

โครงการสาธิตไมโครกริดของศูนย์นวัตกรรมระบบพลังงาน
CInES Microgrid Demonstration : Operation and Control

โดย

นาย พงศ์ภรณ์	เตชะชลประเสริฐ
นาย พยุงศักดิ์	แซ่หลิว
นาย ไพโรจน์	คอนเอม
นาย มหาราช	ราชสีห์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2555

CInES Microgrid Demonstration : Operation and Control

PONGPORN

TAECHACHONPRASERT

PHAYUNGSAK

SAE-LIEW

PHAIROTE

KHON-EM

MAHARAJ

RATCHASI

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN ELECTRICAL ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2012

ปีการศึกษา 2555

โครงการสาธิตไมโครกริดของศูนย์นวัตกรรมระบบพลังงาน
CInES Microgrid Demonstration : Operation and Control

โดย

นาย พงศ์ภรณ์	เตชะชลประเสริฐ
นาย พยุงศักดิ์	แซ่หลิว
นาย ไพโรจน์	คอนแอม
นาย มหาราช	ราชสีห์

อาจารย์ที่ปรึกษา

ดร. สมภพ ผลไม้

ปริญญาานิพนธ์ปีการศึกษา 2555

สาขาวิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โครงการสาธิตไมโครกริดของศูนย์นวัตกรรมระบบพลังงาน

ผู้จัดทำ

- | | |
|-------------------|----------------|
| 1. นาย พงศ์ภรณ์ | เตชะชลประเสริฐ |
| 2. นาย พยุงศักดิ์ | แซ่หลิว |
| 3. นาย ไพโรจน์ | คอนแอม |
| 4. นาย มหาราช | ราชสีห์ |



(ดร. สมภพ ผลไม้)

อาจารย์ที่ปรึกษา

โครงการสาธิตไมโครกริดของศูนย์นวัตกรรมระบบพลังงาน

นายพงศ์ภรณ์ เตชะชลประเสริฐ
นายพยุงศักดิ์ แซ่หลิว
นายไพโรจน์ คอนอม
นายมหาราช ราชสีห์
ดร. สมภพ ผลไม้ อาจารย์ที่ปรึกษา
ปีการศึกษา 2555

บทคัดย่อ

ปฏิญานิพนธ์นี้นำเสนอระบบไมโครกริดสาธิตขนาด 3 kW ของศูนย์นวัตกรรมระบบพลังงาน โดยมีวัตถุประสงค์เพื่อศึกษา ติดตั้ง และควบคุมระบบไมโครกริด ในปฏิญานิพนธ์นี้ได้แบ่งการศึกษาออกเป็นสองส่วนคือ ส่วนของการศึกษาระบบควบคุมไมโครกริดในสถานะแยกอิสระจากกริดซึ่งควบคุมระบบด้วยเทคนิค Angle Droop และ Voltage Droop ซึ่งเป็นวิธีการในการควบคุมของแหล่งกำเนิดไฟฟ้าขนาดเล็ก โดยในการศึกษาในส่วนนี้ได้ทำการจำลองระบบไมโครกริดแบบแยกอิสระจากกริดทั้งระบบสามเฟสและหนึ่งเฟสเพื่อยืนยันความถูกต้องของระบบควบคุมที่ได้ออกแบบ ในส่วนของการทดลองติดตั้งระบบผลิตไฟฟ้าจากเซลล์แสงอาทิตย์ได้ทำการติดตั้งแผงโซลาร์เซลล์ที่รวม 3kW และใช้อินเวอร์เตอร์เฟสเดียวต่อเชื่อมกับระบบไฟฟ้า จากนั้นได้พัฒนาระบบตรวจวัดระยะไกลผ่านการสื่อสาร Modbus ขึ้นมาสำหรับการเก็บข้อมูลและควบคุมในอนาคต

CInES Microgrid Demonstration : Operation and Control

Pongporn Taechachonprasert
Phayungsak Sae-Liew
Phairote Khon-em
Maharaj Ratchasi
Dr. Sompob Polmai Supervisor
Year 2012

Abstract

This project presents CInES 3-kW microgrid demonstration system. The objectives are to study the installation and control of the microgrid system. In this project, the studies are divided into two parts. The first part concerns about control of a microgrid in stand-alone mode using angle and voltage droop techniques. Three-phase and single-phase microgrids are modeled to confirm the validation of the techniques. In the second part, 3-kW solar panels have been installed for evaluation at KMITL. A single-phase solar inverter is used to supply the generated power to the power system. A modbus communication network has been created for data acquisition and control of the microgrid in the future.

กิตติกรรมประกาศ

ในการทำโครงการนี้ผู้จัดทำโครงการได้รับความกรุณาจาก ดร.สมภพ ผลไม้ ซึ่งเป็นอาจารย์ที่ปรึกษาได้ให้คำแนะนำต่างๆและความอนุเคราะห์จากท่านอาจารย์เป็นอย่างมาก ถ้าหากปราศจากท่านโครงการนี้ก็คงไม่สำเร็จลุล่วงไปด้วยดี จึงขอกราบขอบพระคุณท่านเป็นอย่างสูง

นอกจากนั้นผู้จัดทำโครงการขอกราบขอบพระคุณคณาจารย์ ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ รวมทั้งคณาจารย์ในภาควิชาอื่นๆในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกๆท่านที่ได้ประสิทธิ์ประสาทวิชาให้กับข้าพเจ้าตลอดการศึกษาในระดับปริญญาตรี

ขอขอบคุณเพื่อนๆพี่ๆในห้องวิจัย Power Electronic Application and Energy Conversion Laboratory (PEARL LAB) ห้อง ECC-407 ในภาควิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่คอยร่วมทุกข์ร่วมสุขคอยให้กำลังใจและช่วยเหลือผู้จัดทำโครงการ ตลอดระยะเวลาที่ผ่านมา

ขอขอบคุณ พี่ๆเพื่อนๆน้องๆ ในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกคนที่ให้คำแนะนำต่างๆและช่วยเหลือกำลังใจให้ผู้จัดทำเสมอมา

สุดท้ายนี้ผู้จัดทำโครงการขอกราบขอบพระคุณ ครูบาอาจารย์ บิดา มารดา และญาติมิตรของข้าพเจ้าที่ให้คำสั่งสอน และเป็นกำลังใจจากครอบครัวในการให้การสนับสนุนในทุกเรื่องๆ ซึ่งทำให้กลุ่มผู้จัดทำโครงการ เสร็จลุล่วงไปได้ด้วยดี กลุ่มผู้จัดทำขอมอบคุณงามความดีและประโยชน์จากโครงการเล่มนี้ ให้แก่ผู้มีพระคุณทุกท่าน

สารบัญ

	หน้า
บทคัดย่อ.....	I
ABSTRACT.....	III
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญรูป.....	VII
สารบัญตาราง.....	X
บทที่ 1 คำนำ.....	1
1.1 ความเป็นมาและความสำคัญ.....	1
1.2 วัตถุประสงค์ของโครงการ.....	2
1.3 ขอบเขตของการศึกษา.....	2
1.4 แผนการดำเนินโครงการ.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.6 เนื้อหาของปริญญานิพนธ์.....	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง.....	4
2.1 ระบบไมโครกริด.....	4
2.1.1 ความหมายและความเป็นมาของไมโครกริด.....	4
2.1.2 ระบบผลิตไฟฟ้าแบบกระจายศูนย์.....	5
2.1.3 หน้าที่และส่วนประกอบต่างๆของไมโครกริด.....	6
2.1.4 แหล่งพลังงานในไมโครกริด.....	8
2.1.5 ลักษณะต่างๆของไมโครกริด.....	9
2.2 เทคนิค Frequency Droop และ Voltage Droop.....	11
2.2.1 การควบคุมกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือน.....	11
2.2.2 Voltage Droop Control.....	12
2.2.3 Frequency Droop Control.....	13
2.3 วิธีการควบคุมคอนเวอร์เตอร์.....	14
2.3.1 โมดูลการควบคุมมุมแรงดัน.....	15
2.3.2 โมดูลควบคุมขนาดแรงดัน.....	16
2.4 MODBUS Protocol.....	16
2.4.1 MODBUS RTU Mode.....	17
2.4.2 MODBUS ASCII Mode.....	18
บทที่ 3 การออกแบบและทดลองระบบผลิตไฟฟ้าจากเซลล์แสงอาทิตย์.....	20
3.1 แนะนำโปรแกรม PV Syst Ver.5.59 เบื้องต้น.....	20
3.1.1 การออกแบบเบื้องต้น.....	20
3.1.2 การออกแบบโครงการ.....	21

สารบัญ (ต่อ)

	หน้า
3.1.3 การวิเคราะห์ข้อมูลที่วัดได้.....	22
3.1.4 เครื่องมือ.....	22
3.2 การออกแบบการต่อแผง PV เข้ากับ Inverter โดยใช้โปรแกรม.....	23
3.3 การออกแบบการทำงานของไมโครกริดในโหมด stand-alone ของระบบ 3 เฟส.....	32
3.3.1 กำหนดรายละเอียดสำหรับไมโครกริดระบบ 3 เฟสที่จะใช้ในการทดลอง.....	32
3.3.2 ออกแบบ phase lock loop ชนิด 3 เฟส.....	33
3.3.3 ออกแบบ frequency droop coefficient.....	33
3.3.4 ออกแบบ subsystem ที่ชื่อว่า P&Q calculation.....	34
3.3.5 ออกแบบการแปลงแกนของกระแส 3 เฟสเป็นแกน d-q.....	34
3.3.6 ออกแบบ voltage droop coefficient.....	35
3.3.7 ออกแบบระบบควบคุมแรงดันของระบบ.....	35
3.3.8 สร้างระบบ 3 เฟสโดยใช้เทคนิคการเลื่อนเฟสออกไป 120 องศาไฟฟ้า.....	36
3.3.9 Droop Control 1 subsystem.....	36
3.3.10 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 3 เฟส.....	37
3.4 การออกแบบการทำงานของไมโครกริดในโหมด stand-alone ของระบบ 1 เฟส.....	37
3.4.1 กำหนดรายละเอียดสำหรับไมโครกริดระบบ 1 เฟสที่จะใช้ในการทดลอง.....	37
3.4.2 การออกแบบ phase lock loop ชนิด 1 เฟส.....	38
3.4.3 การใช้ phase shifter ในการแปลงแกนระบบ 1 เฟส.....	38
3.4.4 การออกแบบ subsystem ที่ชื่อว่า P&Q calculation.....	39
3.4.5 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 1 เฟส.....	39
บทที่ 4 การทดลองผลการทดลอง.....	40
4.1 แบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ของระบบ 3 เฟส.....	40
4.2 ผลการทดลองแบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 3 เฟส.....	41
4.2.1 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F = 1.....	43
4.2.2 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F = 0.9.....	44
4.2.3 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F = 0.8.....	45
4.3 แบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 1 เฟส.....	46
4.4 ผลการทดลองแบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 1 เฟส.....	47
4.4.1 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F = 1.....	49

สารบัญ (ต่อ)

	หน้า
4.4.2 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =0.9.....	51
4.4.3 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =0.8.....	52
4.5 ผลการทดลองของ Modbus.....	53
4.6 ผลการทดลองติดตั้ง Grid-Connected Photovoltaic Power Generation ขนาด 3 kW.....	59
4.6.1 รูปคลื่นแรงดันและกระแสของอินเวอร์เตอร์.....	59
4.6.2 ผลการจ่ายพลังงานไฟฟ้ารายวัน/เดือน.....	59
บทที่ 5 บทสรุปและข้อเสนอแนะ.....	63
บรรณานุกรม.....	64
ภาคผนวก.....	65
ประวัติผู้เขียน.....	100

สารบัญรูป

รูปที่		หน้า
2.1	ส่วนประกอบต่างๆของไมโครกริด.....	6
2.2	แสดงโมเดลอย่างง่ายของ Microsource.....	11
2.3	กราฟคุณลักษณะของVoltage-Reactive Power (V-Q) droop.....	12
2.4	กราฟแสดงคุณลักษณะของกำลังจริงเทียบกับความถี่.....	13
2.5	กรอบอ้างอิงของตัวควบคุมของ microsource.....	14
2.6	ภาพรวมของการควบคุม.....	15
2.7	รูปการควบคุมของมุมแรงดัน.....	16
2.8	การติดต่อสื่อสารแบบ Master/Slave.....	17
2.9	ลักษณะเฟรมข้อมูลของ MODBUS RTU MODE.....	18
2.10	ลักษณะข้อมูลแต่ละไบต์ของ MODBUS RTU.....	18
2.11	ลักษณะเฟรมข้อมูลของ MODBUS ASCII.....	19
2.12	ลักษณะข้อมูลแต่ละไบต์ของ MODBUS ASCII.....	19
3.1	แบ่งระดับการศึกษาของ PVsyst Ver.5.59.....	20
3.2	การออกแบบเบื้องต้น.....	21
3.3	การออกแบบโครงงาน.....	22
3.4	เครื่องมือ.....	22
3.5	โหมด Project design แบบ grid connected.....	23
3.6	ค่าพิกัดต่างๆถูกตั้งค่าไว้ในโปรแกรม.....	23
3.7	พิกัดตึก L จาก Google Earth.....	24
3.8	กำหนดลักษณะการติดตั้งแผง PV.....	24
3.9	PV แบบ Thin Film รุ่น NA-F121GK.....	25
3.10	กราฟการแสดงผลงาน.....	26
3.11	รูปแบบการต่อ PV.....	26
3.12	การคำนวณหาขนาดพื้นที่.....	27
3.13	ค่าที่ได้จากการ Simulation.....	27
3.14	PV แบบ Polycrystalline รุ่น ND-130T1J.....	28
3.15	กราฟแสดงการทำงาน.....	29
3.16	รูปแบบการต่อ PV.....	30
3.17	คำนวณหาขนาดพื้นที่ที่ต้องการในการติดตั้ง.....	30
3.18	ทำการ Simulation หาค่าพลังงานที่ได้ต่อปี.....	31
3.19	วงจรสมมูลเฟสเดียวของระบบที่ใช้ในการทดลอง.....	32

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.20 ภายใน abc-dq Voltage With 3 phase PLL subsystem.....	33
3.21 ภายใน angle droop control subsystem.....	33
3.22 ภายใน P&Q calculation subsystem.....	34
3.23 ภายใน abc-dq current subsystem.....	34
3.24 ภายใน QV droop characteristic subsystem.....	35
3.25 ภายใน Voltage Generation subsystem.....	35
3.26 ภายใน Droop Control 1 subsystem.....	36
3.27 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 3 เฟสที่พร้อมใช้งาน.....	36
3.28 วงจรสมมูลเฟสเดียวของระบบที่ใช้ในการทดลอง.....	37
3.29 การใช้ phase shifter ในการแปลงแกนระบบ 1 เฟส.....	37
3.30 ภายใน P&Q Calculation subsystem.....	39
3.31 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 1 เฟสที่พร้อมใช้งาน.....	39
3.32 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 1 เฟสที่พร้อมใช้งาน.....	39
4.1 แบบจำลองไมโครกริด 3 เฟสในโหมด stand-alone.....	40
4.2 กำลังไฟฟ้าจริงและเสมือนของคอนเวอร์เตอร์ 1, 2 และกำลังไฟฟ้ารวมตามลำดับ.....	41
4.3 ความถี่ที่จุดต่อร่วม (Point of common, PCC).....	42
4.4 แรงดันและกระแสที่จุดต่อร่วม PCC.....	42
4.5 แบบจำลองไมโครกริด 1 เฟส ในโหมด stand-alone.....	46
4.6 กำลังไฟฟ้าจริงและเสมือนของคอนเวอร์เตอร์ 1, 2 และกำลังไฟฟ้ารวมตามลำดับ.....	48
4.7 ความถี่ที่จุดต่อร่วม (Point of common, PCC).....	48
4.8 แรงดันและกระแสที่จุดต่อร่วม PCC.....	49
4.9 Voltage line1.....	53
4.10 Voltage line1-2 , line3-1.....	53
4.11 Current line1.....	54
4.12 Active Power line1.....	54
4.13 Apparent Power line1.....	54
4.14 Reactive Power line1.....	55
4.15 ผลรวม W,KVA,VAR.....	55
4.16 Power Factor.....	55
4.17 Frequency.....	56
4.18 Real Energy (Wh)	56

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.19 Reactive Energy (VARh).....	56
4.20 หน้าจอแสดงผลของโปรแกรม.....	57
4.21 คลื่นแรงดันและกระแสของอินเวอร์เตอร์ขณะก่อนต่อเข้าหม้อแปลง.....	59
4.22 คลื่นแรงดันและกระแสของอินเวอร์เตอร์ขณะหลังต่อออกจากหม้อแปลง.....	59
4.23 กราฟค่าพลังงานไฟฟ้า (kWh) กับเวลา.....	61
4.24 กราฟค่าพลังงานไฟฟ้า (kWh) กับเวลา.....	61
4.25 กราฟค่าพลังงานไฟฟ้า (kWh) ที่ได้ในแต่ละวัน ประจำเดือนมกราคม.....	62
4.26 กราฟค่าพลังงานไฟฟ้า (kWh) ที่ได้ในแต่ละวัน ประจำเดือนกุมภาพันธ์.....	62

สารบัญตาราง

ตารางที่	หน้า
3.1 ตารางแสดงค่าพารามิเตอร์ของวงจรสมมูลที่ใช้ในการทดลอง.....	32
3.2 ตารางแสดงค่าพารามิเตอร์ของวงจรสมมูลที่ใช้ในการทดลอง.....	38
4.1 พารามิเตอร์ของคอนเวอร์เตอร์.....	40
4.2 การตั้งค่าการเปลี่ยนแปลงทางด้านโหลดเพื่อพิจารณาผลตอบสนอง.....	41
4.3 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 6 kW).....	43
4.4 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 6 kW + Load R ที่ค่าต่างๆ).....	43
4.5 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์.....	44
4.6 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 6kW).....	44
4.7 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 6kW และ Load R ที่ค่าต่างๆ).....	44
4.8 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์.....	45
4.9 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 6kW).....	45
4.10 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R6,000W + Load Rที่ค่าต่างๆ).....	45
4.11 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์.....	46
4.12 พารามิเตอร์ของคอนเวอร์เตอร์.....	47
4.13 การตั้งค่าการเปลี่ยนแปลงทางด้านโหลดเพื่อพิจารณาผลตอบสนอง.....	47
4.14 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 1,200W).....	50
4.15 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 1,200W + Load R ที่ค่าต่างๆ).....	50
4.16 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์.....	50
4.17 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 1,200W).....	51
4.18 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 1,200W + Load RL).....	51
4.19 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์.....	51
4.20 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 1,2000W).....	52
4.21 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 1,200W + Load RL ที่ค่าต่างๆ)...	52
4.22 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์.....	53
4.23 ค่าพลังงานไฟฟ้า (kWh) ที่ได้ระหว่าง 9 มกราคม ถึง 17 กุมภาพันธ์ พ.ศ. 2556.....	60

บทที่ 1 คำนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันพลังงานเป็นปัจจัยพื้นฐานที่สำคัญมาก การผลิตไฟฟ้าในปัจจุบันเป็นการผลิตแบบธรรมดา คือ การผลิตโดยใช้เชื้อเพลิงฟอสซิลจำพวกถ่านหิน น้ำมันดีเซลหรือก๊าซธรรมชาติเป็นหลัก การผลิตไฟฟ้าในรูปแบบนี้จะเป็นการผลิตไฟฟ้าในปริมาณมากจากโรงจักรไฟฟ้าขนาดใหญ่เพื่อส่งจ่ายไปยังระบบจำหน่าย ซึ่งเป็นการจ่ายไฟฟ้าแบบทิศทางเดียว(unidirection) จากผู้ผลิตสู่ผู้บริโภค ดังนั้น ปริมาณไฟฟ้าที่จะผลิตจะต้องสัมพันธ์กับภาระความต้องการใช้ไฟฟ้าของผู้บริโภค ซึ่งมีความหลากหลายทั้งในแง่ของช่วงเวลาหรือตำแหน่งของผู้บริโภค กำลังไฟฟ้าที่เปลี่ยนแปลงอยู่ตลอดเวลาและการส่งจ่ายกำลังไฟฟ้าไปในระยะไกลๆโดยไม่มีการสูญเสีย ทำให้การผลิตไฟฟ้าจากโรงจักรไฟฟ้าขนาดใหญ่จะมีประสิทธิภาพค่อนข้างต่ำ เนื่องจากพลังงานส่วนใหญ่ของเชื้อเพลิงที่ใช้ในการผลิตไฟฟ้าจะเปลี่ยนรูปไปเป็นความร้อนปล่อยทิ้งสู่บรรยากาศและเป็นมลพิษต่อสิ่งแวดล้อม

เนื่องจากปัจจุบันมนุษย์ให้ความสนใจกับปัญหาสิ่งแวดล้อมและบรรเทาปัญหาโลกร้อน (Global Warming) กันมากขึ้น ทำให้เกิดความสนใจเกี่ยวกับการผลิตพลังงานไฟฟ้าแบบกระจายจากแหล่งพลังงานธรรมชาติ หรือพลังงานหมุนเวียน (Renewable Energy Resources) เช่น พลังงานลม พลังงานแสงอาทิตย์ หรือ พลังงานชีวมวล เป็นต้น ซึ่งการผลิตไฟฟ้าแบบกระจายนั้นจะเป็นระบบขนาดเล็ก โดยผู้ใช้มีอิสระที่จะเลือกตำแหน่งติดตั้งที่อยู่ใกล้กับภาระไฟฟ้า ทำให้ระยะห่างระหว่างแหล่งกำเนิดพลังงานไฟฟ้ากับภาระไฟฟ้าที่สั้นมากจึงทำให้ลดสูญเสียพลังงานตามสายส่งเป็นการเพิ่มประสิทธิภาพรวมของระบบ และลดมลพิษที่ถูกปล่อยสู่บรรยากาศอีกด้วย

อุปกรณ์ผลิตพลังงานไฟฟ้าจากแหล่งพลังงานธรรมชาติเหล่านี้ โดยทั่วไปแล้ว กำลังไฟฟ้าที่ผลิตได้จะมีปริมาณที่ผันผวนเป็นอย่างมาก จึงทำให้เกิดความกังวลเกี่ยวกับผลกระทบในแง่ลบทั้งในด้านของคุณภาพไฟฟ้าและความเชื่อถือได้ของระบบในกรณีที่มีการนำแหล่งกำเนิดไฟฟ้าเหล่านี้ จำนวนมากเข้ามาต่อกับระบบไฟฟ้า (Grid Connected Distributed Generation) จึงได้มีการนำเสนอแนวทางการผลิตและส่งจ่ายไฟฟ้าภายในพื้นที่สำหรับระบบไฟฟ้ากำลังขนาดเล็ก โดยที่ผลิตไฟฟ้าได้ในระดับแรงดันต่ำ จึงทำให้สามารถเชื่อมต่อบริเวณเข้ากับภาระไฟฟ้า และเชื่อมต่อกับกริดไฟฟ้าหลักได้ เรียกระบบการผลิตไฟฟ้าขนาดเล็กเหล่านี้ว่า “ไมโครกริด” (microgrid) โดยที่จะจ่ายไฟฟ้าให้กับกลุ่มผู้ใช้ขนาดเล็กที่อยู่ไม่ไกลกัน เช่น หมู่บ้าน สถานศึกษา ศูนย์การค้า โรงงานหรือย่านอุตสาหกรรม เป็นต้น โดยขนาดของไมโครกริดจะมีขนาดใกล้เคียงกับภาระทางไฟฟ้าของไมโครกริดนั้น ดังนั้นสิ่งสำคัญของการผลิตไฟฟ้าด้วยไมโครกริด คือการรักษาสมดุลระหว่างความต้องการการใช้ไฟฟ้ากับปริมาณการผลิตกำลังไฟฟ้าเพื่อรักษาคุณภาพ และความน่าเชื่อถือของระบบ

เนื่องจากไมโครกริดไม่ได้มุ่งหมายเพื่อการขายไฟฟ้าเข้ากับระบบไฟฟ้ากำลังหลัก หากแต่มุ่งเน้นการผลิตและจำหน่ายไฟฟ้าในพื้นที่ของตนเอง การเพิ่มขึ้นของไมโครกริด จะไม่ทำให้เกิดปัญหาเกี่ยวกับระบบไฟฟ้ากำลังหลักใน ขณะเดียวกันก็ช่วยเสริมความมั่นคงของการส่งจ่ายไฟฟ้ายรวมทั้งความเชื่อถือได้ของระบบไฟฟ้าและคุณภาพไฟฟ้าในพื้นที่ของไมโครกริดความต้องการเสริมความมั่นคงทางพลังงานของประเทศก็จะได้รับการตอบสนอง ขณะที่ผลกระทบต่อสิ่งแวดล้อมจาก

การผลิตไฟฟ้าก็น้อยลงจากการใช้พลังงานหมุนเวียนและประสิทธิภาพการใช้พลังงานโดยรวมที่เพิ่มขึ้น

อย่างไรก็ตามไมโครกริดก็ยังมีคุณลักษณะเหมือนกริดหลัก จึงต้องมีการควบคุมคุณภาพของไฟฟ้าให้มีความเสถียรภาพตามมาตรฐานของผู้ใช้ โดยในโครงการนี้ได้นำเทคนิค Angle Droop และ Voltage Droop มาประยุกต์ใช้โดยศึกษาจากการกำหนดแบบจำลองพื้นฐานระบบไมโครกริด โดยที่เทคนิค Angle Droop และ Voltage Droop ทำให้เรามีความสามารถที่จะเชื่อมต่อ Microsource กับระบบได้ทันที (Plug and Play) โดยอาศัยข้อมูล ณ ตำแหน่งนั้น นั่นก็คือแรงดันและความถี่ วิธีการนี้ทำให้เราสามารถต่อขนานระหว่าง Microsource ได้อย่างมีประสิทธิภาพ ทั้งในโหมด Grid Connected และ Standalone เมื่อแหล่งจ่ายภายนอก(Grid)เกิดปัญหา

1.2 วัตถุประสงค์ของโครงการ

1.2.1 เพื่อศึกษาโครงสร้างและหลักการระบบไมโครกริดสาธิต ขนาด 3 kW

1.2.2 เพื่อออกแบบและติดตั้งไมโครกริดสาธิต ขนาด 3 kW

1.2.3 เพื่อศึกษาพัฒนาระบบควบคุมไมโครกริดสาธิต ขนาด 3 kW สำหรับเทคนิค Angle Droop และ Voltage Droop

1.3 ขอบเขตของการศึกษา

โครงการนี้แบ่งเป็นสองส่วนคือ ส่วนแรก เป็นการรวมผลการศึกษาเกี่ยวกับหลักการการติดตั้ง การบำรุงรักษา การจัดเก็บข้อมูลรวมถึงการแสดงผลผ่านทางโปรแกรม Visual Studio 2010 ของระบบไมโครกริดสาธิตขนาด 3 kW ส่วนที่สองเป็นการศึกษาการพัฒนาระบบควบคุมไมโครกริดด้วยเทคนิค Angle Droop และ Voltage Droop จากการสร้างแบบจำลองพื้นฐานโดยใช้โปรแกรม MATLAB/Simulink เพื่อเป็นการรักษาเสถียรภาพและความน่าเชื่อถือของระบบ

1.4 แผนการดำเนินงาน

การดำเนินงาน	ช่วงเวลา									
	มี.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	
ศึกษาโครงสร้าง สเปคของ PV panel และ Inverter ในระบบไมโครกริด รวมทั้งออกแบบการต่อแผง PV เข้ากับ Inverter โดยใช้โปรแกรม PVsyst										
ศึกษารูปแบบ ส่วนประกอบของไมโครกริด และการควบคุมกำลังไฟฟ้าของไมโครกริด ด้วยเทคนิค Angle Droop และ Voltage Droop										
ทดลองใช้ Inverter ในโหมด standalone โดยใช้ AC Supply เป็นแหล่งจ่ายแทน Solar Cell										

บทที่ 2 ทฤษฎีที่เกี่ยวข้อง

2.1 ระบบไมโครกริด

2.1.1 ความหมายและความเป็นมาของไมโครกริด

California Energy Commission ได้ให้นิยามของ “ไมโครกริด” ไว้ว่า ไมโครกริดคือการรวมระบบพลังงานซึ่งประกอบด้วย การเชื่อมต่อของโหนดภายในระบบและทรัพยากรพลังงานแบบกระจาย (Distributed energy resources) ซึ่งเป็นระบบที่สามารถปฏิบัติการคู่ขนานไปกับกริดหลัก (Main Grid) ซึ่งเรียกว่า โหนดเชื่อมต่อกริด (Grid connect) หรือในสภาวะที่แยกตัวอิสระจากกริดหลักซึ่งเรียกว่า โหนดแยกอิสระ (Stand-alone)

ระบบไฟฟ้ากำลังขนาดเล็กโดยใช้แหล่งกำเนิดไฟฟ้าขนาดเล็ก (Microsource) เช่น ไมโครเทอร์ไบน์ (Microturbine) หรือ เซลล์เชื้อเพลิง (Fuel Cells) รวมทั้งการผลิตไฟฟ้าจากแหล่งพลังงานหมุนเวียน หรือรวมเรียกว่า แหล่งกำเนิดไฟฟ้า ณ จุดใช้งาน (Distributed Energy Resource, DER) เพื่อปรับปรุงความน่าเชื่อถือได้ของระบบไฟฟ้าและตอบสนองการเปลี่ยนแปลงที่ตามมาจากการเปิดเสรีด้านพลังงานไฟฟ้า เรียกว่าแนวคิดไมโครกริด (Microgrid Concept)

แนวคิดไมโครกริด หรือ Microgrid Concept โดยถูกนำเสนอ ภายใต้โครงการ CERTS (Consortium for Electric Reliability Technology Solutions) ในสหรัฐอเมริกา ในปี 2002 ภายใต้แนวคิดนี้ทั้งแหล่งกำเนิดไฟฟ้าขนาดเล็กและโหนดภายในพื้นที่จะถูกมองรวมเป็นระบบอิสระขนาดเล็กระบบหนึ่ง ซึ่งให้ทั้งกำลังไฟฟ้าและความร้อน แหล่งกำเนิดไฟฟ้าขนาดเล็กส่วนใหญ่ในระบบนี้จะต้องประกอบขึ้นจากอุปกรณ์อิเล็กทรอนิกส์กำลังเพื่อความยืดหยุ่นในการทำงานและการควบคุม ในสภาวะที่เกิดความผิดปกติขึ้นในระบบไฟฟ้ากำลังหลัก (Grid) ไมโครกริดจะสามารถปลดตัวเองออกจากระบบไฟฟ้ากำลังหลัก และทำงานในแบบไอส์แลนด์ (ผลิตและส่งกำลังไฟฟ้าภายในพื้นที่) ได้อย่างอัตโนมัติ และสามารถเชื่อมต่อกลับเข้าไปกับระบบไฟฟ้ากำลังหลักได้เมื่อความผิดปกติในระบบไฟฟ้ากำลังหลักได้รับการแก้ไขไปแล้ว

โดยแนวคิดนี้ ไมโครกริดจะถูกมองเป็นระบบควบคุมหน่วยหนึ่งจากระบบไฟฟ้ากำลังหลัก และจะไม่สร้างปัญหาเช่นการกระเพื่อมของแรงดันไฟฟ้า การแกว่งไกวของความถี่ไฟฟ้า เป็นต้นให้กับระบบไฟฟ้ากำลังหลัก ไมโครกริดที่มีคุณสมบัตินี้ถูกนิยามให้เป็นพลเมืองดี (Good Citizen) ของระบบกำลังไฟฟ้าหลัก ในขณะที่เดียวกันไมโครกริดยังสามารถช่วยสนับสนุนการทำงานของระบบไฟฟ้ากำลังหลักได้ในลักษณะของบริการเสริม เช่น การให้บริการคุณภาพไฟฟ้าชั้นเยี่ยม (Premium Power) การรักษาระดับแรงดันไฟฟ้า (Voltage Support) เป็นต้น ไมโครกริดที่มีคุณสมบัตินี้ถูกนิยามให้เป็นพลเมืองตัวอย่าง (Model Citizen) ของระบบไฟฟ้ากำลังหลัก แนวความคิดไมโครกริดนั้นนอกจากในสหรัฐอเมริกาแล้ว ยังได้รับการนำเสนอทั้งในยุโรปและญี่ปุ่นด้วยเช่นกัน โดยในยุโรป โดยการนำของสหภาพยุโรปหรืออียู ได้เริ่มต้นโครงการเกี่ยวกับไมโครกริดขึ้นในวันที่ 1 มกราคม ค.ศ. 2003 ส่วนในญี่ปุ่นมีโครงการวิจัยทดลองติดตั้งทดสอบในช่วงปี ค.ศ. 2003-2007 จำนวน 3 สถานีทดลองด้วยกัน

2.1.2 ระบบผลิตไฟฟ้าแบบกระจายศูนย์ Distributed Generation (DG)

ระบบไฟฟ้าทั่วโลกเริ่มตระหนักถึงปัญหาการลดลงของปริมาณของทรัพยากรเชื้อเพลิงฟอสซิลและการทำให้พลังงานและสภาพแวดล้อมต่างๆ ค่อยๆ ทรุดโทรมลงไป ปัญหาเหล่านี้ได้นำไปสู่แนวความคิดใหม่ในการผลิตพลังงานภายในพื้นที่ ณ ระดับการกระจายแรงดันต่างๆ โดยการใช้พลังงานทดแทนและพลังงานหมุนเวียน เช่น ก๊าซธรรมชาติ ก๊าซชีวภาพ พลังงานลม เซลล์แสงอาทิตย์ เซลล์เชื้อเพลิงความร้อนร่วม (Combined heat and power : CHP) ระบบไมโครเทอร์ไบน์ และเครื่องยนต์สเตอร์ลิง (Stirling engine) ซึ่งแหล่งพลังงานเหล่านี้จะเป็นประโยชน์ต่อเครือข่ายการจำหน่าย การผลิตไฟฟ้าในรูปแบบนี้จะเรียกว่า ระบบผลิตไฟฟ้าแบบกระจายศูนย์หรือ Distributed Generation (DG) และแหล่งพลังงานเหล่านี้จะเรียกว่า แหล่งพลังงานแบบกระจายศูนย์ หรือ Distributed energy resources (DER) โดยการผลิตแบบกระจายศูนย์นี้ได้รับการออกแบบให้มีความแตกต่างจากการผลิตไฟฟ้าจากส่วนกลาง (Centralized conventional generation) ในสมัยก่อน และการใช้ระบบจำหน่ายร่วมกับการรวมตัวของ DG จะทำให้ระบบดังกล่าวเป็นระบบที่มีความรวดเร็วในการจำหน่ายไฟฟ้าไปยังส่วนต่างๆ

ในปลายปี 1990 ได้มีการให้ความสำคัญกับการศึกษาประเด็นหลักที่เกี่ยวข้องกับ DG โดยคณะกรรมการของ International Council on Large Electric Systems (CIGRE) และการประชุมนานาชาติและ Exhibition on Electricity Distribution (CIRED) และนำเสนออยู่ในรายงานการประชุม

ข้อกำหนดเกี่ยวกับ DG ที่พบในหลายๆ ประเทศนั้นส่วนใหญ่นั้นจะขึ้นอยู่กับขนาดพิกัดของโรงจักรไฟฟ้า ระดับแรงดันไฟฟ้าในการผลิต ฯลฯ จากการศึกษาและวิจัยได้ข้อสรุปลักษณะทั่วไปของ DG ได้ดังต่อไปนี้

1. ไม่เป็นศูนย์กลางของระบบไฟฟ้าหรือศูนย์กลางการจำหน่ายไฟฟ้า
2. มีขนาดเล็กกว่า 50 เมกะวัตต์
3. แหล่งกำเนิดกำลังไฟฟ้าหรือเครื่องกำเนิดไฟฟ้าแบบกระจายศูนย์มักจะเชื่อมต่อกับระบบจำหน่ายที่มีขนาดของแรงดัน 230/415 V ถึง 145 kV

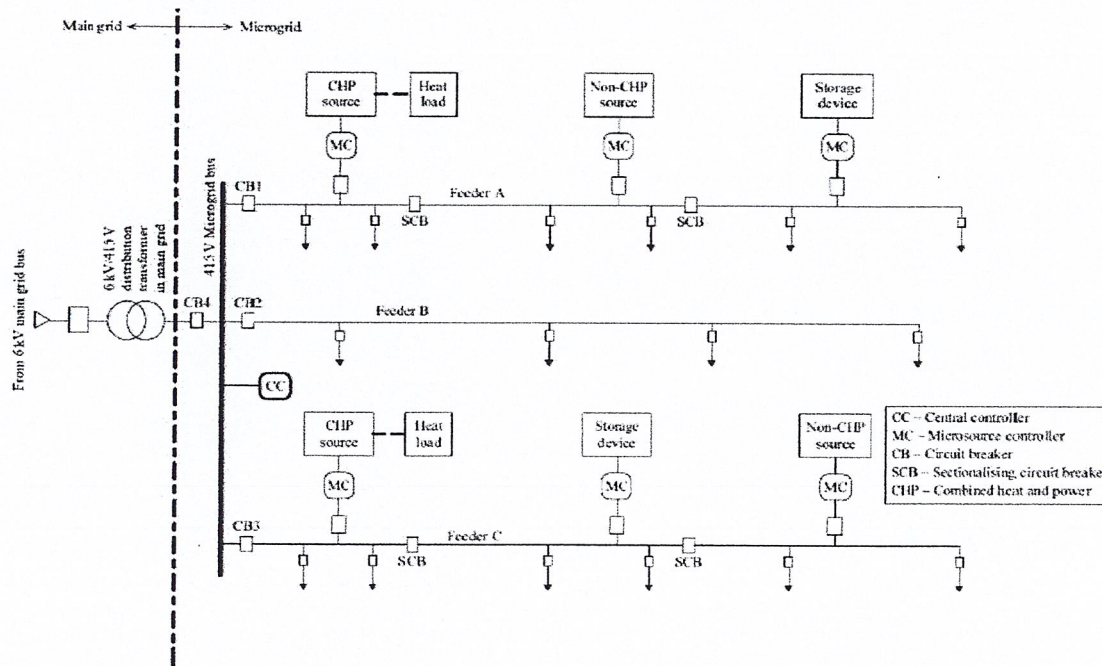
เนื่องจากไมโครกริดไม่ได้มุ่งหมายเพื่อการขายไฟฟ้าเข้ากับระบบไฟฟ้ากำลังหลัก หากแต่มุ่งเน้นการผลิตและจำหน่ายไฟฟ้าในพื้นที่ของตนเอง การเพิ่มขึ้นของไมโครกริด รวมถึง DER จะไม่ทำให้เกิดปัญหากับระบบไฟฟ้ากำลังหลัก ในขณะเดียวกันก็ช่วยเสริมความมั่นคงของการส่งจ่ายไฟฟ้า รวมทั้งความเชื่อถือได้ของระบบไฟฟ้าและคุณภาพไฟฟ้าในพื้นที่ของไมโครกริดความต้องการเสริมความมั่นคงทางพลังงานของประเทศก็จะได้รับการตอบสนอง ขณะที่ผลกระทบต่อสิ่งแวดล้อมจากการผลิตไฟฟ้าน้อยลงจากการใช้พลังงานหมุนเวียน และประสิทธิภาพการใช้พลังงานโดยรวมที่เพิ่มขึ้น นอกจากนี้เนื่องจาก DER มีขนาดเล็กการวางแผนและติดตั้งจึงใช้เวลาและค่าใช้จ่ายน้อยกว่าระบบการผลิตขนาดใหญ่ จึงมีส่วนช่วยกระจายความเสี่ยงของการลงทุนด้านการผลิตไฟฟ้าได้อีกทางหนึ่งจากที่กล่าวมา อาจสามารถสรุปความสำคัญของแนวคิดไมโครกริดได้ดังนี้

1. การรักษาคุณภาพกำลังไฟฟ้าและความน่าเชื่อถือได้ของระบบไฟฟ้า
2. การลดระยะเวลาและค่าใช้จ่ายในการติดตั้ง
3. การลดค่าใช้จ่ายทางพลังงานหรือค่าเชื้อเพลิงลง
4. การลดผลกระทบต่อสิ่งแวดล้อม

5. การรักษาความมั่นคงทางพลังงาน

2.1.3 ส่วนประกอบต่างๆของไมโครกริด และหน้าที่ของแต่ละส่วนประกอบ

ส่วนประกอบต่างๆของไมโครกริดเป็นไปตามรูปที่ 2.1 จะประกอบด้วยภาระทางไฟฟ้าหรือความร้อนรวมทั้งแหล่งจ่ายขนาดเล็ก (microsource) โดยมีการเชื่อมต่อในโครงข่ายระบบจำหน่ายแรงดันต่ำจะเห็นว่าภาระจะถูกลำเลียงไว้ใกล้กับแหล่งจ่ายเพื่อลดการสูญเสียโดยเฉพาะภาระทางความร้อน (CHP source)



รูปที่ 2.1 ส่วนประกอบต่างๆของไมโครกริด

ไมโครกริดนั้นจะเชื่อมต่อกับกริดหลัก (main grid) แรงดันระดับกลางผ่านเซอร์กิตเบรกเกอร์ซึ่งเป็นจุดต่อร่วม (Point of Common Coupling : PCC) หรือ CB4 ซึ่งทำหน้าที่เชื่อมต่อและตัดตอนไมโครกริด ทั้งหมดจากกริดหลักตามโหมดที่เลือกการดำเนินงานนั้นคือ ถ้า CB4 ปิดแสดงว่าไมโครกริดทำงานอยู่ในโหมดเชื่อมต่อกับกริด (grid connect) แต่ถ้า CB4 เปิดแสดงว่าไมโครกริดทำงานอยู่ในโหมดแยกอิสระ (stand-alone)

ไมโครกริดดำเนินการในสองโหมด : (1) เชื่อมต่อกับกริด และ (2) ในโหมดแยกอิสระ โดยในโหมดเชื่อมต่อกับกริดนั้น ไมโครกริดจะยังคงเชื่อมต่อกับกริดหลักทั้งหมดหรือบางส่วนและนำเข้าหรือส่งกำลังไปยังกริดหลัก ในกรณีที่เกิดสัญญาณรบกวนในกริดหลัก ไมโครกริดจะสลับไปยังโหมดแยกอิสระในขณะที่ยังคงจ่ายกำลังให้กับภาระที่สำคัญเหมือนเดิม โหมดแยกอิสระนี้สามารถทำได้โดย

- (i) ตัดการเชื่อมต่อ Microgrid ทั้งหมดโดยเปิด CB4
- (ii) ตัดการเชื่อมต่อ สายป้อน A และ C โดยเปิด CB1 และ CB3

สำหรับตัวเลือก (i) ไมโครกริดจะทำงานเป็นระบบอัตโนมัติที่มีทั้งหมดแหล่งจ่ายขนาดเล็กจะจ่ายกำลังให้กับภาระทั้งหมดในสายป้อน A, B และ C

ในขณะที่ตัวเลือก (ii) สายป้อน A และ C จะจ่ายเพียงโหลดสำคัญในขณะที่สายป้อน B จะถูกตัดทิ้งไปเนื่องจากเป็นโหลดไม่สำคัญ

การดำเนินการและการจัดการของไมโครกริดในโหมดต่างๆจะถูกควบคุมและมีการทำงานร่วมกัน (co-ordinate) ผ่านตัวควบคุมแหล่งจ่าย (MC) และตัวควบคุมส่วนกลาง (CC) ที่มีฟังก์ชันการทำงานดังต่อไปนี้

Microsource Controller (MC)

หน้าที่หลักของ Microsource Controller (MC) คือการควบคุมที่เป็นอิสระของการไหลของกำลังไฟฟ้าและแรงดันที่บัสปลายของแหล่งจ่ายขนาดเล็กในการตอบสนองต่อการรบกวนรวมถึงการเปลี่ยนแปลงของภาระ ซึ่งคาดว่าเป็นอิสระในที่นี้หมายถึงการควบคุมโดยไม่มีคำสั่งจากตัวควบคุมส่วนกลาง(CC) นั่นเองและตัว MC นี้จะต้องทำให้มั่นใจได้ว่าแหล่งจ่ายขนาดเล็กแต่ละตัวจะต้องสามารถเพิ่มระดับการผลิตได้อย่างทันท่วงทีเพื่อจ่ายกำลังไฟฟารวมถึงมีการจัดสรรการผลิตให้กับภาระต่างๆในโหมดแยกอิสระและสามารถกลับมาเชื่อมต่อในโหมดของการเชื่อมต่อกริดได้อย่างอัตโนมัติโดยไม่มีคำสั่งจาก CC ซึ่งคุณลักษณะที่สำคัญที่สุดของ MC คือการตอบสนองที่รวดเร็วต่อการตรวจสอบระดับแรงดันและกระแสภายในพื้นที่โดยไม่มี การนำเอาข้อมูลจาก MC ข้างเคียงมาพิจารณา สองคุณลักษณะพิเศษหลักของ MC คือ MC หนึ่งตัวจะทำงานโดยแยกอิสระไม่ขึ้นกับ MC ตัวอื่นๆในไมโครกริดและมันจะสามารถทำการยกเลิกคำสั่งที่มาจาก CC ที่วิเคราะห์แล้วว่า จะก่อให้เกิดอันตรายต่อแหล่งจ่ายขนาดเล็กภายในระบบ

Central controller (CC)

จะจัดการควบคุมการดำเนินงานของไมโครกริดและการป้องกันผ่าน MC โดยมีวัตถุประสงค์เพื่อ

- เพื่อรักษาระดับแรงดันและความถี่ของภาระปลายทาง ผ่านการควบคุมกำลังไฟฟ้าและความถี่รวมถึงการควบคุมแรงดันไฟฟ้า

- เพื่อเพิ่มประสิทธิภาพของพลังงานในไมโครกริดโดย CC ได้ถูกออกแบบมาให้ทำงานในโหมดอัตโนมัติเมื่อมีการแทรกแซงการทำงานของระบบและในยามจำเป็น

สองโมดูลการทำงานหลักของ CC คือโมดูลการจัดการพลังงาน(Energy Management Module) และโมดูลป้องกันในการทำงานร่วมกัน (Protection Co-ordination Module)

Energy Management Module หรือ EMM จะทำการส่งค่าของจุดอ้างอิงของกำลังไฟฟ้าจริง (active power) และกำลังไฟฟ้าเสมือน (reactive power) แรงดันไฟฟ้าและความถี่ไปยังแต่ละ MC เพื่อใช้เป็นจุดอ้างอิงในการดำเนินงาน

Protection Co-ordination Module หรือ PCM จะทำการตอบสนองต่อความผิดปกติในไมโครกริดรวมถึงกริดหลัก และสภาวะการหายไปของกริด (Loss of Grid : LOG) ด้วยคุณลักษณะดังกล่าวจึงทำให้มั่นใจในการทำงานร่วมกันในการป้องกันไมโครกริดได้อย่างถูกต้อง และยังรวมถึงการปรับตัวต่อการเปลี่ยนแปลงต่อระดับกระแสผิดปกติในระหว่างเกิดการเปลี่ยนสภาวะการทำงานจากโหมดเชื่อมต่อกริดไปเป็นโหมดแยกอิสระ

2.1.4 แหล่งพลังงานในไมโครกริด (Resources in microgrid)

พลังงานทดแทนหรือแหล่งกำเนิดไฟฟ้าแบบใหม่ๆ ได้ถูกนำมาใช้ในไมโครกริด ซึ่งถูกเรียกว่า แหล่งกำเนิดไฟฟ้า ณ จุดใช้งาน (distributed energy resource : DER) หรือ แหล่งจ่ายขนาดเล็ก (microsources) มีวัตถุประสงค์เพื่อรวมนำข้อดีของแหล่งกำเนิดพลังงานที่ทำให้คาร์บอนไดออกไซด์ต่ำและแหล่งกำเนิดพลังงานไฟฟ้าร่วมกับความร้อน (Combined Heat and Power : CHP) ที่มีประสิทธิภาพสูง DER ที่ใช้ในไมโครกริดมีดังนี้

- Combined Heat and Power
- Wind Energy Conversion
- Solar Photovoltaic
- Small-scale hydroelectric
- Storage Device

Combined Heat and Power (CHP)

CHP นั้นเป็นแหล่งผลิตที่เป็นการผลิตพลังงานร่วม (co-generation) ที่ผลิตทั้งพลังงานไฟฟ้าและพลังงานความร้อน ซึ่งมีแนวโน้มอย่างมากที่จะนำมาใช้กับไมโครกริด ข้อดีก็คือมีประสิทธิภาพทางพลังงานโดยการใช้ความร้อนทิ้งที่ได้จากคริวรีนหรือกระบวนการทางอุตสาหกรรม นอกจากนี้ยังสามารถใช้ในการซิลเลอร์สำหรับระบายความร้อนการดูดซึม พร้อมกันของการผลิตไฟฟ้า ความร้อนและระบายความร้อนเป็นที่รู้จักกันเป็น trigeneration หรือ polygeneration ระบบ CHP ช่วยให้การใช้พลังงานที่ดีขึ้นกว่ารุ่นเดิมที่อาจถึงประสิทธิภาพมากกว่า 80% เมื่อเทียบกับที่ประมาณ 35% สำหรับโรงไฟฟ้าแบบเดิม สำหรับการผลิตความร้อนร่วมขนาดเล็ก (Micro - CHP) จะถูกติดตั้งในสถานที่ที่มีขนาดเล็กเช่น บ้านหรืออาคารขนาดเล็ก ส่วนใหญ่หน่วย CHP ในอุตสาหกรรม การผลิตไฟฟ้าเป็นผลิตภัณฑ์หลักส่วนความร้อนเป็นผลพลอยได้ ในขณะที่ระบบ Micro CHP การสร้างความร้อนเป็นผลิตภัณฑ์หลักส่วนไฟฟ้าเป็นผลพลอยได้ Micro-CHP มีความเชื่อถือได้ แข็งแกร่งและราคาถูก มีกำลังการผลิตในช่วง 10-100 กิโลวัตต์

Micro CHP สามารถพัฒนาจากเทคโนโลยีหลายเทคโนโลยี เช่น เครื่องยนต์สเตอร์ลิง (Stirling engines) Reciprocating engines (เครื่องยนต์สันดาปภายใน, Internal Combustion engine) และเซลล์เชื้อเพลิง (Fuel Cell) โดยที่เทคโนโลยีที่กล่าวถึงจะทำหน้าที่ในการผลิตกระแสไฟฟ้าและนำความร้อนเหลือใช้ในการผลิตกระแสไฟฟ้าไปใช้ในการทำความร้อน และถ้าจะผลิตความเย็นก็จะใช้ความร้อนที่ได้นี้ไปเป็นพลังงานสำหรับเครื่องทำความเย็นแบบดูดซึม (absorbtion) เพื่อผลิตความเย็นต่อไป

Wind Energy Conversion

เป็นการเปลี่ยนรูปพลังงานลมเป็นพลังงานไฟฟ้า โดยมีหลักการพื้นฐานโดยใช้กังหันลมต่อเข้ากับเครื่องกำเนิดไฟฟ้าผ่านระบบเกียร์ (gearbox) ซึ่งเครื่องกำเนิดไฟฟ้าที่ใช้จะเป็นเครื่องกำเนิดไฟฟ้าเหนี่ยวนำ กังหันลมจะจับพลังงานจลน์จากการไหลของลมผ่านใบพัดโรเตอร์และส่งพลังงานไปยังเครื่องกำเนิดไฟฟ้าเหนี่ยวนำผ่านระบบเกียร์เพลลาของเครื่องกำเนิดไฟฟ้าจะถูกขับเคลื่อนด้วยกังหันเปลี่ยนเป็นพลังงานไฟฟ้า สำหรับระบบเกียร์นั้นมิใช่เพื่อช่วยปรับระดับความเร็วของโรเตอร์ก่อนที่จะเข้าไปยังเครื่องกำเนิดไฟฟ้ากังหันลมที่ซี่มีทั้งแบบลักษณะการหมุนแนวตั้ง (Horizontal axis) และหมุนในแนวนอน (Vertical axis) ขนาดกำลังการผลิตนั้นปัจจุบันสามารถผลิตได้สูงถึง 5 เม

กะวัตต์ กำลังการผลิตนั้นมีปัจจัยขึ้นอยู่กับสภาพอากาศ โดยจะแปรผันตรงกับความเร็วลม ขนาดใบพัด และความหนาแน่นของอากาศ

Solar Photovoltaic

พลังงานแสงอาทิตย์ เกิดจากปฏิกิริยาฟิวชั่นของดวงอาทิตย์ จะปล่อยพลังงานออกมาในรูป คลื่นแม่เหล็กไฟฟ้า ที่เรียกว่า รังสีแสงอาทิตย์ (Solar Radiation) รังสีนี้จะแพร่กระจายออกทุกทิศทุกทาง โลกของเราก็ได้รับอิทธิพลของรังสีนี้ โดยมีความเข้มของรังสีที่ตกลงบนผิวโลกประมาณ 961 ถึง 1,191 วัตต์ต่อตารางเมตร หรือคิดเป็นพลังงานประมาณ 2,000-2,500 กิโลวัตต์ ชั่วโมงต่อตารางเมตรต่อปี

สำหรับประเทศไทย พื้นที่เกือบทั้งหมดสามารถรับพลังงานจากแสงอาทิตย์เฉลี่ยประมาณ 4.5 กิโลวัตต์-ชั่วโมง/ตารางเมตร/วัน ดังนั้นในพื้นที่ 1 ตารางกิโลเมตร สามารถติดตั้งระบบผลิตไฟฟ้าจากเซลล์แสงอาทิตย์ขนาด 33 เมกะวัตต์ หรือ 165,000 กิโลวัตต์-ชั่วโมง/ตารางเมตร/วัน ในปัจจุบันความต้องการพลังงานไฟฟ้าของประเทศประมาณวันละ 250 ล้านกิโลวัตต์-ชั่วโมง ถ้าต้องการผลิตจากพลังงานแสงอาทิตย์ทั้งหมด จำเป็นต้องใช้พื้นที่ประมาณ 1,500 ตารางกิโลเมตร หรือคิดเป็นพื้นที่ประมาณ 0.3% ของประเทศเท่านั้น ในอดีตการผลิตไฟฟ้าจากเซลล์แสงอาทิตย์มีราคาแพงมาก แต่เนื่องจากปัจจุบันราคาของเซลล์แสงอาทิตย์ได้ลดลงอย่างมาก และมีแนวโน้มว่าจะลดลงอีกเรื่อย ๆ เพราะประชาชนโดยทั่วไปได้ตระหนักถึงสถานะแวดล้อมเป็นพิษ เนื่องจากการใช้ เชื้อเพลิงฟอสซิลในการผลิตพลังงาน จึงหันมาใช้เซลล์แสงอาทิตย์เพิ่มขึ้นเรื่อย ๆ

Small-scale hydroelectric

เป็นการกำเนิดไฟฟ้าโดยใช้กังหันน้ำขนาดเล็กในการขับเคลื่อนเครื่องกำเนิดไฟฟ้า โดยมีหลักการโดยใช้ประโยชน์จากการไหลของน้ำจากแหล่งน้ำขนาดเล็กในการขับเคลื่อนกังหันกำลังการผลิตนั้นแปรผันโดยตรงกับอัตราการไหลของน้ำ ความหนาแน่นของน้ำ และระดับความสูงของหัวน้ำ

Storage Device

เป็นอุปกรณ์ในการเก็บสะสมพลังงานสำหรับจ่ายพลังงานฉุกเฉินในไมโครกริด ซึ่งสามารถจ่ายพลังงานในช่วงเวลาสั้นๆ อุปกรณ์ในการเก็บพลังงานมีดังนี้

- Storage Battery
- Flywheels
- Ultra Capacitor

2.1.5 ลักษณะต่างๆของไมโครกริด

การส่งเสริมไมโครกริดให้ดีขึ้นนั้นเป็นความหวังของแหล่งพลังงานไฟฟ้าอุตสาหกรรม เพราะว่าเป็นประโยชน์ต่างดังนี้

ในด้านสิ่งแวดล้อม

ไม่จำเป็นต้องพูดเลยว่าไมโครกริดนั้น สร้างผลกระทบต่อให้กับสิ่งแวดล้อมน้อยกว่าสถานีไฟฟ้าพลังงานความร้อนมากๆ แต่อย่างไรก็ตามการอ้างถึงความสำเร็จในการส่งเสริมของการควบคุมคาร์บอน และเก็บโครงการของโรงไฟฟ้าพลังงานความร้อนไว้เพราะจะช่วยลดผลกระทบต่อสิ่งแวดล้อมได้อย่างมาก แต่กระนั้นผลประโยชน์บางอย่างของไมโครกริดก็ควรจะมีดังนี้

- การลดการใช้เชื้อเพลิงแก๊สให้น้อยที่สุดและลดการแพร่ของ CO₂ ให้น้อยที่สุด ลดการกำหนดใช้ในวงจำกัด โดยควรจะเป็นกระบวนการเผาไหม้ในขั้นสุดท้ายเพื่อช่วยระบบทั้งหมด เมื่อแหล่งจ่ายงานอื่นทำงานได้ไม่เพียงพอ

- การใช้ไมโครกริดและแหล่งผลิตพลังงานย่อยควรช่วยกันเพิ่มความตระหนัก ของการใช้พลังงานไปในทางที่เหมาะสม

ในด้านการใช้งานและการเงิน

ไมโครกริดนั้นทำให้มีผลกระทบระหว่างแหล่งผลิตพลังงานย่อยกับโหลดได้ดังนี้

- ช่วยส่งเสริมสนับสนุนระบบทั้งหมดด้วยการเพิ่มระดับของแรงดันไฟฟ้าระบบ
- การลดความแออัดของระยะสายป้อนในการส่งและจำหน่ายกำลังไฟฟ้า
- การลดค่าความสูญเสียทางกำลังไฟฟ้าในการส่งและการจำหน่ายกำลังไฟฟ้า
- ช่วยลดการลงทุนสร้างการขยายของระบบส่งจ่ายกำลังไฟฟ้าและระบบผลิตพลังงาน

ไฟฟ้า โดยจัดการใช้ทรัพย์สินในวงจำกัด

คุณภาพของกำลังไฟฟ้า

ส่งเสริมให้ดีขึ้นในคุณภาพของกำลังไฟฟ้าและความมั่นคงของระบบได้ ได้ดังนี้

- การจ่ายพลังงานไฟฟ้าจากศูนย์กลางไปยังพื้นที่ต่างๆ
- มีความสอดคล้องกันอย่างดีระหว่างการจ่ายพลังงานไฟฟ้าเพื่อเพียงพอต่อความต้องการใช้พลังงานไฟฟ้า

ต้องการใช้พลังงานไฟฟ้า

- การลดผลกระทบของ ค่ากำลังไฟฟ้า, แรงดันไฟฟ้า, ความถี่ในการใช้งานได้ เมื่อระบบส่งจ่ายและผลิตกำลังไฟฟ้าของส่วนกลางขาดหายไป

- ลดเวลาของการไม่ทำงานของเครื่องจักรกลไฟฟ้าให้น้อยที่สุด และ เพิ่มการซ่อมแซมให้สำเร็จตลอดการกลับมาเริ่มต้นใช้งานของแหล่งผลิตพลังงานไฟฟ้าย่อย

การประหยัดค่าใช้จ่าย

การประหยัดค่าใช้จ่ายต่อไปนี้จะประสบความสำเร็จใน ไมโครกริด :

- การประหยัดอย่างมีนัยสำคัญมาจากการใช้ประโยชน์จากความร้อนเหลือทิ้งในโหมดของการทำงานของCHP นอกจากนั้นที่แหล่งจ่าย CHP ยังตั้งอยู่ใกล้กับโหลดผู้ใช้งานที่ไม่มีโครงสร้างพื้นฐานที่สำคัญเป็นสิ่งจำเป็นสำหรับการส่งผ่านความร้อนนี้ให้ประสิทธิภาพการใช้พลังงานรวมกว่า 80% เมื่อเทียบกับ จำนวนสูงสุดที่ 40% สำหรับระบบไฟฟ้าทั่วไป

- การประหยัดค่าใช้จ่ายจะมีผลโดยตลอดการรวมของหลายแหล่งผลิตพลังงานไฟฟ้าย่อยขณะที่พวกเขาจะอยู่ในประเทศในโหมด plug-and-play, ค่าใช้จ่ายที่ส่งจ่ายกำลังไฟฟ้ามีการลดลงอย่างมากหรือตัดออก เมื่อรวมกันเป็นไมโครกริด ไฟฟ้าที่ผลิตสามารถใช้ร่วมกันในกลุ่มผู้ใช้งานในประเทศ ซึ่งนอกจากนั้นการลดลงจำเป็นต้องนำเข้า / ส่งออกพลังงานไป / กลับจากกริดหลักที่มีสายป้อนขนาดยาว

ปัญหาการตลาด

ข้อดีต่อไปนี้จะบรรลุในกรณีของตลาดการมีส่วนร่วม:

- การพัฒนาตลาดที่ขับเคลื่อนด้วยขั้นตอนการดำเนินงานของไมโครกริดจะนำไปสู่การลดลงอย่างมีนัยสำคัญจากอำนาจตลาดที่จัดตั้งขึ้นโดยการกระทำของบริษัท ผลิต

- การประยุกต์ใช้อย่างกว้างขวางจาก Plug-and-play modular microsources อาจนำไปสู่การลดราคาพลังงานในตลาดพลังงาน

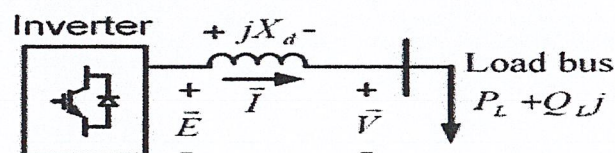
- สมดุลทางเศรษฐกิจที่เหมาะสมระหว่างเครือข่ายการลงทุนและการผลิตไฟฟ้าแบบกระจายศูนย์การใช้มีแนวโน้มที่จะลดระยะยาวราคาของผู้ใช้งานโดยการผลิตไฟฟ้าประมาณ 10%

2.2 เทคนิค Frequency Droop และ Voltage Droop

Frequency และ Voltage Droop เป็นวิธีการในการควบคุมของแหล่งกำเนิดไฟฟ้าขนาดเล็ก (Microsource ,MC) ซึ่งเราสามารถจำลองให้อยู่ในรูปของแหล่งจ่ายแรงดันได้ ซึ่งตัวของ Microsource สามารถจ่ายพลังงานได้ทั้งในโหมด Grid-connected หรือ Stand-aloneได้ เทคนิค Frequency Droop และ Voltage Droop ทำให้เรามีความสามารถที่จะเชื่อมต่อMC กับระบบได้ทันที(Plug and Play) โดยอาศัยข้อมูล ณ ตำแหน่งนั้น นั่นก็คือแรงดันและความถี่ วิธีการนี้ทำให้เราสามารถต่อขนานระหว่าง MC ได้อย่างมีประสิทธิภาพ ทั้งในโหมด Grid Connected และ Stand-alone เมื่อแหล่งจ่ายภายนอก(Grid)เกิดปัญหา

2.2.1 การควบคุมกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือน (Active and Reactive Power Control)

จากรูปที่ 2.2 แสดงโมเดลอย่างง่ายของแหล่งกำเนิดไฟฟ้าขนาดเล็ก (Microsource) ที่ประกอบด้วยเวกเตอร์และทิศทางของแกนอ้างอิงที่เกี่ยวข้อง



รูปที่ 2.2 แสดงโมเดลอย่างง่ายของ Microsource

สมมติให้กำลังไฟฟ้าจริง (P_L) และกำลังไฟฟ้าเสมือน (Q_L) ที่ถูกส่งผ่านโดย Microsource จะถูกอธิบายตามสมการต่อไปนี้

$$P_L = \frac{1}{X_d} E V \sin \delta \quad (2.1)$$

$$Q_L = \frac{1}{X_d} (E V \cos \delta - V^2) \quad (2.2)$$

จากรูปและสมการด้านบนตีความได้ว่า เมื่อ $P_L > 0$ Microsource จะอัดฉีดกำลังไฟฟ้าจริงไปยังโหลดบัสและเมื่อ $P_L < 0$ Microsource จะรับเอากำลังไฟฟ้าจริงมาจากโหลดบัส ส่วนกรณีกำลังไฟฟ้า Reactive เมื่อ $Q_L > 0$ Microsource จะอัดฉีด Inductive reactive Power ไปยังโหลดบัส และเมื่อ $Q_L < 0$ Microsource จะจ่าย Capacitive reactive Power ไปยังโหลดบัส เมื่อมุมกำลัง (Power angle, δ) นั้นมีค่าน้อย (ต่ำกว่า 0.1 rad) จะสามารถประมาณได้ว่า $\sin \delta \approx \delta$ และ $\cos \delta \approx 1$ ทำให้สมการที่ (2.1) และ (2.2) สามารถเขียนในรูปย่อได้เป็น

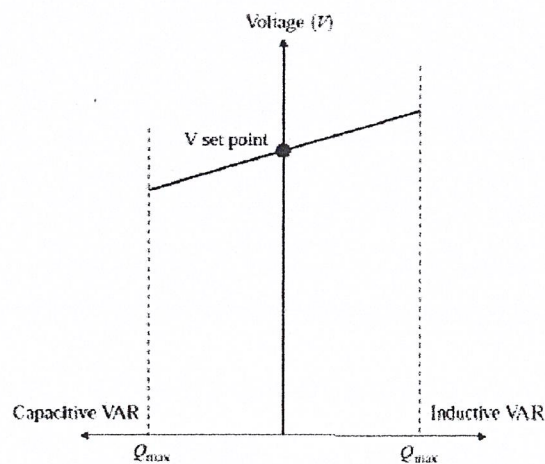
$$P_L = \frac{EV}{X_d} \sin \delta \approx \frac{EV}{X_d} \delta = \frac{EV}{X_d} (\theta_E - \theta_V) \quad (2.3)$$

$$Q_L = \frac{EV}{X_d} \cos \delta - \frac{V^2}{X_d} \approx \frac{V}{X_d} (E - V) \quad (2.4)$$

เนื่องจากตัวควบคุมสามารถวัดแรงดันที่โหนดบัส V และควบคุมแรงดันขาออกของอินเวอร์เตอร์ E ดังนั้นจากสมการ 1.3 แสดงให้เห็นว่า P_L นั้นแปรผันตรงกับมุมกำลัง δ โดยตัวควบคุมสามารถส่งจ่ายกำลังไฟฟ้าได้มากขึ้นโดยควบคุม $\angle \delta$ ให้มีความแตกต่างกันของมุมกำลัง δ มากขึ้น สมการ (2.4) แสดงให้เห็นว่ากำลังไฟฟ้า Reactive นั้นแปรผันตรงกับส่วนต่างของแรงดัน E กับ V ทำให้ตัวควบคุมสามารถควบคุมกำลังไฟฟ้า Reactive ผ่านการควบคุมขนาดของแรงดัน E สมการที่ (2.3) และ (2.4) เป็นสมการพื้นฐานของ Voltage และ Frequency droop

2.2.2 Voltage Droop Control

นอกจากการควบคุมกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือนแล้ว การควบคุมแรงดันที่บัสของไมโครกริดนั้นจำเป็นต้องพิจารณาของเสถียรภาพและความน่าเชื่อถือของไมโครกริด ไมโครกริดที่มีแหล่งจ่ายจำนวนมากอาจมีปัญหาในเรื่องการแกว่งของกำลังเสมือน (Reactive Power) หากปราศจากการควบคุมแรงดันที่เหมาะสม เช่นเดียวกันกับเครื่องกำเนิดไฟฟ้าเชิงโครนัสขนาดใหญ่ ฟังก์ชันควบคุมแรงดันของ MC จะจัดการปัญหากระแสไหลวนที่ฟูลวน (Circulating Reactive Current) ระหว่างไมโครกริดให้เบาบางลง สำหรับกริดหลักนั้นกระแสไหลวนเหล่านี้โดยทั่วไปจะถูกจำกัดโดยอิมพีแดนซ์ขนาดใหญ่ระหว่างเครื่องกำเนิดไฟฟ้าแต่ในส่วนของไมโครกริดนั้นปัญหาค่อนข้างจะชัดเจน เนื่องจากสายป้อนมักต่อเป็นรูปแบบ Radial ซึ่งมีอิมพีแดนซ์ระหว่างแหล่งจ่ายต่ำบางครั้งกระแสไหลวนเหล่านี้อาจเกินค่าพิกัดกระแสของ MC แม้ว่าจะมีการเปลี่ยนแปลงแรงดันที่เล็กน้อยในจุดอ้างอิงแรงดัน ฉะนั้นในไมโครกริดกระแสไหลวนเหล่านี้สามารถถูกควบคุมโดยวิธี (Voltage-Reactive Power (V-Q) droop controller) ดังกราฟแสดงคุณลักษณะลดลงของแรงดันซึ่งแสดงในรูปที่ 2.3



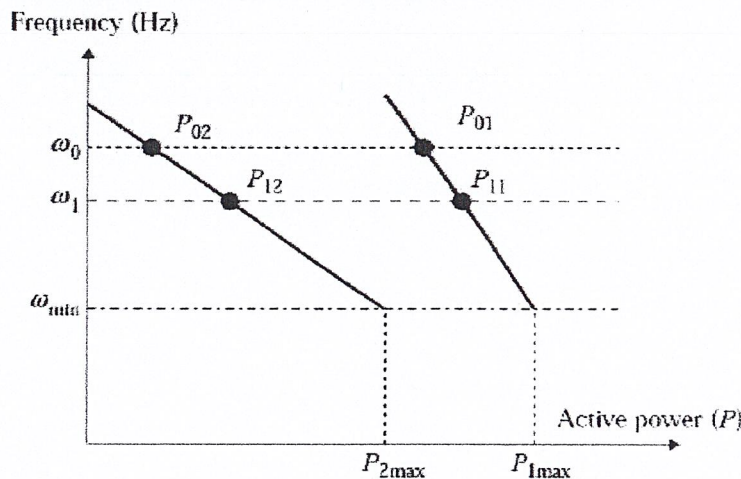
รูปที่ 2.3 กราฟคุณลักษณะของ Voltage-Reactive Power (V-Q) droop controller

จากกำลังไฟฟ้า Reactive นั้นแปรผันตรงกับส่วนต่างของแรงดัน E กับ V ดังนั้นหลักการควบคุม คือ การเพิ่มแรงดันอ้างอิง (Voltage Set Point, E) เมื่อกระแสแอกทีฟของ MC มีลักษณะค่อนข้างเหนี่ยวนำ (Inductive) และลดแรงดันอ้างอิงลง เมื่อกระแสแอกทีฟมีลักษณะค่อนข้างจุไฟฟ้า (Capacitive) ซึ่งการควบคุมกำลังไฟฟ้าเสมือนนั้นจะถูกจำกัดโดยพิกัดของกำลังปรากฏ (VAR; S) ของอินเวอร์เตอร์ และกำลังไฟฟ้าจริง (P) ที่ออกของอินเวอร์เตอร์ของ MC ดังความสัมพันธ์ต่อไปนี้

$$Q_{\max} = \sqrt{(S^2 - P^2)} \quad (2.5)$$

2.2.3 Frequency Droop Control

ตัวควบคุมไมโครกริดต้องทำให้แน่ใจว่าเกิดความเรียบและเปลี่ยนแปลงโดยอัตโนมัติขณะเปลี่ยนจากโหมดเชื่อมต่อกกริดเป็นโหมดแยกอิสระ ในขณะที่เปลี่ยนแปลงเป็นโหมดแยกอิสระนั้น MC ของแต่ละแหล่งจ่าย จะใช้การควบคุมกำลังและความถี่ เพื่อเปลี่ยนจุดที่ดำเนินการทำให้เกิดความสมดุลของกำลังไฟฟ้า กับโหลดใหม่ ตัวควบคุมจะต้องทำงานอย่างอิสระหลังจากการติดตามโหลดโดยไม่ต้องรอคำสั่งจาก CC หรือ MC ตัวอื่น รูปที่ 2.4 แสดงกราฟคุณลักษณะของการลดกำลังและความถี่ (P-f Drooping) ที่ใช้ใน MC สำหรับการควบคุม



รูปที่ 2.4 กราฟแสดงคุณลักษณะของกำลังจริงเทียบกับความถี่ (P-f Drooping)

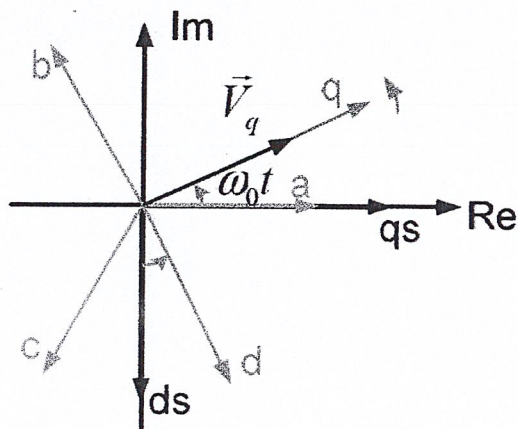
ขณะทำงานในโหมดเชื่อมต่อกกริดโหนดของไมโครกริดจะได้รับพลังงานจากทั้งกริดหลักและจากแหล่งจ่ายขึ้นอยู่กับความต้องการของผู้ใช้แต่หากกริดหลักเกิดการหยุดชะงัก (interrupt) เนื่องจากสาเหตุใดๆก็ตาม ไมโครกริดจะต้องสลับเปลี่ยนเป็นโหมดแยกอิสระอย่างราบเรียบ ในขณะที่เปลี่ยนโหมดนั้นมุมเฟสของแรงดันที่ MC จะเปลี่ยนแปลงด้วยนำไปสู่เกิดการตกลงของกำลังไฟฟ้าขาออกอย่างชัดเจน เป็นผลพวงทำให้ความถี่นั้นเปลี่ยนแปลงด้วย โดยในกรณีนี้แหล่งจ่ายแต่ละตัวจะมีการจัดสรรภาระเสียใหม่อย่างรวดเร็วโดยไม่ผ่านการควบคุมของ CC ยกตัวอย่างเช่น กำหนดให้แหล่งจ่าย 2 แหล่งทำงานที่ความถี่ปกติที่สุดที่ความถี่สูงสุดเป็น $P_{1\max}$ และ $P_{2\max}$ ในโหมด

เชื่อมต่อกับกริดทั้งสองจะทำงานที่ความถี่ที่คิดในการส่งกำลังเท่ากับ P_{01} และ P_{02} ตามลำดับ เมื่อความต้องการโหลดเปลี่ยนไปแหล่งจ่ายจะทำงานที่ความถี่ที่ต่างกันทำให้มุมกำลัง (δ) เปลี่ยนแปลง และความถี่ในการทำงานที่ตกลงไปด้วยกันซึ่งขึ้นอยู่กับสัดส่วนของการรับภาระในแต่ละตัว เกิดเป็นกราฟแสดงคุณลักษณะของกำลังกับความถี่ (P-f) ดังแสดงในรูปที่ (2.3)

เนื่องจากความถี่ในไมโครกริดลดลงแล้ว ดังนั้น MC ก็ต้องรวบรวมฟังก์ชันเพื่อฟื้นฟูการทำงานที่ความถี่ที่คิดภายใต้สัดส่วนของการรับภาระของโหลดที่เหมาะสมวิธีนี้สามารถใช้ได้เนื่องจากสมการ (2.3) แสดงให้เห็นว่า P_L นั้นแปรผันตรงกับมุมกำลัง δ (เปรียบเสมือนความถี่) จึงทำให้ตัวควบคุมสามารถเพิ่มการส่งจ่ายกำลังไฟฟ้าได้มากขึ้นโดยควบคุม $\angle\theta_E$ ให้มีความแตกต่างกันของมุมกำลัง δ มากขึ้น อันจะทำให้ความถี่ลดลงนั่นเอง

2.3 วิธีการควบคุมคอนเวอร์เตอร์

สมมุติให้แรงดันที่บัสนั้นสมดุล และแกน q ในกรอบอ้างอิง dq ที่หมุนด้วยความเร็วเชิงโรตารีนั้นใช้แกนร่วมกับเวกเตอร์แรงดันของบัส \vec{V} จะได้ $\vec{V}_d = 0, \vec{V}_q = \vec{V}$ และฟังก์ชันการแปลงจะถูกอธิบายโดยสมการที่ (2.6) คอนโทรลเลอร์สามารถใช้ \vec{V}_q เป็นตัวแปรควบคุม แทนที่จะใช้ \vec{V} ตามรูปที่ 2.5



รูปที่ 2.5 กรอบอ้างอิงของตัวควบคุมของ microsource

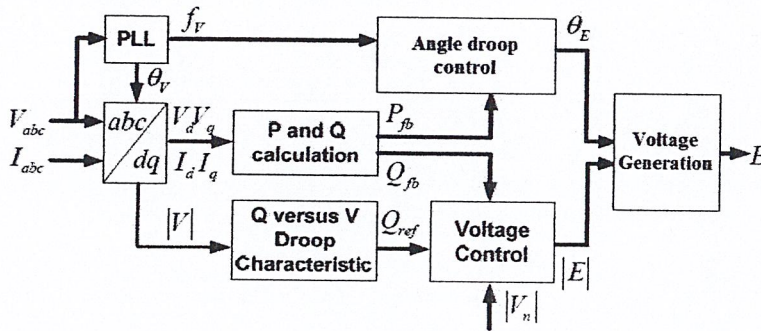
$$\begin{pmatrix} V_q \\ V_d \\ V_0 \end{pmatrix} = \frac{2}{3} \begin{pmatrix} 1 & \cos\theta_v & -\sin\theta_v \\ 0 & \sin\theta_v & \cos\theta_v \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ 0 & -\frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} V_a \\ V_b \\ V_c \end{pmatrix} \quad (2.6)$$

รูปที่ 2.6 แสดงภาพรวมของการควบคุม โดยความถี่ที่บัส f_v นั้นจะถูกปรับมาจาก phase locking loop (PLL) และขนาดของแรงดันที่บัส $|V|$ นั้นจะถูกคำนวณจากบล็อกรับการแปลง

กรอบอ้างอิงหมุน ω_s จะได้มาจากความสัมพันธ์ p-f droop characteristic ของสมการที่ 2.6 และ Q_{ref} จะได้มาจากโมดูล Q-V droop characteristic โดยโมดูล Angle Droop จะใช้เพื่อหามุมเฟส $\angle \theta_E$ แรงดันที่ออกมาจากอินเวอร์เตอร์ \vec{E} ในขณะที่โมดูล Voltage control นั้นจะถูกใช้เพื่อหาขนาดของแรงดันเอาต์พุต $|E|$ โดยแรงดันเอาต์พุต \vec{E} ที่ออกมาจะเกิดจากขนาดและมุมเฟสของฟังก์ชันไซน์นั่นเอง

สมการที่ 2.7 จะถูกใช้เพื่อคำนวณกำลังไฟฟ้าชั่วขณะที่จ่ายโดย microsource

$$\begin{aligned} P_{fb} &= 1.5(V_q I_q + V_d I_d) \\ Q_{fb} &= 1.5(V_q I_d - V_d I_q) \end{aligned} \tag{2.7}$$



รูปที่ 2.6 ภาพรวมของการควบคุม

2.3.1 โมดูลการควบคุมมุมแรงดัน

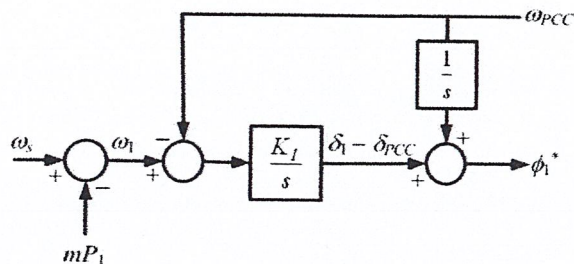
จากจุดเริ่มต้นของการควบคุมก็คือการใช้วิธี droop แบบธรรมดา ตามสมการ

$$\begin{aligned} \omega &= \omega_s - mP \\ V &= V^* - nQ \end{aligned} \tag{2.8}$$

เมื่อ m และ n คือ droop coefficient , ω_s คือความถี่เชิงโคโรนัส, v คือขนาดของแรงดันเอาต์พุตของคอนเวอร์เตอร์ และ ω คือความถี่ของมัน ในขณะที่ p และ q แทนกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือนที่จ่ายโดยคอนเวอร์เตอร์ตามลำดับ เพราะฉะนั้นความถี่และขนาดของแรงดันจะถูกควบคุมผ่านการควบคุมกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือนที่ออกจากแหล่งจ่าย DG.

ลูการควบคุมมุมแรงดันจะถูกแสดงตามรูปที่ 2.7 ความถี่ ω นั้นจะถูกคำนวณจากสมการที่ 1.8 ซึ่งหลังจากนั้นจะถูกเปรียบเทียบกับความถี่ที่ได้จากแรงดันตรงจุดPCC, ω_{pcc} ค่าความผิดพลาดที่ได้จะถูกรวมกับอินทิเกรตของ ω_{pcc} เพื่อหามุม δ_1^* โดยมุม δ_1^* ที่ได้จะหมุนด้วยความเร็วเชิงโคโรนัส ω_1 ซึ่งจะหามุม δ_1 กับแรงดันอ้างอิง

ทั้งนี้หากเราเปลี่ยนค่าของ K_f เราสามารถที่จะเปลี่ยนผลตอบสนองทางไดนามิกส์ของclose loopได้ โดยไม่มีผลกระทบกับการควบคุมความถี่ที่สภาวะคงตัว



รูปที่ 2.7 รูปการควบคุมของมุมแรงดัน

2.3.2 โมดูลควบคุมขนาดแรงดัน

เนื่องจากกำลังไฟฟ้าเสมือน Q_L นั้นเป็นสัดส่วนโดยตรงกับผลต่างของขนาด \vec{E} และ \vec{V} คอนโทรลเลอร์สามารถควบคุม Q_L โดยการเปลี่ยนขนาดของ \vec{E} โมดูลนี้นั้นจะถูกคำนวณขนาดแรงดันเอาต์พุตของอินเวอร์เตอร์ $|E|$ โดยที่ตัวแปรอินพุตคือ กำลังไฟฟ้าเสมือนที่ต้องการ Q_{ref} ตามสมการที่ 1.9 และกำลังไฟฟ้าเสมือนที่ถูกวัด Q_b แรงดันพิกัดที่โหนดบัส $|V_n|$ ส่วนตัวแปรเอาต์พุตคือขนาดของแรงดันของอินเวอร์เตอร์ที่ต้องการ $|E|$

$$Q_{ref} = n(V_{max} - |V|) \quad (2.9)$$

เมื่อ n คือ voltage droop coefficient

ส่วนประกอบของแรงดันที่จะควบคุม ΔV นั้นจะถูกสร้างมาจากค่าผิดพลาด ΔQ ผ่าน PI controller โดย $|E|$ นั้นคือผลรวมของแรงดันพิกัด $|V_n|$ และ ΔV

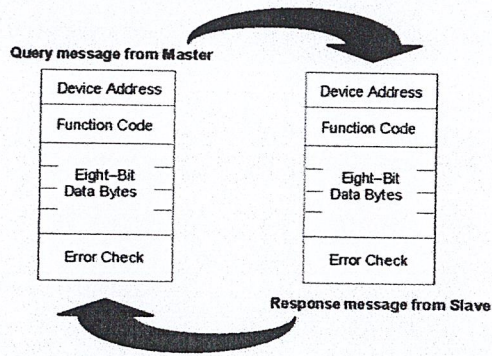
2.4 MODBUS Protocol

MODBUS เป็นโปรโตคอลเพื่อสื่อสารข้อมูลอินพุต/เอาต์พุตและรีจิสเตอร์ภายใน PLC ซึ่งถูกคิดค้นโดย Modicon (ปัจจุบันคือบริษัท Schneider Electric) โปรโตคอล MODBUS ได้เป็นที่ยอมรับกันอย่างกว้างขวางในการติดต่อสื่อสารที่เป็นแบบ Network Protocol อันเนื่องมาจาก MODBUS เป็นระบบเปิด, ไม่มีค่าใช้จ่าย, เชื่อมต่อและพัฒนาได้ง่าย พร้อมทั้งยังสามารถนำโปรโตคอลนี้ไปใช้งานในอุปกรณ์อื่นๆ เช่น Digital Power Meter, RTU (Remote Terminal Unit), Remote I/O, PLC เป็นต้น นอกจากนี้ MODBUS ยังสามารถรองรับและใช้งานร่วมกับ Application จำพวก SCADA และ HMI Software ได้อีกด้วย

โปรโตคอล MODBUS เป็นการสื่อสารข้อมูลในลักษณะ Master/Slave ซึ่งเป็นการสื่อสารจากอุปกรณ์แม่ (Master) เครื่องเดียว ส่วนใหญ่มักเป็นซอฟต์แวร์คอมพิวเตอร์หรืออุปกรณ์แสดงผล HMI ไปยังอุปกรณ์ลูก (Slave) ได้หลาย ๆ เครื่อง โดยสามารถกำหนดหมายเลขอุปกรณ์ได้สูงสุด 255 เครื่อง โดยมีลักษณะการส่งข้อมูล 2 แบบ คือ ข้อมูลแบบแอสกี (ASCII) และข้อมูลแบบเลขฐานสอง (Binary) ในโปรโตคอล MODBUS ที่สื่อสารข้อมูลแบบ ASCII จะเรียก MODBUS ASCII

และโปรโตคอล MODBUS ที่สื่อสารข้อมูลแบบเลขฐานสอง จะเรียก MODBUS RTU ทำให้มีความแตกต่างในการกำหนดค่าพอร์ตสื่อสาร

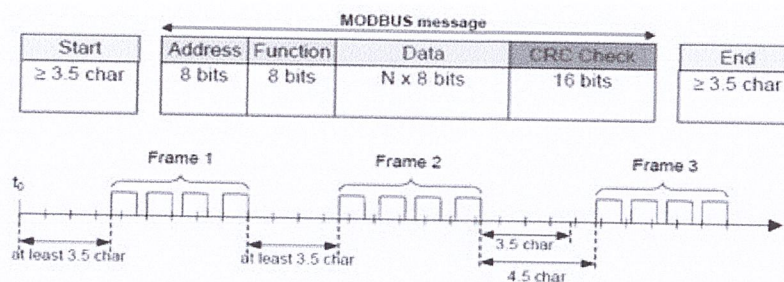
การรับส่งข้อมูลด้วยโปรโตคอล MODBUS สามารถเลือกได้ 2 โหมด คือ โหมด ASCII และ โหมด RTU ซึ่งทั้ง 2 โหมดนี้มีความแตกต่างกันที่การกำหนดรูปแบบของชุดข้อมูลภายในเฟรม จะเลือกโหมดใดก็ได้แต่มีเงื่อนไขว่า อุปกรณ์ทุกตัวที่ต่อร่วมกันอยู่ในบัสหรือเครือข่ายเดียวกันจะต้องตั้งให้เลือกใช้โหมดเดียวกันทั้งหมด



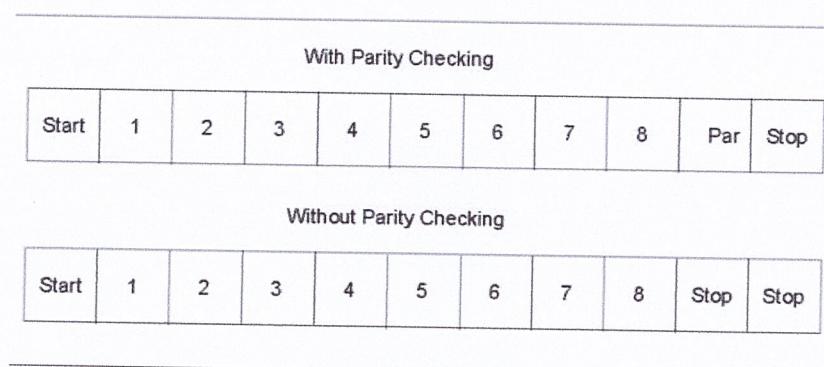
รูปที่ 2.8 การติดต่อสื่อสารแบบ Master/Slave

2.4.1 MODBUS RTU Mode

เฟรมข้อมูลในโหมด RTU ประกอบด้วยข้อมูลแสดงตำแหน่งแอดเดรส 1 ไบต์ (16 bits), หมายเลขฟังก์ชัน 1 ไบต์ (16 bits) ข้อมูลที่ทำการรับส่งจำนวนมากสุดไม่เกิน 252 ไบต์ และรหัสตรวจสอบความถูกต้องของข้อมูลแบบ CRC (Cyclical Redundancy Checking) ขนาด 2 ไบต์ ค่า CRC นี้เป็นค่าที่คำนวณมาจากข้อมูลทุกไบต์ ไม่รวมบิต Start, Stop และ Parity Check โดยที่ตัว Slave ตัวที่ส่งข้อมูลออกมาจะสร้างรหัส CRC แล้วส่งตามท้ายไบต์ข้อมูลออกมา หลังจากนั้นเมื่อ Master ได้รับเฟรมข้อมูลและถอดข้อมูลออกจากเฟรมแล้วจะทำการคำนวณค่า CRC ตามสูตรเดียวกับ Slave เพื่อทำการเปรียบเทียบค่า CRC ทั้ง 2 ค่าว่าตรงกันหรือไม่ หากไม่ตรงกันแสดงว่าเกิดความผิดพลาดในการรับส่งข้อมูลในโหมด RTU การรับส่งข้อมูล 1 ไบต์ ไม่ว่าจะเป็นข้อมูลส่วนใดภายในเฟรมจะต้องทำการส่งบิตข้อมูลรวม 11 บิต คือ บิตเริ่มต้น (Start) 1 บิต, บิตข้อมูล 8 บิต, บิตตรวจสอบ Parity ของข้อมูล 1 บิตและบิตหยุด 1 บิต (Stop) 1 บิต หากเลือกแบบไม่มีบิต Parity ก็จะเป็นแบบ Stop แทน 2 บิต สำหรับการกำหนดให้มีบิต Parity นั้น สามารถเลือกเป็นแบบคู่ (Even Parity) หรือคี่ (Odd Parity) ก็ได้ และหากต้องการออกแบบให้สอดคล้องกับอุปกรณ์ที่มีใช้กันทั่วไปมากที่สุด ควรเลือกแบบคู่โดยที่สามารถปรับเปลี่ยนเป็นแบบคี่หรือไม่มีการตรวจสอบ Parity (No Parity) ได้ด้วย



รูปที่ 2.9 ลักษณะเฟรมข้อมูลของ MODBUS RTU MODE



รูปที่ 2.10 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS RTU

2.4.2 MODBUS ASCII Mode

การรับส่งข้อมูลในโหมด ASCII นั้นมีความแตกต่างจากโหมด RTU ตรงที่ในโหมด RTU ข้อมูลที่จะส่งขนาด 1 ไบต์ นำมารวมกับบิตประกอบต่างๆ ก็สามารถส่งออกไปได้เลย แต่สำหรับโหมด ASCII จะมองข้อมูล 1 ไบต์ นั้นออกมาเป็นตัวอักษร 2 ตัว เช่น ค่า 0x5B ซึ่งเป็นเลขฐานสิบหก ก็จะถูกมองเป็นตัวอักษร '5' และตัวอักษร 'B' จากนั้นก็จะทำการค้นหารหัส ASCII ของตัวอักษรทั้ง 2 ตัวนั้น ซึ่งได้แก่ 0x35 สำหรับ '5' และ 0x42 สำหรับ 'B' แล้วทำการส่งรหัส ASCII ทั้ง 2 ค่านี้ ออกไป ซึ่งจะได้ผลเท่ากับการส่งค่า 0x5B ซึ่งเป็นข้อมูลขนาด 1 ไบต์

ในโหมด RTU จะเห็นได้ว่าการส่งข้อมูลในโหมด ASCII จะต้องทำงานมากกว่าการส่งข้อมูลในโหมด RTU ซึ่งทำให้อัตราเร็วในการสื่อสารมีค่าต่ำกว่า สาเหตุที่เป็นแบบนี้ก็เพราะว่า โหมด ASCII ได้ถูกออกแบบมาสำหรับอุปกรณ์ที่ไม่มีความสามารถในการกำหนดช่วงระยะเวลาของเวลาในการส่งเฟรมข้อมูล อย่างเช่นในโหมด RTU ที่อุปกรณ์สามารถกำหนดได้ว่าจะส่งเฟรมข้อมูลแต่ละเฟรมออกมาด้วยเวลาห่างกันเท่าใด และอุปกรณ์ที่รองรับข้อมูลก็ต้องสามารถตรวจจับและแยกแยะได้ว่า เฟรมข้อมูลแต่ละเฟรมที่รับเข้ามานั้นมีระยะเวลาห่างกันภายในช่วงเวลาที่กำหนดหรือไม่ เพื่อให้สามารถตรวจสอบหาจุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลแต่ละเฟรมได้

แต่ในความเป็นจริงยังมีอุปกรณ์อีกหลายชนิดที่ไม่มีความสามารถพิเศษนี้ จึงต้องใช้วิธีอื่นที่จะช่วยให้สามารถรับรู้จุดเริ่มต้นและจุดสิ้นสุดของเฟรมข้อมูลได้ นั่นได้แก่โหมด ASCII ซึ่งในโหมดนี้จะเริ่มต้นเฟรมข้อมูลด้วยการส่งรหัส ASCII ที่กำหนดให้หมายถึงจุดเริ่มต้น คือ 0x3A ซึ่งตรงกับตัวอักษร ':' ตามด้วยแอดเดรสของ Slave, หมายเลขฟังก์ชัน, ข้อมูล, รหัสตรวจสอบสอง RLC และ

รหัส ASCII 2 ตัว ที่กำหนดให้หมายถึงจุดสิ้นสุด คือ รหัส 0x0D และ 0x0A คือรหัส CR (Carriage Return) และ LF (Line Feed) ตามลำดับ โดยในขณะที่บัสข้อมูลว่างจากการรับส่งข้อมูล อุปกรณ์ทุกตัวจะคอยตรวจสอบข้อมูลในบัสว่ามีรหัส ASCII ของ ':' ออกมาหรือไม่ ถ้ามีก็จะรับรู้ว่าเป็นนี้ได้มีการเริ่มต้นส่งเฟรมข้อมูลออกมาแล้ว ก็จะเข้ากระบวนการรับข้อมูลต่อไป

Start	Address	Function	Data	LRC	End
1 char :	2 chars	2 chars	0 up to 2x252 char(s)	2 chars	2 chars CR,LF

รูปที่ 2.11 ลักษณะเฟรมข้อมูลของ MODBUS ASCII

With Parity Checking

Start	1	2	3	4	5	6	7	Par	Stop
-------	---	---	---	---	---	---	---	-----	------

Without Parity Checking

Start	1	2	3	4	5	6	7	Stop	Stop
-------	---	---	---	---	---	---	---	------	------

รูปที่ 2.12 ลักษณะข้อมูลแต่ละไบต์ของ MODBUS ASCII

MODBUS จะบริการให้อุปกรณ์ติดต่อสื่อสารกันผ่าน Serial Port (RS-232/422/485) แต่ในปัจจุบันได้มีการพัฒนาให้อุปกรณ์สามารถติดต่อสื่อสารกับอุปกรณ์ที่อยู่บนเครือข่าย Ethernet ซึ่งอุปกรณ์ที่ใช้การสื่อสารแบบ MODBUS Protocol ส่วนใหญ่จะเป็น PLCs, DCSs, HMIs, Instruments

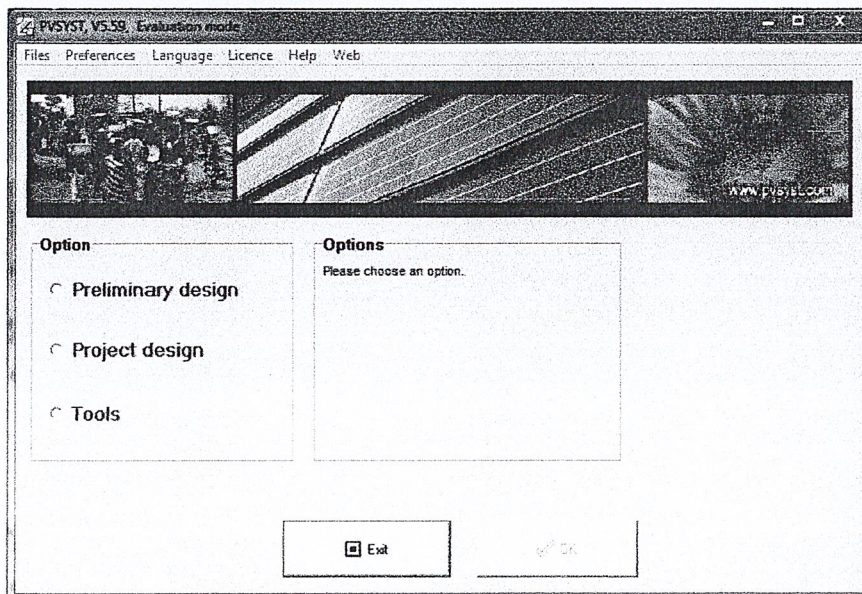
บทที่ 3

การออกแบบและทดลองระบบผลิตไฟฟ้าจากเซลล์แสงอาทิตย์

3.1 แนะนำโปรแกรม PV Syst Ver.5.59 เบื้องต้น

โปรแกรมที่ใช้ในการออกแบบการต่อแผง PV เข้ากับ INVERTER คือ โปรแกรม PVsyst Ver.5.59 โดยโปรแกรม PVsyst Version 5.0 เป็นโปรแกรมสำหรับการศึกษา , ออกแบบ ขนาด และการวิเคราะห์ข้อมูลของระบบของ PV โดยสมบูรณ์ โดยจำแนกการวิเคราะห์ระบบ PV เป็นแบบต่างๆ ได้แก่ ระบบ PV ที่ต่อเข้ากับระจําหน่าย (grid-connected) ระบบ PV ชนิดติดตั้งอิสระ (stand-alone) ระบบ PV สำหรับระบบปั้มน้ (pumping) และระบบ PV แบบไฟฟ้ากระแสตรง (DC grid) รวมทั้งฐานข้อมูลของสภาพอากาศส่วนประกอบต่างๆของระบบ PV เช่น แผง PV Inverter เป็นต้น และกล่องเครื่องมือเกี่ยวกับพลังงานแสงอาทิตย์

PVsyst Ver.5.59 จะแบ่งระดับการศึกษาออกเป็น 3 ระดับ เพื่อตอบสนองต่อระดับการใช้งานต่างๆ ได้แก่

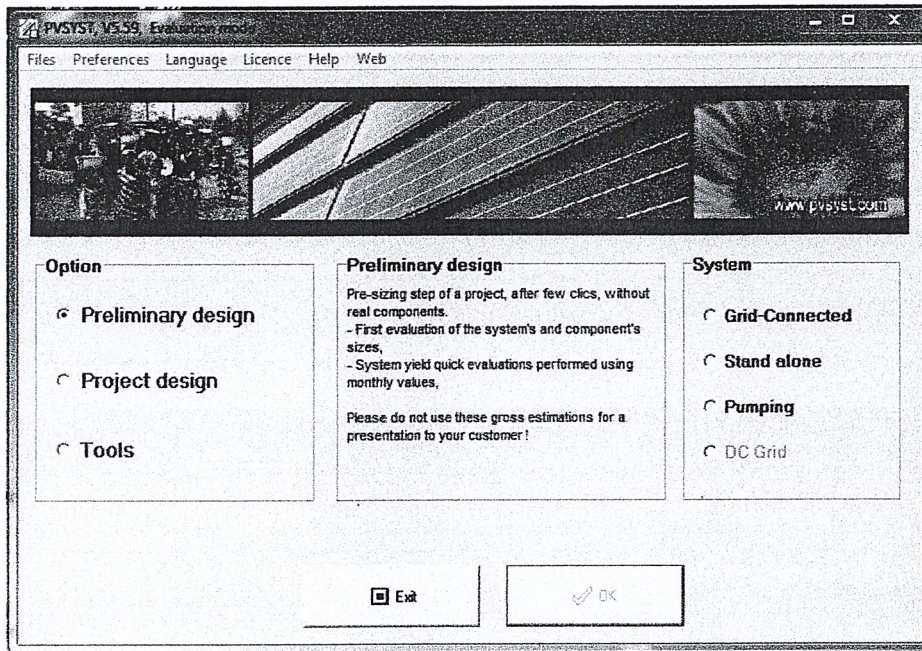


รูปที่ 3.1 แบ่งระดับการศึกษาของ PVsyst Ver.5.59

3.1.1 Preliminary design (การออกแบบเบื้องต้น)

ในโหมดนี้เป็นการประเมินผลผลิตที่ได้และค่าใช้จ่ายคร่าวๆแบบรายเดือนอย่างรวดเร็วมาก เนื่องจากจะพิจารณาคุณสมบัติและค่าตัวแปรทั่วไปของระบบ โดยไม่คำนึงถึงส่วนประกอบต่างๆที่เกิดขึ้นจริงสามารถเลือกระบบที่ต้องการวิเคราะห์ได้ ดังนี้

- ระบบ PV ที่ต่อเข้ากับระจําหน่าย (grid-connected)
- ระบบ PV ชนิดติดตั้งอิสระ (stand-alone)
- ระบบ PV สำหรับระบบปั้มน้ (pumping)



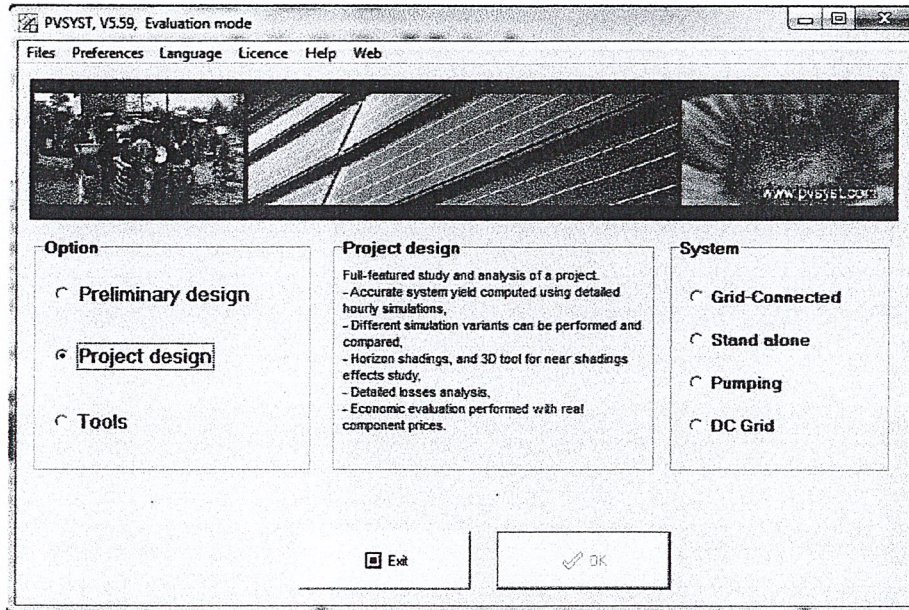
รูปที่ 3.2 Preliminary design (การออกแบบเบื้องต้น)

3.1.2 Project Design (การออกแบบโครงการ)

ในโหมดนี้เป็นการออกแบบระบบและจำลองรายละเอียดต่างๆเป็นรายชั่วโมงโดย โหมดนี้ผู้ใช้สามารถทำการจำลองระบบต่างๆแล้วนำมาเปรียบเทียบกันได้อีกทั้งยังสามารถช่วยในการ ออกแบบ PV array (จำนวนแผงในแต่ละseries และ parallel) ให้เหมาะสมกับชนิดของ Inverter Battery หรือปั๊ม ที่เราเลือกไว้

ผู้ใช้สามารถระบุค่าตัวแปรเพิ่มเติมและวิเคราะห์ผลกระทบต่างๆได้อย่างละเอียด เช่น พฤติกรรมความร้อน สาย คุณภาพของแผง รวมทั้งการสูญเสียเนื่องจากมีสิ่งของมาบัง และมุม องศาการติดตั้ง

ส่วนรายละเอียดการประเมินผลด้านเศรษฐศาสตร์ ผู้ใช้สามารถระบุค่าใช้จ่ายของ ส่วนประกอบได้และค่าใช้จ่ายเพิ่ม รวมทั้งเงื่อนไขในการลงทุน เพื่อที่จะนำมาวิเคราะห์ในการเลือกหา อุปกรณ์ของระบบและประเภทของระบบที่เหมาะสมที่สุด

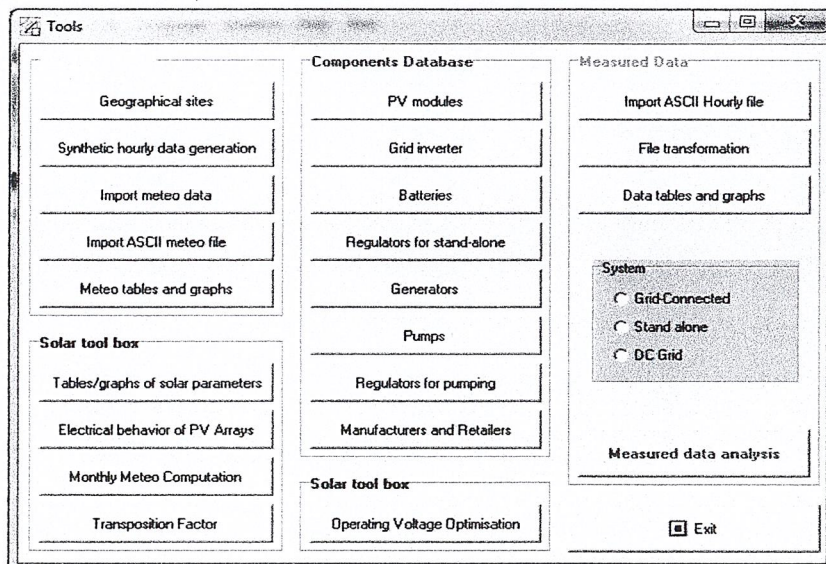


รูปที่ 3.3 Project Design (การออกแบบโครงการ)

3.1.3 Measured data analysis (การวิเคราะห์ข้อมูลที่วัดได้)

ในส่วนนี้จะอยู่ใน tools ซึ่งอนุญาตให้นำข้อมูล(ส่วนใหญ่เป็นไฟล์ASCII) เข้ามาเพื่อสร้างตารางและกราฟประสิทธิภาพในการทำงานจริง แล้วนำตัวแปรที่ได้จากการจำลองมาเปรียบเทียบกัน

3.1.4 Tools

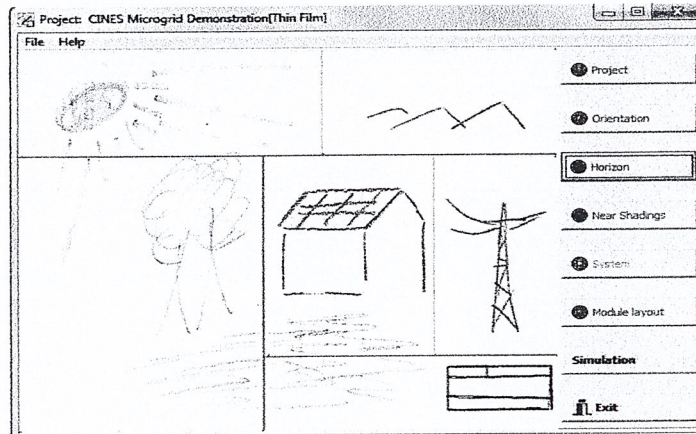


รูปที่ 3.4 Tools (เครื่องมือ)

จะประกอบด้วย ฐานข้อมูลของส่วนประกอบต่างในระบบPV เช่น inverter Generator PV Module Batteries เป็นต้น Solar tools box,สภาพภูมิอากาศ รวมทั้งข้อมูลที่ทำ การวัดแล้ว (Measured data)

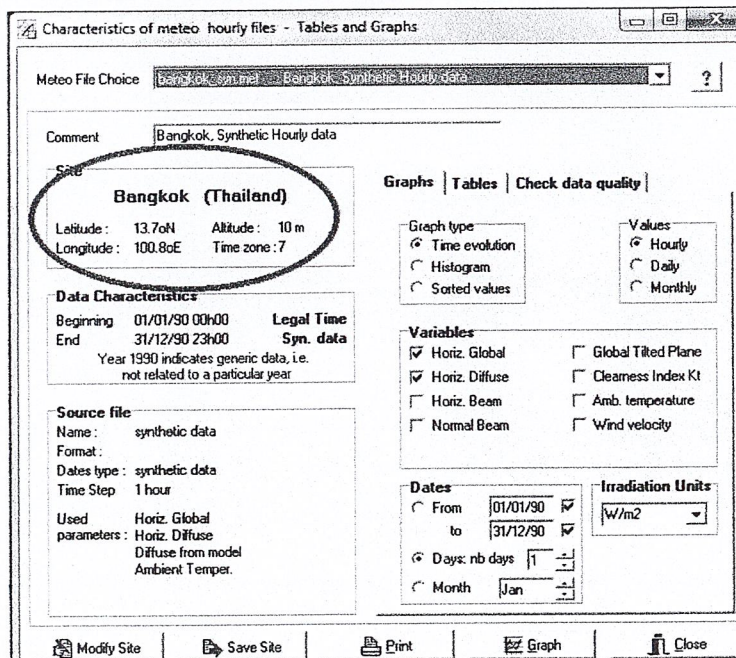
3.2 การออกระบบการต่อแผง PV เข้ากับ Inverter โดยใช้โปรแกรม

ในการออกแบบจะใช้โหมด Project design แบบ grid connected ในการจำลอง และออกแบบตาม assignment ที่กำหนด ชื่อไฟล์ที่ใช้ในการจำลอง คือ CINES Microgrid Demonstration โดยเราจะต้องทำการตั้งค่าต่างๆ ตามที่เราออกแบบไว้ ดังนี้

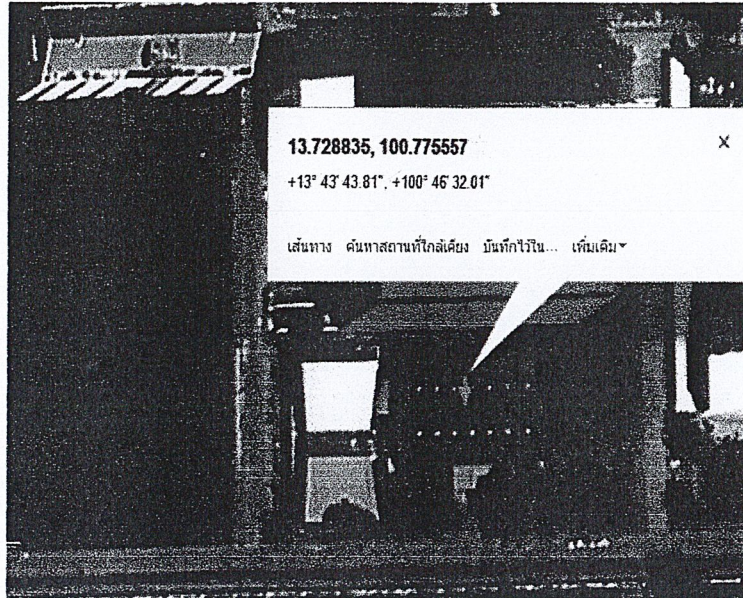


รูปที่ 3.5 โหมด Project design แบบ grid connected

Project จะเป็นการใส่รายละเอียดต่างๆ เช่น ชื่อ ที่อยู่ เป็นต้น รวมทั้งการเลือกที่ตั้ง และสภาพภูมิอากาศที่จะติดตั้งแผง ในการจำลองจะเลือกเป็น Bangkok ดังรูป



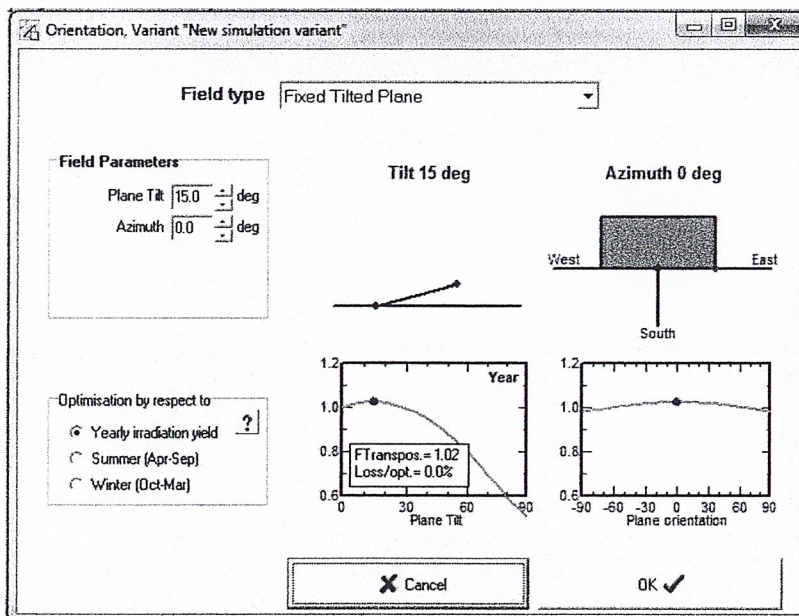
รูปที่ 3.6 ค่าพิกัดต่างๆถูกตั้งค่าไว้ในโปรแกรม



รูปที่ 3.7 พิกัดตึก L จาก Google Earth

จากรูปที่ 3.7 จะเห็นว่า ตึก L จะมี Latitude และ Longitude ใกล้เคียงกันกับที่ตั้งไว้ในโปรแกรม

ได้
Orientation จะเป็นการกำหนดลักษณะการติดตั้งแผง PV พร้อมแสดงค่าพลังงานที่



รูปที่ 3.8 กำหนดลักษณะการติดตั้งแผง PV

ในประเทศไทยตำแหน่งการแผง PV ที่ดีที่สุด คือ การวางให้หันหน้าแผงไปทางทิศใต้ มีมุมเงยประมาณ 15 องศา ซึ่งในการออกแบบเราจะติดตั้งแผงให้อยู่กับที่และมีมุมเงย 15 องศา ดังรูป 3.8

The screenshot shows the 'Grid system definition' window with the following configuration details:

- Global System configuration:**
 - Number of kinds of sub-fields: 1
 - Simplified Schema: [checked]
- Global system summary:**
 - Nb. of modules: 24
 - Module area: 34 m²
 - Nb. of inverters: 1
 - Nominal PV Power: 2.9 kWp
 - Maximum PV Power: 2.9 kWdc
 - Nominal AC Power: 3.0 kWac
- Homogeneous System:**
 - Presizing Help: No Sizing
 - Enter planned power: 3.0 kWp
 - or available area: 36 m²
- Select the PV module:**
 - Sort modules: Power
 - Technology: uCS+SiH
 - Manufacturer: Sharp
 - Module: NA-F121 (GK)
 - Approx. needed modules: 25
 - Sizing voltages: V_{mpp} (60°C) 41.8 V, V_{oc} (-10°C) 64.7 V
- Select the inverter:**
 - Sort inverters by: Power
 - Manufacturer: SMA
 - Model: Sunny Boy SB 3000 TL-20
 - Nb. of inverters: 1
 - Operating Voltage: 188-440 V
 - Global Inverter's power: 3.0 kWac
 - Input maximum voltage: 550 V
- Design the array:**
 - Number of modules and strings:
 - Initial degrad: 10% (should be 2)
 - Mod. in series: 8 (between 5 and 7)
 - Nbre strings: 3 (only possibility 3)
 - Overload loss: 0.0%
 - Prnom ratio: 0.97
 - Operating conditions:
 - V_{mpp} (60°C): 335 V
 - V_{mpp} (20°C): 385 V
 - V_{oc} (-10°C): 518 V
 - Plane irradiance: 1000 W/m² (Max. in data)
 - I_{mpp} (STC): 7.9 A (Max. operating power)
 - I_{sc} (STC): 10.0 A (at 1000 W/m² and 50°C)
 - I_{sc} (at STC): 10.0 A
 - Array nom. Power (STC): 2.9 kWp

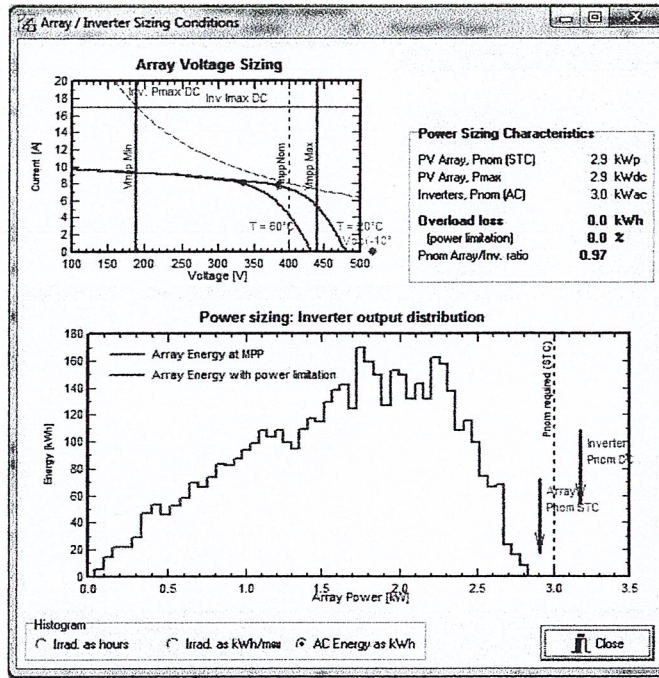
รูปที่ 3.9 PV แบบ Thin Film รุ่น NA-F121GK

System ส่วนนี้เป็นการใส่รายละเอียดต่างๆที่เราออกแบบไว้ รวมทั้ง spec ของอุปกรณ์ในระบบด้วยในการออกแบบระบบจะแบ่งการจำลองออกเป็น 2 ระบบ ได้แก่ ระบบแผง PV แบบ Thin Film (NA-F121GK) และ ระบบแผง PV แบบ Polycrystalline (ND-130T1J)

ระบบแผง PV แบบ Thin Film (NA-F121GK)

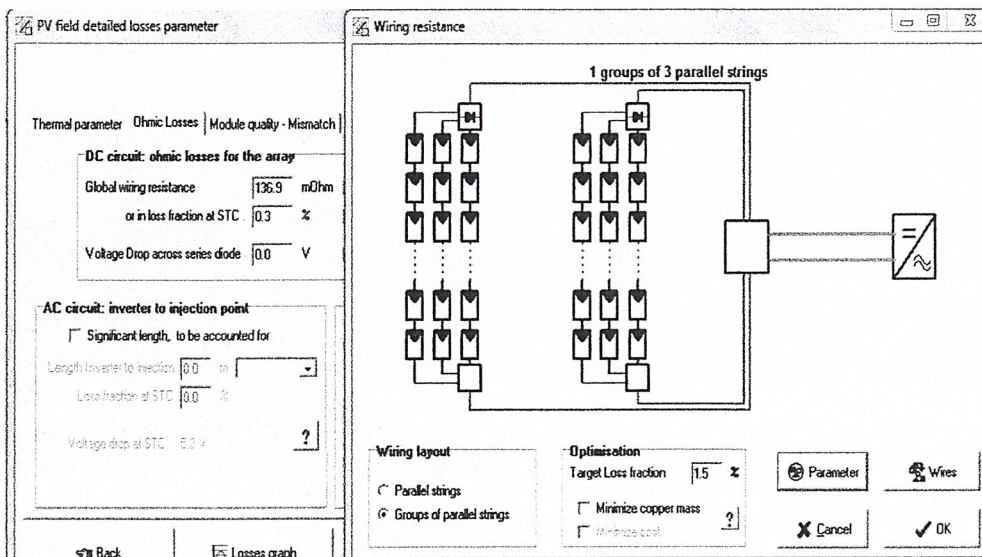
การจำลองระบบขนาด 3.0 kW จะใช้ PV คู่กับ Inverter ขนาด 3.0 kW รุ่น Sunny Boy SB 3000 TL-20 ของ SMA และเราจำกัดจำนวนโมดูลไม่เกิน 24 แผง

เมื่อเราทำการออกแบบการต่อแผง PV ให้ได้จุดการทำงานที่ดีที่สุดแล้ว จากรูปจะเห็นว่า จุดแรงดันขณะทำงานที่อุณหภูมิต่างๆมากกว่าจุดแรงดันการทำงานต่ำสุดและน้อยกว่าจุดแรงดันการทำงานมากที่สุดของตัว Inverter และเลือกการต่อแผงโมดูลอนุกรมกัน 8 แผงเป็น 3 สตริง รวมเป็น 24 แผง จะได้ Total Power ที่ 2.9 kWp



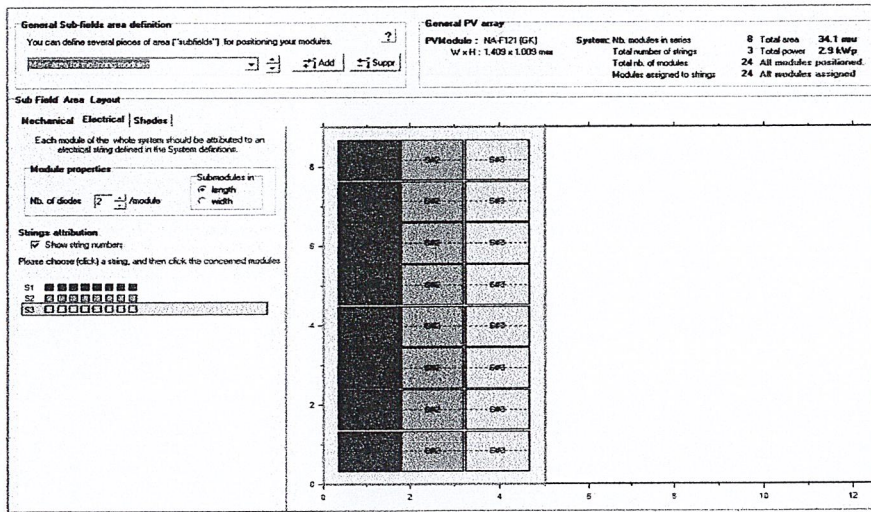
รูปที่ 3.10 กราฟการแสดงผลงาน

จากรูป เป็นการแสดงจุด Vmpp ที่ 60 องศา, 20 องศาและ -10 องศาจะเห็นได้ว่า จุด Vmpp ที่ 60 องศา, 20 องศา จะอยู่ในช่วงการทำงานของ Inverter แต่ที่จุด -10 องศาอยู่นอกการทำงาน เนื่องจากประเทศไทยแทบจะเป็นไปไม่ได้เลยที่จะมีอุณหภูมิต่ำขนาดนั้น เราจึงไม่สนใจจุดดังกล่าว ส่วนกราฟด้านหลังเป็นกราฟแสดงความสัมพันธ์ระหว่างพลังงานที่ได้จากแผง ที่ MPP (Maximum power point) และพลังงานที่ได้จาก Inverter (kWh)



รูปที่ 3.11 รูปแบบการต่อ PV

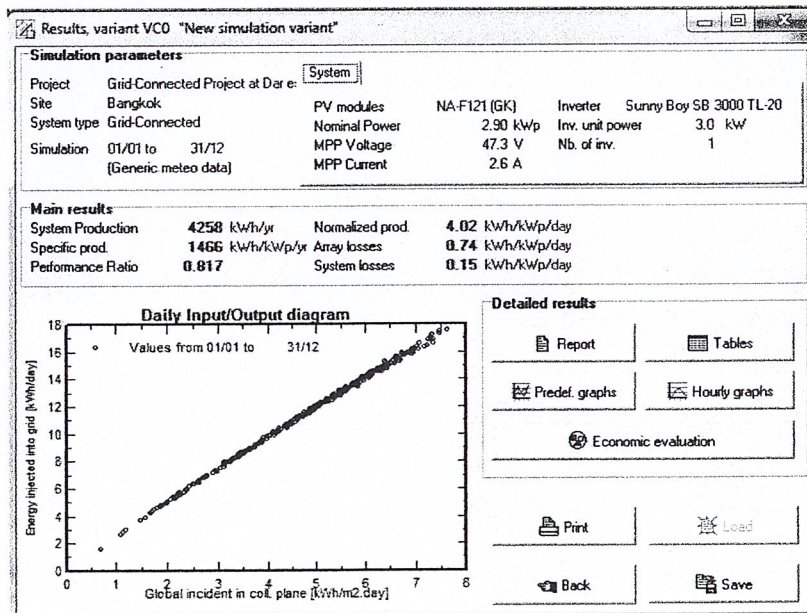
จากนั้นเราจะตั้งค่าความต้านทานสาย ที่ 136.9 mOhm และการสูญเสียต่างๆ 0.3 % เราจะได้รูปแบบการต่อโดยเลือกการต่อแบบกลุ่ม ผ่านไดโอด Junction Box และเข้าสู่ Inverter ดังรูปที่ 3.11 ด้านบน



รูปที่ 3.12 การคำนวณหาขนาดพื้นที่

หลังจากนั้นเราจะมาทำการคำนวณหาขนาดพื้นที่ที่ต้องการในการติดตั้ง จากรูปที่ 12 ด้านล่าง เราจะได้พื้นที่ที่สามารถติดตั้งแผง PV ได้ทั้งหมด คือ ขนาด 34 ตารางเมตร

ขั้นตอนสุดท้ายจะทำการ Simulation หาค่าพลังงานที่ได้ต่อปี จากรูป 3.13 จะได้ 4258kWh/yr และพลังงานที่ได้ต่อวัน คือ 4.02 kWh/kWp/day



รูปที่ 3.13 ค่าที่ได้จากการ Simulation

Estimation Maximum Power

จาก Datasheet

$$P_{\max, \text{input}} = 121 \times 24 = 2904 \text{ W}$$

$\alpha P_m (\%/^{\circ}\text{C}) -0.24$

$$\text{At } 60^{\circ} \quad \alpha = -0.24 \times (60 - 25) = -8.4 \%$$

$$P_{\max, \text{input}} = 0.916 \times 2904 = 2660.1 \text{ W}$$

At 96% Inverter efficiency

$$P_{\max, \text{output}} = 0.96 \times 2660.1 = 2553.7 \text{ W}$$

พบว่าเราเลือกพิกัดของอินเวอร์เตอร์ ที่ 3 kW เป็นขนาดที่เหมาะสม

PV แบบ Polycrystalline รุ่น ND-130T1J

The screenshot shows the 'Grid system definition' software interface with the following configuration details:

- Global System configuration:**
 - Number of kinds of sub-fields: 1
 - Simplified Schema:
- Global system summary:**

Nb. of modules	24	Nominal PV Power	3.1 kWp
Module area	24 m ²	Maximum PV Power	3.0 kWdc
Nb. of inverters	1	Nominal AC Power	3.0 kWac
- Homogeneous System:**
 - Presizing Help: Enter planned power kWp, ... or available area m²
 - Select the PV module:
 - Sort modules: Power, Technology, Manufacturer
 - Module list: 130 Wp 15V Si-poly ND-130T1J SHARP
 - Sizing voltages: V_{mpp} (60°C) 14.9 V, V_{oc} (-10°C) 24.5 V
 - Select the inverter:
 - Sort inverters by: Power, Voltage (max), Manufacturer
 - Inverter list: 3.0 kW 188-440 V 50/60Hz Sunny Boy SB 3000 TL-20 Add100V SMA
 - Nb. of inverters: 1
 - Operating Voltage: 188-440 V, Global Inverter's power: 3.0 kWac
 - Input maximum voltage: 650 V
- Design the array:**
 - Number of modules and strings:
 - Mod. in series: 24 (should be between 13 and 24)
 - Nbre strings: 1
 - Overload loss: 0.0%
 - P_{nom} ratio: 1.04
 - Operating conditions:
 - V_{mpp} (60°C): 358 V
 - V_{mpp} (20°C): 429 V
 - V_{oc} (-10°C): 589 V
 - Plane irradiance: 1000 W/m² (Max. in data STC)
 - Impp (STC): 7.4 A, Max. operating power: 2.8 kW
 - Isc (STC): 8.2 A (at 1000 W/m² and 50°C)
 - Isc (at STC): 8.1 A
 - Array nom. Power (STC): 3.1 kWp

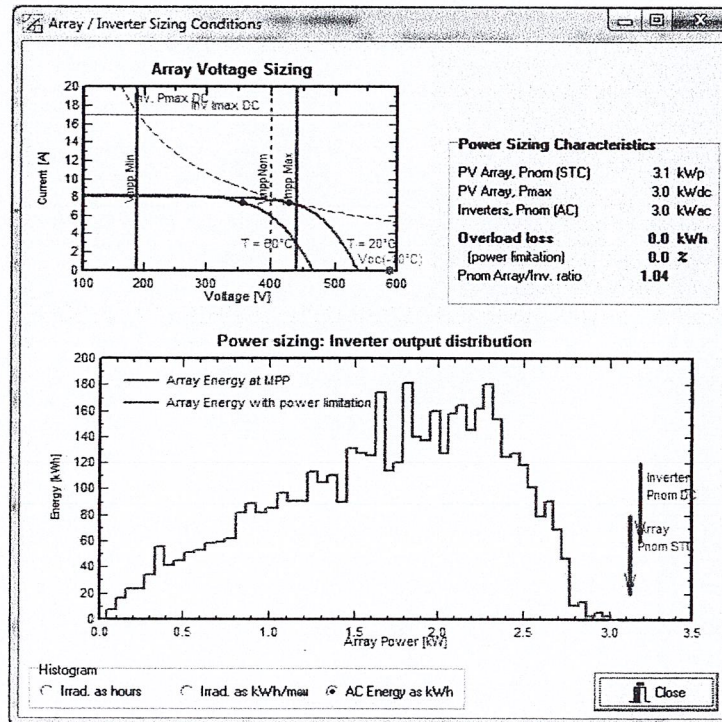
Buttons at the bottom: User's needs, Detailed losses, Cancel, OK

รูปที่ 3.14 PV แบบ Polycrystalline รุ่น ND-130T1J

ในการจำลองระบบขนาด 3.0 kW จะใช้ PV คู่กับ Inverter ขนาด 3.0 kW รุ่น Sunny Boy SB 3000 TL-20 ของ SMA แบบเดี่ยวชนิด Thin Film และจำกัดจำนวนโมดูลไม่เกิน 24 แผง

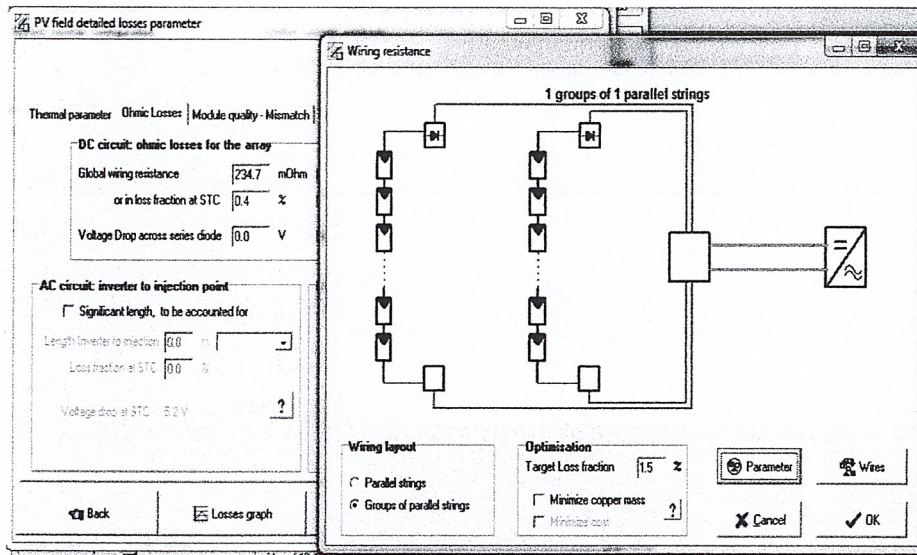
หมายเหตุ : ที่ -10 องศา V_{oc} มีค่ามากกว่า Input Maximum Voltage ของ Inverter ซึ่งในทางปฏิบัติประเทศไทยแทบจะเป็นไปไม่ได้ที่จะมีอุณหภูมิต่ำถึง -10 องศา เราจึงปรับเปลี่ยนค่า Input Maximum Voltage ของ Inverter ใหม่ เพื่อให้โปรแกรมสามารถทำการ Simulation ได้ จาก 550 V เป็น 650V

เมื่อเราทำการออกแบบการต่อแผง PV ให้ได้จุดการทำงานที่ดีที่สุดแล้ว จากรูปจะเห็นว่า จุดแรงดันขณะทำงานที่อุณหภูมิต่างๆมากกว่าจุดแรงดันการทำงานต่ำสุด และน้อยกว่าจุดแรงดันการทำงานมากที่สุดของตัว Inverter และเลือกการต่อแผงโมดูลอนุกรมกัน 24 แผง 1 สตริง รวมเป็น 24 แผง จะได้ Total Power ที่ 3.1 kWp



รูปที่ 3.15 กราฟแสดงการทำงาน

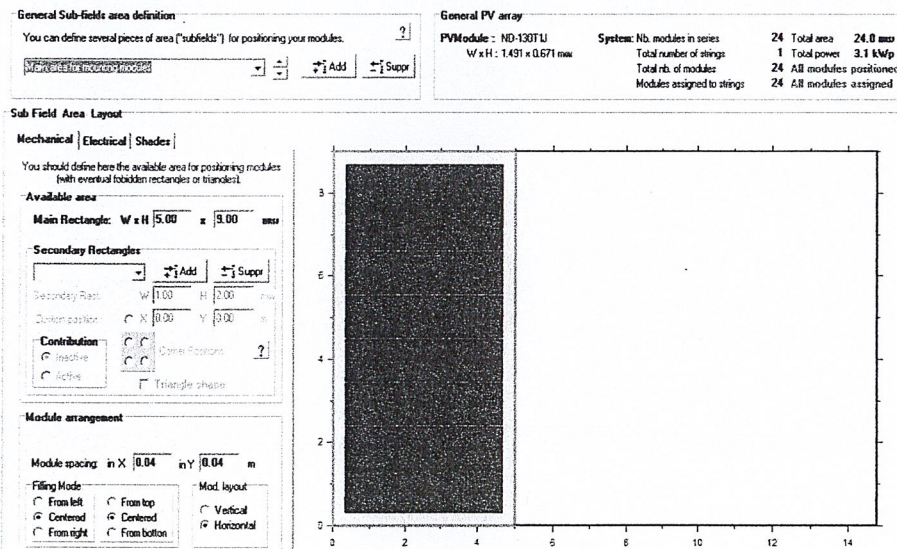
จากรูป เป็นการแสดงจุด V_{mpp} ที่ 60 องศา, 20 องศาและ -10 องศาจะเห็นได้ว่าจุด V_{mpp} ที่ 60 องศา, 20 องศา จะอยู่ในช่วงการทำงานของ Inverter แต่ที่จุด -10 องศา อยู่นอกการทำงาน เนื่องจากประเทศไทยแทบจะเป็นไปไม่ได้เลยที่จะมีอุณหภูมิต่ำขนาดนั้น เราจึงไม่สนใจจุดดังกล่าว ส่วนกราฟด้านล่างเป็นกราฟแสดงความสัมพันธ์ระหว่างพลังงานที่ได้จากแผง ที่ MPP (Maximum power point) และพลังงานที่ได้จาก Inverter (kWh)



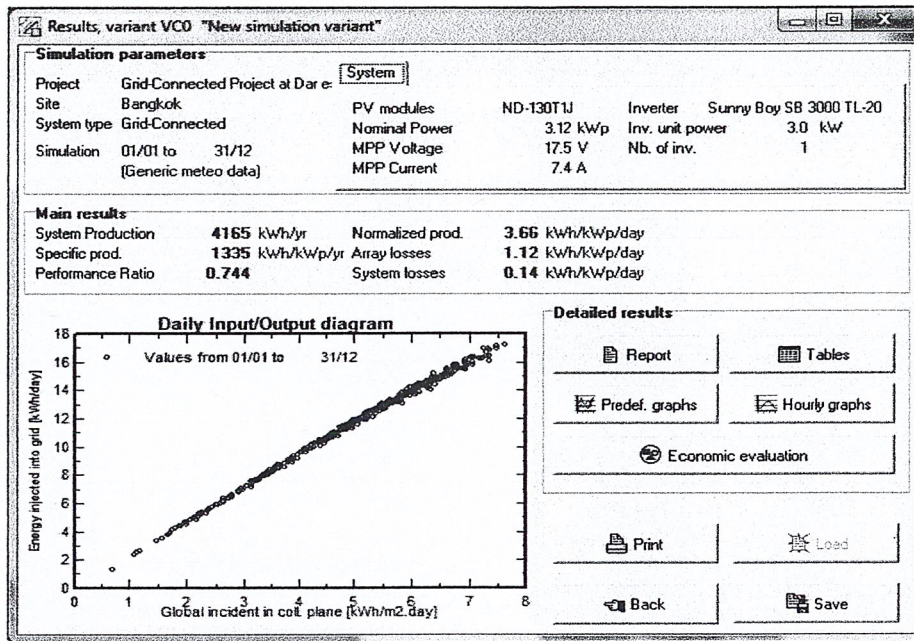
รูปที่ 3.16 รูปแบบการต่อ PV

จากนั้นเราจะตั้งค่าความต้านทานสาย ที่ 136.9 mOhm และการสูญเสียต่างๆ 0.3 % เราจะได้รูปแบบการต่อโดยเลือกการต่อแบบกลุ่ม ผ่านไดโอด Junction Box และเข้าสู่ Inverter ดังรูปบน

หลังจากนั้นเราจะมาทำการคำนวณหาขนาดพื้นที่ที่ต้องการในการติดตั้ง จากรูปด้านล่าง เราจะได้พื้นที่ที่สามารถติดตั้งแผง PV ได้ทั้งหมด คือ ขนาด 24 ตารางเมตร



รูปที่ 3.17 คำนวณหาขนาดพื้นที่ที่ต้องการในการติดตั้ง



รูปที่ 3.18 ทำการ Simulation หาค่าพลังงานที่ได้ต่อปี

ขั้นตอนสุดท้ายจะทำการ Simulation หาค่าพลังงานที่ได้ต่อปีจะได้ 4150kWh/yr และพลังงานที่ได้ต่อวัน คือ 3.66 kWh/kWp/day

Estimation Maximum Power

จาก Datasheet

$$P_{\max, \text{input}} = 130 \times 24 = 3120 \text{ W}$$

αP_m (%/°C) -0.485

$$\text{At } 60^\circ \alpha = -0.485 \times (60 - 25) = -16.975$$

$$P_{\max, \text{input}} = 0.83 \times 3120 = 2589.6 \text{ W}$$

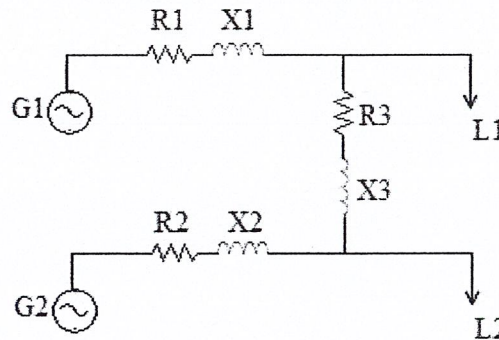
At 96% Inverter efficiency

$$P_{\max, \text{output}} = 0.96 \times 2589.6 = 2486.02 \text{ W}$$

พบว่าเราเลือกฟักัดของอินเวอร์เตอร์ ที่ 3 kW เนื่องจากแรงดันรวมของแผงมันต่ำกว่าจุดทำงานของ Inverter SMA 3000TL

3.3 การออกแบบการทำงานของไมโครกริดในโหมด stand-alone ของระบบ 3 เฟส

3.3.1 กำหนดรายละเอียดสำหรับไมโครกริดระบบ 3 เฟสที่จะใช้ในการทดลอง

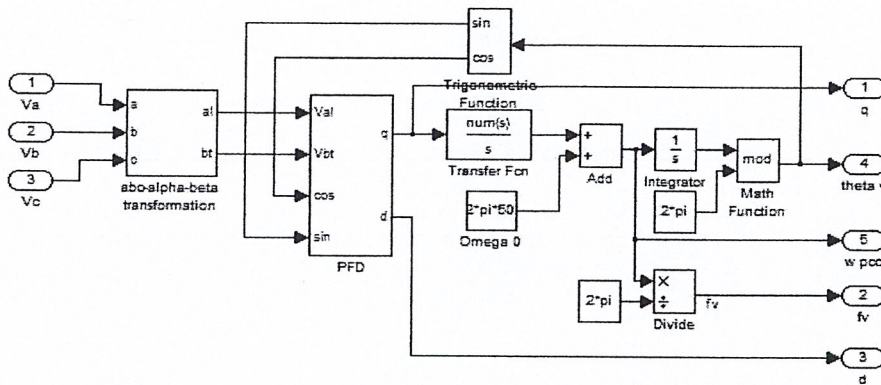


รูปที่ 3.19 วงจรสมมูลเฟสเดียวของระบบที่ใช้ในการทดลอง

ตารางที่ 3.1 ตารางแสดงค่าพารามิเตอร์ของวงจรสมมูลที่ใช้ในการทดลอง

อุปกรณ์	ค่าพารามิเตอร์
G1	power rating 10kVA , Designed for pf = 0.8
G2	power rating 10kVA , Designed for pf = 0.8
R1	0.125 Ω
X1	3.2e-3 H
R2	0.25 Ω
X2	6.4e-3 H
R3	0.25 Ω
X3	100e-6 H
L1	24.2 Ω
L2 (R+jXI)	Case load R Z = 72.6 , 36.3000 , 24.2 , 18.15 , 14.52 Ω
เปลี่ยนแปลง แล้วแต่กรณี	Case load RL at P.F.=0.9 Z= 65.34+31.6456j , 32.67+15.8228j , 21.78+10.5485j , 16.335+7.9114j , 13.068+6.3291j Ω
	Case load RL at P.F.=0.8 Z= 58.08+43.56j , 29.04+21.78j , 19.36+14.52j , 14.52+10.89j , 11.616+8.712j Ω

3.3.2 ออกแบบ phase lock loop ชนิด 3 เฟส โดยใช้สมการที่ 2.6 ในการแปลงแกน



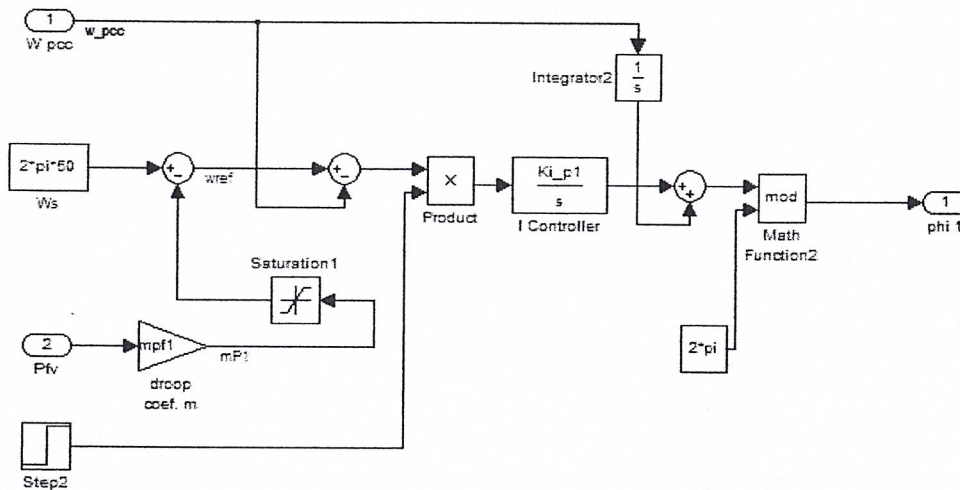
รูปที่ 3.20 ภายใน abc-dq Voltage With 3 phase PLL subsystem

3.3.3 ออกแบบ frequency droop coefficient

เป็นตัวกำหนดกำลังไฟฟ้าจริง โดยยอมให้ความถี่ของระบบตกลงได้ไม่เกิน 3% ของความถี่มูลฐาน ตามความสัมพันธ์ของสมการที่ 1.8

$$\text{Frequency droop coefficient} = \left(\frac{f_{\max} - f_{\min}}{P_{\max}} \right) * 2\pi$$

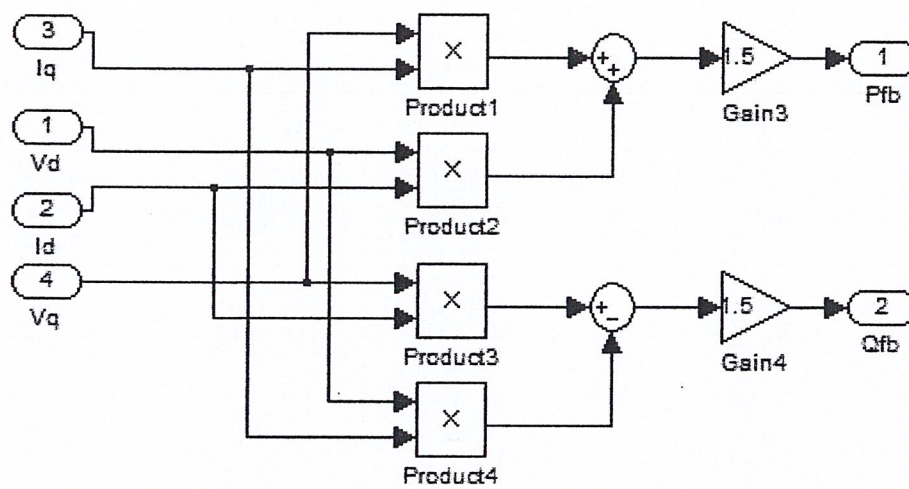
$$\text{mpf1} = \left(\frac{50 - 48.5}{8000} \right) * 2\pi = 0.0001875 * 2\pi$$



รูปที่ 3.21 ภายใน angle droop control subsystem

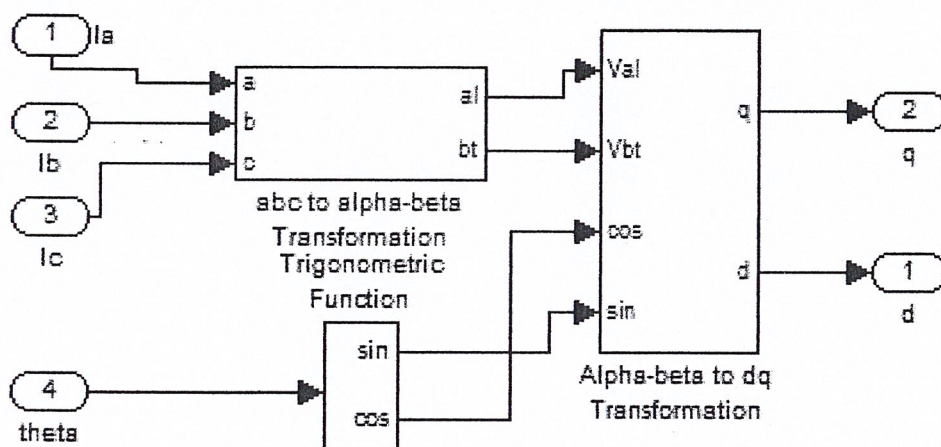
3.3.4 ออกแบบ subsystem ที่ชื่อว่า P&Q calculation

ตามความสัมพันธ์ของสมการที่ 2.7



รูปที่ 3.22 ภายใน P&Q calculation subsystem

3.3.5 ออกแบบการแปลงแกนของกระแส 3 เฟสเป็นแกน d-q ตามรูปที่ 3.5



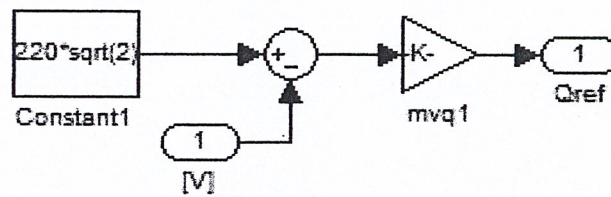
รูปที่ 3.23 ภายใน abc-dq current subsystem

3.3.6 ออกแบบ voltage droop coefficient

เป็นตัวกำหนดกำลังไฟฟ้รีแคทีฟ โดยให้มีค่าความผิดพลาดของขนาดของแรงดันอยู่ในย่านไม่เกิน 2.5% ตามความสัมพันธ์ของสมการที่ 2.8

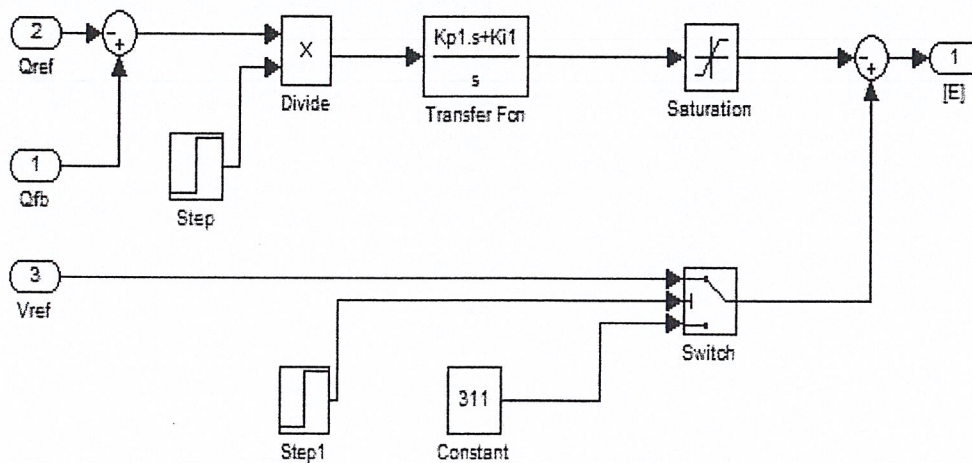
$$\text{Voltage droop coefficient} = \left(\frac{V_{\max} - V_{\min}}{Q_{\max}} \right)$$

$$mvq1 = \left(\frac{225.5 - 214.5}{6000} \right) * \sqrt{2} = 0.002592725$$



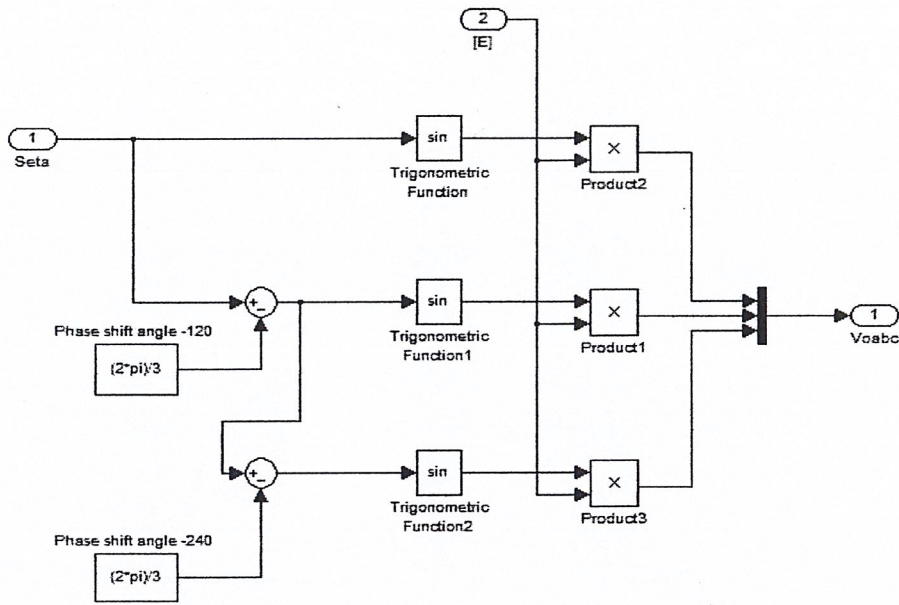
รูปที่ 3.24 ภายใน QV droop characteristic subsystem

3.3.7 ออกแบบระบบควบคุมแรงดันของระบบ



รูปที่ 3.25 ภายใน Voltage Control subsystem

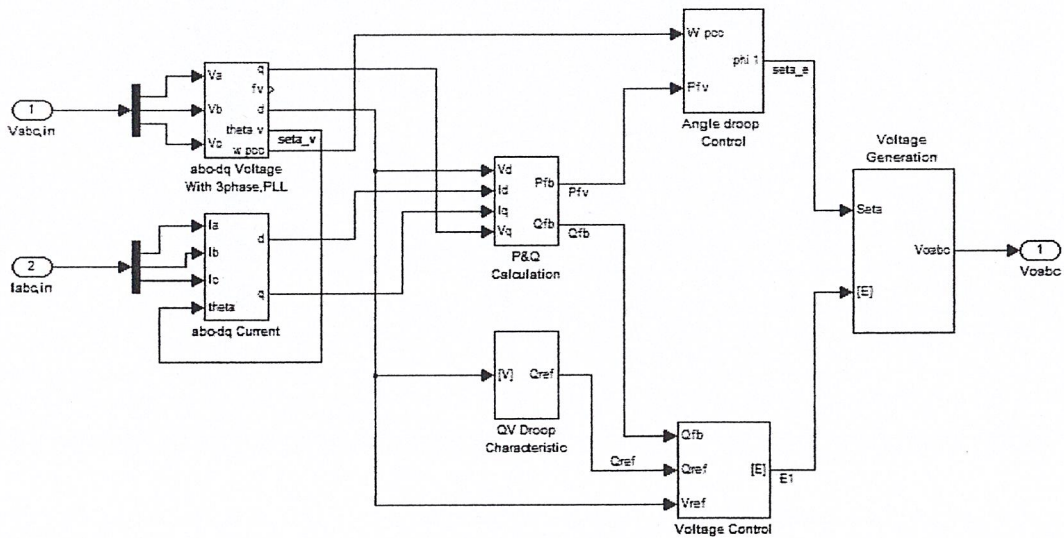
3.3.8 สร้างระบบ 3 เฟสโดยใช้เทคนิคการเลื่อนเฟสออกไป 120 องศาไฟฟ้า



รูปที่ 3.26 ภายใน Voltage Generation subsystem

3.3.9 Droop Control 1 subsystem

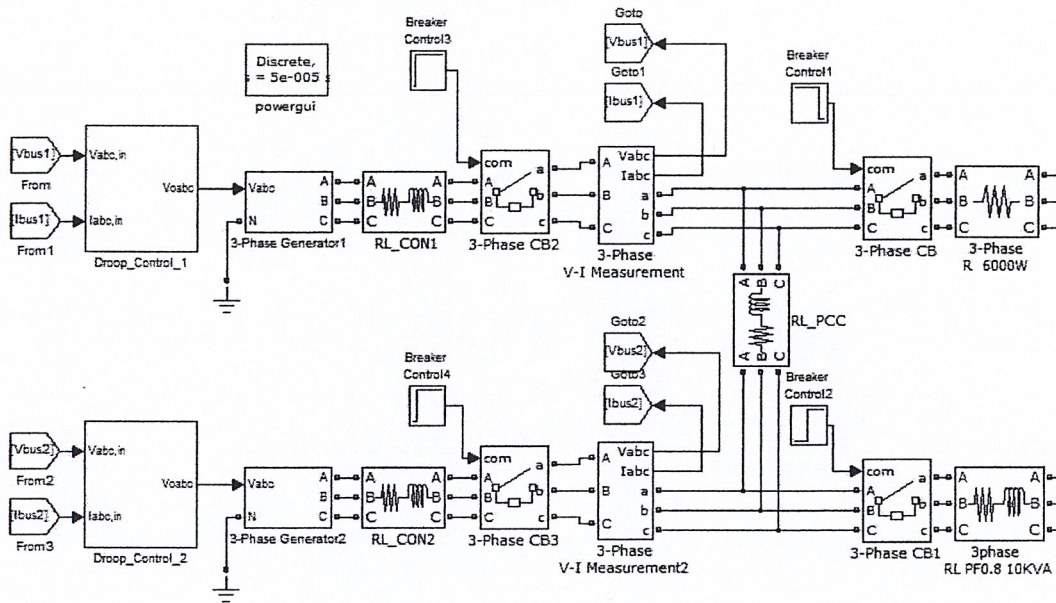
ทำการลากเส้นเชื่อมต่อกันระหว่าง subsystem ทั้งหมดภายใน converter1 ตามรูปที่ 3.32 แล้วยุบรวมเป็น subsystem อีกทีหนึ่ง โดยตั้งชื่อว่า Droop_Control_1



รูปที่ 3.27 ภายใน Droop_Control_1 subsystem

3.3.10 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 3 เฟส

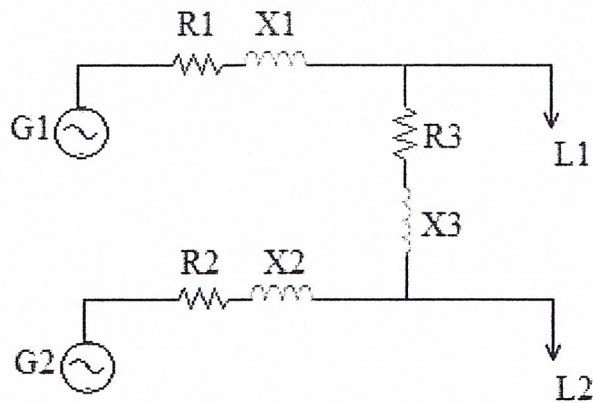
สร้างบล็อกและลากเส้นเชื่อมโยงระบบทั้งหมดตามรูปที่ 3.33



รูปที่ 3.28 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 3 เฟสที่พร้อมใช้งาน

3.4 การออกแบบการทำงานของไมโครกริดในโหมด stand-alone ของระบบ 1 เฟส

3.4.1 กำหนดรายละเอียดสำหรับไมโครกริดระบบ 1 เฟสที่จะใช้ในการทดลอง



รูปที่ 3.29 วงจรสมมูลเฟสเดียวของระบบที่ใช้ในการทดลอง

ตารางที่ 3.2 ตารางแสดงค่าพารามิเตอร์ของวงจรสมมูลที่ใช้ในการทดลอง

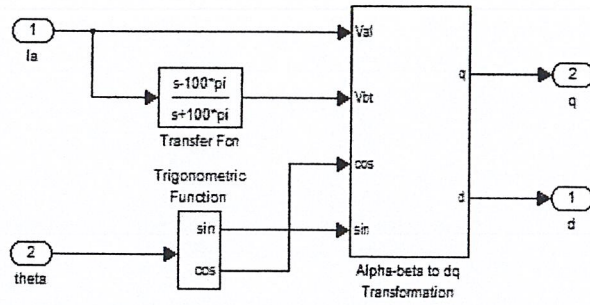
อุปกรณ์	ค่าพารามิเตอร์
G1	power rating 2kVA , Designed for pf = 0.8
G2	power rating 2kVA , Designed for pf = 0.8
R1	0.1446 Ω
X1	4.633e-3 H
R2	0.1446 Ω
X2	4.633e-3 H
R3	0.25 Ω
X3	100e-6 H
L1	40.3333 Ω
L2 (R+jXl)	Case load R Z = 121 , 60.5 , 40.3333 , 30.25 , 24.2 Ω
เปลี่ยนแปลง แล้วแต่กรณี	Case load RL at P.F.=0.9 Z = 108.9+52.7427j , 54.45+26.3713j , 36.3+17.5809j , 27.225+13.1857j , 21.78+10.5485j Ω
	Case load RL at P.F.=0.8 Z = 96.8+72.6j , 48.4+36.3j , 32.2667+24.2j , 24.2+18.15j , 19.36+14.52j Ω

3.4.2 การออกแบบ phase lock loop ชนิด 1 เฟส

ทำการสร้างการจำลองแบบเดียวกับกรณี 3 เฟสทุกประการเพียงแต่เปลี่ยนประเภทของบล็อกจาก 3 เฟส ให้เป็นแบบเฟสเดียวทั้งหมด ซึ่งได้แก่บล็อก series RLC Branch, 1-phase Breaker, 1-phase Active Reactive Power และลดจำนวนเฟสให้เหลือหนึ่งใน Subsystem ดังต่อไปนี้ คือ Voltage Generation, 3-Phase Generator 1 และ 2

3.4.3 การใช้ phase shifter ในการแปลงแกนระบบ 1 เฟส

ในส่วนของ subsystem ทั้ง abc-dq Voltage with PLL และ abc-dq Current ในการเปลี่ยนกรอบอ้างอิงเฟสเดียวเป็น alpha และ beta ให้ใช้บล็อก transfer function เป็น phase shifter โดยตั้งค่า numerator coefficients = [1 -100*pi] และ Denominator coefficients = [1 100*pi]

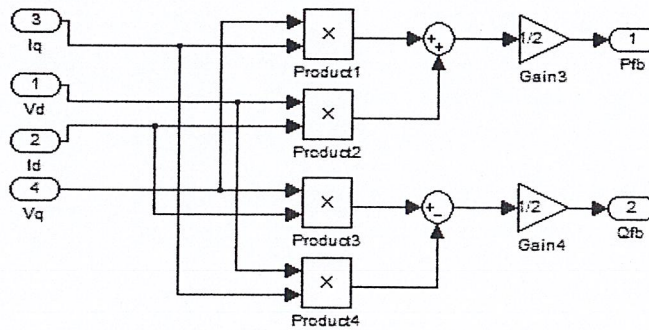


รูปที่ 3.30 การใช้ phase shifter ในการแปลงแกนระบบ 1 เฟส

3.4.4 การออกแบบ subsystem ที่ชื่อว่า P&Q calculation

ในส่วนของ subsystem P&Q Calculation ให้ทำการเปลี่ยนค่า gain ตามรูปที่

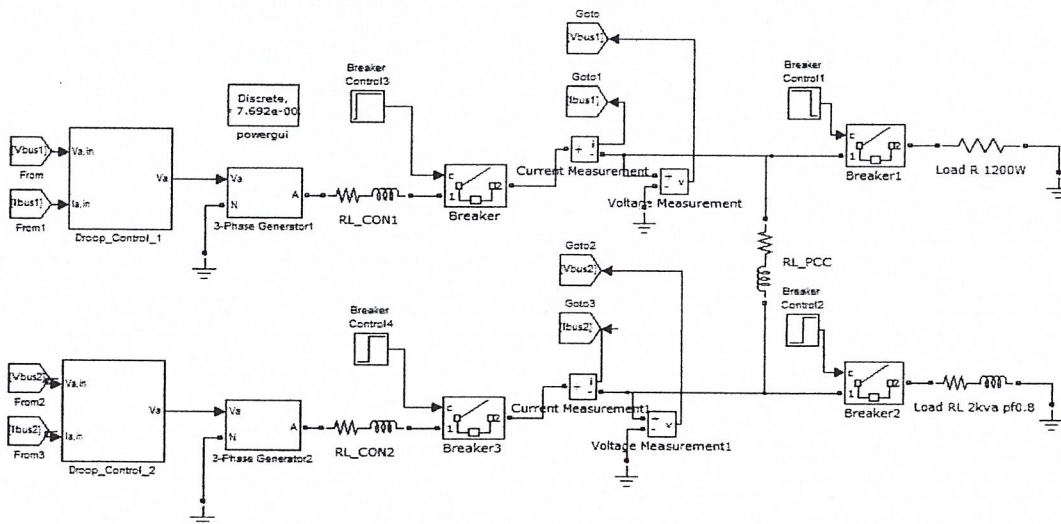
3.36



รูปที่ 3.31 ภายใน P&Q Calculation subsystem

3.4.5 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 1 เฟส

ทำการจูนค่า Kp และ Ki ของตัวแปรทุกตัวเพื่อให้ได้สมรรถนะของระบบตามที่ต้องการ เป็นอันว่าระบบเสถียรสมบูรณ์ โดยหน้าตาของระบบที่สมบูรณ์แบบแล้วจะแสดงในรูปที่ 3.37



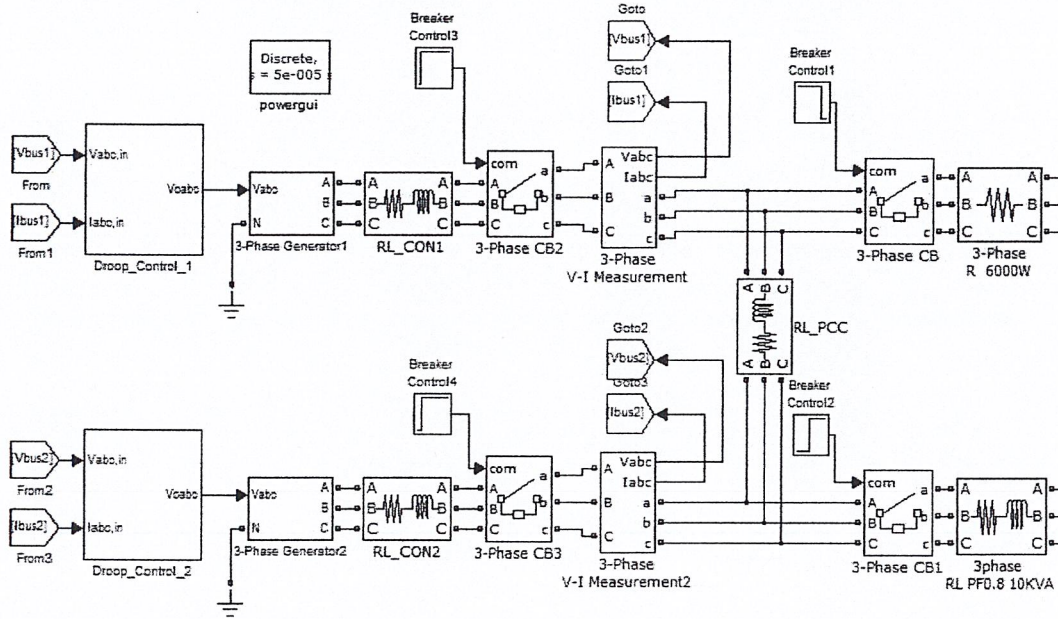
รูปที่ 3.32 แบบจำลองไมโครกริดในโหมด stand-alone ของระบบ 1 เฟสที่พร้อมใช้งาน

บทที่ 4

การทดลองและผลการทดลอง

4.1แบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ของระบบ 3 เฟส

เราจะใช้โปรแกรม Matlab/Simulink โดยจะจำลองการทำงานของระบบเป็นโหมด stand-alone ที่มีคอนเวอร์เตอร์สองตัวต่อเข้ากับจุดเชื่อมต่อ PCC ที่ต่างกัน ทั้งนี้คอนเวอร์เตอร์ทั้งสองมีอัตราส่วน droop coefficient ratio เป็น 1:1 โดยจำลองได้เป็นบล็อกไดอะแกรมดังรูปที่ 4.1



รูปที่ 4.1 แบบจำลองไมโครกริด 3 เฟสในโหมด stand-alone

ทั้งนี้เราจะมีค่าพารามิเตอร์ดังต่อไปนี้

4.1 พารามิเตอร์ของคอนเวอร์เตอร์

Summary	Converter 1	Converter 2
	Maximum power rated 10kva	Maximum power rated 10kva
Basic Setting	$P_{max} = 8kW$ at $P.F. = 0.8$ $Q_{max} = 6kVAR$ $\omega_s = 50 * 2\pi$ rad / s $Z_{con_1} = 0.125 + j3.2e-3$	$P_{max} = 8kW$ at $P.F. = 0.8$ $Q_{max} = 6kVAR$ $\omega_s = 50 * 2\pi$ rad / s $Z_{con_1} = 0.25 + j6.4e-3$
Phase-Locked Loop parameters	$Kp_PLL=1$ $Ki_PLL=10$	$Kp_PLL=1$ $Ki_PLL=10$
phase angle droop parameters	$mpf1=0.0001875*2*pi$ $Ki_p1=0.15$ $Tp1_on_time=0.1$	$mpf2=0.0001875*2*pi$ $Ki_p2=0.15$ $Tp2_on_time=0.4$

Voltage Droop parameters	mvq1=1/0.002592725; Kp1=0.01 Ki1=0.05e-4 Tq1_on_time=0.1	mvq2=1/0.002592725 Kp2=0.01 Ki2=0.05e-4 Tq2_on_time=0.4
--------------------------	---	--

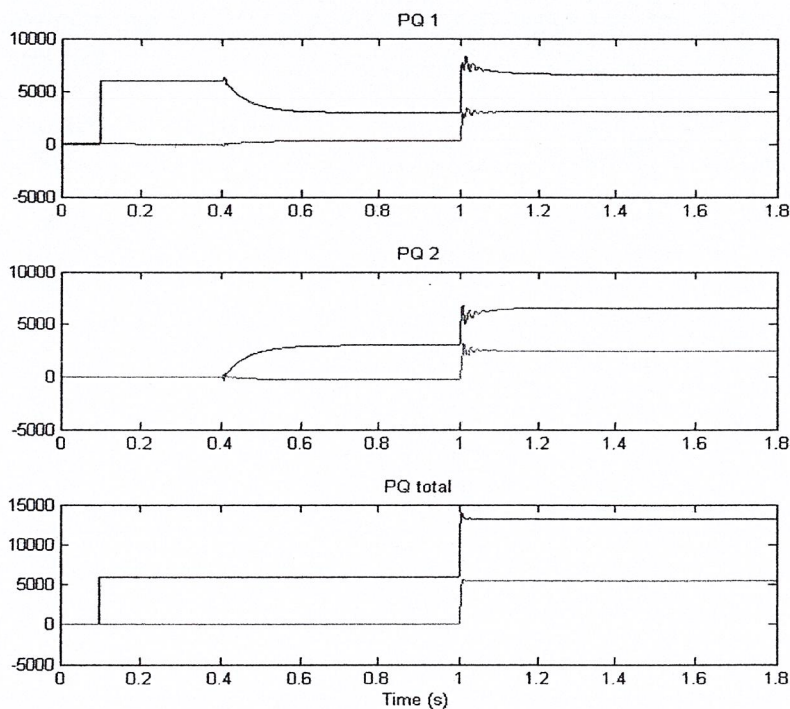
*note Feeder Impedance, $Z_{feeder} = 0.25 + j100e-6$

ตารางที่ 4.2 การตั้งค่าการเปลี่ยนแปลงทางด้านโหลดเพื่อพิจารณาผลตอบสนอง

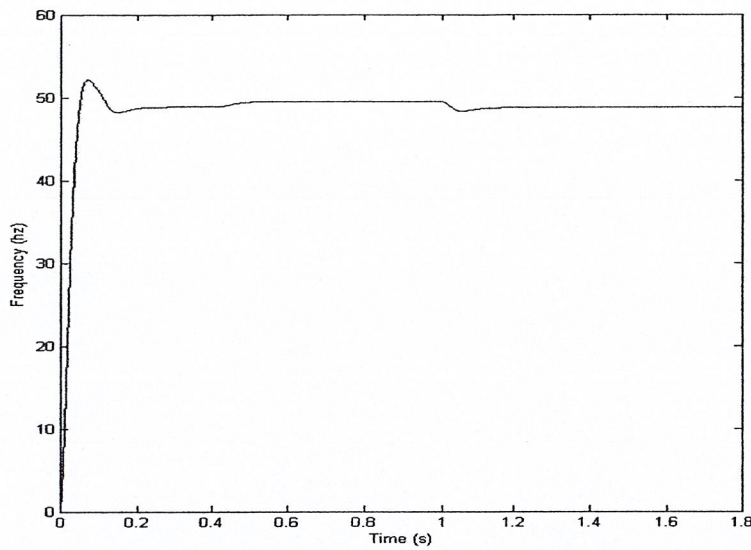
Load timing	Load description:
T1 = 1	T = 0 - 1s Load R = 6kW
T2 = 1.5	T = 1 - 1.5s Load R = 6kW + Load RL (ค่าต่างๆ)
T3 = 2	T > 1.5s Load RL (ค่าต่างๆ)

4.2 ผลการทดลองแบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 3 เฟส

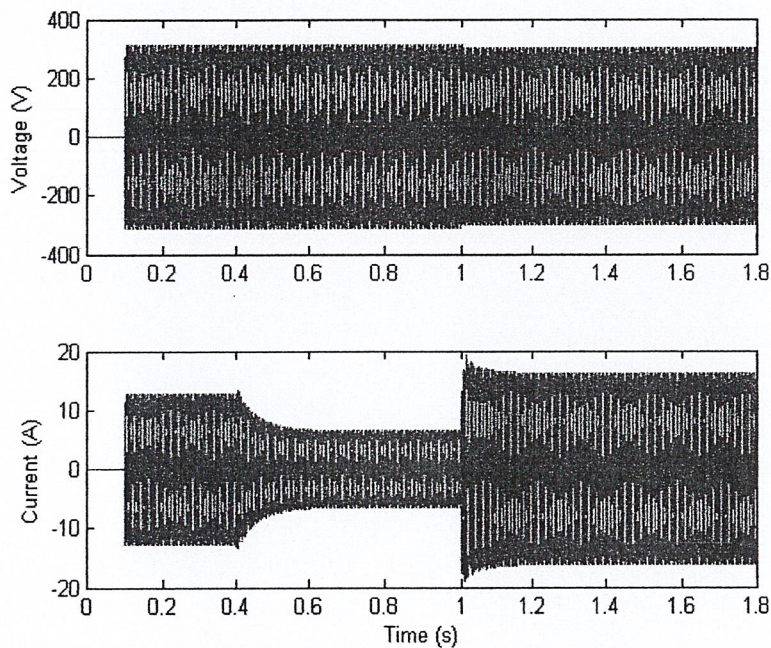
ต่อจากนี้จะเป็นรูปกราฟผลการทดลอง ซึ่งจะแสดงเฉพาะผลลัพธ์การจำลองที่กรณีเริ่มต้นด้วย load R 6kW ต่อมาเราจะเพิ่มโหลด RL 2kva at pf=0.8 ที่เวลา 1.5s และปลดโหลด R ออกที่เวลา 2s คงเหลือเพียงโหลดชนิด RL เพียงอย่างเดียว ตามรูปการแสดงผลที่ 4.2 - 4.4



รูปที่ 4.2 กำลังไฟฟ้าจริงและเสมือนของคอนเวอร์เตอร์ 1, 2 และกำลังไฟฟารวมตามลำดับ



รูปที่ 4.3 ความถี่ที่จุดต่อร่วม (Point of common,PCC)



รูปที่ 4.4 แรงดันและกระแสที่จุดต่อร่วม PCC

จากผลการทดลองเราจะเริ่มทำการตีความทางด้านกำลังไฟฟ้าก่อน โดยตัวแปรที่สนใจในหัวข้อนี้คือกำลังไฟฟ้าจริง (real power) และกำลังไฟฟารีแอกทีฟ (reactive power) โดยวินาทีที่ 0.1 - 0.4 เป็นช่วงเวลาที่เราให้คอนเวอร์เตอร์ 1 จ่ายโหลดความต้านทานเพียงอย่างเดียว พบว่าคอนเวอร์เตอร์ 1 สามารถจ่ายกำลังไฟฟ้าได้ที่ 6,000W ต่อมาเราให้คอนเวอร์เตอร์ 2 ช่วยจ่ายที่วินาที 0.4 พบว่าคอนเวอร์เตอร์ 1 และ 2 ช่วยกันจ่ายกำลังไฟฟ้าจริงคนละ 3,000W ตามอัตราส่วน

droop coefficient ที่ 1:1 โดยใช้เวลาเข้าสู่สภาวะคงตัวโดยใช้เวลาประมาณ 0.3s ต่อมาเราเพิ่มโหลด RL 10kva ที่ pf=0.8 ($P=8,000W$ $Q=6,000VAR$) แล้วพบว่ากำลังไฟฟ้าที่คอนเวอร์เตอร์ทั้งสองสามารถจ่ายรวมกันมีค่า 13,200W , 5,540 VAR ตามลำดับตามรูปที่ 4.2 ซึ่งถือว่าจ่ายได้ใกล้เคียงกับค่าที่ออกแบบไว้

ต่อมาเราจะวิเคราะห์ผลตอบสนองทางด้านความถี่จากรูปที่ 4.3 พบว่าการตอบสนองชั่วคราวในขณะที่เปลี่ยนแปลงโหลดมีค่าความถี่แกว่งไม่มากนัก และความถี่ของระบบในช่วงสภาวะคงตัวอยู่ในย่านที่ออกแบบไว้คือระหว่าง 48.5 ถึง 50 Hz

สุดท้ายเราจะมาดูผลตอบสนองทางด้านแรงดันและกระแสของจุดPCC เราจะพบว่าค่าแรงดันและกระแสที่มีรูปคลื่นไซน์ มีเสถียรภาพ และตอบสนองกับการเปลี่ยนแปลงของโหลดได้ดี

4.2.1 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =1

การเพิ่มโหลดจากโหลดความต้านทาน 6 kW กับโหลดชนิดต่างๆที่เวลา 1.3s

ตารางที่ 4.3 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 6 kW)

Load R 6kW	P_{steady}	Q_{steady}	f	V_{peak}
Converter 1	2,987	270	49.45	310.1
Converter 2	2,987	-270	49.45	311.6

ตารางที่ 4.4 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 6 kW + Load R ที่ค่าต่างๆ)

Converter 1	P_{steady}	Q_{steady}	f	V_{peak}
Load R 6kW+ RL 2kW	3,985	-200	49.25	310.2
Load R 6kW+ RL 4kW	4,980	-150	49.07	310.0
Load R 6kW+ RL 6kW	5,970	-90	48.88	310.4
Load R 6kW+ RL 8kW	6,957	-32.5	48.70	310.4
Load R 6kW+ RL 10kW	7,939	-22.5	48.50	310.5
Converter 2				
Load R 6kW+ RL 2kW	3,985	200	49.25	311.2
Load R 6kW+ RL 4kW	4,980	150	49.07	310.8
Load R 6kW+ RL 6kW	5,970	90	48.88	310.4
Load R 6kW+ RL 8kW	6,957	32.5	48.70	310.0
Load R 6kW+ RL 10kW	7,939	22.5	48.50	310.5

จากตารางที่ 4.4 จะแสดงให้เห็นว่าคอนเวอร์เตอร์ 3 เฟสที่ออกแบบนี้ให้ผลตอบสนองต่อโหลดความต้านทานที่ $pf=1$ เป็นไปตามที่ออกแบบไว้

ตารางที่ 4.5 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์

	กำลังไฟฟ้าในช่วงแรก		กำลังไฟฟ้าในช่วงที่สอง	
	P_{total}	Q_{total}	P_{total}	Q_{total}
Load R 6kW+ RL 2kW	5,974	0	7,970	0
Load R 6kW+ RL 4kW	5,974	0	9,960	0
Load R 6kW+ RL 6kW	5,974	0	11,940	0
Load R 6kW+ RL 8kW	5,974	0	13,914	0
Load R 6kW+ RL 10kW	5,974	0	15,874	0

4.2.2 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =0.9

การเพิ่มโหลดจากโหลดความต้านทาน6kWกับโหลดชนิดต่างๆที่เวลา 1.3s

ตารางที่ 4.6 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 6kW)

Load R 6kW	P_{steady}	Q_{steady}	f	V_{peak}
Converter 1	2,987	270	49.45	310.0
Converter 2	2,987	-270	49.45	311.6

ตารางที่ 4.7 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 6kW + Load R ที่ค่าต่างๆ)

Converter 1	P_{steady}	Q_{steady}	f	V_{peak}
Load R 6kW+ RL 2kVA	3,850	660	49.28	308.7
Load R 6kW+ RL 4kVA	4,700	1,060	49.12	307.5
Load R 6kW+ RL 6kVA	5,540	1,500	48.96	306.0
Load R 6kW+ RL 8kVA	6,350	1,800	48.81	305.0
Load R 6kW+ RL 10kVA	7,170	2,100	48.66	304.0
Converter 2				
Load R 6kW+ RL 2kVA	3,850	200	49.28	309.8
Load R 6kW+ RL 4kVA	4,700	630	49.12	308.0
Load R 6kW+ RL 6kVA	5,540	1,000	48.96	306.0

Load R 6kW+ RL 8kVA	6,350	1,500	48.81	305.0
Load R 6kW+ RL 10kVA	7,170	1,900	48.66	302.8

จากตารางที่ 4.7 จะแสดงให้เห็นว่าคอนเวอร์เตอร์ 3 เฟสที่ออกแบบนี้ ให้ผลตอบสนองต่อโหลดประเภท RL ที่ $pf=0.9$ เป็นไปตามที่ออกแบบไว้

ตารางที่ 4.8 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์

	กำลังไฟฟ้าในช่วงแรก		กำลังไฟฟ้าในช่วงที่สอง	
	P_{total}	Q_{total}	P_{total}	Q_{total}
Load R 6kW+ RL 2kVA	5,974	0	7,700	860
Load R 6kW+ RL 4kVA	5,974	0	9,400	1,690
Load R 6kW+ RL 6kVA	5,974	0	11,080	2,500
Load R 6kW+ RL 8kVA	5,974	0	12,700	3,300
Load R 6kW+ RL 10kVA	5,974	0	14,340	4,000

4.2.3 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F = 0.8

การเพิ่มโหลดจากโหลดความต้านทาน 6kW กับโหลดชนิดต่างๆที่เวลา 1.3s

ตารางที่ 4.9 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 6kW)

Load R 6kW	P_{steady}	Q_{steady}	<i>frequency</i>	V_{peak}
Converter 1	2,987	270	49.44	310.0
Converter 2	2,987	-270	49.44	311.6

ตารางที่ 4.10 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R6,000W+Load Rที่ค่าต่างๆ)

Converter 1	P_{steady}	Q_{steady}	<i>frequency</i>	V_{peak}
Load R 6kW+ RL 2kVA	3,750	850	49.30	308.0
Load R 6kW+ RL 4kVA	4,500	1,400	49.16	306.4
Load R 6kW+ RL 6kVA	5,200	1,950	49.00	305.0
Load R 6kW+ RL 8kVA	5,920	2,500	48.89	302.8
Load R 6kW+ RL 10kVA	6,600	3,090	48.76	300.8
Converter 2				

Load R 6kW+ RL 2kVA	3,750	330	49.30	309.3
Load R 6kW+ RL 4kVA	4,500	920	49.16	307.0
Load R 6kW+ RL 6kVA	5,200	1,480	49.00	305.0
Load R 6kW+ RL 8kVA	5,920	2,000	48.89	302.5
Load R 6kW+ RL 10kVA	6,600	2,450	48.76	300.0

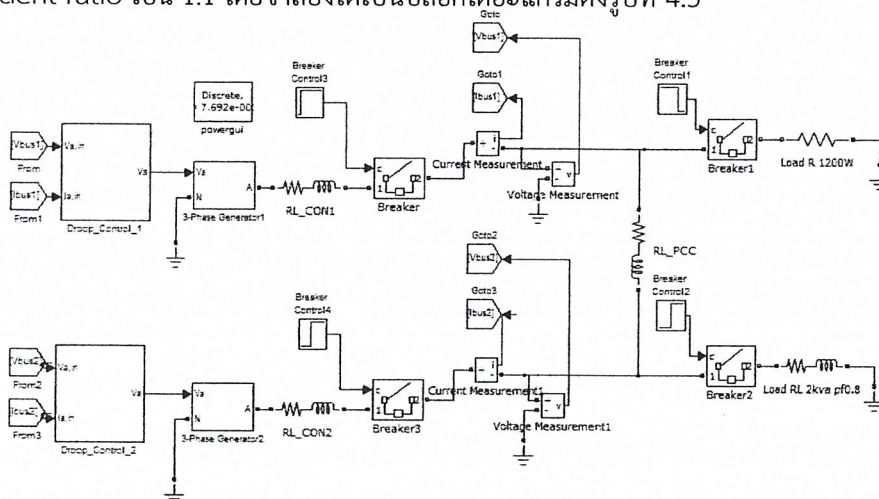
จากตารางที่ 4.10 จะแสดงให้เห็นว่าคอนเวอร์เตอร์ 3 เฟสที่ออกแบบนี้ ให้ผลตอบสนองต่อโหลดประเภท RL ที่ pf=0.8 เป็นไปตามที่ออกแบบไว้

ตารางที่ 4.11 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์

	กำลังไฟฟ้าในช่วงแรก		กำลังไฟฟ้าในช่วงที่สอง	
	P_{total}	Q_{total}	P_{total}	Q_{total}
Load R 6kW+ RL 2kVA	5,974	0	7,500	1,180
Load R 6kW+ RL 4kVA	5,974	0	9,000	2,320
Load R 6kW+ RL 6kVA	5,974	0	10,400	3,430
Load R 6kW+ RL 8kVA	5,974	0	11,840	4,500
Load R 6kW+ RL 10kVA	5,974	0	13,200	5,540

4.3 แบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 1 เฟส

เราจะทำการจำลองการทำงานของระบบเป็นโหมด stand-alone ที่มีคอนเวอร์เตอร์สองตัวต่อเข้ากับจุดเชื่อมต่อ PCC ที่ต่างกัน ทั้งนี้คอนเวอร์เตอร์ทั้งสองมีอัตราส่วน droop coefficient ratio เป็น 1:1 โดยจำลองได้เป็นบล็อกไดอะแกรมดังรูปที่ 4.5



รูปที่ 4.5 แบบจำลองไมโครกริด 1 เฟส ในโหมด stand-alone

ทั้งนี้เราจะมีค่าพารามิเตอร์ดังต่อไปนี้
 ตารางที่ 4.12 พารามิเตอร์ของคอนเวอร์เตอร์

Summary	Converter 1 Maximum power rated 2kva	Converter 2 Maximum power rated 2kva
Basic Setting	$P_{\max} = 1.6kW$ at $P.F. = 0.8$ $Q_{\max} = 1.2kVAR$ $\omega_s = 50 * 2\pi$ rad / s $Z_{con_1} = 0.1446 + j4.633e-3$	$P_{\max} = 1.6kW$ at $P.F. = 0.8$ $Q_{\max} = 1.2kVAR$ $\omega_s = 50 * 2\pi$ rad / s $Z_{con_2} = 0.1446 + j4.633e-3$
Phase-Locked Loop parameters	Kp_PLL=4 Ki_PLL=70	Kp_PLL=4 Ki_PLL=70
phase angle droop parameters	mpf1=0.0009375*2*pi Ki_p1=0.2 Tp1_on_time=0.1	mpf2=0.0009375*2*pi Ki_p2=0.2 Tp2_on_time=0.5
Voltage Droop parameters	mvq1=1/0.01296 Kp1=0.0005 Ki1=1e-4 Tq1_on_time=0.1	mvq2=1/0.01296 Kp2=0.0005 Ki2=1e-4 Tq2_on_time=0.5

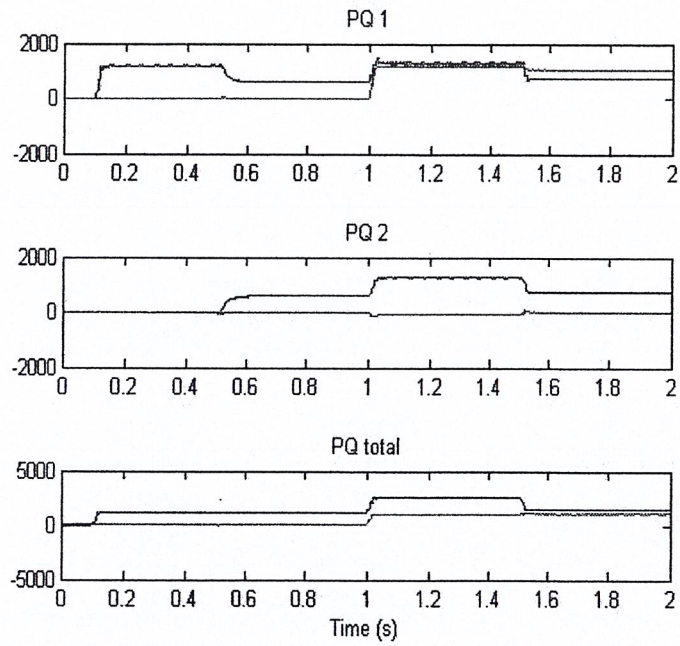
*note Feeder Impedance, $Z_{feeder} = 0.25 + j100e-6$

ตารางที่ 4.13 การตั้งค่าการเปลี่ยนแปลงทางด้านโหลดเพื่อพิจารณาผลตอบสนอง

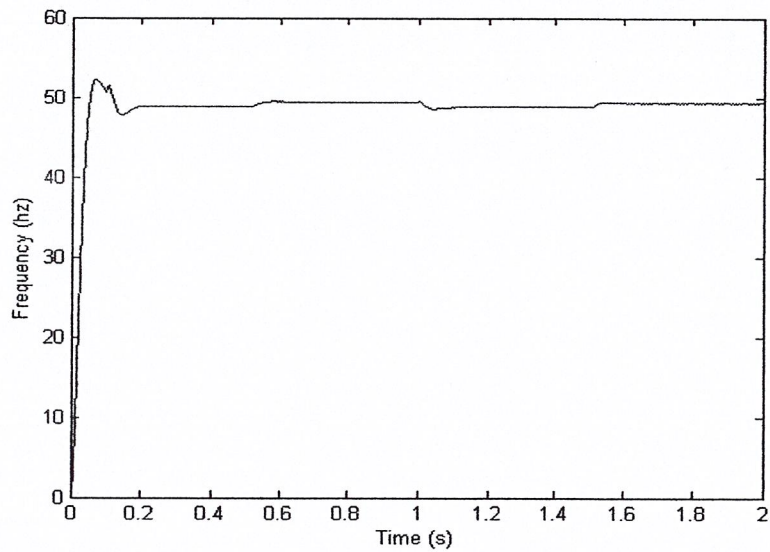
Load timing	Load description:
T1 = 1	T = 0 - 1s Load R = 1,200W
T2 = 1.5	T = 1 - 1.5s Load R = 1,200w + Load RL (ค่าต่างๆ)
T3 = 2	T > 1.5s Load RL (ค่าต่างๆ)

4.4 ผลการทดลองแบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 1 เฟส

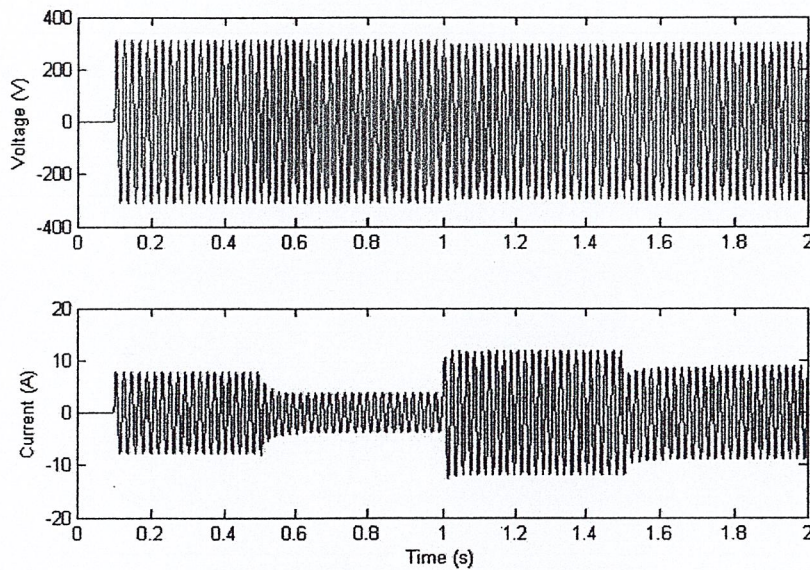
ต่อจากนี้จะเป็นรูปกราฟผลการทดลอง ซึ่งจะแสดงเฉพาะผลลัพธ์การจำลองที่กรณีเริ่มต้นด้วย load R 1.2kW ต่อมาเพิ่มโหลด RL ที่เวลา1.5s และปลดโหลด R ออกที่เวลา2s คงเหลือเพียงโหลดชนิด RL เพียงอย่างเดียว ตามกราฟแสดงผลที่ 4.6 - 4.8



รูปที่ 4.6 กำลังไฟฟ้าจริงและเสมือนของคอนเวอร์เตอร์ 1, 2 และกำลังไฟฟารวมตามลำดับ



รูปที่ 4.7 ความถี่ที่จุดต่อร่วม (Point of common,PCC)



รูปที่ 4.8 แรงดันและกระแสที่จุดต่อร่วม PCC

จากผลการทดลองเราจะเริ่มทำการตีความทางด้านกำลังไฟฟ้าก่อน โดยตัวแปรที่สนใจในหัวข้อนี้คือกำลังไฟฟ้าจริง (real power) และกำลังไฟฟารีแอกทีฟ (reactive power) โดยวินาทีที่ 0.1 - 0.5 เป็นช่วงเวลาที่เราให้คอนเวอร์เตอร์ 1 จ่ายโหลดความต้านทานเพียงอย่างเดียว พบว่าคอนเวอร์เตอร์ 1 สามารถจ่ายกำลังไฟฟ้าได้ที่ 1,180W ต่อมาเราให้คอนเวอร์เตอร์ 2 ช่วยจ่ายที่วินาที 0.5 พบว่าคอนเวอร์เตอร์ 1 และ 2 ช่วยกันจ่ายกำลังไฟฟ้าจริงคนละ 595W ตามอัตราส่วน droop coefficient ที่ 1:1 โดยใช้เวลาเข้าสู่สภาวะคงตัวโดยใช้เวลาประมาณ 0.1s ต่อมาเราเพิ่มโหลด RL 2kva ที่ $pf=0.8$ ($P=1,600W$ $Q=1,200VAR$) แล้วพบว่ากำลังไฟฟ้าที่คอนเวอร์เตอร์ทั้งสองสามารถจ่ายรวมกันมีค่า 2,580W , 1,080 VAR ตามลำดับตามรูปที่ 4.2 ซึ่งถือว่าจ่ายได้ใกล้เคียงกับค่าที่ออกแบบไว้

ต่อมาเราจะวิเคราะห์ผลตอบสนองทางด้านความถี่จากรูปที่ 4.3 พบว่าการตอบสนองชั่วคราวในขณะที่เปลี่ยนแปลงโหลดมีค่าความถี่แกว่งไม่มากนัก และความถี่ของระบบในช่วงสภาวะคงตัวอยู่ในย่านที่ออกแบบไว้คือระหว่าง 48.5 ถึง 50 hz

สุดท้ายเราจะมาดูผลตอบสนองทางด้านแรงดันและกระแสของจุด PCC เราจะพบว่าค่าแรงดันและกระแสนั้นมีรูปคลื่นไซน์ มีเสถียรภาพ และตอบสนองกับการเปลี่ยนแปลงของโหลดได้ดี

4.4.1 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =1

การเพิ่มโหลดจากโหลดความต้านทาน 1,200W กับโหลดชนิดต่างๆ

ตารางที่ 4.14 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 1,200W)

Load R 1,200W	P_{steady}	Q_{steady}	f	V_{peak}
Converter 1	595	0	49.45	310
Converter 2	595	0	49.45	311

ตารางที่ 4.15 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 1,200W + Load R ที่ค่าต่างๆ)

Converter 1	P_{steady}	Q_{steady}	f	$V_{max,min}$
Load R 1,200W + R 400W	790	31	49.26	309.5
Load R 1,200W + R 800W	983	60	49.08	309.0
Load R 1,200W + R 1,200W	1,170	100	48.90	308.6
Load R 1,200W + R 1,600W	1,360	130	48.73	308.0
Load R 1,200W + R 2,000W	1,540	160	48.55	307.5
Converter 2				
Load R 1,200W + R 400W	790	-31	49.26	310.0
Load R 1,200W + R 800W	983	-60	49.08	309.4
Load R 1,200W + R 1,200W	1,170	-100	48.90	308.5
Load R 1,200W + R 1,600W	1,360	-130	48.73	307.8
Load R 1,200W + R 2,000W	1,540	-160	48.55	307.0

จากตารางที่ 4.15 จะแสดงให้เห็นว่าคอนเวอร์เตอร์ 1 เฟสที่ออกแบบนี้ ให้ผลตอบสนองต่อโหลดความต้านทาน ที่ $pf=1$ เป็นไปตามที่ออกแบบไว้

ตารางที่ 4.16 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์

	กำลังไฟฟ้าในช่วงแรก		กำลังไฟฟ้าในช่วงที่สอง	
	P_{total}	Q_{total}	P_{total}	Q_{total}
Load R 1,200W + R 400W	1,190	0	1,580	0
Load R 1,200W + R 800W	1,190	0	1,965	0
Load R 1,200W + R 1,200W	1,190	0	2,340	0
Load R 1,200W + R 1,600W	1,190	0	2,720	0
Load R 1,200W + R 2,000W	1,190	0	3,080	0

4.4.2 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =0.9

การเพิ่มโหลดจากโหลดความต้านทาน 1,200W กับโหลดชนิดต่างๆ

ตารางที่ 4.17 ผลช่วงแรกในขณะที่ต้องโหลดความต้านทานเพียงอย่างเดียว (Load R 1,200W)

Load R 1,200W	P_{steady}	Q_{steady}	f	V_{peak}
Converter 1	595	0	49.45	310
Converter 2	595	0	49.45	311

ตารางที่ 4.18 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 1,200W + Load RL)

Converter 1	P_{steady}	Q_{steady}	f	V_{peak}
Load R 1,200W + RL 400VA	765	192	49.28	308.0
Load R 1,200W + RL 800VA	930	380	49.13	306.0
Load R 1,200W + RL 1,200VA	1,090	555	49.00	304.0
Load R 1,200W + RL 1,600VA	1,240	730	48.84	302.5
Load R 1,200W + RL 2,000VA	1,400	900	48.7	300.5
Converter 2				
Load R 1,200W + RL 400VA	765	-22	49.28	308.6
Load R 1,200W + RL 800VA	930	-47	49.13	306.5
Load R 1,200W + RL 1,200VA	1,090	-65	49.00	304.0
Load R 1,200W + RL 1,600VA	1,240	-87	48.84	302.0
Load R 1,200W + RL 2,000VA	1,400	-100	48.7	300.0

จากตารางที่ 4.18 จะแสดงให้เห็นว่าคอนเวอร์เตอร์ 1 เฟสที่ออกแบบนี้ ให้ผลตอบสนองต่อโหลดประเภท RL ที่ pf=0.9 เป็นไปตามที่ออกแบบไว้

ตารางที่ 4.19 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์

	กำลังไฟฟ้าในช่วงแรก		กำลังไฟฟ้าในช่วงที่สอง	
	P_{total}	Q_{total}	P_{total}	Q_{total}
Load R 1,200W + RL 400VA	1,190	0	1,530	170
Load R 1,200W + RL 800VA	1,190	0	1,860	333
Load R 1,200W + RL 1,200VA	1,190	0	2,180	490

Load R 1,200W + RL 1,600VA	1,190	0	2,480	613
Load R 1,200W + RL 2,000VA	1,190	0	2,800	800

4.4.3 ผลการทดลองที่โหลดที่เปลี่ยนแปลงเป็นโหลดมีค่า P.F =0.8

การเพิ่มโหลดจากโหลดความต้านทาน 1,200W กับโหลดชนิดต่างๆ

ตารางที่ 4.20 ผลช่วงแรกในขณะที่ต่อโหลดความต้านทานเพียงอย่างเดียว (Load R 1,200W)

Load R 1,200 W	P_{steady}	Q_{steady}	f	V_{peak}
Converter 1	595	0	49.44	310
Converter 2	595	0	49.44	310

ตารางที่ 4.21 ผลช่วงที่สองในขณะที่มีการแชร์โหลดร่วมกัน (Load R 1,200W + Load RL ที่ค่าต่างๆ)

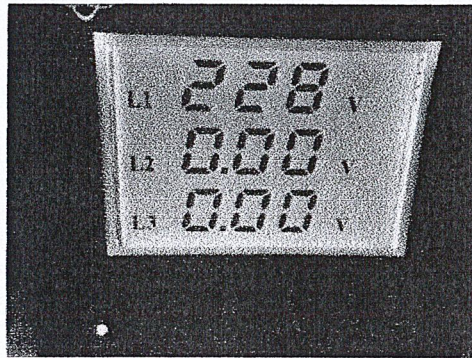
Converter 1	P_{steady}	Q_{steady}	f	V_{peak}
Load R 1,200W + RL 400VA	743	251	49.31	308.0
Load R 1,200W + RL 800VA	885	490	49.17	305.0
Load R 1,200W + RL 1,200VA	1,025	724	49.04	302.6
Load R 1,200W + RL 1,600VA	1,160	943	48.91	300.0
Load R 1,200W + RL 2,000VA	1,290	1160	48.79	298.0
Converter 2				
Load R 1,200W + RL 400VA	743	-17	49.31	308.0
Load R 1,200W + RL 800VA	885	-30	49.17	305.0
Load R 1,200W + RL 1,200VA	1,025	-50	49.04	302.6
Load R 1,200W + RL 1,600VA	1,160	-62	48.91	300.0
Load R 1,200W + RL 2,000VA	1,290	-80	48.79	297.5

จากตารางที่ 4.21 จะแสดงให้เห็นว่าคอนเวอร์เตอร์ 1 เฟสที่ออกแบบนี้ ให้ผลตอบสนองต่อโหลดประเภท RL ที่ pf=0.8 เป็นไปตามที่ออกแบบไว้

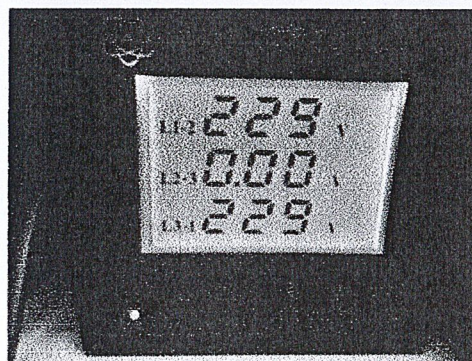
ตารางที่ 4.22 ผลรวมของกำลังไฟฟ้ารวมทั้งสองคอนเวอร์เตอร์

	กำลังไฟฟ้าในช่วงแรก		กำลังไฟฟ้าในช่วงที่สอง	
	P_{total}	Q_{total}	P_{total}	Q_{total}
Load R 1,200W + RL 400VA	1,190	0	1,486	234
Load R 1,200W + RL 800VA	1,190	0	1,770	460
Load R 1,200W + RL 1,200VA	1,190	0	2,050	674
Load R 1,200W + RL 1,600VA	1,190	0	2,320	881
Load R 1,200W + RL 2,000VA	1,190	0	2,580	1,080

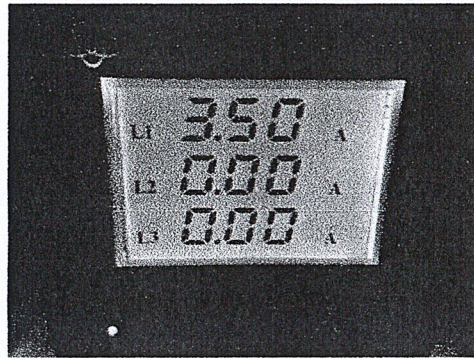
4.5 ผลการทดลองของ Modbus



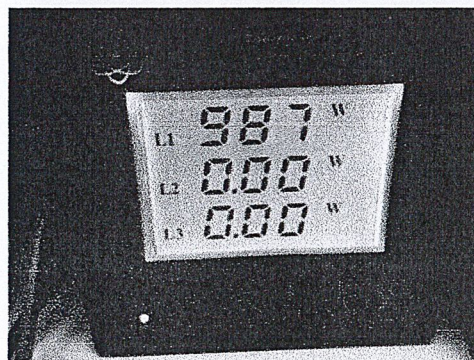
รูปที่ 4.9 Voltage line1



รูปที่ 4.10 Voltage line1-2 , line3-1



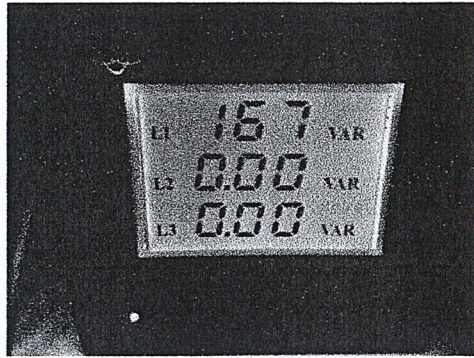
รูปที่ 4.11 Current line1



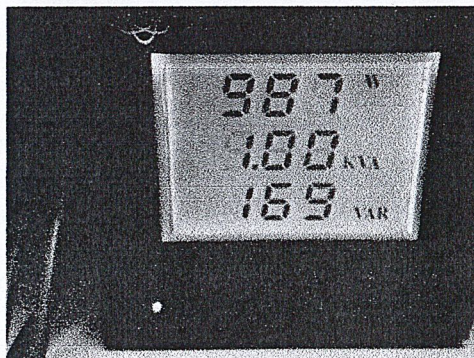
รูปที่ 4.12 Active Power line1



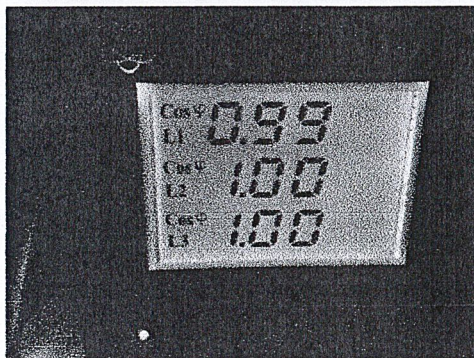
รูปที่ 4.13 Apparent Power line1



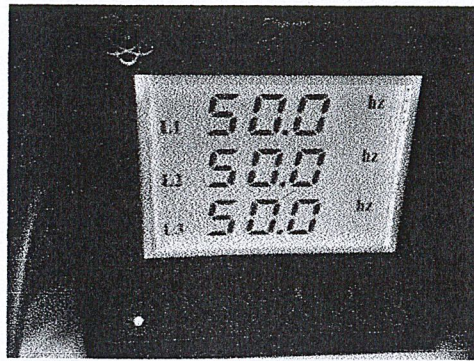
รูปที่ 4.14 Reactive Power line1



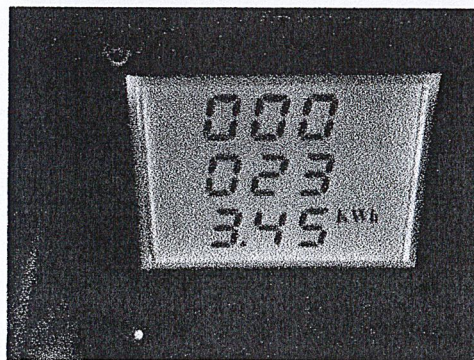
รูปที่ 4.15 ผลรวม W , KVA ,VAR



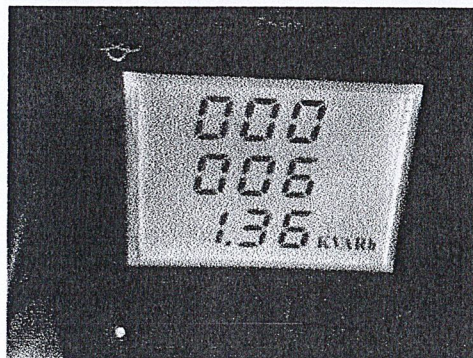
รูปที่ 4.16 Power Factor



รูปที่ 4.17 Frequency



รูปที่ 4.18 Real Energy (Wh)

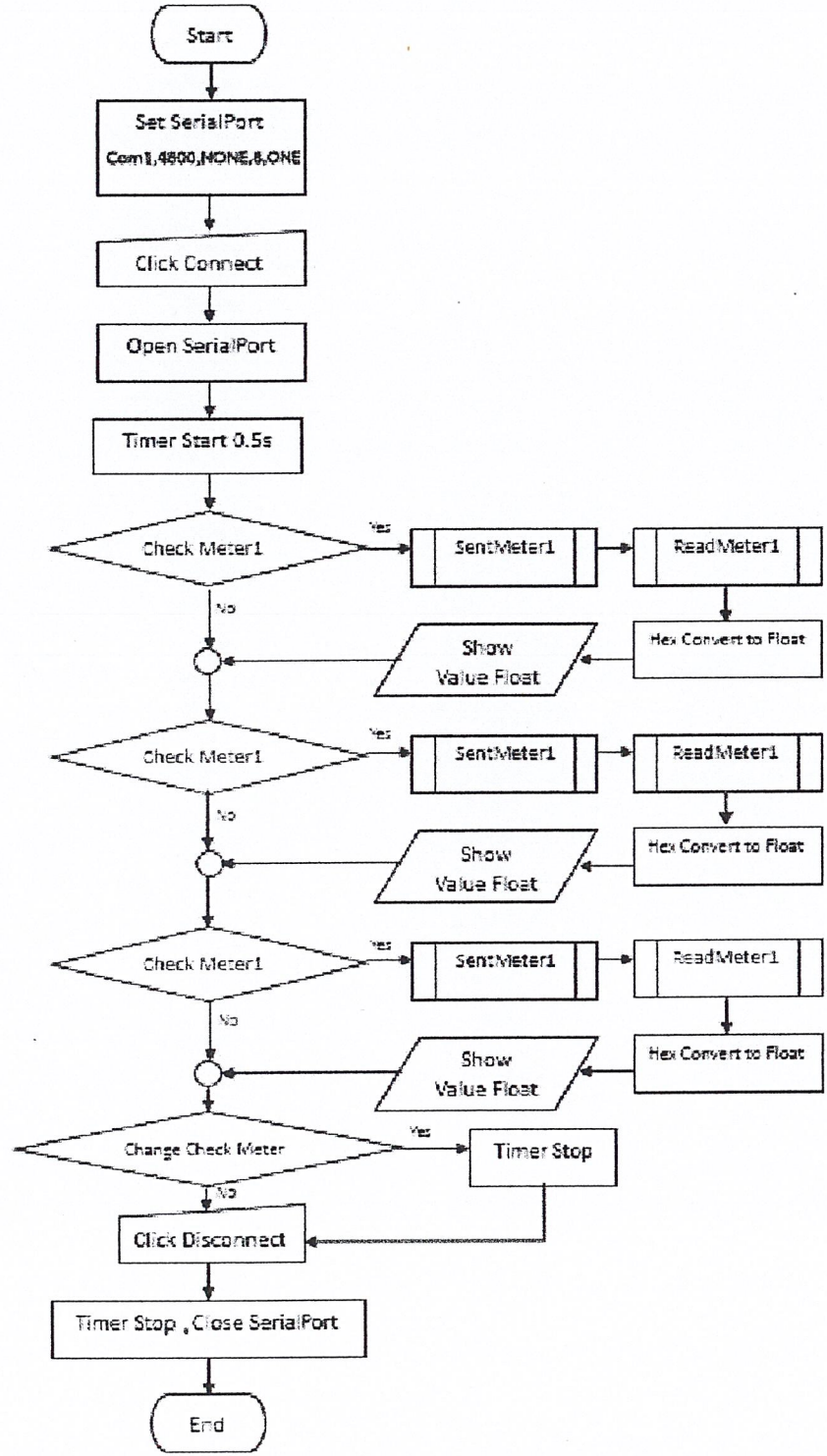


รูปที่ 4.19 Reactive Energy (VARh)

Mesurement	Line 1	Line 2	Line 3	Line 1-2	Line 2-3	Line 3-1	Total
Voltage(V)	227.794			227.794		227.794	
Current(A)	4.04						
Active Power (W)	897.088						897.088
Apparent Power(VA)	920.402						948.883
Reactive Power(VAR)	204.834						204.834
Power Factor	.976	1	1				.976
Frequency(Hz)	50.042	50.042	50.042				
Real Energy (Wh)	233434.2						233433.9
Apparent Energy (VAh)	61360.35						61360.29
Reactive Energy (VARh)	241364.1						241363.8

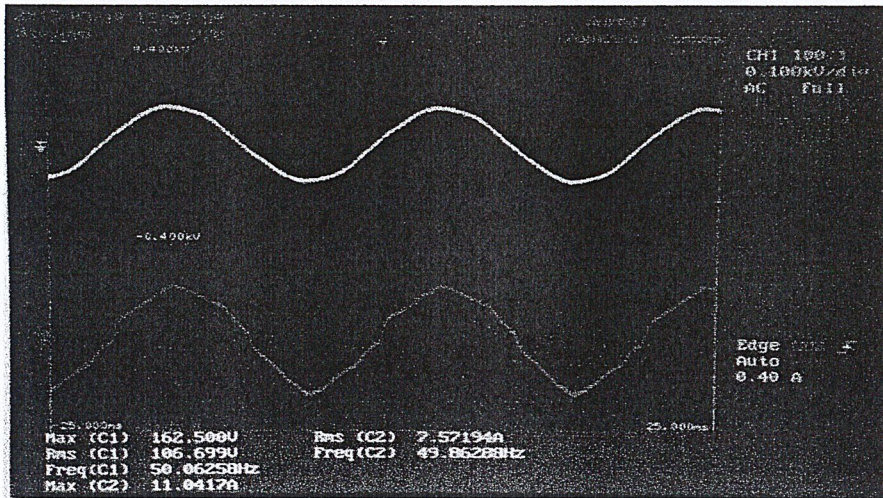
รูปที่ 4.20 หน้าจอแสดงผลของโปรแกรม

System Flowchart ของโปรแกรมแสดงผล

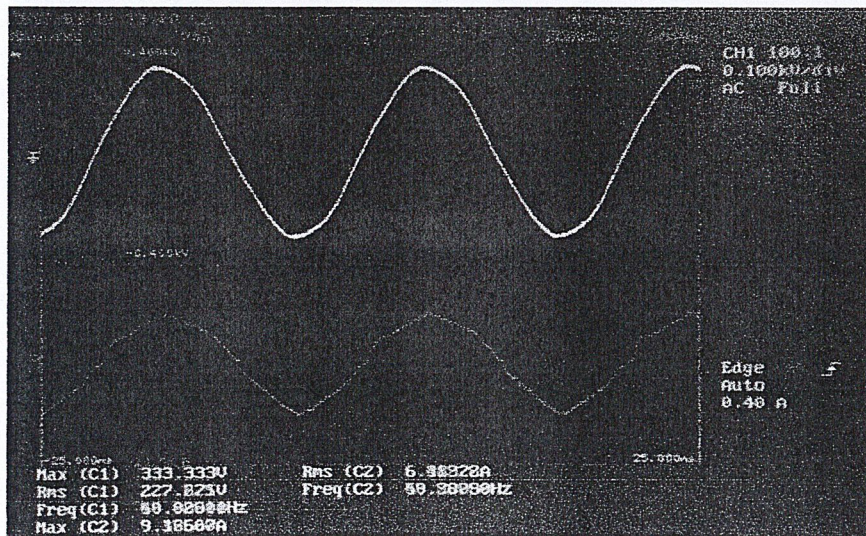


4.6 ผลการทดลองติดตั้ง Grid-Connected Photovoltaic Power Generation ขนาด 3 kW

4.6.1 รูปคลื่นแรงดันและกระแสของอินเวอร์เตอร์



รูปที่ 4.21 คลื่นแรงดันและกระแสของอินเวอร์เตอร์ขณะก่อนต่อเข้าหม้อแปลง



รูปที่ 4.22 คลื่นแรงดันและกระแสของอินเวอร์เตอร์ขณะหลังต่อออกจากหม้อแปลง

4.6.2 ผลการจ่ายพลังงานไฟฟ้ารายวัน/เดือน

ตารางที่ 4.23 ค่าพลังงานไฟฟ้า (kWh) ที่ได้รับระหว่าง 9 มกราคม ถึง 17 กุมภาพันธ์ พ.ศ. 2556

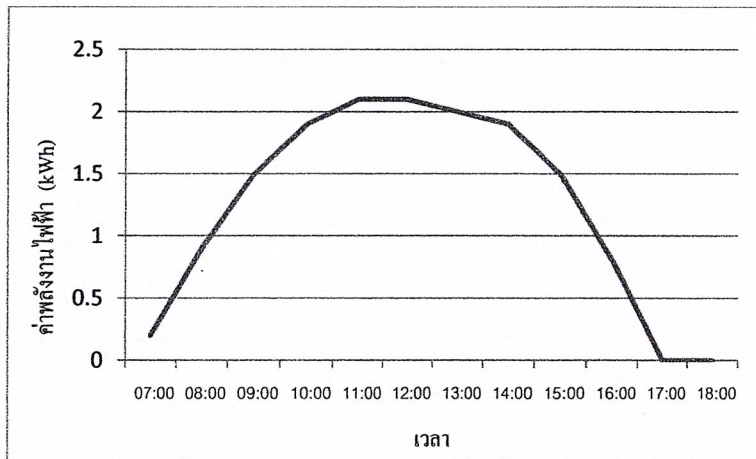
Date	9 Jan	10 Jan	11 Jan	12 Jan	13 Jan	14 Jan	15 Jan	16 Jan	17 Jan	18 Jan	19 Jan	20 Jan	21 Jan	22 Jan	23 Jan	24 Jan	25 Jan	26 Jan	27 Jan	28 Jan
07:00	0.1	0.1	0	0.1	0.1	0.2	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0	0	0	0	0	0.1	0
08:00	0.6	0.6	0.5	0.6	0.8	0.9	0.7	0.7	0.8	0.8	0.8	0.7	0.6	0.5	0.4	0.4	0.1	0.1	0.6	0
09:00	1.2	1.1	1.0	0.3	1.5	1.5	1.3	1.4	1.5	1.6	1.4	1.4	1.2	1.1	0.9	0.8	0.5	0.8	0.8	0
10:00	1.6	1.5	1.3	0.8	2.0	1.9	1.4	1.7	1.9	2.1	1.9	1.8	1.6	1.5	1.4	1.2	1.5	1.0	1.0	0.9
11:00	1.7	1.8	1.4	2.2	1.6	2.1	1.2	2.0	2.1	2.3	2.2	2.0	1.9	1.6	1.6	1.7	1.8	1.2	1.6	0.3
12:00	1.7	1.6	1.1	1.9	2.1	2.1	1.6	2.0	2.2	2.3	2.2	2.1	1.9	1.8	1.5	1.8	1.8	1.5	1.6	1.2
13:00	1.5	1.3	0.4	2.2	1.9	2.0	1.8	1.9	2.2	2.2	2.1	1.9	1.9	1.7	1.6	1.8	1.5	1.3	1.8	1.4
14:00	1.1	1.4	1.3	1.9	1.4	1.9	1.2	1.6	1.9	1.9	1.8	1.7	1.6	1.4	1.5	1.5	1.0	0.7	1.2	0.7
15:00	0.6	1.1	0.2	1.4	0.9	1.5	0.7	1.2	1.3	1.5	1.2	1.2	1.1	1	1.1	1.1	1.1	0.9	1.2	0.8
16:00	0.3	0.6	0	0.8	0.8	0.8	0.6	1.6	0.6	0.8	0.6	0.6	0.5	0.5	0.5	0.5	0.4	0.8	0.5	0.4
17:00	0	0	0	0.1	0.1	0	0.1	0	0	0.1	0	0	0	0	0	0	0	0.1	0.1	0
18:00	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total	10.9	11.6	8.7	12.8	13.6	15.5	11.2	13.6	15.1	16.1	14.9	14.0	12.8	11.9	10.9	11.5	10.4	8.9	11.0	6.2

Date	29 Jan	30 Jan	31 Jan	1 Feb	2 Feb	3 Feb	4 Feb	5 Feb	6 Feb	7 Feb	8 Feb	9 Feb	10 Feb	11 Feb	12 Feb	13 Feb	14 Feb	15 Feb	16 Feb	17 Feb
07:00	0.1	0	0	0.1	0	0	0.1	0.1	0.1	0.1	0	0	0.1	0	0.1	0.1	0.3	0.1	0	0.1
08:00	0.6	0.5	0.4	0.6	0.3	0.2	0.6	0.6	0.6	0.5	0.5	0	0.5	0.6	0.7	0.5	0.5	0.6	0.3	0.7
09:00	1.0	0.6	1.3	1.3	0.4	0.4	1.3	1.3	1.2	1.1	1.1	0.2	1.2	1.1	1.1	1.2	1.1	1.2	0.9	1.2
10:00	1.7	1.6	1.2	1.7	0.5	0.8	1.6	1.6	1.7	1.2	1.2	1.4	1.2	1.4	1.4	1.7	1.5	1.5	1.0	1.4
11:00	2.0	1.8	1.6	1.9	0.8	1.3	2.0	1.7	2.0	1.6	1.6	0.8	1.5	1.9	1.8	1.9	1.7	1.8	1.3	1.5
12:00	1.7	1.9	1.5	2.0	0.5	1.6	2.0	1.4	2.0	1.8	1.8	1.6	1.9	2.0	2.0	1.7	1.9	1.8	1.6	1.2
13:00	1.7	1.9	1.2	1.7	0.8	1.2	1.9	1.7	1.7	1.8	1.8	1.7	2.0	1.9	1.9	1.8	1.9	1.7	1.4	1.8
14:00	1.4	1.3	0.6	1.4	1.1	1.5	1.7	1.7	1.7	1.3	1.3	1.3	1.7	1.6	1.7	1.2	1.7	1.5	1.0	1.5
15:00	0.8	0.7	0.9	1.0	1.0	1.0	1.2	1.3	1.2	1.2	1.0	0.8	1.2	1.2	1.2	1.2	1.3	1.2	0.8	1.1
16:00	0.1	0.3	0.4	0.6	0.5	0.4	0.8	0.7	0.6	0.9	0.5	0.5	0.7	0.7	0.8	0.6	0.6	0.6	0.6	0.6
17:00	0	0	0	0.1	0	0.1	0.1	0.1	0.1	0.5	0.1	0	0.1	0.1	0.1	0.1	0.2	0.1	0.1	0.1
18:00	0	0	0	0	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0
Total	11.5	11.1	9.5	12.4	5.9	8.5	13.3	12.8	13.4	10.5	11.3	8.8	12.2	13.0	12.8	12.0	12.7	12.1	9.0	11.2

จากตารางค่าพลังงานไฟฟ้า (kWh) ที่ได้จากการเก็บข้อมูลระหว่างวันที่ 9 มกราคม ถึง 17 กุมภาพันธ์ พ.ศ. 2556 เราสามารถนำค่า kWh ที่ได้ในแต่ละวันมาพล็อตกราฟ ดังนี้

ค่าพลังงานไฟฟ้า (kWh) รายวัน

ค่าพลังงานไฟฟ้าพลังงานไฟฟ้า (kWh) รวมสูงสุดวันที่ 14 มกราคม 2556 ที่ 15.5 kWh

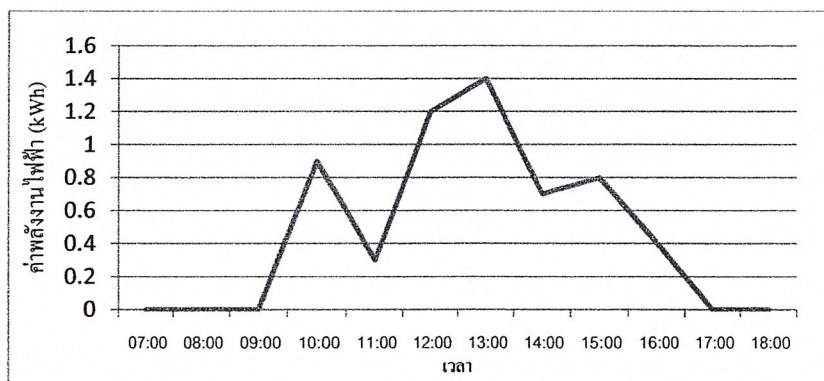


รูปที่ 4.23 กราฟค่าพลังงานไฟฟ้า (kWh) กับเวลา

จากกราฟจะเห็นว่าค่าพลังงานไฟฟ้า (kWh) ที่ได้สูงสุดคือ 2.1 kWh จะอยู่ในช่วงเวลา 11.00 -12.00 น.

หมายเหตุ กรมอุตุนิยมวิทยา รายงานลักษณะอากาศ กรุงเทพมหานครและปริมณฑล มีเมฆบางส่วน กับมีหมอกบางในตอนเช้า อุณหภูมิต่ำสุด 23 - 24 องศาเซลเซียส

ค่าพลังงานไฟฟ้าพลังงานไฟฟ้า (kWh) รวมต่ำสุดวันที่ 28 มกราคม 2556 ที่ 6.2 kWh

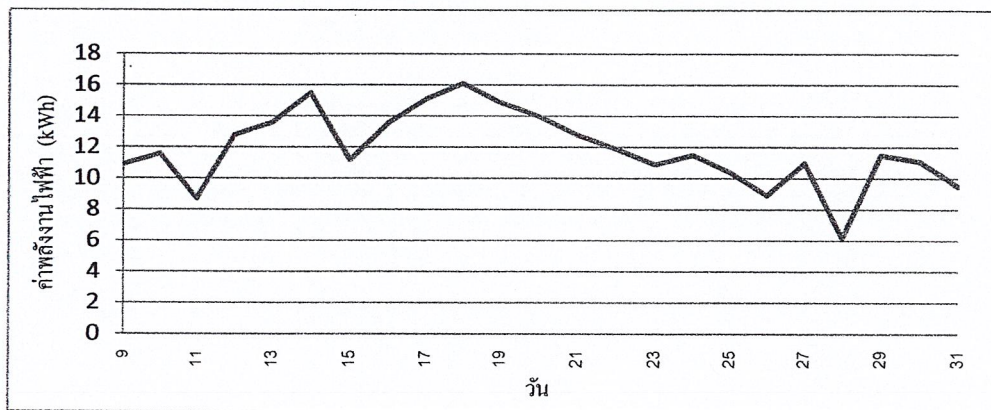


รูปที่ 4.24 กราฟค่าพลังงานไฟฟ้า (kWh) กับเวลา

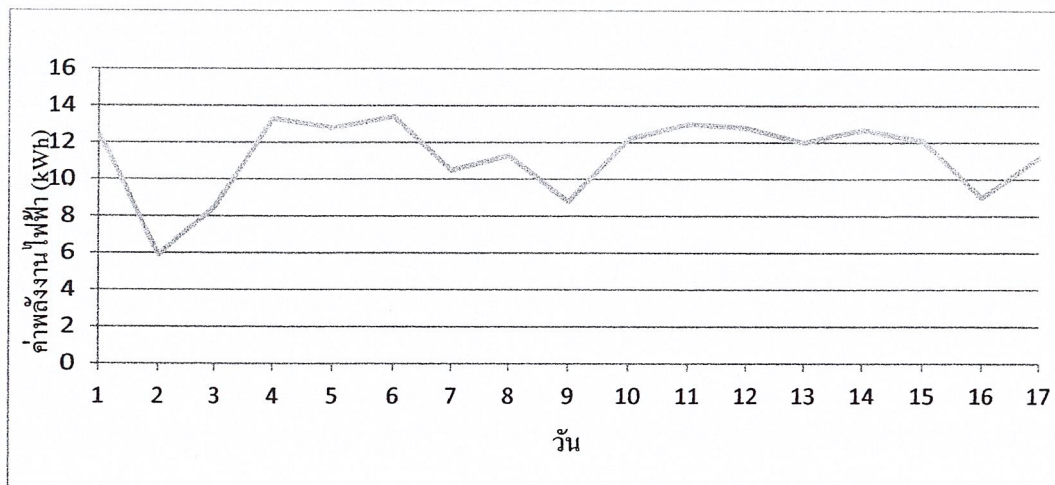
จากกราฟจะเห็นว่าค่าพลังงานไฟฟ้า (kWh) ที่ได้สูงสุดคือ 1.4 kWh จะอยู่ใน
ช่วงเวลา 12.00-13.00 น.

หมายเหตุ กรมอุตุนิยมวิทยา รายงานลักษณะอากาศ กรุงเทพมหานครและปริมณฑลมีฝนฟ้าคะนอง
กระจาย ร้อยละ 40 ของพื้นที่ อุณหภูมิต่ำสุด 24-25 องศาเซลเซียส

ค่าพลังงานไฟฟ้า (kWh) รายเดือน



รูปที่ 4.25 กราฟค่าพลังงานไฟฟ้า (kWh) ที่ได้ในแต่ละวัน ประจำเดือนมกราคม



รูปที่ 4.26 กราฟค่าพลังงานไฟฟ้า (kWh) ที่ได้ในแต่ละวัน ประจำเดือนกุมภาพันธ์

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

โครงการนี้ถูกนำมาวิจัยเพื่อศึกษาและพัฒนาระบบไมโครกริด ซึ่งเป็นแนวโน้มใหม่ ของเครือข่ายของระบบไฟฟ้า ทั้งนี้ไมโครกริดสามารถรักษาเสถียรภาพของตัวเองเพื่อสามารถจ่าย โหลดได้อย่างไม่มีปัญหาทั้งในโหมด Gird-connected และ Stand-alone ในยามที่แหล่งจ่ายการ ไฟฟ้ามีปัญหา โดยโครงการนี้จะแบ่งออกเป็นสองส่วนหลักๆ คือ การจำลองระบบไมโครกริดในโหมด stand-alone และการแสดงผลข้อมูลผ่านคอมพิวเตอร์โดยใช้โปรแกรม Visual Studio 2010

สำหรับการศึกษาการจำลองระบบไมโครกริดในโหมด stand-alone ของระบบทั้ง สามเฟสและเฟสเดียวของแหล่งจ่ายสองตัวที่มีพารามิเตอร์เดียวกัน พบว่าผลลัพธ์ที่ออกมานั้น ค่อนข้างน่าพอใจ กล่าวคือเมื่อเรากำหนดสัดส่วนการแชร์โหลด droop coefficient ของมุมแรงดัน และแรงดันไว้เท่ากันที่ 1:1 แล้ว พบว่ามีการแชร์อัตราส่วนกำลังไฟฟ้าจริง และ กำลังไฟฟ้ารีแอกทีฟ ที่อัตราส่วนประมาณ 1ต่อ1 ภายใต้ระยะเวลาเข้าสู่สภาวะคงตัวที่เหมาะสม

ในส่วนของการแสดงผลข้อมูล จะเป็นการแสดงค่าที่วัดได้จากมิเตอร์ด้วยการส่ง ข้อมูลแบบ RS-485 แล้วแปลงข้อมูล Modbus มาอยู่ในโปรแกรมแสดงผลที่สร้างมาจากโปรแกรม Visual Studio 2010 โดยใช้ภาษา C# เพื่อดูค่าต่างๆจากมิเตอร์ในรูปแบบตัวเลข

5.2 ข้อเสนอแนะ

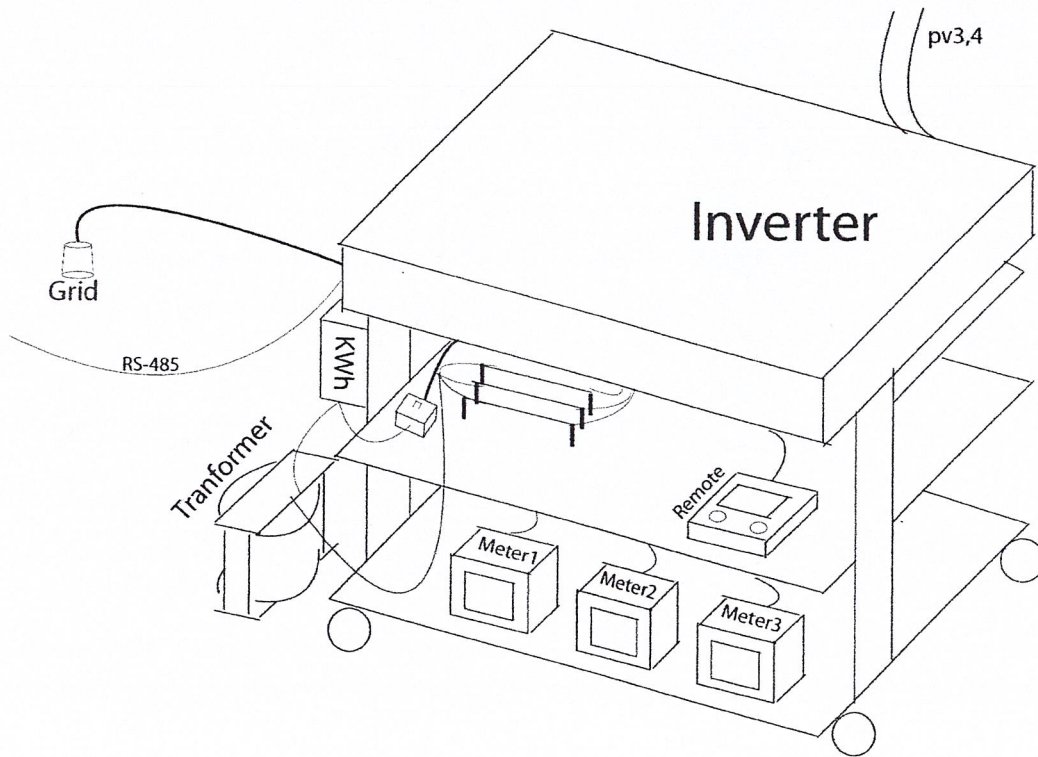
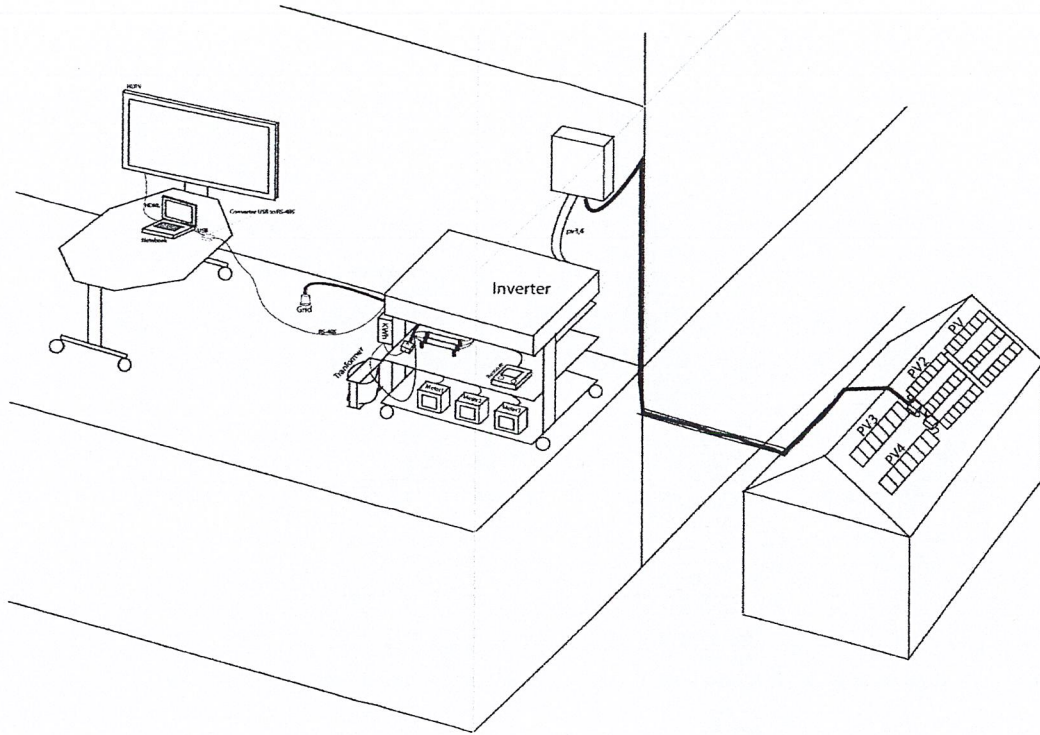
สำหรับโครงการฉบับนี้ยังไม่ได้ทดลองและจัดเก็บข้อมูลการทดลองจริง เพื่อนำมา เปรียบเทียบกับการจำลองด้วยโปรแกรม Matlab ว่าได้ผลลัพธ์ตามที่ออกแบบไว้หรือไม่

บรรณานุกรม

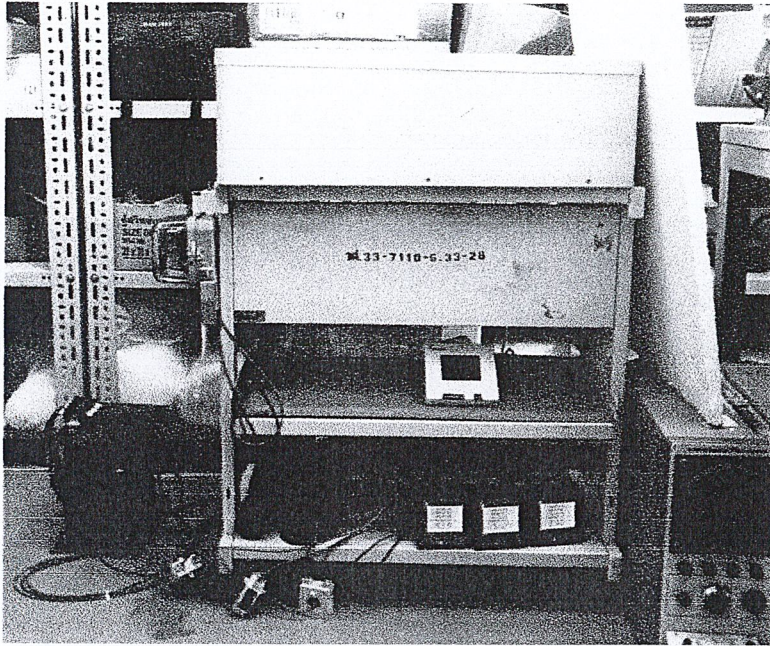
- [1] K. De Brabandere, B. Bolsens, J. Van den Keybus, A. Woyte, J. Driesen and R. Belmans, "A Voltage and Frequency Droop Control Method for Parallel Inverters", *IEEE Power Electronics*, vol.35,2004, pp. 1-5.
- [2] Andrew Mark Bollman, **An Experimental Study of Frequency Droop Control in a Low-Inertia Micgrid**, University of Illinois at Urbana-Champaign, Urbana, Illinois, 2009.
- [3] Hadi Saadat , **Power System Analysis** , McGraw-Hill, Inc., 1999.
- [4] Ritwik Majumder, Arindam Ghosh, Gerard Ledwich and Firuz Zare , **Control of parallel converters for load sharing with seamless transfer between grid connected and islanded modes.**
- [5] Xing Huang, Xinmin Jin, Tianyi Ma and Yibin Tong , **A Voltage and frequency droop control method for microsources.**
- [6] นายจิรวุฒิ ภู่อสาร ,นายณัฐพล นันทวิชาวุธสาร และณัฐพล ป้อมป้องกัน, "ระบบจัดการพลังงานสำหรับไมโครกริดโดยเอเจนท์", ปริญญานิพนธ์วิศวกรรมศาสตรบัณฑิต ,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,ปีการศึกษา 2554
- [7] กิตินันท์ พลสวัสดิ์ , **คู่มือเรียนและใช้งาน Visual C# 2010 ฉบับสมบูรณ์** , นนทบุรี, ไอดีซีฯ, 2554

ภาคผนวก

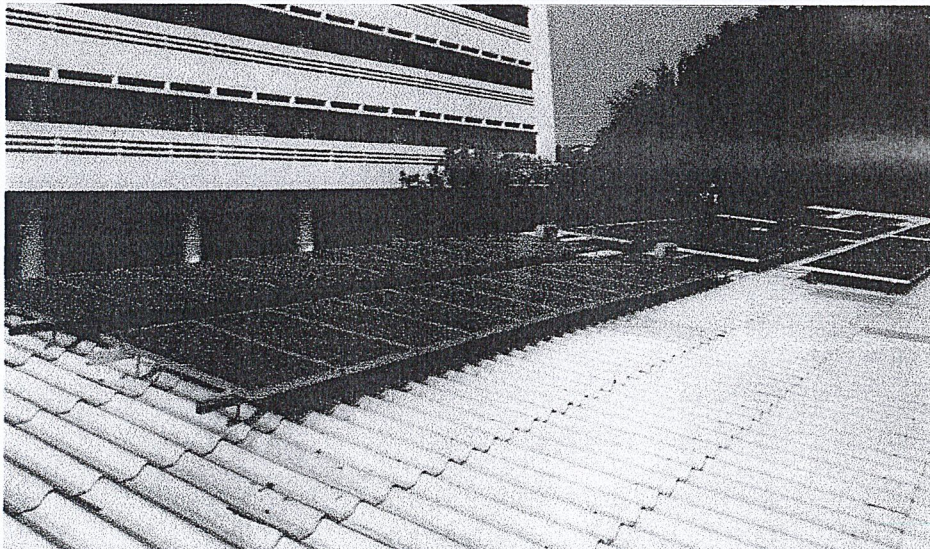
ภาคผนวก ก.
รูปอุปกรณ์ และ การติดตั้ง



รูปแผนผังการติดตั้งอุปกรณ์ทั้งหมด



รูปอุปกรณ์ Inverter และ Meter



รูปแผงโซล่าเซลล์

ภาคผนวก ข.
โปรแกรมแสดงผลข้อมูลจากมิเตอร์

Query Field Name	Example
Slave Address	21 ₁₆
Function Code	04 ₁₆
Starting Address High Byte	20 ₁₆
Starting Address Low Byte	02 ₁₆
Number of Registers High Byte	00 ₁₆
Number Of Registers Low Byte	02 ₁₆
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

รูปแบบการส่งข้อมูลของฟังก์ชัน04 ใน Modbus RTU

Response Field Name	Example
Slave Address	21 ₁₆
Function Code	04 ₁₆
Byte Count	04 ₁₆
Data from High Byte of Register X (e.g., 2002 ₁₆)	02 ₁₆
Data from Low Byte of Register X (e.g., 2002 ₁₆)	58 ₁₆
Data from High Byte of Register X+1 (e.g., 2003 ₁₆)	01 ₁₆
Data from Low Byte of Register X+1 (e.g., 2003 ₁₆)	2C ₁₆
Error Check Low Byte	CRC Low
Error Check High Byte	CRC High

รูปแบบการรับข้อมูลของฟังก์ชัน04 ใน Modbus RTU

โปรแกรมแสดงค่าจากมิเตอร์ โดยใช้ Microsoft Visual Studio 2010 ภาษา C# (.Net Framework 4.0)

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Timers;
using System.Threading;

namespace New_Interface_Meter
{
    public partial class Form1 : Form
    {
        SerialPort port = new SerialPort("COM1", 4800, Parity.None, 8, StopBits.One);
        public Form1()
        {
            InitializeComponent();
            CheckForIllegalCrossThreadCalls = true;
            btnDisconnect.Enabled = false;
        }
        private void Connectport_Click(object sender, EventArgs e)
        {
            port.ReadBufferSize = 100000;
            btnConnect.Enabled = false;
            btnDisconnect.Enabled = true;
            port.Open();
            port.DiscardOutBuffer();
            port.DiscardInBuffer();
            timer1.Interval = 500;
            timer1.Enabled = true;
        }
    }
}
```

```
private void timer1_Tick(object sender, EventArgs e)
{
port.DiscardOutBuffer();
port.DiscardInBuffer();
if (radMeter1.Checked == true)
{
SentMeter1();
Thread.Sleep(500);           //Relay
ReadMeter1();               //Relay
Thread.Sleep(200);
port.DiscardOutBuffer();
port.DiscardInBuffer();
Thread.Sleep(200);
SentMeter2();
Thread.Sleep(500);           //Relay
ReadMeter2();
Thread.Sleep(200);
port.DiscardOutBuffer();
port.DiscardInBuffer();
Thread.Sleep(200);
SentMeter3();
Thread.Sleep(500);           //Relay
ReadMeter3();
Thread.Sleep(200);
port.DiscardOutBuffer();
port.DiscardInBuffer();
Thread.Sleep(200);
SentMeter4();
Thread.Sleep(500);           //Relay
ReadMeter4();               //Relay
Thread.Sleep(200);

port.DiscardOutBuffer();
port.DiscardInBuffer();
}
if (radMeter2.Checked == true)
```

```
{
    SentMeter1M2();
    Thread.Sleep(500);           //Relay
    ReadMeter1M2();             //Relay
    Thread.Sleep(200);
    port.DiscardOutBuffer();
    port.DiscardInBuffer();
    Thread.Sleep(200);
    SentMeter2M2();
    Thread.Sleep(500);           //Relay
    ReadMeter2M2();
    Thread.Sleep(200);
    port.DiscardOutBuffer();
    port.DiscardInBuffer();
    Thread.Sleep(200);
    SentMeter3M2();
    Thread.Sleep(500);           //Relay
    ReadMeter3M2();
    Thread.Sleep(200);           //Relay
    SentMeter4M2();
    Thread.Sleep(500);           //Relay
    ReadMeter4M2();             //Relay
    Thread.Sleep(200);
    port.DiscardOutBuffer();
    port.DiscardInBuffer();
}
if (radMeter3.Checked == true)
{
    SentMeter1M3();
    Thread.Sleep(500);           //Relay
    ReadMeter1M3();             //Relay
    Thread.Sleep(200);
    port.DiscardOutBuffer();
    port.DiscardInBuffer();
    Thread.Sleep(200);
    SentMeter2M3();
    Thread.Sleep(500);           //Relay
    ReadMeter2M3();
```

```

        Thread.Sleep(200);
        port.DiscardOutBuffer();
        port.DiscardInBuffer();
        Thread.Sleep(200);
        SentMeter3M3();
        Thread.Sleep(500);           //Relay
        ReadMeter3M3();              //Relay
        Thread.Sleep(200);
        SentMeter4M3();
        Thread.Sleep(500);           //Relay
        ReadMeter4M3();              //Relay
        Thread.Sleep(200);
        port.DiscardOutBuffer();
        port.DiscardInBuffer();
    }
}
private void radMeter1_CheckedChanged(object sender, EventArgs e)
{
    timer1.Enabled = false;
    Thread.Sleep(500);
    Disconnectport_Click(sender, e);
}
private void radMeter2_CheckedChanged(object sender, EventArgs e)
{
    timer1.Enabled = false;
    Thread.Sleep(500);
    Disconnectport_Click(sender, e);
}
private void radMeter3_CheckedChanged(object sender, EventArgs e)
{
    timer1.Enabled = false;
    Thread.Sleep(500);
    Disconnectport_Click(sender, e);
}
private void Disconnectport_Click(object sender, EventArgs e)
{
    timer1.Enabled = false;
    port.Close();
    btnConnect.Enabled = true
}

```

```

    }
public void SentMeter1()
{
    byte[] btyWrite = new byte[8] { 0x01, 0x04, 0x00, 0x01, 0x00, 0x1C, 0xA0, 0x03 };
    port.Write(btyWrite, 0, 8);
}
public void ReadMeter1()
{
    byte[] btyRead = new byte[61];

    port.Read(btyRead, 0, 61);
    byte[] V1 = new byte[4];
    int i, j;
        j = 0;
    for (i = 3; i <= 6; i++)
    {
        V1[j] = btyRead[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_V1;
    str_V1 = BitConverter.ToString(V1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint num_V1 = uint.Parse(str_V1,
    System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_V1 มา
    byte[] floatVals_V1 = BitConverter.GetBytes(num_V1);
    float f_V1 = BitConverter.ToSingle(floatVals_V1, 0); // ตัวเลขที่ได้คือค่า f
    txtV1.Text = f_V1.ToString("###.###");

    byte[] V2 = new byte[4];
    j = 0;
    for (i = 7; i <= 10; i++)
    {
        V2[j] = btyRead[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_V2;

```

```

str_V2 = BitConverter.ToString(V2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_V2 = uint.Parse(str_V2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V2 มา
byte[] floatVals_V2 = BitConverter.GetBytes(num_V2);
float fl1 = BitConverter.ToSingle(floatVals_V2, 0); //ตัวเลขที่ได้คือค่า f
txtV2.Text = fl1.ToString("###.###");

byte[] V3 = new byte[4];
j = 0;
for (i = 11; i <= 14; i++)
{
    V3[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_V3;
str_V3 = BitConverter.ToString(V3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_V3 = uint.Parse(str_V3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V3 มา
byte[] floatVals_V3 = BitConverter.GetBytes(num_V3);
float f_V3 = BitConverter.ToSingle(floatVals_V3, 0); //ตัวเลขที่ได้คือค่า f
txtV3.Text = f_V3.ToString("###.###");

byte[] V12 = new byte[4];
j = 0;
for (i = 15; i <= 18; i++)
{
    V12[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_V12;
str_V12 = BitConverter.ToString(V12).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_V12 = uint.Parse(str_V12,

```

```

System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V12 มา
byte[] floatVals_V12 = BitConverter.GetBytes(num_V12);
float f_V12 = BitConverter.ToSingle(floatVals_V12, 0); //ตัวเลขที่ได้คือค่า f
txtV12.Text = f_V12.ToString("###.###");

```

```

byte[] V23 = new byte[4];
j = 0;
for (i = 19; i <= 22; i++)
{
    V23[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_V23;
str_V23 = BitConverter.ToString(V23).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_V23 = uint.Parse(str_V23,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V23มา
byte[] floatVals_V23 = BitConverter.GetBytes(num_V23);
float f_V23 = BitConverter.ToSingle(floatVals_V23, 0); //ตัวเลขที่ได้คือค่า f
txtV23.Text = f_V23.ToString("###.###");

```

```

byte[] V31 = new byte[4];
j = 0;
for (i = 23; i <= 26; i++)
{
    V31[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_V31;
str_V31 = BitConverter.ToString(V31).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_V31 = uint.Parse(str_V31,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V31 มา
byte[] floatVals_V31 = BitConverter.GetBytes(num_V31);
float f_V31 = BitConverter.ToSingle(floatVals_V31, 0); //ตัวเลขที่ได้คือค่า f
txtV31.Text = f_V31.ToString("###.###");

```

```

byte[] l1 = new byte[4];
j = 0;
for (i = 27; i <= 30; i++)
{
    l1[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_l1;
str_l1 = BitConverter.ToString(l1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_l1 = uint.Parse(str_l1,
System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_l1 มา
byte[] floatVals_l1 = BitConverter.GetBytes(num_l1);
float f_l1 = BitConverter.ToSingle(floatVals_l1, 0); //ตัวเลขที่ได้คือค่า f
txt1.Text = f_l1.ToString("###.###");

byte[] l2 = new byte[4];
j = 0;
for (i = 31; i <= 34; i++)
{
    l2[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_l2;
str_l2 = BitConverter.ToString(l2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_l2 = uint.Parse(str_l2,
System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_l2 มา
byte[] floatVals_l2 = BitConverter.GetBytes(num_l2);
float f_l2 = BitConverter.ToSingle(floatVals_l2, 0); //ตัวเลขที่ได้คือค่า f
txt2.Text = f_l2.ToString("###.###");

byte[] l3 = new byte[4];
j = 0;
for (i = 35; i <= 38; i++)

```

```

    {
        l3[j] = btyRead[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_l3;
    str_l3 = BitConverter.ToString(l3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint num_l3 = uint.Parse(str_l3,
        System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_l3 มา
    byte[] floatVals_l3 = BitConverter.GetBytes(num_l3);
    float f_l3 = BitConverter.ToSingle(floatVals_l3, 0); //ตัวเลขที่ได้คือค่า f
    txtl3.Text = f_l3.ToString("###.###");

    byte[] W1 = new byte[4];
    j = 0;
    for (i = 39; i <= 42; i++)
    {
        W1[j] = btyRead[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_W1;
    str_W1 = BitConverter.ToString(W1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint num_W1 = uint.Parse(str_W1,
        System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_W1 มา
    byte[] floatVals_W1 = BitConverter.GetBytes(num_W1);
    float f_W1 = BitConverter.ToSingle(floatVals_W1, 0); //ตัวเลขที่ได้คือค่า f
    txtW1.Text = f_W1.ToString("####.###");

    byte[] W2 = new byte[4];
    j = 0;
    for (i = 43; i <= 46; i++)
    {
        W2[j] = btyRead[i];
        j++;
    }

```

```

////-----Hex byteArray to String-----
string str_W2;
str_W2 = BitConverter.ToString(W2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint numW2 = uint.Parse(str_W2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_W2 มา
byte[] floatVals_W2 = BitConverter.GetBytes(numW2);
float f_W2= BitConverter.ToSingle(floatVals_W2, 0); //ตัวเลขที่ได้คือค่า f
txtW2.Text = f_W2.ToString("####.###");

byte[] W3 = new byte[4];
j = 0;
for (i = 47; i <= 50; i++)
{
    W3[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_W3;
str_W3 = BitConverter.ToString(W3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_W3 = uint.Parse(str_W3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_W3 มา
byte[] floatVals_W3 = BitConverter.GetBytes(num_W3);
float f_W3 = BitConverter.ToSingle(floatVals_W3, 0); //ตัวเลขที่ได้คือค่า f
txtW3.Text = f_W3.ToString("####.###");

byte[] Wtotal = new byte[4];
j = 0;
for (i = 51; i <= 54; i++)
{
    Wtotal[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_Wtotal;
str_Wtotal = BitConverter.ToString(Wtotal).Replace("-", ""); // เปลี่ยนที่ค่า btyRead

```

```

uint          num_Wtotal          =          uint.Parse(str_Wtotal,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_Wtotal มา
byte[] floatVals_Wtotal = BitConverter.GetBytes(num_Wtotal);
float f_Wtotal = BitConverter.ToSingle(floatVals_Wtotal, 0); //ตัวเลขที่ได้คือค่า f
txtWtotal.Text = f_Wtotal.ToString("#####.###");

byte[] VA1 = new byte[4];
j = 0;
for (i = 55; i <= 58; i++)
{
    VA1[j] = btyRead[i];
    j++;
}
////-----Hex byteArray to String-----
string str_VA1;
str_VA1 = BitConverter.ToString(VA1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_VA1 = uint.Parse(str_VA1,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_VA1 มา
byte[] floatVals_VA1 = BitConverter.GetBytes(num_VA1);
float f_VA1 = BitConverter.ToSingle(floatVals_VA1, 0); //ตัวเลขที่ได้คือค่า f
txtVA1.Text = f_VA1.ToString("#####.###");
}
public void SentMeter2()
{
byte[] btyWrite2 = new byte[8] { 0x01, 0x04, 0x00, 0x1D, 0x00, 0x1C, 0x61, 0xC5 };
port.Write(btyWrite2, 0, 8);
}
public void ReadMeter2()
{
byte[] btyRead2 = new byte[61];
port.Read(btyRead2, 0, 61);

byte[] VA2 = new byte[4];
int i, j;
j = 0;
for (i = 3; i <= 6; i++)
{VA2[j] = btyRead2[i];

```

```

        j++;
    }
    ///-----Hex byteArray to String-----
    string str_VA2;
    str_VA2 = BitConverter.ToString(VA2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_VA2          =          uint.Parse(str_VA2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_VA2 มา
    byte[] floatVals_VA2 = BitConverter.GetBytes(num_VA2);
    float f_VA2 = BitConverter.ToSingle(floatVals_VA2, 0); //ตัวเลขที่ได้คือค่า f
    txtVA2.Text = f_VA2.ToString("#####.###");

    byte[] VA3 = new byte[4];
    j = 0;
    for (i = 7; i <= 10; i++)
    {
        VA3[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_VA3;
    str_VA3 = BitConverter.ToString(VA3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_VA3          =          uint.Parse(str_VA3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_VA3 = BitConverter.GetBytes(num_VA3);
    float f_VA3 = BitConverter.ToSingle(floatVals_VA3, 0); //ตัวเลขที่ได้คือค่า f
    txtVA3.Text = f_VA3.ToString("#####.###");

    byte[] VAtotal = new byte[4];
    j = 0;
    for (i = 11; i <= 14; i++)
    {
        VAtotal[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_VAtotal;

```

```

str_VAtotal = BitConverter.ToString(VAtotal).Replace("-", ""); // เปลี่ยนที่ค่า btyRead

    ///-----Hex String to Float-----
    uint          num_VAtotal          =          uint.Parse(str_VAtotal,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_VAtotal = BitConverter.GetBytes(num_VAtotal);
    float f_VAtotal = BitConverter.ToSingle(floatVals_VAtotal, 0); //ตัวเลขที่ได้คือค่า f
    txtVAtotal.Text = f_VAtotal.ToString("#####.###");

    byte[] VAR1 = new byte[4];
    j = 0;
    for (i = 15; i <= 18; i++)
    {
        VAR1[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_VAR1;
    str_VAR1 = BitConverter.ToString(VAR1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_VAR1          =          uint.Parse(str_VAR1,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_VAR1 = BitConverter.GetBytes(num_VAR1);
    float f_VAR1 = BitConverter.ToSingle(floatVals_VAR1, 0); //ตัวเลขที่ได้คือค่า f
    txtVAR1.Text = f_VAR1.ToString("####.###");

    byte[] VAR2 = new byte[4];
    j = 0;
    for (i = 19; i <= 22; i++)
    {
        VAR2[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_VAR2;
    str_VAR2 = BitConverter.ToString(VAR2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----

```

```

uint          num_VAR2          =          uint.Parse(str_VAR2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_VAR2 = BitConverter.GetBytes(num_VAR2);
float f_VAR2 = BitConverter.ToSingle(floatVals_VAR2, 0); //ตัวเลขที่ได้คือค่า f
txtVAR2.Text = f_VAR2.ToString("#####.###");

byte[] VAR3 = new byte[4];
j = 0;
for (i = 23; i <= 26; i++)
{
    VAR3[j] = btyRead2[i];
    j++;
}
////-----Hex byteArray to String-----
string str_VAR3;
str_VAR3 = BitConverter.ToString(VAR3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_VAR3          =          uint.Parse(str_VAR3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_VAR3 = BitConverter.GetBytes(num_VAR3);
float f_VAR3 = BitConverter.ToSingle(floatVals_VAR3, 0); //ตัวเลขที่ได้คือค่า f
txtVAR3.Text = f_VAR3.ToString("#####.###");
//-----VARTotal-----
byte[] VARTotal = new byte[4];
j = 0;
for (i = 27; i <= 30; i++)
{
    VARTotal[j] = btyRead2[i];
    j++;
}
////-----Hex byteArray to String-----
string str_VARTotal;
str_VARTotal = BitConverter.ToString(VARTotal).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_VARTotal      =          uint.Parse(str_VARTotal,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_VARTotal = BitConverter.GetBytes(num_VARTotal);
float f_VARTotal = BitConverter.ToSingle(floatVals_VARTotal, 0); //ตัวเลขที่ได้คือค่า f

```

```

txtVARtotal.Text = f_VARtotal.ToString("####.###");
//-----PF1-----
    byte[] PF1 = new byte[4];
    j = 0;
    for (i = 31; i <= 34; i++)
    {
        PF1[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_PF1;
    str_PF1 = BitConverter.ToString(PF1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_PF1          =          uint.Parse(str_PF1,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_PF1 = BitConverter.GetBytes(num_PF1);
    float f_PF1 = BitConverter.ToSingle(floatVals_PF1, 0); //ตัวเลขที่ได้คือค่า f
    txtPF1.Text = f_PF1.ToString("#.###");
//-----PF2-----
    byte[] PF2 = new byte[4];
    j = 0;
    for (i = 35; i <= 38; i++)
    {
        PF2[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_PF2;
    str_PF2 = BitConverter.ToString(PF2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_PF2          =          uint.Parse(str_PF2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_PF2 = BitConverter.GetBytes(num_PF2);
    float f_PF2 = BitConverter.ToSingle(floatVals_PF2, 0); //ตัวเลขที่ได้คือค่า f
    txtPF2.Text = f_PF2.ToString("#.###");
//-----PF3-----
    byte[] PF3 = new byte[4];
    j = 0;

```

```

for (i = 39; i <= 42; i++)
{
    PF3[j] = btyRead2[i];
    j++;
}
////-----Hex byteArray to String-----
string str_PF3;
str_PF3 = BitConverter.ToString(PF3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_PF3          =          uint.Parse(str_PF3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_PF3 = BitConverter.GetBytes(num_PF3);
float f_PF3 = BitConverter.ToSingle(floatVals_PF3, 0); //ตัวเลขที่ได้คือค่า f
txtPF3.Text = f_PF3.ToString("#.###");
//-----PFtotal-----
byte[] PFtotal = new byte[4];
j = 0;
for (i = 43; i <= 46; i++)
{
    PFtotal[j] = btyRead2[i];
    j++;
}
////-----Hex byteArray to String-----
string str_PFtotal;
str_PFtotal = BitConverter.ToString(PFtotal).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_PFtotal      =          uint.Parse(str_PFtotal,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_PFtotal = BitConverter.GetBytes(num_PFtotal);
float f_PFtotal = BitConverter.ToSingle(floatVals_PFtotal, 0); //ตัวเลขที่ได้คือค่า f
txtPFtotal.Text = f_PFtotal.ToString("#.###");
//-----Hz1-----
byte[] Hz1 = new byte[4];
j = 0;
for (i = 47; i <= 50; i++)
{
    Hz1[j] = btyRead2[i];
    j++;
}

```

```

    }
    ///-----Hex byteArray to String-----
    string str_Hz1;
    str_Hz1 = BitConverter.ToString(Hz1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_Hz1          =          uint.Parse(str_Hz1,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_Hz1 = BitConverter.GetBytes(num_Hz1);
    float f_Hz1 = BitConverter.ToSingle(floatVals_Hz1, 0); //ตัวเลขที่ได้คือค่า f
    txtHz1.Text = f_Hz1.ToString("###.###");
    //-----Hz2-----
    byte[] Hz2 = new byte[4];
    j = 0;
    for (i = 51; i <= 54; i++)
    {
        Hz2[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_Hz2;
    str_Hz2 = BitConverter.ToString(Hz2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_Hz2          =          uint.Parse(str_Hz2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_Hz2 = BitConverter.GetBytes(num_Hz2);
    float f_Hz2 = BitConverter.ToSingle(floatVals_Hz2, 0); //ตัวเลขที่ได้คือค่า f
    txtHz2.Text = f_Hz2.ToString("###.###");
    //-----Hz3-----
    byte[] Hz3 = new byte[4];
    j = 0;
    for (i = 55; i <= 58; i++)
    {
        Hz3[j] = btyRead2[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_Hz3;
    str_Hz3 = BitConverter.ToString(Hz3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead

```

```

    ///-----Hex String to Float-----
    uint          num_Hz3          =          uint.Parse(str_Hz3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_Hz3 = BitConverter.GetBytes(num_Hz3);
    float f_Hz3 = BitConverter.ToSingle(floatVals_Hz3, 0); //ตัวเลขที่ได้คือค่า f
    txtHz3.Text = f_Hz3.ToString("###.###");
}

    public void SentMeter3()
    {
        byte[] btyWrite3 = new byte[8] { 0x01, 0x04, 0x00, 0x4F, 0x00, 0x06, 0x41, 0xDF };
        port.Write(btyWrite3, 0, 8);
    }

    public void ReadMeter3()
    {
        byte[] btyRead3 = new byte[17]; //(3*4+5)
        port.Read(btyRead3, 0, 17);
        int i, j;
        ///-----KWh-----
        byte[] KWh = new byte[4];
        j = 0;
        for (i = 3; i <= 6; i++)
        {
            KWh[j] = btyRead3[i];
            j++;
        }
        ///-----Hex byteArray to String-----
        string str_KWh;
        str_KWh = BitConverter.ToString(KWh).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
        ///-----Hex String to Float-----
        uint          num_KWh          =          uint.Parse(str_KWh,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
        byte[] floatVals_KWh = BitConverter.GetBytes(num_KWh);
        float f_KWh = BitConverter.ToSingle(floatVals_KWh, 0); //ตัวเลขที่ได้คือค่า f
        txtKWh.Text = f_KWh.ToString("#####.###");
        ///-----KVAh-----
        byte[] KVAh = new byte[4];
        j = 0;
        for (i = 7; i <= 10; i++)

```

```

    { KVAh[j] = btyRead3[i];
      j++;
    }
    ///-----Hex byteArray to String-----
    string str_KVAh;
str_KVAh = BitConverter.ToString(KVAh).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_KVAh          =          uint.Parse(str_KVAh,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_KVAh = BitConverter.GetBytes(num_KVAh);
    float f_KVAh = BitConverter.ToSingle(floatVals_KVAh, 0); //ตัวเลขที่ได้คือค่า f
    txtKVAh.Text = f_KVAh.ToString("#####.###");
    //-----KVARh-----
    byte[] KVARh = new byte[4];
    j = 0;
    for (i = 11; i <= 14; i++)
    {
        KVARh[j] = btyRead3[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_KVARh;
str_KVARh = BitConverter.ToString(KVARh).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_KVARh          =          uint.Parse(str_KVARh,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_KVARh = BitConverter.GetBytes(num_KVARh);
    float f_KVARh = BitConverter.ToSingle(floatVals_KVARh, 0); //ตัวเลขที่ได้คือค่า f
    txtKVARh.Text = f_KVARh.ToString("#####.###");
    }
public void SentMeter4()
{
    byte[] btyWrite4 = new byte[8] { 0x01, 0x04, 0x00, 0x77, 0x00, 0x12, 0xC0, 0x1D };
    port.Write(btyWrite4, 0, 8);
}
public void ReadMeter4()
{
    byte[] btyRead4 = new byte[41]; //(9*4)+5

```

```

port.Read(btyRead4, 0, 41);
int i, j;

//-----KWh1-----
byte[] KWh1 = new byte[4];
j = 0;
for (i = 3; i <= 6; i++)
{
    KWh1[j] = btyRead4[i];
    j++;
}
////-----Hex byteArray to String-----
string str_KWh1;
str_KWh1 = BitConverter.ToString(KWh1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_KWh1          =          uint.Parse(str_KWh1,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_KWh1 = BitConverter.GetBytes(num_KWh1);
float f_KWh1 = BitConverter.ToSingle(floatVals_KWh1, 0); //ตัวเลขที่ได้คือค่า
f

txtKWh1.Text = f_KWh1.ToString("#####.###");
//-----KWh2-----
byte[] KWh2 = new byte[4];
j = 0;
for (i = 7; i <= 10; i++)
{
    KWh2[j] = btyRead4[i];
    j++;
}
////-----Hex byteArray to String-----
string str_KWh2;
str_KWh2 = BitConverter.ToString(KWh2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_KWh2          =          uint.Parse(str_KWh2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_KWh2 = BitConverter.GetBytes(num_KWh2);
float f_KWh2 = BitConverter.ToSingle(floatVals_KWh2, 0); //ตัวเลขที่ได้คือค่า
f

```

```

txtKWh2.Text = f_KWh2.ToString("#####.###");
//-----KWh3-----
byte[] KWh3 = new byte[4];
j = 0;

for (i = 11; i <= 14; i++)
{
    KWh3[j] = btyRead4[i];
    j++;
}
////-----Hex byteArray to String-----
string str_KWh3;
str_KWh3 = BitConverter.ToString(KWh3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_KWh3          =          uint.Parse(str_KWh3,
System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_V1 มา
byte[] floatVals_KWh3 = BitConverter.GetBytes(num_KWh3);
float f_KWh3 = BitConverter.ToSingle(floatVals_KWh3, 0); //ตัวเลขที่ได้คือค่า
f

txtKWh3.Text = f_KWh3.ToString("#####.###");
//-----KVAh1-----
byte[] KVAh1 = new byte[4];
j = 0;
for (i = 15; i <= 18; i++)
{
    KVAh1[j] = btyRead4[i];
    j++;
}
////-----Hex byteArray to String-----
string str_KVAh1;
str_KVAh1 = BitConverter.ToString(KVAh1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_KVAh1          =          uint.Parse(str_KVAh1,
System.Globalization.NumberStyles.AllowHexSpecifier); // รับค่า str_V1 มา
byte[] floatVals_KVAh1 = BitConverter.GetBytes(num_KVAh1);
float f_KVAh1 = BitConverter.ToSingle(floatVals_KVAh1, 0); //ตัวเลขที่ได้คือค่า f
txtKVAh1.Text = f_KVAh1.ToString("#####.###");
//-----KVAh2-----

```

```

byte[] KVAh2 = new byte[4];
j = 0;
for (i = 19; i <= 22; i++)
{ KVAh2[j] = btyRead4[i];
  j++;
}
////-----Hex byteArray to String-----
string str_KVAh2;
str_KVAh2 = BitConverter.ToString(KVAh2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_KVAh2          =          uint.Parse(str_KVAh2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_KVAh2 = BitConverter.GetBytes(num_KVAh2);
float f_KVAh2 = BitConverter.ToSingle(floatVals_KVAh2, 0); //ตัวเลขที่ได้คือค่า f
txtKVAh2.Text = f_KVAh2.ToString("#####.###");
// -----KVAh3-----
byte[] KVAh3 = new byte[4];
j = 0;
for (i = 23; i <= 26; i++)
{
  KVAh3[j] = btyRead4[i];
  j++;
}
////-----Hex byteArray to String-----
string str_KVAh3;
str_KVAh3 = BitConverter.ToString(KVAh3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint          num_KVAh3          =          uint.Parse(str_KVAh3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_KVAh3 = BitConverter.GetBytes(num_KVAh3);
float f_KVAh3 = BitConverter.ToSingle(floatVals_KVAh3, 0); //ตัวเลขที่ได้คือค่า f
txtKVAh3.Text = f_KVAh3.ToString("#####.###");
//-----KVARh1-----
byte[] KVARh1 = new byte[4];
j = 0;
for (i = 27; i <= 30; i++)
{
  KVARh1[j] = btyRead4[i];

```

```

        j++;
    }
    //-----Hex byteArray to String-----
    string str_KVARh1;
    str_KVARh1 = BitConverter.ToString(KVARh1).Replace("-", ""); // เปลี่ยนที่ค่า btyRead

    ///-----Hex String to Float-----
    uint          num_KVARh1          =          uint.Parse(str_KVARh1,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_KVARh1 = BitConverter.GetBytes(num_KVARh1);
    float f_KVARh1 = BitConverter.ToSingle(floatVals_KVARh1, 0); //ตัวเลขที่ได้คือค่า f
    txtKVARh1.Text = f_KVARh1.ToString("#####.###");
    //-----KVARh2-----
    byte[] KVARh2 = new byte[4];
    j = 0;
    for (i = 31; i <= 34; i++)
    {
        KVARh2[j] = btyRead4[i];
        j++;
    }
    ///-----Hex byteArray to String-----
    string str_KVARh2;
    str_KVARh2 = BitConverter.ToString(KVARh2).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
    ///-----Hex String to Float-----
    uint          num_KVARh2          =          uint.Parse(str_KVARh2,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
    byte[] floatVals_KVARh2 = BitConverter.GetBytes(num_KVARh2);
    float f_KVARh2 = BitConverter.ToSingle(floatVals_KVARh2, 0); //ตัวเลขที่ได้คือค่า f
    txtKVARh2.Text = f_KVARh2.ToString("#####.###");
    //-----KVARh3-----
    byte[] KVARh3 = new byte[4];
    j = 0;
    for (i = 35; i <= 38; i++)
    {
        KVARh3[j] = btyRead4[i];
        j++;
    }
    ///-----Hex byteArray to String-----

```

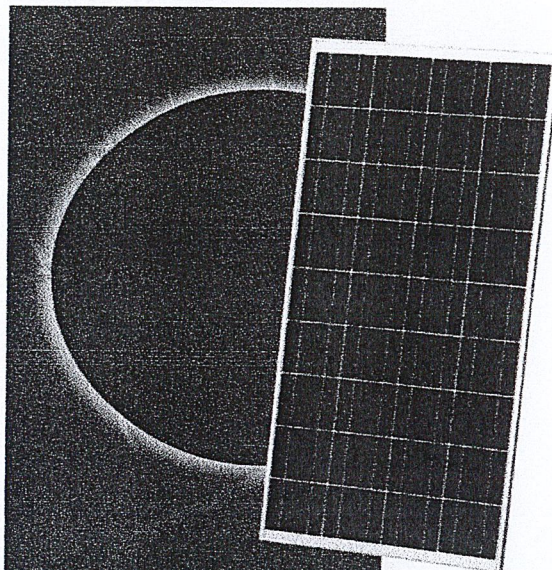
```

string str_KVARh3;
str_KVARh3 = BitConverter.ToString(KVARh3).Replace("-", ""); // เปลี่ยนที่ค่า btyRead
////-----Hex String to Float-----
uint num_KVARh3 = uint.Parse(str_KVARh3,
System.Globalization.NumberStyles.AllowHexSpecifier);// รับค่า str_V1 มา
byte[] floatVals_KVARh3 = BitConverter.GetBytes(num_KVARh3);
float f_KVARh3 = BitConverter.ToSingle(floatVals_KVARh3, 0); //ตัวเลขที่ได้คือค่า f
txtKVARh3.Text = f_KVARh3.ToString("#####.###");
}
}
}

```

หมายเหตุ ส่วนฟังก์ชันอื่นๆที่เหลือ อยู่ในไฟล์ CD

ภาคผนวก ค.
รายละเอียดอุปกรณ์



130 WATT

**DESIGNED FOR
LARGE ELECTRICAL
POWER REQUIREMENTS**

FEATURES

- High-power module (130W) using 156.5mm square multi-crystal silicon solar cells with 13.0% module conversion efficiency
- Photovoltaic module with bypass diode minimises the power drop caused by shade
- Anti reflection coating and BSF (Back-Surface Field) structure to improve cell conversion efficiency, 14.7%
- White tempered glass, EVA resin and a weatherproof film, plus aluminum frame for extended outdoor use
- Nominal 12 volt output for battery charging applications
- Output terminal: Lead wire with waterproof connector
- Certifications: IEC 61215 & IEC 61730

SHARP modules are manufactured in ISO 9001 certified factories.

**POLYCRYSTALLINE SILICON PHOTOVOLTAIC
MODULE WITH 130W MAXIMUM POWER**

A safe, clean, reliable source of energy, Sharp's **ND-130T1J** photovoltaic module is designed for a variety of electrical power requirements. Based on the technology of crystal silicon solar cells developed over 50 years, this module has superb durability to withstand rigorous operating conditions and is suitable for use in most solar systems.

Common applications for the Sharp ND-130T1J include office buildings, private residences, RVs, cabins, vacation homes, solar power stations, pumps, beacons and lighting equipment. As one of the world's leading manufacturer's of photovoltaic modules, Sharp produces an extensive line of high power modules for every electrical power requirement.

รูปรายละเอียดสเปคแผงโซลาร์เซลล์

ND-130T1J – MULTI-PURPOSE MODULE

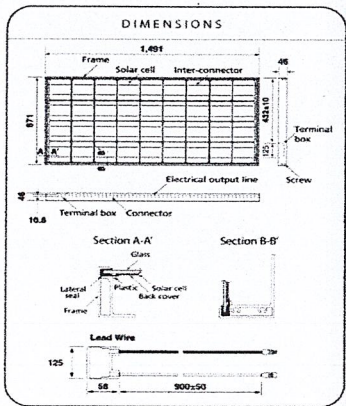
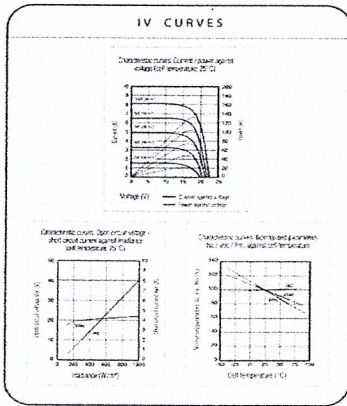
ELECTRICAL CHARACTERISTICS	
Cell	Multi-crystalline (156 Series) / Sharp silicon solar cells
No. of Cells and Connections	36 in series
Open Circuit Voltage (V _{oc})	22.2V
Maximum Power Voltage (V _{mp})	17.4V
Short Circuit Current (I _{sc})	8.09A
Maximum Power Current (I _{mp})	7.48A
Maximum Power (P _{max})	Min. 123.9W Typical 130W
Encapsulated Solar Cell Efficiency (η _{cell})	14.7%
Module Efficiency (η _m)	13.0%
Maximum System Voltage	DC 600V
Series Fuse Rating	15A
Type of Output Terminal	Lead - MC4 Compatible

STC (Standard Test Conditions: 25°C, 1 kW/m², AM1.5)

MECHANICAL CHARACTERISTICS	
Dimensions	1491 x 671 x 46mm
Weight	14.0kg

TEMPERATURE COEFFICIENT		
Temp. Coefficient of P _{max}	-0.465	%/°C
Temp. Coefficient of V _{oc}	-0.078	V/°C
Temp. Coefficient of I _{sc}	0.053	%/°C

ABSOLUTE MAXIMUM RATINGS		
Parameters	Rating	Unit
Operating Temperature	-40 to +90	°C
Storage Temperature	-40 to +90	°C
Dielectric Voltage Withstood	600 max.	V-DC



In the absence of confirmation by device specifications sheets, Sharp takes no responsibility for any defects that may occur in equipment using any Sharp devices shown in catalogues, data books, etc. Contact Sharp in order to obtain the latest device specification sheets before using any Sharp device.

- Design and specifications are subject to change without prior notice.
- Colour variations to products may occur due to printing.
- All information and technical details are correct as at product release date.

SHARP CORPORATION OF AUSTRALIA PTY LTD
 ABN 40 003 039 405
 1 Mentingwood Drive, Huntingwood, NSW 2148
 PO Box 6827, Blacktown, NSW 2148
 Tel: (02) 9830-6600 Fax: (02) 9672 1208
 www.sharp.net.au

© SHARP CORP. PRINTED IN AUSTRALIA

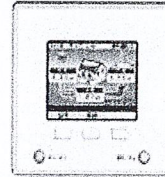


รูปรายละเอียดสเปคแผงโซล่าเซลล์

Details & Spec

No need joint box or booster unit.
 These functions are included in this product.
 MPPT function is fully applicable by string system.

- ▶ High-reliable design·Insulated by high-frequency transformer
- ▶ String system of input circuit for solar cell
- ▶ Maximum efficiency is 94%.
- ▶ Designed for outdoor use (IP55)
- ▶ Stand-alone operation(1.5kW)
- ▶ Color remote-controller
- ▶ Original anti-islanding protection (Patent applied)



Color remote controller

	Z50-30ST2	Z50-55ST3
Phase/output voltage	Single-phase three-wire/ AC202V±10%	
Rated input voltage	DC250V	
Possible input voltage range	DC0V~450V	
Input voltage range	DC80V~420V	
Input circuit	2	3
Rated output	3.0kW	5.5kW
Maximum output current	15A	27.5A
Insulation system	Insulated by high-frequency transformer	
Dimensions (mm)	W676×H490 ×D200 (Without protrusion)	
Case color	Munsell 5Y7/1(70GU)	

*Please contact us if you want custom-made.

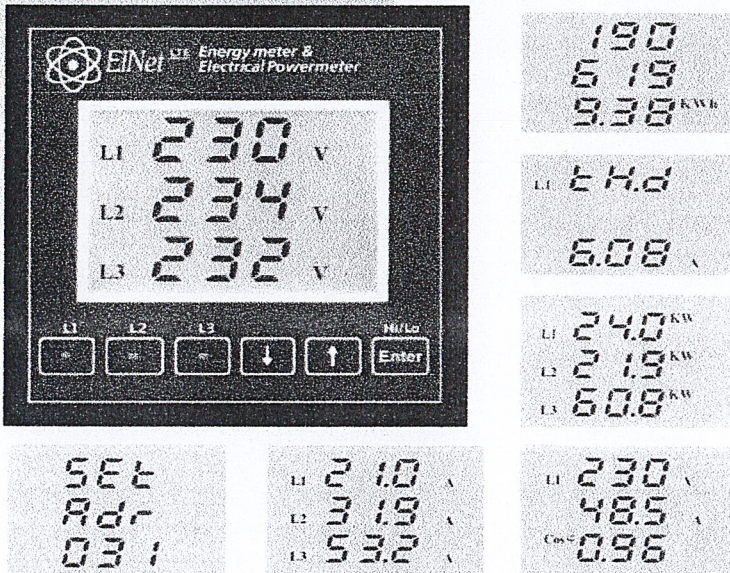
รูปรายละเอียดของ Inverter

EiNet LTE
Energy Meter & Electrical Powermeter

มิเตอร์วัดกำลังงานไฟฟ้า




คุณสมบัติพิเศษ
สามารถวัดกำลังงานไฟฟ้าได้ 3 เฟส
ความเที่ยงตรงสูง 0.5%
วัดค่าพารามิเตอร์ทางไฟฟ้าต่างๆ V, A, Hz
วัดค่ากำลังไฟฟ้า kWh และเพาเวอร์แฟกเตอร์
วัดค่าฮาร์มอนิกของกระแส, แรงดัน THD-I, THD-V
สามารถแสดงค่า Real Power Demand และกระแสได้
สามารถแสดงค่าสูงสุดของทุกค่าที่มีการวัดได้
หน้าจอ LCD, backlight สามารถมองเห็นได้ในที่มืด
หน้าจอมีหลอด LED แสดงสถานะของทุกเฟส
สนับสนุนพอร์ตสื่อสาร RS - 485
มีเอาต์พุตวิเคราะห์ที่สามารถตั้งโปรแกรมได้ (Optional)
เมนูสำหรับการใช้งานง่าย และรวดเร็ว
ติดตั้งใช้งานง่ายที่หน้าตู้คอนโทรล

ข้อมูลจำเพาะทางเทคนิค
ไฟเลี้ยงตัวมิเตอร์ : 110 or 230 VAC 10%
: 60/50 Hz
: 20VA
ขนาด (สูง x กว้าง x ลึก) : 96 x 96 x 80 มม.
น้ำหนัก : 450 กรัม
เงื่อนไขการทำงาน : -20 - 70°C
: 0 - 95%RH
ช่วงการวัด
แรงดันไฟฟ้า : 0 - 650 VAC
แรงดันไฟฟ้า (หม้อแปลง) : สูงสุด 999 KV
กระแสไฟฟ้า (หม้อแปลง) : สูงสุด 999 KA
แรงดัน Input สูงสุด : 1000V
กระแส Input สูงสุด : 6A



รูปรายละเอียดสเปคมิเตอร์

ประวัติผู้เขียน

	<p>นายพุงศักดิ์ แซ่หลิว เกิดวันที่ 1 มกราคม 2534 ภูมิลำเนา จังหวัดกรุงเทพฯ การศึกษา - มัธยมศึกษา โรงเรียนสวดาขรรค์ชัยกัมพลานนท์อนุสรณ์ - ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง</p>
	<p>นายพงศักรณ์ เตชะชลประเสริฐ เกิดวันที่ 24 ธันวาคม 2533 ภูมิลำเนา จังหวัดกรุงเทพฯ การศึกษา - มัธยมศึกษา โรงเรียนไตรมิตรวิทยาลัย - ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง</p>
	<p>นายไพโรจน์ คอนอม เกิดวันที่ 9 สิงหาคม 2533 ภูมิลำเนา จังหวัดเชียงใหม่ การศึกษา - มัธยมศึกษา โรงเรียนจักรคำคณาทร จังหวัดลำพูน - ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง</p>
	<p>นายมหाराช ราชสิทธิ์ เกิดวันที่ 16 พฤศจิกายน 2533 ภูมิลำเนา จังหวัดอุดรธานี การศึกษา - มัธยมศึกษา โรงเรียนอุดรพิทยานุกูล จังหวัดอุดรธานี - ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง</p>

โครงการสาธิตไมโครกริดของศูนย์นวัตกรรมระบบพลังงาน

CInES Microgrid Demonstration: Operation and control

พยุศักดิ์ แซ่หลิว พงศ์ภรณ์ เตชะชลประเสริฐ ไพโรจน์ คอนอม และมหาธราช ราชสีห์

1 ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

1 ซอย ผลองกรัง 1 ลาดกระบัง กรุงเทพมหานคร 10520 โทรศัพท์ 02-329-8000 ต่อ 3925 E-Mail: fad_phayung@hotmail.com

บทคัดย่อ

ปัญญานิพนธ์นี้นำเสนอระบบไมโครกริดสาธิตขนาด 3 kW ของศูนย์นวัตกรรมระบบพลังงาน โดยมีวัตถุประสงค์เพื่อศึกษา คัดตั้ง และควบคุมระบบไมโครกริด ในปัญญานิพนธ์นี้ได้แบ่งการศึกษา ออกเป็นสองส่วนคือ ส่วนของการศึกษาระบบควบคุมไมโครกริดใน สภาวะแยกอิสระจากกริดซึ่งควบคุมระบบด้วยเทคนิค Angle Droop และ Voltage Droop ซึ่งเป็นวิธีการในการควบคุมของแหล่งกำเนิดไฟฟ้า ขนาดเล็ก โดยในการศึกษาในส่วนนี้ได้ทำการจำลองระบบไมโครกริด แบบแยกอิสระจากกริดทั้งระบบสามเฟสและหนึ่งเฟสเพื่อยืนยันความ ถูกต้องของระบบควบคุมที่ได้ออกแบบ ในส่วนของการทดลองติดตั้ง ระบบผลิตไฟฟ้าจากเซลล์แสงอาทิตย์ได้ทำการติดตั้งแผงโซลาร์เซลล์ พิกัดรวม 3kW และใช้อินเวอร์เตอร์เฟสเดียวต่อเชื่อมกับระบบไฟฟ้า จากนั้นได้พัฒนาระบบตรวจวัดระยะไกลผ่านการสื่อสาร Modbus ขึ้นมาสำหรับการเก็บข้อมูลและควบคุมในอนาคต

คำสำคัญ: ไมโครกริด, Angle Droop, Voltage Droop and Modbus

Abstract

This project presents CInES 3-kW microgrid demonstration system. The objectives are to study the installation and control of the microgrid system. In this project, the studies are divided into two parts. The first part concerns about control of a microgrid in stand-alone mode using angle and voltage droop techniques. Three-phase and single-phase microgrids are modeled to confirm the validation of the techniques. In the second part, 3-kW solar panels have been installed for evaluation at KMITL. A single-phase solar inverter is used to supply the generated power to the power system. A modbus communication network has been created for data acquisition and control of the microgrid in the future.

Keyword: Microgrid, Angle Droop, Voltage Droop and Modbus

1. บทนำ

เนื่องจากปัจจุบันมนุษย์ให้ความสนใจกับปัญหาสิ่งแวดล้อม และบรรเทาปัญหาโลกร้อน (Global Warming) กันมากขึ้น ทำให้เกิดความสนใจเกี่ยวกับการผลิตพลังงานไฟฟ้าแบบกระจายจากแหล่ง

พลังงานธรรมชาติ หรือพลังงานหมุนเวียน (Renewable

Energy Resources) เช่น พลังงานลม พลังงานแสงอาทิตย์ หรือ พลังงาน ชีวมวล เป็นต้น ซึ่งการผลิตไฟฟ้าแบบกระจายนั้นจะเป็นระบบขนาดเล็ก

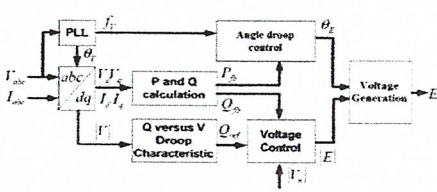
อุปกรณ์ผลิตพลังงานไฟฟ้าจากแหล่งพลังงานธรรมชาติ เหล่านี้ โดยทั่วไปแล้ว กำลังไฟฟ้าที่ผลิตได้จะมีปริมาณที่ผันผวนเป็น อย่างมาก จึงทำให้เกิดความกังวลเกี่ยวกับผลกระทบในแง่ลบทั้งในด้าน ของคุณภาพไฟฟ้าและความเชื่อถือได้ของระบบในกรณีที่มีการนำ แหล่งกำเนิดไฟฟ้าเหล่านี้ จำนวนมากเข้ามาต่อกับระบบไฟฟ้า จึงได้มี การนำเสนอแนวทางการผลิตและส่งจ่ายไฟฟ้าภายในพื้นที่สำหรับระบบ ไฟฟ้ากำลังขนาดเล็ก โดยที่ผลิตไฟฟ้าได้ในระดับแรงดันต่ำ จึงทำให้ สามารถเชื่อมต่อกับระบบเข้ากับการะไฟฟ้า และเชื่อมต่อกับกริดไฟฟ้าหลัก ได้เรียกระบบการผลิตไฟฟ้าขนาดเล็กเหล่านี้ว่า“ไมโครกริด” (Microgrid)

อย่างไรก็ตามไมโครกริดก็ยังมีคุณลักษณะเหมือนกริดหลัก จึง ต้องมีการควบคุมคุณภาพของไฟฟ้าให้มีความเสถียรภาพตามมาตรฐาน ของผู้ใช้ โดยในโครงการนี้ได้นำเทคนิค Voltage Droop และ Frequency Droop มาประยุกต์ใช้โดยศึกษาจากการกำหนดแบบจำลองพื้นฐานระบบ ไมโครกริด

2. ระบบไมโครกริด

ไมโครกริดคือการรวมระบบพลังงานซึ่งประกอบด้วย การเชื่อมต่อของโหลดภายในระบบและทรัพยากรพลังงานแบบกระจาย (Distributed energy resources) ซึ่งเป็นระบบที่สามารถปฏิบัติการคู่ขนาน ไปกับ กริดหลัก (Main Grid) ซึ่งเรียกว่า โหมดเชื่อมต่อกับกริด (Grid connect) หรือในสถานะที่แยกตัวอิสระจากกริดหลักซึ่งเรียกว่าโหมดแยก อิสระ (Stand-alone)

3. วิธีการควบคุมคอนเวอร์เตอร์



รูปที่ 1 ภาพรวมของการควบคุม

รูปที่ 1 แสดงภาพรวมของการควบคุม โดยความถี่ที่บัส f_v นั้นจะถูกปรับมาจาก phase locking loop (PLL) และขนาดของแรงดันที่บัส $|V|$ นั้นจะถูกคำนวณจากบล็อกการแปลงกรอบอ้างอิงหมุน ω_s จะได้มาจากความถี่สัมพันธ์ p-f droop characteristic ของสมการที่ (3) และ Q_{ref} จะได้จากโมดูล Q-V droop characteristic โดยโมดูล Angle Droop จะใช้เพื่อหามุมเฟส $\angle\theta_E$ แรงดันที่ออกมาจากอินเวอร์เตอร์ \vec{E} ในขณะที่โมดูล Voltage control นั้นจะถูกใช้เพื่อหาขนาดของแรงดันเอาต์พุต $|E|$ โดยแรงดันเอาต์พุต \vec{E} ที่ออกมาจะเกิดจากขนาดและมุมเฟสของฟังก์ชันไซน์นั่นเอง

3.1 โมดูลการควบคุมมุมแรงดัน

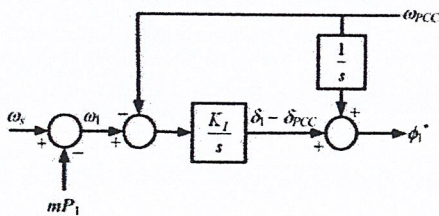
จากจุดเริ่มต้นของการควบคุมก็คือการใช้วิธี droop แบบธรรมดา ตามสมการ

$$\begin{aligned} \omega &= \omega^* - mP \\ V &= V^* - nQ \end{aligned} \quad (1)$$

เมื่อ m และ n คือ droop coefficient, ω_s^* คือความถี่เชิงโครนิส, V คือขนาดของแรงดันเอาต์พุตของคอนเวอร์เตอร์ และ ω คือความถี่ของอินเวอร์เตอร์ตามลำดับ เพราะฉะนั้นความถี่และขนาดของแรงดันจะถูกควบคุมผ่านการควบคุมกำลังไฟฟ้าจริงและกำลังไฟฟ้าเสมือนที่ออกจากแหล่งจ่าย DG.

รูปการควบคุมมุมแรงดันจะถูกแสดงตามรูปที่ 2 ความถี่ ω_1 นั้นจะถูกคำนวณจากสมการที่ 1 ซึ่งหลังจากนั้นจะถูกเปรียบเทียบกับความถี่ที่ได้จากแรงดันตรงจุดPCC, ω_{PCC} ค่าความผิดพลาดที่ได้จะถูกรวมกับอินทิเกรตของ ω_{PCC} เพื่อหา ϕ_1^* โดยมุม ϕ_1^* ที่ได้จะหามุมด้วยความเร็วเชิงโครนิส ω_1 ซึ่งจะทำมุม δ_1^* กับแรงดันอ้างอิง

ทั้งนี้หากเราเปลี่ยนค่าของ K_I เราสามารถที่จะเปลี่ยนผลตอบสนองทางไดนามิกของ close loop ได้โดยไม่มีผลกระทบต่อการทำงานของความถี่ที่สภาวะคงตัว



รูปที่ 2 รูปการควบคุมของมุมแรงดัน

3.2 โมดูลการควบคุมขนาดแรงดัน

เนื่องจากกำลังไฟฟ้าเสมือน Q_L นั้นเป็นสัดส่วนโดยตรงกับผลต่างของขนาด \vec{E} และ \vec{V} คอนโทรลเลอร์สามารถควบคุม Q_L โดยการเปลี่ยนขนาดของ \vec{E} โมดูลนี้นั้นจะถูกคำนวณขนาดแรงดันเอาต์พุตของอินเวอร์เตอร์ $|E|$ โดยที่ตัวแปรอินพุตคือ กำลังไฟฟ้าเสมือนที่ต้องการ Q_{ref} ตามสมการที่ 2 และกำลังไฟฟ้าเสมือนที่ถูกวัด Q_{fb} แรงดันพิกัดที่โหนดบัส $|V_n|$ ส่วนตัวแปรเอาต์พุตคือขนาดของแรงดันของอินเวอร์เตอร์ที่ต้องการ $|E|$

$$Q_{ref} = n(V_{max} - |V|) \quad (2)$$

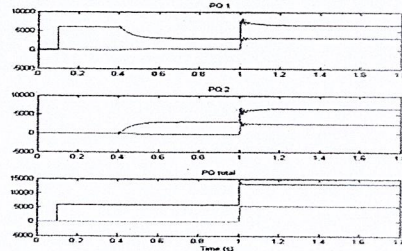
เมื่อ n คือ voltage droop coefficient

ส่วนประกอบของแรงดันที่จะควบคุม ΔV นั้นจะถูกสร้างมาจากค่าผิดพลาด ΔQ ผ่าน PI controller โดย $|E|$ นั้นคือผลรวมของแรงดันพิกัด $|V_n|$ และ ΔV

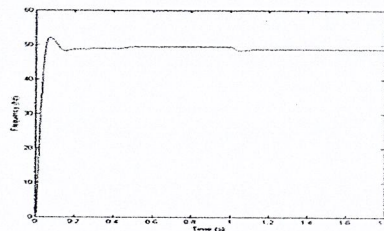
4.ผลการทดลอง

4.1 ผลการทดลองแบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 3 เฟส

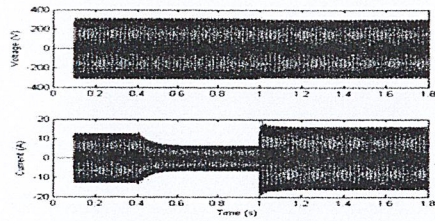
ต่อจากนี้จะเป็นรูปกราฟผลการทดลอง ซึ่งจะแสดงเฉพาะผลลัพธ์การจำลองที่กรณีเริ่มต้นด้วย load R 6kW ต่อมาเราจะเพิ่มโหลด RL 2kva at pf=0.8 ที่เวลา 1.5s และปลดโหลด R ออกที่เวลา 2s คงเหลือเพียงโหลดชนิด RL เพียงอย่างเดียว ตามรูปการแสดงผลที่ 6 - 8



รูปที่ 3 กำลังไฟฟ้าจริงและเสมือนของคอนเวอร์เตอร์ 1, 2 และกำลังไฟฟ้ารวมตามลำดับ



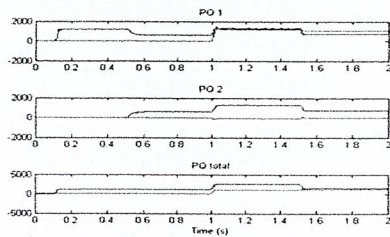
รูปที่ 4 ความถี่ที่จุดต่อร่วม (Point of common, PCC)



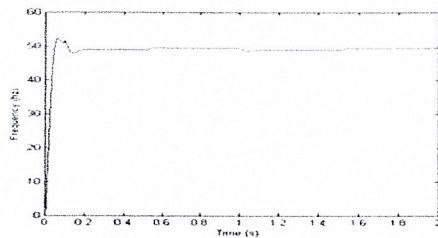
รูปที่ 5 แรงดันและกระแสที่จุดต่อร่วม PCC

4.2 ผลการทดลองแบบจำลองการทำงานของไมโครกริดในโหมด stand-alone ในระบบ 1 เฟส

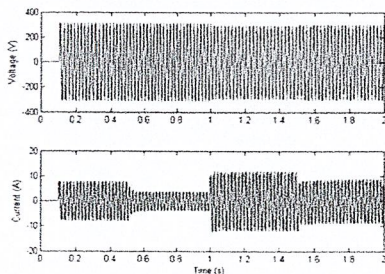
ต่อจากนี้จะเป็นรูปกราฟผลการทดลองซึ่งจะแสดงเฉพาะผลลัพธ์การจำลองที่กรณีเริ่มต้นด้วย load R 1.2kW ต่อมาเพิ่มโหลด RL ที่เวลา 1.5s และปลดโหลด R ออกที่เวลา 2s คงเหลือเพียงโหลดชนิด RL เพียงอย่างเดียว ตามกราฟแสดงผลที่ 9 - 11



รูปที่ 6 กำลังไฟฟ้าจริงและเสมือนของคอนเวอร์เตอร์ 1, 2 และกำลังไฟฟ้ารวมตามลำดับ



รูปที่ 7 ความถี่ที่จุดต่อร่วม (Point of common,PCC)



รูปที่ 8 แรงดันและกระแสที่จุดต่อร่วม PCC

4.3 ผลการทดลองของ Modbus

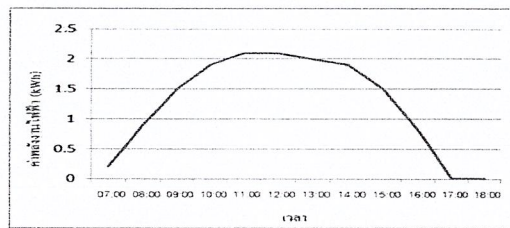
Measurement	Line 1	Line 2	Line 3	Line 1-2	Line 2-3	Line 3-1	Total
Voltage(V)	227.794			227.794		227.794	
Current(A)	4.04						
Active Power (W)	897.088						897.088
Reactive Power(W)	920.402						948.883
Power Factor	204.834						204.834
Phase Factor	976	1	1				976
Empower(W)	50.042	50.042	50.042				
Real Energy (kWh)	233434.2						233433.9
Reactive Energy (kWh)	61360.25						61360.29
Complex Energy (kWh)	241364.1						241363.8

รูปที่ 9 หน้าจอแสดงผลของโปรแกรม

4.4 ผลการจ่ายพลังงานไฟฟ้ารายวัน/เดือน

ค่าพลังงานไฟฟ้า (kWh) รายวัน

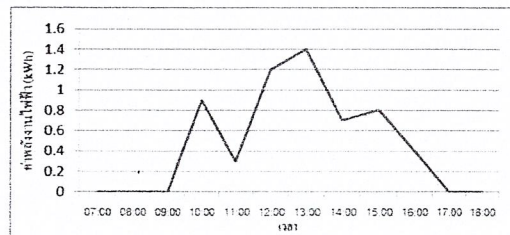
ค่าพลังงานไฟฟ้าพลังงานไฟฟ้า (kWh) รวมสูงสุดวันที่ 14 มกราคม 2557 ที่ 15.5 kWh



รูปที่ 10 กราฟค่าพลังงานไฟฟ้า (kWh) กับเวลา

จากกราฟจะเห็นว่าค่าพลังงานไฟฟ้า (kWh) ที่ได้สูงสุดคือ 2.1 kWh จะอยู่ในช่วงเวลา 11.00 -12.00 น.

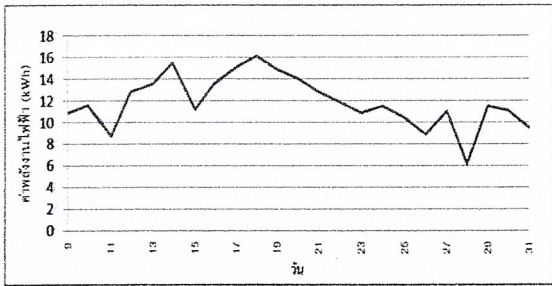
ค่าพลังงานไฟฟ้าพลังงานไฟฟ้า (kWh) รวมค่าสุดวันที่ 28 มกราคม 2556 ที่ 6.2 kWh



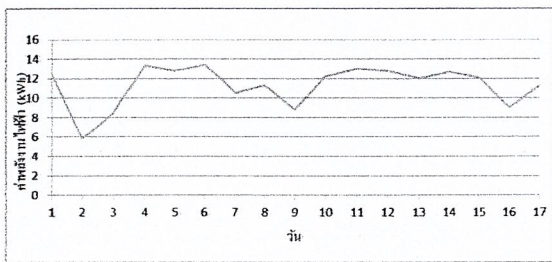
รูปที่ 11 กราฟค่าพลังงานไฟฟ้า (kWh) กับเวลา

จากกราฟจะเห็นว่าค่าพลังงานไฟฟ้า (kWh) ที่ได้สูงสุดคือ 1.4 kWh จะอยู่ในช่วงเวลา 12.00-13.00 น.

ค่าพลังงานไฟฟ้า (kWh) รายเดือน



รูปที่ 12 กราฟค่าพลังงานไฟฟ้า (kWh) ที่ได้ในแต่ละวัน ประจำเดือน มกราคม



รูปที่ 13 กราฟค่าพลังงานไฟฟ้า (kWh) ที่ได้ในแต่ละวัน ประจำเดือน กุมภาพันธ์

5 สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การศึกษาการจำลองระบบไมโครกริดในโหมด stand-alone ของระบบทั้งสามเฟสและเฟสเดียวของแหล่งจ่ายสองตัวที่มีพารามิเตอร์เดียวกัน พบว่าผลลัพธ์ที่ออกมาในก่อนข้างนำพหุใจ กล่าวคือเมื่อเรากำหนดสัดส่วนการแชร์โหลด droop coefficient ของมุมแรงดันและแรงดันไว้เท่ากันที่ 1:1 แล้ว พบว่ามีการแชร์อัตราส่วนกำลังไฟฟ้าจริงและ กำลังไฟฟ้รีแอกทีฟ ที่อัตราส่วนประมาณ 1 ต่อ 1 ภายใต้ระยะเวลาเข้าสู่สภาวะคงตัวที่เหมาะสม

ในส่วนของการแสดงผลข้อมูล จะเป็นการแสดงค่าที่วัดได้จากมิเตอร์ด้วยการส่งข้อมูลแบบ RS-485 แล้วแปลงข้อมูล Modbus มาอยู่ในโปรแกรมแสดงผลที่สร้างมาจากโปรแกรม Visual Studio 2010 โดยใช้ภาษา C# เพื่อค่าต่างๆจากมิเตอร์ในรูปแบบตัวเลข

5.2 ข้อเสนอแนะ

สำหรับโครงการฉบับนี้ควรจะมีการทดลองและจับเก็บข้อมูลการทดลองจริง เพื่อนำมาเปรียบเทียบกับกรจำลองด้วยโปรแกรม Matlab ว่าได้ผลลัพธ์ตามที่ออกแบบไว้หรือไม่

เอกสารอ้างอิง

- [1] นายจิรวุฒิ ภู่อสาร ,นายณัฐพล นันทวิชาวุทธสาร และณัฐพล ป้อมป้องกัน,“ระบบจัดการพลังงานสำหรับไมโครกริดโดยเอเจนท์”,ปริญญาานิพนธ์วิศวกรรมศาสตรบัณฑิต,สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,ปีการศึกษา 2554
- [2] Ritwik Majumder, Arindam Ghosh, Gerard Ledwich and Firuz Zare , “Control of parallel converters for load sharing with seamless transfer between grid connected and islanded modes”

ประวัติผู้เขียน



นายพุงศักดิ์ แซ่หลิว

- ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



นายพงศักรณม์ เดชะชลประเสริฐ

- ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



นายไพโรจน์ คอนอม

- ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง



นายมหาราช ราชสิทธิ์

-ปริญญาตรี สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง