เปิดซอฟท์แวร์ ArchestrA IDE จากนั้นคลิกที่ New Galaxy ดังภาพที่ 3.25

2) ป้อนชื่อ Galaxy ลงในช่อง Galaxy Name ดังภาพที่ 3.26 แล้วจึงคลิก Creat จากนั้นคลิก Connect จึงได้ Galaxy ใหม่ดังภาพที่ 3.27

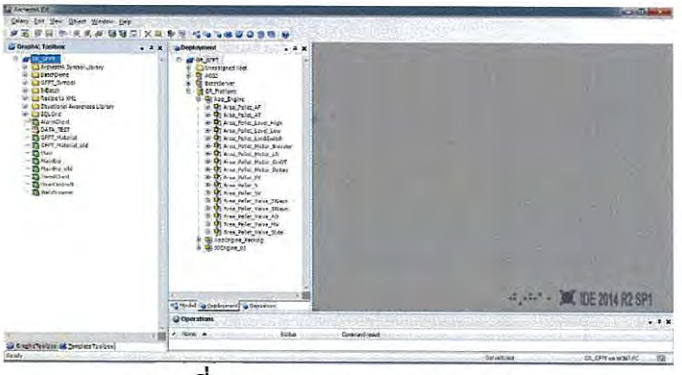


ภาพที่ 3.25 หน้าแรกของการสร้าง Galaxy



ภาพที่ 3.26 การ Creat New Galaxy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

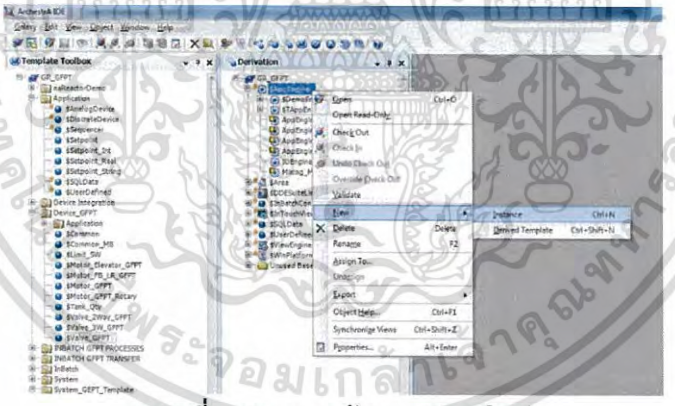


ภาพที่ 3.27 การ Connect Galaxy

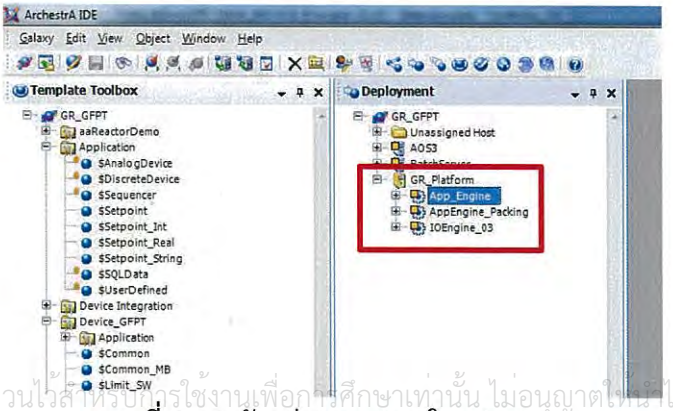
3.4.2 การสร้างส่วนประกอบอื่น ๆ

ในการสร้างส่วนแสดงผลโดยใช้ซอฟท์แวร์ Archestra IDE ขึ้นมานั้นมี Object ซึ่งเป็นส่วนสำคัญในการเก็บข้อมูลของอินพุต/เอาท์พุต จึงต้องสร้างส่วนที่ใช้ในการเก็บ Object ซึ่งมีลำดับดังนี้ Galaxy > Platform > Engine > Area > Object

1) สร้าง Engine โดยคลิกขวาที่ $AppEngine > New > Instance เพื่อสร้าง Engine ใหม่ จากนั้นจึงเปลี่ยนชื่อ Engine ตามต้องการ และลากเข้ามาไว้ใน Platform ที่ต้องการ (GR_Platform) ดังตัวอย่างภาพที่ 3.28 และภาพที่ 3.29



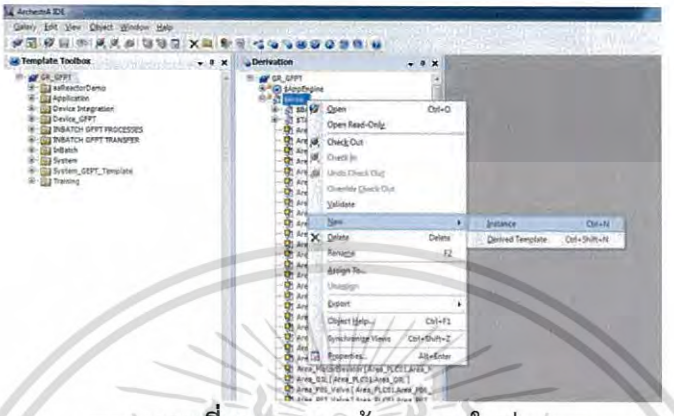
ภาพที่ 3.28 การสร้าง Engine ใหม่



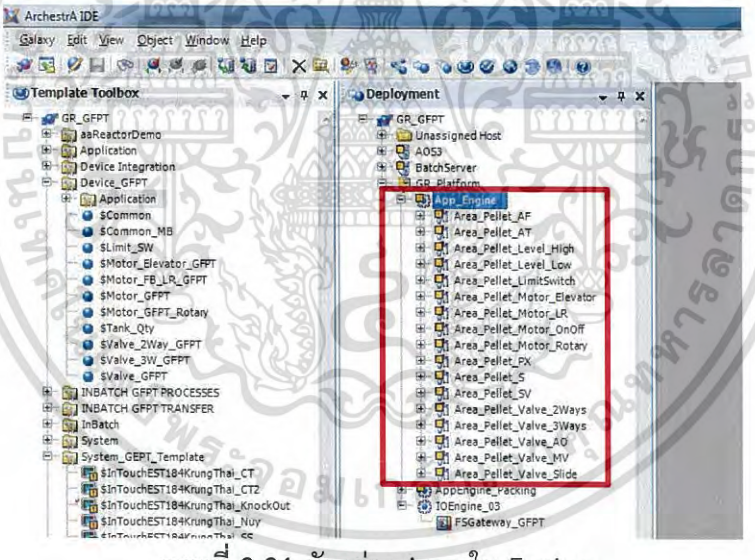
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 3.29 ตัวอย่าง Engine ใน Platform

2) การสร้าง Area โดยคลิกขวาที่ $Area > New > Instance เพื่อสร้าง Area ใหม่จากนั้นจึงเปลี่ยนชื่อ Area ตามที่ต้องการ และลากเข้ามาไว้ใน Engine ที่ต้องการ (App_Engine) ดังตัวอย่างภาพที่ 3.30 และภาพที่ 3.31 การตั้งชื่อ Area ควรให้สอดคล้องกับ Object ที่ต้องการนำมาเก็บ



ภาพที่ 3.30 การสร้าง Area ใหม่



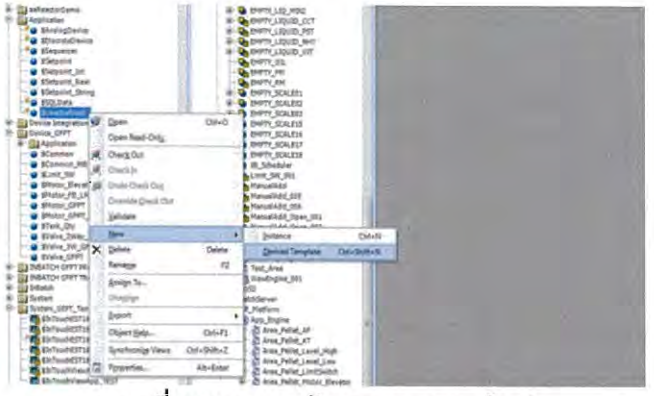
ภาพที่ 3.31 ตัวอย่าง Area ใน Engine

3.4.3 การสร้าง Template

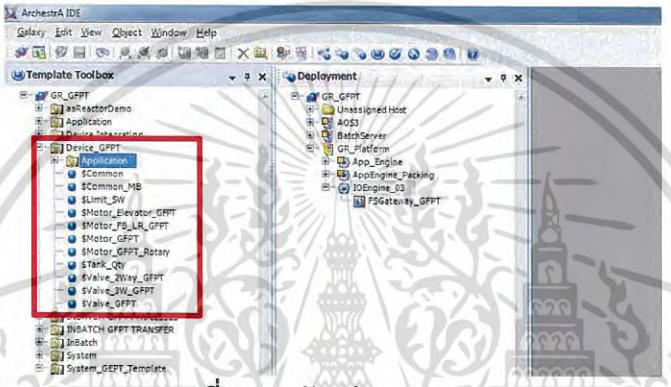
Template ของระบบเป็นเสมือนกับตัวต้นแบบของ Object อุปกรณ์ต่าง ๆ ที่มีคุณสมบัติเหมือนกันแต่มีจำนวนมาก เช่น มอเตอร์, วาล์วสองทางเลือก, ลิมิตสวิตช์ เป็นต้น ซึ่งภายใน Template ประกอบไปด้วยข้อมูลที่ใช้ในการควบคุม และติดตามผลบนส่วนแสดงผลของระบบ SCADA ซึ่งประกอบไปด้วย Attributes, Scripts, Graphics และข้อมูลอื่น ๆ โดยมีวิธีการสร้าง Template ดังนี้

1) การสร้าง Template ใหม่โดยคลิกขวาที่ $UserDefined > New > Derived Template จากนั้นเปลี่ยนชื่อตามที่ต้องการ เช่น Motor_GFPT, Valve_GFPT, Limit_SW เป็นต้น ดังภาพที่ 3.32 และภาพที่ 3.33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.32 การสร้าง Template ใหม่



ภาพที่ 3.33 ตัวอย่าง Template

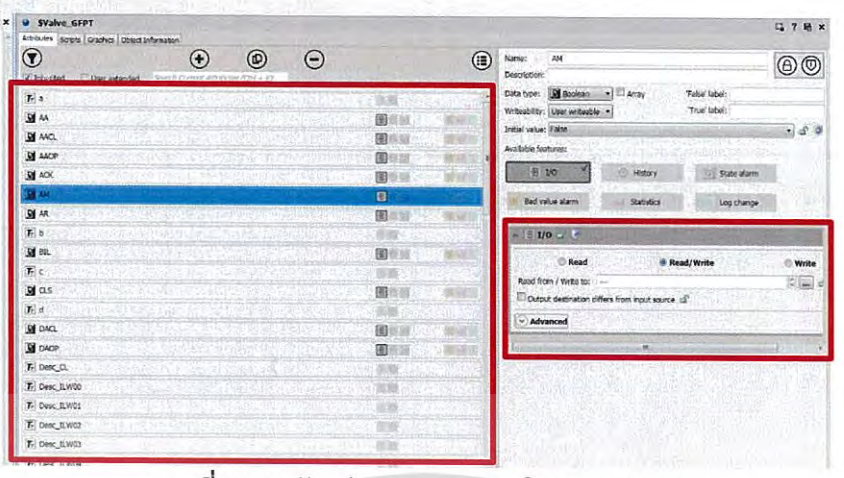
2) การสร้าง Attributes ใน Template โดยการคลิกเครื่องหมายบวก (+) เพื่อเพิ่ม Attributes จากนั้นจึงเลือกประเภท/หน้าที่ เพื่อใช้เป็นส่วนเก็บข้อมูลที่รับมาจาก PLC โดย Attribute สามารถสร้างได้ทั้งแบบ Manual ซึ่งสร้างทีละตัวตามความต้องการ หรือแบบ Auto จากการเขียน Script เพื่อให้สร้าง Attributes ได้ทีละหลาย ๆ ตัวตามต้องการ ซึ่งในครั้งนี้สร้างจาก Script และเลือกให้เป็นประเภท I/O เพื่อที่รับส่งค่าจาก PLC ดังภาพที่ 3.34 สำหรับการเลือกประเภทของ Attribute หากเลือกเป็น I/O จำเป็นต้องมีการกำหนด Input Source ให้กับ Attribute เพื่อให้สามารถรับส่งค่ากับ PLC ได้ โดยมีรูปแบบ ดังนี้ <DINetworkObject_Name>.<Topic_Name>.<Item/Register>

- DINetworkObject_Name คือ ชื่อของ DDESuiteLinkClient Object ที่สร้างในขั้นตอนที่ 3.4.11
- Topic_Name คือ ชื่อของ Topic ที่สร้างในขั้นตอนที่ 3.4.11
- Item/Register คือ ชื่อของ Tag ย่อยของอุปกรณ์ที่สร้างในขั้นตอนที่ 3.3.2 หรือ I/O ของอุปกรณ์

ตัวอย่างการกำหนด Input Source ของ Attribute

- FSGateway_GFPT.OPC_GFPT.P119.AM
- FSGateway_GFPT.OPC_GFPT.M858.AR
- FSGateway_GFPT.OPC_GFPT.M862C.MT
- FSGateway_GFPT.OPC_GFPT.M410.MR
- FSGateway_GFPT.OPC_GFPT.M309.AL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



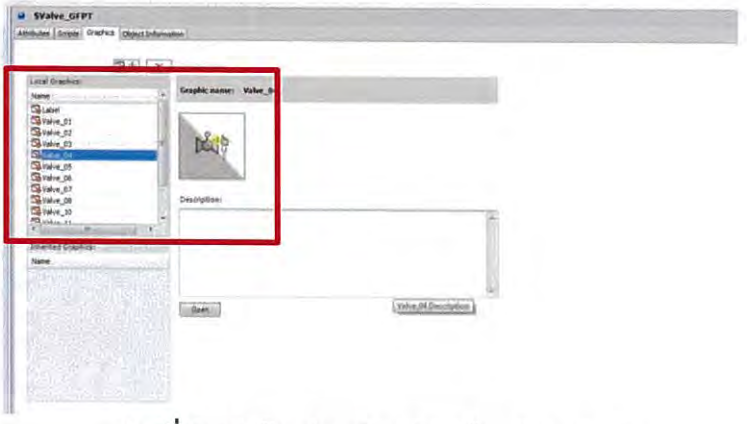
ภาพที่ 3.34 ตัวอย่าง Attributes ใน Template

3) การสร้าง Scripts ใน Template โดยการคลิกเครื่องหมายบวก (+) เพื่อสร้าง Scripts ใหม่ จากนั้นจึงเขียนคำสั่ง หรือโค้ด ดังภาพที่ 3.35 เพื่อให้ทำงานตามคำสั่ง หรือลำดับตามที่เขียนลงไปใน Scripts สามารถนำไปใช้สร้าง Attributes หรือเพื่อกำหนด I/O ในเบื้องต้นได้เช่นกัน

ภาพที่ 3.35 ตัวอย่าง Scripts ใน Template

4) การสร้าง Graphics ใน Template โดยคลิกที่เครื่องหมายบวก (+) จากนั้นจึงตั้งชื่อตามต้องการ ดังภาพที่ 3.36 ซึ่ง Graphics เป็นส่วนที่ใช้แทนตัวอุปกรณ์นั้น ๆ ของ Template โดยสามารถนำเข้าได้จากที่สร้างขึ้นมาแล้ว หรือสร้างขึ้นใหม่ก็ได้เช่นกัน

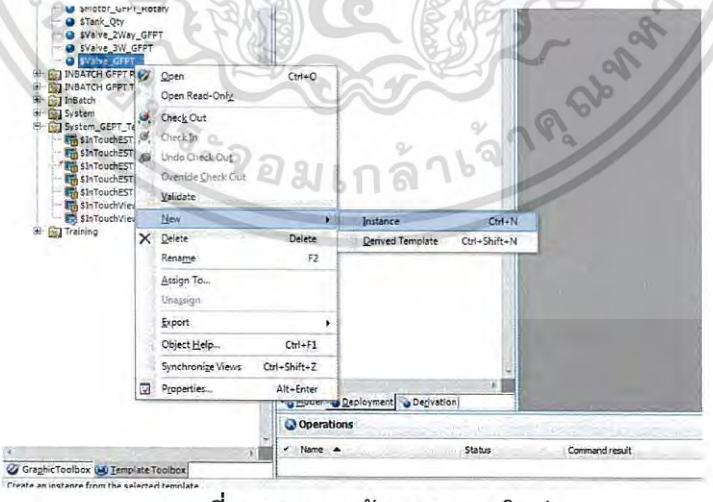
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.36 ตัวอย่าง Graphics ใน Template

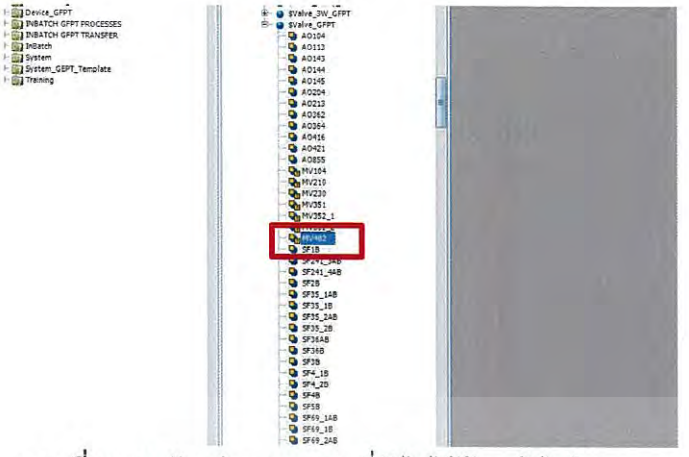
3.4.4 การสร้าง Object จาก Template

Object เสมือนกับตัวแทนส่วนที่เก็บข้อมูลของอุปกรณ์ตัวใดตัวหนึ่งในระบบควบคุม โดย Object ของอุปกรณ์แต่ละตัวสร้างจาก Template ชนิดเดียวกันกับประเภทของอุปกรณ์ตัวที่ต้องการสร้าง Object เช่น หากต้องการสร้าง Object ของ M326 ซึ่งเป็นมอเตอร์ชนิด On-Off ธรรมดา ต้องสร้างจาก Template Motor_GFPT ซึ่งเป็น Template ของมอเตอร์ชนิด On-Off ธรรมดา โดยข้อมูลเบื้องต้นของ Object เช่น Attributes, Scripts, Graphics นั้นเหมือนกับ Template ที่ใช้สร้าง Object ส่วนข้อมูลเฉพาะของ Object นั้นสามารถเพิ่มได้ในภายหลัง วิธีการสร้าง Object มีดังนี้ คลิกขวาที่ Template ที่ต้องการ เช่น $Valve_GFPT > New > Instance จากนั้นเปลี่ยนชื่อ Object ให้เป็นไปตามอุปกรณ์ที่ต้องการ แล้วจึงย้าย Object ไปที่ Area ที่ต้องการ ดังภาพที่ 3.37 ถึงภาพที่ 3.39

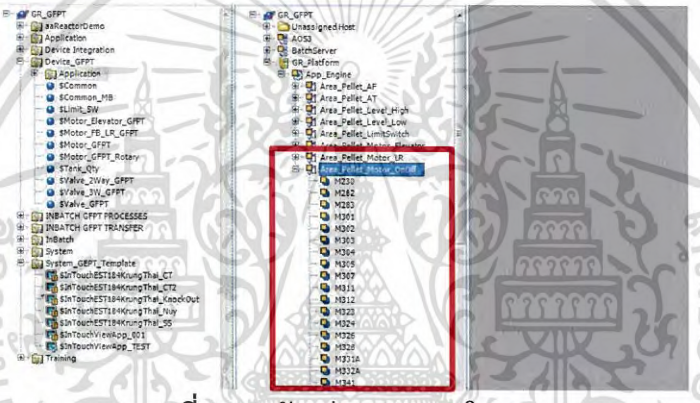


ภาพที่ 3.37 การสร้าง Object ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.38 ตัวอย่าง Object ที่ยังไม่ได้ย้ายเข้าไปใน Area

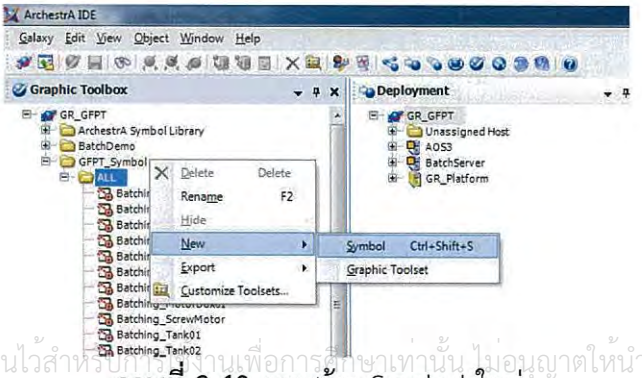


ภาพที่ 3.39 ตัวอย่าง Object ใน Area

3.4.5 การสร้าง Symbol

Symbol เป็นกราฟฟิกที่ใช้สำหรับทั้งสร้าง Graphics ใน Template และสร้างส่วนแสดงผลในซอฟท์แวร์ Wonderware InTouch มีวิธีการสร้างดังนี้

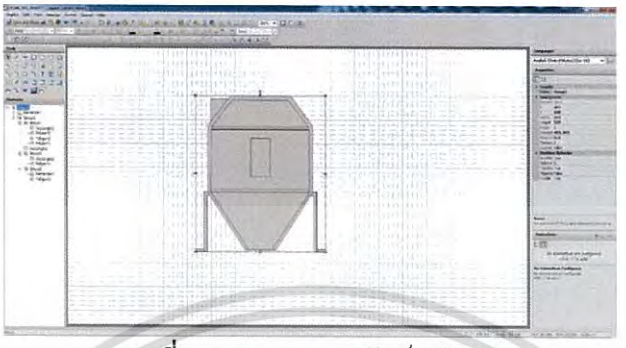
1) การสร้าง Symbol ใหม่โดยคลิกขวาที่แทป Graphic Toolbox > New > Symbol ดังภาพที่ 3.40 จากนั้นเปลี่ยนชื่อตามต้องการ แล้วจึงเปิด Symbol ที่สร้างขึ้นมาเพื่อสร้างกราฟฟิก



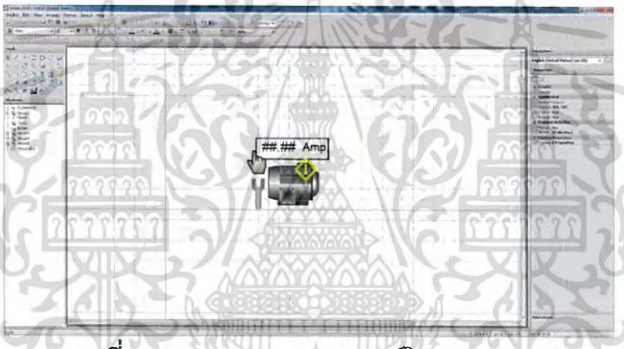
ภาพที่ 3.40 การสร้าง Symbol ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) การวาด Symbol ตามต้องการ โดยอาจวาด Symbol เพื่อใช้แทนอุปกรณ์ต่าง ๆ เช่น ถัง สายพานลำเลียง ท่อ เป็นต้น ดังภาพที่ 3.41 หรืออาจใช้วาดอุปกรณ์ต่าง ๆ เพื่อนำไปใช้สำหรับ Graphics ใน Template ได้เช่นเดียวกัน ดังภาพที่ 3.42

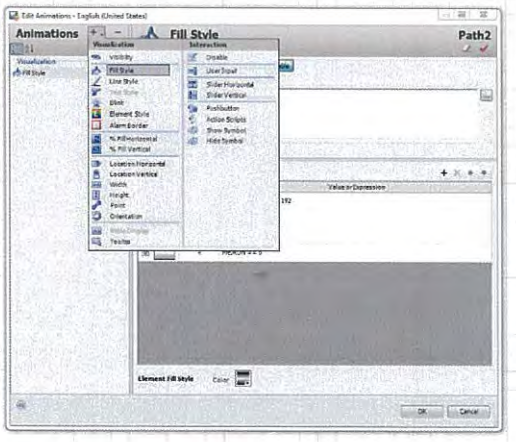


ภาพที่ 3.41 Symbol ถังเก็บวัตถุดิบ



ภาพที่ 3.42 Graphics มอเตอร์ใน Template

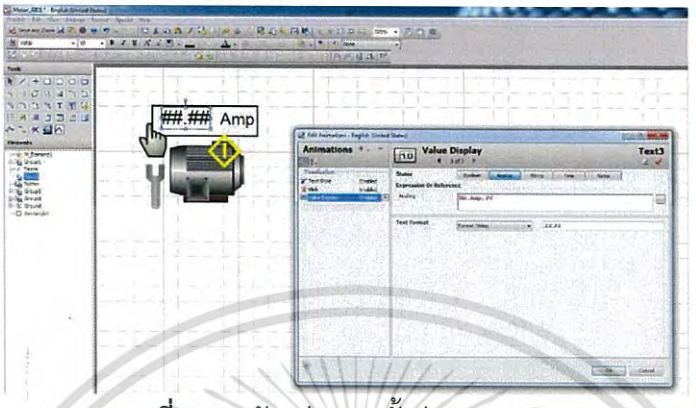
3) การสร้าง Animations สำหรับ Graphics ใน Template เพื่อใช้แทนอุปกรณ์ในกระบวนการ ซึ่ง Animations มีหลากหลายประเภทให้เลือกใช้ เช่น Visibility, Fill Style, Line Style, Blink, User Input และ Action Scripts เป็นต้น ดังภาพที่ 3.43



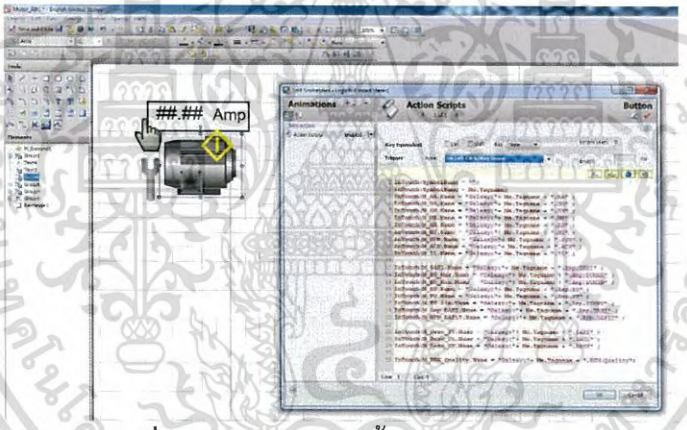
ภาพที่ 3.43 Animations สำหรับ Graphics

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

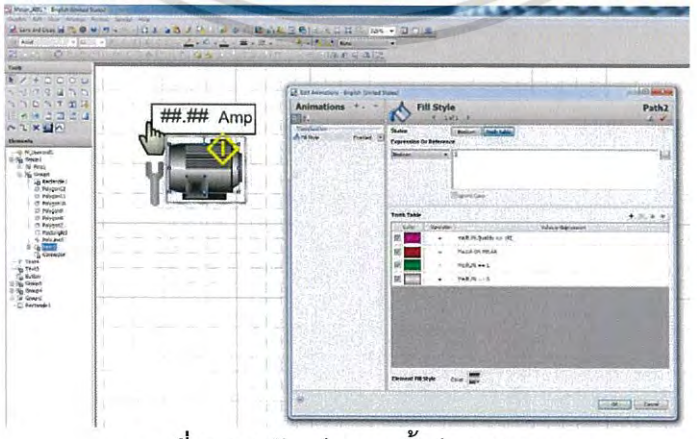
ตัวอย่างการตั้งค่าการใช้งาน Animations สำหรับ Graphics ดังภาพที่ 3.44 ถึงภาพที่ 3.46



ภาพที่ 3.44 ตัวอย่างการตั้งค่า Value Data



ภาพที่ 3.45 ตัวอย่างการตั้งค่า Action Script



ภาพที่ 3.46 ตัวอย่างการตั้งค่า Fill Style

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Graphics ที่สร้างขึ้นมีคุณลักษณะต่างกันไปเพื่อแสดงถึงสถานะการทำงาน เช่น Start, Stop, Alarm และ Connection Loss ดังตารางที่ 3.2

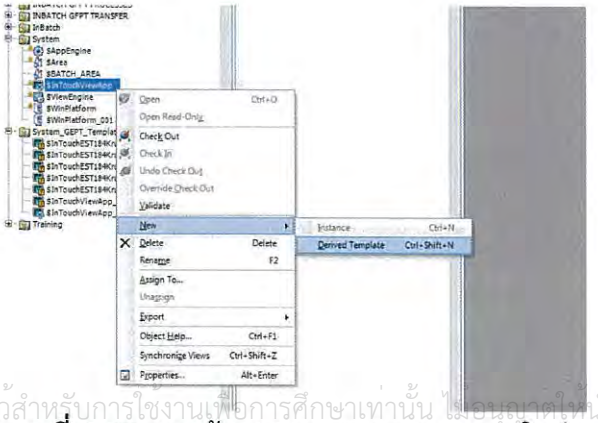
ตารางที่ 3.2 การใช้สีในการกำหนดสถานะอุปกรณ์

Color	Device Status	Blink
	STOP	No
	START	No
	ALARM	Yes
	CONNECTION LOSS	No

3.4.6 การสร้าง HMI ด้วยซอฟท์แวร์ Wonderware InTouch

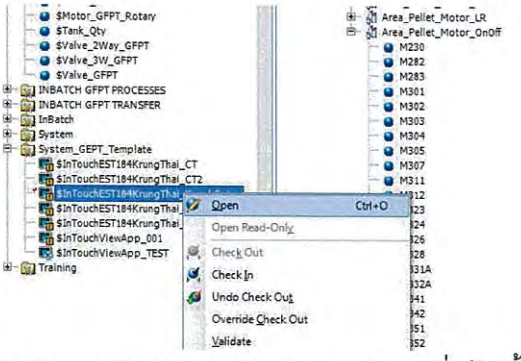
สำหรับส่วนแสดงผล หรือ HMI นั้นสร้างโดยซอฟท์แวร์ Wonderware InTouch ซึ่งถือเป็นซอฟท์แวร์ที่ใช้ในการสร้างกราฟฟิกสำหรับอุตสาหกรรมที่เป็นที่นิยมตัวหนึ่ง โดยซอฟท์แวร์ตัวนี้ถูกติดตั้งมาพร้อมกันกับซอฟท์แวร์ ArchestrA IDE ที่ใช้ในการจัดการข้อมูล Object ของส่วนแสดงผลนั่นเอง มีวิธีการสร้างดังนี้

1) การสร้างกราฟฟิก InTouch Application ใหม่โดยคลิกขวาที่ $InTouchViewApp > New > Derived Template แล้วเปลี่ยนชื่อตามต้องการ จากนั้นเปิด InTouch Application ที่สร้างขึ้นใหม่เพื่อแก้ไขหน้า HMI ตามต้องการโดยคลิกขวาที่ InTouch Application ที่สร้างขึ้นมาใหม่ จากนั้นคลิกที่ Open ดังภาพที่ 3.47 และ 3.48



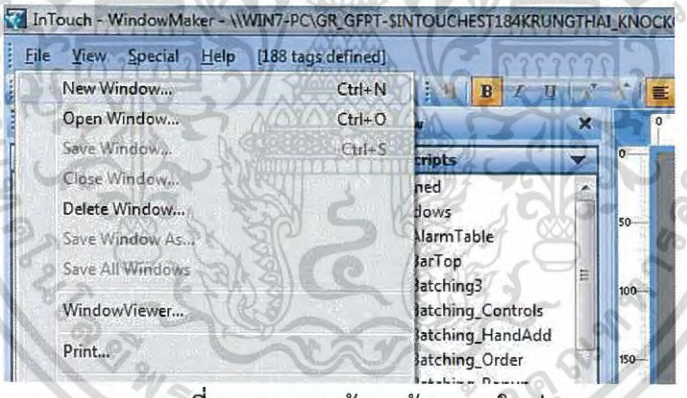
ภาพที่ 3.47 การสร้าง InTouch Application ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

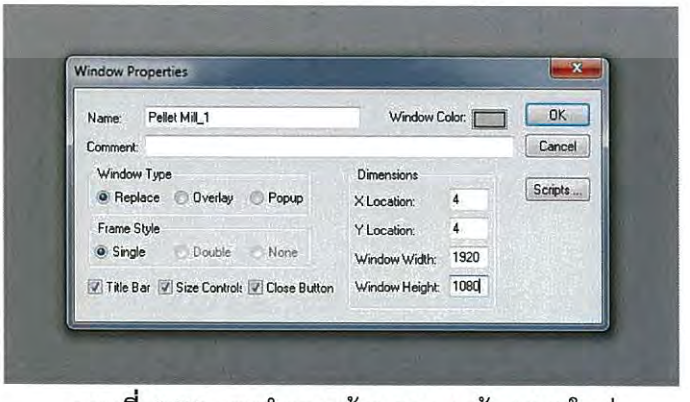


ภาพที่ 3.48 การเปิด InTouch Application ที่สร้างขึ้นใหม่

2) การสร้างหน้า HMI ใหม่ในซอฟท์แวร์ Wonderware InTouch สามารถทำได้ทั้งสร้างหน้า HMI ใหม่ โดยคลิกขวาที่ File > New Window... โดยสามารถตั้งชื่อ, เขียนคำอธิบาย, กำหนดขนาดของหน้า HMI ได้ดังภาพที่ 3.49 และ 3.50 หรือ Import เข้ามาก็ได้เช่นกันโดยคลิกขวาที่ File > Import... > Windows and Scripts ... ดังภาพที่ 3.51 แต่หากเคยสร้างไว้แล้วก็ไม่จำเป็นต้องสร้างขึ้นมาใหม่สามารถแก้ไขได้เลย ดังภาพที่ 3.52

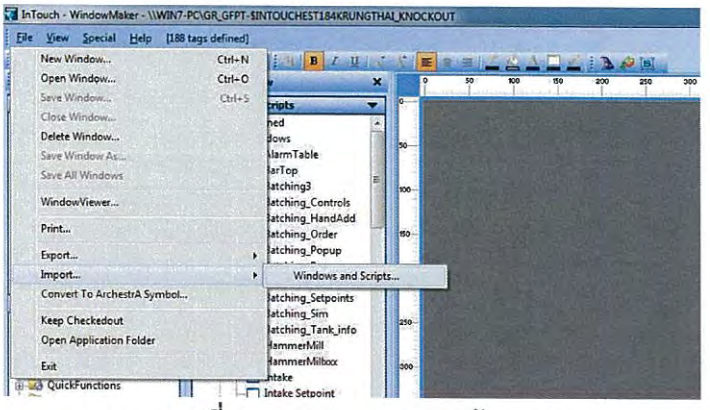


ภาพที่ 3.49 การสร้างหน้า HMI ใหม่



ภาพที่ 3.50 การกำหนดข้อมูลของหน้า HMI ใหม่

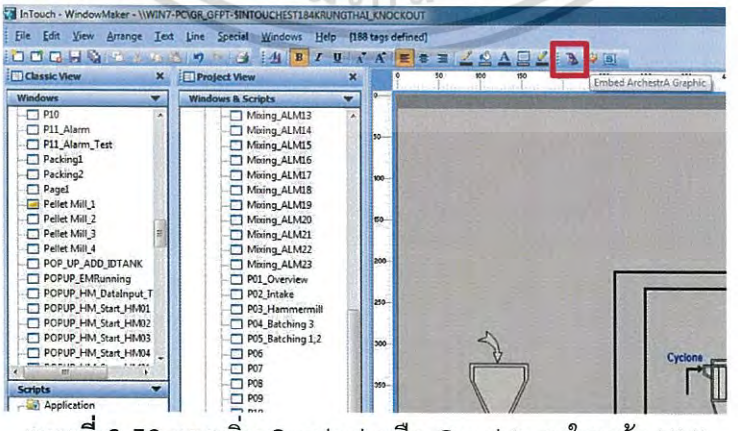
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.51 การ Import หน้า HMI

ภาพที่ 3.52 การเปิดหน้า HMI ที่เคยสร้างไว้

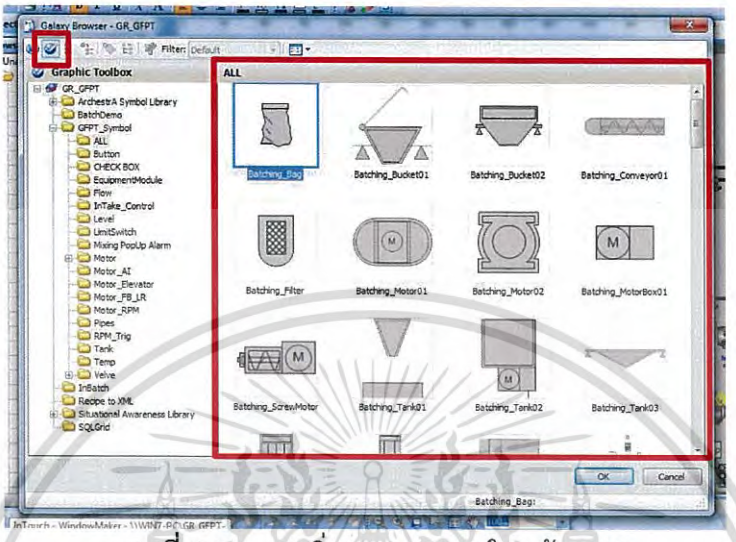
3) การนำ Symbol หรือ Graphic ที่เคยสร้างไว้มาใช้ในหน้า HMI โดยคลิกที่ปุ่ม Embed ArchestrA Graphic ดังภาพที่ 3.53



ภาพที่ 3.53 การเพิ่ม Symbol หรือ Graphic ลงในหน้า HMI

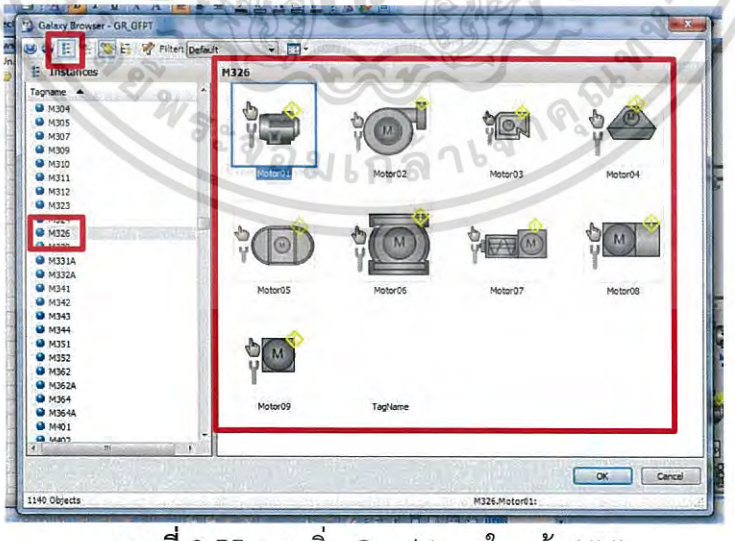
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หากต้องการเพิ่ม Symbol ลงในหน้า HMI ให้คลิกที่ปุ่ม Graphic Toolbox แล้วเลือก Symbol ที่เคยสร้างไว้ก่อนหน้านี้ไปเพิ่มลงในหน้า HMI ได้ดังภาพที่ 3.54



ภาพที่ 3.54 การเพิ่ม Symbol ลงในหน้า HMI

แต่หากต้องการเพิ่ม Graphic ของ Object ลงในหน้า HMI เพื่อแทนอุปกรณ์ใด ๆ ให้คลิกที่ปุ่ม Instance แล้วเลือก Object ที่ต้องการนำ Graphics เพิ่มลงในหน้า HMI แล้วจึงเลือก Graphic ใน Object นั้น ๆ ดังภาพที่ 3.55

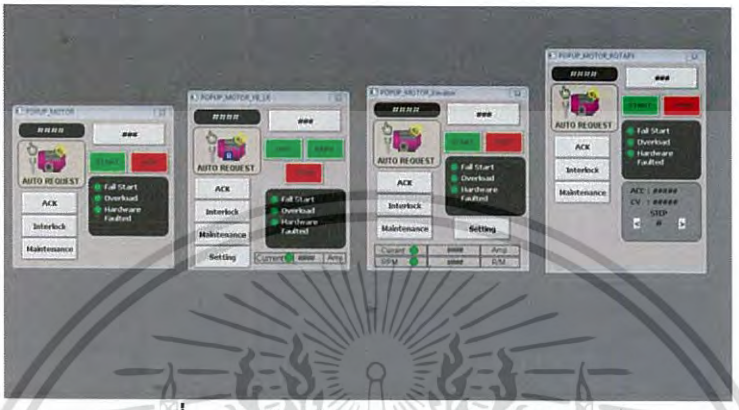


ภาพที่ 3.55 การเพิ่ม Graphic ลงในหน้า HMI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

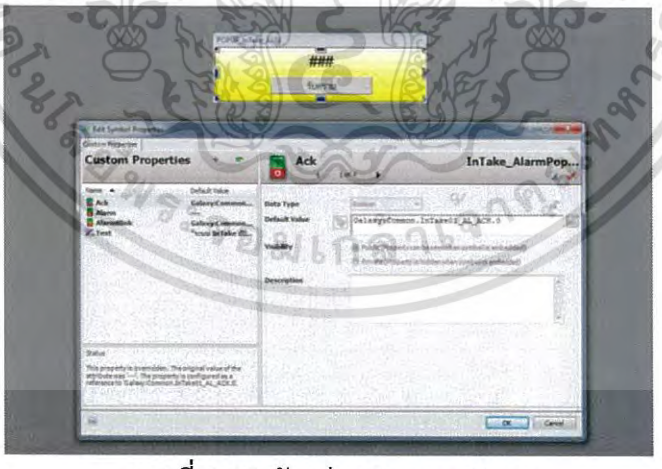
3.4.7 การสร้าง Popup สำหรับอุปกรณ์ต่าง ๆ และ Popup Alarm โดยในการสร้าง Popup มีวิธีการสร้างเช่นเดียวกับสร้างหน้า HMI เพียงแต่ปรับขนาดให้มีขนาดเล็กลง

1) Popup สำหรับอุปกรณ์ต่าง ๆ มีไว้สำหรับควบคุมการเปิด-ปิดอุปกรณ์ ตรวจดูสถานะของอุปกรณ์ (Fail, Fault, Overload) และกำหนดค่าพารามิเตอร์ต่าง ๆ ดังตัวอย่างภาพที่ 3.56



ภาพที่ 3.56 ตัวอย่าง Popup สำหรับอุปกรณ์

2) Popup Alarm มีไว้สำหรับแจ้งเตือนเมื่อเกิดเหตุการณ์ที่ผิดปกติเพื่อให้ผู้ปฏิบัติงานได้ทราบ ดังตัวอย่างภาพที่ 3.57



ภาพที่ 3.57 ตัวอย่าง Popup Alarm

การเกิด Alarm ของระบบ SCADA นี้มีหลักในการ คือ เมื่อค่าของตัวแปรใด ๆ ในการผลิต เช่น ค่าอุณหภูมิของอาหาร ค่ากระแสไฟฟ้าของพัดลม Cooler มีค่าตรงกับเงื่อนไขการเกิด Alarm ดังตารางที่ 3.4 โดยเงื่อนไขเหล่านี้มีค่า Setpoint ดังตารางที่ 3.3 ที่ผู้ปฏิบัติงานต้องกำหนดเองเพื่อควบคุมการผลิตให้เป็นไปตามเป้าหมาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.3 Setpoint ในการตรวจสอบ Alarm

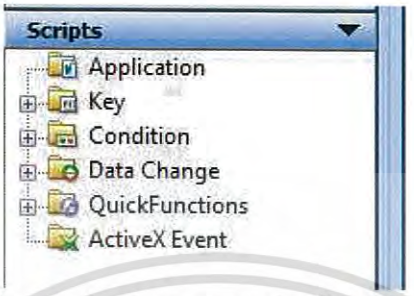
Setpoint	คำอธิบาย	Unit
SP1	ค่าอุณหภูมิต่ำสุดที่ยอมรับได้ของอาหารภายใน Conditioner	°C
SP2	ค่ากระแสไฟฟ้าสูงสุดระดับที่ 1 (Over Current 1)	Amp
SP3	ค่ากระแสไฟฟ้าสูงสุดระดับที่ 2 (Over Current 2)	Amp
SP4	ค่ากระแสไฟฟ้าต่ำสุดระดับที่ 1 (Under Current 1)	Amp
SP5	ค่ากระแสไฟฟ้าต่ำสุดระดับที่ 2 (Under Current 1)	Amp
SP6	ค่าที่ยอมรับได้ระหว่างผลต่างของอุณหภูมิอาหารภายใน Cooler เทียบกับอุณหภูมิอากาศภายนอก (ค่าเฝ้าระวัง)	°C
SP7	ค่าที่ยอมรับได้ระหว่างผลต่างของอุณหภูมิอาหารภายใน Cooler เทียบกับอุณหภูมิอากาศภายนอก (ค่าวิกฤต)	°C
SP8	ค่ากระแสไฟฟ้าต่ำสุด (Under Current) ของพัดลม Cooler	Amp

ตารางที่ 3.4 เงื่อนไขในการเกิด Alarm

เงื่อนไขในการเกิด Alarm	การแก้ไขปัญหา
เมื่ออุณหภูมิของอาหารภายใน Conditioner ต่ำกว่าค่าที่กำหนดไว้ใน SP1	เมื่ออุณหภูมิของอาหารภายใน Conditioner เพิ่มขึ้นจนมีค่าสูงกว่า SP1 Popup จึงหายไป
เมื่อค่ากระแสไฟฟ้าของเครื่องอัดเม็ดสูงจนถึงค่า SP2 (Overcurrent 1) แต่ยังไม่ถึงค่า SP3 (Overcurrent 2)	เมื่อกระแสไฟฟ้าของเครื่องอัดเม็ดกลับมาต่ำกว่า SP2 Popup จึงหายไป
เมื่อค่ากระแสไฟฟ้าของเครื่องอัดเม็ดต่ำกว่าค่า SP4 (Undercurrent 1) แต่ยังไม่ต่ำกว่า SP5 (Undercurrent 2)	เมื่อกระแสไฟฟ้าของเครื่องอัดเม็ดกลับมาสูงกว่า SP4 Popup จึงหายไป
เมื่อผลต่างระหว่างอุณหภูมิของอาหารภายใน Cooler เทียบกับอุณหภูมิอากาศภายนอก มีค่าเบี่ยงเบนมากกว่า SP6 (ค่าเฝ้าระวัง) แต่ยังไม่ถึง SP7 (ค่าวิกฤต)	เมื่อ เมื่อผลต่างระหว่างอุณหภูมิของอาหารภายใน Cooler เทียบกับอุณหภูมิอากาศภายนอก มีค่าเบี่ยงเบนไม่เกิน SP6 Popup จึงหายไป
เมื่อค่ากระแสไฟฟ้าของพัดลม Cooler มีค่าต่ำกว่าที่กำหนดใน SP8	เมื่อค่ากระแสไฟฟ้าของพัดลม Cooler กลับมามีค่าสูงกว่า SP8 Popup จึงหายไป

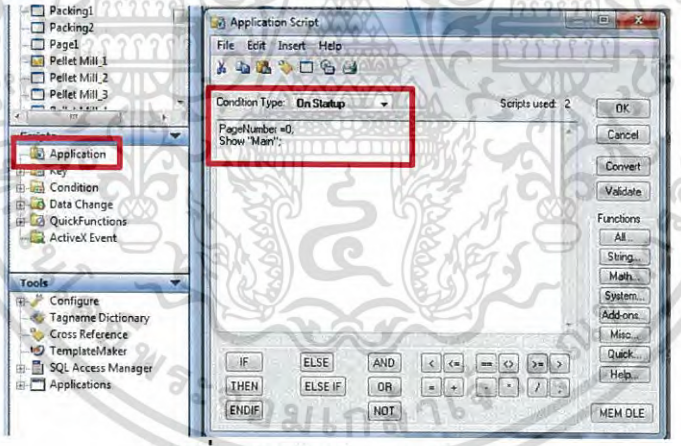
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.8 การสร้างเงื่อนไขต่าง ๆ ให้กับหน้า HMI ซึ่งเรียกว่า Scripts เช่น Application, Key, Condition, Data Change, Quick Functios ดังตัวอย่างภาพที่ 3.58 ซึ่งช่วยให้หน้า HMI ทำงานได้อย่างมีประสิทธิภาพ และเพิ่มความสะดวกให้แก่ผู้ปฏิบัติงาน



ภาพที่ 3.58 ตัวอย่าง Scripts สำหรับหน้าจอ HMI

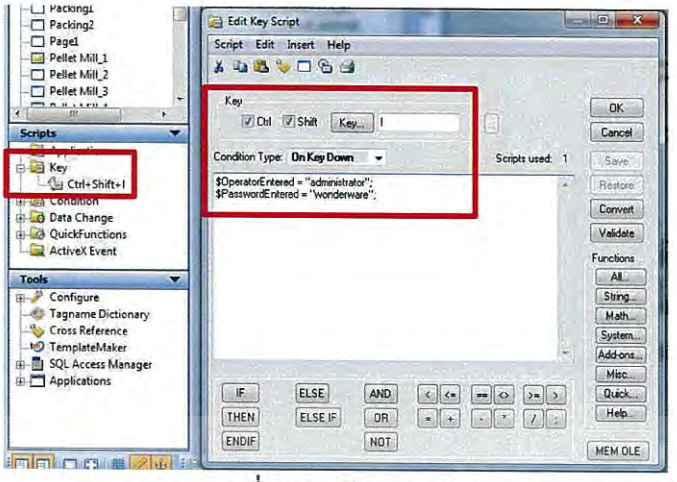
1) Application เป็นการเขียนคำสั่งเพื่อให้ทำงานตามคำสั่ง ในช่วงเวลาที่ต้องการ ดังภาพที่ 3.59 มีคำสั่งให้แสดงหน้า Main เมื่อเริ่มต้นการทำงาน



ภาพที่ 3.59 ตัวอย่าง Application

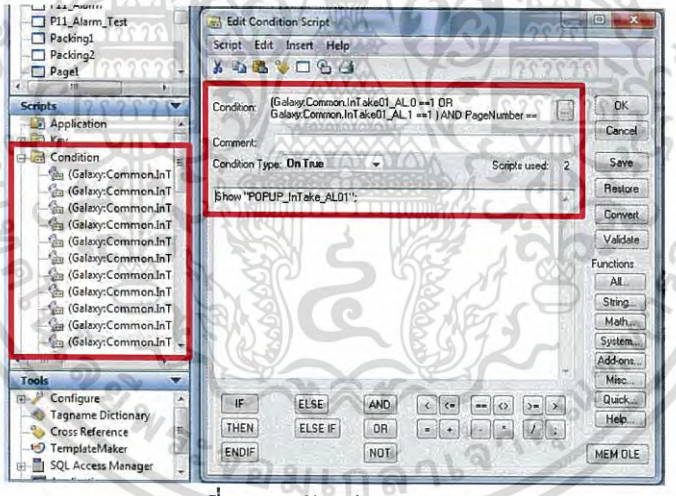
2) Key เป็นการเขียนคำสั่งลัดให้ผู้ใช้งานสามารถกดคีย์ลัดเพื่อความสะดวก และรวดเร็วในการปฏิบัติงาน ดังตัวอย่างภาพที่ 3.60 เป็นคีย์ลัดในการล๊อกอิน โดยเมื่อกดคีย์ลัด Ctrl+Shift+l พร้อมกัน ทำให้ค่าพารามิเตอร์ $OperatorEntered มีค่าเป็น administrator และ $PasswordEntered มีค่าเป็น wonderware

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.60 ตัวอย่าง Key

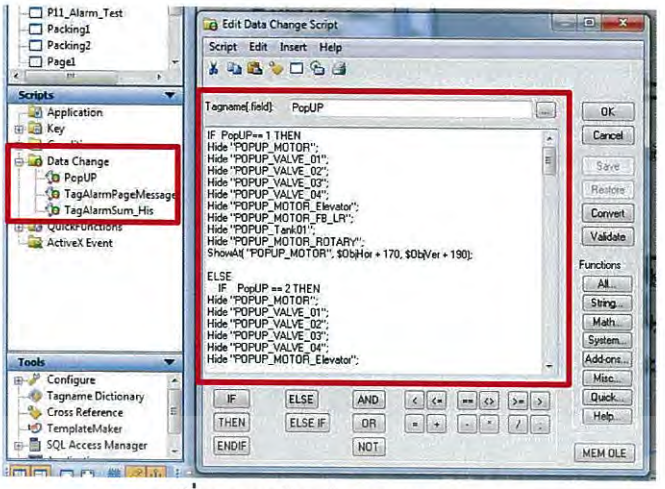
3) Condition เป็นการเช็คเงื่อนไข โดยที่เมื่อเงื่อนไขของหน้า HMI ตรงกันกับเงื่อนไขที่กำหนดไว้ ต้องทำงานตามคำสั่งที่เขียนไว้ ดังตัวอย่างภาพที่ 3.61



ภาพที่ 3.61 ตัวอย่าง Condition

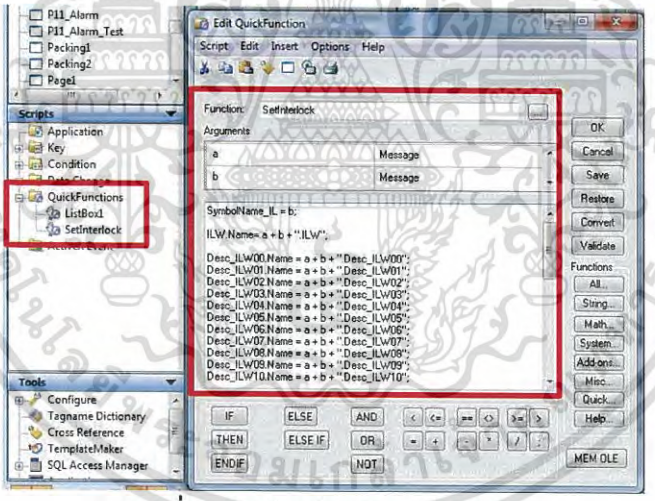
4) Data Change เป็นการเช็คค่าของพารามิเตอร์ใดพารามิเตอร์หนึ่ง แล้วจึงนำไปเทียบกับที่เขียนไว้ จากนั้นจึงทำงานตามคำสั่งที่เขียนไว้ ดังตัวอย่างภาพที่ 3.62 เมื่อค่าของ พารามิเตอร์ PopUP มีค่าเท่ากับ “1” ทำให้ซ่อนหน้า Popup อื่น ๆ ไว้แล้วแสดงเพียงแต่ Popup Motor เพียงหน้าเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.62 ตัวอย่าง Data Change

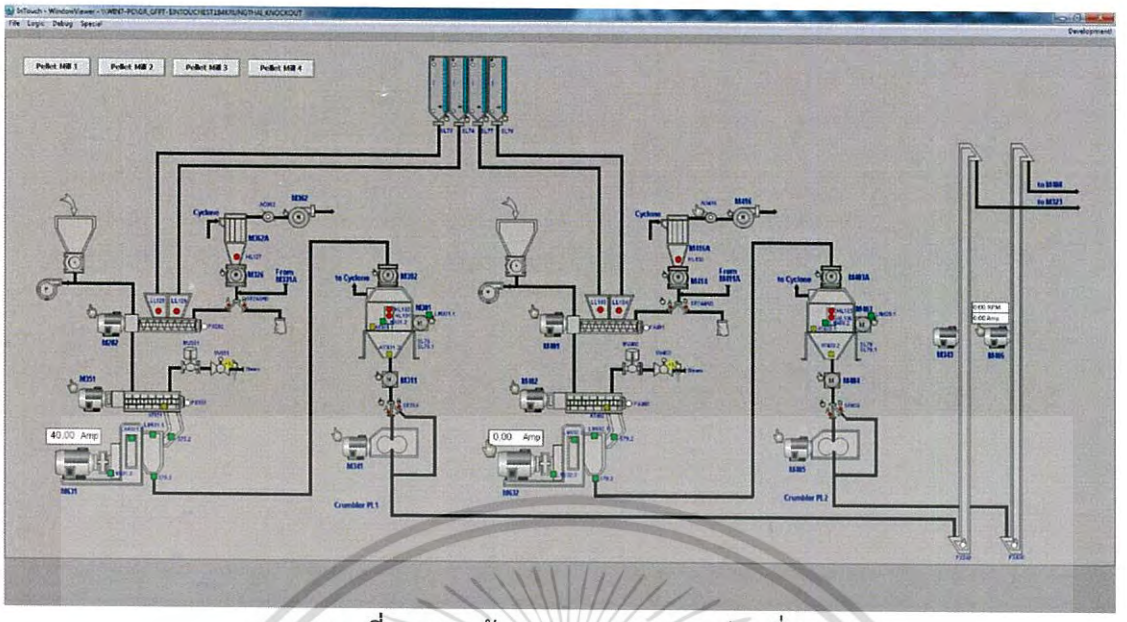
5) QuickFunctions เป็นการเขียนคำสั่ง หรือเงื่อนไขซึ่งสามารถเรียก Script จากที่อื่นและ Animation Link มาใช้งานได้ ซึ่งช่วยให้ลดการเขียนคำสั่งลงได้ ดังตัวอย่างภาพที่ 3.63



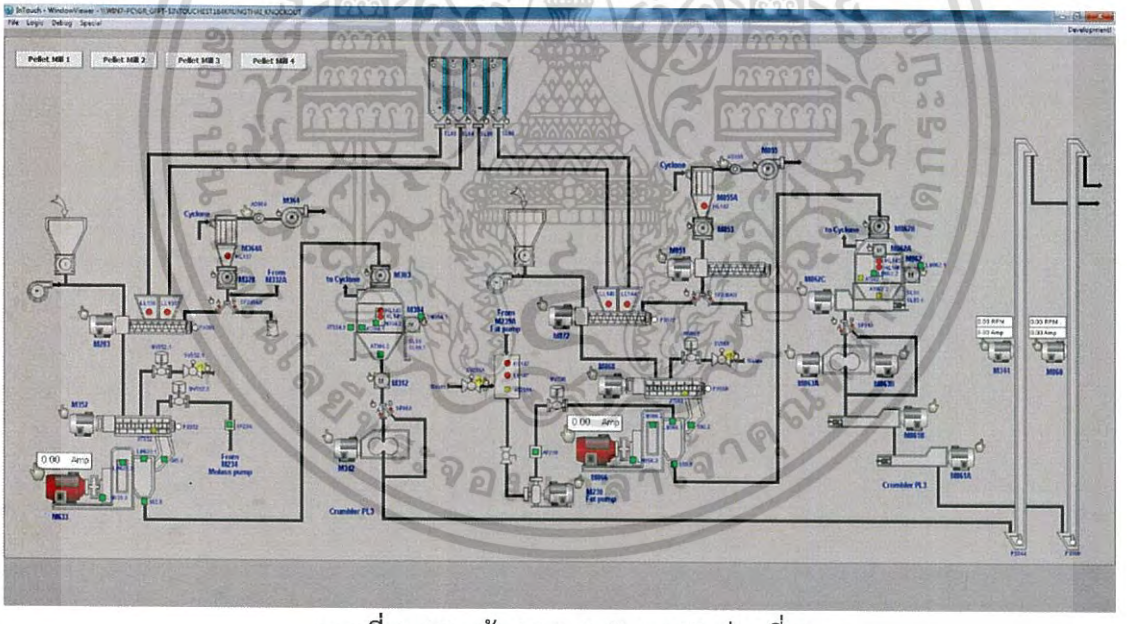
ภาพที่ 3.63 ตัวอย่าง QuickFunctios

จากการดำเนินการตามลำดับข้างต้นจาก ขั้นตอนที่ 3.4.1 – 3.4.8 จึงสามารถสร้างหน้า HMI ที่ใช้สำหรับการแสดงผล และควบคุมกระบวนการอัดเม็ดอาหารสัตว์ โดยแบ่งออกเป็น 4 ส่วนดังภาพที่ 3.64 ถึงภาพที่ 3.67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

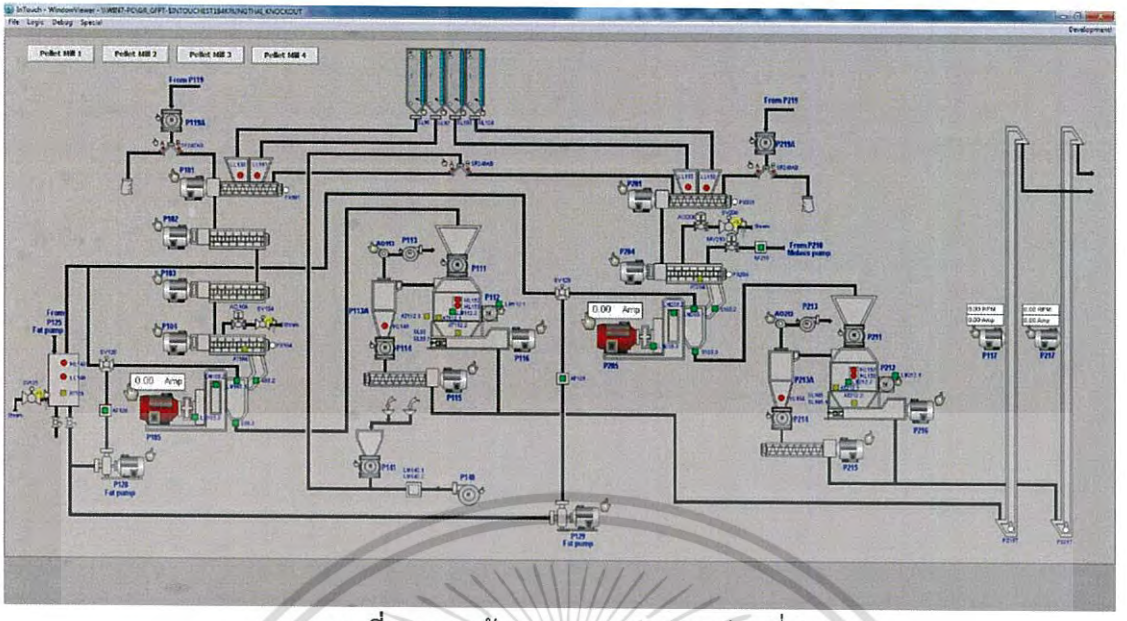


ภาพที่ 3.64 หน้า HMI Pellet Mill ส่วนที่ 1

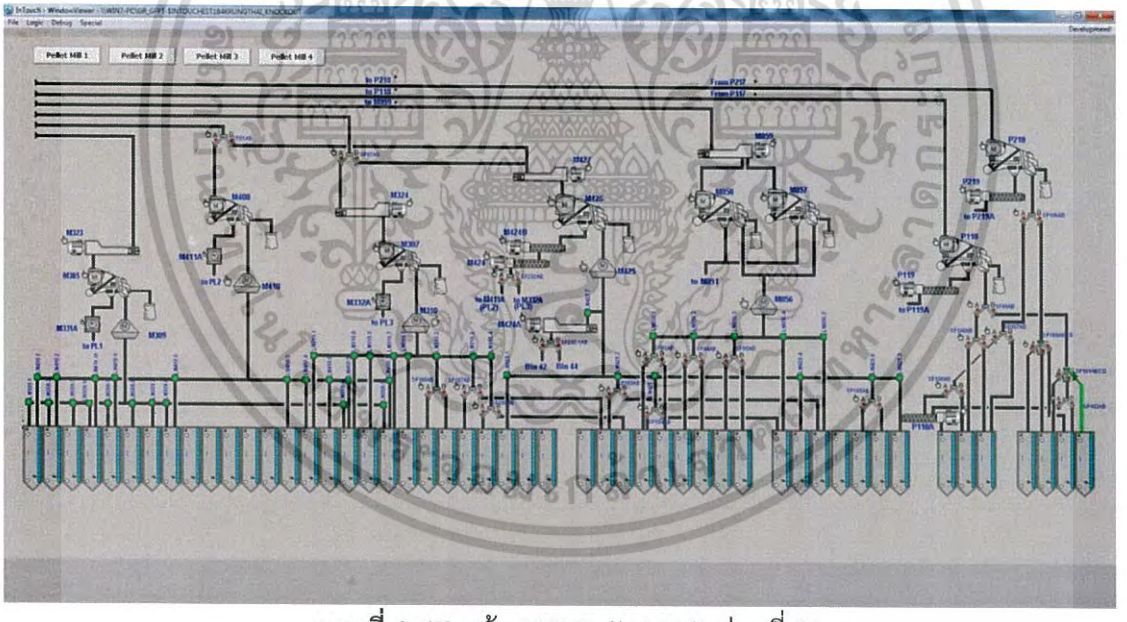


ภาพที่ 3.65 หน้า HMI Pellet Mill ส่วนที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.66 หน้า HMI Pellet Mill ส่วนที่ 3

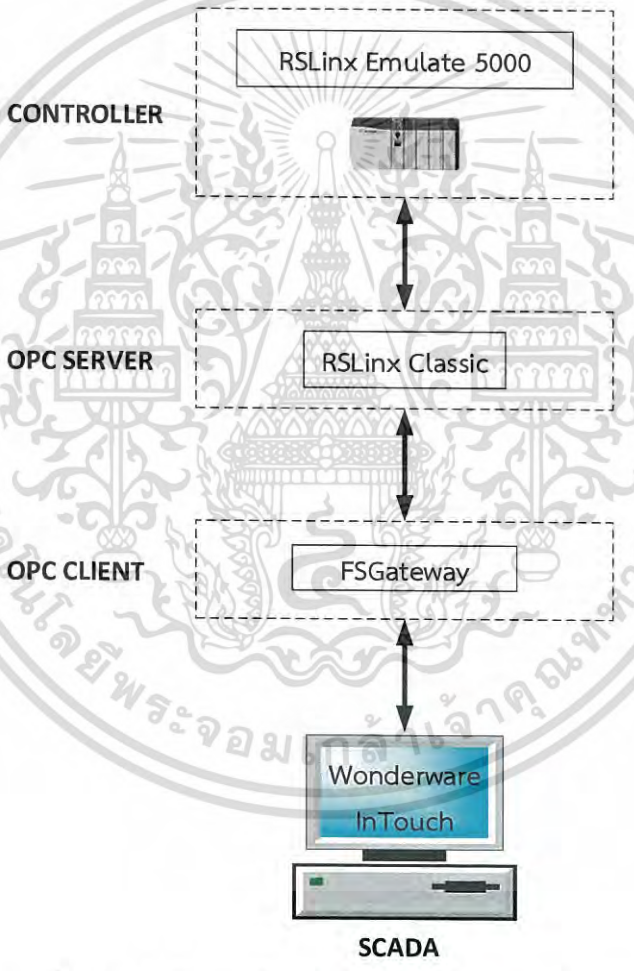


ภาพที่ 3.67 หน้า HMI Pellet Mill ส่วนที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อสร้างส่วนแสดงผลเสร็จแล้วจึงต้องทำการเชื่อมต่อส่วนแสดงผล และโปรแกรมควบคุมอุปกรณ์เข้าด้วยกัน ซึ่งจำเป็นต้องมีตัวกลางในการเชื่อมต่อระหว่าง PLC และส่วนแสดงผล โดยใช้ตัวกลาง คือ ซอฟท์แวร์ System Management Console (SMC) โดยใช้ DAServer Manager และใช้ OPC ที่ชื่อว่า ArchestrA.FSGateway.3 ซึ่งอยู่ในซอฟท์แวร์ System Management Console เป็นตัวช่วยในการเชื่อมต่อ

สำหรับการสร้างการเชื่อมต่อนี้มีวิธีการเชื่อมต่อเป็นลำดับ โดยตัวควบคุม คือ ซอฟท์แวร์ RSLinx Emulate 5000 แทน PLC จริง OPC Server คือ ซอฟท์แวร์ RSLinx Classic และ OPC Client คือ แอปพลิเคชั่น ArchestrA.FSGateway.3 ซึ่งเชื่อมต่อกับซอฟท์แวร์ Wonderware InTouch ที่ใช้ติดตามผล และควบคุมอุปกรณ์ในกระบวนการทั้งหมด ดังภาพที่ 3.68



ภาพที่ 3.68 ลำดับการเชื่อมต่อของส่วนแสดงผล และ PLC

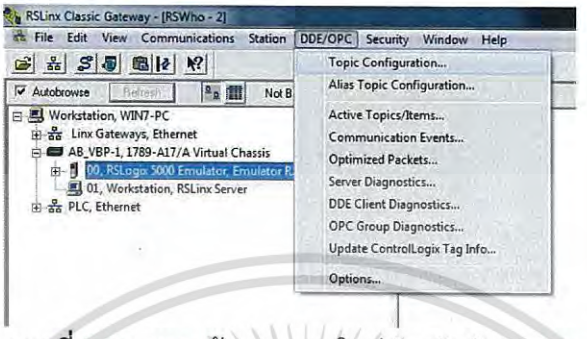
การตั้งค่าการเชื่อมต่อของส่วนแสดงผล และโปรแกรมควบคุมมีขั้นตอนดังต่อไปนี้

3.4.9 การเชื่อมต่อซอฟท์แวร์ RSLinx Emulate 5000 และซอฟท์แวร์ RSLinx Classic

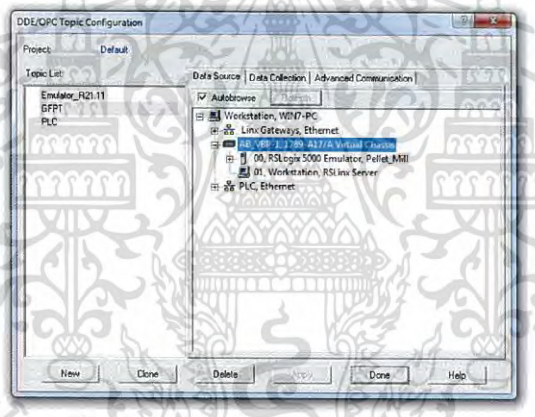
1) เปิดซอฟท์แวร์ RSLinx Emulate 5000 ก่อนแล้วจึงเปิดซอฟท์แวร์ RSLinx Classic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) จากนั้นคลิกขวาที่ DDE/OPC > Topic Configuration... แล้วคลิกที่ New เพื่อสร้าง Topic ใหม่แล้วเปลี่ยนชื่อเป็น GFPT จากนั้นคลิกเลือกตัวควบคุมเพื่อเลือก Data Source ของ Topic นั้น ๆ แล้วคลิก Done ดังภาพที่ 3.69 และภาพที่ 3.70



ภาพที่ 3.69 การสร้าง Topic ใหม่สำหรับตัวควบคุม



ภาพที่ 3.70 การเลือกตัวควบคุมให้กับ Topic

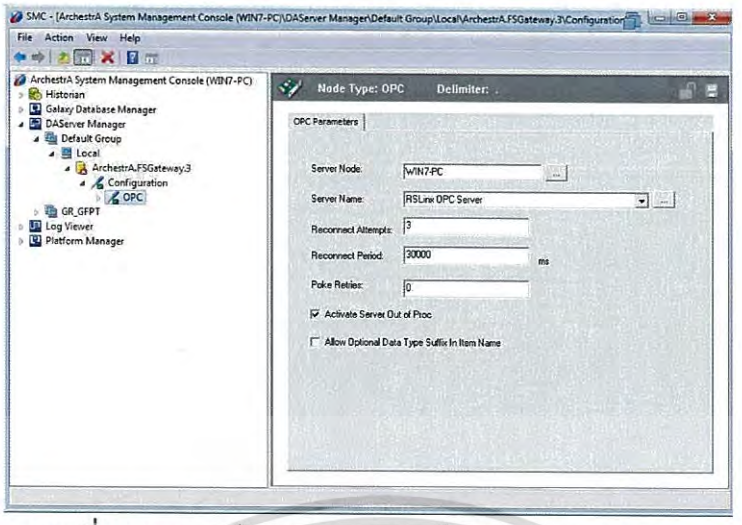
3.4.10 การตั้งค่า DAServer Manager ในซอฟท์แวร์ System Management Console

1) เปิดซอฟท์แวร์ System Management Console จากนั้นคลิก DAServer Manager > Default Group > Local > ArchestrA.FSGateway.3

2) คลิกขวาที่ Configuration > Add OPC Object แล้วเปลี่ยนชื่อเป็น OPC จากนั้นเปลี่ยนค่าของ Object ที่สร้างขึ้นดังนี้ Server Node เปลี่ยนเป็น ชื่อคอมพิวเตอร์ (Computer Name) และ Server Name เปลี่ยนป็น RSLinx OPC Server ดังภาพที่ 3.71

3) คลิกขวาที่ OPC > Add OPC Group Object แล้วเปลี่ยนชื่อเป็น GFPT แล้วเปลี่ยน OPC Item ID Prefix ให้เป็น [GFPT] ดังภาพที่ 3.72

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.71 การสร้าง OPC Object ใน DAServer Manager

ภาพที่ 3.72 การตั้งค่า OPC Object ใน DAServer Manager

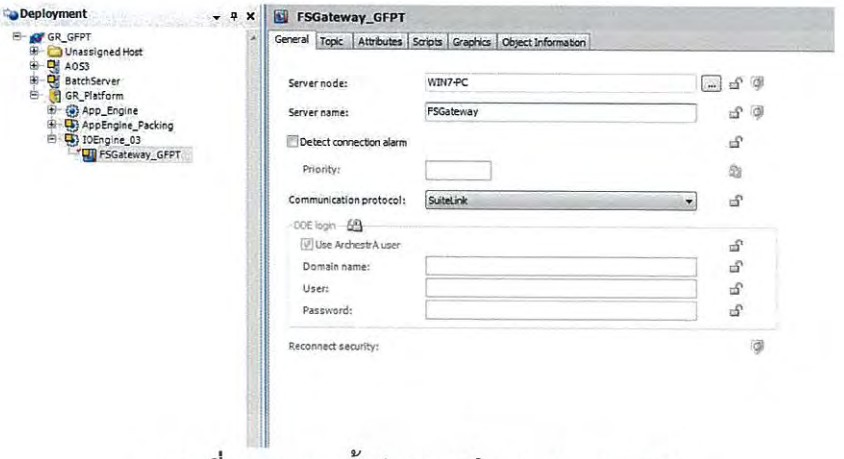
3.4.11 การตั้งค่า OPC ในซอฟท์แวร์ ArchestrA IDE

1) เปิดซอฟท์แวร์ ArchestrA IDE แล้วคลิกขวาที่ $DDESuitLinkClient > New > Instance แล้วเปลี่ยนชื่อเป็น FSGateway_GFPT

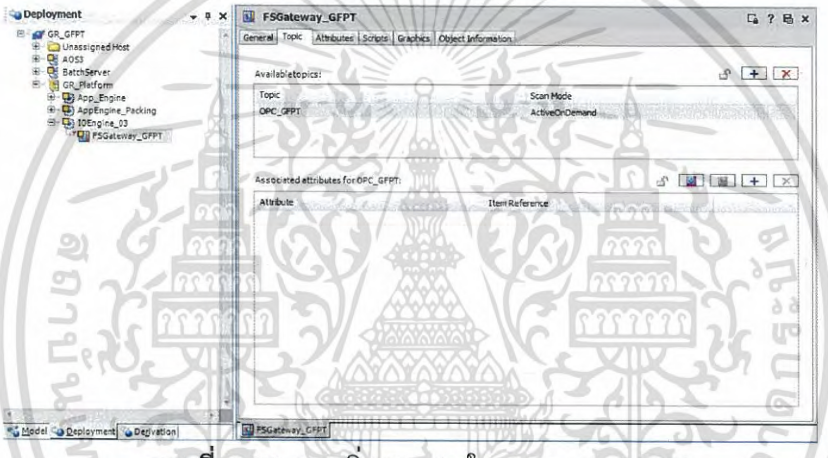
2) เปิด FSGateway_GFPT คลิกที่แทป General แล้วเปลี่ยน Server node เป็น ชื่อคอมพิวเตอร์ และเปลี่ยน Server name เป็น FSGateway ดังภาพที่ 3.73

2) คลิกที่แทป Topic แล้วคลิกเครื่องหมายบวก (+) จากนั้นเปลี่ยนชื่อเป็น OPC_GFPT แล้วคลิกที่ปุ่ม Save ดังภาพที่ 3.74

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.73 การตั้งค่า OPC ใน ArchestrA IDE



ภาพที่ 3.74 การเพิ่ม Topic ใน ArchestrA IDE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการดำเนินงาน

4.1 กล่าวนำ

จากการดำเนินงานใน บทที่ 3 การสร้างระบบสำหรับควบคุมอุปกรณ์ในกระบวนการอัดเม็ดอาหารสัตว์ มีผลการดำเนินงาน การทดสอบระบบดังต่อไปนี้

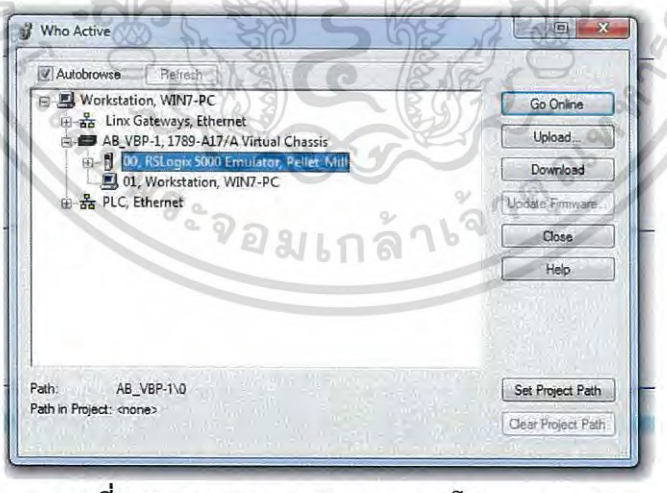
4.2 ผลการทดสอบระบบ

4.2.1 การทดสอบการเชื่อมต่อส่วนแสดงผล และตัวควบคุม

เนื่องจากโครงการนี้อยู่ในขั้นตอนการสร้าง และทดสอบจึงใช้ตัวควบคุมแบบจำลอง คือ ซอฟท์แวร์ RSLinx Emulate 5000 แทน PLC จริง ในการทดสอบการเชื่อมต่อของส่วนแสดงผล และ PLC เพื่อให้สามารถแสดงสถานะการทำงานของอุปกรณ์ และสามารถควบคุมการทำงานของอุปกรณ์ได้ มีขั้นตอนดังนี้

4.2.1.1 การทดสอบการเชื่อมต่อซอฟท์แวร์ Studio 5000 และ PLC ผ่านซอฟท์แวร์ RSLinx Classic

1) เปิดซอฟท์แวร์ RSLinx Emulate 5000 และซอฟท์แวร์ RSLinx Classic จากนั้นจึงเปิดโปรเจคของโปรแกรมควบคุมที่สร้างขึ้นมาผ่านซอฟท์แวร์ Studio 5000 แล้วคลิกขวาที่ Communications > Who Active แล้วลือกตัวควบคุมที่ต้องการ จากนั้นจึงคลิก Download เพื่อทำการดาวน์โหลดโปรแกรมควบคุมสู่ PLC ดังภาพที่ 4.1



ภาพที่ 4.1 การทดสอบ Download โปรแกรม PLC

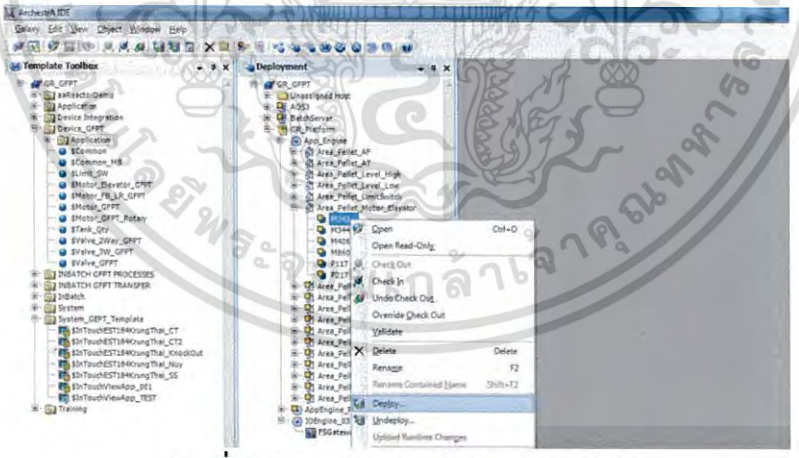
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) คลิกที่แถบเลือกโหมดตัวควบคุม แล้วเปลี่ยนเป็น Run Mode เพื่อเปลี่ยนโหมดให้ตัวควบคุมทำงานตามโปรแกรม ดังภาพที่ 4.2

ภาพที่ 4.2 การเปลี่ยนโหมด PLC

4.2.1.2 การทดสอบการเชื่อมต่อซอฟท์แวร์ Wonderware Intouch และ PLC ผ่านทางซอฟท์แวร์ System Management Console

1) Deploy Object ของอุปกรณ์ต่าง ๆ ที่สร้างขึ้นโดยมีลำดับการ Deploy ดังนี้ Platform > Engine > Area > Object ตามลำดับ ดังภาพที่ 4.3

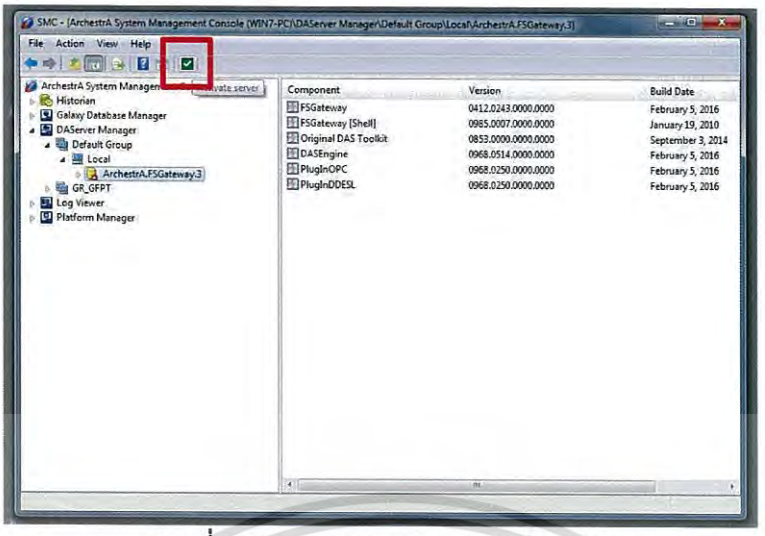


ภาพที่ 4.3 การ Deploy Object ของ HMI

2) เปิดซอฟท์แวร์ Wonderware Intouch จากนั้น Runtime หน้า HMI ที่ได้สร้างขึ้น

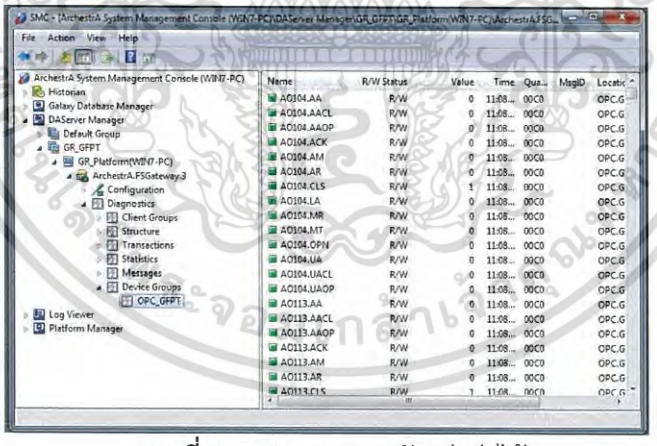
3) เปิดซอฟท์แวร์ System Management Consol แล้วคลิก Activate Server เพื่อเปิดการทำงานของ ArchestrA.FSGateway.3 ดังภาพที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 4.4 การ Activate Server ใน SMC

4) ตรวจสอบ I/O โดยคลิก Dropdown ที่ ArchestrA.FSGateway.3 > Diagnostics > Device Groups หากอุปกรณ์ทุกตัวเป็นสีเขียวแปลว่าสามารถรับค่าได้ถูกต้อง และสามารถส่งข้อมูลไปยัง Wonderware InTouch ได้ ดังภาพที่ 4.5 แต่หากเป็นสีเหลือง หรือสีแดงแปลว่าไม่สามารถรับส่งข้อมูลได้



ภาพที่ 4.5 I/O สามารถรับ-ส่งค่าได้

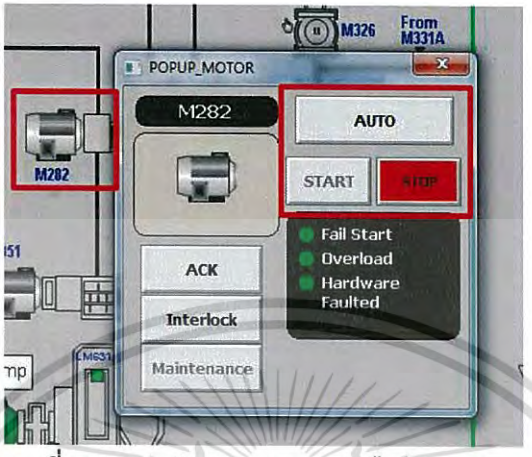
4.2.2 การทดสอบการควบคุมอุปกรณ์ในกระบวนการผ่านส่วนแสดงผล

หลังจากการทดสอบการเชื่อมต่อกันระหว่างส่วนแสดงผล และ PLC แล้วจึงทดสอบการควบคุมอุปกรณ์ต่าง ๆ ในกระบวนการอัดเม็ดอาหารสัตว์ ดังต่อไปนี้

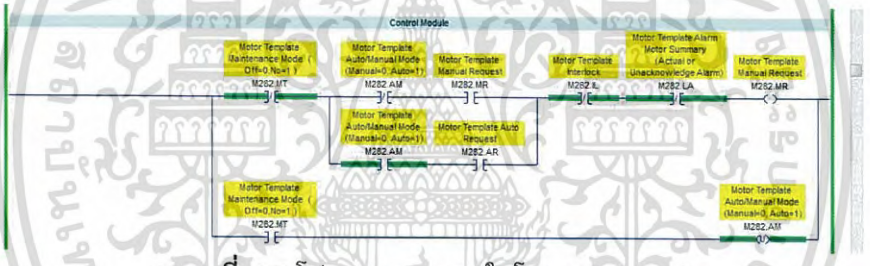
4.2.2.1 การทดสอบโหมดการทำงานของอุปกรณ์ โดยอุปกรณ์แต่ละชนิดมี 2 โหมด คือ Manual Mode และ Auto Mode โดยคลิกที่ตัวอุปกรณ์แล้วมี Popup ของอุปกรณ์ขึ้นมา จากนั้นคลิกเลือกโหมดทำให้ส่งบิต M282.AM เป็นบิตที่ใช้กำหนดโหมดการทำงานของอุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) โหมด Auto เป็นโหมดที่ไม่สามารถเปิดปิดตามได้ต้องการ และต้องทำงานตามโปรแกรมควบคุมเท่านั้น ดังภาพที่ 4.6 ที่แสดงโหมด Auto ของอุปกรณ์ และภาพที่ 4.7 แสดงโปรแกรมควบคุมเมื่อเลือกโหมด Auto

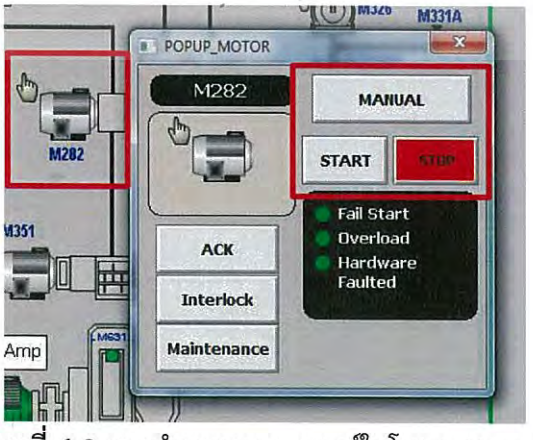


ภาพที่ 4.6 การทำงานของมอเตอร์ในโหมด Auto



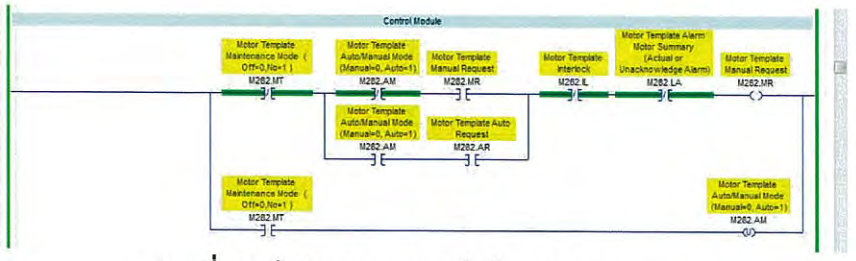
ภาพที่ 4.7 โปรแกรมควบคุมในโหมด Auto

2) โหมด Manual เป็นโหมดที่สามารถเปิดปิดอุปกรณ์ได้ตามต้องการ โดยสังเกตุโหมด Manual ได้จากรูปมือใกล้กันกับมอเตอร์ ดังภาพที่ 4.8 และภาพที่ 4.9 แสดงโปรแกรมควบคุมเมื่อเลือกโหมด Manual



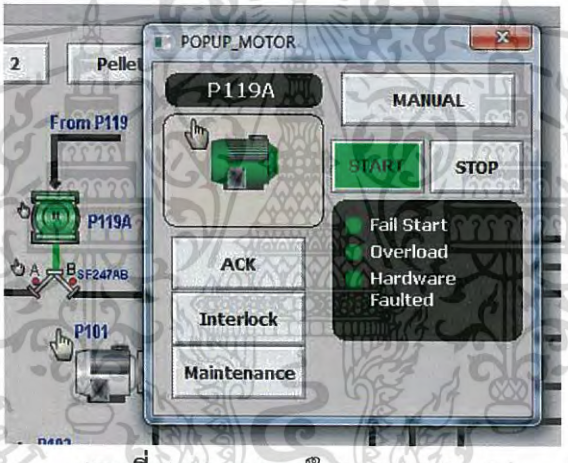
ภาพที่ 4.8 การทำงานของมอเตอร์ในโหมด Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

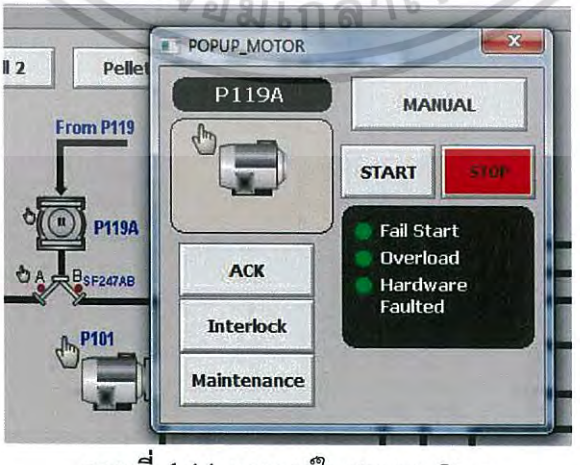


ภาพที่ 4.9 โปรแกรมควบคุมในโหมด Manual

4.2.2.2 การทดสอบเปิดปิดอุปกรณ์ อุปกรณ์ในกระบวนการส่วนใหญ่เป็นอุปกรณ์ที่มีเพียงสองสถานะ คือ เปิด และปิด โดยเมื่อเปิดอุปกรณ์ให้คลิกที่ปุ่ม START ส่วนแสดงผลส่งค่า “1” ไปยังบิต P119A.MR ในทางตรงกันข้าม เมื่อคลิกปุ่ม STOP ส่วนแสดงผลส่งค่า “0” ไปแทน ดังภาพที่ 4.10 และ 4.11 ที่แสดงมอเตอร์ในสถานะ Start และ Stop ตามลำดับ



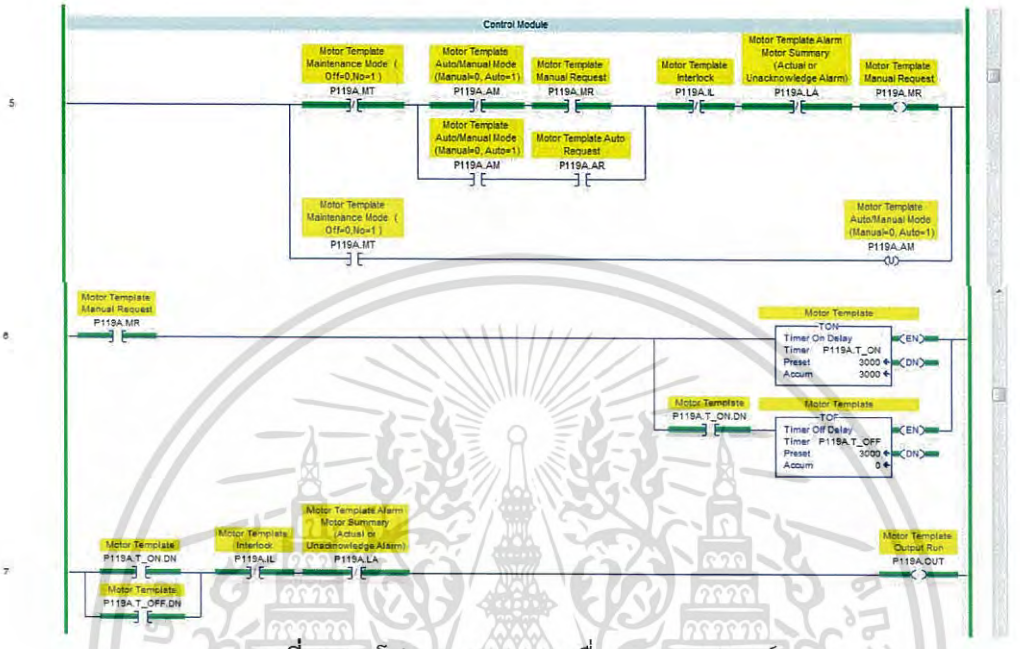
ภาพที่ 4.10 มอเตอร์ในสถานะ Start



ภาพที่ 4.11 มอเตอร์ในสถานะ Stop

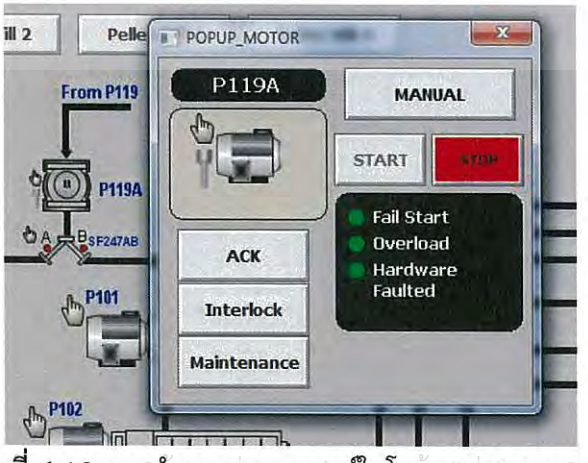
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลำดับการทำงานของโปรแกรมควบคุมเมื่อ Start อุปกรณ์ โดยหากกดปุ่ม Start อุปกรณ์ บิตเอาท์พุต P119A.MR มีค่าเป็น “1” ส่งผลให้ Timer สำหรับหน่วงเวลาทำงาน เมื่อครบเวลา (3000 mS) บิต P119A.T_OFF.DN มีค่าเป็น “1” ทำให้บิตเอาท์พุต P119A.OUT มีค่าเป็น “1” และสั่งให้อุปกรณ์ทำงาน ส่วนการปิดอุปกรณ์ตรงข้ามกับการเปิด ดังภาพที่ 4.12



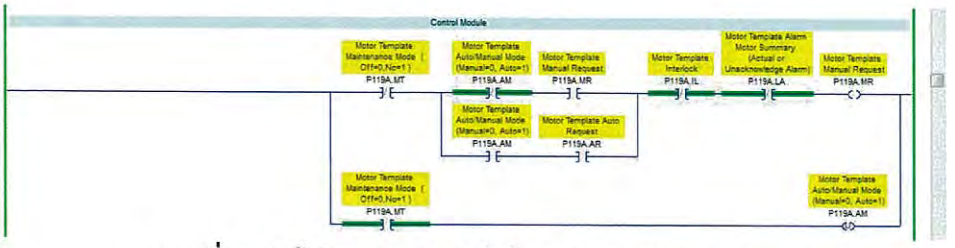
ภาพที่ 4.12 โปรแกรมควบคุมเมื่อ Start อุปกรณ์

4.2.2.3 การทดสอบโหมด Mantainance ซึ่งเป็น Safety โหมดที่มีไว้สำหรับตัดการเชื่อมต่อระหว่างอุปรณ์กับ PLC โหมดนี้ใช้งานเมื่อต้องมีการเข้าไปทำงานกับอุปกรณ์ที่หน้างานเพื่อเปลี่ยนหรือซ่อมแซม โดยเมื่อเข้าโหมด Maintenance แล้วอุปกรณ์ต้องไม่ทำงาน และไม่สามารถสั่งการได้จากส่วนแสดงผล โดยเมื่อคลิกปุ่ม Maintenance แล้วส่วนแสดงผลส่งบิต P119A.MT ไปตัดการทำงานของโปรแกรมควบคุมมอเตอร์ ดังภาพที่ 4.13 แสดงการทำงานของมอเตอร์ในโหมด Maintenance และภาพที่ 4.14 แสดงโปรแกรมควบคุมเมื่อเลือกโหมด Mantainance



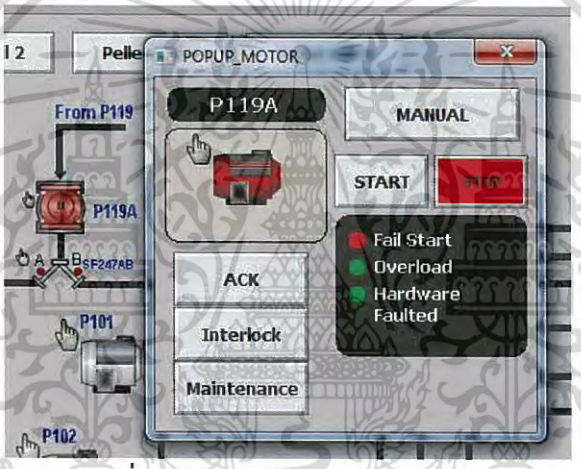
ภาพที่ 4.13 การทำงานของมอเตอร์ในโหมด Maintenance

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



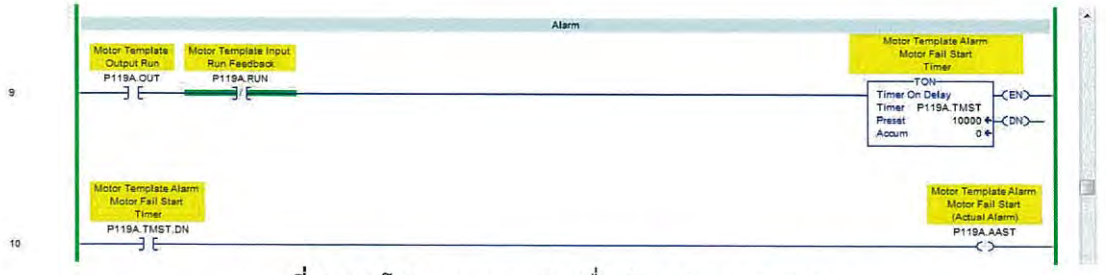
ภาพที่ 4.14 โปรแกรมควบคุมในโหมด Maintenance

4.2.2.4 การทดสอบโหมด Alarm เป็นโหมดใช้สำหรับแจ้งเตือนผู้ปฏิบัติงานเมื่อเกิดเหตุขัดข้องกับอุปกรณ์ หรือกระบวนการ ตัวอย่าง เมื่อสั่งเปิดอุปกรณ์จากหน้าจอ HMI แต่อุปกรณ์ไม่ทำงานตามคำสั่ง หรือไม่มีสัญญาณ Feedback จึงเกิด Fail Start ดังภาพที่ 4.15



ภาพที่ 4.15 ตัวอย่างมอเตอร์ Fail Start

เมื่อสั่งให้อุปกรณ์ให้ทำงานแล้ว แต่ไม่มีสัญญาณ Feedback กลับมาที่บิต P119A.RUN ภายในเวลาที่กำหนดไว้ใน Timer (10000 mS) ส่งผลให้บิต P119A.TMST.DN มีค่าเป็น “1” แล้วบิตเอาท์พุตของ Alarm คือ P119A.UAST มีค่าเป็น “1” ทำให้เกิด Alarm Fail Start ขึ้น ดังภาพที่ 4.16



ภาพที่ 4.16 โปรแกรมควบคุมเมื่อเกิด Alarm Fail Start

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผล ปัญหา และข้อเสนอแนะ

5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานในเรื่อง การสร้างระบบควบคุมสำหรับกระบวนการอัดเม็ดอาหารสัตว์ซึ่งเป็นส่วนหนึ่งของโครงการปรับปรุงระบบควบคุมของโรงงานผลิตอาหารสัตว์แห่งหนึ่งจากเดิมที่ควบคุมอุปกรณ์การผลิตผ่าน Mimic Diagram Panel โดยใช้ผู้ปฏิบัติงานควบคุม ซึ่งประสิทธิภาพของการผลิตขึ้นอยู่กับประสิทธิภาพของผู้ปฏิบัติงานแต่ละบุคคล จึงเปลี่ยนให้เป็นระบบควบคุมการผลิตแบบอัตโนมัติที่ควบคุมอุปกรณ์ในการผลิตผ่านทางส่วนแสดงผลของระบบ SCADA ซึ่งช่วยให้ผู้ปฏิบัติงานสามารถติดตามผล และควบคุมได้จากห้องควบคุมผ่านส่วนแสดงผล โดยไม่จำเป็นต้องลงไปปฏิบัติงานที่หน้างานจริง ทำให้เกิดความปลอดภัยแก่ผู้ปฏิบัติงาน ลดจำนวนผู้ปฏิบัติงาน ลดค่าใช้จ่ายในการว่าจ้างแรงงาน และเพิ่มประสิทธิภาพในการผลิตอีกด้วย

โครงการนี้ได้ทำการสร้างส่วนแสดงผล และส่วนโปรแกรมควบคุมของกระบวนการ โดยได้ทดสอบฟังก์ชันต่าง ๆ ด้วยซอฟท์แวร์ตัวควบคุมจำลอง (RSLinx Emulate 5000) โดยหลังจากนี้สามารถนำส่วนแสดงผล และโปรแกรมควบคุมกระบวนการไปทดสอบกับระบบควบคุมของจริงที่ใช้งานในโรงงานเพื่อทดสอบอินพุต/เอาท์พุต เพื่อปรับปรุงให้เป็นไปตามความต้องการของผู้ปฏิบัติงาน

5.2 ปัญหา และวิธีการแก้ไข

5.2.1 ปัญหาที่พบ

การสร้าง และทดสอบระบบที่สร้างขึ้นไม่เป็นไปตามความต้องการของผู้ปฏิบัติงานจึงมีการปรับเปลี่ยนโปรแกรมควบคุม และส่วนแสดงผลอยู่บ่อย ๆ ทำให้เกิดความล่าช้าในการทำงาน

5.2.2 วิธีการแก้ไข

จากปัญหาเรื่องระบบที่ออกแบบมาไม่ตรงตามความต้องการของผู้ปฏิบัติงานจึงต้องมีการประชุมสรุปงานร่วมกับผู้ปฏิบัติงานทุกคนทุกครั้งก่อนการสร้างระบบในส่วนต่อไป

5.3 ข้อเสนอแนะ

ในการออกแบบระบบ โครงสร้างของทั้งส่วนแสดงผล และโปรแกรมควบคุมออกเป็นส่วน ๆ เพื่อให้ง่ายต่อการจัดการเป็นสิ่งสำคัญอันดับแรกในการพัฒนาระบบควบคุม ซึ่งช่วยลดปัญหาในการปรับเปลี่ยนกราฟฟิก และโปรแกรมควบคุมได้ในระดับหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] อาหาร และการให้อาหารสัตว์ แหล่งที่มา: http://elearning.nsru.ac.th/web_elearning/animal/lesson6_1.php

[2] การผลิตอาหารสัตว์ แหล่งที่มา: http://www.kasetyaso.ac.th/sinee/public1.html

[3] กระบวนการผลิตอาหารเม็ดสัตว์น้ำ แหล่งที่มา: http://thai.pellet-making-machine.com/pellet-making-guide/fish-feed-pellet-making/aquatic-animal-feed-production-procedure.html

[4] SCADA คืออะไร แหล่งที่มา: http://www.binaryadvance.com/SCADA.html

[5] SCADA System แหล่งที่มา: https://riverplusblog.com/tag/scada-คือ/

[6] SCADA แหล่งที่มา: https://en.wikipedia.org/wiki/SCADA

[7] ControlLogix 5570 แหล่งที่มา: http://ab.rockwellautomation.com/Programmable-Controllers/ControlLogix/5570-Controllers

[8] Wonderware Application Server User's Guide แหล่งที่มา: www.logic-control.com/datasheets/1/Application%20Server/IDE.pdf

[9] Wonderware System Platform 2014 New Features Guide แหล่งที่มา: www.wonderware.fi/pdf/Wonderware_SP2014_New_Features_Guide_en_0514.pdf

[10] Studio 5000 แหล่งที่มา: http://www.rockwellautomation.com/rockwellsoftware/products/studio-5000.page/

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้