



## รายงานสหกิจศึกษาฉบับสมบูรณ์

การจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP  
(Foxboro SCP-Based Simulation of Supply and Demand Balancing for  
Power Plant)

นางสาวอรปรียา พรหมเจริญ

หลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิชาวิศวกรรมอัตโนมัติ  
ภาควิชาวิศวกรรมการวัดคุมและควบคุม  
คณะวิศวกรรมศาสตร์  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ปีการศึกษา 2560



## รายงานสหกิจศึกษาฉบับสมบูรณ์

การจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP  
(Foxboro SCP-Based Simulation of Supply and Demand Balancing for  
Power Plant)

นางสาวอรปรียา พรหมเจริญ

หลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอัตโนมัติ

ภาควิชาวิศวกรรมการวัดคุมและควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงการสหกิจศึกษา	การจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP
นักศึกษา	นางสาวอรปรียา พรหมเจริญ
คณะ วิศวกรรมศาสตร์	สาขาวิชา วิศวกรรมอัตโนมัติ
อาจารย์นิเทศ	ผศ.สาท คำมูล รศ.ดร.ไสว พงศ์สวัสดิ์
ผู้นิเทศงาน	คุณกฤษฎา ไสยสุข
สถานประกอบการ	บริษัท ซไนเดอร์ อีเล็กทริก ซิสเต็มส์ (ประเทศไทย) จำกัด

### บทคัดย่อ

การจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP ช่วยเพิ่มประสิทธิภาพในการผลิตกระแสไฟฟ้าของโรงไฟฟ้า ด้วยแนวคิดพื้นฐานของ Plant Management Control (PMC) เป็นการปรับสมดุลระหว่างค่าอุปสงค์ที่เป็นกำลังการผลิตของโรงไฟฟ้าที่มีการใช้ Gas Turbine Generator 2 ตัว และ Stream Turbine Generator 1 ตัว ส่วนค่าอุปทานเป็นความต้องการใช้ไฟฟ้าของลูกค้าซึ่งเป็นโรงงาน 5 แห่ง ซึ่งจะศึกษาในกรณีที่ค่าอุปสงค์และอุปทานไม่สมดุล อาจเกิดได้จาก 2 เหตุการณ์ ที่เรียกว่า Load Shed และ Generator Shed ทั้งนี้ผลของการจำลองสามารถนำมาใช้ป้องกันระบบของการส่งจ่ายพลังงานไฟฟ้า

คำสำคัญ : Load Shed, Generator Shed, Foxboro SCP

**Co-operative Title:** Foxboro SCP-Based Simulation of Supply and Demand Balancing for Power Plant

**Student Intern Name:** Miss. Onpreeya Phromjaroen

**Faculty:** Engineer      **Department:** Automation Engineering

**Advisor Name:** Asst.Prof.Sart Kummool  
Assoc.Prof.Dr.Sawai Pongswatd

**Mentor Name:** Kritsada Saiyasuk

**Company:** Schneider Electric Systems (Thailand) Co.,Ltd.

## ABSTRACT

In order to provide the effectiveness for electrical generation of a power plant, a simulation base on Foxboro SCP for supply and demand balancing is presented. Based on Plant Management Control (PMC), The balance between the supply of the power plant having two gas turbine generators and one stream turbine generator and the demand of five factory customers is examined as an illustrate case study. In case of unbalanced supply and demand, two events called ‘Load Shed’ and ‘Generator Shed’ are occurred. The simulation results can be useful for designing a protection system in power distribution.

**Keywords :** Load Shed, Generator Shed, Foxboro SCP

## กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ นักศึกษาชั้นปีที่ 4 ได้เสนอเป็นส่วนหนึ่งในหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมอัตโนมัติ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2560 ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความอนุเคราะห์จาก บริษัท ชไนเดอร์ อิเล็กทริก ซิสเต็มส์ (ประเทศไทย) จำกัด ที่ให้โอกาสสำหรับนักศึกษาชั้นปีที่ 4 ได้เข้าร่วมโครงการสหกิจศึกษากับบริษัท อีกทั้งขอขอบพระคุณ คุณกฤษฎา ไสยสุข ผู้นิเทศงาน และบุคลากรในบริษัททุกท่านที่คอยแนะนำ ให้คำปรึกษาตลอดระยะเวลาสี่เดือน

ขอขอบพระคุณ ผศ.สาท คำมูล และรศ.ดร.ไสว พงศ์สวัสดิ์ ที่ได้ให้ความเมตตา และคำแนะนำแก่ผู้จัดทำตลอดช่วงระยะเวลาการทำโครงการสหกิจ และขอขอบพระคุณอาจารย์ประจำหลักสูตรวิศวกรรมอัตโนมัติทุกท่านที่ได้ถ่ายทอดวิชาความรู้ทั้งทางด้านทฤษฎี ทางด้านปฏิบัติ ประสบการณ์ ให้คำปรึกษา และให้คำแนะนำตลอดการศึกษาในระดับปริญญาตรีที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมอัตโนมัติ

สุดท้ายนี้ ต้องขอขอบพระคุณครอบครัว ที่ให้การสนับสนุน คำแนะนำ และเป็นกำลังใจที่ดีตลอดมา ขอขอบคุณพี่ๆ และเพื่อน สาขาวิศวกรรมอัตโนมัติ ที่คอยให้กำลังใจ ให้คำแนะนำ และคำปรึกษาในการจัดทำโครงงาน

อรปรียา พรหมเจริญ

## สารบัญ

หน้า

บทคัดย่อ .....	II
ABSTRACT .....	III
กิตติกรรมประกาศ .....	IV
สารบัญ.....	V
สารบัญตาราง.....	VII
สารบัญรูปภาพ.....	VIII
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปริญญาโท.....	1
1.2 วัตถุประสงค์ของปริญญาโท.....	1
1.3 ขอบเขตของปริญญาโท.....	2
1.4 ขั้นตอนการศึกษา .....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ .....	3
บทที่ 2 รายละเอียดของกรณีศึกษา.....	4
2.1 กล่าวนำ .....	4
2.2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.2.1 ธุรกิจผลิตไฟฟ้า .....	4
2.2.2 ระบบโครงข่ายไฟฟ้าอัจฉริยะ.....	6
2.2.3 ระบบควบคุมการผลิตและการจ่ายพลังงานไฟฟ้าที่เป็นกรณีศึกษา (Plant Management Control) .....	11
2.2.4 การนำแนวคิด PMC มาประยุกต์ใช้ในกรณีศึกษา.....	12
2.3 ฮาร์ดแวร์ที่เกี่ยวข้อง .....	13
2.4 ซอฟต์แวร์ที่เกี่ยวข้อง.....	14
บทที่ 3 การจำลองการปรับสมดุลอุปสงค์และอุปทานที่นำเสนอ.....	17
3.1 กล่าวนำ .....	17
3.2 รายละเอียดของระบบ .....	17

## สารบัญ (ต่อ)

	หน้า
3.2.1 โครงสร้างของระบบ .....	17
3.2.2 Flowchart การทำงาน .....	18
3.2.3 ตัวอย่างกรณีศึกษา.....	21
3.3 ขั้นตอนการดำเนินงานด้านซอฟต์แวร์.....	23
3.3.1 โปรแกรม SCP Launcher.....	23
3.3.2 การสร้าง GUI โดย FoxDraw .....	25
3.3.3 การเขียน Database.....	28
3.3.4 การสร้าง Block CALC .....	33
3.3.5 การเขียนโค้ดลงใน Block CALC .....	37
3.4 ส่วนประกอบการทำงานของ Plant Management Control ในโปรแกรม IACC.....	41
3.4.1 CODE : PMC (PLANT MANAGEMENT CONTROL).....	41
3.4.2 CODE : LOADSHED .....	47
3.5 ส่วนประกอบการทำงานของ Plant Management Control ในโปรแกรม FOXDRAW .....	49
3.6 ส่วนประกอบการทำงานของ Plant Management Control ในโปรแกรม FOXVIEW .....	51
<b>บทที่ 4 ผลการดำเนินงาน.....</b>	<b>53</b>
4.1 กล่าวนำ .....	53
4.1.1 ส่วนซอฟต์แวร์.....	53
4.1.2 ส่วนการแสดงผลและควบคุมการทำงาน .....	56
4.2 ผลการดำเนินงาน .....	57
<b>บทที่ 5 สรุปผลและปัญหาในการดำเนินโครงการ .....</b>	<b>58</b>
5.1 สรุปผลการดำเนินงาน .....	58
5.2 ปัญหาและอุปสรรค .....	58
5.3 ข้อเสนอแนะ .....	58
<b>เอกสารอ้างอิง.....</b>	<b>59</b>

## สารบัญตาราง

ตาราง	หน้า
ตารางที่ 1.1 แผนการดำเนินงาน .....	3

## สารบัญรูปรูปภาพ

รูปที่	หน้า
รูปที่ 2.1 Thailand's Electricity Supply Industry Structure.....	5
รูปที่ 2.2 SMART GRID.....	6
รูปที่ 2.3 ระบบไฟฟ้า (Power System).....	7
รูปที่ 2.4 ระบบควบคุมการผลิตและการจ่ายพลังงานไฟฟ้า.....	11
รูปที่ 2.5 H90 Workstation Server.....	13
รูปที่ 2.6 HP EliteDisplay E241i, 24” LED Monitor.....	14
รูปที่ 2.7 SCP Launcher.....	14
รูปที่ 2.8 Building an IACC Control Strategy Diagram.....	15
รูปที่ 2.9 FoxView.....	15
รูปที่ 2.10 FoxDraw Window.....	16
รูปที่ 3.1 โครงสร้างของระบบ.....	17
รูปที่ 3.2 Flowchart IN/OUT Service of GTG 1.....	18
รูปที่ 3.3 Flowchart IN/OUT Service of GTG 2.....	19
รูปที่ 3.4 Flowchart PMC.....	20
รูปที่ 3.5 ภาพรวมการทำงานของกรณีศึกษาเมื่อ 3YB-01 Close.....	21
รูปที่ 3.6 ภาพรวมการทำงานของกรณีศึกษาเมื่อ 3YB-01 Open.....	22
รูปที่ 3.7 ภาพรวมการทำงานของกรณีศึกษาเมื่อเข้ากระบวนการ Load Shed.....	22
รูปที่ 3.8 ภาพรวมการทำงานของกรณีศึกษาเมื่อเข้ากระบวนการ Generator Shed.....	23
รูปที่ 3.9 การสร้าง SCP.....	23
รูปที่ 3.10 NewScpConfigWindow.....	24
รูปที่ 3.11 การ Run SCP.....	24
รูปที่ 3.12 โปรแกรม FoxView.....	25
รูปที่ 3.13 Foxboro: Change Environment.....	25
รูปที่ 3.14 หน้าต่างโปรแกรม FoxView.....	26
รูปที่ 3.15 โปรแกรม FoxDraw.....	26
รูปที่ 3.16 หน้าต่างสำหรับแสดงผลโดยโปรแกรม FoxView.....	27
รูปที่ 3.17 หน้าต่างสำหรับเลือก Database ของ IACC.....	28

## สารบัญรูปรูปภาพ (ต่อ)

รูปที่	หน้า
รูปที่ 3.18 หน้าต่างสำหรับ Logon .....	28
รูปที่ 3.19 หน้าต่าง Plant ของโปรแกรม IACC .....	29
รูปที่ 3.20 Plant Area.....	29
รูปที่ 3.21 Compound.....	30
รูปที่ 3.22 CSD .....	30
รูปที่ 3.23 Assign Default Compound .....	31
รูปที่ 3.24 Function Block Diagram.....	31
รูปที่ 3.25 การตั้งค่า Block .....	32
รูปที่ 3.26 คำอธิบายของการตั้งค่า Block.....	32
รูปที่ 3.27 การเชื่อมTag ของ Block กับ Compound ในโปรแกรม IACC.....	33
รูปที่ 3.28 Connection Dialog: การDelete Typeที่ไม่ต้องการออกConnection Dialog .....	34
รูปที่ 3.29 Connection Dialog: การ Add Type ที่ต้องการ.....	34
รูปที่ 3.30 Select Connection Type.....	35
รูปที่ 3.31 Select Connection Parameter .....	35
รูปที่ 3.32 Select Object Parameter .....	36
รูปที่ 3.33 เมื่อ Connection Dialog เสร็จ .....	36
รูปที่ 3.34 เมื่อเชื่อม Tag ของ Block กับ Compound ในโปรแกรม IACC เสร็จ.....	37
รูปที่ 3.35 การเขียนโค้ดลงในBlock .....	37
รูปที่ 3.36 CALC:PARK Properties.....	38
รูปที่ 3.37 Hierarchy Validation.....	38
รูปที่ 3.38 Validation .....	39
รูปที่ 3.39 Downloading.....	39
รูปที่ 3.40 ยืนยันการติดตั้ง.....	40
รูปที่ 3.41 FOXDRAW : PLANT MANAGEMENT CONTROL .....	49
รูปที่ 3.42 FOXDRAW : GT_PRI_SET.....	49
รูปที่ 3.43 FOXDRAW : ISOCHRONOUS MODEPRISET .....	50
รูปที่ 3.44 FOXDRAW : POPPRIORITY .....	50

## สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
รูปที่ 3.45 FOXVIEW : แสดงผลการทำงานของ PLANT MANAGEMENT CONTROL.....	51
รูปที่ 3.46 FOXVIEW : แสดงผลการทำงานของ COMPOUND OUTPUT (Block DEP1).....	51
รูปที่ 3.47 FOXVIEW : แสดงผลการทำงานของ COMPOUND SEQUENCE (Block LD_LOADSHED).....	52
รูปที่ 4.1 RUN SCP Launcher.....	53
รูปที่ 4.2 บล็อกการทำงานบนซอฟต์แวร์ IACC.....	54
รูปที่ 4.3 โค้ดการทำงานบนซอฟต์แวร์ IACC.....	54
รูปที่ 4.4 แสดงผลและควบคุมบนซอฟต์แวร์ FoxView.....	55
รูปที่ 4.5 ออกแบบและแก้ไขหน้ากราฟิกบนซอฟต์แวร์ FoxDraw.....	55
รูปที่ 4.6 LOAD SHEDDING CONTROL.....	56
รูปที่ 4.7 ส่วนการกำหนดค่าของการจ่ายพลังงานไฟฟ้าให้แก่โรงงานต่างๆ.....	56
รูปที่ 4.8 GENERATOR SHEDDING CONTROL.....	57

# บทที่ 1

## บทนำ

### 1.1 ที่มาและความสำคัญของปริญญานิพนธ์

ในปัจจุบันพลังงานไฟฟ้านับเป็นอีกปัจจัยหนึ่งที่มีความสำคัญในการดำรงชีวิตประจำวัน ทั้งด้านการอุปโภค การบริโภค การคมนาคม รวมทั้งด้านอุตสาหกรรม ทำให้มีการใช้ไฟฟ้าในจำนวนที่มากขึ้นอย่างต่อเนื่อง จึงมีความต้องการในการใช้ไฟฟ้าหรืออุปทาน (Demand) ที่มากขึ้น ดังนั้นเพื่อเป็นการตอบสนองต่อการใช้งานพลังงานไฟฟ้า ในส่วนของโรงไฟฟ้าที่เกี่ยวข้องกับการผลิตหรืออุปสงค์ (Supply) จึงนำแนวคิดการควบคุมการบริหารไฟฟ้า (Plant Management Control : PMC) มาใช้ในระหว่างการปรับสมดุลกำลังการผลิตหรืออุปสงค์ ทั้งนี้เพื่อให้การปรับสมดุลระหว่างอุปสงค์และอุปทานสำหรับโรงไฟฟ้ามีประสิทธิภาพมากขึ้น การจำลองสถานะต่าง ๆ เพื่อศึกษาความเป็นไปได้ของผลกระทบในกรณีที่อุปสงค์และอุปทานมีความไม่สมดุล โดยผลของการจำลองนี้ สามารถนำไปใช้ในการวางแผนป้องกันการหยุดชะงักของกระบวนการผลิตไฟฟ้าที่อาจเกิดขึ้นได้ หรือเหตุการณ์ที่เครื่องกำเนิดไฟฟ้าผลิตไฟฟ้ามากเกินไปเกินความต้องการทางไฟฟ้า ทำให้เกิดการเปลี่ยนแปลง

จากการศึกษาโรงไฟฟ้าแห่งหนึ่งที่มีการนำแนวคิด PMC มาใช้ในการปรับสมดุลอุปสงค์และอุปทาน มีการติดตั้งเครื่องกำเนิดไฟฟ้า (Generator) เพิ่ม ทำให้มีการเปลี่ยนแปลงกำลังการผลิตของโรงไฟฟ้า โรงไฟฟ้าแห่งนี้จึงมีความต้องการในการจำลองการปรับสมดุลอุปสงค์และอุปทานด้วยค่ากำลังการผลิตที่เปลี่ยนแปลงไป

ในปริญญานิพนธ์จึงนำเสนอระบบที่สร้างด้วย Foxboro SCP ที่ใช้ในการจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าแห่งนี้ โดยระบบจำลองที่สร้างขึ้น มีส่วนติดต่อกับผู้ใช้งานแบบกราฟิก (Graphic User Interface : GUI) ที่สามารถจำลองสถานะที่ไม่สมดุลระหว่างฝั่งอุปสงค์และฝั่งอุปทานได้โดยไม่ต้องมีอุปกรณ์ทางด้านฮาร์ดแวร์ (Hardware) จริง

### 1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. สร้างระบบจำลองการปรับสมดุลระหว่างการผลิตไฟฟ้าของโรงไฟฟ้า (Supply) และความต้องการใช้ไฟฟ้าของลูกค้า (Demand) ด้วยโปรแกรม Foxboro SCP ตามแนวคิดการควบคุมการบริหารโรงไฟฟ้า (Plant Management Control : PMC)

2. เขียนโปรแกรมเพื่อควบคุมการจำลองสถานะที่เป็นไปได้ในการปรับสมดุลระหว่างอุปสงค์และอุปทาน

3. สร้างส่วน GUI สำหรับระบบจำลองการปรับสมดุลที่สร้างขึ้น

### 1.3 ขอบเขตของปริญญานิพนธ์

1. ระบบการจำลองการปรับสมดุลที่สร้างขึ้นเป็นระบบที่ใช้สำหรับกรณีศึกษาของระบบการผลิตและการจ่ายพลังงานไฟฟ้า โดยที่มีการกำหนดให้ค่าอุปสงค์เป็นกำลังการผลิตของโรงไฟฟ้าที่มีการใช้ Gas Turbine Generator 2 ตัว และ Stream Turbine Generator 1 ตัว ส่วนค่าอุปทานเป็นความต้องการใช้ไฟฟ้าของลูกค้าซึ่งเป็นโรงงาน 5 แห่ง

2. มีการจำลอง 2 สถานะ คือ Load Shed และ Generator Shed ที่เป็นผลมาจากเงื่อนไขที่ทำให้ค่าอุปสงค์และอุปทานไม่สมดุล โดยที่ผู้ใช้ระบบจำลองสามารถกำหนดค่าลำดับความสำคัญ (Priority) ในการดำเนินการสถานะ Load Shed และ Generator Shed ได้

3. ส่วน GUI ที่สร้างขึ้น มีการแบ่งพื้นที่ในส่วนแสดงผลสำหรับ Load shed และ Generator shed ไว้ในหน้าเดียวกัน เพื่อความสะดวกในการจำลองการปรับสมดุล

### 1.4 ขั้นตอนการศึกษา

1. ศึกษาโปรแกรม FoxView และ FoxDraw การศึกษาข้อมูลการทำงานของระบบการใช้งานเดิมของบริษัท ศึกษากระบวนการทำงานของ PMC (Plant Management Control)

2. ศึกษารายละเอียดที่เกี่ยวข้อง ได้แก่ การศึกษาการใช้งานดีซีเอสของ Foxboro

3. เขียน Logic การทำงาน

4. ปรับปรุงแก้ไขหน้าจอ GUI

5. ทดสอบฟังก์ชันการทำงานของโปรแกรม

6. แก้ไขข้อผิดพลาดของโปรแกรม

7. ทำปริญญานิพนธ์และแก้ไข

ตารางที่ 1.1 แผนการดำเนินงาน

ลำดับ ที่	รายละเอียด	สิงหาคม				กันยายน				ตุลาคม				พฤศจิกายน			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	ศึกษาโปรแกรม	/	/														
2	ศึกษาการทำงานของระบบ			/	/	/											
3	เขียน logic การทำงาน					/	/	/									
4	ปรับปรุงแก้ไขหน้าจอ GUI								/	/							
5	ทดสอบฟังก์ชันการทำงานของโปรแกรม										/						
6	แก้ไขข้อผิดพลาดของโปรแกรม											/	/				
7	ทำปริญญานิพนธ์และแก้ไข													/	/	/	

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ระบบการจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP เพื่อใช้ในการศึกษาความเป็นไปได้ของผลกระทบในกรณีที่อุปสงค์และอุปทานมีความไม่สมดุล โดยผลของการจำลองนี้สามารถนำไปใช้ในการวางแผนป้องกันการหยุดชะงักของกระบวนการผลิตไฟฟ้าที่อาจเกิดขึ้นได้ หรือเหตุการณ์ที่เครื่องกำเนิดไฟฟ้าผลิตไฟฟ้ามากเกินไปเกินความต้องการทางไฟฟ้า ทำให้เกิดการสิ้นเปลือง
2. สามารถจัดลำดับความสำคัญในระบบส่งจ่ายของโรงไฟฟ้าไปยังลูกค้าแต่ละรายได้

## บทที่ 2

### รายละเอียดของกรณีศึกษา

#### 2.1 กล่าวนำ

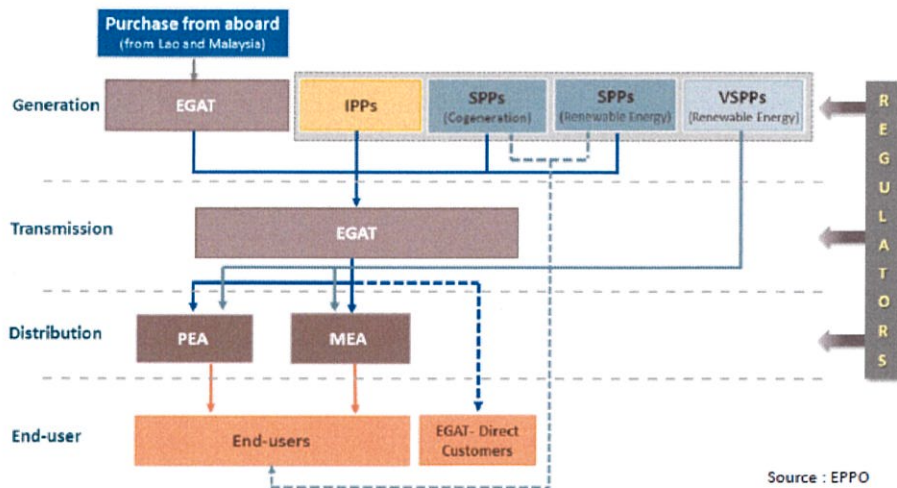
บทนี้เป็นการอธิบายถึงทฤษฎีและหลักการที่เกี่ยวข้องกับระบบ PMC ที่นำมาประยุกต์ในการใช้สำหรับการจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP และ ดีซีเอสของ Foxboro ที่ใช้ในกรณีศึกษา โดยมีรายละเอียด ดังนี้

#### 2.2 ทฤษฎีที่เกี่ยวข้อง

##### 2.2.1 ธุรกิจผลิตไฟฟ้า

ธุรกิจผลิตไฟฟ้ามีลักษณะพิเศษเฉพาะที่สำคัญ คือ

- 1) ไฟฟ้าที่ผลิตได้ไม่สามารถเก็บเป็นสต็อกเหมือนสินค้าอื่น จำเป็นต้องส่งไปยังลูกค้าหรือผู้ใช้ไฟฟ้าทันที โดยผ่านระบบสายส่ง (Transmission System) และระบบจำหน่าย (Distribution System)
  - 2) การเพิ่มกำลังการผลิตไฟฟ้าไม่สามารถทำได้ในระยะเวลาอันสั้น เนื่องจากการก่อสร้างโรงไฟฟ้าต้องใช้เวลา 5-7 ปี ดังนั้นในการลงทุนจะต้องมีการคาดการณ์ความต้องการใช้ไฟฟ้าวล่วงหน้า
  - 3) ไฟฟ้าเป็นระบบสาธารณูปโภคที่จำเป็น ซึ่งทางการมีบทบาทสูง ทั้งในด้านการผลิต การจำหน่าย และเป็นผู้กำกับดูแลทั้งในการกำหนดราคาและการวางแผนการลงทุนในธุรกิจ
- โครงสร้างธุรกิจผลิตไฟฟ้าของไทย เป็นระบบที่รัฐเป็นผู้ซื้อรายเดียว (Enhanced Single Buyer Model) เนื่องจาก EGAT ผู้ขาดระบบสายส่งไฟฟ้า โดยมีการไฟฟ้านครหลวง (MEA) และการไฟฟ้าส่วนภูมิภาค (PEA) รับซื้อไฟฟ้าบางส่วนและทำหน้าที่จำหน่ายไฟฟ้าไปยังผู้ใช้ไฟฟ้า
  - EGAT มีอำนาจในการควบคุมธุรกิจ เนื่องจากมีบทบาทร่วมกับรัฐในการกำหนดแผนพัฒนากำลังการผลิตไฟฟ้าของประเทศ หรือ PDP (Power Development Plan) และแผนพัฒนาพลังงานทดแทนและพลังงานทางเลือก หรือ AEDP (Alternative Energy Development Plan) ซึ่งเป็นแผนการลงทุนในระยะยาวของธุรกิจนี้ในไทย (ปัจจุบันอยู่ระหว่างการดำเนินการตามแผน PDP 2558 และ AEDP 2558 ระหว่างปี 2558-2579) นอกจากนี้ EGAT ยังทำหน้าที่เป็นผู้ตรวจสอบข้อเสนอด้านเทคนิคของโรงไฟฟ้าเอกชนที่จะเข้าระบบด้วย



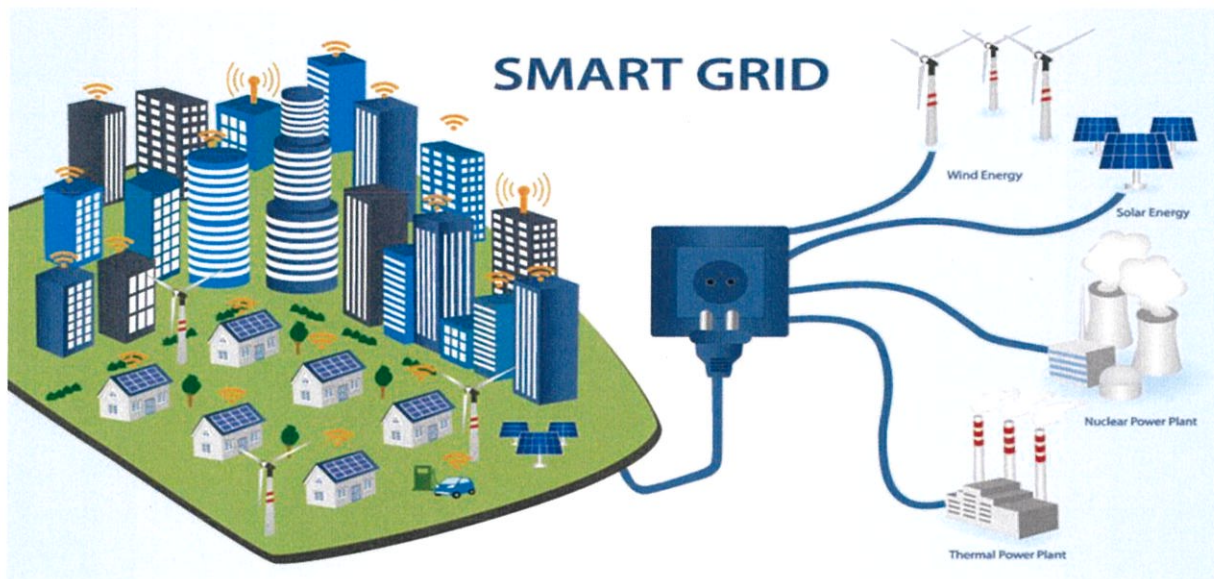
รูปที่ 2.1 Thailand's Electricity Supply Industry Structure

- ในส่วนของผู้ผลิตไฟฟ้าเอกชนแบ่งได้เป็น 3 กลุ่ม ตามกำลังการผลิตติดตั้ง

  - 1) ผู้ผลิตไฟฟ้ารายใหญ่ (Independent Power Producer: IPP) เป็นโรงไฟฟ้าที่มีกำลังการผลิตติดตั้งมากกว่า 90 เมกะวัตต์ ซึ่งใช้เชื้อเพลิงจากก๊าซธรรมชาติและถ่านหินเป็นหลัก กลุ่มนี้ จะมีความเสี่ยงในการรับรู้รายได้ต่ำ เนื่องจากเป็นโรงไฟฟ้าฐาน (Base Load Plant) ที่มีสัญญาซื้อขายไฟฟ้าระยะยาวกับ EGAT โดยมีรายได้จาก 2 ทาง คือ จากปริมาณการจ่ายไฟฟ้าเข้าระบบจริงตามการใช้ของผู้บริโภค และจากรายได้ขั้นต่ำที่จะได้รับตามที่กำหนดในสัญญาขายไฟฟ้าระยะยาวกับ EGAT (Minimum Take)
  - 2) ผู้ผลิตไฟฟ้ารายเล็ก (Small Power Producer: SPP) เป็นโรงไฟฟ้าที่มีกำลังการผลิตติดตั้ง 10-90 เมกะวัตต์ มีทั้งที่ทำสัญญาขายไฟฟ้ากับ EGAT ระยะเวลาตั้งแต่ 20-25 ปี และขายไฟฟ้าตรงให้กับลูกค้าอุตสาหกรรมในพื้นที่ใกล้เคียง กลุ่มนี้จึงอาจมีความเสี่ยงจากความไม่แน่นอนของรายได้ โดยเฉพาะในส่วนของ การขายไฟฟ้าตรงให้กับภาคอุตสาหกรรม ซึ่งภาวะจะผันผวนตามสถานะเศรษฐกิจและทิศทางการอุตสาหกรรมของลูกค้า
  - 3) ผู้ผลิตไฟฟ้ารายเล็กมาก (Very Small Power Producer: VSPP) เป็นโรงไฟฟ้าที่มีกำลังการผลิตติดตั้งน้อยกว่า 10 เมกะวัตต์ มักเป็นการผลิตไฟฟ้าจากวัสดุเหลือใช้ในโรงงานแปรรูปเกษตรเพื่อใช้เองและส่วนที่เหลือจะขายไฟฟ้าให้กับ MEA และ PEA ในอัตรารับซื้อไฟฟ้าระบบ Feed-in Tariff (FIT) ตลอดอายุโครงการตามประเภทเชื้อเพลิง

## 2.2.2 ระบบโครงข่ายไฟฟ้าอัจฉริยะ

ระบบโครงข่ายไฟฟ้าอัจฉริยะ (Smart Grid) คือ การพัฒนาให้ระบบไฟฟ้าสามารถตอบสนองต่อการทำงานได้อย่างชาญฉลาดมากขึ้น หรือมีความสามารถมากขึ้นโดยใช้ทรัพยากรที่น้อยลงอย่างมีประสิทธิภาพ มีความน่าเชื่อถือ มีความปลอดภัย มีความยั่งยืน และเป็นมิตรกับสิ่งแวดล้อม ซึ่งสามารถทำให้เกิดขึ้นได้โดยการประยุกต์ใช้เทคโนโลยีระบบสื่อสารสารสนเทศ (ICT) ระบบเซ็นเซอร์ ระบบเก็บข้อมูล และเทคโนโลยีทางการควบคุมอัตโนมัติเพื่อให้ระบบไฟฟ้ากำลังสามารถรับรู้ข้อมูลสถานะต่างๆ ในระบบมากขึ้นเพื่อใช้ในการตัดสินใจอย่างอัตโนมัติ ทั้งนี้ กระบวนการเหล่านี้จะต้องเกิดขึ้นทั่วทั้งระบบไฟฟ้าครอบคลุม ระบบผลิต ระบบส่ง ระบบจำหน่าย และระบบผู้ใช้ไฟฟ้า

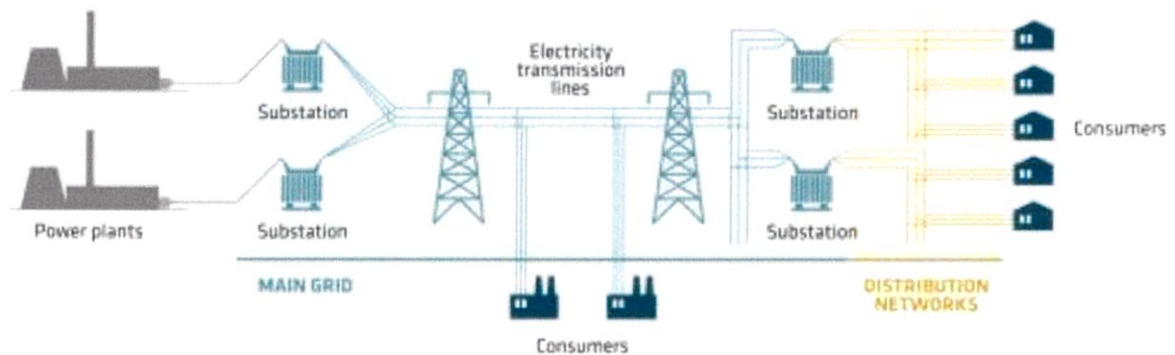


รูปที่ 2.2 SMART GRID

ระบบสมาร์ทกริดมิได้หมายถึงเทคโนโลยีใดเทคโนโลยีหนึ่งโดยเฉพาะ แต่ประกอบด้วยเทคโนโลยีหลากหลายประเภทซึ่งเข้ามาทำงานร่วมกัน โดยครอบคลุมตั้งแต่การประยุกต์ใช้งานเทคโนโลยีเหล่านั้น ตลอดทั้งห่วงโซ่ของระบบไฟฟ้า ตั้งแต่การผลิตไฟฟ้า การส่งไฟฟ้าการจำหน่ายไฟฟ้า ไปจนถึงภาคส่วนของผู้บริโภค นั่นคือ ผู้ใช้ไฟฟ้าทั่วไป ตั้งแต่ภาคบ้านเรือน ภาคอุตสาหกรรมภาคธุรกิจและการพาณิชย์ เป็นต้น นอกจากนี้บริบทของระบบสมาร์ทกริดยังครอบคลุมไปถึงภาคการขนส่งด้วย เช่น เพื่อให้สามารถรองรับการนำรถยนต์ไฟฟ้าเข้ามาใช้งาน เป็นต้น

ระบบไฟฟ้าในปัจจุบันได้ถูกออกแบบและใช้งานมาเป็นระยะเวลายาวนานในอดีต หน้าที่หลักของระบบไฟฟ้าคือการส่งไฟฟ้าจากแหล่งผลิตไปยังผู้ใช้ไฟฟ้า ดังนั้น ภาพของระบบไฟฟ้าสามารถพิจารณาแยกออกได้เป็น 2 ส่วนหลัก ๆ ได้แก่ ภาคอุปทาน (Supply Side) และภาคอุปสงค์ (Demand) สำหรับภาคอุปทานนั้น

ประกอบไปด้วย ระบบผลิตไฟฟ้า (Generation) ได้แก่ โรงไฟฟ้าหรือระบบผลิตไฟฟ้าต่าง ๆ ซึ่งทำหน้าที่ผลิตไฟฟ้าเพื่อจ่ายเข้าสู่ระบบ ไฟฟ้าที่ผลิตขึ้นมาจะถูกแปลงให้มีความดันไฟฟ้าสูงขึ้นก่อนที่จะจ่ายเข้าสู่ระบบส่งไฟฟ้า (Transmission) โดยสาเหตุที่ต้องแปลงความดันให้สูงขึ้นนั้นเพื่อลดความสูญเสียในการส่งไฟฟ้าเป็นระยะทางไกล ระบบส่งไฟฟ้าที่มองเห็นได้ชัดเจนคือสายส่งไฟฟ้าแรงสูงที่ทำหน้าที่ส่งไฟฟ้าข้ามจังหวัด หรือข้ามประเทศเป็นระยะทางไกล เมื่อไฟฟ้าถูกส่งไปยังบริเวณที่มีผู้ใช้ไฟฟ้าอยู่ ความดันไฟฟ้าจะต้องแปลงให้ต่ำลง ทั้งนี้ เพื่อความปลอดภัยของผู้ใช้ไฟฟ้า ดังนั้นความดันไฟฟ้าจะถูกแปลงให้ต่ำลงแล้วจ่ายเข้าสู่ระบบจำหน่าย (Distribution) ระบบจำหน่ายไฟฟ้านั้นทำหน้าที่ส่งไฟฟ้าไปยังผู้ใช้ไฟฟ้าที่กระจายตัวอยู่ในพื้นที่นั้น ๆ อย่างทั่วถึง



รูปที่ 2.3 ระบบไฟฟ้า (Power System)

ระบบไฟฟ้างดเดิมได้ถูกออกแบบเพื่อให้การไหลของกระแสไฟฟ้าเป็นไปในทิศทางเดียวจากโรงไฟฟ้าขนาดใหญ่ไปยังผู้ใช้ไฟฟ้า (เรียกว่าระบบไฟฟ้าแบบรวมศูนย์ หรือ Centralized Power System) โดยกระบวนการทั้งหมดจะเริ่มต้นจากการผลิตไฟฟ้าที่โรงไฟฟ้าขนาดใหญ่ซึ่งมีขนาดพิกัดระดับร้อยเมกะวัตต์ ไฟฟ้าที่ผลิตได้จะถูกแปลงความดันไฟฟ้าให้สูงขึ้นเพื่อส่งเข้าสู่ระบบส่งไฟฟ้า (Transmission network) เพื่อส่งไฟฟ้าในระยะทางไกล เมื่อใกล้ถึงปลายทางไฟฟ้าจะถูกแปลงให้มีแรงดันต่ำลงเนื่องจากเหตุผลด้านความปลอดภัยและจ่ายเข้าสู่ระบบจำหน่ายไฟฟ้า (Distribution network) ซึ่งจะนำไฟฟ้าไปส่งต่อให้กับผู้ใช้ไฟฟ้าต่อไป ซึ่งโดยทั่วไปแล้ว การผลิตไฟฟ้าส่วนมากจะมาจากโรงไฟฟ้าที่ใช้เชื้อเพลิงฟอสซิล เช่น ถ่านหิน ก๊าซธรรมชาติ นอกจากนี้ บางส่วนอาจจะมาจาก พลังงานน้ำ หรือพลังงานนิวเคลียร์ เป็นต้น โดยแหล่งผลิตไฟฟ้างดกล่าวถึงได้ว่าเป็นการผลิตไฟฟ้าที่สามารถควบคุมกำลังไฟฟ้าได้อย่างแน่นอนและมีความต่อเนื่องในกระบวนการผลิต

ผู้ใช้ไฟฟ้าต้องการใช้ไฟฟ้าในช่วงเวลาที่สามารถใช้ได้อย่างอิสระ ดังนั้น การผลิตไฟฟ้าจึงต้องอ้างอิงตามความต้องการไฟฟ้าในช่วงเวลานั้น ๆ เป็นหลัก เรียกได้ว่าเป็นการวางแผนและการควบคุมระบบโครงข่ายไฟฟ้านั้นต้องดำเนินการตามความต้องการไฟฟ้าเป็นหลัก

แม้ระบบไฟฟ้าแบบดั้งเดิมยังคงสามารถใช้งานได้โดยที่ไม่ก่อให้เกิดปัญหาในการจ่ายไฟฟ้า แต่เนื่องด้วยการพัฒนาเทคโนโลยีที่เป็นไปอย่างต่อเนื่อง ทำให้มีเกิดศักยภาพในการปรับปรุงและเพิ่มประสิทธิภาพในการปฏิบัติงานของระบบไฟฟ้า ปัญหาของระบบไฟฟ้าในปัจจุบัน การเข้ามาของพลังงานหมุนเวียน เช่น พลังงานลมและแสงอาทิตย์ ซึ่งกำลังการผลิตไฟฟ้าของแหล่งดังกล่าวในช่วงเวลาต่าง ๆ

### ประโยชน์ของระบบสมาร์ทกริด

เนื่องจากเทคโนโลยีของระบบสมาร์ทกริดมีอย่างมากมายหลากหลาย ครอบคลุมตั้งแต่ระบบผลิตไฟฟ้าไปจนถึงภาคผู้ใช้ไฟฟ้า ประโยชน์ของระบบสมาร์ทกริดจึงมีมากมายหลายด้าน ประโยชน์ที่สำคัญของระบบสมาร์ทกริดสามารถสรุปได้ดังต่อไปนี้

#### 1. การรวมพลังงานหมุนเวียนเข้าในระบบไฟฟ้า (Renewable Energy Integration)

พลังงานหมุนเวียนมีประโยชน์อย่างมากในการนำไปสู่ระบบพลังงานที่ยั่งยืนและเป็นมิตรต่อสิ่งแวดล้อม อย่างไรก็ตาม พลังงานหมุนเวียนแต่ละประเภทก็มีลักษณะเฉพาะตัวที่แตกต่างกันออกไป พลังงานหมุนเวียนบางประเภทอาศัยแหล่งพลังงานจากธรรมชาติเป็นตัวขับเคลื่อนการแปรรูปพลังงานออกมาในรูปแบบที่สามารถใช้งานได้ เช่น พลังงานแสงอาทิตย์ พลังงานลม เป็นต้น ซึ่งแหล่งพลังงานหมุนเวียนในลักษณะนี้เรียกว่าแหล่งพลังงานหมุนเวียนที่มีความผันผวน (Variable Renewable Energy: vRE)

ระบบไฟฟ้าในปัจจุบันอาศัยการบริหารจัดการให้เกิดสมดุลระหว่างการใช้ไฟฟ้ากับการผลิตไฟฟ้าตลอดเวลา นั่นคือ เมื่อใดที่มีความต้องการไฟฟ้าสูง ก็ต้องมีการผลิตไฟฟ้าสูงขึ้นตามไปด้วย ในขณะเดียวกัน เมื่อใดที่มีความต้องการไฟฟ้าน้อยลง ก็ต้องลดกำลังการผลิตไฟฟ้าตามไปด้วย ในอดีตนั้น โรงไฟฟ้าที่ทำหน้าที่ผลิตไฟฟ้าจ่ายเข้าสู่ระบบไฟฟ้าเป็นโรงไฟฟ้าที่อาศัยเชื้อเพลิงฟอสซิลเป็นหลัก ดังนั้นผู้บริหารงานระบบโครงข่ายไฟฟ้าจึงสามารถควบคุมการผลิตไฟฟ้าในช่วงเวลาต่างๆ ได้ตามต้องการ อย่างไรก็ตาม เมื่อแหล่งผลิตไฟฟ้าพลังงานหมุนเวียนที่มีความผันผวนเข้ามาต่อเชื่อมกับระบบในปริมาณที่มากขึ้น จะส่งผลให้การควบคุมภาคการผลิตไฟฟ้าเป็นไปได้น้อยลง รวมถึงหากไม่มีการบริหารจัดการหรือวางแผนที่มีประสิทธิภาพแล้ว ความผันผวนที่เกิดขึ้นจากแหล่งพลังงานหมุนเวียนเหล่านี้จะส่งผลให้ระบบไฟฟ้ามีปัญหาเชิงคุณภาพไฟฟ้าและความน่าเชื่อถือได้

เทคโนโลยีหลายประเภทของระบบสมาร์ทกริดสามารถนำมาประยุกต์ใช้เพื่อแก้ไขข้อจำกัดเหล่านี้ ระบบการพยากรณ์ไฟฟ้าที่ผลิตได้จากพลังงานหมุนเวียนประเภทลมและแสงอาทิตย์สามารถพยากรณ์ล่วงหน้าว่าโรงไฟฟ้าพลังงานลมและแสงอาทิตย์จะมีการผลิตไฟฟ้าในช่วงเวลาต่างๆ ได้มากน้อยเพียงใด ซึ่งเป็นประโยชน์ต่อผู้บริหารงานโครงข่ายไฟฟ้า การพยากรณ์ดังกล่าวนี้จำเป็นต้องอาศัยข้อมูลจากหลายส่วน ไม่ว่าจะเป็นข้อมูลด้านสภาพอากาศรวมถึงการพยากรณ์สภาพอากาศในอนาคต ข้อมูลสถานะการปฏิบัติงานของ

โรงไฟฟ้าต่างๆ และการอาศัยข้อมูลสถิติต่างๆ ในอดีตมาใช้ในการประมวลผล ซึ่งระบบสมาร์ทกริดจะช่วยให้การแลกเปลี่ยนข้อมูลเหล่านี้เป็นไปอย่างมีประสิทธิภาพ

## 2. การลดความต้องการไฟฟ้าสูงสุด (Peak Reduction)

ความต้องการไฟฟ้าสูงสุดของประเทศไทยนั้นมักจะเกิดขึ้นในช่วงเดือนเมษายนซึ่งเป็นฤดูร้อนและเป็นช่วงที่มีอุณหภูมิสูงที่สุด ส่งผลให้การใช้เครื่องปรับอากาศมีปริมาณมากขึ้นในช่วงเวลาดังกล่าว ความต้องการไฟฟ้าสูงสุดนั้นมักจะเกิดขึ้นเพียงช่วงเวลาสั้นๆ ในแต่ละปี อย่างไรก็ตามภาคการผลิตไฟฟ้าจำเป็นต้องจัดเตรียมกำลังการผลิตไฟฟ้าให้เพียงพอสามารถรองรับความต้องการไฟฟ้าสูงสุดได้

สำหรับระบบไฟฟ้าในปัจจุบัน โรงไฟฟ้าแต่ละประเภทที่จ่ายไฟฟ้าเข้าสู่ระบบมีลักษณะเฉพาะตัวที่แตกต่างกันออกไป โรงไฟฟ้าบางประเภทมีอัตราการปรับเปลี่ยนการผลิตไฟฟ้าต่ำ นั่นคือ ใช้เวลานานในการเพิ่มหรือลดการผลิตไฟฟ้า เช่น โรงไฟฟ้าถ่านหิน โรงไฟฟ้าพลังงานิวเคลียร์ เป็นต้น โรงไฟฟ้าเหล่านี้มักจะถูกใช้เป็นโรงไฟฟ้าฐาน (Baseload Power Plant) ซึ่งจะมีการผลิตไฟฟ้าที่ค่อนข้างคงที่ในช่วงระยะเวลาอันยาวนาน สำหรับโรงไฟฟ้าอีกประเภทหนึ่งมีความยืดหยุ่นในการเพิ่มหรือลดกำลังการผลิต ตัวอย่างระบบผลิตไฟฟ้าประเภทนี้ที่สามารถเห็นได้อย่างชัดเจน คือ เครื่องกำเนิดไฟฟ้าเชื้อเพลิงน้ำมันดีเซล ซึ่งติดตั้งตามอาคารใหญ่ๆ เช่น ห้างสรรพสินค้า โรงแรม เป็นต้น หรือตามสถานที่ที่การใช้ไฟฟ้ามีความสำคัญสูงในการปฏิบัติงาน เช่น ท่าอากาศยาน โรงพยาบาล เป็นต้น โดยทั่วไปแล้วเครื่องกำเนิดไฟฟ้าเชื้อเพลิงน้ำมันดีเซลสามารถเริ่มเดินเครื่องและเร่งกำลังการผลิตขึ้นสูงสุดได้ในระยะเวลาประมาณแค่ 15 วินาที สำหรับโรงไฟฟ้าที่มีความยืดหยุ่นในการปฏิบัติงานสูงนั้น เช่น โรงไฟฟ้าพลังน้ำ โรงไฟฟ้ากังหันก๊าซซึ่งใช้ก๊าซธรรมชาติเป็นเชื้อเพลิง โรงไฟฟ้าเชื้อเพลิงน้ำมันดีเซล เป็นต้น โรงไฟฟ้าเหล่านี้มักจะถูกนำมาใช้เป็นโรงไฟฟ้าสำหรับช่วง Peak (Peaking Plant)

หากค่าความต้องการไฟฟ้าสูงสุดเพิ่มขึ้นอย่างต่อเนื่องจะส่งผลให้ต้องมีการก่อสร้างโรงไฟฟ้า Peaking Plant มากขึ้น ซึ่งโรงไฟฟ้าเหล่านี้มีการปฏิบัติงานในช่วงเวลาที่ไม่มากในแต่ละปี ส่งผลให้ค่าใช้จ่ายในการผลิตไฟฟ้าต่อหน่วยสูงกว่าโรงไฟฟ้าฐานเป็นอย่างมาก

ระบบสมาร์ทกริดสามารถนำมาประยุกต์ใช้ในการลดค่าความต้องการไฟฟ้าสูงสุดได้ ระบบการตอบสนองด้านโหลด (Demand Response) แบบอัตโนมัติสามารถสื่อสารและสั่งการให้ภาคผู้ใช้ไฟฟ้าบางส่วนลดการใช้ไฟฟ้าลงในช่วงเวลาที่ต้องการได้ ทั้งนี้ ต้องอาศัยกลไกราคาและสิ่งจูงใจเข้ามามีส่วนร่วมในการดำเนินการ การตอบสนองด้านโหลดนั้นแม้จะสามารถดำเนินการได้ทันทีโดยไม่ต้องอาศัยระบบสมาร์ทกริดเข้ามาเกี่ยวข้อง แต่ระบบสมาร์ทกริดสามารถช่วยให้การแลกเปลี่ยนข้อมูล การสั่งการ เป็นไปอย่างมีประสิทธิภาพมากขึ้น นอกนี้ระบบบริหารจัดการพลังงานสามารถนำมาประยุกต์ใช้ตามบ้านเรือน อาคาร หรือโรงงานต่างๆ สามารถรองรับการตอบสนองด้านโหลดแบบอัตโนมัติได้ ซึ่งสามารถทำให้ผู้ใช้ไฟฟ้าเหล่านั้นสามารถตอบสนองหรือปรับเปลี่ยนการใช้ไฟฟ้าในแต่ละช่วงเวลาได้อย่างสะดวก

## 3. การเพิ่มประสิทธิภาพในการบริหารงานระบบโครงข่ายไฟฟ้า

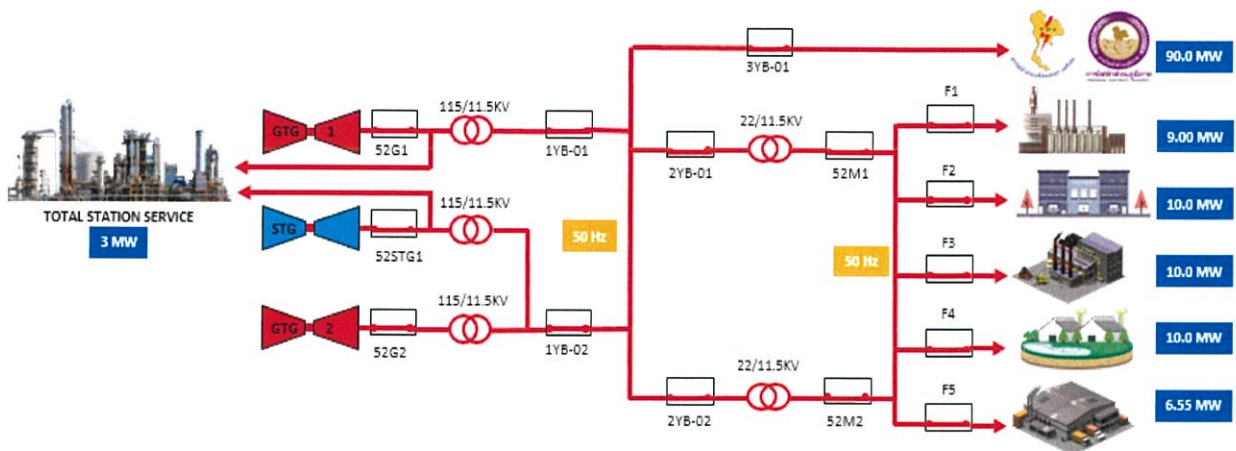
ระบบสมรรถกฤตสามารถทำให้การบริหารจัดการระบบโครงข่ายไฟฟ้าเป็นไปได้โดยมีประสิทธิภาพมากขึ้น เช่น การตรวจจับความผิดปกติขึ้นในระบบไฟฟ้าอย่างอัตโนมัติ (Fault Detection) และการตรวจหาความผิดปกติเหล่านั้นมีความแม่นยำและชัดเจนมากขึ้น เป็นต้น ซึ่งจะทำให้ผู้ปฏิบัติงานโครงข่ายสามารถตอบสนองต่อความผิดปกติเหล่านั้นได้อย่างรวดเร็วกว่าเดิม นอกจากนี้ระบบสมรรถกฤตที่มีความก้าวหน้ามากขึ้นอาจจะสามารถตอบสนองต่อความผิดปกติอื่นๆ โดยอัตโนมัติ เรียกว่า การรักษาเยียวยาตนเอง (Self-healing)

#### 4. เพิ่มการรับรู้ข้อมูลที่เกิดขึ้นในระบบไฟฟ้า

เมื่อเทคโนโลยีระบบสมรรถกฤตถูกนำมาใช้งานมากขึ้นจะทำให้เกิดการรับรู้ข้อมูลต่างๆ ในปริมาณมากขึ้น ข้อมูลบางประเภทนั้นผู้ใช้ไฟฟ้าอาจจะไม่เคยตระหนักรับรู้มาก่อน แต่เดิมนั้นผู้ใช้ไฟฟ้าทราบปริมาณการใช้ไฟฟ้าของตนเองภายในเดือนนั้นๆ โดยทราบจากการจดมิเตอร์ไฟฟ้าที่หมุนไปในรอบเดือนนั้น หรือจากใบแจ้งหนี้ค่าไฟฟ้าที่ได้รับ ในอนาคต เมื่อสมาร์ทมิเตอร์ (Smart Meter) ถูกติดตั้งเข้าไปจะทำให้ผู้ใช้ไฟฟ้าสามารถทราบปริมาณการใช้ไฟฟ้าในช่วงเวลาต่างๆ ได้ละเอียดมากขึ้น ในบางภาคส่วนการรับรู้ข้อมูลเพิ่มเติมเหล่านี้ได้มีบ้างแล้ว เช่น ภาคอุตสาหกรรมที่มีการติดตั้งระบบอ่านมิเตอร์อัตโนมัติ (Automatic Meter Reading: AMR) เป็นต้น

หากมีสภาพแวดล้อมที่เหมาะสม การรับรู้ข้อมูลที่มากขึ้นจะมีศักยภาพในการโน้มนำให้ผู้ใช้ไฟฟ้าเปลี่ยนพฤติกรรมการใช้ไฟฟ้าของตนเองไปสู่รูปแบบที่มีประสิทธิภาพมากขึ้น เช่น หากในอนาคตมีการกำหนดอัตราค่าไฟฟ้าที่ต่างกันในแต่ละช่วงเวลา โดยให้ค่าไฟฟ้าสูงในช่วงที่มีความต้องการใช้ไฟฟ้าสูง เมื่อผู้ใช้ไฟฟ้าทราบว่าตนเองใช้ไฟฟ้าปริมาณมากในช่วงที่ค่าไฟฟ้ามีราคาสูงก็จะเกิดการพิจารณาทบทวน และอาจจะปรับเปลี่ยนการใช้งานอุปกรณ์บางอย่างที่ไม่จำเป็นในช่วงเวลานั้นไปในช่วงเวลาที่ค่าไฟฟ้าต่ำแทน เช่น การใช้เครื่องซักผ้า เครื่องล้างจาน เป็นต้น

## 2.2.3 ระบบควบคุมการผลิตและการจ่ายพลังงานไฟฟ้าที่เป็นกรณีศึกษา (Plant Management Control)



รูปที่ 2.4 ระบบควบคุมการผลิตและการจ่ายพลังงานไฟฟ้า

จากรูป ระบบควบคุมการผลิตและการจ่ายพลังงานไฟฟ้า (Plant Management Control) นั้น ประกอบไปด้วยฝั่ง Generator และ Load ฝั่ง Generator มี Gas Turbine Generator 2 ตัว และ Stream Turbine Generator 1 ตัว ซึ่ง Gas Turbine Generator แทนสัญลักษณ์ด้วย GTG1 และ GTG2 ส่วน Stream Turbine Generator แทนสัญลักษณ์ด้วย STG ส่วนฝั่ง Load มีโรงงานของลูกค้า 5 แห่ง หลักการในการจ่ายพลังงานไปยัง Load นั้น Generator จะจ่ายพลังงานไฟฟ้าให้ใช้ภายในโรงไฟฟ้าของตนเอง, โรงงานของลูกค้า 5 แห่ง และการไฟฟ้าส่วนภูมิภาค (PEA) โดย Generator จะจ่ายพลังงานไฟฟ้าไปยัง Load ได้นั้น ต้องผ่าน Breaker 52G1, 52STG1 และ 52G2 ซึ่งเป็น Breaker ของ Gas Turbine Generator และ Stream Turbine Generator ตามลำดับ ผ่านไปยังหม้อแปลงไฟฟ้า เพื่อแปลงกระแสไฟฟ้า 115/11.5 KV และไปยังสายส่งทั้งสองสายผ่าน Breaker 1YB-01 และ 1YB-02 จากนั้นจะส่งผ่านไปยังสายส่งหลักผ่านไปยัง Breaker 3YB-01, 2YB-01 และ 2YB-02 แล้วผ่านหม้อแปลงไฟฟ้าเพื่อแปลงกระแสไฟฟ้า 22/11.5 KV ผ่านเข้า Breaker 52M1 และ 52M2 สุดท้ายส่งไปยัง Breaker ของโรงงานลูกค้า 5 แห่ง

#### 2.2.4 การนำแนวคิด PMC มาประยุกต์ใช้ในกรณีศึกษา

ในระบบของ PMC จะเริ่มกระบวนการทำงานก็ต่อเมื่อ 3YB-01 OPEN ระบบจะทำการ Load shed เมื่อค่า Load นั้นมีค่ามากกว่าค่า Generator จะเกิดการ Trip Generator ตาม Priority ที่กำหนด ระบบจะเลิก Trip ก็ต่อเมื่อเรายก Breaker หลังจากทำการ Load shed จากนั้นระบบจะเข้ากระบวนการ Generator shed เมื่อค่า Generator มีค่ามากกว่าค่า Load จะเกิดการ Trip Load ตาม Priority ที่กำหนด จากนั้นระบบจะเข้ากระบวนการของ Frequency control ซึ่งในกระบวนการนี้ หาก Frequency มีค่าอยู่ระหว่าง 49.9 และ 50.1 ระบบจะเข้ากระบวนการ Isochronous Mode

## 2.3 ฮาร์ดแวร์ที่เกี่ยวข้อง

### 1. Workstation Sever – H90



รูปที่ 2.5 H90 Workstation Server

Workstation Sever – H90 เป็นเซิร์ฟเวอร์อเนกประสงค์ที่รองรับการใช้งานสำหรับ Microsoft Windows Server 2008 พร้อมกับซอฟต์แวร์ Foxboro Evo S10 ซึ่งเป็นโฮสต์ของ FCP270s มีการรวบรวมและประมวลผลข้อมูลที่เกี่ยวข้องกับแอปพลิเคชัน อีกทั้งมีความสามารถในการให้บริการไฟล์ การแสดงผลกราฟิก และข้อความ

#### คุณสมบัติ

- High-End Processor Speed ความจำ 8 GB Hard Disk ได้สูงสุด 8 ตัวขึ้นอยู่กับ RAID และมีแหล่งจ่ายไฟสำรอง
- มีความสามารถในการควบคุม Host รองรับการเก็บข้อมูล ตรวจสอบการทำงาน และการแสดงผล
- Serves รองรับ Control Core Services software หรือ the Control Software application platform และ a human interface station
- มีความสามารถในการรองรับการใช้โปรแกรม Control Core Service หรือการใช้ Control Software จากการเข้าใช้งานระยะไกลของเครื่อง Client โดยใช้ LANs
- สามารถ Back up โดยใช้ Symantec System Recovery (SSR) software
- การติดต่อสื่อสารแบบ Client/Server ใช้ TCP/IP Protocol
- แหล่งจ่ายไฟสำรอง 230 Vac

## 2. MONITOR

HP EliteDisplay E241i เป็นจอภาพแบบจอแบน 23 นิ้ว มีขนาดภาพ 1920 x 1080 pixels การแสดงผลใช้ได้ทั้ง Digital DVI-D, Display Port และ Analog VGA video signals จอแสดงผลนี้สามารถใช้กับ workstation ที่มีความละเอียด 1920 x 1080 หรือ workstation ที่มีความละเอียดต่ำกว่า

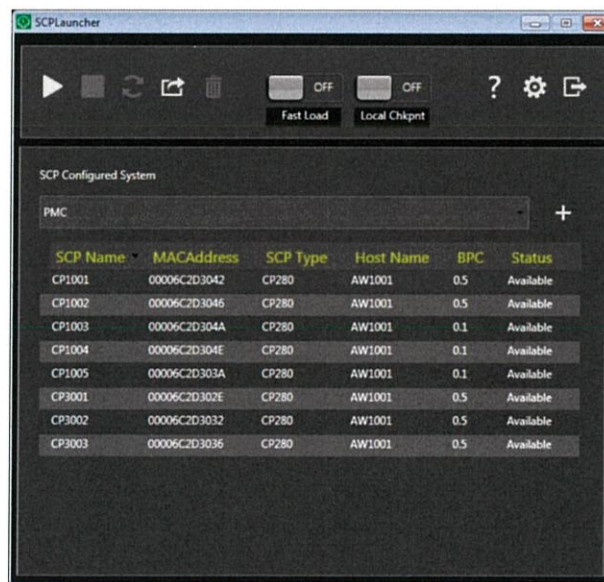


รูปที่ 2.6 HP EliteDisplay E241i, 24" LED Monitor

## 2.4 ซอฟต์แวร์ที่เกี่ยวข้อง

### 1. SCP Launcher

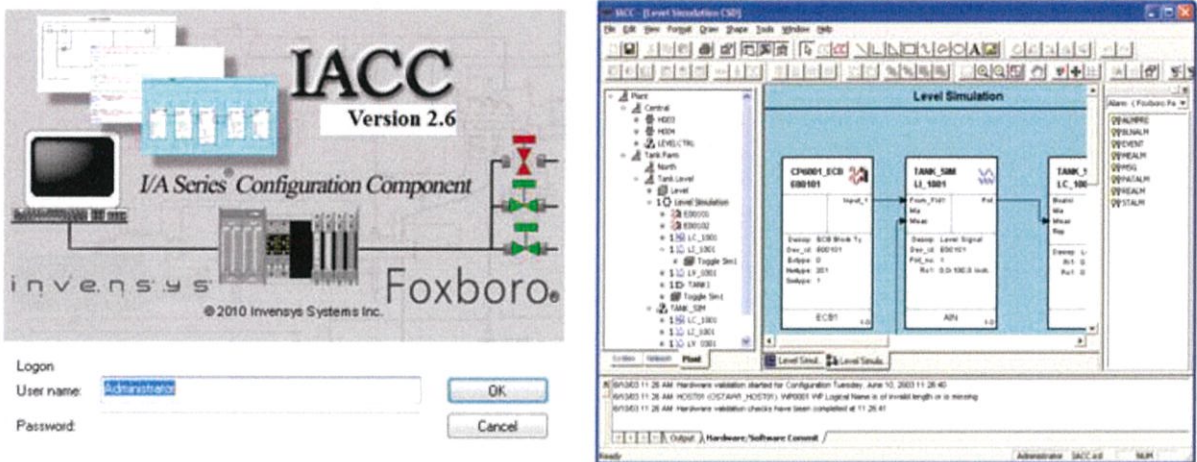
เป็นซอฟต์แวร์ที่ใช้สำหรับจำลองการทำงานของ Field Control Processor



รูปที่ 2.7 SCP Launcher

## 2. I/A Series Configuration Component (IACC)

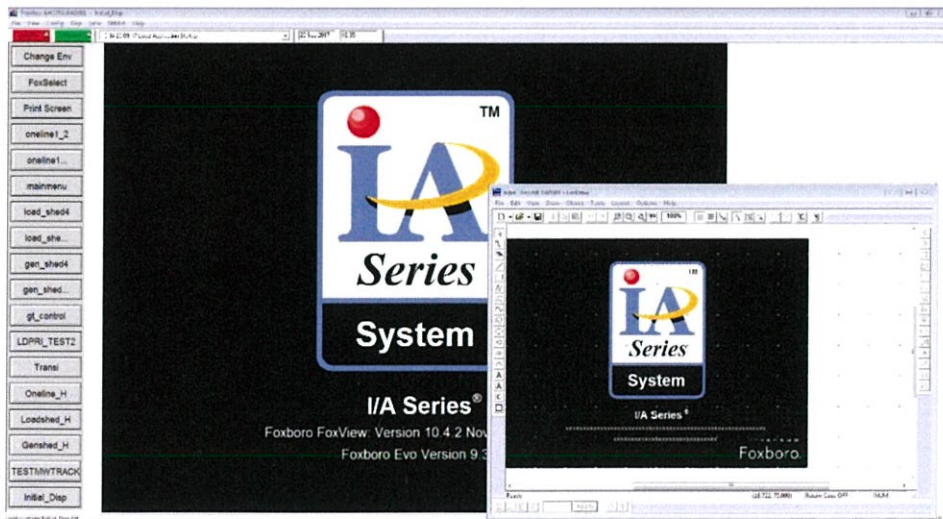
ซอฟต์แวร์ IACC เป็นเครื่องมือที่มีประสิทธิภาพในการควบคุมกระบวนการของ I/A Series ซึ่ง IACC ช่วยลดเวลาในการออกแบบและค่าใช้จ่ายของโครงการโดยใช้เทมเพลตต่างๆ เช่น เทมเพลตกลยุทธ์การควบคุมชุดควบคุมกระบวนการ I/A Series และการกำหนดค่าพรีอเพอร์ดีที่กำหนดไว้ล่วงหน้า ในขณะที่เดียวกันซอฟต์แวร์จะช่วยลดข้อผิดพลาดด้วยเทคนิคคุณภาพ



รูปที่ 2.8 Building an IACC Control Strategy Diagram

## 3. FoxView

FoxView เป็น Foxboro Evo หรือ I/A Series ที่เชื่อมต่อระหว่างผู้ใช้งานกับกระบวนการ ในการควบคุมและแสดงผลการทำงานของระบบ



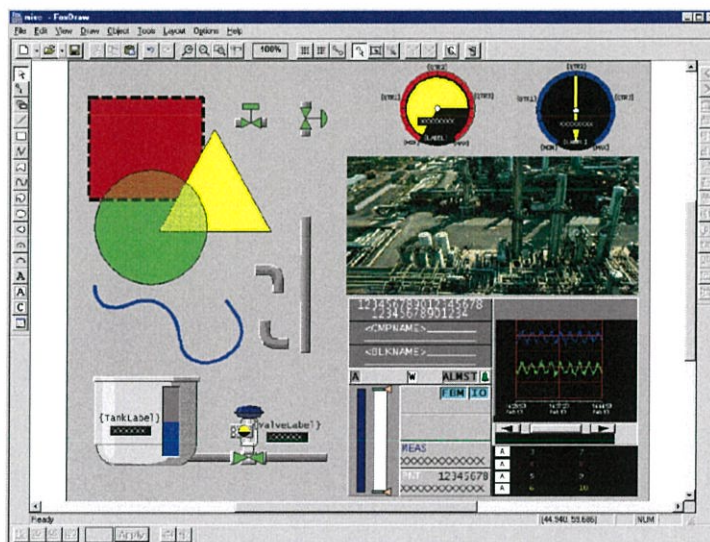
รูปที่ 2.9 FoxView

#### คุณสมบัติ

- สามารถกำหนดการใช้งานได้ตามความต้องการของผู้ใช้งาน
- ดำเนินการทำงานแบบ Real-time และ Historical trending
- เข้าถึงกระบวนการของ process ได้โดยตรงผ่านทางหน้าจอแสดงผล
- ตรวจสอบดูแลและแสดงผลการเตือนภัยโดยใช้ Alarm Manager
- แสดงภาพรวมการทำงานของ Compounds และ Blocks ในการควบคุม Database และ Block detail ผ่านทางหน้าจอแสดงผล
- การเข้าถึงแอปพลิเคชันอื่นๆ (ถ้าได้รับอนุญาต) เช่น FoxDraw™ ( "FoxDraw" ) สำหรับการสร้างและกำหนดค่ากราฟิก, ตัวแก้ไขดีสเพลย์ ( DEdit ), ตัวแปลงจอภาพ ( DConvert ) เป็นต้น

#### 4. FoxDraw

FoxDraw เป็นโปรแกรมที่ใช้ในการสร้าง แก๊ซกราฟิก และกำหนดค่าการแสดงผลของสัญลักษณ์ต่างๆ ที่เราได้จำลองขึ้นจากงานจริง



รูปที่ 2.10 FoxDraw Window

## บทที่ 3

### การจำลองการปรับสมดุลอุปสงค์และอุปทานที่นำเสนอ

#### 3.1 กล่าวนำ

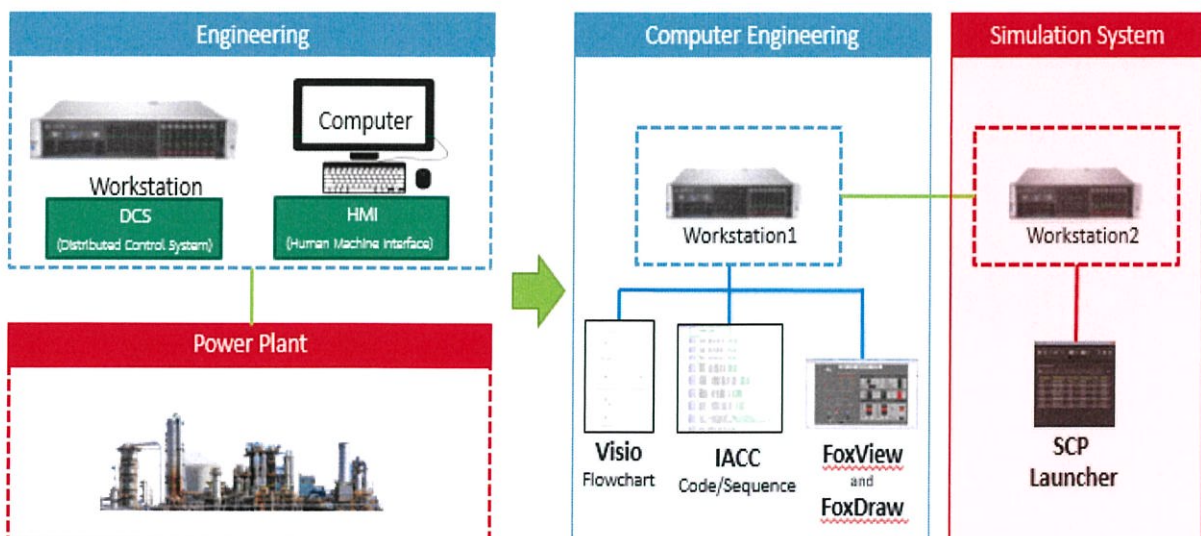
บทนี้เป็นกรอธิบายถึงขั้นตอนการออกแบบการจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP ว่ามีส่วนการออกแบบวางแผน และขั้นตอนการดำเนินการอย่างไร ทั้งนี้จำเป็นที่จะต้องศึกษากระบวนการทำงานของซอฟต์แวร์รวมถึงลำดับขั้นตอนในการติดตั้งซอฟต์แวร์โปรแกรมต่าง ๆ ที่เกี่ยวข้อง และศึกษาฟังก์ชันการทำงานของระบบว่าทำงานอย่างไร โดยระบบนี้ดำเนินการด้วย Foxboro DCS สร้างแก้ไขกราฟิก และควบคุมแสดงผล ผ่านทางโปรแกรม FoxDraw และ FoxView ซึ่งช่วยเพิ่มความสะดวกแก่ผู้ปฏิบัติงาน อีกทั้งยังช่วยเพิ่มประสิทธิภาพของระบบอีกด้วย

#### 3.2 รายละเอียดของระบบ

การจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP สร้างขึ้นเพื่อให้ง่ายต่อการศึกษาโครงสร้างและขั้นตอนการทำงานของ PMC ซึ่งประกอบไปด้วย

##### 3.2.1 โครงสร้างของระบบ

การจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP นี้ ประกอบไปด้วย 2 ส่วนหลัก คือ ส่วนซอฟต์แวร์และฮาร์ดแวร์ ดังรูป

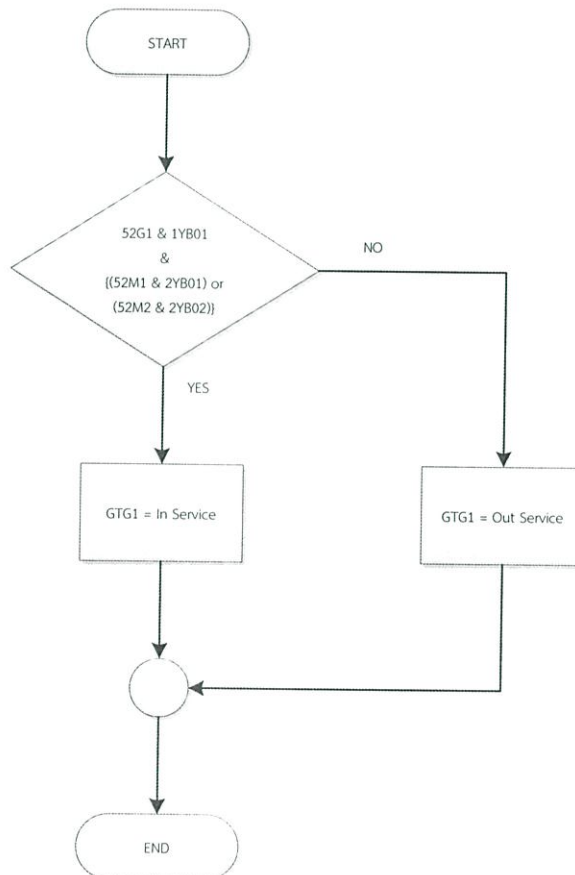


รูปที่ 3.1 โครงสร้างของระบบ

### 3.2.2 Flowchart การทำงาน

Flowchart คือ ผังงานที่แสดงขั้นตอนของคำสั่งการทำงาน เพื่อวางแผนหรือรวบรวมความคิดการเขียนโปรแกรม โปรแกรมจะแสดงลำดับคำสั่งเป็นขั้นตอน (Step By Step) การเขียนผังงานโปรแกรมจะช่วยลดข้อผิดพลาดในการเขียนโปรแกรมลงได้ ทำให้เขียนโปรแกรมง่ายขึ้นและถูกต้อง รวมถึงยังช่วยวิเคราะห์จุดบกพร่องที่เกิดขึ้น

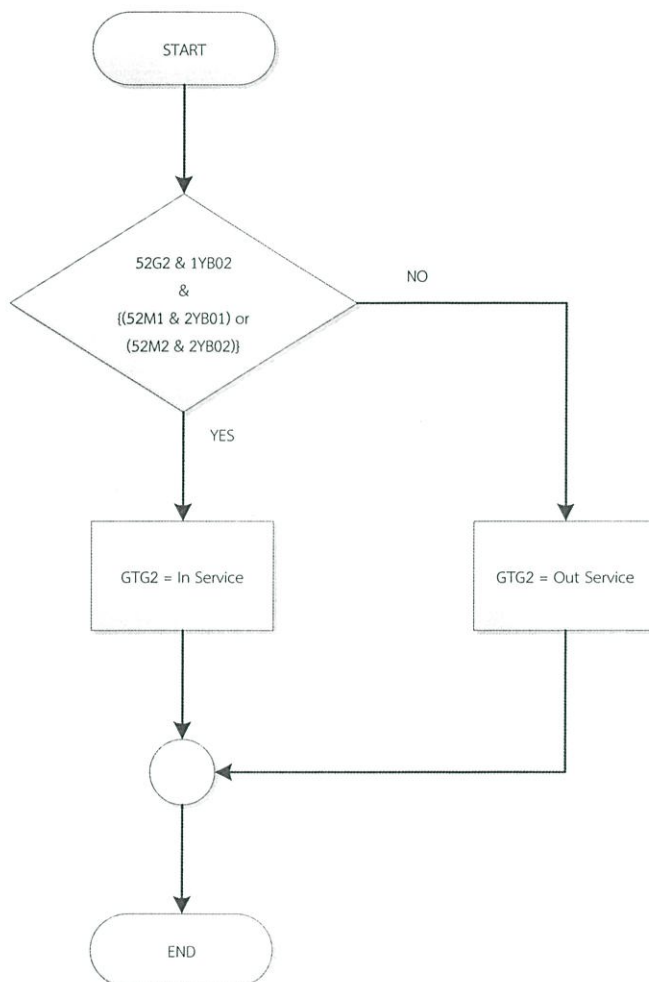
Flowchart IN/OUT Service of GTG 1



รูปที่ 3.2 Flowchart IN/OUT Service of GTG 1

จากรูป เป็น Flowchart ที่แสดงหลักการทำงานของ GTG 1 หรือ Gas Turbine Generator ตัวที่ 1 โดยจะเริ่มการทำงานด้วยการตรวจเช็คเงื่อนไขของระบบ โดยเงื่อนไขที่กำหนด คือ 52G1 & 1YB01 & [52M1 & 2YB01] or [52M2 & 2YB02] หากเข้าเงื่อนไขจะเป็น Yes ส่งผลให้ Gas Turbine Generator ทำงาน (In Service) ถ้าไม่เข้าเงื่อนไขจะเป็น No ส่งผลให้ Gas Turbine Generator ไม่ทำงาน (Out Service)

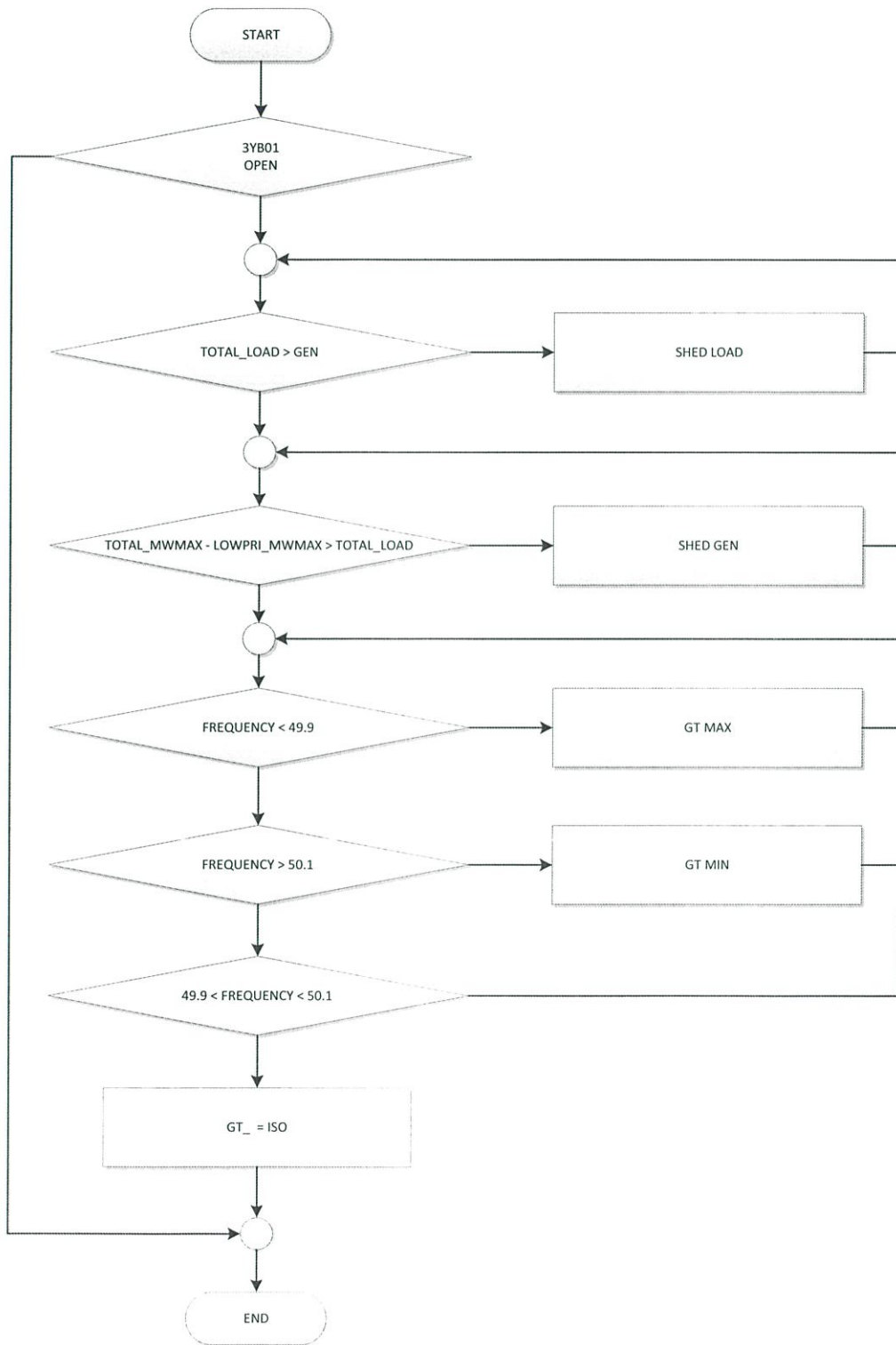
### Flowchart IN/OUT Service of GTG 2



รูปที่ 3.3 Flowchart IN/OUT Service of GTG 2

จากรูป เป็น Flowchart ที่แสดงหลักการทำงานของ GTG 2 หรือ Gas Turbine Generator ตัวที่ 1 โดยจะเริ่มการทำงานด้วยการตรวจเช็คเงื่อนไขของระบบ โดยเงื่อนไขที่กำหนด คือ 52G2 & 1YB02 & [52M1 & 2YB01] or [52M2 & 2YB02] หากเข้าเงื่อนไขจะเป็น Yes ส่งผลให้ Gas Turbine Generator ทำงาน (In Service) ถ้าไม่เข้าเงื่อนไขจะเป็น No ส่งผลให้ Gas Turbine Generator ไม่ทำงาน (Out Service)

# Flowchart PMC



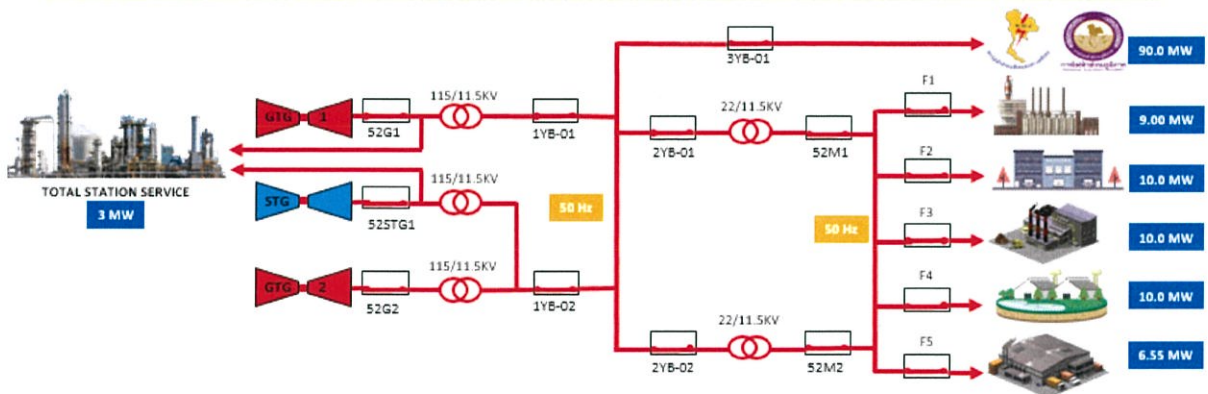
รูปที่ 3.4 Flowchart PMC

ระบบของ PMC จะเริ่มกระบวนการทำงานก็ต่อเมื่อ 3YB-01 OPEN ระบบจะทำการ Load shed เมื่อค่า Load นั้นมีค่ามากกว่าค่า Generator จะเกิดการ Trip Generator ตาม Priority ที่กำหนด ระบบจะเลิก Trip ก็ต่อเมื่อเรายก Breaker หลังจากทำการ Load shed จากนั้นระบบจะเข้ากระบวนการ Generator shed เมื่อค่า Generator มีค่ามากกว่าค่า Load จะเกิดการ Trip Load ตาม Priority ที่กำหนด จากนั้นระบบจะเข้ากระบวนการของ Frequency control ซึ่งในกระบวนการนี้ หาก Frequency มีค่าอยู่ระหว่าง 49.9 และ 50.1 ระบบจะเข้ากระบวนการ Isochronous Mode

### 3.2.3 ตัวอย่างกรณีศึกษา

จากบทที่ 2 การนำแนวคิด PMC มาประยุกต์ใช้ในกรณีศึกษา ดังตัวอย่างกรณีศึกษาต่อไปนี้

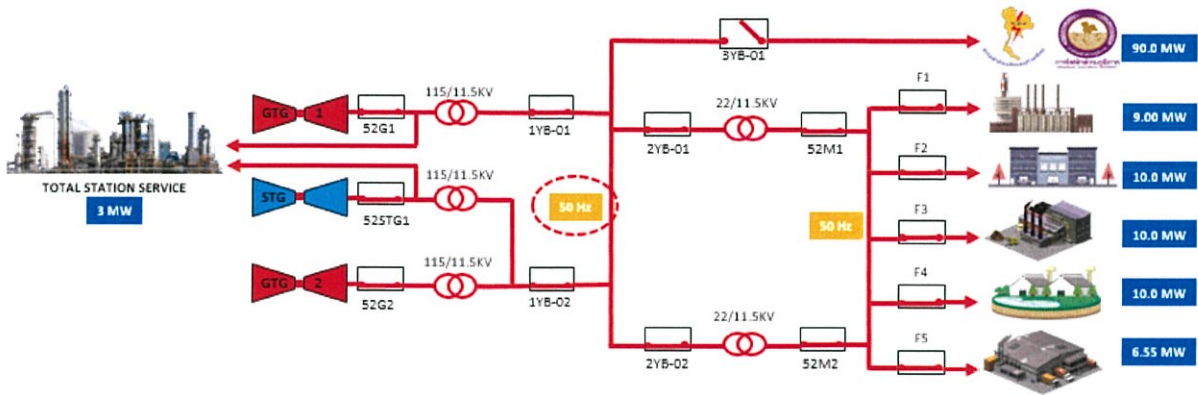
Case No.	Condition	No. of GTGs	No. of STG	Total Load	Total Gen	Scenario
1	Normal Operation	2	1	48.55	42.00	Close 3YB-01



รูปที่ 3.5 ภาพรวมการทำงานของกรณีศึกษาเมื่อ 3YB-01 Close

จากรูปดังกล่าว กำหนดให้ค่า Load มีค่ามากกว่าค่า Generator และ 3YB-01 Close จึงทำให้ระบบยังไม่เริ่มกระบวนการทำงาน

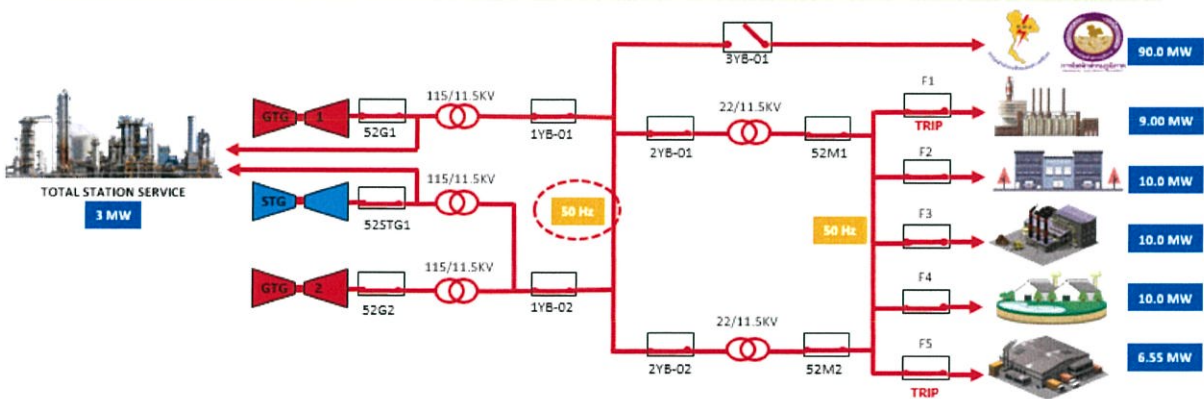
Case No.	Condition	No. of GTGs	No. of STG	Total Load	Total Gen	Scenario
2	Normal Operation	2	1	48.55	42.00	Open 3YB-01



รูปที่ 3.6 ภาพรวมการทำงานของกรณีศึกษาเมื่อ 3YB-01 Open

จากรูป กำหนดให้ 3YB-01 Open ระบบจึงเริ่มกระบวนการทำงาน โดยกระบวนการจะเช็คเงื่อนไขการ Load Shed ก่อนเป็นลำดับแรก

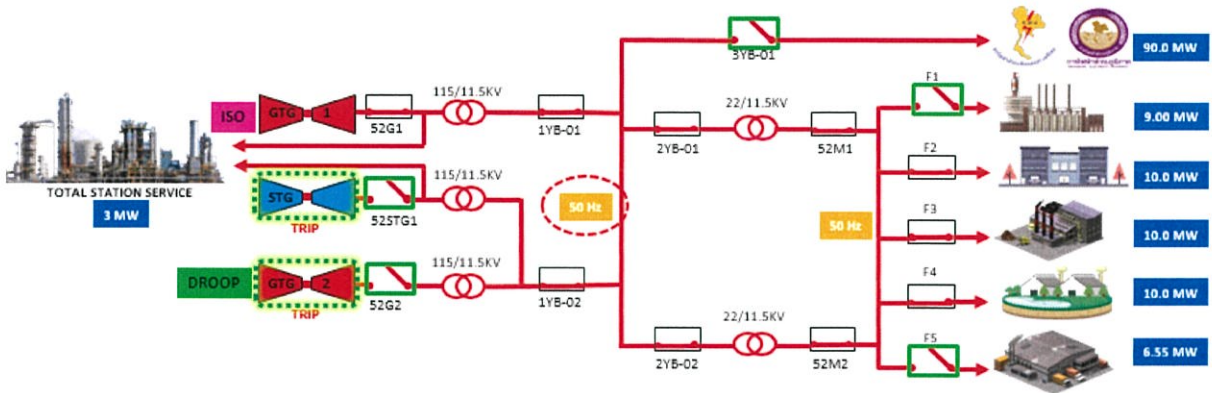
Case No.	Condition	No. of GTGs	No. of STG	Total Load	Total Gen	Scenario
2	Normal Operation	2	1	48.55	42.00	Open 3YB-01



รูปที่ 3.7 ภาพรวมการทำงานของกรณีศึกษาเมื่อเข้ากระบวนการ Load Shed

จากรูป เข้ากระบวนการ Load Shed เพราะเนื่องจากค่า Load มีค่ามากกว่าค่า Generator จะเกิดการ Trip Load ตาม Priority ที่กำหนด ระบบจะเลิก Trip ก็ต่อเมื่อเรายก Breaker

Case No.	Condition	No. of GTGs	No. of STG	Total Load	Total Gen	Scenario
2	Normal Operation	2	1	33.00	42.00	Open 3YB-01



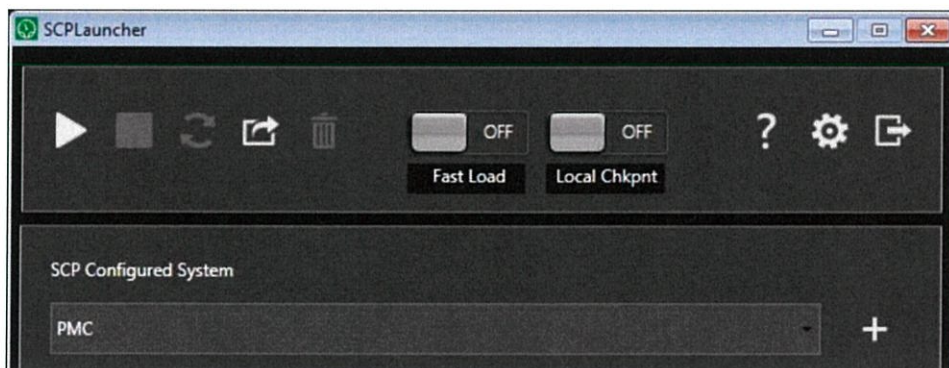
รูปที่ 3.8 ภาพรวมการทำงานของกรณีศึกษาเมื่อเข้ากระบวนการ Generator Shed

จากรูป หลังจากทำการ Load Shed เสร็จ ระบบจะเข้ากระบวนการ Generator Shed เมื่อค่า Generator มีค่ามากกว่าค่า Load จะเกิดการ Trip Load ตาม Priority ที่กำหนด ระบบจะเลิก Trip ก็ต่อเมื่อ เรายก Breaker จากนั้นระบบจะเข้ากระบวนการของ Frequency control ซึ่งในกระบวนการนี้ หาก Frequency มีค่าอยู่ระหว่าง 49.9 และ 50.1 ระบบจะเข้ากระบวนการ Isochronous Mode

### 3.3 ขั้นตอนการดำเนินงานด้านซอฟต์แวร์

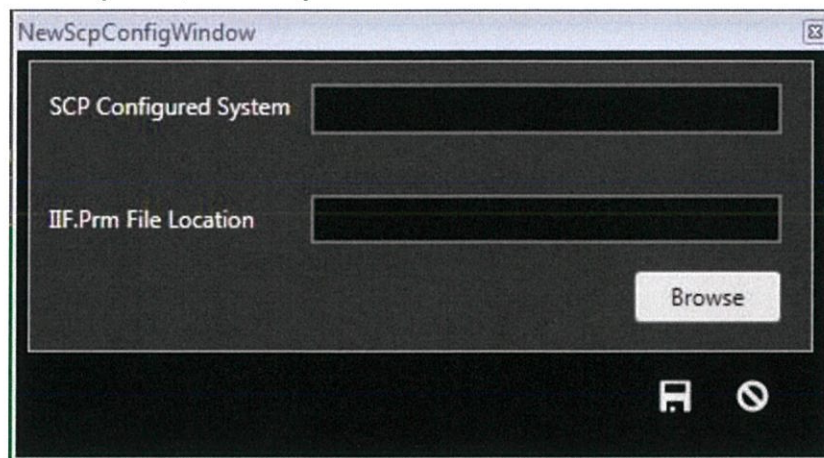
#### 3.3.1 โปรแกรม SCP Launcher

1. กด “+” (เพิ่มชื่อไฟล์ที่เราต้องการสร้าง)



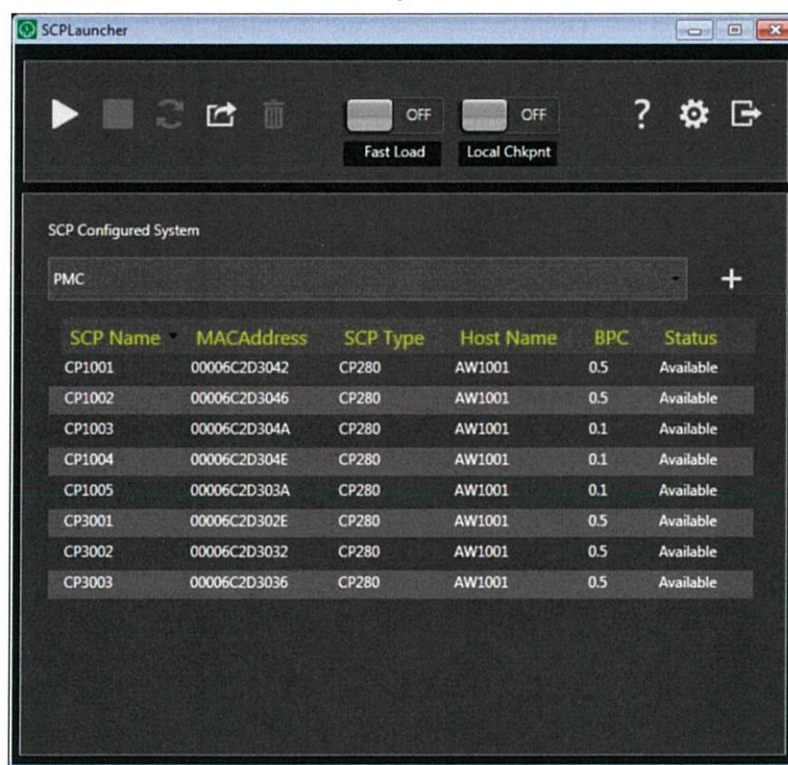
รูปที่ 3.9 การสร้าง SCP

2. กรอกข้อมูลต่างๆ > Save (รูปแผ่นดิส)



รูปที่ 3.10 NewScpConfigWindow

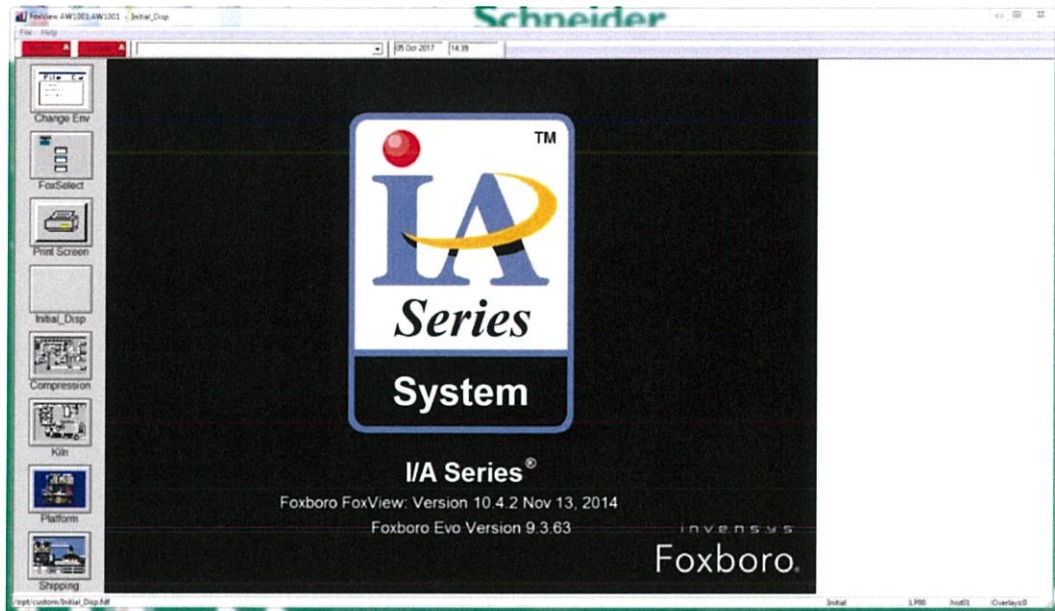
3. กดเลือก SCP ที่เราต้องการ > กด Play



รูปที่ 3.11 การ Run SCP

### 3.3.2 การสร้าง GUI โดย FoxDraw

1. เข้าโปรแกรม FoxView > Chang Env



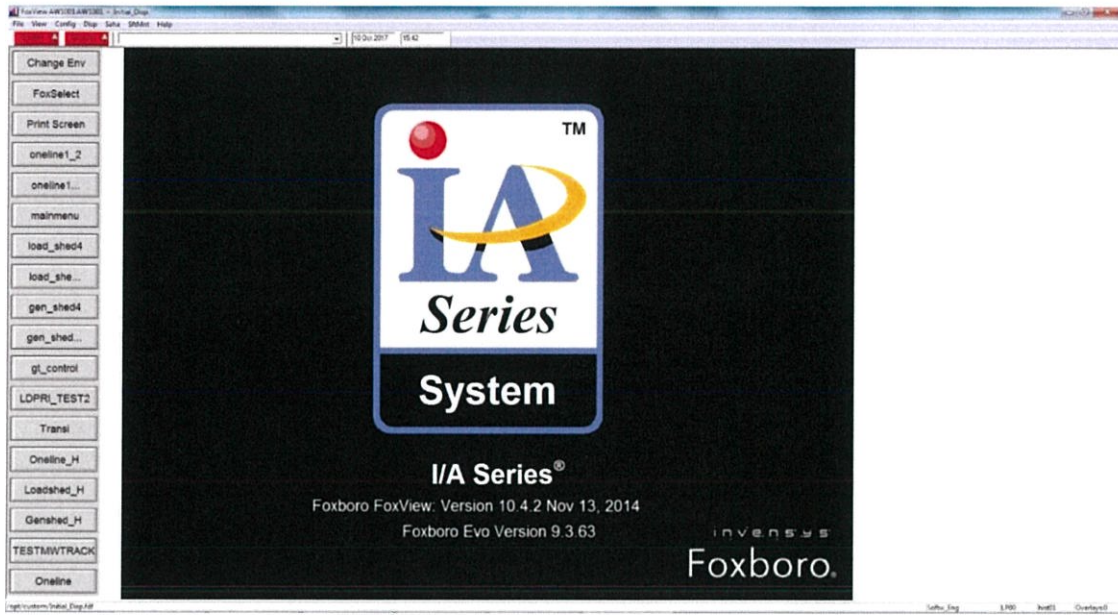
รูปที่ 3.12 โปรแกรม FoxView

2. ตรงส่วน Environment : เลือก Softw\_Eng > OK



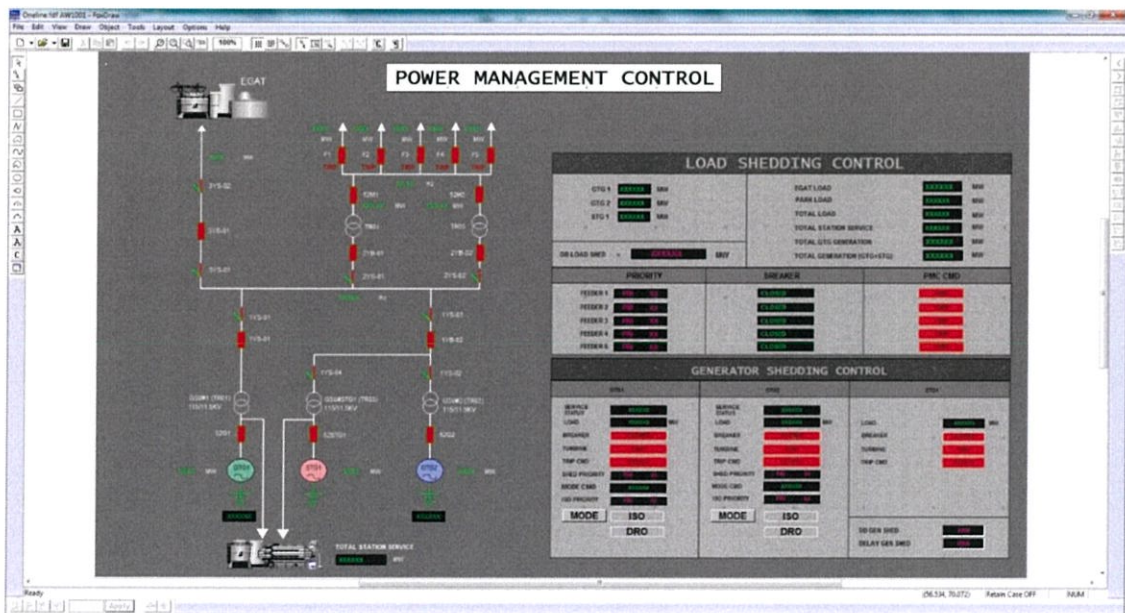
รูปที่ 3.13 Foxboro: Change Environment

3. คลิกขวาตรงหน้าPageที่ว่าง > Open with FoxDraw > OK



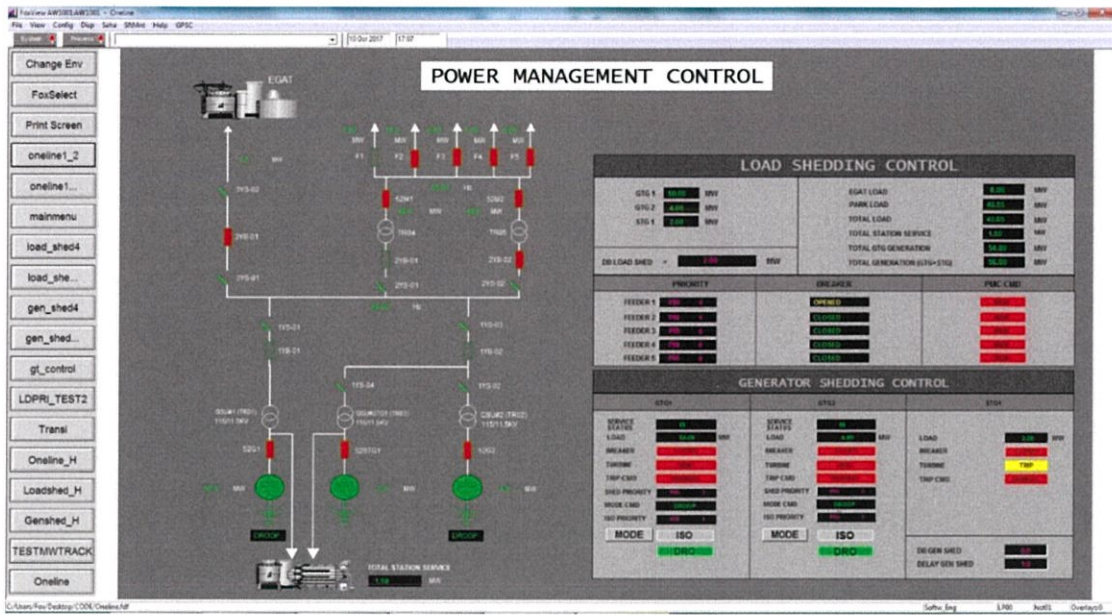
รูปที่ 3.14 หน้าต่างโปรแกรม FoxView

4. สร้างหน้ากราฟิกต่างๆ ดังนี้



รูปที่ 3.15 โปรแกรม FoxDraw

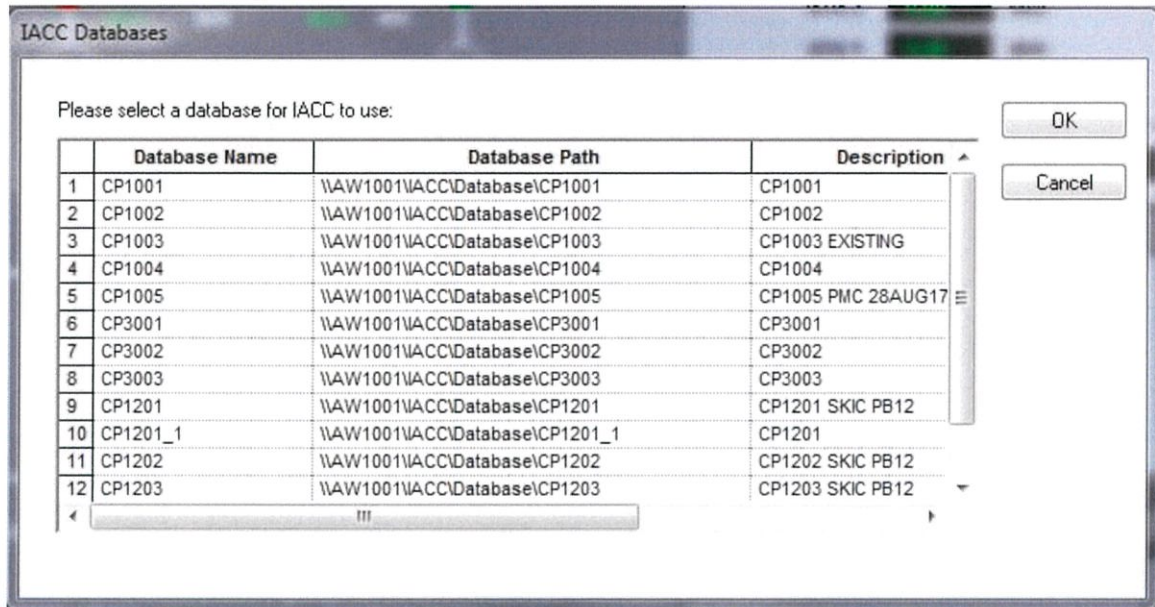
5. Save > FoxView (หน้าแสดงผล)



รูปที่ 3.16 หน้าต่างสำหรับแสดงผลโดยโปรแกรม FoxView

### 3.3.3 การเขียน Database

1. เปิดโปรแกรม IACC
2. เลือก CP ที่ต้องการ (CP1005) > OK



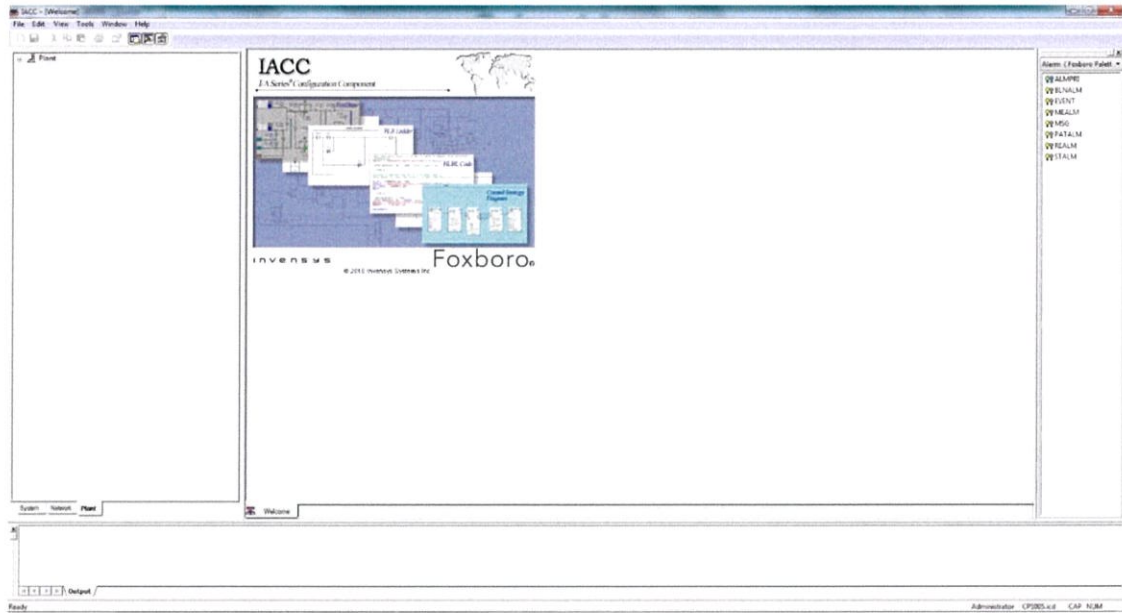
รูปที่ 3.17 หน้าต่างสำหรับเลือก Database ของ IACC

3. หน้า Logon : ไม่ต้องใส่อะไร > OK



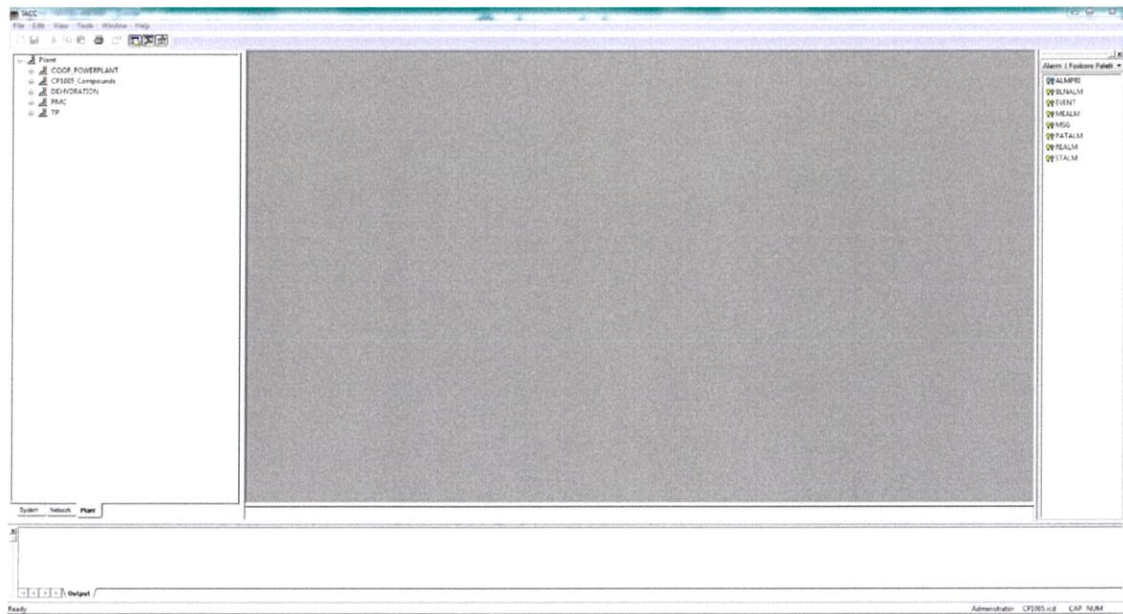
รูปที่ 3.18 หน้าต่างสำหรับ Logon

4. สร้าง Plant Area : คลิกขวาที่ตรงคำว่า Plant > New Plant Area



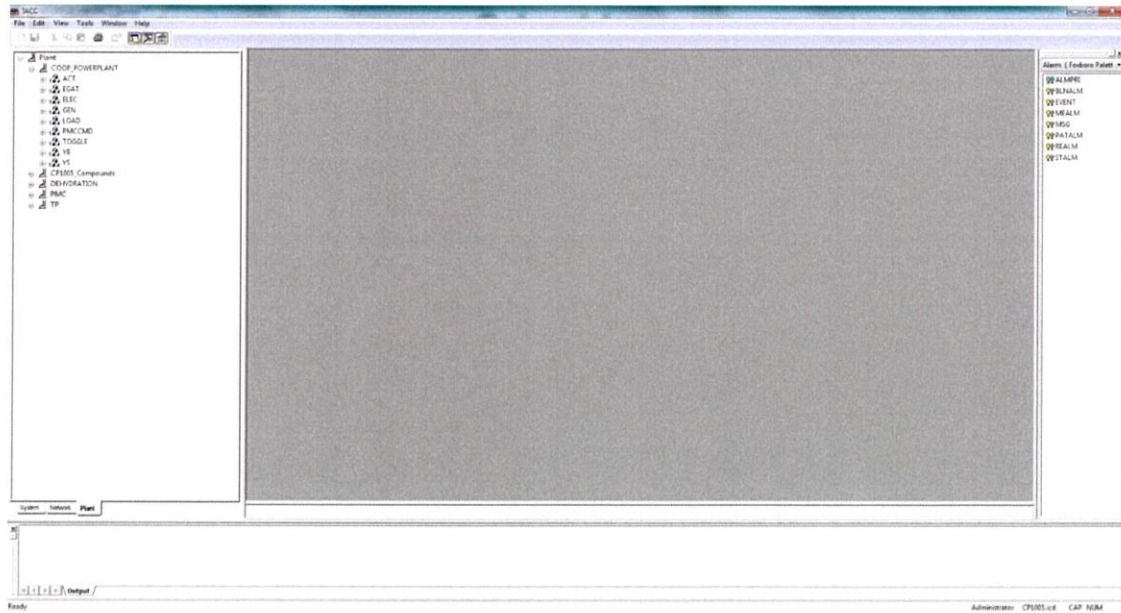
รูปที่ 3.19 หน้าต่าง Plant ของโปรแกรม IACC

5. ตั้งชื่อ Plant Area (COOP\_POWERPLANT)



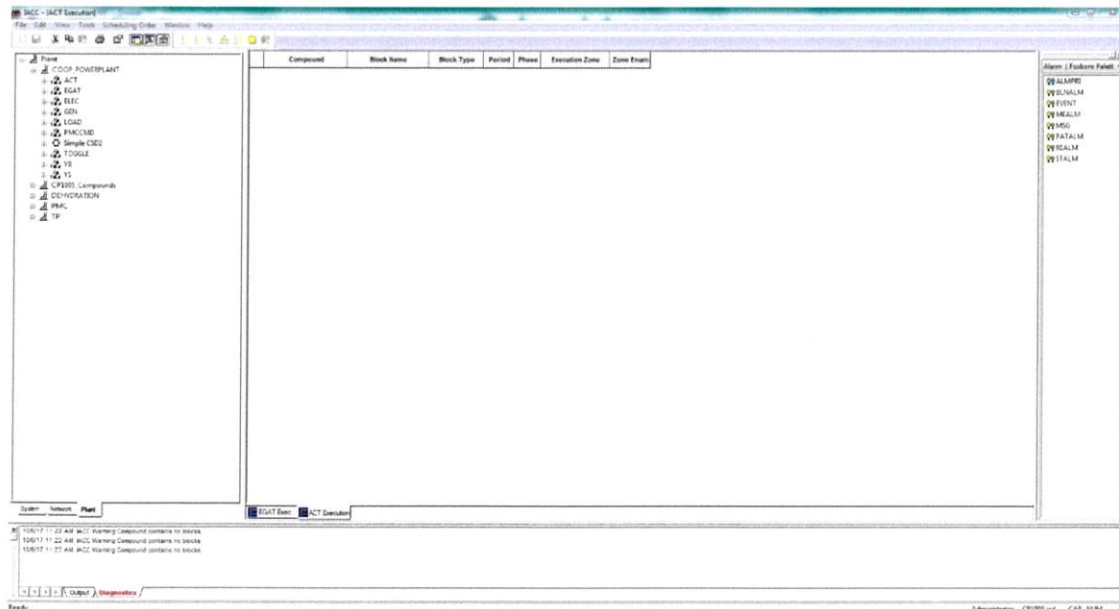
รูปที่ 3.20 Plant Area

6. สร้าง Compound : คลิกขวาตรง Plant Area ที่สร้างขึ้นมา > New > Compound
7. ตั้งชื่อให้กับ Compound : คลิกขวาตรง Compound > Rename



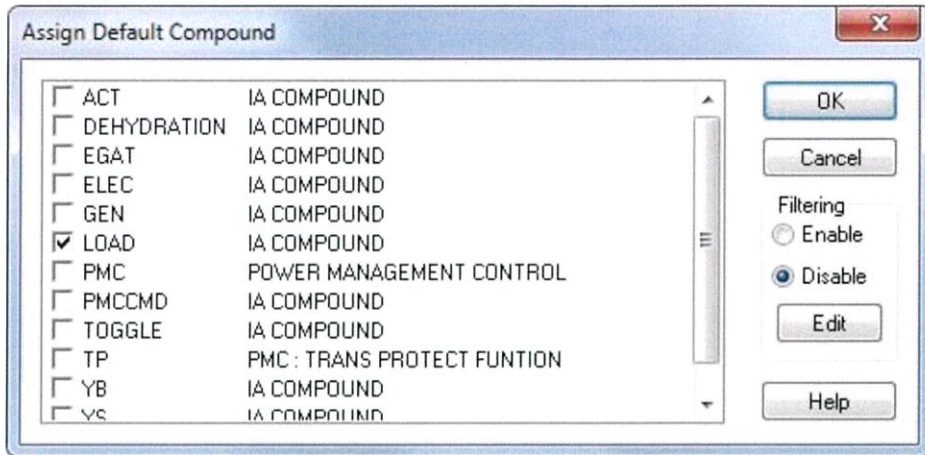
รูปที่ 3.21 Compound

8. สร้าง CSD : คลิกขวาตรง Plant Area ที่สร้างขึ้นมา > New > CSD > Simple CSD
9. ตั้งชื่อให้กับ CSD : คลิกขวาตรง CSD > Rename



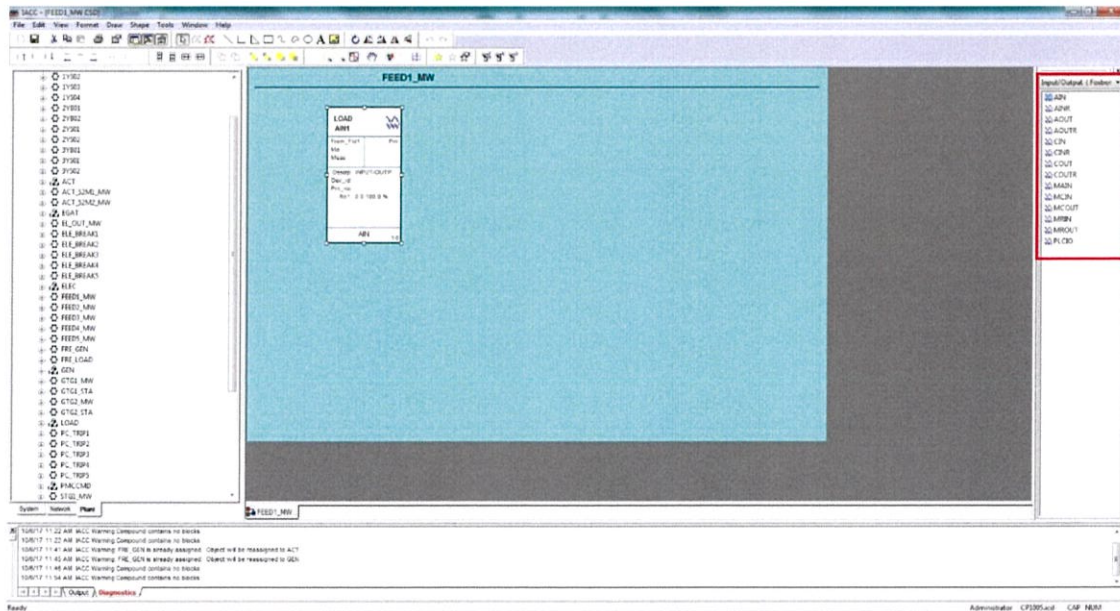
รูปที่ 3.22 CSD

- ย้าย CSD ที่สร้างขึ้นไปอยู่ใน Compound ที่สร้างขึ้น : คลิกขวาที่ CSD ที่ต้องการ > Assign Default Compound > เลือก Compound ที่ต้องการ > OK



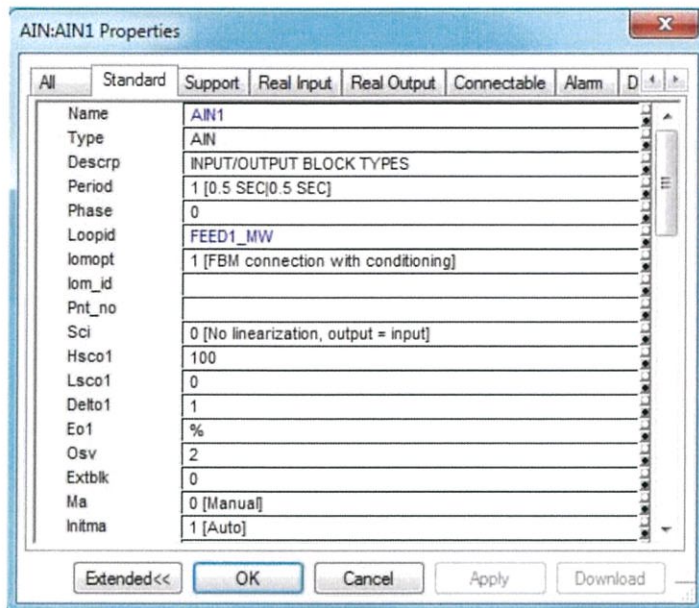
รูปที่ 3.23 Assign Default Compound

- สร้าง Function Block Diagram : Double Click ที่ CSD ที่สร้างขึ้น
- สังเกตด้านขวา เลือกฟังก์ชันที่ต้องการ จากนั้นลากมาวางที่ตรงส่วนหน้า Page



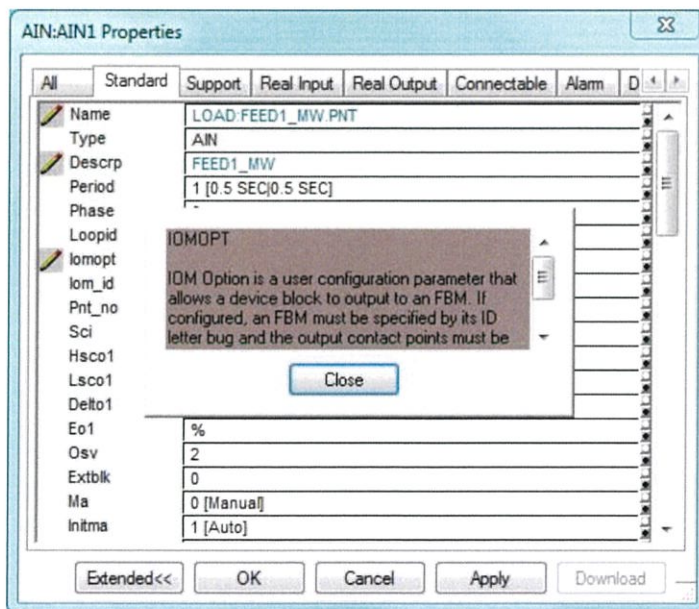
รูปที่ 3.24 Function Block Diagram

13. ตั้งค่า Block : Double Click ที่ Block



รูปที่ 3.25 การตั้งค่า Block

สามารถอ่านคำอธิบายของแต่ละคำสั่งได้โดยการคลิกขวาที่คำสั่งนั้น



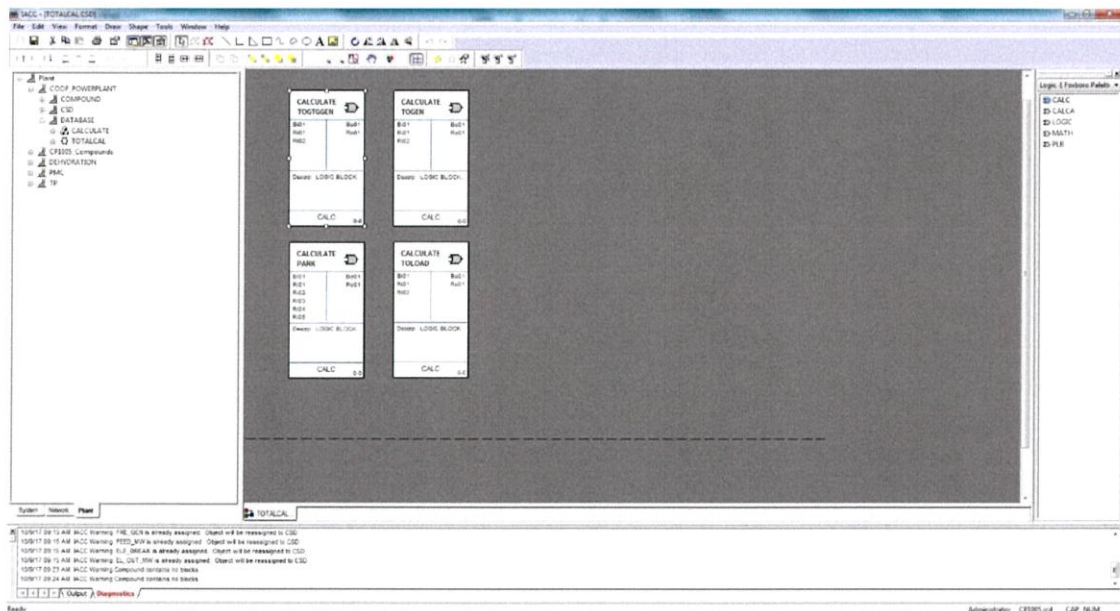
รูปที่ 3.26 คำอธิบายของการตั้งค่า Block

### 3.3.4 การสร้าง Block CALC

1. สร้าง Plant Area ย่อย (DATABASE) ลงใน Plant Area หลัก (COOP\_POWERPLANT) โดยการคลิกขวาตรง Plant Area หลัก (COOP\_POWERPLANT) ที่ได้สร้างขึ้นมาก่อนหน้านั้น > New > Plant Area

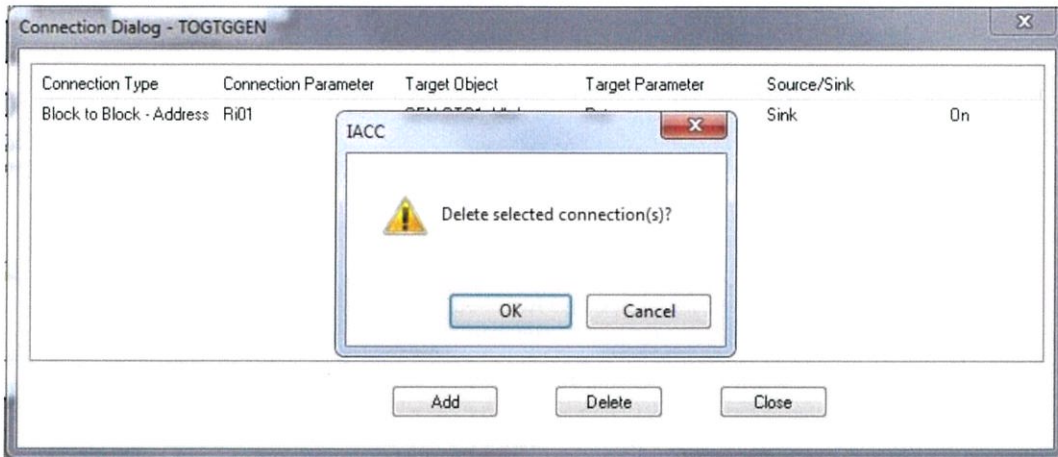
2. สร้าง Compound ลงใน Plant Area ย่อย
3. สร้าง CSD ลงใน Compound
4. ตั้งค่า CSD
5. เลือก Block CALC ทางด้านขวา ลากมาวางยังหน้า Page
6. ตั้งค่า Block CALC : Double Click ที่ Block CALC เพื่อทำการตั้งค่า
7. ทำการเชื่อมTag ของ Block กับ Compound อื่น

#### 7.1 คลิกขวาที่ Block > Connect



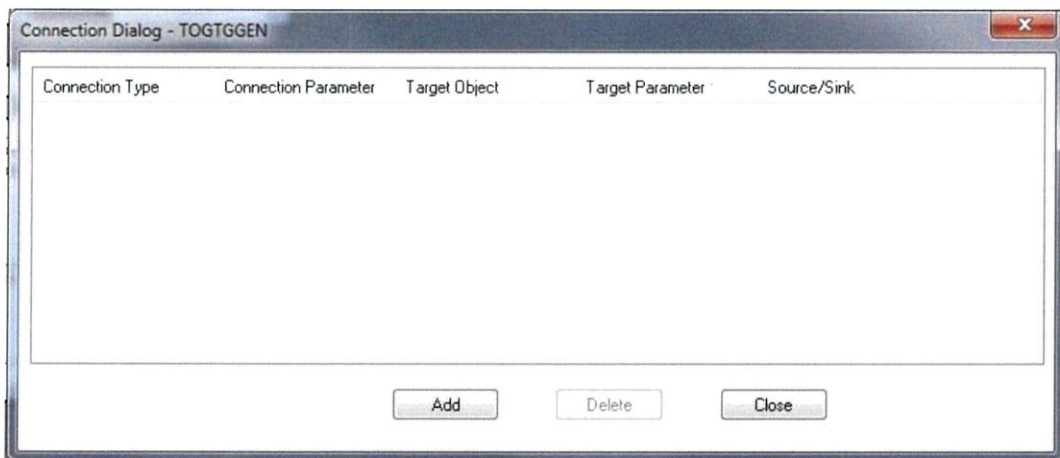
รูปที่ 3.27 การเชื่อมTag ของ Block กับ Compound ในโปรแกรม IACC

7.2 หน้า Connection Dialog : Delete ตัวที่ไม่ต้องการออก โดยการเลือกที่ Connection Type ที่ไม่ต้องการ > Delete > OK



รูปที่ 3.28 Connection Dialog: การDelete Typeที่ไม่ต้องการออกConnection Dialog

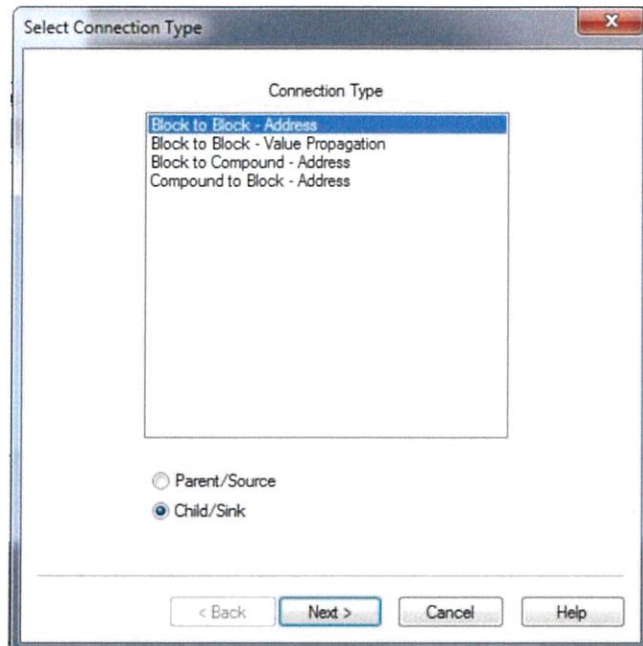
### 7.3 Add



รูปที่ 3.29 Connection Dialog: การ Add Type ที่ต้องการ

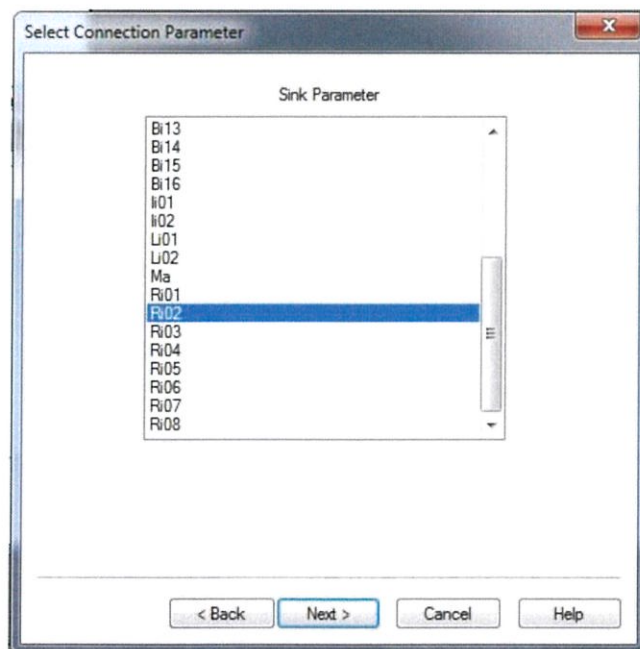
7.4 หน้า Select Connection Type : Block to Block – Address > Child/Sink >

Next



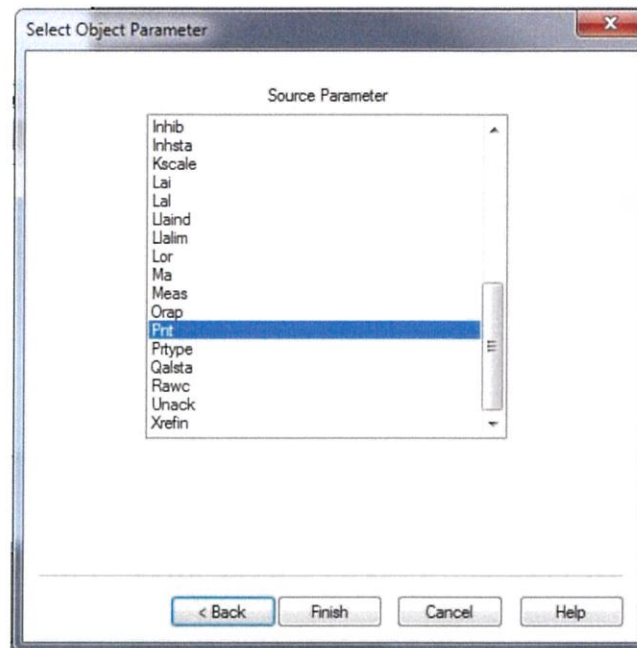
รูปที่ 3.30 Select Connection Type

7.5 หน้า Select Connection Parameter : เลือก Parameter > Next



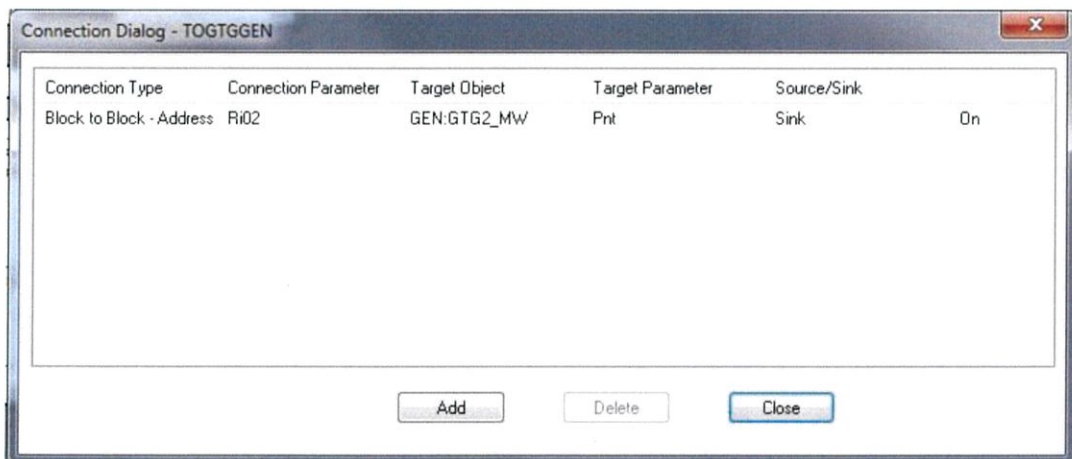
รูปที่ 3.31 Select Connection Parameter

7.6 หน้า Select Object Parameter : เลือกประเภทของไฟล์ที่เป็นตัว Sink > Finish

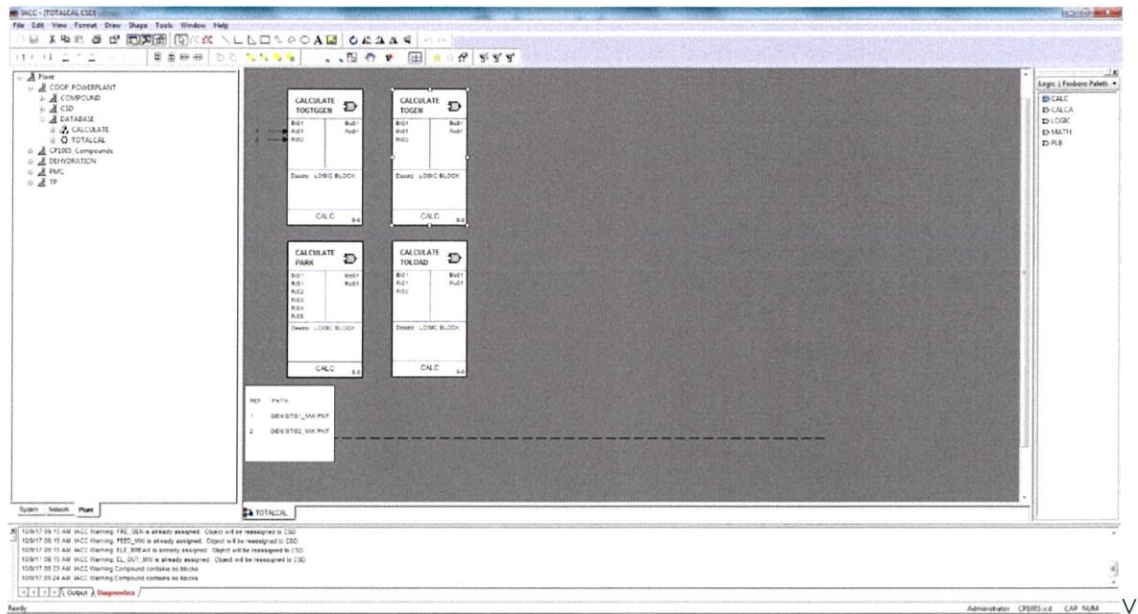


รูปที่ 3.32 Select Object Parameter

7.7 หน้า Connection Dialog : Close



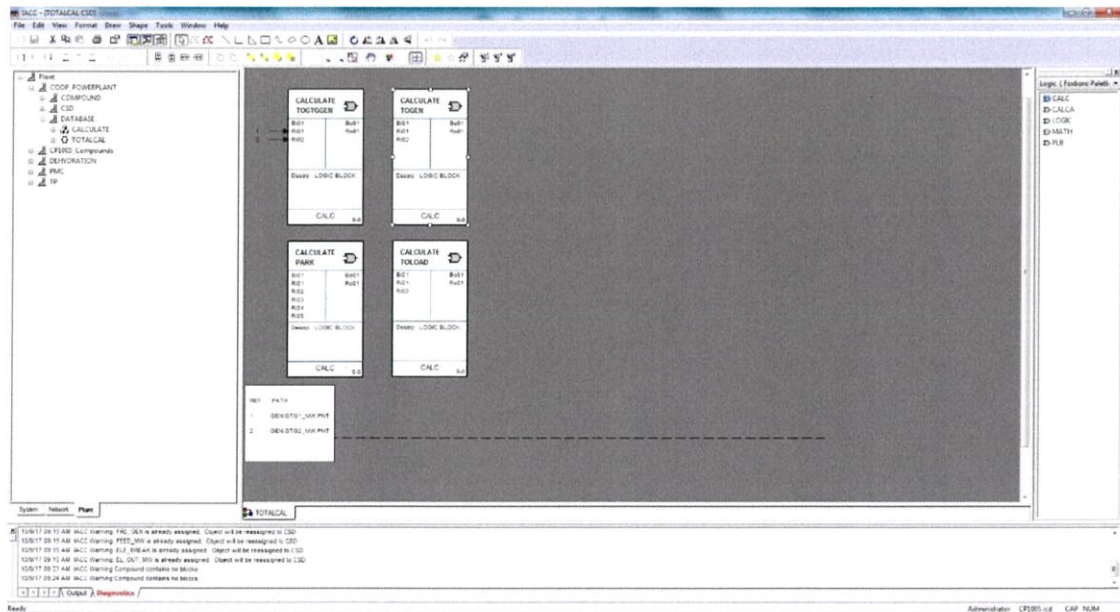
รูปที่ 3.33 เมื่อ Connection Dialog เสร็จ



รูปที่ 3.34 เมื่อเชื่อม Tag ของ Block กับ Compound ในโปรแกรม IACC เสร็จ

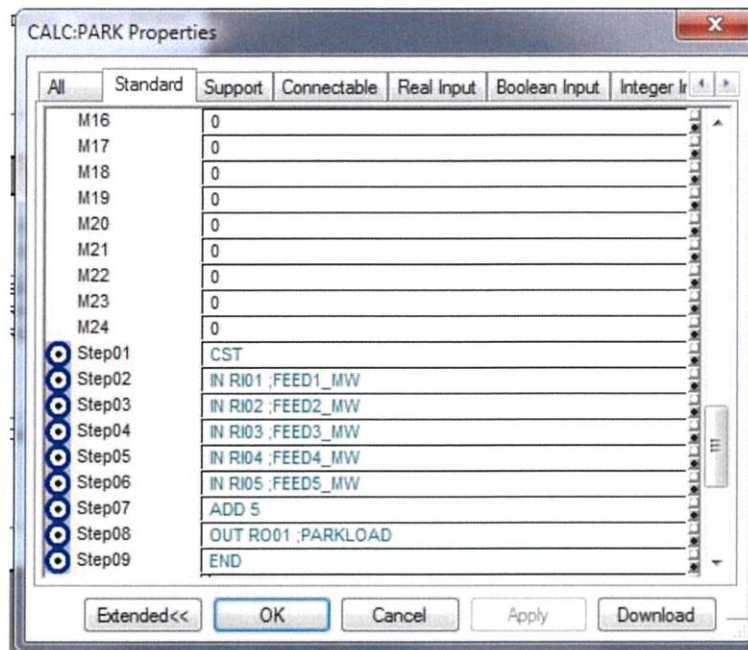
### 3.3.5 การเขียนโค้ดลงใน Block CALC

#### 1. Double Click ที่ Block



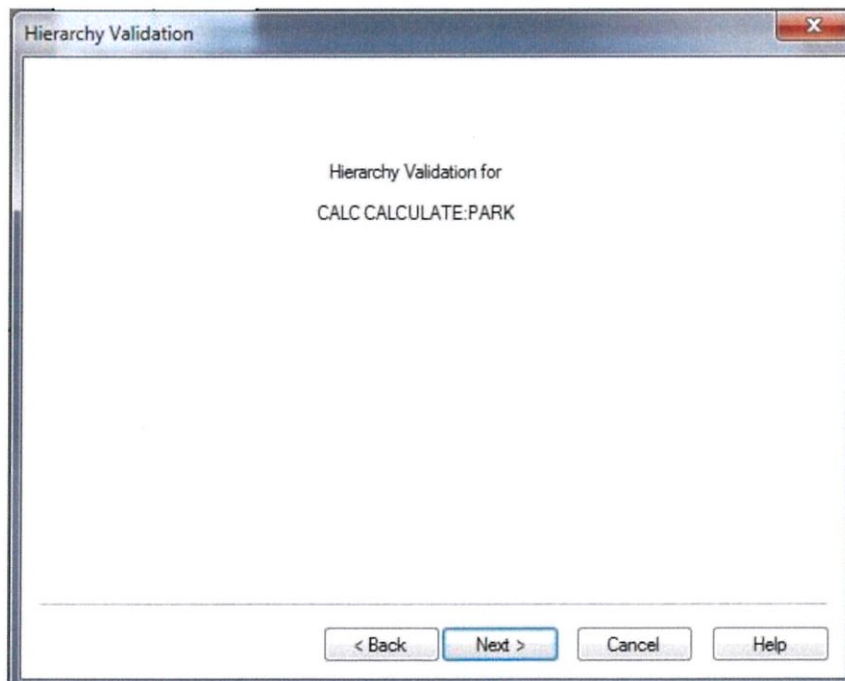
รูปที่ 3.35 การเขียนโค้ดลงในBlock

2. หน้า CALC : Properties (Standard) พิมพ์ได้ลงตรงส่วนที่เขียนว่า Step > Download



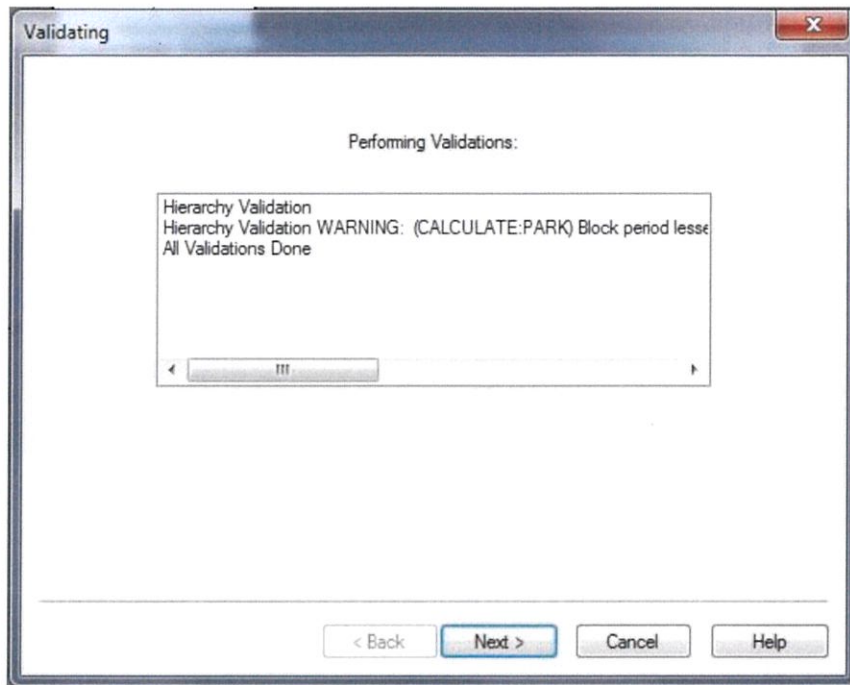
รูปที่ 3.36 CALC:PARK Properties

3. หน้า Hierarchy Validation : Next



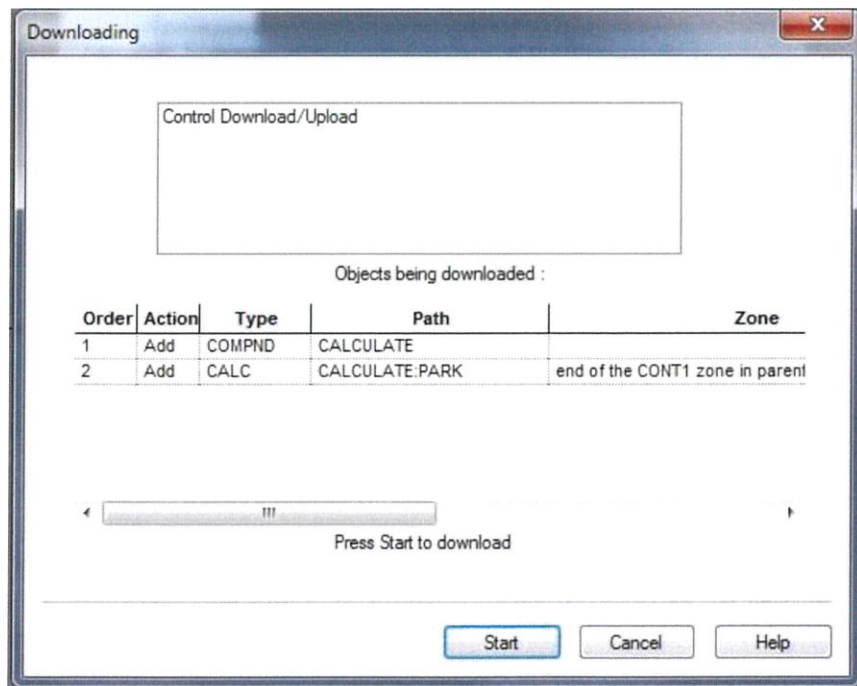
รูปที่ 3.37 Hierarchy Validation

4. หน้า Validation : Next

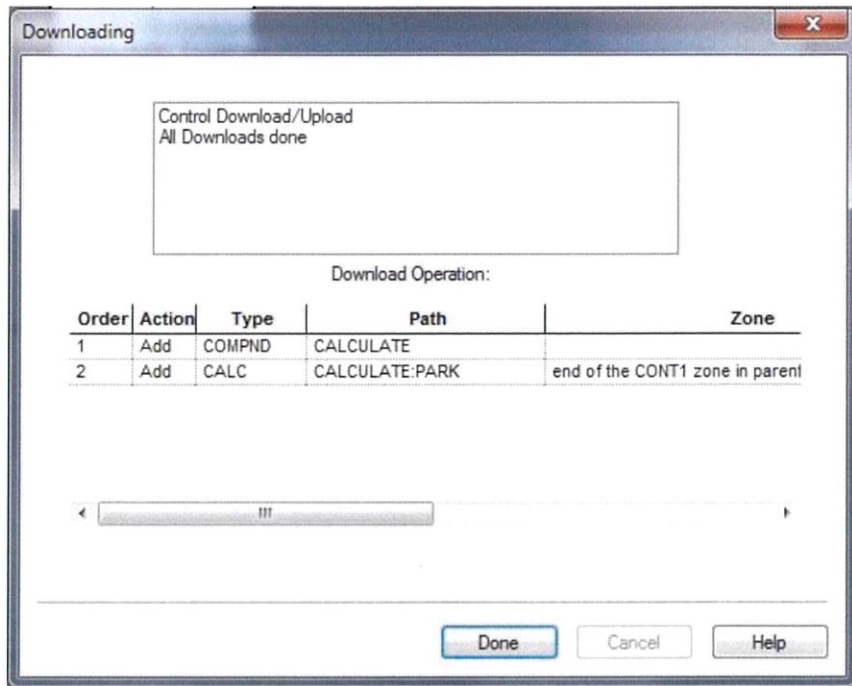


รูปที่ 3.38 Validation

5. หน้า Download : Start > Done



รูปที่ 3.39 Downloading



รูปที่ 3.40 ยืนยันการติดตั้ง

6. นำ Tag ไปติดที่หน้า FoxDraw

### 3.4 ส่วนประกอบการทำงานของ Plant Management Control ในโปรแกรม IACC

#### 3.4.1 CODE : PMC (PLANT MANAGEMENT CONTROL)

```
DEPENDENT_SEQUENCE{*****
**
**      DEPENDENT SEQUENCE      **
**      CONTROL BLOCK          **
**
**      *****
/*GEN_STA*/
#define GTG1STA :GEN:GTG1_STA.CIN  {GTG1_STA}
#define GTG2STA :GEN:GTG2_STA.CIN  {GTG2_STA}
#define STG1STA :GEN:STG1_STA.CIN  {STG1_STA}

/*GEN_MW*/
#define GTG1MW  :GEN:GTG1_MW.PNT   {GTG1_MW}
#define GTG2MW  :GEN:GTG2_MW.PNT   {GTG2_MW}
#define STG1MW  :GEN:STG1_MW.PNT   {STG2_MW}

/*FREQUENCY*/
#define FREGEN  :GEN:FRE_GEN.PNT   {FRE_GEN}
#define FRELOAD :LOADD:FRE_LOAD.PNT {FRE_LOAD}

/*LOAD FEEDER_MW*/
#define MWFEED1 :LOADD:FEED1_MW.PNT {FEED1_MW}
#define MWFEED2 :LOADD:FEED2_MW.PNT {FEED2_MW}
#define MWFEED3 :LOADD:FEED3_MW.PNT {FEED3_MW}
#define MWFEED4 :LOADD:FEED4_MW.PNT {FEED4_MW}
#define MWFEED5 :LOADD:FEED5_MW.PNT {FEED5_MW}

/*PRIORITY FEEDER*/
#define PRIFEED1 :SEQUENCE:FEED1_PRI.VALUE {FEED1_PRI}
#define PRIFEED2 :SEQUENCE:FEED2_PRI.VALUE {FEED2_PRI}
#define PRIFEED3 :SEQUENCE:FEED3_PRI.VALUE {FEED3_PRI}
#define PRIFEED4 :SEQUENCE:FEED4_PRI.VALUE {FEED4_PRI}
#define PRIFEED5 :SEQUENCE:FEED5_PRI.VALUE {FEED5_PRI}

/*BREAKER_ELEC*/
#define EBREAKER1 :EBS:ELE_BREAK1.CIN {ELE_BREAK1}
#define EBREAKER2 :EBS:ELE_BREAK2.CIN {ELE_BREAK2}
#define EBREAKER3 :EBS:ELE_BREAK3.CIN {ELE_BREAK3}
#define EBREAKER4 :EBS:ELE_BREAK4.CIN {ELE_BREAK4}
#define EBREAKER5 :EBS:ELE_BREAK5.CIN {ELE_BREAK5}

/*PMC_CMD*/
#define PMCTRIP1 :PMCCMD:PC_TRIP1.COUT {PC_TRIP1}
#define PMCTRIP2 :PMCCMD:PC_TRIP2.COUT {PC_TRIP2}
#define PMCTRIP3 :PMCCMD:PC_TRIP3.COUT {PC_TRIP3}
#define PMCTRIP4 :PMCCMD:PC_TRIP4.COUT {PC_TRIP4}
#define PMCTRIP5 :PMCCMD:PC_TRIP5.COUT {PC_TRIP5}

/*CALCULATE*/
#define TO_GTGGEN :CALCULATE:TOGTGGEN.R001 {TOGTGGEN:GTG1+GTG2}
#define TO_GEN :CALCULATE:TOGEN.R001 {TOGEN:GTG1+GTG2+STG1}
#define PLOAD :CALCULATE:PARK.R001 {PARKLOAD:F1+F2+F3+F4+F5}
#define TO_LOADD :CALCULATE:TOLOAD.R001 {TO_LOAD:PAKE LOAD+TOTAL STATION SERVICE}
#define TOSTA_MW :CALCULATE:STA_MW.AIN {TOTAL STATION SERVICE:STA_MW}

/*TOGGLE*/
#define TG52M1 :TOGGLE:TG_52M1.CIN {TG_52M1}
#define TG52M2 :TOGGLE:TG_52M2.CIN {TG_52M2}
#define TG52G1 :TOGGLE:TG_52G1.CIN {TG_52G1}
#define TG52G2 :TOGGLE:TG_52G2.CIN {TG_52G2}
#define TG52STG1 :TOGGLE:TG_52STG1.CIN {TG_52STG1}

/*YS*/
#define P1YS_01 :YS:1YS01.CIN {1YS01}
#define P1YS_02 :YS:1YS02.CIN {1YS02}
#define P1YS_03 :YS:1YS03.CIN {1YS03}
#define P1YS_04 :YS:1YS04.CIN {1YS04}
#define P2YS_01 :YS:2YS01.CIN {2YS01}
#define P2YS_02 :YS:2YS02.CIN {2YS02}
#define P3YS_01 :YS:3YS01.CIN {3YS01}
#define P3YS_02 :YS:3YS02.CIN {3YS02}

/*YB*/
#define P1YB_01 :YB:1YB01.CIN {1YB01}
#define P1YB_02 :YB:1YB02.CIN {1YB02}
#define P2YB_01 :YB:2YB01.CIN {2YB01}
#define P2YB_02 :YB:2YB02.CIN {2YB02}
#define P3YB_01 :YB:3YB01.CIN {3YB01}
```

```

/*MW_ACTURE*/
#define MWACT_52M1 :ACT:ACT_52M1_MW.PNT {ACT_52M1_MW}
#define MWACT_52M2 :ACT:ACT_52M2_MW.PNT {ACT_52M2_MW}

/*EGAT_LOAD*/
#define EGLOAD_MW :EGAT:EL_OUT_MW.PNT {EGAT_LOAD_MW}

/*SEQUENCE*/
#define PRIISO1 :SEQUENCE:ISO1_PRI.VALUE {ISO1_PRI_GTG1}
#define PRIISO2 :SEQUENCE:ISO2_PRI.VALUE {ISO2_PRI_GTG2}
#define G1_PRISHED :SEQUENCE:PRISHED_G1.VALUE {PRISHED_G1}
#define G2_PRISHED :SEQUENCE:PRISHED_G2.VALUE {PRISHED_G2}

CONSTANTS {*****
* specify any Constants *
* in the following format: *
*****
* 12_char_name = value ; *
*****}
DROOP = FALSE;
ISO = TRUE;

VARIABLES {*****
* specify any Block Local Variables *
* in the following format: *
*****
* 12_char_name [ , 12_char_name ] : type;*
* *
*where type is one of B, I, R, S, S12, S6*
*****}
LD_END : B;
GN_END : B;
FC_END : B;

USER_LABELS{*****
* specify any user labeled parameters *
* in one of the following formats: *
*****
* 10_char_name : BI00nn; n = 01 -- 24 *
* 10_char_name : BO00nn; n = 01 -- 16 *
* 10_char_name : BA00nn; n = 1 -- 4 *
* 10_char_name : RI00nn; n = 01 -- 15 *
* 10_char_name : RO00nn; n = 01 -- 15 *
* 10_char_name : RA00nn; n = 1 -- 2 *
* 10_char_name : II00nn; n = 1 -- 8 *
* 10_char_name : IO00nn; n = 1 -- 5 *
* 10_char_name : IA0001; *
* 10_char_name : SN00nn; n = 01 -- 10 *
*****}
/*MODE GTG*/
GT1_MODE : BO0001; {GTG1_MODE}
GT2_MODE : BO0002; {GTG2_MODE}

/*FSNL*/
FSNL_STG : BO0003; {FSNL_STG}

/*GTG TRIP*/
GTG1_TRIP : BO0004; {GTG1_TRIP}
GTG2_TRIP : BO0005; {GTG2_TRIP}

/*GTG ISO*/
ISOGT1 : BO0006; {ISO GTG1}
ISOGT2 : BO0007; {ISO GTG2}

/*RESET_TRIP*/
RESET_TRIP : BO0015;

/*DB*/
DB_LOAD : RI0001; {DB_LOAD}
DB_GEN : RI0002; {DB_GEN}

/*DELAY TIME*/
TIME1 : RI0003; {DELAY TIME1}
TIME2 : RI0004; {DELAY TIME2}

```

```

/*LPRIGT MWLOW*/
LPRIGT_LOW : RI0005;

TOTAL_MWLOW : RI0006;
TOTAL_GENMAX : R00001;
LOAD_DB: R00002;
MW_WAT : R00003;

/*TRIP CMD*/
GT1_SHED : B00008;
GT2_SHED : B00009;
STG1 : B00010;

/*SP_GT*/
GT1_SP : RI0007;
GT2_SP : RI0008;

/*GT_MAXMIN*/
GT1_MIN : RI0009;
GT2_MIN : RI0010;
GT1_MAX : RI0011;
GT2_MAX : RI0012;

/*PMC_ISO*/
GT1_PMC_ISO : B00012;
GT2_PMC_ISO : B00013;

/*LOADSHED*/
LOADSHED : B00014;

MWREF_GT1: R00004;
MWREF_GT2: R00005;

{*****
 * Specify any Subroutines
 * *****
 *
 * SUBROUTINE name ( formal arguments ) ;
 * VARIABLES subr. local variables ;
 * STATEMENTS
 * ;
 * ENDSUBROUTINE
 * *****}

{*****
 * Specify any
 * Standard Block Exception Handlers
 * *****
 *
 * BLOCK_EXCEPTION exc_name [ DISABLE ]
 * STATEMENTS
 * ;
 * ENDEXCEPTION
 * *****}

```

```

STATEMENTS{*****
 * Specify the statements here
 * *****}

```

```

WHILE TRUE DO

```

```

    IF (P3YB_01 = FALSE) THEN
        {GT1_ISO := DROOP;}
        IF (ISOGT1 OR ISOGT2) = FALSE THEN
            GT1_MODE := TRUE;
            GT2_MODE := TRUE;
            FSNL_STG := DROOP;

            GT1_SP := GTG1MW;
            GT2_SP := GTG2MW;
            ENDF;

        WAIT TIME1;

```

```

/*LOADSHED*/
IF LD_END = FALSE THEN
<<LSHED>>
    LOAD_DB := TO_LOADD + DB_LOAD;
    IF (LOAD_DB > TO_GTGEN) THEN
        LOADSHED := TRUE;
        RESET_TRIP := FALSE;
        ACTIVATE :SEQUENCE:LD_LOADSHED;
        GOTO LSHED;
    ELSE
        LOADSHED := FALSE;
        RESET_TRIP := TRUE;
        ACTIVATE :SEQUENCE:LD_LOADSHED;
        ENDIF;

    LD_END := TRUE;
ENDIF;

/*GENSHED*/
IF LD_END = TRUE AND GN_END = FALSE THEN
<<GEN_SHED>>
    LOAD_DB := TO_LOADD + DB_GEN;

    STG1 := TRUE;

    TOTAL_GENMAX := TOTAL_MWLOW;
    IF G1_PRISHED = LPRIGT_LOW THEN
        MW_WAT := TOTAL_GENMAX - 60;
        LOAD_DB := TO_LOADD + DB_GEN;

        IF (MW_WAT > LOAD_DB) THEN
            GT1_SHED := TRUE; GOTO CHECKPRI;
        ELSEIF (MW_WAT < LOAD_DB) THEN
            GT1_SHED := FALSE; GOTO CHECKPRI;
        ENDIF;
    ENDIF;

    TOTAL_GENMAX := TOTAL_MWLOW;
    IF G2_PRISHED = LPRIGT_LOW THEN
        MW_WAT := TOTAL_GENMAX - 45;
        LOAD_DB := TO_LOADD + DB_GEN;

        IF (MW_WAT > LOAD_DB) THEN
            GT2_SHED := TRUE; GOTO CHECKPRI;
        ELSEIF (MW_WAT < LOAD_DB) THEN
            GT2_SHED := FALSE; GOTO CHECKPRI;
        ENDIF;
    ENDIF;

    TOTAL_GENMAX := TOTAL_MWLOW;

    IF LPRIGT_LOW = 0 THEN
        IF (G1_PRISHED = LPRIGT_LOW) THEN
            MW_WAT := TOTAL_GENMAX-60;
        ELSEIF (G2_PRISHED = LPRIGT_LOW) THEN
            MW_WAT := TOTAL_GENMAX-45;
        ENDIF;
    ENDIF;

<<CHECKPRI>>
    IF (LPRIGT_LOW = 0) AND (MW_WAT <= LOAD_DB) THEN
        GOTO LAST_GN;
    ELSEIF (LPRIGT_LOW = 0) AND NOT ((MW_WAT<=LOAD_DB) AND (LOAD_DB<=TOTAL_GENMAX)) THEN
        GOTO GEN_SHED;
    ELSE
        WAIT UNTIL (MW_WAT<=LOAD_DB) AND (LOAD_DB<=TOTAL_GENMAX) AFTER TIME2 GOTO GEN_SHED;
    ENDIF;

```

<<LAST\_GN>>

```
GN_END := TRUE;
ENDIF;
      /*WAIT TIME2;
      GN_END := TRUE;*/

      (:ABC:SW1.TOGGLE := FALSE;)}

      /*FREQUENCYCTRL*/
IF FC_END=FALSE THEN
  IF (FREGEN < 49.9) THEN
    IF (GTG1STA = TRUE) AND (GT1_PMC_ISO = FALSE) THEN
      MWREF_GT1 := GT1_MAX;
      GT1_MODE := TRUE;
      ISOGT1 := FALSE;
    ENDIF;
    IF (GTG2STA = TRUE) AND (GT2_PMC_ISO = FALSE) THEN
      MWREF_GT2 := GT2_MAX;
      GT2_MODE := TRUE;
      ISOGT2 := FALSE;
    ENDIF;
    GOTO LAST_GN;
  ENDIF;
  IF (FREGEN > 50.1) THEN
    IF (GTG1STA = TRUE) AND (GT1_PMC_ISO = FALSE) THEN
      MWREF_GT1 := GT1_MIN;
      GT1_MODE := TRUE;
      ISOGT1 := FALSE;
    ENDIF;
    IF (GTG2STA = TRUE) AND (GT2_PMC_ISO = FALSE) THEN
      MWREF_GT2 := GT2_MIN;
      GT2_MODE := TRUE;
      ISOGT2 := FALSE;
    ENDIF;
    GOTO LAST_GN;
  ENDIF;
  IF (FREGEN >= 49.9) AND (FREGEN <= 50.1) THEN
    IF (GTG1STA = TRUE) AND (GT1_PMC_ISO = FALSE) AND PRIISO1 = 1 THEN
      GT1_MODE := FALSE;
      ISOGT1 := TRUE;
    ENDIF;
    IF (GTG2STA = TRUE) AND (GT2_PMC_ISO = FALSE) AND PRIISO2 = 1 THEN
      GT2_MODE := FALSE;
      ISOGT2 := TRUE;
    ENDIF;
    MWREF_GT1 := GTG1MW;
    MWREF_GT2 := GTG2MW;
    GOTO SEL_ISO;
  ENDIF;
```

```
<<SEL_ISO>>
```

```
FC_END := TRUE;  
ENDIF;
```

```
ENDIF;
```

```
IF (P3YE_01 = TRUE) THEN  
STG1      := FALSE;  
GT1_SHED  := FALSE;  
GT2_SHED  := FALSE;  
ISOGT1    := FALSE;  
ISOGT2    := FALSE;  
LOADSHED  := FALSE;  
GT1_MODE  := TRUE;  
GT2_MODE  := TRUE;  
FC_END    := FALSE;  
GN_END    := FALSE;  
LD_END    := FALSE;  
ACTIVATE  :SEQUENCE:LD_LOADSHED;  
ENDIF;
```

```
ENDWHILE
```

```
ENDSEQUENCE
```

### 3.4.2 CODE : LOADSHED

```

DEPENDENT_SEQUENCE{*****
**
**      INDEPENDENT SEQUENCE      **
**      CONTROL BLOCK             **
**                               **
*****}

#define F1PRI :SEQUENCE:FEED1_PRI.VALUE
#define F2PRI :SEQUENCE:FEED2_PRI.VALUE
#define F3PRI :SEQUENCE:FEED3_PRI.VALUE
#define F4PRI :SEQUENCE:FEED4_PRI.VALUE
#define F5PRI :SEQUENCE:FEED5_PRI.VALUE

CONSTANTS {*****
* specify any Constants *
*      in the following format: *
*****
* 12_char_name = value ; *
*****}

SHED = TRUE;
NOTSHED = FALSE;

VARIABLES {*****
* specify any Block Local Variables *
*      in the following format: *
*****
* 12_char_name [ , 12_char_name ] : type.*
*
*where type is one of B, I, R, S, S12, S6*
*****}

USER_LABELS{*****
* specify any user labeled parameters *
*      in one of the following formats: *
*****
* 10_char_name : BI00nn; n = 01 -- 24 *
* 10_char_name : BO00nn; n = 01 -- 16 *
* 10_char_name : BA00nn; n = 1 -- 4 *
* 10_char_name : RI00nn; n = 01 -- 15 *
* 10_char_name : RO00nn; n = 01 -- 15 *
* 10_char_name : RA00nn; n = 1 -- 2 *
* 10_char_name : II00nn; n = 1 -- 8 *
* 10_char_name : IO00nn; n = 1 -- 5 *
* 10_char_name : IA0001; *
* 10_char_name : SN00nn; n = 01 -- 10 *
*****}

LOADSHED : BI0001;
/*FIRST_ACTIVE : BI0020;*/

F1 : BO0001;
F2 : BO0002;
F3 : BO0003;
F4 : BO0004;
F5 : BO0005;
LSC_ON : BO0006;

PRIMAX : RI0001;
DELAY1 : RI0002; /*FOR SIMULATION TEST ONLY*/

{*****
* Specify any Subroutines *
*****
*
* SUBROUTINE name ( formal arguments ) : *
* VARIABLES subr. local variables ; *
* STATEMENTS *
* *
* ENDSUBROUTINE *
*****}

{*****
* Specify any
* Standard Block Exception Handlers *
*****
*
* BLOCK_EXCEPTION exc_name [ DISABLE ] *
* STATEMENTS *
* *
* ENDEXCEPTION *
*****}

```

```

STATEMENTS{*****
*          Specify the statements here          *
*****}
/* IF FIRST_ACTIVE = TRUE THEN
   F1 := NOTSHED; F2 := NOTSHED; F3 := NOTSHED; F4 := NOTSHED; F5 := NOTSHED;
   FIRST_ACTIVE := FALSE;
   EXIT;
ENDIF;

/*<<START>>*/
WAIT DELAY1;

/*LOADSHED FUNCTION*/
IF (LOADSHED = TRUE) THEN
   LSC_ON := TRUE;

   IF PRIMAX = 0 THEN
      F1:=NOTSHED; F2:=NOTSHED; F3:=NOTSHED; F4:=NOTSHED; F5:=NOTSHED;
      GOTO A ;
   ENDIF ;

   /*CHECK FEEDER*/
   IF (F1PRI = PRIMAX) THEN
      F1 := SHED;
   ELSE F1 := NOTSHED;
   ENDIF ;

   IF (F2PRI = PRIMAX) THEN
      F2 := SHED;
   ELSE F2 := NOTSHED;
   ENDIF ;

   IF (F3PRI = PRIMAX) THEN
      F3 := SHED;
   ELSE F3 := NOTSHED;
   ENDIF ;

   IF (F4PRI = PRIMAX) THEN
      F4 := SHED;
   ELSE F4 := NOTSHED;
   ENDIF ;

   IF (F5PRI = PRIMAX) THEN
      F5 := SHED;
   ELSE F5 := NOTSHED;
   ENDIF ;

ELSE LSC_ON := FALSE;
   F1 := NOTSHED; F2 := NOTSHED; F3 := NOTSHED; F4 := NOTSHED; F5 := NOTSHED;

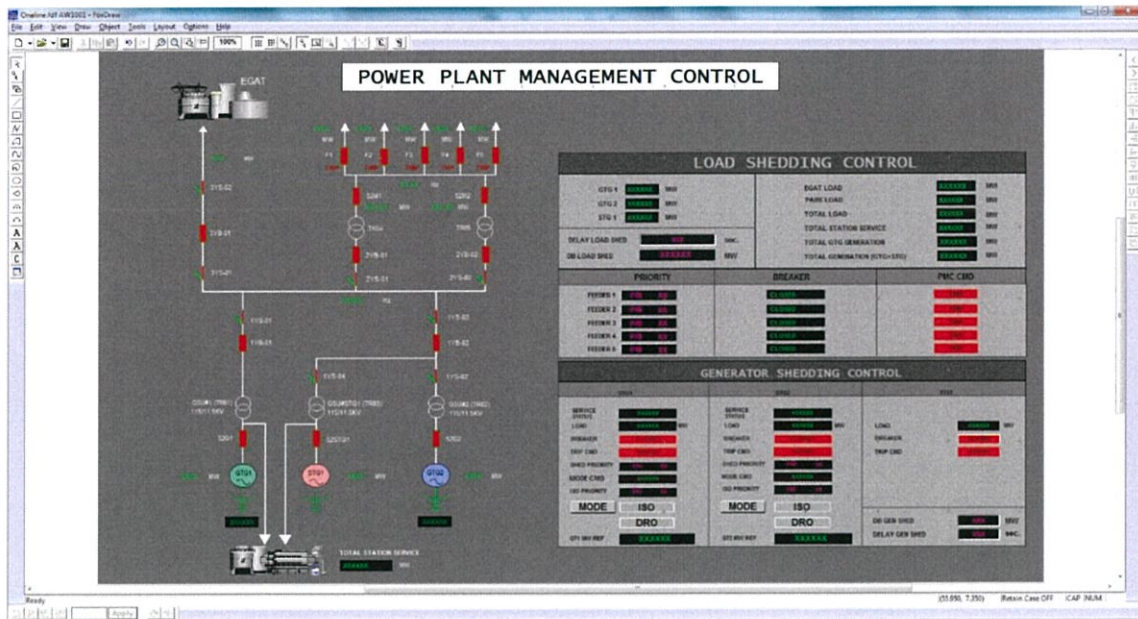
<<A>>

ENDIF;

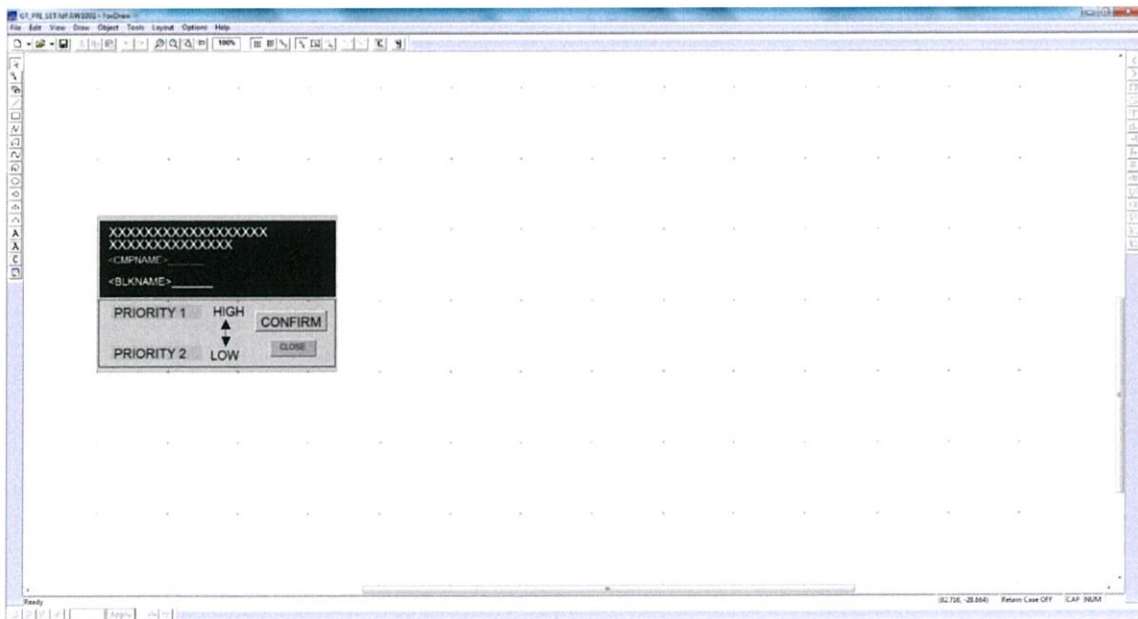
ENDSEQUENCE

```

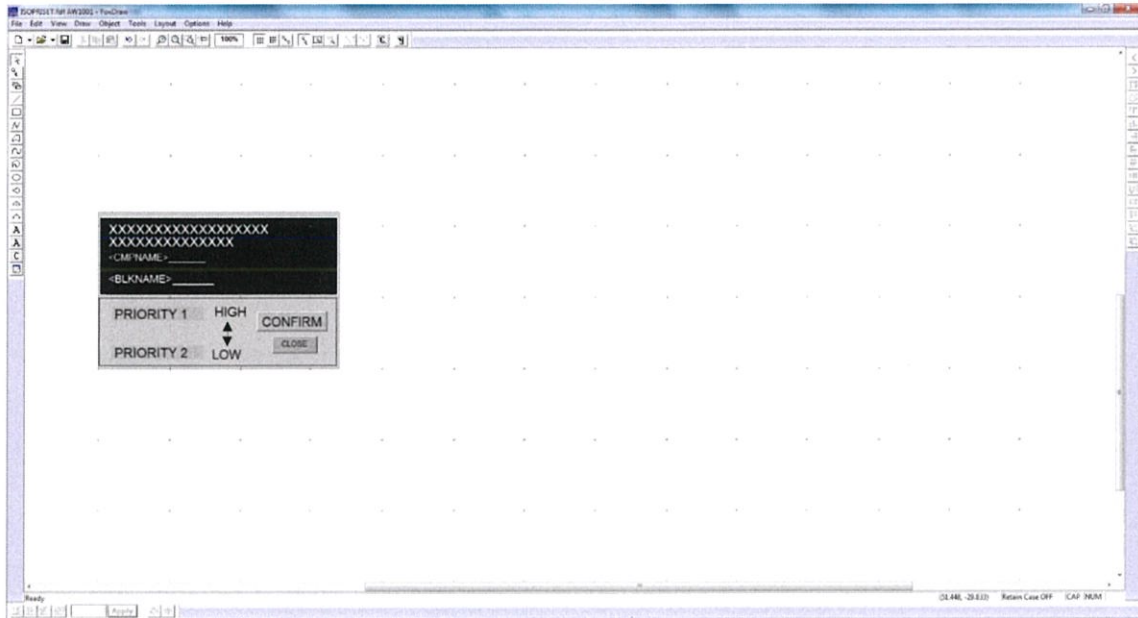
### 3.5 ส่วนประกอบการทำงานของ Plant Management Control ในโปรแกรม FOXDRAW



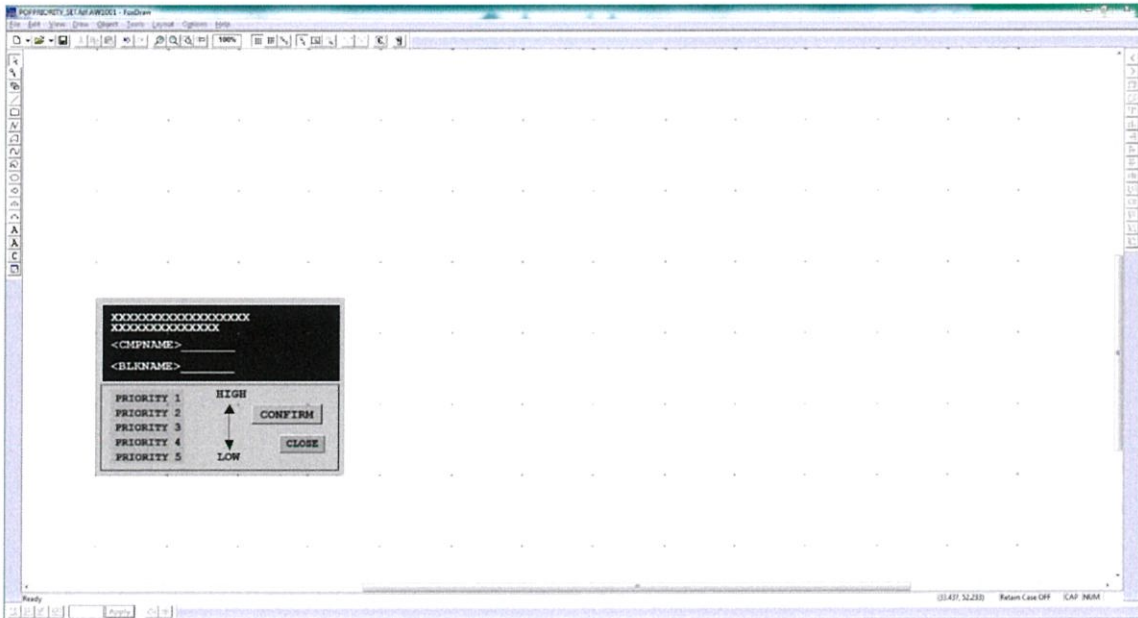
รูปที่ 3.41 FOXDRAW : PLANT MANAGEMENT CONTROL



รูปที่ 3.42 FOXDRAW : GT\_PRI\_SET

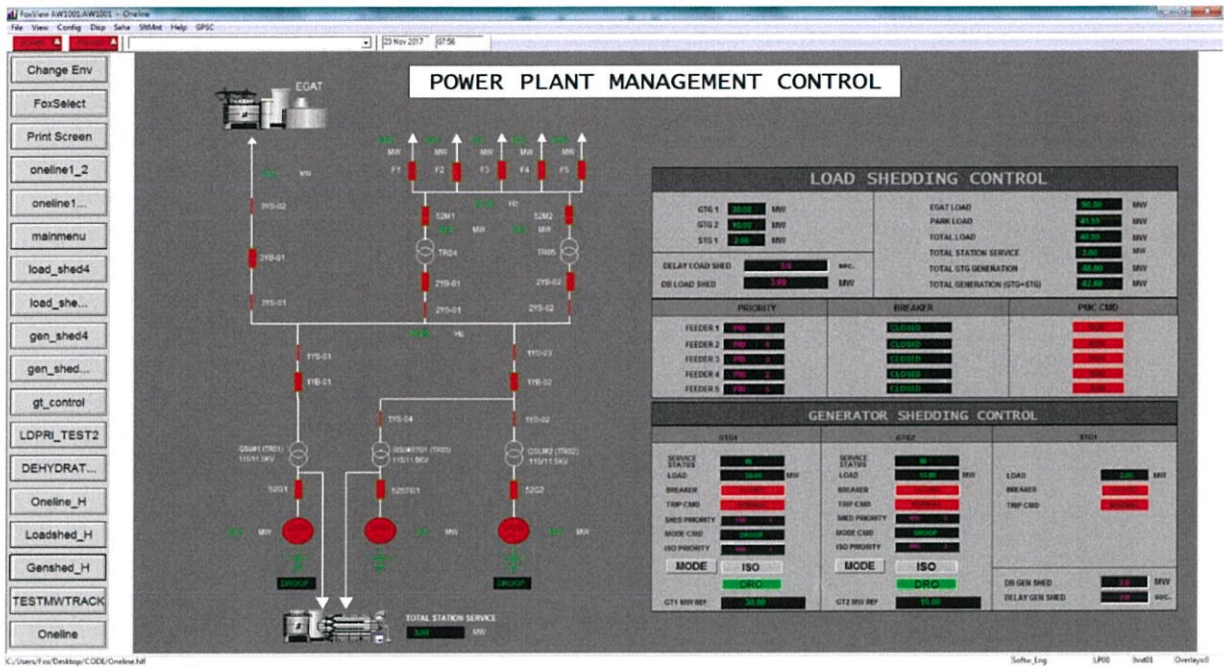


รูปที่ 3.43 FOXDRAW : ISOCHRONOUS MODEPRISET

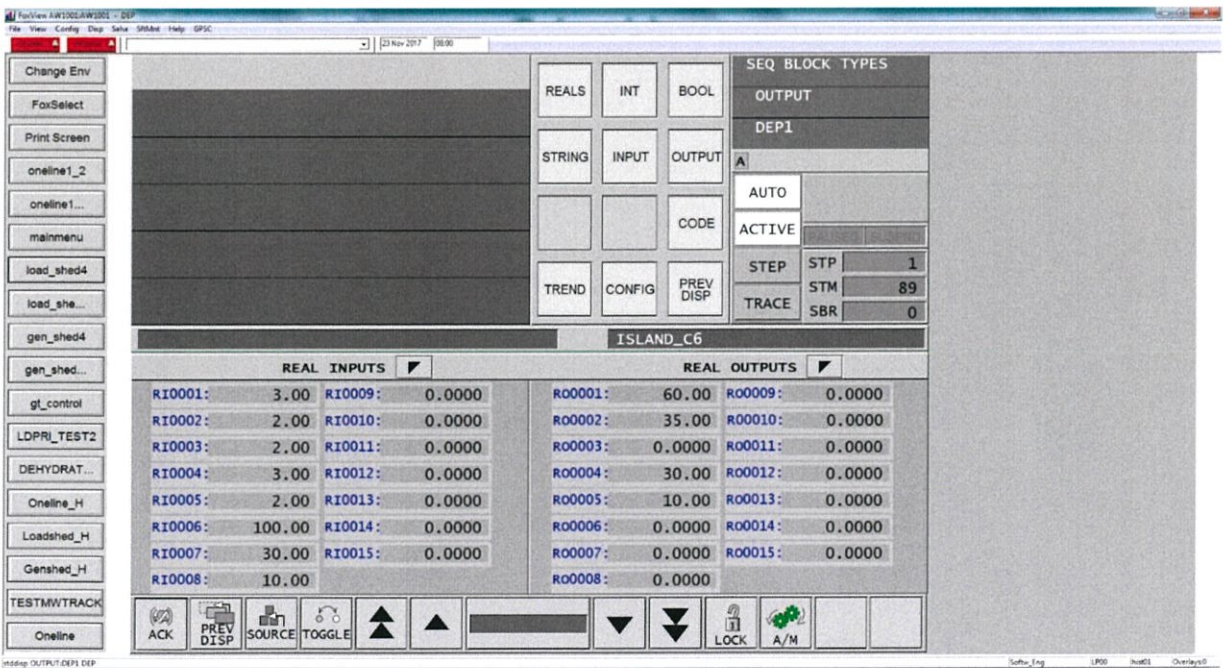


รูปที่ 3.44 FOXDRAW : POPPRIORITY

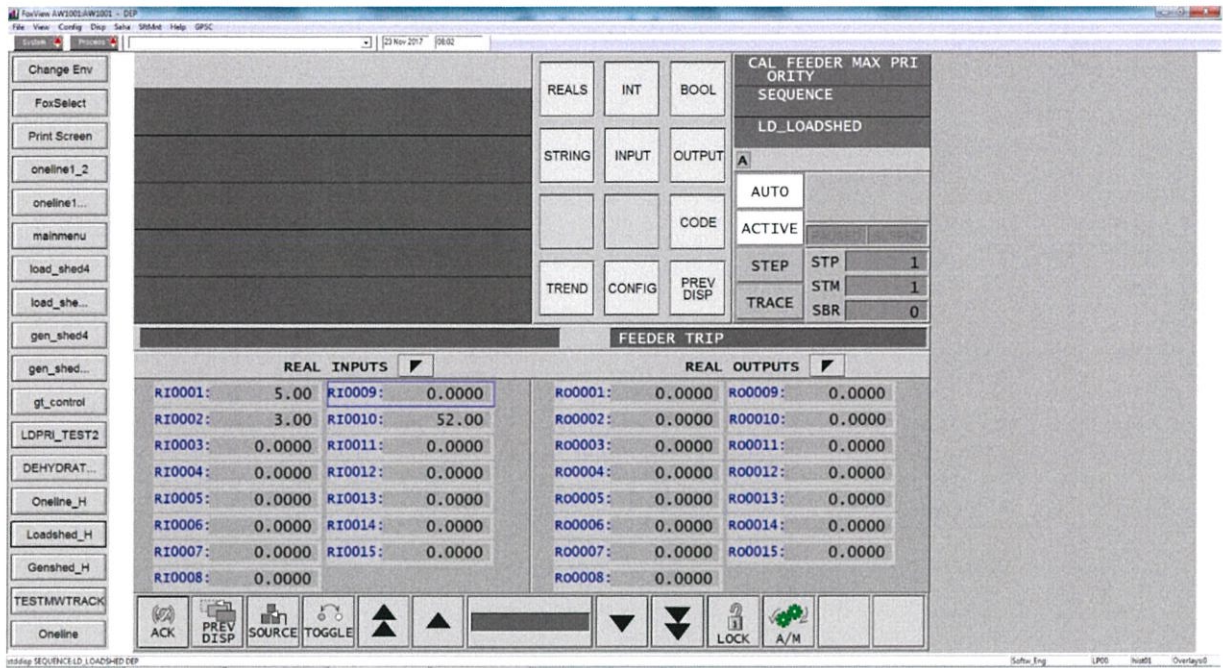
### 3.6 ส่วนประกอบการทำงานของ Plant Management Control ในโปรแกรม FOXVIEW



รูปที่ 3.45 FOXVIEW : แสดงผลการทำงานของ PLANT MANAGEMENT CONTROL



รูปที่ 3.46 FOXVIEW : แสดงผลการทำงานของ COMPOUND OUTPUT (Block DEP1)



รูปที่ 3.47 FOXVIEW : แสดงผลการทำงานของ COMPOUND SEQUENCE (Block LD\_LOADSHED)

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 กล่าวนำ

จากบทที่ 3 ได้มีการกล่าวถึงการดำเนินโครงการในขั้นตอนต่างๆ ของ โครงการงานการจำลองการปรับสมดุล อุปกรณ์และอุปกรณ์สำหรับโรงไฟฟ้าด้วย Foxboro SCP ดังนั้นบทนี้จึงขอกกล่าวถึงผลของการดำเนินงานทั้งหมด โดยแบ่งเป็น 2 ส่วนหลัก ๆ ดังนี้

##### 4.1.1 ส่วนซอฟต์แวร์

ซอฟต์แวร์ที่ใช้กับดีซีเอส Foxboro เพื่อให้ระบบสามารถออกแบบ ควบคุม และสังเกตการณ์การต่างๆ ได้ โดยซอฟต์แวร์ที่ใช้ในการดำเนินการโครงการ Plant Management Control Simulation คือ

##### 1.) ซอฟต์แวร์ SCP Launcher

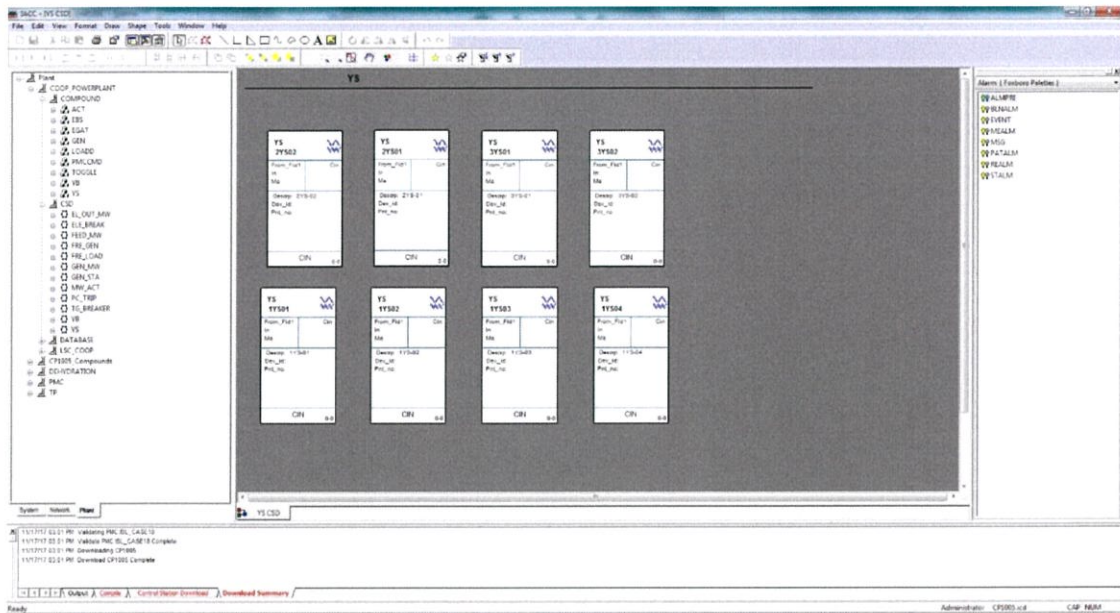
เป็นซอฟต์แวร์ที่จำลองการทำงานของฮาร์ดแวร์โดยที่เราไม่จำเป็นต้องมีฮาร์ดแวร์จริง



รูปที่ 4.1 RUN SCP Launcher

## 2.) ซอฟต์แวร์ I/A Series Configuration Component (IACC)

เป็นซอฟต์แวร์ที่ใช้ในการสร้าง แก๊ส และตั้งค่าให้กับบล็อกและเทมเพลตของระบบ



รูปที่ 4.2 บล็อกการทำงานบนซอฟต์แวร์ IACC

```
<<LAST_GN>>
GN_END = TRUE.
ENDIF.
/*WAIT TIME2
GN_END = TRUE */

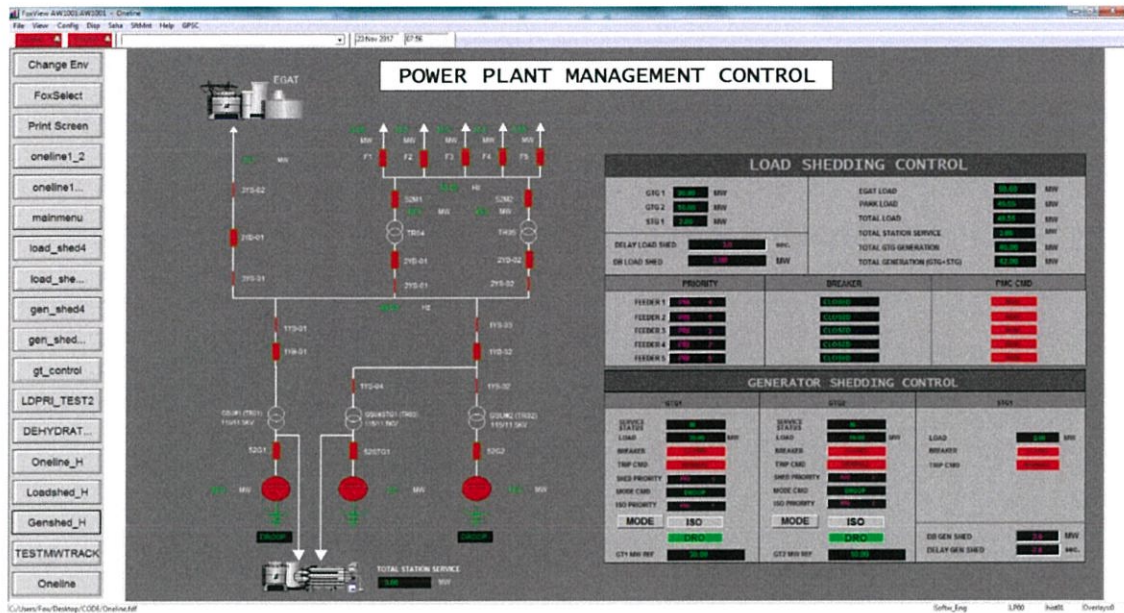
( ABC SW1 TOGGLE = FALSE )

/*FREQUENCYCTRL*/
IF FC_END=FALSE THEN
  IF (FREGEN < 49.9) THEN
    IF (GTG1STA = TRUE) AND (GT1_PMC_ISO = FALSE) THEN
      MUREF_GT1 = GT1_MAX.
      GT1_MODE = TRUE.
      ISOGT1 = FALSE.
    ENDIF.
    IF (GTG2STA = TRUE) AND (GT2_PMC_ISO = FALSE) THEN
      MUREF_GT2 = GT2_MAX.
      GT2_MODE = TRUE.
      ISOGT2 = FALSE.
    ENDIF.
    GOTO LAST_GN.
  ENDIF.
  IF (FREGEN > 50.1) THEN
    IF (GTG1STA = TRUE) AND (GT1_PMC_ISO = FALSE) THEN
      MUREF_GT1 = GT1_MIN.
      GT1_MODE = TRUE.
      ISOGT1 = FALSE.
    ENDIF.
    IF (GTG2STA = TRUE) AND (GT2_PMC_ISO = FALSE) THEN
      MUREF_GT2 = GT2_MIN.
      GT2_MODE = TRUE.
      ISOGT2 = FALSE.
    ENDIF.
    GOTO LAST_GN.
  ENDIF.
  IF (FREGEN >= 49.9) AND (FREGEN <= 50.1) THEN
    IF (GTG1STA = TRUE) AND (GT1_PMC_ISO = FALSE) AND PRIIS01 = 1 THEN
      GT1_MODE = FALSE.
      ISOGT1 = TRUE.
    ENDIF.
    IF (GTG2STA = TRUE) AND (GT2_PMC_ISO = FALSE) AND PRIIS02 = 1 THEN
      GT2_MODE = FALSE.
      ISOGT2 = TRUE.
    ENDIF.
    MUREF_GT1 = GTG1MW.
    MUREF_GT2 = GTG2MW.
  GOTO SEL_ISO.
ENDIF.
<<SEL_ISO>>
```

รูปที่ 4.3 โค้ดการทำงานบนซอฟต์แวร์ IACC

### 3.) ซอฟต์แวร์ FoxView

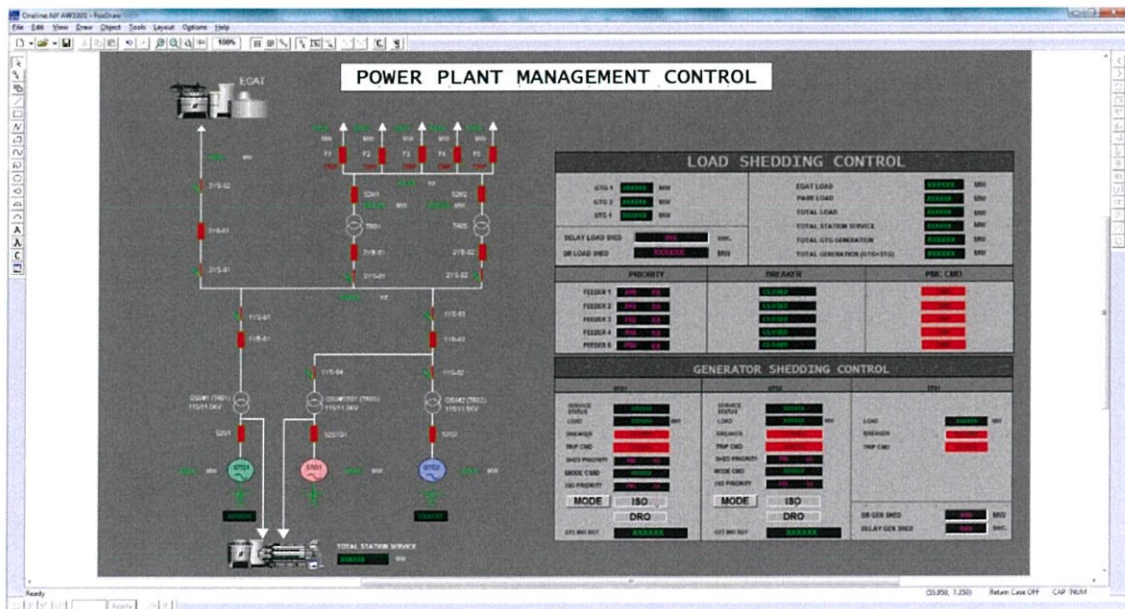
เป็นซอฟต์แวร์ที่ใช้ในการแสดงผลและควบคุมการทำงานบนหน้ากราฟิกของระบบ



รูปที่ 4.4 แสดงผลและควบคุมบนซอฟต์แวร์ FoxView

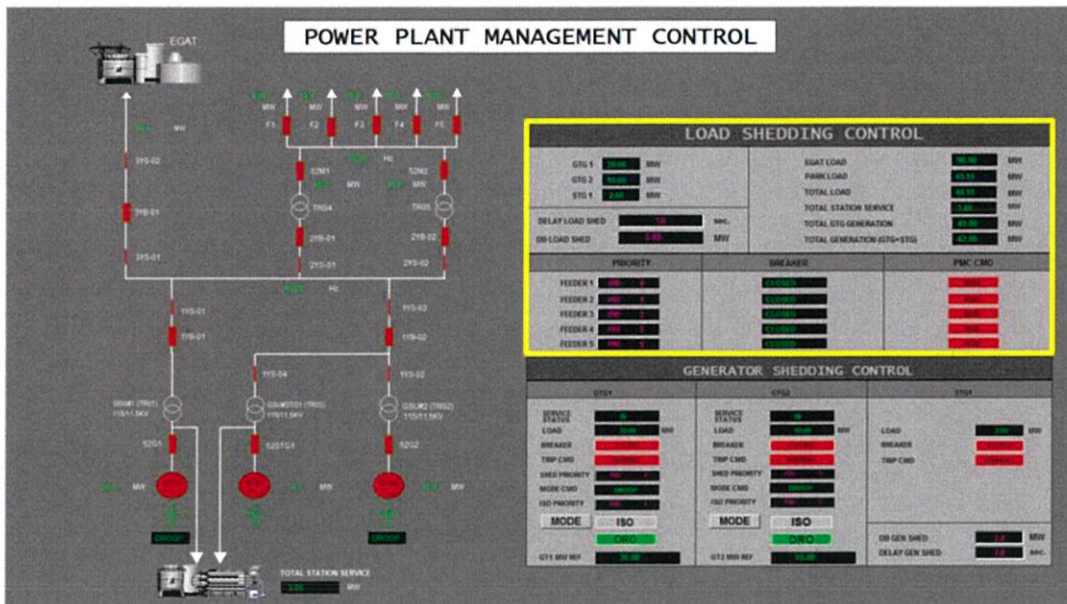
### 4.) ซอฟต์แวร์ FoxDraw

เป็นซอฟต์แวร์ที่ใช้ในการออกแบบและแก้ไขหน้ากราฟิกของระบบ



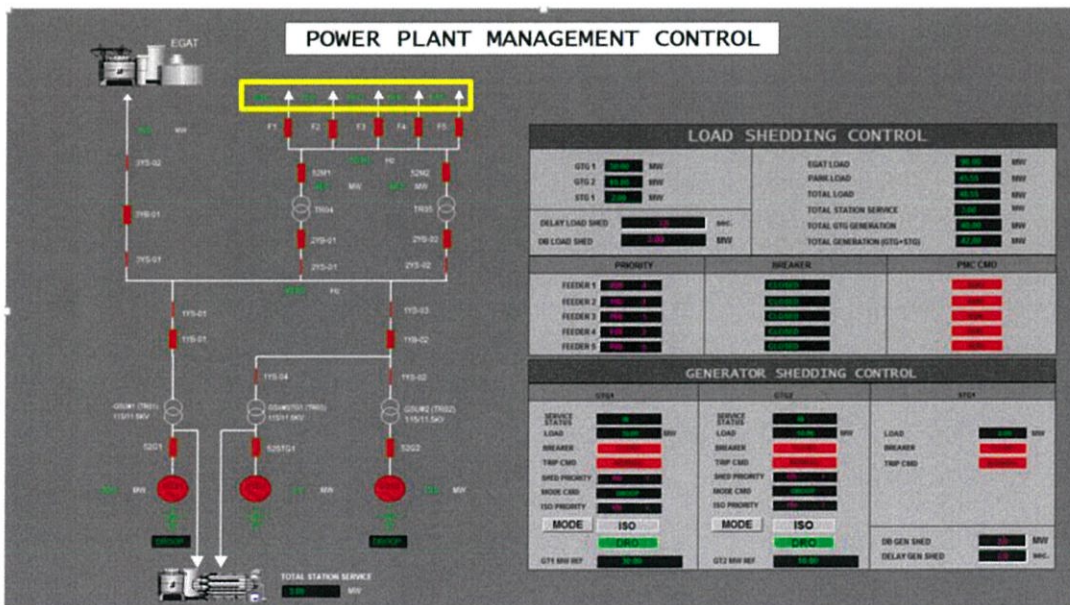
รูปที่ 4.5 ออกแบบและแก้ไขหน้ากราฟิกบนซอฟต์แวร์ FoxDraw

#### 4.1.2 ส่วนการแสดงผลและควบคุมการทำงาน



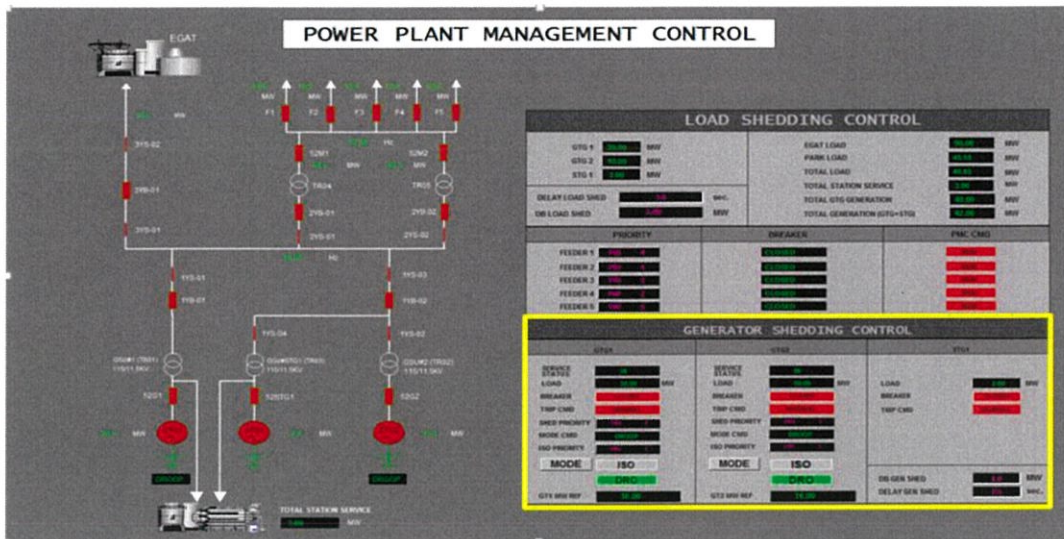
รูปที่ 4.6 LOAD SHEDDING CONTROL

จากรูปที่ 4.6 ส่วนนี้เป็นส่วนที่ใช้ในการควบคุม กำหนดค่าและแสดงผลต่างๆ ที่เกี่ยวข้องกับ Load ซึ่งสามารถกำหนดค่า Priority ในการ Load shed ได้ในส่วนของ Priority อีกทั้งยังมีการแจ้งเตือนเมื่อเกิดการ Trip ของ Load ในระบบตรงส่วน PMC CMD และการ open breaker ตรงส่วน BREAKER



รูปที่ 4.7 ส่วนการกำหนดค่าของการจ่ายพลังงานไฟฟ้าให้แก่โรงงานต่างๆ

จากรูปที่ 4.7 ส่วนนี้เป็นส่วนที่ใช้ในการกำหนดค่าการจ่ายพลังงานไฟฟ้าให้แก่โรงงานต่างๆ ที่เป็นลูกค้าของโรงไฟฟ้าแห่งนี้ ซึ่งสามารถกำหนดค่าได้ตามที่ต้องการขาย



รูปที่ 4.8 GENERATOR SHEDDING CONTROL

จากรูปที่ 4.8 ส่วนนี้เป็นส่วนที่ใช้ในการควบคุม กำหนดค่า และแสดงผลต่างๆ ที่เกี่ยวข้องกับ Generator ซึ่งในส่วนนี้สามารถกำหนดค่า Priority ในการ Generator shed ได้ในส่วน SHED PRIORITY แจ้งเตือนเมื่อเกิดการ Trip ของ Generator ในระบบตรงส่วน TRIP CMD และการ open breaker ตรงส่วน BREAKER อีกทั้งยังสามารถกำหนด Priority ของการเป็น ISOCHRONOUS MODE ได้ตรงส่วน ISOCHRONOUS MODE PRIORITY พร้อมแสดงผลเมื่อ Generator เมื่อตัวใดเป็น ISOCHRONOUS MODE

#### 4.2 ผลการดำเนินงาน

หลังจากที่เขียน logic การทำงาน สร้างและแก้ไขกราฟิกโดยโปรแกรม FoxView และ FoxDraw จากนั้นทดลองโปรแกรม ผลการดำเนินการของโครงการการจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP นั้น ระบบสามารถทำการควบคุมและแสดงผลการ Load Shed และ Generator Shed ได้ โดยระบบจะพยายามทำให้ค่า Load กับค่า Generator มีค่าใกล้เคียงกันมากที่สุด และระบบจะเริ่มทำงานเมื่อ 3YB มัน OPEN ซึ่งในการทำงานของ Load shed จะทำงานก็ต่อเมื่อค่า Load มีค่ามากกว่าค่า Generator (Load มาก = ความถี่ต่ำ, รวนต่ำ) ส่วนในการทำงานของ Generator shed จะทำงานก็ต่อเมื่อค่า Generator มีค่ามากกว่าค่า Load (Load น้อย = ความถี่สูง, รวนสูง) ในการ shed ต่างๆ นั้นจะ shed ตามลำดับ priority ที่เราได้กำหนดไว้

## บทที่ 5

### สรุปผลและปัญหาในการดำเนินโครงการ

#### 5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานโครงการเรื่องการจำลองการปรับสมดุลอุปสงค์และอุปทานสำหรับโรงไฟฟ้าด้วย Foxboro SCP ดำเนินกระบวนการต่างๆด้วยการใช้ดีซีเอสของ Foxboro และทำการสร้างออกแบบการควบคุมผ่านทางโปรแกรม FoxDraw และเชื่อมต่อกระบวนการของ Process กับผู้ใช้งานผ่านทางโปรแกรม FoxView โดยผู้ใช้งานนั้นสามารถควบคุมกระบวนการต่างๆ ผ่านทางหน้าจอแสดงผล ซึ่งในโครงการนี้ระบบสามารถดำเนินการคิด Load shed และ Generator shed ได้ตามวัตถุประสงค์ของการทำโครงการ อีกทั้งระบบยังช่วยเพิ่มเสถียรภาพของทรัพยากรที่ใช้ในการผลิตกระแสไฟฟ้า เพื่อไม่ให้เกิดการดับของไฟฟ้าและการหยุดชะงักที่อาจเกิดขึ้นได้ หรือเหตุการณ์ที่เครื่องกำเนิดไฟฟ้าผลิตไฟฟ้ามากเกินไปเกินความต้องการทางไฟฟ้า อีกทั้งยังช่วยจัดการการจ่ายไฟฟ้าให้เหมาะสมกับผู้ใช้งานในแต่ละช่วงเวลาและช่วย safe อุปกรณ์ให้ทำงานได้อย่างต่อเนื่อง

#### 5.2 ปัญหาและอุปสรรค

เนื่องจากระบบดีซีเอส และภาษาที่ใช้ในการเขียนโค้ดนั้นเป็นสิ่งที่ผู้จัดทำไม่เคยใช้มาก่อน จึงต้องใช้เวลาทำความเข้าใจเป็นอย่างมาก อีกทั้งข้อมูลเกี่ยวกับเรื่องระบบ Plant Management Control นั้น ค่อนข้างหายาก จึงทำให้เกิดความล่าช้าในการปฏิบัติงานและต้องรบกวนเวลาในการทำงานของผู้ให้คำปรึกษา

#### 5.3 ข้อเสนอแนะ

ในการออกแบบระบบ Plant Management Control นั้น ควรทำการศึกษาหาข้อมูลเกี่ยวกับเรื่องของการควบคุมการผลิตและจ่ายพลังงานไฟฟ้าของโรงไฟฟ้าให้ดีและมากที่สุด ก่อนจะทำการออกแบบและควบคุมเพื่อความถูกต้อง ลดความผิดพลาด ประหยัดเวลา และตรงตามวัตถุประสงค์ของการจัดทำระบบ

## เอกสารอ้างอิง

[1] โหลดไฟฟ้าและการจำแนกโหลด

Available : <https://powerplant2.wordpress.com/โหลดไฟฟ้าและการจำแนกโหลด>

[2] รูปแบบของโรงไฟฟ้าพลังงานร่วม (Co-generation plant)

Available : <https://www.gotoknow.org/posts/183558>

[3] ความรู้เกี่ยวกับไฟฟ้า

Available : <http://www.touchtechdesign.com/eppo>

[4] ความรู้เกี่ยวกับ Generator เครื่องกำเนิดไฟฟ้า

Available : <http://chuphoticups.blogspot.com/2014/03/generator.html>

[5] โคนเจนเนอร์เรชั่น ระบบการผลิตพลังงานประสิทธิภาพสูง

Available : <http://www.thaitextile.org/index.php/blog/2014/10/nnnn>

[6] ธุรกิจผลิตไฟฟ้า

Available : <http://gkenginest.blogspot.com/2013/05/electric-power-production-2.html>

[7] โครงข่ายไฟฟ้าอัจฉริยะ (Smart Grid)

Available : <http://www.ee.eng.chula.ac.th/smartgrids/index.php/smart-grids/80--smart-grids/smart-grid-doc/72-smart-grid>

[8] ระบบไฟฟ้าสมาร์ทกริดคืออะไร

Available : <http://www.thai-smartgrid.com/ระบบไฟฟ้าสมาร์ทกริดคืออะไร/>